

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ВЛАДИМИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УДК 658.562

**МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
ЗАДАЧ МЕТРОЛОГИИ.
СТАНДАРТИЗАЦИИ И СЕРТИФИКАЦИИ В
MATLAB**

СЕРГЕЕВ А.Г., ЛАТЫШЕВ М.В., МИЩЕНКО З.В.

Электронное учебное пособие

ВЛАДИМИР 2003

Содержание

Обзор возможностей ППП "Statistics Toolbox". Классификация функций. Часть 1.....	3
Обзор возможностей ППП "Statistics Toolbox". Классификация функций. Часть 2.....	15
Примеры решения задач в ППП "Statistics Toolbox"	28
Графическая подсистема в MATLAB.....	35
Описательная статистика в MATLAB 6.5	93
Линейный регрессионный анализ.....	155
Дисперсионный анализ	164
Планирование эксперимента	189
Статистический контроль качества	273
Список литературы.....	309

Обзор возможностей ППП "Statistics Toolbox".

Классификация функций. Часть 1.

Пакет прикладных программ (ППП) "Statistics Toolbox" ñ это набор программ решающий различные задачи по анализу данных методами математической статистики. Основное количество функций реализовано в виде *.m файлов за исключением некоторых функций являющихся встроенными функциями MATLAB.

Statistics Toolbox включает в себя более 200 М-файлов, решающих следующие группы задач:

1. Законы распределения случайных величин:
 - 1.1. оценка параметров закона распределения по экспериментальным данным;
 - 1.2. расчет функций распределения случайных величин;
 - 1.3. расчет функций плотности распределения случайных величин;
 - 1.4. расчет обратных функции распределения случайных величин (расчет квантилей законов распределения);
 - 1.5. генерация псевдослучайных чисел по заданному закону распределения;
 - 1.6. оценка математического ожидания и дисперсии по заданным виду закона распределения и его параметрам;
 - 1.7. Расчет логарифма функции максимального правдоподобия
2. Описательная статистика;
3. Статистические графики;
4. Статистический контроль качества;
5. Линейный регрессионный анализ;
6. Нелинейный регрессионный анализ;
7. Планирование эксперимента;
8. Анализ многомерных величин:
 - 8.1. Анализ главных компонент;
 - 8.2. Кластерный анализ;
 - 8.3. Канонический корреляционный анализ
 - 8.4. Многомерное шкалирование
 - 8.5. Линейный дискриминантный анализ
 - 8.6. Однофакторный многомерный дисперсионный анализ

9. Статистическая проверка гипотез;
10. Проверка статистических гипотез о согласии распределения экспериментальным данным;
11. Проверка непараметрических гипотез.
12. Вспомогательные функции.

Кроме перечисленных выше 13 групп задач имеющих непосредственное отношение к статистическим функциям в ППП "Statistics Toolbox" существуют дополнительные три группы функций:

13. Запись и чтение данных из файлов;
14. Демонстрационные примеры;
15. Файлы со статистическими данными.

Классификация функций по задачам приведена в соответствии с описанием ППП "Statistics Toolbox" 4.0 MATLAB версии R13.

Законы распределения случайных величин

ППП "Statistics Toolbox" использует 20 основных законов распределения для непрерывных и дискретных случайных величин. Причем для каждого из 20-ти законов распределения предусмотрены 7 функций решающих следующие задачи:

оценка параметров закона распределения по экспериментальным данным:

оценка параметров распределения (fitting) ñ **fit**,

расчет функции максимального правдоподобия (Maximum likelihood estimation) - **like**;

функция плотности распределения (Probability density function - **pdf**);

функция распределения (Cumulative distribution function - **cdf**);

функция обратная закону распределения (Inverse of the cumulative distribution function - **inv**), или иначе функция для расчета квантили распределения;

функция генерации случайных чисел (Random number generator - **rnd**);

функция расчета математического ожидания и дисперсии исходя из значений параметров закона распределения (Mean and variance as a function of the parameters - **stat**).

В скобках приведены соответствующие английские термины и через дефис аббревиатура от них, которая используется как суффикс после сокращенного названия вида закона распределения, для образования названия соответствующей функции.

Классификация функций распределения в MATLAB:

- непрерывные распределения:
 - бета (**Beta** - beta);
 - экспоненциальное (**Exponential** - exp)
 - гамма (**Gamma** - gam)
 - логнормальное (**Lognormal** - logn)
 - нормальное (**Normal** - norm)
 - Релея (**Rayleigh** - rayl)
 - равномерное (**Uniform (Continuous)** - unif)

- Вейбулла (**Weibull** - weib)
- основные статистические законы распределения непрерывных случайных величин:
 - распределение хи-квадрат - χ^2 (**Chi-Square** - chi2);
 - смещенное хи-квадрат распределение (**Noncentral Chi-Square** - nch2);
 - распределение Фишера (**F** - f);
 - смещенное распределение Фишера (**Noncentral F** - ncf);
 - распределение Стьюдента \tilde{t} (**Student's t** - t);
 - смещенное распределение Стьюдента (**Noncentral t** - nct);
- дискретные распределения:
 - биномиальное (**Binomial** - bino);
 - дискретное равномерное (**Uniform Discrete** - unid);
 - геометрическое (**Geometric** - geo);
 - гипергеометрическое (**Hypergeometric** - hyge);
 - отрицательное биномиальное (**Negative Binomial** - nbin);
 - распределение Пуассона (**Poisson** - poiss).

В скобках приведены соответствующие английские термины и через дефис аббревиатура от них, которая используется перед названным выше суффиксом для образования имени соответствующей функции. Полужирным шрифтом выделены буквы, используемые для формирования аббревиатуры. Например, для нормального закона распределения функция плотности вероятности примет вид \tilde{t} normpdf, закон распределения (функция распределения) \tilde{t} normcdf, функция генерации значений случайной величины распределенной по нормальному закону \tilde{t} normrnd и т.д.

Следующие таблицы 1-7 содержат название функций и их краткое описание для первой группы функций, связанной с законами распределения случайных величин.

Таблица 1.

Оценка параметров закона распределения по экспериментальным данным

Название функции	Описание функции
betafit	Оценка параметров бета распределения
binofit	Оценка параметров биномиального распределения
nbinfit	Оценка параметров отрицательного биномиального

	распределения
expfit	Оценка параметров экспоненциального распределения
gamfit	Оценка параметров гамма распределения
normfit	Оценка параметров нормального распределения
poissfit	Оценка параметров распределения Пуассона
raylf	Оценка параметров распределения Релея
unifit	Оценка параметров равномерного распределения
weibfit	Оценка параметров распределения Вейбулла
mle	Расчет функции максимального правдоподобия

Для оценки параметров законов распределения используется метод максимального правдоподобия.

Следует отметить, что все функции данной группы делятся на две подгруппы:

- функции оценки параметров законов распределения;
- функции расчета логарифма функции максимального правдоподобия.

Причем функции последней группы работают в паре с mle и функцией расчета значения критерия максимального правдоподобия.

Таблица 2.

Законы распределения случайных величин

Название функции	Описание функции
betacdf	Бета распределение
binocdf	Биномиальное распределение
cdf	Параметризованная функция распределения
chi2cdf	Функция распределения хи-квадрат
expcdf	Экспоненциальное распределение
ecdf	Эмпирическая функция распределения (на основе оценки Каплана-Мейера)
fcdf	Распределение Фишера
gamcdf	Гамма распределение
geocdf	Геометрическое распределение

hygecdf	Гипергеометрическое распределение
logncdf	Логнормальное распределение
nbincdf	Отрицательное биномиальное распределение
ncfcdf	Смещенное распределение Фишера
nctcdf	Смещенное распределение Стьюдента
ncx2cdf	Смещенное хи-квадрат распределение
normcdf	Нормальное распределение
poisscdf	Распределение Пуассона
raylcdf	Распределение Релея
tcdf	Распределение Стьюдента
unidcdf	Дискретное равномерное распределение
unifcdf	Непрерывное равномерное распределение
weibcdf	Распределение Вейбулла

Следует выделить параметризованную функцию распределения `ncdf`. Данная функция распределения позволяет рассчитать значение любой из 20 функций распределения. Вид функции задается как локальный параметр. Как можно увидеть в тексте файла `cdf.m` эта функция лишь вызывает одну из указанных 20 функций и передает ей параметры закона распределения.

Таблица 3.

Функции плотности распределения случайных величин

Название функции	Описание функции
betapdf	Бета распределение
binopdf	Биномиальное распределение
chi2pdf	Функция распределения хи-квадрат
exppdf	Экспоненциальное распределение
fpdf	Распределение Фишера
gampdf	Гамма распределение
geopdf	Геометрическое распределение
hygepdf	Гипергеометрическое распределение

lognpdf	Логнормальное распределение
nbinpdf	Отрицательное биномиальное распределение
ncfpdf	Смещенное распределение Фишера
nctpdf	Смещенное распределение Стьюдента
ncx2pdf	Смещенное хи-квадрат распределение
normpdf	Нормальное распределение
poisspdf	Распределение Пуассона
raylpdf	Распределение Релея
pdf	Параметризованная функция плотности распределения
tpdf	Распределение Стьюдента
unidpdf	Дискретное равномерное распределение
unifpdf	Непрерывное равномерное распределение
weibpdf	Распределение Вейбулла

Параметризованная функция плотности распределения $f(x)$ pdf является аналогом функции распределения cdf.

Таблица 4.

Обратные функции распределения случайных величин

Название функции	Описание функции
betainv	Бета распределение
binoinv	Биномиальное распределение
chi2inv	Функция распределения хи-квадрат
expinv	Экспоненциальное распределение
finv	Распределение Фишера
gaminv	Гамма распределение
geoinv	Геометрическое распределение
hygeinv	Гипергеометрическое распределение
icdf	Параметризованная обратная функция распределения
logninv	Логнормальное распределение

nbininv	Отрицательное биномиальное распределение
ncfinv	Смещенное распределение Фишера
nctinv	Смещенное распределение Стьюдента
ncx2inv	Смещенное хи-квадрат распределение
norminv	Нормальное распределение
poissinv	Распределение Пуассона
raylinv	Распределение Релея
tinv	Распределение Стьюдента
unidinv	Дискретное равномерное распределение
unifinv	Непрерывное равномерное распределение
weibinv	Распределение Вейбулла

Параметризованная обратная функция распределения \hat{y} icdf является аналогом функции распределения cdf.

Таблица 5.

Генерация псевдослучайных чисел по заданному закону распределения

Название функции	Описание функции
betarnd	Бета распределение
binornd	Биномиальное распределение
chi2rnd	Функция распределения хи-квадрат
exprnd	Экспоненциальное распределение
frnd	Распределение Фишера
gamrnd	Гамма распределение
geornd	Геометрическое распределение
hygernd	Гипергеометрическое распределение
iwishrnd	Обратная матрица случайных чисел распределения Уишарта
lognrnd	Логнормальное распределение
mvnrnd	Многомерное нормальное распределение

mvtrnd	Многомерное распределение Стьюдента
nbinrnd	Отрицательное биномиальное распределение
ncfrnd	Смещенное распределение Фишера
nctrnd	Смещенное распределение Стьюдента
ncx2rnd	Смещенное хи-квадрат распределение
normrnd	Нормальное распределение
poissrnd	Распределение Пуассона
random	Параметризованная функция генерации псевдослучайных чисел
raylrnd	Распределение Релея
trnd	Распределение Стьюдента
unidrnd	Дискретное равномерное распределение
unifrnd	Непрерывное равномерное распределение
weibrnd	Распределение Вейбулла
wishrnd	Матрица случайных чисел распределения Уишарта

Параметризованная обратная функция распределения \hat{n} `random` является аналогом функции распределения `cdf`.

Как видно из табл. 5 кроме 20 основных законов добавились еще 2 функции: `mvnrnd` – функция генерации псевдослучайных чисел многомерного нормального распределения, `mvtrnd` – функция генерации псевдослучайных чисел многомерного распределения Стьюдента, `wishrnd`, `iwishrnd` \hat{n} функции генерации матрицы случайных чисел распределенных по закону Уишарта.

Таблица 6.

Оценка математического ожидания и дисперсии по заданному закону распределения и его параметрам

Название функции	Описание функции
betastat	Бета распределение
binostat	Биномиальное распределение
chi2stat	Функция распределения хи-квадрат
expstat	Экспоненциальное распределение

fstat	Распределение Фишера
gamstat	Гамма распределение
geostat	Геометрическое распределение
hygestat	Гипергеометрическое распределение
lognstat	Логнормальное распределение
nbinstat	Отрицательное биномиальное распределение
ncfstat	Смещенное распределение Фишера
nctstat	Смещенное распределение Стьюдента
ncx2stat	Смещенное хи-квадрат распределение
normstat	Нормальное распределение
poissstat	Распределение Пуассона
raylstat	Распределение Релея
tstat	Распределение Стьюдента
unidstat	Дискретное равномерное распределение
unifstat	Непрерывное равномерное распределение
weibstat	Распределение Вейбулла

Таблица 7.

Расчет логарифма функции максимального правдоподобия

Название функции	Описание функции
betalike	Расчет логарифма функции максимального правдоподобия бета распределения
gamlike	Расчет логарифма функции максимального правдоподобия гамма распределения
normlike	Расчет логарифма функции максимального правдоподобия нормального распределения
weiblike	Расчет логарифма функции максимального правдоподобия распределения Вейбулла
nbinline	Расчет логарифма функции максимального правдоподобия отрицательного биномиального распределения

ППП "Statistics Toolbox" включает в себя 20 основных распределений, но кроме указанных выше законов достаточно часто используются [1]:

- полунормальное распределение;
- распределение Максвелла;
- распределение модуля многомерного нормального вектора;
- распределение Парето;
- распределение Эрланга;
- распределение Лапласа;
- логнормальное распределение вида $f(x) = \frac{1}{x\sigma \lg(10\sqrt{2\pi})} e^{-\frac{(\lg(x)-\mu)^2}{2\sigma^2}}$, где μ – математическое ожидание, σ – среднее квадратическое отклонение;
- распределение Коши;
- логистическое распределение;
- смещенное распределение Вейбулла;
- распределение минимального значения;
- распределение максимального значения;
- двойное показательное распределение;
- распределение Накагами;
- бета распределение 1 рода;
- бета распределение 2 рода;
- стандартное бета распределение 2 рода;
- распределение Sb-Джонсона;
- распределение SI-Джонсона;
- распределение Su-Джонсона.

Наличие максимального количества распределений в статистической системе особенно важно при решении задачи идентификации закона распределения выборки.

Описательная статистика

Функции описательной статистики позволяют определить набор характеристик экспериментальных одно и многомерных данных. Перечень названий функций описательной статистики и их описаний приведен в табл. 8.

Таблица 8.

Функции описательной статистики

Название функции	Описание функции
bootstrp	Бутстреп оценки. Оценка статистик для данных с дополненным объемом выборки посредством математического моделирования
corrcoef	Оценка коэффициента корреляции (функция MATLAB)
cov	Оценка матрицы ковариаций (функция MATLAB)
crosstab	Кросстабуляция для нескольких векторов с положительными целыми элементами
geomean	Среднее геометрическое
grpstats	Сводные статистики по группам
harmmean	Среднее гармоническое
iqr	Разность между 75% и 25% квантилями или между 3-й и 1-ой квантилями
kurtosis	Оценка коэффициента эксцесса (в отечественной литературе коэффициент эксцесса определяется как $\beta_2 = \text{kurtosis} - 3$)
mad	Среднее абсолютное отклонение от среднего значения
mean	Среднее арифметическое (функция MATLAB)
median	Медиана (функция MATLAB)
moment	Оценка центрального момента. Порядок момента задается как аргумент функции.
nanmax	Максимальное значение в выборке. Нечисловые значения в выборке игнорируются.
nanmean	Среднее арифметическое выборки. Нечисловые значения в выборке игнорируются.
nanmedian	Медиана выборки. Нечисловые значения в выборке игнорируются.
nanmin	Минимальное значение в выборке. Нечисловые значения в выборке игнорируются.
nanstd	Оценка среднего квадратического отклонения выборки. Нечисловые значения в выборке игнорируются.
nansum	Сумма элементов выборки. Нечисловые значения в выборке игнорируются.

prctile	Выборочная процентная точка (процентиль)
range	Размах выборки
skewness	Оценка коэффициента асимметрии
std	Оценка среднего квадратического отклонения (функция MATLAB)
tabulate	Определение частот целых положительных элементов вектора случайных значений.
trimmean	Оценка среднего арифметического значения, находящаяся с игнорированием заданного процента минимальных и максимальных элементов в выборки
var	Оценка дисперсии

Как можно видеть из таблицы 7, группа функций описательной статистики позволяет определить все основные характеристики для одномерной выборки:

- оценки начального момента 1 порядка;
- 2 и 4 центральные моменты;
- центральный момент произвольного порядка;

для многомерной выборки:

- ковариационный момент;
- коэффициент парной корреляции.

Кроме того, следует отметить возможность расчета оценок первого начального и второго центрального моментов для векторов и матриц, содержащих нечисловые элементы.

Обзор возможностей ППП "Statistics Toolbox". Классификация функций. Часть 2.

Статистические графики

Любая программная статистическая система характеризуется довольно большим набором специфических для математической статистики графиков. Это связано с необходимостью визуального поиска и представления какой либо закономерности по выборочным данным при наличии погрешностей при сборе исходных данных. Несмотря на то, что MATLAB существенно уступает таким системам как Statistica, SPSS, S-Plus, тем не менее набор функций статистических графиков MATLAB позволяет строить почти все наиболее часто используемые статистические графики. А наличие возможностей программирования графической подсистемы MATLAB позволяет реализовывать и другие виды графиков имеющиеся в Statistica, SPSS, S-Plus. Список функций статистических графиков приведен в табл. 9.

Таблица 9.

Функции статистических графиков

Название функции	Описание функции
<code>boxplot</code>	График "Ящик с усами". График 0%, 25%, 50%, 75%, 100% перцентилей выборки
<code>cdfplot</code>	График кумулятивной кривой по эмпирическим данным
<code>fsurfht</code>	Контурный график заданной функции. Операция построения графика выполняется интерактивно.
<code>gline</code>	Операция прорисовки прямой линии в текущем графике
<code>gname</code>	Нанесение меток на график
<code>gplotmatrix</code>	Матрица графиков рассеяния группированных по общей переменной
<code>gscatter</code>	График рассеяния двух переменных группированных по значениям третьей переменной
<code>lsline</code>	График рассеяния двух переменных с линией регрессии по методу наименьших квадратов
<code>normplot</code>	Нормальный вероятностный график
<code>qqplot</code>	График "квантиль-квантиль" для двух выборок
<code>refcurve</code>	Построение полиномиальной кривой на текущий график
<code>refline</code>	Построение прямой на текущий график

surfht	Контурный график по матрице данных
weibplot	Вероятностный график Вейбулла

Статистический контроль качества

Статистический контроль качества является разделом теории вероятностей и математической статистики и решает следующие основные задачи:

- контроль за состоянием технологических процессов и технических объектов в зависимости от времени с помощью контрольных карт,
- планирование выборочного контроля качества,
- оценка точности и стабильности технологического процесса.

В ППП "Statistics Toolbox" предусмотрено решение 1 и 3 из перечисленных выше задач. Функции контроля качества позволяют строить контрольные карты среднего арифметического значения, среднего квадратического отклонения и экспоненциально взвешенного скользящего среднего. Для решения 3 задачи в ППП "Statistics Toolbox" предусмотрен расчет коэффициентов воспроизводимости C_p , смещения среднего C_{pk} технологического процесса, а также построение графиков воспроизводимости процесса, гистограмм с границами допусков и совмещенной с гистограммой функции плотности нормального распределения. Описание функций приведено в табл. 10.

Таблица 10.

Функции статистического контроля качества

Название функции	Описание функции
capable	Расчет индексов воспроизводимости процесса C_p , C_{pk}
capaplot	График воспроизводимости процесса
ewmaplot	Контрольная карта экспоненциально взвешенного среднего
histfit	Гистограмма по негруппированным экспериментальным данным с наложенной на нее кривой функции плотности распределения нормального закона
normspec	График функции плотности нормального закона с наложенными границами допусков контролируемого параметра
schart	Контрольная карта среднего квадратического отклонения
xbarplot	Контрольная карта среднего арифметического

Следует отметить отсутствие множества распространенных контрольных карт:

- карты размаха (R-карты);
- карты многомерных наблюдений (T^2 Хоттеллинга);
- регрессионной контрольной карты;
- контрольных карт числа: n , c и доли: np , u дефектов в выборке;
- контрольной карты накопленной суммы (CUSUM);
- контрольных карт индивидуальных значений, скользящего размаха и скользящего среднего.

Однако, разработанные алгоритмы для среднего арифметического значения, среднего квадратического отклонения и экспоненциально взвешенного скользящего среднего являются в достаточной степени общими, что позволяет реализовать на их основе любую из вышеперечисленных контрольных карт.

Линейный регрессионный анализ Функции группы линейного регрессионного анализа решают следующие задачи:

- дисперсионного анализа,
- собственно линейного регрессионного анализа - определение точечных и интервальных оценок параметров линейной полиномиальной модели, множественной регрессии,
- пошаговой регрессии,
- регрессионного анализа на основе обобщенной линейной модели,
- регрессионного анализа на основе гребневых оценок,
- робастной оценки параметров регрессионной модели.

Список идентификаторов функций и их краткое описание приведены в табл. 11. Таблица 11.

Функции линейного регрессионного анализа

Название функции	Описание функции
anova1	Однофакторный дисперсионный анализ
anova2	Двухфакторный дисперсионный анализ
anovan	Многофакторный дисперсионный анализ
aocool	Однофакторный анализ ковариационных моделей. Выходными параметрами функции являются: <ul style="list-style-type: none"> • Интерактивный график исходных данных

	<p>линейных математических моделей;</p> <ul style="list-style-type: none"> • Таблица однофакторного дисперсионного анализа • Таблица с оценками параметров математических моделей
<code>dummyvar</code>	Условное кодирование переменных. Функция возвращает матрицу единиц и нулей содержащую число колонок равное сумме чисел возможных значений в столбцах исходной матрицы. Единицы и нули характеризуют отсутствие или наличие определенного значения в каждой колонки исходной матрицы.
<code>friedman</code>	Тест Фридмана (непараметрический двухфакторный дисперсионный анализ Фридмана)
<code>glmfit</code>	Определение параметров обобщенной линейной модели
<code>glmval</code>	Прогнозирование с использованием обобщенной линейной модели
<code>kruskalwallis</code>	Тест Краскала-Уоллиса (непараметрический однофакторный дисперсионный анализ)
<code>leverage</code>	Оценка степени влияния отдельных наблюдений в исходном многомерном множестве данных на значения параметров линии регрессии.
<code>lscov</code>	Линейная регрессия (метод наименьших квадратов) при заданной матрице ковариаций (встроенная функция MATLAB)
<code>manova1</code>	Однофакторный многомерный дисперсионный анализ
<code>manovacluster</code>	Дендрограмма, показывающая группировку исходных данных в кластеры по средним значениям. В качестве исходных данных используются выходные данные однофакторного многомерного дисперсионного анализа (<code>manova1</code>)
<code>multcompare</code>	Множественное сравнение оценок средних, параметров линии регрессии и т.д. В качестве входных параметров используются выходные параметры функций <code>anova1</code> , <code>anova2</code> , <code>anovan</code> , <code>aocool</code> , <code>friedman</code> , <code>kruskalwallis</code> .
<code>polyconf</code>	Определение доверительных интервалов для линии регрессии

polyfit	Полиномиальная регрессия (встроенная функция MATLAB)
polyval	Прогноз с использованием полиномиальной регрессии (встроенная функция MATLAB)
rcoplot	График остатков
regress	Множественная линейная регрессия
regstats	Функция диагностирования линейной множественной модели. Графический интерфейс.
ridge	Линейная регрессия с применением гребневых оценок (ридж-регрессия)
rstool	Интерактивный подбор и визуализация поверхности отклика
robustfit	Робастная оценка параметров регрессионной модели
stepwise	Пошаговая регрессия (графический интерфейс пользователя)
x2fx	Преобразовывает матрицу факторов в матрицу коэффициентов линий регрессии

Нелинейный регрессионный анализ В отличие от группы функций линейного регрессионного анализа в этом случае решаются задачи расчета точечных и интервальных оценок параметров нелинейной регрессионной модели. Список функций приведен в табл. 12.

Таблица 12. Функции нелинейного регрессионного анализа

Название функции	Описание функции
lsqnonneg	Функция реализует метод наименьших квадратов и возвращает только неотрицательные значения параметров модели (встроенная функция MATLAB)
nlinfit	Нелинейный метод наименьших квадратов (метод Гаусса-Ньютона)
nlintool	График прогнозируемых значений
nlparci	Вектор доверительных интервалов для параметров модели
nlpredci	Прогнозируемые значения и их доверительные интервалы

Планирование эксперимента Методы планирования эксперимента позволяют существенно снизить затраты на проведение измерений при

моделировании характеристик сложных объектов и процессов. В ППП "Statistics Toolbox" предусмотрено построение следующих планов:

- полного факторного плана;
- дробного факторного плана;
- центрального композиционного плана;
- планов Бокса-Бенкена;
- планы на основе латинских квадратов;
- D-оптимального плана.

Таблица 13. Функции планирования эксперимента

Название функции	Описание функции
bbdesign	Планы Бокса-Бенкена
candexch	D-оптимальный план (на основе алгоритма перестановки строк для формирования множества возможных значений)
candge	Генерирует множество возможных сочетаний факторов соответствующих D-оптимальному плану
ccdesign	Центральный композиционный план
cordexch	Функция для определения точного D-оптимального плана эксперимента на основе алгоритма обмена координатами
daugment	Определение матрица плана дополняющую матрицу заданного плана до D-оптимального
dcovary	Функция для построения D-оптимального блочного плана
ff2n	Определение плана полного факторного эксперимента для факторов имеющих 2 уровня
fracfact	Функция для формирования двухуровневого дробного факторного плана
fullfact	Функция формирования плана полного факторного эксперимента для числа уровней факторов задаваемых пользователем
hadamard	Матрица Адамара. Матрица Адамара соответствует плану дробного факторного эксперимента для факторов, каждый из которых задан на отрезке [-1 1]. И служит для построения линейной регрессионной модели. (Встроенная функция MATLAB)

lhsdesign	План на основе латинских квадратов
lhsnorm	Латинские квадраты для многомерной нормальной выборки
rowexch	Функция для определения точного D-оптимального плана на основе алгоритма обмена строк

Анализ многомерных величин

Анализ многомерных величин включает в себя следующие основные виды функций: анализ главных компонент; кластерный анализ; канонический корреляционный анализ; многомерное шкалирование; линейный дискриминантный анализ; однофакторный многомерный дисперсионный анализ. Классификация названных функций приведена в табл. 14-17 в соответствии с классификацией принятой в ППП "Statistics Toolbox" 4.0 MATLAB версии R13.

Кластерный анализ Группу кластерного анализа составляют 10 функций, позволяющие рассчитать парные расстояния между группируемыми объектами, описываемыми векторами, объединить объекты в иерархическое бинарное дерево кластеров и отобразить взаимосвязь объектов объединенных в кластеры в виде дендрограммы.

Парные расстояния между объектами могут быть вычислены на основе: евклидова расстояния, нормализованного евклидова расстояния, расстояния Махаланобиса, расстояния Хемминга, расстояния в метрике Минковского. Для кластеризации объектов в иерархическое дерево могут быть использованы следующие алгоритмы:

- "ближайшего соседа";
- "дальнего соседа";
- "средней связи";
- центроидный алгоритм;
- пошаговый алгоритм.

Таблица 14.

Функции кластерного анализа

Название функции	Описание функции
cluster	Деление иерархического дерева кластеров (группировка выходных данных функции linkage) на отдельные кластеры
clusterdata	Группировка матрицы исходных данных в кластеры
cophenet	Расчет коэффициента качества разбиения исходных

	данных на кластеры (этот коэффициент можно рассматривать как аналог коэффициента корреляции, чем его значение ближе к 1, тем лучше выполнено разбиение на кластеры)
dendrogram	Дендрограмма кластеров
inconsistent	Расчет коэффициентов несовместимости для каждой связи в иерархическом дереве кластеров и может использоваться как оценка качества разбиения на кластеры
kmeans	Кластеризация на основе внутригрупповых средних
linkage	Формирование иерархического дерева бинарных кластеров
pdist	Расчет парных расстояний между объектами (векторами) в исходном множестве данных
silhouette	График силуэта кластеров
squareform	Преобразование вектора выходных данных функции pdist в симметричную квадратную матрицу

Функции снижения размерности задачи

Функции этой группы включают 2 вида анализ: кластерный анализ и анализ главных компонент. Анализ главных компонент является наиболее часто используемым методом анализа многомерных случайных величин. Он позволяет снизить размерность задачи за счет перехода к новым координатам, являющимся линейной комбинацией исходных признаков. Практически любая статистическая система содержит процедуру анализа главных компонент. В ППП "Statistics Toolbox" для этой цели используются 3 функции описанные в табл. 15. За проведение кластерного анализа отвечает одна функция введенная в 4.0 версии "Statistics Toolbox".

Таблица 15. Функции снижения размерности задачи

Название функции	Описание функции
factoran	Факторный анализ
pcasov	Функция служит для реализации метода главных компонент по заданной в качестве входного параметра матрице ковариаций
pcares	Функция служит для определения остатка после удаления заданного количества главных компонент

princomp	Функция служит для реализации метода главных компонент по заданной в качестве входного параметра матрице исходных значений
----------	--

Кроме описанных выше кластерного анализа, многомерного дисперсионного анализа и анализа главных компонент дискриминантный анализ является широко распространенным методом анализа и классификации многомерных наблюдений. Дискриминантный анализ служит для классификации полученных на соответствующие группы. Наибольшее распространение получил линейный дискриминантный анализ, реализованный в ППП "Statistics Toolbox". Список функций линейного дискриминантного анализа, канонического корреляционного анализа, многомерное шкалирование, однофакторного многомерного дисперсионного анализа приведен в табл. 16.

Таблица 16. Функции анализа многомерных случайных величин

Название функции	Описание функции
barttest	Тест Бартлета
canoncorr	Канонический корреляционный анализ
cmdscale	Классическое многомерное шкалирование
classify	Линейный дискриминантный анализ
mahal	Функция определяет расстояния Махаланобиса между строками двух матриц, являющихся входными параметрами.
manova1	Однофакторный многомерный дисперсионный анализ
procrustes	Ортогональное вращение, позволяющее поставить в прямое соответствие одно множество точек другому

Нелинейный регрессионный анализ на основе графа возможных решений

Этот раздел полностью был введен в ППП "Statistics Toolbox" 4.0 и состоит из функций, приведенных в табл. 16.

Таблица 16. Функции нелинейного регрессионного анализа на основе графа возможных решений

Название функции	Описание функции
treedisp	Отображает граф возможных решений
treefit	Построение графа возможных решений на основе исходных данных

treeprune	Исключение незначимых решений в графе возможных решений
treetest	Оценка погрешности узлов графа возможных решений
treeval	Оценка параметров регрессионной модели с использованием графа возможных решений

Проверка статистических гипотез

Три группы задач: статистическая проверка гипотез, проверка статистических гипотез о согласии распределения экспериментальным данным и проверка непараметрических гипотез, содержат функции для решения следующих задач математической статистики:

- проверку параметрических гипотез;
- проверку непараметрических гипотез.

Список идентификаторов этих функций и их краткое описание приведены в табл. 17-19.

Таблица 17. Статистическая проверка гипотез

Название функции	Описание функции
ranksum	Ранговый тест Вилкоксона для проверки однородности двух генеральных совокупностей
signrank	Знаковый тест Вилкоксона для проверки гипотезы о равенстве медиан двух выборок
signtest	Знаковый тест для проверки гипотезы о равенстве медиан двух выборок
ttest	t-test для одной выборки. Проверка гипотезы о равенстве (или неравенстве) математического ожидания выборки заданному значению при условии, что величина дисперсии неизвестна. Закон распределения выборки нормальный.
ttest2	t-test для двух выборок. Проверка гипотезы о равенстве (или неравенстве) математических ожиданий двух выборок при условии, что величины дисперсий выборок неизвестны и равны. Закон распределения выборки нормальный.
ztest	Z-тест. Проверка гипотезы о равенстве (или неравенстве) математического ожидания выборки заданному значению при условии, что известна величина дисперсии. Закон распределения выборки нормальный.

Таблица 18. Проверка статистических гипотез о согласии распределения экспериментальным данным

Название функции	Описание функции
jbtest	Тест на соответствие выборки нормальному распределению с неопределенными параметрами нормального распределения. Этот тест является асимптотическим и не может быть использован на малых выборках. Для проверки гипотезы о соответствии выборки нормальному распределению на малых выборках необходимо использовать функцию <code>lillietest</code> .
kstest	Тест Колмогорова-Смирнова на соответствие выборки заданному распределению
kstest2	Тест Колмогорова-Смирнова на соответствие распределений двух выборок
lillietest	Тест на соответствие выборки нормальному распределению. Параметры нормального распределения рассчитываются исходя из значений элементов в выборке.

Таблица 19. Проверка непараметрических гипотез

Название функции	Описание функции
friedman	Тест Фридмана (непараметрический двухфакторный дисперсионный анализ Фридмана)
kruskalwallis	Тест Краскала-Уоллиса (непараметрический однофакторный дисперсионный анализ)
ksdensity	Подгонка функции плотности вероятности по экспериментальным данным
ranksum	Ранговый тест Вилкоксона для проверки однородности двух генеральных совокупностей
signrank	Знаковый тест Вилкоксона для проверки гипотезы о равенстве медиан двух выборок
signtest	Знаковый тест для проверки гипотезы о равенстве медиан двух выборок

Запись и чтение данных из файлов

Обработка большого массива исходных данных при статистической обработке приводит к необходимости их представления, хранения и редактирования в виде двумерной таблицы. Поскольку собственно MATLAB не имеет табличного процессора, то одним из способов обойти эту проблему

является использование текстового файла, в котором данные разделены пробелами, знаками табуляции и символами переноса строки. Для записи и чтения таких файлов служат функции приведенные в табл. 20.

Таблица 20. Запись и чтение данных из файлов

Название функции	Описание функции
caseread	Функция для чтения данных из текстового файла. Возвращает матрицу символов из текстового файла
casewrite	Функция для записи строковой матрицы в текстовый файл
tblread	Функция для чтения табличных данных из текстового файла
tblwrite	Функция для записи табличных данных из текстового файла
tdfread	Функция для чтения табличных данных разделенных знаком табуляции в строке из текстового файла

Демонстрационные примеры

Для демонстрации возможностей ППП "Statistics Toolbox", работы некоторых функций и преимущество одних методов обработки данных перед другими, фирмой MathWorks были разработаны отдельные демонстрационные программы. Их список приведен в табл. 21

Таблица 21. Таблица демонстрационных примеров

Название функции	Описание функции
aocool	Интерактивное средство ковариационного анализа
disttool	Интерактивное средство для исследования функций распределения случайных величин
glmdemo	Пример использования обобщенной линейной модели
randtool	Интерактивное средство для генерации псевдослучайных чисел
polytool	Интерактивное определение параметров полиномиальной модели
rsmdemo	Интерактивное моделирование химической реакции и нелинейный регрессионный анализ
robustdemo	Интерактивное средство для сравнения методов МНК и робастной регрессии

Вспомогательные функции

В ППП "Statistics Toolbox" 4.0 предусмотрены несколько функций выполняющие вспомогательные расчеты при проведении статистического анализа. Список и их назначение приведены в таблице 22.

Таблица 22. Таблица вспомогательных функций

Название функции	Описание функции
combnk	Вычисляет количество комбинаций которыми можно выбрать k объектов из n
grp2idx	Преобразование группирующей переменной в индексы массива
hougen	Функция прогнозирования для модели Хогена
tiedrank	Расчет ранга выборки с учетом ее объема
zscore	Выполняет нормализацию матрицы по колонкам. Приводит значения по колонкам матрицы к нормальным с 0 математическим ожиданием и единичной дисперсией.

Кроме названных выше файлов программ (m-файлов) и демонстрационных примеров каталог ППП "Statistics Toolbox" содержит следующие файлы статистических данных:

- census.mat;
- cities.mat;
- discrim.mat;
- gas.mat;
- hald.mat;
- hogg.mat;
- lawdata.mat;
- mileage.mat;
- moore.mat;
- parts.mat;
- popcorn.mat;
- polydata.mat;
- reaction.mat;
- sat.dat.

Примеры решения задач в ППП "Statistics Toolbox"

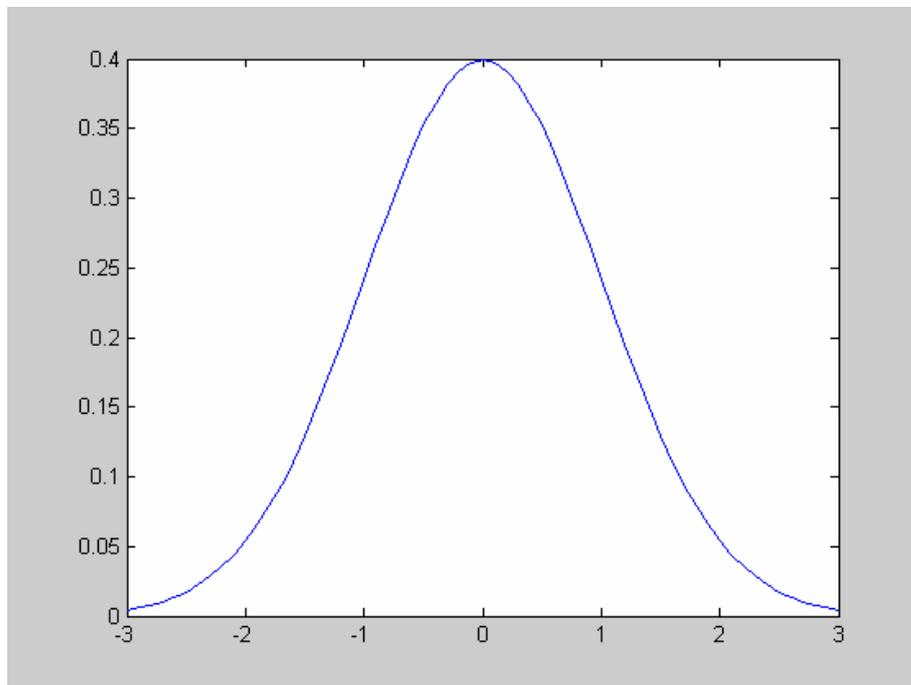
Рассмотрим примеры использования функций ППП "Statistics Toolbox" на примерах:

- использования функций распределения нормального закона;
- анализа одномерной случайной величины;
- корреляционного и линейного регрессионного анализов.

Функции распределения нормального закона

Функция плотности вероятности нормального закона f с математическим ожиданием $Mx=0$ и средним квадратическим отклонением $\sigma=1$.

```
>> Mx=0;  
>> sigma=1;  
>> x=-3:0.1:3;  
>> f=normpdf(x,Mx,sigma);  
>> plot(x,f)
```



Функция распределения вероятностей нормального закона F с математическим ожиданием $Mx=0$ и средним квадратическим отклонением $\sigma=1$.

```
>> F=normcdf(x,Mx,sigma);  
>> plot(x,F)
```

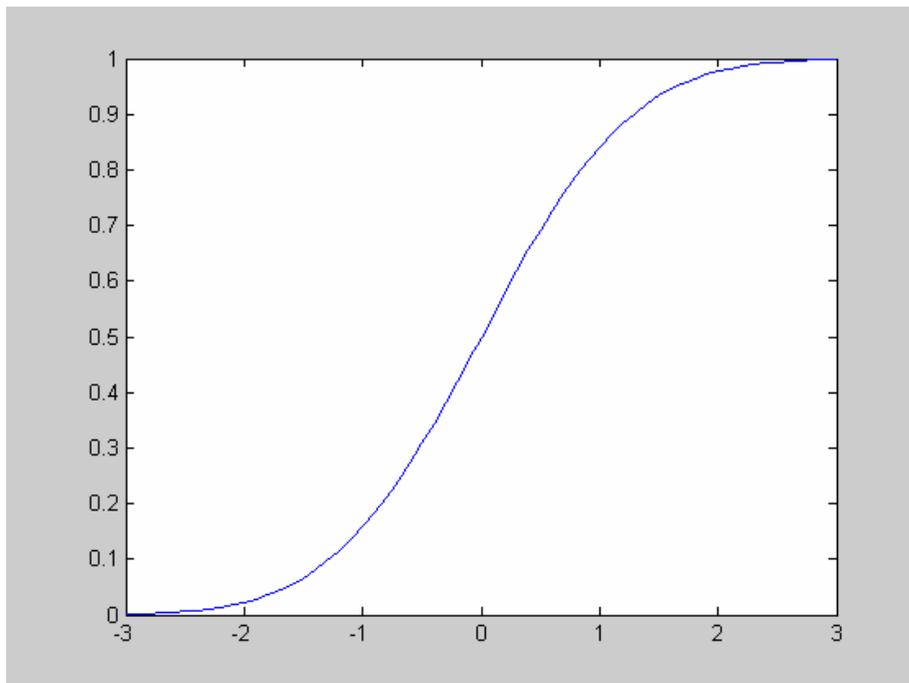
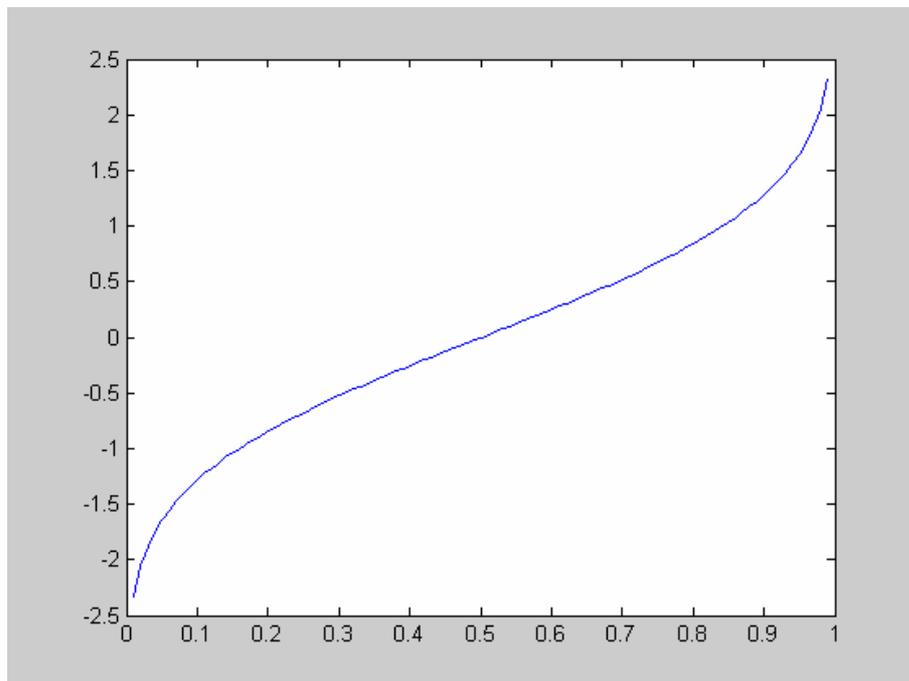


График значения квантили нормального распределения y в зависимости от вероятности появления заданного значения случайной величины p . Параметры нормального распределения $Mx=0$ и $\sigma=1$.

```
>> p=0:0.01:1;
>> y=norminv(p,Mx,sigma);
>> plot(p,y)
```



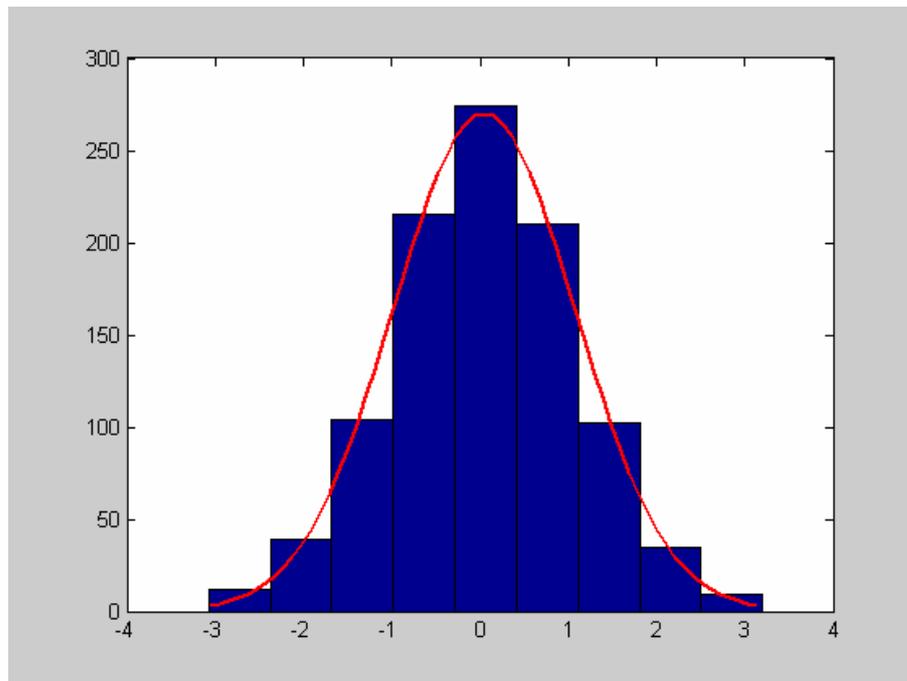
Анализ одномерной случайной величины

Генерация выборки из 1000 псевдослучайных чисел распределенных по нормальному закону с параметрами $Mx=0$ и $\sigma=1$.

```
>> X=normrnd(Mx, sigma, 1000, 1);
```

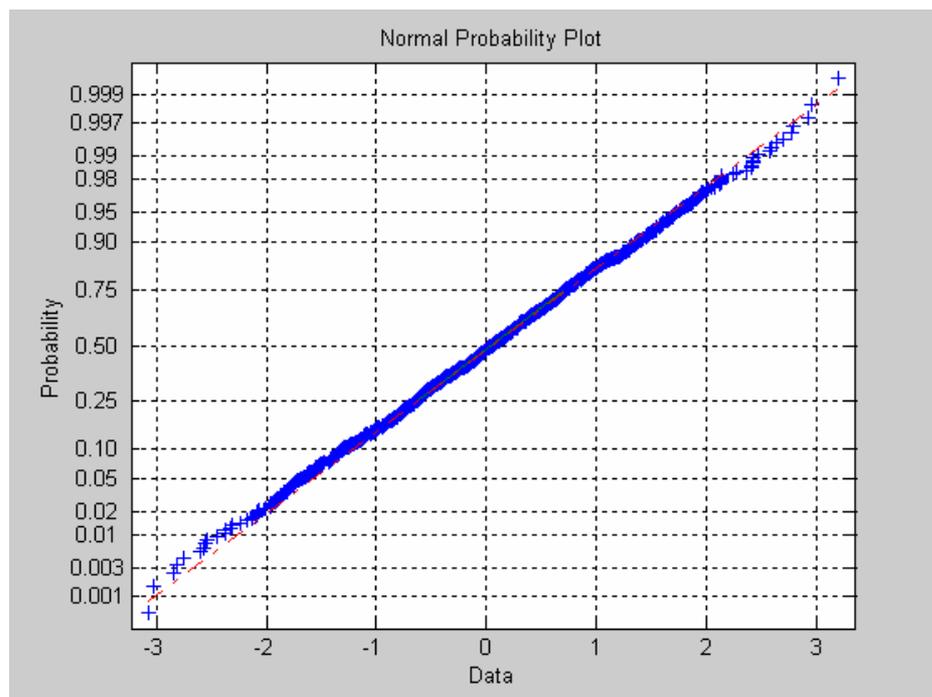
Графическое представление выборки X с помощью гистограммы.

```
>> histfit(X,9)
```



Оценка соответствия выборки X нормальному закону на основе нормального вероятностного графика.

```
>> normplot(X)
```



Как видно из приведенного выше графика практически все точки попадают на прямую линию. Что позволяет сделать вывод о соответствии выборки X нормальному закону.

Расчет точечных (μ , σ) и интервальных (μ_{ci} , σ_{ci}) оценок параметров нормального закона распределения.

```
>> [mu,sigma,muci,sigmaci] = normfit(X)
```

mu =

0.0455

sigma =

1.0313

muci =

-0.0185

0.1095

sigmaci =

0.9880

1.0786

Анализ результатов расчета показывает, что значения параметров нормального закона по которым генерировалась выборка $Mx=0$, $\sigma=1$ близки к расчетным точечным оценкам (μ , σ), и попадают в доверительные интервалы (μ_{ci} , σ_{ci}). Доверительная вероятность при расчете интервальных оценок принята равной 0,95.

Корреляционный и линейный регрессионный анализ

Исследуемая случайная величина является совокупностью 3 векторов распределенных по нормальному, равномерному законам и закону Вейбулла соответственно. Размерность векторов равна 50 элементам.

```
>> X1=normrnd(0,2,50,1);
```

```
>> X2=unifrnd(2,5,50,1);
```

```
>> X3=weibrnd(2,1,50,1);
```

```
>> X=[X1 X2 X3]
```

X =

-1.2055 4.8504 0.1075

-1.9868 2.6934 0.5732

2.3779 3.8205 0.1803

4.7760 3.4579 0.3901

4.5310 4.6739 0.0818

4.6021 4.2863 0.5985

-0.5402 3.3694 0.2377

1.0057 2.0555 0.9831

-0.2384 4.4642 0.9609

-0.0038 3.3341 0.4502

-0.8653 3.8463 0.3431

-0.3896 4.3758 1.1501

1.9707	4.7654	0.8619
0.9372	4.2146	0.5177
-2.7298	2.5288	0.8518
0.5474	3.2171	0.5397
5.2934	4.8064	0.2093
-0.1075	4.7507	0.1711
0.9450	3.2308	0.2087
-4.1601	4.6809	0.3819
-1.6050	2.1737	0.6493
-0.9136	3.0586	0.1850
0.3877	4.4395	0.9116
1.7791	2.0296	0.4197
-3.1833	2.4167	0.2313
-0.6440	2.6083	0.6066
-1.4076	2.5962	0.3955
-1.4886	3.8114	0.2943
0.7426	2.8166	0.5930
2.8746	2.5964	0.4855
0.9198	2.0458	0.7919
1.3215	4.2404	1.5715
2.2487	3.3353	0.3697
1.9588	4.7954	1.0607
-2.6329	3.3980	0.0949
-0.0464	3.2559	1.9497
0.2691	4.5387	0.1583
4.8163	3.5755	0.1454
1.8034	2.6079	1.0427
0.1524	4.0164	0.6684
0.7235	4.5144	0.0734
-4.1173	2.0589	0.0059
-4.6641	4.0438	1.1217
-0.7418	3.1384	0.1110
2.5714	4.4954	0.1774
1.1141	3.5084	0.5415
-0.3605	4.1284	0.1673
-0.0714	3.2867	0.3167
3.8688	2.9139	0.0335
2.6111	2.5690	2.2256

Расчет матрицы коэффициентов корреляции.

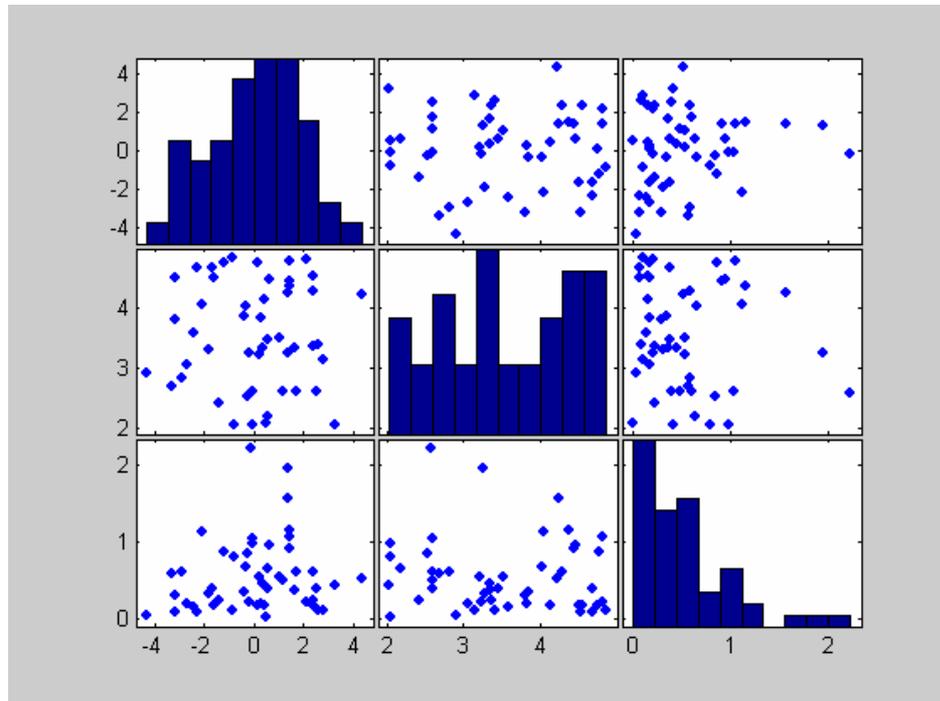
```
>> corrccoef(X)
```

ans =

```
1.0000  0.0062  0.1458
0.0062  1.0000 -0.0976
0.1458 -0.0976  1.0000
```

Поскольку значения коэффициентов корреляции близки нулю, можно сделать вывод об отсутствии взаимосвязи между исследуемыми величинами X1 X2 X3. Этот же вывод следует из анализа матричного графика рассеяния.

```
>> gplotmatrix(X)
```



Определим коэффициенты линейного уравнения регрессии между величинами X1 и X3.

```
>> p = polyfit(X1,X3,1)
```

p =

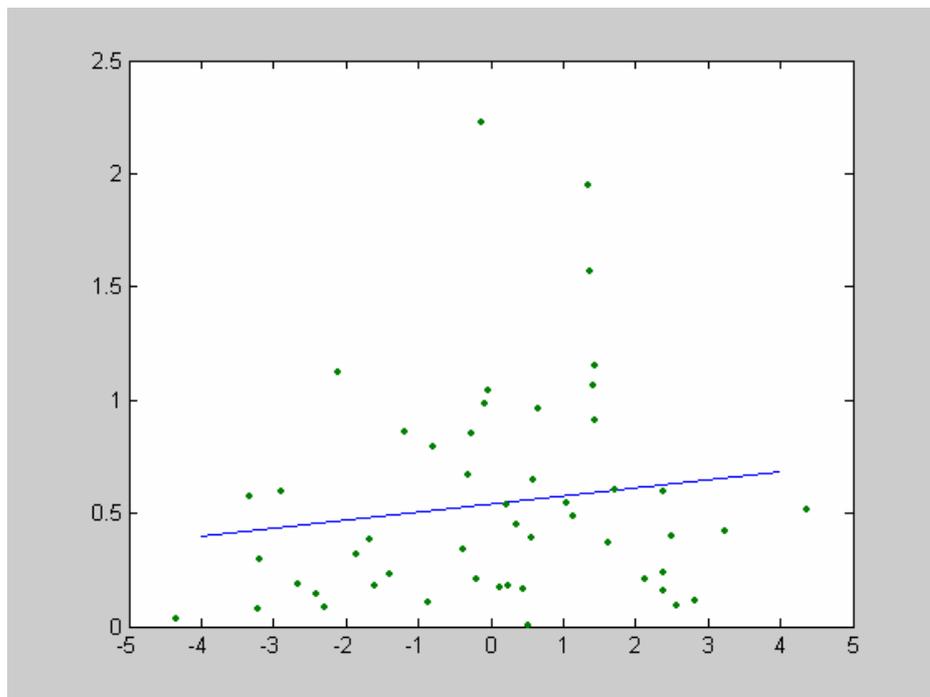
```
0.0356  0.5412
```

График регрессионной модели и экспериментальных точек.

```
>> x=-4:0.01:4;
```

```
>> y = polyval(p,x);
```

```
>> plot(x,y,X1,X3,'')
```



В большинстве статистических систем ввод и редактирование исходных данных осуществляется в электронном табличном редакторе. Поскольку MATLAB не имеет встроенного табличного процессора, то целесообразно использовать существующий редактор. Надстройка Excel Link позволяет объединить вычислительные мощности MATLAB, алгоритмы ППП "Statistics Toolbox" и возможности подготовки исходных данных в табличном виде Microsoft Excel. Excel Link позволяет вызывать функции ППП "Statistics Toolbox" непосредственно из табличного процессора.

Графическая подсистема в MATLAB

boxplot

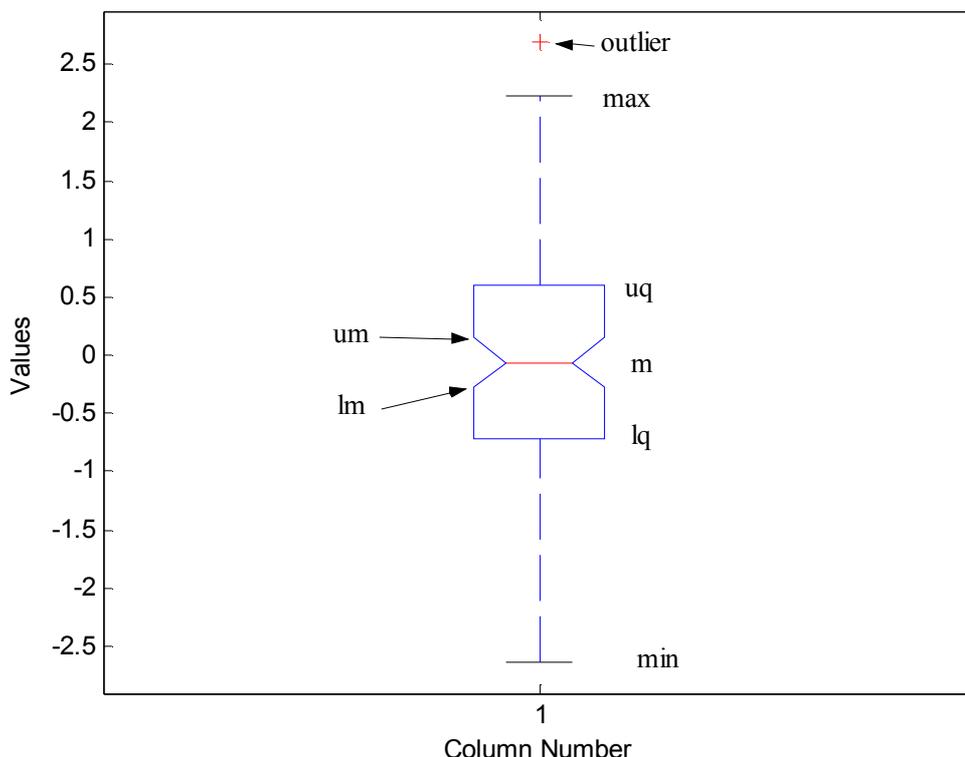
Диаграмма размаха

Синтаксис

```
boxplot(X)
boxplot(X,notch)
boxplot(X,notch,'sym')
boxplot(X,notch,'sym',vert)
boxplot(X,notch,'sym',vert,whis)
```

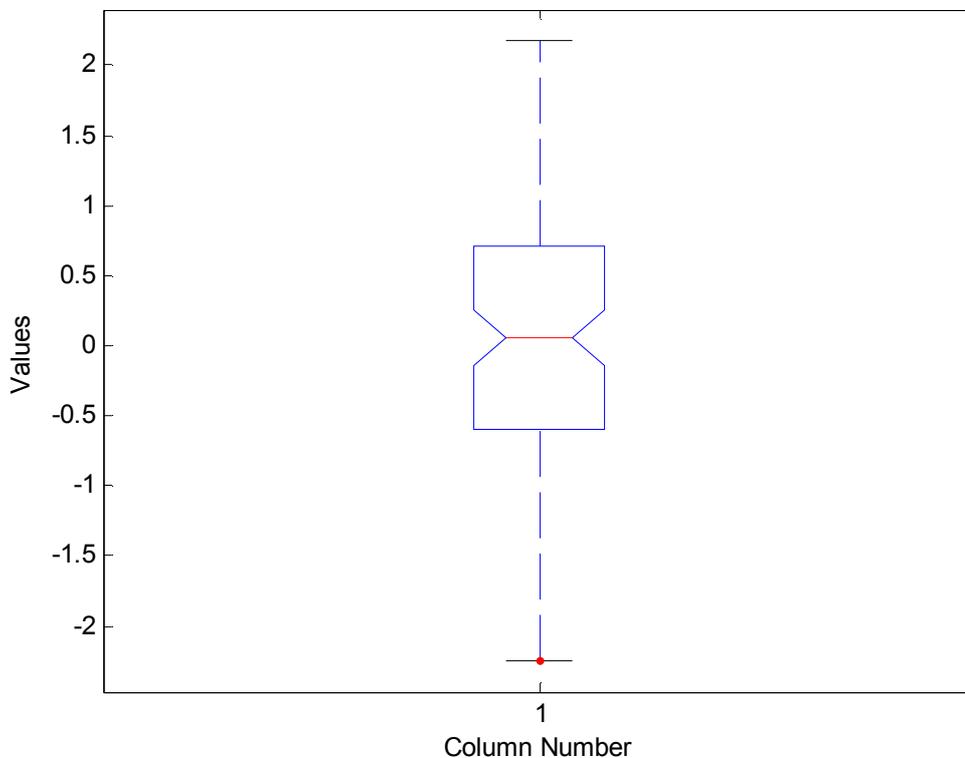
Описание

На диаграмме размаха отображаются медиана m , 25% lq и 75% uq проценти выборки (нижняя и верхняя квартили), верхний max и нижний min пределы выборки за исключением выбросов, выбросы выборки $outlier$, верхняя um и нижняя lm границы доверительного интервала медианы.



Отсутствие выбросов обозначается точкой на нижнем пределе выборки min (см. следующий рис.). Выбросом (грубым промахом) считается значение не попадающее в интервал $[lq-1.5*IQR; uq+1.5*IQR]$, где IQR — величина интерквартильного размаха. При

отсутствии выбросов с обеих сторон от медианы, минимальный `min` и максимальный `max` пределы на диаграмме размаха равны минимальному и максимальному значениям в выборке.



`boxplot(X)` функция позволяет получить диаграмму размаха для каждого столбца матрицы `X` без отметок границ доверительного интервала медианы.

`boxplot(X,notch)` функция позволяет получить диаграмму размаха для каждого столбца матрицы `X` с границами доверительного интервала медианы (`notch=1`) или без таковых (`notch=0`, значение по умолчанию).

`boxplot(X,notch,'sym')` в этом варианте синтаксиса функции входной аргумент `'sym'` позволяет задать вид маркера выброса на графике. Возможные виды маркера приведены в описании функции `LineStyleSpec`. По умолчанию `'sym'='+'`.

`boxplot(X,notch,'sym',vert)` входной аргумент `vert` задает вид ориентации графика: `vert=0` ñ горизонтальное расположение диаграммы размаха, `vert=1` ñ вертикальное расположение (значение по умолчанию).

`boxplot(X,notch,'sym',vert,whis)` входной аргумент `whis` является коэффициентом расположения верхнего `max` и нижнего `min`

пределов выборки. Величины `max` и `min` рассчитываются по формулам:

$$\text{min} = \text{lq} - 1.5 * \text{IQR},$$

$$\text{max} = \text{uq} + 1.5 * \text{IQR}.$$

Примеры использования функции построения диаграммы размаха

Диаграмма размаха для одной выборки

```
>> X = normrnd(0,1,100,1);  
>> boxplot(X)
```

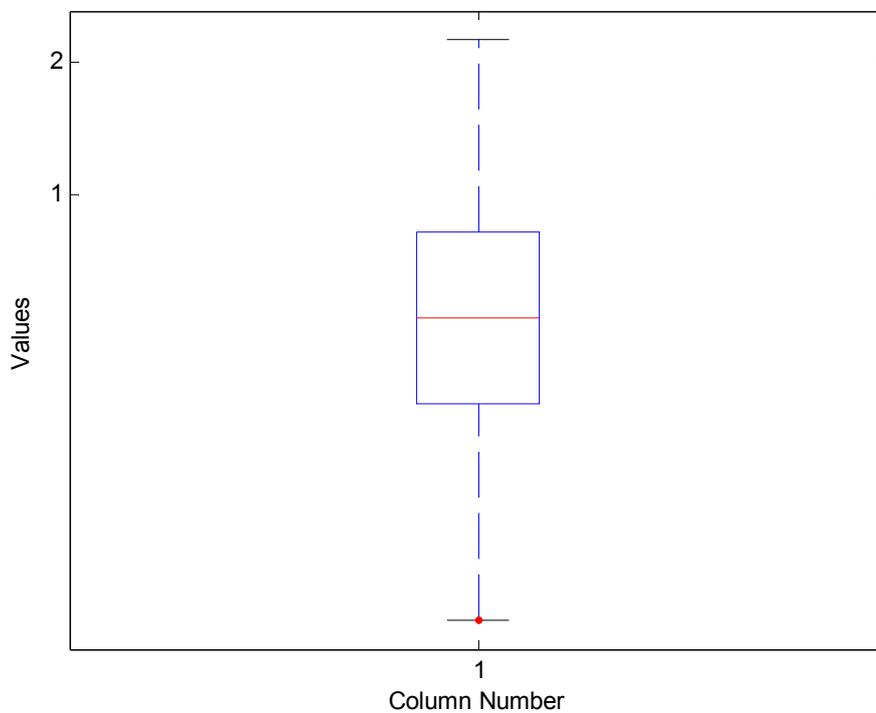


Диаграмма размаха для пяти выборок

```
>> X = normrnd(0,1,100,5);  
>> boxplot(X)
```

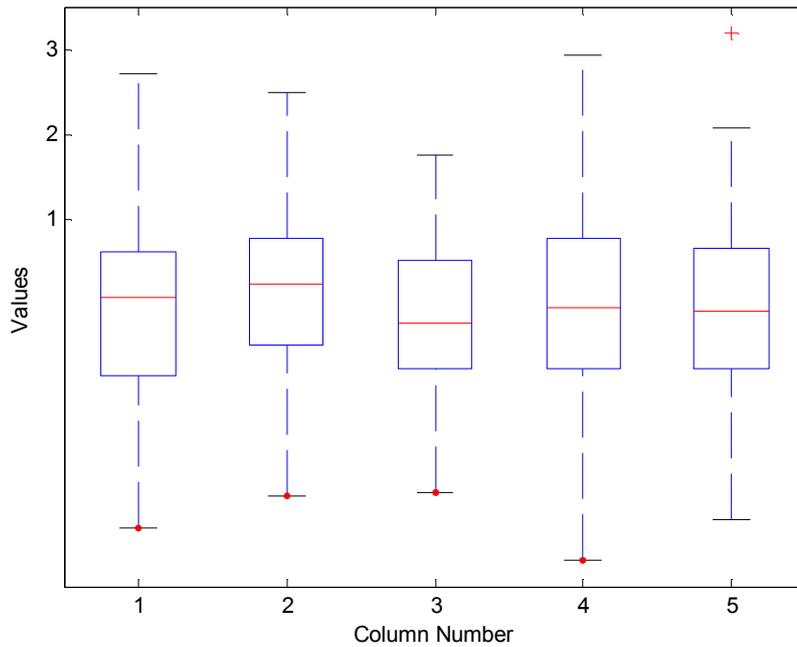


Диаграмма размаха для пяти выборок с границами доверительного интервала медианы

```
>> X = normrnd(0,1,100,5);
>> boxplot(X,1)
```

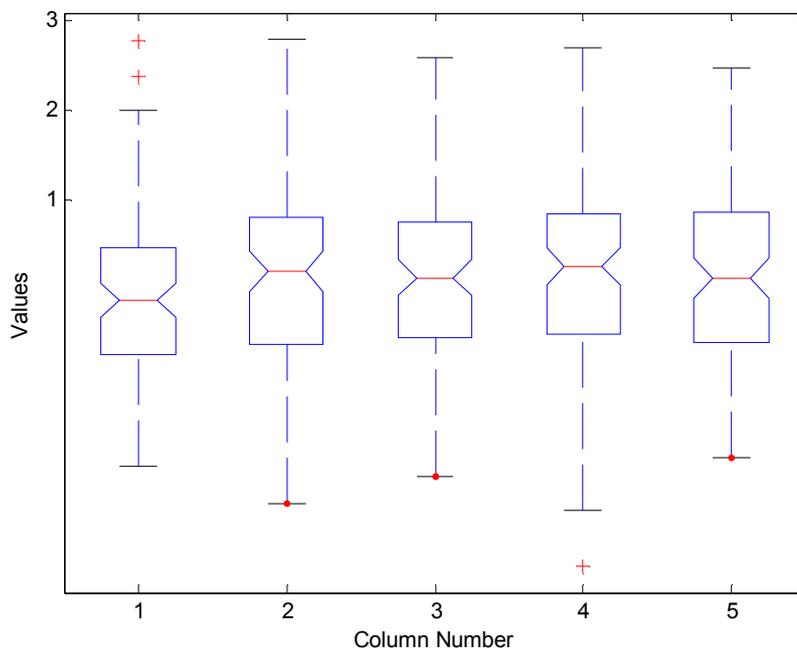


Диаграмма размаха для пяти выборок с границами доверительного интервала медианы с измененным маркером для выбросов.

```
>> X = normrnd(0,1,100,5);
>> boxplot(X,1,'o')
```

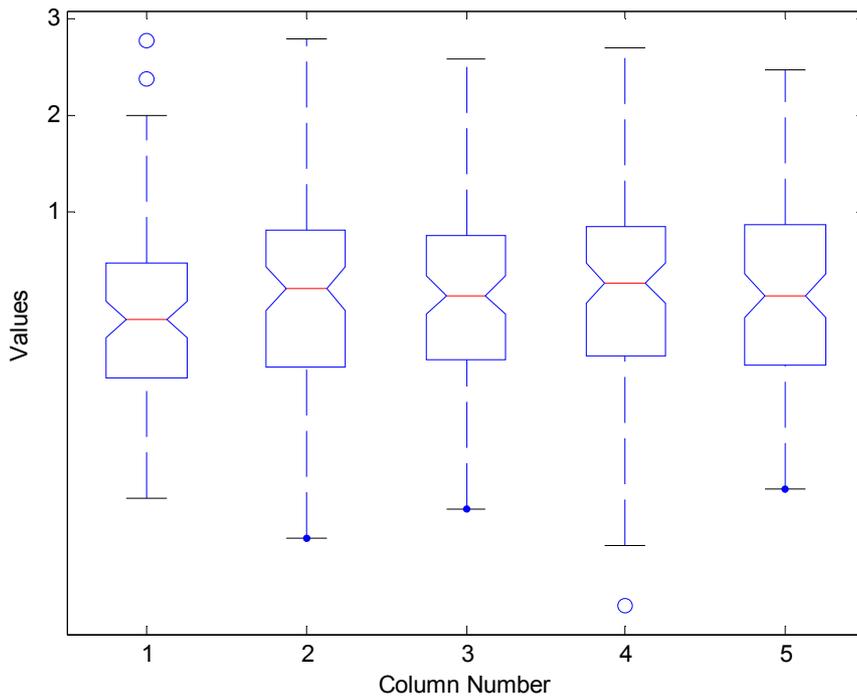


Диаграмма размаха для пяти выборок с границами доверительного интервала медианы с измененным маркером для выбросов и горизонтальной ориентации графика.

```
>> X = normrnd(0,1,100,5);
>> boxplot(X,1,'o',0)
```

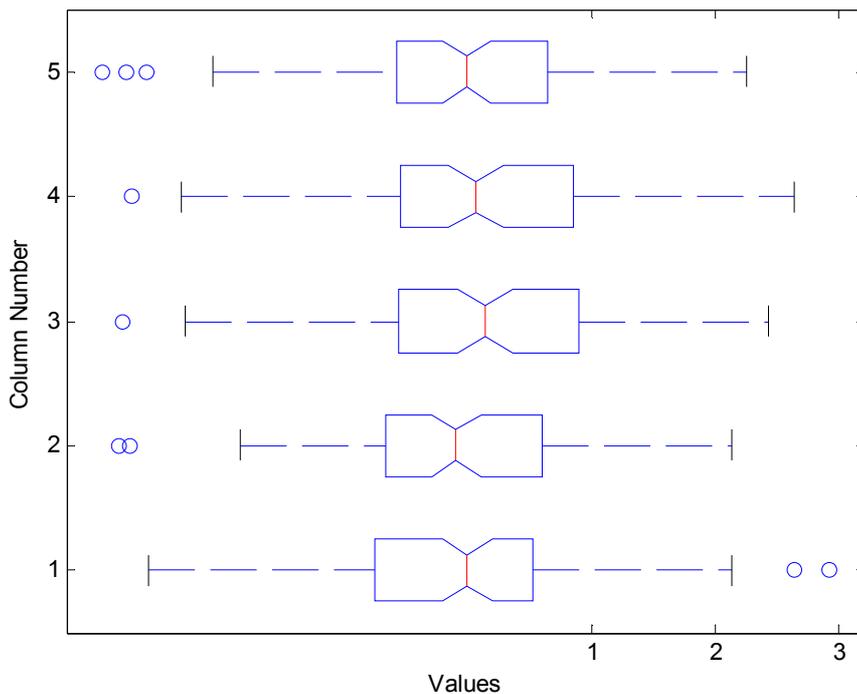
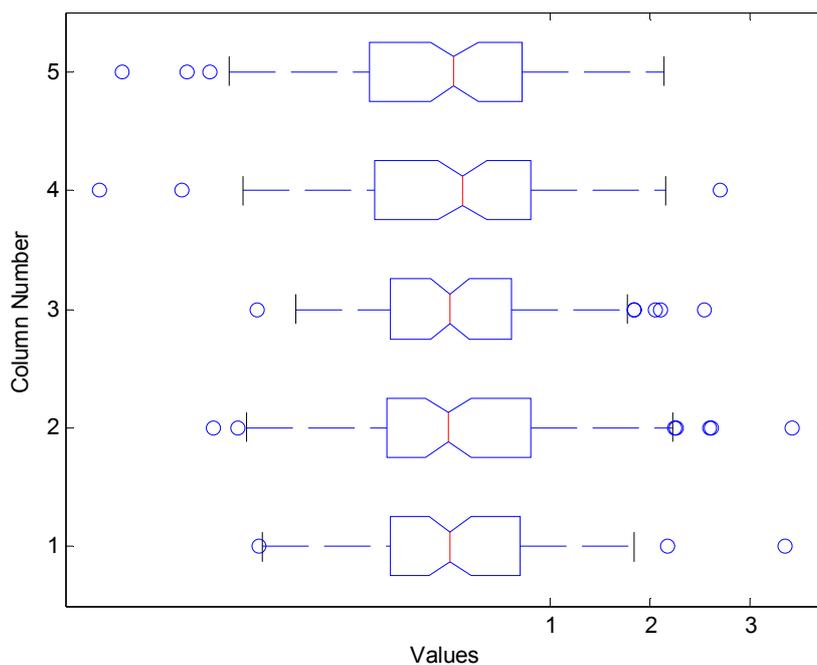


Диаграмма размаха для пяти выборок с границами доверительного интервала медианы с измененным маркером для выбросов и горизонтальной ориентации графика. Границы минимального и максимального пределов смещены на величину интерквартильного размаха относительно квартилей.

```
>> X = normrnd(0,1,100,5);  
>> boxplot(X,1,'o',0,1)
```



cdfplot

График эмпирической функции распределения

Синтаксис

```
cdfplot(X)
h = cdfplot(X)
[h,stats] = cdfplot(X)
```

Описание

`cdfplot(X)` предназначена для построения эмпирической функции распределения (кумулятивной кривой) для выборки X . Выборка X должна быть представлена как вектор.

Функции `kstest`, `kstest2` и `lillietest` предназначенные для проверки непараметрических гипотез являются производными от эмпирической функции распределения. Функция `cdfplot` полезна как графическая иллюстрация соответствия распределения выборки заданному теоретическому распределению при проверке непараметрических гипотез.

`h = cdfplot(X)` функция строит график эмпирической функции распределения и возвращает указатель на кривую графика H .

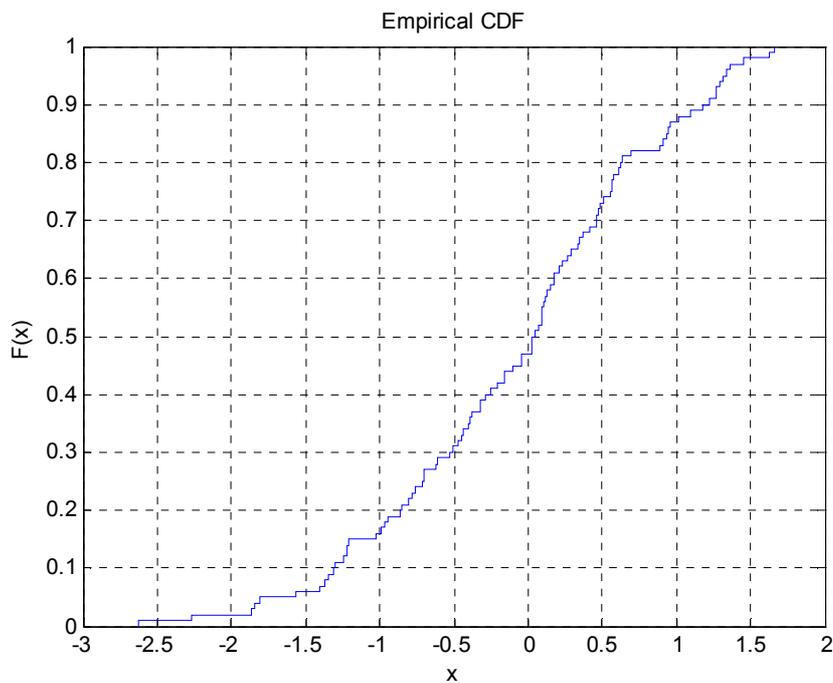
`[h,stats] = cdfplot(X)` функция возвращает структуру `stats`, содержащую следующие поля

Поле stats	Значение
<code>stats.min</code>	Минимальное значение
<code>stats.max</code>	Максимальное значение
<code>stats.mean</code>	Выборочное среднее арифметическое
<code>stats.median</code>	Выборочная медиана (50% процентиль)
<code>stats.std</code>	Выборочное среднее квадратическое отклонение

Примеры использования функции построения эмпирической функции распределения

Эмпирическая функция распределения

```
>> X = normrnd(0,1,100,1);
>> cdfplot(X)
```



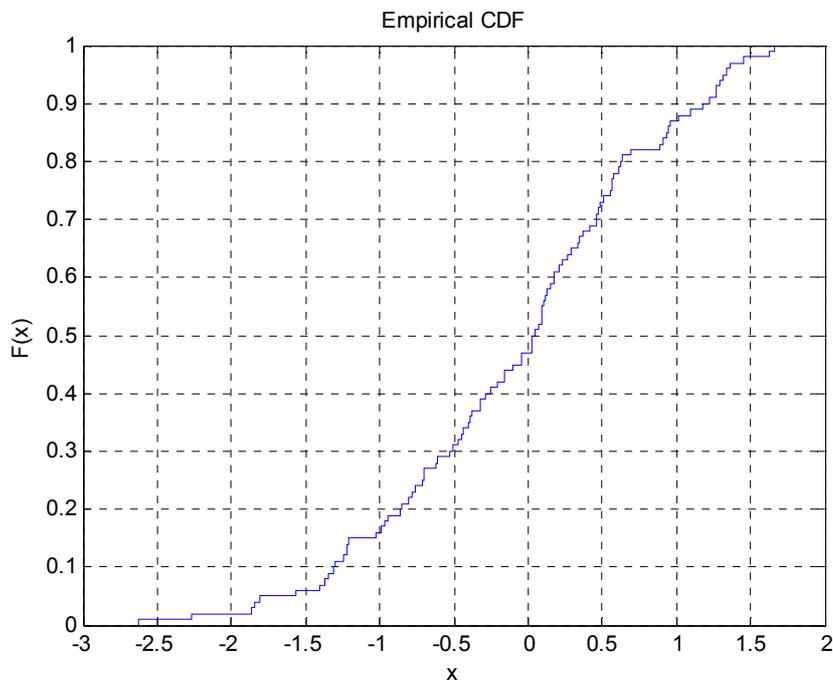
Эмпирическая функция распределения и вывод значения указателя на кумулятивную кривую

```
>> X = normrnd(0,1,100,1);
```

```
>> H = cdfplot(X)
```

```
H =
```

```
3.0013
```



Эмпирическая функция распределения, вывод значения указателя на кумулятивную кривую и характеристик выборки

```
>> X = normrnd(0,1,100,1);
```

```
>> [h,stats] = cdfplot(X)
```

```
h =
```

```
105.0015
```

```
stats =
```

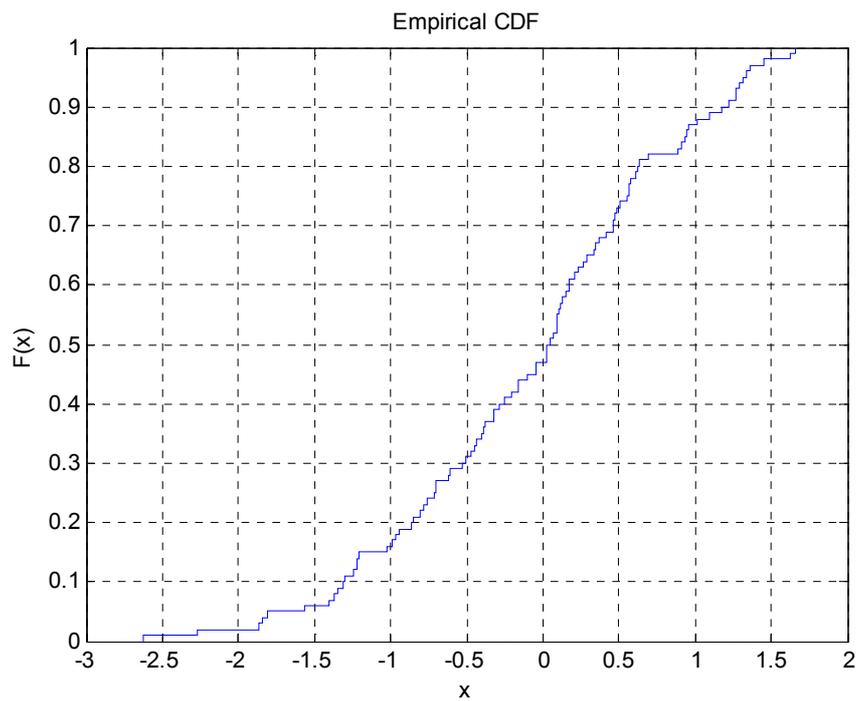
```
min: -2.6272
```

```
max: 1.6559
```

```
mean: -0.0767
```

```
median: 0.0383
```

```
std: 0.9141
```



errorbar

Двумерный график с границами доверительных интервалов

Синтаксис

```
errorbar(X,Y,L,U,symbol)
errorbar(X,Y,L)
errorbar(Y,L)
```

Описание

`errorbar(X,Y,L,U,symbol)` функция предназначена для построения двумерного декартового графика для векторов X и Y с нижней L и верхней U границами доверительного интервала. Входные аргументы X, Y, L, U могут быть заданы как матрицы. В этом случае отдельный график строится для каждого сочетания столбцов X, Y, L, U . Строковый входной аргумент `symbol` задает тип линии графика, вид маркера точек и цвет границ доверительных интервалов.

`errorbar(X,Y,L)` функция предназначена для построения двумерного декартового графика для векторов X и Y с симметричными L границами доверительного интервала относительно Y .

`errorbar(Y,L)` функция предназначена для построения двумерного декартового графика последовательно по значениям вектора Y с симметричными границами доверительного интервала L относительно Y .

Функция `errorbar` является функцией ядра MATLAB.

Примеры использования функции построения двумерного графика с границами доверительных интервалов

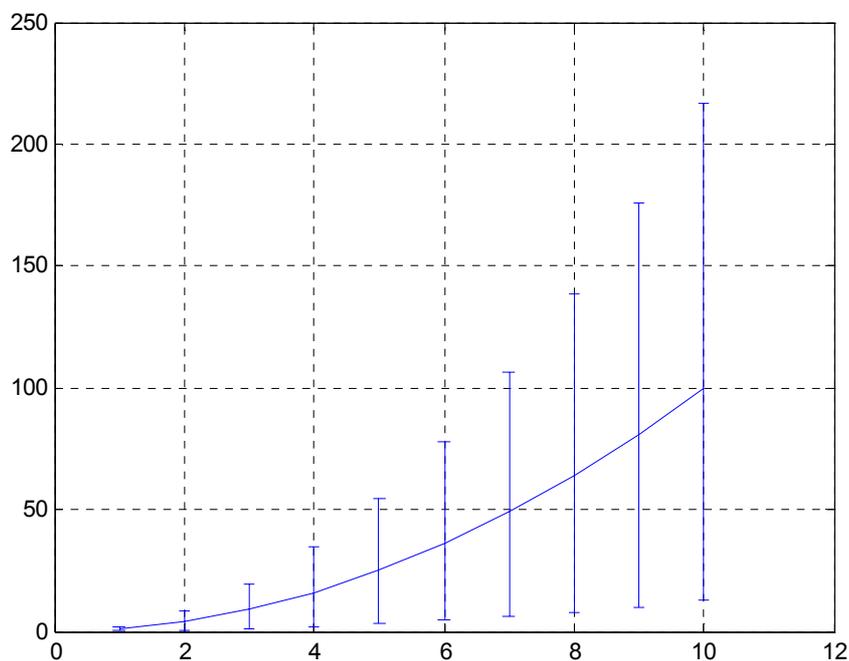
Двумерный декартов график для 2-х векторов с нижней и верхней границами доверительного интервала.

```
>> X=[1 2 3 4 5 6 7 8 9 10]
X =
    1    2    3    4    5    6    7    8    9   10
>> Y=X.^2
Y =
    1    4    9   16   25   36   49   64   81  100
>> L=X.^2- X.^2/8
L =
Columns 1 through 9
```

```

    0.8750    3.5000    7.8750    14.0000    21.8750    31.5000    42.8750    56.0000
70.8750
Column 10
    87.5000
>> U=X.^2+X.^2/6
U =
Columns 1 through 9
    1.1667    4.6667    10.5000    18.6667    29.1667    42.0000    57.1667    74.6667
94.5000
Column 10
    116.6667
>> errorbar(X,Y,L,U)
>> grid on

```



3 графика с границами доверительного интервала для исходных данных заданных как матрицы X,Y,L,U.

```

>> X=[1 2 3 4 5 6 7 8 9 10];
>> X=[X' X'*1.1 X'*1.2];
>> Y=X.^2/1000;
>> L=(X.^2- X.^2/8)/1000;
>> U=(X.^2+X.^2/6)/1000;
>> errorbar(X,Y,L,U)
>> grid on

```

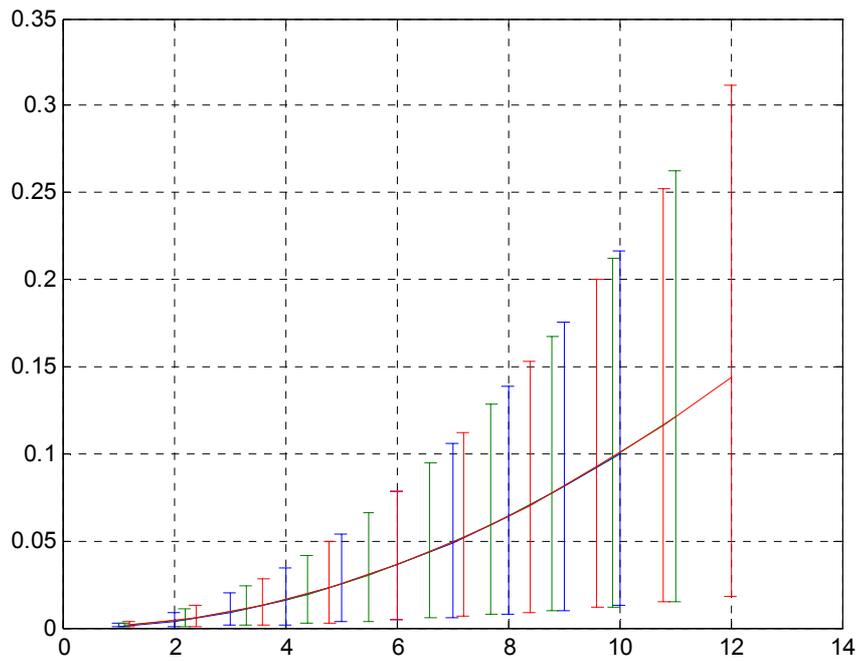
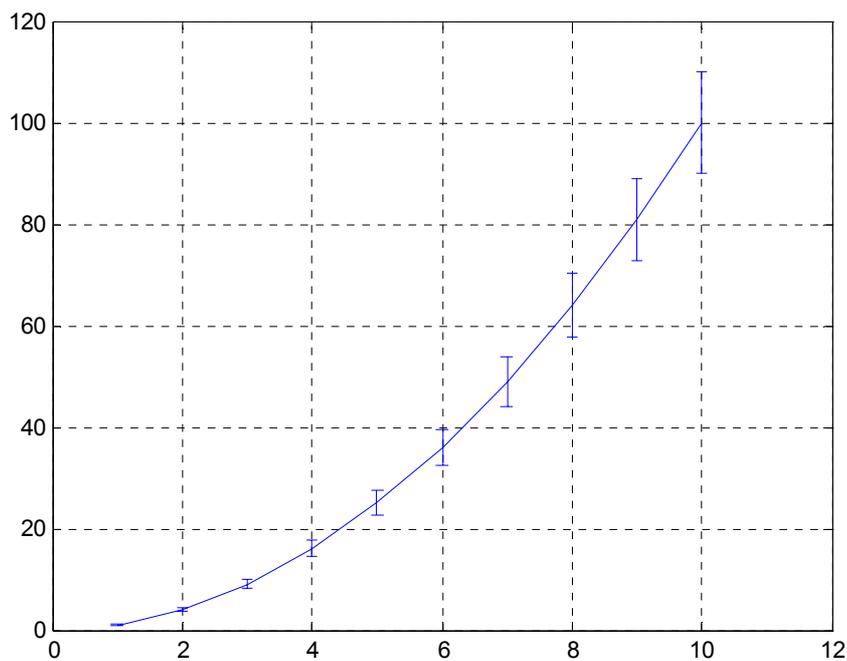


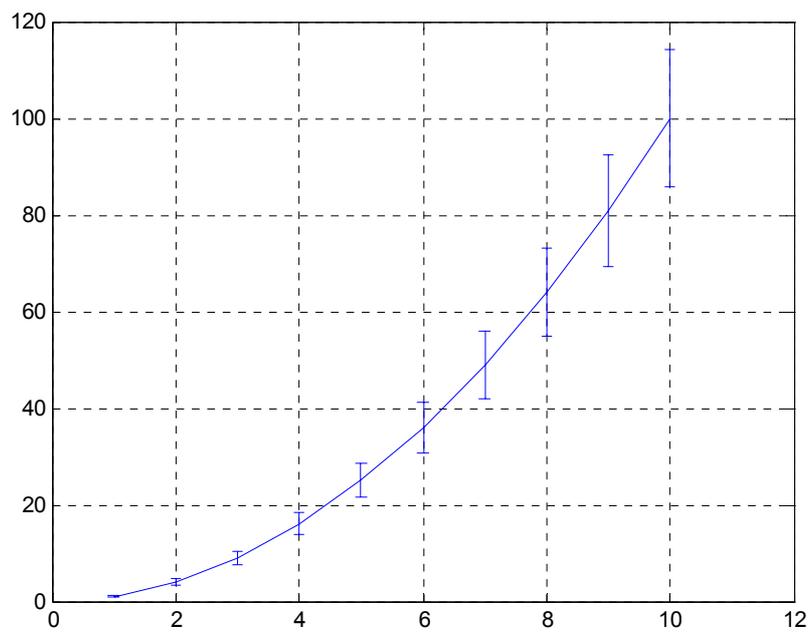
График для векторов с симметричными нижней и верхней границами доверительного интервала.

```
>> X=[1 2 3 4 5 6 7 8 9 10];
>> Y=X.^2;
>> L=X.^2/10;
>> errorbar(X,Y,L)
>> grid on
```



Последовательный график для вектора с симметричными нижней и верхней границами доверительного интервала.

```
>> X=[1 2 3 4 5 6 7 8 9 10];  
>> Y=[1 2 3 4 5 6 7 8 9 10].^2;  
>> L=X.^2/7;  
>> errorbar(Y,L)  
>> grid on
```



fsurfht

Интерактивный контурный график функции

Синтаксис

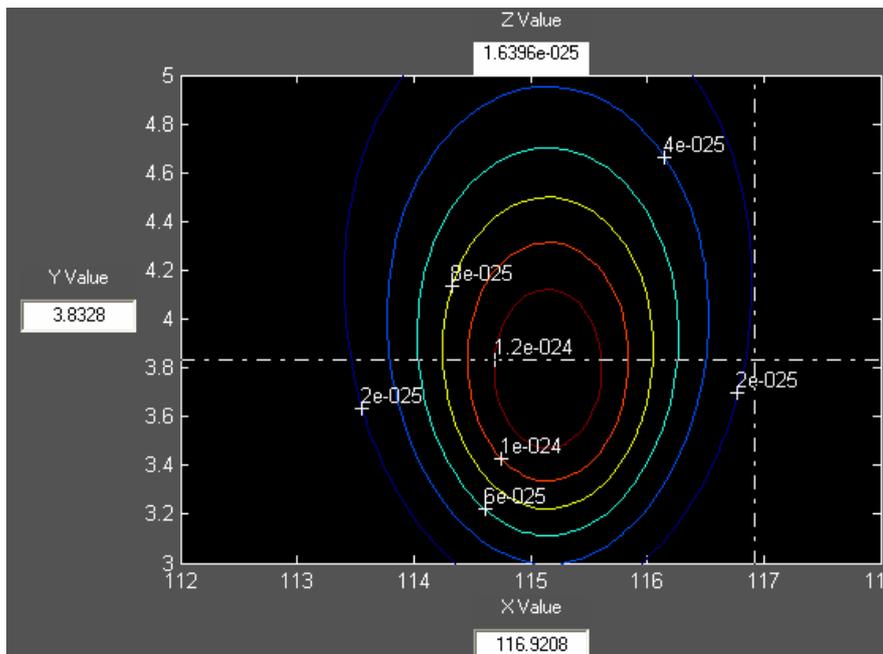
```
fsurfht('fun',xlims,ylims)
fsurfht('fun',xlims,ylims,p1,p2,p3,p4,p5)
```

Описание

`fsurfht('fun',xlims,ylims)` функция предназначена для построения интерактивного контурного графика для функции 'fun', в заданных пределах: по оси абсцисс $\tilde{}$ xlims; по оси ординат - ylims. Вид функции 'fun' заданной как строковая переменная. Пределы по осям задаются в виде векторов [xmin xmax] и [ymin ymax], где xmin, ymin $\tilde{}$ минимальные пределы по осям x и y, xmax, ymax $\tilde{}$ максимальные пределы по осям x и y.

`fsurfht('fun',xlims,ylims,p1,p2,p3,p4,p5)` функция построения интерактивного контурного графика функции получает пять дополнительных входных аргументов p1,p2,p3,p4,p5 и передает их функции 'fun' .

Пересекающиеся вертикальная и горизонтальная штрих-пунктирные линии определяют текущую пару значений по осям X Value и Y Value, для которой рассчитывается значение выходной функции Z Value. Положение точки пересечения линий, определяющее текущую пару значений x и y, можно изменить левой кнопкой мыши, или явно указать значения координат в строках ввода X Value и Y Value. Значение функции будет пересчитано с изменением положения точки пересечения линий или при вводе одной из координат в строку ввода.



Примеры использования функции построения интерактивного контурного графика функции

Построение функции максимального правдоподобия Гаусса для данных из файла `gas.mat`. Загрузка переменных и их значений:

```
>> load gas
```

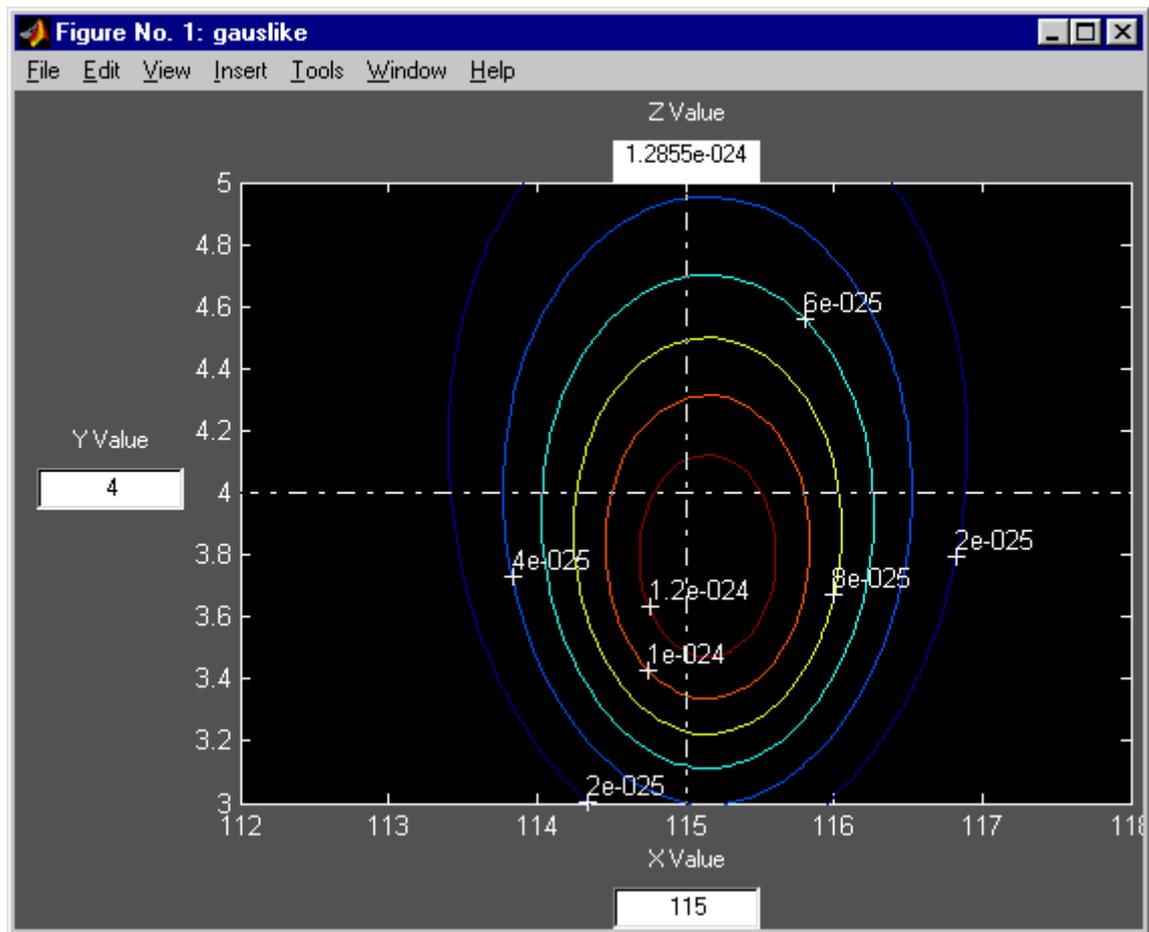
Создание `m`-файла функции `gauslike.m` со следующим текстом программы:

```
function z = gauslike(mu,sigma,p1)
n = length(p1);
z = ones(size(mu));
for i = 1:n
z = z .* (normpdf(p1(i),mu,sigma));
end
```

Функция `gauslike` основана на вызове функции плотности распределения вероятностей нормального закона `normpdf` с математическим ожиданием `mu`, средним квадратическим отклонением `sigma`. В основе расчет лежит предположение о нормальном распределении цен на газ (входной переменной `price1`), и как следствие, график является функцией максимального правдоподобия выборки.

Вызов функции интерактивного контурного графика функции `fsurfht`

```
>> fsurfht('gauslike',[112 118],[3 5],price1)
```



Как следует из приведенного выше рисунка выборочное среднее (значение по оси x) является максимальным. Выборочное среднее арифметическое отклонение не попадает в максимум (значение по оси y).

Максимальные среднее арифметическое и среднее квадратическое отклонение определяются как

```
>> mumax = mean(price1)
mumax =
    115.1500
>> sigmax = std(price1)*sqrt(19/20)
sigmax =
    3.7719
```

gline

Интерактивное построение линии в графическом окне

Синтаксис

```
gline(fig)
h = gline(fig)
gline
```

Описание

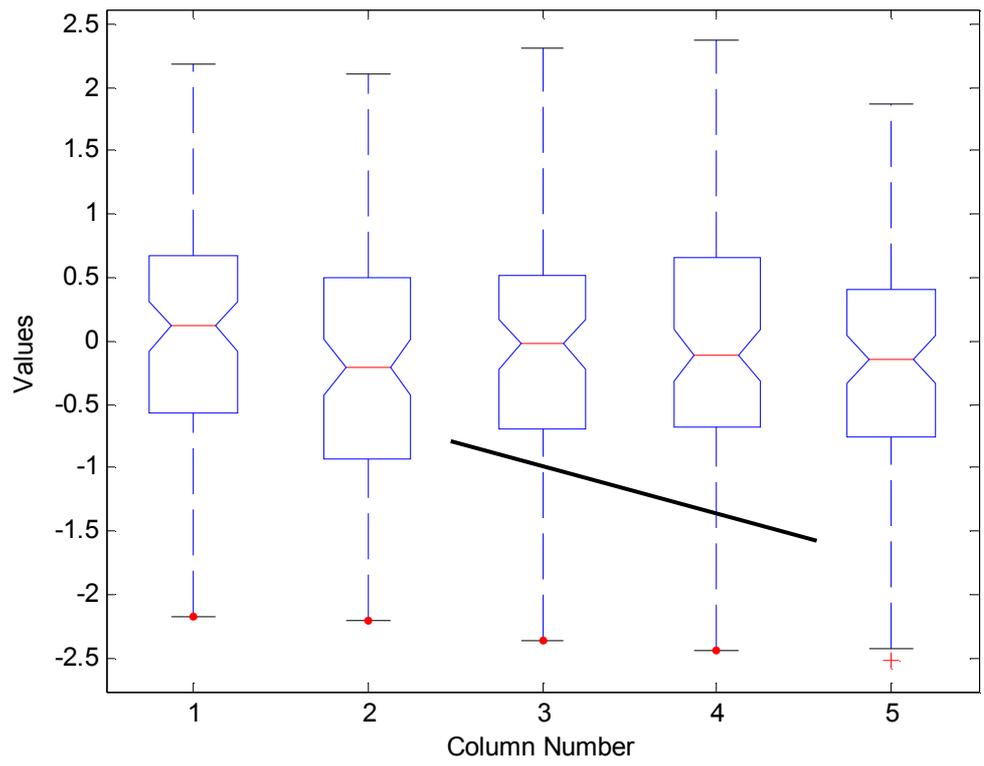
`gline(fig)` функция предназначена для интерактивного построения сегмента линии в графическом окне `fig`. Сегмент линии определяется между двумя нажатиями левой кнопки мыши.

`h = gline(fig)` функция позволяет построить сегмент линии в графическом окне `fig` и возвращает указатель на сегмент `h`.

`gline` вызов функции без аргументов позволяет построить сегмент линии в текущем графическом окне.

Примеры использования функции интерактивного построения сегмента линии

```
>> X = normrnd(0,1,100,5);
>> boxplot(X,1)
>> h = gline(1)
h =
    141.0002
```



gname

Интерактивная вставка меток точек на графике

Синтаксис

```
gname('cases')
gname
h = gname('cases',line_handle)
```

Описание

`gname('cases')` функция предназначена для вставки меток точек на графике в интерактивном режиме. Метки определяются входным вектором строк 'cases'. Количество элементов в векторе 'cases' должно быть равно числу точек на графике. Порядок точек на графике соответствует последовательности элементов вектора 'cases'. Вставка выполняется после выделения мышью одной или нескольких точек на графике. Одна точка выделяется однократным щелчком левой кнопки мыши. Чтобы вставить метки для группы точек необходимо при нажатой левой кнопке мыши создать прямоугольную зону, охватывающую выделяемые значения. Для завершения вставки меток используются клавиши Enter или Escape.

`gname` функция без входных аргументов в качестве меток использует номера выделенных точек или зон выделения.

`h = gname('cases',line_handle)` функция возвращает вектор указателей `h` на текстовые объекты графика. Скаляр `line_handle` предназначен для выделения кривой графика, если в одних осях откладываются несколько зависимостей.

`gname` используется совместно с функциями построения графиков: `plot`, `scatter`, `gscatter`, `plotmatrix`, `gplotmatrix`.

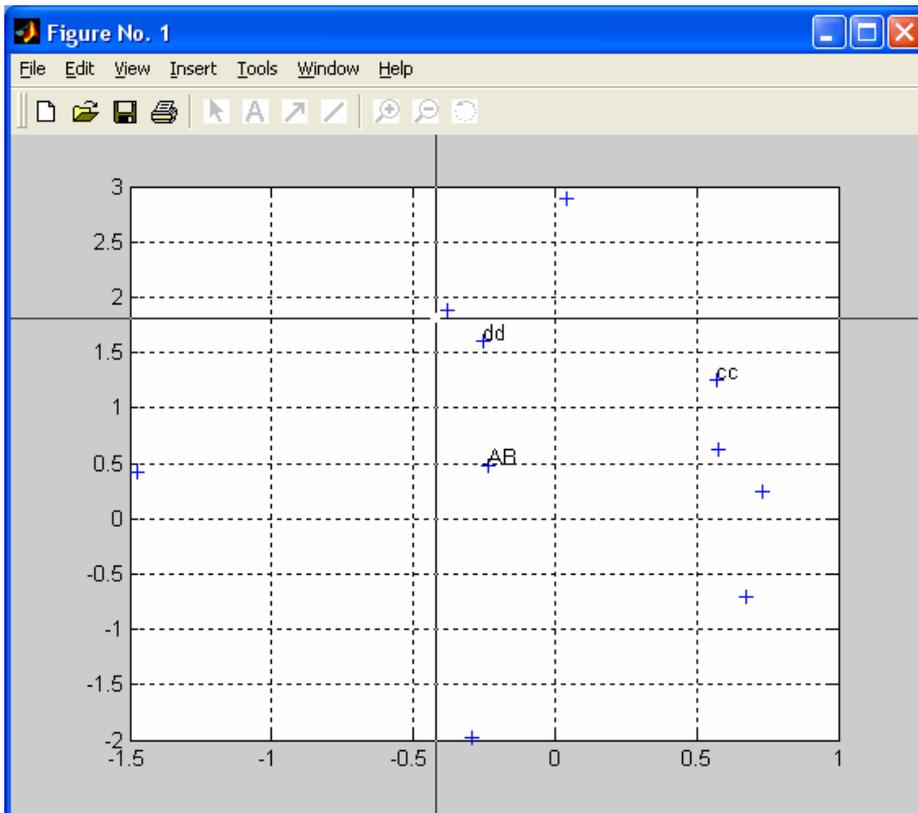
Примеры использования функции интерактивной вставки меток точек на графике

Интерактивная вставка меток точек графика. Метки задаются вектором строк.

```
>> x=normrnd(0,1,10,1);
>> y=normrnd(0,2,10,1);
>> plot(x,y,'+')
>> grid on
>> cases=['AA';'BB';'CC'; 'bb';'cc';'dd';'Ab';'Bc';'Cd';'AB']
cases =
```

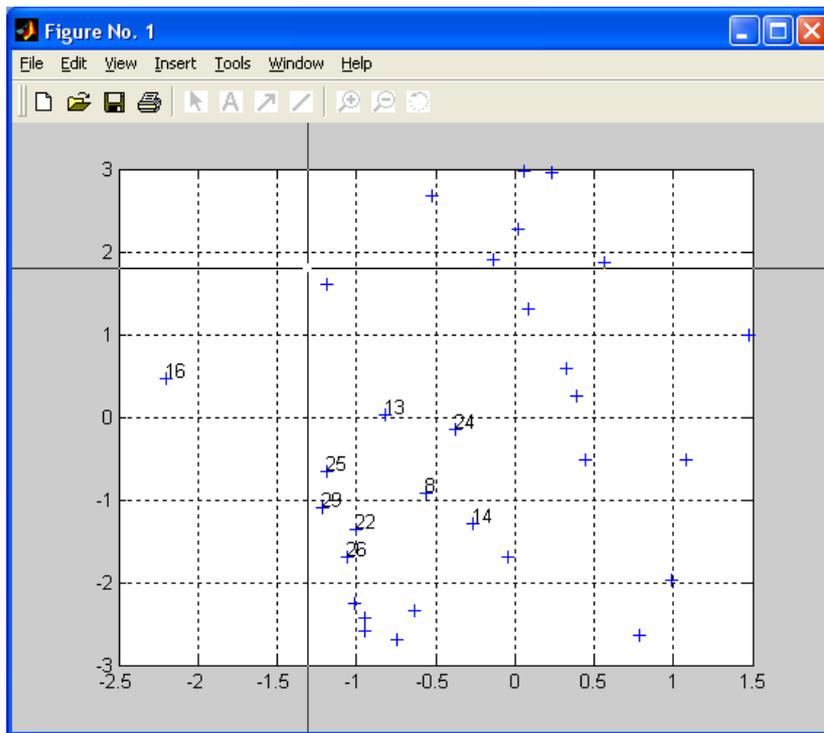
AA
BB
CC
bb
cc
dd
Ab
Bc
Cd
AB

>> gname(cases)



Интерактивная вставка меток точек графика. Метки задаются по умолчанию, как номера точек при построении графика.

```
>> x=normrnd(0,1,30,1);  
>> y=normrnd(0,2,30,1);  
>> plot(x,y,'+')  
>> grid on  
>> gname
```



Интерактивная вставка меток точек первого графика. Метки задаются вектором строк.

```

>> cases=['AA';'BB';'CC'; 'bb';'cc';'dd';'Ab';'Bc';'Cd';'AB'];
>> x=1:1:10;
>> y1=x+normrnd(0,1,1,1);
>> y2=x+normrnd(1,2,1,1);
>> plot(x,y1,'+', x,y2,'o')
>> grid on
>> line_handle = findobj(gca,'Type','line')
line_handle =
    3.0076
   103.0038
>> get(line_handle(1))
    Color = [0 0.5 0]
    EraseMode = normal
    LineStyle = none
    LineWidth = [0.5]
    Marker = o
    MarkerSize = [6]
    MarkerEdgeColor = auto
    MarkerFaceColor = none
    XData = [ (1 by 10) double array]
    YData = [ (1 by 10) double array]
    ZData = []
    BeingDeleted = off
    ButtonDownFcn =

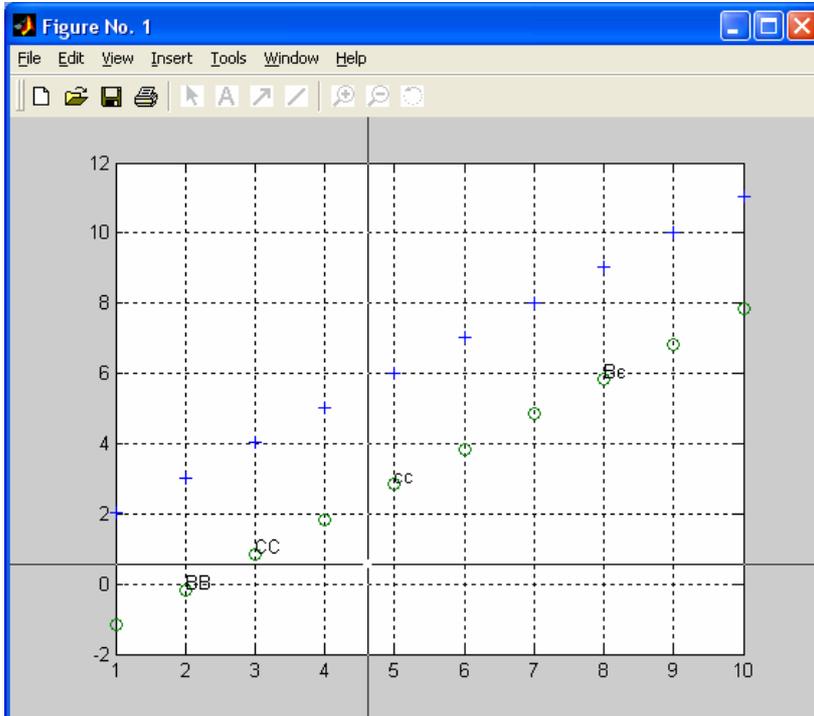
```

```
Children = []
Clipping = on
CreateFcn =
DeleteFcn =
BusyAction = queue
HandleVisibility = on
HitTest = on
Interruptible = on
Parent = [101.002]
Selected = off
SelectionHighlight = on
Tag =
Type = line
UIContextMenu = []
UserData = []
Visible = on
```

```
>> h = gname(cases,line_handle(1))
```

```
h =
```

```
102.0109
104.0043
105.0045
106.0022
```



gplotmatrix

Матрица графиков рассеяния

Синтаксис

```
gplotmatrix(x,y,g)
gplotmatrix(x,y,g,'clr','sym',siz)
gplotmatrix(x,y,g,'clr','sym',siz,'doleg')
gplotmatrix(x,y,g,'clr','sym',siz,'doleg','dispopt')
gplotmatrix(x,y,g,'clr','sym',siz,'doleg','dispopt','
xnam','ynam')
[h,ax,bigax] = gplotmatrix(...)
```

Описание

`gplotmatrix(x,y,g)` функция предназначена для создания матрицы двумерных декартовых графиков рассеяния, исходными данными для каждого графика являются соответствующие столбцы матриц x, y . Точки в матрице графиков сгруппированы по значениям входного аргумента g . Матрицы x, y должны иметь одинаковое значение строк. Если x содержит p столбцов, а y q столбцов, то матрица графиков рассеяния будет иметь размерность $p \times q$. Если матрица y не задана, или y определена как пустая матрица $y=[]$, то будет построена квадратная матрица графиков по парам столбцов x .

Группирующая переменная g может быть задана как вектор, массив строк или массив ячеек строк. Количество элементов вектора g должно быть равно числу строк матриц x, y . Точки на графиках принадлежат к одной группе, если равны соответствующие значения элементов g . Группа обозначается в матрице графиков рассеяния одинаковым цветом и типом маркера. В общем случае, g может быть представлена массивом ячеек переменных любого типа. В этом случае точки графиков принадлежат к одной группе, если равны все переменные в ячейках массива.

`gplotmatrix(x,y,g,'clr','sym',siz)` входные аргументы `'clr'`, `'sym'`, `siz` предназначены для задания цвета, типа и размера маркера точек групп соответственно. `'clr'` задается как массив строк согласно спецификации цветов функции `plot`. По умолчанию `'clr'='bgrcmyk'`. `'sym'` массив символов в соответствии с правилами задания типа маркера функции `plot`. Значение по умолчанию `'sym'='.'`. `siz` - вектор размеров маркеров, определяется по умолчанию значением свойства `'defaultlinemarkersize'`. Если число элементов векторов `'clr'`, `'sym'`, `siz` меньше количества групп, то `gplotmatrix` будет циклически повторять заданные значения.

`plotmatrix(x,y,g,'clr','sym',siz,'doleg')` входной параметр `'doleg'` управляет выводом легенды графика. `'doleg'='on'`, значение по умолчанию, соответствует отображению легенды графика, `'doleg'='off'` и легенда не выводится в графическое окно.

`plotmatrix(x,y,g,'clr','sym',siz,'doleg','dispopt')` входной аргумент `'dispopt'` задает тип диагонального элемента в матрице графиков рассеяния, построенной по столбцам x . Возможны следующие диагональные элементы:

Значение <code>'dispopt'</code>	Тип диагонального элемента
'none'	Пустой диагональный элемент
'hist'	Гистограмма выборки $X_{1:n,j}$, где $j=1..m$; n и число строк; m и число столбцов. Значение по умолчанию.
'variable'	Вывод названий переменных

`plotmatrix(x,y,g,'clr','sym',siz,'doleg','dispopt','xnam','ynam')` аргументы `'xnam'`, `'ynam'` предназначена для задания названий столбцов матриц x , y и соответствующих осей графиков. `'xnam'`, `'ynam'` должны быть заданы как векторы строковых переменных. Количество строк в векторах `'xnam'`, `'ynam'` должно быть равно числу столбцов матриц x , y соответственно.

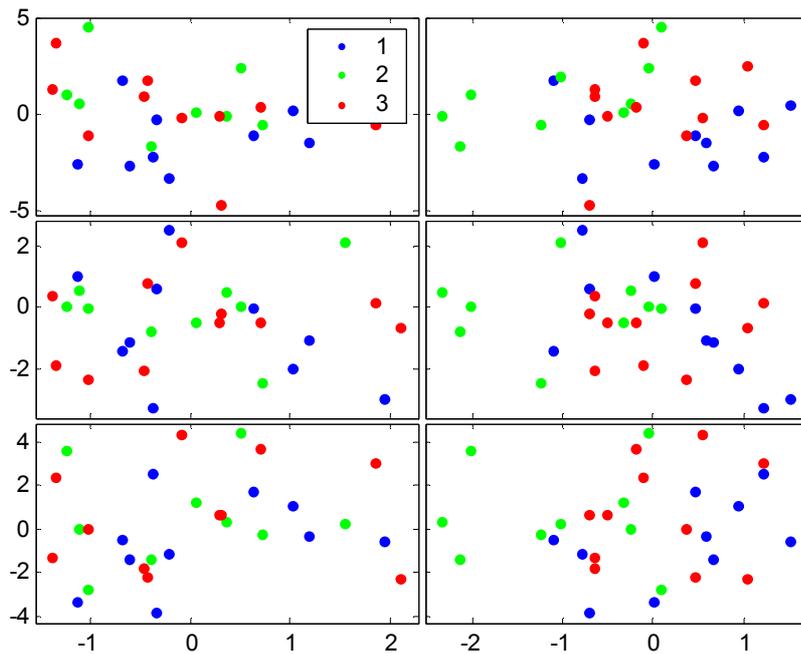
`[h,ax,bigax] = plotmatrix(...)` функция возвращает следующие выходные аргументы: h и массив указателей на линии графика; ax и матрица указателей на оси каждого графика; $bigax$ и указатель на оси системы координат графического окна, относительно которых определяется положение отдельных графиков в матрице. Такие команды как `title`, `xlabel`, `ylabel` добавляют соответствующие метки в графическое окно в системе координат графического окна с указателем равным $bigax$.

Примеры использования функции создания матрицы графиков рассеяния

Матрица графиков рассеяния с размерностью 2×3 разделена на 3 группы

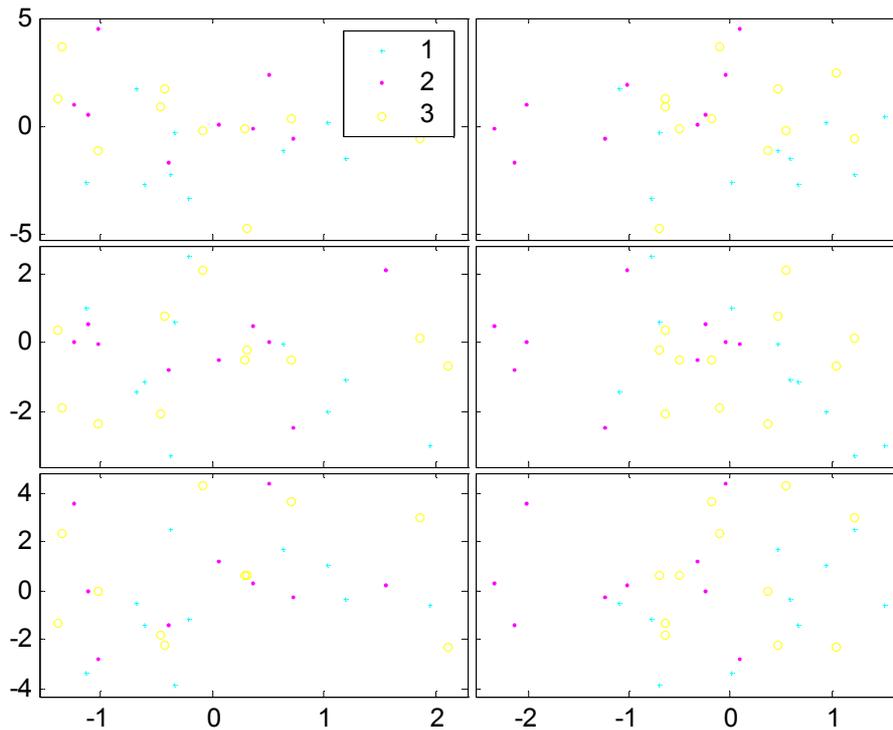
```
>> x=normrnd(0,1,30,2);
>> y=normrnd(0,2,30,3);
>> g=unidrnd(3,30,1);
```

```
>> gplotmatrix(x,y,g)
```



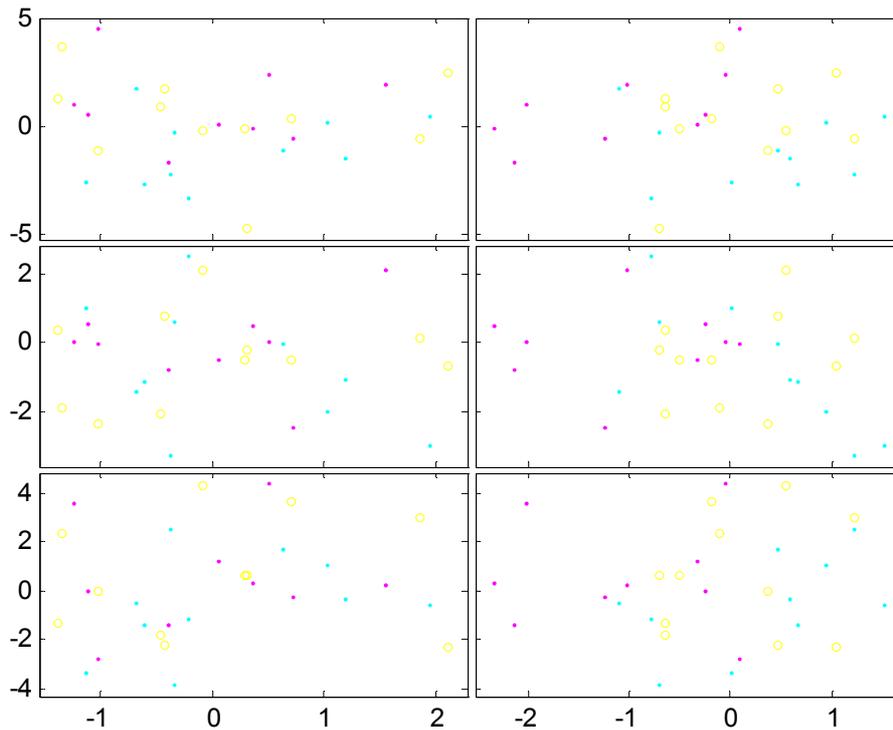
Матрица графиков рассеяния с размерностью 2×3 разделена на 3 группы. Параметры маркеров заданы как 'clr'='cmy', 'sym'='+.o', siz=[1 2 3].

```
>> x=normrnd(0,1,30,2);  
>> y=normrnd(0,2,30,3);  
>> g=unidrnd(3,30,1);  
>> gplotmatrix(x,y,g,'cmy','+.o',[1 2 3])
```



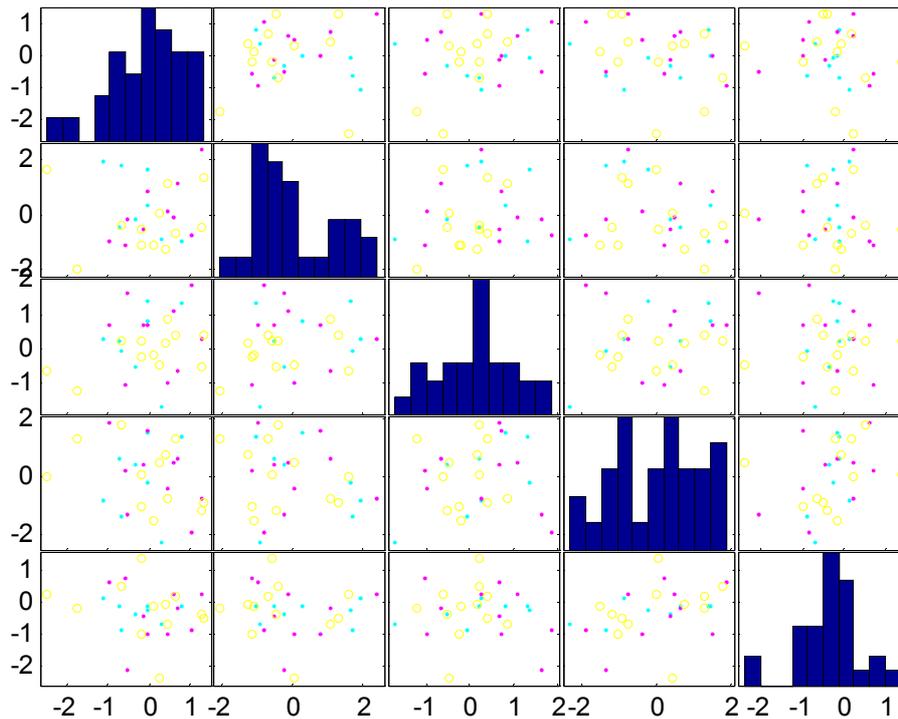
Матрица графиков рассеяния с размерностью 2×3 разделена на 3 группы. Параметры маркеров заданы как 'clr'='cmy', 'sym'='+.o', siz=[1 2 3]. Легенда на график не выводится.

```
>> x=normrnd(0,1,30,2);
>> y=normrnd(0,2,30,3);
>> g=unidrnd(3,30,1);
>> gplotmatrix(x,y,g,'cmy','+.o',[1 2 3], 'off')
```



Матрица графиков рассеяния с размерностью 5×5 разделена на 3 группы. График и рассеяния строятся по столбцам матрицы x . Матрица y задана пустой. Параметры маркеров заданы как `'clr'='cmy'`, `'sym'='+.o'`, `size=[1 2 3]`. Легенда на график не выводится. В качестве диагональных элементов используются гистограммы столбцов матрицы x .

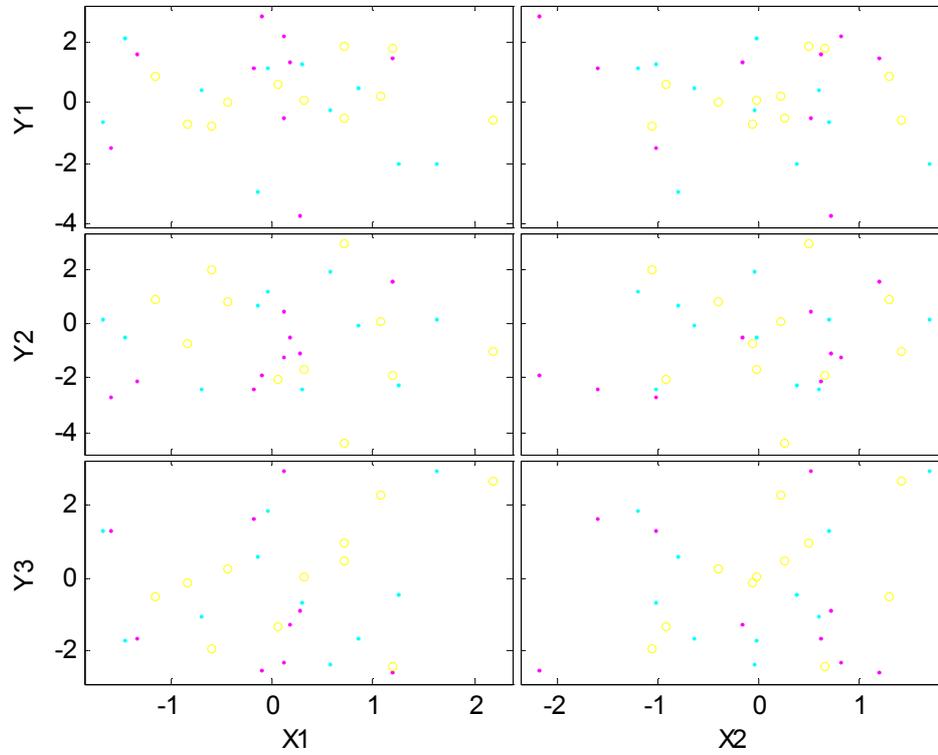
```
>> x=normrnd(0,1,30,5);
>> y=[];
>> g=unidrnd(3,30,1);
>> gplotmatrix(x,y,g,'cmy','+.o',[1 2 3], 'off', 'hist')
```



Матрица графиков рассеяния с размерностью 2×3 разделена на 3 группы. Параметры маркеров заданы как 'clr='cmy', 'sym'='+.o', siz=[1 2 3]. Легенда на график не выводится. Заданы подписи по осям графиков. Выводятся значения указателей на объекты графика.

```
>> x=normrnd(0,1,30,2);
>> y=normrnd(0,2,30,3);
>> g=unidrnd(3,30,1);
>> xnam=['X1';'X2'];
>> ynam=['Y1';'Y2';'Y3'];
>> [h,ax,bigax]=gplotmatrix(x,y,g,'cmy','+.o',[1 2 3], 'off', 'hist', xnam, ynam)
h(:,,1) =
    142.0004    134.0004
    126.0004    118.0004
    110.0004    102.0006
h(:,,2) =
    143.0004    135.0004
    127.0004    119.0004
    111.0004    103.0006
h(:,,3) =
    144.0004    136.0004
    128.0004    120.0004
    112.0004    104.0006
ax =
    141.0004    133.0004
```

125.0004 117.0004
109.0004 101.0016
bigax =
100.0010



gscatter

График рассеяния

Синтаксис

```
gscatter(x,y,g)
gscatter(x,y,g,'clr','sym',siz)
gscatter(x,y,g,'clr','sym',siz,'doleg')
gscatter(x,y,g,'clr','sym',siz,'doleg','xnam','ynam')
h = gscatter(...)
```

Описание

`gscatter(x,y,g)` функция предназначена для создания двумерного графика рассеяния. Исходными данными являются векторы `x`, `y`. Точки на графике рассеяния сгруппированы по значениям входного аргумента `g`. Векторы `x`, `y`, `g` должны иметь одинаковое количество элементов. Группирующая переменная `g` может быть задана как вектор, массив строк или массив ячеек строк. Точки на графиках принадлежат к одной группе, если равны соответствующие значения элементов `g`. Группа обозначается на графике рассеяния одинаковым цветом и типом маркера. В общем случае, `g` может быть представлена массивом ячеек переменных любого типа. В этом случае точки графиков принадлежат к одной группе, если равны все переменные в ячейках массива.

`gscatter(x,y,g,'clr','sym',siz)` входные аргументы `'clr'`, `'sym'`, `siz` предназначены для задания цвета, типа и размера маркера точек групп соответственно. `'clr'` задается как массив строк согласно спецификации цветов функции `plot`. По умолчанию `'clr'='bgrcmtyk'`. `'sym'` массив символов в соответствии с правилами задания типа маркера функции `plot`. Значение по умолчанию `'sym'='.'`. `siz` - вектор размеров маркеров, определяется по умолчанию значением свойства `'defaultlinemarkersize'`. Если число элементов векторов `'clr'`, `'sym'`, `siz` меньше количества групп, то `gscatter` будет циклически повторять заданные значения.

`gscatter(x,y,g,'clr','sym',siz,'doleg')` входной параметр `'doleg'` управляет выводом легенды графика. `'doleg'='on'`, значение по умолчанию, соответствует отображению легенды графика, `'doleg'='off'` ñ легенда не выводится в графическое окно.

`gscatter(x,y,g,'clr','sym',siz,'doleg','xnam','ynam')` аргументы `'xnam'`, `'ynam'` предназначена для задания названий осей графика. `'xnam'`, `'ynam'` должны быть заданы как строковые

переменные. Если 'хnam', 'уnam' не заданы, то в качестве названий осей используются идентификаторы переменных x, y.

`h = gscatter(...)` функция возвращает массив указателей h на объекты графика.

Примеры использования функции создания графика рассеяния

График рассеяния, точки которого разделены на 3 группы

```
>> x=normrnd(0,1,30,1);  
>> y=normrnd(0,2,30,1);  
>> g=unidrnd(3,30,1);  
>> gscatter(x,y,g)  
>> grid on
```

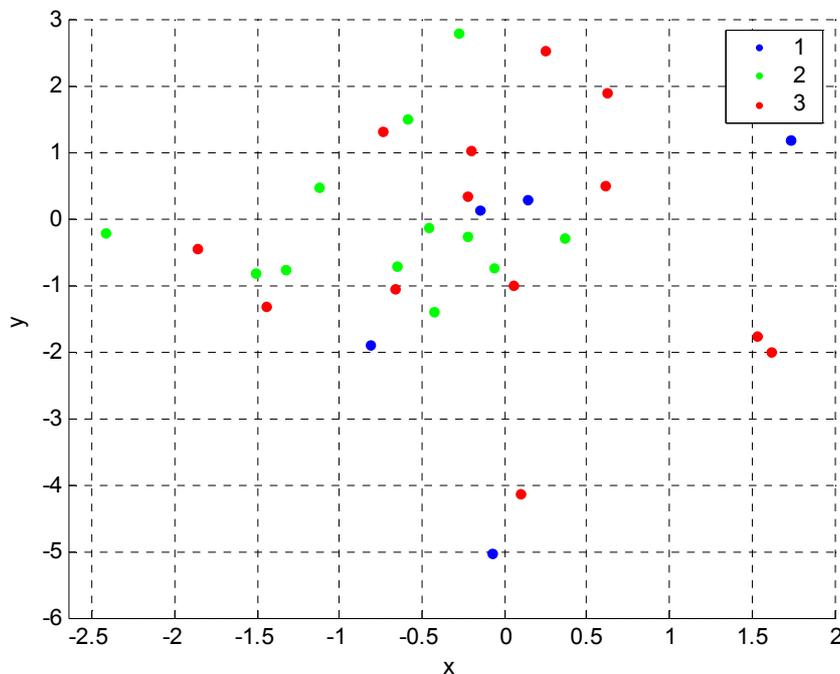


График рассеяния, точки которого разделены на 3 группы. Параметры маркеров заданы как 'clr='cmy', 'sym'='+', siz=10.

```
>> x=normrnd(0,1,30,1);  
>> y=normrnd(0,2,30,1);  
>> g=unidrnd(3,30,1);  
>> gscatter(x,y,g,'cmy','+',10)  
>> grid on
```

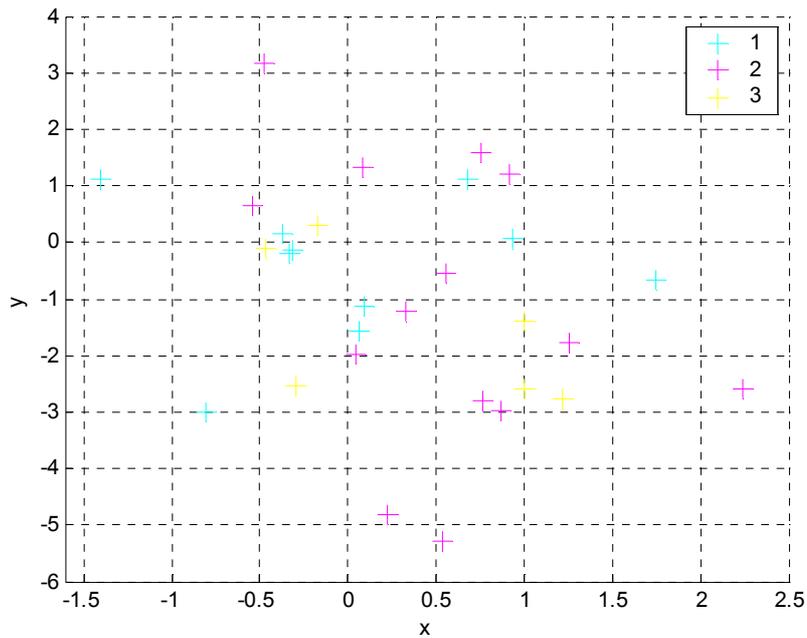


График рассеяния, точки которого разделены на 3 группы. Параметры маркеров заданы как 'clr='cmy', 'sym']='+.o', siz=[5 8 10]. Легенда на график не выводится.

```
>> x=normrnd(0,1,30,1);
>> y=normrnd(0,2,30,1);
>> g=unidrnd(3,30,1);
>> gscatter(x,y,g,'cmy','+.o',[5 8 10], 'off')
>> grid on
```

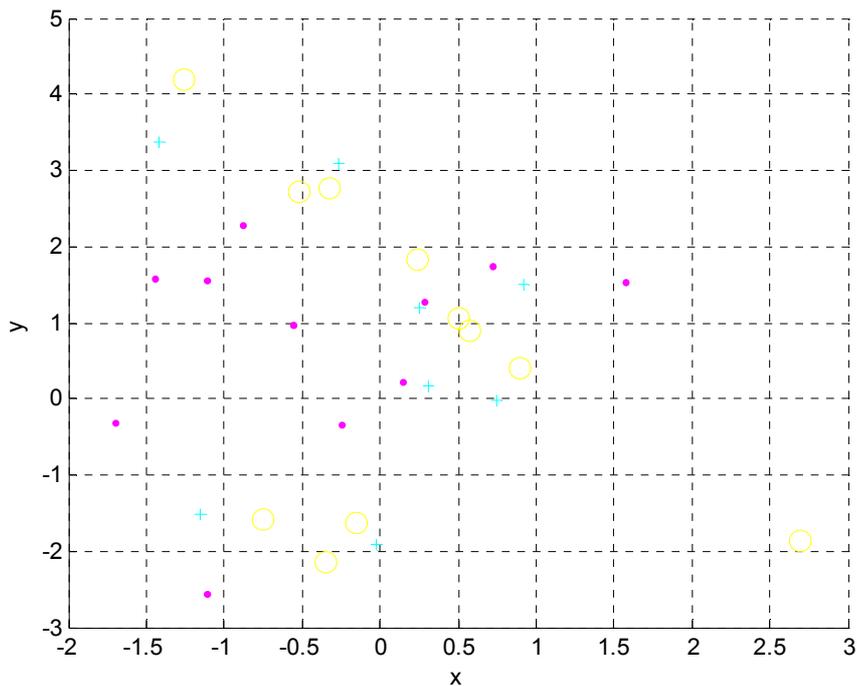
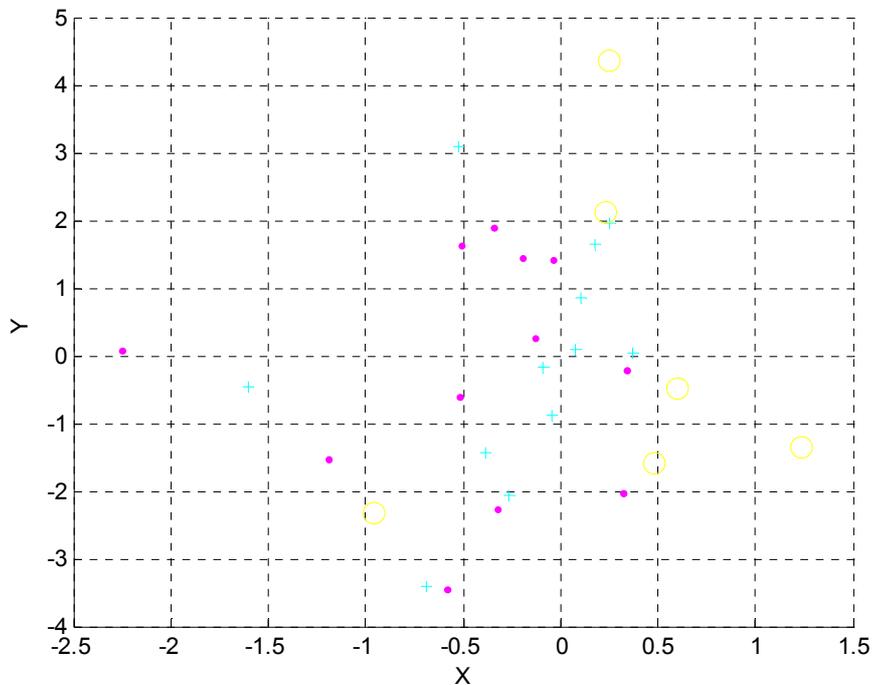


График рассеяния, точки которого разделены на 3 группы. Параметры маркеров заданы как 'clr'='cmy', 'sym'='+.o', siz=[5 8 10]. Легенда на график не выводится. Заданы подписи по осям графика. Выводятся значения указателей на объекты графика.

```
>> x=normrnd(0,1,30,1);  
>> y=normrnd(0,2,30,1);  
>> g=unidrnd(3,30,1);  
>> xnam='X';  
>> ynam='Y';  
>> h = gscatter(x,y,g,'cmy','+.o',[5 8 10], 'off', xnam, ynam)  
h =  
101.0023  
102.0013  
103.0013  
>> grid on
```



lsline

График линейной регрессионной модели

Синтаксис

```
lsline  
h = lsline
```

Описание

`lsline` функция добавляет линейную регрессионную модель для текущего двумерного графика. Коэффициенты линейной модели рассчитываются по методу наименьших квадратов. Для графиков исходных данных `y` которых `LineStyle='-','--','.-'` линейная регрессионная модель не создается. Если в одних координатных осях построены несколько зависимостей, то для каждой из них создается собственный график линейной регрессионной модели.

`h = lsline` функция возвращает вектор указателей `h` на линии графика.

Примеры использования функции создания графика линейной регрессионной модели

График линейной регрессионной модели для вектора `y`. Значения `y` располагаются по порядку следования.

```
>> y = [2 3.4 5.6 8 11 12.3 13.8 16 18.8 19.9]';  
>> plot(y,'+')  
>> grid on  
>> lsline
```

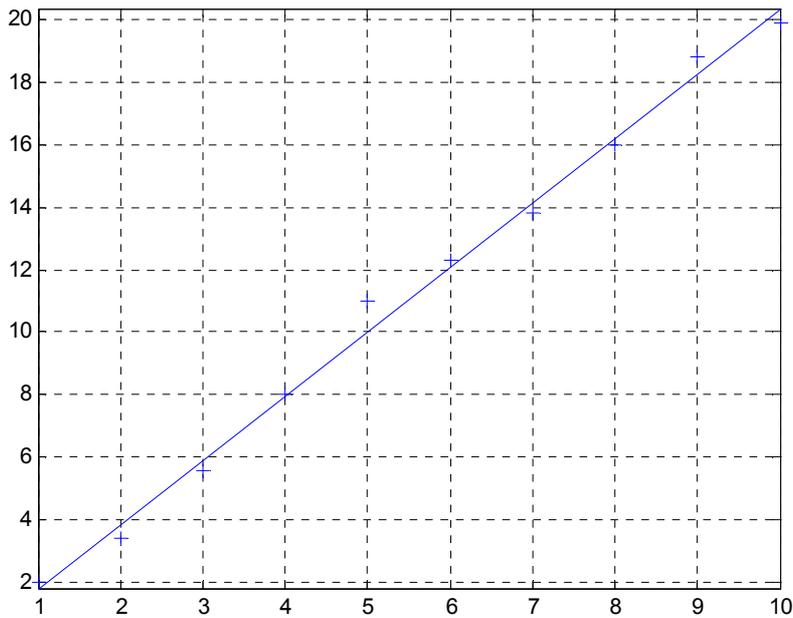
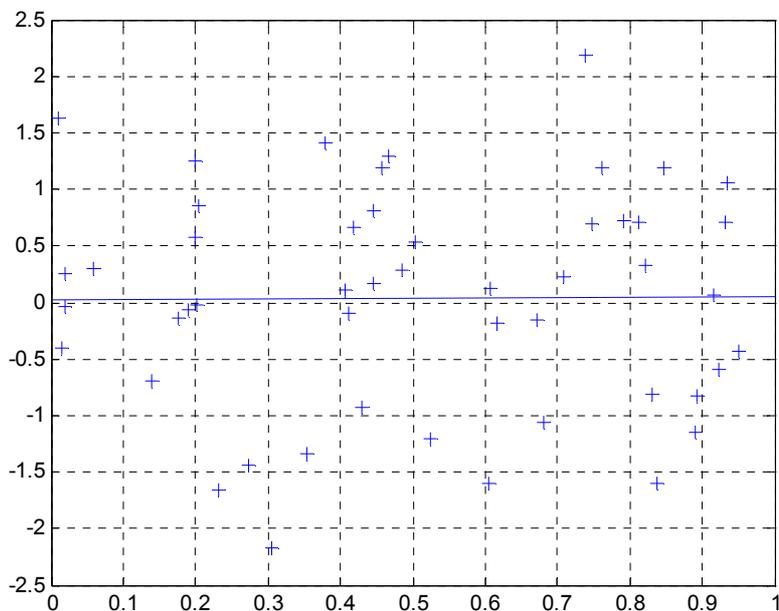


График линейной регрессионной модели для векторов x , y .

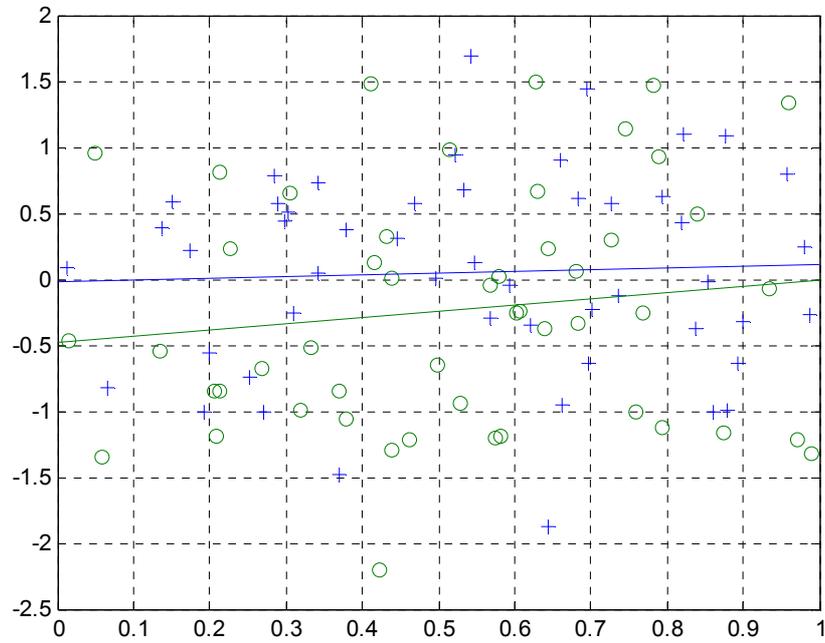
```
>> x =unifrnd(0,1,50,1);
>> y =normrnd(0,1,50,1);
>> plot(x,y,'+')
>> grid on
>> lsline
```



Графики линейной регрессионной модели в одних координатных осях для пары векторов x , y .

```
>> x =unifrnd(0,1,50,2);
>> y =normrnd(0,1,50,2);
```

```
>> plot(x(1:50,1),y(1:50,1),'+', x(1:50,2),y(1:50,2), 'o')
>> grid on
>> h=lsline
h =
109.0002 110.0002
```



normplot

Нормальный вероятностный график

Синтаксис

```
normplot(X)  
h = normplot(X)
```

Описание

`normplot(X)` функция позволяет получить нормальный вероятностный график для выборки X , заданной как вектор. Если X задана в виде матрицы, то для каждой выборки (столбца X) создается нормальный вероятностный график.

Выборочные значения отображаются на графике маркерами \circ . Прямая линия проходит через точки соответствующие 1 и 3 квартилям для каждого столбца X . Полученная прямая экстраполируется до минимального и максимального значений выборки.

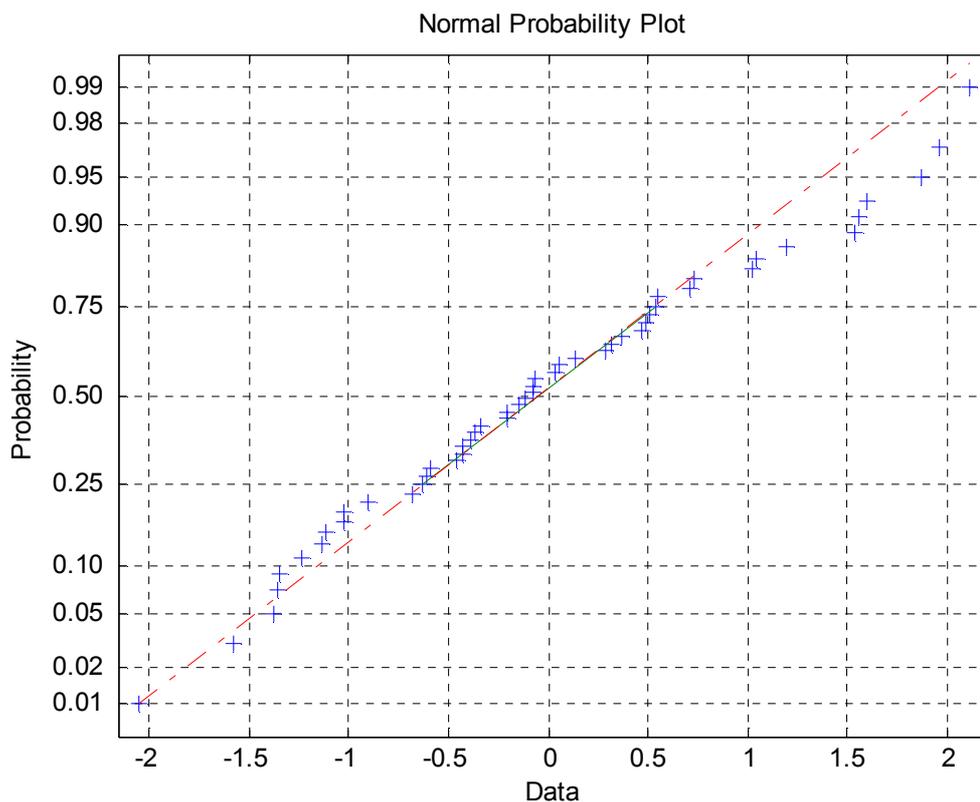
Если выборка не противоречит нормальному закону, значения X должны попасть или быть близки к прямой линии.

`h = normplot(X)` функция возвращает вектор указателей h на линии графика.

Примеры использования функции создания вероятностного нормального графика

Вероятностный нормальный график для выборки распределенной по нормальному закону с нулевым математическим ожиданием и единичной дисперсией. Объем выборки равен 50 элементам. Выборка задана как векторная переменная. Выводятся указатели на объекты графика.

```
>> x = normrnd(0,1,50,1);  
>> h = normplot(x)  
h =  
    111.0002  
     3.0028  
    102.0012
```

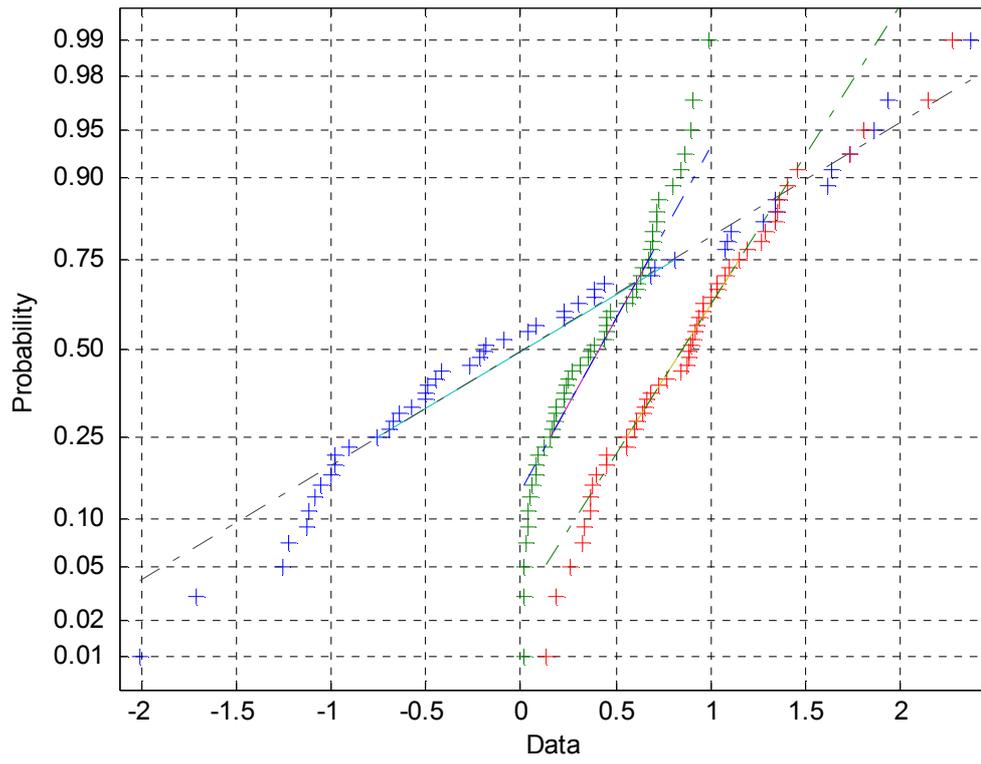


Вероятностный нормальный график для 3-х выборок распределенных по нормальному закону, равномерному закону и закону Вейбулла. Объем выборок равен 50 элементам. Выборки заданы как матричная переменная. Выводятся указатели на объекты графика.

```
>> x1 = normrnd(0,1,50,1);
>> x2 = unifrnd(0,1,50,1);
>> x3 = weibrnd(1,2,50,1);
>> h = normplot([x1 x2 x3])
h =
```

```
3.0035
102.0021
103.0011
104.0011
105.0011
106.0006
107.0007
108.0005
109.0005
```

Normal Probability Plot



pareto

Диаграмма Парето

Синтаксис

```
pareto(y)
pareto(y, names)
h = pareto(...)
```

Описание

`pareto(y, names)` функция позволяет построить диаграмму Парето для частот дефектов `y`. Элементы вектора `y` должны быть целыми положительными числами. Входной аргумент `names` — вектор названий дефектов, задается как вектор строковых переменных. Размерность векторов `y`, `names` должна быть одинаковой.

В варианте синтаксиса `pareto(y)` вместо названий дефектов используются их порядковые номера в векторе `y`.

`h = pareto(...)` функция возвращает указатели на объекты графика.

Диаграмма Парето используется при статистическом контроле качества. Гистограмма на диаграмме Парето соответствует частоте возникновения определенного вида дефекта. Значения частот дефектов сортируются по убыванию. Верхняя линия на графике соответствует кумулятивной кривой частот возникновения дефектов.

Примеры использования функции создания диаграммы Парето

Диаграмма Парето. В качестве идентификаторов дефектов используется вектор строк. В качестве выходных параметров выступают указатели на элементы графика: гистограмму и кумулятивную кривую.

```
>> defects = ['pits '; 'cracks'; 'holes '; 'dents ']
defects =
pits
cracks
holes
dents
>> quantity = [5 3 19 25]
quantity =
    5    3   19   25
>> h=pareto(quantity,defects)
```

h =
101.0023
102.0027

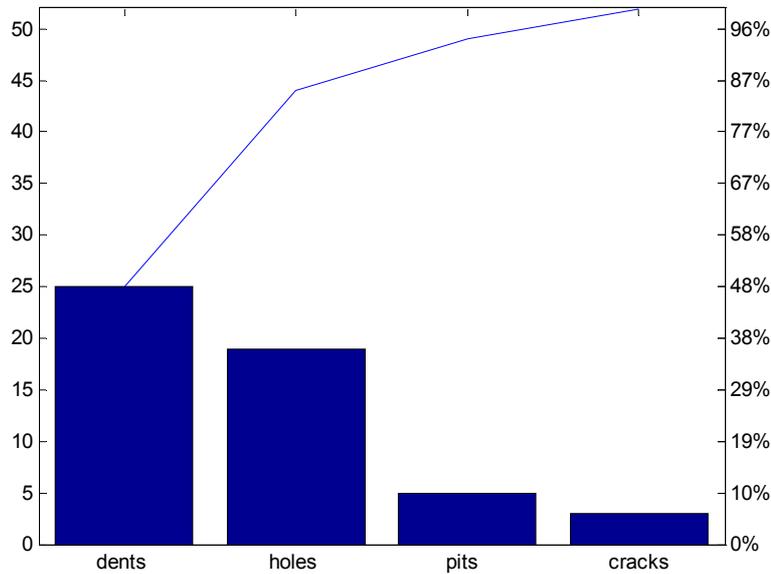
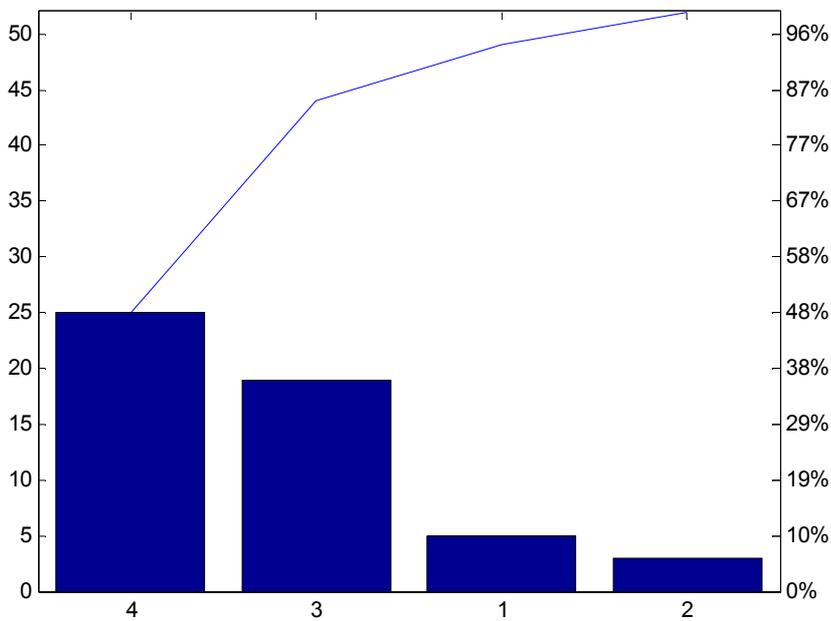


Диаграмма Парето. В качестве идентификаторов используются номера дефектов в векторе quantity.

```
>> quantity = [5 3 19 25];  
>> pareto(quantity)
```



qqplot

График \hat{i} квантиль-квантиль

Синтаксис

```
qqplot(X)  
qqplot(X, Y)  
qqplot(X, Y, pvec)  
h = qqplot(...)
```

Описание

`qqplot(X)` функция позволяет построить график \hat{i} квантиль-квантиль для выборки X и нормального закона. Выборка X должна быть задана в виде вектора. Если распределение значений X не противоречит нормальному закону, то график должен быть линейным. Если X задана как матрица, то график создается для каждого столбца.

`qqplot(X, Y)` функция предназначена для создания графика \hat{i} квантиль-квантиль по выборкам X, Y . Если X и Y имеют одинаковое распределение, то график должен быть линейным. Для X, Y , заданных как матрицы, график \hat{i} квантиль-квантиль создается по их соответствующим столбцам. Размерность X, Y должна совпадать.

Выборочные значения отображаются на графике маркерами \hat{i} . Прямая линия проходит через точки соответствующие 1 и 3 квартилям каждого из распределений. Полученная прямая экстраполируется до минимального и максимального значений выборок.

`qqplot(X, Y, pvec)` такой вариант синтаксиса функции позволяет задать вектор квантилей $pvec$.

`h = qqplot(X, Y, pvec)` функция возвращает вектор указателей на линии графика.

Примеры использования функции создания графика \hat{i} квантиль-квантиль

График \hat{i} квантиль-квантиль для одной нормально распределенной выборки

```
>> X = normrnd(0,1,50,1);  
>> qqplot(X)
```

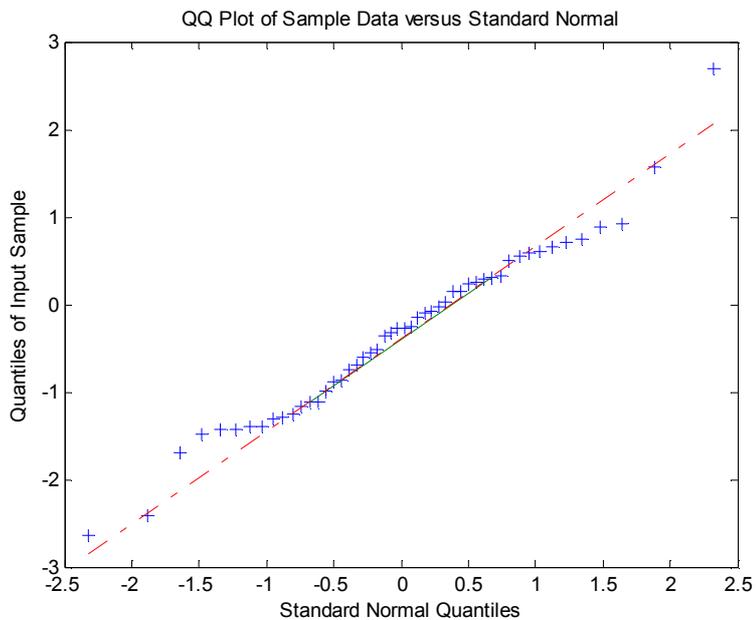


График $\hat{\text{квантиль-квантиль}}$ для нескольких нормально распределенных выборок

```
>> X = normrnd(0,1,50,2);
>> qqplot(X)
```

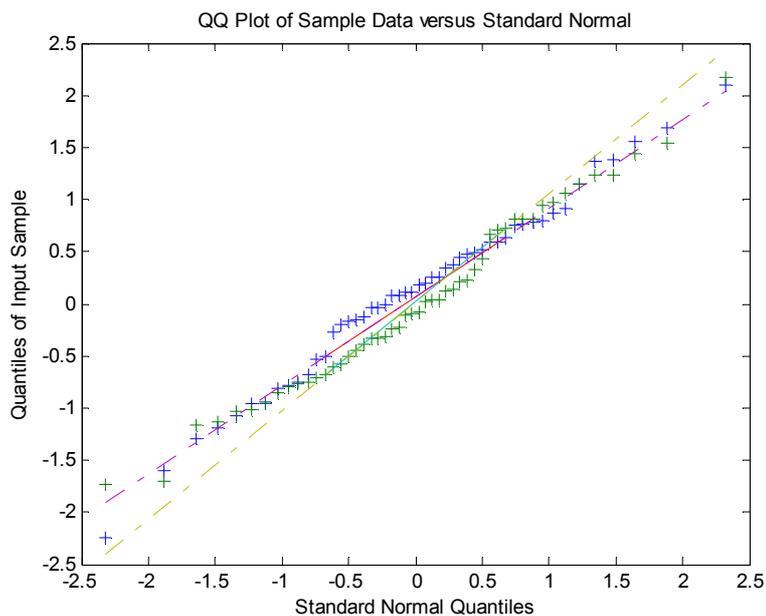


График $\hat{\text{квантиль-квантиль}}$ для сравнения законов распределения двух выборок

```
>> X = normrnd(0,1,50,1);
>> Y = unifrnd(0,1,50,1);
>> qqplot(X,Y)
```

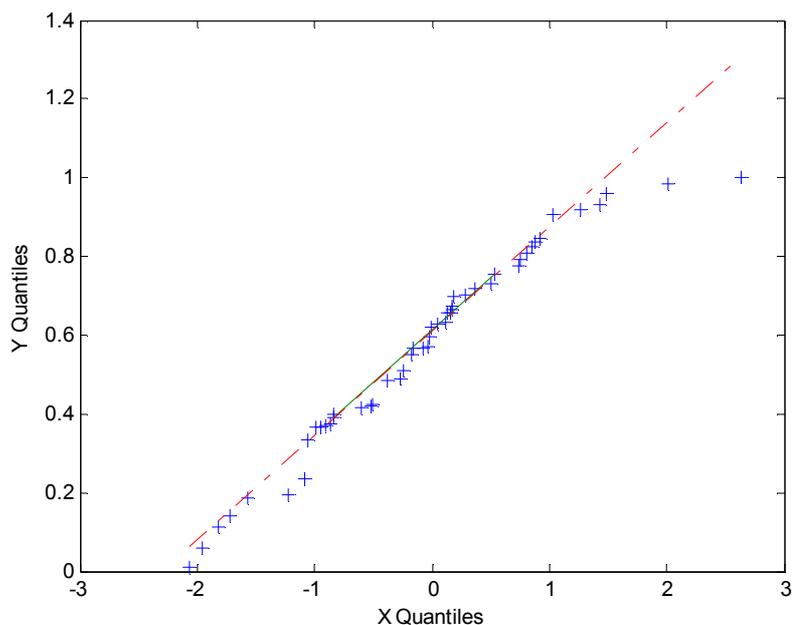


График іквантиль-квантильї для сравнения законов распределения двух пар выборок

```
>> X = normrnd(0,1,50,2);
>> Y = unifrnd(0,1,50,2);
>> qqplot(X,Y)
```

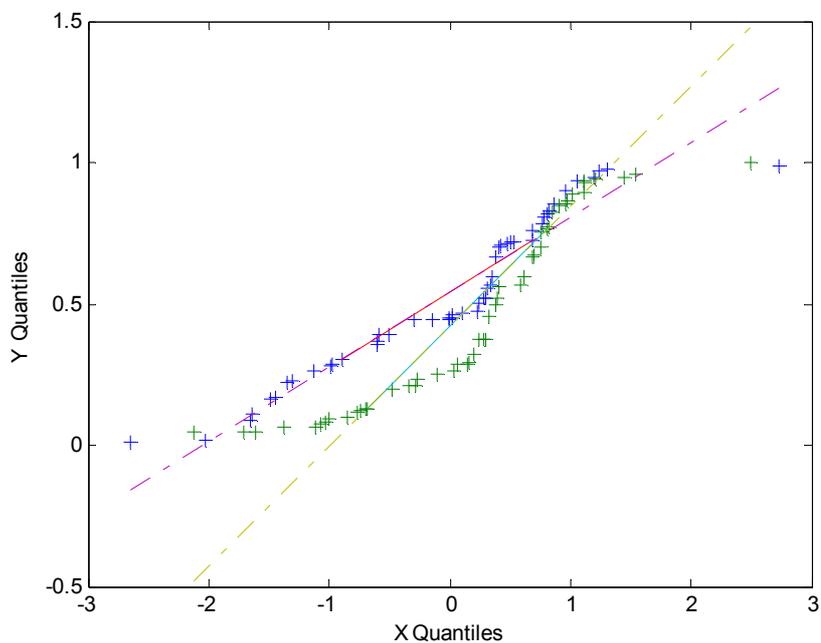
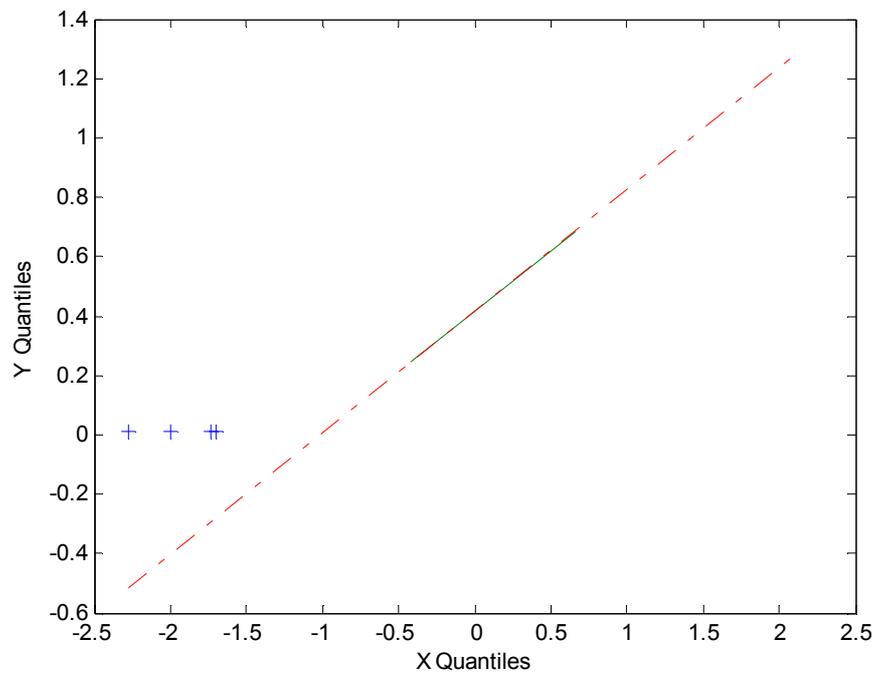


График іквантиль-квантильї для сравнения распределения двух пар выборок. Заданы значения квантилей как элементы вектора rves. Выводятся значения указателей на объекты графика.

```
>> X = normrnd(0,1,50,1);
```

```
>> Y = unifrnd(0,1,50,1);  
>> pvec=[1 2 3 4];  
>> h=qqplot(X,Y,pvec)  
h =  
106.0015  
101.0093  
102.0043
```



rcoplot

График остатков

Синтаксис

```
rcoplot(r,rint)
```

Описание

`rcoplot(r,rint)` функция предназначена для построения графика остатков r с доверительными интервалами $rint$. Остатки определяются как разность внутригрупповых средних зависимой переменной и соответствующих значений регрессионной модели. Величины остатков и доверительных интервалов откладываются по оси ординат в порядке их следования в векторе r . Векторы r и $rint$ являются выходными аргументами функции `regress`.

Примеры использования функции создания графика остатков

```
>> X = [ones(10,1) (1:10)']
X =
     1     1
     1     2
     1     3
     1     4
     1     5
     1     6
     1     7
     1     8
     1     9
     1    10
>> y = X * [10;1] + normrnd(0,0.1,10,1)
y =
 11.0441
 12.0565
 12.9306
 14.0834
 14.7763
 16.1098
 16.9998
 17.8385
 18.8771
 20.0207
>> [b,bint,r,rint] = regress(y,X,0.05)
b =
 10.0351
```

```
0.9888
bint =
  9.8561 10.2142
  0.9600 1.0177
r =
  0.0201
  0.0437
 -0.0710
  0.0929
 -0.2030
  0.1417
  0.0429
 -0.1072
 -0.0575
  0.0973
rint =
 -0.2059 0.2462
 -0.1962 0.2837
 -0.3178 0.1758
 -0.1560 0.3419
 -0.4008 -0.0052
 -0.0933 0.3766
 -0.2162 0.3020
 -0.3439 0.1294
 -0.2952 0.1803
 -0.1129 0.3076
>> rcoplot(r,rint)
```

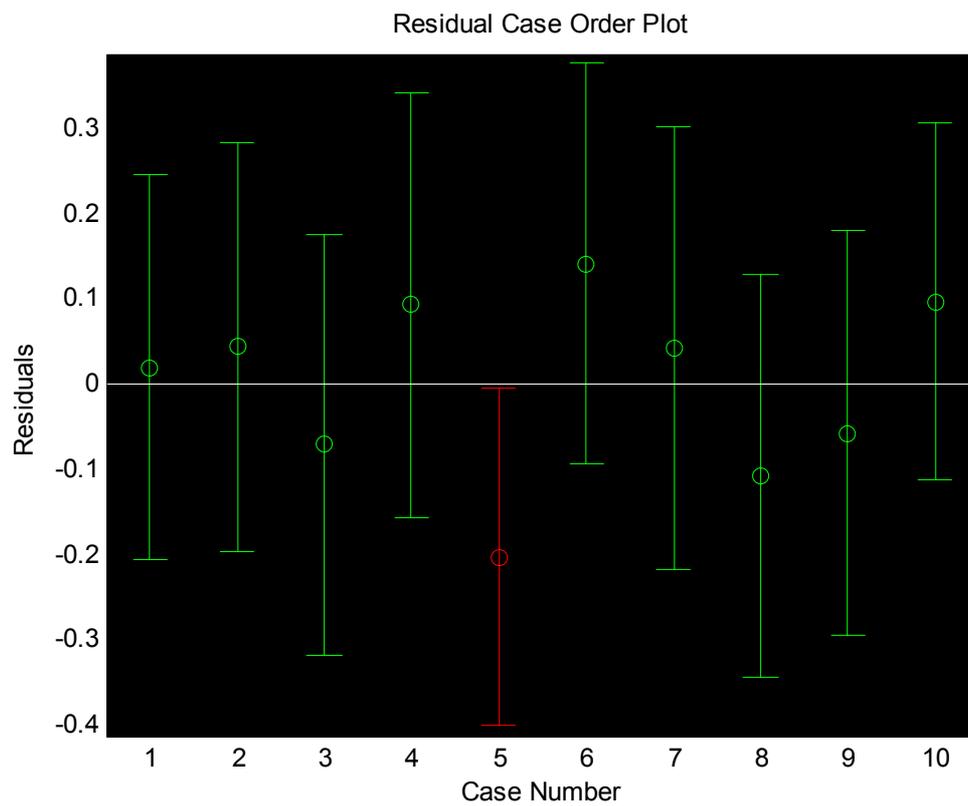


График показывает величины остатков и 95% доверительных интервалов. Центральная линия соответствует нулевой разницы внутригруппового среднего и значения регрессионной модели.

refcurve

График полиномиальной модели для текущей диаграммы рассеяния

Синтаксис

```
h = refcurve(p)
```

Описание

`refcurve(p)` функция добавляет на текущую диаграмму рассеяния график полиномиальной модели `p`. Вид полинома определяется формулой:

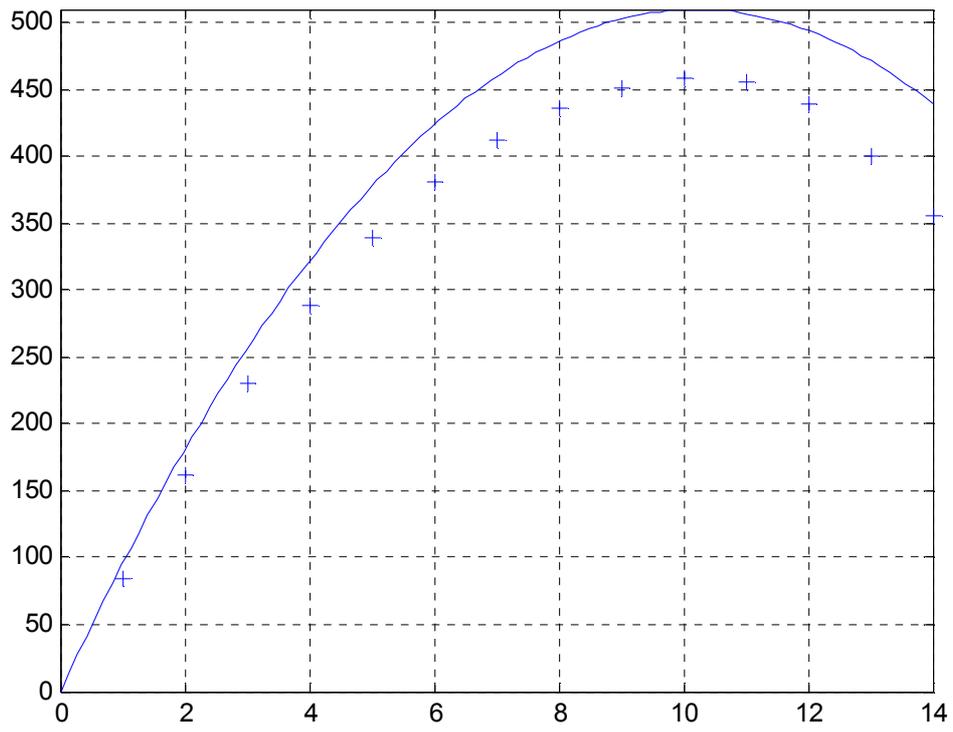
$$y = p_1x^n + p_2x^{(n-1)} + \dots + p_nx + p_{n+1}$$

Следует отметить, что полиномиальная модель задается вектором коэффициентов `p`, где `p(1)` — коэффициент при высшем порядке аргумента. Порядок полинома определяется как `(size(p)-1)`.

`h = refcurve(p)` функция возвращает указатель `h` на кривую полиномиальной модели.

Примеры использования функции создания графика полиномиальной модели для текущей диаграммы рассеяния

```
>> h = [85 162 230 289 339 381 413 437 452 458 456 440 400 356];  
>> plot(h,'+')  
>> grid on  
>> h=refcurve([-4.9 100 0])  
h =  
102.0045
```



refline

График линейного полинома для текущей диаграммы рассеяния

Синтаксис

```
refline(slope, intercept)
refline(slope)
h = refline(slope, intercept)
refline
```

Описание

`refline(slope, intercept)` функция добавляет график линейного полинома с заданными коэффициентом наклона `slope` и постоянным смещением `intercept` к текущей диаграмме рассеяния.

`refline(slope)` в этом варианте синтаксиса функции коэффициенты линейного полинома определяются как элементы вектора `slope`:

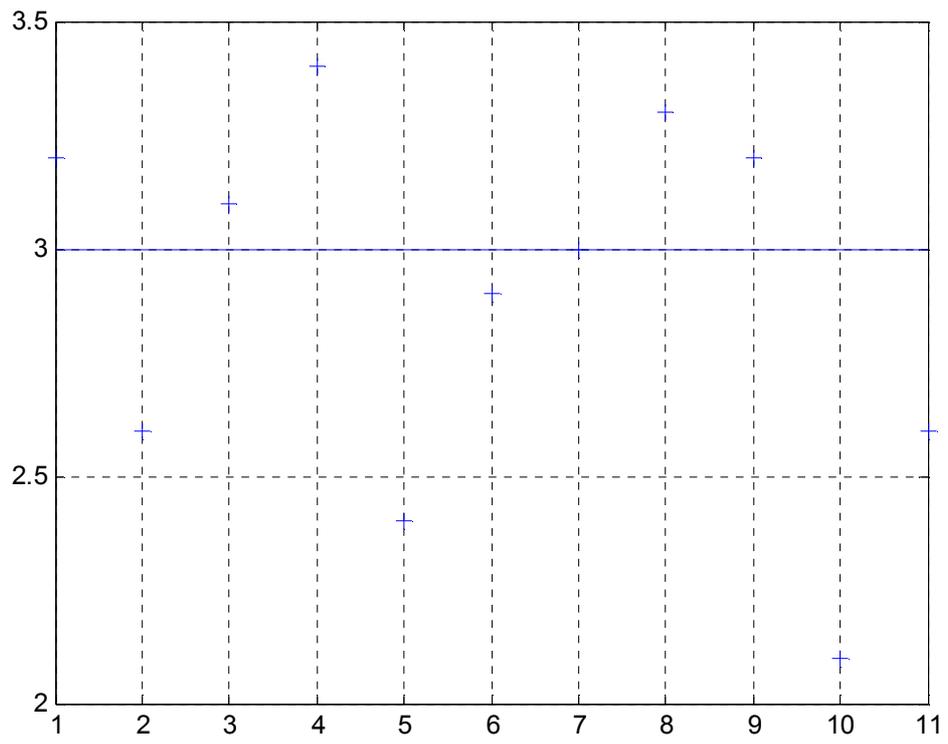
$$y = \text{slope}(2) + \text{slope}(1)*x$$

`h = refline(slope, intercept)` функция возвращает указатель на график линейного полинома.

Для исходных данных зависимости которых имеют `LineStyle='-','--','.-'`, график линейного полинома не создается.

Примеры использования функции создания графика линейного полинома для текущей диаграммы рассеяния

```
>> y = [3.2 2.6 3.1 3.4 2.4 2.9 3.0 3.3 3.2 2.1 2.6]';
>> plot(y, '+')
>> refline(0,3)
>> grid on
```



surfht

Интерактивный контурный график

Синтаксис

```
surfht (Z)  
surfht (x, y, Z)
```

Описание

`surfht (Z)` функция предназначена для построения интерактивного контурного графика матрицы Z . Значения элементов матрицы Z являются аппликатами поверхности отклика для аргументов x , y . Независимые переменные x и y определяются как номера столбцов и строк матрицы Z соответственно.

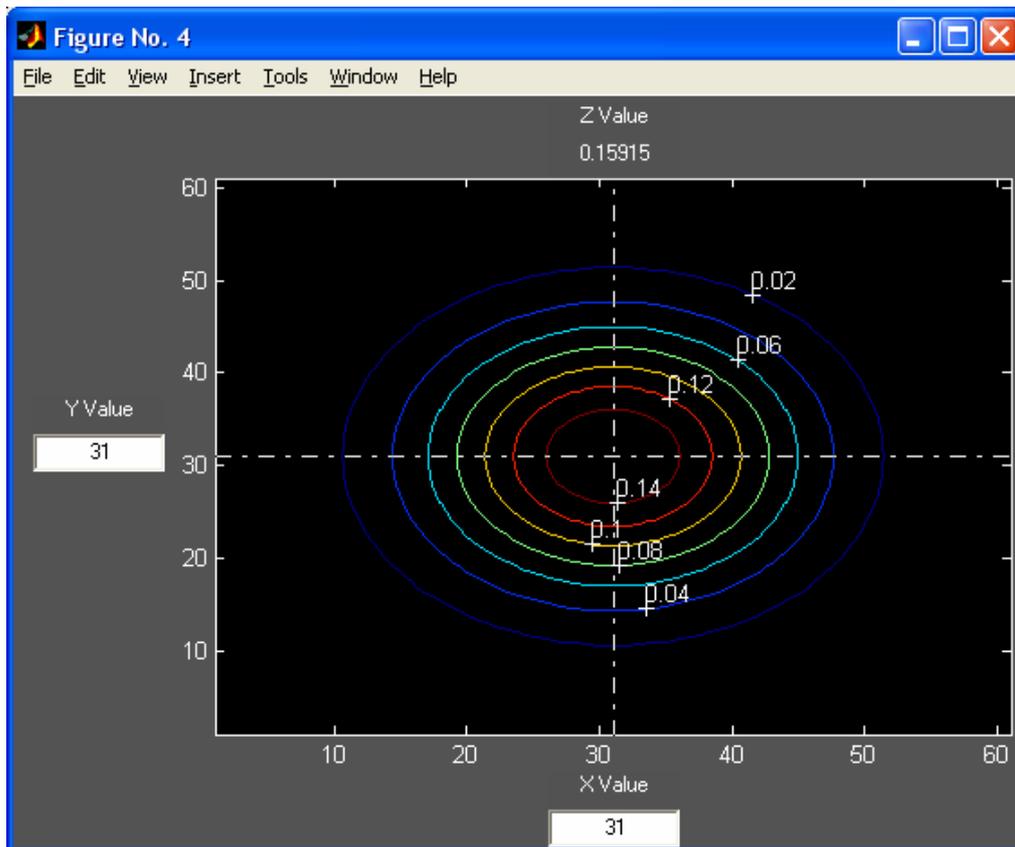
`surfht (x, y, Z)` этот вариант синтаксиса функции позволяет в явном виде задать значения независимых переменных x и y . Количество элементов вектора x должно быть равно числу столбцов матрицы Z . Число элементов вектора y должно соответствовать количеству строк матрицы Z .

Пересекающиеся вертикальная и горизонтальная штрих-пунктирные линии на графике определяют текущую пару значений по осям X Value и Y Value, для которой рассчитывается значение выходной функции Z Value. Положение точки пересечения линий, определяющее текущую пару значений x и y , можно изменить левой кнопкой мыши, или явно указать значения координат в строках ввода X Value и Y Value. Значение функции будет пересчитано с изменением положения точки пересечения линий или при вводе одной из координат в строку ввода.

Примеры использования функции интерактивного контурного графика

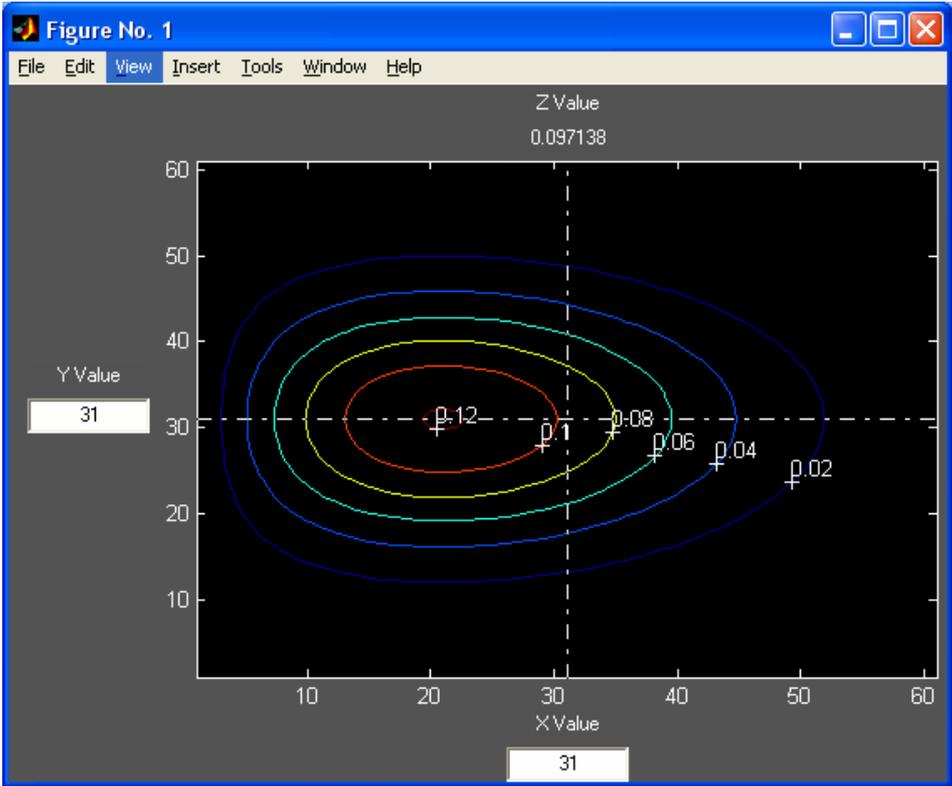
Интерактивный контурный график для матрицы аппликат

```
>> X=-3:0.1:3;  
>> Y=-3:0.1:3;  
>> [x y]=meshgrid(X,Y);  
>> Z=normpdf(x,0,1).* normpdf(y,0,1);  
>> surfht(Z)
```



Интерактивный контурный график для матрицы аппликат и явно заданных независимых переменных

```
>> X=0:0.1:6;
>> Y=-3:0.1:3;
>> [x y]=meshgrid(X,Y);
>> Z=raylpdf(x,2).* normpdf(y,0,1);
>> surfht(Z)
```



weibplot

Вероятностный график Вейбулла

Синтаксис

```
weibplot(X)  
h = weibplot(X)
```

Описание

`weibplot(X)` функция позволяет построить вероятностный график Вейбулла для выборки X , заданной как вектор. Если X задана в виде матрицы, то для каждой выборки (столбца X) создается вероятностный график Вейбулла.

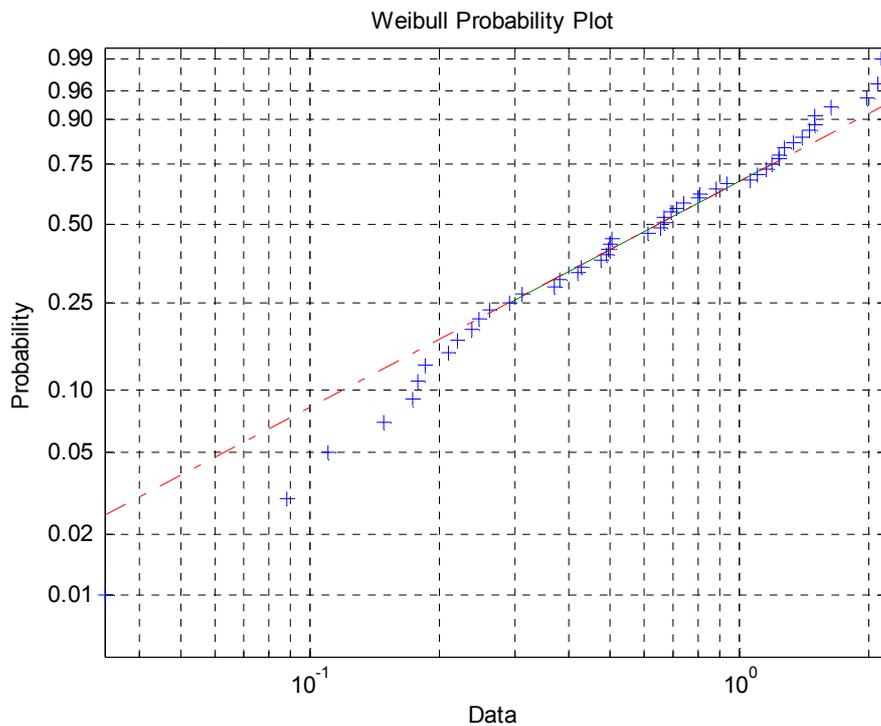
Выборочные значения отображаются на графике маркерами \times . Если выборка не противоречит закону Вейбулла, значения X должны попасть или быть близки к прямой линии.

`h = weibplot(X)` функция возвращает вектор указателей h на линии графика.

Примеры использования функции создания вероятностного графика Вейбулла

Вероятностный график для выборки распределенной по закону Вейбулла. Объем выборки равен 50 элементам. Выборка задана как векторная переменная. Выводятся указатели на объекты графика.

```
>> r = weibrnd(1.2,1.5,50,1);  
>> h=weibplot(r)  
h =  
    115.0005  
    214.0010  
    215.0005
```

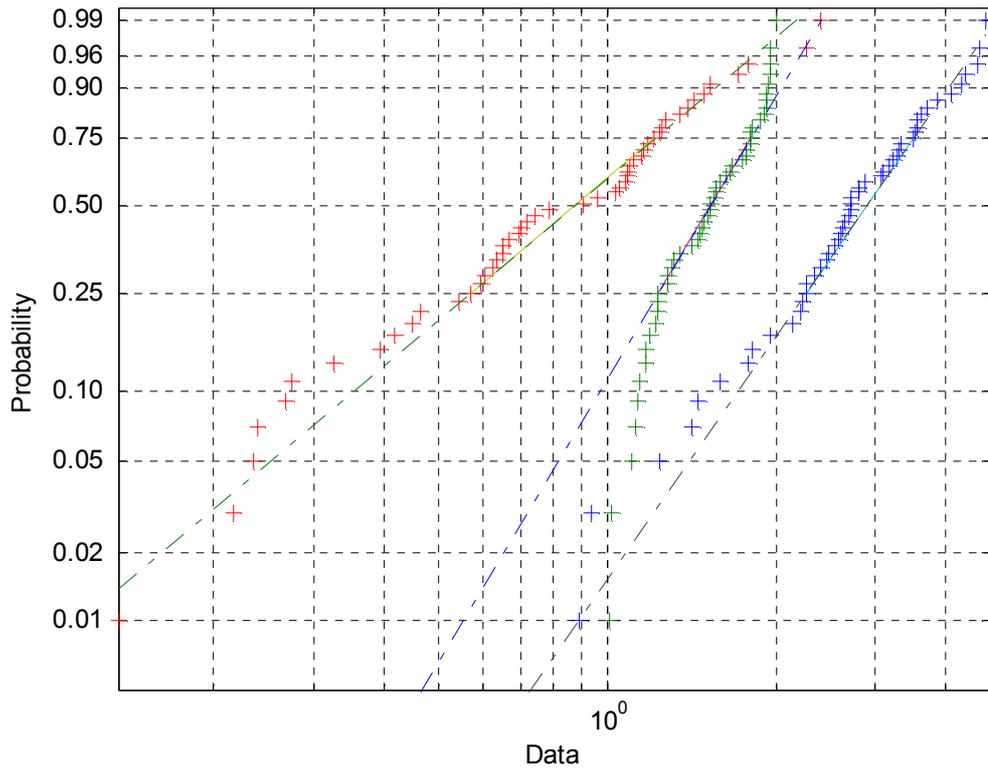


Вероятностный график Вейбулла для 3-х выборок распределенных по нормальному закону, равномерному закону и закону Вейбулла. Объем выборок равен 50 элементам. Выборки заданы как матричная переменная. Выводятся указатели на объекты графика.

```
>> x1 = normrnd(3,1,50,1);
>> x2 = unifrnd(1,2,50,1);
>> x3 = weibrnd(1,2,50,1);
>> h = weibplot ([x1 x2 x3])
```

```
h =
    3.0056
   102.0057
   103.0039
   104.0037
   105.0028
   106.0022
   107.0024
   108.0031
   109.0017
```

Weibull Probability Plot



Описательная статистика в MATLAB 6.5

mean

Расчет среднего арифметического значения

Синтаксис

```
m = mean(X)
```

Описание

`m = mean(X)` функция предназначена для расчета среднего арифметического значения m выборки X . Выборка X может быть представлена в виде вектора или матрицы. Расчет проводится по формуле

$$\bar{X}_j = \frac{1}{n} \sum_{i=1}^n X_{ij},$$

n — число элементов в векторе, или строк в матрице X .

Если X задана как вектор, то среднее арифметическое значение рассчитывается по всем его элементам. Для выборки определенной в виде матрицы среднее арифметическое значение рассчитывается для каждого столбца X .

`mean` является функцией ядра `matlab`.

Примеры использования функции расчета среднего арифметического значения

Расчет среднего арифметического значения для выборки X заданной в виде вектора

```
>> X = normrnd(0,1,100,1);
```

```
>> xbar = mean(X)
```

```
xbar =  
-0.0573
```

Расчет среднего арифметического значения для выборки X заданной в виде матрицы

```
>> X = normrnd(0,1,100,5);
```

```
>> xbar = mean(X)
```

```
xbar =  
-0.0088 -0.0903 -0.0695 0.0525 -0.1840
```

std

Расчет несмещенной точечной оценки среднего квадратического отклонения выборки

Синтаксис

```
s = std(X)
```

Описание

`s = std(X)` функция предназначена для расчета несмещенной точечной оценки среднего квадратического отклонения s выборки X . Выборка X может быть представлена в виде вектора или матрицы. Если X задан как вектор, то среднее квадратическое отклонение рассчитывается по всем его элементам. Для выборки определенной в виде матрицы среднее квадратическое отклонение рассчитывается для каждого столбца X .

Расчет проводится по формуле

$$s_j = \left(\frac{1}{n-1} \sum_{i=1}^n (X_{ij} - \bar{X}_j)^2 \right)^{\frac{1}{2}},$$

где \bar{X} - среднее арифметическое значение, $\bar{X}_j = \sum_{i=1}^n X_{ij} / n$, n - число элементов в векторе, или строк в матрице X .

`std` является функцией ядра `matlab`.

Примеры использования функции расчета несмещенной точечной оценки среднего квадратического отклонения выборки

Расчет точечной оценки среднего квадратического отклонения для выборки X заданной в виде вектора

```
>> X = normrnd(0,1,100,1);  
>> s = std(X)  
s =  
    0.9975
```

Расчет точечной оценки среднего квадратического отклонения для выборки X заданной в виде матрицы

```
>> X = normrnd(0,1,100,5);  
>> s = std(X)  
s =  
    0.9927    1.1022    0.9771    0.9909    1.0306
```

var

Расчет выборочной дисперсии

Синтаксис

$$D = \text{var}(X)$$

$$D = \text{var}(X, 1)$$

$$D = \text{var}(X, w)$$

Описание

$D = \text{var}(X)$ функция предназначена для расчета точечной оценки дисперсии D выборки X . Выборка X может быть представлена в виде вектора или матрицы. Если X задана как вектор, то точечная оценка дисперсии рассчитывается по всем его элементам. Для выборки определенной в виде матрицы точечная оценка дисперсии рассчитывается для каждого столбца X .

Вариант синтаксиса $\text{var}(X)$ предназначен для расчета точечной оценки дисперсии по формуле

$$D_j = \frac{1}{n-1} \sum_{i=1}^n (X_{ij} - \bar{X}_j)^2, \quad (1)$$

где \bar{X}_j - среднее арифметическое значение выборки, $\bar{X}_j = \sum_{i=1}^n X_{ij} / n$, n - число элементов в векторе, или строк в матрице X . Если выборка распределена по нормальному закону $\text{var}(X)$ позволяет получить несмещенную точечную оценку с минимальной дисперсией.

$D = \text{var}(x, 1)$ синтаксис функции предназначен для расчета точечной оценки дисперсии выборки по формуле

$$D_j = \frac{1}{n} \sum_{i=1}^n (X_{ij} - M[X]_j)^2, \quad (2)$$

где $M[X]_j$ - математическое ожидание выборки.

$D = \text{var}(X, w)$ функция предназначена для расчета точечной оценки дисперсии выборки X с учетом весов отдельных значений w . Веса значений выборки w задаются как вектор с положительными элементами. Количество элементов вектора w должно быть равно числу строк в матрице X . Если выборка X задана как вектор, то число элементов X и w должно быть одинаковым.

Таким образом, функция `var` поддерживает два основным метода оценки дисперсии выборки: оценку по методу максимального правдоподобия - (2) и определение несмещенной оценки с минимальной дисперсией - (1).

Примеры использования функции расчета точечной оценки дисперсии выборки

Расчет точечной оценки дисперсии выборки X заданной как вектор по формуле (1)

```
>> X = normrnd(0,1,100,1);  
>> D = var(X)  
D =  
    0.7712
```

Расчет точечной оценки дисперсии выборки X заданной как матрица по формуле (1)

```
>> X = normrnd(0,1,100,5);  
>> D = var(X)  
D =  
    0.9541    0.8784    1.0472    0.8623    0.9662
```

Расчет точечной оценки дисперсии выборки X заданной как матрица по формуле (2)

```
>> X = normrnd(0,1,100,5);  
>> D = var(X,1)  
D =  
    0.9445    0.8696    1.0367    0.8537    0.9565
```

Расчет точечной оценки дисперсии выборки X заданной как вектор с учетом весов наблюдаемых значений

```
>> X = normrnd(0,1,100,1);  
>> w=unidrnd(10,100,1);  
>> D = var(X,w)  
D =  
    0.7967
```

Расчет точечной оценки дисперсии выборки X заданной как матрица с учетом весов наблюдаемых значений

```
>> X = normrnd(0,1,100,5);  
>> w=unidrnd(10,100,1);  
>> D = var(X,w)  
D =  
    0.9582    0.9632    0.9897    0.8257    0.9583
```

median

Расчет медианы выборки

Синтаксис

```
m = median(X)
```

Описание

`m = median(X)` функция предназначена для расчета медианы m выборки X . Медиана рассчитывается как 50% процентиль выборки и является робастной оценкой центра группирования значений выборки при наличии выбросов. Выборка X может быть представлена в виде вектора или матрицы. Если X задана как вектор, то медиана рассчитывается по всем его элементам. Для выборки определенной в виде матрицы медиана рассчитывается для каждого столбца X .

Функция `median` использует функцию сортировки `sort`. Это может повлечь увеличение времени расчета для большой матрицы X .

`median` является функцией ядра `matlab`.

Примеры использования функции расчета медианы выборки

Расчет медианы выборки X заданной как вектор

```
>> X = 1:10
X =
     1     2     3     4     5     6     7     8     9    10
>> m = median(X)
m =
     5.5000
```

Расчет медианы выборки X заданной в виде матрицы

```
>> X=unidrnd(10,5,5)
X =
     3     4     7     6     2
     6     7     9     8     8
    10     7     4     6    10
     4     4     5     8     9
     7     5     6     5     8
>> m = median(X)
m =
     6     5     6     6     8
```

Демонстрация устойчивости медианы как точечной оценки центра группирования значений выборки при наличии выбросов

```
>> X=unidrnd(10,1,10)
```

```
X =
```

```
1 4 6 3 6 1 6 8 10 8
```

```
>> m = median(X)
```

```
m =
```

```
6
```

```
>> X(1,10)=1000;
```

```
>> X
```

```
X =
```

```
Columns 1 through 9
```

```
1 4 6 3 6 1 6 8 10
```

```
Column 10
```

```
1000
```

```
>> m = median(X)
```

```
m =
```

```
6
```

geomean

Расчет среднего геометрического выборки

Синтаксис

```
m = geomean(X)
```

Описание

`m = geomean(X)` — функция предназначена для расчета значения среднего геометрического m выборки X . Если X задана как вектор, то среднее геометрическое значение рассчитывается по всем его элементам. Для выборки определенной в виде матрицы среднее геометрическое значение рассчитывается для каждого столбца X .

Расчет среднего геометрического выборки выполняется по формуле

$$m_j = \left[\prod_{i=1}^n X_{ij} \right]^{\frac{1}{n}},$$

где n — объем выборки.

Примеры использования функции расчета среднего геометрического значения

Расчет среднего геометрического значения для выборки X заданной в виде вектора

```
>> X = normrnd(10,1,100,1);  
>> m = geomean(X)  
m =  
9.8288
```

Расчет среднего геометрического значения для выборки X заданной как матрица

```
>> X = normrnd(10,1,100,5);  
>> m = geomean(X)  
m =  
9.8756 9.9409 9.8043 9.8842 10.0197
```

Величина среднего геометрического выборки X должна быть меньше или равна среднему арифметическому значению

```
>> X = normrnd(10,1,100,5);  
>> m = geomean(X)  
m =  
9.8945 10.0551 9.8076 9.9412 9.8391  
>> xbar = mean(X)
```

xbar =

9.9457 10.0980 9.8506 10.0064 9.8963

harmmean

Расчет среднего гармонического выборки

Синтаксис

```
m = harmmean(X)
```

Описание

`m = harmmean(X)` функция предназначена для расчета значения среднего гармонического m выборки X . Если X задан как вектор, то среднее гармоническое значение рассчитывается по всем его элементам. Для выборки определенной в виде матрицы среднее гармоническое значение рассчитывается для каждого столбца X .

Расчет среднего гармонического выборки выполняется по формуле

$$m_j = \frac{n}{\sum_{i=1}^n \frac{1}{X_{ij}}},$$

где n — объем выборки.

Примеры использования функции расчета среднего гармонического значения

Расчет среднего гармонического значения выборки X заданной в виде вектора

```
>> X = normrnd(10,1,100,1);  
>> m = harmmean(X)  
m =  
9.8263
```

Расчет среднего гармонического значения выборки X заданной в виде матрицы

```
>> X = normrnd(10,1,100,5);  
>> m = harmmean(X)  
m =  
10.0391 10.0635 10.0522 10.1031 9.8092
```

Величина среднего гармонического выборки X должна быть меньше или равна среднему арифметическому значению

```
>> X = normrnd(10,1,100,5);  
>> m = harmmean(X)  
m =
```

```
9.8292 10.0088 10.0028 9.9007 9.9302
>> xbar = mean(X)
xbar =
9.9206 10.1203 10.1265 10.0111 10.0221
```

range

Расчет размаха выборки

Синтаксис

```
R = range(X)
```

Описание

`R = range(X)` функция предназначена для расчета размаха R выборки X . Если X задана как вектор, то размах рассчитывается по всем его элементам. Для выборки определенной в виде матрицы размах рассчитывается для каждого столбца X .

Расчет размаха выполняется по формуле

$$R_j = \max\{X_j\} - \min\{X_j\}.$$

Размах является точечной оценкой разброса значений выборки. Размах чувствителен к грубым промахам, что делает его ненадежной оценкой разброса.

Примеры использования функции расчета размаха выборки

Расчет размаха выборки X заданной в виде вектора

```
>> X = normrnd(10,1,100,1);
```

```
>> R = range(X)
```

```
R =
```

```
5.7830
```

Расчет размаха выборки X заданной в виде матрицы

```
>> X = normrnd(10,1,100,5);
```

```
>> R = range(X)
```

```
R =
```

```
4.4707 6.2069 5.4199 4.2831 4.5227
```

Оценка влияния грубых промахов на величину размаха

```
>> X = normrnd(10,1,100,1);
```

```
>> R = range(X)
```

```
R =
```

```
4.9442
```

```
>> X(100,1) = -8;
```

```
>> R = range(X)
```

```
R =
```

```
20.8149
```

kurtosis

Расчет точечной оценки коэффициента эксцесса выборки

Синтаксис

```
k = kurtosis(X)
k = kurtosis(X, flag)
```

Описание

`k = kurtosis(X)` функция предназначена для расчета точечной оценки коэффициента эксцесса k выборки X . Если X задана как вектор, то точечная оценка коэффициента эксцесса рассчитывается по всем его элементам. Для выборки определенной в виде матрицы точечная оценка коэффициента эксцесса рассчитывается для каждого столбца X .

Расчет точечной оценки коэффициента эксцесса выборки выполняется по формуле

$$k_j = \frac{E(X_j - \bar{X}_j)^4}{\sigma_j^4},$$

где \bar{X}_j - среднее арифметическое значение выборки X_j , σ_j - точечная оценка среднего квадратического отклонения выборки X_j , $E(t)$ - наиболее вероятная оценка параметра t .

Коэффициент эксцесса показывает насколько выборка X по наклону кривой функции плотности вероятности соответствует нормальному закону. Для нормального закона коэффициент эксцесса равен 3. Законы распределения с более острой вершиной, чем у нормального имеют коэффициент эксцесса более 3 и с менее острой вершиной $\hat{\mu}$ менее 3.

Примечание: в отечественной литературе коэффициент эксцесса определяется по формуле $k_j = \frac{E(X_j - \bar{X}_j)^4}{\sigma_j^4} - 3$. Таким образом, коэффициент эксцесса нормального закона равен 0.

`k = kurtosis(X, flag)` функция позволяет рассчитать несмещенную (`flag=0`) и смещенную (`flag=1`, значение по умолчанию) точечную оценку коэффициента эксцесса k выборки X . Величина смещения выборочного коэффициента эксцесса зависит от объема выборки. Если X является выборкой из генеральной совокупности, для

получения несмещенной точечной оценки коэффициента эксцесса flag должен быть равен 0.

Примеры использования функции расчета точечной оценки коэффициента эксцесса выборки

Расчет точечной оценки коэффициента эксцесса выборки X, заданной в виде вектора

```
>> X = normrnd(10,1,100,1);  
>> k = kurtosis(X)  
k =  
    2.6987
```

Расчет точечной оценки коэффициента эксцесса выборки X заданной в виде матрицы

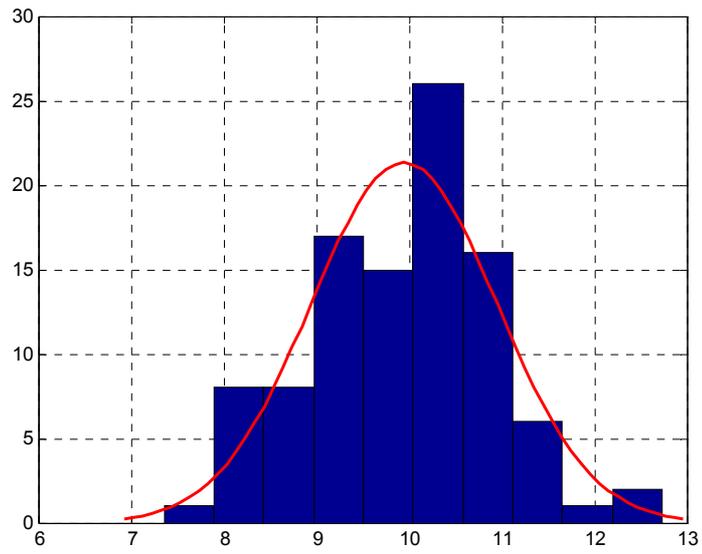
```
>> X = normrnd(10,1,100,5);  
>> k = kurtosis(X)  
k =  
    2.4775    2.9325    2.6933    3.0013    3.0669
```

Расчет смещенной и несмещенной точечных оценок коэффициента эксцесса выборки X

```
>> X = normrnd(10,1,100,1);  
>> k = kurtosis(X,1)  
k =  
    2.7772  
>> k = kurtosis(X,0)  
k =  
   -0.1718
```

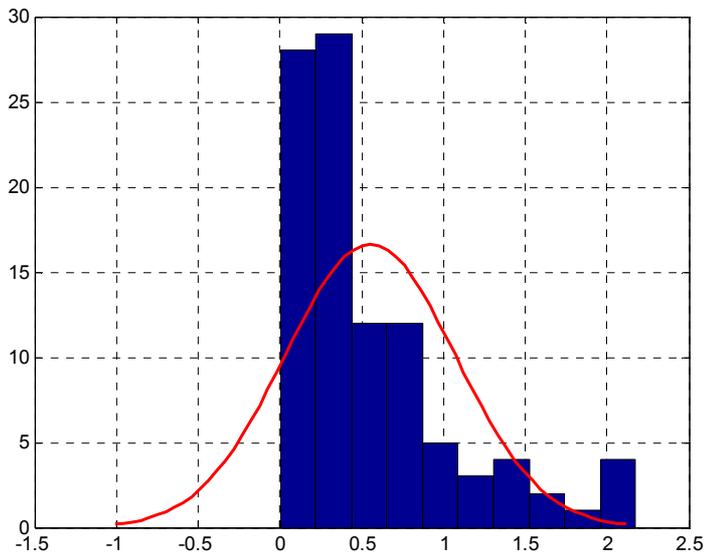
Сравнение распределения выборок и значений коэффициента эксцесса для нормального закона и закона Вейбулла

```
>> X1 = normrnd(10,1,100,1);  
>> X2 = weibrnd(2,1,100,1);  
>> X = [X1 X2];  
>> k = kurtosis(X)  
k =  
    3.1280    4.5821  
>> histfit(X1)  
>> grid on
```



```
>> histfit(X2)
```

```
>> grid on
```



mad

Расчет среднего абсолютного отклонения значений выборки

Синтаксис

```
y = mad(X)
```

Описание

`y = mad(X)` функция предназначена для расчета среднего абсолютного отклонения y выборки X от среднего арифметического значения. Выборка X может быть представлена в виде вектора или матрицы. Если X задана как вектор, то среднее абсолютное отклонение значений выборки рассчитывается по всем его элементам. Для выборки определенной в виде матрицы среднее абсолютное отклонение рассчитывается для каждого столбца X .

Среднее абсолютное отклонение является менее эффективной оценкой разброса значений выборки из нормально распределенной генеральной совокупности по сравнению с точечной оценкой среднего квадратического отклонения.

Для нормально распределенной генеральной совокупности точечная оценка среднего квадратического отклонения равна произведению среднего абсолютного отклонения на 1,3.

Примеры использования функции расчета среднего абсолютного отклонения значений выборки

Расчет среднего абсолютного отклонения выборки X заданной как вектор

```
>> X = normrnd(10,1,100,1);
```

```
>> y = mad(X)
```

```
y =
```

```
0.7325
```

Расчет среднего абсолютного отклонения выборки X , заданной в виде матрицы

```
>> X = normrnd(10,1,100,5);
```

```
>> y = mad(X)
```

```
y =
```

```
0.7628 0.9216 0.8363 0.7577 0.8859
```

Определение эффективности точечной оценки среднего квадратического отклонения, рассчитанной как произведение среднего

абсолютного отклонения на 1,3, для 100 нормально распределенных выборок.

```
>> X = normrnd(0,1,100,100);  
>> s = std(X);  
>> s_MAD = 1.3 * mad(X);  
>> efficiency = (norm(s - 1)./norm(s_MAD - 1)).^2  
efficiency =  
    0.6730
```

moment

Расчет центрального момента выборки произвольного порядка

Синтаксис

```
m = moment(X,k)
```

Описание

`m = moment(X,k)` функция предназначена для расчета центрального момента m порядка k выборки X . Значение порядка момента k должно быть целым положительным числом. Если X задана как вектор, то центральный момент m рассчитывается по всем его элементам. Для выборки определенной в виде матрицы центральный момент m рассчитывается для каждого столбца X .

Первый центральный момент равен нулю. Второй центральный момент является точечной оценкой дисперсии и определяется по формуле

$$D_j = \frac{1}{n} \sum_{i=1}^n (X_{ij} - M[X]_j)^2 ,$$

где $M[X]_j$ - математическое ожидание выборки, n - объем выборки (число элементов в векторе, или строк в матрице X).

Центральный момент k -го порядка рассчитывается по формуле

$$m_j = E(X_j - \bar{X}_j)^k ,$$

где \bar{X}_j - среднее арифметическое значение выборки X_j , $E(t)$ - наиболее вероятная оценка параметра t .

Примеры использования функции расчета центрального момента произвольного порядка выборки

Расчет второго центрального момента (точечной оценки дисперсии) выборки X , заданной в виде вектора

```
>> X = normrnd(10,1,100,1);
```

```
>> m = moment(X,2)
```

```
m =
```

```
0.7940
```

Расчет второго центрального момента (точечной оценки дисперсии) выборки X , заданной в виде матрицы

```
>> X = normrnd(10,1,100,5);  
>> m = moment(X,2)  
m =  
    0.7615    0.7820    0.9083    0.8813    1.1147
```

skewness

Расчет точечной оценки коэффициента асимметрии выборки

Синтаксис

```
y = skewness(X)  
y = skewness(X, flag)
```

Описание

`y = skewness(X)` функция предназначена для расчета точечной оценки коэффициента асимметрии y выборки X . Если X задана как вектор, то точечная оценка коэффициента асимметрии рассчитывается по всем его элементам. Для выборки определенной в виде матрицы точечная оценка коэффициента асимметрии рассчитывается для каждого столбца X .

Расчет проводится по формуле

$$y_j = \frac{E(X_j - \bar{X}_j)^3}{\sigma_j^3},$$

где \bar{X}_j - среднее арифметическое значение выборки X , σ_j - точечная оценка среднего квадратического отклонения выборки X , $E(t)$ - наиболее вероятная оценка параметра t .

Коэффициент асимметрии выборки является мерой смещенности распределения относительно среднего арифметического значения.

Отрицательный коэффициент асимметрии соответствует распределению смещенному влево относительно среднего значения. Положительный коэффициент асимметрии соответствует распределению смещенному вправо относительно среднего значения. Для нормального закона, или любого другого симметричного распределения, коэффициент асимметрии равен нулю.

`y = skewness(X, flag)` функция позволяет рассчитать несмещенную (`flag=0`) и смещенную (`flag=1`, значение по умолчанию) точечную оценку коэффициента асимметрии y выборки X . Величина смещения выборочного коэффициента асимметрии зависит от объема выборки. Если X является выборкой из генеральной совокупности, для получения несмещенной точечной оценки коэффициента асимметрии `flag` должен быть равен 0.

Примеры использования функции расчета точечной оценки коэффициента асимметрии выборки

Расчет точечной оценки коэффициента асимметрии выборки X, заданной в виде вектора

```
>> X = normrnd(10,1,100,1);  
>> y = skewness (X)  
y =  
    0.1302
```

Расчет точечной оценки коэффициента асимметрии выборки X заданной в виде матрицы

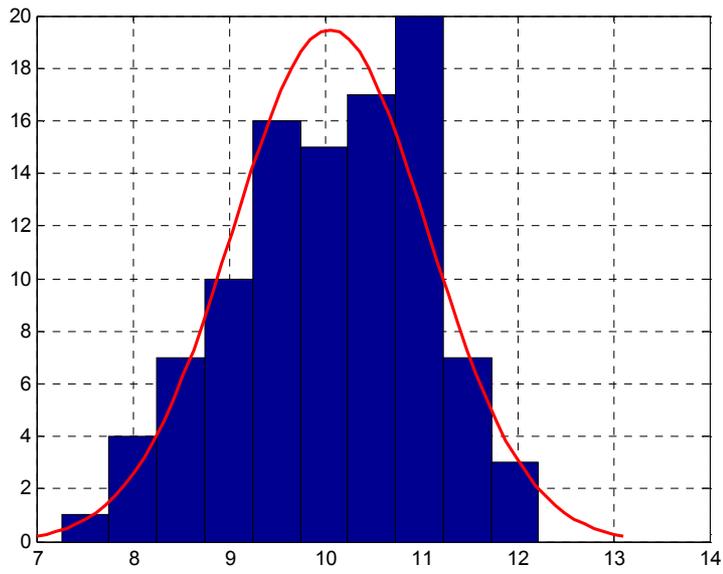
```
>> X = normrnd(10,1,100,5);  
>> y = skewness (X)  
y =  
    0.2924 -0.3086  0.0883  0.0849  0.2744
```

Расчет смещенной и несмещенной точечных оценок коэффициента асимметрии выборки X

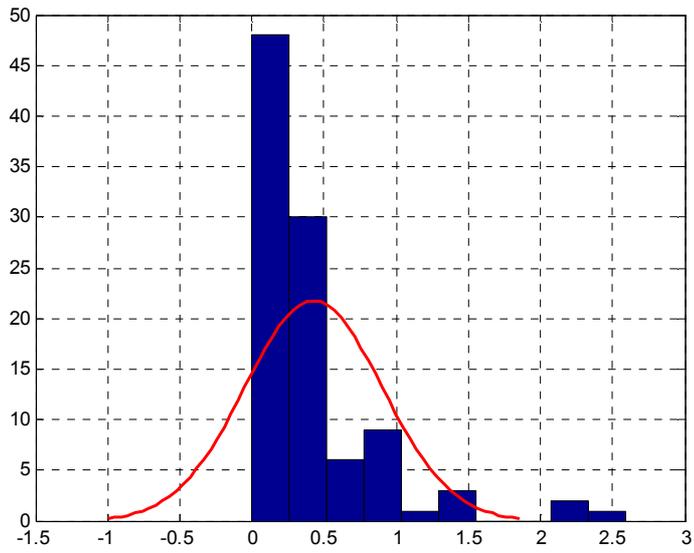
```
>> X = normrnd(10,1,100,1);  
>> y = skewness (X,1)  
y =  
    0.6063  
>> y = skewness (X,0)  
y =  
    0.6156
```

Сравнение распределения выборок и значений коэффициента асимметрии для нормального закона и закона Вейбулла

```
>> X1 = normrnd(10,1,100,1);  
>> X2 = weibrnd(2,1,100,1);  
>> X = [X1 X2];  
>> y = skewness (X)  
y =  
   -0.3116  2.5254  
>> histfit(X1)  
>> grid on
```



```
>> histfit(X2)  
>> grid on
```



iqr

Расчет интерквартильного размаха выборки

Синтаксис

```
y = iqr(X)
```

Описание

$y = \text{iqr}(X)$ функция предназначена для расчета интерквартильного размаха y выборки X . Интерквартильный размах представляет разницу между 75% и 25% процентилями выборки. Интерквартильный размах является робастной оценкой разброса значений выборки.

Если X задана как вектор, то интерквартильный размах рассчитывается по всем его элементам. Для выборки определенной в виде матрицы интерквартильный размах рассчитывается для каждого столбца X .

Интерквартильный размах является более репрезентативной оценкой разброса значений выборки по сравнению с точечной оценкой среднего квадратического отклонения, но менее эффективен при оценке разброса нормально распределенных данных.

Точечная оценка среднего квадратического отклонения для нормально распределенной генеральной совокупности может быть получена как произведение интерквартильного размаха на 0,7413.

Примеры использования функции расчета интерквартильного размаха выборки

Расчет интерквартильного размаха выборки X , заданной в виде вектора

```
>> X = normrnd(10,1,100,1);
```

```
>> y = iqr(X)
```

```
y =
```

```
1.3242
```

Расчет интерквартильного размаха выборки X , заданной в виде матрицы

```
>> X = normrnd(10,1,100,5);
```

```
>> y = iqr(X)
```

```
y =
```

```
1.4223 1.4877 1.3366 0.9497 1.7574
```

Определение эффективности точечной оценки среднего квадратического отклонения, рассчитанной как произведение интерквартильного размаха выборки на 0,7413, для 100 нормально распределенных выборок.

```
>> X = normrnd(0,1,100,100);  
>> s = std(X);  
>> s_IQR = 0.7413 * iqr(X);  
>> efficiency = (norm(s - 1)./norm(s_IQR - 1)).^2  
efficiency =  
    0.5409
```

nanmax

Поиск максимального значения в выборке содержащей нечисловые элементы

Синтаксис

```
m = nanmax(X)
[m,ndx] = nanmax(X)
m = nanmax(X,Y)
```

Описание

`m = nanmax(X)` функция предназначена для поиска максимального значения `m` в выборке `X` содержащей значения элементов равные NaN. Нечисловые значения элементов матрицы `X` NaN рассматриваются как пропущенные. Если `X` задана как вектор, то поиск максимального значения производится по всем его элементам. Для выборки определенной в виде матрицы поиск максимального значения выполняется для каждого столбца `X`.

`[m,ndx] = nanmax(X)` функция поиска максимального значения `m` и номеров максимальных значений `ndx` в выборке `X`, содержащей значения элементов равные NaN. Номера максимальных значений `ndx` представляются в виде вектора.

`m = nanmax(X,Y)` функция выполняет поиск максимальных элементов `m` матриц `X` и `Y` с игнорированием элементов со значениями равными NaN. Сравнение соответствующих элементов матриц `X`, `Y` выполняется попарно. Размерности `X` и `Y` должны совпадать. Результатом является матрица `m` с размерностью матриц `X` и `Y`.

Примеры использования функции поиска максимального значения в выборке содержащей нечисловые элементы

Поиск максимального значения выборки `X`, заданной в виде вектора

```
>> X = normrnd(10,1,100,1);
>> X([1 10 50 66 54]) = [NaN NaN NaN NaN NaN];
>> m = nanmax(X)
m =
    12.1832
```

Поиск максимальных значений выборок в матрице `X`

```
>> X = normrnd(10,1,100,5);
>> X([1 120 250 366 454]) = [NaN NaN NaN NaN NaN];
>> m = nanmax(X)
```

```
m =  
12.1122 12.3093 12.3726 11.8705 12.6903
```

Поиск максимальных значений выборок и их индексов в матрице X

```
>> X = normrnd(10,1,100,5);  
>> X([1 120 250 366 454]) = [NaN NaN NaN NaN NaN];  
>> [m,ndx] = nanmax(X)
```

```
m =  
12.1764 12.7316 12.4953 11.7621 12.9495  
ndx =  
67 56 36 52 77
```

Поиск максимальных элементов матриц X и Y с игнорированием элементов равных NaN

```
>> X = normrnd(10,1,5,5);  
>> Y = normrnd(10,1,5,5);  
>> X([1 7 12 15 25]) = [NaN NaN NaN NaN NaN];  
>> Y([2 6 11 16 23]) = [NaN NaN NaN NaN NaN];  
>> m = nanmax(X,Y)
```

```
m =  
7.0228 11.6830 12.2472 11.4445 9.5813  
11.6275 10.5553 10.0032 9.9256 11.4149  
10.8844 10.7595 11.9916 10.2624 9.1526  
9.5941 10.0967 11.6241 10.1192 10.1661  
9.8473 11.1580 10.1396 10.3323 10.9260
```

nanmin

Поиск минимального значения в выборке содержащей нечисловые элементы

Синтаксис

```
m = nanmin(X)
[m,ndx] = nanmin(a)
m = nanmin(a,b)
```

Описание

`m = nanmin(X)` функция предназначена для поиска минимального значения `m` в выборке `X` содержащей значения элементов равные NaN. Нечисловые значения элементов матрицы `X` NaN рассматриваются как пропущенные. Если `X` задана как вектор, то поиск минимального значения производится по всем его элементам. Для выборки определенной в виде матрицы поиск минимального значения выполняется для каждого столбца `X`.

`[m,ndx] = nanmin(X)` функция поиска минимального значения `m` и номеров минимальных значений `ndx` в выборке `X`, содержащей значения элементов равные NaN. Номера минимальных значений `ndx` представляются в виде вектора.

`m = nanmin(X,Y)` функция выполняет поиск минимальных элементов `m` матриц `X` и `Y` с игнорированием элементов со значениями равными NaN. Сравнение соответствующих элементов матриц `X`, `Y` выполняется попарно. Размерности `X` и `Y` должны совпадать. Результатом является матрица `m` с размерностью матриц `X` и `Y`.

Примеры использования функции поиска минимального значения в выборке содержащей нечисловые элементы

Поиск минимального значения выборки `X`, заданной в виде вектора

```
>> X = normrnd(10,1,100,1);
>> X([1 10 50 66 54]) = [NaN NaN NaN NaN NaN];
>> m = nanmin(X)
m =
    7.9151
```

Поиск минимальных значений выборок в матрице `X`

```
>> X = normrnd(10,1,100,5);
>> X([1 120 250 366 454]) = [NaN NaN NaN NaN NaN];
>> m = nanmin(X)
```

```
m =  
8.3019 7.6298 6.4973 6.7146 7.3728
```

Поиск минимальных значений выборок и их индексов в матрице X

```
>> X = normrnd(10,1,100,5);  
>> X([1 120 250 366 454]) = [NaN NaN NaN NaN NaN];  
>> [m,ndx] = nanmin(X)
```

```
m =  
7.6796 7.8707 7.9710 8.0647 7.3797  
ndx =  
71 75 29 19 37
```

Поиск минимальных элементов матриц X и Y с игнорированием нечисловых элементов

```
>> X = normrnd(10,1,5,5);  
>> Y = normrnd(10,1,5,5);  
>> X([1 7 12 15 25]) = [NaN NaN NaN NaN NaN];  
>> Y([2 6 11 16 23]) = [NaN NaN NaN NaN NaN];  
>> m = nanmin(X,Y)
```

```
m =  
11.0258 11.2535 10.3844 12.2433 10.3521  
8.1952 11.0665 9.3263 9.8507 9.9351  
10.5221 6.9539 10.0804 9.1112 9.9209  
9.7920 9.7282 11.2769 8.2508 9.8360  
9.7798 9.7111 10.5591 7.8597 9.3344
```

nansum

Расчет суммы элементов выборки содержащей нечисловые элементы

Синтаксис

```
y = nansum(X)
```

Описание

`y = nansum(X)` функция предназначена для расчета суммы `y` элементов выборки `X` содержащей значения элементов равные NaN. Нечисловые значения элементов матрицы `X` NaN рассматриваются как пропущенные. Если `X` задана как вектор, то расчет суммы производится по всем его элементам. Для выборки определенной в виде матрицы расчет суммы элементов выполняется для каждого столбца `X`.

Примеры использования функции расчета суммы элементов выборки с игнорированием нечисловых элементов

Расчет суммы элементов выборки заданной как вектор

```
>> X = normrnd(10,1,100,1);  
>> X([1 10 50 66 54]) = [NaN NaN NaN NaN NaN];  
>> y = nansum (X)  
y =  
952.5220
```

Расчет суммы элементов выборки заданной как матрица

```
>> X = normrnd(10,1,100,5);  
>> X([1 120 250 366 454]) = [NaN NaN NaN NaN NaN];  
>> y = nansum (X)  
y =  
978.4871 982.8880 987.7201 977.3189 983.8354
```

nanmean

Расчет среднего арифметического значения выборки содержащей нечисловые элементы

Синтаксис

```
y = nanmean(X)
```

Описание

`y = nanmean(X)` функция предназначена для расчета среднего арифметического `y` выборки `X` содержащей значения элементов равные NaN. Нечисловые значения элементов матрицы `X` NaN рассматриваются как пропущенные. Если `X` задана как вектор, то расчет среднего арифметического производится по всем его элементам. Для выборки определенной в виде матрицы расчет среднего арифметического выполняется для каждого столбца `X`.

Примеры использования функции расчета среднего арифметического выборки содержащей нечисловые элементы

Расчет среднего арифметического выборки, заданной как вектор

```
>> X = normrnd(10,1,100,1);  
>> X([1 10 50 66 54]) = [NaN NaN NaN NaN NaN];  
>> y = nanmean(X)  
y =  
10.0669
```

Расчет средних арифметических значений выборок, заданных в виде матрицы

```
>> X = normrnd(10,1,100,5);  
>> X([1 120 250 366 454]) = [NaN NaN NaN NaN NaN];  
>> y = nanmean(X)  
y =  
9.9490 10.1019 9.8540 9.9973 9.8967
```

nanmedian

Расчет медианы выборки содержащей нечисловые элементы

Синтаксис

```
y = nanmedian(X)
```

Описание

`y = nanmedian(X)` функция предназначена для расчета медианы `y` выборки `X`, содержащей значения элементов равные NaN. Нечисловые значения элементов матрицы `X` NaN рассматриваются как пропущенные. Если `X` задана как вектор, то расчет медианы производится по всем его элементам. Для выборки определенной в виде матрицы расчет медианы выполняется для каждого столбца `X`.

Примеры использования функции расчета медианы выборки с нечисловыми элементами

Расчет медианы выборки, заданной как вектор

```
>> X = normrnd(10,1,100,1);  
>> X([1 10 50 66 54]) = [NaN NaN NaN NaN NaN];  
>> y = nanmedian (X)  
y =  
    9.8785
```

Расчет медиан выборок, заданных в виде матрицы

```
>> X = normrnd(10,1,100,5);  
>> X([1 120 250 366 454]) = [NaN NaN NaN NaN NaN];  
>> y = nanmedian (X)  
y =  
    10.2096    10.1491    10.2606    10.1257    9.9874
```

nanstd

Расчет точечной оценки среднего квадратического отклонения выборки содержащей нечисловые элементы

Синтаксис

```
y = nanstd(X)
```

Описание

`y = nanstd(X)` функция предназначена для расчета точечной оценки среднего квадратического отклонения y выборки X , содержащей значения элементов равные NaN. Нечисловые значения элементов матрицы X NaN рассматриваются как пропущенные. Если X задана как вектор, то расчет точечной оценки среднего квадратического отклонения производится по всем его элементам. Для выборки определенной в виде матрицы расчет точечной оценки среднего квадратического отклонения выполняется для каждого столбца X .

Примеры использования функции расчета точечной оценки среднего квадратического отклонения выборки с нечисловыми элементами

Расчет точечной оценки среднего квадратического отклонения выборки, заданной как вектор

```
>> X = normrnd(10,1,100,1);  
>> X([1 10 50 66 54]) = [NaN NaN NaN NaN NaN];  
>> y = nanstd(X)  
y =  
    0.9424
```

Расчет точечных оценок среднего квадратического отклонения выборок, заданных в виде матрицы

```
>> X = normrnd(10,1,100,5);  
>> X([1 120 250 366 454]) = [NaN NaN NaN NaN NaN];  
>> y = nanstd(X)  
y =  
    1.0444    1.0968    1.0272    0.9683    1.1566
```

grpstats

Расчет точечной и интервальной оценок математического ожидания категоризованной переменной

Синтаксис

```
means = grpstats(X,group)
[means,sem,counts,name] = grpstats(X,group)
[means,sem,counts,name] = grpstats(x,group,alpha)
```

Описание

`means = grpstats(X,group)` функция предназначена для расчета среднего арифметического значения `means` категоризованной переменной `X`. Среднее арифметическое значение рассчитывается для каждой категории. Деление на категории выполняется при помощи входного аргумента `group`. Выборка негруппированных значений `X` может быть задана как вектор или матрица. Значения вектора или матрицы `X` принадлежат к одной категории, если равны соответствующие значения переменной `group`. Входной аргумент `group` может быть представлен как вектор, массив строк или массив ячеек строковых переменных. Также `group` может быть массивом ячеек, содержащим несколько сгруппированных переменных, например `{G1 G2 G3}`. В последнем случае наблюдения принадлежат к одной группе, если равны между собой все сгруппированные переменные. Размерность векторов `X` и `group` должна совпадать. Если выборка `X` задана как матрица, категоризирующая переменная `group` должна быть представлена как вектор. Количество строк матрицы `X` и элементов вектора `group` должно быть одинаковым.

`[means,sem,counts,name] = grpstats(x,group)` функция позволяет рассчитать точечную `means` и интервальную `sem` оценки математического ожидания категоризованной переменной `X`, количество элементов в каждой категории `counts`, и отобразить список названий категорий `name`.

Переменная `name` полезна для установления соответствия между полученными результатами расчета `means`, `sem` и заданными категориями, если элементы `group` не являются целыми числами.

`[means,sem,counts,name] = grpstats(x,group,alpha)` этот вариант синтаксиса функции кроме результатов расчета значений `means`, `sem`, `counts` и вывода названий `name` строит график средних арифметических значений и их $100 \cdot (1 - \alpha)\%$ доверительных интервалов по категориям в порядке возрастания последних.

Примеры использования функции расчета точечной и интервальной оценок математического ожидания категоризованной переменной

Расчет точечной оценки математического ожидания для категоризованной выборки заданной как вектор. Выборка x из 100 наблюдений делится на 4 категории. Категории кодируются целыми числами в диапазоне от 1 до 4.

```
>> group = unidrnd(4,100,1);
>> x = normrnd(0,1,100,1);
>> means = grpstats(x,group)
means =
    0.0369
    0.0317
    0.0759
    0.0454
```

Расчет точечной оценки математического ожидания для категоризованной выборки заданной в виде матрицы. Переменная X задается как матрица с размерностью 100×5 и представляет собой 5 выборок по 100 элементов в каждой. Математическое ожидание при генерации выборок изменяется от 1 до 5. Значения в каждой выборке делятся на 4 группы. Категоризирующая переменная `group` задается как вектор целых чисел с размерностью 100×1 .

```
>> group = unidrnd(4,100,1);
>> true_mean = 1:5;
>> true_mean = true_mean(ones(100,1),:);
>> x = normrnd(true_mean,1);
>> means = grpstats(x,group)
means =
    1.0369    1.5801    2.8850    4.3252    5.1890
    1.0317    1.9425    3.0184    3.8486    4.9424
    1.0759    1.9433    2.8528    4.0344    4.6057
    1.0454    1.9528    2.9190    3.8409    4.7507
```

Расчет точечных `means` и интервальных `sem` оценок математического ожидания категоризованной переменной X , количества элементов в каждой категории `counts`, и отображения списка названий категорий `name`. Переменная X задается как матрица с размерностью 100×5 . Категории кодируются вектором целых чисел на 4 группы.

```
>> group = unidrnd(4,100,1);
>> x = normrnd(0,1,100,5);
>> [means,sem,counts,name] = grpstats(x,group)
means =
```

```

-0.2741  0.0263 -0.1199 -0.1949  0.0658
 0.0205  0.1609 -0.0889  0.1441 -0.1134
-0.2905  0.0505 -0.0386  0.1086 -0.2904
 0.5349 -0.0313  0.0819  0.5119 -0.3288
sem =
 0.2056  0.1608  0.1764  0.1989  0.1852
 0.1506  0.1960  0.2535  0.2000  0.2050
 0.1678  0.1209  0.1544  0.1434  0.1537
 0.2332  0.2603  0.2468  0.1457  0.1845
counts =
 29  29  29  29  29
 25  25  25  25  25
 29  29  29  29  29
 17  17  17  17  17
name =
 '1'
 '2'
 '3'
 '4'

```

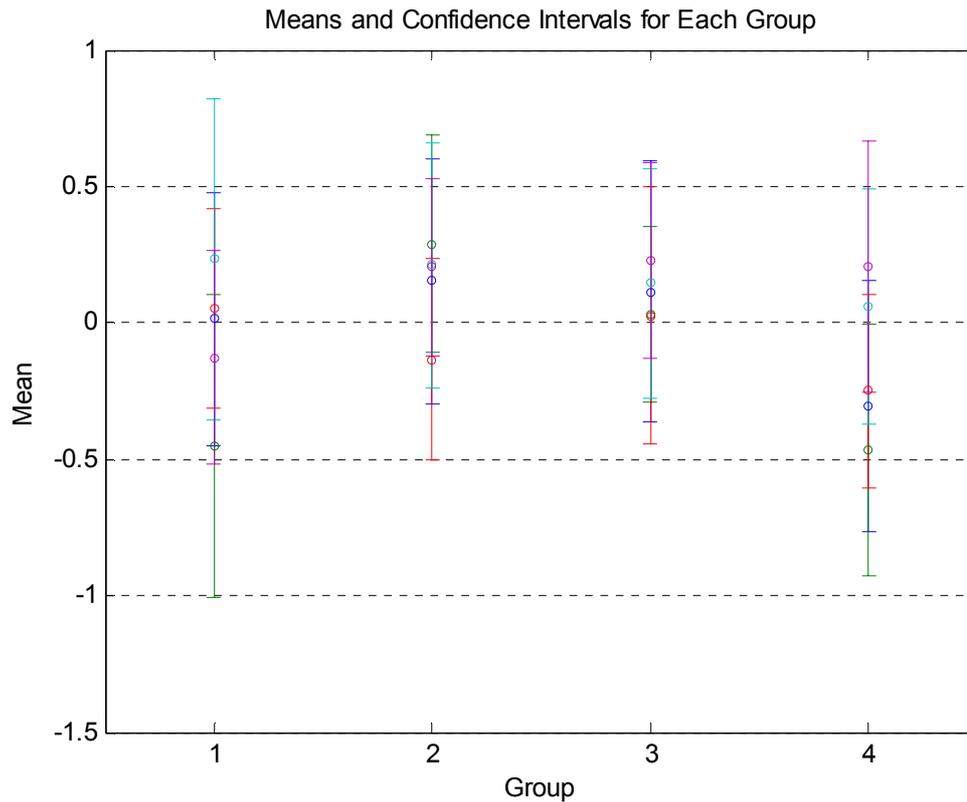
Расчет значений means, sem, counts, вывод name и графика средних арифметических значений с 95% доверительными интервалами по соответствующим категориям.

```

>> group = unidrnd(4,100,1);
>> x = normrnd(0,1,100,5);
>> [means,sem,counts,name] = grpstats(x,group,0.05)
means =
 0.0135 -0.4496  0.0543  0.2349 -0.1285
 0.1529  0.2906 -0.1331  0.2133  0.2041
 0.1151  0.0340  0.0273  0.1469  0.2279
-0.3031 -0.4670 -0.2492  0.0589  0.2070
sem =
 0.2215  0.2651  0.1741  0.2816  0.1873
 0.2205  0.1940  0.1795  0.2196  0.1597
 0.2330  0.1574  0.2300  0.2046  0.1756
 0.2227  0.2238  0.1720  0.2088  0.2229
counts =
 19  19  19  19  19
 29  29  29  29  29
 28  28  28  28  28
 24  24  24  24  24
name =
 '1'
 '2'
 '3'

```

'4'



Расчет значений means, sem, counts и вывод списка категорий name. Переменная X задается как матрица с размерностью 20x5. Категории определяются строковым массивом и выборка делится на 4 группы.

```
>> group ={'A' 'B' 'C' 'D' 'A' 'B' 'A' 'B' 'C' 'D' 'C' 'D' 'A' 'A' 'B' 'C' 'D' 'B' 'C' 'D'}
```

```
group =
```

```
'A'  
'B'  
'C'  
'D'  
'A'  
'B'  
'A'  
'B'  
'C'  
'D'  
'C'  
'D'  
'A'  
'A'  
'B'  
'C'  
'D'
```

```

'B'
'C'
'D'
>> x = normrnd(0,1,20,5)
x =
-0.2463 -0.2979 -0.5299  0.8683  0.7193
-0.1457  1.1543  0.5411 -0.8048 -0.2831
-1.1690  1.0461  0.6817 -0.7527 -1.4250
-0.0220  2.1269  0.5386 -0.7458  0.4615
 0.6183 -0.6558 -0.5100 -0.3097  1.0915
 1.8659 -1.1424 -1.3221 -1.5219 -1.0443
 0.0819  0.9490 -0.6107  0.8265 -2.8428
 1.6080 -0.4046 -0.5653 -0.6130  0.9968
-0.3807 -0.3843  0.0862  0.9597  0.0765
-1.2996  0.4820  0.6915  1.9730 -1.8667
-0.7240  0.4438  2.1338  0.2950 -0.6136
-0.5650  0.3811 -0.0029 -0.3927  1.1694
 0.6217  1.1023 -0.0895  0.5759 -0.5750
-1.3355  0.8564 -0.2550 -1.1414 -0.2648
-0.1231 -1.1785 -0.8742  0.0611  0.0047
-1.1028  0.4020  0.4229  0.0123 -0.0394
-2.7532 -0.5842 -0.1334 -0.1681 -0.5054
 0.2520 -0.9795  0.5396 -0.6873 -1.1578
-0.8581  0.1151  0.8752 -0.9907  0.7104
 1.1354  0.0685 -1.2508 -0.0498  0.7282
>> [means,sem,counts,name] = grpstats(x,group)
means =
 0.3502 -0.8565 -0.5624  0.2891 -0.0456
 0.2269 -0.1071 -0.5212 -0.7214  0.4791
-1.2206  0.6396  0.3454 -0.0956 -0.2041
-0.2389 -0.2455  0.0123  0.4581  0.4169
sem =
 0.6679  0.3892  0.6913  0.7837  0.3769
 0.2133  0.4039  0.5240  0.3131  0.3355
 0.5199  0.3487  0.6232  0.2142  0.2502
 0.4466  0.2645  0.4143  0.4821  0.3391
counts =
 5  5  5  5  5
 5  5  5  5  5
 5  5  5  5  5
 5  5  5  5  5
name =
'A'
'B'
'C'

```

'D'

crosstab

Кросс-табуляция значений нескольких векторов

Синтаксис

```
table = crosstab(col1,col2)
table = crosstab(col1,col2,col3,...)
[table,chi2,p] = crosstab(col1,col2)
[table,chi2,p,label] = crosstab(col1,col2)
```

Описание

`table = crosstab(col1,col2)` функция выполняет расчет частот повторяемости `table` пар целых положительных значений векторов `col1`, `col2`. Результат расчета выводится в виде матрицы частот `table`. Размерность матрицы равна $m \times n$, где m – количество значений элементов в векторе `col1`, n – количество значений элементов в векторе `col2`. Если векторы `col1`, `col2` содержат вещественные значения, массивы символов, строковые массивы ячеек, то в соответствие каждому значению `col1`, `col2` ставится целое положительное число и выполняется кросс-табуляция по этим числам.

`table = crosstab(col1,col2,col3,...)` функция возвращает n -мерный массив частот `table` сочетаний значений векторов `col1`, `col2`, `col3`,..., где n – количество векторов в списке входных переменных. Значение массива `table(i,j,k,...)` соответствует частоте повторяемости сочетаний значений `col1(i)`, `col2(j)`, `col3(k)`.

`[table,chi2,p] = crosstab(col1,col2)` функция выполняет расчет частот повторяемости `table` пар значений векторов `col1`, `col2`, значения статистики χ^2 `chi2`, уровня значимости `p`. Значение статистики `chi2` используется для проверки статистической гипотезы о независимости строк и столбцов матрицы частот `table`. Значение `p` является уровнем значимости при проверке указанной статистической гипотезы. Значение уровня значимости `p` близкое к нулю позволяет принять гипотезу о независимости строк и рядов матрицы частот `table`.

`[table,chi2,p,label] = crosstab(col1,col2)` функция кроме частот повторяемости `table`, значения статистики χ^2 `chi2`, уровня значимости `p`, возвращает массив ячеек `label`, содержащий значения входных аргументов, распределенных последовательно по столбцам матрицы. Значение `label(i,j)` является элементом вектора `colj` определяющим i -ю группу в j -м измерении.

Примеры использования функции кросс-табуляции значений нескольких векторов

Расчет частот повторяемости пар целых положительных значений двух векторов. Значения вектора r1 изменяются в диапазоне 1 ÷ 3, вектора r2 ñ в диапазоне 1 ÷ 2.

```
>> r1 = unidrnd(3,50,1);  
>> r2 = unidrnd(2,50,1);  
>> table = crosstab(r1,r2)  
table =  
    7    5  
   15    8  
    9    6
```

Расчет частот повторяемости сочетаний положительных значений трех векторов. Значения вектора r1 изменяются в диапазоне 1 ÷ 3, вектора r2 ñ в диапазоне 1 ÷ 2, вектора r3 ñ в диапазоне 1...5.

```
>> r1 = unidrnd(3,50,1);  
>> r2 = unidrnd(2,50,1);  
>> r3= unidrnd(5,50,1);  
>> table = crosstab(r1,r2,r3)  
table(:,:,1) =  
    1    1  
    3    2  
    2    2  
table(:,:,2) =  
    0    0  
    5    1  
    1    1  
table(:,:,3) =  
    4    1  
    3    1  
    2    1  
table(:,:,4) =  
    0    1  
    3    1  
    2    1  
table(:,:,5) =  
    2    2  
    1    3  
    2    1
```

Расчет частот повторяемости пар значений двух векторов, значения статистики χ^2 , уровня значимости.

```
>> r1 = unidrnd(3,50,1);
```

```

>> r2 = unidrnd(2,50,1);
>> [table,chi2,p] = crosstab(r1,r2)
table =
     2     5
    10    14
    10     9
chi2 =
    1.3038
p =
    0.5211

```

Расчет частот повторяемости пар значений двух векторов, значения статистики χ^2 , уровня значимости, а также вывод значений элементов группируемых векторов.

```

>> r1 = unidrnd(3,50,1);
>> r2 = unidrnd(2,50,1);
>> [table,chi2,p,label] = crosstab(r1,r2)
table =
     7     6
    10    15
     8     4
chi2 =
    2.4103
p =
    0.2997
label =
    '1'  '1'
    '2'  '2'
    '3'  []

```

Пример использования строковых переменных при кросс-табуляции.

```

>> r1 = ['a b c a b c a b c b c b c'];
>> r2 = ['a b a b a b a b b b a a'];
>> [table,chi2,p,label] = crosstab(r1,r2)
table =
     2     1
     2     3
     2     3
chi2 =
    0.6603
p =
    0.7188
label =

```

'a' 'a'
'b' 'b'
'c' []

tabulate

Расчет частот элементов вектора

Синтаксис

```
table = tabulate(x)
tabulate(x)
```

Описание

`table = tabulate(x)` функция позволяет получить матрицу `table` частот элементов вектора `x`. Элементы вектора `x` должны быть целыми положительными числами. Матрица `table` содержит три столбца. В первом столбце содержатся значения элементов вектора `x`, во втором `n` частота значений, в третьем `n` относительная частота значений, выраженная в процентах от объема выборки.

`tabulate(x)` вариант синтаксиса функции без выходных аргументов выводит матрицу частот в виде отформатированной таблицы.

Примеры использования функции расчета частот элементов вектора

Расчет матрицы частот элементов вектора

```
>> x = unidrnd(5,50,1);
>> table = tabulate(x)
table =
```

```
 1  14  28
 2  12  24
 3   6  12
 4   6  12
 5  12  24
```

Расчет матрицы частот элементов вектора и вывод форматированной таблицы

```
>> x = unidrnd(5,50,1);
>> tabulate(x)
Value  Count  Percent
```

```
 1     14   28.00%
 2     12   24.00%
 3      6   12.00%
 4      6   12.00%
 5     12   24.00%
```

trimmean

Расчет среднего арифметического значения выборки с исключением заданного процента минимальных и максимальных элементов в выборке

Синтаксис

```
m = trimmean(X,percent)
```

Описание

`m = trimmean(X,percent)` функция предназначена для расчета среднего арифметического m выборки X с исключением заданного процента наблюдений `percent` в выборке. Исключение выполняется для минимальных и максимальных значений с долей наблюдений равной `percent/2`. Полученная оценка математического ожидания является робастной. Если данные принадлежат к одному распределению, то среднее арифметическое полученное с игнорированием заданного процента наблюдений менее эффективно по сравнению со средним арифметическим.

Если X задана как вектор, то среднее арифметическое значение рассчитывается по всем его элементам. Для выборки определенной в виде матрицы среднее арифметическое значение рассчитывается для каждого столбца X .

Для X заданной как матрица (набор выборок) можно определить вектор `percent` для каждой выборки отдельно. В этом случае число элементов в векторе `percent` должно быть равно числу столбцов X .

Примеры использования функции расчета среднего арифметического значения выборки с исключением заданного процента минимальных и максимальных элементов в выборке

Расчет среднего арифметического значения выборки с игнорированием 5% минимальных и 5% максимальных значений выборки заданной как вектор

```
>> X=normrnd(0,1,100,1);  
>> percent = 10  
percent =  
    10  
>> m = trimmean(X,percent)  
m =  
    0.0594
```

Расчет среднего арифметического значения выборки с игнорированием 5% минимальных и 5% максимальных значений выборки заданной как матрица

```
>> X=normrnd(0,1,100,5);  
>> percent = 10  
percent =  
    10  
>> m = trimmean(X,percent)  
m =  
    0.0582 -0.0580  0.1150 -0.1523  0.0109
```

Расчет средних арифметических значений 5 выборок с игнорированием 5%, 10%, 7%, 12%, 20% минимальных и максимальных значений для каждой выборки

```
>> X=normrnd(0,1,100,5);  
>> percent = [5 10 7 12 20]  
percent =  
     5  10   7  12  20  
>> m = trimmean(X,percent)  
m =  
 -0.1101 -0.0970  0.1581  0.1432  0.1737
```

Оценка эффективности среднего арифметического с 10% исключением максимальных и минимальных значений по сравнению со средним арифметическим, рассчитанных без исключения элементов выборки.

```
>> x = normrnd(0,1,100,100);  
>> m = mean(x);  
>> trim = trimmean(x,10);  
>> sm = std(m);  
>> strim = std(trim);  
>> efficiency = (sm/strim).^2  
efficiency =  
    0.9684
```

prctile

Расчет процентилей выборки

Синтаксис

```
Y = prctile(X,p)
```

Описание

`Y = prctile(X,p)` функция предназначена для расчета процентилей Y выборки X , соответствующих вероятности попадания случайной величины в интервал $(-\infty, Y]$ с вероятностью p . Значение вероятности p должно находиться в интервале от 0 до 100%.

Если X задана как вектор, то процентиль Y рассчитывается по всем его элементам для вероятности p . Вероятность p может быть задана как скаляр или вектор. Значения процентиля рассчитываются для каждого элемента вектора p .

Для выборки определенной в виде матрицы процентиль рассчитывается для каждого столбца X . Для матрицы X и вектора p процентиль рассчитывается для всех значений p и для каждого столбца. Размерность матрицы Y равна $m \times n$, где m — число элементов вектора p , n — число столбцов матрицы X .

Примеры использования функции расчета процентилей выборки

Расчет 10% процентиля вектора X

```
>> X = normrnd(0,1,100,1);  
>> p=10;  
>> Y = prctile(X,p)  
Y =  
-1.1015
```

Расчет 10%, 20%, 30% процентилей вектора X

```
>> X = normrnd(0,1,100,1);  
>> p=[10 20 30];  
>> Y = prctile(X,p)  
Y =  
-1.2258 -1.0402 -0.6829
```

Расчет 10% процентилей для нескольких выборок матрицы X

```
>> X = normrnd(0,1,100,5);  
>> p=10;  
>> Y = prctile(X,p)  
Y =
```

-1.2635 -1.1216 -1.3599 -1.3920 -1.0285

Расчет 10%, 20%, 30% процентилей для нескольких выборок матрицы X

```
>> X = normrnd(0,1,100,5);
```

```
>> p=[10 20 30];
```

```
>> Y = prctile(X,p)
```

Y =

-1.4662 -1.2327 -1.4568 -1.5269 -1.4533

-0.9595 -0.7126 -1.0071 -0.9093 -0.9905

-0.5902 -0.2939 -0.6884 -0.5700 -0.6566

Графическое представление 10%, 30%, 50%, 70%, 90% процентилей выборки X

```
>> X = normrnd(0,1,100,1);
```

```
>> p=[10 30 50 70 90];
```

```
>> Y = prctile(X,p)
```

Y =

-1.1015 -0.3344 0.1162 0.5745 1.1421

```
>> hist(X)
```

```
>> grid on
```

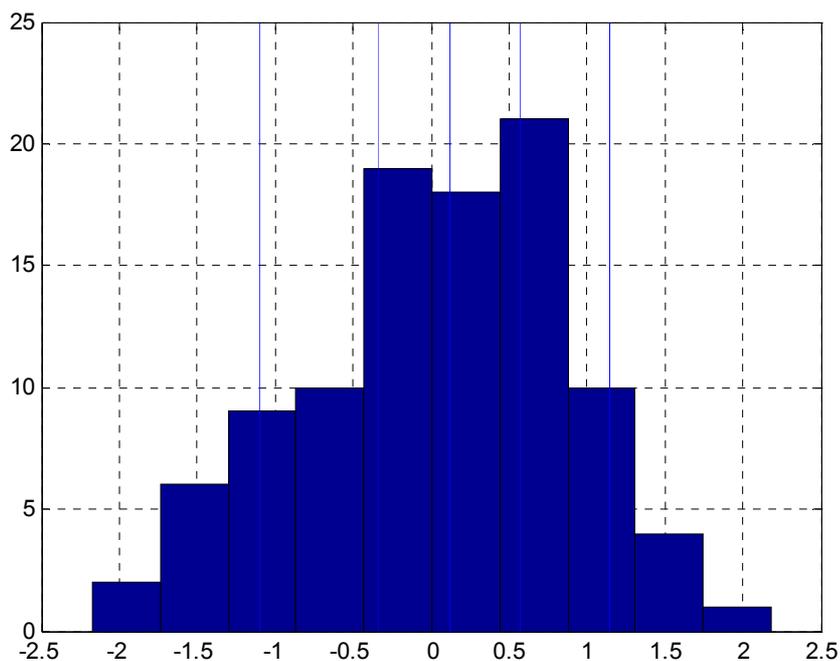
```
>> H=line ([Y(1) Y(1)], [0 25]);
```

```
>> H1=line ([Y(2) Y(2)], [0 25]);
```

```
>> H2=line ([Y(3) Y(3)], [0 25]);
```

```
>> H3=line ([Y(4) Y(4)], [0 25]);
```

```
>> H4=line ([Y(5) Y(5)], [0 25]);
```



bootstrp

Расчет бутстреп-оценок выборки

Синтаксис

```
bootstat = bootstrp(nboot, 'bootfun', d1, d2, ...)  
[bootstat, bootsam] = bootstrp(...)
```

Описание

`bootstat = bootstrp(nboot, 'bootfun', d1, d2, ...)` увеличивает в `nboot` раз размер выборки из исходного набора данных, `d1, d2, ...`, с последующей передачей выборки для анализа в функцию 'bootfun'. Входной аргумент `nboot` должен быть положительным целым числом. Входные данные должны иметь одинаковое число строк `n`. Каждая сформированная бутстреп выборка содержит `n` строк выбранных случайным образом (с замещением) из переданного множества входных данных `d1, d2, ...`

Каждая строка выходной переменной, `bootstat`, содержит результаты расчета функцией 'bootfun' статистик бутстреп выборок. Если функция 'bootfun' возвращает несколько выходных переменных, `bootstat` содержит только первую из них. Если первая возвращаемая функцией 'bootfun' переменная является матрицей, то она преобразуется в вектор-столбец. Полученный вектор присваивается выходной переменной `bootstat`.

`[bootstat, bootsam] = bootstrp(...)` функция возвращает значения статистик `bootstat` и матрицу бутстреп индексов `bootsam`. Каждый из `nboot` столбцов в матрице `bootsam` содержит номера значений, которые были извлечены из входного набора данных для составления соответствующей бутстреп выборки. Например, если `d1, d2, ...` содержат по 16 значений и `nboot=4`, тогда размерность матрицы `bootsam` будет равна 16×4 . первый столбец содержит номера 16 значений извлеченных из `d1, d2, ...` и т.д. для первой из 4 бутстреп выборок, второй столбец содержит индексы для второй бутстреп выборки и т.д. (Бутстреп индексы являются теми же самыми для всех входных наборов данных)

Примеры использования функции расчета бутстреп-оценок выборки

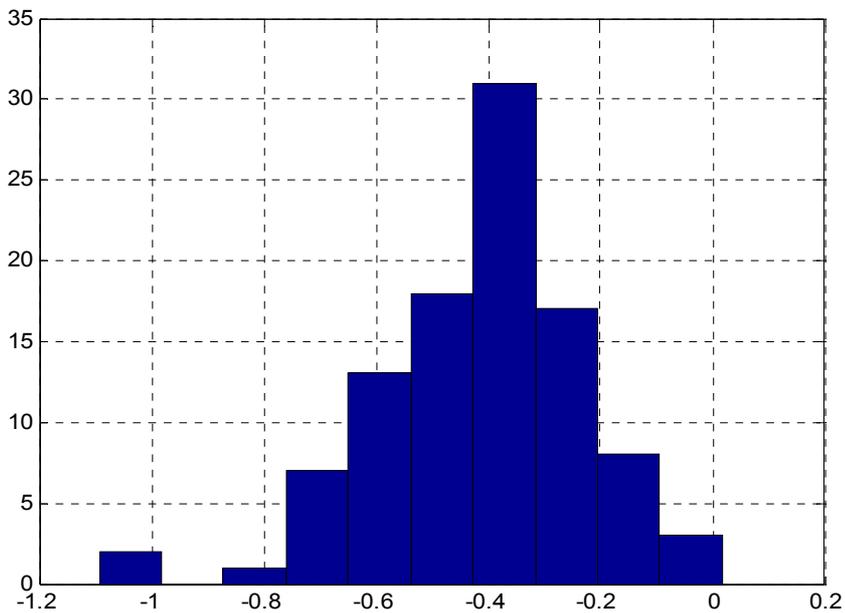
Расчет среднего арифметического значения элементов вектора на 20 элементов. Генерируется 10 бутстреп выборок.

```
>> X = normrnd(0,1,20,1);  
>> [bootstat,bootsam] = bootstrp(10,'mean', X)
```


	18	15	10	15	3	19	15	4	9
1									
16	8	6	7	11	18	17	3	8	10
10	11	10	3	7	9	11	16	20	15

Графическое представление полученных результатов

```
>> X = normrnd(0,1,20,1);
>> [bootstat,bootsam] = bootstrp(100,'mean', X);
>> hist(bootstat)
>> grid on
```



Сравнение результатов расчета среднего арифметического по исходной выборке и средних арифметических значений полученных методом бутстреп оценок

```
>> X = normrnd(0,1,20,1);
>> m = mean(X)
m =
    -0.0826
>> [bootstat,bootsam] = bootstrp(5,'mean', X)
bootstat =
    0.0199
   -0.7018
   -0.1665
   -0.2772
   -0.3740
bootsam =
```

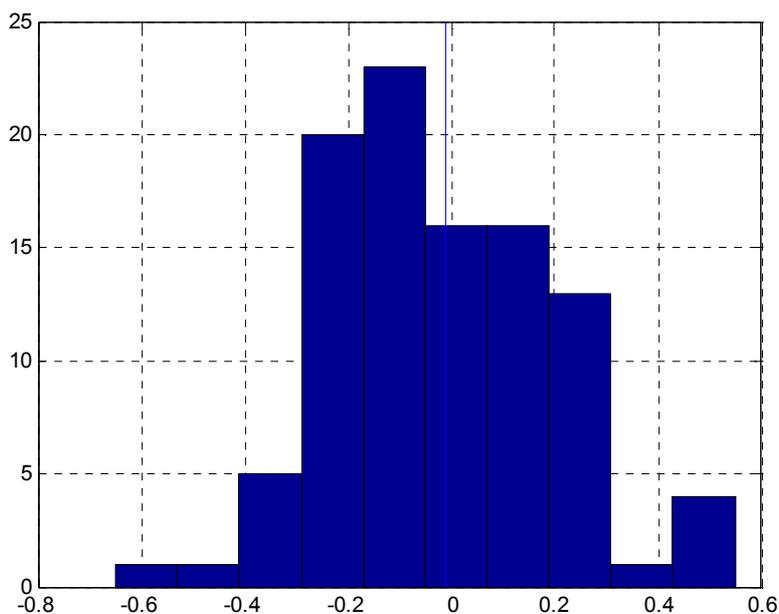
16	15	20	1	2
15	13	18	10	12
20	4	16	11	1
20	18	10	18	1
4	19	3	16	5
12	20	13	9	5
4	19	9	20	4
14	3	6	13	5
9	18	17	3	13
2	2	19	7	15
12	9	1	7	3
9	15	14	3	2
13	1	12	6	12
10	11	1	18	5
5	10	12	11	2
11	4	8	5	17
7	18	2	9	15
16	18	5	6	14
1	11	15	1	4
17	4	9	1	11

Графическое представление результатов расчета среднего арифметического по исходной выборке и средних арифметических значений полученных методом бутстреп оценок

```

>> X = normrnd(0,1,20,1);
>> m = mean(X)
m =
-0.0110
>> [bootstat,bootsam] = bootstrp(100,'mean', X);
>> hist(bootstat)
>> grid on
>> H=line ([m m], [0 25]);

```



Расчет бутстреп оценки коэффициента эксцесса выборки для матрицы с размерностью 5×5 . Генерируется 10 бутстреп выборок.

```
>> X = normrnd(0,1,5,5);
```

```
X =
```

```
-1.0082  0.2710  -0.3135  -0.3609  0.4938
-0.6646  1.5350  -0.6022  0.5536  -0.8709
 0.5582  -1.0523  1.2591  -1.5564  0.0798
-1.1885  0.6256  0.8585  -0.2067  -0.5216
-0.7755  -0.7976  -2.1053  -0.4256  -1.4139
```

```
>> [bootstat,bootsam] = bootstrp(10,'kurtosis', X)
```

```
bootstat =
```

```
1.1667  1.1667  1.1667  1.1667  1.1667
2.6310  2.4268  1.2410  2.3911  2.3623
3.1957  3.0127  2.9198  3.1830  1.7715
1.2219  1.4334  2.9249  1.4517  1.4670
2.8096  2.6053  1.3214  2.5753  2.5508
2.7328  1.5063  2.0226  2.5297  2.1854
2.7544  2.2273  1.2500  3.2284  2.8191
1.2057  1.1850  3.1529  1.1873  2.2188
1.1667  1.1667  1.1667  1.1667  1.1667
1.6549  1.9115  2.4535  1.2021  2.4930
```

```
bootsam =
```

```
5  2  4  4  2  2  5  4  1  2
1  4  4  4  3  5  2  3  1  4
1  2  1  3  2  3  1  5  4  1
5  3  4  3  2  4  5  3  4  5
1  4  3  2  4  4  5  4  1  2
```

```
>> k = kurtosis(X)
```

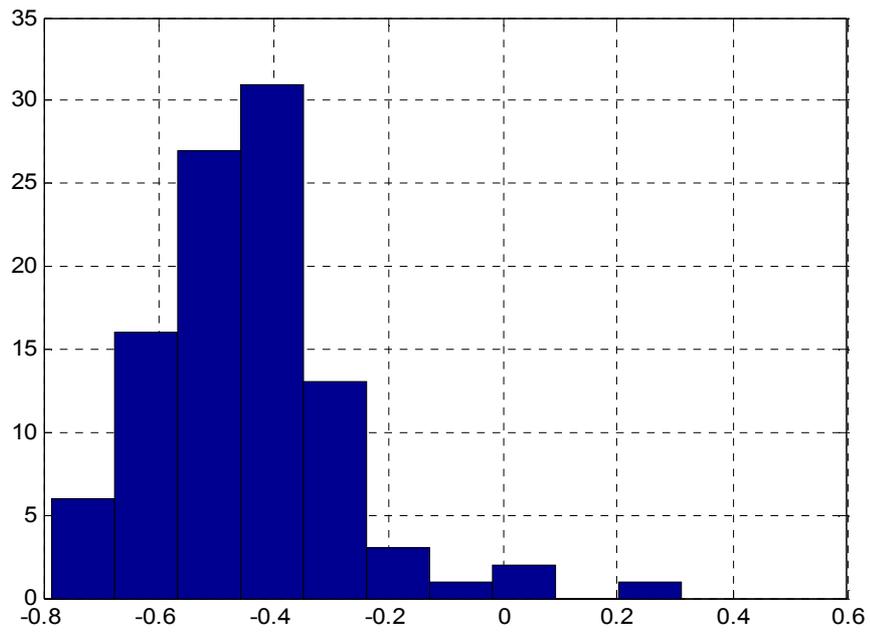

11	20	17	18	8	9	8	8	17	16
15	16	2	2	2	16	17	2	17	15
5	15	5	15	5	3	10	2	16	14
2	20	14	13	5	14	14	19	18	9
5	19	14	12	16	6	13	17	7	5
4	5	16	4	20	13	15	9	8	8
17	12	15	3	19	11	6	9	10	11
8	4	2	9	14	1	17	4	5	9
8	16	6	11	14	10	3	18	8	20
3	20	18	3	9	11	8	11	4	9
6	8	8	1	6	18	11	7	9	7

Графическое представление полученных результатов

```

>> X1 = normrnd(0,1,20,1);
>> X2 = normrnd(0,1,20,1);
>> [bootstat,bootsam] = bootstrp(100,'corrcoef', X1, X2);
>> hist(bootstat(:,2))
>> grid on

```



corrcoef

Расчет матрицы парных коэффициентов корреляции

Синтаксис

```
R = corrcoef(X)
R = corrcoef(x,y)
[R,P]=corrcoef(...)
[R,P,RLO,RUP]=corrcoef(...)
[...] =corrcoef(..., 'param1', val1, 'param2', val2, ...)
```

Описание

`R = corrcoef(X)` функция предназначена для расчета матрицы парных коэффициентов корреляции R выборок представленных в виде матрицы X . Наблюдения располагаются построчно в матрице X , выборки n по столбцам.

Расчет (i,j) элемента матрицы R осуществляется по формуле

$$R(i,j) = \frac{C(i,j)}{\sqrt{C(i,i) \cdot C(j,j)}}$$

где $C = \text{cov}(X)$ n матрица ковариаций.

`R = corrcoef(x,y)` функция предназначена для расчета матрицы парных коэффициентов корреляции R векторов x и y . Тот же результат можно получить при использовании `corrcoef([x y])`.

`[R,P]=corrcoef(...)` функция возвращает матрицы парных коэффициентов корреляции R и уровней значимости P , используемых при проверке гипотезы об отсутствии корреляции. Каждое значение P является значением вероятности получить величину коэффициента корреляции более, чем рассчитанное выборочное значение под действием случайных факторов когда истинное значение коэффициента корреляции равно нулю. Если $P(i,j)$ менее 0,05, то значение коэффициента корреляции $R(i,j)$ является значимым.

`[R,P,RLO,RUP]=corrcoef(...)` функция возвращает матрицы парных коэффициентов корреляции R , уровней значимости P , нижних RLO и верхних RUP границ 95% доверительных интервалов коэффициентов корреляции.

`[...] =corrcoef(..., 'param1', val1, 'param2', val2, ...)` В этом варианте синтаксиса функции дополнительные входные параметры определяют:

'alpha'	Значение уровня значимости. Доверительная вероятность определяется как $100 \cdot (1 - \alpha)\%$. По умолчанию уровень значимости равен 0,05, что соответствует 95% доверительному интервалу коэффициента корреляции.
'rows'	Определяет способ исключения строк матрицы X со значениями NaN при расчете коэффициента корреляции. Возможные значения параметра: 'all' ñ используются все строки (значение по умолчанию), 'complete' ñ исключаются строки со значениями NaN, 'pairwise' ñ при расчете $R(i,j)$ исключаются строки, содержащие NaN в столбце i или j.

Значения уровня значимости рассчитываются на основе преобразования коэффициента корреляции в t статистику с $n-2$ степенями свободы, где n ñ количество строк матрицы X. Границы доверительного интервала коэффициента корреляции рассчитываются на основании того, что статистика, рассчитанная как $0.5 \cdot \log((1+R)/(1-R))$ имеет асимптотическое приближение к нормальному закону с дисперсией равной $1/(n-3)$. Рассчитанные таким образом границы доверительного интервала коэффициента корреляции являются точными при больших выборках, когда X распределены по многомерному нормальному закону. Параметр 'rows' равный 'pairwise' может привести к получению матрицы R которая не будет положительно определенной.

Функция corrcoef является функцией ядра MATLAB.

Примеры использования функции расчета матрицы парных коэффициентов корреляции

Расчет матрицы парных коэффициентов корреляции для 5 выборок с объемом 20 элементов представленных в виде матрицы

```
>> X = normrnd(0,1,20,5);
>> R = corrcoef(X)
R =
    1.0000    0.4873    0.3854    0.0931   -0.2074
    0.4873    1.0000    0.6276   -0.3016   -0.3204
    0.3854    0.6276    1.0000   -0.4313   -0.4647
    0.0931   -0.3016   -0.4313    1.0000    0.1005
   -0.2074   -0.3204   -0.4647    0.1005    1.0000
```

Расчет матрицы парных коэффициентов корреляции для 2 выборок с объемом 20 элементов представленных в виде 2 векторов

```
>> X = normrnd(0,1,20,1);
>> Y = normrnd(0,1,20,1);
>> R = corrcoef(X,Y)
```

```
R =  
 1.0000 -0.4462  
-0.4462  1.0000
```

Расчет матриц парных коэффициентов корреляции и уровней значимости для 5 выборок с объемом 20 элементов представленных в виде матрицы

```
>> X = normrnd(0,1,20,5);
```

```
>> [R P] = corrcoef(X)
```

```
R =  
 1.0000 -0.1259  0.1972 -0.1178 -0.0923  
-0.1259  1.0000 -0.1805  0.2883  0.0972  
 0.1972 -0.1805  1.0000  0.0367 -0.0704  
-0.1178  0.2883  0.0367  1.0000  0.1538  
-0.0923  0.0972 -0.0704  0.1538  1.0000
```

```
P =  
 1.0000  0.5970  0.4046  0.6208  0.6987  
 0.5970  1.0000  0.4463  0.2177  0.6836  
 0.4046  0.4463  1.0000  0.8780  0.7680  
 0.6208  0.2177  0.8780  1.0000  0.5175  
 0.6987  0.6836  0.7680  0.5175  1.0000
```

Расчет матриц парных коэффициентов корреляции, уровней значимости и границ 95% доверительного интервала для 5 выборок с объемом 20 элементов представленных в виде матрицы

```
>> X = normrnd(0,1,20,5);
```

```
>> [R P RLO RUP] = corrcoef(X)
```

```
R =  
 1.0000 -0.2635  0.1917 -0.0096 -0.1274  
-0.2635  1.0000 -0.4230  0.3252 -0.1207  
 0.1917 -0.4230  1.0000  0.2903  0.2034  
-0.0096  0.3252  0.2903  1.0000  0.1955  
-0.1274 -0.1207  0.2034  0.1955  1.0000
```

```
P =  
 1.0000  0.2616  0.4181  0.9678  0.5924  
 0.2616  1.0000  0.0631  0.1618  0.6124  
 0.4181  0.0631  1.0000  0.2143  0.3897  
 0.9678  0.1618  0.2143  1.0000  0.4088  
 0.5924  0.6124  0.3897  0.4088  1.0000
```

```
RLO =  
 1.0000 -0.6323 -0.2741 -0.4502 -0.5395  
-0.6323  1.0000 -0.7290 -0.1370 -0.5346  
-0.2741 -0.7290  1.0000 -0.1746 -0.2628  
-0.4502 -0.1370 -0.1746  1.0000 -0.2704  
-0.5395 -0.5346 -0.2628 -0.2704  1.0000
```

```
RUP =
    1.0000    0.2026    0.5846    0.4347    0.3339
    0.2026    1.0000    0.0240    0.6711    0.3400
    0.5846    0.0240    1.0000    0.6494    0.5926
    0.4347    0.6711    0.6494    1.0000    0.5872
    0.3339    0.3400    0.5926    0.5872    1.0000
```

Расчет матриц парных коэффициентов корреляции, уровней значимости и границ 99% доверительного интервала для 5 выборок с объемом 20 элементов представленных в виде матрицы

```
>> X = normrnd(0,1,20,5);
>> [R P RLO RUP] = corrcoef(X,'alpha',0.01)
```

```
R =
    1.0000    0.4454    0.2195   -0.2765    0.2113
    0.4454    1.0000    0.2026    0.1240   -0.0572
    0.2195    0.2026    1.0000   -0.3124   -0.1067
   -0.2765    0.1240   -0.3124    1.0000   -0.1876
    0.2113   -0.0572   -0.1067   -0.1876    1.0000
```

```
P =
    1.0000    0.0491    0.3526    0.2380    0.3712
    0.0491    1.0000    0.3916    0.6026    0.8106
    0.3526    0.3916    1.0000    0.1800    0.6545
    0.2380    0.6026    0.1800    1.0000    0.4284
    0.3712    0.8106    0.6545    0.4284    1.0000
```

```
RLO =
    1.0000   -0.1448   -0.3814   -0.7204   -0.3887
   -0.1448    1.0000   -0.3963   -0.4622   -0.5928
   -0.3814   -0.3963    1.0000   -0.7388   -0.6242
   -0.7204   -0.4622   -0.7388    1.0000   -0.6721
   -0.3887   -0.5928   -0.6242   -0.6721    1.0000
```

```
RUP =
    1.0000    0.8018    0.6899    0.3283    0.6854
    0.8018    1.0000    0.6806    0.6348    0.5135
    0.6899    0.6806    1.0000    0.2928    0.4759
    0.3283    0.6348    0.2928    1.0000    0.4094
    0.6854    0.5135    0.4759    0.4094    1.0000
```

Расчет матриц парных коэффициентов корреляции, уровней значимости и границ 99% доверительного интервала для 5 выборок с объемом 20 элементов представленных в виде матрицы, содержащую нечисловые элементы NaN. Режим исключения нечисловых элементов парный построчный - 'pairwise'.

```
>> X = normrnd(0,1,20,5);
>> X([1 20 25 36 45 90 99]) = [NaN NaN NaN NaN NaN];
>> [R P RLO RUP] = corrcoef(X,'alpha',0.01,'rows','pairwise')
```

```

R =
    1.0000    0.1399   -0.0245    0.1995   -0.2121
    0.1399    1.0000    0.2357   -0.0335    0.0030
   -0.0245    0.2357    1.0000   -0.1962    0.1095
    0.1995   -0.0335   -0.1962    1.0000    0.1604
   -0.2121    0.0030    0.1095    0.1604    1.0000

P =
    1.0000    0.6053    0.9256    0.4274    0.4304
    0.6053    1.0000    0.3465    0.8950    0.9913
    0.9256    0.3465    1.0000    0.4209    0.6758
    0.4274    0.8950    0.4209    1.0000    0.5249
    0.4304    0.9913    0.6758    0.5249    1.0000

RLO =
    1.0000   -0.5180   -0.6125   -0.4324   -0.7305
   -0.5180    1.0000   -0.4011   -0.6035   -0.6116
   -0.6125   -0.4011    1.0000   -0.6872   -0.5216
   -0.4324   -0.6035   -0.6872    1.0000   -0.4647
   -0.7305   -0.6116   -0.5216   -0.4647    1.0000

RUP =
    1.0000    0.6938    0.5809    0.7000    0.4614
    0.6938    1.0000    0.7188    0.5591    0.6153
    0.5809    0.7188    1.0000    0.4180    0.6631
    0.7000    0.5591    0.4180    1.0000    0.6788
    0.4614    0.6153    0.6631    0.6788    1.0000

```

Пример генерации матрицы случайных чисел с размерностью 30×4, у которой 4 столбец коррелирован с остальными столбцами

Генерация некоррелированных выборок

```
>> x = randn(30,4);
```

Создание 4 столбца коррелированного с первыми тремя

```
>> x(:,4) = sum(x,2);
```

Расчет матриц точечных оценок коэффициента корреляции и значений уровня значимости

```
>> [r,p] = corrcoef(x)
```

```

r =
    1.0000   -0.3566    0.1929    0.3457
   -0.3566    1.0000   -0.1429    0.4461
    0.1929   -0.1429    1.0000    0.5183
    0.3457    0.4461    0.5183    1.0000

p =
    1.0000    0.0531    0.3072    0.0613
    0.0531    1.0000    0.4511    0.0135
    0.3072    0.4511    1.0000    0.0033
    0.0613    0.0135    0.0033    1.0000

```

Определение индексов значимых коэффициентов корреляции

```
>> [i,j] = find(p<0.05)
```

```
ans =
```

```
4 2
```

```
4 3
```

```
2 4
```

```
3 4
```

cov

Расчет ковариационной матрицы

Синтаксис

```
C = cov(X)
C = cov(x,y)
```

Описание

`C = cov(X)` функция предназначена для расчета ковариационной матрицы C . Если задана одна выборка, X - вектор, C является дисперсией выборки. Если X матрица, где строки являются наблюдениями, а столбцы выборками, C представляет собой ковариационную матрицу. По диагонали матрицы C расположены значения дисперсий выборок X .

`C = cov(x,y)` функция предназначена для расчета ковариационной матрицы C для двух выборок x , y заданных как векторы-столбцы. Размерность векторов должна совпадать. Тот же результат можно получить при использовании варианта вызова `cov([x y])`.

Функция `cov` является функцией ядра MATLAB.

Алгоритм, используемый при расчете ковариационной матрицы

```
[n,p] = size(X);
X = X - ones(n,1) * mean(X);
Y = X' * X / (n-1);
```

Примеры использования функции расчета матрицы ковариаций

Расчет ковариационной матрицы для 5 выборок с объемом 20 элементов представленных в виде матрицы

```
>> X = normrnd(0,1,20,5);
>> C = cov(X)
```

```
C =
    1.0823   -0.1735   -0.0628    0.2288   -0.1609
   -0.1735    1.0714    0.0230   -0.2807    0.0125
   -0.0628    0.0230    0.8124    0.0208   -0.0703
    0.2288   -0.2807    0.0208    0.9275    0.0782
   -0.1609    0.0125   -0.0703    0.0782    0.3810
```

Расчет ковариационной матрицы для 2 выборок с объемом 20 элементов представленных в виде 2 векторов

```
>> X = normrnd(0,1,20,1);
```

```
>> Y = normrnd(0,1,20,1);
```

```
>> C = cov (X,Y)
```

```
C =
```

```
0.8092 0.2905
```

```
0.2905 0.9515
```

Линейный регрессионный анализ

polyconf

Расчет значений и доверительных интервалов однофакторной регрессионной полиномиальной модели произвольного порядка

Синтаксис

```
[Y,DELTA] = polyconf(p,X,S)
[Y,DELTA] = polyconf(p,X,S,alpha)
```

Описание

[Y,DELTA] = polyconf(p,X,S) функция позволяет рассчитать значения Y однофакторной регрессионной полиномиальной модели произвольного порядка в заданных точках X. Коэффициенты регрессионной модели задаются вектором p.

Вектор коэффициентов p содержит n+1 элемент, расположенных по убыванию степени независимой переменной согласно формуле:

$$p(x) = p_1 x^n + p_2 x^{n-1} + \dots + p_n x + p_{n+1},$$

где n – порядок полинома.

Входной аргумент S рассчитывается с использованием функции polyfit. Результаты расчета значений зависимой переменной могут быть представлены как Y±DELTA, где DELTA соответствует 95% доверительному интервалу. При расчете DELTA предполагается, что отклонения значений зависимой переменной y от регрессионной модели независимы и распределены по нормальному закону с постоянной дисперсией.

[Y,DELTA] = polyconf(p,X,S,alpha) функция позволяет рассчитать границы доверительного интервала DELTA для уровня значимости alpha. Доверительная вероятность определяется как 100(1-alpha)%.

Входной аргумент X может быть задан как вектор или матрица. Значения регрессионной модели рассчитываются для каждого элемента X. Размерность X, Y и DELTA будет одинаковой.

Примеры использования функции расчета значений однофакторной полиномиальной регрессионной модели произвольного порядка

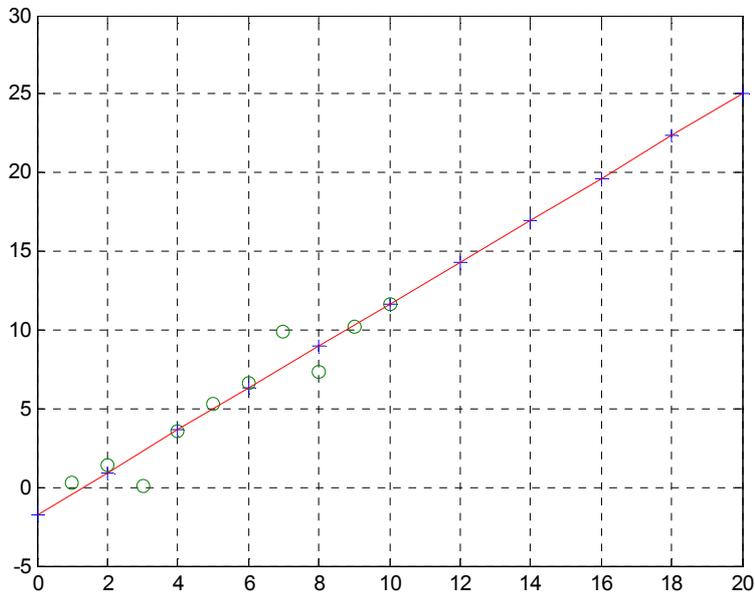
Расчет значений линейной модели и их 95% доверительных интервалов

```
>> x=1:1:10;
>> d=normrnd(0,2,1,10);
>> y=x+d;
>> [p S] = polyfit(x,y,1)
p =
    1.3378   -1.7450
S =
    R: [2x2 double]
    df: 8
    normr: 3.6995
>> X = 0:2:20;
>> [Y,DELTA] = polyconf(p,X,S)
Y =
Columns 1 through 7
   -1.7450    0.9307    3.6063    6.2820    8.9577   11.6334   14.3091
Columns 8 through 11
```

```

16.9848 19.6605 22.3362 25.0119
DELTA =
Columns 1 through 7
 3.6528 3.3702 3.2024 3.1678 3.2706 3.4986 3.8297
Columns 8 through 11
 4.2396 4.7080 5.2190 5.7613
>> f=p(1)*X+ p(2);
>> plot(X,Y,'+',x,y,foi,X,f)
>> grid on

```

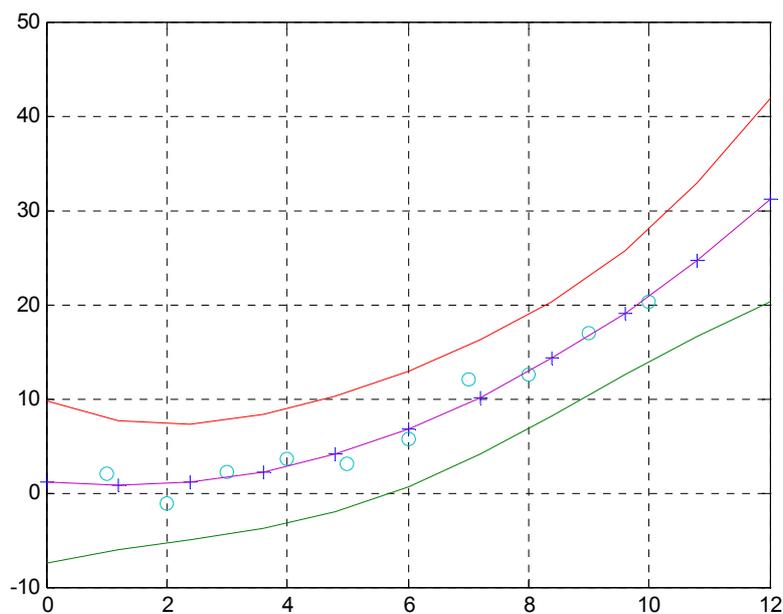


Расчет значений квадратической модели и их 99% доверительных интервалов

```

>> x=1:1:10;
>> d=normrnd(0,2,1,10);
>> y=x.^2/5+d;
>> [p S] = polyfit(x,y,2)
p =
 0.2601 -0.6233 1.1314
S =
 R: [3x3 double]
 df: 7
 normr: 4.1858
>> X = 0:1.2:12;
>> alpha = 0.01;
>> [Y,DELTA] = polyconf(p,X,S,alpha)
Y =
Columns 1 through 7
 1.1314 0.7580 1.1335 2.2580 4.1314 6.7537 10.1251
Columns 8 through 11
14.2454 19.1146 24.7328 31.0999
DELTA =
Columns 1 through 7
 8.5473 6.8320 6.1153 6.0258 6.1153 6.1259 6.0382
Columns 8 through 11
 6.0700 6.6497 8.1858 10.7948
>> Y_min=Y-DELTA;
>> Y_max=Y+DELTA;
>> f=p(1)*X.^2+ p(2).* X + p(3);
>> plot(X,Y,'+', X, Y_min,X,Y_max,x,y,'o',X,f)
>> grid on

```



Расчет значений квадратической модели и их 99% доверительных интервалов для матрицы значений X.

```
>> x=1:1:10;
>> d=normrnd(0,2,1,10);
>> y=x.^2/5+d;
>> [p S] = polyfit(x,y,2)
p =
    0.2601  -0.6233  1.1314
S =
    R: [3x3 double]
    df: 7
    normr: 4.1858
>> X = magic(3)
X =
     8     1     6
     3     5     7
     4     9     2
>> alpha = 0.01;
>> [Y,DELTA] = polyconf(p,X,S,alpha)
Y =
    12.7887    0.7682    6.7537
     1.6021    4.5164    9.5112
     2.7992   16.5863    0.9251
DELTA =
     6.0227     7.0429     6.1259
     6.0227     6.1259     6.0535
     6.0535     6.2609     6.2609
```

polyfit

Расчет коэффициентов однофакторной регрессионной полиномиальной модели произвольного порядка

Синтаксис

```
[p,S] = polyfit(x,y,n)
```

Описание

`p = polyfit(x,y,n)` функция позволяет рассчитать коэффициенты p полиномиальной регрессионной модели n -й степени для выборки x , y методом наименьших квадратов, где x - независимая переменная, y - зависимая переменная. Зависимая и независимая переменные задаются как векторы с одинаковым числом элементов. Вектор коэффициентов p содержит $n+1$ элемент, расположенных по убыванию степени независимой переменной согласно формуле:

$$p(x) = p_1 x^n + p_2 x^{n-1} + \dots + p_n x + p_{n+1}.$$

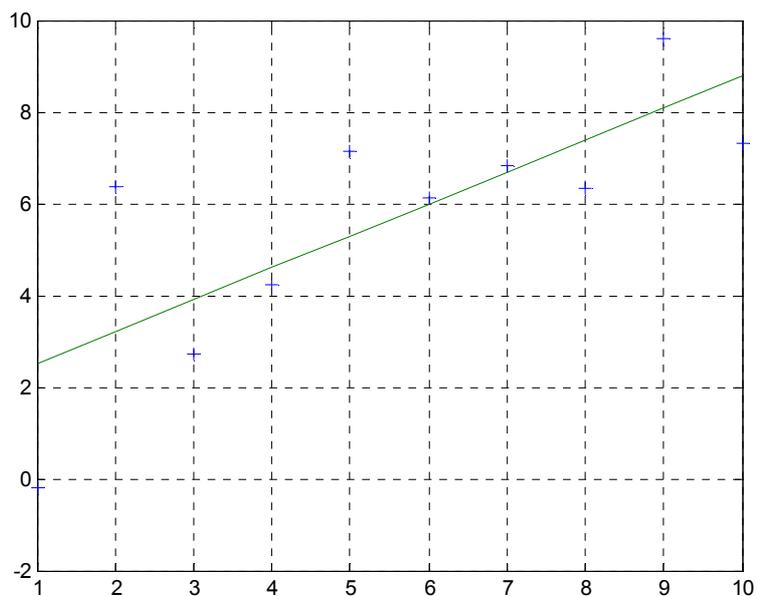
`[p,S] = polyfit(x,y,n)` функция возвращает коэффициенты p полиномиальной регрессионной модели n -й степени и матрицы S для выборки x , y . S используется в качестве входного аргумента функции `polyval` для расчета границ доверительного интервала значений полиномиальной регрессионной модели в заданных точках. Если отклонения значений зависимой переменной y от регрессионной модели независимы и распределены по нормальному закону с постоянной дисперсией, границы доверительного интервала, полученные `polyval`, включают как минимум 50% рассчитанных значений.

`polyfit` является стандартной функцией MATLAB.

Примеры использования функции расчета коэффициентов однофакторной регрессионной полиномиальной модели произвольного порядка

Расчет коэффициентов линейной модели

```
>> x=1:1:10;  
>> d=normrnd(0,2,1,10);  
>> y=x+d;  
>> p = polyfit(x,y,1)  
p =  
    1.0000  -0.4326  
>> f=p(1)*x+ p(2);  
>> plot(x,y,'+',x,f)  
>> grid on
```

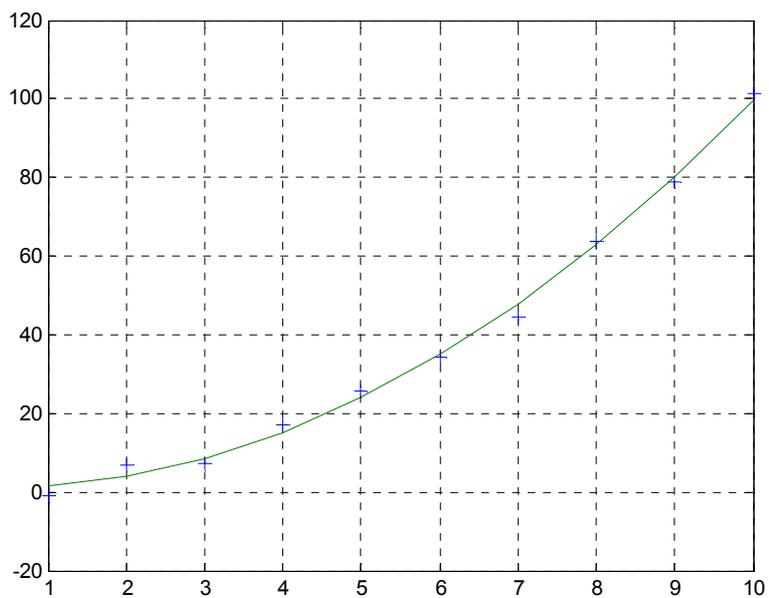


Расчет коэффициентов квадратической модели и матрицы S

```

>> x=1:1:10;
>> d=normrnd(0,2,1,10);
>> y=x.^2+d;
>> [p S] = polyfit(x,y,2)
p =
    1.0512   -0.6534    0.9738
S =
    R: [3x3 double]
    df: 7
    normr: 6.1614
>> S.R
ans =
-159.1634 -19.0056 -2.4189
     0 -4.8771 -1.8510
     0  0  0.8502
>> f=p(1)*x.^2+ p(2).*x+ p(3);
>> plot(x,y,'+',x,f)
>> grid on

```



polyval

Расчет значений однофакторной регрессионной полиномиальной модели произвольного порядка

Синтаксис

```
Y = polyval(p,X)
[Y,DELTA] = polyval(p,X,S)
```

Описание

`Y = polyval(p,X)` функция позволяет рассчитать значения Y однофакторной регрессионной полиномиальной модели произвольного порядка в заданных точках X . Коэффициенты регрессионной модели задаются вектором p .

Вектор коэффициентов p содержит $n+1$ элемент, расположенных по убыванию степени независимой переменной согласно формуле:

$$p(x) = p_1 x^n + p_2 x^{n-1} + \dots + p_n x + p_{n+1},$$

где n – порядок полинома.

`[Y,DELTA] = polyval(p,X,S)` функция позволяет рассчитать значения Y однофакторной регрессионной полиномиальной модели произвольного порядка и границы 95% доверительного интервала $DELTA$ в заданных точках X . Результаты расчета значений зависимой переменной могут быть представлены как $Y \pm DELTA$. Входной аргумент S рассчитывается с использованием функции `polyfit`. Если отклонения значений зависимой переменной y от регрессионной модели независимы и распределены по нормальному закону с постоянной дисперсией, границы доверительного интервала, полученные `polyval`, включают как минимум 50% рассчитанных значений.

Входной аргумент X может быть задан как вектор или матрица. Значения регрессионной полиномиальной модели рассчитываются для каждого элемента X . Размерность X , Y и $DELTA$ будет одинаковой.

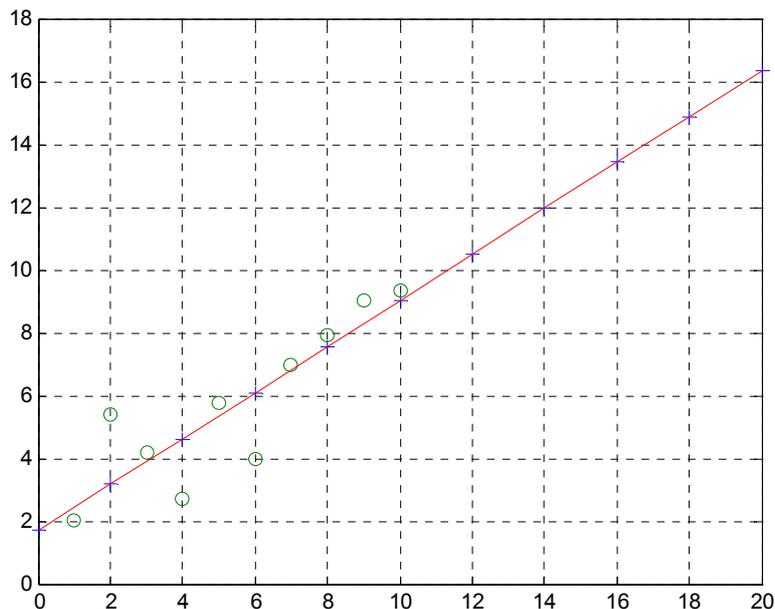
`polyval` является стандартной функцией MATLAB.

Примеры использования функции расчета значений однофакторной регрессионной полиномиальной модели произвольного порядка

Расчет значений линейной модели

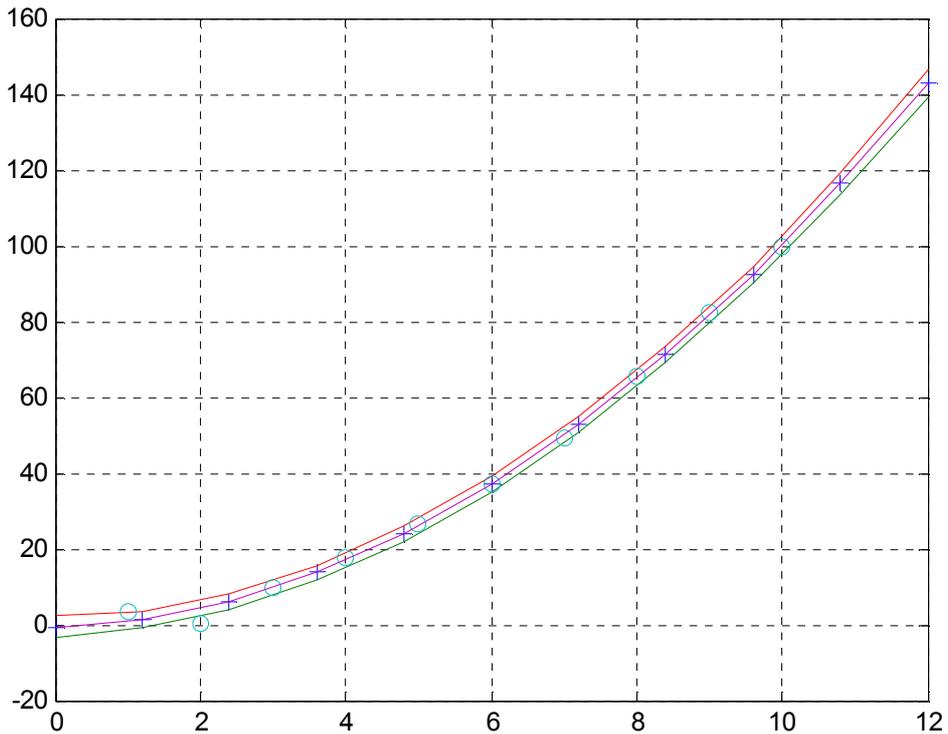
```
>> x=1:1:10;
>> d=normrnd(0,2,1,10);
>> y=x+d;
>> [p S] = polyfit(x,y,1)
p =
    0.7334    1.6928
S =
    R: [2x2 double]
    df: 8
    normr: 3.7728
>> X = 0:2:20;
>> Y = polyval(p,X)
Y =
Columns 1 through 7
    1.6928    3.1597    4.6266    6.0934    7.5603    9.0272    10.4940
Columns 8 through 11
    11.9609    13.4277    14.8946    16.3615
>> f=p(1)*X+ p(2);
```

```
>> plot(X,Y,'+',x,y,íoi,X,f)
>> grid on
```



Расчет значений квадратической модели и их 95% доверительных интервалов

```
>> x=1:1:10;
>> d=normrnd(0,2,1,10);
>> y=x.^2+d;
>> [p S] = polyfit(x,y,2)
p =
    0.9491    0.6023   -0.7749
S =
    R: [3x3 double]
    df: 7
    normr: 4.9811
>> X = 0:1:12;
>> [Y DELTA] = polyval(p,X,S)
Y =
Columns 1 through 7
   -0.7749    1.3145    6.1372   13.6933   23.9826   37.0053   52.7613
Columns 8 through 11
   71.2507   92.4734  116.4294  143.1187
DELTA =
Columns 1 through 7
    2.9065    2.3232    2.0795    2.0490    2.0795    2.0831    2.0533
Columns 8 through 11
    2.0641    2.2612    2.7836    3.6708
>> Y_min=Y-DELTA;
>> Y_max=Y+DELTA;
>> f=p(1)*X.^2+ p(2).* X + p(3);
>> plot(X,Y,'+', X, Y_min,X,Y_max,x,y,'o',X,f)
>> grid on
```



Расчет значений квадратической модели и их 95% доверительных интервалов для матрицы значений X.

```
>> x=1:1:10;
>> d=normrnd(0,2,1,10);
>> y=x.^2/5+d;
>> [p S] = polyfit(x,y,2)
p =
    0.2601   -0.6233    1.1314
S =
     R: [3x3 double]
     df: 7
     normr: 4.1858
>> X = magic(3)
X =
     8     1     6
     3     5     7
     4     9     2
>> [Y DELTA] = polyval(p,X,S)
Y =
    12.7887    0.7682    6.7537
     1.6021    4.5164    9.5112
     2.7992   16.5863    0.9251
DELTA =
     1.7210     2.0126     1.7505
     1.7210     1.7505     1.7298
     1.7298     1.7891     1.7891
```

Дисперсионный анализ

`anova1`

Однофакторный дисперсионный анализ (ANOVA)

Синтаксис

```
p = anova1(X)
p = anova1(X,group)
p = anova1(X,group,'displayopt')
[p,table] = anova1(...)
[p,table,stats] = anova1(...)
```

Описание

`p = anova1(X)` функция позволяет провести однофакторный дисперсионный анализ для сравнения средних арифметических значений одной или нескольких выборок одинакового объема. Выборки определяются входным аргументом `X`. `X` задается как матрица с размерностью $m \times n$, где m – число наблюдений в выборке (число строк `X`), n – количество выборок (число столбцов матрицы `X`). Выборки должны быть независимыми. Выходным аргументом функции является уровень значимости p нулевой гипотезы. Нулевая гипотеза состоит в том, что все выборки в матрице `X` взяты из одной генеральной совокупности или из разных генеральных совокупностей с равными средними арифметическими. p является вероятностью ошибки первого рода, или вероятностью необоснованно отвергнуть нулевую гипотезу. Если значение $p \approx 0$, то нулевая гипотеза может быть отвергнута, т.е. хотя бы одно среднее арифметическое отличается от остальных значений. Выбор критического уровня значимости $p_{кр}$ для условия принятия нулевой гипотезы

$$p \geq p_{кр},$$

предоставлен исследователю. В большинстве практических случаев $p_{кр}$ принимают равным 0,05; 0,01.

При проведении дисперсионного анализа общая дисперсия делится на две составляющие:

- межгрупповую дисперсию – дисперсию средних арифметических значений выборок матрицы `X` относительно общего среднего;
- внутригрупповую дисперсию – дисперсию значений выборок относительно выборочных средних.

Результаты расчета отображаются в двух графических окнах. В первое окно выводится таблица с результатами однофакторного дисперсионного анализа, во второе – диаграмма размаха для средних арифметических по заданным выборкам.

Таблица с результатами однофакторного дисперсионного анализа содержит 6 столбцов (рис. 1):

1. вид дисперсии (Source):
 - внутригрупповая (Columns),
 - межгрупповая (Error),
 - общая (Total);
2. сумму квадратов разностей (SS) между средним арифметическим и значениями выборки по каждому виду дисперсии;
3. число степеней свободы по каждому виду дисперсии (df);
4. среднее значение суммы квадратов разностей (MS) по каждому виду дисперсии, определяемое как отношение SS/df ;
5. значение статистики Фишера (F статистики) для MS;
6. значение уровня значимости p ($Prob>F$) для рассчитанного значения статистики F. Если величина F увеличивается, то значение p должно уменьшаться.

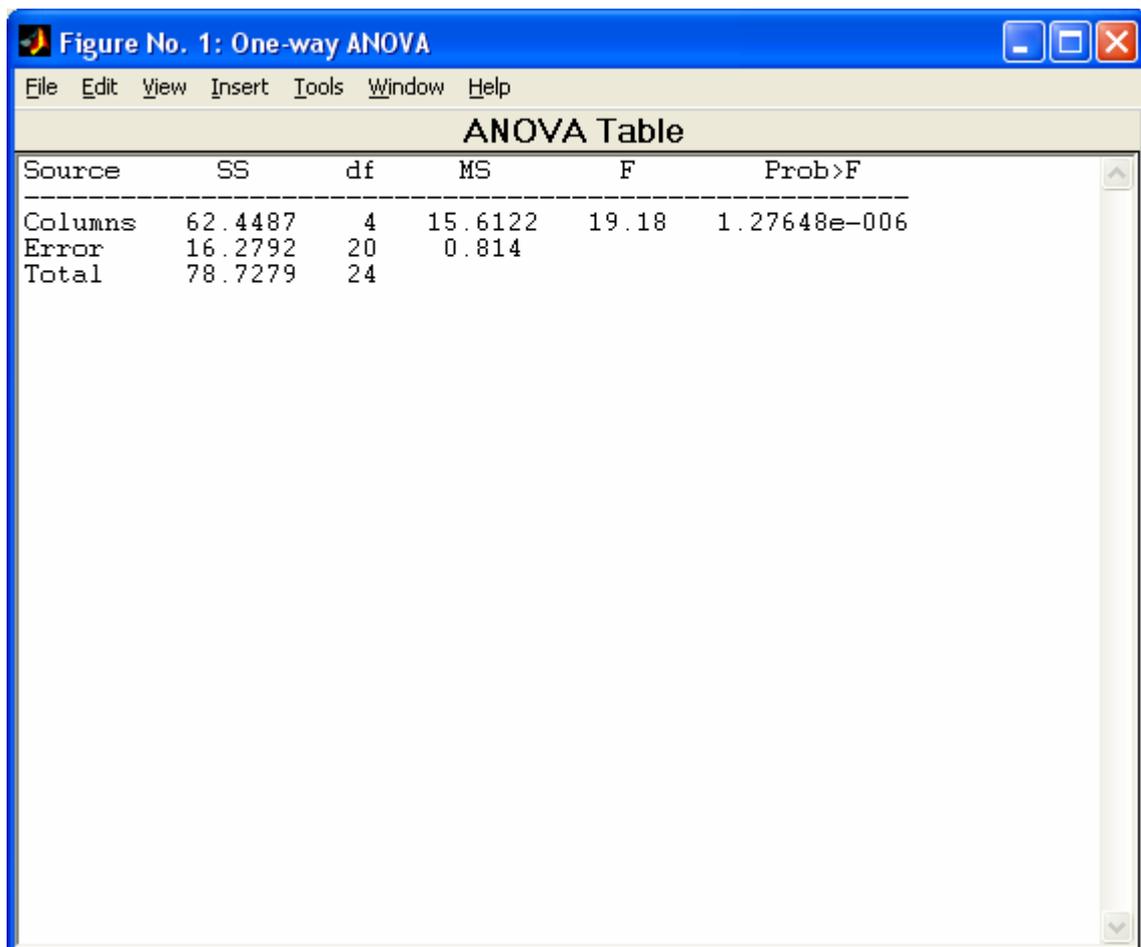


Рис. 1. Таблица однофакторного дисперсионного анализа.

Диаграмма размаха средних арифметических значений строится по выборкам, т.е. по столбцам матрицы X , и является аналогом диаграммы размаха для медиан выборок, получаемой при помощи функции `boxplot`. Чем больше разница между центральными линиями на диаграмме размаха (средними арифметическими выборки), тем больше разница между выборочными значениями статистики F и меньше соответствующие значения уровня значимости p .

`p = anova1(X, group)` входной аргумент `group` задает метки выборок X и названий соответствующих им графиков на диаграмме размаха. Метки выборок `group` задаются вектором строк или массивом ячеек. Число элементов вектора `group` должно быть равно количеству столбцов матрицы X .

Выборки могут быть заданы как вектор X . Деление элементов X на выборки выполняется при помощи входного аргумента `group`. `group` может быть задан вектор, символьный массив (вектор строк) или массив ячеек. Принадлежность элемента вектора X в выборке определяется одинаковым значением соответствующего элемента вектора `group`. Размерность векторов X и `group` должна совпадать.

Значения вектора `group` являются метками соответствующих графиков выборок на диаграмме размаха средних арифметических. Векторная форма задания выборок `X` по группирующей переменной `group` позволяет проводить однофакторный дисперсионный анализ по выборкам неодинакового объема. Если элемент вектора группирующей переменной `groupi` содержит пустую строку, пустую ячейку массива или нечисловое значение `NaN`, то соответствующий элемент данных в векторе `X` игнорируется при расчете.

`p = anova1(X,group,'displayopt')` входной аргумент `'displayopt'` позволяет явно задать отображение графических окон с таблицей результатов дисперсионного анализа и диаграммой размаха `'displayopt'='on'`, или подавить вывод графических окон `'displayopt'='off'`. Значение по умолчанию `'displayopt'='on'`.

`[p,table] = anova1(...)` функция возвращает таблицу с результатами однофакторного дисперсионного анализа в текстовой форме в командное окно MATLAB.

`[p,table,stats] = anova1(...)` функция возвращает структуру `stats` используемую для проведения попарных сравнений средних арифметических выборок. Проверка параметрической гипотезы о равенстве двух средних арифметических выполняется при помощи функции `multcompare`. Структура данных `stats` передается функции `multcompare` как входной аргумент.

Однофакторный дисперсионный анализ основан на следующих предположениях:

1. выборки распределены нормально;
2. выборки имеют одинаковую дисперсию;
3. наблюдения в выборках взаимно независимы.

Однофакторный дисперсионный анализ нечувствителен к малым отклонениям от первого и второго условий.

Примеры использования функции однофакторного дисперсионного анализа

Однофакторный дисперсионный анализ для пяти выборок с математическими ожиданиями изменяющимися в пределах от 1 до 5.

```
>> X = meshgrid(1:5)
```

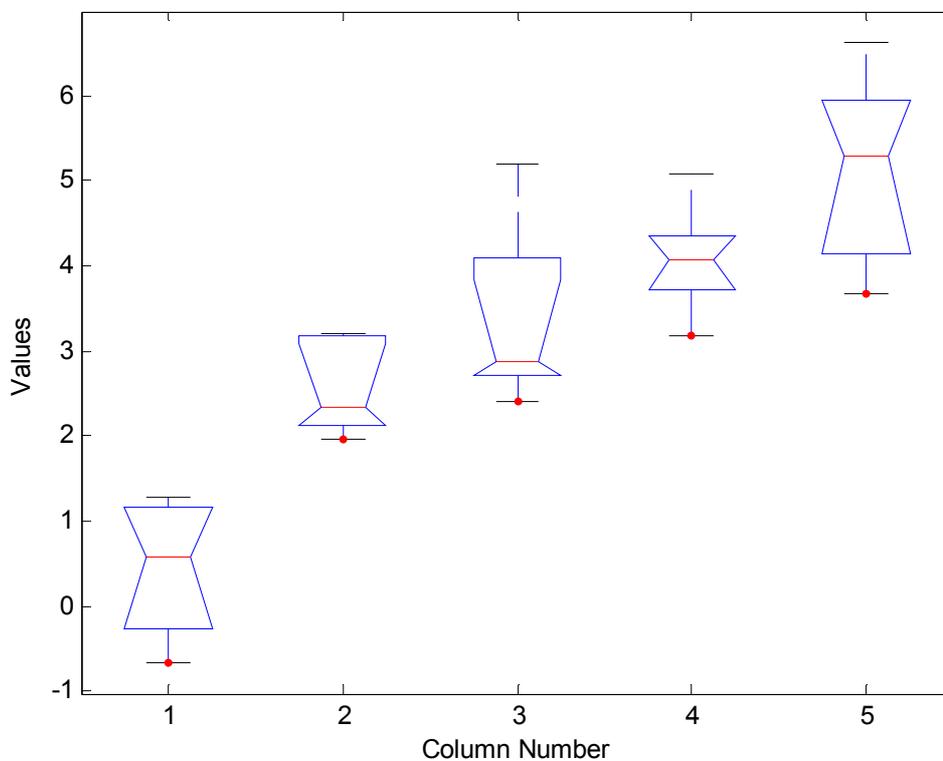
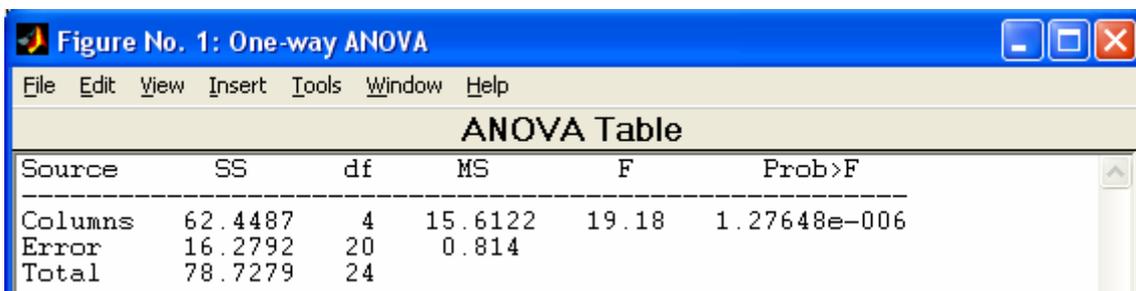
```
X =
```

```
 1  2  3  4  5
 1  2  3  4  5
```

```

1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
>> X = X + normrnd(0,1,5,5)
X =
    0.5674    3.1909    2.8133    4.1139    5.2944
   -0.6656    3.1892    3.7258    5.0668    3.6638
    1.1253    1.9624    2.4117    4.0593    5.7143
    1.2877    2.3273    5.1832    3.9044    6.6236
   -0.1465    2.1746    2.8636    3.1677    4.3082
>> p = anova1(X)
p =
    1.2765e-006

```

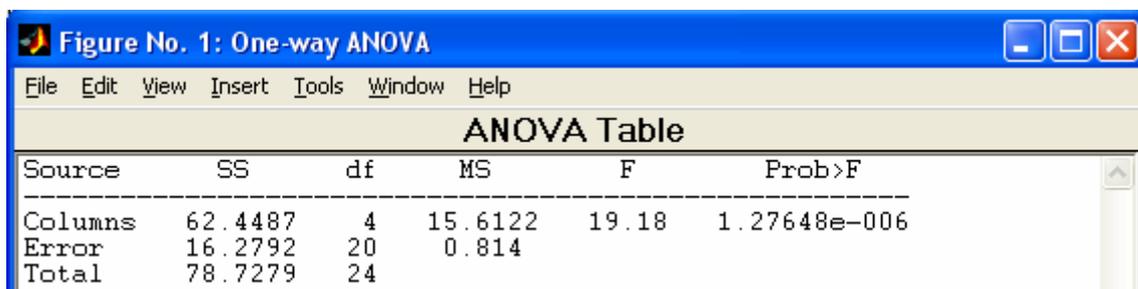


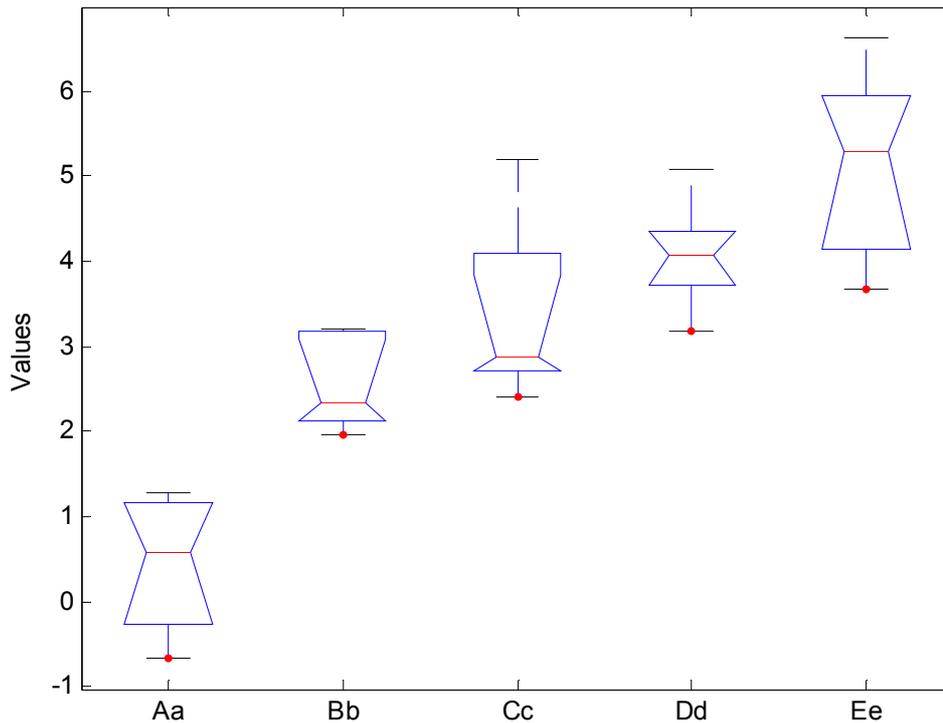
Однофакторный дисперсионный анализ для пяти выборок с математическими ожиданиями, изменяющимися в пределах от 1 до 5. Метки выборок заданы переменной group.

```

>> X = meshgrid(1:5)
X =
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
>> X = X + normrnd(0,1,5,5)
X =
     0.5674     3.1909     2.8133     4.1139     5.2944
    -0.6656     3.1892     3.7258     5.0668     3.6638
     1.1253     1.9624     2.4117     4.0593     5.7143
     1.2877     2.3273     5.1832     3.9044     6.6236
    -0.1465     2.1746     2.8636     3.1677     4.3082
>> group = ['Aa'; 'Bb'; 'Cc'; 'Dd'; 'Ee']
group =
Aa
Bb
Cc
Dd
Ee
>> p = anova1(X, group)
p =
    1.2765e-006

```



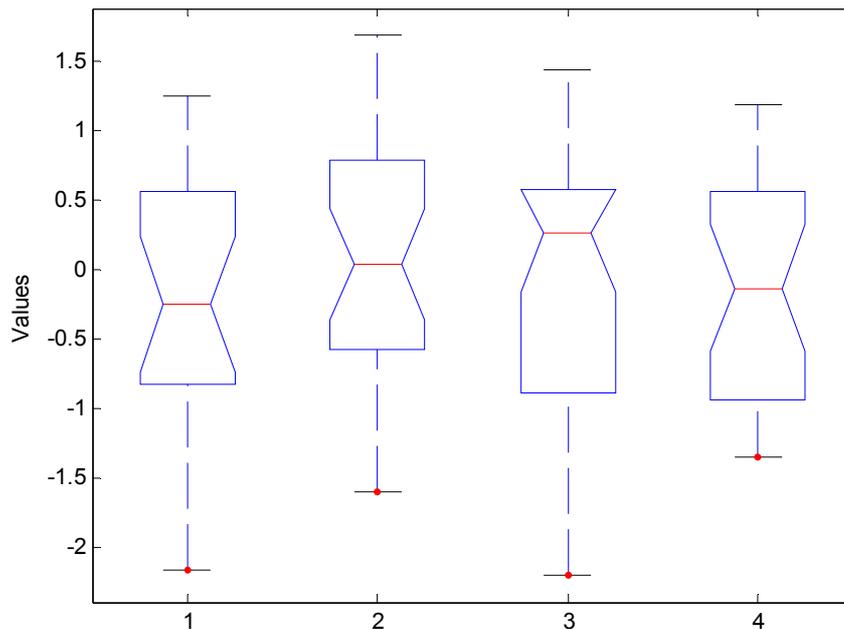


Однофакторный дисперсионный анализ для пяти выборок с переменным объемом. Группировка значений вектора X на выборки выполняется вектором целых чисел group.

```
>> X = normrnd(0,1,104,1);
>> group = unidrnd(4,104,1);
>> p = anova1(X, group)
p =
0.5473
```

Figure No. 5: One-way ANOVA

ANOVA Table					
Source	SS	df	MS	F	Prob>F
Groups	1.6684	3	0.55615	0.71	0.5473
Error	78.152	100	0.78152		
Total	79.8205	103			



Однофакторный дисперсионный анализ для пяти выборок с математическими ожиданиями, изменяющимися в пределах от 1 до 5. Метки выборок заданы переменной `group`. Функция возвращает таблицу с результатами однофакторного дисперсионного анализа в текстовой форме в командное окно MATLAB. Функция `anova1` возвращает структуру данных `stats`.

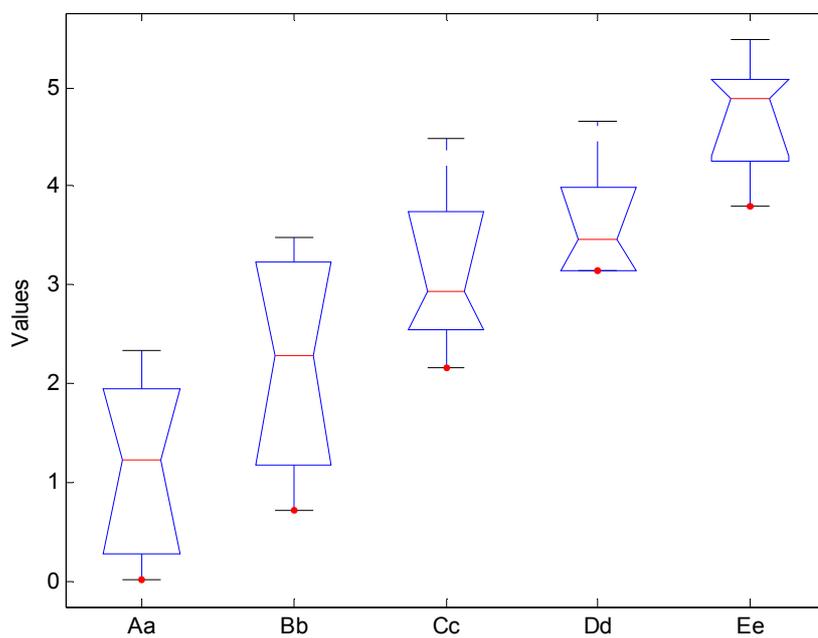
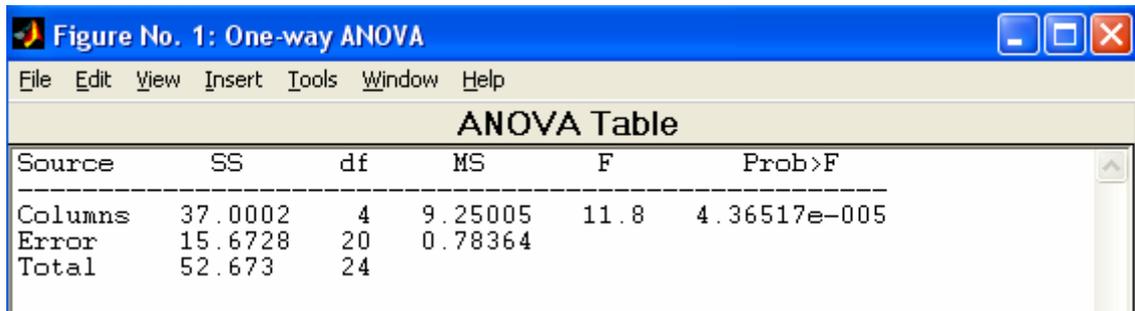
```
>> X = meshgrid(1:5);
>> X = X + normrnd(0,1,5,5);
>> group = ['Aa'; 'Bb'; 'Cc'; 'Dd'; 'Ee'];
>> [p,table,stats] = anova1(X, group)
p =
    4.3652e-005
table =
Columns 1 through 5
    'Source'    'SS'    'df'    'MS'    'F'
    'Columns'  [37.0002] [ 4] [9.2501] [11.8040]
    'Error'    [15.6728] [20] [0.7836] []
    'Total'    [52.6730] [24] [] []
Column 6
    'Prob>F'
    [4.3652e-005]
    []
    []
stats =
    gnames: {5x1 cell}
    n: [5 5 5 5 5]
```

source: 'anova1'

means: [1.1484 2.1861 3.1478 3.6339 4.7007]

df: 20

s: 0.8852



```
>> stats.gnames
```

```
ans =
```

```
'Aa'
```

```
'Bb'
```

```
'Cc'
```

```
'Dd'
```

```
'Ee'
```

anova2

Двухфакторный дисперсионный анализ

Синтаксис

```
p = anova2(X, reps)
p = anova2(X, reps, 'displayopt')
[p, table] = anova2(...)
[p, table, stats] = anova2(...)
```

Описание

`anova2(X, reps)` функция позволяет провести двухфакторный дисперсионный анализ для сравнения средних арифметических значений m столбцов и n строк в матрице X . Данные в столбцах и строках соответствуют разным уровням факторов A и B . Количество повторных наблюдений для комбинаций факторов A и B задается входным аргументом `reps`. `Reps` должен быть постоянным целым числом. Для переменного числа повторений по какому либо фактору используется функция `anova`.

Если фактор A имеет два уровня, фактор B три уровня и рассматриваются два повторных наблюдения для сочетаний уровней факторов A и B (`reps=2`), то матрица X примет следующий вид

$$\begin{array}{cc} \begin{array}{c} A = 1 \\ A = 2 \end{array} & \begin{array}{c} A = 2 \\ A = 1 \end{array} \\ \left[\begin{array}{cc} x_{111} & x_{121} \\ x_{112} & x_{122} \\ x_{211} & x_{221} \\ x_{212} & x_{222} \\ x_{311} & x_{321} \\ x_{312} & x_{322} \end{array} \right] & \left. \begin{array}{l} \} B = 1 \\ \} B = 2 \\ \} B = 3 \end{array} \right\} \end{array}$$

При `reps=1` (значение по умолчанию) функция `anova2` позволяет рассчитать вектор уровней значимости p содержащий два элемента:

1. уровень значимости $p(1)$ нулевой гипотезы, H_{0A} , состоящей в том, что все выборки соответствующие уровням фактора A , т.е. все столбцы в приведенной выше матрице X , извлечены из той же генеральной совокупности;
2. уровень значимости $p(2)$ нулевой гипотезы, H_{0B} , состоящей в том, что все выборки соответствующие уровням фактора B , т.е. все строки в приведенной выше матрице X , извлечены из той же генеральной совокупности;

И если количество повторений `reps71`, `anova2` возвращает третий элемент вектора `p`:

3. уровень значимости p нулевой гипотезы, H_{0AB} , состоящей в том, что отсутствует эффект взаимодействия между факторами А и В.

p является вероятностью ошибки первого рода, или вероятностью необоснованно отвергнуть нулевую гипотезу. Если значение p близко к нулю, то нулевая гипотеза может быть отвергнута. $p \approx 0$ для гипотезы H_{0A} означает, что хотя бы одно из средних арифметических, рассчитанных по столбцам, отличается от остальных значений, т.е. существует значимый главный эффект фактора А. $p \approx 0$ для гипотезы H_{0B} означает, что хотя бы одно из средних арифметических, рассчитанных по строкам, отличается от остальных значений, т.е. существует значимый главный эффект фактора В. $p \approx 0$ для гипотезы H_{0AB} означает, что существует значимый эффект взаимодействия между факторами А и В.

Выбор критического уровня значимости $p_{кр}$ для условия принятия нулевой гипотезы

$$p \geq p_{кр},$$

предоставлен исследователю. В большинстве практических случаев $p_{кр}$ принимают равным 0,05; 0,01.

При проведении дисперсионного анализа общая дисперсия делится на 3 или 4 составляющие в зависимости от значения `reps`:

- дисперсия между средними арифметическими по столбцам X (дисперсия между средними арифметическими по фактору А);
- дисперсия между средними арифметическими по строкам X (дисперсия между средними арифметическими по фактору В);
- дисперсия, обусловленная взаимодействием между строками и столбцами X (рассчитывается, если `reps72`);
- остаточная дисперсия.

Таблица с результатами двухфакторного дисперсионного анализа отображается в графическом окне.

Таблица с результатами двухфакторного дисперсионного анализа содержит 6 столбцов (рис. 1):

1. вид дисперсии (Source):

- дисперсия между средними арифметическими по столбцам X (Columns),

- дисперсия между средними арифметическими по строкам X (Rows),
 - дисперсия обусловленная взаимодействием между строками и столбцами X (Interaction),
 - остаточная дисперсия (Error)
 - общая дисперсия (Total);
2. сумму квадратов разностей (SS) между средним арифметическим и значениями выборки по каждому виду дисперсии;
 3. число степеней свободы по каждому виду дисперсии (df);
 4. среднее значение суммы квадратов разностей (MS) по каждому виду дисперсии, определяемое как отношение SS/df ;
 5. значение статистики Фишера (F статистики) для MS;
 6. значение уровня значимости p ($Prob>F$) для рассчитанного значения статистики F. Если величина F увеличивается, то значение p должно уменьшаться.

The screenshot shows a window titled "Figure No. 1: Two-way ANOVA" with a menu bar (File, Edit, View, Insert, Tools, Window, Help) and a table titled "ANOVA Table". The table contains the following data:

Source	SS	df	MS	F	Prob>F
Columns	15.75	2	7.875	56.7	0
Rows	4.5	1	4.5	32.4	0.0001
Interaction	0.0833	2	0.04167	0.3	0.7462
Error	1.6667	12	0.13889		
Total	22	17			

Рис. 1. Таблица двухфакторного дисперсионного анализа.

`p = anova2(X, reps, 'displayopt')` входной аргумент `'displayopt'` позволяет отобразить графическое окно с таблицей результатов

дисперсионного анализа `'displayopt'='on'`, или подавить вывод графического окна `'displayopt'='off'`. Значение по умолчанию `'displayopt'='on'`.

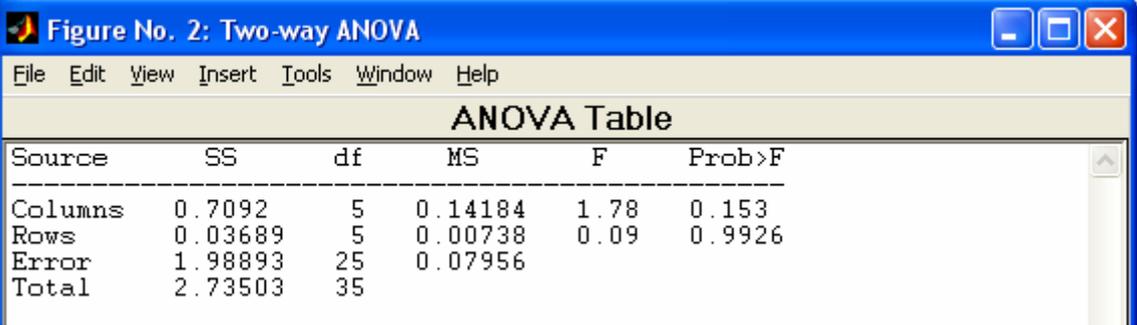
`[p,table] = anova2(...)` функция возвращает таблицу с результатами однофакторного дисперсионного анализа в текстовой форме в командное окно MATLAB.

`[p,table,stats] = anova2(...)` функция возвращает структуру `stats` используемую для проведения попарных сравнений средних арифметических выборок. Проверка параметрической гипотезы о равенстве двух средних арифметических выполняется при помощи функции `multcompare`. Структура данных `stats` передается функции `multcompare` как входной аргумент.

Примеры использования функции двухфакторного дисперсионного анализа

Двухфакторный дисперсионный анализ для шести уровней по каждому фактору и единичному числу повторений.

```
>> X = unifrnd(0,1,6,6);  
>> p = anova2(X,1)  
p =  
    0.1530    0.9926
```



The screenshot shows a MATLAB window titled "Figure No. 2: Two-way ANOVA". The window contains a table titled "ANOVA Table" with the following data:

Source	SS	df	MS	F	Prob>F
Columns	0.7092	5	0.14184	1.78	0.153
Rows	0.03689	5	0.00738	0.09	0.9926
Error	1.98893	25	0.07956		
Total	2.73503	35			

Двухфакторный дисперсионный анализ для шести уровней по первому фактору, двум уровням по второму фактору и 3-х повторений.

```
>> X = normrnd(0,1,6,6);  
>> p = anova2(X,3)  
p =  
    0.8380    0.9897    0.2309
```

Figure No. 1: Two-way ANOVA

ANOVA Table					
Source	SS	df	MS	F	Prob>F
Columns	2.1133	5	0.42266	0.41	0.838
Rows	0.0002	1	0.00017	0	0.9897
Interaction	7.6918	5	1.53837	1.49	0.2309
Error	24.8218	24	1.03424		
Total	34.6271	35			

Двухфакторный дисперсионный анализ для шести уровней по первому фактору, двум уровням по второму фактору и 3-х повторений. Функция возвращает вектор уровней значимости p , таблицу двухфакторного дисперсионного анализа `table`, структуру `stats` для парных сравнений средних арифметических выборок.

```
>> X = normrnd(0,1,6,6);
>> [p,table,stats] = anova2(X,3)
```

`p =`

```
0.1751 0.4596 0.0366
```

`table =`

```
'Source'      'SS'      'df' 'MS'      'F'      'Prob>F'
'Columns'     [ 5.2429] [ 5] [1.0486] [1.6911] [0.1751]
'Rows'        [ 0.3502] [ 1] [0.3502] [0.5648] [0.4596]
'Interaction' [ 8.8611] [ 5] [1.7722] [2.8582] [0.0366]
'Error'       [14.8813] [24] [0.6201] []      []
'Total'       [29.3355] [35] []      []      []
```

`stats =`

```
source: 'anova2'
sigmasq: 0.6201
colmeans: [-0.2958 0.0404 -0.6404 0.2944 0.4974 -0.2469]
coln: 6
rowmeans: [0.0401 -0.1571]
rown: 18
inter: 1
pval: 0.0366
df: 24
```

Figure No. 1: Two-way ANOVA

ANOVA Table					
Source	SS	df	MS	F	Prob>F
Columns	5.2429	5	1.04858	1.69	0.1751
Rows	0.3502	1	0.3502	0.56	0.4596
Interaction	8.8611	5	1.77222	2.86	0.0366
Error	14.8813	24	0.62005		
Total	29.3355	35			

anovan

Многофакторный дисперсионный анализ (MANOVA)

Синтаксис

```
p = anovan(X,group)
p = anovan(X,group,'model')
p = anovan(X,group,'model',sstype)
p = anovan(X,group,'model',sstype,gnames)
p =
anovan(X,group,'model',sstype,gnames,'displayopt')
[p,table] = anovan(...)
[p,table,stats] = anovan(...)
[p,table,stats,terms] = anovan(...)
```

Описание

`p = anovan(X,group)` функция позволяет провести многофакторный дисперсионный анализ для сравнения средних арифметических значений N выборок. Выборки задаются в виде вектора X . Факторы и их уровни определяются массивом ячеек `group`. Каждая ячейка аргумента `group` содержит список уровней факторов соответствующих наблюдениям в векторе X . Список в каждой ячейке может быть вектором, массивом символов, или строковым массивом ячеек, и должен содержать такое же количество элементов, что и вектор X .

Пример задания входных аргументов выборки X из 8 элементов и аргумента `group` для 3-х факторов изменяющихся на двух уровнях:

```
X = [x1 x2 x3 x4 x5 x6 x7 x8];
group = {[1 2 1 2 1 2 1 2]; ['hi';'hi';'lo';'lo';'hi';'hi';'lo';'lo']; {'may' 'may' 'may'
'may' 'june' 'june' 'june' 'june'}};
```

Уровни первого фактора кодируются целыми числами, второго \tilde{n} вектором строк, третьего \tilde{n} списком строк. Значение x_1 соответствует уровню первого фактора равного 1, второго - 'hi', третьего - 'may'.

Выходной вектор p содержит значения уровней значимости нулевой гипотезы относительно главных эффектов. Элемент $p(1)$ является уровнем значимости нулевой гипотезы, H_{0A} , состоящей в том, что выборки соответствующие уровням фактора A взяты из одной и той же генеральной совокупности. Элемент $p(2)$ является уровнем значимости нулевой гипотезы, H_{0B} , состоящей в том, что выборки соответствующие уровням фактора B взяты из одной и той же

генеральной совокупности. И так далее по всем остальным элементам вектора $p(i)$.

p является вероятностью ошибки первого рода, или вероятностью необоснованно отвергнуть нулевую гипотезу (H_0). H_0 может быть отвергнута, если p близко к нулю.

$p \approx 0$ для гипотезы H_{0A} означает, что хотя бы одно из средних арифметических, рассчитанных по выборкам соответствующим уровням фактора A , отличается от остальных значений, т.е. существует значимый главный эффект фактора A . $p \approx 0$ для гипотезы H_{0B} означает, что хотя бы одно из средних арифметических, рассчитанных по выборкам соответствующим уровням фактора B , отличается от остальных значений, т.е. существует значимый главный эффект фактора B .

Выбор критического уровня значимости $p_{кр}$ для условия принятия нулевой гипотезы

$$p \geq p_{кр},$$

предоставлен исследователю. В большинстве практических случаев $p_{кр}$ принимают равным 0,05; 0,01.

При проведении дисперсионного анализа общая дисперсия по умолчанию делится на:

- дисперсию, обусловленную изменениями уровней факторов для заданной регрессионной модели;
- остаточная дисперсия.

Таблица с результатами многофакторного дисперсионного анализа отображается в графическом окне.

Таблица с результатами двухфакторного дисперсионного анализа содержит 6 столбцов:

1. вид дисперсии (Source);
2. сумму квадратов разностей (SS) между средним арифметическим и значениями выборки по каждому виду дисперсии;
3. число степеней свободы по каждому виду дисперсии (df);
4. среднее значение суммы квадратов разностей (MS) по каждому виду дисперсии, определяемое как отношение SS/df ;
5. значение статистики Фишера (F статистики) для MS;

6. значение уровня значимости p ($\text{Prob}>F$) для рассчитанного значения статистики F .

$p = \text{anovan}(X, \text{group}, 'model')$ функция позволяет провести многофакторный дисперсионный анализ для заданной регрессионной модели. Выходной вектор p содержит результаты расчета уровней значимости для нулевых гипотез относительно факторов регрессионной модели. Вид регрессионной модели определяется входным аргументом *'model'*. Предусмотрены следующие виды регрессионных моделей: *'linear'* – линейная регрессионная модель; *'interaction'* – регрессионная модель, включающая эффекты парных взаимодействий; *'full'* – полная регрессионная модель, включающая главные эффекты и эффекты парных взаимодействий. Кроме указанных строковых значений, входной аргумент *'model'* может быть задан как целое число или вектор.

Если входной аргумент *'model'* задан как целое число равное k ($k > N$) функция anovan позволяет оценить значимость взаимодействий факторов до k уровня. Значения $k=1$ и $k=2$ соответствуют *'model'* равной *'linear'* и *'interaction'* соответственно. $k=N$ соответствует *'model'='full'*.

Использование вектора целых чисел при задании *'model'* позволяет оценить значимость заданных главных эффектов и взаимодействий факторов. Каждый элемент вектора *'model'* кодирует в десятичной форме двоичное число, определяющее главный эффект или эффект взаимодействия. Примеры кодирования главных эффектов или эффектов взаимодействия для 3 факторов приведены в следующей таблице

Двоичный аналог	Десятичный элемент вектора <i>'model'</i>	Вид эффекта при дисперсионном анализе
[0 0 1]	1	Главный эффект А
[0 1 0]	2	Главный эффект В
[1 0 0]	4	Главный эффект С
[0 1 1]	3	Эффект взаимодействия АВ
[1 1 0]	6	Эффект взаимодействия ВС
[1 0 1]	5	Эффект взаимодействия АС
[1 1 1]	7	Эффект взаимодействия АВС

Например, если *'model'=[2 4 6]* выходной вектор p содержит значения уровня значимости для проверки нулевых гипотез главных эффектов В и С и эффекта взаимодействия ВС в приведенном порядке. Одним из возможных способов задания вектора *'model'* является модификация выходной параметр *terms*. Вектор *terms* кодирует

главные эффекты и эффекты взаимодействий факторов согласно формату входного аргумента 'model'. Например, если после проведения дисперсионного анализа по трем факторам для 'model'=[2 4 6] эффект взаимодействий ВС оказался статистически незначим, то для коррекции главных эффектов В и С необходимо использовать 'model'=[2 4].

`p = anovan(X,group,'model',sstype)` функция позволяет провести многофакторный дисперсионный анализ для сравнения средних арифметических значений N выборок для заданного способа расчета суммы квадратов отклонений. Способ расчета суммы квадратов отклонений определяется входным аргументом sstype. Предусмотрено 3 способа расчета, sstype равен 1, 2, 3 соответственно. Значение по умолчанию sstype=3. Способ расчета суммы квадратов отклонений влияет на результаты многофакторного дисперсионного анализа для выборок неодинакового объема.

Сумма квадратов отклонений для какого либо эффекта фактора определяется при сравнении двух моделей. При первом способе расчета для model(i) SS является разницей остаточных сумм квадратов отклонений множества факторов model(j), где j=1..i, и текущего фактора. Второй способ расчета SS для model(i) представляет собой разность остаточной суммы квадратов отклонений главных эффектов за исключением model(i) и остаточной суммы квадратов отклонений главных эффектов с добавлением model(i). Третий способ расчета SS для model(i) представляет собой разность остаточной суммы квадратов отклонений для всех факторов в model за исключением model(i) и остаточной суммы квадратов отклонений по всем факторам в model.

Рассмотрим пример формирования сумм квадратов отклонений 1, 2, 3 типов для двух факторов А и В. Модель может содержать два главных эффекта факторов А и В и эффект взаимодействия АВ. Порядок эффектов в векторе 'model' соответствует последовательности: А, В, АВ. Величина $R(\sum)$ представляет остаточную сумму квадратов отклонений для модели, например $R(A,B,AB)$ - остаточная сумма квадратов соответствующая полной модели, $R(A)$ - остаточная сумма квадратов соответствующая главному эффекту фактора А, $R(1)$ - остаточная сумма квадратов соответствующая общему среднему арифметическому. Три способа расчета суммы квадратов для рассматриваемого примера приведены в следующей таблице:

Эффект	1 способ (sstype=1)	2 способ (sstype=2)	3 способ (sstype=3)
А	$R(1)-R(A)$	$R(B)-R(A,B)$	$R(B,AB)-R(A,B,AB)$

B	$R(A)-R(A,B)$	$R(A)-R(A,B)$	$R(A,AB)-R(A,B,AB)$
AB	$R(A,B)-R(A,B,AB)$	$R(A,B)-R(A,B,AB)$	$R(A,B)-R(A,B,AB)$

Модели при расчете SS по третьему способу имеют наложенное ограничение по среднему квадратическому отклонению. Это означает, например для остаточной суммы квадратов $R(B,AB)$, массив эффектов AB ограничен по сумме от 0 до A для каждого значения B и до B для каждого значения A.

`p = anovan(X, group, 'model', sstype, gnames)` входной аргумент `gnames` служит для задания меток для N факторов в таблице дисперсионного анализа. `gnames` может быть задан как вектор строк строковый массив ячеек, где каждый элемент массива соответствует одному наблюдению. Если входной аргумент `gnames` не задан, в качестве меток по умолчанию принимаются: 'X1', 'X2', 'X3', ..., 'XN'.

`p = anovan(X, group, 'model', sstype, gnames, 'displayopt')` входной аргумент `'displayopt'` позволяет отобразить графическое окно с таблицей результатов дисперсионного анализа `'displayopt'='on'`, или подавить вывод графического окна `'displayopt'='off'`. Значение по умолчанию `'displayopt'='on'`.

`[p, table] = anovan(...)` функция возвращает таблицу с результатами многофакторного дисперсионного анализа, включая метки факторов, в текстовой форме в командном окне MATLAB.

`[p, table, stats] = anovan(...)` функция возвращает структуру `stats` используемую для проведения попарных сравнений средних арифметических выборок. Проверка параметрической гипотезы о равенстве двух средних арифметических выполняется при помощи функции `multcompare`. Структура данных `stats` передается функции `multcompare` как входной аргумент.

`[p, table, stats, terms] = anovan(...)` функция возвращает вектор номеров главных эффектов и эффектов взаимодействий `terms`, использовавшихся при проведении многофакторного дисперсионного анализа. Формат вектора `terms` соответствует формату вектора `'model'`. Если при вызове `anovan` был использован вектор `'model'`, то `terms='model'`.

Примеры использования функции многофакторного дисперсионного анализа

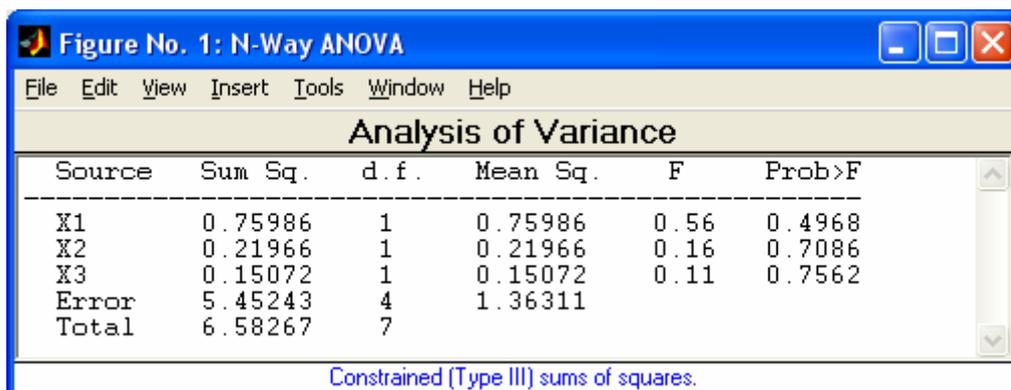
Дисперсионный анализ для 3-х факторов изменяющихся двух уровнях.

```
>> X=normrnd(0,1,8,1)
X =
```

```

1.0668
0.0593
-0.0956
-0.8323
0.2944
-1.3362
0.7143
1.6236
>> group = {[1 2 1 2 1 2 1 2]; ['hi';'hi';'lo';'lo';'hi';'hi';'lo';'lo']; {'may' 'may'
'may' 'may' 'june' 'june' 'june' 'june'}}
group =
    [1x8 double]
    [8x2 char ]
    {1x8 cell }
>> p = anovan(X,group)
p =
    0.4968
    0.7086
    0.7562

```

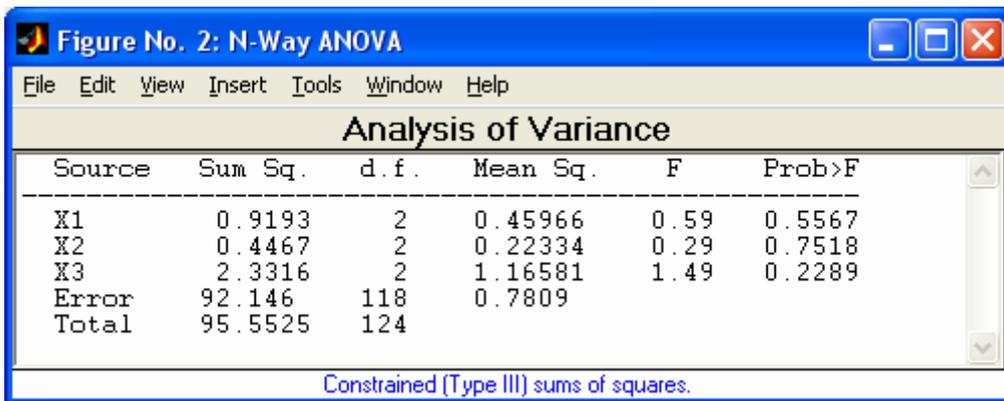


Дисперсионный анализ для 5-ти факторов изменяющихся 3-х уровнях.

```

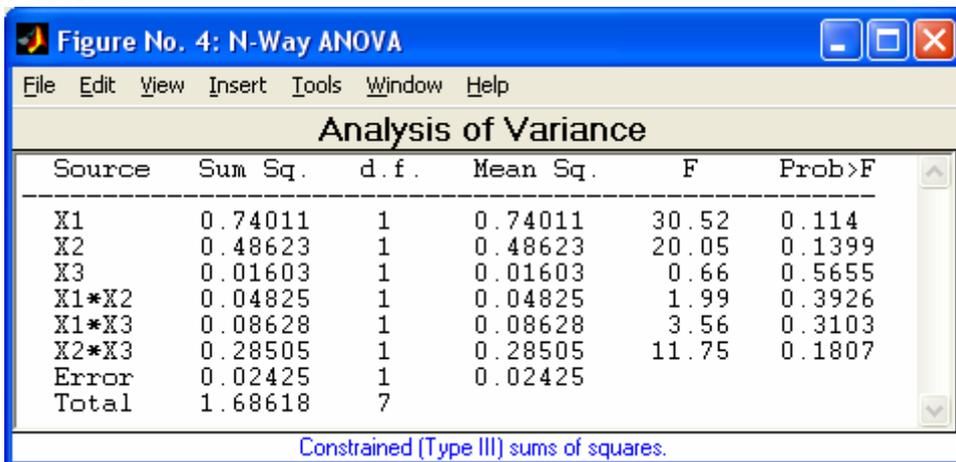
>> X=normrnd(0,1,125,1)
>> group = unidrnd(3,125,3)
>> group = {group (1:125,1); group (1:125,2); group (1:125,3)}
>> p = anovan(X,group)
p =
    0.5567
    0.7518
    0.2289

```



Дисперсионный анализ для 3-х факторов, изменяющихся двух уровнях, и модели, содержащей эффекты взаимодействий факторов.

```
>> X=normrnd(0,1,8,1);
>> group = {[1 2 1 2 1 2 1 2]; ['hi';'hi';'lo';'lo';'hi';'hi';'lo';'lo']; {'may' 'may'
'may' 'may' 'june' 'june' 'june' 'june'}};
>> model='interaction';
>> p = anovan(X,group,model)
p =
0.1140
0.1399
0.5655
0.3926
0.3103
0.1807
```



Дисперсионный анализ для 3-х факторов: A, B и C, изменяющихся двух уровнях, и $model=[2\ 4\ 6]$ для проверки нулевых гипотез относительно главных эффектов B и C и эффекта взаимодействия BC.

```
>> X=normrnd(0,1,8,1);
>> group = {[1 2 1 2 1 2 1 2]; ['hi';'hi';'lo';'lo';'hi';'hi';'lo';'lo']; {'may' 'may'
'may' 'may' 'june' 'june' 'june' 'june'}};
>> model=[2 4 6];
```

```
>> p = anovan(X,group,model)
```

```
p =
```

```
0.7621
```

```
0.7667
```

```
0.4625
```

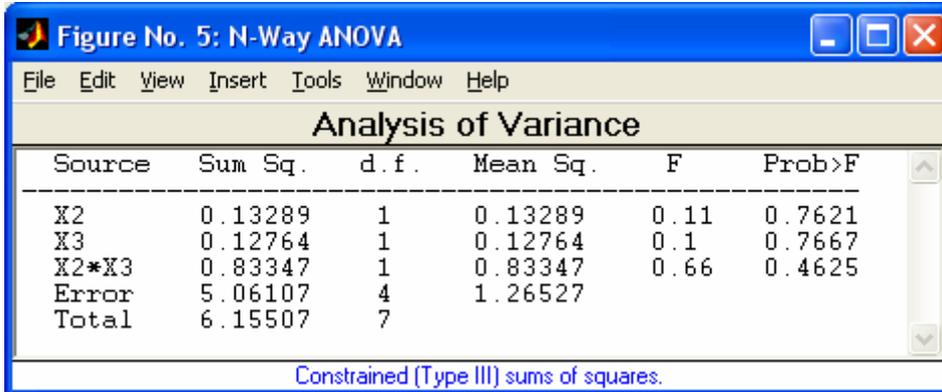


Figure No. 5: N-Way ANOVA

File Edit View Insert Tools Window Help

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X2	0.13289	1	0.13289	0.11	0.7621
X3	0.12764	1	0.12764	0.1	0.7667
X2*X3	0.83347	1	0.83347	0.66	0.4625
Error	5.06107	4	1.26527		
Total	6.15507	7			

Constrained (Type III) sums of squares.

Дисперсионный анализ для 3-х факторов: А, В и С, изменяющихся двух уровнях, и $model=[2\ 4\ 6]$ для проверки нулевых гипотез относительно главных эффектов В и С и эффекта взаимодействия ВС. И второго способа расчета суммы квадратов разностей (SS) по каждому виду дисперсии.

```
>> X=normrnd(0,1,8,1);
```

```
>> group = {[1 2 1 2 1 2 1 2]; ['hi';'hi';'lo';'lo';'hi';'hi';'lo';'lo']; {'may' 'may'  
'may' 'may' 'june' 'june' 'june' 'june'}};
```

```
>> model=[2 4 6];
```

```
>> sstype = 2;
```

```
>> p = anovan(X,group,model, sstype)
```

```
p =
```

```
0.0810
```

```
0.4209
```

```
0.5171
```

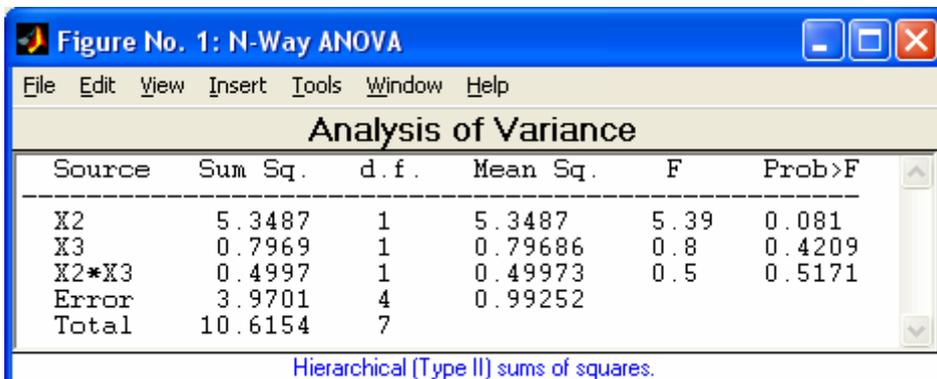


Figure No. 1: N-Way ANOVA

File Edit View Insert Tools Window Help

Analysis of Variance

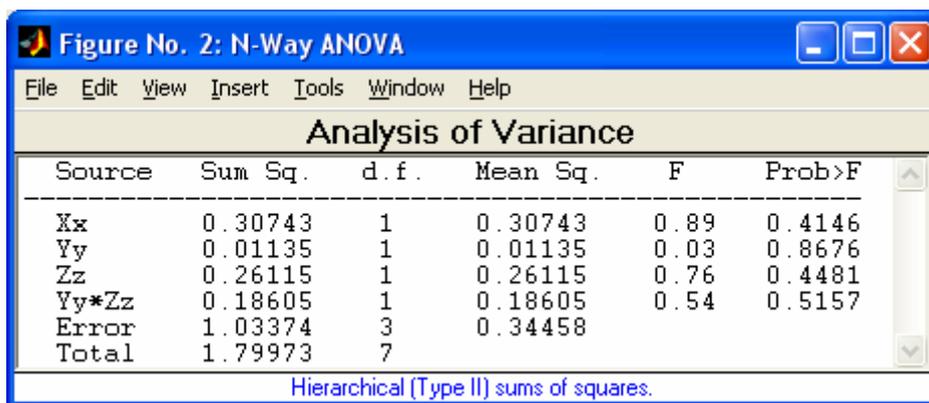
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X2	5.3487	1	5.3487	5.39	0.081
X3	0.7969	1	0.79686	0.8	0.4209
X2*X3	0.4997	1	0.49973	0.5	0.5171
Error	3.9701	4	0.99252		
Total	10.6154	7			

Hierarchical (Type II) sums of squares.

Дисперсионный анализ для 3-х факторов: А, В и С, изменяющихся двух уровнях, и $model=[1\ 2\ 4\ 6]$ для проверки нулевых гипотез

относительно главных эффектов A, B, C и эффекта взаимодействия BC. И второго способа расчета суммы квадратов разностей (SS) по каждому виду дисперсии. В качестве меток используется массив ячеек `gnames = {'Xx'; 'Yy'; 'Zz'}`.

```
>> X=normrnd(0,1,8,1);
>> group = {[1 2 1 2 1 2 1 2]; ['hi';'hi';'lo';'lo';'hi';'hi';'lo';'lo']; {'may' 'may'
'may' 'may' 'june' 'june' 'june' 'june'}};
>> model=[1 2 4 6];
>> sstype = 2;
>> gnames = {'Xx'; 'Yy'; 'Zz'};
>> p = anovan(X,group,model,sstype,gnames)
p =
    0.4146
    0.8676
    0.4481
    0.5157
```



Дисперсионный анализ для 3-х факторов: A, B и C, изменяющихся двух уровнях, и модели, содержащей эффекты взаимодействий факторов. И второго способа расчета суммы квадратов разностей (SS) по каждому виду дисперсии. В качестве меток используется массив ячеек `gnames = {'Xx'; 'Yy'; 'Zz'}`. Кроме уровней значимости для указанных нулевых гипотез, выводятся таблица многофакторного дисперсионного анализа, структура для проведения попарного сравнения средних, вектор использованных факторов и их взаимодействий.

```
>> X=normrnd(0,1,8,1);
>> group = {[1 2 1 2 1 2 1 2]; ['hi';'hi';'lo';'lo';'hi';'hi';'lo';'lo']; {'may' 'may'
'may' 'may' 'june' 'june' 'june' 'june'}};
>> model='interaction';
>> sstype = 2;
>> gnames = {'Xx'; 'Yy'; 'Zz'};
>> [p,table,stats,terms] = anovan(X,group,model,sstype,gnames)
```

p =

0.2308
0.2221
0.1346
0.0714
0.1167
0.3139

table =

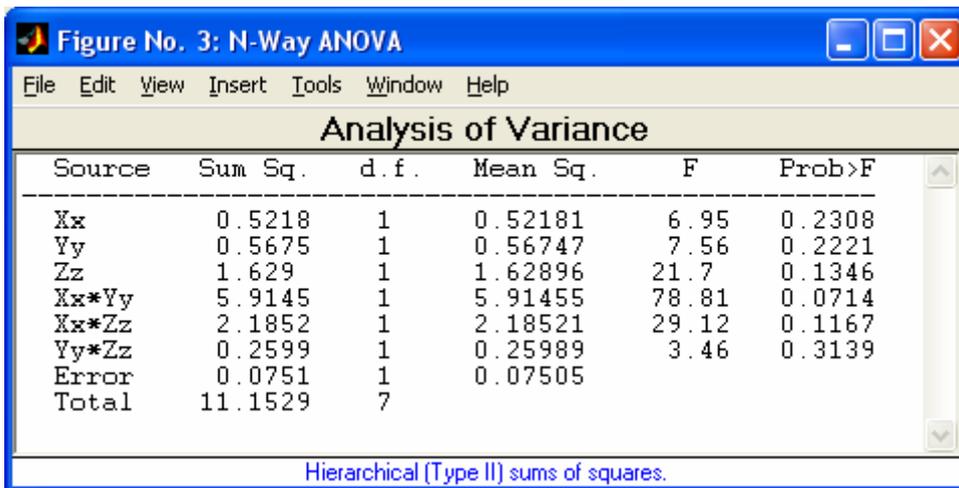
'Source'	'Sum Sq.'	'd.f.'	'Singular?'	
'Mean Sq.'	'F'	'Prob>F'		
'Xx'	[0.5218]	[1]	[0]	[]
0.5218	[6.9526]	[0.2308]		
'Yy'	[0.5675]	[1]	[0]	[]
0.5675	[7.5609]	[0.2221]		
'Zz'	[1.6290]	[1]	[0]	[]
1.6290	[21.7042]	[0.1346]		
'Xx*Yy'	[5.9145]	[1]	[0]	[]
5.9145	[78.8050]	[0.0714]		
'Xx*Zz'	[2.1852]	[1]	[0]	[]
2.1852	[29.1156]	[0.1167]		
'Yy*Zz'	[0.2599]	[1]	[0]	[]
0.2599	[3.4627]	[0.3139]		
'Error'	[0.0751]	[1]	[0]	[]
0.0751	[]	[]		
'Total'	[11.1529]	[7]	[0]	[]
[]	[]	[]		

stats =

source: 'anovan'
coeffs: [7x1 double]
Rtr: [7x7 double]
dfe: 1
mse: 0.0751
permvec: [1 2 3 4 5 6 7]
terms: [6x1 double]
nlevels: [3x1 double]
termcols: [7x1 double]

terms =

1
2
4
3
5
6



Планирование эксперимента

ff2n

Генерация матрицы полного факторного эксперимента для факторов изменяющихся на 2-х уровнях

Синтаксис

```
X = ff2n(n)
```

Описание

$X = \text{ff2n}(n)$ функция позволяет получить матрицу плана X полного факторного эксперимента при условии, что все факторы входящие в план изменяются на двух уровнях. Входной аргумент n — количество факторов в эксперименте. Размерность матрицы плана эксперимента равна $2^n \times n$, где 2^n — число строк или число опытов в эксперименте, n — число столбцов. Элементы матрицы плана эксперимента выражаются в кодированной форме и равны 0 или 1. 0 соответствует минимальному значению фактора, 1 — максимальному значению фактора. Матрица плана эксперимента X не рандомизирована и представляет собой последовательный перебор всех возможных комбинаций значений факторов.

Примеры использования функции генерации полного факторного эксперимента для факторов, изменяющихся на 2-х уровнях

Матрица плана полного 5-ти факторного эксперимента

```
>> n=5
```

```
n =
```

```
5
```

```
>> X = ff2n(n)
```

```
X =
```

```
0 0 0 0 0
0 0 0 0 1
0 0 0 1 0
0 0 0 1 1
0 0 1 0 0
0 0 1 0 1
0 0 1 1 0
0 0 1 1 1
0 1 0 0 0
0 1 0 0 1
0 1 0 1 0
0 1 0 1 1
0 1 1 0 0
```

```

0 1 1 0 1
0 1 1 1 0
0 1 1 1 1
1 0 0 0 0
1 0 0 0 1
1 0 0 1 0
1 0 0 1 1
1 0 1 0 0
1 0 1 0 1
1 0 1 1 0
1 0 1 1 1
1 1 0 0 0
1 1 0 0 1
1 1 0 1 0
1 1 0 1 1
1 1 1 0 0
1 1 1 0 1
1 1 1 1 0
1 1 1 1 1

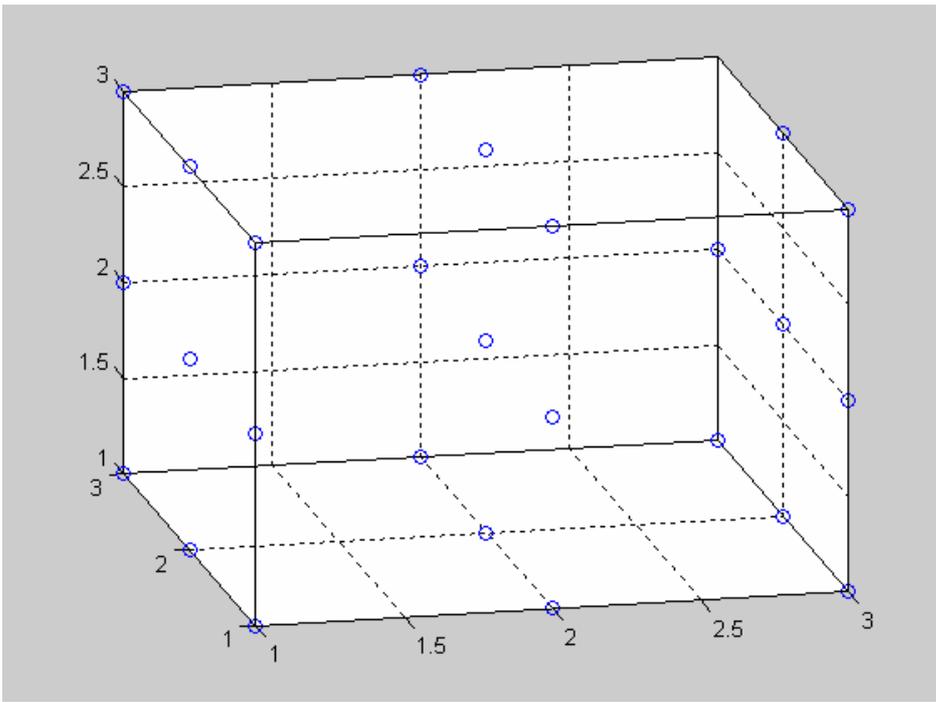
```

Графически матрицу планирования полного факторного эксперимента для трех факторов можно представить следующим образом

```

>> n=3;
>> X = ff2n(n)
X =
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1
>> x= X(:,1);
>> y= X(:,2);
>> z= X(:,3);
>> plot3(x,y,z,'o')
>> grid on

```



fullfact

Генерация матрицы полного факторного эксперимента для произвольного числа и количества уровней факторов

Синтаксис

```
design = fullfact(levels)
```

Описание

`design = fullfact(levels)` функция предназначена для генерации матрицы полного факторного эксперимента `design` при произвольном числе факторов и заданном количестве уровней для каждого фактора. Количество факторов и число уровней определяется вектором `levels` в формате `levels=[n m k ...]`, где `n`, `m`, `k` и т.д. – число уровней первого, второго, третьего факторов и т.д. Элементы матрицы плана эксперимента представляются кодированными значениями и соответственно равны `1..n`, `1..m`, `1..k` и т.д. `1` соответствует минимальному значению фактора и далее по возрастанию. Матрица плана эксперимента не рандомизирована и представляет собой последовательный перебор всех возможных комбинаций значений факторов.

Примеры использования функции генерации полного факторного эксперимента для произвольного числа и количества уровней факторов

Генерация матрицы плана эксперимента для трех факторов изменяющихся на 3 уровнях.

```
>> levels=[3 3 3]
levels =
     3     3     3
>> design = fullfact(levels)
design =
     1     1     1
     2     1     1
     3     1     1
     1     2     1
     2     2     1
     3     2     1
     1     3     1
     2     3     1
     3     3     1
     1     1     2
     2     1     2
```

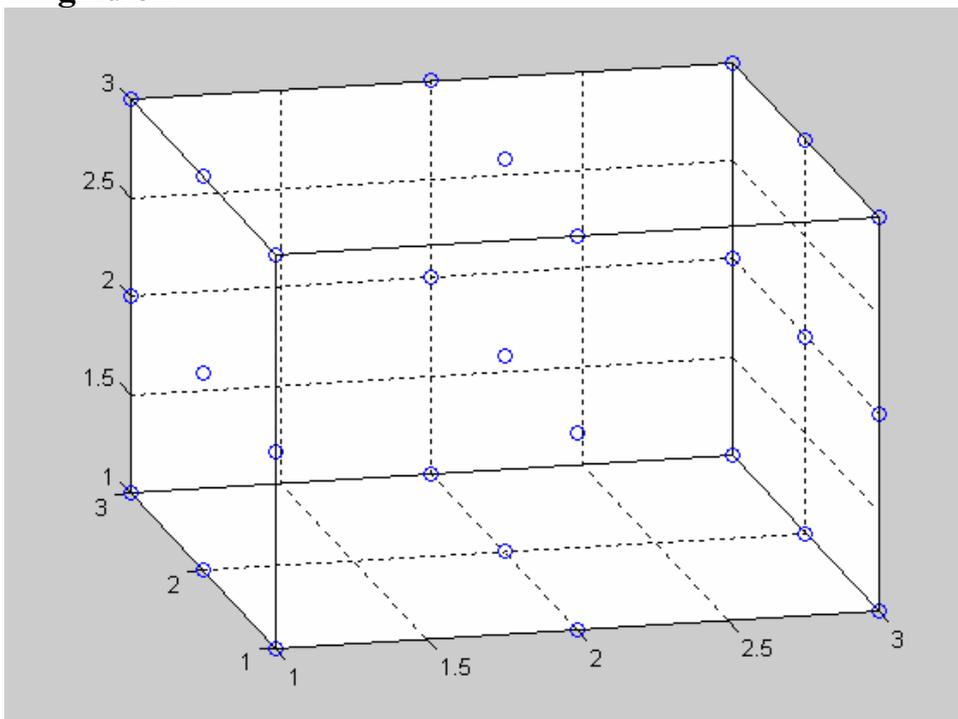
3	1	2
1	2	2
2	2	2
3	2	2
1	3	2
2	3	2
3	3	2
1	1	3
2	1	3
3	1	3
1	2	3
2	2	3
3	2	3
1	3	3
2	3	3
3	3	3

Графически матрицу планирования эксперимента для трех факторов изменяющихся на трех уровнях можно представить следующим образом

```

>> levels=[3 3 3];
>> design = fullfact(levels);
>> x=design(:,1);
>> y=design(:,2);
>> z=design(:,3);
>> plot3(x,y,z,'o')
>> grid on

```

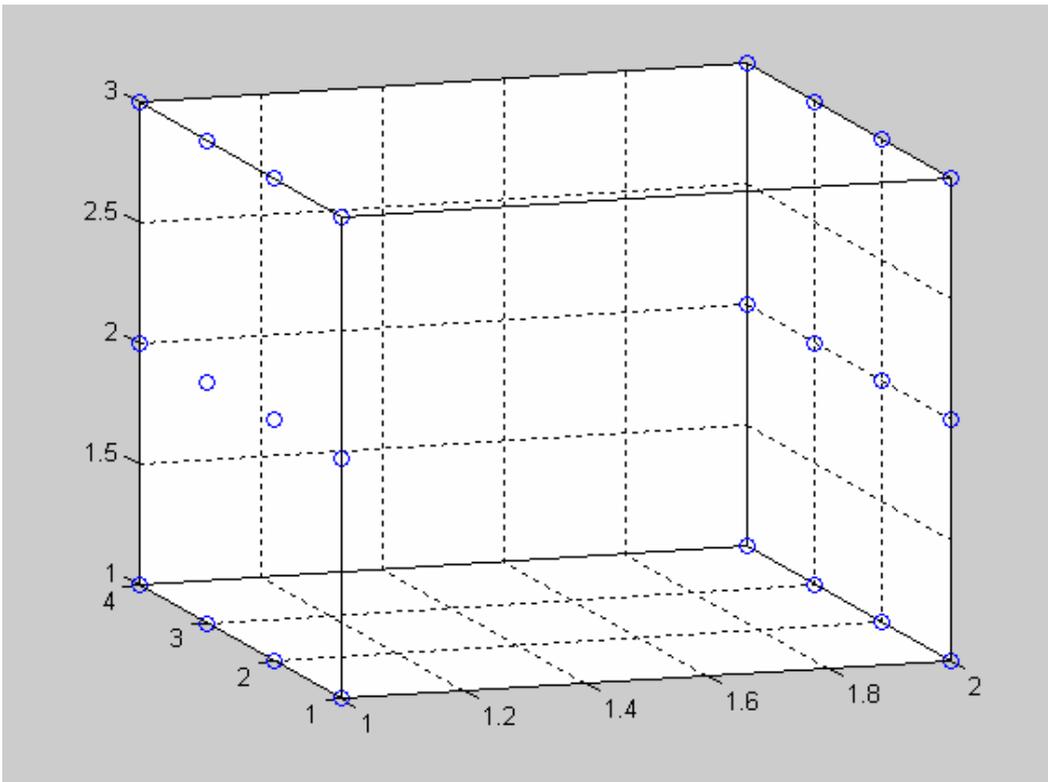


Генерация матрицы плана эксперимента для трех факторов изменяющихся на $2 \times 4 \times 3$ уровнях соответственно

```
>> levels=[2 4 3]
levels =
     2     4     3
>> design = fullfact(levels)
design =
     1     1     1
     2     1     1
     1     2     1
     2     2     1
     1     3     1
     2     3     1
     1     4     1
     2     4     1
     1     1     2
     2     1     2
     1     2     2
     2     2     2
     1     3     2
     2     3     2
     1     4     2
     2     4     2
     1     1     3
     2     1     3
     1     2     3
     2     2     3
     1     3     3
     2     3     3
     1     4     3
     2     4     3
```

Графически матрицу планирования эксперимента для трех факторов изменяющихся на $2 \times 4 \times 3$ уровнях можно представить следующим образом

```
>> levels=[2 4 3]
>> design = fullfact(levels)
>> x=design(:,1);
>> y=design(:,2);
>> z=design(:,3);
>> plot3(x,y,z,'o')
>> grid on
```



fracfact

Генерация матрицы дробного факторного эксперимента для произвольного числа факторов изменяющихся на двух уровнях

Синтаксис

```
x = fracfact('gen')
[x,conf] = fracfact('gen')
```

Описание

`x = fracfact('gen')` служит для генерации матрицы `x` плана дробного факторного эксперимента согласно генератору `'gen'`. Значения факторов в матрице плана эксперимента `x` равны 1 и -1. 1 соответствует максимальному уровню фактора, -1 – минимальному уровню. Генератор `'gen'` является строковой переменной, состоящей из подстрок разделенных пробелами. Подстрока может состоять из одной и более букв латинского алфавита. В первую очередь задаются односимвольные подстроки, кодирующие факторы для матрицы полного факторного эксперимента. Эта матрица является основой дробного факторного плана. Правила определения значений остальных факторов в матрице дробного плана задаются в виде сочетаний символов, кодирующих факторы полного факторного эксперимента. Для этих факторов значения определяются как произведения значений кодированных переменных, соответствующих первой группе факторов. Например, для генератора `'gen'='a b ab'` первые два фактора: `a` и `b`, образуют матрицу полного факторного плана вида:

```
>> x = ff2n(2)
```

```
x =
    0    0
    0    1
    1    0
    1    1
```

Подстрока `ab` определяет значение третьего фактора как построчное произведение `a` и `b`.

```
>> x = fracfact('a b ab')
```

```
x =
   -1   -1    1
   -1    1   -1
    1   -1   -1
    1    1    1
```

Для факторов второй группы комбинация символов в подстроке определяют вид смешивания их главных эффектов. Т.е., в предыдущем примере главный эффект третьего фактора β_3 будет смешан с парным эффектом первых двух факторов β_{12} :

$$\beta_3 \rightarrow \beta_3 + \beta_{12}.$$

Матрица дробного факторного плана x является частью матрицы полного факторного плана эксперимента образованного всеми факторами. Число строк матрицы плана x равно 2^n , где n — количество односимвольных подстрок в генераторе. Количество столбцов в матрице плана должно быть равно числу подстрок в генераторе.

`[x,conf] = fracfact('gen')` этот вариант синтаксиса функции возвращает матрицу `conf` показывающую способ смешивания главных эффектов факторов и их парных взаимодействий.

Примеры использования функции генерации матрицы дробного факторного эксперимента

Пример 1

Целью эксперимента является оценка величин главных (линейных) эффектов 4 факторов на функцию отклика при проведении восьми опытов. Каждый опыт проводится без повторений.

Проведение полного факторного эксперимента потребует $4^2=16$ опытов. Однако, если на основании априорных данных предположить незначимость тройных взаимодействий, то можно оценить величины главных эффектов четырех факторов в восьми опытах.

Матрица планирования дробного факторного эксперимента x в этом случае примет вид

```
>> [x,conf] = fracfact('a b c abc')
```

```
x =
```

```
-1 -1 -1 -1
-1 -1  1  1
-1  1 -1  1
-1  1  1 -1
 1 -1 -1  1
 1 -1  1 -1
 1  1 -1 -1
 1  1  1  1
```

```
conf =
```

```
'Term'  'Generator'  'Confounding'
```

'X1'	'a'	'X1'
'X2'	'b'	'X2'
'X3'	'c'	'X3'
'X4'	'abc'	'X4'
'X1*X2'	'ab'	'X1*X2 + X3*X4'
'X1*X3'	'ac'	'X1*X3 + X2*X4'
'X1*X4'	'bc'	'X1*X4 + X2*X3'
'X2*X3'	'bc'	'X1*X4 + X2*X3'
'X2*X4'	'ac'	'X1*X3 + X2*X4'
'X3*X4'	'ab'	'X1*X2 + X3*X4'

Первые 3 столбца матрицы x представляют матрицу полного факторного эксперимента для трех факторов a , b , c . Четвертый столбец получен построчным перемножением значений первых трех факторов. Матрица $conf$ показывает, что такое планирование эксперимента позволяет оценить не смешанные главные эффекты для всех четырех факторов. Парные взаимодействия смешаны друг с другом. Например, эффект от парного взаимодействия первого и второго факторов смешан с эффектом парного взаимодействия третьего и четвертого факторов: $X1*X2 + X3*X4$. Это не позволяет отдельно оценить их эффекты.

В случае, если после проведения указанных восьми опытов выяснится что суммарный эффект от взаимодействия $X1*X2+X3*X4$ значим, то целесообразно определить какое парное взаимодействие значимо $X1*X2$ или $X3*X4$. Для этого дробную реплику из первых восьми опытов можно дополнить второй репликой до полного факторного эксперимента. Матрица второй реплики задается тем же генератором, но с обратным знаком для четвертого фактора:

```
>> fracfact('a b c -abc')
```

```
ans =
```

```
-1 -1 -1 1
-1 -1 1 -1
-1 1 -1 -1
-1 1 1 1
1 -1 -1 -1
1 -1 1 1
1 1 -1 1
1 1 1 -1
```

Пример 2

Необходимо оценить величины линейных эффектов восьми факторов. Поскольку в большинстве случаев парные эффекты взаимодействия больше эффектов взаимодействия высших порядков, то необходимо

так составить дробную реплику, чтобы линейные эффекты не были смешаны с ними при минимальном количестве опытов. Поведение полного факторного эксперимента потребовало бы $8^2=256$ опытов. Использование дробной реплики полного факторного эксперимента позволяет при 16 опытах получить оценки главных эффектов 8 факторов не смешанных с парными взаимодействиями. Реплику 16/256 для названных условий можно получить с использованием следующего генератора

```
>> [x, conf] = fracfact('a b c d abc acd abd bcd')
```

```
x =
```

```
-1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 1 -1 1 1 1
-1 -1 1 -1 1 1 -1 1
-1 -1 1 1 1 -1 1 -1
-1 1 -1 -1 1 -1 1 1
-1 1 -1 1 1 1 -1 -1
-1 1 1 -1 -1 1 1 -1
-1 1 1 1 -1 -1 -1 1
1 -1 -1 -1 1 1 1 -1
1 -1 -1 1 1 -1 -1 1
1 -1 1 -1 -1 -1 1 1
1 -1 1 1 -1 1 -1 -1
1 1 -1 -1 -1 1 -1 1
1 1 -1 1 -1 -1 1 -1
1 1 1 -1 1 -1 -1 -1
1 1 1 1 1 1 1 1
```

```
conf =
```

Term	Generator	Confounding
X1	a	X1
X2	b	X2
X3	c	X3
X4	d	X4
X5	abc	X5
X6	acd	X6
X7	abd	X7
X8	bcd	X8
X1*X2	ab	X1*X2 + X3*X5 + X4*X7 + X6*X8
X1*X3	ac	X1*X3 + X2*X5 + X4*X6 + X7*X8
X1*X4	ad	X1*X4 + X2*X7 + X3*X6 + X5*X8
X1*X5	bc	X1*X5 + X2*X3 + X4*X8 + X6*X7
X1*X6	cd	X1*X6 + X2*X8 + X3*X4 + X5*X7
X1*X7	bd	X1*X7 + X2*X4 + X3*X8 + X5*X6
X1*X8	abcd	X1*X8 + X2*X6 + X3*X7 + X4*X5
X2*X3	bc	X1*X5 + X2*X3 + X4*X8 + X6*X7

X2*X4	bd	X1*X7 + X2*X4 + X3*X8 + X5*X6
X2*X5	ac	X1*X3 + X2*X5 + X4*X6 + X7*X8
X2*X6	abcd	X1*X8 + X2*X6 + X3*X7 + X4*X5
X2*X7	ad	X1*X4 + X2*X7 + X3*X6 + X5*X8
X2*X8	cd	X1*X6 + X2*X8 + X3*X4 + X5*X7
X3*X4	cd	X1*X6 + X2*X8 + X3*X4 + X5*X7
X3*X5	ab	X1*X2 + X3*X5 + X4*X7 + X6*X8
X3*X6	ad	X1*X4 + X2*X7 + X3*X6 + X5*X8
X3*X7	abcd	X1*X8 + X2*X6 + X3*X7 + X4*X5
X3*X8	bd	X1*X7 + X2*X4 + X3*X8 + X5*X6
X4*X5	abcd	X1*X8 + X2*X6 + X3*X7 + X4*X5
X4*X6	ac	X1*X3 + X2*X5 + X4*X6 + X7*X8
X4*X7	ab	X1*X2 + X3*X5 + X4*X7 + X6*X8
X4*X8	bc	X1*X5 + X2*X3 + X4*X8 + X6*X7
X5*X6	bd	X1*X7 + X2*X4 + X3*X8 + X5*X6
X5*X7	cd	X1*X6 + X2*X8 + X3*X4 + X5*X7
X5*X8	ad	X1*X4 + X2*X7 + X3*X6 + X5*X8
X6*X7	bc	X1*X5 + X2*X3 + X4*X8 + X6*X7
X6*X8	ab	X1*X2 + X3*X5 + X4*X7 + X6*X8
X7*X8	ac	X1*X3 + X2*X5 + X4*X6 + X7*X8

Матрица conf показывает, что линейные эффекты восьми факторов не смешаны с парными взаимодействиями. Парные взаимодействия смешаны друг с другом. Применение других генераторов не всегда позволяет достичь такого смешивания, например

```
>> [x, conf] = fracfact('a b c d ab cd ad bc')
```

```
x =
```

```

-1 -1 -1 -1 1 1 1 1
-1 -1 -1 1 1 -1 -1 1
-1 -1 1 -1 1 -1 1 -1
-1 -1 1 1 1 1 -1 -1
-1 1 -1 -1 -1 1 1 -1
-1 1 -1 1 -1 -1 -1 -1
-1 1 1 -1 -1 -1 1 1
-1 1 1 1 -1 1 -1 1
1 -1 -1 -1 -1 1 -1 1
1 -1 -1 1 -1 -1 1 1
1 -1 1 -1 -1 -1 -1 -1
1 -1 1 1 -1 1 1 -1
1 1 -1 -1 1 1 -1 -1
1 1 -1 1 1 -1 1 -1
1 1 1 -1 1 -1 -1 1
1 1 1 1 1 1 1 1

```

```
conf =
```

```
'Term' 'Generator' 'Confounding'
```

'X1'	'a'	'X1 + X2*X5 + X4*X7'
'X2'	'b'	'X2 + X1*X5 + X3*X8'
'X3'	'c'	'X3 + X2*X8 + X4*X6'
'X4'	'd'	'X4 + X1*X7 + X3*X6'
'X5'	'ab'	'X5 + X1*X2'
'X6'	'cd'	'X6 + X3*X4'
'X7'	'ad'	'X7 + X1*X4'
'X8'	'bc'	'X8 + X2*X3'
'X1*X2'	'ab'	'X5 + X1*X2'
'X1*X3'	'ac'	'X1*X3 + X5*X8 + X6*X7'
'X1*X4'	'ad'	'X7 + X1*X4'
'X1*X5'	'b'	'X2 + X1*X5 + X3*X8'
'X1*X6'	'acd'	'X1*X6 + X3*X7'
'X1*X7'	'd'	'X4 + X1*X7 + X3*X6'
'X1*X8'	'abc'	'X1*X8 + X3*X5'
'X2*X3'	'bc'	'X8 + X2*X3'
'X2*X4'	'bd'	'X2*X4 + X5*X7 + X6*X8'
'X2*X5'	'a'	'X1 + X2*X5 + X4*X7'
'X2*X6'	'bcd'	'X2*X6 + X4*X8'
'X2*X7'	'abd'	'X2*X7 + X4*X5'
'X2*X8'	'c'	'X3 + X2*X8 + X4*X6'
'X3*X4'	'cd'	'X6 + X3*X4'
'X3*X5'	'abc'	'X1*X8 + X3*X5'
'X3*X6'	'd'	'X4 + X1*X7 + X3*X6'
'X3*X7'	'acd'	'X1*X6 + X3*X7'
'X3*X8'	'b'	'X2 + X1*X5 + X3*X8'
'X4*X5'	'abd'	'X2*X7 + X4*X5'
'X4*X6'	'c'	'X3 + X2*X8 + X4*X6'
'X4*X7'	'a'	'X1 + X2*X5 + X4*X7'
'X4*X8'	'bcd'	'X2*X6 + X4*X8'
'X5*X6'	'abcd'	'X5*X6 + X7*X8'
'X5*X7'	'bd'	'X2*X4 + X5*X7 + X6*X8'
'X5*X8'	'ac'	'X1*X3 + X5*X8 + X6*X7'
'X6*X7'	'ac'	'X1*X3 + X5*X8 + X6*X7'
'X6*X8'	'bd'	'X2*X4 + X5*X7 + X6*X8'
'X7*X8'	'abcd'	'X5*X6 + X7*X8'

Как следует из анализа матрицы conf использование генератора 'a b c d ab cd ad bc' приводит к получению главных эффектов смешанных с парными взаимодействиями.

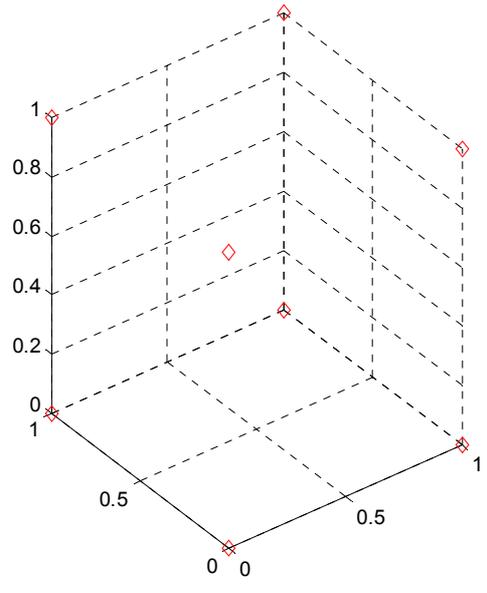
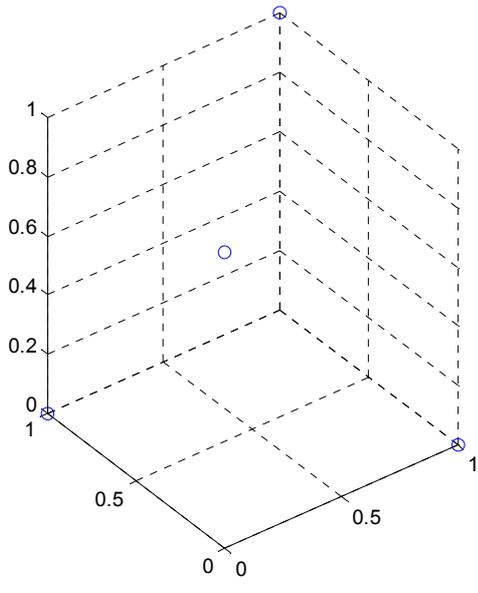
Графическое сравнение планов дробного и полного факторных экспериментов для трех факторов изменяющихся на двух уровнях.

```
>> n=3;
>> X = fracfact('a b ab')
```

```

X =
  -1  -1   1
  -1   1  -1
   1  -1  -1
   1   1   1
>> X = (X + 1)/2
X =
   0   0   1
   0   1   0
   1   0   0
   1   1   1
>> x=X(:,1);
>> y=X(:,2);
>> z=X(:,3);
>> subplot(1,2,1)
>> plot3(x,y,z,'o')
>> grid on
>> Y = ff2n(n)
Y =
   0   0   0
   0   0   1
   0   1   0
   0   1   1
   1   0   0
   1   0   1
   1   1   0
   1   1   1
>> x1= Y (:,1);
>> y1= Y (:,2);
>> z1= Y (:,3);
>> subplot(1,2,2)
>> plot3(x1,y1,z1,'dr')
>> grid on

```



hadamard

Генерация матрицы Адамара

Синтаксис

```
H = hadamard(n)
```

Описание

`H = hadamard(n)` функция предназначена для генерации матрицы Адамара n -го порядка.

Матрица Адамара H — это матрица с ортогональными столбцами, элементами которой являются 1 и -1, для которой должно выполняться условие

$$H' \cdot H = n \cdot I,$$

где $[n \ n] = \text{size}(H)$, $I = \text{eye}(n,n)$.

Матрица Адамара размерностью $n \times n$, при $n > 2$, существует, если $\text{rem}(n,4) = 0$. Для входного аргумента n должно выполняться условие - n , $n/12$, $n/20$ являются степенью 2.

Примеры использования функции генерации матрицы Адамара

Генерация матрицы Адамара с размерностью 4×4

```
>> H = hadamard(4)
```

```
H =
```

```
 1  1  1  1
 1 -1  1 -1
 1  1 -1 -1
 1 -1 -1  1
```

Проверка выполнения условия $H' \cdot H = n \cdot I$,

```
>> n=4
```

```
n =
```

```
 4
```

```
>> H = hadamard(4);
```

```
>> I = eye(n,n)
```

```
I =
```

```
 1  0  0  0
 0  1  0  0
 0  0  1  0
 0  0  0  1
```

```
>> I*4==H'*H
```

ans =

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

bbdesign

Генерация матрицы плана Бокса-Бенкина

Синтаксис

```
D = bbdesign(nfactors)
D =
bbdesign(nfactors, 'pname1', pvalue1, 'pname2', pvalue2, .
..)
[D,blk] = bbdesign(...)
```

Описание

`D = bbdesign(nfactors)` функция позволяет получить матрицу `D` плана Бокса-Бенкина для `nfactors` факторов. Размерность матрицы плана эксперимента `D` равна `nfactors` × `nfactors`. Количество опытов в матрице равно числу факторов. Элементами матрицы `D` являются 1 и -1 кодирующие максимальные и минимальные значения факторов соответственно.

`[D,blk] = bbdesign(nfactors)` в отличии от первого варианта синтаксиса функция возвращает вектор `blk` номеров блоков опытов. Блоки позволяют сгруппировать опыты таким образом, чтобы минимизировать влияние изменений внешних условий на значения оцениваемых параметров.

`[...] = bbdesign(nfactors, 'pname1', pvalue1, 'pname2', pvalue2, ...)` такой вариант синтаксиса функции позволяет задать дополнительные параметры `'pname'` и их значения `pvalue`:

- `'center'` – количество повторений опытов в центральных точках плана;
- `'blocksize'` – максимальное количество точек в блоке плана.

Дополнительные аргументы задаются в виде пары: `название параметра`, `значение параметра`.

Примечание: в работах Бокса и Бенкина предложены планы для числа факторов 3-7, 9-12, 16. Для указанного числа факторов функция `bbdesign` генерирует матрицы планов экспериментов согласно Боксу и Бенкину. Для числа факторов отличного от приведенных значений матрица плана рассчитывается по алгоритму аналогичному изложенному в работах Бокса и Бенкина.

0	0	0	1	1	1
0	-1	-1	0	0	2
0	-1	1	0	0	2
0	1	-1	0	0	2
0	1	1	0	0	2
-1	0	0	-1	0	2
-1	0	0	1	0	2
1	0	0	-1	0	2
1	0	0	1	0	2
0	0	-1	0	-1	2
0	0	-1	0	1	2
0	0	1	0	-1	2
0	0	1	0	1	2
-1	0	0	0	-1	2
-1	0	0	0	1	2
1	0	0	0	-1	2
1	0	0	0	1	2
0	-1	0	-1	0	2
0	-1	0	1	0	2
0	1	0	-1	0	2
0	1	0	1	0	2
0	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	2
0	0	0	0	0	2
0	0	0	0	0	2

Генерация матрицы плана Бокса-Бенкина для 4 факторов разделенного на блоки при условии 5 повторных опытов в центральной точке плана и максимальном числе точек в блоке плана равном 10.

```
>> [D,blk] = bbdesign(4, 'center', 5, 'blocksize', 10)
```

```
D =
```

-1	-1	0	0
-1	1	0	0
1	-1	0	0
1	1	0	0
0	0	-1	-1
0	0	-1	1
0	0	1	-1
0	0	1	1
-1	0	0	-1
-1	0	0	1
1	0	0	-1

2
2
3

Графически матрицу плана Бокса-Бенкина для 3 факторов можно представить следующим образом

```
>> D = bbdesign(3)
```

```
D =
```

```
-1 -1 0  
-1 1 0  
1 -1 0  
1 1 0  
-1 0 -1  
-1 0 1  
1 0 -1  
1 0 1  
0 -1 -1  
0 -1 1  
0 1 -1  
0 1 1  
0 0 0  
0 0 0  
0 0 0
```

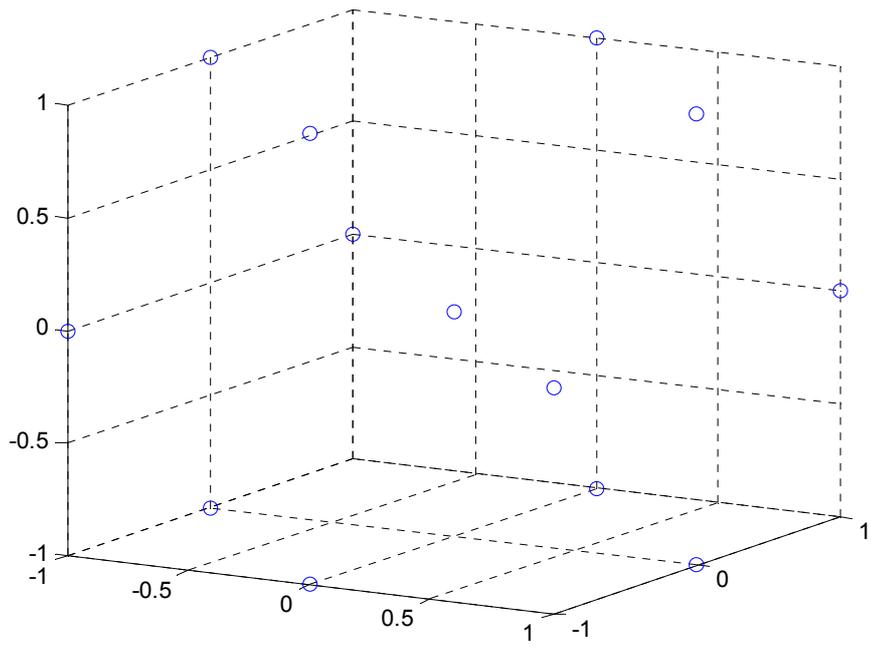
```
>> x= D(:,1);
```

```
>> y= D(:,2);
```

```
>> z= D(:,3);
```

```
>> plot3(x,y,z,'o')
```

```
>> grid on
```



ccdesign

Генерация матрицы центрального композиционного плана

Синтаксис

```
D = ccdesign(nfactors)
D =
ccdesign(nfactors, 'pname1', pvalue1, 'pname2', pvalue2, .
..)
[D,blk] = ccdesign(...)
```

Описание

`D = ccdesign(nfactors)` функция предназначена для генерации матрицы `D` центрального композиционного плана для `nfactors` факторов. Размерность матрицы плана эксперимента `D` равна $nfactors \times nfactors$. Количество опытов в матрице равно числу факторов. Значения факторов нормализованы таким образом, чтобы вершины куба плана были равны -1 и 1.

`[D,blk] = ccdesign(nfactors)` функция возвращает вектор номеров блоков `blk` опытов в плане эксперимента. Блоки позволяют сгруппировать опыты таким образом, чтобы минимизировать влияние изменений внешних условий на значения оцениваемых параметров.

`[...] = ccdesign(nfactors, 'pname1', pvalue1, 'pname2', pvalue2, ...)` такой вариант синтаксиса позволяет задать следующие дополнительные параметры `'pname'` и их значения `pvalue`:

'center'	Число центральных точек в плане:	
	Целое число	Определяет число центральных точек в плане
	'uniform'	Число центральных точек выбирается для обеспечения равной точности в области проведения эксперимента
	'orthogonal'	Число центральных точек выбирается для обеспечения ортогональности плана. Принимается как значение по умолчанию.
'fraction'	Вид дробной реплики от полного факторного плана. Задается в виде степени Ω . Например:	
	0	Полный план
	1	1/2 реплика
	2	1/4 реплика
'type'	Вид центрального композиционного плана. Возможные	

	значения 'inscribed', 'circumscribed', 'faced'.
'blocksize'	Максимальное количество точек в блоке плана

Дополнительные аргументы задаются в виде пары: $\{\text{название параметра}, \text{значение параметра}\}$.

Примеры использования функции генерации матрицы центрального композиционного плана

Генерация матрицы центрального композиционного плана для 3 факторов

```
>> D = ccdesign(3)
```

```
D =
```

```
-1.0000 -1.0000 -1.0000
-1.0000 -1.0000  1.0000
-1.0000  1.0000 -1.0000
-1.0000  1.0000  1.0000
 1.0000 -1.0000 -1.0000
 1.0000 -1.0000  1.0000
 1.0000  1.0000 -1.0000
 1.0000  1.0000  1.0000
-1.6818  0      0
 1.6818  0      0
  0     -1.6818  0
  0     1.6818  0
  0      0     -1.6818
  0      0     1.6818
  0      0      0
  0      0      0
  0      0      0
  0      0      0
  0      0      0
  0      0      0
  0      0      0
  0      0      0
  0      0      0
  0      0      0
  0      0      0
```

Генерация матрицы центрального композиционного плана для 5 факторов разделенного на блоки

```
>> [D,blk] = ccdesign(5)
```

```
D =
```

```
-1 -1 -1 -1  1
-1 -1 -1  1 -1
-1 -1  1 -1 -1
```

```
blk =
```

```
1
1
1
```

-1	-1	1	1	1	1
-1	1	-1	-1	-1	1
-1	1	-1	1	1	1
-1	1	1	-1	1	1
-1	1	1	1	-1	1
1	-1	-1	-1	-1	1
1	-1	-1	1	1	1
1	-1	1	-1	1	1
1	-1	1	1	-1	1
1	1	-1	-1	1	1
1	1	-1	1	-1	1
1	1	1	-1	-1	1
1	1	1	1	1	1
-2	0	0	0	0	2
2	0	0	0	0	2
0	-2	0	0	0	2
0	2	0	0	0	2
0	0	-2	0	0	2
0	0	2	0	0	2
0	0	0	-2	0	2
0	0	0	2	0	2
0	0	0	0	-2	2
0	0	0	0	2	2
0	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	2
0	0	0	0	0	2
0	0	0	0	0	2
0	0	0	0	0	2
0	0	0	0	0	2
0	0	0	0	0	2
0	0	0	0	0	2

Генерация матрицы Ω дробной реплики центрального композиционного плана для 5 факторов разделенного на блоки с числом центральных точек в плане равном 7. Максимальное количество точек в блоке плана n 20.

`>> [D,blk] = ccdesign(5, 'center', 7, 'fraction', 1, 'blocksize', 20)`

D =

-1	-1	-1	-1	1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	-1	1	1	1
-1	1	-1	-1	-1

2
2
2
2
2
2
2
2
2
2
1
1
1
2
2
2
2

Генерация матрицы 1/2 дробной реплики центрального композиционного плана для 5 факторов разделенного на блоки с числом центральных точек расположенных с целью обеспечения равной точности в области проведения эксперимента. Вид центрального композиционного плана - 'inscribed'.

```
>> [D,blk] = ccdesign(5, 'center', 'uniform', 'fraction', 1, 'type', 'inscribed')
```

D =

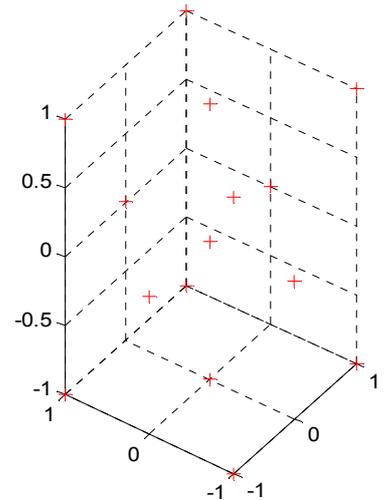
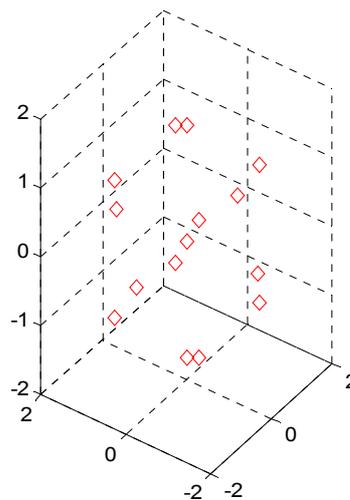
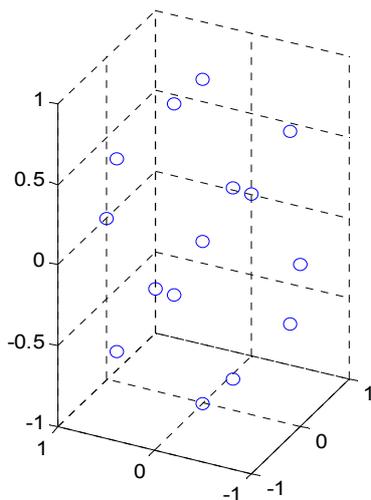
-0.5000	-0.5000	-0.5000	-0.5000	0.5000
-0.5000	-0.5000	-0.5000	0.5000	-0.5000
-0.5000	-0.5000	0.5000	-0.5000	-0.5000
-0.5000	-0.5000	0.5000	0.5000	0.5000
-0.5000	0.5000	-0.5000	-0.5000	-0.5000
-0.5000	0.5000	-0.5000	0.5000	0.5000
-0.5000	0.5000	0.5000	-0.5000	0.5000
-0.5000	0.5000	0.5000	0.5000	-0.5000
0.5000	-0.5000	-0.5000	-0.5000	-0.5000
0.5000	-0.5000	-0.5000	0.5000	0.5000
0.5000	-0.5000	0.5000	-0.5000	0.5000
0.5000	-0.5000	0.5000	0.5000	-0.5000
0.5000	0.5000	-0.5000	-0.5000	0.5000
0.5000	0.5000	-0.5000	0.5000	-0.5000
0.5000	0.5000	0.5000	-0.5000	-0.5000
0.5000	0.5000	0.5000	0.5000	0.5000
-1.0000	0	0	0	0
1.0000	0	0	0	0
0	-1.0000	0	0	0
0	1.0000	0	0	0

Графическое представление 3-х видов центрального композиционного плана для 3 факторов

```
>> nfact=3;  
>> D = ccdesign(nfact, 'type', 'inscribed');  
>> x= D(:,1);  
>> y= D(:,2);  
>> z= D(:,3);  
>> subplot(1,3,1)  
>> plot3(x,y,z,'o')  
>> grid on
```

```
>> D1 = ccdesign(nfact, 'type', 'circumscribed');  
>> x1= D1(:,1);  
>> y1= D1(:,2);  
>> z1= D1(:,3);  
>> subplot(1,3,2)  
>> plot3(x1,y1,z1,'dr')  
>> grid on
```

```
>> D2 = ccdesign(nfact, 'type', 'faced');  
>> x2= D2(:,1);  
>> y2= D2(:,2);  
>> z2= D2(:,3);  
>> subplot(1,3,3)  
>> plot3(x2,y2,z2,'+r')  
>> grid on
```



rowexch

Генерация матрицы D-оптимального плана на основе алгоритма перестановки строк

Синтаксис

```
settings = rowexch(nfactors,nruns)
[settings,X] = rowexch(nfactors,nruns)
[settings,X] = rowexch(nfactors,nruns,'model')
[settings,X] =
rowexch(...,'param1',value1,'param2',value2,...)
```

Описание

`settings = rowexch(nfactors,nruns)` функция позволяет получить матрицу значений факторов D-оптимального плана `settings` на основе алгоритма перестановки строк в кодированных переменных, изменяющихся на двух уровнях (-1 ñ минимальное значение фактора, 1 ñ максимальное значение фактора). `settings` представляет собой матрицу D-оптимального плана эксперимента за исключением столбца соответствующего постоянному члену. Входными параметрами являются число факторов `nfactors` и количество опытов `nruns`. В качестве уравнения регрессии принимается линейная модель. Размерность матрицы `settings` равна `nruns` строк на `nfactors` столбцов.

`[settings,X] = rowexch(nfactors,nruns)` кроме матрицы значений факторов `settings` функция `rowexch` возвращает матрицу D-оптимального плана эксперимента `X`.

`[settings,X] = rowexch(nfactors,nruns,'model')` этот вариант синтаксиса функции предусматривает генерацию матриц `settings` и `X` для заданных числа факторов `nfactors`, количества опытов `nruns` и вида математической модели `'model'`. Предусмотрены следующие виды математических моделей:

Значение 'model'	Вид математической модели
'linear'	Линейная модель с постоянным членом. Значение по умолчанию.
'interaction'	Линейная модель с постоянным членом и эффектами взаимодействия.
'quadratic'	Полная квадратическая модель, включающая линейные, квадратические эффекты, эффекты взаимодействий, постоянный член.
'purequadratic'	Неполная квадратическая модель, включающая линейные, квадратические эффекты, постоянный

	член.
--	-------

[settings, X] = rowexch(..., 'param1', value1, 'param2', value2,...) кроме числа факторов nfactores, количества опытов nruns, вида математической модели 'model' предусмотрены следующие дополнительные входные аргументы:

'display'	Вывод значения счетчика итераций. Возможные значения 'display': 'on' ñ вывод в командное окно, 'off' ñ отмена отображения. Значение по умолчанию 'on'.
'init'	Начальная матрица значений факторов с размерностью nruns ð nfactores. По умолчанию предусматривается формирование начальной матрицы значений факторов случайным образом.
'maxiter'	Максимальное число итераций. Значение по умолчанию - 10.

Дополнительные аргументы задаются в виде пары: ñназвание параметра, значение параметра.

Примечание. Функция rowexch осуществляет поиск D-оптимального плана на основе алгоритма перестановки строк. На первом этапе генерируется начальное множество точек в факторном пространстве, которые могут быть включены в план эксперимента. На втором этапе посредством перестановки строк в начальной матрице плана, формируется результирующая матрица плана эксперимента по критерию минимизации дисперсии коэффициентов уравнения регрессии. Если необходимо задать начальное множество точек отличное от генерируемого по умолчанию функцией rowexch используются функции sandgen and sandexch.

Примеры использования функции генерации матрицы D-оптимального плана на основе алгоритма перестановки строк

Генерация матрицы значений факторов D-оптимального плана для 4 факторов и 10 опытов

```
>>nfactores=4;
>> nruns=10;
>>settings = rowexch(nfactores,nruns)
settings =
    -1  -1   1   1
    -1  -1  -1  -1
     1   1   1  -1
     1  -1   1   1
    -1   1   1  -1
```

```

1 -1 -1 -1
1 1 -1 1
-1 1 -1 1
-1 -1 1 1
1 -1 -1 -1

```

Генерация матрицы значений факторов и матрицы D-оптимального плана для 5 факторов и 12 опытов

```

>>nfactors=5;
>> nruns=12;
>>[settings,X] = rowexch(nfactors,nruns)

```

settings =

```

1 1 -1 -1 1
-1 -1 -1 -1 1
1 -1 -1 1 -1
1 -1 1 -1 -1
1 -1 1 -1 1
-1 -1 1 1 1
1 1 1 1 -1
-1 -1 -1 1 -1
-1 1 1 -1 -1
-1 1 1 1 1
-1 1 -1 -1 -1
1 1 -1 1 1

```

X =

```

1 1 1 -1 -1 1
1 -1 -1 -1 -1 1
1 1 -1 -1 1 -1
1 1 -1 1 -1 -1
1 1 -1 1 -1 1
1 -1 -1 1 1 1
1 1 1 1 1 -1
1 -1 -1 -1 1 -1
1 -1 1 1 -1 -1
1 -1 1 1 1 1
1 -1 1 -1 -1 -1
1 1 1 -1 1 1

```

Генерация матрицы значений факторов и матрицы D-оптимального плана для 4 факторов, 20 опытов и полной квадратической модели.

```

>>nfactors=4;
>> nruns=20;
>> model='quadratic';

```

```
>>[settings,X] = rowexch(nfactors,nruns,model)
```

```
settings =
```

```
 1 -1  1 -1
 0  0  0  0
-1  0  1 -1
-1 -1  1  0
 1  1 -1  1
 1 -1  1  1
 1  0 -1 -1
 0 -1 -1  1
 1  1  1 -1
-1 -1 -1 -1
 1 -1 -1  0
-1  1 -1  1
-1 -1  0  1
-1  1  0 -1
 0  1 -1 -1
 0 -1  0 -1
-1  1  1  1
 1  1  1  1
-1  0 -1  0
 1  1  0  0
```

```
X =
```

```
 1  1 -1  1 -1 -1  1 -1 -1  1 -1  1  1  1  1
 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 1 -1  0  1 -1  0 -1  1  0  0 -1  1  0  1  1
 1 -1 -1  1  0  1 -1  0 -1  0  0  1  1  1  0
 1  1  1 -1  1  1 -1  1 -1  1 -1  1  1  1  1
 1  1 -1  1  1 -1  1  1 -1 -1  1  1  1  1  1
 1  1  0 -1 -1  0 -1 -1  0  0  1  1  0  1  1
 1  0 -1 -1  1  0  0  0  1 -1 -1  0  1  1  1
 1  1  1  1 -1  1  1 -1  1 -1 -1  1  1  1  1
 1 -1 -1 -1 -1  1  1  1  1  1  1  1  1  1  1
 1  1 -1 -1  0 -1 -1  0  1  0  0  1  1  1  0
 1 -1  1 -1  1 -1  1 -1 -1  1 -1  1  1  1  1
 1 -1 -1  0  1  1  0 -1  0 -1  0  1  1  0  1
 1 -1  1  0 -1 -1  0  1  0 -1  0  1  1  0  1
 1  0  1 -1 -1  0  0  0 -1 -1  1  0  1  1  1
 1  0 -1  0 -1  0  0  0  0  1  0  0  1  0  1
 1 -1  1  1  1 -1 -1 -1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1 -1  0 -1  0  0  1  0  0  0  0  1  0  1  0
 1  1  1  0  0  1  0  0  0  0  0  1  1  0  0
```

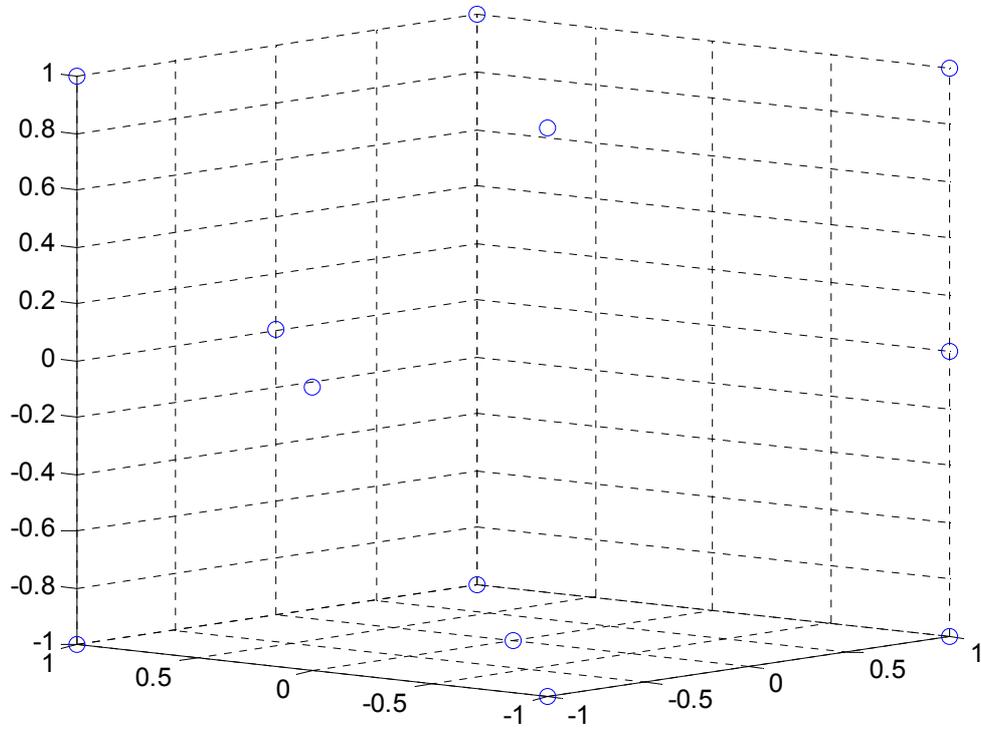
Генерация матрицы значений факторов и матрицы D-оптимального плана для 4 факторов, 4 опытов и неполной квадратической модели. A является начальной матрицей для генерации D-оптимального плана эксперимента.

```
>>nfactors=4;
>> nruns=4;
>> model='purequadratic';
>> A=[1 0 1 0; 0 0 0 0; 1 1 1 1; 0 0 1 1]
A =
    1    0    1    0
    0    0    0    0
    1    1    1    1
    0    0    1    1
>>[settings,X] = rowexch(nfactors,nruns,model,'init',A)
settings =
   -1    1   -1    1
    1   -1   -1   -1
    1    1    1    0
    0   -1    1    1
X =
    1   -1    1   -1    1    1    1    1    1
    1    1   -1   -1   -1    1    1    1    1
    1    1    1    1    0    1    1    1    0
    1    0   -1    1    1    0    1    1    1
```

Графическое представление матрицы D-оптимального плана рассчитанной для 3 факторов, 12 опытов и полной квадратической модели

```
>>nfactors=3;
>> nruns=12;
>> model='quadratic';
>>settings = rowexch(nfactors,nruns,model)
settings =
   -1    1   -1
    0    0   -1
    1    1   -1
    1   -1   -1
    1    1    1
   -1    1    1
    1   -1    1
   -1   -1   -1
   -1   -1    1
    1   -1    0
   -1    0    0
```

```
0 1 0
>> x= settings (:,1);
>> y= settings (:,2);
>> z= settings (:,3);
>> plot3(x,y,z,'o')
>> grid on
```



candexch

Генерация D-оптимального плана из начального множества точек в факторном пространстве на основе алгоритма перестановки строк

Синтаксис

```
rlist = candexch(C,nrows)
rlist =
candexch(C,nrows,'param1',value1,'param2',value2,...)
```

Описание

`rlist = candexch(C,nrows)` функция предназначена для генерации плана эксперимента `rlist` из начального множества точек `C` в факторном пространстве на основе алгоритма перестановки строк. Матрица `C`, размерностью $n \times p$, содержит значения p эффектов модели для каждой из n точек. Входной параметр `nrows` задает количество опытов в плане эксперимента. Выходной параметр `rlist` является вектором с числом элементов `nrows` и задает номера выбранных строк из начального множества точек `C`.

Функция `candexch` выбирает матрицу начального плана X случайным образом. Затем на основе алгоритма замены строк X строками начального множества точек `C` формируется D-оптимальный план эксперимента. Вычисления прекращаются, если будет выполнено условие $\max\{\det[X' \cdot X]\}$, или достигнуто максимальное число итераций.

`rlist = candexch(C,nrows,'param1',value1,'param2',value2,...)` этот вариант синтаксиса функции позволяет определить следующие дополнительные входные аргументы:

'display'	Вывод значения счетчика итераций. Возможные значения 'display': 'on' ñ вывод в командное окно, 'off' ñ отмена отображения. Значение по умолчанию 'on'.
'init'	Начальная матрица значений факторов с размерностью $nrows \times p$. По умолчанию принимается случайное подмножество строк начального множества <code>C</code> .
'maxiter'	Максимальное число итераций. Значение по умолчанию - 10.

Дополнительные аргументы задаются в виде пары: ñназвание параметра, значение параметра.

Примеры использования функции генерации матрицы D-оптимального плана из начального множества точек в факторном пространстве на основе алгоритма перестановки строк

Генерация матрицы значений факторов X D-оптимального плана из начального множества точек в факторном пространстве для 12 опытов. Начальное множество xcand генерируется при помощи функции candgen для 3 факторов и полной квадратической модели.

```
>> nfact=3;  
>> model='quadratic';  
>> xcand = candgen(nfact,model);  
>> nrow=12;  
>> rlist = candexch(xcand,nrow)
```

```
rlist =
```

```
9  
1  
7  
3  
3  
7  
9  
1  
1  
7  
3  
9
```

```
>> X = xcand(rlist,:)
```

```
X =
```

```
1 1 -1  
-1 -1 -1  
-1 1 -1  
1 -1 -1  
1 -1 -1  
-1 1 -1  
1 1 -1  
-1 -1 -1  
-1 -1 -1  
-1 1 -1  
1 -1 -1  
1 1 -1
```

Генерация матрицы D-оптимального плана X из начального множества точек C в факторном пространстве для 12 опытов. Начальное множество C генерируется при помощи функции полного

факторного эксперимента fullfact для 3 факторов на 5 уровнях. Матрица значений факторов полного факторного эксперимента приводится к единичному кубу F. На полученную матрицу значений факторов накладывается ограничение по величине построчной суммы меньше или равной 1,51: $\text{sum}(F,2) \leq 1.51$. Затем формируется матрица F со строками, удовлетворяющими указанному ограничению. Начальное множество C является матрицей значений факторов соответствующих постоянному члену, линейным и квадратическим эффектам.

```
>> F = (fullfact([5 5 5])-1)/4;
```

```
>> T = sum(F,2) <= 1.51
```

```
>> F = F(T,:)
```

```
>> C = [ones(size(F,1),1) F F.^2]
```

```
C =
```

1.0000	0	0	0	0
0	0			
1.0000	0.2500	0	0	0.0625
0	0			
1.0000	0.5000	0	0	0.2500
0	0			
1.0000	0.7500	0	0	0.5625
0	0			
1.0000	1.0000	0	0	1.0000
0	0			
1.0000	0	0.2500	0	0
0.0625	0			
1.0000	0.2500	0.2500	0	0.0625
0.0625	0			
1.0000	0.5000	0.2500	0	0.2500
0.0625	0			
1.0000	0.7500	0.2500	0	0.5625
0.0625	0			
1.0000	1.0000	0.2500	0	1.0000
0.0625	0			
1.0000	0	0.5000	0	0
0.2500	0			
1.0000	0.2500	0.5000	0	0.0625
0.2500	0			
1.0000	0.5000	0.5000	0	0.2500
0.2500	0			
1.0000	0.7500	0.5000	0	0.5625
0.2500	0			
1.0000	1.0000	0.5000	0	1.0000
0.2500	0			

1.0000	0	0.7500	0	0
0.5625	0			
1.0000	0.2500	0.7500	0	0.0625
0.5625	0			
1.0000	0.5000	0.7500	0	0.2500
0.5625	0			
1.0000	0.7500	0.7500	0	0.5625
0.5625	0			
1.0000	0	1.0000	0	0
1.0000	0			
1.0000	0.2500	1.0000	0	0.0625
1.0000	0			
1.0000	0.5000	1.0000	0	0.2500
1.0000	0			
1.0000	0	0	0.2500	0
0	0.0625			
1.0000	0.2500	0	0.2500	0.0625
0	0.0625			
1.0000	0.5000	0	0.2500	0.2500
0	0.0625			
1.0000	0.7500	0	0.2500	0.5625
0	0.0625			
1.0000	1.0000	0	0.2500	1.0000
0	0.0625			
1.0000	0	0.2500	0.2500	0
0.0625	0.0625			
1.0000	0.2500	0.2500	0.2500	0.0625
0.0625	0.0625			
1.0000	0.5000	0.2500	0.2500	0.2500
0.0625	0.0625			
1.0000	0.7500	0.2500	0.2500	0.5625
0.0625	0.0625			
1.0000	1.0000	0.2500	0.2500	1.0000
0.0625	0.0625			
1.0000	0	0.5000	0.2500	0
0.2500	0.0625			
1.0000	0.2500	0.5000	0.2500	0.0625
0.2500	0.0625			
1.0000	0.5000	0.5000	0.2500	0.2500
0.2500	0.0625			
1.0000	0.7500	0.5000	0.2500	0.5625
0.2500	0.0625			
1.0000	0	0.7500	0.2500	0
0.5625	0.0625			
1.0000	0.2500	0.7500	0.2500	0.0625
0.5625	0.0625			

1.0000	0.5000	0.7500	0.2500	0.2500
0.5625	0.0625			
1.0000	0	1.0000	0.2500	0
1.0000	0.0625			
1.0000	0.2500	1.0000	0.2500	0.0625
1.0000	0.0625			
1.0000	0	0	0.5000	0
0	0.2500			
1.0000	0.2500	0	0.5000	0.0625
0	0.2500			
1.0000	0.5000	0	0.5000	0.2500
0	0.2500			
1.0000	0.7500	0	0.5000	0.5625
0	0.2500			
1.0000	1.0000	0	0.5000	1.0000
0	0.2500			
1.0000	0	0.2500	0.5000	0
0.0625	0.2500			
1.0000	0.2500	0.2500	0.5000	0.0625
0.0625	0.2500			
1.0000	0.5000	0.2500	0.5000	0.2500
0.0625	0.2500			
1.0000	0.7500	0.2500	0.5000	0.5625
0.0625	0.2500			
1.0000	0	0.5000	0.5000	0
0.2500	0.2500			
1.0000	0.2500	0.5000	0.5000	0.0625
0.2500	0.2500			
1.0000	0.5000	0.5000	0.5000	0.2500
0.2500	0.2500			
1.0000	0	0.7500	0.5000	0
0.5625	0.2500			
1.0000	0.2500	0.7500	0.5000	0.0625
0.5625	0.2500			
1.0000	0	1.0000	0.5000	0
1.0000	0.2500			
1.0000	0	0	0.7500	0
0	0.5625			
1.0000	0.2500	0	0.7500	0.0625
0	0.5625			
1.0000	0.5000	0	0.7500	0.2500
0	0.5625			
1.0000	0.7500	0	0.7500	0.5625
0	0.5625			
1.0000	0	0.2500	0.7500	0
0.0625	0.5625			

```

    1.0000    0.2500    0.2500    0.7500    0.0625
0.0625    0.5625
    1.0000    0.5000    0.2500    0.7500    0.2500
0.0625    0.5625
    1.0000            0    0.5000    0.7500            0
0.2500    0.5625
    1.0000    0.2500    0.5000    0.7500    0.0625
0.2500    0.5625
    1.0000            0    0.7500    0.7500            0
0.5625    0.5625
    1.0000            0            0    1.0000            0
0    1.0000
    1.0000    0.2500            0    1.0000    0.0625
0    1.0000
    1.0000    0.5000            0    1.0000    0.2500
0    1.0000
    1.0000            0    0.2500    1.0000            0
0.0625    1.0000
    1.0000    0.2500    0.2500    1.0000    0.0625
0.0625    1.0000
    1.0000            0    0.5000    1.0000            0
0.2500    1.0000

```

```
>> R = candexch(C,12)
```

```
R =
```

```
22
```

```
56
```

```
15
```

```
46
```

```
69
```

```
72
```

```
35
```

```
51
```

```
1
```

```
46
```

```
44
```

```
1
```

```
>> X = F(R,:)
```

```
X =
```

```

    0.5000    1.0000            0
            0    1.0000    0.5000
    1.0000    0.5000            0
    1.0000            0    0.5000
    0.5000            0    1.0000
            0    0.5000    1.0000
    0.5000    0.5000    0.2500

```

	0	0.5000	0.5000
	0	0	0
1.0000		0	0.5000
0.5000		0	0.5000
0		0	0

Генерация матрицы значений факторов X D-оптимального плана из начального множества точек в факторном пространстве для 12 опытов. Начальное множество `xcand` генерируется при помощи функции `candgen` для 3 факторов и полной квадратической модели. В качестве дополнительного параметра задана матрица A начального приближения при поиске D-оптимального плана из множества `xcand`. Матрица A является подматрицей `xcand`, первые ее 12 строк.

```
>> nfactores=3;
>> model='quadratic';
>> xcand = candgen(nfactores,model)
```

```
xcand =
```

```
-1 -1 -1
 0 -1 -1
 1 -1 -1
-1  0 -1
 0  0 -1
 1  0 -1
-1  1 -1
 0  1 -1
 1  1 -1
-1 -1  0
 0 -1  0
 1 -1  0
-1  0  0
 0  0  0
 1  0  0
-1  1  0
 0  1  0
 1  1  0
-1 -1  1
 0 -1  1
 1 -1  1
-1  0  1
 0  0  1
 1  0  1
-1  1  1
 0  1  1
 1  1  1
```

```
>> nrows=12;  
>> A= xcand (1: nrows,:)
```

```
A =  
-1 -1 -1  
 0 -1 -1  
 1 -1 -1  
-1  0 -1  
 0  0 -1  
 1  0 -1  
-1  1 -1  
 0  1 -1  
 1  1 -1  
-1 -1  0  
 0 -1  0  
 1 -1  0
```

```
>> rlist = candexch(xcand,nrows,'init',A)
```

```
rlist =  
 1  
 1  
 3  
 7  
 3  
 9  
 7  
 1  
 9  
 9  
 3  
 7
```

```
>> X = xcand(rlist,:)
```

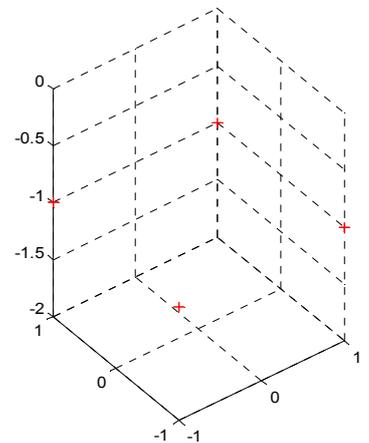
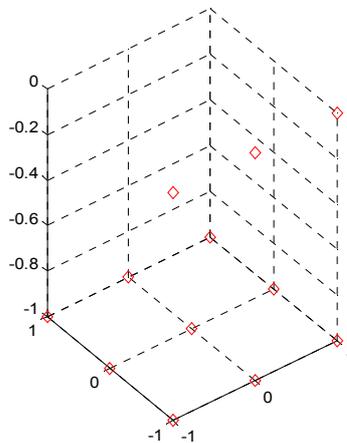
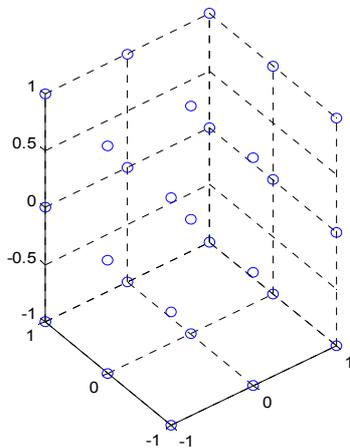
```
X =  
-1 -1 -1  
-1 -1 -1  
 1 -1 -1  
-1  1 -1  
 1 -1 -1  
 1  1 -1  
-1  1 -1  
-1 -1 -1  
 1  1 -1  
 1  1 -1  
 1 -1 -1  
-1  1 -1
```

Графическое представление начального множества x_{cand} , матрицы начального приближения A и матрицы значений факторов X .

```
>> x= xcand (:,1);  
>> y= xcand (:,2);  
>> z= xcand (:,3);  
>> subplot(1,3,1)  
>> plot3(x,y,z,'o')  
>> grid on
```

```
>> x1= A(:,1);  
>> y1= A (:,2);  
>> z1= A (:,3);  
>> subplot(1,3,2)  
>> plot3(x1,y1,z1,'dr')  
>> grid on
```

```
>> x2= X(:,1);  
>> y2= X (:,2);  
>> z2= X (:,3);  
>> subplot(1,3,3)  
>> plot3(x2,y2,z2,'+r')  
>> grid on
```



candgen

Генерация начального множества точек в факторном пространстве для D-оптимального плана

Синтаксис

```
xcand = candgen(nfactors, 'model')  
[xcand, fxcand] = candgen(nfactors, 'model')
```

Описание

`xcand = candgen(nfactors, 'model')` функция предназначена для генерации начального множества точек `xcand` в факторном пространстве, соответствующего D-оптимальному плану с числом факторов `nfactors` и видом модели регрессии `'model'`. `xcand` является матрицей значений факторов. Матрица выходных значений `xcand` содержит `nfactors` столбцов. Предусмотрены следующие виды математической модели:

Значение 'model'	Вид математической модели
'linear'	Линейная модель с постоянным членом. Значение по умолчанию.
'interaction'	Линейная модель с постоянным членом и эффектами взаимодействия.
'quadratic'	Полная квадратическая модель, включающая линейные, квадратические эффекты, эффекты взаимодействий, постоянный член.
'purequadratic'	Неполная квадратическая модель, включающая линейные, квадратические эффекты, постоянный член.

Кроме строкового значения входной аргумент `model` может быть задан как вектор или матрица аналогично такому же аргументу функции `x2fx`. Функция `x2fx` позволяет выполнить преобразование матрицы значений факторов `X` в матрицу плана эксперимента `D`. В случае, если `X` и `model` заданы как векторы, то матрица плана эксперимента формируется по правилу: каждый столбец `D` является последовательным возведением `X` в степень элемента вектора `model`. Размерность матрицы `D` равна $n \times m$, где n — число элементов вектора `X`, m — число элементов вектора `model`. Т.е., вектор `model` является списком степеней полинома регрессионной модели для одного фактора `X`. Если `X` и `model` заданы как матрицы, то столбец матрицы `D` формируются по формуле:

$$D(i,j)=\text{prod}(X(i,:).^{\text{model}(j,:)}),$$

т.е., ij -й элемент матрицы D является произведением элементов i -й строки матрицы X возведенных последовательно в степени j -й строки матрицы $model$. Таким образом, количество столбцов $model$ должно быть равно числу столбцов матрицы X .

`[xcand, fxcand] = candgen(nfactors, model)` функция возвращает матрицу значений факторов `xcand` и матрицу значений степеней `fxcand`.

Примечание: Функция `gowexch` генерирует начальное множество точек в факторном пространстве с помощью функции `candgen` и формирует матрицу D -оптимального плана с помощью функции `candexch`. Раздельное использование функций `candgen` и `candexch` целесообразно в случае, когда необходимо изменить начальное множество точек.

Примеры использования функции генерации начального множества точек в факторном пространстве для D -оптимального плана

Генерация начального множества в виде матрицы значений факторов D -оптимального плана для 3 факторов и полной квадратической модели

```
>> nfactors=3;  
>> model='quadratic';  
>> xcand = candgen(nfactors,model)
```

```
xcand =  
-1 -1 -1  
 0 -1 -1  
 1 -1 -1  
-1  0 -1  
 0  0 -1  
 1  0 -1  
-1  1 -1  
 0  1 -1  
 1  1 -1  
-1 -1  0  
 0 -1  0  
 1 -1  0  
-1  0  0  
 0  0  0  
 1  0  0  
-1  1  0  
 0  1  0  
 1  1  0
```

```
-1 -1 1
0 -1 1
1 -1 1
-1 0 1
0 0 1
1 0 1
-1 1 1
0 1 1
1 1 1
```

Генерация начального множества в виде матрицы значений факторов и матрицы D-оптимального плана для 3 факторов и неполной квадратической модели

```
>> nfactors=3;
>> model='purequadratic';
>> [xcand,fxcand] = candgen(nfactors,model)
```

xcand =

```
-1 -1 -1
0 -1 -1
1 -1 -1
-1 0 -1
0 0 -1
1 0 -1
-1 1 -1
0 1 -1
1 1 -1
-1 -1 0
0 -1 0
1 -1 0
-1 0 0
0 0 0
1 0 0
-1 1 0
0 1 0
1 1 0
-1 -1 1
0 -1 1
1 -1 1
-1 0 1
0 0 1
1 0 1
-1 1 1
0 1 1
1 1 1
```

fxcand =

```

1 -1 -1 -1 1 1 1
1 0 -1 -1 0 1 1
1 1 -1 -1 1 1 1
1 -1 0 -1 1 0 1
1 0 0 -1 0 0 1
1 1 0 -1 1 0 1
1 -1 1 -1 1 1 1
1 0 1 -1 0 1 1
1 1 1 -1 1 1 1
1 -1 -1 0 1 1 0
1 0 -1 0 0 1 0
1 1 -1 0 1 1 0
1 -1 0 0 1 0 0
1 0 0 0 0 0 0
1 1 0 0 1 0 0
1 -1 1 0 1 1 0
1 0 1 0 0 1 0
1 1 1 0 1 1 0
1 -1 -1 1 1 1 1
1 0 -1 1 0 1 1
1 1 -1 1 1 1 1
1 -1 0 1 1 0 1
1 0 0 1 0 0 1
1 1 0 1 1 0 1
1 -1 1 1 1 1 1
1 0 1 1 0 1 1
1 1 1 1 1 1 1

```

Генерация начального множества в виде матрицы значений факторов и матрицы D-оптимального плана для 3 факторов и регрессионной модели заданной в виде матрицы model.

```

>> nfactors=3;
>> model=[0 1 2; 0 1 2; 0 1 2; 0 1 2]
model =
    0    1    2
    0    1    2
    0    1    2
    0    1    2
>> [xcand, fxcand] = candgen(nfactors, model)
xcand =
   -1   -1   -1
    1   -1   -1
   -1    1   -1
    1    1   -1
   -1   -1    0

```

```

1 -1 0
-1 1 0
1 1 0
-1 -1 1
1 -1 1
-1 1 1
1 1 1
fxcand =
-1 -1 -1 -1
-1 -1 -1 -1
1 1 1 1
1 1 1 1
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
-1 -1 -1 -1
-1 -1 -1 -1
1 1 1 1
1 1 1 1

```

Графическое представление начального множества xcand значений 3-х факторов и полной квадратической модели.

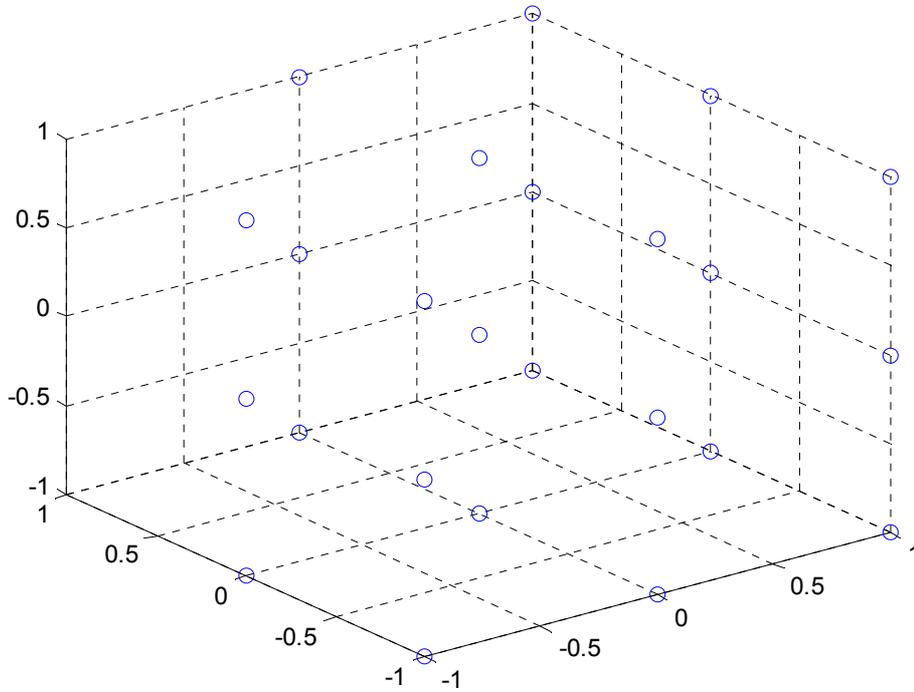
```

>> nfact=3;
>> model='quadratic';
>> xcand = candgen(nfact,model)
xcand =
-1 -1 -1
0 -1 -1
1 -1 -1
-1 0 -1
0 0 -1
1 0 -1
-1 1 -1
0 1 -1
1 1 -1
-1 -1 0
0 -1 0
1 -1 0
-1 0 0
0 0 0
1 0 0
-1 1 0
0 1 0
1 1 0

```

```
-1 -1 1  
0 -1 1  
1 -1 1  
-1 0 1  
0 0 1  
1 0 1  
-1 1 1  
0 1 1  
1 1 1
```

```
>> x= xcand (:,1);  
>> y= xcand (:,2);  
>> z= xcand (:,3);  
>> plot3(x,y,z,'o')  
>> grid on
```



x2fx

Преобразование матрицы значений факторов в матрицу плана эксперимента

Синтаксис

$D = x2fx(X)$

$D = x2fx(X, 'model')$

Описание

$D = x2fx(X)$ - функция предназначена для преобразования матрицы значений факторов X в матрицу плана эксперимента D для линейной регрессионной модели.

$D = x2fx(X, 'model')$ - функция предназначена для преобразования матрицы значений факторов X в матрицу плана эксперимента D для заданной регрессионной модели 'model'. Предусмотрены следующие виды регрессионных моделей:

Значение 'model'	Вид математической модели
'linear'	Линейная модель с постоянным членом. Значение по умолчанию.
'interaction'	Линейная модель с постоянным членом и эффектами взаимодействия.
'quadratic'	Полная квадратическая модель, включающая линейные, квадратические эффекты, эффекты взаимодействий, постоянный член.
'purequadratic'	Неполная квадратическая модель, включающая линейные, квадратические эффекты, постоянный член.

Порядок столбцов в матрице D в общем случае для 'model'='quadratic' соответствует последовательности:

1. постоянный член;
2. линейные эффекты факторов X_i , $i=1..k$, где k - количество факторов в плане эксперимента, k соответствует числу столбцов в матрице X ;
3. парные эффекты взаимодействий в следующей последовательности (1,2), (1,3), ..., (1,k), (2,3), ..., (k-1,k);
4. квадратичные эффекты факторов, следующие в порядке X_i^2 , $i=1..k$.

Для регрессионных моделей 'interaction', 'purequadratic' исключаются столбцы, соответствующие отсутствующим в этих моделях эффектам. Порядок следования остальных эффектов факторов остается тем же, что и для 'model'='quadratic'.

Входной аргумент *model* может быть задан в виде вектора или матрицы. В случае, если *X* и *model* заданы как векторы, то матрица плана эксперимента формируется по правилу: каждый столбец *D* является последовательным возведением *X* в степень элемента вектора *model*. Размерность матрицы *D* равна $n \times m$, где *n* – число элементов вектора *X*, *m* – число элементов вектора *model*. Т.е., вектор *model* является списком степеней полинома регрессионной модели для одного фактора *X*. Если *X* и *model* заданы как матрицы, то столбец матрицы *D* формируются по формуле:

$$D(i,j)=\text{prod}(X(i,:).^{\text{model}(j,:)}),$$

т.е. *ij*-й элемент матрицы *D* является произведением элементов *i*-й строки матрицы *X* возведенных последовательно в степени *j*-й строки матрицы *model*. Таким образом, количество столбцов *model* должно быть равно числу столбцов матрицы *X*.

x2fx является вспомогательной для функций *rstool*, *regstats*, and *cordexch*.

Примеры использования функции преобразования матрицы значений факторов в матрицу плана эксперимента

Преобразование матрицы значений факторов *X* в матрицу плана эксперимента *D* для линейной модели, заданной по умолчанию

```
>> X = [1 2 3;4 5 6]'
```

```
X =
```

```
1 4
2 5
3 6
```

```
>> D = x2fx(X)
```

```
D =
```

```
1 1 4
1 2 5
1 3 6
```

В матрице *D* первый столбец соответствует постоянному члену, 2 и 3 столбцы – линейным эффектам x_1 , x_2 .

Преобразование матрицы значений факторов *X* в матрицу плана эксперимента *D* для полной квадратической модели

```

>> X = [1 2 3;4 5 6]'
X =
     1     4
     2     5
     3     6
>> model = 'quadratic'
model =
quadratic
>> D = x2fx(X,model)
D =
     1     1     4     4     1    16
     1     2     5    10     4    25
     1     3     6    18     9    36

```

В матрице D первый столбец соответствует постоянному члену, 2 и 3 столбцы – линейным эффектам x_1 , x_2 , четвертый – эффекту взаимодействия x_1x_2 , последние два - квадратичным эффектам x_1 , x_2 .

Преобразование вектора значений фактора X в матрицу плана эксперимента D для модели заданной в виде вектора.

```

>> x = [1 2 3]'
x =
     1
     2
     3
>> model = [0 1 2]'
model =
     0
     1
     2
>> D = x2fx(x,model)
D =
     1     1     1
     1     2     4
     1     3     9

```

В матрице D первый столбец соответствует постоянному члену, 2 и 3 столбцы – линейному и квадратическому эффектам x_1 .

cordexch

Генерация матрицы D-оптимального плана на основе алгоритма изменения координат

Синтаксис

```
settings = cordexch(nfactors, nruns)
[settings, X] = cordexch(nfactors, nruns)
[settings, X] = cordexch(nfactors, nruns, 'model')
[settings, X] =
cordexch(..., 'param1', value1, 'param2', value2, ...)
```

Описание

`settings = cordexch(nfactors, nruns)` функция позволяет получить матрицу значений факторов `settings` D-оптимального плана на основе алгоритма изменения координат для линейной регрессионной модели. Количество факторов задается входным аргументом `nfactors`, число опытов в плане `nruns`. Размерность матрицы `settings` составляет `nruns` строк, `nfactors` столбцов.

`[settings, X] = cordexch(nfactors, nruns)` в отличии от первого варианта синтаксиса функция возвращает матрицу D-оптимального плана `X`.

`[settings, X] = cordexch(nfactors, nruns, 'model')` функция позволяет получить матрицу значений факторов `settings` и матрицу D-оптимального плана эксперимента `X` на основе алгоритма изменения координат для заданного числа факторов `nfactors`, числа опытов `nruns` и вида уравнения регрессии `'model'`. Возможны следующие виды модели регрессии

Значение <i>'model'</i>	Вид математической модели
'linear'	Линейная модель с постоянным членом. Значение по умолчанию.
'interaction'	Линейная модель с постоянным членом и эффектами взаимодействия.
'quadratic'	Полная квадратическая модель, включающая линейные, квадратические эффекты, эффекты взаимодействий, постоянный член.
'purequadratic'	Неполная квадратическая модель, включающая линейные, квадратические эффекты, постоянный член.

Кроме строкового значения входной аргумент `model` может быть задан как вектор или матрица аналогично такому же аргументу функции `x2fx`. Функция `x2fx` позволяет выполнить преобразование матрицы значений факторов X в матрицу плана эксперимента D . В случае, если X и `model` заданы как векторы, то матрица плана эксперимента формируется по правилу: каждый столбец D является последовательным возведением X в степень элемента вектора `model`. Размерность матрицы D равна $n \times m$, где n — число элементов вектора X , m — число элементов вектора `model`. Т.е., вектор `model` является списком степеней полинома регрессионной модели для одного фактора X . Если X и `model` заданы как матрицы, то столбец матрицы D формируются по формуле:

$$D(i,j)=\text{prod}(X(i,:).^{\text{model}(j,:)}),$$

т.е., ij -й элемент матрицы D является произведением элементов i -й строки матрицы X возведенных последовательно в степени j -й строки матрицы `model`. Таким образом, количество столбцов `model` должно быть равно числу столбцов матрицы X .

`[settings, X]` = `cordexch(..., 'param1', value1, 'param2', value2, ...)`
 дополнительные входные аргументы по отношению к предыдущим вариантам синтаксиса позволяют:

'display'	Вывод значения счетчика итераций. Возможные значения 'display': 'on' — вывод в командное окно, 'off' — отмена отображения. Значение по умолчанию 'on'.
'init'	Начальная матрица значений факторов с размерностью <code>nruns</code> × <code>nfactors</code> . По умолчанию начальное множество точек в факторном пространстве выбирается случайным образом.
'maxiter'	Максимальное число итераций. Значение по умолчанию - 10.

Примечание. Функция `cordexch` выполняет поиск D -оптимального плана на основе алгоритма перестановки координат. На первом этапе генерируется начальный план эксперимента. На втором этапе выполняется изменение каждой координаты точек плана с целью минимизации дисперсии коэффициентов уравнения регрессии.

Примеры использования функции генерации матрицы D -оптимального плана на основе алгоритма изменения координат

Генерация матрицы значений факторов D -оптимального плана для трех факторов и 6 опытов. Принимается по умолчанию линейная модель.

```

>> nfacto=3;
>> nruns=6;
>> settings = cordexch(nfacto,nruns)
settings =
    1  -1  -1
    1   1  -1
   -1   1  -1
    1   1   1
   -1  -1   1
   -1  -1  -1

```

Генерация матрицы значений факторов и матрицы D-оптимального плана для трех факторов, 12 опытов и полной квадратической модели.

```

>> nfacto=3;
>> nruns=12;
>> [settings,X] = cordexch(nfacto,nruns,'quadratic')
settings =
   -1   1  -1
    1  -1  -1
   -1  -1   1
   -1   0   0
    0   0  -1
   -1   1   1
   -1  -1  -1
    1   1  -1
    1  -1   1
    1   1   1
    0   1   1
    0  -1   0
X =
    1  -1   1  -1  -1   1  -1   1   1   1
    1   1  -1  -1  -1  -1   1   1   1   1
    1  -1  -1   1   1  -1  -1   1   1   1
    1  -1   0   0   0   0   0   1   0   0
    1   0   0  -1   0   0   0   0   0   1
    1  -1   1   1  -1  -1   1   1   1   1
    1  -1  -1  -1   1   1   1   1   1   1
    1   1   1  -1   1  -1  -1   1   1   1
    1   1  -1   1  -1   1  -1   1   1   1
    1   1   1   1   1   1   1   1   1   1
    1   0   1   1   0   0   1   0   1   1
    1   0  -1   0   0   0   0   0   1   0

```

Генерация матрицы значений факторов и матрицы D-оптимального плана для 3 факторов, 20 опытов и регрессионной модели заданной в виде матрицы model.

```
>> nfactors=3;
>> nruns=20;
>> model=[ 0 1 2; 1 2 3; 0 1 2 ; 1 2 3]
model =
    0    1    2
    1    2    3
    0    1    2
    1    2    3
>> [settings,X] = cordexch(nfactors,nruns,model)
settings =
    1    1    1
    1    1   -1
   -1   -1    1
    1   -1    1
    1   -1   -1
   -1    1   -1
    1    1   -1
   -1   -1    1
   -1   -1   -1
   -1    1    1
    1   -1   -1
    1   -1    1
    1   -1   -1
    1    1    1
   -1    1   -1
   -1   -1   -1
    1    1    1
    1   -1    1
    1    1    1
   -1   -1    1
X =
    1    1    1    1
    1   -1    1   -1
   -1   -1   -1   -1
   -1    1   -1    1
   -1   -1   -1   -1
    1    1    1    1
    1   -1    1   -1
   -1   -1   -1   -1
   -1    1   -1    1
    1   -1    1   -1
```

```

-1 -1 -1 -1
-1 1 -1 1
-1 -1 -1 -1
1 1 1 1
1 1 1 1
-1 1 -1 1
1 1 1 1
-1 1 -1 1
1 1 1 1
-1 -1 -1 -1

```

Генерация матриц значений факторов и D-оптимального плана для трех факторов, 12 опытов, полной квадратической модели. В качестве дополнительного входного аргумента задается матрица начального приближения A.

```

>> nfactors=3;
>> nruns=12;
>> A=[0 0 0; 1 0 0; -1 0 0; 0 1 0; 1 1 -1; -1 0 -1; 1 0 0; 0 0 1; -1 1 -1; 1 0 -1; 1 1
1; 0 -1 0]

```

```

A =
0 0 0
1 0 0
-1 0 0
0 1 0
1 1 -1
-1 0 -1
1 0 0
0 0 1
-1 1 -1
1 0 -1
1 1 1
0 -1 0

```

```

>> [settings,X] = cordexch(nfactors,nruns,'quadratic','init',A)

```

```

settings =
0 0 -1
1 -1 -1
-1 -1 1
-1 1 1
1 1 -1
-1 -1 -1
-1 0 0
1 -1 1
-1 1 -1
1 0 0

```

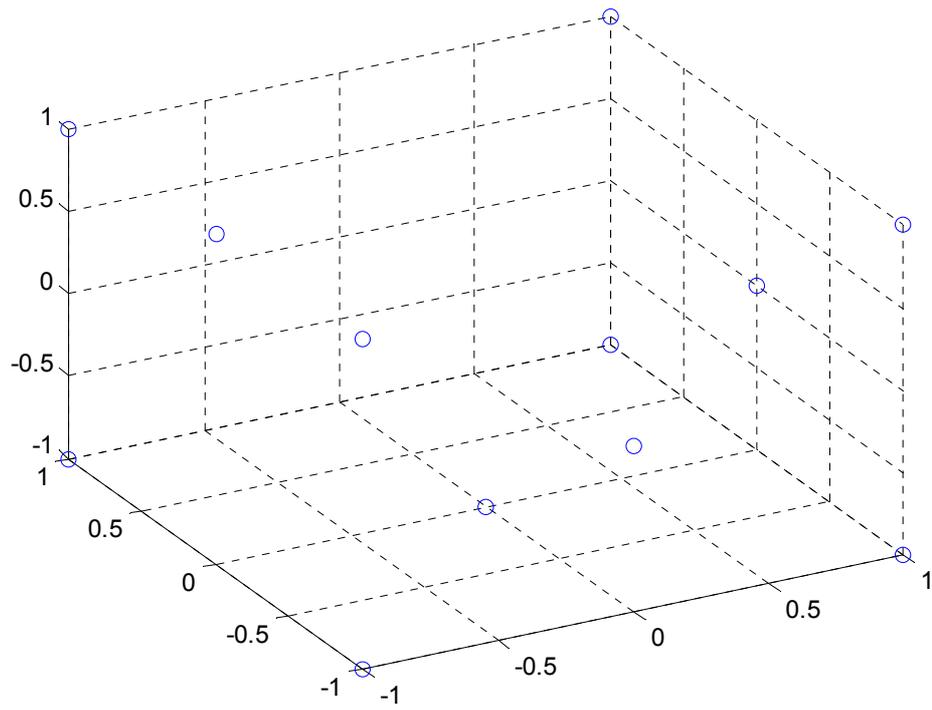
$$\begin{matrix}
 & 1 & 1 & 1 \\
 & 0 & -1 & 0 \\
 X = & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\
 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\
 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 \\
 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\
 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 \\
 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0
 \end{matrix}$$

Графическое представление матрицы значений факторов D-оптимального плана для трех факторов, 12 опытов и полной квадратической модели

```

>> nfactors=3;
>> nruns=12;
>> settings = cordexch(nfactors,nruns,'quadratic')
>> x= settings (:,1);
>> y= settings (:,2);
>> z= settings (:,3);
>> plot3(x,y,z,'o')
>> grid on

```



daugment

Дополнение матрицы значений факторов до D-оптимального плана эксперимента

Синтаксис

```
settings = daugment(startdes, nruns)
[settings, X] = daugment(startdes, nruns)
[settings, X] = daugment(startdes, nruns, 'model')
[settings, X] =
daugment(..., 'param1', value1, 'param2', value2, ...)
```

Описание

`settings = daugment(startdes, nruns)` функция предназначена для дополнения `nruns` опытов к входной матрице `startdes` с целью получения D-оптимальной матрицы значений факторов плана эксперимента `settings`. Функция основана на алгоритме изменения координат.

`[settings, X] = daugment(startdes, nruns)` в отличии от первого варианта синтаксиса функция возвращает матрицу `X` D-оптимального плана эксперимента.

`[settings, X] = daugment(startdes, nruns, 'model')` функция позволяет получить матрицу значений факторов `settings` и матрицу D-оптимального плана эксперимента `X` на основе алгоритма изменения координат для исходной матрицы значений факторов `startdes`, заданного числа опытов `nruns` и вида уравнения регрессии `'model'`. Возможны следующие модели регрессии

Значение <code>'model'</code>	Вид математической модели
<code>'linear'</code>	Линейная модель с постоянным членом. Значение по умолчанию.
<code>'interaction'</code>	Линейная модель с постоянным членом и эффектами взаимодействия.
<code>'quadratic'</code>	Полная квадратическая модель, включающая линейные, квадратические эффекты, эффекты взаимодействий, постоянный член.
<code>'purequadratic'</code>	Неполная квадратическая модель, включающая линейные, квадратические эффекты, постоянный член.

Кроме строкового значения входной аргумент `model` может быть задан как вектор или матрица согласно такому же аргументу функции `x2fx`.

Функция `x2fx` позволяет выполнить преобразование матрицы значений факторов X в матрицу плана эксперимента D . В случае, если X и `model` заданы как векторы, то матрица плана эксперимента формируется по правилу: каждый столбец D является последовательным возведением X в степень элемента вектора `model`. Размерность матрицы D равна $n \times m$, где n – число элементов вектора X , m – число элементов вектора `model`. Т.е., вектор `model` является списком степеней полинома регрессионной модели для одного фактора X . Если X и `model` заданы как матрицы, то столбец матрицы D формируются по формуле:

$$D(i,j)=\text{prod}(X(i,:).^{\text{model}(j,:)}),$$

т.е., ij -й элемент матрицы D является произведением элементов i -й строки матрицы X возведенных последовательно в степени j -й строки матрицы `model`. Таким образом, количество столбцов `model` должно быть равно числу столбцов матрицы X .

`[settings, X] = daugment(..., 'param1', value1, 'param2', value2, ...)`
 дополнительные входные аргументы по отношению к предыдущим вариантам синтаксиса позволяют:

'display'	Вывод значения счетчика итераций. Возможные значения 'display': 'on' – вывод в командное окно, 'off' – отмена отображения. Значение по умолчанию 'on'.
'init'	Начальная матрица значений факторов с размерностью <code>nruns</code> × <code>nfactors</code> . По умолчанию начальное множество точек в факторном пространстве выбирается случайным образом.
'maxiter'	Максимальное число итераций. Значение по умолчанию - 10.

Примеры использования функции дополнения матрицы значений факторов до D -оптимального плана эксперимента

Дополнение матрицы значений факторов полного факторного эксперимента 2^2 до D -оптимальной матрицы значений факторов. Модель регрессии принимается по умолчанию линейной. Количество добавляемых опытов 5.

```
>> nruns=5;
>> startdes = [-1 -1; 1 -1; -1 1; 1 1]
startdes =
    -1    -1
     1    -1
    -1     1
     1     1
```

```
>> settings = daugment(startdes,nruns)
```

```
settings =
```

```
-1 -1  
1 -1  
-1 1  
1 1  
1 -1  
-1 -1  
-1 1  
1 1  
-1 -1
```

Дополнение матрицы значений факторов полного факторного эксперимента 2^2 до D-оптимальной матрицы значений факторов и матрицы плана эксперимента для полной квадратической модели. Количество добавляемых опытов 5.

```
>> nruns=5;
```

```
>> startdes = [-1 -1; 1 -1; -1 1; 1 1]
```

```
startdes =
```

```
-1 -1  
1 -1  
-1 1  
1 1
```

```
>> [settings,X] = daugment(startdes,nruns,'quadratic')
```

```
settings =
```

```
-1 -1  
1 -1  
-1 1  
1 1  
-1 0  
1 0  
0 1  
0 -1  
0 0
```

```
X =
```

```
1 -1 -1 1 1 1  
1 1 -1 -1 1 1  
1 -1 1 -1 1 1  
1 1 1 1 1 1  
1 -1 0 0 1 0  
1 1 0 0 1 0  
1 0 1 0 0 1  
1 0 -1 0 0 1  
1 0 0 0 0 0
```

Дополнение матрицы значений факторов полного факторного эксперимента 2^2 до D-оптимальной матрицы значений факторов и матрицы плана эксперимента. Количество добавляемых опытов 5. Модель регрессии задается в виде матрицы model.

```
>> nruns=5;
>> startdes = [-1 -1; 1 -1; -1 1; 1 1]
startdes =
    -1    -1
     1    -1
    -1     1
     1     1
>> model=[0 1; 1 2; 2 3]
model =
     0     1
     1     2
     2     3
>> [settings,X] = daugment(startdes,nruns,model)
settings =
-1.0000 -1.0000
 1.0000 -1.0000
-1.0000  1.0000
 1.0000  1.0000
-1.0000  1.0000
 0.0184  1.0000
-0.3333 -1.0000
-0.3333  1.0000
-1.0000 -1.0000
X =
-1.0000 -1.0000 -1.0000
-1.0000  1.0000 -1.0000
 1.0000 -1.0000  1.0000
 1.0000  1.0000  1.0000
 1.0000 -1.0000  1.0000
 1.0000  0.0184  0.0003
-1.0000 -0.3333 -0.1111
 1.0000 -0.3333  0.1111
-1.0000 -1.0000 -1.0000
```

Дополнение матрицы значений факторов полного факторного эксперимента 2^2 до D-оптимальной матрицы значений факторов и матрицы плана эксперимента для полной квадратической модели. Количество добавляемых опытов 5. В качестве дополнительного входного аргумента задается матрица начального приближения A.

```
>> nruns=5;
```

```

>> startdes = [-1 -1; 1 -1; -1 1; 1 1]
startdes =
    -1    -1
     1    -1
    -1     1
     1     1
>> A=[0 0; 0 1; 1 0; 1 1; -1 1]
A =
     0     0
     0     1
     1     0
     1     1
    -1     1
>> [settings,X] = daugment(startdes,nruns,'quadratic','init',A)
settings =
    -1    -1
     1    -1
    -1     1
     1     1
     0     0
     0    -1
     1     0
     0     1
    -1     0
X =
     1    -1    -1     1     1     1
     1     1    -1    -1     1     1
     1    -1     1    -1     1     1
     1     1     1     1     1     1
     1     0     0     0     0     0
     1     0    -1     0     0     1
     1     1     0     0     1     0
     1     0     1     0     0     1
     1    -1     0     0     1     0

```

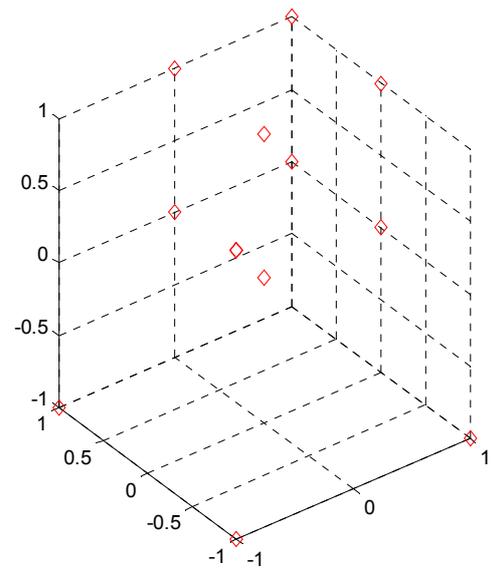
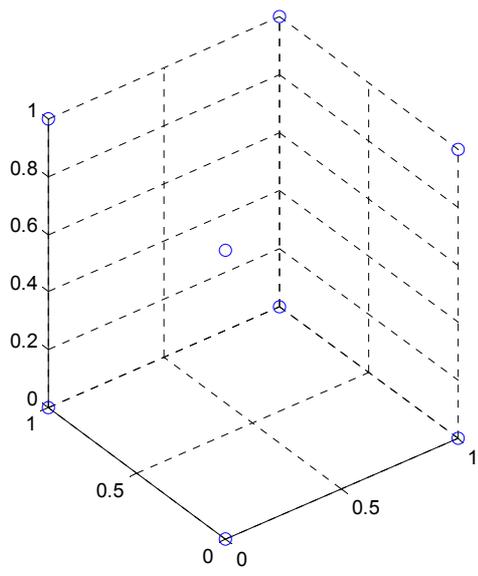
Графическое представление матрицы значений факторов полного факторного эксперимента 2^3 X и дополненной до D-оптимальной матрицы значений факторов settings. Количество добавляемых опытов 5.

```

>> startdes = ff2n(3)
startdes =
     0     0     0
     0     0     1
     0     1     0
     0     1     1

```

```
1 0 0
1 0 1
1 1 0
1 1 1
>> nruns=5;
>> settings = daugment(startdes,nruns)
settings =
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1
-1 -1 1
1 -1 -1
-1 1 -1
-1 -1 -1
-1 -1 1
>> x= X (:,1);
>> y= X (:,2);
>> z= X (:,3);
>> subplot(1,2,1)
>> plot3(x,y,z,'o')
>> grid on
>> x1= settings (:,1);
>> y1= settings (:,2);
>> z1= settings (:,3);
>> subplot(1,2,2)
>> plot3(x1,y1,z1,'dr')
>> grid on
```



dcovary

Генерация D-оптимального блочного плана

Синтаксис

```
settings = dcovary(nfactors,covariates)
[settings,X] = dcovary(nfactors,covariates)
[settings,X] = dcovary(nfactors,covariates,'model')
[settings,X] =
dcovary(...,'param1',value1,'param2',value2,...)
```

Описание

`settings = dcovary(nfactors,covariates)` функция позволяет получить матрицу значений факторов `settings` D-оптимального плана разделенного на блоки `covariates` для заданного числа факторов `nfactors`. Количество опытов в плане эксперимента задается числом строк в матрице `covariates`. Матрица значений факторов `settings` формируется последовательно из двух матриц: матрицы значений факторов D-оптимального плана и матрицы, кодирующей блоки плана. Размерность матрицы `settings` равна $n \times m$, где n – количество строк матрицы `covariates`, $m = nfactors + k$, где k – количество столбцов матрицы `settings`. Функция основана на алгоритме изменения координат.

`[settings,X] = dcovary(nfactors,covariates)` в отличии от первого варианта синтаксиса функция возвращает матрицу D-оптимального плана `X`.

`[settings,X] = dcovary(nfactors,covariates,'model')` функция позволяет получить матрицу значений факторов `settings` и матрицу D-оптимального плана эксперимента `X` на основе на основе матрицы `covariates`, задающей блоки и количество опытов, заданного числа факторов `nfactors`, и вида уравнения регрессии `'model'`. Возможны следующие модели регрессии

Значение <i>'model'</i>	Вид математической модели
'linear'	Линейная модель с постоянным членом. Значение по умолчанию.
'interaction'	Линейная модель с постоянным членом и эффектами взаимодействия.
'quadratic'	Полная квадратическая модель, включающая линейные, квадратические эффекты, эффекты взаимодействий, постоянный член.
'purequadratic'	Неполная квадратическая модель, включающая

	линейные, квадратические эффекты, постоянный член.
--	--

Кроме строкового значения входной аргумент `model` может быть задан как вектор или матрица согласно такому же аргументу функции `x2fx`. Функция `x2fx` позволяет выполнить преобразование матрицы значений факторов X в матрицу плана эксперимента D . В случае, если X и `model` заданы как векторы, то матрица плана эксперимента формируется по правилу: каждый столбец D является последовательным возведением X в степень элемента вектора `model`. Размерность матрицы D равна $n \times m$, где n – число элементов вектора X , m – число элементов вектора `model`. Т.е., вектор `model` является списком степеней полинома регрессионной модели для одного фактора X . Если X и `model` заданы как матрицы, то столбец матрицы D формируются по формуле:

$$D(i,j)=\text{prod}(X(i,:).^{\text{model}(j,:)}),$$

т.е., ij -й элемент матрицы D является произведением элементов i -й строки матрицы X возведенных последовательно в степени j -й строки матрицы `model`. Таким образом, количество столбцов `model` должно быть равно числу столбцов матрицы X .

`[settings, X] = dcovary(..., 'param1', value1, 'param2', value2, ...)`
 дополнительные входные аргументы по отношению к предыдущим вариантам синтаксиса позволяют:

'display'	Вывод значения счетчика итераций. Возможные значения 'display': 'on' – вывод в командное окно, 'off' – отмена отображения. Значение по умолчанию 'on'.
'init'	Начальная матрица значений факторов с размерностью $n_{\text{runs}} \times n_{\text{factors}}$. По умолчанию начальное множество точек в факторном пространстве выбирается случайным образом.
'maxiter'	Максимальное число итераций. Значение по умолчанию - 10.

Примечание. Функция `dcovary` выполняет поиск D -оптимального плана на основе алгоритма перестановки координат. На первом этапе генерируется начальный план эксперимента, содержащий постоянные кодовые значения, разделяющие матрицу плана на блоки, и переменные значения – значения уровней факторов. На втором этапе выполняется изменение для каждой переменной координат точек плана с целью минимизации дисперсии коэффициентов уравнения регрессии.

Примеры использования функции генерации D-оптимального блочного плана

Генерация матрицы значений факторов D-оптимального плана для 3 факторов разделенного на 2 блока по 4 опыта в каждом. Деление на блоки задается посредством векторной переменной `covariates`: ' -1^a кодирует первый блок, ' 1^a кодирует второй блок. Четвертый столбец в матрице `settings` является блоковой переменной.

```
>> nfactors=3;
>> covariates=[-1 -1 -1 -1 1 1 1 1]'
covariates =
  -1
  -1
  -1
  -1
   1
   1
   1
   1
>> settings = dcovary(nfactors,covariates)
settings =
   1  -1   1  -1
   1   1  -1  -1
  -1  -1  -1  -1
  -1   1   1  -1
   1  -1   1   1
  -1   1   1   1
  -1  -1  -1   1
   1   1  -1   1
```

Генерация матрицы значений факторов и матрицы D-оптимального плана для 4 факторов разделенного на 3 блока по 3 опыта в каждом. Деление на блоки задается посредством векторной переменной `covariates`.

```
>> nfactors=4;
>> covariates=[-1 -1 -1 1 1 1 2 2 2]'
covariates =
  -1
  -1
  -1
   1
   1
   1
   2
   2
   2
```

```

2
2
2
>> [settings,X] = dcovary(nfactors,covariates)
settings =
  1  1 -1 -1 -1
 -1  1  1 -1 -1
 -1 -1 -1  1 -1
  1 -1  1  1  1
 -1  1  1  1  1
 -1  1 -1 -1  1
 -1 -1 -1 -1  2
  1  1 -1  1  2
  1 -1  1 -1  2

```

```

X =
  1  1  1 -1 -1 -1
  1 -1  1  1 -1 -1
  1 -1 -1 -1  1 -1
  1  1 -1  1  1  1
  1 -1  1  1  1  1
  1 -1  1 -1 -1  1
  1 -1 -1 -1 -1  2
  1  1  1 -1  1  2
  1  1 -1  1 -1  2

```

Генерация матрицы значений факторов и матрицы D-оптимального плана для 2 факторов разделенного на 4 блока по 2 опыта в каждом. Деление на блоки задается посредством матрицы covariates. Матрица covariates формируется с помощью функции dummyvar преобразованием вектора [1 1 2 2 3 3 4 4]. В качестве уравнения регрессии принимается линейная модель.

```

>> covariates = dummyvar([1 1 2 2 3 3 4 4])
covariates =
  1  0  0  0
  1  0  0  0
  0  1  0  0
  0  1  0  0
  0  0  1  0
  0  0  1  0
  0  0  0  1
  0  0  0  1
>> nfactors=2;
>> [settings,X] = dcovary(nfactors,covariates(:,1:3),'linear')
settings =

```

```

-1  1  1  0  0
 1 -1  1  0  0
 1  1  0  1  0
-1 -1  0  1  0
 1 -1  0  0  1
-1  1  0  0  1
 1  1  0  0  0
-1 -1  0  0  0
X =
 1 -1  1  1  0  0
 1  1 -1  1  0  0
 1  1  1  0  1  0
 1 -1 -1  0  1  0
 1  1 -1  0  0  1
 1 -1  1  0  0  1
 1  1  1  0  0  0
 1 -1 -1  0  0  0

```

Генерация матрицы значений факторов и матрицы D-оптимального плана для 2 факторов разделенного на 4 блока по 2 опыта в каждом. Деление на блоки задается посредством вектора covariates. Модель регрессии задается как матрица model.

```

>> covariates=[1 1 2 2 3 3 4 4]';
>> nfactors=2;
>> model=[0 1 2; 0 1 2; 0 1 2]
model =
 0  1  2
 0  1  2
 0  1  2
>> [settings,X] = dcovary(nfactors,covariates, model)
settings =
-0.9852 -1.0000  1.0000
 0.5776 -1.0000  1.0000
-0.9644 -1.0000  2.0000
 0.7559 -1.0000  2.0000
-0.2949 -1.0000  3.0000
 0.4443 -1.0000  3.0000
 0.9369 -1.0000  4.0000
-0.6887 -1.0000  4.0000
X =
-1 -1 -1
-1 -1 -1
-4 -4 -4
-4 -4 -4

```

```

-9 -9 -9
-9 -9 -9
-16 -16 -16
-16 -16 -16

```

Генерация матрицы значений факторов и матрицы D-оптимального плана для 3 факторов разделенного на 3 блока по 3 опыта в каждом. Деление на блоки задает вектор covariates. В качестве дополнительного параметра используется матрица начального приближения A.

```

>> nfactors=3;
>> covariates=[-1 -1 -1 1 1 1 2 2 2]';
>> A=[ 0 0 0; 1 0 0; 0 1 0; -1 0 0; -1 -1 0; 0 0 -1; 1 -1 1; -1 -1 1; 1 -1 -1]

```

```

A =
  0  0  0
  1  0  0
  0  1  0
 -1  0  0
 -1 -1  0
  0  0 -1
  1 -1  1
 -1 -1  1
  1 -1 -1

```

```

>> [settings,X] = dcovary(nfactors,covariates,'purequadratic')

```

```

settings =
  1 -1 -1 -1
 -1  0  1 -1
  0  1  0 -1
 -1 -1  1  1
  1  0  0  1
  0  0 -1  1
 -1  1 -1  2
  0 -1  0  2
  1  1  1  2

```

```

X =
  1  1 -1 -1 -1  1  1  1  1
  1 -1  0  1 -1  1  0  1  1
  1  0  1  0 -1  0  1  0  1
  1 -1 -1  1  1  1  1  1  1
  1  1  0  0  1  1  0  0  1
  1  0  0 -1  1  0  0  1  1
  1 -1  1 -1  2  1  1  1  4
  1  0 -1  0  2  0  1  0  4
  1  1  1  1  2  1  1  1  4

```

Графическое представление матрицы D-оптимального плана эксперимента для 3 факторов, разделенного на 2 блока по 5 опытов в каждом. Матрица D-оптимального плана рассчитывается для неполной квадратической модели.

```
>> nfactors=3;
>> covariates=[0 0 0 0 1 1 1 1 1];
>> settings = dcovary(nfactors,covariates,'purequadratic')
settings =
    -1    1    0    0
    -1    0    1    0
     1   -1    1    0
     1    1   -1    0
     0    0   -1    0
     0    1    1    1
     1    0    0    1
     0   -1    0    1
    -1    1   -1    1
    -1   -1   -1    1
```

Первый блок

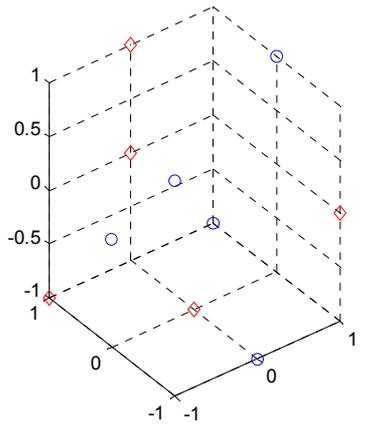
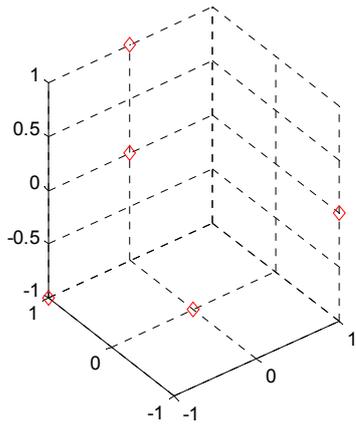
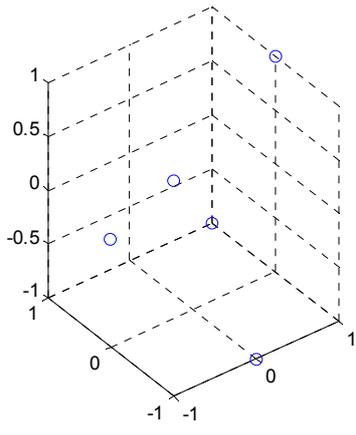
```
>> x= settings (1:5,1);
>> y= settings (1:5,2);
>> z= settings (1:5,3);
>> subplot(1,3,1)
>> plot3(x,y,z,'o')
>> grid on
```

Второй блок

```
>> x1= settings (6:10,1);
>> y1= settings (6:10,2);
>> z1= settings (6:10,3);
>> subplot(1,3,2)
>> plot3(x1,y1,z1,'dr')
>> grid on
```

Общий график

```
>> subplot(1,3,3)
>> plot3(x,y,z,'o',x1,y1,z1,'dr')
>> grid on
```



rsmdemo

Демонстрационная функция D-оптимального планирования и моделирования результатов эксперимента на примере химического процесса

Синтаксис

rsmdemo

Описание

rsmdemo – функция предназначена для демонстрации методов: D-оптимального планирования эксперимента, регрессионного анализа и представления поверхности отклика для множества факторов, оценки параметров нелинейной модели Хогена. Демонстрация работы с указанными методами проводится на примере химической реакции 3 реагентов. Целью планирования эксперимента является поиск максимума коэффициента выхода полезного продукта. Функция rsmdemo основана на графическом интерфейсе с пользователем.

Интерфейс состоит из 3-х графических окон:

- окна моделирования параметров химической реакции (рис. 1),
- окна результатов измерений (рис. 2),
- окна результатов эксперимента (рис. 3).

Назначение элементов окна моделирования параметров химической реакции:

- Строки ввода Hydrogen, n-Pentane, Isopentane и соответствующие им полосы прокрутки позволяют установить давление водорода, n-пентана и изопентана в химическом реакторе;
- Строка ввода Reaction Rate отображает результат моделирования коэффициента выхода полезного продукта;
- Строка ввода Runs Left показывает количество проведенных опытов. Значения изменяются от 13 до 0;
- Кнопка Run предназначена для моделирования одного опыта и расчета коэффициента выхода полезного продукта при заданных значениях Hydrogen, n-Pentane, Isopentane с учетом случайных отклонений;
- Кнопка Export позволяет создать в рабочей среде matlab матричную переменную значений независимых переменных и вектор коэффициента выхода полезного продукта. Названия переменных задаются в диалоговом окне после выбора кнопки, по умолчанию reactants и rate.

- Кнопки Close и Help предназначены для закрытия окна и вызова помощи.

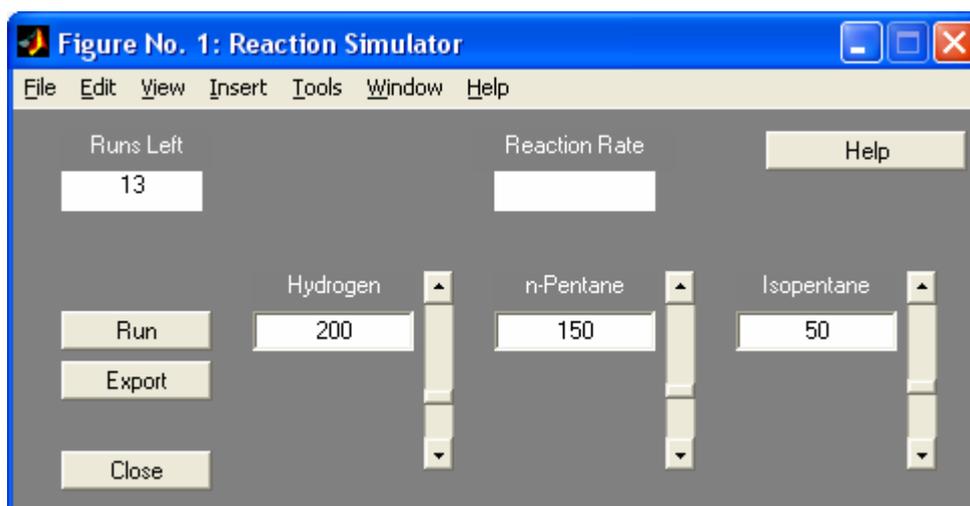


Рис. 1. Окно моделирования параметров химической реакции

Окно результатов измерений отображает результаты моделирования для опытов, проводимых пользователем по своему усмотрению в виде таблицы. Кнопка Analyze вызывает графическое окно функции rstool для представления результатов регрессионного анализа. Меню Plot позволяет построить графики зависимости коэффициента выхода реакции от величины давления каждого из компонентов.

Run #	Hydrogen	n-Pentane	Isopentane	Reaction Rate
1	200	150	50	5.49
2	237	150	50	4.71
3	237	128	50	4.33
4	237	128	72	3.34
5	163	128	72	3.68
6	163	172	72	6.00
7	163	172	94	4.80
8	163	128	94	2.87
9	144	128	94	3.05
10	144	273	94	8.07
11	144	273	48	11.13
12	144	186	48	8.48
13	181	186	48	7.21

Рис. 2. Окно результатов измерений

Окно результатов эксперимента предназначено для D-оптимального планирования и моделирования результатов 3-х факторного эксперимента в автоматическом режиме. Кнопка Do Experiment запускает процесс планирования и моделирования результатов эксперимента. Кнопка Response Surface вызывает графическое окно функции rstool для отображения результатов регрессионного анализа.

Кнопка Nonlinear Model вызывает графическое окно функции nlintool для отображения результатов регрессионного анализа по модели Хогена-Ватсона. Меню Plot позволяет построить графики зависимости коэффициента выхода реакции от величины давления каждого из компонентов.

Run #	Hydrogen	n-Pentane	Isopentane	Reaction Rate
1	100	80	10	8.66
2	285	300	120	5.90
3	100	300	10	18.38
4	470	300	65	6.63
5	470	300	10	8.41
6	470	190	120	2.55
7	100	300	120	7.64
8	100	80	120	0.05
9	100	190	65	7.40
10	285	190	10	8.56
11	470	80	120	0.03
12	470	80	10	2.50
13	285	80	65	1.57

Рис. 3. Окно результатов эксперимента

Функция rsmdemo позволяет сравнить эффективность применения метода планирования эксперимента по сравнению со случайным выбором параметров и проведением опытов при одинаковом их количестве. Под случайным выбором параметров реакции понимается проведение пассивных наблюдений за характеристиками химической реакции или поиск оптимальных условий реакции по усмотрению пользователя.

На первом этапе демонстрации пользователь в окне моделирования параметров химической реакции устанавливает значения давления реагентов в химическом реакторе. При нажатии кнопки Run выполняется расчет коэффициента выхода полезного продукта. Результат расчета отображается в строке ввода Reaction Rate. При расчете коэффициента выхода учитываются как значения давления реагентов, так и воздействие случайных факторов. Принимается, что распределение значений коэффициента выхода подчиняется нормальному закону с параметрами: математическое ожидание 1, среднее квадратическое отклонение 0,05. Расчет коэффициента выхода у выполняется по формуле

$$y = 1.25 * (p_2 - p_3 / 1.5183) / (1 + 0.064 * p_1 + 0.0378 * p_2 + 0.1326 * p_3) * \text{normrnd}(1, 0.05), \quad (1)$$

где p_1 – давление водорода, p_2 – давление n-пентана, p_3 – давление изопентана.

Таким образом, при одинаковых величинах факторов значения коэффициента выхода полезного продукта будут изменяться от опыта к опыту.

Установленные значения факторов и результаты моделирования коэффициента выхода отображаются в окне результатов измерений. Пользователь должен провести 13 опытов.

По окончании моделирования опытов пользователь выполняет регрессионный анализ экспериментальных данных. С этой целью в окне результатов измерений предусмотрены кнопка Analyze и меню Plot. При выборе кнопки Analyze для проведения регрессионного анализа вызывается функция `rstool(x,y,[],[],xname,yname)`. Технология графического представления данных функции `rstool` предназначена для построения зависимости одной независимой переменной от множества независимых переменных (3 и более). Регрессионный анализ при помощи функции `rstool` предусматривает возможности выбора линейной, не полной квадратической, полной квадратической моделей и линейной модели с эффектами взаимодействий факторов. Выбор команды в меню Plot позволяет получить декартовы графики зависимостей коэффициента выхода от давления каждого их компонентов в отдельности.

Второй этап включает 2 фазы: D-оптимальное планирование и моделирование результатов эксперимента по формуле (1). Планирование и моделирование эксперимента проводится в автоматическом режиме после нажатия кнопки Do Experiment. Формирование матрицы значений факторов проводится с использованием алгоритма изменения координат для 3 факторов, 13 опытов и полной квадратической модели. С этой целью используется функция: `settings=cordexch(3,13,'q')`. Согласно полученной матрице уровней факторов по формуле (1) проводится расчет значений коэффициента выхода полезного продукта. Матрица уровней факторов и вектор значений коэффициента выхода выводятся в табличном виде в окне результатов эксперимента.

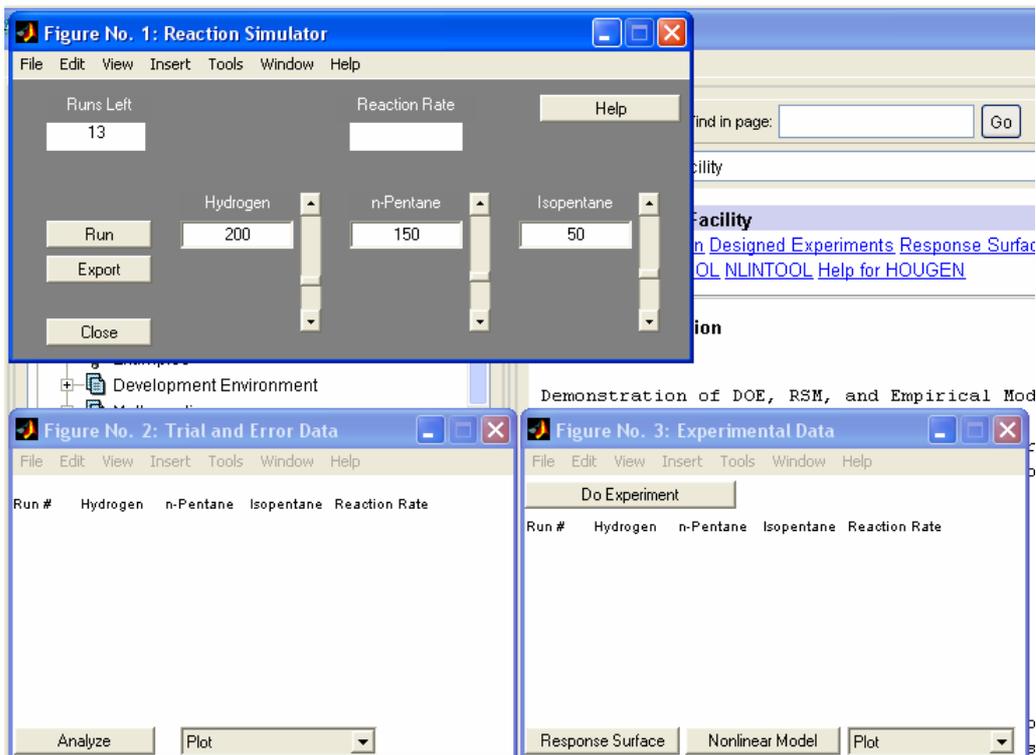
По окончании моделирования опытов пользователь выполняет регрессионный анализ экспериментальных данных. С этой целью в окне результатов эксперимента предусмотрены кнопки Response Surface, Nonlinear Model и меню Plot. При выборе кнопки Response Surface вызывается функция `rstool` и анализ проводится аналогично первому этапу. Кнопка Nonlinear Model позволяет в качестве модели химической реакции использовать функцию Хогена-Ватсона:

$$y = (b1*x2 - x3/b5)./(1+b2*x1+b3*x2+b4*x3).$$

Расчет коэффициентов модели Хогена-Ватсона выполняется с использованием функции `nlintool`: `nlintool(x,y,'hougen',beta0,[],xname,yname)`. Способ отображения результатов регрессионного анализа функции `nlintool` аналогична `rstool`.

Пример использования функции `rsmdemo`

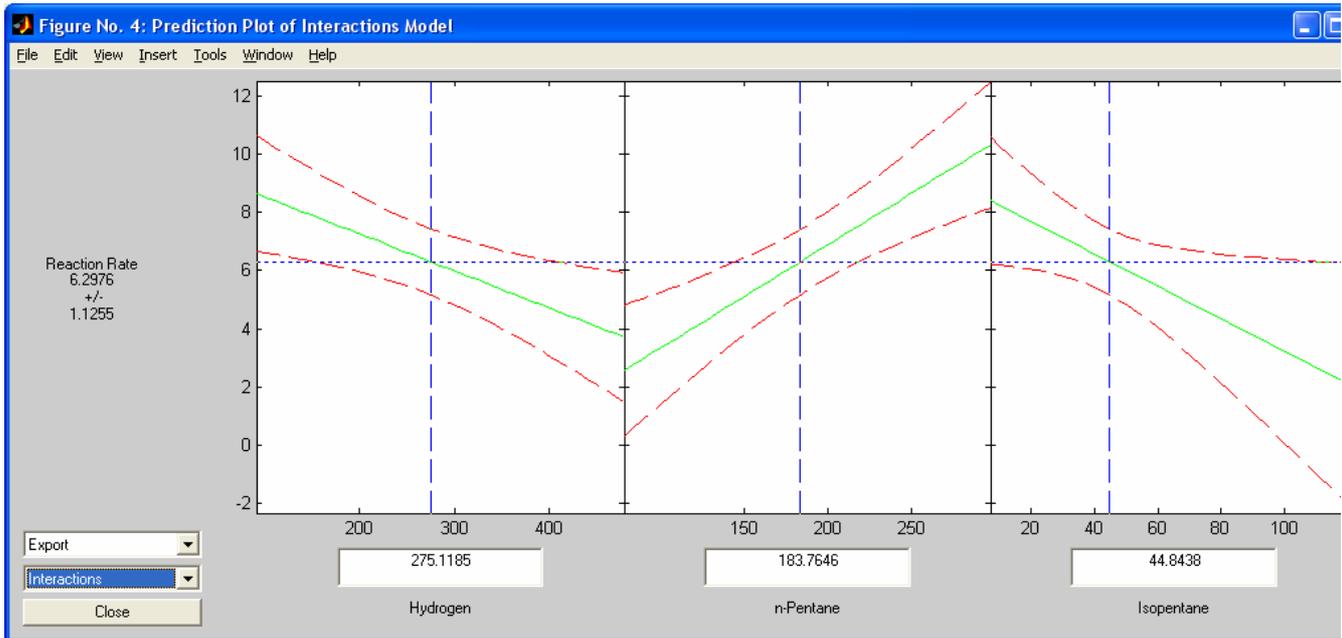
`>> rsmdemo`



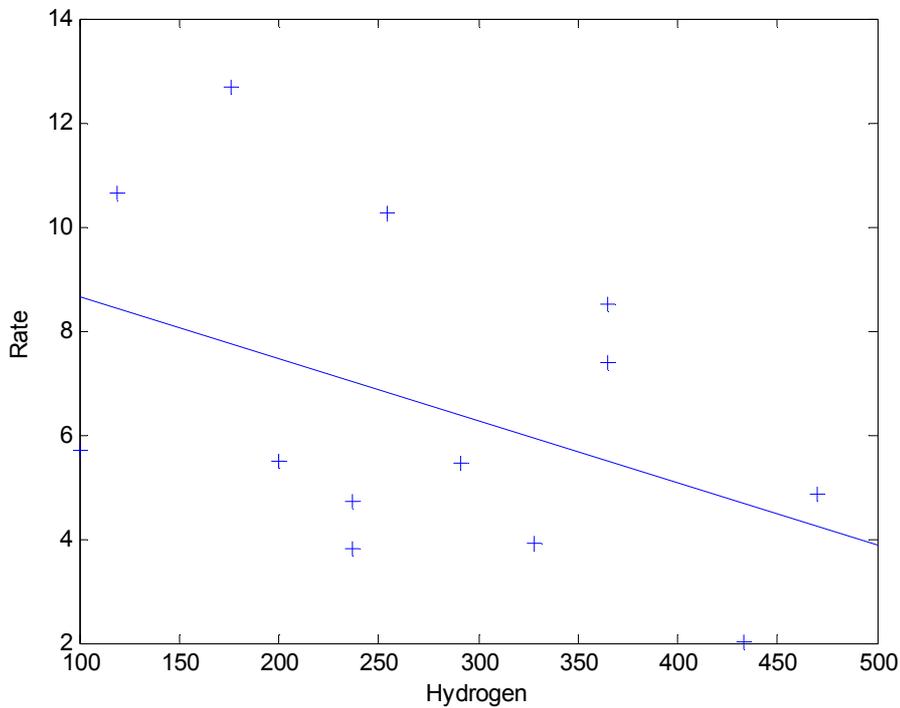
Моделирование пассивных наблюдений за химической реакцией. Результаты 13 наблюдений приведены на следующем рисунке

Run #	Hydrogen	n-Pentane	Isopentane	Reaction Rate
1	200	150	50	5.49
2	237	150	50	4.71
3	237	128	61	3.80
4	178	277	25	12.67
5	100	84	25	5.69
6	470	292	120	4.87
7	433	125	86	2.03
8	119	190	31	10.65
9	291	146	31	5.45
10	328	168	74	3.92
11	254	234	10	10.25
12	365	234	10	8.52
13	365	212	10	7.38

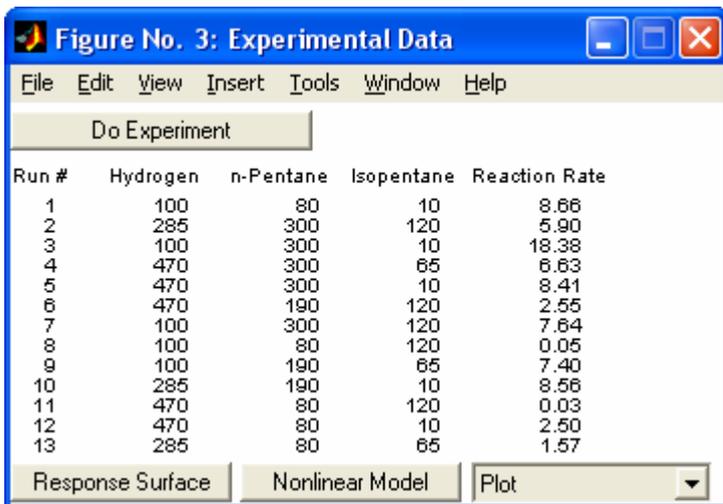
Регрессионный анализ полученных данных для линейной модели с эффектами взаимодействий (кнопка Analyze)



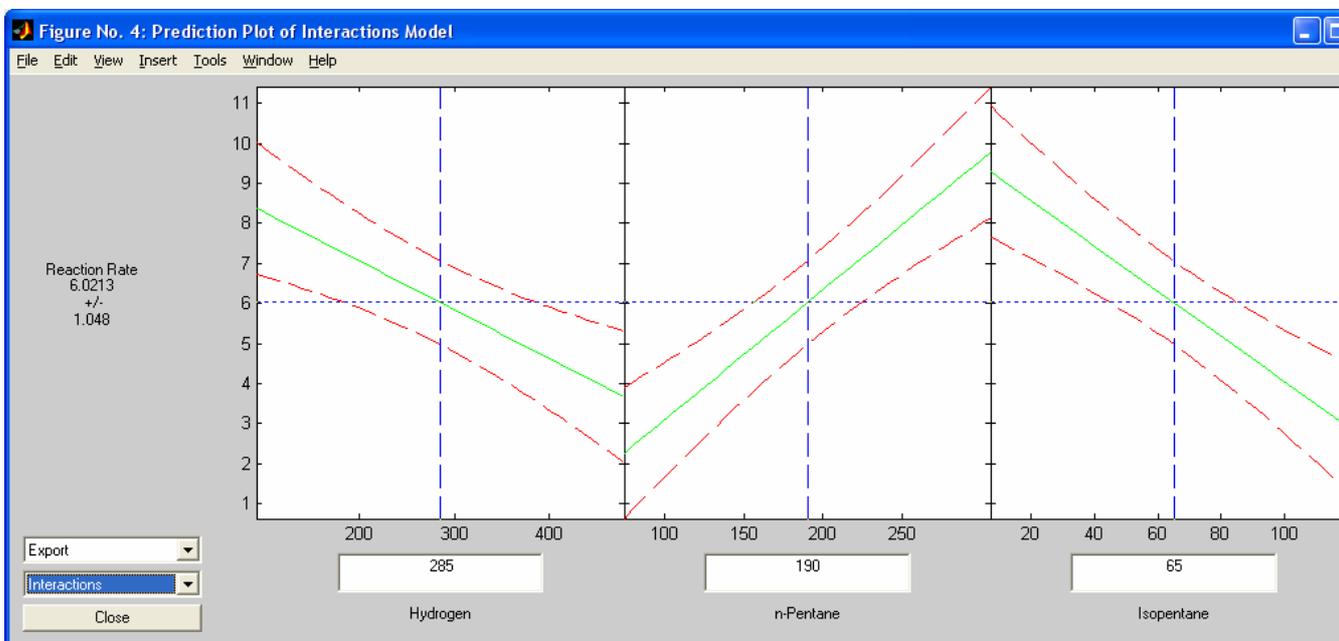
Зависимость коэффициента выхода полезного продукта от давления водорода (Меню Plot, Reaction Rate vs. Hydrogen)



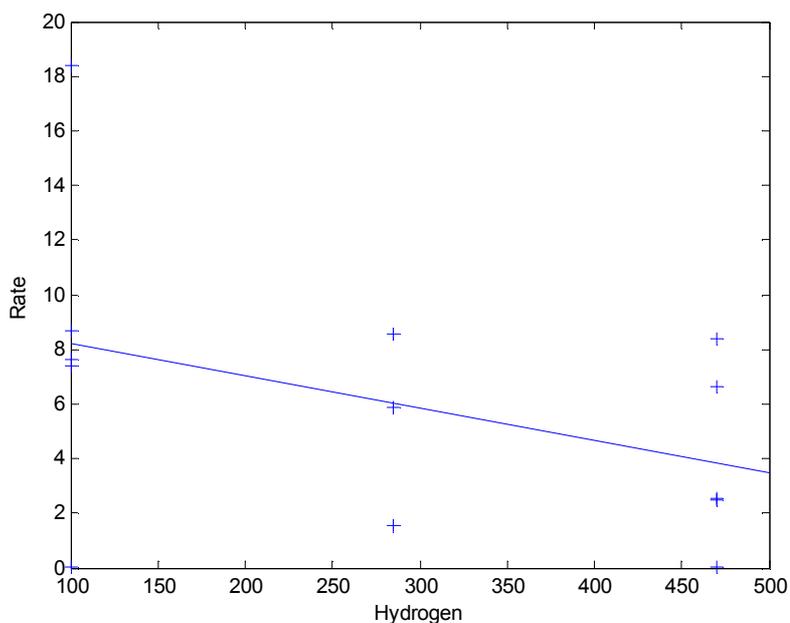
Планирование и моделирование результатов эксперимента (Кнопка Do Experiment)



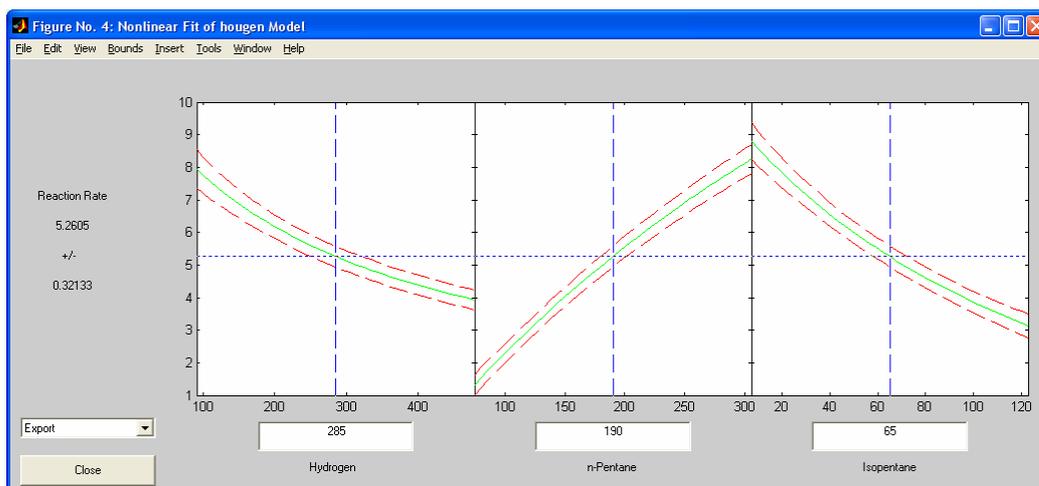
Регрессионный анализ результатов эксперимента для линейной модели с эффектами взаимодействия (кнопка Response Surface)



Зависимость коэффициента выхода полезного продукта от давления водорода (Меню Plot, Reaction Rate vs. Hydrogen)



Регрессионный анализ результатов эксперимента для модели Хогена-Ватсона (кнопка Nonliner Model)



Из анализа приведенных графиков следует, что применение D-оптимального планирования эксперимента обеспечивает получение оценки коэффициентов уравнения регрессии с меньшей дисперсией (меньшим доверительным интервалом). Использование модели Хогена-Ватсона обеспечивает меньшую погрешность при описании процесса протекания химической реакции.

Статистический контроль качества

xbarplot

Контрольная карта средних арифметических значений (\bar{X} контрольная карта)

Синтаксис

```
xbarplot (DATA)
xbarplot (DATA, conf)
xbarplot (DATA, conf, specs)
xbarplot (DATA, conf, specs, 'sigmaest ')
[outlier, h] = xbarplot (...)
```

Описание

`xbarplot (DATA)` функция позволяет получить контрольную карту средних арифметических значений (\bar{X}) для выборки DATA. Размерность матрицы DATA равна $n \times m$, где n – количество выборок (строк DATA), m – объем выборки (число столбцов DATA). Последовательность выборок должна соответствовать порядку сбора исходных данных. На графике контрольной карты отображаются выборочные средние значения \bar{X}_i , $i=1..n$, центральная линия, соответствующая общему среднему арифметическому $\bar{\bar{X}}$, верхняя UCL и нижняя LCL контрольные границы.

Общее среднее арифметическое рассчитывается по формуле:

$$\bar{\bar{X}} = \frac{1}{n} \sum_{i=1}^n \bar{X}_i .$$

Верхняя и нижняя контрольные границы определяются как

$$UCL = \bar{\bar{X}} + 3\sigma_{\bar{X}}, \quad LCL = \bar{\bar{X}} - 3\sigma_{\bar{X}},$$

$\sigma_{\bar{X}}$ - среднее квадратическое отклонение $\bar{\bar{X}}$.

Если технологический процесс является статистически управляемым, то при 1000 выборках количество точек вышедших за контрольные пределы не должно превышать 3-х в случайном порядке. Таким образом, при малом количестве выборок выход выборочного среднего арифметического \bar{X}_i за контрольные границы означает потерю процессом статистической управляемости.

`xbarplot(DATA, conf)` функция предназначена для построения \bar{X} контрольной карты для выборки DATA с заданной доверительной вероятностью `conf` для верхней и нижней контрольных границ. По умолчанию значение `conf` принимается равной 0.9973. Величина `conf=0.9973` соответствует интервалу ± 3 средних квадратических отклонений общего среднего:

```
>> norminv(1 - (1-.9973)/2)
ans =
    3.0000
```

Для расчета доверительной вероятности `conf` соответствующей $\pm k\sigma_{\bar{x}}$ используется выражение $1-2*(1-normcdf(k))$. Например, для $k=2$ значение `conf` равно

```
>> k = 2;
>> conf = 1-2*(1-normcdf(k))
conf =
    0.9545
```

`xbarplot(DATA, conf, specs)` функция предназначена для построения контрольной карты средних арифметических значений с заданной доверительной вероятностью расположения контрольных границ `conf` и границами допусков на параметр технологического процесса `specs`. Границы допусков определяются как вектор с двумя элементами: `specs(1)` – нижняя граница допуска, `specs(2)` – верхняя граница допуска.

`xbarplot(DATA, conf, specs, 'sigmaest')` позволяет получить \bar{X} контрольную карту с заданной доверительной вероятностью `conf`. Строковый параметр `'sigmaest'` служит для определения способа расчета точечной оценки стандартного отклонения общего среднего. Возможны следующие способы расчета $\sigma_{\bar{x}}$:

Значение 'sigmaest'	Способ расчета точечной оценки стандартного отклонения
'range', 'r'	Для расчета точечной оценки $\sigma_{\bar{x}}$ используется средний выборочный размах. Применяется для объема выборки не более 25 элементов.
'variance', 'v'	Точечная оценка $\sigma_{\bar{x}}$ принимается равной корню квадратному из дисперсии рассчитанной по всем выборкам

's'	Точечная оценка $\sigma_{\bar{x}}$ принимается равной среднему арифметическому средних квадратических отклонений выборок
-----	--

[outlier, h] = xbarplot(DATA, conf, specs) Выходными параметрами функции являются: outlier ñ вектор номеров выборок (строк матрицы DATA), средние арифметические значения которых вышли за контрольные границы, h ñ вектор указателей на объекты графика контрольной карты.

Примеры использования функции построения контрольной карты средних арифметических значений

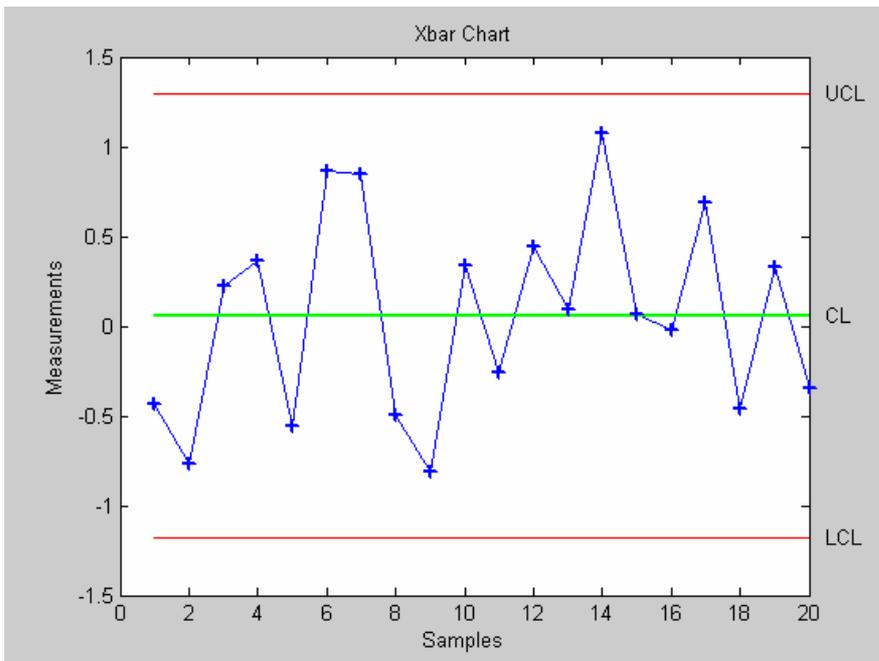
\bar{X} контрольная карта

>>DATA=normrnd(0,1,20,4)

DATA =

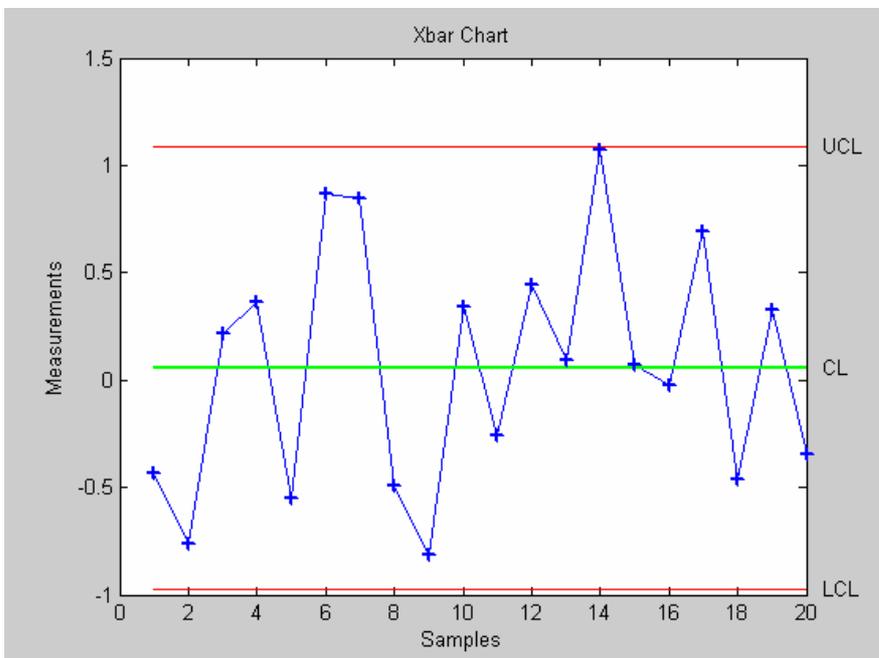
```
-0.4326  0.2944 -1.6041  0.0000
-1.6656 -1.3362  0.2573 -0.3179
 0.1253  0.7143 -1.0565  1.0950
 0.2877  1.6236  1.4151 -1.8740
-1.1465 -0.6918 -0.8051  0.4282
 1.1909  0.8580  0.5287  0.8956
 1.1892  1.2540  0.2193  0.7310
-0.0376 -1.5937 -0.9219  0.5779
 0.3273 -1.4410 -2.1707  0.0403
 0.1746  0.5711 -0.0592  0.6771
-0.1867 -0.3999 -1.0106  0.5689
 0.7258  0.6900  0.6145 -0.2556
-0.5883  0.8156  0.5077 -0.3775
 2.1832  0.7119  1.6924 -0.2959
-0.1364  1.2902  0.5913 -1.4751
 0.1139  0.6686 -0.6436 -0.2340
 1.0668  1.1908  0.3803  0.1184
 0.0593 -1.2025 -1.0091  0.3148
-0.0956 -0.0198 -0.0195  1.4435
-0.8323 -0.1567 -0.0482 -0.3510
```

>>xbarplot(DATA)



\bar{X} контрольная карта с контрольными границами равными $\pm 2,5\sigma_{\bar{x}}$

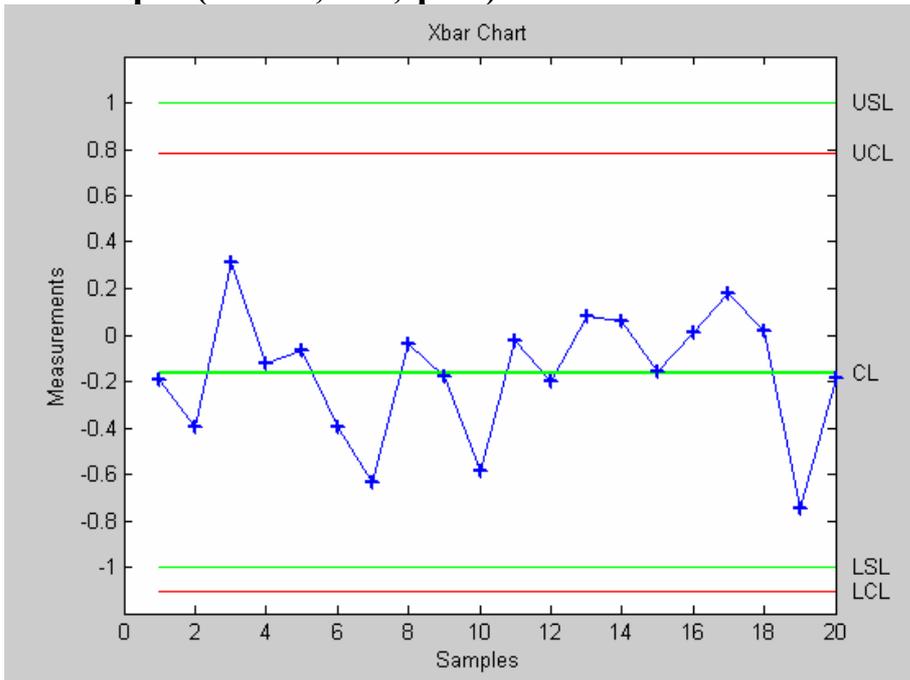
```
>> DATA=normrnd(0,1,20,4);
>> k = 2.5;
>> conf = 1-2*(1-normcdf(k))
conf =
    0.9876
>> xbarplot(DATA,conf)
```



\bar{X} контрольная карта с контрольными границами равными $\pm 2\sigma_{\bar{x}}$ и границами допусков технологического параметра -1; 1.

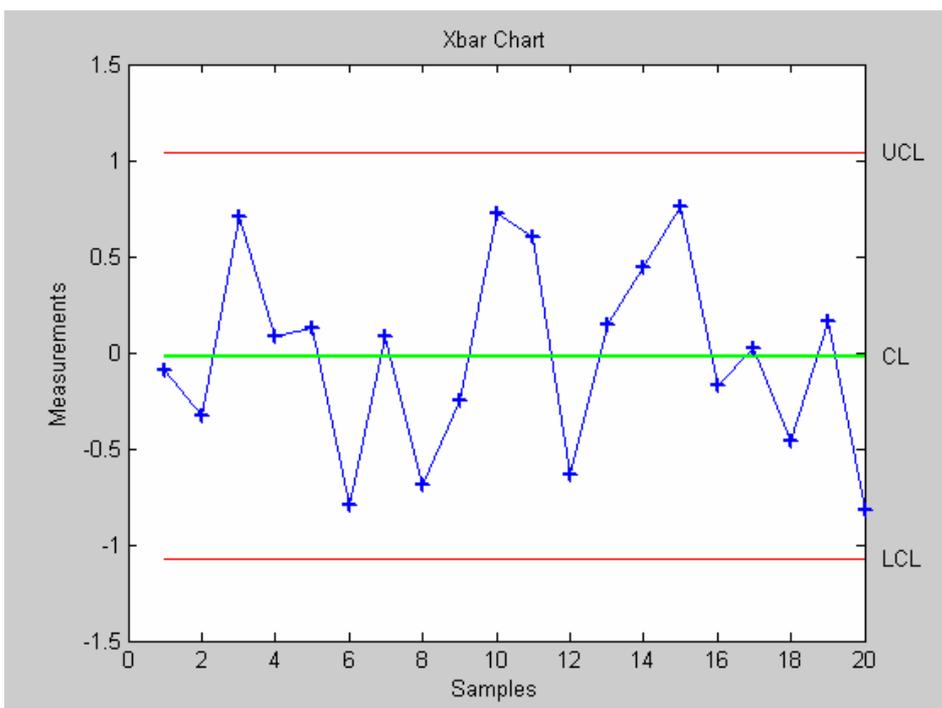
```
>> DATA=normrnd(0,1,20,4);
>> k=2;
>> conf = 1-2*(1-normcdf(k));
```

```
>> specs=[-1 1];
>> xbarplot(DATA,conf,specs)
```



\bar{X} контрольная карта с контрольными границами равными $\pm 2\sigma_{\bar{x}}$. Для расчета точечной оценки стандартного отклонения $\sigma_{\bar{x}}$ используется средний выборочный размах.

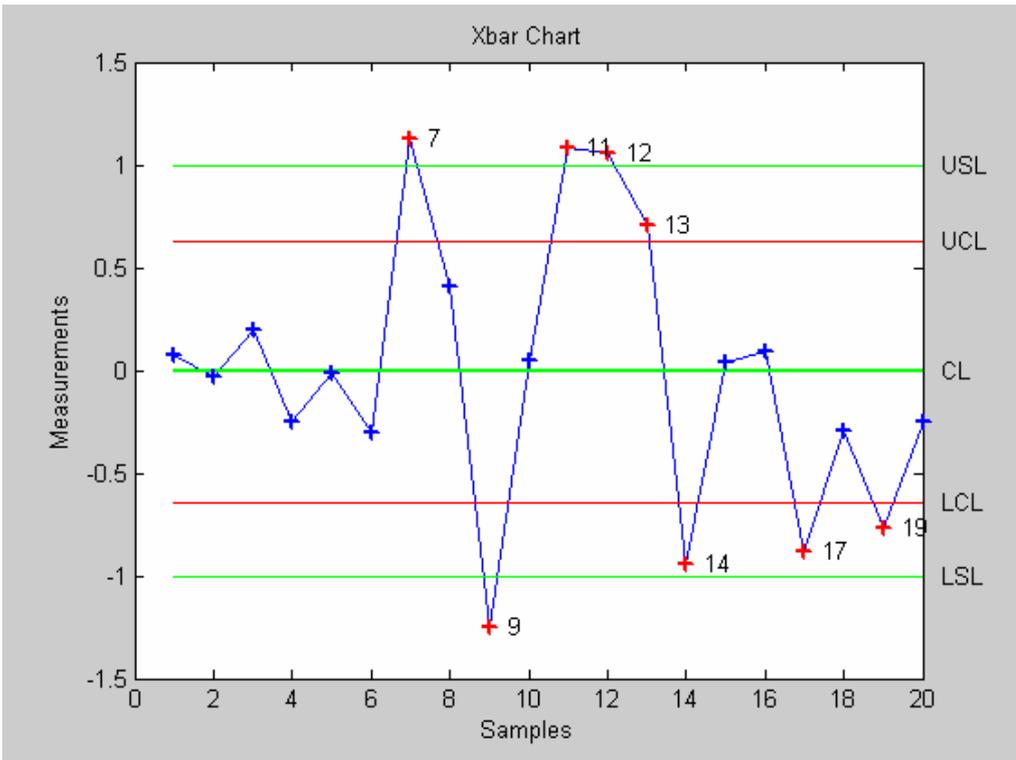
```
>> DATA=normrnd(0,1,20,4);
>> k=2;
>> conf=1-2*(1-normcdf(k));
>> specs=[-1 1];
>> sigmaest='range';
>> xbarplot(DATA,conf,specs,sigmaest)
```



Следует отметить, в этом варианте синтаксиса функции `xbarplot` не предусмотрено построение границ допусков на технологический параметр. Согласно тексту файла `xbarplot.m` границы допусков будут построены только при 3-х входных параметрах.

\bar{X} контрольная карта с контрольными границами равными $\pm 1,5\sigma_{\bar{x}}$ и границами допусков технологического параметра в интервале $-1; 1$. В качестве выходных параметров выступают вектор номеров выборок вышедших за контрольные границы `outlier`, и вектор указателей объектов графика `h`.

```
>> DATA=normrnd(0,1,20,4);
>> k=1.5;
>> conf=1-2*(1-normcdf(k));
>> specs=[-1 1];
>> [outlier,h] = xbarplot(DATA,conf,specs)
outlier =
    7
    9
   11
   12
   13
   14
   17
   19
h =
    3.0027
  102.0040
  103.0016
  104.0016
  105.0016
  106.0010
  121.0004
  122.0004
```



capable

Расчет индексов воспроизводимости

Синтаксис

```
p = capable(data, specs)
[p, Cp, Cpk] = capable(data, specs)
```

Описание

`p = capable(data, specs)` функция позволяет рассчитать вероятность p выхода значений выборки `data` за границы допусков `specs`. Выборка `data` задается как вектор. Границы допусков представляются в виде двухэлементного вектора: `specs(1)` – нижняя граница допуска, `specs(2)` – верхняя граница допуска.

Исходными предположениями при расчете p являются: 1. не противоречие выборки `data` нормальному закону с постоянными математическим ожиданием и дисперсией, 2. статистической независимостью результатов измерений в выборке.

`[p, Cp, Cpk] = capable(data, specs)` позволяет рассчитать вероятность p выхода значений выборки `data` за границы допусков `specs`, индексы воспроизводимости процесса C_p , C_{pk} .

Индекс C_p (индекс потенциальной пригодности) представляет собой отношение разности верхней USL и нижней LSL границ поля допуска к произведению $6 \cdot \sigma$, где σ – точечная оценка среднего квадратического отклонения выборки:

$$C_p = \frac{USL - LSL}{6\sigma}.$$

Для центрированного технологического процесса (выборочное среднее совпадает с номинальным значением параметра технологического процесса) значение $C_p = 1$ соответствует отношению числа дефектов к числу изделий равному $1/1000$. Согласно положениям метода 'шесть сигма'^a долю дефектов необходимо снижать до величин $10/1000000$ и менее. Величина доли дефектов $1/1000000$ соответствует значению $C_p = 1,6$. Таким образом, уменьшение величины коэффициента C_p соответствует улучшению качества технологического процесса по уровню дефектности при центрированном технологическом процессе.

Индекс C_{pk} (индекс смещенности технологического процесса) рассчитывается по формуле

$$C_{pk} = \min\left[\frac{USL - \mu}{3\sigma}, \frac{\mu - LSL}{3\sigma}\right],$$

где μ — среднее арифметическое выборки data.

Из приведенной выше формулы следует, что индекс воспроизводимости C_{pk} является отношением минимальной разности среднего арифметического выборки data и верхней или нижней границы поля допуска параметра к трем точечным оценкам среднего квадратического отклонения. Индекс $C_{pk} = 1$ для центрированного технологического процесса при $C_p = 1$.

Примеры использования функции расчета индексов воспроизводимости

Расчет вероятности брака.

```
>> data=normrnd(0,1,100,1);  
>> specs=[-2 2];  
>> p = capable(data,specs)  
p =  
    0.0413
```

Расчет вероятности брака и индексов воспроизводимости процесса C_p , C_{pk} .

```
>> data=normrnd(0,1,100,1);  
>> specs=[-2 2];  
>> [p,Cp,Cpk] = capable(data,specs)  
p =  
    0.0216  
Cp =  
    0.7672  
Cpk =  
    0.7452
```

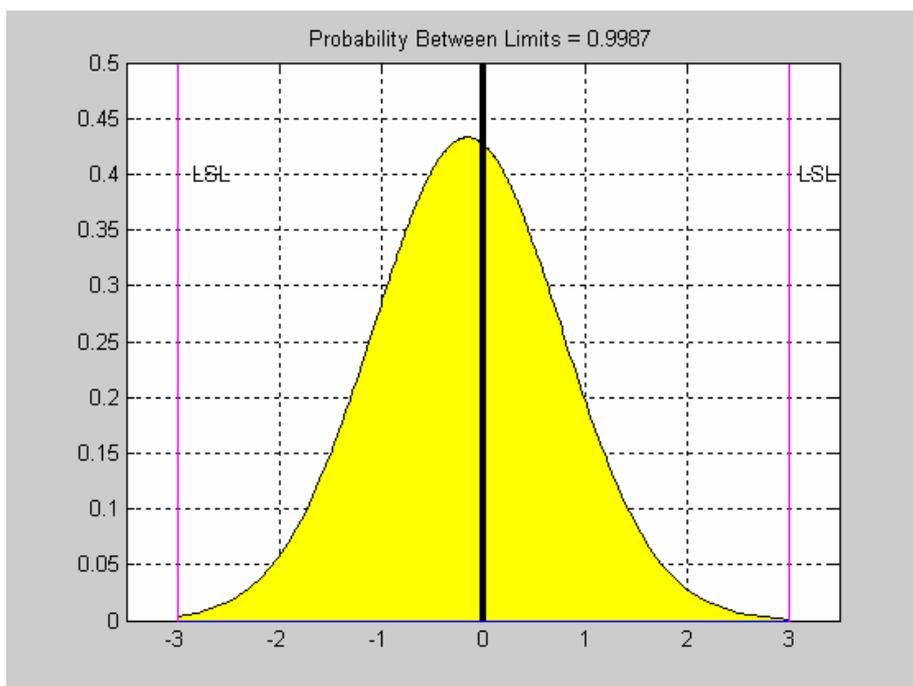
Пример центрированного процесса с границами рассеяния параметра ($\mu \pm 3\sigma$) совпадающими с границами допусков.

```
>> data=normrnd(0,1,100,1);  
>> specs=[-3 3];  
>> [p,Cp,Cpk] = capable(data,specs)  
p =  
    0.0013
```

```
Cp =  
1.0856  
Cpk =  
1.0316
```

Графическое представление централизованного технологического процесса с границами рассеяния параметра совпадающими с границами допусков.

```
>> data=normrnd(0,1,100,1);  
>> specs=[-3 3];  
>> [p,Cp,Cpk] = capable(data,specs)  
p =  
0.0013  
Cp =  
1.0856  
Cpk =  
1.0316  
>> capaplot(data,specs)  
>> H = line([0 0],[0 0.5])  
>> set(H,'LineWidth',3, 'Color','k')  
>> H1=line([specs(1) specs(1)],[0 0.5])  
>> set(H1,'Color','m')  
>> text(specs(1) + 0.1,0.4,'LSL');  
>> H2=line([specs(2) specs(2)],[0 0.5])  
>> set(H2,'Color','m')  
>> text(specs(2) + 0.1, 0.4,'USL');  
>> grid on
```

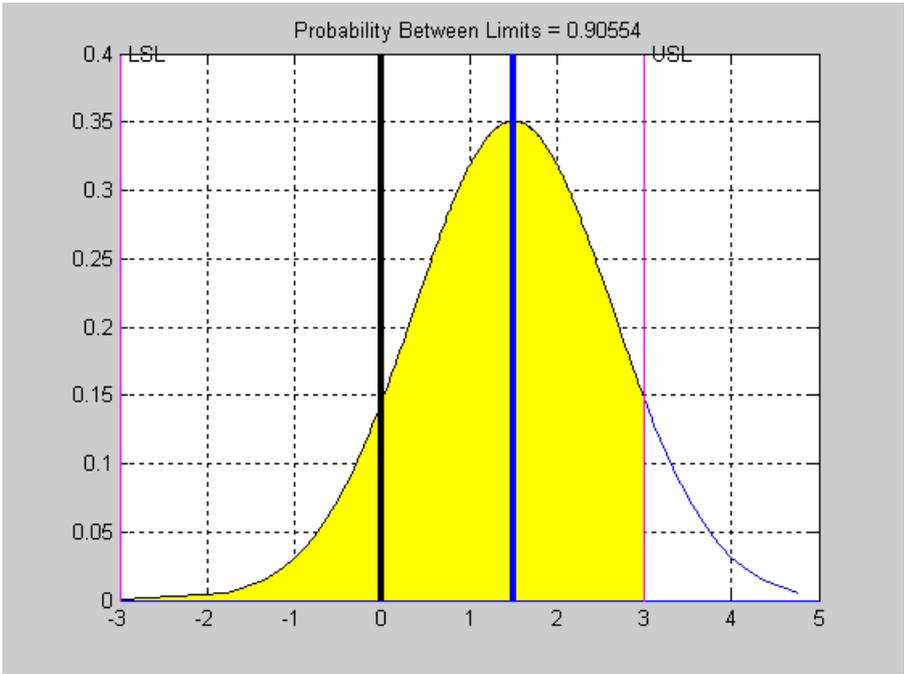


Пример смещенного вправо параметра технологического процесса с границами рассеяния параметра совпадающими с границами допусков. Номинальное значение параметра равно нулю.

```
>> data=normrnd(1.5,1,100,1);
>> specs=[-3 3];
>> [p,Cp,Cpk] = capable(data,specs)
p =
    0.0945
Cp =
    0.8797
Cpk =
    0.4380
```

Графическое представление смещенного вправо параметра технологического процесса и границами рассеяния параметра совпадающими с границами допусков.

```
>> data=normrnd(1.5,1,100,1);
>> specs=[-3 3];
>> capaplot(data,specs)
>> H = line([0 0],[0 0.4])
>> set(H,'LineWidth',3, 'Color','k')
>> H0 = line([mean(data) mean(data)],[0 0.4])
>> set(H0,'LineWidth',3, 'Color','b')
>> H1=line([specs(1) specs(1)],[0 0.4])
>> set(H1,'Color','m')
>> text(specs(1) + 0.1,0.4,'LSL');
>> H2=line([specs(2) specs(2)],[0 0.4])
>> set(H2,'Color','m')
>> text(specs(2) + 0.1, 0.4,'USL');
>> grid on
```



schart

Контрольная карта средних квадратических отклонений (s-контрольная карта)

Синтаксис

```
schart (DATA, conf)
schart (DATA, conf, specs)
schart (DATA, conf, specs)
[outliers, h] = schart (DATA, conf, specs)
```

Описание

`schart(data)` функция позволяет получить контрольную карту средних квадратических отклонений выборочных значений DATA. Размерность матрицы DATA равна $n \times m$, где n – количество выборок (строк DATA), m – объем выборки (число столбцов DATA). Последовательность выборок должна соответствовать порядку сбора исходных данных. На графике контрольной карты отображаются выборочные средние квадратические отклонения (s_i , $i=1..n$), центральная линия (\bar{s}), верхняя UCL и нижняя LCL контрольные границы.

Среднее арифметическое выборочных средних квадратических отклонений рассчитывается по формуле:

$$\bar{s} = \frac{1}{n} \sum_{i=1}^n s_i .$$

Верхняя и нижняя контрольные границы рассчитываются по формулам

$$UCL = \bar{s} + 3\sigma_s , \quad LCL = \bar{s} - 3\sigma_s ,$$

σ_s - среднее квадратическое отклонение \bar{s} .

Если технологический процесс является статистически управляемым, то при 1000 выборках количество точек вышедших за контрольные пределы не должно превышать 3-х в случайном порядке. Таким образом, при малом количестве выборок выход выборочного среднего квадратического отклонения за контрольные границы означает потерю процессом статистической управляемости.

`schart (DATA, conf)` функция предназначена для построения s-контрольной карты выборки DATA с заданной доверительной вероятностью `conf` для верхней и нижней контрольных границ. По

умолчанию значение `conf` принимается равной 0.9973. Величина `conf=0.9973` соответствует интервалу $\pm 3\sigma_s$:

```
>> norminv(1 - (1-.9973)/2)
ans =
    3.0000
```

Для расчета доверительной вероятности `conf` соответствующей $\pm k\sigma_s$ используется выражение $1-2*(1-normcdf(k))$. Например, для $k=2$ значение `conf` равно

```
>> k = 2;
>> conf = 1-2*(1-normcdf(k))
conf =
    0.9545
```

`schart(DATA, conf, specs)` функция предназначена для построения `s` контрольной карты с заданной доверительной вероятностью расположения контрольных границ `conf` и границами допусков `specs`. Границы допусков определяются как вектор с двумя элементами: `specs(1)` – нижняя граница допуска, `specs(2)` – верхняя граница допуска.

`[outliers, h] = schart(data, conf, specs)` выходными параметрами функции являются: `outlier` – вектор номеров выборок (строк матрицы `DATA`), средние квадратические отклонения которых вышли за контрольные границы, `h` – вектор указателей на объекты графика контрольной карты.

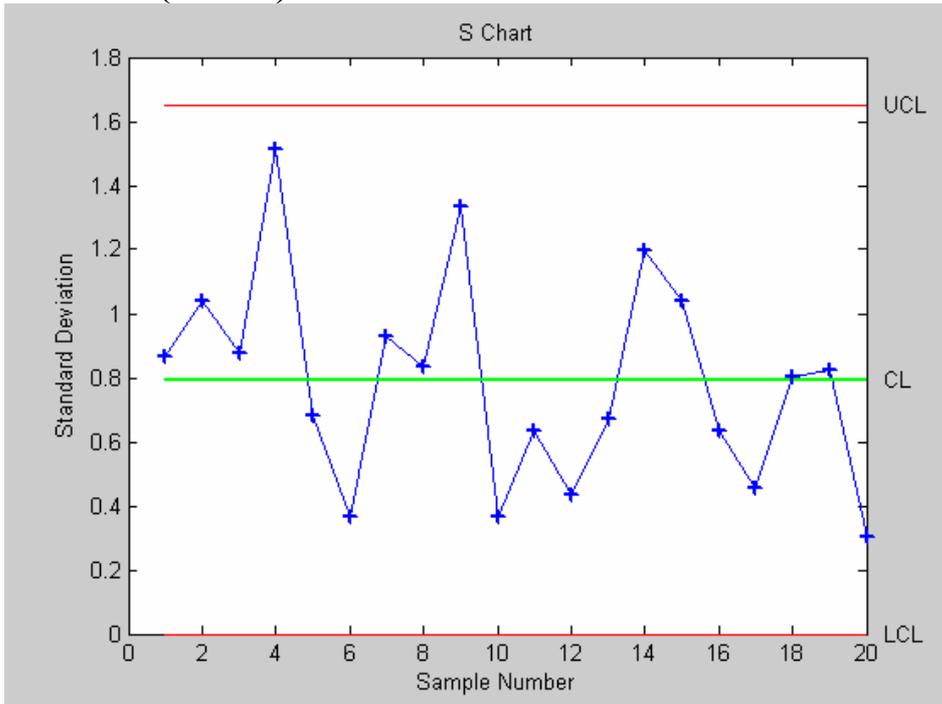
Примеры использования функции построения контрольной карты средних квадратических отклонений

`s` контрольная карта

```
>> DATA=normrnd(0,1,20,5)
DATA =
   -0.4326    0.2944   -1.6041    0.0000    0.6232
   -1.6656   -1.3362    0.2573   -0.3179    0.7990
    0.1253    0.7143   -1.0565    1.0950    0.9409
    0.2877    1.6236    1.4151   -1.8740   -0.9921
   -1.1465   -0.6918   -0.8051    0.4282    0.2120
    1.1909    0.8580    0.5287    0.8956    0.2379
    1.1892    1.2540    0.2193    0.7310   -1.0078
   -0.0376   -1.5937   -0.9219    0.5779   -0.7420
    0.3273   -1.4410   -2.1707    0.0403    1.0823
    0.1746    0.5711   -0.0592    0.6771   -0.1315
   -0.1867   -0.3999   -1.0106    0.5689    0.3899
```

0.7258	0.6900	0.6145	-0.2556	0.0880
-0.5883	0.8156	0.5077	-0.3775	-0.6355
2.1832	0.7119	1.6924	-0.2959	-0.5596
-0.1364	1.2902	0.5913	-1.4751	0.4437
0.1139	0.6686	-0.6436	-0.2340	-0.9499
1.0668	1.1908	0.3803	0.1184	0.7812
0.0593	-1.2025	-1.0091	0.3148	0.5690
-0.0956	-0.0198	-0.0195	1.4435	-0.8217
-0.8323	-0.1567	-0.0482	-0.3510	-0.2656

>> **schart(DATA)**



s контрольная карта для контрольных границ равных $\pm 2,5\sigma_s$

>> **DATA=normrnd(0,1,20,10);**

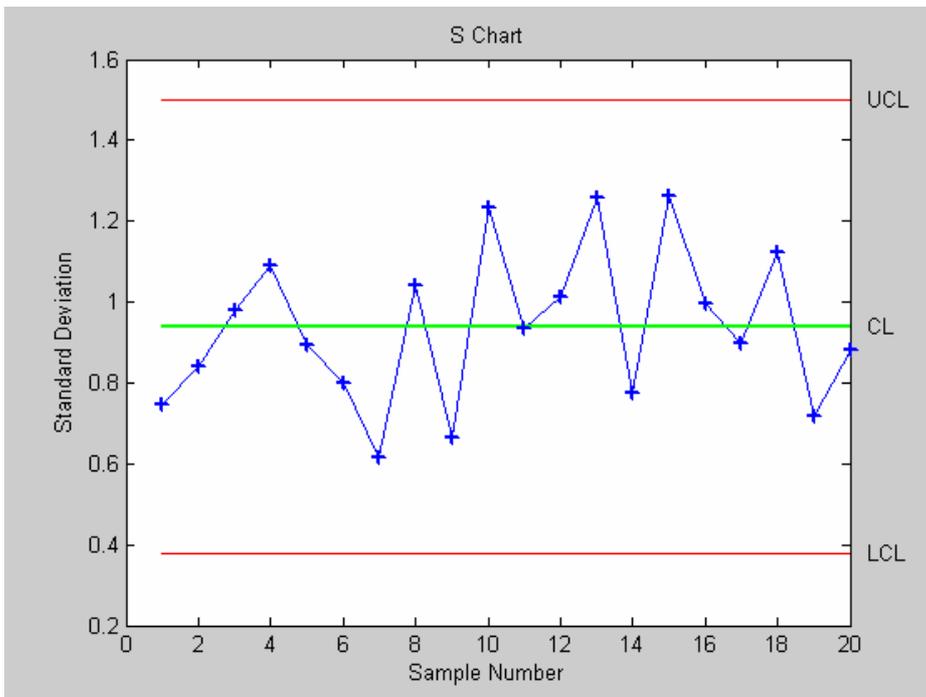
>> **k = 2.5;**

>> **conf = 1-2*(1-normcdf(k))**

conf =

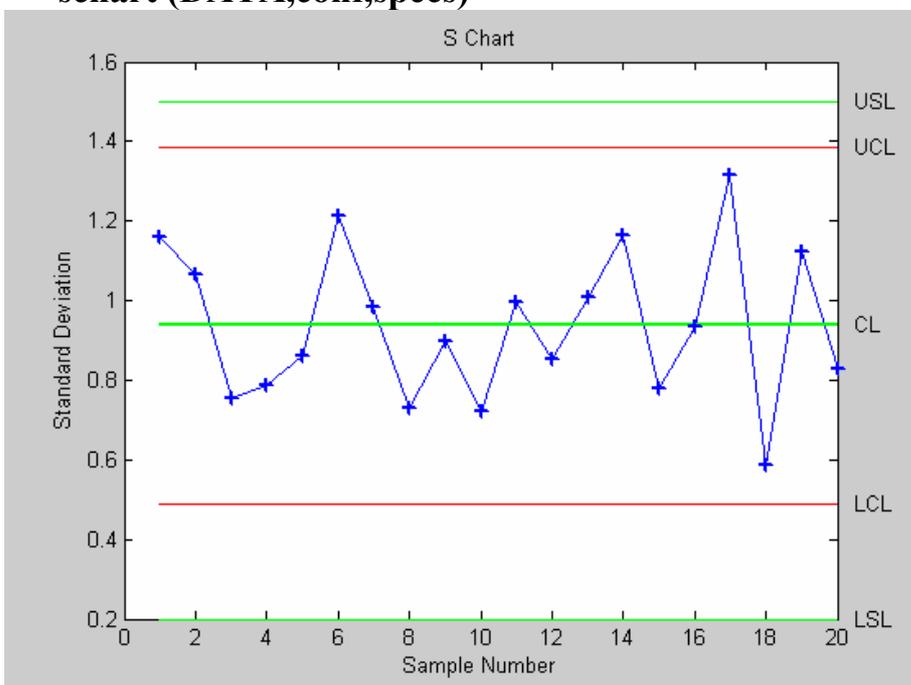
0.9876

>> **schart(DATA,conf)**



s контрольная карта для контрольных границ равных $\pm 2\sigma_s$ и границами допусков LSL=0,2; USL=1,5.

```
>> DATA=normrnd(0,1,20,10);
>> k=2;
>> conf=1-2*(1-normcdf(k));
>> specs=[0.2 1.5];
>> schart (DATA,conf,specs)
```



s контрольная карта для контрольных границ равных $\pm 1,5\sigma_s$ и границами допусков LSL=0,2; USL=1,5. В качестве выходных параметров выступают вектор номеров выборок вышедших за

контрольные границы \hat{n} outlier, и вектор указателей на объекты графика \hat{n} h.

```
>> DATA=normrnd(0,1,20,10);  
>> k=1.5;  
>> conf=1-2*(1-normcdf(k));  
>> specs=[0.2 1.5];  
>> [outlier,h] = schart (DATA,conf,specs)
```

outlier =

9

h =

3.0027

102.0040

103.0016

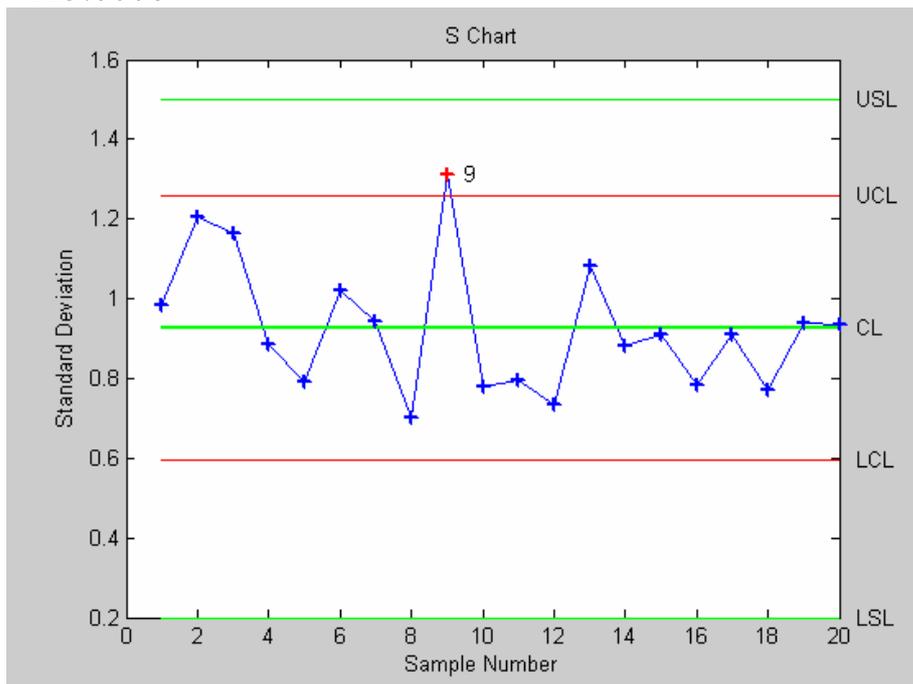
104.0016

105.0016

106.0010

114.0007

115.0006



capaplot

График воспроизводимости процесса

Синтаксис

```
p = capaplot(data, specs)
[p, h] = capaplot(data, specs)
```

Описание

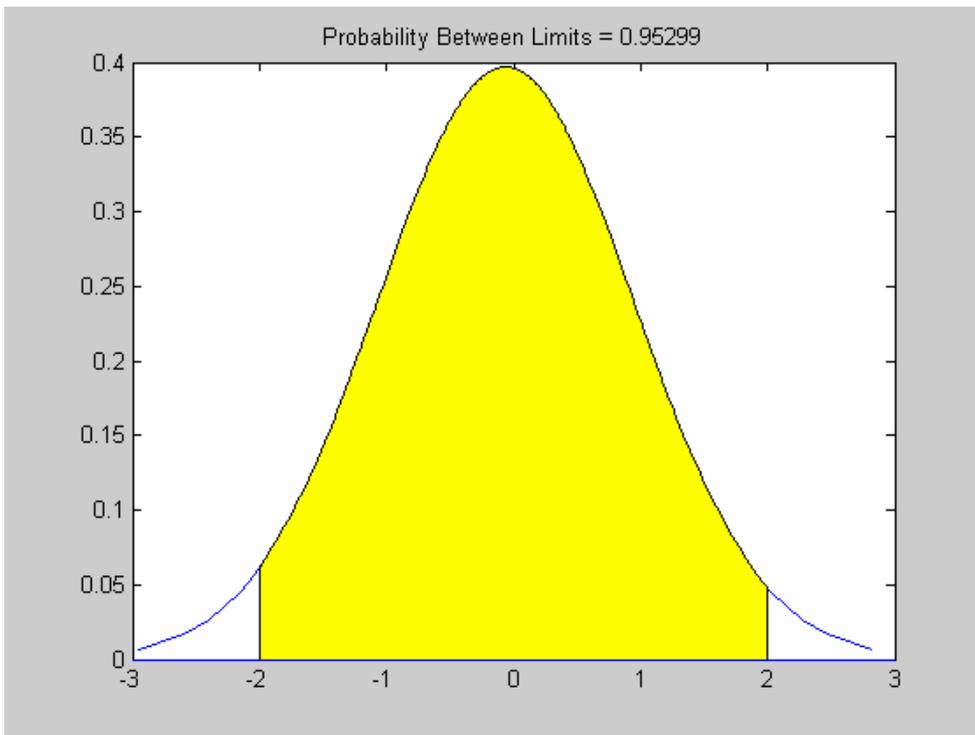
`p = capaplot(data, specs)` позволяет построить график функции плотности нормального закона по точечным оценкам математического ожидания и среднего квадратического отклонения выборки `data` с наложенными границами допусков параметра `specs`. Выборка входных значений `data` должна быть представлена как вектор. Предполагается, что выборка `data` не противоречит нормальному закону. Выходной параметр `p` является вероятностью попадания значения случайной величины в границы допусков `specs`. Границы допусков задаются в виде двухэлементного вектора: `specs(1)` – нижняя граница допуска, `specs(2)` – верхняя граница допуска. Вероятность попадания значений параметра технологического процесса на графике представляется закрашенной областью между границами допусков `specs`, осью абсцисс и кривой функции плотности распределения вероятности нормального закона.

`[p, h] = capaplot(data, specs)` функция кроме величины `p` возвращает массив указателей на элементы графика воспроизводимости `h`.

Примеры использования функции построения графика воспроизводимости процесса

Пример несмещенного технологического процесса с нулевым математическим ожиданием и единичной дисперсией. Нижняя и верхняя границы допусков равны `LSL=-2; USL=2`.

```
>> data = normrnd(0,1,50,1);
>> p = capaplot(data,[-2 2])
p =
    0.9530
```



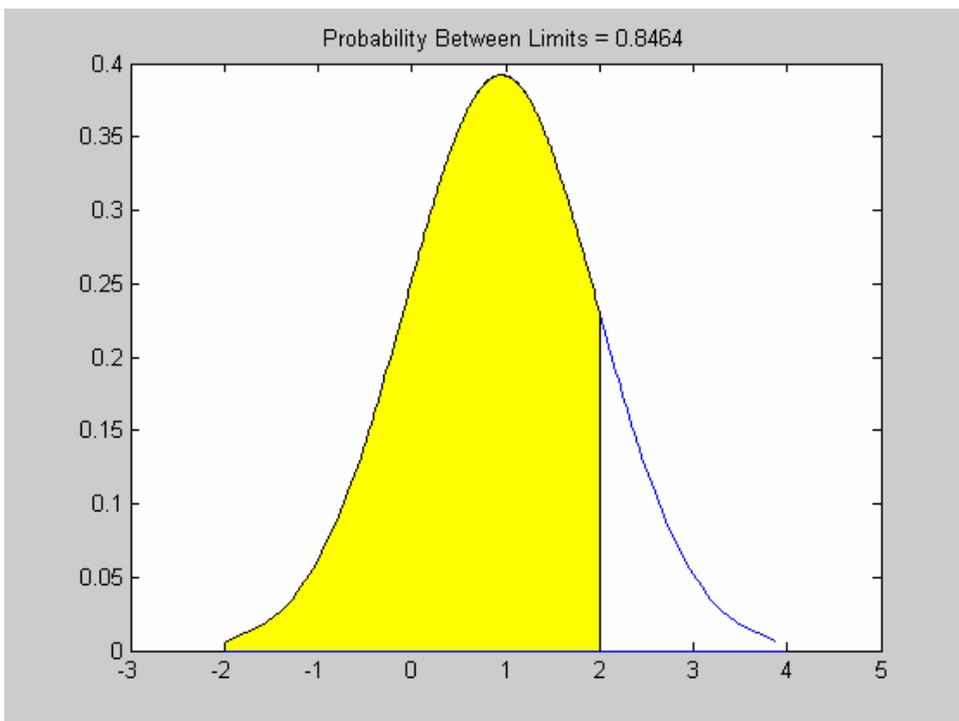
Пример смещенного технологического процесса с единичным математическим ожиданием и единичной дисперсией. Нижняя и верхняя границы допусков равны $LSL=-2$; $USL=2$.

```
>> data = normrnd(1,1,50,1);
```

```
>> p = capplot(data,[-2 2])
```

```
p =
```

```
0.8464
```



Использование функции `capaplot` для получения указателей на элементы графика воспроизводимости. Процесс несмещенный с нулевым математическим ожиданием и единичной дисперсией. Нижняя и верхняя границы допусков равны $LSL=-2$; $USL=2$.

```
>> data = normrnd(1,1,50,1);
>> [p h] = capaplot(data,[-2 2])
p =
    0.8163
h =
    129.0002
     3.0046
    102.0051
>> h(1)
ans =
    129.0002
>> get(h(1))
    Color = [0 0 1]
    EraseMode = normal
    LineStyle = -
    LineWidth = [0.5]
    Marker = none
    MarkerSize = [6]
    MarkerEdgeColor = auto
    MarkerFaceColor = none
    XData = [ (1 by 250) double array]
    YData = [ (1 by 250) double array]
    ZData = []
    BeingDeleted = off
    ButtonDownFcn =
    Children = []
    Clipping = on
    CreateFcn =
    DeleteFcn =
    BusyAction = queue
    HandleVisibility = on
    HitTest = on
    Interruptible = on
    Parent = [101.001]
    Selected = off
    SelectionHighlight = on
    Tag =
    Type = line
    UIContextMenu = []
    UserData = []
```

Visible = on

ewmplot

Контрольная карта экспоненциально взвешенного скользящего среднего (EWMA контрольная карта)

Синтаксис

```
ewmplot(data)
ewmplot(data, lambda)
ewmplot(data, lambda, alpha)
ewmplot(data, lambda, alpha, specs)
h = ewmplot(...)
```

Описание

`ewmplot(data)` функция предназначена для построения контрольной карты экспоненциально взвешенного скользящего среднего для выборки `data`. Размерность матрицы `DATA` равна $n \times m$, где n – количество выборок (строк `DATA`), m – объем выборки (число столбцов `DATA`). Последовательность выборок (строк матрицы `data`) должна соответствовать порядку сбора исходных данных. На графике контрольной карты отображаются выборочное скользящее среднее \bar{X}_i , $i=1..n$, центральная линия, соответствующая общему среднему $\bar{\bar{X}}$, верхняя `UCL` и нижняя `LCL` контрольные границы.

Верхняя и нижняя контрольные границы определяются как

$$UCL = \bar{\bar{X}} + 3\sigma_{\bar{X}}, \quad LCL = \bar{\bar{X}} - 3\sigma_{\bar{X}},$$

$\sigma_{\bar{X}}$ - среднее квадратическое отклонение $\bar{\bar{X}}$.

`ewmplot(data, lambda)` функция служит для построения контрольной карты экспоненциально взвешенного скользящего среднего для выборки `data` и переменной `lambda`, отвечающей за степень влияния предыдущих значений на текущее скользящее среднее. Большее значение `lambda` соответствует большему весу предыдущих наблюдений. Величина `lambda` должна находиться в интервале `[0 1]`. По умолчанию `lambda=0.4`.

`ewmplot(data, lambda, alpha)` функция позволяет построить контрольную карту экспоненциально взвешенного скользящего среднего для выборки `data` с заданным весом предыдущих наблюдений `lambda` и уровнем значимости `alpha` для верхней и нижней контрольных границ. По умолчанию `alpha=0,0027`, это соответствует $\pm 3\sigma_{\bar{X}}$:

```
>> norminv(1-0.0027/2)
```

```
ans =  
    3.0000
```

Для расчета уровня значимости α соответствующего $\pm k$ средних квадратических отклонений скользящего среднего используется выражение $2*(1-\text{normcdf}(k))$. Например, для $k=2$ значение conf равно

```
>> k = 2;  
>> alpha = 2*(1-normcdf(k))  
alpha =  
    0.0455
```

`ewmplot(data, lambda, alpha, specs)` функция служит для построения контрольной карты экспоненциально взвешенного скользящего среднего для выборки `data` с заданным весом предыдущих наблюдений `lambda`, уровнем значимости `alpha` и границ допуска параметра `specs`. Границы допусков определяются как вектор с двумя элементами: `specs(1)` — нижняя граница допуска, `specs(2)` — верхняя граница допуска.

`h = ewmplot(...)` в этом варианте синтаксиса допустимы любые перечисленные выше входные параметры, выходным параметром `h` является вектор указателей на объекты графика контрольной карты.

Примеры использования функции построения контрольной карты экспоненциально взвешенного скользящего среднего

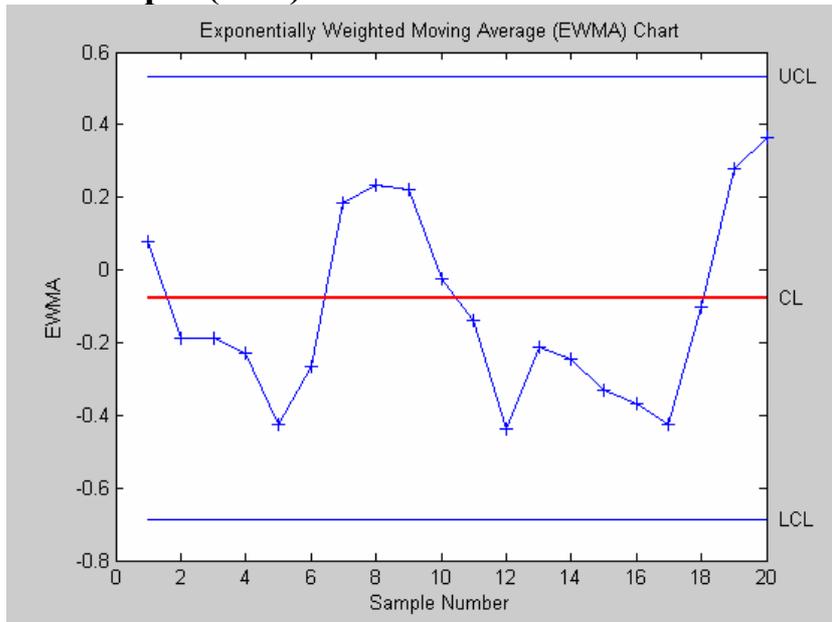
EWMA контрольная карта для технологического процесса с постоянным нулевым математическим ожиданием и единичной дисперсией.

```
>> data = normrnd(0,1,20,5)
```

```
data =  
    0.4634    1.4766    0.6970   -0.5102   -0.5608  
   -0.9241   -0.8138   -1.3664   -0.0067    0.1793  
   -0.6497    0.6450    0.3630   -0.5255   -0.7715  
    0.6229   -1.3099   -0.5670    0.7177   -0.9434  
   -1.3351   -0.8674   -1.0442    1.0884   -1.4076  
    1.0477   -0.4742    0.6971    0.5006   -1.9061  
    0.8633    0.2224    0.4840    2.7718   -0.0653  
   -0.6424    1.8713   -0.1938   -0.1603    0.6721  
    0.6600    0.1100   -0.3781    0.4295    0.2061  
    1.2941   -0.4113   -0.8864   -1.9668   -0.0081  
    0.3146    0.5112   -1.8402   -0.5460    0.0200  
    0.8596   -1.1991   -1.6282   -1.8884   -0.5584
```

0.1287	-0.0964	-1.1738	-0.1080	1.8861
0.0166	0.4458	-0.4154	-1.3161	-0.2200
-0.0728	-0.2958	0.1751	-0.6726	-1.4144
-0.9943	-0.1680	0.2294	-0.9024	-0.3028
-0.7474	0.1795	-1.2409	-0.1548	-0.5696
-0.0308	0.4211	0.7000	0.9472	-0.1215
0.9884	1.6777	0.4269	1.5504	-0.3902
-0.5990	1.9969	1.4548	0.4290	-0.8443

>>ewmaplot(data)

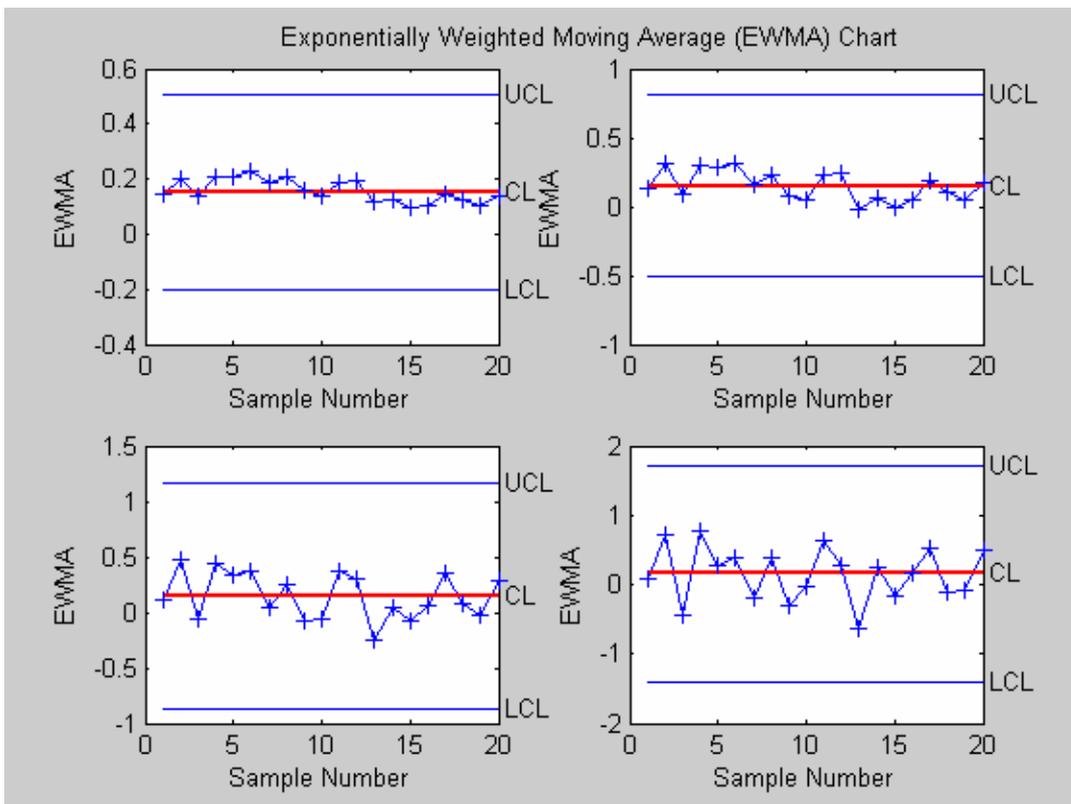


Вид EWMA контрольной карты для технологического процесса с постоянным нулевым математическим ожиданием, единичной дисперсией и коэффициентом влияния $\lambda=[0.1 \ 0.3 \ 0.6 \ 1]$.

```

>>data=normrnd(0,1,20,5);
>>lambda=0.1;
>> subplot(2,2,1)
>>ewmaplot(data,lambda)
>>lambda=0.3;
>> subplot(2,2,2)
>>ewmaplot(data,lambda)
>>lambda=0.6;
>> subplot(2,2,3)
>>ewmaplot(data,lambda)
>>lambda=1;
>> subplot(2,2,4)
>>ewmaplot(data,lambda)

```



EWMA контрольная карта для контрольных границ на уровне $\pm 2,5\sigma_{\bar{x}}$.

```
>>data=normrnd(0,1,20,5);
```

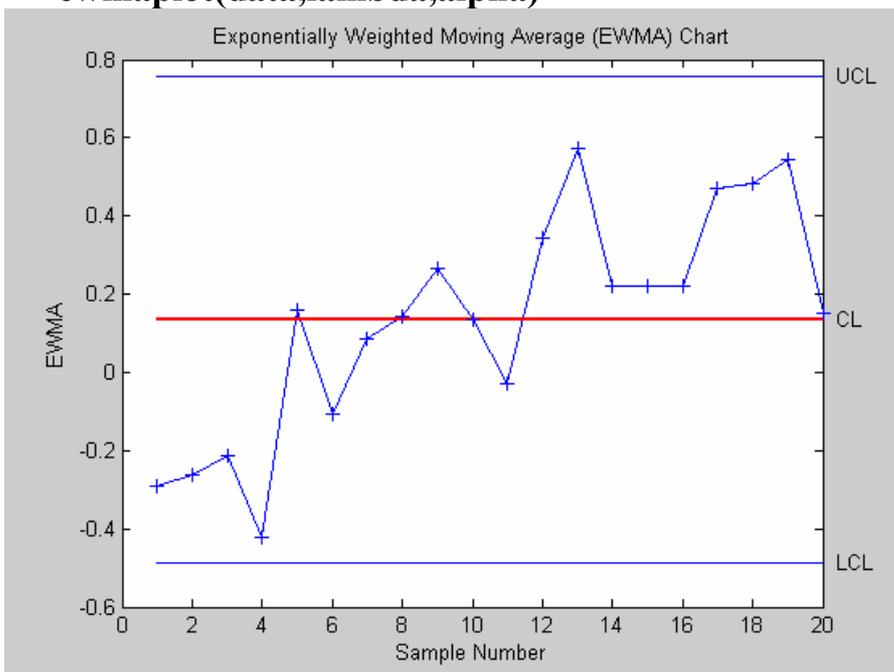
```
>>lambda=0.5;
```

```
>> k = 2.5;
```

```
>> alpha =2*(1-normcdf(k))
```

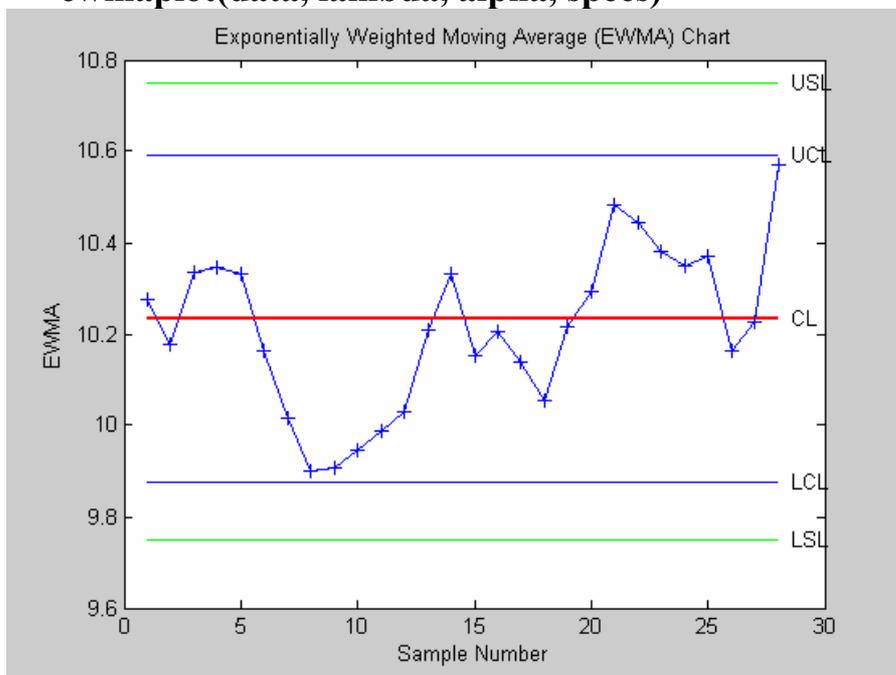
```
alpha =  
    0.0124
```

```
>> ewmplot(data,lambda,alpha)
```



EWMA контрольная карта для технологического процесса с линейно нарастающим математическим ожиданием, равным $10+0.02*t(:,ones(4,1))$ при $t = (1:28)'$, постоянной дисперсией, равной $0,5^2$. Выборка исходных данных `data` является матрицей с размерностью 28×4 , где количество выборок равно 28, число измерений в выборке $n = 4$. Контрольная карта экспоненциально взвешенного скользящего среднего строится для постоянной сглаживания $\lambda = 0.4$, контрольных границ соответствующих уровню значимости $\alpha = 0.01$. Границы допусков параметра равны $LSL=9,75$; $USL=10,75$.

```
>> t = (1:28)';
>> data = normrnd(10+0.02*t(:,ones(4,1)),0.5);
>> lambda = 0.4;
>> alpha = 0.01;
>> specs=[9.75 10.75];
>> ewmaplot(data, lambda, alpha, specs)
```



Получение доступа к свойствам объектов EWMA контрольной карты.

```
>> data=normrnd(0,1,20,5);
>> lambda=0.5;
>> alpha = 0.0124;
>> h=ewmaplot(data,lambda,alpha)
h =
113.0018
  3.0089
102.0116
103.0048
104.0048
105.0048
```

```
>>get(h(1))
Color = [0 0 1]
EraseMode = normal
LineStyle = -
LineWidth = [0.5]
Marker = none
MarkerSize = [6]
MarkerEdgeColor = auto
MarkerFaceColor = none
XData = [ (1 by 20) double array]
YData = [ (1 by 20) double array]
ZData = []
BeingDeleted = off
ButtonDownFcn =
Children = []
Clipping = on
CreateFcn =
DeleteFcn =
BusyAction = queue
HandleVisibility = on
HitTest = on
Interruptible = on
Parent = [101.002]
Selected = off
SelectionHighlight = on
Tag =
Type = line
UIContextMenu = []
UserData = []
Visible = on
```

histfit

Гистограмма с наложенным графиком функции плотности распределения вероятностей нормального закона

Синтаксис

```
histfit(data)
histfit(data,nbins)
h = histfit(data,nbins)
```

Описание

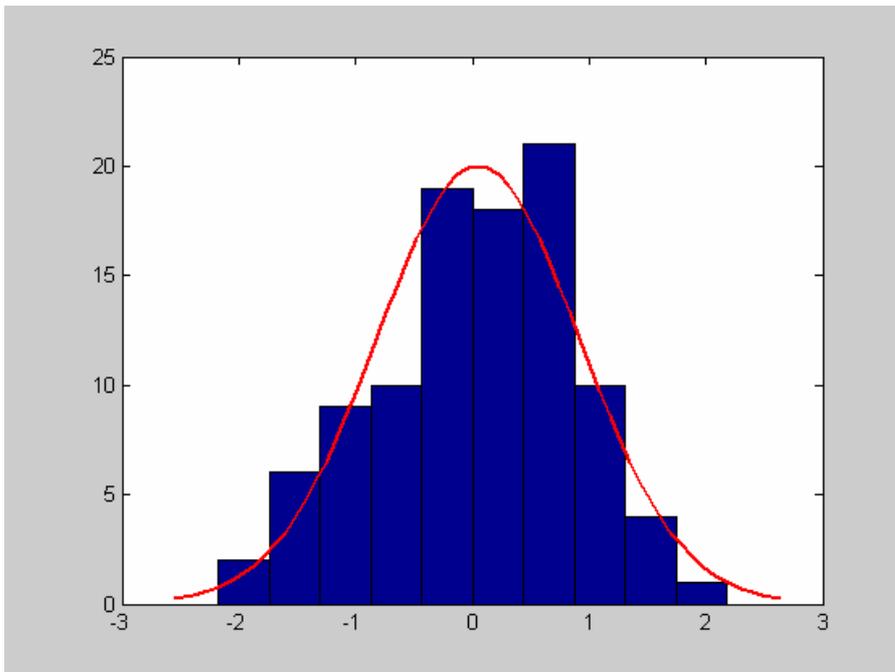
`histfit(data,nbins)` функция позволяет построить гистограмму с наложенной функцией плотности вероятности нормального закона по выборке `data` с числом интервалов `nbins`. Выборка `data` должна быть задана как вектор негруппированных значений. Группировка выборки `data` выполняется по интервалам равной длины. Если параметр `nbins` не задан, он принимается равным ближайшему целому из корня квадратного числа элементов в выборке `data`.

`h = histfit(data,nbins)` в качестве выходного параметра выступает вектор указателей на элементы графика. Первый элемент вектора `h(1)` является указателем на гистограмму, `h(2)` — указатель на кривую плотности вероятности нормального закона.

Примеры использования функции гистограммы с графиком функции плотности нормального закона

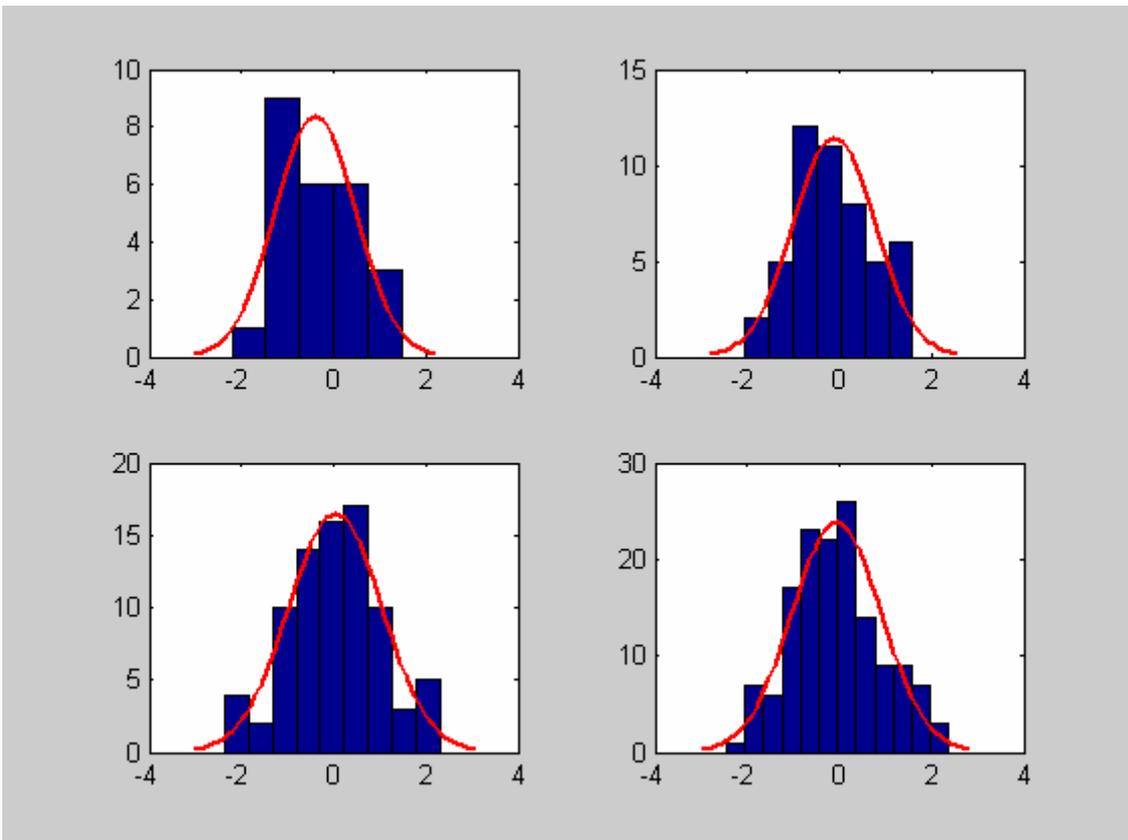
Гистограмма с наложенным графиком функции плотности распределения вероятностей нормального закона. Число интервалов группирования определяется по умолчанию, как ближайшее целое к корню квадратному из объема выборки.

```
>>data=normrnd(0,1,100,1);
>>histfit(data)
```



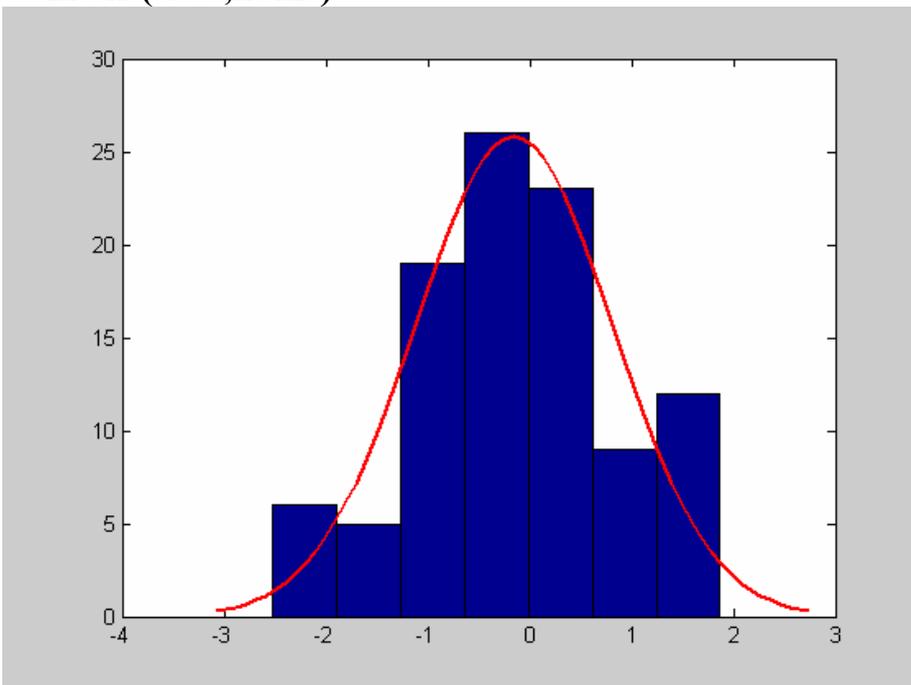
Вид получаемых графиков в зависимости от объема выборки при одинаковых параметрах генерируемой выборки.

```
>>data=normrnd(0,1,25,1);  
>>subplot(2,2,1)  
>>histfit(data)  
>>data=normrnd(0,1,49,1);  
>>subplot(2,2,2)  
>>histfit(data)  
>>data=normrnd(0,1,81,1);  
>>subplot(2,2,3)  
>>histfit(data)  
>>data=normrnd(0,1,144,1);  
>>subplot(2,2,4)  
>>histfit(data)
```



Гистограмма с наложенным графиком функции плотности распределения вероятностей нормального закона с заданным числом интервалов группирования.

```
>>data=normrnd(0,1,100,1);
>>nbins=7;
>>histfit(data,nbins)
```



Использование указателей на элементы графика и получение списка свойств гистограммы.

```
>>data=normrnd(0,1,100,1);
>>nbins=7;
>>h=histfit(data,nbins)
h =
    101.0029
    102.0010
>>get(h(1))
    AlphaDataMapping = scaled
    CData = [ (4 by 7) double array]
    CDataMapping = scaled
    FaceVertexAlphaData = []
    FaceVertexCData = [ (36 by 1) double array]
    EdgeAlpha = [1]
    EdgeColor = [0 0 0]
    EraseMode = normal
    FaceAlpha = [1]
    FaceColor = flat
    Faces = [ (7 by 4) double array]
    LineStyle = -
    LineWidth = [0.5]
    Marker = none
    MarkerEdgeColor = auto
    MarkerFaceColor = none
    MarkerSize = [6]
    Vertices = [ (36 by 2) double array]
    XData = [ (4 by 7) double array]
    YData = [ (4 by 7) double array]
    ZData = []
    FaceLighting = flat
    EdgeLighting = none
    BackFaceLighting = reverselit
    AmbientStrength = [0.3]
    DiffuseStrength = [0.6]
    SpecularStrength = [0.9]
    SpecularExponent = [10]
    SpecularColorReflectance = [1]
    VertexNormals = [ (36 by 3) double array]
    NormalMode = auto
    BeingDeleted = off
    ButtonDownFcn =
    Children = []
    Clipping = on
    CreateFcn =
    DeleteFcn =
    BusyAction = queue
```

HandleVisibility = on
HitTest = on
Interruptible = on
Parent = [100.001]
Selected = off
SelectionHighlight = on
Tag =
Type = patch
UIContextMenu = []
UserData = []
Visible = on

normspec

График плотности распределения вероятностей нормального закона с границами допусков параметра

Синтаксис

```
p = normspec(specs, mu, sigma)
[p, h] = normspec(specs, mu, sigma)
```

Описание

`p = normspec(specs, mu, sigma)` функция предназначена для построения графика плотности распределения вероятностей нормального закона с заданными параметрами: математическим ожиданием `mu`, средним квадратическим отклонением `sigma`, и границами допусков параметра `specs`. Границы допусков определяются как вектор с двумя элементами: `specs(1)` – нижняя граница допуска, `specs(2)` – верхняя граница допуска. Выходной параметр `p` – вероятность попадания значения параметра в границы допусков `specs`. Если допуск на параметр является односторонним, то отсутствующая граница допуска задается как бесконечное значение. При одностороннем ограничении слева правая граница задается как `Inf`, т.е. `specs(1)=A`, `specs(2)=Inf`, или `specs=[A Inf]`, где `A` – нижняя граница допуска. При одностороннем ограничении справа левая граница задается как `-Inf`, т.е. `specs(1)=-Inf`, `specs(2)=B`, или `specs=[-Inf B]`, где `B` – верхняя граница допуска.

`[p, h] = normspec(specs, mu, sigma)` в этом варианте синтаксиса кроме вероятности попадания значения параметра в границы допусков `p`, возвращается вектор указателей элементов графика `h`.

Примеры использования функции графика плотности распределения вероятностей нормального закона с границами допусков параметра

График плотности распределения вероятностей несмещенного нормального технологического процесса с двусторонними границами допусков на параметр и единичной дисперсией.

```
>> mu = 0;
>> sigma = 1;
>> specs = [-2 2];
>> p = normspec(specs, mu, sigma)
p =
    0.9545
```

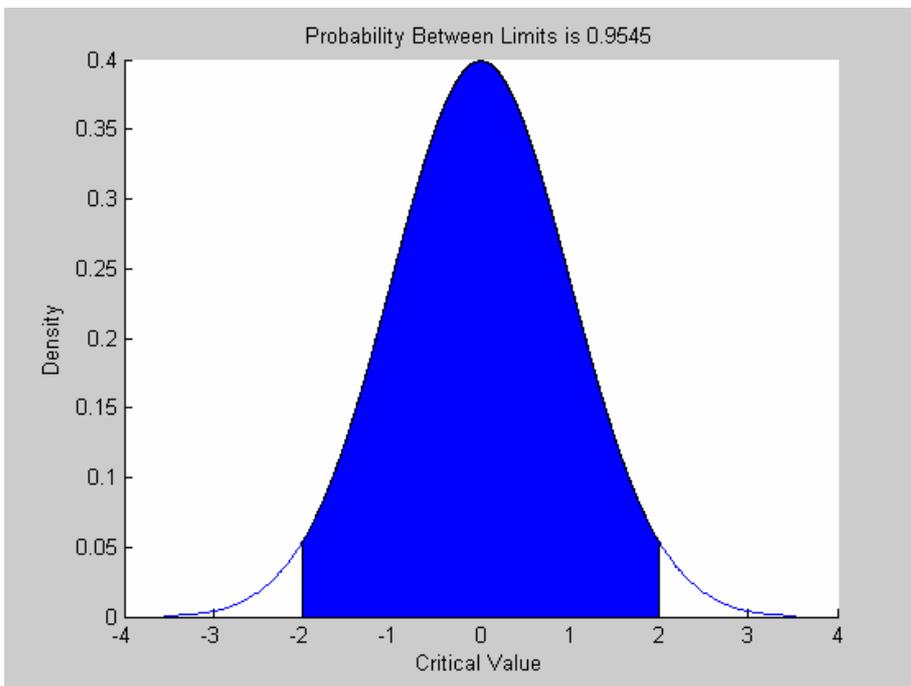
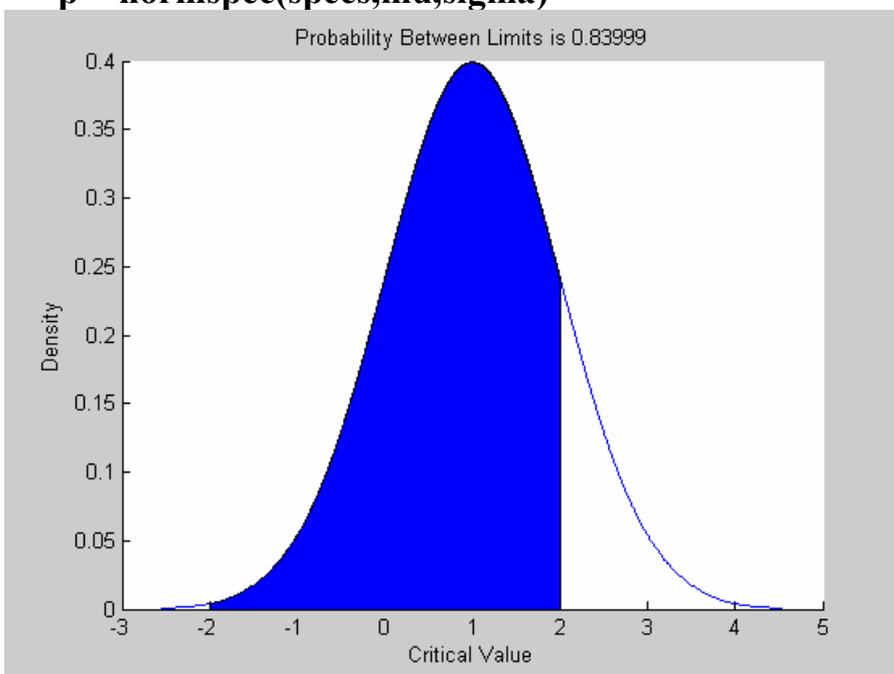


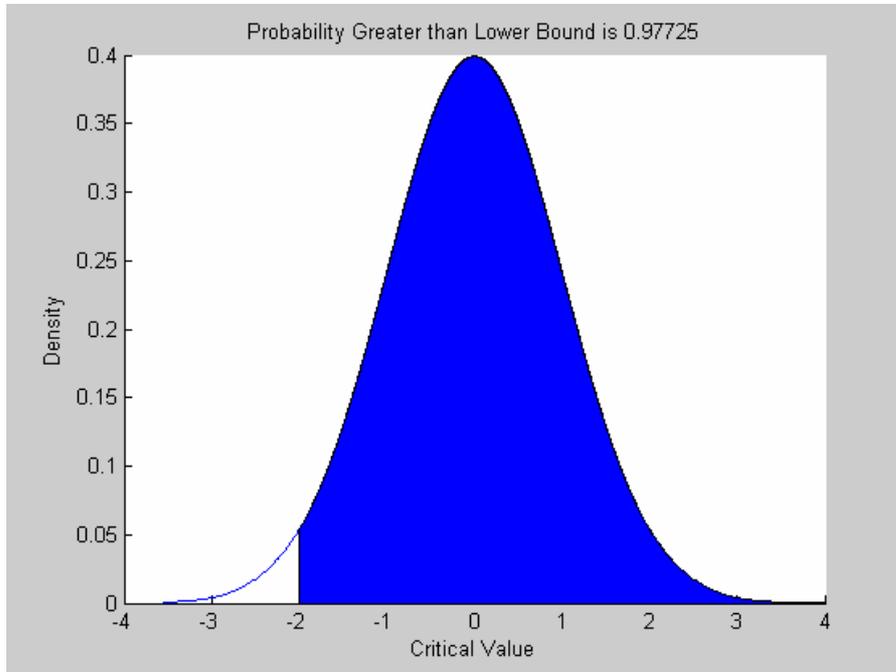
График плотности распределения вероятностей смещенного нормального технологического процесса с двусторонними границами допусков на параметр и единичной дисперсией.

```
>> mu = 1;
>> sigma = 1;
>> specs = [-2 2];
>> p = normspec(specs,mu,sigma)
```

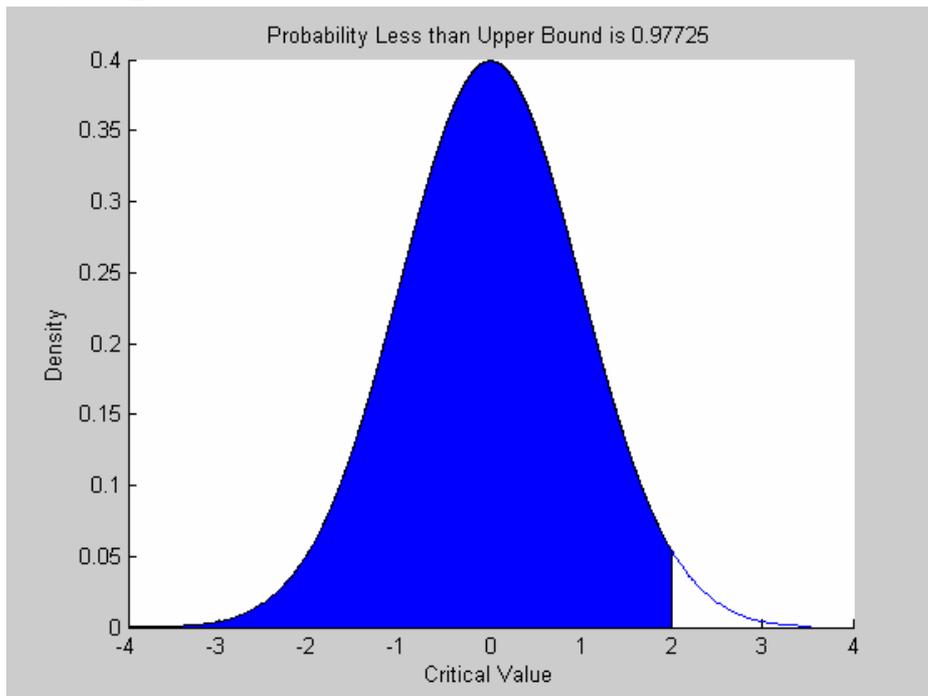


Графики плотности распределения вероятностей несмещенного нормального технологического процесса с односторонними границами допусков на параметр и единичной дисперсией.

```
>> mu = 0;  
>> sigma = 1;  
>> specs = [-2 Inf];  
>> subplot(2,1,1)  
>> p = normspec(specs,mu,sigma)  
p =  
0.9772
```



```
>> specs = [-Inf 2];  
>> p = normspec(specs,mu,sigma)  
p =  
0.9772
```



Использование указателей на элементы графика и получение списка свойств.

```
>> mu = 0;
>> sigma = 1;
>> specs = [-2 2];
>> [p h] = normspec(specs,mu,sigma)
p =
    0.9545
h =
    206.0006
    207.0005
    208.0005
>>get(h(1))
    Color = [0 0 1]
    EraseMode = normal
    LineStyle = -
    LineWidth = [0.5]
    Marker = none
    MarkerSize = [6]
    MarkerEdgeColor = auto
    MarkerFaceColor = none
    XData = [ (1 by 2500) double array]
    YData = [ (1 by 2500) double array]
    ZData = []
    BeingDeleted = off
    ButtonDownFcn =
    Children = []
    Clipping = on
    CreateFcn =
    DeleteFcn =
    BusyAction = queue
    HandleVisibility = on
    HitTest = on
    Interruptible = on
    Parent = [205]
    Selected = off
    SelectionHighlight = on
    Tag =
    Type = line
    UIContextMenu = []
    UserData = []
    Visible = on
```

Список литературы

Литература по Statistics Toolbox

1. Потемкин В.Г. Система MATLAB 5 для студентов. М.: Диалог-МИФИ, 1998. 314 с.
2. Потемкин В.Г. Система MATLAB. Справочное пособие. М.: Диалог-МИФИ, 1998. 350 с.
3. Дьяконов В. Круглов В. Математические пакеты расширения MATLAB. Специальный справочник. СПб.: Питер, 2001. 400 с.

Литература по математической статистике

1. Дюк В.А. Обработка данных на ПК в примерах. СПб: Питер, 1997.
2. Гайдышев И. Анализ и обработка данных. Специальный справочник. - СПб.: Питер, 2001.
3. Айвазян С.А., Бухштабер В.М., Юнюков И.С., Мешалкин Л.Д. Прикладная статистика: Классификация и снижение размерности. М.: Финансы и статистика, 1989.
4. Кречетов Н. Продукты для интеллектуального анализа данных. Рынок программных средств, №14, 1997, с.32-39.
5. Киселев М., Соломатин Е.. Средства добычи знаний в бизнесе и финансах. Открытые системы, № 4, 1997, с.41-44.
6. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Исследование зависимостей. М.: Финансы и статистика, 1985.
7. Болч, Б.У., Хуань, К.Д. Многомерные статистические методы для экономики. М.: Статистика, 1979.
8. Вапник В.Н. Восстановление зависимостей по эмпирическим данным. М.: Наука, 1979.
9. Дюрэн Б., Оделл П. Кластерный анализ. М.: Статистика, 1977.
10. Ермаков С.М., Жиглявский А.А. Математическая теория оптимального эксперимента. М.: Наука, 1982.
11. Кокс Д.Р. Оукс Д. «Анализ данных типа времени жизни», М.: Финансы и статистика, 1988.
12. Лоули Д., Максвелл А. Факторный анализ как статистический метод. М.: Мир, 1967.

13. Мостеллер, Ф., Тьюки, Дж. Анализ данных и регрессия. В 2-х вып. М.: Финансы и статистика, 1982.
14. Розин Б.Б. Теория распознавания образов в экономических исследованиях. М.: Статистика, 1973.
15. Справочник по прикладной статистике. М.: Финансы и статистика, 1990.
16. Хардле В. Прикладная непараметрическая регрессия. М.: Мир, 1993.
17. Харман Г. Современный факторный анализ. М.: Статистика, 1972.
18. Хьюбер П. Робастность в статистике. М.: Мир, 1984.
19. Шеффе Г. Дисперсионный анализ. М.: Наука, 1980.
20. Эфрон, Б. Нетрадиционные методы многомерного статистического анализа: Сб. статей. М.: Финансы и статистика, 1988.
21. Магнус Я.Р., Катышев П.К., Пересецкий А.А. Эконометрика. Начальный курс. Учебное пособие. Изд. 3-е, испр. и доп. - М.: Дело, 2000. - 248 с. ISBN 5-7749-0055-X
22. Катышев П. К., Пересецкий А. А. Сборник задач к начальному курсу эконометрики 72 стр. Москва: Дело, 1999 ISBN: 5-7749-0137-8
23. Доугерти К. Введение в эконометрику. - М.:ИНФРА-М,1997.- 402 с.:ил.- Библиогр.:с.384-386. ISBN 5-86225-458-7; 0-19-50346-4.
24. Айвазян С. А., Мхитарян В. С. Прикладная статистика и основы эконометрики. Юнити, 1998, 1022 стр. ISBN 5-238-00013-8
25. Многомерный статистический анализ в экономике. Под ред. В. Н. Тамашевича. -- М.: Юнити-Дана, 1999. -- 598 с. ISBN 5-238-00099-5.
26. Вероятность и математическая статистика: Энциклопедия. / Гл.ред.: Ю.В.Прохоров - М.: Большая Российская энциклопедия, 1999, 910 стр. ISBN 5-85270-265-X.
27. Экономическая статистика. Эконометрика : Методические материалы по экономическим дисциплинам для преподавателей средних школ и вузов. Под ред. Л. С. Гребнева [Образцова О. И., Назарова О. В., Канторович Г. Г.]. Высшая школа экономики, 2000, 112 с. ISBN: 5-7598-0135-X.

28. Бородич С.А. Эконометрика, Серия: Экономическое образование, 2001, 408с. ISBN: 985-6516-45-5.
29. Кремер Н.Ш., Путко Б.А. Эконометрика, ЮНИТИ, 2002, 311с. ISBN: 5-238-00333-1
30. Бард Й. Нелинейное оценивание параметров. М.: Статистика, 1979.
31. Демиденко Е. З. Линейная и нелинейная регрессия. М.: Финансы и статистика, 1981.
32. Джонстон Д. Эконометрические методы. М.: Статистика, 1980.
33. Дрейпер, Н., Смит, Г. Прикладной регрессионный анализ: В 2-х кн. ó М: Финансы и статистика, 1986.
34. Зельнер А. Байесовские методы в эконометрии. М.: Статистика, 1980.
35. Лимер Э. Статистический анализ неэкспериментальных данных. М.: Финансы и статистика, 1983.
36. Маленко Э. Статистические методы эконометрии. М.: Статистика, (вып. 1 - 1975, вып. 2 - 1976).
37. Песаран, М., Слейтер, Л. Динамическая регрессия: теория и алгоритмы. М: Финансы и статистика, 1984.
38. Себер Дж. Линейный регрессионный анализ. М.: Мир, 1980.
39. Андерсон Т. Статистический анализ временных рядов. М.: Мир, 1976.
40. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление. (Вып. 1, 2.) М.: Мир, 1972. [Классика]
41. Дженкинс Г., Ваттс Д. Спектральный анализ и его применения. М: Мир, 1971.
42. Гренджер К., Хатанака М. Спектральный анализ временных рядов в экономике. М.: Статистика, 1972.
43. Кендэл М. Временные ряды. М.: Финансы и статистика, 1981.

Ссылки на сайты

<http://www.mathworks.com>

Сайт фирмы производителя MATLAB. Содержит краткое описание и документацию MATLAB R13. На сайте имеется форум пользователей, приведены примеры решения задач в MATLAB.

<http://www.exponenta.ru>

Образовательный математический сайт. Содержит описание математических пакетов, статистических программных систем, электронные книги, примеры решения задач в MATLAB. На сайте представлены методические разработки по статистике преподавателей ВУЗов.

<http://www.softline.ru>

Сайт крупнейшего продавца программного обеспечения в РФ фирмы SoftLine. На сайте представлены материалы по наиболее известному научно-техническому зарубежному и отечественному программному обеспечению. Кроме описания продаваемого программного обеспечения на сайте выставлены некоторые демонстрационные версии.

<http://www.statistica.ru>

Сайт по статистической системе Statistica. Содержит примеры использования статистических процедур в экономике, промышленности, медицине и д.т. На сайте представлен широко известный электронный учебник по статистике - StatSoft, Inc. (2001). Электронный учебник по промышленной статистике. Москва, StatSoft. WEB: http://www.statsoft.ru/home/portal/textbook_ind/default.htm.

<http://www.mathsoft.com>

Сайт фирмы производителя системы MATHCAD. Интересен подраздел "Статистика" раздела электронные книги. В разделе приведены примеры решения статистических задач, а также выставлена широко известная электронная книга по прикладной статистике "Applied Statistics".

имеры решения задач.

<http://www.statsoft.ru/>

Сайт фирмы StatSoft. Содержит информацию об универсальной статистической системе Statistica. В разделе Download

представлены демонстрационные версии Statistica и ряд программ на Statistica Basic. Приведено решение основных задач статистического анализа данных: визуализация, добыча и анализа данных, прогнозирование, контроль качества, классификация, мониторинг, анализ стабильности технологических процессов. Сайт полезен примерами применения статистических процедур на практике.

<http://www.spss.ru/>

Сайт компании SPSS Inc. производителя широко известной статистической системы SPSS. Содержит описание последней версии SPSS. Содержит примеры статистического анализа данных.

<http://inftech.webservis.ru/it/database/datamining>

Сайт Санкт-Петербургского института информатики и автоматизации РАН. Подробно рассмотрены вопросы интеллектуального анализа данных. Приведены примеры использования статистических методов реализованных в различных системах для интеллектуального анализа данных.

<http://www.amstat.org/> - сайт американской статистической ассоциации. Приведены руководства и статьи по использованию статистических методов обработки информации для анализа экспериментальных данных. Действует форум для посетителей сайта. Представлен раздел ссылок на зарубежные, главным образом англоязычные, web узлы содержащие информацию о статистических методах обработки информации. Рассмотрены вопросы статистической обработки информации в различных системах. В частности приведены примеры имитационного статистического моделирования различных систем с использованием Statistics Toolbox^a MATLAB.

<http://www.alexbar.narod.ru>

На сайте размещены материалы по обработке экспериментальной информации в частности описаны метод главных компонент, робастная регрессия, дискриминантный анализ, факторный анализ.

<http://www.math.rsu.ru/mexmat/kvm/MME/>

На сайте описаны универсальные математические пакеты и прикладные программы. Для MATLAB приведены описания некоторых модулей и пр

infoscope.forth.ru; <http://algolist.manual.ru/maths/>

Сайты содержат справочник по основным используемым статистическим распределениям.

<http://www.nag.co.uk>

Сайт NAG's Statistical software. Содержит описание основных статистических процедур и алгоритмов. Отдельно рассмотрено вопрос реализации методов статистической обработки данных в электронных таблицах, например в Excel и приведено описание возможностей Statistical Add-Ins for Excel.

Большинство книг по MATLAB и его пакетам расширения вышли в издательствах BHV, Питер-Пресс и Диалог-МИФИ. Книги по статистическим методам и учебники по теории вероятностей и математической статистике можно найти на сайтах издательств "Высшая школа", "Финансы и статистика". Адреса сайтов названных издательств приведены ниже в порядке их упоминания:

<http://www.bhv.ru/>

<http://www.piter.com>

<http://www.bitex.ru/~dialog/>

<http://www.v-shkola.ru/>

www.finstat.ru