# SCILAB: Библиотеки

# Глава 6. Библиотека статистики

## Общая критика:

Раздел статистики впервые представлен в версии 2.7. Поэтому в нем попадаются ошибки. Сообщайте о найденных ошибках производителю INRIA!!! Это в наших общих интересах.

В описании определении синтаксиса команд зачастую (и даже как правило) не приводится формула, по которой вычисляется результат данной команды. Мне приходилось подбирать эти формулы самой, исходя из текста подпрограмм. Это большая оплошность производителя: всегда проверяйте результаты выполнения команд на простых примерах. Может быть вычисляется совсем не то, что Вы ожидаете.

Для тех, кто хочет самостоятельно ознакомится с текстовым содержанием подпрограммы библиотеки, следует найти интересующий Вас файл с расширением **.sci** в подкаталоге Scilab `../macros/statistics/`, открыть его в любом текстовым редакторе и попытаться понять его содержание. Имена файлов совпадают с именами подпрограмм. Текст подпрограмм написан на языке пакета Scilab.

---

## Содержание главы:

*Замечание:* О способах создания последовательность случайных чисел для использования в примерах смотрите в главе 7 "Библиотека функций распределения" Удобное для статистики графическое представление в виде гистограммы осуществляется с помощью команд **histplot** и **hist3d** (смотри главу 5 "Графика").

---

# Электронные учебники по статистике

- А. Д. Манита. Учебник по теории вероятности `http://teorver-online.narod.ru/teorver73.html"`
- Учебник StatSoft `http://www.statsoft.ru/home/textbook/default.htm`
- Учебник "Математическая статистика" `http://jenpc.nstu.nsk.su/uchebnik2/`
- Электронный статистический словарь `http://www.kgtu.runnet.ru/WD/TUTOR/textbook/glossary/default.htm`

---

# Как вычислить выборочное среднее значение?

**Способ 1.**
С помощью команды **mean**.
Для набора величин **($x_1$,$x_2$,...,$x_n$)** выборочное среднее совпадает со средним арифметическим и вычисляется по формуле:

$$y = \frac{\sum_{i=1}^{n} x_i}{n}$$

где **n** - объем выборки.
Синтаксис
**y=mean(x)**
**y=mean(x,'r')**
**y=mean(x,'c')**

Параметры
**x** : действительный вектор или матрица
**y** : скаляр или вектор

Если **x** - матрица, у которой число строк равно **n_row**, а число столбцов равно **n_col**, то среднее вычисляется как сумма всех элементов матрицы, деленная на произведения числа строк на число столбцов:
**y=(1/(n_row*n_col))*sum(x)**

Команда **y=mean(x,'r')** (или **y=mean(x,1)**) является "строковым" средним значением. Возвращает вектор-строку **y(j)= mean(x(:,j))**

Если **x** является матрицей:
**y=mean(x,'r')** (или **y=mean(x,1)**) возвращает в качестве каждого элемента вектора-строки **y** среднее арифметическое по соответствующему столбцу матрицы **x**.

**y(j)= mean(x(:,j))**

**y=mean(x,'c')** (или
**y=mean(x,2)**) возвращает в качестве каждого элемента вектора-столбца **y** среднее арифметическое по соответствующей строке матрицы **x**.

Пример.
```
x=[1,2,10;7,7.1,7.01];
```
Результат:
```
x =
! 1. 2. 10. !
! 7. 7.1 7.01 !
M=mean(x) // (1+2+10+7+7.1+7.01)/6
```
Результат:
```
M =
5.685

mean(x,'r')
```
Результат:
```
M_col = ! 4. 4.55 8.505 ! // !(1+7)/2 (2+7.1)/2 (10+7.2)/1!

mean(x,'c')
```
Результат:
```
M_col=
! 4.3333333 ! // (1+2+10)/3
! 7.0366667 ! // (7+7.1+7.01)/3
```

**Способ 2.**
С помощью команды **meanf** вычисляется среднее арифметическое с учетом веса. Для набора величин **($x_1$,$x_2$,...,$x_n$)** выборочное среднее совпадает со средним арифметическим и вычисляется по формуле:

$$y = \frac{\sum_{i=1}^{n} f_i x_i}{\sum_{i=1}^{n} f_i}$$

где **n** - объем выборки.
Синтаксис
**y=meanf(x,fre)**
**y=meanf(x,fre,'r')** или **m=meanf(x,fre,1)**
**y=meanf(x,fre,'c')** или **m=meanf(x,fre,2)**

Параметры
**x** : действительный вектор или матрица
**fre** : действительный вектор или матрица той же размерности, что и **x**

Команда **y=meanf(x,fre,'r')** (или **y=mean(x,fre,2)**) является "строковым" средним значением.

Команда **y=meanf(x,fre,'c')** (или **y=meanf(x,fre,2)**) является "столбцовым" значением.

# Как найти среднее геометрическое?

С помощью команды **geomean**.

Для набора величин **($x_1$,$x_2$,...,$x_n$)** выборочное среднее совпадает со средним арифметическим и вычисляется по формуле:

$$gm = \left(\prod_{i=1}^{n} x_i\right)^{1/n}$$

где **n** - объем выборки.

Синтаксис
**gm=geomean(x)**
**gm=geomean(x,'r')**( или **gm=geomean(x,1)**)
**gm=geomean(x,'c')**(или **gm=geomean(x,2)**)

Параметры
**x** :действительный или комплексный вектор (матрица)
**gm**: скаляр, вектор-строка или вектор-столбец.
Команды вычисляет среднее геометрическое для вектора или матрицы **x**.

Для вектора или матрицы **x** команда **gm=geomean(x)** возвращает скаляр **gm**, равный геометрическому среднему всех элементов **x**.
Команда **gm=geomean(x,'r')** (или **gm=gmean(x,1)** ) возвращает в качестве каждого элемента вектора-строки **gm** среднее геометрическое по соответствующему столбцу матрицы **x**.
Команда **gm=geomean(x,'c')** (или **gm=gmean(x,2)** ) возвращает в качестве каждого элемента вектора-столбца **gm** среднее геометрическое по соответствующей строке матрицы **x**.

Если произведение всех элементов **<0**, то результатом команды **geomean** может быть комплексным числом.

Пример.
```
x=[3 1 5];
g=geomean(x) // (3*1*5)^(1/3)
```
Результат:
```
g =
2.4662121
```

***Замечание:*** В команде **geomean** на текущую версию 2.7 и 2.7.2 для Windows на данный момент содержится ошибка.

Для тех, кто хочет ее исправить самостоятельно, следует найти файл **geomean.sci** в подкаталоге Scilab `../macros/statistics/` и заменить в тексте этой подпрограммы строку

1) `gm=prod(x)^(1/length(a))`

на

`gm=prod(x)^(1/length(x))`

2) строку

`gm=prod(x,orien).^(1/size(a,orien))`

на

`gm=prod(x,orien).^(1/size(x,orien))`

3) сохранить исправленный файл **geomean.sci** под тем же именем.

4) стереть бинарный файл **geomean.bin**, так как он содержит скомпилированную неправильную программу.

После этих операций выполнить из меню главного окна **File** - **Exec** и открыть файл **geomean.sci** из соответствующего каталога.

Другим вариантом является открытие файла в окне редактора **SciPad** и загрузка его оттуда.

Учтите, что компиляция при таких способах не происходит и в новой сессии подпрограмму **geomean** подгружать заново. О том, как ее перекомпилировать в следующий раз (когда сама освою).

---

# Как вычислить гармоническое среднее выборки случайных чисел?

С помощью команды **harmean**.

Для набора величин **($x_1$,$x_2$,...,$x_n$)** гармоническое среднее вычисляется по формуле:

$$gm = \frac{n}{\sum_{i=1}^{n} 1/x_i}$$

где **n** - число наблюдений (объем выборки).

Синтаксис

**hm=harmean(x)**

**hm=harmean(x,'r')**(или, эквивалентно, **hm=harmean(x,1)**)

**hm=harmean(x,'c')**(или, эквивалентно, **hm=harmean(x,2)**)

Параметры

**x**: действительный или комплексный вектор (матрица)

**hm**: скаляр

Если **x** является матрицей: **hm=mean(x,'r')** (или **hm=harmean(x,1)** ) возвращает в качестве каждого элемента вектора-строки **hm** гармоническое среднее по соответствующему столбцу матрицы **x** **hm(j)= harmean(x(:,j))**

**hm=harmean(x,'c')** (или **hm=harmean(x,2)** ) возвращает в качестве каждого элемента вектора-столбца **hm** гармоническое среднее по соответствующей строке матрицы **x**.

***Замечание:*** Кажется, что в некоторых операционных системах при вычислении **harmean(x,'r')** и **harmean(x,'c')** на данный момент содержится ошибка.

Пример.
```
x=[2 4 6];
h=harmean(x);
```

Результат:
```
h =
3.2727273
q=3/(1/2+1/4+1/6); // сравните результаты
```
Результат:
```
q =
3.2727273
```

---

# Как найти размах (амплитуду) распределения случайных величин?

С помощью команды **strange**.

Синтаксис
**[r]=strange(x)**
**[r]=strange(x,'r')** (или **[r]=strange(x,1)**)
**[r]=strange(x,'c')** (или **[r]=strange(x,2)**)

Параметры
**x** :действительный или комплексный вектор (матрица)

Команда **[r]=strange(x)** возвращает расстояние между самым большим и самым маленьким значением матрицы **x** и определяет размах вектора или матрицы **x**.
Команда **[r]=strange(x,'r')** (или **[r]=strange(x,1)**) возвращает вектор-строку размахов по каждому столбцу.
Команда **[r]=strange(x,'c')** (или **[r]=strange(x,2)**) возвращает вектор-столбец размахов по каждой стролке.

Пример 1.
```
x=[2 1 -6 45 0 -2];
r=strange(x)
```

Результат:
```
r =
51. // 45-(-6)=51
```

Пример 2.
```
x=[4 3 -7 10 -20];
y=[2 1 -6 45 0 ];
z=[x;y];
r_all=strange(z); //размах всей матрицы = max(z)-min(z)
r_col=strange(z,"r"); //размах по столбцам
```

```
r_row=strange(z,"c"); //размах по строкам
```

Результат:

```
r_all =
65.

r_col =
! 2. 2. 1. 35. 20. !

r_row =
! 30. ! // =10-(-20)
! 51. ! // =45-(-6)
```

# Как вычислить медиану для распределения величин?

С помощью команды **median**.

Вспомним:
Медиана **x1/2** делит область изменения **x** соответственно на 2 интервала, попадания в которые имеют равные вероятности.
Геометрический смысл: Выборочной медианой называется значение аргумента, которое делит гистограмму плотности вероятности на две равные части.

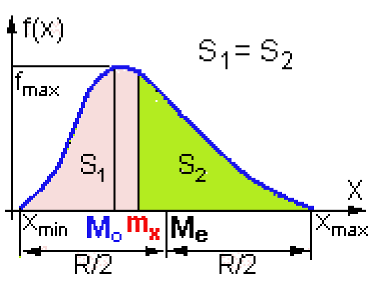
Синтаксис
**y=median(x)**
**y=median(x,'r')**
**y=median(x,'c')**

Параметры

**x** : действительный вектор или матрица
**y** : скаляр или вектор

Для вектора или матрицы **x**, **y=median(x)** возвращает скаляр **y** медианы всех значений элементов **x**. **y=median(x,'r')** (или **y=median(x,1)**) является «строковой» медианой. возвращает в качестве каждого элемента вектора-строки **y** медиану по соответствующему столбцу матрицы **x**.

**y=median(x,'c')** (или **y=median(x,2)**) является «столбцовой» медианой и возвращает в качестве каждого элемента вектора-столбца **y** медиану по соответствующей строке матрицы **x**.

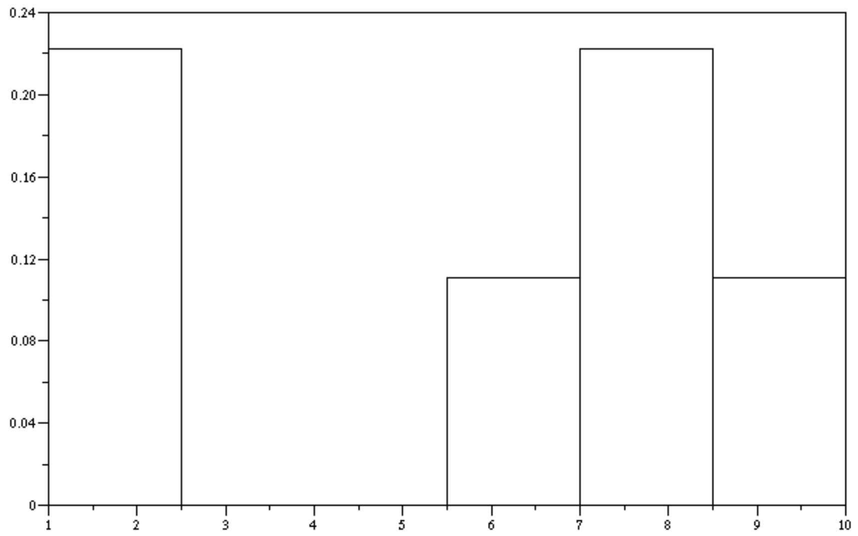



На этом рисунке **$m_x$** является медианой.

Пример.

```
A=[1,2,10;7,7.1,7.01];
histplot(6,A)
```

Результат:




```
y=median(A)
```

Результат:

```
y = 7.005
```

Из рисунка видно, что площадь двух левых столбиков равна площади двух правых столбиков, что совпадает с полученным нами результатом `y=7.005`.

---

# Как вычислить квартиль распределения случайных величин?

С помощью команды **quart**.
Квартили **x1/4**, **x1/2** и **x3/4** делят область изменения **x** соответственно на четыре интервала, попадания в которые имеют равные вероятности.
Различаются три квартиля: **x1/4** - первые 25%, **x1/2** - 50% (совпадает с медианой), **x3/4** - 75%.

Синтаксис
**s=quart(x)**
**s=quart(x,'r')** или **m=quart(x,1)**.
**s=quart(x,'c')** или **m=quart(x,2)**.

Параметры
**x** : действительный или комплексный вектор (матрица)
**s**: вектор. Если **x** - вектор, то имеет размерность 3.

Квартилю **x1/4** соответствует величина **s(1)**, квартилю **x1/2** - величина **s(2)** и квартилю **x3/4** - величина **s(3)**.

Для команды существует «строковый» вариант
**s=quart(x,'r')** или **m=quart(x,1)**
и «столбцовый» вариант
**s=quart(x,'c')** или **m=quart(x,2)**.

Пример.
```
x=[6 7 0 7 10 4 2 2 7 1]
q=quart(x)
q =
! 1.75 !
! 5. !
! 7. !
```

---

# Как вычислить квартильный размах распределения случайных величин?

С помощью команды **iqr**.
Квартильный размах, по определению, равен: верхний квартиль минус нижняя квартиль (75% квартиль минус 25% квартиль). Так как верхний квартиль - это значение, слева от которого находятся 75% наблюдений, а нижний квартиль - это значение, слева от которого находится 25% наблюдении, то квартильный размах представляет собой интервал вокруг медианы, который содержит 50% наблюдений (значений случайной величины).
Команда **iqr** вычисляет область **iqr= верхний квартиль - нижний квартиль**.

Синтаксис
**q=iqr(x)**
**q=iqr(x,'r')** (или, эквивалентно, **q=iqr(x,1)**)
**q=iqr(x,'c')** (или, эквивалентно, **q=iqr(x,2)**)

Параметры
**x** : действительный или комплексный вектор (матрица)

Фактически результат команды **iqr** в случае, если **x** - вектор, может быть получен с помощью следующих комбинаций команд:
```
r=quart(x);
q=x(3)-x(1)
```

Для команды существует «строковый»
**q=iqr(x,'r')** (или, эквивалентно, **q=iqr(x,1)**)
и «столбцовый» вариант
**q=iqr(x,'c')** (или, эквивалентно, **q=iqr(x,2)**)

Пример.
```
x=[6 7 0 7 10 4 2 2 7 1];
q=iqr(x);
```
Результат:
```
q =
5.25
//Для сравнения
k=quart(x);
```

Результат:
```
k =
! 1.75 !
! 5. !
! 7. !
mu_q=k(3)-k(1);
```
Результат:
```
my_k =
5.25
```

---

# Как вычислить процентиль распределения?

С помощью команды **perctl**.
Процентили **x0,01 … x0,99** делят область изменения **x** соответственно на 100 интервалов, попадания в которые имеют равные вероятности.

Синтаксис **p=perctl(x,y)**

Параметры
**x** : действительный или комплексный вектор (матрица)
**y** : вектор положительных значений, каждое из которых удовлетворяет условию: **0 < yi < 100**. В противном случае команда не работает и дает сообщение об ошибке. Вектор может иметь повторяющиеся значения элементов.
**p**: матрица размером **length(y)** на **2** и содержит в первом столбце значение процентиля. Второй столбец этой матрицы содержит место вычисленного процентиля в начальной матрице **x**.

Пример.
```
x=ones(1,5);
// задали все одинаковые значения =1 и все интервалы будут иметь равный
процентиль
y=90:93;
p=perctl(x,y)
```
Результат:
```
p =
! 90.9 90. !
! 91.91 91. !
! 92.92 92. !
! 93.93 93. !
```
Мы получили во втором столбце номер процентиля. У нас это 90-тый, 91-тый, 92-ой и 93-тий, а в правом столбце соответствующее значение этих процентилей. На самом деле интервал все время делится на 100 частей, просто для краткости нам выдаются только избранные нами результаты.

---

# Как вычислить центральные моменты всех порядков?

С помощью команды **cmoment** вычисляется центральный момент или порядок **ord** элементов **x**.

Центральный момент порядка **k** в пакете Scilab определяется по следующей формуле:

$$mom = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - Mx)^{ord}$$

, где **Mx** - является выборочным средним, **n** - число элементов выборки, **ord** - порядок центрального момента.

***Замечание:*** Обратите внимание, что в разных источниках приводятся разные определения центрального момента: это касается нормировки на **(n-1)** и **n**.
Более того: в пакете Scilab при вычислении центрального момента (команда **cmoment**) нормировка **(n-1)**, а при вычислении нецентрального момента (команда **moment**) нормировка на **n**. Это с моей точки зрения очень плохо!

Синтаксис
**mom=cmoment(x,ord)**
**mom=cmoment(x,ord,'r')** или **mom=cmoment(x,ord,1)**
**mom=cmoment(x,ord,'c')** или **mom=cmoment(x,ord,2)**

Параметры
**x** : действительный или комплексный вектор (матрица)
**ord** : положительное целое

Команда **mom=cmoment(x,ord,'r')** или **mom=cmoment(x,ord,1)** используется для вычисления "столбцового" центрального момента.
Команда **mom=cmoment(x,ord,'c')** или **mom=cmoment(x,ord,2)** используется для вычисления "строкового" центрального момента.

Пример.
```
x=[0.2113249 0.0002211 0.6653811 0.7560439 0.3303271 0.6283918]
k=2;
n=length(x); // число элементов в x
mom=cmoment(x,k)
// Для проверки
mx=mean(x);
w=0;
for i = 1:n, w=w+(x(i)-mx)^k;end;
w=w/(n-1)
```

Результат:
```
mom =
.0887018

w =
.0887018
```

---

# Как вычислить нецентральные моменты всех порядков?

С помощью команды **moment**.

Нецентральный момент порядка **k** в пакете Scilab вычисляется по следующей формуле (для вектора):

$$mom = \frac{1}{n}\sum_{i=1}^{n}(x_i)^{ord}$$

, где **Mx** - является выборочным средним, **n** - число элементов выборки, **ord** - порядок центрального момента.

***Замечание:*** Обратите внимание, что в разных источниках приводятся разные определения центрального момента: это касается нормировки на **(n-1)** и **n**.
Более того: в пакете Scilab при вычислении центрального момента (команда **cmoment**) нормировка **(n-1)**, а при вычислении нецентрального момента (команда **moment**) нормировка на **n**. Это с моей точки зрения очень плохо!
Синтаксис
**mom=moment(x,ord)**
**mom=moment(x,ord,'r')** или **mom=moment(x,ord,1)**
**mom=moment(x,ord,'c')** или **mom=moment(x,ord,2)**
Параметры
**x** : действительный или комплексный вектор или матрица
**ord** : положительное целое число Команда **moment(x,ord)** вычисляет нецентральный момент порядка **ord** элементов вектора (матрицы) **x**.

Для команды существует «строковый» **mom=moment(x,ord,'r')** или **mom=moment(x,ord,1)**
и «столбцовый» вариант
**mom=moment(x,ord,'c')** или **mom=moment(x,ord,2)**

Пример.
```
x=[0.2113249 0.0002211 0.6653811 0.7560439 0.3303271 0.6283918]
k=3;
n=length(x); // число элементов в x
mom=moment(x,k)
w=0;
for i = 1:n, w=w+x(i)^k;end;
w=w/n
```

Результат:
```
mom =
.1700601

w =
.1700601
```

---

# Как вычислить простое (абсолютное) среднее отклонение?

С помощью команды **mad**.

Абсолютное среднее отклонение вычисляется для набора величин **($x_1$,$x_2$,...,$x_n$)**по следующей формуле:

$$s = \frac{\sum_{i=1}^{n} |x_i - Mx|}{n}$$

, где **Mx** - является выборочным средним, **n** - число элементов выборки.

Синтаксис
**s2=mad(x)**
**s2=mad(x,'r')** или **s2=mad(x,1)**
**s2=mad(x,'c')** или **s2=mad(x,2)**

Параметры
**x** : действительный или комплексный вектор или матрица
Если **x** является вектором, то
**s2=(|$x_1$-M|+|$x_2$-M|+|$x_n$-M|)/n** ,
где **M=mean(x)** - среднее арифметическое значение.

**s2=mad(x,'r')** (или **s2=mad(x,1)** ) вычисляет «строковое» среднее отклонение
**s2=mad(x,'c')** (или **s2=mad(x,2)** ) вычисляет «столбцовое» среднее отклонение.

Пример.
```
x=[565 6.234 143];
d=mad(x)
```

Результат:
```
d =
217.948
```

```
// Для проверки
n=3;
s=abs(x(1)-mean(x))+abs(x(2)-mean(x))+abs(x(3)-mean(x));
s=s/n
```

Результат:
```
s =
217.948
```

---

# Как вычислить среднее квадратичное отклонение?

**Способ 1.**
С помощью команды **msd**.
Пусть **x1**,**x2**, **x3**, ...**xn** - случайные величины, **n** - объем выборки, а **xm** - выборочное среднее (арифметическое).
***Замечание:*** Выборочное среднее **xm** может быть получено с помощью команды **mean(xm)**.
Среднее квадратичное отклонение вычисляется по формуле: **y=sqrt(D)**, где **D=((x1-xm)$^2$+(x2-xm)$^2$+...+(xn-xm)$^2$)/n** ; **xm=mean(x)**
Среднее квадратичное отклонение в пакете Scilab вычисляется по следующей формуле (для вектора):

$$y = \sqrt{\frac{\sum_{i=1}^{n}(x_i - Mx)^2}{n}}$$

, где **n** - число элементов и **Mx** - выборочное среднее.

***Замечание:*** Обратите внимание, что результат этой команды отличается от результата, полученного с помощью команды **stdev** только нормировкой.

Синтаксис
**y=msd(x)**
**y=msd(x,'r')** или **m=msd(x,1)**
**y=msd(x,'c')** или **m=msd(x,2)**

Параметры
**x** : действительный или комплексный вектор или матрица

"Строковые" и "столбцовые" варианты всех команд устроены аналогично командам **mean**, **harmean** и т. д.
Команда **y=msd(x,'r')** (или **y=msd(x,1)** ) вычисляет «строковое» среднеквадратичное (или “стандартное”)отклонение.
Команда **y=msd(x,'c')** (или **m=msd(x,2)** ) вычисляет «столбцовое» средне квадратичное отклонение.

Пример.
```
x=[20 40 30];
m=msd(x)
```
Результат:
```
m =
8.1649658

// Для проверки
xm=mean(x);
n=3; // число элементов вектора x
q=(x(1)-xm)^2+(x(2)-xm)^2+(x(3)-xm)^2;
w=sqrt(q/n)
```

Результат:
```
w = 8.1649658
```
Получили результат, равный **m**. Следовательно, команда считает правильно.

**Способ 2.**
С помощью команды **stdev** .

Среднее квадратичное отклонение (standard deviation) в пакете Scilab вычисляется по следующей формуле (для вектора):

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - Mx)^2}{n-1}}$$

, где **n** - число элементов и **Mx** - выборочное среднее.

***Замечания***
1) Обратите внимание, что результат этой команды отличается от результата, полученного с помощью команды **msd** только нормировкой.
2) В Scilab есть еще команда **st_deviation**, которая совершенно идентична команде **stdev**, как по синтаксису, так и по результату.

Синтаксис
**s=stdev**
**s=stdev(x,'r')** или **m=stdev(x,1)**
**s=stdev(x,'c')** или **m=stdev(x,2)**

Параметры
**x** : действительный или комплексный вектор или матрица
**s**: скаляр

Команда **s=stdev(x,'r')** (или **s=stdev(x,1)** ) является "строковым вариантом".
Команда **s=stdev(x,'c')** (или **s=stdev(x,2)**) является "столбцовым" вариантом.

Пример.
```
x=[0.2113249 0.0002211 0.6653811 0.7560439 0.3303271 0.6283918]
n=length(x); // число элементов в x
s=stdev(x)
//для проверки
mx=mean(x);
w=0;
for i = 1:n, w=w+(x(i)-mx)^2;end;
w=sqrt(w/(n-1))
```

Результат:
```
s =
.2978284

w =
.2978284
```

**Способ 3.**
С помощью команды **stdevf** .
Среднее отклонение с учетом веса (взвешенное среднее квадратичное отклонение)в пакете Scilab вычисляется по следующей формуле (для вектора):

$$s = \frac{\sqrt{\sum_{i=1}^{n} f_i (x_i - Mx)^2}}{\sum_{i=1}^{n} f_i - 1}$$

, где $\mathbf{x_i}$ - набор случайных величин, $\mathbf{f_i}$ - набор соответствующих им весов, **n** - число элементов, а **Mx=meanf(x,f)** - среднее арифметическое с учетом веса.

Синтаксис
**s=stdevf(x,fre)**
**s=stdevf(x,fre,'r')** или **s=stdevf(x,fre,1)**
**s=stdevf(x,fre,'c')** или **s=stdevf(x,fre,2)**

Параметры
**x**: действительный или комплексный вектор (матрица)
**fre**: вектор (матрица)того же размера, что и **x**.

Пример.
```
x=[0.2113249 0.0002211 0.6653811 0.7560439 0.9546254 0.6283918]
fre=[1 2 3 3 4 3]
n=length(x); // число элементов в x
s=stdevf(x,fre)
//для проверки
w=0;
for i = 1:n, w=w+fre(i)*(x(i)-meanf(x,fre))^2;end;
w=sqrt(w)/(sum(fre)-1)
```

Результат:
```
s =
.0800239

w =
.0800239
```

# Как вычислить дисперсию значений вектора или матрицы?

**Способ 1.**
С помощью команды **variance**.

Значение дисперсии для вектора случайных величин **$x=(x_1, x_2, \ldots ,x_n)$** по формуле:

$$s = \frac{\sum_{i=1}^{n}(x_i - Mx)^2}{n-1} \quad , \quad \text{где} \quad Mx = \frac{\sum_{i=1}^{n} x_i}{n}$$

***Важное замечание***:
Обычно в русских учебниках в формулу для дисперсии входит деление не на величину **(n-1)**, а на величину **n**.

Синтаксис
**s=variance(x)**
**s=variance(x,'r')** или **m=variance(x,1)**
**s=variance(x,'c')** или **m=variance(x,2)**

Параметры
**x** : действительный или комплексный вектор (матрица)

Команда **s=variance(x,'r')** (или **s=variance(x,1)**) является «строковым» вариантом.
Команда **s=variance(x,'c')** (или **s=variance(x,2))** является «столбцовым» вариантом.

Пример.
```
x=[2 0 6 7 4 0.6];
d=variance(x)
```

Результат:
```
d =
8.2666667
// Для проверки
n=length(x); // число элементов вектора x
mx=mean(x);
y=x-mean(x);
w=y^2;
w=sum(w)/(n-1)
```

Результат (совпадает со значением `d`):
```
w =
8.2666667
```

**Способ 2.**
С помощью команды **variancef**. Эта команда вычисляет дисперсию с учетом веса, если наблюдаемые значения **x** неравнорассеяны (неравноточны).
Значение дисперсии для вектора случайных величин $\mathbf{x=(x_1, x_2, ... ,x_n)}$ с учетом вектора веса $\mathbf{f=(f_1, f_2, ... ,f_n)}$ по формуле:

$$s = \frac{\sum_{i=1}^{n}(x_i * f_i - Mx)^2}{n-1} \quad , \quad \text{где} \quad Mx = \frac{\sum_{i=1}^{n} x_i}{n}$$

Синтаксис
**s=variancef(x,fre)** (или **s=variancef(x,fre,'*')**)
**s=variancef(x,fre,'r')** (или **s=variancef(x,fre,1)** )
**s=variancef(x,fre,'c')** (или **s=variancef(x,fre,2)** )

Параметры
**x** : действительный или комплексный вектор (матрица)
**fre**: имеет тот же тип и размер, что и **x**, и является векирором (матрицей) веса наблюдений.

Матрица весов **fre** должна быть того же типа, что и **x**. В вычислении учитывается значение вектора (матрицы) соответствующих весов. Почему-то для возможности осуществления счета сумма всех элементов **fre (=sum(fre))** должна быть > 1.

***Замечание:***
Если все веса матрицы **x** равны, т.е. все элементы матрицы **fre** равны единице, то результаты выполнения команд **variancef** и **variance** будут совпадать.

Пример.
```
x=[2 5 7 1];
fre=[1 1 1 1];
v=variance(x)
```
Результат:
```
v =
7.5833333
vf=variancef(x,fre)
```

Результат:
```
vf =
7.5833333 // совпадает с v
```

# Как преобразовать матрицу в «усредненную» по амплитуде?

**Способ 1.**
С помощью команды **center**. В результате работы команды произойдет уменьшение значения каждого элемента матрицы данных (т. е. каждую случайную величину)на среднюю амплитуду, равную среднему арифметическому из элементов матрицы. (англ. термин = mean deviation).

Синтаксис
**s=center(x)**
**s=center(x,'r')** или **s=center(x,1)**
**s=center(x,'c')** или **s=center(x,2)**

Параметры
**x**, **s**: действительный или комплексный вектор (матрица)

Команда вычисляет из матрицы **x** в «усредненную» матрицу **s**, которая вычисляется следующим образом:
**s** - это матрица, каждый элемент которой **s(i,j)** вычисляется как разность соответствующего элемента **x(i,j)** и **xbar**:
**s(i,j )=x(i,j)-xbar** ,
где **xbar=(x(1)+x(2)+…+x(n))/n**
Значение величины **xbar** является выборочным средним значением матрицы **x** и может быть вычислено с помощью команды **mean(x)**.
Фактически из всех элементов матрицы вычитается значение выборочного среднего матрицы.

**s= center (x,'r')** (или **s= center (x,1)** ) «строковое» среднеквадратичное (или “стандартное”)отклонение.
**s= center (x,'c')** (или **s= center (x,2)** ) «столбцовое» среднеквадратичное отклонение.

Пример.
```
x=[0.2113249 0.0002211 0.6653811; 0.7560439 0.3303271 0.6283918]
s=center(x)
```

Результат:
```
s =
! - .2206234 - .4317272 .2334328 !
! .3240956 - .1016212 .1964435 !
//Для сравнения
bar=mean(x);
z=x-bar
```

Результат:
```
z =
! - .2206234 - .4317272 .2334328 !
! .3240956 - .1016212 .1964435 !
```

Мы видим, что значение `s` совпадает со значением `z`.

**Способ 2.**
С помощью команды **wcenter**.
Команда производит преобразование матрицы в "усредненную" (mean deviation)по амплитуде с учетом дисперсии по следующей формуле (для вектора):

$$s_i = (x_i - Mx)/\sigma \quad ,$$

где $$Mx = \sum_{i=1}^{n} x_i/n \quad ,$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - Mx)^2}{n-1}} \quad .$$

Делитель в формуле для $\mathbf{s_i}$ быть вычислен с помощью команд **stdev(x)**, величина **Mx** - является средним арифметическим и может быть получено с помощью команды **mean**.

Синтаксис
**s=wcenter(x)**
**s=wcenter(x,'r')** или **s=wcenter(x,1)**
**s=wcenter(x,'c')** или **s=wcenter(x,2)**

Параметры
**x**, **s**: действительный или комплексный вектор (матрица)

Команда **s=wcenter(x,'r')** (или **s=wcenter(x,1)**) - это «строковый» вариант команды, возвращающий
**s(i,j)** как **(x(i,j)-xbarv(j))/sigmav(j)**,
где **xbarv(j)** - выборочное среднее значение (команда **mean**) значений элементов **j**-го столбца и
**sigmav(j)** является стандартным отклонением **j**-го столбца матрицы **x**.

Команда **s=wcenter(x,'r')** (или **s=wcenter(x,1)**) - это «столбцовый» вариант команды, возвращающий
**s(i,j)** как **(x(i,j)-xbarv(j))/sigmav(j)**,
где **xbarv(j)** - выборочное среднее значение (mean) значений элементов **j**-го строки и
**sigmav(j)** является стандартным отклонением (standard deviation) **j**-й строки матрицы **x**.

Пример.
```
x=[21 2 6 7];
n=length(x);
wc=wcenter(x)
// для проверки
sigma=stdev(x);
mx=mean(x);
s=zeros(n);
for i = 1:n, s(i)=(x(i)-mx)/sigma;end;
w=s'
```

Результат:
```
wc =
! 1.4481324 - .8447439 - .3620331 - .2413554 !

w =
! 1.4481324 - .8447439 - .3620331 - .2413554 !
```

---

# Как вычислить соотношение Фишера (Fischer ratio)?

**Способ 1.**
С помощью команды **ftest**.
Синтаксис
**f=ftest(samples)**
**[f,p]=ftest(samples)**
Параметры
**samples** : действительная или комплексная матрица размером **nr** на nc

Пример.
```
samples=[46 55 54;
53 54 50;
49 58 51;
50 61 51;
46 52 49]
[f,p]=ftest(samples)
```

**Способ 2.**
С помощью команды **ftuneq**. Вычисляется соотношение Фишера для образцов неравной длины. Смотри подробно **help ftuneq**.

---

# Как вычислить частоту встречаемости значения случайной величины?

С помощью команды **tabul**.
Синтаксис
**[m]=tabul(x)**

Параметры
**x** : действительный, комплексный вектор (матрица) или вектор (матрица) из символьных переменных

Если **x** - числовой вектор или матрица, то **m**: матрица из двух столбцов, в первом из которых содержатся числовые значения вектора **x**, отсортированные по убыванию (упорядоченный по убыванию вариационный ряд), а во втором содержится целое число, равное тому, как часто повторяется это значение в векторе x. Если элементы **x** являются символьными, то **m** - является списком.

Пример 1.
```
x=[2 3 3 4 2 5 4 4 4 5];
t=tabul(x)
```

Результат:
```
t =
! 5. 2. !
! 4. 4. !
! 3. 2. !
! 2. 2. !
```

Пример 2.
```
y=["cat" "cat" "dog" "dog" "pig"];
t=tabul(y)
```

Результат:
```
t =
t(1)
!cat !
! !
!dog !
! !
!pig !

t(2)
! 2. !
! 2. !
! 1. !
```

---

# Как вычислить коэффициенты регрессии двух величин?

С помощью команды **regress**. Регрессия в теории вероятностей и математической статистике, зависимость среднего значения какой-либо величины от некоторой другой величины или от нескольких величин.
Результатом выполнения команды будет такая матрица **coefs =[a b]** размером 1 на 2, что **y=$C_1$+$C_2$*x** будет уравнением, аппроксимирующим наши дискретные данные по методу наименьших квадратов согласно регрессионной модели.

Команда **regress** вычисляет коэффициенты регрессии $C_1$ и $C_2$, по следующим формулам:

$$C_2 = \frac{\sum_{i=1}^{n}(x_i - Mx)(y_i - My)}{\sum_{i=1}^{n}(x_i - Mx)^2}, \quad C_1 = My - C_2 Mx,$$

$$\text{где} \quad Mx = \frac{\sum_{i=1}^{n} x_i}{n} \quad \text{и} \quad My = \frac{\sum_{i=1}^{n} y_i}{n}.$$

Синтаксис
**coefs=regress(x,y)**

Параметры **x**, **y** : действительные или комплексные векторы с одинаковым числом элементов **n**.

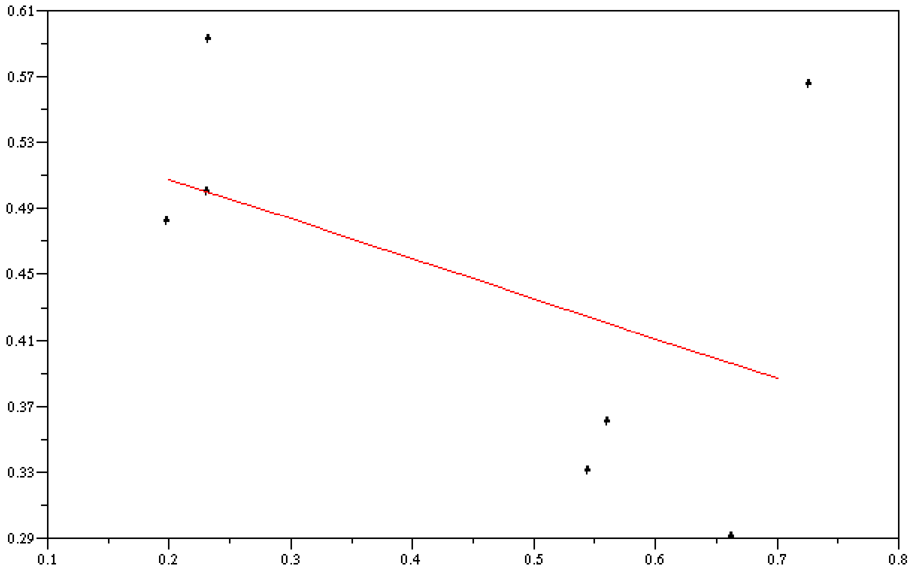
Значение **coefs(1)** равно $C_1$ из приведенной выше формулы, а значение **coefs(2)** соответственно равно $C_2$.

Пример.

```
x=[0.5608486 0.6623569 0.7263507 0.1985144 0.5442573 0.2320748 0.2312237];
y=[0.3616361 0.2922267 0.5664249 0.4826472 0.3321719 0.5935095 0.5015342];
coefs=regress(x,y)
plot2d(x,y,-8); // -8 - означает то, что точки не соединены линиями
t=0.2:0.05:0.7;
q=coefs(1)+coefs(2)*t;
plot2d(t,q,5); // это аппроксимирующая кривая
```

Результат:

```
coefs = ! .5563731 ! ! - .2422534 !
```



## Как вычислить коэффициент корреляции?

С помощью команды **correl** (лат.Correlatio - взаимозависимость).
В пакете Scilab коэффициент корреляции **rho** между двумя наборами случайных величин **x** и **y** с учетом матрицы весов **f** определяется по следующим формулам:

$$f = \begin{pmatrix} f_{11} \cdots f_{1m} \\ \cdots \\ f_{n1} \cdots f_{nm} \end{pmatrix} \quad , \qquad x = (x_1 \cdots x_n) \quad , \qquad y = (y_1 \cdots y_m) \quad ,$$

$$p_x = \begin{pmatrix} \sum_{k=1}^{n} f_{1k} \\ \cdots \\ \sum_{k=1}^{n} f_{nk} \end{pmatrix} \quad , \qquad p_y = \left( \sum_{k=1}^{m} f_{k1} \quad \cdots \quad \sum_{k=1}^{m} f_{km} \right) \quad ,$$

$$M_x = \sum_{k=1}^{n} x_k (p_x)_k \quad , \quad M_y = \sum_{k=1}^{m} y_k (p_y)_k \quad ,$$

$$s_{xy} = \frac{\sum_{k=1}^{n} \sum_{l=1}^{m} (x_k - M_x)(y_l - M_y) f_{kl}}{\sum_{i=1}^{n} \sum_{j=m}^{n} f_{ij}} \quad ,$$

$$s_x = \sqrt{\sum_{k=1}^{n} p_x (x_k - M_x)^2} \quad , \qquad s_y = \sqrt{\sum_{k=1}^{m} p_y (y_k - M_y)^2} \quad ,$$

$$\rho = \frac{s_{xy}}{s_x s_y} \quad .$$

***Замечание:*** Вектора **x** и **y** могут иметь разную длину.

Синтаксис
**rho=correl(x,y,fre)**

Параметры
**x** : действительный или комплексный вектор
**y** : действительный или комплексный вектор
**fre** : матрица размера **length(x)** на **length(y)**

Команда **correl(x,y,fre)** вычисляет корреляцию двух величин **x** и **y**. В весовой матрице **fre** элемент и индексом **(i,j)** соответствует величина или номер частоты $\mathbf{x_i}$, $\mathbf{y_j}$.

Пример.
```
x=[2.5 7.5 12.5 17.5]
h=[0 1 2]
fre=[.03 .12 .07;.02 .13 .11;.01 .13 .14;.01 .09 .14]
rho=correl(x,h,fre)
```

Результат:
```
rho =
.2097870
```

---

# Как вычислить ковариацию двух величин?

С помощью команды **covar**.
Ковариация служит мерой взаимной связи между случайными величинами **y** и **x**, то есть стремление одной случайной величины возрастать или убывать при возрастании или убывании другой случайной величины. Ковариация характеризует меру стохастической связи между случайными величинами. Если случайные величины независимы, то ковариация равна нулю. Обратное верно не всегда.

Синтаксис
**s=covar(x,y,fre)**

Параметры
**x** : действительный или комплексный вектор
**y** : действительный или комплексный вектор
**fre** : матрица размера **length(x)** на **length(y)**

Команда **covar(x,y,fre)** вычисляет ковариацию двух величин **x** и **y**.
В матрице **fre** элемент и индексом **(i,j)** соответствует величина или номер частоты $(\mathbf{x_i y_j})$.
Смотри подробно **help covar**.

Пример.
```
x=[10 20 30 40]
y=[10 20 30 40]
fre=[.20 .04 .01 0; .10 .36 .09 0; 0 .05 .10 0; 0 0 0 .05];
s=covar(x,y,fre)
```

---