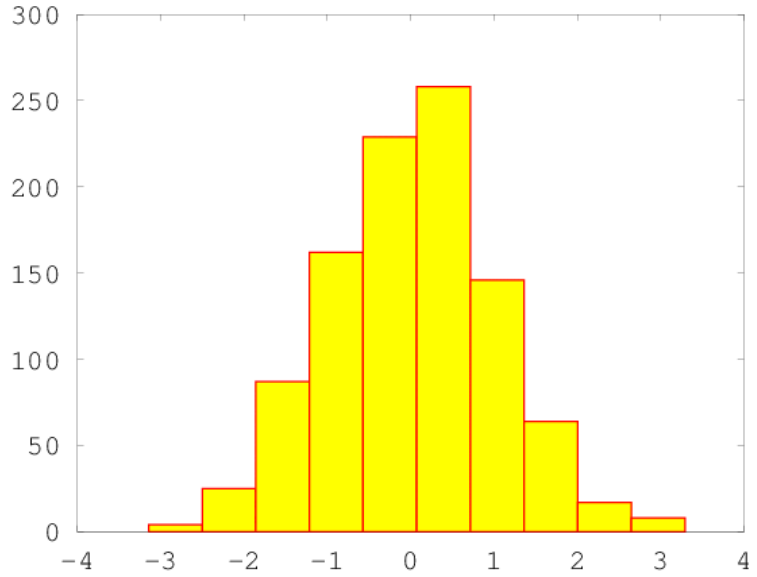
Занятие 7. Статистика в GNU Octave



1. Среднее значение и доверительный интервал выборки
2. Распределения
3. Гистограммы и генерация случайных величин
4. Циклы `for` и `while`, конструкция `if…then…else`
5. Создание m-скриптов и m-функций

Среднее и доверительный интервал

Функция	Вызов	Описание

`mean`	`mean(x)`
	`mean(x,1)`
	`mean(x,2)`
		Среднее значение вектора x
		Средние значения столбцов матрицы x
		Средние значения строк матрицы x

`std`	`std(x)`
	`std(x,f)`
	`std(x,f,1)`
	`std(x,f,2)`
		Стандартное отклонение вектора x
		$f = 0$ – несмещ.дисп., $f = 1$ – смещ.дисп.
		Стандартное отклонение столбцов матрицы x
		Стандартное отклонение строк матрицы x

`tinv`	`tinv(p,f)`	Левосторонний квантиль t-распределения
`median`	`См. mean`	Расчёт медианы
`numel`	`numel(x)`	Число элементов вектора или матрицы x
`sum`	`См. mean`	Расчёт суммы

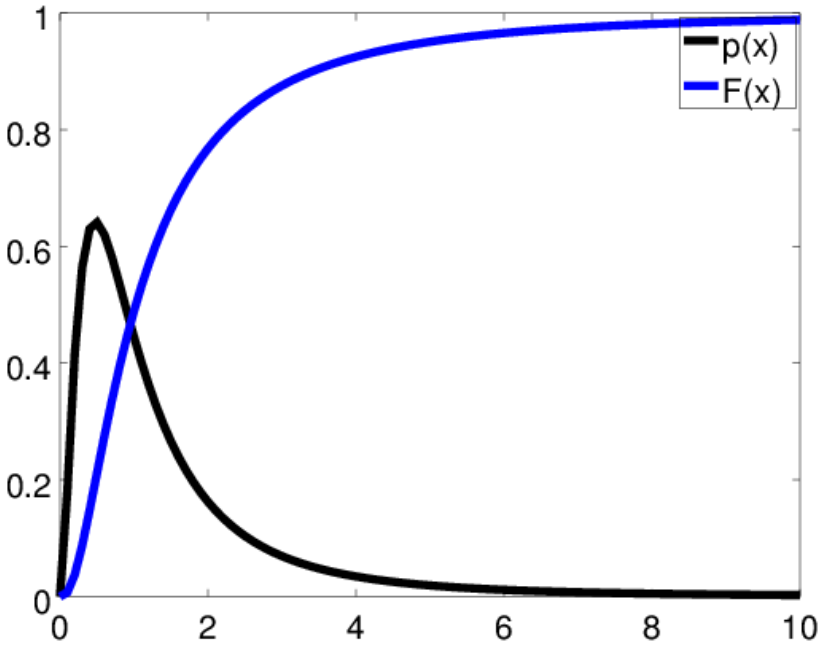
```
>> x = [66.30 66.30 66.26 66.37 66.30];
>> mean(x)
ans =  66.306
>> std(x)
ans =  0.039749
>> std(x)*tinv(0.975,numel(x)-1)/sqrt(numel(x))
ans =  0.049355
```

Функции распределения и квантили

Распределение	$p(x)$	$F(x)$	Левосторонний квантиль
Нормальное	`normpdf(x,mu,s)`	`normcdf(x,mu,s)`	`norminv(p,mu,s)`
Стьюдента (t)	`tpdf(x,f)`	`tcdf(x,f)`	`tinv(p,f)`
Пирсона (χ^2)	`chi2pdf(x,f)`	`chi2cdf(x,f)`	`chi2inv(p,f)`
Фишера (F)	`fpdf(x,f1,f2)`	`fcdf(x,f1,f2)`	`finv(p,f1,f2)`

Pdf – probability density function (функция плотности вероятности)
Cdf – cumulative distribution function (интегральная функция распределения)
Inv – inverse function (обратная функция)

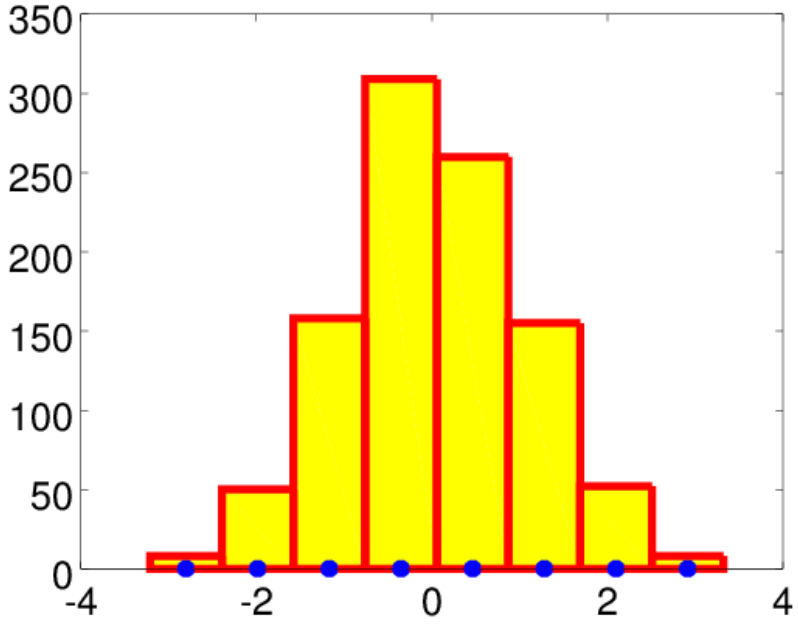
```
>> x = 0:0.1:10;
>> y1 = fpdf(x,6,5);
>> y2 = fcdf(x,6,5);
>> plot(x,y1,'k-',LineWidth',5);
>> hold on;
>> plot(x,y1,'k-',LineWidth',5);
>> hold off;
>> legend('p(x)','F(x)');
>> set(gca,'FontSize',20);
```



Случайные числа и гистограммы

Функция	Вызов	Описание
`rand`	`rand(m,n)`	Матрица mxn из равномерно распределенных случайных чисел из интервала [0;1]
`randn`	`randn(m,n)`	Матрица mxn нормально распределенных случайных чисел ($\mu = 0; \sigma = 1$)
`hist`	`hist(x,n)` `[nn,xx]=hist(x,n)`	График с гистограммой Гистограмма в числах (x – данные, n – число карманов, nn – частоты, xx – центры карманов)

```
>> x = randn(1,1000);
>> hist(x,8, 'FaceColor',
'yellow', EdgeColor, 'red',
'LineWidth', 5);
>> [nn,xx] = hist(x,8);
>> hold on;
>> plot(0,xx,'bo');
>> hold off;
```



Скрипты и функции

Интерактивный режим – не единственный возможный
GNU Octave – полноценный язык программирования, дающий возможность хранить исходные тексты программ в файлах 2 видов

Все файлы с исходными текстами – текстовые и с расширением .m

m-скрипт

Не имеет входных или выходных аргументов. Создаёт переменные в глобальном рабочем пространстве

scriptex.m:

```
% Comment with help
x = magic(3);
disp('Hello');

>> scriptex
Hello
>> disp(x)
   8   1   6
   3   5   7
   4   9   2
```

m-фунцкия

Есть входные и выходные аргументы. Создаёт переменные в локальном рабочем пространстве

funcex.m

```
% Comment with help
function [s,p] = sumprod(a,b)
s = a+b;
p = a*b;
end

>> [s,p]=sumprod(2,3)
s =  5
p =  6
```

Функции sprintf и fprintf

Функция `sprintf`: форматный вывод в строку

```
str = sprintf(fmt, arg1, arg2, ….);
```

Функция `fprintf`: форматный вывод на экран или в файл

```
fprintf(fmt, arg1, arg2, ….); % На экран
fprintf(fd, fmt, arg1, arg2,…); % В файл
```

`fd` – дескриптор файла

`fmt` – строка с описанием формата вывода. Последовательности, начинающиеся на %, заменяются входными аргументами

Формат	Значение	Формат	Значение
`%d`	Целое	`%f`	Десятичная дробь
`%.10d`	Целое, дополненное до 10 знаков нулями слева	`%.3f`	Десятичная дробь с 3 знаками после запятой
`%10d`	Целое, дополненное до 10 знаков пробелами слева	`%10.3f`	То же, но дополненная до 10 знаков пробелами слева
`%s`	Строка	`%e`	Экспоненциальная запись
`\n`	Перевод строки	`%15.3e`	То же, но с 3 знаками после запятой и дополненная до 15 знаков пробелами слева

См. документацию по функциям `sprintf`, `fprintf` и `printf` в языке C

Функции sprintf и fprintf

Использование со скалярами и строками

```
>> fprintf('%d\n\n', 10);
10

>> fprintf('%10.3f, %5.1f\n',
  2.1, 3.5);
     2.100,   3.5

>> fprintf('%15.3e\n', 123456);
      1.235e+05

>> fprintf('%s %e %f %d\n',
  'ab', 1, 2, 3, 4);
ab 1.000000e+00 2.000000 3

>> sprintf('%.3e', 1.3)
ans = 1.300e+00

>> fprintf('%X %o', 255, 63)
 FF 77
```

Использование с матрицами

```
>> fprintf('%d\n\n', 10);
10

>> sprintf('<%.3f>', rand(1,4));
ans=<0.416><0.568><0.225><0.467>

>> sprintf('%d',[1 2 3;4 5 6]);
ans = 1 4 2 5 3 6

>> fprintf('%d %d %d\n',
  [1 2 3;4 5 6]')
1 2 3
4 5 6
```

Порядок хранения элементов матрицы – «Фортрановский» (по столбцам)

Циклы for и while

Цикл for

Код:

```
for i=1:10
    fprintf('%d ', i);
end
```

Результат:

```
 1  2  3  4  5  6  7  8  9  10
```

Код:

```
for i=[1  3  10  15]
    fprintf('%d ', i);
end
```

Результат:

```
1  3  5  10
```

Код:

```
for i=[1  2  3; 4  5  6]
    fprintf('%d ',i)
end
```

Результат:

```
1  4  2  5  3  6
```

Цикл while

Код:

```
x = 1;
while x < 32
    fprintf('%d ', x);
    x = x * 2;
end
fprintf('\n');
```

Результат:

```
1  2  4  8  16
```

Заменяйте циклы на матричные операции при первой возможности «Think vectorized»

Конструкция if…then…elseif…else

Конструкция

```
if условие_1
   ..операторы..
elseif условие_2
   ..операторы..
else условие_n
   ..операторы..
end
```

Ветви с else и elseif не являются обязательными

Пример

```
if x^2+y^2<1
   disp('Внутри круга');
elseif x^2+y^2==1
   disp('На окружности');
else
   disp('Вне круга');
end
```

Оператор	Значение
~	«НЕ»
==	Равно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно
~=	Не равно
&&	Скалярное «И»
||	Скалярное «ИЛИ»

Не путайте == и =

```
>> a = 0;
>> if a = 1;disp('KU-KU');end;
KU-KU
```

Логические матрицы

Работа с подматрицами

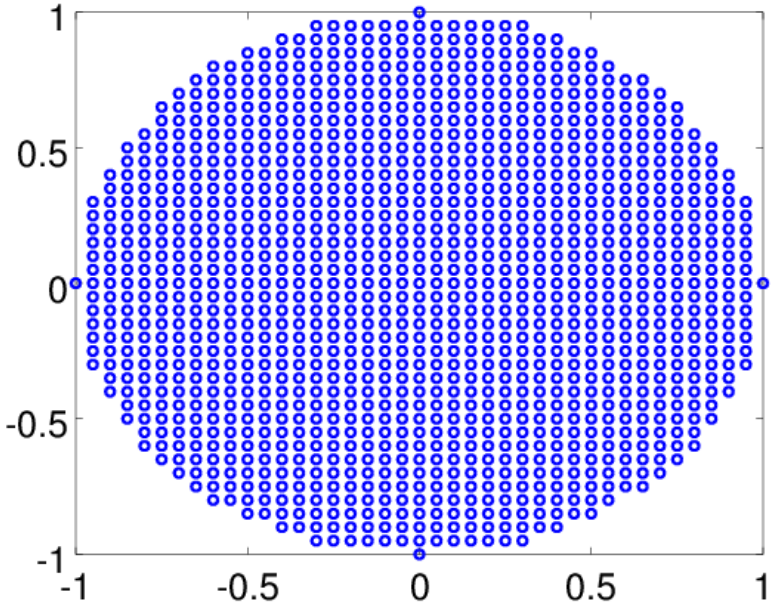
```
>> a = magic(3)
a =
    8    1    6
    3    5    7
    4    9    2

>> ii = a < 6
ii =
    0    1    0
    1    1    0
    1    0    1

>> a(ii) = NaN
a =
      8    NaN      6
    NaN    NaN      7
    NaN      9    NaN
```

Выборка значений из матрицы

```
>> xv = -1:0.05:1;
>> [X,Y] = meshgrid(xv,xv);
>> ii = X.^2 + Y.^2 <= 1;
>> plot(X(ii),Y(ii),'bo');
>> set(gca,'FontSize',24);
>> print(gcf,'a','-dpng','-r75');
```



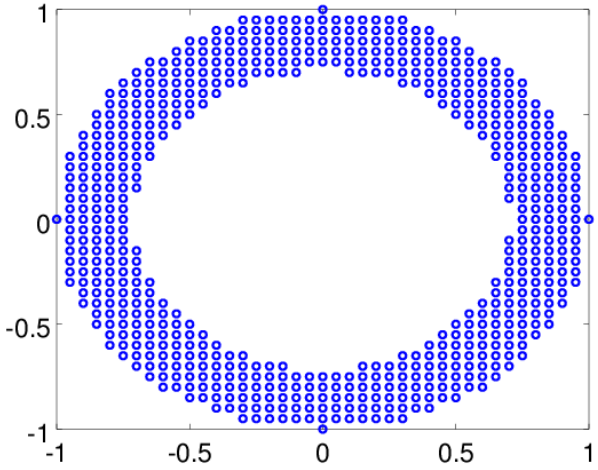
Логические матрицы: векторные логические операции

Оператор	Значение
~	«НЕ»
&	Матричное «И»
|	Матричное «ИЛИ»

Операции сравнения – те же, что и для скаляров

Функция	Значение
any(x,1)	Векторное «ИЛИ» по столбцам матрицы
any(x,2)	Векторное «ИЛИ» по строкам матрицы
all(x,1)	Векторное «И» по столбцам матрицы
all(x,2)	Векторное «И» по строкам матрицы

```
>> a = [1 1 1; 1 1 0; 0 0 0];
>> a
   1   1   1
   1   1   0
   0   0   0
>> any(a,1)
   1   1   1
>> any(a,2)
   1
   1
   0
>> all(a,1)
0   0   0
>> all(a,2)
   1
   0
   0
```




```
>> xv = -1.5:0.05:1.5;
>> [X,Y] = meshgrid(xv,xv);
>> r2 = X.^2 + Y.^2;
>> ii = 0.5 <= r2 & r2 <= 1;
>> plot(X(ii),Y(ii),'bo');
```

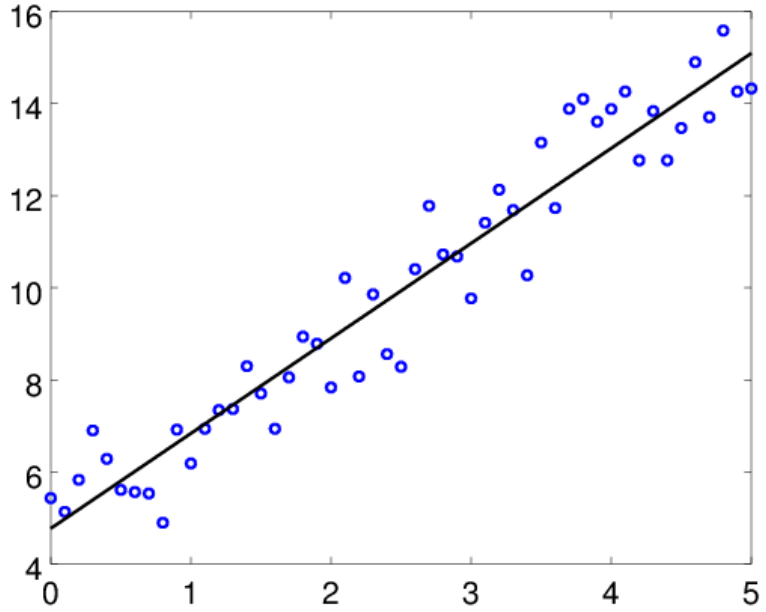
Метод наименьших квадратов: одномерный случай

$$\begin{cases} ax_1 + b = y_1 \\ \dots \\ ax_n + b = y_n \end{cases} \Leftrightarrow \begin{pmatrix} x_1 & 1 \\ \dots & \dots \\ x_3 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix} \Leftrightarrow X\beta = y$$

Т.к. матрица X – не квадратная, то записать $\beta = X^{-1}y$ нельзя

Но Octave/MATLAB решит эту систему уравнений, если написать `b=X\y`

```
>> x = (0:0.1:5)';
>> y = 2*x + 5 + randn(size(x));
>> X = [x ones(size(x))];
>> beta = X\y
beta =
   2.0653
   4.7701
>> close all;
>> plot(x,y,'bo','LineWidth',2);
>> hold on;
>> yfunc = @(x)beta(1)*x+beta(2);
>> plot(x,yfunc(x),'k-','LineWidth',2);
>> hold off;
>> print(gcf,'graph','-dpng','-r75');
```



Полезные команды

Команда	Значение
`cd имя_каталога`	Сменить текущий каталог
`clc`	Очистить консоль
`clear all`	Уничтожить все переменные (и ряд других структур вроде пользовательских типов данных, загруженных MEX-файлов и т.п.)
`close all`	Закрыть все окна с графиками
`dir (или ls)`	Вывести содержимое текущего каталога
`edit filename.m`	Отредактировать файл filename.m или создать его (если он отсутствует)
`figure`	Создать новое окно для вывода графиков
`help funcname`	Выдать справку по функции funcname
`hold on` `hold off`	Выводить новый график в существующее окно Выводить новый график в новое окно (старое закроется)
`pause`	Ждать нажатия ENTER пользователем
`pwd`	Узнать текущий каталог
`whos`	Вывести существующие переменные