

SCILAB: Библиотеки

Глава 7. Библиотека функций распределения

Содержание главы:

- Как построить создать последовательность случайных величин?
 - Как построить последовательность случайных величин с заданным распределением?
 - Как найти интегральную функцию распределения функции известных распределений?
 - Как найти интегральную функцию распределения функции нормального распределения (Гаусса)?
-

Как построить создать последовательность случайных величин?

С помощью команды **rand**.

Синтаксис

rand(m1,m2,... [,key])

rand(x [, key])

rand()

rand(key)

rand("seed" [,n])

rand("info")

Параметры

mi : целые числа

key : символьная переменная, принимающая значение "**uniform**" и "**normal**". Значение "**uniform**" соответствует равномерному распределению, а "**normal**" - Гауссовскому.

x : матрица. Во внимание принимается только ее размер.

Примеры использования.

1) **rand(m1,m2)** Образование случайной матрицы из m1 строк и m2 столбцов (m1 на m2).

2) **rand(m1,m2,...,mn)** Образование случайной матрицы размером m1 на m2,.. на mn.

3) **rand(A)** Образование случайной матрицы такого же размера, какого была матрица A.

Матрица **rand(A)** комплексная, если матрица A была комплексная.

s=1:4; a=rand(s) a = ! .9184708 .0437334 .4818509 .2639556 !

- 4) `rand()` без аргументов дает случайное скалярное число, случайным образом изменяющееся при следующем вызове.
- 5) `rand("normal")` или `rand("uniform")` позволяют задать распределение случайных чисел.
- 6) `rand("info")` возвращает значение переменной **key**.
- 7) `rand("seed" [,n])` позволяет ре-инициализировать генератор за счет изменения ядра.
-

Как построить последовательность случайных величин с заданным распределением?

С помощью команды **grand**, которая тоже является генератором случайных чисел, но более сложным.

В отличие от команды **rand**, команда **grand** может создавать последовательности случайных чисел из различных распределений.

Синтаксис

Y=grand(m, n, dist_type [,p1,...,pk])

Y=grand(X, dist_type [,p1,...,pk])

Y=grand(n, dist_type [,p1,...,pk])

S=grand(action [,q1,...,ql])

Параметры

m, n : целые числа, определяющие желаемый размер матрицы **Y**

X : матрица. Используется только данные о ее размере (задает **m** на **n**)

dist_type : символьная переменная, указывающая, распределение случайной величины какого типа мы создаем. Принимает определенные значения ('bin', 'nor', 'poi', ...)

p1, ..., pk : параметры (действительные или целые), необходимые для полного определения распределения типа **dist_type**

Y : результирующая матрица из случайных величин размера **m** на **n**

action : символьная переменная, указывающая на базу генератора (ов). Значение параметра **action**, равное 'setgen' меняет текущую базу генератора, 'getgen' извлекает имя текущего базового генератора, 'getsd' извлекает начальное число для генерации случайных чисел текущего генератора и т.д.

q1, ..., ql : параметры (обычно строка) необходимые для определения параметра **action**.

S: результат применения команды (обычно строка или столбец действительных векторов)

С помощью команды **grand** можно получить следующие распределения случайных чисел:

- Бета -распределение
- Биномиальное распределение (два варианта)
- Распределение hi-квадрат (два варианта)
- Экспоненциальное распределение
- F- дисперсионное распределение (Фишера) (два варианта)
- Гамма распределение
- Нормальное распределение Гаусса-Лапласа
- Мультивариативное Гауссово распределение
- Геометрическое распределение
- Распределение Маркова

- Полиномиальное распределение
- Распределение Пуассона
- Распределение, состоящее из перестановки из n-элементов
- Равномерное распределение (четыре варианта)

Подробный синтаксис этих команд смотрите с помощью команды **help grand**.

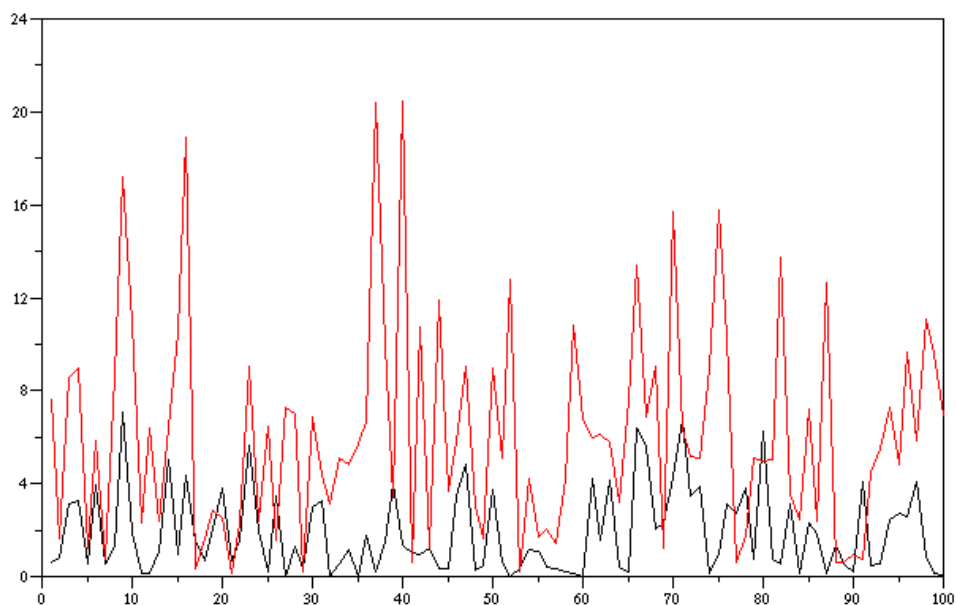
Кроме выбора различного закона распределения случайных величин, начиная с версии Scilab 2.7, можно выбирать между несколькими базовыми генераторами. В **help grand** приводятся параметры команды, отвечающие за разные варианты генератора случайных чисел.

Пример.

Создадим и построим экспоненциальное распределение случайных величин со средним арифметическим, равным Av .

```
m=100;
n=1;
Av=2;
// Av -величина среднего значения для построенной последовательности случайных
// величин
Y=grand(m,n,'exp',Av);
plot2d(1:100,Y); // Черная линия
Av2=5;
Y2=grand(m,n,'exp',Av2);
plot2d(1:100,Y2,5); // Красная линия
```

Результат:



```
//Проверка
m2=mean(Y2)
```

Результат:

```
m2 =  
6.1287607  
m=mean(Y)
```

Результат:

```
m =  
1.8847255
```

Мы видим, что "заказанное" среднее Δv и Δv_2 несколько отличаются от вычисленного по построенным случайным величинам. Если мы увеличим число элементов в последовательности, то результат будет точнее (ближе к значению Δv).

Для сравнения:

```
m=10000;  
n=1;  
 $\Delta v=2$ ;  
Y=grand(m,n,'exp', $\Delta v$ );  
m3=mean(Y);
```

Результат:

```
m3 =  
2.0290163
```

Как найти интегральную функцию распределения функции известных распределений ?

В Scilab существуют следующие команды определения интегральной функции для конкретных распределений:

- cdfnor** - для нормального распределения
- cdfbet** - для Beta распределения
- cdfbin** - для биномиального распределения
- cdfchi** - для хи-квадрат распределения
- cdfchn** - для нецентрального хи-квадрат распределения
- cdff** - для F-распределения (Фишера)
- cdffnc** - для нецентрального F-распределения (Фишера)
- cdfgam** - для гамма распределения
- cdfnbn** - для отрицательного биномиального распределения
- cdfnor** - - для нормального распределения (команда описана подробно)
- cdfpoi** - - для распределения Пуассона (Poisson)
- cdft** - для T-распределения Стьюдента (Student)

Об использовании команды **cdfnor** было описано подробно ниже. Остальные команды работают по аналогичной схеме. Подробно смотри help соответствующих команд.

Как найти интегральную функцию распределения функции нормального распределения (Гаусса)?

С помощью команды **cdfnor**.

Синтаксис

[P,Q]=cdfnor("PQ",X,Mean,Std) определяет интеграл от нормального распределения с заданными параметрами **X**, **Mean** и **Std**.

Нормальное распределение, определенное в Scilab, пропорционально величине

$\exp(-0.5 * (X - \text{Mean})^2 / \text{Std}^2)$,

где **Mean** и **Std** - параметры распределения.

Эта функция представляет собой колоколообразную кривую. Параметр **Mean** - точка максимума и одновременно центра симметрии, а **Std** - расстояние от этого центра до точки перегиба.

Замечание:

В справочнике Бронштейна и Семендяева определение нормального распределения (Гаусса) отличается от этого умножением на коэффициент $1/(\sqrt{2\pi})\text{Std}$. Команда **cdfnor** вычисляет **P**, равное значению интеграла от $-\infty$ до **X**, нормального распределения, заданного параметрами **Mean** и **Std**. Значение **Q=1-P**.

Вычисление любого из параметров распределения дает возможность получить значения всех остальных. Команда в другом варианте синтаксиса позволяет вычислить любой из недостающих параметров, зная этот интеграл и значения всех остальных параметров.

[X]=cdfnor("X",Mean,Std,P,Q)

[Mean]=cdfnor("Mean",Std,P,Q,X)

[Std]=cdfnor("Std",P,Q,X,Mean)

Параметры

P,Q,X,Mean,Std : действительные вектора одной и той же длины

P,Q (Q=1-P) : интеграл от $-\infty$ до **X** для нормального распределения. Область значений: (0, 1].

X : Верхний предел интегрирования. Область входных значений: ($-\infty$, $+\infty$)

Mean : Среднее нормальной плотности. Область входных значений: ($-\infty$, $+\infty$)

Std : стандартное среднее отклонение (корень из дисперсии). Область входных значений: (0, $+\infty$).

Пример.

x=0.5;

Mean=3;

Std=2;

[P,Q]=cdfnor("PQ",x,Mean,Std)

Результат:

Q =

.8943502

P = .1056498

Полученное **Q** равно интегралу от $-\infty$ до **x=0.5** кривой нормального распределения с параметрами **x=0.5**, **Mean=3** и **Std=2**.