# Введение в финансовую математику. Лабораторная работа № 1.

Zhdanov Pavel, C4111

## Вариант 12.

## Задание:

Реализуйте на R собственный генератор случайной выборки с плотностью

$$p(x) = \frac{(2x-1)}{4}$$

$$x[2;3]$$

1. Проверьте выполненение условия нормировки. Нормируйте соответствующим образом, если условие нарушеатся.
2. Постройте график функции распределения.
3. Сгенерируйте выборку из:
   - 10,
   - 100,
   - 1000 элементов.
4. Постройте гистограмму частот для выборки из предыдущего пункта и сравните ее с плотностью распределения
5. Напишите функцию, которая будет выводить следующую статистику по выборке: минимальное/максимальное значение, размах, среднее, выборочную дисперсию (с поправкой и без), среднеквадратическое отклонение, медиану, 25%-ную и 75%-ную квантили, коэффициент вариации, коэффициент ассиметрии, коэффициент эксцесса.

6. Примените функцию из предыдущего пункта к сгенерированной выборке.

## Решение:

Функция плотности распределения

```
task_p = function(xx) {
  pp = rep(0.0, length(xx));
  for (i in c(1:length(xx))) {
    if (xx[i] >= 2.0 && xx[i] <= 3.0){
      pp[i] = (2 * xx[i] - 1)/4
```

```
    }
  }
  return (pp)
}
```

Проверка того, что условие нормировки выполняется

```
print(integrate(task_p, lower=2, upper=3));

## 1 with absolute error < 1.1e-14
```

Значения функции распределения посчитанные через плотность распределения

```
fd_F = function(xx){
  FF = rep(0.0, length(xx));
  for (i in c(1:length(xx))){
    ans = integrate(task_p, min(xx), xx[i]);
    FF[i] = ans$value;
  }
  return (FF);
}
```

Подсчет функции распределения через явное задание

$$F(x) = \int_{-\infty}^{x} f(t)dt$$

$$F(x) = \int_{-\infty}^{x} \frac{(2t-1)}{4} dt$$

$$F(x) = \begin{cases} 0, x < 2, \\ \dfrac{x^2 - x - 2}{4}, 2 \leq x \leq 3, \\ 1, x > 3 \end{cases}$$

```
task_F = function(xx){
  FF = rep(0.0, length(xx));
  for (i in c(1:length(xx))) {
    if (xx[i] >= 2.0 && xx[i] <= 3.0){
      FF[i] = (xx[i]*xx[i] - xx[i] - 2.0)/4.0
    } else if (xx[i] > 3.0) {
      FF[i] = 1.0;
    }
  }
  return (FF)
}
```

Функция для генерации выборки с данным распределением с помощи функции равномерного распределения

$$F(x) = \frac{x^2 - x - 2}{4}$$

$$x^2 - x - (2 + 4 * F(x)) = 0$$

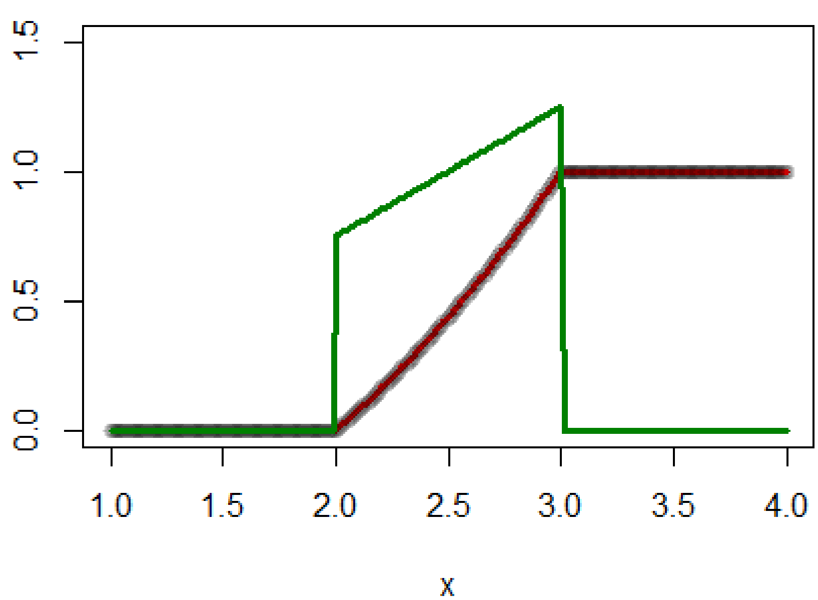
$$x_1 = \frac{1 + \sqrt{9 + 16 * F(x)}}{2}$$

$$x_2 = \frac{1 - \sqrt{9 + 16 * F(x)}}{2}$$

Подходит x1 поскольку переводит значения из [0:1] в [2;3]

```
gen_sample = function(n){
  unif = runif(n);
  ans = rep(0.0, n)
  for (i in c(1:n)) {
    ans[i] = (1 + sqrt(9 + 16 * unif[i]))/2
  }
  return (ans)
}

x = c(100:400) / 100.0;
plot(x, task_F(x), type = 'l', xlab = 'x', ylab = '', col = rgb(1.0,0.0,0.0,1.0)
, lwd = 3.0, xlim = c(1.0, 4.0), ylim = c(0.0, 1.5));
# Отрисовка функции плотности распределения
lines(x, fd_F(x), type = 'p', col=rgb(0.0,0.0,0.0,0.2), lwd=2.0);
# Отрисовка функции распределения
lines(x, task_p(x), type = 'l', col=rgb(0.0,0.5,0.0,1.0), lwd=3.0);
```




Генерация выборок из распределения

```
task_sample10 = gen_sample(10);
task_sample100 = gen_sample(100);
```

```
task_sample1000 = gen_sample(1000);
n_columns = 50;
```

График для выборки из 10 примеров

```
hist(task_sample10, breaks = 10, freq = F, col=rgb(0.0,0.0,1.0,0.33), xlim=c(1.0
, 4.0),ylim=c(-0.1, 3.1), xlab='X', ylab='', main='')
plot(ecdf(task_sample10), col=rgb(1.0,0.0,0.0,1.0), add=TRUE, lwd=2.0);
lines(x, task_p(x), type='l', col=rgb(0.0,0.5,0.0,1.0), lwd=3.0);
```

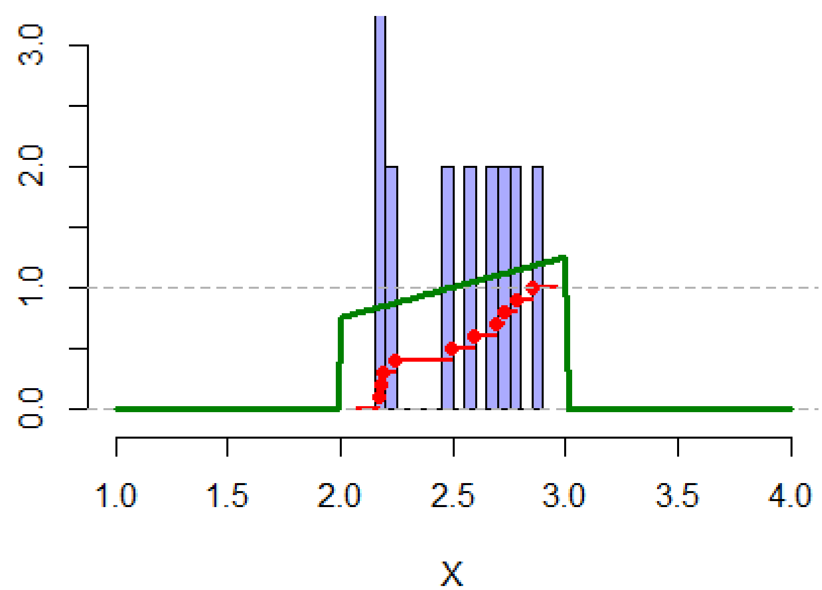



График для выборки из 100 примеров

```
hist(task_sample100, breaks=n_columns, freq=F, col=rgb(0.0,0.0,1.0,0.33), xlim=c
(1.0, 4.0), ylim=c(-0.1, 3.1), xlab='X', ylab='', main='');
plot(ecdf(task_sample100), col=rgb(1.0,0.0,0.0,1.0), add=TRUE, lwd=2.0);
lines(x, task_p(x), type='l', col=rgb(0.0,0.5,0.0,1.0), lwd=3.0);
```

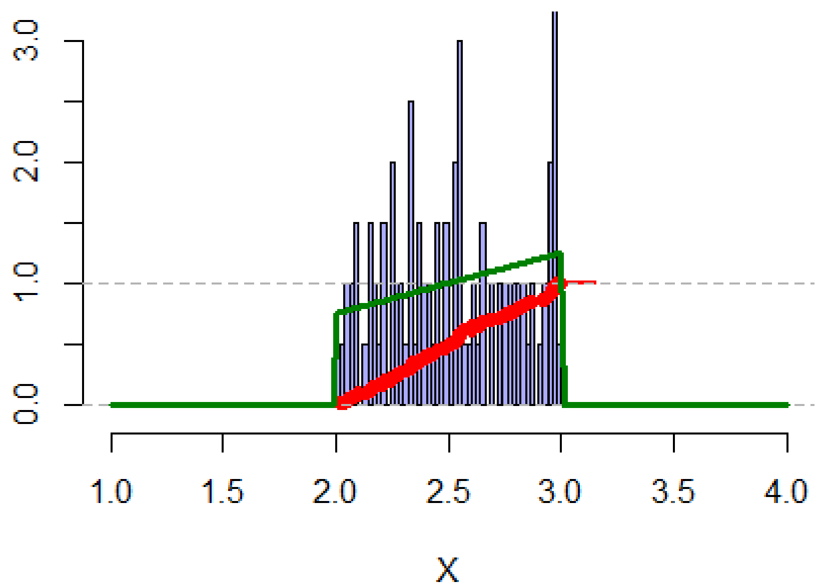
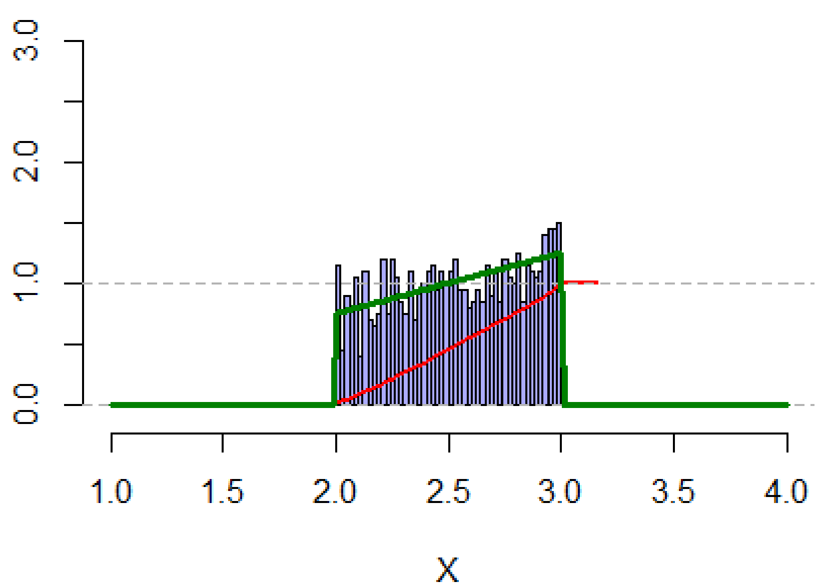



График для выборки из 1000 примеров

```
hist(task_sample1000, breaks=n_columns, freq=F, col=rgb(0.0,0.0,1.0,0.33), xlim=
c(1.0, 4.0), ylim=c(-0.1, 3.1), xlab='X', ylab='', main='');
plot(ecdf(task_sample1000), col=rgb(1.0,0.0,0.0,1.0), add=TRUE, lwd=2.0);
lines(x, task_p(x), type='l', col=rgb(0.0,0.5,0.0,1.0), lwd=3.0);
```



Функция, возвращающая статистику по выборке

```
library(moments);

## Warning: package 'moments' was built under R version 3.5.2

statistics = function(sample){
  min = min(sample);
  max = max(sample);
  width = max - min;
  mean = mean(sample);
  var = var(sample);
  not_corrected_var = sum((sample - mean) * (sample - mean))/length(sample);
  sd = sqrt(var);
  median = median(sample);
  q1 = quantile(sample, 0.25);
  q3 = quantile(sample, 0.75);
  var_coeff = sd/mean;
  skewness = skewness(sample);
  kurtosis = kurtosis(sample);
  return (round(c(min, max, width, mean, var, not_corrected_var, sd,
                  median, q1, q3, var_coeff, skewness, kurtosis), 2));
}
```

Получение конечного результата

```
dfs = data.frame(cbind(
  statistics(task_sample10),
  statistics(task_sample100),
  statistics(task_sample1000)), row.names =
    c('Min:', 'Max:', 'Width:', 'Mean:', 'var:', "var':",
      'Sd:', 'Median:', 'Q1:', 'Q3:', 'Var coef:', 'Skewness:',
      'Kurtosis:'));

dfs

##                X1   X2    X3
## Min:       2.17 2.01  2.00
## Max:       2.85 2.99  3.00
## Width:     0.68 0.98  1.00
## Mean:      2.49 2.52  2.54
## var:       0.07 0.08  0.08
## var':      0.07 0.08  0.08
## Sd:        0.27 0.29  0.29
## Median:    2.54 2.52  2.54
## Q1:        2.21 2.28  2.29
## Q3:        2.72 2.77  2.80
## Var coef:  0.11 0.11  0.11
## Skewness: -0.08 0.08 -0.11
## Kurtosis:  1.35 1.86  1.82
```

Выводы:

1. Была получена выборка из заданной плотности распределения на основе выборки из равномерного распределения с помощью метода обратного преобразования.
2. С увелечинем размера выборки, сгенерированная выборка лучше описывается заданной функцией плотности распределения.