Лабораторная работа №1

«Первичный графический анализ статистических данных»

Тема: Динамика изменения оценок по теории вероятности и математической статистики за два года.

Выполнил: Николай Окуньков, 18ПИ-2.

Цель работы: Создание статистического ряда и изучение графических методов первичного анализа статистических данных с использованием встроенных в базовую версию пакета R функций.

Теоретическая часть:

Одномерное непрерывное равномерное распределение – распределение случайной вещественной величины, принимающей значения, принадлежащие некоторому промежутку конечной длины, характеризующееся тем, что плотность вероятности на этом промежутке почти всюду постоянна.

Нормальное распределение (распределение Гаусса) – распределение вероятностей, которое в одномерном случае задаётся функцией плотности вероятности, совпадающей с функцией Гаусса:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

где ***σ^2*** – стандартное отклонение, ***σ*** — среднеквадратическое отклонение распределения, ***μ*** — математическое ожидание (среднее значение), медиана и мода распределения.

Пусть X1, ..., Xn – выборка из распределения вероятности, определенная на некотором вероятностном пространстве. Тогда её *выборочным средним* называется случайная величина:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i m_i$$

где ***x(i)*** – элемент выборки под номером ***i***, ***m(i)*** – частота выборки данного элемента, а **n** – общий объем выборки.

Относительная частота случайного события – отношение числа появления данного события к общему числу проведенных одинаковых испытаний, в каждом из которых могло появиться или не появиться данное событие.

Математическое ожидание – среднее (взвешенное по вероятностям возможных значений) значение случайной величины. Для непрерывных случайных величин находится по следующей формуле:

$$M[X] = \int_{-\infty}^{\infty} x f_X(x)\, dx$$

Дисперсия – мера разброса значений случайной величины относительно её математического ожидания. Находится по формуле:

$$D[X] = M[X^2] - (M[X])^2.$$

Среднеквадратическое отклонение – показатель рассеивания значений случайной величины относительно её математического ожидания. Находится по формуле:

$$\sigma = \sqrt{D[X]}$$

Гистограмма – наглядное представление функции плотности вероятности некоторой случайной величины, построенное по выборке.

Коробка с усами (диаграмма размаха) – график, использующийся в описательной статистике, компактно изображающий одномерное распределение вероятностей. Границами ящика служат первый и третий квартили (25-й и 75-й процентили соответственно), линия в середине ящика — медиана (50-й процентиль). Концы усов — края статистически значимой выборки (без выбросов), минимальное и максимальное наблюдаемые значения данных по выборке (в этом случае выбросы отсутствуют).

Квантиль – значение, которое заданная случайная величина не превышает с фиксированной вероятностью. Если вероятность задана в процентах, то квантиль называется процентилем или перцентилем.

- 0,25 – квантиль называется первым или нижним квартилем;
- 0,50 – квантиль называется медианой или вторым квартилем;
- 0,75 – квантиль называется третьим или верхним квартилем.

Процентиль – это процентная доля элементов из выборки стандартизации, первичный результат которых ниже данного первичного показателя.

Ход работы:

Задание 1:

1. Создать два массива данных оценок по стобальной системе, объемом 51 и 62 элемента. Для этого использовать случайный набор чисел из нормального распределения с математическим ожиданием 70 и 65 и среднеквадратичным отклонением 30 и 40, соответственно. Начальный отсчет – Ваш номер в списке группы. Все числа >100 и все отрицательные числа заменить на специальную переменную NA.

Для решения задачи была использована функция **R rnorm()** и собственная функция **enterNA()**:

rnorm(n, mean= , sd=) – функция для создания выборки, используя нормальное распределение;

(Параметры: **n** – количество элементов выборки; **mean** –математическое ожидание; **sd** – среднеквадратическое отклонение.)

enterNA(arr) – нужна для замены всех отрицательных значений и значений, превышающих 100 на NA.

(Параметры: **arr** — выборка.)

```
enterNA <- function(array){
  for (i in 1:length(array)) {
    if ((array[i]<0 || 100<array[i])&&!is.na(array[i])) {
      array[i] <- NA
    }
  }
  return(array)
}
```

Код:

```r
N1    <- 51  # кол-во элементов в выборке
mean1 <- 70  # мат. ожидание
sd1   <- 30  # среднеквадратичное отклонение

set.seed(13)

# создание выборки и замена значений, который <0 или >100, на NA
sample1 <- rnorm(N1, mean = mean1, sd = sd1); sample1
sample1 <- enterNA(array = sample1); sample1

N2    <- 62  # кол-во элементов в выборке
mean2 <- 65  # мат. ожидание
sd2   <- 40  # среднеквадратичное отклонение

# создание выборки и замена значений, который <0 или >100, на NA
sample2 <- rnorm(N1, mean = mean1, sd = sd1); sample2
sample2 <- enterNA(array = sample2); sample2
```

Результат:

```
> N1    <- 51  # кол-во элементов в выборке
> mean1 <- 70  # мат. ожидание
> sd1   <- 30  # среднеквадратичное отклонение
> set.seed(13)
> # создание выборки и замена значений, который <0 или >100, на NA
> sample1 <- rnorm(N1, mean = mean1, sd = sd1); sample1
 [1]  86.629808  61.591842 123.254901  75.619604 104.275784  82.465784
 [7] 106.885197  77.100390  59.038517 103.154328  37.192181  83.856127
[13]  29.170464  14.319185  56.804338  64.181593 111.892945  73.019897
[19]  66.566836  91.066757  77.876280 125.084899  80.722073  38.637696
[25]  88.605524  74.480636  26.220494   9.188686  38.291267  48.155689
[31]  69.753680  95.433921  58.495255  54.204655  61.803221  51.827752
[37]  60.013981  62.753874  44.116738  44.590877  73.010210 117.701006
[43]  86.994846 118.434385  55.940495  48.216958  39.299830  11.865534
[49]  78.314419 112.250610  78.193876
> sample1 <- enterNA(array = sample1); sample1
 [1] 86.629808 61.591842        NA 75.619604        NA 82.465784        NA
 [8] 77.100390 59.038517        NA 37.192181 83.856127 29.170464 14.319185
[15] 56.804338 64.181593        NA 73.019897 66.566836 91.066757 77.876280
[22]        NA 80.722073 38.637696 88.605524 74.480636 26.220494  9.188686
[29] 38.291267 48.155689 69.753680 95.433921 58.495255 54.204655 61.803221
[36] 51.827752 60.013981 62.753874 44.116738 44.590877 73.010210        NA
[43] 86.994846        NA 55.940495 48.216958 39.299830 11.865534 78.314419
[50]        NA 78.193876
> N2    <- 62  # кол-во элементов в выборке
> mean2 <- 65  # мат. ожидание
> sd2   <- 40  # среднеквадратичное отклонение
> # создание выборки и замена значений, который <0 или >100, на NA
> sample2 <- rnorm(N1, mean = mean1, sd = sd1); sample2
 [1]  92.66575  59.52945  53.61428  77.03086  61.06515  44.78572  94.79531
 [8] 114.51074  90.99027  32.15278  78.94815  65.56579  43.33233 100.39198
[15]  42.38425  52.78316 104.51096 104.31474  62.81672  37.39594  68.15659
[22]  54.49908  12.76979  73.21469  34.67874 122.36281  58.03904  83.27318
[29]  83.50838  67.71814  78.92540  34.16936  10.09374 111.65539  67.52549
[36]  81.77543  37.51691 118.06361 100.12207  81.39687  53.03484  33.58666
[43]  29.07095  27.51601  62.33266  33.23722  76.41503  72.01671  95.69905
[50]  58.92482  88.34492
> sample2 <- enterNA(array = sample2); sample2
 [1] 92.66575 59.52945 53.61428 77.03086 61.06515 44.78572 94.79531       NA
 [9] 90.99027 32.15278 78.94815 65.56579 43.33233       NA 42.38425 52.78316
[17]       NA       NA 62.81672 37.39594 68.15659 54.49908 12.76979 73.21469
[25] 34.67874       NA 58.03904 83.27318 83.50838 67.71814 78.92540 34.16936
[33] 10.09374       NA 67.52549 81.77543 37.51691       NA       NA 81.39687
[41] 53.03484 33.58666 29.07095 27.51601 62.33266 33.23722 76.41503 72.01671
[49] 95.69905 58.92482 88.34492
```

Задание 2:

2. Разбить полученные данные на категории (использовать R-функции cut(), table()):

а) по пятибальной системе:	б) по европейской системе:
2 – баллы от 0 до 50;	F - баллы от 0 до 30;
3 – баллы от 51 до 68;	FX - баллы от 31 до 50;
4 – баллы от 69 до 81;	E - баллы от 51 до 60;
5 – баллы от 86 до 100.	D - баллы от 61 до 68;
	C - баллы от 69 до 85;
	B - баллы от 86 до95;
	A - баллы от 96 до 100.

Для решения задачи были использованы собственные функции **to5()** и **toEU()**:
to5(array) – переводит оценки из 100-бальной системы в 5-бальную посредством разбиения выборки с помощью функции **cut()**, которая получает на вход вектор и делит их на равные или заранее заданные интервалы;

(Параметры: **array** — выборка.)

```
# функция для перевода в пятибальную систему оцениванию
to5 <- function(array){
  array <- cut(x = array, breaks = c(0, 50, 68, 81, 100))
  array.f <- factor(array)
  levels(array.f) <- c("2", "3", "4", "5")
  array.o <- ordered(array.f, labels = c("2", "3", "4", "5"))
  return(array.o)
}
```

toEU(array) – переводит оценки из 100-бальной системы в 7-бальную посредством разбиения выборки с помощью функции **cut()**, которая получает на вход вектор и делит их на равные или заранее заданные интервалы.

(Параметры: **array** — выборка)

```
# функция для перевода в европейскую систему оценивания
toEU <- function(array){
  array <- cut(x = array, breaks = c(0, 30, 50, 60, 68, 85, 95, 100))
  array.f <- factor(array)
  levels(array.f) <- c("F", "FX", "E", "D", "C", "B", "A")
  array.o <- ordered(array.f, labels = c("F", "FX", "E", "D", "C", "B", "A"))
  return(array.o)
}
```

Для проверки корректности работы этих функций я использовал функцию R **table()**:
table(x) – возвращает таблицу с частотами встречаемости каждого значения **x**.

(Параметры: **x** — набор значений.)

Код:

```
# переведем выборки с пятибальные и европейские системы счисления
marks1.5point <- to5(array = sample1); table(marks1.5point)
marks1.EUpoint <- toEU(array = sample1); table(marks1.EUpoint)

marks2.5point <- to5(array = sample2); table(marks2.5point)
marks2.EUpoint <- toEU(array = sample2); table(marks2.EUpoint)
```

Результат:

```
> # переведем выборки с пятибальные и европейские системы счисления
> marks1.5point <- to5(array = sample1); table(marks1.5point)
marks1.5point
 2  3  4  5
13 12 10  7
> marks1.EUpoint <- toEU(array = sample1); table(marks1.EUpoint)
marks1.EUpoint
 F FX  E  D  C  B  A
 5  8  6  6 12  4  1
> marks2.5point <- to5(array = sample2); table(marks2.5point)
marks2.5point
 2  3  4  5
14 13  7  9
> marks2.EUpoint <- toEU(array = sample2); table(marks2.EUpoint)
marks2.EUpoint
 F FX  E  D  C  B  A
 4 10  7  6 11  4  1
```

Задание 3:

3. Создать таблицу относительных частот в каждой категории.

Для решения задачи были использованы собственные функция **data.frame()**:

data.frame() - создает таблицу данных из поименованных или непоименованных аргументов;

Код:

```
# создадим таблицу относительных частот для оценок в пятибальной системе счисления
t1 <- data.frame(row.names = levels(marks1.5point), table(marks1.5point)); t1
t2 <- data.frame(row.names = levels(marks2.5point), table(marks1.5point)); t2
table1 <- data.frame(row.names = levels(marks2.5point), marks1  = prop.table( t1[, -1]), marks2  = prop.table( t2[, -1])); table1

# создадим таблицу относительных частот для оценок в европейской системе счисления
t1 <- data.frame(row.names = levels(marks1.EUpoint), table(marks1.EUpoint)); t1
t2 <- data.frame(row.names = levels(marks2.EUpoint), table(marks1.EUpoint)); t2
table2 <- data.frame(row.names = levels(marks2.EUpoint), marks1  = prop.table( t1[, -1]), marks2  = prop.table( t2[, -1])); table2
```

Результат:

```
> # создадим таблицу относительных частот для оценок в пятибальной системе счисления
> t1 <- data.frame(row.names = levels(marks1.5point), table(marks1.5point)); t1
  marks1.5point Freq
2             2   13
3             3   12
4             4   10
5             5    7
> t2 <- data.frame(row.names = levels(marks2.5point), table(marks1.5point)); t2
  marks1.5point Freq
2             2   13
3             3   12
4             4   10
5             5    7
> table1 <- data.frame(row.names = levels(marks2.5point), marks1  = prop.table( t1[, -1]), marks2  = prop.table( t2[, -1])); table1
     marks1    marks2
2 0.3095238 0.3095238
3 0.2857143 0.2857143
4 0.2380952 0.2380952
5 0.1666667 0.1666667
> # создадим таблицу относительных частот для оценок в европейской системе счисления
> t1 <- data.frame(row.names = levels(marks1.EUpoint), table(marks1.EUpoint)); t1
   marks1.EUpoint Freq
F               F    5
FX             FX    8
E               E    6
D               D    6
C               C   12
B               B    4
A               A    1
> t2 <- data.frame(row.names = levels(marks2.EUpoint), table(marks1.EUpoint)); t2
   marks1.EUpoint Freq
F               F    5
FX             FX    8
E               E    6
D               D    6
C               C   12
B               B    4
A               A    1
> table2 <- data.frame(row.names = levels(marks2.EUpoint), marks1  = prop.table( t1[, -1]), marks2  = prop.table( t2[, -1])); table2
       marks1     marks2
F  0.11904762 0.11904762
FX 0.19047619 0.19047619
E  0.14285714 0.14285714
D  0.14285714 0.14285714
C  0.28571429 0.28571429
B  0.09523810 0.09523810
A  0.02380952 0.02380952
```

	marks1	marks2
2	0.3095238	0.3095238
3	0.2857143	0.2857143
4	0.2380952	0.2380952
5	0.1666667	0.1666667

	marks1	marks2
F	0.11904762	0.11904762
FX	0.19047619	0.19047619
E	0.14285714	0.14285714
D	0.14285714	0.14285714
C	0.28571429	0.28571429
B	0.09523810	0.09523810
A	0.02380952	0.02380952

Задание 4:

4. Построить гистограммы и график функции плотности на одном рисунке для каждой выборки, взяв соответствующие интервалы для построения гистограмм.

Для решения задачи были использованы собственные функции **par()**, **hist()**, **lines()**, **density()**:

par() – функция для разделения пространства, на котором будут графики, на некое определенное количество маленьких пространств, для того, чтобы нарисовать сразу несколько графиков на одном экране;

hist(x, breaks =) – функция для создания гистограмм частот значений переменной **x** (аргумент **breaks =** можно использовать, чтобы изменить принятое по умолчанию количество столбцов);

(Параметры: **x** — переменная; **breaks** — количество столбцов.)

lines(x, col = , lwb =) – функция для создания линий на графиках;

(Параметры: **x** — значения; **col** — цвет; **lwd** — толщина линии.)

density() – функция для нахождения ядерных плотностей вероятностей (ядерная плотность вероятности — оценка случайной величины);

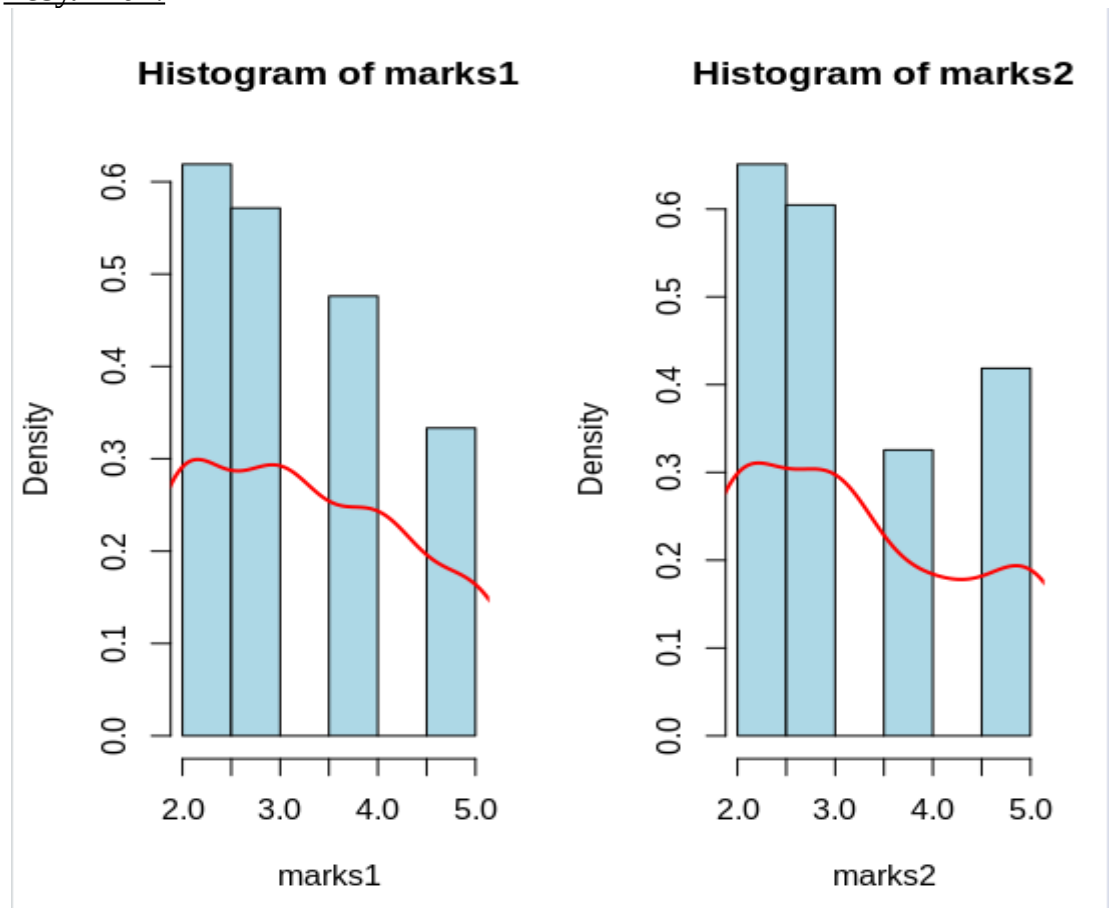
Код:

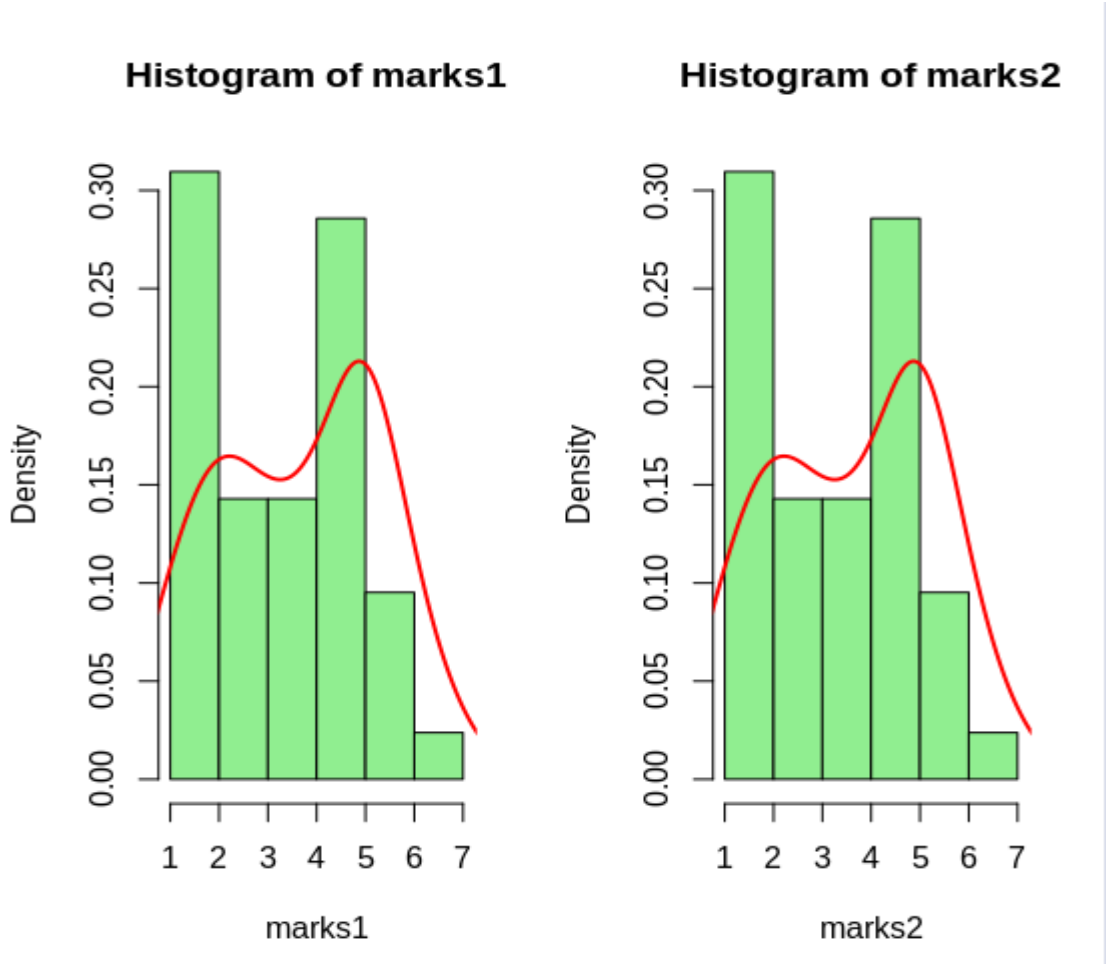
```
# разделение окна на 2 части для построения сразу двух гистограмм
par(mfcol=c(1, 2))

# построение гистограмм для пятибальной системы оценивания
marks1 <- as.numeric(as.character(na.omit(marks1.5point)))
hist(marks1, col = "lightblue", freq = FALSE)
lines(density(marks1, na.rm = TRUE), col = "red", lwd = 2)
marks2 <- as.numeric(as.character(na.omit(marks2.5point)))
hist(marks2, col = "lightblue", freq = FALSE)
lines(density(marks2, na.rm = TRUE), col = "red", lwd = 2)

# построение гистограмм для пятибальной системы оценивания
marks1 <- as.numeric(na.omit(marks1.EUpoint)); marks1
hist(marks1, col = "lightgreen", freq = FALSE)
lines(density(marks1, na.rm = TRUE), col = "red", lwd = 2)
marks2 <- as.numeric(na.omit(marks1.EUpoint)); marks2
hist(marks2, col = "lightgreen", freq = FALSE)
lines(density(marks2, na.rm = TRUE), col = "red", lwd = 2)
```

Результат:


Histogram of marks1
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2.0
3.0
4.0
5.0
marks1
Histogram of marks2
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2.0
3.0
4.0
5.0
marks2


Histogram of marks1
Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
1
2
3
4
5
6
7
marks1
Histogram of marks2
Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
1
2
3
4
5
6
7
marks2

Задание 5:

5. На одном рисунке построить «ящики с усами» для всех выборок.

Для решения задачи были использованы функции **boxplot()** и **dev.off()**:
dev.off() – функция для очистки окна вывода.
boxplot(x, main = , ylab =) – функция для построения диаграмм размахов («коробок с усами»).
(Параметры: **x** — выборка; **main** — название графика; **ylab** — название осиOY.)
Код:

```
dev.off()
# разделение окна на 4 части для построения сразу четырех коробок с усами
par(mfcol=c(1, 4))

boxplot(marks1.5point, main="marks1 5-point sys ", ylab="mark")
boxplot(marks1.EUpoint, main="marks1 EU-point sys ", ylab="mark")
boxplot(marks2.5point, main="marks2 5-point sys ", ylab="mark")
boxplot(marks2.EUpoint, main="marks2 EU-point sys ", ylab="mark")
```

Результат: