

Федеральное государственное бюджетное образовательное  
учреждение  
высшего образования  
«Саратовский государственный технический университет  
имени Гагарина Ю.А.»

Кафедра «Прикладные информационные технологии»

**ОТЧЕТ**  
**по лабораторной работе номер 6**

Студента гр. 62-ПИНФ21:  
Нефедова Данила Вадимовича  
Проверил доцент кафедры ПИТ:  
Бровко Александр Валерьевич

Саратов 2019

# 1

```
#
# 1. С использованием встроенной справки help изучить функцию seq().
#

help("seq")

#
# 2. Объяснить результаты, получаемые при выполнении следующих команд:
#   Записать результаты, получаемые с использованием предыдущих команд,
#   в векторы x1, ..., x7.
#

# Сгенерирует последовательность из 11 чисел от 0 до 1, расположенных на
# одинаковом расстоянии друг от друга.
x1 <- seq(0, 1, length.out = 11)

# Сгенерирует последовательность целых чисел 1, 2, ..., length(along.width),
# где along.width - это stats:rnorm(20), длина которого равна 20. Следовательно,
# последовательность будет от 1 до 20.
#
# Более детальное объяснение: https://stackoverflow.com/q/39750718/8086115.
x2 <- seq(stats::rnorm(20))

# Генерирует последовательность чисел от 1, 1 + by, ..., до числа, не
# большего, чем верхняя граница (в данном случае - 9).
x3 <- seq(1, 9, by = 2)

# Генерирует последовательность чисел от 1 до 9 с шагом PI. При таком шаге
# конец последовательности не превысит 9 и будет равен примерно 7.283185.
x4 <- seq(1, 9, by = pi)

# Сгенерирует последовательности чисел 1, 4.
x5 <- seq(1, 6, by = 3)

# Сгенерирует последовательность чисел от 1.575 до 5.125 с шагом 0.05.
x6 <- seq(1.575, 5.125, by = 0.05)

# Сгенерирует последовательность от 1 до 17, тоже самое, что 1:17.
x7 <- seq(17)

#
# 3. Сохранить рабочее пространство на диск:
#

save.image(file = "workspace.RData")
```

```

#
# 4. Очистить рабочее пространство (удалить все именованные переменные).
#

rm(list = ls())

#
# 5. Загрузить рабочее пространство из файла.
#

load("workspace.RData")

```

## 2

```

#
# 1. С использованием встроенной справки help,
#    изучить функции matrix, solve и операцию %*%.
#

help("matrix")
help("solve")
help("%*%")

#
# 2. Сгенерировать матрицу A размером 10x10 элементов, заполненную
#    случайными числами (использовать гауссово или нормальное распределение).
#

A <- matrix(rnorm(10 * 10, mean = 0, sd = 1), 10, 10)

#
# 3. Создать вектор X, заполненный числами от 1 до 10 (всего 10 элементов).
#

X <- 1:10

#
# 4. Выполнить операции A * X и A %*% X. Объяснить получаемые
#    результаты (что именно происходит в каждом из этих случаев).
#

# Выполняется по-элементное умножение столбцов матрицы A на вектор X.
print(A * X)

# Выполняется матричное умножение матрицы A на вектор X.
#
# Сам по себе вектор X не является ни вектором-столбцом, ни вектором-
# строкой. Оператор %*% сам приводит этот вектор к размерности,
# необходимой для умножения.
print(A %*% X)

```

```
#  
# 5. Создать вектор B, выполнив матричное умножение A %*% X.  
#
```

```
B <- A %*% X
```

```
#  
# 6. Решить систему линейных уравнений, используя матрицу  
# A и вектор B в качестве вектора правых частей. Сравнить  
# полученные результаты с вектором X.  
#
```

```
# В качестве решения этой СЛАУ мы должны получить исходный  
# вектор X. Т. е. числа от 1 до 10.  
solve(A, B)
```

### 3

```
#  
# 1. Загрузите данные из файла в датафрейм HousePrice, используя  
# функцию read.table.  
#
```

```
HousePrice <- read.table("houses.data")
```

```
#  
# 2. Загрузите данные из файла в датафрейм HousePrice1,  
# используя функцию read.table. Загрузку следует выполнить  
# так, чтобы первая строка воспринималась как заголовки столбцов,  
# а не как текстовые данные.  
#
```

```
HousePrice1 <- read.table("houses1.data", header = TRUE)
```

```
#  
# 3. Выполните команду HP <- edit(HousePrice), записав в  
# переменную HP отредактированные данные (измените несколько  
# значений в открывшемся окне редактора).  
#
```

```
HP <- edit(HousePrice)
```

```
#  
# 4. Сохраните фрейм HP в файл "myframe.txt", используя функцию  
# write.table. Сохранение следует выполнить так, чтобы содержимое  
# файла имело тот же формат, что и файл "houses1.data", то есть  
# без номеров строк и без кавычек у текстовых данных.  
#
```

```
write.table(HP, file = "myframe.txt", row.names = FALSE, quote = FALSE)
```

```
#
# 5. Выведите на консоль содержимое столбца Rooms у датафреймов HousePrice,
#   HousePrice1, HP. Подсказка: используйте функции attach() и detach().
#

print(HousePrice$Rooms)
print(HousePrice1$Rooms)
print(HP$Rooms)
```

## 4

```
#
# 1. Создайте вектор T из 100 элементов, заполненный случайными
#   значениями (использовать нормальное или гауссово распределение).
#

T <- rnorm(100, mean = 0, sd = 1)

#
# 2. Примените следующие команды:
#
#   plot(T)
#   plot(sort(T))
#   hist(T)
#   plot(density(T))
#   rug(T)
#

plot(T)
plot(sort(T))
hist(T)
plot(density(T))
rug(T)

#
# 3. Выведите на консоль содержимое встроенного набора данных cars.
#

print(cars)

#
# 4. Примените команду plot(cars), объясните полученные результаты.
#

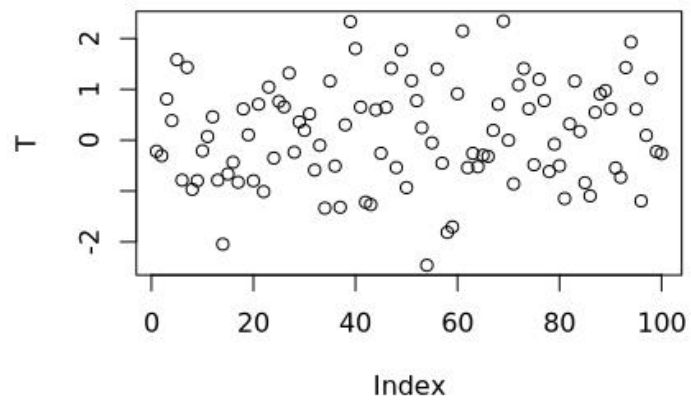
plot(cars)

#
# 5. Постройте графики функций:
#
#   sin – в диапазоне от 0 до 2π
```

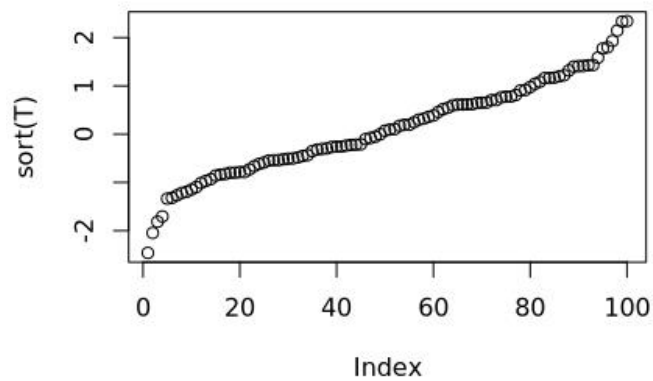
```
# cos - в диапазоне от 0 до 2π  
# exp - в диапазоне от 0 до 5  
# log - в диапазоне от 0 до 10  
#
```

```
plot(sin, 0, 2 * pi)  
plot(cos, 0, 2 * pi)  
plot(exp, 0, 5)  
plot(log, 0, 10)
```

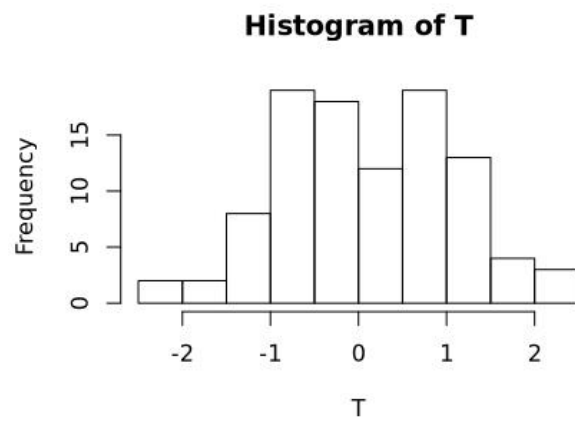
**plot(t)**



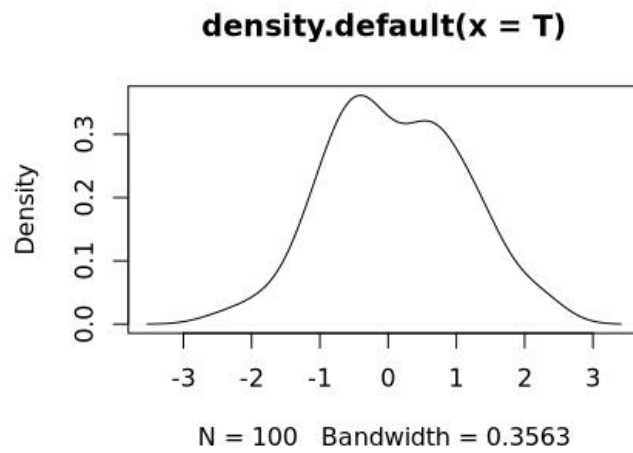
**plot(sort(t))**



`hist(t)`



`plot(density(t))`



`rug(t)`

