Домашнее задание № 1

Potapova Marina

2024-10-19

Работа с данными

По адресу http://people.math.umass.edu/~anna/Stat597AFall2016/rnf6080.dat (http://people.math.umass.edu/~anna/Stat597AFall2016/rnf6080.dat) можно получить набор данных об осадках в Канаде с 1960 по 1980 годы. Необходимо загрузить эти данные при помощи read.table. Воспользуйтесь справкой, чтобы изучить аргументы, которые принимает функция.

1. Загрузите данные в датафрейм, который назовите data.df.

```
url <- "http://people.math.umass.edu/~anna/Stat597AFall2016/rnf6080.dat"
data.df <- read.table(url, header = FALSE)
```

2. Сколько строк и столбцов в data.df? Если получилось не 5070 наблюдений 27 переменных, то проверяйте аргументы.

```
dim(data.df)
```

```
## [1] 5070   27
```

3. Получите имена колонок из data.df.

```
colnames(data.df)
```

```
##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11" "V12"
## [13] "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23" "V24"
## [25] "V25" "V26" "V27"
```

4. Найдите значение из 5 строки седьмого столбца.

```
data.df[5, 7]
```

```
## [1] 0
```

5. Напечатайте целиком 2 строку из data.df

```
data.df[2, ]
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21
## 2 60  4  2  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0   0
##   V22 V23 V24 V25 V26 V27
## 2   0   0   0   0   0   0
```

6.1 Объясните, что делает следующая строка кода names(data.df) <- c(“year”, “month”, “day”, seq(0,23)).

```
names(data.df) <- c("year", "month", "day", seq(0, 23))
```

Первые три колонки — “year” (год), “month” (месяц) и “day” (день) показывают дату. Остальные 24 колонки — от 0 до 23 показывают осадки за каждый час дня.

6.2 Воспользуйтесь функциями head и tail, чтобы просмотреть таблицу. Что представляют собой последние 24 колонки?

```
# Первые строки
head(data.df)
```

```
##   year month day 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## 1   60     4   1 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 2   60     4   2 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 3   60     4   3 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 4   60     4   4 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5   60     4   5 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 6   60     4   6 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
# Последние строки
tail(data.df)
```

```
##      year month day 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
## 5065   80    11  25 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5066   80    11  26 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5067   80    11  27 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5068   80    11  28 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5069   80    11  29 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5070   80    11  30 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0
##      23
## 5065  0
## 5066  0
## 5067  0
## 5068  0
## 5069  0
## 5070  0
```

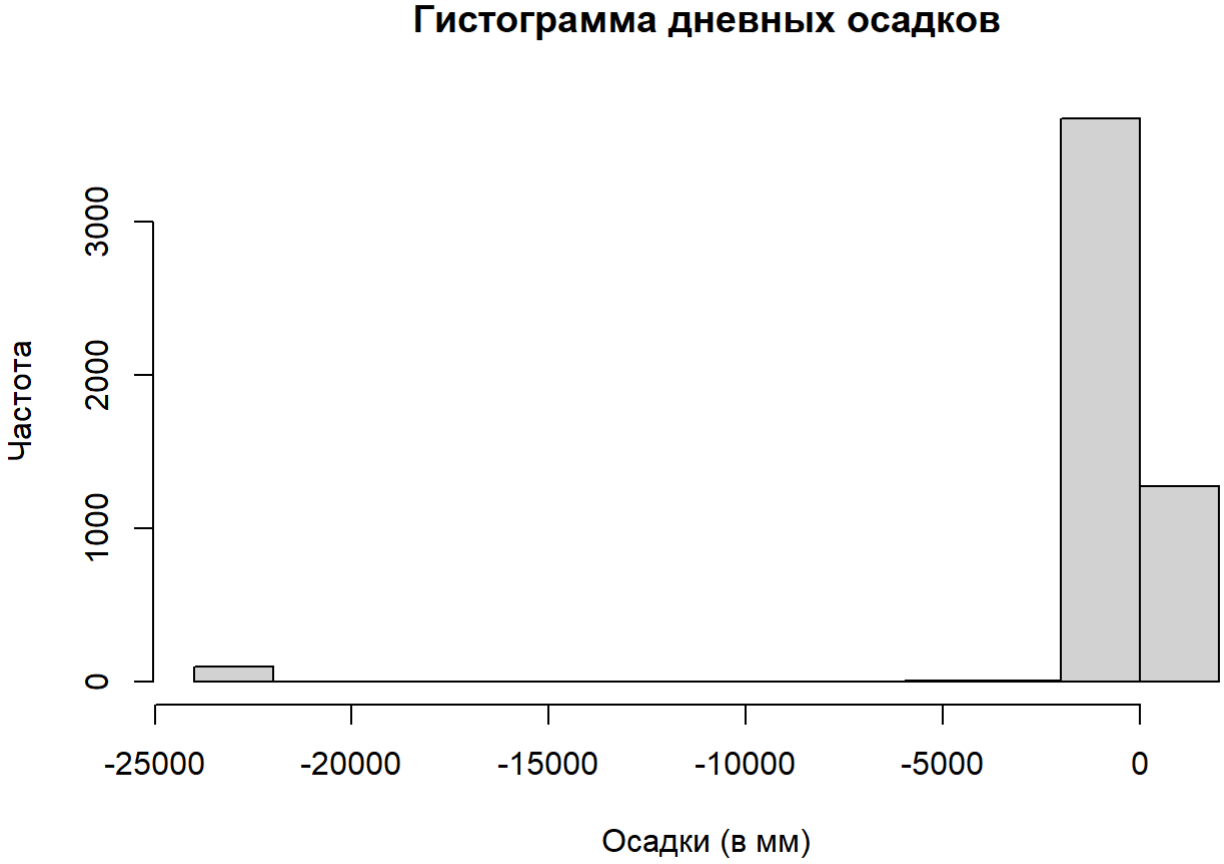
Первый вывод с head(data.df) покажет нам первые строки датафрейма, которые помогут понять структуру данных и какие значения записаны в первых колонках (год, месяц, день). Последние 24 колонки, о которых мы говорим в выводе tail(data.df), содержат данные об осадках по часам в течение дня.

7.1 Добавьте новую колонку с названием daily, в которую запишите сумму крайних правых 24 колонок.

```
data.df$daily <- rowSums(data.df[, 4:27])
```

7.2 Постройте гистограмму по этой колонке. Какие выводы можно сделать?

```
hist(data.df$daily, main = "Гистограмма дневных осадков", xlab = "Осадки (в мм)", ylab = "Частота")
```



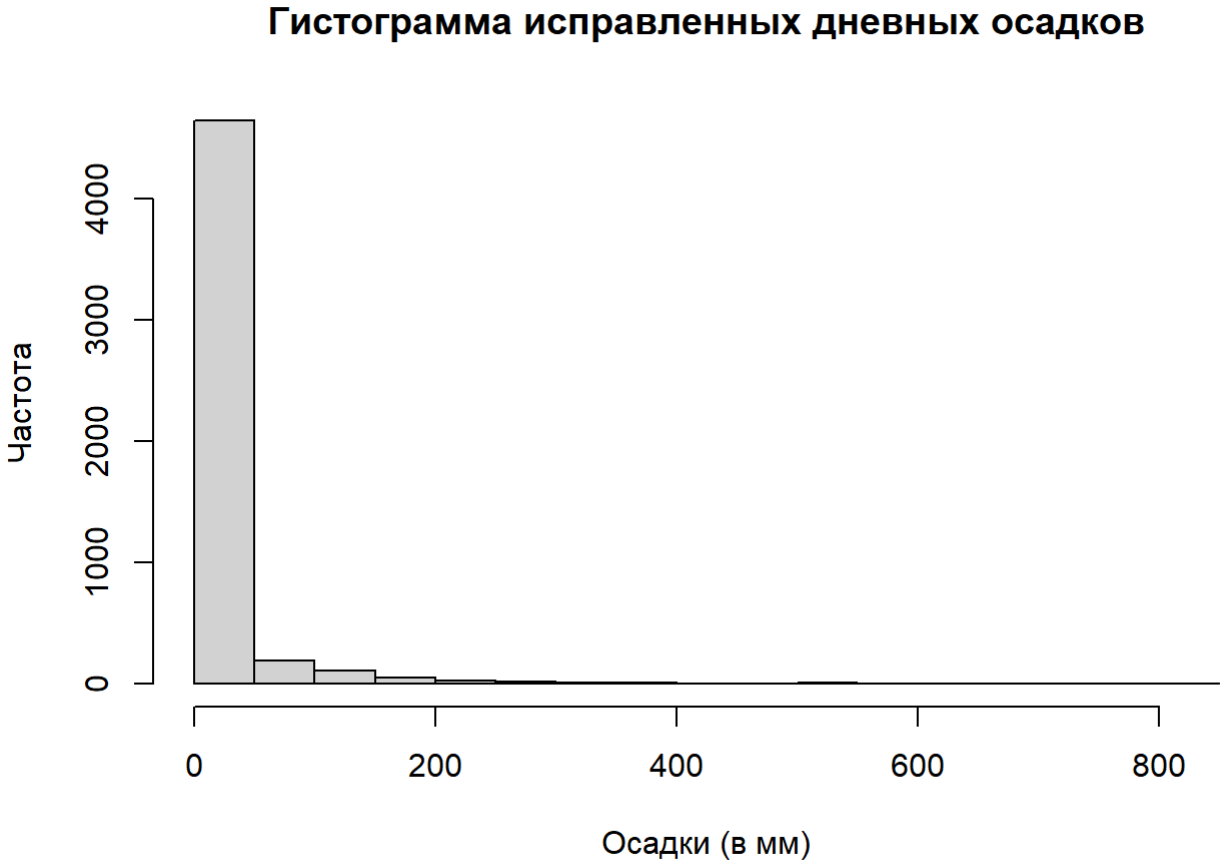
На гистограмме можно увидеть распределение дневных осадков. Если большая часть значений сосредоточена в низком диапазоне, это может указывать на то, что дожди в основном небольшие.

8.1 Создайте новый датафрейм fixed.df в котром исправьте замеченную ошибку.

```
fixed.df <- data.df
fixed.df$daily[fixed.df$daily < 0] <- 0
```

8.2 Постройте новую гистограмму, поясните почему она более корректна.

```
hist(fixed.df$daily, main = "Гистограмма исправленных дневных осадков", xlab = "Осадки (в мм)", ylab =
"Частота")
```



Синтаксис и типизирование

1. Для каждой строки кода поясните полученный результат, либо объясните почему она ошибочна.

```
v <- c("4", "8", "15", "16", "23", "42")
max(v) # вернет "42", т.к. сравнение строк
```

```
## [1] "8"
```

```
sort(v) # сортирует как строки
```

```
## [1] "15" "16" "23" "4"  "42" "8"
```

```
# sum(v) выдаст ошибку, потому что текст нельзя складывать как числа
```

2. Для следующих наборов команд поясните полученный результат, либо объясните почему они ошибочна.

```
v2 <- c("5",7,12)
# v2[2] + 2[3] ошибка, потому что мы пытаемся обратиться к элементу, который не существует

df3 <- data.frame(z1="5",z2=7,z3=12)
df3[1,2] + df3[1,3] # сложение: 7 + 12 = 19
```

```
## [1] 19
```

```
l4 <- list(z1="6", z2=42, z3="49", z4=126)
l4[[2]] + l4[[4]] # вернет 168, сумма чисел
```

```
## [1] 168
```

```
# l4[2] + l4[4] ошибка, потому что мы складываем списки, а не числа
```

Работа с функциями и операторами

1. Оператор двоеточие создаёт последовательность целых чисел по порядку. Этот оператор — частный случай функции seq(), которую вы использовали раньше. Изучите эту функцию, вызвав команду ?seq. Испольуя полученные знания выведите на экран:

i. Числа от 1 до 10000 с инкрементом 372.

```
seq(1, 10000, by = 372)
```

```
##  [1]    1  373  745 1117 1489 1861 2233 2605 2977 3349 3721 4093 4465 4837 5209
## [16] 5581 5953 6325 6697 7069 7441 7813 8185 8557 8929 9301 9673
```

ii. Числа от 1 до 10000 длиной 50.

```
seq(1, 10000, by = 372)
```

```
##  [1]    1  373  745 1117 1489 1861 2233 2605 2977 3349 3721 4093 4465 4837 5209
## [16] 5581 5953 6325 6697 7069 7441 7813 8185 8557 8929 9301 9673
```

seq(1, 10000, length.out = 50)

2. Функция rep() повторяет переданный вектор указанное число раз. Объясните разницу между rep(1:5,times=3) и rep(1:5, each=3).

```
rep(1:5, times=3) # повторяет всю последовательность (1, 2, 3, 4, 5) три раза
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

```
rep(1:5, each=3) # повторяет каждое число три раза
```

```
##  [1] 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5
```