

Домашнее задание №1

Бабикова Евгения
17 10 2017

Работа с данными

В `data.df` содержится 5070 объектов и 27 переменных.

Имена колонок:

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

Значение ячейки 5-ой строки 7-го столбца:

```
## 0
```

Вторая строка:

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

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

Просмотр таблицы с помощью команды `head()`

```
## 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  
## 1 60 4 1 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  
## 3 60 4 3 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  
## 5 60 4 5 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  
## 22 23  
## 1 0 0  
## 2 0 0  
## 3 0 0  
## 4 0 0  
## 5 0 0  
## 6 0 0
```

и `tail()`

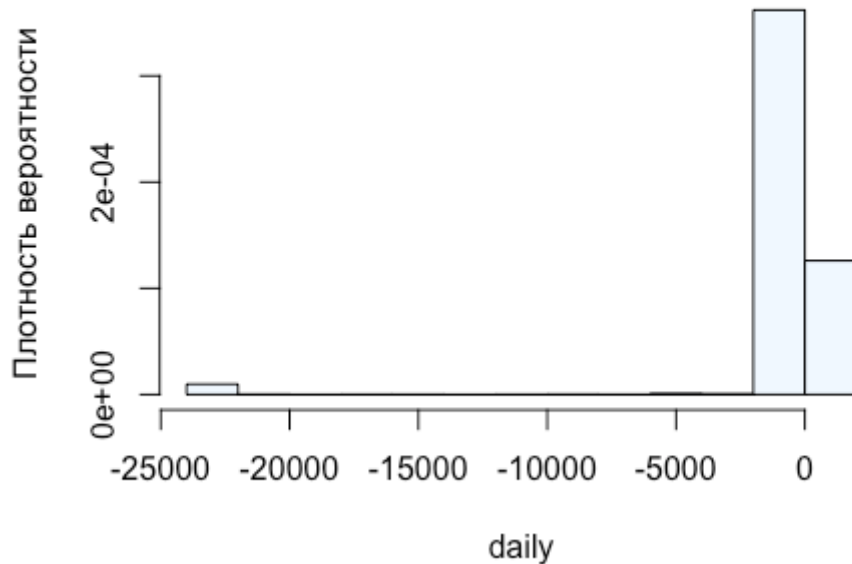
```
## year month day 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19  
20  
## 5065 80 11 25 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0  
## 5066 80 11 26 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0  
## 5067 80 11 27 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0  
## 5068 80 11 28 0 0 0 0 0 0 0 0 0 0 0 0 0 0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

```
## 5065 0 0 0
## 5066 0 0 0
## 5067 0 0 0
## 5068 0 0 0
## 5069 0 0 0
## 5070 0 0 0
```

Последние 24 колонки - количество осадков за час.

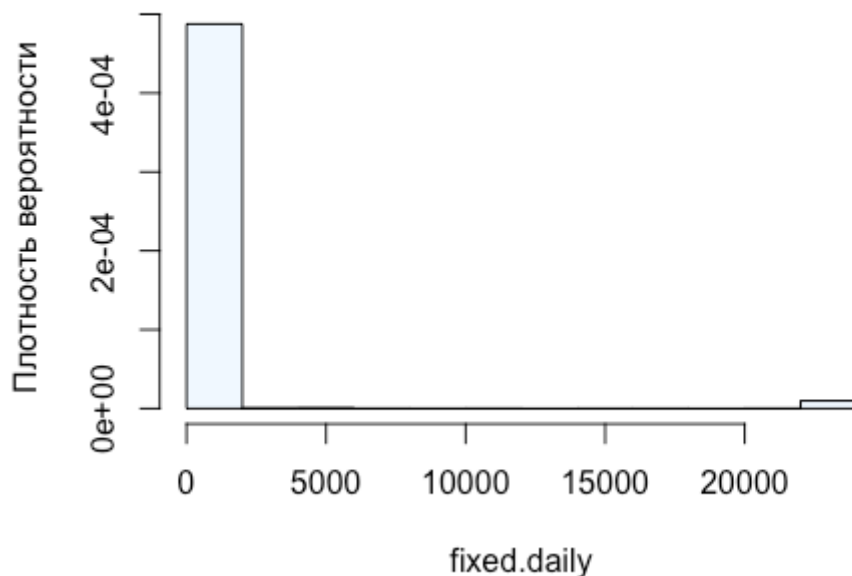
Гистограмма колонки `daily` - сумма крайних 24-х столбцов.

Гистограмма суммы крайних 24-столбцов



Полученная гистограмма не позволяет сделать выводы по данным, так как строится неверно. Чтобы исправить это, необходимо находить модуль суммы столбцов, содержащих количество осадков по часам. Результат этого добавлен к датафрейму `fixed.daily`.

Гистограмма суммы крайних 24-столбцов



Синтаксис и типизирование

Вектор `v`:

```
## 4 8 15 16 23 42
## Максимальное значение: 8
## Вектор по возрастанию: 15 16 23 4 42 8
```

Команда `max()` для вектора `v` возвращает второй элемент данного вектора, так как он

имеет наибольший код символа (ASCII). По такому же признаку работает команда `sort()`. Сумму для `v` посчитать невозможно, так как данная операция производится только с числовыми или логическими значениями.

Сумму `v2[2] + v2[3]` посчитать нельзя, так как вектор `v2` состоит из элементов типа `char`.

Датафрейм `df3`:

```
##    z1 z2 z3
## 1   5  7 12
## df3[1,2] + df3[1,3] = 19
```

Сумму `df3[1,2] + df3[1,3]` посчитать можно. В датафрейме каждый столбец - вектор, содержащий данные определенного типа, которые могут не совпадать. В данном случае `df3` состоит из числовых и категориальных данных.

Список `l4`:

```
## $z1
## [1] "6"
##
## $z2
## [1] 42
##
## $z3
## [1] "49"
##
## $z4
## [1] 126
##
## l4[[2]] + l4[[4]] = 168
```

Сумму `l4[[2]] + l4[[4]]` посчитать можно. Список может содержать объекты разных типов (здесь целые числа и символы). Операции с элементами списка возможны только через `[[]]`. Поэтому невозможно выполнить команду `l4[2] + l4[4]`.

Работа с функциями и операторами

Числа от 1 до 10000 с инкрементом 372:

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

Числа от 1 до 10000 длиной 50:

```
## 1 205.0612 409.1224 613.1837 817.2449 1021.306 1225.367 1429.429
## 1633.49 1837.551 2041.612 2245.673 2449.735 2653.796 2857.857 3061.918
## 3265.98 3470.041 3674.102 3878.163 4082.224 4286.286 4490.347 4694.408
## 4898.469 5102.531 5306.592 5510.653 5714.714 5918.776 6122.837 6326.898
## 6530.959 6735.02 6939.082 7143.143 7347.204 7551.265 7755.327 7959.388
## 8163.449 8367.51 8571.571 8775.633 8979.694 9183.755 9387.816 9591.878
## 9795.939 10000
```

Повтор чисел от 1 до 5 три раза (`times`):

```
## 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

Повтор каждого числа из последовательности от 1 до 5 по три раза (`each`):

```
## 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5
```