Урок 2. Работа с данными базовыми средствами R

В этом уроке мы уже непосредственно поработаем в данными из файла `firms.csv`. Файл содержит данные об эффективности рекламных кампаний с сайта IBM. Файл и описание данных можно найти на сайте IBM.

```
dat <- read.csv("/Users/allat/Desktop/firms.csv")
View(dat)  # прокомментировать, что в столбцах
```

Какую информацию о таблице мы можем получить?

Можем определить число наблюдений и число показателей в датафрейме. А можно узнать гораздо больше — структуру датафрейма: число наблюдений и переменных, типы переменных и примеры значений, которые они принимают. Сделать это можно с помощью функции `str()`:

```
str(dat) # прокомментировать, где какой тип
```

```
## 'data.frame':    548 obs. of  7 variables:
##  $ MarketID        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ MarketSize      : Factor w/ 4 levels "","Large","Medium",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ LocationID      : int  1 1 1 1 2 2 2 2 3 3 ...
##  $ AgeOfStore      : int  4 4 4 4 5 5 5 5 12 12 ...
##  $ Promotion       : int  3 3 3 3 2 2 2 2 1 1 ...
##  $ Week            : int  1 2 3 4 1 2 3 4 1 2 ...
##  $ SalesInThousands: num  33.7 35.7 29 39.2 27.8 ...
```

Как и в случае с векторами, можно посмотреть на первые несколько строк в таблице:

```
head(dat)
```

```
##   MarketID MarketSize LocationID AgeOfStore Promotion Week
## 1        1     Medium          1          4         3    1
## 2        1     Medium          1          4         3    2
## 3        1     Medium          1          4         3    3
## 4        1     Medium          1          4         3    4
## 5        1     Medium          2          5         2    1
## 6        1     Medium          2          5         2    2
##   SalesInThousands
## 1            33.73
## 2            35.67
## 3            29.03
## 4            39.25
## 5            27.81
## 6            34.67
```

Или на последние:

```
tail(dat)
```

```
##     MarketID MarketSize LocationID AgeOfStore Promotion Week
## 543       10      Large        919          2         1    3
## 544       10      Large        919          2         1    4
## 545       10      Large        920         14         2    1
## 546       10      Large        920         14         2    2
## 547       10      Large        920         14         2    3
## 548       10      Large        920         14         2    4
##     SalesInThousands
```

```
## 543            57.20
## 544            64.34
## 545            50.20
## 546            45.75
## 547            44.29
## 548            49.41
```

Если нужно получить более содержательное, статистическое описание данных, можно воспользоваться функцией `summary()`:

```
summary(dat)
```

```
##     MarketID       MarketSize    LocationID      AgeOfStore
##  Min.   : 1.000          :  1   Min.   :  1.0   Min.   : 1.000
##  1st Qu.: 3.000   Large :167   1st Qu.:216.0   1st Qu.: 4.000
##  Median : 6.000   Medium:320   Median :504.0   Median : 7.000
##  Mean   : 5.715   Small : 60   Mean   :479.7   Mean   : 8.569
##  3rd Qu.: 8.000                3rd Qu.:708.0   3rd Qu.:12.000
##  Max.   :10.000                Max.   :920.0   Max.   :28.000
##                                                NA's   :17
##    Promotion          Week      SalesInThousands
##  Min.   :1.000   Min.   :1.00   Min.   :17.34
##  1st Qu.:1.000   1st Qu.:1.75   1st Qu.:42.56
##  Median :2.000   Median :2.50   Median :50.20
##  Mean   :2.029   Mean   :2.50   Mean   :53.47
##  3rd Qu.:3.000   3rd Qu.:3.25   3rd Qu.:60.44
##  Max.   :3.000   Max.   :4.00   Max.   :99.65
##                                 NA's   :3
```

Эта функция для текстовых столбцов вернет число уникальных элементов каждого вида (число фирм на разных рынках в `MarketSize`), а для числовых — статистические характеристики: минимальное и максимальное значение, среднее арифметическое (`Mean`), медиану (`Median`) и нижний и верхний квартили (`1st Qu.` и `3rd Qu`). Так, например, мы можем сказать, что число продаж фирм в 50% случаев не превышает значение 50.20 (медиана), в 25% случаев — значение 42.56 (нижний квартиль) и в 75% случаев — значение 60.44 (верхний квартиль).

При работе с данными часто приходится удалять пропущенные значения, потому что иначе мы не сможем полноценно работать с таблицей (многие функции не работают при наличии `NA`, а у некоторых необходимо указывать дополнительный аргумент — учитывать `NA` или нет).

```
# удаляем строки, содержащие NA
dat <- na.omit(dat)
```

Если мы хотим обратиться к конкретному показателю в таблице и рассматривать его как вектор элементов, нужно использовать символ `$`.

```
dat$AgeOfStore  # число лет на рынке
```

```
##   [1]  4  4  4  4  5  5  5  5 12 12 12  1  1 10 10 10 10 10 10 10 10 15 15
##  [24] 15 15 10 10 10 10  6  6  6  6  5  5  5  5  5  5  5  5 12 12 12 12 12
##  [47] 12 12 12 22 22 22 22  8  8  8  8 22 22 22 22 19 19 19 19  8  8  8  5
##  [70]  5  5  5  4  4  4  4  8  8  8  8 12 12 12 12  1  1  1  1 19 19 19 19
##  [93] 11 11 11 11  1  1  1  1  1  1  1  1  1  1  1 19 19 19 19 13 13 13 13
## [116]  3  3  3  4  4  4  4  5  5  5  5  7  7  7  7  4  4  4  4  5  5  5  2
## [139]  2  2  2  7  7  7  7  3  3  3  3 23 23 23  6  6  6  6  1  1  1  7  7
## [162]  7  9  9  9  3  3  3  3  2  2  2 13 13 13 13  7  7  7 28 28 28 28  8
## [185]  8  8  8  5  5  5 25 25 25 19 19 19 19  8  8  8  8  4  4  4  4  3  3
## [208]  3  3  6  6  6  6  9  9  9  9 14 14 14 14 11 11 11 11 23 23 23 23  6
```

```
## [231]  6  6  1  1  1  1  1  1 12 12 12 12  6  6  6  6 19 19 19 19  2  2  2
## [254]  2  4  4  4  4  1  1  1  1 12 12 12 12  5  5  5  5  5  5  5  8  8  8
## [277]  8  7  7  7  7 24 24 24 24  4  4  4  4  7  7  7  7  6  6  6  6  8  8
## [300]  8  8  7  7  7  7  4  4  4  4  4  4  4  4 11 11 11 11  8  8  8  8  8
## [323]  8  8  8  3  3  3  3 18 18 18 18  9  9  9  9  4  4  4  4 11 11 11 11
## [346]  1  1  1  1 18 18 18 18  8  8  8  8  1  1  1  1  1  1  1  1 13 13 13
## [369] 13  4  4  4  4 27 27 27 27 22 22 22 22 17 17 17 17  5  5  5  5 15 15
## [392] 15 15  1  1  1  1 24 24 24 24  1  1  1  1  9  9  9  9  3  3  3  3  9
## [415]  9  9  9 10 10 10 10  1  1  1  1  6  6  6  6  5  5  5  5  1  1  1  1
## [438] 20 20 20 20  9  9  9  9 13 13 13 13  7  7  7  7  1  1  1  1  7  7  7
## [461]  7 10 10 10 10  2  2  2  2 13 13 13 13 10 10 10 10  3  3  3  3  1  1
## [484]  1  1  1  1  1  1  1  1  1  1  6  6  6  6 24 24 24 24  9  9  9  9  3
## [507]  3  3  3  7  7  7  7 14 14 14 14  6  6  6  6  2  2  2  2 14 14 14 14
```

Так как полученный результат является вектором, при необходимости к его элементам можно обращаться уже знакомым образом:

```
dat$AgeOfStore[2]  # 2ой элемент
```

```
## [1] 4
```

Точно так же, используя `$`, можно добавлять в датафрейм новые столбцы. Например, добавим логарифмированный показатель продаж `LogSales`:

```
dat$LogSales <- log(dat$SalesInThousands)
```

Часто при работе с данными возникает необходимость выбрать несколько показателей или определенную группу наблюдений и анализировать их отдельно — чтобы не загружать каждый раз огромную базу с ненужными показателями.

Можем выбрать несколько столбцов и сохранить их в другой датафрейм `dat1`:

```
dat1 <- dat[2:4] # 2 и 4 - порядковые номера столбцов, от 2 до 4
```

Если выбираем столбцы не подряд, их номера обязательно нужно оформить в виде вектора:

```
dat[c(1, 3)] # не просто dat[1, 3]
```

В противном случае получится совсем не то:

```
dat[1, 3]
```

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

Это «совсем не то» связано с тем, что, когда мы указываем в квадратных скобках числа через запятую, R воспринимает первое число как номер строки, второе число — как номер столбца (как в матрицах — сначала строка, потом столбец). Можем посмотреть на исходную базу и убедиться в этом. В целом, манипуляции с выбором строк и столбцов по номеру в датафреймах ничем не отличается от работы с матрицами в R.

Если хотим отобрать из базы определенные наблюдения, это тоже можно сделать с помощью квадратных скобок. Например, мы хотим выбрать данные за первую неделю:

```
week1 <- dat[dat$Week == 1, ]
View(week1)
```

Тут важно не забыть поставить запятую, чтобы R понимал, что мы накладываем условие на строки, а столбцы берем все, что есть. Можем сочетать условия. Например, выбрать данные за первую неделю по компаниям среднего уровня:

```r
View(dat[dat$Week == 1 & dat$MarketSize == "Medium", ])
```

На этом мы пока закончим знакомство с датафреймами и перейдем к практическому заданию. А в следующем уроке поговорим о довольно мощной библиотеке `tidyverse`.