Математические пакеты

Таблицы данных в R и работа с ними

Сучков Андрей Игоревич

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

31 октября 2020 г.

Таблицы данных (data frame)

Определение

- Таблица данных – это более широко используемый по сравнению с матрицей объект, поскольку разные столбцы могут содержать разные типы данных (числовой, текстовый и т. д.)
- Таблица данных – это самая часто используемая структура данных в R
- Для создание таблицы данных используется функция `data.frame(col1, col2, col3, ...)`
- `coli` – вектор, любого типа (текстового, числового или логического) – будущие столбцы таблицы
- Имена столбцов можно изменить с помощью функции `names()`

Таблицы данных (data frame)

Создание таблицы данных

Листинг 1: Создание таблицы данных

```
patientID <- c(1, 2, 3, 4)
age <- c(25, 34, 28, 52)
diabetes <- c("Type1", "Type2", "Type1", "Type1")
status <- c("Poor", "Improved", "Excellent", "Poor")
patientdata <- data.frame(patientID, age, diabetes, status)
```

	patientID	age	diabetes	status
1	1	25	Type1	Poor
2	2	34	Type2	Improved
3	3	28	Type1	Excellent
4	4	52	Type1	Poor

Таблицы данных (data frame)

Обозначение элементов таблицы данных

Листинг 2: Создание таблицы данных

```
patientdata[1:2]  ## return columts "patientID" and "age"

patientdata[c("diabetes", "status")]

patientdata$age

table(patientdata$diabetes, patientdata$status)
```

Таблицы данных (data frame)

Функции attach, detach и with

1. Функция `attach()` добавляет указанную таблицу данных к пути поиска R. Когда указывается имя переменной, программа ищет эту переменную в таблицах данных, включенных в траекторию поиска
2. Функция `detach()` удаляет таблицу данных из пути поиска. Данная функция ничего не делает с самим объектом. Ввод этой команды необязателен, но он полезен при программировании, и про него не следует забывать
3. Функция `with()` ипользуется так же как и связка `attach-detach`. Ограничение функции `with()` заключается в том, что она не действует за пределами фигурных скобок

Таблицы данных (data frame)

Пример использования attach-detach

Листинг 3: Использование attach-detach

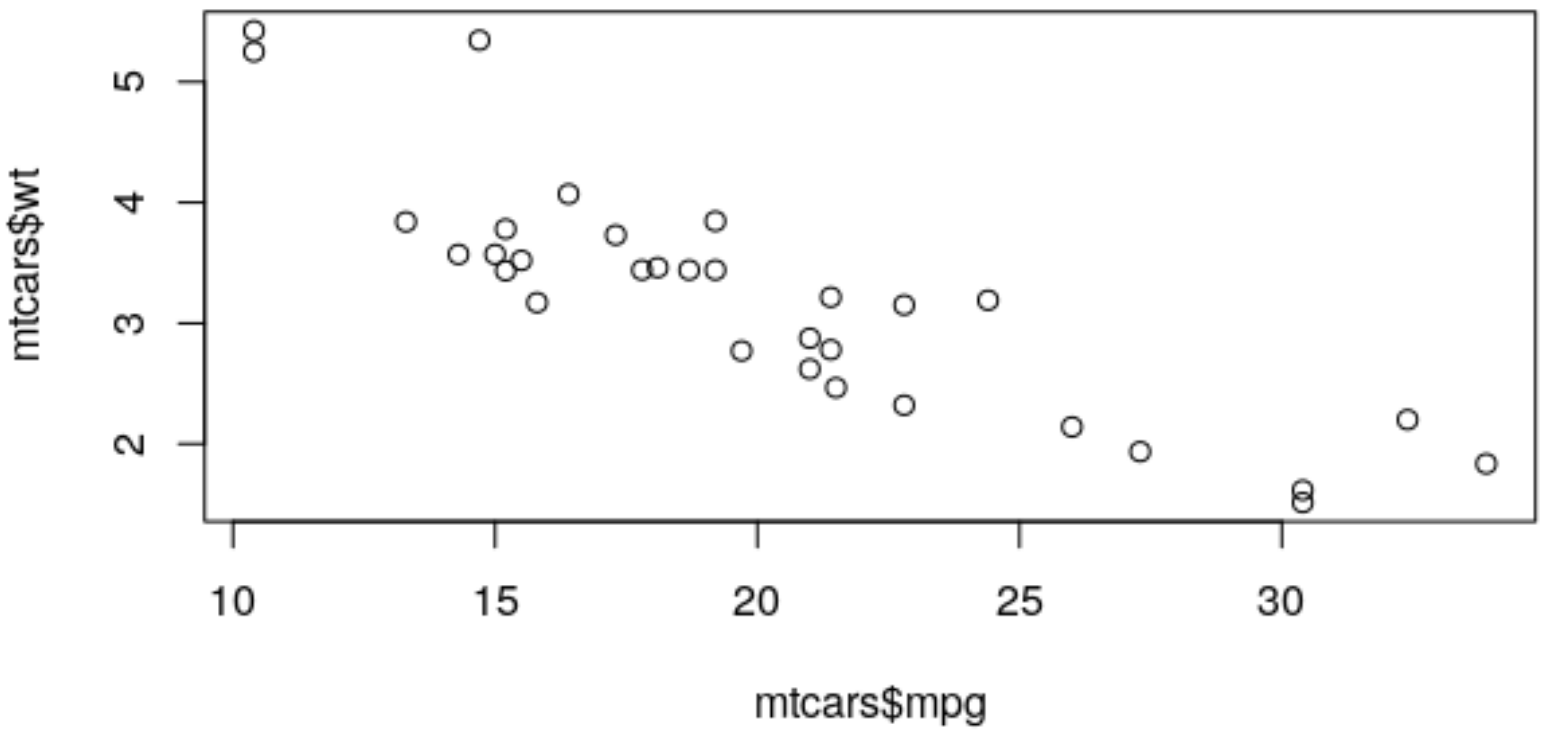
```
summary(mtcars$mpg)
plot(mtcars$mpg, mtcars$disp)
plot(mtcars$mpg, mtcars$wt)

## Use attach-detach
attach(mtcars)
  summary(mpg)
  plot(mpg, disp)
  plot(mpg, wt)
detach(mtcars)
```

Таблицы данных (data frame)

Результаты

```
> summary(mtcars$mpg)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.40   15.43   19.20   20.09   22.80   33.90
```



Таблицы данных (data frame)

Пример использования attach-detach

Листинг 4: Использование with()

```
with(mtcars, {
  nokeepstats <- summary(mpg)  # we not found this object
  keepstats <<- summary(mpg)
  plot(mpg, disp)
  plot(mpg, wt)
})
```

Наблюдение

В большинстве руководств по R рекомендуется использовать with(), а не attach()-detach(). Но здесь зависит от индивидуальных предпочтений.

Факторы (factors)

Определение

- Факторы – это категориальные (номинальные и порядковые) данные
- Факторы очень важны в R, поскольку они определяют, как данные будут проанализированы и графически представлены
- Функция `factor()` сохраняет категориальные данные в виде вектора из целых чисел в диапазоне от одного до k (где k – число уникальных значений категориальной переменной) и в виде внутреннего вектора из цепочки символов (исходных значений переменной), соответствующим этим целым числам
- Присвоение числовых значений происходит в алфавитном порядке

Листинг 5: Создание факторов

```
diabetes <- c("Type1", "Type2", "Type1", "Type1")
diabetes <- factor(diabetes)  # transform (1, 2, 1, 1):
                              # 1 = Type1
                              # 2 = Type2

status <- c("Poor", "Improved", "Excellent", "Poor")
status <- factor(status,          # transform (3, 2, 1, 3)
                 ordered=TRUE)    # 1 = Excellent
                                  # 2 = Improved
                                  # 3 = Poor

status <- factor(status,  # transform (1, 2, 3, 1)
                 ordered=TRUE,
                 levels=c("Poor",
                          "Improved",
                          "Excellent"))
```

Факторы (factors)

Пример работы с факторами

Листинг 6: Использование факторов

```
patientID <- c(1, 2, 3, 4)
age <- c(25, 34, 28, 52)
diabetes <- c("Type1", "Type2", "Type1", "Type1")
status <- c("Poor", "Improved", "Excellent", "Poor")
diabetes <- factor(diabetes)
status <- factor(status, order=TRUE)
patientdata <- data.frame(patientID, age, diabetes, status)

str(patientdata)
summary(patientdata)
```

Списки (list)

Определение

- Список – это упорядоченный набор объектов (компонентов)
- Список может объединять разные (возможно, не связанные между собой) объекты под одним именем
- Список можно создать при помощи функции `list(obj1, obj2, ...)`
- Объектам в списке можно присваивать имена: `list(name1=obj1, name2=obj2, ...)`

Списки (list)

Создание списка

Листинг 7: Создание списка

```r
g <- "My First List"
h <- c(25, 26, 18, 39)
j <- matrix(1:10, nrow=5)
k <- c("one", "two", "three")

mylist <- list(title=g, ages=h, j, k)

mylist[[2]]         # return c(25, 26, 18, 39)
mylist[["ages"]]    # equivalent
```

Полезные функции для работы с объектами

- `length(obj)` – число элементов/компонентов объекта
- `dim(obj)` – число измерений объекта
- `class(obj)` – класс или тип объекта
- `cbind(obj1, obj2, ...)` – объединяет объекты в виде столбцов
- `rbind(obj1, obj2, ...)` – объединяет объекты в виде строк
- `head(obj)` – выводит на экран первую часть объекта
- `tail(obj)` – выводит на экран последнюю часть объекта
- `ls()` – выводит на экран список имеющихся объектов