Pair_Programming_alternative_slicing

HDS

2024-08-13

HDS, August 13, 2024 Checked 01/03/2025

Alternative Slicing

The dplyr package has a lot of nice tools for sorting and slicing and summarizing but there are other ways to do it.

In base R we can slice data frames using conditional statements, these are often used to select both rows and columns

Last week we used Tidyverse tools to sort and slice, we need some familiarity with base R as well.

Why discuss this

R almost always has at least three valid ways to do everything The dplyr style slicing and sorting is really convenient, but dplyr is only available in R, in Python and other languages, slicing is done in the base R style.

The base R style slicing was taken from linear algebra (aka matrix algebra), and appears in many other languages.

The tidyverse methods are derived from SQL-based methods of sorting and slicing. These are also seen in the Polars package for Python.

There is also a package called sqldf that allows you to use SQL queries on R dataframes. If you have a lot of experience with sql and want to leverage that, look at sqldf:

https://www.geeksforgeeks.org/how-to-write-a-sql-query-in-r/ (https://www.geeksforgeeks.org/how-to-write-a-sql-query-in-r/)

We don't have time in DSE5001 to teach this, but if you do know SQL well, you can teach yourself sqldf pretty easily.

We will use mtcars as an example again to look at base R style manipulations.

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

We can access portions of the dataframe using numeric ranges

We give the rows first, then the columns (this is a linear algebra convention most languages follow, rows, then columns)

```
mtcars[1:3,1:5]
```

```
##                mpg cyl disp  hp drat
## Mazda RX4     21.0   6  160 110 3.90
## Mazda RX4 Wag 21.0   6  160 110 3.90
## Datsun 710    22.8   4  108  93 3.85
```

We can access columns by name

```
mtcars[1:3,c("mpg","cyl","drat")]
```

```
##                mpg cyl drat
## Mazda RX4     21.0   6 3.90
## Mazda RX4 Wag 21.0   6 3.90
## Datsun 710    22.8   4 3.85
```

We can access by a condition statement, usually for rows, much like a filter (well, identically to a filter)

```
mtcars[mtcars$mpg>28,c("mpg","cyl")]
```

```
##                 mpg cyl
## Fiat 128       32.4   4
## Honda Civic    30.4   4
## Toyota Corolla 33.9   4
## Lotus Europa   30.4   4
```

We can do more complex slices using logical OR (|) and AND (&) operators

```
mtcars[(mtcars$cyl==6)&(mtcars$hp>150),c("mpg","cyl")]
```

```
##               mpg cyl
## Ferrari Dino 19.7   6
```

Which

We can get the row numbers of variables that mean a specific condition using the which operation

```
which((mtcars$cyl==6)&(mtcars$hp>150))
```

```
## [1] 30
```

we can then look at that row

leaving the columns blank gives us all columns

```
mtcars[30,]
```

```
##               mpg cyl disp  hp drat   wt qsec vs am gear carb
## Ferrari Dino 19.7   6  145 175 3.62 2.77 15.5  0  1    5    6
```

This is helpful looking for a minimum or maximum, for example

```
max_hp_row=which(mtcars$hp==max(mtcars$hp))
mtcars[max_hp_row,]
```

```
##               mpg cyl disp  hp drat   wt qsec vs am gear carb
## Maserati Bora  15   8  301 335 3.54 3.57 14.6  0  1    5    8
```

Question/Action

-Find the car with the highest mpq and print out it's row

```
max_mpg_row=which(mtcars$mpg==max(mtcars$mpg))
max_mpg_row
```

```
## [1] 20
```

-Find the cars with 8 cylinders and hp less than 150

```
slice<-which((mtcars$cyl==8)&(mtcars$hp<150))
slice
```

```
## integer(0)
```

```
mtcars[slice,]
```

```
##  [1] mpg  cyl  disp hp   drat wt   qsec vs   am   gear carb
## <0 rows> (or 0-length row.names)
```

Trick question, there aren't any.

tapply

Tapply is an alternative to the summarize function in dplyr

It will apply any function to a column of a dataframe, grouped by one or more other variables

we are applying the function mean() to the data mtcars$mpg, grouped by cyl and am

```
tapply(mtcars$mpg,list(mtcars$cyl,mtcars$am),mean)
```

```
##        0        1
## 4 22.900 28.07500
## 6 19.125 20.56667
## 8 15.050 15.40000
```

Question/Action

Load the iris data

```
data(iris)
```

Find the mean value of Sepal.Length grouped by species

```
tapply(iris$Sepal.Length,list(iris$Species),mean)
```

```
##     setosa versicolor  virginica
##      5.006      5.936      6.588
```

Find the standard deviation of Sepal.Length by species as well

```
tapply(iris$Sepal.Length,list(iris$Species),sd)
```

```
##     setosa versicolor  virginica
##  0.3524897  0.5161711  0.6358796
```