dataframe

Nicholus Tint Zaw

2022-08-14

What we are going to cover in this exercise?

1. Creating a Data Frame
2. Exploring dataframe
3. Dataframe indexing
4. Running Functions on Dataframes

Creating a Data Frame

as.dataframe

```
# Definition of vectors
# ref: https://towardsdatascience.com/introduction-to-data-frames-in-r-b9a6302d9a56

name <- c("Mercury", "Venus", "Earth", "Mars",
          "Jupiter", "Saturn", "Uranus", "Neptune")

type <- c("Terrestrial planet", "Terrestrial planet",
          "Terrestrial planet", "Terrestrial planet",
          "Gas giant", "Gas giant", "Gas giant", "Gas giant")

diameter <- c(0.382, 0.949, 1, 0.532, 11.209, 9.449, 4.007, 3.883)

rotation <- c(58.64, -243.02, 1, 1.03, 0.41, 0.43, -0.72, 0.67)

rings <- c(FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, TRUE)

# Create a data frame from the vectors
planets_df <- data.frame(name, type, diameter, rotation, rings)
```

Exploring dataframe

```
# import base-dataframe mtcars
df <- mtcars
```

```r
# view dataframe
View(planets_df)

View(df)

# return the column names
names(planets_df)
```

```
## [1] "name"     "type"     "diameter" "rotation" "rings"
```

```r
names(df)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

```r
# print subset of dataframe
head(df)
```

```
##                   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
```

```r
tail(df)
```

```
##                 mpg cyl  disp  hp drat    wt qsec vs am gear carb
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

```r
# dimension of dataframe
dim(df) # returns the number of rows first, then the number of columns.
```

```
## [1] 32 11
```

```r
# number of observation in dataframe
length(df$mpg)
```

```
## [1] 32
```

```r
# check below syntax
length(df)
```

```
## [1] 11
```

```r
# class of dataframe and columns
class(df)
```

```
## [1] "data.frame"
```

```r
class(df$cyl)
```

```
## [1] "numeric"
```

```r
# check structure of dataframe
str(df)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```r
str(planets_df)
```

```
## 'data.frame':    8 obs. of  5 variables:
##  $ name    : chr  "Mercury" "Venus" "Earth" "Mars" ...
##  $ type    : chr  "Terrestrial planet" "Terrestrial planet" "Terrestrial planet" "Terrestrial planet
##  $ diameter: num  0.382 0.949 1 0.532 11.209 ...
##  $ rotation: num  58.64 -243.02 1 1.03 0.41 ...
##  $ rings   : logi  FALSE FALSE FALSE FALSE TRUE TRUE ...
```

Dataframe indexing

using square brackets []

```r
# Return the value in the first row and first column:
df[1,1]
```

```
## [1] 21
```

```r
# Return the value in the second row and first column:
df[2,1]
```

```
## [1] 21
```

```r
# Return the value in the third row and second column:
df[3,2]
```

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

```r
# Return all the values in the first row:
df[1,]
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4  21   6  160 110  3.9 2.62 16.46  0  1    4    4
```

```r
# Return the values in the first through seventh rows, in the second column:
mtcars[1:7,2]
```

```
## [1] 6 6 4 6 8 6 8
```

use the $ operator to refer to the column within the dataframe

```r
df$cyl
```

```
##  [1] 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4 8 6 8 4
```

```r
df$wt
```

```
##  [1] 2.620 2.875 2.320 3.215 3.440 3.460 3.570 3.190 3.150 3.440 3.440 4.070
## [13] 3.730 3.780 5.250 5.424 5.345 2.200 1.615 1.835 2.465 3.520 3.435 3.840
## [25] 3.845 1.935 2.140 1.513 3.170 2.770 3.570 2.780
```

Some advance indexing

application of condition in indexing

```r
# return only the rows of data with cars that have four cylinders
df[which(df$cyl == 4),1]
```

```
##  [1] 22.8 24.4 22.8 32.4 30.4 33.9 21.5 27.3 26.0 30.4 21.4
```

```r
# return the cars withmore than 90 horsepower.
df[which(df$hp > 90),]
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
```

```
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
# return the cars with high mpg and low weight.
df[which(df$mpg > 28 & df$wt < 2),]
```

```
##                 mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Honda Civic    30.4   4 75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla 33.9   4 71.1  65 4.22 1.835 19.90  1  1    4    1
## Lotus Europa   30.4   4 95.1 113 3.77 1.513 16.90  1  1    5    2
```

```
# use indexing on just one column of a dataframe
df$mpg[which(df$cyl == 4)]
```

```
##  [1] 22.8 24.4 22.8 32.4 30.4 33.9 21.5 27.3 26.0 30.4 21.4
```

```
# run functions on subsets of a column of data
mean(df$mpg[which(df$cyl == 4)])
```

```
## [1] 26.66364
```

Running Functions on Dataframes

```
# How many values are in the mpg column?
length(df$mpg)
```

```
## [1] 32
```

```r
# What is the average horsepower of these cars?
mean(df$hp)
```

```
## [1] 146.6875
```

```r
# What is the range of the weights of these cars?
range(df$wt)
```

```
## [1] 1.513 5.424
```

```r
# What is the frequency of cylinder type?
table(df$cyl)
```

```
##
##  4  6  8
## 11  7 14
```

```r
table(planets_df$type)
```

```
##
##          Gas giant Terrestrial planet
##                  4                  4
```