Lecture 3: Data Frames

STAT 450, Fall 2021

Data sets in R are represented as objects called `data frames`. The columns of a data frame are the **variables**, and the rows of a data frame are the **observations**. The columns of a data frame can be different types (e.g., numeric, character, logical).

As an example, lets start by examining a data frame called `mtcars`, which is already loaded into R. This data set comes from the 1974 *Motor Trend* magazine, and contains variables (columns) on automobile design and performance for 32 different automobile models (rows).

```
head(mtcars) # preview first several rows
```

```
##                    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
```

The `head()` function provides a preview of the first several rows of the data frame. You can also type `mtcars` into the console, and it will print out the entire data set. But for larger data sets this is not advisable. To learn more about this data, use the help menu by typing `help(mtcars)` into the console.

Below are some functions that provide information on the size of a data frame:

```
nrow(mtcars) # number of rows
```

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

```
ncol(mtcars) # number of columns
```

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

```
dim(mtcars) # dimension
```

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

We can also extract the row and column names:

```
names(mtcars) # names of the columns (variables)
```

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

```
row.names(mtcars) # names of the rows (car models)
```

```
##  [1] "Mazda RX4"           "Mazda RX4 Wag"       "Datsun 710"
##  [4] "Hornet 4 Drive"      "Hornet Sportabout"   "Valiant"
##  [7] "Duster 360"          "Merc 240D"           "Merc 230"
## [10] "Merc 280"            "Merc 280C"           "Merc 450SE"
## [13] "Merc 450SL"          "Merc 450SLC"         "Cadillac Fleetwood"
## [16] "Lincoln Continental" "Chrysler Imperial"   "Fiat 128"
## [19] "Honda Civic"         "Toyota Corolla"      "Toyota Corona"
## [22] "Dodge Challenger"    "AMC Javelin"         "Camaro Z28"
## [25] "Pontiac Firebird"    "Fiat X1-9"           "Porsche 914-2"
## [28] "Lotus Europa"        "Ford Pantera L"      "Ferrari Dino"
## [31] "Maserati Bora"       "Volvo 142E"
```

Subsetting rows and columns

To subset a specific column, or variable, use the `$` operator. For example, we can extract the `mpg` column (miles per gallon) from `mtcars`:

```
mtcars$mpg
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

Notice that when we subset a column this way, we are actually extracting a vector. So we can compute summary statistics using the vector functions mentioned in the previous lecture.

```
mean(mtcars$mpg)
```

```
## [1] 20.09062
```

```
median(mtcars$mpg)
```

```
## [1] 19.2
```

```
summary(mtcars$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.43   19.20   20.09   22.80   33.90
```

Columns and rows of a data frame can also be extracted by index (position) or name using brackets `[]`:

```
mtcars[1, ] # extract first row
```

```
##           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
```

```
mtcars[, 1] # extract first column
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
mtcars["Datsun 710", ] # extract row for Datsun 710
```

```
##             mpg cyl disp hp drat   wt  qsec vs am gear carb
## Datsun 710 22.8   4  108 93 3.85 2.32 18.61  1  1    4    1
```

```
mtcars[2, 3] # extract element in the second row and third column
```

```
## [1] 160
```

```
mtcars[2:3, ] # extract second and third row
```

```
##                mpg cyl disp  hp drat    wt  qsec vs am gear carb
## 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
```

Exercise: Calculate some summary statistics (min, max, mean, median, etc.) for the car weight variable (`wt`).

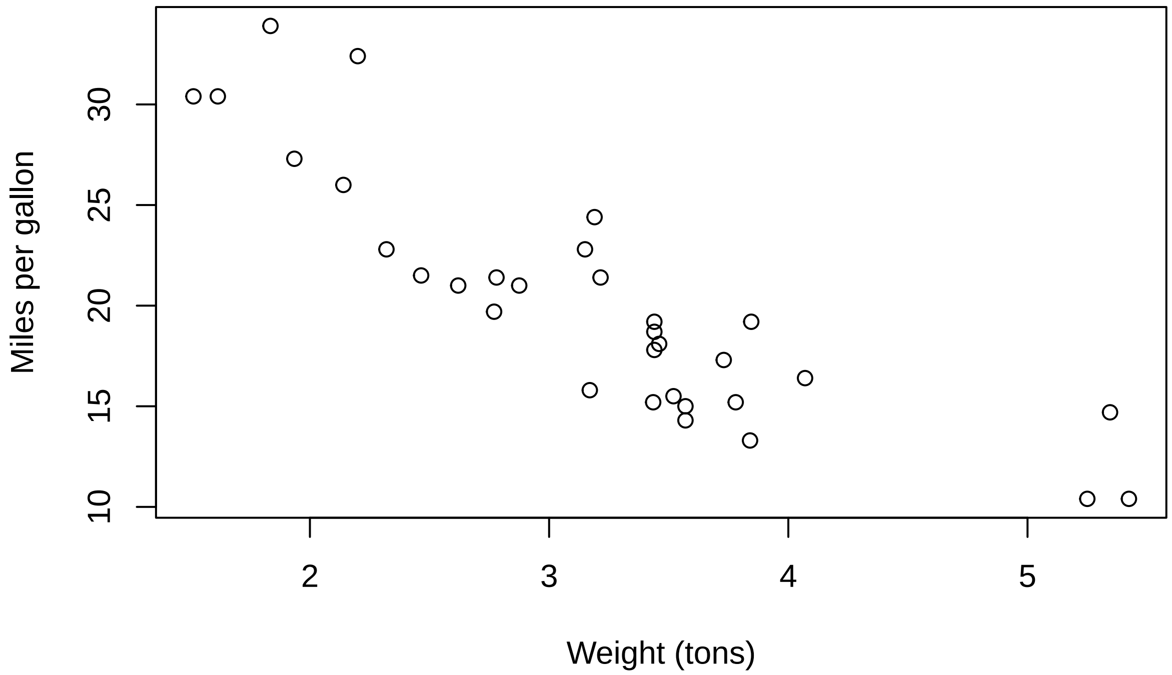
Exercise:

a) Subset the 15th row of `mtcars`
b) Subset the row for the "Maserati Bora"
c) Subset the last 5 rows of `mtcars`

Scatter Plot

Scatter plots are a type of graph used to show the relationship between two numeric variables. To make a scatter plot in R use the `plot()` function. For example, below is scatter plot with car weight on the x-axis and mileage on the y-axis. Each point represents a different car model from the `mtcars` data frame.

```
plot(mtcars$wt, mtcars$mpg, xlab="Weight (tons)", ylab="Miles per gallon")
```



Based on this scatter plot we can see that there is a decreasing and linear relationship between weight and mileage. That is, as weight increases the mileage decreases, on average, for car models in this data set, which makes sense.

Exercise: Make a plot with `hp` on the x-axis and `mpg` on the y-axis. Label x-axis “Horsepower” and the y-axis “Miles per gallon”. Describe the association between the two variables.

Handling Missing Data

The function `na.omit()` can be used to remove rows that have missing data. For example, let's create a data frame with some `NA` values.

```r
df1 <- data.frame(
  x = c(7, 6, 4, 10, NA, NA),
  y = c(3, 4, NA, 2, 5 , 4),
  z = c(7, 8, NA, 10 , 6, NA)
)
df1
```

```
##    x  y  z
## 1  7  3  7
## 2  6  4  8
## 3  4 NA NA
## 4 10  2 10
## 5 NA  5  6
## 6 NA  4 NA
```

The third, fifth, and sixth rows contain `NA` values. The `na.omit()` function will remove these rows.

```r
df2 <- na.omit(df1)
df2
```

```
##    x y  z
## 1  7 3  7
## 2  6 4  8
## 4 10 2 10
```