Introduction to Dataframes

TRSM Bootcamps R Team

Some tips:

1. If concepts do not make sense, be sure to ask questions.
2. If you do not understand the concepts after asking, follow along and it will either eventually make sense or we can explain it in detail for you during a break.
3. Lines starting with "##" represent the result of the code that was ran.

What is a dataframe?

- Table of row and columns
- Named spreadsheet
- A data frame is a list of variables of the same number of row with unique row names, given class data.frame.
- Each column can have different data types (numerical or strings)

Note: In a dataframe, rows are called ***records*** and columns are called ***variables***.

```
d <- data.frame(courses=c("ECN340", "MTH108", "CPS109", "ECN702", "ITM320"),
                rates=c(9.7, 5.4, 7.8, 6.5, 8.9))
d
```

```
##   courses rates
## 1  ECN340   9.7
## 2  MTH108   5.4
## 3  CPS109   7.8
## 4  ECN702   6.5
## 5  ITM320   8.9
```

Importing Dataframes

- Use read_csv function from tidyverse package to import dataframes

```
# Open the tidyverse package
library(tidyverse)

# Import the dataframe and assign it a name
housing_df <- read_csv("https://www.dropbox.com/s/tvvtf9dwjufo7os/housing_train.csv?dl=1")
```

Indexing with Dataframes

- Indexing is simple way of splicing a dataset so that only particular row or columns are displayed
- For those experienced with Python, indexing starts at 1 in R instead of 0
  - d[1:3,] -> rows 1 to 3, all columns
  - d[1:4, 2] -> rows 1 to 4 in only the second column
  - d[1, 1] -> first row, first column

Exercises:

1. Retrieve rows 10-15 of the first 5 columns in housing_df.
2. Display the value in the thirty-fifth row of the eighteenth column of housing_df.

Exploring dataframes

- **unique(df$column_name)**: returns the unique values in the column of a dataframe

```
unique(d$courses)
```

```
## [1] "ECN340" "MTH108" "CPS109" "ECN702" "ITM320"
```

- **nrow(df$column_name)**: returns the number of rows

```
nrow(d)
```

```
## [1] 5
```

- **ncol(df$column_name)**: returns the number of cols

```
ncol(d)
```

```
## [1] 2
```

- **View(df)**: shows the dataframe in a spreadsheet

```
View(housing_df)
```

- **sd(df$column_name, na.rm=T)**: Standard deviation of the column
  - There might be missing values in the given column
  - na.rm = T will ***exclude*** the missing values and compute Standard Deviation

```
# This will result in NA
sd(housing_df$LotFrontage)
```

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

```
# This one excludes the missing values (NA)
sd(housing_df$LotFrontage, na.rm = T)
```

```
## [1] 24.28475
```

- **mean(df$column_name, na.rm = T)**: Mean of the column
  - There might be missing values in the given column
  - na.rm = T will ***exclude*** the missing values and compute mean

```
# This will result in NA
mean(housing_df$LotFrontage)
```

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

```
# This one excludes the missing values (NA)
mean(housing_df$LotFrontage, na.rm = T)
```

```
## [1] 70.04996
```

Exercise:

1. What is the average of SalePrice column in housing_df?
2. How many rows and columns does the housing_df have?
3. How many unique values are in the ***MSSubClass*** column? (CHALLENGE: Try and see if you can use a second function to sell you the total number instead of counting)