R-Help Session 1

January 29th, 2019

Downloading Data into R

- Importing directly from GitHub
  - https://github.com/brianlukoff/sta371g
  - Data ➔ Click on relevant data set
  - Click on Raw
  - Copy the URL and place it in this command, like this
    - ```
      Motorcycles <-
      read.csv("https://raw.githubusercontent.com/brianlukoff/sta371g/
      master/data/Motorcycles.csv ")
      ```
- Import “Motorcycles.csv” into R-Studio
  - Save “Motorcycles.xlsx” file through Microsoft Excel
  - In R-Studio, under “Environment” tab, click “Import Dataset”
  - Click “From Excel…”
  - Click “Browse…” and navigate to the appropriate folder
  - Click “Import”.
      ```

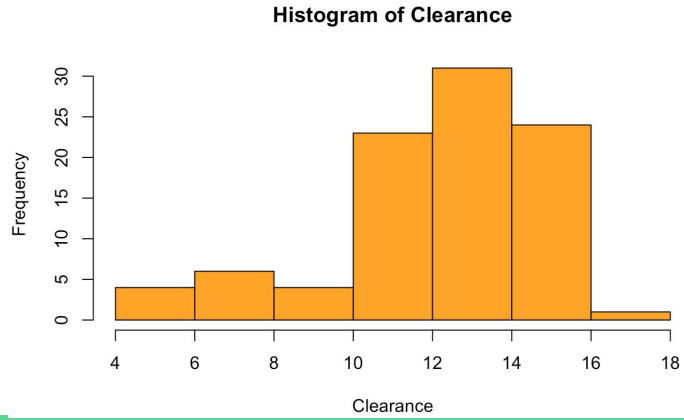
Simplifying Commands

- Naming variables helps to simplify running regressions
- You can name variables using a generic command: “Name <- Data$Variable”
  - For example, if I want to name the MSRP Variable as just “MSRP”, I would do the following:
    - `> MSRP <- Motorcycles$MSRP`
- Practice with remaining variables

```
> Model <- Motorcycles$Model
> MSRP <- Motorcycles$MSRP
> Bore <- Motorcycles$Bore
> Displacement <- Motorcycles$Displacement
> Clearance <- Motorcycles$Clearance
> EngineStrokes <- Motorcycles$EngineStrokes
> Wheelbase <- Motorcycles$Wheelbase
```

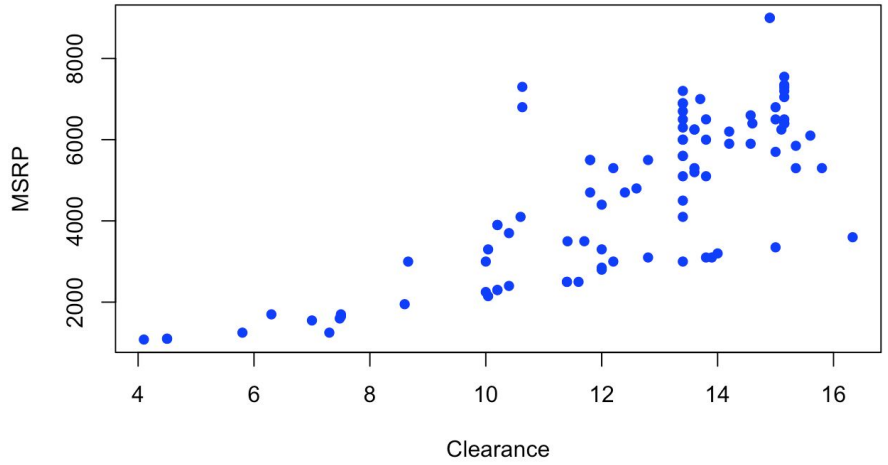
Creating a Histogram

- Create a Histogram of the “Clearance” Variable
  - Run the command: `> hist(Clearance)`
  - If your variables are unnamed, you can also run: `> hist(Motorcycles$Clearance)`
  - You can add colors using the ‘col’ command: `> hist(Clearance, col='orange')`
  - You can add labels using the ‘xlab’ and ‘ylab’ commands:
    - `> hist(Clearance, col ='orange', xlab = 'Clearance', ylab = 'Frequency')`
  - Make sure to use apostrophes after command, and to separate commands using commas



Creating a Scatterplot

- Create a Scatterplot of “MSRP” vs. “Clearance”
  - Run the command: `> plot(MSRP ~ Clearance)`
    - Take note: The variable on the left will appear in the y-axis
  - For a fancy plot, run: `> plot(MSRP ~ Clearance, pch=16, col='blue')`
    - “pch=16” will fill in the circles, and “col=’blue’” will turn your data points blue



Finding Correlation

- There looks to be some linear correlation between MSRP and Clearance
- In order to analyze correlation between the two variables, run the command:
  - `> cor(MSRP, Clearance)`
- What correlation do you find? What does this tell you about the data?

Running a Simple Regression

- Run a regression predicting MSRP from Clearance, and assign the result to a variable called “model1”:
  - Run the command: `> model1 <- lm(MSRP ~ Clearance)`
  - To analyze the regression, run the command: > summary(model1)
- In the readout below, you can analyze the intercept, slope and R^2 Value

```
Call:
lm(formula = MSRP ~ Clearance)

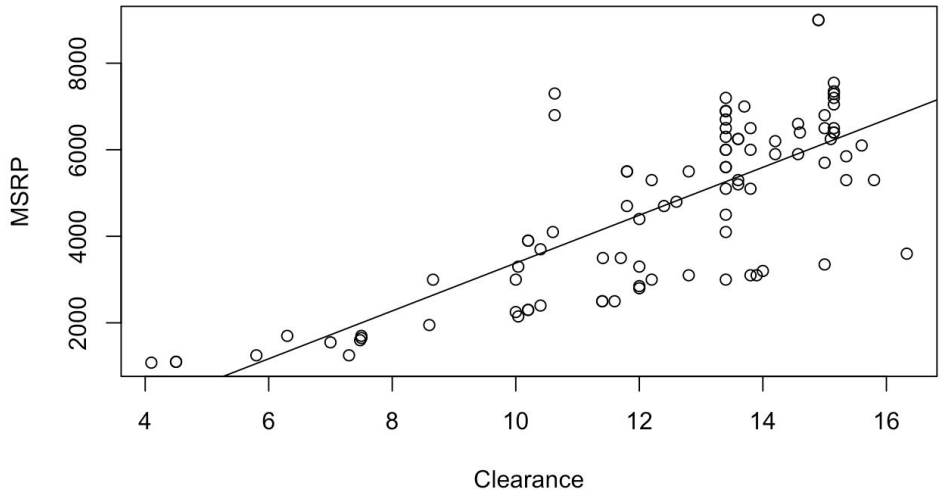
Residuals:
    Min      1Q  Median      3Q     Max
-3286.3  -764.4   166.5   759.2  3568.0

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2149.69     608.41  -3.533 0.000647 ***
Clearance     553.22      48.25  11.466  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1287 on 91 degrees of freedom
Multiple R-squared:  0.591,     Adjusted R-squared:  0.5865
F-statistic: 131.5 on 1 and 91 DF,  p-value: < 2.2e-16
```

Visualizing Your Regression

- Visualize your regression by running the following two commands:
  - `> plot(MSRP ~ Clearance, data = Motorcycles)`
  - `> abline(model1)`
- Running the two commands will draw a line of best fit over a scatterplot of the data



Predictions and Intervals

- Making predictions using our model
  - `predict(model1, list(Clearance = 10.00))`
- Making prediction intervals
  - Confidence Intervals
    - Predicting the mean value of Y for a particular X.
    - Among all motorcycles with clearance of 10.00, on average, what will the MSRP be?
    - `predict(model1, list(Clearance = 10.00), interval = 'confidence')`
  - Prediction Intervals
    - Predicting Y for a single new case
    - For a particular motorcycle with clearance 10.00, what is its MSRP?
    - `predict(model1, list(Clearance = 10.00), interval = 'prediction')`

Checking Regression Assumptions (LINE)

- Linearity
  - `> plot(model1)`
  - Look at the red line (should be relatively horizontal)
- Independence
  - Infer based on question
- Normality of residuals
  - `qqnorm(resid(model1))`
  - `hist(resid(model1))`
  - `Or just do plot(model1)`
- Equal Variance (Homoscedasticity)
  - `plot(model1)`
  - Look at the first plot: Residuals vs fitted