Week2_Practice

ChrisCoussa

10/6/2018

```
###### Monday 8 Oct 2018 ######

library(DevFarn2)
```

```
## Loading required package: qcc
```

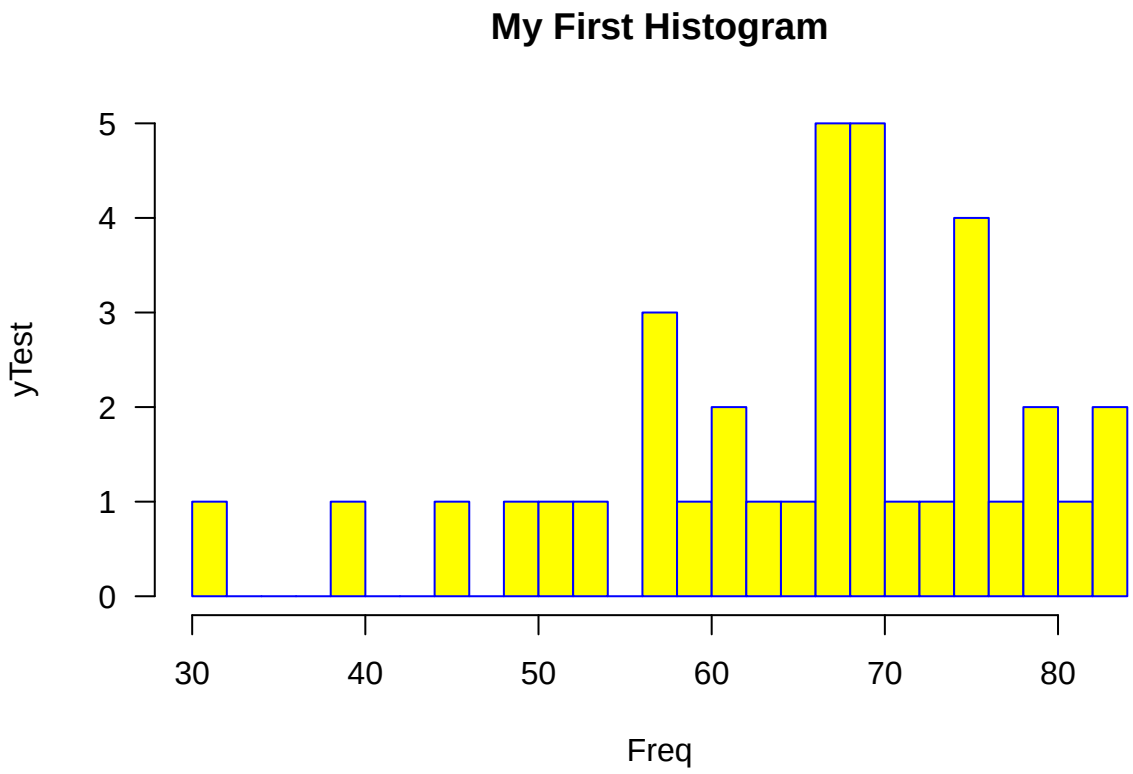
```
## Package 'qcc' version 2.7
```

```
## Type 'citation("qcc")' for citing this R package in publications.
```

```
library(qcc)

# Importing Data_sets, click on the "Import Dataset" in the window to the right. It will
# download the needed packages and you can search for the file you want to import.
# It will show up in the RStudio enviornment as a seperate file.

# After you install qcc, you can install the DevFarn2 file you downloaded form Bb
# Go to install package and switch from package repo to archive, and install
# from there.

data("e1.1")
# Displays the type of file you're working with. With e1.1, it's "numeric".
class(e1.1)
```

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

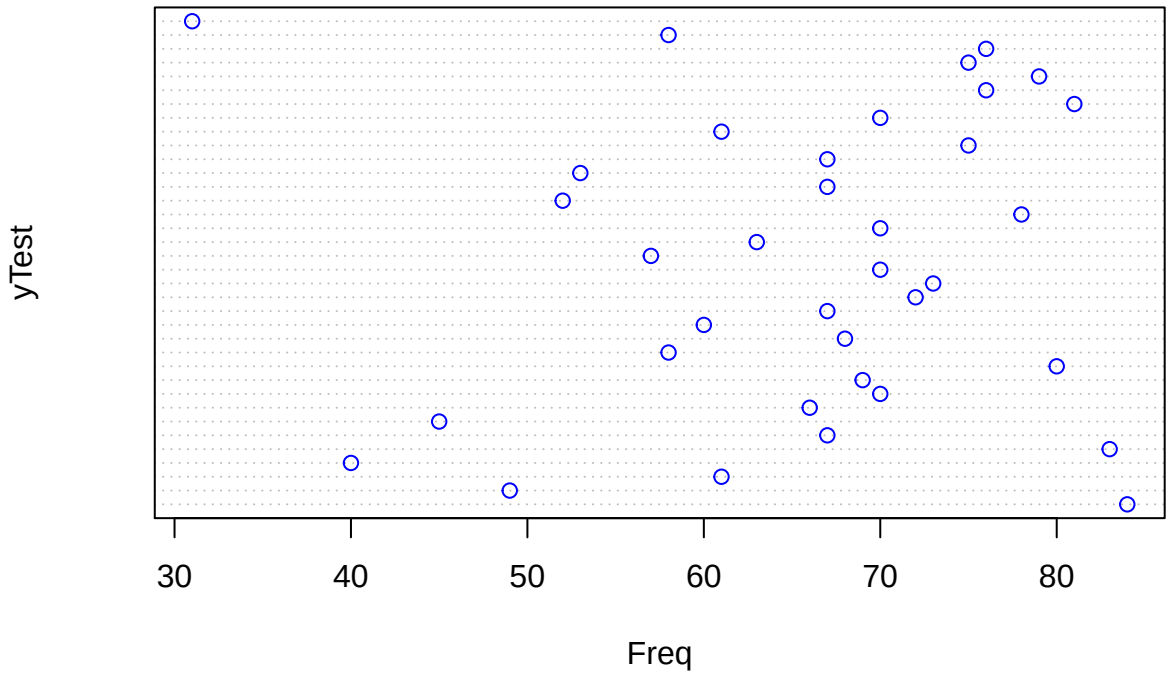
```
# Example Histogram
hist(e1.1, main = "My First Histogram", xlab = "Freq", ylab = "yTest",
border = "blue", col = "yellow", las = 1, breaks = 20)
```




```
stem(e1.1)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   3 | 1
##   4 | 059
##   5 | 23788
##   6 | 01136777789
##   7 | 000023556689
##   8 | 0134
```

```
dotchart(e1.1, main = "Example Dot Plot", xlab = "Freq", ylab = "yTest", col = "blue")
```




```
# Importing another dataset from the book
data("Furnace")
View(Furnace)
# Displays the type of file you're working with. With Furnace, it's a "data.frame".
class(Furnace)
```

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

```
dim(Furnace)
```

```
## [1] 90 10
```

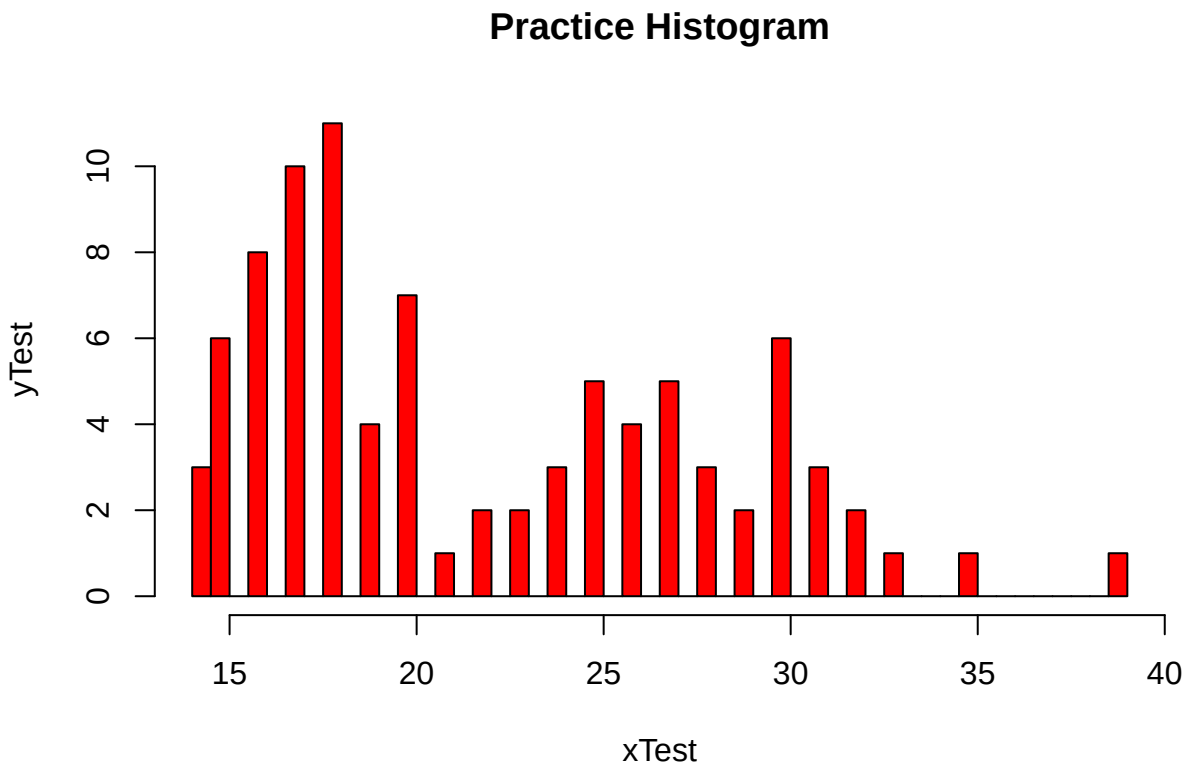
```
head(Furnace)
```

```
##   Type CH.Area CH.Shape CH.HT CH.Liner House Age BTU.In BTU.Out Damper
## 1    1      28        1    20        2     3   8   7.87    8.25      1
## 2    2     144        2    26        0     2  75   9.43    9.66      1
## 3    1      80        3    30        1     2  44   7.16    8.33      1
## 4    2     100        2    24        0     2  75   8.67    8.82      1
## 5    3     168        3    35        1     2  30  12.31   12.06      1
## 6    3      28        1    17        2     3   4   9.84    9.67      1
```

```
tail(Furnace)
```

```
##    Type CH.Area CH.Shape CH.HT CH.Liner House Age BTU.In BTU.Out Damper
## 85    1     144        1    30        0     2  85  10.50   10.77      2
## 86    2     100        2    24        0     2  70  14.35   15.26      2
## 87    2      96        1    17        0     1  40  13.42   14.53      2
## 88    1     100        2    20        1     2  99   6.35    6.84      2
## 89    1     100        2    20        1     1  14   9.83   10.92      2
## 90    1      28        1    28        1     2  55  12.16   13.05      2
```

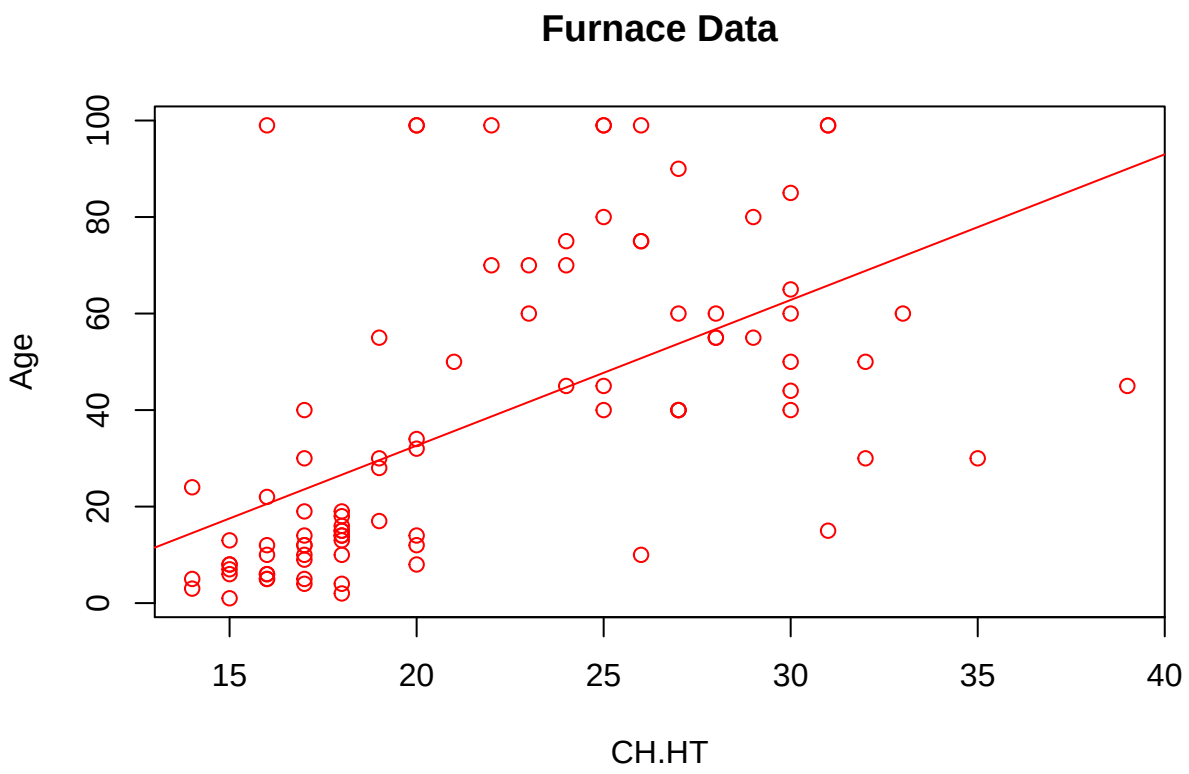
```r
# Histogram of CH.HT column. $CH.HT is the command to tell R to get just the data
# from the CH.HT column
hist(Furnace$CH.HT, main = "Practice Histogram", xlab = "xTest", ylab = "yTest",
     breaks = 50, col = "red")
```




```r
cor(Furnace$CH.HT, Furnace$Age)
```

```
## [1] 0.5751901
```

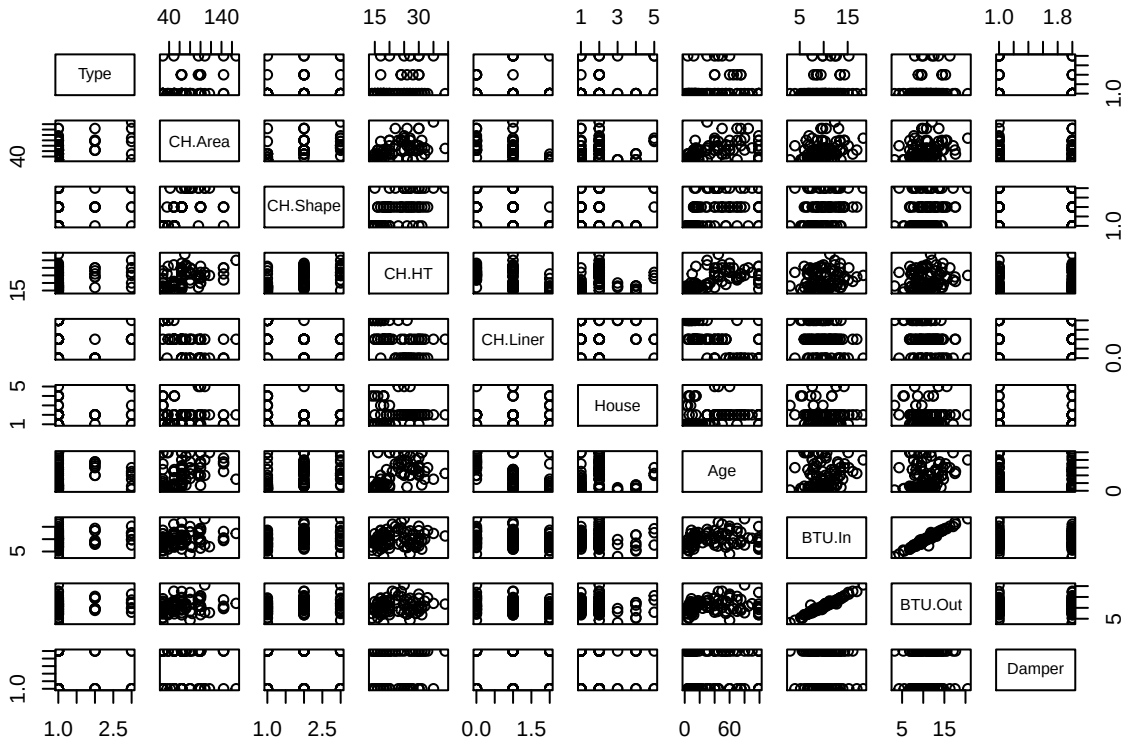
```r
# Regression line (x, y) for Furnace
lm_furnaceModel = lm(Furnace$Age ~ Furnace$CH.HT, data = Furnace)
plot(Furnace$CH.HT, Furnace$Age, main = "Furnace Data", xlab = "CH.HT", ylab = "Age",
     col = "Red")
abline(lm_furnaceModel, col = "red")
```




```
# scatter.smooth(Furnace$CH.HT, Furnace$Age, main = "Furnace Data", xlab = "CH.HT",
      # ylab = "Age", col = "Black")
summary(lm_furnaceModel)
```

```
##
## Call:
## lm(formula = Furnace$Age ~ Furnace$CH.HT, data = Furnace)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.831 -13.758  -9.058  13.133  78.442
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -27.7340    10.4068  -2.665  0.00916 **
## Furnace$CH.HT   3.0182     0.4576   6.596 3.04e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.58 on 88 degrees of freedom
## Multiple R-squared:  0.3308, Adjusted R-squared:  0.3232
## F-statistic: 43.51 on 1 and 88 DF,  p-value: 3.038e-09
```

```
# Pairs displays a matrix of Furnace
pairs(Furnace)
```




```
# Create a vector of data
x = c(12, 7, 3, 4.2, 18, 2, 54, -21, 8, -5)

# Find the mean of the dummy vecotor
xMean = mean(x)
print(xMean)
```

```
## [1] 8.22
```

```
class(x) # Numeric class
```

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

```
# Get the median
median(x)
```

```
## [1] 5.6
```

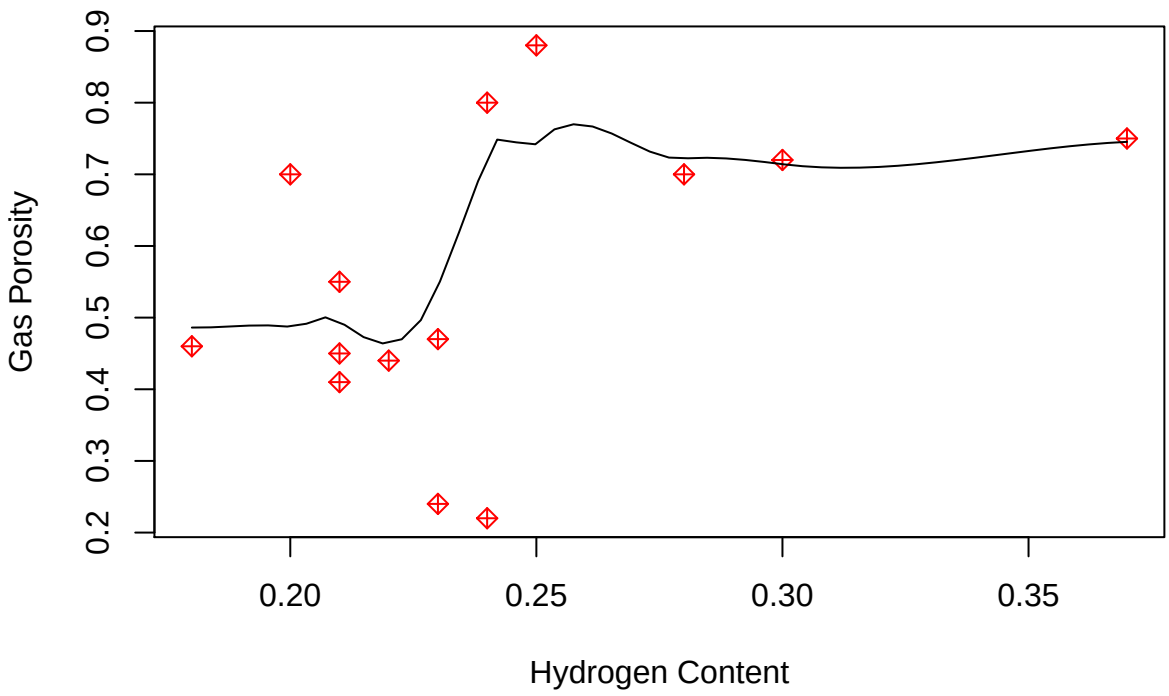
```
###### Example for Linear Model ######
x = c(.18, .20, .21, .21, .21, .22, .23, .23, .24, .24, .25, .28, .30, .37)
y = c(.46, .70, .41, .45, .55, .44, .24, .47, .22, .80, .88, .70, .72, .75)
mean(x)
```

```
## [1] 0.2407143
```

```
mean(y)
```

```
## [1] 0.5564286
```

```
plot(x, y, main = "Scatter Plot Example", xlab = "Hydrogen Content",
      ylab = "Gas Porosity", pch = 9, col = "red")
scatter.smooth(x, y, main = "Scatter Plot Example", xlab = "Hydrogen Content",
      ylab = "Gas Porosity", pch = 9, col = "red")
```




```
sum(x)
```

```
## [1] 3.37
```

```
sum(y)
```

```
## [1] 7.79
```

```
sum(x^2)
```

```
## [1] 0.8419
```

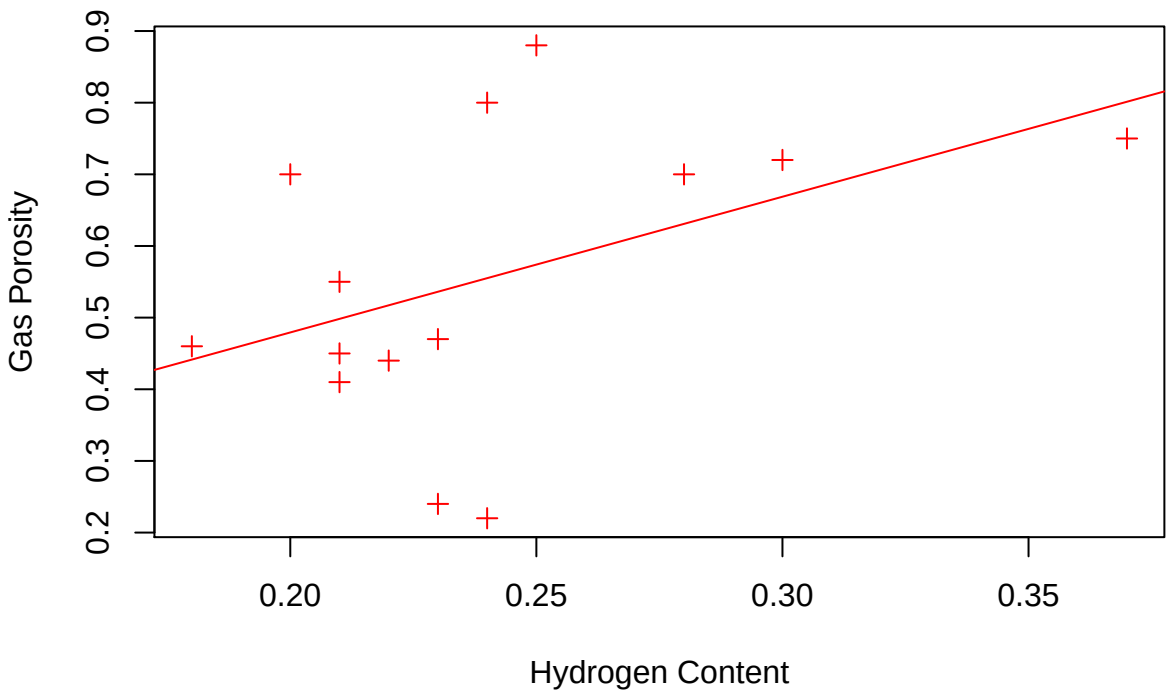
```
sum(y^2)
```

```
## [1] 4.8805
```

```
###### Method for finding Correlation Coefficient (R^2) ######
cor(x, y)
```

```
## [1] 0.4491168
```

```
###### Building a Linear Model for the Data ######
# Converting 2 vectors to a dataframe
dataExample = data.frame(x, y)

# Build a model for linear regression on full dataframe
# y = mx + b ----> b is y intercept
linearModel = lm(y ~ x, data = dataExample)
plot(x, y, main = "Scatter Plot Example", xlab = "Hydrogen Content",
      ylab = "Gas Porosity", pch = 3, col = "red")

# abline fits the linear line for the dataframe in the scatter plot
abline(linearModel, col = "red")
```




```
summary(linearModel)
```

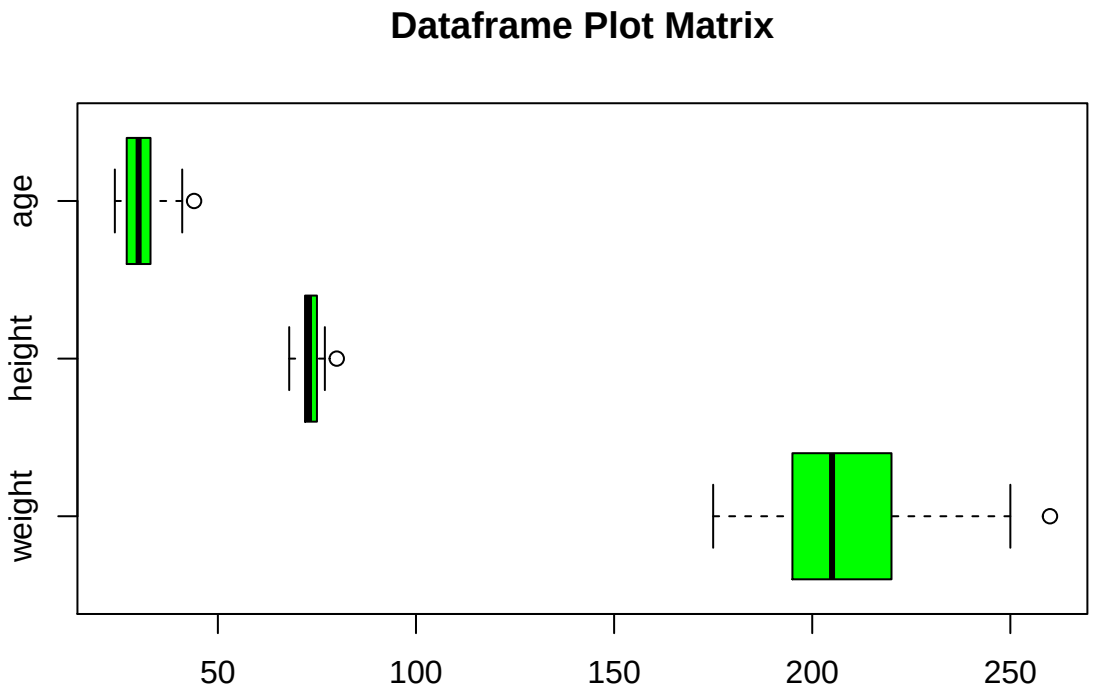
```
##
## Call:
## lm(formula = y ~ x, data = dataExample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33508 -0.07443 -0.01484  0.06481  0.30598
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1005     0.2667   0.377    0.713
## x             1.8941     1.0878   1.741    0.107
##
## Residual standard error: 0.1906 on 12 degrees of freedom
## Multiple R-squared:  0.2017, Adjusted R-squared:  0.1352
## F-statistic: 3.032 on 1 and 12 DF,  p-value: 0.1072
```

```
# From the table below y = mx + b is
# y = 1.89x + 0.10

###### Wednesday 10 Oct 2018 ######
# Multipe Linear Regression with 3 variables
weight = c(226, 250, 180, 205, 200, 215, 200, 180, 180, 182, 248, 260, 210, 225, 205, 195,
      175, 220, 195, 210, 195, 200, 210, 220, 185)
height = c(75, 77, 73, 75, 71, 72, 74, 72, 72, 73, 75, 80, 75, 75, 75, 75, 68, 74, 71, 73,
       71, 74, 72, 70, 72)
age = c(30, 40, 28, 25, 27, 40, 31, 41, 24, 26, 44, 28, 29, 26, 34, 32, 29, 25, 28, 26,
       31, 34, 33, 33, 30)
```

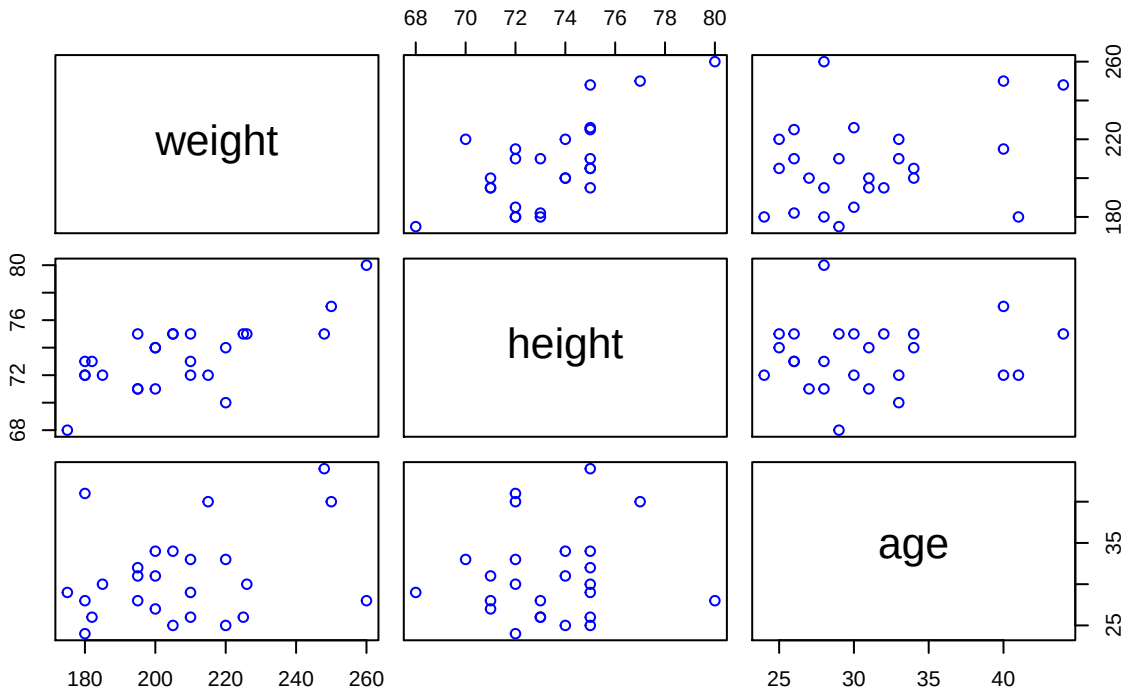
```r
# Create a dataframe from the 3 vectors
my_data_frame = data.frame(weight, height, age)
# View(my_data_frame)
# Buid the linear regression model on the three var dataframe
lm_model2 = lm(weight ~ height + age, data = my_data_frame)
summary(lm_model2)$r.squared
```

```
## [1] 0.5434694
```

```r
# R^2 (corrilation coefficient) = 0.5434694
# y = mx + b from the table summary ----->  y-hat = 6.2351(x1) + 1.0739(x2) - 283.8123
# Coefficient of the determination

###### Visualization practice ######
boxplot(my_data_frame,  main = "Dataframe Plot Matrix", col = "green", horizontal = TRUE)
```




```r
plot(my_data_frame, main = "Dataframe Plot Matrix", col = "blue")
```




```
###### Pie Charts ######
slices = c(10, 12, 4, 16, 8)
lbls = c("US", "UK", "Australia", "Germany", "France")
pie(slices, labels = lbls, main = "Pie Chart of Countries")
```