Week3_Practice

ChrisCoussa

10/15/2018

```
###### Week 3 Session 1 ######
library(qcc)
```

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

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

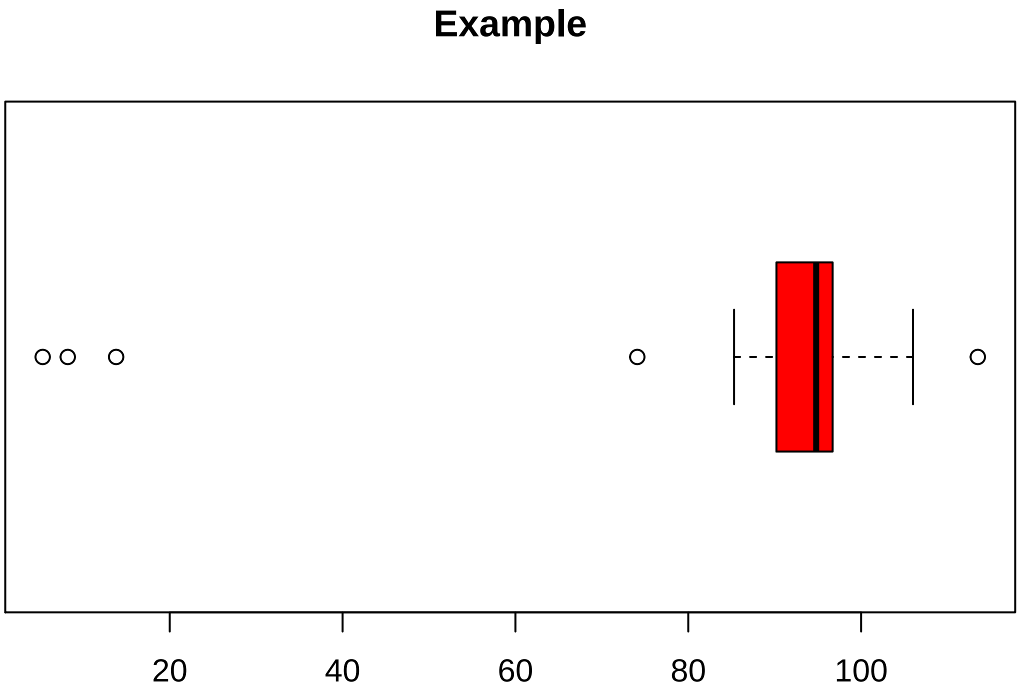
```
library(DevFarn2)
library(readxl)

# Importing a data frame from the DevFarn2 examples
data(e2.16)
# Summary of the e2.16 data frame (descriptive analysis of the data, range is max - min)
summary(e2.16)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.30   90.20   94.80   84.86   96.70  113.50
```

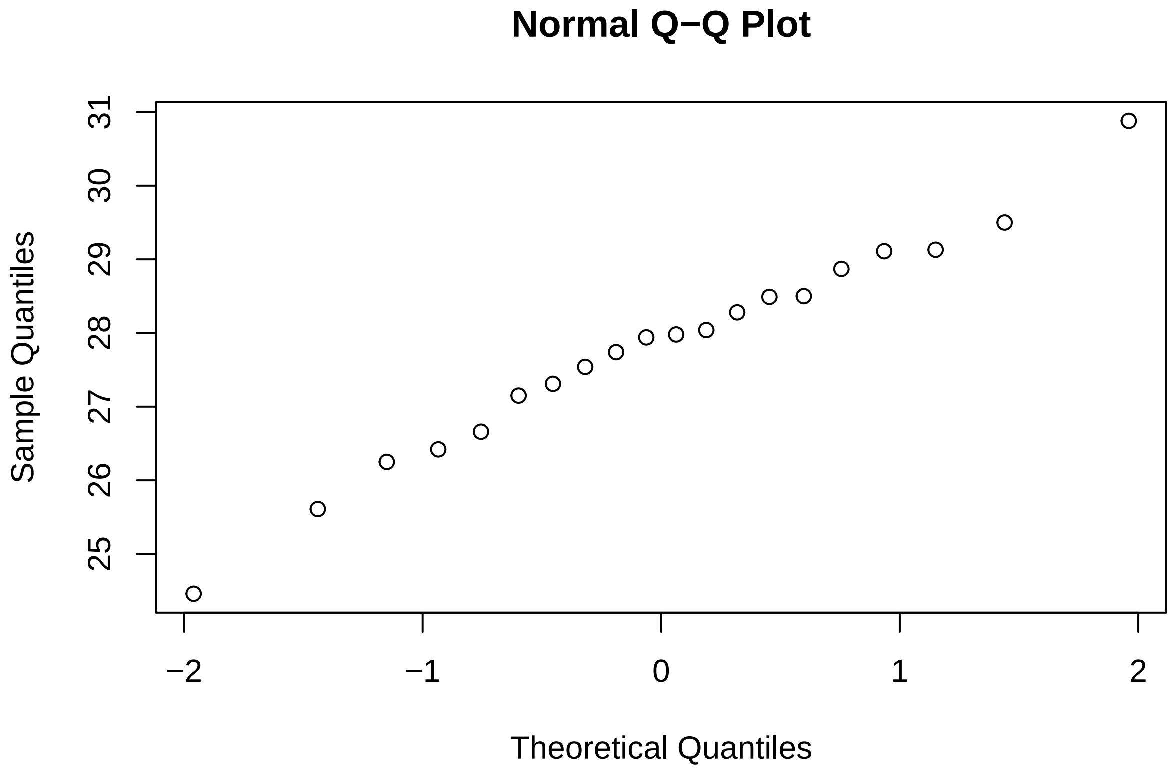
```
# View the data frame in a separate window
# View(e2.16)

boxplot(e2.16, main = "Example", col = "red", horizontal = TRUE)
```



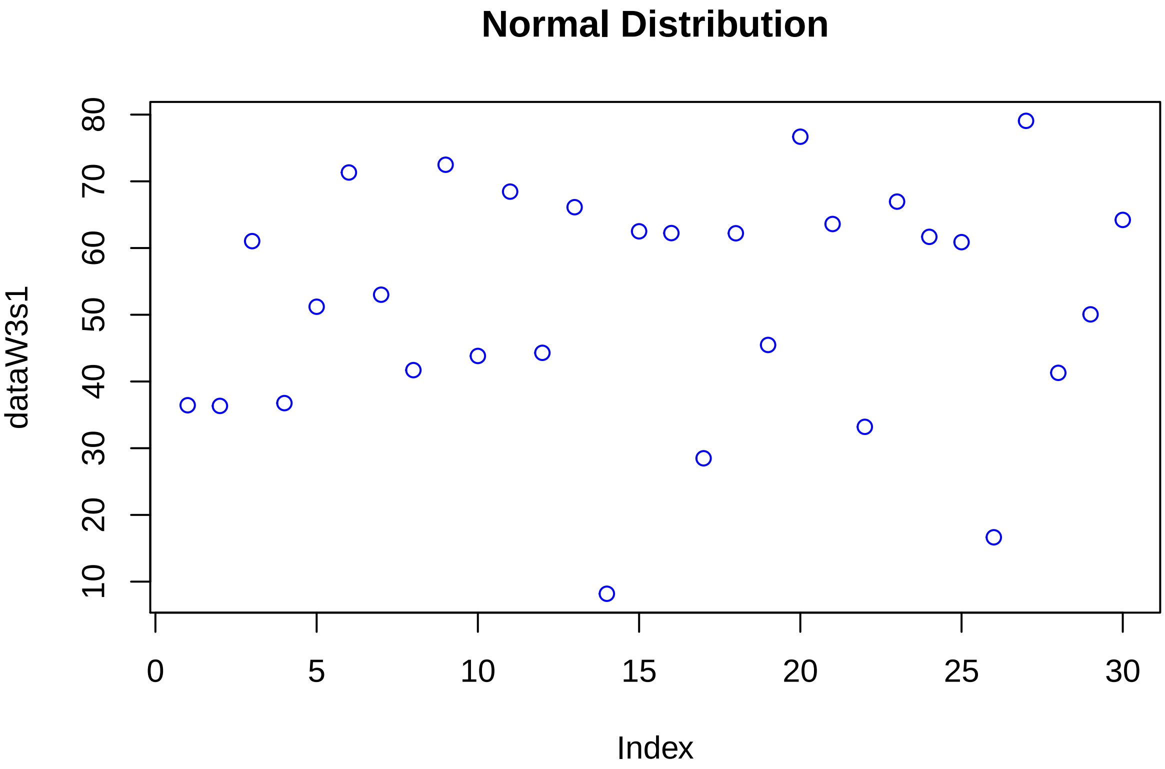
```
# Boxplot shows 3 possible outliers at the bottom of the plot, under 20.

###### Quantile-Quantile Plots (qq plot) ######
# If you have a Normal Distribution of variables, take the qq plot of the data it
# is a straight line. 2 variable dataset
data(e2.18)
qqnorm(e2.18$X1) # the $x1 means that you are going into the e2.18 dataset and getting
```

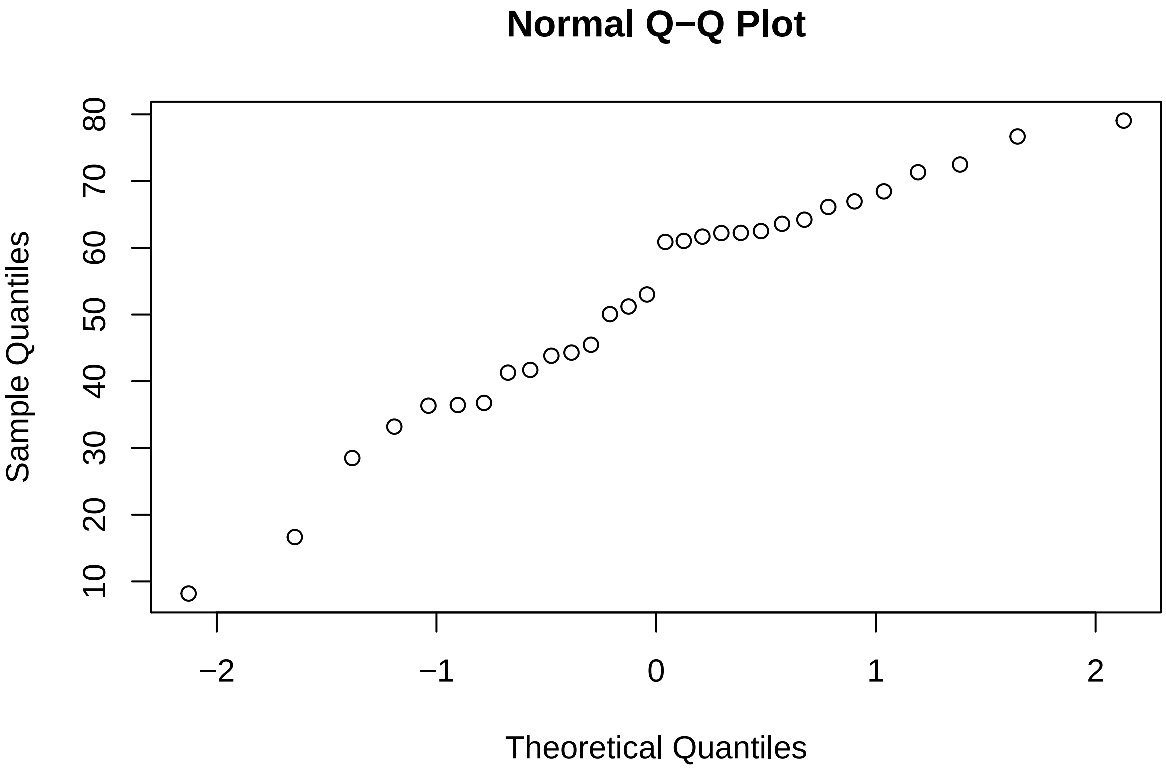


```
# the data for for the x1 variable

###### Creating a data frame with Normal Distrubution using rnorm() ######
dataW3s1  = rnorm(30, mean = 50, sd = 20)
plot(dataW3s1, main = "Normal Distribution", col = "blue")
```



```
# Given that the data frame dataW2s1 has Normal Distribution, the qq plot should be straight
qqnorm(dataW3s1)
```



```
###### Working with data of three variables (x1, x2, and y)
data(e3.15)
```

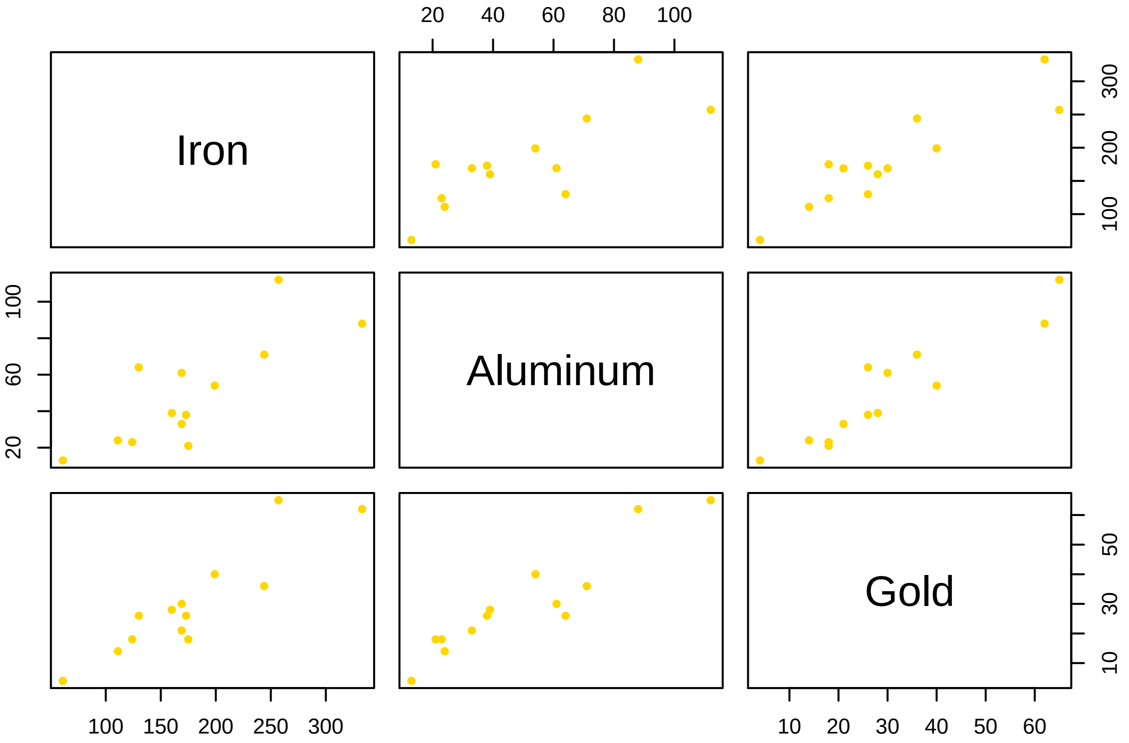
```
summary(e3.15)
```

```
##        x1              x2               y
##  Min.   : 61.0   Min.   : 13.00   Min.   : 4.00
##  1st Qu.:130.0   1st Qu.: 24.00   1st Qu.:18.00
##  Median :169.0   Median : 39.00   Median :26.00
##  Mean   :177.3   Mean   : 49.31   Mean   :29.85
##  3rd Qu.:199.0   3rd Qu.: 64.00   3rd Qu.:36.00
##  Max.   :333.0   Max.   :112.00   Max.   :65.00
```

```
dim(e3.15)
```

```
## [1] 13  3
```

```
# Create a scatter matrix for all three data sets of the data frame
# Use labels to change the labels of the plots
pairs(e3.15, labels = c("Iron", "Aluminum", "Gold"), col = "gold", pch = 20)
```



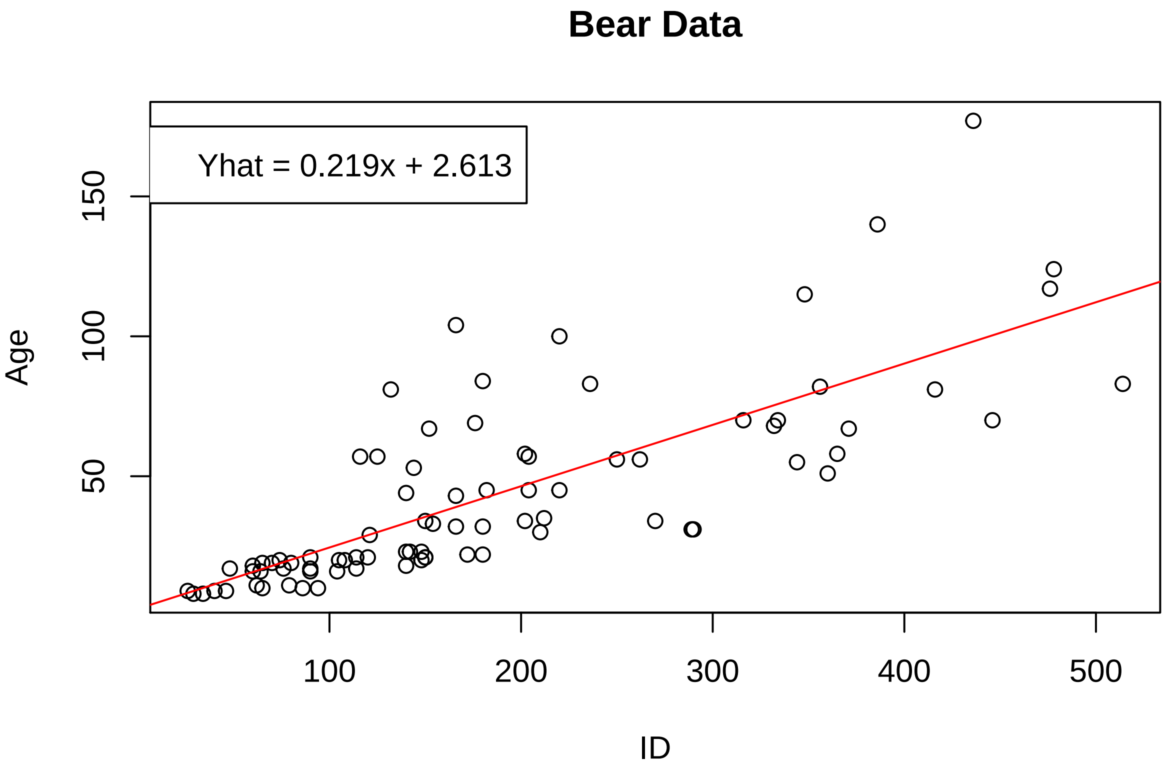
```
data("Bears")
# str shows internal structure of the dataset, alternative to summary
str(Bears)
```

```
## 'data.frame':    143 obs. of  12 variables:
##  $ Name   : Factor w/ 99 levels "","Abe","Adam",..: 6 9 9 9 9 18 18 24 24 72 ...
##  $ ID     : int  39 41 41 41 41 43 43 45 45 48 ...
##  $ Age    : int  19 19 20 23 29 19 20 55 67 81 ...
##  $ Month  : int  7 7 8 11 5 7 8 7 7 9 ...
##  $ Sex    : int  1 2 2 2 2 1 1 1 1 1 ...
##  $ Head.L : num  10 11 12 12.5 12 11 12 16.5 16.5 15.5 ...
##  $ Head.W : num  5 6.5 6 5 6 5.5 5.5 9 9 8 ...
##  $ Neck.G : num  15 20 17 20.5 18 16 17 28 27 31 ...
##  $ Length : num  45 47.5 57 59.5 62 53 56 67.5 78 72 ...
##  $ Chest.G: num  23 24 27 38 31 26 30.5 45 49 54 ...
##  $ Weight : int  65 70 74 142 121 80 108 344 371 416 ...
##  $ Obs.No : int  1 1 2 3 4 1 2 1 2 1 ...
```

```r
# Create a linear model from the Bears dataframe using 2 variables
plot(Bears$Weight, Bears$Age, main = "Bear Data", xlab = "ID", ylab = "Age",
     col = "black")
# REMEBER: When you created a linear model you have to put your y variable first!
# So reverse the order from your plot in order for the linear line to be correct.
lmBears = lm(Bears$Age ~ Bears$Weight, data = Bears)
summary(lmBears)
```

```
##
## Call:
## lm(formula = Bears$Age ~ Bears$Weight, data = Bears)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.159 -11.751  -3.693   2.373  78.849
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.61341    4.40772   0.593    0.555
## Bears$Weight  0.21912    0.01992  11.001   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.68 on 81 degrees of freedom
##   (60 observations deleted due to missingness)
## Multiple R-squared:  0.5991, Adjusted R-squared:  0.5941
## F-statistic:   121 on 1 and 81 DF,  p-value: < 2.2e-16
```

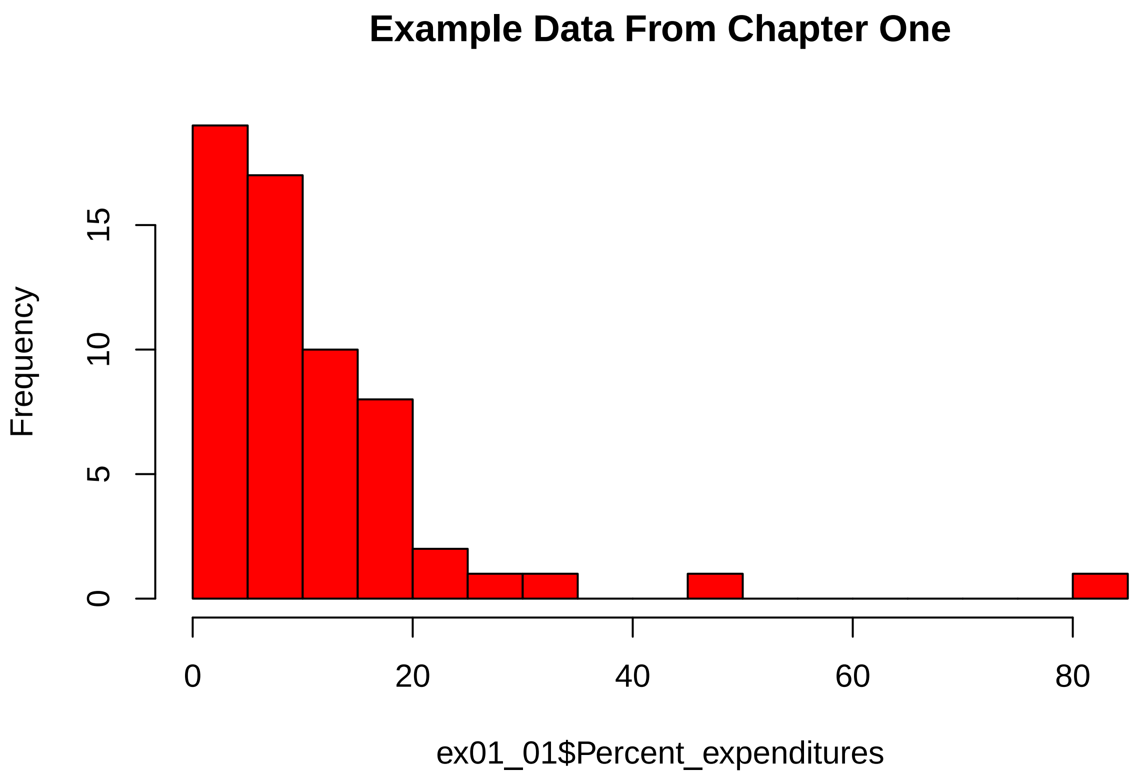
```r
# Yhat = 0.219x + 2.613
abline(lmBears, col = "red")
legend(x = 175, y=NULL, legend = "Yhat = 0.219x + 2.613", col = "black", bg = "white")
```



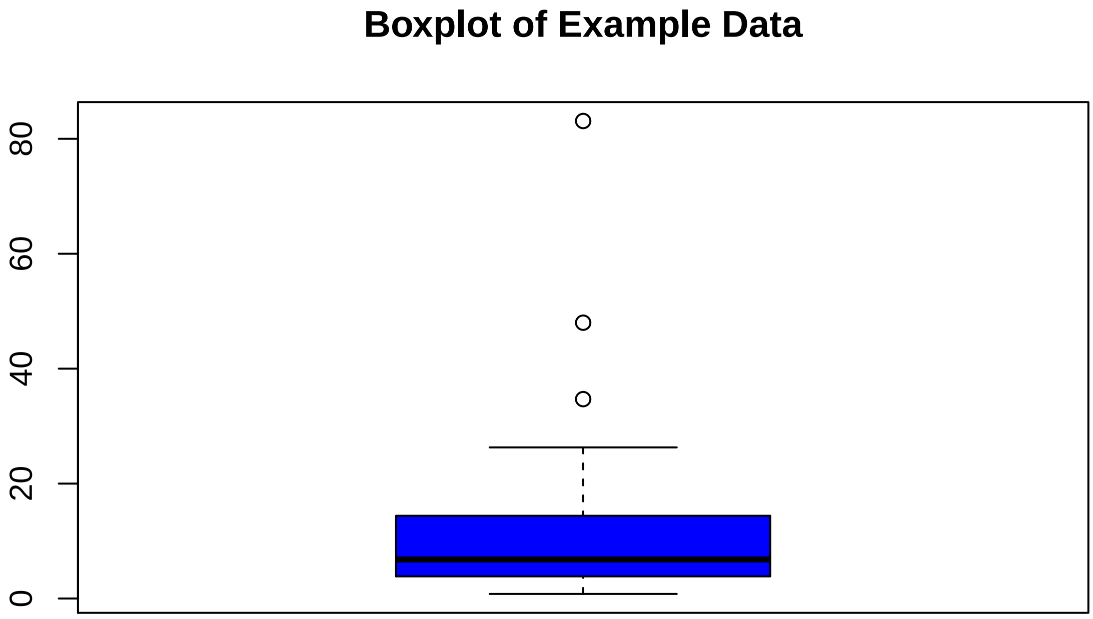
```
###### Week 3 Day 6 ######

# Working with data imprted from an Excel file

# Bringing in the data from the excel file requires the redxl library and the file path
ex01_01 <- read_excel("/Users/ccoussa717/Desktop/College Classes/National University/CSC220_AppliedStat
hist(ex01_01$Percent_expenditures, main = "Example Data From Chapter One", breaks = 20, col = "red")
```



```
boxplot(ex01_01, main = "Boxplot of Example Data", col = "blue")
```



```
sd(ex01_01$Percent_expenditures) # Standard Deviation is 12.7411
```

```
## [1] 12.7411
```

```
mean(ex01_01$Percent_expenditures) # Mean is 10.89167
```

```
## [1] 10.89167
```

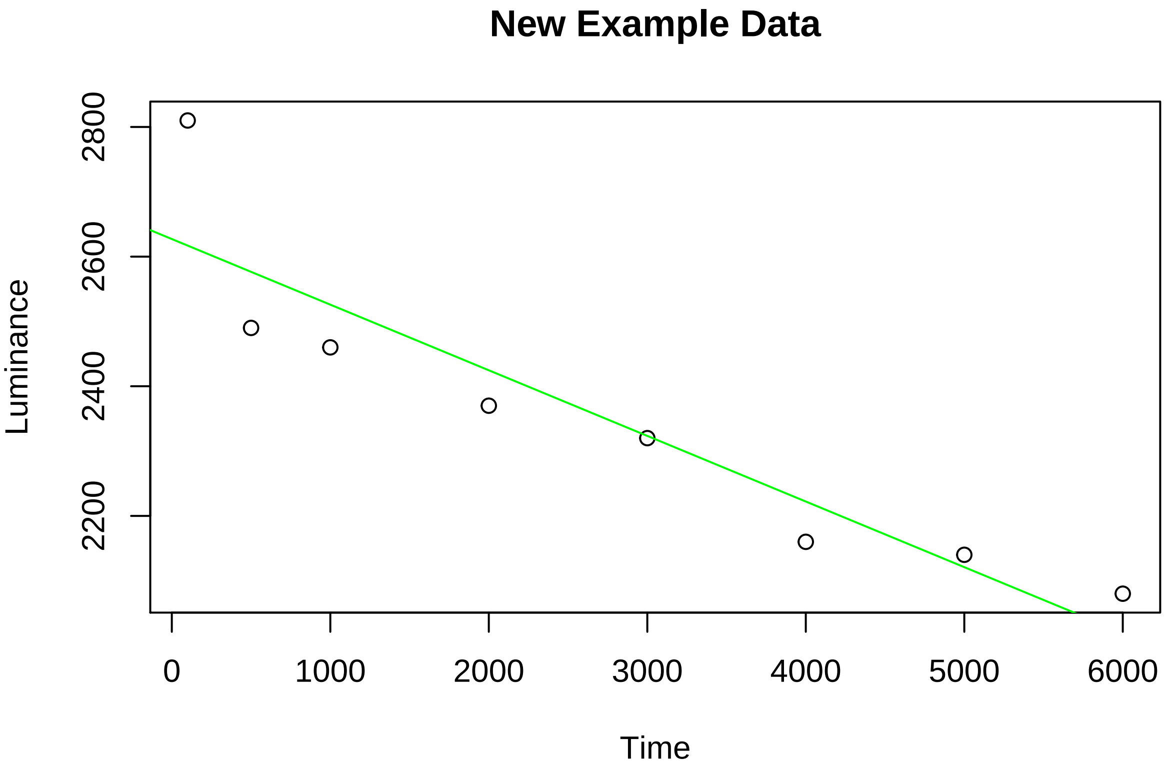
```
var(ex01_01$Percent_expenditures) # Variance is 162.3357
```

```
## [1] 162.3357
```

```r
ex01_04 <- read_excel("/Users/ccoussa717/Desktop/College Classes/National University/CSC220_AppliedStats
linearModelEx04 = lm(ex01_04$Luminance ~ ex01_04$Time, data = ex01_04)
plot(ex01_04, main = "New Example Data")
summary(linearModelEx04) # Yhat = -8.388x + 22443.468
```

```
##
## Call:
## lm(formula = ex01_04$Luminance ~ ex01_04$Time, data = ex01_04)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -86.49 -63.06 -29.00  29.42 193.01
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2627.11066   58.81707  44.666 8.43e-09 ***
## ex01_04$Time   -0.10124    0.01741  -5.814  0.00114 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 99.95 on 6 degrees of freedom
## Multiple R-squared:  0.8492, Adjusted R-squared:  0.8241
## F-statistic:  33.8 on 1 and 6 DF,  p-value: 0.001137
```

```r
abline(linearModelEx04, col = "green")
```


New Example Data
Luminance
Time