Assignment1

Tristan De Alwis

9/3/2019

Short Answers:

Q: About how long did this assignment take you? Did you feel it was too long, too short, or reasonable? A: ~16 Hours. I felt it was reasonable with the pace of the class.

Q: Who, if anyone, did you work with on this assignment? A: No one, but I was asked for help by other students

Q: What questions do you have relating to any of the material we have covered so far in class? A: I don't believe I understand Q-Q plots correctly, or to interprete different visualizations. Can we go over that again? I'd also like to better understand how we might be asked to compute certain things (i.e. histograms, transformations, Q-Q plots) we did here on an exam?

Loading Neccessary Libraries

```
suppressPackageStartupMessages(library(vcd))
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(moments))
suppressPackageStartupMessages(library(MASS))
suppressPackageStartupMessages(library(reshape))
```

Question 1

a

Loading in Dataset

```
waves <- read.csv("waves.csv")
```

Summarize Data

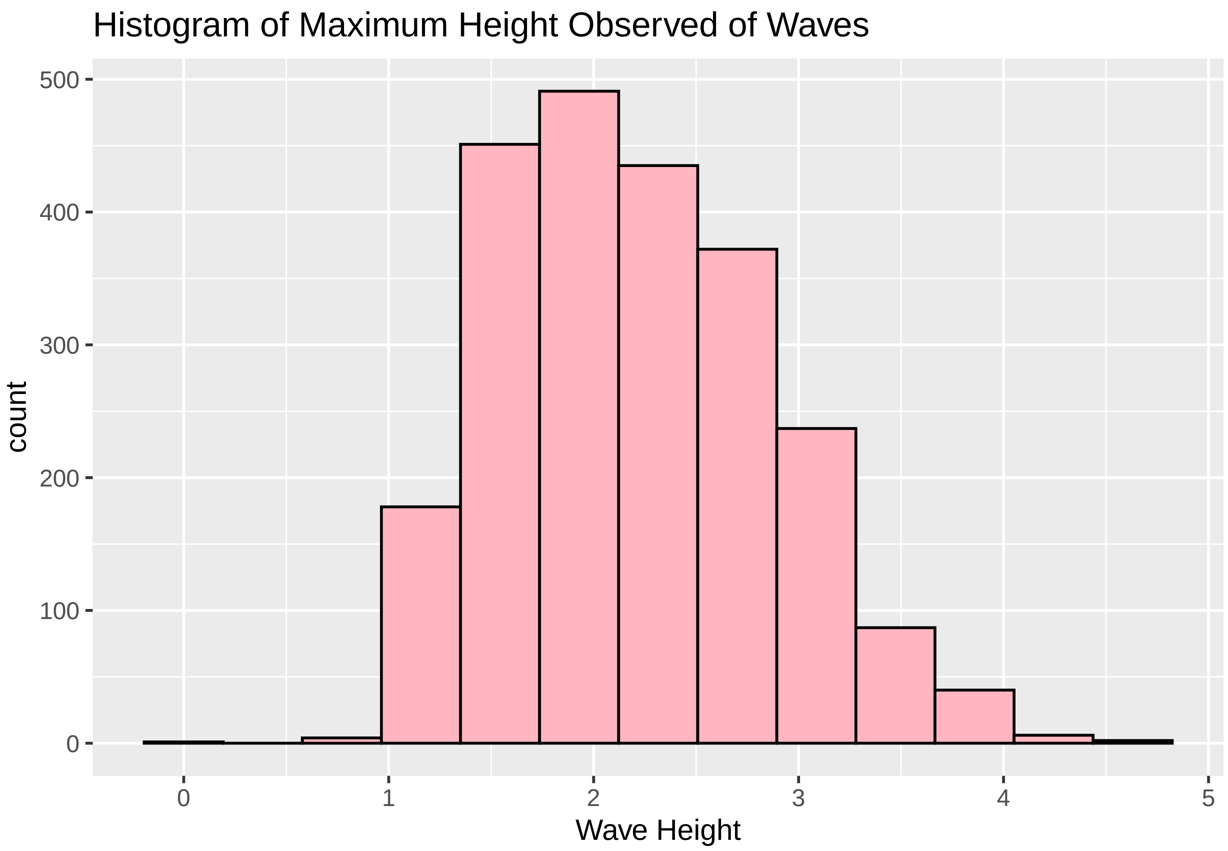
```
summary(waves)
```

```
##    Date.Time          Hs             Hmax             Tz       
##  10/1/19:  48   Min.   :0.160   Min.   :0.190   Min.   :3.693  
##  10/2/19:  48   1st Qu.:1.000   1st Qu.:1.680   1st Qu.:5.296  
##  10/3/19:  48   Median :1.298   Median :2.150   Median :5.824  
##  10/4/19:  48   Mean   :1.309   Mean   :2.208   Mean   :5.877  
##  10/5/19:  48   3rd Qu.:1.579   3rd Qu.:2.660   3rd Qu.:6.416  
##  10/6/19:  48   Max.   :2.464   Max.   :4.820   Max.   :8.663  
##  (Other):2016                                                  
##        Tp         Peak.Direction        SST       
##  Min.   : 4.051   Min.   : 19.00   Min.   :20.65  
##  1st Qu.: 7.757   1st Qu.: 87.00   1st Qu.:23.95  
##  Median : 8.852   Median : 94.00   Median :25.65  
##  Mean   : 9.091   Mean   : 94.85   Mean   :24.96  
##  3rd Qu.:10.409   3rd Qu.:104.00   3rd Qu.:26.30  
##  Max.   :14.795   Max.   :157.00   Max.   :28.10  
## 
```

b

Creating Histogram of Hmax w/ title, axes label, and 13 bins.

```
# hist1 <- hist(waves$Hmax, main = 'Histogram of Maximum Height \n Observed
# of Waves', xlab = 'Wave Height', breaks=13, col = 'light pink') hist1

# abline(v = mean(waves$Hmax), col='Blue') text(mean(waves$Hmax)+.3, 600,
# substitute(paste(bar(x), '=', m), list(m = round(mean(waves$Hmax),3))),
# col='Blue')

hist1 <- ggplot(waves, aes(x = Hmax)) + geom_histogram(bins = 13, fill = "light pink",
    color = "1") + labs(title = "Histogram of Maximum Height Observed of Waves",
    x = "Wave Height")

hist1
```



The Histogram has the characteristic of One Center, a bell curve slightly skewed left, and the spread is short (.190 to 4.820).

c

How many bins does Sturges' formula suggest we use for the histogram in part b? Show your work.

```
# Sturges Formula: k = log_2(n) + 1

k1 <- log2(2304) + 1
cat("Surges calculation: ", ceiling(k1), "\n")  #Rounds up to next whole number (a bin must be quantize
```

```
## Surges calculation:  13
```

```r
k2 <- nclass.Sturges(waves$Hmax)  #Double checking
cat("Sturges function: ", k2, "\n")
```

```
## Sturges function:  13
```

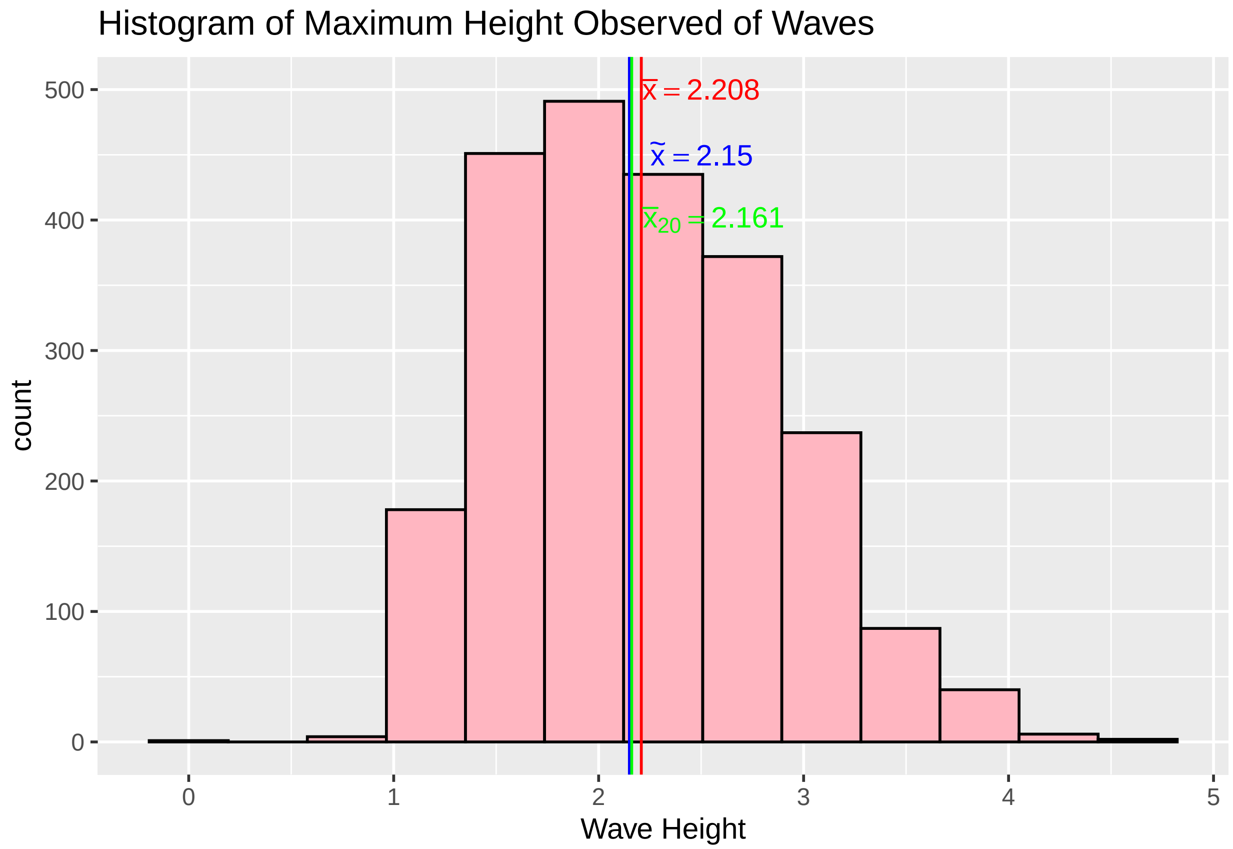
d

Calculate the mean (red), median (blue), and 20% trimmed mean (green) of the maximum wave height. Report the mean, median, and 20% trimmed mean on the histogram.

```r
# Mean visualization
hist1 <- hist1 + geom_vline(aes(xintercept = mean(Hmax)), color = "red") + annotate("text",
    x = 2.5, y = 500, label = paste("bar(x)==", round(mean(waves$Hmax), 3)),
    parse = T, color = "red")

# Median visualization
hist1 <- hist1 + geom_vline(aes(xintercept = median(Hmax)), color = "blue") +
    annotate("text", x = 2.5, y = 450, label = paste("tilde(x)==", round(median(waves$Hmax),
        3)), parse = T, color = "blue")

# 20% trimmed mean visualization
hist1 <- hist1 + geom_vline(aes(xintercept = mean(waves$Hmax, trim = 0.2), 3),
    color = "green") + annotate("text", mean(waves$Hmax, trim = 0.2) + 0.4,
    y = 400, label = paste("bar(x)[20]==", round(mean(waves$Hmax, trim = 0.2),
        3)), parse = T, color = "green")
```

```
## Warning: Ignoring unknown aesthetics: x
```

```r
hist1
```



e

Calculate and report the 25th and 75th percentiles.

```
quantile(waves$Hmax, c(0.25, 0.75))
```

```
##  25%  75%
## 1.68 2.66
```

f

Calculate and report the interquartile range.

```
cat("IQR:       ", 2.66 - 1.68, "\n")
```

```
## IQR:        0.98
```

```
cat("IQR func:  ", IQR(waves$Hmax), "\n")
```

```
## IQR func:   0.98
```

g

Calculate and report the standard span, the lower fence, and the upper fence.

```
# Standard span = 1.5 × IQR OR 1.5 × (Q3 - Q1)
cat("Standard Span: ", 1.5 * IQR(waves$Hmax), "\n")
```

```
## Standard Span:  1.47
```

```r
# Lower Fence = Q1 - (1.5 * IQR).
lower_fence <- quantile(waves$Hmax, 0.25) - 1.5 * IQR(waves$Hmax)
cat("Lower Fence:   ", lower_fence, "\n")
```

```
## Lower Fence:    0.21
```

```r
# Upper Fence = Q3 + (1.5 * IQR)
upper_fence <- quantile(waves$Hmax, 0.75) + 1.5 * IQR(waves$Hmax)
cat("Upper Fence:   ", upper_fence, "\n")
```

```
## Upper Fence:    4.13
```

h

Are there any outliers? Subset the outlying points.

```r
# Upper outliers
paste("Upper Outliers: ", waves[waves$Hmax >= upper_fence, ])
```

```
## [1] "Upper Outliers:  c(21, 21, 33, 33)"
## [2] "Upper Outliers:  c(2.454, 2.311, 1.988, 2.054)"
## [3] "Upper Outliers:  c(4.5, 4.82, 4.18, 4.26)"
## [4] "Upper Outliers:  c(7.034, 6.829, 5.979, 5.47)"
## [5] "Upper Outliers:  c(10.347, 11.418, 7.759, 7.312)"
## [6] "Upper Outliers:  c(92, 87, 97, 85)"
## [7] "Upper Outliers:  c(25.7, 25.7, 26.85, 26.8)"
```

```r
# Lower Outliers
paste("Lower Outliers: ", waves[waves$Hmax <= lower_fence, ])
```

```
## [1] "Lower Outliers:  25"    "Lower Outliers:  0.16"
## [3] "Lower Outliers:  0.19"  "Lower Outliers:  3.693"
## [5] "Lower Outliers:  9.452" "Lower Outliers:  121"
## [7] "Lower Outliers:  21.9"
```

i

Calculate and report the variance, standard deviation, and coefficient of variation of the maximum wave heigt.

```r
# Variance:
cat("Variance:      ", var(waves$Hmax), "\n")
```

```
## Variance:       0.4208286
```

```r
# Standard Deviation:
cat("Standard Dev:  ", sd(waves$Hmax), "\n")
```

```
## Standard Dev:   0.648713
```

```r
# Coefficient of Variation:
cat("Coef. of Var.: ", sd(waves$Hmax)/mean(waves$Hmax), "\n")
```

```
## Coef. of Var.:  0.2937797
```

j

Calculate and report the skewness. Comment on this value and how it matches with what you visually see in the histogram.

```r
# Calculating Skewness
skew1 <- skewness(waves$Hmax)
cat("Skewness: ", skew1, "\n")
```

```
## Skewness:  0.4493749
```

```r
if (skew1 > 0) {
    cat("Dataset is slighty skewed right")
} else {
    cat("Dataset is slighty skewed left")
}
```

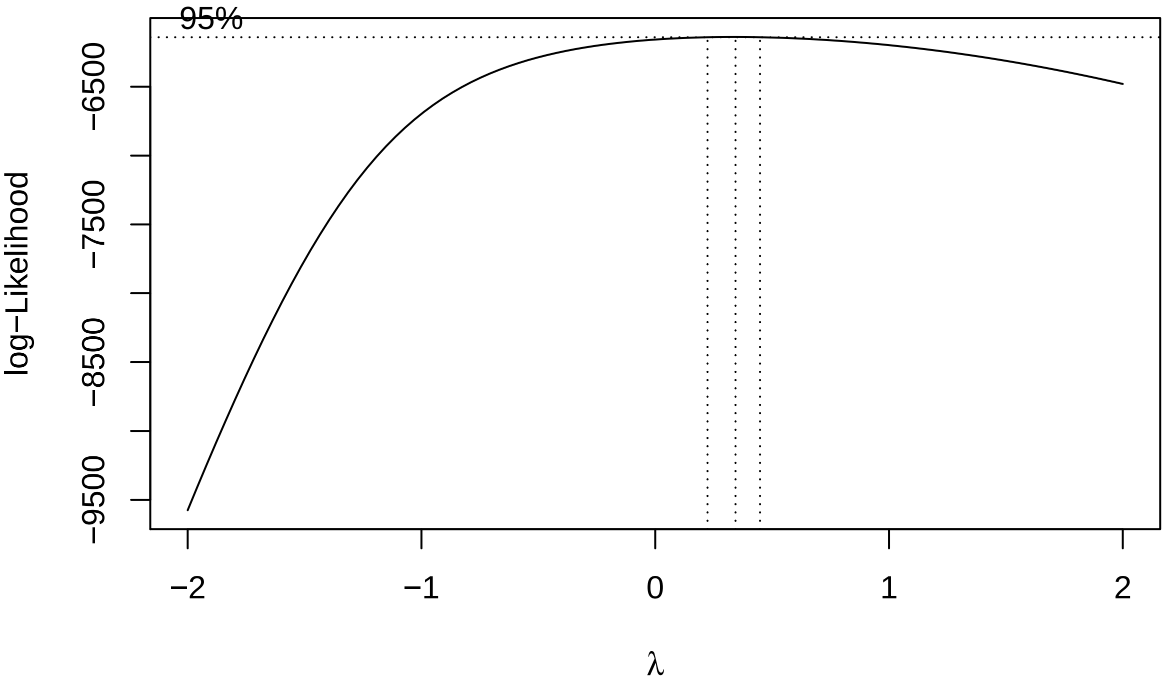
```
## Dataset is slighty skewed right
```

Although I am confident in the mathematical results, I cannot confidently agree nor disagree with the visualization. I think it is hard to see the skewness to the right because the mean falls slightly less than the midpoint (2.15) of the data set, and the frequency of the values are squshed so close together with a small tail to the right.

k

Use a Box-Cox power transformation to appropriately transform the data. In particular, use the boxcox() function in the MASS library. Report the recommended transformation. Do not apply this transformation to the data yet. (Note: the boxcox function automatically produces a plot. You do NOT need to make this in ggplot2.)

```r
x <- waves$Hmax
bc1 <- boxcox(x ~ 1)
```



```r
lambda <- bc1$x[bc1$y == max(bc1$y)]
cat("Reccomended Box Cox Transformation is: ", lambda)
```

```
## Reccomended Box Cox Transformation is:  0.3434343
```

l

Apply the exact Box-Cox recommended transformation (rounded to three decimal places) to the data (this transformation is hereon referred to as the Box-Cox transformed data). Use the summary() function to summarize the results of this transformation.

```
waves$bcdata <- round((waves$Hmax^lambda - 1)/lambda, 3)

summary(waves$bcdata)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -1.266   0.568   0.876   0.873   1.163   2.086
```

m

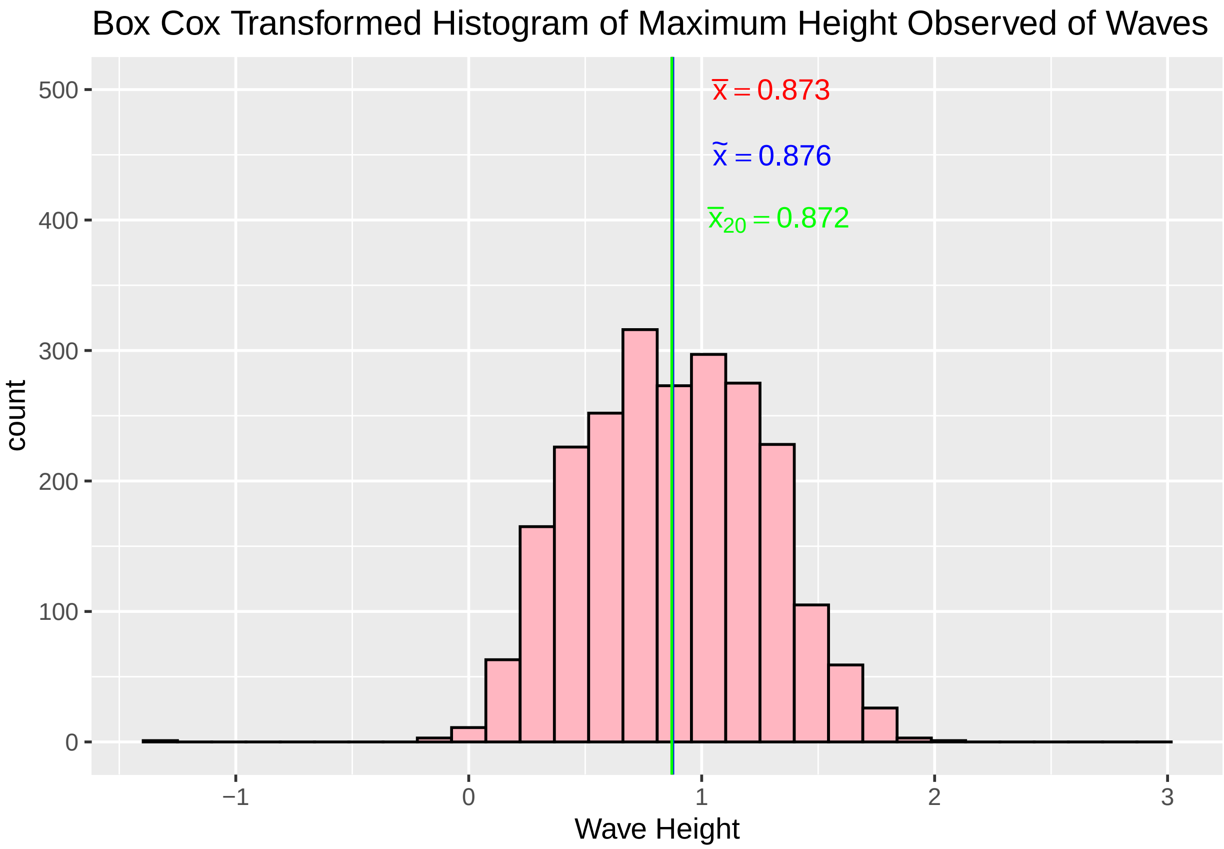
Create a histogram of the Box-Cox transformed data. On this histogram, report the mean, median, and 20% trimmed mean using the same formatting options as in part d above. Comment on the center shape and spread.

```
# hist(waves$transdata)

hist2 <- ggplot(waves, aes(bcdata)) + geom_histogram(fill = "light pink", color = "1") +
    labs(title = "Box Cox Transformed Histogram of Maximum Height Observed of Waves",
        x = "Wave Height")

# Mean visualization
hist2 <- hist2 + geom_vline(aes(xintercept = mean(waves$bcdata)), color = "red") +
    annotate("text", x = 1.3, y = 500, label = paste("bar(x)==", round(mean(waves$bcdata),
        3)), parse = T, color = "red")

# Median visualization
hist2 <- hist2 + geom_vline(aes(xintercept = median(waves$bcdata)), color = "blue") +
    annotate("text", x = 1.3, y = 450, label = paste("tilde(x)==", round(median(waves$bcdata),
        3)), parse = T, color = "blue")

# 20% trimmed mean visualization
hist2 <- hist2 + geom_vline(aes(xintercept = mean(waves$bcdata, trim = 0.2),
    3), color = "green") + annotate("text", mean(waves$bcdata, trim = 0.2) +
    0.46, y = 400, label = paste("bar(x)[20]==", round(mean(waves$bcdata, trim = 0.2),
    3)), parse = T, color = "green")
```

```
## Warning: Ignoring unknown aesthetics: x
```

```
hist2
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Here, the shape is one-center and the spread is more bell-shpaed and is no longer left-skewed. It is also worth noting the mean, median, and trimmed mean are much closer in value now.

n

As an alternative to the Box-Cox transformation, let's also use a square root transformation. Apply the square root transformation to the original Hmax data (this transformation is hereon referred to as the square root transformed data). Use the summary() function to summarize the results of this transformation.

```
waves$sqrtdata <- sqrt(waves$Hmax)

summary(waves$sqrtdata)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4359  1.2961  1.4663  1.4699  1.6310  2.1954
```

o

Create a histogram of the square root transformed data. On this histogram, report the mean, median, and 20% trimmed mean using the same formatting options as in part d above. Comment on the center shape and spread.

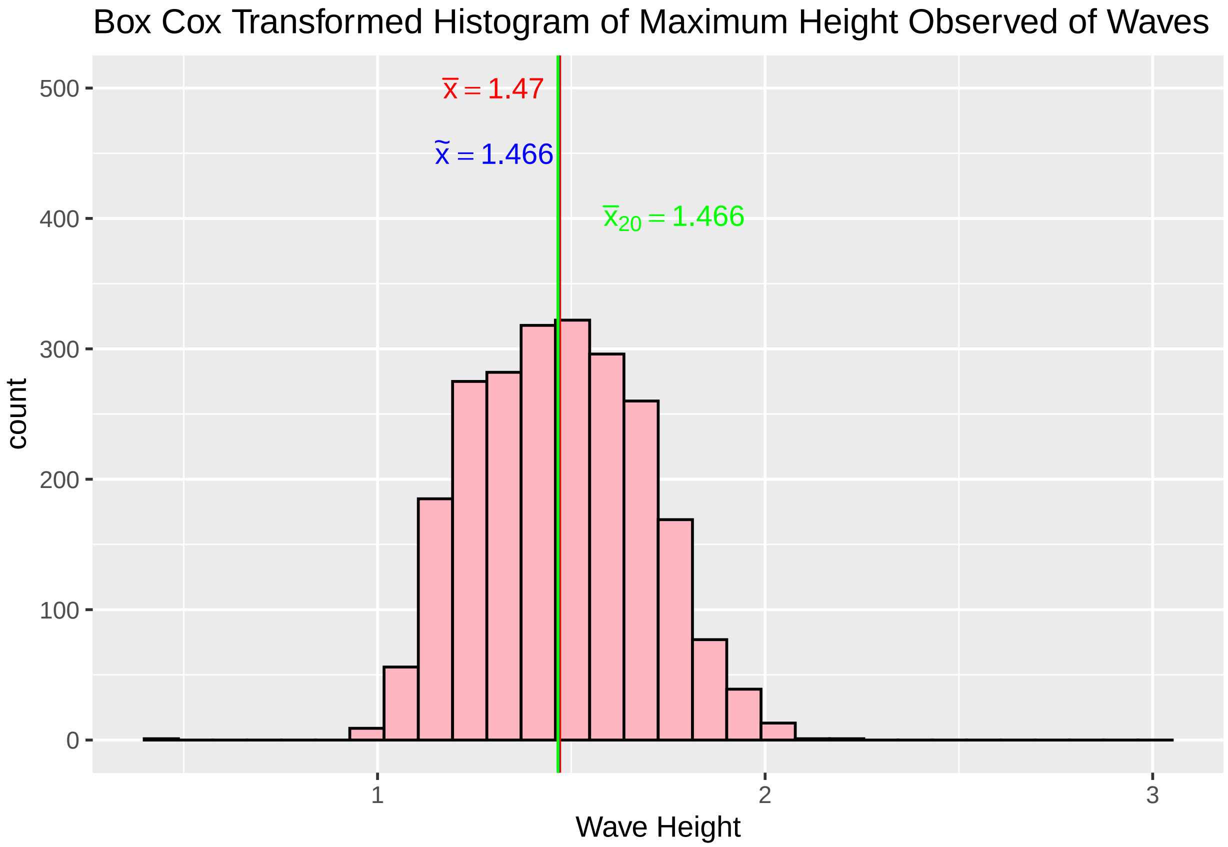
```
hist3 <- ggplot(waves, aes(sqrtdata)) + geom_histogram(fill = "light pink",
    color = "1") + labs(title = "Box Cox Transformed Histogram of Maximum Height Observed of Waves",
    x = "Wave Height")

# Mean visualization
hist3 <- hist3 + geom_vline(aes(xintercept = mean(waves$sqrtdata)), color = "red") +
    annotate("text", x = 1.3, y = 500, label = paste("bar(x)==", round(mean(waves$sqrtdata),
```

```r
        3)), parse = T, color = "red")

# Median visualization
hist3 <- hist3 + geom_vline(aes(xintercept = median(waves$sqrtdata)), color = "blue") +
    annotate("text", x = 1.3, y = 450, label = paste("tilde(x)==", round(median(waves$sqrtdata),
        3)), parse = T, color = "blue")

# 20% trimmed mean visualization
hist3 <- hist3 + geom_vline(aes(xintercept = mean(waves$sqrtdata, trim = 0.2),
    3), color = "green") + annotate("text", mean(waves$sqrtdata, trim = 0.2) +
    0.3, y = 400, label = paste("bar(x)[20]==", round(mean(waves$sqrtdata, trim = 0.2),
    3)), parse = T, color = "green")
```

```
## Warning: Ignoring unknown aesthetics: x
```

```r
hist3
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

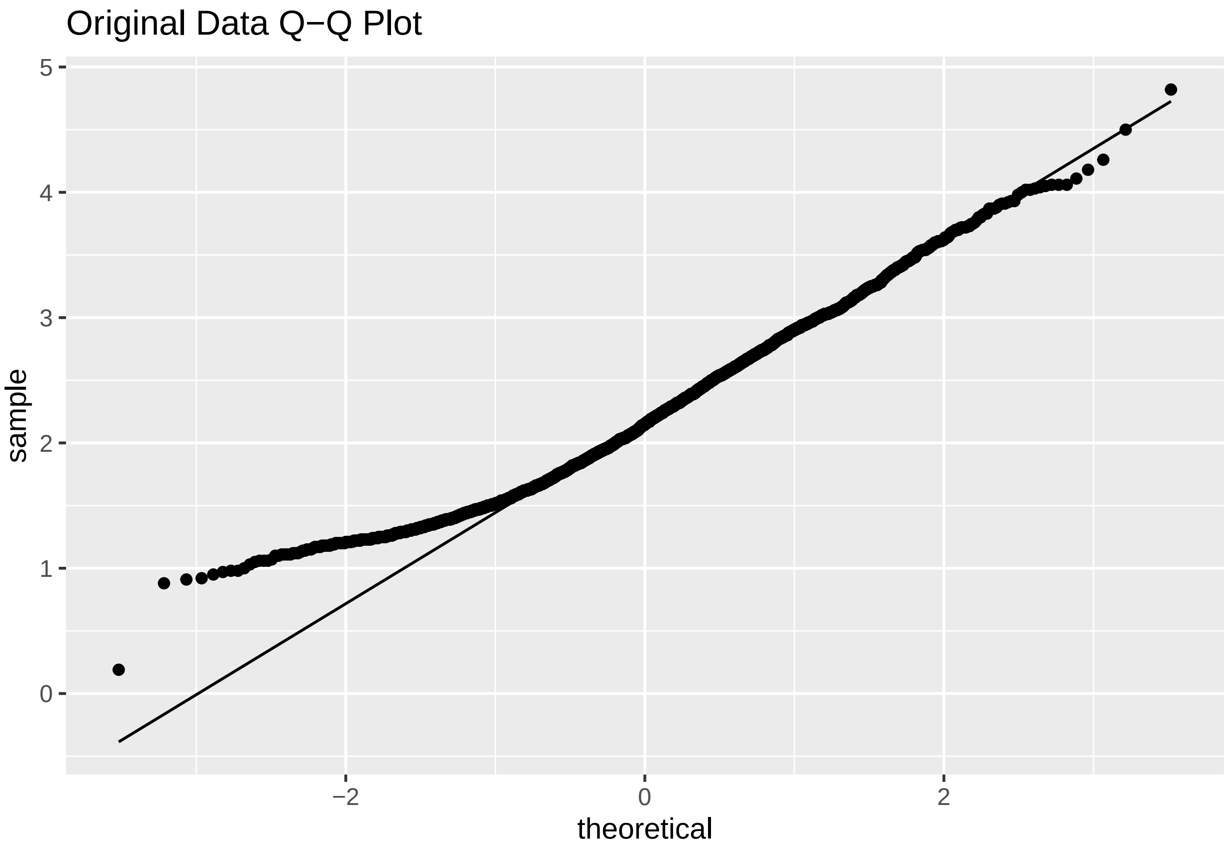


Here, the shape is one-center and the spread is also more bell-shpaed and is no longer left-skewed like the original data. The mean, median, and trimmed mean are also close in value now.

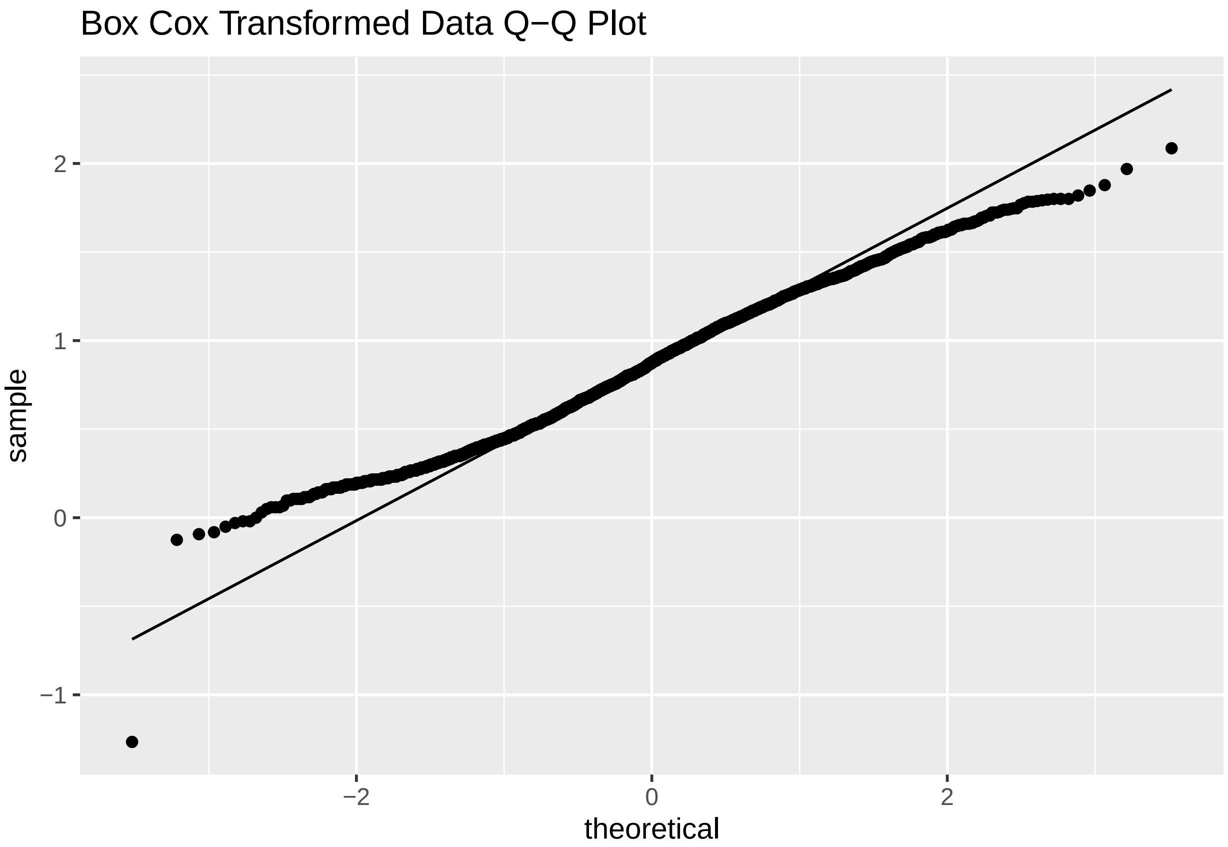
p

Create a qqplot for the original data, a qqplot for the Box-Cox transformed data, and a qqplot of the square root transformed data. Comment on the results.

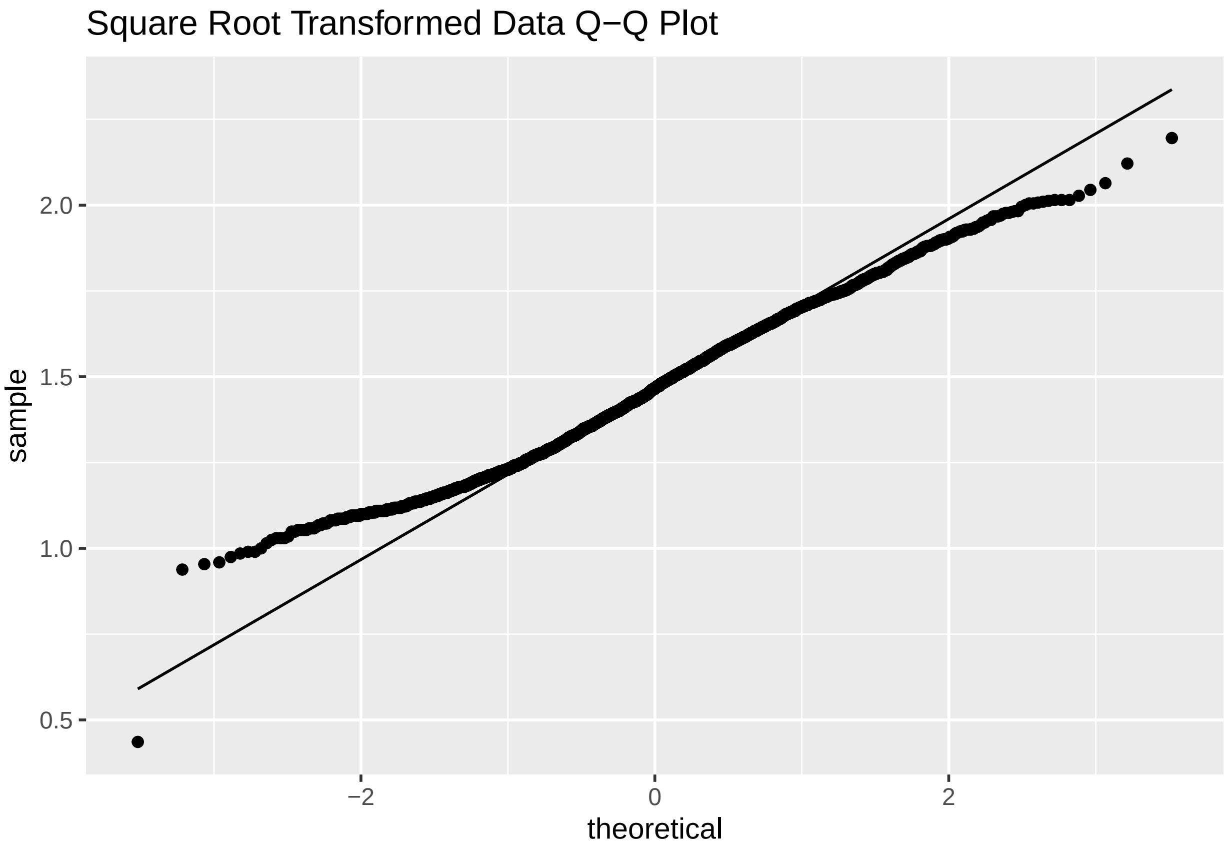
```r
qq1 <- ggplot(waves, aes(sample = Hmax)) + stat_qq() + stat_qq_line() + labs(title = "Original Data Q-Q
qq1
```



```
qq2 <- ggplot(waves, aes(sample = bcdata)) + stat_qq() + stat_qq_line() + labs(title = "Box Cox Transfo
qq2
```



```
qq3 <- ggplot(waves, aes(sample = sqrtdata)) + stat_qq() + stat_qq_line() +
    labs(title = "Square Root Transformed Data Q-Q Plot")
qq3
```



All three of these plots indicate that characteristics of the data are similar, and the transformations did not alter the representation of the data significantly. For instance, we see in all three that the data is slightly skewed left and that the tails are not as spread apart. It is slightly not normally distributed, though we saw this from our histogram earlier.

q

Evaluate the empirical rule for the original data, the Box-Cox transformed data, and the square root trasnformed data. In particular, make a table similar to that on slide 94 of Chapter 2 notes. Comment on the results. Do either of the transformed data seem to be "better" to work with? Note, you can use code similar to the following to answer this question:

```r
### Create a matrix named 'mat' with 9 rows & 5 columns
mat <- matrix(NA, nrow = 9, ncol = 5)

### Set row names and column names
rownames(mat) <- c("Original", "", "", "Box-Cox", "", "", "Square Root", "",
    "")
colnames(mat) <- c("x", "xbar-k*s", "xbar+k*s", "Theoretical %", "Actual %")

### Fill in known quantities
mat[, 1] <- c(1, 2, 3)
mat[, 4] <- c(68, 95, 99.7)

### Fill in calculated values (I only give a preview of this and leave the
### remaining calculations for you).  I use 'orig' as the original data,
### 'bcdat' as the Box-Cox transformed data, and 'srdat' as the square
### root-transformed data Name your variables anything you'd like.
```

```r
# Original Data
mat[1, 2] <- mean(waves$Hmax) - 1 * sd(waves$Hmax)
mat[2, 2] <- mean(waves$Hmax) - 2 * sd(waves$Hmax)
mat[3, 2] <- mean(waves$Hmax) - 3 * sd(waves$Hmax)

mat[1, 3] <- mean(waves$Hmax) + 1 * sd(waves$Hmax)
mat[2, 3] <- mean(waves$Hmax) + 2 * sd(waves$Hmax)
mat[3, 3] <- mean(waves$Hmax) + 3 * sd(waves$Hmax)

mat[1, 5] <- sum(waves$Hmax > mean(waves$Hmax) - 1 * sd(waves$Hmax) & waves$Hmax <
    mean(waves$Hmax) + 1 * sd(waves$Hmax))/length(waves$Hmax) * 100
mat[2, 5] <- sum(waves$Hmax > mean(waves$Hmax) - 2 * sd(waves$Hmax) & waves$Hmax <
    mean(waves$Hmax) + 2 * sd(waves$Hmax))/length(waves$Hmax) * 100
mat[3, 5] <- sum(waves$Hmax > mean(waves$Hmax) - 3 * sd(waves$Hmax) & waves$Hmax <
    mean(waves$Hmax) + 3 * sd(waves$Hmax))/length(waves$Hmax) * 100

# Box Cox Transformed Data
mat[4, 2] <- mean(waves$bcdata) - 1 * sd(waves$bcdata)
mat[5, 2] <- mean(waves$bcdata) - 2 * sd(waves$bcdata)
mat[6, 2] <- mean(waves$bcdata) - 3 * sd(waves$bcdata)

mat[4, 3] <- mean(waves$bcdata) + 1 * sd(waves$bcdata)
mat[5, 3] <- mean(waves$bcdata) + 2 * sd(waves$bcdata)
mat[6, 3] <- mean(waves$bcdata) + 3 * sd(waves$bcdata)

mat[4, 5] <- sum(waves$bcdata > mean(waves$bcdata) - 1 * sd(waves$bcdata) &
    waves$bcdata < mean(waves$bcdata) + 1 * sd(waves$bcdata))/length(waves$bcdata) *
    100
mat[5, 5] <- sum(waves$bcdata > mean(waves$bcdata) - 2 * sd(waves$bcdata) &
    waves$bcdata < mean(waves$bcdata) + 2 * sd(waves$bcdata))/length(waves$bcdata) *
    100
mat[6, 5] <- sum(waves$bcdata > mean(waves$bcdata) - 3 * sd(waves$bcdata) &
    waves$bcdata < mean(waves$bcdata) + 3 * sd(waves$bcdata))/length(waves$bcdata) *
    100

# Sqrt Transformed Data
mat[7, 2] <- mean(waves$sqrtdata) - 1 * sd(waves$sqrtdata)
mat[8, 2] <- mean(waves$sqrtdata) - 2 * sd(waves$sqrtdata)
mat[9, 2] <- mean(waves$sqrtdata) - 3 * sd(waves$sqrtdata)

mat[7, 3] <- mean(waves$sqrtdata) + 1 * sd(waves$sqrtdata)
mat[8, 3] <- mean(waves$sqrtdata) + 2 * sd(waves$sqrtdata)
mat[9, 3] <- mean(waves$sqrtdata) + 3 * sd(waves$sqrtdata)

mat[7, 5] <- sum(waves$sqrtdata > mean(waves$sqrtdata) - 1 * sd(waves$sqrtdata) &
    waves$sqrtdata < mean(waves$sqrtdata) + 1 * sd(waves$sqrtdata))/length(waves$sqrtdata) *
    100
mat[8, 5] <- sum(waves$sqrtdata > mean(waves$sqrtdata) - 2 * sd(waves$sqrtdata) &
    waves$sqrtdata < mean(waves$sqrtdata) + 2 * sd(waves$sqrtdata))/length(waves$sqrtdata) *
    100
mat[9, 5] <- sum(waves$sqrtdata > mean(waves$sqrtdata) - 3 * sd(waves$sqrtdata) &
    waves$sqrtdata < mean(waves$sqrtdata) + 3 * sd(waves$sqrtdata))/length(waves$sqrtdata) *
    100
```

```r
### Create a table
library(knitr)
kable(x = mat, digits = 2, row.names = T, format = "markdown")
```

	x	xbar-k*s	xbar+k*s	Theoretical %	Actual %
Original	1	1.56	2.86	68.0	64.50
	2	0.91	3.51	95.0	96.40
	3	0.26	4.15	99.7	99.78
Box-Cox	1	0.49	1.26	68.0	63.54
	2	0.10	1.65	95.0	97.31
	3	-0.29	2.03	99.7	99.91
Square Root	1	1.25	1.69	68.0	63.54
	2	1.03	1.91	95.0	97.14
	3	0.82	2.12	99.7	99.91

Looking at the Actial percentages and the difference from theoretical values, I do not see a signifigant difference. If I were to use either the original data, or the box cox transformed data, or the square root transformed data, I believe any would be good for reporting purposes on this dataset.

Question 2

a

Read the data into RStudio and print the first 6 entries of the dataset.

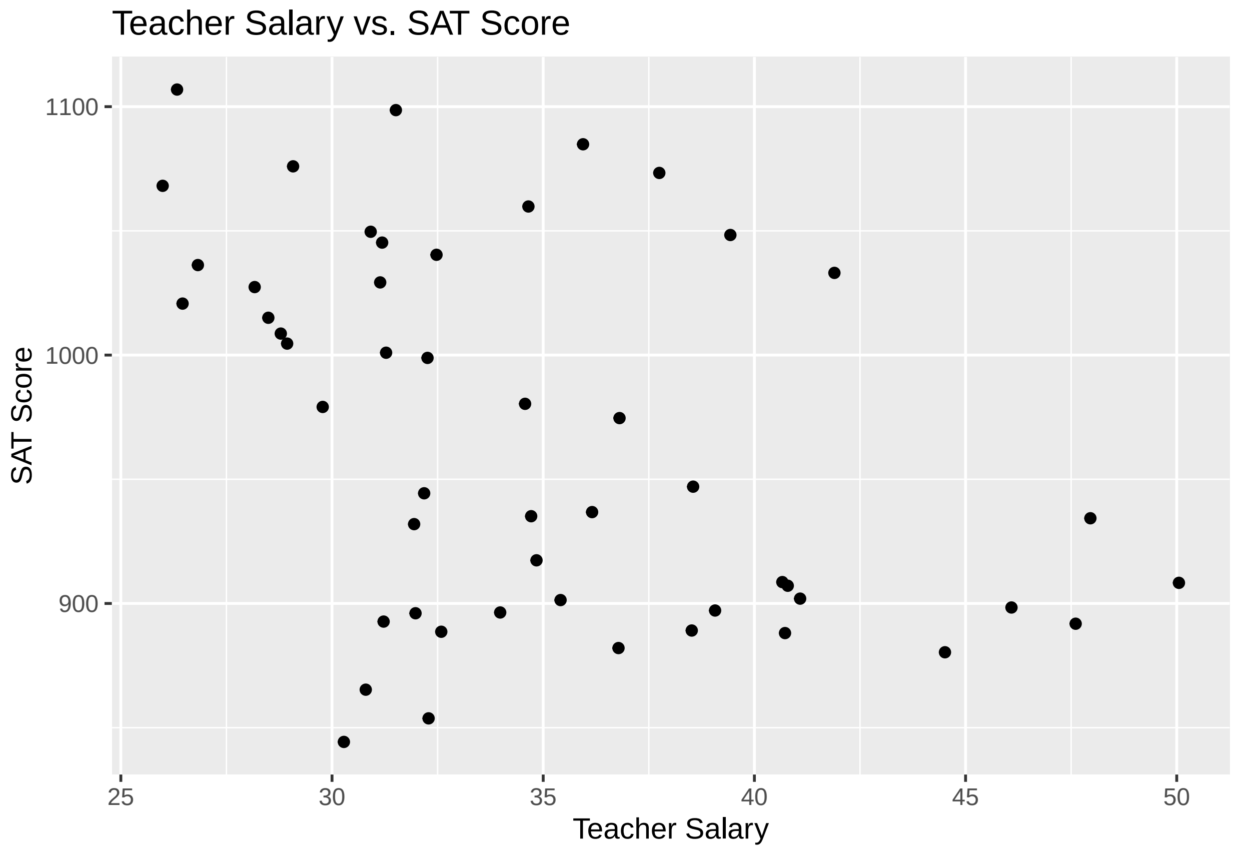
```r
SAT <- read.csv("SAT.csv")
head(SAT, n = 6)
```

```
##        state expend ratio salary frac verbal math  sat
## 1    Alabama  4.405  17.2 31.144    8    491  538 1029
## 2     Alaska  8.963  17.6 47.951   47    445  489  934
## 3    Arizona  4.778  19.3 32.175   27    448  496  944
## 4   Arkansas  4.459  17.1 28.934    6    482  523 1005
## 5 California  4.992  24.0 41.078   45    417  485  902
## 6   Colorado  5.443  18.4 34.571   29    462  518  980
```

b

Plot total SAT score (y axis) against teacher salary (x axis). Make sure to label your axes. Comment on the form, strength, and direction of the plot.

```r
scat1 <- ggplot(SAT, aes(x = salary, y = sat)) + geom_jitter(aes(x = salary,
    y = sat)) + labs(title = "Teacher Salary vs. SAT Score", x = "Teacher Salary",
    y = "SAT Score")

scat1
```



The form of the plot appears to be very scattered with no immediately distinguishable trend other than higher salary does not appear to produce more higher SAT scores than lower salaried teachers. It could potentially even be negatively correlated. Meaning student results are positively impacted by lower salaried teachers

c

Calculate the correlation between SAT score and teacher salary. Why is this an interesting result?

```
cat("Correlation Coef.: ", cor(SAT$salary, SAT$sat))
```

```
## Correlation Coef.:  -0.4398834
```

This is interesting because it is just as I suggested. Student results are positively impacted by lower salaried teachers, which would not be an immediate guess by many.

d

From help(SAT), we can see that frac is the percentage of all elligible students taking the SAT. Create a categorical variable based on frac. In particular, categorize frac into “low”, “medium”, and “high” groups, based on whether the percentage was in the range “(0,22)”, “(22,49)”,“(49,81)”, respectively. Do this using the cut function as follows:

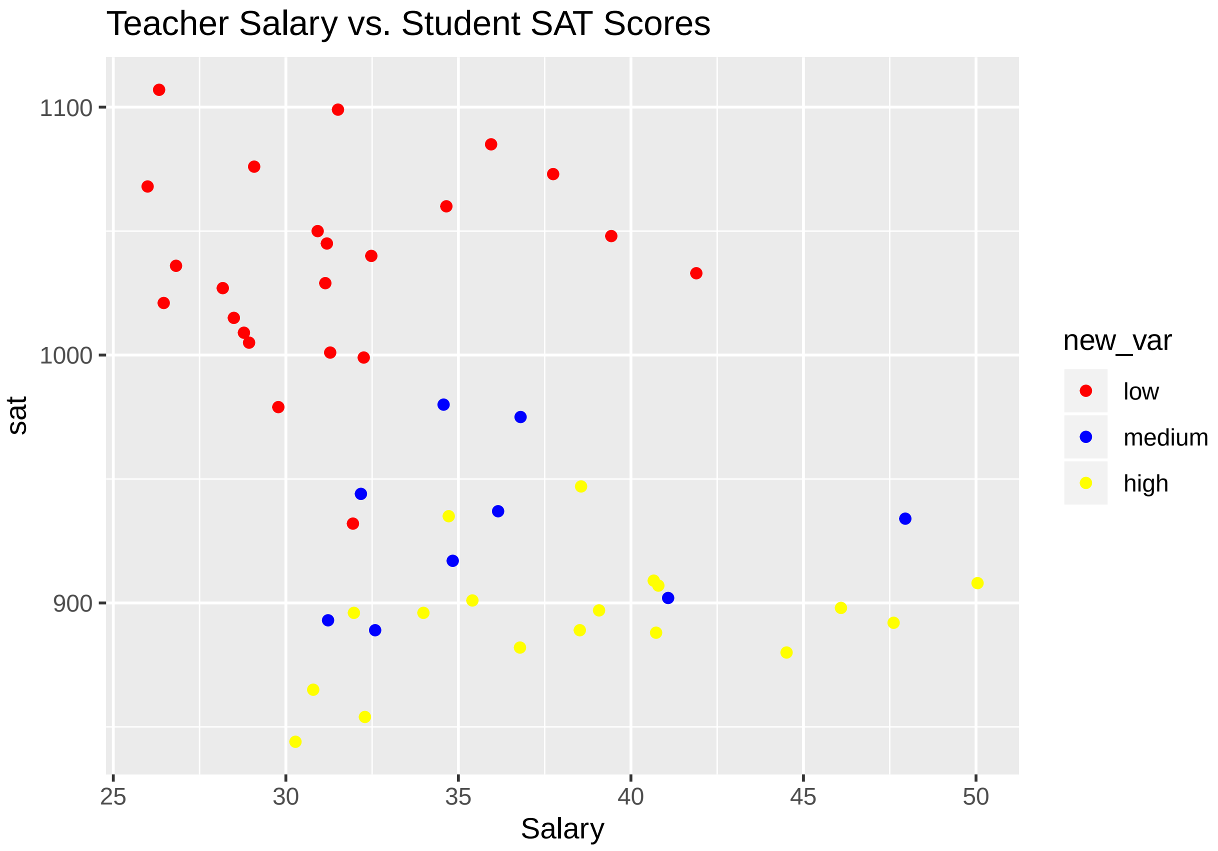
```
SAT$new_var <- cut(SAT$frac, breaks = c(0, 22, 49, 81), labels = c("low", "medium",
    "high"))
SAT$new_var
```

```
##  [1] low    medium medium low    medium medium high   high   medium high
## [11] high   low    low    high   low    low    low    low    high   high
## [21] high   low    low    low    low    low    low    medium high   high
## [31] low    high   high   low    medium low    high   high   high   high  
## [41] low    low    medium low    high   high   medium low    low    low   
## Levels: low medium high
```

e

Plot total SAT score (y axis) against teacher salary (x axis), but now color points based on which frac group then fall into. You can do this by specifying the col=new_var option in the plot() function. Comment on the results.

```
scat2 <- ggplot(data = SAT, aes(salary, sat, color = new_var)) + geom_point() +
    scale_color_manual(values = c(low = "red", medium = "blue", high = "yellow")) +
    labs(title = "Teacher Salary vs. Student SAT Scores", x = "Salary", "SAT Scores")

scat2
```



f

Calculate the correlation between SAT score and teacher salary for each of the three groups specified by new_var. Comment on the results. You can use the following code to get started:

```
# Correlation betwen low salary & SAT Scores
cat("Low Salary Correlation:    ", cor(SAT$salary[SAT$new_var == "low"], SAT$sat[SAT$new_var ==
    "low"]), "\n")
```

```
## Low Salary Correlation:     0.08032564
```

```
# Correlation betwen medium salary & SAT Scores
cat("Medium Salary Correlation: ", cor(SAT$salary[SAT$new_var == "medium"],
    SAT$sat[SAT$new_var == "medium"]), "\n")
```

```
## Medium Salary Correlation:  0.1071504
```

```
# Correlation betwen high salary & SAT Scores
cat("High Salary Correlation:   ", cor(SAT$salary[SAT$new_var == "high"], SAT$sat[SAT$new_var ==
    "high"]), "\n")
```

```
## High Salary Correlation:    0.3432556
```

You should see that the correlations for each group have a different sign from when all the data are considered together. This is an example of Simpson's paradox, since by considering a third variable (in this case frac) the direction of association between two variables of interest has changed.

g

Create two new categorical variables. The first will be an indicator for whether expend is greater than 7. The second will be an indicator for whether sat is greater than 900. Use the following example code:

```
# Sorting values of SAT scores and states expenditures
SAT$expendCAT <- ifelse(SAT$expend > 7, "high", "low")
SAT$highSAT <- ifelse(SAT$sat > 900, "greater", "lower")

# head(SAT, n=10)

# Grouping greater/lower SAT scores & high/low expenditures
prop.table(table(SAT$expendCAT, SAT$highSAT), 1)
```

```
##
##          greater     lower
##   high 0.4444444 0.5555556
##   low  0.7804878 0.2195122
```

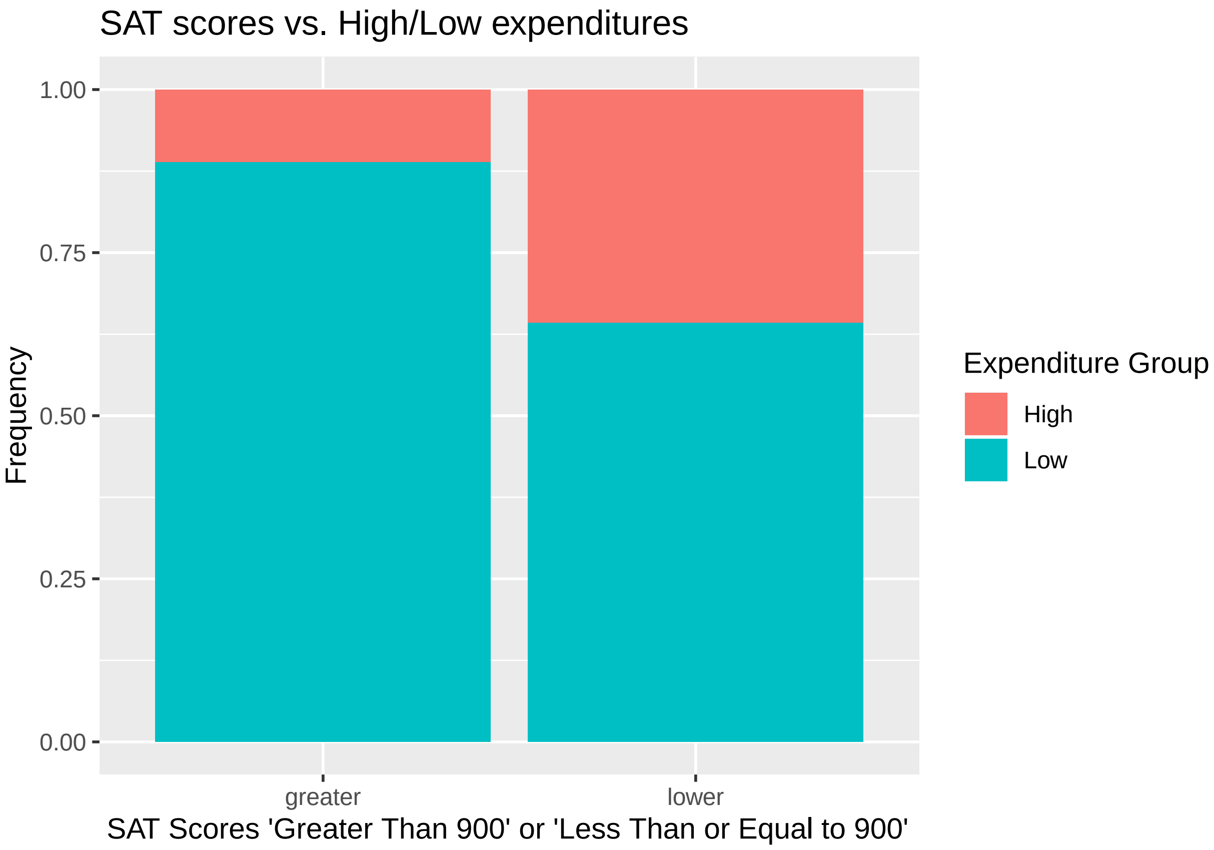
```
tab1 <- table(SAT$expendCAT, SAT$highSAT)
new <- melt(tab1)

new
```

```
##   Var.1   Var.2 value
## 1  high greater     4
## 2   low greater    32
## 3  high   lower     5
## 4   low   lower     9
```

h

Create a stacked barplot with a bar for each SAT score group. Each bar should be broken up into two pieces: one for high expenditures and one for low expenditures Make sure to label your axes and add a legend. Comment on the results. Why does this result seem counterintuitive?

```
# Plotting by proportion
bar1 <- ggplot(data = new, aes(x = Var.2, y = value, fill = Var.1)) + geom_bar(position = "fill",
    stat = "identity") + labs(title = "SAT scores vs. High/Low expenditures",
    x = "SAT Scores 'Greater Than 900' or 'Less Than or Equal to 900'", y = "Frequency") +
    scale_fill_discrete(name = "Expenditure Group", labels = c("High", "Low"))

bar1
```



The results seems to show that students who scored less than or equal to 900 equally came from states that expended more money or less. However, for students who did score greater than 900 more students came from states that spent less. This is on par with our scatterplot we visuallized previously. We saw that higher scoring SAT students tended to come from lower salaried teachers. This may indicate that higher SAT scores are not dependant on teacher's salaries.