Econ 104 Project 3

Pedram Bazargani

2022-12-03

Data description: Data Set contains Cost Data for U.S. Airlines, 90 Observations On 6 Firms For 15 Years, 1970-1984. Using the data, which contains information about the variables: Airline, Year, Q (output in revenue passenger miles), PF (Fuel price), LF (Load factor, the average capacity utilization of the fleet), we try to answer the following question:

Source: https://www.kaggle.com/datasets/sandhyakrishnan02/paneldata?resource=download

```
# loading data for question 1:
airline_data <- read.csv("PanelData.csv")
head(airline_data)
```

```
##   ï..I T       C        Q     PF       LF
## 1    1 1 1140640 0.952757 106650 0.534487
## 2    1 2 1215690 0.986757 110307 0.532328
## 3    1 3 1309570 1.091980 110574 0.547736
## 4    1 4 1511530 1.175780 121974 0.540846
## 5    1 5 1676730 1.160170 196606 0.591167
## 6    1 6 1823740 1.173760 265609 0.575417
```

```
names(airline_data)
```

```
## [1] "ï..I" "T"    "C"    "Q"    "PF"   "LF"
```

Question 1, Part 1:

```
summary(airline_data)
```

```
##       ï..I           T            C                Q          
##  Min.   :1.0   Min.   : 1   Min.   :  68978   Min.   :0.03768  
##  1st Qu.:2.0   1st Qu.: 4   1st Qu.: 292046   1st Qu.:0.14213  
##  Median :3.5   Median : 8   Median : 637001   Median :0.30503  
##  Mean   :3.5   Mean   : 8   Mean   :1122524   Mean   :0.54499  
##  3rd Qu.:5.0   3rd Qu.:12   3rd Qu.:1345968   3rd Qu.:0.94528  
##  Max.   :6.0   Max.   :15   Max.   :4748320   Max.   :1.93646  
##        PF                LF        
##  Min.   : 103795   Min.   :0.4321  
##  1st Qu.: 129848   1st Qu.:0.5288  
##  Median : 357434   Median :0.5661  
##  Mean   : 471683   Mean   :0.5605  
##  3rd Qu.: 849840   3rd Qu.:0.5947  
##  Max.   :1015610   Max.   :0.6763
```

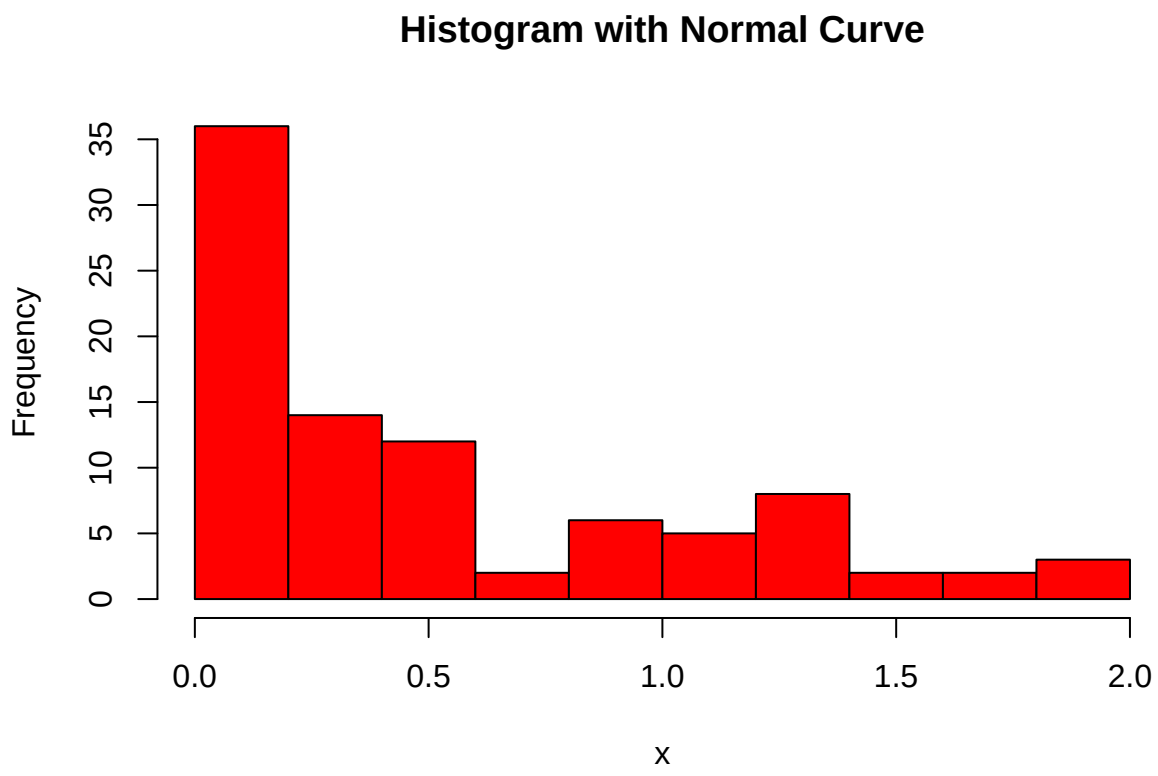
```
ls(airline_data)
```

```
## [1] "C"    "ï..I" "LF"   "PF"   "Q"    "T"
```

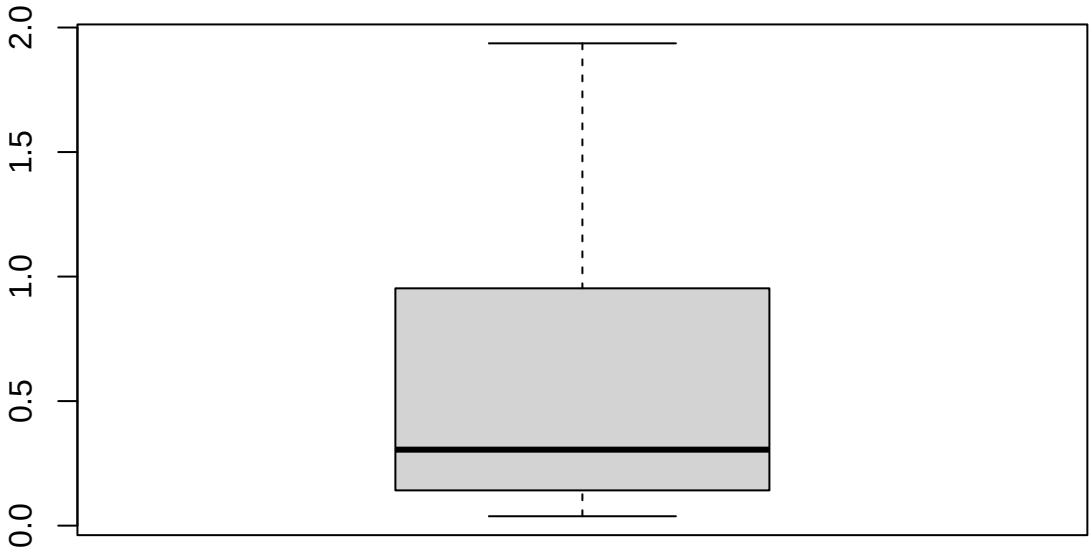
Looking at the summary output, there exists tight ranges for most of the variables with roughly normalized distributions or skewed right distributions. This can be attributed to the different airlines and how their cost structures evolved over time with some most likely having lower cost bases irrelevant to other variables. They could also be more sensitive to change. It seems that although most airlines have similar prices in market-determined variables like Fuel Price, Cost can still vary drastically.

Q: Output in revenue passenger miles

```
x <- airline_data$Q

#Histogram
h <- hist(x, breaks=10, col="red", main="Histogram with Normal Curve")
```

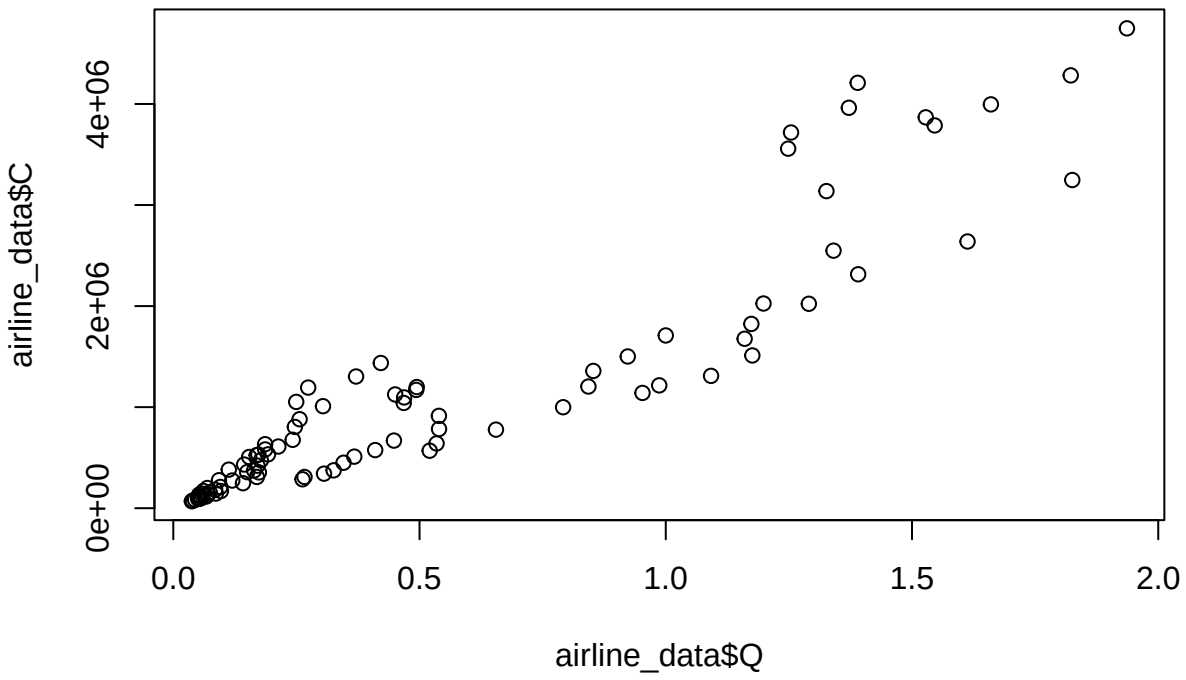



```
#Boxplot
boxplot(unlist(airline_data$Q), xlab="Revenue Passenger Miles")
```

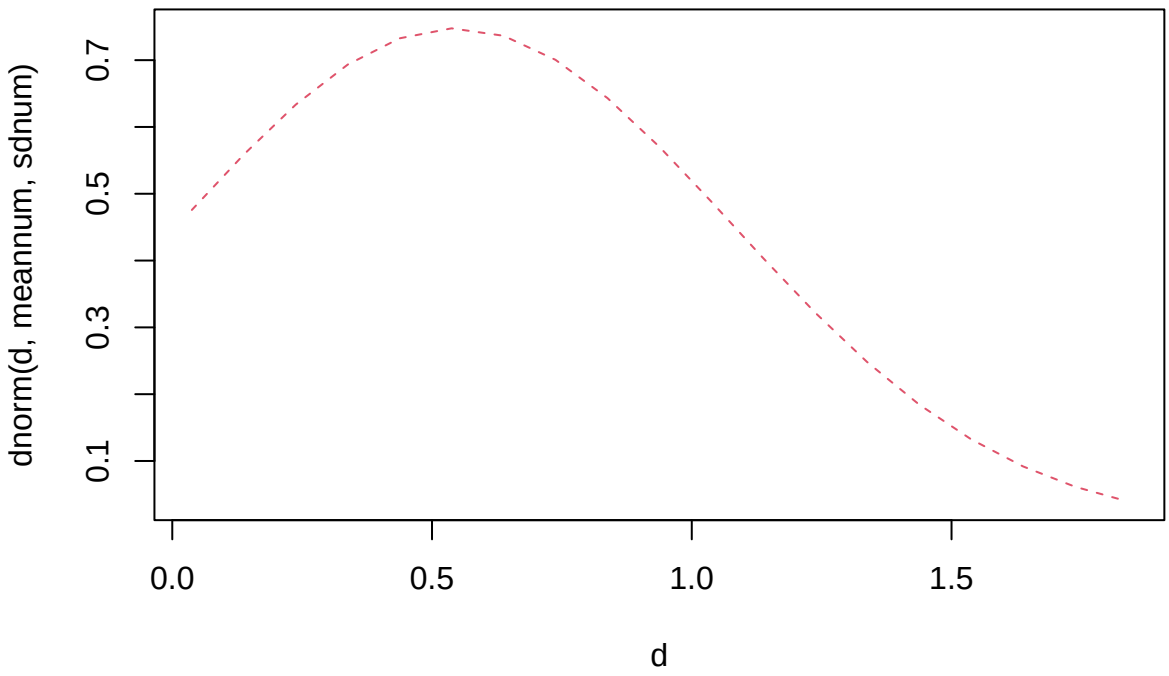


Revenue Passenger Miles

```
plot(airline_data$C~airline_data$Q)
```



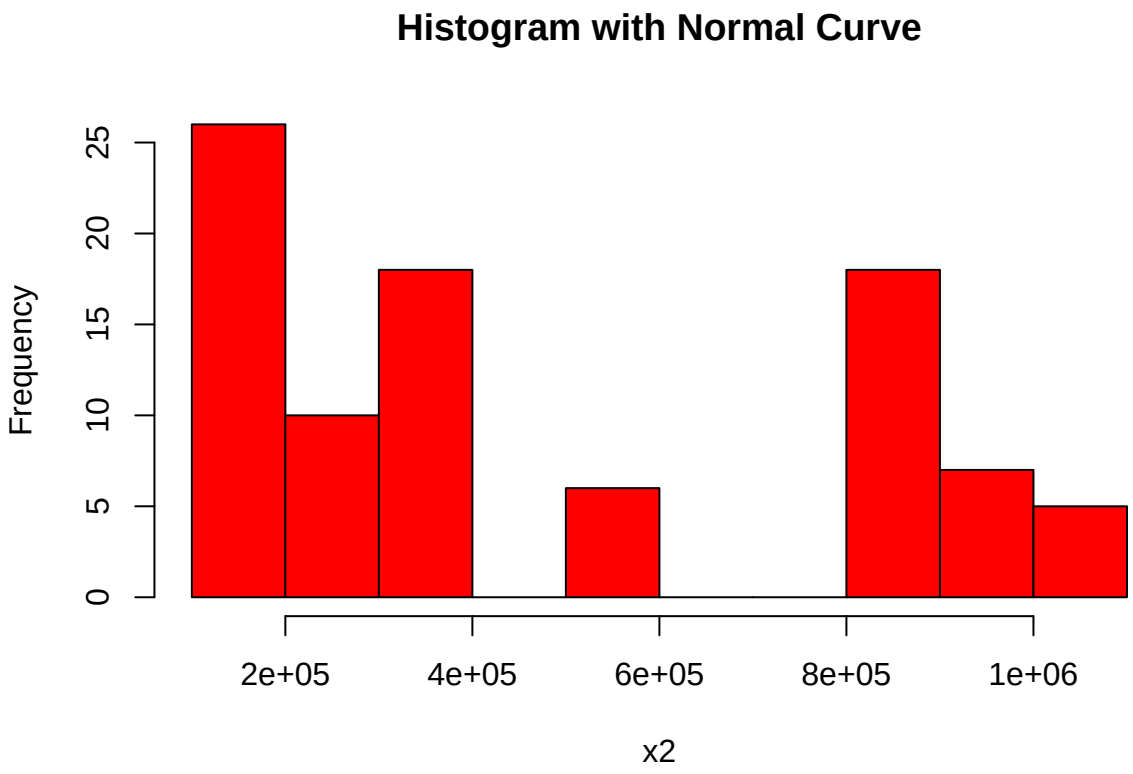

```
#scatterplot
meannum = mean(airline_data$Q,rm.na=TRUE)
sdnum = sd(airline_data$Q)
d <- seq(from=min(airline_data$Q),to=max(airline_data$Q),by=0.1)
plot(x =d,y=dnorm(d,meannum,sdnum),lty=2,col=2,type="l")
```



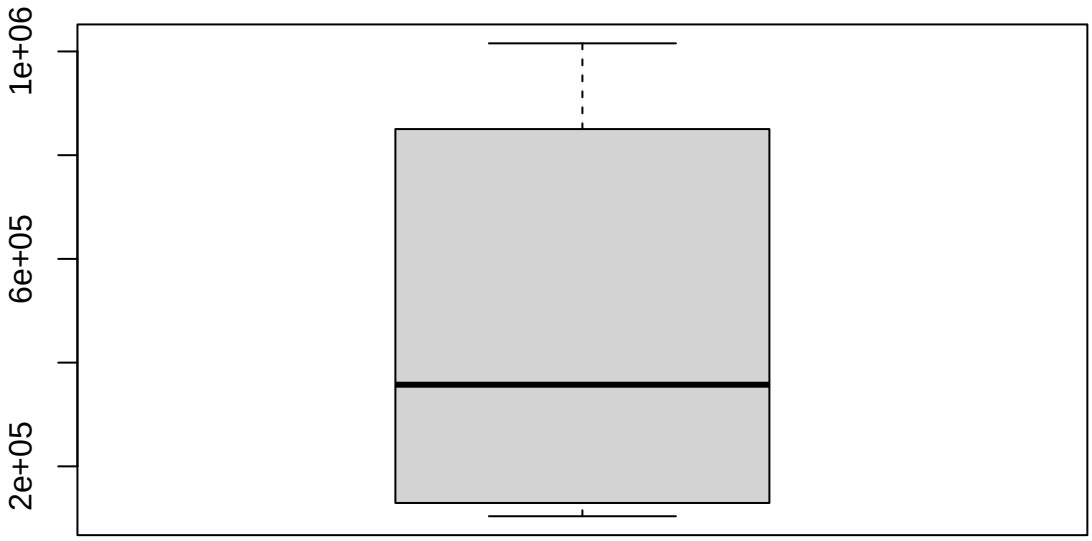
Comments: For our x variable, Q, which denotes Output in Revenue Passenger Miles, Histogram with a normal curve, we found that our variable's frequency peaks at approximately 35 and is centered between 0 and .25, with a majority of the variable being centered between 0 and .5. In our fitted distribution, we found that a majority of our variable is centered around 0 to 1.5, then plateaus downward around 1.5. Our dnorm (d, meannum, sdnum) peaks at around 0.07. Looking more specifically at our histogram, it appears to be skewed right with most of the data centered at the bottom end. Looking at our boxplot, the Q data has a median of .305 and a mean of .545 with no outliers shown. In our scatterplot, there exists a strong positive correlation between Cost and Revenue Passenger Miles. We run it against cost as that is our response variable.

Fuel Price

```
x2 <- airline_data$PF
h <- hist(x2, breaks=10, col="red", main="Histogram with Normal Curve")
```

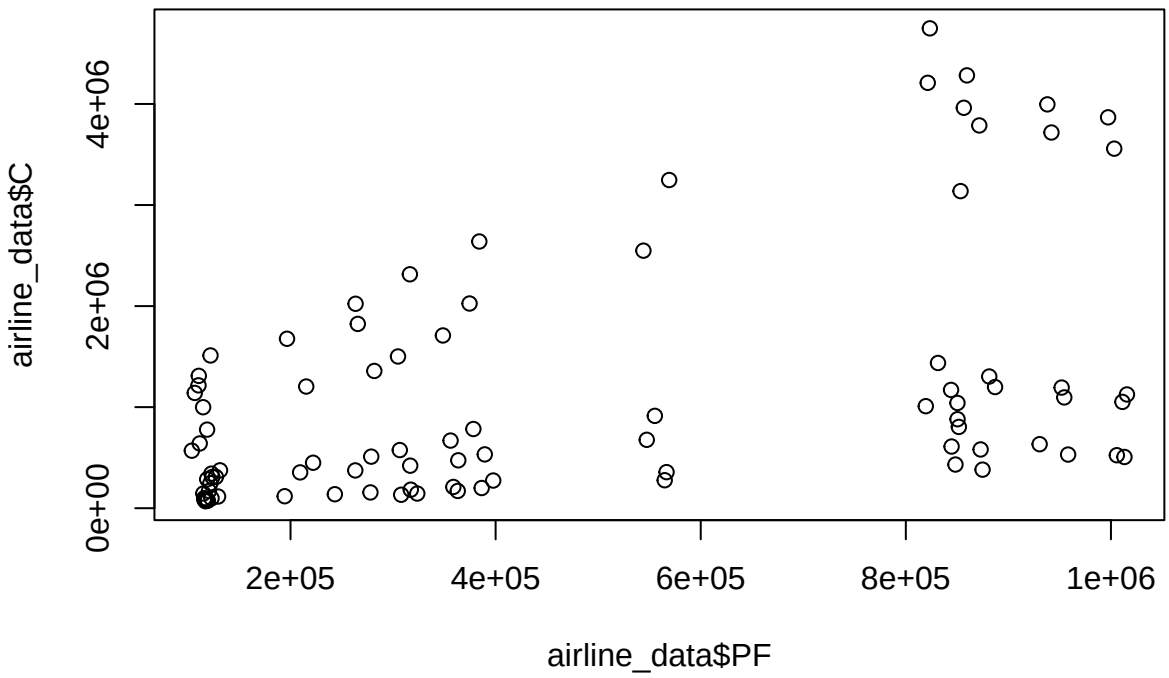



```
boxplot(unlist(airline_data$PF), xlab="Fuel Price")
```

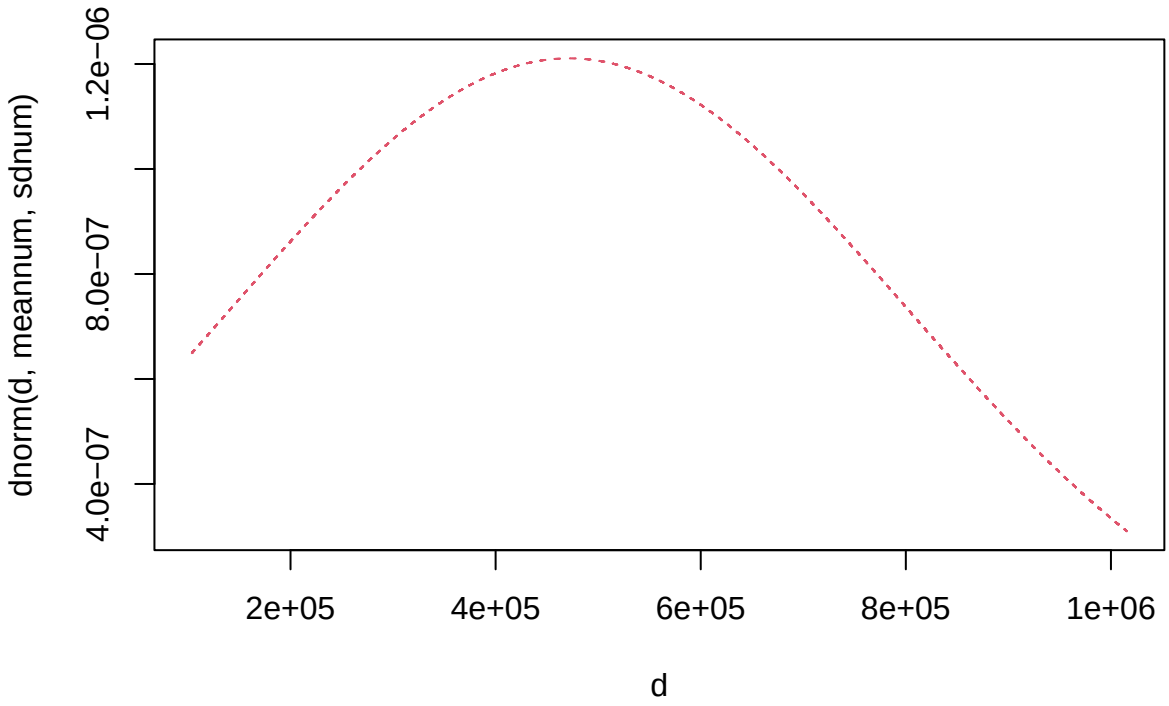


Fuel Price

```
plot(airline_data$C~airline_data$PF)
```



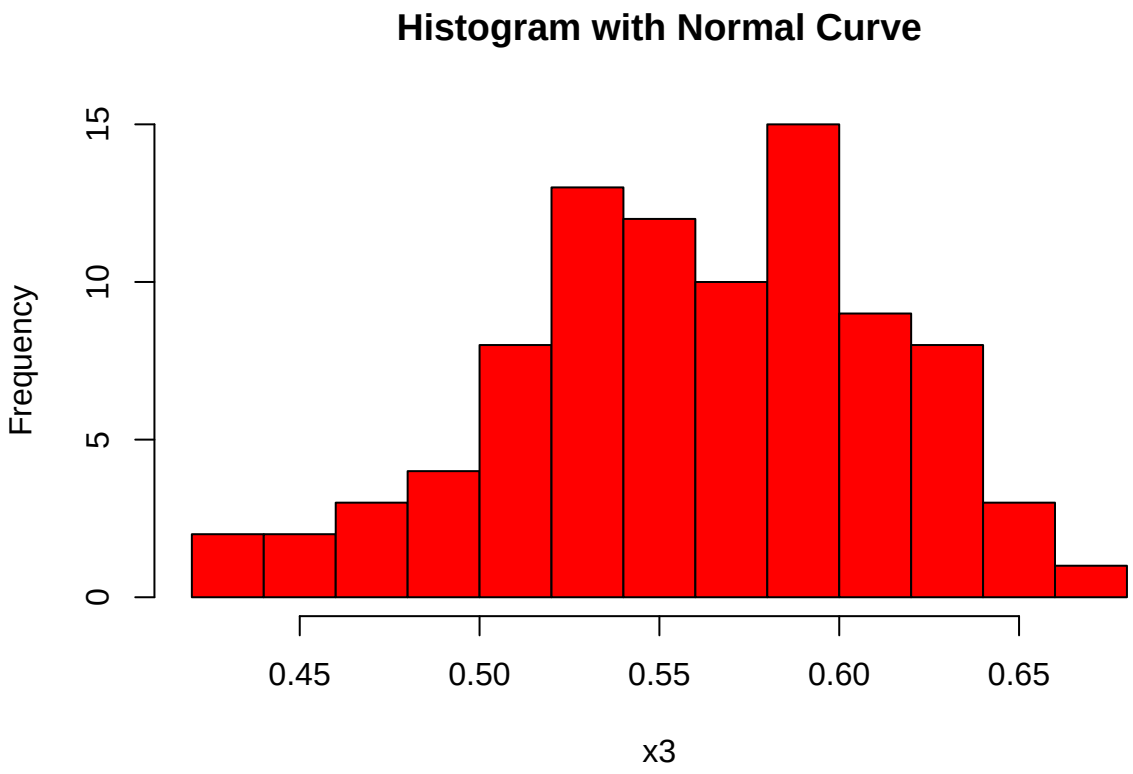

```
meannum = mean(airline_data$PF,rm.na=TRUE)
sdnum = sd(airline_data$PF)
d <- seq(from=min(airline_data$PF),to=max(airline_data$PF),by=0.1)
plot(x =d,y=dnorm(d,meannum,sdnum),lty=2,col=2,type="l")
```



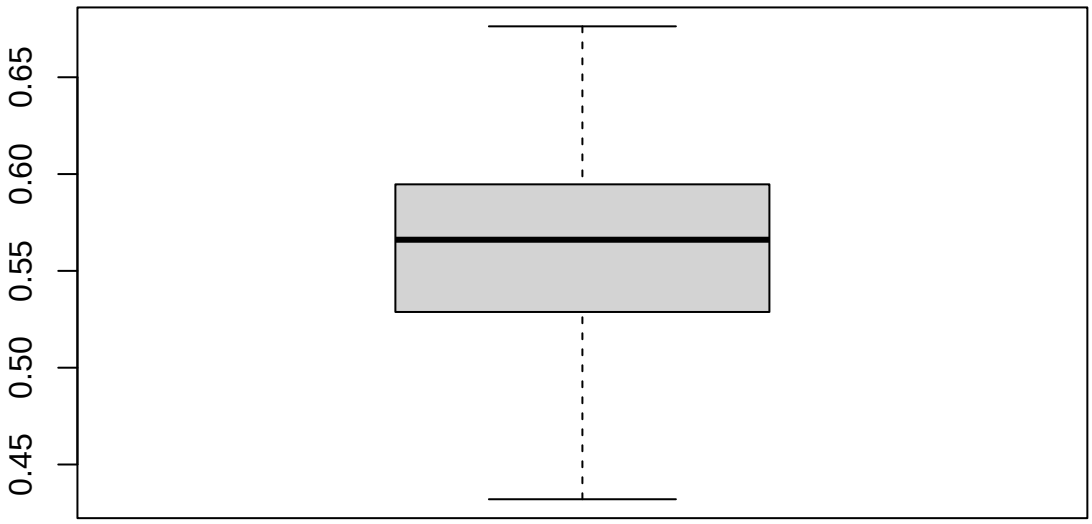
Comments: For our x variable, PF, which denotes Fuel Price, Histogram with a normal curve, we found that our variable's frequency peaks at approximately 30 and is centered between 0 and 2x10^5, with a majority of the variable being centered between 0 and 4x10^5. In our fitted distribution, we found that a majority of our variable is centered around 0 to 4x10^5, then plateaus downward around 1x10^6. Our dnorm (d, meannum, sdnum) peaks at around 1.2x10^6. Looking more specifically at our histogram, the data is divided in two showing us a distinct bimodal distribution. This is probably due to fixed differences in Fuel Price for each airline. Looking at our boxplot, the PF data has a median of 357,434 and a mean of 471,683 with no outliers shown. Due to the distinct bimodal distribution, it makes sense that our boxplot looks to be normally distributed. In our scatterplot, there exists a range of correlations as each airline probably has differing factors when it comes to cost outside of fuel price. The general trend however is that as Fuel Price rises so does Cost. We run it against cost as that is our response variable.

LF: Load factor, or average capacity utilization of the fleet

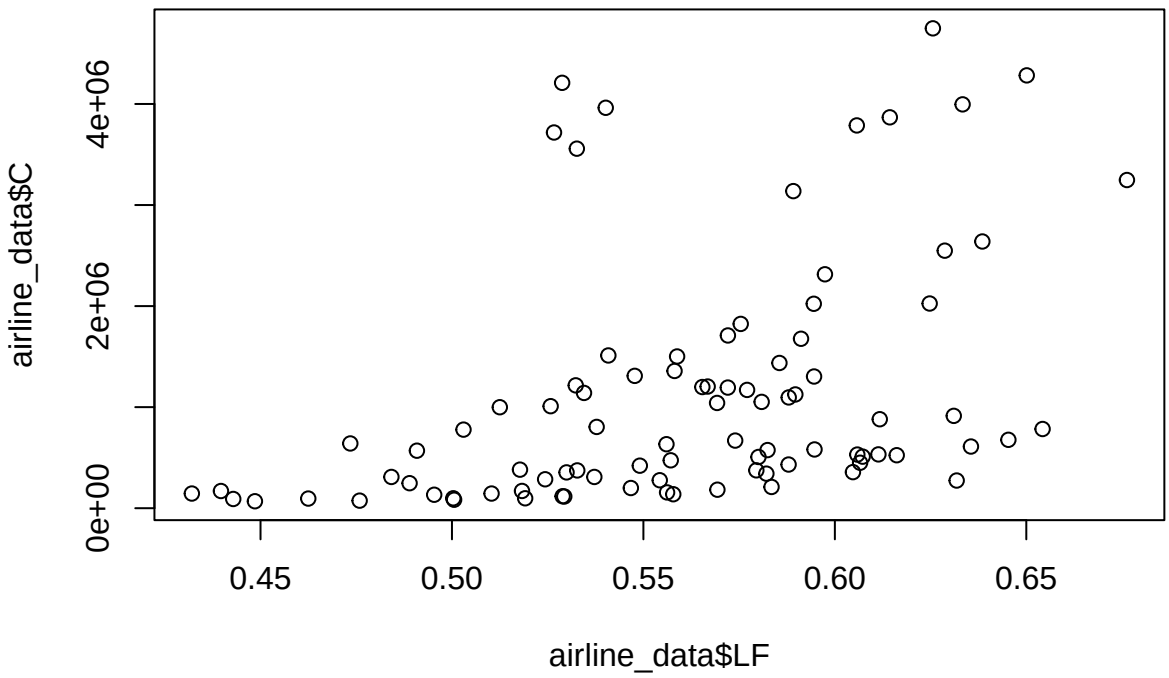
```
x3 <- airline_data$LF
h <- hist(x3, breaks=10, col="red", main="Histogram with Normal Curve")
```



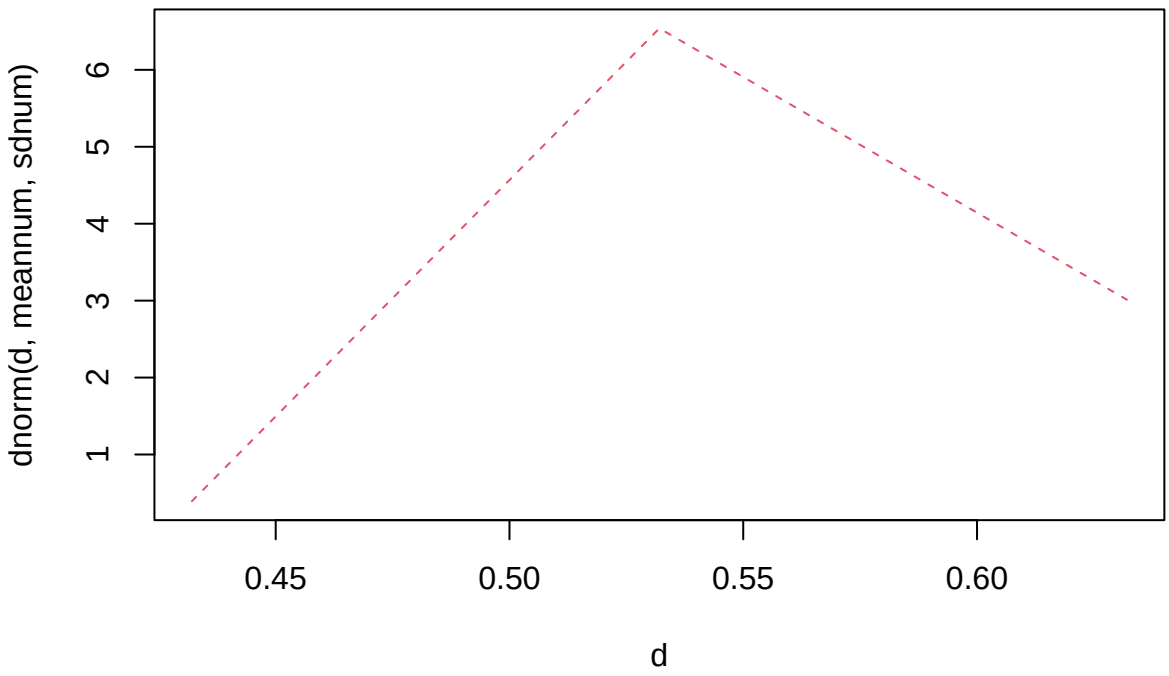

```
boxplot(unlist(airline_data$LF), xlab="Load Factor")
```




```
plot(airline_data$C~airline_data$LF)
```



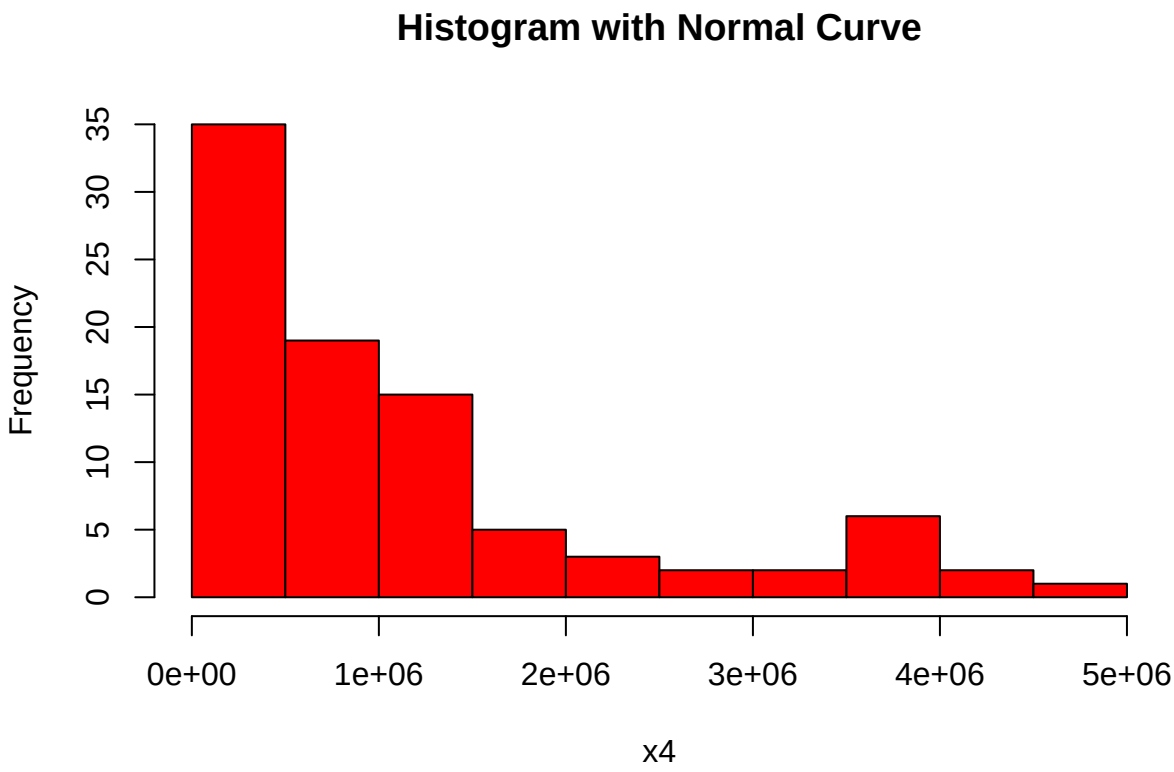

```
meannum = mean(airline_data$LF,rm.na=TRUE)
sdnum = sd(airline_data$LF)
d <- seq(from=min(airline_data$LF),to=max(airline_data$LF),by=0.1)
plot(x =d,y=dnorm(d,meannum,sdnum),lty=2,col=2,type="l")
```



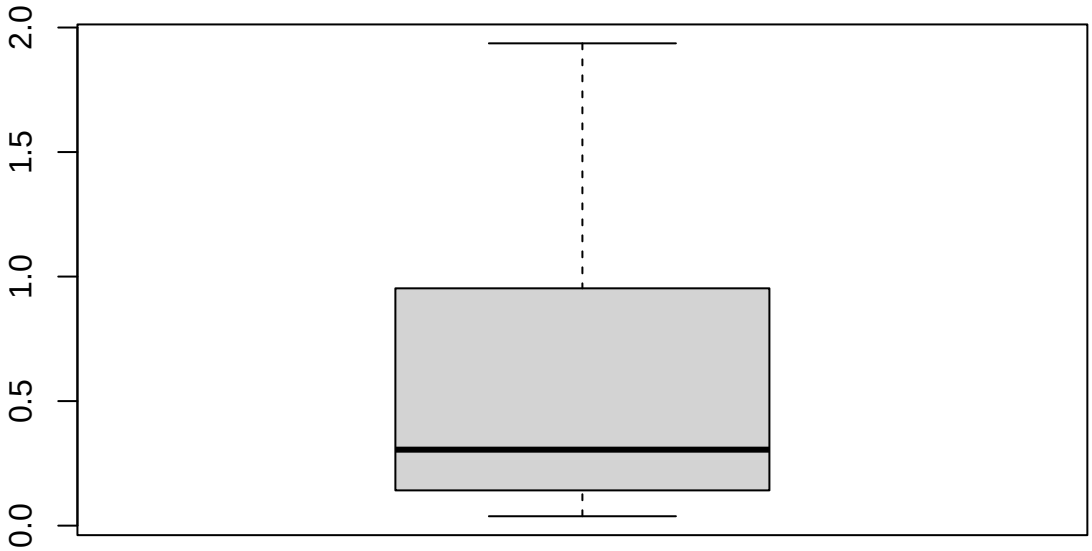
Comments: For our x variable, LF, histogram which denotes Load factor or the average capacity utilization of the fleet, we found that our variable's frequency peaks at approximately 15 and is centered between .5 and .6, with a majority of the variable being centered between .45 and .65. In our fitted distribution, we found that a majority of our variable is centered around .3 to .6, then plateaus downward around .65. Our dnorm (d, meannum, sdnum) peaks at around 6. Looking more specifically at our histogram, it appears to be distributed normally with an even distribution of data above and below the highest concentration. Looking at our boxplot, the LF data has a median of .567 and a mean of .561 with no outliers shown and a normal distribution. In our scatterplot, there exists a weak positive correlation between the Cost and Load factor. As the Load Factor increases so does Cost to a small degree. We run it against cost as that is our response variable.

C: Total costs in $1000s:

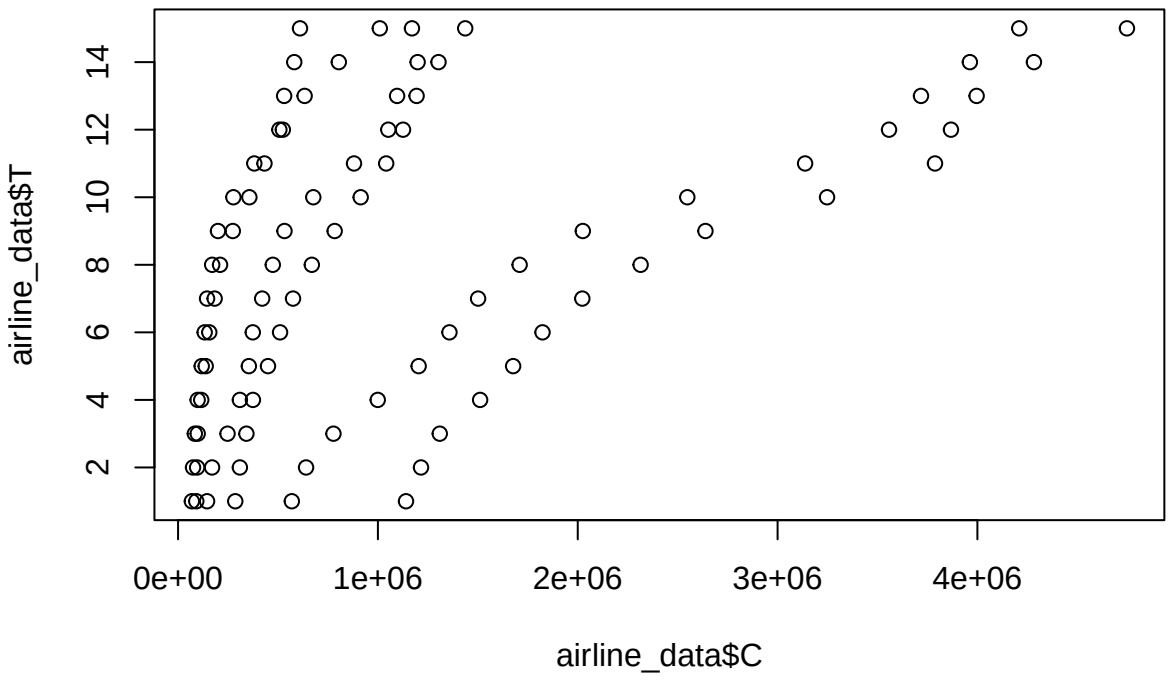
```
x4 <- airline_data$C
h <- hist(x4, breaks=10, col="red", main="Histogram with Normal Curve")
```



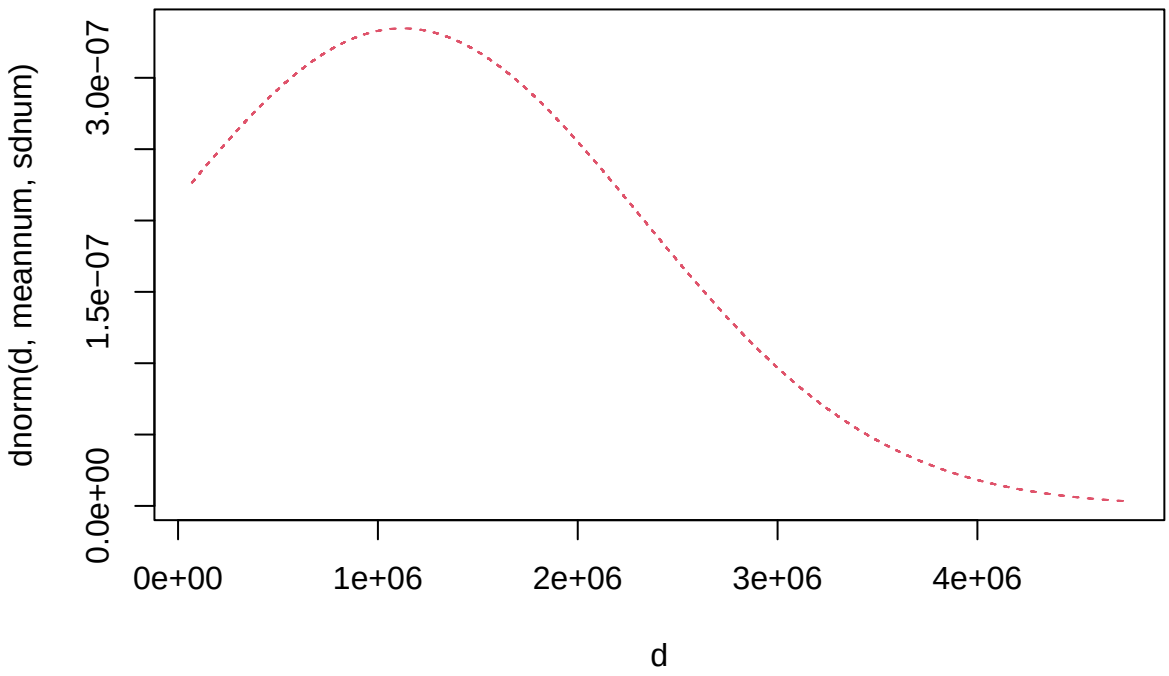

```
boxplot(unlist(airline_data$Q), xlab="Total Cost")
```




```
plot(airline_data$T~airline_data$C)
```




```
meannum = mean(airline_data$C,rm.na=TRUE)
sdnum = sd(airline_data$C)
d <- seq(from=min(airline_data$C),to=max(airline_data$C),by=0.1)
plot(x =d,y=dnorm(d,meannum,sdnum),lty=2,col=2,type="l")
```



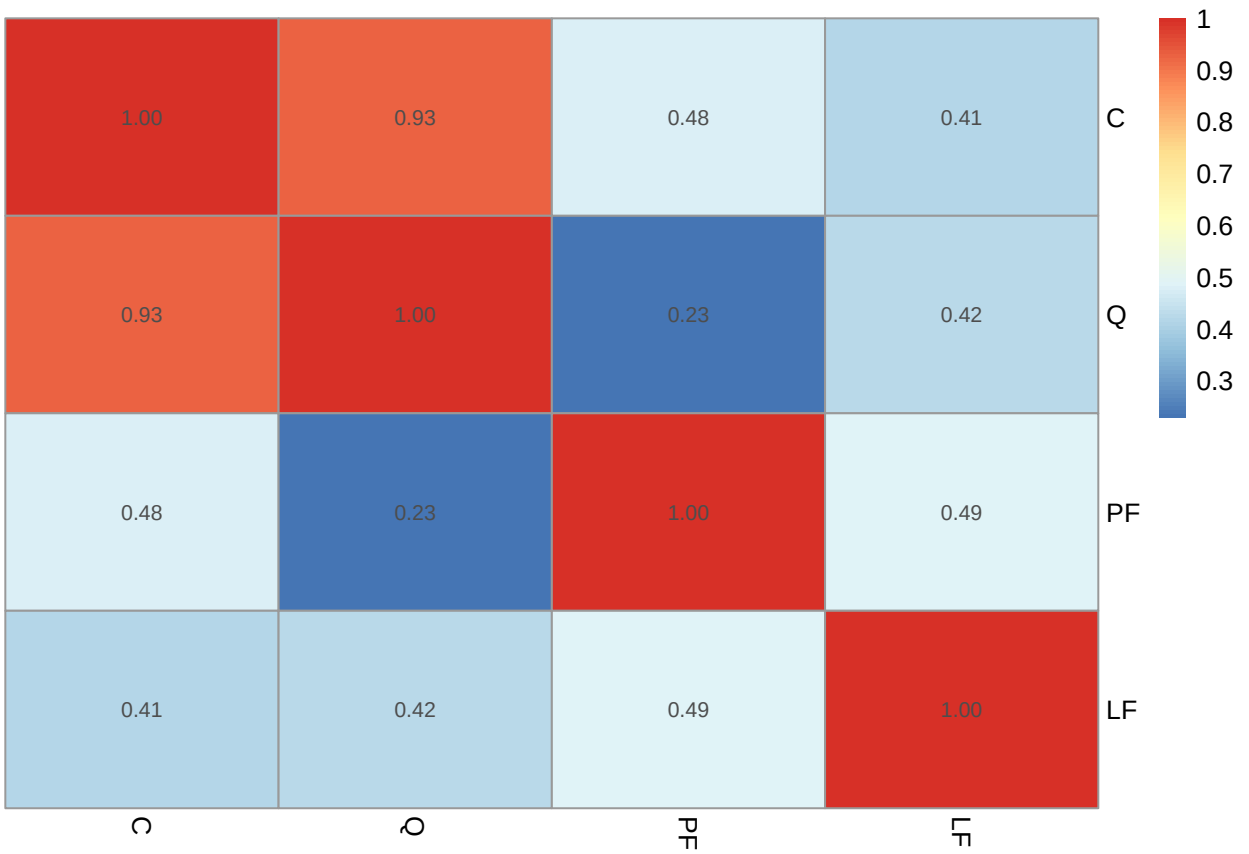
Comments: For our x variable, C, which denotes Total Cost in $1000s, Histogram with a normal curve, we found that our variable's frequency peaks at approximately 35 and is centered between 0 and 2x10^6, with a majority of the variable being centered between 0 and 4x10^6. In our fitted distribution, we found that a majority of our variable is centered around 0 to 1x10^6, then plateaus downward around 4x10^6. Our dnorm (d, meannum, sdnum) peaks at around 3x10^-7. Looking more specifically at our histogram, it appears to be skewed right with most of the data center at the bottom end. Looking at our boxplot, the C data has a median of 637001 and a mean of 1122524 with no outliers shown. In our scatterplot, as C is the response variable we can see the varying costs associated with each different airline over time. Some airlines experience drastically differing Costs as time went on and that progression is shown in the scatterplot.

Correlation Heatmaps:

```
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 4.1.3
```

```
airline_heatmap <- airline_data[ ,c(3,4,5,6)]
matrix=cor(airline_heatmap)
write.table(matrix,"coefficient_matrix.txt",sep="\t")
pheatmap(matrix,cluster_rows=F,cluster_cols=F,display_numbers=T)
```



Based on the Heatmap, we can get an idea of the correlation between each of the variables. As evident, most of the variables either have a strong positive correlation or a weak positive correlation. Intuitively, each of these variables should have a degree of correlation as they are strongly related to overall costs. PF and LF have a lower correlation most likely due to extraneous factors that differ from airline to airline and change the bottom line.

Question 1, Part 3:

```
#POOLED MODEL

library(readr)
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
library(plm) # Linear Models for Panel Data
```

```
## Warning: package 'plm' was built under R version 4.1.3
```

```
library(AER)
```

```
## Warning: package 'AER' was built under R version 4.1.3

## Loading required package: car
```

```
## Warning: package 'car' was built under R version 4.1.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.1.3

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 4.1.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.1.3

## Loading required package: survival
```

```
library(coefplot)
```

```
## Warning: package 'coefplot' was built under R version 4.1.3

## Loading required package: ggplot2
```

```
airline_data <- read_csv("PanelData.csv")
```

```
## Rows: 90 Columns: 6

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): I, T, C, Q, PF, LF
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(airline_data)
```

```
## # A tibble: 6 x 6
##       I     T       C     Q     PF    LF
##   <dbl> <dbl>   <dbl> <dbl>  <dbl> <dbl>
## 1     1     1 1140640 0.953 106650 0.534
## 2     1     2 1215690 0.987 110307 0.532
## 3     1     3 1309570 1.09  110574 0.548
## 4     1     4 1511530 1.18  121974 0.541
## 5     1     5 1676730 1.16  196606 0.591
## 6     1     6 1823740 1.17  265609 0.575
```

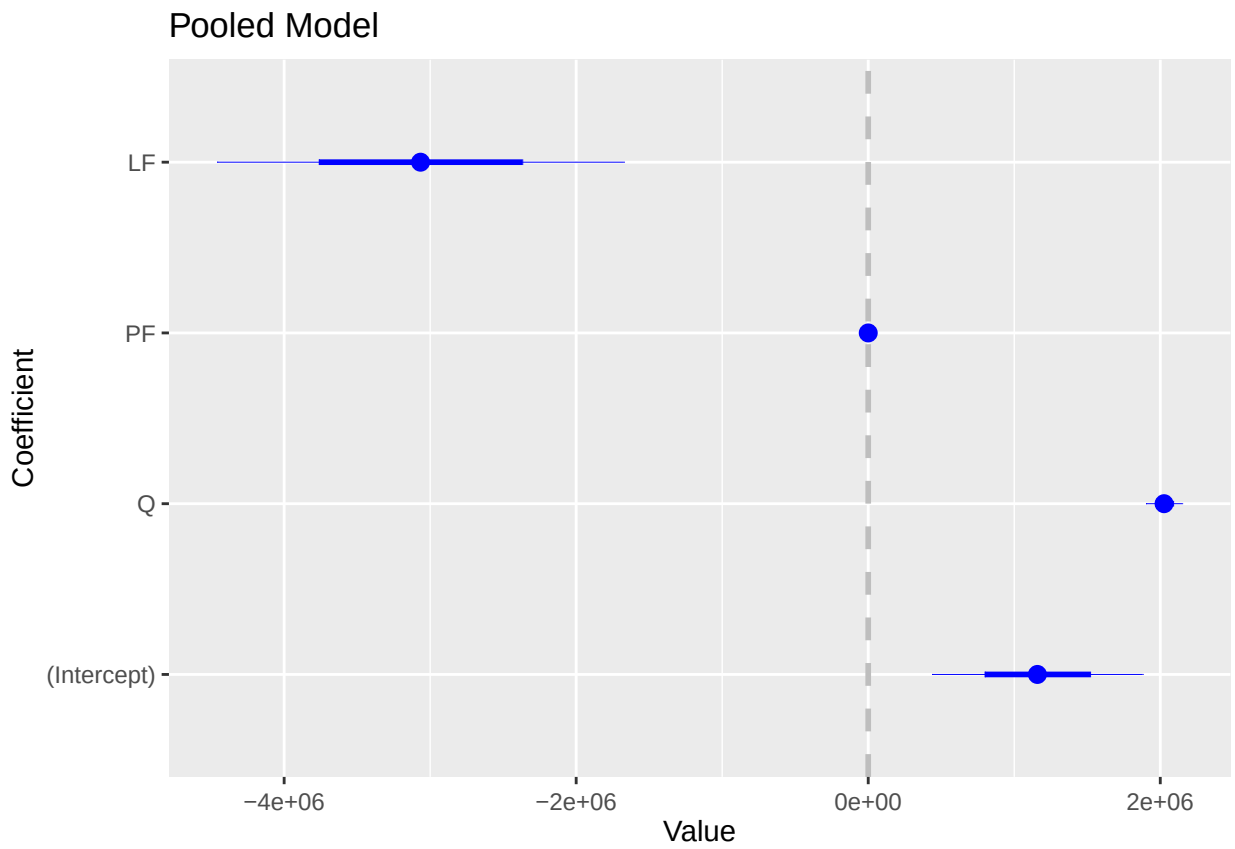
```
dim(airline_data)
```

```
## [1] 90  6
```

```
mreg.pooled <- plm(C ~ Q + PF + LF, data=airline_data, model="pooling")
summary(mreg.pooled)
```

```
## Pooling Model
##
## Call:
## plm(formula = C ~ Q + PF + LF, data = airline_data, model = "pooling")
##
## Balanced Panel: n = 6, T = 15, N = 90
##
## Residuals:
##    Min. 1st Qu.  Median 3rd Qu.    Max.
## -520654 -250270   37333  208690  849700
##
## Coefficients:
##                Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)  1.1586e+06  3.6059e+05  3.2129   0.00185 **
## Q            2.0261e+06  6.1807e+04 32.7813 < 2.2e-16 ***
## PF           1.2253e+00  1.0372e-01 11.8138 < 2.2e-16 ***
## LF          -3.0658e+06  6.9633e+05 -4.4027 3.058e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    1.2647e+14
## Residual Sum of Squares: 6.8177e+12
## R-Squared:      0.94609
## Adj. R-Squared: 0.94421
## F-statistic: 503.118 on 3 and 86 DF, p-value: < 2.22e-16
```

```
coefplot(mreg.pooled, title="Pooled Model")
```



In our Pooled Model, we have an R-Squared Value of 0.94609 and a small p-value of 2.22e-16.

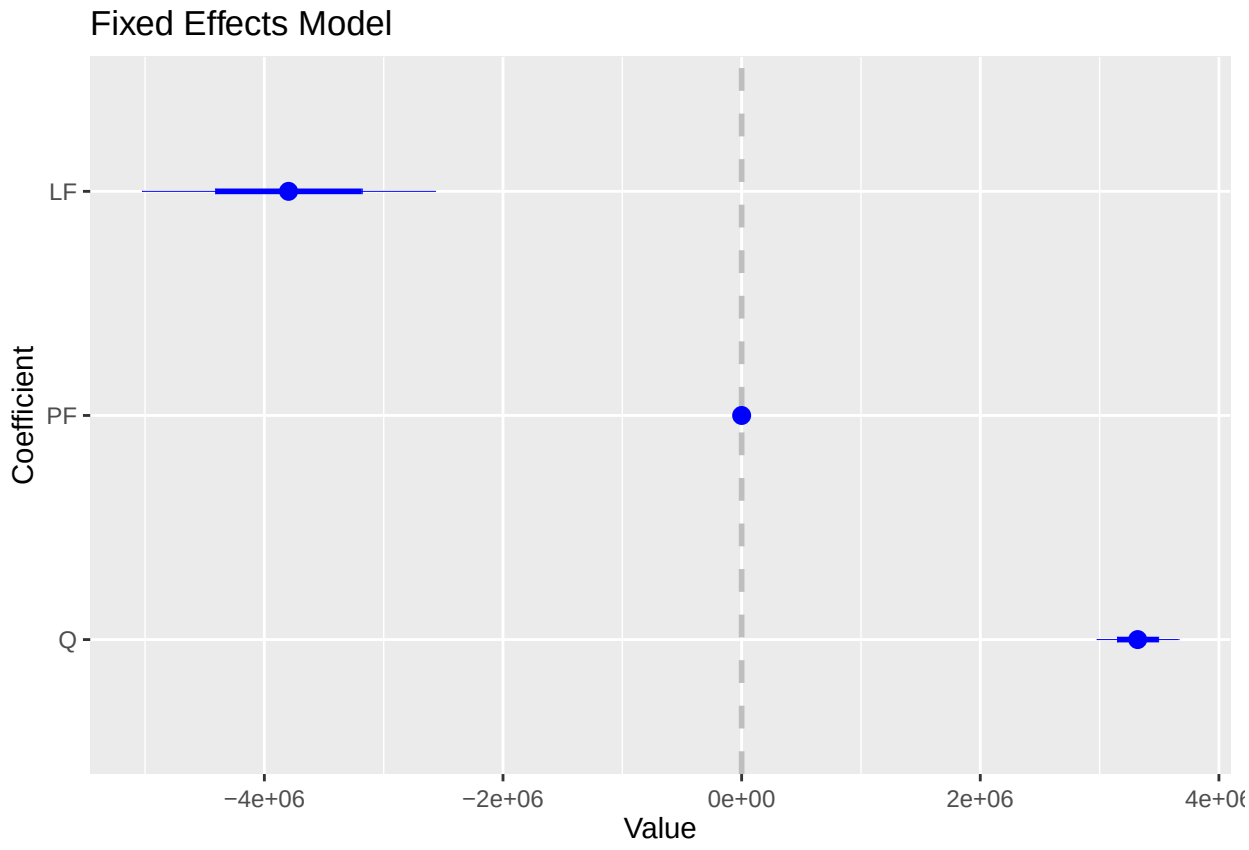
In our coefficient plot for the Pooled Model, our confidence interval of our variable LF is centered at approximately -3.1e+06, with our first lag values in between approximately -3.8e+06 to -2.2e+06, and our second lag values stretching from -4.3e+06 to -1.3e+06. Our PF variable is at 0. Our Q variable is centered around 3.1e+06 with first lag values between 1.9e+06 to 2.1e+06. Our intercept value is centered around 1.1e+06 with first lag values from 0.8e+06 to 1.3e+06. Our second lag values range from 0.5e+06 to 1.9e+06.

```
# FIXED EFFECTS MODEL
mreg.fixed <- plm(C ~ Q + PF + LF, data = airline_data, model="within")
summary(mreg.fixed)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = C ~ Q + PF + LF, data = airline_data, model = "within")
##
## Balanced Panel: n = 6, T = 15, N = 90
##
## Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -551783 -159259    1796       0  137226  499296
##
## Coefficients:
##       Estimate  Std. Error t-value  Pr(>|t|)
## Q   3.3190e+06  1.7135e+05 19.3694 < 2.2e-16 ***
```

```
## PF  7.7307e-01  9.7319e-02  7.9437 9.698e-12 ***
## LF -3.7974e+06  6.1377e+05 -6.1869 2.375e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    5.0776e+13
## Residual Sum of Squares: 3.5865e+12
## R-Squared:      0.92937
## Adj. R-Squared: 0.92239
## F-statistic: 355.254 on 3 and 81 DF, p-value: < 2.22e-16
```

```
coefplot(mreg.fixed, title="Fixed Effects Model")
```



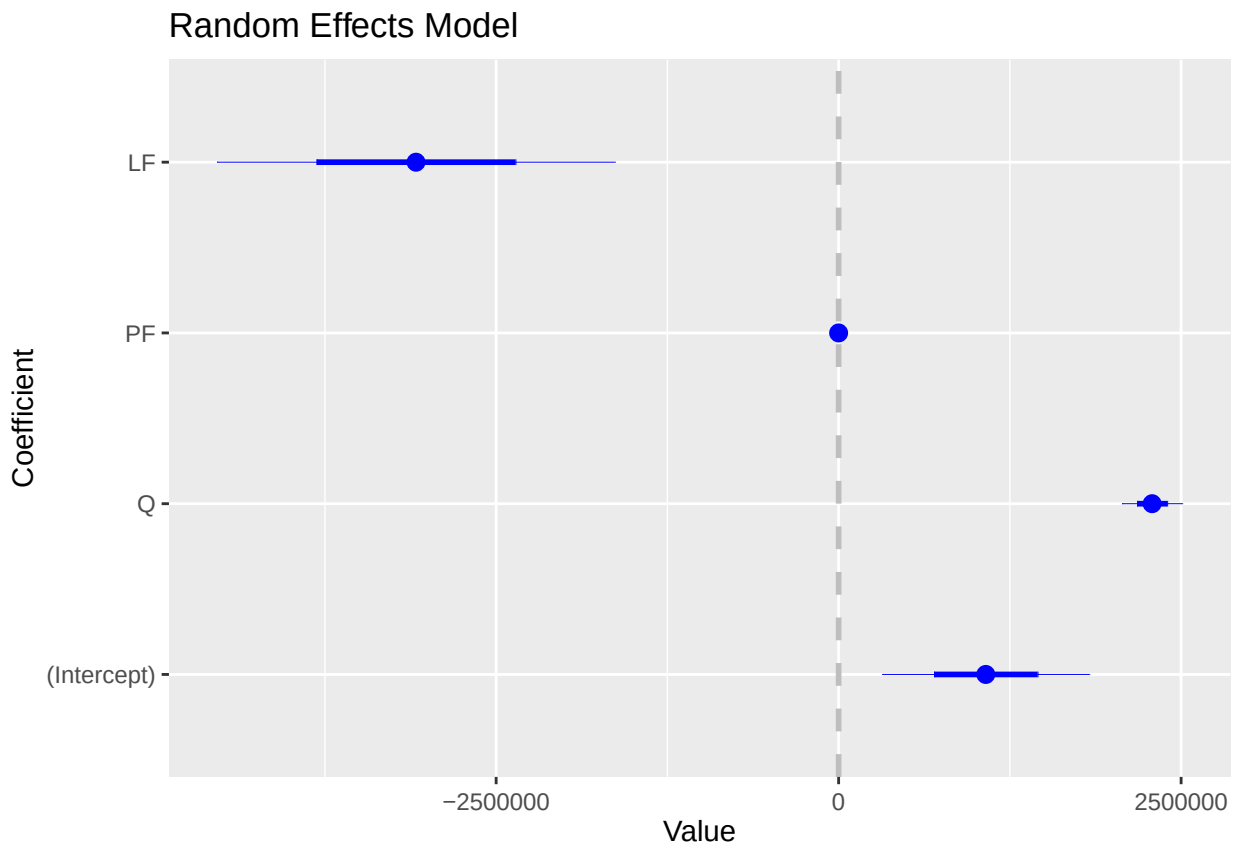

```
# In our Fixed Effects Model, we have an R-Squared value of 0.92937 and a small p-value of 2.22e-16.

# In our coefficient plot for the Fixed Effects Model, our confidence interval of our variable LF is ce
```

```
#RANDOM EFFECTS MODEL
mreg.random <- plm(C ~ Q + PF + LF, data= airline_data, model="random")
summary(mreg.random)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
```

```
## Call:
## plm(formula = C ~ Q + PF + LF, data = airline_data, model = "random")
##
## Balanced Panel: n = 6, T = 15, N = 90
##
## Effects:
##                     var   std.dev share
## idiosyncratic 4.428e+10 2.104e+05 0.793
## individual    1.154e+10 1.074e+05 0.207
## theta: 0.5486
##
## Residuals:
##    Min. 1st Qu.  Median 3rd Qu.    Max.
## -535726 -238494   49890  207491  722934
##
## Coefficients:
##                Estimate  Std. Error z-value  Pr(>|z|)
## (Intercept)  1.0743e+06  3.7747e+05  2.8461  0.004427 **
## Q            2.2886e+06  1.0949e+05 20.9015 < 2.2e-16 ***
## PF           1.1236e+00  1.0344e-01 10.8622 < 2.2e-16 ***
## LF          -3.0850e+06  7.2568e+05 -4.2512 2.126e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    6.6198e+13
## Residual Sum of Squares: 5.8721e+12
## R-Squared:      0.91129
## Adj. R-Squared: 0.9082
## Chisq: 883.501 on 3 DF, p-value: < 2.22e-16
```

```
coefplot(mreg.random, title="Random Effects Model")
```




```
# In our Random Effects Model, we have an R-Squared value of 0.91129 and a small p-value of 2.22e-16
# In our Random Effects Model plot for the Fixed Effects Model our confidence interval of our variable
```

```
#Joint F Test between Pooled and Fixed Effects Models
pFtest(mreg.fixed, mreg.pooled)
```

```
##
##  F test for individual effects
##
## data:  C ~ Q + PF + LF
## F = 14.595, df1 = 5, df2 = 81, p-value = 3.467e-10
## alternative hypothesis: significant effects
```

```
#Since F>p-value, we reject the null hypothesis that the Pooled Model is better and conclude that the F
#Hausman Test between Fixed and Random Effects Models
phtest(mreg.fixed, mreg.random)
```

```
##
##  Hausman Test
##
## data:  C ~ Q + PF + LF
## chisq = 60.87, df = 3, p-value = 3.832e-13
## alternative hypothesis: one model is inconsistent
```

```
#p-value is small, so we would reject the null that the Random Effects Model is better and conclude tha
```

Conclusions: Thus, after running all three models on the given panel data, we concluded using Hausman tests and joint F-tests that the Fixed Effects Model is our preferred model.

Question 2

Data description: This is a data set consisting of features for tracks fetched using Spotify's Web API. The tracks are labeled '1' or '0' ('Hit' or 'Flop') depending on some criteria of the author. This data set can be used to make a classification model that predicts whether a track would be a 'Hit' or not. Out of all the variables available we decided to us energy, key, loudness, duration_ms, and chorus_hit as the dependent variables based on a purely qualitative analysis.

Source: https://www.kaggle.com/datasets/theoverman/the-spotify-hit-predictor-dataset

```
spotify_data_orginal <- read.csv("Econ104_Project3_datafile_spotify.csv")
```

```
head(spotify_data_orginal)
```

```
##                                track                      artist
## 1                        Wild Things                 Alessia Cara
## 2                          Surfboard                    Esquivel!
## 3                       Love Someone                  Lukas Graham
## 4 Music To My Ears (feat. Tory Lanez)                Keys N Krates
## 5       Juju On That Beat (TZ Anthem) Zay Hilfigerrr & Zayion McCall
## 6          Here's To Never Growing Up                  Avril Lavigne
##                                  uri danceability energy key loudness mode
## 1 spotify:track:2ZyuwVvV6Z3XJaXIFbspeE        0.741  0.626   1   -4.826    0
## 2 spotify:track:61APOtq25SCMuK0V5w2Kgp        0.447  0.247   5  -14.661    0
## 3 spotify:track:2JqnpexlO9dmvjUMCaLCLJ        0.550  0.415   9   -6.557    0
## 4 spotify:track:0cjfLhk8WJ3etPTCseKXtk        0.502  0.648   0   -5.698    0
## 5 spotify:track:1lItf5ZXJc1by9SbPeljFd        0.807  0.887   1   -3.892    1
## 6 spotify:track:0qwcGscxUHGZTgq0zcaqk1        0.482  0.873   0   -3.145    1
##   speechiness acousticness instrumentalness liveness valence   tempo
## 1      0.0886      0.02000            0.000   0.0828   0.706 108.029
## 2      0.0346      0.87100            0.814   0.0946   0.250 155.489
## 3      0.0520      0.16100            0.000   0.1080   0.274 172.065
## 4      0.0527      0.00513            0.000   0.2040   0.291  91.837
## 5      0.2750      0.00381            0.000   0.3910   0.780 160.517
## 6      0.0853      0.01110            0.000   0.4090   0.737 165.084
##   duration_ms time_signature chorus_hit sections target
## 1      188493              4   41.18681       10      1
## 2      176880              3   33.18083        9      0
## 3      205463              4   44.89147        9      1
## 4      193043              4   29.52521        7      0
## 5      144244              4   24.99199        8      1
## 6      214320              4   32.17301       12      1
```

```
names(spotify_data_orginal)
```

```
##  [1] "track"            "artist"           "uri"              "danceability"
```

```
##  [5] "energy"           "key"              "loudness"         "mode"
##  [9] "speechiness"      "acousticness"     "instrumentalness" "liveness"
## [13] "valence"          "tempo"            "duration_ms"      "time_signature"
## [17] "chorus_hit"       "sections"         "target"
```

```
# loading the data
spotify_data <- data.frame(spotify_data_orginal[, c(1, 2, 5, 6, 7, 15, 17, 19)])
```

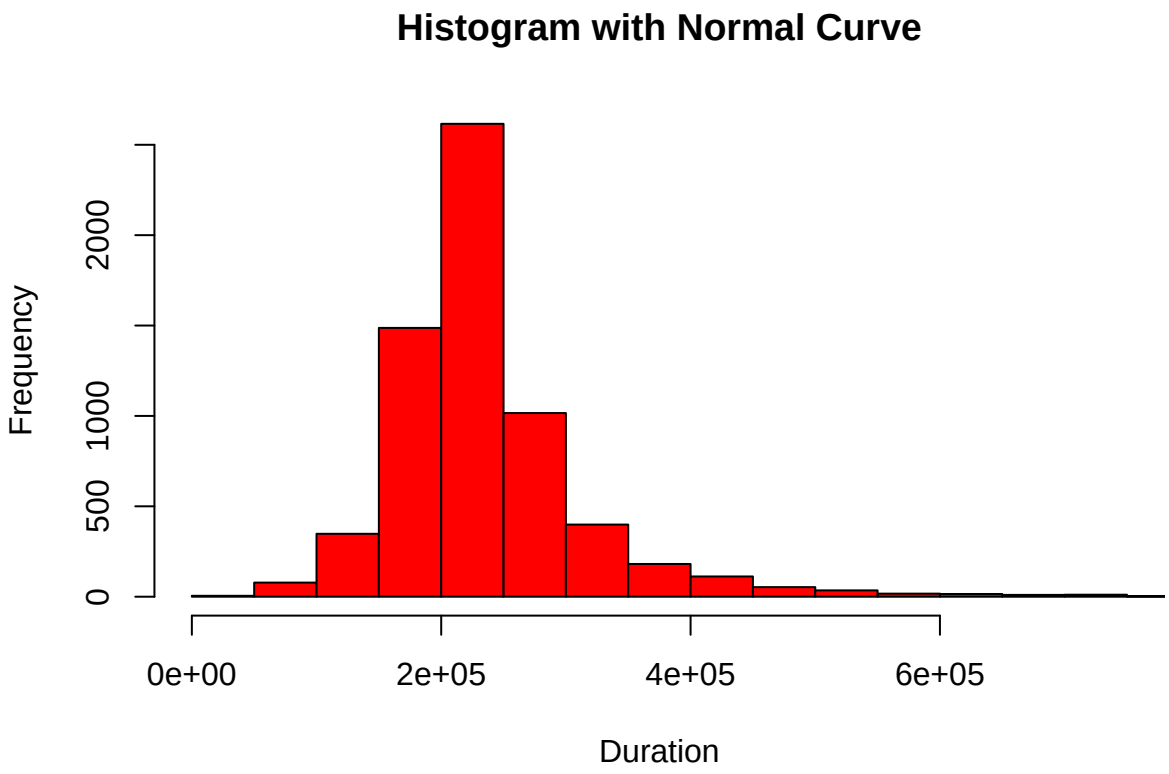
```
# checking if data has been loading correctly
names(spotify_data)
```

```
## [1] "track"       "artist"      "energy"      "key"         "loudness"
## [6] "duration_ms" "chorus_hit"  "target"
```

Question 2, Part 1:

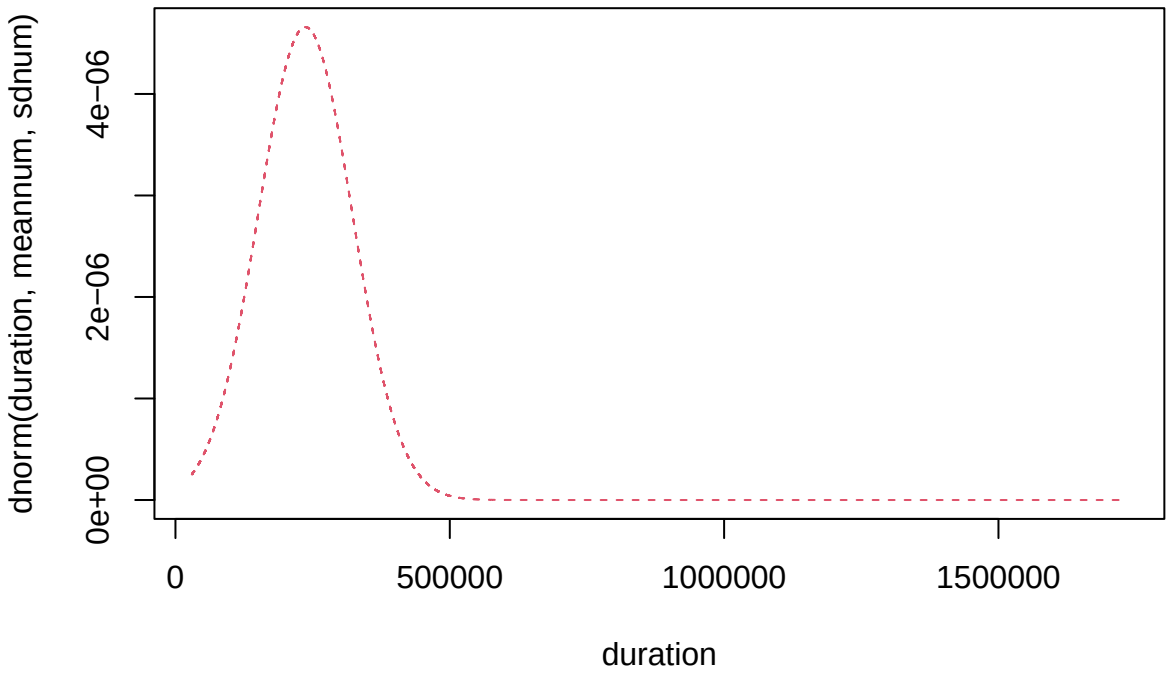
Duration: This variable measures the length of a track in milliseconds

```
#Histogram:
Duration <- spotify_data$duration_ms
Duration <- hist(Duration, breaks=25, col="red", main="Histogram with Normal Curve", xlim = c(0, 750000)
```



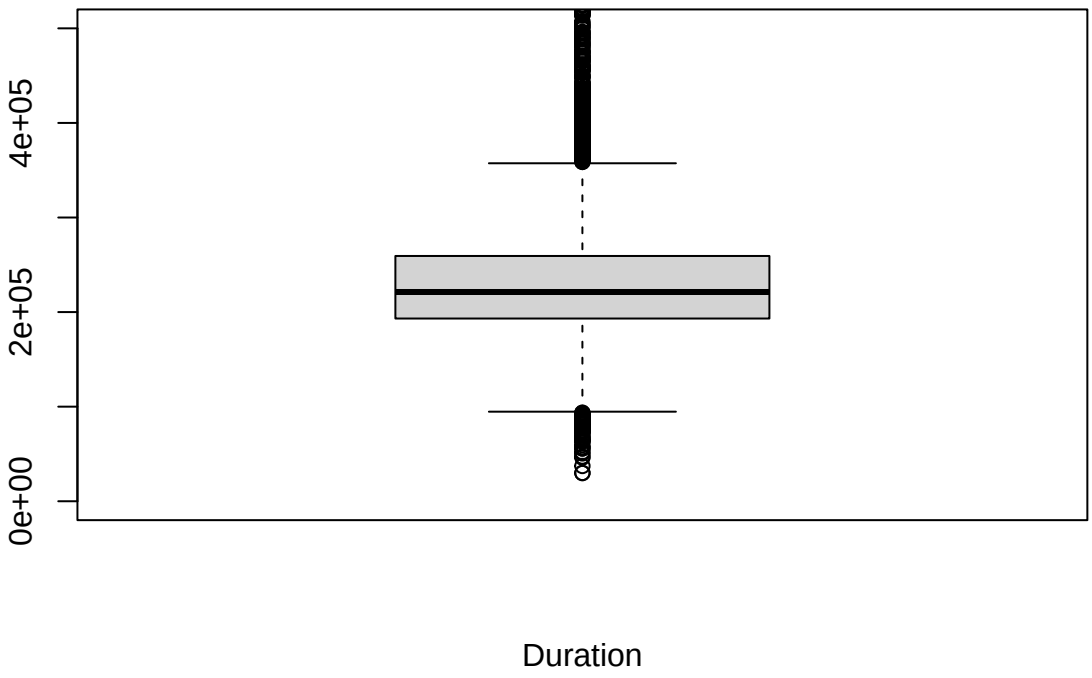
Comments: For our x variable, Duration, Histogram with a normal curve, we found that our variable's frequency peaks at approximately 2x10^5 with a frequency of over 2500. This variable appears to be approximately normal in distribution

```r
#Fitted Distribution:
meannum = mean(spotify_data$duration_ms,rm.na=TRUE)
sdnum = sd(spotify_data$duration_ms)
duration <- seq(from=min(spotify_data$duration_ms),to=max(spotify_data$duration_ms),by= 0.1)
plot(x =duration,y=dnorm(duration,meannum,sdnum),lty=2,col=2,type="l")
```



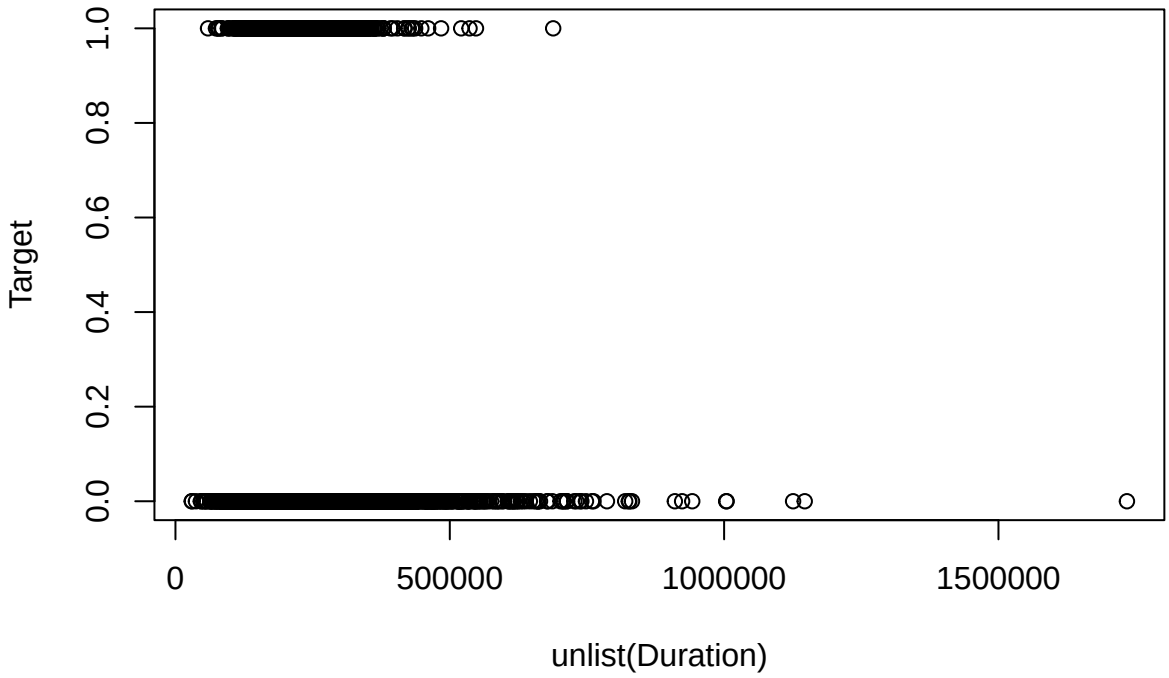
Comments: In our fitted distribution, we found that a majority of our variable is centered around 100000 to 400000. Our dnorm (d, meannum, sdnum) peaks at around 4x10^-6

```r
#Boxplot:
Duration <- spotify_data$duration_ms
boxplot(unlist(Duration), ylim = c(0, 500000), xlab="Duration")
```



Comments: The min of the plot is 1x10^5 and the max is 4x10^5, the interquartile range is approximately 2.0x10^5 to 2.6x10^5 and the median is approximately 2.3x10^5

```
#Scatterplot:
Duration <- spotify_data$duration_ms
Target <- spotify_data$target
plot(Target~unlist(Duration))
```



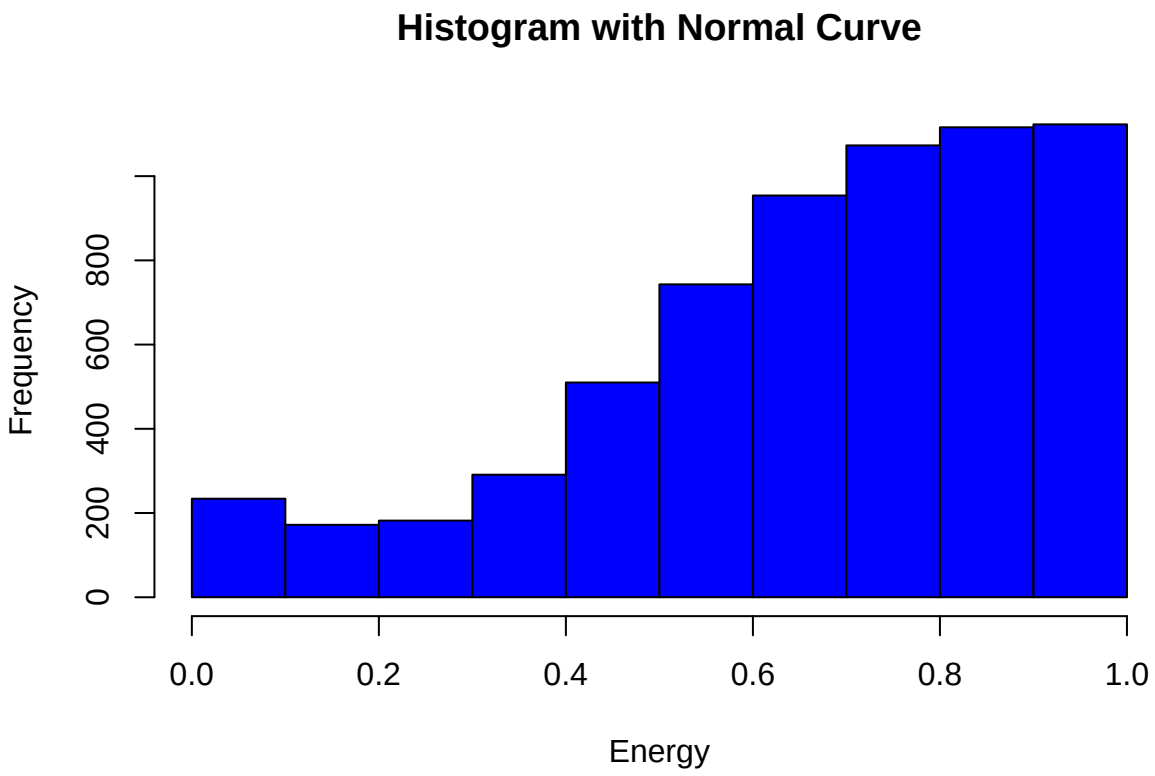
Comments: Since the scatterplot must run the variable 'Duration' against the independent variable, but the independent variable is binary, it does not convey any meaningful information

```
#Five-number Summary:
Duration <- spotify_data$duration_ms
fivenum(Duration)
```

```
## [1]   29853.0  193200.0  221246.5  259333.0 1734201.0
```

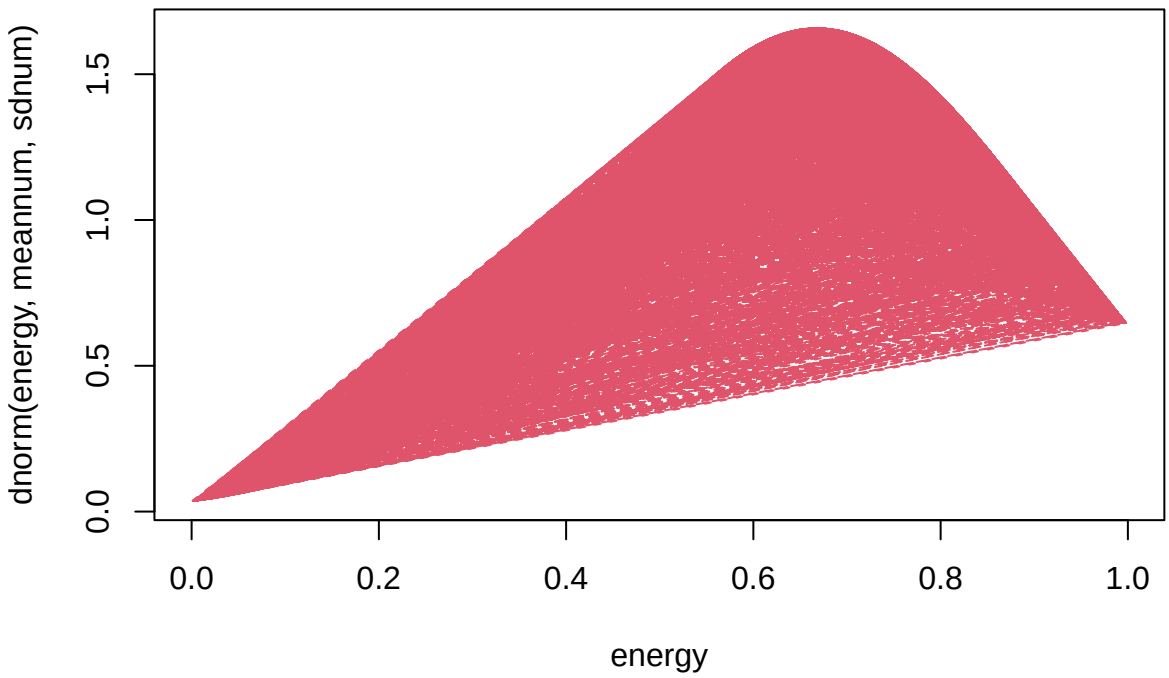
Energy - Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity

```
#Histogram:
Energy <- spotify_data$energy
Energy <- hist(Energy, breaks=10, col="blue", main="Histogram with Normal Curve")
```



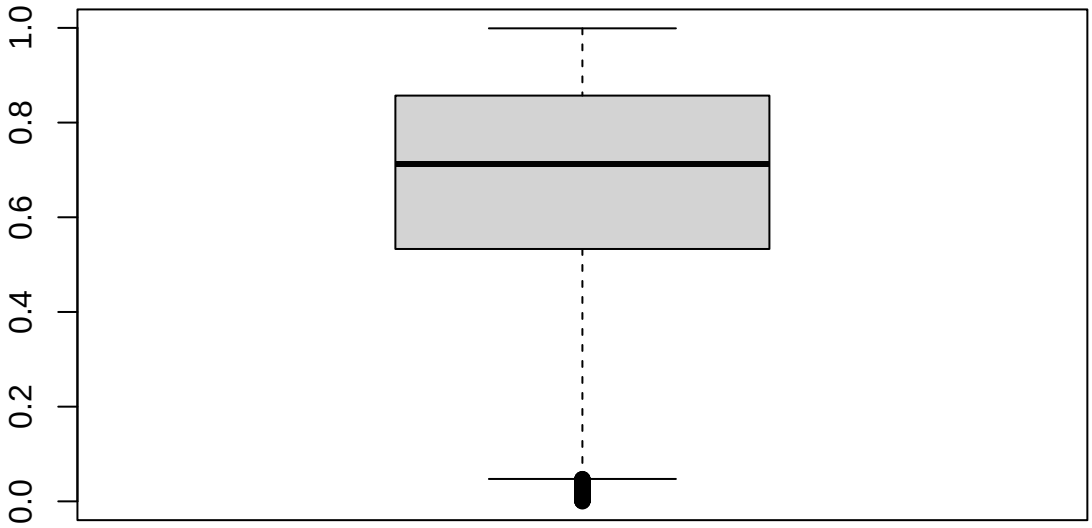
Comments: For our x variable, Energy, Histogram with a normal curve, we found that our variable's frequency peaks as it approaches 1 with over 1000 observations. This would imply that it is right distributed, but due to the nature of our data, we do not need to correct for this

```
#Fitted Distribution:
energy <- spotify_data$energy
meannum = mean(spotify_data$energy,rm.na=TRUE)
sdnum = sd(spotify_data$energy)
duration <- seq(from=min(spotify_data$energy),to=max(spotify_data$energy),by= 0.1)
plot(x =energy,y=dnorm(energy,meannum,sdnum),lty=2,col=2,type="l")
```



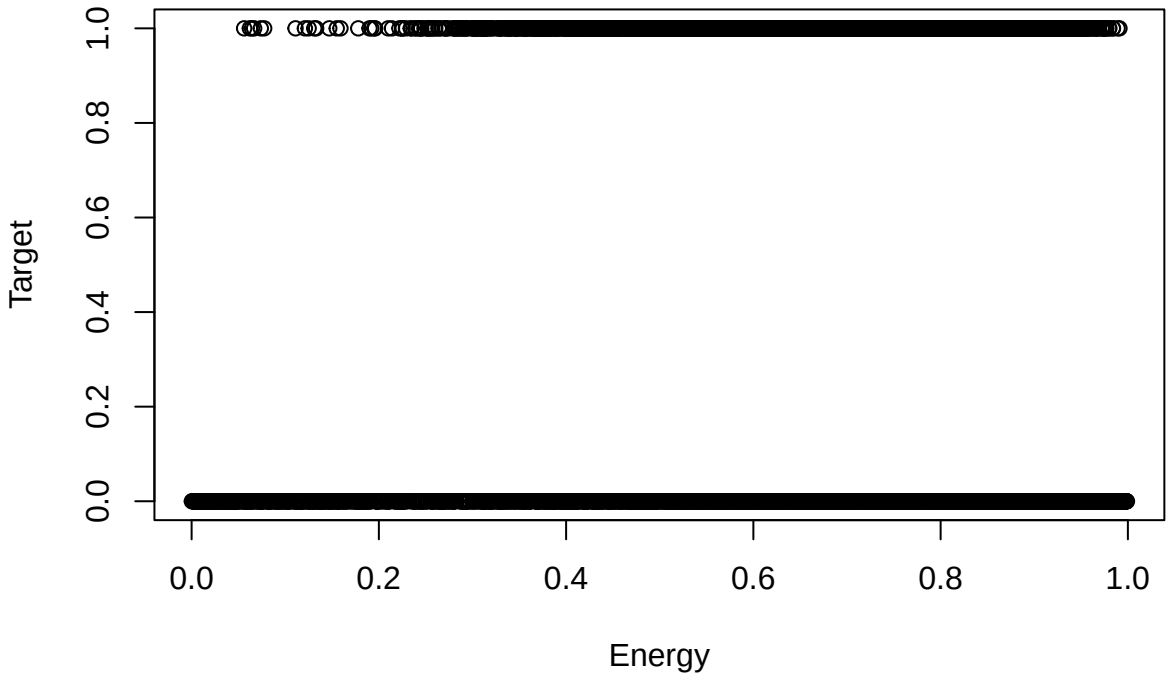
Comments: In our fitted distribution, we found that a majority of our variable is centered around 0.4 to 0.8 Our dnorm (d, meannum, sdnum) peaks at over 1.5

```
#Boxplot:
Energy <- spotify_data$energy
boxplot(unlist(energy), xlab="energy")
```



Comments: The min of the plot is 0.1 and the max is 1, the interquartile range is ~0.4 and the median is approximately 0.7

```
#Scatterplot:
Energy <- spotify_data$energy
Target <- spotify_data$target
plot(Target~Energy)
```



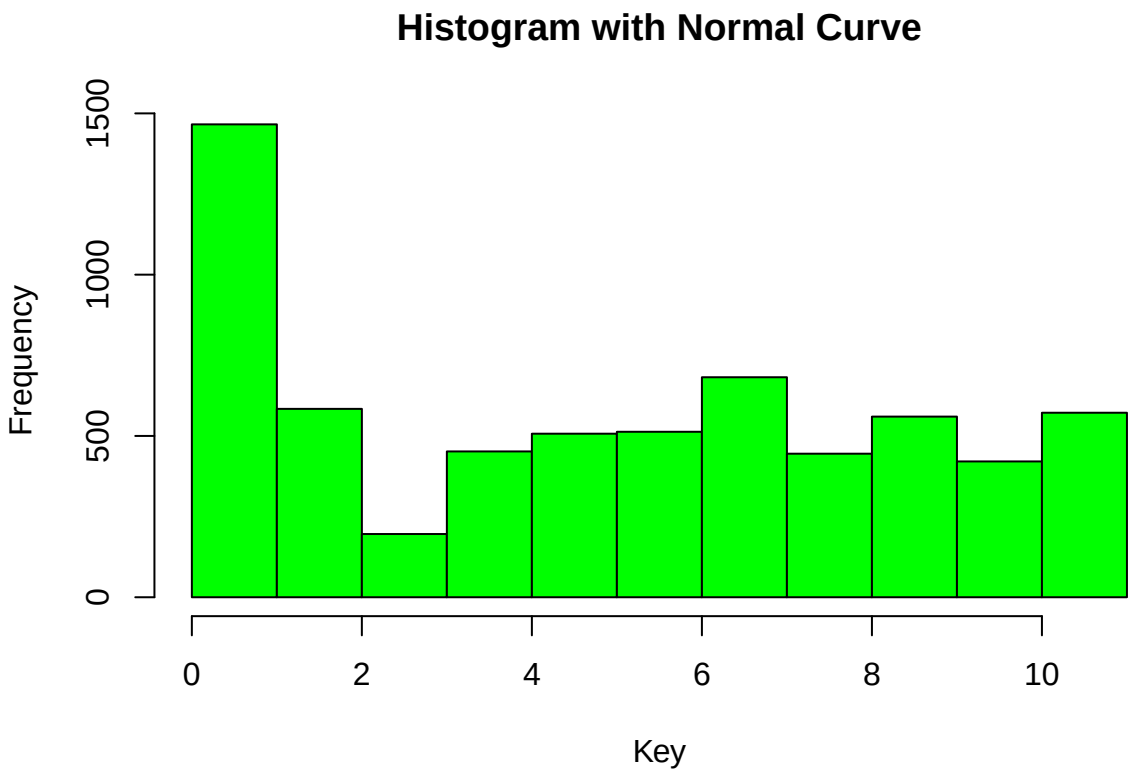
Comments: Since the scatterplot must run the variable 'Energy' against the independent variable, but the independent variable is binary, it does not convey any meaningful information

```
#Five-number Summary:
energy <- spotify_data$energy
fivenum(energy)
```

```
## [1] 0.000251 0.533000 0.712500 0.857000 0.999000
```

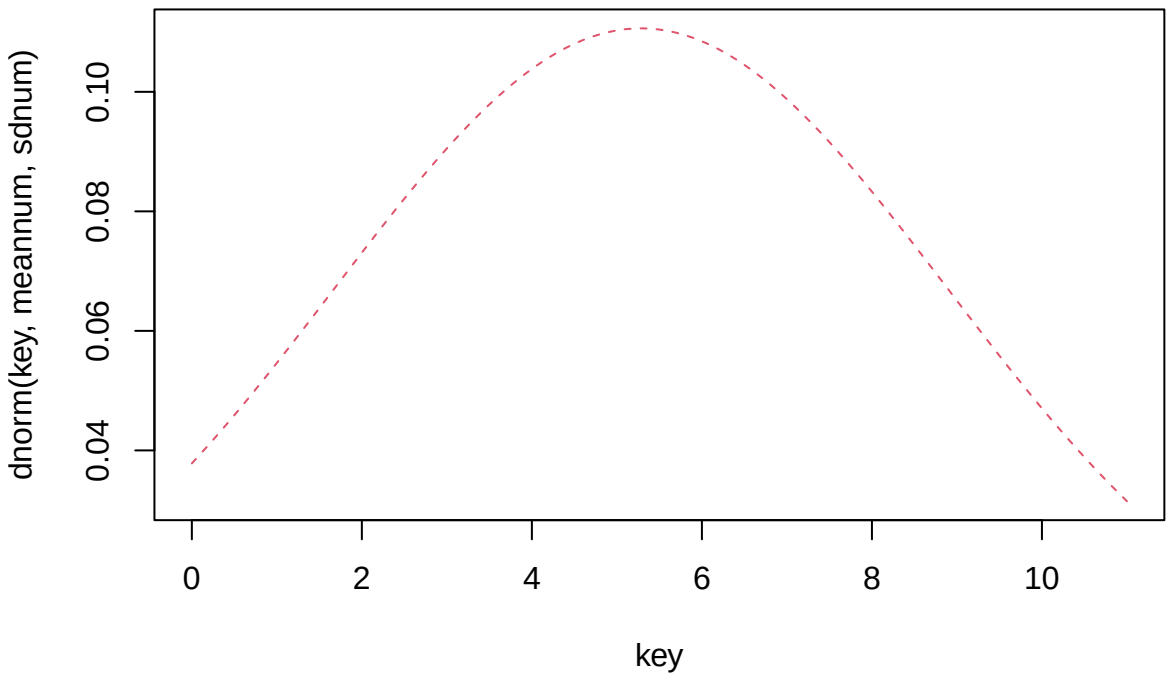
Key - This variable is the key of the track

```
#Histogram:
Key <- spotify_data$key
Key <- hist(Key, breaks=12, col="green", main="Histogram with Normal Curve")
```



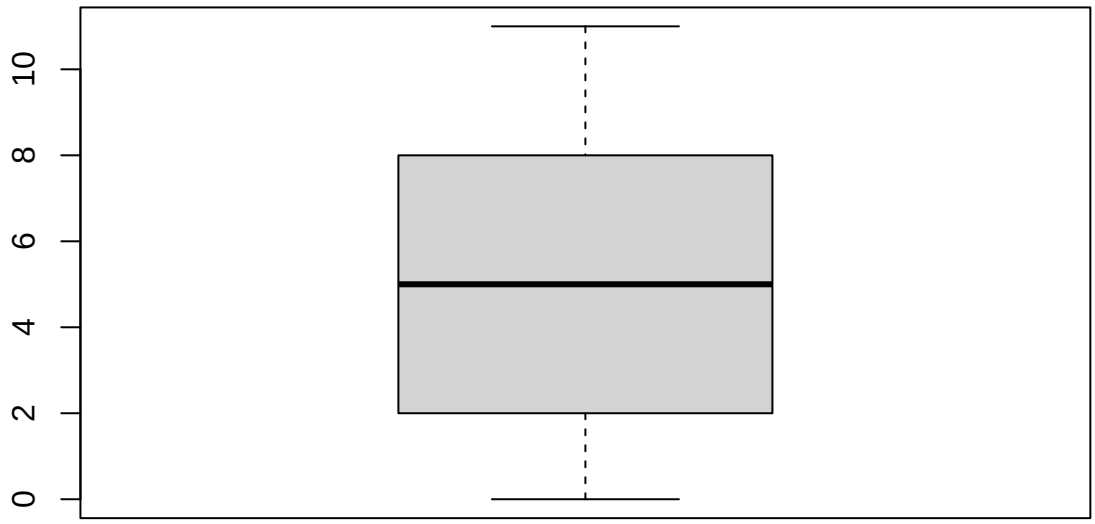
Comments: For our x variable, Key, Histogram with a normal curve, we found that our variable's frequency peaks at approximately 1 with a frequency of approximately 1500. It is evident that this variable is left skewed, however given the nature of the data, that is not a problem for us.

```
#Fitted Distribution:
key <- spotify_data$key
meannum = mean(spotify_data$key,rm.na=TRUE)
sdnum = sd(spotify_data$key)
key <- seq(from=min(spotify_data$key),to=max(spotify_data$key),by= 0.1)
plot(x =key,y=dnorm(key,meannum,sdnum),lty=2,col=2,type="l")
```



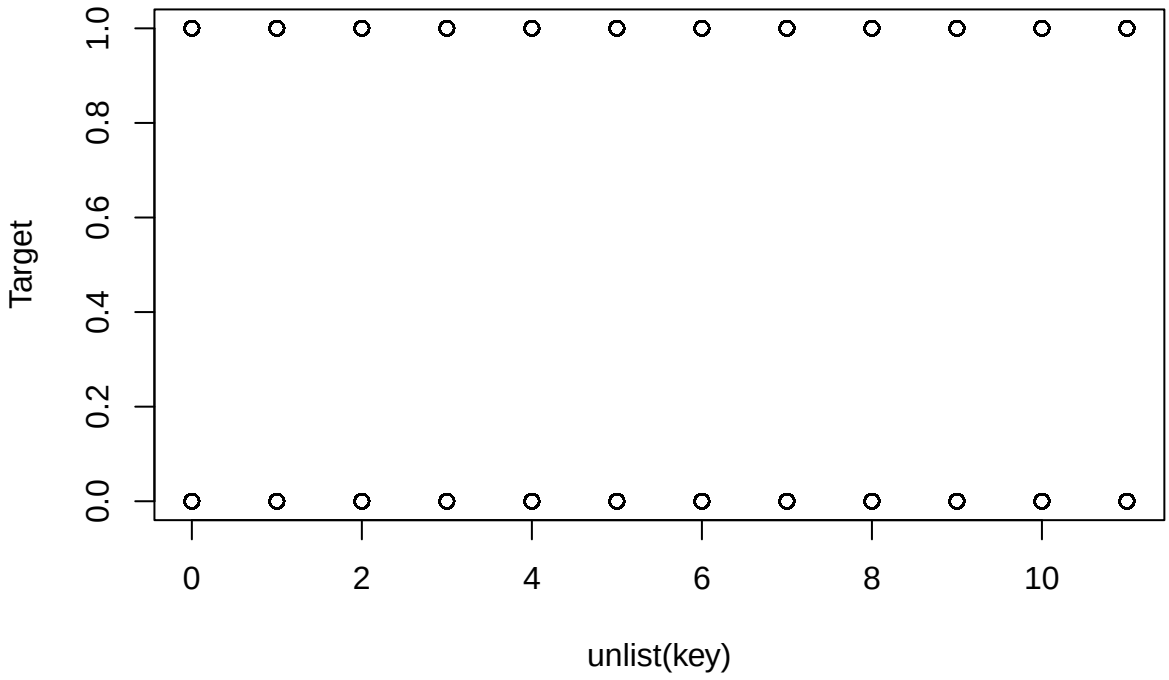
Comments: In our fitted distribution, we found that a majority of our variable is centered around 4 to 6. Our dnorm (d, meannum, sdnum) peaks at around 0.1

```
#Boxplot:
key <- spotify_data$key
boxplot(unlist(key), xlab="key")
```



The min of the plot is 0 and the max is 12, the interquartile range is 6 and the median is 5.

```
#Scatterplot:
key <- spotify_data$key
Target <- spotify_data$target
plot(Target~unlist(key))
```



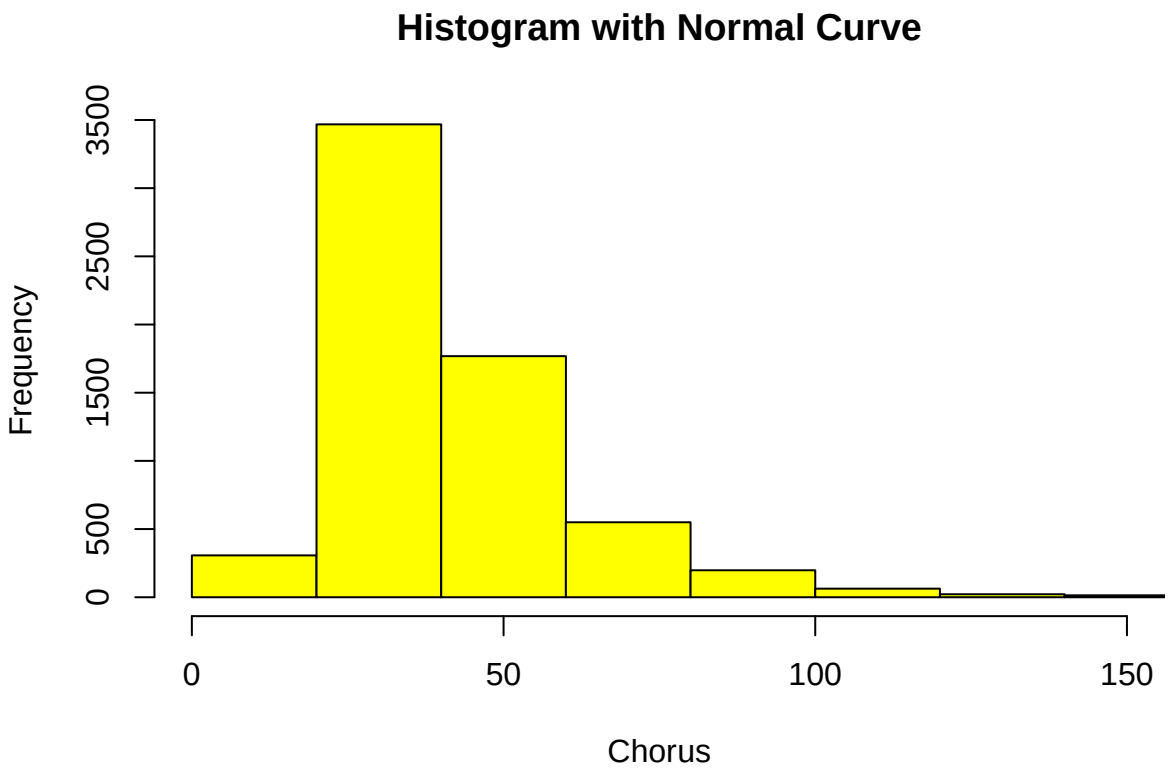
Comments: Since the scatterplot must run the variable 'Key' against the independent variable, but the independent variable is binary, it does not convey any meaningful information

```
#Five-number Summary:
key <- spotify_data$key
fivenum(key)
```

```
## [1]  0  2  5  8 11
```

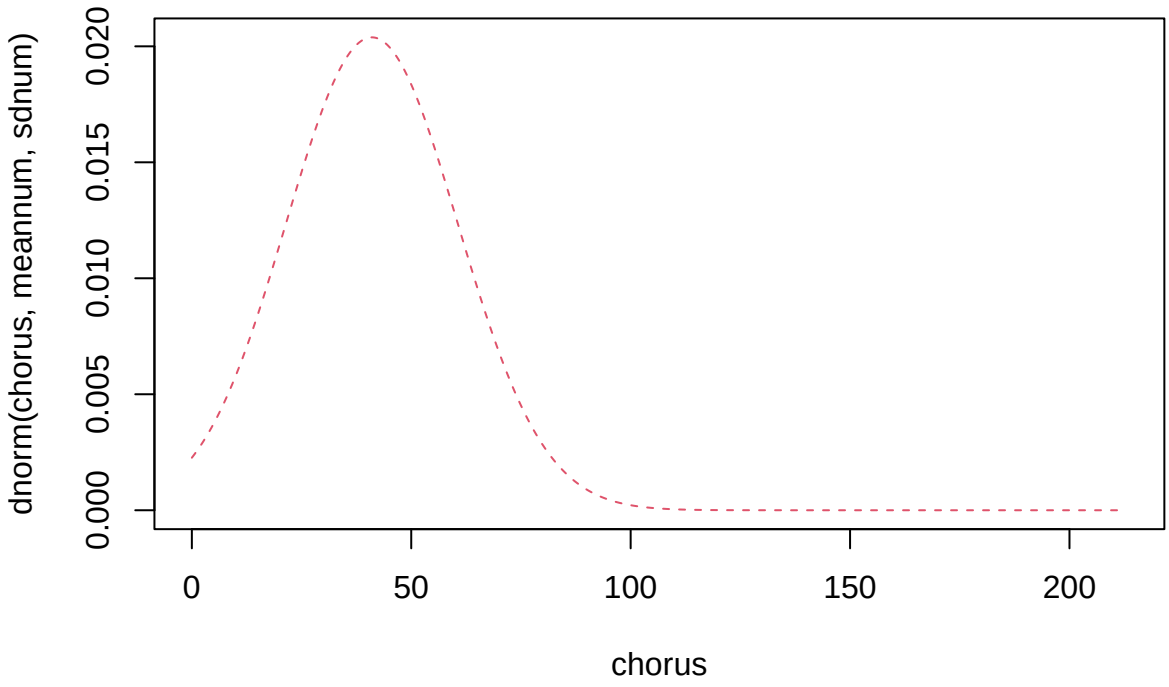
Chorus_hit - This is the author's best estimate of when the chorus would start for the track. It's the timestamp of the start of the third section of the track.

```
#Histogram:
Chorus <- spotify_data$chorus_hit
Chorus <- hist(Chorus, breaks=10, col="yellow", xlim = c(0,150), main="Histogram with Normal Curve")
```



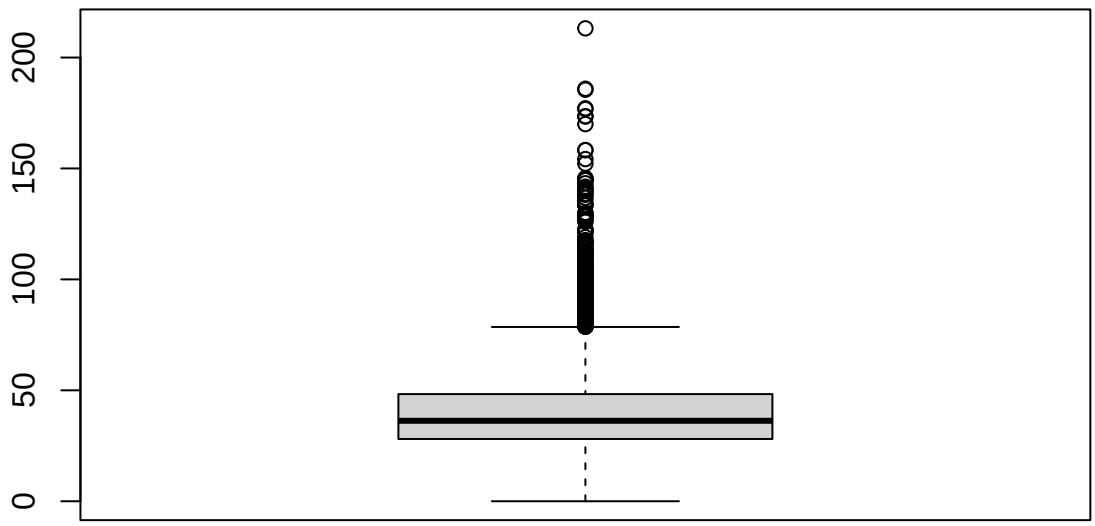
Comments: For our x variable, Chorus, Histogram with a normal curve, we found that our variable's frequency peaks at approximately 30-40 with a frequency of approximately 3500. It appears to be slightly left skewed.

```
#Fitted Distribution:
chorus <- spotify_data$chorus_hit
meannum = mean(spotify_data$chorus_hit,rm.na=TRUE)
sdnum = sd(spotify_data$chorus_hit)
chorus <- seq(from=min(spotify_data$chorus_hit),to=max(spotify_data$chorus_hit),by= 0.1)
plot(x =chorus,y=dnorm(chorus,meannum,sdnum),lty=2,col=2,type="l")
```



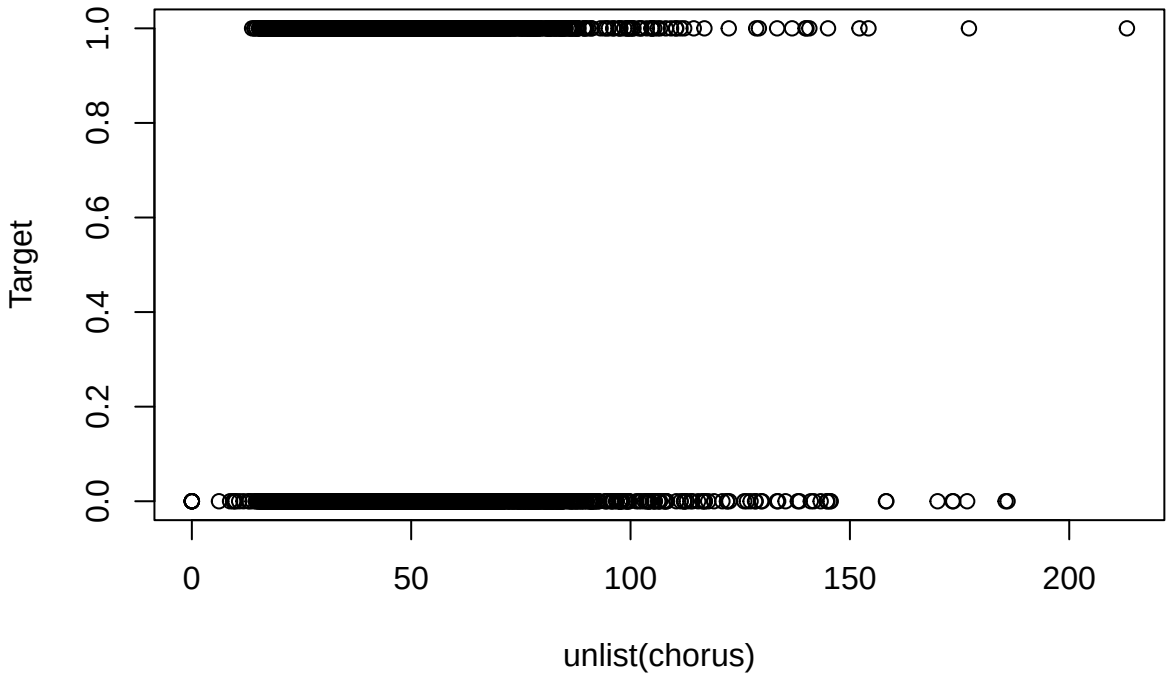
Comments: In our fitted distribution, we found that a majority of our variable is centered around 50. Our dnorm (d, meannum, sdnum) peaks at around 0.02

```
#Boxplot:
chorus <- spotify_data$chorus_hit
boxplot(unlist(chorus), xlab="chorus")
```



Comments: The min of the plot is 0 and the max is ~80, the interquartile range is ~20and the median is approximately 40

```
#Scatterplot:
chorus <- spotify_data$chorus_hit
Target <- spotify_data$target
plot(Target~unlist(chorus))
```



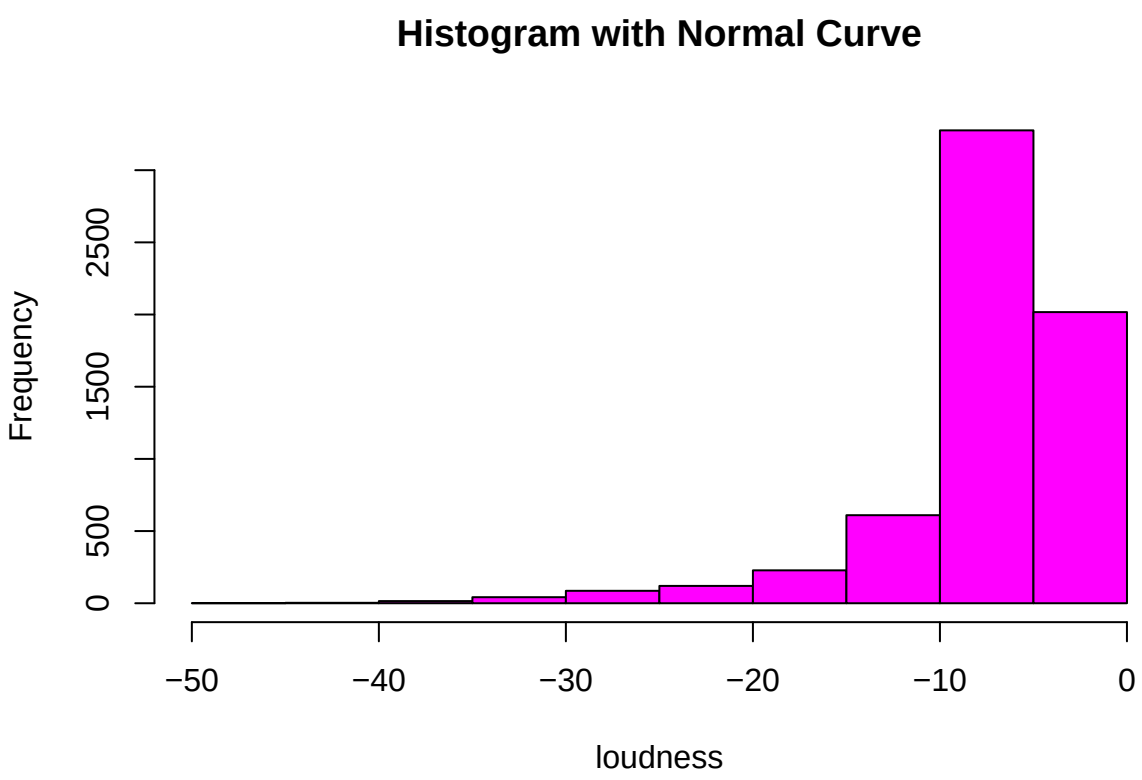
Comments: Since the scatterplot must run the variable 'Chorus' against the independent variable, but the independent variable is binary, it does not convey any meaningful information

```
#Five-number Summary:
chorus <- spotify_data$chorus_hit
fivenum(chorus)
```

```
## [1]   0.00000  28.05859  36.26537  48.29587 213.15499
```

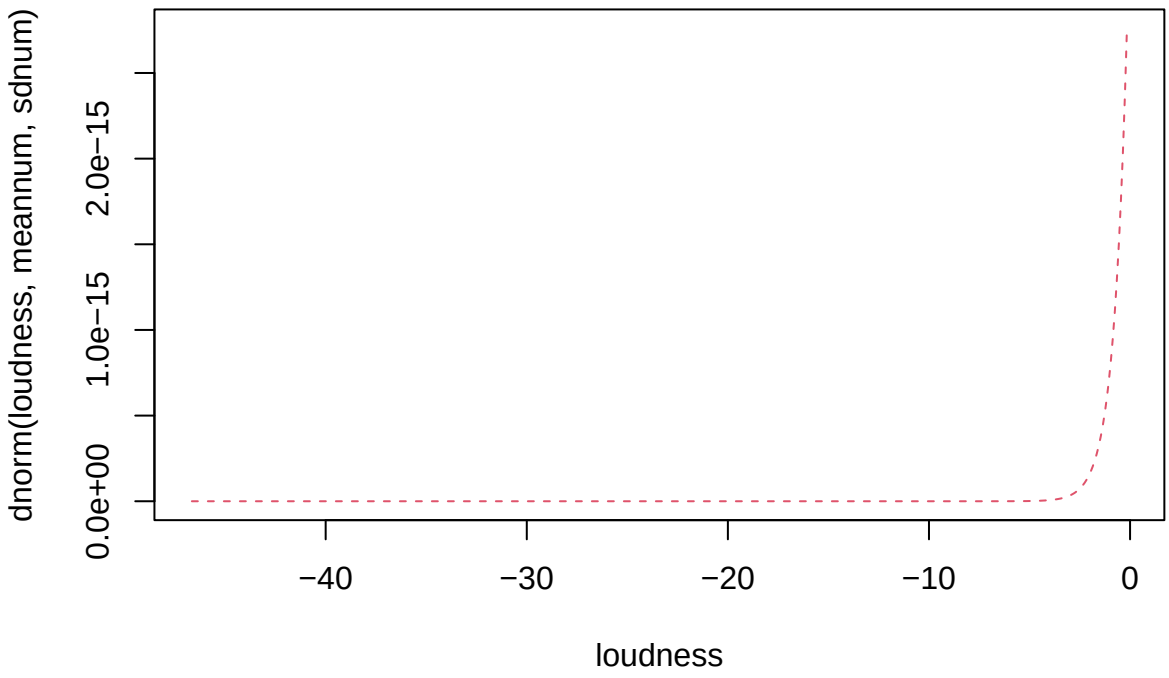
Loudness - The loudness of the track in decibels

```
#Histogram:
loudness <- spotify_data$loudness
loudness <- hist(loudness, breaks=10, col="magenta", main="Histogram with Normal Curve")
```



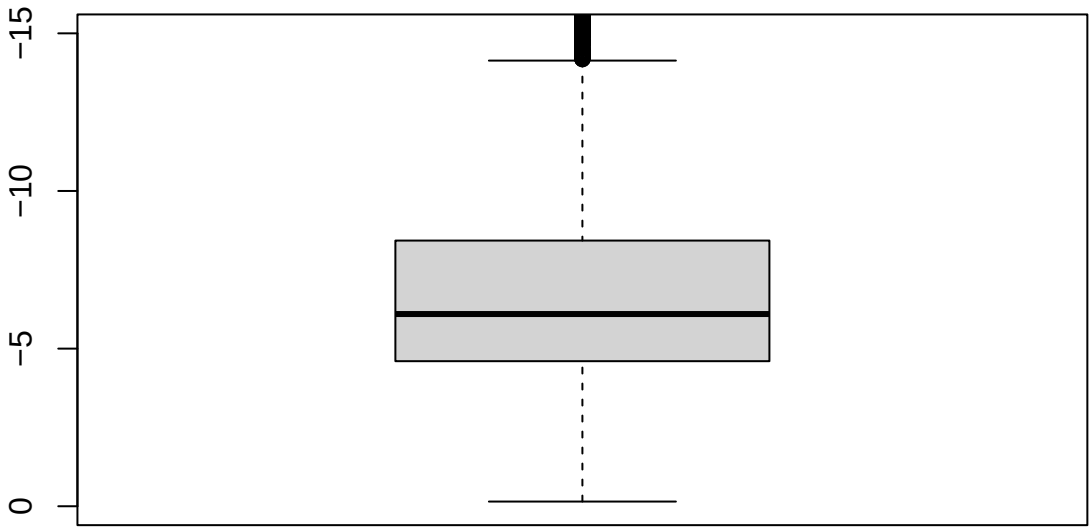
Comments: For our x variable, Loudness, Histogram with a normal curve, we found that our variable's frequency peaks at approximately 5-10 with a frequency of over 3000 It is evident that this variable is right skewed, however given the nature of the data, that is not a problem for us.

```
#Fitted Distribution:
loudness <- spotify_data$loudness
meannum = mean(spotify_data$chorus_hit,rm.na=TRUE)
sdnum = sd(spotify_data$loudness)
loudness <- seq(from=min(spotify_data$loudness),to=max(spotify_data$loudness),by= 0.1)
plot(x =loudness,y=dnorm(loudness,meannum,sdnum),lty=2,col=2,type="l")
```



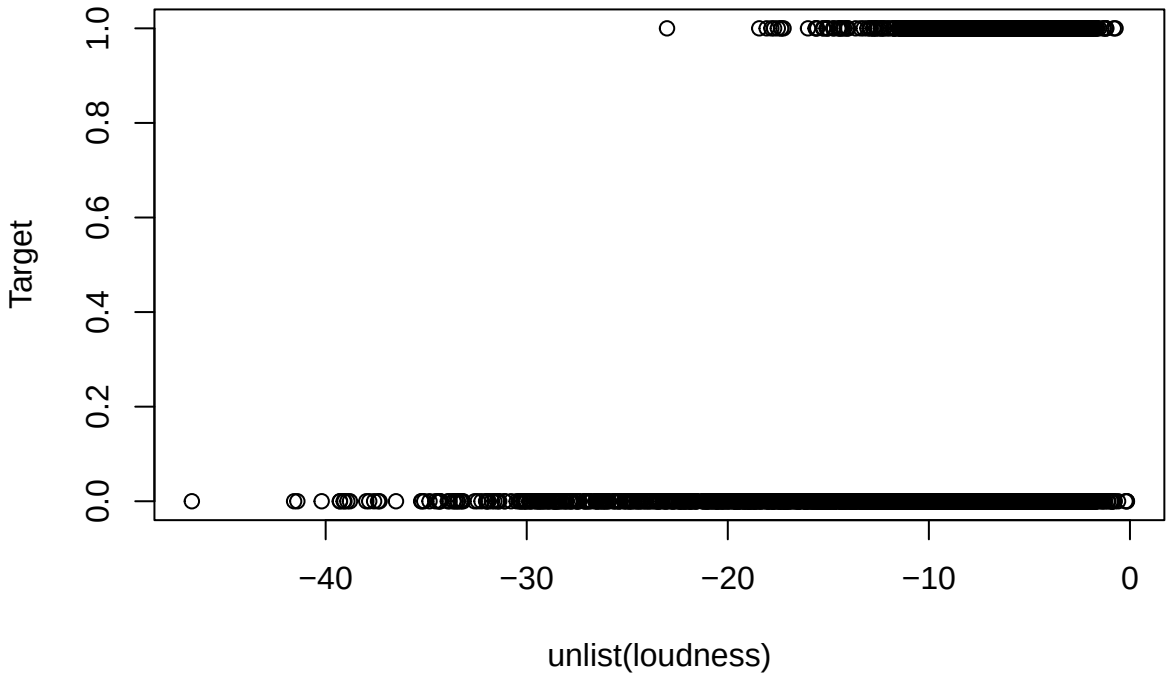
Comments: In our fitted distribution, we found that a majority of our variable is centered around 0. Our dnorm (d, meannum, sdnum) peaks at around 3x10^-15

```
#Boxplot:
loudness <- spotify_data$loudness
boxplot(unlist(loudness), xlab="loudness",ylim = c(0,-15))
```



Comments: The min of the plot is 0 and the max is 14, the interquartile range is 3 and the median is approximately 6

```
#Scatterplot:
loudness <- spotify_data$loudness
Target <- spotify_data$target
plot(Target~unlist(loudness))
```

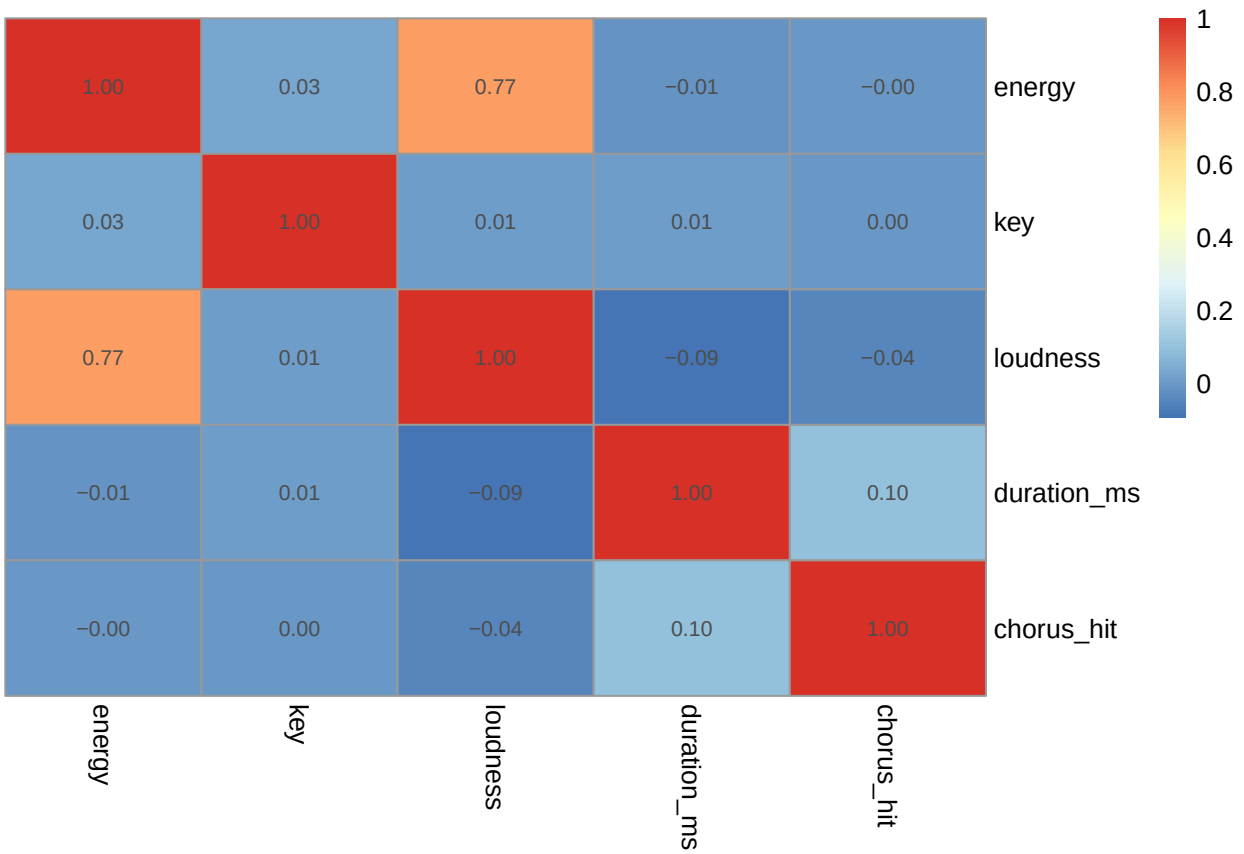


Comments: Since the scatterplot must run the variable 'Loudness' against the independent variable, but the independent variable is binary, it does not convey any meaningful information

```
#Five-number Summary:
loudness <- spotify_data$loudness
fivenum(loudness)
```

```
## [1] -46.6550  -8.4270  -6.0965  -4.6010  -0.1490
```

```
#Correlation
library(pheatmap)
spotify_heatmap <- spotify_data_orginal[ ,c(5,6,7,15,17)]
matrix=cor(spotify_heatmap)
write.table(matrix,"coefficient_matrix.txt",sep="\t")
pheatmap(matrix,cluster_rows=F,cluster_cols=F,display_numbers=T)
```



Question 2, Part 3:

(a) Linear probability model:

```
#running the linear probability model
lpm_spotify <- lm(target ~ energy + key + loudness + duration_ms + chorus_hit, data = spotify_data)
summary(lpm_spotify)
```

```
##
## Call:
## lm(formula = target ~ energy + key + loudness + duration_ms +
##     chorus_hit, data = spotify_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0086 -0.4342  0.1768  0.3951  1.6608
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.834e+00  3.876e-02  47.322  < 2e-16 ***
## energy      -9.537e-01  3.654e-02 -26.101  < 2e-16 ***
## key          1.797e-03  1.534e-03   1.172    0.241
## loudness     6.391e-02  1.688e-03  37.860  < 2e-16 ***
## duration_ms -7.352e-07  6.543e-08 -11.238  < 2e-16 ***
```

```
## chorus_hit  -1.164e-03  2.842e-04  -4.095 4.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4421 on 6392 degrees of freedom
## Multiple R-squared:  0.219,  Adjusted R-squared:  0.2184
## F-statistic: 358.5 on 5 and 6392 DF,  p-value: < 2.2e-16
```

This shows that in our linear model, our coefficients show that energy, loudness, duration, and chorus_hit are significant in determining whether or not a song is a hit, and this 21.9% of the dependent variable outcomes are explained by this model.

```
# running the probit model:
probit_spotify <- glm(target ~ energy + key + loudness + duration_ms + chorus_hit, family=binomial(link=
data = spotify_data)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(probit_spotify)
```

```
##
## Call:
## glm(formula = target ~ energy + key + loudness + duration_ms +
##     chorus_hit, family = binomial(link = "probit"), data = spotify_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7303  -0.9306   0.1021   0.9276   3.3471
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.429e+00  1.627e-01  33.368   <2e-16 ***
## energy      -3.771e+00  1.370e-01 -27.529   <2e-16 ***
## key          4.972e-03  4.842e-03   1.027   0.3044
## loudness     3.074e-01  9.406e-03  32.678   <2e-16 ***
## duration_ms -2.685e-06  2.568e-07 -10.459   <2e-16 ***
## chorus_hit  -2.814e-03  9.308e-04  -3.023   0.0025 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8869.5  on 6397  degrees of freedom
## Residual deviance: 6864.0  on 6392  degrees of freedom
## AIC: 6876
##
## Number of Fisher Scoring iterations: 6
```

```
# running the logit model:
logit_spotify <- glm(target ~ energy + key + loudness + duration_ms + chorus_hit, family=binomial(link=
data = spotify_data)

summary(logit_spotify)
```

```
##
## Call:
## glm(formula = target ~ energy + key + loudness + duration_ms +
##     chorus_hit, family = binomial(link = "logit"), data = spotify_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6305  -0.9151   0.1195   0.9074   3.0423
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.400e+00  2.950e-01  31.863  < 2e-16 ***
## energy      -6.541e+00  2.431e-01 -26.907  < 2e-16 ***
## key          8.827e-03  8.125e-03   1.086  0.27733
## loudness     5.316e-01  1.711e-02  31.078  < 2e-16 ***
## duration_ms -4.594e-06  4.480e-07 -10.253  < 2e-16 ***
## chorus_hit  -4.969e-03  1.570e-03  -3.165  0.00155 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8869.5  on 6397  degrees of freedom
## Residual deviance: 6842.5  on 6392  degrees of freedom
## AIC: 6854.5
##
## Number of Fisher Scoring iterations: 5
```

Conclusion: The logit model has a lower AIC than the probit model, and thus it explains the data better than the probit model.