```r
# Assignment: ASSIGNMENT 3
# Name: Ramani, Aarti
# Date: 2022-12-14

# Assignment: ASSIGNMENT 3.2
# Name: Ramani, Aarti
# Date: 2022-12-14

## Load the ggplot2 package
library(ggplot2)

## Set the working directory to the root of DSC 520 directory
setwd("C:/Masters/GitHub/Winter2022/Ramani-DSC520")

#List the name of each field and what you believe the data type and intent is of
#the data included in each field (Example: Id - Data Type: varchar
#(contains text and numbers) Intent: unique identifier for each row)

## Load the `data/r4ds/heights.csv` to
heights_df <- read.csv("data/acs-14-1yr-s0201.csv")


colnames(heights_df)
```

```
## [1] "Id"                     "Id2"                    "Geography"
## [4] "PopGroupID"             "POPGROUP.display.label" "RacesReported"
## [7] "HSDegree"               "BachDegree"
```

```r
#Run the following functions and provide the results: str(); nrow(); ncol()
str(heights_df)
```

```
## 'data.frame':    136 obs. of  8 variables:
##  $ Id                    : chr  "0500000US01073" "0500000US04013" "0500000US04019" "0500000US06001"
##  $ Id2                   : int  1073 4013 4019 6001 6013 6019 6029 6037 6059 6065 ...
##  $ Geography             : chr  "Jefferson County, Alabama" "Maricopa County, Arizona" "Pima County,
##  $ PopGroupID            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ POPGROUP.display.label: chr  "Total population" "Total population" "Total population" "Total popul
##  $ RacesReported         : int  660793 4087191 1004516 1610921 1111339 965974 874589 10116705 314551
##  $ HSDegree              : num  89.1 86.8 88 86.9 88.8 73.6 74.5 77.5 84.6 80.6 ...
##  $ BachDegree            : num  30.5 30.2 30.8 42.8 39.7 19.7 15.4 30.3 38 20.7 ...
```

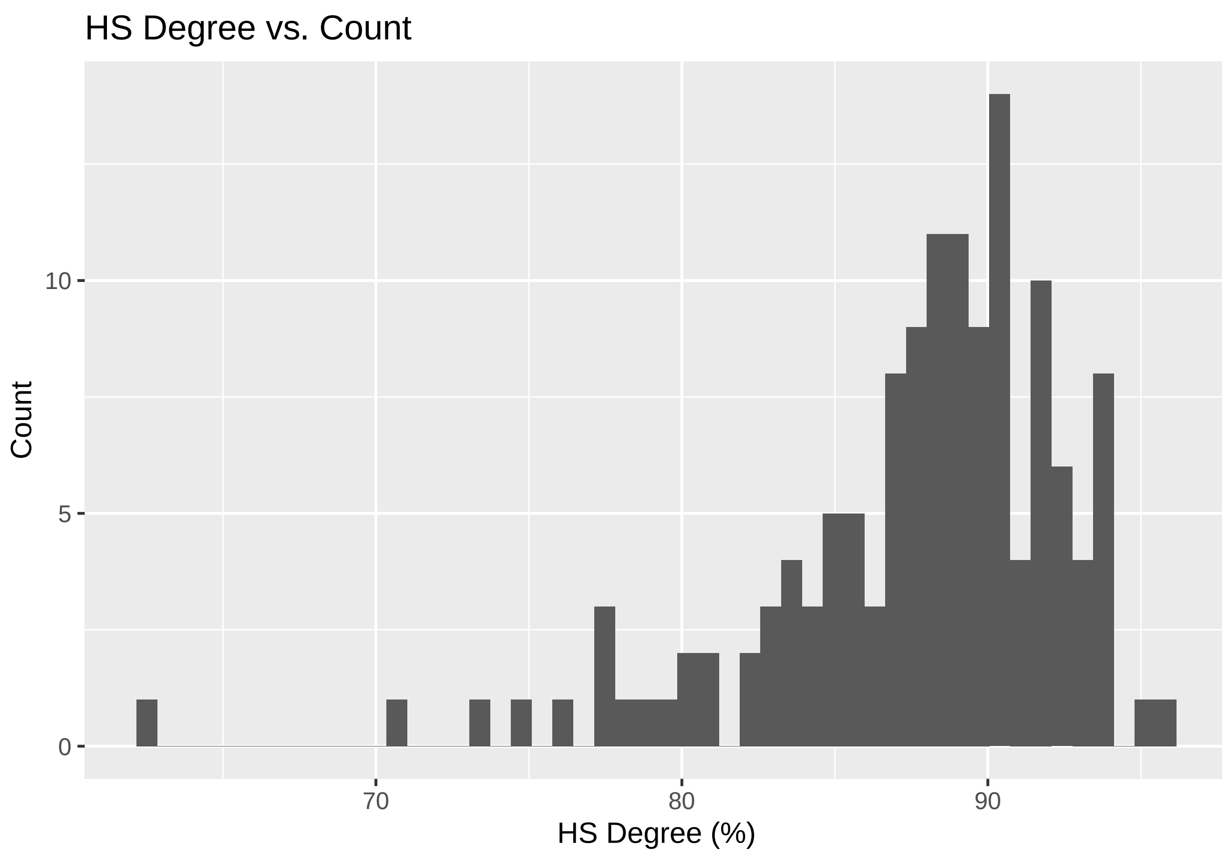
```r
nrow(heights_df)
```

```
## [1] 136
```

```r
ncol(heights_df)
```

```
## [1] 8
```

```r
#Create a Histogram of the HSDegree variable using the ggplot2 package.
#Set a bin size for the Histogram that you think best visuals the data
#(the bin size will determine how many bars display and how wide they are)
#Include a Title and appropriate X/Y axis labels on your Histogram Plot.
library(ggplot2)
ggplot(heights_df, aes(HSDegree)) + geom_histogram(bins=50) + ggtitle("HS Degree vs. Count") + xlab("HS
```




```r
#Answer the following questions based on the Histogram produced:
#Based on what you see in this histogram, is the data distribution unimodal?
# > This is  a unimodal distribution since it only has one peak

# > Standard deviation = 5.117941
# sd(heights_df$HSDegree)

#  Is it approximately symmetrical?
# > No, the histogram is not symmetrical. The left and right sides are not symmetrical.

#  Is it approximately bell-shaped?
# > The plot looks bell shaped but is skewed and unsymmetrical.

#  Is it approximately normal?
# > shapiro.test(heights_df$HSDegree) W = 0.87736, p-value = 3.194e-09. p<.001 - Not normal
shapiro.test(heights_df$HSDegree)
```

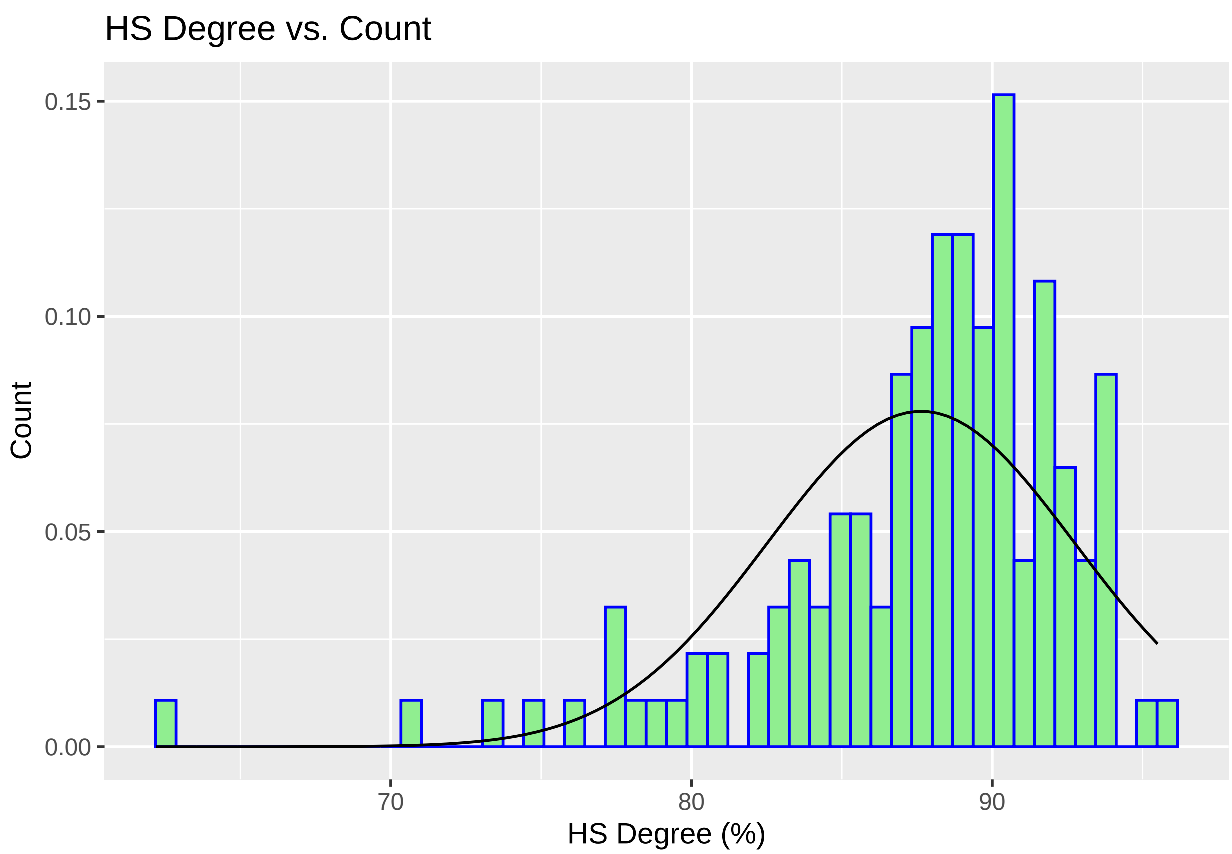
```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  heights_df$HSDegree
## W = 0.87736, p-value = 3.194e-09
```

```
#  If not normal, is the distribution skewed? If so, in which direction?
# > The histogram is left or negatively skewed since the mean is lower than the median and most of the

#  Include a normal curve to the Histogram that you plotted.
#Explain whether a normal distribution can accurately be used as a model for this data.
# > The graph is skewed and does not qualify for normal distribution.
# > For a normal distribution, 1 to 100% of area of the plot should be under the normal curve.
ggplot(heights_df, aes(HSDegree))+ geom_histogram(aes(y=..density..),color = "blue",bins=50,fill="light
  stat_function(fun = dnorm,args = list(mean = mean(heights_df$HSDegree),sd = sd(heights_df$HSDegree)),
  ggtitle("HS Degree vs. Count") + xlab("HS Degree (%)") + ylab("Count")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
```

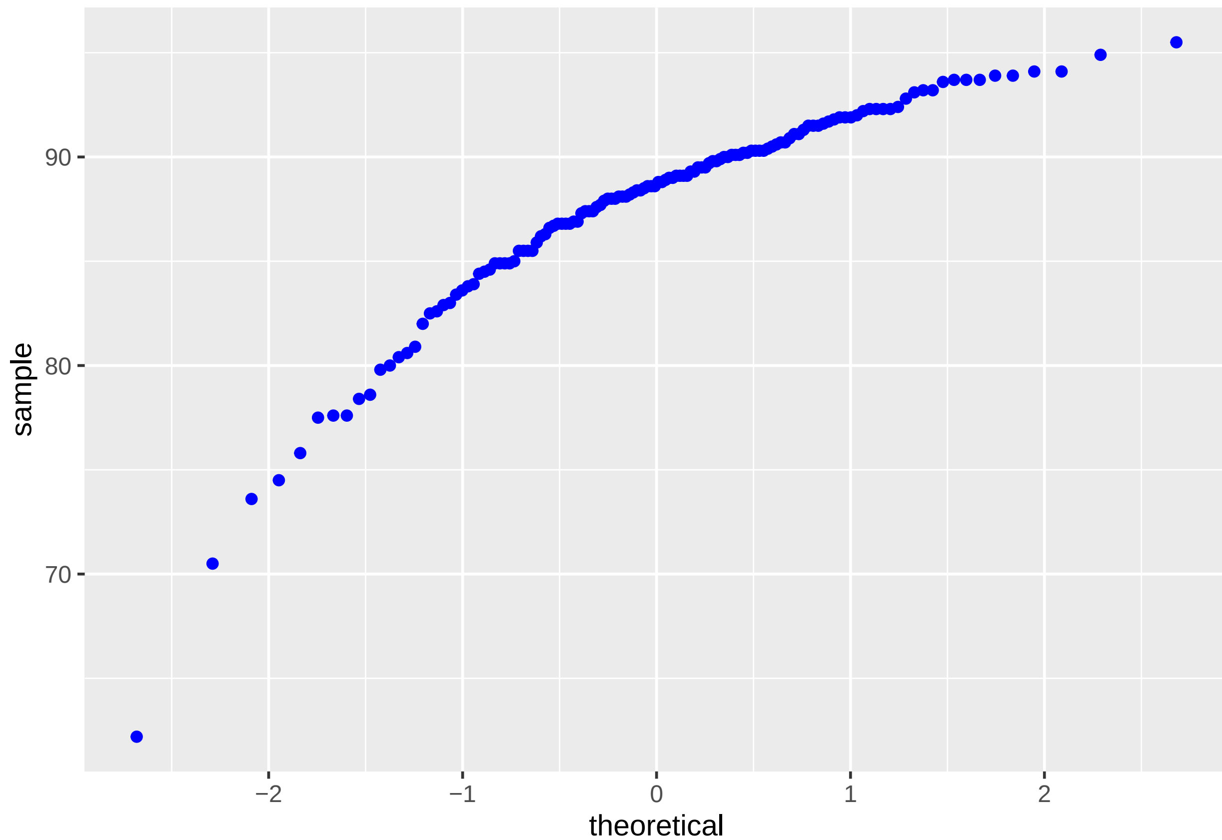



```
#Create a Probability Plot of the HSDegree variable.
library(qqplotr)
```

```
##
## Attaching package: 'qqplotr'
```

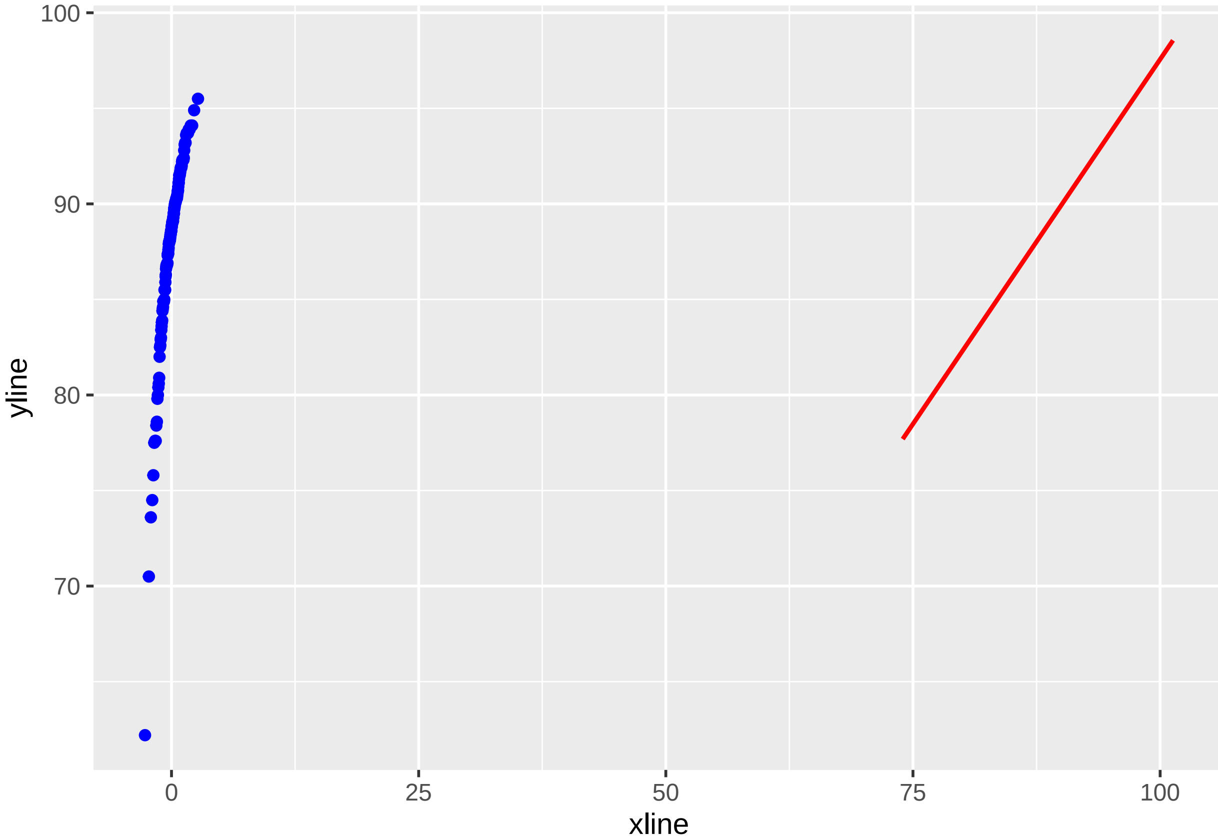
```
## The following objects are masked from 'package:ggplot2':
##
##     stat_qq_line, StatQqLine
```

```
ggplot(data = heights_df, aes(sample = HSDegree)) + stat_qq(colour="blue")
```




```
#Answer the following questions based on the Probability Plot:
#Based on what you see in this probability plot, is the distribution approximately normal? Explain how y
ggplot(data = heights_df, aes(sample = HSDegree)) + stat_qq(colour="blue") + stat_qq_line(colour="red")
```

```
## Warning: The following aesthetics were dropped during statistical transformation: sample
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```
# > The distribution is not normal. Plot is away from the normal line.

#If not normal, is the distribution skewed? If so, in which direction? Explain how you know.
# > Plot is way away from the normal line and since data is away from the X-axis, it is left
# > skewed.

# Now that you have looked at this data visually for normality, you will now quantify normality with nu
library(pastecs)
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
describe(heights_df)
```

```
##                        vars   n     mean       sd  median  trimmed
## Id*                       1 136    68.50    39.40    68.5    68.50
## Id2                       2 136 26833.13 15429.11 26112.0 26542.96
## Geography*                3 136    68.50    39.40    68.5    68.50
## PopGroupID                4 136     1.00     0.00     1.0     1.00
```

```
## POPGROUP.display.label*    5 136       1.00       0.00      1.0      1.00
## RacesReported              6 136 1144400.99 1090507.89 832707.5 927231.74
## HSDegree                   7 136      87.63       5.12     88.7     88.28
## BachDegree                 8 136      35.46       9.51     34.1     35.23
##                               mad      min       max     range  skew kurtosis
## Id*                         50.41      1.0     136.0     135.0  0.00    -1.23
## Id2                      20778.64   1073.0   55079.0   54006.0  0.05    -1.34
## Geography*                  50.41      1.0     136.0     135.0  0.00    -1.23
## PopGroupID                   0.00      1.0       1.0       0.0   NaN      NaN
## POPGROUP.display.label*      0.00      1.0       1.0       0.0   NaN      NaN
## RacesReported           314163.68 500292.0 10116705.0 9616413.0  4.98    33.50
## HSDegree                     3.78     62.2      95.5      33.3 -1.67     4.35
## BachDegree                   8.23     15.4      60.3      44.9  0.33    -0.28
##                               se
## Id*                         3.38
## Id2                      1323.04
## Geography*                  3.38
## PopGroupID                  0.00
## POPGROUP.display.label*     0.00
## RacesReported           93510.28
## HSDegree                    0.44
## BachDegree                  0.82
```

```
stat.desc(heights_df$HSDegree,basic = TRUE, norm = TRUE)
```

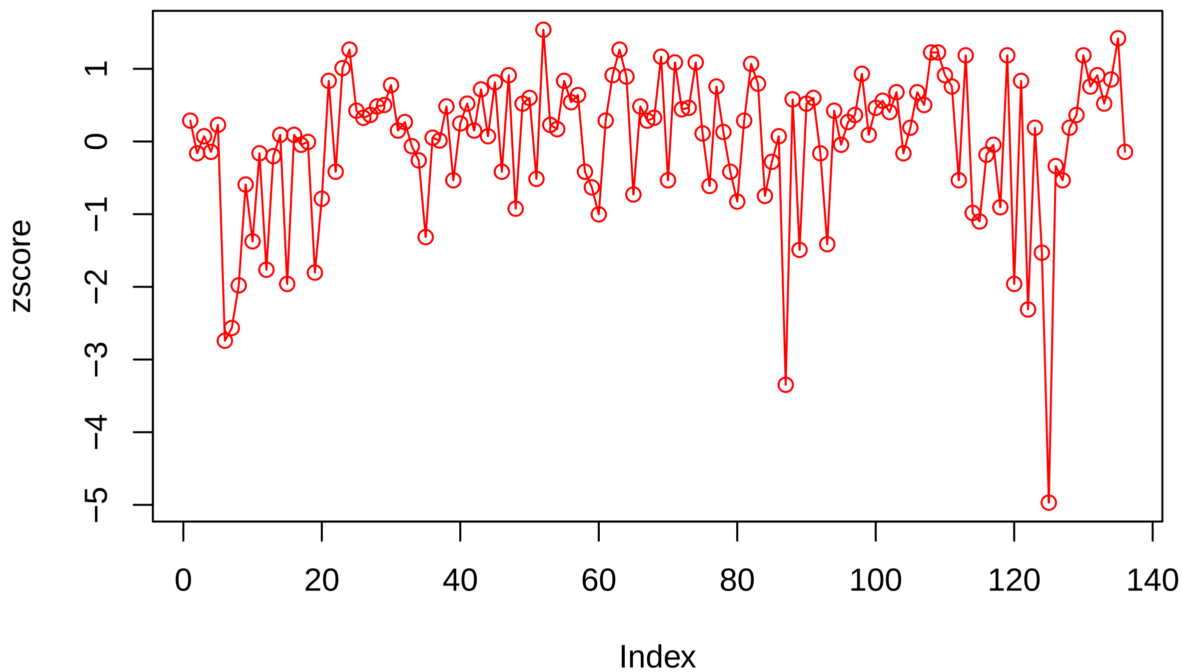
```
##       nbr.val      nbr.null        nbr.na           min           max
##  1.360000e+02  0.000000e+00  0.000000e+00  6.220000e+01  9.550000e+01
##         range           sum        median          mean       SE.mean
##  3.330000e+01  1.191800e+04  8.870000e+01  8.763235e+01  4.388598e-01
##  CI.mean.0.95           var       std.dev      coef.var      skewness
##  8.679296e-01  2.619332e+01  5.117941e+00  5.840241e-02 -1.674767e+00
##      skew.2SE      kurtosis      kurt.2SE    normtest.W    normtest.p
## -4.030254e+00  4.352856e+00  5.273885e+00  8.773635e-01  3.193634e-09
```

```
zscore <- (heights_df$HSDegree - mean(heights_df$HSDegree))/sd(heights_df$HSDegree)
zscore
```

```
##   [1]  0.286765161 -0.162634350  0.071834960 -0.143095241  0.228147834
##   [6] -2.741796762 -2.565944779 -1.979771504 -0.592494752 -1.374059119
##  [11] -0.162634350 -1.764841303 -0.201712568  0.091374069 -1.960232394
##  [16]  0.091374069 -0.045399695 -0.006321476 -1.803919521 -0.787885844
##  [21]  0.833860218 -0.416642769  1.009712201  1.263720620  0.423538925
##  [26]  0.325843380  0.364921598  0.482156253  0.501695362  0.775242891
##  [31]  0.149991397  0.267226052 -0.064938804 -0.260329896 -1.315441791
##  [36]  0.052295851  0.013217633  0.482156253 -0.533877424  0.247686943
##  [41]  0.521234471  0.149991397  0.716625563  0.071834960  0.814321109
##  [46] -0.416642769  0.912016655 -0.924659608  0.521234471  0.599390908
##  [51] -0.514338315  1.537268149  0.228147834  0.169530506  0.833860218
##  [56]  0.540773581  0.638469126 -0.416642769 -0.631572970 -1.002816045
##  [61]  0.286765161  0.912016655  1.263720620  0.892477546 -0.729268516
##  [66]  0.482156253  0.286765161  0.325843380  1.166025074 -0.533877424
##  [71]  1.087868638  0.443078035  0.462617144  1.087868638  0.110913179
##  [76] -0.612033861  0.755703781  0.130452288 -0.416642769 -0.826964062
```

```
##  [81]  0.286765161  1.068329528  0.794782000 -0.748807625 -0.279869005
##  [86]  0.071834960 -3.347509146  0.579851799 -1.491293774  0.521234471
##  [91]  0.599390908 -0.162634350 -1.413137337  0.423538925 -0.045399695
##  [96]  0.267226052  0.364921598  0.931555764  0.091374069  0.462617144
## [101]  0.560312690  0.403999816  0.677547345 -0.162634350  0.189069615
## [106]  0.677547345  0.501695362  1.224642402  1.224642402  0.912016655
## [111]  0.755703781 -0.533877424  1.185564183 -0.983276935 -1.100511591
## [116] -0.182173459 -0.045399695 -0.905120499  1.185564183 -1.960232394
## [121]  0.833860218 -2.311936360  0.189069615 -1.530371992 -4.969255208
## [126] -0.338486333 -0.533877424  0.189069615  0.364921598  1.185564183
## [131]  0.755703781  0.912016655  0.521234471  0.853399327  1.420033494
## [136] -0.143095241
```

```
plot(zscore, type="o", col="red")
```




```
# In several sentences provide an explanation of the result produced for skew, kurtosis, and z-scores.
# In addition, explain how a change in the sample size may change your explanation?

# > Skew and kurtosis are measure of asymmetry and irregularities in the data.
# > For a normal distribution, skew and kurtosis should be 0. In this case, the skew is
# > -1.674767 and kurtosis is 4.352856
# > A negative skew represents a left skew.
# > A positive kurtosis represents a pointy and heavy-tailed distribution.
# > Data in a left skew, positive kurtosis will be concentrated on the right side of the
# > distribution graph.
```

```
# > Mean = 87.63    Standard Deviation = 5.117941.
# > Z-score helps measure the standard deviation from the mean.
# > A positive z-score implies the individual value is greater than the mean, negative z-score # > impl
# > Larger the sample size accuracy of mean increases
# > (Z-score)2 x SD x (1-SD)/ME2 = Sample Size
# > Sample size is directly proportional to zscore. If the sample size decreases,
# > zscore decreases, which implies the confidence level of accuracy decreases.
# > Also the margin of error in a small sample is high.
```