

Kyle Salgado-Gouker

Prof. Denton

DSC 520 – Section T303

December 17, 2022

I have answered the questions first and listed the complete output at the end of this document.

i. 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)

- Id (alphanumeric character id, character 8-9 = "US", next two characters (numeric) are state)
- Id2 (numeric part of ID following "US")
- Geography (character data, name of population district, "county name, state")
- PopGroupID (an integer, always 1)
- POPGROUP.display.label (character data, "Total Population", PopGroupID description)
- RacesReported (misleading field name; it is actually the total population of the county, an integer)
- HSDegree (percentage with secondary school degree)
- BachDegree (percentage with bachelor degree)

ii. Run the following functions and provide the results: str(); nrow(); ncol()

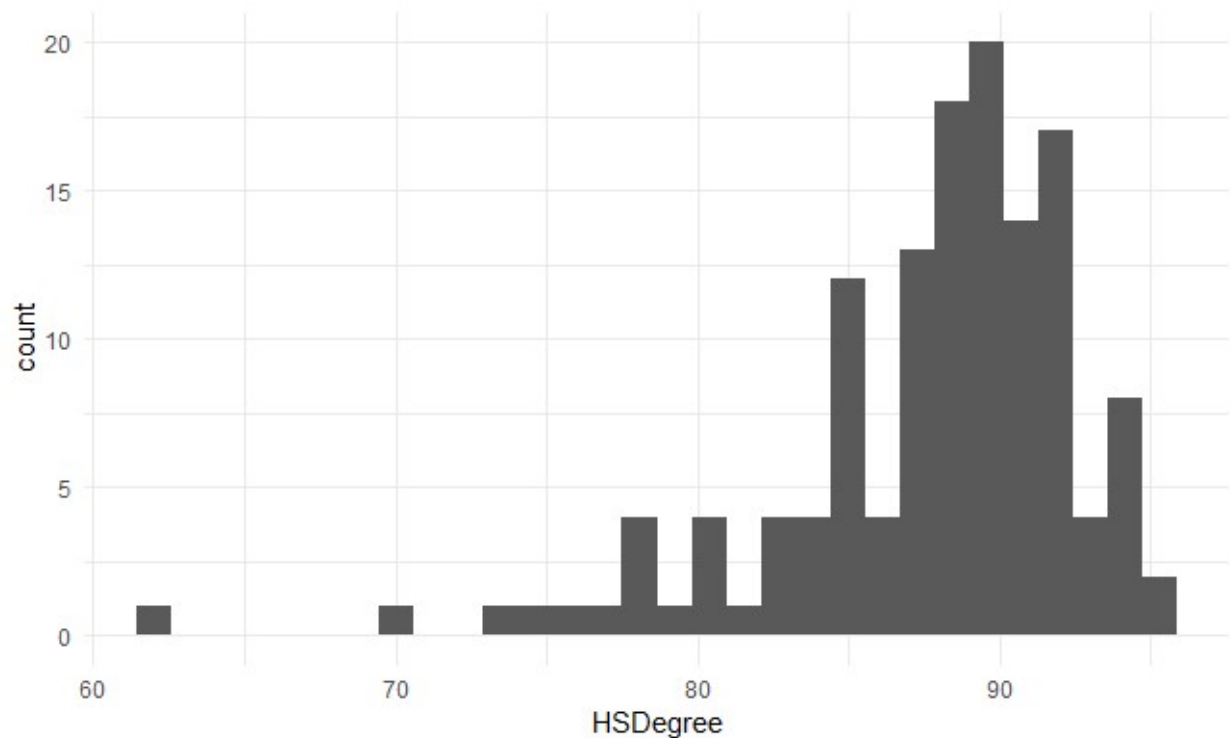
```

> str(census_df)
'data.frame': 136 obs. of 8 variables:
 $ Id      : chr  "05000000us01073" "05000000us04013" "05000000us04019" "05000000us06001" ...
 $ Id2     : int   1073 4013 4019 6001 6013 6019 6029 6037 6059 6065 ...
 $ Geography : chr   "Jefferson County, Alabama" "Maricopa County, Arizona" "Pima County, Arizona"
 "Alameda County, California" ...
 $ PopGroupID : int    1 1 1 1 1 1 1 1 1 1 ...
 $ POPGROUP.display.label: chr   "Total population" "Total population" "Total population" "Total population" ...
 $ RacesReported : int   660793 4087191 1004516 1610921 1111339 965974 874589 10116705 3145515 2329271
 ...
 $ 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 ...

> nrow(census_df)
[1] 136
> ncol(census_df)
[1] 8

```

iii. Create a Histogram of the HSDegree variable using the ggplot2 package.



1. Set a bin size for the Histogram that you think best visualizes the data (the bin size will determine how many bars display and how wide they are)

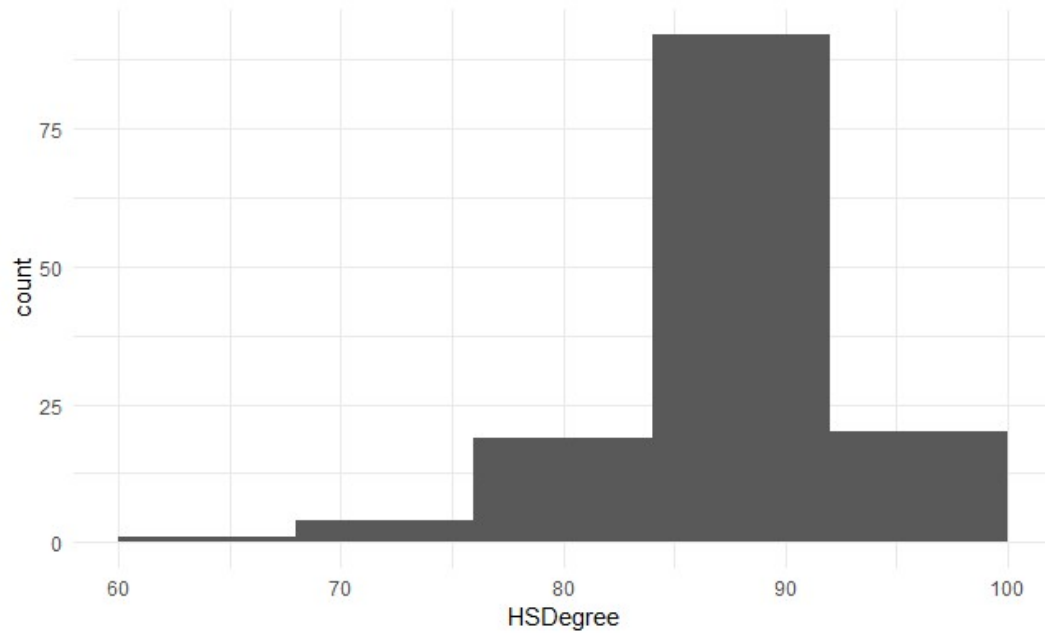
Used Sturge's Rule to determine adequate bin size:

$n = 136$

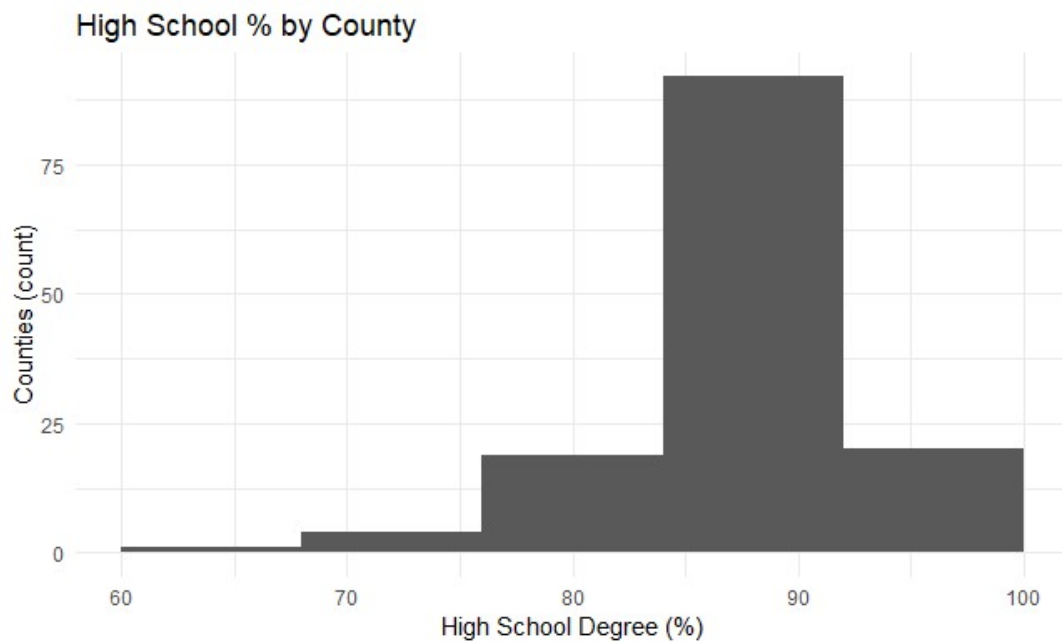
$\log_2 n = \log_2 136 = 7.0874628412503$

Sturge's Rule = $\log_2 136 + 1 = 7.0874628412503 + 1 = 8.0874628412503$

Rounded to 8.



2. Include a Title and appropriate X/Y axis labels on your Histogram Plot.



iv. Answer the following questions based on the Histogram produced:

1. Based on what you see in this histogram, is the data distribution unimodal?

Yes, it has a peak at about 90.

2. Is it approximately symmetrical?

Not really. It rises at a lower slope on the left and falls steeper on the right, thus producing a negative skew.

3. Is it approximately bell-shaped?

No, as there is much less data to the right of the peak than to the left.

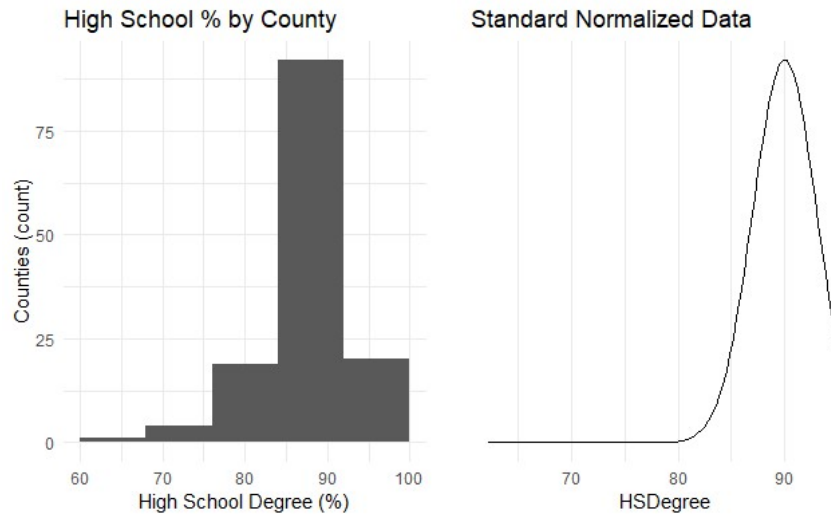
4. Is it approximately normal?

Yes. Even though the data is negatively skewed to the left, most of the data range conforms to normality.

5. If not normal, is the distribution skewed? If so, in which direction?

It is approximately normal, but the distribution is still skewed to the left (negatively).

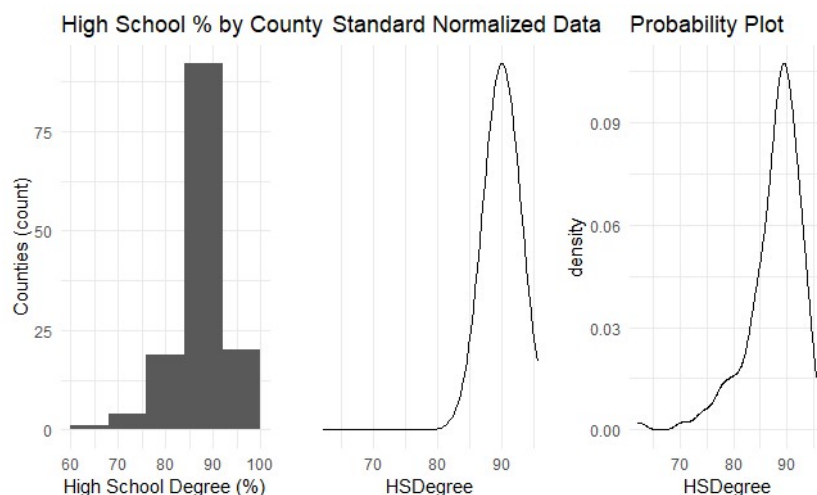
6. Include a normal curve to the Histogram that you plotted.



7. Explain whether a normal distribution can accurately be used as a model for this data.

Although across much of the range the data conforms to normality, I still think it is a dangerous tactic to use a normal distribution to approximate the data, because of outliers. The range from 83% to 92%, which encompasses most of the data, seems approximately normal; on the other hand, all the data outside this range is going to be wrong. Ultimately, I would not make this assumption.

v. Create a Probability Plot of the HSDegree variable.



vi. Answer the following questions based on the Probability Plot:

1. Based on what you see in this probability plot, is the distribution approximately normal?

Explain how you know.

It is normal over a significant range, but not outside the range. I extracted data from high school degree %, calculated the mean and standard deviation, and then found that far more data is outside of the standard deviation to the left (below 82.52%) instead of the right (92.74%).

2. If not normal, is the distribution skewed? If so, in which direction? Explain how you know.

The distribution is negatively skewed to the left. There is more data outside of the standard deviation to the left than the right.

vii. Now that you have looked at this data visually for normality, you will now quantify normality with numbers using the `stat.desc()` function. Include a screen capture of the results produced.

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

viii. 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?

```

z_scores <- (data-mean(data))/sd(data)
pos_z_scores <- abs(z_scores)
total_count <- length(pos_z_scores)
within_one_sd <- sum(pos_z_scores<=1)
within_two_sd <- sum(pos_z_scores<=2)
within_three_sd <- sum(pos_z_scores<=3)
within_four_sd <- sum(pos_z_scores<=4)
within_five_sd <- sum(pos_z_scores<=5)
factor_one_sd <- within_one_sd/total_count
factor_two_sd <- within_two_sd/total_count
factor_three_sd <- within_three_sd/total_count
factor_one_sd >= .68
# TRUE!
factor_two_sd >= .96
# TRUE!
factor_three_sd >= .997
# FALSE!

```

The data shows significant skewing to the left (-1.67) as well as a high degree of kurtosis (>4).

Moreover, the data does not satisfy the 68/96/99.7 rule for normal distribution. A larger number of samples might decrease the kurtosis and concentrate more data to the mean; however, the nature of this data is influenced by socio-economic conditions, and because of poor distribution of income, there will always be outliers. A larger number of samples (say 1000) also increases the bin width to 11, and that can affect how the data is interpreted.

Output Assignment 3.2:

```
> # Assignment: ASSIGNMENT 3-2
> # Name: Salgado-Gouker, Kyle
> # Date: 2022-12-17
>
> ## Load the ggplot2 package
> library(ggplot2)
> theme_set(theme_minimal())
>
> ## Set the working directory to the root of your DSC 520 directory
> setwd("C:\\Users\\kyles\\OneDrive\\Documents\\GitHub\\dsc520")
>
>
> # Id (alphanumeric character id, character 8-9 = "US", next two characters
(numeric) are state)
> # Id2 (numeric part of ID following "US")
> # Geography (character data, name of population district, "county name,
state")
> # PopGroupID (an integer, always 1)
> # POPGROUP.display.label (character data, "Total Population", PopGroupID
description)
> # RacesReported (misleading field name, It is actually the total population
of the county, an integer)
> # HSDegree (percentage with secondary school degree)
> # BachDegree (percentage with bachelor degree)
>
> ## Load the census
> census_df <- read.csv("data/acs-14-1yr-s0201.csv")
> str(census_df)
'data.frame':    136 obs. of  8 variables:
 $ Id              : chr  "05000000US01073" "05000000US04013"
"05000000US04019" "05000000US06001" ...
 $ Id2             : int   1073 4013 4019 6001 6013 6019 6029 6037 6059
6065 ...
 $ Geography       : chr   "Jefferson County, Alabama" "Maricopa County,
Arizona" "Pima County, Arizona" "Alameda County, California" ...
 $ PopGroupID      : int    1 1 1 1 1 1 1 1 1 ...
 $ POPGROUP.display.label: chr  "Total population" "Total population" "Total
population" "Total population" ...
 $ RacesReported   : int   660793 4087191 1004516 1610921 1111339 965974
874589 10116705 3145515 2329271 ...
 $ 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 ...
> nrow(census_df)
[1] 136
> ncol(census_df)
[1] 8
>
> # iii. Create a Histogram of the HSDegree variable using the ggplot2
> # package.
> ggplot(census_df, aes(HSDegree)) + geom_histogram()
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
>
```



```

> # 1. Set a bin size for the Histogram that you think best
> # visualizes the data (the bin size will determine how many
> # bars display and how wide they are)
> # Used Sturge's Rule to determine adequate bin size (result =
7.0874628412503 + 1)
> ggplot(census_df, aes(HSDegree)) + geom_histogram(binwidth = 8)
>
> # 2. Include a Title and appropriate X/Y axis labels on your
> # Histogram Plot.
> p1 <- ggplot(census_df, aes(HSDegree)) + geom_histogram(binwidth = 8) +
+   ggtitle("High School % by County") +
+   xlab("High School Degree (%)") + ylab("Counties (count)")
> p1
>
> # iv. Answer the following questions based on the Histogram
> # produced:
> #
> # 1. Based on what you see in this histogram, is the data
> # distribution unimodal? (YES, one peak at about 90)
> # 2. Is it approximately symmetrical? (Not really. It rises at a lower
slope on the left and falls steeper on the right, thus a negative skew)
> # 3. Is it approximately bell-shaped? (No. There is much less data to the
right of the peak than to the left)
> # 4. Is it approximately normal? (Yes. Even though the data is negatively
skewed to the left, most of the data range conforms to normality.)
> # 5. If not normal, is the distribution skewed? If so, in which
> # direction? (Skewed to the left)
> # A Standard Normal Distribution is a type of normal distribution with a
mean of 0 and a standard deviation of 1
> # 6. Include a normal curve to the Histogram that you
> # plotted.
> #install.packages("gridExtra")
> library(gridExtra)
> p2 <- ggplot(census_df, aes(HSDegree)) + stat_function(fun = dnorm, args =
list(mean = 90, sd = 3)) + ylab("") +
+   scale_y_continuous(breaks = NULL) + ggtitle(" Standard Normalized
Data")
> grid.arrange(p1, p2, ncol=2)
>
> # 7. Explain whether a normal distribution can accurately
> # be used as a model for this data.
> # I think it is a dangerous tactic to use a normal distribution to
approximate the data. The range from 83% to 92%, which
> # encompasses most of the data, seems approximately accurate. On the other
hand, all the data outside this range is going
> # to be wrong. From there, we just use Murphy's Law. Ultimately, I would
not use this method.
>
> # v. Create a Probability Plot of the HSDegree variable.
> #install.packages("qqplotr")
> library(qqplotr)
>
> # vi. Answer the following questions based on the Probability Plot:
> # 1. Based on what you see in this probability plot, is the
> # distribution approximately normal? Explain how you
> # know.

```

```

> # It is normal over a small range, but not when it is outside the range.
There is
> # a significant part of the range where it is not normal. It is negatively
skewed
> # to the left. I extracted data from high school degree %, calculated the
mean and
> # standard deviation, and then found that far more data is outside of the
standard
> # deviation to the left (below 82.52%) instead of the right (92.74%).
> p3 <- ggplot(census_df) + geom_density(aes(x=HSDegree)) +
ggtitle("Probability Plot")
> grid.arrange(p1, p2, p3, ncol=3)
>
> # 2. If not normal, is the distribution skewed? If so, in which
> # direction? Explain how you know.
> # Answered above.
> data<-census_df$HSDegree #extracts the data from population
column
> mean(data)
[1] 87.63235
> # [1] 87.63235
> sd(data)
[1] 5.117941
> # [1] 5.117941
>
> # vii. Now that you have looked at this data visually for normality,
> # you will now quantify normality with numbers using the
> # stat.desc() function. Include a screen capture of the results
> # produced.
> # install.packages("pastecs")
> library(pastecs)
> stat.desc(data, basic = TRUE, desc = TRUE, norm = TRUE)
      nbr.val      nbr.null      nbr.na      min      max
range      sum
1.360000e+02 0.000000e+00 0.000000e+00 6.220000e+01 9.550000e+01
3.330000e+01 1.191800e+04
      median      mean      SE.mean  CI.mean.0.95      var
std.dev      coef.var
8.870000e+01 8.763235e+01 4.388598e-01 8.679296e-01 2.619332e+01
5.117941e+00 5.840241e-02
      skewness      skew.2SE      kurtosis      kurt.2SE      normtest.W
normtest.p
-1.674767e+00 -4.030254e+00 4.352856e+00 5.273885e+00 8.773635e-01
3.193634e-09
>
> # nbr.val      nbr.null      nbr.na      min      max
range      sum      median
> # 1.360000e+02 0.000000e+00 0.000000e+00 6.220000e+01 9.550000e+01
3.330000e+01 1.191800e+04 8.870000e+01
> # mean      SE.mean  CI.mean.0.95      var      std.dev
coef.var      skewness      skew.2SE
> # 8.763235e+01 4.388598e-01 8.679296e-01 2.619332e+01 5.117941e+00
5.840241e-02 -1.674767e+00 -4.030254e+00
> # kurtosis      kurt.2SE      normtest.W      normtest.p
> # 4.352856e+00 5.273885e+00 8.773635e-01 3.193634e-09
>

```

```

> # a 3 kurtosis means Normal. Larger than 3 means more data is in the
head/tail than a normal distribution.
> # a normal graph has 0 skewness
> # this data, however, has a negative (left) skewness of -1.67, so (again)
not normal.
> # stat-desc() also provides skew.2SE and kurt.2SE.
> #
> # By converting skew and kurtosis to z-scores,
> # it is possible to determine how common (or uncommon) the level of skew
and kurtosis in our sample truly are.
> # The value of skew.2SE and kurt.2SE are equal to skew and kurtosis divided
by 2 standard errors.
> # By normalizing skew and kurtosis in this way, if skew.2SE and kurt.2SE
are greater than 1,
> # we can conclude that there is only a 5% chance (i.e.  $p < 0.05$ ) of
obtaining values of skew and kurtosis
> # as or more extreme than this by chance.
> #
> # Because these normalized values involve dividing by 2 standard errors,
they are sensitive
> # to the size of the sample. skew.2SE and kurt.2SE are most appropriate for
relatively small samples, 30-50.
> # For larger samples, it is best to compute values corresponding to 2.58SE
( $p < 0.01$ ) and 3.29SE ( $p < 0.001$ ).
> # In very large samples, say 200 observations or more, it is best to look
at the shape of the
> # distribution visually and consider the actual values of skew and
kurtosis, not their normalized values.
> #
>
> # for z-scores, use 68, 96, 99.7 rule, that is 68% of data within 1 sd, 96
within 2 sd, and 99.7 within 3.
> z_scores <- (data-mean(data))/sd(data)
> pos_z_scores <- abs(z_scores)
> total_count <- length(pos_z_scores)
> within_one_sd <- sum(pos_z_scores<=1)
> within_two_sd <- sum(pos_z_scores<=2)
> within_three_sd <- sum(pos_z_scores<=3)
> within_four_sd <- sum(pos_z_scores<=4)
> within_five_sd <- sum(pos_z_scores<=5)
> factor_one_sd <- within_one_sd/total_count
> factor_two_sd <- within_two_sd/total_count
> factor_three_sd <- within_three_sd/total_count
> factor_one_sd >= .68
[1] TRUE
> # TRUE!
> factor_two_sd >= .96
[1] TRUE
> # TRUE!
> factor_three_sd >= .997
[1] FALSE
> # FALSE!
>
> # viii. 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?
>

```

```
> # The data shows significant skewing to the left (1.67), a high degree of
kurtosis (>4). Moreover, the
> # data does not satisfy the 68/96/99.7 rule for normal distribution. A
larger number of samples might
> # decrease the kurtosis and concentrate more data to the mean; however the
nature of this data is influenced
> # by socio-economic markers, and because of poor distribution of income,
there will always be outliers. A
> # larger number of samples (say 1000) also increases the bin width to 11,
and that can affect how the data is
> # interpreted.
>
```