```
---
title: "ASSIGNMENT 2.1_AmericanCommunitySurvey"
author: "Bhushan Suryawanshi"
date: '2020-06-10'
output:
  html_document: default
---
```

This is your second exercise with real data. This time, instead of a bank of test scores, we will use the 2014 American Community Survey. These data are maintained by the US Census Bureau and are designed to show how communities are changing.

Through asking questions of a sample of the population, it produces national data on more than 35 categories of information, such as education, income, housing, and employment.

For this assignment, you will need to load and activate the ggplot2 package. (I urge you to do the DataCamp exercise first!). For this deliverable, you should provide the following:

1. What are the elements in your data (including the categories and data types)?

```
acs_df <- read.csv("acs-14-1yr-s0201.csv", stringsAsFactors = FALSE)
head(acs_df)
```

```
##               Id  Id2                      Geography PopGroupID
## 1 0500000US01073 1073       Jefferson County, Alabama          1
## 2 0500000US04013 4013        Maricopa County, Arizona          1
## 3 0500000US04019 4019            Pima County, Arizona          1
## 4 0500000US06001 6001      Alameda County, California          1
## 5 0500000US06013 6013 Contra Costa County, California          1
## 6 0500000US06019 6019       Fresno County, California          1
##   POPGROUP.display.label RacesReported HSDegree BachDegree
## 1       Total population        660793     89.1       30.5
## 2       Total population       4087191     86.8       30.2
## 3       Total population       1004516     88.0       30.8
## 4       Total population       1610921     86.9       42.8
## 5       Total population       1111339     88.8       39.7
## 6       Total population        965974     73.6       19.7
```

```
sapply(acs_df, class)
```

```
##                     Id                    Id2              Geography
##            "character"              "integer"            "character"
##             PopGroupID POPGROUP.display.label          RacesReported
##              "integer"            "character"              "integer"
##               HSDegree             BachDegree
##              "numeric"              "numeric"
```

Answer - Head shows the sample data in the data set. The function sapply helps to get datatype of the column. In this dataset we have 8 columns out of which 5 are columns are numerical data, And 3 are categorical data.

2. Please provide the output from the following functions: str(); nrow(); ncol()

```
      str(acs_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 ...
```

```
      nrow(acs_df)
```

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

```
      ncol(acs_df)
```

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

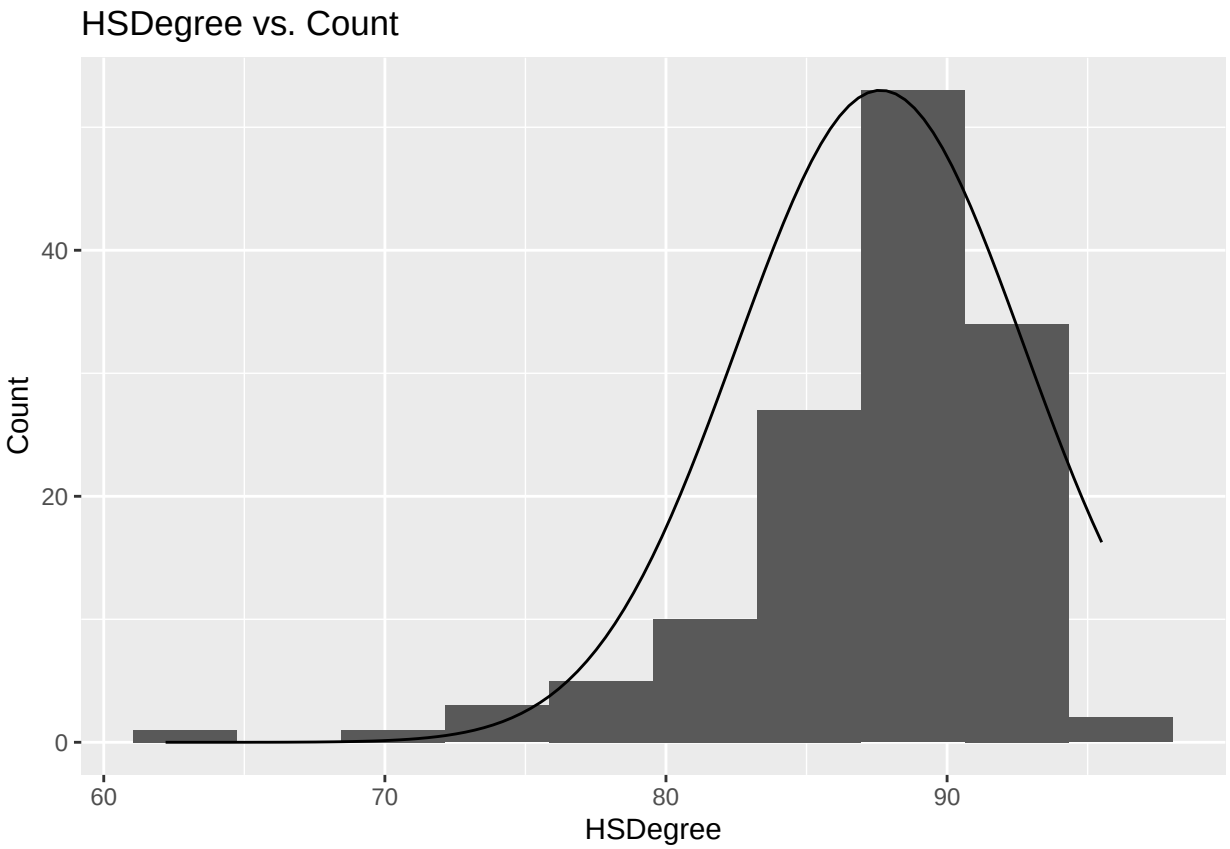
Answer - Above function str also helps to get column details of the dataset. The function nrow() and ncol() gives number of rows and number of columns in dataset respectively.

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

a. Set a bin size for the Histogram.

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

```
      library(ggplot2)
      bw = 5
      n_obs = sum(!is.na(acs_df$HSDegree))
      ggplot(acs_df, aes(HSDegree)) + geom_histogram(bins=10) + ggtitle('HSDegree vs. Count') + xlab('HS
```



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

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

b. Is it approximately symmetrical?

c. Is it approximately bell-shaped?

d. Is it approximately normal?

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

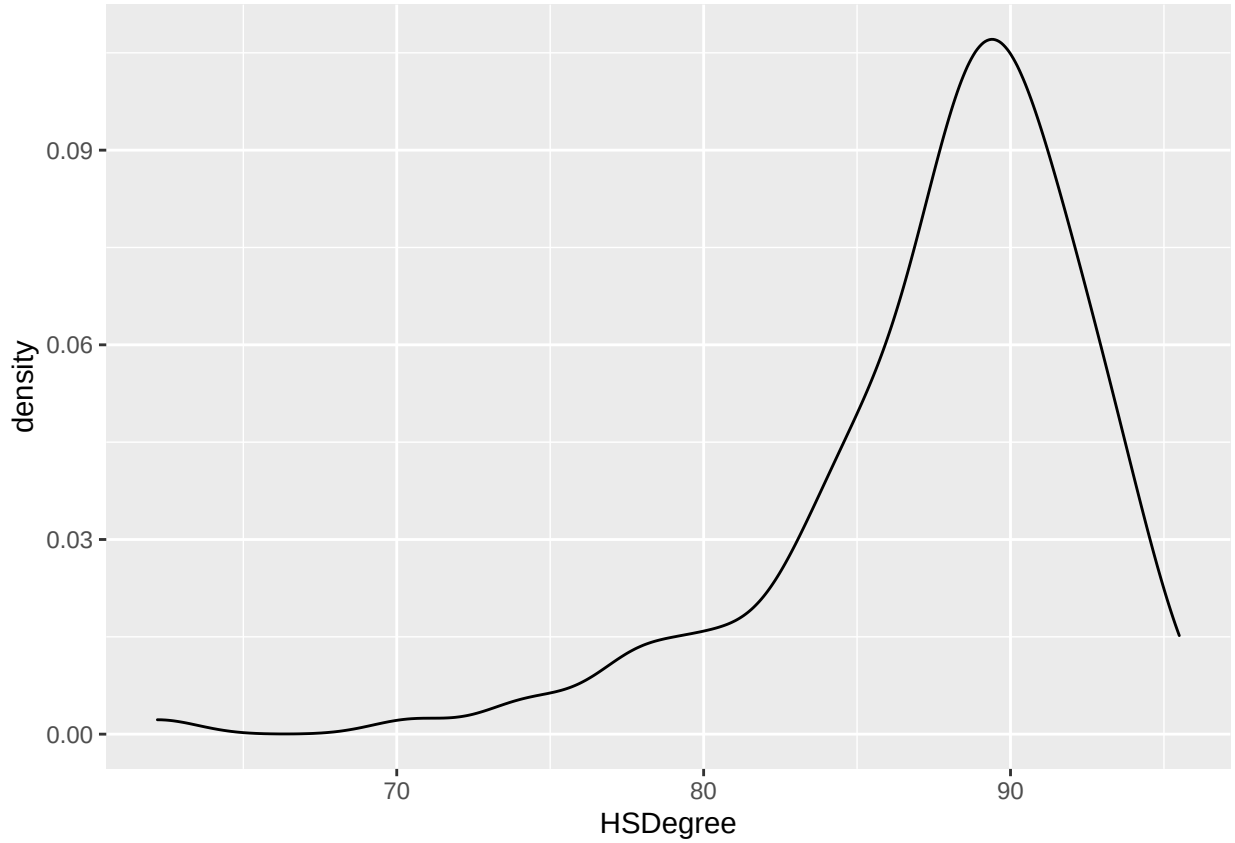
f. Include a normal curve to the Histogram that you plotted.

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

Answer - A unimodal distribution is a distribution with one clear peak or most frequent value. Based on the above histogram we can see that the data shown is unimodel but not normally distributed. It is negatively skewed distribution.

5. Create a Probability Plot of the HSDegree variable.

```
ggplot(acs_df, aes(HSDegree)) +  geom_density()
```



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

```
summary(acs_df$HSDegree)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   62.20   85.50   88.70   87.63   90.75   95.50
```

a. Based on what you see in this probability plot, is the distribution approximately normal? Explain how you know.

Answer - Based on probability distribution we can say the distribution is approximately normal because more than 60% of the data is near the mean of the distribution. Also we can observe data between 1st and 3rd quartile it is approximately more than 60%.

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

Answer - The data is not exactly normal but negatively skewed because mean is less than median. In normal distribution difference between mean and median is 0.

7. 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.

```
  library(pastecs)
  stat.desc(acs_df)
```

```
##          Id          Id2 Geography PopGroupID POPGROUP.display.label
## nbr.val  NA 1.360000e+02        NA        136                      NA
## nbr.null NA 0.000000e+00        NA          0                      NA
## nbr.na   NA 0.000000e+00        NA          0                      NA
## min      NA 1.073000e+03        NA          1                      NA
## max      NA 5.507900e+04        NA          1                      NA
## range    NA 5.400600e+04        NA          0                      NA
## sum      NA 3.649306e+06        NA        136                      NA
## median   NA 2.611200e+04        NA          1                      NA
## mean     NA 2.683313e+04        NA          1                      NA
## SE.mean  NA 1.323036e+03        NA          0                      NA
## CI.mean  NA 2.616557e+03        NA          0                      NA
## var      NA 2.380576e+08        NA          0                      NA
## std.dev  NA 1.542911e+04        NA          0                      NA
## coef.var NA 5.750024e-01        NA          0                      NA
##          RacesReported     HSDegree   BachDegree
## nbr.val   1.360000e+02 1.360000e+02  136.0000000
## nbr.null  0.000000e+00 0.000000e+00    0.0000000
## nbr.na    0.000000e+00 0.000000e+00    0.0000000
## min       5.002920e+05 6.220000e+01   15.4000000
## max       1.011671e+07 9.550000e+01   60.3000000
## range     9.616413e+06 3.330000e+01   44.9000000
## sum       1.556385e+08 1.191800e+04 4822.7000000
## median    8.327075e+05 8.870000e+01   34.1000000
## mean      1.144401e+06 8.763235e+01   35.4610294
## SE.mean   9.351028e+04 4.388598e-01    0.8154527
## CI.mean   1.849346e+05 8.679296e-01    1.6127146
## var       1.189207e+12 2.619332e+01   90.4349886
## std.dev   1.090508e+06 5.117941e+00    9.5097313
## coef.var  9.529072e-01 5.840241e-02    0.2681741
```

```
  library(e1071)
  skewness(acs_df$HSDegree)
```

```
## [1] -1.674767
```

```
  kurtosis(acs_df$HSDegree)
```

```
## [1] 4.352856
```

```
  library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
  a <- sqldf("SELECT DISTINCT id FROM acs_df  ORDER BY RANDOM(*) LIMIT 5")
  small_acs_df = sqldf("SELECT * FROM acs_df WHERE id IN a")
  small_acs_df
```

```
##               Id   Id2                          Geography PopGroupID
## 1 0500000US06065  6065      Riverside County, California          1
## 2 0500000US25005 25005     Bristol County, Massachusetts          1
## 3 0500000US36059 36059           Nassau County, New York          1
## 4 0500000US41067 41067         Washington County, Oregon          1
## 5 0500000US42101 42101 Philadelphia County, Pennsylvania          1
##   POPGROUP.display.label RacesReported HSDegree BachDegree
## 1       Total population       2329271     80.6       20.7
## 2       Total population        554194     82.5       25.7
## 3       Total population       1358627     90.7       43.2
## 4       Total population        562998     90.2       39.7
## 5       Total population       1560297     82.6       26.0
```

```
  skewness(small_acs_df$HSDegree)
```

```
## [1] 0.2313056
```

```
  kurtosis(small_acs_df$HSDegree)
```

```
## [1] -2.209122
```

8. 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?

Answer - Skewness talks about lack of symmetry in the data. Where as kurtosis talks about pointiness at the end, The sample size or size of data does affect the distribution. As shown in the above example we picked random 5 data points from the give sample. Then we calculated skewness and kurtosis for current sample size of 136 observations. We see that if we use all data then the skewness is -1.67 where as for smaller sample it shows -0.79. Same is the case with kurtosis, with all data we get kurtosis 4.35 and with small sample it is -1.22, Which means sample size does affect the analysis. Getting right sample size is key to data analysis. In our example if we increase sample size we may get normal curve and we have higher probability of having more population with HSDegree.