Test Scores Comparison using R programming

Kesav Adithya Venkidusamy

Bellevue university - Master of Science in Data Science

Course Name: DSC520-T301 Statistics for Data Science (2221-1)

Assignment: Week 4.2 Assignment

Instructor: Dr Richard Bushart

Due Date: 09/26/2021

Assignment 4.2.1

##Import dataset Scores.csv

## Set the working directory to the root of your DSC 520 directory

```
setwd("E:/Personal/Bellevue University/Course/github/dsc520")
```

## Load the `data/scores.csv` to df

```
scores_df <- read.csv("data/scores.csv")
```

1. **Use the appropriate R functions to answer the following questions**

    a. **What are the observational units in this study?**

    The observational units in this study are to compare the performance of students using course grades and total points earned in the course

    b. **Identify the variables mentioned in the narrative paragraph and determine which are categorical and quantitative?**

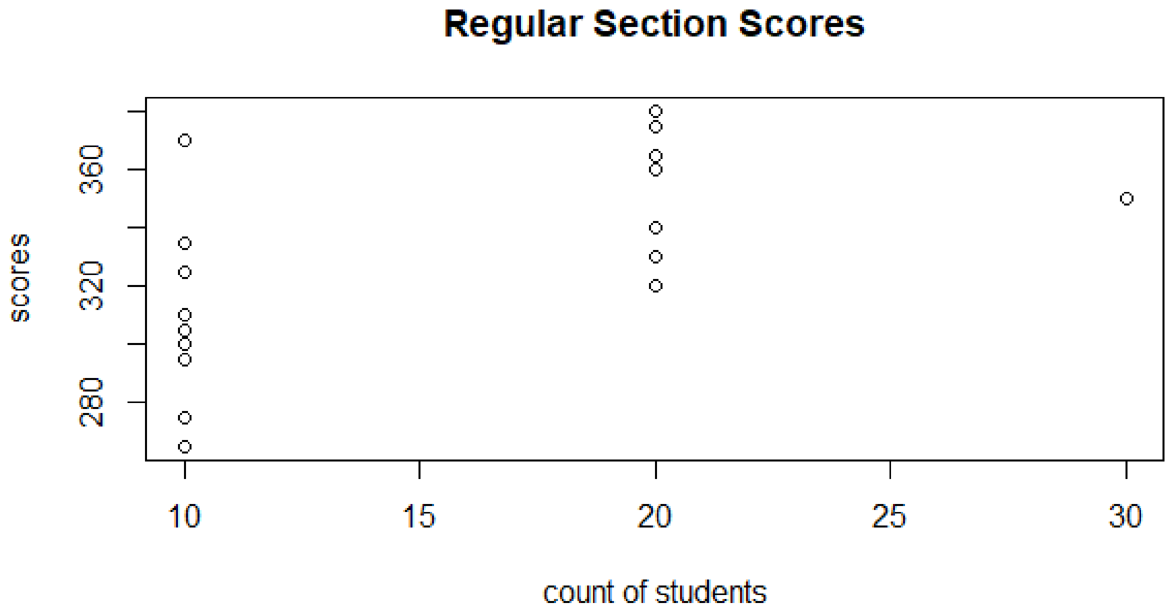
    The variables mentioned in the problem statement are course grades and total points earned in the course. Here, total points earned by students are quantitative and course grade is categorical.

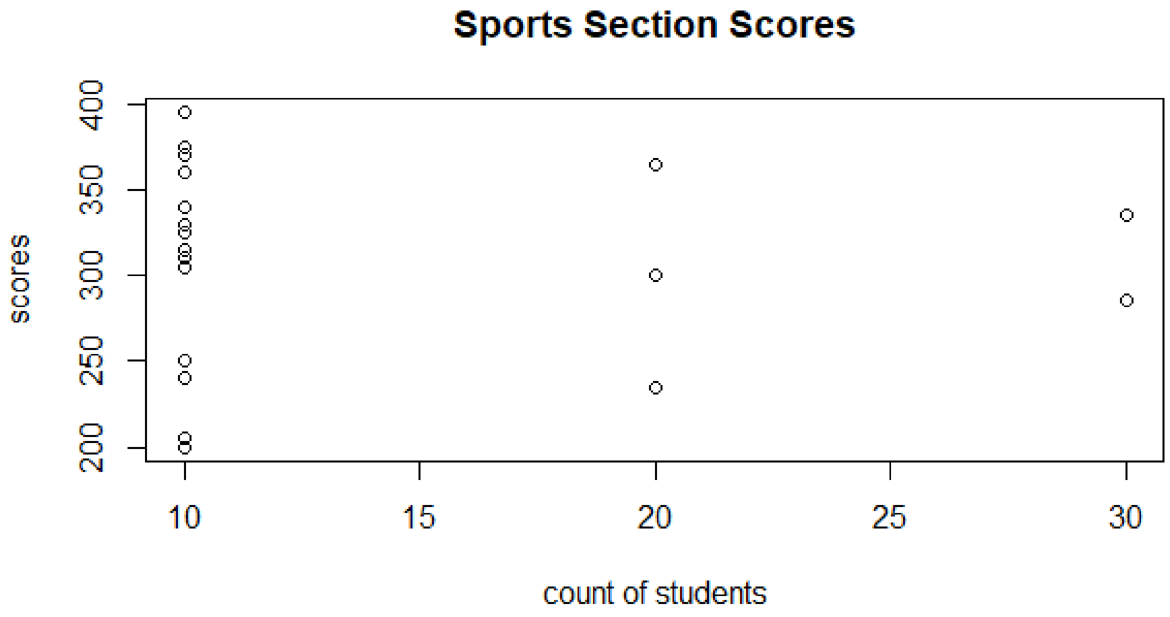
    c. **Create one variable to hold a subset of your data set that contains only the Regular Section and one variable for the Sports Section**

    ```
    regular_scores_df <- filter(scores_df, Section == 'Regular')
    
    sports_scores_df <- filter(scores_df, Section == 'Sports')
    ```

    d. **Use the Plot function to plot each Sections scores and the number of students achieving that score. Use additional Plot Arguments to label the graph and give each axis an appropriate label.**

```
plot(regular_scores_df$Score,regular_scores_df$Count,xlab = "scores" ,ylab =
"count of students", main = "Regular Section Scores")
plot(sports_scores_df$Score,sports_scores_df$Count,xlab = "scores" ,ylab =
"count of students", main = "Sports Section Scores")
```



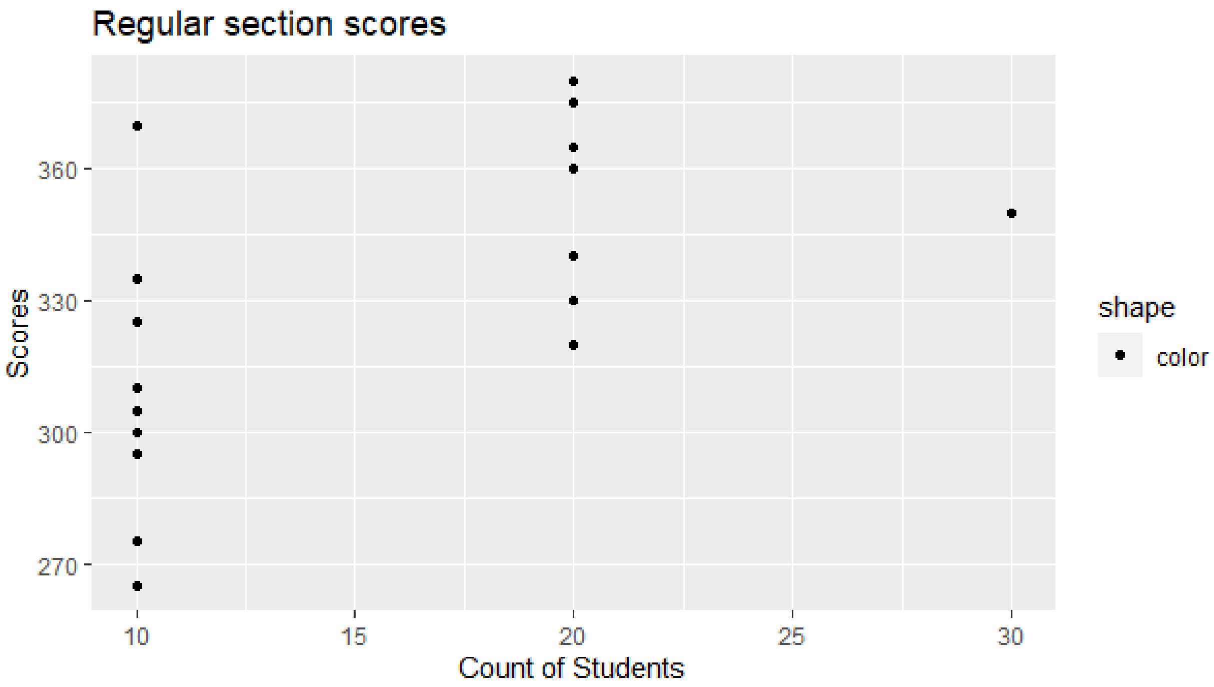


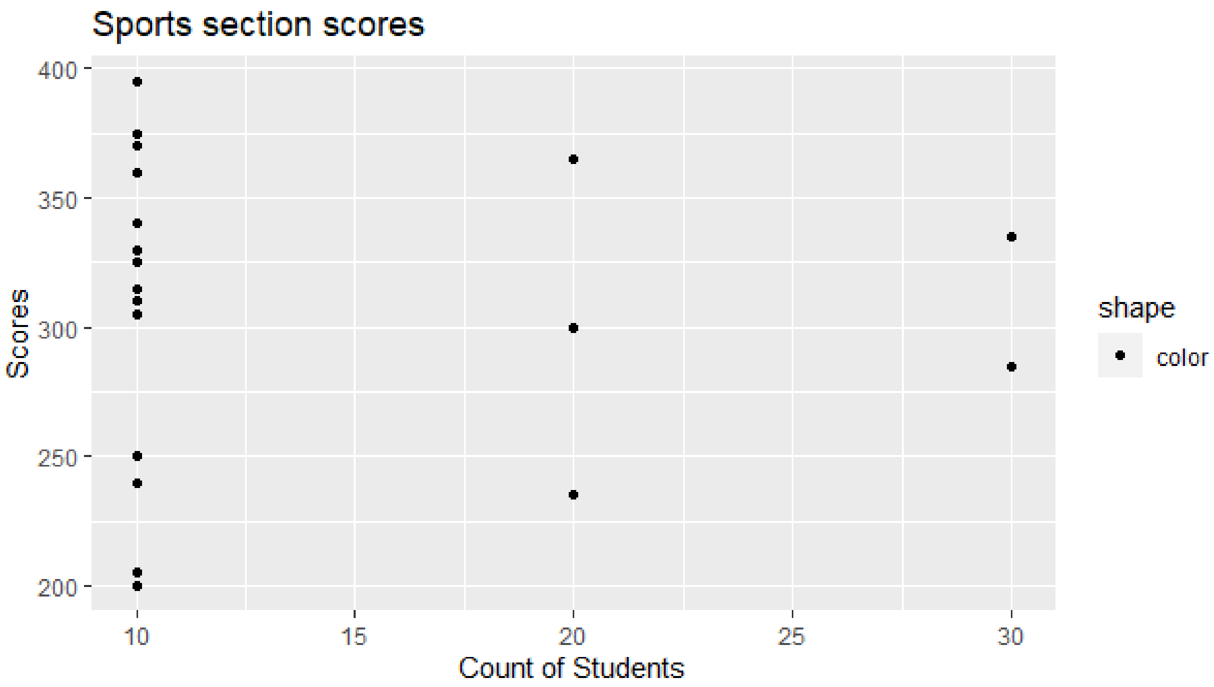
Using ggplot2

```
library(ggplot2)

ggplot(regular_scores_df,aes(x = Count, y = Score, shape = 'color')) + geom_point() +
ggtitle("Regular section scores") + xlab("Count of Students") + ylab("Scores")

ggplot(sports_scores_df,aes(x = Count, y = Score, shape = 'color')) + geom_point() +
ggtitle("Sports section scores") + xlab("Count of Students") + ylab("Scores")
```

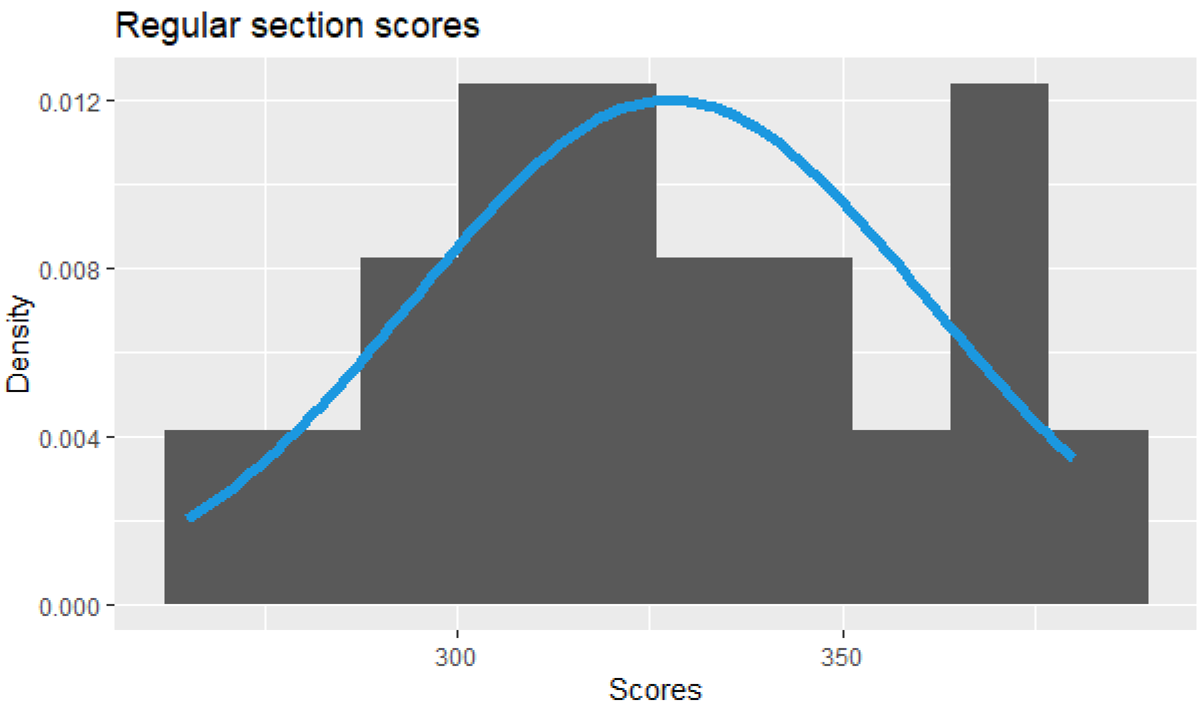


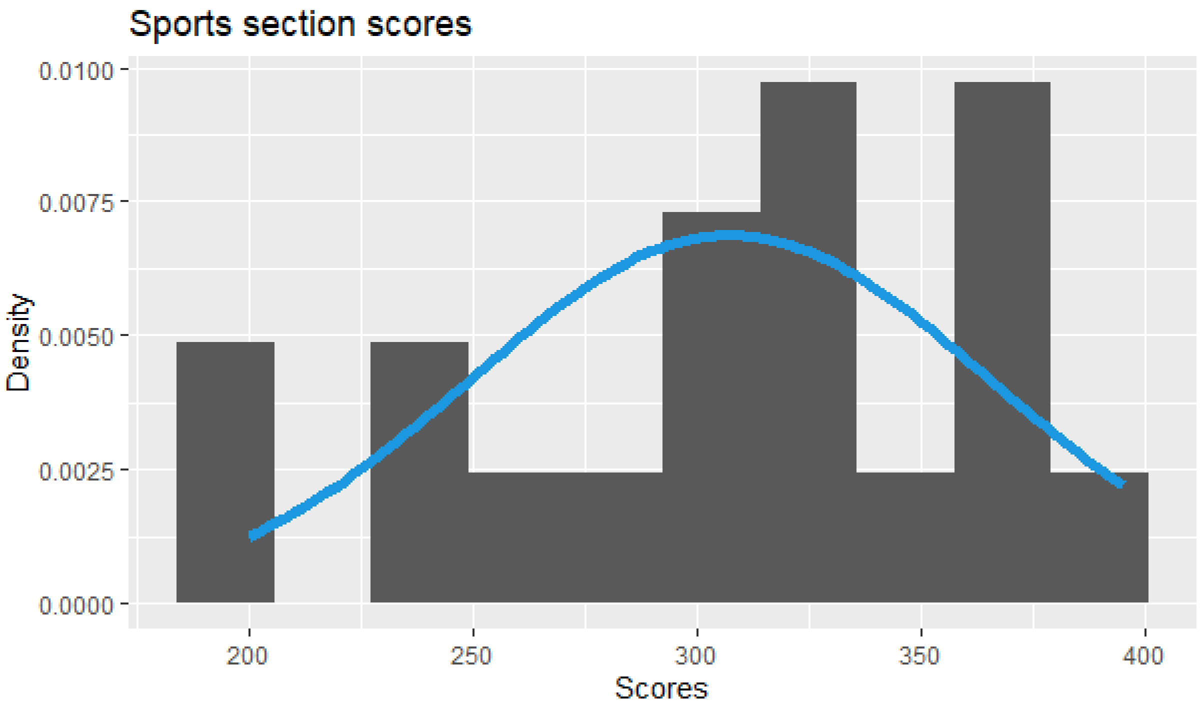


```
##Histogram

ggplot(regular_scores_df,aes(Score)) + geom_histogram(bins=10)

ggplot(sports_scores_df,aes(Score)) + geom_histogram(bins=10)
```





```
##Sharpiro.test

shapiro.test(regular_scores_df$Score)

shapiro.test(sports_scores_df$Score)
```

```
##Sharpiro.test

> shapiro.test(regular_scores_df$Score)


        Shapiro-Wilk normality test


data:  regular_scores_df$Score

W = 0.96952, p-value = 0.7668


> shapiro.test(sports_scores_df$Score)


        Shapiro-Wilk normality test


data:  sports_scores_df$Score

W = 0.94456, p-value = 0.318
```

Comparing and contrasting the point distributions between the two sections, looking at both tendency and consistency: Can you say that one section tended to score more points than the other? Justify and explain your answer.

The distribution of both regular and sports sections are fairly normal. From the above 2 histograms, we could see the regular section is centered around 325, however, the sports section is centered around 300. However, Shapiro test indicated that regular section marks are normally distributed (p > 0.5) but, sports section scores are not normally distributed (p < 0.5). Hence, Regular section tended to score more points than Sports section.

Did every student in one section score more points than every student in the other section? If not, explain what a statistical tendency means in this context.

```
##stat.desc

library(pastecs)

stat.desc(regular_scores_df$Score)
stat.desc(sports_scores_df$Score)


stat.desc(regular_scores_df$Score)
     nbr.val      nbr.null        nbr.na           min           max         range 
  19.0000000     0.0000000     0.0000000   265.0000000   380.0000000   115.0000000 
         sum        median          mean       SE.mean  CI.mean.0.95           var 
6225.0000000   325.0000000   327.6315789     7.6315789    16.0333524  1106.5789474 
     std.dev      coef.var 
  33.2652814     0.1015326 
> stat.desc(sports_scores_df$Score)
     nbr.val      nbr.null        nbr.na           min           max         range 
  19.0000000     0.0000000     0.0000000   200.0000000   395.0000000   195.0000000 
         sum        median          mean       SE.mean  CI.mean.0.95           var 
5840.0000000   315.0000000   307.3684211    13.3134085    27.9704333  3367.6900585 
     std.dev      coef.var 
  58.0318021     0.1888021 
```

As stated above, the mean for Regular section is higher than the mean for Sports section. The variance is also less for Regular section compared to Sports section. Regular section significantly performed better compared to Sports section. However, not all the students in Regular section scored better than every student in Sports section. The central tendency implied that Regular section is more likely to score better than Sports section.

What could be one additional variable that was not mentioned in the narrative that could be influencing the point distributions between the two sections?

One additional variable that is not mentioned in the narrative could be the class size. It could be possible the Sports section has less number of students enrolled compared to Regular section as this is concentrated towards only one subject.

Assignment 4.2.2

##Housing dataset

```
library("readxl")
```

## Set the working directory to the root of your DSC 520 directory

```
setwd("E:/Personal/Bellevue University/Course/github/dsc520")
```

## Load the `data/scores.csv` to df

```
housing_data <- read_excel("data/week-7-housing.xlsx")
attributes(housing_data)
```

```
attributes(housing_data)
$class
[1] "tbl_df"     "tbl"        "data.frame"
$names
 [1] "Sale Date"                "Sale Price"
 [3] "sale_reason"              "sale_instrument"
 [5] "sale_warning"             "sitetype"
 [7] "addr_full"                "zip5"
 [9] "ctyname"                  "postalctyn"
[11] "lon"                      "lat"
[13] "building_grade"           "square_feet_total_living"
[15] "bedrooms"                 "bath_full_count"
[17] "bath_half_count"          "bath_3qtr_count"
```

```
[19] "year_built"              "year_renovated"

[21] "current_zoning"          "sq_ft_lot"

[23] "prop_type"               "present_use"

str(housing_data)

> str(housing_data)

tibble [12,865 x 24] (S3: tbl_df/tbl/data.frame)

 $ Sale Date               : POSIXct[1:12865], format: "2006-01-03" "2006-01-03" ...

 $ Sale Price              : num [1:12865] 698000 649990 572500 420000 369900 ...

 $ sale_reason             : num [1:12865] 1 1 1 1 1 1 1 1 1 1 ...

 $ sale_instrument         : num [1:12865] 3 3 3 3 3 15 3 3 3 3 ...

 $ sale_warning            : chr [1:12865] NA NA NA NA ...

 $ sitetype                : chr [1:12865] "R1" "R1" "R1" "R1" ...

 $ addr_full               : chr [1:12865] "17021 NE 113TH CT" "11927 178TH PL NE" "13315
174TH AVE NE" "3303 178TH AVE NE" ...

 $ zip5                    : num [1:12865] 98052 98052 98052 98052 98052 ...

 $ ctyname                 : chr [1:12865] "REDMOND" "REDMOND" NA "REDMOND" ...

 $ postalctyn              : chr [1:12865] "REDMOND" "REDMOND" "REDMOND"
"REDMOND" ...

 $ lon                     : num [1:12865] -122 -122 -122 -122 -122 ...

 $ lat                     : num [1:12865] 47.7 47.7 47.7 47.6 47.7 ...

 $ building_grade          : num [1:12865] 9 9 8 8 7 7 10 10 9 8 ...

 $ square_feet_total_living: num [1:12865] 2810 2880 2770 1620 1440 4160 3960 3720 4160
2760 ...

 $ bedrooms                : num [1:12865] 4 4 4 3 3 4 5 4 4 4 ...

 $ bath_full_count         : num [1:12865] 2 2 1 1 1 2 3 2 2 1 ...

 $ bath_half_count         : num [1:12865] 1 0 1 0 0 1 0 1 1 0 ...

 $ bath_3qtr_count         : num [1:12865] 0 1 1 1 1 1 1 0 1 1 ...

 $ year_built              : num [1:12865] 2003 2006 1987 1968 1980 ...
 $ year_renovated         : num [1:12865] 0 0 0 0 0 0 0 0 0 0 ...
 $ current_zoning         : chr [1:12865] "R4" "R4" "R6" "R4" ...
 $ sq_ft_lot              : num [1:12865] 6635 5570 8444 9600 7526 ...
 $ prop_type              : chr [1:12865] "R" "R" "R" "R" ...
 $ present_use            : num [1:12865] 2 2 2 2 2 2 2 2 2 2 ...
```

##Use the apply function on a variable in your dataset

```
sales_price <- housing_data["Sale Price"]
mean_sales <- apply(sales_price,2, mean)
print(mean_sales)
> ##Use the apply function on a variable in your dataset
> sales_price <- housing_data["Sale Price"]
> mean_sales <- apply(sales_price,2, mean)
> print(mean_sales)
Sale Price
  660737.7
```

##Use the aggregate function on a variable in your dataset

##Mean of Sale Price by bedrooms

```
aggregate(`Sale Price` ~ bedrooms, housing_data, mean)


> aggregate(`Sale Price` ~ bedrooms, housing_data, mean)
   bedrooms Sale Price
1         0   844059.5
2         1   722814.1
3         2   544946.4
4         3   564958.6
```

```
5         4   735910.0
6         5   836974.0
7         6   767494.3
8         7  1307281.7
9         8  1122500.0
10         9   581500.0
11       10   450000.0
12       11  1825000.0
```

##Use the plyr function on a variable in your dataset – more specifically, I want to see you split some data, perform a modification to the data, and then bring it back together

I calculate price_per_sq_ft based on square_feet_total_living and Sale Price

```
library(dplyr)

housing_subset_df <- select(housing_data, square_feet_total_living, `Sale Price`)

head(housing_subset_df)


housing_subset_df <- rename(housing_subset_df, replace = c("Sale Price" = "sales_price"))

head(housing_subset_df)


housing_subset_df <- mutate(housing_subset_df, price_per_sq_ft =
format(round(sales_price/square_feet_total_living,2)))

head(housing_subset_df)


price_per_sq_feet <- select(housing_subset_df,price_per_sq_ft)

head(price_per_sq_feet)


housing_df_new <- bind_cols(housing_data, price_per_sq_feet)

head(housing_df_new)
```

```
> housing_subset_df <- select(housing_data, square_feet_total_living, `Sale Price`)

> head(housing_subset_df)

# A tibble: 6 x 2

  square_feet_total_living `Sale Price`

                     <dbl>        <dbl>

1                     2810       698000

2                     2880       649990

3                     2770       572500

4                     1620       420000

5                     1440       369900

6                     4160       184667

> housing_subset_df <- rename(housing_subset_df, replace = c("Sale Price" = "sales_price"))

> head(housing_subset_df)

# A tibble: 6 x 2

  square_feet_total_living sales_price

                     <dbl>       <dbl>

1                     2810      698000

2                     2880      649990

3                     2770      572500

4                     1620      420000

5                     1440      369900

6                     4160      184667

> housing_subset_df <- mutate(housing_subset_df, price_per_sq_ft =
format(round(sales_price/square_feet_total_living,2)))

> head(housing_subset_df)

# A tibble: 6 x 3
```

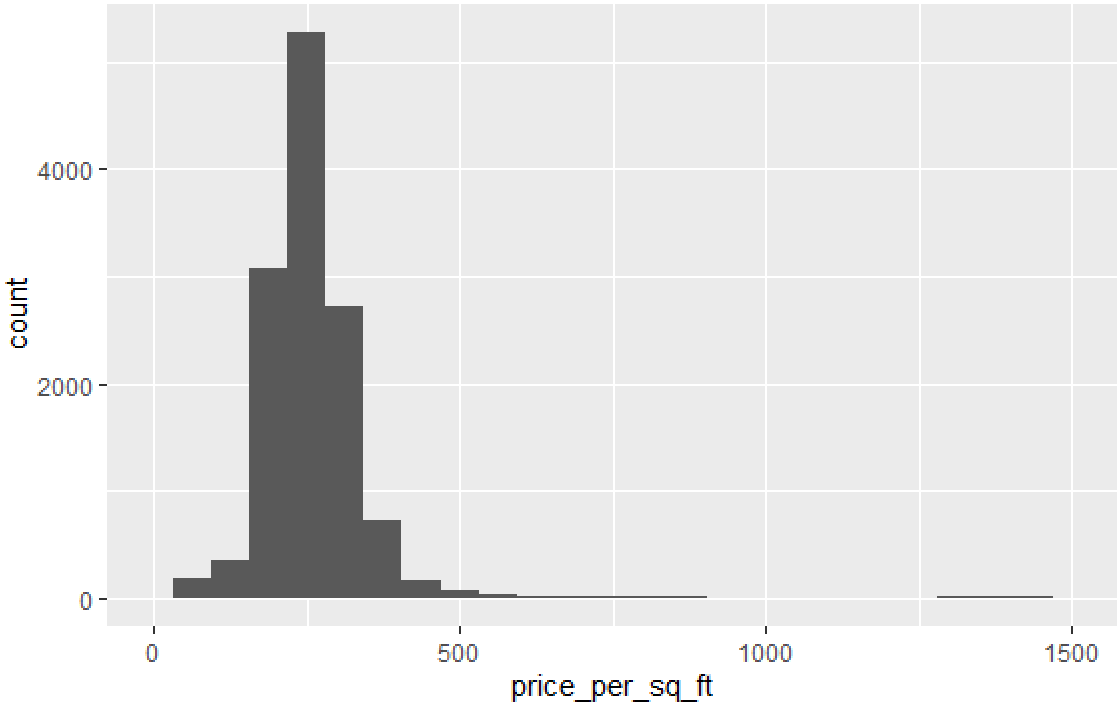
```
  square_feet_total_living sales_price price_per_sq_ft
                     <dbl>       <dbl> <chr>
1                     2810      698000 " 248.40"
2                     2880      649990 " 225.69"
3                     2770      572500 " 206.68"
4                     1620      420000 " 259.26"
5                     1440      369900 " 256.88"
6                     4160      184667 "  44.39"
> price_per_sq_feet <- select(housing_subset_df,price_per_sq_ft)
> head(price_per_sq_feet)
# A tibble: 6 x 1
  price_per_sq_ft
  <chr>
1 " 248.40"
2 " 225.69"
3 " 206.68"
4 " 259.26"
5 " 256.88"
6 "  44.39"
> housing_df_new <- bind_cols(housing_data, price_per_sq_feet)
> head(housing_df_new)
# A tibble: 6 x 25
  `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warning sitetype
  <dttm>                     <dbl>       <dbl>           <dbl> <chr>        <chr>
1 2006-01-03 00:00:00       698000           1               3 NA           R1
2 2006-01-03 00:00:00       649990           1               3 NA           R1
3 2006-01-03 00:00:00       572500           1               3 NA           R1
```

```
4 2006-01-03 00:00:00     420000        1             3 NA          R1
5 2006-01-03 00:00:00     369900        1             3 15          R1
6 2006-01-03 00:00:00     184667        1            15 18 51       R1
# ... with 19 more variables: addr_full <chr>, zip5 <dbl>, ctyname <chr>,
#   postalctyn <chr>, lon <dbl>, lat <dbl>, building_grade <dbl>,
#   square_feet_total_living <dbl>, bedrooms <dbl>, bath_full_count <dbl>,
#   bath_half_count <dbl>, bath_3qtr_count <dbl>, year_built <dbl>,
#   year_renovated <dbl>, current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>,
#   present_use <dbl>, price_per_sq_ft <chr>
```

Check distributions of the data

I have used price_per_sq_ft to check the distribution of data

```
##Check distributions of the data
ggplot(housing_df_new, aes(`price_per_sq_ft`)) + geom_histogram(bins=25) + xlim(0, 1500)
```



From the graph, we could see the data is positively skewed.

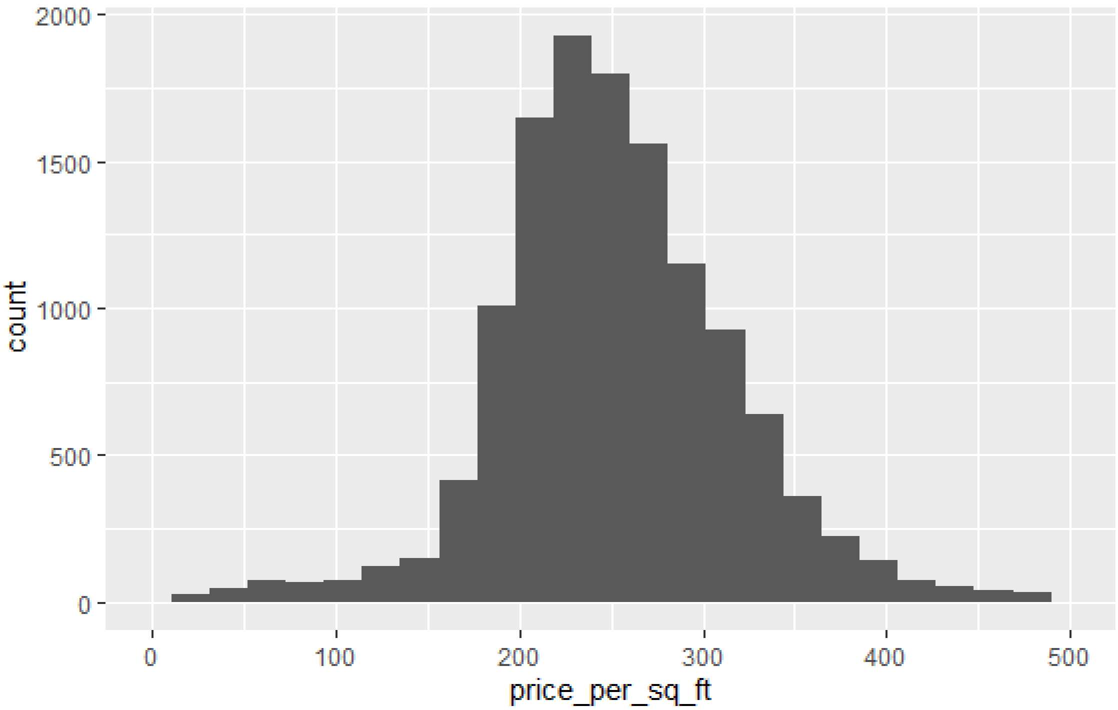
##Identify if there are any outliers

##Limiting the price_per_sq_ft to less than $500

```
housing_df_new <- filter(housing_df_new, price_per_sq_feet <= 500)
head(housing_df_new)
```

##Distribution after removing outliers

```
ggplot(housing_df_new, aes(`price_per_sq_ft`)) + geom_histogram(bins=25) + xlim(0, 500)
```



Create at least 2 new variables

##Create at least 2 new variables

```
housing_data_new <- mutate(housing_data,
price_per_sq_ft=sales_price/square_feet_total_living, house_price =
case_when((price_per_sq_ft <= 100) ~ 'Low',(price_per_sq_ft > 100 & price_per_sq_ft <= 250)
~ 'Medium',(price_per_sq_ft > 250) ~ 'High'))
```

attributes(housing_data_new)

```
>attributes(housing_data_new)

$names

 [1] "Sale Date"                "Sale Price"               "sale_reason"

 [4] "sale_instrument"          "sale_warning"             "sitetype"

 [7] "addr_full"                "zip5"                     "ctyname"

[10] "postalctyn"               "lon"                      "lat"

[13] "building_grade"           "square_feet_total_living" "bedrooms"

[16] "bath_full_count"          "bath_half_count"          "bath_3qtr_count"

[19] "year_built"               "year_renovated"           "current_zoning"

[22] "sq_ft_lot"                "prop_type"                "present_use"

[25] "price_per_sq_ft"          "house_price"
```

```
housing_data_new %>% select(price_per_sq_ft, house_price)
```

```
# A tibble: 12,865 x 2

   price_per_sq_ft$`Sale Price` house_price

                          <dbl> <chr>

 1                         248.  Medium

 2                         226.  Medium

 3                         207.  Medium

 4                         259.  High

 5                         257.  High

 6                          44.4 Low
 7                      265.  High
 8                      235.  Medium
 9                      159.  Medium
10                       236.  Medium
# ... with 12,855 more rows
```