Exercise 1

General Guidelines:

To start a new source file, go to `File` → `New File` → `R Script` or click the icon with a green plus sign at the top left corner. You can write your codes in the source file first and then run them in the console using Ctrl + Enter (Windows) or Command + Return (Mac)

In this homework, you will work with 3 data sets from the ISLR textbook. You will also encounter a few functions we did not cover in class. This will give you some practice on how to use a new function for the first time. You can try following steps:

1. Start by typing `?new_function` in your Console to open up the help page
2. Read the help page of this `new_function`. The description might be too technical for now. That's OK. Pay attention to the Usage and Arguments, especially the argument `x` or `x`, `y` (when two arguments are required)
3. At the bottom of the help page, there are a few examples. Run the first few lines to see how it works
4. Apply it in your homework questions

It is highly likely that you will encounter error messages while doing this homework. Here are a few steps that might help get you through it.

1. Locate which line is causing this error first
2. Check if you may have a typo in the code. Sometimes another person can spot a typo faster than you.
3. If you enter the code without any typo, try googling the error message
4. Scroll through the top few links see if any of them helps
5. If the error persists, post your error message on Piazza along with the code causing the problem and a few lines before and after it. If possible, include the stackoverflow links that you think might be helpful.

ISLR Chapter 2 Q8

This exercise relates to the College data set, which can be found in the file College.csv. It contains a number of variables for 777 different universities and colleges in the US. The variables are

- Private : Public/private indicator
- Apps : Number of applications received
- Accept : Number of applicants accepted
- Enroll : Number of new students enrolled
- Top10perc : New students from top 10 % of high school class
- Top25perc : New students from top 25 % of high school class
- F.Undergrad : Number of full-time undergraduates
- P.Undergrad : Number of part-time undergraduates
- Outstate : Out-of-state tuition
- Room.Board : Room and board costs
- Books : Estimated book costs
- Personal : Estimated personal spending
- PhD : Percent of faculty with Ph.D.'s
- Terminal : Percent of faculty with terminal degree
- S.F.Ratio : Student/faculty ratio
- perc.alumni : Percent of alumni who donate

- Expend : Instructional expenditure per student
- Grad.Rate : Graduation rate

Before reading the data into R, it can be viewed in Excel or a text editor. Make sure that you have the directory set to the correct location for the data.

(a) Use the `read.csv()` function to read the data into R. Call the loaded data `college`.

(b) Look at the data using the `View()` function. You should notice that the first column is just the name of each university. Load your data and then try the following commands:

```
#set your working directory ,fill in your code after this line

#read in the file College.csv using read.csv()
college = read.csv("College.csv")
#Give data frame college rownames
rownames(college) <- college[,1]
#View(college)
```

(c) You should see that there is now a row.names column with the name of each university recorded. This means that R has given each row a name corresponding to the appropriate university. R will **not** try to perform calculations on the row names. Next, we will remove the first column in the data where the names are stored. Try

```
#Use a negative number to generate a subset with all but one columnn
# college[, -c(1, 2, 3)]  will generate a subset with all but the first three columns
college <- college[,-1]
#View(college)
```

Now you should see that the first data column is Private. Note that another column labeled row.names now appears before the Private column. However, this is not a data column but rather the name that R is giving to each row.

i. Use the `summary()` function to produce a numerical summary of the variables in the data set. *Hint*: `summary()` takes in an object such as data.frame and return the summery results

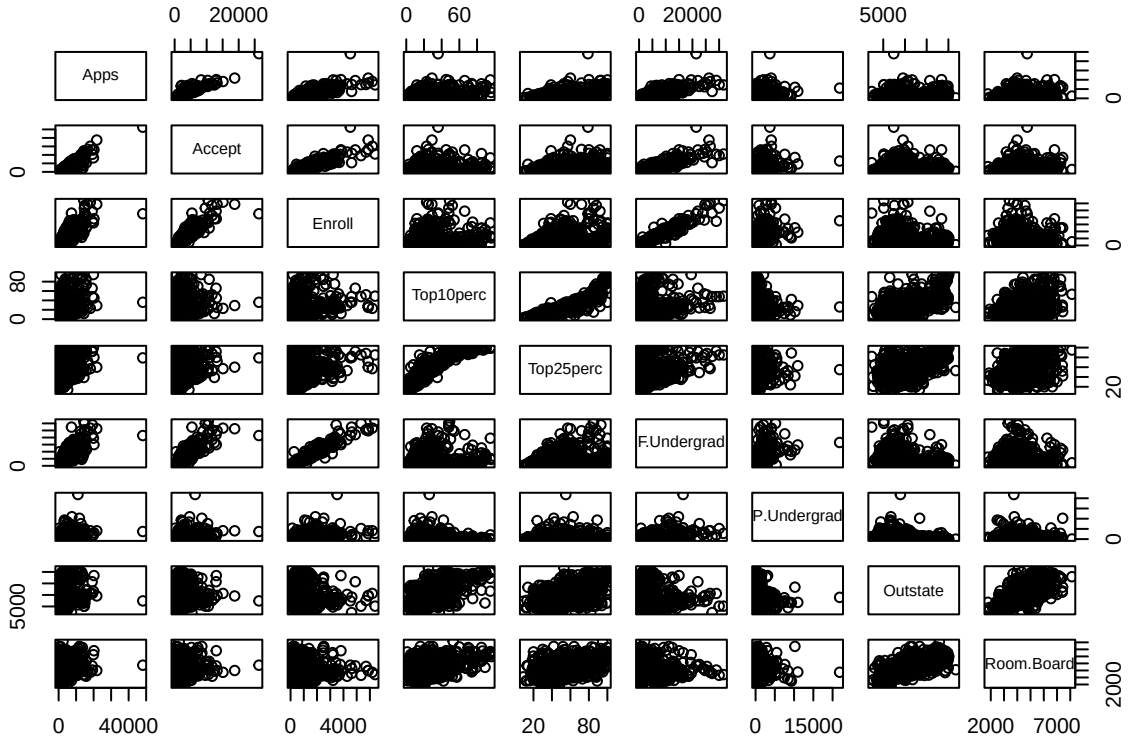
```
summary(college)
```

```
##    Private               Apps           Accept          Enroll
##  Length:777         Min.   :   81   Min.   :   72   Min.   :  35
##  Class :character   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242
##  Mode  :character   Median : 1558   Median : 1110   Median : 434
##                     Mean   : 3002   Mean   : 2019   Mean   : 780
##                     3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902
##                     Max.   :48094   Max.   :26330   Max.   :6392
##    Top10perc       Top25perc      F.Undergrad     P.Undergrad
##  Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0
##  1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0
##  Median :23.00   Median : 54.0   Median : 1707   Median :  353.0
##  Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3
##  3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0
##  Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0
##     Outstate       Room.Board       Books           Personal
##  Min.   : 2340   Min.   :1780   Min.   :  96.0   Min.   : 250
##  1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850
##  Median : 9990   Median :4200   Median : 500.0   Median :1200
##  Mean   :10441   Mean   :4358   Mean   : 549.4   Mean   :1341
##  3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700
##  Max.   :21700   Max.   :8124   Max.   :2340.0   Max.   :6800
```

```
##       PhD            Terminal       S.F.Ratio      perc.alumni
##  Min.   :  8.00   Min.   : 24.0   Min.   : 2.50   Min.   : 0.00
##  1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00
##  Median : 75.00   Median : 82.0   Median :13.60   Median :21.00
##  Mean   : 72.66   Mean   : 79.7   Mean   :14.09   Mean   :22.74
##  3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00
##  Max.   :103.00   Max.   :100.0   Max.   :39.80   Max.   :64.00
##      Expend        Grad.Rate
##  Min.   : 3186   Min.   : 10.00
##  1st Qu.: 6751   1st Qu.: 53.00
##  Median : 8377   Median : 65.00
##  Mean   : 9660   Mean   : 65.46
##  3rd Qu.:10830   3rd Qu.: 78.00
##  Max.   :56233   Max.   :118.00
```

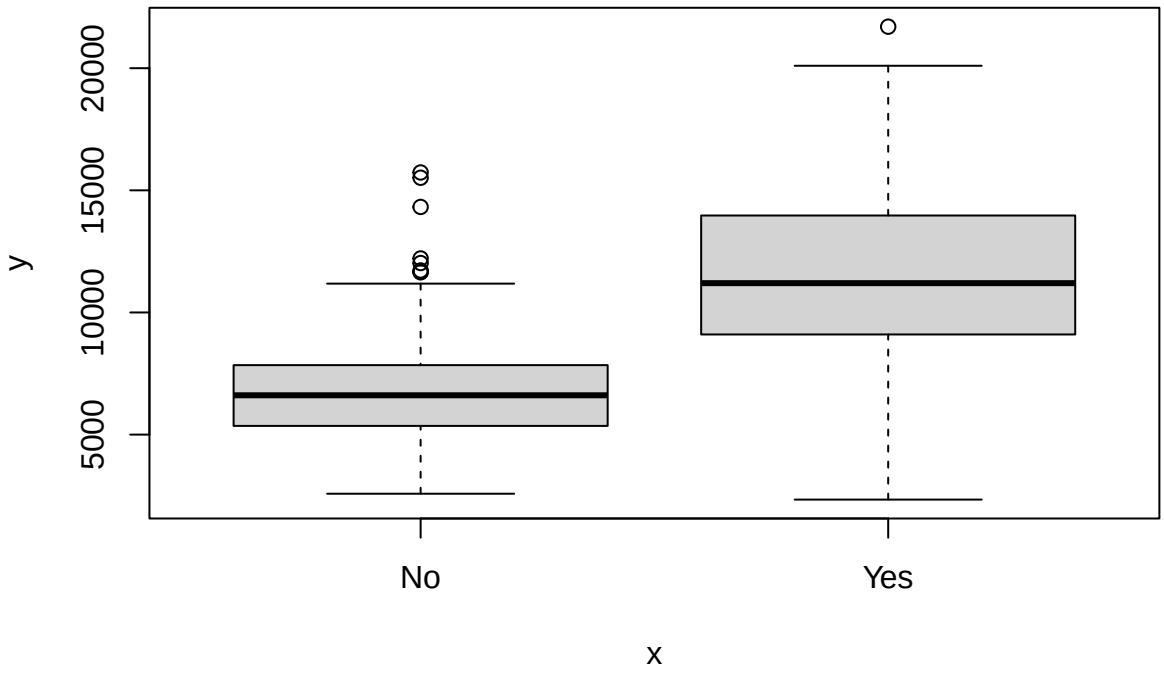
ii. Use the `pairs()` function to produce a scatterplot matrix of the first ten columns or variables of the data. Recall that you can reference the first ten columns of a data frame `dat` using `dat[,1:10]`

```
pairs(college[,2:10])
```



iii. Use the `plot()` function to produce side-by-side boxplots of Outstate versus Private. *Hint*: `plot()` takes two arguments one vector for x axis and one vector for y axis.

```
plot(factor(college$Private), college$Outstate)
```



iv. Create a new qualitative variable ("Yes" or "No"), called Elite, by binning the Top10perc variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes exceeds 50 %. Try the following commands:

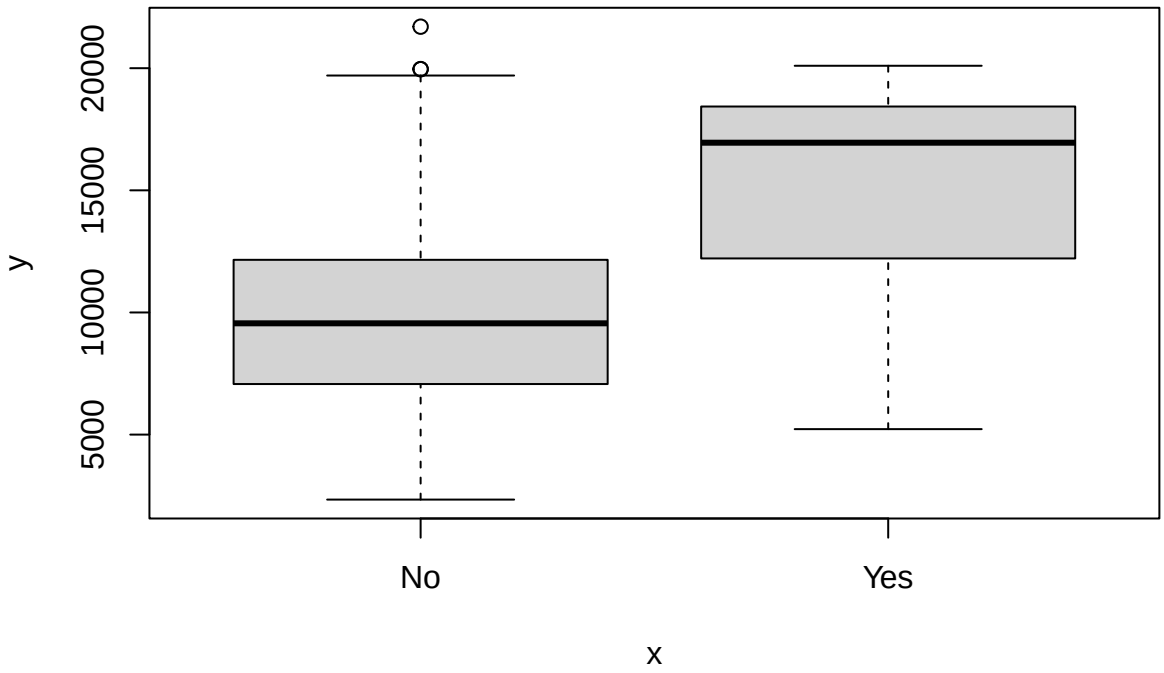
```
# replicate "No" for the same times as the number of colleges using rep()
Elite <- rep("No",nrow(college))
# change the values in Elite for colleges with proportion of students
# coming from the top 10% of their high school classes
# exceeds 50 % to "Yes"
Elite[college$Top10perc >50] <- "Yes"
# as.factor change ELite, a character vector to a factor vector
# (we will touch on factors later in class)
Elite <- as.factor(Elite)
# add the newly created vector to the college data frame
college <- data.frame(college ,Elite)
```

Use the `summary()` function to see how many elite universities there are. Now use the `plot()` function to produce side-by-side boxplots of Outstate versus Elite. *Hint*: `summary()` can also take a column

```
summary(college$Elite)
```

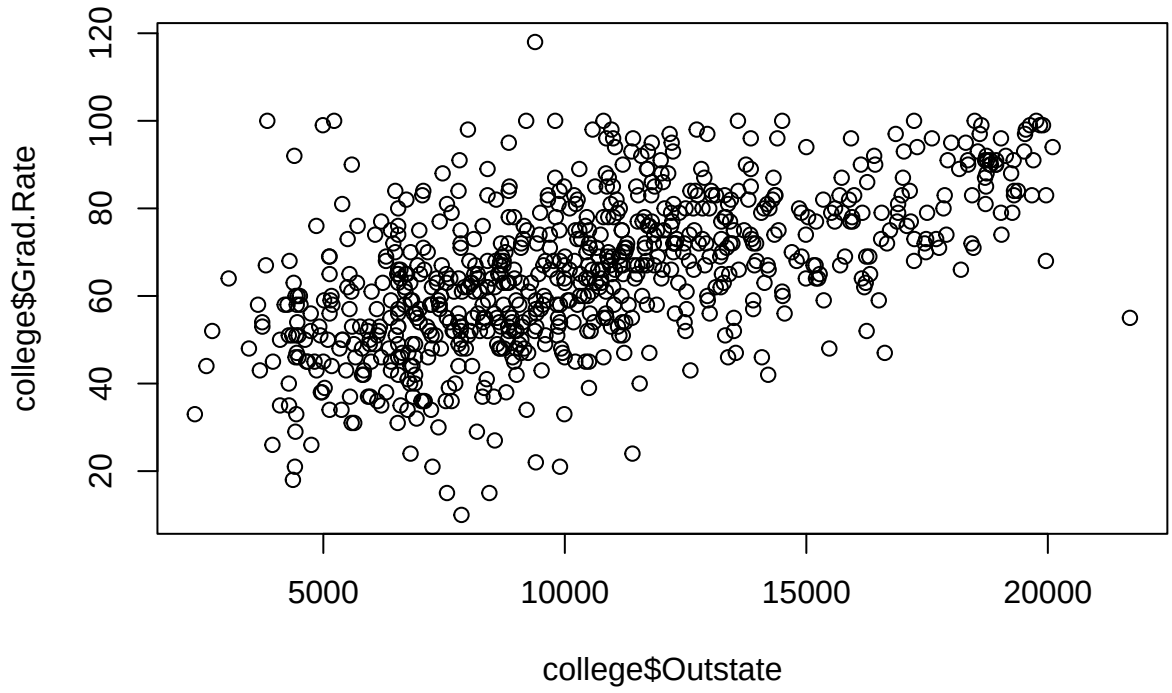
```
##  No Yes
## 699  78
```

```
plot(college$Elite, college$Outstate)
```

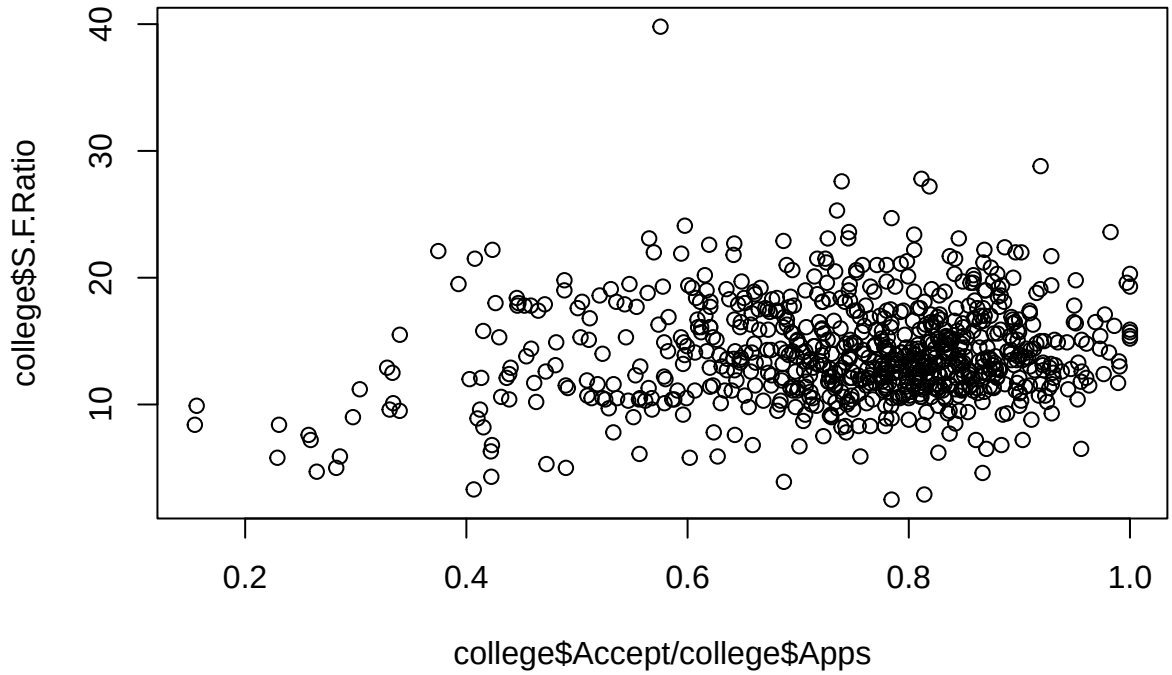


v. Continue exploring the data, and provide a brief summary of what you discover.

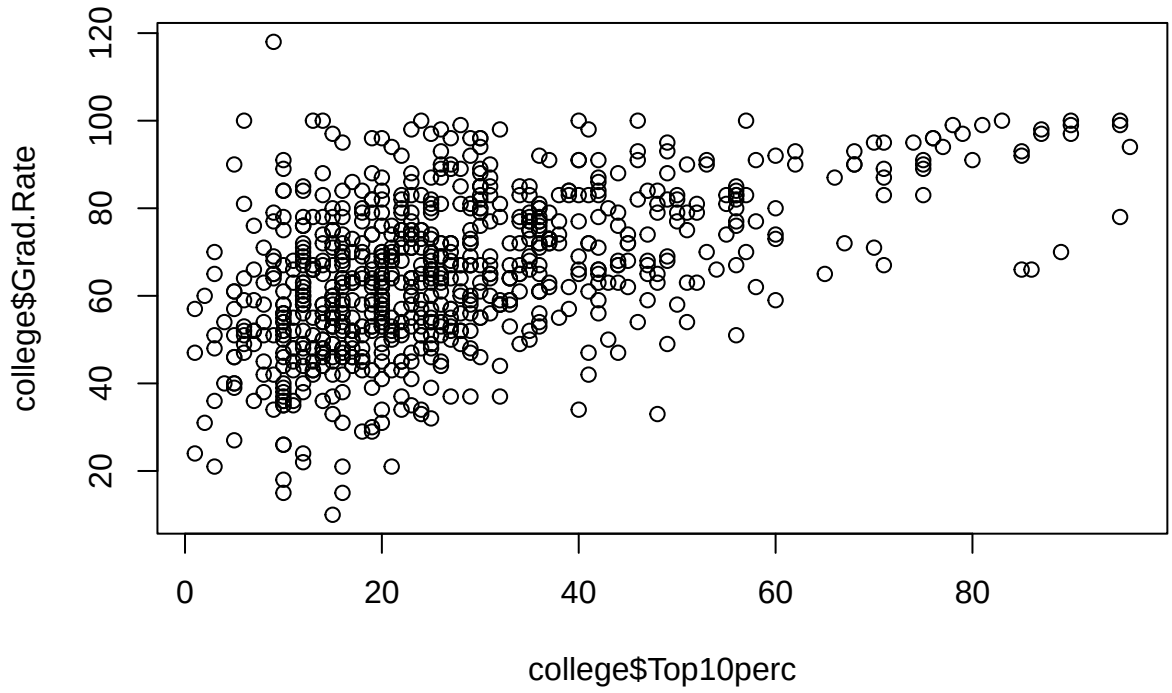
```
plot(college$Outstate, college$Grad.Rate)
```




```
# High tuition correlates to high graduation rate.
plot(college$Accept / college$Apps, college$S.F.Ratio)
```




```
# Colleges with low acceptance rate tend to have low S:F ratio.
plot(college$Top10perc, college$Grad.Rate)
```




```
# Colleges with the most students from top 10% perc don't necessarily have
# the highest graduation rate. Also, rate > 100 is erroneous!
```

ISLR Chapter 2 Q9

This exercise involves the Auto data set. `na.omit()` removes the missing values from the data and returns a new data frame.

```
#load the Auto.csv into a variable called auto using read.csv()
auto <- read.csv("Auto.csv")

# remove all rows with missing values using na.omit()
auto <- na.omit(auto)
```

We can use `class()` to check which of the columns are quantitative (numeric or integer), and which are qualitative(logical or character). And `sapply()` function takes in a data frame and a function (in this case `class()`), apply the class function to each column. Try the following commands

```
#apply the class() function to each column of auto data frame
sapply(auto, class)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration
##    "numeric"    "integer"    "numeric"    "integer"    "integer"    "numeric"
##         year       origin         name
##    "integer"    "integer"  "character"
```

(a) What is the range of each quantitative columns? You can answer this using the `range()` function. *Hint*: You can call `range()` function individually on each column. You can also subset the quantitative columns using quant_cols and then use `sapply` the function `range()` with the data frame with only quantitative columns

```
sapply(auto[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,]  9.0         3           68         46   1613          8.0   70
## [2,] 46.6         8          455        230   5140         24.8   82
```

(b) Using the functions `mean()` and `sd()`. Find out what is the mean and standard deviation of each quantitative columns?

```
sapply(auto[, 1:7], mean)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration
##    23.445918     5.471939   194.411990   104.469388  2977.584184    15.541327
##         year
##    75.979592
```

```
sapply(auto[, 1:7], sd)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration
##     7.805007     1.705783   104.644004    38.491160   849.402560     2.758864
##         year
##     3.683737
```

(c) Now remove the 10th through 85th observations (rows). What is the range, mean, and standard deviation of each column in the subset of the data that remains? *Hint*: We've seen removing columns in question 8. To remove the rows, we can use the negative sign - again. For example, auto[-c(1,3),] removes the first and third row

```
newAuto = auto[-(10:85),]
dim(newAuto) == dim(auto) - c(76,0)
```

```
## [1] TRUE TRUE
```

```
newAuto[9,] == auto[9,]
```

```
##    mpg cylinders displacement horsepower weight acceleration year origin name
## 9 TRUE      TRUE         TRUE       TRUE   TRUE         TRUE TRUE   TRUE TRUE
```

```
newAuto[10,] == auto[86,]
```

```
##     mpg cylinders displacement horsepower weight acceleration year origin name
## 87 TRUE      TRUE          TRUE       TRUE   TRUE         TRUE TRUE   TRUE TRUE
```

```
sapply(newAuto[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,] 11.0         3           68         46   1649          8.5   70
## [2,] 46.6         8          455        230   4997         24.8   82
```

```
sapply(newAuto[, 1:7], mean)
```

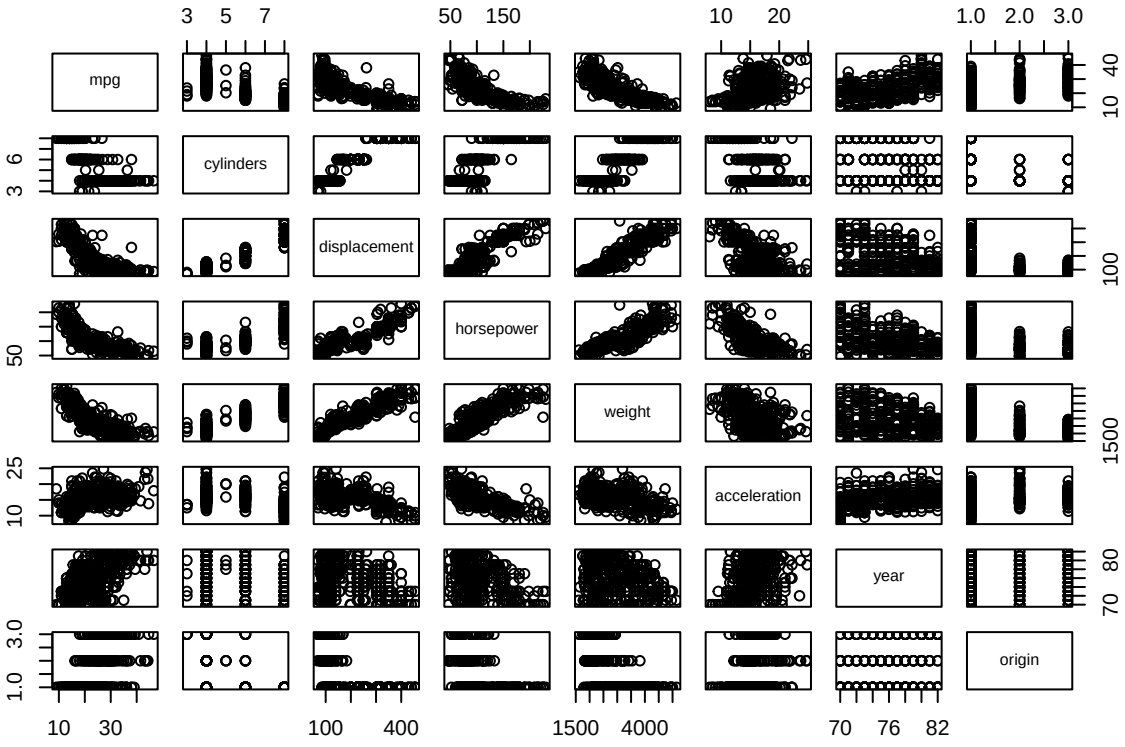
```
##          mpg    cylinders displacement   horsepower       weight acceleration
##    24.404430     5.373418   187.240506   100.721519  2935.971519    15.726899
##         year
##    77.145570
```

```
sapply(newAuto[, 1:7], sd)
```

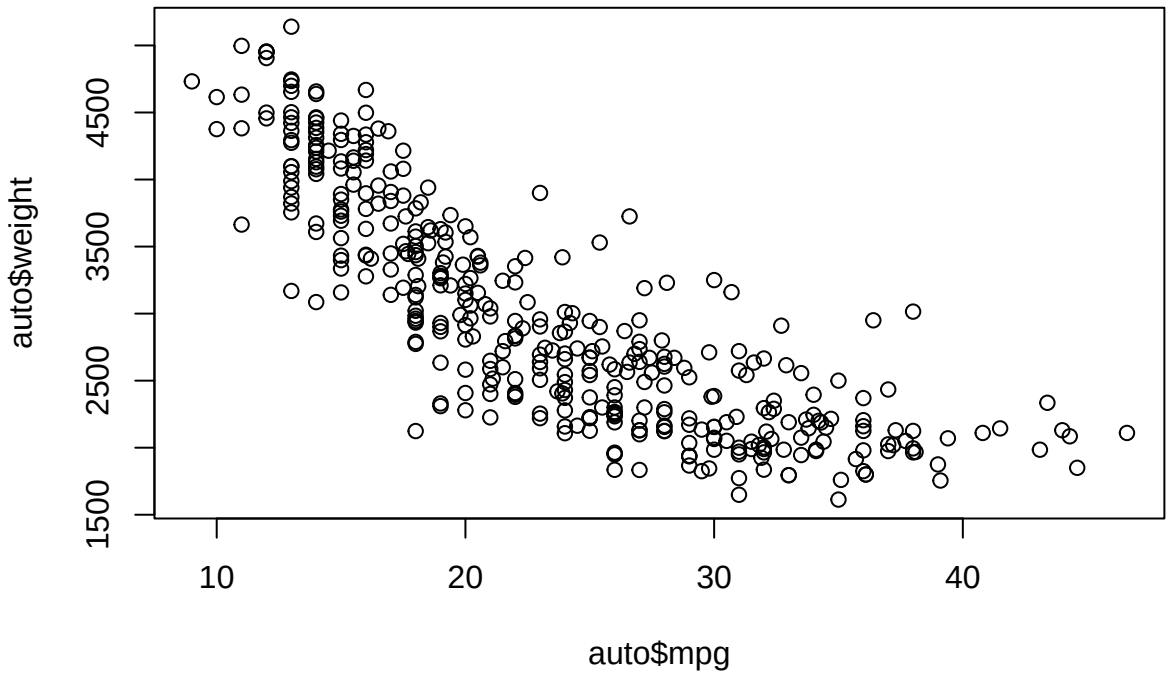
```
##          mpg    cylinders displacement   horsepower       weight acceleration
##     7.867283     1.654179    99.678367    35.708853   811.300208     2.693721
##         year
##     3.106217
```

(d) Using the full data set, investigate the columns graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the columns Comment on your findings.

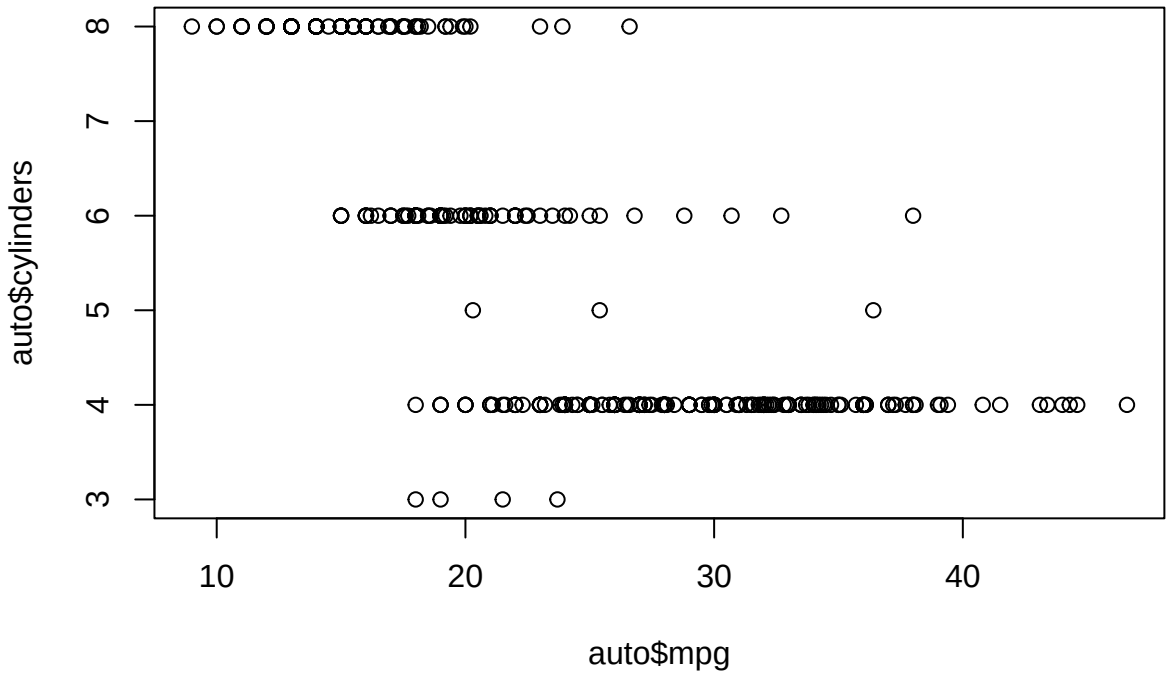
```
pairs(auto[, 1:8])
```



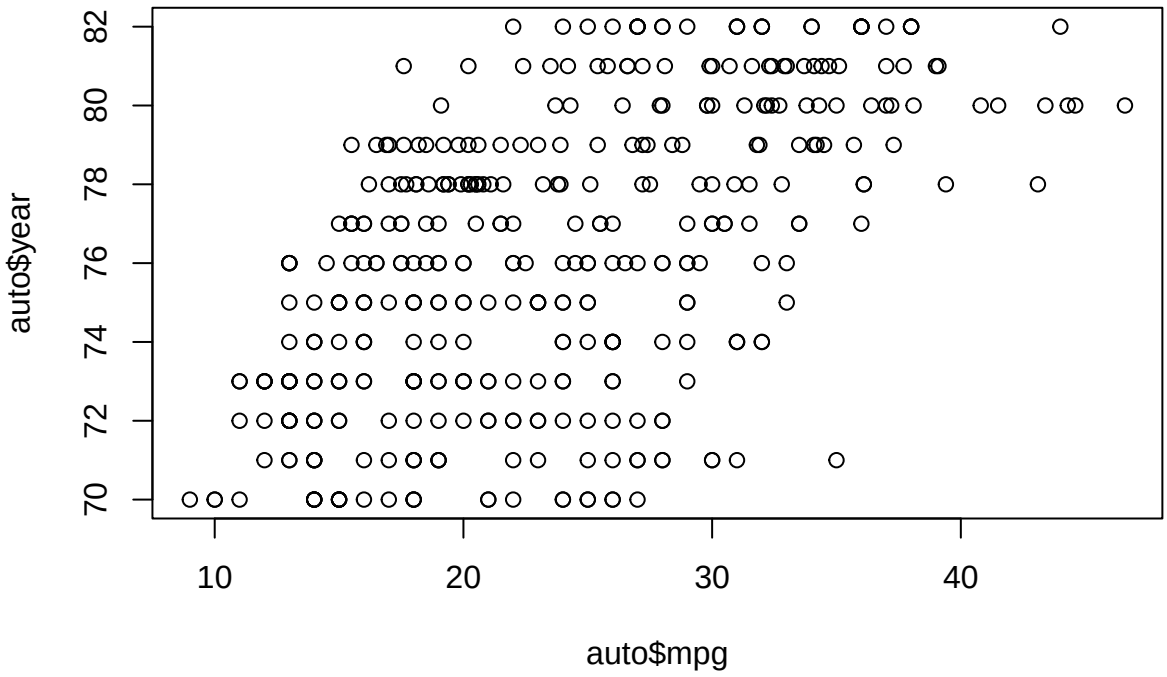

```
plot(auto$mpg, auto$weight)
```




```
# Heavier weight correlates with lower mpg.
plot(auto$mpg, auto$cylinders)
```



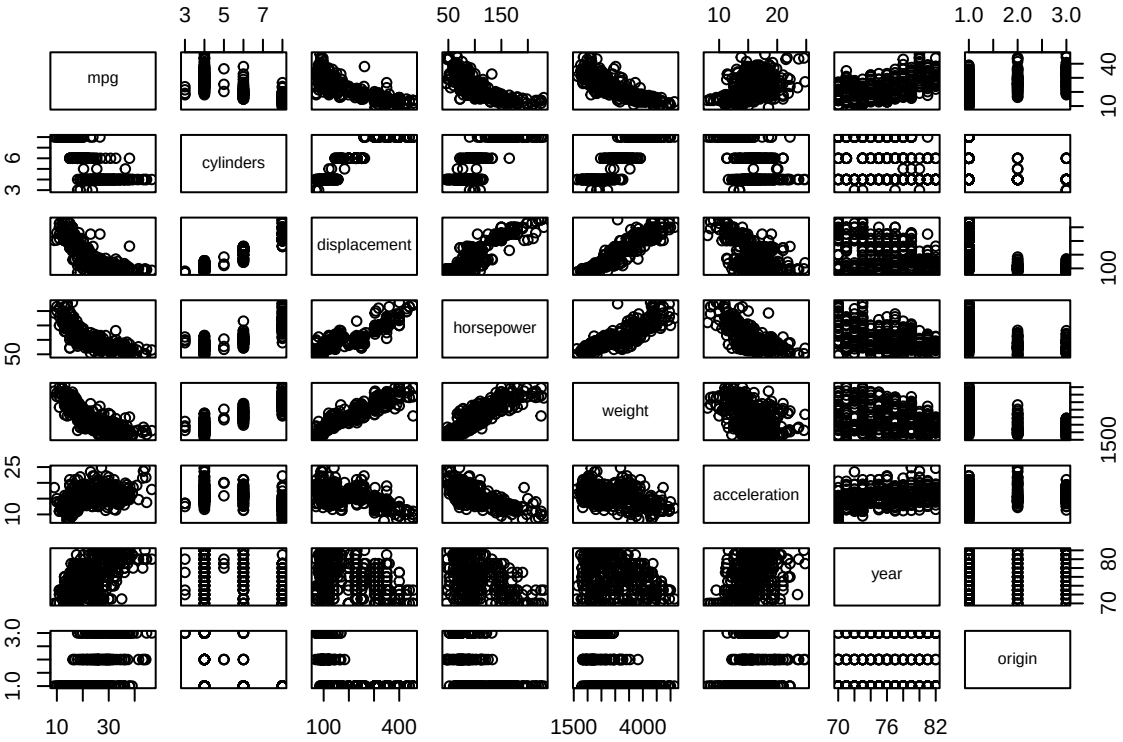

```
# More cylinders, less mpg.
plot(auto$mpg, auto$year)
```




```
# Cars become more efficient over time.
```

(e) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.

```
pairs(auto[,1:8])
```



ISLR Chapter 2 Q10

This exercise involves the Boston housing data set.

(a) To begin, load in the Boston data set. The Boston data set is part of the MASS library in R.

```
library(MASS)
?Boston
```

Now the data set is contained in the object Boston.

```
#Boston
```

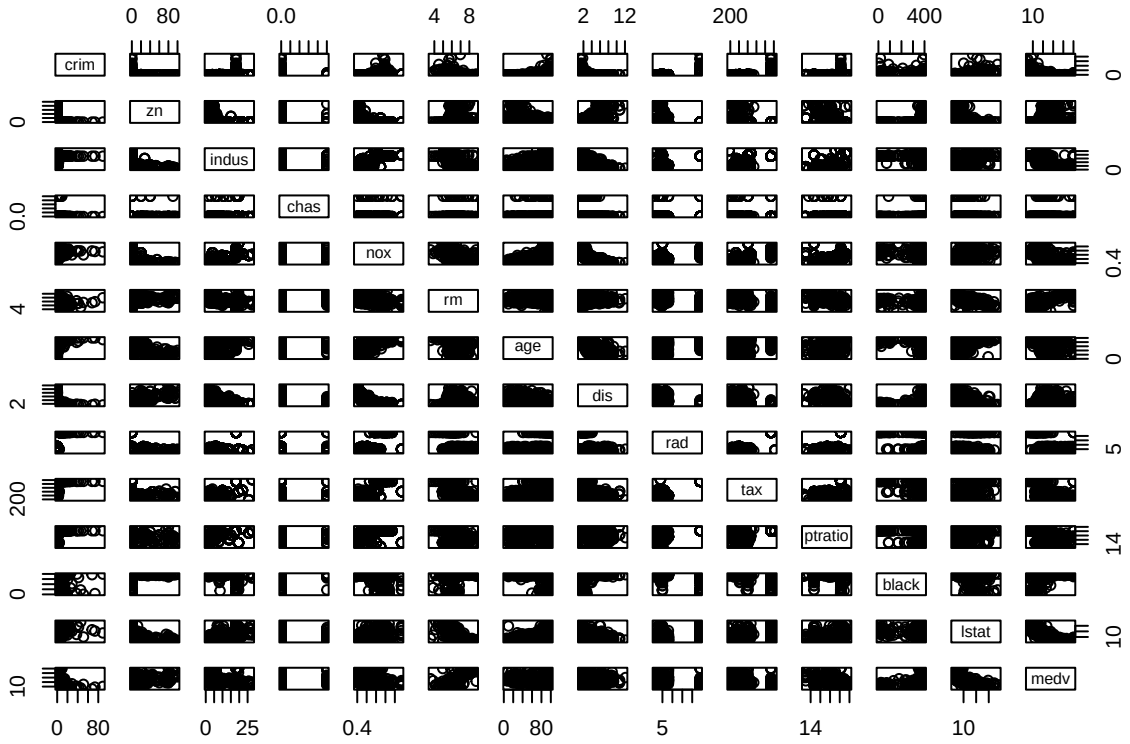
Read about the data set:

```
?Boston
```

How many rows are in this data set? How many columns? What do the rows and columns represent?

(b) Make some pairwise scatterplots of the columns in this data set. Describe your findings. *Hint*: Use function `pairs()`

```
pairs(Boston)
```



(c) Are any of the columns associated with per capita crime rate? If so, explain the relationship.

(d) Do any of the suburbs of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher ratios? Comment on the **range** of each these columns.

(e) How many of the suburbs in this data set bound the Charles river? *Hint*: Subset the data using a logical vector to check if variable `chas==1`, then use `dim()` to see the numer of suburbs.

```
dim(Boston[Boston$chas == 1,])
```

```
## [1] 35 14
```

(f) Using `median()`, find out what is the median pupil-teacher ratio among the towns in this data set?

```
median(Boston$ptratio)
```

```
## [1] 19.05
```

(g) Which suburb of Boston has lowest median value of owner-occupied homes? What are the values of the other variables(columns) for that suburb, and how do those values compare to the overall ranges for those variables(columns)? Comment on your findings. *Hint*: function `which.min()` gives the index of the lowest values in a vector. You can use this function to create a index for the suburb of Boston with the lowest median value of owner-occupied homes. Try

```
which.min(Boston$medv)
```

```
## [1] 399
```

```
which.max(Boston$medv)
```

```
## [1] 162
```

```
Boston[which.min(Boston$medv),]
```

```
##        crim zn indus chas   nox    rm age    dis rad tax ptratio black lstat
## 399 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 396.9 30.59
##     medv
## 399    5
```

(h) In this data set, how many of the suburbs average more than seven rooms per dwelling? More than eight rooms per dwelling? Comment on the suburbs that average more than eight rooms per dwelling.

```
dim(Boston[Boston$rm>7,])
```

```
## [1] 64 14
```

```
dim(Boston[Boston$rm>8,])
```

```
## [1] 13 14
```

```
summary(Boston[Boston$rm>8,])
```

```
##       crim              zn            indus             chas        
##  Min.   :0.02009   Min.   : 0.00   Min.   : 2.680   Min.   :0.0000  
##  1st Qu.:0.33147   1st Qu.: 0.00   1st Qu.: 3.970   1st Qu.:0.0000  
##  Median :0.52014   Median : 0.00   Median : 6.200   Median :0.0000  
##  Mean   :0.71879   Mean   :13.62   Mean   : 7.078   Mean   :0.1538  
##  3rd Qu.:0.57834   3rd Qu.:20.00   3rd Qu.: 6.200   3rd Qu.:0.0000  
##  Max.   :3.47428   Max.   :95.00   Max.   :19.580   Max.   :1.0000  
##       nox               rm             age              dis        
##  Min.   :0.4161   Min.   :8.034   Min.   : 8.40   Min.   :1.801  
##  1st Qu.:0.5040   1st Qu.:8.247   1st Qu.:70.40   1st Qu.:2.288  
##  Median :0.5070   Median :8.297   Median :78.30   Median :2.894  
##  Mean   :0.5392   Mean   :8.349   Mean   :71.54   Mean   :3.430  
##  3rd Qu.:0.6050   3rd Qu.:8.398   3rd Qu.:86.50   3rd Qu.:3.652  
##  Max.   :0.7180   Max.   :8.780   Max.   :93.90   Max.   :8.907  
##       rad              tax           ptratio          black       
##  Min.   : 2.000   Min.   :224.0   Min.   :13.00   Min.   :354.6  
##  1st Qu.: 5.000   1st Qu.:264.0   1st Qu.:14.70   1st Qu.:384.5  
##  Median : 7.000   Median :307.0   Median :17.40   Median :386.9  
##  Mean   : 7.462   Mean   :325.1   Mean   :16.36   Mean   :385.2  
##  3rd Qu.: 8.000   3rd Qu.:307.0   3rd Qu.:17.40   3rd Qu.:389.7  
##  Max.   :24.000   Max.   :666.0   Max.   :20.20   Max.   :396.9
```

```
##      lstat           medv      
##  Min.   :2.47   Min.   :21.9  
##  1st Qu.:3.32   1st Qu.:41.7  
##  Median :4.14   Median :48.3  
##  Mean   :4.31   Mean   :44.2  
##  3rd Qu.:5.12   3rd Qu.:50.0  
##  Max.   :7.44   Max.   :50.0
```