assign2sol

1(a). Use the read.csv() function to read the data into R. Call the loaded data college. Make sure that you have the directory set to the correct location for the data

```
college <- read.csv("https://scads.eecs.wsu.edu/wp-content/uploads/2017/09/College.csv")
```

1(b). Look at the data using the fix() function. You should notice that the first column is just the name of each university. We don't really want R to treat this as data. However, it may be

```
rownames (college)= college[,1]
fix(college)
```

However, we still need to eliminate the first column in the data where the names are stored. Try

```
college =college [,-1]
fix(college)
```

1(c).i. Use the summary() function to produce a numerical summary of the variables in the data set. (Respond to this question with the mean graduation rate included in the summary result).

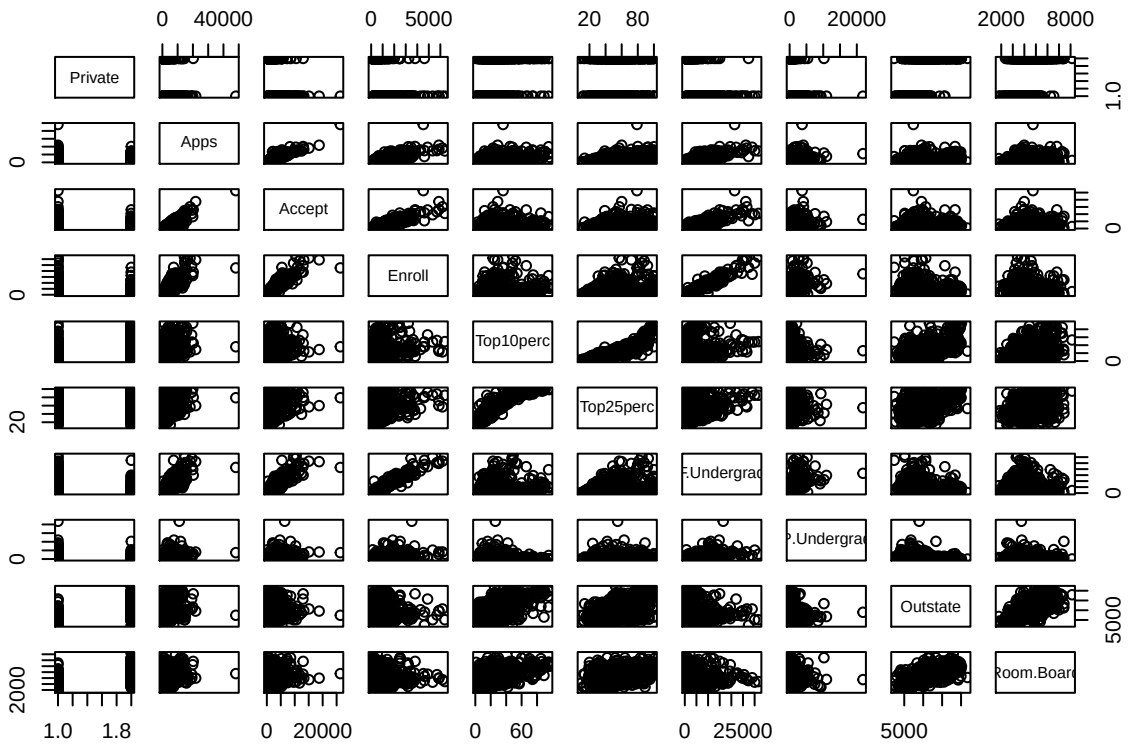
```
summary(college)
```

```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
```

```
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

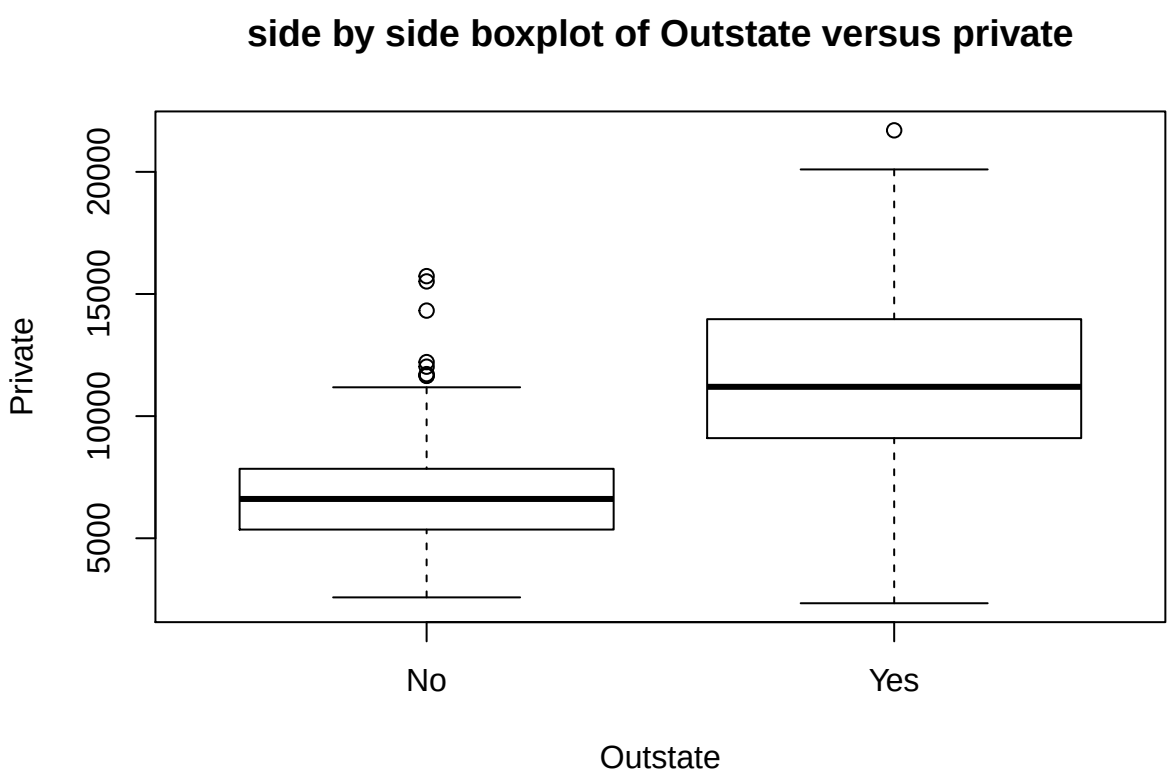
ii. Use the pairs() function to produce a scatterplot matrix of the first ten columns or variables of the data. Recall that you can reference the first ten columns of a matrix A using A[,1:10].

```
pairs(college[,1:10])
```

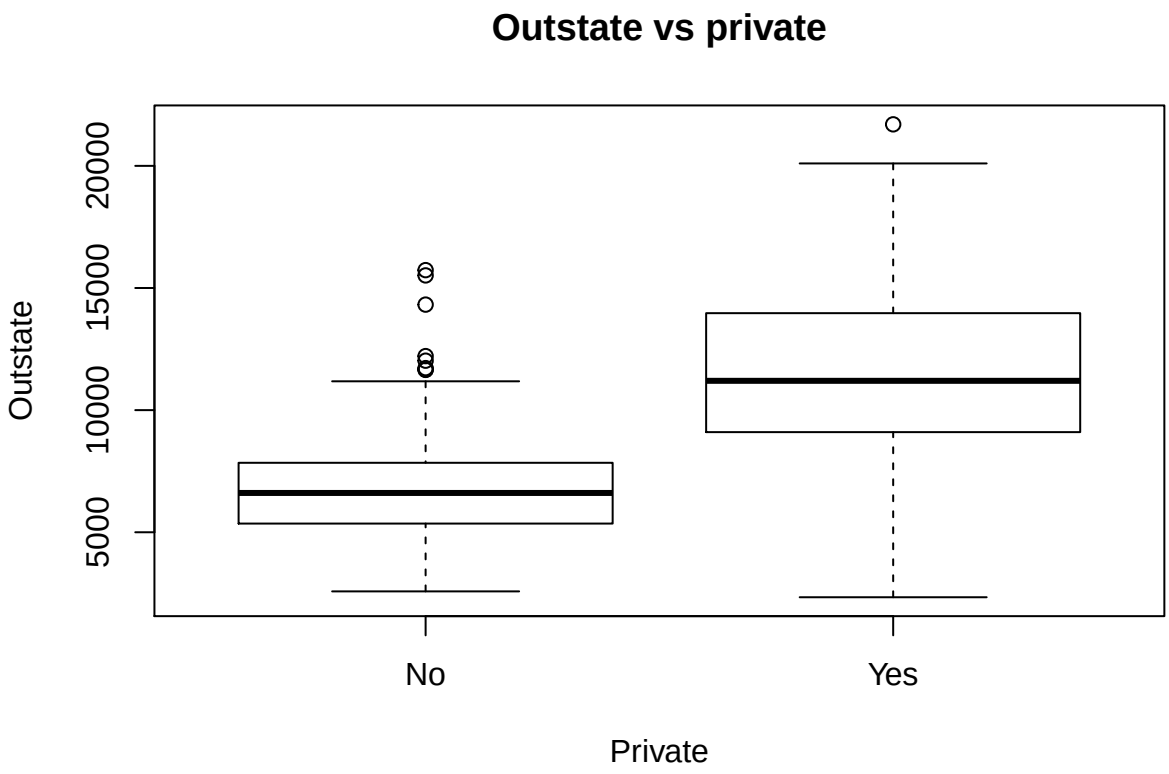


iii. Use the plot() function to produce side-by-side boxplots of Outstate versus Private.

```
plot(x=college$Private, y=college$Outstate,
     main="side by side boxplot of Outstate versus private",
     xlab="Outstate",
     ylab="Private")
```




```
boxplot(Outstate ~ Private,
        data = college,
        xlab = "Private",
        ylab = "Outstate",
        main = "Outstate vs private")
```



iv. Create a new qualitative variable, called Top, by binning the Top25perc variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 25% of their high school classes exceeds 50%.

```
Top=rep("No",nrow(college ))
 Top[college$Top25perc >50]=" Yes"
   Top=as.factor(Top)
 college=data.frame(college, Top)
```

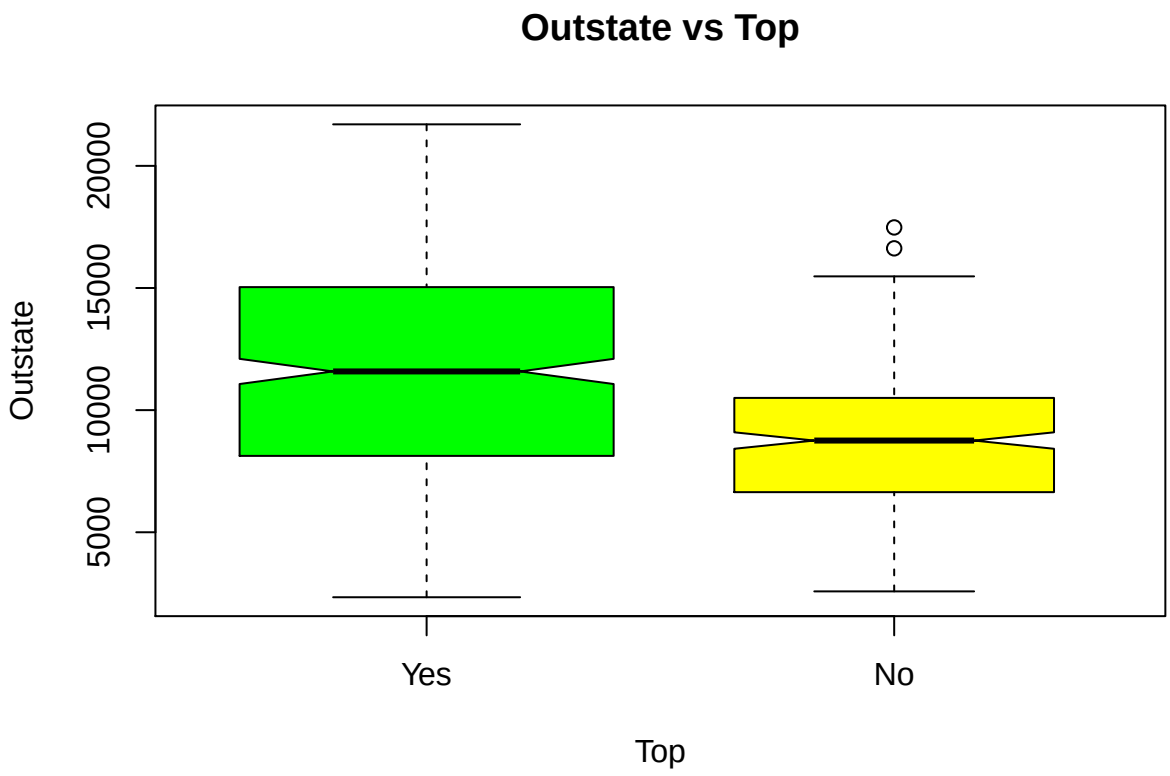
Use the summary() function to see how many top universities there are.

```
summary(Top)
```

```
##  Yes   No
##  449  328
```

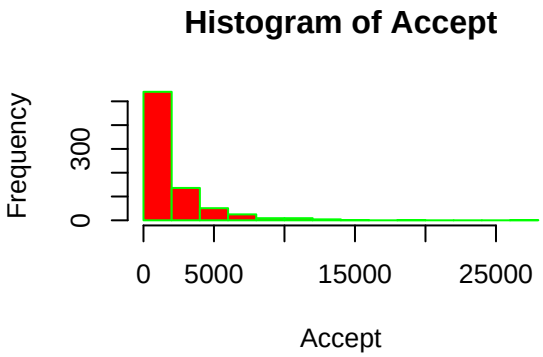
Now use the plot() or boxplot() function to produce side-by-side boxplots of Outstate with respect to the two Top categories (Yes and No). Ensure that this figure has an appropriate title and axis labels.

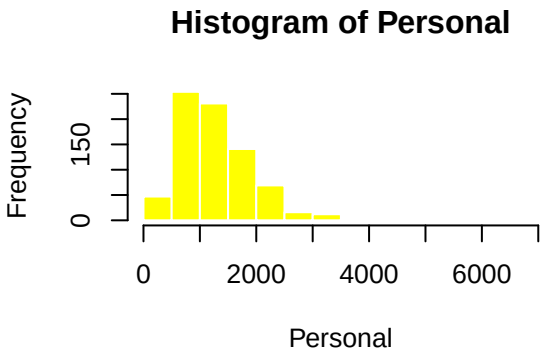
```
boxplot(Outstate ~ Top, data = college, xlab = "Top",
    ylab = "Outstate",
     main = "Outstate vs Top",
    notch = TRUE,
    varwidth = TRUE,
    col = c("green","yellow"),
    names = c("Yes","No"))
```

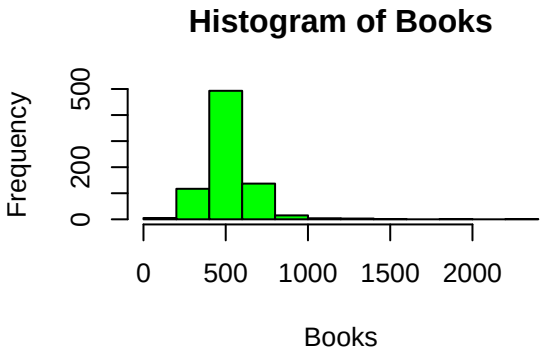


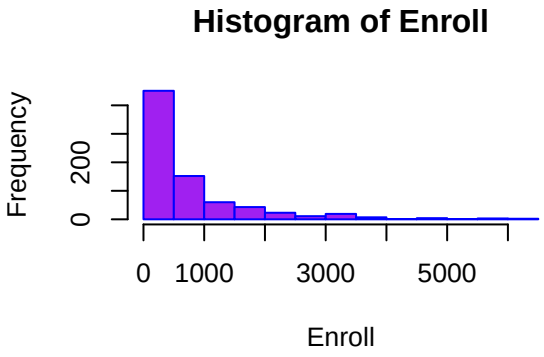
v. Use the hist() function to produce some histograms with differing numbers of bins for a few of the quantitative variables. You may find the command par(mfrow=c(2,2)) useful: it will divide the print window into four regions so that four plots can be made simultaneously. Modifying the arguments to this function will divide the screen in other ways. Again, ensure that this figure has an appropriate title and axis labels. auto

```
par(mfrow=c(2,2))
Enroll<- college$Enroll
Books<- college$Books
Personal<- college$Personal
Accept<- college$Accept
hist(Accept, xlab = "Accept",col = "red",border = "green")
hist(Personal, xlab = "Personal",col = "yellow",border = "White")
hist(Books, xlab = "Books",col = "green",border = "black")
hist(Enroll, xlab = "Enroll",col = "Purple",border = "blue")
```





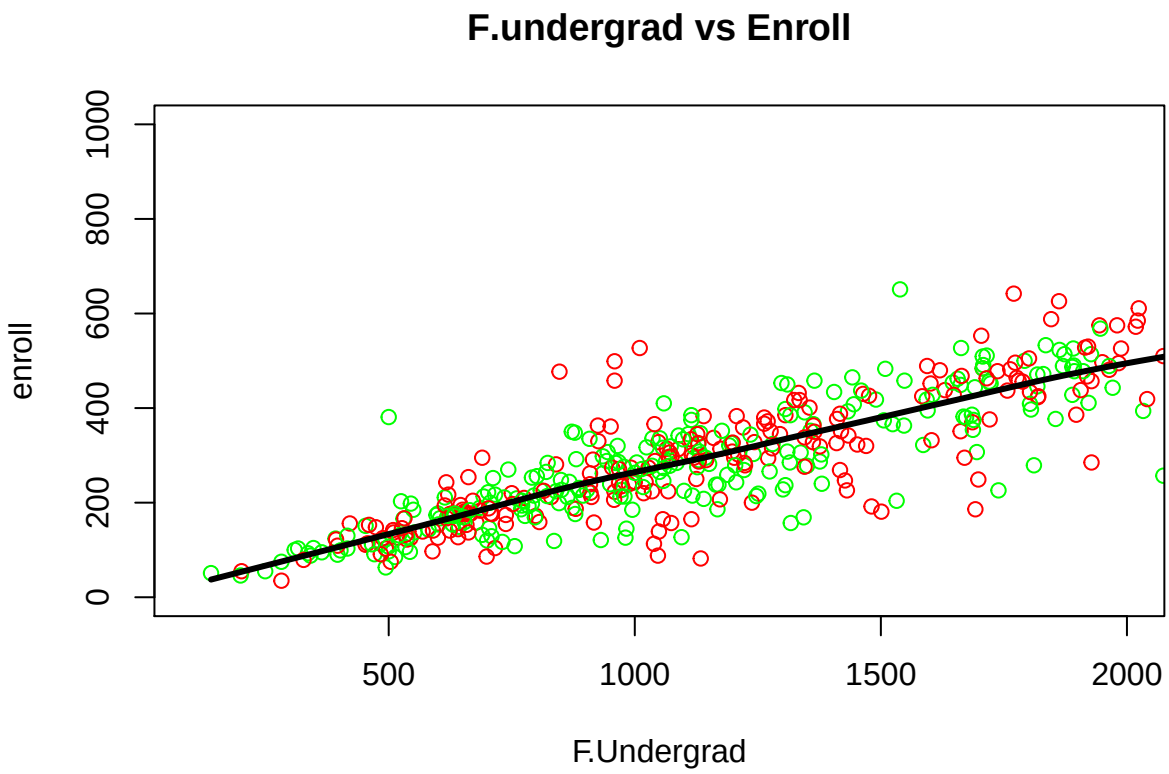




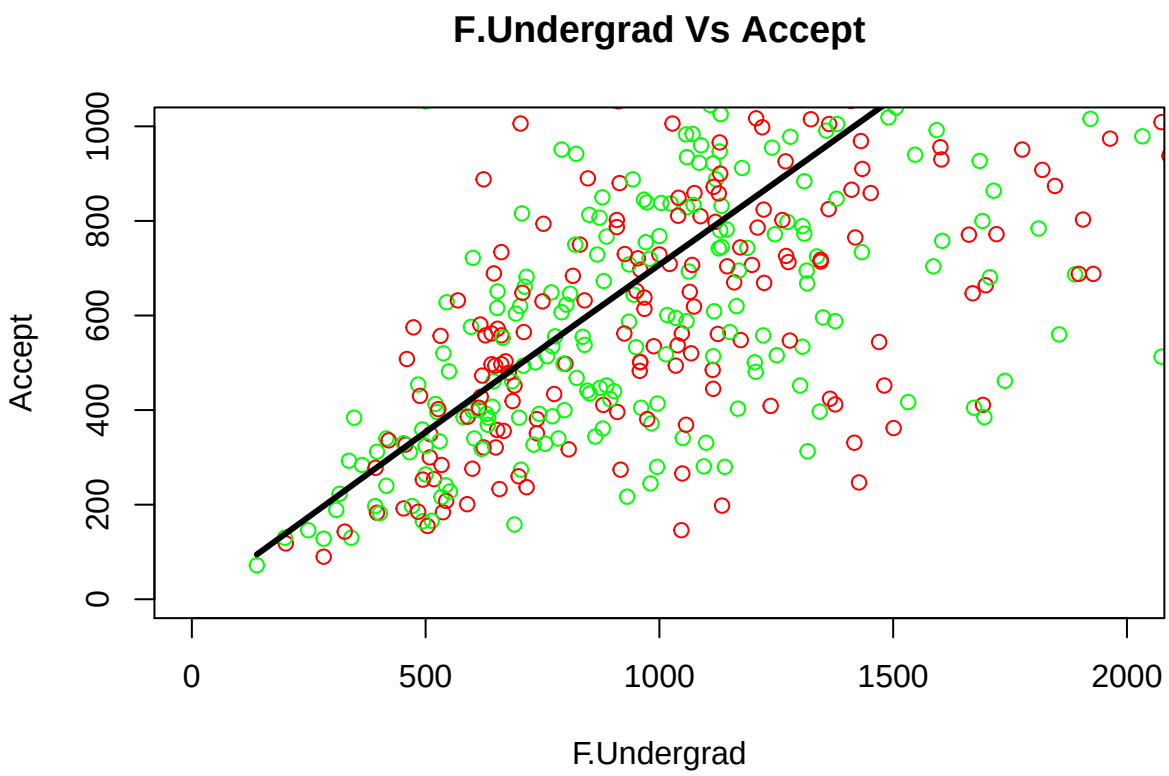
vi. Continue exploring the data, and provide a brief summary of what you discover. You may use additional plots or numerical descriptors as needed. Feel free to think outside the box on this one but if you want something to point you in the right direction, look at the summary statistics for various features, and think about what they tell you. Perhaps try plotting various features from the dataset against each other and see if any patterns emerge.

Solution: After plotting different possible combination of graphs. we can relate some variable such as mentioned below: (1.) The variables F.Undergrad and Enroll has almost a linear relationship. i.e if the variable enroll increases there will also be a increase in F.Undergrad. (2.)The variables F.Undergrad and Accept has almost a linear relationship. i.e if the variable Accept increases there will also be a increase in F.Undergrad. (3.) Variables Apps and Accept have almost a linear relationship.In other words they both are directly proportional in nature.

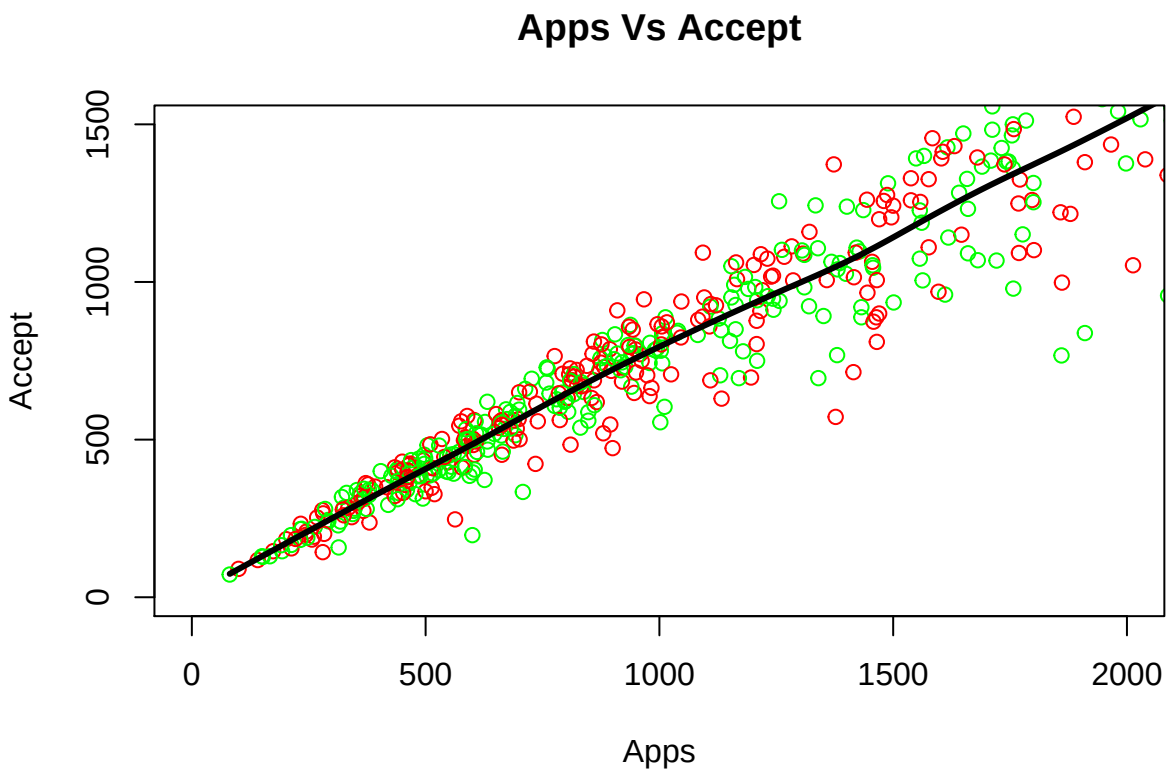
```
plot(x = college$F.Undergrad,
     y = college$Enroll ,
     xlim= c(100, 2000),
     ylim= c(00,1000),
     xlab = "F.Undergrad",
     ylab = "enroll",
     main ="F.undergrad vs Enroll",
     col = c("green","red"))

lines(smooth.spline(college$F.Undergrad,college$Enroll),lwd=3)
```




```
plot(x = college$F.Undergrad,
     y = college$Accept ,
     xlim=c(0,2000),
     ylim=c(0,1000),
     xlab ="F.Undergrad",
     ylab = "Accept",
     main = "F.Undergrad Vs Accept",
     col = c("green","red"))
lines(smooth.spline(college$F.Undergrad,college$Accept),lwd=3)
```




```
plot(x = college$Apps,
      y = college$Accept ,
      xlim=c(0,2000),
      ylim=c(0,1500),
      xlab ="Apps",
      ylab = "Accept",
      main = "Apps Vs Accept",
      col = c("green","red"))
 lines(smooth.spline(college$Apps,college$Accept),lwd=3)
```



2. This exercise involves the Auto.csv data set found on the course website. Make sure that the missing values have been removed from the data. To do this, consider the na.strings parameter of read.csv(), as well as the na.omit() function.

```
auto <- read.csv("https://scads.eecs.wsu.edu/wp-content/uploads/2017/09/Auto.csv", na.strings="?")
   auto <- na.omit(auto)
```

(a) Which of the predictors are quantitative, and which are qualitative?

Solution: Quantitaive=cylinders,weight || Qualitative=mpg,displacement, horsepower,year,origin,name

(b) What is the range of each quantitative predictor? You can answer this using the range() function. Hint: consider using R's sapply() function to take the range of multiple features in a single function call.

```
sapply((list(auto$mpg,
             auto$cylinders,
             auto$displacement,
             auto$horsepower,
             auto$weight,
             auto$acceleration,
             auto$year,
             auto$origin)),
       range)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]  9.0    3   68   46 1613  8.0   70    1
## [2,] 46.6    8  455  230 5140 24.8   82    3
```

(c) What is the mean and standard deviation of each quantitative predictor?

```
sapply((list(auto$mpg,
             auto$cylinders,
             auto$displacement,
             auto$horsepower,
             auto$weight,
             auto$acceleration,
             auto$year,
             auto$origin)),
       mean)
```

```
## [1]   23.445918    5.471939  194.411990  104.469388 2977.584184   15.541327
## [7]   75.979592    1.576531
```

```
sapply((list(auto$mpg,
             auto$cylinders,
             auto$displacement,
             auto$horsepower,
             auto$weight,
             auto$acceleration,
             auto$year,
             auto$origin)),
       sd)
```

```
## [1]   7.8050075   1.7057832 104.6440039  38.4911599 849.4025600   2.7588641
## [7]   3.6837365   0.8055182
```

(d) Now remove the 25th through 75th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains?

```
 newauto <- auto[-c(25:75), ]
View(newauto)

sapply((list(newauto$mpg,
             newauto$cylinders,
             newauto$displacement,
             newauto$horsepower,
             newauto$weight,
             newauto$acceleration,
             newauto$year,
             newauto$origin)),
       range)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] 11.0    3   68   46 1649  8.0   70    1
## [2,] 46.6    8  455  230 4997 24.8   82    3
```

```
sapply((list(newauto$mpg,
             newauto$cylinders,
             newauto$displacement,
             newauto$horsepower,
             newauto$weight,
             newauto$acceleration,
             newauto$year,
             newauto$origin)),
       mean)
```

```
## [1]   24.195894    5.360704  187.167155  101.395894 2920.947214   15.650147
```

```
## [7]   76.683284    1.612903
```

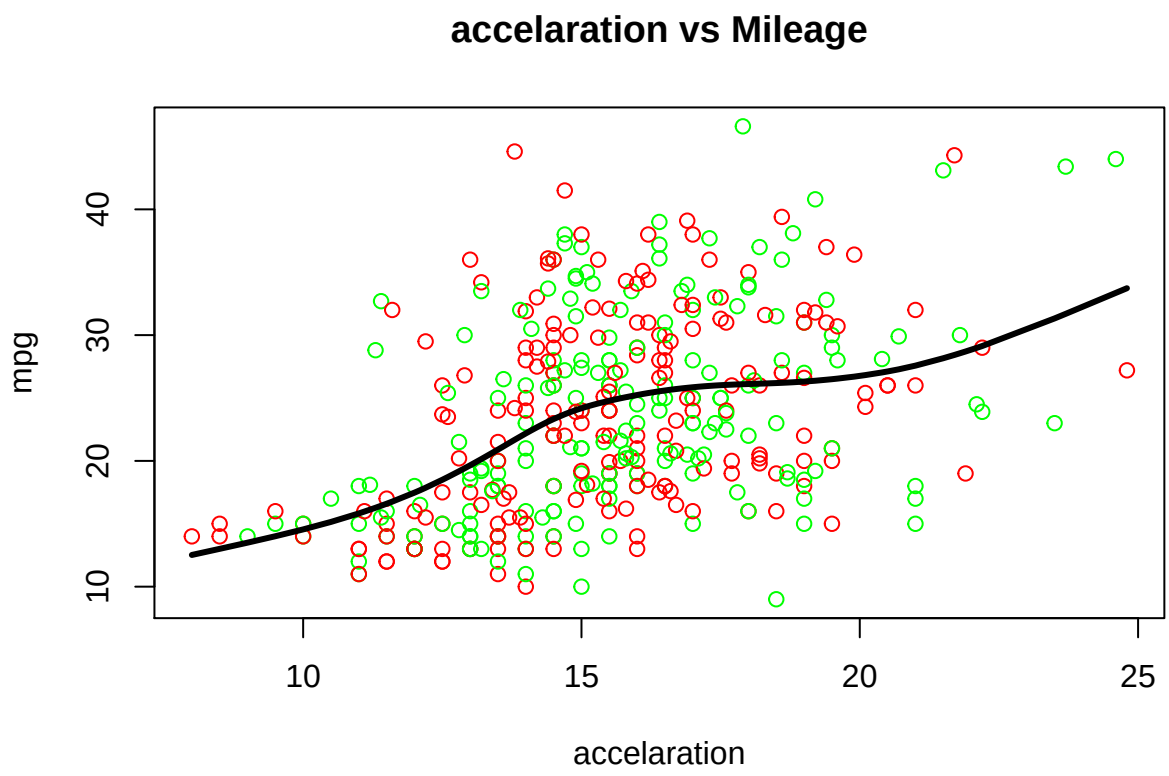
```
sapply((list(newauto$mpg,
             newauto$cylinders,
             newauto$displacement,
             newauto$horsepower,
             newauto$weight,
             newauto$acceleration,
             newauto$year,
             newauto$origin)),
       sd)
```

```
## [1]   7.7205330   1.6579873 101.1198397  36.2987423 799.6367540   2.7552156
## [7]   3.4247347   0.8169225
```

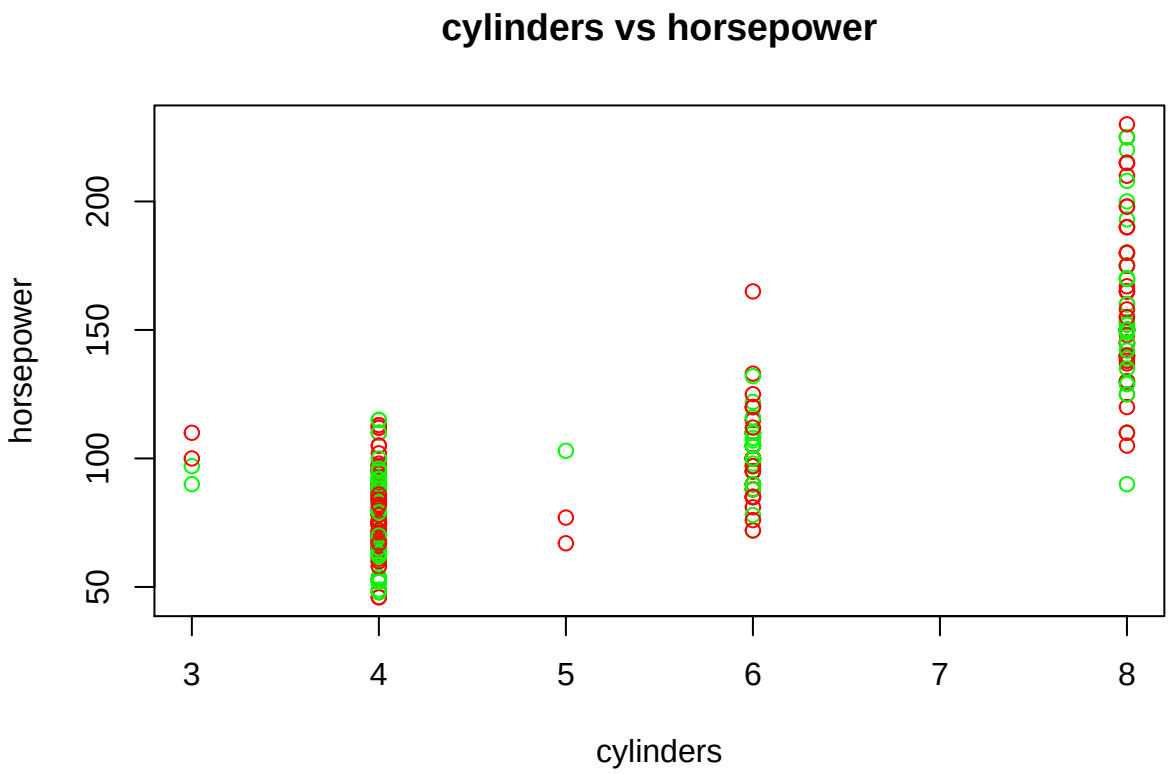
(e) Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

Solution: (1.) In the accelaration Vs. Mileage graph the relationship here is different. The variable mpg goes high only in a certain interval of acceleration. The variable mpg performs better from interval of accelaration x=(15,20). (2.) The graph between cylinders and horsepower depict that if the variable cylinder increases the variable horsepower goes high in performance.

```
plot(x = auto$acceleration,
     y = auto$mpg,
     xlab = "accelaration",
     ylab = "mpg",
     main = "accelaration vs Mileage",
     col = c("green","red"))
 lines(smooth.spline(auto$acceleration, auto$mpg),lwd=3)
```



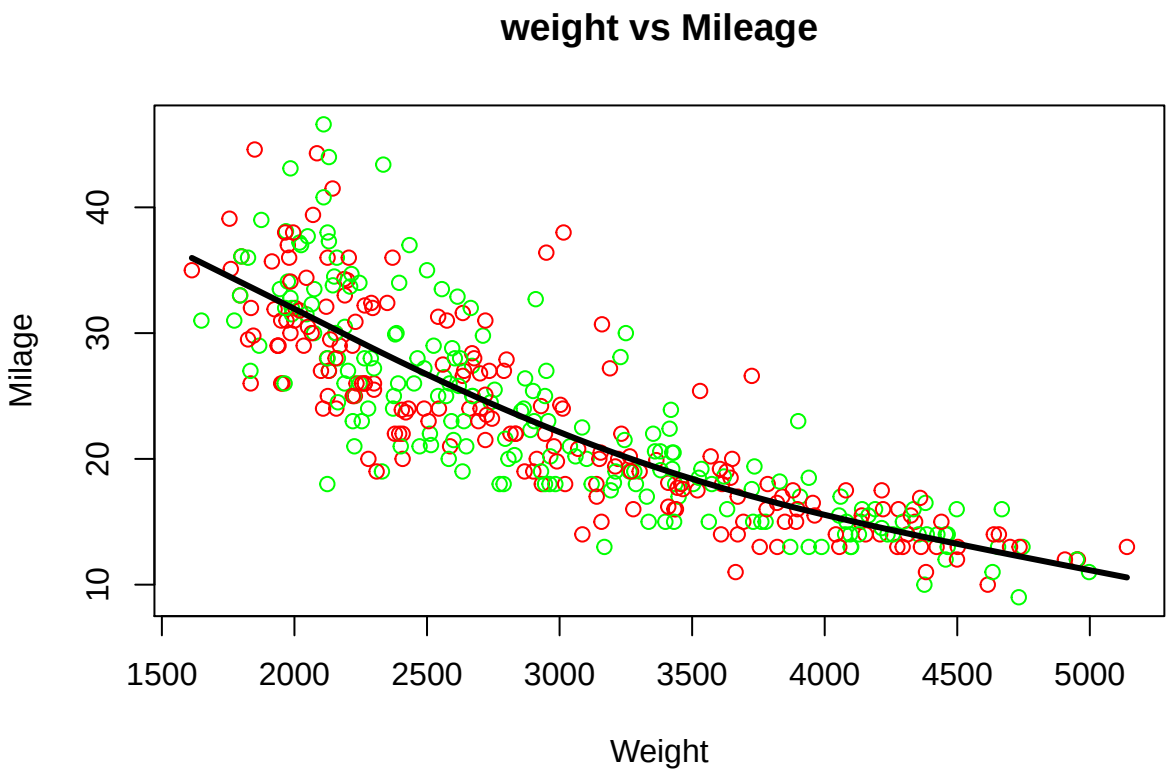

```
plot(x = auto$cylinders,
     y = auto$horsepower,
     xlab = "cylinders",
     ylab = "horsepower",
     main = "cylinders vs horsepower",
     col = c("green","red"))
```



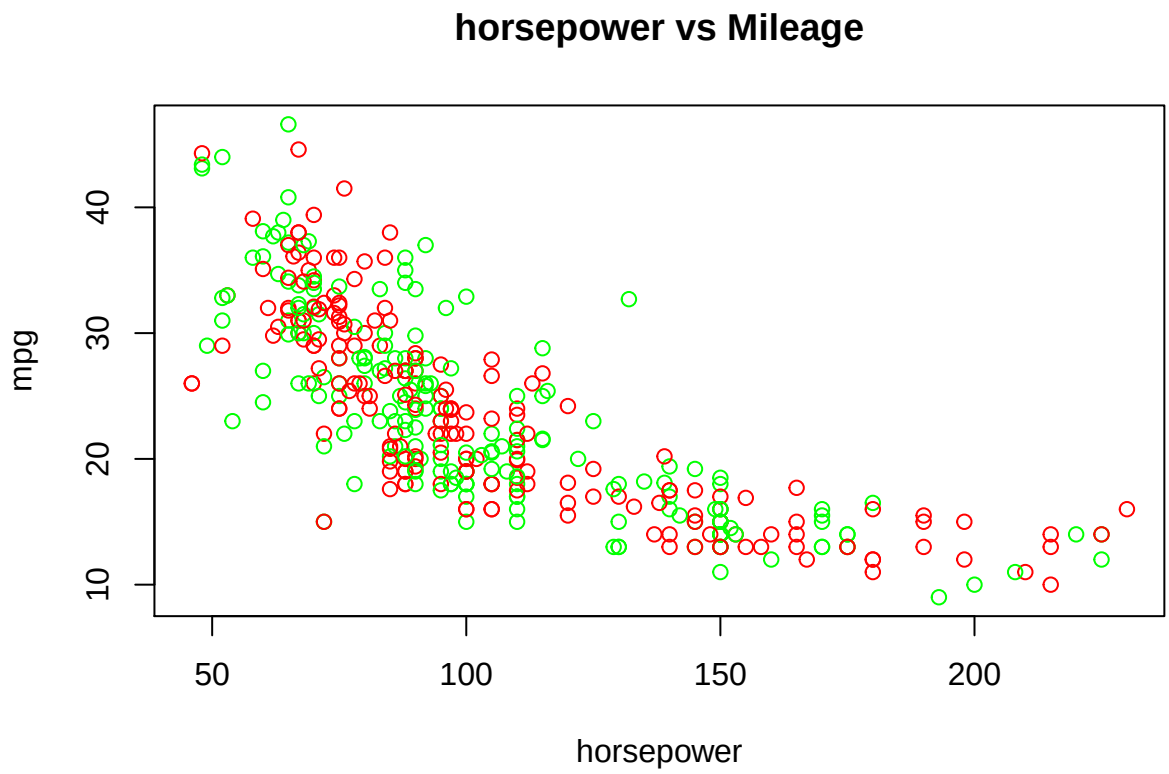
(f) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.

Solution: (1.) The plot between weight and mpg depicts that relationship is inversely proportional. Variable mpg is inversely proportional to weight variable in the given data. So a prediction of mpg can be made if weight variable is given. (2.) The graph between mpg and horsepower exhibit that the relationship is inversely proportional. That means if the variable horsepower increases then mpg variable goes down and the vice versa.The variable mpg is predictable if values of horsepower is given.

```
plot(x = auto$weight,
     y = auto$mpg,
     xlab = "Weight",
     ylab = "Milage",
     main = "weight vs Mileage",
     col = c("green","red"))
lines(smooth.spline(auto$weight, auto$mpg),lwd=3)
```




```
plot(x = auto$horsepower,
     y = auto$mpg,
     xlab = "horsepower",
     ylab = "mpg",
     main = "horsepower vs Mileage",
     col = c("green","red"))
```


horsepower vs Mileage
mpg
10
20
30
40
50
100
150
200
horsepower