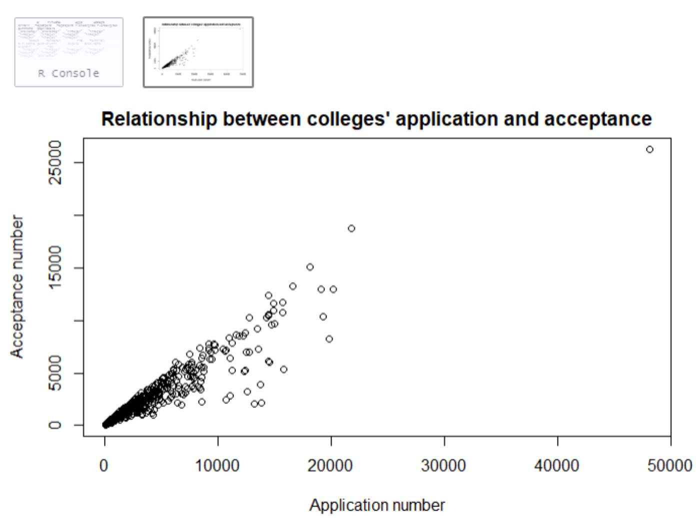
1.a

```
```{r 1.a}
##1.a
college <- read.csv("E:/WSU Graduate/CPT_S 575 Data Science/College.csv")

```
```

1.b

```
```{r 1.b}

##1.b
median(college$Room.Board)

college_private <- subset(college, Private == "Yes")

median(college_private$Room.Board)

college_public <- subset(college, Private == "No")

median(college_public$Room.Board)

```
```

```
[1] 4200
[1] 4400
[1] 3708
```

1.c

I firstly checked which features are numeric then I plot a scatterplot to show the relation ship between colleges' application and acceptance.

```
```{r 1.c}
##1.c
sapply(college, class)

plot(college$Apps, college$Accept, main = "Relationship between colleges' application and acceptance", xlab = "Application number", ylab =
"Acceptance number")

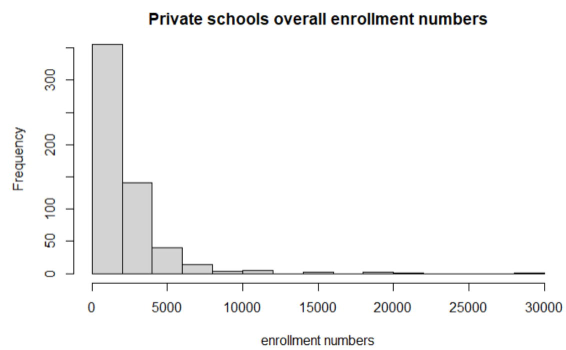
```
```

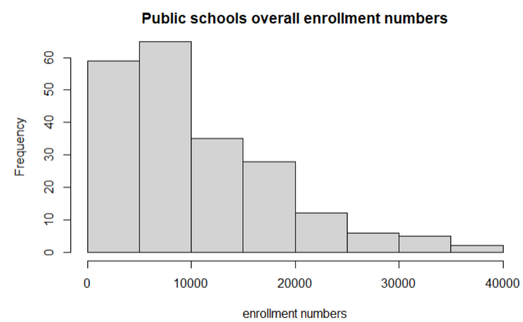
```
          X    Private       Apps     Accept     Enroll  Top10perc  Top25perc
F.Undergrad P.Undergrad   Outstate Room.Board
"character" "character"  "integer"  "integer"  "integer"  "integer"  "integer"
"integer"   "integer"  "integer"  "integer"
      Books   Personal        PhD   Terminal  S.F.Ratio perc.alumni     Expend
Grad.Rate     overall        Top
  "integer"   "integer"  "integer"  "integer"  "numeric"  "integer"  "integer"
"integer"   "integer" "character"
```



1.d

```{r 1.d}
##1.d
college$overall = college$F.Undergrad + college$P.Undergrad

college_private <- subset(college, Private == "Yes")

college_public <- subset(college, Private == "No")

hist(college_private$overall, xlab = "enrollment numbers", main = "Private schools overall enrollment numbers")

hist(college_public$overall, xlab = "enrollment numbers", main = "Public schools overall enrollment numbers")
```





1.e

```{r 1.e}
##1.e
college$Top <- ifelse(college$Top <- college$Top10perc >= 75, "Yes", "No")

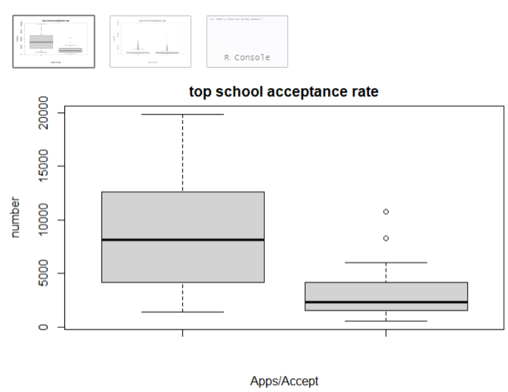
college_top <- subset(college, Top == "Yes")

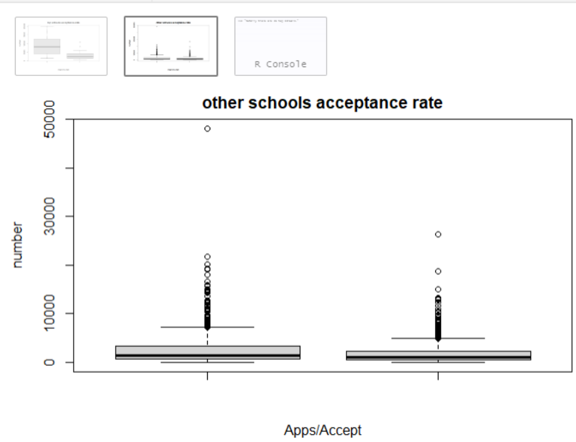
college_nontop <- subset(college, Top == "No")

boxplot(college_top$Apps, college_top$Accept, main = "top school acceptance rate", xlab = "Apps/Accept", ylab = "number")

boxplot(college_nontop$Apps, college_nontop$Accept, main = "top school acceptance rate", xlab = "Apps/Accept", ylab = "number")

print("totally there are 26 top schools.")
```





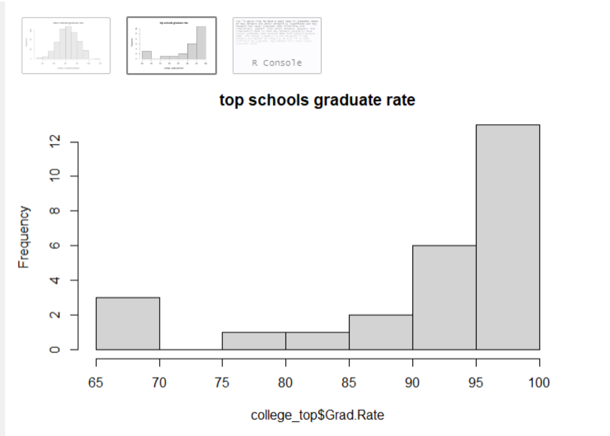
Totally there are 26 top universities.

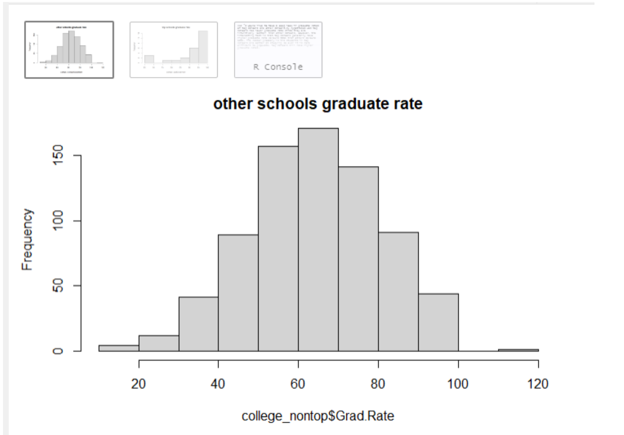
1.f

````
```{r (1.f)}
##1.f
hist(college_nontop$Grad.Rate, main = "other schools graduate rate")

hist(college_top$Grad.Rate, main = "top schools graduate rate")

print("I would like to take a deep look in graduate rates of top schools and other schools.My hypotheses was top schools had lower graduate rate since they are intuitively 'better' than other schools. However, the interesting fact is that top schools generally have higher graduate rate (around 90%) than others (around 60%). The reason probably is the students in top schools are better at studying so even it is more difficult to graduate, top schools sill have higher graduate rates.")
```
````





I would like to take a deep look in graduate rates of top schools and other schools. My hypotheses was top schools had lower graduate rate since they are intuitively 'better' than other schools. However, the interesting fact is that top schools generally have higher graduate rate (around 90%) than others (around 60%). The reason probably is the students in top schools are better at studying so even it is more difficult to graduate, top schools sill have higher graduate rates.

2.a

````
```{r 2.a}
##2.a
forestfires <- read.csv("E:/WSU Graduate/CPT_S 575 Data Science/forestfires.csv")

sapply(forestfires,class)

print("'FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain', 'area' can be considered as quantitative predictors. 'month' and 'day' can be considered as quanlitative predictors but they are easily represented as quantative predictors.")
```
````

'FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain', 'area' can be considered as quantitative predictors. 'month' and 'day' can be considered as qualitative predictors, but they are easily represented as quantitative predictors."

2.b

````
```{r 2.b}
##2.b
data.frame(Predictors = c("FFMC", "DMC", "DC","ISI", "temp", "RH", "wind", "rain", "area"),
           range = c(max(forestfires$FFMC)-min(forestfires$FFMC), max(forestfires$DMC)-min(forestfires$DMC),
max(forestfires$DC)-min(forestfires$DC), max(forestfires$ISI)-min(forestfires$ISI), max(forestfires$temp)-min(forestfires$temp),
max(forestfires$RH)-min(forestfires$RH),max(forestfires$wind)-min(forestfires$wind),max(forestfires$rain)-min(forestfires$rain),
max(forestfires$area)-min(forestfires$area)),
           mean = c(mean(forestfires$FFMC),mean(forestfires$DMC),mean(forestfires$DC),mean(forestfires$ISI),mean(forestfires$temp),mean(fores
tfires$RH),mean(forestfires$wind),mean(forestfires$rain),mean(forestfires$area)),
           standard_deviation = c(sd(forestfires$FFMC),sd(forestfires$DMC),sd(forestfires$DC),sd(forestfires$ISI),sd(forestfires$temp),sd(for
estfires$RH),sd(forestfires$wind),sd(forestfires$rain),sd(forestfires$area)))
```
````

Predictors <chr>	range <dbl>	mean <dbl>	standard_deviation <dbl>
FFMC	77.50	90.64468085	5.5201108
DMC	290.20	110.87234043	64.0464822
DC	852.70	547.94003868	248.0661917
ISI	56.10	9.02166344	4.5594772
temp	31.10	18.88916828	5.8066253
RH	85.00	44.28820116	16.3174692
wind	9.00	4.01760155	1.7916526
rain	6.40	0.02166344	0.2959591
area	1090.84	12.84729207	63.6558185

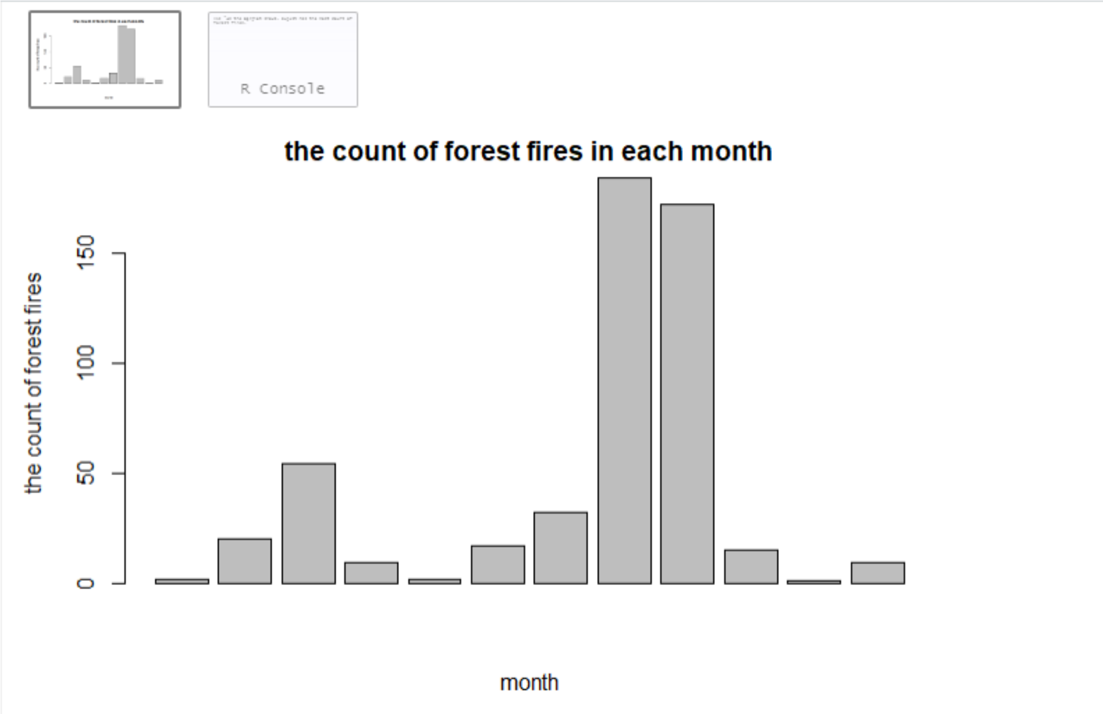
9 rows

2.c

```
```{r 2.c}
##2.c
dfremain <- forestfires[-c(20:70),]

data.frame(Predictors = c("FFMC", "DMC", "DC","ISI", "temp", "RH", "wind", "rain", "area"),
           range = c(max(dfremain$FFMC)-min(dfremain$FFMC), max(dfremain$DMC)-min(dfremain$DMC), max(dfremain$DC)-min(dfremain$DC),
max(dfremain$ISI)-min(dfremain$ISI), max(dfremain$temp)-min(dfremain$temp),
max(dfremain$RH)-min(dfremain$RH),max(dfremain$wind)-min(dfremain$wind),max(dfremain$rain)-min(dfremain$rain),
max(dfremain$area)-min(dfremain$area)),
           mean = c(mean(dfremain$FFMC),mean(dfremain$DMC),mean(dfremain$DC),mean(dfremain$ISI),mean(dfremain$temp),mean(dfremain$RH),mean(df
remain$wind),mean(dfremain$rain),mean(dfremain$area)),
           standard_deviation = c(sd(dfremain$FFMC),sd(dfremain$DMC),sd(dfremain$DC),sd(dfremain$ISI),sd(dfremain$temp),sd(dfremain$RH),sd(df
remain$wind),sd(dfremain$rain),sd(dfremain$area)))

```
```

Predictors <chr>	range <dbl>	mean <dbl>	standard_deviation <dbl>
FFMC	77.50	90.62188841	5.7429895
DMC	290.20	113.52167382	65.7845884
DC	852.70	548.04012876	249.1977150
ISI	22.70	8.98927039	4.1109312
temp	31.10	18.94163090	5.9027226
RH	85.00	44.59442060	16.5912495
wind	9.00	4.01244635	1.8179084
rain	6.40	0.02403433	0.3116754
area	1090.84	14.25332618	66.9058989

9 rows

2.d

```
```{r 2.d}
##2.d
barplot(
  c(sum(forestfires$month == "jan"),
     sum(forestfires$month == "feb"),
     sum(forestfires$month == "mar"),
     sum(forestfires$month == "apr"),
     sum(forestfires$month == "may"),
     sum(forestfires$month == "jun"),
     sum(forestfires$month == "jul"),
     sum(forestfires$month == "aug"),
     sum(forestfires$month == "sep"),
     sum(forestfires$month == "oct"),
     sum(forestfires$month == "nov"),
     sum(forestfires$month == "dec")), main = "the count of forest fires in each month", xlab = "month", ylab = "the count of forest fires")

  print("As the barplot shows, august has the most count of forest fires.")

```
```



As the bar plot shows, august has the most count of forest fires.

2.e

I firstly transferred all “month” and “day” data to numeric data.

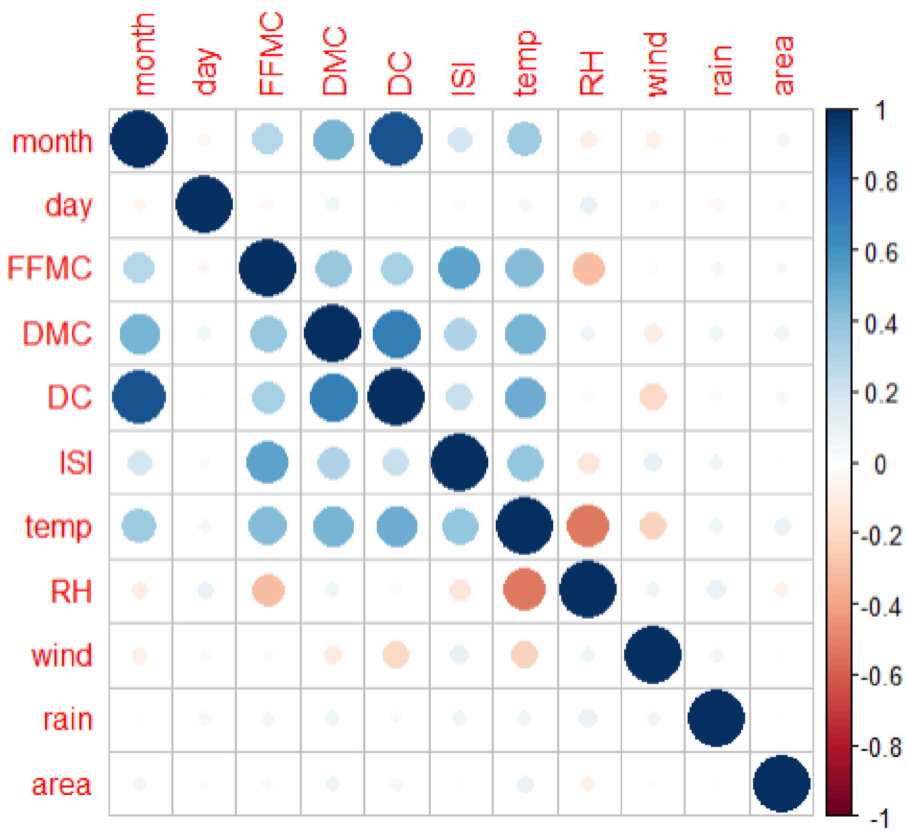
```{r}
##2.e
forestfires[forestfires == "jan"] <- 1
forestfires[forestfires == "feb"] <- 2
forestfires[forestfires == "mar"] <- 3
forestfires[forestfires == "apr"] <- 4
forestfires[forestfires == "may"] <- 5
forestfires[forestfires == "jun"] <- 6
forestfires[forestfires == "jul"] <- 7
forestfires[forestfires == "aug"] <- 8
forestfires[forestfires == "sep"] <- 9
forestfires[forestfires == "oct"] <- 10
forestfires[forestfires == "nov"] <- 11
forestfires[forestfires == "dec"] <- 12
forestfires[forestfires == "mon"] <- 1
forestfires[forestfires == "tue"] <- 2
forestfires[forestfires == "wed"] <- 3
forestfires[forestfires == "thu"] <- 4
forestfires[forestfires == "fri"] <- 5
forestfires[forestfires == "sat"] <- 6
forestfires[forestfires == "sun"] <- 7

```

```{r}
forestfires$month <- as.numeric(forestfires$month)
forestfires$day <- as.numeric(forestfires$day)
forestfires$RH <- as.numeric(forestfires$RH)
sapply(forestfires, class)
```

```
    month       day      FFMC       DMC        DC       ISI      temp        RH      wind      rain      area 
"numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" 
```

Then create correlation matrix for all relevant variables.

```
```{r 2.e}
forestfires.cor = cor(forestfires)
corrplot(forestfires.cor)
```
```



2.f

As above figure shown, temperature in degrees Celsius(temp) and relative humidity(RH) might be useful in predicting area since the area burned by the forest fire has higher positive correlation coefficient with temp, and higher negative correlation coefficient with RH, which means the burned area goes up with the higher temperature and lower relative humidity in some cases.