STATS_204_HW1

Qi Wang

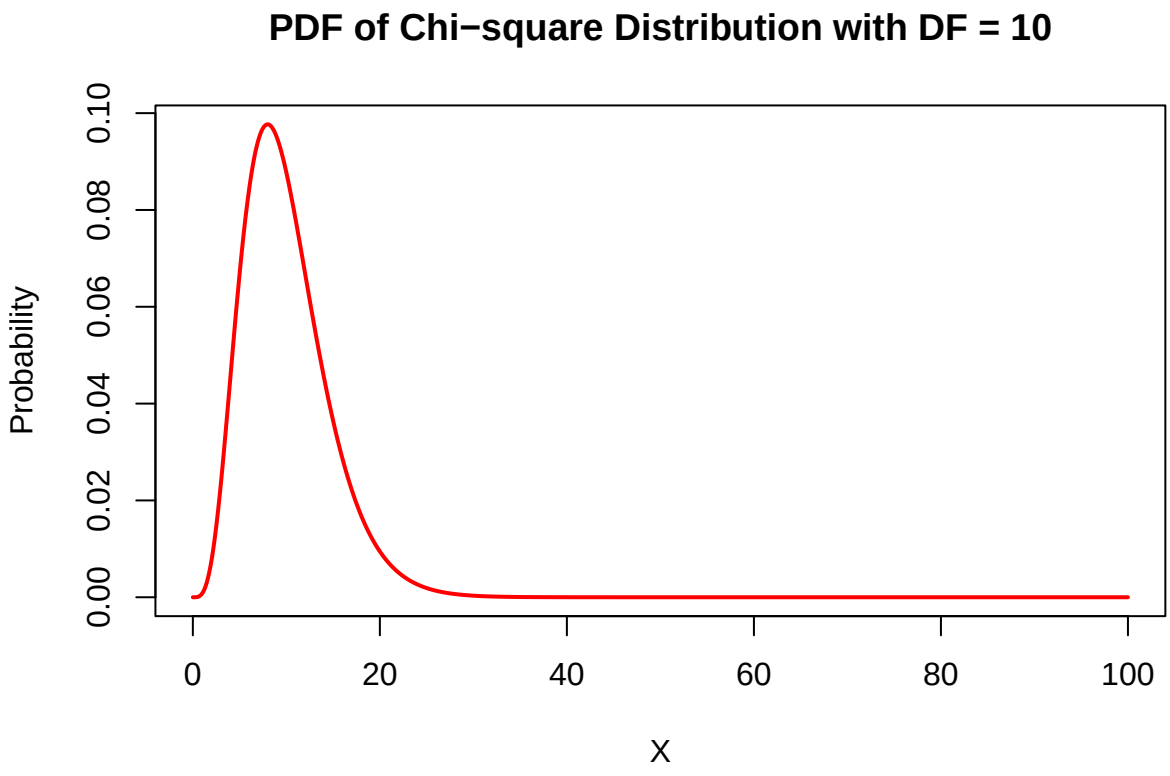
1. Use the qnorm function in R to find the quartiles (i.e., 25th, 50th and 75th percentiles) of the normal distribution with mean 100 and standard deviation 10.

```
ans <- matrix(qnorm(c(0.25,0.5,0.75), mean = 100, sd = 10), 1, 3)
colnames(ans) <- c("0.25 quantile", "0.5 quantile", "0.75 quantile")
print(ans)
```

```
##      0.25 quantile 0.5 quantile 0.75 quantile
## [1,]       93.2551          100      106.7449
```

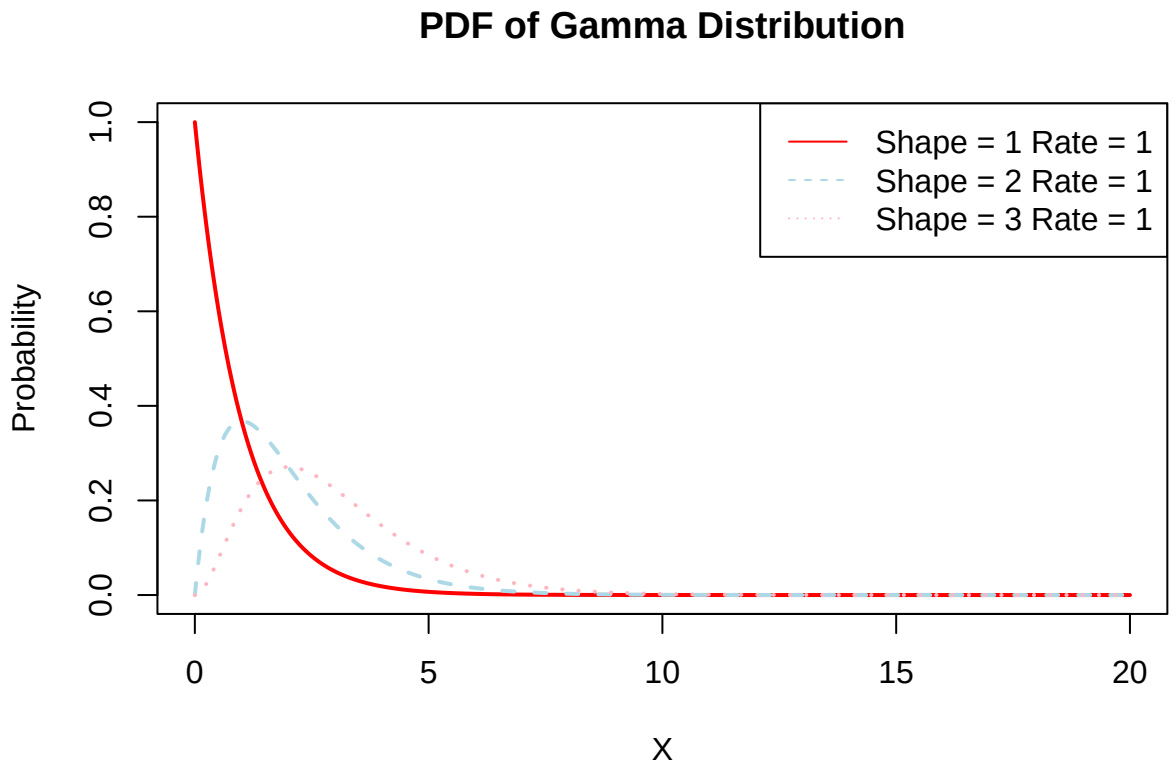
2.Use the curve function in R to display the graph of a $\chi^2(10)$ (10 corresponds to the degrees of freedom). Use a range of 0 to 100 for the x-axis. The chi-square density function is dchisq.

```
curve(dchisq(x, df = 10), from = 0, to = 100, n = 1000, type = 'l', xlab = "X",
      ylab = "Probability", lwd = 2, col = 'red',
      main = "PDF of Chi-square Distribution with DF = 10")
```



3.(Gamma densities). Use the curve function to display the graph of the gamma density with shape parameter 1 and rate parameter 1. Then use the curve function with add=TRUE to display the graphs of the gamma density with shape parameter k and rate 1 for 2,3, all in the same graphics window. The gamma density function is dgamma. Consult the help file ?dgamma to see how to specify the parameters.

```
curve(dgamma(x,shape = 1, rate = 1), from = 0, to = 20, n = 1000, type = 'l',
      xlab = "X", ylab = "Probability", lwd = 2, col = 'red',
      main = "PDF of Gamma Distribution")
curve(dgamma(x,shape = 2, rate = 1), from = 0, to = 20, n = 1000, type = 'l',
      xlab = "X", ylab = "Probability", lwd = 2, col = 'lightblue', add = TRUE
      , lty = 2)
curve(dgamma(x,shape = 3, rate = 1), from = 0, to = 20, n = 1000, type = 'l',
      xlab = "X", ylab = "Probability", lwd = 2, col = 'lightpink', add = TRUE
      , lty = 3)
legend("topright", c("Shape = 1 Rate = 1","Shape = 2 Rate = 1",
       "Shape = 3 Rate = 1"), col =  c('red','lightblue','lightpink'),
       lty = c(1,2,3))
```



4. (Binomial CDF). Let X be the number of "ones" obtained in 12 rolls of a fair die. Then X has a Binomial(n = 12, p = 1/3) distribution. Compute a table of cumulative binomial probabilities (the CDF) for x=0,1,...,12 by two methods:

(1) using cumsum and the result of Exercise 1.4

(2) using the pbinom function. What is P (X > 7)?

```
p_1 <- vector()
for (i in 0:12) {
  p_1[i+1] <- choose(12, i) * (1/3)^i * (2/3)^(12-i)
}
print(cumsum(p_1))
```

```
##  [1] 0.007707347 0.053951426 0.181122646 0.393074678 0.631520714 0.822277544
##  [7] 0.933552360 0.981241568 0.996144445 0.999456196 0.999952958 0.999998118
## [13] 1.000000000
```

```
p_2 <- pbinom(0:12, size = 12, prob = 1/3)
print(p_2)
```

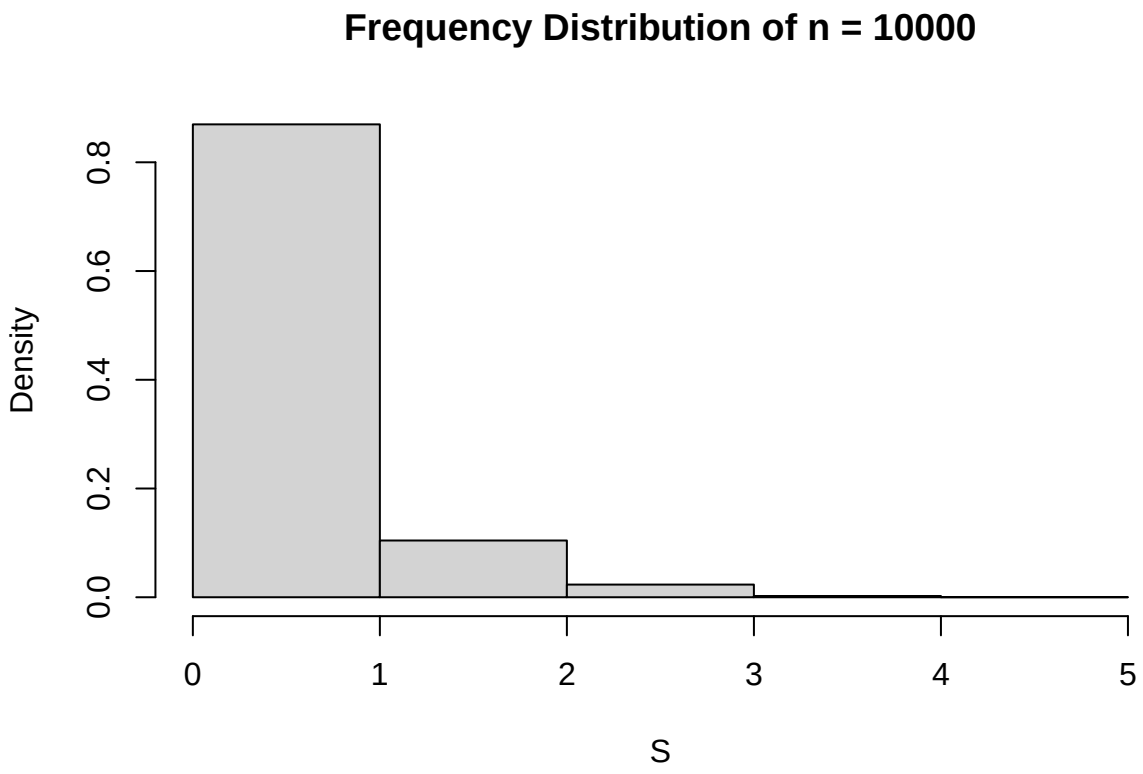
```
##  [1] 0.007707347 0.053951426 0.181122646 0.393074678 0.631520714 0.822277544
##  [7] 0.933552360 0.981241568 0.996144445 0.999456196 0.999952958 0.999998118
## [13] 1.000000000
```

```
print(1-pbinom(7, size = 12, prob = 1/3))
```

```
## [1] 0.01875843
```

5.(Simulated "horsekicks" data). The rpois function generates random observations from a Poisson distribution. In Example 1.3, we compared the deaths due to horsekicks to a Poisson distribution with mean $\lambda = 0.61$, and in Example 1.4 we simulated random Poisson($\lambda = 0.61$) data. Use the rpois function to simulate very large (n = 1000 and n = 10000) Poisson($\lambda = 0.61$) random samples. Find the frequency distribution, mean and variance for the sample. Compare the theoretical Poisson density with the sample proportions (see Example 1.4).

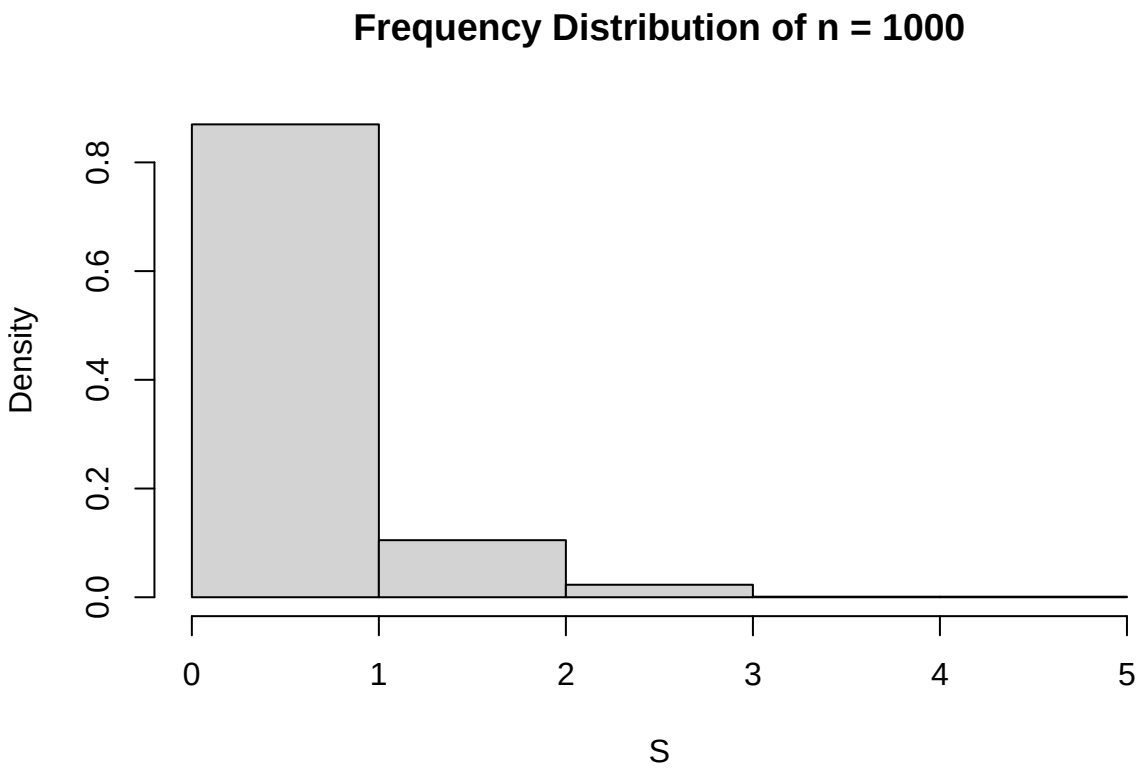
```
set.seed(0)
S <- rpois(10000, lambda = 0.61)
Freq <- table(S)/10000
Real <- dpois(0:5, lambda = 0.61)
RES <- cbind(Freq, Real)
rownames(RES) <- c(0, 1, 2, 3, 4, 5)
colnames(RES) <- c("Simulation", "Calculated by PMF")
mu <- mean(S)
sigma <- var(S)
hist(S, freq = FALSE, breaks = 0:5, main = "Frequency Distribution of n = 10000")
```




```
print(RES)
```

```
##   Simulation Calculated by PMF
## 0     0.5465      0.5433508691
## 1     0.3233      0.3314440301
## 2     0.1044      0.1010904292
## 3     0.0233      0.0205550539
## 4     0.0023      0.0031346457
## 5     0.0002      0.0003824268
```

```
set.seed(1)
S <- rpois(1000, lambda = 0.61)
Freq <- table(S)/1000
Real <- dpois(0:5, lambda = 0.61)
RES <- cbind(Freq, Real)
rownames(RES) <- c(0, 1, 2, 3, 4, 5)
colnames(RES) <- c("Simulation", "Calculated by PMF")
mu <- mean(S)
sigma <- var(S)
hist(S, freq = FALSE, breaks = 0:5, main = "Frequency Distribution of n = 1000")
```




```
print(RES)
```

```
##   Simulation Calculated by PMF
## 0      0.564      0.5433508691
## 1      0.306      0.3314440301
## 2      0.105      0.1010904292
## 3      0.023      0.0205550539
## 4      0.001      0.0031346457
## 5      0.001      0.0003824268
```

We can see that the simulation's frequency distribution is not so far from our real distribution. As the size of sampling increases, the accuracy will be improved. Here is the mean and variance of the simulated Poission distribution:

Mean:

```
print(mu)
```

```
## [1] 0.594
```

Variance:

```
print(sigma)
```

```
## [1] 0.6217858
```

6.(horsekicks, continued). Refer to Example 1.3. Using the ppois function, compute the cumulative distribution function (CDF) for the Poisson distribution with mean $\lambda = 0.61$, for the values 0 to 4. Compare these probabilities with the empirical CDF. The empirical CDF is the cumulative sum of the sample proportions p, which is easily computed using the cumsum func- tion. Combine the values of 0:4, the CDF, and the empirical CDF in a matrix to display these results in a single table.

```
cdf_emp  <- cumsum(Freq)
cdf_real <- ppois(0:4, lambda = 0.61)
cdf_both <- cbind(cdf_emp[1:5], cdf_real)
colnames(cdf_both) <- c("Empirical","Real")
print(cdf_both)
```

```
##   Empirical      Real
## 0     0.564 0.5433509
## 1     0.870 0.8747949
## 2     0.975 0.9758853
## 3     0.998 0.9964404
## 4     0.999 0.9995750
```

7.(Custom standard deviation function). Write a function sd.n similar to the function var.n in Example 1.5 that will return the estimate $\hat{\sigma}$ (the square root of $\hat{\sigma}^2$). Try this function on the temperature data of Example 1.1.

```
sd.n <- function(x){
  v <- var(x)
  n <- length(x)
  s_d <- sqrt(v*(n-1)/n)
  return(s_d)
}
temps = c(51.9, 51.8, 51.9, 53)
sd.n(temps)
```

```
## [1] 0.4924429
```

8.(Euclidean norm function). Write a function norm that will compute the Euclidean norm of a numeric vector. The Euclidean norm of a vector x = (x1,…,xn) is

$$||x|| = \sqrt{\Sigma_{i=1}^{n} x_i^2}$$

Use vectorized operations to compute the sum. Try this function on the vectors (0,0,0,1) and (2,5,2,4) to check that your function result is correct.

```
NORM <- function(x){
  eu_norm <- sqrt(sum(x^2))
  return(eu_norm)
}
NORM(c(0,0,0,1))
```

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

```
NORM(c(2,5,2,4))
```

```
## [1] 7
```

9.(Numerical integration). Use the curve function to display the graph of the function

$$f(x) = e^{-x^2}/(1+x^2)$$

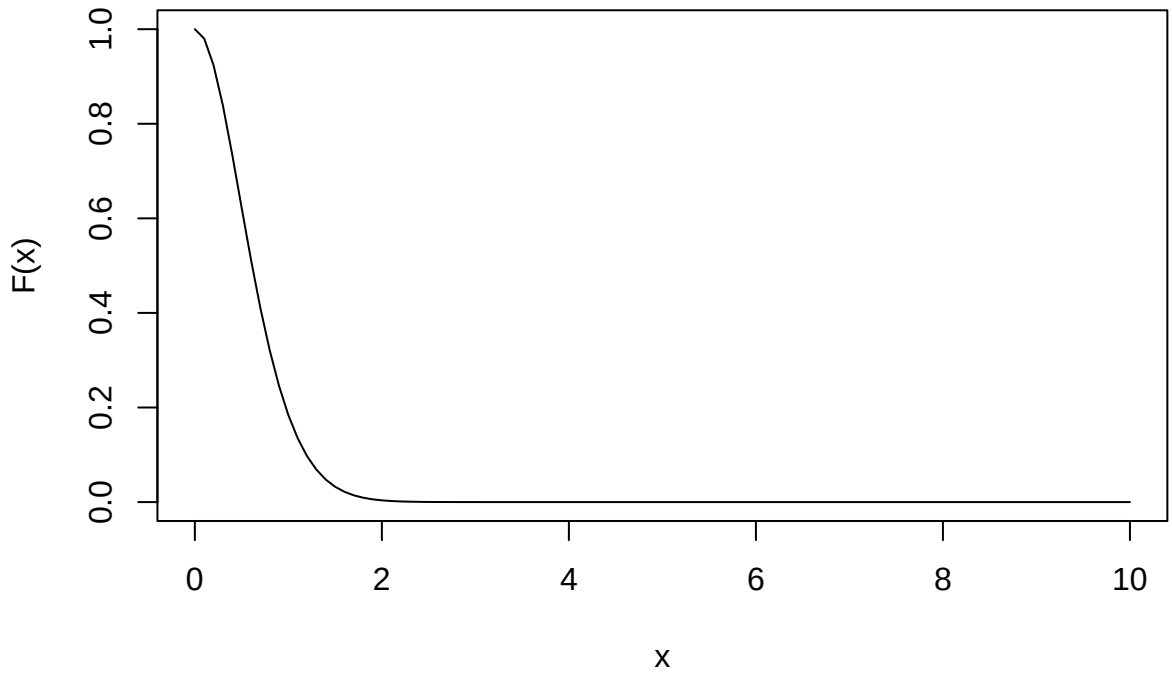
on the interval $0 \leq x \leq 10$. Then use the integrate function to compute the value of the integral:

$$\int_0^\infty \frac{e^{-x^2}}{1+x^2} dx$$

The upper limit at infinity is specified by upper=Inf in the integrate function.

```
F <- function(x){
  out <- exp(-x^2)/(1+x^2)
  return(out)
}

curve(F, from = 0, to = 10)
```



```
INT <- integrate(F, lower = 0, upper = Inf)
print(INT)
```

```
## 0.6716467 with absolute error < 8.3e-05
```

10.Construct a matrix with 10 rows and 2 columns, containing random standard normal data:

x = matrix(rnorm(20), 10, 2)

This is a random sample of 10 observations from a standard bivariate normal distribution. Use the apply function and your norm function from Exercise 1.10 to compute the Euclidean norms for each of these 10 observations.

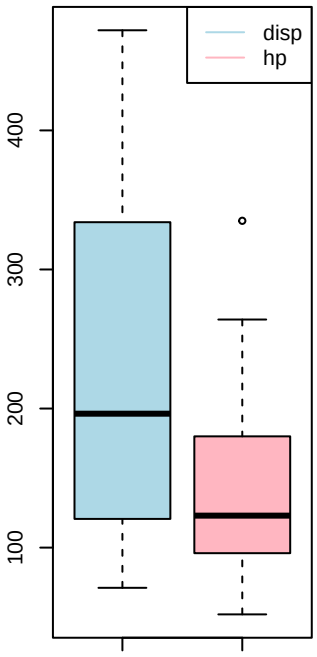
```
set.seed(0)
data <- matrix(rnorm(20), 10, 2)
apply(data, 1, NORM)
```

```
##  [1] 1.4758484 0.8630434 1.7565542 1.3049385 0.5113289 1.5939847 0.9622128
##  [8] 0.9393527 0.4357215 2.7044148
```

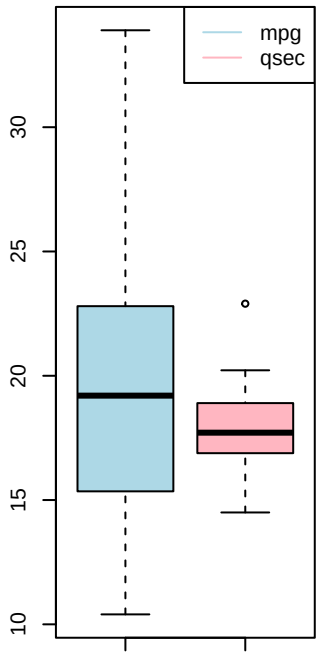
11.(mtcars data). Display the mtcars data included with R and read the documentation using ?mtcars. Display parallel boxplots of the quantitative variables. Display a pairs plot of the quantitative variables. Does the pairs plot reveal any possible relations between the variables?

PS: Since different variables have different range, I have put the variables with similar ranges together.

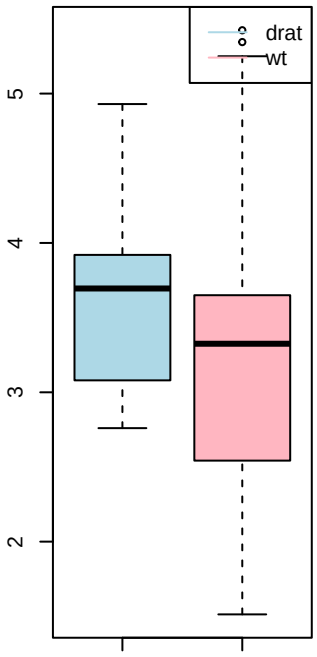
```
par(mfrow = c(1,3))
boxplot(mtcars$disp, mtcars$hp, xlab = "Displacement and Gross horsepower",
        col =c("lightblue", "lightpink"))
legend("topright", c("disp", "hp"),
       col = c("lightblue", "lightpink"), lty = c(1,1))
boxplot(mtcars$mpg, mtcars$qsec, xlab = "Miles/Gallon and 1/4 Mile Time",
        col =c("lightblue", "lightpink"))
legend("topright", c("mpg", "qsec"),
       col = c("lightblue", "lightpink"), lty = c(1,1))
boxplot(mtcars$drat, mtcars$wt, xlab = "Rear axle ratio and Weight",
        col =c("lightblue", "lightpink"))
legend("topright", c("drat", "wt"),
       col = c("lightblue", "lightpink"), lty = c(1,1))
```



Displacement and Gross horsepower

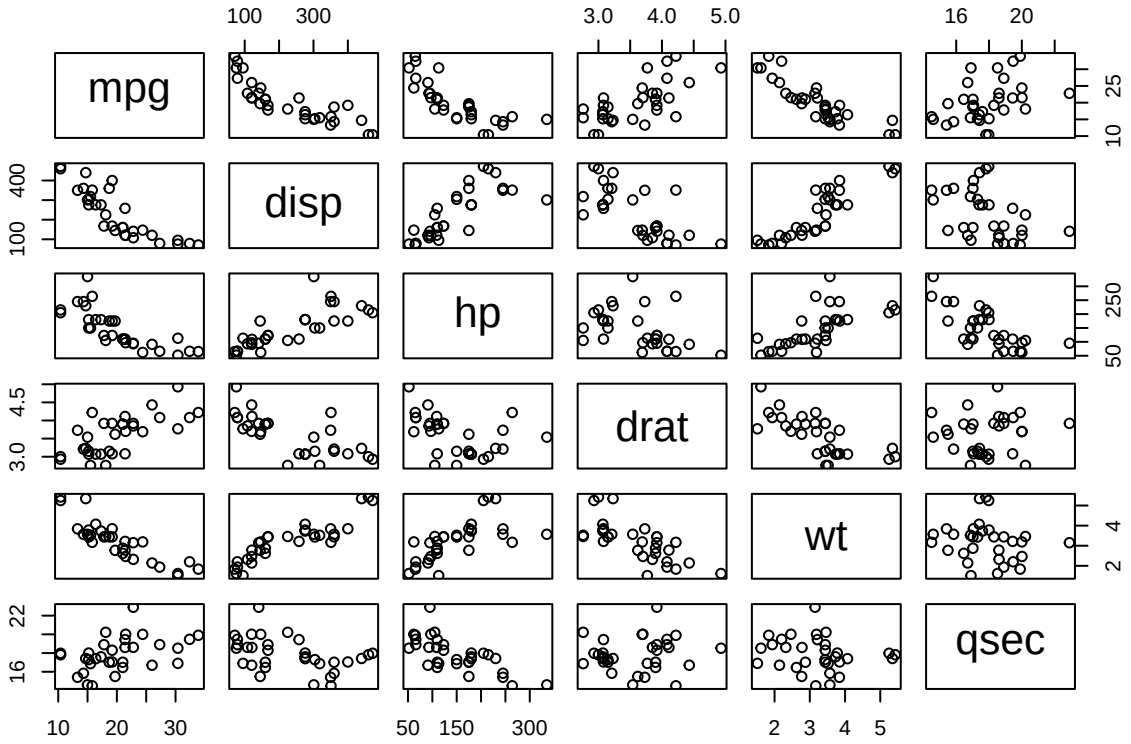


Miles/Gallon and 1/4 Mile Time



Rear axle ratio and Weight

```
Q_variables <- cbind(mtcars$mpg, mtcars$disp, mtcars$hp,
                     mtcars$drat, mtcars$wt, mtcars$qsec)
colnames(Q_variables) <- c("mpg", "disp", "hp", "drat", "wt", "qsec")
pairs(Q_variables)
```



From the pair plots above, we can see that: Miles per gallon is negatively related to the displacement, horsepower and weight. Horsepower has a positive relationship with the displacement and weight. There could be other relationships between them, however, we need hypothesis testing.

12.(mammals data). Refer to Example 2.7. Create a new variable r equal to the ratio of brain size over body size. Using the full mammals data set, order the mammals data by the ratio r. Which mammals have the largest ratios of brain size to body size? Which mammals have the smallest ratios? (Hint: use head and tail on the ordered data.)

```
library(MASS)
r <- mammals$brain / mammals$body

mammals_new <- cbind(mammals,r)

head(mammals_new[order(mammals_new$r, decreasing = TRUE),])
```

```
##                             body  brain        r
## Ground squirrel            0.101   4.00 39.60396
## Owl monkey                 0.480  15.50 32.29167
## Lesser short-tailed shrew  0.005   0.14 28.00000
## Rhesus monkey              6.800 179.00 26.32353
## Galago                     0.200   5.00 25.00000
## Little brown bat           0.010   0.25 25.00000
```

```
tail(mammals_new[order(mammals_new$r, decreasing = TRUE),])
```

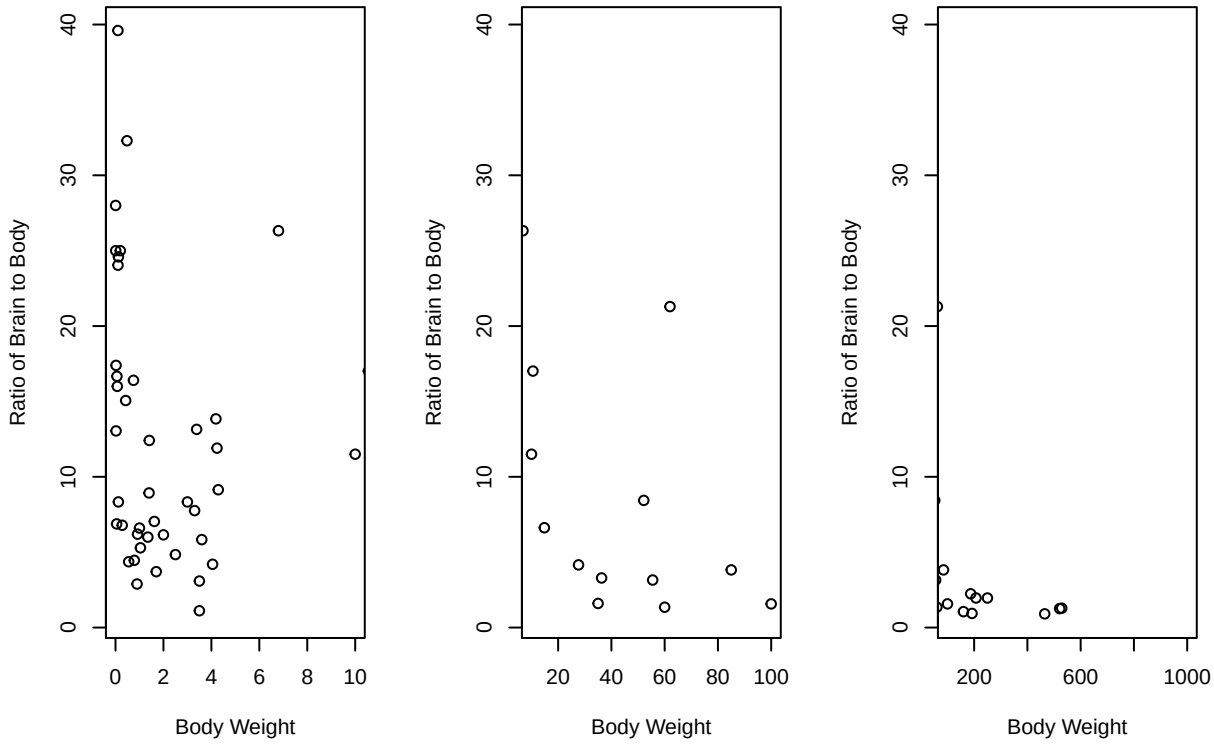
```
##                     body  brain         r
## Horse              521.0  655.0 1.2571977
## Water opossum        3.5    3.9 1.1142857
## Brazilian tapir    160.0  169.0 1.0562500
## Pig                192.0  180.0 0.9375000
## Cow                465.0  423.0 0.9096774
## African elephant 6654.0 5712.0 0.8584310
```

Therefore, "Ground squirrel" has the largest ratios of brain size to body size. And "African elephant" has the smallest one.

(mammals data, continued). Refer to Exercise 2.5. Construct a scatterplot of the ratio r = brain/body vs body size for the full mammals data set.

To make the scatterplot easier to observe, I truncated the body weight variable from the dataset.

```
par(mfrow = c(1,3))
plot(x = mammals_new$body, y = mammals_new$r, xlim = c(0,10),
     xlab = "Body Weight", ylab = "Ratio of Brain to Body")
plot(x = mammals_new$body, y = mammals_new$r, xlim = c(10,100),
     xlab = "Body Weight", ylab = "Ratio of Brain to Body")
plot(x = mammals_new$body, y = mammals_new$r, xlim = c(100,1000),
     xlab = "Body Weight", ylab = "Ratio of Brain to Body")
```



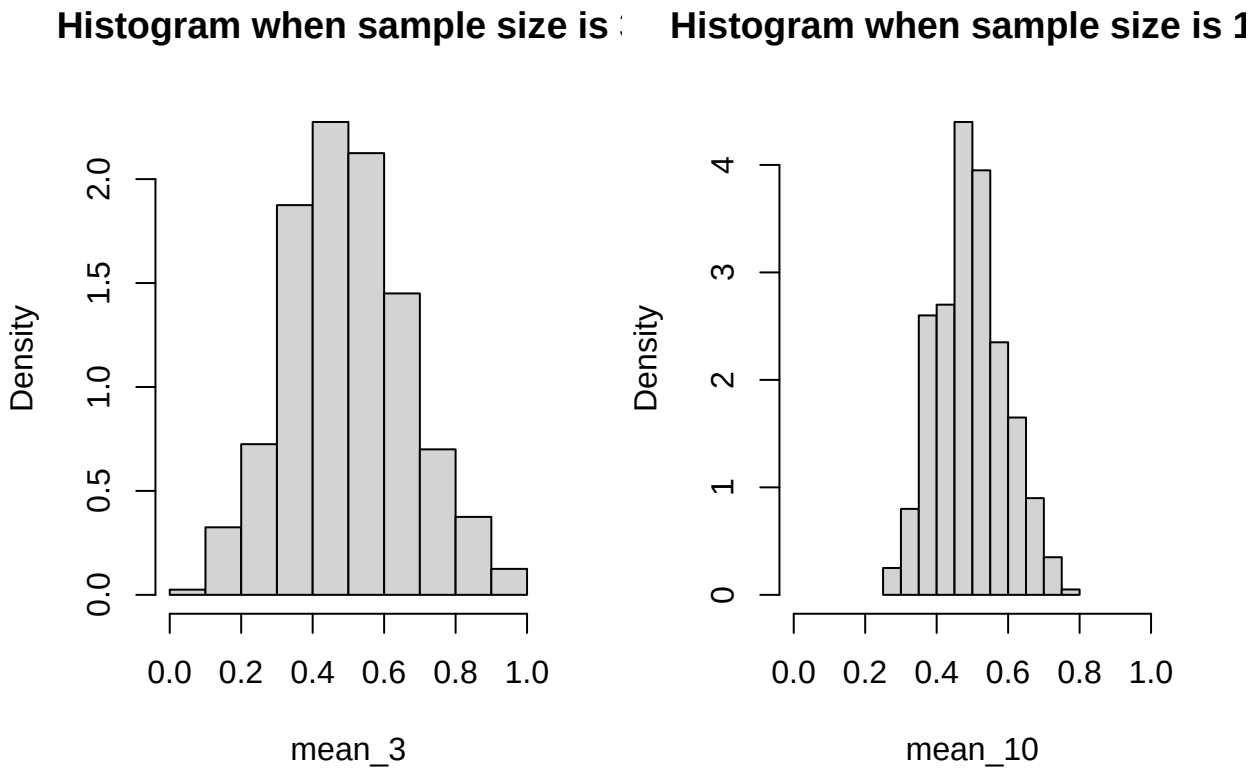
It could be observed from the scatterplot that as body weight increases, the ratio has a trend to become smaller and smaller. I guess it is because the size of brain is not ratio to the size of body.

13.(Central Limit Theorem with simulated data). Refer to Example 2.6, where we computed sample means for each row of the randu data frame. Repeat the analysis, but instead of randu, create a matrix of random numbers using runif.

```
set.seed(0)
uni_num_3 <- matrix(runif(400*3),400,3)
```

(Central Limit Theorem, continued). Refer to Example 2.6 and Exercise 2.7, where we computed sample means for each row of the data frame. Repeat the analysis in Exercise 2.7, but instead of sample size 3 generate a matrix that is 400 by 10 (sample size 10). Compare the histogram for sample size 3 and sample size 10. What does the Central Limit Theorem tell us about the distribution of the mean as sample size increases?

```
set.seed(1)
uni_num_10 <- matrix(runif(400*10),400,10)
mean_3 <- apply(uni_num_3, 1, mean)
mean_10 <- apply(uni_num_10, 1, mean)
par(mfrow = c(1,2))
hist(mean_3, main = "Histogram when sample size is 3", freq = FALSE)
hist(mean_10, main = "Histogram when sample size is 10", freq = FALSE, xlim = c(0,1))
```



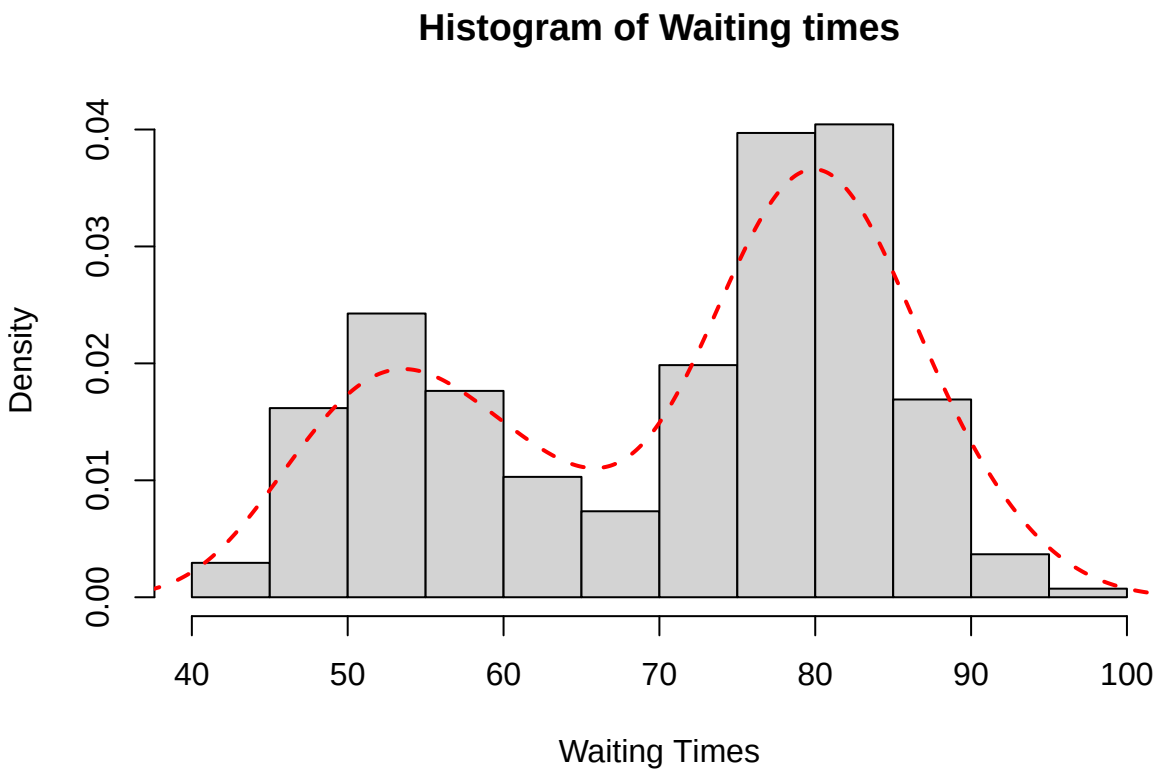
From the chart we can see that both of the mean of the samples follow a normal distribution. When the sample size rises from 3 to 10, the distribution of the mean becomes more centered, which means that it has smaller variance but still the same mean.

14.(“Old Faithful” histogram). Use hist to display a probability histogram of the waiting times for the Old Faithful geyser in the faithful data set (see Example A.3). (Use the argument prob=TRUE or freq=FALSE.)

```
hist(faithful$waiting, main = "Histogram of Waiting times",
     xlab = "Waiting Times", prob = TRUE)
```

(“Old Faithful” density estimate). Use hist to display a probability histogram of the waiting times for the Old Faithful geyser in the faithful data set (see Example A.3) and add a density estimate using lines.

```
hist(faithful$waiting, main = "Histogram of Waiting times",
     xlab = "Waiting Times", prob = TRUE)
lines(density(faithful$waiting),col = 'red', lwd = 2, lty = 2)
```



15. Question Omitted

(a)

```
library(ISLR)
write.csv(College, file = "college.csv")
college <- read.csv("college.csv")
```

(b)

```
#We need lots of preparations for using the fix function like XQuartz package and some other R tools.
rownames(college) <- college[,1]
#fix(college)
```

I have tried how to use fix() function to make adjustment of the data, it is a amazing tool inside R.

(c.i)

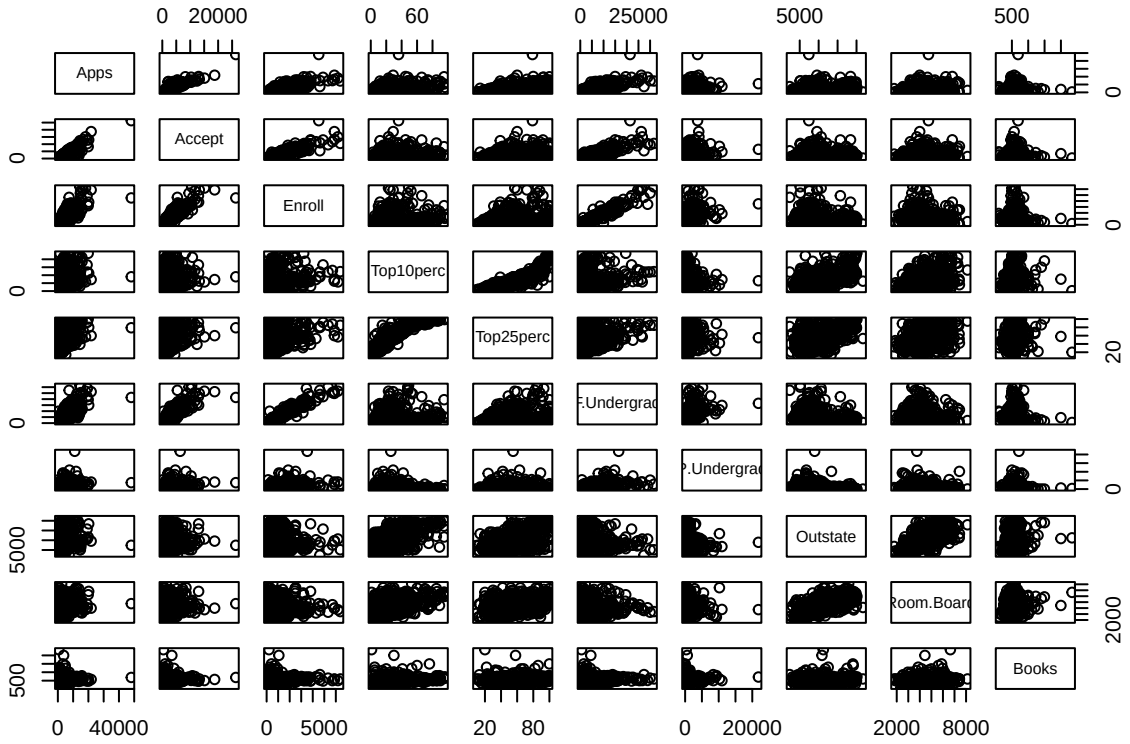
```
summary(college)
```

```
##        X                Private                 Apps           Accept
```

```
##  Length:777         Length:777         Min.   :   81   Min.   :   72
##  Class :character   Class :character   1st Qu.:  776   1st Qu.:  604
##  Mode  :character   Mode  :character   Median : 1558   Median : 1110
##                                        Mean   : 3002   Mean   : 2019
##                                        3rd Qu.: 3624   3rd Qu.: 2424
##                                        Max.   :48094   Max.   :26330
##      Enroll       Top10perc       Top25perc      F.Undergrad
##  Min.   :  35   Min.   : 1.00   Min.   :  9.0   Min.   :  139
##  1st Qu.: 242   1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992
##  Median : 434   Median :23.00   Median : 54.0   Median : 1707
##  Mean   : 780   Mean   :27.56   Mean   : 55.8   Mean   : 3700
##  3rd Qu.: 902   3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005
##  Max.   :6392   Max.   :96.00   Max.   :100.0   Max.   :31643
##   P.Undergrad         Outstate       Room.Board       Books
##  Min.   :    1.0   Min.   : 2340   Min.   :1780   Min.   :  96.0
##  1st Qu.:   95.0   1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0
##  Median :  353.0   Median : 9990   Median :4200   Median : 500.0
##  Mean   :  855.3   Mean   :10441   Mean   :4358   Mean   : 549.4
##  3rd Qu.:  967.0   3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0
##  Max.   :21836.0   Max.   :21700   Max.   :8124   Max.   :2340.0
##     Personal          PhD            Terminal       S.F.Ratio
##  Min.   : 250   Min.   :  8.00   Min.   : 24.0   Min.   : 2.50
##  1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50
##  Median :1200   Median : 75.00   Median : 82.0   Median :13.60
##  Mean   :1341   Mean   : 72.66   Mean   : 79.7   Mean   :14.09
##  3rd Qu.:1700   3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50
##  Max.   :6800   Max.   :103.00   Max.   :100.0   Max.   :39.80
##   perc.alumni        Expend        Grad.Rate
##  Min.   : 0.00   Min.   : 3186   Min.   : 10.00
##  1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00
##  Median :21.00   Median : 8377   Median : 65.00
##  Mean   :22.74   Mean   : 9660   Mean   : 65.46
##  3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00
##  Max.   :64.00   Max.   :56233   Max.   :118.00
```

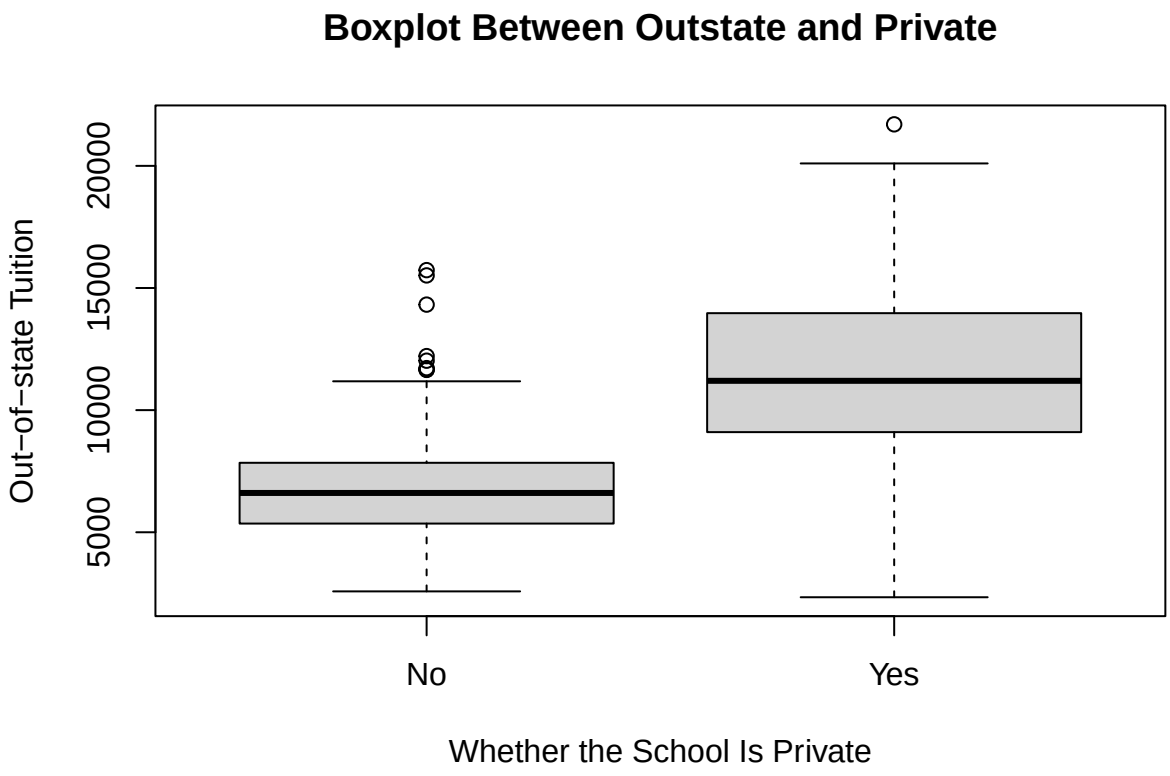
(c.ii)

```
pairs(college[,3:12])
```



(c.iii)

```
boxplot(college$Outstate ~ college$Private,
        main = "Boxplot Between Outstate and Private",
        xlab = "Whether the School Is Private",
        ylab = "Out-of-state Tuition"
        )
```



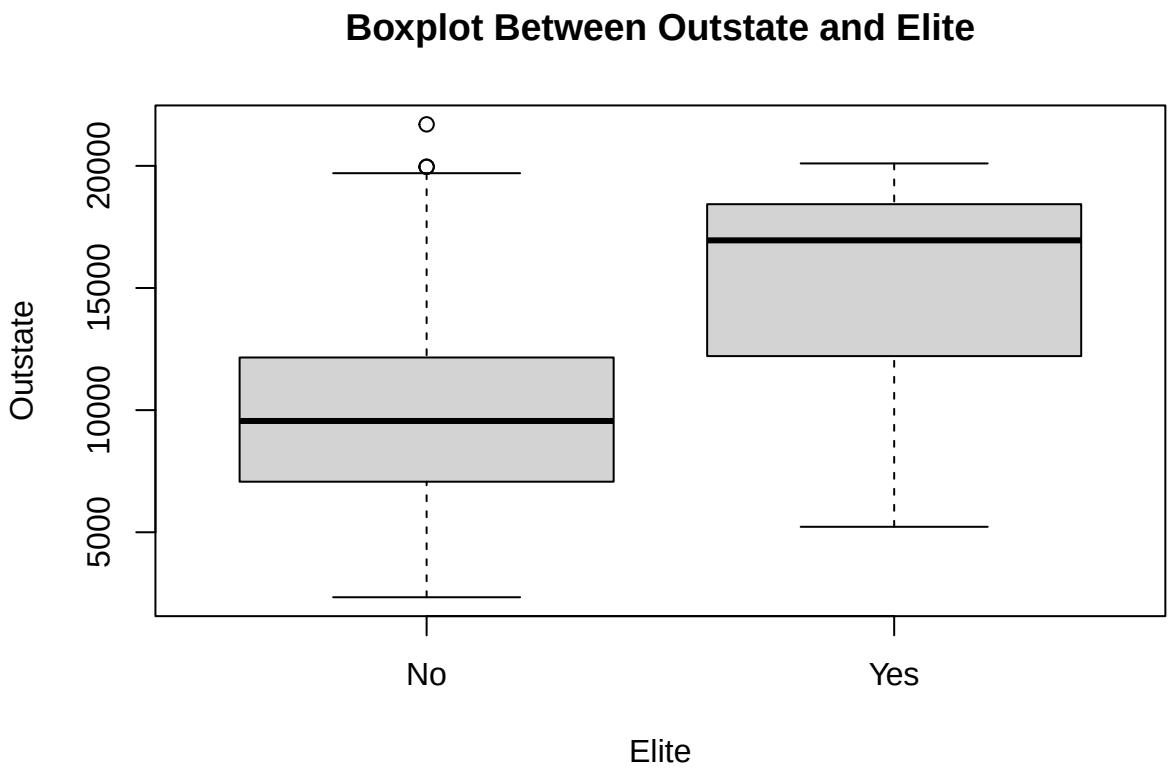
(c.iv)

```
Elite <- rep("No",nrow(college))
Elite[college$Top10perc >50]="Yes"
Elite <- as.factor(Elite)
college <- data.frame(college,Elite)
summary(Elite)
```

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

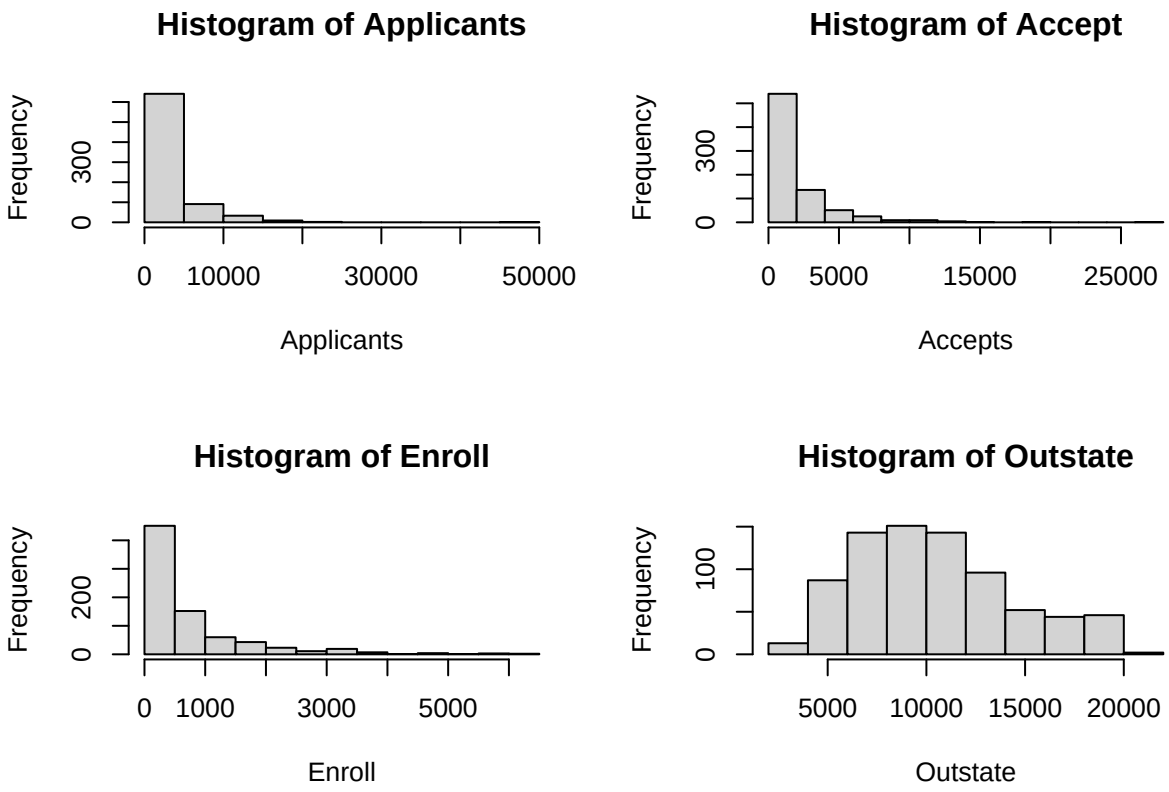
There are 78 elite universities.

```
boxplot(college$Outstate ~ college$Elite,
        main = "Boxplot Between Outstate and Elite",
        xlab = "Elite", ylab = "Outstate")
```



(c.v)

```
par(mfrow = c(2,2))
hist(college$Apps, main = "Histogram of Applicants", xlab = "Applicants")
hist(college$Accept, main = "Histogram of Accept", xlab = "Accepts")
hist(college$Enroll, main = "Histogram of Enroll", xlab = "Enroll")
hist(college$Outstate, main = "Histogram of Outstate", xlab = "Outstate")
```



(c.vi)

1. From the pair plots we can find that schools with more applicants will have more probability to have a larger number of accepted applicants and enrolled students.
2. If the school is a private school, it has relatively more outstate tuition compared with those not.
3. If the school is elite school, is has relatively more ourstate tuition compared with those not.
4. Schools with larger percent of students from top 10% of H.S. class tend to have larger percent of students from top 25% of H.S. class.

(This is an obvious conclusion, however, when we are making data analysis, this should be considered carefully because this may cause the dependece of these two variables.)

16. Question Omitted

(a)

```
library(MASS)
nrow(Boston)
```

```
## [1] 506
```

```
ncol(Boston)
```

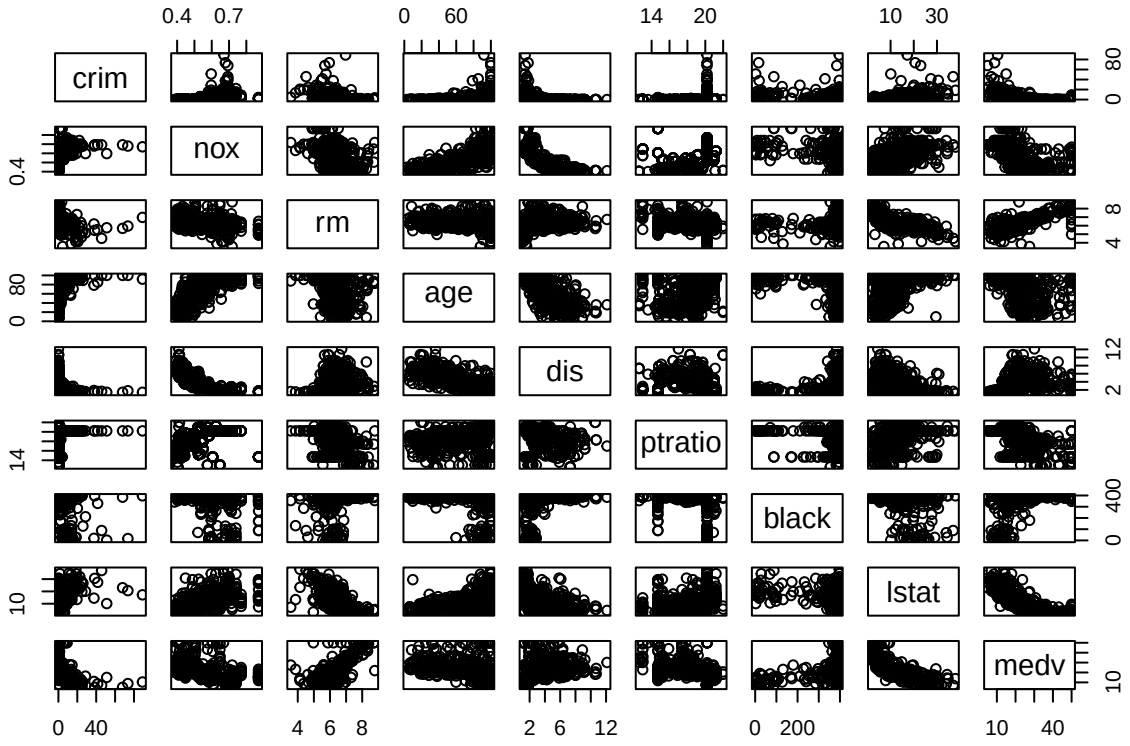
```
## [1] 14
```

```
# I can also use dim(Boston).
```

We have 506 rows and 14 columns. Each row means one observation, i.e., a piece of data. Each column indicates a variabl(predictor).

(b)

```
pairs(Boston[,c(1,5:8,11:14)])
```



Findings:

1. As the weighted mean of distances to five Boston employment centers increases, the nitrogen oxides has a decreasing trend, i.e. they are negatively related.
2. Areas with great lower status of population tends to have smaller average number of rooms per dwelling.
3. Those who have larger median value of owner-occupied homes tends to have greater average number of rooms per dwelling.
4. Lower status of the population and median value of owner-occupied homes seems to be negatively related.

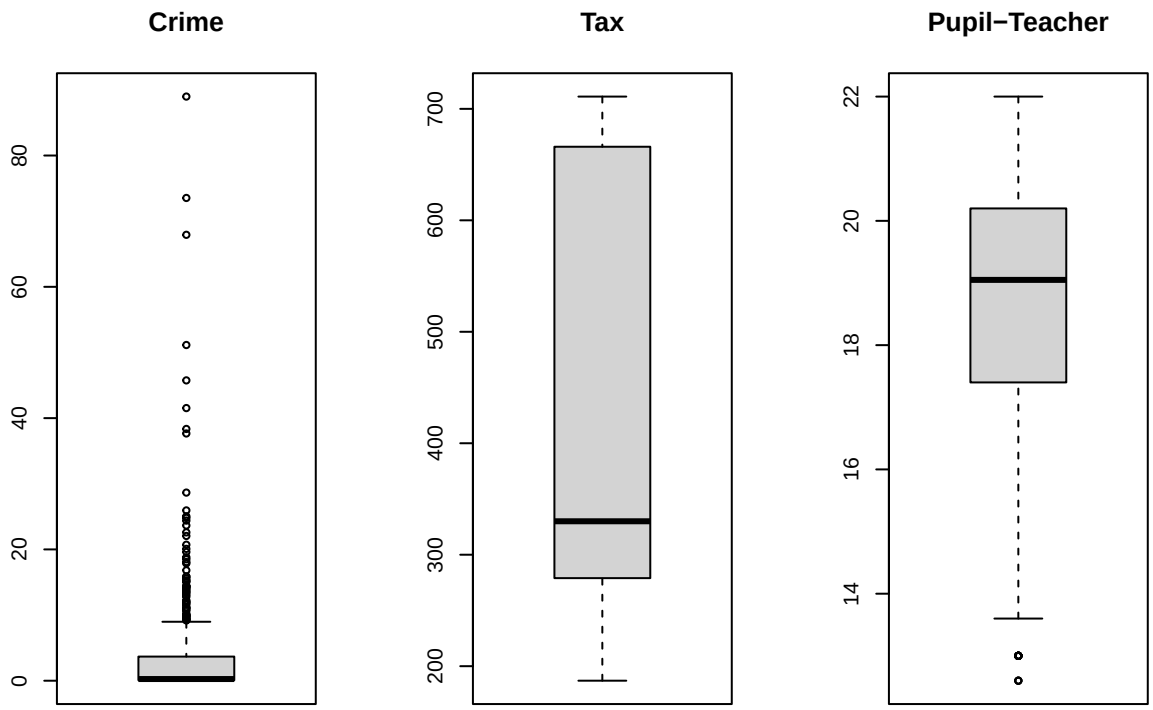
(c) The most obvious two predictors are "age" and "dis". As the proportion of owner-occupied units built prior to 1940 increases, the per capita crime seems to have a increasing trend. However, as the distance from the five Boston employment centers increases, the crime seems to have a decreasing trend.

(d)

```
res_Bos <- cbind(range(Boston$crim), range(Boston$tax), range(Boston$ptratio))
rownames(res_Bos) <- c("Min", "Max")
colnames(res_Bos) <- c("Crim", "Tax", "Pupil-Teacher Ratio")
print(res_Bos)
```

```
##         Crim Tax Pupil-Teacher Ratio
## Min  0.00632 187                12.6
## Max 88.97620 711                22.0
```

```
par(mfrow = c(1,3))
boxplot(Boston$crim, main = "Crime")
boxplot(Boston$tax, main = "Tax")
boxplot(Boston$ptratio, main = "Pupil-Teacher")
```



Therefore, we can say some suburbs have particular high crime rates. However, the other two variables seems not so widely spread. Let me find them:

```
head(Boston[order(Boston$crim, decreasing = TRUE),])
```

```
##        crim zn indus chas   nox    rm   age    dis rad tax ptratio  black lstat
## 381 88.9762  0  18.1    0 0.671 6.968  91.9 1.4165  24 666    20.2 396.90 17.21
## 419 73.5341  0  18.1    0 0.679 5.957 100.0 1.8026  24 666    20.2  16.45 20.62
## 406 67.9208  0  18.1    0 0.693 5.683 100.0 1.4254  24 666    20.2 384.97 22.98
## 411 51.1358  0  18.1    0 0.597 5.757 100.0 1.4130  24 666    20.2   2.60 10.11
## 415 45.7461  0  18.1    0 0.693 4.519 100.0 1.6582  24 666    20.2  88.27 36.98
## 405 41.5292  0  18.1    0 0.693 5.531  85.4 1.6074  24 666    20.2 329.46 27.38
##     medv
## 381 10.4
## 419  8.8
## 406  5.0
## 411 15.0
## 415  7.0
## 405  8.5
```

```
head(Boston[order(Boston$tax, decreasing = TRUE),])
```

```
##        crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 489 0.15086  0 27.74    0 0.609 5.454 92.7 1.8209   4 711    20.1 395.09 18.06
## 490 0.18337  0 27.74    0 0.609 5.414 98.3 1.7554   4 711    20.1 344.05 23.97
## 491 0.20746  0 27.74    0 0.609 5.093 98.0 1.8226   4 711    20.1 318.43 29.68
## 492 0.10574  0 27.74    0 0.609 5.983 98.8 1.8681   4 711    20.1 390.11 18.07
## 493 0.11132  0 27.74    0 0.609 5.983 83.5 2.1099   4 711    20.1 396.90 13.35
## 357 8.98296  0 18.10    1 0.770 6.212 97.4 2.1222  24 666    20.2 377.73 17.60
##     medv
## 489 15.2
## 490  7.0
## 491  8.1
## 492 13.6
## 493 20.1
## 357 17.8
```

```
head(Boston[order(Boston$ptratio, decreasing = TRUE),])
```

```
##        crim zn indus chas   nox    rm  age     dis rad tax ptratio  black lstat
## 355 0.04301 80  1.91    0 0.413 5.663 21.9 10.5857   4 334    22.0 382.80  8.05
## 356 0.10659 80  1.91    0 0.413 5.936 19.5 10.5857   4 334    22.0 376.04  5.57
## 128 0.25915  0 21.89    0 0.624 5.693 96.0  1.7883   4 437    21.2 392.11 17.19
## 129 0.32543  0 21.89    0 0.624 6.431 98.8  1.8125   4 437    21.2 396.90 15.39
## 130 0.88125  0 21.89    0 0.624 5.637 94.7  1.9799   4 437    21.2 396.90 18.34
## 131 0.34006  0 21.89    0 0.624 6.458 98.9  2.1185   4 437    21.2 395.04 12.60
##     medv
## 355 18.2
## 356 20.6
## 128 16.2
## 129 18.0
## 130 14.3
## 131 19.2
```

Here we find those suburb that have exremely high crime rates, high tax rates and high pupil-teacher ratio rates.

(e)

```
sum(Boston$chas)
```

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

Therefore, there are 35 suburbs in this data set bound the Charles river.

(f)

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

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

(g)

```
head(Boston[order(Boston$medv, decreasing = FALSE),])
```

```
##         crim zn indus chas   nox    rm   age    dis rad tax ptratio  black
## 399 38.35180  0 18.10    0 0.693 5.453 100.0 1.4896  24 666    20.2 396.90
## 406 67.92080  0 18.10    0 0.693 5.683 100.0 1.4254  24 666    20.2 384.97
## 401 25.04610  0 18.10    0 0.693 5.987 100.0 1.5888  24 666    20.2 396.90
## 400  9.91655  0 18.10    0 0.693 5.852  77.8 1.5004  24 666    20.2 338.16
## 415 45.74610  0 18.10    0 0.693 4.519 100.0 1.6582  24 666    20.2  88.27
## 490  0.18337  0 27.74    0 0.609 5.414  98.3 1.7554   4 711    20.1 344.05
##     lstat medv
## 399 30.59  5.0
## 406 22.98  5.0
## 401 26.77  5.6
## 400 29.97  6.3
## 415 36.98  7.0
## 490 23.97  7.0
```

We can see that 399 and 406 suburb have the smallest median value of owner-occupied homes. Take suburb # 399 as an example:

```
apply(as.matrix(Boston), 2, rank)[399,]
```

```
##    crim      zn   indus    chas     nox      rm     age     dis     rad     tax
##   500.0   186.5   383.5   236.0   427.5    39.0   485.0    29.0   440.5   435.5
## ptratio   black   lstat    medv
##   380.5   446.0   495.0     1.5
```

We can see the crime rate is very high since the larger rank means the larger number. Proportion of non-retail business acres per town, nitrogen oxides, index of accessibility to radial highways, proportion of blacks and lower status of the population also have the larger numbers.

(h)

```
nrow(Boston[which(Boston$rm > 7),])
```

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

```
nrow(Boston[which(Boston$rm > 8),])
```

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

In all, there are 64 suburbs average more than 7 rooms per dwelling, and 13 suburbs average more than 8 rooms per dwelling.

Let's see their other index's rank:

```
apply(as.matrix(Boston), 2, rank)[which(Boston$rm > 8),]
```

```
##     crim    zn indus chas   nox  rm   age   dis   rad   tax ptratio black lstat
## 98   174 186.5  49.0  236 109.0 496 245.0 276.0  32.5 109.0   188.0 446.0  38.0
## 164  351 186.5 464.5  489 348.5 502 377.5 140.0 250.0 323.5    34.5 207.5  18.0
## 205   18 503.5  45.5  236  45.0 494  71.5 378.5 137.5  39.5    34.5 234.0   6.0
## 225  267 186.5 172.5  236 196.5 499 258.0 234.5 362.5 210.5   135.5 179.0  36.0
## 226  298 186.5 172.5  236 196.5 505 282.5 234.5 362.5 210.5   135.5 162.0  50.0
## 227  284 186.5 172.5  236 196.5 495 307.5 254.5 362.5 210.5   135.5 199.0  14.0
## 233  306 186.5 172.5  236 205.5 501 235.0 302.0 362.5 210.5   135.5 185.0   4.0
## 234  273 186.5 172.5  236 205.5 497 222.5 286.0 362.5 210.5   135.5 153.0  32.5
## 254  281 414.5 146.5  236  67.5 498  10.0 495.5 342.0 251.5   262.0 446.0  22.0
## 258  309 395.0  86.5  236 392.5 504 309.0  76.0 250.0  84.5     9.5 221.0  67.0
## 263  296 395.0  86.5  236 392.5 503 351.0 158.0 250.0  84.5     9.5 195.0  91.0
## 268  307 395.0  86.5  236 303.5 500 208.0 175.0 250.0  84.5     9.5 176.0 141.5
## 365  376 186.5 383.5  489 466.5 506 280.5  94.5 440.5 435.5   380.5  90.0  72.5
##      medv
## 98  474.0
## 164 498.5
## 205 498.5
## 225 484.0
## 226 498.5
## 227 472.0
## 233 477.0
## 234 488.0
## 254 479.0
## 258 498.5
## 263 490.0
## 268 498.5
## 365 276.0
```

Firstly, most of them is not bounded by Charles River. Secondly, their median value of owner-occupied homes are high and average number of rooms per dwelling are very high, too.