STAT 435 HW1

Peiran Chen

3/31/2022

1.

a)

Taking a parametric approach will have following pros:
- Does not need a lot of data
- Simplifies the problem because it is generally much easier to estimate a set of parameters

and cons:
- The model we choose will usually not match the true unknown form of f, and if the chosen model is too far from the true f, then our estimate will be poor.

Taking a nonparametric approach will have following pros:
- Avoid unnecessary assumptions about the functional form of f will have the potential to accurately fit a wider range of possible shapes for f.

and cons:
- A very large number of observations is required in order to obtain an accurate estimate for f.

b)

For parametric approach, I would say when we have a small number of observations to work with, such as getting survey on people's blood pressure and hours of physical exercises they do each week. And we know that having more time to exercises will result in a lower blood pressure as a matter of fact. Hence we can make assumptions to f in this case to be a linear model:

$$blood\ pressure \approx \beta_0 + \beta_1 \times physical\ exercises$$

c)

For non-parametric approach, I would say when we have a lot of data to work with, we can use this method to do the same prediction as part b).

2.

a)

In this case, I would expect the inflexible methods to perform better. Since sample size is small, there won't be enough data for flexible methods such as deep learning and etc. Also, the number of predictors are large, hence it would be a good practice to use OLS so that we have more interpretability. Hence inflexible methods tends to perform better.

b)

In this case, I would expect the flexible method to perform better. Because we have a larger sample, a flexible method can take advantage of that and get more information out of it. The large n also greatly reduces the risk of over-fitting with flexible method.

c)

Inflexible methods have a lot of trouble picking up non-linear relationships, so we should prefer a flexible method.

d)

In this case, I would expect inflexible method to perform better. Higher variance will tend to introduce noise. The high variance of error terms means that the sample will have a lot of noise in the relationship. Therefore we should prefer an inflexible method that is less likely to put more weight on the noise hence gave us more accurate result.

3.

a)

It is a regression problem. And the goal is prediction, where $n = 50, p = 8$.

b)

It is a classification problem. And the goal is inference, where $n = 50, p = 6$.

4.

a) and b)

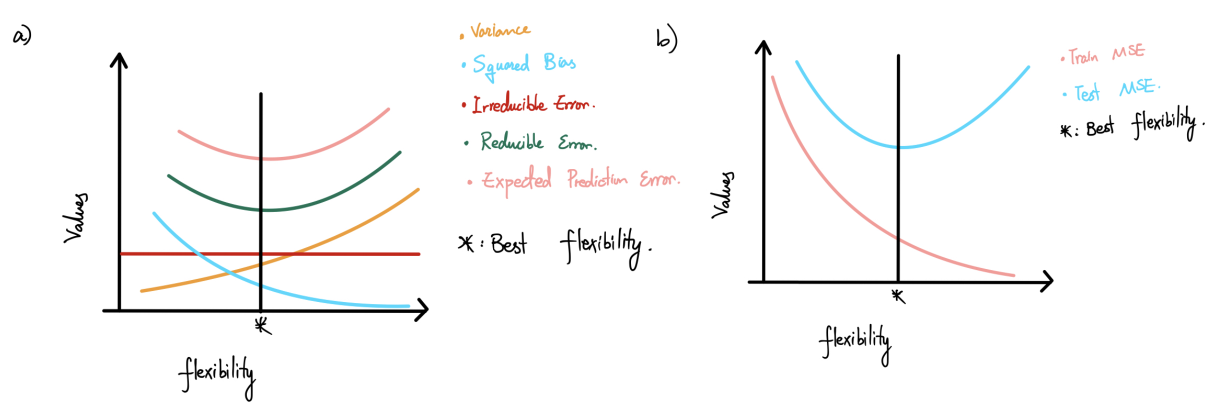


Figure 1: a) & b)

c)

$\hat{f}$ that has a smallest Bias and extremely high Variance would be when the true f is linear, but $\hat{f}$ is derived with a very flexible approach, which will result in over-fitting, and $\hat{f}$ put too much weight on the irreducible error hence introduce high Variance despite having smallest Bias.

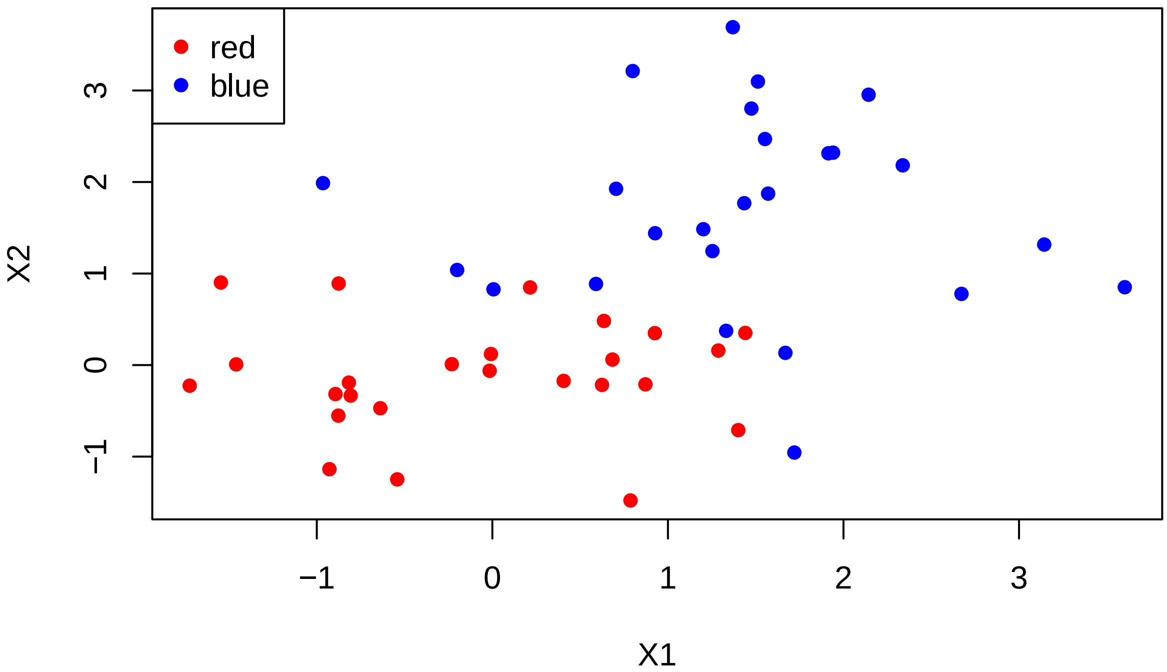
d)

One of the example of $\hat{f}$ in this case would be a linear fit(least squares). And the true f is not so linear, hence it will introduce high bias since we assume the data is a linear fit. And It tends to have low Variance since the result from a linear fit is consistent.

5.

a)

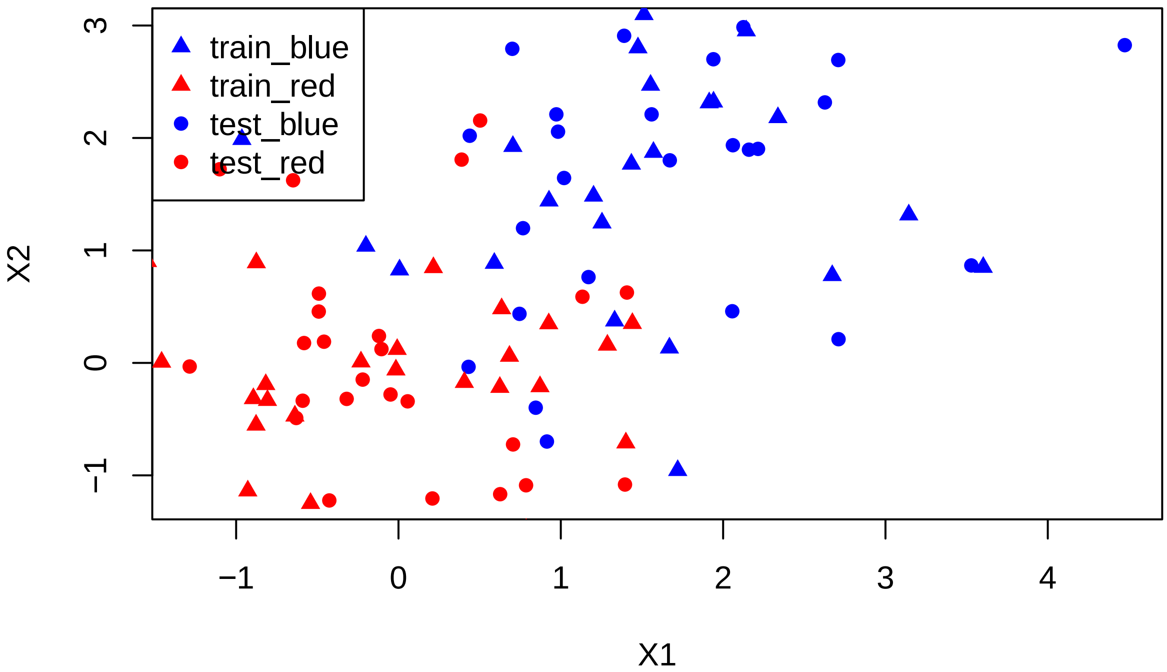
```
n <- 25

red <- data.frame(X1 = rnorm(n, 0, 1), X2 = rnorm(n, 0, 1)) %>%
  mutate(color = "red")
blue <- data.frame(X1 = rnorm(n, 1.5, 1), X2 = rnorm(n, 1.5, 1)) %>%
  mutate(color = "blue")

trainset <- rbind(red, blue)

plot(trainset$X1, trainset$X2,
     col = trainset$color,  xlab = "X1", ylab = "X2", pch = 16)
legend("topleft",
       c("red", "blue"),
       col = c("red", "blue"),
       pch = 16
       )
```



###

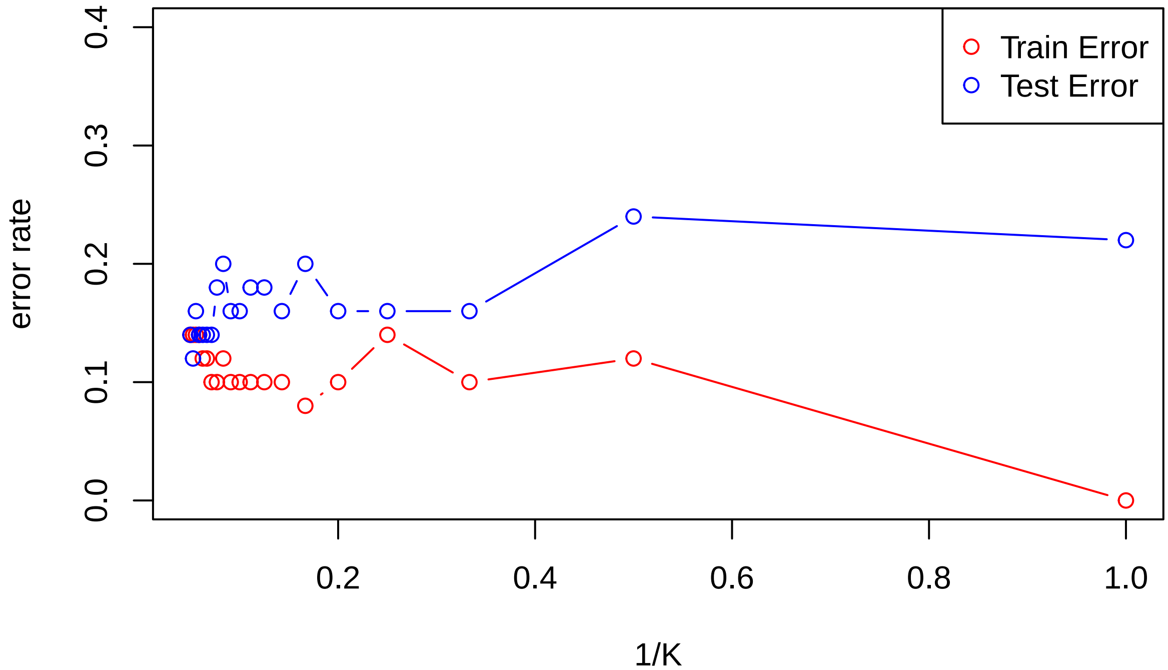
b.

```
# Generating test set
test_red <- data.frame(X1 = rnorm(n, 0, 1), X2 = rnorm(n, 0, 1)) %>%
  mutate(color = "red")
test_blue <- data.frame(X1 = rnorm(n, 1.5, 1), X2 = rnorm(n, 1.5, 1)) %>%
  mutate(color = "blue")

testset <- rbind(test_red, test_blue)

plot(testset$X1, testset$X2, col = testset$color, pch = 16, xlab = "X1", ylab = "X2")
points(trainset$X1, trainset$X2, col = trainset$color, pch = 17)
legend("topleft",
       c("train_blue", "train_red", "test_blue", "test_red"),
       col = c("blue", "red", "blue", "red"),
       pch = c(17,17,16,16)
       )
```



c)

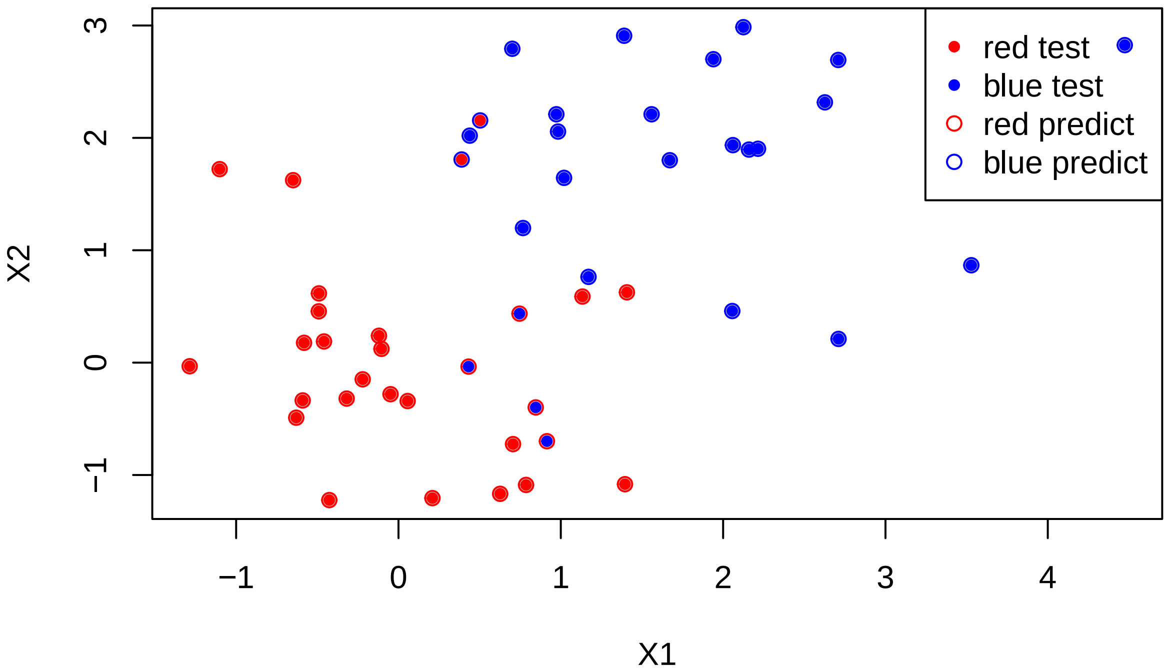
```
library(class)
train <- trainset[,1:2]
test <- testset[,1:2]

train_error <- rep(0, 20)
test_error <- rep(0, 20)
k <- c(1:20)


cl <- factor(c(rep("red", 25),rep("blue", 25)))
smallest_test_error <- data.frame(1, NA)
for (i in k) {
  predictions_train <- knn(train, train, cl, k = i, prob = FALSE)
  train_error[i] <- mean(predictions_train != cl)
  predictions_test <- knn(train, test, cl, k = i, prob = FALSE)
  test_error[i] <- mean(predictions_test != cl)
  if(test_error[i] < smallest_test_error[1]){
    smallest_test_error[1] = test_error[i]
    smallest_test_error[2] = i
  }
}
plot(1/k , train_error, col='red', type = 'b', ylim = c(0, 0.4), ylab = "error rate", xlab = "1/K")
points(1/k, test_error, col='blue', type = 'b')
legend("topright",
       col = c("red", "blue"),
       c("Train Error", "Test Error"),
       pch = c(1,1))
```



As k decreases, $1/k$ increases, which means the level of flexibility increases. And our result shows that training error can be reduced to zero, however, our test error tends to wobble and create a “U-Shape”. Hence, it's critical to choose a proper number of neighbors k.

d)

```
plot(testset$X1, testset$X2, col = testset$color, pch = 20, xlab = "X1", ylab = "X2")
points(testset$X1, testset$X2, col = paste0("",knn(train, test, cl, k = smallest_test_error[2], prob = F
legend("topright",
       c("red test", "blue test", "red predict", "blue predict"),
       col = c("red", "blue", "red", "blue"),
       pch = c(20,20,1,1))
```



###

e).

The formula gives us the intuition on how to calculate the Bayes Error Rate, it has nothing to do with knn, just a nature of how probabilities fluctuates. That is, we can see from mathematically proofing using PDF and see how many are overlaping. Let:

$$X_1 \sim N(0, 1)$$
$$X_2 \sim N(1.5, 1)$$

And let c denote the point of intersection where the PDF's meet. Then, the area can be given

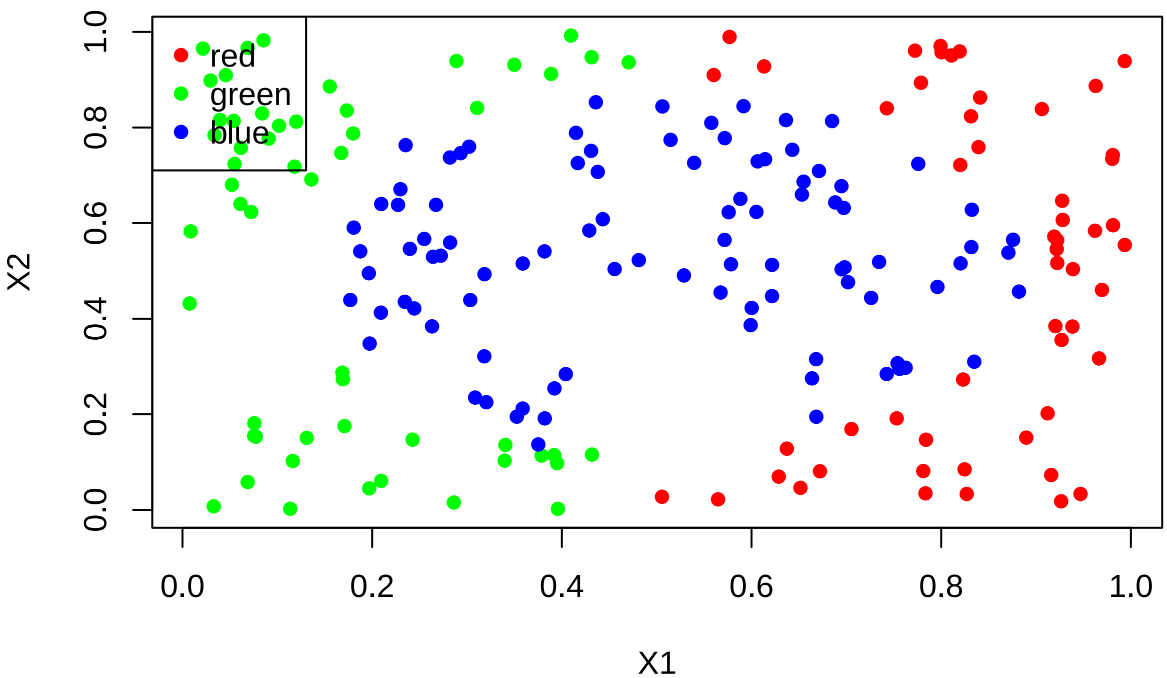
$$\begin{aligned}
P(X_1 > c) + P(X_2 < c) &= 1 - F_1(c) + F_2(c) \\
&= 1 - \frac{1}{2}\text{erf}(\frac{c-\mu_1}{\sqrt{2}\sigma_1}) + \frac{1}{2}\text{erf}(\frac{c-\mu_2}{\sqrt{2}\sigma_2}) \\
c &= \frac{\mu_2\sigma^2 - \sigma_2(\mu_1\sigma_2 + \sigma_1\sqrt{(\mu_1-\mu_2)^2 + 2(\sigma_1^2 - \sigma_2^2\log(\frac{\sigma_1}{\sigma_2}))}}{\sigma_1^2 - \sigma_2^2} \\
&\text{However, in this case, variance are the same} \\
c &= \frac{\mu_1 + \mu_2}{2} = 1.25
\end{aligned}$$

And Bayes Error Rate would be the volume of that area squared divided by the whole area of two normal distribution formed in $\mathbb{R}^3$, $ P(X_1> c) + P(X_2 < c) = 0.8025$. but I don't know how to do it :(. In this case, the Bayes Error Rate would be , and it relates to part c) and d) where the Test Error is always going to be greater than the Bayes Error Rate. Hence it is just like the irreducible error term, where it is the best we can do when fitting data.

6.

a)

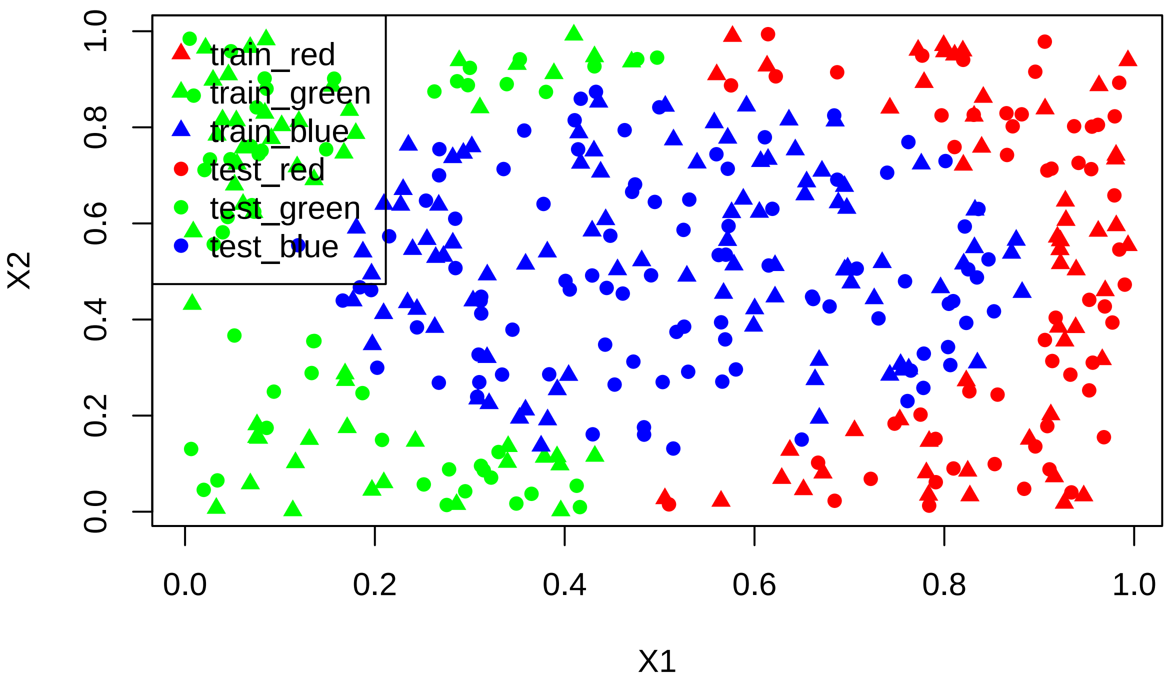
```
n <- 200

determine_color <- function(X1, X2){
  f = (X1 - 0.5)^2 + (X2 - 0.5)^2
  ifelse(f > 0.15, ifelse(X1 > 0.5, "red", "green"), "blue")
}

trainset <- tibble(X1 = runif(n,0,1),
                   X2 = runif(n,0,1),
                   color = determine_color(X1, X2))

plot(trainset$X1, trainset$X2,
     col = trainset$color,  xlab = "X1", ylab = "X2", pch = 16)
legend("topleft",
       c("red", "green", "blue"),
       col = c("red","green", "blue"),
       pch = 16
       )
```



b).

```
testset <- tibble(X1 = runif(n,0,1),
                  X2 = runif(n,0,1),
                  color = determine_color(X1, X2))
```

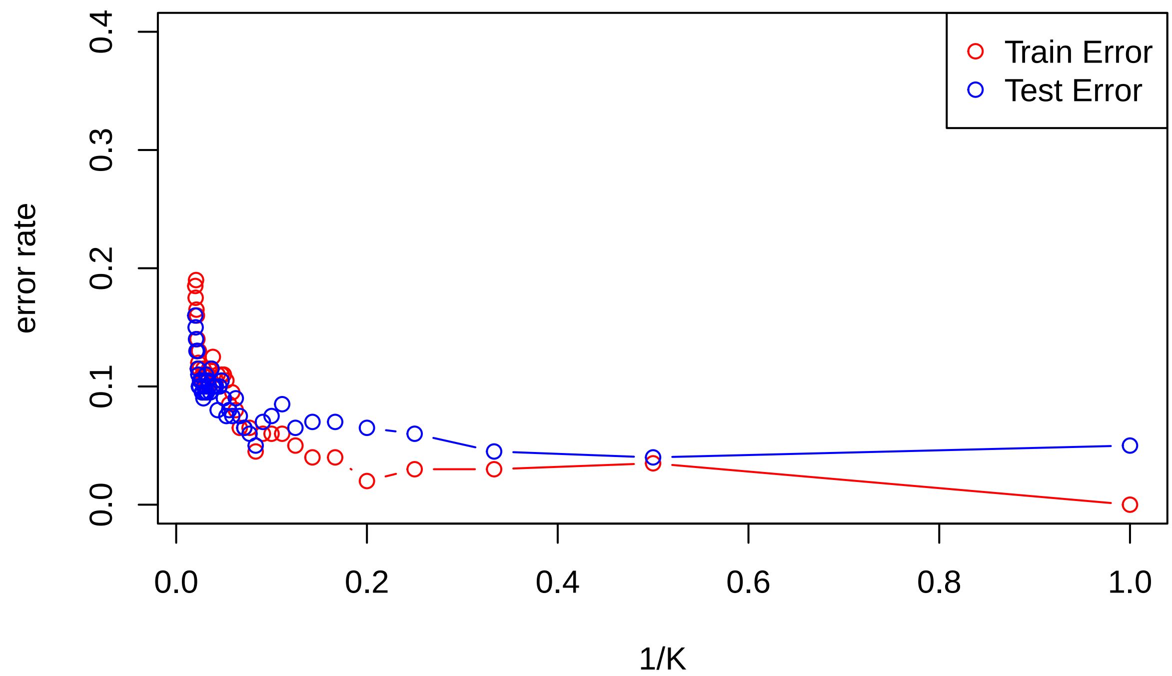
```r
plot(testset$X1, testset$X2, col = testset$color, pch = 16, xlab = "X1", ylab = "X2")
points(trainset$X1, trainset$X2, col = trainset$color, pch = 17)
legend("topleft",
       c("train_red","train_green","train_blue","test_red", "test_green", "test_blue"),
       col = c("red", "green", "blue", "red","green","blue"),
       pch = c(17,17,17,16,16,16)
       )
```



c).

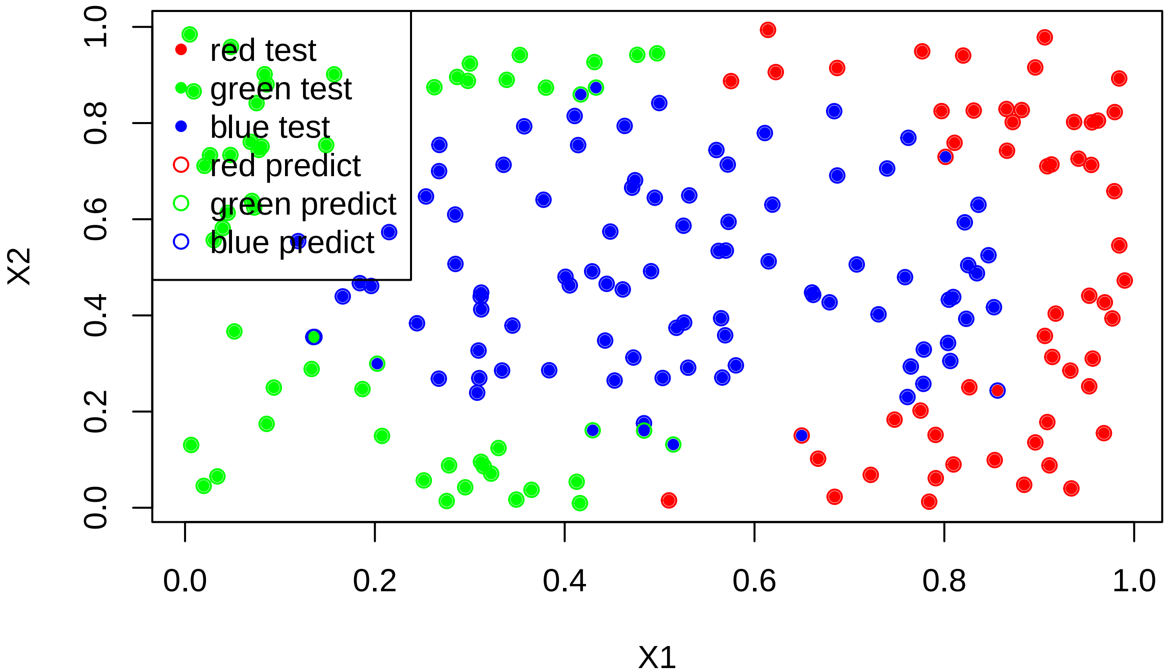
```r
library(class)
train <- trainset[,1:2]
test <- testset[,1:2]

train_error <- rep(0, 50)
test_error <- rep(0, 50)
k <- c(1:50)


smallest_test_error <- data.frame(1, NA)
for (i in k) {
  predictions_train <- knn(train, train, cl = trainset$color, k = i, prob = FALSE)
  train_error[i] <- mean(predictions_train != trainset$color)
  predictions_test <- knn(train, test, cl = trainset$color, k = i, prob = FALSE)
  test_error[i] <- mean(predictions_test != testset$color)
  if(test_error[i] < smallest_test_error[1]){
    smallest_test_error[1] = test_error[i]
    smallest_test_error[2] = i
  }
}
```

```
plot(1/k , train_error, col='red', type = 'b', ylim = c(0, 0.4), ylab = "error rate", xlab = "1/K")
points(1/k, test_error, col='blue', type = 'b')
legend("topright",
       col = c("red", "blue"),
       c("Train Error", "Test Error"),
       pch = c(1,1))
```



d).

```
plot(testset$X1, testset$X2, col = testset$color, pch = 20, xlab = "X1", ylab = "X2")
points(testset$X1, testset$X2, col = paste0("",knn(train, test, trainset$color, k = smallest_test_error
legend("topleft",
       c("red test", "green test", "blue test", "red predict", "green predict", "blue predict"),
       col = c("red","green", "blue", "red","green", "blue"),
       pch = c(20,20,20,1,1,1)
       )
```



###

e).

```
bayes_error_rate <- mean(train_error)
```

In this case, the Bayes Error Rate would be 0.0961, and it relates to part c) and d) where the Test Error is always going to be greater than the Bayes Error Rate. Hence it is just like the irreducible error term, where it is the best we can do when fitting data.

7.

```
library(ISLR2)
```

a)

```
data <- Boston
head(data)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

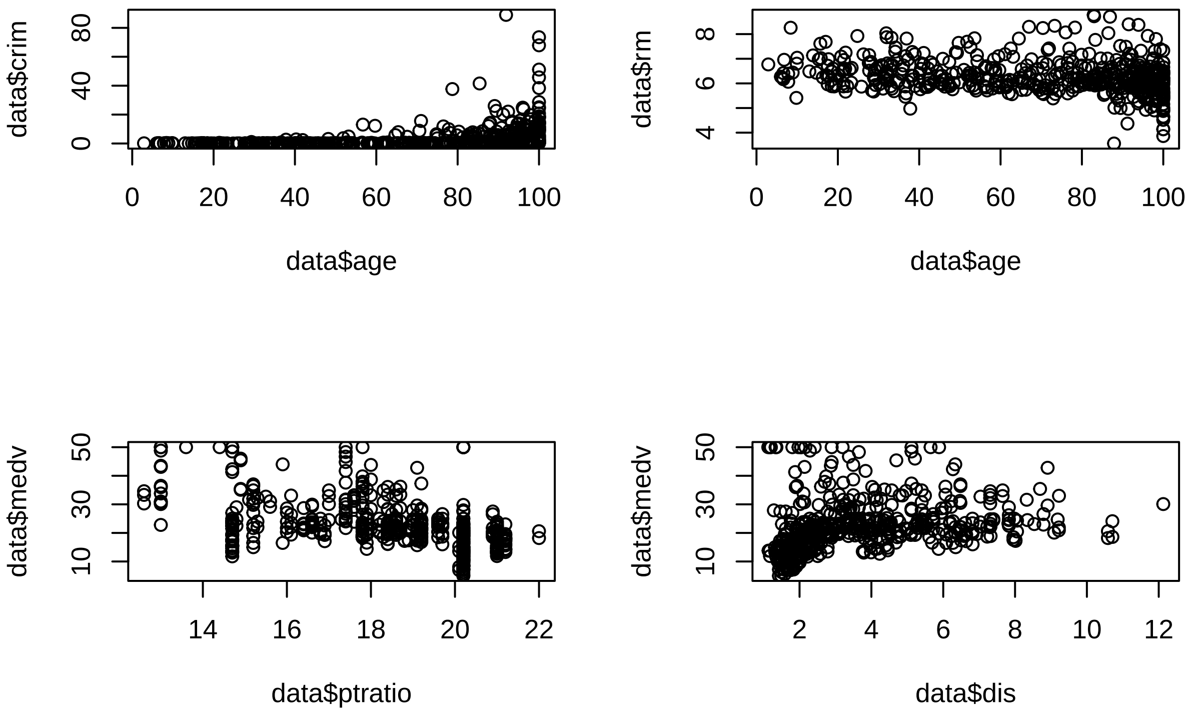
```
# Get number of rows
row_number <- nrow(data)
```

```
# Get number of columns
col_number <- ncol(data)
```

And there are 13 columns, and 506 rows. Number of rows represent the number of observations in our dataset. And number of columns represent the number of features/predictors each observations have we have.

b)

```
par(mfrow =c(2,2))
plot(data$age, data$crim) # check and see if crime rate is related to age of the units build prior to 1
plot(data$age, data$rm) # check if number of rooms is related to age of the units build prior to 1940
plot(data$ptratio, data$medv) # check if the number of students per teacher has an effect on median val
plot(data$dis, data$medv) # check if the distance to employment centers has an effect on median value o
```



My findings are:

- newily built units tends to have lower per captia crime rate by town
- newily build units tend to have more rooms compare to old units
- the higher student per teacher ratio, the lower their median house prices tend to bee
- People lives close to the business center of Boston tend to have lower median house prices

c.

From my finding in part b. I believe that there's relationship between proportion of owner-occupied units built prior to 1940 and per capita crime rate.

And the relationship is exponential, as the proportion of old houses grow, the number of per capita crime rate grows exponentially.

d.

```r
range(data$crim)
```

```
## [1]  0.00632 88.97620
```

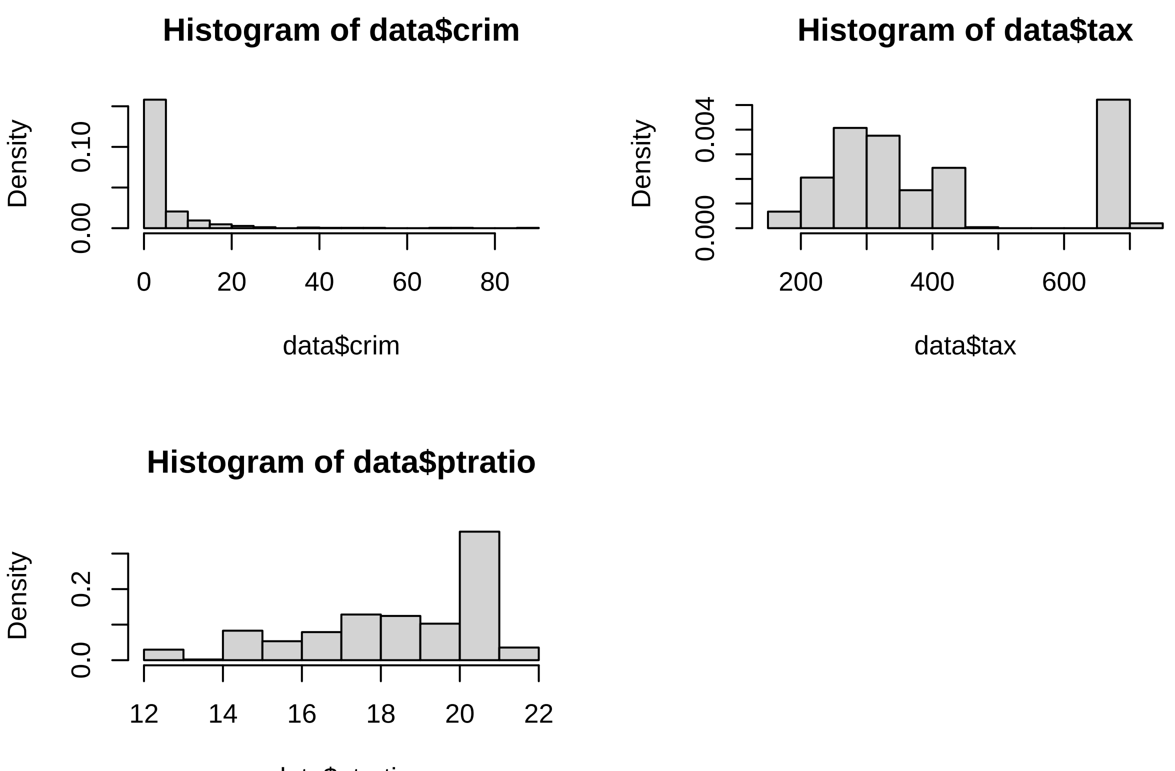
```r
range(data$tax)
```

```
## [1] 187 711
```

```r
range(data$ptratio)
```

```
## [1] 12.6 22.0
```

```r
par(mfrow = c(2,2))
hist(data$crim, freq = FALSE, breaks = 25)
hist(data$tax, freq = FALSE)
hist(data$ptratio, freq = FALSE)
```



As we can see from the above range and histograms of Per Capita Crime Rate, Tax rate and pupil-teacher ratio. We can see that the crime rate in Boston has a longer tail, hence, despite lower crime rate in most towns, some town in Boston could have really high crime rate. And for tax, there's a huge divide between suburbs. And for pupil-teacher ratio, it is skewed to the left, and peaked at 20-21.

e.

```
sum(data$chas)
```

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

35 suburbs in this data set bound the Charles river

f.

```
mean(data$ptratio)
```

```
## [1] 18.45553
```

```
sd(data$ptratio)
```

```
## [1] 2.164946
```

Mean of pupil-teacher ratio is 18.45553 and standard deviation of pupil-teacher ratio is 2.164946.

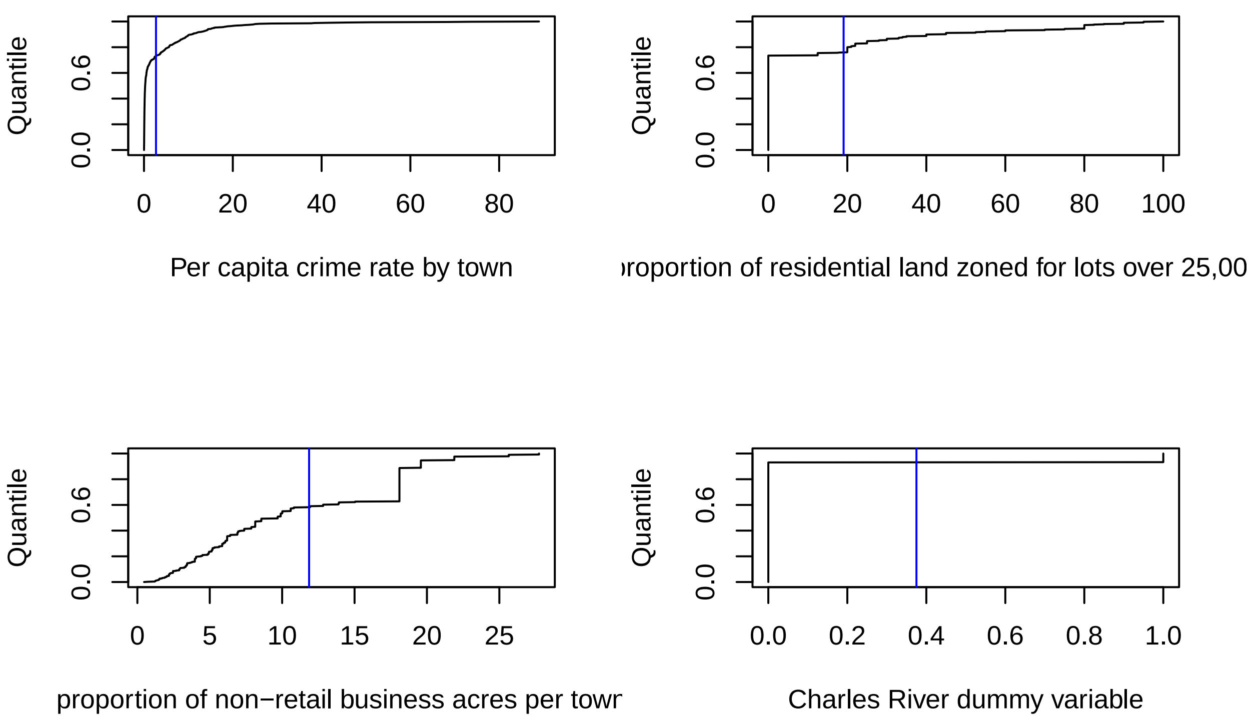
g.

```
highest_medv <- sqldf("SELECT * FROM data
             WHERE medv =(SELECT MAX(medv) FROM data)")
highest_medv
```

```
##       crim zn indus chas    nox    rm   age    dis rad tax ptratio lstat medv
## 1  1.46336  0 19.58    0 0.6050 7.489  90.8 1.9709   5 403    14.7  1.73   50
## 2  1.83377  0 19.58    1 0.6050 7.802  98.2 2.0407   5 403    14.7  1.92   50
## 3  1.51902  0 19.58    1 0.6050 8.375  93.9 2.1620   5 403    14.7  3.32   50
## 4  2.01019  0 19.58    0 0.6050 7.929  96.2 2.0459   5 403    14.7  3.70   50
## 5  0.05602  0  2.46    0 0.4880 7.831  53.6 3.1992   3 193    17.8  4.45   50
## 6  0.01381 80  0.46    0 0.4220 7.875  32.0 5.6484   4 255    14.4  2.97   50
## 7  0.02009 95  2.68    0 0.4161 8.034  31.9 5.1180   4 224    14.7  2.88   50
## 8  0.52693  0  6.20    0 0.5040 8.725  83.0 2.8944   8 307    17.4  4.63   50
## 9  0.61154 20  3.97    0 0.6470 8.704  86.9 1.8010   5 264    13.0  5.12   50
## 10 0.57834 20  3.97    0 0.5750 8.297  67.0 2.4216   5 264    13.0  7.44   50
## 11 0.01501 90  1.21    1 0.4010 7.923  24.8 5.8850   1 198    13.6  3.16   50
## 12 4.89822  0 18.10    0 0.6310 4.970 100.0 1.3325  24 666    20.2  3.26   50
## 13 5.66998  0 18.10    1 0.6310 6.683  96.8 1.3567  24 666    20.2  3.73   50
## 14 6.53876  0 18.10    1 0.6310 7.016  97.5 1.2024  24 666    20.2  2.96   50
## 15 9.23230  0 18.10    0 0.6310 6.216 100.0 1.1691  24 666    20.2  9.53   50
## 16 8.26725  0 18.10    1 0.6680 5.875  89.6 1.1296  24 666    20.2  8.88   50
```

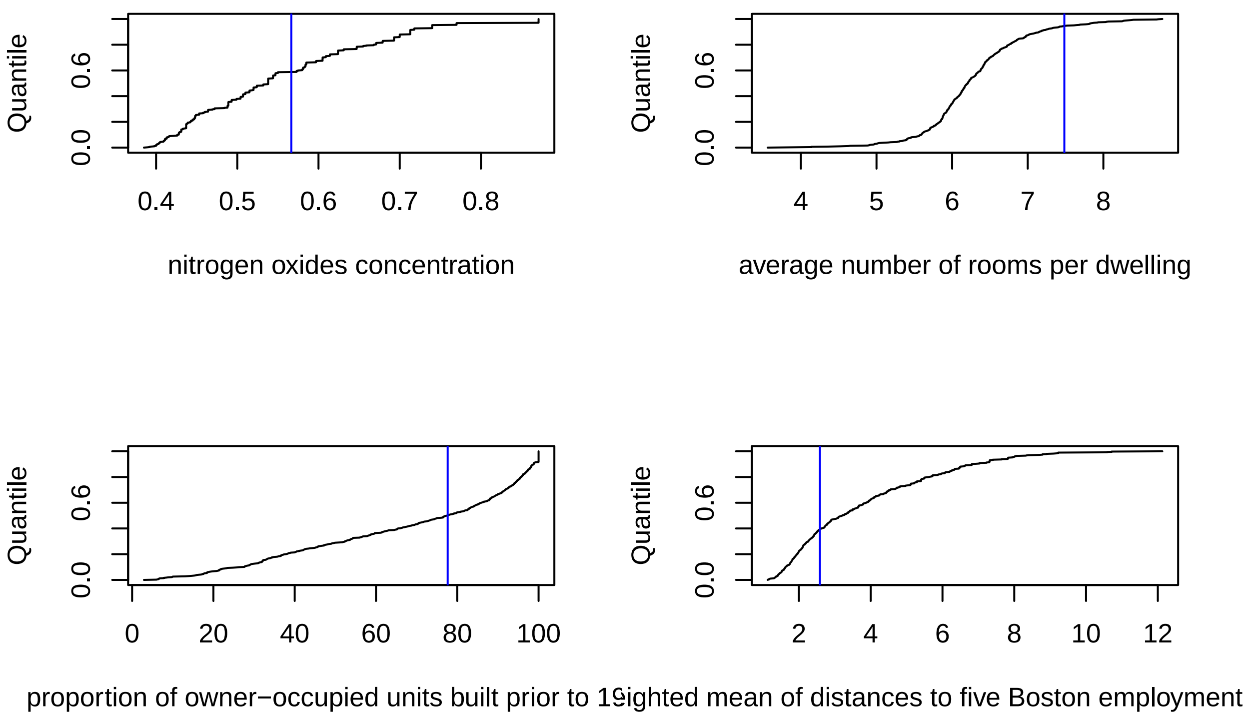
```
n <- row_number
par(mfrow = c(2,2))
plot(sort(data$crim),(1:n - 1)/(n - 1), type="l",
xlab = "Per capita crime rate by town",
```

```
ylab = "Quantile")
abline(v=mean(highest_medv$crim), col="blue")


plot(sort(data$zn), (1:n - 1)/(n - 1), type="l",
xlab = "proportion of residential land zoned for lots over 25,000 sq.ft",
ylab = "Quantile")
abline(v=mean(highest_medv$zn), col="blue")


plot(sort(data$indus), (1:n - 1)/(n - 1), type="l",
xlab = "proportion of non-retail business acres per town",
ylab = "Quantile")
abline(v=mean(highest_medv$indus), col="blue")


plot(sort(data$chas), (1:n - 1)/(n - 1), type="l",
xlab = "Charles River dummy variable",
ylab =  "Quantile")
abline(v=mean(highest_medv$chas), col="blue")
```



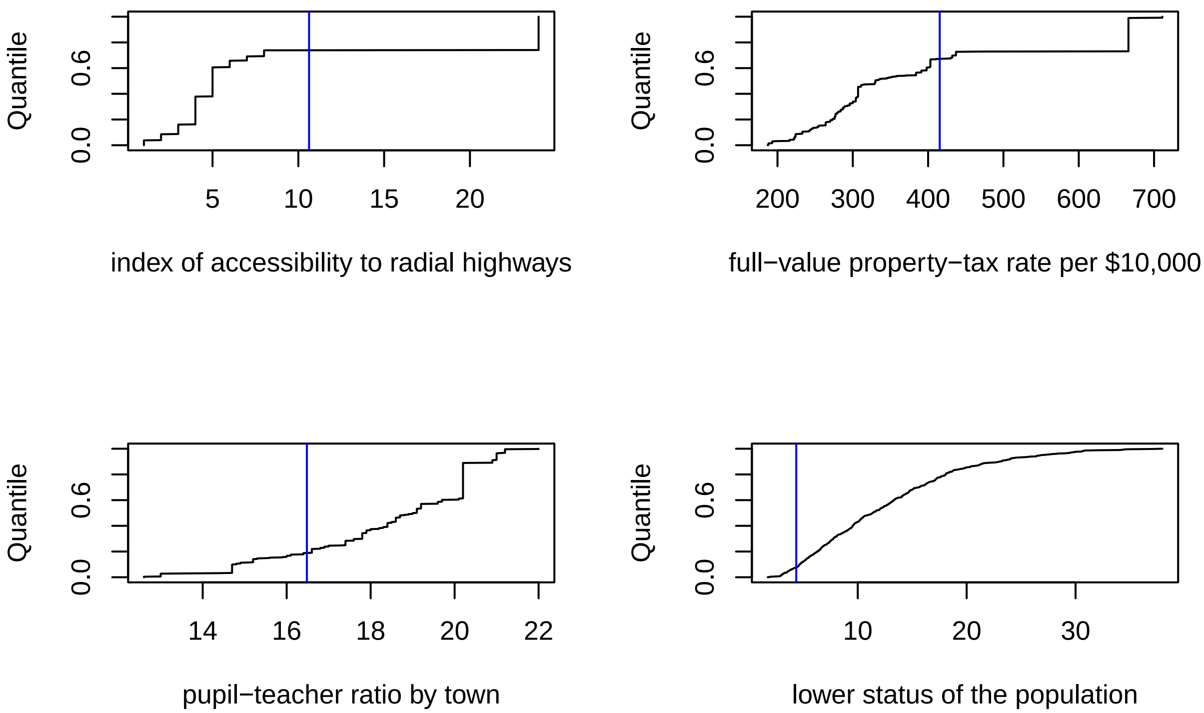

```
par(mfrow = c(2,2))

plot(sort(data$nox),(1:n - 1)/(n - 1), type="l",
xlab = "nitrogen oxides concentration",
ylab = "Quantile")
abline(v=mean(highest_medv$nox), col="blue")


plot(sort(data$rm), (1:n - 1)/(n - 1), type="l",
```

```r
xlab = "average number of rooms per dwelling",
ylab = "Quantile")
abline(v=mean(highest_medv$rm), col="blue")


plot(sort(data$age), (1:n - 1)/(n - 1), type="l",
xlab = "proportion of owner-occupied units built prior to 1940",
ylab = "Quantile")
abline(v=mean(highest_medv$age), col="blue")


plot(sort(data$dis), (1:n - 1)/(n - 1), type="l",
xlab = "weighted mean of distances to five Boston employment centres",
ylab =  "Quantile")
abline(v=mean(highest_medv$dis), col="blue")
```




```r
par(mfrow = c(2,2))

plot(sort(data$rad), (1:n - 1)/(n - 1), type="l",
xlab = "index of accessibility to radial highways",
ylab = "Quantile")
abline(v=mean(highest_medv$rad), col="blue")


plot(sort(data$tax), (1:n - 1)/(n - 1), type="l",
xlab = "full-value property-tax rate per $10,000",
ylab =  "Quantile")
abline(v=mean(highest_medv$tax), col="blue")

plot(sort(data$ptratio), (1:n - 1)/(n - 1), type="l",
```

```r
xlab = "pupil-teacher ratio by town",
ylab = "Quantile")
abline(v=mean(highest_medv$ptratio), col="blue")


plot(sort(data$lstat), (1:n - 1)/(n - 1), type="l",
xlab = "lower status of the population",
ylab =  "Quantile")
abline(v=mean(highest_medv$lstat), col="blue")
```




```r
summary(Boston)
```

```
##       crim                 zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          lstat
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   : 1.73
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.: 6.95
##  Median : 5.000   Median :330.0   Median :19.05   Median :11.36
```

```
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :12.65
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:16.95
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :37.97
##       medv
##  Min.   : 5.00
##  1st Qu.:17.02
##  Median :21.20
##  Mean   :22.53
##  3rd Qu.:25.00
##  Max.   :50.00
```

As we can see from above quantile graphs, we can see that suburbs with highest median value of owner-occupied homes(in blue line) tends to:

- have low per capita crime rate by town
- have smaller proportion of residential land
- at 50th percentile for proportion of non-retail business acres per town
- at around 50th percentile for nitrogen oxides concentration
- have more rooms
- at around 50th percentile for proportion of owner-occupied units built prior to 1940
- closer to five Boston employment centers
- between median and 3rd quantile in terms of accessibility to radial highways
- at around 50th percentile for full-value property-tax rate per $10,000.
- have smaller pupil-teacher ratio by town
- have smaller percent of lower status of the population

h.

```
sqldf("SELECT COUNT(*) AS more_than_six FROM Boston
      WHERE rm > 6")
```

```
##   more_than_six
## 1           333
```

```
sqldf("SELECT * FROM Boston
      WHERE rm > 8")
```

```
##       crim zn indus chas    nox    rm  age    dis rad tax ptratio lstat medv
## 1  0.12083  0  2.89    0 0.4450 8.069 76.0 3.4952   2 276    18.0  4.21 38.7
## 2  1.51902  0 19.58    1 0.6050 8.375 93.9 2.1620   5 403    14.7  3.32 50.0
## 3  0.02009 95  2.68    0 0.4161 8.034 31.9 5.1180   4 224    14.7  2.88 50.0
## 4  0.31533  0  6.20    0 0.5040 8.266 78.3 2.8944   8 307    17.4  4.14 44.8
## 5  0.52693  0  6.20    0 0.5040 8.725 83.0 2.8944   8 307    17.4  4.63 50.0
## 6  0.38214  0  6.20    0 0.5040 8.040 86.5 3.2157   8 307    17.4  3.13 37.6
## 7  0.57529  0  6.20    0 0.5070 8.337 73.3 3.8384   8 307    17.4  2.47 41.7
## 8  0.33147  0  6.20    0 0.5070 8.247 70.4 3.6519   8 307    17.4  3.95 48.3
## 9  0.36894 22  5.86    0 0.4310 8.259  8.4 8.9067   7 330    19.1  3.54 42.8
## 10 0.61154 20  3.97    0 0.6470 8.704 86.9 1.8010   5 264    13.0  5.12 50.0
## 11 0.52014 20  3.97    0 0.6470 8.398 91.5 2.2885   5 264    13.0  5.91 48.8
## 12 0.57834 20  3.97    0 0.5750 8.297 67.0 2.4216   5 264    13.0  7.44 50.0
## 13 3.47428  0 18.10    1 0.7180 8.780 82.9 1.9047  24 666    20.2  5.29 21.9
```

It tends to have lower crime rate.