STAT 435 HW1

Chongyi Xu

April 5, 2018

1. We will perform k-nearest-neighbors in this problem, in a setting with 2 classes, 25 observations per class, and p = 2 features. We will call one class the "red" class and the other class the "blue" class. The observations in the red class are drawn i.i.d. from a $N_p(\mu_r, I)$ distribution, and the observations in the blue class are drawn i.i.d. from a $N_p(\mu_b, I)$ distribution, where $\mu_r = \binom{0}{0}$ is the mean in the red class, and where $\mu_b = \binom{1.5}{1.5}$is the mean in the blue class.

a. Generate a training set, consisting of 25 observations from the read class and 25 observations from the blue class. Plot the training set. Make sure that the axes are properly labeled, and that the observations are colored according to their class label.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

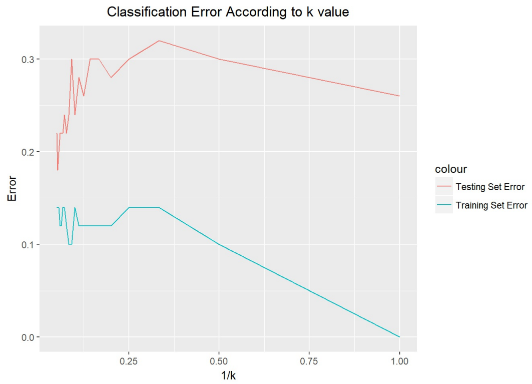
```
set.seed(12345)
train <- matrix(NA, 50, 2)
label <- rep('', 50)
# red
train[1:25, 1] <- rnorm(n=25, mean=0, sd=1)
train[1:25, 2] <- rnorm(n=25, mean=0, sd=1)
label[1:25] <- 'red'

# blue
train[26:50, 1] <- rnorm(n=25, mean=1.5, sd=1)
train[26:50, 2] <- rnorm(n=25, mean=1.5, sd=1)
label[26:50] <- 'blue'

train_dat <- data.frame(feature1=train[,1], feature2=train[,2])
plot <- ggplot(train_dat, aes(x=feature1, y=feature2)) + geom_point(aes(color=label)) + scale_color_manual(valu
es=c('blue', 'red')) + ggtitle("Training Set") + theme(plot.title=element_text(hjust=0.5)) + xlab('X1') + ylab(
'X2')
plot
```

b. Now generate a test set consisting of 25 observations from the red class and 25 observations from the blue class. On a single plot, display both the training and test set, using one symbol to indicate training observations (e.g. circles) and another symbol to indicate the test observations (e.g. squares). Make sure that the axes are properly labeled, that the symbols for training and test observations are explained in a legend, and that the observations are colored according to their class label.

```
test <- matrix(NA, 50, 2)
testlab <- rep('', 50)
# red
test[1:25, 1] <- rnorm(n=25, mean=0, sd=1)
test[1:25, 2] <- rnorm(n=25, mean=0, sd=1)
testlab[1:25] <- 'red'

# blue
test[26:50, 1] <- rnorm(n=25, mean=1.5, sd=1)
test[26:50, 2] <- rnorm(n=25, mean=1.5, sd=1)
testlab[26:50] <- 'blue'

test_dat <- data.frame(feature1=test[,1], feature2=test[,2])
ggplot() + geom_point(data=train_dat, aes(x=feature1, y=feature2, color=label, shape='Training Set')) + geom_po
int(data=test_dat, aes(x=feature1, y=feature2, color=testlab, shape='Testing Set')) + scale_color_manual(values
=c('blue', 'red')) + ggtitle("Training Set and Testing Set Data") + theme(plot.title=element_text(hjust=0.5)) +
 xlab('X1') + ylab('X2')
```

c. Using the knn function in the library class, fit a k-nearest neighbors model on the training set, for a range of values of k from 1 to 20. Make a plot that displays the value of 1=k on the x-axis, and classification error (both training error and test error) on the y-axis. Make sure all axes and curves are properly labeled. Explain your results.

```
library(class)
k <- 20
err <- matrix(NA, k, 2)
for (kk in 1:k) {
  test_train <- knn(train, train, cl=label, k=kk)
  err[kk, 1] <- sum(test_train != label) / 50
  test_test <- knn(train, test, cl=label, k=kk)
  err[kk, 2] <- sum(test_test != testlab) / 50
}

x = 1 / (1:k)
ggplot() + geom_line(aes(x=x, y=err[, 1], color='Training Set Error')) + geom_line(aes(x=x, y=err[, 2], color='
Testing Set Error')) + ggtitle('Classification Error According to k value') + theme(plot.title=element_text(hju
st=0.5)) + xlab('1/k') + ylab('Error')
```



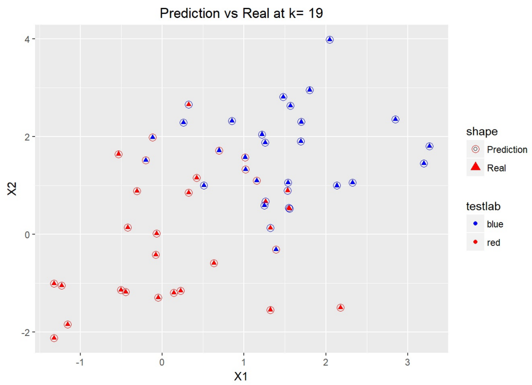
From the graph above, it can be seen that as $\frac{1}{k}$ increases (k decreases), the classification error of training set decreases but the classification error of testing set increases. The reason is that as k decreases, the model becomes more flexible but overfitting. In the extreme case, when $k = 1$, the model is excessively flexible and overfits.

d. For the value of k that resulted in the smallest test error in part (c) above, make a plot displaying the test observations as well as their true and predicted class labels. Make sure that all axes and points are clearly labeled.

```
k <- which(err[, 2]==min(err[, 2]))[1]
prediction <- knn(train, test, cl=label)
blues <- which(prediction=='blue')
reds <- which(prediction=='red')

plot <- ggplot() + geom_point(aes(x=test[, 1], y=test[, 2], color=testlab, shape='Real')) + scale_color_manual(
values=c('blue', 'red')) + geom_point(aes(x=test[blues, 1], y=test[blues, 2], shape='Prediction'), color='blue'
, cex=3, lwd=2) + geom_point(aes(x=test[reds, 1], y=test[reds, 2], shape='Prediction'), color='red', cex=3, lwd
=2) + scale_shape_manual(values=c(1, 17)) + ggtitle(paste('Prediction vs Real at k=', k)) + xlab('X1') + ylab('
X2') + theme(plot.title=element_text(hjust=0.5))
```

```
## Warning: The plyr::rename operation has created duplicates for the
## following name(s): (`size`)

## Warning: The plyr::rename operation has created duplicates for the
## following name(s): (`size`)
```

```
plot
```



e. In this example, what is the Bayes error rate? Justify your anwer.

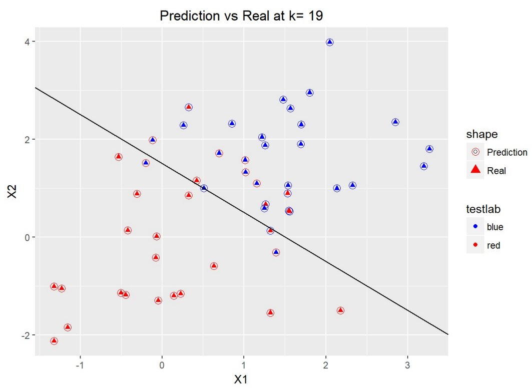
Bayes error rate is given by

$$\begin{aligned} err &= 1 - E(max_j Pr(Y=j|X)) \\ &= 1 - E(max_j \frac{P(X|Y=j)P(Y=j)}{P(X)}) \\ &= 1 - E(max_{j\in\{blue,red\}} \frac{P(X|Y=j)P(Y=j)}{P(X)}), P(Y=j) = \frac{1}{2} \\ &= 1 - \int max\{\frac{P(X|Y=blue)}{2P(X)}, \frac{P(X|Y=red)}{2P(X)}\} * P(X)dx \\ &= 1 - \frac{1}{2}\int max\{P(X|Y=blue),\ where\ P(X|Y=red)\}dx \\ &= 1 - \frac{1}{2}\int_{E_1} P(X|Y=blue)dx - \frac{1}{2}\int_{E_2} P(X|Y=red)dx \end{aligned}$$

where E_1 denotes the event that $X_1 \in [a_1, b_1], X_2 \in [a_2, b_2]$such that it is more likely to be blue, and E_2 denotes for the similar event for being red.

Back to graph, we would like to find out the interval for those events.

```
plot + geom_abline(slope=-1, intercept=1.5)
```



We can see that below the line $X_2 = -X_1 + \frac{3}{2}$, it is more likely to be red and above the line, it is more likely to be blue. Therefore,

$$\begin{aligned} err &= 1 - \frac{1}{2}\int_{E_1} P(X|Y=blue)dx - \frac{1}{2}\int_{E_2} P(X|Y=red)dx \\ &= 1 - \frac{1}{2}\int_{X_2 > -X_1+\frac{3}{2}} P(X|Y=blue)dx - \frac{1}{2}\int_{X_2 < -X_1+\frac{3}{2}} P(X|Y=red)dx \end{aligned}$$

```
1 - pnorm(sqrt(1.5^2+1.5^2)/2)
```

```
## [1] 0.1444222
```

So the Bayes error is 0.1444222.

2. We will once again perform k-nearest-neighbors in a setting with p = 2 features. But this time, we'll generate the data differently: let

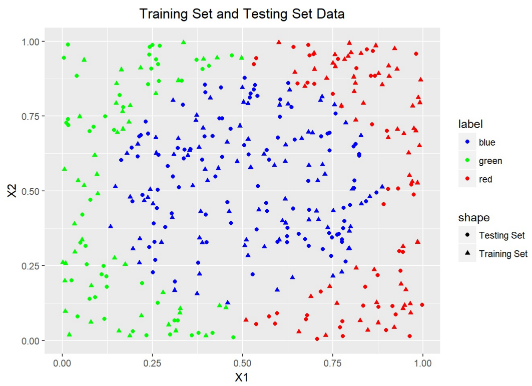
$X1 \sim Unif[0,1]$ and $X2 \sim Unif[0,1]$, i.e. the observations for each feature are i.i.d. from a uniform distribution. An observation belongs to class "red" if $(X_1-0.5)^2+(X_2-0.5)^2>0.15$ and $X_1>0.5$; to class "green" if $(X_1-0.5)^2+(X_2-0.5)^2>0.15$ and $X_1 \leq 0.5$; and to class "blue" otherwise.

a. Generate a training set of n = 200 observations. (You will want to use the R function runif.) Plot the training set. Make sure that the axes are properly labeled, and that the observations are colored according to their class label.

```
set.seed(12345)
train <- matrix(NA, 200, 2)
train_label <- rep('', 200)
# Xs
train[, 1] <- runif(n=200, min=0, max=1)
train[, 2] <- runif(n=200, min=0, max=1)

for (i in 1:200) {
  if (((train[i,1]-0.5)^2+(train[i,2]-0.5)^2>0.15) & (train[i,1]>0.5)) {
    train_label[i] = "red"
  } else if (((train[i,1]-0.5)^2+(train[i,2]-0.5)^2>0.15) & (train[i,1]<=0.5)) {
    train_label[i] = "green"
  } else {
    train_label[i] = "blue"
  }
}

train_dat <- data.frame(feature1=train[,1], feature2=train[,2])
plot <- ggplot(train_dat, aes(x=feature1, y=feature2)) + geom_point(aes(color=train_label)) + scale_color_manua
l(values=c('blue', 'green', 'red')) + ggtitle("Training Set") + theme(plot.title=element_text(hjust=0.5)) + xla
b('X1') + ylab('X2')
plot
```

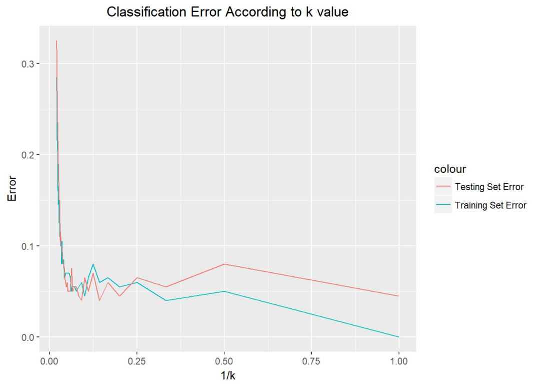
b. Now generate a test set consisting of 25 observations from the red class and 25 observations from the blue class. On a single plot, display both the training and test set, using one symbol to indicate training observations (e.g. circles) and another symbol to indicate the test observations (e.g. squares). Make sure that the axes are properly labeled, that the symbols for training and test observations are explained in a legend, and that the observations are colored according to their class label.

```
test <- matrix(NA, 200, 2)
test_label <- rep('', 200)
# red
test[, 1] <- runif(n=200, min=0, max=1)
test[, 2] <- runif(n=200, min=0, max=1)

for (i in 1:200) {
  if (((test[i,1]-0.5)^2+(test[i,2]-0.5)^2>0.15) & (test[i,1]>0.5)) {
    test_label[i] = "red"
  } else if (((test[i,1]-0.5)^2+(test[i,2]-0.5)^2>0.15) & (test[i,1]<=0.5)) {
    test_label[i] = "green"
  } else {
    test_label[i] = "blue"
  }
}

test_dat <- data.frame(feature1=test[,1], feature2=test[,2])
ggplot() + geom_point(data=train_dat, aes(x=feature1, y=feature2, color=train_label, shape='Training Set')) + g
eom_point(data=test_dat, aes(x=feature1, y=feature2, color=test_label, shape='Testing Set')) + scale_color_manu
al(values=c('blue', 'green', 'red'), name='label') + ggtitle("Training Set and Testing Set Data") + theme(plot.
title=element_text(hjust=0.5)) + xlab('X1') + ylab('X2')
```



c. Using the knn function in the library class, fit a k-nearest neighbors model on the training set, for a range of values of k from 1 to 50. Make a plot that displays the value of 1=k on the x-axis, and classification error (both training error and test error) on the y-axis. Make sure all axes and curves are properly labeled. Explain your results.

```
k <- 50
err <- matrix(NA, k, 2)
for (kk in 1:k) {
  test_train <- knn(train, train, cl=train_label, k=kk)
  err[kk, 1] <- sum(test_train != train_label) / 200
  test_test <- knn(train, test, cl=train_label, k=kk)
  err[kk, 2] <- sum(test_test != test_label) / 200
}

x = 1 / (1:k)
ggplot() + geom_line(aes(x=x, y=err[, 1], color='Training Set Error')) + geom_line(aes(x=x, y=err[, 2], color='
Testing Set Error')) + ggtitle('Classification Error According to k value') + theme(plot.title=element_text(hju
st=0.5)) + xlab('1/k') + ylab('Error')
```

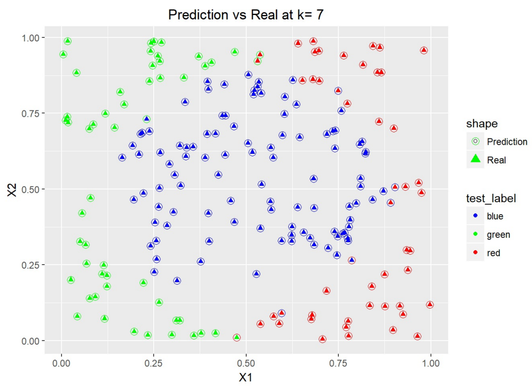


From the graph above, we can see that the classification error has a completely different curve comparing to problem 1. This indicates that the $k = 1$ case does not overfit as much as it does in problem 1.

d. For the value of k that resulted in the smallest test error in part (c) above, make a plot displaying the test observations as well as their true and predicted class labels. Make sure that all axes and points are clearly labeled.

```
k <- which(err[, 2]==min(err[, 2]))[1]
prediction <- knn(train, test, cl=train_label)
blues <- which(prediction=='blue')
greens <- which(prediction=='green')
reds <- which(prediction=='red')

ggplot() + geom_point(aes(x=test[, 1], y=test[, 2], color=test_label, shape='Real')) + scale_color_manual(value
s=c('blue', 'green', 'red')) + geom_point(aes(x=test[blues, 1], y=test[blues, 2], shape='Prediction'), color='b
lue', cex=3, lwd=2) + geom_point(aes(x=test[reds, 1], y=test[reds, 2], shape='Prediction'), color='red', cex=3,
 lwd=2) + geom_point(aes(x=test[greens, 1], y=test[greens, 2], shape='Prediction'), color='green', cex=3, lwd=2
) + scale_shape_manual(values=c(1, 17)) + ggtitle(paste('Prediction vs Real at k=', k)) + xlab('X1') + ylab('X2
') + theme(plot.title=element_text(hjust=0.5))
```

```
## Warning: The plyr::rename operation has created duplicates for the
## following name(s): (`size`)

## Warning: The plyr::rename operation has created duplicates for the
## following name(s): (`size`)

## Warning: The plyr::rename operation has created duplicates for the
## following name(s): (`size`)
```



e. In this example, what is the Bayes error rate?

In this problem, since Y is well-defined as a piece-wise constant function, we will have $max_j P(Y = j|X) = 1$ for all X. So the error will be $err = 1 - max_j P(Y = j|X) = 0$

In part (c) and (d), we found that the data will not overfit too much even with small k values. This is due to the well-defined Y function. Under this circumstance, the three kinds of data (blue, green and red) does not overlap (according to the graph above) and it supports us to derive a more complex model (such as $k = 1$) without overfitting the data.

3. For each scenario, determine whether it is a regression or a classification problem, determine whether the goal is inference or prediction, and state the values of n (sample size) and p (number of predictors).

a. I want to predict each student's final exam score based on his or her homework scores. There are 50 students enrolled in the course, and each student has completed 8 homeworks.

A regression problem and the final exam score is quantative. We would like to predict the scores (the goal is prediction). The sample size $n = 50$ and $p = 8$ for 8 homework scores.

b. I want to understand the factors that contribute to whether or not a student passes this course. The factors that I consider are (i) whether or not the student has previous programming experience; (ii) whether or not the student has previously studied linear algebra; (iii) whether or not the student has taken a previous stats/probability course; (iv) whether or not the student attends office hours; (v) the student's overall GPA; (vi) the student's year (e.g. freshman, sophomore, junior, senior, or grad student). I have data for all 50 students enrolled in the course.

A classification problem. The goal is inference since we are interested in if these factors contributes to passing the course or not. The sample size $n = 50$ and $p = 6$ for 6 different categories of factors we are interested in.

4. In each setting, would you generally expect a flexible or an inflexible statistical machine learning method to perform better? Justify your answer.

a. Sample size n is very small, and number of predictors p is very large.

An inflexible method. With large number of predictors, a flexible method will result in overfitting.

b. Sample size n is very large, and number of predictors p is very small.

A flexible method. Since n is large and p is small, a flexible model will have less bias without overfitting the data too much.

c. Relationship between predictors and response is highly non-linear.

A flexible method. An inflexible model is not good enough at telling the non-linearity of the response.

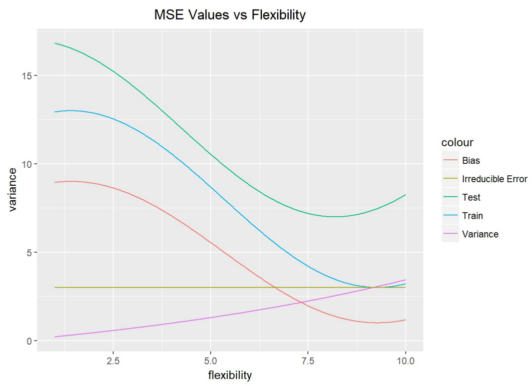
d. The variance of the error terms, i.e. $\sigma^2 = Var(\epsilon)$ is extremely high.

An inflexible method. Since high variance will make the model overfitting the data if we are using a flexible method.

5. This question has to do with the bias-variance decomposition.

a. Make a sketch of typical (squared) bias, variance, training error, test error, and Bayes (or irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods to more flexible approaches. The x-axis should represent the amount of flexibility in the model, and the y-axis should represent the values of each curve. There should be five curves.

```
flexibility <- seq(from=1, to=10, by=0.01)
variance <- 2*exp(flexibility*0.1)-2
train <- 5*cos(0.4*flexibility+12) + 8
test <- 5*cos(0.4*flexibility+25) + 12
bias <- 4*cos(0.4*flexibility+12) + 5
irreducible_err <- 3

ggplot() + geom_line(aes(x=flexibility, y=variance, color='Variance')) + geom_line(aes(x=flexibility, y=train,
color='Train')) + geom_line(aes(x=flexibility, y=bias, color='Bias')) + geom_line(aes(x=flexibility, y=irreduci
ble_err, color='Irreducible Error')) + geom_line(aes(x=flexibility, y=test, color='Test')) + ggtitle('MSE Value
s vs Flexibility') + theme(plot.title=element_text(hjust=0.5))
```



b. Explain why each of the five curves has the shape displayed in (a).

- Bias: With increasing flexibility, bias decreases and it will have a greater decreasing speed rather than variance increasing.
- Variance: With increasing flexibility, variance increases.
- Training Error: With increasing flexibility, the training error will decrease because a mroe flexible model will fits the data better which will decrease the training error.
- Testing Error: With increasing flexibility, the testing error will generally decrease but if the model overfits, the testing error will significantly increase at that point.
- Irreducible Error: Will be a constant since it is irreducible and would not change due to the flexibility.

6. This exercise involves the Boston housing data set, which is part of the MASS library in R.

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

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.3.3
```

```
dat <- Boston

nrow(Boston)
```

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

```
ncol(Boston)
```

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

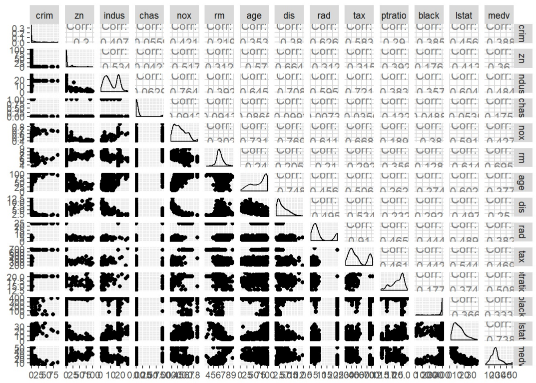
It has 506 rows and 14 columns. And it contains the columns according to the documentation: * crim: per capita crime rate by town. * zn: proportion of residential land zoned for lots over 25,000 sq.ft. * indus: proportion of non-retail business acres per town. * chas: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise). * nox: nitrogen oxides concentration (parts per 10 million). * rm: average number of rooms per dwelling. * age: proportion of owner-occupied units built prior to 1940. * dis: weighted mean of distances to five Boston employment centres. * rad: index of accessibility to radial highways. * tax: full-value property-tax rate per $10,000. * ptratio: pupil-teacher ratio by town. * black: $1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town. * lstat: lower status of the population (percent). * medv: median value of owner-occupied homes in $1000s.

b. Make some pairwise scatterplots of the predictors (columns) in this data set. Describe your findings.

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 3.3.3
```

```
ggpairs(Boston)
```



We can get a lot of correlations from the pair-wise plot above. For example, the (nox, dis) pair tells that nitrogen oxides concentration is highly related with weighted mean of distances to five Boston employment centres which makes sense.

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

```
cor(Boston)["crim", ]
```

```
##        crim          zn       indus        chas         nox          rm 
##  1.00000000 -0.20046922  0.40658341 -0.05589158  0.42097171 -0.21924670 
##         age         dis         rad         tax     ptratio       black 
##  0.35273425 -0.37967009  0.62550515  0.58276431  0.28994558 -0.38506394 
##       lstat        medv 
##  0.45562148 -0.38830461
```

From the covariance above, we can see that rad and tax has relatively high association with per capita crime rate. There are also some relatively weak associations such as indus, nox, lstat.

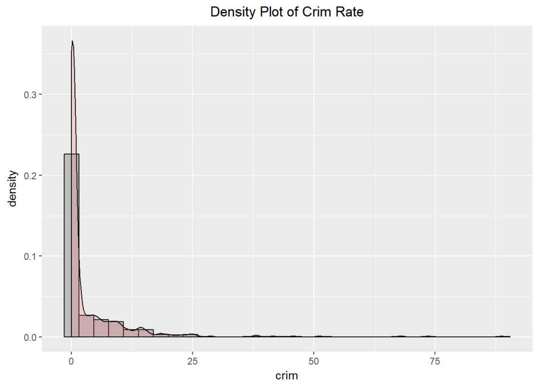
d. Do any of the suburbs of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher ratios? Comment on the range of each predictor.

Use histogram and density plot to find if any suburbs satisfies the conditions above.

- Crime Rate

```
ggplot(dat=Boston, aes(x=crim)) + geom_histogram(aes(y=..density..), color="black", fill="grey") + geom_density
(alpha=.2, fill="#FF6666") + ggtitle('Density Plot of Crim Rate') + theme(plot.title=element_text(hjust=0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can see that most suburbs have really low crime rate (close to 0) but there are a few suburbs have high crime rate. The range of crime rate is widely spread.

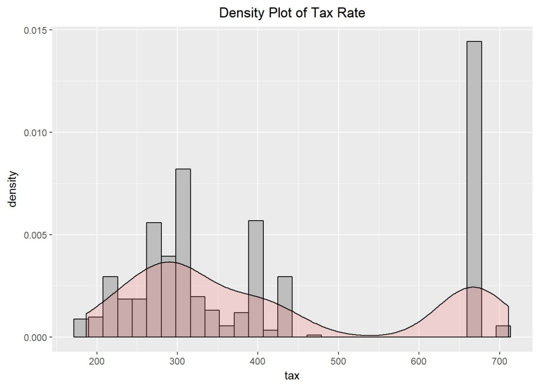
```
range(Boston$crim)
```

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

- Tax Rate

```
ggplot(dat=Boston, aes(x=tax)) + geom_histogram(aes(y=..density..), color="black", fill="grey") + geom_density(
alpha=.2, fill="#FF6666") + ggtitle('Density Plot of Tax Rate') + theme(plot.title=element_text(hjust=0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



From the graph above, we can see that the sububrs that have low tax rate and those have high tax rate are generally in two groups.

```
range(Boston$tax)
```

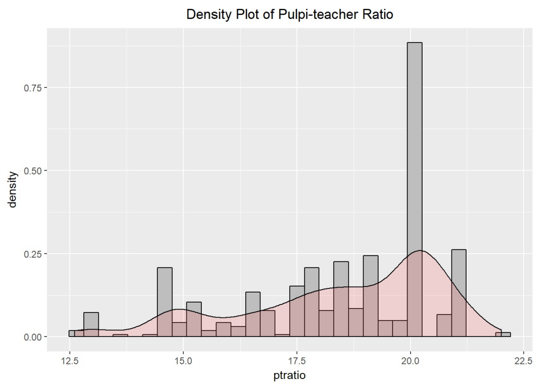
```
## [1] 187 711
```

And the range is also wide.

- Pupil-teacher Ratio

```
ggplot(dat=Boston, aes(x=ptratio)) + geom_histogram(aes(y=..density..), color="black", fill="grey") + geom_dens
ity(alpha=.2, fill="#FF6666") + ggtitle('Density Plot of Pulpi-teacher Ratio') + theme(plot.title=element_text(
hjust=0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



It can be seen that there are a few schools have low pulpi-teacher ratio but generally the ratio is pretty high.

```
range(Boston$ptratio)
```

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

e. How many suburbs in this data set bound the Charles river?

```
length(which(Boston$chas==1))
```

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

There are 35 suburbs in this data set bound the Charles river.

f. What are the mean and standard deviation of the pupil-teacher ratio among the towns in this data set?

```
mean(Boston$ptratio)
```

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

```
sd(Boston$ptratio)
```

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

The mean of the pupil-teacher ratio is 18.46 and the standard deviation is 2.16

g. Which suburb of Boston has highest median value of owner-occupied homes? What are the values of the other predictors for that suburb, and how do those values compare to the overall ranges for those predictors?

```
Boston[which(Boston$medv==max(Boston$medv)),]
```

	crim	zn	indus	chas	nox	rm	age	dis
	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
162	1.46336	0	19.58	0	0.6050	7.489	90.8	1.9709
163	1.83377	0	19.58	1	0.6050	7.802	98.2	2.0407
164	1.51902	0	19.58	1	0.6050	8.375	93.9	2.1620
167	2.01019	0	19.58	0	0.6050	7.929	96.2	2.0459
187	0.05602	0	2.46	0	0.4880	7.831	53.6	3.1992
196	0.01381	80	0.46	0	0.4220	7.875	32.0	5.6484
205	0.02009	95	2.68	0	0.4161	8.034	31.9	5.1180
226	0.52693	0	6.20	0	0.5040	8.725	83.0	2.8944
258	0.61154	20	3.97	0	0.6470	8.704	86.9	1.8010
268	0.57834	20	3.97	0	0.5750	8.297	67.0	2.4216

1-10 of 16 rows | 1-9 of 15 columns Previous **1** 2 Next

From the table above, we can see that there are two groups that one with relatively high crime rate (~9%) and one with lower (~0% to ~2%). However, both groups have relatively low crime rate comparing to the total range of the crime rate in the town. The age spread widely from 24 to 100. And the index of accessibility to radial highways are mostly grouping at 5 and 24. Comparing to the total range, these suburbs are all close to Boston employment centres.

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

```
length(which(Boston$rm > 6))
```

```
## [1] 333
```

There are 333 suburbs have an average that is more than six rooms per dwelling.

```
length(which(Boston$rm > 8))
```

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

There are 13 suburbs have an average that is more than eight rooms per dwelling.

```
above8 <- apply(Boston[which(Boston$rm > 8),], 2, mean)
above8
```

```
##        crim          zn       indus        chas         nox          rm 
##   0.7187954  13.6153846   7.0784615   0.1538462   0.5392385   8.3485385 
##         age         dis         rad         tax     ptratio       black 
##  71.5384615   3.4301923   7.4615385 325.0769231  16.3615385 385.2107692 
##       lstat        medv 
##   4.3100000  44.2000000
```

We can see that the group that has more than eight rooms per dwelling (say, $Group_1$)has extremely low crime rate and high median value of owner-occupied home in $1000s.

```
above8 - apply(Boston[which(Boston$rm <= 8),], 2, mean)
```

```
##         crim           zn        indus         chas          nox 
##  -2.97105975   2.31112498  -4.16533156   0.08690903  -0.01586418 
##           rm          age          dis          rad          tax 
##   2.11832751   3.04170697  -0.37447118  -2.14292401 -85.35309721 
##      ptratio        black        lstat         medv 
##  -2.14921205  29.28922765  -8.56306288  22.23853955
```

From comparing the $Group_1$and the rest, we can see that the $Group_1$has less crime rate, less proportion of non-retail business acres per town, less tax and less percentage of population. But it has more proportion of residential land zoned for lots over 25,000 sq.ft. and a larger median value of owner-occupied home in $1000s.