Bruce Campbell ST-617 HW 1

Wed Jun 29 08:36:49 2016

Chapter 2

Problem 7

```
library("knitr")
library("pander")
X = matrix(data = c(0, 2, 0, 0, -1, 1, 0, 3, 0, 1, 1, 0, 1, 0, 0, 0, 3, 2, 1,
    1, 0), nrow = 7, ncol = 3)

pander(X)
```

0	3	0
2	0	0
0	1	3
0	1	2
-1	0	1
1	1	1
0	0	0

```
D <- as.matrix(dist(X, method = "euclidean", diag = FALSE, upper = FALSE, p = 2))

# Adding class labels
DF <- cbind(X, c("Red", "Red", "Red", "Green", "Green", "Green", "UNK"))
```

This is the distance matrix

```
pander(D)
```

1	2	3	4	5	6	7
0	3.606	3.606	2.828	3.317	2.449	3
3.606	0	3.742	3	3.162	1.732	2
3.606	3.742	0	1	2.449	2.236	3.162
2.828	3	1	0	1.732	1.414	2.236
3.317	3.162	2.449	1.732	0	2.236	1.414
2.449	1.732	2.236	1.414	2.236	0	1.732
3	2	3.162	2.236	1.414	1.732	0

And the distances from the test points to the training points is

```
testDist <- D[7, -7]
pander(testDist)
```

1	2	3	4	5	6
3	2	3.162	2.236	1.414	1.732

When K=1 the distance to the nearest neighbor is 1.4142136 - note we removed the test point.

```
index <- which.min(D[7, -7])
classLabel <- (DF[index, 4])
```

a)

The predicted label for K=1 is Green

The KNN classifier estimates the class conditional probability using K nearest neighbors as

$$P(Y = color|X = x_0) = \frac{1}{k}\sum_{N_k} I(y_i == color)$$

for K=1 this reduces to setting the color to that of the nearest neighbor - which is green.

When K=3 we have

```
Z <- sort(testDist, index.return = TRUE)

class1stNearest <- Z$ix[1]
class2dnNearest <- Z$ix[2]
class3rdNearest <- Z$ix[3]
```

b)

For K=3 the classes of the three nearest neighbors are

```
pander(c(DF[class3rdNearest, 4], DF[class2dnNearest, 4], DF[class3rdNearest,
    4]))
```

Red, *Green* and *Red*

And

$$P(Y = red|X = (0,0,0)) = \frac{1}{3}2) = \frac{2}{3}$$

$$P(Y = green|X = (0,0,0)) = \frac{1}{3}1) = \frac{1}{3}$$

So KNN with K=3 classifies the test point as red.

c)

If the optimal decision boundary is highly non-linear we would expect the best value of K to be a small number. This is because local information is lost when including large number of points. When K gets very large we average over a large area and the optimal decision boundary is effectively smoothed out. When K is small we classify based on local information and we expect KNN to preform better for highly non linear boundaries.

Chapter 2

Problem 10

a)

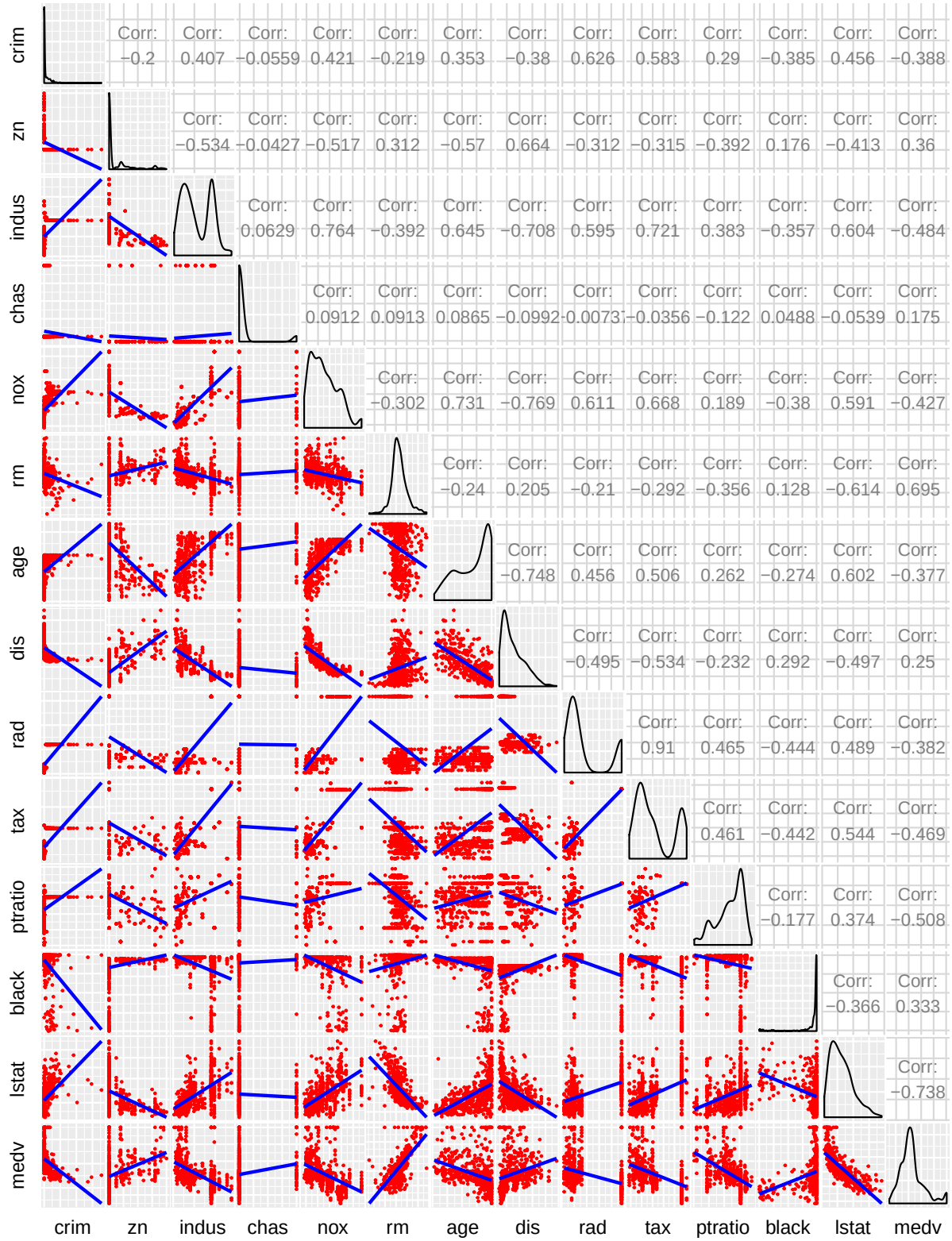
There are 506 observations in the Boston data set which contain a variety of variables purported to affect housing values in the suburbs of Boston.

This data frame contains the following columns:

- 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(Bk - 0.63)ˆ2 where Bk is the proportion of blacks by town.
- lstat : lower status of the population (percent).
- medv : median value of owner-occupied homes in $1000s

b)

```
library(MASS)
# library(ggplot2) require(GGally)
library(pander)
DF <- Boston
```



We note

- several bimodal variables : indus,rad,tax

- some linear relationships : (dis,age) (rm,mdev)
- a few mildly non linear relationships (nox,age) (nox,dis)

c)

chas has a higher crime rate at a value of 0

certain values of nox (nox >.57) are associated with a higher crime rate

older towns have a higher crime rate

Additionally there is a spike in crime at a indus value of 19

d)

The crime rate variable has a long tail.

There are 11 observations with a value higher than 25.

The tax rate appears bimodal with 137 towns having a rate above 650 and the rest below 450

There are some towns with a low pupil teacher ratio. We note that the tax rate for these towns does not fall in the upper bracket.

We use the summary function to look at the ranges of the variables

```
S <- summary(DF)
pander(S)
```

Table 4: Table continues below

crim	zn	indus	chas	nox
Min. : 0.00632	Min. : 0.00	Min. : 0.46	Min. :0.00000	Min. :0.3850
1st Qu.: 0.08204	1st Qu.: 0.00	1st Qu.: 5.19	1st Qu.:0.00000	1st Qu.:0.4490
Median : 0.25651	Median : 0.00	Median : 9.69	Median :0.00000	Median :0.5380
Mean : 3.61352	Mean : 11.36	Mean :11.14	Mean :0.06917	Mean :0.5547
3rd Qu.: 3.67708	3rd Qu.: 12.50	3rd Qu.:18.10	3rd Qu.:0.00000	3rd Qu.:0.6240
Max. :88.97620	Max. :100.00	Max. :27.74	Max. :1.00000	Max. :0.8710

Table 5: Table continues below

rm	age	dis	rad	tax
Min. :3.561	Min. : 2.90	Min. : 1.130	Min. : 1.000	Min. :187.0
1st Qu.:5.886	1st Qu.: 45.02	1st Qu.: 2.100	1st Qu.: 4.000	1st Qu.:279.0
Median :6.208	Median : 77.50	Median : 3.207	Median : 5.000	Median :330.0
Mean :6.285	Mean : 68.57	Mean : 3.795	Mean : 9.549	Mean :408.2
3rd Qu.:6.623	3rd Qu.: 94.08	3rd Qu.: 5.188	3rd Qu.:24.000	3rd Qu.:666.0
Max. :8.780	Max. :100.00	Max. :12.127	Max. :24.000	Max. :711.0

ptratio	black	lstat	medv
Min. :12.60	Min. : 0.32	Min. : 1.73	Min. : 5.00
1st Qu.:17.40	1st Qu.:375.38	1st Qu.: 6.95	1st Qu.:17.02

ptratio	black	lstat	medv
Median :19.05	Median :391.44	Median :11.36	Median :21.20
Mean :18.46	Mean :356.67	Mean :12.65	Mean :22.53
3rd Qu.:20.20	3rd Qu.:396.23	3rd Qu.:16.95	3rd Qu.:25.00
Max. :22.00	Max. :396.90	Max. :37.97	Max. :50.00

e)

```
countBoundingCharles <- nrow(DF[DF$chas == 1, ])
```

There are 35 towns identified as bordering the Charles.

f)

From the summary table above we see that the median pupil to teacher ration is 19.05.

g)

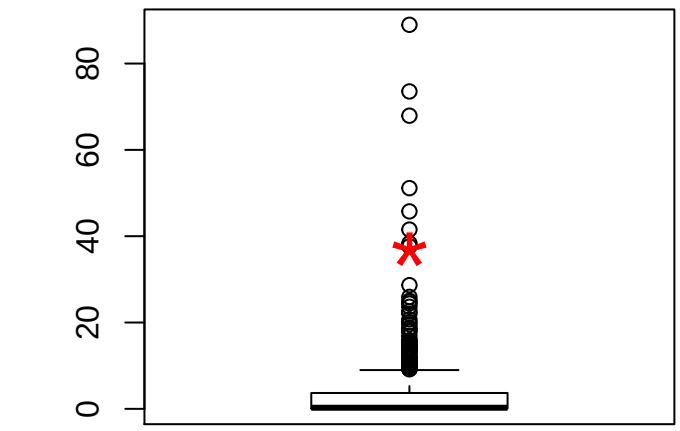
```
lowMed <- DF[which.min(DF$medv), ]
pander(lowMed)
```

Table 7: Table continues below

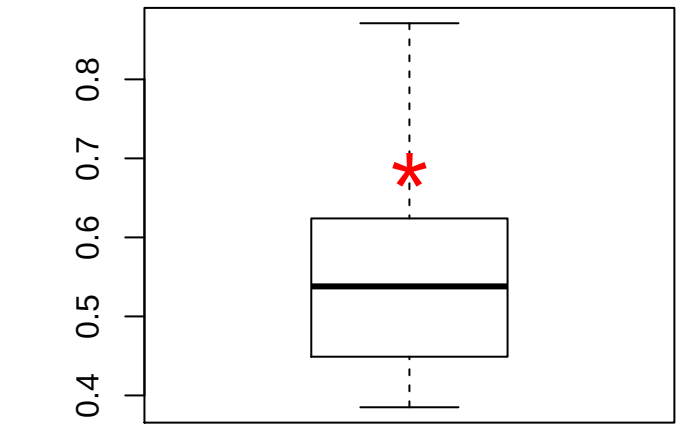
	crim	zn	indus	chas	nox	rm	age	dis	rad	tax
399	38.35	0	18.1	0	0.693	5.453	100	1.49	24	666

	ptratio	black	lstat	medv
399	20.2	396.9	30.59	5

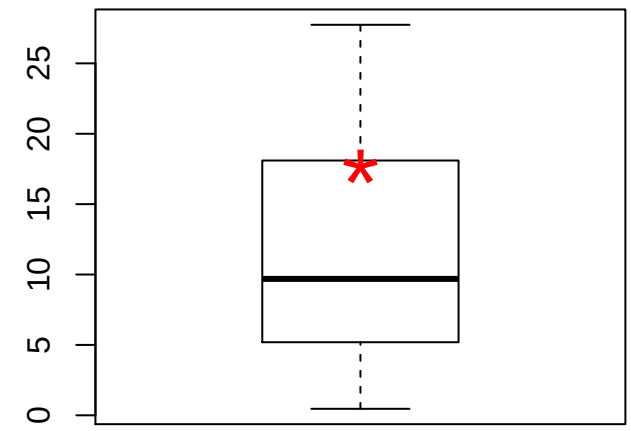
The town with the lowest median value of owner occupied houses has a high crime rate as indicated by the box plot below.

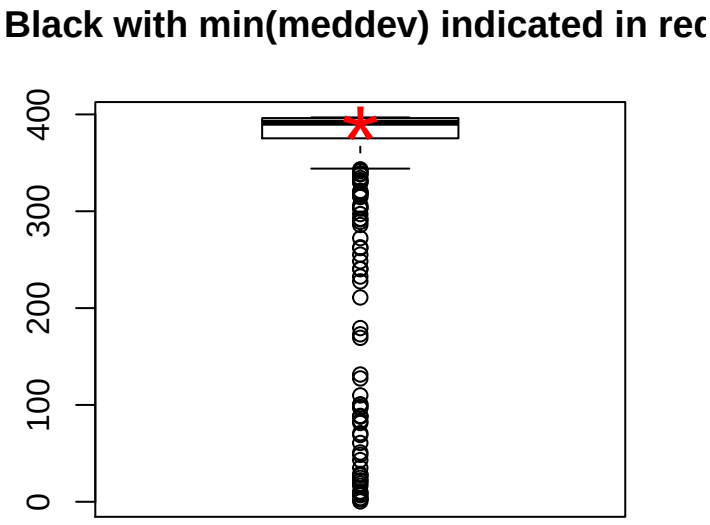


Nox is elevated for this town



Indus is elevated for this town





Black is elevated for this town

h)

```
sevenRooms <- DF[DF$rm > 7, ]

eightRooms <- DF[DF$rm > 8, ]
```

There are 64 dwellings with more than seven rooms, and there are 13 with more than eight rooms.

```
pander(eightRooms)
```

Table 9: Table continues below

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax
98	0.1208	0	2.89	0	0.445	8.069	76	3.495	2	276
164	1.519	0	19.58	1	0.605	8.375	93.9	2.162	5	403
205	0.02009	95	2.68	0	0.4161	8.034	31.9	5.118	4	224
225	0.3153	0	6.2	0	0.504	8.266	78.3	2.894	8	307
226	0.5269	0	6.2	0	0.504	8.725	83	2.894	8	307
227	0.3821	0	6.2	0	0.504	8.04	86.5	3.216	8	307
233	0.5753	0	6.2	0	0.507	8.337	73.3	3.838	8	307
234	0.3315	0	6.2	0	0.507	8.247	70.4	3.652	8	307
254	0.3689	22	5.86	0	0.431	8.259	8.4	8.907	7	330
258	0.6115	20	3.97	0	0.647	8.704	86.9	1.801	5	264
263	0.5201	20	3.97	0	0.647	8.398	91.5	2.288	5	264
268	0.5783	20	3.97	0	0.575	8.297	67	2.422	5	264
365	3.474	0	18.1	1	0.718	8.78	82.9	1.905	24	666

	ptratio	black	lstat	medv
98	18	396.9	4.21	38.7
164	14.7	388.4	3.32	50
205	14.7	390.6	2.88	50
225	17.4	385.1	4.14	44.8
226	17.4	382	4.63	50
227	17.4	387.4	3.13	37.6
233	17.4	385.9	2.47	41.7
234	17.4	378.9	3.95	48.3
254	19.1	396.9	3.54	42.8
258	13	389.7	5.12	50
263	13	386.9	5.91	48.8
268	13	384.5	7.44	50
365	20.2	354.6	5.29	21.9

Chapter 3

Problem 8

a)

```
rm(list = ls())
library(ISLR)
library(pander)
DF <- Auto
pander(names(DF))
```

mpg, *cylinders*, *displacement*, *horsepower*, *weight*, *acceleration*, *year*, *origin* and *name*

```
lm.fit <- lm(mpg ~ horsepower, data = DF)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = DF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
Yhat <- function(beta0, beta1, predictor) {
    result <- beta0 + beta1 * predictor
    return(unname(result))
}

mpg_h98 <- Yhat(lm.fit$coefficients["(Intercept)"], lm.fit$coefficients["horsepower"],
    98)
```

We note that there is a negative relationship between horsepower and mpg. When horsepower goes up mpg goes down. We note that the coefficients β_0 and β_1 are large compared to their standard errors and that the p-value of the t-statistic is very small

The confidence intervals for the regression coefficients are not too wide as seen below

```
pander(confint(lm.fit))
```

	2.5 %	97.5 %
(Intercept)	38.53	41.35
horsepower	-0.1705	-0.1452

The predicted value of mpg for a horsepower of 98 is 24.4670772 as calculated by our function above. Using the predict function we get the prediction intervals

```
predict(lm.fit, data.frame(horsepower = c(98)), interval = "prediction")
```

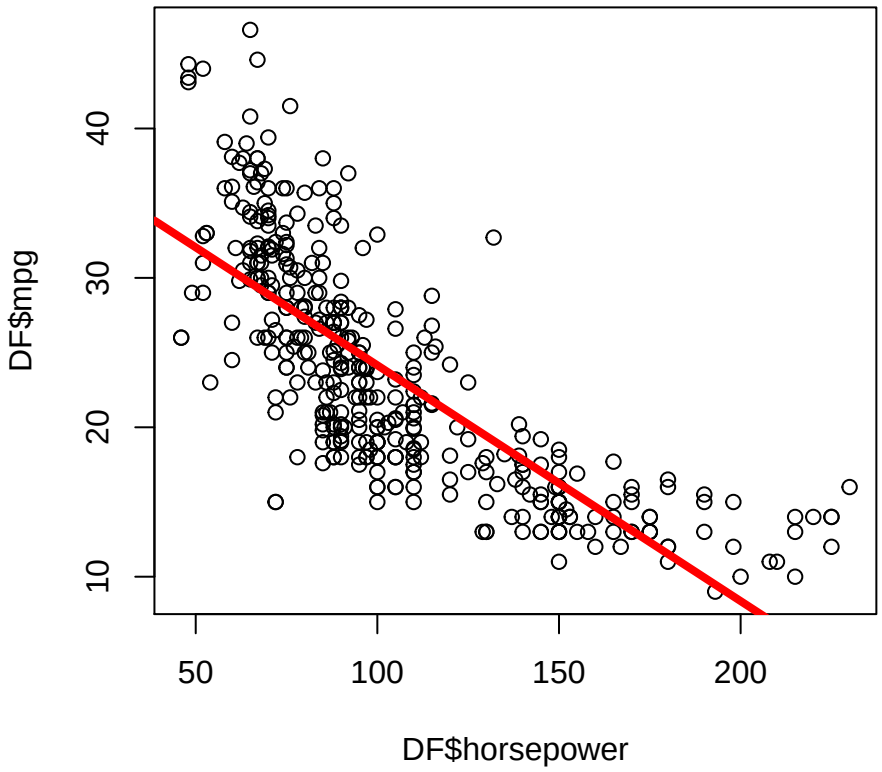
```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

and confidence intervals for this point

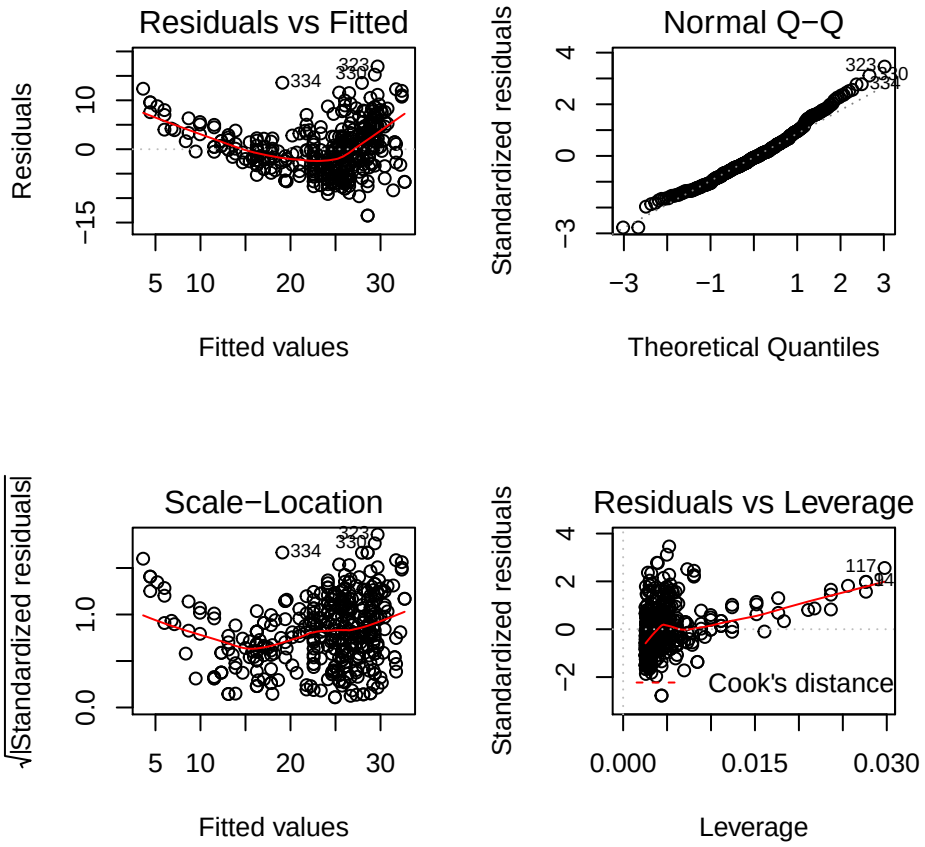
```
predict(lm.fit, data.frame(horsepower = c(98)), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

b)



c) Diagnostic plots



From the diagnostic plots we see that there are some high leverage points and that residuals are higher at the boundaries of the range of the predictor. We also note some divergence from normality for the standardized residuals.

Chapter 3

Problem 13

a)

```
rm(list = ls())

X <- rnorm(100, 0, 1)

epsilon <- rnorm(100, 0, 0.25)

Y <- -1 + 0.5 * X + epsilon
```

c)

The length of y is 100, $\beta_0 = -1$ and $\beta_1 = 0.5$

d)

We note that there is a positive linear relation between X and Y, that the midpoint of theh range of X is rougly 0 and the midpoint of the range of Y is rougly -1. We also note the dispersion of the data along the diagonal is about $\frac{1}{4}$

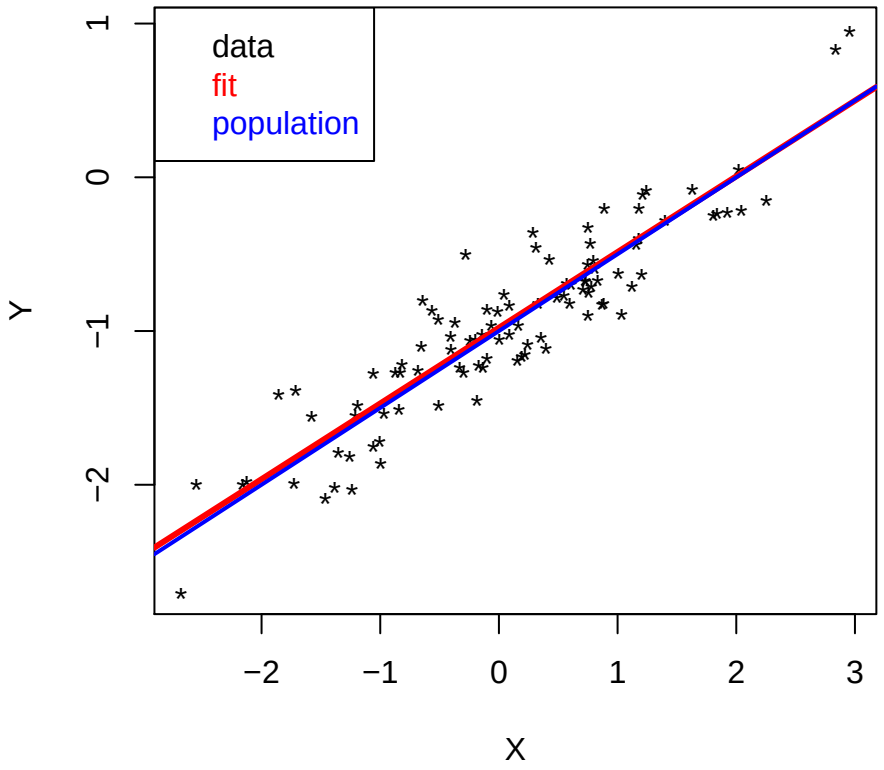
e)

```
DF <- data.frame(predictor = X, response = Y)
lm.fit <- lm(response ~ predictor, data = DF)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = response ~ predictor, data = DF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43038 -0.18153 -0.01185  0.16174  0.62539
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.97881    0.02411  -40.60   <2e-16 ***
## predictor    0.49249    0.02143   22.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2405 on 98 degrees of freedom
## Multiple R-squared:  0.8435, Adjusted R-squared:  0.8419
## F-statistic: 528.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

The estimated values of the regression coefficients are very close to the actual values of -1 and .5.

f)



g)

```
lm_poly.fit <- lm(response ~ predictor + I(predictor^2), data = DF)
summary(lm_poly.fit)
```

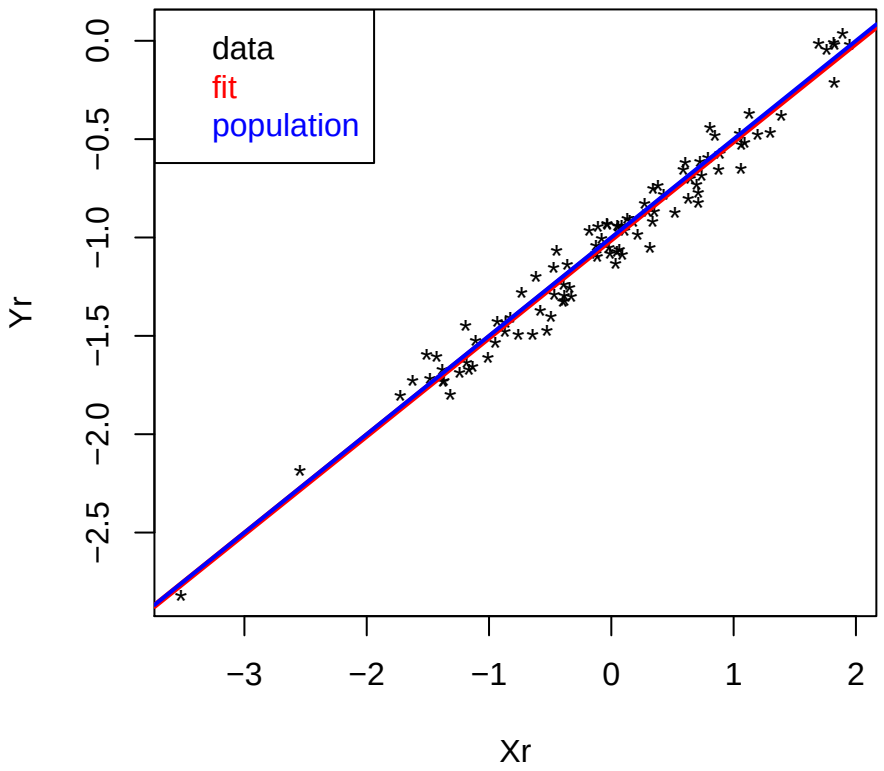
```
##
## Call:
## lm(formula = response ~ predictor + I(predictor^2), data = DF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47367 -0.18045 -0.00569  0.17278  0.63933
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.99411    0.02964 -33.534   <2e-16 ***
## predictor       0.49107    0.02151  22.829   <2e-16 ***
## I(predictor^2)  0.01218    0.01370   0.889    0.376
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2407 on 97 degrees of freedom
## Multiple R-squared:  0.8448, Adjusted R-squared:  0.8416
## F-statistic:   264 on 2 and 97 DF,  p-value: < 2.2e-16
```

There is very little evidence that adding a polynomial term to the regression has improves the fit. RSE and R squared were not markedly affected by the addition of the quadratic term. We also note the p-value of the quadratic term is high and the coefficient is near 0, reflecting it's insignificance in the model.

h) Reducing error

```
##
## Call:
## lm(formula = response ~ predictor, data = DF)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.190678 -0.070198  0.000928  0.071274  0.175254
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.009144   0.009135  -110.5   <2e-16 ***
## predictor    0.498786   0.009170    54.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09127 on 98 degrees of freedom
## Multiple R-squared:  0.9679, Adjusted R-squared:  0.9676
## F-statistic:  2959 on 1 and 98 DF,  p-value: < 2.2e-16
```



When we reduce the error in the data the median residual and RES are decreased. Mutliple R square is increased. All indicates of improved performance in the fit.

i)

Confidence interval for the first model

```
confint(lm.fit)
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.0266536 -0.9309732
## predictor    0.4499702  0.5350148
```

Confidence interval for the second model

```
confint(lm_r.fit)
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.0272721 -0.9910157
## predictor    0.4805894  0.5169829
```

As expected our confidence interval in the second model is smaller than the first.

Chapter 4

Problems 6 and 9

Problem 6

For a logistic regression model we fit

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2}}$$

```
beta_0 = -6
beta_1 = 0.05
beta_2 = 1

x_1 = 40
x_2 = 3.5

P_x <- function(beta_0, beta_1, beta_2, x_1, x_2) {
    e_x_dot_y <- exp(beta_0 + beta_1 * x_1 + beta_2 * x_2)
    result = e_x_dot_y/(1 + e_x_dot_y)
    return(unname(result))
}

prob_A = P_x(beta_0, beta_1, beta_2, x_1, x_2)
```

If a student studies 40 hours and has a GPA of 3.5 the estimated probability of getting an A is 0.3775407

b) Calculate the number of hours of study needed to have a 50% chance of getting an A

Let $\alpha = \beta_0 + \beta_2 * x_2 = -2.5$ and treat x_1 as an unknown in the log-odds equation

$$log\left(\frac{P(X)}{1-P(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

Which simplifies to $x_1 = \frac{2}{3} * \frac{1}{0.05} = 13.33$ when we put it 3.5 for x_2 and 0.5 for $P(X)$

Problem 9

a)

We use the definition of odds

$$odds = \frac{P(X)}{1-P(X)}$$

to calculate the probability. If the odds are .37 then $P(X) = \frac{.37}{1.37} = .27$ So the fraction of people defaulting with an odds of .37 is .27

b)

If someone has a 16% chance of default on a credit card payment then the associated odds is $\frac{.16}{(1-.16)} = .19$.