Name: Loh Jing Wei

Student ID: 30856183

FIT2086

Assignment 3

Question 1

1.1

Predictors possibly associated with fuel efficiency: *Eng.Displacement, AspirationSC, AspirationTC, AspirationTS, No.Gears, Lockup.Torque.ConverterY, Drive.SysF.*

Explanation: These variables have p-value <0.05.

p-values of <0.05 mean that the chance of seeing an association just by chance, if there was no association at the population level, is lower than 5% which is unlikely. Hence it is strong evidence against null that the coefficient for these variables is 0.

3 variables which appear to be the strongest predictors: *Eng.Displacement, AspirationTC, Drive.SysF.*

Explanation: There are 4 variables with p-value < 0.001, they are Eng.Displacement, AspirationTC, Drive.SysF and No.Gears. However No.Gears has higher p-value than the other 3 variables. Hence the 3 variables picked have the smallest p-value among the other variables.

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              -2.167e+02  1.550e+02  -1.398 0.162706
Model.Year                1.156e-01  7.679e-02   1.505 0.132856
Eng.Displacement         -1.331e+00  1.861e-01  -7.151 3.22e-12 ***
No.Cylinders              5.730e-03  1.206e-01   0.048 0.962117
AspirationOT             -1.034e-01  1.240e+00  -0.083 0.933569
AspirationSC             -7.990e-01  4.064e-01  -1.966 0.049842 *
AspirationTC             -1.217e+00  2.201e-01  -5.528 5.31e-08 ***
AspirationTS             -1.351e+00  6.720e-01  -2.010 0.044935 *
No.Gears                 -1.940e-01  5.158e-02  -3.760 0.000191 ***
Lockup.Torque.ConverterY -5.621e-01  1.974e-01  -2.847 0.004602 **
Drive.SysA                6.138e-02  2.706e-01   0.227 0.820624
Drive.SysF                1.535e+00  2.930e-01   5.239 2.41e-07 ***
Drive.SysP               -9.766e-01  5.639e-01  -1.732 0.083967 .
Drive.SysR                2.081e-01  2.551e-01   0.816 0.415071
Max.Ethanol              -8.956e-03  6.100e-03  -1.468 0.142704
Fuel.TypeGM               8.096e-01  1.004e+00   0.806 0.420647
Fuel.TypeGP               4.064e-01  2.425e-01   1.676 0.094372 .
Fuel.TypeGPR              8.418e-02  2.458e-01   0.343 0.732106
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

1.2

4 of our predictors have a p-value small enough to pass the Bonferroni threshold of 0.05/17 = 0.002941176.

Using the Bonferroni procedure, the predictors possibly associated with fuel efficiency are **Eng.Displacement, AspirationTC, No.Gears, Drive.SysF.**

```
> sum(pvalues < 0.05/17)
[1] 4
```

1.3

Coefficient for variable Eng.Displacement is -1.330991

- **The expected value of target (fuel efficiency) decreased by 1.330991 per unit increase in variable Eng.Displacement.** In other words, for every additional litre gain by Engine Displacement, the mean fuel efficiency decreases by 1.330991 km/l.

Coefficient for variable Drive.SysF is 1.535284

- **The expected value of target (fuel efficiency) of car with a Front-wheel drive system is 1.535284 km/l higher than car with 4-wheel drive system.**

```
> # coefficient for variable Eng.Displacement is -1.330991
> fullmod$coefficients[[3]]
[1] -1.330991
> # coefficient for variable Drive.SysF is 1.535284
> fullmod$coefficients[[12]]
[1] 1.535284
```

1.4

Regression equation

$\widehat{Comb.FE}$ = 16.3611935 - 1.3164709*Eng.Displacement + 0.1336877*AspirationOT - 0.5706151*AspirationSC - 1.0717452*AspirationTC - 1.3248883*AspirationTS - 0.1747731*No.Gears - 0.5731985*Lockup.Torque.ConverterY + 0.1934040*Drive.SysA + 1.5475411*Drive.SysF - 1.0801801*Drive.SysP + 0.2842354*Drive.SysR

```
  (Intercept)      Eng.Displacement          AspirationOT          AspirationSC
   16.3611935            -1.3164709             0.1336877            -0.5706151
AspirationTC         AspirationTS              No.Gears Lockup.Torque.ConverterY
   -1.0717452            -1.3248883            -0.1747731            -0.5731985
   Drive.SysA            Drive.SysF            Drive.SysP            Drive.SysR
    0.1934040             1.5475411            -1.0801801             0.2842354
```

1.5(a)

```
> fuel_efficiency[33,]
   Model.Year Eng.Displacement No.Cylinders Aspiration No.Gears Lockup.Torque.Converter Drive.Sys
33       2019              1.4            4         TC        6                       Y         F
   Max.Ethanol Fuel.Type  Comb.FE
33          10         G 14.46861
```

Mean fuel efficiency of the new car = 16.3611935 - 1.3164709*1.4 - 1.0717452*1 - 0.1747731*6 - 0.5731985*1 + 1.5475411*1 = 13.37209km/l

We can use predict and set interval parameter as “confidence” to find the confidence interval

```
> y_hat = predict(step.fit.bic,fuel_efficiency[33,],interval="confidence")
> y_hat
         fit      lwr      upr
33 13.37209 12.99409 13.75009
```

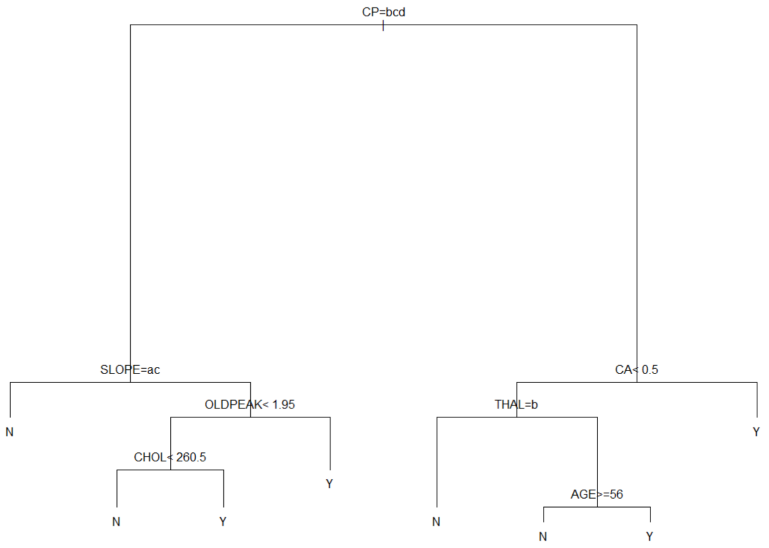
Confidence interval is (12.99409, 13.75009)

1.5(b)

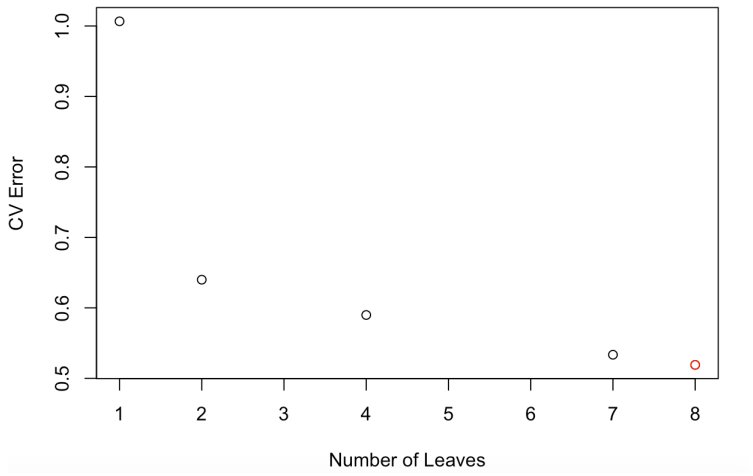
The current car has a mean fuel efficiency of 11km/l, whereas the **new car has a mean fuel efficiency of 13.37209km/l.** The model suggests that the new car has **better fuel efficiency** than the current car.

Question 2

2.1

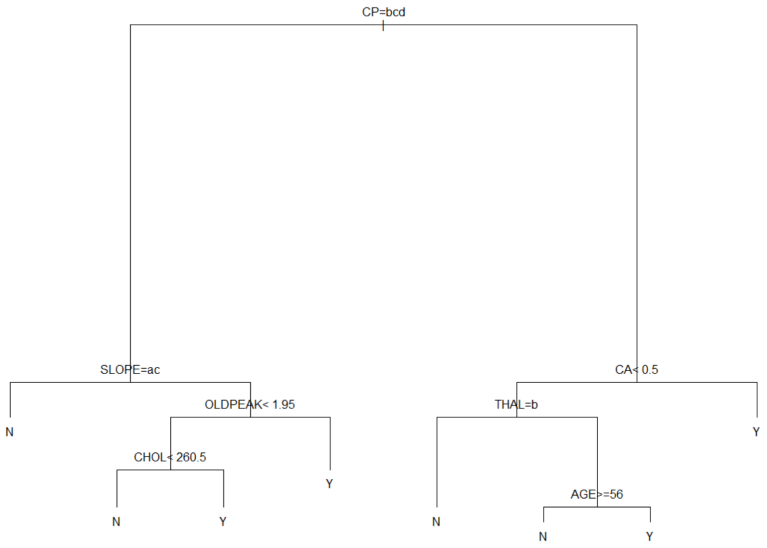


The fitted decision tree shows that there are 8 leaf nodes and 7 variables, the variables use are: *CP, SLOPE, OLDPEAK, CHOL, CA, THAL, AGE*



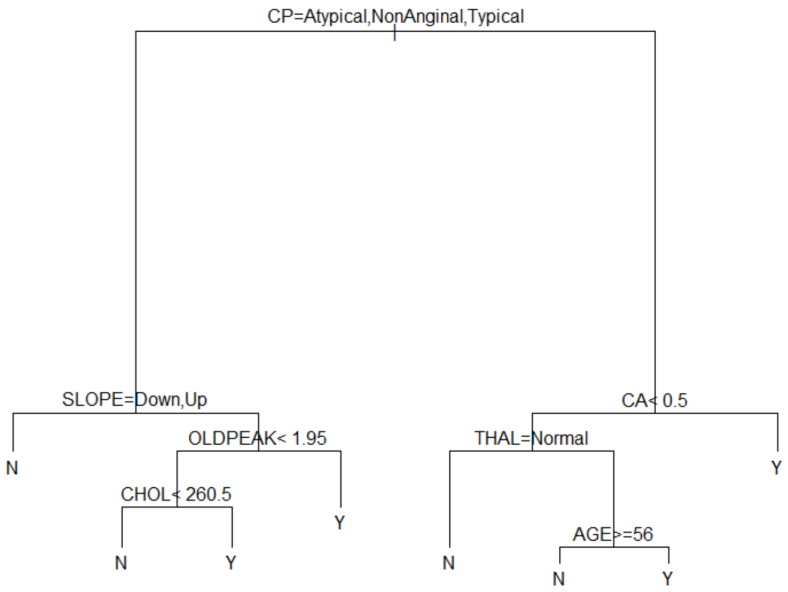
The **CV plot** shows that the **best tree size (appropriate size) is around 8.** The tree should have 8 leaf nodes

```
1) root 210 99 N (0.52857143 0.47142857)
  2) CP=Atypical,NonAnginal,Typical 106 23 N (0.78301887 0.21698113)
    4) SLOPE=Down,Up 69  6 N (0.91304348 0.08695652) *
    5) SLOPE=Flat 37 17 N (0.54054054 0.45945946)
     10) OLDPEAK< 1.95 30 10 N (0.66666667 0.33333333)
       20) CHOL< 260.5 21  3 N (0.85714286 0.14285714) *
       21) CHOL>=260.5 9  2 Y (0.22222222 0.77777778) *
     11) OLDPEAK>=1.95 7  0 Y (0.00000000 1.00000000) *
  3) CP=Asymptomatic 104 28 Y (0.26923077 0.73076923)
    6) CA< 0.5 47 23 N (0.51063830 0.48936170)
     12) THAL=Normal 21  4 N (0.80952381 0.19047619) *
     13) THAL=Fixed.Defect,Reversible.Defect 26  7 Y (0.26923077 0.73076923)
       26) AGE>=56 8  3 N (0.62500000 0.37500000) *
       27) AGE< 56 18  2 Y (0.11111111 0.88888889) *
    7) CA>=0.5 57  4 Y (0.07017544 0.92982456) *
```



The **CV pruned decision tree** shows that there are **8 leaf nodes and 7 variables**, the variables use are: *CP, SLOPE, OLDPEAK, CHOL, CA, THAL, AGE*

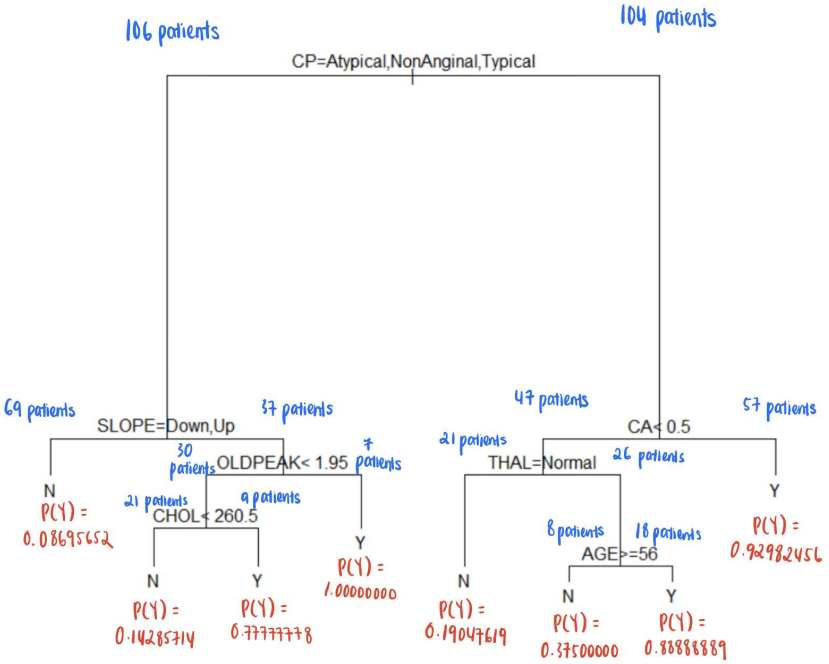
2.2



The conditions required for the tree to predict that someone has heart disease is as following:

- **CP** (Chest pain type) is *Atypical, NonAnginal or Typical*, **SLOPE** (Slope of the peak exercise ST segment) is *Flat*, **OLDPEAK** (Exercise induced ST depression relative to rest) is *greater than or equal to 1.95*
- **CP** (Chest pain type) is *Atypical, NonAnginal or Typical*, **SLOPE** (Slope of the peak exercise ST segment) is *Flat*, **OLDPEAK** (Exercise induced ST depression relative to rest) is *greater than or equal to 1.95,* **CHOL** (Serum cholesterol in mg/dl) is *greater than or equal to 260.5mg/dl.*
- **CP** (Chest pain type) is *Asymptomatic pain*, **CA** (Number of major vessels colored by fluoroscopy) is *less than 0.5*, **THAL** (Thallium scanning results) is *Fixed.Defect or Reversible.Defect*, **AGE** (Age of patient in years) is *less than 56.*
- **CP** (Chest pain type) is *Asymptomatic pain*, **CA** (Number of major vessels colored by fluoroscopy) is *greater than or equal to 0.5*.

2.3


106 patients
104 patients
CP=Atypical,NonAnginal,Typical
69 patients
37 patients
47 patients
57 patients
SLOPE=Down,Up
30 patients
OLDPEAK< 1.95
7 patients
21 patients
CA< 0.5
26 patients
THAL=Normal
N
P(Y) = 0.08695652
21 patients
9 patients
CHOL< 260.5
Y
P(Y) = 1.00000000
8 patients
18 patients
AGE>=56
Y
P(Y) = 0.92982456
N
P(Y) = 0.14285714
Y
P(Y) = 0.77777778
N
P(Y) = 0.19047619
N
P(Y) = 0.37500000
Y
P(Y) = 0.88888889

2.4

Predictor combination of **CP (Chest pain type) = Atypical, NonAnginal or Typical** and **SLOPE (Slope of the peak exercise ST segment) = Down, Up** result in lowest probability of having heart disease.

The lowest probability of having heart disease is 0.08695652

2.5

Logistic regression model without pruning to the heart data:

```
Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)          -4.354772   3.297064  -1.321 0.186567
AGE                  -0.036849   0.029331  -1.256 0.208989
SEXM                  1.424438   0.635509   2.241 0.024999 *
CPAtypical           -1.696740   0.784841  -2.162 0.030627 *
CPNonAnginal         -2.318935   0.623722  -3.718 0.000201 ***
CPTypical            -2.166047   0.806103  -2.687 0.007208 **
TRESTBPS              0.024926   0.013042   1.911 0.055971 .
CHOL                  0.007278   0.004641   1.568 0.116818
FBS>120              -0.851375   0.883442  -0.964 0.335195
RESTECGNormal        -0.683555   0.470289  -1.453 0.146091
RESTECGST.T.Wave      0.086449   3.144500   0.027 0.978067
THALACH              -0.011682   0.014004  -0.834 0.404185
EXANGY                0.615957   0.526106   1.171 0.241686
OLDPEAK               0.546550   0.285426   1.915 0.055511 .
SLOPEFlat             1.534432   1.087477   1.411 0.158244
SLOPEUp               0.052141   1.171927   0.044 0.964513
CA                    1.215978   0.328943   3.697 0.000218 ***
THALNormal            0.852195   1.119830   0.761 0.446655
THALReversible.Defect 1.712748   1.063607   1.610 0.107328
```

Logistic regression model after pruning using stepwise selection with the KIC score:

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -7.248413   2.232140  -3.247 0.001165 **
SEXM          1.802856   0.533646   3.378 0.000729 ***
CPAtypical   -2.184817   0.703118  -3.107 0.001888 **
CPNonAnginal -2.599144   0.558932  -4.650 3.32e-06 ***
CPTypical    -2.369844   0.753460  -3.145 0.001659 **
TRESTBPS      0.021501   0.011787   1.824 0.068140 .
CHOL          0.008167   0.004200   1.944 0.051854 .
OLDPEAK       0.581819   0.260840   2.231 0.025710 *
SLOPEFlat     1.931508   0.994042   1.943 0.052006 .
SLOPEUp       0.206602   1.086994   0.190 0.849257
CA            1.074811   0.285071   3.770 0.000163 ***
---
```

There are **10 variables used in the final logistic regression model**, the variables are: *SEXM, CPATYPICAL, CPNonAnginal, CPTypical, TRESTBPS, CHOL, OLDPEAK, SLOPEFLAT, SLOPEUP and CA*

From question 2.1 we know that there are **7 variables used in the CV pruned decision tree,** the variables are: *CP, SLOPE, OLDPEAK, CHOL, CA, THAL, AGE.*

As compared to the CV pruned decision tree, the logistic regression model has additional variables such as SEX and TRESTBPS.

As **compared to the** logistic regression model**,** the CV pruned decision tree has additional variables such as THAL and AGE.

The **most important predictor for logistic regression model** is *CPNonAnginal* with the lowest p-value of $3.32x10^{-6}$

2.6

Regression equation

$P(HD = Y)$ **=** -7.248413 + 1.802856*SEXM - 2.184817*CPAtypical - 2.599144*CPNonAnginal - 2.369844*CPTypical + 0.021501*TRESTBPS + 0.008167*CHOL + 0.581819*OLDPEAK + 1.931508*SLOPEFlat + 0.206602*SLOPEUp + 1.074811*CA

2.7

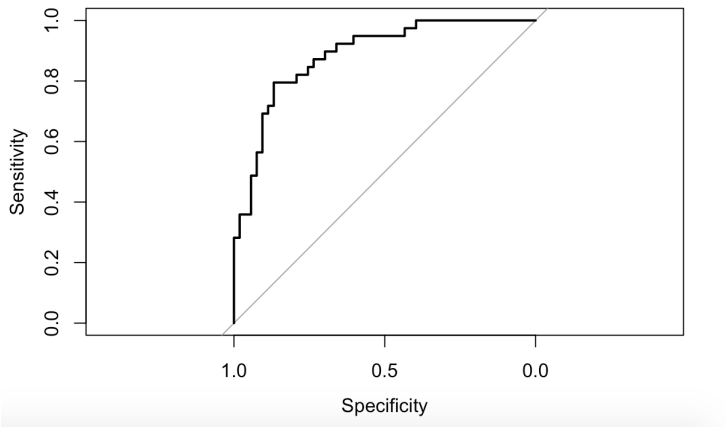
Coefficient for variable CA is 1.074811

- There is a **positive relation between CA and probability of patients having heart disease.**
- The probability of patients having heart disease increased when CA (number of major vessels coloured by fluoroscopy) increased.

2.8

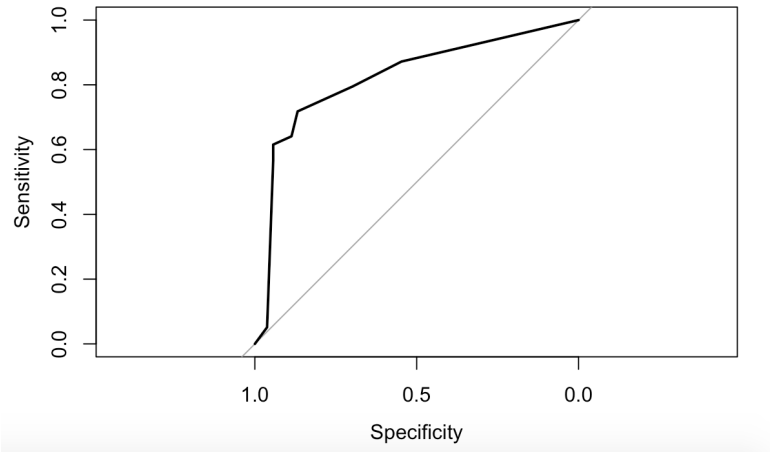
Performance statistic for **logistic regression model** using stepwise selection with the KIC score

```
------------------------------------------------------------------------------
Performance statistics:

Confusion matrix:

    target
pred  N  Y
   N 45  8
   Y  8 31

Classification accuracy = 0.826087
Sensitivity             = 0.7948718
Specificity             = 0.8490566
Area-under-curve        = 0.8853411
Logarithmic loss        = 39.43705

------------------------------------------------------------------------------
```



Performance statistic for **CV pruned tree**

```
--------------------------------------------------------------------------------
Performance statistics:

Confusion matrix:

    target
pred  N  Y
   N 47 14
   Y  6 25

Classification accuracy = 0.7826087
Sensitivity             = 0.6410256
Specificity             = 0.8867925
Area-under-curve        = 0.8214804
Logarithmic loss        = 87.37257

--------------------------------------------------------------------------------
```



Sensitivity of

- Logistic regression model: 0.7948718
- CV pruned tree: 0.6410256

The sensitivity of the models represent the ability to detect people with heart disease. The logistic regression model has sensitivity of 0.7948718, whereas the CV pruned tree has sensitivity of 0.6410256. **Logistic regression has better sensitivity** than CV pruned trees.

Specificity of

- Logistic regression model: 0.8490566
- CV pruned tree: 0.8867925

The specificity of the models represent the ability to detect non-diseased people.
The logistic regression model has specificity of 0.8490566, whereas the CV pruned tree has sensitivity of 0.8867925. **CV pruned tree has better specificity** than logistic regression model

Overall classification accuracy of

- Logistic regression model: 0.826087
- CV pruned tree: 0.7826087

The logistic regression model has overall classification accuracy of 0.826087, whereas the CV pruned tree has overall classification accuracy of 0.7826087. **Logistic regression has better overall classification accuracy** than CV pruned trees. Hence, the logistic regression model estimates the probabilities of being diseased/undiseased better than the tree.

Area-under-curve (AUC) of

- Logistic regression model: 0.8853411
- CV pruned tree: 0.8214804

The **logistic regression model has a higher AUC** than the CV pruned tree where the curve bends further away from the diagonal line, as shown on the plot below.

Overall, the logistic regression model is better because it has a higher sensitivity, overall classification accuracy and AUC, even though it has a lower specificity as compared to the tree. The sensitivity has greater importance than specificity when diagnosing the patients with heart disease. When diagnosing a patient with heart disease, making a false negative (due to low sensitivity) can have critical effects - for example, if we misidentify a patient as not having heart disease, we may not treat the patient in time and allow the disease to worsen. However, making a false positive (due to low specificity) might just get the patient another round of check up.

2.9

```
> predict(cv.tree.hd$best.tree,heart.test[10,])
            N         Y
10 0.07017544 0.9298246
```

The **conditional probability of having heart disease predicted by the tree** is 0.9298246.

The **conditional probability of not having heart disease predicted by the tree** is 0.07017544.

```
> predict(kic.hd,heart.test[10,])
      10
7.597887
```

The **conditional probability of having heart disease predicted by the logistic regression model** is 0.9994987.

The **conditional probability of not having heart disease predicted by the logistic regression model** is 1 - 0.9994987 = 0.0005013

The **odds having heart disease** are defined as

$$O = \frac{P(HD=Y)}{P(HD=N)}$$

The **odds** of 10th patient having heart disease using the conditional probabilities **predicted by the tree** is 13.25.

$$O = \frac{0.9298246}{0.07017544} = 13.25$$

The **odds** of 10th patient having heart disease using the conditional probabilities **predicted by the logistic regression model** is 1993.813.

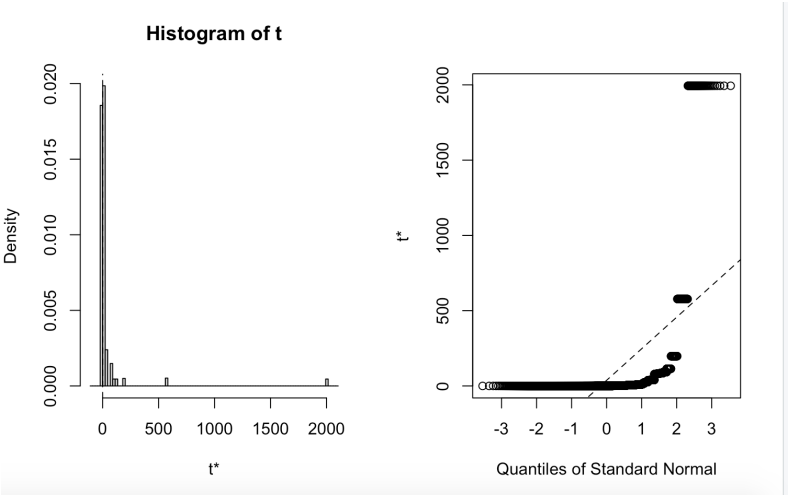
$$O = \frac{0.9994987}{0.0005013} = 1993.813$$

2.10

The Confidence interval between 65th and 66th patients have minor differences, most of the CI overlapped between the two patients. The minor differences are probably caused by the randomness of the bootstrap procedure. Hence, I believe there is not enough evidence to suggest a real difference in odds of having heart disease between the two patients.

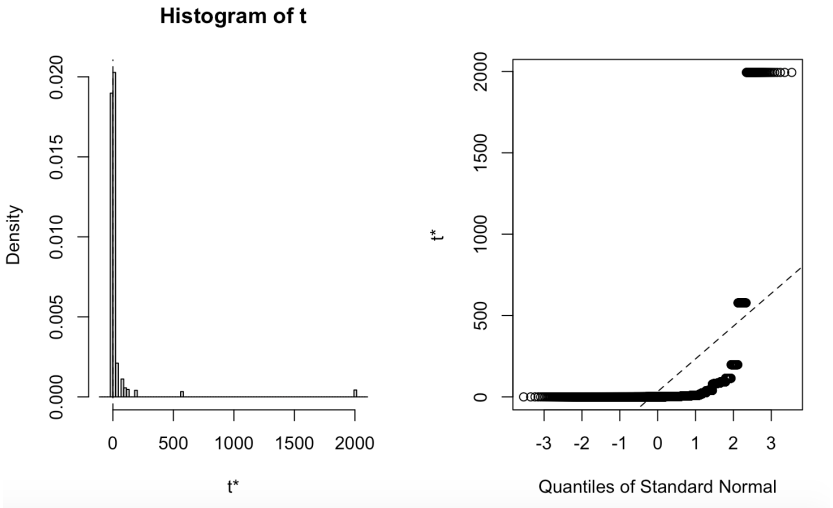
The 95% CI for 65th patient using bootstrap procedure is (0.0025, 85.0735)

```
> boot.ci(bs,conf=0.95,type="bca")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = bs, conf = 0.95, type = "bca")

Intervals :
Level       BCa
95%   ( 0.0025, 85.0735 )
Calculations and Intervals on Original Scale
```



The 95% CI for 66th patient using bootstrap procedure is (0.0104, 92.1059)

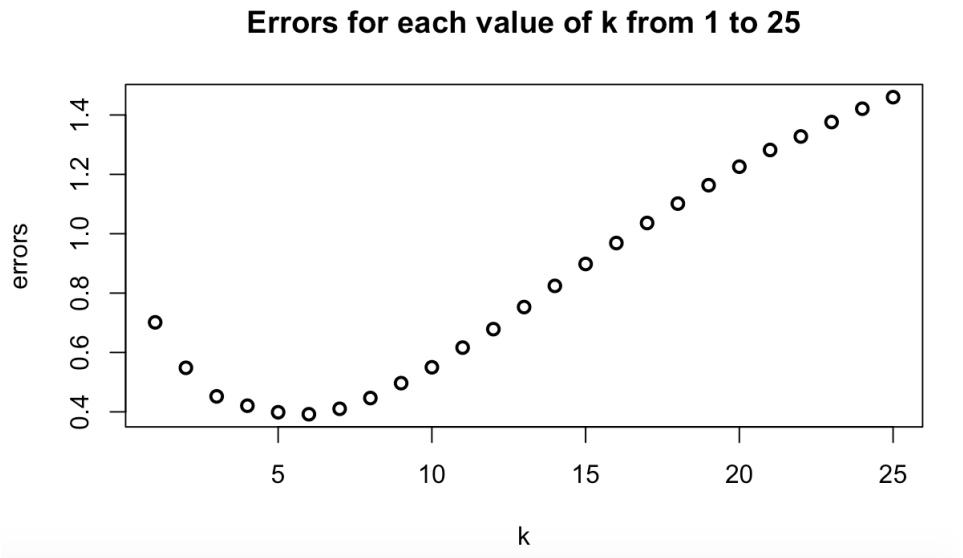
```
> boot.ci(bs,conf=0.95,type="bca")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = bs, conf = 0.95, type = "bca")

Intervals :
Level       BCa
95%   ( 0.0104, 92.1059 )
Calculations and Intervals on Original Scale
```



Question 3

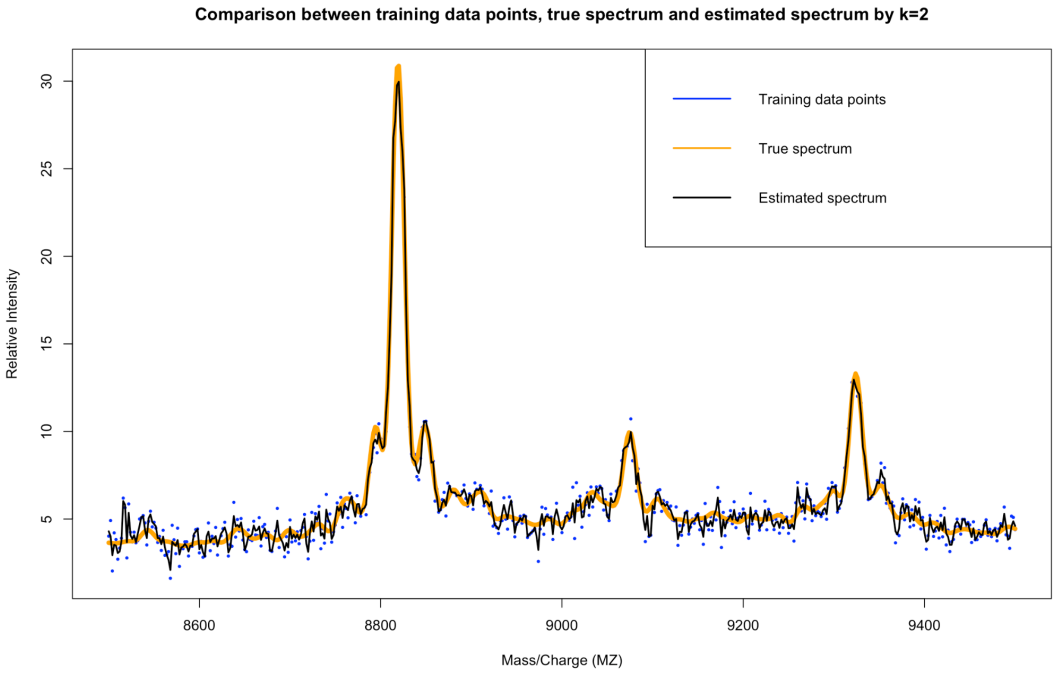
3.1

Plot of errors against the various value of k

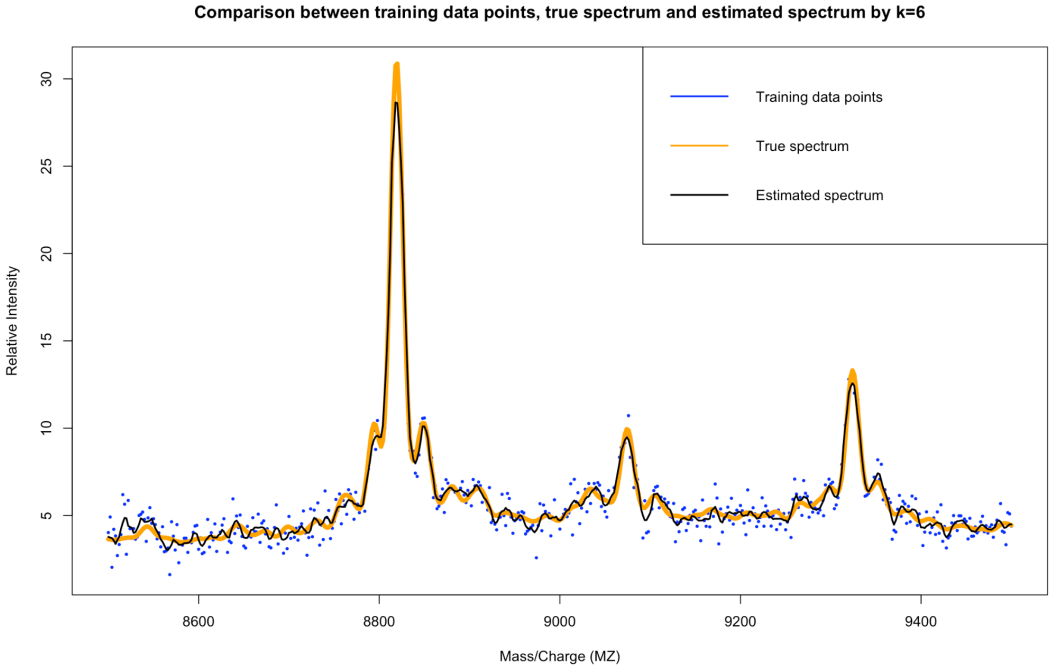


3.2

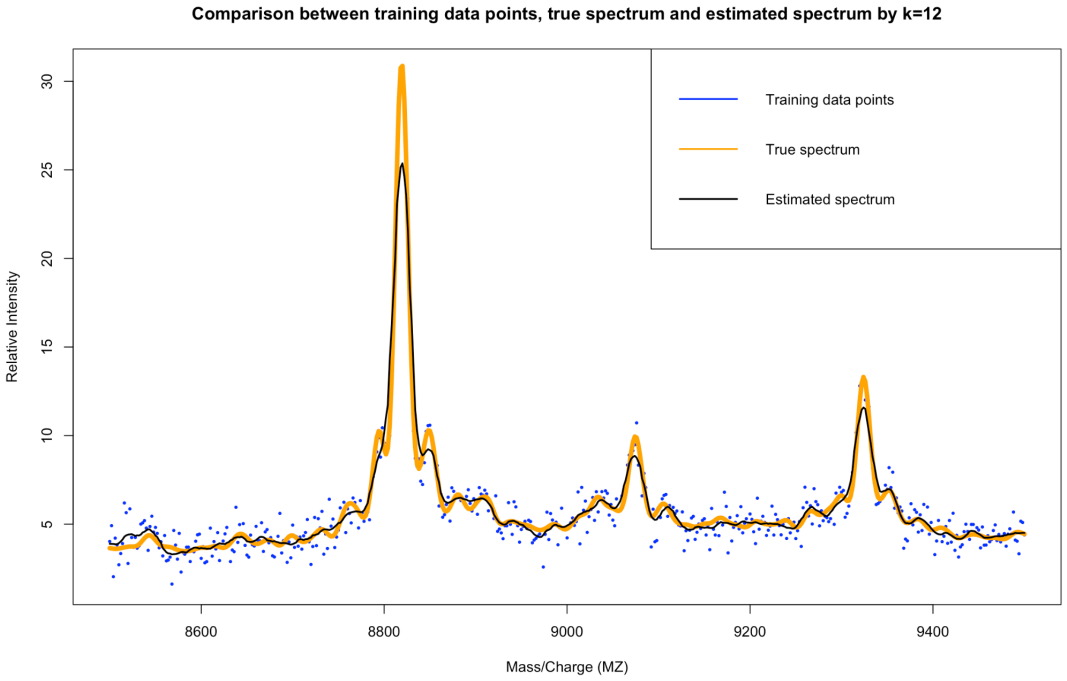
Graph for k = 2



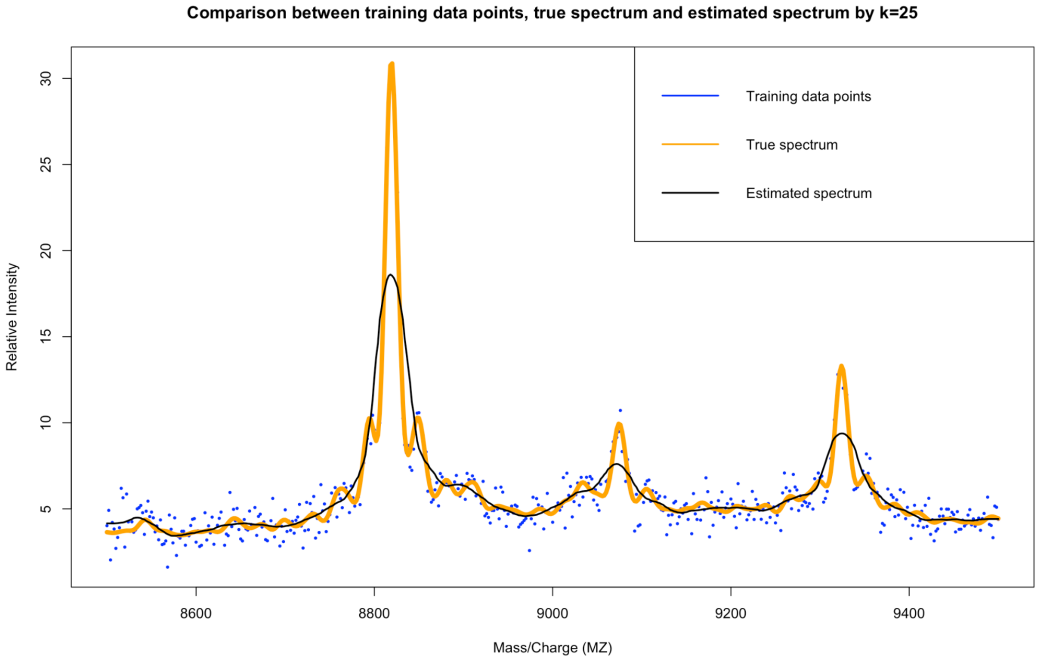
Graph for k = 6



Graph for k = 12



Graph for k = 25



3.3

The k option in knn controls the size of neighbourhood used when making predictions, i.e., how many individuals from the training data are used to form a prediction on the tests data.

Qualitatively:

From the 4 graphs we plotted in 3.2

- When k=2, the estimated spectrum (black line) tried to fit exactly to the training data points (blue dots). However, this is because the model learnt the noise and random variation of the training data (such as the background noise). Hence, when we compare the estimated spectrum with the true spectrum, we can see that it is more jagged than the true spectrum/has more peaks than the true spectrum (orange line). As a result, this model has a poor ability to predict new, unseen data and is likely overfit.
- When k=6, the estimated spectrum (black line) follows the underlying trend of the training data, it does not try to fit every single point of the training data. The black line (estimated spectrum) has roughly the same shape as the orange line (true spectrum), as well as having many peaks with similar heights as the orange line. The model using k=6 has a good estimate of the spectrum.
- When k=12, the estimated spectrum does not have the same value of the highest peak as the true spectrum. The black line (estimated spectrum) has roughly the same shape as the orange line (true spectrum), however the heights of the peak are slightly lower as compared to the orange line. This model is likely underfitting.
- When k=25, the estimated spectrum did not succeed in replicating the shape of the true spectrum. It is much less jagged than the true spectrum/has much fewer peaks than the true spectrum. This model is underfitting and will lead to systemic errors when predicting new data.

We can tell from all 4 graphs that as the k value increases, the model gets simpler and tends to underfit. When k value decreases, the model is more complex and tends to overfit.

Quantitatively:

From the plot of errors (root mean squared error) against the various values of k (k from 1 to 25), we can see that varying k-value changes the errors of prediction. When k is around 6, we have the lowest error when estimating the intensity and varying k-value changes the errors of prediction.

```
> sqrt(mean((fitted(kknn(intensity~., ms.train, ms.test, kernel="optimal", k=2) ) - ms.test$intensity)^2))
[1] 0.54835
> sqrt(mean((fitted(kknn(intensity~., ms.train, ms.test, kernel="optimal", k=6) ) - ms.test$intensity)^2))
[1] 0.3919784
> sqrt(mean((fitted(kknn(intensity~., ms.train, ms.test, kernel="optimal", k=12) ) - ms.test$intensity)^2))
[1] 0.6786136
> sqrt(mean((fitted(kknn(intensity~., ms.train, ms.test, kernel="optimal", k=25) ) - ms.test$intensity)^2))
[1] 1.460003
```

From the RMSE values for k = 2, 6, 12, 25, we can see that when k=6 the RMSE value is lowest.

3.4

The estimated spectrum plotted in 3.2 with **k=6** achieves our dual aims of providing a smooth low-noise estimate of background level as well as accurate estimation of the heights of the peaks. Hence, I believe that the kNN method is able to achieve this aim.

3.5

```
> knn$best.parameters$k
[1] 5
```

Using the **cross validation functionality** in the kknn package, we estimate the **best value of k is 5.** Whereas **using the plot computed in 3.1** using the actual mean-squared error, we estimate the **best value of k to be 6.**

3.6

The formula for sample standard deviation is given by:

Formula

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N-1}}$$

s = sample standard deviation

N = the number of observations

x_i = the observed values of a sample item

$\overline{x}$ = the mean value of the observations

First, we find the mean error when we use k=5.
Then find the errors for each individual, by finding the difference between the estimated data set and the ms.measured.2022.csv dataset.
By calculating the sum of squares of these errors minus the mean error we found, dividing it by the number of individuals/observations minus by 1, we can find the standard deviation of sensor noise.
The standard deviation of sensor/measurement noise is 0.5906997.

```
> ytest.hat = fitted(kknn(intensity ~ .,ms.train, ms.test,
+                         kernel = "optimal", k = 5) )
> mean.diff = mean(ytest.hat - ms.train$intensity)
> actual.dff = ytest.hat-ms.train$intensity
> # Standard deviation of sensor/measurement noise
> sqrt(sum((actual.dff-mean.diff)^2)/(501-1))
[1] 0.5906997
```

3.7

The value of MZ which corresponds to the maximum estimated intensity is **8818**

```
> max_index_estimated = which.max(ytest.hat)
> max_MZ_estimated = ms.train$MZ[max_index]
> max_MZ_estimated
[1] 8818
```

3.8

In Question 3.7, we obtained the MZ value corresponding to the highest intensity which is 8818 (Indice = 160). In Question 3.5, we determined that the best value of k using the kknn package is 5.

When k = 3, the 95% confidence interval for the k-nearest neighbours estimate of intensity at a MZ value = 1181 is **(26.34, 30.66)**.

```
> boot.ci(bs_intensity,conf=0.95,type="bca")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = bs_intensity, conf = 0.95, type = "bca")

Intervals :
Level       BCa
95%   (26.34, 30.66 )
Calculations and Intervals on Original Scale
```

When k = 5 (from question 3.5), the 95% confidence interval for the k-nearest neighbours estimate of intensity at a MZ value = 1181 is **(25.39, 30.57).**

```
> boot.ci(bs_intensity,conf=0.95,type="bca")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = bs_intensity, conf = 0.95, type = "bca")

Intervals :
Level       BCa
95%   (25.39, 30.57 )
Calculations and Intervals on Original Scale
```

When k = 20, the 95% confidence interval for the k-nearest neighbours estimate of intensity at a MZ value = 1181 is **(15.28, 26.33).**

```
> boot.ci(bs_intensity,conf=0.95,type="bca")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = bs_intensity, conf = 0.95, type = "bca")

Intervals :
Level       BCa
95%   (15.28, 26.33 )
Calculations and Intervals on Original Scale
```

Range / size of CI:
k = 3, 30.66-26.34 = 4.32
k = 5, 30.57-25.39 = 5.18
k = 20, 26.33-15.28 = 11.05

As we can see **the range** of the CI has a general trend of **increase as the value of k increases**. I think the reason behind this is as we have discussed in Question 3.3, when the k value increases, the model gets simpler and tends to underfit, hence the wide confidence interval for huge k (k=20 in this case).

Moreover, both the lower and upper bounds of CI decreases as the value of k decreases, causing the peak value estimated to be very low for huge k.