HW 3

Student Name

9/24/2024

1

Let $E[X] = \mu$. Show that $Var[X] := E[(X - E[X])^2] = E[X^2] - (E[X])^2$. Note, all you have to do is show the second equality (the first is our definition from class).

```
"E[X-E[X]^2] = E[X^2 - 2XE[X]+(E[X])^2] By Foil
            = E[X^2] - E[2XE[X]] + E[E[X]^2] Distribute the Expectation
            = E[X^2] - 2E[X]E[X] + E[X]^2->  Because E[x] is constant
            = E[X^2] - 2(E[X])^2 + E[X]^2  Multiplication of second term
            = E[X^2] - (E[X])^2           Combine like terms"
```

```
## [1] "E[X-E[X]^2] = E[X^2 - 2XE[X]+(E[X])^2] By Foil\n            = E[X^2] - E[2XE[X]] + E[E
[X]^2] Distribute the Expectation\n            = E[X^2] - 2E[X]E[X] + E[X]^2->  Because E[x] is
constant\n            = E[X^2] - 2(E[X])^2 + E[X]^2  Multiplication of second term\n
= E[X^2] - (E[X])^2           Combine like terms"
```

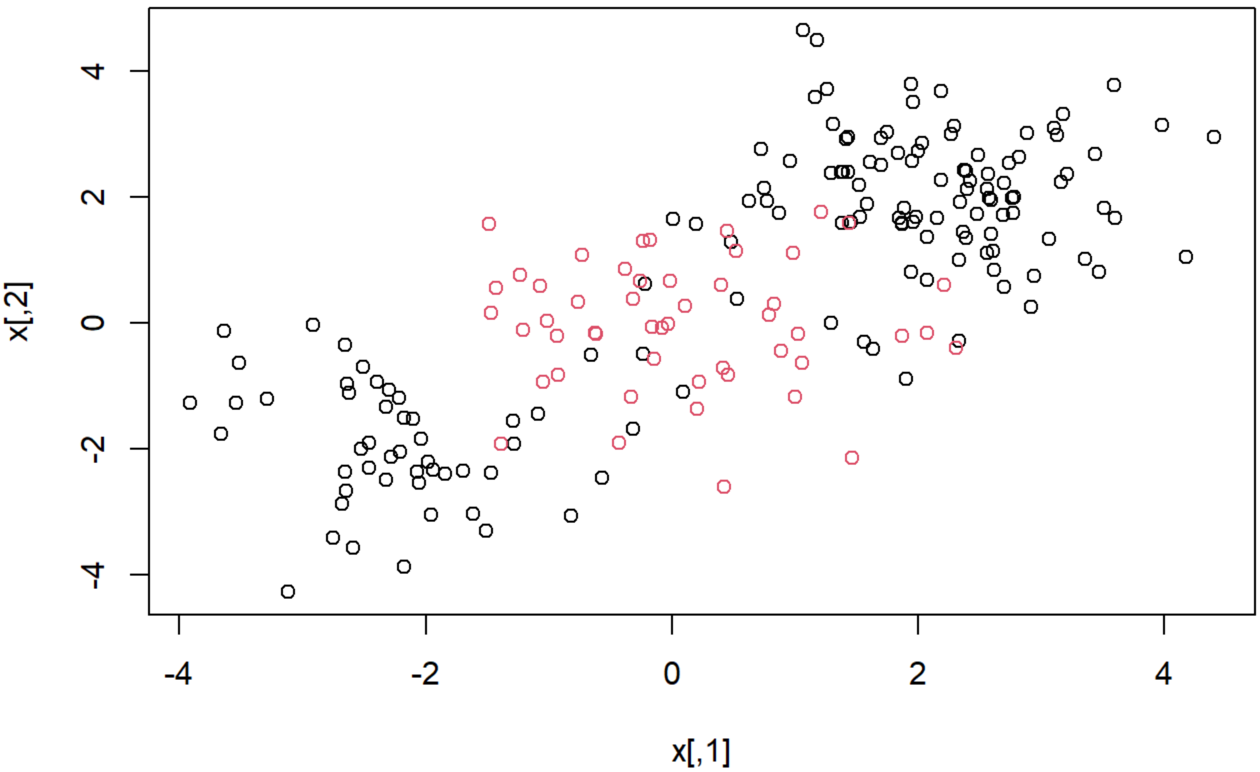
2

In the computational section of this homework, we will discuss support vector machines and tree-based methods. I will begin by simulating some data for you to use with SVM.

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.2.3
```

```
set.seed(1)
x=matrix(rnorm(200*2),ncol=2)
x[1:100,]=x[1:100,]+2
x[101:150,]=x[101:150,]-2
y=c(rep(1,150),rep(2,50))
dat=data.frame(x=x,y=as.factor(y))
plot(x, col=y)
```




```
print(dat)
```

```
##            x.1           x.2 y
## 1    1.37354619  2.4094018397 1
## 2    2.18364332  3.6888732862 1
## 3    1.16437139  3.5865884334 1
## 4    3.59528080  1.6690921993 1
## 5    2.32950777 -0.2852355353 1
## 6    1.17953162  4.4976615898 1
## 7    2.48742905  2.6670661668 1
## 8    2.73832471  2.5413273360 1
## 9    2.57578135  1.9866004769 1
## 10   1.69461161  2.5101084230 1
## 11   3.51178117  1.8356241682 1
## 12   2.38984324  2.4206946433 1
## 13   1.37875942  1.5997532560 1
## 14  -0.21469989  0.6297921225 1
## 15   3.12493092  2.9878382675 1
## 16   1.95506639  3.5197450255 1
## 17   1.98380974  1.6912594308 1
## 18   2.94383621  0.7467102444 1
## 19   2.82122120  2.6422413057 1
## 20   2.59390132  1.9552908631 1
## 21   2.91897737  0.2667815932 1
## 22   2.78213630  2.0021318597 1
## 23   2.07456498  1.3696996661 1
## 24   0.01064830  1.6590314201 1
## 25   2.61982575  0.8434276374 1
## 26   1.94387126  3.8031419079 1
## 27   1.84420449  1.6688679636 1
## 28   0.52924762  0.3944865877 1
## 29   1.52184994  2.1971934387 1
## 30   2.41794156  2.2631756464 1
## 31   3.35867955  1.0141732996 1
## 32   1.89721227 -0.8889206717 1
## 33   2.38767161  1.3595182974 1
## 34   1.94619496  2.5705076359 1
## 35   0.62294044  1.9402767240 1
## 36   1.58500544  1.9018212560 1
## 37   1.60571005  2.5608207286 1
## 38   1.94068660  0.8135413614 1
## 39   3.10002537  3.0967770443 1
## 40   2.76317575  1.9946559717 1
## 41   1.83547640  2.7073106674 1
## 42   1.74663832  3.0341077347 1
## 43   2.69696338  2.2234804149 1
## 44   2.55666320  1.1212923871 1
## 45   1.31124431  3.1629645560 1
## 46   1.29250484 -0.0001649448 1
## 47   2.36458196  1.4552092600 1
## 48   2.76853292  1.7443292908 1
## 49   1.88765379  1.8338789632 1
## 50   2.88110773  3.0204639088 1
## 51   2.39810588  2.1362218931 1
```

```
## 52   1.38797361  2.4071676034 1
## 53   2.34111969  1.9303451870 1
## 54   0.87063690  1.7523356584 1
## 55   3.43302370  2.6955508066 1
## 56   3.98039990  3.1462283572 1
## 57   1.63277852 -0.4030962149 1
## 58   0.95586537  2.5727395552 1
## 59   2.56971963  2.3747244068 1
## 60   1.86494540  1.5747322784 1
## 61   4.40161776  2.9510128076 1
## 62   1.96076000  1.6107628183 1
## 63   2.68973936  1.7156693382 1
## 64   2.02800216  2.8574097781 1
## 65   1.25672679  3.7196272991 1
## 66   2.18879230  2.2700549009 1
## 67   0.19504137  1.5778159902 1
## 68   3.46555486  0.8108867051 1
## 69   2.15325334  1.6689670211 1
## 70   4.17261167  1.0601706735 1
## 71   2.47550953  1.7410674169 1
## 72   1.29005357  2.3943791682 1
## 73   2.61072635  1.1481429080 1
## 74   1.06590237  4.6491668811 1
## 75   0.74636660  2.1560116757 1
## 76   2.29144624  3.1302072675 1
## 77   1.55670813 -0.2891239798 1
## 78   2.00110535  2.7410011572 1
## 79   2.07434132  0.6837548395 1
## 80   1.41047905  2.9198036776 1
## 81   1.43133127  2.3981301555 1
## 82   1.86482138  1.5924714207 1
## 83   3.17808700  3.3242586302 1
## 84   0.47643320  1.2987683308 1
## 85   2.59394619  1.4193856958 1
## 86   2.33295037  0.9989278190 1
## 87   3.06309984  1.3318213932 1
## 88   1.69581608  2.9451849534 1
## 89   2.37001881  2.4337021495 1
## 90   2.26709879  3.0051592177 1
## 91   1.45747997  1.6098813359 1
## 92   3.20786781  2.3763702918 1
## 93   3.16040262  2.2441649245 1
## 94   2.70021365  0.5737426576 1
## 95   3.58683345  3.7784292875 1
## 96   2.55848643  2.1344476609 1
## 97   0.72340779  2.7655989992 1
## 98   1.42673459  2.9551366769 1
## 99   0.77538739  1.9494342986 1
## 100  1.52659936  1.6941845802 1
## 101 -2.62036668 -1.1063262976 1
## 102 -1.95788413 -3.0472981491 1
## 103 -2.91092165 -0.0286626138 1
```

```
## 104 -1.84197123 -2.3836321063 1
## 105 -2.65458464 -0.3458546977 1
## 106 -0.23271273 -0.4877873060 1
## 107 -1.28329252 -1.9170342664 1
## 108 -1.08982577 -1.4327790851 1
## 109 -1.61581464 -3.0245484795 1
## 110 -0.31782392 -1.6769934970 1
## 111 -2.63573645 -0.9563875416 1
## 112 -2.46164473 -1.9009215131 1
## 113 -0.56771776 -2.4541369092 1
## 114 -2.65069635 -2.6557818525 1
## 115 -2.20738074 -2.0359224226 1
## 116 -2.39280793 -0.9308385393 1
## 117 -2.31999287 -2.4839749303 1
## 118 -2.27911330 -2.1210101113 1
## 119 -1.50581167 -3.2941400038 1
## 120 -2.17733048 -1.5056871640 1
## 121 -2.50595746 -0.6920984799 1
## 122 -0.65696117 -0.5029589906 1
## 123 -2.21457941 -1.1852972691 1
## 124 -2.17955653 -3.8697887902 1
## 125 -2.10019074 -1.5179704959 1
## 126 -1.28733369 -1.5438643967 1
## 127 -2.07356440 -2.3534002858 1
## 128 -2.03763417 -1.8295105291 1
## 129 -2.68166048 -2.8640359541 1
## 130 -2.32427027 -1.3207692260 1
## 131 -1.93983956 -2.3271010147 1
## 132 -2.58889449 -3.5690821851 1
## 133 -1.46850381 -2.3674507562 1
## 134 -3.51839408 -0.6355650709 1
## 135 -1.69344214 -2.3342813647 1
## 136 -3.53644982 -1.2672499578 1
## 137 -2.30097613 -1.0534143598 1
## 138 -2.52827990 -1.9956012957 1
## 139 -2.65209478 -2.3523223055 1
## 140 -2.05689678 -2.5296955091 1
## 141 -3.91435943 -1.2604107744 1
## 142 -0.82341669 -3.0634574155 1
## 143 -3.66497244 -1.7537891565 1
## 144 -2.46353040 -2.2894993666 1
## 145 -3.11592011 -4.2648893565 1
## 146 -2.75081900 -3.4088504561 1
## 147  0.08716655 -1.0839806712 1
## 148 -1.98260438 -2.1912789505 1
## 149 -3.28630053 -1.1967167839 1
## 150 -3.64060553 -0.1125255367 1
## 151  0.45018710  1.4738811811 2
## 152 -0.01855983  0.6772684923 2
## 153 -0.31806837  0.3799626866 2
## 154 -0.92936215 -0.1927984265 2
## 155 -1.48746031  1.5778917949 2
```

```
## 156 -1.07519230  0.5962341093 2
## 157  1.00002880 -1.1735769409 2
## 158 -0.62126669 -0.1556425349 2
## 159 -1.38442685 -1.9189098203 2
## 160  1.86929062 -0.1952588461 2
## 161  0.42510038 -2.5923276699 2
## 162 -0.23864710  1.3140021672 2
## 163  1.05848305 -0.6355430010 2
## 164  0.88642265 -0.4299788387 2
## 165 -0.61924305 -0.1693183323 2
## 166  2.20610246  0.6122181740 2
## 167 -0.25502703  0.6783401772 2
## 168 -1.42449465  0.5679519725 2
## 169 -0.14439960 -0.5725426039 2
## 170  0.20753834 -1.3632912563 2
## 171  2.30797840 -0.3887222443 2
## 172  0.10580237  0.2779141325 2
## 173  0.45699881 -0.8230811216 2
## 174 -0.07715294 -0.0688409345 2
## 175 -0.33400084 -1.1676623261 2
## 176 -0.03472603 -0.0083090142 2
## 177  0.78763961  0.1288554016 2
## 178  2.07524501 -0.1458756285 2
## 179  1.02739244 -0.1639109567 2
## 180  1.20790840  1.7635520028 2
## 181 -1.23132342  0.7625865124 2
## 182  0.98389557  1.1114310807 2
## 183  0.21992480 -0.9232069528 2
## 184 -1.46725003  0.1643418384 2
## 185  0.52102274  1.1548251871 2
## 186 -0.15875460 -0.0565214245 2
## 187  1.46458731 -2.1293606482 2
## 188 -0.76608200  0.3448457621 2
## 189 -0.43021175 -1.9049554456 2
## 190 -0.92610950 -0.8111701531 2
## 191 -0.17710396  1.3240043213 2
## 192  0.40201178  0.6156368493 2
## 193 -0.73174817  1.0916689555 2
## 194  0.83037317  0.3066048615 2
## 195 -1.20808279 -0.1101587625 2
## 196 -1.04798441 -0.9243127731 2
## 197  1.44115771  1.5929137537 2
## 198 -1.01584747  0.0450105981 2
## 199  0.41197471 -0.7151284007 2
## 200 -0.38107605  0.8652230997 2
```

2.1

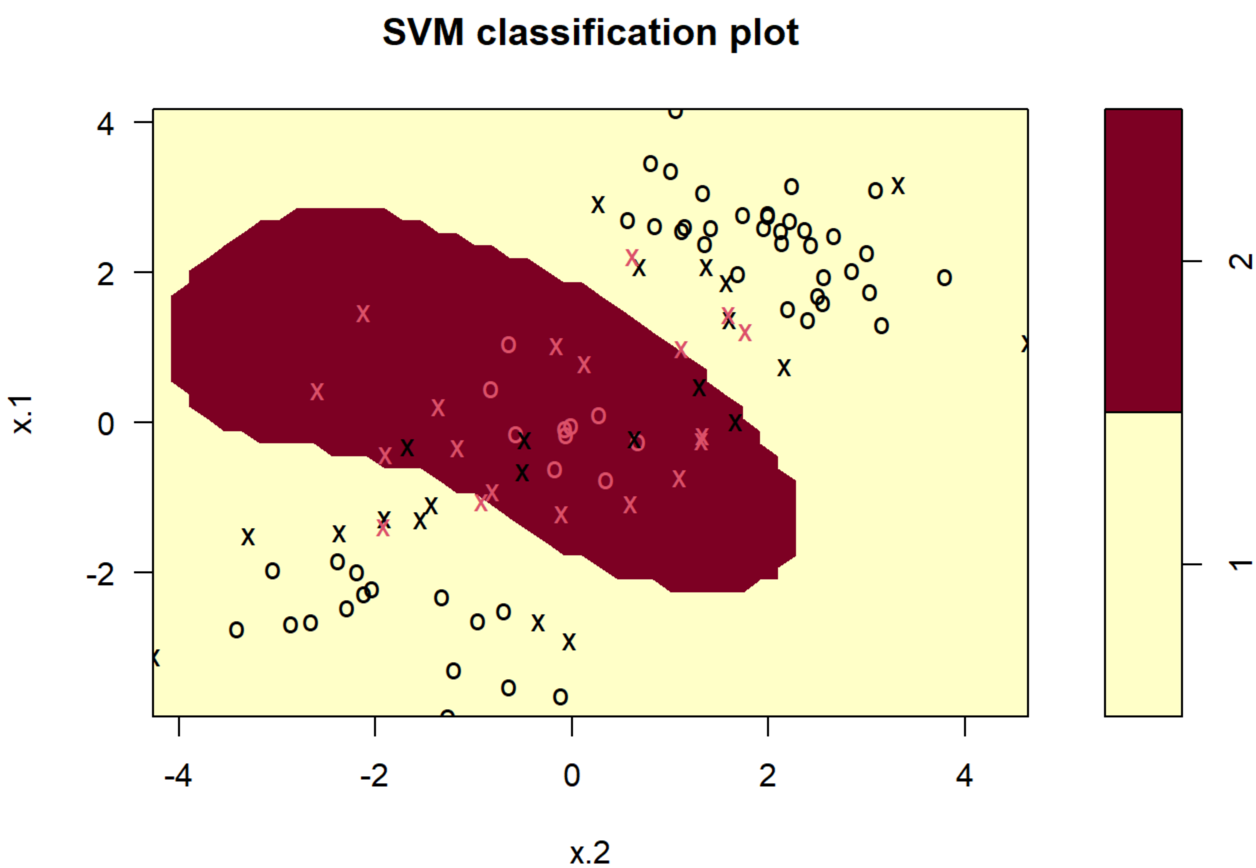
Quite clearly, the above data is not linearly separable. Create a training-testing partition with 100 random observations in the training partition. Fit an svm on this training data using the radial kernel, and tuning parameters $\gamma = 1$, cost $= 1$. Plot the svm on the training data.

```
set.seed(1)

sample <- sample(1:nrow(dat), 0.5 * nrow(dat))
train <- dat[sample, ]
test <- dat[-sample, ]
svm_fit = svm(y~., data =train, kernel = "radial", cost =1, gamma= 1)
svm_fit
```

```
##
## Call:
## svm(formula = y ~ ., data = train, kernel = "radial", cost = 1, gamma = 1)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  41
```

```
plot(svm_fit, train)
```




```
train
```

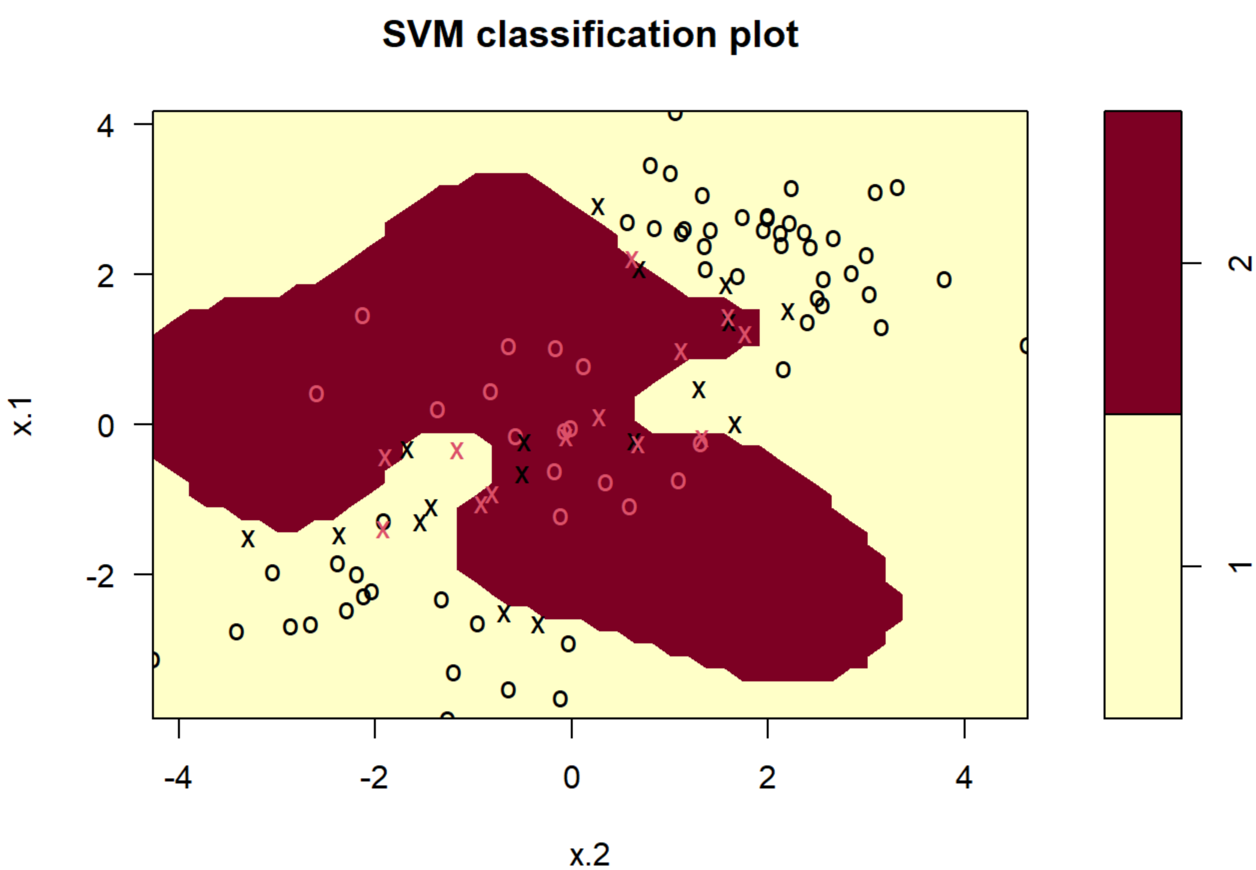
```
##             x.1         x.2 y
## 68   3.46555486  0.810886705 1
## 167 -0.25502703  0.678340177 2
## 129 -2.68166048 -2.864035954 1
## 162 -0.23864710  1.314002167 2
## 43   2.69696338  2.223480415 1
## 14  -0.21469989  0.629792122 1
## 187  1.46458731 -2.129360648 2
## 51   2.39810588  2.136221893 1
## 85   2.59394619  1.419385696 1
## 21   2.91897737  0.266781593 1
## 106 -0.23271273 -0.487787306 1
## 182  0.98389557  1.111431081 2
## 74   1.06590237  4.649166881 1
## 7    2.48742905  2.667066167 1
## 73   2.61072635  1.148142908 1
## 79   2.07434132  0.683754840 1
## 37   1.60571005  2.560820729 1
## 105 -2.65458464 -0.345854698 1
## 110 -0.31782392 -1.676993497 1
## 165 -0.61924305 -0.169318332 2
## 34   1.94619496  2.570507636 1
## 190 -0.92610950 -0.811170153 2
## 126 -1.28733369 -1.543864397 1
## 89   2.37001881  2.433702150 1
## 172  0.10580237  0.277914132 2
## 33   2.38767161  1.359518297 1
## 84   0.47643320  1.298768331 1
## 163  1.05848305 -0.635543001 2
## 70   4.17261167  1.060170673 1
## 188 -0.76608200  0.344845762 2
## 42   1.74663832  3.034107735 1
## 166  2.20610246  0.612218174 2
## 111 -2.63573645 -0.956387542 1
## 148 -1.98260438 -2.191278951 1
## 156 -1.07519230  0.596234109 2
## 20   2.59390132  1.955290863 1
## 44   2.55666320  1.121292387 1
## 121 -2.50595746 -0.692098480 1
## 87   3.06309984  1.331821393 1
## 176 -0.03472603 -0.008309014 2
## 173  0.45699881 -0.823081122 2
## 40   2.76317575  1.994655972 1
## 25   2.61982575  0.843427637 1
## 119 -1.50581167 -3.294140004 1
## 122 -0.65696117 -0.502958991 1
## 39   3.10002537  3.096777044 1
## 170  0.20753834 -1.363291256 2
## 134 -3.51839408 -0.635565071 1
## 24   0.01064830  1.659031420 1
## 195 -1.20808279 -0.110158762 2
## 130 -2.32427027 -1.320769226 1
```

```
## 45   1.31124431  3.162964556 1
## 146 -2.75081900 -3.408850456 1
## 22   2.78213630  2.002131860 1
## 115 -2.20738074 -2.035922423 1
## 104 -1.84197123 -2.383632106 1
## 161  0.42510038 -2.592327670 2
## 144 -2.46353040 -2.289499367 1
## 145 -3.11592011 -4.264889356 1
## 103 -2.91092165 -0.028662614 1
## 75   0.74636660  2.156011676 1
## 13   1.37875942  1.599753256 1
## 159 -1.38442685 -1.918909820 2
## 177  0.78763961  0.128855402 2
## 23   2.07456498  1.369699666 1
## 189 -0.43021175 -1.904955446 2
## 174 -0.07715294 -0.068840934 2
## 141 -3.91435943 -1.260410774 1
## 29   1.52184994  2.197193439 1
## 108 -1.08982577 -1.432779085 1
## 48   2.76853292  1.744329291 1
## 175 -0.33400084 -1.167662326 2
## 149 -3.28630053 -1.196716784 1
## 191 -0.17710396  1.324004321 2
## 31   3.35867955  1.014173300 1
## 102 -1.95788413 -3.047298149 1
## 17   1.98380974  1.691259431 1
## 186 -0.15875460 -0.056521425 2
## 133 -1.46850381 -2.367450756 1
## 197  1.44115771  1.592913754 2
## 83   3.17808700  3.324258630 1
## 118 -2.27911330 -2.121010111 1
## 114 -2.65069635 -2.655781852 1
## 90   2.26709879  3.005159218 1
## 150 -3.64060553 -0.112525537 1
## 107 -1.28329252 -1.917034266 1
## 64   2.02800216  2.857409778 1
## 94   2.70021365  0.573742658 1
## 179  1.02739244 -0.163910957 2
## 96   2.55848643  2.134447661 1
## 169 -0.14439960 -0.572542604 2
## 60   1.86494540  1.574732278 1
## 193 -0.73174817  1.091668956 2
## 93   3.16040262  2.244164924 1
## 180  1.20790840  1.763552003 2
## 10   1.69461161  2.510108423 1
## 1    1.37354619  2.409401840 1
## 196 -1.04798441 -0.924312773 2
## 59   2.56971963  2.374724407 1
## 26   1.94387126  3.803141908 1
```

2.2

Notice that the above decision boundary is decidedly non-linear. It seems to perform reasonably well, but there are indeed some misclassifications. Let's see if increasing the cost [1] helps our classification error rate. Refit the svm with the radial kernel, $\gamma = 1$, and a cost of 10000. Plot this svm on the training data.

```
svm_fit2 = svm(y~., data =train, kernel = "radial", cost =1000, gamma = 1)
plot(svm_fit2, train)
```



2.3

It would appear that we are better capturing the training data, but comment on the dangers (if any exist), of such a model.

This runs the risk of overfitting as the test data may not fit very well compared to the training data

2.4

Create a confusion matrix by using this svm to predict on the current testing partition. Comment on the confusion matrix. Is there any disparity in our classification results?

```
#remove eval = FALSE in above
table(true=dat[-sample,"y"], pred=predict(svm_fit2, newdata=dat[-sample,]))
```

```
##     pred
## true  1  2
##    1 66 13
##    2  2 19
```

Yes, 1 seems to be predicted correctly much more than 2. ##

Is this disparity because of imbalance in the training/testing partition? Find the proportion of class `2` in your training partition and see if it is broadly representative of the underlying 25% of class 2 in the data as a whole.

```
y_column <- train[['y']]
y_column_int <- (as.numeric(as.character(y_column)))
count_2 <- sum(y_column_int == 2)
count_2 / 100
```

```
## [1] 0.29
```

Student Response There does not seem to be a huge difference between the distribution of Y in the training data versus the data as a whole. There is only a difference of 4 between the 29 present in the training versus the 25 percent of the actual data ##

Let's try and balance the above to solutions via cross-validation. Using the `tune` function, pass in the training data, and a list of the following cost and γ values: {0.1, 1, 10, 100, 1000} and {0.5, 1,2,3,4}. Save the output of this function in a variable called `tune.out`.

```
tune_cost <- c(0.1,  1, 10, 100, 1000)
tune_gamma <- c(.5, 1,2,3,4)
set.seed(1)
tune.out <- tune(svm,
                 y ~ .,
                 data = train,
                 ranges = list(gamma = tune_gamma, cost = tune_cost))
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  gamma cost
##    0.5    1
##
## - best performance: 0.12
##
## - Detailed performance results:
##    gamma  cost error dispersion
## 1    0.5 1e-01  0.28 0.15491933
## 2    1.0 1e-01  0.25 0.13540064
## 3    2.0 1e-01  0.28 0.14757296
## 4    3.0 1e-01  0.28 0.15491933
## 5    4.0 1e-01  0.29 0.14491377
## 6    0.5 1e+00  0.12 0.07888106
## 7    1.0 1e+00  0.14 0.09660918
## 8    2.0 1e+00  0.15 0.10801234
## 9    3.0 1e+00  0.15 0.10801234
## 10   4.0 1e+00  0.16 0.09660918
## 11   0.5 1e+01  0.15 0.10801234
## 12   1.0 1e+01  0.16 0.10749677
## 13   2.0 1e+01  0.19 0.15238839
## 14   3.0 1e+01  0.20 0.16329932
## 15   4.0 1e+01  0.18 0.13984118
## 16   0.5 1e+02  0.17 0.11595018
## 17   1.0 1e+02  0.21 0.15238839
## 18   2.0 1e+02  0.18 0.14757296
## 19   3.0 1e+02  0.20 0.13333333
## 20   4.0 1e+02  0.21 0.11972190
## 21   0.5 1e+03  0.23 0.14944341
## 22   1.0 1e+03  0.20 0.14142136
## 23   2.0 1e+03  0.23 0.12516656
## 24   3.0 1e+03  0.27 0.11595018
## 25   4.0 1e+03  0.31 0.15951315
```

I will take `tune.out` and use the best model according to error rate to test on our data. I will report a confusion matrix corresponding to the 100 predictions.

```
table(true=dat[-sample,"y"], pred=predict(tune.out$best.model, newdata=dat[-sample,]))
```

```
##     pred
## true  1  2
##    1 72  7
##    2  1 20
```

2.5

Comment on the confusion matrix. How have we improved upon the model in question 2 and what qualifications are still necessary for this improved model.
The confusion matrix seems to assert our new svm model better fits the testing data. The gamma and cost that we used initially are present in our tune function, so we know that our new gamma and our new cost variable will at least be as good as the ones we had before. However, we still need to qualify that ther may be a better polynomial kernel or different combinations that we have not used yet.

3

Let’s turn now to decision trees.

```
library(kmed)
```

```
## Warning: package 'kmed' was built under R version 4.2.3
```

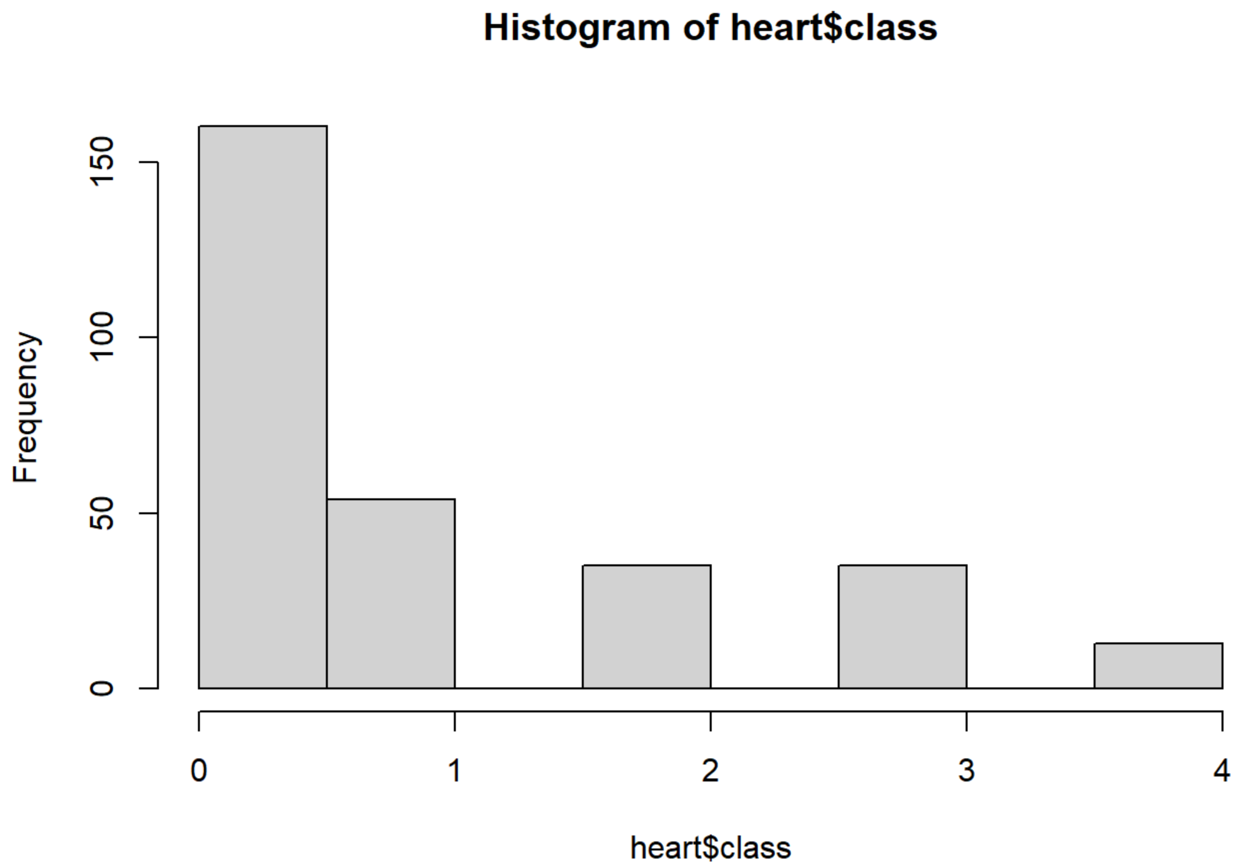
```
data(heart)
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.2.3
```

3.1

The response variable is currently a categorical variable with four levels. Convert heart disease into binary categorical variable. Then, ensure that it is properly stored as a factor.

```
hist(heart$class)
```




```
heart$cp <- as.numeric(as.character(heart$cp))
heart$sex <- as.factor(heart$sex)
heart$fbs <- as.factor(heart$fbs)
heart$exang <- as.factor(heart$exang)
heart$class <- as.numeric(heart$class)
High <- ifelse(heart$cp <= 2, "low", "high")
High <- as.factor(High)
heart$high <- High
heart
```

```
##      age   sex cp trestbps chol   fbs restecg thalach exang oldpeak slope ca
## 1    63  TRUE  1      145  233  TRUE       2     150 FALSE     2.3     3  0
## 2    67  TRUE  4      160  286 FALSE       2     108  TRUE     1.5     2  3
## 3    67  TRUE  4      120  229 FALSE       2     129  TRUE     2.6     2  2
## 4    37  TRUE  3      130  250 FALSE       0     187 FALSE     3.5     3  0
## 5    41 FALSE  2      130  204 FALSE       2     172 FALSE     1.4     1  0
## 6    56  TRUE  2      120  236 FALSE       0     178 FALSE     0.8     1  0
## 7    62 FALSE  4      140  268 FALSE       2     160 FALSE     3.6     3  2
## 8    57 FALSE  4      120  354 FALSE       0     163  TRUE     0.6     1  0
## 9    63  TRUE  4      130  254 FALSE       2     147 FALSE     1.4     2  1
## 10   53  TRUE  4      140  203  TRUE       2     155  TRUE     3.1     3  0
## 11   57  TRUE  4      140  192 FALSE       0     148 FALSE     0.4     2  0
## 12   56 FALSE  2      140  294 FALSE       2     153 FALSE     1.3     2  0
## 13   56  TRUE  3      130  256  TRUE       2     142  TRUE     0.6     2  1
## 14   44  TRUE  2      120  263 FALSE       0     173 FALSE     0.0     1  0
## 15   52  TRUE  3      172  199  TRUE       0     162 FALSE     0.5     1  0
## 16   57  TRUE  3      150  168 FALSE       0     174 FALSE     1.6     1  0
## 17   48  TRUE  2      110  229 FALSE       0     168 FALSE     1.0     3  0
## 18   54  TRUE  4      140  239 FALSE       0     160 FALSE     1.2     1  0
## 19   48 FALSE  3      130  275 FALSE       0     139 FALSE     0.2     1  0
## 20   49  TRUE  2      130  266 FALSE       0     171 FALSE     0.6     1  0
## 21   64  TRUE  1      110  211 FALSE       2     144  TRUE     1.8     2  0
## 22   58 FALSE  1      150  283  TRUE       2     162 FALSE     1.0     1  0
## 23   58  TRUE  2      120  284 FALSE       2     160 FALSE     1.8     2  0
## 24   58  TRUE  3      132  224 FALSE       2     173 FALSE     3.2     1  2
## 25   60  TRUE  4      130  206 FALSE       2     132  TRUE     2.4     2  2
## 26   50 FALSE  3      120  219 FALSE       0     158 FALSE     1.6     2  0
## 27   58 FALSE  3      120  340 FALSE       0     172 FALSE     0.0     1  0
## 28   66 FALSE  1      150  226 FALSE       0     114 FALSE     2.6     3  0
## 29   43  TRUE  4      150  247 FALSE       0     171 FALSE     1.5     1  0
## 30   40  TRUE  4      110  167 FALSE       2     114  TRUE     2.0     2  0
## 31   69 FALSE  1      140  239 FALSE       0     151 FALSE     1.8     1  2
## 32   60  TRUE  4      117  230  TRUE       0     160  TRUE     1.4     1  2
## 33   64  TRUE  3      140  335 FALSE       0     158 FALSE     0.0     1  0
## 34   59  TRUE  4      135  234 FALSE       0     161 FALSE     0.5     2  0
## 35   44  TRUE  3      130  233 FALSE       0     179  TRUE     0.4     1  0
## 36   42  TRUE  4      140  226 FALSE       0     178 FALSE     0.0     1  0
## 37   43  TRUE  4      120  177 FALSE       2     120  TRUE     2.5     2  0
## 38   57  TRUE  4      150  276 FALSE       2     112  TRUE     0.6     2  1
## 39   55  TRUE  4      132  353 FALSE       0     132  TRUE     1.2     2  1
## 40   61  TRUE  3      150  243  TRUE       0     137  TRUE     1.0     2  0
## 41   65 FALSE  4      150  225 FALSE       2     114 FALSE     1.0     2  3
## 42   40  TRUE  1      140  199 FALSE       0     178  TRUE     1.4     1  0
## 43   71 FALSE  2      160  302 FALSE       0     162 FALSE     0.4     1  2
## 44   59  TRUE  3      150  212  TRUE       0     157 FALSE     1.6     1  0
## 45   61 FALSE  4      130  330 FALSE       2     169 FALSE     0.0     1  0
## 46   58  TRUE  3      112  230 FALSE       2     165 FALSE     2.5     2  1
## 47   51  TRUE  3      110  175 FALSE       0     123 FALSE     0.6     1  0
## 48   50  TRUE  4      150  243 FALSE       2     128 FALSE     2.6     2  0
## 49   65 FALSE  3      140  417  TRUE       2     157 FALSE     0.8     1  1
## 50   53  TRUE  3      130  197  TRUE       2     152 FALSE     1.2     3  0
## 51   41 FALSE  2      105  198 FALSE       0     168 FALSE     0.0     1  1
```

```
## 52   65  TRUE  4      120  177 FALSE       0     140 FALSE     0.4      1  0
## 53   44  TRUE  4      112  290 FALSE       2     153 FALSE     0.0      1  1
## 54   44  TRUE  2      130  219 FALSE       2     188 FALSE     0.0      1  0
## 55   60  TRUE  4      130  253 FALSE       0     144  TRUE     1.4      1  1
## 56   54  TRUE  4      124  266 FALSE       2     109  TRUE     2.2      2  1
## 57   50  TRUE  3      140  233 FALSE       0     163 FALSE     0.6      2  1
## 58   41  TRUE  4      110  172 FALSE       2     158 FALSE     0.0      1  0
## 59   54  TRUE  3      125  273 FALSE       2     152 FALSE     0.5      3  1
## 60   51  TRUE  1      125  213 FALSE       2     125  TRUE     1.4      1  1
## 61   51 FALSE  4      130  305 FALSE       0     142  TRUE     1.2      2  0
## 62   46 FALSE  3      142  177 FALSE       2     160  TRUE     1.4      3  0
## 63   58  TRUE  4      128  216 FALSE       2     131  TRUE     2.2      2  3
## 64   54 FALSE  3      135  304  TRUE       0     170 FALSE     0.0      1  0
## 65   54  TRUE  4      120  188 FALSE       0     113 FALSE     1.4      2  1
## 66   60  TRUE  4      145  282 FALSE       2     142  TRUE     2.8      2  2
## 67   60  TRUE  3      140  185 FALSE       2     155 FALSE     3.0      2  0
## 68   54  TRUE  3      150  232 FALSE       2     165 FALSE     1.6      1  0
## 69   59  TRUE  4      170  326 FALSE       2     140  TRUE     3.4      3  0
## 70   46  TRUE  3      150  231 FALSE       0     147 FALSE     3.6      2  0
## 71   65 FALSE  3      155  269 FALSE       0     148 FALSE     0.8      1  0
## 72   67  TRUE  4      125  254  TRUE       0     163 FALSE     0.2      2  2
## 73   62  TRUE  4      120  267 FALSE       0      99  TRUE     1.8      2  2
## 74   65  TRUE  4      110  248 FALSE       2     158 FALSE     0.6      1  2
## 75   44  TRUE  4      110  197 FALSE       2     177 FALSE     0.0      1  1
## 76   65 FALSE  3      160  360 FALSE       2     151 FALSE     0.8      1  0
## 77   60  TRUE  4      125  258 FALSE       2     141  TRUE     2.8      2  1
## 78   51 FALSE  3      140  308 FALSE       2     142 FALSE     1.5      1  1
## 79   48  TRUE  2      130  245 FALSE       2     180 FALSE     0.2      2  0
## 80   58  TRUE  4      150  270 FALSE       2     111  TRUE     0.8      1  0
## 81   45  TRUE  4      104  208 FALSE       2     148  TRUE     3.0      2  0
## 82   53 FALSE  4      130  264 FALSE       2     143 FALSE     0.4      2  0
## 83   39  TRUE  3      140  321 FALSE       2     182 FALSE     0.0      1  0
## 84   68  TRUE  3      180  274  TRUE       2     150  TRUE     1.6      2  0
## 85   52  TRUE  2      120  325 FALSE       0     172 FALSE     0.2      1  0
## 86   44  TRUE  3      140  235 FALSE       2     180 FALSE     0.0      1  0
## 87   47  TRUE  3      138  257 FALSE       2     156 FALSE     0.0      1  0
## 89   53 FALSE  4      138  234 FALSE       2     160 FALSE     0.0      1  0
## 90   51 FALSE  3      130  256 FALSE       2     149 FALSE     0.5      1  0
## 91   66  TRUE  4      120  302 FALSE       2     151 FALSE     0.4      2  0
## 92   62 FALSE  4      160  164 FALSE       2     145 FALSE     6.2      3  3
## 93   62  TRUE  3      130  231 FALSE       0     146 FALSE     1.8      2  3
## 94   44 FALSE  3      108  141 FALSE       0     175 FALSE     0.6      2  0
## 95   63 FALSE  3      135  252 FALSE       2     172 FALSE     0.0      1  0
## 96   52  TRUE  4      128  255 FALSE       0     161  TRUE     0.0      1  1
## 97   59  TRUE  4      110  239 FALSE       2     142  TRUE     1.2      2  1
## 98   60 FALSE  4      150  258 FALSE       2     157 FALSE     2.6      2  2
## 99   52  TRUE  2      134  201 FALSE       0     158 FALSE     0.8      1  1
## 100  48  TRUE  4      122  222 FALSE       2     186 FALSE     0.0      1  0
## 101  45  TRUE  4      115  260 FALSE       2     185 FALSE     0.0      1  0
## 102  34  TRUE  1      118  182 FALSE       2     174 FALSE     0.0      1  0
## 103  57 FALSE  4      128  303 FALSE       2     159 FALSE     0.0      1  1
## 104  71 FALSE  3      110  265  TRUE       2     130 FALSE     0.0      1  1
```

```
## 105  49  TRUE  3      120  188 FALSE       0     139 FALSE     2.0      2  3
## 106  54  TRUE  2      108  309 FALSE       0     156 FALSE     0.0      1  0
## 107  59  TRUE  4      140  177 FALSE       0     162  TRUE     0.0      1  1
## 108  57  TRUE  3      128  229 FALSE       2     150 FALSE     0.4      2  1
## 109  61  TRUE  4      120  260 FALSE       0     140  TRUE     3.6      2  1
## 110  39  TRUE  4      118  219 FALSE       0     140 FALSE     1.2      2  0
## 111  61 FALSE  4      145  307 FALSE       2     146  TRUE     1.0      2  0
## 112  56  TRUE  4      125  249  TRUE       2     144  TRUE     1.2      2  1
## 113  52  TRUE  1      118  186 FALSE       2     190 FALSE     0.0      2  0
## 114  43 FALSE  4      132  341  TRUE       2     136  TRUE     3.0      2  0
## 115  62 FALSE  3      130  263 FALSE       0      97 FALSE     1.2      2  1
## 116  41  TRUE  2      135  203 FALSE       0     132 FALSE     0.0      2  0
## 117  58  TRUE  3      140  211  TRUE       2     165 FALSE     0.0      1  0
## 118  35 FALSE  4      138  183 FALSE       0     182 FALSE     1.4      1  0
## 119  63  TRUE  4      130  330  TRUE       2     132  TRUE     1.8      1  3
## 120  65  TRUE  4      135  254 FALSE       2     127 FALSE     2.8      2  1
## 121  48  TRUE  4      130  256  TRUE       2     150  TRUE     0.0      1  2
## 122  63 FALSE  4      150  407 FALSE       2     154 FALSE     4.0      2  3
## 123  51  TRUE  3      100  222 FALSE       0     143  TRUE     1.2      2  0
## 124  55  TRUE  4      140  217 FALSE       0     111  TRUE     5.6      3  0
## 125  65  TRUE  1      138  282  TRUE       2     174 FALSE     1.4      2  1
## 126  45 FALSE  2      130  234 FALSE       2     175 FALSE     0.6      2  0
## 127  56 FALSE  4      200  288  TRUE       2     133  TRUE     4.0      3  2
## 128  54  TRUE  4      110  239 FALSE       0     126  TRUE     2.8      2  1
## 129  44  TRUE  2      120  220 FALSE       0     170 FALSE     0.0      1  0
## 130  62 FALSE  4      124  209 FALSE       0     163 FALSE     0.0      1  0
## 131  54  TRUE  3      120  258 FALSE       2     147 FALSE     0.4      2  0
## 132  51  TRUE  3       94  227 FALSE       0     154  TRUE     0.0      1  1
## 133  29  TRUE  2      130  204 FALSE       2     202 FALSE     0.0      1  0
## 134  51  TRUE  4      140  261 FALSE       2     186  TRUE     0.0      1  0
## 135  43 FALSE  3      122  213 FALSE       0     165 FALSE     0.2      2  0
## 136  55 FALSE  2      135  250 FALSE       2     161 FALSE     1.4      2  0
## 137  70  TRUE  4      145  174 FALSE       0     125  TRUE     2.6      3  0
## 138  62  TRUE  2      120  281 FALSE       2     103 FALSE     1.4      2  1
## 139  35  TRUE  4      120  198 FALSE       0     130  TRUE     1.6      2  0
## 140  51  TRUE  3      125  245  TRUE       2     166 FALSE     2.4      2  0
## 141  59  TRUE  2      140  221 FALSE       0     164  TRUE     0.0      1  0
## 142  59  TRUE  1      170  288 FALSE       2     159 FALSE     0.2      2  0
## 143  52  TRUE  2      128  205  TRUE       0     184 FALSE     0.0      1  0
## 144  64  TRUE  3      125  309 FALSE       0     131  TRUE     1.8      2  0
## 145  58  TRUE  3      105  240 FALSE       2     154  TRUE     0.6      2  0
## 146  47  TRUE  3      108  243 FALSE       0     152 FALSE     0.0      1  0
## 147  57  TRUE  4      165  289  TRUE       2     124 FALSE     1.0      2  3
## 148  41  TRUE  3      112  250 FALSE       0     179 FALSE     0.0      1  0
## 149  45  TRUE  2      128  308 FALSE       2     170 FALSE     0.0      1  0
## 150  60 FALSE  3      102  318 FALSE       0     160 FALSE     0.0      1  1
## 151  52  TRUE  1      152  298  TRUE       0     178 FALSE     1.2      2  0
## 152  42 FALSE  4      102  265 FALSE       2     122 FALSE     0.6      2  0
## 153  67 FALSE  3      115  564 FALSE       2     160 FALSE     1.6      2  0
## 154  55  TRUE  4      160  289 FALSE       2     145  TRUE     0.8      2  1
## 155  64  TRUE  4      120  246 FALSE       2      96  TRUE     2.2      3  1
## 156  70  TRUE  4      130  322 FALSE       2     109 FALSE     2.4      2  3
```

```
## 157  51  TRUE  4      140  299 FALSE       0     173  TRUE     1.6     1  0
## 158  58  TRUE  4      125  300 FALSE       2     171 FALSE     0.0     1  2
## 159  60  TRUE  4      140  293 FALSE       2     170 FALSE     1.2     2  2
## 160  68  TRUE  3      118  277 FALSE       0     151 FALSE     1.0     1  1
## 161  46  TRUE  2      101  197  TRUE       0     156 FALSE     0.0     1  0
## 162  77  TRUE  4      125  304 FALSE       2     162  TRUE     0.0     1  3
## 163  54 FALSE  3      110  214 FALSE       0     158 FALSE     1.6     2  0
## 164  58 FALSE  4      100  248 FALSE       2     122 FALSE     1.0     2  0
## 165  48  TRUE  3      124  255  TRUE       0     175 FALSE     0.0     1  2
## 166  57  TRUE  4      132  207 FALSE       0     168  TRUE     0.0     1  0
## 168  54 FALSE  2      132  288  TRUE       2     159  TRUE     0.0     1  1
## 169  35  TRUE  4      126  282 FALSE       2     156  TRUE     0.0     1  0
## 170  45 FALSE  2      112  160 FALSE       0     138 FALSE     0.0     2  0
## 171  70  TRUE  3      160  269 FALSE       0     112  TRUE     2.9     2  1
## 172  53  TRUE  4      142  226 FALSE       2     111  TRUE     0.0     1  0
## 173  59 FALSE  4      174  249 FALSE       0     143  TRUE     0.0     2  0
## 174  62 FALSE  4      140  394 FALSE       2     157 FALSE     1.2     2  0
## 175  64  TRUE  4      145  212 FALSE       2     132 FALSE     2.0     2  2
## 176  57  TRUE  4      152  274 FALSE       0      88  TRUE     1.2     2  1
## 177  52  TRUE  4      108  233  TRUE       0     147 FALSE     0.1     1  3
## 178  56  TRUE  4      132  184 FALSE       2     105  TRUE     2.1     2  1
## 179  43  TRUE  3      130  315 FALSE       0     162 FALSE     1.9     1  1
## 180  53  TRUE  3      130  246  TRUE       2     173 FALSE     0.0     1  3
## 181  48  TRUE  4      124  274 FALSE       2     166 FALSE     0.5     2  0
## 182  56 FALSE  4      134  409 FALSE       2     150  TRUE     1.9     2  2
## 183  42  TRUE  1      148  244 FALSE       2     178 FALSE     0.8     1  2
## 184  59  TRUE  1      178  270 FALSE       2     145 FALSE     4.2     3  0
## 185  60 FALSE  4      158  305 FALSE       2     161 FALSE     0.0     1  0
## 186  63 FALSE  2      140  195 FALSE       0     179 FALSE     0.0     1  2
## 187  42  TRUE  3      120  240  TRUE       0     194 FALSE     0.8     3  0
## 188  66  TRUE  2      160  246 FALSE       0     120  TRUE     0.0     2  3
## 189  54  TRUE  2      192  283 FALSE       2     195 FALSE     0.0     1  1
## 190  69  TRUE  3      140  254 FALSE       2     146 FALSE     2.0     2  3
## 191  50  TRUE  3      129  196 FALSE       0     163 FALSE     0.0     1  0
## 192  51  TRUE  4      140  298 FALSE       0     122  TRUE     4.2     2  3
## 194  62 FALSE  4      138  294  TRUE       0     106 FALSE     1.9     2  3
## 195  68 FALSE  3      120  211 FALSE       2     115 FALSE     1.5     2  0
## 196  67  TRUE  4      100  299 FALSE       2     125  TRUE     0.9     2  2
## 197  69  TRUE  1      160  234  TRUE       2     131 FALSE     0.1     2  1
## 198  45 FALSE  4      138  236 FALSE       2     152  TRUE     0.2     2  0
## 199  50 FALSE  2      120  244 FALSE       0     162 FALSE     1.1     1  0
## 200  59  TRUE  1      160  273 FALSE       2     125 FALSE     0.0     1  0
## 201  50 FALSE  4      110  254 FALSE       2     159 FALSE     0.0     1  0
## 202  64 FALSE  4      180  325 FALSE       0     154  TRUE     0.0     1  0
## 203  57  TRUE  3      150  126  TRUE       0     173 FALSE     0.2     1  1
## 204  64 FALSE  3      140  313 FALSE       0     133 FALSE     0.2     1  0
## 205  43  TRUE  4      110  211 FALSE       0     161 FALSE     0.0     1  0
## 206  45  TRUE  4      142  309 FALSE       2     147  TRUE     0.0     2  3
## 207  58  TRUE  4      128  259 FALSE       2     130  TRUE     3.0     2  2
## 208  50  TRUE  4      144  200 FALSE       2     126  TRUE     0.9     2  0
## 209  55  TRUE  2      130  262 FALSE       0     155 FALSE     0.0     1  0
## 210  62 FALSE  4      150  244 FALSE       0     154  TRUE     1.4     2  0
```

```
## 211  37 FALSE  3      120  215 FALSE       0     170 FALSE     0.0     1  0
## 212  38  TRUE  1      120  231 FALSE       0     182  TRUE     3.8     2  0
## 213  41  TRUE  3      130  214 FALSE       2     168 FALSE     2.0     2  0
## 214  66 FALSE  4      178  228  TRUE       0     165  TRUE     1.0     2  2
## 215  52  TRUE  4      112  230 FALSE       0     160 FALSE     0.0     1  1
## 216  56  TRUE  1      120  193 FALSE       2     162 FALSE     1.9     2  0
## 217  46 FALSE  2      105  204 FALSE       0     172 FALSE     0.0     1  0
## 218  46 FALSE  4      138  243 FALSE       2     152  TRUE     0.0     2  0
## 219  64 FALSE  4      130  303 FALSE       0     122 FALSE     2.0     2  2
## 220  59  TRUE  4      138  271 FALSE       2     182 FALSE     0.0     1  0
## 221  41 FALSE  3      112  268 FALSE       2     172  TRUE     0.0     1  0
## 222  54 FALSE  3      108  267 FALSE       2     167 FALSE     0.0     1  0
## 223  39 FALSE  3       94  199 FALSE       0     179 FALSE     0.0     1  0
## 224  53  TRUE  4      123  282 FALSE       0      95  TRUE     2.0     2  2
## 225  63 FALSE  4      108  269 FALSE       0     169  TRUE     1.8     2  2
## 226  34 FALSE  2      118  210 FALSE       0     192 FALSE     0.7     1  0
## 227  47  TRUE  4      112  204 FALSE       0     143 FALSE     0.1     1  0
## 228  67 FALSE  3      152  277 FALSE       0     172 FALSE     0.0     1  1
## 229  54  TRUE  4      110  206 FALSE       2     108  TRUE     0.0     2  1
## 230  66  TRUE  4      112  212 FALSE       2     132  TRUE     0.1     1  1
## 231  52 FALSE  3      136  196 FALSE       2     169 FALSE     0.1     2  0
## 232  55 FALSE  4      180  327 FALSE       1     117  TRUE     3.4     2  0
## 233  49  TRUE  3      118  149 FALSE       2     126 FALSE     0.8     1  3
## 234  74 FALSE  2      120  269 FALSE       2     121  TRUE     0.2     1  1
## 235  54 FALSE  3      160  201 FALSE       0     163 FALSE     0.0     1  1
## 236  54  TRUE  4      122  286 FALSE       2     116  TRUE     3.2     2  2
## 237  56  TRUE  4      130  283  TRUE       2     103  TRUE     1.6     3  0
## 238  46  TRUE  4      120  249 FALSE       2     144 FALSE     0.8     1  0
## 239  49 FALSE  2      134  271 FALSE       0     162 FALSE     0.0     2  0
## 240  42  TRUE  2      120  295 FALSE       0     162 FALSE     0.0     1  0
## 241  41  TRUE  2      110  235 FALSE       0     153 FALSE     0.0     1  0
## 242  41 FALSE  2      126  306 FALSE       0     163 FALSE     0.0     1  0
## 243  49 FALSE  4      130  269 FALSE       0     163 FALSE     0.0     1  0
## 244  61  TRUE  1      134  234 FALSE       0     145 FALSE     2.6     2  2
## 245  60 FALSE  3      120  178  TRUE       0      96 FALSE     0.0     1  0
## 246  67  TRUE  4      120  237 FALSE       0      71 FALSE     1.0     2  0
## 247  58  TRUE  4      100  234 FALSE       0     156 FALSE     0.1     1  1
## 248  47  TRUE  4      110  275 FALSE       2     118  TRUE     1.0     2  1
## 249  52  TRUE  4      125  212 FALSE       0     168 FALSE     1.0     1  2
## 250  62  TRUE  2      128  208  TRUE       2     140 FALSE     0.0     1  0
## 251  57  TRUE  4      110  201 FALSE       0     126  TRUE     1.5     2  0
## 252  58  TRUE  4      146  218 FALSE       0     105 FALSE     2.0     2  1
## 253  64  TRUE  4      128  263 FALSE       0     105  TRUE     0.2     2  1
## 254  51 FALSE  3      120  295 FALSE       2     157 FALSE     0.6     1  0
## 255  43  TRUE  4      115  303 FALSE       0     181 FALSE     1.2     2  0
## 256  42 FALSE  3      120  209 FALSE       0     173 FALSE     0.0     2  0
## 257  67 FALSE  4      106  223 FALSE       0     142 FALSE     0.3     1  2
## 258  76 FALSE  3      140  197 FALSE       1     116 FALSE     1.1     2  0
## 259  70  TRUE  2      156  245 FALSE       2     143 FALSE     0.0     1  0
## 260  57  TRUE  2      124  261 FALSE       0     141 FALSE     0.3     1  0
## 261  44 FALSE  3      118  242 FALSE       0     149 FALSE     0.3     2  1
## 262  58 FALSE  2      136  319  TRUE       2     152 FALSE     0.0     1  2
```

```
## 263  60 FALSE  1      150  240 FALSE       0     171 FALSE     0.9     1  0
## 264  44  TRUE  3      120  226 FALSE       0     169 FALSE     0.0     1  0
## 265  61  TRUE  4      138  166 FALSE       2     125  TRUE     3.6     2  1
## 266  42  TRUE  4      136  315 FALSE       0     125  TRUE     1.8     2  0
## 268  59  TRUE  3      126  218  TRUE       0     134 FALSE     2.2     2  1
## 269  40  TRUE  4      152  223 FALSE       0     181 FALSE     0.0     1  0
## 270  42  TRUE  3      130  180 FALSE       0     150 FALSE     0.0     1  0
## 271  61  TRUE  4      140  207 FALSE       2     138  TRUE     1.9     1  1
## 272  66  TRUE  4      160  228 FALSE       2     138 FALSE     2.3     1  0
## 273  46  TRUE  4      140  311 FALSE       0     120  TRUE     1.8     2  2
## 274  71 FALSE  4      112  149 FALSE       0     125 FALSE     1.6     2  0
## 275  59  TRUE  1      134  204 FALSE       0     162 FALSE     0.8     1  2
## 276  64  TRUE  1      170  227 FALSE       2     155 FALSE     0.6     2  0
## 277  66 FALSE  3      146  278 FALSE       2     152 FALSE     0.0     2  1
## 278  39 FALSE  3      138  220 FALSE       0     152 FALSE     0.0     2  0
## 279  57  TRUE  2      154  232 FALSE       2     164 FALSE     0.0     1  1
## 280  58 FALSE  4      130  197 FALSE       0     131 FALSE     0.6     2  0
## 281  57  TRUE  4      110  335 FALSE       0     143  TRUE     3.0     2  1
## 282  47  TRUE  3      130  253 FALSE       0     179 FALSE     0.0     1  0
## 283  55 FALSE  4      128  205 FALSE       1     130  TRUE     2.0     2  1
## 284  35  TRUE  2      122  192 FALSE       0     174 FALSE     0.0     1  0
## 285  61  TRUE  4      148  203 FALSE       0     161 FALSE     0.0     1  1
## 286  58  TRUE  4      114  318 FALSE       1     140 FALSE     4.4     3  3
## 287  58 FALSE  4      170  225  TRUE       2     146  TRUE     2.8     2  2
## 289  56  TRUE  2      130  221 FALSE       2     163 FALSE     0.0     1  0
## 290  56  TRUE  2      120  240 FALSE       0     169 FALSE     0.0     3  0
## 291  67  TRUE  3      152  212 FALSE       2     150 FALSE     0.8     2  0
## 292  55 FALSE  2      132  342 FALSE       0     166 FALSE     1.2     1  0
## 293  44  TRUE  4      120  169 FALSE       0     144  TRUE     2.8     3  0
## 294  63  TRUE  4      140  187 FALSE       2     144  TRUE     4.0     1  2
## 295  63 FALSE  4      124  197 FALSE       0     136  TRUE     0.0     2  0
## 296  41  TRUE  2      120  157 FALSE       0     182 FALSE     0.0     1  0
## 297  59  TRUE  4      164  176  TRUE       2      90 FALSE     1.0     2  2
## 298  57 FALSE  4      140  241 FALSE       0     123  TRUE     0.2     2  0
## 299  45  TRUE  1      110  264 FALSE       0     132 FALSE     1.2     2  0
## 300  68  TRUE  4      144  193  TRUE       0     141 FALSE     3.4     2  2
## 301  57  TRUE  4      130  131 FALSE       0     115  TRUE     1.2     2  1
## 302  57 FALSE  2      130  236 FALSE       2     174 FALSE     0.0     2  1
##     thal class high
## 1      6     0  low
## 2      3     2 high
## 3      7     1 high
## 4      3     0 high
## 5      3     0  low
## 6      3     0  low
## 7      3     3 high
## 8      3     0 high
## 9      7     2 high
## 10     7     1 high
## 11     6     0 high
## 12     3     0  low
## 13     6     2 high
```

```
## 14     7     0  low
## 15     7     0 high
## 16     3     0 high
## 17     7     1  low
## 18     3     0 high
## 19     3     0 high
## 20     3     0  low
## 21     3     0  low
## 22     3     0  low
## 23     3     1  low
## 24     7     3 high
## 25     7     4 high
## 26     3     0 high
## 27     3     0 high
## 28     3     0  low
## 29     3     0 high
## 30     7     3 high
## 31     3     0  low
## 32     7     2 high
## 33     3     1 high
## 34     7     0 high
## 35     3     0 high
## 36     3     0 high
## 37     7     3 high
## 38     6     1 high
## 39     7     3 high
## 40     3     0 high
## 41     7     4 high
## 42     7     0  low
## 43     3     0  low
## 44     3     0 high
## 45     3     1 high
## 46     7     4 high
## 47     3     0 high
## 48     7     4 high
## 49     3     0 high
## 50     3     0 high
## 51     3     0  low
## 52     7     0 high
## 53     3     2 high
## 54     3     0  low
## 55     7     1 high
## 56     7     1 high
## 57     7     1 high
## 58     7     1 high
## 59     3     0 high
## 60     3     0  low
## 61     7     2 high
## 62     3     0 high
## 63     7     1 high
## 64     3     0 high
## 65     7     2 high
```

```
## 66     7     2 high
## 67     3     1 high
## 68     7     0 high
## 69     7     2 high
## 70     3     1 high
## 71     3     0 high
## 72     7     3 high
## 73     7     1 high
## 74     6     1 high
## 75     3     1 high
## 76     3     0 high
## 77     7     1 high
## 78     3     0 high
## 79     3     0  low
## 80     7     3 high
## 81     3     0 high
## 82     3     0 high
## 83     3     0 high
## 84     7     3 high
## 85     3     0  low
## 86     3     0 high
## 87     3     0 high
## 89     3     0 high
## 90     3     0 high
## 91     3     0 high
## 92     7     3 high
## 93     7     0 high
## 94     3     0 high
## 95     3     0 high
## 96     7     1 high
## 97     7     2 high
## 98     7     3 high
## 99     3     0  low
## 100    3     0 high
## 101    3     0 high
## 102    3     0  low
## 103    3     0 high
## 104    3     0 high
## 105    7     3 high
## 106    7     0  low
## 107    7     2 high
## 108    7     1 high
## 109    7     2 high
## 110    7     3 high
## 111    7     1 high
## 112    3     1 high
## 113    6     0  low
## 114    7     2 high
## 115    7     2 high
## 116    6     0  low
## 117    3     0 high
## 118    3     0 high
```

```
## 119    7     3 high
## 120    7     2 high
## 121    7     3 high
## 122    7     4 high
## 123    3     0 high
## 124    7     3 high
## 125    3     1  low
## 126    3     0  low
## 127    7     3 high
## 128    7     3 high
## 129    3     0  low
## 130    3     0 high
## 131    7     0 high
## 132    7     0 high
## 133    3     0  low
## 134    3     0 high
## 135    3     0 high
## 136    3     0  low
## 137    7     4 high
## 138    7     3  low
## 139    7     1 high
## 140    3     0 high
## 141    3     0  low
## 142    7     1  low
## 143    3     0  low
## 144    7     1 high
## 145    7     0 high
## 146    3     1 high
## 147    7     4 high
## 148    3     0 high
## 149    3     0  low
## 150    3     0 high
## 151    7     0  low
## 152    3     0 high
## 153    7     0 high
## 154    7     4 high
## 155    3     3 high
## 156    3     1 high
## 157    7     1 high
## 158    7     1 high
## 159    7     2 high
## 160    7     0 high
## 161    7     0  low
## 162    3     4 high
## 163    3     0 high
## 164    3     0 high
## 165    3     0 high
## 166    7     0 high
## 168    3     0  low
## 169    7     1 high
## 170    3     0  low
## 171    7     3 high
```

```
## 172    7     0 high
## 173    3     1 high
## 174    3     0 high
## 175    6     4 high
## 176    7     1 high
## 177    7     0 high
## 178    6     1 high
## 179    3     0 high
## 180    3     0 high
## 181    7     3 high
## 182    7     2 high
## 183    3     0  low
## 184    7     0  low
## 185    3     1 high
## 186    3     0  low
## 187    7     0 high
## 188    6     2  low
## 189    7     1  low
## 190    7     2 high
## 191    3     0 high
## 192    7     3 high
## 194    3     2 high
## 195    3     0 high
## 196    3     3 high
## 197    3     0  low
## 198    3     0 high
## 199    3     0  low
## 200    3     1  low
## 201    3     0 high
## 202    3     0 high
## 203    7     0 high
## 204    7     0 high
## 205    7     0 high
## 206    7     3 high
## 207    7     3 high
## 208    7     3 high
## 209    3     0  low
## 210    3     1 high
## 211    3     0 high
## 212    7     4  low
## 213    3     0 high
## 214    7     3 high
## 215    3     1 high
## 216    7     0  low
## 217    3     0  low
## 218    3     0 high
## 219    3     0 high
## 220    3     0 high
## 221    3     0 high
## 222    3     0 high
## 223    3     0 high
## 224    7     3 high
```

```
## 225    3     1 high
## 226    3     0  low
## 227    3     0 high
## 228    3     0 high
## 229    3     3 high
## 230    3     2 high
## 231    3     0 high
## 232    3     2 high
## 233    3     1 high
## 234    3     0  low
## 235    3     0 high
## 236    3     3 high
## 237    7     2 high
## 238    7     1 high
## 239    3     0  low
## 240    3     0  low
## 241    3     0  low
## 242    3     0  low
## 243    3     0 high
## 244    3     2  low
## 245    3     0 high
## 246    3     2 high
## 247    7     2 high
## 248    3     1 high
## 249    7     3 high
## 250    3     0  low
## 251    6     0 high
## 252    7     1 high
## 253    7     0 high
## 254    3     0 high
## 255    3     0 high
## 256    3     0 high
## 257    3     0 high
## 258    3     0 high
## 259    3     0  low
## 260    7     1  low
## 261    3     0 high
## 262    3     3  low
## 263    3     0  low
## 264    3     0 high
## 265    3     4 high
## 266    6     2 high
## 268    6     2 high
## 269    7     1 high
## 270    3     0 high
## 271    7     1 high
## 272    6     0 high
## 273    7     2 high
## 274    3     0 high
## 275    3     1  low
## 276    7     0  low
## 277    3     0 high
```

```
## 278    3     0 high
## 279    3     1  low
## 280    3     0 high
## 281    7     2 high
## 282    3     0 high
## 283    7     3 high
## 284    3     0  low
## 285    7     2 high
## 286    6     4 high
## 287    6     2 high
## 289    7     0  low
## 290    3     0  low
## 291    7     1 high
## 292    3     0  low
## 293    6     2 high
## 294    7     2 high
## 295    3     1 high
## 296    3     0  low
## 297    6     3 high
## 298    7     1 high
## 299    7     1  low
## 300    7     2 high
## 301    7     3 high
## 302    3     1  low
```

3.2

Train a classification tree on a 240 observation training subset (using the seed I have set for you). Plot the tree.

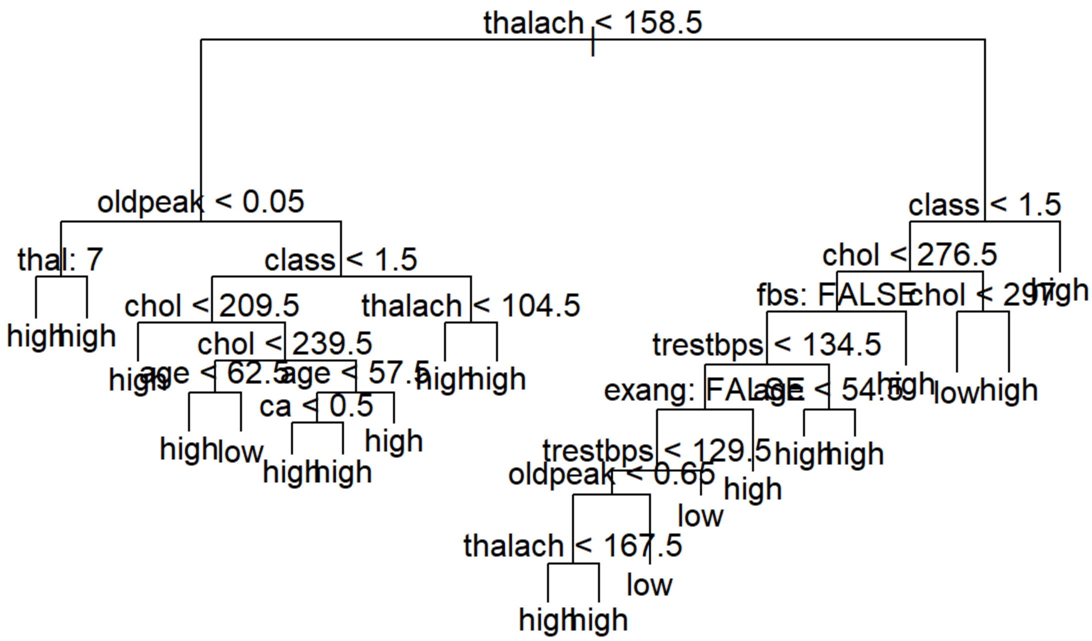
```
set.seed(101)
sample_t <- sample(1:nrow(heart), 0.81 * nrow(heart))
train_h <- heart[sample_t,]
test_h <- heart[-sample_t,]
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.2.3
```

```
## Loading required package: rpart
```

```
library(rpart)
heart.tree <- tree(high ~. -cp, data = heart, subset = sample_t)
par(xpd = NA) # otherwise on some devices the text is clipped

plot(heart.tree)
text(heart.tree, pretty = 0)
```



3.3

Use the trained model to classify the remaining testing points. Create a confusion matrix to evaluate performance. Report the classification error rate.

```
tree.pred <- predict(heart.tree, test_h, type = "class")
conf_matrix <- table(True = test_h$high, Predicted = tree.pred)
print(conf_matrix)
```

```
##       Predicted
## True   high low
##   high   37   3
##   low    16   1
```

```
38/(38+19)
```

```
## [1] 0.6666667
```

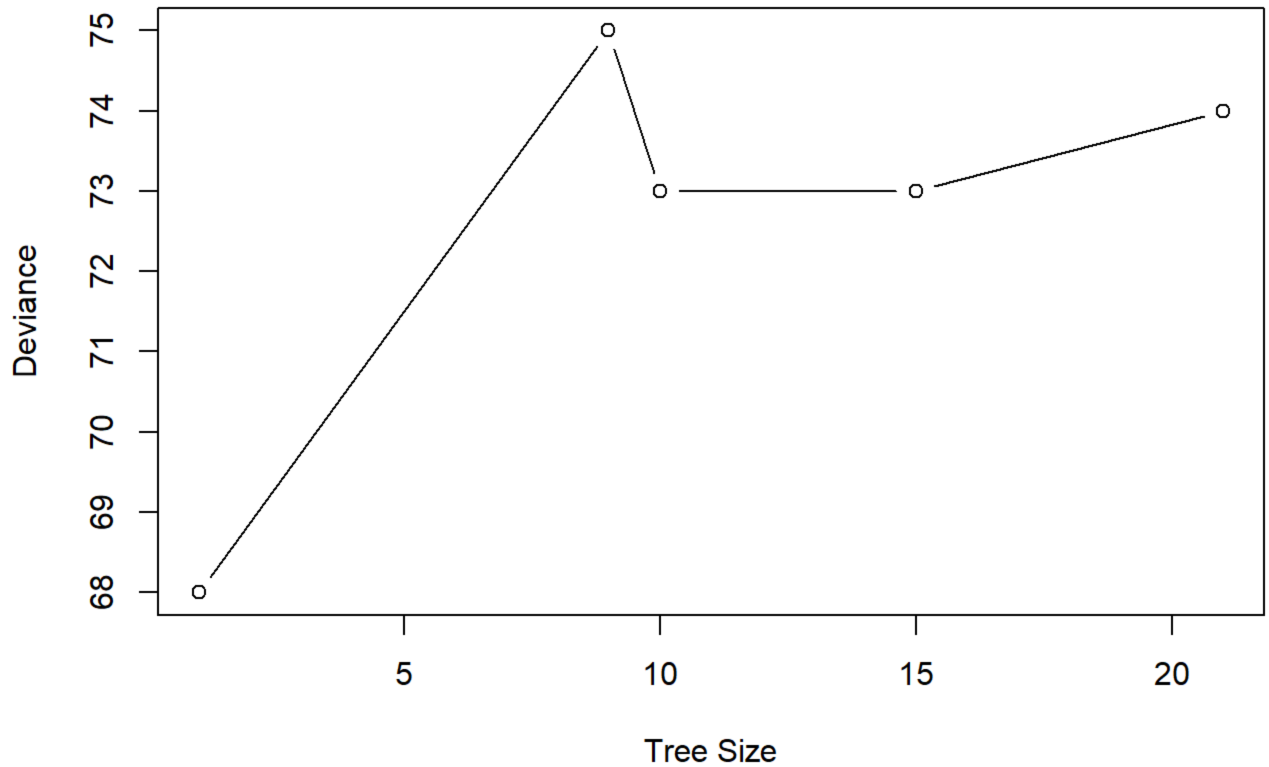
3.4

Above we have a fully grown (bushy) tree. Now, cross validate it using the `cv.tree` command. Specify cross validation to be done according to the misclassification rate. Choose an ideal number of splits, and plot this tree. Finally, use this pruned tree to test on the testing set. Report a confusion matrix and the misclassification rate.

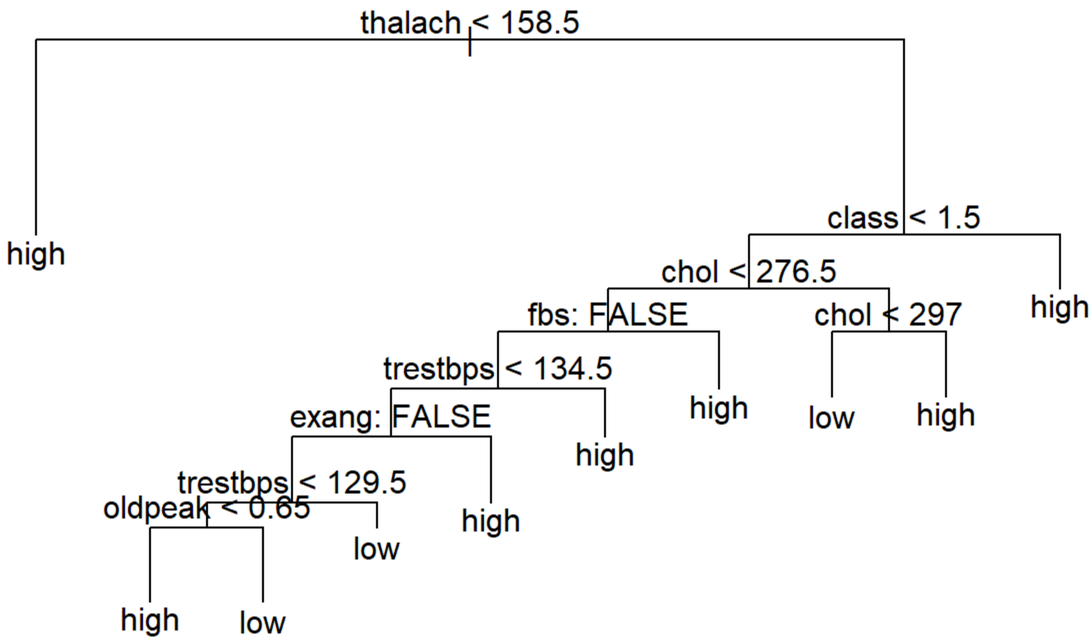
```
set.seed(101)
cv.heart <- cv.tree(heart.tree, FUN = prune.misclass)
print(cv.heart)
```

```
## $size
## [1] 21 15 10  9  1
##
## $dev
## [1] 74 73 73 75 68
##
## $k
## [1] -Inf 0.00 0.20 1.00 2.25
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv.heart$size, cv.heart$dev, type = "b", xlab = "Tree Size", ylab = "Deviance")
```




```
prune.heart_tree <- prune.misclass(heart.tree, best = 10)
plot(prune.heart_tree)
text(prune.heart_tree, pretty=0)
```




```
tree.pred_2 <- predict(prune.heart_tree, newdata = test_h, type = "class")
conf_matrix <- table(True = test_h$high, Predicted = tree.pred_2)
print(conf_matrix)
```

```
##       Predicted
## True   high low
##   high   38   2
##   low    17   0
```

```
tree.pred <- predict(heart.tree, test_h, type = "class")
conf_matrix <- table(True = test_h$high, Predicted = tree.pred_2)
print(conf_matrix)
```

```
##       Predicted
## True   high low
##   high   38   2
##   low    17   0
```

```
38/(38+19)
```

```
## [1] 0.6666667
```

3.5

Discuss the trade-off in accuracy and interpretability in pruning the above tree.

Student Input In this instance there was no differnece between misclassification rate before and after pruning, this was able to help us classify without needing such a long tree, and makes the tree easier to use/work with. However, typically long overfit trees do not test well, and this new one while shorter and easier to interpret, may be more generalizeable.
##

Discuss the ways a decision tree could manifest algorithmic bias.
Decision trees mostly manifest algorithmic bias through fully grown trees which are not applicable and overfit the test data, or training data which does not reflect the true data.
Student Answer

---

1. Remember this is a parameter that decides how smooth your decision boundary should be↩