Exercise #9

Classification and Regression Trees

Brian Schweigler; 16-102-071

19/05/2022

Preliminaries

Load the required libraries

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.0.5
```

Set a seed for later:

```
set.seed(1786397)
```

Loading the low weight dataset, set Status as a factor and show an overview:

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
weight_df = read.csv("LowWeight.txt", sep = "\t", header = TRUE)
summary(weight_df)
```

```
##        id            low_bw            age        mother_weight
##  Min.   :  4.0   Min.   :0.0000   Min.   :14.00   Min.   : 80.0
##  1st Qu.: 68.0   1st Qu.:0.0000   1st Qu.:19.00   1st Qu.:110.0
##  Median :123.0   Median :0.0000   Median :23.00   Median :121.0
##  Mean   :121.1   Mean   :0.3122   Mean   :23.24   Mean   :129.8
##  3rd Qu.:176.0   3rd Qu.:1.0000   3rd Qu.:26.00   3rd Qu.:140.0
##  Max.   :226.0   Max.   :1.0000   Max.   :45.00   Max.   :250.0
##       race       smoking_status   premat_labour     hypertension
##  Min.   :1.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :1.000   Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   :1.847   Mean   :0.3915   Mean   :0.1958   Mean   :0.06349
##  3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :3.000   Max.   :1.0000   Max.   :3.0000   Max.   :1.00000
##  uterine_irrit        visits        birth_weight
##  Min.   :0.0000   Min.   :0.0000   Min.   : 709
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:2414
##  Median :0.0000   Median :0.0000   Median :2977
##  Mean   :0.1481   Mean   :0.7937   Mean   :2945
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:3475
##  Max.   :1.0000   Max.   :6.0000   Max.   :4990
```

No NAs; the range of the values can't be gauged without further information.

Loading the heart dataset:

```
heart_df = read.csv("Heart.txt",
                                        header = TRUE,
                                        sep = "\t",
                                        comment.char = "#")
summary(heart_df)
```

```
##        ID              age              sex              pain      
##  Min.   :  1.00   Min.   : 3.00   Min.   :0.0000   Min.   :1.000  
##  1st Qu.: 68.75   1st Qu.:47.75   1st Qu.:0.0000   1st Qu.:3.000  
##  Median :136.50   Median :55.00   Median :1.0000   Median :3.000  
##  Mean   :136.50   Mean   :54.24   Mean   :0.6765   Mean   :3.173  
##  3rd Qu.:204.25   3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000  
##  Max.   :272.00   Max.   :77.00   Max.   :1.0000   Max.   :4.000  
##       pres        cholesterol        sugar           electro     
##  Min.   : 94.0   Min.   :125.0   Min.   :0.0000   Min.   :0.000  
##  1st Qu.:120.0   1st Qu.:212.8   1st Qu.:0.0000   1st Qu.:0.000  
##  Median :130.0   Median :245.0   Median :0.0000   Median :2.000  
##  Mean   :131.3   Mean   :249.3   Mean   :0.1471   Mean   :1.029  
##  3rd Qu.:140.0   3rd Qu.:278.0   3rd Qu.:0.0000   3rd Qu.:2.000  
##  Max.   :200.0   Max.   :564.0   Max.   :1.0000   Max.   :2.000  
##    gramstein           rate           angina            fiss      
##  Min.   :-4.500   Min.   : 71.0   Min.   :0.0000   Min.   :11.00  
##  1st Qu.: 9.300   1st Qu.:132.8   1st Qu.:0.0000   1st Qu.:22.00  
##  Median :10.100   Median :153.5   Median :0.0000   Median :25.00  
##  Mean   : 9.975   Mean   :149.6   Mean   :0.3346   Mean   :24.94  
##  3rd Qu.:10.700   3rd Qu.:166.0   3rd Qu.:1.0000   3rd Qu.:28.00  
##  Max.   :13.300   Max.   :202.0   Max.   :1.0000   Max.   :39.00  
##       peak          slope          vessels            thal      
##  Min.   :0.00   Min.   :1.000   Min.   :0.0000   Min.   :3.000  
##  1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:3.000  
##  Median :0.80   Median :2.000   Median :0.0000   Median :3.000  
##  Mean   :1.05   Mean   :1.588   Mean   :0.6765   Mean   :4.713  
##  3rd Qu.:1.65   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:7.000  
##  Max.   :6.20   Max.   :3.000   Max.   :3.0000   Max.   :7.000  
##       blst          disease     
##  Min.   :50.14   Min.   :1.000  
##  1st Qu.:57.50   1st Qu.:1.000  
##  Median :66.01   Median :1.000  
##  Mean   :65.28   Mean   :1.449  
##  3rd Qu.:71.88   3rd Qu.:2.000  
##  Max.   :79.77   Max.   :2.000
```

No NAs; the range of the values can’t be gauged without further information.

1a. Create a regression tree using the variable birth_weight as a target. Plot the resulting model.

First we will remove the data that is not of interest in our predictions; being the ID and whether it is low birth weight (which is what we ant to predict)

```
useful_weight <- weight_df[,-(1:2)]
summary(useful_weight)
```

```
##       age        mother_weight        race       smoking_status  
##  Min.   :14.00   Min.   : 80.0   Min.   :1.000   Min.   :0.0000  
##  1st Qu.:19.00   1st Qu.:110.0   1st Qu.:1.000   1st Qu.:0.0000
##  Median :23.00   Median :121.0   Median :1.000   Median :0.0000
##  Mean   :23.24   Mean   :129.8   Mean   :1.847   Mean   :0.3915
##  3rd Qu.:26.00   3rd Qu.:140.0   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :45.00   Max.   :250.0   Max.   :3.000   Max.   :1.0000
##  premat_labour     hypertension     uterine_irrit        visits
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.00000   Median :0.0000   Median :0.0000
##  Mean   :0.1958   Mean   :0.06349   Mean   :0.1481   Mean   :0.7937
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :3.0000   Max.   :1.00000   Max.   :1.0000   Max.   :6.0000
##   birth_weight
##  Min.   : 709
##  1st Qu.:2414
##  Median :2977
##  Mean   :2945
##  3rd Qu.:3475
##  Max.   :4990
```

Now we divide this into test (30%) and train (70%) data set:

```
training = round(nrow(useful_weight) * 0.7)
training_index = sample(c(1:nrow(useful_weight)), training)
training_data = useful_weight[training_index,]
summary(training_data)
```

```
##       age        mother_weight        race       smoking_status
##  Min.   :14.00   Min.   : 80.0   Min.   :1.000   Min.   :0.0000
##  1st Qu.:19.00   1st Qu.:110.0   1st Qu.:1.000   1st Qu.:0.0000
##  Median :22.00   Median :120.0   Median :2.000   Median :0.0000
##  Mean   :22.61   Mean   :127.8   Mean   :1.902   Mean   :0.4242
##  3rd Qu.:25.00   3rd Qu.:140.0   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :35.00   Max.   :241.0   Max.   :3.000   Max.   :1.0000
##  premat_labour     hypertension     uterine_irrit        visits
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.00000   Median :0.0000   Median :0.0000
##  Mean   :0.2121   Mean   :0.07576   Mean   :0.1439   Mean   :0.7424
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :3.0000   Max.   :1.00000   Max.   :1.0000   Max.   :4.0000
##   birth_weight
##  Min.   : 709
##  1st Qu.:2381
##  Median :2913
##  Mean   :2869
##  3rd Qu.:3384
##  Max.   :4174
```

```
testing_data = useful_weight[-training_index,]
summary(testing_data)
```

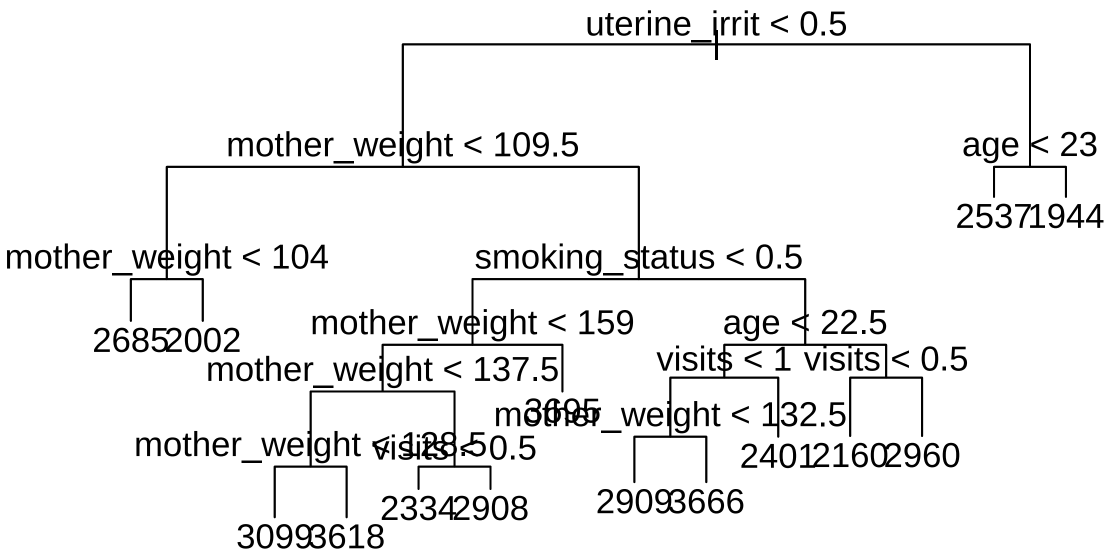
```
##       age        mother_weight        race       smoking_status
##  Min.   :14.00   Min.   : 89.0   Min.   :1.000   Min.   :0.0000
##  1st Qu.:20.00   1st Qu.:112.0   1st Qu.:1.000   1st Qu.:0.0000
##  Median :23.00   Median :125.0   Median :1.000   Median :0.0000
##  Mean   :24.68   Mean   :134.5   Mean   :1.719   Mean   :0.3158  
##  3rd Qu.:28.00   3rd Qu.:147.0   3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :45.00   Max.   :250.0   Max.   :3.000   Max.   :1.0000  
##  premat_labour     hypertension     uterine_irrit        visits      
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.00000   Median :0.0000   Median :1.0000  
##  Mean   :0.1579   Mean   :0.03509   Mean   :0.1579   Mean   :0.9123  
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:1.0000  
##  Max.   :2.0000   Max.   :1.00000   Max.   :1.0000   Max.   :6.0000  
##   birth_weight 
##  Min.   :1588  
##  1st Qu.:2750  
##  Median :3104  
##  Mean   :3120  
##  3rd Qu.:3629  
##  Max.   :4990  
```

Now we can perform the regression tree model:

```
reg_tree_model <-
    tree(birth_weight ~ ., training_data, split = "deviance")
summary(reg_tree_model)
```

```
## 
## Regression tree:
## tree(formula = birth_weight ~ ., data = training_data, split = "deviance")
## Variables actually used in tree construction:
## [1] "uterine_irrit"  "mother_weight"  "smoking_status" "visits"        
## [5] "age"           
## Number of terminal nodes:  14 
## Residual mean deviance:  288700 = 34060000 / 118 
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## -1398.0  -333.4    12.5     0.0   359.8  1693.0 
```

```
plot(reg_tree_model)
text(reg_tree_model, pretty = 0, cex = 1.1)
```



Here we have a tree, that will definitely need pruning.

2. Calculate the train and test MSE. Describe the results you obtained.

This is quite straightforward for both sets:

```
training_predictions <-
    predict(reg_tree_model, training_data, type = "vector")
training_MSE = mean ((training_predictions - training_data$birth_weight) ^
                                            2)
training_MSE
```

```
## [1] 258053.2
```

```
test_predictions <-
    predict(reg_tree_model, testing_data, type = "vector")
test_MSE = mean ((test_predictions - testing_data$birth_weight) ^ 2)
test_MSE
```

```
## [1] 681214
```

As the magnitude of the distance is data dependant, we can only say that the training MSE is of a factor 2.5 smaller than the MSE of the test predictions.

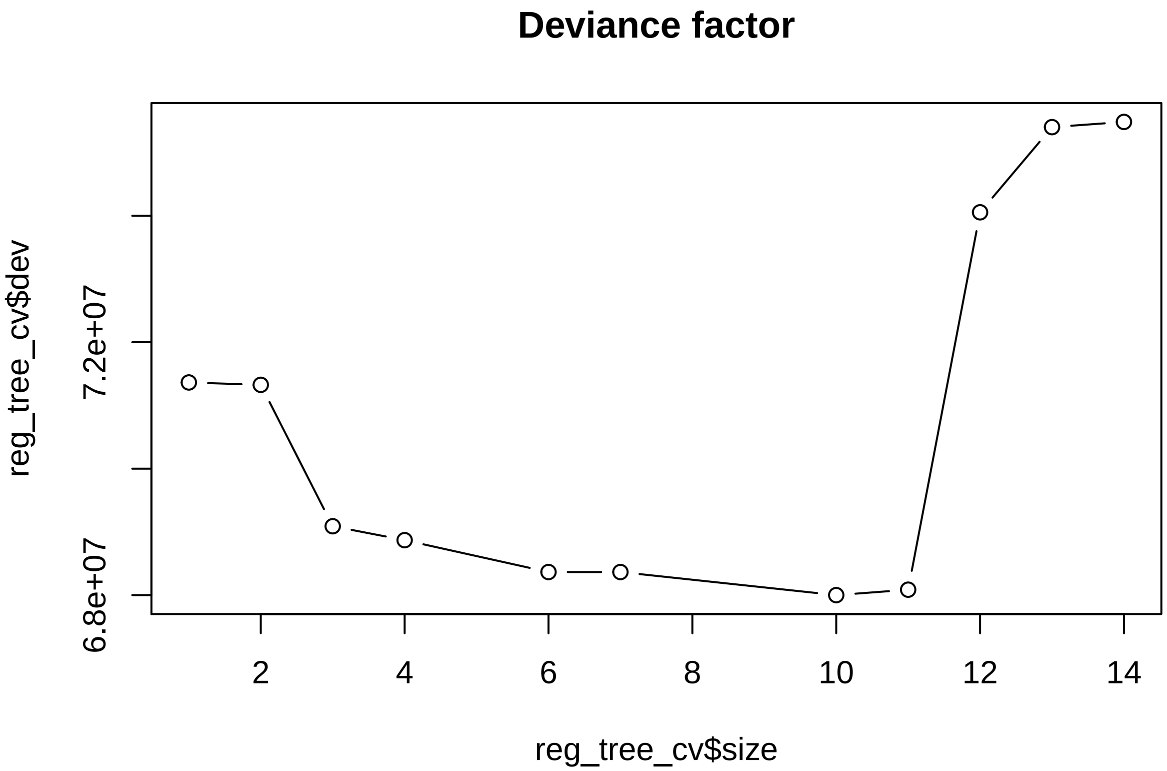
3. Should your regression tree be pruned? If yes, which strategy would you use? Compare the previous test MSE with the one obtained with the pruned tree. Plot the new model.

Yes, definitely needs to be pruned (and will improve Test MSE as well):

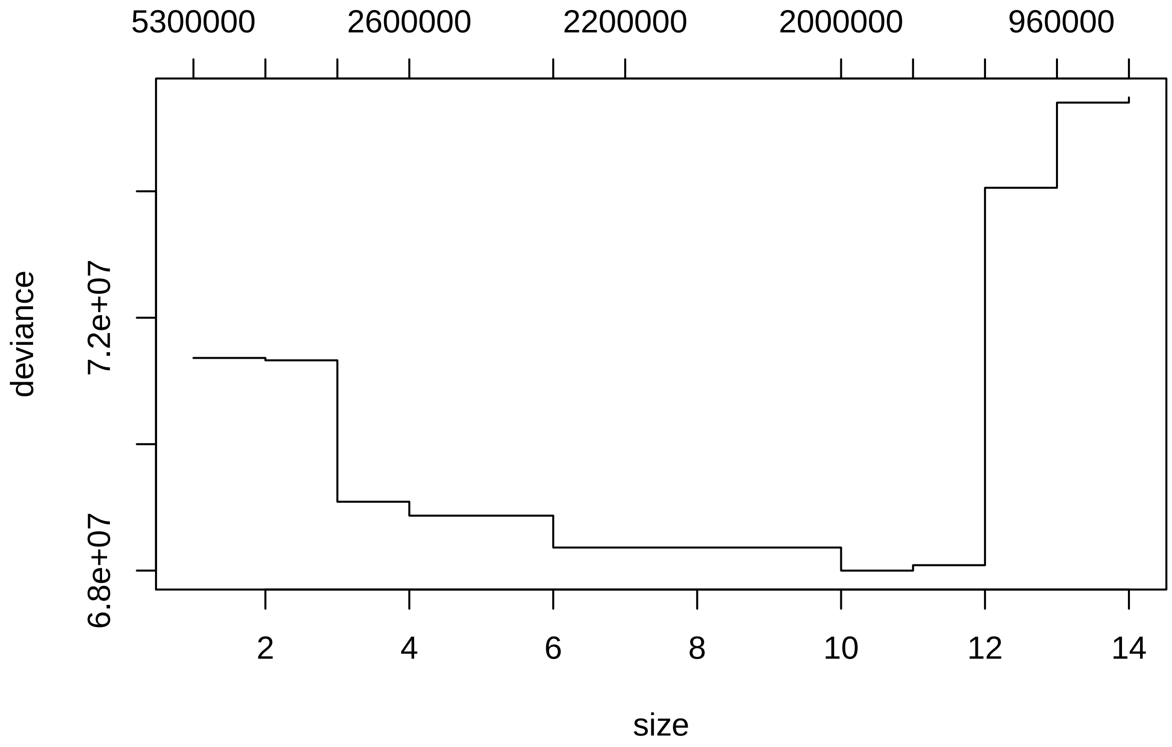
```
reg_tree_cv <- cv.tree(reg_tree_model, K = 10)
reg_tree_cv
```

```
## $size
##  [1] 14 13 12 11 10  7  6  4  3  2  1
##
## $dev
##  [1] 75484756 75402480 74055229 68085436 67999676 68364811 68364811 68869252
##  [9] 69089187 71326023 71363536
##
## $k
##  [1]      -Inf  962311.5 1295424.1 1802468.0 1961496.5 2163848.6 2185691.7
##  [8] 2631874.4 2836538.1 4849571.4 5291856.6
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
dev_min = which(reg_tree_cv$dev == min(reg_tree_cv$dev))
dev_min_size = reg_tree_cv$size[dev_min]
plot(reg_tree_cv$size,
         reg_tree_cv$dev,
         main = "Deviance factor",
         type = "b")
```



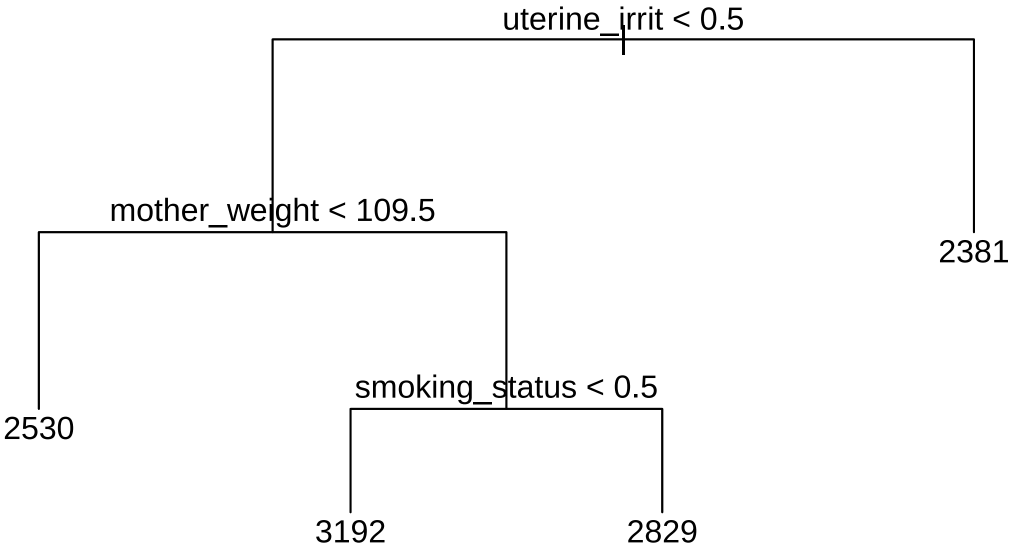
```
plot(reg_tree_cv)
```



```
reg_tree_pruned <-  prune.tree(reg_tree_model, best = 4)
summary(reg_tree_pruned)
```

```
##
## Regression tree:
## snip.tree(tree = reg_tree_model, nodes = c(3L, 4L, 11L, 10L))
## Variables actually used in tree construction:
## [1] "uterine_irrit"  "mother_weight"  "smoking_status"
## Number of terminal nodes:  4
## Residual mean deviance:  422100 = 54030000 / 128
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1694.000  -416.500    -3.992     0.000   513.900  1467.000
```

```
plot(reg_tree_pruned)
text(reg_tree_pruned, pretty = 0)
```



```
test_predictions_2 <-
    predict(reg_tree_pruned, testing_data, type = "vector")
test_MSE_2 = mean ((test_predictions_2 - testing_data$birth_weight) ^ 2)
test_MSE_2
```

```
## [1] 470319.2
```

Comparing with the unpruned, we definitely have an improved MSE now:

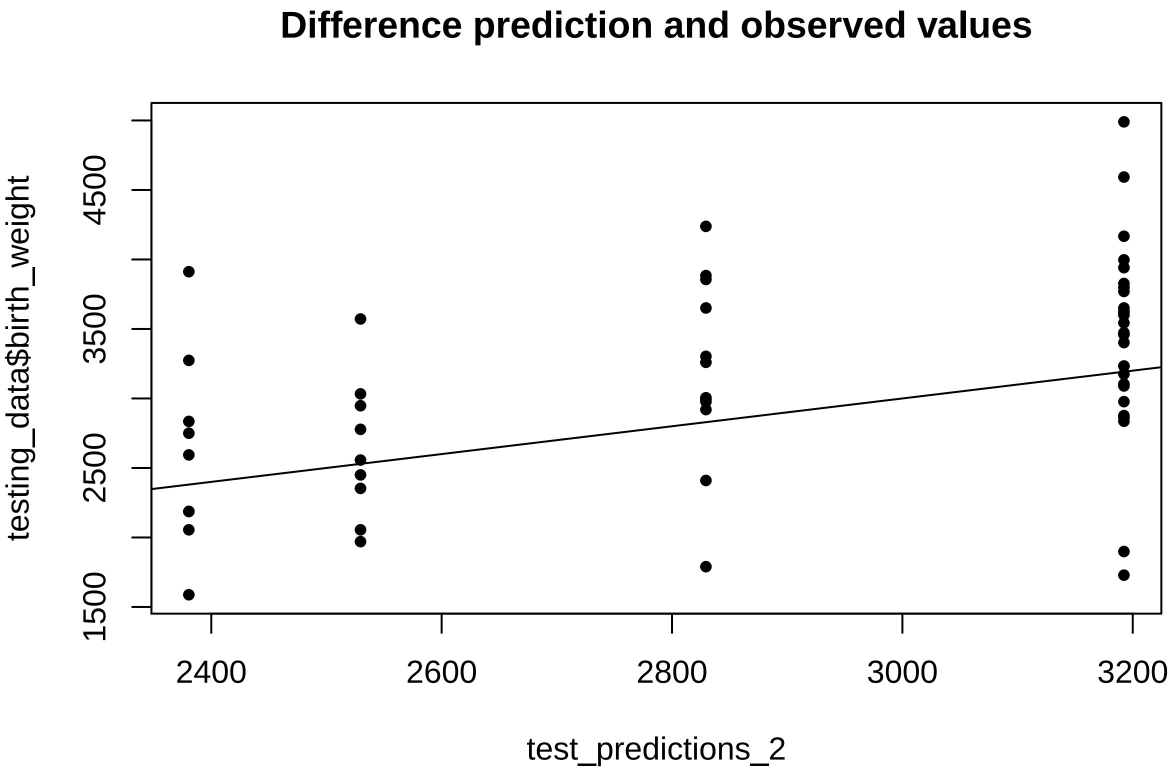
```
test_MSE
```

```
## [1] 681214
```

```
test_MSE_2
```

```
## [1] 470319.2
```

```
plot(test_predictions_2, testing_data$birth_weight,
     main="Difference prediction and observed values",
     pch=20)
abline(0,1)
aMean <-  sqrt(mean((test_predictions_2 - testing_data$birth_weight)^2))
abline(h = test_MSE_2, lty="dotted", col="red")
```


Difference prediction and observed values
testing_data$birth_weight
1500
2500
3500
4500
2400
2600
2800
3000
3200
test_predictions_2

4. Create a classification tree using the variable disease as a target. Plot the resulting model.

First we will factorize disease (among others), and remove ID.

```
heart_df$disease <-
    factor(
        heart_df$disease,
        levels = c(1, 2),
        labels = c("A", "P")
    )

heart_df$sex <-
    factor(
        heart_df$sex,
        levels = c(0, 1),
        labels = c("0", "1")
    )

heart_df$sugar <-
    factor(
        heart_df$sugar,
        levels = c(0, 1),
        labels = c("0", "1")
    )

useful_heart <- heart_df[, -1]
summary(useful_heart)
```

```
##       age         sex          pain            pres        cholesterol
##  Min.   : 3.00   0: 88   Min.   :1.000   Min.   : 94.0   Min.   :125.0
##  1st Qu.:47.75   1:184   1st Qu.:3.000   1st Qu.:120.0   1st Qu.:212.8
##  Median :55.00           Median :3.000   Median :130.0   Median :245.0
##  Mean   :54.24           Mean   :3.173   Mean   :131.3   Mean   :249.3
##  3rd Qu.:61.00           3rd Qu.:4.000   3rd Qu.:140.0   3rd Qu.:278.0
##  Max.   :77.00           Max.   :4.000   Max.   :200.0   Max.   :564.0
##  sugar      electro        gramstein           rate           angina
##  0:232   Min.   :0.000   Min.   :-4.500   Min.   : 71.0   Min.   :0.0000
##  1: 40   1st Qu.:0.000   1st Qu.: 9.300   1st Qu.:132.8   1st Qu.:0.0000
##          Median :2.000   Median :10.100   Median :153.5   Median :0.0000
##          Mean   :1.029   Mean   : 9.975   Mean   :149.6   Mean   :0.3346
##          3rd Qu.:2.000   3rd Qu.:10.700   3rd Qu.:166.0   3rd Qu.:1.0000
##          Max.   :2.000   Max.   :13.300   Max.   :202.0   Max.   :1.0000
##       fiss            peak          slope          vessels
##  Min.   :11.00   Min.   :0.00   Min.   :1.000   Min.   :0.0000
##  1st Qu.:22.00   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :25.00   Median :0.80   Median :2.000   Median :0.0000
##  Mean   :24.94   Mean   :1.05   Mean   :1.588   Mean   :0.6765
##  3rd Qu.:28.00   3rd Qu.:1.65   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :39.00   Max.   :6.20   Max.   :3.000   Max.   :3.0000
##       thal            blst       disease
##  Min.   :3.000   Min.   :50.14   A:150
##  1st Qu.:3.000   1st Qu.:57.50   P:122
##  Median :3.000   Median :66.01
##  Mean   :4.713   Mean   :65.28
##  3rd Qu.:7.000   3rd Qu.:71.88  
##  Max.   :7.000   Max.   :79.77  
```

Now we divide this into test (30%) and train (70%) data set:

```
training_h = round(nrow(useful_heart) * 0.7)
training_index_h = sample(c(1:nrow(useful_heart)), training_h)
training_data_h = useful_heart[training_index_h,]
summary(training_data_h)
```

```
##       age         sex          pain          pres        cholesterol    sugar  
##  Min.   : 3.00   0: 59   Min.   :1.0   Min.   : 94.0   Min.   :125.0   0:163  
##  1st Qu.:48.00   1:131   1st Qu.:3.0   1st Qu.:120.0   1st Qu.:214.0   1: 27  
##  Median :54.00           Median :3.0   Median :129.5   Median :244.0          
##  Mean   :53.93           Mean   :3.2   Mean   :129.9   Mean   :250.4          
##  3rd Qu.:60.00           3rd Qu.:4.0   3rd Qu.:140.0   3rd Qu.:282.0          
##  Max.   :77.00           Max.   :4.0   Max.   :200.0   Max.   :564.0          
##     electro         gramstein            rate           angina      
##  Min.   :0.0000   Min.   :-4.500   Min.   : 71.0   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.: 9.325   1st Qu.:132.0   1st Qu.:0.0000  
##  Median :0.0000   Median :10.100   Median :154.0   Median :0.0000  
##  Mean   :0.9316   Mean   : 9.992   Mean   :149.2   Mean   :0.3316  
##  3rd Qu.:2.0000   3rd Qu.:10.700   3rd Qu.:166.8   3rd Qu.:1.0000  
##  Max.   :2.0000   Max.   :13.300   Max.   :194.0   Max.   :1.0000  
##       fiss            peak           slope          vessels      
##  Min.   :11.00   Min.   :0.000   Min.   :1.000   Min.   :0.0000  
##  1st Qu.:21.00   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.0000  
##  Median :24.00   Median :0.800   Median :2.000   Median :0.0000  
##  Mean   :24.71   Mean   :1.103   Mean   :1.589   Mean   :0.6895  
##  3rd Qu.:28.00   3rd Qu.:1.800   3rd Qu.:2.000   3rd Qu.:1.0000  
##  Max.   :39.00   Max.   :6.200   Max.   :3.000   Max.   :3.0000  
##       thal            blst         disease
##  Min.   :3.000   Min.   :50.21   A:99   
##  1st Qu.:3.000   1st Qu.:57.22   P:91   
##  Median :3.000   Median :65.74          
##  Mean   :4.705   Mean   :65.13          
##  3rd Qu.:7.000   3rd Qu.:72.66          
##  Max.   :7.000   Max.   :79.77          
```

```
testing_data_h = useful_heart[-training_index_h,]
summary(testing_data_h)
```

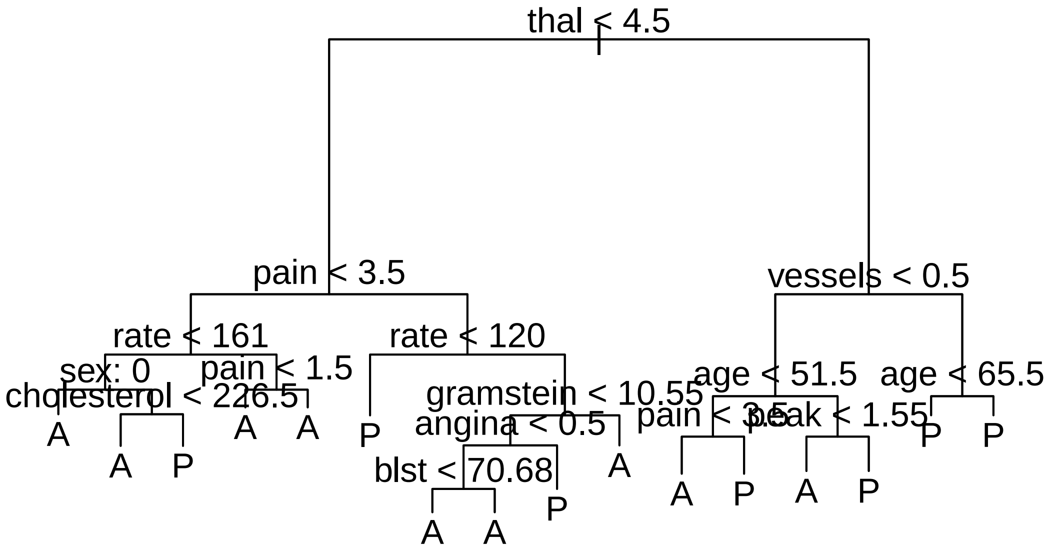
```
##       age         sex         pain           pres        cholesterol    sugar 
##  Min.   :29.00   0:29   Min.   :1.00   Min.   : 94.0   Min.   :126.0   0:69  
##  1st Qu.:47.50   1:53   1st Qu.:2.00   1st Qu.:123.2   1st Qu.:210.2   1:13  
##  Median :56.00          Median :3.00   Median :132.0   Median :249.0         
##  Mean   :54.98          Mean   :3.11   Mean   :134.5   Mean   :246.5         
##  3rd Qu.:61.75          3rd Qu.:4.00   3rd Qu.:141.5   3rd Qu.:275.5         
##  Max.   :76.00          Max.   :4.00   Max.   :192.0   Max.   :417.0         
##     electro        gramstein           rate           angina      
##  Min.   :0.000   Min.   : 6.900   Min.   : 95.0   Min.   :0.0000  
##  1st Qu.:0.000   1st Qu.: 9.200   1st Qu.:137.2   1st Qu.:0.0000  
##  Median :2.000   Median : 9.900   Median :152.5   Median :0.0000  
##  Mean   :1.256   Mean   : 9.934   Mean   :150.4   Mean   :0.3415  
##  3rd Qu.:2.000   3rd Qu.:10.800   3rd Qu.:163.0   3rd Qu.:1.0000  
##  Max.   :2.000   Max.   :12.100   Max.   :202.0   Max.   :1.0000  
##       fiss            peak            slope          vessels      
##  Min.   :14.00   Min.   :0.0000   Min.   :1.000   Min.   :0.0000  
##  1st Qu.:23.00   1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.0000  
##  Median :25.00   Median :0.5500   Median :2.000   Median :0.0000  
##  Mean   :25.48   Mean   :0.9293   Mean   :1.585   Mean   :0.6463  
##  3rd Qu.:29.00   3rd Qu.:1.5750   3rd Qu.:2.000   3rd Qu.:1.0000  
##  Max.   :35.00   Max.   :4.2000   Max.   :3.000   Max.   :3.0000  
##       thal            blst       disease
##  Min.   :3.000   Min.   :50.14   A:51   
##  1st Qu.:3.000   1st Qu.:57.92   P:31   
##  Median :3.000   Median :67.61          
##  Mean   :4.732   Mean   :65.63          
##  3rd Qu.:7.000   3rd Qu.:71.62          
##  Max.   :7.000   Max.   :79.66          
```

Now we can perform the regression tree model:

```
reg_tree_model_h <-
    tree(as.factor(disease) ~ ., training_data_h, split = "deviance")
summary(reg_tree_model_h)
```

```
## 
## Classification tree:
## tree(formula = as.factor(disease) ~ ., data = training_data_h, 
##     split = "deviance")
## Variables actually used in tree construction:
##  [1] "thal"        "pain"        "rate"        "sex"         "cholesterol"
##  [6] "gramstein"   "angina"      "blst"        "vessels"     "age"        
## [11] "peak"       
## Number of terminal nodes:  16 
## Residual mean deviance:  0.3625 = 63.07 / 174 
## Misclassification error rate: 0.07368 = 14 / 190
```

```
plot(reg_tree_model_h)
text(reg_tree_model_h, pretty = 0, cex = 1.1)
```


thal < 4.5
pain < 3.5
vessels < 0.5
rate < 161
rate < 120
age < 51.5
age < 65.5
sex: 0
pain < 1.5
cholesterol < 226.5
gramstein < 10.55
angina < 0.5
pain < 3.5
peak < 1.55
blst < 70.68
A
A
P
A
A
P
A
A
A
P
A
A
P
A
P
P
P

5. Compute the confusion matrix for your model and calculate the accuracy, sensitivity and specificity. Describe the results you obtained.

First we'll need to make some predictions:

```
Test_Output = predict(reg_tree_model_h, testing_data_h, type = "class")
Test_Error = mean(Test_Output != testing_data_h$disease)
Test_Error
```

```
## [1] 0.3292683
```

```
confusion_mat_h <-
    table(testing_data_h$disease, Test_Output)[2:1, 2:1]
confusion_mat_h
```

```
##    Test_Output
##      P  A
##   P 24  7
##   A 20 31
```

```
TP = confusion_mat_h[1]
TN = confusion_mat_h[4]
FP = confusion_mat_h[2]
FN = confusion_mat_h[3]

precision = TP / (TP + FP)
print(sprintf("Precision = %f", precision))
```

```
## [1] "Precision = 0.545455"
```

```
recall = TP / (TP + FN)
print(sprintf("Recall a.k.a. Sensitivity = %f", recall))
```

```
## [1] "Recall a.k.a. Sensitivity = 0.774194"
```

```
specificity = TN / (FP + TN)
print(sprintf("Specifcity = %f", specificity))
```

```
## [1] "Specifcity = 0.607843"
```

```
F1 = (2 * recall * precision) / (recall + precision)
print(sprintf("F1 measure = %f", F1))
```

```
## [1] "F1 measure = 0.640000"
```

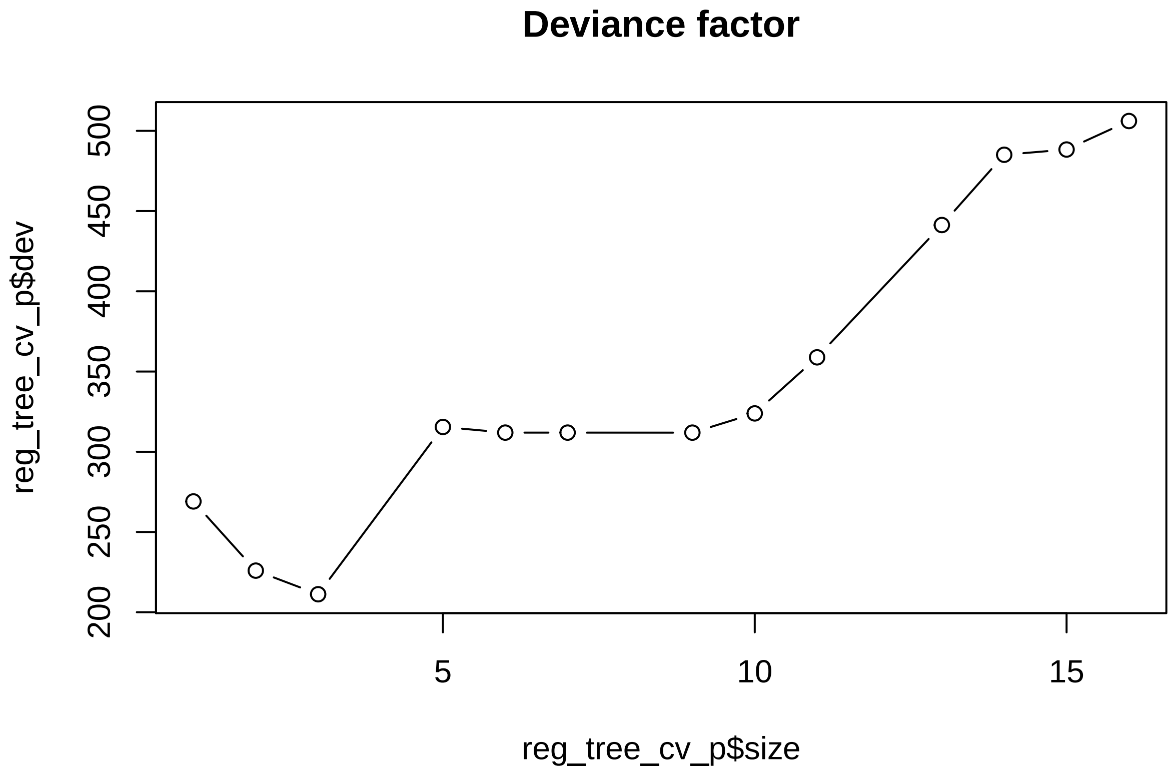
6. Should your classification tree be pruned? If yes, which strategy would you use? Compare the previous results with the one obtained with the pruned tree. Plot the new model.

Yes, definitely should be pruned.

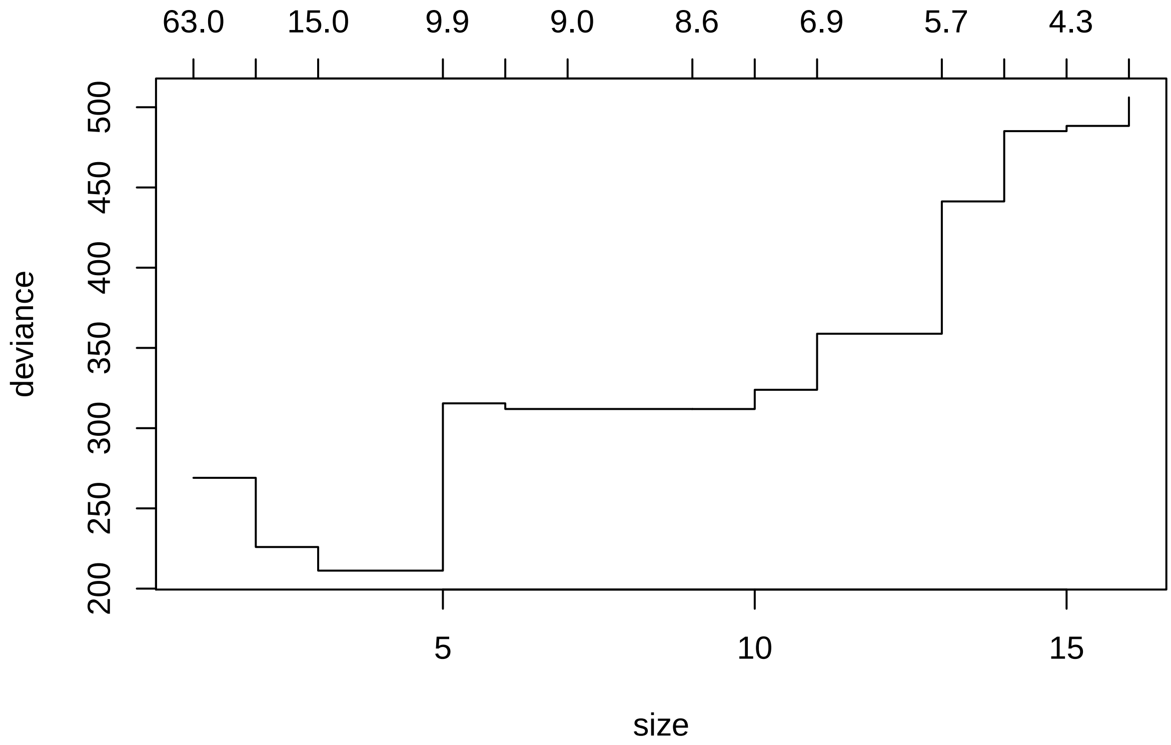
```
reg_tree_cv_p <- cv.tree(reg_tree_model_h, K = 10)
reg_tree_cv_p
```

```
## $size
##  [1] 16 15 14 13 11 10  9  7  6  5  3  2  1
##
## $dev
##  [1] 506.1109 488.3555 485.0909 441.3029 358.8499 323.9147 311.9191 311.9342
##  [9] 311.9342 315.4582 211.2155 225.9188 269.0511
##
## $k
##  [1]      -Inf  4.348524  4.674842  5.727542  6.898485  8.439046  8.612810
##  [8]  9.025632  9.113858  9.918374 14.861525 25.023762 62.555503
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
dev_min_p = which(reg_tree_cv_p$dev == min(reg_tree_cv_p$dev))
dev_min_size_p = reg_tree_cv_p$size[dev_min_p]
plot(reg_tree_cv_p$size,
         reg_tree_cv_p$dev,
         main = "Deviance factor",
         type = "b")
```



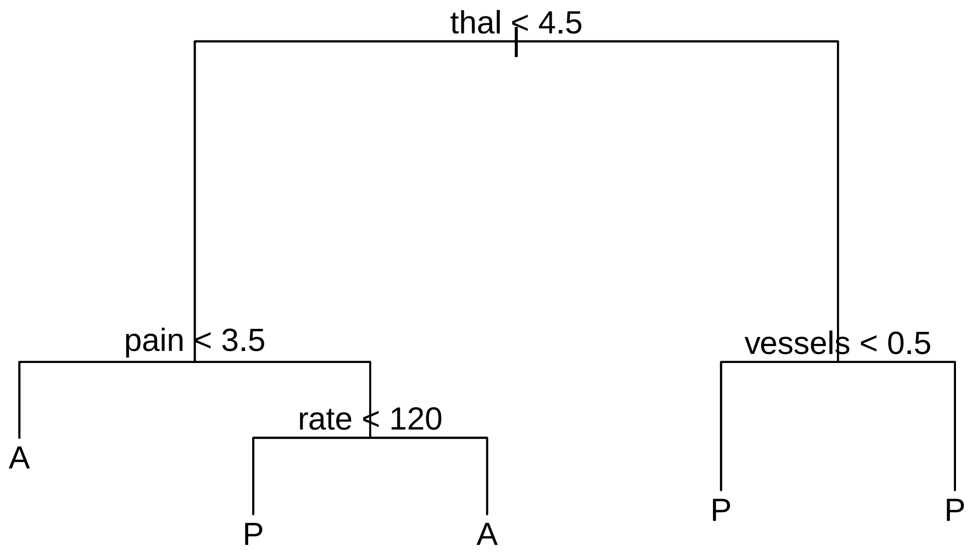
```
plot(reg_tree_cv_p)
```



```
reg_tree_pruned_p <-  prune.tree(reg_tree_model_h, best = 4)
summary(reg_tree_pruned_p)
```

```
##
## Classification tree:
## snip.tree(tree = reg_tree_model_h, nodes = c(7L, 4L, 11L, 6L))
## Variables actually used in tree construction:
## [1] "thal"    "pain"    "rate"    "vessels"
## Number of terminal nodes:  5
## Residual mean deviance:  0.7879 = 145.8 / 185
## Misclassification error rate: 0.1789 = 34 / 190
```

```
plot(reg_tree_pruned_p)
text(reg_tree_pruned_p, pretty = 0)
```



Now for some predictions first:

But first, some predictions:

```
Test_Output_2 = predict(reg_tree_pruned_p, testing_data_h, type = "class")
Test_Error_2 = mean(Test_Output_2 != testing_data_h$disease)
Test_Error_2
```

```
## [1] 0.2560976
```

```
confusion_mat_p <-
    table(testing_data_h$disease, Test_Output_2)[2:1, 2:1]
confusion_mat_p
```

```
##    Test_Output_2
##      P  A
##   P 24  7
##   A 14 37
```

```
TP_p = confusion_mat_p[1]
TN_p = confusion_mat_p[4]
FP_p = confusion_mat_p[2]
FN_p = confusion_mat_p[3]

precision_p = TP_p / (TP_p + FP_p)
print(sprintf("Precision = %f", precision_p))
```

```
## [1] "Precision = 0.631579"
```

```
recall_p = TP_p / (TP_p + FN_p)
print(sprintf("Recall a.k.a. Sensitivity = %f", recall_p))
```

```
## [1] "Recall a.k.a. Sensitivity = 0.774194"
```

```
specificity_p = TN_p / (FP_p + TN_p)
print(sprintf("Specifcity = %f", specificity_p))
```

```
## [1] "Specifcity = 0.725490"
```

```
F1_p = (2 * recall_p * precision_p) / (recall_p + precision_p)
print(sprintf("F1 measure = %f", F1_p))
```

```
## [1] "F1 measure = 0.695652"
```

Everything except recall has improved after the pruning.