STAT 425 - HW8

Josh Janda

03 May, 2020

```r
library(rpart)
library(randomForest)
library(tidyverse)
library(faraway)
```

Problem 1.

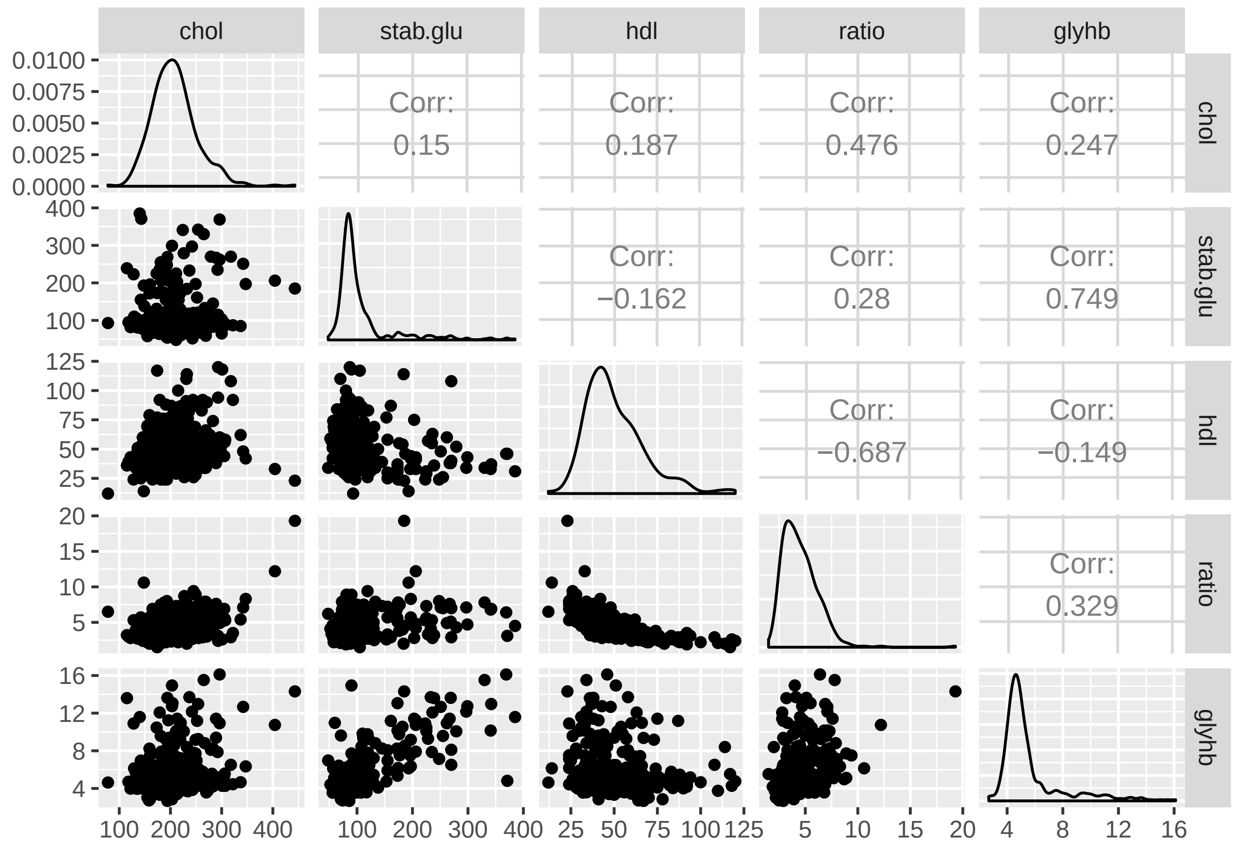
Four hundred and three African Americans were interviewed in a study to understand the prevalence of obesity, diabetes and other cardiovascular risk factors in central Virginia. Data is presented in diabetes. In this question we want to build a regressiontree-based model for predicting glycosolated hemoglobin (glyhb) in term of the other relevant variables.

A.

Plot the response against each of the predictors and comment on the apparent strength of the relationship observed.

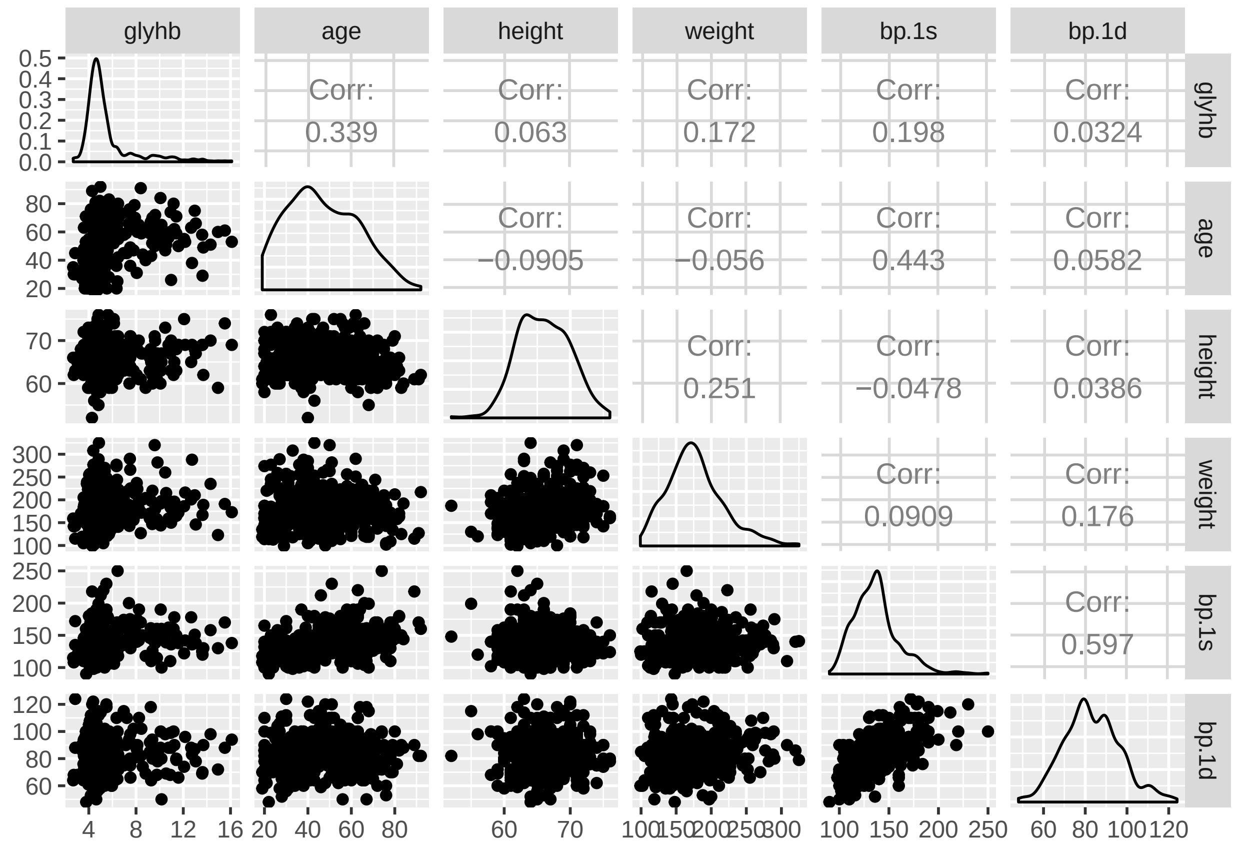
```r
diabetes_rel = diabetes %>% select(-c(id)) # remove any observations with NA values
numeric_diabetes = diabetes_rel %>% select_if(is.numeric)
char_diabetes = diabetes_rel %>% select_if(is.factor) %>% mutate(glyhb = diabetes_rel$glyhb)
```

```r
GGally::ggpairs(numeric_diabetes[, 1:5], progress = FALSE)
```



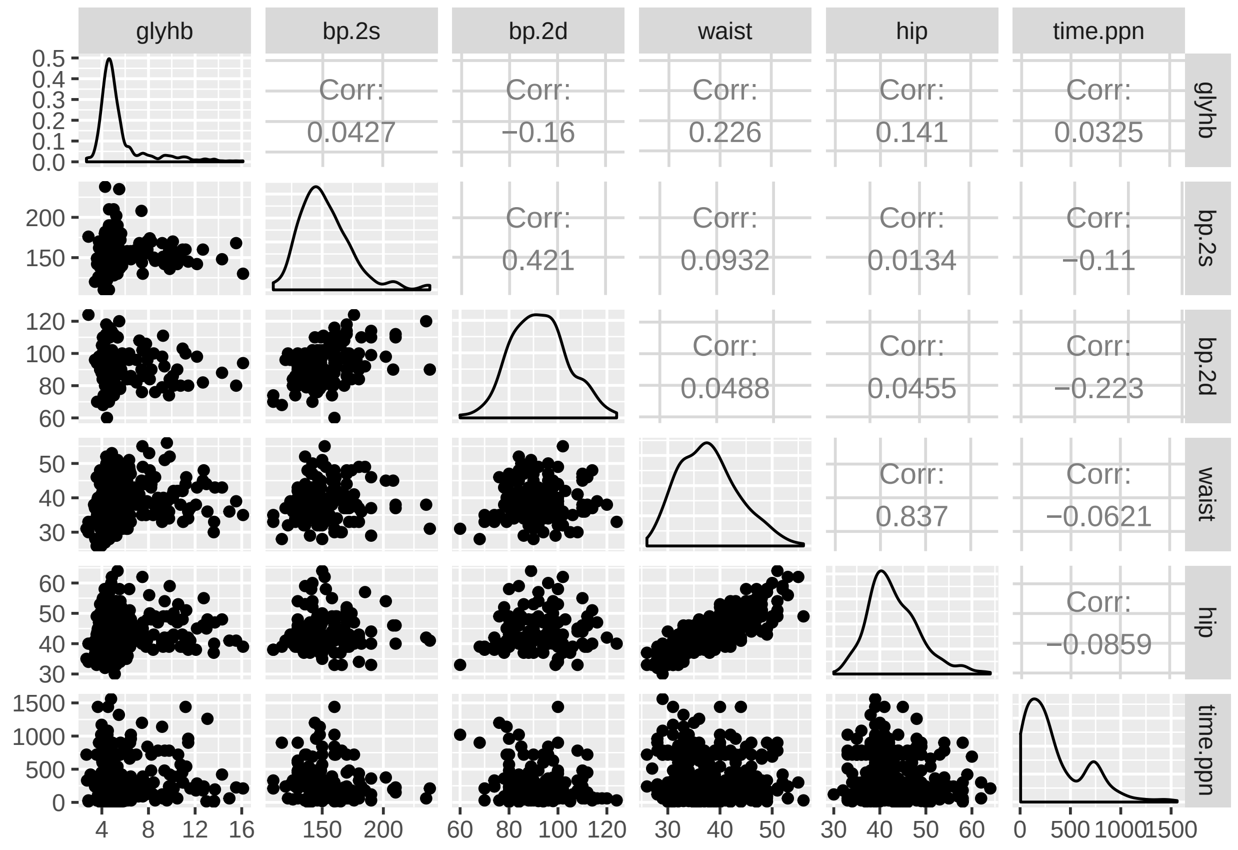
This first plot shows the first four numeric variables (`cholesterol`, `stabilized glucose`, `high density lipoprotein`, `ratio`), plotted against one another as well as the target variable of `glyhb`. Looking at the scatter plots and correlation, each of these predictors show a weak/moderate positive or negative correlation with `glyhb` with no signs of multicollinearity. The variable with the strongest correlation to `glyhb` is `stab.glu`, which represents stabilized glucose in the person.

```
GGally::ggpairs(numeric_diabetes[, c(5, 6:10)], progress = FALSE)
```



This plot shows the next five numeric variables (`age`, `height`, `weight`, `first systolic blood pressure`, and `first diastolic blood pressure`), plotted against one another as well as the target variable of `glyhb`. Looking at the scatter plots and correlation, `age`, `weight`, and `bp.1s` show weak/moderate correlation with the response variable of `glyhb`. Variables `height` and `bp.1d` show extremely weak correlation, possibly suggesting that these variables are not useful as a predictor for `glycisilated hemoglobin`. I will look more into removal of these variables when investigating missing values.

```
GGally::ggpairs(numeric_diabetes[, c(5, 11:15)], progress = FALSE)
```



This last plot shows the next five numeric variables (`second systolic blood pressure`, `second diastolic blood pressure`, `waist`, `hip`, and `postprandial time`), plotted against one another as well as the target variable of `glyhb`. Looking at the scatter plots and correlation I see that `bp.2d`, `waist`, and `hip` demonstrate weak/moderate absolute correlation with the response variable. Variables `bp.2s` and `time.ppn` show extremely weak correlation, possibly suggesting that these variables are not useful as a predictor for `glycisilated hemoglobin`. I will look more into removal of these variables when investigating missing values. It should be noted that for these variables there is a case of multicollinearity between variables `hip` and `waist`, which makes sense as hip and waist sizes are typically determinants of one another.

Next, I will look into categorical variables of `location`, `gender`, and `frame`.

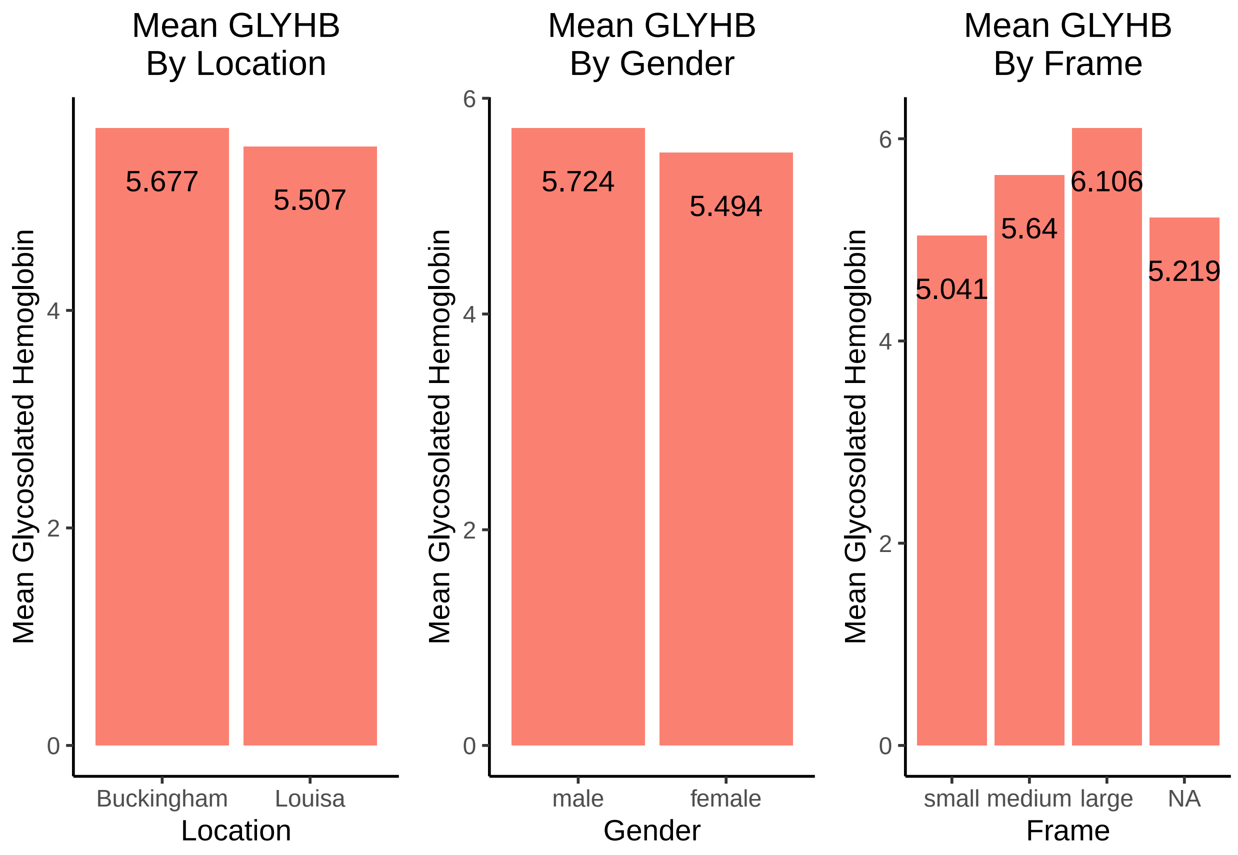
```r
char_diabetes %>% select(location, glyhb) %>%
  group_by(location) %>% summarise(mean_glyhb = mean(glyhb, na.rm = TRUE)) %>%
  ggplot(aes(x = location, y = mean_glyhb)) +
  geom_bar(stat = 'identity', fill = 'salmon') +
  geom_text(aes(label = round(mean_glyhb, 3)), vjust = 3) +
  labs(x = 'Location', y = 'Mean Glycosolated Hemoglobin',
       title = 'Mean GLYHB\nBy Location') +
  theme(panel.background = element_blank(),
        axis.line = element_line(color = 'black'),
        plot.title = element_text(hjust = 0.5)) -> location_plot

char_diabetes %>% select(gender, glyhb) %>%
  group_by(gender) %>% summarise(mean_glyhb = mean(glyhb, na.rm = TRUE)) %>%
  ggplot(aes(x = gender, y = mean_glyhb)) +
  geom_bar(stat = 'identity', fill = 'salmon') +
  geom_text(aes(label = round(mean_glyhb, 3)), vjust = 3) +
  labs(x = 'Gender', y = 'Mean Glycosolated Hemoglobin',
       title = 'Mean GLYHB\nBy Gender') +
  theme(panel.background = element_blank(),
        axis.line = element_line(color = 'black'),
        plot.title = element_text(hjust = 0.5)) -> gender_plot

char_diabetes %>% select(frame, glyhb) %>%
  group_by(frame) %>% summarise(mean_glyhb = mean(glyhb, na.rm = TRUE)) %>%
  ggplot(aes(x = frame, y = mean_glyhb)) +
  geom_bar(stat = 'identity', fill = 'salmon') +
  geom_text(aes(label = round(mean_glyhb, 3)), vjust = 3) +
  labs(x = 'Frame', y = 'Mean Glycosolated Hemoglobin',
       title = 'Mean GLYHB\nBy Frame') +
  theme(panel.background = element_blank(),
        axis.line = element_line(color = 'black'),
        plot.title = element_text(hjust = 0.5)) -> frame_plot
```

```
## Warning: Factor `frame` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

```r
gridExtra::grid.arrange(location_plot, gender_plot,
                        frame_plot, nrow = 1)
```



The plot above looks at mean `Glycosolated Hemoglobin` between groups of each categorical variable.

For `location` I can see that mean `glyhb` does not differ by much between Buckingham and Louisa, with a mean difference of ~.17. This suggests to me that location does not play a large role as a useful predictor.

For 'gender' I can see that mean `glyhb` also does not differ by a great amount between male and female, with a mean difference of ~.25. It should be noted that females have the lower `glyhb` compared to males.

Lastly, for `frame` I can see that there are strong differences between mean `glyhb` and small, medium, and large frames. Note that there is an NA factor, which will be looked into.

```
chisq.test(char_diabetes$location, char_diabetes$glyhb)
```

```
## Warning in chisq.test(char_diabetes$location, char_diabetes$glyhb): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  char_diabetes$location and char_diabetes$glyhb
## X-squared = 237.3, df = 238, p-value = 0.5006
```

For the test above, we get a p-value of ~.50 so I fail to reject H_0 that there is no relationship between `glyhb` and `location`. Location will be removed as a predictor.

```
chisq.test(char_diabetes$gender, char_diabetes$glyhb)
```

```
## Warning in chisq.test(char_diabetes$gender, char_diabetes$glyhb): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  char_diabetes$gender and char_diabetes$glyhb
## X-squared = 235.85, df = 238, p-value = 0.5272
```

For the test above, we get a p-value of ~.53 so I fail to reject H_0 that there is no relationship between `glyhb` and `gender`. Gender will be removed as a predictor.

```
chisq.test(char_diabetes$frame, char_diabetes$glyhb)
```

```
## Warning in chisq.test(char_diabetes$frame, char_diabetes$glyhb): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  char_diabetes$frame and char_diabetes$glyhb
## X-squared = 477.52, df = 468, p-value = 0.3704
```

For the test above, we get a p-value of ~.37 so I fail to reject H_0 that there is no relationship between `glyhb` and `frame`. Frame will be removed as a predictor.

Overall, from testing I will be removing all three categorical variables.

B.

Investigate the pattern of missing values in the data. By eliminating a combination of rows and columns, produce a reduced dataset that contains no missing values.

For starters, I will remove all three categorical variables.

```
new_diabetes = diabetes_rel %>% select(-c(location, gender, frame))

summary(new_diabetes)
```

```
##       chol          stab.glu          hdl             ratio       
##  Min.   : 78.0   Min.   : 48.0   Min.   : 12.00   Min.   : 1.500  
##  1st Qu.:179.0   1st Qu.: 81.0   1st Qu.: 38.00   1st Qu.: 3.200  
##  Median :204.0   Median : 89.0   Median : 46.00   Median : 4.200  
##  Mean   :207.8   Mean   :106.7   Mean   : 50.45   Mean   : 4.522  
##  3rd Qu.:230.0   3rd Qu.:106.0   3rd Qu.: 59.00   3rd Qu.: 5.400  
##  Max.   :443.0   Max.   :385.0   Max.   :120.00   Max.   :19.300  
##  NA's   :1                       NA's   :1        NA's   :1       
##      glyhb            age            height          weight     
##  Min.   : 2.68   Min.   :19.00   Min.   :52.00   Min.   : 99.0  
##  1st Qu.: 4.38   1st Qu.:34.00   1st Qu.:63.00   1st Qu.:151.0  
##  Median : 4.84   Median :45.00   Median :66.00   Median :172.5  
##  Mean   : 5.59   Mean   :46.85   Mean   :66.02   Mean   :177.6  
##  3rd Qu.: 5.60   3rd Qu.:60.00   3rd Qu.:69.00   3rd Qu.:200.0  
##  Max.   :16.11   Max.   :92.00   Max.   :76.00   Max.   :325.0  
##  NA's   :13                      NA's   :5       NA's   :1      
##      bp.1s           bp.1d            bp.2s           bp.2d       
##  Min.   : 90.0   Min.   : 48.00   Min.   :110.0   Min.   : 60.00  
##  1st Qu.:121.2   1st Qu.: 75.00   1st Qu.:138.0   1st Qu.: 84.00  
##  Median :136.0   Median : 82.00   Median :149.0   Median : 92.00  
##  Mean   :136.9   Mean   : 83.32   Mean   :152.4   Mean   : 92.52  
##  3rd Qu.:146.8   3rd Qu.: 90.00   3rd Qu.:161.0   3rd Qu.:100.00  
##  Max.   :250.0   Max.   :124.00   Max.   :238.0   Max.   :124.00  
##  NA's   :5       NA's   :5        NA's   :262     NA's   :262     
##      waist           hip           time.ppn     
##  Min.   :26.0   Min.   :30.00   Min.   :   5.0  
##  1st Qu.:33.0   1st Qu.:39.00   1st Qu.:  90.0  
##  Median :37.0   Median :42.00   Median : 240.0  
##  Mean   :37.9   Mean   :43.04   Mean   : 341.2  
##  3rd Qu.:41.0   3rd Qu.:46.00   3rd Qu.: 517.5  
##  Max.   :56.0   Max.   :64.00   Max.   :1560.0  
##  NA's   :2      NA's   :2       NA's   :3       
```

Looking at the summary table above, all but two variables (`age` and `stab.glu`) have missing values. My method for removing missing values will be to drop the columns with a mass amount of NA values which are `bp.2s` and `bp.2d`. I will then remove all remaining rows that contain an NA value.

I will also remove any variables that had extremely weak correlation with the response variable, which are `time.ppn` and `bp.1d`

```
final_diabetes = new_diabetes %>% select(-c(bp.2s, bp.2d, time.ppn, bp.1d)) %>%
  drop_na()
summary(final_diabetes)
```

```
##       chol          stab.glu          hdl             ratio
##  Min.   : 78.0   Min.   : 48.0   Min.   : 12.00   Min.   : 1.500
##  1st Qu.:179.0   1st Qu.: 81.0   1st Qu.: 38.00   1st Qu.: 3.200
##  Median :204.0   Median : 90.0   Median : 46.00   Median : 4.200
##  Mean   :207.7   Mean   :107.5   Mean   : 50.41   Mean   : 4.531
##  3rd Qu.:230.0   3rd Qu.:108.0   3rd Qu.: 59.00   3rd Qu.: 5.400
##  Max.   :443.0   Max.   :385.0   Max.   :120.00   Max.   :19.300
##      glyhb             age            height          weight
##  Min.   : 2.680   Min.   :19.00   Min.   :52.00   Min.   : 99
##  1st Qu.: 4.390   1st Qu.:34.00   1st Qu.:63.00   1st Qu.:151
##  Median : 4.860   Median :45.00   Median :66.00   Median :174
##  Mean   : 5.595   Mean   :46.92   Mean   :66.01   Mean   :178
##  3rd Qu.: 5.630   3rd Qu.:60.00   3rd Qu.:69.00   3rd Qu.:200
##  Max.   :16.110   Max.   :92.00   Max.   :76.00   Max.   :325
##      bp.1s           waist            hip
##  Min.   : 90.0   Min.   :26.00   Min.   :30.00
##  1st Qu.:122.0   1st Qu.:33.00   1st Qu.:39.00
##  Median :136.0   Median :37.00   Median :42.00
##  Mean   :137.4   Mean   :37.96   Mean   :43.08
##  3rd Qu.:148.0   3rd Qu.:41.00   3rd Qu.:46.00
##  Max.   :250.0   Max.   :56.00   Max.   :64.00
```

The dataset now contains zero missing values and only predictors with moderate to strong correlation with the response.

C.

Fit the default tree. From the output answer the following questions: How many observations had stab.glu < 158? What was the mean response for these observations? What characterizes the largest terminal node in the tree? What is the mean response of this node?

```r
set.seed(27)
first_tree = rpart(glyhb ~ ., data = final_diabetes, cp = 0.01)
first_tree
```

```
## n= 377
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 377 1842.38700  5.595199
##    2) stab.glu< 158 333  550.79970  5.015105
##      4) stab.glu< 105.5 276  308.60200  4.794420
##        8) age< 55.5 211  116.22810  4.591754 *
##        9) age>=55.5 65  155.57430  5.452308 *
##      5) stab.glu>=105.5 57  163.66990  6.083684
##       10) bp.1s< 141.5 31   52.84848  5.479032 *
##       11) bp.1s>=141.5 26   85.97445  6.804615
##         22) chol< 206.5 11   31.49120  5.810000 *
##         23) chol>=206.5 15   35.62136  7.534000 *
##    3) stab.glu>=158 44  331.45870  9.985455
##      6) chol< 193.5 15   71.90724  8.312000 *
##      7) chol>=193.5 29  195.81710 10.851030
##        14) stab.glu< 274.5 22  126.87840 10.225450 *
##        15) stab.glu>=274.5 7   33.26995 12.817140 *
```

There are 333 observations with a `stab.glu` value less than 158.

From the output above, the largest terminal node in the tree is when `stag.glu` is greater than or equal to 158 and `chol` is less than 193.5. If the person satisfies these two filters, then the predicted mean `glyhb` value is 8.312.

```r
final_diabetes %>% filter(stab.glu < 158) %>% summarise(mean(glyhb))
```

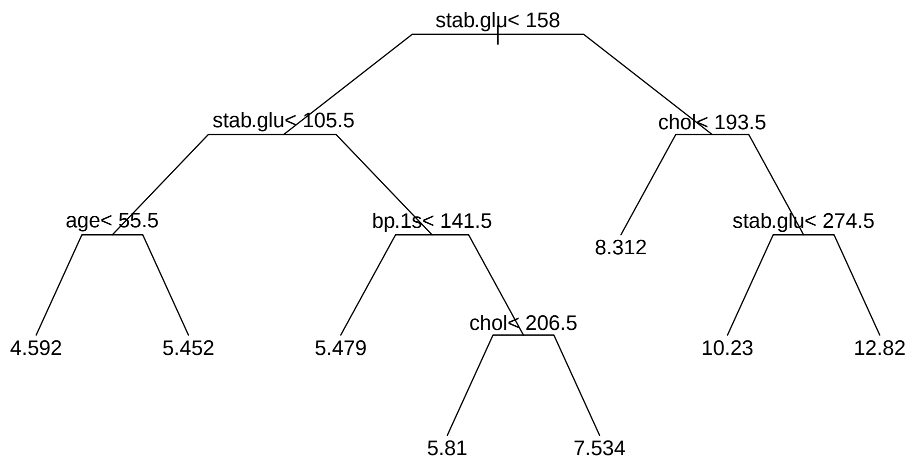
```
##   mean(glyhb)
## 1    5.015105
```

Those with a `stablized glucose` value of less than 158 have a mean `glycosolated hemoglobin` value of ~5.02.

D.

Make a plot of the tree. What feature of the plot reveals the most important predictor?

```r
plot(first_tree,compress=T,uniform=T,branch=0.4,margin=.10)
text(first_tree)
```



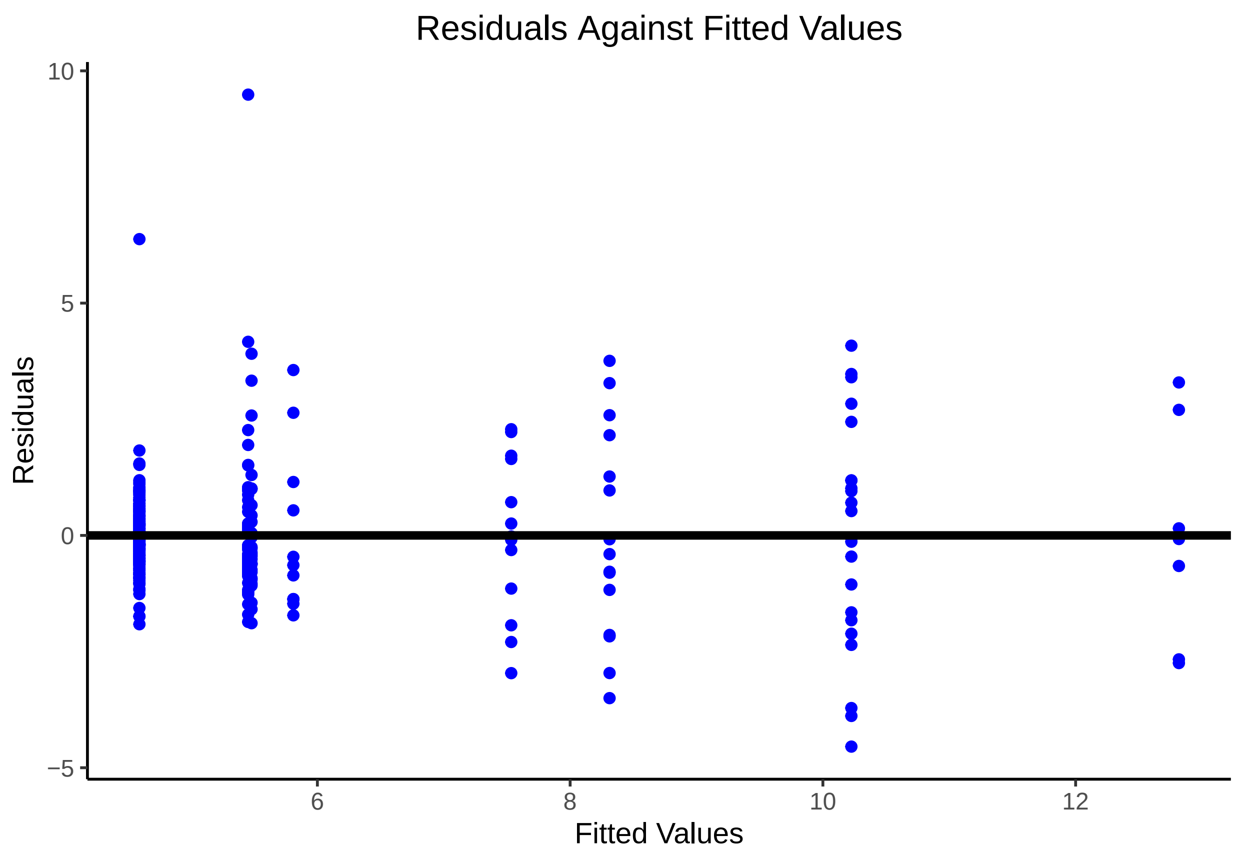
The plot above allows me to visualize the tree fully and understand where splits are occuring. The plot reveals that the most important predictor for determining mean `glyhb` is the variable `stab.glu`, as it is the variable split on for the root node and also the variable split on for two terminal nodes on the right side. Having the most amount of splits tells me this is one of the most important variables.

E.

Plot the residuals against the fitted values for this tree. Comment.

```
resid_df = data.frame(fitted = predict(first_tree, final_diabetes),
                      residuals = residuals(first_tree))

ggplot(data = resid_df, aes(x = fitted, y = residuals)) +
  geom_point(color = 'blue') +
  geom_hline(yintercept = 0, color = 'black', lwd = 1.5) +
  labs(x = 'Fitted Values', y = 'Residuals', title = 'Residuals Against Fitted Values') +
  theme(panel.background = element_blank(),
        axis.line = element_line(color = 'black'),
        plot.title = element_text(hjust = 0.5))
```

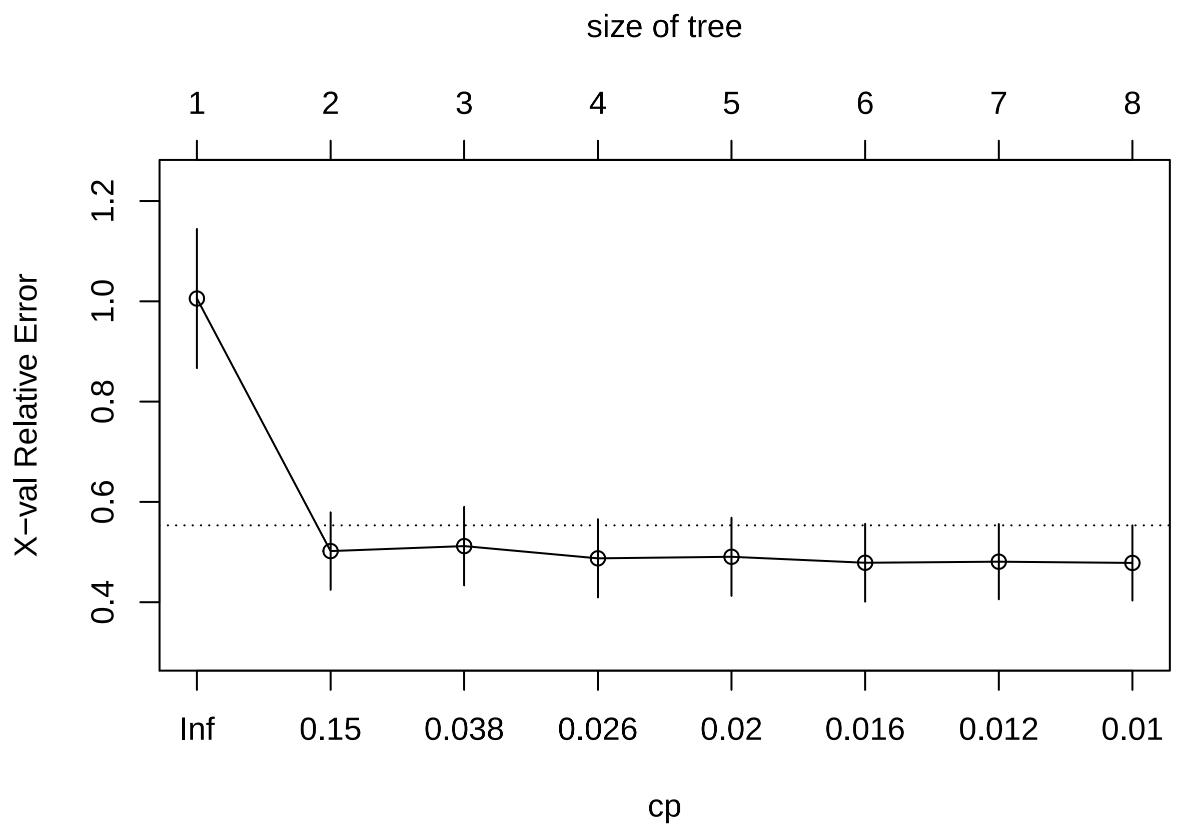


Looking at the plot above, there does not seem to be much, if any, heteroskedasticity. This plot looks good to me.

F.

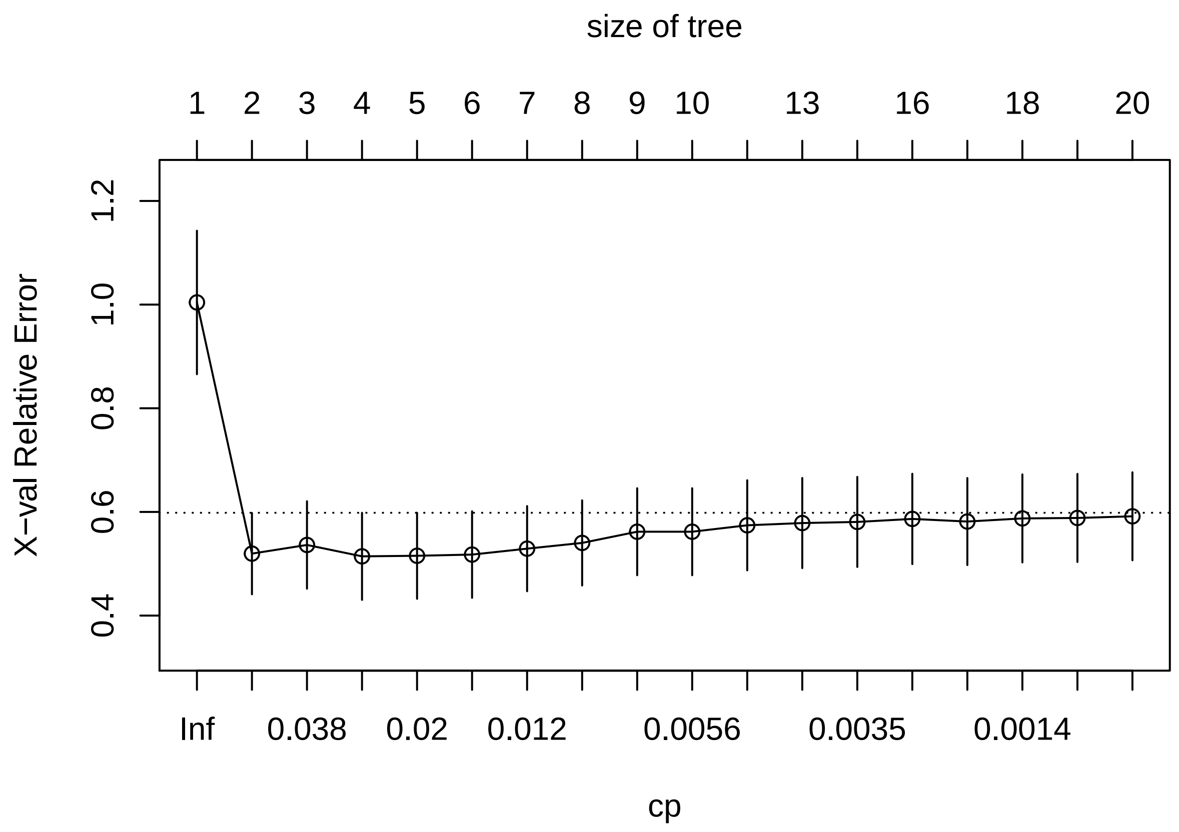
Select the optimal tree using Cross-validation. What would be the smallest tree that could be reasonably used?

```
plotcp(first_tree)
```



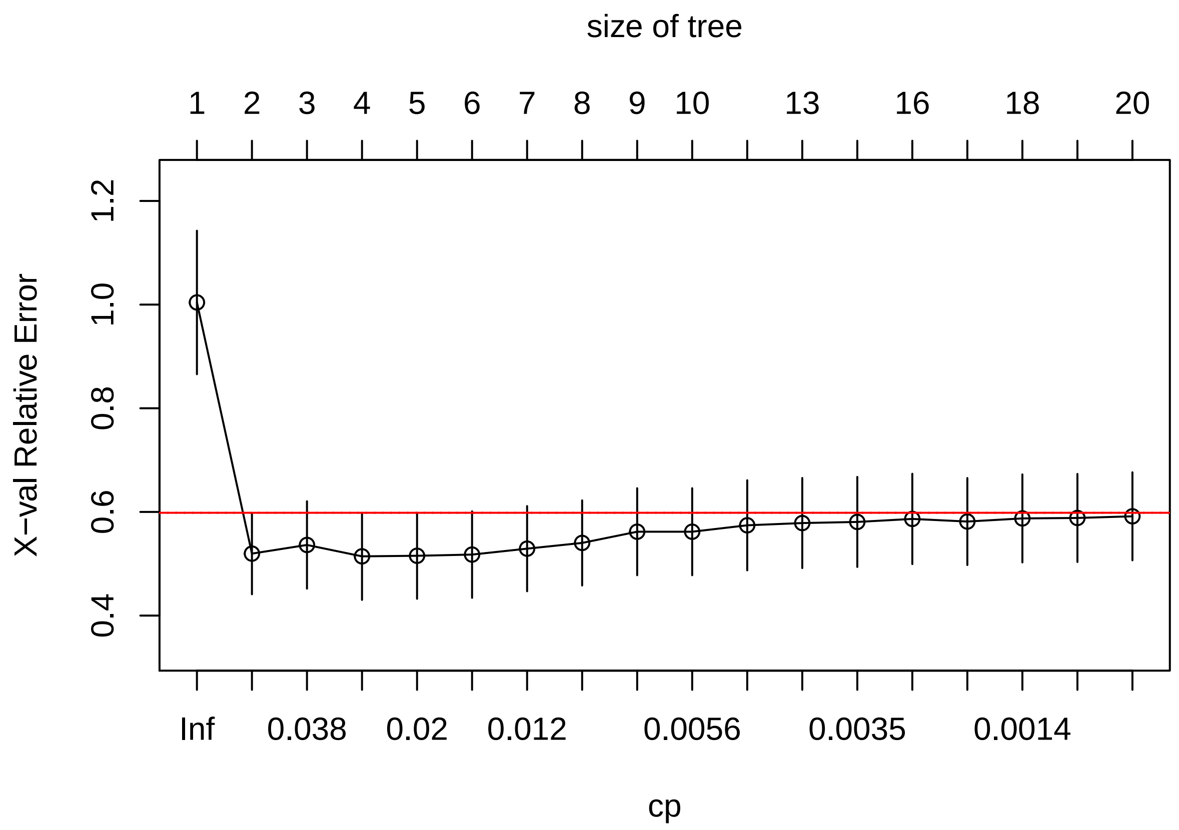
Looking at the plot above, using a cp value os .01 creates trees that are too small as the relative error is still decreasing at a cp value of 0.01.

```r
second_tree = rpart(glyhb ~ ., data = final_diabetes, cp = 0.001)
plotcp(second_tree)
```



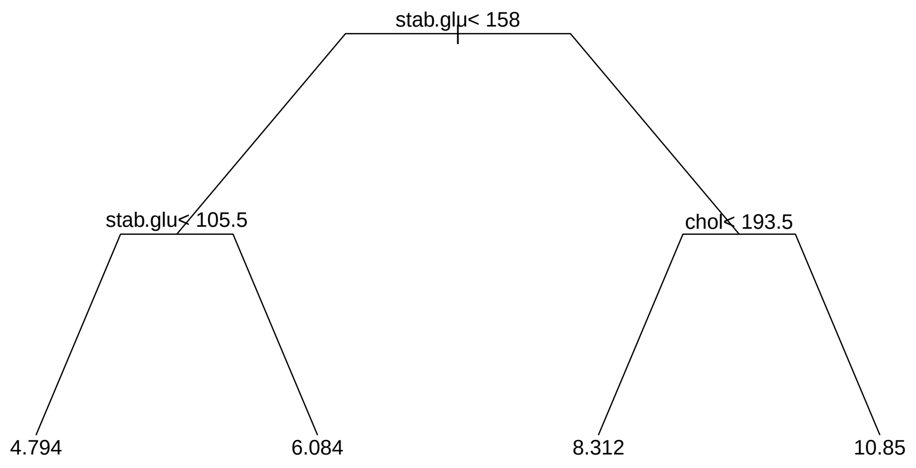
Using the new plot above, we see that as cp increases past .01 the relative error actually begins to increase which I do not want. I will be selecting the tree with the smallest CV error plus one standard error.

```
myCPtable = second_tree$cptable
id.min = which.min(myCPtable[,'xerror'])
my1se.err = myCPtable[id.min,'xerror'] + myCPtable[id.min, 'xstd']
plotcp(second_tree)
abline(h=my1se.err, col="red")
```



Using the output above, the smallest theoretical tree size that can be used is a tree of size two as it is under the minimum error plus one standard error. I will be selecting the tree using the smallest CV error approach, which chooses the tree with minimum CV error.

```
CP.min = (myCPtable[id.min, 'CP'] + myCPtable[(id.min-1), 'CP'])/2
tree.min = prune.rpart(second_tree, CP.min)
plot(tree.min,compress=T,uniform=T,branch=0.4,margin=.10)
text(tree.min)
```



The tree with the minimum CV error was a tree of size four and CP value of 0.0272836.

Overall, the optimal tree for this data is a tree of size four but in reality a tree of size two could be used and still be a good fit.

Problem 2.

The data set wbca comes from a study of breast cancer in Wisconsin. There are 681 cases of potentially cancerous tumors of which 238 are actually malignant.

A.

Fit a binary regression with Class as the response and the other nine variables as predictors. Report the residual deviance and associated degrees of freedom. Can this information be used to determine if the models fits the data? Explain.

```r
bin_reg = glm(Class ~ ., data = wbca, family = binomial)
summary(bin_reg)
```

```
##
## Call:
## glm(formula = Class ~ ., family = binomial, data = wbca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.48282  -0.01179   0.04739   0.09678   3.06425
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.16678    1.41491   7.892 2.97e-15 ***
## Adhes       -0.39681    0.13384  -2.965  0.00303 **
## BNucl       -0.41478    0.10230  -4.055 5.02e-05 ***
## Chrom       -0.56456    0.18728  -3.014  0.00257 **
## Epith       -0.06440    0.16595  -0.388  0.69795
## Mitos       -0.65713    0.36764  -1.787  0.07387 .
## NNucl       -0.28659    0.12620  -2.271  0.02315 *
## Thick       -0.62675    0.15890  -3.944 8.01e-05 ***
## UShap       -0.28011    0.25235  -1.110  0.26699
## USize        0.05718    0.23271   0.246  0.80589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.388  on 680  degrees of freedom
## Residual deviance:  89.464  on 671  degrees of freedom
## AIC: 109.46
##
## Number of Fisher Scoring iterations: 8
```

The residual deviance for this model is 89.464 on 671 degrees of freedom. Using this as model criterion can determine if the model fits the data as the residual deviance shows how well the response is prediced when including all variables in the model where the null deviance shows how well the response is predicted when only including an intercept.

B.

Use AIC as the criterion to determine the best subset of variables. (Use the step function).

```r
best_mod = step(bin_reg)
```

```
## Start:  AIC=109.46
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +
##     UShap + USize
##
##         Df Deviance    AIC
## - USize  1   89.523 107.52
## - Epith  1   89.613 107.61
## - UShap  1   90.627 108.63
## <none>       89.464 109.46
## - Mitos  1   93.551 111.55
## - NNucl  1   95.204 113.20
## - Adhes  1   98.844 116.84
## - Chrom  1   99.841 117.84
## - BNucl  1  109.000 127.00
## - Thick  1  110.239 128.24
##
## Step:  AIC=107.52
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +
##     UShap
##
```

```
##         Df Deviance    AIC
## - Epith  1   89.662 105.66
## - UShap  1   91.355 107.36
## <none>       89.523 107.52
## - Mitos  1   93.552 109.55
## - NNucl  1   95.231 111.23
## - Adhes  1   99.042 115.04
## - Chrom  1  100.153 116.15
## - BNucl  1  109.064 125.06
## - Thick  1  110.465 126.47
##
## Step:  AIC=105.66
## Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap
##
##         Df Deviance    AIC
## <none>       89.662 105.66
## - UShap  1   91.884 105.88
## - Mitos  1   93.714 107.71
## - NNucl  1   95.853 109.85
## - Adhes  1  100.126 114.13
## - Chrom  1  100.844 114.84
## - BNucl  1  109.762 123.76
## - Thick  1  110.632 124.63
```

According to AIC criterion, the best subset of variables to include in this model are:

- `UShap` : cell shape uniformity
- `Mitos` : mitoses
- `NNucl` : normal nucleoli
- `Adhes` : marginal adhesion
- `Chrom` : bland chromatin
- `BNucl` : bare nuclei
- `Thick` : clump thickness

C.

Suppose that a cancer is classified as benign if $p > 0.5$ and malignant if $p < 0.5$. Compute the number of errors of both types (false positives and false negatives) that will be made if this method is applied to the current data with the reduced model.

```r
predictions = best_mod$fitted.values
prediction_classes = factor(ifelse(predictions > .50, 1, 0))

cf_mtx = caret::confusionMatrix(prediction_classes, factor(wbca$Class))
cf_mtx$table
```

```
##           Reference
## Prediction   0   1
##          0 227   9
##          1  11 434
```

False positive, or the type 1 error, is the total number of observations predicted as `malignant` but are actually `benign`. There are a toital of 9 false positives predicted by the reduced model, which gives a false positive rate of 0.020316.

False negative, or the type 2 error, is the total number number of observations predicted as `benign` but are actually `malignant`. There are a total of 11 false negatives predicted by the reduced model, which gives a false negative rate of 0.0462185.