Analysis of Biopsy Data from Breast Cancer Patients

Data Descriptions

The data was collected from the biopsied cells of seven hundred breast cancer tumors. The goal of these biopsies was to determine if the tumors were benign or malignant. This determination was based on nine characteristics of the cells which were ranked from 1(benign) to 10(malignant):

1) Clump Thickness – This refers to how the cells aggregate. If they are monolayered they are benign and if they form on top of each other, creating a thick clump, they are malignant

2) Uniform Size – All cells of the same type should be the same size. If the cells vary in size, they could be malignant

3) Uniform Shape – All cells of the same type should be the same shape. If they vary in cell shape they could be malignant

4) Marginal Adhesion – This refers to how well the cells stick together. Healthy cells have a strong ability to stick together whereas cancerous cells do not

5) Single Epithelial Size – Epithelial cells should all be equal in size. If they are not it could be a sign of cancer
6) Bare nuclei – The nucleus of the cell should be surrounded by the rest of the components of the cell, known as the cytoplasm. If the nucleus is not surrounded by cytoplasm the cell could be malignant
7) Bland Chromatin – The chromatin should have a uniform texture. If the texture is coarse the cell could be malignant

8) Normal Nucleoli – In a healthy cell the nucleoli is small and hard detect via imagery. Enlarged nucleoli could be a sign of cancer

9) Mitosis – cells that multiply at an uncontrollable rate could be malignant

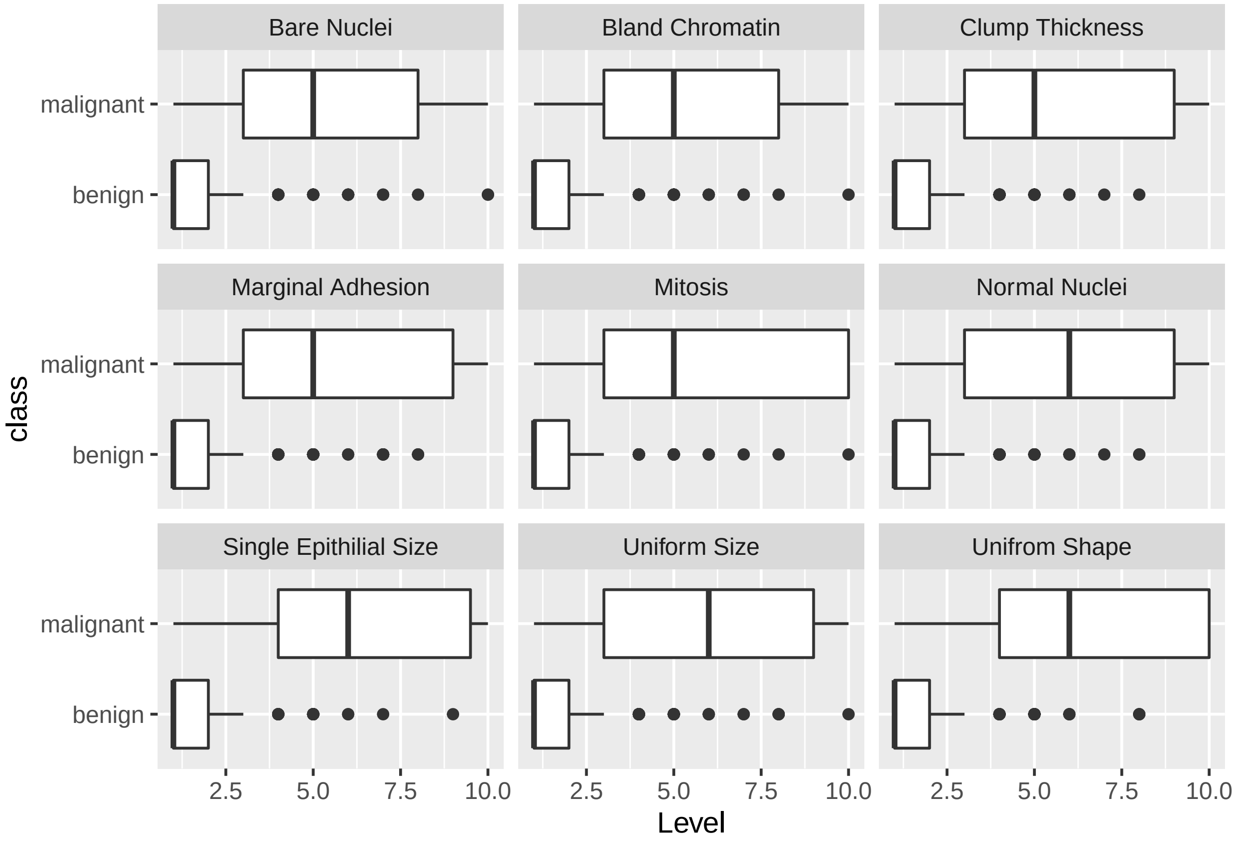
The column labelled ID is the patient ID to ensure anonymity. Finally the column labelled Class represents the diagnosis, either benign or malignant, of the patient.

Our goal is to use the data to determine a model that best represent the factors that effect the malignancy of a breast cancer tumor. This model will be used to predict whether a patient is benign or malignant with the least amount of false positives and false negatives possible.

Preliminary Analysis

The data set has 16 missing values under column “Bare Nuclei”. We need to determine if we should impute the missing values or simply remove the data rows all together. A popular imputation method when using categorical data is the K Nearest Neighbor process which averages the k nearest numbers in the neighboring columns to estimate the missing number. The KNN process is very sensitive to outliers because it chooses numbers based solely on its distance from the missing number. Thus, it increased our bias without improving the results so it was not the method we chose to use. Since only 2% of our data was missing completely at random, we decide to remove the data rows that contained the missing values. Next we clean up the data by adding appropriate column names and removing the column "ID" because it does not provide any value to our model.

We want to gain a better understanding of the data we are working with. We use "ggplot" to create a boxplot for each variable and the class of the patient. We can see that all variables behave similarly, with values between 1 and 2 classified as benign and values greater than 2 classified as malignant. The spread for malignancy is much larger than benign patients so we will keep it in mind when testing our data. The boxplots for each variable look similar so we cannot rule any as significant.



Once we build our model, we should check for collinearity. To check for this we use the Variance Inflation Factor which measures the amount of collinearity in multivariable regression.

Model Building

Now we will create our logistic model with the 'glm' function by setting the family to binomial. Let's look at the summary of the model to see the effect of each variable.

```
model = glm(Class~., family = binomial, data=df.train)
summary(model)
```

```
Call:
```

```
glm(formula = Class ~ ., family = binomial, data = df.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5410  -0.1059  -0.0496   0.0147   2.1478

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -10.77357    1.58581  -6.794 1.09e-11 ***
Clump_Thickness     0.64984    0.20604   3.154  0.00161 **
Uniform_Size       -0.17261    0.28042  -0.616  0.53819
Uniform_Shape       0.25444    0.28934   0.879  0.37919
Marginal_Adhesion   0.36114    0.15120   2.388  0.01692 *
Single_Epith_Size  -0.08931    0.23138  -0.386  0.69949
Bare_Nuclei         0.55771    0.13975   3.991 6.59e-05 ***
Bland_Chromatin     0.49400    0.23570   2.096  0.03609 *
Normal_Nuclei       0.46988    0.17385   2.703  0.00688 **
Mitosis             0.65013    0.42580   1.527  0.12681
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 624.576  on 477  degrees of freedom
Residual deviance:  65.596  on 468  degrees of freedom
AIC: 85.596

Number of Fisher Scoring iterations: 8
```

The summary indicates that Clump_Thickness and Bare_Nuclei are the leading predictors of the malignancy of the cells while others like Uniform_Size and Uniform_Shape are not significant.

Let’s check for collinearity using the VIF function from the ‘car’ package. A variance inflation factor of greater than 5 indicates that there could be severe correlation between predictor variables.

```r
vif(model)
```

```
  Clump_Thickness      Uniform_Size     Uniform_Shape Marginal_Adhesion
         1.541802          3.607154          3.194895          1.295117
Single_Epith_Size       Bare_Nuclei   Bland_Chromatin     Normal_Nuclei
         1.599297          1.292432          1.378905          1.550112
          Mitosis
         1.058635
```

All of our variables have a variance inflation factor of less than 5 which means that there is no severe correlation between predictor variables.

We can see if our model can be improved by removing variables that do not have a significant effect on whether cells will be malignant or benign by looking at how the residual deviance and AIC change as variables are removed. We want residual deviance to be low as it shows how well the response is predicted by the model when the variable predictors are included. The AIC should be low as well as it shows how well the model fits the data without overfitting it. The AIC score rewards models with a high goodness-of-fit score and penalizes models that are overly complex.

Let’s remove the first variable with a high p-value, Uniform_Size, and see how it affects the model.

```r
model2 = glm(Class~.-Uniform_Size, family = binomial, data=df.train)
summary(model2)
```

```
Call:
glm(formula = Class ~ . - Uniform_Size, family = binomial, data = df.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4777  -0.1123  -0.0518   0.0145   2.1507

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -10.59171    1.53495  -6.900 5.19e-12 ***
Clump_Thickness     0.64753    0.20985   3.086  0.00203 **
Uniform_Shape       0.12918    0.21853   0.591  0.55443
Marginal_Adhesion   0.33772    0.14521   2.326  0.02003 *
Single_Epith_Size  -0.08358    0.23454  -0.356  0.72158
Bare_Nuclei         0.54464    0.13666   3.985 6.74e-05 ***
Bland_Chromatin     0.47562    0.23535   2.021  0.04329 *
Normal_Nuclei       0.43666    0.16224   2.691  0.00711 **
Mitosis             0.62316    0.41951   1.485  0.13742
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 624.576  on 477  degrees of freedom
Residual deviance:  65.954  on 469  degrees of freedom
AIC: 83.954

Number of Fisher Scoring iterations: 8
```

Removing Uniform_Size from the model increased the residual deviance by a very small amount while decreasing the AIC, indicating that we should exclude this variable.

Now let’s remove Uniform_Shape.

```r
model3 = glm(Class~.-Uniform_Shape, family = binomial, data=df.train)
summary(model3)
```

```
Call:
glm(formula = Class ~ . - Uniform_Shape, family = binomial, data = df.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5867  -0.1082  -0.0476   0.0138   2.0916

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)       -10.873614   1.584355  -6.863 6.74e-12 ***
Clump_Thickness     0.712766   0.201537   3.537 0.000405 ***
```

```
Uniform_Size         0.007915   0.198997   0.040 0.968273
Marginal_Adhesion    0.359995   0.150789   2.387 0.016967 *
Single_Epith_Size   -0.076239   0.232186  -0.328 0.742646
Bare_Nuclei          0.568378   0.139167   4.084 4.42e-05 ***
Bland_Chromatin      0.495760   0.233763   2.121 0.033940 *
Normal_Nuclei        0.467185   0.170057   2.747 0.006010 **
Mitosis              0.641832   0.420812   1.525 0.127204
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 624.576  on 477  degrees of freedom
Residual deviance:  66.319  on 469  degrees of freedom
AIC: 84.319

Number of Fisher Scoring iterations: 8
```

The residual deviance and AIC both increased so we should not remove Uniform_Shape from the model.

After going through each of the variables with high p-values, this is the model we decided to go with.

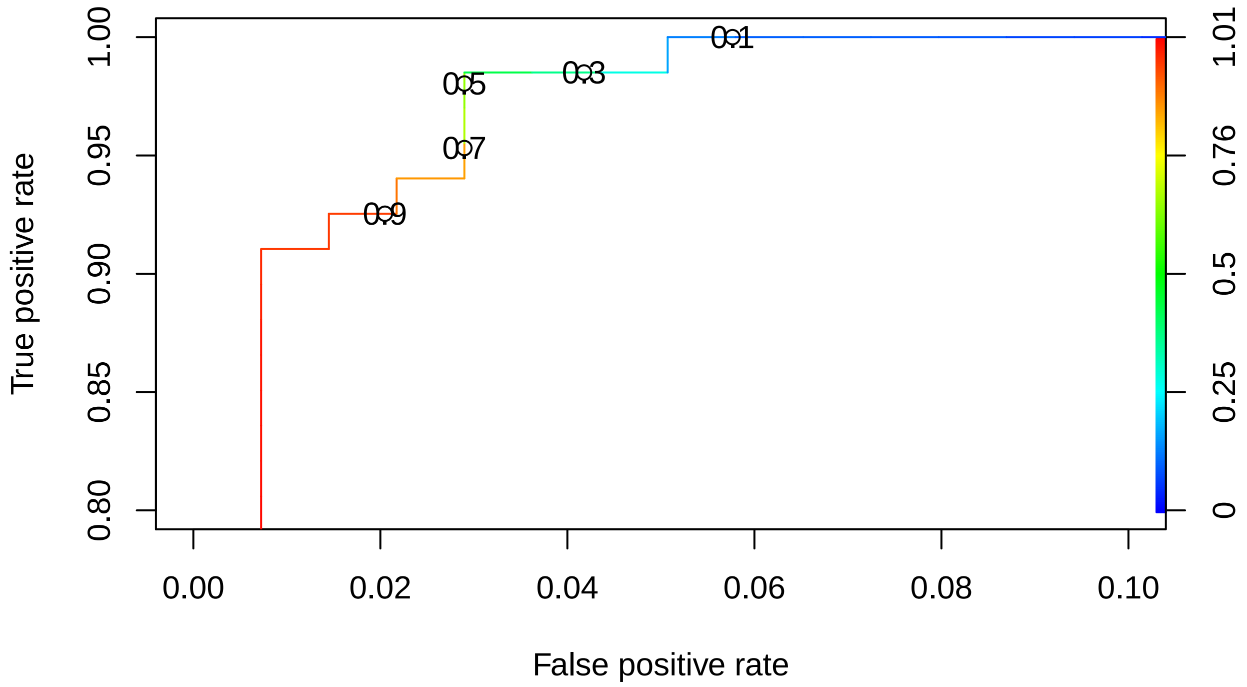
```
model8 = glm(Class~.-Uniform_Size-Single_Epith_Size, family = binomial, data=df.train)
summary(model8)
```

```
Call:
glm(formula = Class ~ . - Uniform_Size - Single_Epith_Size, family = binomial,
    data = df.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4440  -0.1118  -0.0524   0.0160   2.1949

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -10.5917     1.5247  -6.947 3.73e-12 ***
Clump_Thickness     0.6334     0.2036   3.111  0.00187 **
Uniform_Shape       0.1245     0.2158   0.577  0.56411
Marginal_Adhesion   0.3292     0.1427   2.307  0.02105 *
Bare_Nuclei         0.5399     0.1368   3.946 7.95e-05 ***
Bland_Chromatin     0.4443     0.2179   2.039  0.04144 *
Normal_Nuclei       0.4191     0.1536   2.729  0.00635 **
Mitosis             0.6241     0.4131   1.511  0.13085
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 624.576  on 477  degrees of freedom
Residual deviance:  66.082  on 470  degrees of freedom
AIC: 82.082

Number of Fisher Scoring iterations: 8
```

Removing Uniform_Size and Single_Epith_Size gives us the best combination of low AIC and residual deviance. The residual deviance is higher than the original model but it is only by a very slight amount and the lower AIC is worth the trade off.

Now that we have our model, let's see how it performs. The ROCR package will be utilized with the 'predict' function to see how many times our model correctly predicts malignancy. Then we'll plot an ROC curve to get a visualization of the false positive and true positive rates.

```r
res = predict(model8, df.test, type = "response")
ROCRPRed = prediction(res, df.test$Class)
ROCRPerf = performance(ROCRPRed, "tpr", "fpr")
plot(ROCRPerf, colorize=TRUE, print.cutoffs.at=seq(0.1, by=0.2), xlim=c(0,0.1), ylim=c(0.8,1))
```



This plot is very useful because it shows how many true positives and false positives the model gives at a certain cutoff. For example, a cutoff of 0.5 means that all cells with a probability of 0.5 or higher of being malignant will be classified as being malignant while those with probabilities lower than 0.5 will be classified as benign.

We create a confusion matrix with different acceptance rates so we can see how it affects the true and false positivity rates.

```r
table(ActualValue=df.test$Class, PredictedValue=res>0.4)
```

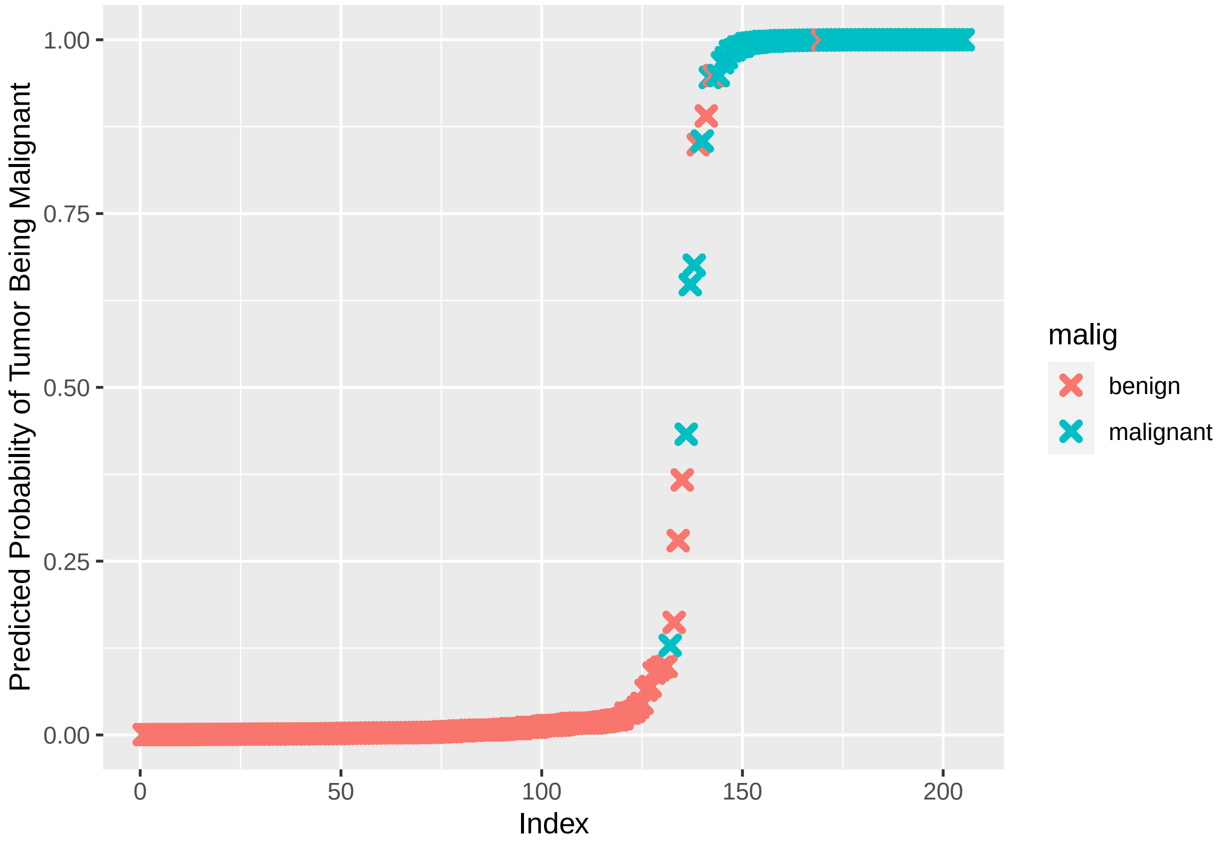
```
           PredictedValue
ActualValue FALSE TRUE
  benign      134    4
  malignant     1   66
```

```
table(ActualValue=df.test$Class, PredictedValue=res>0.7)
```

```
           PredictedValue
ActualValue FALSE TRUE
  benign      134    4
  malignant     4   63
```

As indicated by the ROC curve, the higher cutoff values decrease the true positivity rate and false positivity rate. Since our data involves diagnosing malignant tumors, it is important to keep the false negative rate low as this would be telling someone who has a malignant tumor that it is benign. We found that a cutoff of 0.4 gives a good balance of low false negatives while still maintaining a high true positive rate.

The graph below illustrates the models predictions with the actual results represented by the different colours.



This shows why lowering the cutoff improves the accuracy of the model as some malignant tumors are being underestimated which would cause false negatives.

Conclusion

Uniform Size and Single Epithithial Size were not significant in predicting the malignancy of tumor cells so our model does not include these variables. Our fitted model reduces the null deviance and AIC without impacting the residual deviance by a significant amount and is able to predict the testing dataset with >90% accuracy.

For further analysis, we could run the model multiple times because our original and revised model are similar. New training and testing data would help confirm our results and help identify possible overfitting.

Code

```
library(ggplot2)
library(cowplot)
library(reshape2)
library(ROCR)
library(VIM)
library(car)

setwd("E:\\Google Drive\\School\\Fourth Year\\Stats 4864\\Final Project")
df = read.csv("biopsy.csv", header = TRUE)
df[,-11] = lapply(df[,-11], as.integer)
#df = kNN(df, variable = "V6", k = 5)
df = na.omit(df)
#df$V6_imp = NULL

colnames(df) = c("ID", "Clump_Thickness", "Uniform_Size", "Uniform_Shape", "Marginal_Adhesion",
                 "Single_Epith_Size", "Bare_Nuclei", "Bland_Chromatin", "Normal_Nuclei",
                 "Mitosis", "Class")
df$ID = NULL

for (i in 1:9) {
  df[,i] = as.integer(df[,i])
}
df$Class = factor(df$Class)

ggdf = data.frame("Level" = c(df$Clump_Thickness, df$Uniform_Size, df$Uniform_Shape,
                              df$Marginal_Adhesion, df$Single_Epith_Size, df$Bare_Nuclei,
                              df$Bland_Chromatin, df$Normal_Nuclei, df$Mitosis),
                  "Type" = c("Clump Thickness", "Uniform Size", "Unifrom Shape",
                             "Marginal Adhesion", "Single Epithilial Size", "Bare Nuclei",
                             "Bland Chromatin", "Normal Nuclei", "Mitosis"), "class" = C(df$Class))
ggplot(ggdf, aes(x = Level, y = class)) + geom_boxplot() + facet_wrap(~Type)

train = sample(nrow(df), 0.7*nrow(df))
df.train = df[train,]
df.test = df[-train,]

model = glm(Class~., family = binomial, data=df.train)
summary(model)

vif(model)

model2 = glm(Class~.-Uniform_Size, family = binomial, data=df.train)
summary(model2)

model3 = glm(Class~.-Uniform_Shape, family = binomial, data=df.train)
summary(model3)

model8 = glm(Class~.-Uniform_Size-Single_Epith_Size, family = binomial, data=df.train)
summary(model8)

res = predict(model8, df.test, type = "response")
```

```
ROCRPRed = prediction(res, df.test$Class)
ROCRPerf = performance(ROCRPRed, "tpr", "fpr")
plot(ROCRPerf, colorize=TRUE, print.cutoffs.at=seq(0.1, by=0.2), xlim=c(0,0.1), ylim=c(0.8,1))

table(ActualValue=df.test$Class, PredictedValue=res>0.4)
table(ActualValue=df.test$Class, PredictedValue=res>0.7)

predicted.test = predict(model8, df.test, type = "response")
predicted.data = data.frame(prob.of.malig=predicted.test, malig = df.test$Class)
predicted.data = predicted.data[order(predicted.data$prob.of.malig, decreasing = F),]
predicted.data$rank = 1:nrow(predicted.data)

p1 = ggplot(data=predicted.data, aes(x=rank, y=prob.of.malig)) +
  geom_point(aes(color=malig), alpha=1, shape=4, stroke=2) +
  xlab("Index") + ylab("Predicted Probability of Tumor Being Malignant")
p1
```