Invasive Breast Cancer Diagnosis Using Logistic Regression

Travis Fekkers and Ryan Conroy

January 24, 2021

Abstract

Our research objective was to find the optimal logistic regression model for our data set, which analyzed patients with breast cancer, and gave statistics regarding different features and measurements of the patient's breast cancer tumor. Through our analysis, we aimed to effectively try to predict whether a patient's breast cancer tumor is either invasive (malignant) or non-invasive (benign). Our methods included a first-order logistical regression model, then utilizing a centered predictor model with interactions for all the predictor variables. To optimize this updated centered model, we put it through a step wise regression to find the optimal model to help predict the diagnosis of the tumor based on the predictors in the data set. Our final model concluded that a few centered predictors are most effective at predicting a patient's diagnosis.

Introduction

The data set was created by Dr. Wolberg beginning in 1984. Each instance in the data set is an individual who is experiencing a breast cancer tumor. The data set includes 30 features that were recorded from a digitized image of a fine needle aspirate of a breast tumor. The response variable in this data set is the cancer diagnosis of the patient. Patients with invasive breast cancer can be either diagnosed as M (malignant – abnormal cells divide uncontrollably and destroy body tissue) or B (Benign – Does not invade nearby tissue or spread other parts of the body). We will be using these predictor variables to predict if a breast cancer tumor is Malignant or Benign: radius mean, texture, area, smoothness, symmetry, concavity, compactness. All of these predictor fields are measurements and observations of given breast cancer tumors.

Data Characteristics

Below is a brief definition of all the variables of interest.

Radius: the mean of the distance from center to points on the perimeter Texture: the standard deviation of the gray-scale values Perimeter: the distance around the shape Area: the size of the cancer tumor Smoothness: local variation in radius length Concavity: severity of the concave portions of the contour

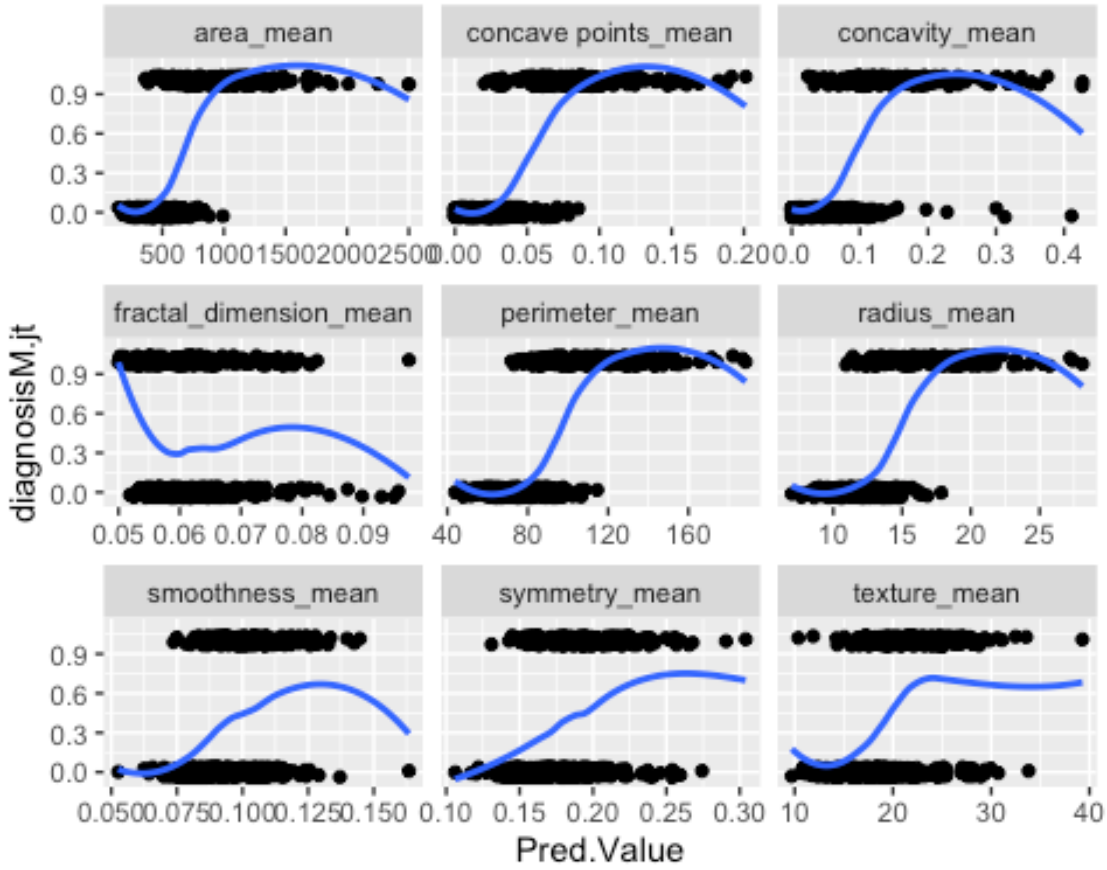
Response Variable- Diagnosis: Binary Variable, means malignant when 1, and benign when 0.

Import dataset

```
library(readxl)
data_2 <- read_excel("~/Desktop/data_2.xlsx")
#head(data_2)
```

Check the relationships between the diagnosis probability and the predictor variables

```
library (tidyr)
library(ggplot2)
hp.stack = data_2[, 2:11] %>%
  pivot_longer (!diagnosisM, names_to = "Pred.Name", values_to =
"Pred.Value")
hp.stack$diagnosisM.jt = jitter (hp.stack$diagnosisM, 0.2)

qplot (Pred.Value, diagnosisM.jt, data=hp.stack) +
  facet_wrap(~Pred.Name, scales = 'free_x') +
  geom_smooth (method = "loess", se=F)

## `geom_smooth()` using formula 'y ~ x'
```

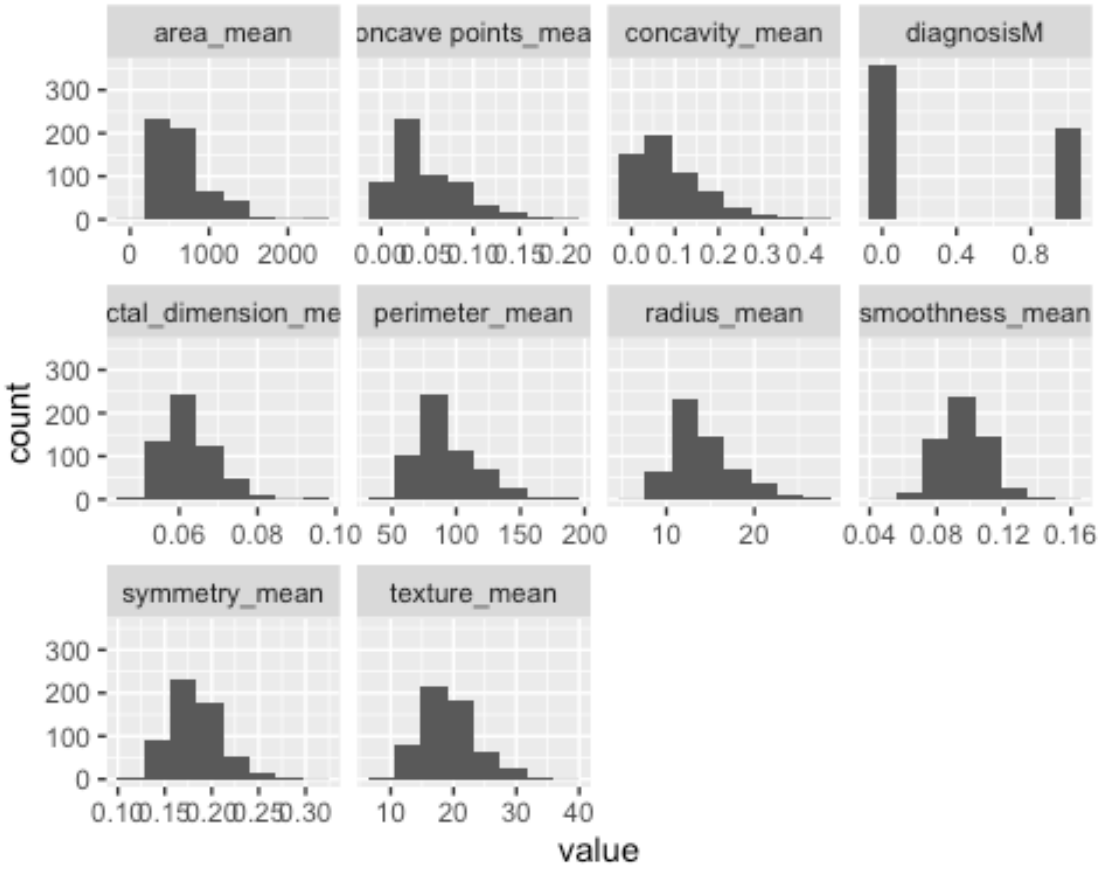


By analyzing each predictor variable's relationship with the diagnosis, we can see almost all of them have a strong positive relationship except for the fractal dimension mean.

However, it seems that smoothness has a downward curve as the smoothness units get to their highest values. This phenomenon seems to be due to a single outlier at the bottom right-hand corner of the plot. Besides that, all of the other predictors seem to be more likely to be malignant tumors if their values increase in units. Before we get too far, we should double-check each predictors distribution to determine if transformations may be necessary.

Check the distribution of Predictor Variables

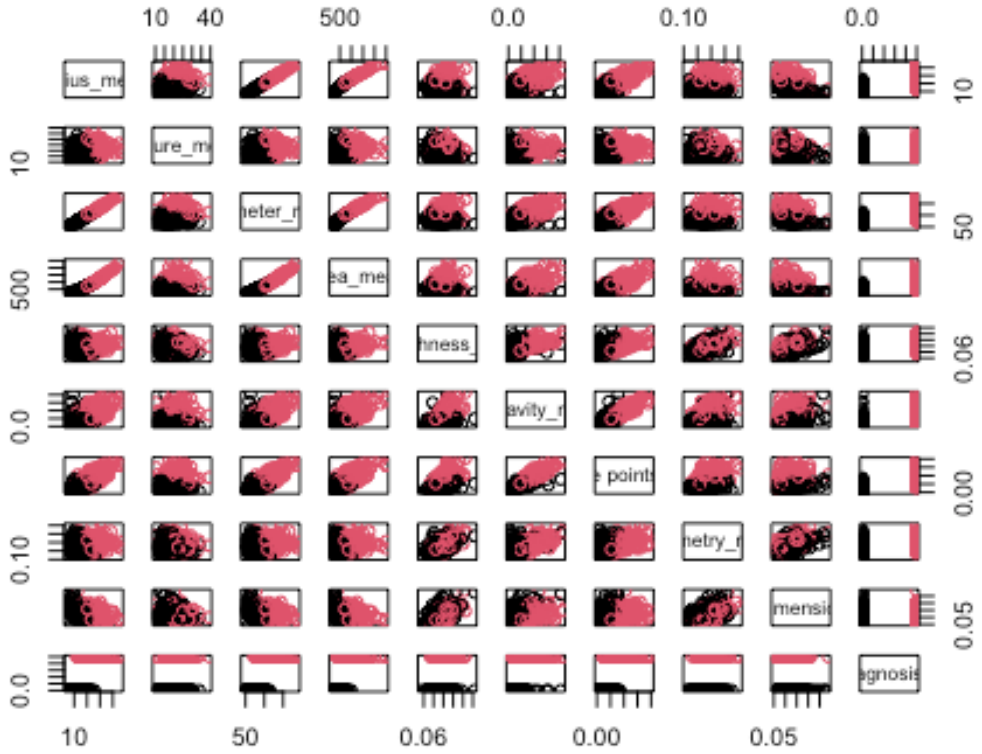
```
library (ggplot2)
library (tidyr)

ggplot(gather(data_2 [, 2:11]), aes(value)) +
  geom_histogram(bins = 8) +
  facet_wrap(~key, scales = 'free_x')
```



Most of the variables do not show concerning signs of obvious left or right skewness. Through these histograms, we can also see that the only variable within the data set that is categorical seems to be the response variable (diagnosisM). The rest seem to have distributions that are spread randomly throughout and are not simply a binary categorical variable. When analyzing the data set, we can also see that all fields seem to be qualitative, so no categorical label encoding needs to be done. Luckily, all categorical data is already binary (diagnosisM).

Check the Linear Relationships

```
pairs (data_2[,2:11],col=data_2$diagnosisM+1)
```




```
cormat = cor (data_2[,2:11], use = "complete.obs")
round (cormat, 2)
```

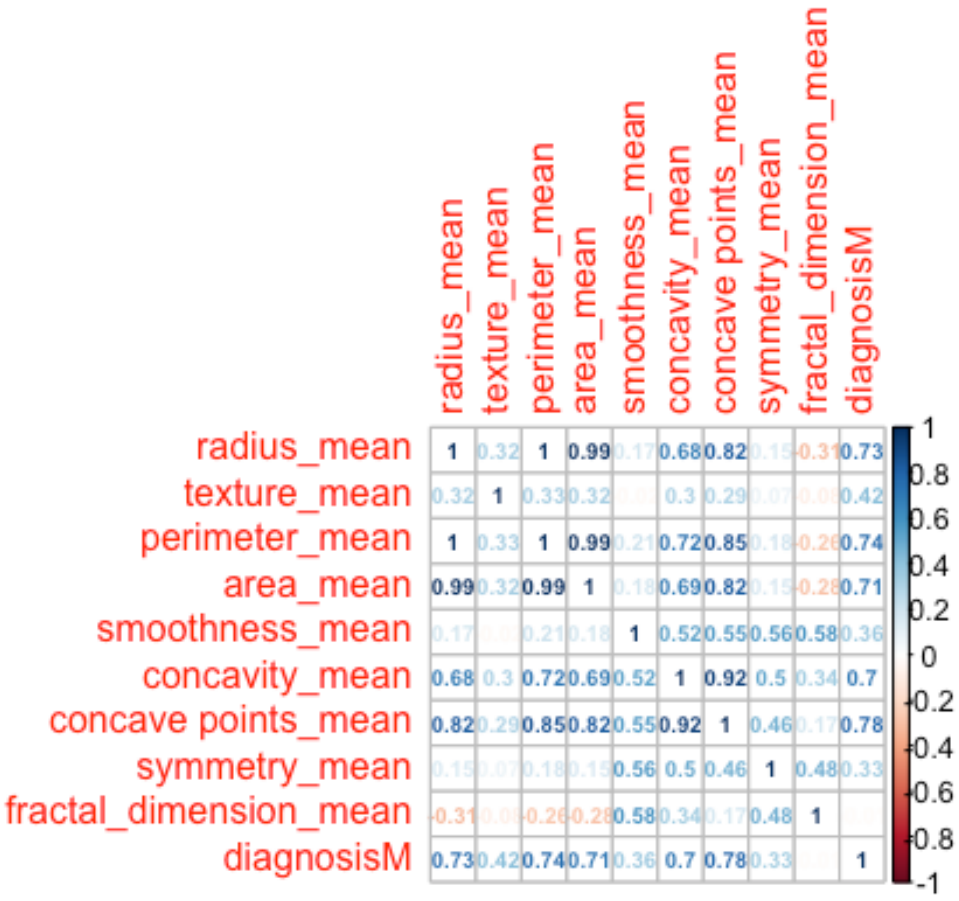
```
##                          radius_mean texture_mean perimeter_mean area_mean
## radius_mean                     1.00         0.32           1.00      0.99
## texture_mean                    0.32         1.00           0.33      0.32
## perimeter_mean                  1.00         0.33           1.00      0.99
## area_mean                       0.99         0.32           0.99      1.00
## smoothness_mean                 0.17        -0.02           0.21      0.18
## concavity_mean                  0.68         0.30           0.72      0.69
## concave points_mean             0.82         0.29           0.85      0.82
## symmetry_mean                   0.15         0.07           0.18      0.15
## fractal_dimension_mean         -0.31        -0.08          -0.26     -0.28
## diagnosisM                      0.73         0.42           0.74      0.71
##                          smoothness_mean concavity_mean concave points_mean
## radius_mean                         0.17           0.68                0.82
## texture_mean                       -0.02           0.30                0.29
## perimeter_mean                      0.21           0.72                0.85
## area_mean                           0.18           0.69                0.82
## smoothness_mean                     1.00           0.52                0.55
```

```
## concavity_mean                     0.52           1.00                0.92
## concave points_mean                0.55           0.92                1.00
## symmetry_mean                      0.56           0.50                0.46
## fractal_dimension_mean             0.58           0.34                0.17
## diagnosisM                         0.36           0.70                0.78
##                         symmetry_mean fractal_dimension_mean diagnosisM
## radius_mean                      0.15                 -0.31       0.73
## texture_mean                     0.07                 -0.08       0.42
## perimeter_mean                   0.18                 -0.26       0.74
## area_mean                        0.15                 -0.28       0.71
## smoothness_mean                  0.56                  0.58       0.36
## concavity_mean                   0.50                  0.34       0.70
## concave points_mean              0.46                  0.17       0.78
## symmetry_mean                    1.00                  0.48       0.33
## fractal_dimension_mean           0.48                  1.00      -0.01
## diagnosisM                       0.33                 -0.01       1.00
```

This scatter plot matrix gives us a good understanding of each variables relationship with the diagnosis. When the diagnosis is malignant, the plot shows red. Through the graphs, we can see that some of these predictor variables have an obvious positive relationship with the diagnosis. For example, we can see that as area_mean increases in units, the likelihood of being diagnosed as malignant is much higher. This same relationship seems to be obvious with perimeter_mean as well. However, some predictor variables such as texture_mean and concavity_mean don't seem to be so obvious.

Find the correlation of each predictor variable to the response variable

```
transcorr = cor(data_2 [,2:11], use= "complete.obs")
library("corrplot")

## corrplot 0.84 loaded

## corrplot 0.84 loaded
corrplot (transcorr, method = "number", number.cex=0.6)
```



We find that 5 pairs of variables with high correlation coefficients to our target variable diagnosisM. First, concave points_mean and diagnosisM have a correlation coefficient of 0.78. Secondly, perimeter and diagnosisM have a correlation coefficient of 0.74. Radius_mean and diagnosisM have a positive correlation of 0.73. Area mean and diagnosisM have a correlation of 0.71. Lastly, concavity mean and diagnosisM have a correlation of 0.7. There are multiple variables that have high correlation coefficients to our target variable diagnosisM.

Our model contains no categorical predictor, therefore there is no need for a two-way frequency table. All predictor variables are numeric and quantitative. Let's begin modeling.

Fit a first-order model

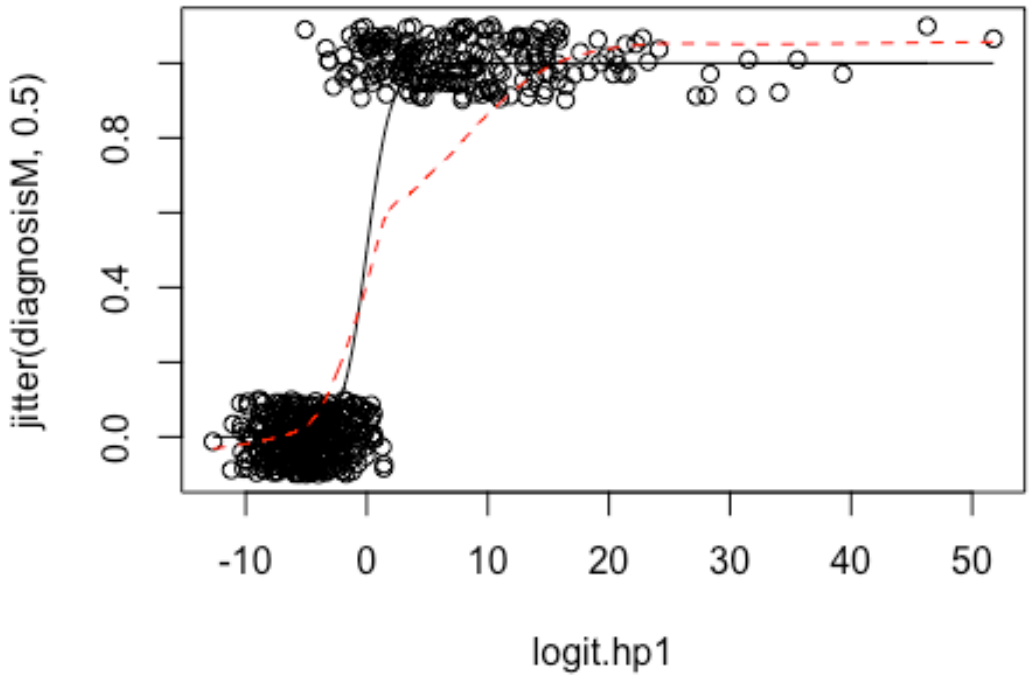
```
fit1 = glm (diagnosisM ~ radius_mean + texture_mean + perimeter_mean +
area_mean + smoothness_mean + concavity_mean  + symmetry_mean +
fractal_dimension_mean, data=data_2,family=binomial)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(fit1)

##
## Call:
## glm(formula = diagnosisM ~ radius_mean + texture_mean + perimeter_mean +
```

```
##     area_mean + smoothness_mean + concavity_mean + symmetry_mean +
##     fractal_dimension_mean, family = binomial, data = data_2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8104  -0.1591  -0.0347   0.0062   3.1944
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -14.70107   11.71594  -1.255 0.209554
## radius_mean              -2.81224    3.03923  -0.925 0.354802
## texture_mean              0.38622    0.06384   6.050 1.45e-09 ***
## perimeter_mean            0.16462    0.32591   0.505 0.613494
## area_mean                 0.03363    0.01582   2.126 0.033542 *
## smoothness_mean         128.20759   25.16244   5.095 3.48e-07 ***
## concavity_mean           21.21286    6.16832   3.439 0.000584 ***
## symmetry_mean            16.27599   10.71088   1.520 0.128618
## fractal_dimension_mean -113.86263   67.53598  -1.686 0.091804 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 151.93  on 560  degrees of freedom
## AIC: 169.93
##
## Number of Fisher Scoring iterations: 9
```

Through the summary table above, we can see that only a few of the fields seem to be significant at the 5% level. These significant fields are the texture_mean, area_mean, smoothness_mean, and concavity_mean. Realistically, these fields could be the most logical interpretations of predicting a malignant tumor. With our current knowledge of oncology and human biology, we feel like tumors that are larger (more area) could possibly be good indicators for if a tumor is malignant. One surprise was how perimeter was not a significant predictor. However, perhaps this field overlapped too much with area to be a significant field. One more surprise was smoothness and fractal dimensions coefficient's being so high. While the rest were hovering around 0, these two fields had coefficients over 100 units. Perhaps after going through step wise regression and adjusting the magnitudes, we can better understand how our predictors effect the diagnosis.

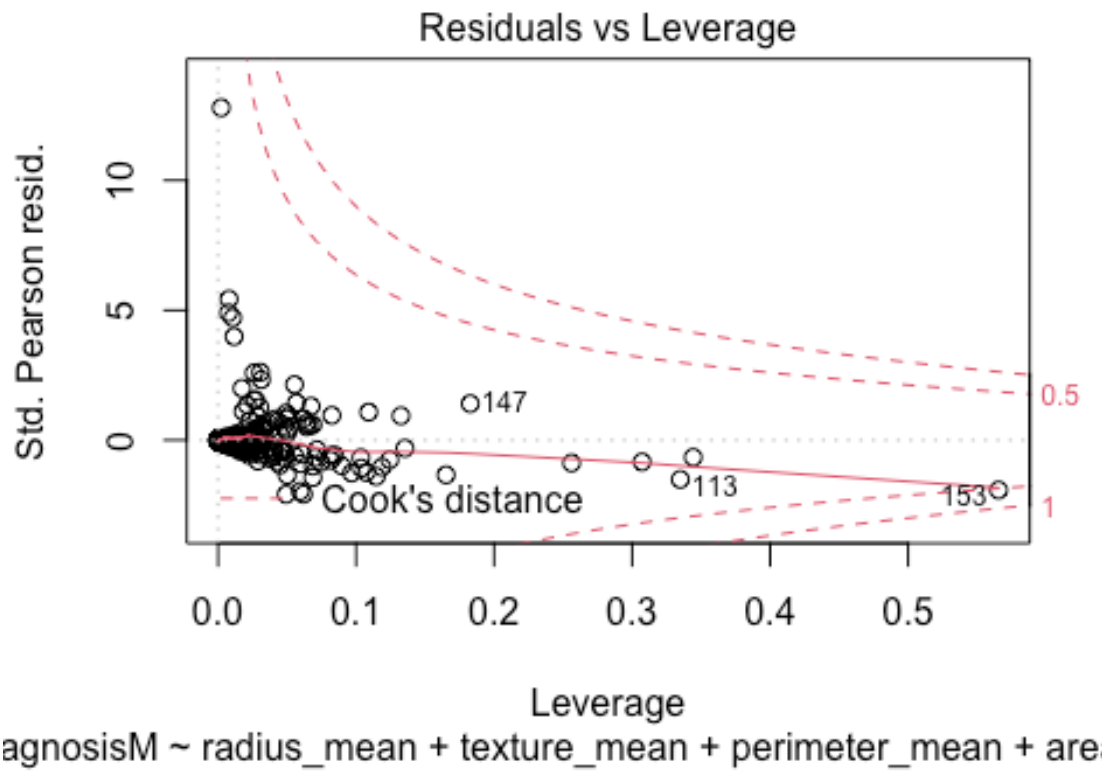
Check the residuals

```
logit.hp1 = predict (fit1)
plot (jitter (diagnosisM, 0.5) ~ logit.hp1, data=data_2)
pihat.hp1 = predict (fit1, type='response')
pihat.ord = order (pihat.hp1)
lines (logit.hp1 [pihat.ord], pihat.hp1 [pihat.ord])
lines (lowess (logit.hp1 [pihat.ord], pihat.hp1 [pihat.ord]), col='red',
lty=2)
```



As we can see from the plot of the initial model, there is a higher density of failures as the predicted logit value of the model increase. This gives us a good indication that the model may predict a tumor being benign or malignant better than chance.

```
plot(fit1, which=5)
```



Through the residual plots, we can see that that linearity seems to be alright by looking at the residual vs fitted plot. However, we can see a notable outlier labeled 298. This shows up on the normal quantile plot as well. We will see if this shows up in later models too, but for now, we will just note this as an outlier. There seems to be a bit of curvature in the normal quantile plot too. It seems to be relatively okay, but the plot seems to leave the line quite a bit at both edges of the plot. The Cook's distance plot also shows that the point at 153 may be flagged from this model. For right now, we will look to check an improved model through step wise regression, but it is important to note these points and outliers early.

```
car::vif (fit1)
```

```
##            radius_mean           texture_mean         perimeter_mean
##              633.088589              1.780396            305.997741
##              area_mean        smoothness_mean         concavity_mean
##              121.347630              2.990698              3.243271
##         symmetry_mean fractal_dimension_mean
##              1.911676              6.466964
```

The summary table shows that there are signs of multicollinearity within the first-order model with all of the predictor variables included. The cut-off is 5, and we see some extremely high VIF values >5. Let's put the model through a step wise regression to try to clean up the model.

Stepwise regression

```
cc = complete.cases (data_2)
stepfit = step (fit1, direction = 'both', k = log (sum (data_2)))

## Start:  AIC=364.09
## diagnosisM ~ radius_mean + texture_mean + perimeter_mean + area_mean +
##     smoothness_mean + concavity_mean + symmetry_mean +
fractal_dimension_mean

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                           Df Deviance    AIC
## - perimeter_mean           1   152.19 340.77
## - radius_mean              1   152.79 341.37
## - symmetry_mean            1   154.26 342.84
## - fractal_dimension_mean   1   154.88 343.46
## - area_mean                1   156.46 345.04
## - concavity_mean           1   164.00 352.59
## <none>                         151.93 364.09
## - smoothness_mean          1   184.42 373.00
## - texture_mean             1   202.42 391.00

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=340.77
## diagnosisM ~ radius_mean + texture_mean + area_mean + smoothness_mean +
##     concavity_mean + symmetry_mean + fractal_dimension_mean

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                          Df Deviance    AIC
## - radius_mean             1   153.55 318.56
## - symmetry_mean           1   154.49 319.50
## - fractal_dimension_mean  1   155.07 320.08
## - area_mean               1   156.63 321.64
## - concavity_mean          1   170.51 335.52
## <none>                        152.19 340.77
## - smoothness_mean         1   184.53 349.55
## + perimeter_mean          1   151.93 364.09
## - texture_mean            1   204.38 369.40

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=318.56
## diagnosisM ~ texture_mean + area_mean + smoothness_mean + concavity_mean +
##     symmetry_mean + fractal_dimension_mean

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                          Df Deviance    AIC
## - symmetry_mean           1   155.74 297.18
## - fractal_dimension_mean  1   156.32 297.76
## - concavity_mean          1   172.59 314.02
## <none>                        153.55 318.56
## - smoothness_mean         1   184.58 326.01
## + radius_mean             1   152.19 340.77
## + perimeter_mean          1   152.79 341.37
## - texture_mean            1   205.58 347.02
## - area_mean               1   218.72 360.15
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=297.18
## diagnosisM ~ texture_mean + area_mean + smoothness_mean + concavity_mean +
##     fractal_dimension_mean

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                           Df Deviance    AIC
## - fractal_dimension_mean   1   158.11 275.98
## - concavity_mean           1   178.68 296.55
## <none>                         155.74 297.18
## - smoothness_mean          1   195.58 313.44
## + symmetry_mean            1   153.55 318.56
## + radius_mean              1   154.49 319.50
## + perimeter_mean           1   155.02 320.03
## - texture_mean             1   207.04 324.91
## - area_mean                1   221.62 339.49

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=275.98
## diagnosisM ~ texture_mean + area_mean + smoothness_mean + concavity_mean

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##                            Df Deviance    AIC
## <none>                          158.11 275.98
## - concavity_mean            1   182.53 276.82
## - smoothness_mean           1   197.56 291.85
## + fractal_dimension_mean    1   155.74 297.18
## + symmetry_mean             1   156.32 297.76
## + perimeter_mean            1   156.80 298.24
## + radius_mean               1   156.93 298.37
## - texture_mean              1   209.73 304.02
## - area_mean                 1   340.60 434.89
```

```
summary (stepfit)
```

```
##
## Call:
## glm(formula = diagnosisM ~ texture_mean + area_mean + smoothness_mean +
##     concavity_mean, family = binomial, data = data_2)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.86649  -0.15162  -0.03710   0.01339   3.16165
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -30.887464   3.929528  -7.860 3.83e-15 ***
## texture_mean      0.381711   0.062645   6.093 1.11e-09 ***
## area_mean         0.015166   0.001932   7.849 4.19e-15 ***
## smoothness_mean 119.515647  21.141614   5.653 1.58e-08 ***
## concavity_mean   19.392935   3.876045   5.003 5.64e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 158.11  on 564  degrees of freedom
## AIC: 168.11
##
## Number of Fisher Scoring iterations: 8
```

Through a step wise regression on the first-order model, we can see that the optimal starting model should include texture_mean, area_mean, smoothness_mean, and concavity_mean. All of these seem to be significant at the 5% level. It is interesting to see that smoothness still has an extremely high coefficient for this model. Though the significance looks great, it is interesting to note that the residual deviance for this model went up from 151.93 to 158.11 compared to the first model. Therefore, though this is the optimal model, by removing the other predictors we now have a model with more deviance.

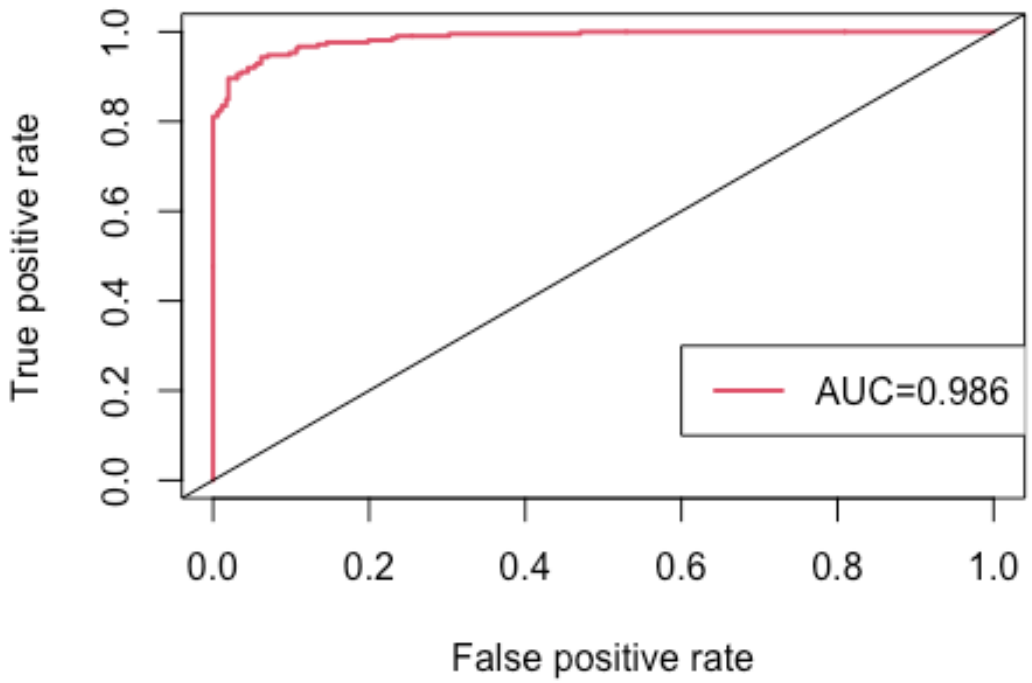
Now let's double-check for multicollinearity

```
car::vif (stepfit)

##    texture_mean       area_mean smoothness_mean  concavity_mean
##        1.671849        1.877026        2.118881        1.171662
```

The summary table shows that every variable from our step wise regression first-order model is okay to use and passes the assumption that there is no multicollinearity. The cutoff is to try to have every variable below 5. As we can see, none of the variables even exceed 3. This is a major improvement compared to our first initial model.

Put first-order model through an ROC Curve

```
par (mfrow=c(1,1))
library(ROCR)
pred1 <- prediction(stepfit$fitted.values, stepfit$y)
perf1 <- performance(pred1,"tpr","fpr")
auc1 <- performance(pred1,"auc")@y.values[[1]]
auc1

## [1] 0.9859548

plot(perf1, lwd=2, col=2)
abline(0,1)
legend(0.6, 0.3, c(paste ("AUC=", round (auc1, 4), sep="")),   lwd=2, col=2)
```



By analyzing the ROC curve and checking the AUC value given within the chart, we can see that this current model gives a very good indicator of predicting whether the breast cancer is malignant or not. AUC values track the amount of area under the curve, and the goal is to stay far away from the curve. In this model, we get an AUC value of 0.9859, which is a very good value, especially considering this is our first-order model.

Let's find the cut-off point for the first-order model

```
roc.x = slot (perf1, "x.values") [[1]]
roc.y = slot (perf1, "y.values") [[1]]
cutoffs = slot (perf1, "alpha.values") [[1]]

auc.table = cbind.data.frame(cutoff=pred1@cutoffs,
                             tp=pred1@tp, fp=pred1@fp, tn=pred1@tn,
fn=pred1@fn)
names (auc.table) = c("Cutoff", "TP", "FP", "TN", "FN")
auc.table$sensitivity = auc.table$TP / (auc.table$TP + auc.table$FN)
auc.table$specificity = auc.table$TN / (auc.table$TN + auc.table$FP)
auc.table$FalsePosRate = 1 - auc.table$specificity
auc.table$sens_spec = auc.table$sensitivity + auc.table$specificity

# Find the row(s) in the AUC table where sensitivity + specificity is
maximized

auc.best = auc.table [auc.table$sens_spec == max (auc.table$sens_spec),]
auc.best

##       Cutoff  TP FP  TN FN sensitivity specificity FalsePosRate sens_spec
## 44 0.3712637 200 22 335 12   0.9433962   0.9383754   0.06162465  1.881772

# Plot the maximum point(s) on the ROC plot

#points (auc.best$FalsePosRate, auc.best$sensitivity, cex=1.3)
```

It seems that the optimal cut-off point for this model is 0.3713. We can see that at this given cut-off point, the sensitivity and specificity of the model are both well over 0.90. These high indicators represent that there is a high probability that the model will predict the right outcome. Specifically for sensitivity, the outcome being at 0.94 represents that our model is 94% likely to predict a true positive (predicting malignant tumors when the tumors are actually malignant). Whereas, our model also has a 93.8% chance of predicting a true negative (predicting benign tumors when they are actually benign).

Let's create a centered interaction model

```
my.center = function (y) y - mean (y)
data_2$texture.c = my.center (data_2$texture_mean)
data_2$area.c = my.center (data_2$area_mean)
data_2$smoothness.c = my.center (data_2$smoothness_mean)
data_2$concavity.c = my.center (data_2$concavity_mean)
```

Fit the centered model with interaction

```
fit2 = glm (diagnosisM ~ (texture.c +  area.c + smoothness.c +
concavity.c)^2, data=data_2,family=binomial)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(fit2)

##
## Call:
## glm(formula = diagnosisM ~ (texture.c + area.c + smoothness.c +
##     concavity.c)^2, family = binomial, data = data_2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1490  -0.1605  -0.0719   0.0000   3.1914
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                7.123e-03  3.204e-01   0.022 0.982262
## texture.c                  5.996e-01  1.011e-01   5.933 2.97e-09 ***
## area.c                     1.870e-02  2.690e-03   6.953 3.59e-12 ***
## smoothness.c               1.468e+02  3.681e+01   3.988 6.67e-05 ***
## concavity.c                4.108e+01  1.005e+01   4.086 4.39e-05 ***
## texture.c:area.c           1.699e-03  6.320e-04   2.688 0.007195 **
## texture.c:smoothness.c     1.954e+01  6.084e+00   3.211 0.001321 **
## texture.c:concavity.c      1.637e+00  1.862e+00   0.879 0.379299
## area.c:smoothness.c       -6.907e-02  1.320e-01  -0.523 0.600892
## area.c:concavity.c         1.275e-01  4.638e-02   2.749 0.005983 **
## smoothness.c:concavity.c   2.232e+03  6.176e+02   3.614 0.000302 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 122.05  on 558  degrees of freedom
## AIC: 144.05
##
## Number of Fisher Scoring iterations: 10
```

Through the summary table above, it seems that almost all of the interactions are significant, and all of the original variables are still significant at the 5% level when centered. This is a good sign going forward for our model. Now let's put the model through a step wise regression to find an optimal model

Stepwise Regression on interaction effects

```
cc = complete.cases (data_2)
stepfit2 = step (fit2, direction = 'both', k = log (sum (data_2)))

## Start:  AIC=381.35
## diagnosisM ~ (texture.c + area.c + smoothness.c + concavity.c)^2

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                             Df Deviance    AIC
## - area.c:smoothness.c        1   122.31 358.04
## - texture.c:concavity.c      1   122.84 358.57
## - texture.c:area.c           1   131.47 367.20
## - area.c:concavity.c         1   132.41 368.14
## - texture.c:smoothness.c     1   134.35 370.08
## - smoothness.c:concavity.c   1   144.71 380.44
## <none>                           122.05 381.35

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=358.04
## diagnosisM ~ texture.c + area.c + smoothness.c + concavity.c +
##     texture.c:area.c + texture.c:smoothness.c + texture.c:concavity.c +
##     area.c:concavity.c + smoothness.c:concavity.c

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##                             Df Deviance    AIC
## - texture.c:concavity.c      1   123.38 335.53
## - texture.c:area.c           1   131.58 343.74
## - area.c:concavity.c         1   133.69 345.85
## - texture.c:smoothness.c     1   134.98 347.13
## - smoothness.c:concavity.c   1   144.72 356.88
## <none>                           122.31 358.04
## + area.c:smoothness.c        1   122.05 381.35

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=335.53
## diagnosisM ~ texture.c + area.c + smoothness.c + concavity.c +
##     texture.c:area.c + texture.c:smoothness.c + area.c:concavity.c +
##     smoothness.c:concavity.c

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                             Df Deviance    AIC
## - texture.c:area.c           1   135.06 323.64
## - area.c:concavity.c         1   138.74 327.32
## - texture.c:smoothness.c     1   142.86 331.45
## - smoothness.c:concavity.c   1   144.72 333.31
## <none>                           123.38 335.53
## + texture.c:concavity.c      1   122.31 358.04
## + area.c:smoothness.c        1   122.84 358.57

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=323.64
## diagnosisM ~ texture.c + area.c + smoothness.c + concavity.c +
##     texture.c:smoothness.c + area.c:concavity.c + smoothness.c:concavity.c

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                             Df Deviance    AIC
## - area.c:concavity.c         1   146.07 311.08
## - texture.c:smoothness.c     1   147.46 312.47
## - smoothness.c:concavity.c   1   153.49 318.50
## <none>                           135.06 323.64
## + texture.c:area.c           1   123.38 335.53
## + texture.c:concavity.c      1   131.58 343.74
## + area.c:smoothness.c        1   134.59 346.75

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=311.08
## diagnosisM ~ texture.c + area.c + smoothness.c + concavity.c +
##     texture.c:smoothness.c + smoothness.c:concavity.c

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                             Df Deviance    AIC
## - texture.c:smoothness.c     1   153.34 294.78
## - smoothness.c:concavity.c   1   153.84 295.28
## <none>                           146.07 311.08
## + area.c:concavity.c         1   135.05 323.64
## + texture.c:area.c           1   138.74 327.32
## + texture.c:concavity.c      1   138.90 327.48
## + area.c:smoothness.c        1   145.89 334.48
## - area.c                     1   330.25 471.69
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=294.78
## diagnosisM ~ texture.c + area.c + smoothness.c + concavity.c +
##     smoothness.c:concavity.c

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                             Df Deviance    AIC
## - smoothness.c:concavity.c  1   158.11 275.98
## <none>                          153.34 294.78
## + texture.c:concavity.c     1   142.47 307.48
## + texture.c:smoothness.c    1   146.07 311.08
## + area.c:concavity.c        1   147.46 312.47
## + texture.c:area.c          1   151.50 316.51
## + area.c:smoothness.c       1   153.24 318.25
## - texture.c                 1   203.75 321.62
## - area.c                    1   339.85 457.71

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=275.98
## diagnosisM ~ texture.c + area.c + smoothness.c + concavity.c

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                            Df Deviance    AIC
## <none>                          158.11 275.98
## - concavity.c               1   182.53 276.82
## - smoothness.c              1   197.56 291.85
## + smoothness.c:concavity.c  1   153.34 294.78
## + texture.c:smoothness.c    1   153.84 295.28
## + texture.c:concavity.c     1   155.37 296.80
## + texture.c:area.c          1   155.91 297.34
## + area.c:concavity.c        1   157.77 299.20
## + area.c:smoothness.c       1   158.11 299.55
## - texture.c                 1   209.73 304.02
## - area.c                    1   340.60 434.89
```

```
summary (stepfit2)
```

```
##
## Call:
## glm(formula = diagnosisM ~ texture.c + area.c + smoothness.c +
##     concavity.c, family = binomial, data = data_2)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.86649  -0.15162  -0.03710   0.01339   3.16165
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.353869   0.218852  -1.617    0.106
## texture.c      0.381711   0.062645   6.093 1.11e-09 ***
## area.c         0.015166   0.001932   7.849 4.19e-15 ***
## smoothness.c 119.515647  21.141614   5.653 1.58e-08 ***
## concavity.c   19.392935   3.876045   5.003 5.64e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 158.11  on 564  degrees of freedom
## AIC: 168.11
##
## Number of Fisher Scoring iterations: 8
```

Through a step wise regression on our centered model with interaction effects, we found that the step-wise procedure retained the predictors texture.c, area.c, smoothness.c, and

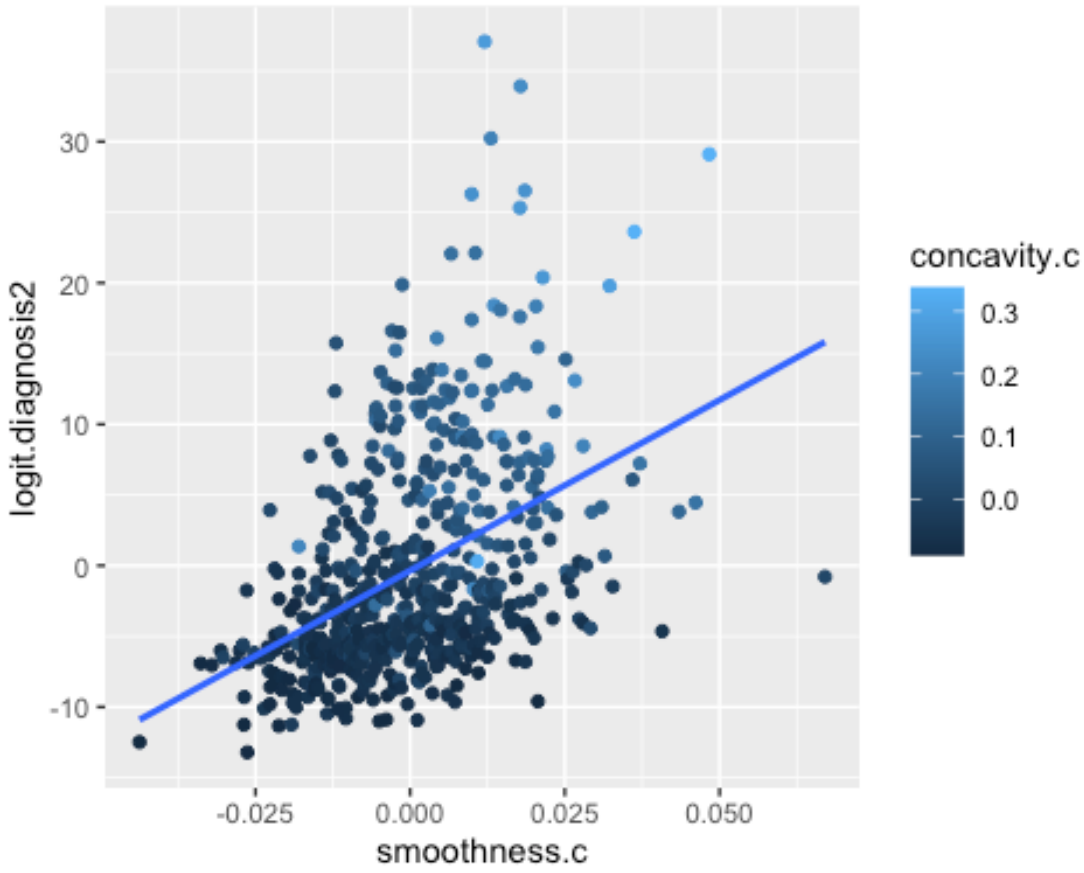
concavity.c. This is an interesting find for our model, it seems that none of the interactions helped generate the best model, and it seems that our model looks exactly like the original step wise regressed first-order model, except that the variables are centered. Therefore, this must be the optimal model to use when trying to find whether a cancerous tumor is malignant or not. One thing to note, is smoothness seems to have an extremely high impact on deciding if the tumor is malignant or not. It seems to be statistically significant, however, it still seems odd that this variable has such a higher impact on predicting our response variable.

We can see that there were no interaction effects left within our final model. However, let's interpret the effect of the most significant interaction effects using an interaction plot.

Interpret most significant interaction effect from prior model

Plot and interpret the smoothness by concavity interaction:

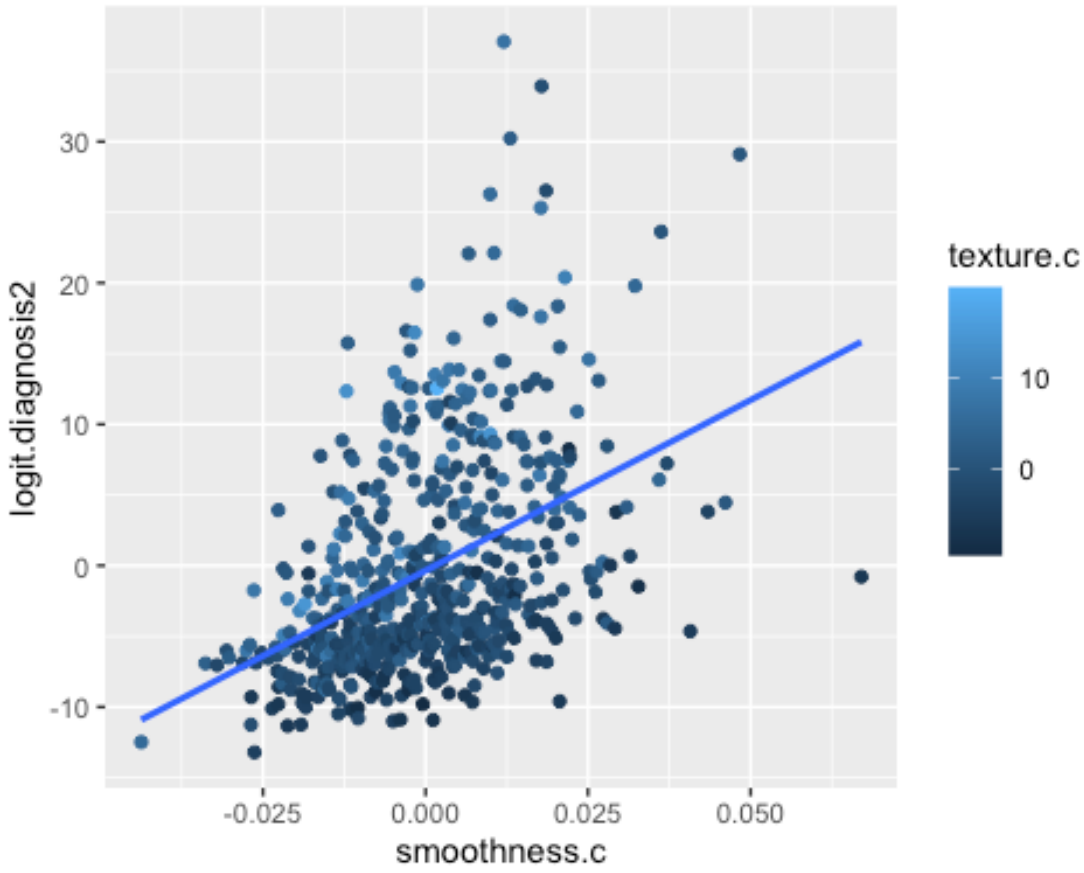
```
library (ggplot2)

# Save predicted values in the data frame
data_2$logit.diagnosis2 = predict (stepfit2)
data_2$pihat.diagnosis2 = predict (stepfit2, type='response')

library (dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

# Average the predicted values by smoothness and concavity
by.agpc = group_by (data_2, smoothness.c, concavity.c)
summ.agpc = summarize (by.agpc,
                       logit.diagnosis2 = mean (logit.diagnosis2),
                       pihat.diagnosis2 = mean (pihat.diagnosis2))

## `summarise()` regrouping output by 'smoothness.c' (override with `.groups`
argument)

# Interaction plot using predicted values averaged across concavity
qplot (smoothness.c, logit.diagnosis2, data=summ.agpc, color=concavity.c) +
  geom_smooth (method="lm", se=F)

## `geom_smooth()` using formula 'y ~ x'
```



The interaction between smoothness.c and concavity.c seems to be a strong positive relationship. Remember, this interaction is not used in the final model, however, we can see there seems to be a decent correlation between the two. We can see the observations with higher concavity have a lighter color, and we can see more of the lighter colored observations tend to be further on the x-axis compared to the darker ones (lower concavity). Therefore, it seems that as smoothness increases, the concavity in the tumor also seems to increase.

Plot and interpret the smoothness by texture interaction:

```
# Average the predicted values by smoothness and concavity
by.agpc = group_by (data_2, smoothness.c, texture.c)
summ.agpc = summarize (by.agpc,
                       logit.diagnosis2 = mean (logit.diagnosis2),
                       pihat.diagnosis2 = mean (pihat.diagnosis2))

## `summarise()` regrouping output by 'smoothness.c' (override with `.groups`
argument)

# Interaction plot using predicted values averaged across concavity
qplot (smoothness.c, logit.diagnosis2, data=summ.agpc, color=texture.c) +
  geom_smooth (method="lm", se=F)

## `geom_smooth()` using formula 'y ~ x'
```



The second most significant interaction (also not used in the model) was between texture and smoothness. The relationship between these two does not look as obvious compared to smoothness and concavity. We can see that observations with higher texture are lighter colors than ones with less texture. It seems that there is no clear relationship between how texture and smoothness interact with each other. There seems to be light dots scattered all around and dark spots scattered in similar places. I cannot make a clear correlation for this interaction, even though it seemed to be the second most significant interaction. Let's move onto more analysis of the final model.

Final Model Analysis

```
summary(stepfit2)

##
## Call:
## glm(formula = diagnosisM ~ texture.c + area.c + smoothness.c +
##     concavity.c, family = binomial, data = data_2)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.86649  -0.15162  -0.03710   0.01339   3.16165
##
```

```
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.353869   0.218852  -1.617    0.106
## texture.c      0.381711   0.062645   6.093 1.11e-09 ***
## area.c         0.015166   0.001932   7.849 4.19e-15 ***
## smoothness.c 119.515647  21.141614   5.653 1.58e-08 ***
## concavity.c   19.392935   3.876045   5.003 5.64e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 158.11  on 564  degrees of freedom
## AIC: 168.11
##
## Number of Fisher Scoring iterations: 8
```

We have already briefly discussed our final model's results earlier in the report, however, let's dig into the details. As discussed earlier, all parameter estimate p-values are statistically significant at the 5% level. This is great news at first glance. We can also analyze the parameter estimation results. When we record the odds ratios, we will dig deeper into what each coefficient means, but at first glance, we can, once again, see that smoothness and concavity are much larger than the other coefficients. Through analyzing, we can see that smoothness has much smaller unit scales compared to the other values. For example, the area variable has a range between 0 and 25,000, while the smoothness variable has a range between 0 and 0.16. We can see that to account for these massive range differences, the coefficient for smoothness should be higher to truly show an effect on being diagnosed with an invasive tumor. This same phenomenon exists with concavity (who had a coefficient of 19.39), where its range was between 0 and 0.4. Much like smoothness, the large coefficient helps to better interpret the effect of the variables on the diagnosis, given that the scale is much smaller than the other predictor variables.

Since the coefficients are so large, we can see that smoothness and concavity have larger standard errors. Because the coefficients for these variables have such magnitude, the standard error is much larger than we see with the other predictor variables (which all have coefficients that are between 0 and 1).

Another major point to note is the z-values of each coefficient to determine each variable's relative effect on predicting the diagnosis of each patient. We can see that area has the highest z-value relative to the others, and therefore, can be said to have the largest influence on determining the diagnosis. Since each z-value seems to be between 5 and 7, we can see that each variable in our final model seems to have a fairly large and significant effect on determining the diagnosis of each patient. However, it is interesting to see that area and texture seem to be the best indicators in the model at predicting the response variable.

Odds Ratios and Predicted Confidence Intervals

```
exp(stepfit2$coefficients*c(1,10,500,.05,.1))

##  (Intercept)    texture.c       area.c smoothness.c  concavity.c
##    0.7019672   45.4723890 1964.2809239  393.7760440    6.9538363

##each odds ratio has a different scale now and interpretations need to
address the different scale for each change in units
exp (confint (stepfit2)*c(1,10,500,.05,.1))

## Waiting for profiling to be done...

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                    2.5 %       97.5 %
## (Intercept)    0.4553288     1.079405
## texture.c     14.3337211   170.090627
## area.c       352.2120627 15917.620964
## smoothness.c  54.1879509  3532.949377
## concavity.c    3.3175471    15.741972
```

In the summary table above, we can see the odds ratios for each variable from the final model. Their interpretations are below (Note: look below to understand the actual changes since they are all on different ratio scales).

For each 1/10 (0.1) unit increase in texture, a person is 45.47 times more likely to be diagnosed with malignant breast cancer.

For each 1/500 (0.002) unit increase in area, a person is 1,964.28 times more likely to be diagnosed with malignant breast cancer.

For each 1/0.5 (2) unit increase in smoothness, a person is 393.78 times more likely to be diagnosed with malignant breast cancer.

For each 1/0.1 (10) unit increase in concavity, a person is 6.95 times more likely to be diagnosed with malignant breast cancer.

In the summary table above, we can also see the confidence intervals for each given variable in the model. These interpretations are below:

For each 1/10 (0.1) unit increase in texture, we can be 95% confident that the odds ratio for being diagnosed with malignant breast cancer is between 14.33 and 170.09 times higher.

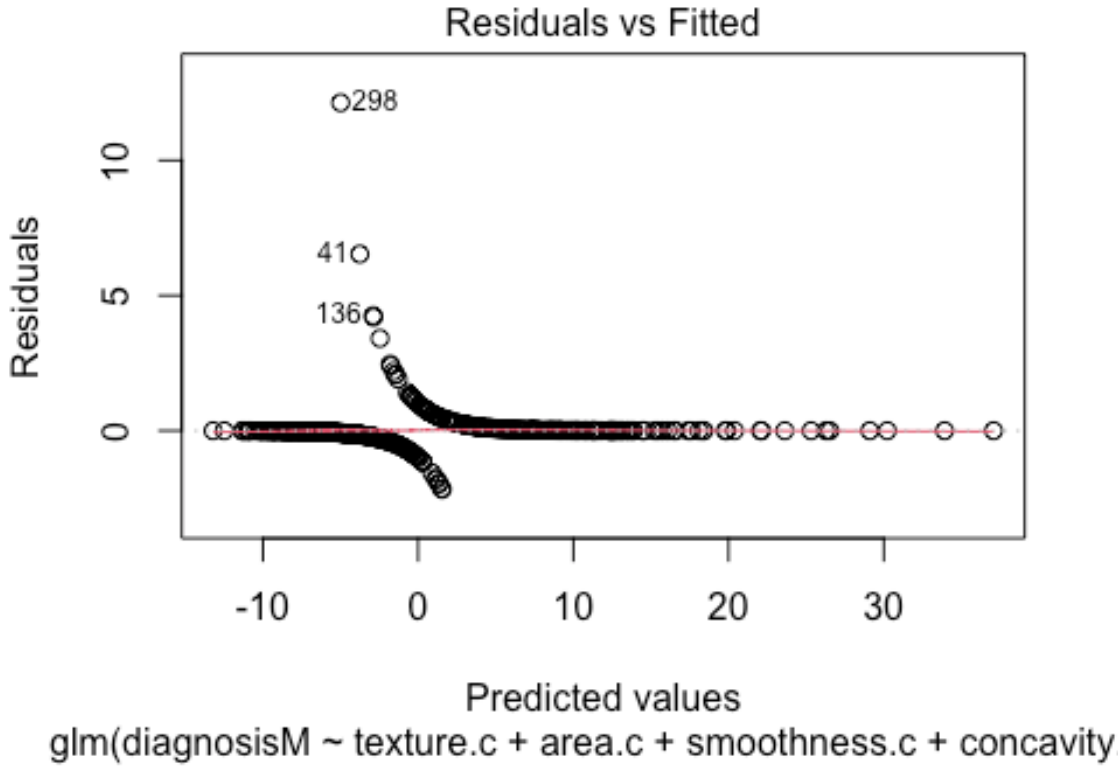
For each 1/500 (0.002) unit increase in area, we can be 95% confident that the odds ratio for being diagnosed with malignant breast cancer is between 352.21 and 1,5917.62 times higher.

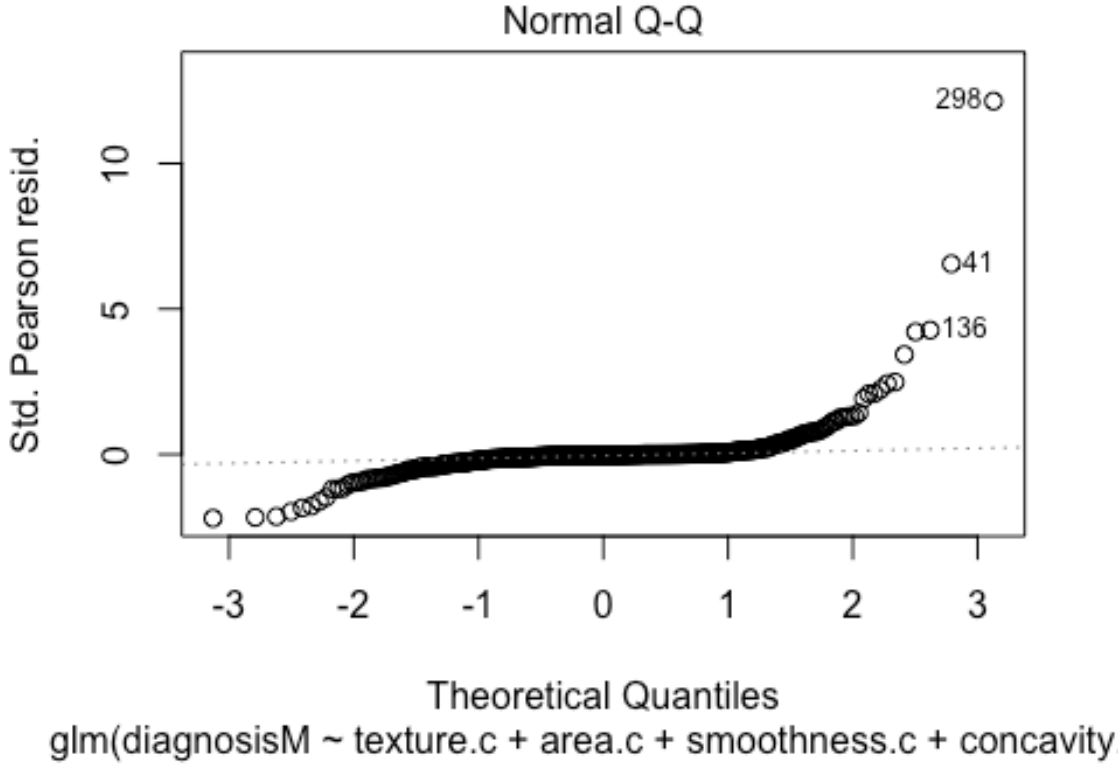
For each 1/0.5 (2) unit increase in smoothness, we can be 95% confident that the odds ratio for being diagnosed with malignant breast cancer is between 54.19 and 3,532.95 times higher.

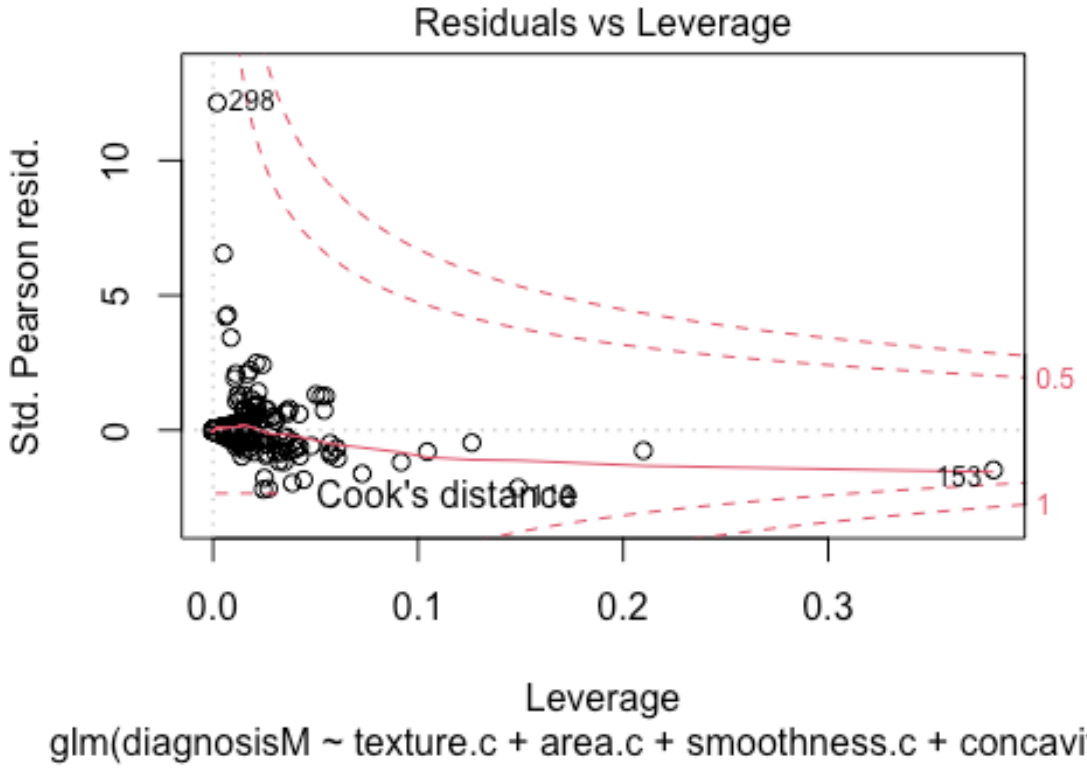
For each 1/0.1 (10) unit increase in concavity, we can be 95% confident that the odds ratio for being diagnosed with malignant breast cancer is between 3.32 and 15.74 times higher.

Residual Diagnositics on Final Model

```
plot(stepfit2, which=c(1,2,5))
```


Residuals vs Fitted
298
41
136
Residuals
10
5
0
-10
0
10
20
30
Predicted values
glm(diagnosisM ~ texture.c + area.c + smoothness.c + concavity


Normal Q-Q
Std. Pearson resid.
298
41
136
10
5
0
-3
-2
-1
0
1
2
3
Theoretical Quantiles
glm(diagnosisM ~ texture.c + area.c + smoothness.c + concavity



The residual plots look almost identical to the ones discussed earlier in the report because we are using the same variables, and the only difference is that they are now centered. By analyzing the residual vs. fitted plot, we can see a few large outliers toward the top of the second plot of lines. It seems that rows 298, 41, and 136 contain large outliers relative to the other residuals in the model. Overall, the plot seems to be consistent with linearity, however, it is important to note that these points do not seem to fit with the rest of the model.

By analyzing the normal quantile plot, we can see that points 136, 41, and 298 come up as large outliers again. Besides these points, there seems to be a bit of curvature toward the bottom left and top right areas of the plot. Though there is a bit of curvature and outliers, the residual plot seems to be adequate enough to assume a normal distribution for our model. Though it is not perfect, the plot does not show enough obvious evidence to claim the model is not normally distributed.

The Cook's distance plot also shows that the point at 153 may be flagged from this model. This means that this given point may have more leverage than the others. This point is extremely close to the 0.5 mark on the Cook's Distance scale, however, it is not quite within that range yet. This row should be noted as containing possibly more leverage than the others, but deleting the row may not be necessary. In general, we must acknowledge this row with high leverage, but additional deletion does not seem necessary given the location of the point on this plot.

P-values for goodness-of-fit test and likelihood ratio test, with interpretations

Lack of Fit test

```
pchisq(deviance(stepfit2),df.residual(stepfit2), lower=F )

## [1] 1
```

The null hypothesis for a lack of fit test is that the model is reasonable and no obvious lack of fit for the model. Since we got a perfect 1 for the p-value, we can conclude that the model is reasonable to use for predicting the diagnosis for breast cancer.

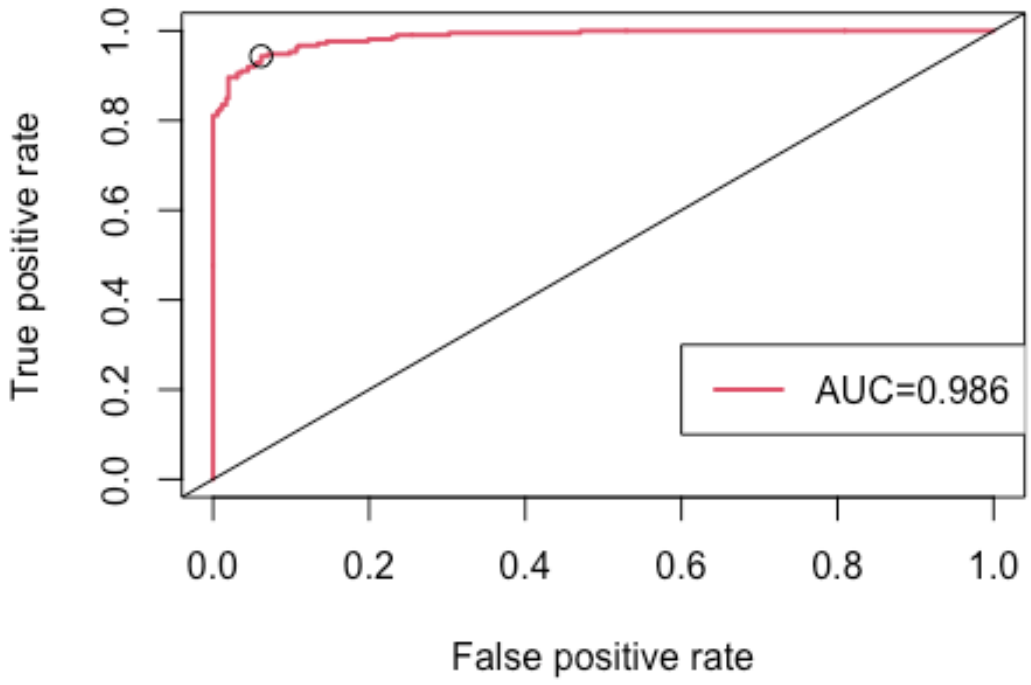
We will now conduct a likelihood ratio test in order to determine if the model is significant

```
1 - pchisq(stepfit2$null.deviance - stepfit2$deviance, stepfit2$df.null -
stepfit2$df.residual)

## [1] 0
```

We yield a p-value of p<.05. This means ours final model results has a highly significant difference between the null deviance and residual deviance.

```
par (mfrow=c(1,1))
library(ROCR)
pred1 <- prediction(stepfit2$fitted.values, stepfit2$y)
perf1 <- performance(pred1,"tpr","fpr")
auc1 <- performance(pred1,"auc")@y.values[[1]]
auc1

## [1] 0.9859548

plot(perf1, lwd=2, col=2)
abline(0,1)
legend(0.6, 0.3, c(paste ("AUC=", round (auc1, 4), sep="")),   lwd=2, col=2)


roc.x = slot (perf1, "x.values") [[1]]
roc.y = slot (perf1, "y.values") [[1]]
cutoffs = slot (perf1, "alpha.values") [[1]]
auc.table = cbind.data.frame(cutoff=pred1@cutoffs,
 tp=pred1@tp, fp=pred1@fp, tn=pred1@tn,
fn=pred1@fn)
names (auc.table) = c("Cutoff", "TP", "FP", "TN", "FN")
auc.table$sensitivity = auc.table$TP / (auc.table$TP + auc.table$FN)
auc.table$specificity = auc.table$TN / (auc.table$TN + auc.table$FP)
auc.table$FalsePosRate = 1 - auc.table$specificity
auc.table$sens_spec = auc.table$sensitivity + auc.table$specificity
```

```
auc.best = auc.table [auc.table$sens_spec == max (auc.table$sens_spec),]
auc.best

##       Cutoff  TP FP  TN FN sensitivity specificity FalsePosRate sens_spec
## 44 0.3712637 200 22 335 12   0.9433962   0.9383754   0.06162465  1.881772

points (auc.best$FalsePosRate, auc.best$sensitivity, cex=1.3)
```



After analyzing the ROC curve and looking at the AUC value, we can see our final model does a great job of predicting if breast cancer is benign or malignant. The final model ROC curve looks very similar to our first-order model's ROC Curve. Our AUC value of 98.59 is very close to 1 and shows that our model is effective at predicting if a tumor is benign or malignant. The ROC curve suggests the predictive ability of this model is better than random guessing since the AUC (0.986) is larger than 0.5. The optimal cutoff for classification is a fitted probability of 0.3712637. This cutoff has a specificity of 0.9383754, which means the false positive rate is 0.0617. The cutoff has a sensitivity (true positive rate) of 0.9433962. The cutoff is shown as a black circle on the ROC curve.

Conclusion

Through the report above, we can begin to see that our model seems to do a good job of predicting the diagnosis of the tumor for breast cancer patients. Through our results in the lack of fit test and the likelihood ratio test, we can conclude that our model is reasonable to

use, and an optimal model compared to a smaller (or null) model. We can also see that through our ROC curve, our model has an extremely close AUC value to 1 (which basically means perfect interpretation). At the optimal cutoff, our model has a true positivity rate of 94.34% and a true negative rate of 93.84%. Both of these test results are pretty high and explain how our model seems to do a good job of explaining whether the breast cancer tumor is invasive or not. Through interpreting our coefficients, we can also see that the area of the tumor seems to have the most influence on determining the diagnosis of the patient. The area has the highest z-value which has a p-value that is significant at the 1% level. We can also see that the next most influential variable seems to be texture, which is also significant at the 1% level. This claim seems fairly reasonable too, where there seems to be a little bit of research about texture's influence on predicting the diagnosis of tumors (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3452665/). I could not find a direct article that shows research on the area of the breast tumor and its influence on predicting the diagnosis, however, we can make a reasonable hypothesis to claim that a larger breast tumor should likely have a higher likelihood of being an invasive tumor.