Exam 2

Louis Bensard

April 24, 2018

Problem 1:

BMI numerical:

First of all, let's clean up the data to make it usable and keep only what's necessary for our model:

```
data = read.csv("C:/Users/Louis/Documents/UPMC/M1/Spring 2018/MATH 535/midterm2/colon2017.csv",
                header=T, stringsAsFactors = FALSE)

#cleaning up data and keeping only the necessary
data1 = data
data1 = data1[,-20:-33] #not using those predictors here
data1 = data1[,-1:-4] #we get rid of the patient ID as well, irrelevant
data1 = data1[,-12:-13] #not necessary
data1 = data1[,-2:-3] #we are going to use BMI and not Height and Weight
data1[data1$Race =="White",4]="W"
data1[data1$Race =="white",4]="W"
data1$Race[127] = "W" #glitch in the matrix, need to change this one manually
```

Now, let:

```
x1 = data1$Gender #categorical
x2 = data1$BMI #numerical
x3 = data1$Age #numerical
x4 = data1$Race #categorical
x5 = data1$Tobacco #categorical
x6 = data1$DM #categorical
x7 = data1$CAD.PAD #categorical
x8 = data1$Cancer #categorical
x9 = data1$Albumin..g.dL. #numerical
x10 = data1$Operative.Length #numerical
y = data1$Anastamotic.Leak #categorical
```

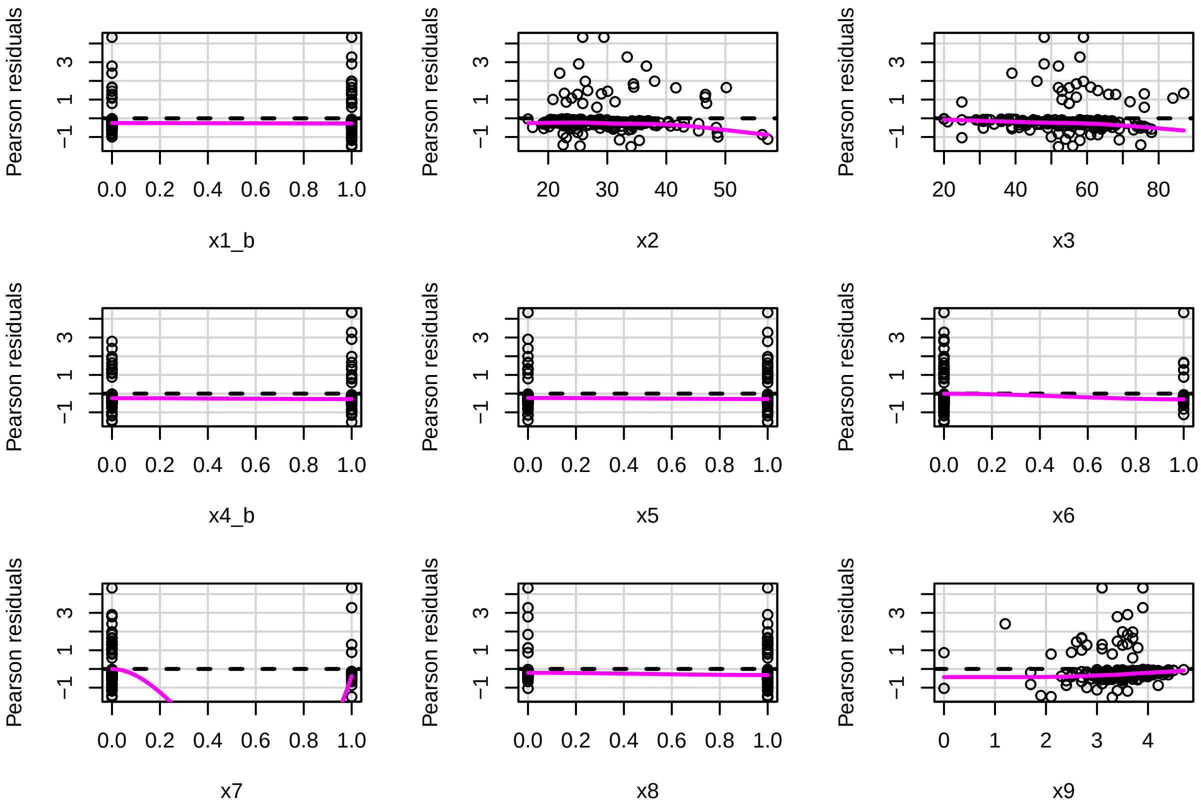
Let's start with a linear model and let's use the glm() function to perform a logistic regression of the model:

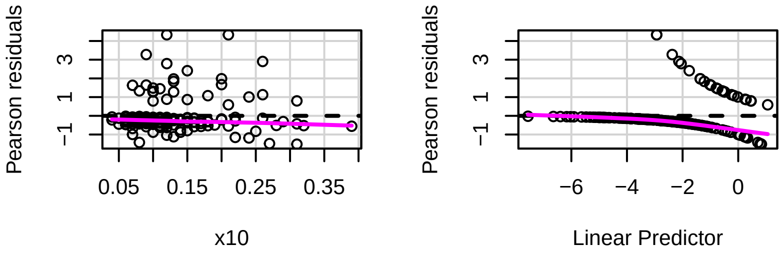
$$y \sim x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}$$

```
model = glm(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+x10, family = "binomial", data=data1)

library(car)
```

```
## Warning: package 'car' was built under R version 3.4.4

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.4.4
```

```r
#All categorical variables need to be 0 or 1 for the residualPlots() function from car package to work
#so I change temporarely the predictors x1 (Gender) and x4 (Race) to 0's and 1's
#???but keeping the same basic linar model
x1_b = x1; x4_b = x4
x1_b[x1_b=="Male"]=1 ; x1_b[x1_b=="Female"]=0; x1_b = as.numeric(x1_b)
x4_b[x4_b=="AA"]=1 ; x4_b[x4_b=="W"]=0; x4_b = as.numeric(x4_b)

model_b = glm(y ~ x1_b+x2+x3+x4_b+x5+x6+x7+x8+x9+x10, family = "binomial", data=data1)

residualPlots(model_b)
```





```
##       Test stat Pr(>|Test stat|)
## x1_b     0.0000          1.00000
## x2       1.9020          0.16786
## x3       0.0056          0.94010
## x4_b     0.0000          1.00000
## x5       0.0000          1.00000
## x6       0.0000          1.00000
## x7       0.0000          1.00000
## x8       0.0000          1.00000
## x9       0.0181          0.89288
## x10      3.8207          0.05062 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All residual plots look good, overall the errors are nicely spread around 0, except for x7 that is acting up, but I don't worry too much about x7 as we will see that it is not a significant predictor.

Therefore, we are going to stick with a linear model, we will see in problem 2 what predictor are significant.

```
summary(model)
```

```
##
## Call:
## glm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10, family = "binomial", data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5455  -0.4899  -0.3060  -0.1282   2.4431
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.92114    2.09214  -3.308 0.000939 ***
## x1Male       0.82283    0.51895   1.586 0.112839
## x2           0.09486    0.03227   2.940 0.003282 **
## x3           0.08437    0.02646   3.188 0.001431 **
## x4W         -0.23469    0.51959  -0.452 0.651494
## x5           1.02614    0.57343   1.789 0.073541 .
## x6          -0.85182    0.64633  -1.318 0.187529
## x7          -0.68212    0.73646  -0.926 0.354334
## x8           0.59864    0.59499   1.006 0.314348
## x9          -1.36235    0.36698  -3.712 0.000205 ***
## x10          7.05757    3.60032   1.960 0.049965 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 148.35  on 178  degrees of freedom
## Residual deviance: 111.56  on 168  degrees of freedom
## AIC: 133.56
##
## Number of Fisher Scoring iterations: 6
```

$x2 = BMI$ is the predictor of interest here. If BMI increases of 1 unit, then the confidence interval of the corresponding Odds Ratio is:

$$CI_1 = exp\{\beta_{BMI} \pm t^* \cdot ste(\beta_{BMI})\}$$

Thus, we have:

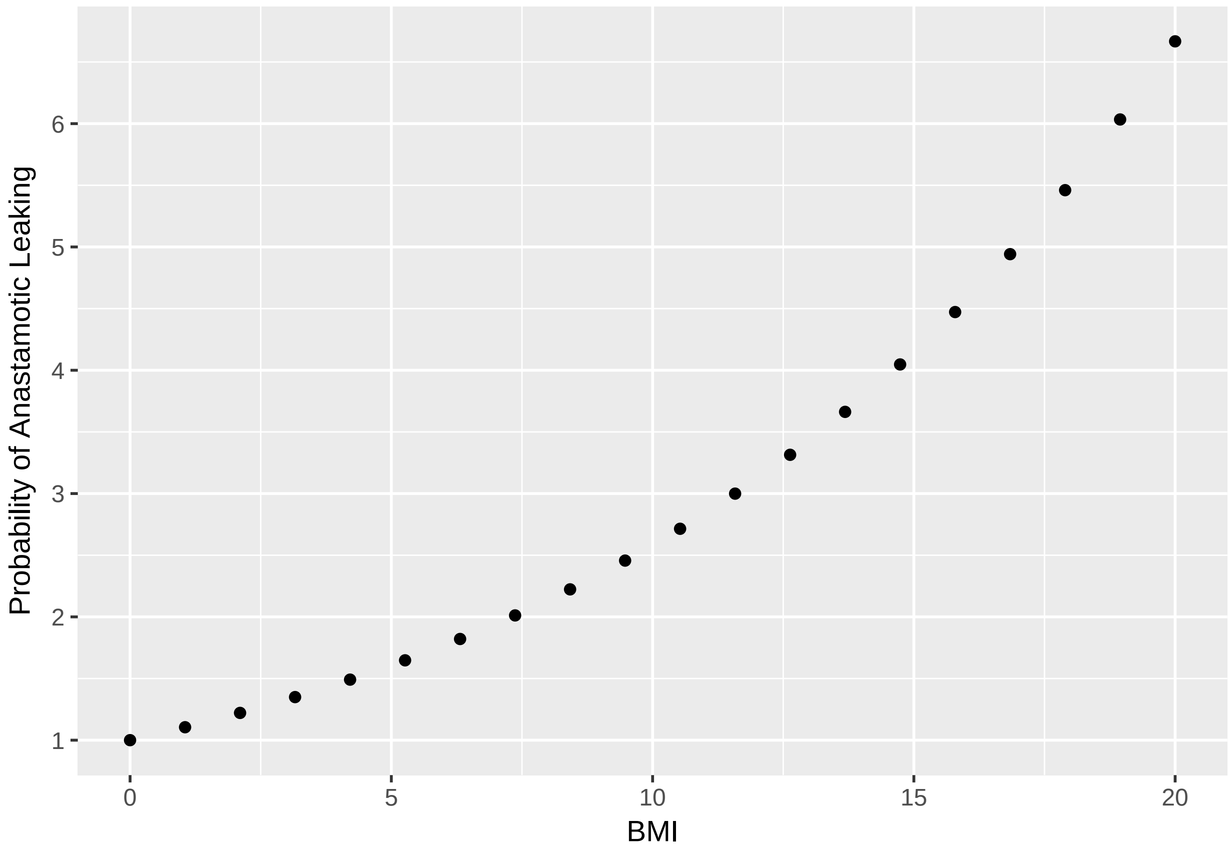
```
beta_1 = model$coefficients[3]
ste_1 = summary(model)$coefficients[3,2]

#Confidence interval
t_star = 1.973
L = exp(beta_1 - t_star*ste_1)
U = exp(beta_1 + t_star*ste_1)
CI_1 = sort(c(L,U))
print(CI_1)
```

```
##       x2       x2
## 1.031693 1.171779
```

Thus, since the confidence interval contains values only above 1, we are confident to say that a 1-unit increase does increase the risk of anastamotic leaking (by a factor $\simeq 1.0995073$. Let's see how an increase $1, 2, ..., 20$ units in BMI impact the risk of anastamotic leaking.

```
#plot
library(ggplot2)
z1 = seq(0,20,length=20)
z2 = exp(z1*beta_1)

df = data.frame(z1,z2)
```

```r
ggplot(df, aes(x=z1,y=z2))+
    geom_point()+
    labs(x="BMI", y="Probability of Anastamotic Leaking")
```



Therefore, as expected, in the context of other potential predictors, the more we increase BMI, the more the Odds Ratio resulting from that increase gets bigger. Which means the risk of anastamotic leaking gets bigger as the BMI increases and other predictors stay constant.

BMI categorical:

```r
data1$new_BMI = "healthy"
data1$new_BMI[data1$BMI>=30 & data1$BMI<35] = "overweight"
data1$new_BMI[data1$BMI>=35 & data1$BMI<40] = "severely overweight"
data1$new_BMI[data1$BMI>=40] = "obese"

x2_new = data1$new_BMI

model_new = glm(y ~ x1+x2_new+x3+x4+x5+x6+x7+x8+x9+x10, family = "binomial", data=data1)
summary(model_new)
```

```
##
## Call:
## glm(formula = y ~ x1 + x2_new + x3 + x4 + x5 + x6 + x7 + x8 +
##     x9 + x10, family = "binomial", data = data1)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5161  -0.5226  -0.3186  -0.1465   2.4529
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -4.76907    1.73555  -2.748 0.005998 **
## x1Male                       0.82533    0.53168   1.552 0.120589
## x2_newobese                  2.06802    0.79342   2.606 0.009148 **
## x2_newoverweight             0.33552    0.69193   0.485 0.627744
## x2_newseverely overweight  1.09743    0.98657   1.112 0.265981
## x3                           0.08327    0.02658   3.133 0.001733 **
## x4W                         -0.34738    0.53320  -0.651 0.514728
## x5                           0.96946    0.55800   1.737 0.082319 .
## x6                          -0.69770    0.65746  -1.061 0.288598
## x7                          -0.50950    0.72603  -0.702 0.482826
## x8                           0.64320    0.58948   1.091 0.275219
## x9                          -1.26409    0.36658  -3.448 0.000564 ***
## x10                          7.15818    3.53367   2.026 0.042794 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 148.35  on 178  degrees of freedom
## Residual deviance: 113.49  on 166  degrees of freedom
## AIC: 139.49
##
## Number of Fisher Scoring iterations: 6
```

Let's compute the Odds Ratios of Overweight vs Healthy, Severely Overweight vs Healthy and Obese vs Healthy:

```
#odds of overweight vs healthy
OR1 = exp(model_new$coefficients[4])
print(OR1)
```

```
## x2_newoverweight
##          1.39867
```

```
#odds of severely overweigth vs healthy
OR2 = exp(model_new$coefficients[5])
print(OR2)
```

```
## x2_newseverely overweight
##                  2.996454
```

```
#odds of obese vs healthy
OR3 = exp(model_new$coefficients[3])
print(OR3)
```

```
## x2_newobese
##     7.90914
```

Thus, someone overweight is about (without confidence interval) 1.36 times more likely to get an anastamotic leaking than a healthy person. Someone severely overweight is 3 times more likely to get an anastamotic leaking than a healthy person. Someone obese is 7.9 times more likely to get an anastamotic leaking than a healthy person.

Problem 2:

```
summary(model)
```

```
##
## Call:
## glm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10, family = "binomial", data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5455  -0.4899  -0.3060  -0.1282   2.4431
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.92114    2.09214  -3.308 0.000939 ***
## x1Male       0.82283    0.51895   1.586 0.112839
## x2           0.09486    0.03227   2.940 0.003282 **
## x3           0.08437    0.02646   3.188 0.001431 **
## x4W         -0.23469    0.51959  -0.452 0.651494
## x5           1.02614    0.57343   1.789 0.073541 .
## x6          -0.85182    0.64633  -1.318 0.187529
## x7          -0.68212    0.73646  -0.926 0.354334
## x8           0.59864    0.59499   1.006 0.314348
## x9          -1.36235    0.36698  -3.712 0.000205 ***
## x10          7.05757    3.60032   1.960 0.049965 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 148.35  on 178  degrees of freedom
## Residual deviance: 111.56  on 168  degrees of freedom
## AIC: 133.56
##
## Number of Fisher Scoring iterations: 6
```

If we look at the p-values in the above summary of the complete model, we can have a rough idea of which predictor we might want to drop. Indeed, x1, x4, x6, x7 and x8 have high p-values so they might be good candidates to drop. Let's run Lasso with all 10 predictors to get another overall point of view.

```
library(elasticnet)
```

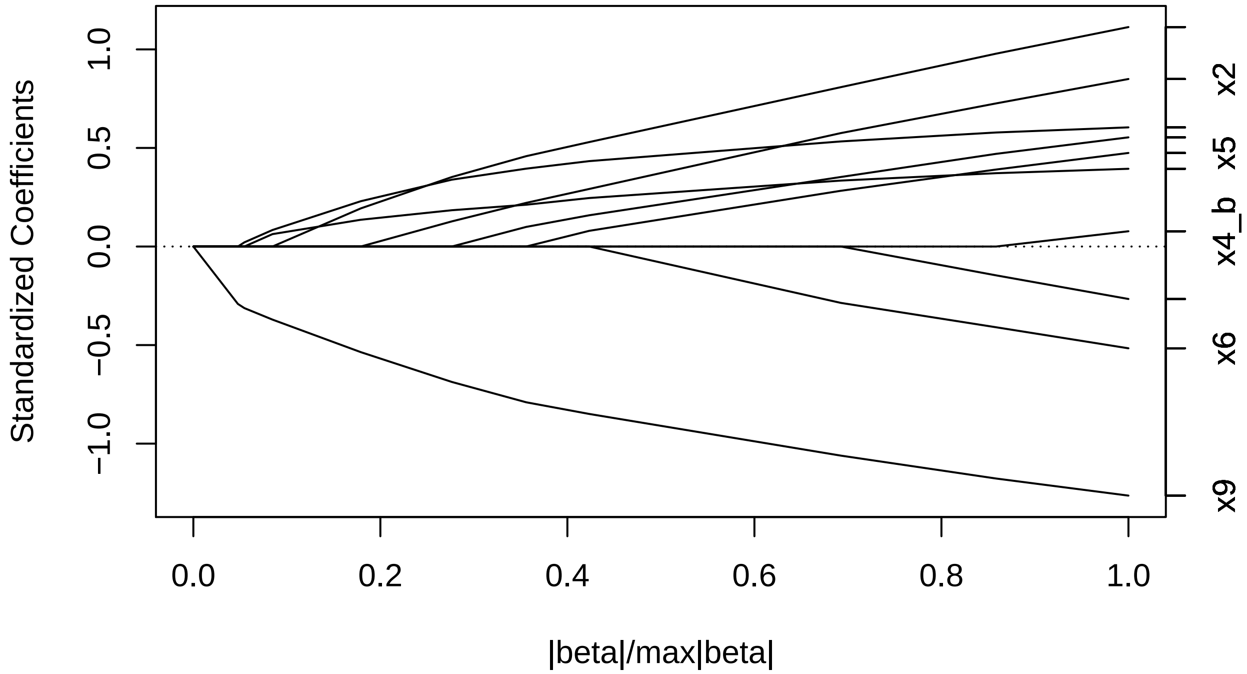
```
## Loading required package: lars

## Loaded lars 1.2
```

```
#x1 and x4 are here again trnasformed to 0's and 1's to make enet() work

X = cbind(x1_b,x2,x3,x4_b,x5,x6,x7,x8,x9,x10)

elnet = enet(x=X, y=y, lambda=0)
```

```r
print(elnet$beta.pure)
```

```
##          x1_b          x2          x3       x4_b          x5          x6
## 0  0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  0.00000000
## 1  0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  0.00000000
## 2  0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  0.00000000
## 3  0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  0.00000000
## 4  0.00000000 0.000000000 0.001112950 0.00000000 0.00000000  0.00000000
## 5  0.00000000 0.001240145 0.002028667 0.00000000 0.00000000  0.00000000
## 6  0.01486856 0.002152917 0.002638631 0.00000000 0.00000000  0.00000000
## 7  0.02369102 0.002838687 0.003049906 0.00000000 0.01189010  0.00000000
## 8  0.05272491 0.005595301 0.004653319 0.00000000 0.04227794 -0.05281681
## 9  0.07027295 0.007059025 0.005631664 0.00000000 0.05840628 -0.07561145
## 10 0.08278727 0.008261365 0.006407826 0.01161752 0.07097524 -0.09530894
##             x7          x8          x9        x10
## 0   0.00000000 0.000000000  0.00000000 0.00000000
## 1   0.00000000 0.000000000 -0.03183574 0.00000000
## 2   0.00000000 0.000000000 -0.03414992 0.02526906
## 3   0.00000000 0.009437874 -0.04052218 0.09994169
## 4   0.00000000 0.020362938 -0.05856271 0.27425638
## 5   0.00000000 0.027607588 -0.07507443 0.40457389
## 6   0.00000000 0.031807275 -0.08634652 0.47123452
## 7   0.00000000 0.036899760 -0.09281908 0.51680220
## 8   0.00000000 0.050181589 -0.11591854 0.63599187
## 9  -0.03400668 0.055801208 -0.12862245 0.69024369
## 10 -0.06182509 0.059157055 -0.13809000 0.72118404
## attr(,"scaled:scale")
##  [1]   6.6894397 102.8521575 173.7321511   6.6593069   6.6869338
##  [6]   5.4177445   4.3053818   6.6660149   9.1513653   0.8380991
```

```r
plot(elnet)
```



We can see very clearly that the predictors x4_b (Race) and x7 (CAD/PAD) go to 0 way faster than the other ones, so we are confident that we can get rid of them. The next predictors to consider dropping are x6, x1 and x5, however it is not that clear if they are significant or not so I am going to take them out of the model 1 by 1 and keep them only if the AIC of the model goes up after doing so.

```
model1 = glm(y ~ x1+x2+x3+x5+x8+x9+x10, family = "binomial", data=data1); print(model1$aic)
```

```
## [1] 130.4615
```

```
model2 = glm(y ~ x2+x3+x5+x8+x9+x10, family = "binomial", data=data1); print(model2$aic)
```

```
## [1] 130.3497
```

```
model3 = glm(y ~ x2+x3+x8+x9+x10, family = "binomial", data=data1); print(model3$aic)
```

```
## [1] 132.2513
```

So, dropping x6 does decrease AIC so we can drop it, same for x1. However dropping x5 increases AIC so we should keep it in the model.

What about x8? Lasso doesn't seem to worry about it but its p-value was pretty high, let's drop it and see if the AIC drops.

```
model4 = glm(y ~ x2+x3+x5+x9+x10, family = "binomial", data=data1); print(model4$aic)
```

```
## [1] 130.1718
```

The AIC dropped,so we drop x8 as well.

Running AIC and dropping all other predictors one by one shows that there is no other predictor to drop and thus $y \sim x_2 + x_3 + x_5 + x_9 + x_{10}$ is the best model.

As a result, the significant predictors in our model to predict the risk of anastamotic leaking are: BMI, Age, Tobacco, Albumin and Operation Length.

Now let's determine what impact (negative or positive) those predictors have on the risk on anastamotic leaking:

```
best_model = glm(y ~ x2+x3+x5+x9+x10, family = "binomial", data=data1)
summary(best_model)
```

```
##
## Call:
## glm(formula = y ~ x2 + x3 + x5 + x9 + x10, family = "binomial",
##     data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5766  -0.5841  -0.3275  -0.1598   2.3145
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.54614    1.87144  -2.964 0.003041 **
## x2           0.07171    0.02911   2.463 0.013767 *
## x3           0.07452    0.02290   3.253 0.001141 **
## x5           0.86566    0.51685   1.675 0.093961 .
## x9          -1.29694    0.34722  -3.735 0.000188 ***
## x10          8.56448    3.35520   2.553 0.010692 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 148.35  on 178  degrees of freedom
## Residual deviance: 118.17  on 173  degrees of freedom
## AIC: 130.17
##
## Number of Fisher Scoring iterations: 6
```

Here are the confidence intervals of the Odds Ratio resulting from a 1-unit increase in x_i (i =2,3,5,8,9,10). (Computed the same way I did in problem 1, so I skipped the details).

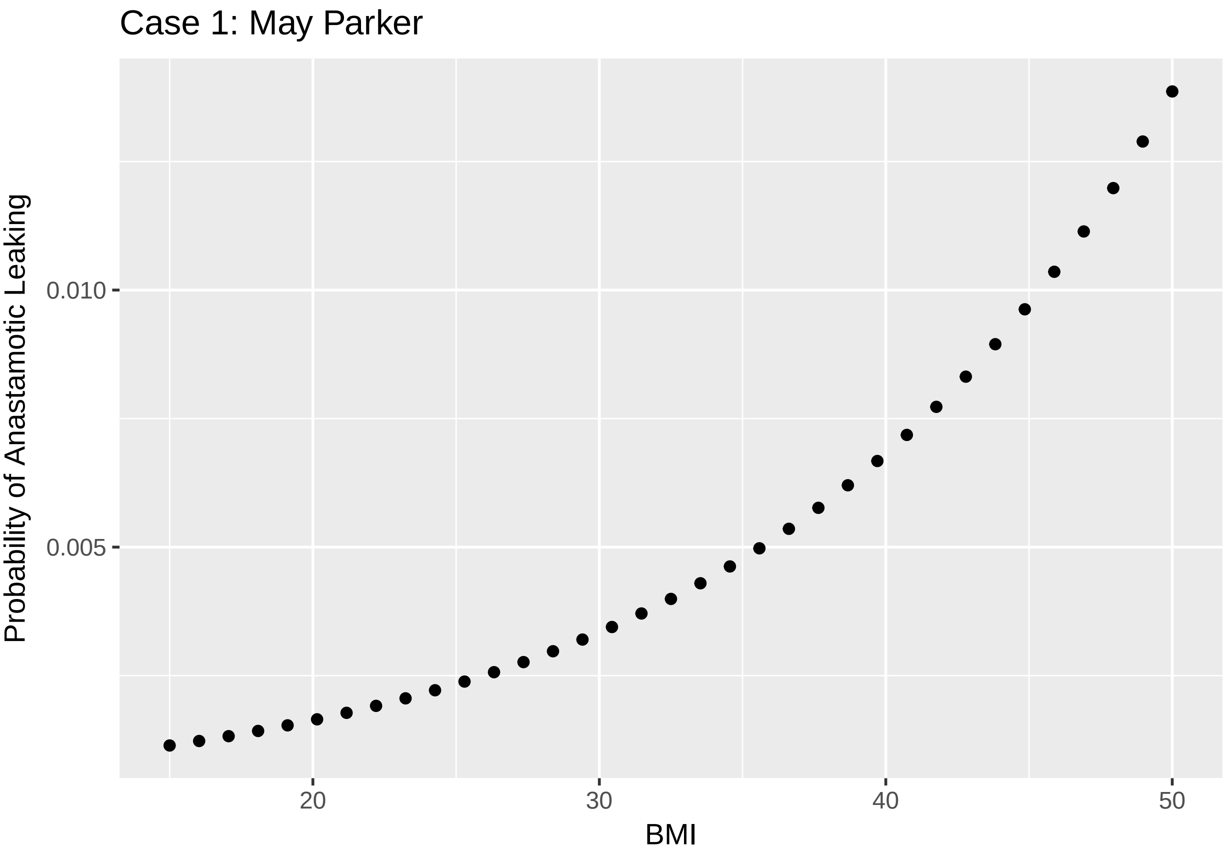
- x2: $CI_2 = [1.01, 1.14]$, an increase in BMI (still) increases the risk of anastamotic leaking.
- x3: $CI_3 = [1.02, 1.12]$, an increase in Age increases the risk of anastamotic leaking.
- x5: $CI_5 = [0.97, 8.09]$, Smoking does increases the risk of anastamotic leaking compared to non-smoking. (small overlap of 1 but very small so it's okay).
- x9: $CI_9 = [0.15, 0.59]$, an increase in Albumin decreases the risk of anastamotic leaking.
- x10: $CI_{10} = [3.94, 2.47 \cdot 10^6]$, an increase in Operation Length increases the risk of anastamotic leaking.

Note that the bigger the increase in each predictor's unit, the bigger the effect (positive or negative) on the risk of getting anastamotic leaking.

Problem 3:

Case 1: May Parker

```
library(ggplot2)

vect_bmi = seq(15,50,length=35)
y_predict1 = predict(best_model,newdata=data.frame(x2=vect_bmi,x3=35, x5=0,
                     x9=4.2, x10=0.0625), type="response")

df1 = data.frame(vect_bmi, y_predict1)
ggplot(df1, aes(x=vect_bmi,y_predict1))+
    geom_point()+
    labs(x="BMI", y="Probability of Anastamotic Leaking", title="Case 1: May Parker")
```

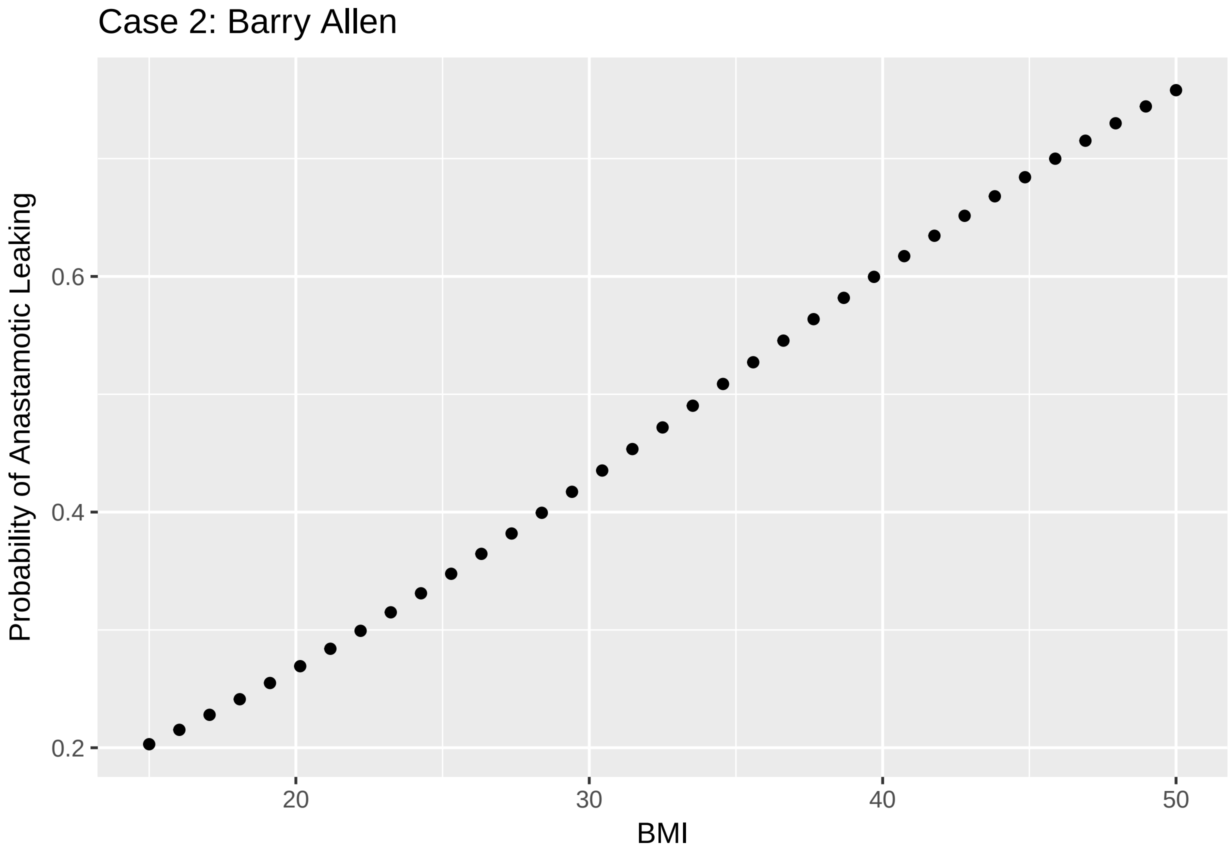


So, we can see that a young, non-smoker person such as May Parker that had a relatively short operation with a good Albumin level is very unlikely to get an anastamotic leaking, even though her BMI is really high. We can still note though that this probability can be multiplied by 4 if May Parker was Obese (BMI>40) instread of being Healthy (BMI<30), but stays relatively low (less than 2%).

Case 2: Barry Allen

```
y_predict2 = predict(best_model,newdata=data.frame(x2=vect_bmi,x3=62, x5=1,
                                x9=2.8, x10=0.1458), type="response")
```

```
df2 = data.frame(vect_bmi, y_predict2)
ggplot(df2, aes(x=vect_bmi,y_predict2))+
    geom_point()+
    labs(x="BMI", y="Probability of Anastamotic Leaking", title="Case 2: Barry Allen")
```



We can clearly observe that a smoker, old and with a long operation and a low Albumin level is way more likely to get an anastamotic leaking than someone like May Parker. Indeed, even with an healthy BMI, this person would still have between 20% and 30% risk of having an anastamotic leaking.