Logistic Regression

Vazgen Tadevosyan

February 21, 2019

For this Homework, you are required to submit both Markdown and HTML files with your answers and code in it. Be sure that the .Rmd file is working, so when I run it, there would be no errors and represent the same information as HTML. Write your code and interpretations under each question. The interpretations of the results need to be written below or above all the charts, summaries, or tables. Do not remove problems from your Markdown file.

Use biodata.csv dataset uploaded on Moodle to analyze the relationship between the presence of kidney disease and different factors. The description of the variables is given in a separate file. Pay close attention to the names of axes, titles, and labels.

Problem 1. 2 pt.

Load the file.

Get rid of variables which are irrelevant for logistic regression analysis using function select(). Also, skip variables which are absent in .txt file).

Check whether the data types are correct, if not make appropriate corrections assigning labels to each level according to the data description.

For all **numeric** variables replace missing values by column mean.

Create new dataset without missing data (remove observations with missing values for **categorical** variables)

How many variables and observations do you have before and now?

```r
library(ggplot2)
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
```

```r
library(tidyr)
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(ggcorrplot)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##
##     combine
```

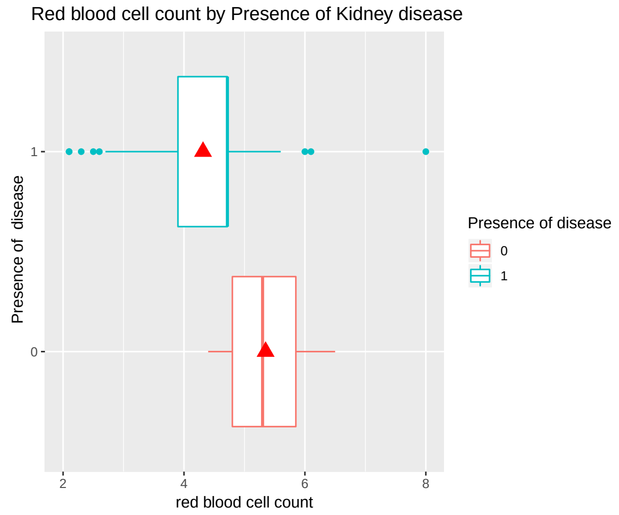
```
df<-read.csv("biodata.csv",na.strings = "?")
df<-df %>% select(-c(sc, pcv,pe,id))
df$su<-ordered(df$su,levels=c(0,1,2,3,4,5))

df$class<-factor(df$class)
sapply(df, function(x) sum(is.na(x)))
df[duplicated(df$id),][1]
sapply(df, function(x) sum(is.na(x)))
df<-df%>% mutate_if(is.numeric , na.aggregate)
df<-na.omit(df)
```

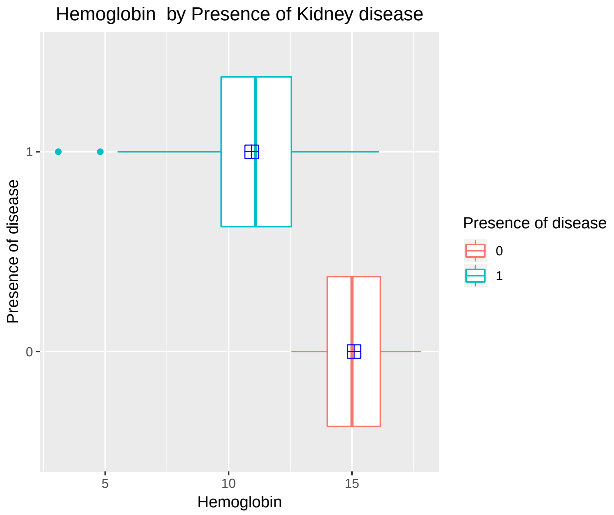
Initially there were 381 rows and 21 columns in our dataset after data cleaning process we have 322 and 17 columns.

Problem 2. 3 pt.

a. Check the relationship between each numeric variable and the presence of kidney disease. Save only the two most important numeric variables using boxplots. Comment on it.
b. Use the glm() function to perform a logistic regression with Class as the response and one of **numeric** variables as the predictor (use results of 2a). Use the summary() function to print the result. Is your explanatory variable significant? Why?
c. Plot the response and the predictor, and sigmoid line. For this task, you should create a sequence of x values and predict your model for them, then add to your graph

```
g1=ggplot(data = df, aes(x = class, y = rbcc,color = class))+
  geom_boxplot()+
  scale_x_discrete(labels= levels(df$class))+scale_y_continuous(limits=c(2, 8))+
  stat_summary(fun.y = "mean",geom = 'point',col='red', shape=17, size=4)+
  labs(title =" Red blood cell count by Presence of Kidney disease" ,y='red blood cell count',x=" Preser
  theme(plot.title = element_text(hjust = 0.5))+coord_flip()
g2 = ggplot(data = df, aes(x = class, y = hemo,color = class))+
  geom_boxplot()+
  scale_x_discrete(labels= levels(df$class) )+
  stat_summary(fun.y = "mean",geom = 'point',col='blue', shape=12, size=4)+
  labs(title ="Hemoglobin  by Presence of Kidney disease" ,color =  "Presence of disease",y='Hemoglobin
grid.arrange(g1, g2,nrow=1)
```





People who has the Kidney disease tend to have lower red blood cell counts and Hemoglobin than people who does not have illness.

```
model<-glm(class~hemo,data = df,family = 'binomial')
summary(model)
```

```
##
## Call:
## glm(formula = class ~ hemo, family = "binomial", data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.83666  -0.31947   0.01972   0.26640   2.91560
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  21.6504     2.5627   8.448   <2e-16 ***
## hemo         -1.6078     0.1904  -8.446   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 440.36  on 321  degrees of freedom
## Residual deviance: 166.08  on 320  degrees of freedom
## AIC: 170.08
##
## Number of Fisher Scoring iterations: 7
```

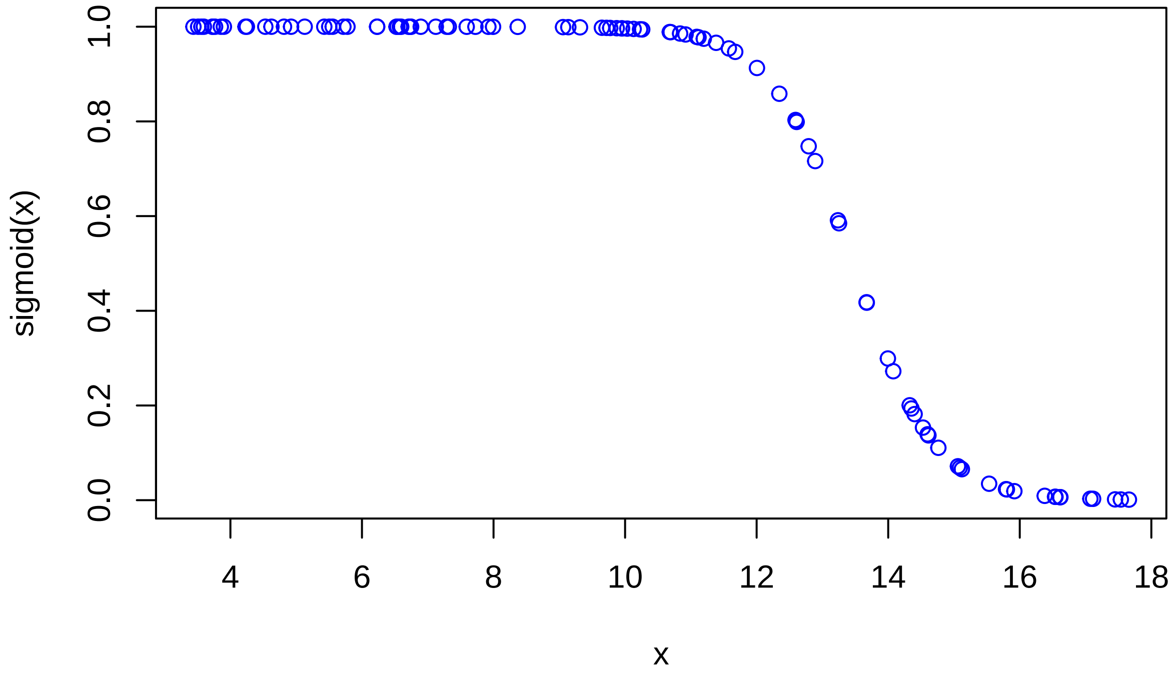
```
exp(coef(model))
```

```
##  (Intercept)         hemo
## 2.527197e+09 2.003181e-01
```

We set Confidence level=95% so alpha is equal to 0.05. Our H0 is coefficient is equal to 0 H1 it is not equal. P-value is less than alpha so we reject H0 and claim that variable hemoglobin is significant and one unit increase in it decreases the odds to have disease by (1- 2.003181e-01)*100 percent.

```
sigmoid <- function (x) 1 / (1 + exp(-21.6504+1.6078 * x))
x<-runif(min = 3.1,max = 17.8,n=100)
```

```
plot(x, sigmoid(x), col='blue')
```

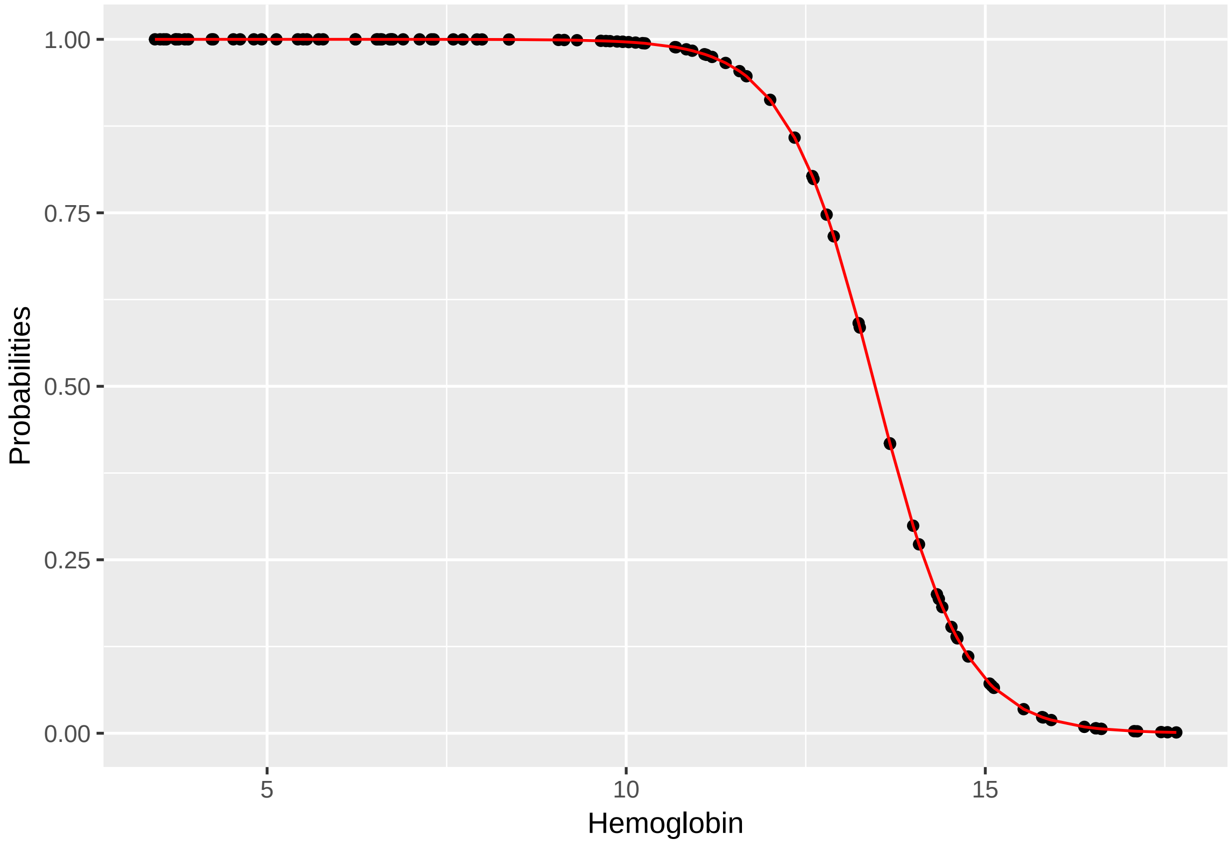


```
probs<-predict(model,data.frame(hemo=x),type='response')
predict.glm(model,data.frame(hemo=x))
```

```
##          1          2          3          4          5          6
##  5.6455417 15.3784127 15.5932095  2.8799583 -4.9482525 10.8015403
##          7          8          9         10         11         12
## 11.0443297  1.3785301 -0.8518043 11.0779761  3.6485282 12.3574759
##         13         14         15         16         17         18
## -4.6849581  9.2298806  4.4568719 -3.7609324 12.7193516  3.0359234
##         19         20         21         22         23         24
##  1.8017667 10.8259147  1.3983649 11.6315072 -5.8677998  4.2271858
##         25         26         27         28         29         30
##  9.4429715 -0.3316499 10.8625938  6.0251341 -6.4045773 13.4031680
##         31         32         33         34         35         36
## -2.0851833  0.3680668 -5.0682129 11.1395455 -1.3847387 11.1588123
##         37         38         39         40         41         42
## 10.1987058 -1.7096738  6.6741278 -5.0579695  4.0935743 -3.7392006
##         43         44         45         46         47         48
## 15.9286080 -3.9433252 -1.8425756  9.8805185 -1.4273451 10.5659770
##         49         50         51         52         53         54
##  0.9252972 15.6598561 -6.5454835  5.6544112 13.7357006  1.4048523
##         55         56         57         58         59         60
##  8.9130470  3.7767029  0.3419647 14.8502610  5.2062815  5.9355988
##         61         62         63         64         65         66
## 12.9243441  2.3485657 -0.9834149  9.9326846 -2.6599156 12.4566555
##         67         68         69         70         71         72
##  5.5186684 16.0072537  7.0885670  5.7764199 15.4536897 -2.5634000
##         73         74         75         76         77         78
## 13.9088392 11.6331532 10.8267740 -1.8231712 14.8161972  6.9593488
##         79         80         81         82         83         84
##  3.3463528 -0.3352807 16.1227328  5.3631423  8.7987090 -4.9369389
##         85         86         87         88         89         90
## 14.2160247 -5.8005641 -1.5048036 15.6493084 12.8035244 14.3664144
##         91         92         93         94         95         96
## -3.3262455  5.1526782 -6.7434872  3.8200421  8.1964528 -2.6099821
##         97         98         99        100
##  4.4795002  6.1387595  1.0846917 15.8675956
```

```
new<-data.frame("Hemoglobin"=x,"Probabilities"=probs)
ggplot(new,aes(Hemoglobin,Probabilities))+geom_point()+geom_line(fun = function(x) 1/(1+exp(21.6504+1.6
```

```
## Warning: Ignoring unknown parameters: fun
```



So if state threshold 0.5 we can say that when a people hemoglobin less than 14 they will be predicted to have disease.

Problem 3. 4 pt.

Use the glm() function to perform a logistic regression with Class as the response and one of **categorical** variables as the predictor (chose significant one). Use the summary() function to print the results.

a. Interpret the coefficients of the explanatory variable in terms of absolute and exponential values?
b. Compute probability for the base value of your explanatory variable. Comment on it.

c. What do Null deviance and residual deviance of summary output mean?
d. Calculate the value of the exponent of the b1 coefficient using only your data and functions addmargins() and table().

```
model<-glm(class~rbc,data = df,family = "binomial")
summary(model)
```

```
##
## Call:
## glm(formula = class ~ rbc, family = "binomial", data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0963  -1.0834   0.4854   1.2744   1.2744
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.0794     0.3354   6.200 5.65e-10 ***
## rbcnormal    -2.3045     0.3605  -6.393 1.63e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 440.36  on 321  degrees of freedom
## Residual deviance: 381.49  on 320  degrees of freedom
## AIC: 385.49
##
## Number of Fisher Scoring iterations: 4
```

```
probabilityOfbase<-predict(model,newdata = data.frame(rbc='abnormal'),type = 'response')
print(probabilityOfbase)
```

```
##         1
## 0.8888889
```

```
addmargins(table(df$rbc,df$class))
```

```
##
##              0   1 Sum
##   abnormal  10  80  90
##   normal   129 103 232
##   Sum      139 183 322
```

```
p1gn<-103/232
p1ga<-80/90
oddsnormal<-p1gn/(1-p1gn)
oddsabnormal<-p1ga/(1-p1ga)

rat<-oddsnormal/oddsabnormal
rat
```

```
## [1] 0.0998062
```

```
print(rat)
```

```
## [1] 0.0998062
```

```r
exp(coef(model))
```

```
## (Intercept)   rbcnormal
##   8.0000000   0.0998062
```

Probability of having disease given that a person has abnormal red blood cell is 0.8888889 consequently odds of it is equal to 0.8888889/(1-0.8888889)= 8. People who' red blood cells are normal have 91% less odds to have disease compared to people have abnormal cells.

For our example, we have a value of 440.35 on 321 degrees of freedom. Including the independent variables rbc decreased the deviance to 381.49 on 320 degrees of freedom, a significant reduction in deviance. The Residual Deviance has reduced approximately by 60 with a loss of two degrees of freedom.

Problem 4. 4 pt.

a.Use the full data set to perform the model with Class as a dependent variable. Use the stepAIC() function to obtain the best model based on AIC criteria. Use the backward selection procedure. Do not show the output.

b. Remove all non-significant variables from the last model. Show only the best model(all variables must be significant at least at the level 0.01). Use the summary() function to print the result.
c. Pick your own new observation and predict the y value. Comment on it.
d. Is it possible to calculate R square for logistic regression?

```r
summary(model_AIC)
```

```
##
## Call:
## glm(formula = class ~ age + bp + rbc + bgr + hemo + rbcc + htn +
##     cad, family = "binomial", data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.90602  -0.04205   0.00000   0.00003   2.96198
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  28.92773    7.39267   3.913 9.11e-05 ***
## age          -0.06148    0.02508  -2.451 0.014231 *
## bp            0.08240    0.04225   1.950 0.051141 .
## rbcnormal    -5.13429    1.40433  -3.656 0.000256 ***
## bgr           0.05760    0.01367   4.213 2.52e-05 ***
## hemo         -1.93304    0.42164  -4.585 4.55e-06 ***
## rbcc         -1.94837    0.89069  -2.187 0.028708 *
## htnyes       21.09955 1700.38908   0.012 0.990100
## cadyes       -5.63840    2.13268  -2.644 0.008198 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 440.356  on 321  degrees of freedom
## Residual deviance:  55.581  on 313  degrees of freedom
## AIC: 73.581
##
## Number of Fisher Scoring iterations: 20
```

```r
final<-glm(class ~rbc + bgr + hemo,data=df,family = 'binomial')
summary(final)
```

```
##
## Call:
## glm(formula = class ~ rbc + bgr + hemo, family = "binomial",
##     data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.04446  -0.13753   0.00051   0.05026   2.45960
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 24.199549   4.087032   5.921 3.20e-09 ***
## rbcnormal   -3.758434   0.847104  -4.437 9.13e-06 ***
## bgr          0.043263   0.009165   4.720 2.36e-06 ***
## hemo        -1.970961   0.299331  -6.585 4.56e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 440.356  on 321  degrees of freedom
## Residual deviance:  97.913  on 318  degrees of freedom
## AIC: 105.91
##
## Number of Fisher Scoring iterations: 8
```

```r
predict(final,newdata = data.frame(bp=mean(df$bp),rbc='normal',bgr=mean(df$bgr),hemo=mean(df$hemo),rbcc
```

```
##       1
## 0.84106
```

```r
nullmod <- glm(class~1,data = df, family="binomial")
R2McFadden<-1-logLik(final)/logLik(nullmod)
R2McFadden
```

```
## 'log Lik.' 0.7776502 (df=4)
```

About 84 percent we are sure that our new observation has disease, and as out treshhold is 0.5 , we classify it as 1.

For logistic regression McFadden's pseudo-R squared is used: R2McFadden=1-log(Lc)/log(Lnull) where Lc denotes the (maximized) likelihood value from the current fitted model, and Lnull denotes the corresponding value but for the null model - the model with only an intercept and no covariates. So, when our model is predicting all of the variation in the outcome,lc will be 1 so log(Lc) will be 0,R2McFadden=1. This means, closer value of R2McFadden to 1 indicates of better model. In our case it is 0.7776502. So model explains variation by 77%.

Problem 5. 7 pt.

a. Divide the data frame into Train and Test sets (70:30), such that the proportion for training and testing sets must refer to the proportion of whole data. Do not forget about the set.seed() function.
b. Now fit two logistic regression models using training data. Both models should be the result of Problem 4a. and Problem 4b.
c. For the first model (which contains only significant coefficients) predict the probability of the presence

of chronic kidney disease for testing set. Compute the confusion matrix using table() function. Figure out the overall fraction of correct predictions, sensitivity, and specificity for the held out data using only confusion matrix. Check your computations using the function confusionMatrix().

d. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.
e. What is the difference between the ROC and the precision-recall curve (PRC)? When do we need to consider PRC? Why? Plot the PRC if it is applicable to your models?
f. Plot ROC curve for both models using the function ggroc {pROC}. Which models is the best? Why?

```
library(caret)
```

```
## Loading required package: lattice
```

```
set.seed(1)
train_index<-createDataPartition(df$class,p = 0.7,list = F)
train<-df[train_index,]
test<-df[-train_index,]
table(train$class)/sum(table(train$class))
```

```
##
##         0         1
## 0.4317181 0.5682819
```

```
table(test$class)/sum(table(test$class))
```

```
##
##         0         1
## 0.4315789 0.5684211
```

```
#in this case our benchmark is 56 percent accuracy.
model<-glm(final$formula,data = train,family = 'binomial')
modelAIC<-glm(model_AIC$formula,data = train,family = 'binomial')
```

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

```
pred<-predict(model,newdata = test,type = 'response')
pred_class<-ifelse(pred>0.5,1,0)
conf<-addmargins(table(test$class,pred_class))
Accuracy<-sum(c(conf[1,1],conf[2,2]))/conf[3,3]
sensitivity<-50/54
specificity<-39/41
print(paste("Overall accuracy is",Accuracy,"Sensitivity is",sensitivity,"and specificity" ,specificity))
```

```
## [1] "Overall accuracy is 0.936842105263158 Sensitivity is 0.925925925925926 and specificity 0.9512195
```

```
confusionMatrix(as.factor(pred_class),test$class,positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 39  4
##          1  2 50
##
##                Accuracy : 0.9368
##                  95% CI : (0.8676, 0.9765)
##     No Information Rate : 0.5684
##     P-Value [Acc > NIR] : 9.001e-16
##
##                   Kappa : 0.872
##  Mcnemar's Test P-Value : 0.6831
##
##             Sensitivity : 0.9259
##             Specificity : 0.9512
##          Pos Pred Value : 0.9615
##          Neg Pred Value : 0.9070
##              Prevalence : 0.5684
##          Detection Rate : 0.5263
##    Detection Prevalence : 0.5474
##       Balanced Accuracy : 0.9386
##
##        'Positive' Class : 1
##
```

Overall model accuracy is 0.936842105263158 , which is very good if we take the portion of classes into account (43% 57%).We are interested in predicting disease so we should pay attention to sensitivity which measures the proportion of actual positives that are correctly identified as such.In our case it is 0.9259 which is better than benchmark (No Information Rate : 0.5684). Overall model is significant P-Value [Acc > NIR] : <2e-16. Classifiers have a similar proportion of errors on the test set. McNamara's Test P-Value : 0.6831

e. What is the difference between the ROC and the precision-recall curve (PRC)? When do we need to consider PRC? Why? Plot the PRC if it is applicable to your models?
f. Plot ROC curve for both models using the function ggroc {pROC}. Which models is the best? Why?

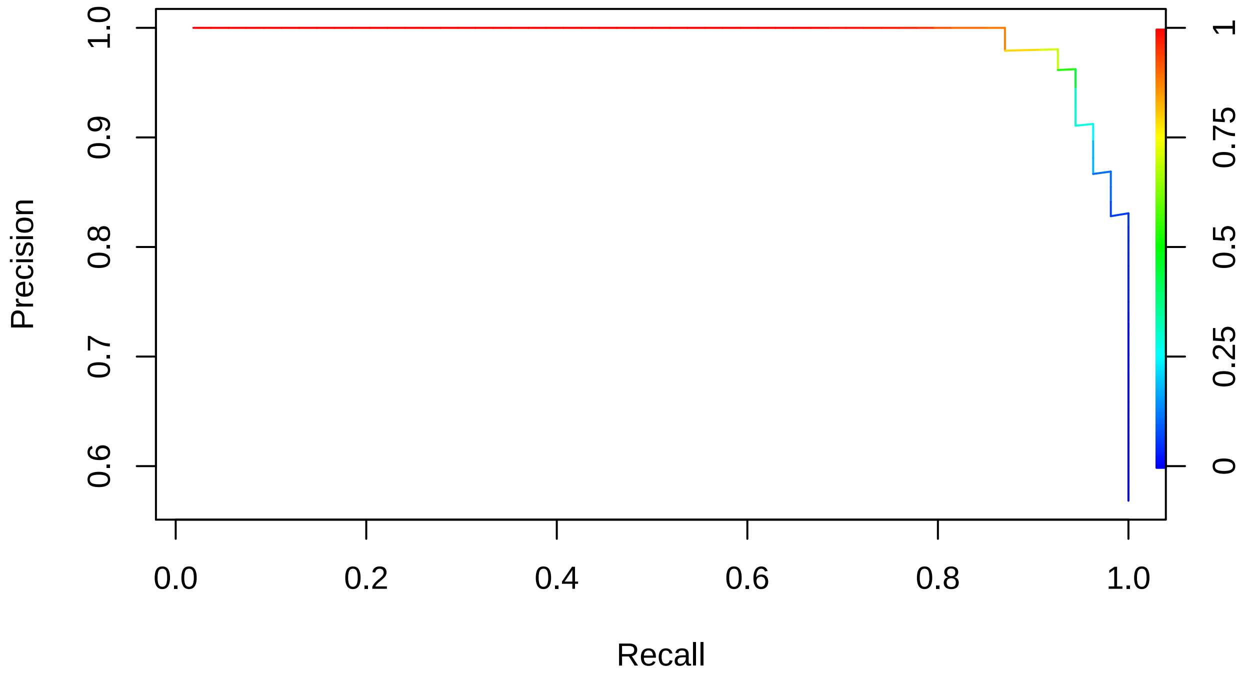
```r
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
p_test<-prediction(pred,test$class)
perf<-performance(p_test,"prec","rec")
plot(perf,colorize=T)
```



```r
exp(-0.6)
```

```
## [1] 0.5488116
```

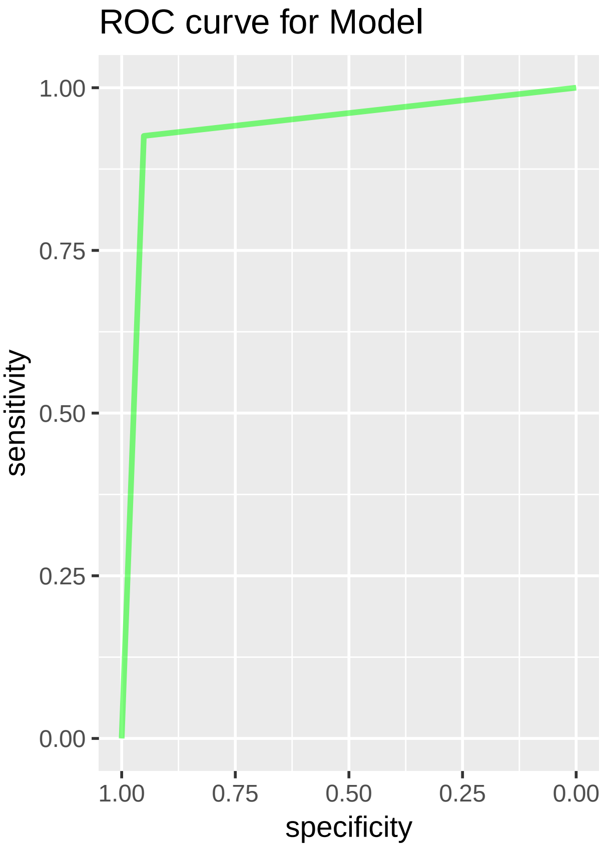
```r
library(pROC)
```

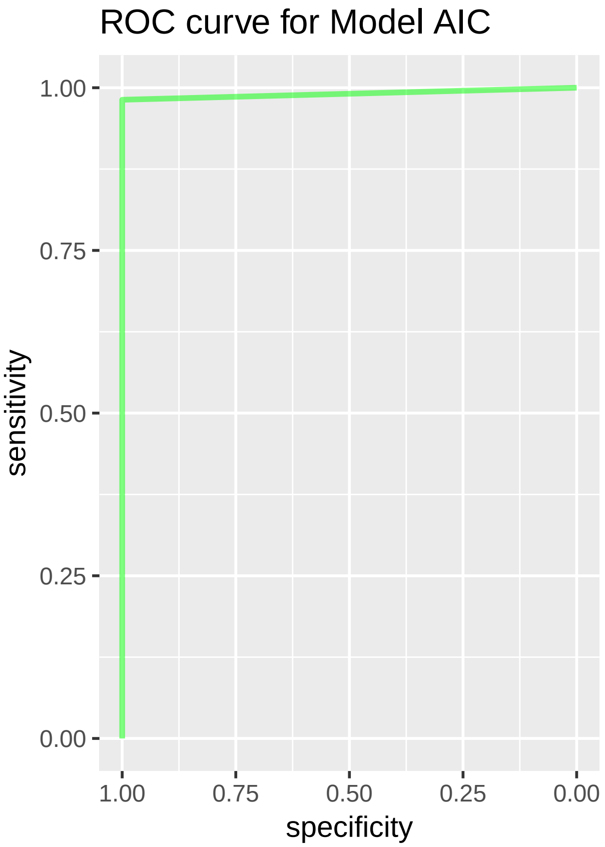
```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
rrr<-roc(test$class,pred_class)
g1<-ggroc(rrr,alpha = 0.5, colour = "green", linetype = 1, size = 1)+ggtitle("ROC curve for Model")



pred1<-predict(model_AIC,newdata = test,type = 'response')
pred_class1<-ifelse(pred1>0.5,1,0)
rrr1<-roc(test$class,pred_class1)
g2<-ggroc(rrr1,alpha = 0.5, colour = "green", linetype = 1, size = 1)+ggtitle("ROC curve for Model AIC"

grid.arrange(g1, g2,nrow=1)
```





ROC curves should be used when there are roughly equal numbers of observations for each class. Precision-Recall curves can be used when there is inbalanced data. The reason is ROC curves present an optimistic picture of the model on datasets with an imbalanced class .If the proportion of positive to negative instances changes in a test set, the ROC curves will not change. For this problem we can use Precision-recall curve] plots, it provides the viewer with an accurate prediction of future classification performance due to the fact that they evaluate the fraction of true positives among positive predictions.In our case our data is balanced as proportion of classes is following(43% and 57%).However from the plot we can see that Area under Curve is large (close to 1) so overall model is predicting very well. Regarding ROC curves we can state that both models are acceptable as the AUC are bigger than benchmark 0.5.However as we see the provided by Step AIC performs better as AUC is bigger than the model we defined.