EXERCISE 10.2

Jahedur Rahman

2/15/2022

##1a

For this problem, you will be working with the thoracic surgery data set from the University of California Irvine machine learning repository. This dataset contains information on life expectancy in lung cancer patients after surgery. The underlying thoracic surgery data is in ARFF format. This is a text-based format with information on each of the attributes. You can load this data using a package such as foreign or by cutting and pasting the data section into a CSV file.

```
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
```

```
library(stats)
library(ggplot2)
library(caTools)
# Load ThoraricSurgery.csv
thoracic_surgery_df=read.csv('ThoraricSurgery.csv')
```

1bi

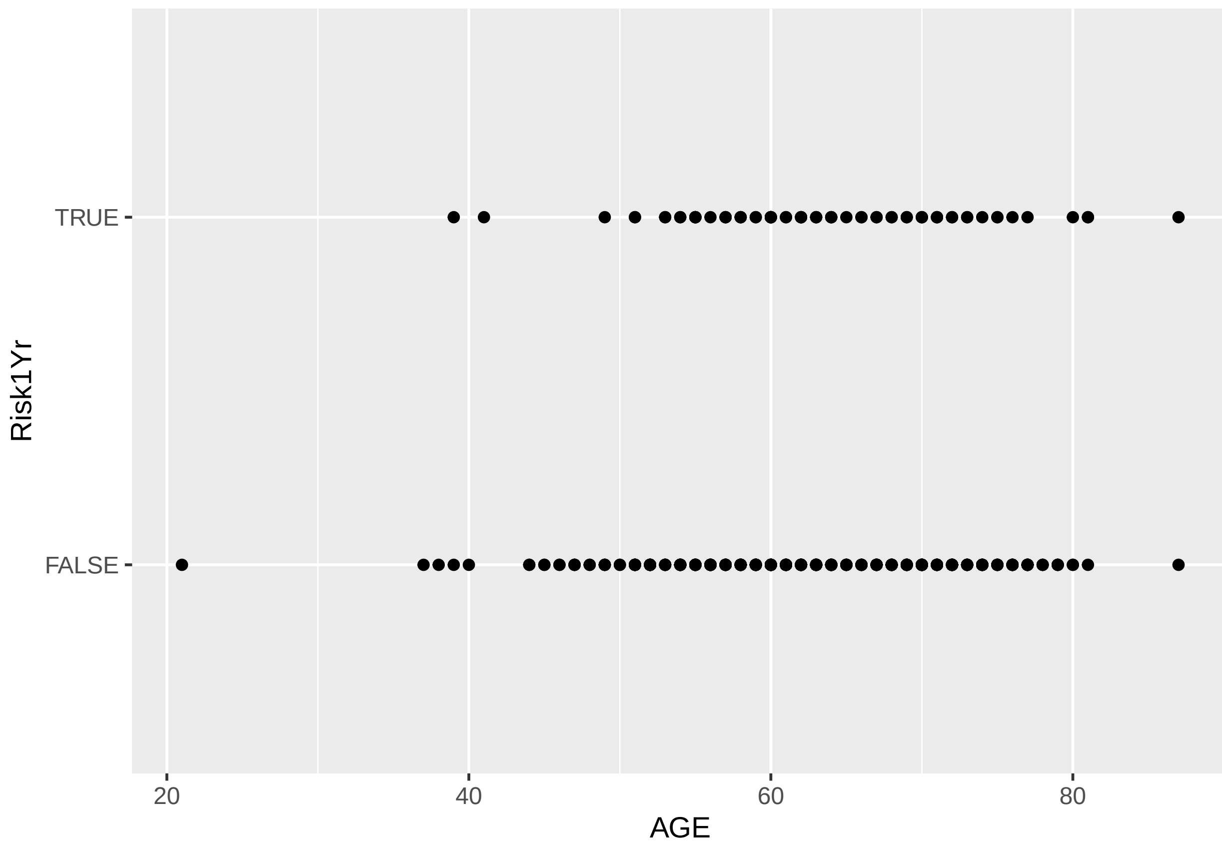
Fit a binary logistic regression model to the data set that predicts whether or not the patient survived for one year (the Risk1Y variable) after the surgery. Use the glm() function to perform the logistic regression. See Generalized Linear Models for an example. Include a summary using the summary() function in your results.

```
head(thoracic_surgery_df)
```

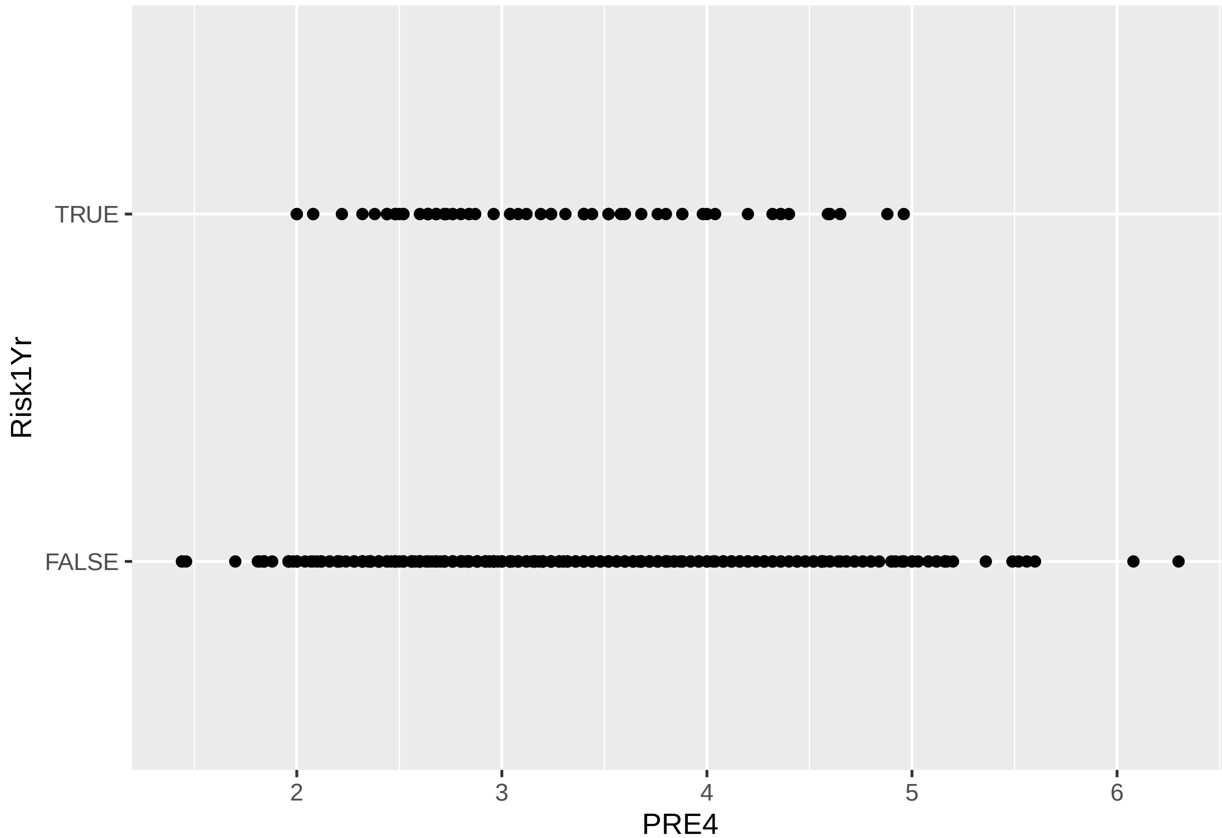
```
##   id  DGN PRE4 PRE5 PRE6  PRE7  PRE8  PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25
## 1  1 DGN2 2.88 2.16 PRZ1 FALSE FALSE FALSE  TRUE  TRUE  OC14 FALSE FALSE FALSE
## 2  2 DGN3 3.40 1.88 PRZ0 FALSE FALSE FALSE FALSE FALSE  OC12 FALSE FALSE FALSE
```

```
## 3  3 DGN3 2.76 2.08 PRZ1 FALSE FALSE FALSE  TRUE FALSE  OC11 FALSE FALSE FALSE
## 4  4 DGN3 3.68 3.04 PRZ0 FALSE FALSE FALSE FALSE FALSE  OC11 FALSE FALSE FALSE
## 5  5 DGN3 2.44 0.96 PRZ2 FALSE  TRUE FALSE  TRUE  TRUE  OC11 FALSE FALSE FALSE
## 6  6 DGN3 2.48 1.88 PRZ1 FALSE FALSE FALSE  TRUE FALSE  OC11 FALSE FALSE FALSE
##   PRE30 PRE32 AGE Risk1Yr
## 1  TRUE FALSE  60   FALSE
## 2  TRUE FALSE  51   FALSE
## 3  TRUE FALSE  59   FALSE
## 4 FALSE FALSE  54   FALSE
## 5  TRUE FALSE  73    TRUE
## 6 FALSE FALSE  51   FALSE
```

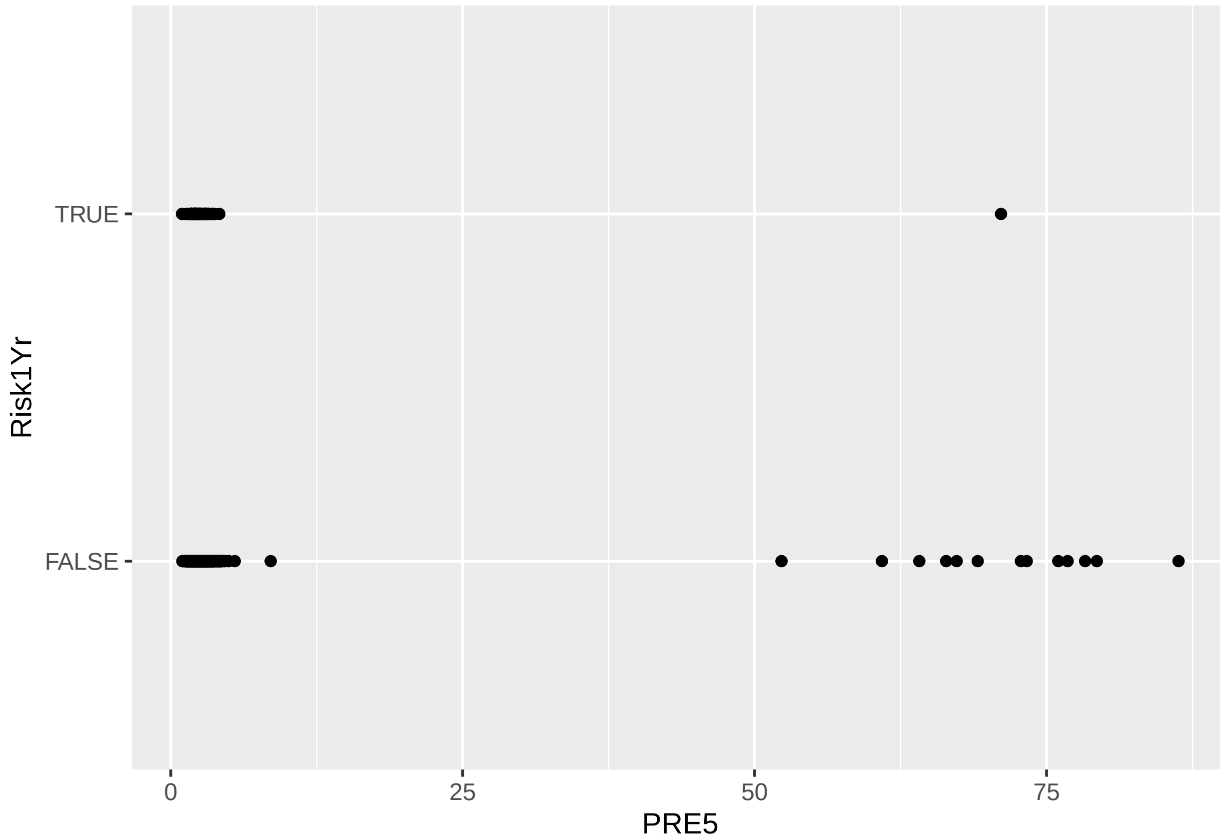
```
ggplot(thoracic_surgery_df, aes(AGE, Risk1Yr)) + geom_point()
```




```
ggplot(thoracic_surgery_df, aes(PRE4, Risk1Yr)) + geom_point()
```




```
ggplot(thoracic_surgery_df, aes(PRE5, Risk1Yr)) + geom_point()
```




```r
# In PRE5 almost all values over 50 have a FALSE Risk1Yr
thoracic_surgery_df$PRE5_50<-as.numeric(thoracic_surgery_df$PRE5 >= 50)
View(thoracic_surgery_df)

# Use the glm() function to perform the logistic regression.
thoracic_surgery_glm <- glm(Risk1Yr ~ PRE5_50 + PRE6 + PRE9 + PRE17 + PRE30, data = thoracic_surgery_df

# Include a summary using the summary() function in your results.
summary(thoracic_surgery_glm)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ PRE5_50 + PRE6 + PRE9 + PRE17 + PRE30,
##     family = binomial(), data = thoracic_surgery_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1780  -0.5502  -0.5502  -0.3738   2.3933
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.8052     0.4642  -6.043 1.51e-09 ***
## PRE5_50      -1.1283     1.0947  -1.031  0.30269
## PRE6PRZ1      0.1791     0.3344   0.536  0.59221
## PRE6PRZ2      0.8030     0.5426   1.480  0.13887
## PRE9TRUE      1.1889     0.4529   2.625  0.00866 **
```

```
## PRE17TRUE     1.0583     0.4100   2.581  0.00985 **
## PRE30TRUE     0.8148     0.4352   1.872  0.06116 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 395.61  on 469  degrees of freedom
## Residual deviance: 376.80  on 463  degrees of freedom
## AIC: 390.8
##
## Number of Fisher Scoring iterations: 5
```

1bii

According to the summary, which variables had the greatest effect on the survival rate?

> The variables that had the greatest effect on the survival rate are PRE9 and PRE17. In addition, both of the variables had p values of about 0.01, so they are significant.

1biii

To compute the accuracy of your model, use the dataset to predict the outcome variable. The percent of correct predictions is the accuracy of your model. What is the accuracy of your model?

```
# Add a column for the probability of Risk1Yr based on the model
thoracic_surgery_df$probability<-fitted(thoracic_surgery_glm)
head(thoracic_surgery_df)
```

```
##   id  DGN PRE4 PRE5 PRE6  PRE7  PRE8  PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25
## 1  1 DGN2 2.88 2.16 PRZ1 FALSE FALSE FALSE  TRUE  TRUE  OC14 FALSE FALSE FALSE
## 2  2 DGN3 3.40 1.88 PRZ0 FALSE FALSE FALSE FALSE FALSE  OC12 FALSE FALSE FALSE
## 3  3 DGN3 2.76 2.08 PRZ1 FALSE FALSE FALSE  TRUE FALSE  OC11 FALSE FALSE FALSE
## 4  4 DGN3 3.68 3.04 PRZ0 FALSE FALSE FALSE FALSE FALSE  OC11 FALSE FALSE FALSE
## 5  5 DGN3 2.44 0.96 PRZ2 FALSE  TRUE FALSE  TRUE  TRUE  OC11 FALSE FALSE FALSE
## 6  6 DGN3 2.48 1.88 PRZ1 FALSE FALSE FALSE  TRUE FALSE  OC11 FALSE FALSE FALSE
##   PRE30 PRE32 AGE Risk1Yr PRE5_50 probability
## 1  TRUE FALSE  60   FALSE       0  0.14047903
## 2  TRUE FALSE  51   FALSE       0  0.12020985
## 3  TRUE FALSE  59   FALSE       0  0.14047903
## 4 FALSE FALSE  54   FALSE       0  0.05704306
## 5  TRUE FALSE  73    TRUE       0  0.23371271
## 6 FALSE FALSE  51   FALSE       0  0.06747828
```

```
# Add a column for T and F for predictions based on the probability above 0.25
thoracic_surgery_df$probability_TF<-if_else(thoracic_surgery_df$probability > .25, T, F)
head(thoracic_surgery_df)
```

```
##   id  DGN PRE4 PRE5 PRE6  PRE7  PRE8  PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25
## 1  1 DGN2 2.88 2.16 PRZ1 FALSE FALSE FALSE  TRUE  TRUE  OC14 FALSE FALSE FALSE
## 2  2 DGN3 3.40 1.88 PRZ0 FALSE FALSE FALSE FALSE FALSE  OC12 FALSE FALSE FALSE
```

```
## 3  3 DGN3 2.76 2.08 PRZ1 FALSE FALSE FALSE  TRUE FALSE  OC11 FALSE FALSE FALSE
## 4  4 DGN3 3.68 3.04 PRZ0 FALSE FALSE FALSE FALSE FALSE  OC11 FALSE FALSE FALSE
## 5  5 DGN3 2.44 0.96 PRZ2 FALSE  TRUE FALSE  TRUE  TRUE  OC11 FALSE FALSE FALSE
## 6  6 DGN3 2.48 1.88 PRZ1 FALSE FALSE FALSE  TRUE FALSE  OC11 FALSE FALSE FALSE
##   PRE30 PRE32 AGE Risk1Yr PRE5_50 probability probability_TF
## 1  TRUE FALSE  60   FALSE       0  0.14047903          FALSE
## 2  TRUE FALSE  51   FALSE       0  0.12020985          FALSE
## 3  TRUE FALSE  59   FALSE       0  0.14047903          FALSE
## 4 FALSE FALSE  54   FALSE       0  0.05704306          FALSE
## 5  TRUE FALSE  73    TRUE       0  0.23371271          FALSE
## 6 FALSE FALSE  51   FALSE       0  0.06747828          FALSE
```

```
# Compare predicted values with actual values
thoracic_compare <- table(actual = thoracic_surgery_df$Risk1Yr, predicted = thoracic_surgery_df$probabil
thoracic_compare
```

```
##        predicted
## actual  FALSE TRUE
##   FALSE   369   31
##   TRUE     55   15
```

```
# Compute the accuracy
(thoracic_compare[[1,1]] + thoracic_compare [[2,2]]) / sum(thoracic_compare)
```

```
## [1] 0.8170213
```

The acccuracy of the model is about 82%.

2a

Fit a logistic regression model to the binary-classifier-data.csv dataset

```
library(mlogit)
```

```
## Loading required package: dfidx
```

```
##
## Attaching package: 'dfidx'
```

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

```
# Load binary-classifier-data.csv
binary_classifier_df <- read.csv('binary-classifier-data.csv')
head(binary_classifier_df)
```

```
##   label        x        y
## 1     0 70.88469 83.17702
## 2     0 74.97176 87.92922
```

```
## 3     0 73.78333 92.20325
## 4     0 66.40747 81.10617
## 5     0 69.07399 84.53739
## 6     0 72.23616 86.38403
```

```r
# Use the glm() function to perform the logistic regression.
binary_classifier_glm <-glm(label ~ x + y, data = binary_classifier_df, family = binomial())
```

2bi

What is the accuracy of the logistic regression classifier?

```r
summary(binary_classifier_glm)
```

```
##
## Call:
## glm(formula = label ~ x + y, family = binomial(), data = binary_classifier_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3728  -1.1697  -0.9575   1.1646   1.3989
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.424809   0.117224   3.624  0.00029 ***
## x           -0.002571   0.001823  -1.411  0.15836
## y           -0.007956   0.001869  -4.257 2.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2075.8  on 1497  degrees of freedom
## Residual deviance: 2052.1  on 1495  degrees of freedom
## AIC: 2058.1
##
## Number of Fisher Scoring iterations: 4
```

```r
# Add a column for the probability of label based on the model
binary_classifier_df$probability <-fitted(binary_classifier_glm)
head(binary_classifier_df)
```

```
##   label        x        y probability
## 1     0 70.88469 83.17702   0.3967211
## 2     0 74.97176 87.92922   0.3852176
## 3     0 73.78333 92.20325   0.3779152
## 4     0 66.40747 81.10617   0.4034378
## 5     0 69.07399 84.53739   0.3952460
## 6     0 72.23616 86.38403   0.3898045
```

```r
# Add a column for 1 and 0 for predictions based on the probability above or equal to 0.43
binary_classifier_df$probability_label<-if_else(binary_classifier_df$probability >= .43, 1, 0)
head(binary_classifier_df)
```

```
##   label        x        y probability probability_label
## 1     0 70.88469 83.17702   0.3967211                 0
## 2     0 74.97176 87.92922   0.3852176                 0
## 3     0 73.78333 92.20325   0.3779152                 0
## 4     0 66.40747 81.10617   0.4034378                 0
## 5     0 69.07399 84.53739   0.3952460                 0
## 6     0 72.23616 86.38403   0.3898045                 0
```

```r
# Compare predicted values with actual values
binary_compare <- table(actual = binary_classifier_df$label, predicted = binary_classifier_df$probabilit
binary_compare
```

```
##       predicted
## actual   0   1
##      0 249 518
##      1  58 673
```

```r
# Compute the accuracy
(binary_compare[[1,1]] + binary_compare[[2,2]]) / sum(binary_compare)
```

```
## [1] 0.6154873
```

The accuracy is about 62%. After testing out different thresholds, 0.43 gave the best accuracy. An accuracy of 62% can mean that the variables did not have a linear relationship.