1 Packages

In [9]:
```
install.packages('regclass')
```
executed in 27.5s, finished 16:32:06 2019-11-17

. . .

In [11]:
```
install.packages('ISLR')
```
executed in 4.70s, finished 16:32:38 2019-11-17

. . .

In [13]:
```
install.packages('pROC')
```
executed in 3.23s, finished 16:33:28 2019-11-17

. . .

In [14]:
```
rm(list=ls())
library(readxl)
library(Hmisc)
library(MASS)
library(caret)
library(regclass)
library(ISLR)
library(boot)
library(vcd)
library(pROC)
```
executed in 64ms, finished 16:33:31 2019-11-17

. . .

2 File

In [32]:
```
setwd("D:/BAX401/HW3")
Q2<-read.csv('Q2.csv')
colnames(Q2)<-c('id','join','age','churn','spend')
Q2$join<-as.factor(Q2$join)
Q2$churn<-as.factor(Q2$churn)
Q2
```
executed in 63ms, finished 17:02:52 2019-11-17

. . .

In [33]:
```
 str(Q2)
```
executed in 24ms, finished 17:02:56 2019-11-17

```
'data.frame':   199 obs. of  5 variables:
 $ id   : int  1000201 1000202 1000203 1000204 1000205 1000206 1000207 1000208 1000209 1000210
...
 $ join : Factor w/ 2 levels "0","1": 2 1 1 2 2 2 2 1 1 2 ...
 $ age  : int  7 7 8 2 5 3 5 8 7 5 ...
 $ churn: Factor w/ 2 levels "0","1": 1 2 2 2 1 2 1 2 1 2 ...
 $ spend: int  88 103 45 113 99 68 86 58 106 50 ...
```

3 Logistic Regression

In [34]:
```
mylogit1<-glm(churn~join+age+spend,data=Q2,family=binomial(link="logit"))
summary(mylogit1)
```

executed in 35ms, finished 17:03:00 2019-11-17

```
Call:
glm(formula = churn ~ join + age + spend, family = binomial(link = "logit"),
    data = Q2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6753  -1.2113   0.7973   1.0979   1.2894

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.474797   0.523983   0.906  0.36487
join1        0.916584   0.355287   2.580  0.00988 **
age         -0.055849   0.071598  -0.780  0.43537
spend       -0.002819   0.005655  -0.498  0.61815
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 268.95  on 198  degrees of freedom
Residual deviance: 260.42  on 195  degrees of freedom
AIC: 268.42

Number of Fisher Scoring iterations: 4
```

4 Confusion Matrix

In [35]:
```
confmat1<-confusion_matrix(mylogit1) #Predict True/False Positive/Negative (TP,TN,FP.FN)
confmat1
```

executed in 26ms, finished 17:03:11 2019-11-17

	Predicted 0	Predicted 1	Total
Actual 0	24	57	81
Actual 1	17	101	118
Total	41	158	199

In [36]:
```
#first get predicted values
preddata<-with(Q2,data.frame(id,join,age,churn,spend))
probdefault<-predict(mylogit1,newdata=preddata,type="response")
preddefault<-ifelse(probdefault > 0.5, 1,0) #at what level should we say prob(default)=1

#Let's determine Accuracy manually first
missclass<-preddefault!=Q2$churn
misclasserror<-round(mean(preddefault!=Q2$churn),4)
print(paste('Accuracy',1-misclasserror)) #To determine accuracy manually
```

executed in 31ms, finished 17:03:14 2019-11-17

```
[1] "Accuracy 0.6281"
```

In [37]:
```
confMat2<-confusionMatrix(data = as.factor(preddefault),reference = as.factor(Q2$churn),positive
confMat2 ###Note, because of how this matrix is strutured, 0,0 becomes true positive -- thus we s
```

executed in 24ms, finished 17:03:17 2019-11-17

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  24  17
         1  57 101

               Accuracy : 0.6281
                 95% CI : (0.557, 0.6954)
    No Information Rate : 0.593
    P-Value [Acc > NIR] : 0.1743

                  Kappa : 0.165

 Mcnemar's Test P-Value : 5.797e-06

            Sensitivity : 0.8559
            Specificity : 0.2963
         Pos Pred Value : 0.6392
         Neg Pred Value : 0.5854
             Prevalence : 0.5930
         Detection Rate : 0.5075
   Detection Prevalence : 0.7940
      Balanced Accuracy : 0.5761

       'Positive' Class : 1
```

5 F Measure

In [30]:
```
(2*confMat2[['byClass']][["Pos Pred Value"]]*confMat2[['byClass']][["Sensitivity"]])/(confMat2[['
```

executed in 22ms, finished 16:49:25 2019-11-17

0.731884057971015

6 Train - Test

In [39]:
```
set.seed(20)
sample_siz = floor(0.75*nrow(Q2))  # creates a value for dividing the data into train and test. I
sample_siz #how big?
train_index = sample(seq_len(nrow(Q2)),size = sample_siz)# Randomly identifies therows equal to s

train=Q2[train_index,] #creates the training dataset with row numbers stored in train_ind
test=Q2[-train_index,]  # creates the test dataset excluding the row numbers mentioned in train_i

#Logistic Regression Model Estimation
mylogit_train<-glm(churn~join+age+spend,data=Q2,family=binomial(link="logit"))

#coefficients
summary(mylogit_train)

#Predict using Test data
preddata_test<-with(test,data.frame(id,join,age,churn,spend))
probdefault_test<-predict(mylogit_train,newdata=preddata_test,type="response")
preddefault_test<-ifelse(probdefault_test > 0.5, 1,0) #at what level should we say prob(default)=

missclass_test<-preddefault_test!=test$churn
misclasserror_test<-round(mean(preddefault_test!=test$churn),4)
print(paste('Accuracy',1-misclasserror_test))
```

executed in 51ms, finished 17:16:37 2019-11-17

```
149


Call:
glm(formula = churn ~ join + age + spend, family = binomial(link = "logit"),
    data = Q2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6753  -1.2113   0.7973   1.0979   1.2894

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.474797   0.523983   0.906  0.36487
join1        0.916584   0.355287   2.580  0.00988 **
age         -0.055849   0.071598  -0.780  0.43537
spend       -0.002819   0.005655  -0.498  0.61815
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 268.95  on 198  degrees of freedom
Residual deviance: 260.42  on 195  degrees of freedom
AIC: 268.42

Number of Fisher Scoring iterations: 4


[1] "Accuracy 0.74"
```

In [40]:
```
anova(mylogit, test="Chisq")
```

executed in 39ms, finished 17:17:27 2019-11-17

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL	NA	NA	198	268.9530	NA
join	1	7.689494	197	261.2635	0.005554321

7 K Fold

In [41]:
```
set.seed(20)
cv.error.10=rep(0 ,10)
for (i in 1:10){
  glm.fit=glm(churn~join+age+spend,data=Q2,family=binomial(link="logit"))
  cv.error.10[i]=cv.glm(Q2,glm.fit,K=10)$delta[1]
}
cv.error.10
```

executed in 402ms, finished 17:18:44 2019-11-17

0.247018780510476 0.243152391763003 0.240020440820298 0.245207389084416 0.241789260507343
0.241440670724236 0.241034541449709 0.23751352108389 0.240915772890144 0.239373497752602

In [ ]: