**Building a robust Geodemographic Segmentation Model**

**Applying Logistic Regression and step by step building a model**

Grouping of customers by similarities of their behavior and using prior knowledge to predict any future trends and basically predict future behavior. Here, we did Churn Modeling to understand when your customers are going to leave and who’s more likely to leave, who’s less likely to leave in a bank scenario.

*#Geodemographic segmentation model*

*#Complete data set*

library(readxl)

Churn\_Modelling <- read\_excel("C:/Users/bvkka/Desktop/Udemy/Data Science/Churn-Modelling.xlsx")

View(Churn\_Modelling)

*# creating dummy variables*

install.packages("plyr")

library(plyr)

Churn\_Modelling$Geography <- revalue(Churn\_Modelling$Geography,c("France"=0))

Churn\_Modelling$Geography <- revalue(Churn\_Modelling$Geography,c("Spain"=1))

Churn\_Modelling$Geography <- revalue(Churn\_Modelling$Geography,c("Germany"=2))

Churn\_Modelling$Gender <- revalue(Churn\_Modelling$Gender,c("Female"=0))

Churn\_Modelling$Gender <- revalue(Churn\_Modelling$Gender,c("Male"=1))

Female<-as.numeric(Churn\_Modelling$Gender==0)

Spain<-as.numeric(Churn\_Modelling$Geography==1)

Germany<-as.numeric(Churn\_Modelling$Geography==2)

*#character to numeric*

Churn\_Modelling$Gender<-as.numeric(as.character(Churn\_Modelling$Gender))

Churn\_Modelling$Geography<-as.numeric(as.character(Churn\_Modelling$Geography))

wealthAccumulation<-(Churn\_Modelling$Balance)/(Churn\_Modelling$Age)

Age1<-Churn\_Modelling$Age

Balance1<-Churn\_Modelling$Balance

Balance2<-log10(Balance1+1)

wealthAccumulationlog<-log10(Balance1/Age1+1)

*#logestic regression*

*#creating a model*

model1<-glm(formula = Exited ~ CreditScore+wealthAccumulationlog+Age+NumOfProducts+IsActiveMember+Female+Germany, binomial(link="logit"),data =Churn\_Modelling)

summary(model1)

*#library(dplyr)*

prob\_predict1=predict(model1,type = 'response')

summary(prob\_predict1)

y\_pred=ifelse(prob\_predict1>0.5,1,0) *#vector of predictions*

y\_pred

*#Making the confusion Matrix*

cm=table(y\_pred,Churn\_Modelling$Exited)

cm

TN<-cm[1] *#7681*

FN<-cm[2] *#282*

FP<-cm[3] *#1605*

TP<-cm[4] *#432*

ActualYes<-FN+TP

ActualNo<-TN+FP

PredictedYes<-FP+TP

PredictedNo<-TN+FN

*#accuracy (TP+TN)/(TP+TN+FN+FP)*

accuracy1=mean(y\_pred==Churn\_Modelling$Exited)

accuracy1

*#misclassification rate: (FP+FN)/(TP+TN+FN+FP)*

MR<-(FP+FN)/(TN+FN+FP+TP)

*#TPR : When it's actually yes, how often does it predict yes?*

TPR<-TP/(ActualYes)

*#FPR : when it's actually no, how often does it predict yes?*

FPR<-FP/(ActualNo)

*#precision : when it predicts yes, how often is it correct? TP/TP+FP*

precision1=precision<-diag(cm)/colSums(cm)

precision1

*#recall :*

recall1=recall<-diag(cm)/rowSums(cm)

recall1

*#ROC Curve*

library(pROC)

myROC<-roc(response=Churn\_Modelling$Exited,predictor = prob\_predict1,positve='prob\_predict1')

plot(myROC)

pred1<-prediction(prob\_predict1,Churn\_Modelling$Exited)

roc.perf=performance(pred1,measure = "tpr",x.measure = "fpr")

ggplot(mode)

*#auc 0.7669*

auc(roc(Churn\_Modelling$Exited,prob\_predict1))

*#####now we add a new test data and see how classifier predicts#####*

*#test data set*

*##merged test data with train data except the lasr column, model should predict that*

library(readxl)

Churn\_Modelling\_testt <- read\_excel("C:/Users/bvkka/Desktop/Udemy/Data Science/Churn-Modelling-testt.xlsx")

View(Churn\_Modelling\_testt)

Churn\_Modelling\_testt$Gender <- revalue(Churn\_Modelling\_testt$Gender,c("Female"=0))

Churn\_Modelling\_testt$Gender <- revalue(Churn\_Modelling\_testt$Gender,c("Male"=1))

Churn\_Modelling\_testt$Geography <- revalue(Churn\_Modelling\_testt$Geography,c("France"=0))

Churn\_Modelling\_testt$Geography <- revalue(Churn\_Modelling\_testt$Geography,c("Spain"=1))

Churn\_Modelling\_testt$Geography <- revalue(Churn\_Modelling\_testt$Geography,c("Germany"=2))

Churn\_Modelling\_testt$Gender<-as.numeric(as.character(Churn\_Modelling\_testt$Gender))

Churn\_Modelling\_testt$Geography<-as.numeric(as.character(Churn\_Modelling\_testt$Geography))

model2<-glm(formula = Exited ~ CreditScore+wealthAccumulationlog+Age+NumOfProducts+IsActiveMember+Female+Germany, binomial(link="logit"),data =Churn\_Modelling\_testt)

summary(model2)

*#predicting the test set results*

prob\_pred2=predict(model1,type='response',newdata = Churn\_Modelling\_testt) *#for predicitng we only need predictors , but not response*

summary(prob\_pred2)

y\_pred2=ifelse(prob\_pred2>0.5,1,0) *#vector of predictions*

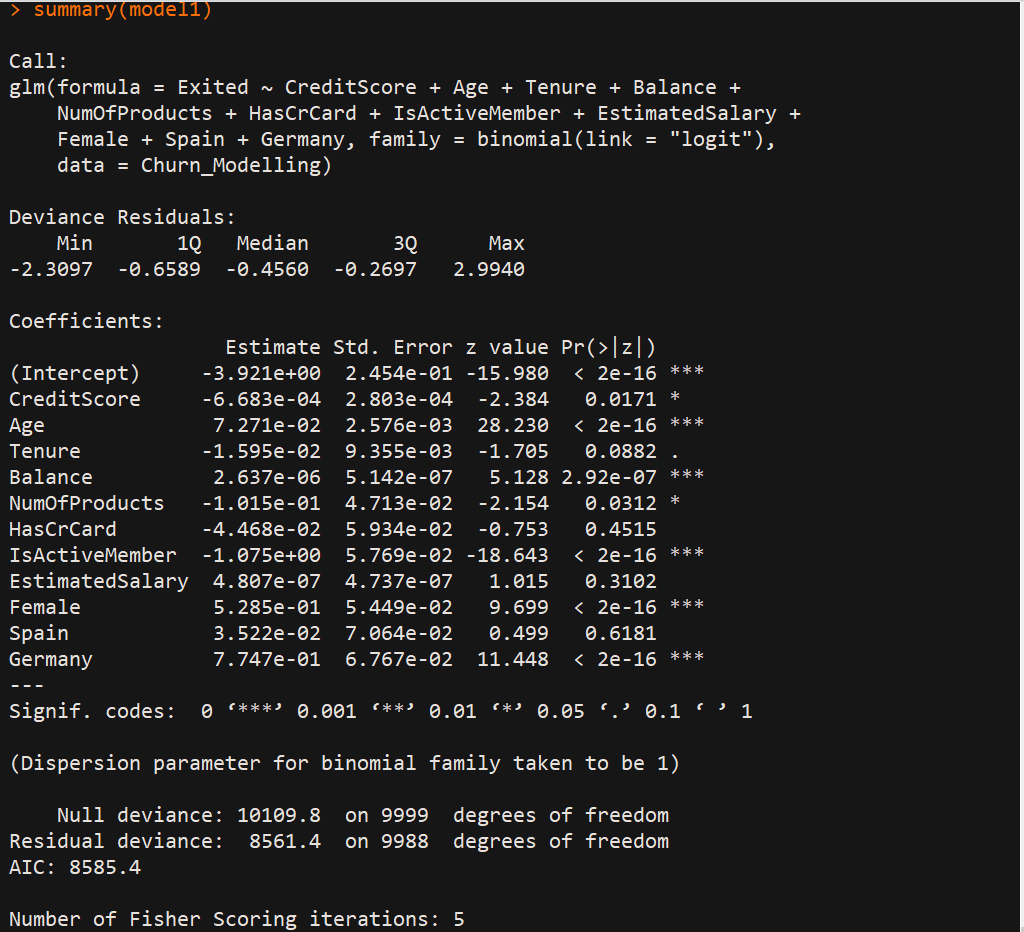
y\_pred2

*#Making the confusion Matrix*

cm=table(y\_pred2,Churn\_Modelling\_testt$Exited)

cm

1. Building the model – First iteration

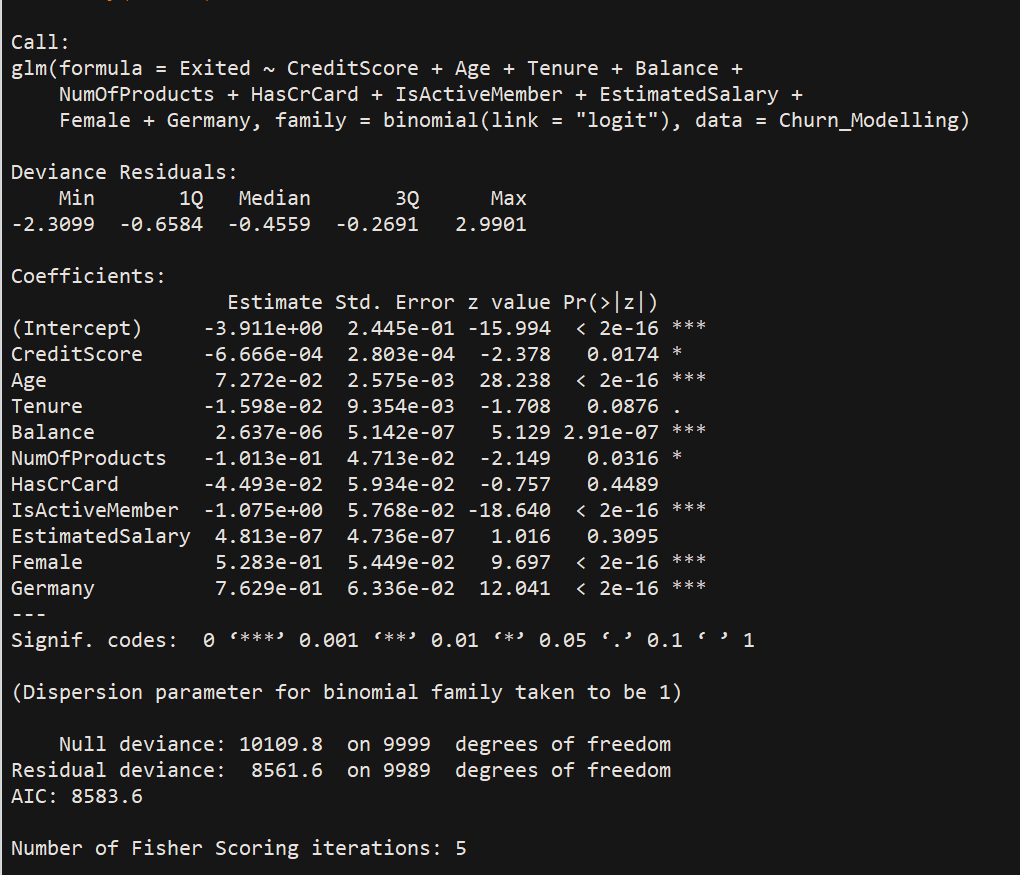


1. Applying backward elimination: step by step

Now checking the p values, we see that p value of Country: Spain is very high (0.6181)

Greater than threshold(p=0.05), so we are excluding Spain.

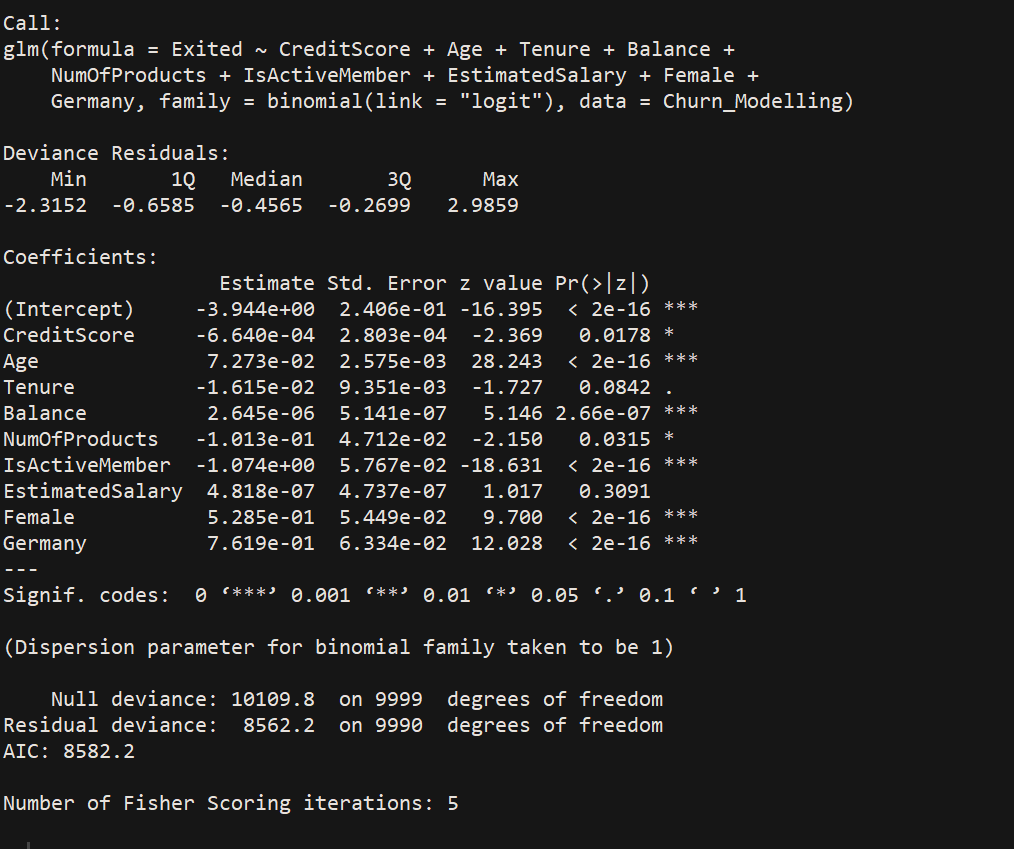
2nd Iteration:



Now checking the p values, we see that p value of HasCrCard is very high (0.4489)

Greater than threshold(p=0.05), so we are excluding that as well.

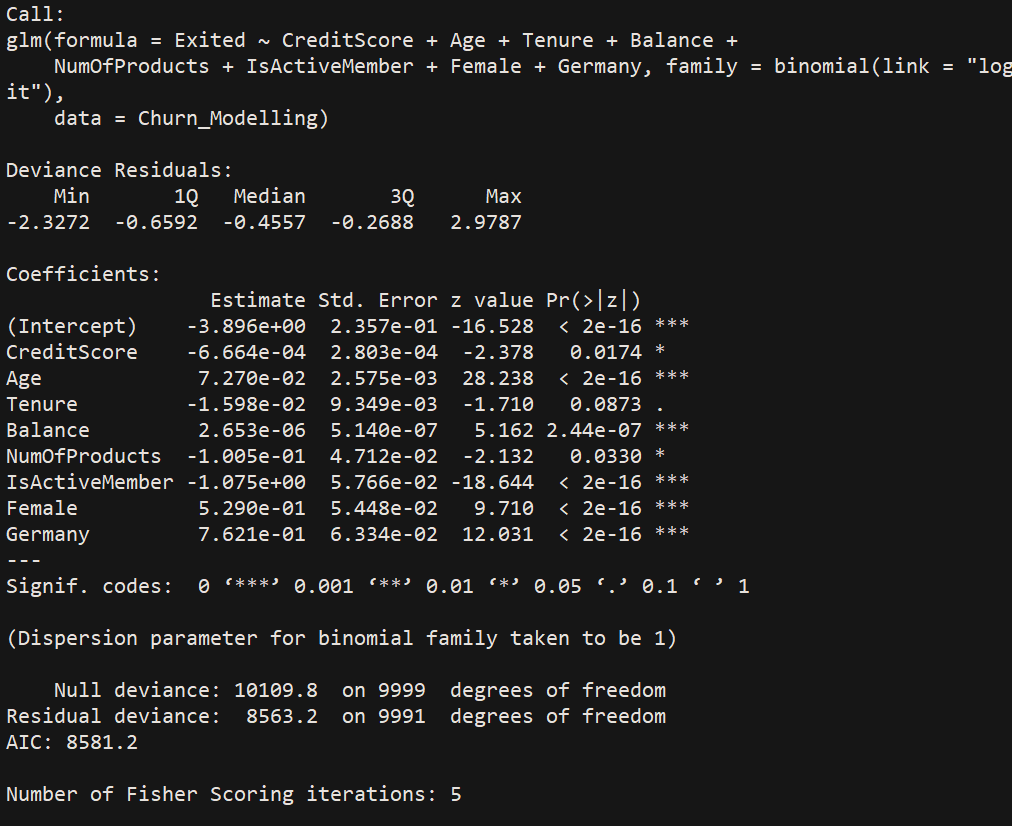
3nd Iteration:



Now checking the p values, we see that p value of EstimatedSalary is high (0.3091)

Greater than threshold(p=0.05), so we are excluding that as well.

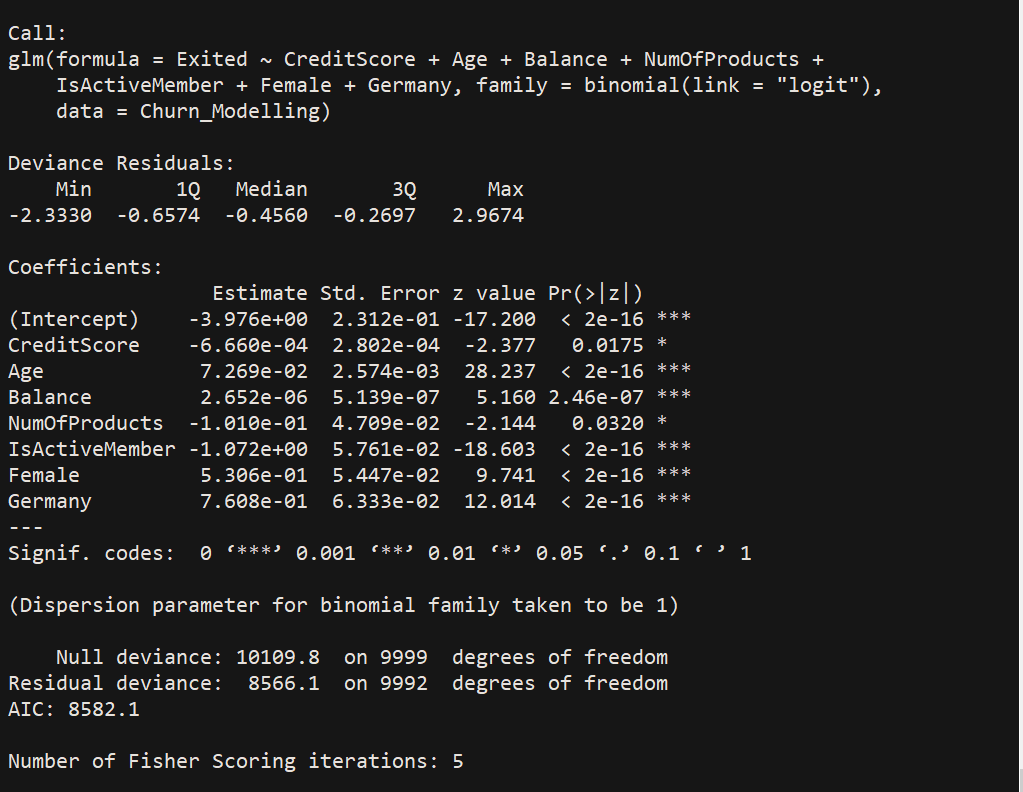
4th Iteration:



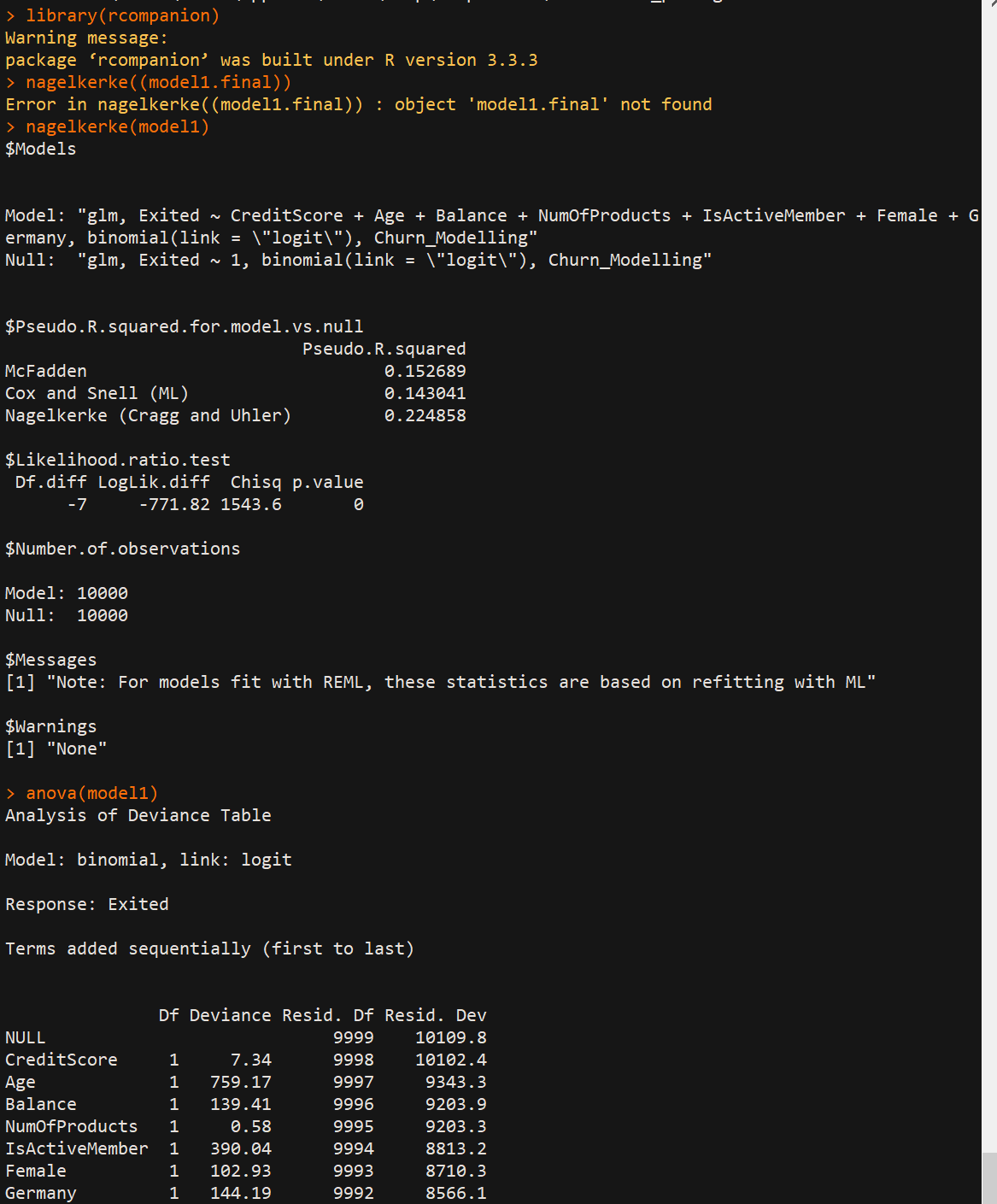
Now checking the p values, we see that p value of Tenure is high (0.0873)

Greater than threshold(p=0.05), so we are excluding that as well.

5th Iteration:

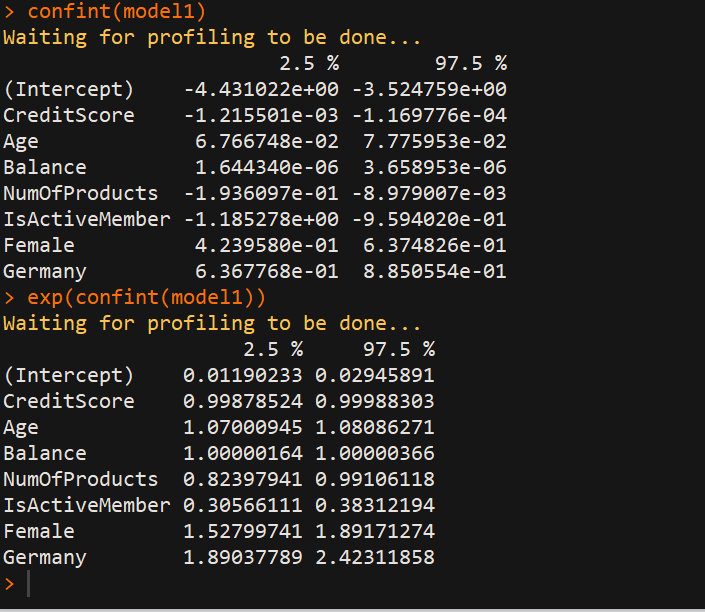


For R squared – use library rcompanion and also finding ANOVA for the model



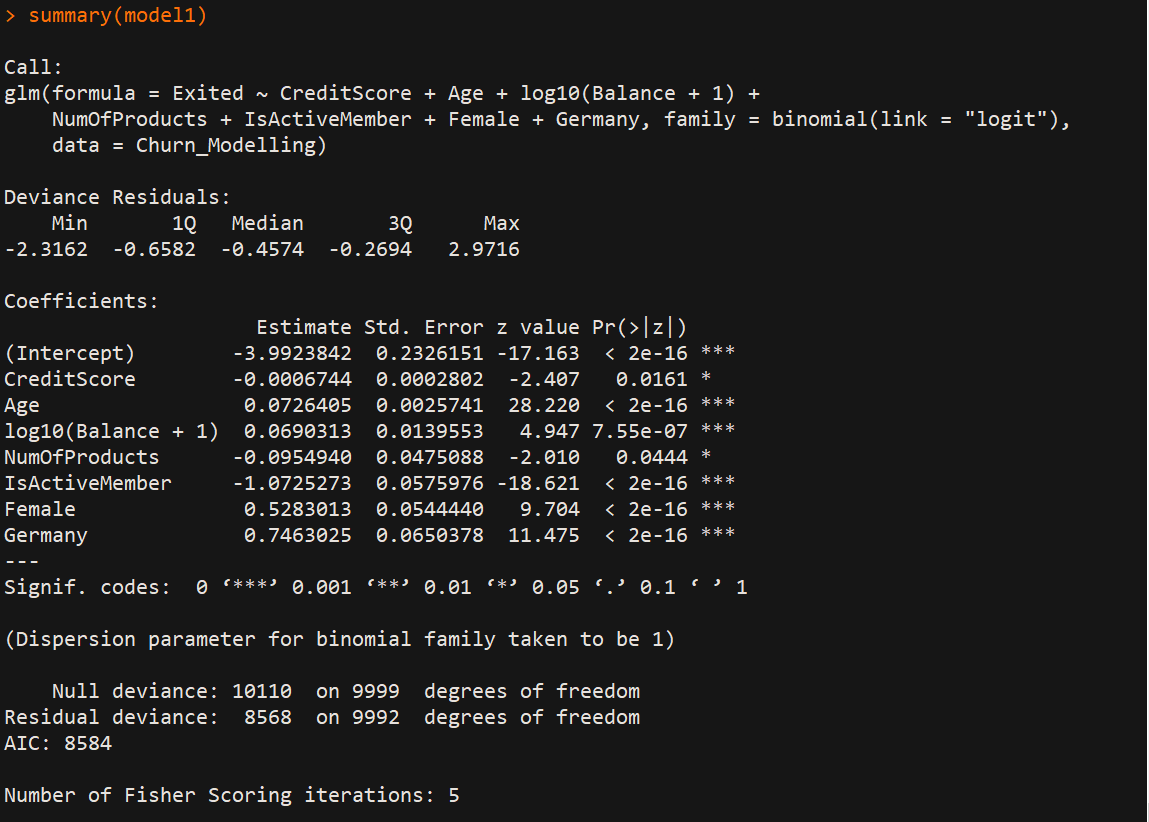
Coefficients and exponential coefficients

At 95% CI

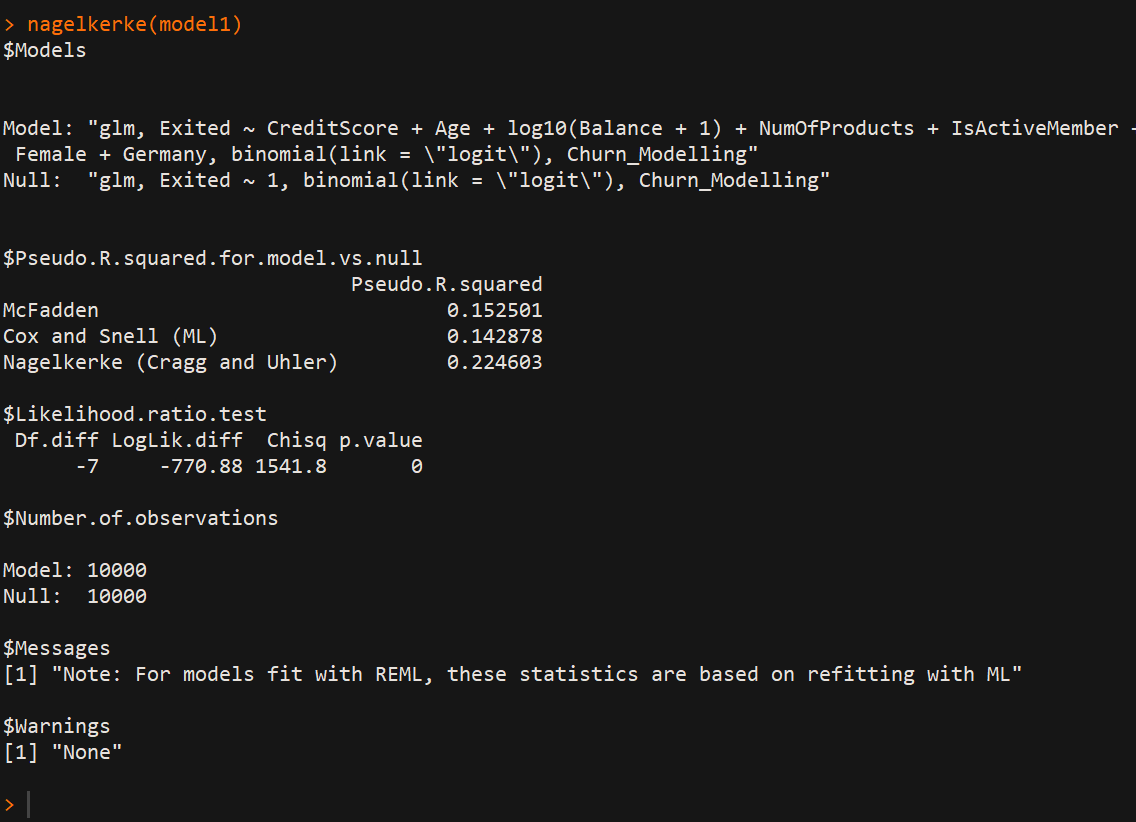


1. Step 3: Applying significant transformations to the independent variables: to get better results or make the model more robust:

We are going to change variable ‘balance’ to ‘log10(balance+1)’ i.e applying logarithmic transformation to the varaiable and then run the model again



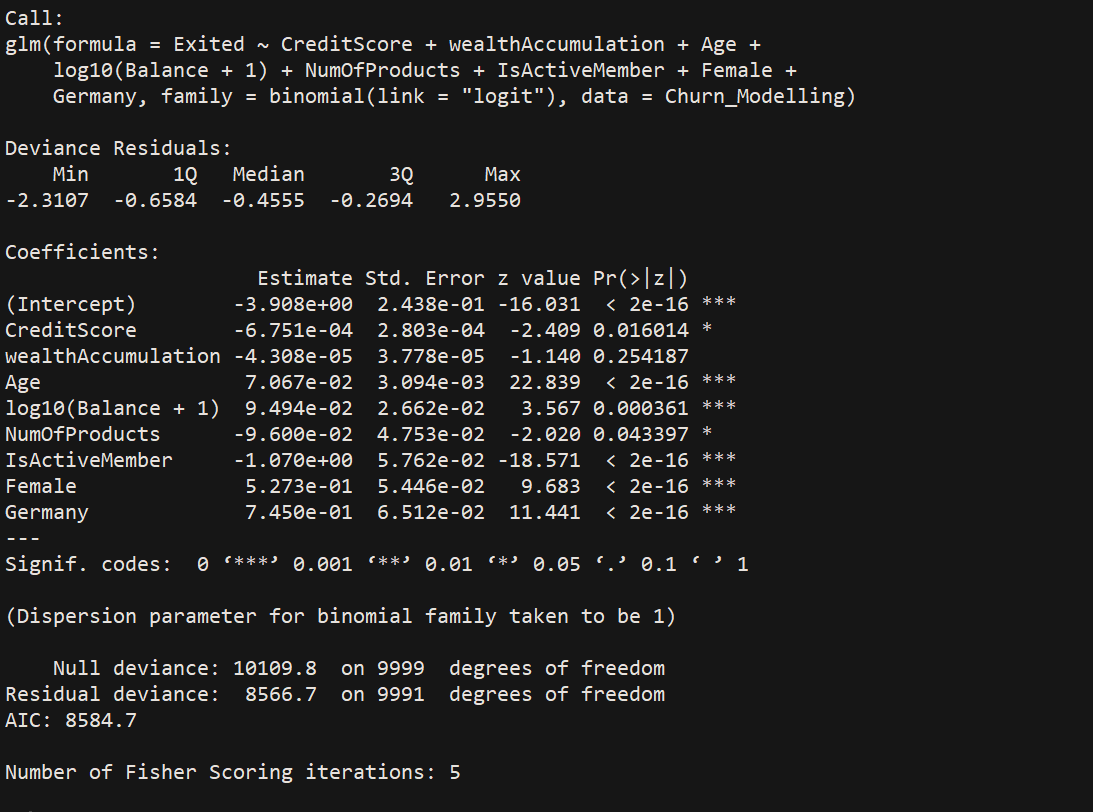
We notice that the p value of balance (After log transformation) has become more significant.



But now R squared dropped a bit. Nevertheless, I prefer to keep the balance as log balance, we don’t always get such results. Sometimes it happens, good thing is accuracy increased.

1. Creating a derived variable:

Wealth Accumulation: Balance/Age

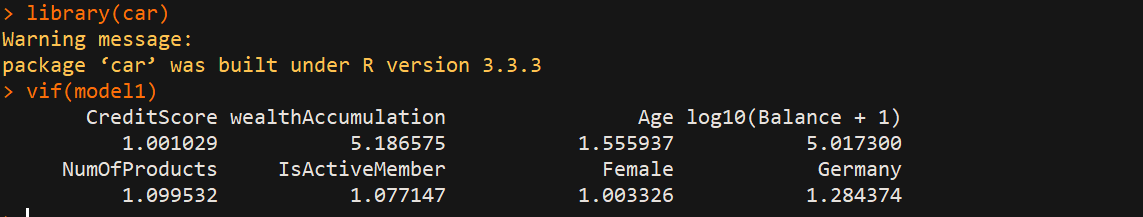




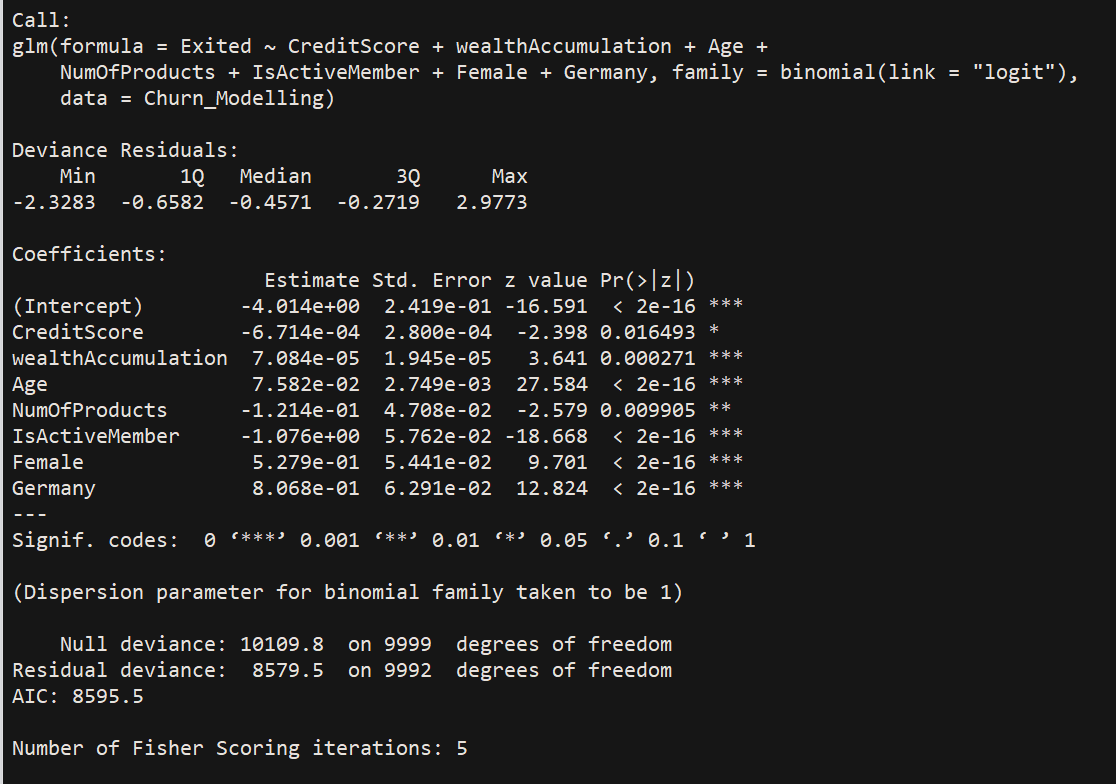
Hmmm now this new variable is not significant at all, even r square decreased. But, we are going to remove other variables to make this significant. We will see that in multicollinearity using VIF.

1. Checking for multicollinearity using VIF:

Sometimes independent variables are corelated and can damage the model. So, here wealth accumulation, age and balance are all linked. Do VIF and remove the variable with VIF > 5.

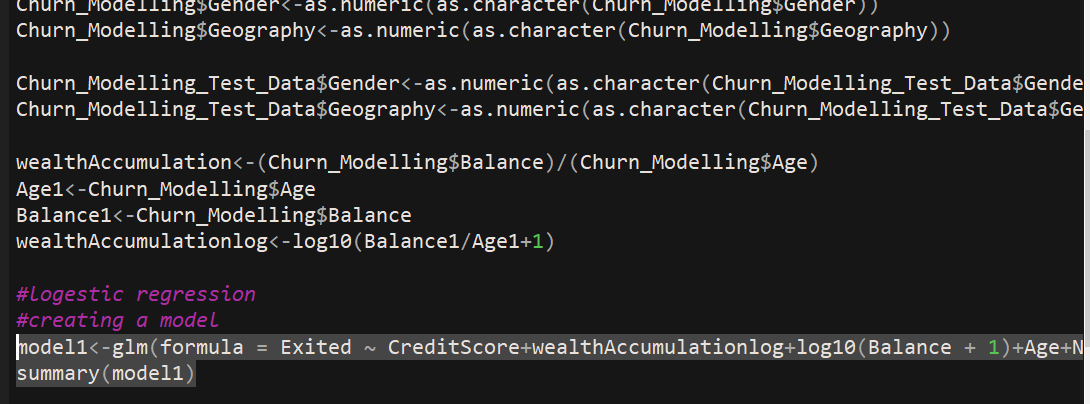


Here, you can see wealth acc and balance having vif>5. Lets go ahead and remove Balance variable and then run the model.

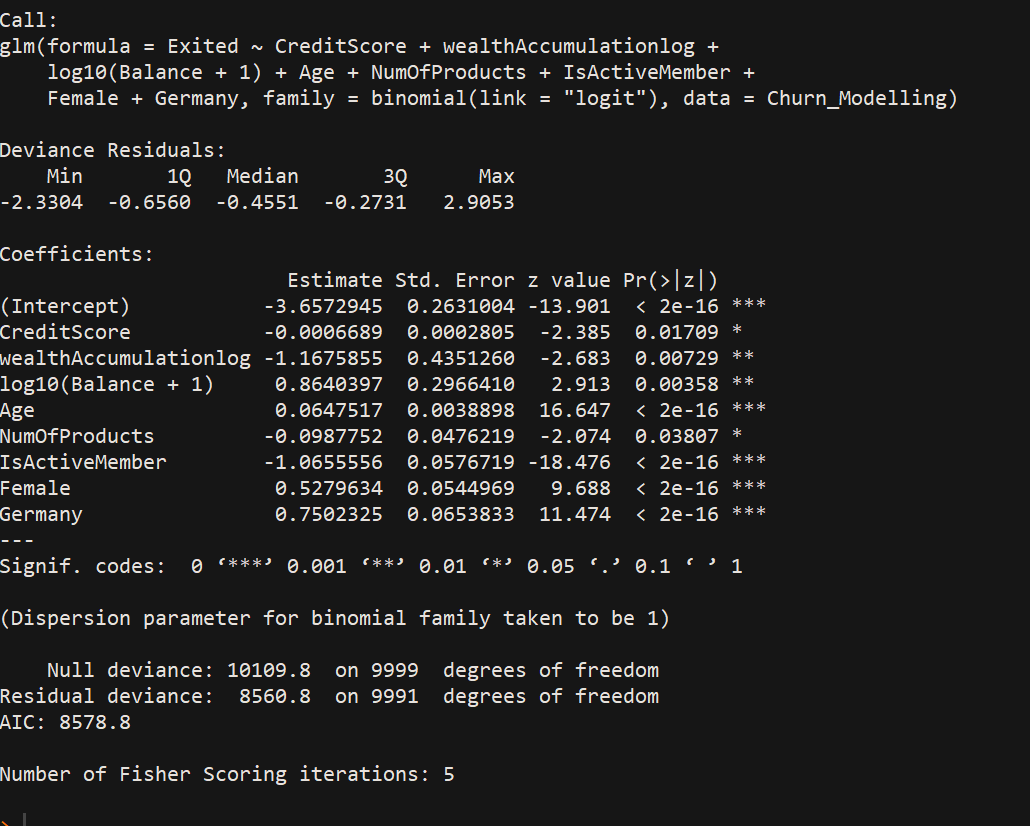


We can notice the wealth accumulation variable is now very significant.

Now let us try making another transformation. Lets do log of both balance and Wealth Accumulation and include them in the model and run again.

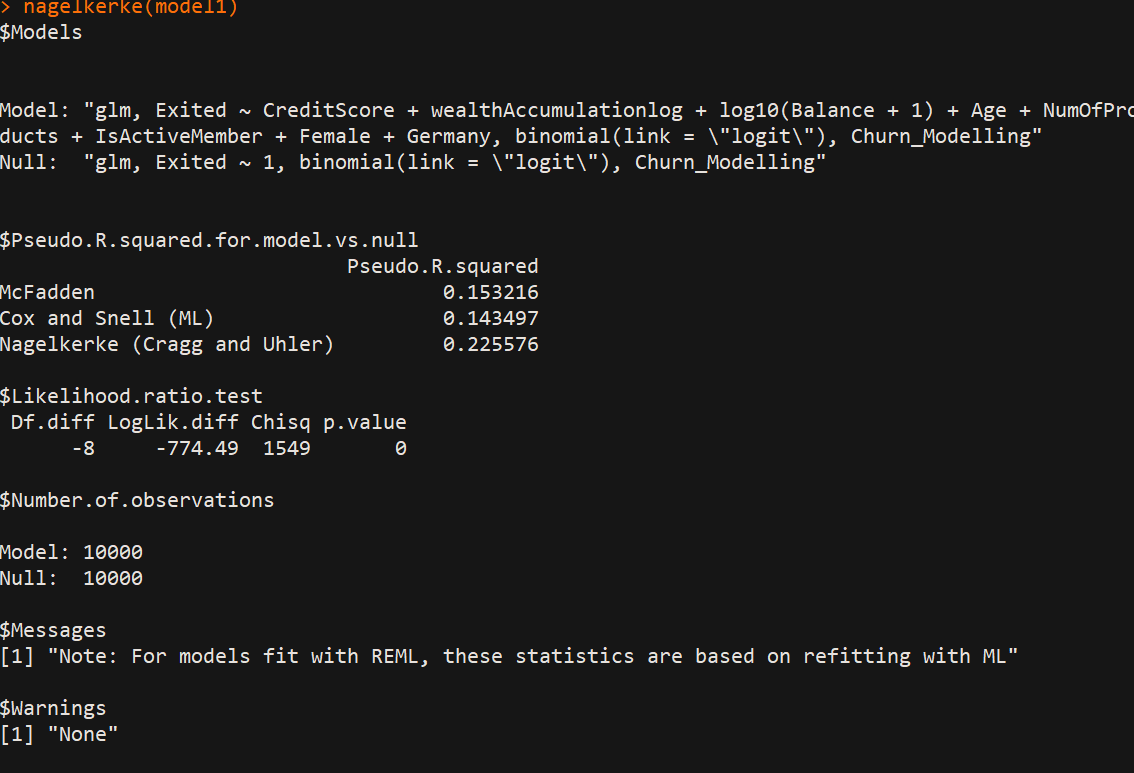


We notice that both Wealth accumulation log and Balance log variables are significant in this model now.

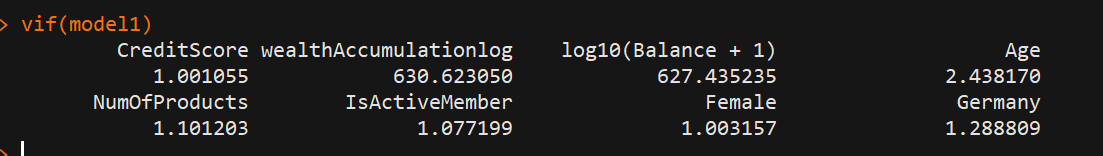




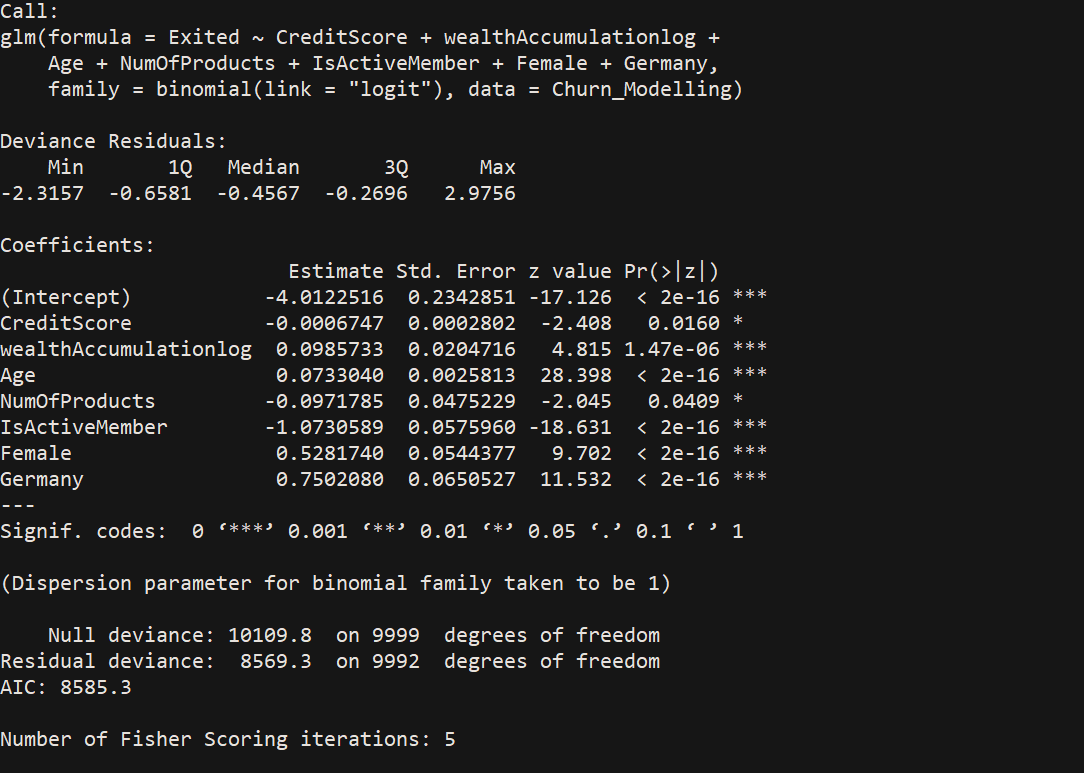
We also note that R squared value is increased and better than previous ones

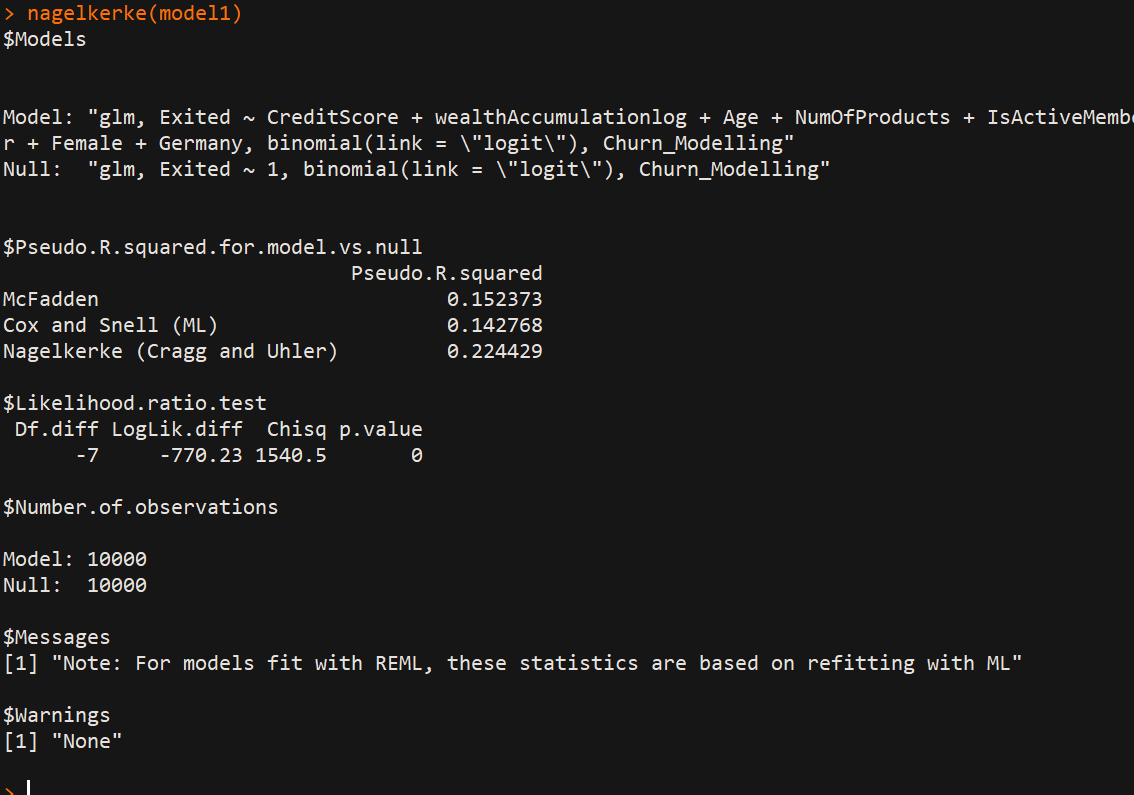


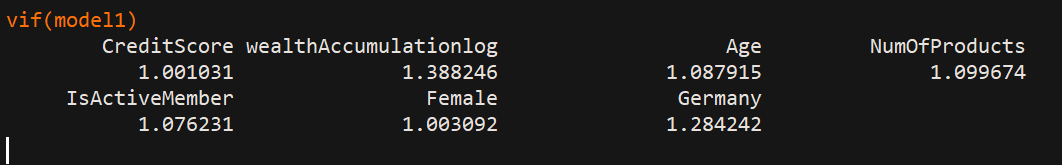
But, we cannot conclude that the model is robust now. Check VIF now



The values of WA and balance are through the roof, so it means these both are the same, extra ordinary collinearity. So, we have to omit one which makes the model robust, p values significant and R squared values high, So by log balance out and check



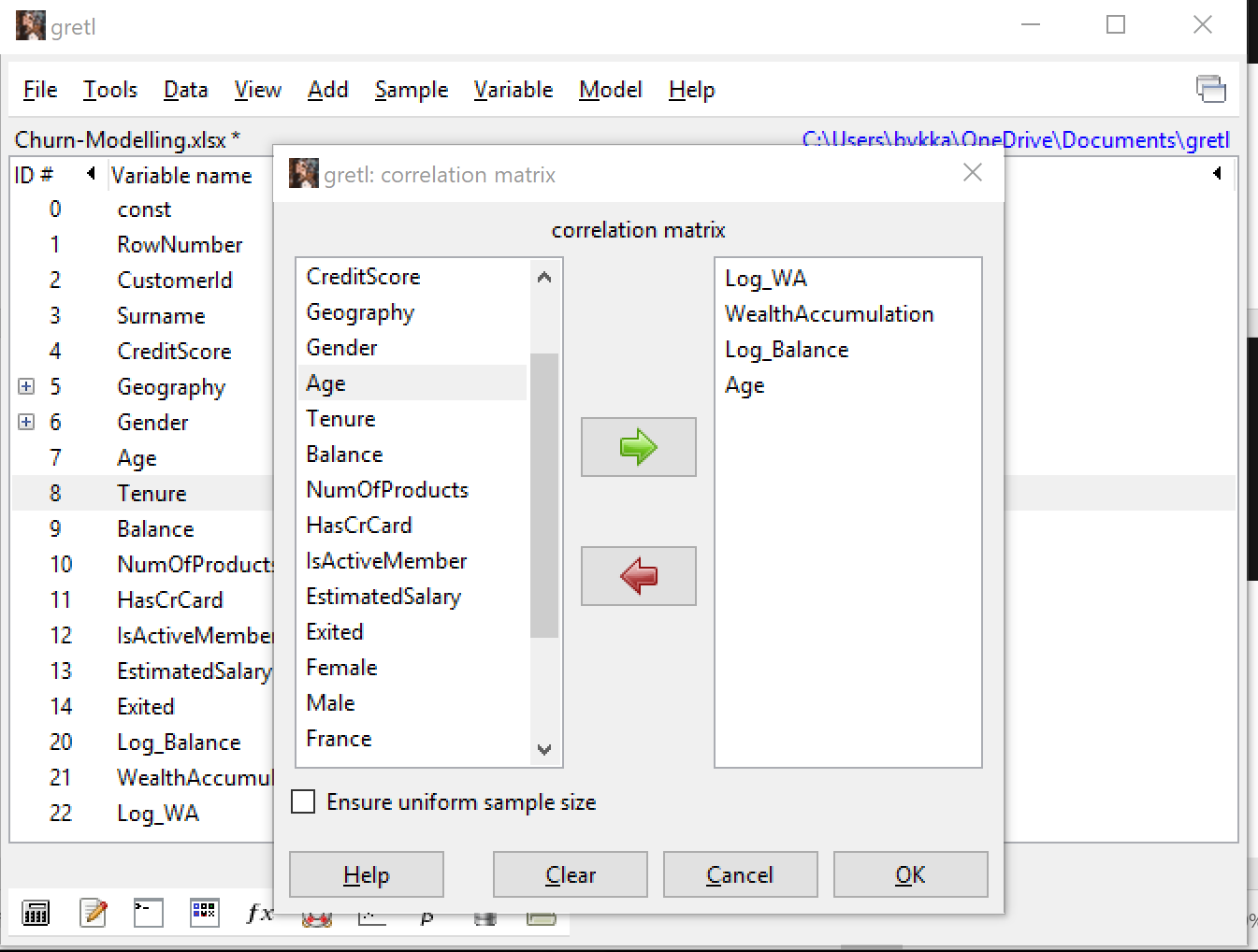


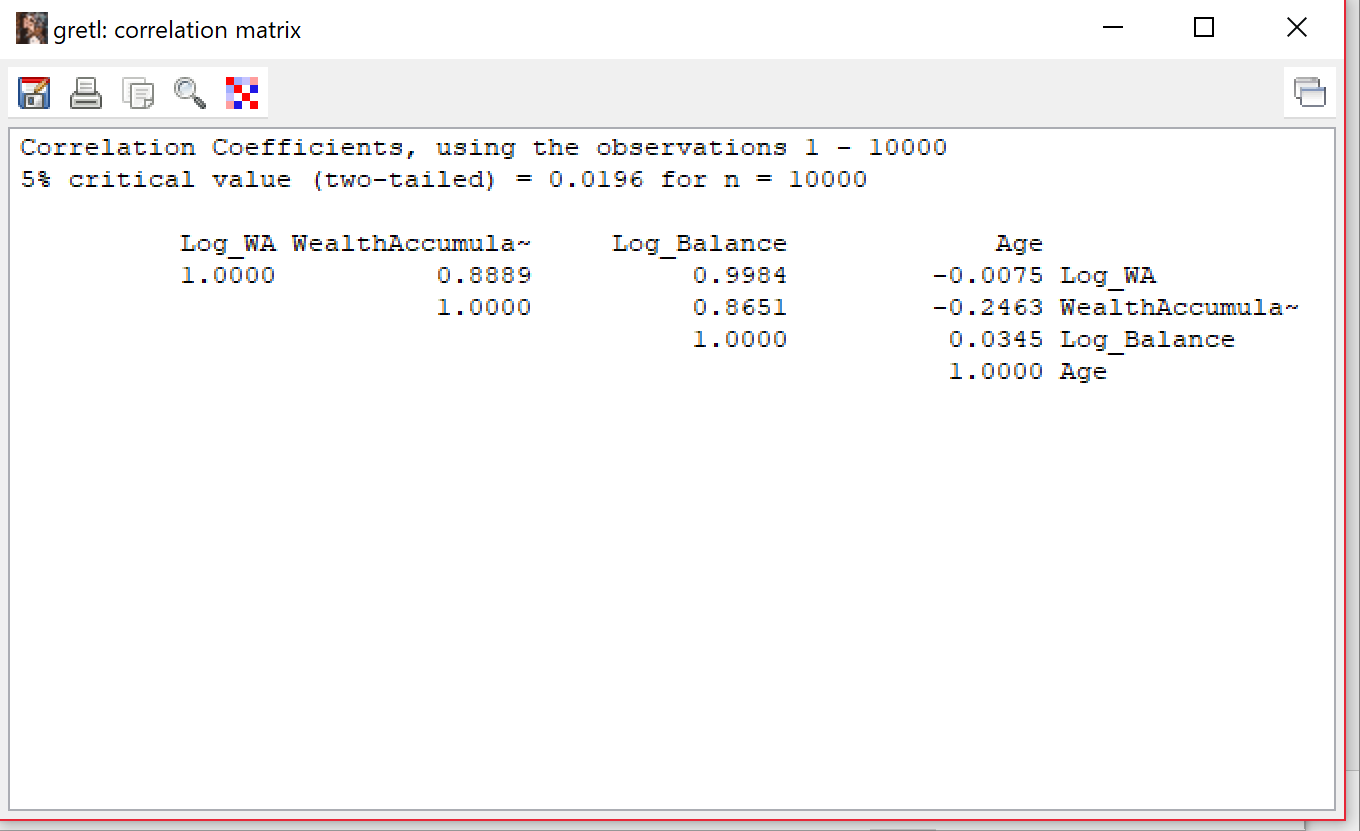


Everything satisfied now 😊

1. Correlation Matrix: a **correlation matrix**, which is used to investigate the dependence between multiple variables at the same time. The result is a table containing the **correlation coefficients**between each variable and the others.

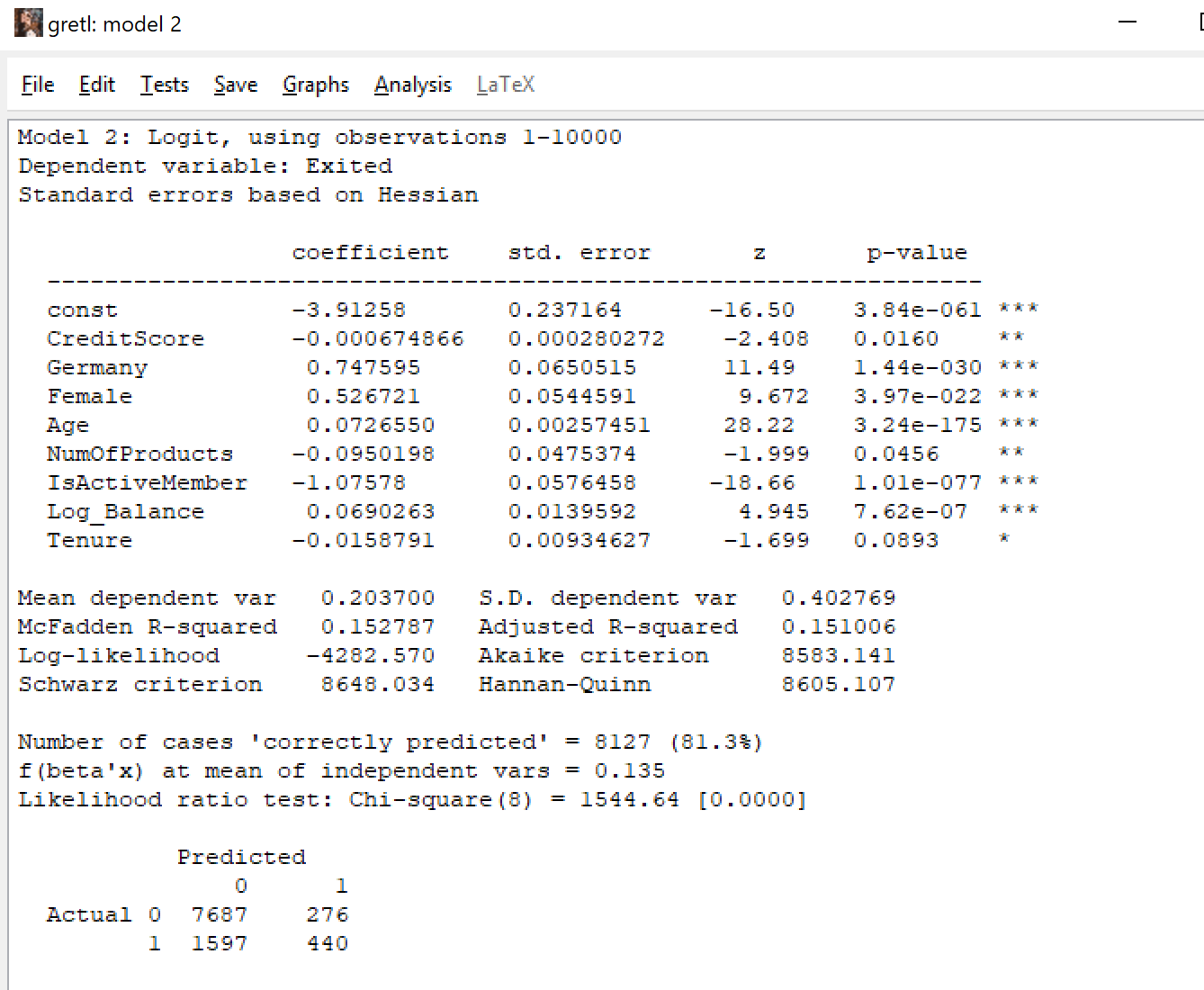
I’m showing using gretl



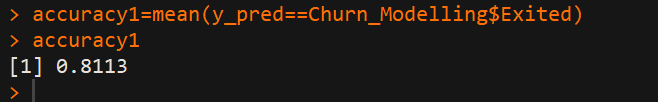


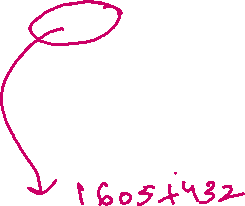
You can see log balance and log wa are almost same, therefore bad for our model and we had to remove one.

1. Final trained model:

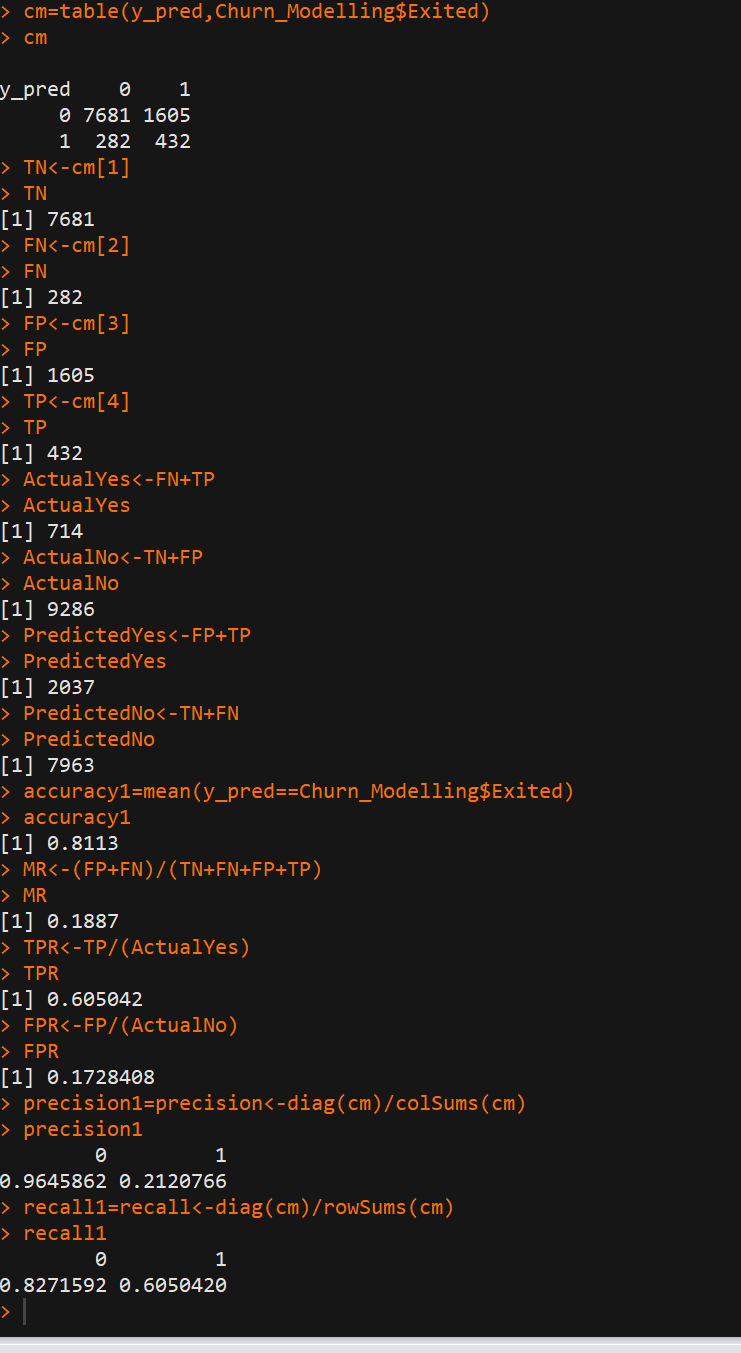


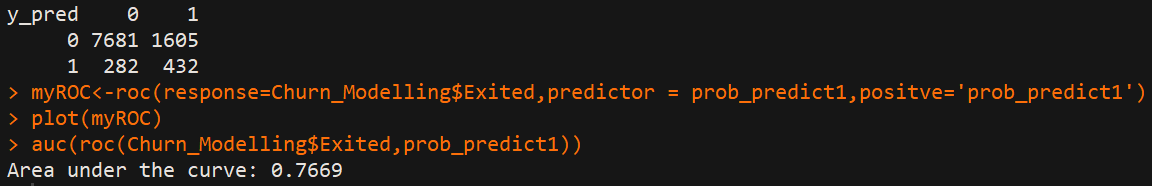


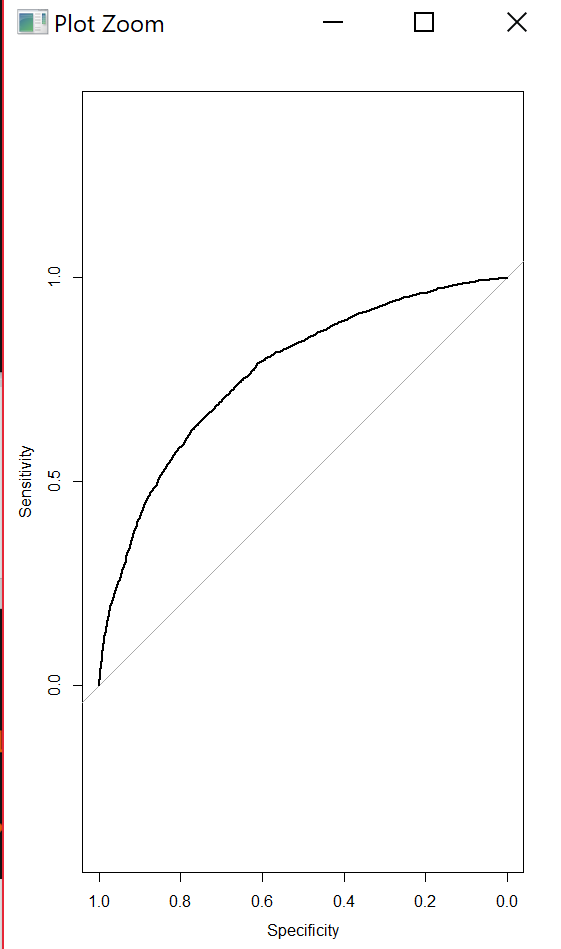


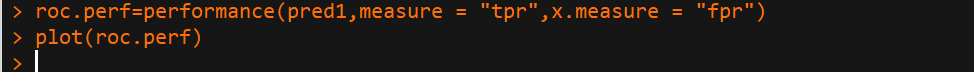


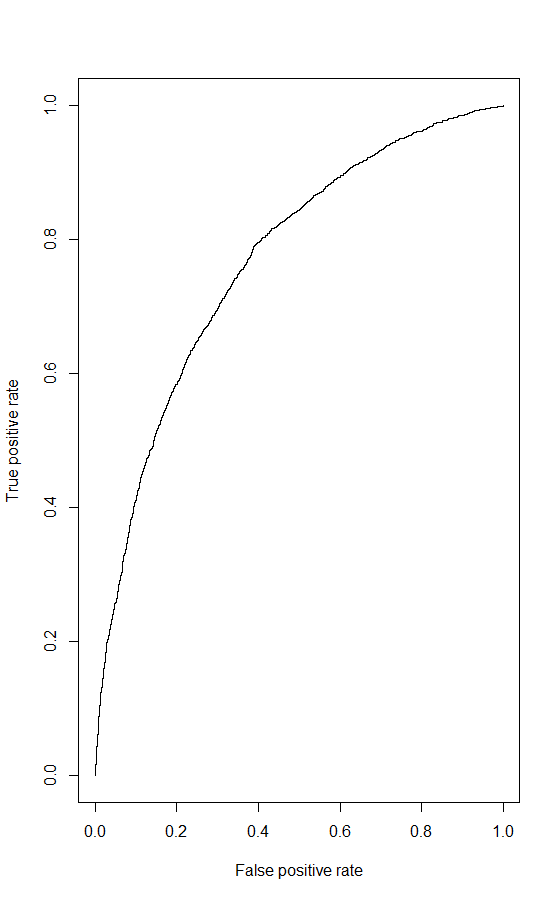
Assessing model using Confusion Matrix table in R :





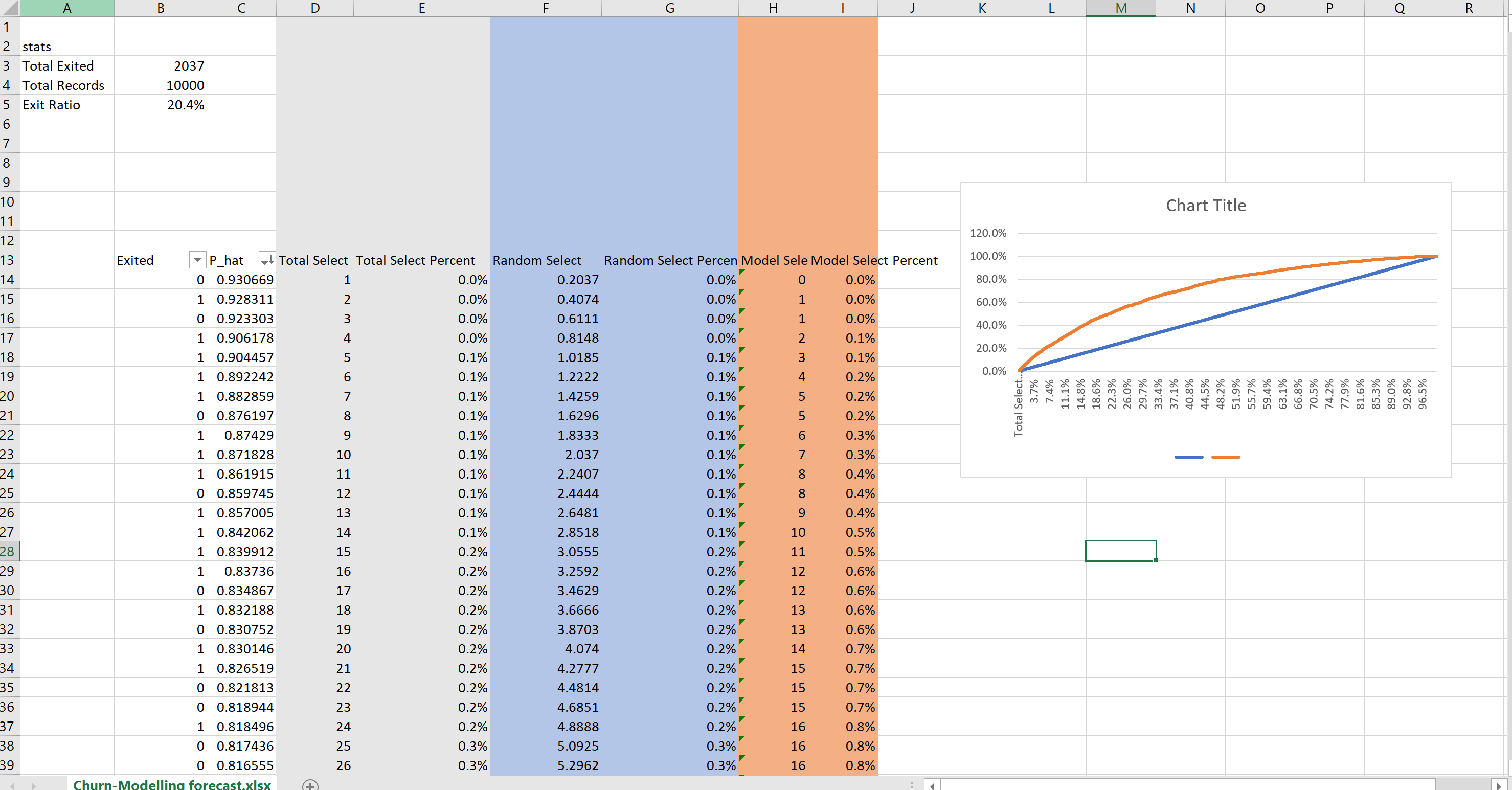






**Assessing my model:**

1. Build a CAP curve





Same 81% I showed in R studio by using accuracy = mean(y\_pred==ChurnModelling$Exited)

So, this is all we did for the train data set, train model basically has accuracy of 81%. It’s a good model, not best. But, we have to use test data to prove if the model behaves well with the new data set.

**Test Data: Additional 1000 added and when model trying to predict ->75% which is less than 80%(trained model), since here only less data set, so more jagged lines.**

