> library(ROCR)

> library(sqldf)

> set.seed(1984)

> cc = read.csv("D:/academic-projects/regressao/qconlondon2016\_sample\_data.csv")

> cc <- sqldf("select case when fraudulent = 'True' then

+ 1 else 0 end fraudulent,

+ charge\_time, amount, card\_country, card\_use\_24h

+ from cc")

> #Cria valores de teste e treino

> teste = sample(1:nrow(cc),round(0.3\*nrow(cc)))

> ccteste = cc[teste,]

> cctrain = cc[-teste,]

> str(cctrain)

'data.frame': 62575 obs. of 5 variables:

$ fraudulent : int 0 0 0 0 0 0 0 0 0 0 ...

$ charge\_time : Factor w/ 45586 levels "2015-12-13T11:55:30Z",..: 45586 45586 45586 45586 45586 45586 45586 45586 45586 45586 ...

$ amount : int 20484 1211 8396 1480 535 1632 10305 2783 939 2256 ...

$ card\_country: Factor w/ 3 levels "AU","GB","US": 3 3 3 3 3 3 3 3 3 3 ...

$ card\_use\_24h: int 0 0 1 3 3 0 1 0 0 0 ...

> #Altera vaiaveis para tipo numerico

> ccteste$charge\_time = as.numeric(ccteste$charge\_time)

> cctrain$charge\_time = as.numeric(cctrain$charge\_time)

> cctrain$fraudulent = as.numeric(cctrain$fraudulent)

> table(cc$card\_country)

AU GB US

2145 2754 84494

> modelo = glm(fraudulent~., data=cctrain, family=binomial(link="logit"))

Warning message:

glm.fit: probabilidades ajustadas numericamente 0 ou 1 ocorreu

> summary(modelo)

Call:

glm(formula = fraudulent ~ ., family = binomial(link = "logit"),

data = cctrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-3.9344 -0.0338 -0.0021 0.0000 5.0567

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.852e+02 4.164e+00 44.469 <2e-16 \*\*\*

charge\_time -4.139e-03 9.305e-05 -44.479 <2e-16 \*\*\*

amount 6.851e-07 5.433e-07 1.261 0.207

card\_countryGB -9.187e+00 3.397e-01 -27.048 <2e-16 \*\*\*

card\_countryUS -9.672e+00 2.309e-01 -41.888 <2e-16 \*\*\*

card\_use\_24h 2.424e-03 9.754e-04 2.485 0.013 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 86739.3 on 62574 degrees of freedom

Residual deviance: 4352.6 on 62569 degrees of freedom

AIC: 4364.6

Number of Fisher Scoring iterations: 13

> predict\_teste = predict(modelo, newdata=ccteste, type="response")>0.5

> c\_matrix=table(ccteste$fraudulent,predict\_test)

> c\_matrix=table(ccteste$fraudulent,predict\_teste)

> print(c\_matrix)

predict\_teste

FALSE TRUE

0 13481 51

1 184 13102

> cat('Accuracy: ', sum(diag(c\_matrix))/sum(c\_matrix)\*100, ' %')

Accuracy: 99.12372 %