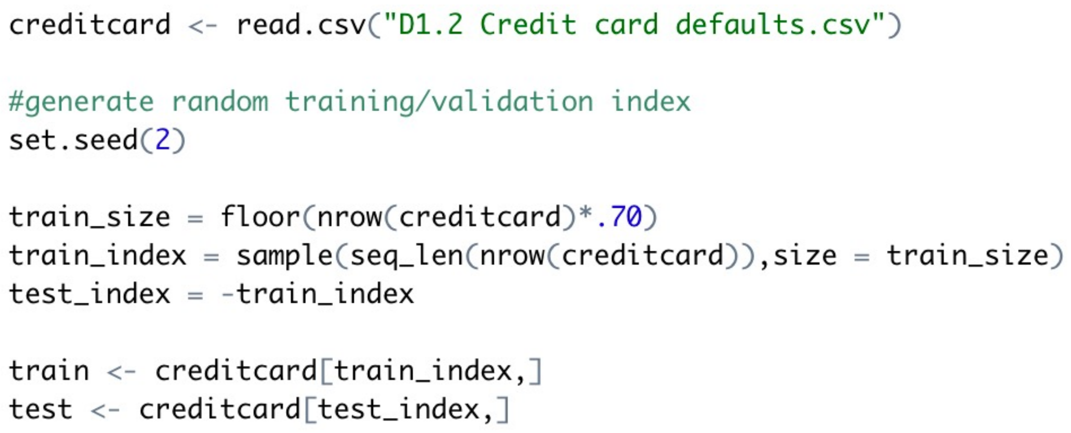
Qiutong Shi Assignment 1

Customer Analytics SP22 Fri 8:30 – 11:30

1. Generate a random training/validation index that implements a 70/30 split

After reading the CSV file into creditcard, tell R I want a training size of 70%, so it samples 70% number of rows in creditcard. The test index would be what is not train index in creditcard.

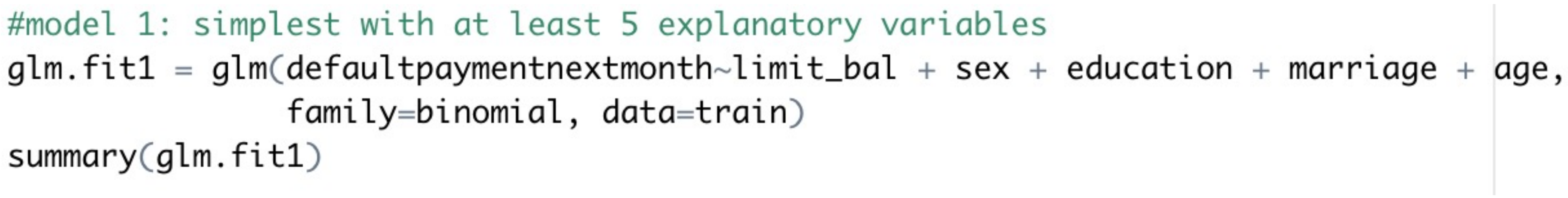
For convenience, the code of train and test set is included.



```
creditcard <- read.csv("D1.2 Credit card defaults.csv")

#generate random training/validation index
set.seed(2)

train_size = floor(nrow(creditcard)*.70)
train_index = sample(seq_len(nrow(creditcard)),size = train_size)
test_index = -train_index

train <- creditcard[train_index,]
test <- creditcard[test_index,]
```

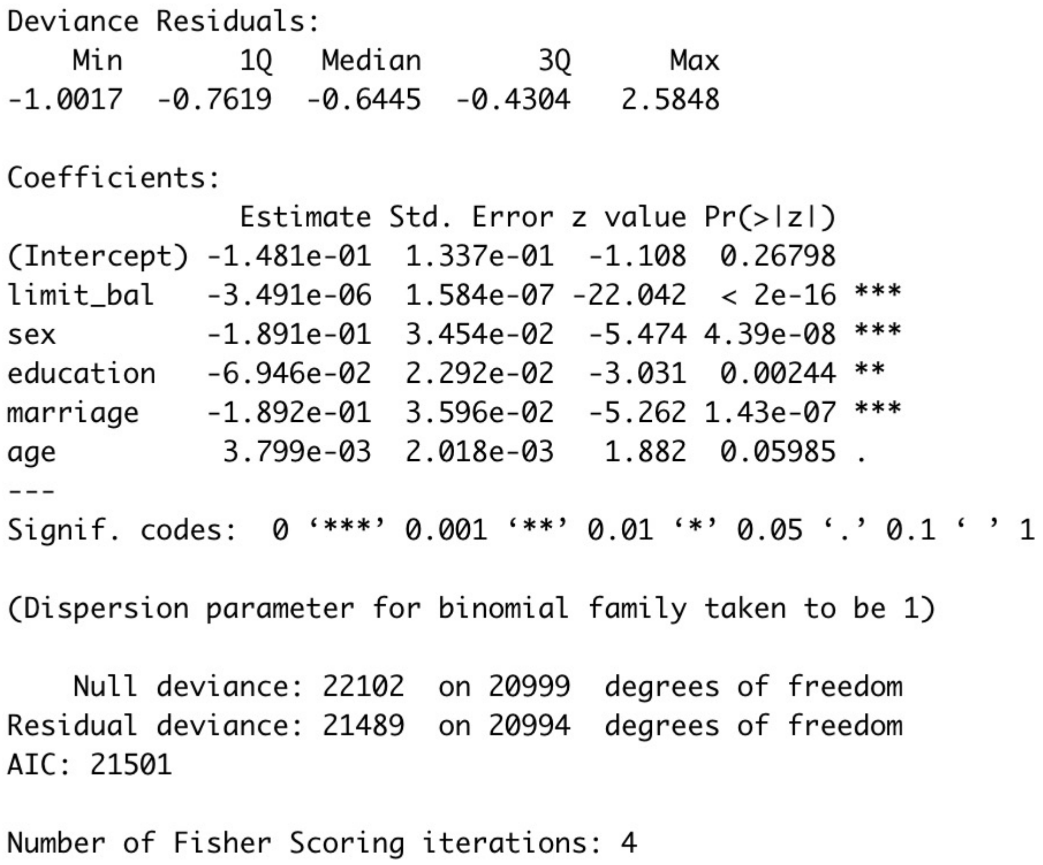
2.1 simple logistic regression: reasoning + code

For the simpler model, I chose limit_bal, sex, education, marriage, age to see if they can tell whether a customer defaulted on his/ her debt. I am selecting these variables for model 1 because I am curious whether ignoring past payment, bill statement and previous payment information can get me a helpful model.



```
#model 1: simplest with at least 5 explanatory variables
glm.fit1 = glm(defaultpaymentnextmonth~limit_bal + sex + education + marriage + age,
               family=binomial, data=train)
summary(glm.fit1)
```

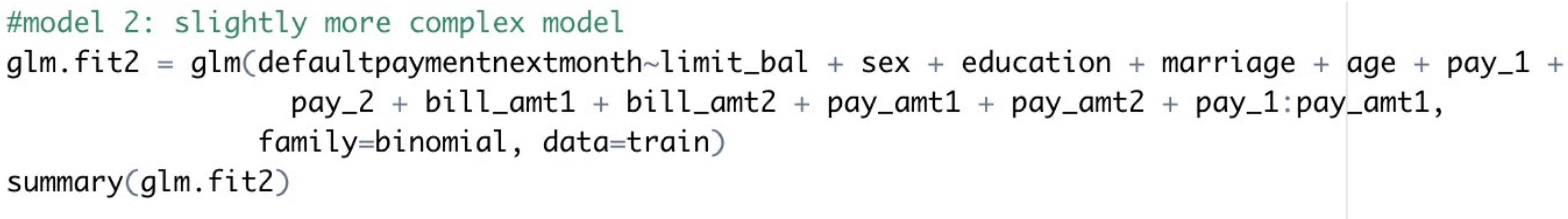
2.1 simple logistic regression: summary of model 1



```
Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.0017  -0.7619  -0.6445  -0.4304   2.5848  

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -1.481e-01  1.337e-01  -1.108  0.26798    
limit_bal   -3.491e-06  1.584e-07 -22.042  < 2e-16 ***
sex         -1.891e-01  3.454e-02  -5.474 4.39e-08 ***
education   -6.946e-02  2.292e-02  -3.031  0.00244 ** 
marriage    -1.892e-01  3.596e-02  -5.262 1.43e-07 ***
age          3.799e-03  2.018e-03   1.882  0.05985 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22102  on 20999  degrees of freedom
Residual deviance: 21489  on 20994  degrees of freedom
AIC: 21501

Number of Fisher Scoring iterations: 4
```

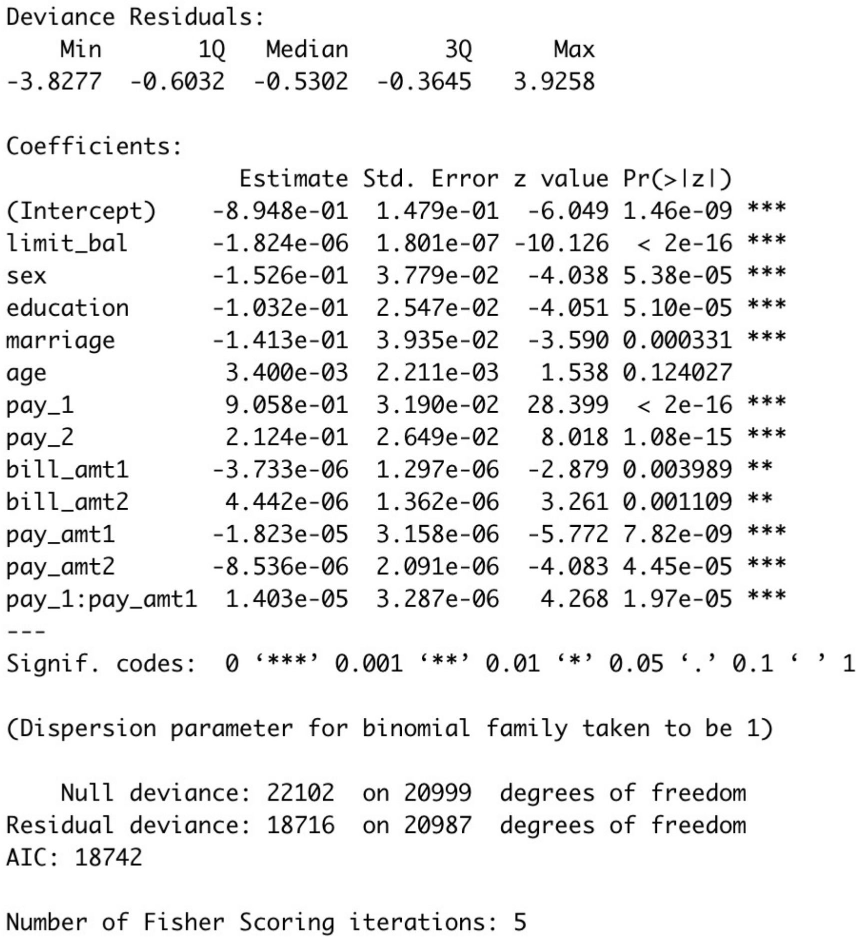
2.2 “complex” logistic regression: reasoning + code

For model 2, I am adding variables bill_amt and pay_amt 1 and 2 to see if adding recent financial circumstances help improve the model. I am also adding history of past payment in the recent 2 months. Since I suspect there could be a relationship between amount of the most recent previous statement(circumstance) and the most recent history of past payment(activity), I explicitly add an interaction term between these two variables.



```
#model 2: slightly more complex model
glm.fit2 = glm(defaultpaymentnextmonth~limit_bal + sex + education + marriage + age + pay_1 +
                 pay_2 + bill_amt1 + bill_amt2 + pay_amt1 + pay_amt2 + pay_1:pay_amt1,
               family=binomial, data=train)
summary(glm.fit2)
```

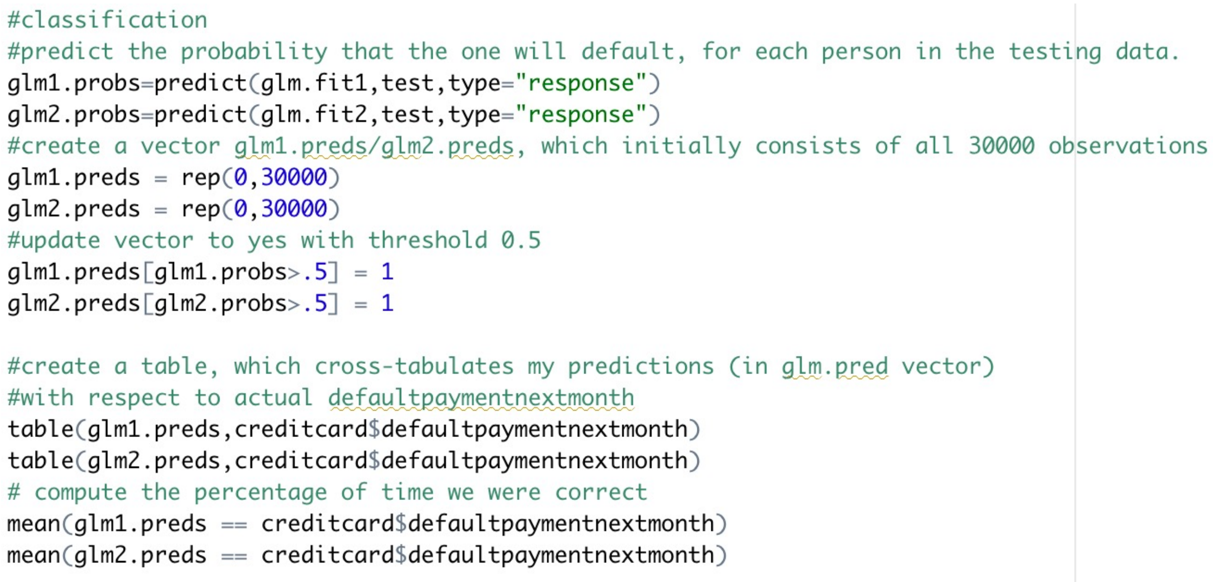
2.2 “complex” logistic regression: summary of model 2



```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.8277  -0.6032  -0.5302  -0.3645   3.9258

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -8.948e-01  1.479e-01  -6.049 1.46e-09 ***
limit_bal      -1.824e-06  1.801e-07 -10.126  < 2e-16 ***
sex            -1.526e-01  3.779e-02  -4.038 5.38e-05 ***
education      -1.032e-01  2.547e-02  -4.051 5.10e-05 ***
marriage       -1.413e-01  3.935e-02  -3.590 0.000331 ***
age             3.400e-03  2.211e-03   1.538 0.124027
pay_1           9.058e-01  3.190e-02  28.399  < 2e-16 ***
pay_2           2.124e-01  2.649e-02   8.018 1.08e-15 ***
bill_amt1      -3.733e-06  1.297e-06  -2.879 0.003989 **
bill_amt2       4.442e-06  1.362e-06   3.261 0.001109 **
pay_amt1       -1.823e-05  3.158e-06  -5.772 7.82e-09 ***
pay_amt2       -8.536e-06  2.091e-06  -4.083 4.45e-05 ***
pay_1:pay_amt1  1.403e-05  3.287e-06   4.268 1.97e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22102  on 20999  degrees of freedom
Residual deviance: 18716  on 20987  degrees of freedom
AIC: 18742

Number of Fisher Scoring iterations: 5
```

3. Do any of your models exhibit signs of overfitting?

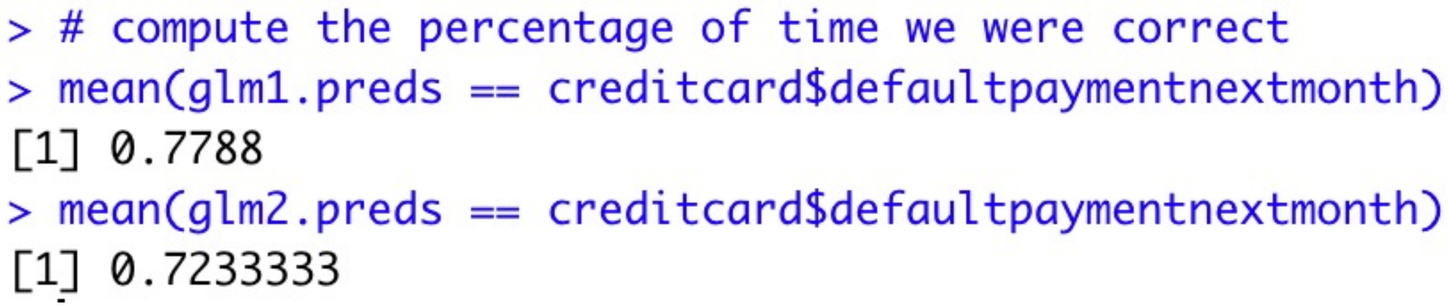
I decide to define a model as overfitting if it fails to successfully predict defaultpaymentnextmonth by looking at the confusion matrix statistics (which I coded the table and mean explcitly due to a failure in installing package “caret”)



```
#classification
#predict the probability that the one will default, for each person in the testing data.
glm1.probs=predict(glm.fit1,test,type="response")
glm2.probs=predict(glm.fit2,test,type="response")
#create a vector glm1.preds/glm2.preds, which initially consists of all 30000 observations
glm1.preds = rep(0,30000)
glm2.preds = rep(0,30000)
#update vector to yes with threshold 0.5
glm1.preds[glm1.probs>.5] = 1
glm2.preds[glm2.probs>.5] = 1

#create a table, which cross-tabulates my predictions (in glm.pred vector)
#with respect to actual defaultpaymentnextmonth
table(glm1.preds,creditcard$defaultpaymentnextmonth)
table(glm2.preds,creditcard$defaultpaymentnextmonth)
# compute the percentage of time we were correct
mean(glm1.preds == creditcard$defaultpaymentnextmonth)
mean(glm2.preds == creditcard$defaultpaymentnextmonth)
```

3. Do any of your models exhibit signs of overfitting?

I would say both of my models exhibit signs of overfitting as the accuracy of predictions are both below 80%, indicating that the models are failing to predict correctly with the test data. The more flexible model is having more trouble with overfitting than the simpler one. The accuracy of model 1 is 0.7788; the accuracy of model 2 is 0.7233333.

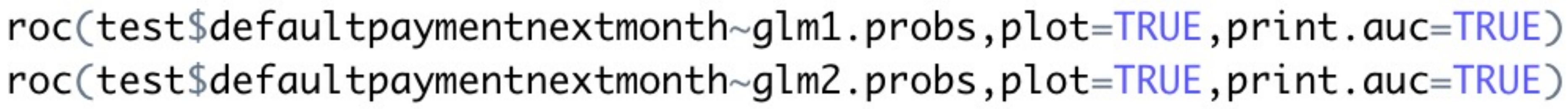


```
> # compute the percentage of time we were correct
> mean(glm1.preds == creditcard$defaultpaymentnextmonth)
[1] 0.7788
> mean(glm2.preds == creditcard$defaultpaymentnextmonth)
[1] 0.7233333
```

4. Provide a discussion of which of the two models you would prefer for the purpose of identifying consumers who will default in the future.

I would prefer model 2 for identifying despite it is slightly less accurate in testing (problem 3). The main reason is that model 2 has a smaller AIC and a smaller residual deviance than model 1, which suggests that model 2 also explains the data better.

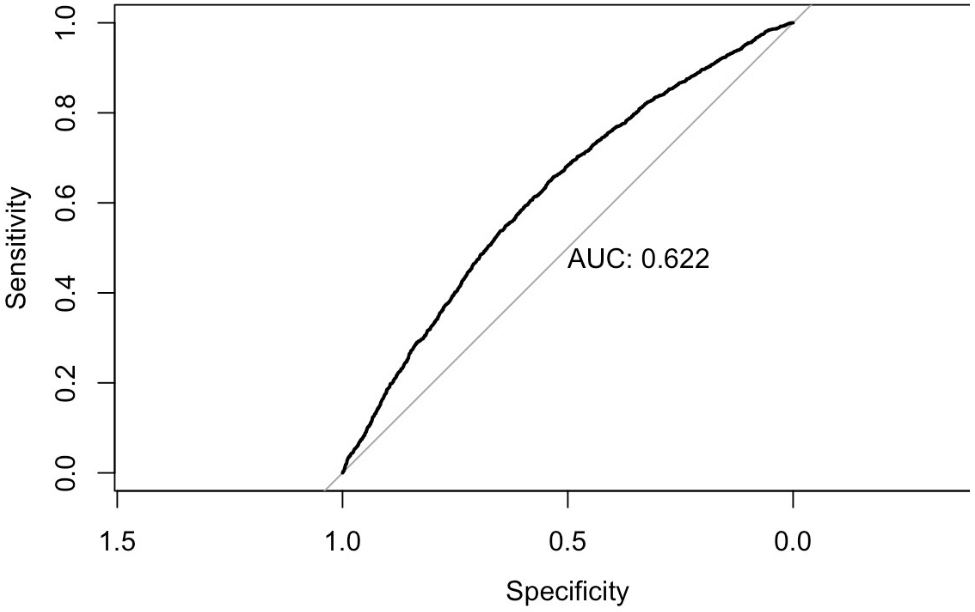
Meanwhile, I also want to the AUCs of the testing output of the two models.

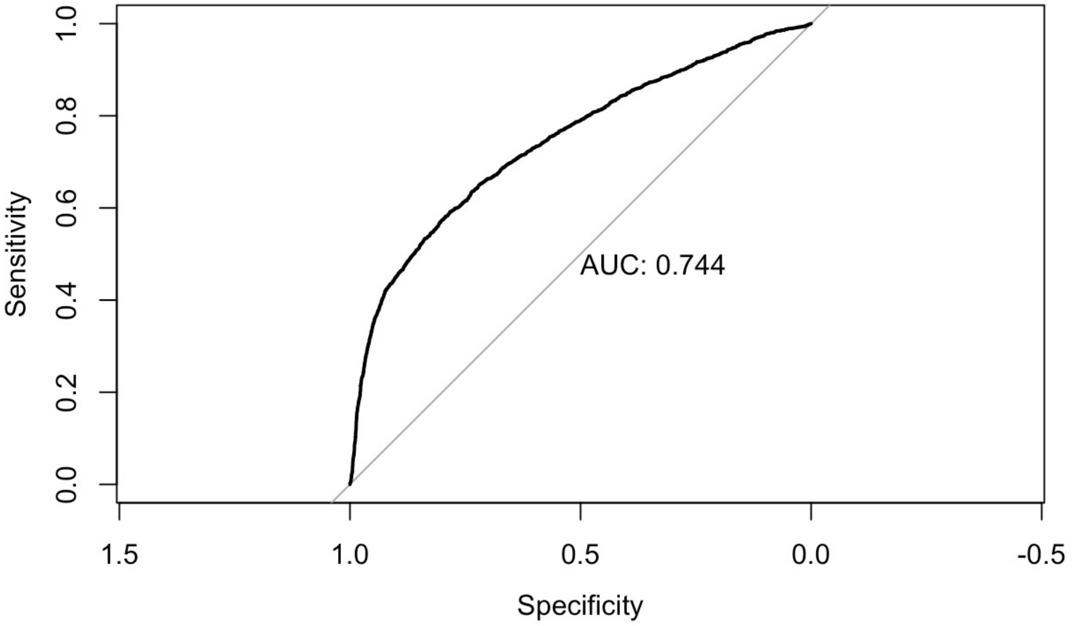


```
roc(test$defaultpaymentnextmonth~glm1.probs,plot=TRUE,print.auc=TRUE)
roc(test$defaultpaymentnextmonth~glm2.probs,plot=TRUE,print.auc=TRUE)
```

4. Provide a discussion of which of the two models you would prefer for the purpose of identifying consumers who will default in the future.

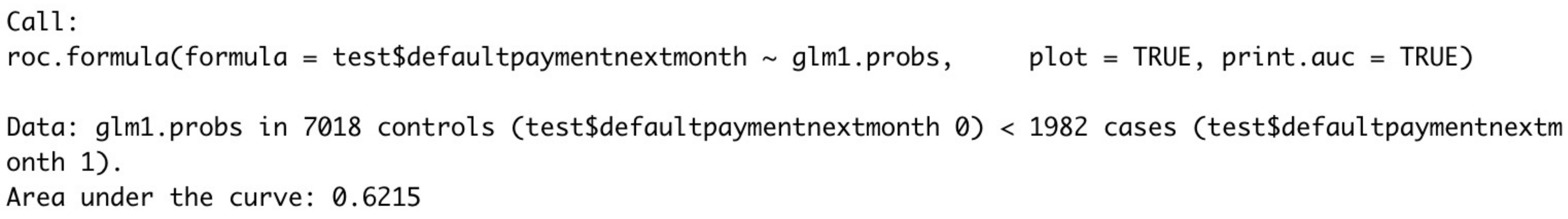
The AUC of model 2(right-hand figure) is slightly larger, still making model 2 more preferable.



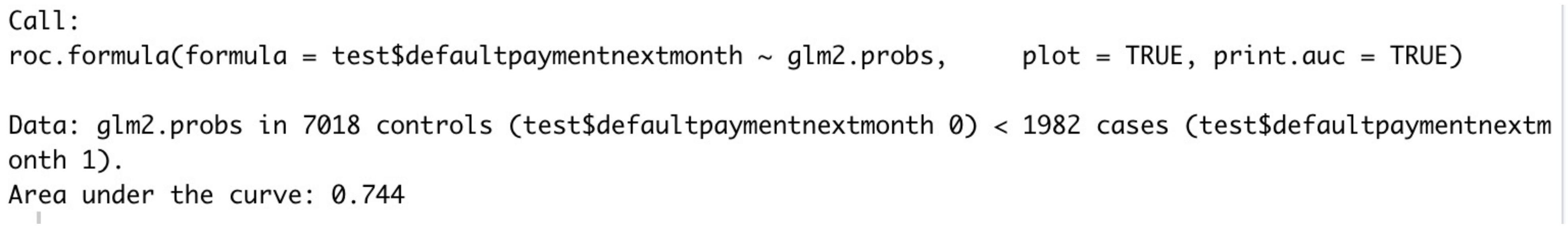


4. Provide a discussion of which of the two models you would prefer for the purpose of identifying consumers who will default in the future.

Here are the details of output in R. Model 2 is the one below.



```
Call:
roc.formula(formula = test$defaultpaymentnextmonth ~ glm1.probs,     plot = TRUE, print.auc = TRUE)

Data: glm1.probs in 7018 controls (test$defaultpaymentnextmonth 0) < 1982 cases (test$defaultpaymentnextm
onth 1).
Area under the curve: 0.6215
```



```
Call:
roc.formula(formula = test$defaultpaymentnextmonth ~ glm2.probs,     plot = TRUE, print.auc = TRUE)

Data: glm2.probs in 7018 controls (test$defaultpaymentnextmonth 0) < 1982 cases (test$defaultpaymentnextm
onth 1).
Area under the curve: 0.744
```