Credit Card Default and Risk Scores with Logistic Regression

San Diego State University

December 12, 2017

Abstract

In this paper, we obtain simulated credit card default data from the `ISLR` package in `R`. The data contains information on a dichotomous response, credit card default, and the goal is to obtain a logistic regression model that accurately predicts the classification of an individual. The paper obtains two logistic regression models, one with a covariate indicating if an individual is a student and one without this covariate. We explore these models and compare them with AIC, an area under the precision-recall ROC curve, and precision-recall values based on optimized cutoff values. The final model contains only two covariates, the balance of the loan and income of the individual. We interpret the final model in terms of odds ratios and obtain confidence intervals for the values obtained. Finally, based on this final model we build a Credit Risk Score that ranges from 300 to 850.

I. Introduction

Credit card default is an issue among companies in the financial industry. If payment default on loans can be detected early enough, we may be able to set financial precautions to maximize revenue. We would like to assess the risk of an individual defaulting on their loans based on their loan balance, income, and whether the individual is a student. Therefore, we construct a logistic regression model to predict whether or not an individual will default on their loan. The data obtained is the *Default* dataset contained in the `ISLR` package in `R`. The data contains 1000 observations and 4 attributes. There are no missing observations within this dataset. In the paper, we consider two logistic regression models. The first logistic regression model includes only the covariates `balance` and `income`. The second logistic regression model includes all of the covariates `balance`, `income` and `student`. We do not consider any transformations or interactions in this paper. From these two models, we choose the one that fits the data the sufficiently well based on AIC, the area under the precision-recall ROC curve, and misclassification rate, while maintaining simplicity. We also obtain confidence intervals for the odds ratios associated with the final model. Lastly, we present a function that assesses the risk of an individual called `Risk-Score`. This score is a scale from 300 to 850.

II. Methods

The data contains simulated information on whether or not an individual has defaulted on their credit card loan payments. The covariates in the dataset are the balance of the loan, the income of the individual, and whether the individual is a student, respectively. In this data set, 333 of 1000 individuals defaulted and stopped paying their loans. As seen in red in Table 1, this is only about 3% of the total population. The data also contains information on the balance and income of these individuals. Table 1 also indicates that the mean balance of the loan is about $835. However for individuals who have defaulted, this average increases to 1747. Likewise, the mean income decreases slightly for those who defaulted on their loans. From the sample data, 29 % of the data contains students. Of those who defaulted, 38% of those individuals are students.

Continuous	*Mean*	*Standard Dev.*	*Min*	*Max*	*Mean Default*
Balance	835	483	0	2654	1747
Income	33517	13336	772	73554	32089
Categorical	*Prop. of Total*	*Standard Dev.*	*Total Postive*	*Total Obs.*	*Prop. Total Default*
Student:					
Yes	0.29	0.46	2944	10000	0.38
Default:					
Yes	0.03	0.179	333	10000	1

Table 1: *This table indicates the summary statistics for the data. The first column is the mean or the proportion of the total. The second column is the standard deviation. The third column is the min for continuous variables and total positive observations for categorical variables. The fourth column is the max for continuous variables and a total number of observations for categorical variables. The last column is the mean or proportion of those who defaulted classifying as the specific covariate.*

The methodology of the data analysis begins with splitting the data 50/50 between training and testing data sets. We then conduct exploratory data analysis, then build two models based on the training set. We then obtain AIC, AUC and cutoff values from the model based on the testing set. We then explore the model 3 times with different seed values and obtain average AIC, AUC, and cutoff values. We finally build the final model selected with 10-fold cross-validation and obtain a cross-validated error rate.

III. Results

i. Exploratory Data Analysis

The response variable, `default` is a binary variable. The default rate of this data set is approximately 3%. Figure 1 contains boxplots of the balance of the loan and the income of the individual. The first balance boxplot indicates the balance over all individuals, and the second balance indicates the balance overall individuals who are also classified as students. Not only does a higher balance indicate individuals are more likely to default, but it also indicates that students, in general, have higher balances among the default groups. The boxplot of income for all individuals indicates no clear differences among those who defaulted and those who did not. There is a clear difference in income of those who are students but not the levels of default.

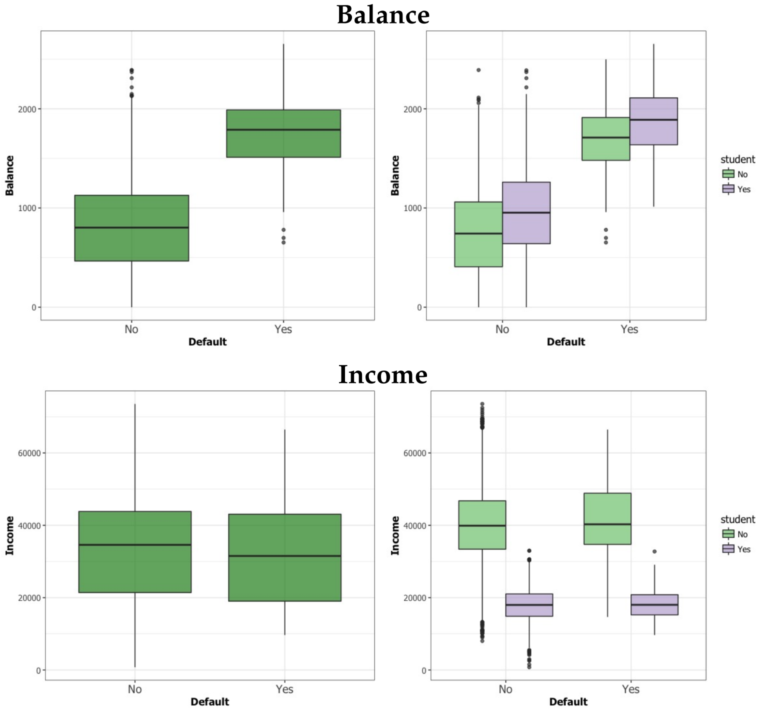


Figure 1: *This figure indicates two pairs of boxplots of balance and income. The first boxplot column is that all individuals regardless of student status, while the second boxplot column considers it is a student.*

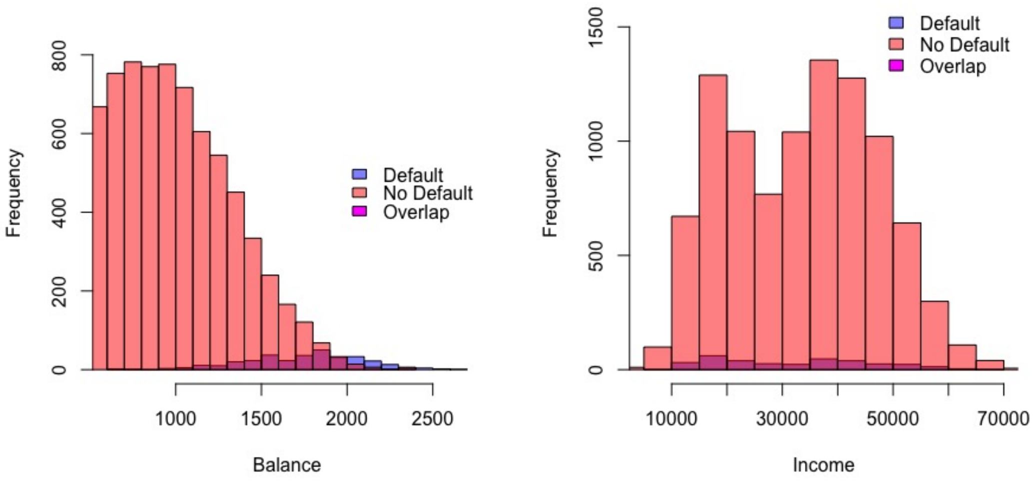


Figure 2: *This figure indicates the histograms of balance and income. Both of these histograms show those individuals who defaulted on their loans in blue, those who did not in red, and the overlap of the histogram in purple.*

Figure 2 contains histograms of balance and income, separated by those who defaulted. These histograms are consistent with the boxplots of the data. However, we notice that the data for balance is skewed left, but those who defaulted seem to have a location shift above those who did not default. The frequency is also consistent with the fact that only 3% of the data contains individuals who defaulted. The second histogram indicates the overall income of all individuals. As also seen in the boxplot, there does not seem to be any mean differences in the income of those who defaulted and those who did not. The variance-covariance plot found in the appendix indicates little correlation between balance and income.

ii. Model Selection

We begin by building a logistic regression model with all covariates. We then build another model with all covariates excluding student status. We compare these two models with AIC and the area under the precision-recall ROC curve. In Table 2, we obtain the average AIC, Area under the precision-recall ROC curve, and cutoff values for both models. We also obtain a contingency table proportion to the total number of observations. The averages are over three models trained with different seeds and are obtained with the testing set.

Based on this table, there does not indicate to be any statistically significant differences among the two models. The first model obtains an AIC value of 740, and the second model obtains an AIC value one point lower. However, the cutoff value and the area under the precision-recall ROC curve obtain the exact same values rounded to three and two decimal places respectively. Based on the first training set models, the contingency tables do not indicant any differences as well. Therefore, we choose the model with only the covariates balance and income as our final model.

Avg	Covariates Included	AIC	Cutoff	AUC	Contingency	(seed 1)	
Model 1	Balance	740	0.035	0.94		True	False
	Income				Pred. True	0.0298	0.1244
					Pred. False	0.0036	0.8422
Model 2	Balance	739	0.035	0.94		True	False
	Income				Pred. True	0.0298	0.1244
	Student				Pred. False	0.0036	0.8422

Table 2: *This table indicates the average AIC, AUC, cutoff values for three different models trained on randomly selected data points. The contingency tables and other values are that of the testing set.*

iii. Model Inferences

Based on the final model, we obtain Table 3, we notice immediately that both covariates are significant. The point estimates are very close to zero, and the odds ratios are very close to one. However, since the sample size of our data is $10,000$, the estimates are statistically significant.

	Coefficient	OR	SE(OR)	P-value	95% C.I. OR
(Intercept)	-12.080	0.00	0.00	< 0.0001	$(0.00000, 0.00001)$
Balance	0.006	1.01	0.00	< 0.0001	$(1.00542, 1.00679)$
Income	0.000	1.00	0.00	0.00123	$(1.00000, 1.00003)$

Table 3: *This table indicates the model coefficients, the odds ratios, the standard error of the odds ratios, the p-value of the test performed that $H_0 : OR = 1$ vs $H_0 : OR \neq 1$, and a confidence interval for the odds ratio.*

In Table 3, we can interpret the odds ratio of the main effect coefficients. For every \$100 increase in balance, there is an expected 83% increase in odds of default, assuming income is fixed. Likewise, for every \$1000 increase in income, there is an expected 1.9% increase in odds of default, assuming the default is fixed.

iv. Credit Risk Score

Figure 3 indicates a 3 Dimensional plot of Credit Risk Score as a function of income and balance. Credit Risk Score ranges from 300 to 850, with higher score values indicating a lower risk of default. As income ranges from 10000 to 70000, the Risk Score is fairly consistent. However, when the balance of the loan increases from 0 to 2654, we notice the Risk Score drops dramatically. This indicates that those individuals with a lower score are more likely to default on their credit card loan payments.

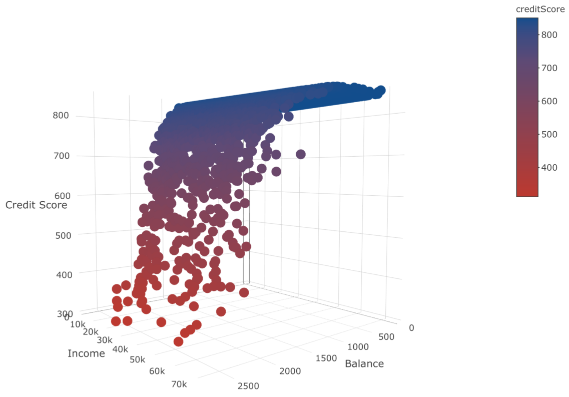


Figure 3: *A 3D scatter plot with Credit Risk Score as a function of the loan balance and individual income.*

IV. Discussion

Based on the simulated credit card default data set, we are able to accurately classify about 87% of the testing data. This does not deviate much from sample to sample. An overall conclusion is that student status does not supply much information to the outcome of default. This is possible because students are primarily either in deferment or on a grace period while in school. The biggest contributing factor to the outcome of default is the loan balance. Overwhelmingly, a higher balance increased the odds of default over a lower balance. This effect is seen as an expected 83% increase in odds of default for every $100 increase in balance.

The effect for income is more subtle, however, there is still a small positive association. The positive association is surprising, as any thought experiment would conclude that a higher increase in income would decrease the odds of default, not increase. However, based on the estimates we conclude that a $1000 increase in annual income would only raise the expected odds of default by 1.9%. Or put another way, for every $10000 increase in annual income, the expected odds of default increase by 20%. This is possible because those individuals who attain more wealth may have determined that they could afford to take out a higher loan amount, biting off more than they can chew.

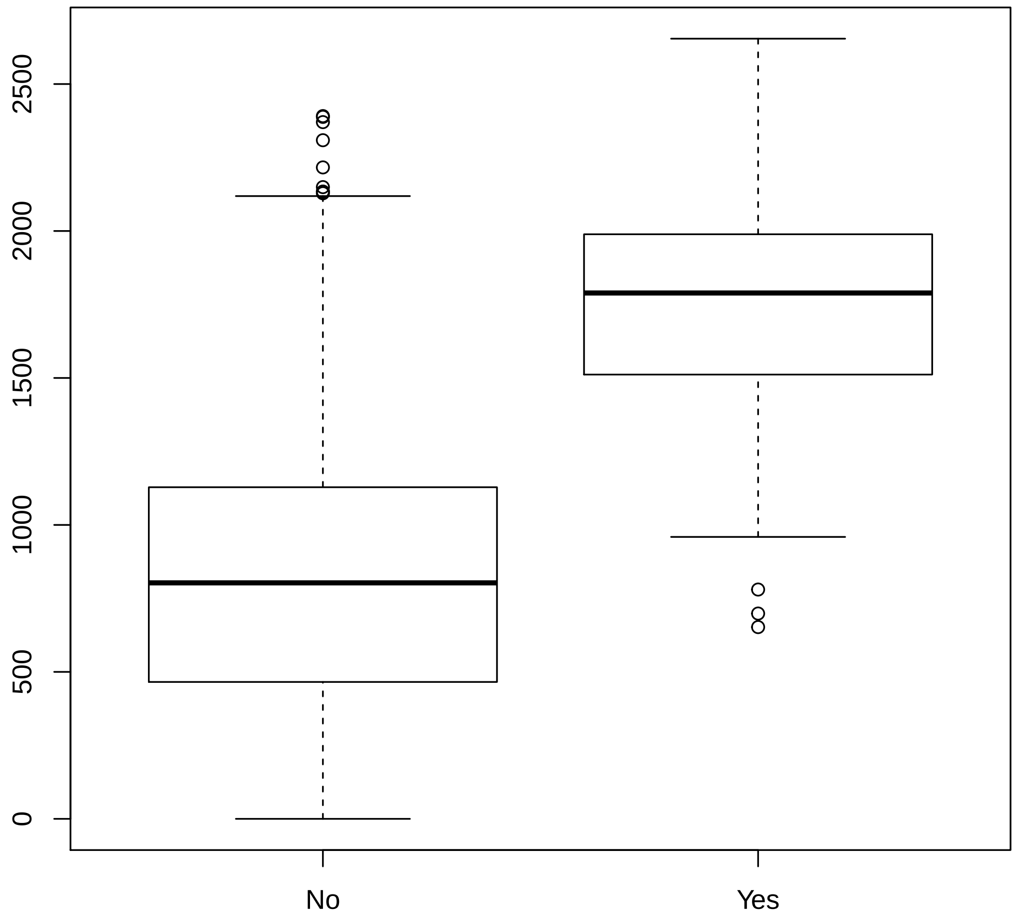
The first limitation of the analysis a lack of more variables. With only three covariates, the analysis is very limited. Although we do obtain decent model estimates, there is no doubt that the model could be improved if there were more covariates. The second limitation is a lack of error balance. We did not consider the fact that a *false negative* is more burdensome than a *false positive*. Therefore, we did not choose thresholds to balance the two in favor of the loan servicer. We only considered a balance that simultaneously optimizes sensitivity and specificity.

Based on this data analysis, further research may include looking into different covariates and considering interactions among covariates. We could also possibly look into the effects of the risk credit score on those individuals who were falsely determined to default, and what impacts on the loan servicer this score has.

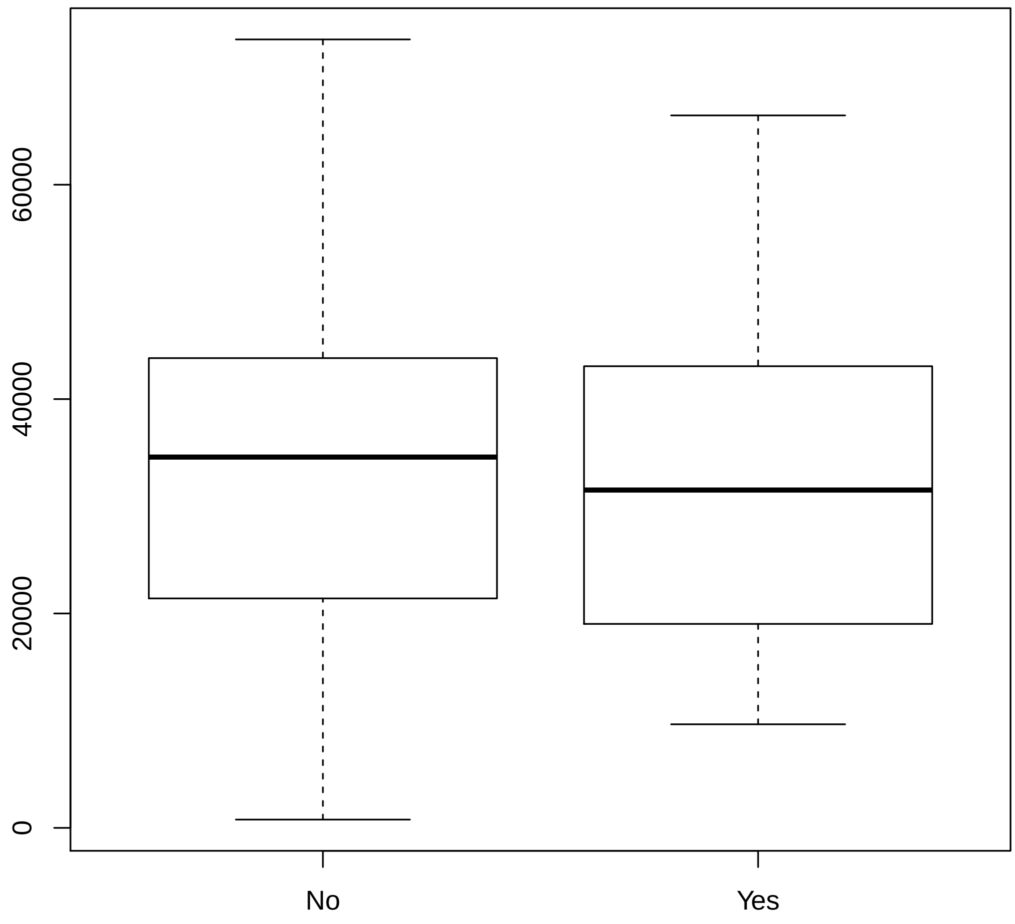
Appendix A

```
################################################################################
##################################### Packages #################################
################################################################################


library(ISLR)
library(ROCR)

## Loading required package:  gplots
##
## Attaching package:  'gplots'
## The following object is masked from 'package:stats':
##
##     lowess

library(boot) # for cross-validation
library(corrplot)

## corrplot 0.84 loaded

library(scatterplot3d)
library(plotly)

## Loading required package:  ggplot2
##
## Attaching package:  'plotly'
## The following object is masked from 'package:ggplot2':
##
##     last_plot
## The following object is masked from 'package:stats':
##
##     filter
## The following object is masked from 'package:graphics':
##
##     layout

library(tidyverse)

## Loading tidyverse:  tibble
## Loading tidyverse:  tidyr
## Loading tidyverse:  readr
## Loading tidyverse:  purrr
## Loading tidyverse:  dplyr
## Conflicts with tidy packages -----------------------
## filter():  dplyr, plotly, stats
## lag():    dplyr, stats

################################################################################
#################################### Functions #################################
################################################################################
rocplot=function(pred, truth, ...){
  predob = prediction (pred, truth)
  perf = performance (predob , "tpr", "fpr")
  plot(perf ,...)}


opt.cut = function(perf, pred){
  cut.ind = mapply(FUN=function(x, y, p){
    d = (x - 0)^2 + (y-1)^2
    ind = which(d == min(d))
    c(sensitivity = y[[ind]], specificity = 1-x[[ind]],
      cutoff = p[[ind]])
  }, perf@x.values, perf@y.values, pred@cutoffs)
}

# Function that will compute sensitivity and specificity at any given cutoff
se.sp <- function (cutoff, pred){
  sens <- performance(pred,"sens")
  spec <- performance(pred,"spec")
  num.cutoff <- which.min(abs(sens@x.values[[1]] - cutoff))
  return(list(Cutoff=sens@x.values[[1]][num.cutoff],
              Sensitivity=sens@y.values[[1]][num.cutoff],
              Specificity=spec@y.values[[1]][num.cutoff]))
}

risk.score <- function(fit.logistic, data){ # 300 to 850
  pred <- predict(fit.logistic, newdata = data, type = "response") #prob default

  #Risk score of default: Use Credit score
  p.star <- 1-pred
  return((p.star*550) + 300)
} #Risk/ Credit score
################################################################################
#################################### Data #################################
################################################################################
# This assignment is based on ISLR Chapter 5, exercise 5
head(Default)

##   default student   balance    income
## 1      No      No  729.5265 44361.625
## 2      No     Yes  817.1804 12106.135
## 3      No      No 1073.5492 31767.139
## 4      No      No  529.2506 35704.494
## 5      No      No  785.6559 38463.496
## 6      No     Yes  919.5885  7491.559

dim(Default)

## [1] 10000     4

n = dim(Default)[1]   # sample size
sum(is.na(Default))

## [1] 0

names(Default)
```

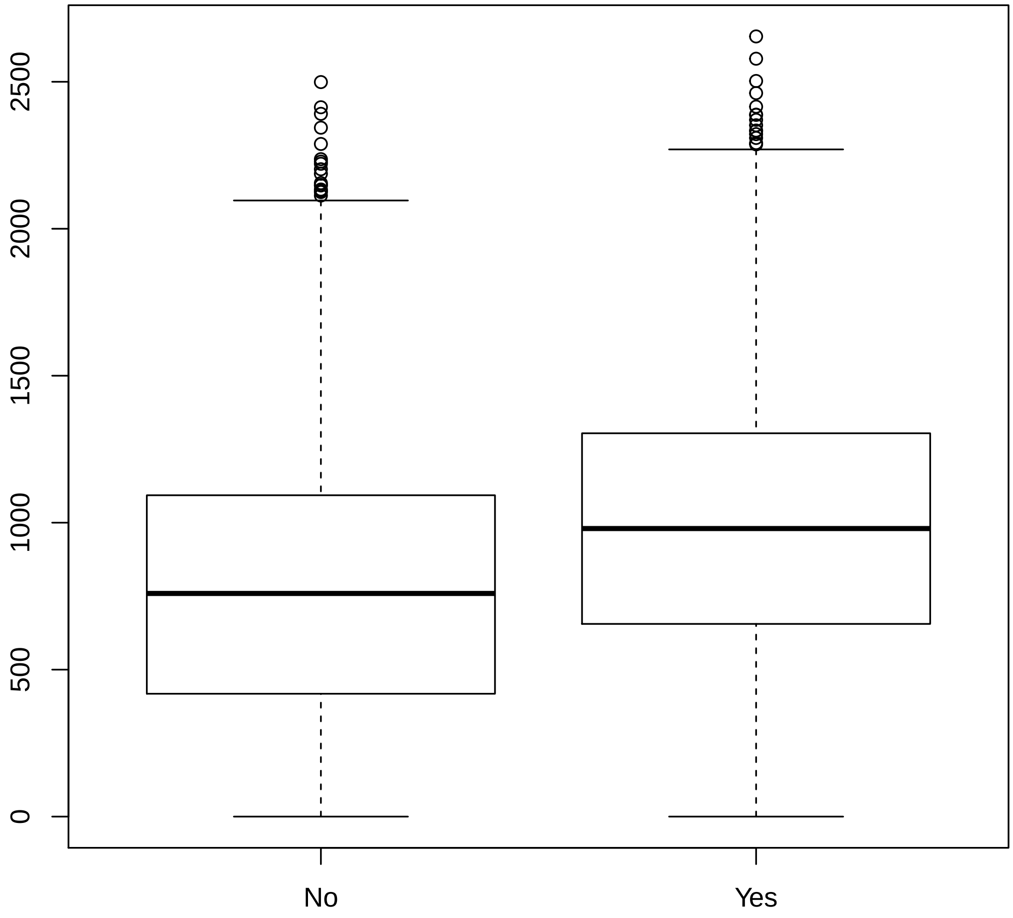
```
## [1] "default" "student" "balance" "income"

attach(Default)
#########
# Task 1, EDA: response is 'default'
# Summarize response and relationship between response and the three explanatory variables.
#########

boxplot(balance~default, data = Default) #there is a difference in means
```



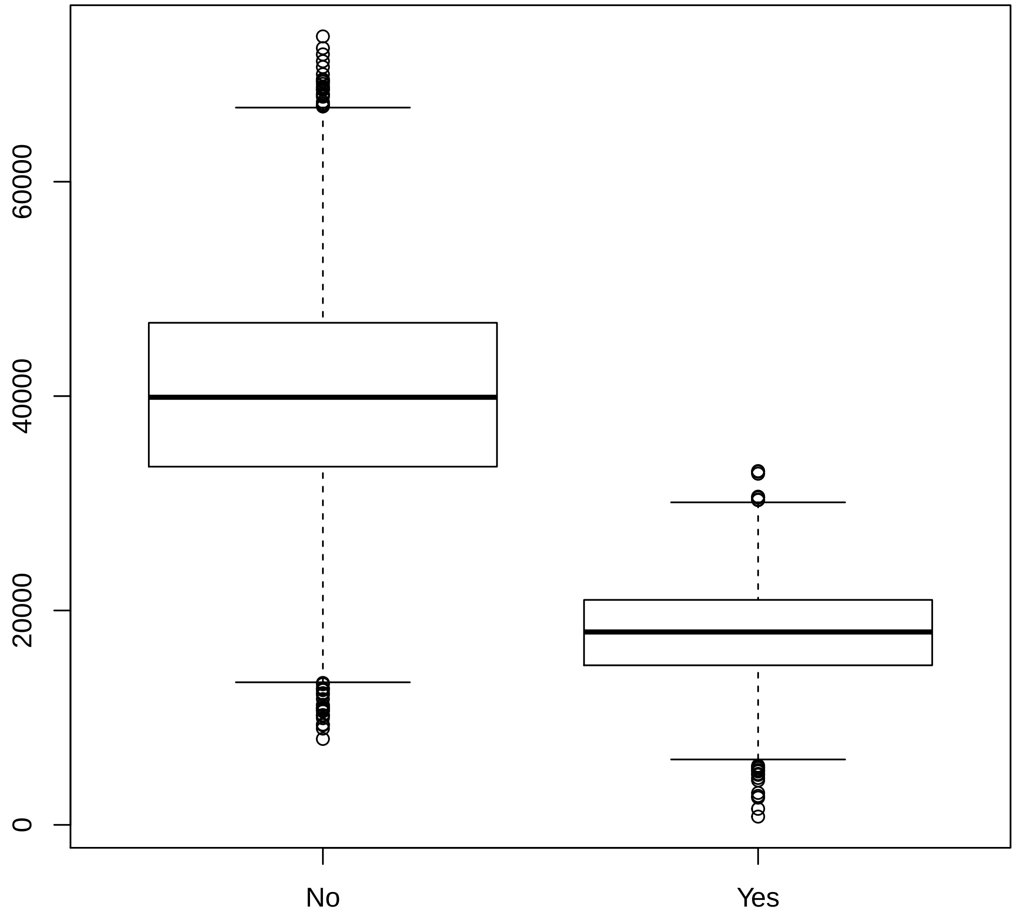
```
boxplot(income~default, data = Default) #There is not a difference in means
```



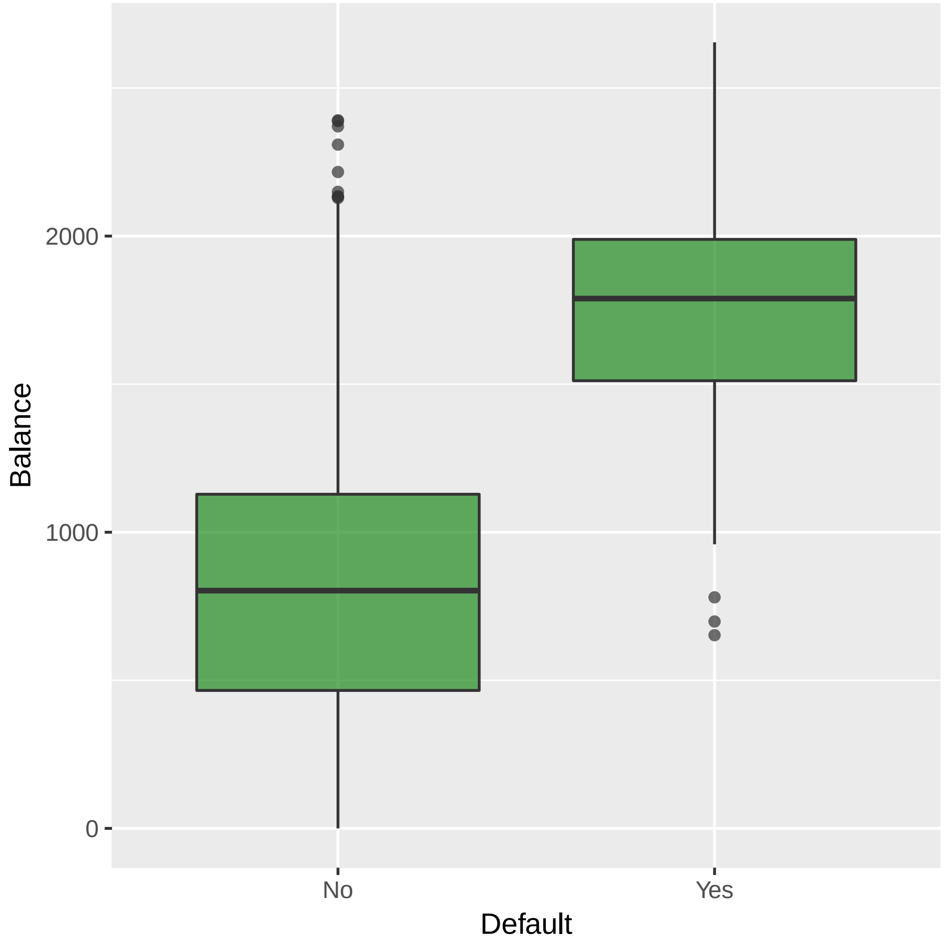
```
boxplot(balance~student, data = Default) #No  differences in mean
```



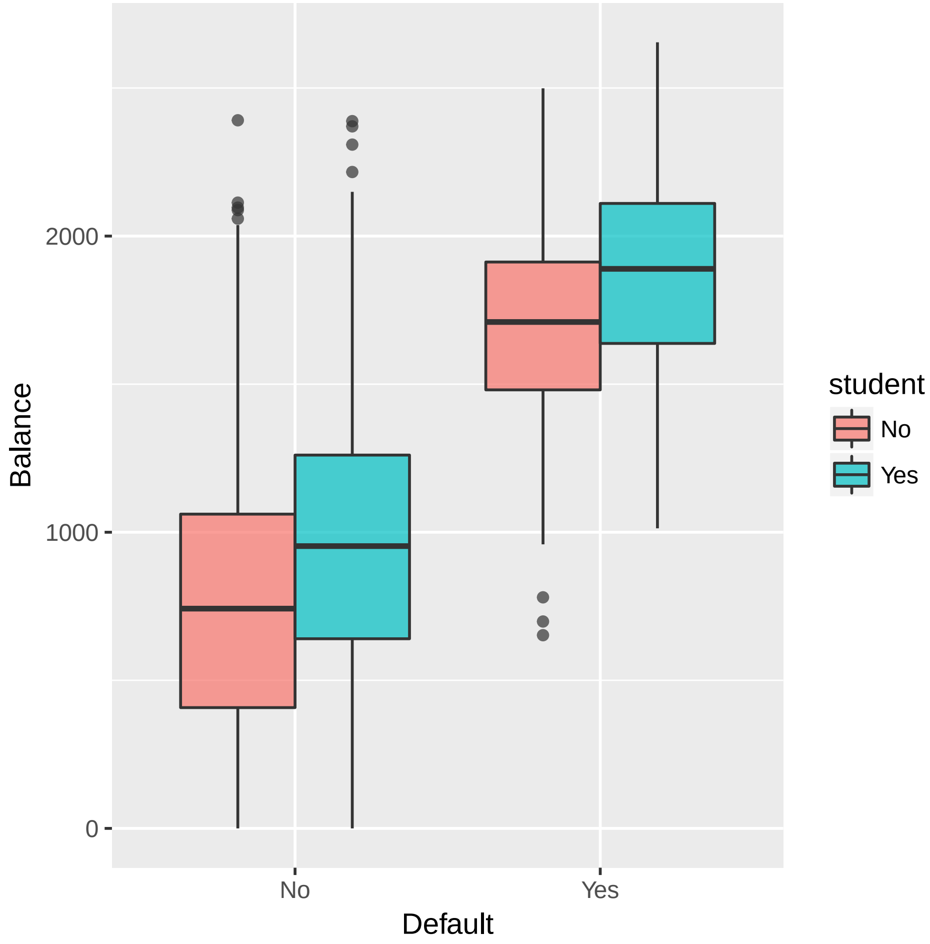
```
boxplot(income~student, data = Default) #differences in mean
```



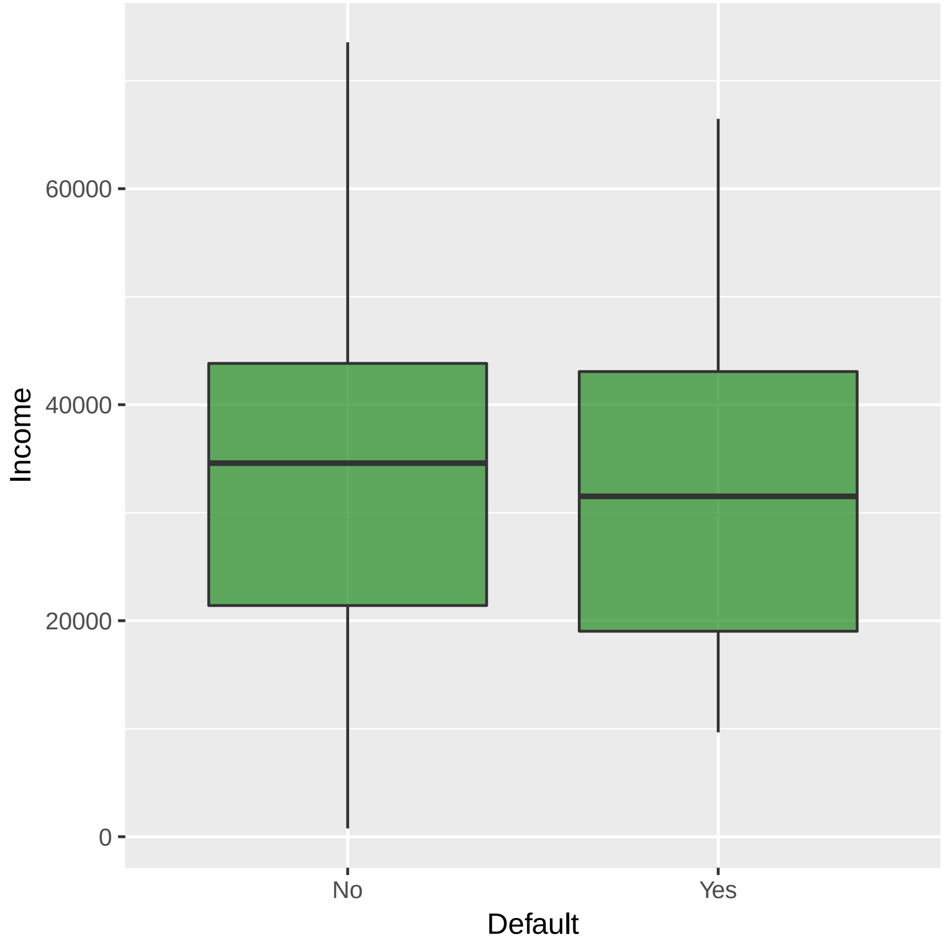
```
ggplot(Default, aes(x = default, y = balance)) +
  geom_boxplot(alpha=0.7, fill = "forestgreen") +
  scale_y_continuous(name = "Balance") +
  scale_x_discrete(name = "Default")
```



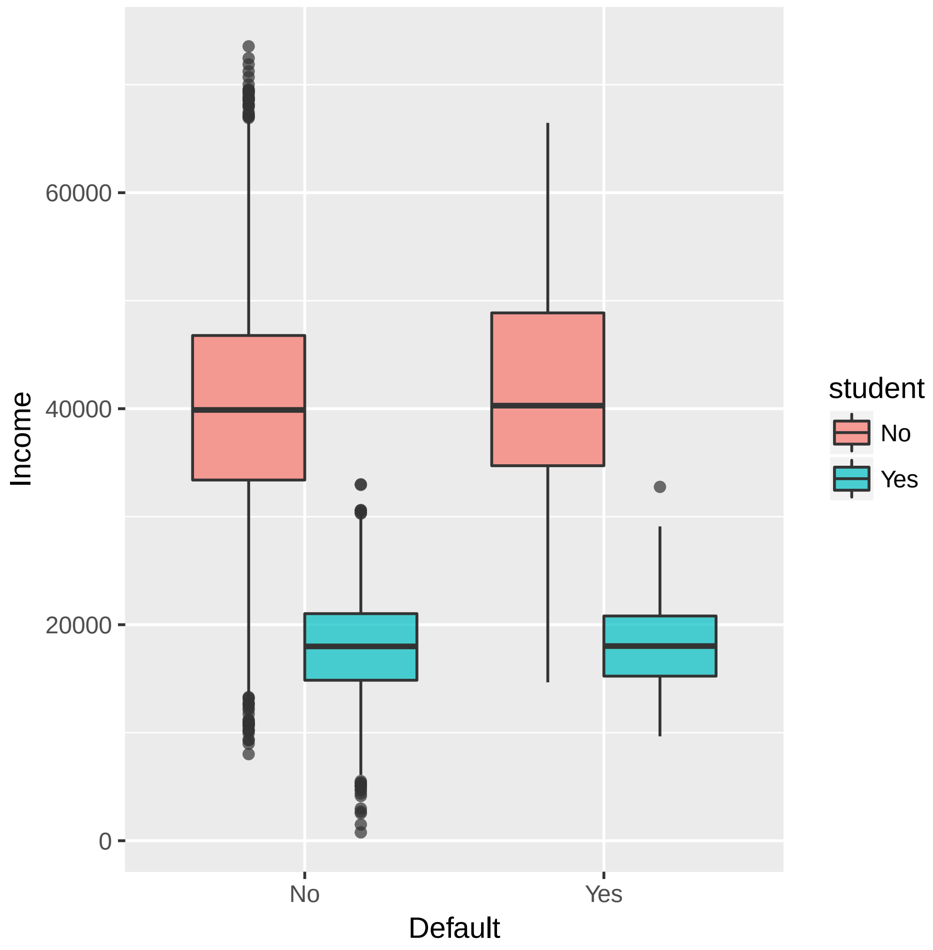
```
ggplot(Default, aes(x = default, y = balance, fill = student)) +
  geom_boxplot(alpha=0.7) +
  scale_y_continuous(name = "Balance") +
  scale_x_discrete(name = "Default")
```



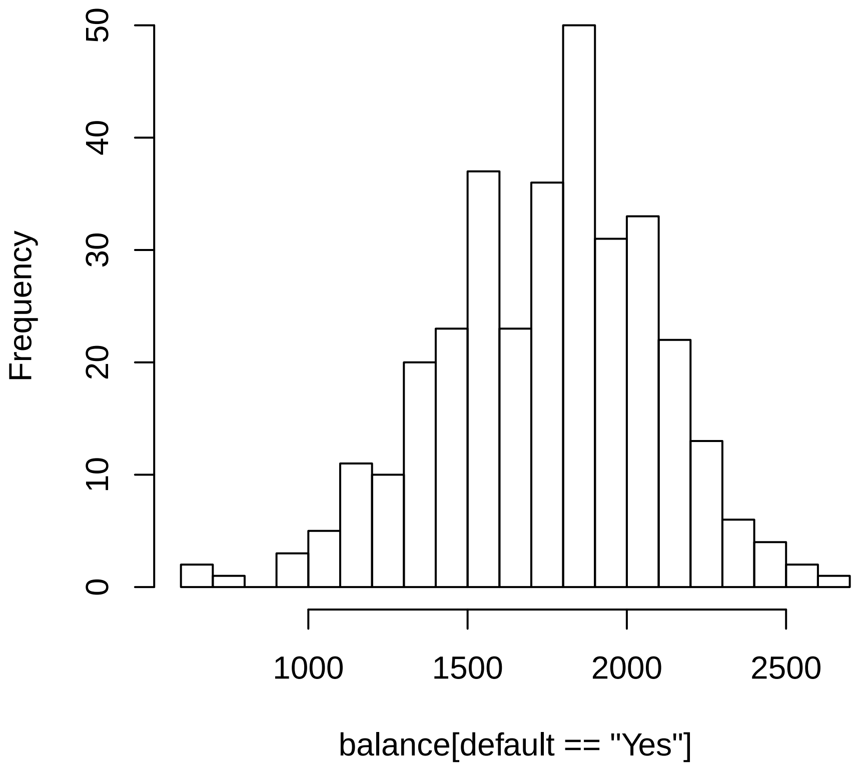
```
ggplot(Default, aes(x = default, y = income)) +
  geom_boxplot(alpha=0.7, fill = "forestgreen") +
  scale_y_continuous(name = "Income") +
  scale_x_discrete(name = "Default")
```



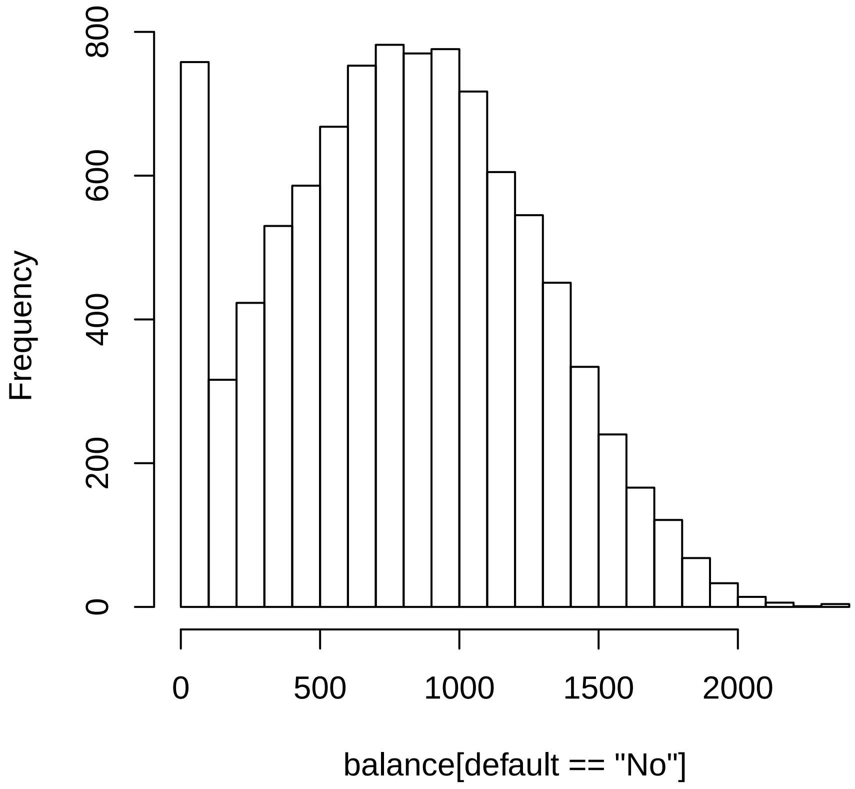
```
ggplot(Default, aes(x = default, y = income, fill = student)) +
  geom_boxplot(alpha=0.7) +
  scale_y_continuous(name = "Income") +
  scale_x_discrete(name = "Default")
```



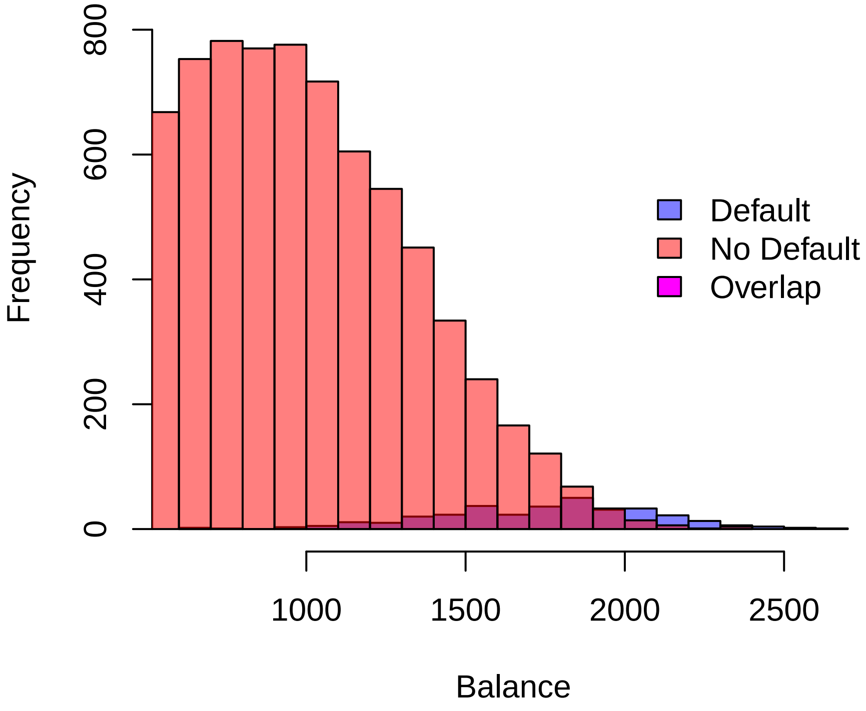
```r
h1 <- hist(balance[default == "Yes"], main = "", breaks = 20)
```



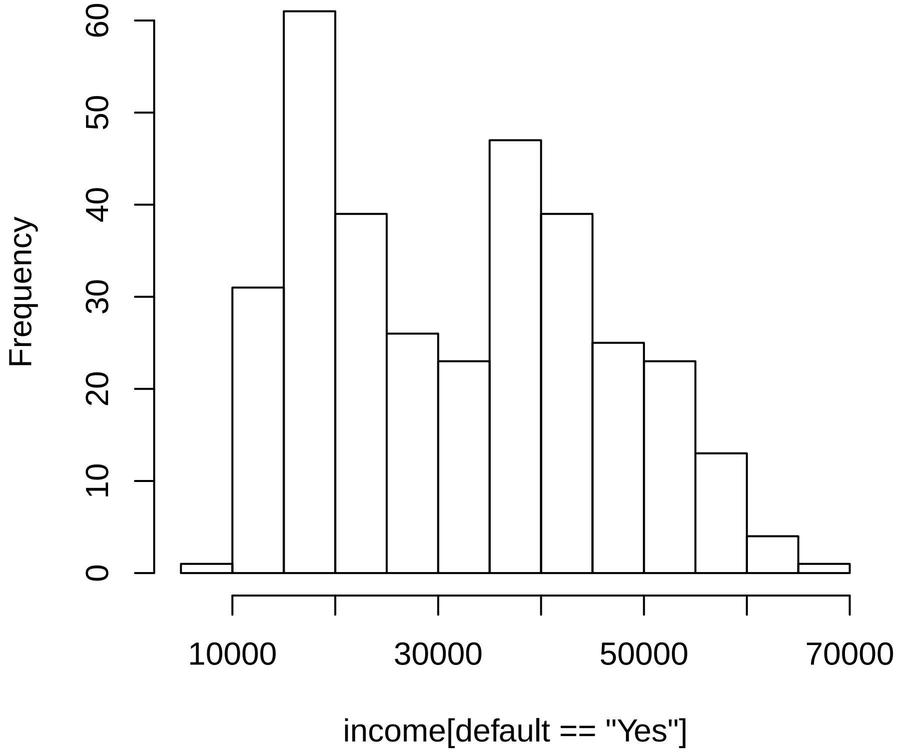
```
h2 <-hist(balance[default == "No"], main = "", breaks = 20)
```



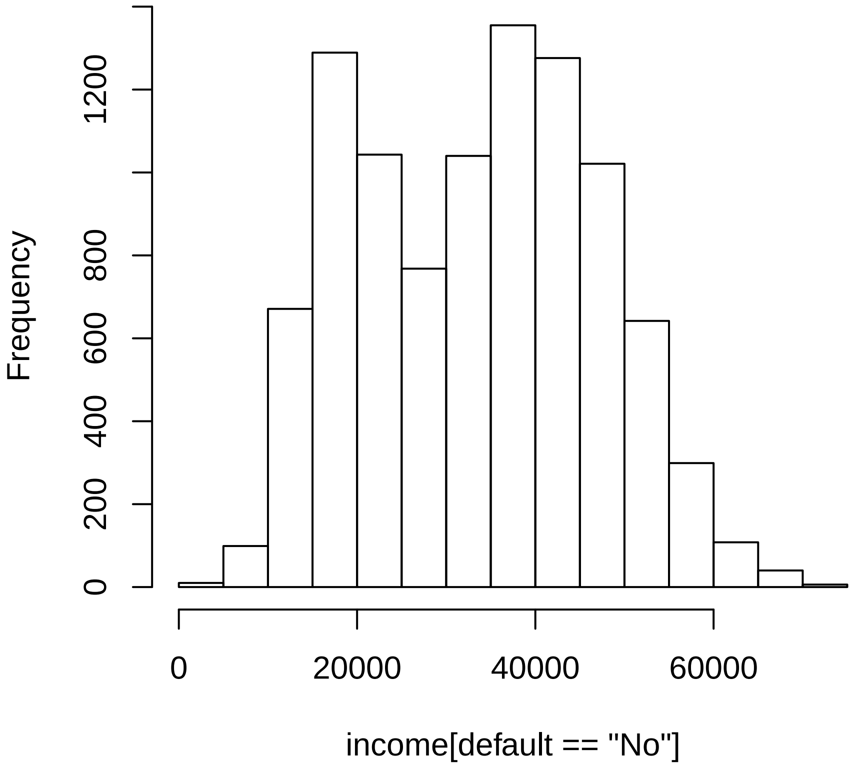
```
plot(h1, ylim = c(0, 900), col = rgb(0, 0,1, .5), main = "", xlab = "Balance")
plot(h2, add = T, col = rgb(1, 0,0, .5))
legend("right", legend = c("Default", "No Default", "Overlap"),
       fill = c(rgb(0, 0,1, .5), rgb(1, 0,0, .5), rgb(1, 0,1, 1)), bty = "n")
```



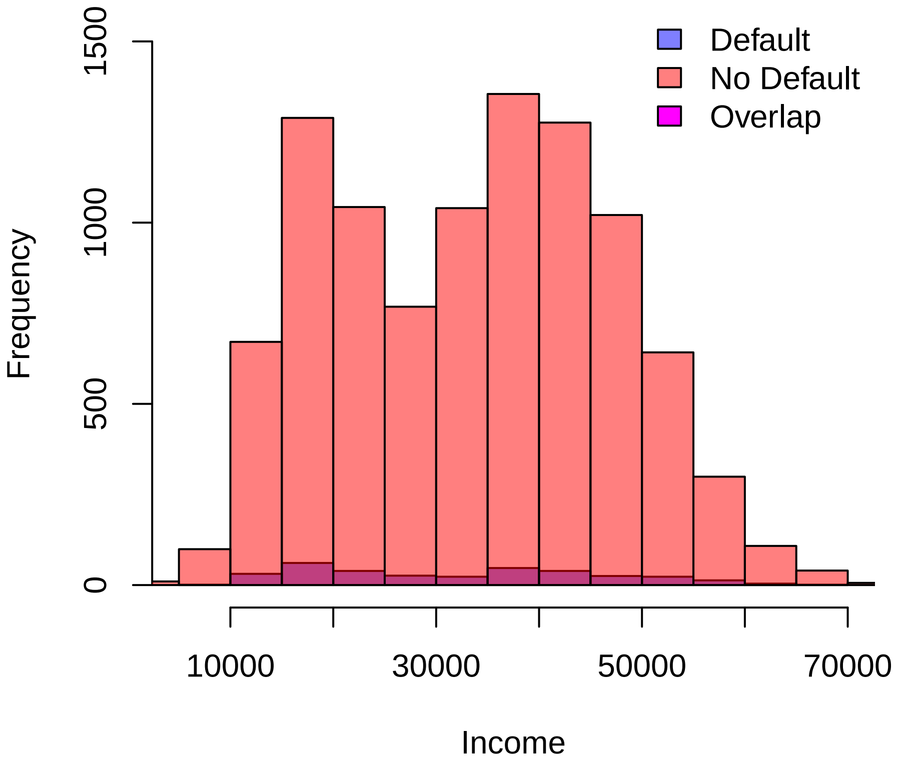
```
#People who default tend to have a higher balance

h1 <- hist(income[default == "Yes"], main = "", breaks = 20)
```



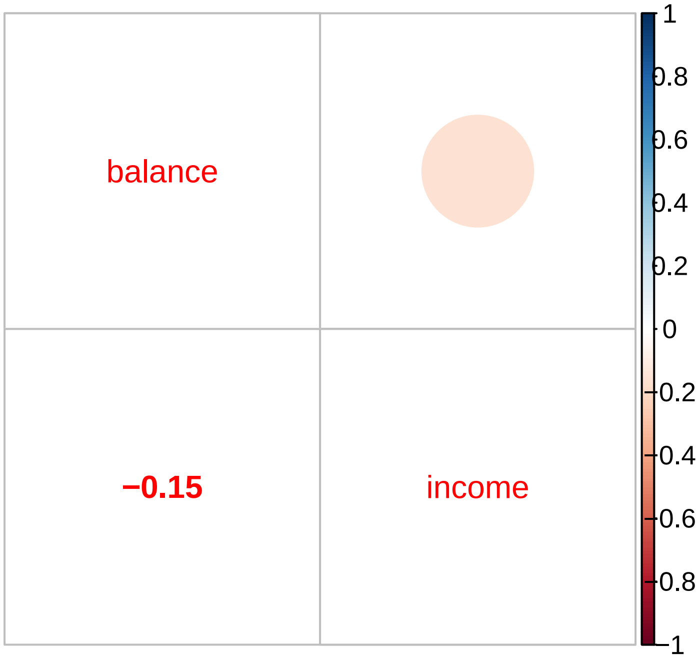
```
h2 <-hist(income[default == "No"], main = "", breaks = 20)
```



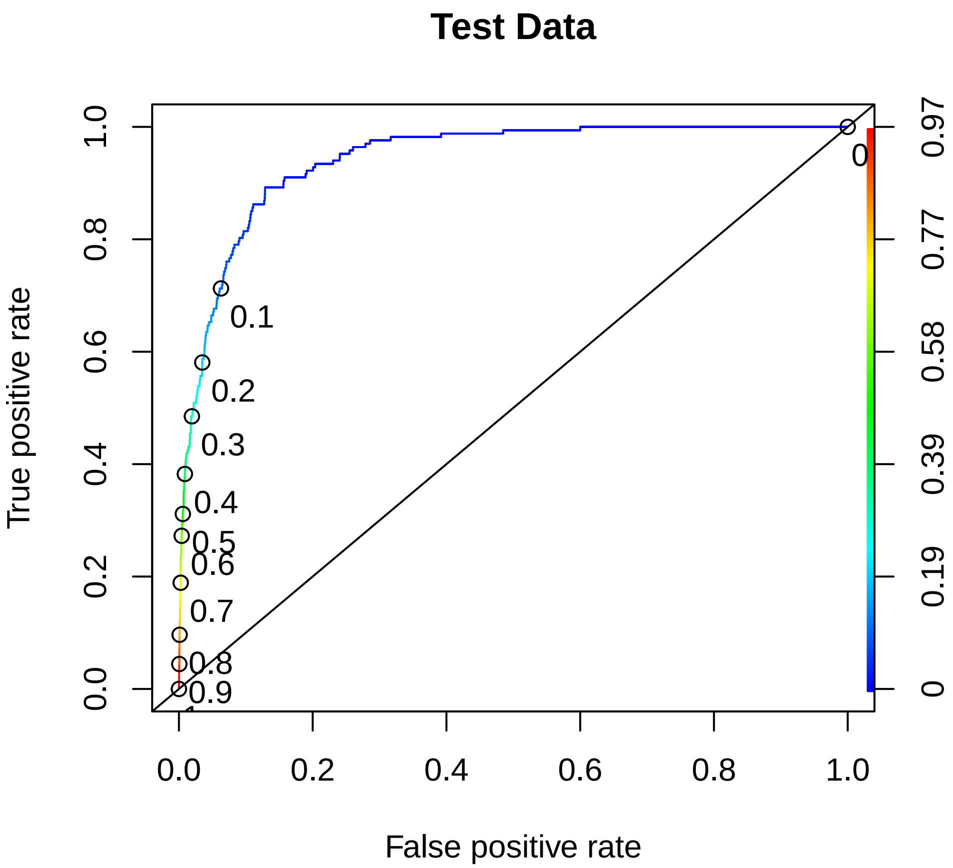
```
plot(h1, ylim = c(0, 1550), col = rgb(0, 0,1,.5), main = "", xlab = "Income")
plot(h2, add = T, col = rgb(1, 0,0, .5))
legend("topright", legend = c("Default", "No Default", "Overlap"),
       fill = c(rgb(0, 0,1, .5), rgb(1, 0,0, .5), rgb(1, 0,1, 1)), bty = "n")
```



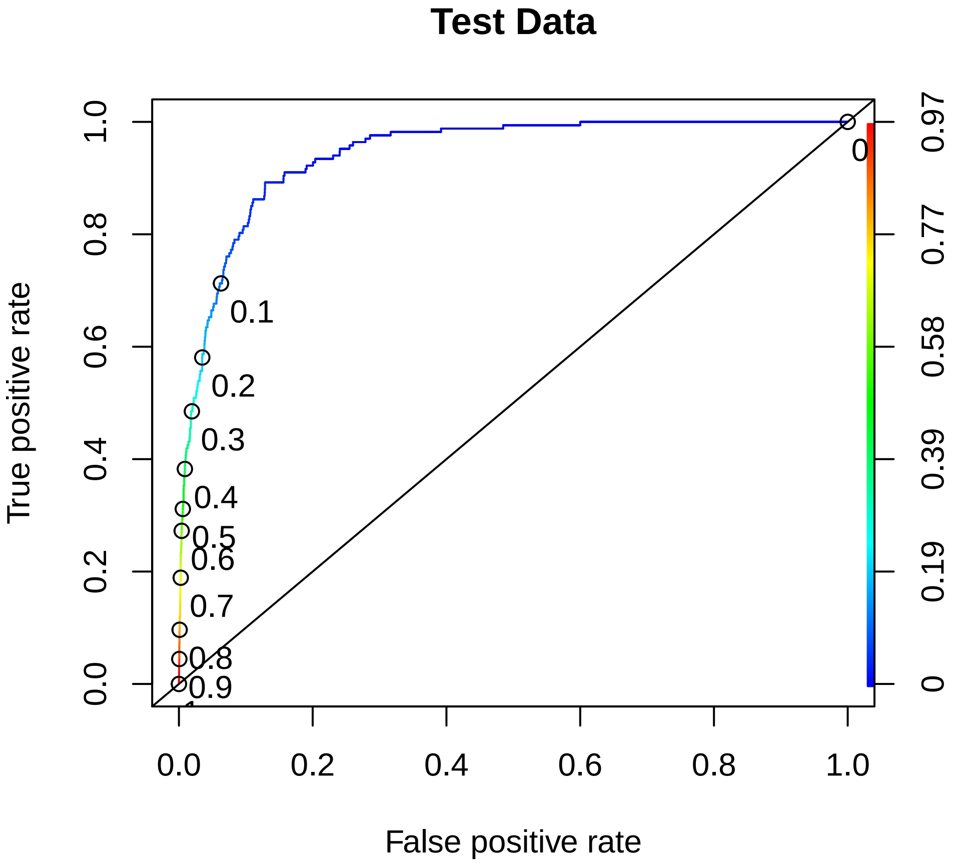
```
corrp <- cor(Default[,c(3, 4)])
corrplot.mixed(corrp, lower.col = 'red') #Weak negative correlation
```



```
table(default, student)

##        student
## default   No  Yes
##     No  6850 2817
##     Yes  206  127

chisq.test(default, student) #No association/ may not include student in model

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  default and student
## X-squared = 12.117, df = 1, p-value = 0.0004997

#########
# Task 2, Model 1: logistic regression model of default on income and balance
# Perform training/testing evaluation of Model 1
# Suggested measures:
# - ROC curve
# - Sensitivity and specificity at 0.5 cutoff (prob of default > 0.5)
# - Sensitivity and specificity at an "optimal" cutoff from ROC curve
# Recommendation: Repeat 3 times at different seeds (training/testing split) and
#   compare sensitivity and specificity at 0.5 cutoff.
#########
################################################################################
###################################### Model 1 #################################
################################################################################

# Split data into training and testing sets
p = 0.5
set.seed(1)  # set the random number generator seed
train = sample(n, p*n)  # random sample percentage out of n; this creates the index list

#Without student
fit_train = glm(default~balance+income, family=binomial(link=logit),
                data=Default, subset = train)
test_probs = predict.glm(fit_train, Default, type="response")[-train]



#Without Student
rocplot(test_probs, Default[-train,"default"], main="Test Data", colorize = T,
        print.cutoffs.at = seq(0,1, by = .1), text.adj = c(-.2, 1.7))
abline(a=0, b=1)
```



```
# Statistics off the ROC
pred = prediction(test_probs, Default[-train,"default"])
# calculating AUC
auc1 <- performance(pred,"auc")
# convert S4 class to vector
auc1 <- unlist(slot(auc1, "y.values"))
auc1

## [1] 0.9426981

#[1] 0.9426981 Pretty good
aic1 <- AIC(fit_train)

# Compute optimal cutoff
# Present sensitivity and specificity for that optimal cutoff
roc.perf = performance(pred, measure="tpr", x.measure="fpr")
print(c1 <-opt.cut(roc.perf, pred)) # 0.03517643

##                   [,1]
## sensitivity 0.89221557
## specificity 0.87130147
## cutoff      0.03517643

se.sp(.5, pred) # Sensitivity and specificity at 0.5 cutoff

## $Cutoff
##      1877
## 0.5012066
##
## $Sensitivity
## [1] 0.3113772
##
## $Specificity
## [1] 0.9942065

table((test_probs >  0.03517643), default[-train]) #Use optimal cutoff

##
##           No  Yes
##   FALSE 4211   18
##   TRUE   622  149

prop.table(table((test_probs >  0.03517643), default[-train]))

##
##               No    Yes
##   FALSE 0.8422 0.0036
##   TRUE  0.1244 0.0298

#With student
################################################################################
###################################### Model 2 #################################
################################################################################
fit_train.stud = glm(default˜balance+income +student, family=binomial(link=logit),
                data=Default, subset = train)
test_probs.stud = predict.glm(fit_train, Default, type="response")[-train]



#With Student with same seed
rocplot(test_probs.stud, Default[-train,"default"], main="Test Data", colorize = T,
        print.cutoffs.at = seq(0,1, by = .1), text.adj = c(-.2, 1.7))
abline(a=0, b=1)
```



```
# Statistics off the ROC
pred = prediction(test_probs.stud, Default[-train,"default"])
# calculating AUC
auc1stud <- performance(pred,"auc")
# convert S4 class to vector
auc1stud <- unlist(slot(auc1stud, "y.values"))
auc1stud

## [1] 0.9426981

aic1stud <- AIC(fit_train.stud)

#[1] 0.9426981 Pretty good

# Compute optimal cutoff
# Present sensitivity and specificity for that optimal cutoff
roc.perf = performance(pred, measure="tpr", x.measure="fpr")
print(c1stud <-opt.cut(roc.perf, pred)) # 0.03517643

##                   [,1]
## sensitivity 0.89221557
## specificity 0.87130147
## cutoff      0.03517643
```

```
se.sp(.5, pred) # Sensitivity and specificity at 0.5 cutoff
```

```
## $Cutoff
##      1877
## 0.5012066
##
## $Sensitivity
## [1] 0.3113772
##
## $Specificity
## [1] 0.9942065
```

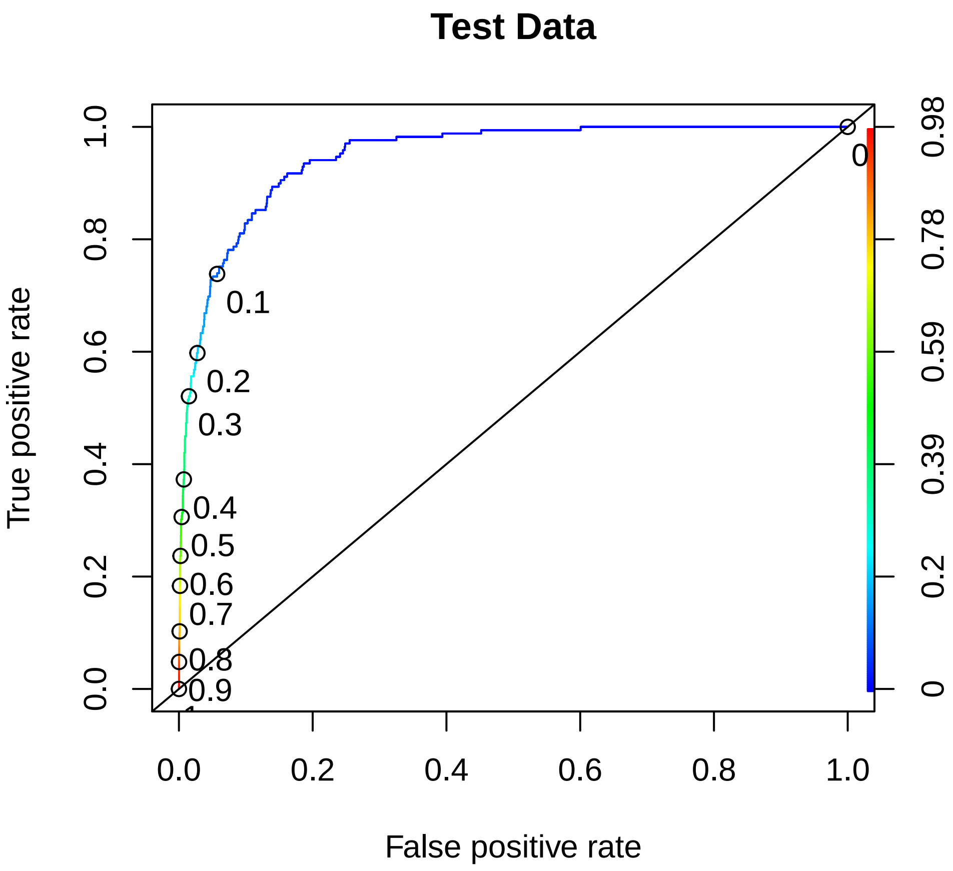
```
table((test_probs.stud >  0.03517643), default[-train]) #Use optimal cutoff
```

```
##
##           No  Yes
##   FALSE 4211   18
##   TRUE   622  149
```

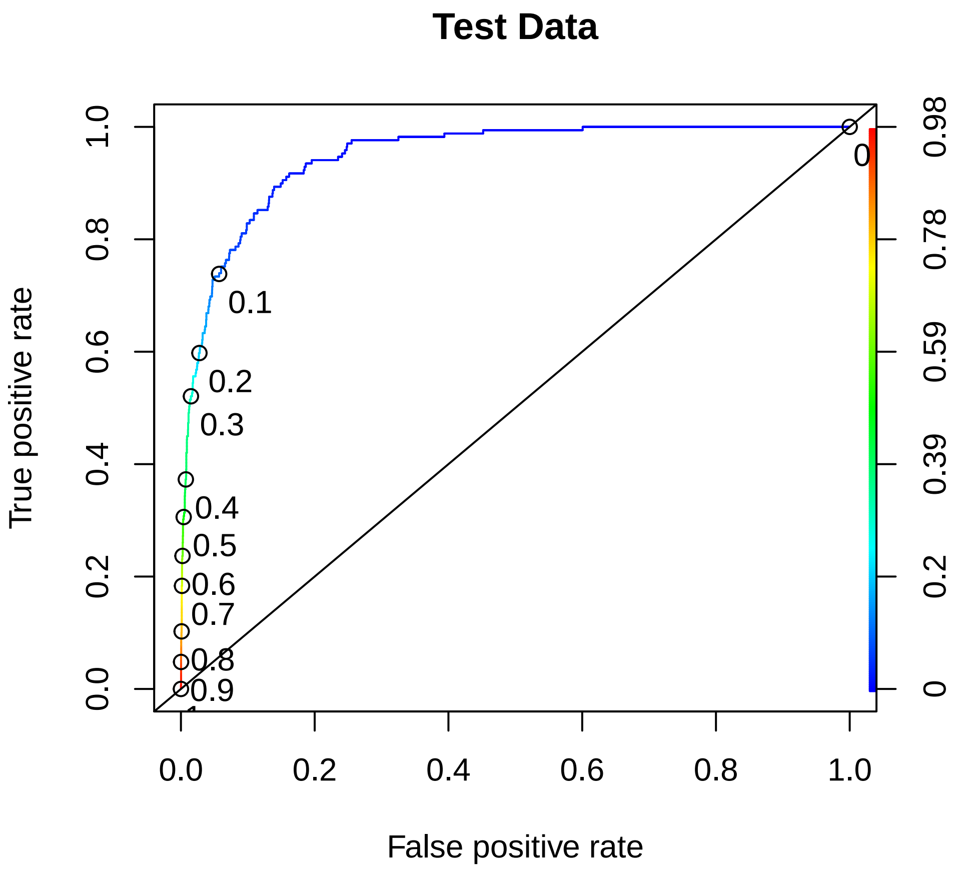
```
#They look very similar
prop.table(table((test_probs.stud >  0.03517643), default[-train]))
```

```
##
##             No    Yes
##   FALSE 0.8422 0.0036
##   TRUE  0.1244 0.0298
```

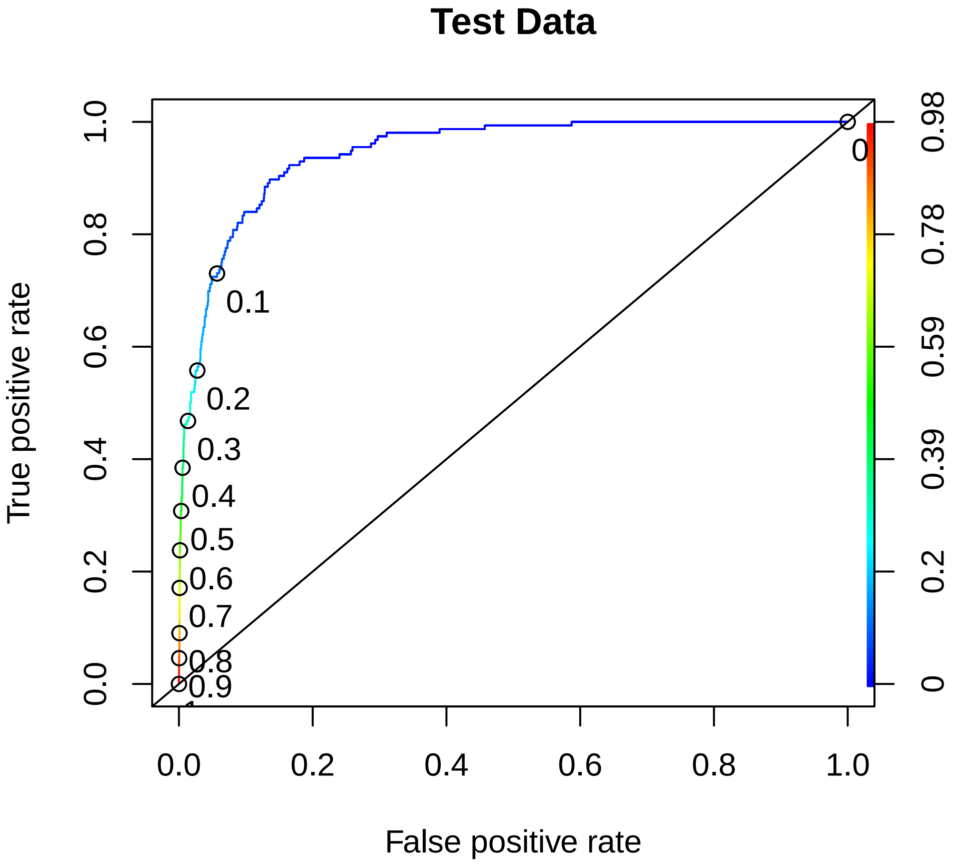
```
#################################################################################
################################   Same models   ################################
################################     seed = 2    ################################
p = 0.5
set.seed(2)  # set the random number generator seed
train = sample(n, p*n)  # random sample percentage out of n; this creates the index list

#Without student
fit_train = glm(default~balance+income, family=binomial(link=logit),
                data=Default, subset = train)
test_probs = predict.glm(fit_train, Default, type="response")[-train]



#Without Student
rocplot(test_probs, Default[-train,"default"], main="Test Data", colorize = T,
        print.cutoffs.at = seq(0,1, by = .1), text.adj = c(-.2, 1.7))
abline(a=0, b=1)
```



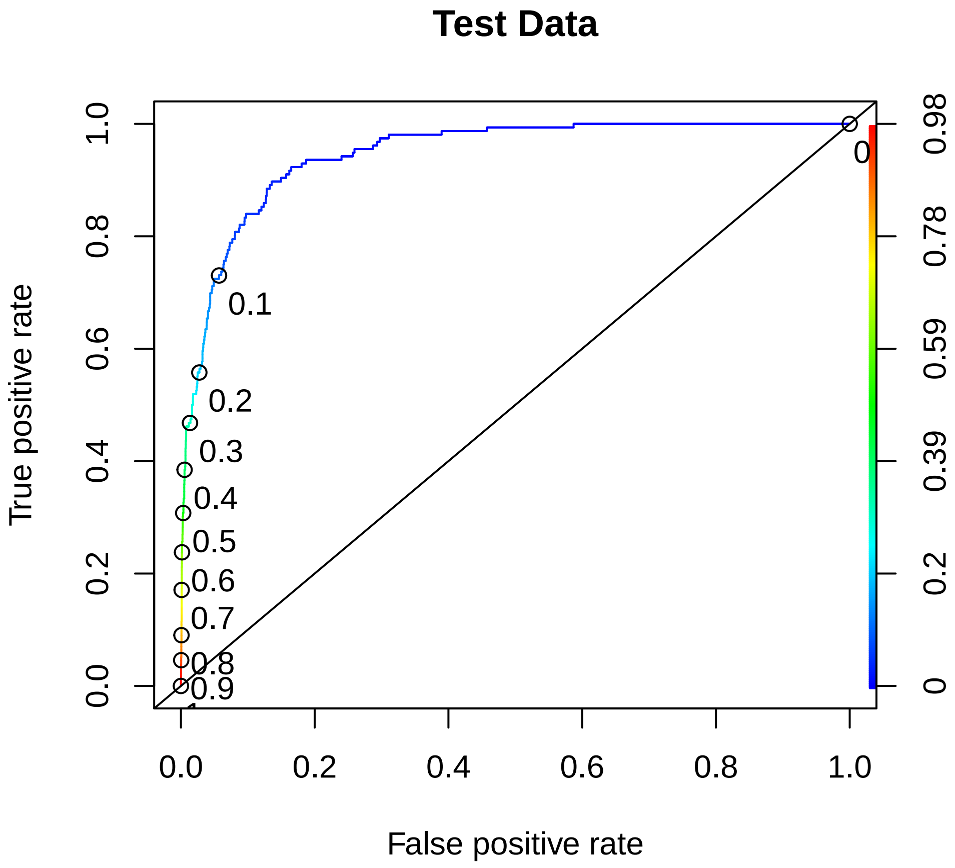
```
# Statistics off the ROC
pred = prediction(test_probs, Default[-train,"default"])
# calculating AUC
auc2 <- performance(pred,"auc")
# convert S4 class to vector
auc2 <- unlist(slot(auc2, "y.values"))
auc2

## [1] 0.9477989

#[1] 0.9477989 Almost the same value
aic2 <- AIC(fit_train)

# Compute optimal cutoff
# Present sensitivity and specificity for that optimal cutoff
roc.perf = performance(pred, measure="tpr", x.measure="fpr")
print(c2 <-opt.cut(roc.perf, pred))

##                   [,1]
## sensitivity 0.89349112
## specificity 0.86069137
## cutoff      0.03198994

se.sp(.5, pred) # Sensitivity and specificity at 0.5 cutoff

## $Cutoff
##      7336
## 0.4961785
##
## $Sensitivity
## [1] 0.3076923
##
## $Specificity
## [1] 0.9958601

table((test_probs >   0.03198994), default[-train]) #Use optimal cutoff

##
##           No  Yes
##   FALSE 4158   18
##   TRUE   673  151

#With student
################################################################################
###################################### Model 2 #################################
################################################################################
fit_train.stud = glm(default~balance+income +student, family=binomial(link=logit),
                     data=Default, subset = train)
test_probs.stud = predict.glm(fit_train, Default, type="response")[-train]



#With Student with same seed
rocplot(test_probs.stud, Default[-train,"default"], main="Test Data", colorize = T,
        print.cutoffs.at = seq(0,1, by = .1), text.adj = c(-.2, 1.7))
abline(a=0, b=1)
```



```
# Statistics off the ROC
pred = prediction(test_probs.stud, Default[-train,"default"])
# calculating AUC
auc2stud <- performance(pred,"auc")
# convert S4 class to vector
auc2stud <- unlist(slot(auc2stud, "y.values"))
auc2stud

## [1] 0.9477989

#[1] 0.9477989 Pretty good
aic2stud <- AIC(fit_train.stud)

# Compute optimal cutoff
# Present sensitivity and specificity for that optimal cutoff
roc.perf = performance(pred, measure="tpr", x.measure="fpr")
print(c2stud <-opt.cut(roc.perf, pred))# 0.03198994

##                   [,1]
## sensitivity 0.89349112
## specificity 0.86069137
## cutoff      0.03198994

se.sp(.5, pred) # Sensitivity and specificity at 0.5 cutoff

## $Cutoff
##      7336
## 0.4961785
##
## $Sensitivity
## [1] 0.3076923
##
## $Specificity
## [1] 0.9958601

table((test_probs.stud >  0.03198994), default[-train]) #Use optimal cutoff

##
##           No  Yes
##   FALSE 4158   18
##   TRUE   673  151

#They look very similar to seed 1



#################################################################################
################################   Same models   ################################
################################     seed = 3    ################################
p = 0.5
set.seed(3)  # set the random number generator seed
train = sample(n, p*n)  # random sample percentage out of n; this creates the index list

#Without student
fit_train = glm(default~balance+income, family=binomial(link=logit),
                data=Default, subset = train)
test_probs = predict.glm(fit_train, Default, type="response")[-train]



#Without Student
rocplot(test_probs, Default[-train,"default"], main="Test Data", colorize = T,
        print.cutoffs.at = seq(0,1, by = .1), text.adj = c(-.2, 1.7))
abline(a=0, b=1)
```

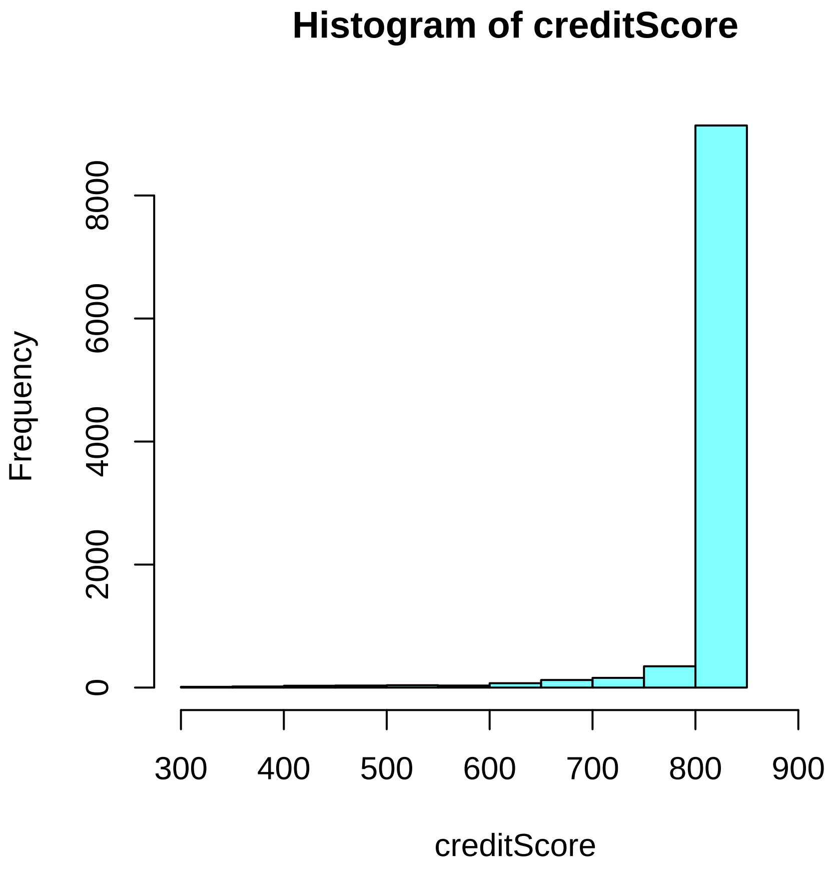


```
# Statistics off the ROC
pred = prediction(test_probs, Default[-train,"default"])
# calculating AUC
auc3 <- performance(pred,"auc")
# convert S4 class to vector
auc3 <- unlist(slot(auc3, "y.values"))
auc3

## [1] 0.9463478

#[1] 0.9463478 Almost the same value
aic3 <- AIC(fit_train)

# Compute optimal cutoff
# Present sensitivity and specificity for that optimal cutoff
roc.perf = performance(pred, measure="tpr", x.measure="fpr")
print(c3 <-opt.cut(roc.perf, pred))

##                    [,1]
## sensitivity 0.89743590
## specificity 0.86436829
## cutoff      0.03251589

se.sp(.5, pred) # Sensitivity and specificity at 0.5 cutoff

## $Cutoff
##      3482
## 0.5134872
##
## $Sensitivity
## [1] 0.3076923
##
## $Specificity
## [1] 0.9966969

table((test_probs >   0.03251589), default[-train]) #Use optimal cutoff

##
##           No  Yes
##   FALSE 4187   17
##   TRUE   657  139

#With student
################################################################################
###################################### Model 2 #################################
################################################################################
fit_train.stud = glm(default~balance+income +student, family=binomial(link=logit),
                     data=Default, subset = train)
test_probs.stud = predict.glm(fit_train, Default, type="response")[-train]



#With Student with same seed = 3
rocplot(test_probs.stud, Default[-train,"default"], main="Test Data", colorize = T,
        print.cutoffs.at = seq(0,1, by = .1), text.adj = c(-.2, 1.7))
abline(a=0, b=1)
```

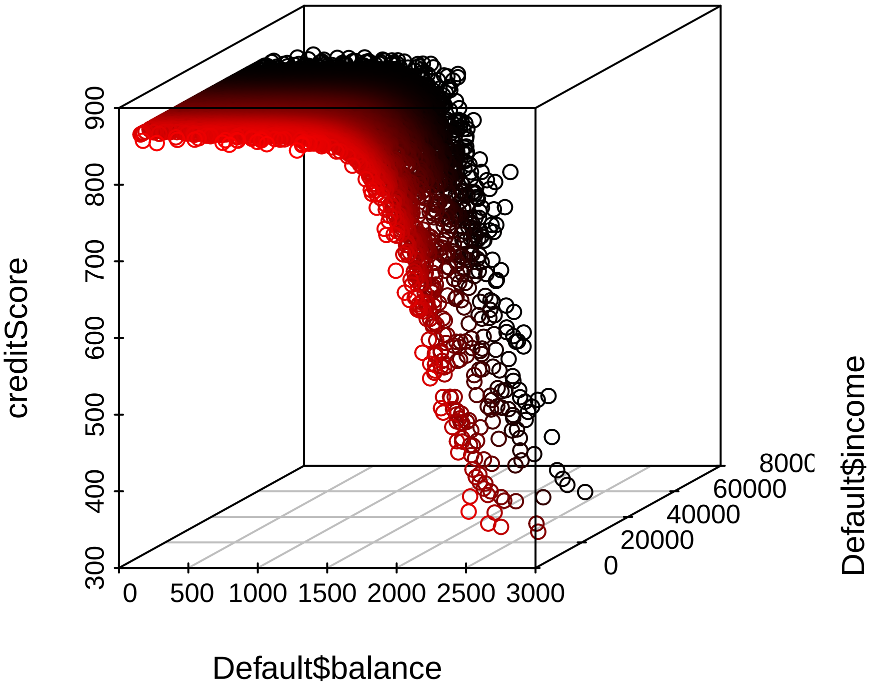


```
# Statistics off the ROC
pred = prediction(test_probs.stud, Default[-train,"default"])
# calculating AUC
auc3stud <- performance(pred,"auc")
# convert S4 class to vector
auc3stud <- unlist(slot(auc3stud, "y.values"))
auc3stud

## [1] 0.9463478

#[1] 0.9463478 Pretty good
aic3stud <- AIC(fit_train.stud)

# Compute optimal cutoff
# Present sensitivity and specificity for that optimal cutoff
roc.perf = performance(pred, measure="tpr", x.measure="fpr")
print(c3stud <-opt.cut(roc.perf, pred)) # 0.03251589

##                    [,1]
## sensitivity 0.89743590
## specificity 0.86436829
## cutoff      0.03251589

se.sp(.5, pred) # Sensitivity and specificity at 0.5 cutoff

## $Cutoff
##      3482
## 0.5134872
##
## $Sensitivity
## [1] 0.3076923
##
## $Specificity
## [1] 0.9966969

table((test_probs.stud >  0.03251589), default[-train]) #Use optimal cutoff

##
##           No  Yes
##   FALSE 4187   17
##   TRUE   657  139

#They look very similar to seed 1

###########
# Task 4, K-fold cross-validation evaluation of the two models
#  Recommend 10-fold cross-validation, as leave-one-out is slow on this size data set
###########

# First, 10-fold cv on Model 1 (default on balance and income)
fit1 = glm(default~balance+income, family=binomial(link=logit), data=Default)
summary(fit1)

##
## Call:
## glm(formula = default ~ balance + income, family = binomial(link = logit),
##     data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4725  -0.1444  -0.0574  -0.0211   3.7245
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
## income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1579.0  on 9997  degrees of freedom
## AIC: 1585
##
## Number of Fisher Scoring iterations: 8

set.seed(17) # set the random number generator seed
# 10-fold cv, compute misclassification rate
# Note that this R function also provides a second component with a bias adjustment
cv.error.10 = cv.glm(Default, fit1, K=10)
cv.error.10$delta[1]

## [1] 0.021442

#[1] 0.021442

set.seed(17) # set the random number generator seed
# Second, 10-fold cv on Model 2 (default on balance, income, AND student)
fit2 = glm(default~balance+income+student, family=binomial(link=logit), data=Default)
summary(fit2)

##
## Call:
## glm(formula = default ~ balance + income + student, family = binomial(link = logit),
##     data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4691  -0.1418  -0.0557  -0.0203   3.7383
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
##
## Number of Fisher Scoring iterations: 8

cv.error.10.stud = cv.glm(Default, fit2, K=10)
cv.error.10.stud$delta[1]

## [1] 0.02137601

#[1]  0.02137601

# We choose ton
```

```
##########
# Task 5, compute a 'risk score' of default from the model you choose, Model 1 or Model 2
#  Consider the package scatterplot3d to present a static 3D plot and/or
#  rgl with function plot3d for a spinning 3D plot (risk score against balance and income)
##########


set.seed(1)
fit.final <- glm(default~balance+income, family=binomial(link=logit),
                                  data=Default, subset = train)
creditScore <- risk.score(fit.final, data = Default)

hist(creditScore, col = rgb(0,1,1,.5), xlim = c(300, 950))
```



```
scatterplot3d(x = Default$balance, y = Default$income, z = creditScore,
              angle = 30, highlight.3d  = T, type = "p")
```



```
p <- plot_ly(Default, x = ~Default$balance, y = ~Default$income, z = ~creditScore
             , color = ~creditScore, colors = c('#BF382A', '#0C4B8E')) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'Balance'),
                      yaxis = list(title = 'Income'),
                      zaxis = list(title = 'Credit Score')))
p

## Error in loadNamespace(name):  there is no package called 'webshot'

dev.off()

## null device
##           1

#
mean(auc1, auc2, auc3)

## [1] 0.9426981

mean(auc1stud, auc2stud, auc2stud)
```

```
## [1] 0.9426981

mean(aic1, aic2, aic3)

## [1] 740.3964

mean(aic1stud, aic2stud, aic3stud)

## [1] 739.8096

mean(c1[3], c2[3], c3[3])

## [1] 0.03517643

mean(c1stud[3], c2stud[3], c3stud[3])

## [1] 0.03517643
```