Credit Risk

Alexis Koffi

08/02/2019

Introduction

This project aims to improve on the state of the art in credit scoring by predicting the probability that somebody will experience financial distress in the next few years.

When a borrower fails to fully reimburse a loan, referred to as a loan default, this results in a loss to the bank. Banks keep information on the default behaviour of past customers which can be used to predict the default behaviour of new customers, therefore helping assess risk and determine interest rates. Two widely used methods for calculating these predictions are logistic regression and classification trees.

We will use the following library:

```r
library(tidyverse)
```

Data Exploration

The data used comes from Kaggle, available here. Notice that we need to login before downloadind the zip file **"GiveMeSomeCredit.zip"**. We have already two files **cs-training.csv** for the training set and **cs-test.csv** for the testing set.

```r
# /!\ Automatic download doesn't work with the code below, due to kaggle login + password

# download.file("https://www.kaggle.com/c/2551/download-all", "GiveMeSomeCredit.zip")
# unzip("GiveMeSomeCredit.zip", exdir = "data")
# file.remove("GiveMeSomeCredit.zip")

# So, I manually download the zip file and unzip it in a directory called "data" before working with

training <- read.csv("data/cs-training.csv")
test <- read.csv("data/cs-test.csv")
str(training)
```

```
## 'data.frame':    150000 obs. of  12 variables:
##  $ X                                   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ SeriousDlqin2yrs                    : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ RevolvingUtilizationOfUnsecuredLines: num  0.766 0.957 0.658 0.234 0.907 ...
##  $ age                                 : int  45 40 38 30 49 74 57 39 27 57 ...
##  $ NumberOfTime30.59DaysPastDueNotWorse: int  2 0 1 0 1 0 0 0 0 0 ...
##  $ DebtRatio                           : num  0.803 0.1219 0.0851 0.036 0.0249 ...
##  $ MonthlyIncome                       : int  9120 2600 3042 3300 63588 3500 NA 3500 NA 23684 ...
##  $ NumberOfOpenCreditLinesAndLoans     : int  13 4 2 5 7 3 8 8 2 9 ...
##  $ NumberOfTimes90DaysLate             : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ NumberRealEstateLoansOrLines        : int  6 0 0 0 1 1 3 0 0 4 ...
##  $ NumberOfTime60.89DaysPastDueNotWorse: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NumberOfDependents                  : int  2 1 0 0 0 1 0 0 NA 2 ...
```

List of variables

Here is a view of the the variables :

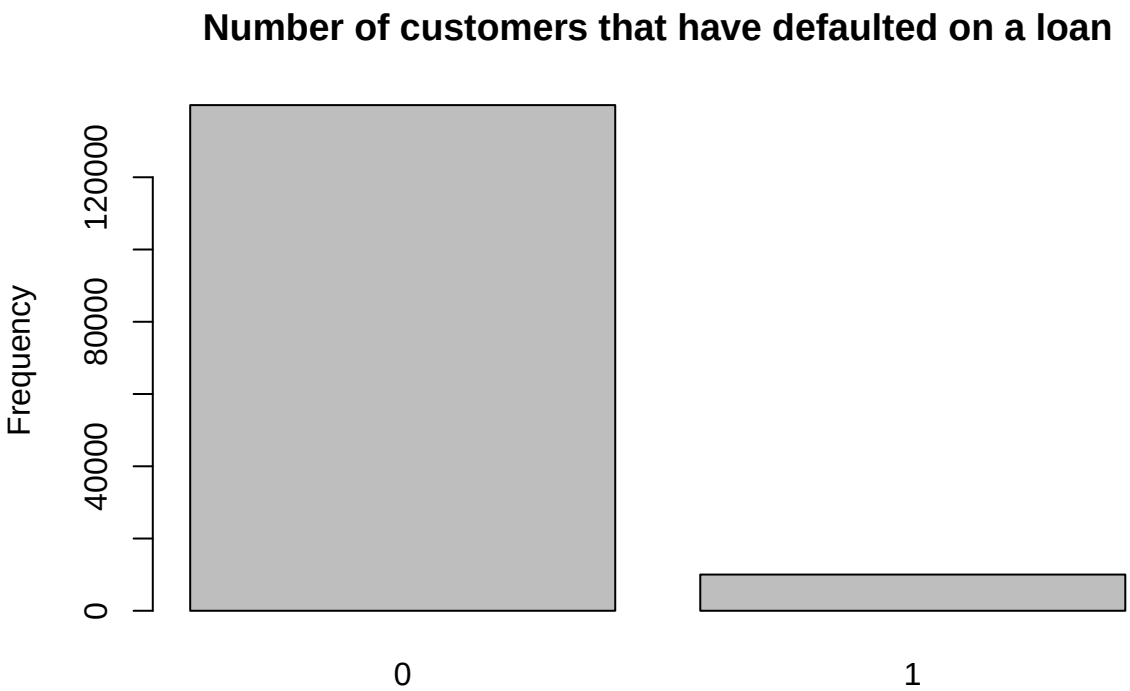
SeriousDlqin2yrs : indicates default behaviour, with 1 representing default (loan repayment is overdue by 90 days or more) and 0 representing non-default.

```
# SeriousDlqin2yrs Proportion
prop.table(table(training$SeriousDlqin2yrs))
```

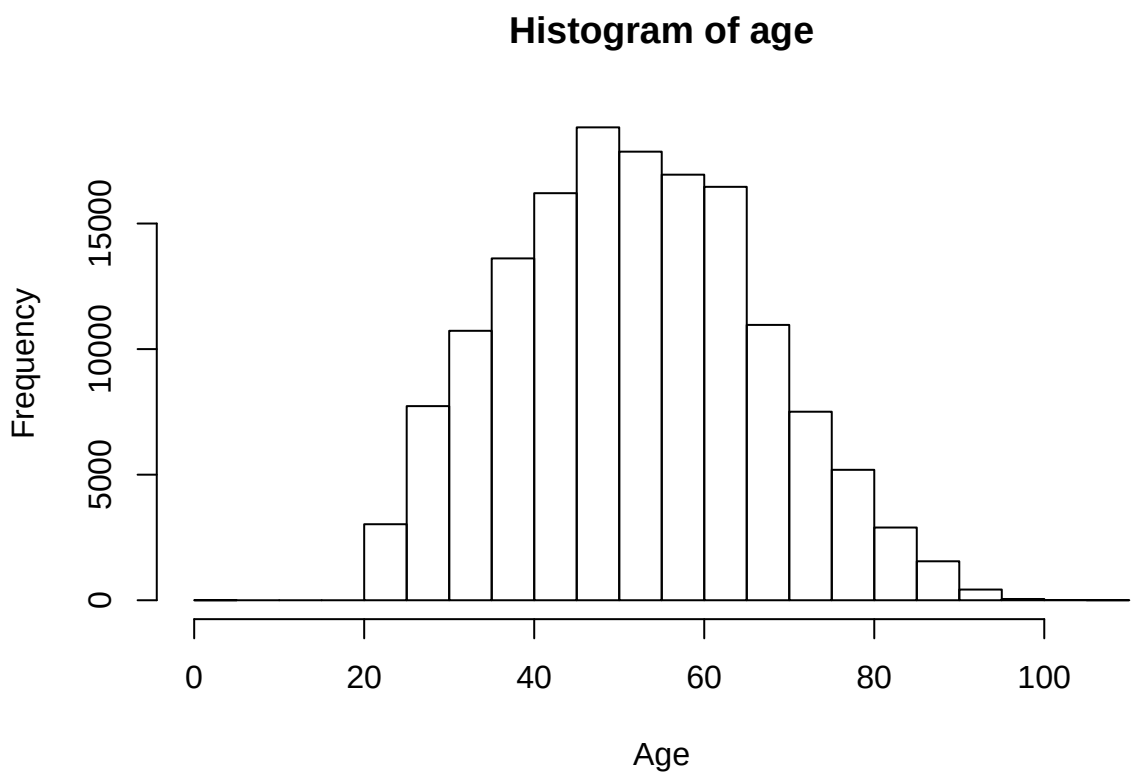
```
##
##       0       1
## 0.93316 0.06684
```

Default is a rare event. As seen in the table above, less than 7% of customers in this dataset defaulted on a loan.

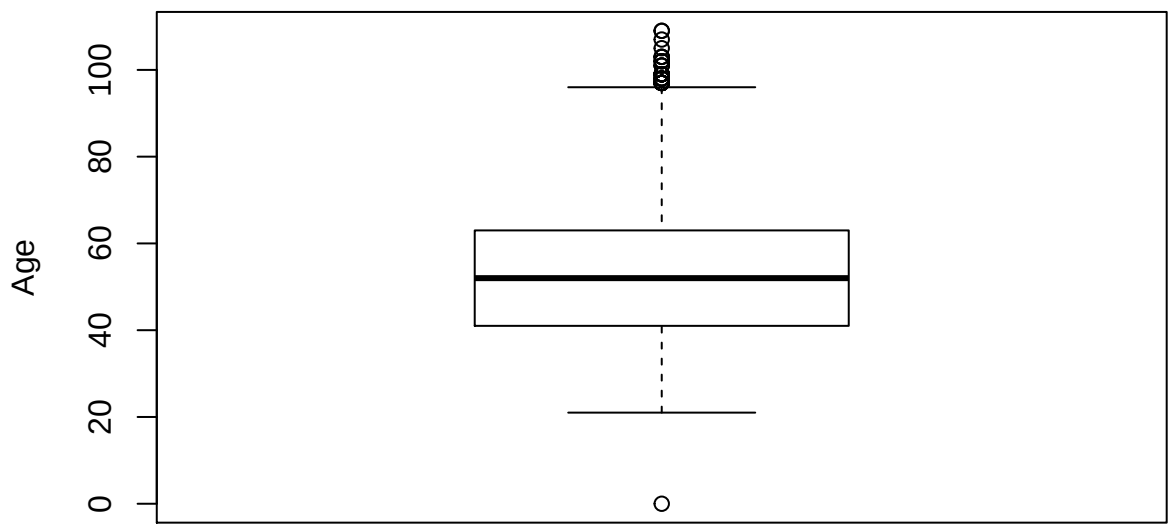
```
counts <- table(training$SeriousDlqin2yrs)
barplot(counts, main = "Number of customers that have defaulted on a loan", ylab = "Frequency")
```




```
hist(training$age, main = "Histogram of age", xlab = "Age")
```

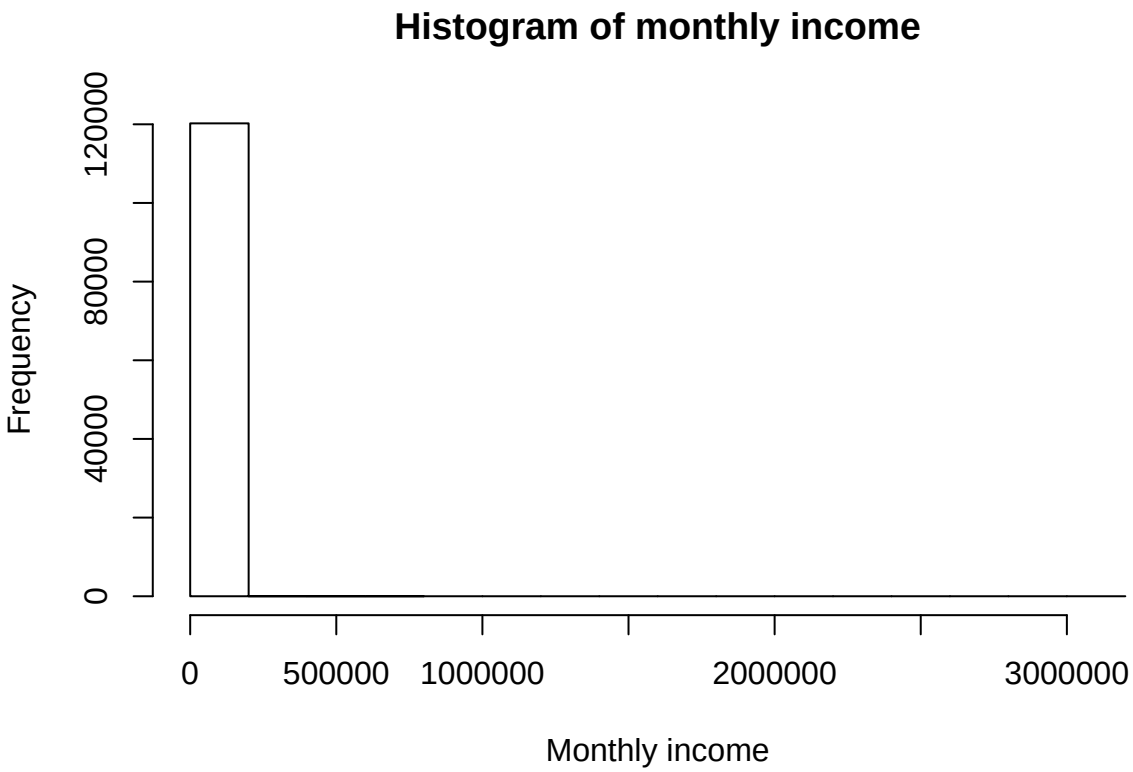



```r
boxplot(training$age, ylab = "Age")
```

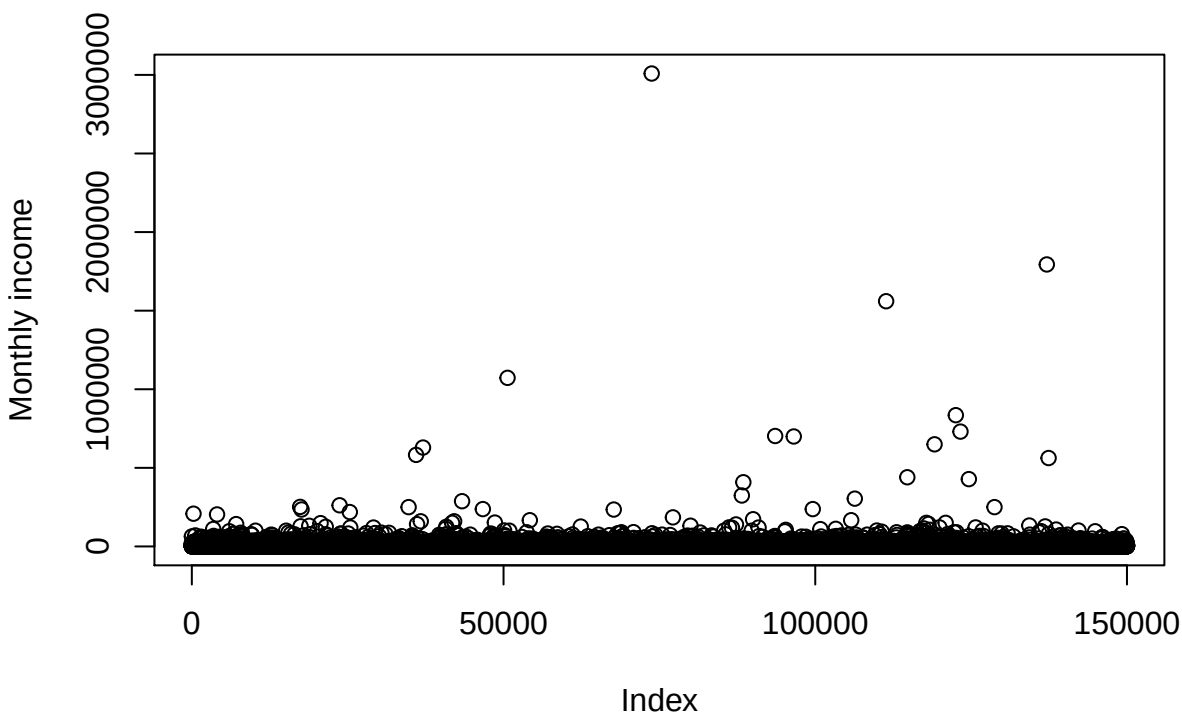


Age is normally distributed with one outlier (`Age == 0`).

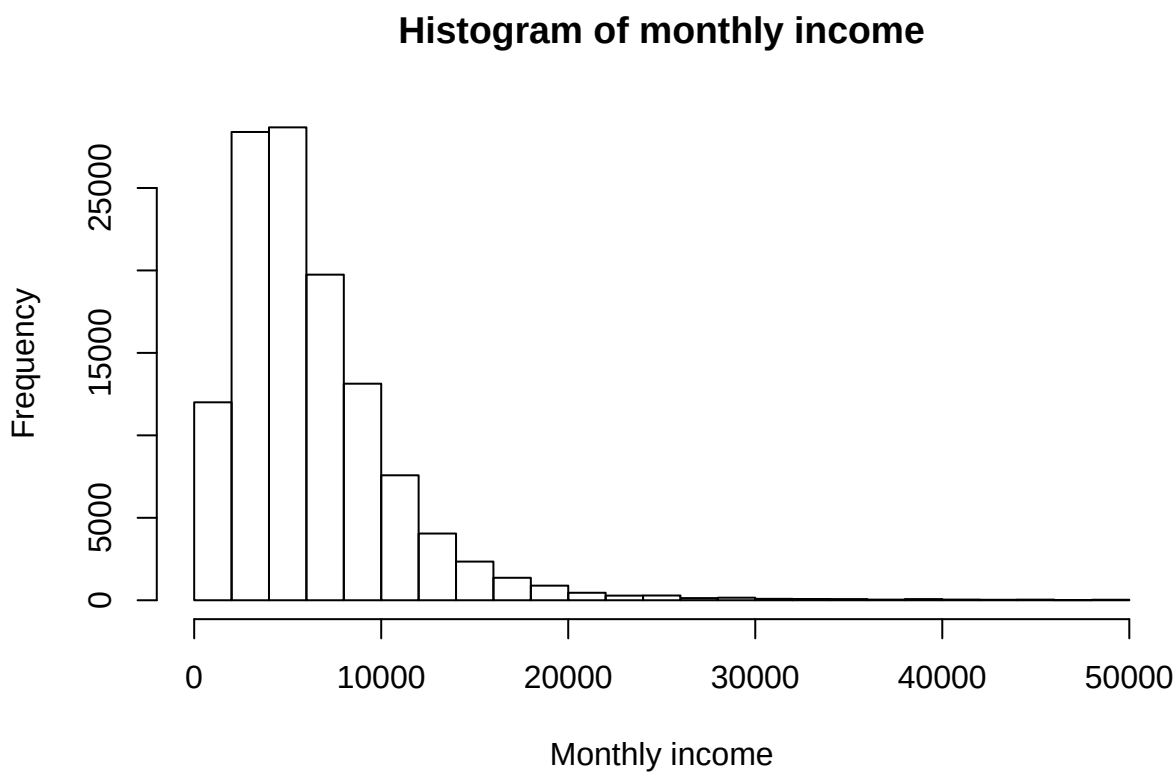
```r
hist(training$MonthlyIncome, main = "Histogram of monthly income", xlab = "Monthly income")
```



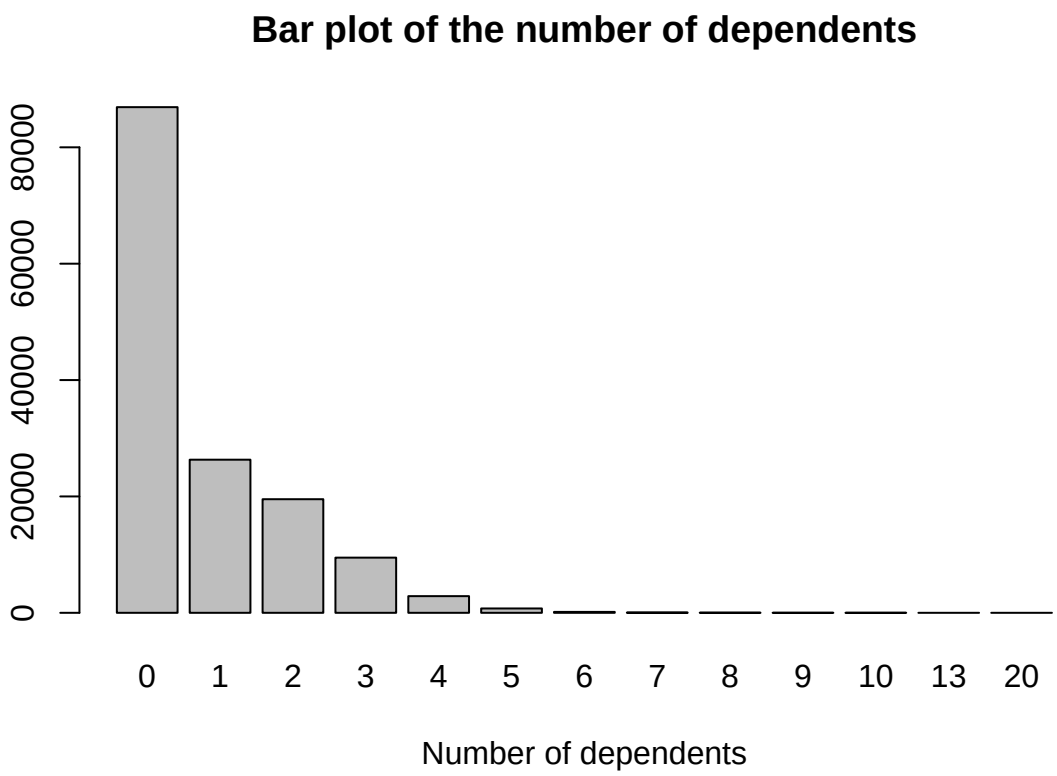

```r
plot(training$MonthlyIncome, ylab = "Monthly income")
```



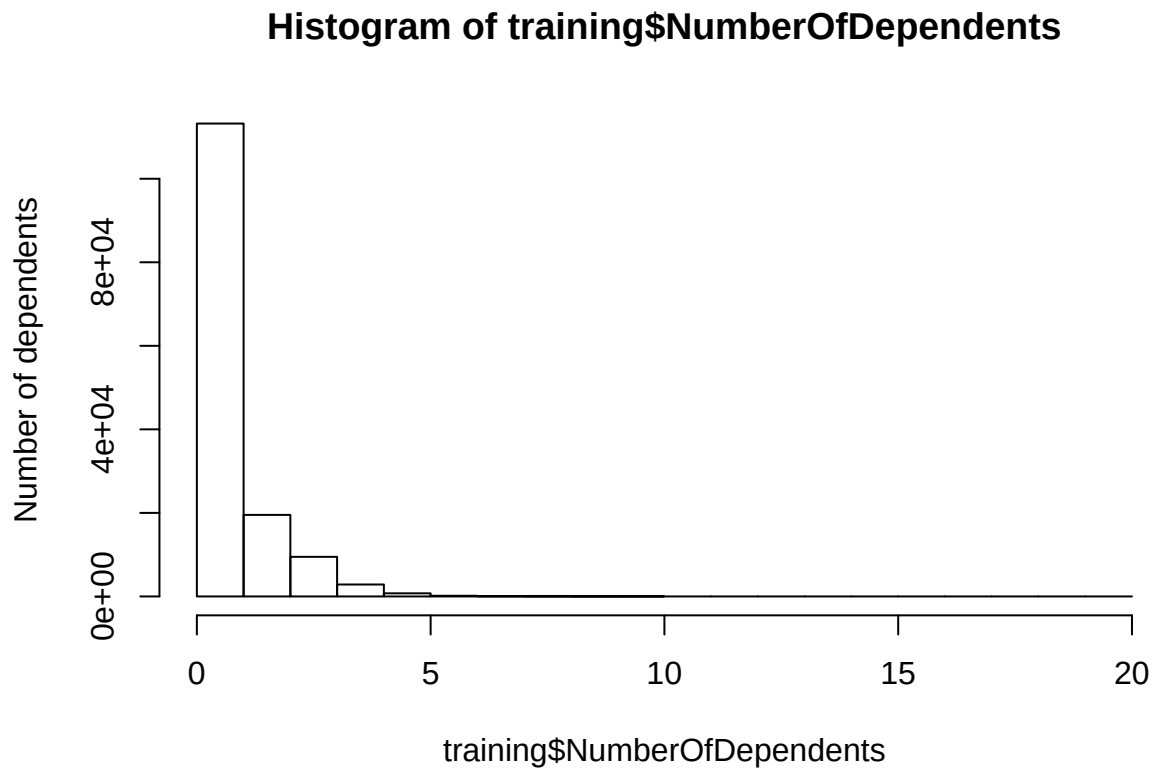

```r
index_highincome <- which(training$MonthlyIncome > 50000)
new_data <- training[-index_highincome, ]

hist(new_data$MonthlyIncome, main = "Histogram of monthly income", xlab = "Monthly income")
```




```r
counts <- table(training$NumberOfDependents)
barplot(counts, main = "Bar plot of the number of dependents", xlab = "Number of dependents")
```



```r
hist(training$NumberOfDependents, ylab = "Number of dependents")
```



Most borrowers have less than one dependent.

```r
training_summary <- training[-1]
summary(training_summary)
```

```
##  SeriousDlqin2yrs  RevolvingUtilizationOfUnsecuredLines      age
##  Min.   :0.00000   Min.   :    0.00                     Min.   :  0.0
##  1st Qu.:0.00000   1st Qu.:    0.03                     1st Qu.: 41.0
##  Median :0.00000   Median :    0.15                     Median : 52.0
##  Mean   :0.06684   Mean   :    6.05                     Mean   : 52.3
##  3rd Qu.:0.00000   3rd Qu.:    0.56                     3rd Qu.: 63.0
##  Max.   :1.00000   Max.   :50708.00                     Max.   :109.0
##
##  NumberOfTime30.59DaysPastDueNotWorse   DebtRatio        MonthlyIncome
##  Min.   : 0.000                       Min.   :     0.0   Min.   :      0
##  1st Qu.: 0.000                       1st Qu.:     0.2   1st Qu.:   3400
##  Median : 0.000                       Median :     0.4   Median :   5400
##  Mean   : 0.421                       Mean   :   353.0   Mean   :   6670
##  3rd Qu.: 0.000                       3rd Qu.:     0.9   3rd Qu.:   8249
##  Max.   :98.000                       Max.   :329664.0   Max.   :3008750
##                                                          NA's   :29731
##  NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
##  Min.   : 0.000                  Min.   : 0.000
##  1st Qu.: 5.000                  1st Qu.: 0.000
##  Median : 8.000                  Median : 0.000
##  Mean   : 8.453                  Mean   : 0.266
##  3rd Qu.:11.000                  3rd Qu.: 0.000
##  Max.   :58.000                  Max.   :98.000
```

```
##
##  NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
##  Min.   : 0.000               Min.   : 0.0000
##  1st Qu.: 0.000               1st Qu.: 0.0000
##  Median : 1.000               Median : 0.0000
##  Mean   : 1.018               Mean   : 0.2404
##  3rd Qu.: 2.000               3rd Qu.: 0.0000
##  Max.   :54.000               Max.   :98.0000
##
##  NumberOfDependents
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 0.000
##  Mean   : 0.757
##  3rd Qu.: 1.000
##  Max.   :20.000
##  NA's   :3924
```

MonthlyIncome and *NumberOfDependents* have some `NA`s. One method for dealing with missing values is by replacing them with the median.

```
replace_NA <- function(x){
  index_NA <- which(is.na(x))
  x[index_NA] <- median(x, na.rm = TRUE)
  x
}

training$MonthlyIncome <- replace_NA(training$MonthlyIncome)
training$NumberOfDependents <- replace_NA(training$NumberOfDependents)
```

As seen in the plots and summary above, there appear to be some outliers. However, I will leave these for now because of the many observations.

Methods

We are interested in predicting the probability of default in the `test` dataset.

```
head(test$SeriousDlqin2yrs)
```

```
## [1] NA NA NA NA NA NA
```

Note *SeriousDlqin2yrs* is `NA` for each observation.

We can predict the probability of default by using logistic regression, which has an output between 0 and 1.

```
log_model <- glm(SeriousDlqin2yrs ~ age, family = "binomial", data = training)
summary(log_model)
```

```
##
## Call:
## glm(formula = SeriousDlqin2yrs ~ age, family = "binomial", data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7948  -0.4113  -0.3445  -0.2833   2.9591
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.9902369  0.0366606  -27.01   <2e-16 ***
## age         -0.0334177  0.0007579  -44.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 73616  on 149999  degrees of freedom
## Residual deviance: 71539  on 149998  degrees of freedom
## AIC: 71543
##
## Number of Fisher Scoring iterations: 6
```

Interpretation of coefficients

The coefficient for *age* is negative. So, for every increase in *age* there is a lower probability of default (one extra year of age lowers the probability of default by about 3%).

Multiple variables

```r
log_model_multi <- glm(SeriousDlqin2yrs ~ . - X, family = "binomial", data = training)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(log_model_multi)
```

```
##
## Call:
## glm(formula = SeriousDlqin2yrs ~ . - X, family = "binomial",
##     data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2963  -0.3918  -0.3148  -0.2534   4.9449
##
## Coefficients:
##                                        Estimate Std. Error z value
## (Intercept)                           -1.319e+00  4.196e-02 -31.435
## RevolvingUtilizationOfUnsecuredLines -3.680e-05  6.110e-05  -0.602
## age                                   -2.842e-02  8.297e-04 -34.259
## NumberOfTime30.59DaysPastDueNotWorse  5.034e-01  1.110e-02  45.349
## DebtRatio                             -2.753e-05  1.043e-05  -2.640
## MonthlyIncome                         -3.593e-05  3.145e-06 -11.423
## NumberOfOpenCreditLinesAndLoans       -7.680e-03  2.522e-03  -3.045
## NumberOfTimes90DaysLate                4.681e-01  1.521e-02  30.769
## NumberRealEstateLoansOrLines           6.833e-02  1.058e-02   6.457
## NumberOfTime60.89DaysPastDueNotWorse -9.395e-01  1.771e-02 -53.034
## NumberOfDependents                     9.552e-02  9.065e-03  10.538
##                                       Pr(>|z|)
## (Intercept)                            < 2e-16 ***
## RevolvingUtilizationOfUnsecuredLines  0.54700
## age                                    < 2e-16 ***
## NumberOfTime30.59DaysPastDueNotWorse  < 2e-16 ***
```

```
## DebtRatio                              0.00829 **
## MonthlyIncome                          < 2e-16 ***
## NumberOfOpenCreditLinesAndLoans        0.00232 **
## NumberOfTimes90DaysLate                < 2e-16 ***
## NumberRealEstateLoansOrLines          1.07e-10 ***
## NumberOfTime60.89DaysPastDueNotWorse  < 2e-16 ***
## NumberOfDependents                     < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 73616  on 149999  degrees of freedom
## Residual deviance: 67515  on 149989  degrees of freedom
## AIC: 67537
##
## Number of Fisher Scoring iterations: 6
```

RevolvingUtilizationOfUnsecuredLines should be removed from the model because it is highly non-significant ($p = 0.45885$). I decided to keep *NumberOfOpenCreditLinesAndLoans* because it is only slightly above 0.05.

```
log_model_multi <- glm(SeriousDlqin2yrs ~ . - X - RevolvingUtilizationOfUnsecuredLines, family = "binom
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Predicting the probability of default

Missing values

```
str(test)
```

```
## 'data.frame':    101503 obs. of  12 variables:
##  $ X                                   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ SeriousDlqin2yrs                    : logi  NA NA NA NA NA NA ...
##  $ RevolvingUtilizationOfUnsecuredLines: num  0.8855 0.4633 0.0433 0.2803 1 ...
##  $ age                                 : int  43 57 59 38 27 63 50 79 68 23 ...
##  $ NumberOfTime30.59DaysPastDueNotWorse: int  0 0 0 1 0 0 0 1 0 98 ...
##  $ DebtRatio                           : num  0.1775 0.5272 0.6876 0.926 0.0199 ...
##  $ MonthlyIncome                       : int  5700 9141 5083 3200 3865 4140 0 3301 NA 0 ...
##  $ NumberOfOpenCreditLinesAndLoans     : int  4 15 12 7 4 4 5 8 4 0 ...
##  $ NumberOfTimes90DaysLate             : int  0 0 0 0 0 0 0 0 0 98 ...
##  $ NumberRealEstateLoansOrLines        : int  0 4 1 2 0 0 0 1 1 0 ...
##  $ NumberOfTime60.89DaysPastDueNotWorse: int  0 0 0 0 0 0 0 0 0 98 ...
##  $ NumberOfDependents                  : int  0 2 2 0 1 1 3 1 0 0 ...
```

```
summary(test)
```

```
##        X          SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines
##  Min.   :     1   Mode:logical     Min.   :    0.000
##  1st Qu.: 25376   NA's:101503      1st Qu.:    0.030
##  Median : 50752                    Median :    0.153
##  Mean   : 50752                    Mean   :    5.310
##  3rd Qu.: 76128                    3rd Qu.:    0.564
##  Max.   :101503                    Max.   :21821.000
##
##       age         NumberOfTime30.59DaysPastDueNotWorse   DebtRatio
```

```
##  Min.   : 21.00   Min.   : 0.0000                      Min.   :     0.00
##  1st Qu.: 41.00   1st Qu.: 0.0000                      1st Qu.:     0.17
##  Median : 52.00   Median : 0.0000                      Median :     0.36
##  Mean   : 52.41   Mean   : 0.4538                      Mean   :   344.48
##  3rd Qu.: 63.00   3rd Qu.: 0.0000                      3rd Qu.:     0.85
##  Max.   :104.00   Max.   :98.0000                      Max.   :268326.00
##
##  MonthlyIncome     NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
##  Min.   :      0   Min.   : 0.000                  Min.   : 0.0000
##  1st Qu.:   3408   1st Qu.: 5.000                  1st Qu.: 0.0000
##  Median :   5400   Median : 8.000                  Median : 0.0000
##  Mean   :   6855   Mean   : 8.454                  Mean   : 0.2967
##  3rd Qu.:   8200   3rd Qu.:11.000                  3rd Qu.: 0.0000
##  Max.   :7727000   Max.   :85.000                  Max.   :98.0000
##  NA's   :20103
##  NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
##  Min.   : 0.000               Min.   : 0.0000
##  1st Qu.: 0.000               1st Qu.: 0.0000
##  Median : 1.000               Median : 0.0000
##  Mean   : 1.013               Mean   : 0.2703
##  3rd Qu.: 2.000               3rd Qu.: 0.0000
##  Max.   :37.000               Max.   :98.0000
##
##  NumberOfDependents
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 0.000
##  Mean   : 0.769
##  3rd Qu.: 1.000
##  Max.   :43.000
##  NA's   :2626
```

There are many `NA`s in the **test** dataset. We need to fix this because missing values are contagious (i.e., the model will predict `NA` for any row that has atleast 1 `NA` for any predictor variable). As seen above in the descriptive statistics, *MonthlyIncome* and *NumberOfDependents* have missing values.

```r
test_replace <- test
test_replace$MonthlyIncome <- replace_NA(test_replace$MonthlyIncome)
test_replace$NumberOfDependents <- replace_NA(test_replace$NumberOfDependents)
```

```r
predictions <- predict(log_model_multi, newdata = test_replace, type = "response")
range(predictions)
```

```
## [1] 2.220446e-16 9.990662e-01
```

```r
test_replace$SeriousDlqin2yrs <- predictions
test_prob <- data.frame(test_replace$X, test_replace$SeriousDlqin2yrs)
names(test_prob)[names(test_prob) == "test_replace.X"] <- "Id"
names(test_prob)[names(test_prob) == "test_replace.SeriousDlqin2yrs"] <- "Probability"
```

Results

We can see the top ten of the predictions :

```
test_prob %>%
  arrange(desc(.$Probability)) %>%
  head(10)
```

```
##       Id Probability
## 1  88036   0.9990662
## 2  43790   0.9949228
## 3  58280   0.9923666
## 4  86490   0.9817607
## 5  67744   0.9817183
## 6  87564   0.9589671
## 7  42663   0.9536939
## 8   1234   0.9331801
## 9  69975   0.9191722
## 10 86860   0.9159203
```

Conclusion

I have implemented a logistic regression model to predict the probability of a customer defaulting on a loan based on applicant information such as age and monthly income. These probabilities can be used to assess risk and determine what interest rate to charge for a loan.