Using Logistic Regression in R

Mike Levine

June 28, 2019

Welcome to Using Logistic Regression in R! In this tutorial, we'll go over a) what logistic regression is, b) its usefulness and c) some examples.

Like linear or multiple regression, logistic regression is a tool for making a binary prediction. In other words, regression can help us make a numeric prediction (like a person's income). Logistic regression, on the other hand, can help us make a binary prediction (such as "will this person buy my product?"). On a more technical and statistical side, the key difference between the two is that logistic regression is logarithmic.

Note: *this assignment was completed through the Fox School of Business at Temple University's Master of Science in Business Analytics program. It was completed through STAT 5607-308 (Advanced Business Statistics) in spring 2019.*

In this tutorial, we'll be using a mock Credit Approval Dataset which is a collection of credit card applications and the credit approval decisions. The variable names are:

- **Approved**: Categorical (+ approved, - not approved)
- **Gender**: Categorical (a-female, b-male)
- **Age**: Continuous
- **Debt**: Continuous
- **Married**: Categorical (l, u, y)
- **YearsEmployed**: Continuous
- **PriorDefault**: Categorical (1-defaulted on prior loan, 0-otherwise)
- **Employed**: Categorical (t-employed, f-not employed)
- **Credit Score**: Continuous
- **Income**: Continuous

Let's start by setting out working directory and pulling in our dataset, credit.csv.

```
setwd("C:/Users/mdlev/OneDrive/Documents/Education/Graduate - Temple
University/2nd Semester/Advanced Business Statistics/R Datasets")

dat <- read.csv("credit.csv", header = T)
```

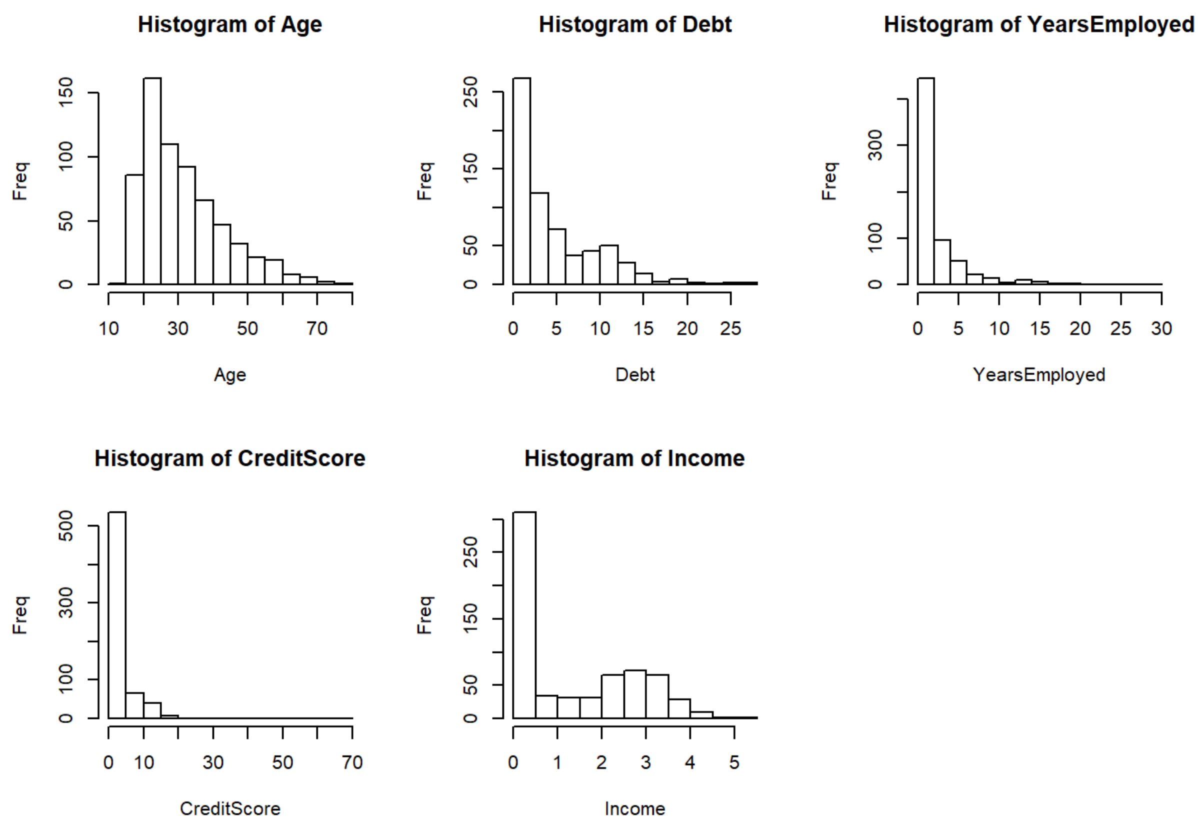
Next, let's look at the means, medians and standard deviations our variables.

```
library(psych)
describe(dat)
```

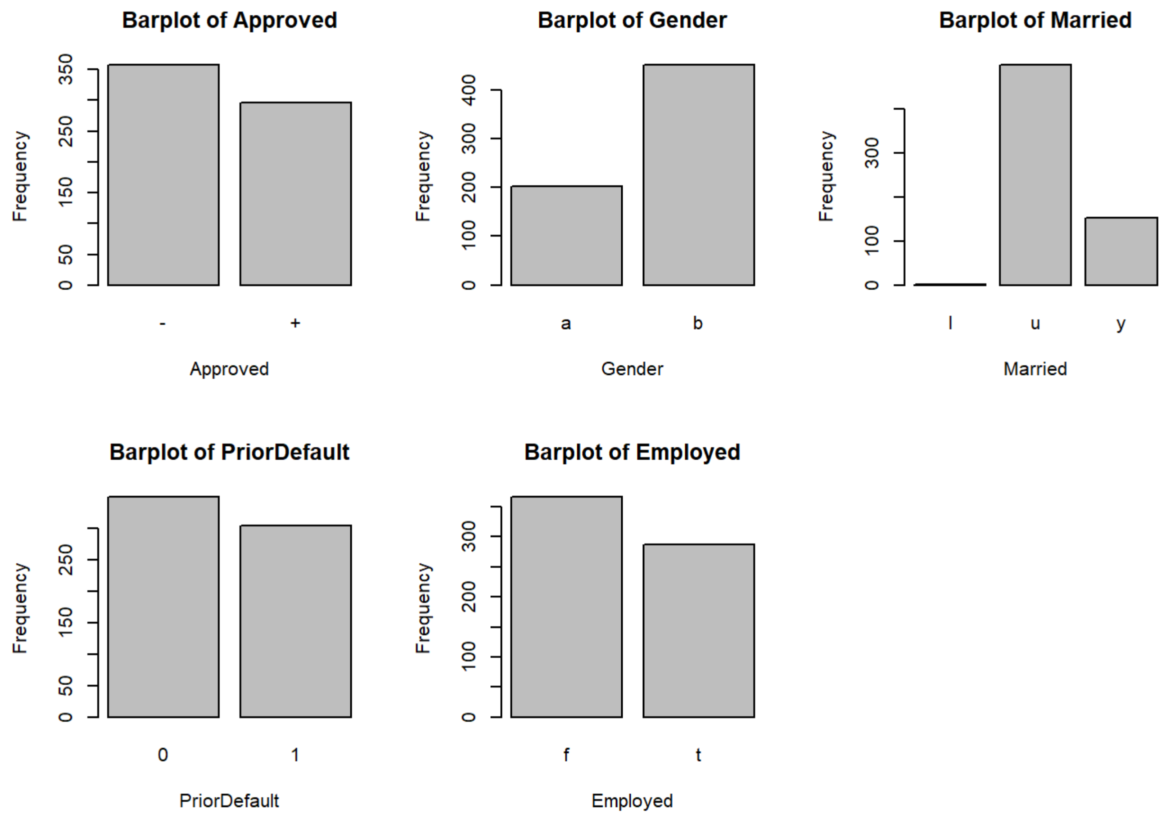
```
##                vars   n  mean    sd median trimmed   mad   min   max range
## Approved*         1 653  1.45  0.50   1.00    1.44  0.00  1.00  2.00   1.0
## Gender*           2 653  1.69  0.46   2.00    1.74  0.00  1.00  2.00   1.0
## Age               3 653 31.50 11.84  28.42   30.05 10.26 13.75 76.75  63.0
## Debt              4 653  4.83  5.03   2.84    4.05  3.34  0.00 28.00  28.0
## Married*          5 653  2.23  0.43   2.00    2.17  0.00  1.00  3.00   2.0
## YearsEmployed     6 653  2.24  3.37   1.00    1.50  1.36  0.00 28.50  28.5
## PriorDefault      7 653  0.47  0.50   0.00    0.46  0.00  0.00  1.00   1.0
## Employed*         8 653  1.44  0.50   1.00    1.42  0.00  1.00  2.00   1.0
## CreditScore       9 653  2.50  4.97   0.00    1.41  0.00  0.00 67.00  67.0
## Income           10 653  1.30  1.38   0.78    1.17  1.15  0.00  5.00   5.0
##                 skew kurtosis   se
## Approved*       0.19    -1.97 0.02
## Gender*        -0.82    -1.34 0.02
## Age             1.07     0.79 0.46
## Debt            1.48     2.19 0.20
## Married*        1.16    -0.29 0.02
## YearsEmployed   2.90    11.20 0.13
## PriorDefault    0.14    -1.98 0.02
## Employed*       0.24    -1.94 0.02
## CreditScore     5.03    48.26 0.19
## Income          0.49    -1.26 0.05
```

Let’s also create some graphs illustrating the distribution of each of our variables.

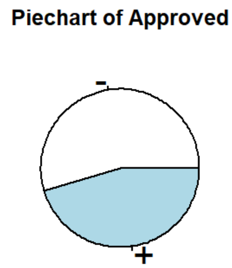
```
par(mfrow=c(2,3))
hist(dat$Age, main = "Histogram of Age", ylab = "Freq", xlab = "Age")
hist(dat$Debt, main = "Histogram of Debt", ylab = "Freq", xlab = "Debt")
hist(dat$YearsEmployed, main = "Histogram of YearsEmployed", ylab = "Freq",
xlab = "YearsEmployed")
hist(dat$CreditScore, main = "Histogram of CreditScore", ylab = "Freq", xlab
= "CreditScore")
hist(dat$Income, main = "Histogram of Income", ylab = "Freq", xlab =
"Income")

tab1 <- table(dat$Approved)
tab2 <- table(dat$Gender)
tab3 <- table(dat$Married)
tab4 <- table(dat$PriorDefault)
tab5 <- table(dat$Employed)

par(mfrow=c(2,3))
```

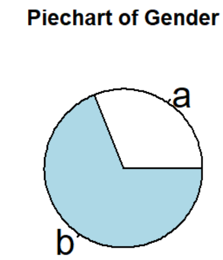


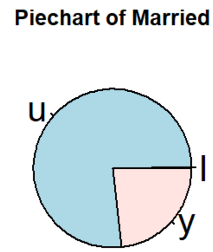

```
barplot(tab1, main = "Barplot of Approved", ylab = "Frequency", xlab =
"Approved")
barplot(tab2, main = "Barplot of Gender", ylab = "Frequency", xlab =
"Gender")
barplot(tab3, main = "Barplot of Married", ylab = "Frequency", xlab =
"Married")
barplot(tab4, main = "Barplot of PriorDefault", ylab = "Frequency", xlab =
"PriorDefault")
barplot(tab5, main = "Barplot of Employed", ylab = "Frequency", xlab =
"Employed")

par(mfrow=c(2,3))
```

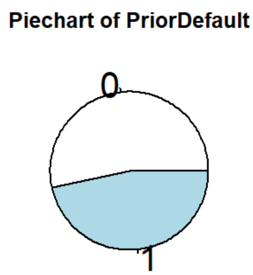


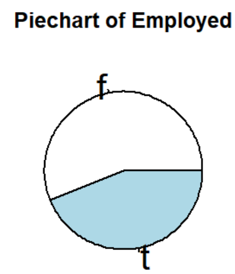

```
pie(tab1, main = "Piechart of Approved", cex=2)
pie(tab2, main = "Piechart of Gender", cex=2)
pie(tab3, main = "Piechart of Married", cex=2)
pie(tab4, main = "Piechart of PriorDefault", cex=2)
pie(tab5, main = "Piechart of Employed", cex=2)
```











Next, let’s relevel our ‘Approved’ variable to make sure that ‘Not Approved’ is first. In our ‘Approved’ variable, a ‘+’ means approved and a ‘-’ means not approved.

```
dat$Approved <- relevel(dat$Approved, "-")
```

Now, let’s run an initial logistic regression model to predict the ‘Approved’ variable. Looking at our model’s output, we can see that the most significant variables are PriorDefault and Income (followed by CreditScore and YearsEmployed). We know this based off the p-values being less than 0.05 for those variables.

```
Credit_mod1 <- glm(Approved ~ ., data = dat, family  = binomial())
summary(Credit_mod1)
##
## Call:
## glm(formula = Approved ~ ., family = binomial(), data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6304  -0.3409  -0.1826   0.5103   2.8867
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    17.136775 618.705945   0.028 0.977903
## Genderb         0.126112   0.276759   0.456 0.648623
## Age            -0.011016   0.011902  -0.926 0.354672
## Debt           -0.007309   0.025812  -0.283 0.777057
```

```
## Marriedu      -16.571826 618.705884  -0.027 0.978631
## Marriedy      -17.373602 618.705929  -0.028 0.977598
## YearsEmployed   0.110857   0.049567   2.237 0.025319 *
## PriorDefault   -3.785195   0.315442 -12.000  < 2e-16 ***
## Employedt       0.373770   0.344159   1.086 0.277461
## CreditScore     0.115160   0.056596   2.035 0.041875 *
## Income          0.366948   0.101115   3.629 0.000285 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 899.54  on 652  degrees of freedom
## Residual deviance: 415.30  on 642  degrees of freedom
## AIC: 437.3
##
## Number of Fisher Scoring iterations: 13
```

Now, let's run a new logistic regression model with only the variables we found significant in our first model. This time, all four variables we selected are significant in predicting the approval status. The p-values are below 0.05 for each.

```
Credit_mod2 <- glm(Approved ~ YearsEmployed + PriorDefault + CreditScore +
Income, data = dat, family  = binomial())
summary(Credit_mod2)
##
## Call:
## glm(formula = Approved ~ YearsEmployed + PriorDefault + CreditScore +
##     Income, family = binomial(), data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8440  -0.3666  -0.2494   0.5273   2.6424
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.13795    0.21194   0.651 0.515118
## YearsEmployed  0.11564    0.04707   2.457 0.014006 *
## PriorDefault  -3.60280    0.29461 -12.229  < 2e-16 ***
## CreditScore    0.15330    0.04576   3.350 0.000809 ***
## Income         0.38850    0.09699   4.006 6.19e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 899.54  on 652  degrees of freedom
## Residual deviance: 432.99  on 648  degrees of freedom
## AIC: 442.99
##
## Number of Fisher Scoring iterations: 6
```

Next, let's run the Chi-squared test to check for overall model significance. In looking at the output of our code below, we can see that the p-value is zero. This tells us that the model is useful.

```
modelChi <- Credit_mod2$null.deviance - Credit_mod2$deviance
chidf <- Credit_mod2$df.null -Credit_mod2$df.residual
chisq.prob <- 1 - pchisq(modelChi, chidf)
chisq.prob
## [1] 0
```

Next, let's calculate the odds ratio (OR) for each of the variable coefficients. This will help us to make our model a little more interpretable for a stakeholder.

```
Credit_mod2$coeff
##   (Intercept) YearsEmployed  PriorDefault   CreditScore        Income
##     0.1379474     0.1156439    -3.6027970     0.1532995     0.3885040
exp(Credit_mod2$coeff)
##   (Intercept) YearsEmployed  PriorDefault   CreditScore        Income
##     1.1479152     1.1225961     0.0272474     1.1656741     1.4747729
```

In looking at our odds ratio output above, we can tell a few things.

First, the odds of a credit applicant being approved, who has more work experience, are 1.12 times higher than those of a credit applicant who has less work experience.

The odds of a credit applicant being approved, who previously defaulted on a loan, are 0.03 times higher than those of a credit applicant who did not previously default on a loan.

The odds of a credit applicant being approved, who has a higher credit score, are 1.17 times higher than those of a credit applicant who has a lower credit score.

The odds of a credit applicant being approved, who has a higher income, are 1.47 times higher than those of a credit applicant who has a lower income.

Lastly, let's look at our confidence intervals from above. Below we have the ranges of what our predictions can be.

```
library(MASS)
exp(confint(Credit_mod2))
## Waiting for profiling to be done...
##                     2.5 %    97.5 %
## (Intercept)   0.75825231 1.7433283
## YearsEmployed 1.02805991 1.2366219
## PriorDefault  0.01486118 0.0473895
## CreditScore   1.07017265 1.2803066
## Income        1.22379259 1.7916339
```