Bicheng (Tony) Zhang

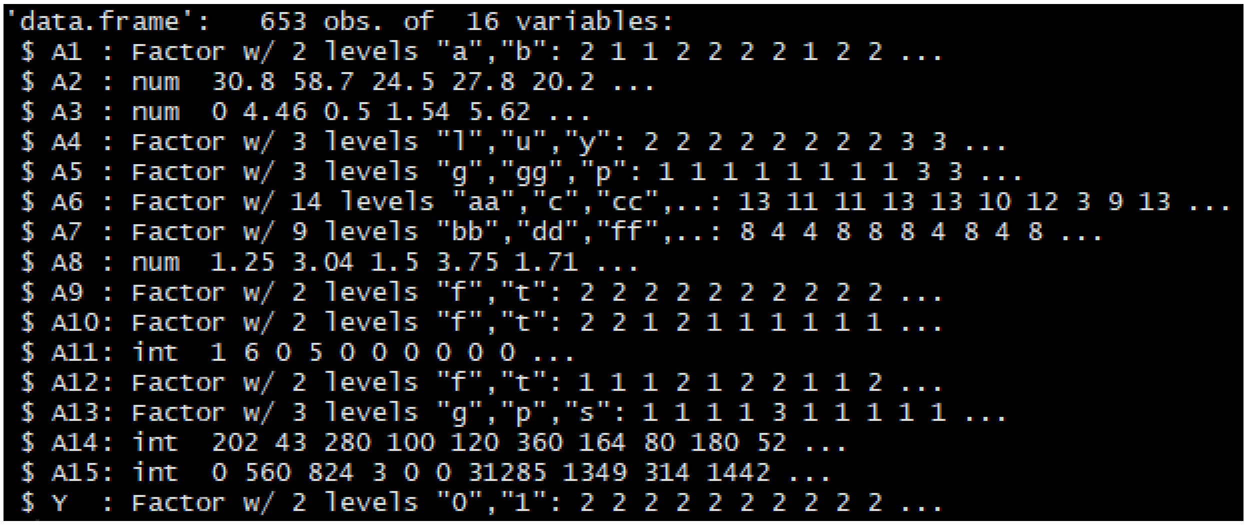
Introduction:

Applying for credit is an important and inevitable part of life, especially for young adults. Building a good credit history is paramount to ensure that you open yourself to more financial possibilities in the future, such as buying a car or house or starting your own business, and in case of unforeseen emergencies and expenses you may have more credit available to help get through. Your credit history may even be compared/considered with other applicants when applying for a job! These are some of the motivations for undertaking a report/analysis on a dataset that contains acceptance/rejection results and information from credit card applications.

Exploratory Data Analysis:

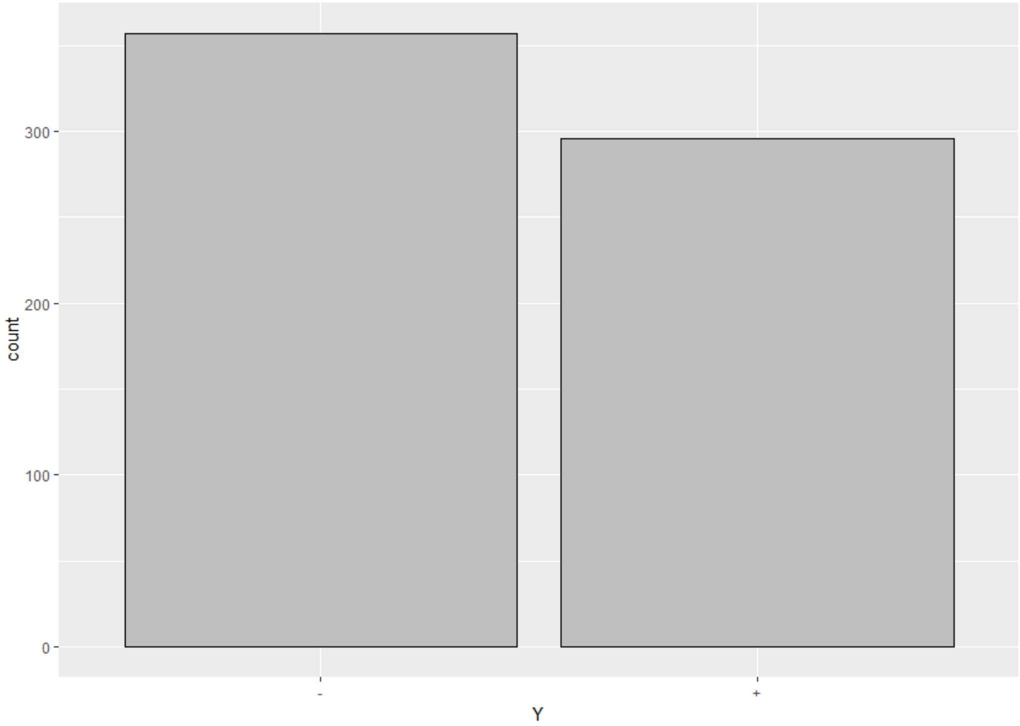
The dataset was obtained from the University of California, Irvine, Machine Learning Repository website, and the data file is called “crx.data.” There is a total of 690 observations, with 16 different predictor variables. In the data file, the variables/columns are separated by comma values. There are a mix of continuous and categorical predictor variables, and the last column (response variable) takes two values, either “+” or “-”. A value of “+” indicates the observation’s application for a credit card was approved while a value of “-” indicates that it was not approved. Missing values have been marked with “?” symbols. Because the file concerns sensitive information, all names and variables/levels have been changed to meaningless symbols so that the data is completely anonymized.

Because the data is anonymized, imputation for the missing values will not be considered, and I will just delete any observation with a “?” value. After cleaning the dataset, a description of the dataset was obtained below using R (removed 37 observations with missing values):



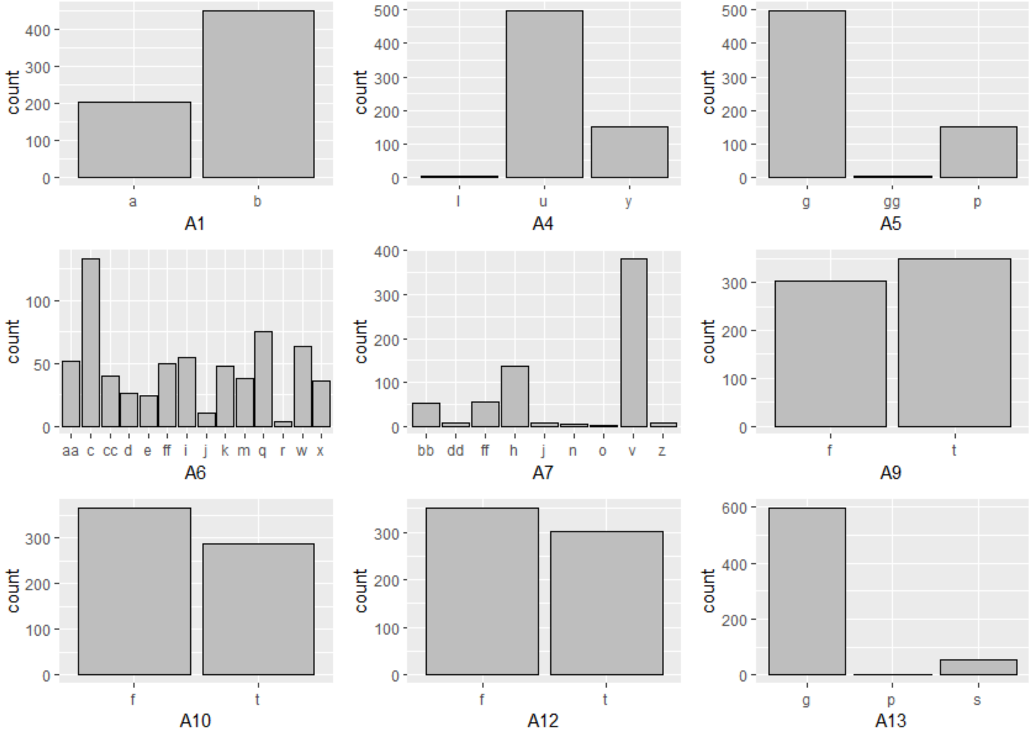
```
'data.frame':   653 obs. of  16 variables:
 $ A1 : Factor w/ 2 levels "a","b": 2 1 1 2 2 2 2 1 2 2 ...
 $ A2 : num  30.8 58.7 24.5 27.8 20.2 ...
 $ A3 : num  0 4.46 0.5 1.54 5.62 ...
 $ A4 : Factor w/ 3 levels "l","u","y": 2 2 2 2 2 2 2 2 3 3 ...
 $ A5 : Factor w/ 3 levels "g","gg","p": 1 1 1 1 1 1 1 1 3 3 ...
 $ A6 : Factor w/ 14 levels "aa","c","cc",..: 13 11 11 13 13 10 12 3 9 13 ...
 $ A7 : Factor w/ 9 levels "bb","dd","ff",..: 8 4 4 8 8 8 4 8 4 8 ...
 $ A8 : num  1.25 3.04 1.5 3.75 1.71 ...
 $ A9 : Factor w/ 2 levels "f","t": 2 2 2 2 2 2 2 2 2 2 ...
 $ A10: Factor w/ 2 levels "f","t": 2 2 1 2 1 1 1 1 1 1 ...
 $ A11: int  1 6 0 5 0 0 0 0 0 0 ...
 $ A12: Factor w/ 2 levels "f","t": 1 1 1 2 1 2 2 1 1 2 ...
 $ A13: Factor w/ 3 levels "g","p","s": 1 1 1 1 3 1 1 1 1 1 ...
 $ A14: int  202 43 280 100 120 360 164 80 180 52 ...
 $ A15: int  0 560 824 3 0 0 31285 1349 314 1442 ...
 $ Y  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

Response variable:



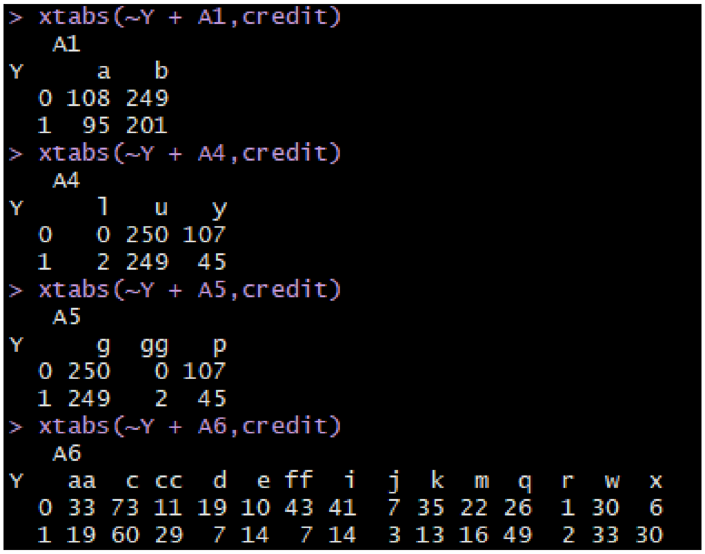
There are 357 instances of applications being rejected, and 296 applications accepted.

Categorical variables:

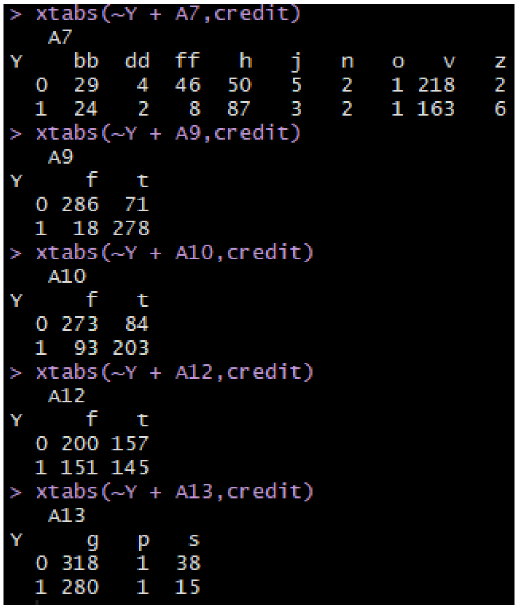


There appears to be several variables where there is only one or a very small number of observation(s) of a certain level. Specifically, in the variables A4 (‘l’), A5 (‘gg’), A6 (‘r’), A7 (‘dd’, ‘j’, ‘n’, ‘o’, and ‘z’), and A13 (‘p’).

Next, I wanted to check the levels of each categorical variable and how they contribute to our response:

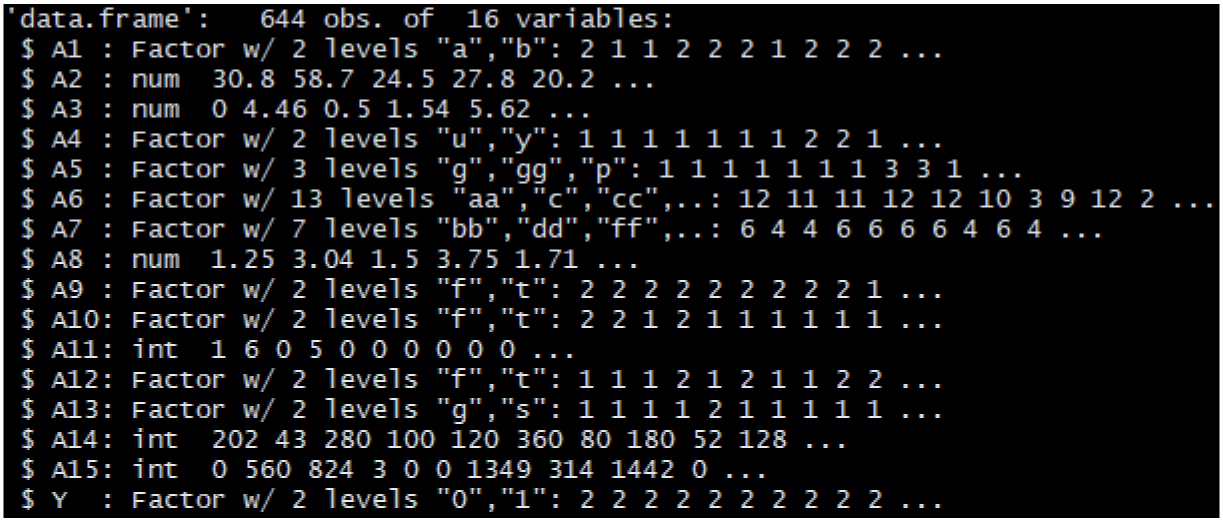


```
> xtabs(~Y + A1,credit)
   A1
Y     a   b
  0 108 249
  1  95 201
> xtabs(~Y + A4,credit)
   A4
Y     l   u   y
  0   0 250 107
  1   2 249  45
> xtabs(~Y + A5,credit)
   A5
Y     g  gg   p
  0 250   0 107
  1 249   2  45
> xtabs(~Y + A6,credit)
   A6
Y   aa  c cc  d  e ff  i  j  k  m  q  r  w  x
  0 33 73 11 19 10 43 41  7 35 22 26  1 30  6
  1 19 60 29  7 14  7 14  3 13 16 49  2 33 30
```



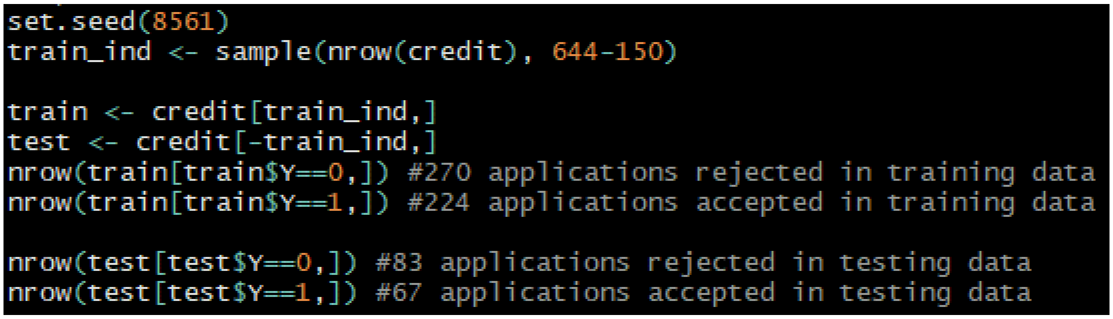
```
> xtabs(~Y + A7,credit)
   A7
Y    bb  dd  ff   h   j   n   o   v   z
  0  29   4  46  50   5   2   1 218   2
  1  24   2   8  87   3   2   1 163   6
> xtabs(~Y + A9,credit)
   A9
Y     f   t
  0 286  71
  1  18 278
> xtabs(~Y + A10,credit)
   A10
Y     f   t
  0 273  84
  1  93 203
> xtabs(~Y + A12,credit)
   A12
Y     f   t
  0 200 157
  1 151 145
> xtabs(~Y + A13,credit)
   A13
Y     g   p   s
  0 318   1  38
  1 280   1  15
```

From the crosstabs of the categorical variables and the response variable above, I decided to remove some more observations with some low numbers in the above crosstabs. Specifically, in variables A4, A5, A6, A7 & A13 the levels “l”, “gg”, “r”, “n” & “o”, and “p” respectively were all dropped from the dataset. This resulted in 9 more observations being removed, and a total of 46 observations including those with missing values removed from the original dataset, resulting in our final dataset below:



```
'data.frame':   644 obs. of  16 variables:
 $ A1 : Factor w/ 2 levels "a","b": 2 1 1 2 2 2 1 2 2 2 ...
 $ A2 : num  30.8 58.7 24.5 27.8 20.2 ...
 $ A3 : num  0 4.46 0.5 1.54 5.62 ...
 $ A4 : Factor w/ 2 levels "u","y": 1 1 1 1 1 1 1 2 2 1 ...
 $ A5 : Factor w/ 3 levels "g","gg","p": 1 1 1 1 1 1 1 3 3 1 ...
 $ A6 : Factor w/ 13 levels "aa","c","cc",..: 12 11 11 12 12 10 3 9 12 2 ...
 $ A7 : Factor w/ 7 levels "bb","dd","ff",..: 6 4 4 6 6 6 6 4 6 4 ...
 $ A8 : num  1.25 3.04 1.5 3.75 1.71 ...
 $ A9 : Factor w/ 2 levels "f","t": 2 2 2 2 2 2 2 2 2 1 ...
 $ A10: Factor w/ 2 levels "f","t": 2 2 1 2 1 1 1 1 1 1 ...
 $ A11: int  1 6 0 5 0 0 0 0 0 0 ...
 $ A12: Factor w/ 2 levels "f","t": 1 1 1 2 1 2 1 1 2 2 ...
 $ A13: Factor w/ 2 levels "g","s": 1 1 1 1 2 1 1 1 1 1 ...
 $ A14: int  202 43 280 100 120 360 80 180 52 128 ...
 $ A15: int  0 560 824 3 0 0 1349 314 1442 0 ...
 $ Y  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

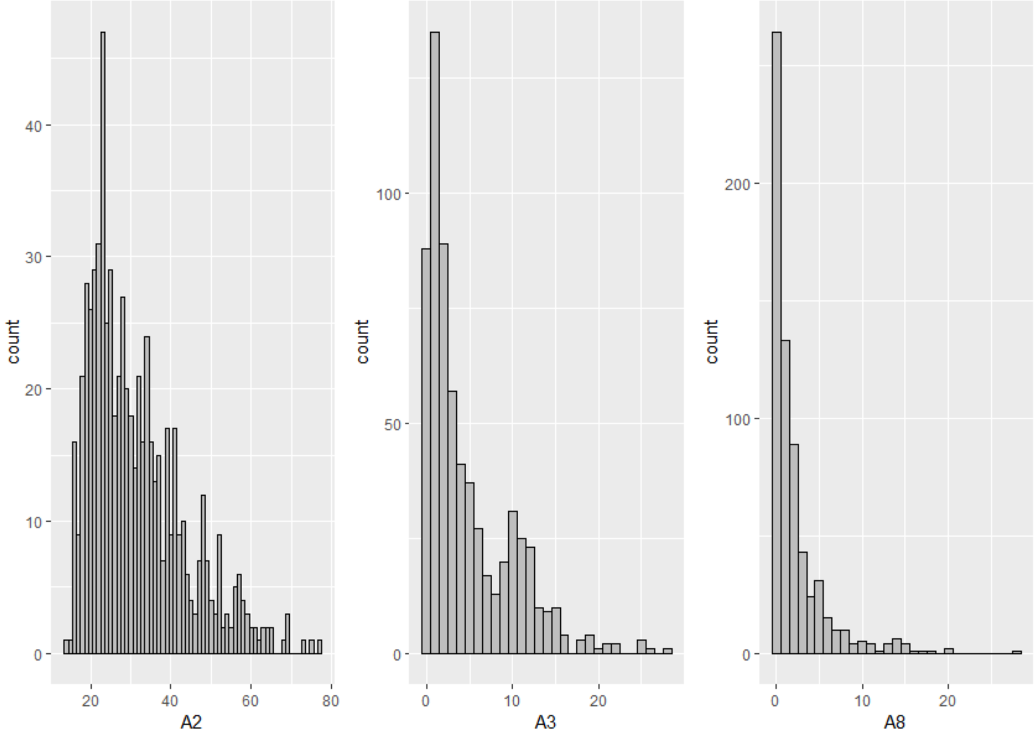
Next, I split the data two datasets, one for training to build a model and the other for testing and cross validation purposes:

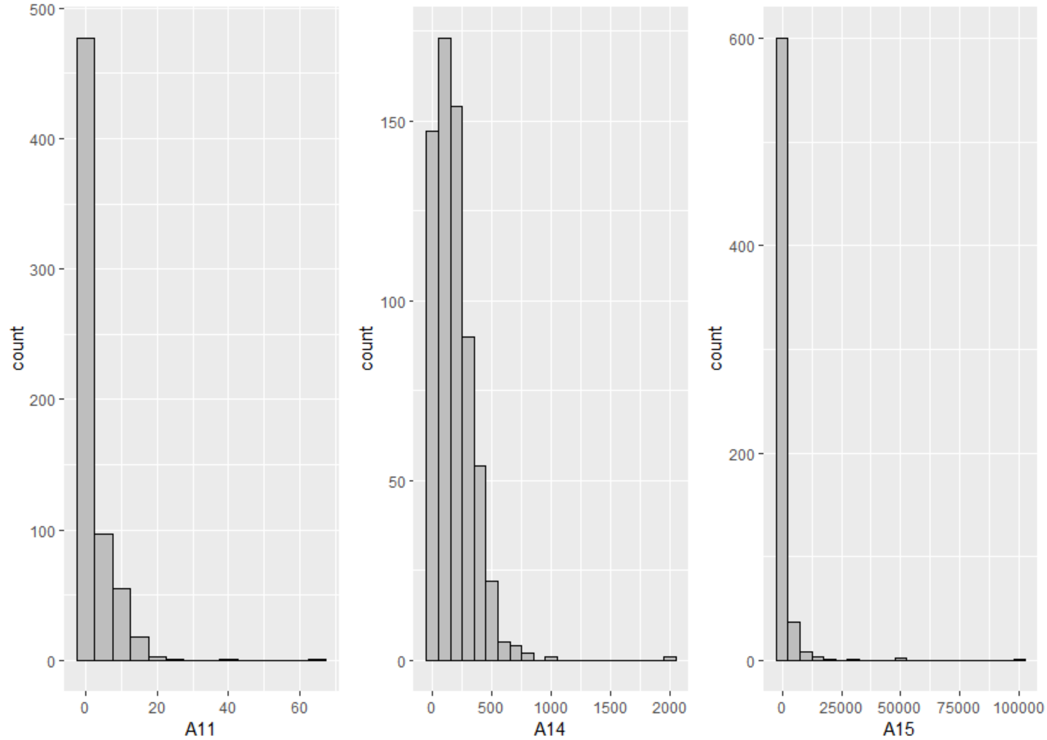


```
set.seed(8561)
train_ind <- sample(nrow(credit), 644-150)

train <- credit[train_ind,]
test <- credit[-train_ind,]
nrow(train[train$Y==0,]) #270 applications rejected in training data
nrow(train[train$Y==1,]) #224 applications accepted in training data

nrow(test[test$Y==0,]) #83 applications rejected in testing data
nrow(test[test$Y==1,]) #67 applications accepted in testing data
```

I set the test data set to have exactly 150 observations, and the training data set to have 494.

Continuous variables:

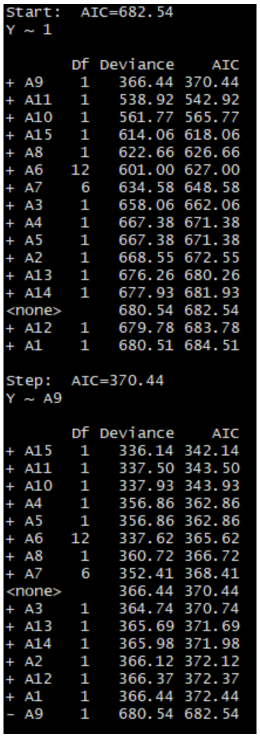




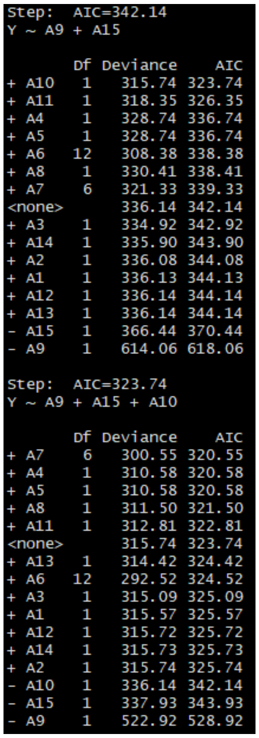
The distributions for these variables are all right skewed, with outliers obvious in the last 3.

Model Selection:

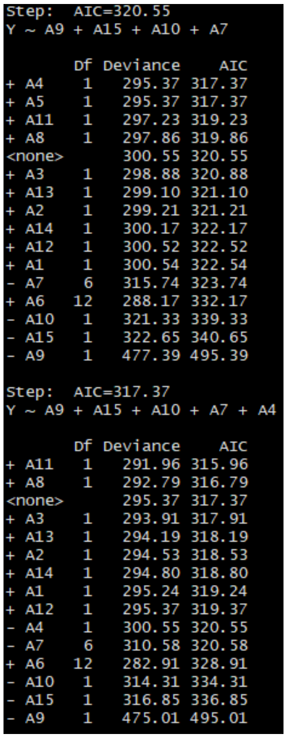
The stepwise model selection procedure was then used with the training data, starting with the null model with only the intercept term as a predictor, and we consider all 15 predictor variables.



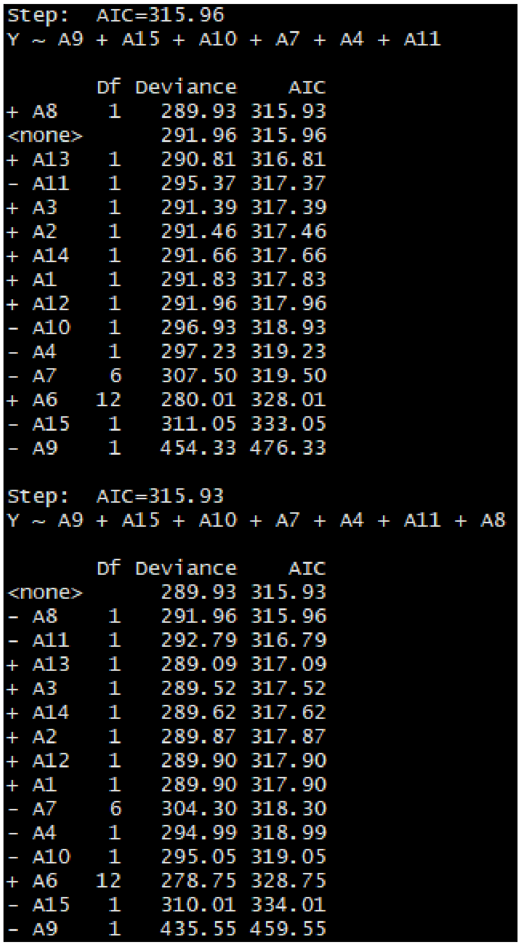
```
Start:  AIC=682.54
Y ~ 1

       Df Deviance    AIC
+ A9    1   366.44 370.44
+ A11   1   538.92 542.92
+ A10   1   561.77 565.77
+ A15   1   614.06 618.06
+ A8    1   622.66 626.66
+ A6   12   601.00 627.00
+ A7    6   634.58 648.58
+ A3    1   658.06 662.06
+ A4    1   667.38 671.38
+ A5    1   667.38 671.38
+ A2    1   668.55 672.55
+ A13   1   676.26 680.26
+ A14   1   677.93 681.93
<none>      680.54 682.54
+ A12   1   679.78 683.78
+ A1    1   680.51 684.51

Step:  AIC=370.44
Y ~ A9

       Df Deviance    AIC
+ A15   1   336.14 342.14
+ A11   1   337.50 343.50
+ A10   1   337.93 343.93
+ A4    1   356.86 362.86
+ A5    1   356.86 362.86
+ A6   12   337.62 365.62
+ A8    1   360.72 366.72
+ A7    6   352.41 368.41
<none>      366.44 370.44
+ A3    1   364.74 370.74
+ A13   1   365.69 371.69
+ A14   1   365.98 371.98
+ A2    1   366.12 372.12
+ A12   1   366.37 372.37
+ A1    1   366.44 372.44
- A9    1   680.54 682.54
```



```
Step:  AIC=342.14
Y ~ A9 + A15

       Df Deviance    AIC
+ A10   1   315.74 323.74
+ A11   1   318.35 326.35
+ A4    1   328.74 336.74
+ A5    1   328.74 336.74
+ A6   12   308.38 338.38
+ A8    1   330.41 338.41
+ A7    6   321.33 339.33
<none>      336.14 342.14
+ A3    1   334.92 342.92
+ A14   1   335.90 343.90
+ A2    1   336.08 344.08
+ A1    1   336.13 344.13
+ A12   1   336.14 344.14
+ A13   1   336.14 344.14
- A15   1   366.44 370.44
- A9    1   614.06 618.06

Step:  AIC=323.74
Y ~ A9 + A15 + A10

       Df Deviance    AIC
+ A7    6   300.55 320.55
+ A4    1   310.58 320.58
+ A5    1   310.58 320.58
+ A8    1   311.50 321.50
+ A11   1   312.81 322.81
<none>      315.74 323.74
+ A13   1   314.42 324.42
+ A6   12   292.52 324.52
+ A3    1   315.09 325.09
+ A1    1   315.57 325.57
+ A12   1   315.72 325.72
+ A14   1   315.73 325.73
+ A2    1   315.74 325.74
- A10   1   336.14 342.14
- A15   1   337.93 343.93
- A9    1   522.92 528.92
```



```
Step:  AIC=320.55
Y ~ A9 + A15 + A10 + A7

       Df Deviance    AIC
+ A4    1   295.37 317.37
+ A5    1   295.37 317.37
+ A11   1   297.23 319.23
+ A8    1   297.86 319.86
<none>      300.55 320.55
+ A3    1   298.88 320.88
+ A13   1   299.10 321.10
+ A2    1   299.21 321.21
+ A14   1   300.17 322.17
+ A12   1   300.52 322.52
+ A1    1   300.54 322.54
- A7    6   315.74 323.74
+ A6   12   288.17 332.17
- A10   1   321.33 339.33
- A15   1   322.65 340.65
- A9    1   477.39 495.39

Step:  AIC=317.37
Y ~ A9 + A15 + A10 + A7 + A4

       Df Deviance    AIC
+ A11   1   291.96 315.96
+ A8    1   292.79 316.79
<none>      295.37 317.37
+ A3    1   293.91 317.91
+ A13   1   294.19 318.19
+ A2    1   294.53 318.53
+ A14   1   294.80 318.80
+ A1    1   295.24 319.24
+ A12   1   295.37 319.37
- A4    1   300.55 320.55
- A7    6   310.58 320.58
+ A6   12   282.91 328.91
- A10   1   314.31 334.31
- A15   1   316.85 336.85
- A9    1   475.01 495.01
```

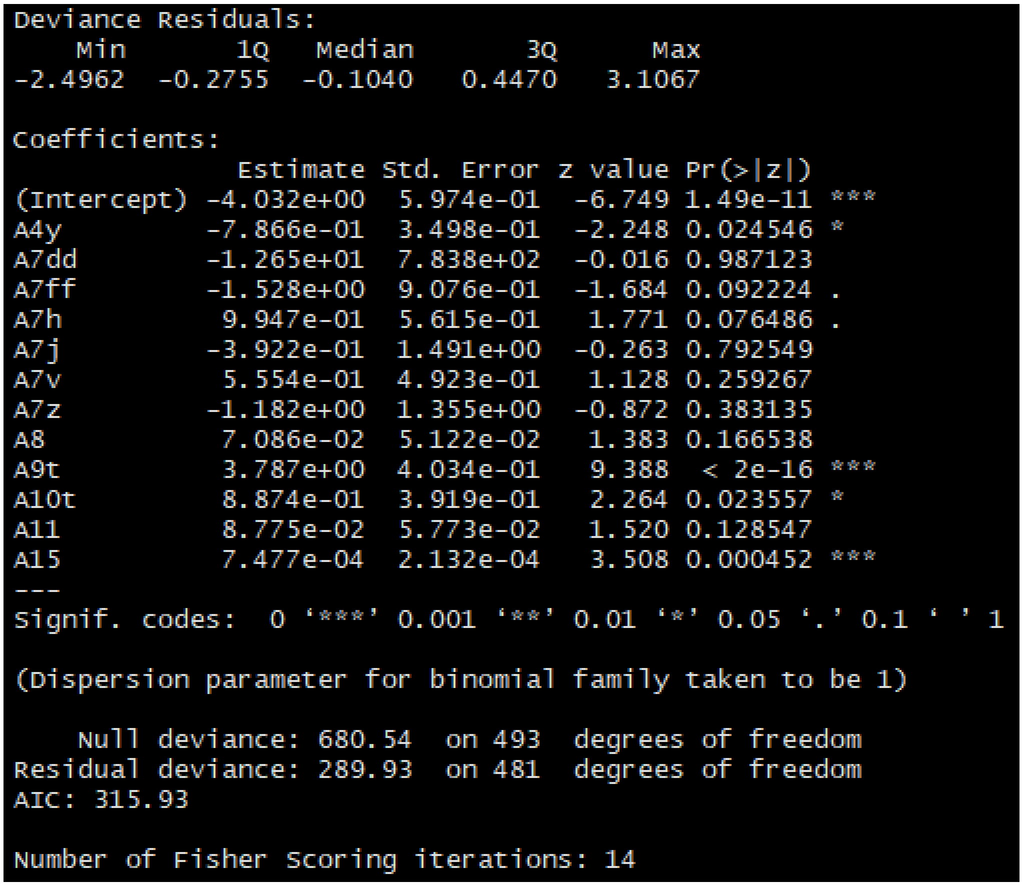


```
Step:  AIC=315.96
Y ~ A9 + A15 + A10 + A7 + A4 + A11

       Df Deviance    AIC
+ A8    1   289.93 315.93
<none>      291.96 315.96
+ A13   1   290.81 316.81
- A11   1   295.37 317.37
+ A3    1   291.39 317.39
+ A2    1   291.46 317.46
+ A14   1   291.66 317.66
+ A1    1   291.83 317.83
+ A12   1   291.96 317.96
- A10   1   296.93 318.93
- A4    1   297.23 319.23
- A7    6   307.50 319.50
+ A6   12   280.01 328.01
- A15   1   311.05 333.05
- A9    1   454.33 476.33

Step:  AIC=315.93
Y ~ A9 + A15 + A10 + A7 + A4 + A11 + A8

       Df Deviance    AIC
<none>      289.93 315.93
- A8    1   291.96 315.96
- A11   1   292.79 316.79
+ A13   1   289.09 317.09
+ A3    1   289.52 317.52
+ A14   1   289.62 317.62
+ A2    1   289.87 317.87
+ A12   1   289.90 317.90
+ A1    1   289.90 317.90
- A7    6   304.30 318.30
- A4    1   294.99 318.99
- A10   1   295.05 319.05
+ A6   12   278.75 328.75
- A15   1   310.01 334.01
- A9    1   435.55 459.55
```

After 7 steps, the model with the lowest Akaike Information Criterion score of 315.98 was the one with the seven predictors A4, A7, A8, A9, A10, A11, and A15.

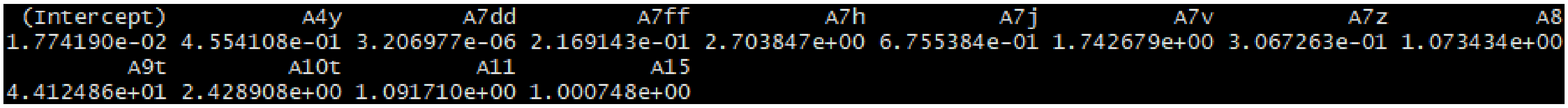
Model Fit and Interpretation:

Below is the resulting logistic regression model from the stepwise selection procedure:



```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4962  -0.2755  -0.1040   0.4470   3.1067

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.032e+00  5.974e-01  -6.749 1.49e-11 ***
A4y         -7.866e-01  3.498e-01  -2.248 0.024546 *
A7dd        -1.265e+01  7.838e+02  -0.016 0.987123
A7ff        -1.528e+00  9.076e-01  -1.684 0.092224 .
A7h          9.947e-01  5.615e-01   1.771 0.076486 .
A7j         -3.922e-01  1.491e+00  -0.263 0.792549
A7v          5.554e-01  4.923e-01   1.128 0.259267
A7z         -1.182e+00  1.355e+00  -0.872 0.383135
A8           7.086e-02  5.122e-02   1.383 0.166538
A9t          3.787e+00  4.034e-01   9.388  < 2e-16 ***
A10t         8.874e-01  3.919e-01   2.264 0.023557 *
A11          8.775e-02  5.773e-02   1.520 0.128547
A15          7.477e-04  2.132e-04   3.508 0.000452 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 680.54  on 493  degrees of freedom
Residual deviance: 289.93  on 481  degrees of freedom
AIC: 315.93

Number of Fisher Scoring iterations: 14
```

We can convert the coefficients to odds ratios (shown below) to more easily interpret them:



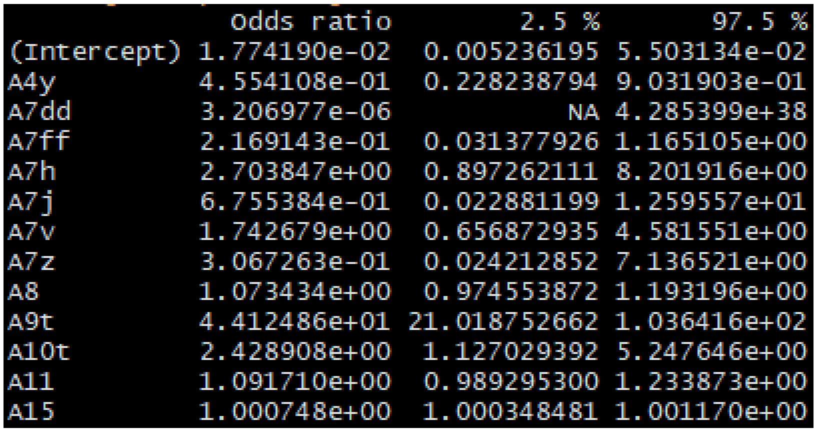
```
 (Intercept)          A4y         A7dd         A7ff          A7h          A7j          A7v          A7z           A8
1.774190e-02 4.554108e-01 3.206977e-06 2.169143e-01 2.703847e+00 6.755384e-01 1.742679e+00 3.067263e-01 1.073434e+00
         A9t         A10t          A11          A15
4.412486e+01 2.428908e+00 1.091710e+00 1.000748e+00
```

Holding all other variables constant:

- Being in the "y" category vs being in the "u" category for the A4 variable decreases odds of being accepted by $e^{-0.07866}$ = 0.4554 (odds for someone in "y" to be accepted 54.46% lower than someone in "u" to be accepted)
- Being in the "ff" and "h" category vs being in the "bb" category for the A7 variable decrease and increase odds of being accepted by 0.2169 (-78.31%) and 2.704 (170.4%) respectively, and so on…
- For every 1 unit increase in the A8 variable, the odds of being accepted increase by 1.0734 or 7.34%
- Being in the "t" category vs being in the "f" category for the A9 and A10 variable increase odds of being accepted by 44.12 (4312%) and 2.4289 (142.89%) respectively

- For every 1 unit increase in the A11 variable, the odds of being accepted increase by 1.0917 or 9.17%
- For every 1 unit increase in the A15 variable, the odds of being accepted stay almost the same (increase by 0.0748%)

Below are the 95% confidence intervals for these odds ratios:



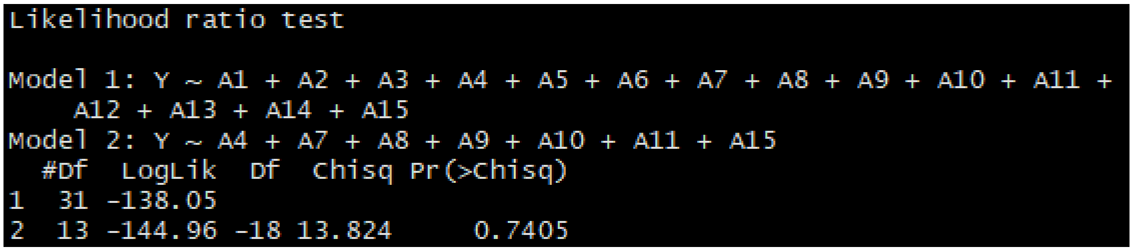
```
              Odds ratio        2.5 %       97.5 %
(Intercept) 1.774190e-02  0.005236195 5.503134e-02
A4y         4.554108e-01  0.228238794 9.031903e-01
A7dd        3.206977e-06           NA 4.285399e+38
A7ff        2.169143e-01  0.031377926 1.165105e+00
A7h         2.703847e+00  0.897262111 8.201916e+00
A7j         6.755384e-01  0.022881199 1.259557e+01
A7v         1.742679e+00  0.656872935 4.581551e+00
A7z         3.067263e-01  0.024212852 7.136521e+00
A8          1.073434e+00  0.974553872 1.193196e+00
A9t         4.412486e+01 21.018752662 1.036416e+02
A10t        2.428908e+00  1.127029392 5.247646e+00
A11         1.091710e+00  0.989295300 1.233873e+00
A15         1.000748e+00  1.000348481 1.001170e+00
```

The resulting CI for the odds ratio of level “dd” for A7 here can be ignored, there are only 6 observations in this level so its results will not be very meaningful.

Goodness of Fit:

Likelihood Ratio Test:

Next, I wanted to run a likelihood ratio test comparing this reduced model that resulted from the stepwise selection procedure to the full model with all 15 predictor variables. The null hypothesis to test is that this resulting model is true. So, in this case, if we get a small p-value and reject the null, it would provide evidence against using this reduced model in favor of the full model with all 15 predictors.



```
Likelihood ratio test

Model 1: Y ~ A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10 + A11 +
    A12 + A13 + A14 + A15
Model 2: Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15
  #Df  LogLik  Df  Chisq Pr(>Chisq)
1  31 -138.05
2  13 -144.96 -18 13.824     0.7405
```

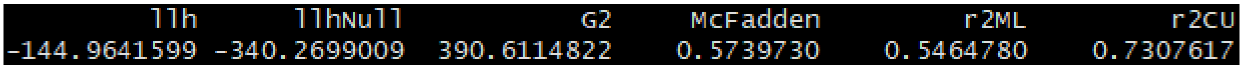
The p-value is 0.7405, meaning that we fail to reject H_0, and cannot conclude that the full model with all 15 predictors is better than the reduced model resulting from the stepwise selection procedure.

McFadden’s Pseudo R^2:

This pseudo R^2 ranges from 0 to just under 1, and is defined as:

$$1 - \frac{log(fit)}{log(null)}$$

where log(fit) is the log likelihood value for our fitted model and log(null) is log likelihood value for the null model.

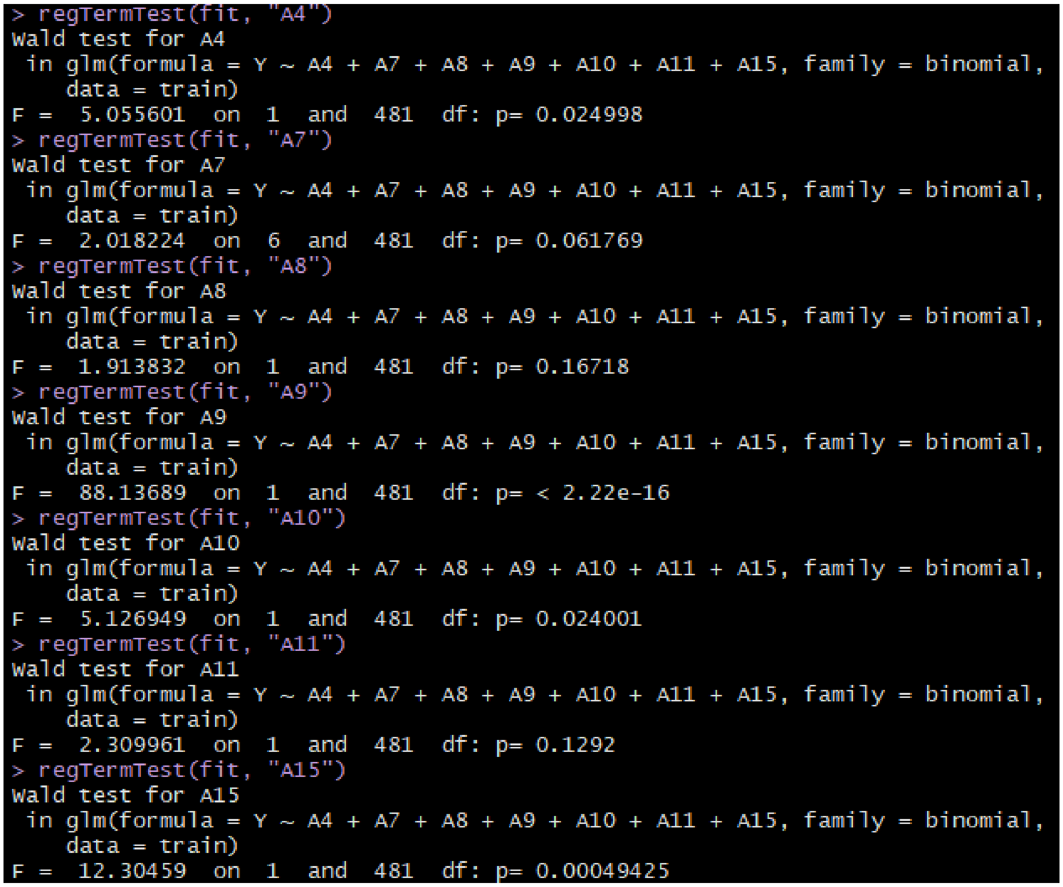


```
         llh      llhNull           G2     McFadden         r2ML         r2CU
-144.9641599 -340.2699009  390.6114822    0.5739730    0.5464780    0.7307617
```

From the above R output, the McFadden’s R^2 for our fitted model is 0.573973, telling us that our selected model is a good fit for predicting credit card application acceptance.

Wald Test:

The Wald test is conducted by taking the ratio of the square of the regression coefficient and the square of its standard error. It tells us the statistical significance of each coefficient in our model.

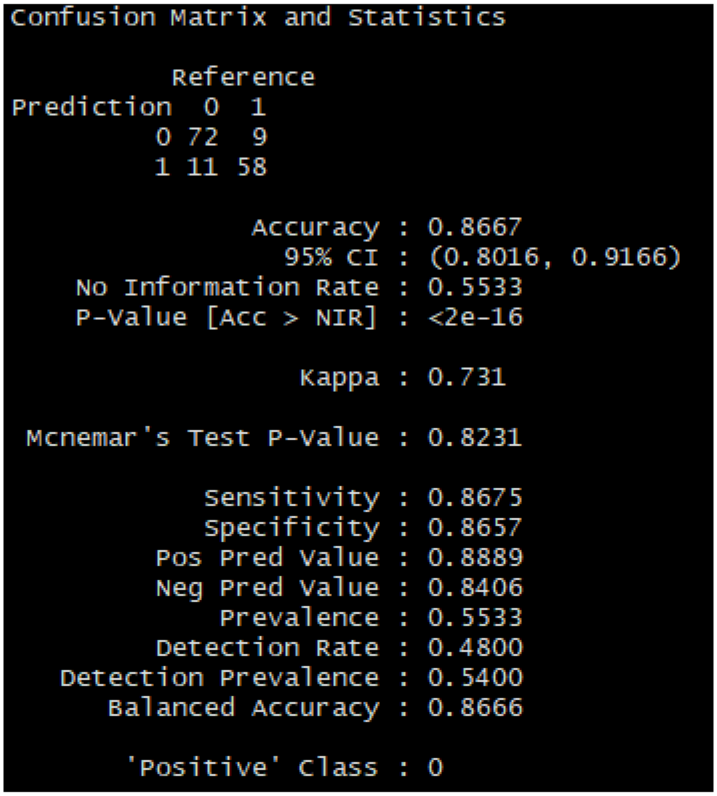


```
> regTermTest(fit, "A4")
Wald test for A4
 in glm(formula = Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15, family = binomial,
    data = train)
F =  5.055601  on  1  and  481  df: p= 0.024998
> regTermTest(fit, "A7")
Wald test for A7
 in glm(formula = Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15, family = binomial,
    data = train)
F =  2.018224  on  6  and  481  df: p= 0.061769
> regTermTest(fit, "A8")
Wald test for A8
 in glm(formula = Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15, family = binomial,
    data = train)
F =  1.913832  on  1  and  481  df: p= 0.16718
> regTermTest(fit, "A9")
Wald test for A9
 in glm(formula = Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15, family = binomial,
    data = train)
F =  88.13689  on  1  and  481  df: p= < 2.22e-16
> regTermTest(fit, "A10")
Wald test for A10
 in glm(formula = Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15, family = binomial,
    data = train)
F =  5.126949  on  1  and  481  df: p= 0.024001
> regTermTest(fit, "A11")
Wald test for A11
 in glm(formula = Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15, family = binomial,
    data = train)
F =  2.309961  on  1  and  481  df: p= 0.1292
> regTermTest(fit, "A15")
Wald test for A15
 in glm(formula = Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15, family = binomial,
    data = train)
F =  12.30459  on  1  and  481  df: p= 0.00049425
```

The Wald test for variables A8 and A11 gave p-values greater than 0.10, but not greater than 0.20 so we don’t necessarily have to remove them from the model.

Cross Validation of Predicted Values:

Now, we can use the test data set to see how well the fitted model does in predicting whether a sample’s credit card application will be rejected or accepted. We will classify the observation’s credit card application in the test data as “accepted” if the fitted logisitc regression equation returns anything greater than or equal to 0.5 and classify as “rejected” if less than 0.5. Using this classification rule, we obtain the confusion matrix below, its overall accuracy, and sensitivity and specificity.



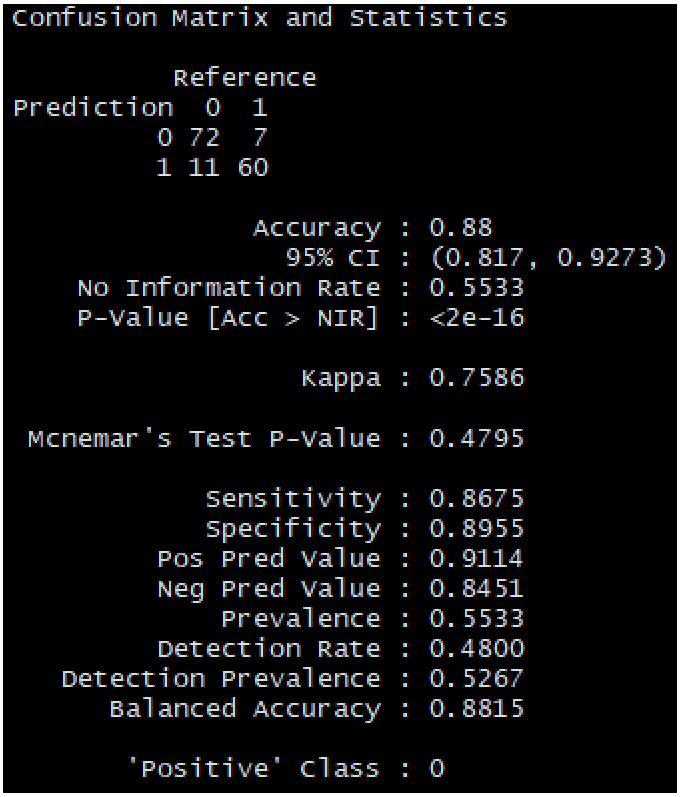
```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 72  9
         1 11 58

               Accuracy : 0.8667
                 95% CI : (0.8016, 0.9166)
    No Information Rate : 0.5533
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.731

 Mcnemar's Test P-Value : 0.8231

            Sensitivity : 0.8675
            Specificity : 0.8657
         Pos Pred Value : 0.8889
         Neg Pred Value : 0.8406
             Prevalence : 0.5533
         Detection Rate : 0.4800
   Detection Prevalence : 0.5400
      Balanced Accuracy : 0.8666

       'Positive' Class : 0
```

Here, sensitivity (true positive rate) is the proportion of actual application acceptances that were correctly identified as such by our fitted model, while specificity (true negative rate) is the proportion of actual application rejections that were correctly identified.

Our overall accuracy is 86.67%, sensitivity is 86.75%, and specificity is 86.57%.

10-fold cross validation

We can employ the K-fold cross validation method as a further cross validation measurement of our logistic regression fit. The procedure for 10-fold CV is roughly as follows: we randomize the order of our dataset, then partition it into 10 unique equally sized groups or “folds”. Then we take the 1st fold out and use it for cross validation (testing) while the other 9 folds are used to train the logistic regression model and used to predict credit card application acceptance in our “testing fold”, obtaining evaluation metrics such as accuracy, specificity, and sensitivity. We then repeat this process 10 times, taking the 2nd fold out for cross validation with the 1st fold and the other 8 being used to train the model, and so on. These 10 results can then be averaged to produce a single estimation.

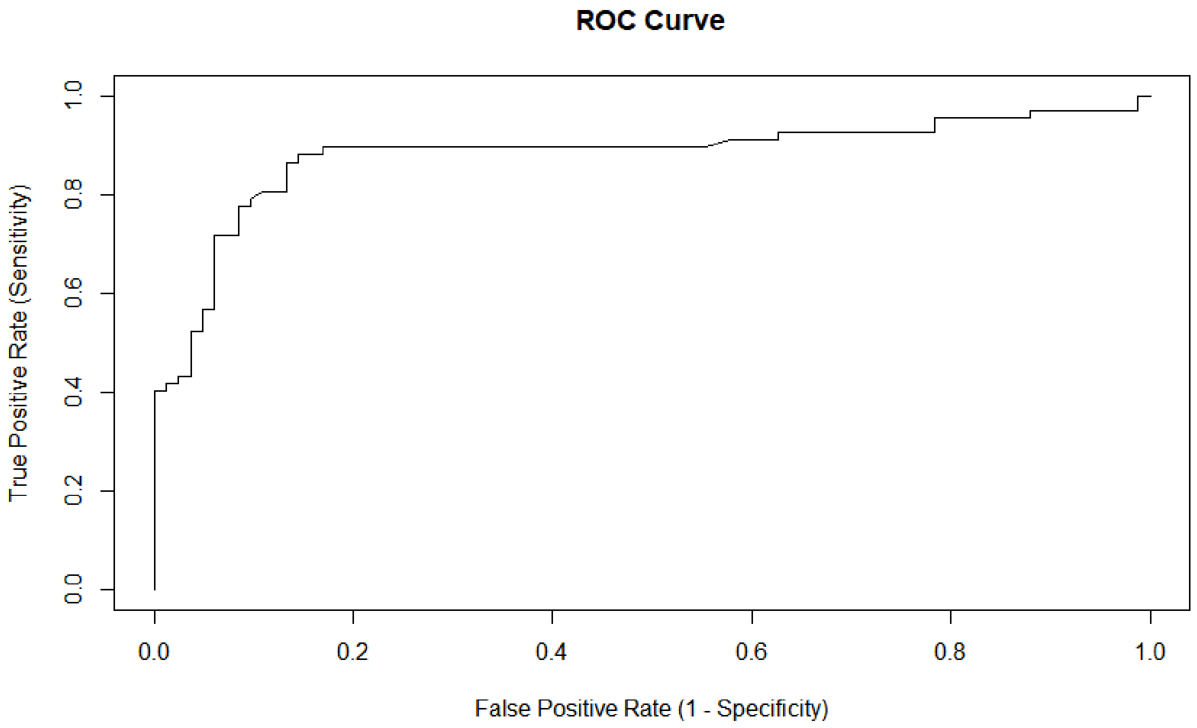


```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 72  7
         1 11 60

               Accuracy : 0.88
                 95% CI : (0.817, 0.9273)
    No Information Rate : 0.5533
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7586

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 0.8675
            Specificity : 0.8955
         Pos Pred Value : 0.9114
         Neg Pred Value : 0.8451
             Prevalence : 0.5533
         Detection Rate : 0.4800
   Detection Prevalence : 0.5267
      Balanced Accuracy : 0.8815

       'Positive' Class : 0
```

Using the 10-fold CV resulted in a slightly higher overall accuracy and specificity.

ROC Curve:

The Receiver Operator Characteristic curve plots (1 – Specificity) on the x-axis against Sensitivity on the y-axis and shows the trade-off between the rate of correct predictions with the rate of incorrect predictions. We are interested in the area under the ROC curve, which ranges from 0.5 to 1 with higher values indicating that the fitted model is great at distinguishing between the 2 classes of our response variable.



The area under the curve (AUC) is 0.8806869, indicating that the model does a good job at discriminating between credit card applications that were accepted and those that were rejected.

Appendix (R Code):

```
library(ggplot2)
library(Rmisc)
library(lmtest)
library(caret)
library(pscl)
library(survey)
library(ROCR)

# Read in and clean the data
setwd("C:/Users/Tony/Google Drive/GSU Graduate School/STAT 8561 Linear Statistical Analysis I/Project")
credit <- read.table("crx.data", sep=",", na.strings = "?")
names(credit) <- c("A1","A2","A3","A4","A5","A6","A7","A8","A9","A10","A11","A12","A13","A14","A15","Y")
not_missing <- (apply(is.na(credit), 1, sum) == 0)
credit <- credit[not_missing,]
str(credit)

# Data Exploration

# Response variable:
ggplot(credit, aes(x = Y)) + geom_bar(stat = "count", color="black", fill="gray")

#Recode response variable into a binary class
credit$Y <- as.character(credit$Y)
credit$Y[credit$Y=="-"] <- 0
credit$Y[credit$Y=="+"] <- 1
credit$Y <- as.factor(credit$Y)
str(credit)
```

```
nrow(credit[credit$Y==0,]) #357 applications rejected
nrow(credit[credit$Y==1,]) #296 applications accepted

# Categorical variables
xtabs(~Y + A1,credit)
xtabs(~Y + A4,credit)
xtabs(~Y + A5,credit)
xtabs(~Y + A6,credit)
xtabs(~Y + A7,credit)
xtabs(~Y + A9,credit)
xtabs(~Y + A10,credit)
xtabs(~Y + A12,credit)
xtabs(~Y + A13,credit)

p1 <- ggplot(credit, aes(x = A1)) + geom_bar(colour="black", fill="gray")
p4 <- ggplot(credit, aes(x = A4)) + geom_bar(colour="black", fill="gray")
p5 <- ggplot(credit, aes(x = A5)) + geom_bar(colour="black", fill="gray")
p6 <- ggplot(credit, aes(x = A6)) + geom_bar(colour="black", fill="gray")
p7 <- ggplot(credit, aes(x = A7)) + geom_bar(colour="black", fill="gray")
p9 <- ggplot(credit, aes(x = A9)) + geom_bar(colour="black", fill="gray")
p10 <- ggplot(credit, aes(x = A10)) + geom_bar(colour="black", fill="gray")
p12 <- ggplot(credit, aes(x = A12)) + geom_bar(colour="black", fill="gray")
p13 <- ggplot(credit, aes(x = A13)) + geom_bar(colour="black", fill="gray")

multiplot(p1, p4, p5, p6, p7, p9, p10, p12, p13, layout = matrix(c(1,2,3,4,5,6,7,8,9), nrow = 3, byrow=T))

# Continuous variables
p2 <- ggplot(credit, aes(x = A2)) + geom_histogram(binwidth = 1, colour="black", fill="gray")
p3 <- ggplot(credit, aes(x = A3)) + geom_histogram(binwidth = 1, colour="black", fill="gray")
```

```
p8 <- ggplot(credit, aes(x = A8)) + geom_histogram(binwidth = 1, colour="black", fill="gray")
p11 <- ggplot(credit, aes(x = A11)) + geom_histogram(binwidth = 5, colour="black", fill="gray")
p14 <- ggplot(credit, aes(x = A14)) + geom_histogram(binwidth = 100, colour="black", fill="gray")
p15 <- ggplot(credit, aes(x = A15)) + geom_histogram(binwidth = 5000, colour="black", fill="gray")

multiplot(p2, p3, p8, layout = matrix(c(1,2,3), nrow = 1, byrow=T))
multiplot(p11, p14, p15, layout = matrix(c(1,2,3), nrow = 1, byrow=T))

# Remove observations in the categorical variables with low number of values in some levels
credit <- credit[c(-which(credit$A4=="l")),] # These 2 obs that were dropped also had level of 1 for "gg" for A5
credit$A4 <- droplevels(credit$A4)
credit <- credit[c(-which(credit$A6=="r")),] # 3 obs dropped
credit$A6 <- droplevels(credit$A6)
credit <- credit[c(-which(credit$A7=="n"|credit$A7=="o")),] # 3 obs dropped
credit$A7 <- droplevels(credit$A7)
credit <- credit[c(-which(credit$A13=="p")),] # 1 obs dropped, total of 9 obs dropped, n = 644
credit$A13 <- droplevels(credit$A13)
str(credit)

#Separate into train and test data
set.seed(8561)
train_ind <- sample(nrow(credit), 644-150)

train <- credit[train_ind,]
test <- credit[-train_ind,]
nrow(train[train$Y==0,]) #270 applications rejected in training data
nrow(train[train$Y==1,]) #224 applications accepted in training data
nrow(test[test$Y==0,]) #83 applications rejected in testing data
nrow(test[test$Y==1,]) #67 applications accepted in testing data

# Model selection:
# Stepwise Model Selection
fit.null <- glm(Y ~ 1, data = train, family = binomial)
fit.full <- glm(Y ~ ., data = train, family = binomial)
select <- step(fit.null, scope = list(lower = fit.null, upper = fit.full), direction = "both")

fit <- glm(Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15, data = train, family = binomial)
summary(fit)
exp(coef(fit)) # Odds ratios
exp(cbind("Odds ratio"=coef(fit), confint(fit)))

# Goodness of fit
# LRT:
lrtest(fit.full, fit)
# McFadden's pseudo R^2:
pR2(fit.new)
# Wald Test:
regTermTest(fit, "A4")
regTermTest(fit, "A7")
regTermTest(fit, "A8")
regTermTest(fit, "A9")
regTermTest(fit, "A10")
regTermTest(fit, "A11")
regTermTest(fit, "A15")

# Cross Validation
```

```
pred  <-  predict(fit, newdata=test, type="response")
Y.pred <- NULL
for (i in 1:length(pred)){ # Classify as accepted if >=0.5, and rejected if <0.5
  if(pred[i] >= 0.5){
    Y.pred[i] <- 1}
  else{
    Y.pred[i] <- 0}
}
Y.pred <- as.factor(Y.pred)
confusionMatrix(data=Y.pred, test$Y)

# ROC curve and AUC
pred_ROC <- prediction(pred, test$Y)
perf <- performance(pred_ROC, measure = "tpr", x.measure = "fpr")
plot(perf, main = "ROC Curve", xlab = "False Positive Rate (1 - Specificity)", ylab = "True Positive Rate (Sensitivity)")


auc <- performance(pred_ROC, measure = "auc")
auc <- auc@y.values[[1]]
auc #0.8806869

# 10-fold CV
ctrl <- trainControl(method = "repeatedcv", number = 10, savePredictions = TRUE)
test_fit <- train(Y ~ A4 + A7 + A8 + A9 + A10 + A11 + A15,  data=credit, method="glm", family="binomial",
                  trControl = ctrl, tuneLength = 5)
pred <-  predict(test_fit, newdata=test)
confusionMatrix(data=pred, test$Y)
```