Default of Credit Card Clients

Onkar Deshpande, Magdalene Fogarasi
Jingwei Li, Huixuan Zhu, Yuyan Xu
Ming Zhang, Han Zhou, Tingting Xuan

Background

- Data represents demographics, credit card spending, and default rate of customers in Taiwan
- Bill amount and payment amount every month: April - September of 2005
- Credit limit and default rate at the end of the study
- Demographics collected include:
    - Age, education, gender, and marital status

Data collected from: UCI Machine Learning Repository

Impact and Importance

- Credit card companies are interested in how to best choose credible customers
- Interesting to companies to see when spending is higher
  - Promotions
- Default rate prediction/probability
  - How to screen individuals who will not pay back their bills

Hypotheses Outline

1. Equal Means of Bill Amount
    a. Customers spending is the same (independent from the month)
    b. Spending Trend of customers with respect to month
2. Default Rate Prediction
    a. Model using demographics information to predict individuals who are more likely to default on payments
3. Credit Limit Determination
    a. Predicting the credit limit by multi-linear model

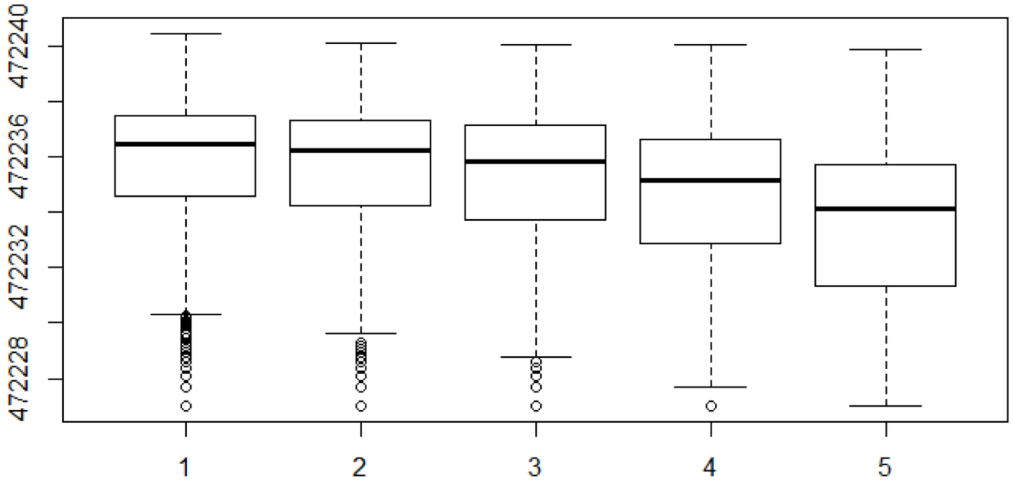
Hypothesis 1: Equal Means?

Hypothesis: The mean bill amounts for each month is the same throughout the payment period

H_0: $\mu_{May} = \mu_{June} = ... = \mu_{September}$;
H_a:At least one ≠

- Multiple equality demands for ANOVA method. Assumption for which is that the columns have equal variance.
- To check equal variance, F-test was conducted between all pair of columns
- Result of F-test was also checked visually
- Hence, used Bonferroni test at significance level 0.1 to check equal means over consecutive months to identify trend in the data.

Variance test

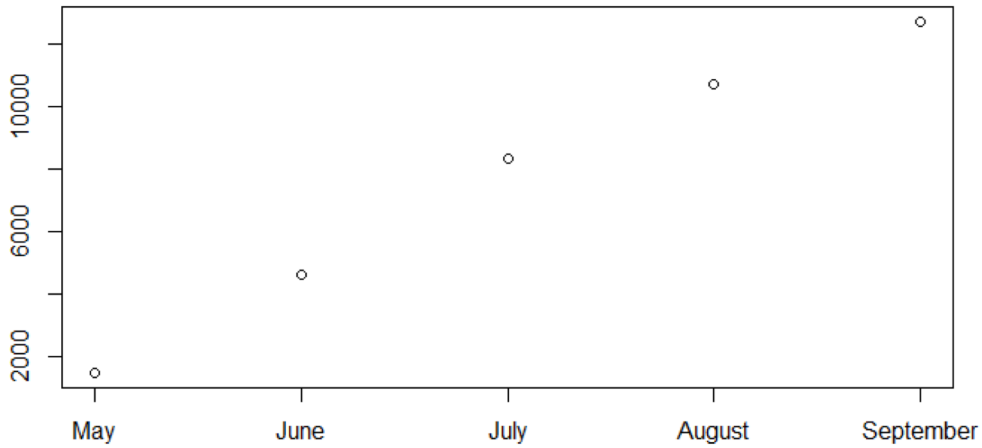


Variance in Bill Amount per month is not equal

Comparison	P Value
May- June	2.2e-16
June-July	2.2e-16
July-August	2.2e-16
August- September	2.2e-16
May-July	2.2e-16
May-August	2.2e-16
May-September	2.2e-16
June-August	2.2e-16
June-September	2.2e-16
July-September	2.2e-16

Bonferroni Test @ significance 0.10

Comparison	P-Value
May-June	< 2.2e-16
June-July	<2.2e-16
July-August	2.543e-12
August- September	2.085e-09



Overall Trend:

The mean amount spent is increasing from May to September when benchmarked against the initial April Bill Amount

Hypothesis 1: Conclusion

- Bill amounts are not the same throughout months
  - Consumer spending in different months varies
  - Further work: Factors that affect spending and validate our findings

Hypothesis 2: Understanding variables contributing to default rate

Data Preprocessing:

1. Choosing the history of past payment, amount of bill statement and amount of previous payment as our variable candidates, the periods are six month.
2. Using the summation of the history of past payment in 6 months as one variable. Since there are some negative values, the same positive value is added to the summation in order to adjust the minimum values to positive one. Thus we get a new variable.
3. Setting the negative values in the bill statements and previous payments as "1" (relatively few negative numbers). After log transformation, they will be 0.
4. Log transformation for all the chosen variables.

Introduction of Sampling method

1. The dataset contains defaulters vs non-defaulters in 1:4 ratio, hence we used balanced sampling method to fit a good model.
2. We divided the samples with payment default into two parts: one is training samples (we will call them “training 1 samples” in the following) and another as testing samples (we will call them “testing 1 samples in the following).
3. If we want to train n samples totally, we will first choose n/2 samples from the samples with non-defaulters. Then, we will choose the left n/2 samples from the training 1 samples.
4. If n/2 is large than the total number of the training 1 samples, we will first choose the whole training 1 samples and next resampling from the training 1 samples for the left.
5. This method can guarantee enough and balanced sample numbers for both groups containing defaulters and non-defaulters.

Logistic regression

Here,we want to predict the value of the payment default. It is a classification problem. However, the payment default has a discrete value--0 or 1. The linear regression algorithm will produce poor results. To fix this problem, we will choose the hypothesis as follows:

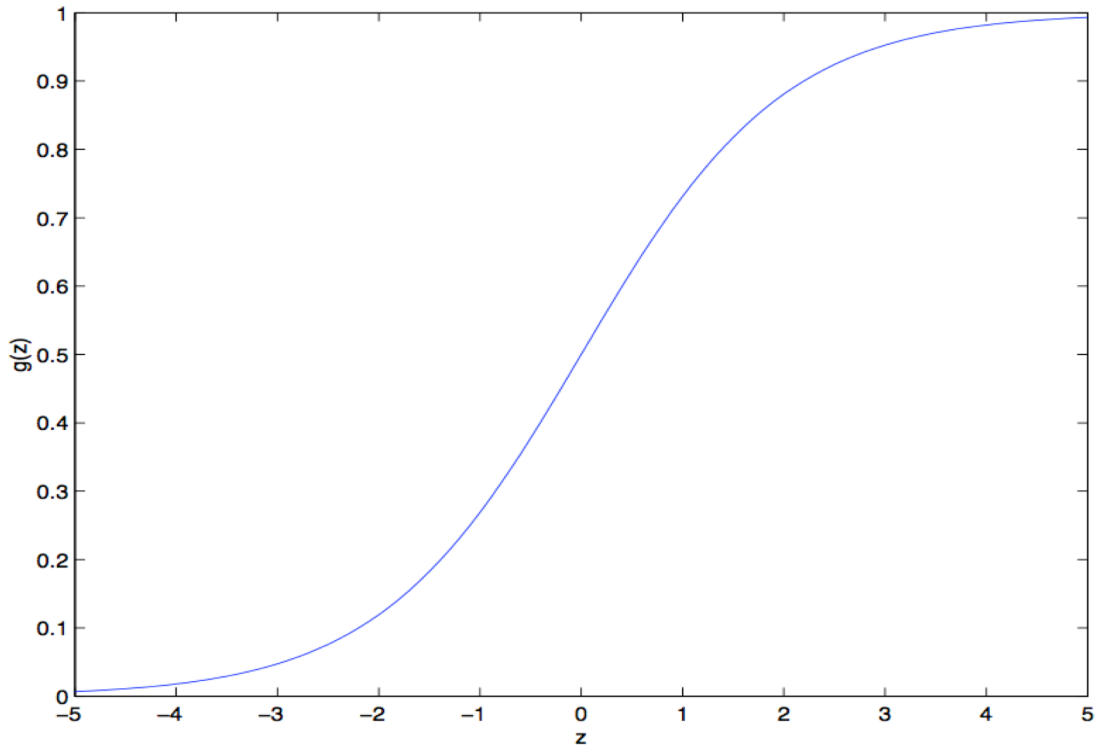
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}},$$

where

$$g(z) = \frac{1}{1 + e^{-z}}$$

is called the **logistic function** or the **sigmoid function**. Here is a plot showing $g(z)$:

@cited from Stanford Open Classes, CS229 Lecture notes, Andrew Ng



@cited from Stanford Open Classes, CS229 Lecture notes, Andrew Ng

Model Analysis

```
Call:
glm(formula = Y ~ X12 + X13 + X14 + X15 + X16 + X17 + X18 + X19 +
    X20 + X21 + X22 + X23 + X24, family = binomial(link = "logit"),
    data = training_data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.84393  -1.04797  -0.02375   1.06849   2.45139

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.07538    0.05560  -1.356  0.17519
X12         -0.16328    0.01536 -10.627  < 2e-16 ***
X13          0.14485    0.02051   7.063 1.63e-12 ***
X14          0.17765    0.02086   8.517  < 2e-16 ***
X15          0.06722    0.02065   3.255  0.00113 **
X16         -0.01453    0.01999  -0.727  0.46712
X17          0.14195    0.01816   7.816 5.46e-15 ***
X18         -0.26659    0.01342 -19.858  < 2e-16 ***
X19         -0.24030    0.01387 -17.326  < 2e-16 ***
X20         -0.19337    0.01367 -14.145  < 2e-16 ***
X21         -0.11383    0.01360  -8.370  < 2e-16 ***
X22         -0.13033    0.01409  -9.252  < 2e-16 ***
X23         -0.06243    0.01078  -5.792 6.96e-09 ***
X24          1.40346    0.07205  19.480  < 2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

⟹ **Insignificant !!!**

```
Call:
glm(formula = Y ~ X12 + X13 + X14 + X15 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24, family = binomial(link = "logit"),
    data = training_data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.87397  -1.04092  -0.02913   1.07116   2.45882

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.03877    0.05460   0.710    0.478
X12         -0.15891    0.01541 -10.315  < 2e-16 ***
X13          0.15712    0.02054   7.650 2.01e-14 ***
X14          0.16513    0.02081   7.936 2.08e-15 ***
X15          0.08959    0.01979   4.527 5.99e-06 ***
X17          0.15395    0.01696   9.078  < 2e-16 ***
X18         -0.29738    0.01360 -21.874  < 2e-16 ***
X19         -0.25372    0.01384 -18.332  < 2e-16 ***
X20         -0.18684    0.01363 -13.706  < 2e-16 ***
X21         -0.13280    0.01149 -11.554  < 2e-16 ***
X22         -0.12823    0.01401  -9.156  < 2e-16 ***
X23         -0.07353    0.01084  -6.785 1.16e-11 ***
X24          1.27428    0.06966  18.293  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

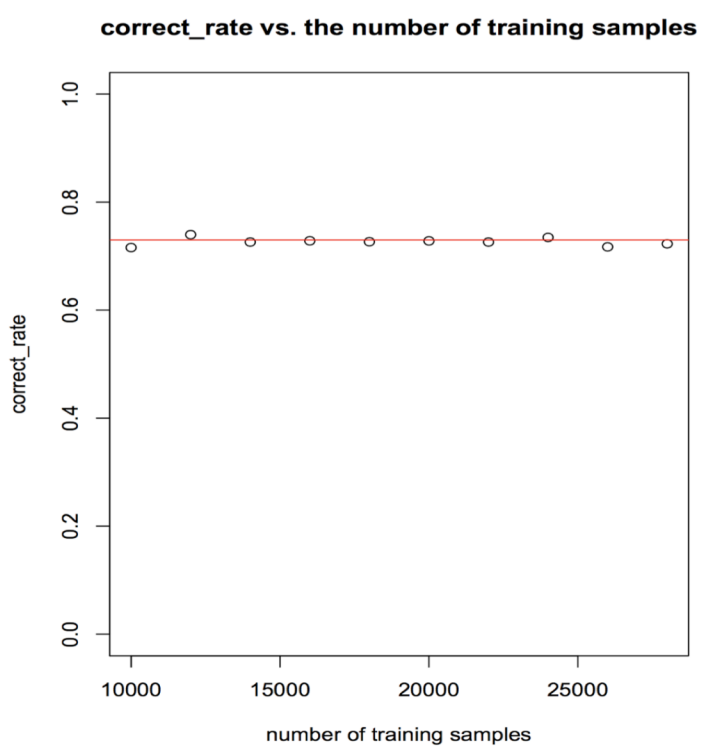
The results of the classification

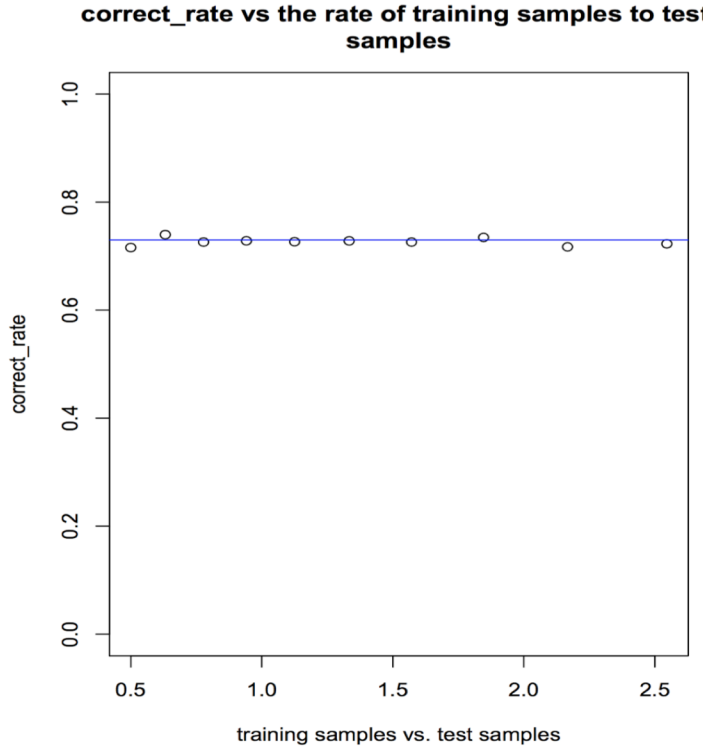
The classification results

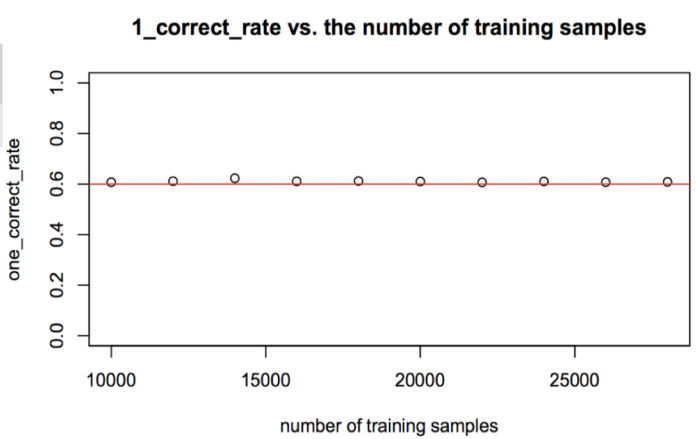
	Test value / Predict value	0	1
Number of training samples = 10000	0	13274	591
	1	5090	1045
	accuracy	0.723	0.639
Number of training samples = 12000	Test value / Predict value	0	1
	0	13052	630
	1	4312	1006
	accuracy	0.752	0.615
Number of training samples = 14000	Test value / Predict value	0	1
	0	12033	597
	1	4331	1039
	accuracy	0.735	0.635
Number of training samples = 16000	Test value / Predict value	0	1
	0	11366	618
	1	3998	1018
	accuracy	0.740	0.622
Number of training samples = 18000	Test value / Predict value	0	1
	0	10591	598
	1	3773	1038
	accuracy	0.737	0.634

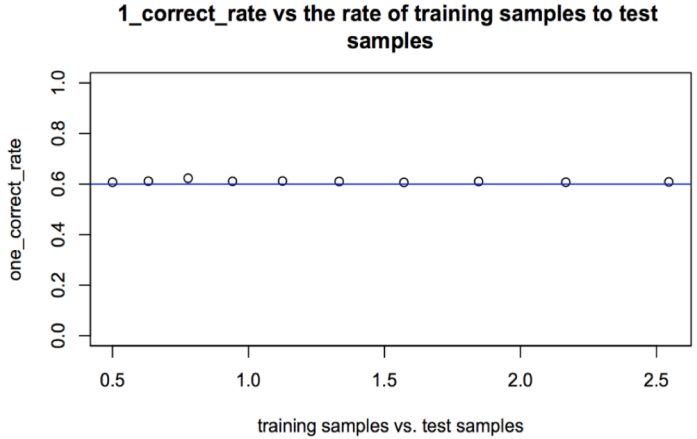
	Test value / Predict value	0	1
Number of training samples = 20000	0	9919	630
	1	3445	1006
	accuracy	0.742	0.615
Number of training samples = 22000	Test value / Predict value	0	1
	0	9133	602
	1	3231	1034
	accuracy	0.739	0.632
Number of training samples = 24000	Test value / Predict value	0	1
	0	8552	636
	1	2812	1000
	accuracy	0.753	0.611
Number of training samples = 26000	Test value / Predict value	0	1
	0	7569	598
	1	2795	1038
	accuracy	0.730	0.634
Number of training samples = 28000	Test value / Predict value	0	1
	0	6939	623
	1	2425	1013
	accuracy	0.741	0.620

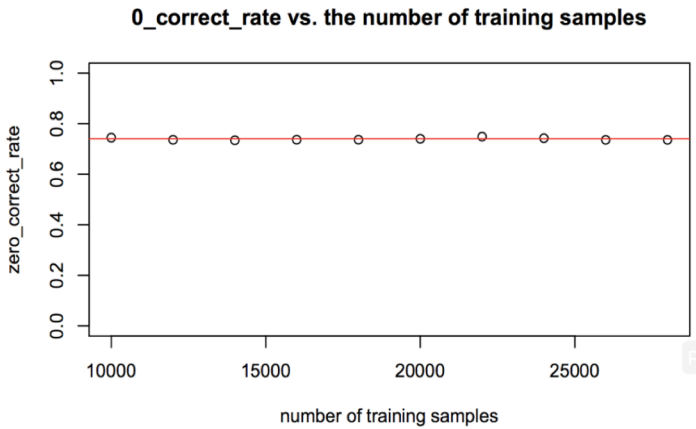
The results of the classification

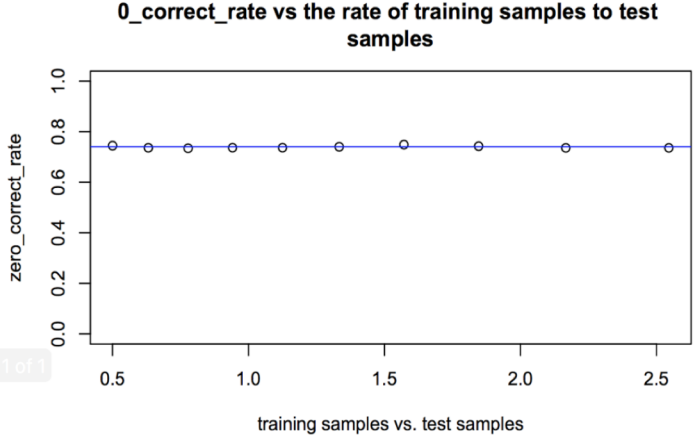





1_correct_rate vs. the number of training samples
one_correct_rate
0.0
0.2
0.4
0.6
0.8
1.0
10000
15000
20000
25000
number of training samples


1_correct_rate vs the rate of training samples to test samples
one_correct_rate
0.0
0.2
0.4
0.6
0.8
1.0
0.5
1.0
1.5
2.0
2.5
training samples vs. test samples


0_correct_rate vs. the number of training samples
zero_correct_rate
0.0
0.2
0.4
0.6
0.8
1.0
10000
15000
20000
25000
number of training samples


0_correct_rate vs the rate of training samples to test samples
zero_correct_rate
0.0
0.2
0.4
0.6
0.8
1.0
0.5
1.0
1.5
2.0
2.5
training samples vs. test samples

Hypothesis 3： Multiple Linear Regression

Data Preprocessing:

- Remove some meaningless rows, such as “Education =5 or 6” or “Marital Status = 0 or 3.”
- Create “X24” as a new variable that denotes the sum of history of payment, X6 to X11, which can show how people like to pay money back to bank, and make it to a positive number.
- Also remove negative values in bill row, x12-x17, because negative statement makes no sense here.
- Last but not least, we normalized columns for bills and payments.

Hypothesis 3: Outline For our Model

Hypothesis Test:

- Good Model or Not? Are all the coefficients significant in our model?

Outline:

- Examine correlation between each row to make sure there is no overfitting problem in our model. Then, we choose PCA method to create a variable which can represent the rows that have high correlation.
- Fit a multi-linear model, use box-cox transformation and eliminate outliers and observations which surpass cook's distance.
- Show the result of our model and provide a way for us to increase our credit limit efficiently and effectively

Correlation and PCA procedure

High Correlation between bill statement

```
> cor(ams_572_data[13:18]) #bill
          X12       X13       X14       X15       X16       X17
X12 1.0000000 0.9515816 0.8918916 0.8617716 0.8319828 0.8058471
X13 0.9515816 1.0000000 0.9277088 0.8939405 0.8620776 0.8350851
X14 0.8918916 0.9277088 1.0000000 0.9256453 0.8863133 0.8568140
X15 0.8617716 0.8939405 0.9256453 1.0000000 0.9404560 0.9029626
X16 0.8319828 0.8620776 0.8863133 0.9404560 1.0000000 0.9477847
X17 0.8058471 0.8350851 0.8568140 0.9029626 0.9477847 1.0000000
```

Do PCA to these columns and get a Score using $score

```
Importance of components:
                          Comp.1     Comp.2     Comp.3     Comp.4      Comp.5      Comp.6
Standard deviation     2.332708 0.54779684 0.33338364 0.25643810 0.203767049 0.199915721
Proportion of Variance 0.906921 0.05001356 0.01852411 0.01096008 0.006920168 0.006661049
Cumulative Proportion  0.906921 0.95693459 0.97545870 0.98641878 0.993338951 1.000000000
```

```
> pc$loadings

Loadings:
    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
X12 -0.401  0.542  0.453 -0.215 -0.524 -0.141
X13 -0.410  0.430  0.115         0.758  0.223
X14 -0.412  0.167 -0.633  0.548 -0.223 -0.227
X15 -0.415 -0.180 -0.432 -0.608 -0.120  0.474
X16 -0.410 -0.430        -0.263  0.249 -0.712
X17 -0.401 -0.526  0.431  0.453 -0.154  0.384
```

Fit a Multiple Linear Model

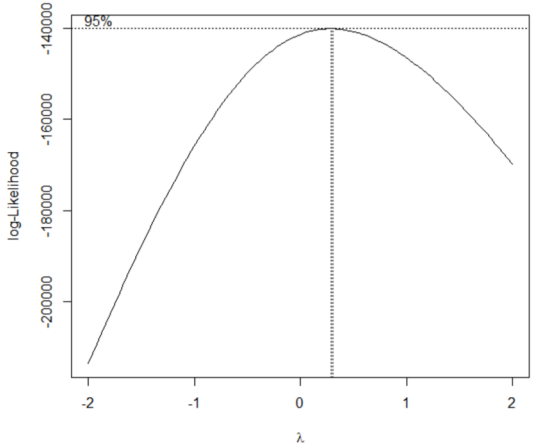
$$\text{Credit}^1 = \beta_0 + \beta_1 \text{Sex} + \beta_2 \text{Marital} + \beta_3 \text{Edu} + \beta_4 Age + \beta_5 \sum_{from\ x6\ to\ x11} history\ of\ payment$$

$$+ \beta_6 Score + \sum_{j=7}^{12} \beta_j \sum_{i=18}^{23} X_i$$

Since the result is not very satisfied, we do box-cox transformation to make it better.

$$y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y_i) & \text{if } \lambda = 0, \end{cases}$$

Our result: Lambda =0.3



Fit a New Multiple Linear Model

$$\text{Credit}^{0.3} = \beta_0 + \beta_1 \text{Sex} + \beta_2 \text{Marital} + \beta_3 \text{Edu} + \beta_4 Age + \beta_5 \sum_{from\ x6\ to\ x11} history\ of\ payment$$

$$+ \beta_6 Score + \sum_{j=7}^{12} \beta_j \sum_{i=18}^{23} X_i$$

However, the model would be better if we eliminate outliers to remove points with a strong influence on data - which is also known as cook's distance. In R, we use OutlierTest function to test whether there are outliers, we will eliminate these outliers if p value of this test is less than 0.05, which means we can say this point is outlier. Also, after eliminating outliers, we will introduce the concept of cook's distance to help us improve our model even better. Generally, if Di is larger than 0.5, we say that this point should be eliminated. Cook's distance Di is:

$$D = \frac{\left\| \hat{Y} - \hat{Y}_{(-i)} \right\|^2}{\rho \hat{\sigma}^2} = \frac{\left(\hat{\beta} - \hat{\beta}_{(-i)} \right)' X' X (\beta - \beta_{(-i)})}{\rho \hat{\sigma}^2}$$

Results

After considering outliers and influential points by introducing cook's distance, we have eliminated 380 rows or, in other word, observations. And our model has been actually improved after box-cox transformation, eliminating outliers and influential points:

```
Residuals:
    Min      1Q  Median      3Q     Max
-741028  -71457  -23333   50431  606065

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 243892.52    3771.86  64.661  < 2e-16 ***
X22           5527.40    1255.67   4.402 1.08e-05 ***
X32         -49162.17    1377.78 -35.682  < 2e-16 ***
X33         -73531.17    1889.64 -38.913  < 2e-16 ***
X42         -16714.19    1394.94 -11.982  < 2e-16 ***
X5            1555.18      77.35  20.105  < 2e-16 ***
X24          -7527.84     112.09 -67.161  < 2e-16 ***
score       -17597.25     299.02 -58.849  < 2e-16 ***
X18           5673.75     648.13   8.754  < 2e-16 ***
X19           5120.34     641.95   7.976 1.56e-15 ***
X20           8262.95     643.04  12.850  < 2e-16 ***
X21           8886.95     634.49  14.007  < 2e-16 ***
X22           9860.27     639.81  15.411  < 2e-16 ***
X23          11923.94     635.07  18.776  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 103700 on 29149 degrees of freedom
Multiple R-squared:  0.3645,    Adjusted R-squared:  0.3643
F-statistic:  1286 on 13 and 29149 DF,  p-value: < 2.2e-16
```

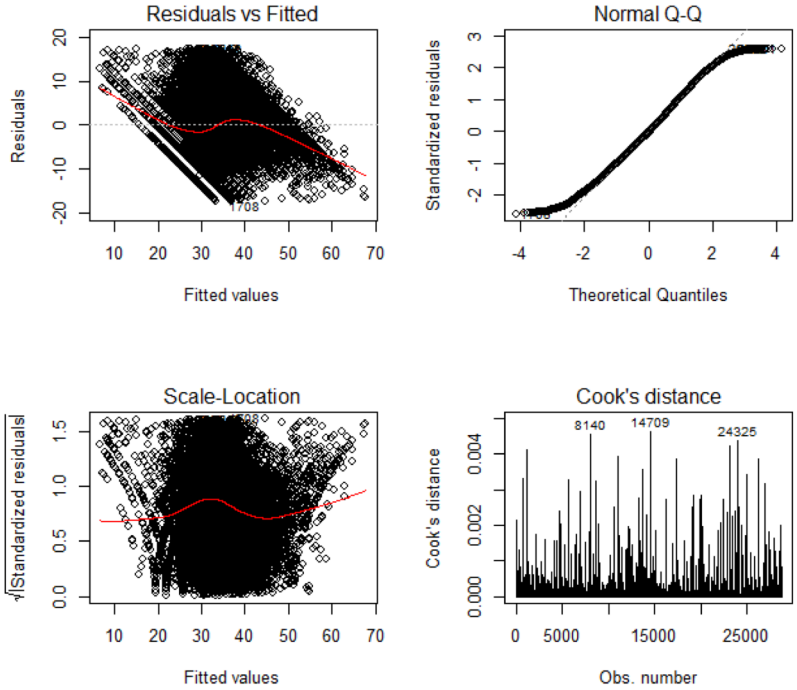
```
Residuals:
     Min       1Q   Median       3Q      Max
-17.5669  -4.7855  -0.3251   4.7102  17.4851

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 40.694001   0.248041 164.061   <2e-16 ***
X22          1.034987   0.082566  12.535   <2e-16 ***
X32         -3.628971   0.090607 -40.052   <2e-16 ***
X33         -5.587453   0.124326 -44.942   <2e-16 ***
X42         -1.377854   0.091671 -15.030   <2e-16 ***
X5           0.103695   0.005083  20.399   <2e-16 ***
X24         -0.611321   0.007450 -82.058   <2e-16 ***
score       -1.354231   0.019601 -69.091   <2e-16 ***
X18          0.406839   0.046011   8.842   <2e-16 ***
X19          0.628559   0.056813  11.064   <2e-16 ***
X20          0.536237   0.047043  11.399   <2e-16 ***
X21          0.546651   0.044066  12.405   <2e-16 ***
X22          0.655606   0.044807  14.632   <2e-16 ***
X23          0.820687   0.046125  17.793   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.772 on 28769 degrees of freedom
Multiple R-squared:  0.4265,    Adjusted R-squared:  0.4262
F-statistic:  1646 on 13 and 28769 DF,  p-value: < 2.2e-16
```

Results (cont'd)

--Residual Analysis



1. The reason why we get this graph is that credit limit is not exactly continuous in our dataset.
2. Our residuals fit a normal distribution very well.
3. Another graph to represent fitted values vs residuals.
4. After eliminating some points over our abline, it looks better.

1	2
3	4

Hypothesis 3: Conclusion

```
Residuals:
     Min       1Q   Median       3Q      Max
-17.5669  -4.7855  -0.3251   4.7102  17.4851

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 40.694001   0.248041 164.061   <2e-16 ***
X22          1.034987   0.082566  12.535   <2e-16 ***
X32         -3.628971   0.090607 -40.052   <2e-16 ***
X33         -5.587453   0.124326 -44.942   <2e-16 ***
X42         -1.377854   0.091671 -15.030   <2e-16 ***
X5           0.103695   0.005083  20.399   <2e-16 ***
X24         -0.611321   0.007450 -82.058   <2e-16 ***
score       -1.354231   0.019601 -69.091   <2e-16 ***
X18          0.406839   0.046011   8.842   <2e-16 ***
X19          0.628559   0.056813  11.064   <2e-16 ***
X20          0.536237   0.047043  11.399   <2e-16 ***
X21          0.546651   0.044066  12.405   <2e-16 ***
X22          0.655606   0.044807  14.632   <2e-16 ***
X23          0.820687   0.046125  17.793   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.772 on 28769 degrees of freedom
Multiple R-squared:  0.4265,    Adjusted R-squared:  0.4262
F-statistic:  1646 on 13 and 28769 DF,  p-value: < 2.2e-16
```

$$Explaination\ of\ Factors \begin{cases} X2 \begin{cases} X22 = 0\ Male \\ X22 = 1\ Female \end{cases} \\ X3 \begin{cases} X32 = 0, X33 = 0\ Graduate\ School \\ X32 = 1, X33 = 0\ University \\ X32 = 0, X33 = 1\ HighSchool \end{cases} \\ X4 \begin{cases} X42 = 0\ Married \\ X42 = 1\ Single \end{cases} \end{cases}$$

First of all, all variables in our model are significant @0.001.
Then, there is no better model after doing stepwise in R, which means our model is best model given such variables.
According to this model, we **Conclude** that:

$$Higher\ Credit\ Limit \begin{cases} Female \\ Graduate\ School \\ Married \\ Older \\ Pay\ back\ before\ the\ due \\ Spend\ Less \\ Pay\ More \end{cases}$$

Questions?

Appendix: Hypothesis 1 Code

```
#reading dataset
data = read.csv('dataset.csv', header = TRUE)
data_bill_amt = data[,c(13:18)]

#preprocessing subtracting april month data from rest other cols
diff.names <- colnames(data_bill_amt)[1:5]
diff <- vector('list',length(diff.names))
names(diff) <- diff.names
diff <- matrix(nrow = dim(data)[1],ncol=5)
dimnames(diff) <- list(rownames(diff),colnames(data_bill_amt)[1:5])
for (col_name in colnames(diff)){
  diff[,col_name] <- (data_bill_amt[,col_name] -data_bill_amt$BILL_AMT6)
}
## Plotting for boxplot, to check visually constant variance boxplot(c(log(diff)-
min(diff))~rep(c(1:dim(diff)[2]),each=dim(diff)[1]))
```

```
##f - test to check for equal variance
for (i in 1:4){
                    for (j in (i+1):5){
                                        print(var.test(diff[,i],diff[,j]))
                    }
}
#Equality on means of two bill amount with unequal var [t-test]
for (i in 1:4){
          print(t.test(diff[,i],diff[,i+1],var.equal = FALSE, conf.level = (1-
0.1/4)))
}
##plotting mean data
mean_amt <-
c(mean(diff[,5]),mean(diff[,4]),mean(diff[,3]),mean(diff[,2]),mean(diff[,1]))
month_name <- c('May','June', 'July','August', 'September')
plot(mean_amt, xlab = "Month", ylab = "Mean amount",  xaxt='n',
ann=FALSE)
axis(1, at= 1:5, labels=month_name)
```

Appendix Hypothesis#2 R code

```
### code of logistic regression
setwd("/Users/jingweili/Documents/572/572project")
 credit = read.csv("credit_data1_copy.csv", header = T)
 attach(credit)
ID1 = credit[Y==1,1]
credit1 = credit[ID1,]
 k1 = dim(credit1)[1]
 k2 = 5000
 credit1_training_id = sample(1:k1,k2)
 credit1_training = credit1[credit1_training_id,]
 testing_data_1 = credit1[-credit1_training_id,]
 credit0 = credit[-ID1,]
 k3 = dim(credit0)[1]
 training_number = seq(10000,28000,by=2000)
 n=30000
 l = length(training_number)
 testing_number = rep(1,times=l)
 correct_rate = rep(1,times=l)
 zero_correct = rep(1,times=l)
 zero_false = rep(1,times=l)
 zero_correct_rate = rep(1,times=l)
 one_correct = rep(1,times=l)
 one_false = rep(1,times=l)
 one_correct_rate = rep(1,times=l)
 for (i in 1:l){
 m = training_number[i]

 if((m/2) < 2*k2){                                  ###banlanced sampling method
    training_id1 = sample(1:k2,m/2-k2)
    training_data_1 = rbind(credit1_training,credit1_training[training_id1,])
    }
 else if((m/2) >= 2*k2){
    training_id1 = sample(1:k2,m/2-2*k2)
    training_data_1 =  rbind(credit1_training,credit1_training,credit1_training[training_id1,])
    }
 training_data_id0 = sample(1:k3,m/2)
 training_data_0 = credit0[training_data_id0,]
 testing_data_0 = credit0[-training_data_id0,]
 training_data = rbind(training_data_1,training_data_0)
 testing_data = rbind(testing_data_1,testing_data_0)
 Y_testing = testing_data[,15]
 testing_number[i] = length(Y_testing)
 library(ISLR)
 credit_model = glm(Y~ X12 + X13 + X14 + X15 + X17+X18+X19+X20+X21+X22+X23+X24, data = training_data, family = binomial(link = "logit"))
 credit_pred_probs = predict(credit_model, testing_data, type = "response")    ###logistic regression
 summary(credit_model)
 credit_pred_Y = rep("0",testing_number[i])
 credit_pred_Y[credit_pred_probs>0.5] = "1"
 table(credit_pred_Y, Y_testing)                #to produce the confusion matrix----the classification results

 correct_rate[i] = mean(credit_pred_Y == Y_testing)
 zero_correct[i] = table(credit_pred_Y, Y_testing)[1,1]
 zero_false[i] = table(credit_pred_Y, Y_testing)[2,1]
 zero_correct_rate[i] = zero_correct[i]/(zero_correct[i]+zero_false[i])
 one_correct[i] = table(credit_pred_Y, Y_testing)[2,2]
 one_false[i] = table(credit_pred_Y, Y_testing)[1,2]
 one_correct_rate[i] = one_correct[i]/(one_correct[i]+one_false[i])

 }
 correct_rate
 zero_correct_rate
 one_correct_rate

###plot the total corract_rate of classification
 par(mfcol=c(1,2))
 plot(training_number,correct_rate,ylim=c(0,1),xlab="number of training samples")
 abline(h=0.73,col="red")
 title(main = "correct_rate vs. the number of training samples")
 plot(training_number/(testing_number),correct_rate,ylim=c(0,1),xlab="training samples vs. test samples")
 abline(h=0.73,col="blue")
 title(main = "correct_rate vs the rate of training samples to test \nsamples")

### plot the correct_rate of o and 1
 par(mfrow=c(2,2))
 plot(training_number,one_correct_rate,ylim=c(0,1),xlab="number of training samples")
 abline(h=0.6,col="red")
 title(main = "1_correct_rate vs. the number of training samples")
 plot(training_number/(testing_number),one_correct_rate,ylim=c(0,1),xlab="training samples vs. test samples")
 abline(h=0.6,col="blue")
 title(main = "1_correct_rate vs the rate of training samples to test \nsamples")
 plot(training_number,zero_correct_rate,ylim=c(0,1),xlab="number of training samples")
 abline(h=0.74,col="red")
 title(main = "0_correct_rate vs. the number of training samples")
 plot(training_number/(testing_number),zero_correct_rate,ylim=c(0,1),xlab="training samples vs. test samples")
 abline(h=0.74,col="blue")
 title(main = "0_correct_rate vs the rate of training samples to test \nsamples")
```

Appendix Hypothesis#3 R code

```
library(car)
library(MASS)
#reading dataset
data = read.csv('default.csv', header = TRUE)
#removing irrelavent data from education and maritial status
data <- subset(data, !(X3 %in% c(0,4,5,6)))
data <- subset(data, !(X4 %in% c(0,3)))
data = data[-1,]
#make it to numeric
for(i in 6:25){
  data[,i] = as.numeric(as.character(data[,i]))
}
data$X1 = as.numeric(as.character(data$X1))
#check if they are numeric
sapply(data,class)
#sum of bill payments, converting the negatives to postive
data$sum_pay <- data$X6+data$X7+data$X8+data$X9+data$X10+data$X11
data$sum_pay <- data$sum_pay - min(data$sum_pay)+1
ams_572_data <- data
attach(ams_572_data)

#normalize
for (j in 13:24) {
  ams_572_data[,j] = (ams_572_data[,j]-mean(ams_572_data[,j]))/sd(ams_572_data[,j])
}

colnames(ams_572_data)[colnames(ams_572_data)=="sum_pay"] = "X24" #calculate the sum of X6 to X11
detach(ams_572_data)
attach(ams_572_data)
cor(ams_572_data[13:18]) #bill
cor(ams_572_data[19:23]) #payment
cor(ams_572_data[7:12])  #due date
#PCA Analysis to get the vector that can denote all columns in our data
pc = princomp(ams_572_data[,13:18],cor = T,scores = T)
summary(pc)
pc$loadings
score = pc$scores[,1]
fit1 = lm(X1~X2+X3+X4+X5+X24+score+X18+X19+X20+X21+X22+X23)
summary(fit1)
#BOXCOX transformation
bc = boxcox(fit1)
bc$x[which.max(bc$y)]
fit2 = lm(X1^0.3~X2+X3+X4+X5+X24+score+X18+X19+X20+X21+X22+X23)
summary(fit2)
#where x2-x4 are factors
#we need consider the influence of OutLier
outlier = outlierTest(fit2)
while(outlier$p<0.01){
  row = as.numeric(names(outlier$p))
  ams_572_data = ams_572_data[c(-row),]
  detach(ams_572_data)
  attach(ams_572_data)
  pc = princomp(ams_572_data[,13:18],cor = T,scores = T)
  score = pc$scores[,1]
  fit2 = lm(X1^0.3~X2+X3+X4+X5+X24+score+X18+X19+X20+X21+X22+X23)
  outlier = outlierTest(fit2)
}
result<-cooks.distance(fit2)
cook=4/(fit2$df)
plot(fit2,which=4,cook.levels=cook)
abline(h=0.004,lty=2,col=2)
sum(result>0.004)
Data_new = ams_572_data[!result>0.004,]
detach(ams_572_data)
attach(Data_new)
pc = princomp(Data_new[,13:18],cor = T,scores = T)
score = pc$scores[,1]
fit3 <- lm( X1^0.3 ~ X2+X3+X4+X5+X24+score+X18+X19+X20+X21+X22+X23+X24,data = Data_new)
summary(fit3)
par(mfrow=c(2,2))
plot(fit3,which = 1:4)
```