To: Executive Board of Trans Lease, Inc.
Subject: Customer Default Analysis
Date: May 10, 2020

To prevent the company from experiencing losses, a data analysis has been conducted in order to develop a classification model that allows prediction of which customers would meet or fail their financial obligations with the company.

EXECUTIVE SUMMARY

Major Finding(s).

- The average net worth of customers who did not default is 21 times higher than those customers who did default. (Exhibit A)
- Customers who did not default have a debt-to-income ratio (dti) is under 75%, while most default customers are above that. (Exhibit B)
- On average, customers who did not default have a FICO score that is 46 points higher than customers who did default. (Exhibit C).

Recommendation(s) for Action.

- We recommend that the company continues following its current Credit Policy, as only 0.7% of the transactions sampled had defaulted. (Exhibit D)
- We also recommend that the company make homeownership a requirement of the Credit Policy, and do not lend to non-homeowners, as they defaulted 100% of the time in our sample data. (Exhibit E)

Analytical Overview.

The initial step in this analysis is making sure the data set being analyzed is clean. Our dataset contained 7403 data points, along with 19 variables. There were no missing values in the data set used for this analysis. While there were outliers for some of the characteristic variables themselves, they all appeared to be logical in the context of the data set and were therefore not deleted. (Exhibit F)

Because the variable we are trying to predict is “default,” which is categorial with values 0 and 1, each variable is analyzed as a whole and also by each value of “default.”

An exploratory data analysis was conducted on the variables using a combination of summary statistics and several graphical visualizations such as histograms, box plots, and scatter plots. From there, conclusions were drawn to establish which characteristics are associated with default and non-default customers with the goal to build a classification model that can predict that behavior.

Documentation Page

Exhibit A. Summary Statistics of Net Worth by Default and Non-Default Customers.

```
 Descriptive statistics by group
group: 0
   vars    n    mean      sd  median trimmed     mad    min     max   range  skew kurtosis       se
X1    1 7351 3356232 1675427 3385621 3361742 2135562 400862 6246892 5846030 -0.03    -1.18 19541.24
------------------------------------------------------------------------------------------------
group: 1
   vars  n     mean      sd median  trimmed     mad   min    max  range  skew kurtosis       se
X1    1 52 160182.6 75329.7 175879 163424.7 94805.6 23750 261623 237873 -0.29    -1.27 10446.35
```

Exhibit. B. Debt-to-Income Ratio (DTI) by Default and Non-Default Customers.

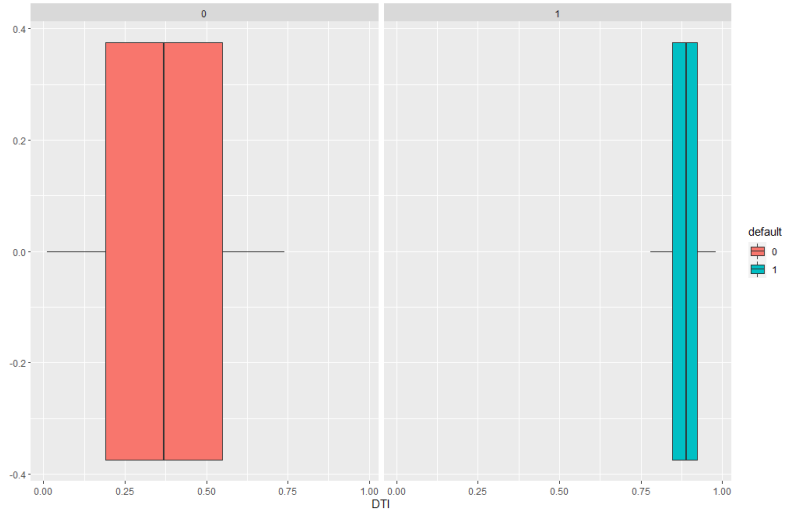


Exhibit C. FICO scores by Default and Non-Default Customers.

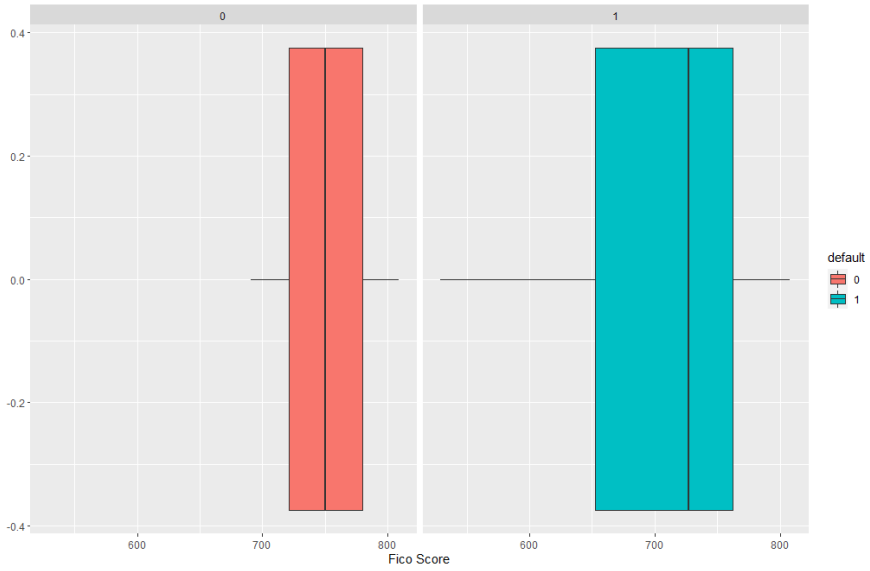


Exhibit D. Default %.

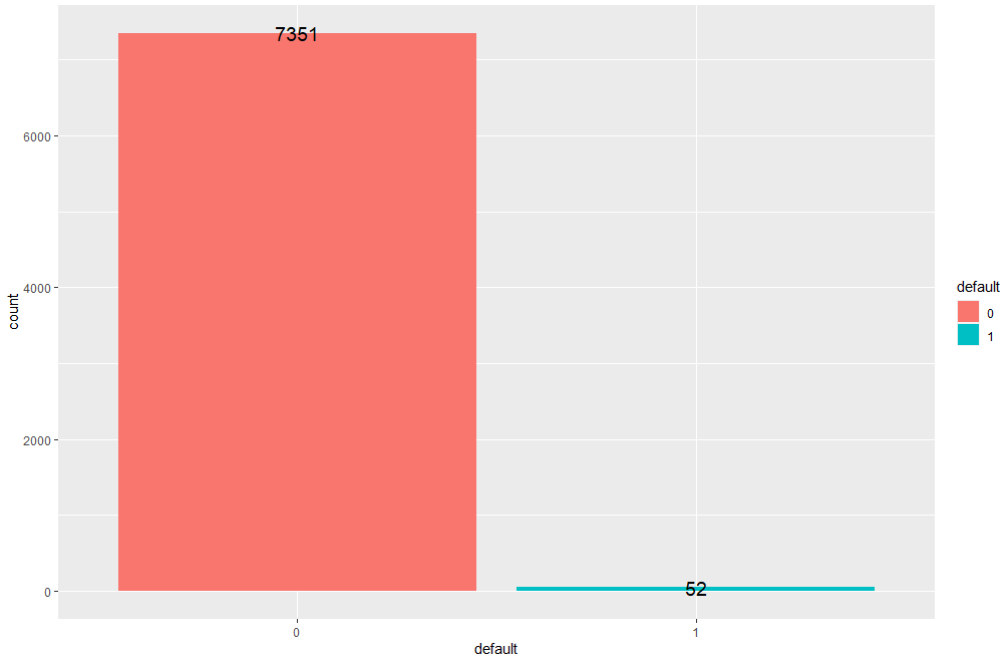


Exhibit E. Homeownership by Default and Non-Default Customers.

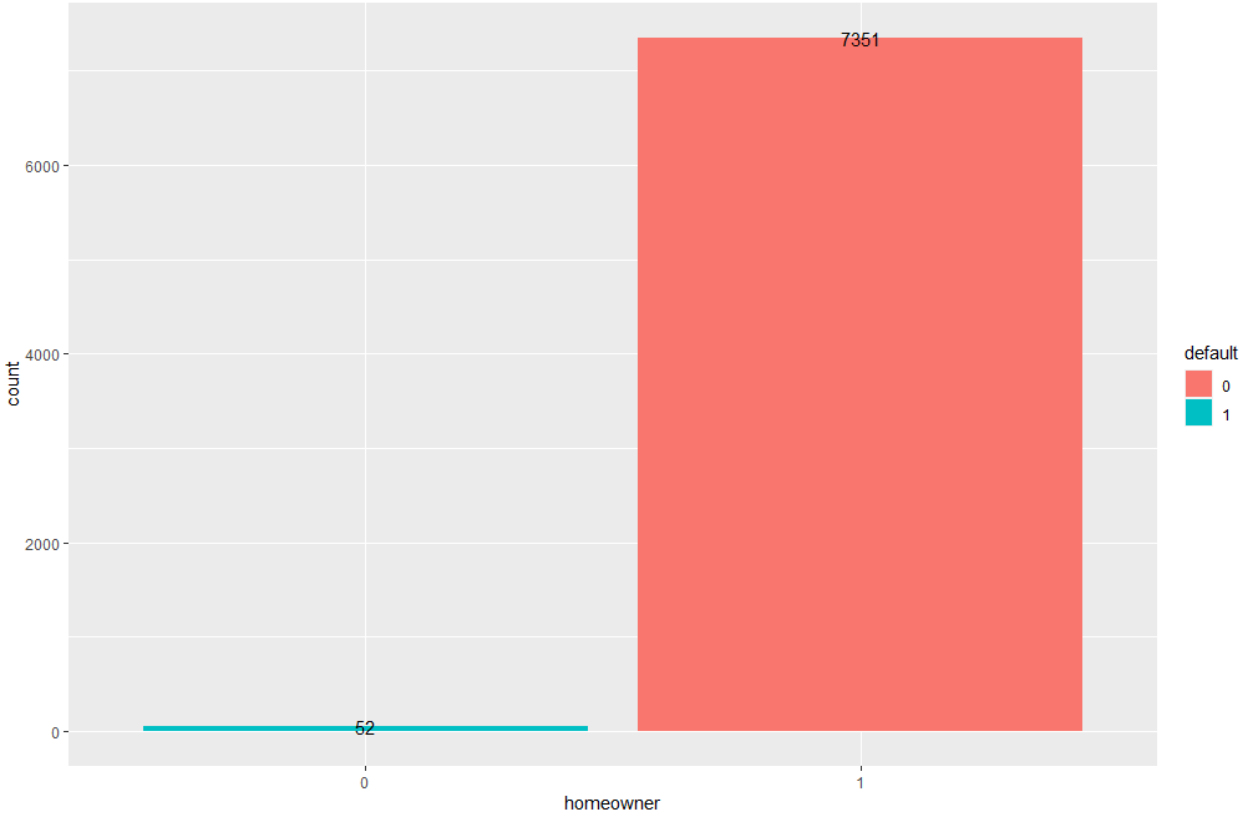
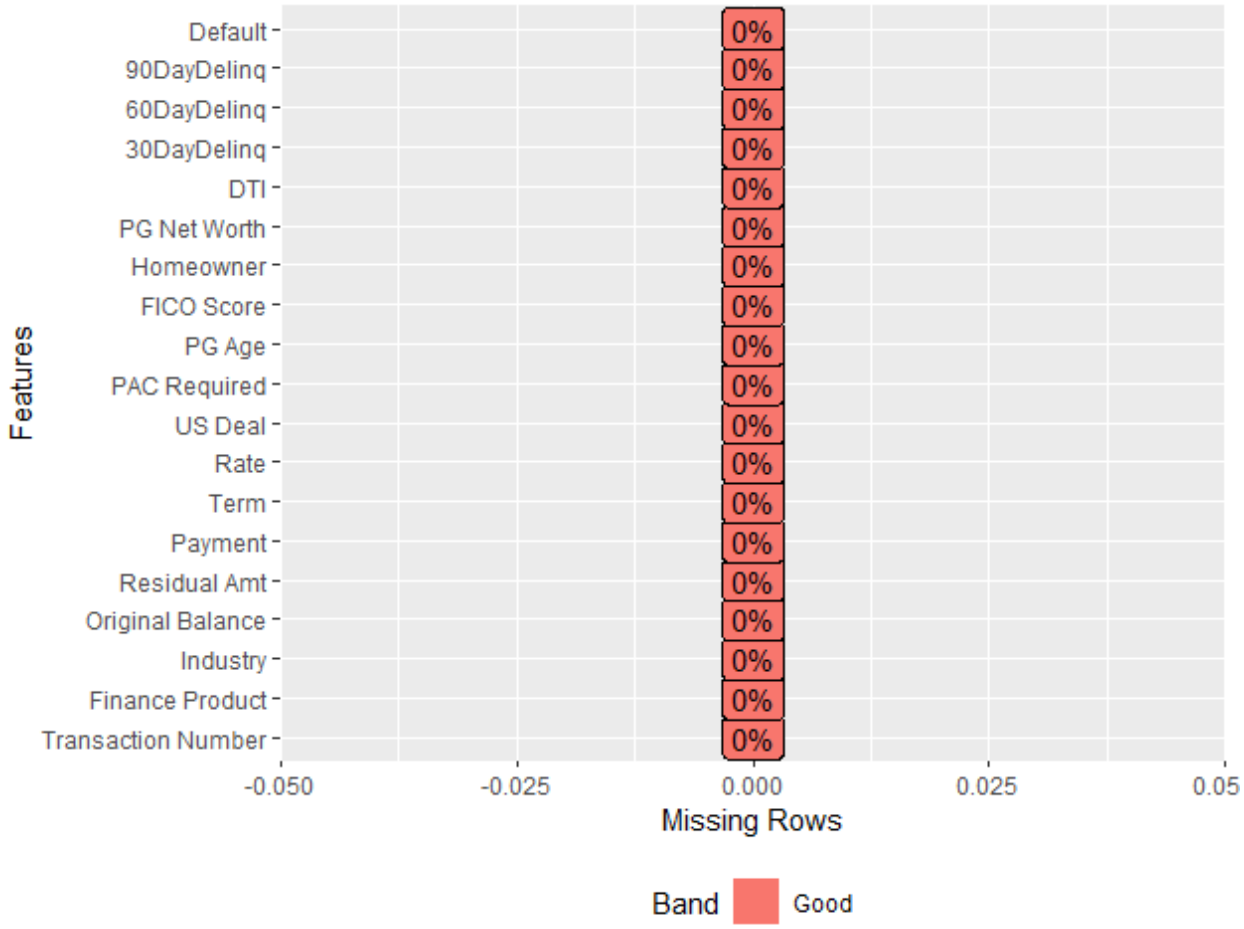


Exhibit F. Structure of Sample Data Set.

```
> str(loanData)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':        7403 obs. of  19 variables:
 $ Transaction Number: num  1 2 3 4 5 6 7 8 9 10 ...
 $ Finance Product   : chr  "TRAC" "TRAC" "TRAC" "TRAC" ...
 $ Industry          : chr  "OTR Trucking" "OTR Trucking" "OTR Trucking" "OTR Trucking" ...
 $ Original Balance  : num  9313 158234 167781 93960 120909 ...
 $ Residual Amt      : num  1863 31647 33556 18792 24182 ...
 $ Payment           : num  1035 2221 2380 2610 1679 ...
 $ Term              : num  9 61 63 38 36 38 38 38 30 46 ...
 $ Rate              : num  0.1 0.0524 0.0528 0.0667 0.0662 0.0661 0.0653 0.0653 0.0681 0.0755 ...
 $ US Deal           : num  1 1 1 1 1 1 1 1 1 1 ...
 $ PAC Required      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PG Age            : num  52 43 19 61 31 57 59 52 21 36 ...
 $ FICO Score        : num  730 729 784 794 729 704 769 720 714 757 ...
 $ Homeowner         : num  1 1 1 1 1 1 1 1 1 1 ...
 $ PG Net Worth      : num  558291 2818876 1302722 1284320 2624638 ...
 $ DTI               : num  0.58 0.28 0.45 0.08 0.25 0.14 0.1 0.4 0.5 0.42 ...
 $ 30DayDelinq       : num  0 0 0 0 1 0 0 0 0 0 ...
 $ 60DayDelinq       : num  0 0 0 0 0 0 0 0 0 0 ...
 $ 90DayDelinq       : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Default           : num  0 0 0 0 0 0 0 0 0 0 ...
```



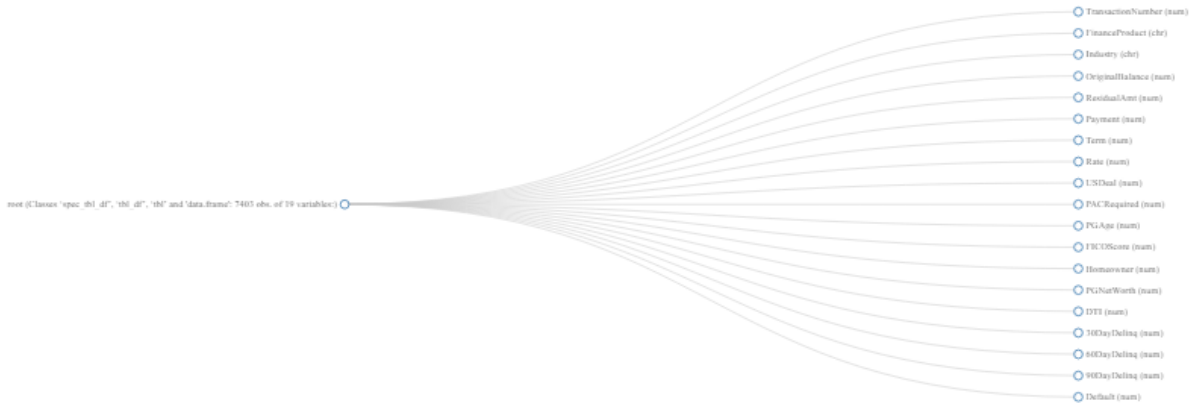
Appendix A – The Data Set

Column	Description
Transaction Number	The unique ID for each transaction record in the dataset.
Finance Product	The type of contract between the borrower and lender.
Industry	The company defined Industry that the borrower falls under.
Original Balance	The original contract amount advanced to the borrower.
Residual Amt	The residual/balloon payment at the end of the contract.
Payment	The borrower's payment amount.
Term	The term of the contract (months).
Rate	The rate of the contract.
US Deal	Whether the borrower is a US company (1) or not (0).
PAC Required	Whether the borrower was on automatic payments (1) or not (0).
PG Age	The age of the personal borrower.
FICO Score	The credit score (using the FICO model) of the personal borrower.
Homeowner	Whether the personal borrower is a homeowner (1) or not (0).
PG Net Worth	The net worth (assets minus liabilities) of the personal borrower.
DTI	The personal borrower's debt-to-income ratio.
30DayDelinq	The number of times the borrower was 30 days past due.
60DayDelinq	The number of times the borrower was 60 days past due.
90DayDelinq	The number of times the borrower was 90 days past due.
Default	Whether the borrower defaulted (1) or not (0).

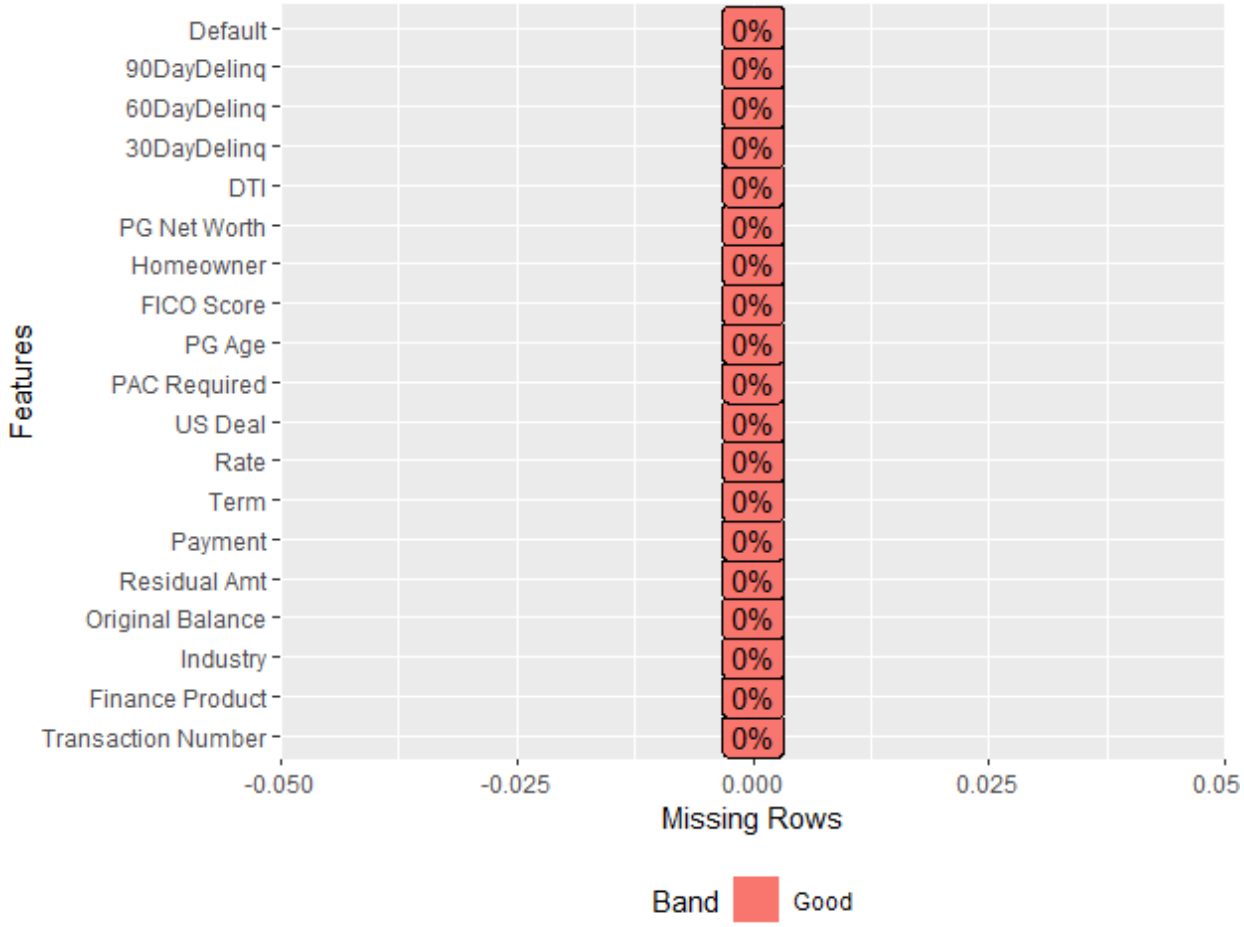
Appendix B – Overall Data Set Exploration

```
> str(loanData)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':        7403 obs. of  19 variables:
 $ Transaction Number: num  1 2 3 4 5 6 7 8 9 10 ...
 $ Finance Product   : chr  "TRAC" "TRAC" "TRAC" "TRAC" ...
 $ Industry          : chr  "OTR Trucking" "OTR Trucking" "OTR Trucking" "OTR Trucking" ...
 $ Original Balance  : num  9313 158234 167781 93960 120909 ...
 $ Residual Amt      : num  1863 31647 33556 18792 24182 ...
 $ Payment           : num  1035 2221 2380 2610 1679 ...
 $ Term              : num  9 61 63 38 36 38 38 38 30 46 ...
 $ Rate              : num  0.1 0.0524 0.0528 0.0667 0.0662 0.0661 0.0653 0.0653 0.0681 0.0755 ...
 $ US Deal           : num  1 1 1 1 1 1 1 1 1 1 ...
 $ PAC Required      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PG Age            : num  52 43 19 61 31 57 59 52 21 36 ...
 $ FICO Score        : num  730 729 784 794 729 704 769 720 714 757 ...
 $ Homeowner         : num  1 1 1 1 1 1 1 1 1 1 ...
 $ PG Net Worth      : num  558291 2818876 1302722 1284320 2624638 ...
 $ DTI               : num  0.58 0.28 0.45 0.08 0.25 0.14 0.1 0.4 0.5 0.42 ...
 $ 30DayDelinq       : num  0 0 0 0 1 0 0 0 0 0 ...
 $ 60DayDelinq       : num  0 0 0 0 0 0 0 0 0 0 ...
 $ 90DayDelinq       : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Default           : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
> plot_str(loanData)
```



```
> plot_missing(loanData)
```




```
#Remove spacing in colNames
library(janitor)
loanData = clean_names(loanData)
head(loanData)
```

```
  transaction_num~ finance_product industry original_balance residual_amt payment  term   rate
             <dbl> <chr>           <chr>                <dbl>        <dbl>   <dbl> <dbl>  <dbl>
1                1 TRAC            OTR Tru~              9313        1863.    1035.     9 0.1
2                2 TRAC            OTR Tru~            158234       31647.    2221.    61 0.0524
3                3 TRAC            OTR Tru~            167781       33556.    2380.    63 0.0528
4                4 TRAC            OTR Tru~             93960       18792     2610     38 0.0667
5                5 TRAC            OTR Tru~            120909       24182.    1679.    36 0.0662
6                6 TRAC            OTR Tru~             62573       12515.    1738.    38 0.0661
# ... with 11 more variables: us_deal <dbl>, pac_required <dbl>, pg_age <dbl>,
#   fico_score <dbl>, homeowner <dbl>, pg_net_worth <dbl>, dti <dbl>, x30day_delinq <dbl>,
#   x60day_delinq <dbl>, x90day_delinq <dbl>, default <dbl>
```

```
#Checking formating issues with categorial variables

unique(loanData$finance_product)
unique(loanData$industry)
```

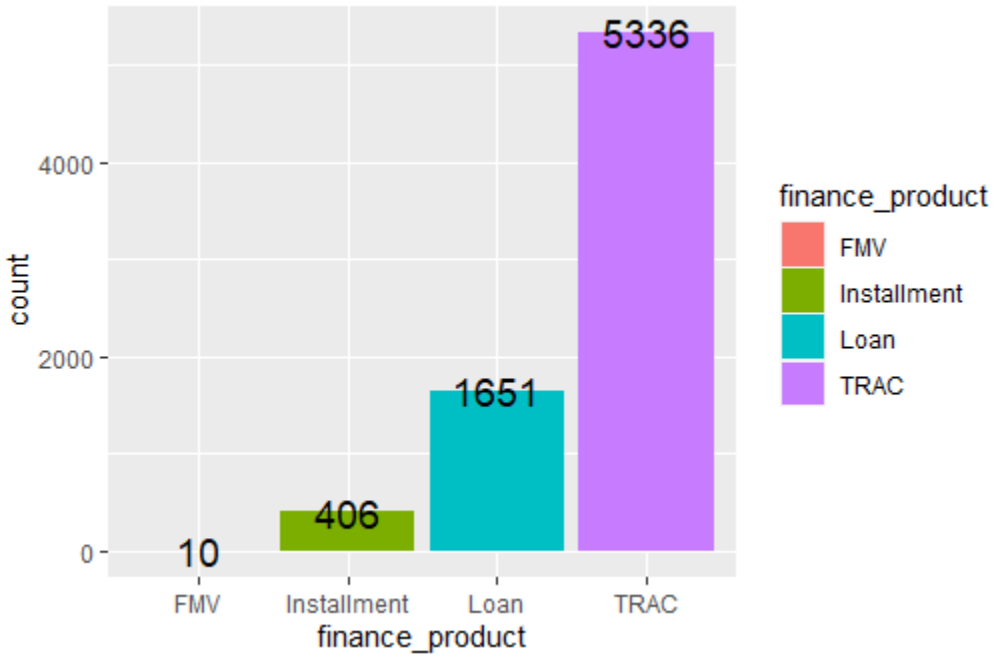
```
> unique(loanData$finance_product)
[1] "TRAC"        "Loan"        "Installment" "FMV"
> unique(loanData$industry)
[1] "OTR Trucking"       "Petroleum"          "Tool Truck"          "Other"
[5] "Oil & Gas"          "Hydrovac - Non O&G" "Motorcoach"
```

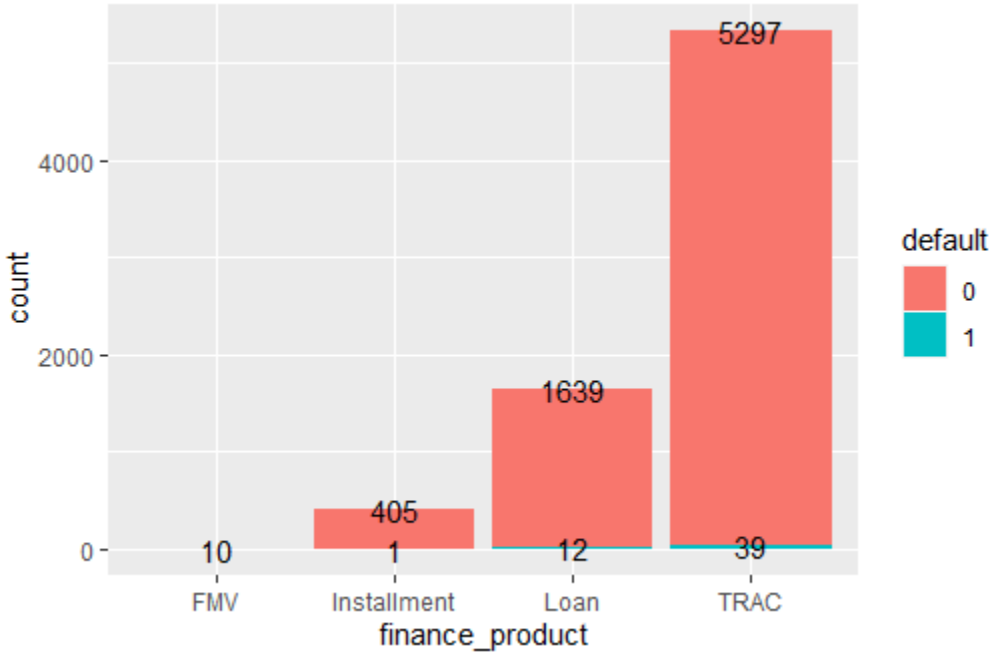
```
#Declearing categorical variables as a factor

loanData$finance_product=factor(loanData$finance_product)
loanData$industry=factor(loanData$industry)
loanData$default=factor(loanData$default)
loanData$us_deal=factor(loanData$us_deal)
loanData$homeowner=factor(loanData$homeowner)
loanData$pac_required=factor(loanData$pac_required)
```

```
> str(loanData)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':        7403 obs. of  19 variables:
 $ transaction_number: num  1 2 3 4 5 6 7 8 9 10 ...
 $ finance_product   : Factor w/ 4 levels "FMV","Installment",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ industry          : Factor w/ 7 levels "Hydrovac - Non O&G",..: 5 5 5 5 5 5 5 5 5 6 ...
 $ original_balance  : num  9313 158234 167781 93960 120909 ...
 $ residual_amt      : num  1863 31647 33556 18792 24182 ...
 $ payment           : num  1035 2221 2380 2610 1679 ...
 $ term              : num  9 61 63 38 36 38 38 38 30 46 ...
 $ rate              : num  0.1 0.0524 0.0528 0.0667 0.0662 0.0661 0.0653 0.0653 0.0681 0.0755 ...
 $ us_deal           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ pac_required      : Factor w/ 1 level "0": 1 1 1 1 1 1 1 1 1 1 ...
 $ pg_age            : num  52 43 19 61 31 57 59 52 21 36 ...
 $ fico_score        : num  730 729 784 794 729 704 769 720 714 757 ...
 $ homeowner         : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ pg_net_worth      : num  558291 2818876 1302722 1284320 2624638 ...
 $ dti               : num  0.58 0.28 0.45 0.08 0.25 0.14 0.1 0.4 0.5 0.42 ...
 $ x30day_delinq     : num  0 0 0 0 1 0 0 0 0 0 ...
 $ x60day_delinq     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ x90day_delinq     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ default           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

From the overall data exploration, there are 7403 observations, 19 variables, there are no missing values, and no formatting issues with categorical variables. The variables finance product, industry, default, us deal, pac required, and homeowner were set as a factor.

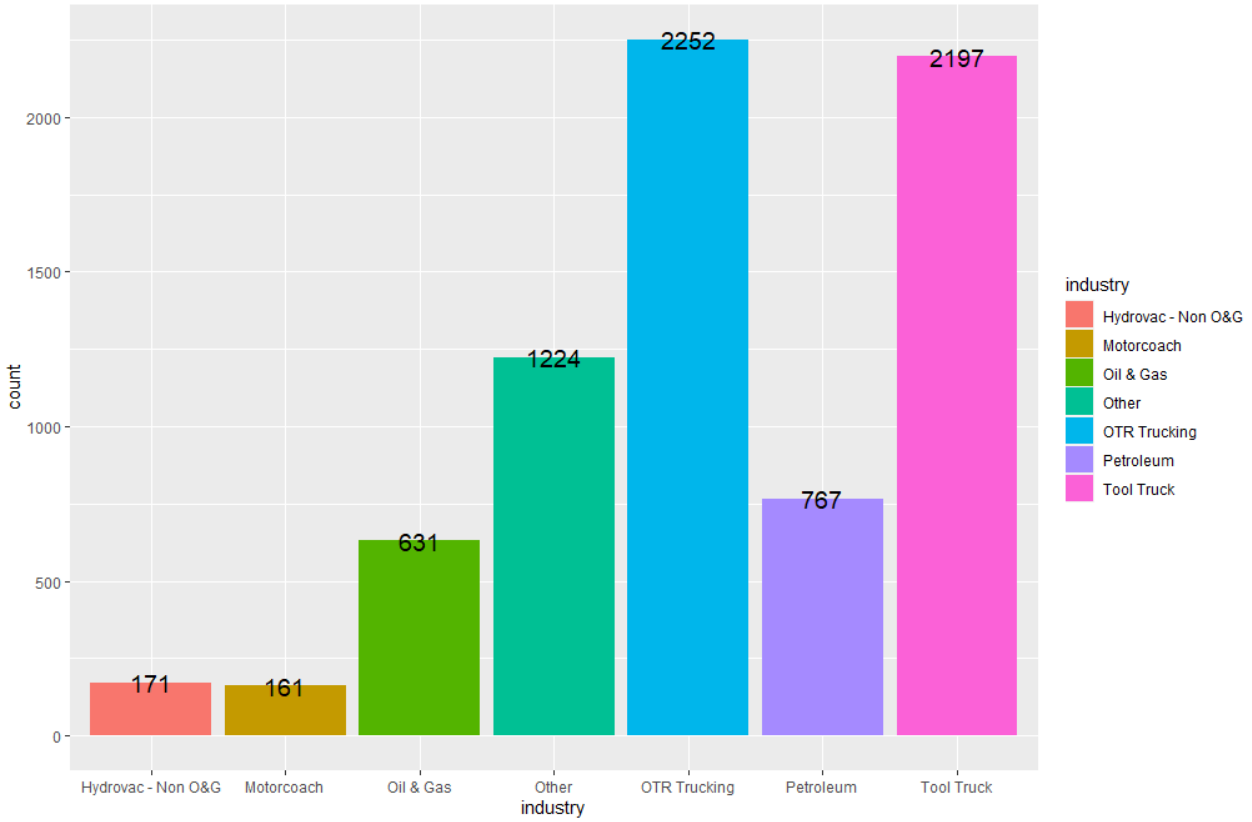
Appendix C – Finance Product



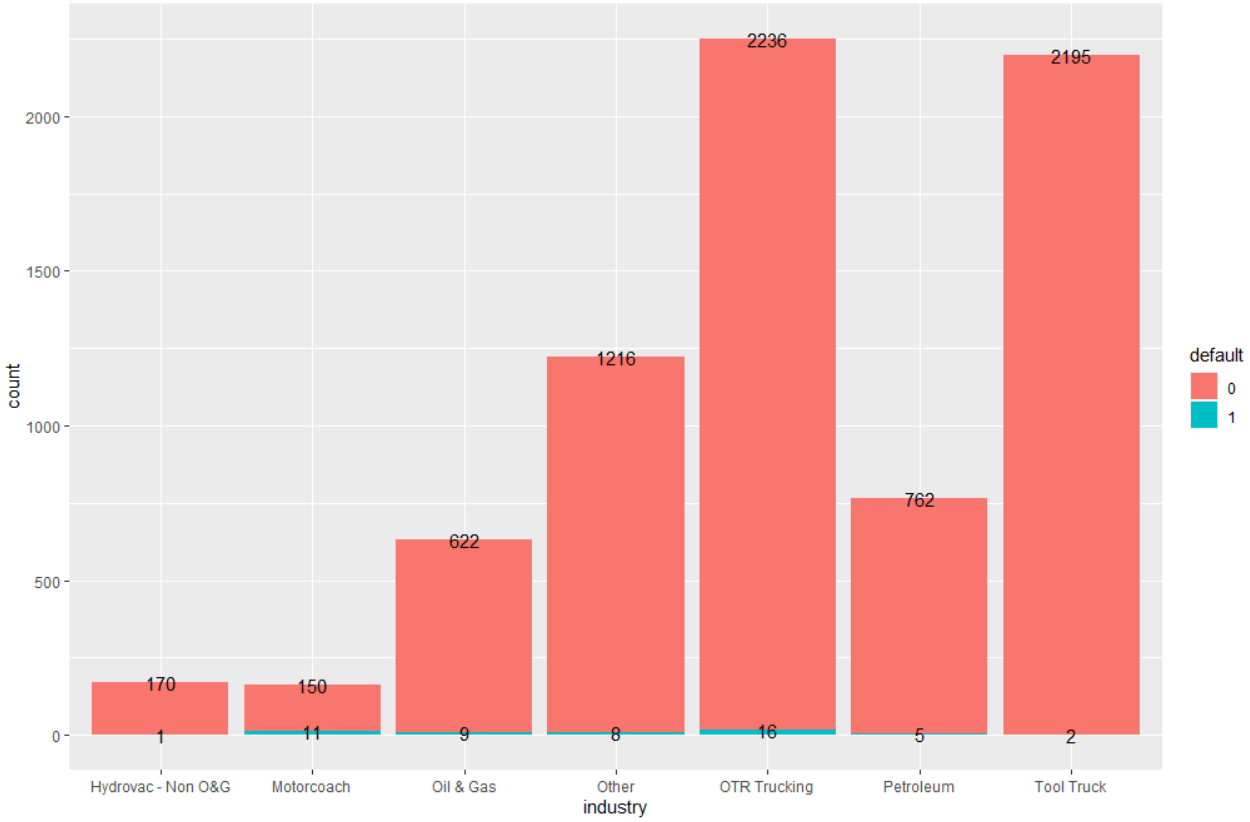


From the previous plots, 72.08% of the customers have a TRAC finance product. Among those customers, 0.73% are in default. Of the 22.30% of the customers that have a loan, 0.73% are in default. Across all lines of finance products, 0.70% of customers are in default.

Appendix D – Industry



OTR Trucking is the most representative industry for the company with 30.42% of all finance products taken by customers, followed by Tool Truck with 29.68%, other with 16.53%, and oil & gas 8.5%.



OTR Trucking industry has the highest number of default customers (16), followed by Motorcoach (11), and oil and gas (9).

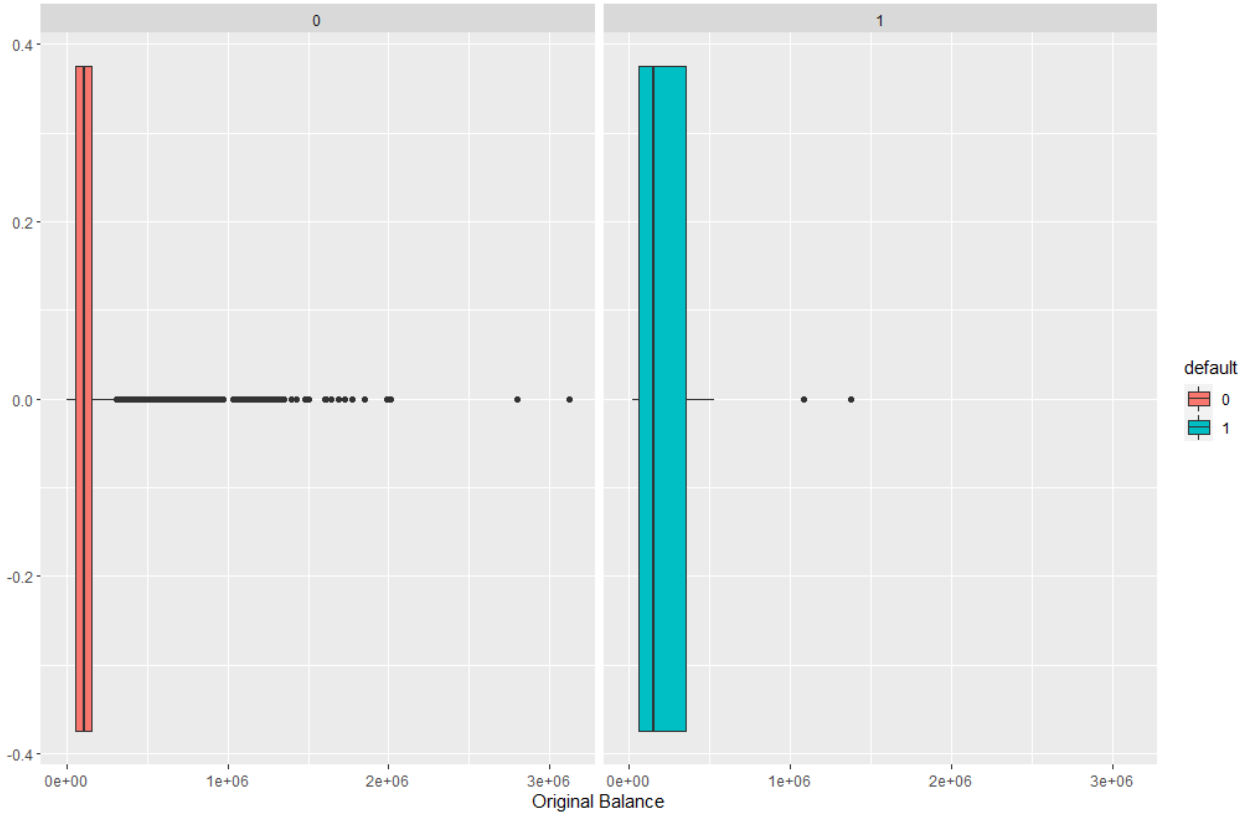
Appendix E – Original Balance

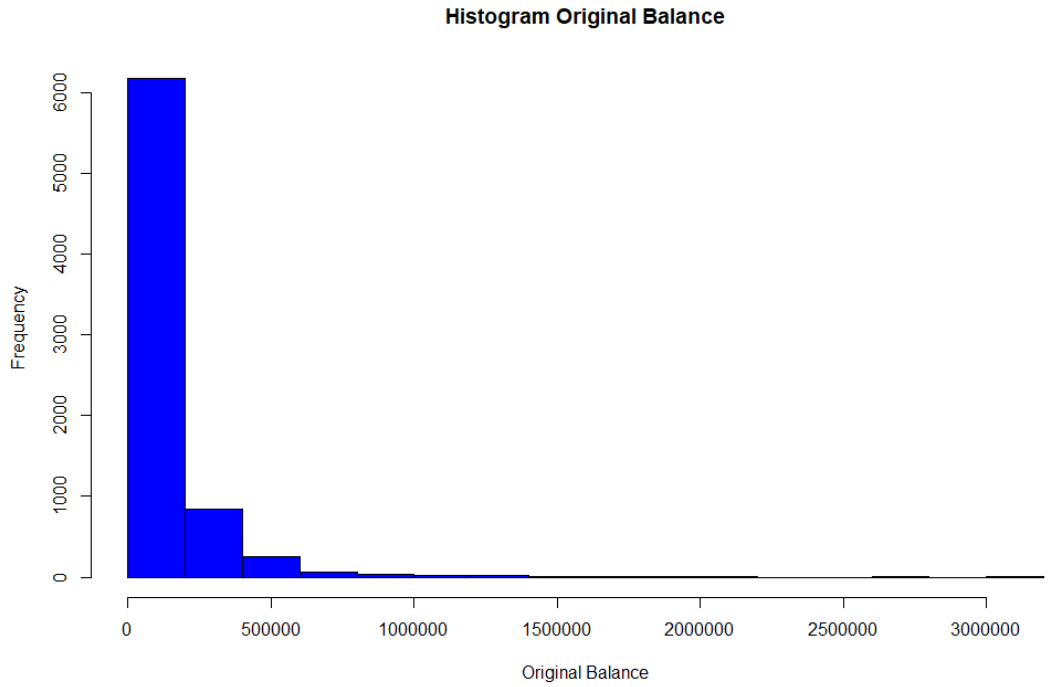
```
> summary(loanData$original_balance)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0   55445  108162  143532  156441 3121835
```

```
library(psych)
```

```
describeBy(loanData$original_balance, group = loanData$default)
```

```
 Descriptive statistics by group
group: 0
   vars    n     mean     sd median  trimmed      mad min     max   range skew kurtosis      se
X1    1 7351 142835.7 168211 108041 112217.3 75685.25   0 3121835 3121835 5.15    46.49 1961.92
------------------------------------------------------------------------------------------
group: 1
   vars  n     mean       sd   median  trimmed      mad   min     max   range skew kurtosis       se
X1    1 52 242030.3 258352.1 155526.5 200575.9 175121.8 21627 1377050 1355423 2.28      6.6 35826.99
```



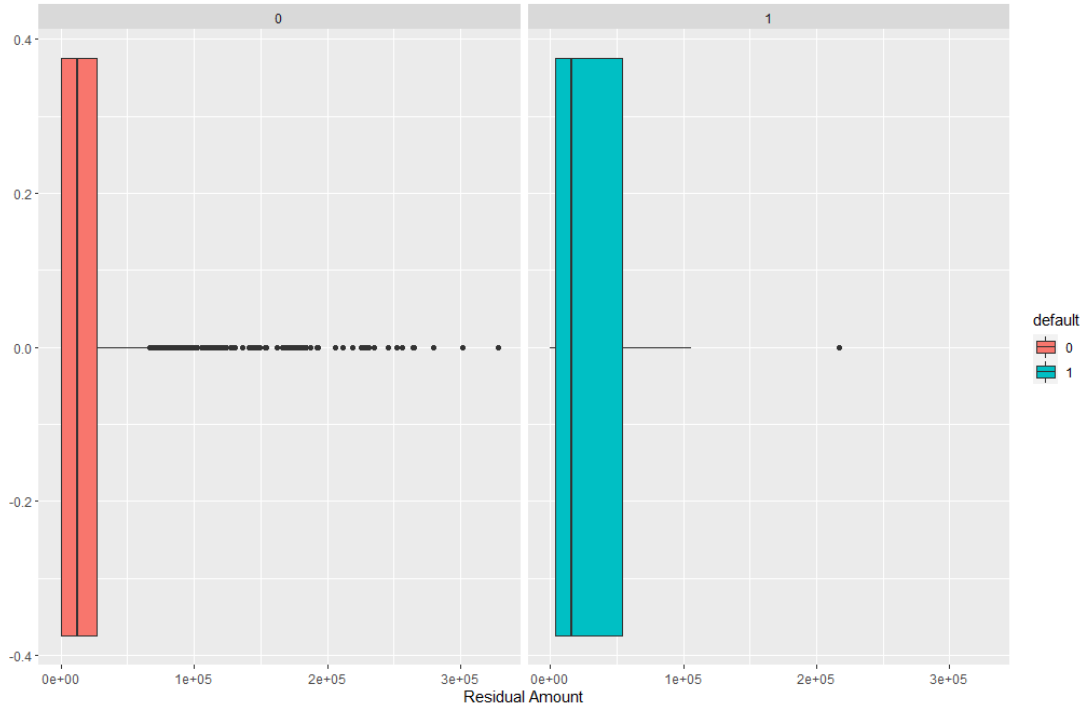


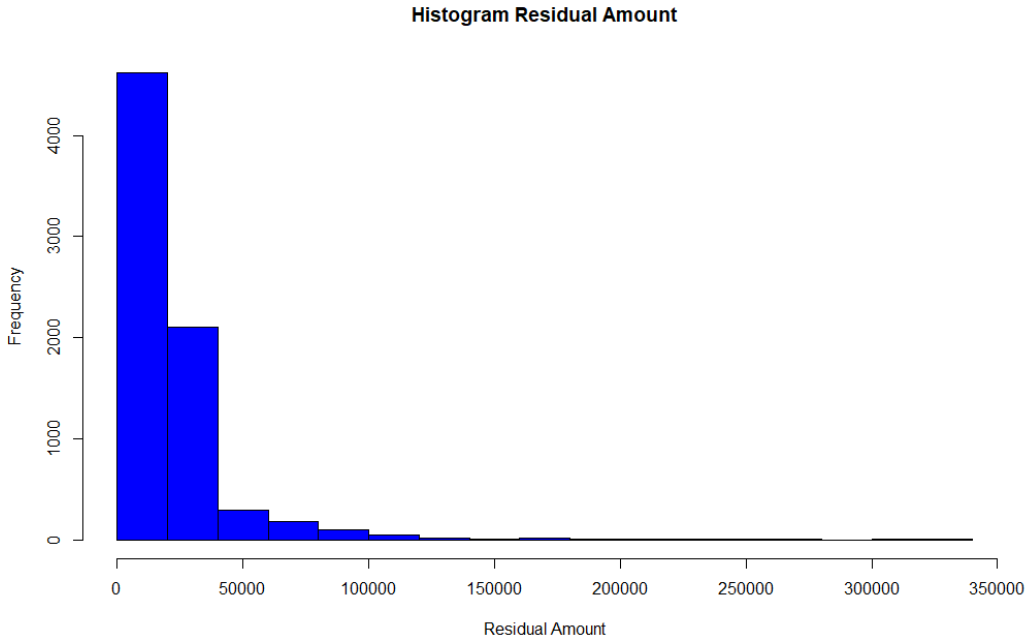
Original balance has a median of $108,162 across all types of customers. However, customers in default have an original balance median of $155,526 that is higher than non-default customers

($108,041). From the boxplot above, we can identify several outliers, but these points are still important to take into consideration because there is not a standard original balance amount per customer and this amount can vary based on the credit needs of each customer.

Appendix E – Residual Amount

```
> summary(loanData$residual_amt)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0       0   12195   18198   26718  328531
> describeBy(loanData$residual_amt, group = loanData$default)

 Descriptive statistics by group
group: 0
   vars    n     mean       sd  median  trimmed      mad min    max  range skew kurtosis     se
X1    1 7351 18081.75 24403.84 12192.2 13680.26 18076.16   0 328531 328531 3.85    26.03 284.63
------------------------------------------------------------------------------------------
group: 1
   vars  n     mean       sd  median trimmed      mad min      max    range skew kurtosis      se
X1    1 52 34692.62 43017.06 16331.5 27875.3 24213.08   0 216634.8 216634.8 1.79     4.08 5965.39
```

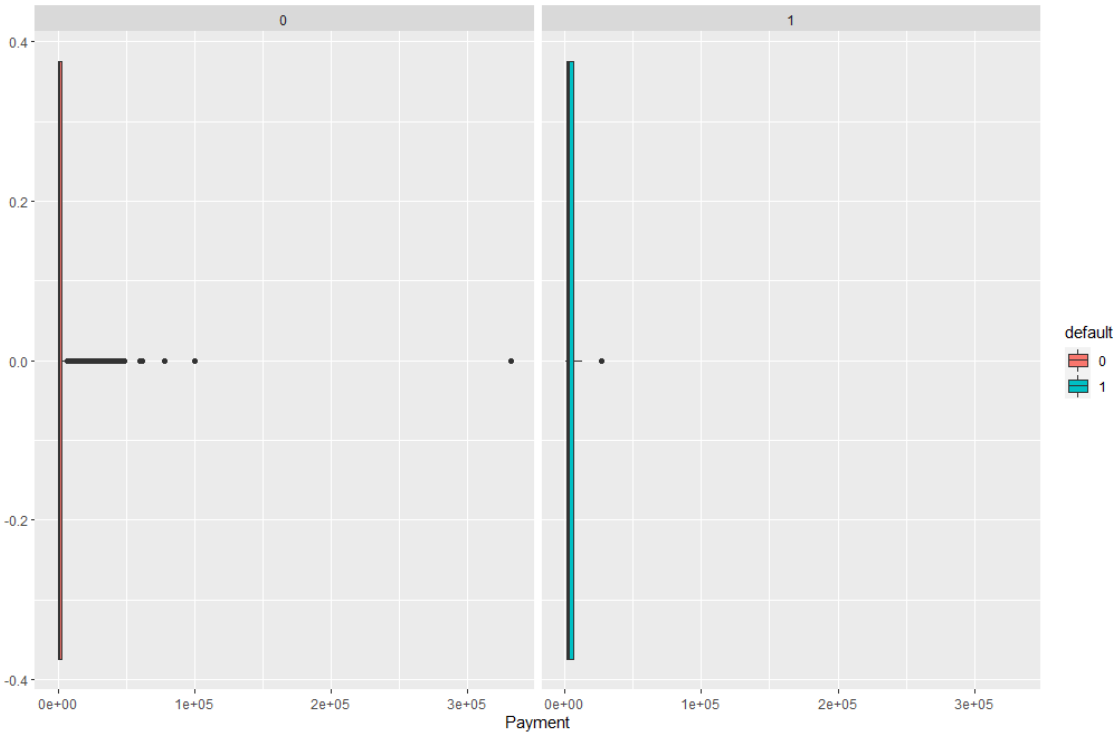


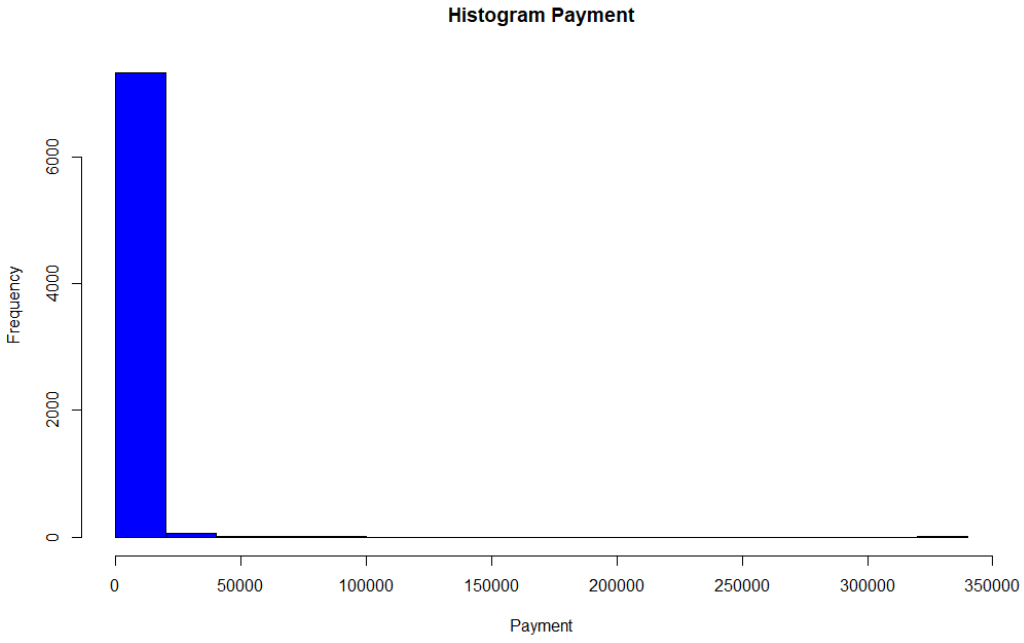


Residual amount has a median of $12,195 across all types of customers. However, customers in default have a residual amount median of $16,632 that is higher than non-default customers ($12,192). From the boxplot above, we can identify several outliers, but these points are still important to take into consideration because there is not a standard residual amount per customer.

Appendix F – Payment

```
> summary(loanData$payment)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
     0.0    471.9   1487.8   2769.1   3089.6 331890.3

> describeBy(loanData$payment, group = loanData$default)

 Descriptive statistics by group
group: 0
   vars    n    mean      sd  median trimmed     mad min      max    range  skew kurtosis    se
X1    1 7351 2758.28 5863.27 1479.19  1867.8 1547.52   0 331890.3 331890.3 26.86  1373.46 68.39
------------------------------------------------------------------------------------------
group: 1
   vars  n    mean      sd  median trimmed     mad    min      max    range skew kurtosis     se
X1    1 52 4304.66 4463.96 2938.36 3634.19 3511.38 383.79 26943.57 26559.78 2.64    10.35 619.04
```

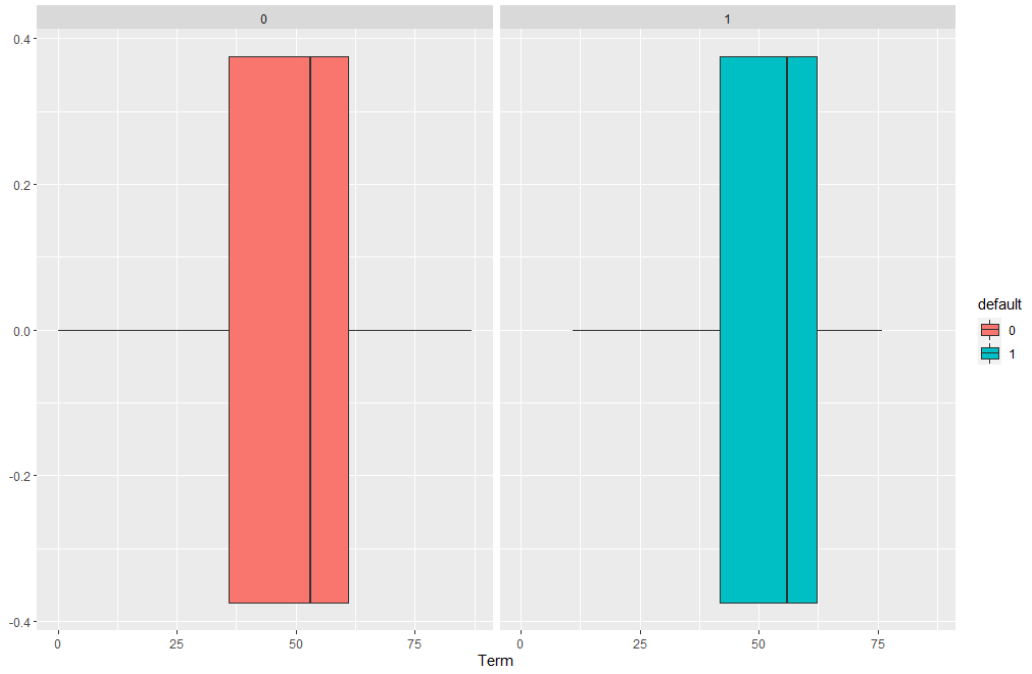


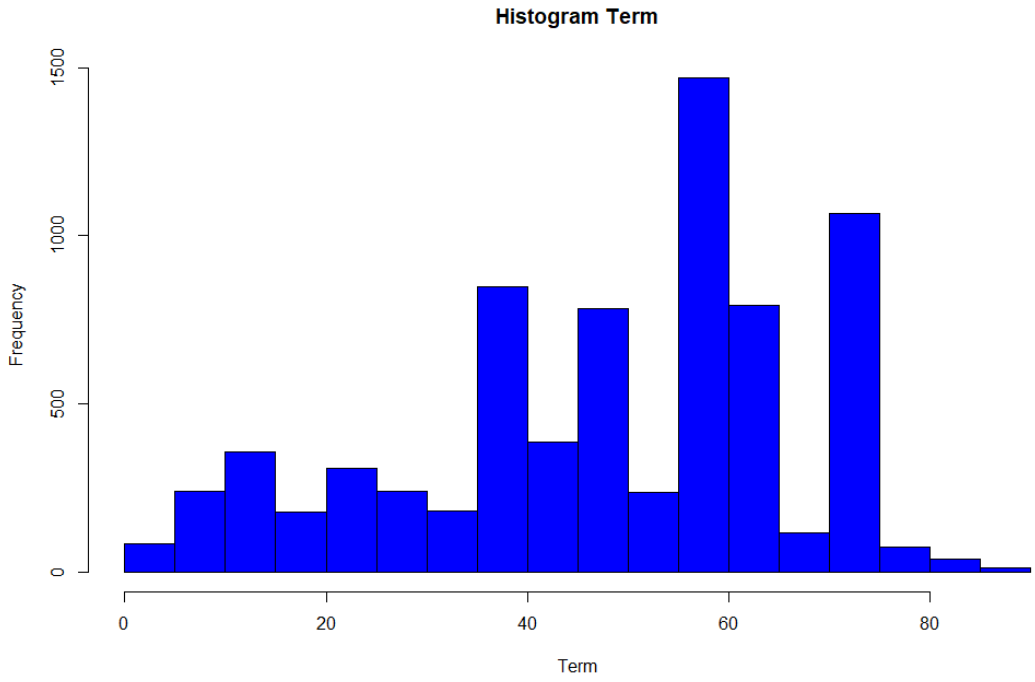


Payment has a median of $1,488 across all types of customers. However, customers in default have a payment amount median of $2,938.36 that is higher than non-default customers ($1,479.19). From the boxplot above, we can identify several outliers, but these points are still important to take into consideration because they depend on the amount of the loan taken.

Appendix G – Term

```
> summary(loanData$term)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   36.00   53.00   48.65   61.00   87.00

> describeBy(loanData$term, group = loanData$default)

 Descriptive statistics by group
group: 0
   vars    n  mean    sd median trimmed   mad min max range  skew kurtosis   se
X1    1 7351 48.62 19.41     53   50.17 19.27   0  87    87 -0.56     -0.6 0.23
------------------------------------------------------------------------------
group: 1
   vars  n  mean    sd median trimmed   mad min max range  skew kurtosis   se
X1    1 52 52.54 16.54     56   53.95 16.31  11  76    65 -0.69     0.02 2.29
```

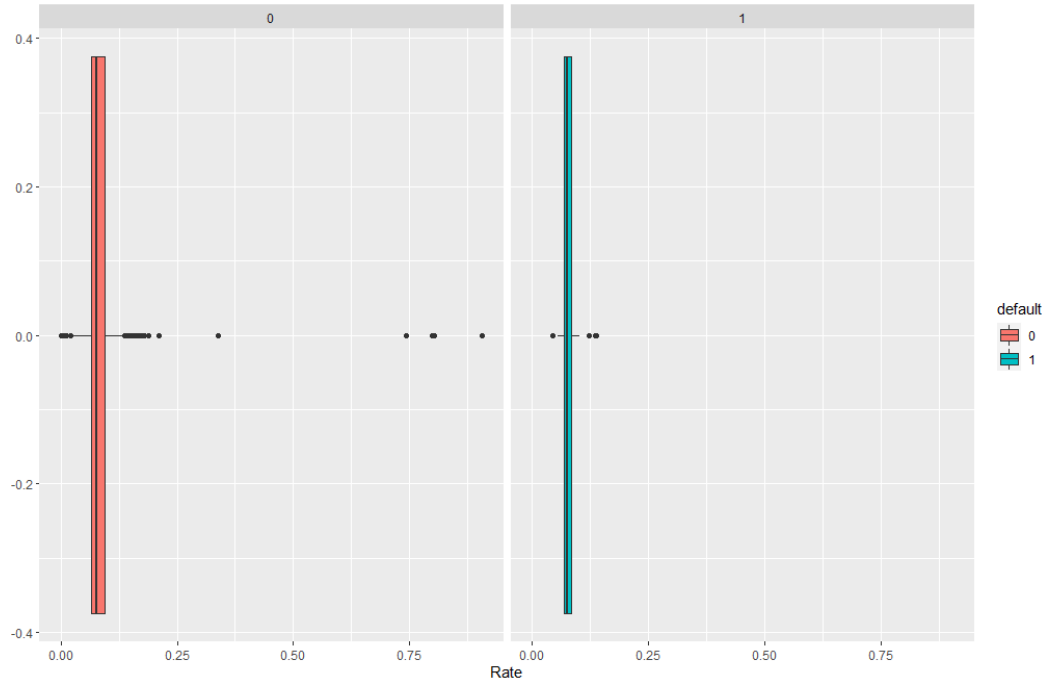


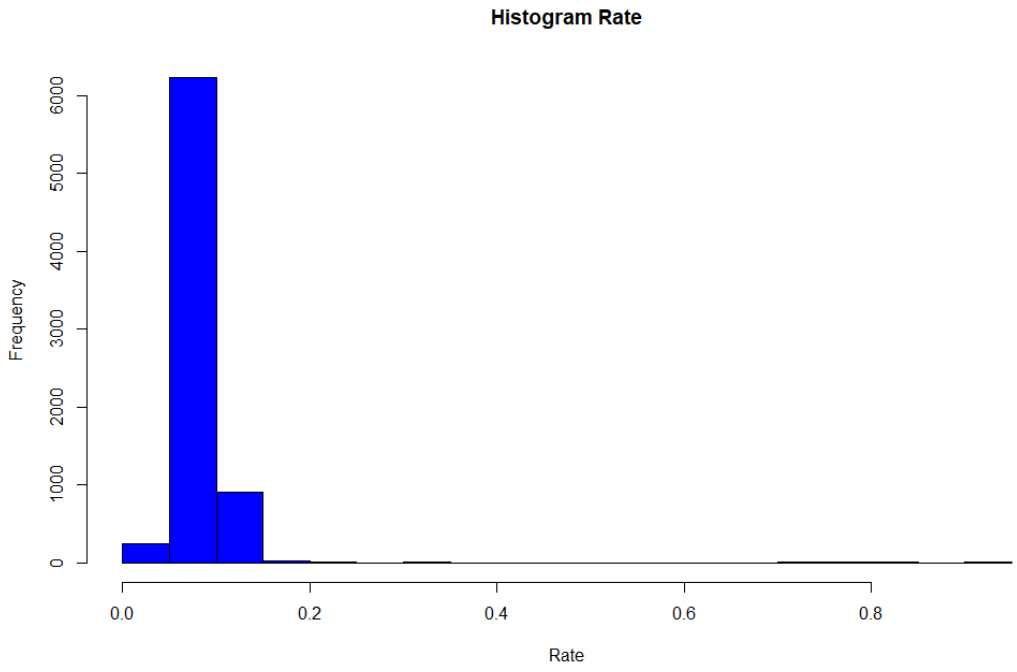


From above can be concluded that default customers have a term average of 52.54 months, that is slightly higher than non-default customers which is 48.62 months.

Appendix H – Rate

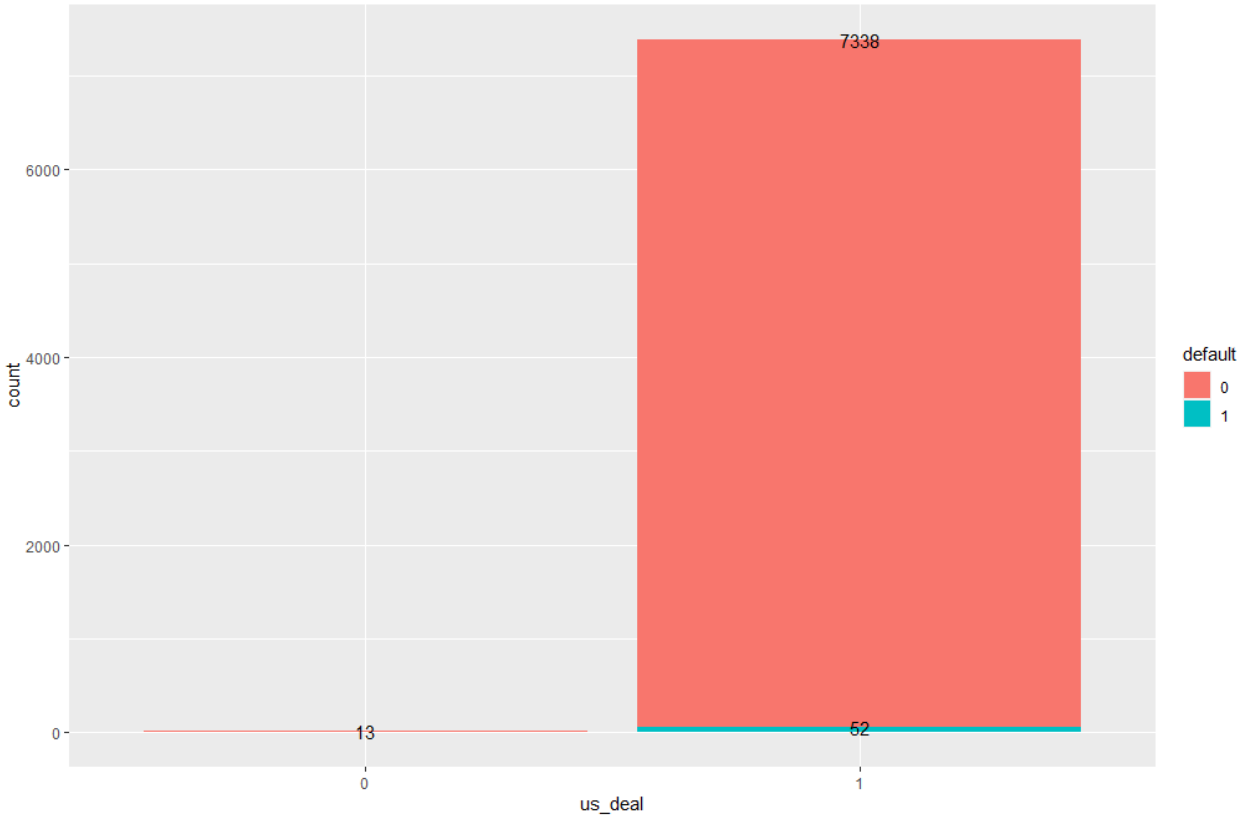
```
> summary(loanData$rate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.06540 0.07630 0.07995 0.09360 0.90660

> describeBy(loanData$rate, group = loanData$default)

 Descriptive statistics by group
group: 0
   vars    n mean   sd median trimmed  mad min  max range  skew kurtosis se
X1    1 7351 0.08 0.03   0.08    0.08 0.02   0 0.91  0.91 10.45   275.12  0
------------------------------------------------------------------------------
group: 1
   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis se
X1    1 52 0.08 0.02   0.08    0.08 0.01 0.05 0.14  0.09 1.63     3.58  0
```





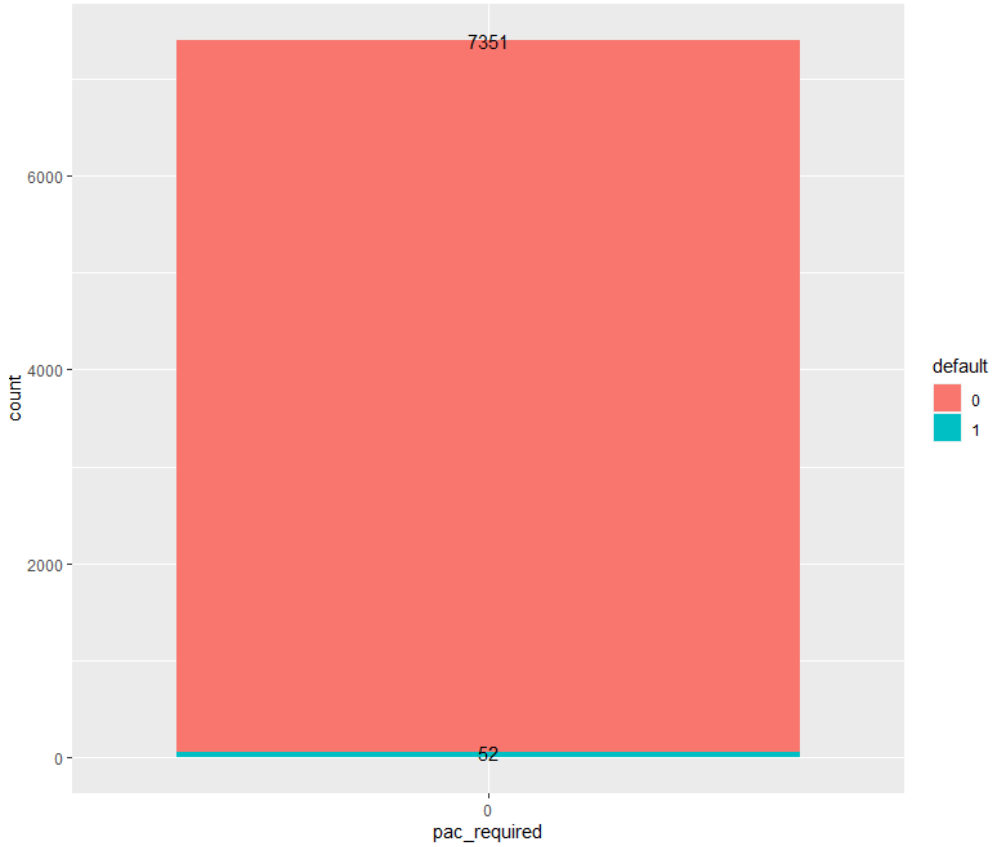
From above we can conclude that the median rate for both default and non-default customers is 8%. Some outliers have been identified but we can't discard them because the rates can vary based on the risk that the customer represents.

Appendix I – US deal



From above can be conclude that 99.8% of customers are in the United States and all default customers belong to that group.

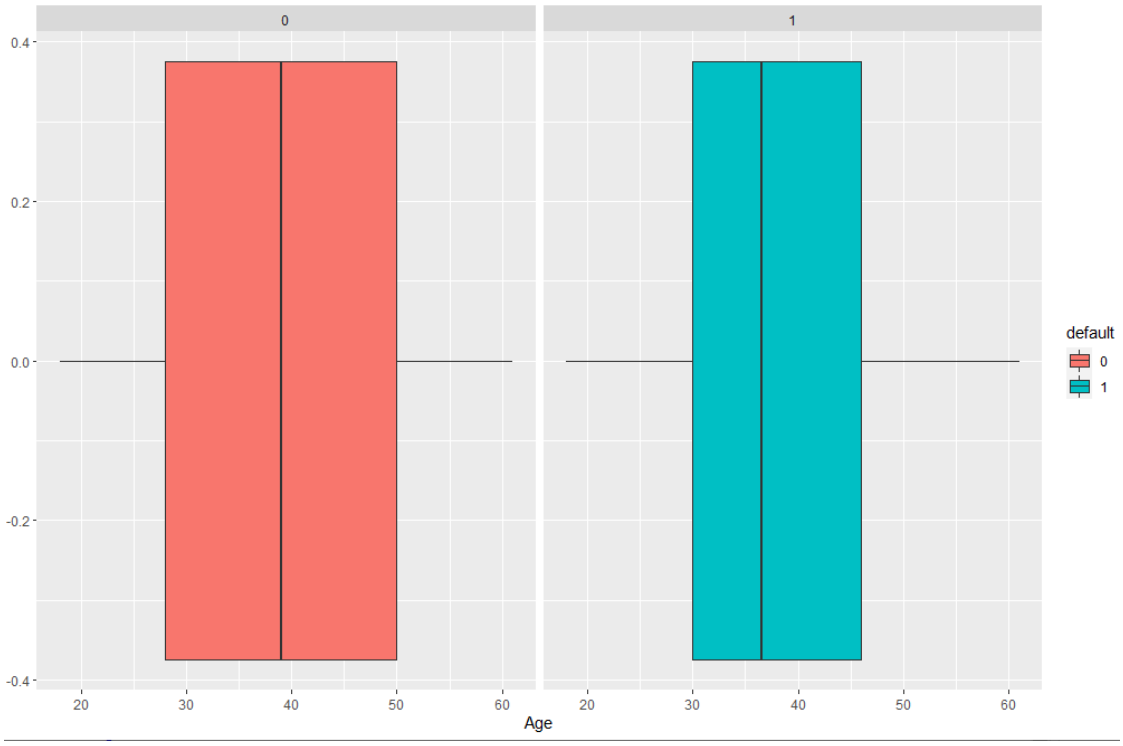
Appendix J – PAC Required (Auto Payment)

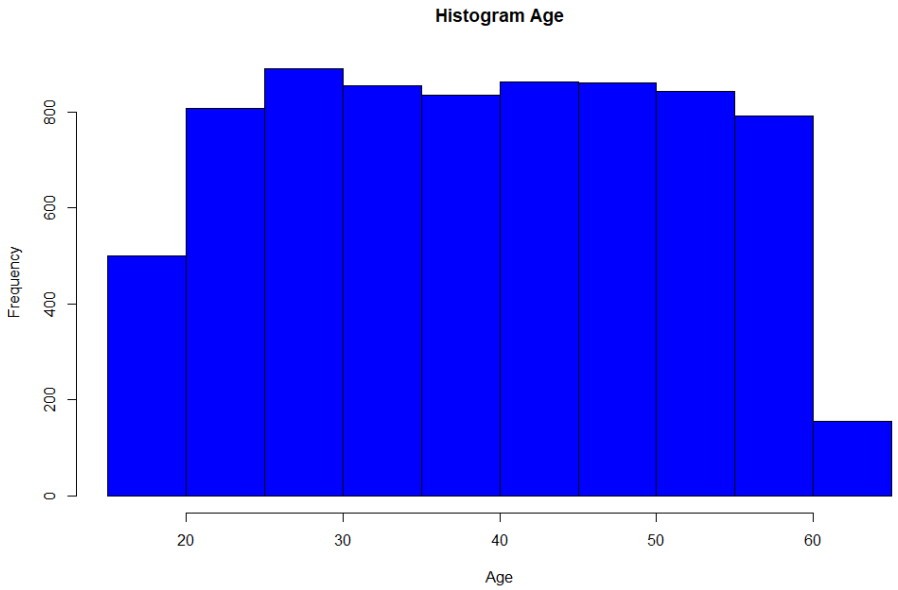


The graphic above shows that none of the customers had auto-payment.

Appendix K – PG (Personal Guarantor) Age.

```
> summary(loanData$pg_age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.00   28.00   39.00   39.38   50.00   61.00

> describeBy(loanData$pg_age, group = loanData$default)

 Descriptive statistics by group
group: 0
   vars    n  mean   sd median trimmed   mad min max range skew kurtosis   se
X1    1 7351 39.39 12.6     39   39.38 16.31  18  61    43 0.01    -1.19 0.15
------------------------------------------------------------------------------
group: 1
   vars  n  mean    sd median trimmed   mad min max range skew kurtosis   se
X1    1 52 37.56 11.39   36.5    37.1 11.86  18  61    43 0.32    -0.79 1.58
```



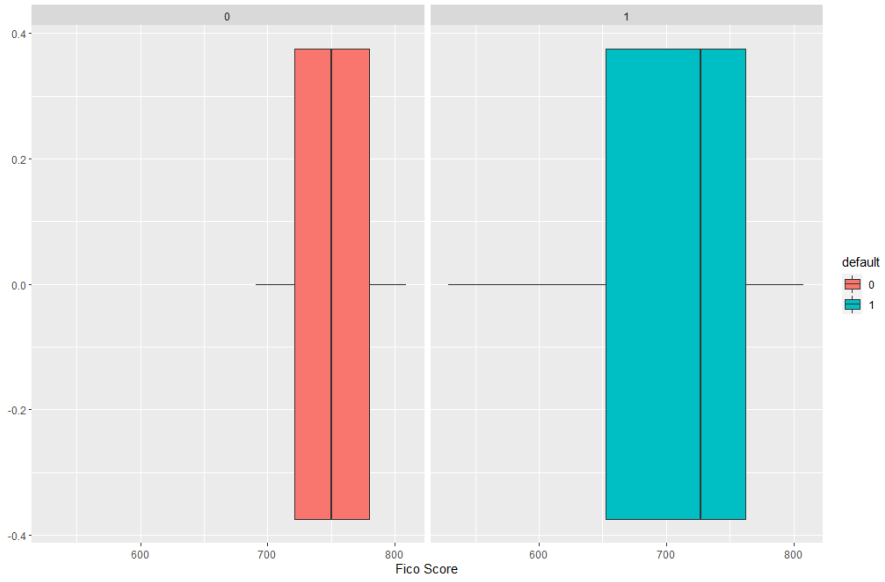


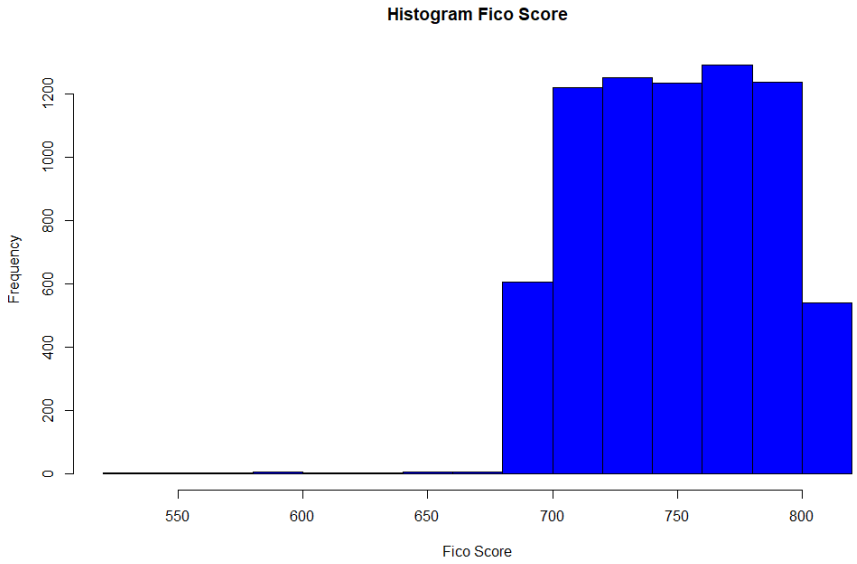
From above, we conclude that customers average age is 39, and customers in default are on average 2.5 years younger than non-default customers.

Appendix L – FICO Score

```
> summary(loanData$fico_score)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  529.0   721.0   750.0   749.9   779.0   809.0
```

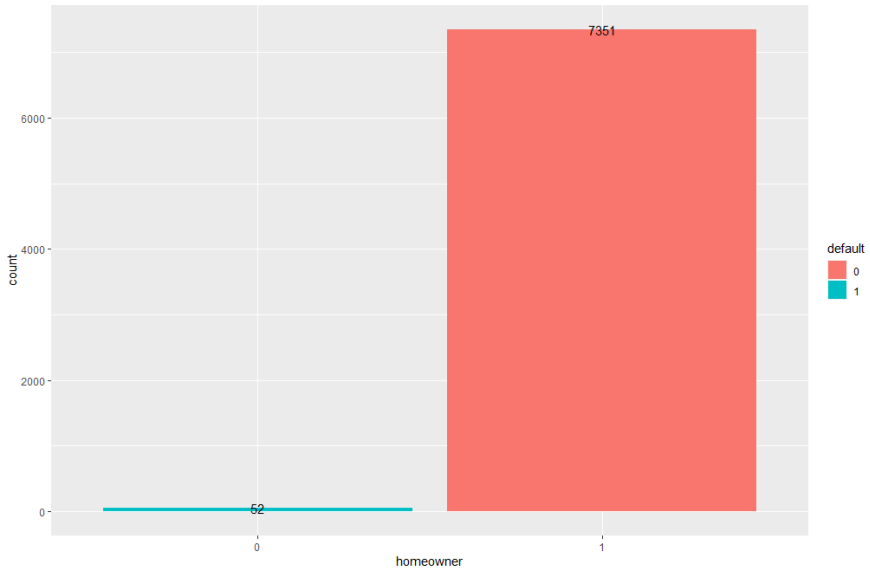
```
> describeBy(loanData$fico_score, group = loanData$default)

 Descriptive statistics by group
group: 0
   vars    n  mean    sd median trimmed mad min max range  skew kurtosis  se
X1    1 7351 750.2 34.16    750  750.24  43 691 809   118 -0.01     -1.2 0.4
------------------------------------------------------------------------------
group: 1
   vars  n   mean    sd median trimmed   mad min max range  skew kurtosis    se
X1    1 52 704.25 73.13    727  710.21 61.53 529 808   279 -0.68    -0.71 10.14
```





From the above, we conclude that the customers' average FICO score is 750. However, customers in default have on average a lower FICO score (704) in contrast with non-default customers with an average FICO score of 750.

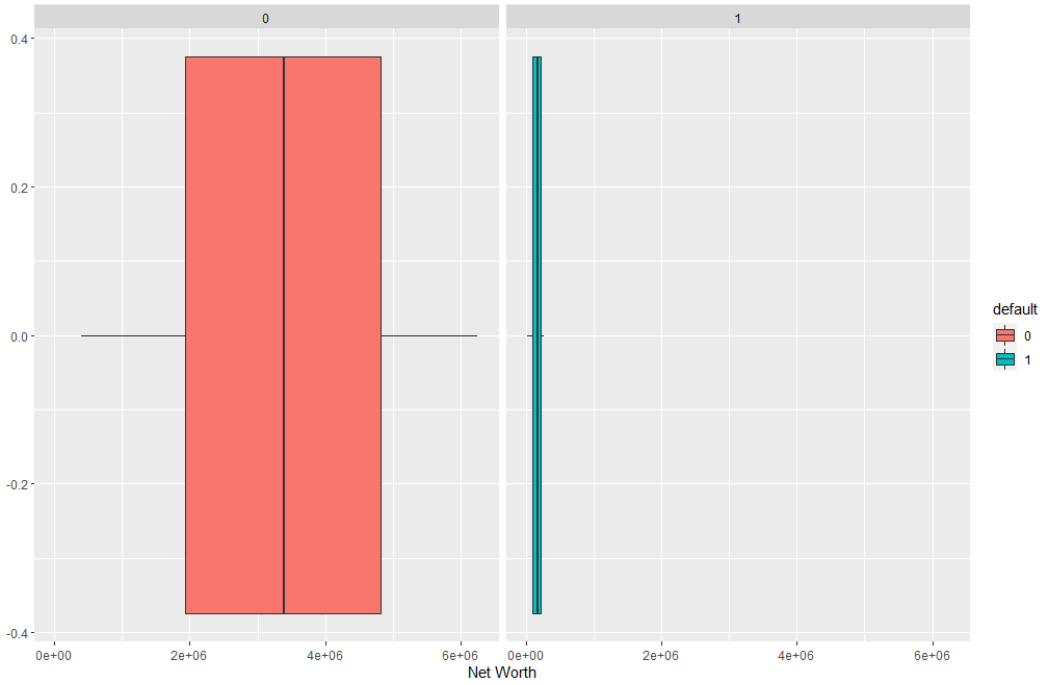
Appendix M – Homeownership

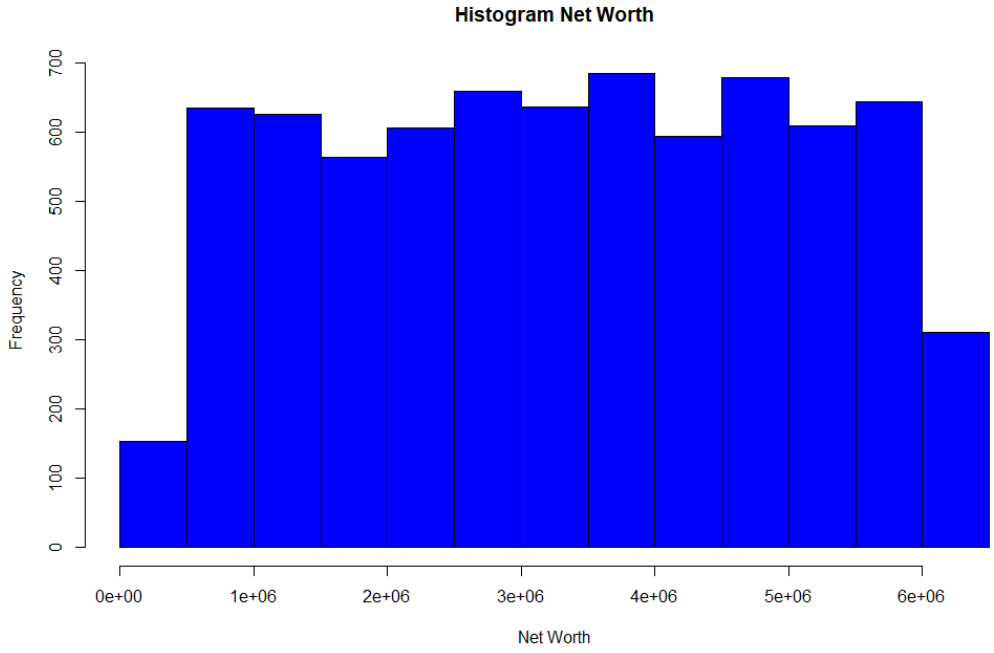


From above can be concluded that no customers in default are homeowners.

Appendix N – Net Worth

```
> summary(loanData$pg_net_worth)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23750 1894900 3366343 3333782 4801584 6246892


 Descriptive statistics by group
group: 0
   vars    n    mean      sd  median trimmed     mad    min     max   range  skew kurtosis       se
X1    1 7351 3356232 1675427 3385621 3361742 2135562 400862 6246892 5846030 -0.03    -1.18 19541.24
------------------------------------------------------------------------------------------------
group: 1
   vars  n     mean      sd median  trimmed     mad   min    max  range  skew kurtosis       se
X1    1 52 160182.6 75329.7 175879 163424.7 94805.6 23750 261623 237873 -0.29    -1.27 10446.35
```

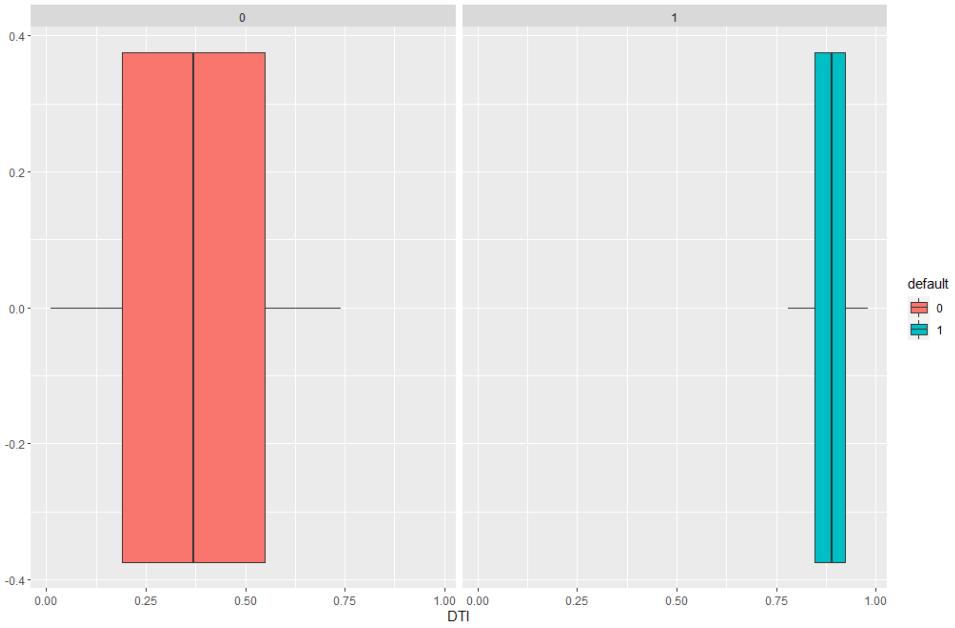


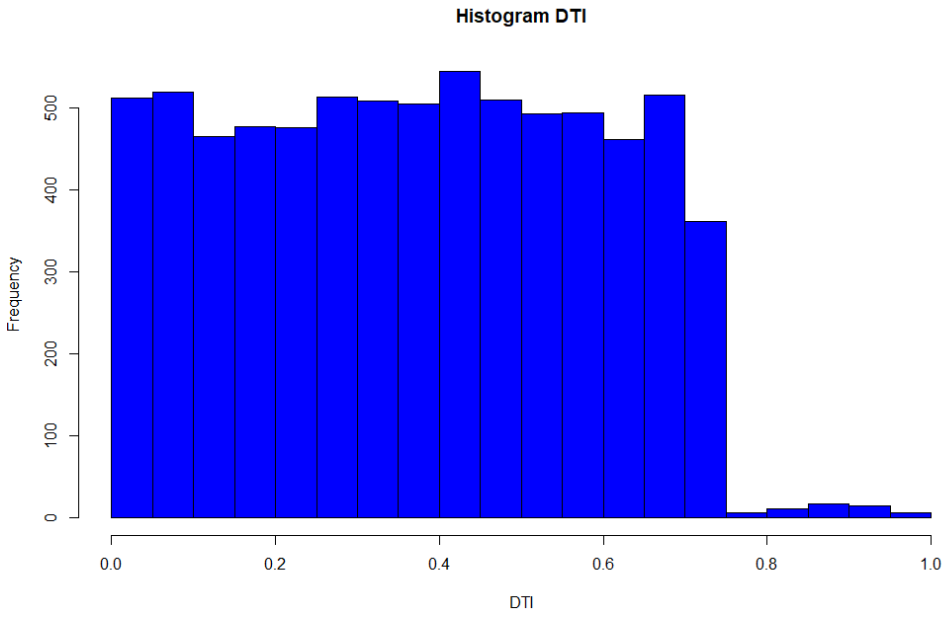


From above, we conclude that customers in default have an average net worth of $160,183 while non-default customers' net worth is $3,356,232.

Appendix O – DTI (Debt-to-Income Ratio)

```
> summary(loanData$dti)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0100  0.1900  0.3800  0.3767  0.5600  0.9800

 Descriptive statistics by group
group: 0
   vars    n mean   sd median trimmed  mad  min  max range skew kurtosis se
X1    1 7351 0.37 0.21   0.37    0.37 0.27 0.01 0.74  0.73    0    -1.18  0
------------------------------------------------------------------------------------------
group: 1
   vars  n mean   sd median trimmed  mad  min  max range  skew kurtosis   se
X1    1 52 0.88 0.05   0.89    0.88 0.06 0.78 0.98   0.2 -0.16    -0.92 0.01
```

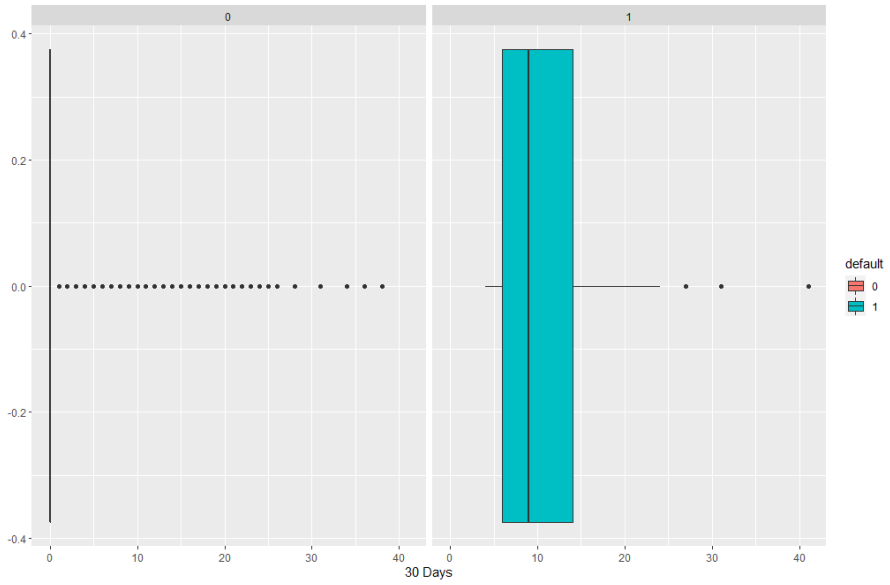


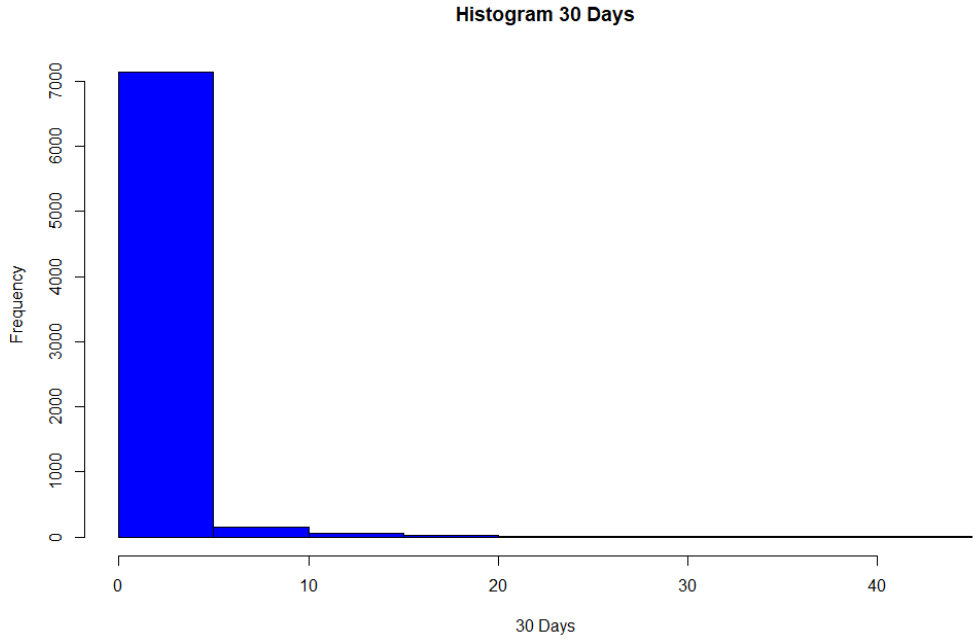


From above can be concluded that customers in default have an average DTI of 88% while non-default customers' DTI is 37%.

Appendix P– 30 Days Past Due

```
> summary(loanData$x30day_delinq)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.6443  0.0000 41.0000

 Descriptive statistics by group
group: 0
   vars    n mean   sd median trimmed mad min max range skew kurtosis   se
X1    1 7351 0.57 2.32      0    0.05   0   0  38    38 7.04    64.75 0.03
------------------------------------------------------------------------------
group: 1
   vars  n  mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 52 11.19 7.39      9     9.9 5.93   4  41    37 1.83     3.96 1.02
```



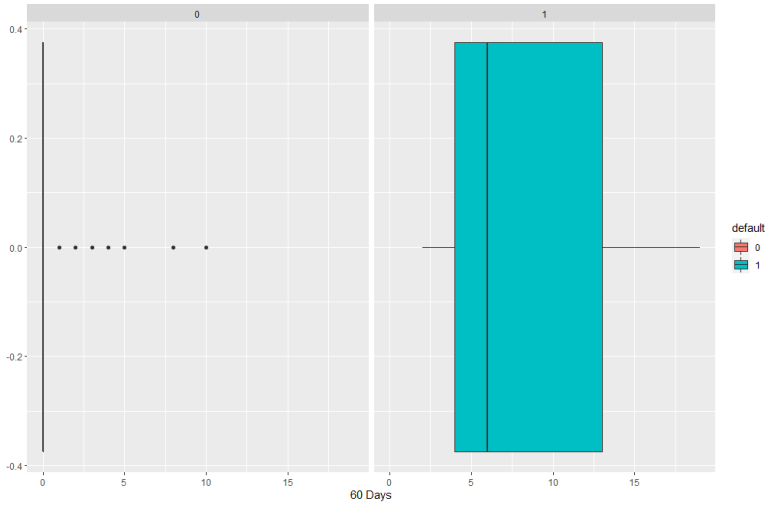


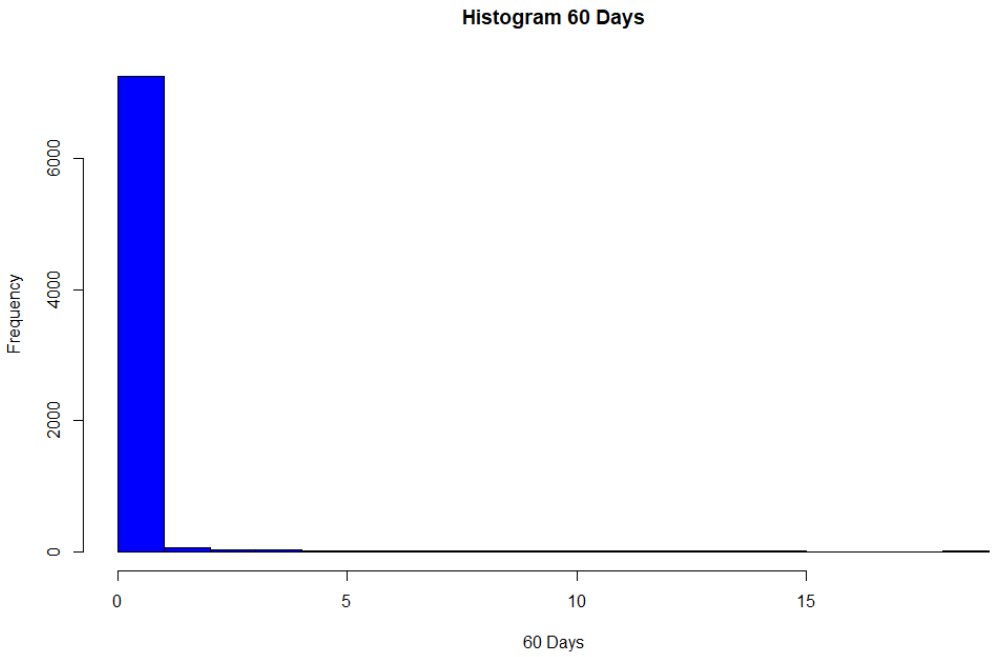
From above can be concluded that the default customers have a median of 9 occurrences where they did not make a payment in a 30-day period, while non-default customers median is zero (0). However, there are some outliers that tell us that some of those non-default customers had atypical payment behavior, hence the reason for retaining those outliers is because they have important information about customers' payment behavior.

Appendix Q – 60 Days Past Due

```
> summary(loanData$x60day_delinq)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.1112  0.0000 19.0000
```

```
> describeBy(loanData$x60day_delinq, group = loanData$default)

 Descriptive statistics by group
group: 0
   vars    n mean  sd median trimmed mad min max range  skew kurtosis se
X1    1 7351 0.06 0.4      0       0   0   0  10    10 11.97   202.05  0
------------------------------------------------------------------------------
group: 1
   vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 52 7.92 4.89      6     7.4 4.45   2  19    17 0.78    -0.68 0.68
```

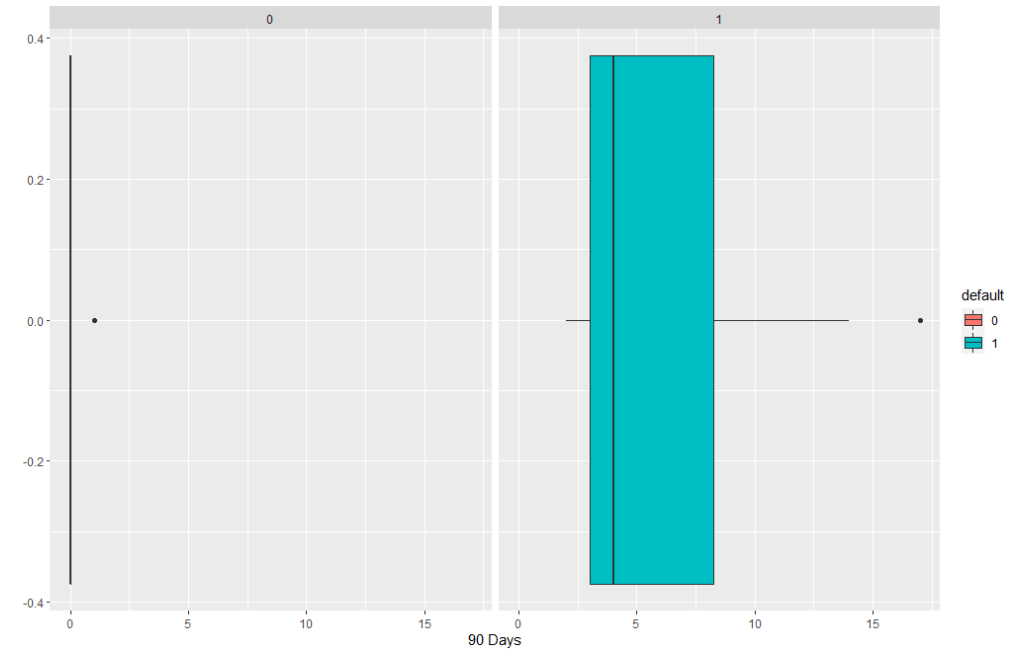


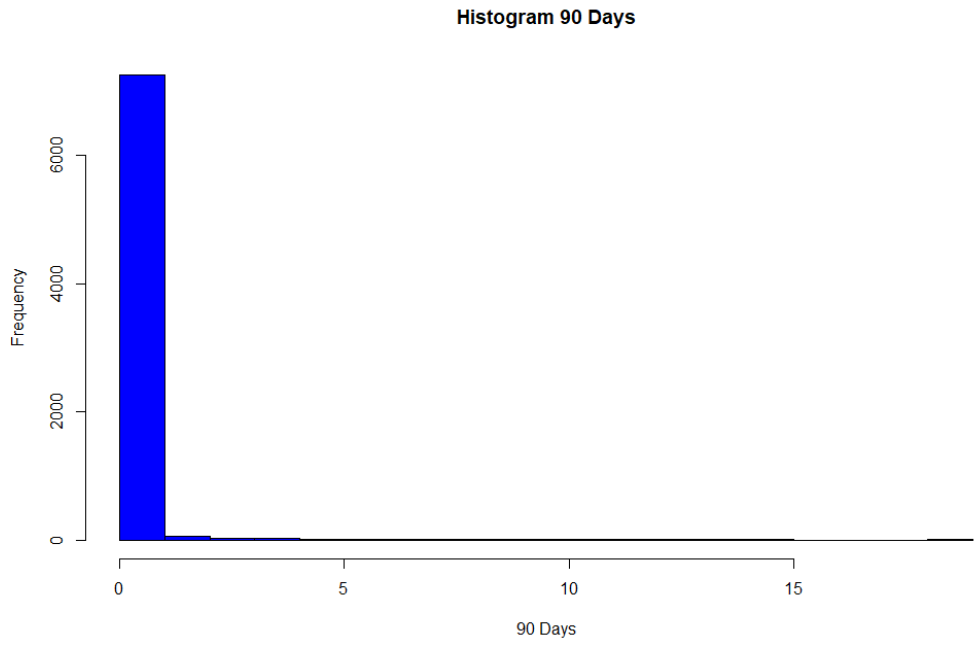


From above can be concluded that the default customers have a median of 6 occurrences where they did not make a payment in a 60-day period, while non-default customers' median is zero (0). However, there are some outliers that tell us that some of those non-default customers had an atypical payment behavior.

Appendix R – 90 Days Past Due

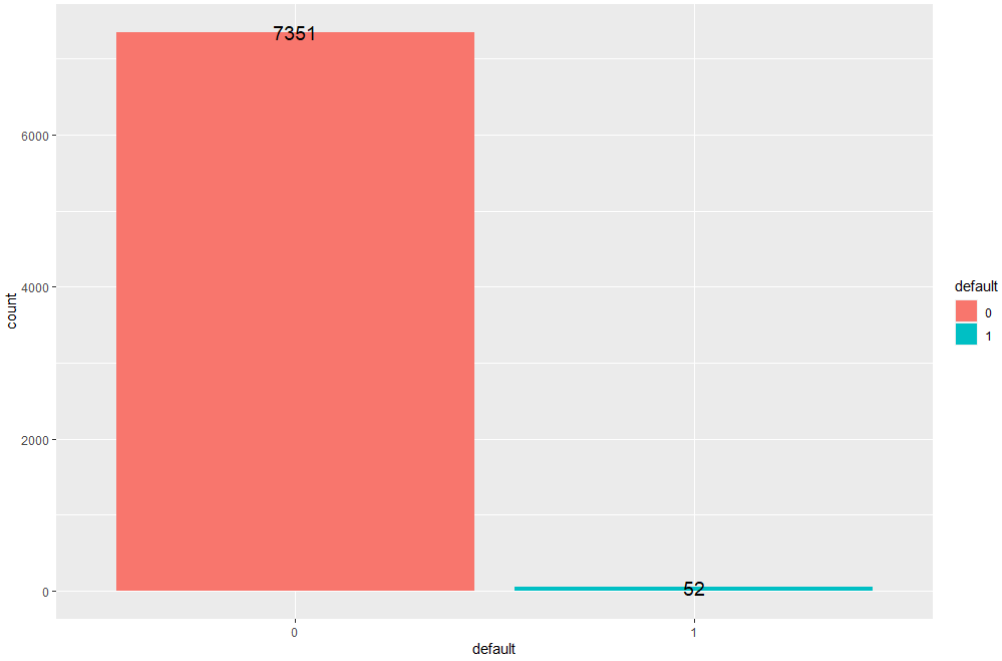
```
> summary(loanData$x90day_delinq)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
 0.00000  0.00000  0.00000  0.04593  0.00000 17.00000


 Descriptive statistics by group
group: 0
   vars    n mean   sd median trimmed mad min max range skew kurtosis se
X1    1 7351 0.01 0.07      0       0   0   0   1     1 13.8    188.4  0
------------------------------------------------------------------------------------
group: 1
   vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 52 5.81 4.08      4    5.24 2.97   2  17    15 1.02    -0.19 0.57
```





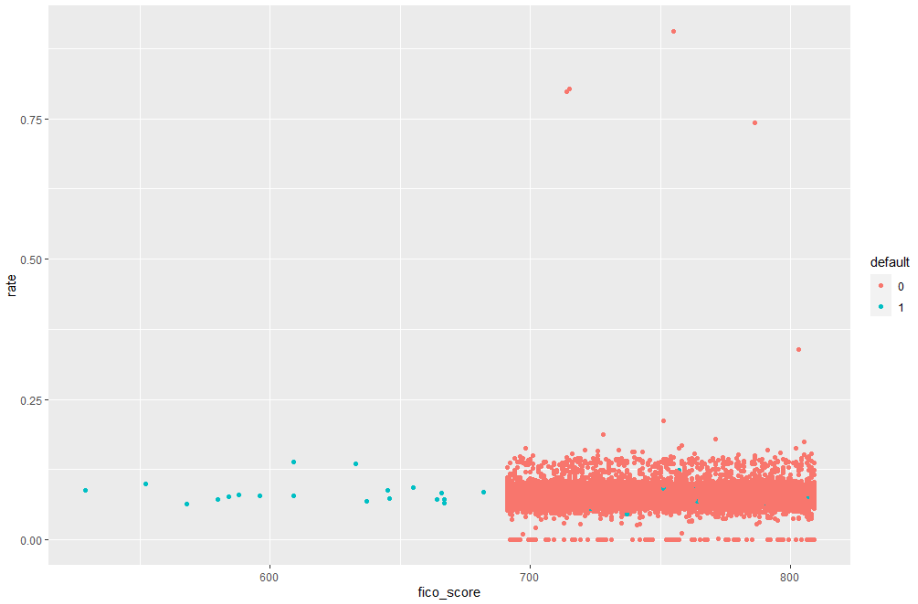
From above can be concluded that the default customers have a median of 4 occurrences where they did not make a payment in a 90-day period, while non-default customers' median is zero (0).

Appendix S – Default

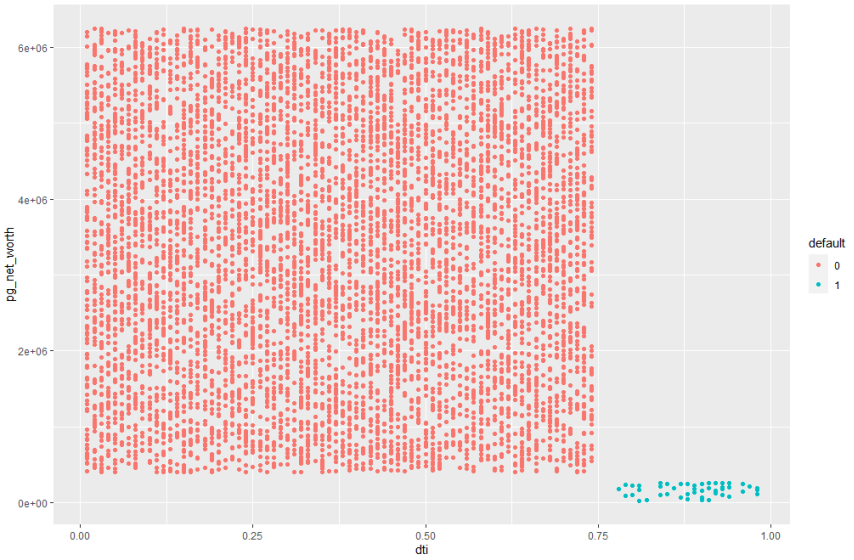


From above can be concluded that 99.30% of all customers are not in default.

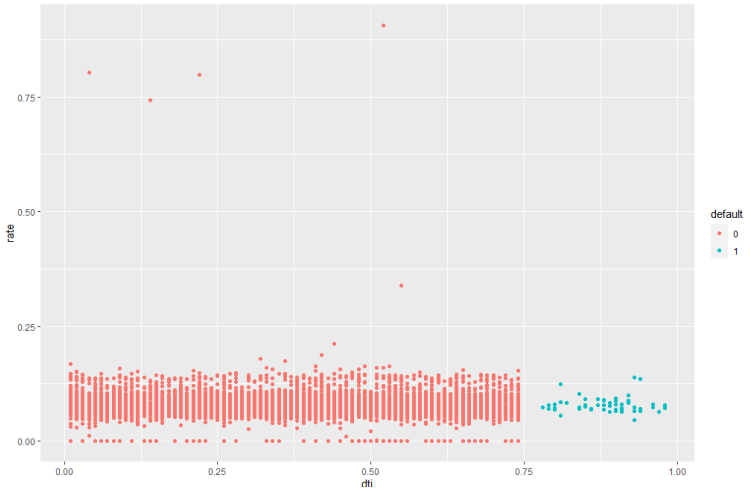
Appendix T – Bivariate Analysis.



The graphic above shows that default customers share similar rates with non-default customers even though they have lower FICO scores.



The graphic above shows us that non-default customers have a higher net worth and lower dti in contrast with default customers.



The previous graphic shows us that default customers have higher dti and share same interest rates with non-default customers.

Conclusions.

From the EDA can be concluded that non-default customers are homeowners, they have an average net worth 21 times higher than default customers; their dti is under 75%, while for default customers, it is above that percent; also, non-default customers' FICO score is on average 46 points higher than default customers' scores; finally, non-default customers on average have 0 occurrences where their payments were past due at a 30-day period, 60-day period, and 90-day period.

Appendix U – Linear Regression

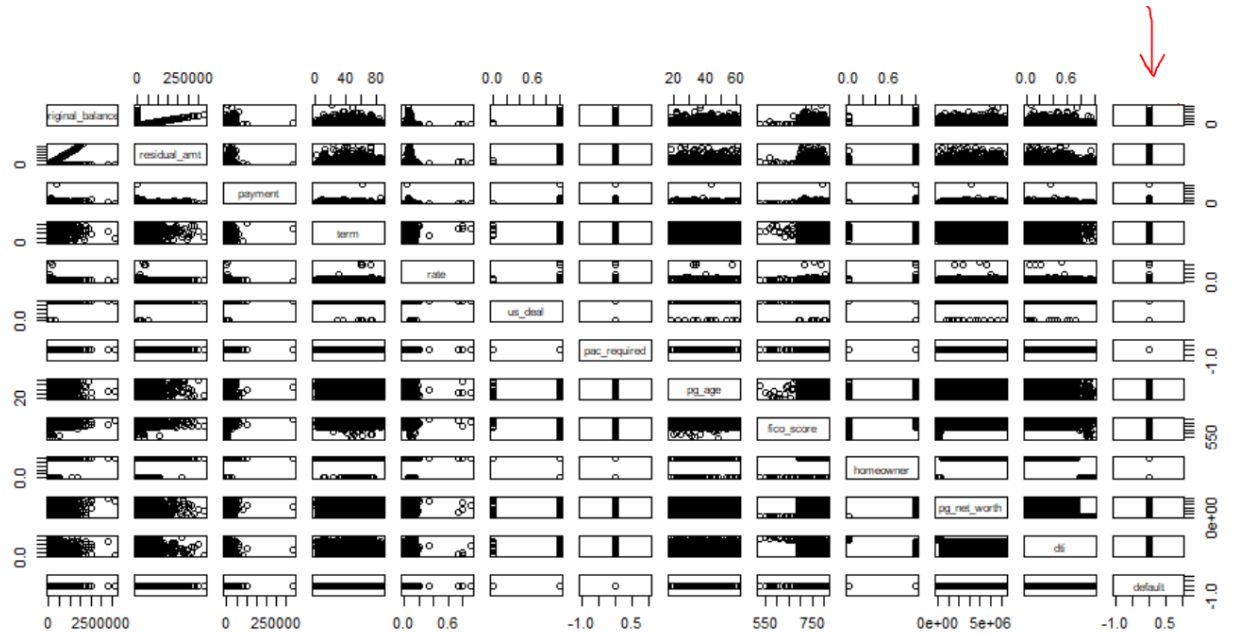
Since there was such a large focus on linear regression in this class, this was the first model we attempted with our data. Since the target of our analysis is what factors influence whether or not a customer defaults, we will still use default as our target variable. After removing the transaction id column, the summary of our multiple linear regression model is as follows:

```
> linmodel = lm(default ~ finance_product + industry + original_balance + residual_amt + payment
+               + term + rate + us_deal + pac_required + pg_age + fico_score + homeowner + pg_net_worth
+               + dti + x30day_delinq + x60day_delinq + x90day_delinq, data = loanData)
> summary(linmodel)

Call:
lm(formula = default ~ finance_product + industry + original_balance +
    residual_amt + payment + term + rate + us_deal + pac_required +
    pg_age + fico_score + homeowner + pg_net_worth + dti + x30day_delinq +
    x60day_delinq + x90day_delinq, data = loanData)

Residuals:
   Min     1Q Median     3Q    Max
     0      0      0      0      0

Coefficients: (1 not defined because of singularities)
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                      0          0      NA       NA
finance_productInstallment       0          0      NA       NA
finance_productLoan              0          0      NA       NA
finance_productTRAC              0          0      NA       NA
industryMotorcoach               0          0      NA       NA
industryOil & Gas                0          0      NA       NA
industryOther                    0          0      NA       NA
industryOTR Trucking             0          0      NA       NA
industryPetroleum                0          0      NA       NA
industryTool Truck               0          0      NA       NA
original_balance                 0          0      NA       NA
residual_amt                     0          0      NA       NA
payment                          0          0      NA       NA
term                             0          0      NA       NA
rate                             0          0      NA       NA
us_deal                          0          0      NA       NA
pac_required                    NA         NA      NA       NA
pg_age                           0          0      NA       NA
fico_score                       0          0      NA       NA
homeowner                        0          0      NA       NA
pg_net_worth                     0          0      NA       NA
dti                              0          0      NA       NA
x30day_delinq                    0          0      NA       NA
x60day_delinq                    0          0      NA       NA
x90day_delinq                    0          0      NA       NA

Residual standard error: 0 on 7379 degrees of freedom
Multiple R-squared:    NaN,     Adjusted R-squared:    NaN
F-statistic:   NaN on 23 and 7379 DF,  p-value: NA
```

Based on the summary output, linear regression might not be the best model for our data. This is since as our target variable, default, is binary. However, we still want to test the assumptions of MLR.

For the linearity assumption, we can look at a scatter plot of the x-variables vs. the default column:

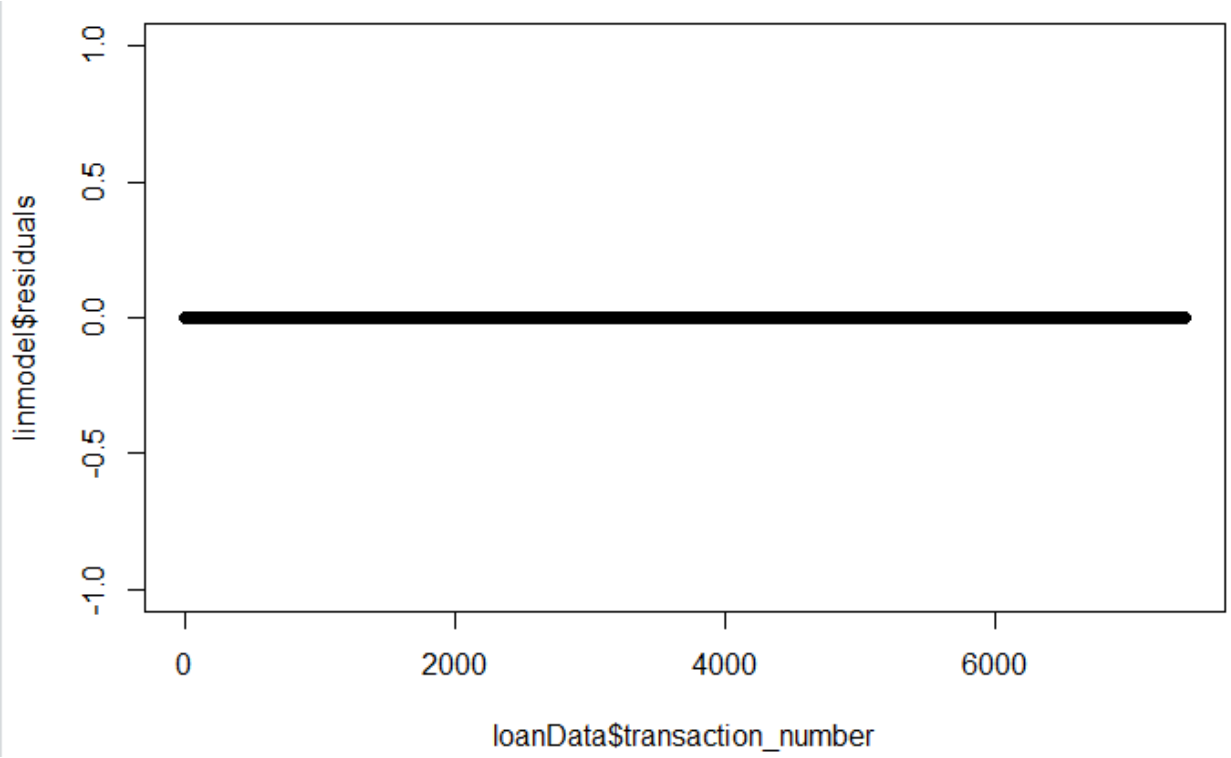
```
pairs(~original_balance + residual_amt + payment
      + term + rate + us_deal + pac_required + pg_age + fico_score + homeowner + pg_net_worth
      + dti + default, data=loanData)
```



We can see that none of the independent variables have a continuous linear relationship with the default column, and such, it does not meet the linearity assumption.

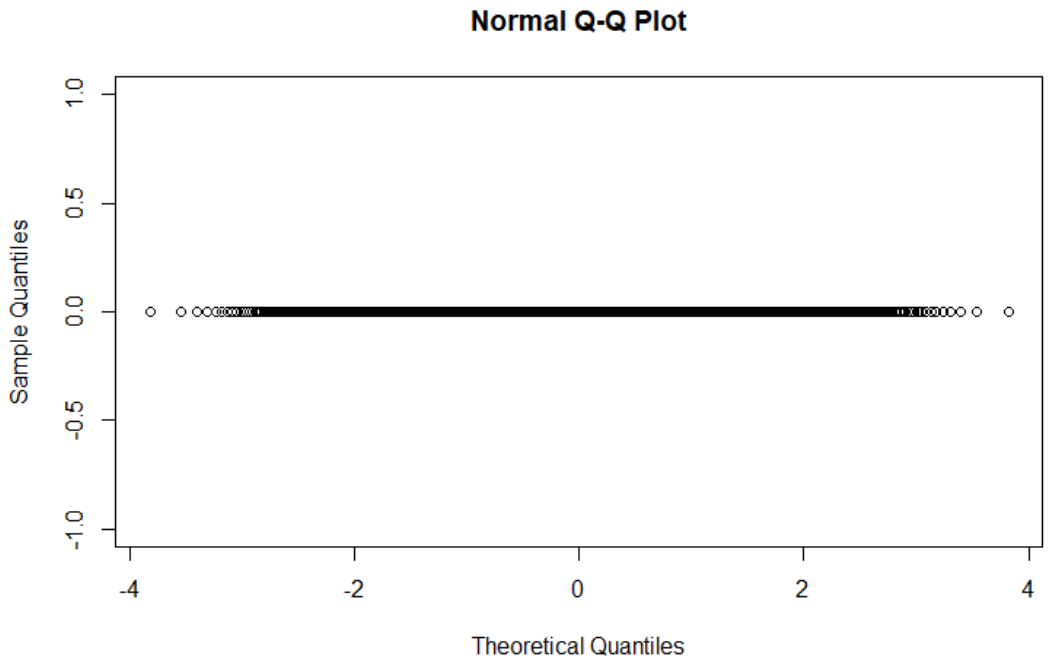
In order to test the independence of the residuals, we can use the Durbin-Watson statistic to test for autocorrelation, as well as plot the residuals over time:

```
> library(car)
> durbinWatsonTest(linmodel)
 lag Autocorrelation D-W Statistic p-value
   1             NaN           NaN      NA
 Alternative hypothesis: rho != 0
> plot(loanData$transaction_number, linmodel$residuals)
```



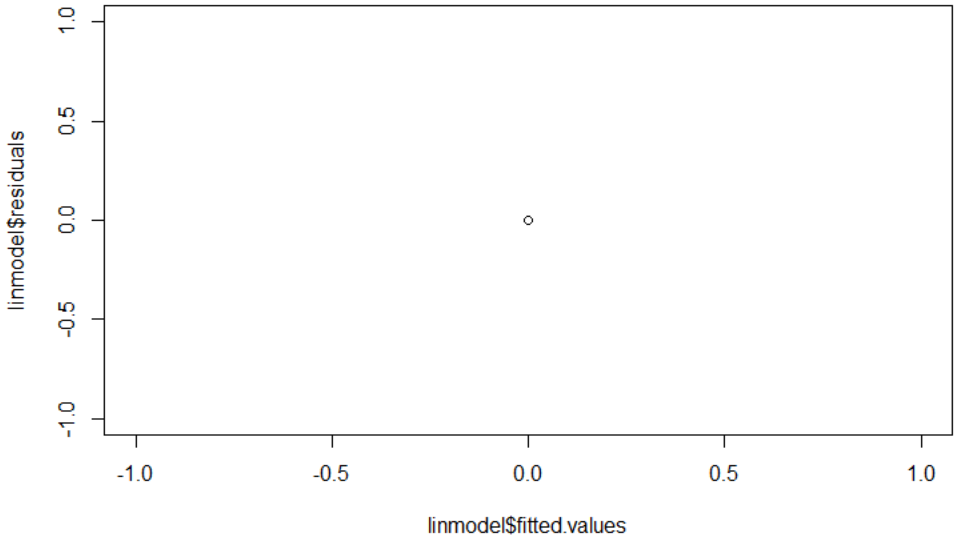
Since there was no autocorrelation for the D-W test, and the plot of residuals over time is constant at zero, this is another clue that MLR is not the best model for our data.

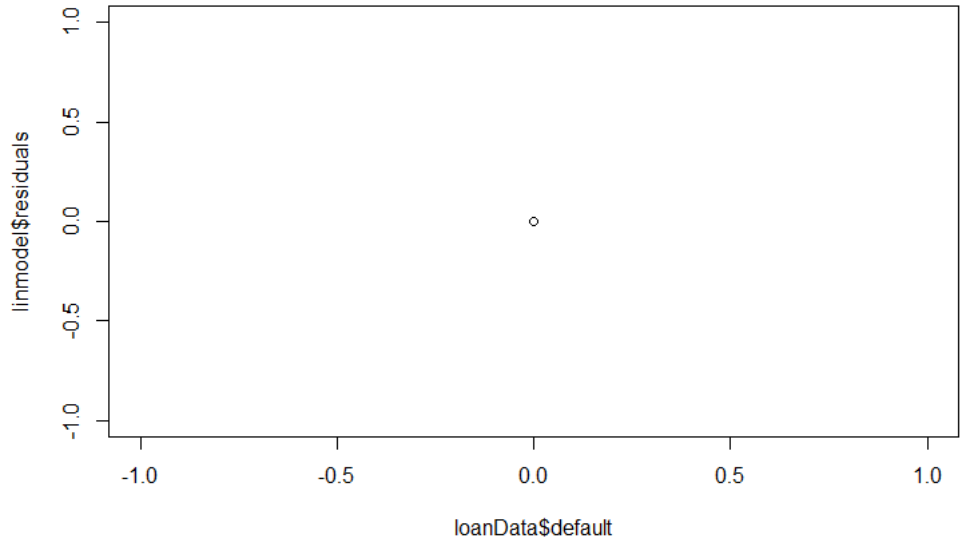
To test the normality assumption, we can look at a normal probability plot of the residuals:



Just like plotting the residuals over time, the normal probability plot shows that the error are not normal, and are constant with zero error; which violates the normality assumption of MLR.

To check the assumption of constant variance, we can plot the residuals against the predicted values, as well as plotting the residuals against the x-values:





As we can tell from both of these plots, the assumption of constant variance is not satisfied.

Finally, we need to test that there is no multicollinearity among the independent variables. To do this, we can look at the VIF:

```
> vif(linmodel)
Error in vif.default(linmodel) :
  there are aliased coefficients in the model
```

The output suggests that we have run into perfect multicollinearity, and the model does not meet this assumption.

Since all the assumptions were not satisfied, and all our beta coefficients were zero, instead of attempting to correct the assumptions, we will move on and try Logistic Regression, since our target variable is binary.

Appendix V – Logistic Regression.

The first thing we did prior to modeling the data with logistic regression was to ensure our categorical variables were treated properly. We set the "finance product" and "industry" columns as factors, so when the regression algorithm was run, R treats them as indicator variables with k-1 columns; with k being the number of unique categories.

The first model we ran was a generalized logistic regression model with all the variables except for "transaction_number" included as predictor variables. However, there was a warning message thrown: "glm.fit: algorithm did not converge". When looking at the summary, the p-values for all variables were exactly 1.

```
> model1 <- glm(default ~ finance_product + industry + original_balance + residual_amt + payment
+               + term + rate + us_deal + pac_required + pg_age + fico_score + homeowner + pg_net_worth
+               + dti + x30day_delinq + x60day_delinq + x90day_delinq, data = loanData, family = "binomial")
warning message:
glm.fit: algorithm did not converge
```

```
> summary(model1)

Call:
glm(formula = default ~ finance_product + industry + original_balance +
    residual_amt + payment + term + rate + us_deal + pac_required +
    pg_age + fico_score + homeowner + pg_net_worth + dti + x30day_delinq +
    x60day_delinq + x90day_delinq, family = "binomial", data = loanData)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-2.409e-06  -2.409e-06  -2.409e-06  -2.409e-06  -2.409e-06

Coefficients: (1 not defined because of singularities)
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -2.657e+01  2.019e+05       0        1
finance_productInstallment  2.267e-13  1.153e+05       0        1
finance_productLoan         1.756e-13  1.141e+05       0        1
finance_productTRAC         4.687e-13  1.148e+05       0        1
industryMotorcoach          1.131e-13  3.967e+04       0        1
industryOil & Gas          -1.686e-13  3.112e+04       0        1
industryOther              -6.244e-14  2.959e+04       0        1
industryOTR Trucking        1.972e-13  2.902e+04       0        1
industryPetroleum          -1.432e-13  3.101e+04       0        1
industryTool Truck         -2.484e-13  2.977e+04       0        1
original_balance            1.953e-19  3.704e-02       0        1
residual_amt               -4.229e-18  2.602e-01       0        1
payment                    -2.762e-18  8.478e-01       0        1
term                       -9.574e-15  2.216e+02       0        1
rate                        3.430e-12  1.640e+05       0        1
us_deal                     5.607e-14  9.906e+04       0        1
pac_required                       NA         NA      NA       NA
pg_age                      7.441e-15  3.292e+02       0        1
fico_score                 -1.597e-15  1.207e+02       0        1
homeowner                   3.163e-13  8.903e+04       0        1
pg_net_worth               -9.675e-20  2.483e-03       0        1
dti                         4.415e-13  1.959e+04       0        1
x30day_delinq              -1.881e-14  2.183e+03       0        1
x60day_delinq              -4.823e-15  1.137e+04       0        1
x90day_delinq              -2.063e-14  1.588e+04       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 0.0000e+00  on 7402  degrees of freedom
Residual deviance: 4.2949e-08  on 7379  degrees of freedom
AIC: 48
```

After researching this output, we suspect our data might have perfect/complete separation. Complete separation happens when the outcome variable separates a predictor variable, or combination of predictor variables completely (Introduction to SAS). Looking back at our EDA, we believe this is happening for two reasons.

The first is that there is a small number of occurrences of defaults compared to the number of predictor variables. One rule of thumb we found for fitting a logistic regression model was to have one degree of freedom (predictor variable) for every 10 occurrences (RajeshS). Since there are only 52 defaults in our dataset and we have 17 predictor variables (not including "Transaction Number"), our ratio is 0.32 occurrences:1 predictor variable. The second reason we believe this is happening, is that there are columns with a perfect event status split for defaults vs. non-defaults. All defaults had a value of at least one in the "90DayDelinq" column. In addition, all defaults had a zero value for "Homeownership". Compare this to non-defaulted contracts where all of them were homeowners and there were zero values for the 90DayDelinq column, and it's clear we're probably dealing with complete separation.

In order to handle this problem, we used George Heinze's penalized likelihood model. (Heinze, G., & Schemper, M.) After removing the columns with perfect separation, and then performing our penalized regression in a stepwise manner, the final model output is as follows:

```
> library(logistf)
> model6 = logistf(default ~ original_balance + residual_amt + payment + pg_net_worth, data=loanData)
> summary(model6)
logistf(formula = default ~ original_balance + residual_amt +
    payment + pg_net_worth, data = loanData)

Model fitted by Penalized ML
Confidence intervals and p-values by Profile Likelihood Profile Likelihood Profile Likelihood Profile
d

                        coef     se(coef)    lower 0.95    upper 0.95        Chisq            p
(Intercept)      -9.751622e+00 2.036361e+00 -2.379699e+01 -5.823811e+00 22.337062300 2.287489e-06
original_balance  2.643240e-06 8.447062e-07 -4.837739e-08  6.899920e-06  3.782265344 5.179849e-02
residual_amt      9.837718e-06 7.091201e-06 -2.286943e-05  3.446289e-05  1.005702430 3.159346e-01
payment           2.362481e-05 7.869452e-06 -8.974476e-06  4.859712e-05  3.565660494 5.898627e-02
pg_net_worth     -5.955196e-08 4.940190e-07 -2.616650e-06  2.916435e-06  0.004814393 9.446825e-01

Likelihood ratio test=11.65029 on 4 df, p=0.02015044, n=7403
Wald test = 19.00205 on 4 df, p = 0.0007852171

Covariance-Matrix:
              [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  4.146768e+00 -1.768134e-07 -8.034352e-06 -5.114901e-06 -7.698864e-07
[2,] -1.768134e-07  7.135286e-13 -1.210380e-12  8.117675e-13 -1.517183e-13
[3,] -8.034352e-06 -1.210380e-12  5.028513e-11  1.240912e-11  1.362854e-12
[4,] -5.114901e-06  8.117675e-13  1.240912e-11  6.192828e-11  1.250642e-13
[5,] -7.698864e-07 -1.517183e-13  1.362854e-12  1.250642e-13  2.440548e-13
```

From the output we can see that of the variables with no perfect separation, the original balance, residual amount, payment and net worth of the guarantor provided the best penalized model. However, we must ask ourselves, does removing variables that perfectly explain whether a customer defaults make sense in our application?

According to the *"Assumptions of Logistic Regression"* article, Logistic Regression does not share the same assumptions and Linear Regression. Instead, the assumptions of Logistic Regression are:

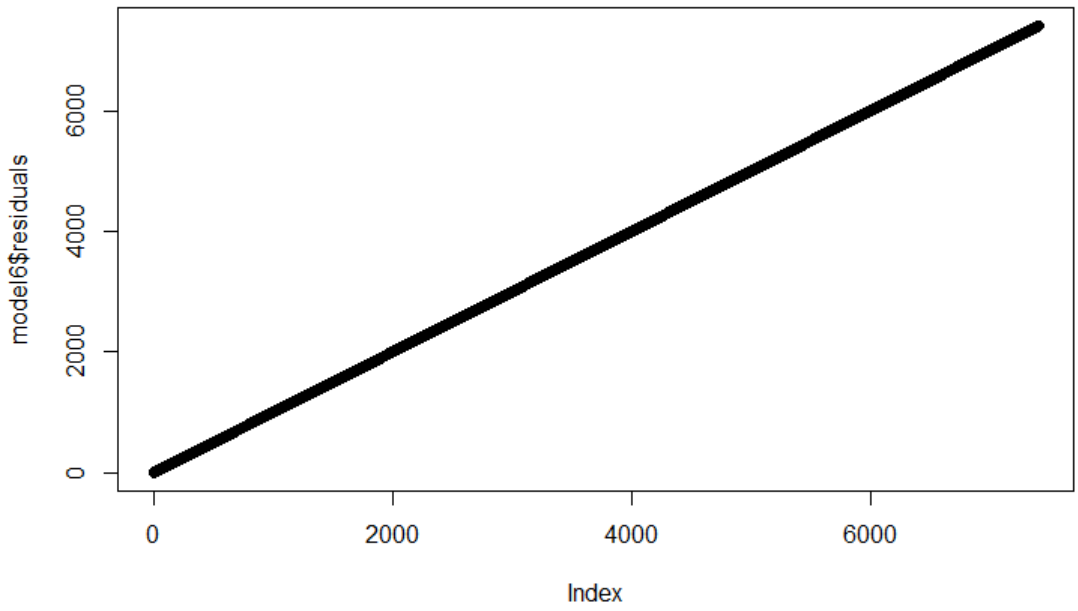
- "Binary logistic regression requires the dependent variable to be binary and ordinal logistic regression requires the dependent variable to be ordinal. Reducing an ordinal or even metric variable to dichotomous level loses a lot of information, which makes this test inferior compared to ordinal logistic regression in these cases.
- Secondly, since logistic regression assumes that P(Y=1) is the probability of the event occurring, it is necessary that the dependent variable is coded accordingly. That is, for a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Thirdly, the model should be fitted correctly. Neither over fitting nor under fitting should occur. That is only the meaningful variables should be included, but also all meaningful variables should be included. A good approach to ensure this is to use a stepwise method to estimate the logistic regression.

- Fourthly, the error terms need to be independent. Logistic regression requires each observation to be independent. That is that the data-points should not be from any dependent samples design, e.g., before-after measurements, or matched pairings. Also the model should have little or no multicollinearity. That is that the independent variables should be independent from each other. However, there is the option to include interaction effects of categorical variables in the analysis and the model. If multicollinearity is present centering the variables might resolve the issue, i.e. deducting the mean of each variable. If this does not lower the multicollinearity, a factor analysis with orthogonally rotated factors should be done before the logistic regression is estimated.
- Fifthly, logistic regression assumes linearity of independent variables and log odds. Whilst it does not require the dependent and independent variables to be related linearly, it requires that the independent variables are linearly related to the log odds. Otherwise the test underestimates the strength of the relationship and rejects the relationship too easily, that is being not significant (not rejecting the null hypothesis) where it should be significant. A solution to this problem is the categorization of the independent variables. That is transforming metric variables to ordinal level and then including them in the model. Another approach would be to use discriminant analysis, if the assumptions of homoscedasticity, multivariate normality, and absence of multicollinearity are met.
- Lastly, it requires quite large sample sizes. Because maximum likelihood estimates are less powerful than ordinary least squares (e.g., simple linear regression, multiple linear regression); whilst OLS needs 5 cases per independent variable in the analysis, ML needs at least 10 cases per independent variable, some statisticians recommend at least 30 cases for each parameter to be estimated." (REFERENCE)

We've proved that the first, second and third assumptions have been met through our EDA and the fact that our target variable is binary. So, we need to focus on the third, fourth and fifth assumptions.

In order to test whether or not the error terms of our penalized model are independent, we can plot the residuals over time:

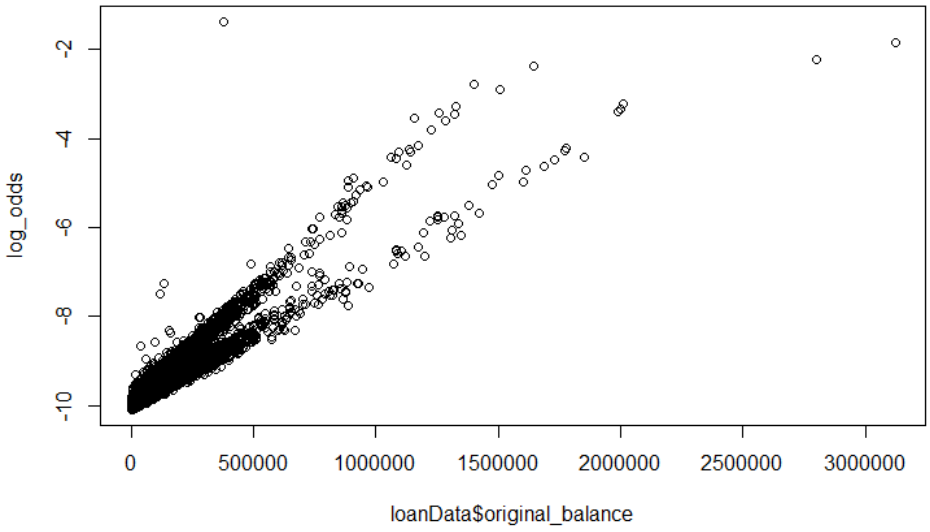
```
> plot(loanData$transaction_number, model6$residuals)
```



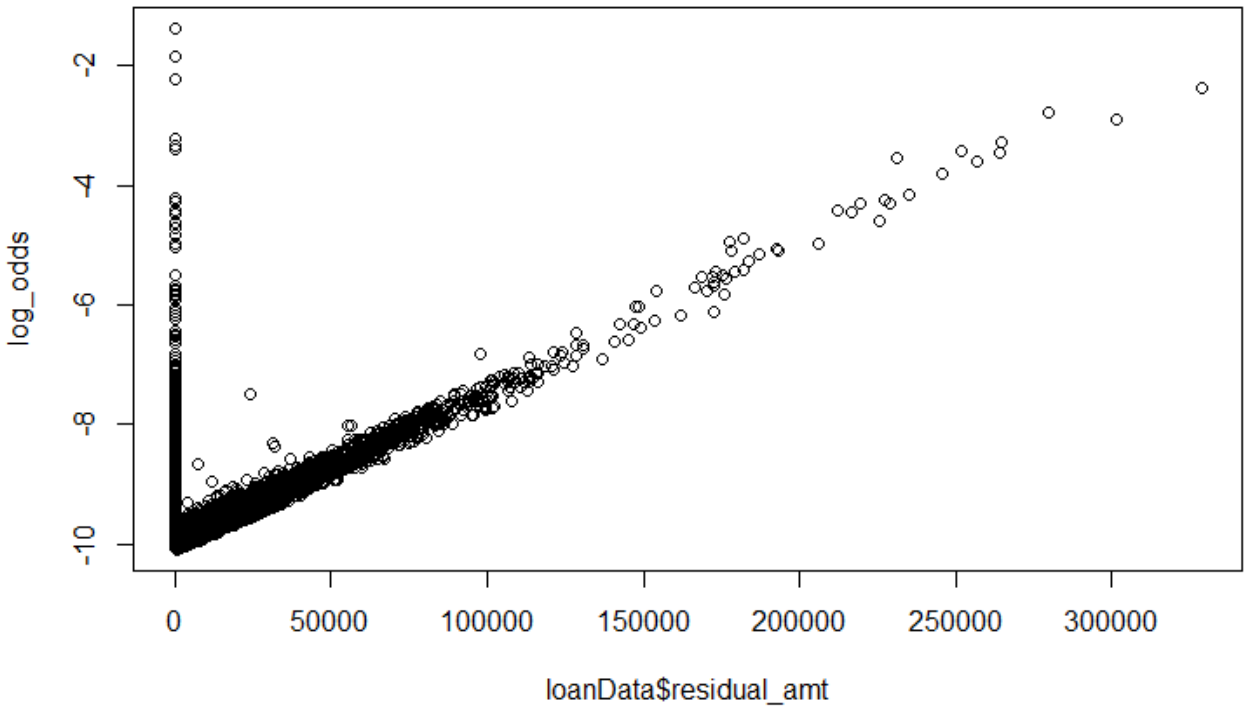
We can see that the errors are perfectly linear, and do show a constant pattern. So, this assumption is not met.

In order to test the linearity of independent variables and log odds, we can plot them against each other:

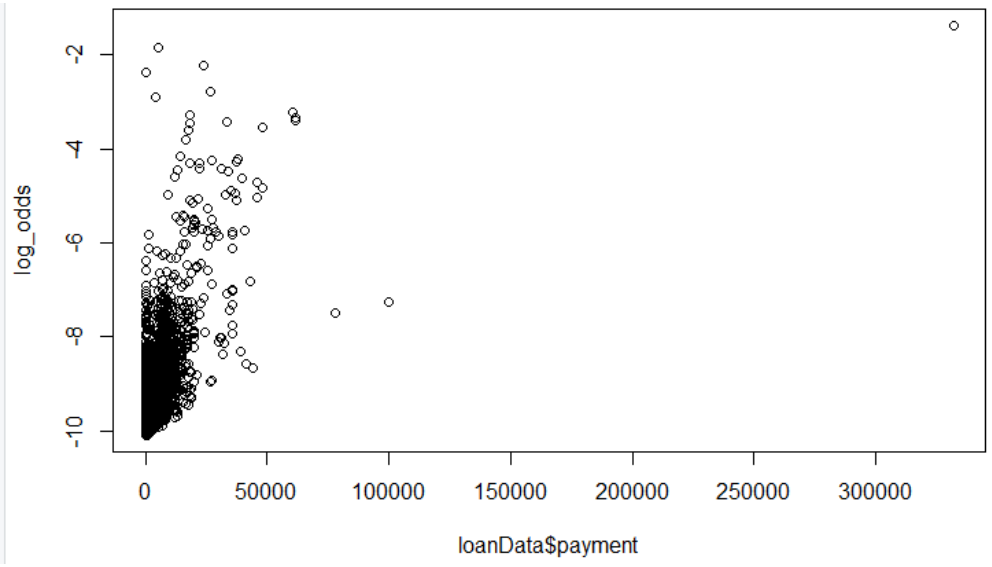
```
> odds = model6$predict
> log_odds = log(odds)
> plot(loanData$original_balance, log_odds)
```



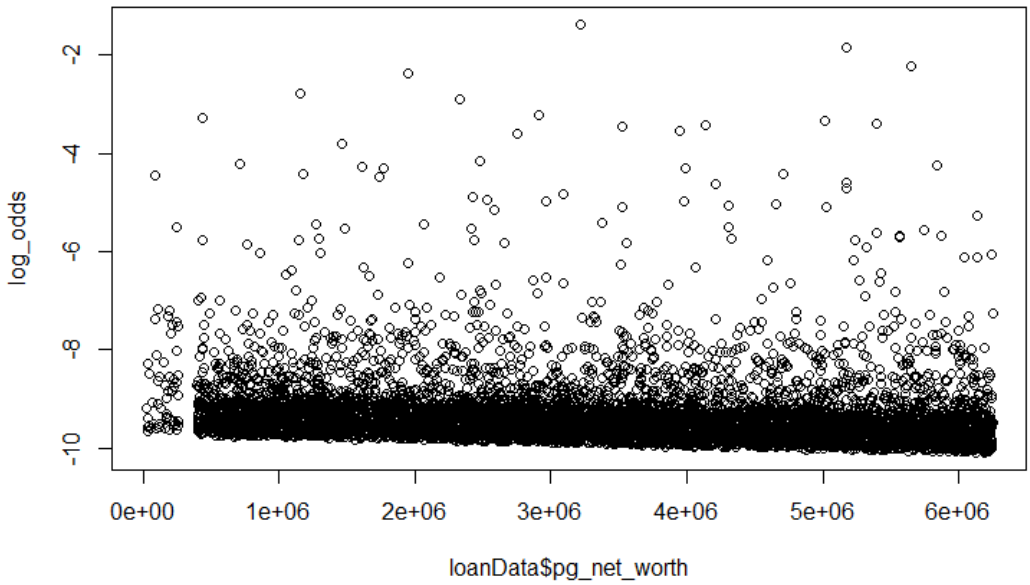

```
> plot(loanData$residual_amt, log_odds)
```




```
> plot(loanData$payment,log_odds)
```



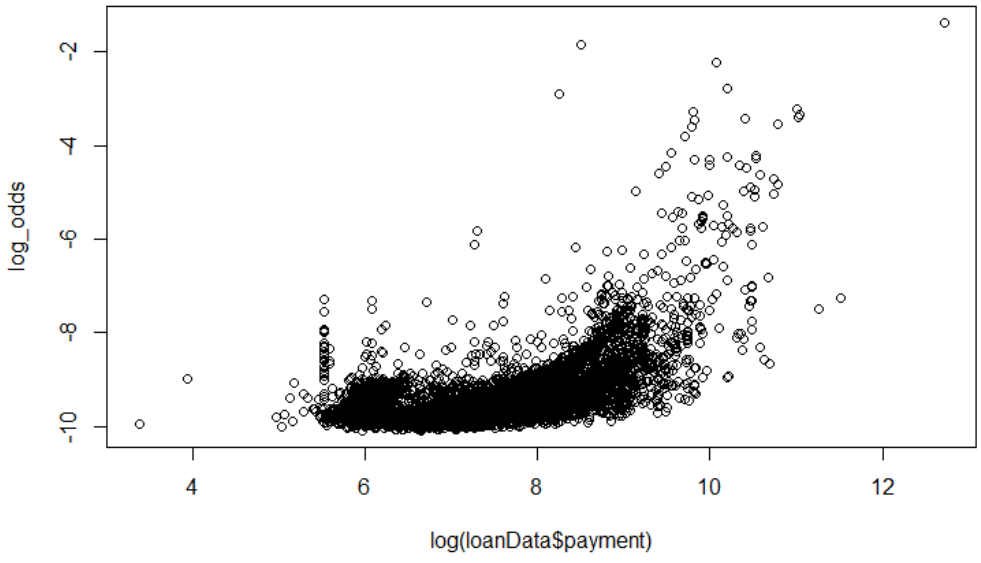

```
> plot(loanData$pg_net_worth, log_odds)
```



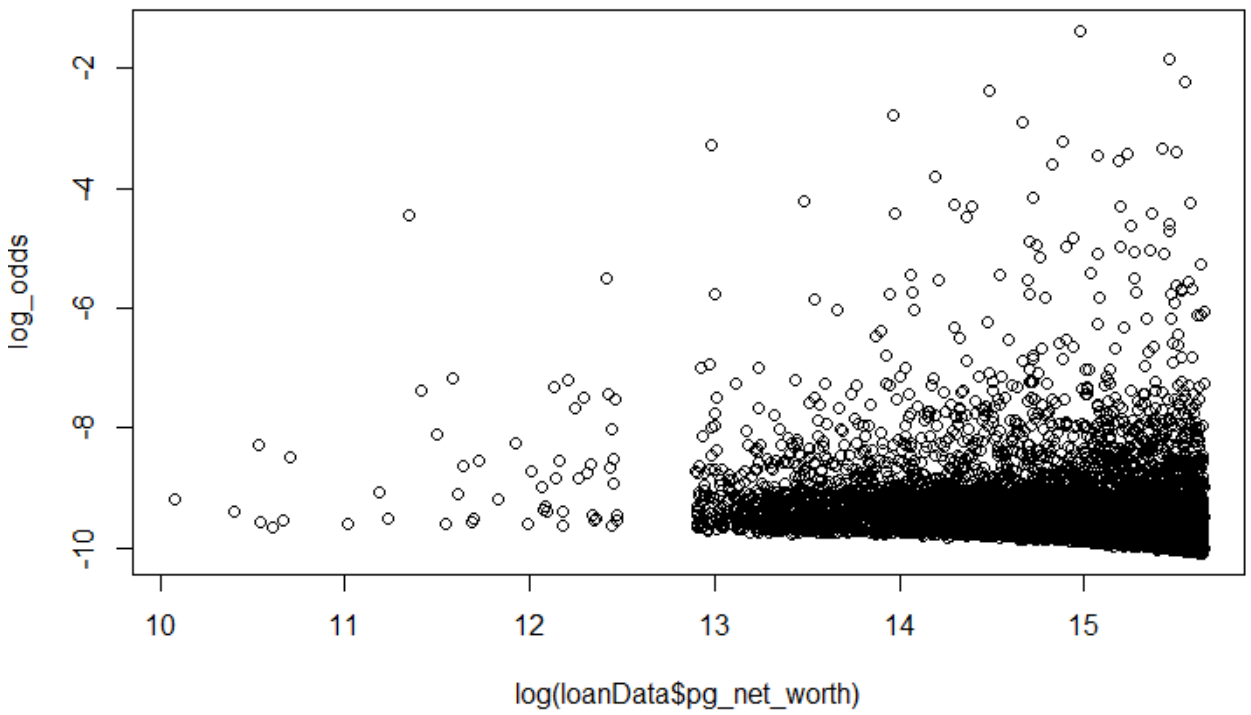
We can see from these graphs that the original balance and residual amount columns have a linear relationship with the log odds, however the payment and net worth columns do not.

To try and fix this, we can take the natural log of those columns to see if it helps:

```
plot(log(loanData$payment),log_odds)
```




```
> plot(log(loanData$pg_net_worth), log_odds)
```



Looking at these graphs, we can see that the natural log transformation of the independent variables somewhat helped the assumption; but not in a straight linear way.

Finally, we need to check the assumption of large sample sizes. As mentioned previously when discussing the need for a penalized model, since we only have a small number of default occurrences (52 out of 7403), it is not enough to meet this assumption.

Ultimately, based on our research and previous analysis, we'd recommend that the company continue utilizing their current credit policy, and reject any applicants who are not homeowners. However, we do not recommend using a logistic regression model due to the complete separation and not meeting the assumptions; and instead recommend pursuing other machine learning classification models to more accurately predict whether a potential customer is going to default.

Appendix W – Other Classification Models.

In order to successfully predict customers' payment behavior, 2 additional different machine learning classification models have been implemented: k-nearest neighbor and Random Forest Classification.

K-Nearest Neighbor.

The K-Nearest Neighbor (KNN) algorithm has been implemented to classify default and non-default customers based on the variables: dti, FICO score, homeowner, and net worth. Those variables were selected because they provide important information from potential customers before the company lends them money in order for the credit department to predict if the customers will default or not in the future.

It was decided to use 50% of the data points to train the model and use the 50% left to test it. The reason behind this decision is because there are just a few non-default data points, which get equally distributed between testing and training when portioning the data set 50/50. The variables used to train the model were normalized because the range among all of them was extremely broad.

K values of 1, 5, and 10 were used to see how they perform in terms of correct proportion of classification.

```
> CrossTable(x=test.def,y=knn.5,prop.chisq = FALSE)

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  3703

             | knn.5
    test.def |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |      3678 |         0 |      3678 |
             |     1.000 |     0.000 |     0.993 |
             |     1.000 |     0.000 |           |
             |     0.993 |     0.000 |           |
-------------|-----------|-----------|-----------|
           1 |         0 |        25 |        25 |
             |     0.000 |     1.000 |     0.007 |
             |     0.000 |     1.000 |           |
             |     0.000 |     0.007 |           |
-------------|-----------|-----------|-----------|
Column Total |      3678 |        25 |      3703 |
             |     0.993 |     0.007 |           |
-------------|-----------|-----------|-----------|
```

The test data consisted of 3703 data points. Out of which 3678 have been accurately predicted as non-default in nature which constitutes 99.3%. Also, 25 out of 3703 observations were accurately predicted as default in nature which constitutes 0.7%.

There were no cases of false negatives, meaning no data points were recorded which actually were default in nature but got predicted as non-default. There were also no cases of false positives meaning no data points that were non-default in nature got predicted as default.

The total accuracy of the model is 100 %. Same results were obtained for values of k of 1 and 10.

Cross Validation.

Leave one out cross validation is used to measure the accuracy of the model. For each row of the data set, the k nearest (in Euclidean distance) other date set points are found, and the classification is decided by majority vote, with ties broken at random. If there are ties for the kth nearest point, all candidates are included in the vote.

It has been decided to use a k = 5.

```
> n=knn.cv(loanData2.subset,loanData2$default, k=5)

> sum(loanData2$default == n)/7403
[1] 1
```

From above can be concluded that the model has a 100% accuracy.

From the EDA, we conclude that no customers in default are homeowners. The KNN algorithm was implemented excluding the homeowner variable to analyze the accuracy of the model under that scenario. K values of 1, 5, and 10 were used.

K = 1

```
> CrossTable(x=test.def,y=knn.1,prop.chisq = FALSE)

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  3703

             | knn.1
    test.def |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |      3678 |         0 |      3678 |
             |     1.000 |     0.000 |     0.993 |
             |     1.000 |     0.000 |           |
             |     0.993 |     0.000 |           |
-------------|-----------|-----------|-----------|
           1 |         0 |        25 |        25 |
             |     0.000 |     1.000 |     0.007 |
             |     0.000 |     1.000 |           |
             |     0.000 |     0.007 |           |
-------------|-----------|-----------|-----------|
Column Total |      3678 |        25 |      3703 |
             |     0.993 |     0.007 |           |
-------------|-----------|-----------|-----------|
```

The test data consisted of 3703 data points. Out of which 3678 have been accurately predicted as non-default in nature which constitutes 99.3%. Also, 25 out of 3703 observations were accurately predicted as default in nature which constitutes 0.7%.

There were no cases of false negatives, meaning no data points were recorded which actually were default in nature but got predicted as non-default. There were also no cases of false positives meaning no data points that were non-default in nature got predicted as default.

The total accuracy of the model is 100 %.

K = 5

```
> CrossTable(x=test.def,y=knn.5,prop.chisq = FALSE)

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  3703

             | knn.5
    test.def |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |      3678 |         0 |      3678 |
             |     1.000 |     0.000 |     0.993 |
             |     0.999 |     0.000 |           |
             |     0.993 |     0.000 |           |
-------------|-----------|-----------|-----------|
           1 |         3 |        22 |        25 |
             |     0.120 |     0.880 |     0.007 |
             |     0.001 |     1.000 |           |
             |     0.001 |     0.006 |           |
-------------|-----------|-----------|-----------|
Column Total |      3681 |        22 |      3703 |
             |     0.994 |     0.006 |           |
-------------|-----------|-----------|-----------|
```

The test data consisted of 3703 data points. Out of which 3678 have been accurately predicted as non-default in nature which constitutes 99.3%. Also, 22 out of 3703 observations were accurately predicted as default in nature which constitutes 0.6%.

There were 3 cases of false negatives, meaning data points were recorded which actually were default in nature but got predicted as non-default. There were also no cases of false positives meaning no data points that were non-default in nature got predicted as default.

The total accuracy of the model is 99.99%.

K = 10

```
> CrossTable(x=test.def,y=knn.10,prop.chisq = FALSE)

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  3703

             | knn.10
    test.def |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |      3678 |         0 |      3678 |
             |     1.000 |     0.000 |     0.993 |
             |     0.999 |     0.000 |           |
             |     0.993 |     0.000 |           |
-------------|-----------|-----------|-----------|
           1 |         5 |        20 |        25 |
             |     0.200 |     0.800 |     0.007 |
             |     0.001 |     1.000 |           |
             |     0.001 |     0.005 |           |
-------------|-----------|-----------|-----------|
Column Total |      3683 |        20 |      3703 |
             |     0.995 |     0.005 |           |
-------------|-----------|-----------|-----------|
```

The test data consisted of 3703 data points. Out of which 3678 have been accurately predicted as non-default in nature which constitutes 99.3%. Also, 20 out of 3703 observations were accurately predicted as default in nature which constitutes 0.5%.

There were 5 cases of false negatives, meaning data points were recorded which actually were default in nature but got predicted as non-default. There were also no cases of false positives meaning no data points that were non-default in nature got predicted as default.

The total accuracy of the model is 99.86%.

Cross Validation

Leave one out cross validation is used to measure the accuracy of the model.

It has been decided to use k = 5.

```
> n=knn.cv(loanData2.subset,loanData2$default, k=5)
> sum(loanData2$default == n)/7403
[1] 0.9997298
```

We can conclude that after excluding homeowner variable of the model, it still has a good accuracy, 99.97%.

Based on the previous results, the KNN algorithm is a good machine learning model to predict customers' payment behavior.

Random Forest Classification

A random forest model is run to classify customers based on their default status. In our data - set, we consider the "default" column to be our response variable. Columns "transaction number" and "pac required" are omitted since they do not contribute towards the model. The categorical variables are changed to become factors.

```
> str(selData)
tibble [7,403 × 17] (S3: tbl_df/tbl/data.frame)
 $ finance_product : Factor w/ 4 levels "FMV","Installment",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ industry        : Factor w/ 7 levels "Hydrovac - Non O&G",..: 5 5 5 5 5 5 5 5 5 6 ...
 $ original_balance: num [1:7403] 9313 158234 167781 93960 120909 ...
 $ residual_amt    : num [1:7403] 1863 31647 33556 18792 24182 ...
 $ payment         : num [1:7403] 1035 2221 2380 2610 1679 ...
 $ term            : num [1:7403] 9 61 63 38 36 38 38 38 30 46 ...
 $ rate            : num [1:7403] 0.1 0.0524 0.0528 0.0667 0.0662 0.0661 0.0653 0.0653 0.0681 0.0755 ...
 $ us_deal         : num [1:7403] 1 1 1 1 1 1 1 1 1 1 ...
 $ pg_age          : num [1:7403] 52 43 19 61 31 57 59 52 21 36 ...
 $ fico_score      : num [1:7403] 730 729 784 794 729 704 769 720 714 757 ...
 $ homeowner       : num [1:7403] 1 1 1 1 1 1 1 1 1 1 ...
 $ pg_net_worth    : num [1:7403] 558291 2818876 1302722 1284320 2624638 ...
 $ dti             : num [1:7403] 0.58 0.28 0.45 0.08 0.25 0.14 0.1 0.4 0.5 0.42 ...
 $ x30day_delinq   : num [1:7403] 0 0 0 0 1 0 0 0 0 0 ...
 $ x60day_delinq   : num [1:7403] 0 0 0 0 0 0 0 0 0 0 ...
 $ x90day_delinq   : num [1:7403] 0 0 0 0 0 0 0 0 0 0 ...
 $ default         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

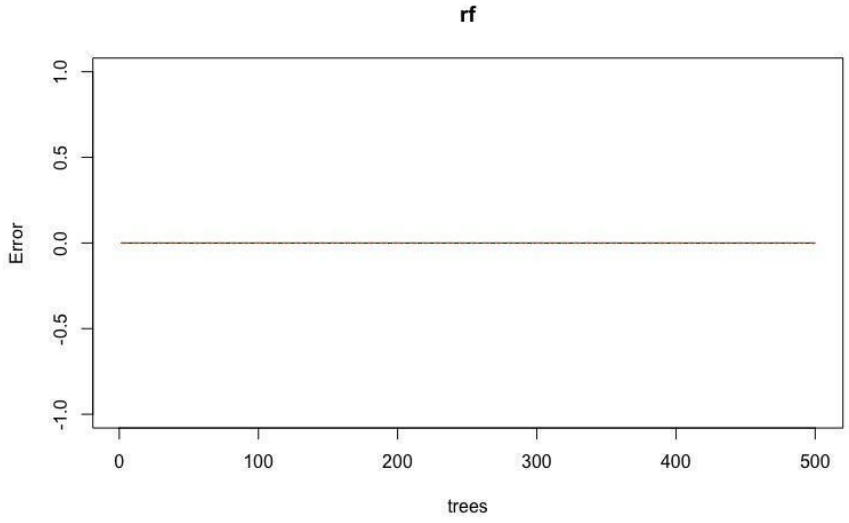
An 80-20 split is done on the data with 80% to be the training set and 20% to be the testing set. The random forest classification model is then run on the training set, results of which are shown below.

```
> rf = randomForest(default ~ ., data = train)
> print(rf)

Call:
 randomForest(formula = default ~ ., data = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 0%
Confusion matrix:
     0  1 class.error
0 5884  0           0
1    0 42           0
```

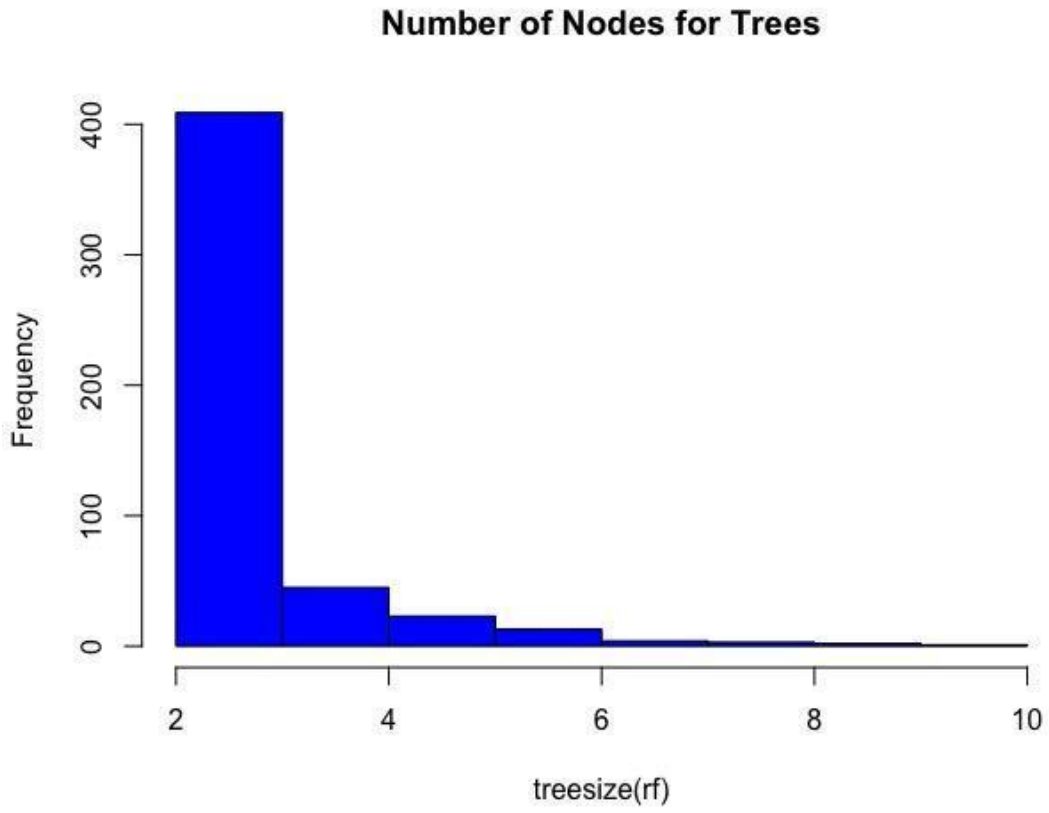
We notice that the model split into 500 classification trees with 4 variables tested at each tree. The majority of votes received to each classification is then given as the final result. The confusion matrix indicates that the model correctly predicted 5884 times if a customer had not defaulted and 42 times in which the customer had defaulted. There is 0 error on these predictions.

```
> confusionMatrix(p2, test$default)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1467    0
         1    0   10

               Accuracy : 1
                 95% CI : (0.9975, 1)
    No Information Rate : 0.9932
    P-Value [Acc > NIR] : 4.388e-05
```

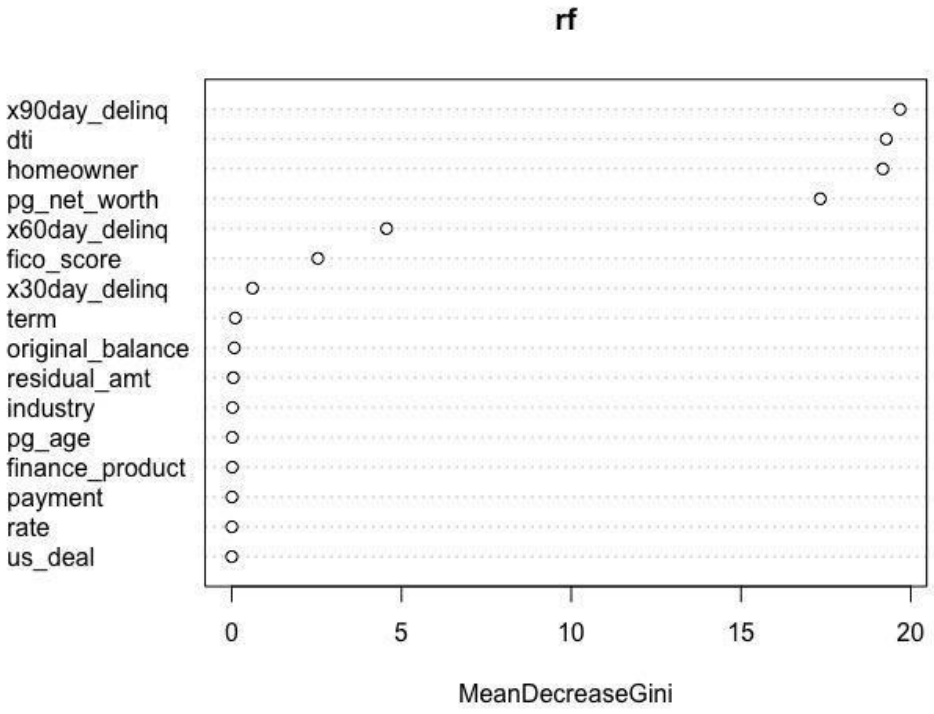
A look at the predictions performed on the test set also gives an accuracy of 1 within 95% confidence interval. This proves that our predictions on the training set are accurate. A plot of the error rate below confirms the accuracy of the classification model and the consistency of the data.



There is no error at any number of trees which tells us that all the predictions were accurate. The random forest model is not tuned any further since we already achieved an accuracy of 1.



A look at the number of nodes for trees tells us that out of around 400 trees had 2 nodes while there were also trees which had up to 8 nodes.




```
> importance (rf)
                  MeanDecreaseGini
finance_product       0.015169705
industry              0.022752970
original_balance      0.074760850
residual_amt          0.043652103
payment               0.006493172
term                  0.104701457
rate                  0.005001434
us_deal               0.000000000
pg_age                0.015693701
fico_score            2.536858114
homeowner            19.177984618
pg_net_worth         17.333694739
dti                  19.277389821
x30day_delinq         0.611785494
x60day_delinq         4.560463061
x90day_delinq        19.683355091
```

The above plots indicate the GINI importance for our model. GINI importance measures the average gain of purity by splits of a given variable. If the variable is useful, it tends to split mixed labeled nodes into pure single class nodes. A higher decrease in GINI means that a particular

predictor variable plays a more prominent role in data partitioning into our required classes which here is default, 1 or 0. We notice that the factors 90 Day Delinq, DTI, Homeowner & PG Net worth play a very prominent role.

Appendix X – Citations.

Heinze, G., & Schemper, M. (2002). A solution to the problem of separation in logistic regression.*Statistics in Medicine*, 21(16), 2409–2419. doi: 10.1002/sim.1047

RajeshSRajeshS 2933 bronze badges, & lbelzilelbelzile 40833 silver badges66 bronze badges. (1967, July 1). Logistic Regression: p values all '1', yet model fits perfectly. Retrieved April 27, 2020,from https://stats.stackexchange.com/questions/307205/logistic-regression-p-values-all-1-yet-model-fits-perfectly

"Assumptions of Logistic Regression". Statistics Solutions. https://www.statisticssolutions.com/wp-content/uploads/wp-post-to-pdf-enhanced-cache/1/assumptions-of-logistic-regression.pdf

Introduction to SAS. UCLA: Statistical Consulting Group. from https://stats.idre.ucla.edu/sas/modules/sas-learning-moduleintroduction-to-the-features-of-sas/ (accessed August 22, 2016).