BT2103 AY 22/23 SEM 1

Fong Weng Loke, Jesper A0233284H
Tan Sin Chez, Jaron A0238881R
Wang Liang Bing A0242199X

1. Description of Dataset

It contains payment information of 30,000 credit card holders obtained from a bank in Taiwan. Each data sample is described by 23 feature attributes (columns B to X). The target feature (column Y) to be predicted is binary valued 0 (= not default) or 1 (= default).

2. Data modeling problem

We want to determine the optimal model that can be used to predict whether a person is likely to default on their credit card repayment based on the information collected about the individual.

3. Attributes of dataset

1	ID	ID of each client
2	LIMIT_BAL	Amount of given credit in NT dollars (includes individual and family/supplementary credit
3	SEX	Gender (1 = male, 2 = female)
4	EDUCATION	Level of Education (1 = graduate school, 2 = university, 3 = high school, 4 = others, 0,5,6 = unknown)
5	MARRIAGE	Marital Status (1 = married, 2 = single, 0,3 = others/unknown)
6	AGE	Age in Years
7	PAY_0	Repayment status in September, 2005 (-1 = pay duly, 1 = payment delay for one month, 2 = payment delay for two months, …. 8 = payment delay for eight months, 9 = payment delay for nine months and above)
8	PAY_2	Repayment status in August, 2005 (scale same as above)
9	PAY_3	Repayment status in July, 2005 (scale same as above)
10	PAY_4	Repayment status in June, 2005 (scale same as above)
11	PAY_5	Repayment status in May, 2005 (scale same as above)
12	PAY_6	Repayment status in April, 2005 (scale same as above)
13	BILL_AMT1	Amount of bill statement in September, 2005 (NT dollar)
14	BILL_AMT2	Amount of bill statement in August, 2005 (NT dollar)
15	BILL_AMT3	Amount of bill statement in July, 2005 (NT dollar)
16	BILL_AMT4	Amount of bill statement in June, 2005 (NT dollar)
17	BILL_AMT5	Amount of bill statement in May, 2005 (NT dollar)
18	BILL_AMT6	Amount of bill statement in April, 2005 (NT dollar)
19	PAY_AMT1	Amount of previous payment in September, 2005 (NT dollar)
20	PAY_AMT2	Amount of previous payment in August, 2005 (NT dollar)
21	PAY_AMT3	Amount of previous payment in July, 2005 (NT dollar)
22	PAY_AMT4	Amount of previous payment in June, 2005 (NT dollar)
23	PAY_AMT5	Amount of previous payment in May, 2005 (NT dollar)
24	PAY_AMT6	Amount of previous payment in April, 2005 (NT dollar)
25	default_payment_next_month	Default payment (0 = no, 1 = yes)

4. Exploratory data analysis

Types of Variable:

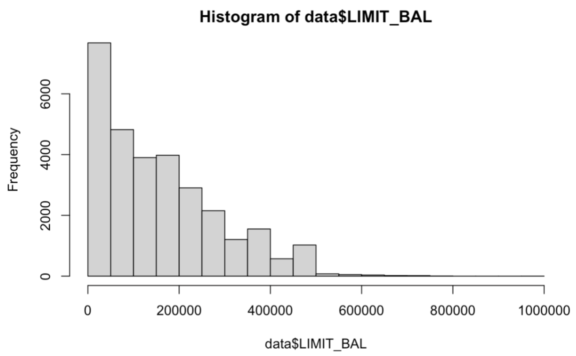
- Continuous Data (LIMIT_BAL, AGE, BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5, PAY_AMT6)
- Categorial Data (SEX, EDUCATION, MARRIAGE, PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6)
- Target Variable (categorical) : default_payment_next_month

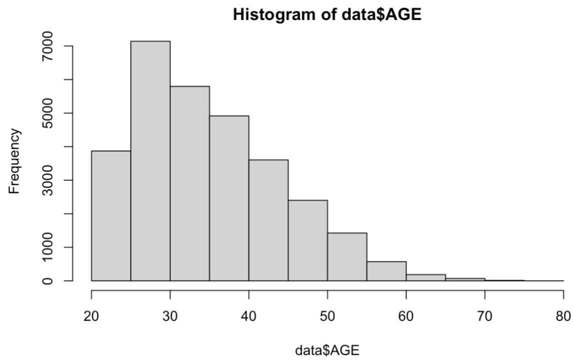
We conducted further analysis to observe the distribution of continuous variables in the dataset.

Continuous Data	**LIMIT_BAL**	**AGE**	**BILL_AMT1**	**BILL_AMT2**	**BILL_AMT3**	**BILL_AMT4**	**BILL_AMT5**
Min.	10,000	21.00	-165,580	-69,777	-157,264	-170,000	-81,334
1st Q	50,000	28.00	3,559	2,985	2,666	2,327	1,763
Median	140,000	34.00	22,382	21,200	20,088	19,052	18,104
Mean	167,484	35.49	51,223	49,179	47,013	43,263	40,311
3rd Q	240,000	41.00	67,091	64,006	60,165	54,506	50,190
Max.	1,000,000	79.00	964,511	983,931	1,664,089	891,586	927,171

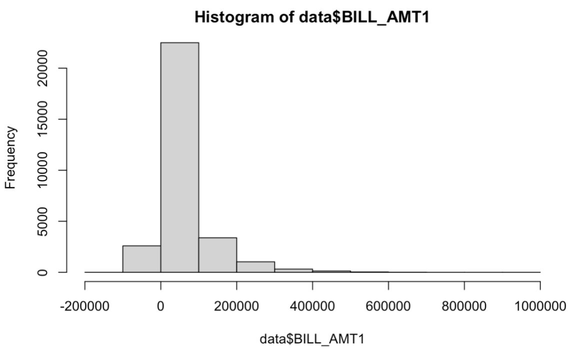
Continuous Data	**BILL_AMT6**	**PAY_AMT1**	**PAY_AMT2**	**PAY_AMT3**	**PAY_AMT4**	**PAY_AMT5**	**PAY_AMT6**
Min.	-339,603	0	0	0	0	0.0	0.0
1st Q	1,256	1,000	833	390	296	252.5	117.8
Median	17,071	2,100	2,009	1,800	1,500	1,500.0	1,500.0
Mean	38,872	5,664	5,921	5,226	4,826	4,799.4	5,215.5
3rd Q	49,198	5,006	5,000	4,505	4,013	4,031.5	4,000.0
Max.	961,664	873,552	1,684,259	896,040	621,000	426,529.0	528,666.0

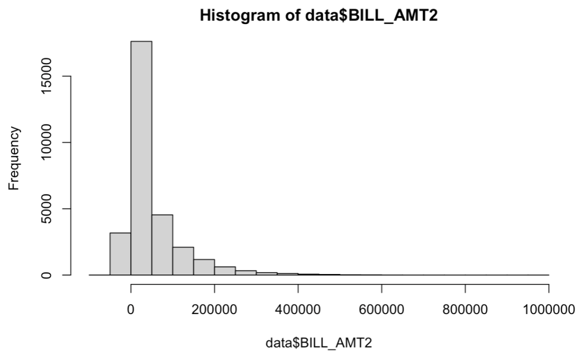
We also plotted histograms for the continuous variables to observe the general profile trend of the customers.

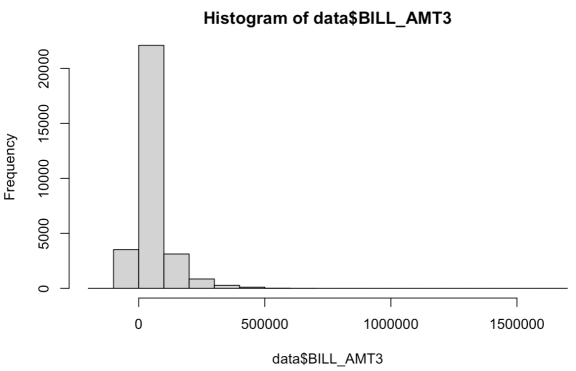


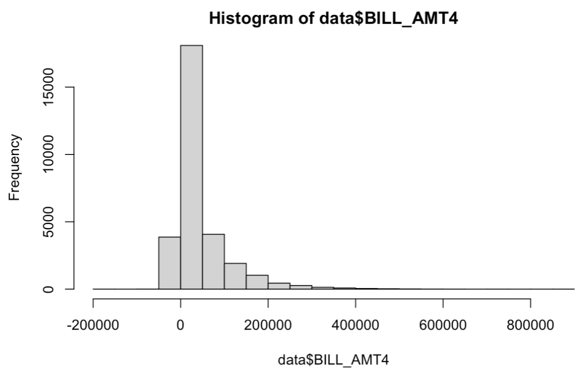


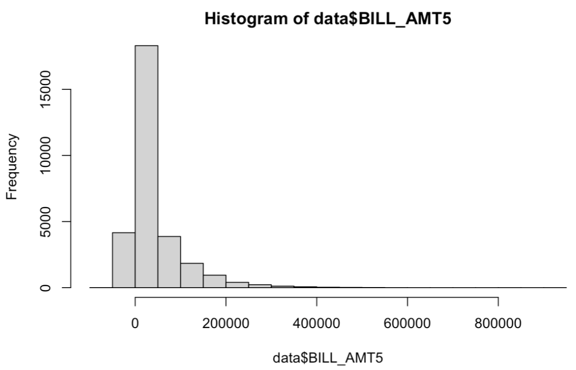
- **LIMIT_BAL**: Majority of the customers have a limit balance ranging from $0 to $500,000.
- **AGE**: Majority of the customers are within the 20-50 years old.

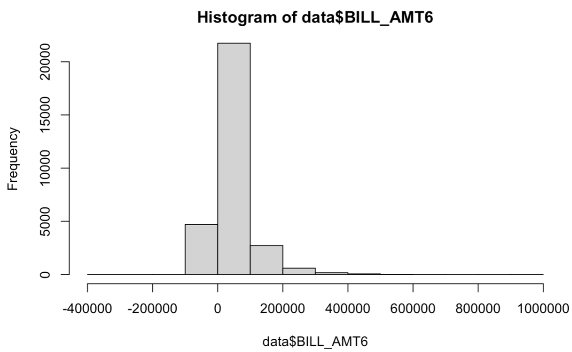




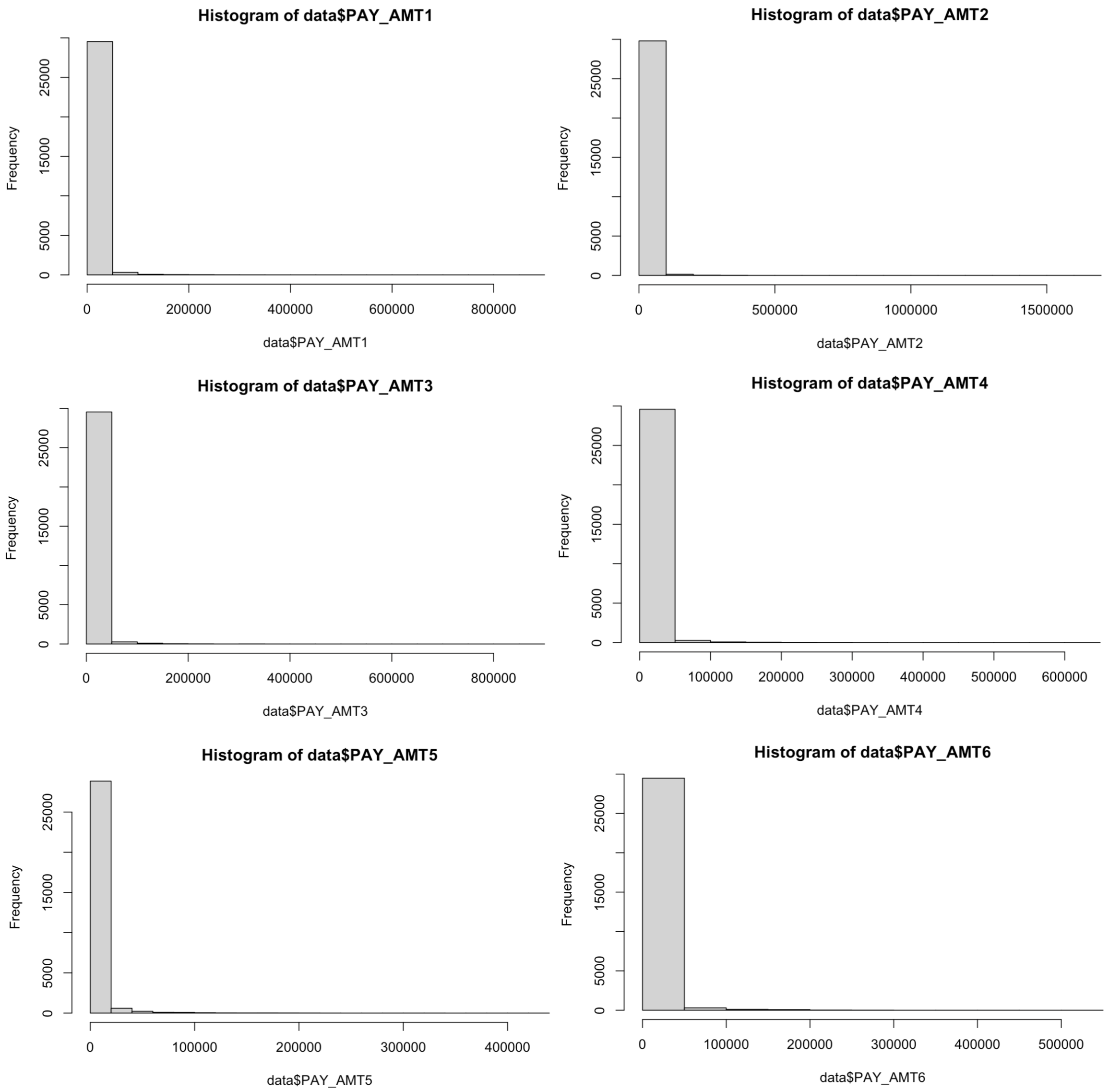








- **BILL_AMT1 - 6**: Highest concentration of customers have bill amount ranging from (-$100,000) to ($100,000)

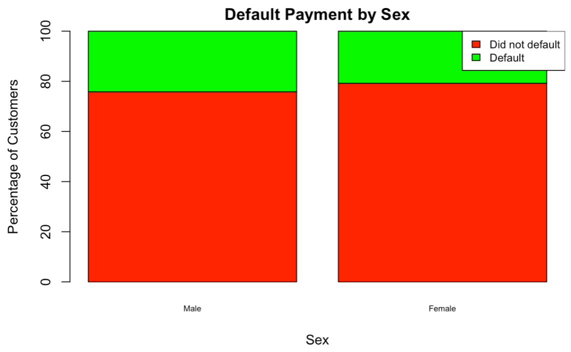


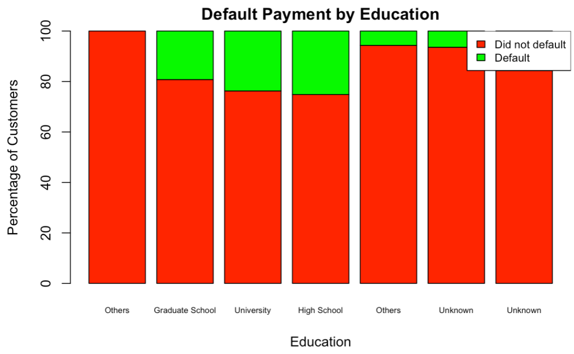
- **PAY_AMT**: Majority of the customers made a payment between $0 - $50,000, and the amount paid follows the same general distribution each month

We classified categorical variables based on the frequency of each response.

Categorical Data	SEX		EDUCATION		MARRIAGE		PAY_0		PAY_2	
	1	11,888	0	14	0	54	-2	2,759	-2	3,782
	2	18,112	1	10,585	1	13,659	-1	5,686	-1	6,050
			2	14,030	2	15,964	0	14,737	0	15,730
			3	4,917	3	323	1	3,688	1	28
			4	123			2	2,667	2	3,927
			5	280			3	322	3	326
			6	51			4	76	4	99
							5	26	5	25
							6	11	6	12
							7	9	7	20

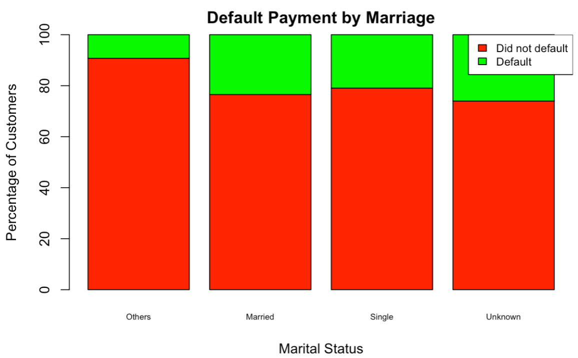
Categorical Data	PAY_3		PAY_4		PAY_5		PAY_6		default_payment_next_month	
	-2	4,085	-2	4,348	-2	4,546	-2	4,895	0	23,364
	-1	5,938	-1	5,687	-1	5,539	-1	5,740	1	6,636
	0	15,764	0	15,455	0	16,947	0	16,286		
	1	4	1	2	1	2,626	1	2,766		
	2	3,819	2	3,159	2	178	2	184		
	3	240	3	180	3	84	3	49		
	4	76	4	69	4	17	4	13		
	5	21	5	35	5	4	5	19		
	6	23	6	5	6	58	6	46		
	7	27	7	58	7	1	7	2		





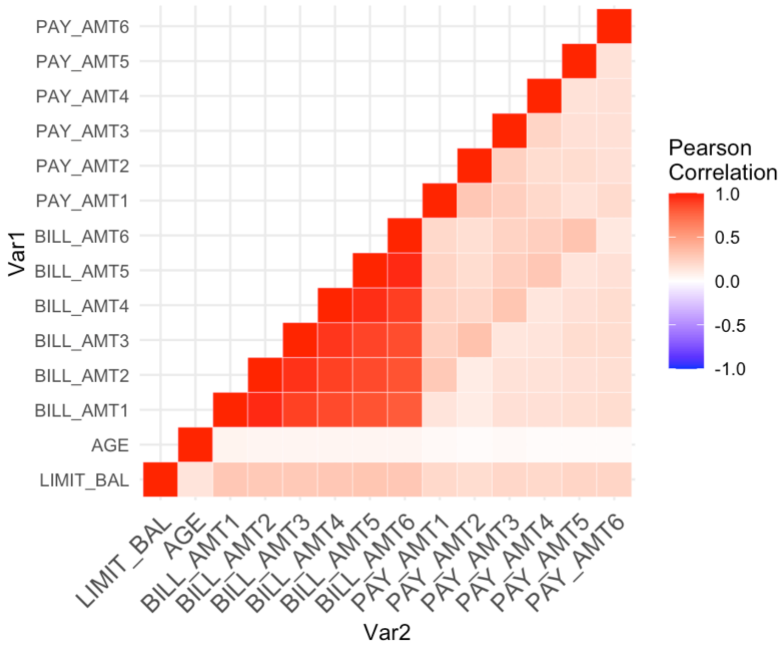
- **SEX**: Dataset has more female customers than male customers, however by the percentage of defaulters for each gender, male customers is more likely to default payment than female customers.

- **EDUCATION**: Customers with only high school education are msot likely to default payment. Furthermore, the lower the level of education, the more likely the customer defaults the payment.



- **MARRIAGE**: Married customers are more likely to default payment. Unknown and Others have very small sample size so can be taken to be insignificant.

We also plotted the correlation matrix between the continuous variables to check for multicollinearity between the independent variables.



- From the Pearson Correlation matrix, it can be seen that BILL_AMT1-6 have a high correlation which is expected that the customers lifestyle is likely to be the same over a few months and have similar expenses.

- We may have to combine the independent variables or remove them for the prediction model subsequently

5. Data pre-processing

Drop the ID column as it will not be useful in predicting default payment

EDUCATION: group 0,4,5,6 together as 4, since 5 and 6 is “unknown”, 4 is “others” and 0 is not explicitly classified.

MARRIAGE: similarly, we group 0 and 3 as 3 together since 3 is “others” and 0 is not explicitly classified.

PAY_0 to PAY_6: group 0,-1 and -2 together as 0 since 0 and -2 is not explicitly classified, so they are assumed to be representing payment made duly.

We remove all rows that have a value of 0 for BILL_AMT1 to BILL_AMT6, since these customers had a bill statement amount of 0 for the last 6 months, indicating that they are likely not using the card anymore, and thus they will likely not default payment the next month.

6. Feature selection

We introduce a new feature, the credit utilisation ratio, which is a common metric used by consumers to maintain a good credit score, as well as by banks to decide the terms of the loan provided. It is likely a good indicator of whether customers will default on payment on their credit card.

Credit utilisation ratio will be calculated by taking (BILL_AMT1 - PAY_AMT1) / LIMIT_BAL, since it is defined as the current amount of revolving credit being used (BILL_AMT - PAY_AMT) divided by the total amount of revolving credit available (LIMIT_BAL). Only BILL_AMT1 and PAY_AMT1 is used since we want to calculate the most recent credit utilisation ratio of the customer.

We introduce another new feature by combining PAY_0 to PAY_6 for all customers. This new feature will be called MONTHS_DELAYED, which represents the total number of months, over the past 6 months, that customers have been delaying their payment for. For example, if a customer delays payment for the month of May for 3 months (PAY_5 = 3), the month of July for 2 months (PAY_3 = 2) and pays duly for all other months (PAY_0, PAY_2, PAY_4, PAY_6 = 0), MONTHS_DELAYED will be 5.

We scale these 2 new features as well as AGE since they have different ranges and some models do better if features have roughly the same magnitude.

For categorical features SEX, MARRIAGE and EDUCATION, chi-square test is used to determine if they are each independent from our target feature, default.payment.next.month.

p-value <dbl>	Feature <chr>
3.229506e-179	SEX
2.534775e-09	EDUCATION
7.163905e-33	MARRIAGE

All 3 categorical features have a p-value < 0.05, so we can conclude that they are not independent from the target feature, at the 5% level of significance, and will be useful as features to train our models.

For continuous features AGE, CREDIT_UTILISATION_RATIO, MONTHS_DELAYED, BILL_AMT2 to BILL_AMT6 and PAY_AMT2 to PAY_AMT6 (BILL_AMT1 and PAY_AMT1 are not considered since they were used in creating CREDIT_UTILISATION_RATIO), we use linear discriminant analysis to determine if they will be useful in our model. Linear discriminant analysis selects the best linear combination of continuous features that separates the classes of the categorical target variable. PAY_0 to PAY_6 will not be considered as features since they were used in the creation of the MONTHS_DELAYED.

```
Coefficients of linear discriminants:
                                   LD1
AGE                       3.826426e-02
BILL_AMT2                -1.122829e-06
BILL_AMT3                 9.643273e-07
BILL_AMT4                -7.633808e-07
BILL_AMT5                 4.268197e-08
BILL_AMT6                -8.857328e-08
PAY_AMT2                 -2.178477e-06
PAY_AMT3                 -3.924232e-07
PAY_AMT4                 -2.032959e-06
PAY_AMT5                 -1.973723e-06
PAY_AMT6                 -1.890752e-06
CREDIT_UTILISATION_RATIO  1.792414e-01
MONTHS_DELAYED            1.044172e+00
```

A higher coefficient represents a higher weight of that feature in the best linear combination selected to predict the target variable. A lower coefficient represents a lower weight of that feature (not as important in predicting the target variable).
AGE, CREDIT_UTILISATION_RATIO AND MONTHS_DELAYED have the highest coefficients, and are of at least 4 orders of magnitude larger than the rest, with AGE having the smallest coefficient out of these 3. Thus, they are likely to be useful features in our model. On the other hand, BILL_AMT2 to BILL_AMT6 and PAY_AMT2 to PAY_AMT6 will likely not be useful as features in our model. They are likely not useful as features since a customer’s singular payment amount or bill amount in an independent month will likely not have an impact on whether or not he will default in the future.

Next, we will perform both forward and backward stepwise regression using the regsubsets function from the R package *leaps* to search for the best subsets of input attributes.

```
Subset selection object
Call: regsubsets.formula(default.payment.next.month ~ CREDIT_UTILISATION_RATIO +
    MONTHS_DELAYED + AGE + SEX + MARRIAGE + EDUCATION, data = train.data2,
    method = "forward")
9 Variables  (and intercept)
                         Forced in Forced out
CREDIT_UTILISATION_RATIO     FALSE      FALSE
MONTHS_DELAYED               FALSE      FALSE
AGE                          FALSE      FALSE
SEX2                         FALSE      FALSE
MARRIAGE2                    FALSE      FALSE
MARRIAGE3                    FALSE      FALSE
EDUCATION2                   FALSE      FALSE
EDUCATION3                   FALSE      FALSE
EDUCATION4                   FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: forward
         CREDIT_UTILISATION_RATIO MONTHS_DELAYED AGE SEX2 MARRIAGE2 MARRIAGE3 EDUCATION2 EDUCATION3 EDUCATION4
1  ( 1 ) " "                      "*"            " " " "  " "       " "       " "        " "        " "
2  ( 1 ) "*"                      "*"            " " " "  " "       " "       " "        " "        " "
3  ( 1 ) "*"                      "*"            " " " "  "*"       " "       " "        " "        " "
4  ( 1 ) "*"                      "*"            " " " "  "*"       " "       " "        " "        "*"
5  ( 1 ) "*"                      "*"            " " "*"  "*"       " "       " "        " "        "*"
6  ( 1 ) "*"                      "*"            " " "*"  "*"       " "       "*"        " "        "*"
7  ( 1 ) "*"                      "*"            " " "*"  "*"       " "       "*"        "*"        "*"
8  ( 1 ) "*"                      "*"            "*" "*"  "*"       " "       "*"        "*"        "*"
```

maxAdj.R2 <int>	minCP <int>	minBIC <int>	minRSS <int>
7	7	4	8

```
Subset selection object
Call: regsubsets.formula(default.payment.next.month ~ CREDIT_UTILISATION_RATIO +
    MONTHS_DELAYED + AGE + SEX + MARRIAGE + EDUCATION, data = train.data2,
    method = "backward")
9 Variables  (and intercept)
                         Forced in Forced out
CREDIT_UTILISATION_RATIO     FALSE      FALSE
MONTHS_DELAYED               FALSE      FALSE
AGE                          FALSE      FALSE
SEX2                         FALSE      FALSE
MARRIAGE2                    FALSE      FALSE
MARRIAGE3                    FALSE      FALSE
EDUCATION2                   FALSE      FALSE
EDUCATION3                   FALSE      FALSE
EDUCATION4                   FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: backward
         CREDIT_UTILISATION_RATIO MONTHS_DELAYED AGE SEX2 MARRIAGE2 MARRIAGE3 EDUCATION2 EDUCATION3 EDUCATION4
1  ( 1 ) " "                      "*"            " " " "  " "       " "       " "        " "        " "
2  ( 1 ) "*"                      "*"            " " " "  " "       " "       " "        " "        " "
3  ( 1 ) "*"                      "*"            " " " "  "*"       " "       " "        " "        " "
4  ( 1 ) "*"                      "*"            " " " "  "*"       " "       " "        " "        "*"
5  ( 1 ) "*"                      "*"            " " "*"  "*"       " "       " "        " "        "*"
6  ( 1 ) "*"                      "*"            " " "*"  "*"       " "       "*"        " "        "*"
7  ( 1 ) "*"                      "*"            " " "*"  "*"       " "       "*"        "*"        "*"
8  ( 1 ) "*"                      "*"            "*" "*"  "*"       " "       "*"        "*"        "*"
```

maxAdj.R2 <int>	minCP <int>	minBIC <int>	minRSS <int>
7	7	4	8

Both forward and backward methods selected the same subset of input variables. The subsets with the highest adjusted r-squared, lowest BIC and lowest Mallow's CP, subsets 7, 7 and 4 respectively, do not contain AGE in the subset. In our linear discriminant analysis, AGE was also shown to have the lowest coefficient out of the features that we chose. Therefore, we have decided to drop AGE as a feature.

The features we will use are:
- MARRIAGE
- EDUCATION
- SEX
- MONTHS_DELAYED
- CREDIT_UTILISATION_RATIO

7. Model selection

The models we will be training are:
1. Logistic regression
2. Support vector machine
3. Neural networks
4. Naive bayes
5. Random forest

Model 1 - Logistic Regression

Since the target variable default.payment.next.month is binary, we will use family = binomial in our model.

Strengths:
1. There is no need to worry about correlated features when doing feature selection, which is the case for many data sets.
2. Easy to implement and interpret since the model gives us a formula.
3. Gives both a measure of how important(magnitude of coefficient) a feature is as well as its association(positive or negative) with the target variable.
4. Very efficient to train.
5. Small errors are tolerable
6. Less inclined to over-fitting.

Weakness:
1. Overfitting may occur when the number of observations is less than the number of features, but this is not the case for our problem.
2. Requires observations to be independent from each other. This should not be an issue for our problem since credit card spending and user information should be independent of each other, other than in the rare and occasional cases that data is from different cards used by the same user.
3. The biggest limitation is the assumption of linearity between the independent variable and the independent variables, which is rare in real-world situations.

Model 2 - Support Vector Machines

Strengths:
1. Works well when there is clear separation between classes of the response variable.
2. No strict restriction of the number of observations given the number of features as SVM is not prone to overfitting.
3. With the introduction of the kernel, SVM works well even without the assumption of linearity between the data samples.

Weakness:

1. SVM is not suitable for datasets with many data points.
2. Does not perform well when the target classes are overlapping frequently.
3. SVM are “black boxes”. It is dIfficult to interpret how the model predicts different classes. In our case, this information might be useful for banks to understand the causes or reasons for a customer defaulting payment the next month.

Model 3 - Neural Networks

```
size         RMSE
   1 7.201015e-06
   2 2.483131e-05
   3 2.247775e-05
   4 1.182661e-05
   5 6.783285e-06
   6 2.503784e-05
   7 7.930345e-06
   8 6.040943e-06
   9 7.807357e-06
  10 2.734949e-05
```

From the matrix above, we can see that size = 8 gives us the model with the lowest RMSE (root mean squared error), so our model will be trained with size = 8.

iteration <dbl>	rmse <dbl>
100	2.045609e-04
120	7.346376e-04
140	4.942424e-04
160	1.413789e-05
180	1.558940e-04
200	2.021769e-04
220	4.091400e-05
240	1.323504e-04
260	4.935110e-05
280	5.508658e-06

iteration <dbl>	rmse <dbl>
300	2.896692e-04
320	2.632602e-05
340	1.683909e-05
360	6.272858e-05
380	6.040943e-06
400	6.040943e-06
420	6.040943e-06
440	6.040943e-06
460	6.040943e-06
480	6.040943e-06

iteration <dbl>	rmse <dbl>
500	6.040943e-06
520	6.040943e-06
540	6.040943e-06
560	6.040943e-06
580	6.040943e-06
600	6.040943e-06
620	6.040943e-06
640	6.040943e-06
660	6.040943e-06
680	6.040943e-06

iteration <dbl>	rmse <dbl>
700	6.040943e-06
720	6.040943e-06
740	6.040943e-06
760	6.040943e-06
780	6.040943e-06
800	6.040943e-06
820	6.040943e-06
840	6.040943e-06
860	6.040943e-06
880	6.040943e-06

From the data frames above, we can see that from 380 iterations onwards, root means squared error of the model reaches its lowest value. We will choose a maxit of 600 to provide some leeway, in the case that future data points are added.

Strengths:

1. Works well even when there is a non-linear relationship between the dependent and independent variables.
2. Neural Networks have the ability to work with insufficient knowledge, such as when there is missing data or inputs. There is minimal restriction on the input variables.

Weakness:

1. Similar to SVM, neural networks is a “black box”.
2. Relatively more time consuming compared to the other algorithms, but is not a problem for our dataset since it is not exceptionally large.
3. Requires much more data than other traditional machine learning algorithms.

Model 4 - Naive Bayes

Strengths:

1. Does not require too much data.
2. Easy and quick to implement.
3. Not heavily affected by irrelevant features.

Weakness:

1. Similar to logistic regression, naive bayes requires observations to be independent from each other, but this should not be an issue as explained earlier.
2. Assumes equal importance of features, which is not the case in many real world problems where different variables have different weights.

Model 5 - Random Forest

Strengths:

1. Handles large datasets efficiently and well, such as our dataset.
2. Easy to implement.

Weakness:

1. Does not work well with sparse data. However, this is not the case with our dataset since it contains no missing value.

8. Model evaluation

For our model evaluation, we are going to use the train/test split procedure where we will split the dataset into two parts, so that the model can be trained and tested on different data.
Since we are trying to classify our data into customers that default on their payment and those who don't default, we will be using the following metrics computed from a confusion matrix to evaluate our model: Accuracy, Precision, Sensitivity, F1 score, Average Class Accuracy and False Negative Rate.

The imbalance ratio of our dataset is 3.446761, which indicates that there is an imbalance in the train data. Because classification accuracy can mask poor performance of our models when the data is unbalanced, in this report we will evaluate the models based on average class accuracy rather than accuracy.

Below is a summary of the metrics we are using for our train and test data.

For train data:

Metric <chr>	Linear Model <chr>	SVM <chr>	Neural Network <chr>	Naive Bayes <chr>	Random Forest <chr>
Accuracy	0.809	0.809	0.81	0.808	0.806
Precision	0.623	0.643	0.626	0.631	0.611
Sensitivity	0.348	0.305	0.355	0.316	0.334
F1score	0.447	0.414	0.453	0.421	0.432
Average Class Accuracy	0.644	0.629	0.647	0.632	0.637
False Negative Rate	0.652	0.695	0.645	0.684	0.666

For test data:

Metric <chr>	Linear Model <chr>	SVM <chr>	Neural Network <chr>	Naive Bayes <chr>	Random Forest <chr>
Accuracy	0.813	0.811	0.814	0.813	0.815
Precision	0.663	0.683	0.664	0.68	0.674
Sensitivity	0.367	0.317	0.368	0.338	0.362
F1score	0.472	0.433	0.474	0.452	0.471
Average Class Accuracy	0.656	0.637	0.656	0.645	0.655
False Negative Rate	0.633	0.683	0.632	0.662	0.637

For evaluation on the test split of the dataset, random forest has the highest accuracy, SVM has the highest precision, neural networks has the highest sensitivity, F1 and average class accuracy. It also has the lowest false negative rate.

Conclusion

The neural networks model is the best model as it does the best in the sensitivity, F1 and average class accuracy. More importantly, it has the lowest false negative rate among all the models. We used false negative rate in addition to the common statistics to evaluate the models because it is more important for banks to reduce the number of incorrect non-default predictions compared to an incorrect default prediction. Making an incorrect non-default prediction when a customer actually defaults will cause the bank to be more likely to incur a loss compared to if an incorrect default prediction was made.
Hence, we conclude that the neural networks model is the best model to predict defaults and non-defaults.

9. Room for improvement

K-fold cross-validation can be done as it gives an even better estimate of out-of-sample performance so we can make a more fair comparison of model performances as it ensures well-balanced splits.

To improve our findings, banks should also collect more data so that additional features have a higher correlation to the target class. Most of the features (such as PAY_AMT AND BILL_AMT) have little correlation to the target variable, hence we can collect other data such as their customer's purchase history and length of credit history, which might be more useful in helping us to predict if a customer will default on their payment or not.

In addition, banks should also collect data on the customer's salaries. If customers have higher or more consistent streams of income, they will be less likely to default on their payments.

Finally, since the dataset provided was from 2005, it may be outdated and the findings may not be as valuable now. We should attempt to collect more recent data.

In [5]:
```
library(plyr)
library(dplyr)
library(readxl)
library(ggplot2)
library(reshape2)
library(tidyr)


data <- read.table("card.csv",sep=",",skip=2,header=FALSE)
header <- scan("card.csv",sep=",",nlines=2,what=character())

names(data) = c("ID", "LIMIT_BAL", "SEX", "EDUCATION", "MARRIAGE", "AGE",
                "PAY_0", "PAY_2", "PAY_3", "PAY_4", "PAY_5", "PAY_6",
                "BILL_AMT1", "BILL_AMT2", "BILL_AMT3", "BILL_AMT4","BILL_AMT5", "BILL_
                "PAY_AMT1", "PAY_AMT2", "PAY_AMT3", "PAY_AMT4", "PAY_AMT5", "PAY_AMT6'
                "default.payment.next.month")

head(data)
set.seed(1234)

#check if there is any Null data
sum(is.na(data))

#general info for continuous and categorical data
summary(data)

count(data, 'SEX')
count(data, 'EDUCATION')
count(data, 'MARRIAGE')
count(data, 'PAY_0')
count(data, 'PAY_2')
count(data, 'PAY_3')
count(data, 'PAY_4')
count(data, 'PAY_5')
count(data, 'PAY_6')
count(data, 'default.payment.next.month')

options(scipen=999)

#visualisation

hist(data$LIMIT_BAL)
hist(data$AGE)

hist(data$BILL_AMT1)
hist(data$BILL_AMT2)
hist(data$BILL_AMT3)
hist(data$BILL_AMT4)
hist(data$BILL_AMT5)
hist(data$BILL_AMT6)

hist(data$PAY_AMT1)
hist(data$PAY_AMT2)
hist(data$PAY_AMT3)
hist(data$PAY_AMT4)
hist(data$PAY_AMT5)
hist(data$PAY_AMT6)
```

```
data2 <- data %>% dplyr::select(SEX, default.payment.next.month)
data2.gather <- data2 %>% gather(key = "SEX", value = "default.payment.next.month") %>
data2.spread <- data2.gather %>% spread(key = SEX, value = freq)
value <- as.matrix(data2.spread[2:3])
data_percentage <- apply(value, 2, function(x){x*100/sum(x,na.rm=T)})

campaigns <- c("Male", "Female")
colors <- c("red", "green")
legend <- c("Did not default", "Default")

barplot(data_percentage,
        main = "Default Payment by Sex",
        names.arg = campaigns,
        xlab = "Sex",
        ylab = "Percentage of Customers",
        col = colors,
        beside = FALSE,
        cex.names = 0.6,
        ylim = c(0,100))

legend("topright", legend, cex = 0.8, fill = colors)


data3 <- data %>% dplyr::select(MARRIAGE, default.payment.next.month)
data3.gather <- data3 %>% gather(key = "MARRIAGE", value = "default.payment.next.month
data3.spread <- data3.gather %>% spread(key = MARRIAGE, value = freq)
value3 <- as.matrix(data3.spread[2:5])
data_percentage3 <- apply(value3, 2, function(x){x*100/sum(x,na.rm=T)})

campaigns3 <- c("Others", "Married", "Single", "Unknown")


barplot(data_percentage3,
        main = "Default Payment by Marital Status",
        names.arg = campaigns3,
        xlab = "Marital Status",
        ylab = "Percentage of Customers",
        col = colors,
        beside = FALSE,
        cex.names = 0.6,
        ylim = c(0,100))

legend("topright", legend, cex = 0.8, fill = colors)

data4 <- data %>% dplyr::select(EDUCATION, default.payment.next.month)
data4.gather <- data4 %>% gather(key = "EDUCATION", value = "default.payment.next.mont
data4.spread <- data4.gather %>% spread(key = EDUCATION, value = freq)
value4 <- as.matrix(data4.spread[2:8])
data_percentage4 <- apply(value4, 2, function(x){x*100/sum(x,na.rm=T)})

campaigns4 <- c("Others", "Graduate School", "University", "High School", "Others", "U


barplot(data_percentage4,
        main = "Default Payment by Education",
        names.arg = campaigns4,
        xlab = "Education",
        ylab = "Percentage of Customers",
        col = colors,
        beside = FALSE,
```

```
        cex.names = 0.6,
        ylim = c(0,100))

legend("topright", legend, cex = 0.8, fill = colors)


corr_data <- data %>% dplyr::select(LIMIT_BAL, AGE, BILL_AMT1, BILL_AMT2, BILL_AMT3, B
                             PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5, PAY_AMT

corr_matrix <- round(cor(corr_data),2)

lower_tri <- function(corr_matrix){
    corr_matrix[upper.tri(corr_matrix)] <- NA
    return(corr_matrix)
  }

upper_tri <- function(corr_matrix){
    corr_matrix[lower.tri(corr_matrix)]<- NA
    return(corr_matrix)
}

final_tri <- upper_tri(corr_matrix)
final_tri

melted_corr <- melt(final_tri, na.rm = TRUE)

ggplot(data = melted_corr, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
 coord_fixed()


#data pre processing
data2 <- data[-c(1)]

data2$SEX = as.factor(data2$SEX)
data2$EDUCATION = as.factor(ifelse(data$EDUCATION %in% c(0,4,5,6),4, data$EDUCATION))
data2$MARRIAGE = as.factor(ifelse(data$MARRIAGE %in% c(0,3), 3, data$MARRIAGE))

data2$PAY_0 = (ifelse(data2$PAY_0 %in% c(0,-1,-2), 0, data2$PAY_0))
data2$PAY_2 = (ifelse(data2$PAY_2 %in% c(0,-1,-2), 0, data2$PAY_2))
data2$PAY_3 = (ifelse(data2$PAY_3 %in% c(0,-1,-2), 0, data2$PAY_3))
data2$PAY_4 = (ifelse(data2$PAY_4 %in% c(0,-1,-2), 0, data2$PAY_4))
data2$PAY_5 = (ifelse(data2$PAY_5 %in% c(0,-1,-2), 0, data2$PAY_5))
data2$PAY_6 = (ifelse(data2$PAY_6 %in% c(0,-1,-2), 0, data2$PAY_6))

data2 <- subset(data2, BILL_AMT1 != "0" & BILL_AMT2 != "0" & BILL_AMT3 != "0" & BILL_A


data2$CREDIT_UTILISATION_RATIO = (data2$BILL_AMT1 - data2$PAY_AMT1) / data2$LIMIT_BAL

data2$MONTHS_DELAYED = (data2$PAY_0 + data2$PAY_2 + data2$PAY_3 + data2$PAY_4 + data2$

data2$CREDIT_UTILISATION_RATIO = scale(data2$CREDIT_UTILISATION_RATIO)
data2$MONTHS_DELAYED = scale(data2$MONTHS_DELAYED)
```

```
data2$AGE = scale(data2$AGE)

data2$default.payment.next.month = as.factor(data2$default.payment.next.month)

#feature selection

chistat <- matrix(0,3,2)
class = as.factor(data2[,24])
vars = c("SEX", "EDUCATION","MARRIAGE")
for (i in 1:3) {
  x = as.factor(data2[,i])
  tbl = table(x,class)
  cat( "\n Attribute = " , i, vars[i], "\n")
  print(tbl)
  chi2res <- chisq.test(tbl)
  print(chi2res)
  chistat[i,1] <- chi2res$statistic
  chistat[i,2] <- chi2res$p.value
}

df <- data.frame(chistat[,1:2],vars)
names(df) <- c("chi2 stat","p-value","Feature")
df

library(MASS)

subsetdf = data2[c(5,13:17,19:26)]

model = lda(default.payment.next.month~., data = subsetdf)
model

n = nrow(data2)
index <- 1:nrow(data2)
testindex <- sample(index, trunc(n)/4)
test.data2 <- data2[testindex,]
train.data2 <- data2[-testindex,]

library(leaps)



outforward = regsubsets(default.payment.next.month ~ CREDIT_UTILISATION_RATIO + MONTHS

outbackward = regsubsets(default.payment.next.month ~ CREDIT_UTILISATION_RATIO + MONTH

summary(outforward)
summary(outbackward)

plot(outforward,scale="r2")
plot(outbackward,scale="r2")

res.sum <- summary(outforward)
data.frame(
  maxAdj.R2 = which.max(res.sum$adjr2),
  minCP = which.min(res.sum$cp),
  minBIC = which.min(res.sum$bic),
  minRSS = which.min(res.sum$rss)
)

res.sum1 <- summary(outbackward)
```

```
data.frame(
  maxAdj.R2 = which.max(res.sum1$adjr2),
  minCP = which.min(res.sum1$cp),
  minBIC = which.min(res.sum1$bic),
  minRSS = which.min(res.sum1$rss)
)

#model selection


# linear model
library(InformationValue)

linearmodel = glm(default.payment.next.month ~ CREDIT_UTILISATION_RATIO + MONTHS_DELAY
summary(linearmodel)


#SVM model
library(e1071)

svm = svm(default.payment.next.month ~ CREDIT_UTILISATION_RATIO + MONTHS_DELAYED + SEX



#neural network
library(nnet)
library(NeuralNetTools)

size_rmse_matrix = matrix(nrow = 10, ncol = 2)
colnames(size_rmse_matrix) <- c("size", "RMSE")
for (i in 1:10) {
  set.seed(1234)
  nn = nnet(default.payment.next.month ~ CREDIT_UTILISATION_RATIO + MONTHS_DELAYED + S

size_rmse_matrix[i,1] = i
size_rmse_matrix[i,2] = sqrt(mean(nn$residuals)^2) #rmse
}
size_rmse_matrix


rmse <- NULL
iteration <- NULL
for (i in seq(100,1000,by = 20)) {
  set.seed(1234)
  nn = nnet(default.payment.next.month ~ CREDIT_UTILISATION_RATIO + MONTHS_DELAYED + S
  rmse <- append(rmse, sqrt(mean(nn$residuals)^2))
  iteration <- append(iteration, i)

}

data.frame(cbind(iteration,rmse))

set.seed(1234)
neural = nnet(default.payment.next.month ~ CREDIT_UTILISATION_RATIO + MONTHS_DELAYED +


#naive bayes model
library(naivebayes)
set.seed(1234)
naive = naive_bayes(default.payment.next.month ~ CREDIT_UTILISATION_RATIO + MONTHS_DEL
```

```
#random forest model
library(randomForest)
set.seed(1234)
forest = randomForest(default.payment.next.month ~ CREDIT_UTILISATION_RATIO + MONTHS_D
print(forest)

#model evaluation
getIMR <- function(data2){
    minCl <- names(which.min(table(data2$default.payment.next.month)))
    sum(data2$default.payment.next.month!=minCl)/sum(data2$default.payment.next.month=
}
getIMR(data2)

metric_title = c("Accuracy", "Precision", "Sensitivity", "F1 score", "Average Class Ac

eval = function(actual, pred) {
tp = sum(pred == 1 & actual == 1)
tn = sum(pred == 0 & actual == 0)
fp = sum(pred == 1 & actual == 0)
fn = sum(pred == 0 & actual == 1)

accuracy = round((tp + tn) / (tp + tn + fp + fn),3)
precision = round(tp / (fp + tp),3)
sensitivity= round(tp / (fn + tp),3)
f1score = round((2 * precision * sensitivity) / (precision + sensitivity),3)
avg_class_acc = round(((tp / (tp + fn)) + (tn / (tn + fp))) / 2 ,3)
FNR = round(fn / (fn + tp),3)

print(table(actual = actual, predicted = pred))
cat("\n")

return (c(accuracy, precision, sensitivity, f1score, avg_class_acc, FNR)) }


print_eval_output = function(trainset, testset) {
  output = cbind(trainset, testset)
  output = cbind(metric_title, output)
  output = data.frame(output)
  names(output) = c("Metric", "Train", "Test")
  print(output)
  return (output) }


#GLM
#train
linearmodel_train = predict.glm(linearmodel, newdata = train.data2, type = "response")
optcut = optimalCutoff(train.data2$default.payment.next.month,linearmodel_train)
linearmodel_train_pred = as.factor(ifelse(linearmodel_train < optcut,0,1))
linearmodel_train_metric = eval(train.data2$default.payment.next.month, linearmodel_tr

#test
linearmodel_test = predict.glm(linearmodel, newdata = test.data2, type = "response")
linearmodel_test_pred = as.factor(ifelse(linearmodel_test < optcut,0,1))
linearmodel_test_metric = eval(test.data2$default.payment.next.month, linearmodel_test

#print output
linearmodel_output = print_eval_output(linearmodel_train_metric, linearmodel_test_metr
```

```
#SVM
svm_train_pred = predict(svm, newdata = train.data2, type = "class")
svm_train_metric = eval(train.data2$default.payment.next.month, svm_train_pred)

#test
svm_test_pred = predict(svm, newdata = test.data2, type = "class")
svm_test_metric = eval(test.data2$default.payment.next.month, svm_test_pred)
#print output
svm_output = print_eval_output(svm_train_metric, svm_test_metric)



#NN
#train
neural_train_pred = factor(predict(neural,data = train.data2, type = c("class")))
neural_train_metric = eval(train.data2$default.payment.next.month, neural_train_pred)

#test
neural_test_pred = factor(predict(neural,newdata = test.data2, type = c("class")))
neural_test_metric = eval(test.data2$default.payment.next.month, neural_test_pred)

#print output
neural_output = print_eval_output(neural_train_metric, neural_test_metric)


#Naive bayes
#train
naive_train_pred = predict(naive, data = train.data2, type = "class")
naive_train_metric = eval(train.data2$default.payment.next.month, naive_train_pred)

#test
naive_test_pred = predict(naive, newdata = test.data2, type = "class")
naive_test_metric = eval(test.data2$default.payment.next.month, naive_test_pred)

#print output
naive_output = print_eval_output(naive_train_metric, naive_test_metric)


#random forest
#train
forest_train_pred = predict(forest, data = train.data2, type = "class")
forest_train_metric = eval(train.data2$default.payment.next.month, forest_train_pred)

#test
forest_test_pred = predict(forest, newdata = test.data2, type = "class")
forest_test_metric = eval(test.data2$default.payment.next.month, forest_test_pred)

#print output
forest_output = print_eval_output(forest_train_metric, forest_test_metric)



model_results = c(linearmodel_output, svm_output, neural_output, naive_output, forest_
metric_names = c("Accuracy", "Precision", "Sensitivity", "F1score", "Average Class Acc
model_names = c("Linear Model", "SVM", "Neural Network", "Naive Bayes", "Random Forest

all_train_output = cbind(linearmodel_output$Train, svm_output$Train, neural_output$Tra
all_test_output = cbind(linearmodel_output$Test, svm_output$Test, neural_output$Test,
```

```
final_train_output = data.frame(cbind(metric_names, all_train_output))
names(final_train_output) = append("Metric", model_names)
final_train_output

final_test_output = data.frame(cbind(metric_names, all_test_output))
names(final_test_output) = append("Metric", model_names)
final_test_output
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	B
1	20000	2	2	1	24	2	2	-1	-1	...	0	
2	120000	2	2	2	26	-1	2	0	0	...	3272	
3	90000	2	2	2	34	0	0	0	0	...	14331	
4	50000	2	2	1	37	0	0	0	0	...	28314	
5	50000	1	2	1	57	-1	0	-1	0	...	20940	
6	50000	1	1	2	37	0	0	0	0	...	19394	

0

```
       ID          LIMIT_BAL            SEX          EDUCATION    
 Min.   :    1   Min.   :  10000   Min.   :1.000   Min.   :0.000  
 1st Qu.: 7501   1st Qu.:  50000   1st Qu.:1.000   1st Qu.:1.000  
 Median :15000   Median : 140000   Median :2.000   Median :2.000  
 Mean   :15000   Mean   : 167484   Mean   :1.604   Mean   :1.853  
 3rd Qu.:22500   3rd Qu.: 240000   3rd Qu.:2.000   3rd Qu.:2.000  
 Max.   :30000   Max.   :1000000   Max.   :2.000   Max.   :6.000  
    MARRIAGE          AGE            PAY_0             PAY_2        
 Min.   :0.000   Min.   :21.00   Min.   :-2.0000   Min.   :-2.0000  
 1st Qu.:1.000   1st Qu.:28.00   1st Qu.:-1.0000   1st Qu.:-1.0000  
 Median :2.000   Median :34.00   Median : 0.0000   Median : 0.0000  
 Mean   :1.552   Mean   :35.49   Mean   :-0.0167   Mean   :-0.1338  
 3rd Qu.:2.000   3rd Qu.:41.00   3rd Qu.: 0.0000   3rd Qu.: 0.0000  
 Max.   :3.000   Max.   :79.00   Max.   : 8.0000   Max.   : 8.0000  
     PAY_3             PAY_4             PAY_5             PAY_6        
 Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000  
 1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000  
 Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000  
 Mean   :-0.1662   Mean   :-0.2207   Mean   :-0.2662   Mean   :-0.2911  
 3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000  
 Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000  
   BILL_AMT1         BILL_AMT2        BILL_AMT3         BILL_AMT4      
 Min.   :-165580   Min.   :-69777   Min.   :-157264   Min.   :-170000  
 1st Qu.:   3559   1st Qu.:  2985   1st Qu.:   2666   1st Qu.:   2327  
 Median :  22382   Median : 21200   Median :  20089   Median :  19052  
 Mean   :  51223   Mean   : 49179   Mean   :  47013   Mean   :  43263  
 3rd Qu.:  67091   3rd Qu.: 64006   3rd Qu.:  60165   3rd Qu.:  54506  
 Max.   : 964511   Max.   :983931   Max.   :1664089   Max.   : 891586  
   BILL_AMT5        BILL_AMT6          PAY_AMT1         PAY_AMT2      
 Min.   :-81334   Min.   :-339603   Min.   :     0   Min.   :      0  
 1st Qu.:  1763   1st Qu.:   1256   1st Qu.:  1000   1st Qu.:    833  
 Median : 18105   Median :  17071   Median :  2100   Median :   2009  
 Mean   : 40311   Mean   :  38872   Mean   :  5664   Mean   :   5921  
 3rd Qu.: 50191   3rd Qu.:  49198   3rd Qu.:  5006   3rd Qu.:   5000  
 Max.   :927171   Max.   : 961664   Max.   :873552   Max.   :1684259  
    PAY_AMT3         PAY_AMT4         PAY_AMT5           PAY_AMT6       
 Min.   :     0   Min.   :     0   Min.   :     0.0   Min.   :     0.0  
 1st Qu.:   390   1st Qu.:   296   1st Qu.:   252.5   1st Qu.:   117.8  
 Median :  1800   Median :  1500   Median :  1500.0   Median :  1500.0  
 Mean   :  5226   Mean   :  4826   Mean   :  4799.4   Mean   :  5215.5  
 3rd Qu.:  4505   3rd Qu.:  4013   3rd Qu.:  4031.5   3rd Qu.:  4000.0  
 Max.   :896040   Max.   :621000   Max.   :426529.0   Max.   :528666.0  
 default.payment.next.month
 Min.   :0.0000            
 1st Qu.:0.0000            
 Median :0.0000            
 Mean   :0.2212            
 3rd Qu.:0.0000            
 Max.   :1.0000            
```

"SEX"	n
SEX	30000

"EDUCATION"	n
EDUCATION	30000

"MARRIAGE"	n
MARRIAGE	30000

"PAY_0"	n
PAY_0	30000

"PAY_2"	n
PAY_2	30000

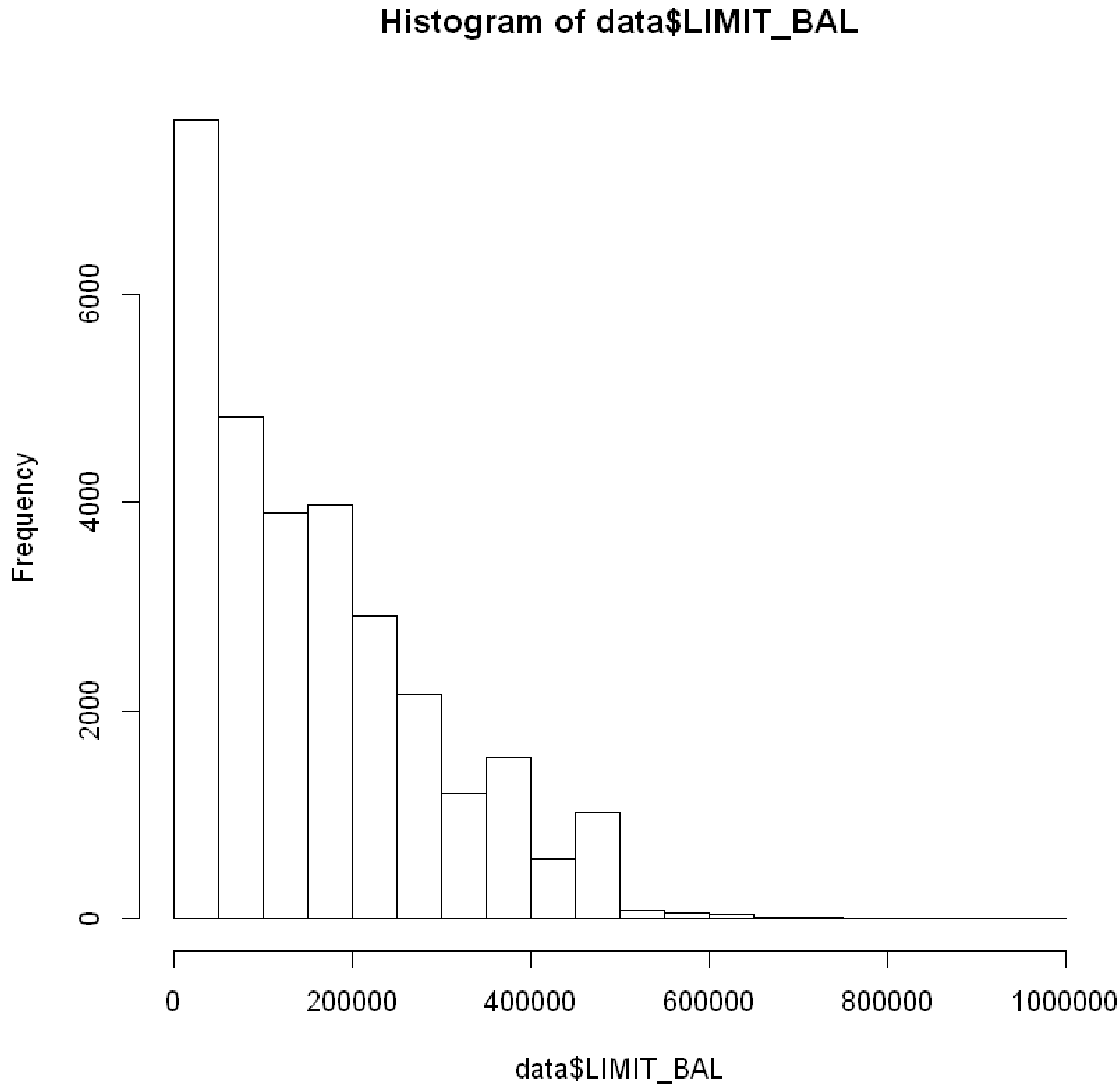
"PAY_3"	n
PAY_3	30000

"PAY_4"	n
PAY_4	30000

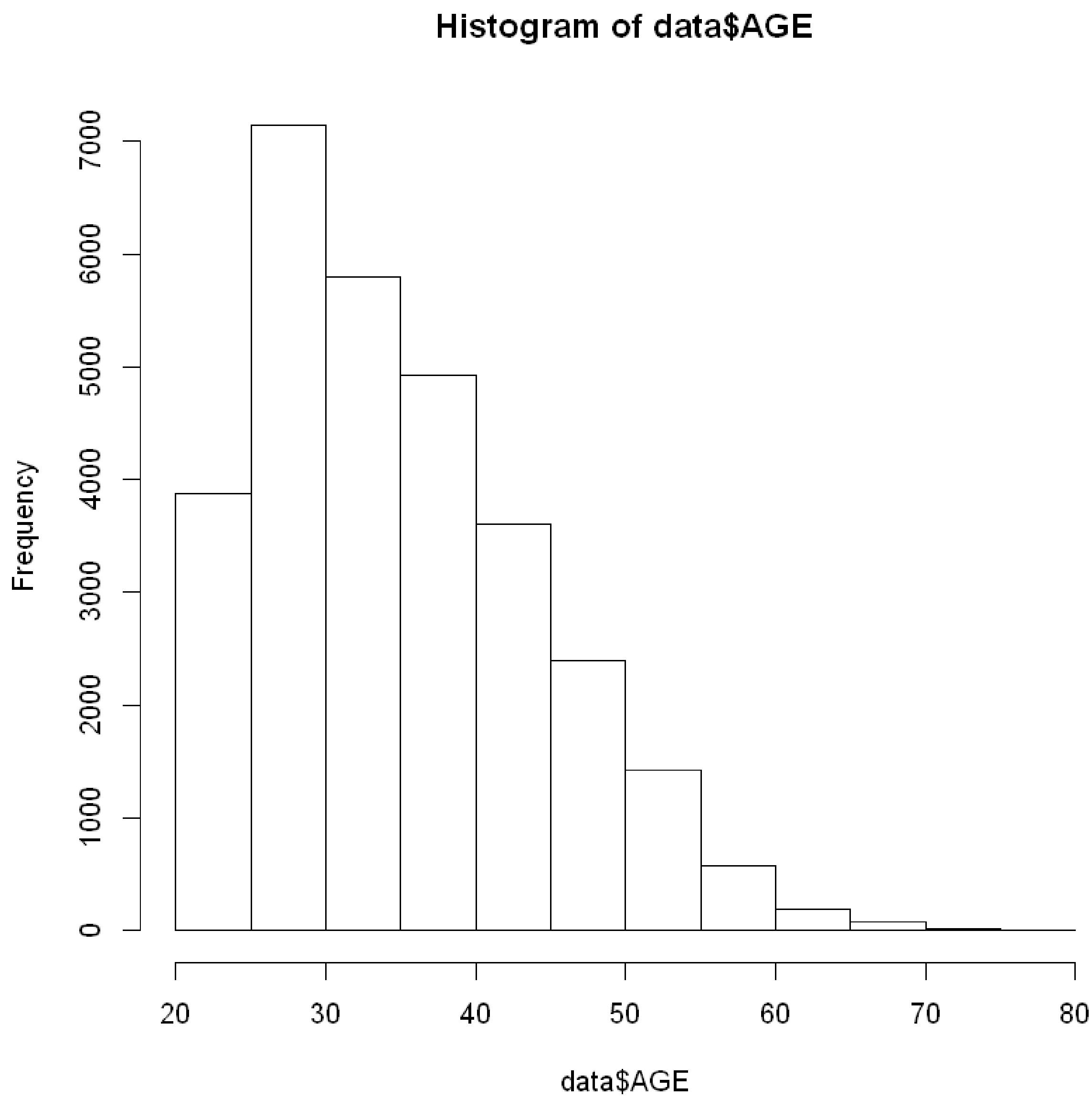
"PAY_5"	n
PAY_5	30000

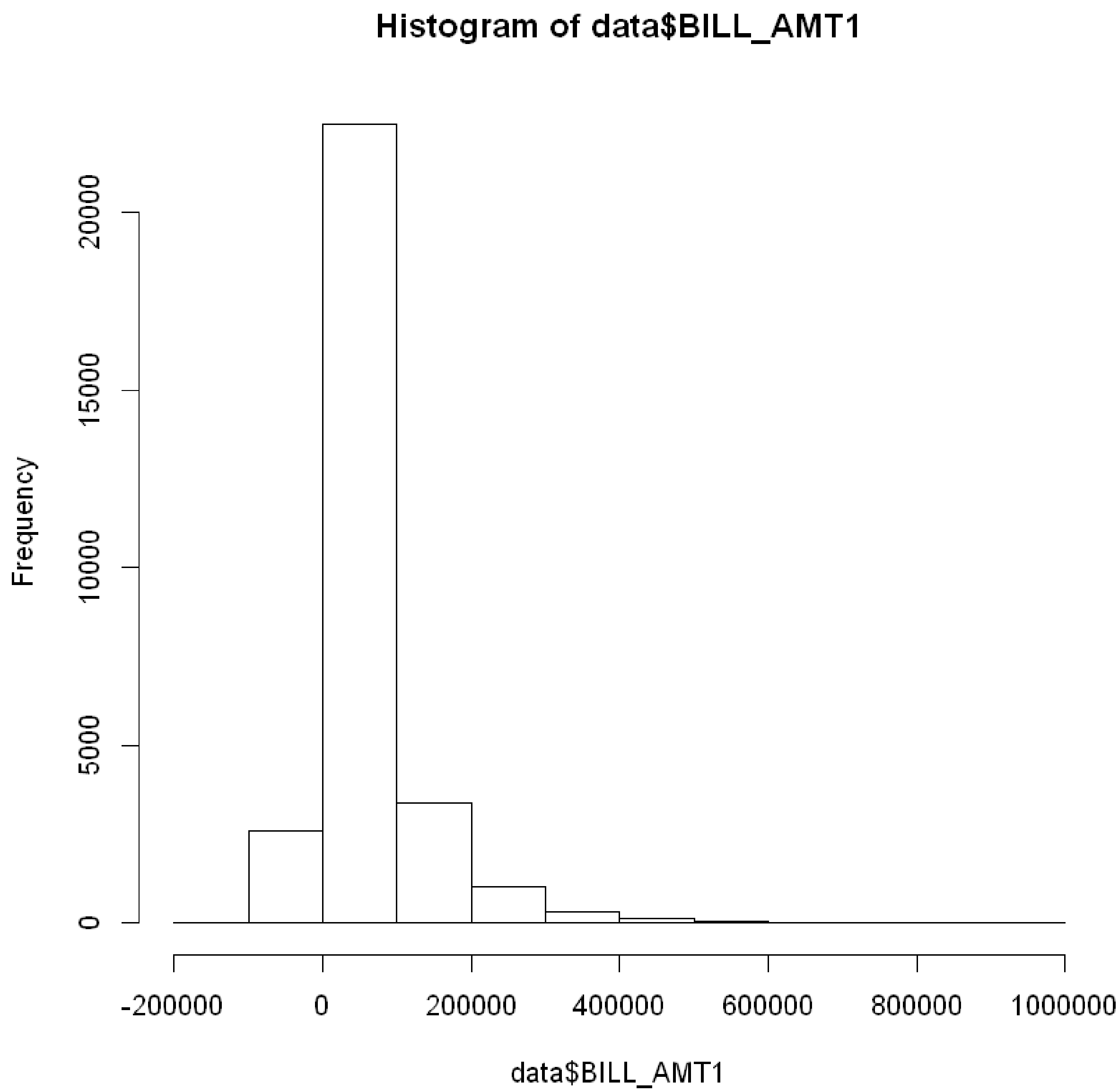
"PAY_6"	n
PAY_6	30000

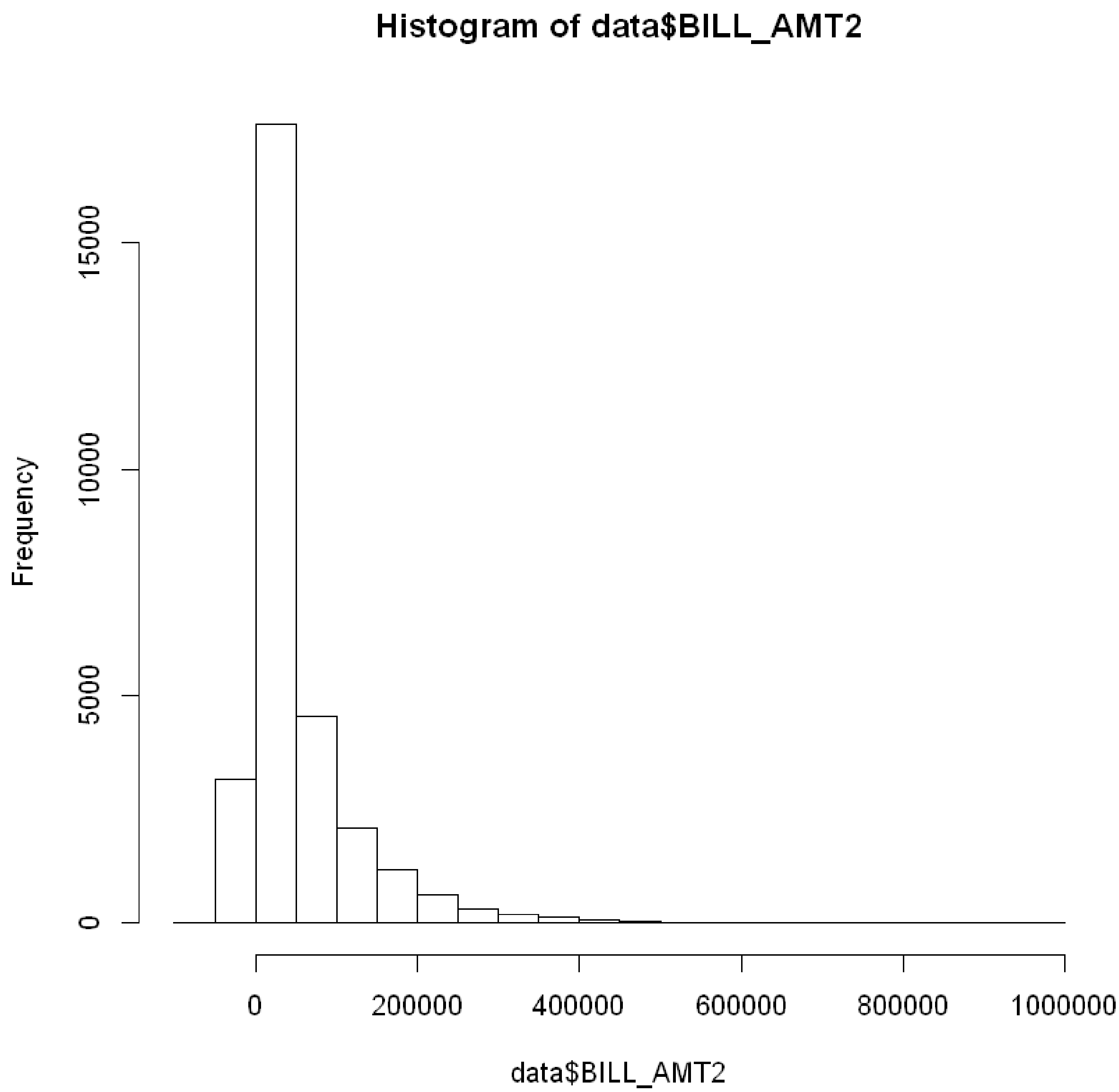
"default.payment.next.month"	n
default.payment.next.month	30000

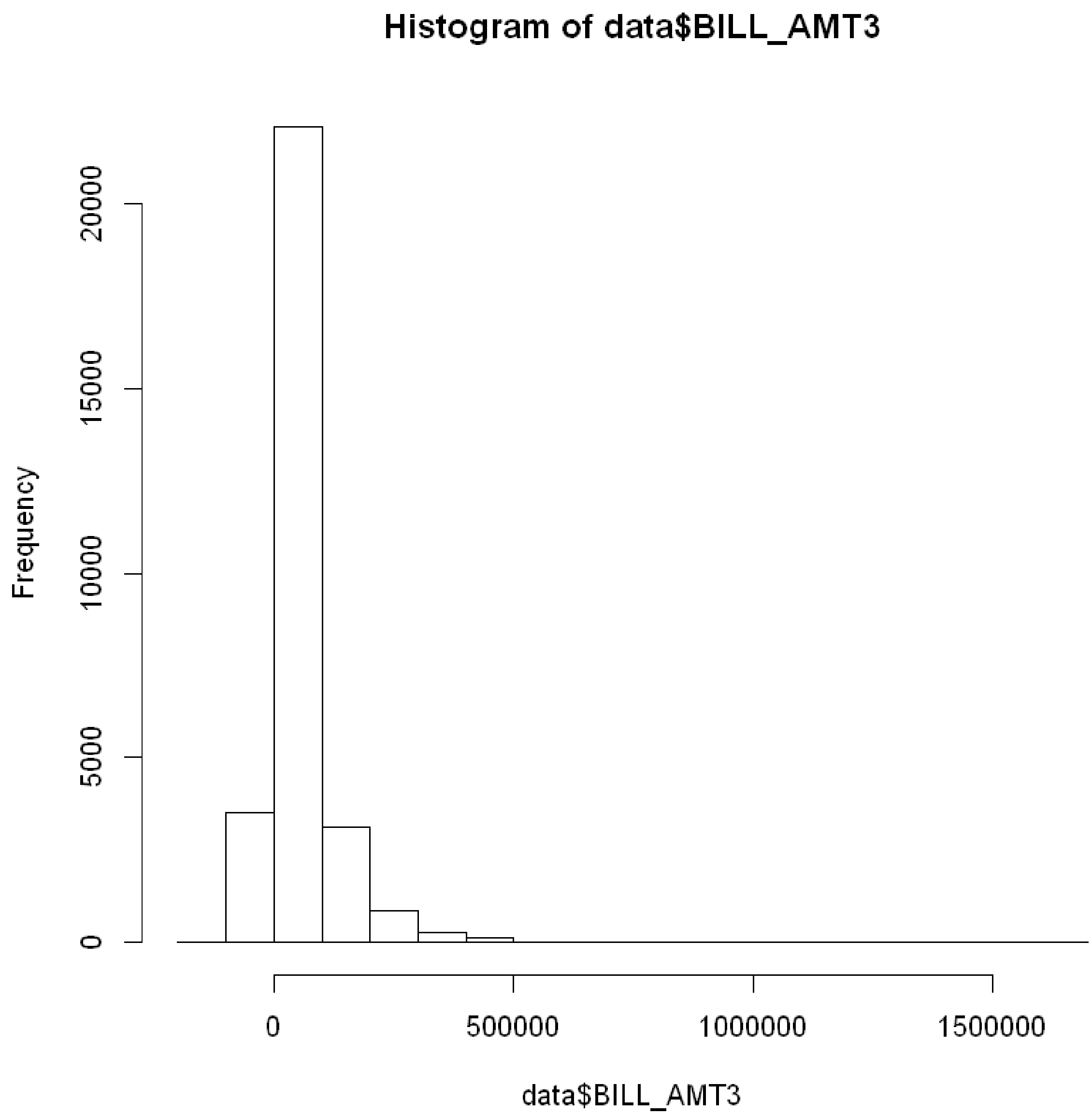


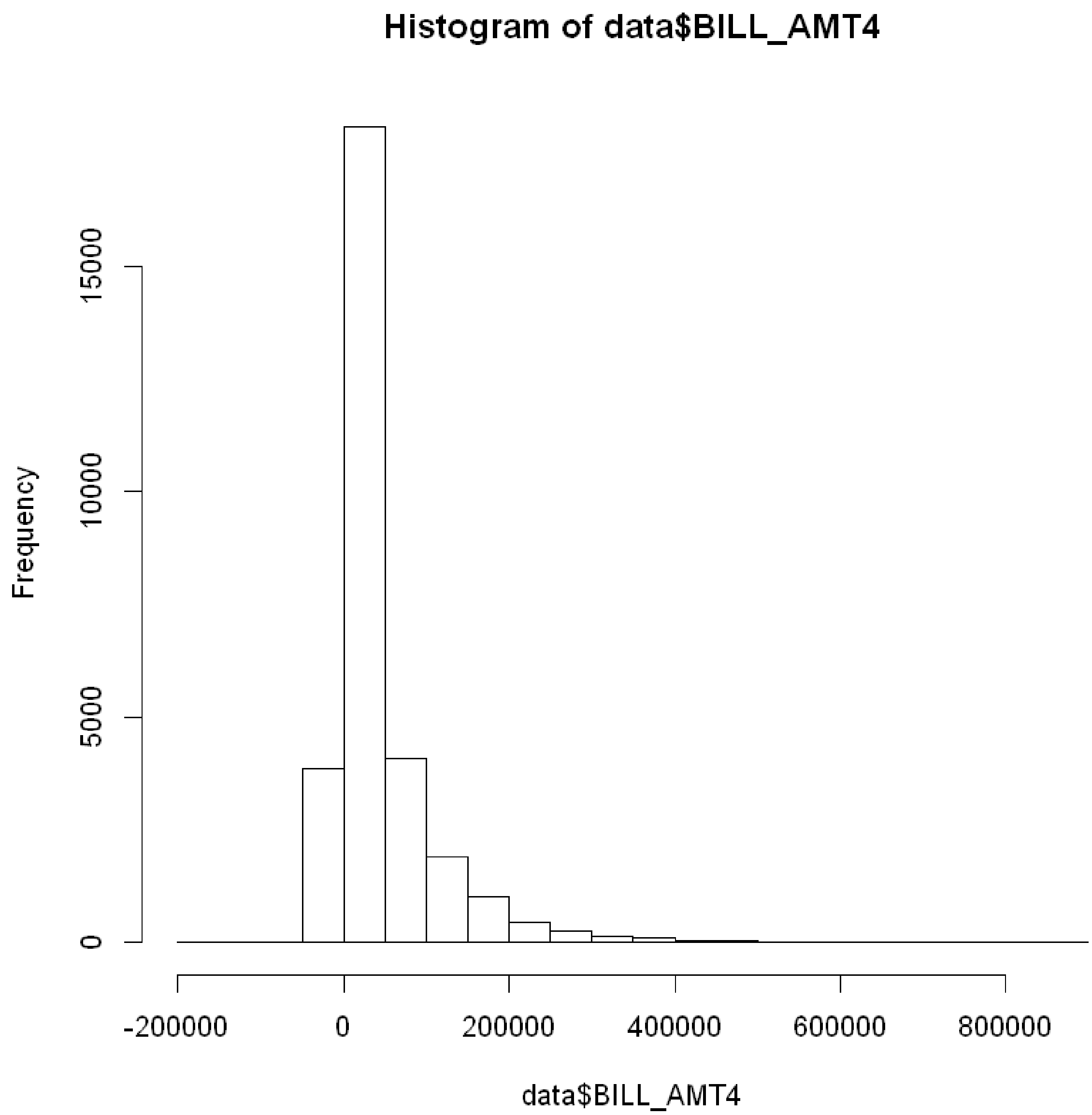
Histogram of data$LIMIT_BAL
Frequency
0
2000
4000
6000
0
200000
400000
600000
800000
1000000
data$LIMIT_BAL

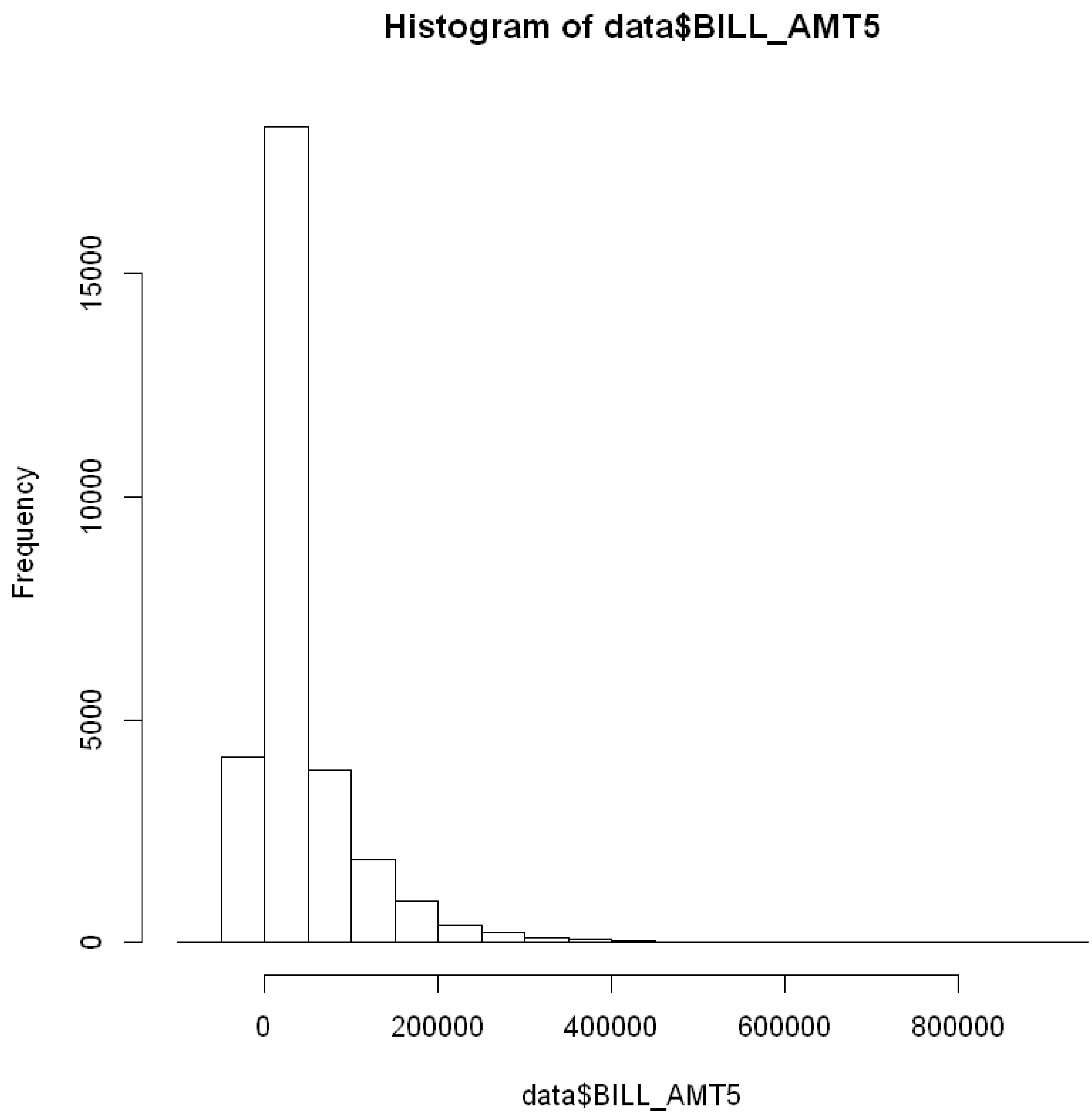

Histogram of data$AGE
Frequency
0
1000
2000
3000
4000
5000
6000
7000
20
30
40
50
60
70
80
data$AGE

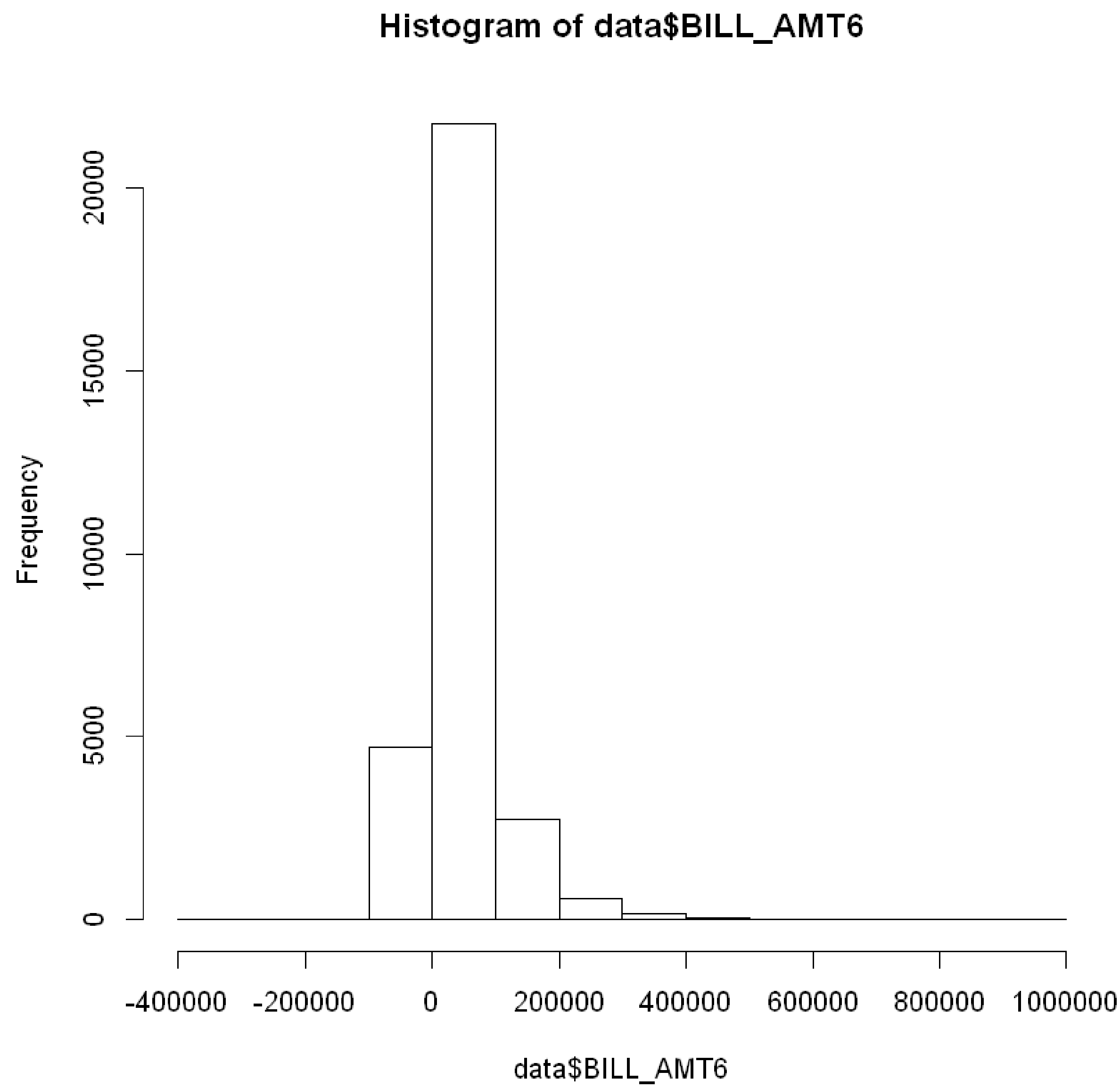

Histogram of data$BILL_AMT1
Frequency
0
5000
10000
15000
20000
-200000
0
200000
400000
600000
800000
1000000
data$BILL_AMT1

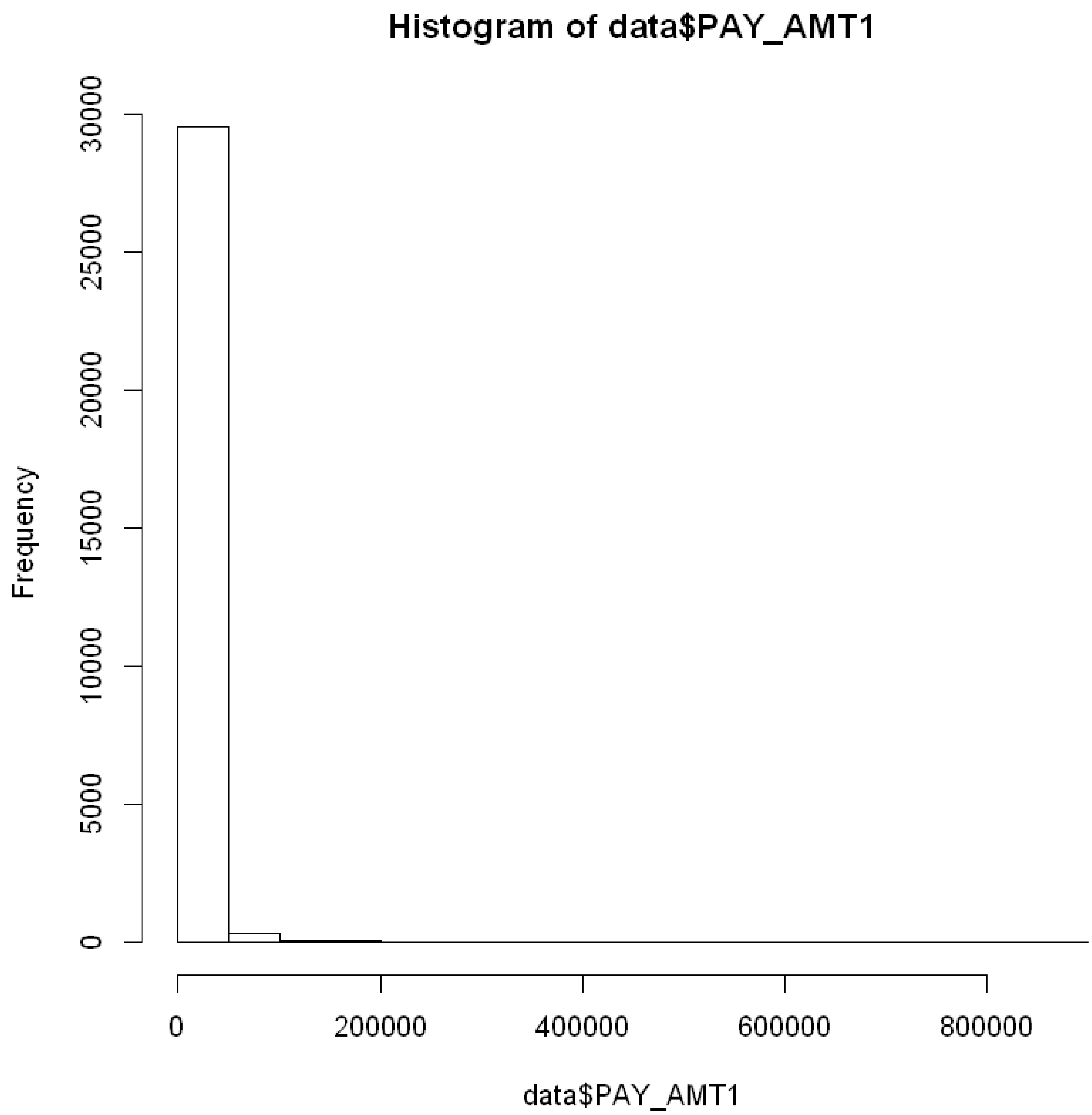

Histogram of data$BILL_AMT2
Frequency
0
5000
10000
15000
0
200000
400000
600000
800000
1000000
data$BILL_AMT2

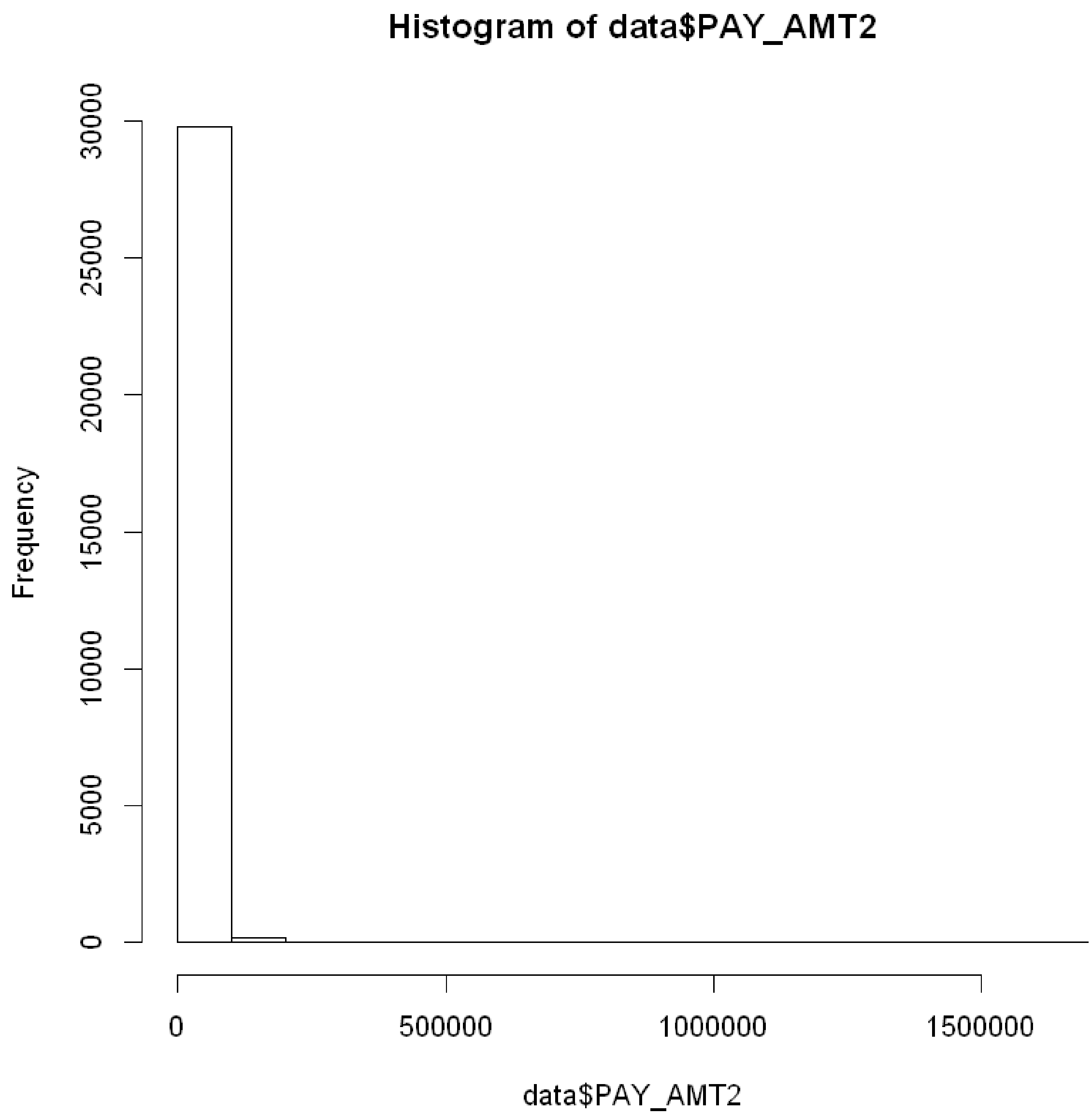

Histogram of data$BILL_AMT3
Frequency
0
5000
10000
15000
20000
0
500000
1000000
1500000
data$BILL_AMT3

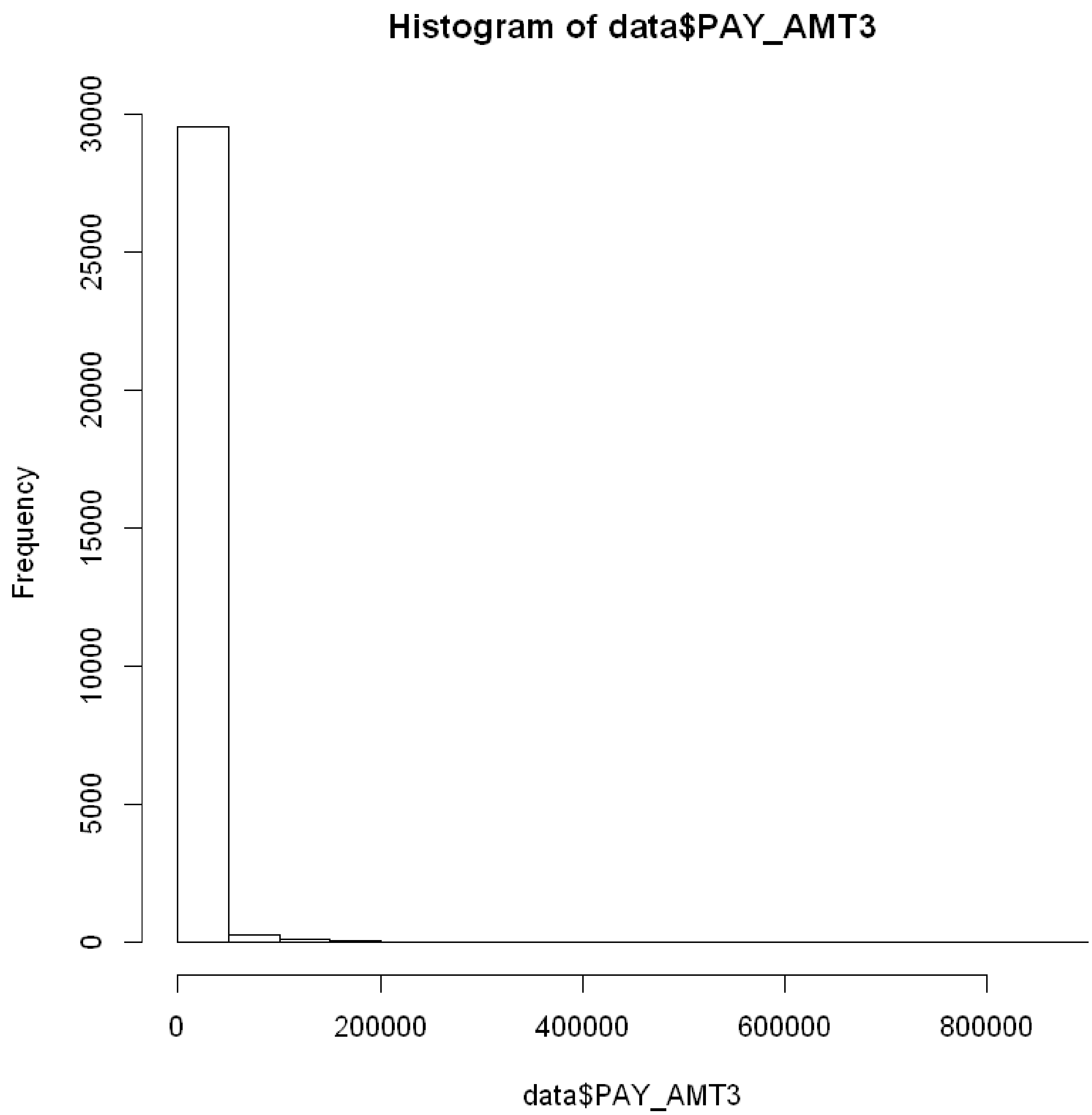

Histogram of data$BILL_AMT4
Frequency
0
5000
10000
15000
-200000
0
200000
400000
600000
800000
data$BILL_AMT4

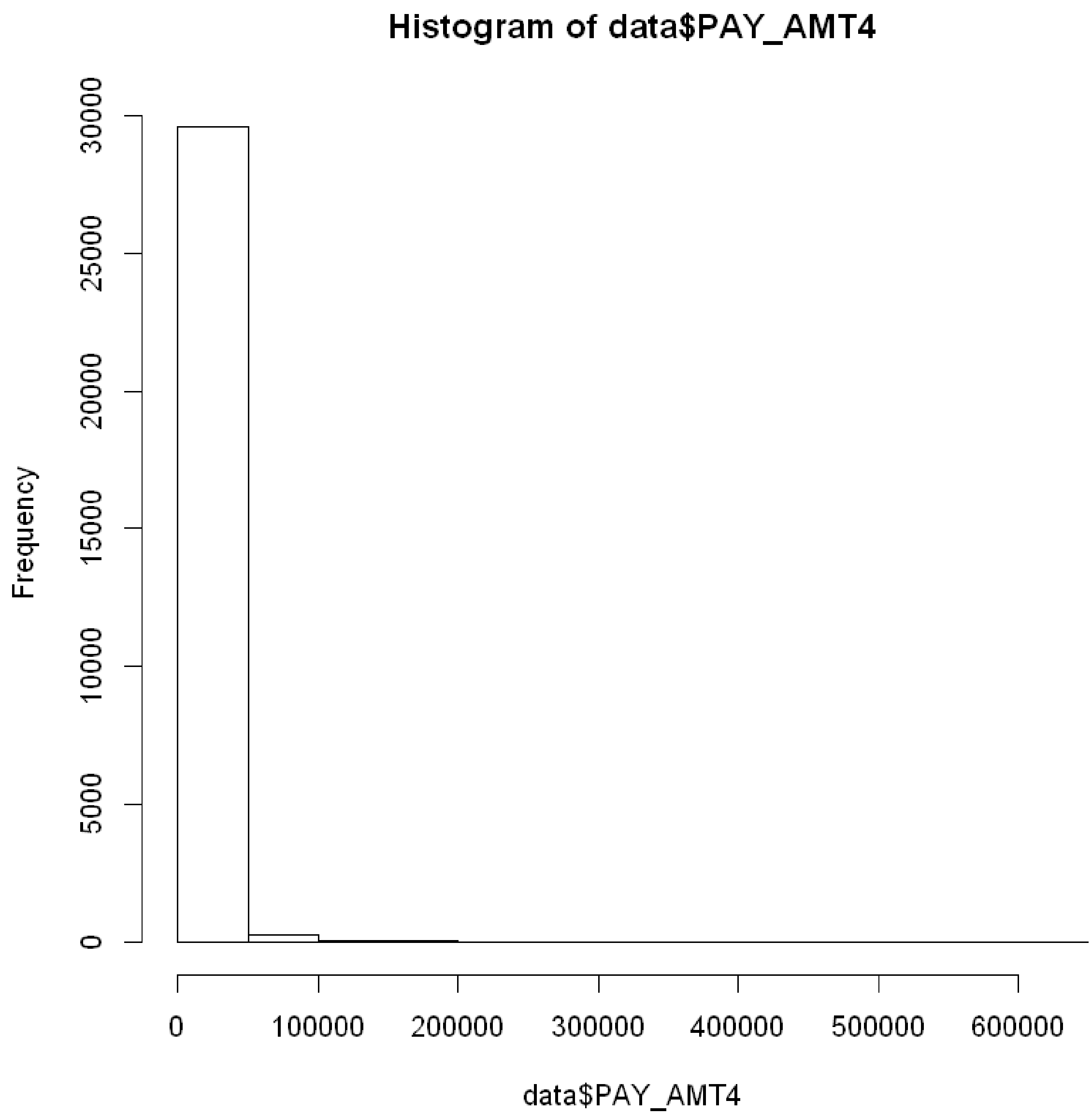

Histogram of data$BILL_AMT5
Frequency
0
5000
10000
15000
0
200000
400000
600000
800000
data$BILL_AMT5

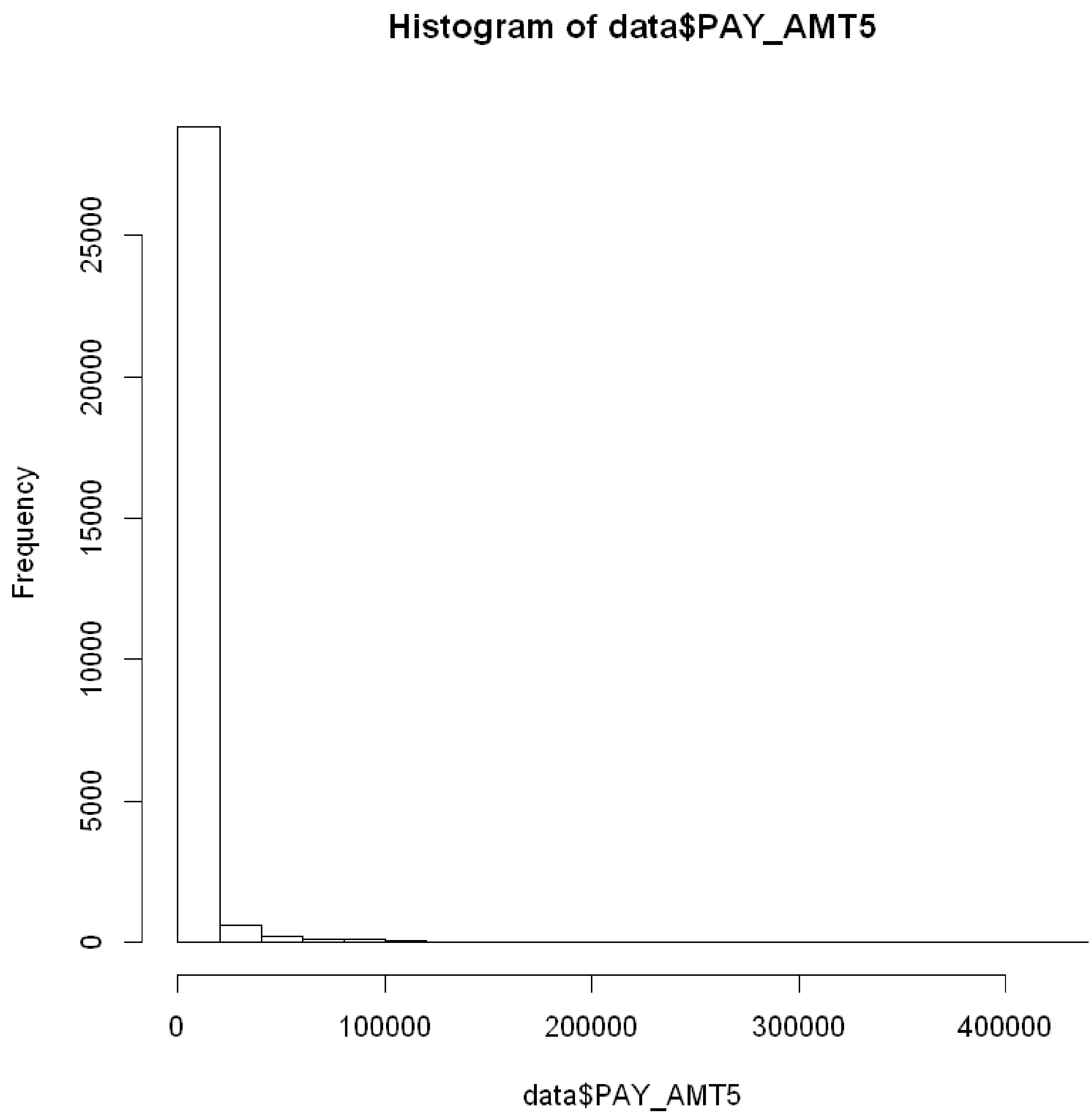

Histogram of data$BILL_AMT6
Frequency
0
5000
10000
15000
20000
-400000
-200000
0
200000
400000
600000
800000
1000000
data$BILL_AMT6

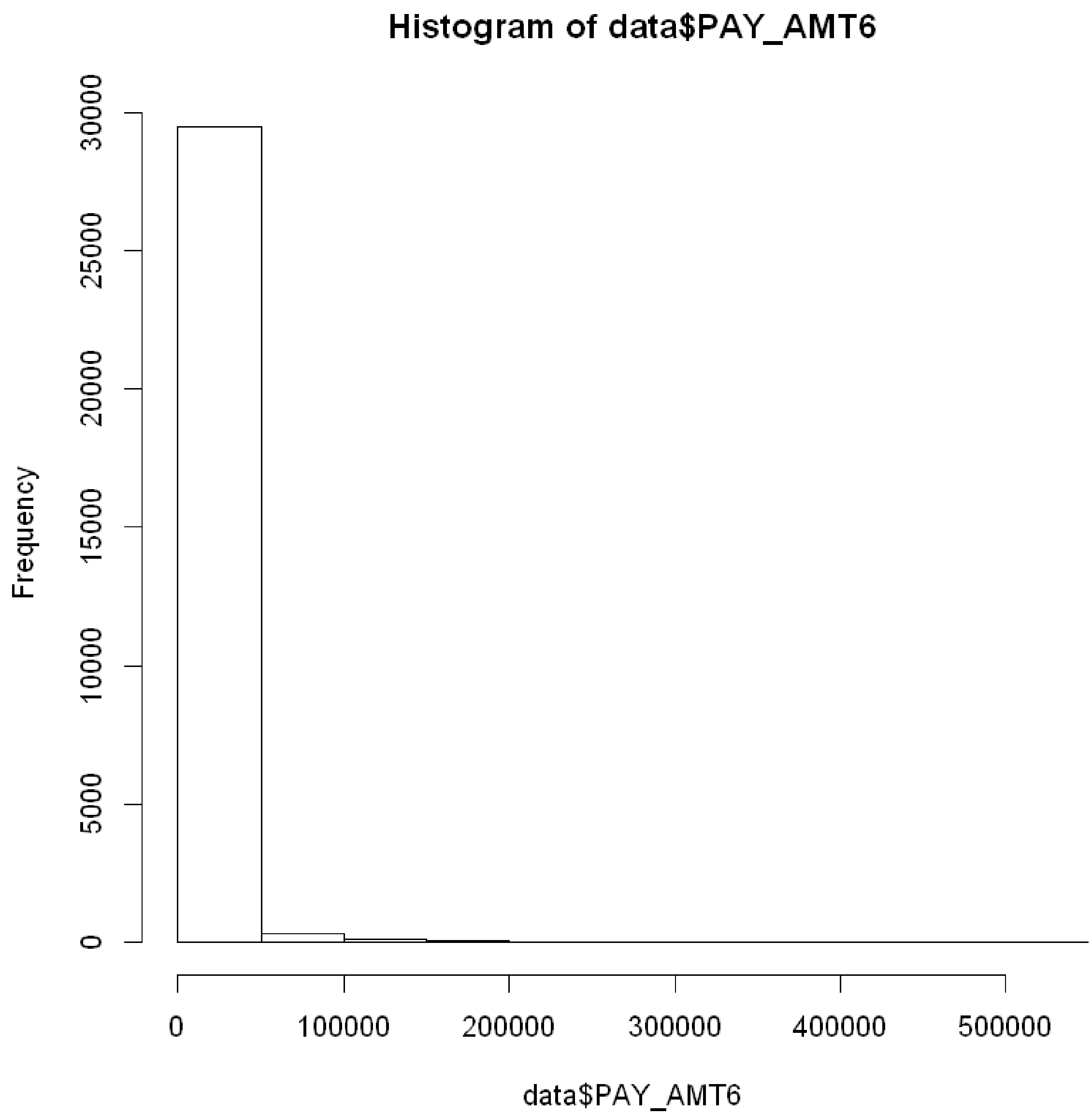

Histogram of data$PAY_AMT1
Frequency
0
5000
10000
15000
20000
25000
30000
0
200000
400000
600000
800000
data$PAY_AMT1

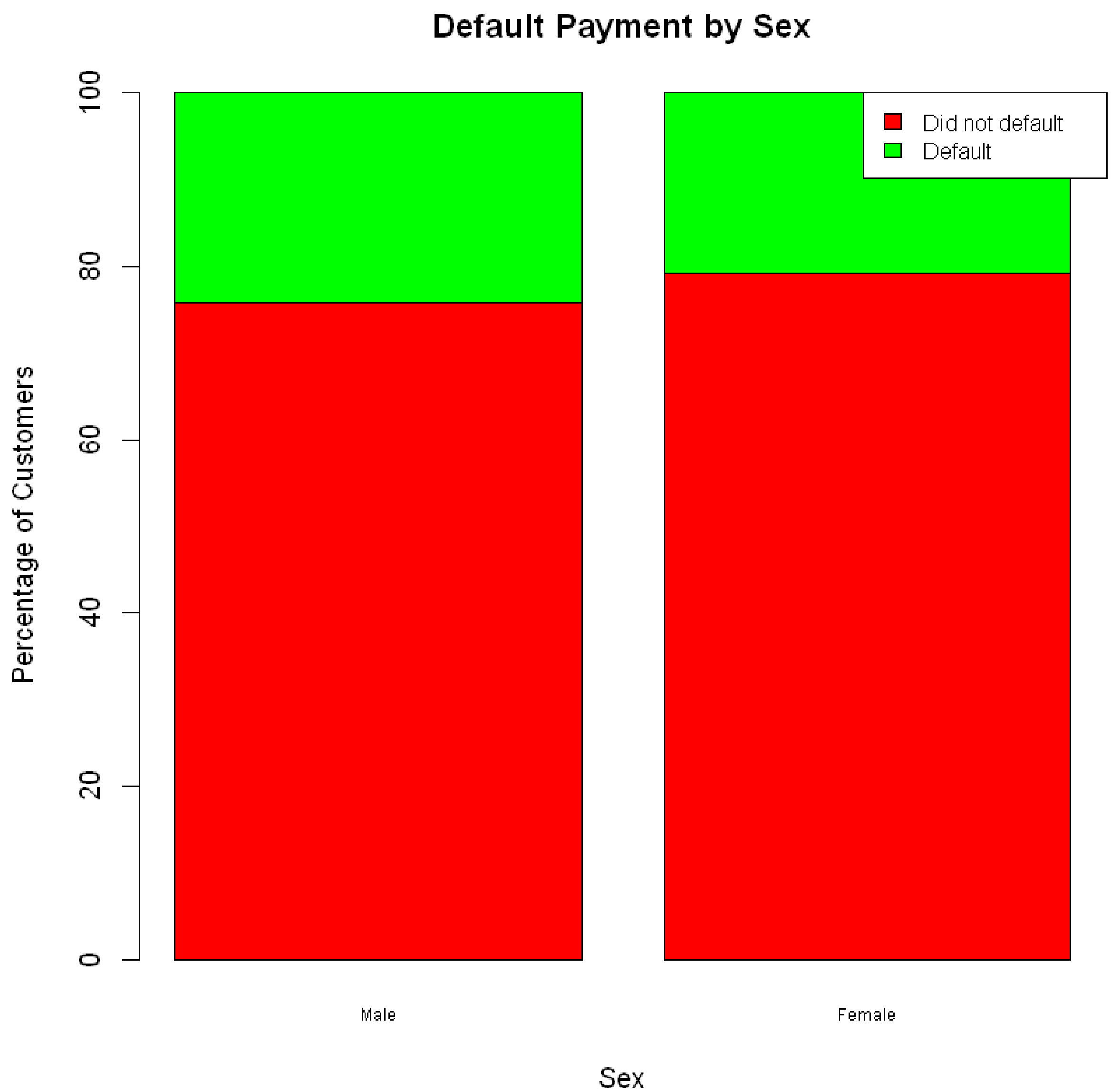

Histogram of data$PAY_AMT2
Frequency
0
5000
10000
15000
20000
25000
30000
0
500000
1000000
1500000
data$PAY_AMT2

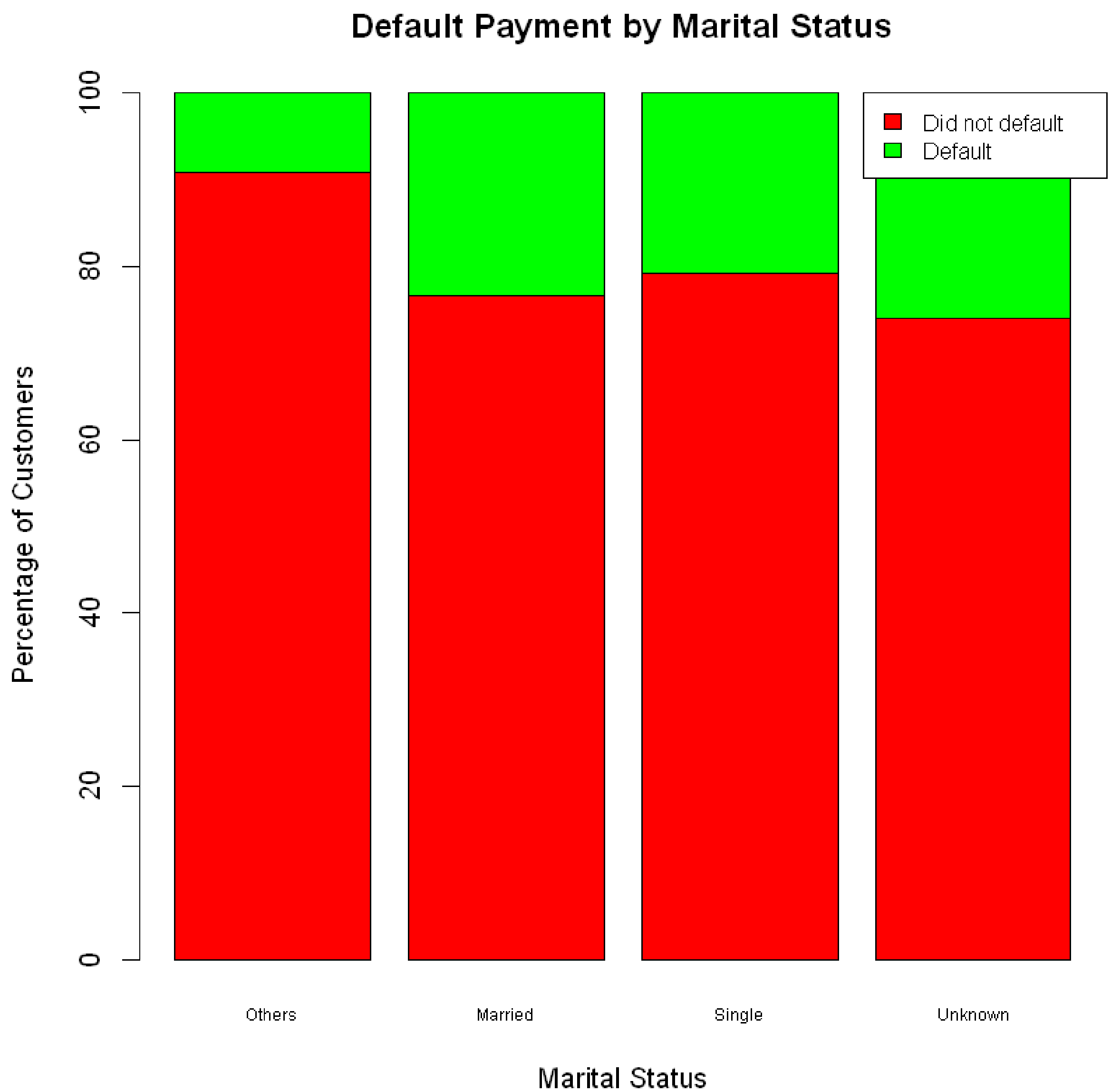

Histogram of data$PAY_AMT3
Frequency
0
5000
10000
15000
20000
25000
30000
0
200000
400000
600000
800000
data$PAY_AMT3


Histogram of data$PAY_AMT4
Frequency
0
5000
10000
15000
20000
25000
30000
0
100000
200000
300000
400000
500000
600000
data$PAY_AMT4

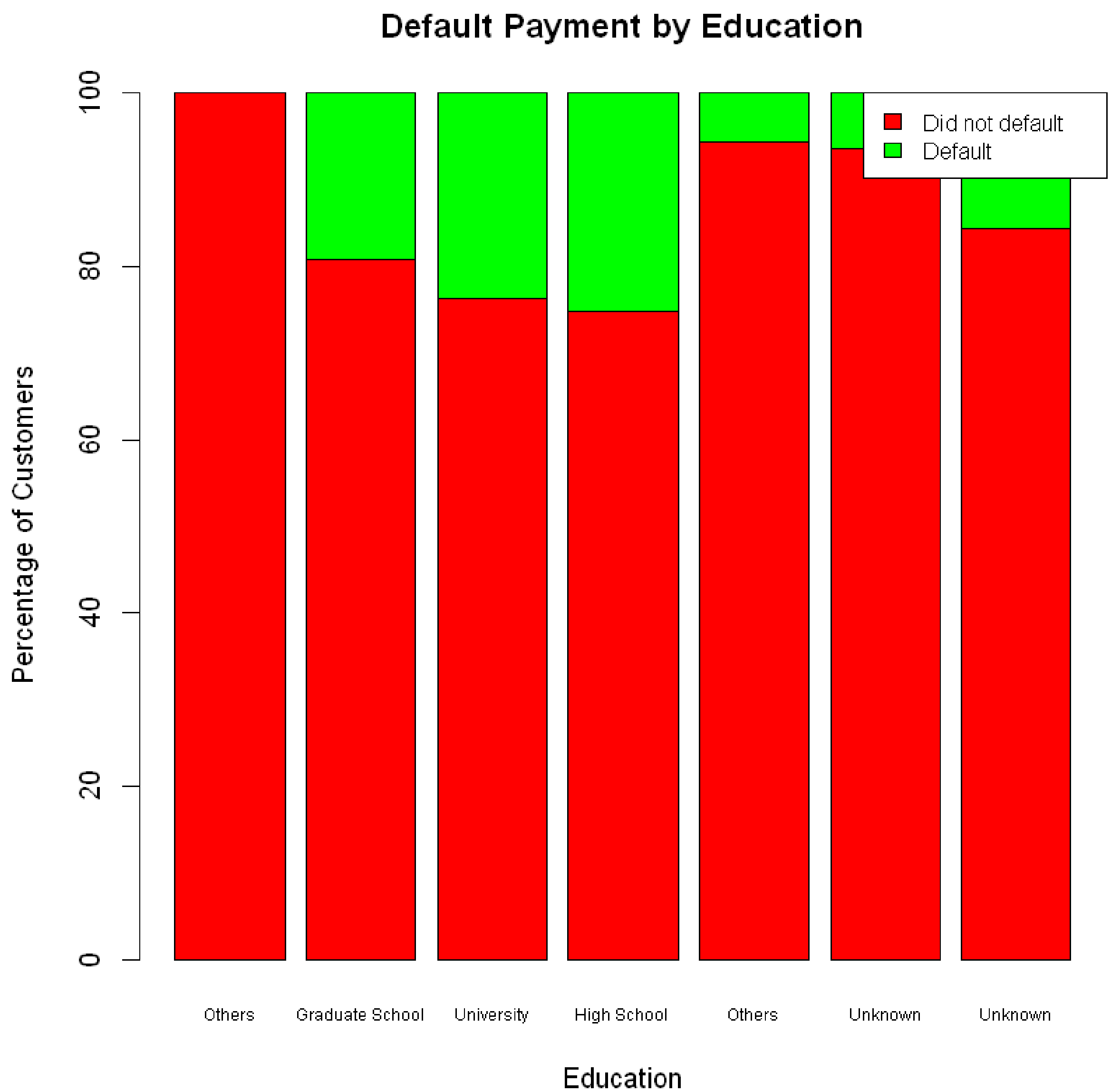

Histogram of data$PAY_AMT5
Frequency
0
5000
10000
15000
20000
25000
0
100000
200000
300000
400000
data$PAY_AMT5


Histogram of data$PAY_AMT6
Frequency
0
5000
10000
15000
20000
25000
30000
0
100000
200000
300000
400000
500000
data$PAY_AMT6


Default Payment by Sex
Percentage of Customers
0
20
40
60
80
100
Did not default
Default
Male
Female
Sex


Default Payment by Marital Status
Percentage of Customers
0
20
40
60
80
100
Did not default
Default
Others
Married
Single
Unknown
Marital Status

	LIMIT_BAL	AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT
LIMIT_BAL	1	0.14	0.29	0.28	0.28	0.29	0.30	0.2
AGE	NA	1.00	0.06	0.05	0.05	0.05	0.05	0.0
BILL_AMT1	NA	NA	1.00	0.95	0.89	0.86	0.83	0.8
BILL_AMT2	NA	NA	NA	1.00	0.93	0.89	0.86	0.8
BILL_AMT3	NA	NA	NA	NA	1.00	0.92	0.88	0.8
BILL_AMT4	NA	NA	NA	NA	NA	1.00	0.94	0.9
BILL_AMT5	NA	NA	NA	NA	NA	NA	1.00	0.9
BILL_AMT6	NA	NA	NA	NA	NA	NA	NA	1.0
PAY_AMT1	NA	NA	NA	NA	NA	NA	NA	N
PAY_AMT2	NA	NA	NA	NA	NA	NA	NA	N
PAY_AMT3	NA	NA	NA	NA	NA	NA	NA	N
PAY_AMT4	NA	NA	NA	NA	NA	NA	NA	N
PAY_AMT5	NA	NA	NA	NA	NA	NA	NA	N
PAY_AMT6	NA	NA	NA	NA	NA	NA	NA	N


Default Payment by Education
Percentage of Customers
0
20
40
60
80
100
Did not default
Default
Others
Graduate School
University
High School
Others
Unknown
Unknown
Education

```
 Attribute =  1 SEX
        class
x             0    1
  10000     218  161
  16000       2    0
  20000     977  559
  30000     821  505
  40000     110   85
  50000    2094  776
  60000     533  216
  70000     479  206
  80000     953  306
  90000     396  142
  100000    623  236
  110000    419  120
  120000    429  131
  130000    443  126
  140000    485  146
  150000    669  140
  160000    388   92
  170000    355   58
  180000    588  122
  190000    158   44
  200000    900  193
  210000    459   73
  220000    327   73
  230000    475   73
  240000    394   76
  250000    232   38
  260000    343   65
  270000    172   23
  280000    346   52
  290000    238   42
  300000    310   59
  310000    216   21
  320000    234   30
  327680      0    1
  330000    112   12
  340000    168   26
  350000    168   26
  360000    465   58
  370000     54    6
  380000    115   13
  390000    133   12
  400000    207   23
  410000     55    9
  420000    130   16
  430000     60    8
  440000     60   12
  450000     88   14
  460000     61    8
  470000     62    9
  480000     63    5
  490000     51    6
  500000    554   52
  510000     15    1
  520000     17    2
  530000      9    1
  540000      5    0
  550000     13    6
```

```
  560000     9    1
  570000     7    0
  580000     9    1
  590000     4    0
  600000    12    3
  610000    11    0
  620000     8    1
  630000     5    1
  640000     7    0
  650000     2    0
  660000     3    0
  670000     3    0
  680000     3    1
  690000     1    0
  700000     8    0
  710000     4    0
  720000     2    1
  730000     2    0
  740000     1    1
  750000     4    0
  760000     1    0
  780000     2    0
  800000     2    0
  1000000    1    0
```

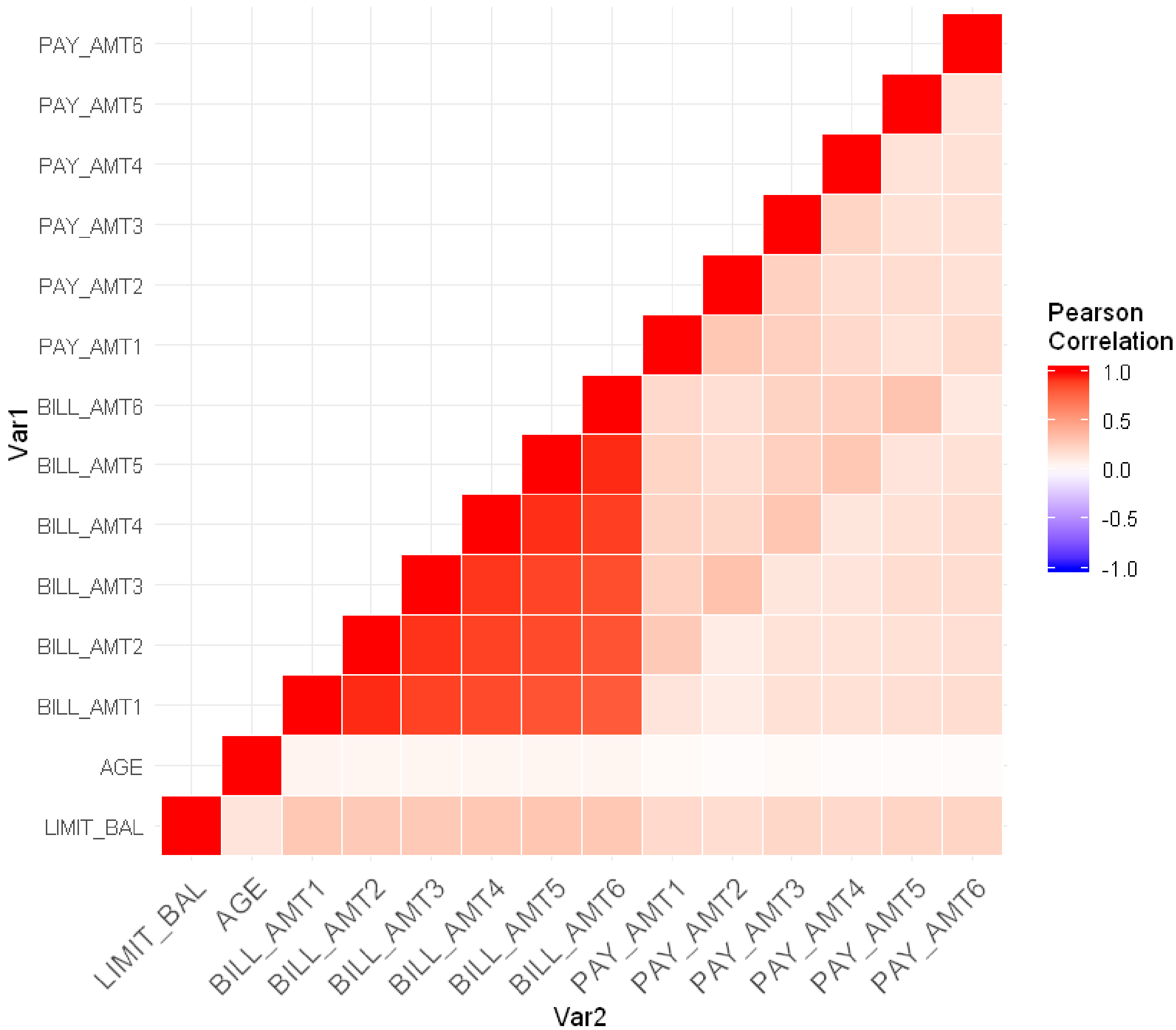
```
Warning message in chisq.test(tbl):
"Chi-squared approximation may be incorrect"
```

```
        Pearson's Chi-squared test

data:  tbl
X-squared = 1101.1, df = 80, p-value < 0.00000000000000022


 Attribute =  2 EDUCATION
   class
x       0     1
  1  7369  2357
  2 11193  2968

        Pearson's Chi-squared test with Yates' continuity correction

data:  tbl
X-squared = 35.512, df = 1, p-value = 0.000000002535


 Attribute =  3 MARRIAGE
   class
x      0    1
  1 6493 1490
  2 8769 2812
  3 2999 1003
  4  301   20

        Pearson's Chi-squared test

data:  tbl
X-squared = 152.62, df = 3, p-value < 0.00000000000000022
```

chi2 stat	
1101.11670	0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000
35.51205	0.0000000025347753076836860447202531831933924877375829964876174926757812500000000000
152.62186	0.0000000000000000000000000000000071639045540447308288954653132662997450097464025020

```
Call:
lda(default.payment.next.month ~ ., data = subsetdf)

Prior probabilities of groups:
        0         1 
0.7770754 0.2229246 

Group means:
           AGE BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6 PAY_AMT2
0 -0.003348431  59642.59  57533.87  53148.96  49584.49  48074.87 7486.807
1  0.011672032  56677.52  54668.63  51234.95  48518.78  47203.52 3837.310
  PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 CREDIT_UTILISATION_RATIO MONTHS_DELAYED
0 6487.120 5994.465 6001.332 6344.609              -0.05894794     -0.2191849
1 3814.601 3589.948 3695.132 3922.166               0.20548201      0.7640395

Coefficients of linear discriminants:
                                      LD1
AGE                       0.03826426483793
BILL_AMT2                -0.00000112282925
BILL_AMT3                 0.00000096432728
BILL_AMT4                -0.00000076338075
BILL_AMT5                 0.00000004268197
BILL_AMT6                -0.00000008857328
PAY_AMT2                 -0.00000217847721
PAY_AMT3                 -0.00000039242316
PAY_AMT4                 -0.00000203295876
PAY_AMT5                 -0.00000197372271
PAY_AMT6                 -0.00000189075233
CREDIT_UTILISATION_RATIO  0.17924140891699
MONTHS_DELAYED            1.04417183737955
```

```
Error in library(leaps): there is no package called 'leaps'
Traceback:

1. library(leaps)
```



PAY_AMT6
PAY_AMT5
PAY_AMT4
PAY_AMT3
PAY_AMT2
PAY_AMT1
BILL_AMT6
BILL_AMT5
BILL_AMT4
BILL_AMT3
BILL_AMT2
BILL_AMT1
AGE
LIMIT_BAL
Var1
LIMIT_BAL
AGE
BILL_AMT1
BILL_AMT2
BILL_AMT3
BILL_AMT4
BILL_AMT5
BILL_AMT6
PAY_AMT1
PAY_AMT2
PAY_AMT3
PAY_AMT4
PAY_AMT5
PAY_AMT6
Var2
Pearson Correlation
1.0
0.5
0.0
-0.5
-1.0

In [ ]:

In [ ]: