Introduction

A charitable organization wishes to develop a machine learning model to improve the cost effectiveness of their direct marketing campaigns to previous donors. According to their recent mailing records, the typical overall response rate is 10%. Out of those who respond (donate) to the mailing, the average donation is $14.50. Each mailing costs $2.00 to produce and send; the mailing includes a gift of personalized address labels and assortment of cards and envelopes. It is not cost effective to mail everyone because the expected profit from each mailing is $14.50 x 0.10 – $2 = -$0.55.

There are two objects of this assignment. Firstly, to develop a classification model using data from the most recent campaign that can effectively capture likely donors so that the expected net profit is maximized. Secondly, develop a prediction model to predict donation amounts for donors.

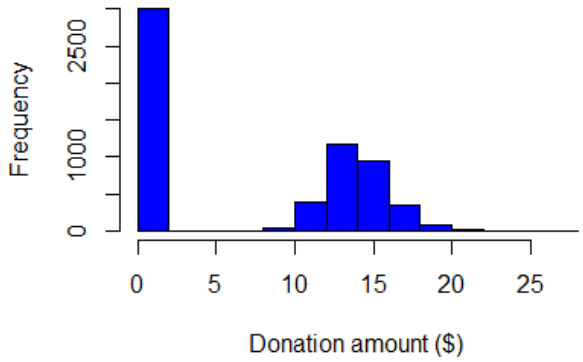
Prior to building the classification and predictive models, the data will be explored to identify missing values, outliers and potential variable transformations. The transformed data set will then be used to build models using a variety of machine learning techniques. The models will be compared against each other and the best model will be selected based on model validation metrics.

1.0 Data Exploration

The charity dataset contains 24 variables and has been split into 3984 training observations, 2018 validation observations, and 2007 test observations. The variables in the data set can be viewed in Table 1 as well as a brief definition for each variable. The data set is a mix of categorical and continuous variables.

Table 1 - Variables in the charity data set.

VARIABLE	DEFINITION
ID Number	ID. Do NOT use this as a predictor variable in any model
REG1, REG2, REG3, REG4	There are five geographic regions; only four are needed for analysis since if a potential donor falls into none of the four he or she must be in the other region.
HOME	1 = homeowner, 0 = not a homeowner
CHLD	Number of children
HINC	Household income (7 categories)
GENF	Gender (0 = Male, 1 = Female)
WRAT	Wealth Rating (Wealth rating uses median family income and population statistics from each area to index relative wealth within each state. The segments are denoted 0-9, with 9 being the highest wealth group and 0 being the lowest.
AVHV	Average Home Value in potential donor's neighborhood in $ thousands
INCM	Median Family Income in potential donor's neighborhood in $ thousands
INCA	Average Family Income in potential donor's neighborhood in $ thousands
PLOW	Percent categorized as “low income” in potential donor's neighborhood
NPRO	Lifetime number of promotions received to date
TGIF	Dollar amount of lifetime gifts to date
LGIF	Dollar amount of largest gift to date
RGIF	Dollar amount of most recent gift
TDON	Number of months since last donation
TLAG	Number of months between first and second gift
AGIV	Average dollar amount of gifts to date
DONR	Classification Response Variable (1 = Donor, 0 = Non-donor)
DAMT	Prediction Response Variable (Donation Amount in $).



Weighted sampling has been used, over-representing the responders so that the training and validation samples have approximately equal numbers of donors and non-donors, around 3000 each. The histogram shown in Figure 1 shows the distribution of the donation amounts. It can be seen that when a donation is made, the average amount is approximately $14.5.

Figure 1 – Histogram of donation amount frequency.

1.1 Summary Statistics

Table 2 shows the summary statistics for the continuous variables in the data set. The exploratory data analysis revealed that there were no missing values however a number of variables had highly skewed distributions.

Table 2 - Summary statistics for the numerical variables.

Variable	n	miss	mean	sd	skew	krts	min	max	IQR
chld	8009	0	1.72	1.4	0.27	-0.8	0	5	3
hinc	8009	0	3.91	1.47	0.01	-0.09	1	7	2
wrat	8009	0	6.91	2.43	-1.35	0.79	0	9	3
avhv	8009	0	182.65	72.72	1.54	4.49	48	710	84
incm	8009	0	43.47	24.71	2.05	8.32	3	287	27
inca	8009	0	56.43	24.82	1.94	7.88	12	305	28
plow	8009	0	14.23	13.41	1.36	1.89	0	87	17
npro	8009	0	60.03	30.35	0.31	-0.62	2	164	46
tgif	8009	0	113.07	85.48	6.55	107.61	23	2057	74
lgif	8009	0	22.94	29.95	7.82	110.48	3	681	15
rgif	8009	0	15.66	12.43	2.63	13.94	1	173	13
tdon	8009	0	18.86	5.78	1.10	2.13	5	40	7
tlag	8009	0	6.36	3.70	2.42	8.42	4	34	3
agif	8009	0	11.681	6.567	1.78	6.02	1	72	8

1.2 Data Preparation

Boxplots and histograms were used to visually explore the distributions of the variables. Transformed variables were created using a combination of indicator variables and the Box-Cox transformation procedure. Justification for the transformations can be seen in Appendix 1. Table 3 shows a list of the additional transformed variables that were created for use in the regression and classification modelling.

Table 3 - List of treated variables to be used.

G0_CHLD	G0_PLOW	T_INCM	T_LGIF	G2_TLAG
G3_CHLD	G4_PLOW	T_INCA	T_RGIF	G7_TLAG
G2_WRAT	G21_PLOW	T_NPRO	TDON	
G7_WRAT	T_AVHV	T_TGIF	T_AGIF	

1.3 Cross Validation

Prior to modelling the variables were standardized to have zero mean and a standard deviation of one. The data set was also split into training, validation and test data sets. The number of records in each of these data sets is shown in Table 4.

Table 4 - Count of records in the training and test data sets.

Dataset	Number of records
Training	3984
Validation	2018
Test	2007

2.0 Classification Modelling

The following section describes the different classification modelling techniques that were used to predict the expected net profit from a mailing campaign. The techniques include linear discriminate analysis, quadratic discriminant analysis, logistic regression, random forest, k-nearest neighbor and support vector machines.

The profit calculation used $14.50 as the average donor amount and each mailing cost $2.00. The following steps were followed to calculate the maximum profit for each classifier type.

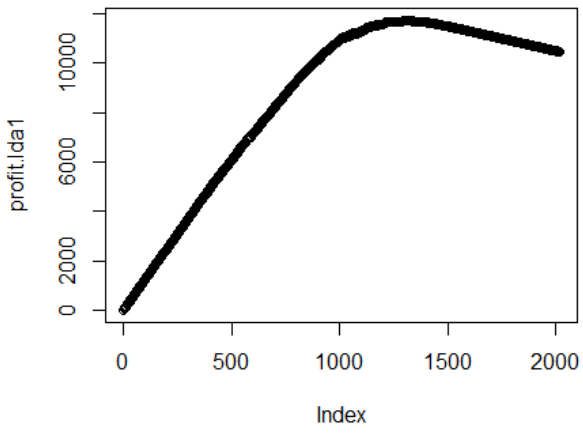
1. Calculate the posterior probabilities for the validation dataset.
2. Sort DONR in order of the posterior probabilities from highest to lowest.
3. Calculate the cumulative sum of (14.5 x DONR – 2) as you go down the list.
4. Then, find the maximum of this profit function.

2.1 Linear Discriminant Analysis

The first model was created using linear discriminant analysis. Multiple models were fit using different variations of predictor variables. When tested on the validation set, the model shown in Table 5 resulted in the highest profit. The maximum profit of $11,732 occurred when the number of mailings equalled 1297.

Table 5 – Linear Discriminant Analysis model that resulted in maximized profit.

```
model.lda1 <- lda(donr ~ reg1 + reg2 +  reg4 + home + G0_chld + G3_chld +
                         I(hinc^2) + genf + wrat + T_avhv + incm + inca +
                         G0_plow  + G4_plow  + G21_plow + npro + tgif +
                         rgif + tdon + I(tdon^2) + G2_tlag + G7_tlag,
                         data.train.std.c)
```



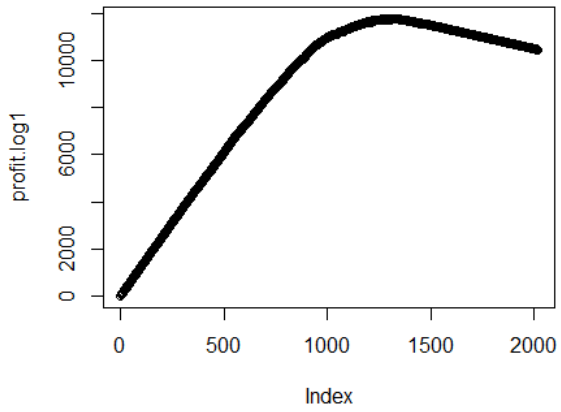

```
                    c.valid
chat.valid.lda1       0   1
                  0 710  11
                  1 309 988

n.mail.valid = 309+988 = 1297
profit=14.5*988-2*1297 =$11732
```

2.2 Logistic Regression

The second model was created using logistic regression. Multiple models were fitted including the full model which used all of the variables to models with different variations of the predictor variables. When tested on the validation set, the model shown in Table 6 resulted in a maximum profit of $11,772.5 with the number of mailings being 1313.

Table 6 – Logistic regression model that resulted in maximized profit.

```
model.log1 <- glm(donr ~ reg1 + reg2 + reg4 + home + G0_chld + G3_chld +
                         I(hinc^2) + genf + wrat + T_avhv + incm + inca +
                         G0_plow  + G4_plow  + G21_plow + npro + tgif +
                         rgif + tdon + I(tdon^2) + G2_tlag + G7_tlag,
                         data.train.std.c, family=binomial("logit"))
```



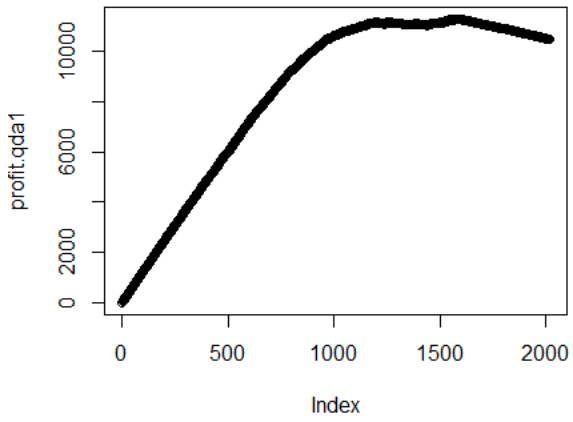

```
                    c.valid
chat.valid.log1       0   1
                  0 699   6
                  1 320 993

n.mail.valid = 320+993 = 1313
profit=14.5*993-2*1313 =$11772.5
```

2.3 Quadratic Discriminant Analysis

The third model was created using quadratic discriminant analysis. Several models were fitted using a variety of predictor variables. When tested on the validation set, the model shown in Table 7 resulted in a maximum profit of $11,269.5 with the number of mailings being 1579.

Table 7 – Quadratic Discriminant Analysis model that resulted in maximized profit.

```
model.qda1 <- qda(donr ~ reg1 + reg2 +  reg4 + home + G0_chld + G3_chld +
                         hinc + I(hinc^2) +  wrat + incm + T_inca + plow +
                         tgif +  tdon + I(tdon^2) + G2_tlag + G7_tlag +
                         agif, data.train.std.c)
```

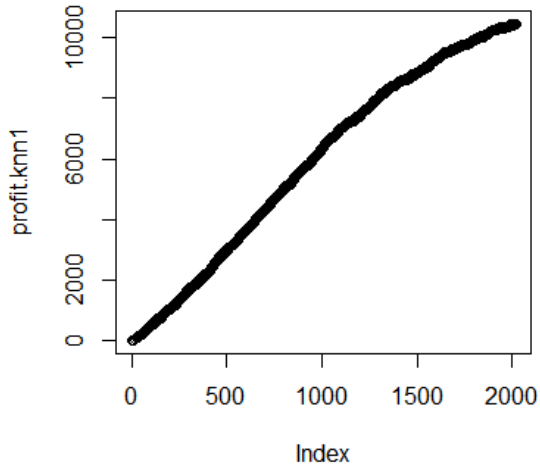



```
                     c.valid
chat.valid.qda1        0   1
                   0 435   4
                   1 584 995

n.mail.valid = 584+995 = 1579
profit=14.5*995-2*1579 =$11269.5
```

2.4 K-Nearest Neighbor Clustering

The K Nearest Neighbor (KNN) model was the worst performing classifier. Several KNN models were fitted using a variety of different predictor variable combinations. The model shown in Table 8 used all of the untransformed predictor variables and had 12 clusters. The lowest posterior probability that was calculated was 0.5 and the cutoff point that maximized profit was also calculated to be 0.5. This made the classification quite inaccurate as seen in the confusion matrix in Table 8.

Table 8 – K-Nearest Neighbor Clustering model (K=12).



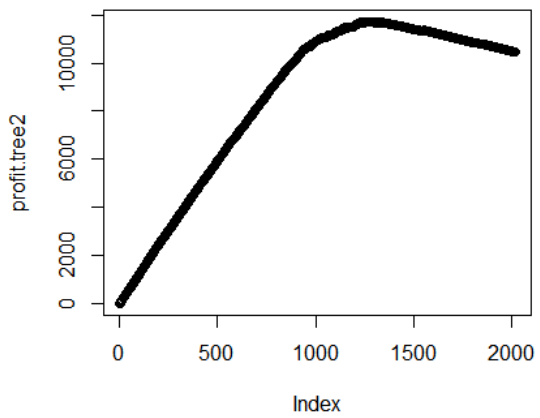

```
                    c.valid
chat.valid.knn1       0   1
              0      90  31
              1     929 968


n.mail.valid = 929+968 = 1897
profit=14.5*968-2*1897 =$10242
```

2.5 Bagged Random Forest

The variable TREE_DONR was created to change the DONR numerical variable into a categorical variable to enable the use of the randomForest function for building a classification model (1 = YES, 0 = NO). The untransformed predictor variables were used as decision trees can handle non-linear patterns. The model shown in Table 9 resulted in a maximum profit of $11,690 when the number of mailings equalled 1260.

Table 9 – Bagged random forest model that resulted in maximized profit.

```
randomForest(formula = tree_donr ~ reg1 + reg2 + reg3 + reg4 + home + chld +
                                   hinc + genf + wrat + avhv + incm + inca +
                                   plow + npro + tgif + lgif + rgif + tdon +
                                   tlag + agif, data = data.train.std.c,
                                   mtry = 20, importance = TRUE)

Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 20
```



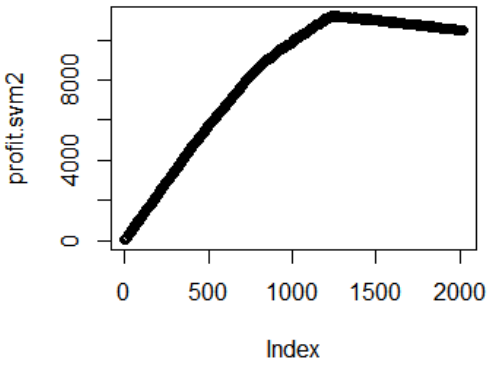

```
                     c.valid
chat.valid.tree2       0   1
               0     739  19
               1     280 980


n.mail.valid = 280+980 = 1260
14.5*980-2*1260 =$11690
```

2.6 Support Vector Machine Model

The final classification model used the support vector machine modelling technique. A radial kernel was selected. The gamma value of 0.5 and cost of 10 was selected based off tuning results. The model shown in Table 10 resulted in a maximised profit of $11,182.5 when the number of mailings equalled 1260.

Table 10 – Support Vector Machine model.

```
svm(formula = tree_donr ~ reg1 + reg2 + reg3 + reg4 + home + chld + hinc +
                          genf + wrat + avhv + incm + inca + plow + npro +
                          tgif + lgif + rgif + tdon + tlag + agif,
                          data = data.train.std.c, kernel = "radial",
                          gamma = 0.5, cost = 10, probability = TRUE)

SVM-Type:  C-classification
SVM-Kernel:  radial
       cost:  10
      gamma:  0.5
Number of Support Vectors:  3755
                 c.valid
chat.valid.svm2     0    1
                0  704   54
                1  315 945

n.mail.valid = 315+945 = 1260
profit=14.5*945-2*1260 =$11182.5
```



2.7 Classification Model Comparisons

This section of the report compares the model validation metrics of the six different classification models with the aim of selecting the best model for predicting the expected net profit from a mailing campaign. Table 11 shows that the logistic regression model resulted in the highest profit of $11,772.5.

Table 11 – Profit comparison table.

Model	Number of Mailings	Profit
Linear Discriminant Analysis	1297	$11,732
Logistic Regression	1313	$11,772.5
Quadratic Discriminant Analysis	1579	$11,269.5
KNN Clustering	1897	$10,242
Bagged Random Forest	1260	$11,690
Support Vector Machine	1260	$11182.5

3.0. Regression Modelling

The following sections describe the different regression modelling techniques that were used to predict the expected donation amount if a donation was made. The regression techniques include least squares regression, principal component regression, general additive model regression and random forests.

3.1 Least Squares Regression

The first model was created using least squares regression. Multiple models were fitted using a variety of different predictor variable combinations. Variables were added and removed based on their significance values and the effect that they had on the validation set mean squared error. The model shown in Table 12 resulted in a mean squared error of 1.50.

Table 12 – Linear regression model with minimised mean squared error.

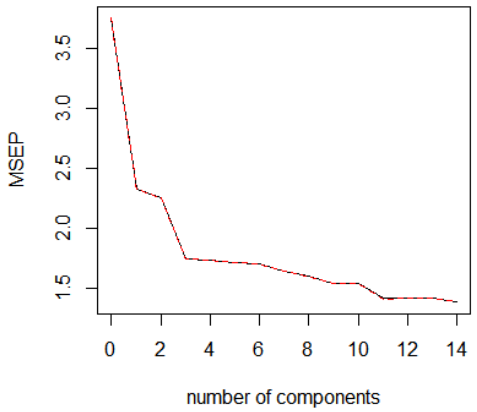
```
lm(formula = damt ~ reg2 + reg3 + reg4 + home + G0_chld + G3_chld +
    hinc + G2_wrat + G7_wrat + incm + T_tgif + T_lgif + T_rgif +
    T_agif, data = data.train.std.y)
Residuals:
    Min      1Q  Median      3Q     Max
-3.0187 -0.7604 -0.1675  0.4961 10.0846
Coefficients:
            Estimate Std. Error t value             Pr(>|t|)
(Intercept) 14.18940    0.04129 343.638 < 0.0000000000000002
reg2        -0.07233    0.02851  -2.537               0.0113
reg3         0.33982    0.03224  10.539 < 0.0000000000000002
reg4         0.66947    0.03306  20.248 < 0.0000000000000002
home         0.21888    0.05571   3.929  0.00008828798407796
G0_chld     -0.42156    0.02891 -14.580 < 0.0000000000000002
G3_chld     -0.42885    0.05046  -8.498 < 0.0000000000000002
hinc         0.51287    0.03650  14.050 < 0.0000000000000002
G2_wrat      0.71435    0.08951   7.980  0.00000000000000244
G7_wrat      0.51513    0.09370   5.498  0.00000004342968604
incm         0.16699    0.02459   6.790  0.00000000001477220
T_tgif       0.20789    0.02720   7.644  0.00000000000003271
T_lgif       0.46156    0.06250   7.385  0.00000000000022411
T_rgif       0.43749    0.05200   8.413 < 0.0000000000000002
T_agif       0.34913    0.04977   7.016  0.00000000000313265

Residual standard error: 1.169 on 1980 degrees of freedom
Multiple R-squared:  0.638,    Adjusted R-squared:  0.6354
F-statistic: 249.3 on 14 and 1980 DF,  p-value: < 0.00000000000000022
```

3.2 Principal Component Regression

The second model was created using principal component regression. Multiple models were fitted using various combinations of the predictor variables. The model shown in Table 13 used the predictor variables from the least square regression in Section 3.1. Using eleven principal components, the model shown in Table 13 resulted in a mean squared error of 1.57 when tested on the validation data set.

Table 13 – Model built using principal component regression.

```
Principal component regression, fitted with the singular value decomposition
algorithm. Cross-validated using 10 random segments.

pcr(formula = damt ~ reg2 + reg3 + reg4 + home + G0_chld + G3_chld + hinc +
                     G2_wrat + G7_wrat + incm + T_tgif + T_lgif + T_rgif +
                     T_agif, data = data.train.std.y,
                     scale = TRUE, validation = "CV")

TRAINING: % variance explained
      1 comps  2 comps  3 comps  4 comps
X       19.0    33.04    43.78    52.00
damt    37.9    40.16    53.62    53.88
      5 comps  6 comps  7 comps  8 comps
X       59.91    67.68    75.03    82.10
damt    54.60    54.89    56.85    57.67
      9 comps  10 comps  11 comps  12 comps
X       88.90     93.56      97.2     98.84
damt    59.46     59.58      62.9     62.92
      13 comps  14 comps
X        99.66     100.0
damt     62.92      63.8
```

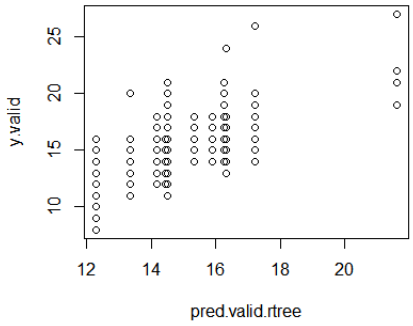


3.3 Decision Tree Regression

The third regression model was created using a decision tree. Based on the cross validation results it was decided to leave it unpruned. Due to the ability of decisions trees to handle non-linear patterns, the untransformed variables were used as predictor variables. The first model was fit using all the variables. Variables that were not included in the tree were then removed. The final model is shown in Table 14. The model resulted in a mean squared error of 2.24 when tested on the validation set.

Table 14 – Decision tree model that minimised validation mean squared error.

```
Regression tree:
tree(formula = damt ~ reg3 + reg4 + chld +
                      lgif + rgif,
               data = data.train.std.y)

Number of terminal nodes:  11
Residual mean deviance:  1.917 = 3802 / 1984
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3.5640 -0.8840 -0.1951  0.0000  0.7943  8.4360
```

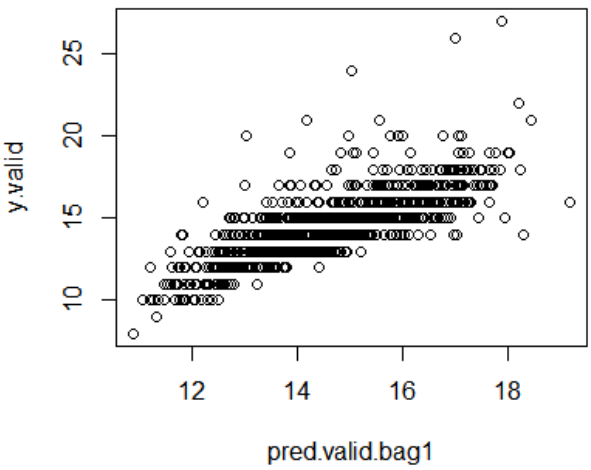


3.4 Random Forest Bagged

The fourth regression model was created using a special case of a random forest, bagging. This is where the number of variables tried at each split is equal to the number of predictor variables (m = p). The model shown in Table 15 resulted in a mean squared error of 1.70 when tested on the validation set.

Table 15 – Bagged random forest model.

```
model.bag1 = randomForest(damt ~ reg1 + reg2 + reg3 + reg4 + home + chld +
                          hinc + genf + wrat + avhv + incm + inca + plow +
                          npro + tgif + lgif + rgif + tdon + tlag + agif,
                         data = data.train.std.y, mtry=20, importance= TRUE)
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 20
Mean of squared residuals: 1.49039
% Var explained: 60.22
Variable importance
        %IncMSE IncNodePurity
reg1  0.2633663      20.33080
reg2  8.0343994      55.19952
reg3 33.4657779     181.40927
reg4 84.3934526     675.00932
home -1.6118116      30.07290
chld 47.2039718     471.52268
hinc 32.9207302     296.04837
genf  0.5405559      31.95948
wrat 18.8616004     291.12696
avhv 11.0130643     221.83359
incm 19.0112748     215.06867
inca 14.0543589     182.82998
plow 16.3859923     182.49231
npro  9.9157805     228.56402
tgif 17.2194162     272.03600
lgif 33.9670032    1132.73414
rgif 39.1307943    1865.73649
tdon  1.4631342     184.83061
tlag -0.6574439     135.64068
agif 28.6521195     632.97851
```

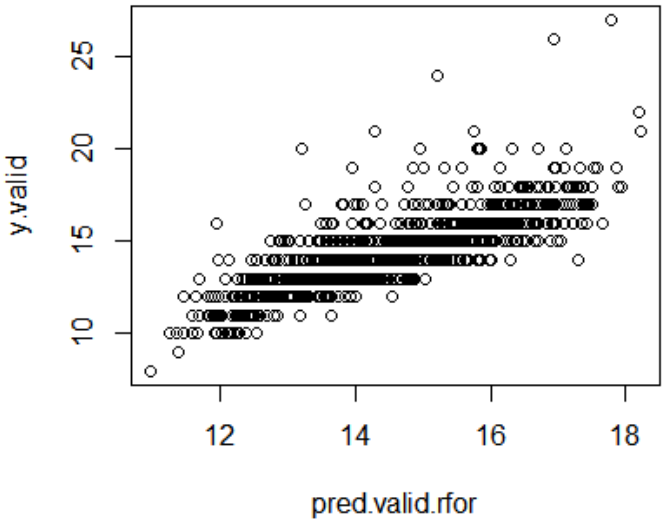


3.5 Random Forest (mtry=5)

The fifth model was created by changing the number of variables tried at each split to five. The model shown in Table 16 resulted in a mean squared error of 1.66 when tested on the validation set.

Table 16 – Random forest model with mtry=five.

```
model.rfor = randomForest(damt ~ reg1 + reg2 + reg3 + reg4 + home + chld +
                            hinc + genf + wrat + avhv + incm + inca + plow +
                            npro + tgif + lgif + rgif + tdon + tlag + agif,
                          data = data.train.std.y, mtry=20, importance= TRUE)

Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 5
Mean of squared residuals: 1.486771
% Var explained: 60.32

Variable importance
         %IncMSE IncNodePurity
reg1  6.58478636      50.02560
reg2 15.25785342     141.99263
reg3 24.56062352     128.37134
reg4 54.98166337     549.51273
home  0.71033550      33.13847
chld 40.63597864     420.42684
hinc 25.44315940     256.08164
genf -0.28720264      38.92270
wrat 12.27594058     225.54246
avhv  7.90741267     252.86770
incm 15.30621623     245.51201
inca 13.46770491     234.31033
plow 16.23402695     219.27519
npro  9.96946566     253.65999
tgif 10.65493888     291.35826
lgif 30.72030574    1204.71031
rgif 31.93147259    1182.08415
tdon  1.28471391     192.63774
tlag -0.01795471     157.61132
agif 28.59808095    1097.47705
```



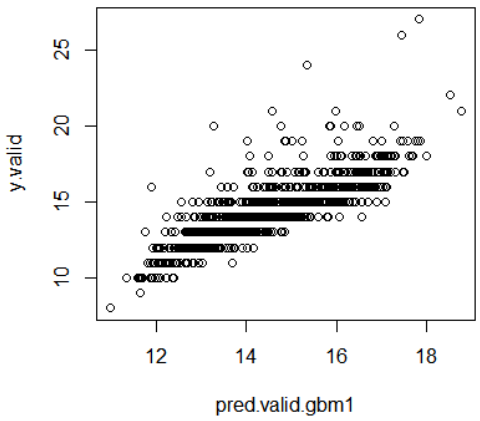
3.6 Boosted Regression Tree

The sixth regression model used a boosted regression tree to fit the data set. The boosted method allows the interaction depth and shrinkage parameters to be tuned. The model shown in Table 17 resulted in a mean squared error value of 1.54 when tested on the validation set.

Table 17 – Boosted regression tree model.

```
model.gbm1 <- gbm(damt ~ reg2 + reg3 + reg4 + chld + hinc + wrat + incm +
                         plow + tgif + lgif + rgif + agif,
              data = data.train.std.y, distribution = "gaussian",
              n.trees=5000, interaction.depth=4, shrinkage=0.001, verbose=F)

A gradient boosted model with gaussian loss function.
5000 iterations were performed.
There were 12 predictors of which 12 had non-zero influence.
```

```
     var   rel.inf
rgif rgif 22.902218
lgif lgif 18.523959
agif agif 15.148593
reg4 reg4 12.540163
chld chld  9.439035
hinc hinc  5.458290
wrat wrat  4.900301
reg3 reg3  3.388014
tgif tgif  2.389400
incm incm  1.961664
plow plow  1.772677
reg2 reg2  1.575687
```



3.7 General Additive Model

The seventh regression model fitted was a general additive model (GAM). The variables from the least squares regression were chosen as a base for building the GAM. Multiple models were fitted and the model shown in Table 18 resulted in a mean squared error of 1.49 when tested on the validation set.

Table 18 – General additive model.

```
Call: gam(formula = damt ~ reg1 + reg2 + reg4 + home + G0_chld + G3_chld +
                           s(hinc, 3) + s(wrat, 5) + T_tgif + T_lgif +
                           T_rgif + s(agif, 2), data = data.train.std.y)

Deviance Residuals:
    Min      1Q  Median      3Q     Max
-2.4550 -0.7553 -0.1809  0.4870 10.2893

(Dispersion Parameter for gaussian family taken to be 1.3798)

    Null Deviance: 7474.744 on 1994 degrees of freedom
Residual Deviance: 2725.104 on 1975 degrees of freedom
AIC: 6325.73
```

3.8 Regression Model Comparisons

This section of the report compares the model validation metrics of the seven different regression models with the aim of selecting the best model for predicting the expected gift amounts from donors. Table 19 shows the mean squared error values for training and validation data sets for each of the regression models. The decision tree had the highest validation error. The error was improved by using bagging and boosting. The least squares regression and the general additive model had the lowest MSE and were approximately the same. The least squares regression was chosen to predict the donor amount as it is less complex than the general additive model.

Table 19 - Comparison of the regression model mean squared error values.

Model	MSE Train	MSE Valid
Least Squares	1.35631	1.495806
Principal Components	-	1.569526
Decision Tree	1.906007	2.241075
Random Forest Bagged	1.49039	1.704569
Random Forest (mtry=5)	1.486771	1.655456
Boosted Regression Tree	1.217787	1.535696
General Additive Model	1.365967	1.48537

4.0 Conclusion

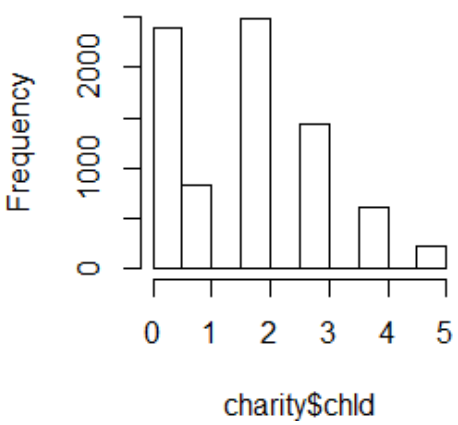
The least squares regression model was selected to predict the donation amounts in the test dataset. The logistic regression model was selected to classify the donation responses in the test dataset. In order to account for the weighted sampling, the optimal mailing rate in the validation data needed to be adjusted before applying it to the test data. The optimal test mailing rate was adjusted using the below steps. The example uses the value of 0.7 as the optimal validation mailing rate.

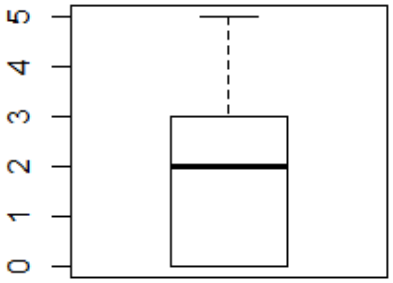
1. Adjust this mailing rate using 0.7/(0.5/0.1) = 0.14.
2. Adjust the "non-mailing rate" using (1 – 0.7)/((1 – 0.5)/(1 – 0.1)) = 0.54.
3. Scale the mailing rate so that it is a proportion: 0.14/(0.14 + 0.54) = 0.206.
4. The optima test mailing rate is thus 0.206.

Based on the logistic regression model, the 344 highest posterior probabilities were selected to be mailed to. The predictions on the test set were exported to a csv file titled MKD.csv.

Appendix 1 – Data Exploration

CHLD Variable

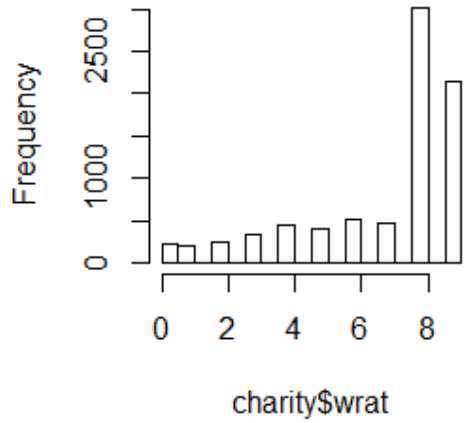


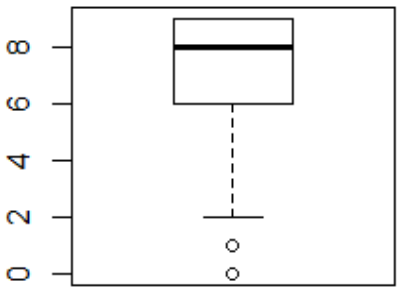


Transformed by creating two indicator variables due to large proportion of zero children. Indicator variables created are:

Indicator Variable	Count
G0_chld (chld > 0 & chld <=3)	4764
G3_chld (chld > 3)	844
Remaining (intercept)	2401

WRAT Variable

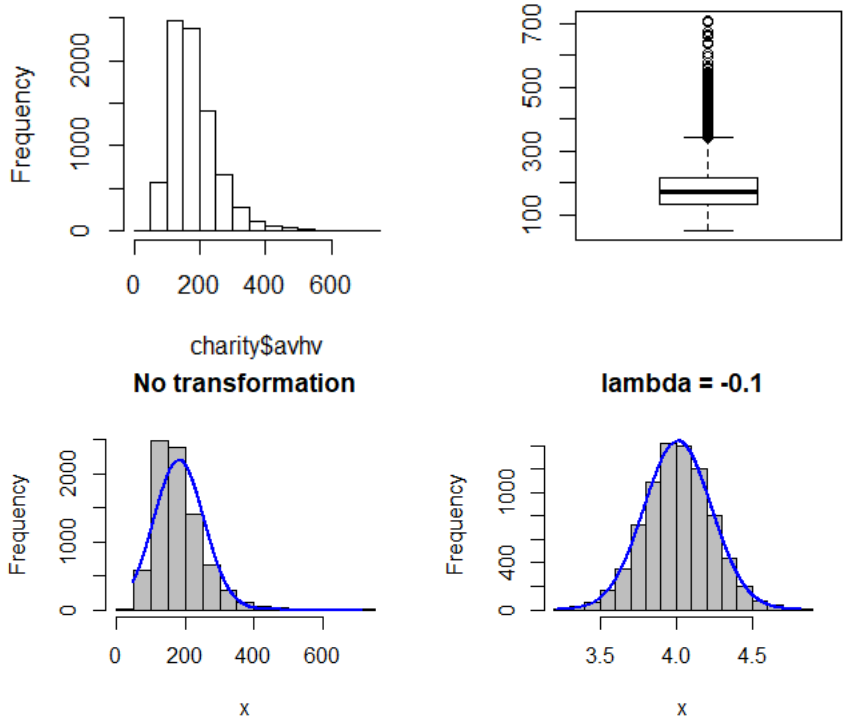




Transformed by creating two indicator variables due to large skew.

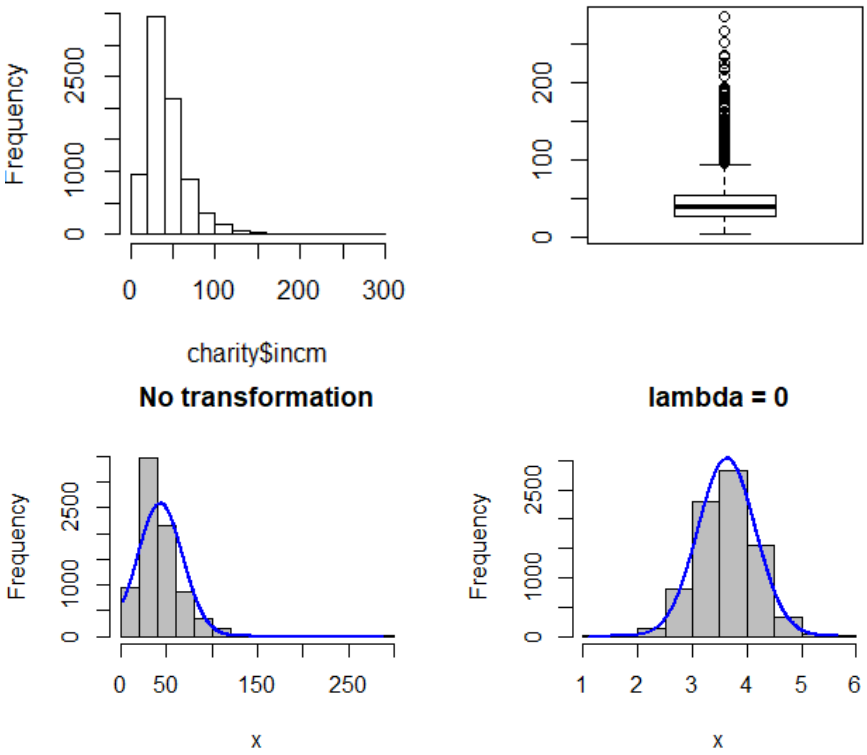
Indicator Variable	Count
G2_wrat (wrat > 2 & wrat <=7)	2180
G7_wrat (wrat >7)	5159
Remaining (intercept)	670

AVHV Variable



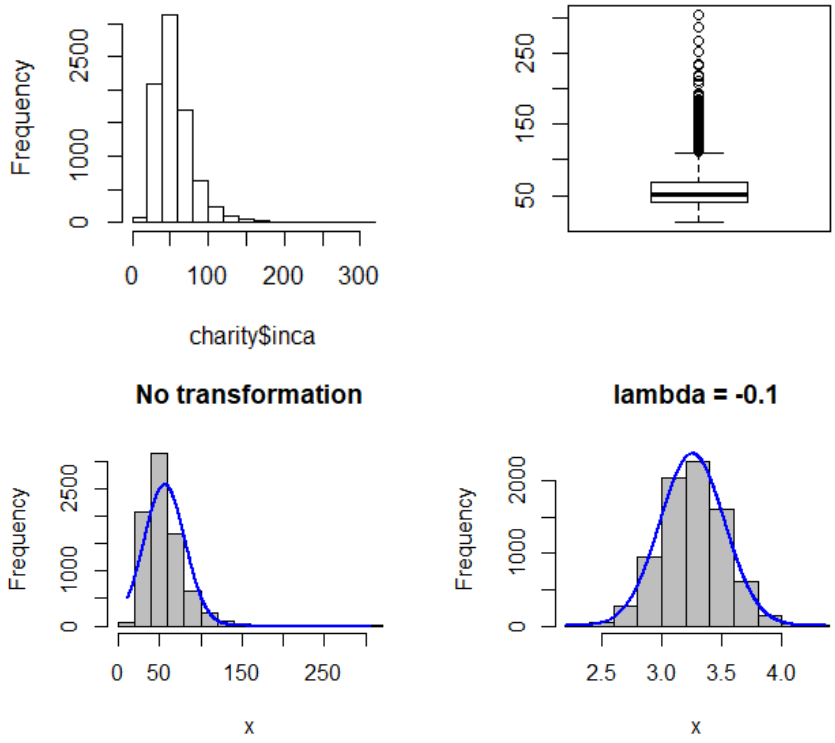
Using the box-cox transformation procedure a lambda value of -0.1 was selected. Above graph shows the change in distribution. The transformed variable was named T_avhv.

INCM Variable



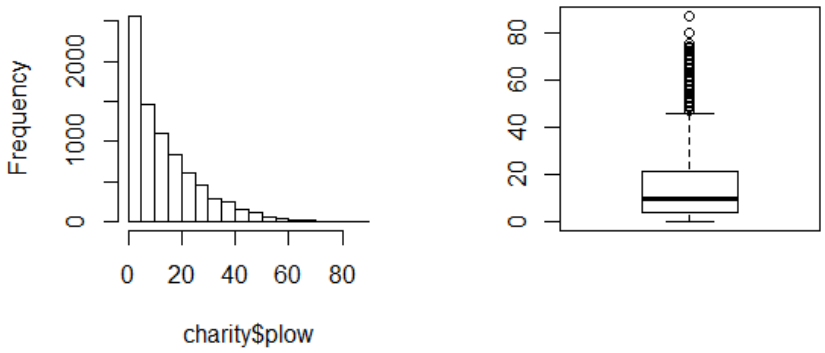
Using the box-cox transformation procedure a lambda value of 0 was selected (log transformation). Above graph shows the change in distribution. The transformed variable was named T_incm.

INCA Variable



Using the box-cox transformation procedure a lambda value of -0.1 was selected. Above graph shows the change in distribution. The transformed variable was named T_inca.

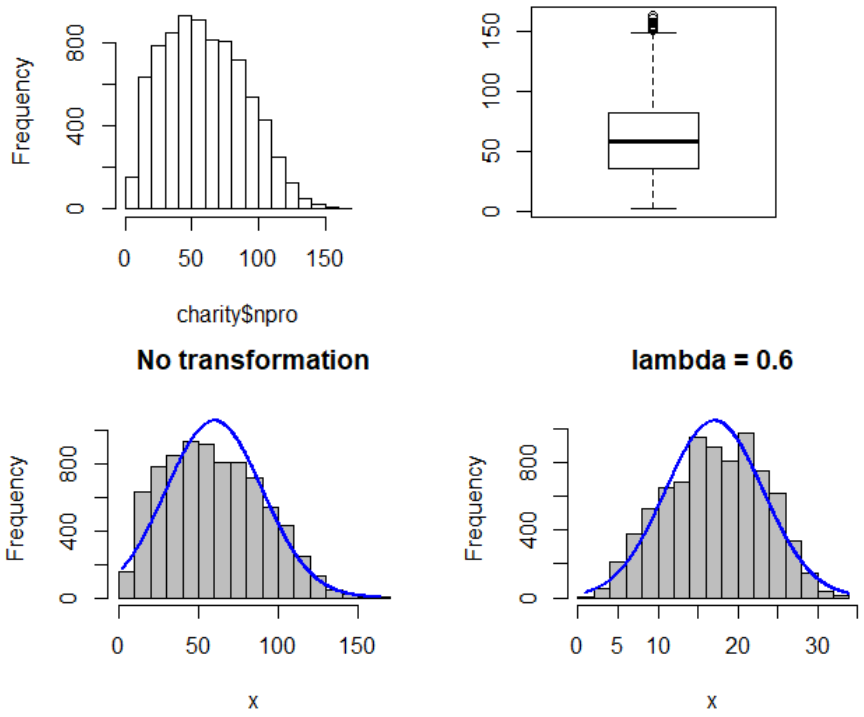
PLOW Variable



Transformed by creating three indicator variables due to large proportion of zero children. Indicator variables created are:

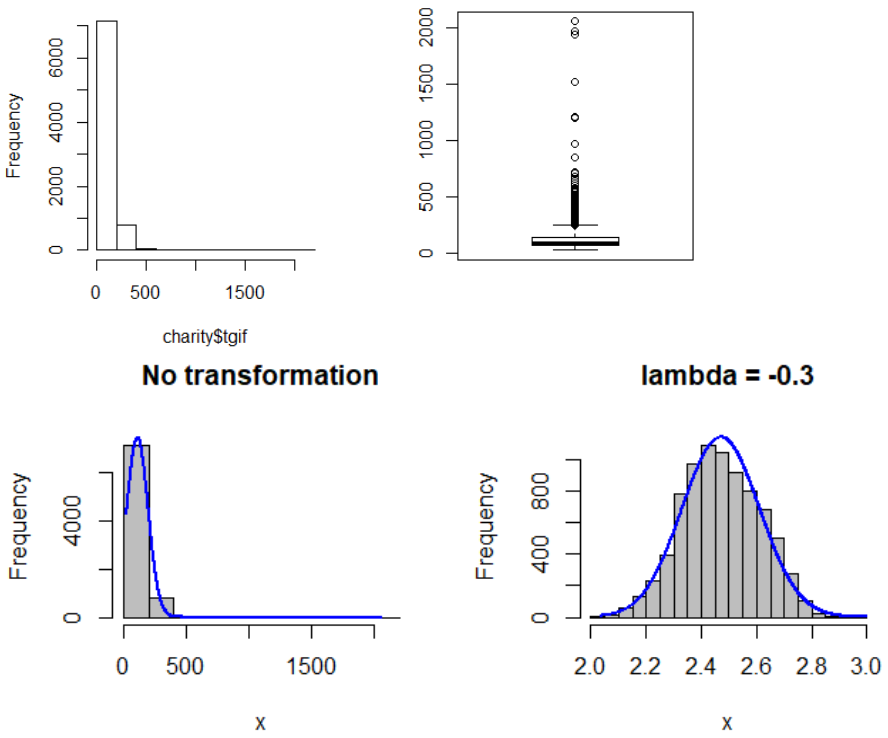
Indicator Variable	Count
G0_plow (plow > 0 & plow <=4)	1570
G4_plow (plow > 4 & plow <=21	3899
G21_plow (plow >21)	1904
Remaining (intercept)	636

NPRO Variable



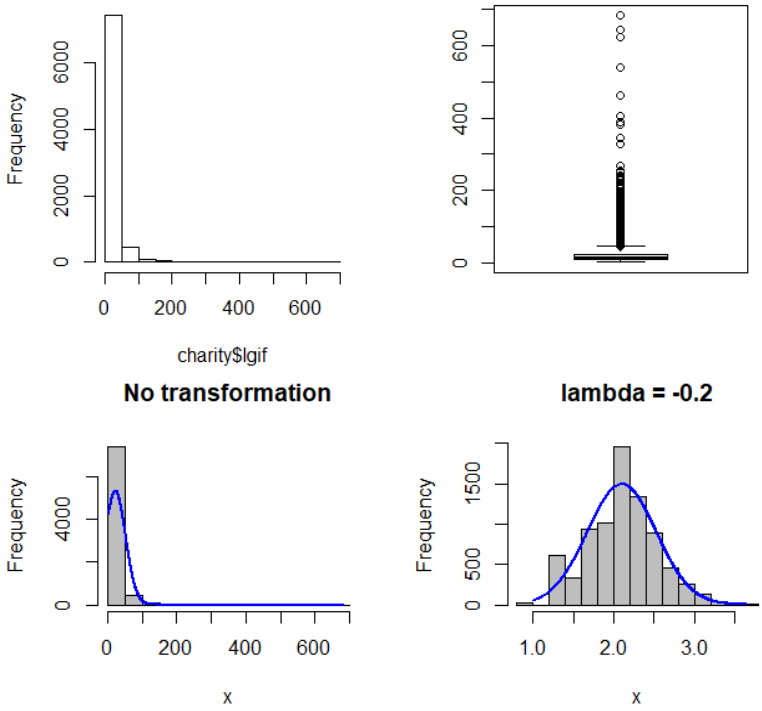
Using the box-cox transformation procedure a lambda value of 0.6 was selected. Above graph shows the change in distribution. The transformed variable was named T_npro.

TGIF Variable



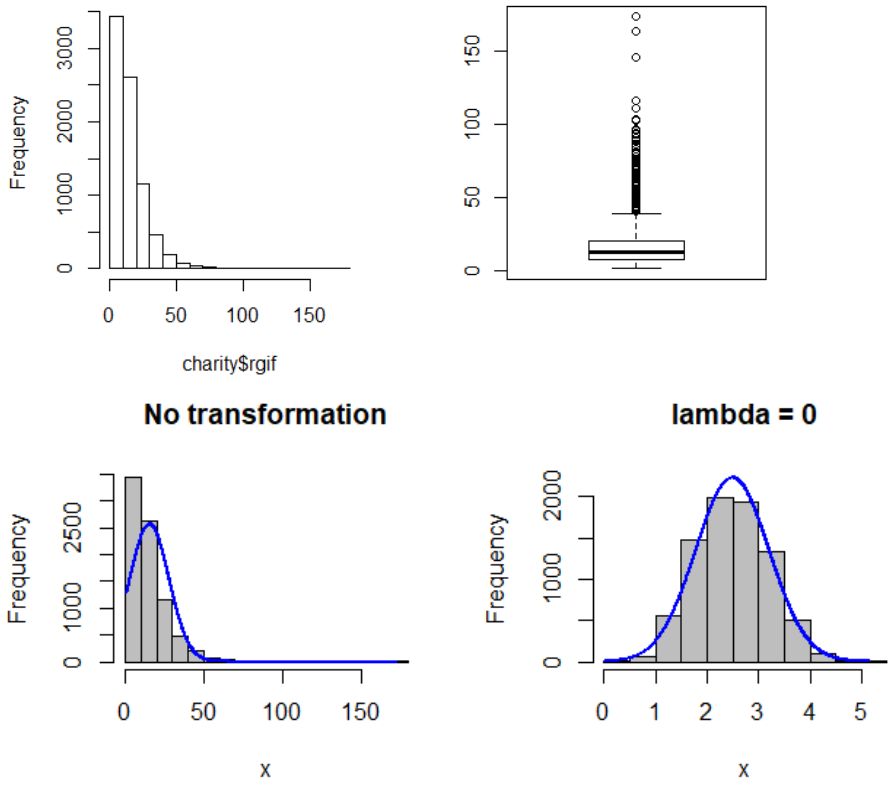
Using the box-cox transformation procedure a lambda value of -0.3 was selected. Above graph shows the change in distribution. The transformed variable was named T_tgif.

LGIF Variable



Using the box-cox transformation procedure a lambda value of -0.2 was selected. Above graph shows the change in distribution. The transformed variable was named T_lgif.

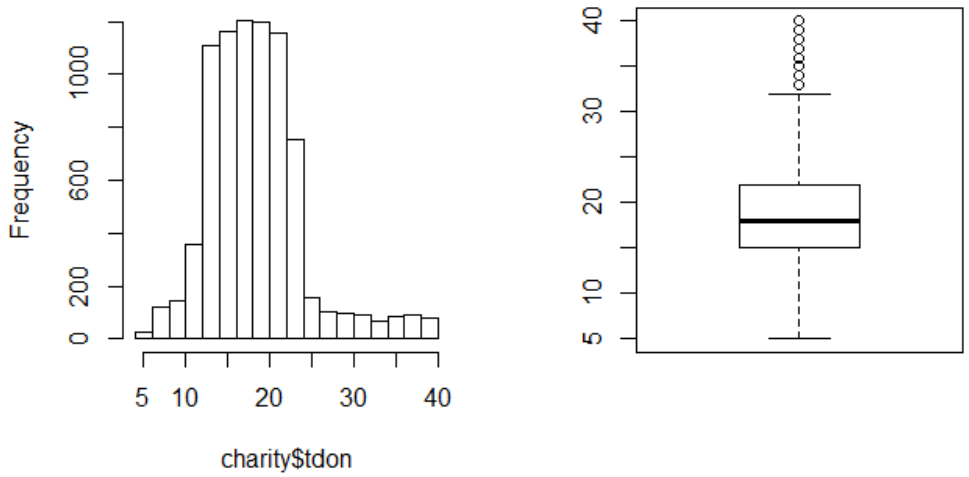
RGIF Variable



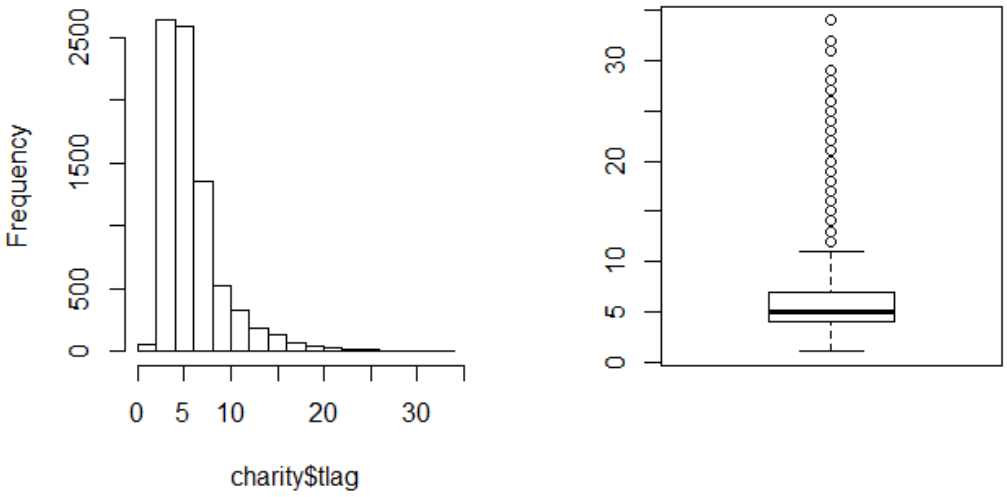
Using the box-cox transformation procedure a lambda value of 0 (log transformation) was selected. Above graph shows the change in distribution. The transformed variable was named T_rgif.

TDON Variable

The variable was left untransformed.



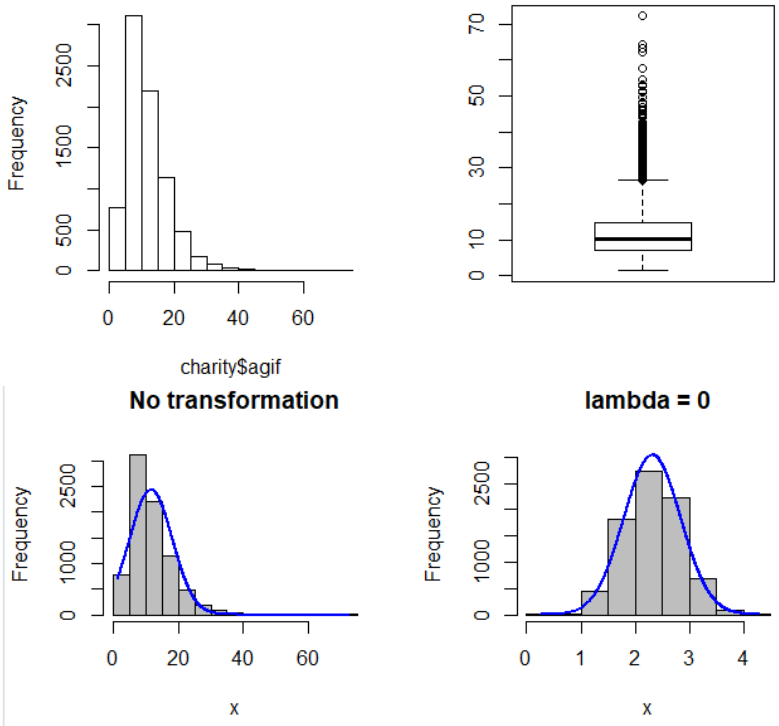
TLAG Variable



Transformed by creating two indicator variables.

Indicator Variable	**Count**
G2_tlag (tlag > 2 & tlag <=7)	6228
G7_tlag (tlag>7)	1723
Remaining (intercept)	58

AGIF Variable



Using the box-cox transformation procedure a lambda value of 0 (log transformation) was selected. Above graph shows the change in distribution. The transformed variable was named T_agif.

HINC Variable

The variable was left untransformed.