Direct-Mail Fundraising

Jason Louwagie

8/13/2021

Background

A national veterans organization wishes to develop a predictive model to improve the cost-effectiveness of their direct marketing campaign. The organization, with its in-house database of over 13 million donors, is one of the largest direct-mail fundraisers in the United States. According to their recent mailing records, the overall response rate is 5.1%. Out of those who responded (donated), the average donation is $13.00. Each mailing, which includes a gift of personalized address labels and assortments of cards and envelopes, costs $0.68 to produce and send. Using these facts, we take a sample of this dataset to develop a classification model that can effectively capture donors so that the expected net profit is maximized. Weighted sampling was used, under-representing the non-responders so that the sample has equal numbers of donors and non-donors.

Business Objectives and Goals

The goal of this analysis is to maximize the United States National Veterans Organization's cost-effectiveness of their direct marketing campaign.

The objective of the analysis is to create a predictive classification model to capture mail recipients that will make a donation to maximze the United States National Veterans Organization expected net profit.

Data Sources and Data Used

In order to do achieve our business objectives and goals, the predictive model will be created and tested against a dataset made available to us, Fundraising.rds. The fundraising.rds dataset has 3,000 observations with 50% donors and 50% non-donors.

The reason our data set is using weighted sampling is important for our classification models. There can be a negative effect to our models if there is a difference in distribution for our classes. We would not want to use a simple random sample because it can be bias towards any particular class that may be more frequent.

Once this model is made, we will use the model against an additional dataset, future_fundraising.rds, to make our predictions. The future_fundraising.rds dataset has 120 observations.

To start our analysis, we will first want to load libraries that will be used throughout our analysis.

```
library(readr)
library(caret)
library(car)
library(MASS)
library(dplyr)
library(class)
```

Type of Analysis Performed: What, Why, Findings

We will begin our anaylsis by conducting exploratory data analysis on our fundraising.rds dataset. We will first need to load our dataset in order to do so.

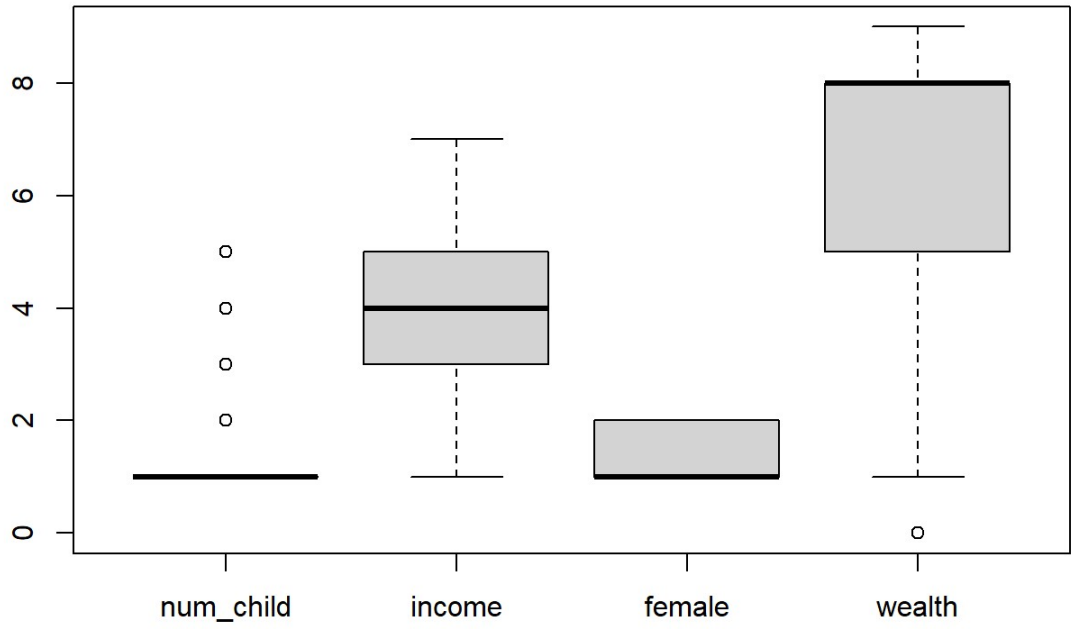
```
fundraising <- read_rds("C:/Users/moonw/Documents/UTSA MSDA Graduate Program/2_Summer 2021/STA 6543/Final Project - Modeling Competition/fundraising.rds")
```

Now that we have our dataset loaded, we will review the summary statistics by removing any null values that may be present, reviewing boxplots of each attribute, each attribute's correlation with each other, and observing if there is any collinearity amongst the attributes.

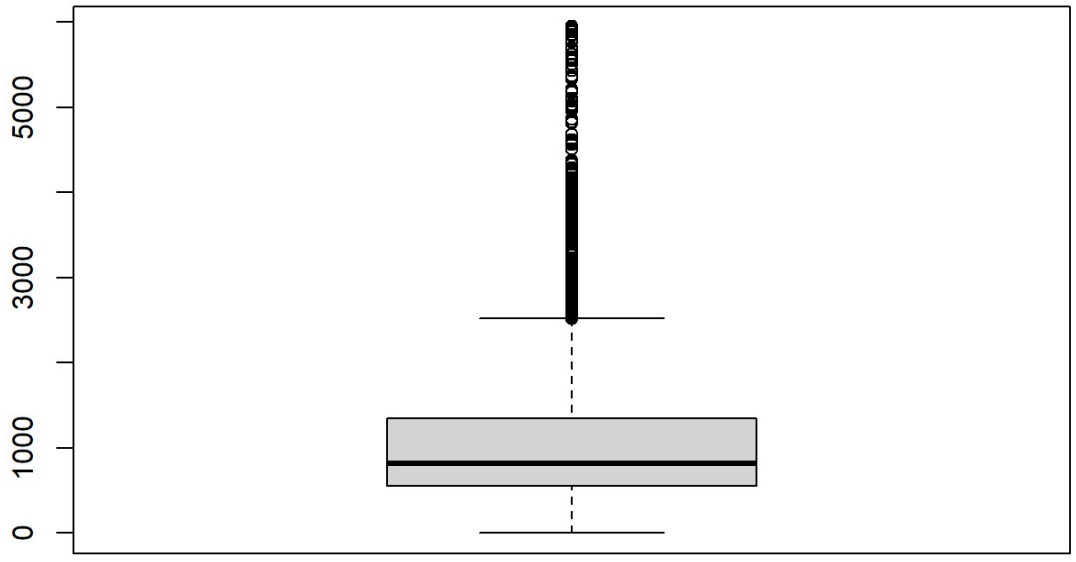
```
attach(fundraising)
fundraising=na.omit(fundraising)

summary(fundraising)
```

```
##  zipconvert2 zipconvert3 zipconvert4 zipconvert5 homeowner    num_child
##  No :2352    Yes: 551    No :2357    No :1846    Yes:2312   Min.   :1.000
##  Yes: 648    No :2449    Yes: 643    Yes:1154    No : 688   1st Qu.:1.000
##                                                             Median :1.000
##                                                             Mean   :1.069
##                                                             3rd Qu.:1.000
##                                                             Max.   :5.000
##      income      female         wealth        home_value      med_fam_inc
##  Min.   :1.000   Yes:1831   Min.   :0.000   Min.   :   0.0   Min.   :   0.0
##  1st Qu.:3.000   No :1169   1st Qu.:5.000   1st Qu.: 554.8   1st Qu.: 278.0
##  Median :4.000              Median :8.000   Median : 816.5   Median : 355.0
##  Mean   :3.899              Mean   :6.396   Mean   :1143.3   Mean   : 388.4
##  3rd Qu.:5.000              3rd Qu.:8.000   3rd Qu.:1341.2   3rd Qu.: 465.0
##  Max.   :7.000              Max.   :9.000   Max.   :5945.0   Max.   :1500.0
##   avg_fam_inc       pct_lt15k        num_prom      lifetime_gifts
##  Min.   :   0.0   Min.   : 0.00   Min.   : 11.00   Min.   :  15.0
##  1st Qu.: 318.0   1st Qu.: 5.00   1st Qu.: 29.00   1st Qu.:  45.0
##  Median : 396.0   Median :12.00   Median : 48.00   Median :  81.0
##  Mean   : 432.3   Mean   :14.71   Mean   : 49.14   Mean   : 110.7
##  3rd Qu.: 516.0   3rd Qu.:21.00   3rd Qu.: 65.00   3rd Qu.: 135.0
##  Max.   :1331.0   Max.   :90.00   Max.   :157.00   Max.   :5674.9
##   largest_gift       last_gift      months_since_donate    time_lag
##  Min.   :   5.00   Min.   :  0.00   Min.   :17.00       Min.   : 0.000
##  1st Qu.:  10.00   1st Qu.:  7.00   1st Qu.:29.00       1st Qu.: 3.000
##  Median :  15.00   Median : 10.00   Median :31.00       Median : 5.000
##  Mean   :  16.65   Mean   : 13.48   Mean   :31.13       Mean   : 6.876
##  3rd Qu.:  20.00   3rd Qu.: 16.00   3rd Qu.:34.00       3rd Qu.: 9.000
##  Max.   :1000.00   Max.   :219.00   Max.   :37.00       Max.   :77.000
##     avg_gift            target
##  Min.   :  2.139   Donor   :1499
##  1st Qu.:  6.333   No Donor:1501
##  Median :  9.000
##  Mean   : 10.669
##  3rd Qu.: 12.800
##  Max.   :122.167
```

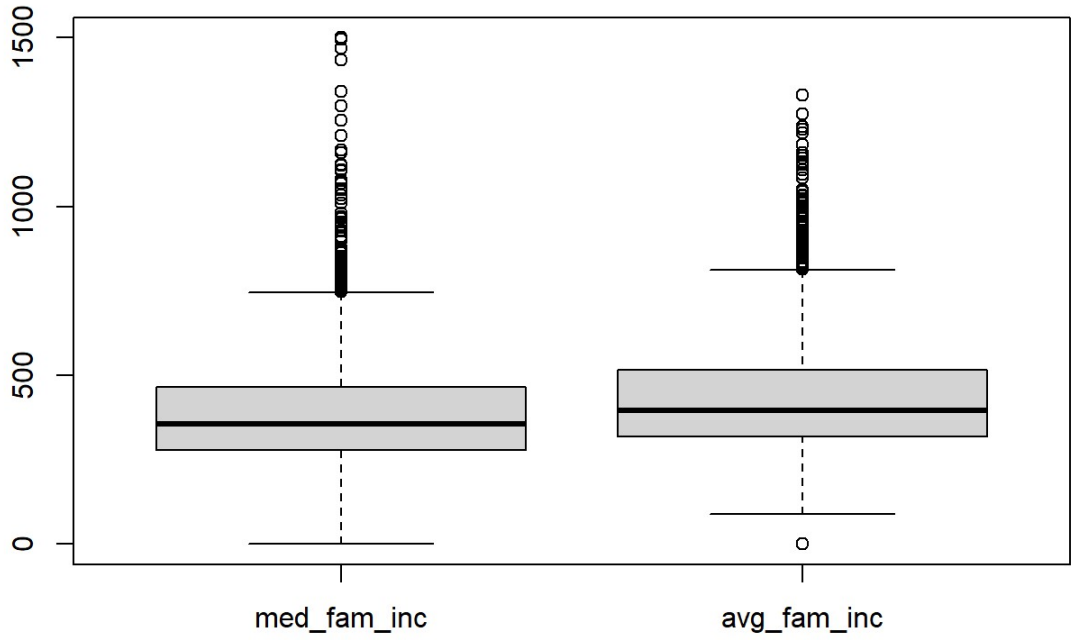
```
boxplot(fundraising[6:9])
```



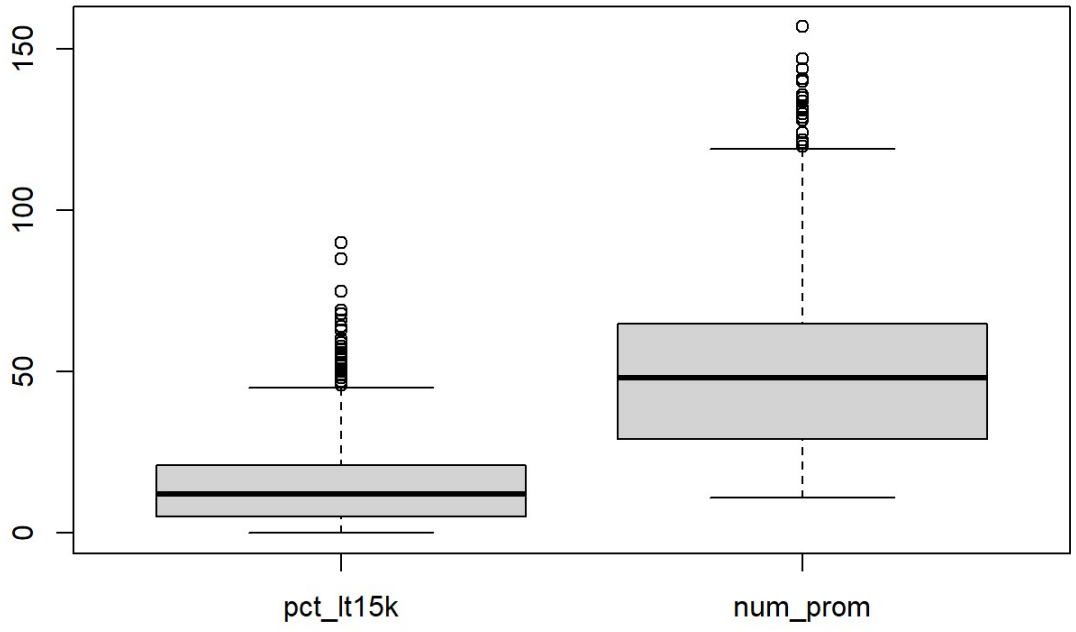

```
boxplot(fundraising[10])
```



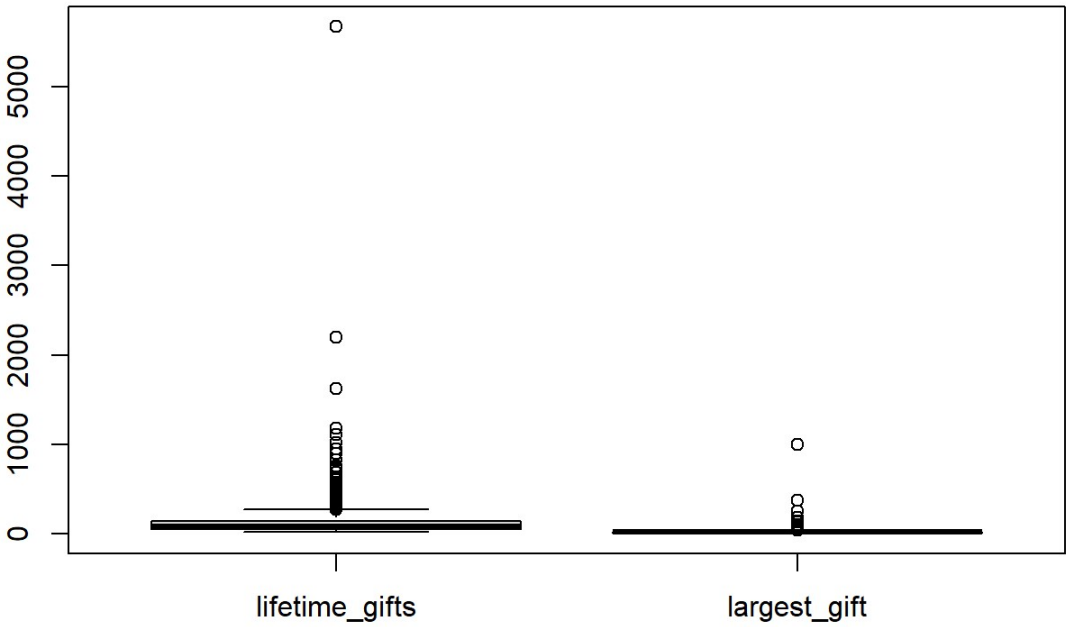

```
boxplot(fundraising[11:12])
```



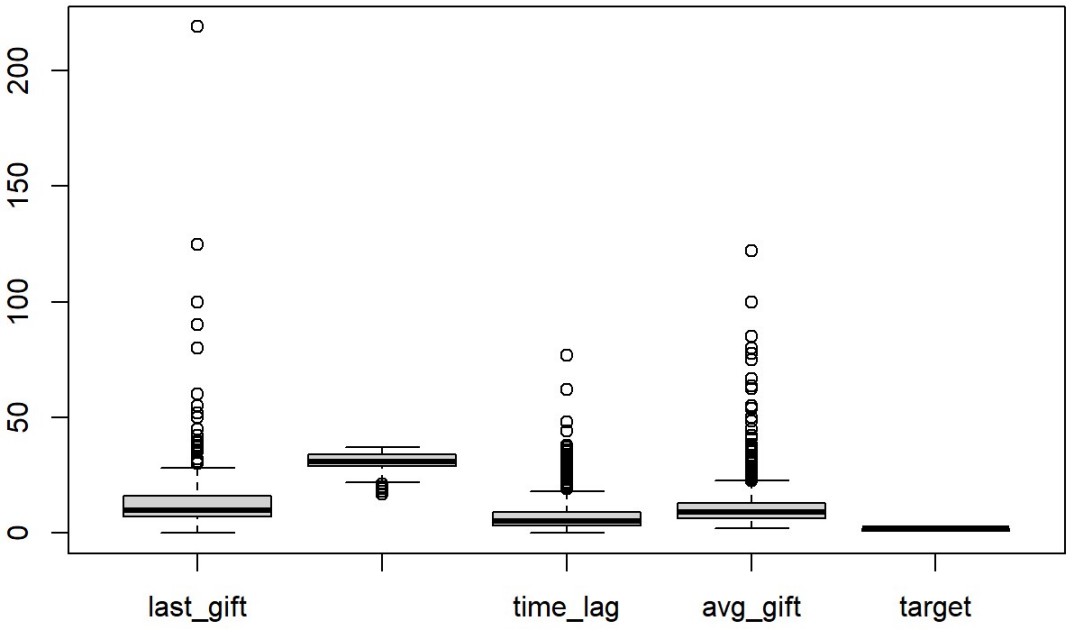

```
boxplot(fundraising[13:14])
```




```
boxplot(fundraising[15:16])
```

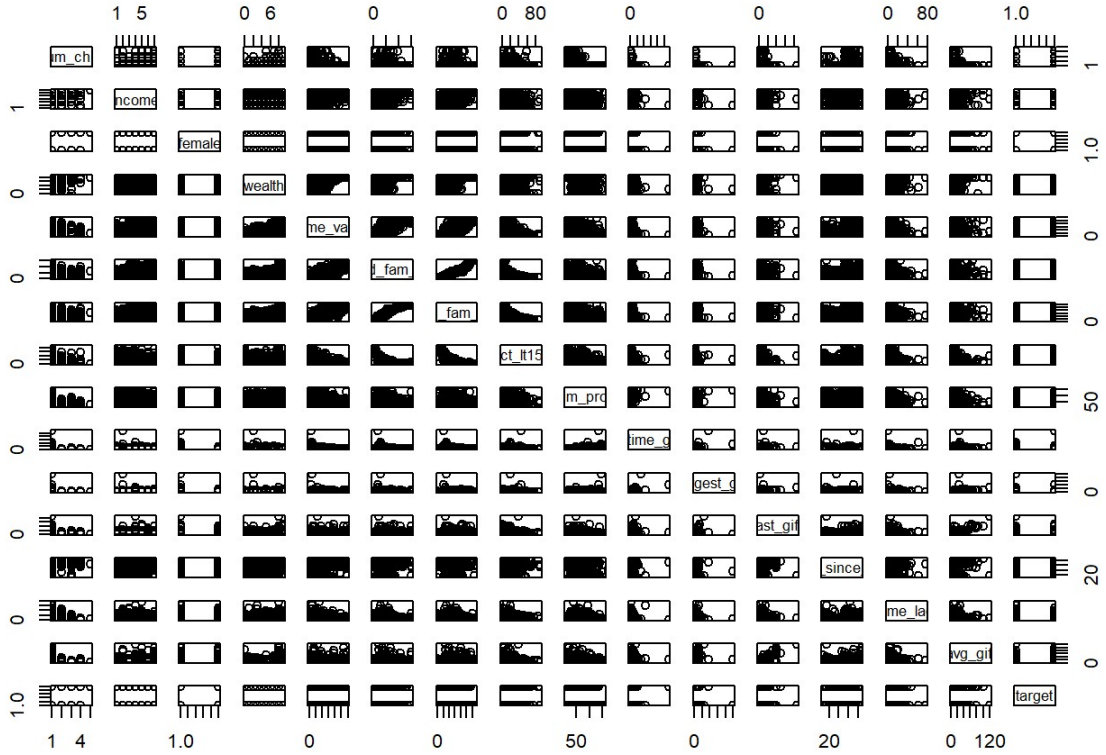



```
boxplot(fundraising[17:21])
```



From the summary of the data set and the boxplots of the each attribute, we can observe that the attributes num_child, wealth, home_value, med_fam_inc, avg_fam_inc, pct_lt15k, num_prom, large_gifts, largest_gifts, lasts_gift, time_lag, and avg_gift are all right skewed and contain outliers. We can also observe that the attriubte months_since_donated is left skewed.

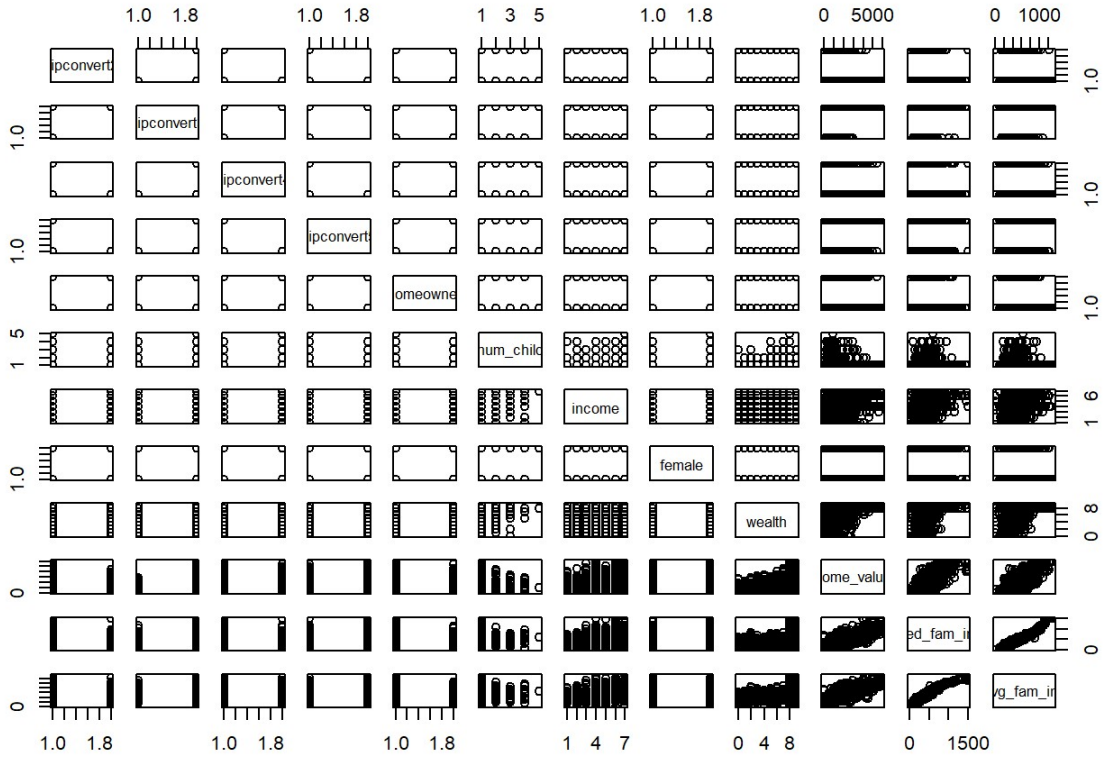
```
pairs(fundraising[6:21])
```



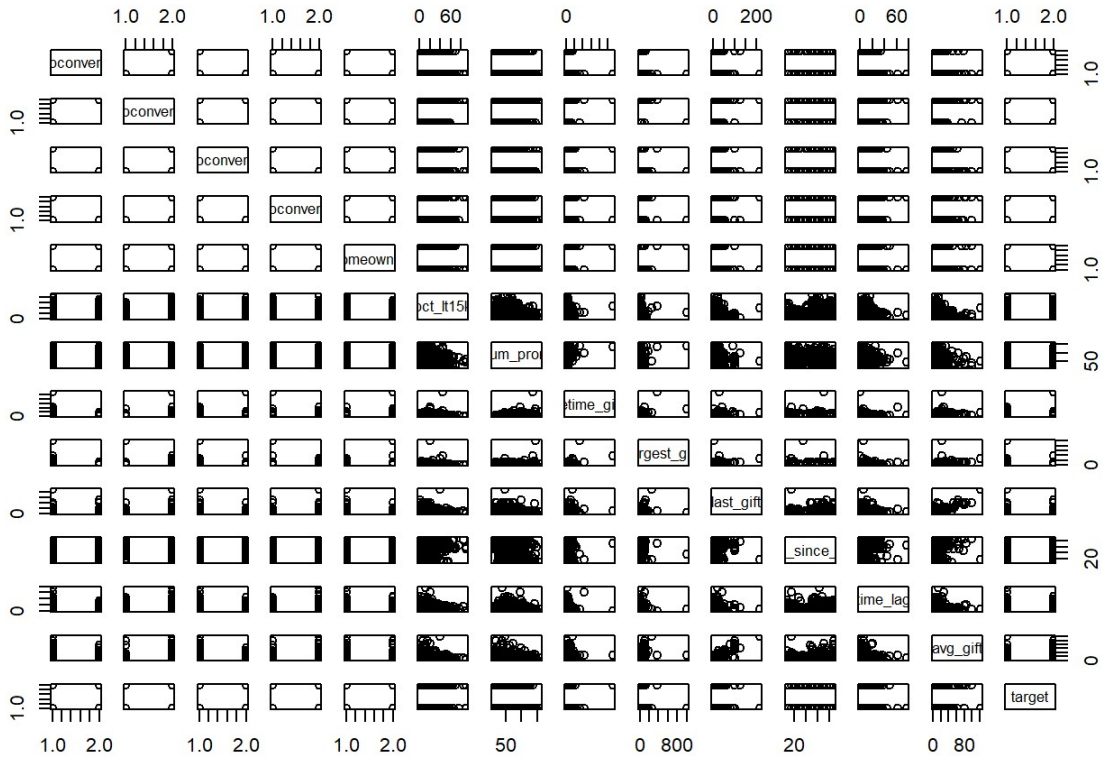
From the scatter plots above, which compares all attribtues except the zipcodes and homeowner attributes, we can take note that there is a positive correlation between the med_fam_inc and avg_fam_inc, med_fam_inc and home_value, and avg_fam_inc and home_value.

Additionally, we can also see that there is a negative correlation between med_fam_inc and pct_lt15k, home_value and pct_lt15k, and avg_fam_inc and pct_lt15k.

```
pairs(fundraising[1:12])
```




```
pairs2 <- fundraising %>% select(-(6:12))
pairs(pairs2)
```



The two scatter plots above allow us to compare the zipcodes and homeowner attributes to the rest of the data. From these, we can observe there there is no real correlation with these attributes to any others.

Now that we have reviewed the summary statistics, we will now review our data for

collinearity. Collinearity takes place when there are two or more variables that are highly correlated to the point that they cannot independently predict the value of the response variable. Collinearity reduces a variable's statistical significance and the attributes with a high collinear value must be removed one at a time and retested until there are no longer any attributes that have a high collinear value. Our threshold to remove a variable will be 5. We will fit our response variable, target, to a logisitic regression model and use the vif() function to find any collinear attributes. We will then remove attributes one at a time until we no longer have high collinear values above 5.

```
model_glm = glm(target~.,family = "binomial", data = fundraising)

vif(model_glm)
```

```
##         zipconvert2         zipconvert3         zipconvert4         zipconvert5
##        8.790721e+06        7.784501e+06        8.750031e+06        1.225688e+07
##           homeowner           num_child              income              female
##        1.132353e+00        1.026222e+00        1.311414e+00        1.016256e+00
##              wealth          home_value         med_fam_inc         avg_fam_inc
##        1.523860e+00        3.308054e+00        1.877177e+01        2.100427e+01
##           pct_lt15k            num_prom     lifetime_gifts        largest_gift
##        2.102697e+00        1.964905e+00        2.135341e+00        2.098192e+00
##           last_gift months_since_donate            time_lag            avg_gift
##        3.945654e+00        1.132696e+00        1.038106e+00        4.232227e+00
```

From the values above, we can observe that zipconvert2 has the highest vif() value above 5. Due to this, we will remove this variable from our logistic model and retest for high collinear values.

```
model_glm = glm(target~.-zipconvert2,family = "binomial", data = fundraising)

vif(model_glm)
```

```
##         zipconvert3         zipconvert4         zipconvert5           homeowner
##            1.555523            1.591039            1.983755            1.133003
##           num_child              income              female              wealth
##            1.026233            1.311900            1.016280            1.520999
##          home_value         med_fam_inc         avg_fam_inc          pct_lt15k
##            3.289635           18.773445           21.005426            2.099468
##            num_prom      lifetime_gifts        largest_gift           last_gift
##            1.962553            2.137299            2.105473            3.952911
## months_since_donate            time_lag            avg_gift
##            1.131956            1.037745            4.238023
```

From the values above, we can observe that avg_fam_inc has the highest vif() value above 5. Due to this, we will remove this variable from our logistic model and retest for high collinear values.

```
model_glm = glm(target~.-zipconvert2-avg_fam_inc,family = "binomial", data = fundraising)

vif(model_glm)
```

```
##         zipconvert3         zipconvert4         zipconvert5           homeowner
##            1.552217            1.590326            1.973780            1.129662
##           num_child              income              female              wealth
##            1.026185            1.300439            1.016322            1.518717
##          home_value         med_fam_inc           pct_lt15k            num_prom
##            3.069859            3.912247            1.984977            1.960837
##      lifetime_gifts        largest_gift          last_gift months_since_donate
##            2.131605            2.090869            3.965786            1.131913
##            time_lag            avg_gift
##            1.033901            4.245008
```

We can observe there are no longer vif() values that are above 5. This indicates that there is no longer a need to remove any more predictors.

Exclusions

From our exploratory data analysis, we have removed the attributes zipconvert2 and avg_fam_inc. We will reflect this by removing these attributes and creating a new dataset called fundraising1.

```
fundraising1 <- fundraising %>% select(-c(1,12))
```

Cut-Off Analysis

Moving forward, we will be using .05 as our cut off for statistical signifiance. For selecting our predictive models, we will only chose two models that have an error rate less than 50%.

Methodology Used, Background, and Benefits

Partitioning

In order to create a predictive model and test its accuracy, it is neccessary to split our data into training and testing subsets. I chose to partition the data in two ways: an 80-20 split and Cross-Validation.

80-20 Split

```
set.seed(12345)


index = sample(nrow(fundraising1), 0.8*(nrow(fundraising1)))

train = fundraising1[index,]
test = fundraising1[-index,]
```

Cross Validation

```
train_control <- trainControl(method="repeatedcv",number=10,repeats=3)
```

Logistic Regression

In order to see if a logistic regression model would work best as our predictive model, we will fit our train subset of fundraising1 to a glm() function. We will then create a prediction and apply it to our test subset to see the accuracy of our predictive model.

```
glm.fit = glm(target~.,data = train, family = "binomial")

glm.probs = predict(glm.fit, newdata = test, type = "response")
glm.pred=rep("No Donor",length(glm.probs))
glm.pred[glm.probs > 0.5] = "Donor"
table(glm.pred, test$target)
```

```
##
## glm.pred   Donor No Donor
##   Donor      119      149
##   No Donor   171      161
```

```
mean(glm.pred != test$target)
```

```
## [1] 0.5333333
```

We observe an error rate for .533, which indicates a poor predictive capability of our model.

In an atttempt to improve our error rate, we will observe the summary of the logistic model and remove variables based on their p-values.

```
summary(glm.fit)
```

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8159  -1.1423  -0.7549   1.1671   1.6900
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.660e+00  5.017e-01  -3.309 0.000935 ***
## zipconvert3No       -9.195e-02  1.334e-01  -0.689 0.490644
## zipconvert4Yes       2.813e-02  1.282e-01   0.219 0.826382
## zipconvert5Yes       4.950e-02  1.209e-01   0.410 0.682108
## homeownerNo          1.545e-01  1.057e-01   1.462 0.143765
## num_child            3.314e-01  1.278e-01   2.594 0.009493 **
## income              -5.105e-02  2.865e-02  -1.782 0.074710 .
## femaleNo             2.477e-02  8.584e-02   0.289 0.772868
## wealth              -1.923e-02  1.995e-02  -0.964 0.335010
## home_value          -6.640e-05  7.669e-05  -0.866 0.386580
## med_fam_inc          1.721e-04  4.779e-04   0.360 0.718835
## pct_lt15k           -5.053e-03  4.845e-03  -1.043 0.296994
## num_prom            -4.153e-03  2.565e-03  -1.619 0.105408
## lifetime_gifts       2.938e-04  4.025e-04   0.730 0.465526
## largest_gift        -2.186e-03  3.372e-03  -0.648 0.516911
## last_gift            1.411e-02  8.672e-03   1.627 0.103653
## months_since_donate  5.359e-02  1.126e-02   4.761 1.93e-06 ***
## time_lag            -4.208e-04  7.722e-03  -0.054 0.956542
## avg_gift             4.779e-03  1.235e-02   0.387 0.698795
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3327.0  on 2399  degrees of freedom
## Residual deviance: 3255.2  on 2381  degrees of freedom
## AIC: 3293.2
##
## Number of Fisher Scoring iterations: 4
```

To determine statistical significance, we will utilize our hypothesis test: H0: B1 = 0 vs Ha: B1 != 0. If a predictor's p-value is greater than .05, the predictor is not statisitcally significant and we fail to reject the null hypothesis. Of all the p-values, only num_child and months_since_donate have p-values less than .05, therefore we reject the null hypothesis and determine these values to be statistically significant.

From this, we will refit our logistic model with just num_child and months_since_donate. We will then create a prediction and apply it to our test subset to see the accuracy of our predictive model.

```
glm.fit2 = glm(target~num_child+months_since_donate,data = train, family = "binomial")

glm.probs2 = predict(glm.fit2, newdata = test, type = "response")
glm.pred2=rep("No Donor",length(glm.probs))
glm.pred2[glm.probs2 > 0.5] = "Donor"
table(glm.pred2, test$target)
```

```
##
## glm.pred2  Donor No Donor
##   Donor       101       140
##   No Donor    189       170
```

```
mean(glm.pred2 != test$target)
```

```
## [1] 0.5483333
```

We observe an error rate for .548, which indicates a poor predictive capability of our model and no improvement when we reduced our predictors.

Linear Discriminant Analysis

Here we will be fitting our data to a LDA model to observe if this would be a good predictive model. We will fit the train subset to a lda() function and then test its accuracy against our test subset.

```
lda.fit = lda(target~.,data=train)

lda.pred = predict(lda.fit, test)$class
table(lda.pred, test$target)
```

```
##
## lda.pred   Donor No Donor
##   Donor       171       162
##   No Donor    119       148
```

```
mean(lda.pred != test$target)
```

```
## [1] 0.4683333
```

We observe an error rate for .468, which indicates a decent predictive capability of our model. This means we have a greater than 50% probability of making the correct prediction.

Quadratic Discriminant Analysis

Here we will be fitting our data to a QDA model to observe if this would be a good predictive model. We will fit the train subset to a qda() function and then test its accuracy against our train subset.

```
qda.fit = qda(target~., data=train)

qda.pred = predict(qda.fit, test)$class
table(qda.pred, test$target)
```

```
##
## qda.pred   Donor No Donor
##   Donor        53        46
##   No Donor    237       264
```

```
mean(qda.pred != test$target)
```

```
## [1] 0.4716667
```

We observe an error rate for .471, which indicates a decent predictive capability of our model. This means we have a greater than 50% probability of making the correct prediction.

K-Nearest Neighbors

Here we will be fitting our data to a KNN model to observe if this would be a good predictive model. We will fit the cross validation training subset, train_control, to a knn function and then test its accuracy against our test subset. In this assessment, we will only use the variables we found to be statistically significant in our logisitic regression model, num_child and months_since_donate.

```
knn <- train(target~num_child+months_since_donate,data = train,method='knn',trControl = train
_control, tuneLength = 20)

knnpred = predict(knn, test)
mean(knnpred != test$target)
```

```
## [1] 0.455
```

We observe an error rate for .448, which indicates our best predictive model so far. This means we have a greater than 50% probability of making the correct prediction.

Model Performance and Validation Results

Now we will use our best performing model, KNN with an error rate of .448, to make our predictions.

In order to apply our predictive model, we will need to load our data set, future_fundraising.rds.

```
funtest <-  read_rds("C:/Users/moonw/Documents/UTSA MSDA Graduate Program/2_Summer 2021/STA 6
543/Final Project - Modeling Competition/future_fundraising.rds")
```

K-Nearest Neighbor Performance and Results

```
knnpred =predict(knn, funtest)

write.table(knnpred, file = "knnpredictions.csv", col.names = c("value"), row.names = FALSE)
```

Recommendations

I would recommend that the United States National Veterans Organization utilize a K-Nearest Neighbor predictive model to best improve their odds of maximizing their expected net profit and increasing the cost effectiveness of their direct mail fundraising.

When doing so, I would advise the organization to prioritize their focus on the number of children the mail recipeint has and the last time the recipient donated. The more children a recipient has, the more likely they are to donate, and the longer it has been since the recipient last donated, the more likely they are to donate.