Raquel Ocasio

D214 – Data Analytics Graduate Capstone

Task 2: Data Analytics Report and Executive Summary

June 2, 2024

Western Governors University

Part A: Research Question

Can a multiple linear regression model be constructed based on the research dataset? The contribution of this study to the field of Data Analytics and the MSDA program is to create a predictive model which can estimate the year-to-date return of mutual funds so a brokerage can provide investment recommendations to its clients. This study will utilize multiple linear regression to analyze the year-to-date return of funds and estimate their future prices. Intrinio (2024b) found that a regression model can be useful for capturing relationships between variables that may affect the year-to-date return of a mutual fund. By understanding the trend of prices over time, it should be possible to predict the year-to-date return given historical data.

The null hypothesis is: A predictive multiple linear regression model cannot be made from the research dataset. The alternative hypothesis is: A predictive multiple linear regression model can be constructed from the research dataset at a model accuracy >= 70%.

Part B: Data Collection

The dataset to be used for this study is named "US Funds dataset from Yahoo Finance" and is publicly available from Kaggle.com (US Funds Dataset From Yahoo Finance, 2021). The dataset is made up of several CSV files. The file used for this study is "MutualFunds.csv" and contains 23,783 observations. There were no challenges encountered during the collection of the data.

The dataset includes the following variables:

Variable Name	**Type**	**Dependent or Independent**
total_net_assets	Numerical	independent
annual_holdings_turnover	Numerical	independent
fund_annual_report_net_expense_ratio	Numerical	independent
fund_prospectus_net_expense_ratio	Numerical	independent
fund_prospectus_gross_expense_ratio	Numerical	independent
fund_max_12b1_fee	Numerical	independent
fund_max_front_end_sales_load	Numerical	independent
year-to-date_return	Numerical	dependent

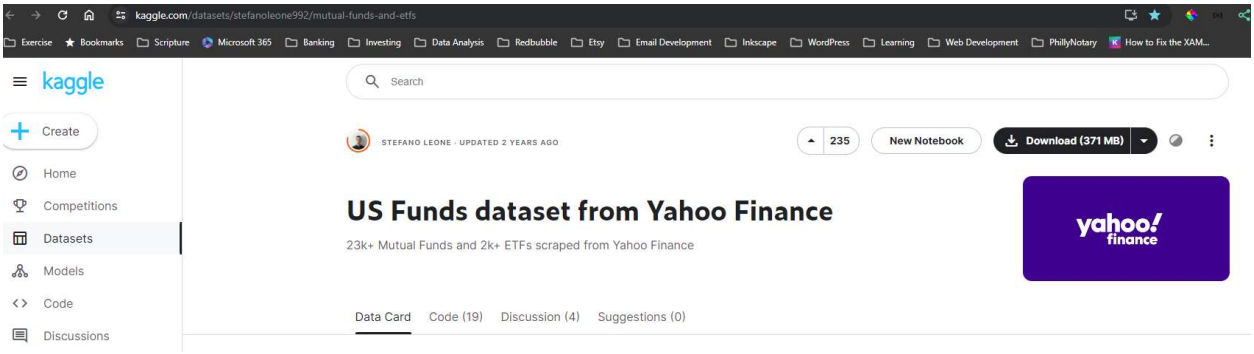
Limitations: The dataset is limited by the completeness and accuracy of the information made available by Yahoo Finance.

Delimitations: The chosen dataset provides values for mutual funds in existence at the time the data was collected, which may impact the ability of the model to accurately predict values for any mutual funds created since then.

Part C: Data Extraction and Preparation

The R programming language was used for data extraction, preparation, and analysis. One advantage of using these tools and techniques is that is provides a clear understanding of data before analysis. One disadvantage of using these tools and techniques is that it is very manual in nature and therefore time consuming.

The data was downloaded as a CSV file from the Kaggle.com website.



The CSV file was read into a dataframe. The variables to be used in the analysis were copied to a new dataframe. The structure of the dataframe was viewed to confirm the variables were copied.

```
library(ggplot2)

# Read data from CSV file into a dataframe
fundsdata <- read.csv("MutualFunds.csv")
#typeof(fundsdata$year_to_date_return)

# copy columns for study into new dataframe
fundsdata_study <- fundsdata[ , c("total_net_assets", "annual_holdings_turnover", "fund_annual_report_net_expense_ratio", "fund_prospectus_net_expense_ratio", "fund_prospectus_gross_expense_ratio", "fund_max_12b1_fee", "fund_max_front_end_sales_load", "year_to_date_return")]

# view structure of dataframe
str(fundsdata_study)
```

```
## 'data.frame':    23783 obs. of  8 variables:
##  $ total_net_assets                  : num  2.98e+09 1.95e+08 2.59e+04 2.08e+09 2.59e+04 ...
##  $ annual_holdings_turnover          : num  0.74 0.91 NA 0.44 NA 1.09 NA NA NA ...
##  $ fund_annual_report_net_expense_ratio: num  0.0122 0.0109 0.0058 0.0108 0.0038 0.0063 0.0083 0.0108 0.0023 0.0058 ...
##  $ fund_prospectus_net_expense_ratio   : num  0.0122 0.0109 0.0058 0.0108 0.0038 0.0063 0.0083 0.0108 0.0023 0.0058 ...
##  $ fund_prospectus_gross_expense_ratio : num  0.0136 0.0109 0.006 0.0112 0.004 0.0112 0.0085 0.011 0.0025 0.0059 ...
##  $ fund_max_12b1_fee                 : num  0.0024 NA NA 0.0025 NA NA 0.0025 0.005 NA NA ...
##  $ fund_max_front_end_sales_load     : num  0.0575 NA NA 0.045 NA NA 0.0575 NA NA NA ...
##  $ year_to_date_return               : num  0.21 0.191 NA 0.246 NA ...
```

The first step in data preparation was to view summary statistics for the dataset.

```
# summary statistics
summary(fundsdata_study)
```

```
##  total_net_assets    annual_holdings_turnover
##  Min.   :1.000e+01   Min.   :  0.0042
##  1st Qu.:1.446e+08   1st Qu.:  0.2700
##  Median :6.443e+08   Median :  0.5200
##  Mean   :4.924e+09   Mean   :  0.9328
##  3rd Qu.:2.453e+09   3rd Qu.:  0.9200
##  Max.   :7.534e+11   Max.   :110.6600
##  NA's   :34          NA's   :1808
##  fund_annual_report_net_expense_ratio fund_prospectus_net_expense_ratio
##  Min.   :0.00010                      Min.   :0.00010
##  1st Qu.:0.00660                      1st Qu.:0.00660
##  Median :0.00950                      Median :0.00950
##  Mean   :0.01047                      Mean   :0.01047
##  3rd Qu.:0.01340                      3rd Qu.:0.01340
##  Max.   :0.11800                      Max.   :0.11800
##  NA's   :210                          NA's   :216
##  fund_prospectus_gross_expense_ratio fund_max_12b1_fee
##  Min.   : 0.00010                    Min.   :0.000
##  1st Qu.: 0.00750                    1st Qu.:0.002
##  Median : 0.01100                    Median :0.002
##  Mean   : 0.01945                    Mean   :0.005
##  3rd Qu.: 0.01610                    3rd Qu.:0.010
##  Max.   :13.44620                    Max.   :0.010
##  NA's   :78                          NA's   :12891
##  fund_max_front_end_sales_load year_to_date_return
##  Min.   :0.004                 Min.   :-0.5228
##  1st Qu.:0.040                 1st Qu.: 0.0261
##  Median :0.052                 Median : 0.0920
##  Mean   :0.047                 Mean   : 0.0935
##  3rd Qu.:0.058                 3rd Qu.: 0.1497
##  Max.   :0.085                 Max.   : 0.5789
##  NA's   :20090                 NA's   :401
```

The next step was to check for and remove any duplicate values. There were 273 instances of duplicate values.

```
# check for duplicate values
sum(duplicated(fundsdata_study))
```

```
## [1] 273
```

```
# remove duplicates
fundsdata_study <- fundsdata_study[!duplicated(fundsdata_study), ]
# confirm duplicates are removed
sum(duplicated(fundsdata_study))
```

```
## [1] 0
```

The next step was to check for and treat any outlier values. No outlier values were found.

```
# check for outlier values
# Function to calculate z-scores and display results for specified columns
calculate_z_scores <- function(data, columns, threshold = 3) {
  for (column in columns) {
    if (column %in% names(data)) {
      # Calculate z-scores using scale()
      z_scores <- scale(data[[column]])

      # Find the count of outliers
      outliers_count <- sum(abs(z_scores) > threshold)

      # Print results
      cat("Column:", column, "\n")
      cat("Count of Outliers:", outliers_count, "\n\n")
    } else {
      cat("Column", column, "not found in the data frame.\n\n")
    }
  }
}

# List of columns to check
columns <- c("total_net_assets", "annual_holdings_turnover", "fund_annual_report_net_expense_ratio", "fund_prospectus_net_ex
pense_ratio", "fund_prospectus_gross_expense_ratio", "fund_max_12b1_fee", "fund_max_front_end_sales_load", "year_to_date_ret
urn")

# Call the function
calculate_z_scores(fundsdata_study, columns)
```

```
## Column: total_net_assets
## Count of Outliers: NA
##
## Column: annual_holdings_turnover
## Count of Outliers: NA
##
## Column: fund_annual_report_net_expense_ratio
## Count of Outliers: NA
##
## Column: fund_prospectus_net_expense_ratio
## Count of Outliers: NA
##
## Column: fund_prospectus_gross_expense_ratio
## Count of Outliers: NA
##
## Column: fund_max_12b1_fee
## Count of Outliers: NA
##
## Column: fund_max_front_end_sales_load
## Count of Outliers: NA
##
## Column: year_to_date_return
## Count of Outliers: NA
```

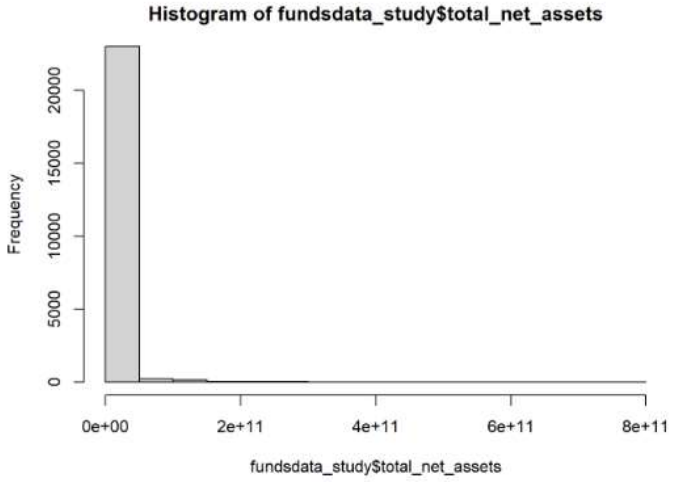
The next step was to check for and treat any missing values. All the variables were found to have missing values.

```
# check for missing values
colSums(is.na(fundsdata_study))
```

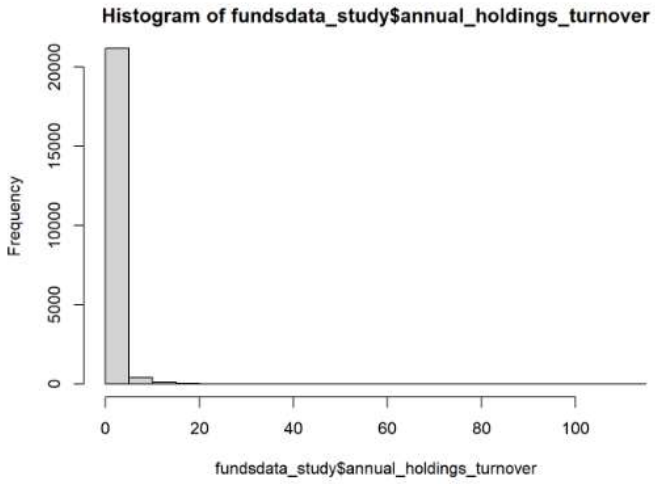
```
##                    total_net_assets            annual_holdings_turnover
##                                  32                                1788
## fund_annual_report_net_expense_ratio    fund_prospectus_net_expense_ratio
##                                 210                                 216
##  fund_prospectus_gross_expense_ratio                  fund_max_12b1_fee
##                                  78                               12685
##        fund_max_front_end_sales_load                year_to_date_return
##                                19819                                391
```

The distribution of each variable was inspected to determine the imputation treatment.

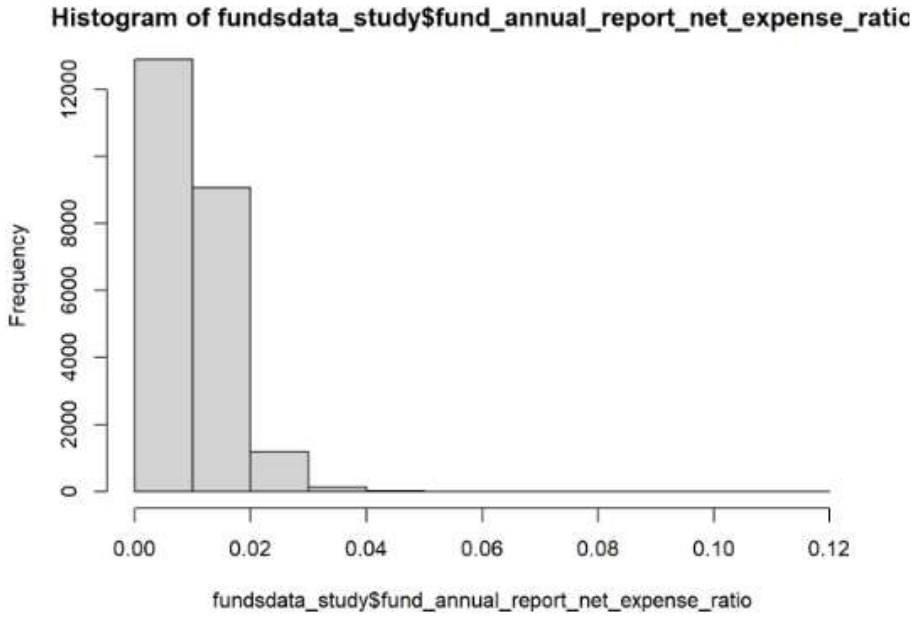
```
# histograms to inspect distribution of the variables
par(mar = c(5, 4, 2, 2))
hist(fundsdata_study$total_net_assets)
```



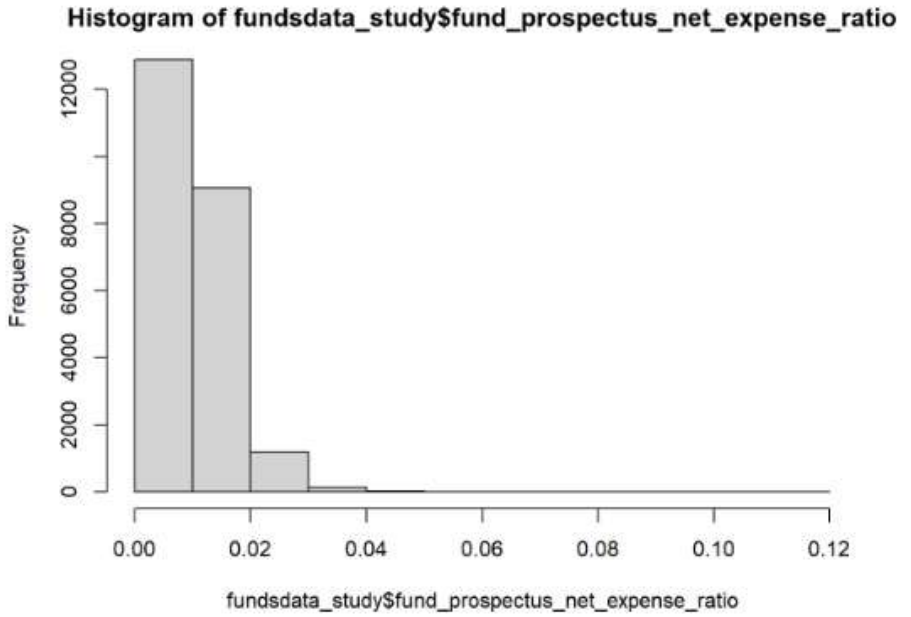

```
hist(fundsdata_study$annual_holdings_turnover)
```



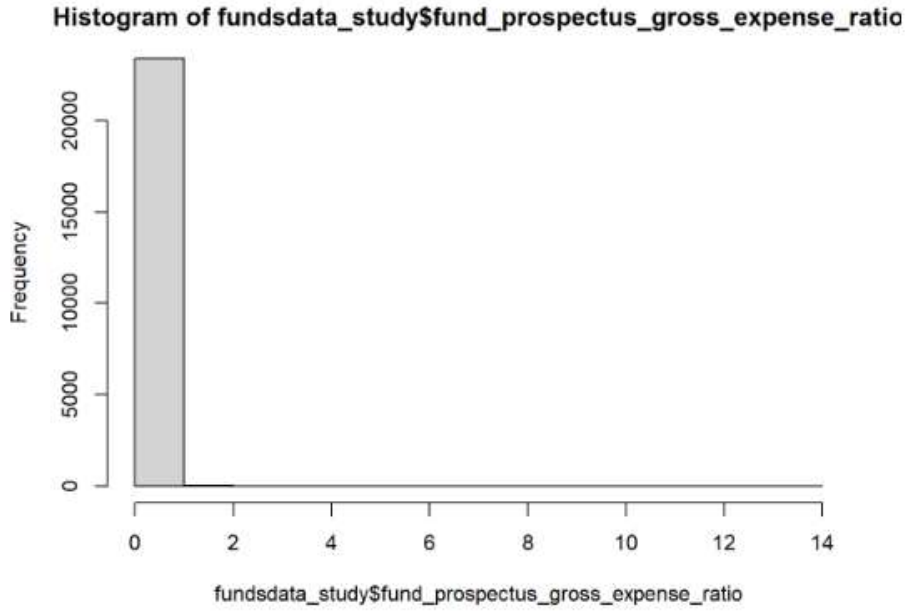
```
hist(fundsdata_study$fund_annual_report_net_expense_ratio)
```



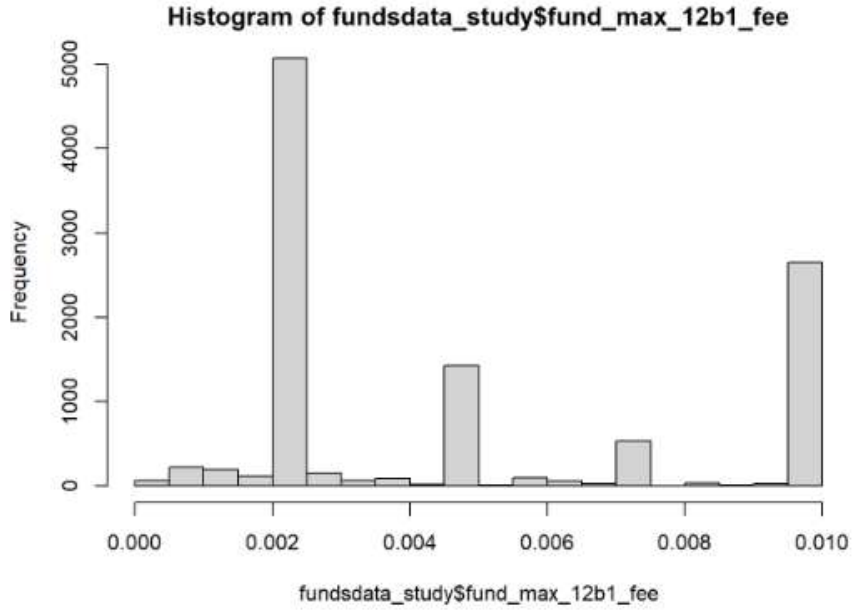

```
hist(fundsdata_study$fund_prospectus_net_expense_ratio)
```



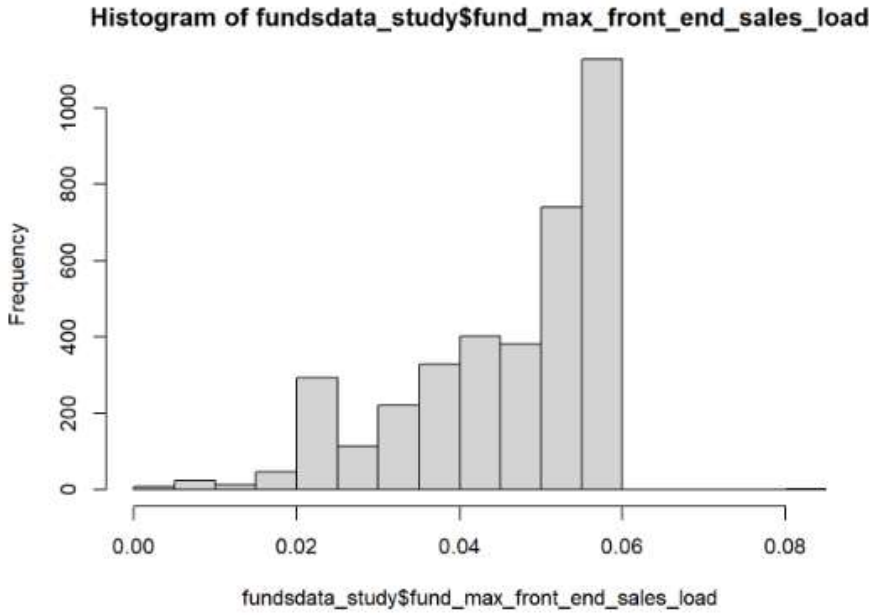
```
hist(fundsdata_study$fund_prospectus_gross_expense_ratio)
```



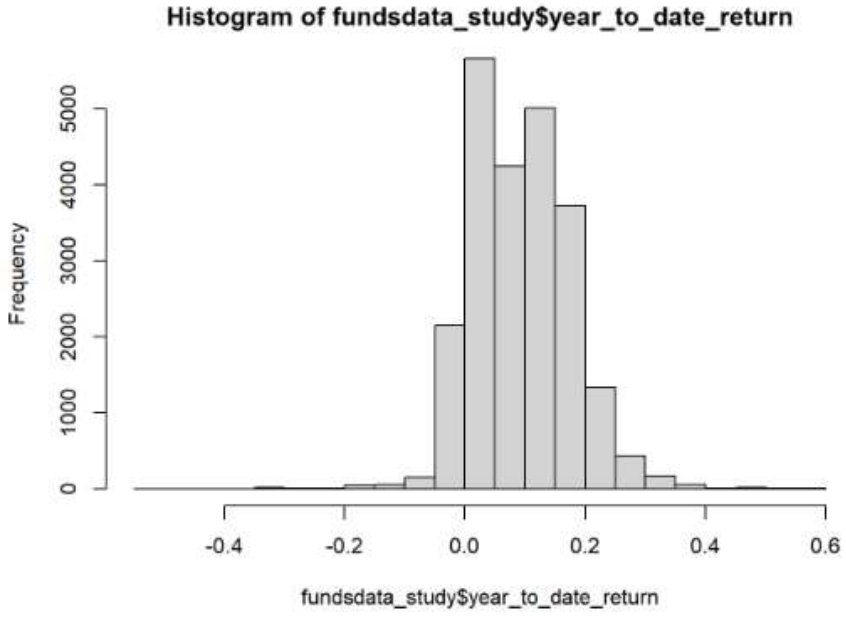

```
hist(fundsdata_study$fund_max_12b1_fee)
```



```
hist(fundsdata_study$fund_max_front_end_sales_load)
```




```
hist(fundsdata_study$year_to_date_return)
```



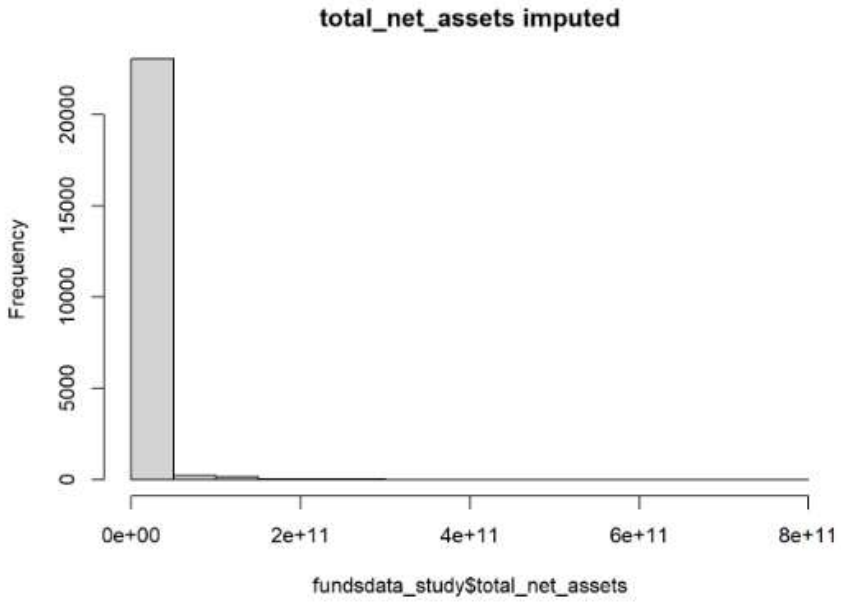
The missing values were imputed using a method appropriate for the distribution of the variable. The variables were checked again to confirm the missing values were imputed.

```
# impute missing values
fundsdata_study$total_net_assets[is.na(fundsdata_study$total_net_assets)] <- median(fundsdata_study$total_net_assets, na.rm=
TRUE)

fundsdata_study$annual_holdings_turnover[is.na(fundsdata_study$annual_holdings_turnover)] <- median(fundsdata_study$annual_h
oldings_turnover, na.rm=TRUE)

fundsdata_study$fund_annual_report_net_expense_ratio[is.na(fundsdata_study$fund_annual_report_net_expense_ratio)] <- median
(fundsdata_study$fund_annual_report_net_expense_ratio, na.rm=TRUE)

fundsdata_study$fund_prospectus_net_expense_ratio[is.na(fundsdata_study$fund_prospectus_net_expense_ratio)] <- median(fundsd
ata_study$fund_prospectus_net_expense_ratio, na.rm=TRUE)

fundsdata_study$fund_prospectus_gross_expense_ratio[is.na(fundsdata_study$fund_prospectus_gross_expense_ratio)] <- median(fu
ndsdata_study$fund_prospectus_gross_expense_ratio, na.rm=TRUE)

fundsdata_study$fund_max_12b1_fee[is.na(fundsdata_study$fund_max_12b1_fee)] <- median(fundsdata_study$fund_max_12b1_fee, na.
rm=TRUE)

fundsdata_study$fund_max_front_end_sales_load[is.na(fundsdata_study$fund_max_front_end_sales_load)] <- median(fundsdata_stud
y$fund_max_front_end_sales_load, na.rm=TRUE)

fundsdata_study$year_to_date_return[is.na(fundsdata_study$year_to_date_return)] <- mean(fundsdata_study$year_to_date_return,
na.rm=TRUE)

# verify missing value were imputed
colSums(is.na(fundsdata_study))
```

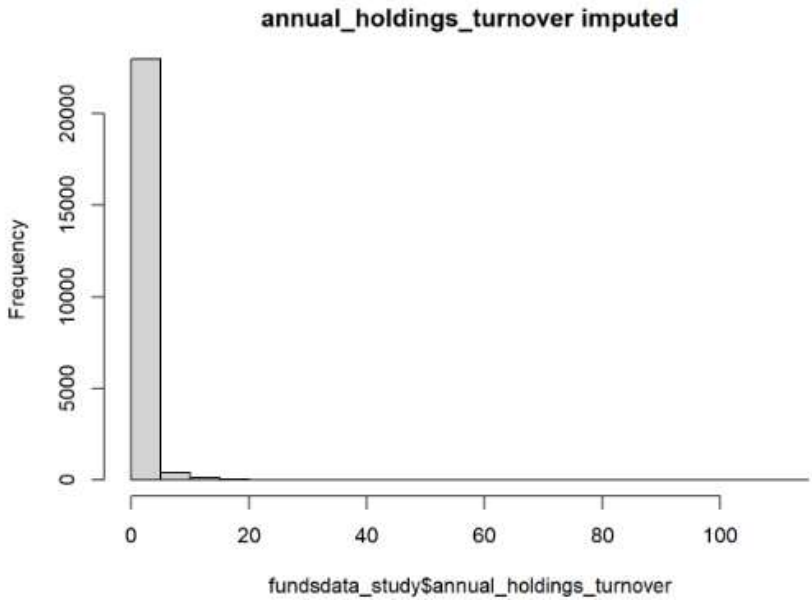
```
##                    total_net_assets             annual_holdings_turnover
##                                   0                                    0
## fund_annual_report_net_expense_ratio    fund_prospectus_net_expense_ratio
##                                   0                                    0
##  fund_prospectus_gross_expense_ratio                   fund_max_12b1_fee
##                                   0                                    0
##        fund_max_front_end_sales_load                year_to_date_return
##                                   0                                    0
```

The distribution of the variables was inspected after imputation.

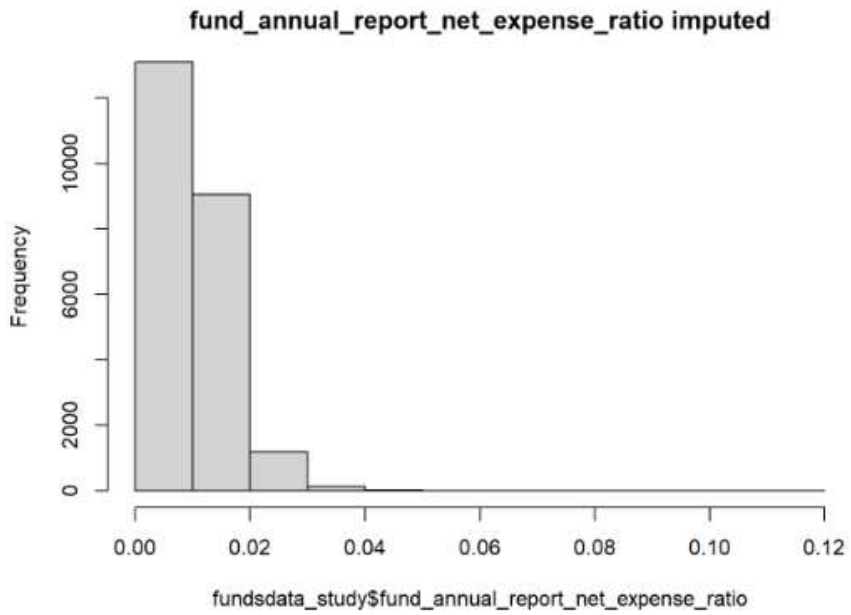
```
# verify distribution of data after imputation with new histograms
par(mar = c(5, 4, 2, 2))
hist(fundsdata_study$total_net_assets, main='total_net_assets imputed')
```



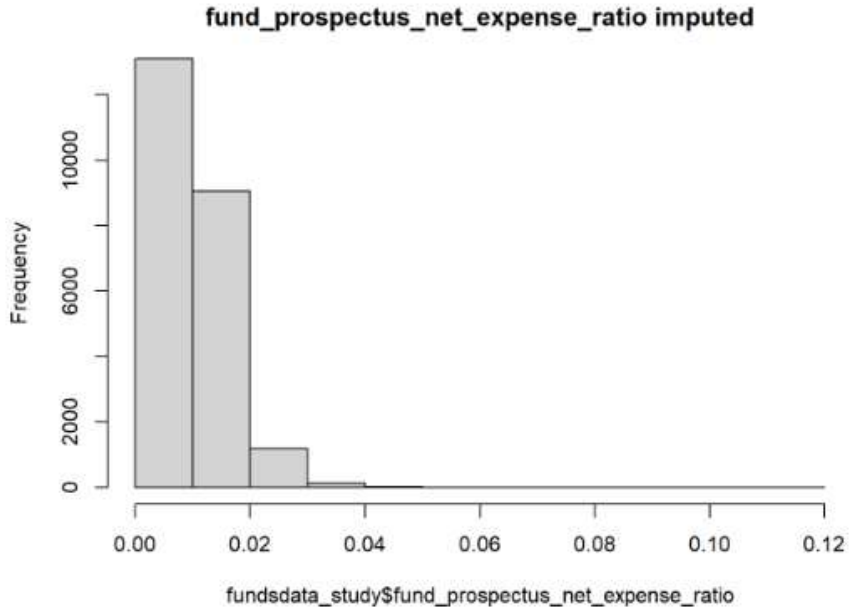

```
hist(fundsdata_study$annual_holdings_turnover, main='annual_holdings_turnover imputed')
```



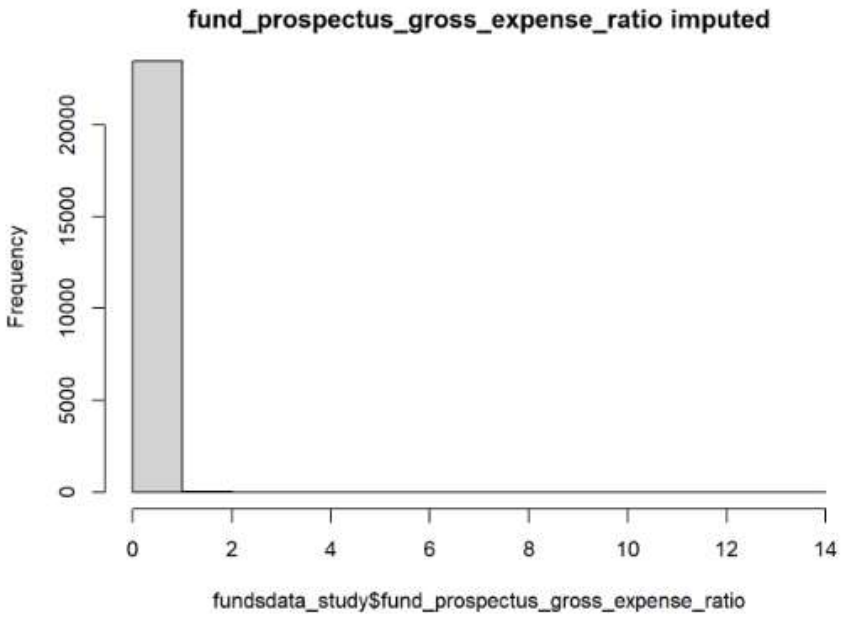
```
hist(fundsdata_study$fund_annual_report_net_expense_ratio, main='fund_annual_report_net_expense_ratio imputed')
```



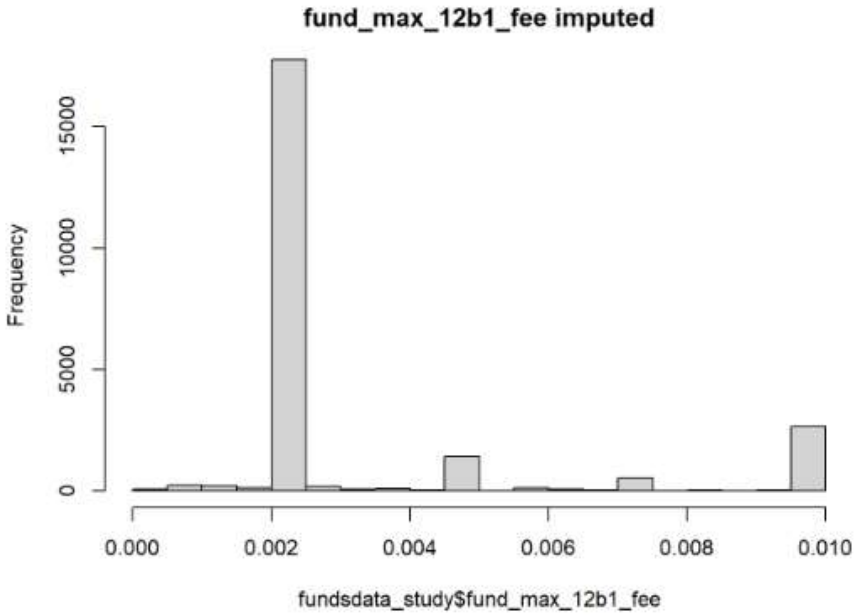

```
hist(fundsdata_study$fund_prospectus_net_expense_ratio, main='fund_prospectus_net_expense_ratio imputed')
```



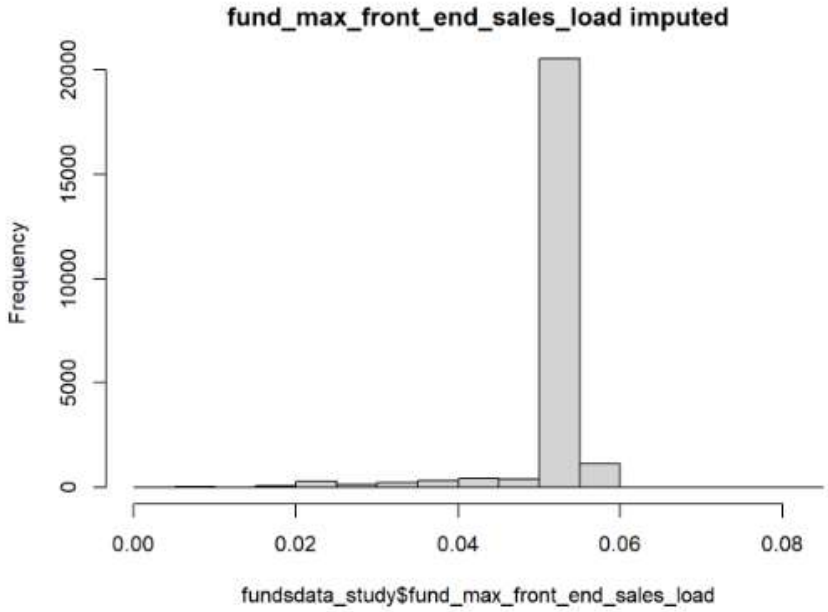
```
hist(fundsdata_study$fund_prospectus_gross_expense_ratio, main='fund_prospectus_gross_expense_ratio imputed')
```



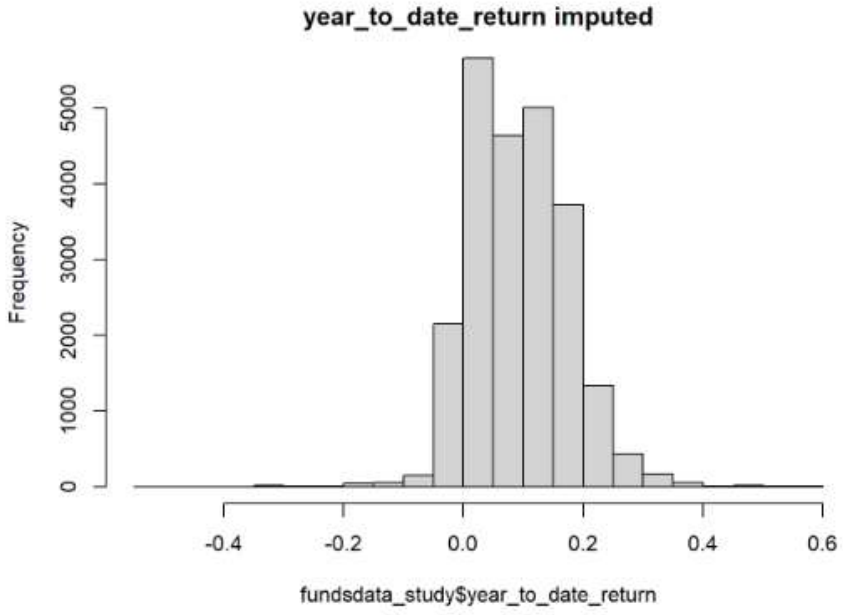

```
hist(fundsdata_study$fund_max_12b1_fee, main='fund_max_12b1_fee imputed')
```



```
hist(fundsdata_study$fund_max_front_end_sales_load, main='fund_max_front_end_sales_load imputed')
```




```
hist(fundsdata_study$year_to_date_return, main='year_to_date_return imputed')
```

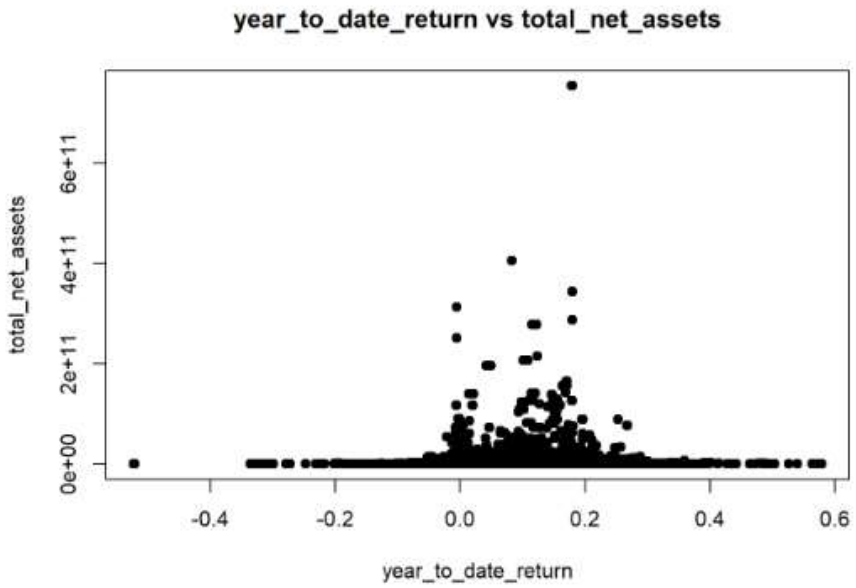


Part D: Analysis

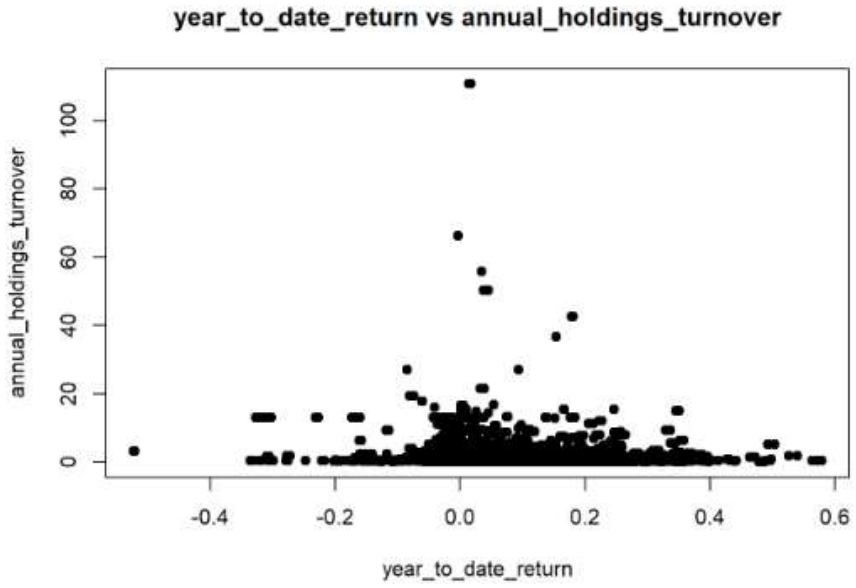
A multiple linear regression method was used to create a predictive model for the year_to_date_return variable. This method was selected to determine the strength of the relationships between the independent variables and the dependent variable. The first step was to check for linear relationships between the variables. This was done by generating a scatterplot of

each independent variable compared to the dependent variable. The scatterplot method was selected because it would help determine if a multiple linear regression would be a good model to use with this dataset. The independent variables do not appear to have a linear relationship with the dependent variable.

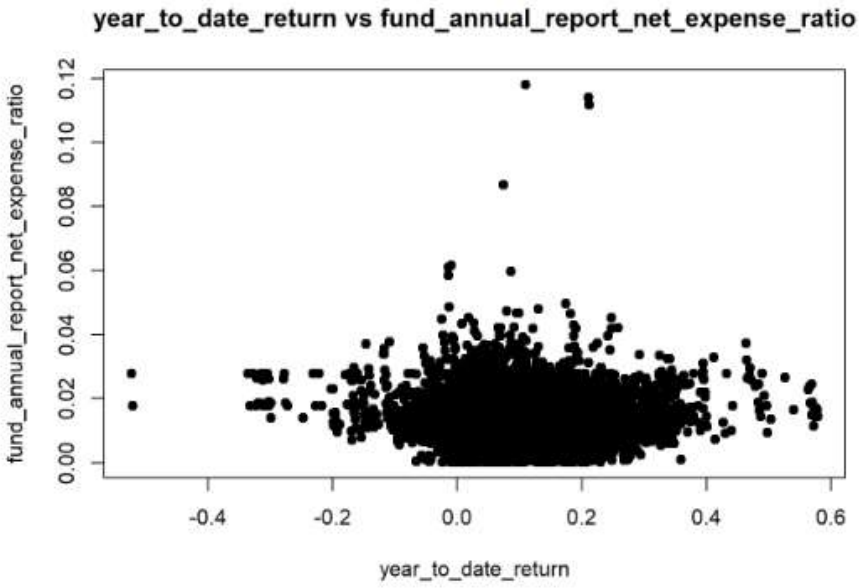
```
# scatterplots to inspect linearity
attach(fundsdata_study)
plot(year_to_date_return, total_net_assets, main="year_to_date_return vs total_net_assets",
     xlab="year_to_date_return", ylab="total_net_assets", pch=19)
```



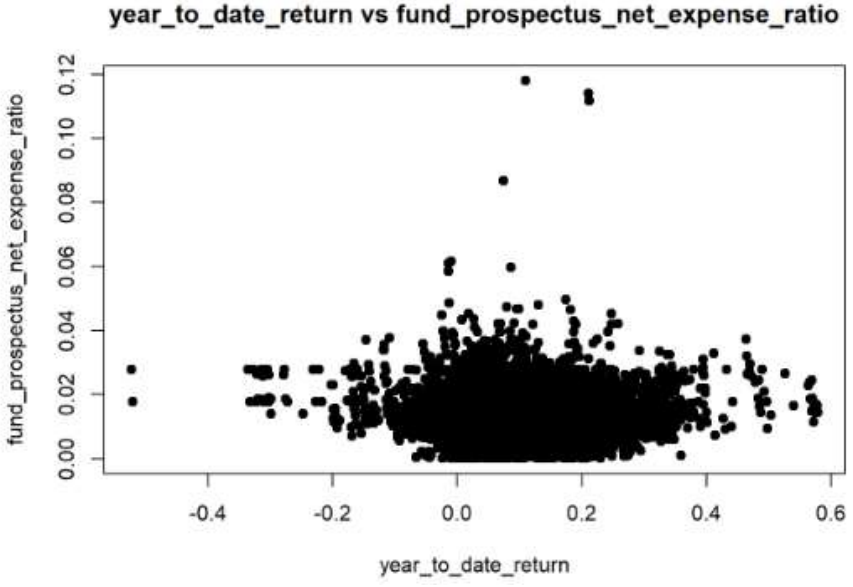

```
plot(year_to_date_return, annual_holdings_turnover, main="year_to_date_return vs annual_holdings_turnover",
     xlab="year_to_date_return", ylab="annual_holdings_turnover", pch=19)
```



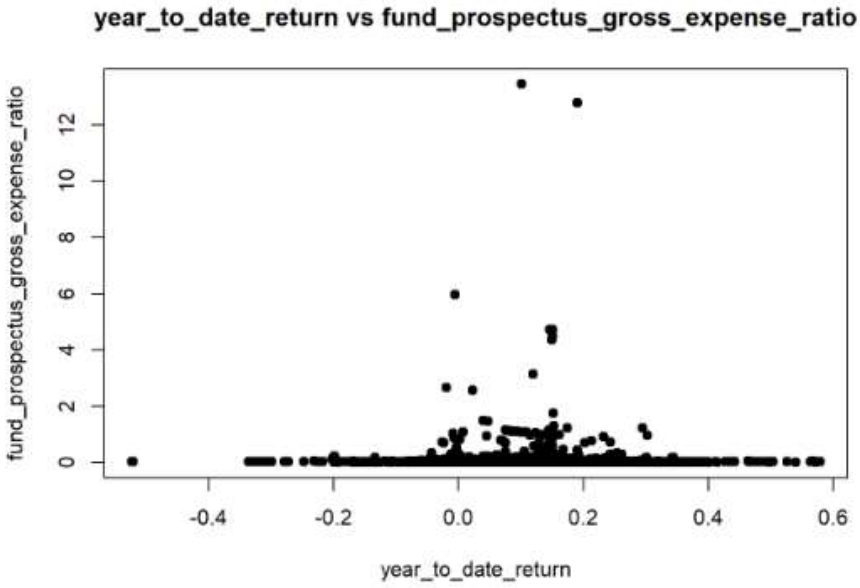
```
plot(year_to_date_return, fund_annual_report_net_expense_ratio, main="year_to_date_return vs fund_annual_report_net_expense_
ratio",
     xlab="year_to_date_return", ylab="fund_annual_report_net_expense_ratio", pch=19)
```



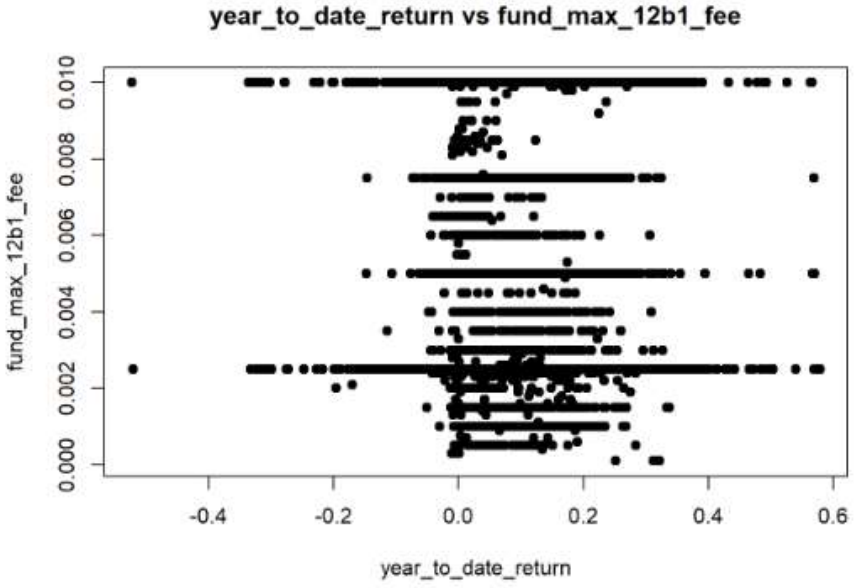

```
plot(year_to_date_return, fund_prospectus_net_expense_ratio, main="year_to_date_return vs fund_prospectus_net_expense_rati
o",
     xlab="year_to_date_return", ylab="fund_prospectus_net_expense_ratio", pch=19)
```



```
plot(year_to_date_return, fund_prospectus_gross_expense_ratio, main="year_to_date_return vs fund_prospectus_gross_expense_ratio",
     xlab="year_to_date_return", ylab="fund_prospectus_gross_expense_ratio", pch=19)
```




```
plot(year_to_date_return, fund_max_12b1_fee, main="year_to_date_return vs fund_max_12b1_fee",
     xlab="year_to_date_return", ylab="fund_max_12b1_fee", pch=19)
```



The initial multiple linear regression model was created using all the independent variables. The summary of the initial model was displayed to inspect the coefficients and residual standard error, as well as to determine which variables were not significant based on the t-value.

```
# initial model
model_initial <- lm(year_to_date_return ~ total_net_assets + annual_holdings_turnover + fund_annual_report_net_expense_ratio
+ fund_prospectus_net_expense_ratio + fund_prospectus_gross_expense_ratio + fund_max_12b1_fee + fund_max_front_end_sales_loa
d, data = fundsdata_study)

# initial model stats output
summary(model_initial)
```

```
##
## Call:
## lm(formula = year_to_date_return ~ total_net_assets + annual_holdings_turnover +
##     fund_annual_report_net_expense_ratio + fund_prospectus_net_expense_ratio +
##     fund_prospectus_gross_expense_ratio + fund_max_12b1_fee +
##     fund_max_front_end_sales_load, data = fundsdata_study)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62742 -0.06049 -0.00380  0.05116  0.46755
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         -4.207e-02  5.300e-03  -7.939 2.14e-15 ***
## total_net_assets                     1.294e-13  2.476e-14   5.225 1.75e-07 ***
## annual_holdings_turnover            -5.096e-03  2.488e-04 -20.484  < 2e-16 ***
## fund_annual_report_net_expense_ratio -7.913e-01  7.365e+00  -0.107   0.9144
## fund_prospectus_net_expense_ratio     3.225e+00  7.364e+00   0.438   0.6614
## fund_prospectus_gross_expense_ratio   5.681e-03  3.318e-03   1.712   0.0869 .
## fund_max_12b1_fee                   -4.782e+00  2.613e-01 -18.301  < 2e-16 ***
## fund_max_front_end_sales_load        2.547e+00  1.015e-01  25.093  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07925 on 23502 degrees of freedom
## Multiple R-squared:  0.05908,    Adjusted R-squared:  0.0588
## F-statistic: 210.8 on 7 and 23502 DF,  p-value: < 2.2e-16
```

The initial model error rate was calculated to determine the error of prediction. The initial model was found to have an error rate of 84.6%.

```
# calculate initial model error rate
sigma(model_initial)/mean(fundsdata_study$year_to_date_return)
```

```
## [1] 0.8461137
```

The next step was to generate a final model by only including those variables with a positive t-value. The summary of the final model was displayed to inspect the coefficients and residual standard error.

```
# remove insignificant variables to generate final model
model_final <- lm(year_to_date_return ~ total_net_assets + fund_prospectus_net_expense_ratio + fund_prospectus_gross_expense
_ratio + fund_max_front_end_sales_load, data = fundsdata_study)

# final model stats output
summary(model_final)
```

```
##
## Call:
## lm(formula = year_to_date_return ~ total_net_assets + fund_prospectus_net_expense_ratio +
##     fund_prospectus_gross_expense_ratio + fund_max_front_end_sales_load,
##     data = fundsdata_study)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63457 -0.06234 -0.00205  0.05291  0.47963
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -4.680e-02  5.374e-03  -8.708  < 2e-16 ***
## total_net_assets                   8.638e-14  2.500e-14   3.455 0.000552 ***
## fund_prospectus_net_expense_ratio  9.378e-01  9.165e-02  10.233  < 2e-16 ***
## fund_prospectus_gross_expense_ratio 6.191e-03  3.367e-03   1.839 0.065971 .
## fund_max_front_end_sales_load      2.520e+00  1.029e-01  24.489  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08043 on 23505 degrees of freedom
## Multiple R-squared:  0.03075,    Adjusted R-squared:  0.03059
## F-statistic: 186.4 on 4 and 23505 DF,  p-value: < 2.2e-16
```

The last step was to calculate the final model error rate to determine the error of prediction. The final model was found to have an error rate of 85.9%.

```
# calculate final model error rate
sigma(model_final)/mean(fundsdata_study$year_to_date_return)
```

```
## [1] 0.8586996
```

One advantage of using these techniques is that it provides detailed metrics that can be used further for determining model accuracy. One disadvantage of using these techniques is that it requires a manual iteration process to potentially produce a better model.

Part E: Data Summary and Implications

While the initial model error rate of 84.6% was better than the final model, the error rate was very high. This indicates that neither model was a good predictor for the dependent variable, therefore we fail to reject the null hypothesis.

This analysis is limited by the number of independent variables used.

Based on these results, it is recommended that the models not be used for predicting values of the dependent variable.

Part F: Sources

Dataset:

US Funds dataset from Yahoo Finance. (2021, December 11). Kaggle. Retrieved May 19, 2024 from https://www.kaggle.com/datasets/stefanoleone992/mutual-funds-and-etfs

Sources:

Intrinio, B. (2024b, January 22). How to predict stock prices using linear Regression. Retrieved May 19, 2024 from https://intrinio.com/blog/how-to-predict-stock-prices-using-linear-regression