Springboard - Data Science Capstone Project

Rucha Joshi

August 5, 2018

Maritime Data Intelligence System - Predicting the revenue of Ship Chartering Business

Introduction:

Shipping has been key in facilitating trade and globalization. It is one of the largest industry in the world. This sector handles roughly 60% of the value and 75% of the tonnage of world trade according to Lloyd's MIU.
Chartering is an activity in the shipping industry whereby a shipowner hires out the use of his vessel to a charterer. The contract between the parties is called a "charter party".
A voyage charter is the hiring of a vessel and crew for a voyage between a load port and a discharge port. The charterer pays the vessel owner on a per-ton or lump-sum basis. The owner pays the port costs (excluding stevedoring), fuel costs and crew costs. The payment for the use of the vessel is known as freight. In some cases a charterer may own cargo and employ a shipbroker to find a ship to deliver the cargo for a certain price, called freight rate. Freight rates may be on a per-ton basis over a certain route (e.g. for iron ore between Brazil and China), in World scale points (in case of oil tankers) or alternatively may be expressed in terms of a total sum - normally in U.S. dollars - per day for the agreed duration of the charter.A charterer may also be a party without a cargo who takes a vessel on charter for a specified period from the owner and then trades the ship to carry cargoes at a profit above the hire rate, or even makes a profit in a rising market by re-letting the ship out to other charterers.
Depending on the type of ship and the type of charter, normally a standard contract form called a charter party is used to record the exact rate, duration and terms agreed between the shipowner and the charterer.
Time Charter Equivalent is a standard shipping industry performance measure used primarily to compare period-to-period changes in a shipping company's performance despite changes in the mix of charter types.

The objective of this project is to maximize the revenues in ship chartering/contracting business through efficient operational planning and optimization of resources by analyzing their previous years data.

Problem Statement:

Shipping industry is very competitive, in addition to competing with each other, they must fight with all four modes of freight transportation: air, rail, water and road. Recently these conscious shippers are getting more aware of all the costs by analyzing their history data to make more precise decisions.
Prediction of the revenue on the particular charter by creating a machine learning model based on the previous year's available data will help the shipper to choose best trades to get the maximum profit.
Our goal in this analysis is to help shipowners to choose the charter wisely to maximize their profits by identifying the cost affecting factors on the voyage.
We will build the model that can explain which factors have predictive power.

Data Set:

The dataset was received from the ship chartering company.
The dataset includes about 185,000 observations which has voyages from 2009 across the world. The other datasets are Port details, Vessel details and Voyage details.

The data includes following important fields -
1. Voyage Id - Id assigned to each voyage
2. Vessel Id - Id assigned to each vessel
3. Commencing Date - Start date for the voyage
4. Completing Date - End date for the voyage
5. Vessel types - There are total 12 different types of vessels.
6. Vessel description - Is vessel a Tanker or a Bulk
7. Cargo types- There are 20 unique values of cargo types.
8. First load port - This is the port where first vessel got loaded.
9. Last discharge port - This is the last discharge port for the voyage. There can be multiple discharge ports for single voyage.
10. Cargo type - It gives information about what type of cargo vessel is carrying.
11. Cargo lift - It tells us how much cargo the vessel is carrying.
12. Dwt - Deadweight tonnage of the ship.
13. Estimated earnings and Actual Earnings
14. Trade Area - The area where the voyage is travelling.
15. Estimated voyage days and Actual voyage days
16. Ballast days - when vessel is empty
17. Laden days - when vessel is loaded
18. Base Currency - Currency of earnings
19. Operators details
20. Laden and Ballast Speed - this is finalized at the charter party.
21. Vessel properties such as built year, owner etc.

And there are few more fields for internal use of the shippers which are not relevant to our analysis.

Data Limitations:

Some of the observations has missing values as all this data has been manually entered in the system.
The data contains cancelled voyages as well so we need separate that data as we don't have more details about those voyages we can not consider those cancellations reasons for our prediction. Weather is one of the major factors in the earnings of these voyages. We have not received any weather data as fuel consumption depends on the weather and it is a major factor of daily expenses of a voyage.
Any conclusions from this study should take into account that including these factors may have led to different results.

Data Wrangling:

The dataset was comprehensive with few missing values. It required some cleanup and reformatting. The steps taken are described below. For detailed review of the code, R markdown document explaining each step with code is included.

Rows which were not needed for analysis were removed for example observations which had Id = -1 were removed from the dataset. In addition, column headings in the data set needed to be transformed from ambiguous names which gives clear description of each variable.

The dataset imported into R was stored in the data frame where I checked for NA's and spaces and tried to remove factor levels. These transformations were helpful to conduct preliminary exploration and data visualization.

Now data had lot of categorical variables like vessel type, cargo type; so, I tried to combine similar types into one category so that we can have limited categories to find the trends in the data.

Similarly, first loading port, last discharge port and trade area were replaced with region names so that we can find which are the regions gives more earnings and which trade area has more earnings.
In the ports data few observations had 0 longitude and 0 latitude, so we decided to remove these records.

For a single voyage there are multiple records with different data types as A - actual, E - Estimate and P-positional. We decided to use actual data for our analysis as estimate and positional type had lot of missing values.

If first loading port and/or last discharge port has blank value or actual earnings has zero value then those voyages got cancel due to some reason.

After doing all this cleaning and reformatting the data is ready for preliminary exploration.

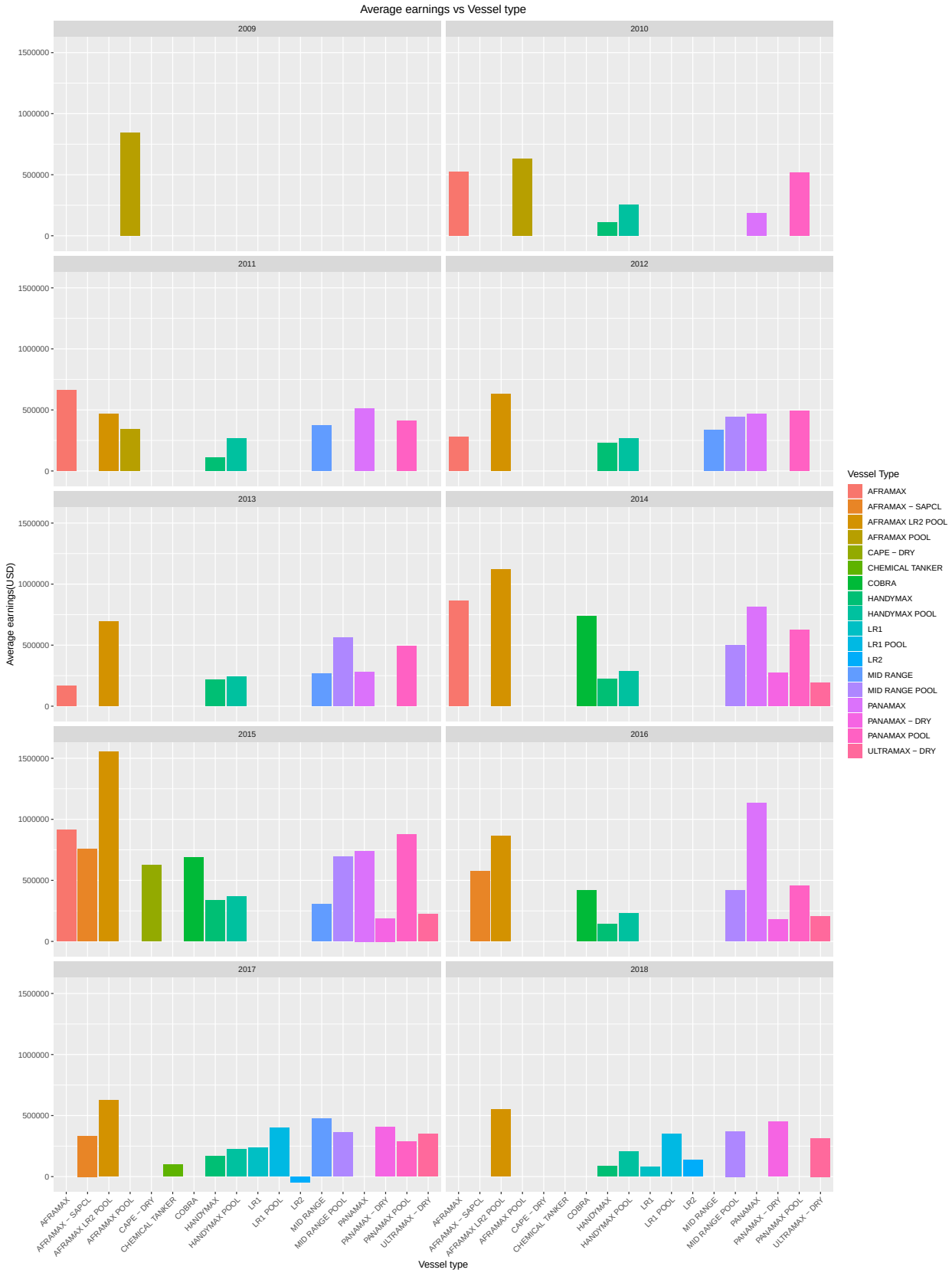
Preliminary Exploration:

The goal of preliminary exploration was to find independent variables that appear to have some predictive power.

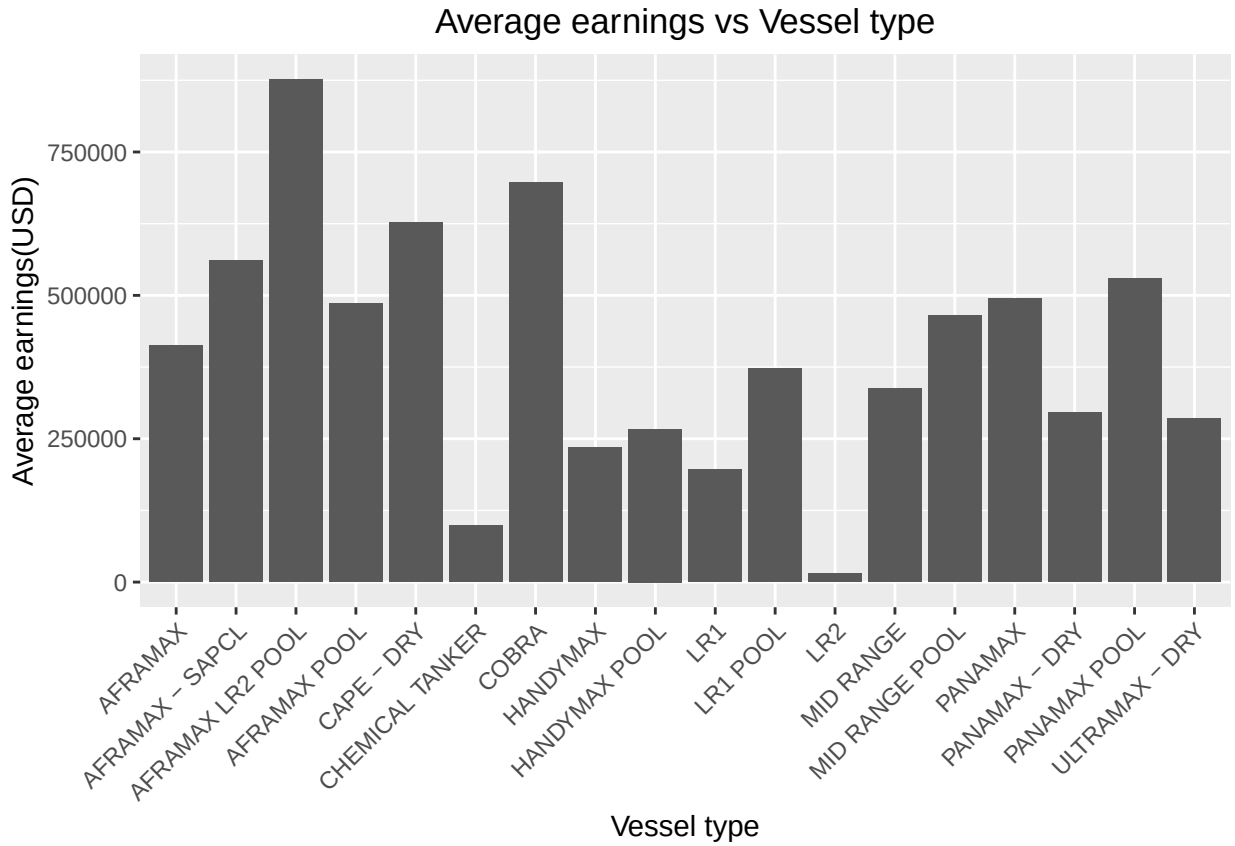
I. First we tried to find the top earning vessels by plotting sum of the earnings per year for each vessel type.

It shows that every year Handymax, Midrange are having high earnings. Chemical tanker is in business in only 2017. Cobra was in business just for 3 years from 2014, 2015, 2016.
From the plots, out of the total earnings of all the vessel types for all the years, about 80% of the earnings are from Midrange, Handymax, Panamax and Aframax. In 2017 out of $28 million business $23 million business is done by Midrange, Handymax, Aframax.

II. The next plot is average of the earnings per year for all vessel types.

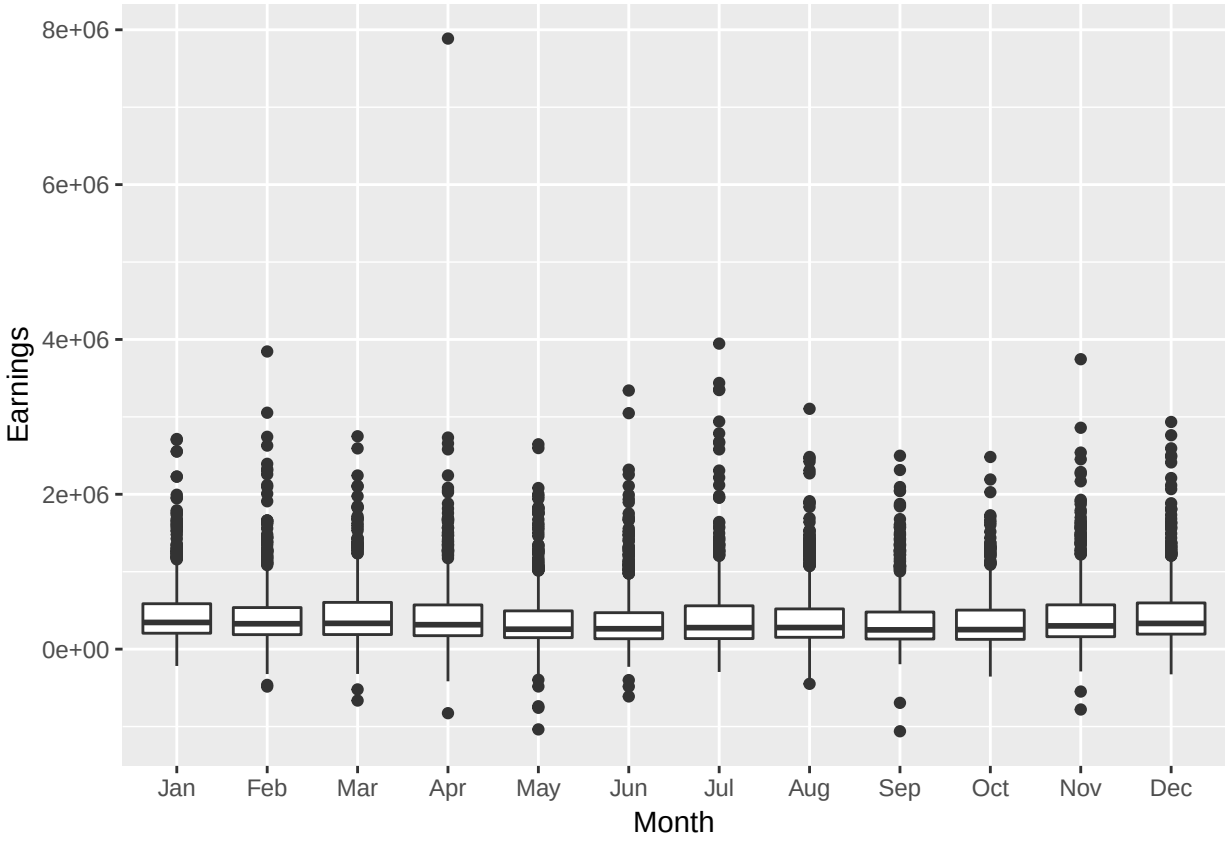


III. Next plots give average earnings for each vessel type for all the years.



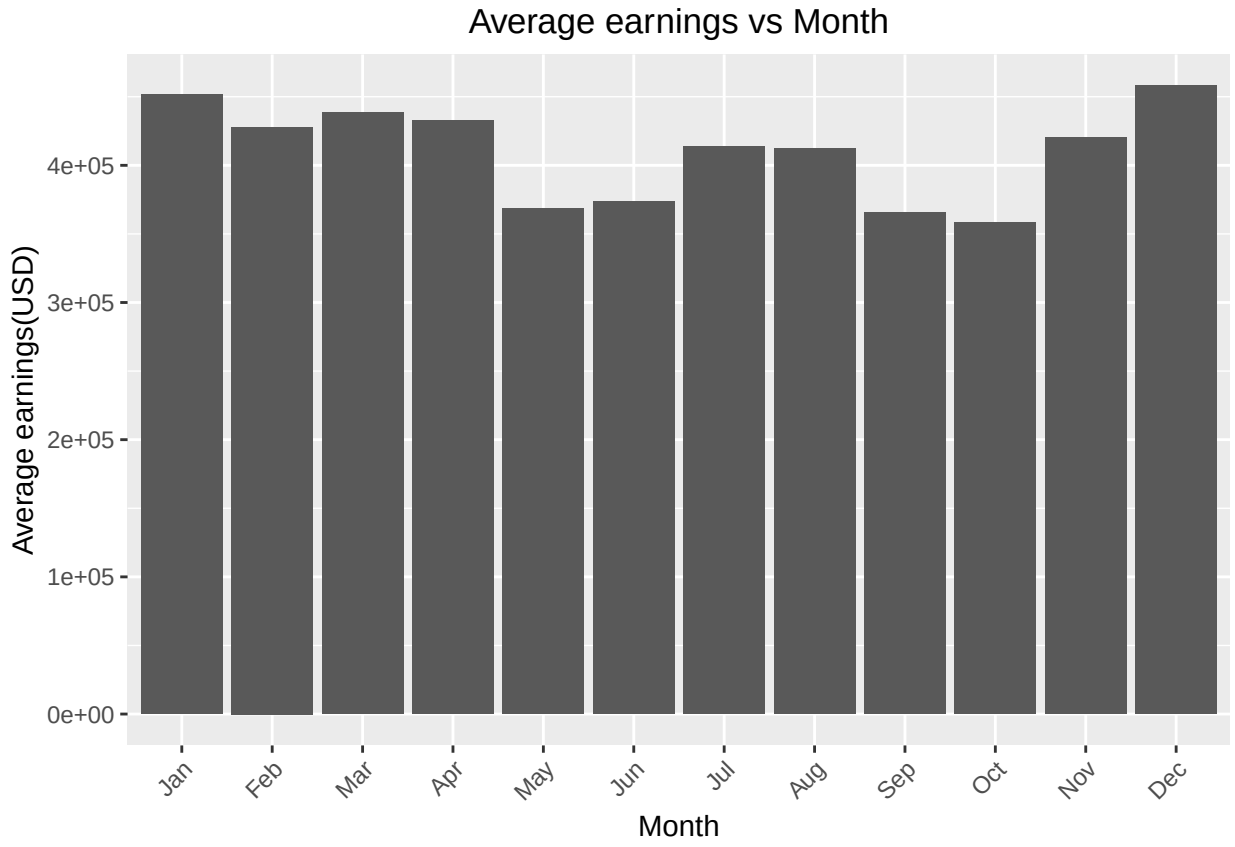
Here it is showing Cobra has highest earnings but Cobra was used just for 3 years (2014,2015,2016). This plot is not so useful as some vessel types has data for 10 years and some are used for couple of years whereas some are recent in the business. It will not be fare to compare these results to find the best one.

IV. I used box plot to plot monthly earning distribution to find out seasonality.



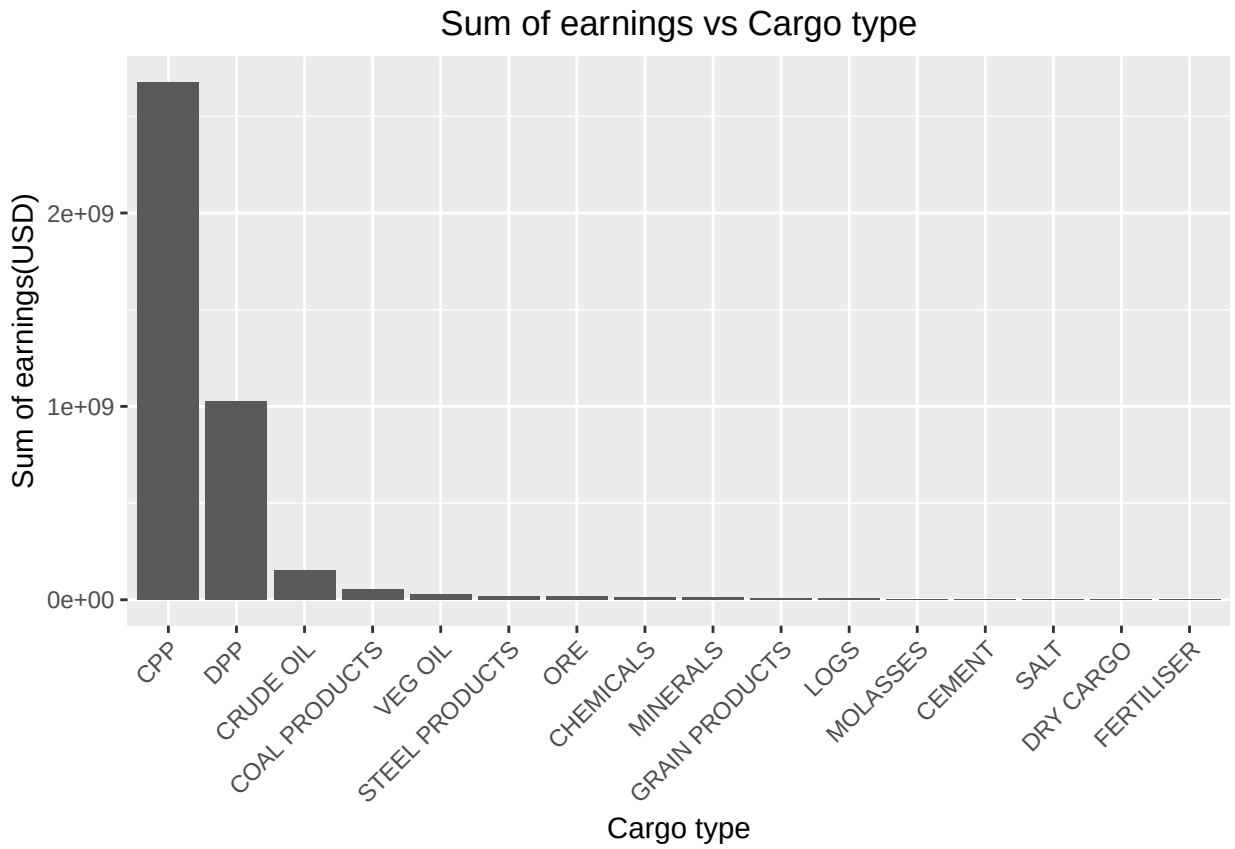
Plot does not show much of seasonality.

Plot showing Month Vs Average earnings



This shows the whole year business is going on.

V. This plot has been plotted to find top cargo types from sum of earnings.



First 3 cargo types are generating 90% of the business.

Based on this analysis, earnings of these voyages are dependent on mainly below mentioned factors:
1. Vessel type - there are different types of vessels; most profitable are MIDRANGE, HANDYMAX, AFRAMAX, PANAMAX
2. Cargo type- there are mainly two categories of cargo - dry cargo (coal, iron etc.) and liquid cargo (crude oil, gasoline, vegetable oil etc.). Most profitable are FO, CPP, ULSD, Gasoline, Naphtha and Gas Oil etc.
3. Dwt - this is the capacity of the vessel.
4. Actual Cargo lift - How much cargo the particular vessel is carrying affects the earnings.
5. Month - The period of the year in which this voyage is happening also affects the total earnings.
6. Region - the region in which the voyage is starting, loading and discharging also affect its earnings. Most profitable region is Transatlantic area.
7. Daily expense - It includes their daily fuel usage, port fees, jetty charges, food for crew etc. Their 60% of expense is on fuel.

We can say from all of the above plots that voyages with vessel types like MIDRANGE, HANDYMAX carrying FO, CPP, ULSD, Gasoline, Naphtha and Gas Oil etc. in the transatlantic area have more earnings making the business profitable.

Machine Learning:

In this section we built model that predict the earnings of the voyage. We built a linear regression model.

Linear Regression:

Linear regression is used to predict the value of an outcome variable Y based on one or more input predictor variables X. The aim is to establish a linear relationship (a mathematical formula) between the predictor variable(s) and the response variable, so that, we can use this formula to estimate the value of the response Y, when only the predictors (Xs) values are known.

We first split the data into training dataset and testing dataset. The training dataset was used to build the model to find predictive relationship. The testing data set was used to assess the performance of the model.

The training dataset used 70% of the observations and testing data set used 30% of the observations. The set.seed function was used to make sure that dependent variable was balanced in both datasets.

```
#Split the data into training dataset and testing dataset with a 70/30 ratio.
set.seed(666)
split <- sample(nrow(Final), floor(0.7*nrow(Final)))
Final_train <- Final[split,]
Final_test <- Final[-split,]
```

Check the rows for training and testing the datasets

```
#Check the rows for training and testing the datasets
nrow(Final_train)
```

```
## [1] 6858
```

```
nrow(Final_test)
```

```
## [1] 2940
```

We can see that, there are number of observations in the training dataset and observations in the testing dataset are consistent with 70/30 ratio.

We used forward selection approach in building the linear regression model, which starts with no predictors in the model, iteratively adds the most contributive predictors, and stops when the improvement is no longer statistically significant.

Build a linear model.

Model 1 -

Model1 starts with fewer independent variables.

```
# Model1 starts with fewer independent variables.
model1 <- lm(Earnings~VesselType + Duration, Final_train)

#Linear Regression Diagnostics
summary(model1)
```

```
##
## Call:
## lm(formula = Earnings ~ VesselType + Duration, data = Final_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1942378  -129259   -14541   109785  2696652
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -45910.7    29719.7  -1.545  0.12244
## VesselTypeAFRAMAX - SAPCL    154598.4    67247.2   2.299  0.02154 *
## VesselTypeAFRAMAX LR2 POOL   287766.2    31501.1   9.135  < 2e-16 ***
## VesselTypeAFRAMAX POOL        85178.2    69692.8   1.222  0.22168
## VesselTypeCAPE - DRY         273419.9   299298.9   0.914  0.36099
## VesselTypeCHEMICAL TANKER   -391757.5   299309.4  -1.309  0.19062
## VesselTypeCOBRA              318212.9    59257.2   5.370 8.13e-08 ***
## VesselTypeHANDYMAX            27578.7    32414.2   0.851  0.39490
## VesselTypeHANDYMAX POOL       35923.1    29473.2   1.219  0.22295
## VesselTypeLR1               -437480.3   174413.2  -2.508  0.01215 *
## VesselTypeLR1 POOL          -178121.1    44809.2  -3.975 7.11e-05 ***
## VesselTypeLR2               -656828.0   116240.0  -5.651 1.66e-08 ***
## VesselTypeMID RANGE          -31330.2    50625.1  -0.619  0.53602
## VesselTypeMID RANGE POOL      89824.0    29373.6   3.058  0.00224 **
## VesselTypePANAMAX            -15008.5    45928.4  -0.327  0.74384
## VesselTypePANAMAX - DRY     -327148.7    38061.0  -8.595  < 2e-16 ***
## VesselTypePANAMAX POOL        62860.2    31060.6   2.024  0.04303 *
## VesselTypeULTRAMAX - DRY    -205404.5    42532.0  -4.829 1.40e-06 ***
## Duration                      13758.9      226.4  60.767  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 297900 on 6839 degrees of freedom
## Multiple R-squared:  0.4702, Adjusted R-squared:  0.4688
## F-statistic: 337.1 on 18 and 6839 DF,  p-value: < 2.2e-16
```

Model 2 -

```r
# Model 2 adding more independent variables.
model2 <- lm(Earnings~VesselType + Duration + Cargo + Qty, Final_train)

#Linear Regression Diagnostics
summary(model2)
```

```
##
## Call:
## lm(formula = Earnings ~ VesselType + Duration + Cargo + Qty,
##     data = Final_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1958675  -129893   -12766   110928  2741261
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -1.210e+05  1.117e+05  -1.084 0.278568
## VesselTypeAFRAMAX - SAPCL    1.462e+05  6.780e+04   2.157 0.031075 *
## VesselTypeAFRAMAX LR2 POOL   2.699e+05  3.167e+04   8.521  < 2e-16 ***
## VesselTypeAFRAMAX POOL       8.235e+04  6.956e+04   1.184 0.236520
## VesselTypeCAPE - DRY         1.488e+05  2.987e+05   0.498 0.618336
## VesselTypeCHEMICAL TANKER   -3.172e+05  2.986e+05  -1.062 0.288168
## VesselTypeCOBRA              3.786e+05  6.084e+04   6.223 5.16e-10 ***
## VesselTypeHANDYMAX           8.351e+04  3.453e+04   2.418 0.015620 *
## VesselTypeHANDYMAX POOL      9.265e+04  3.184e+04   2.910 0.003629 **
## VesselTypeLR1               -4.060e+05  1.747e+05  -2.324 0.020139 *
## VesselTypeLR1 POOL          -1.586e+05  4.528e+04  -3.503 0.000462 ***
## VesselTypeLR2               -6.862e+05  1.159e+05  -5.920 3.38e-09 ***
## VesselTypeMID RANGE          1.948e+04  5.167e+04   0.377 0.706108
## VesselTypeMID RANGE POOL     1.380e+05  3.145e+04   4.389 1.16e-05 ***
## VesselTypePANAMAX            2.373e+04  4.642e+04   0.511 0.609139
## VesselTypePANAMAX - DRY     -3.114e+05  3.845e+04  -8.099 6.52e-16 ***
## VesselTypePANAMAX POOL       9.149e+04  3.174e+04   2.883 0.003957 **
## VesselTypeULTRAMAX - DRY    -1.558e+05  4.353e+04  -3.580 0.000346 ***
## Duration                     1.364e+04  2.265e+02  60.225  < 2e-16 ***
## CargoCHEMICALS              -8.342e+04  1.308e+05  -0.638 0.523783
## CargoCOAL PRODUCTS          -2.005e+04  1.102e+05  -0.182 0.855642
## CargoCPP                    -1.105e+04  1.061e+05  -0.104 0.917094
## CargoCRUDE OIL              -9.499e+04  1.074e+05  -0.885 0.376363
## CargoDPP                    -1.885e+04  1.063e+05  -0.177 0.859267
## CargoDRY CARGO              -7.297e+04  1.826e+05  -0.400 0.689383
## CargoFERTILISER             -2.947e+05  1.825e+05  -1.614 0.106485
## CargoGRAIN PRODUCTS         -1.254e+05  1.262e+05  -0.994 0.320424
## CargoLOGS                   -1.210e+05  1.390e+05  -0.870 0.384062
## CargoMINERALS               -4.941e+04  1.262e+05  -0.392 0.695414
## CargoMOLASSES               -1.213e+05  1.417e+05  -0.856 0.391780
## CargoORE                    -6.836e+04  1.213e+05  -0.563 0.573240
## CargoSALT                   -1.222e+05  1.610e+05  -0.759 0.448011
## CargoSTEEL PRODUCTS         -4.091e+04  1.164e+05  -0.352 0.725142
## CargoVEG OIL                 6.483e+04  1.162e+05   0.558 0.577031
## Qty                          1.281e+00  2.132e-01   6.009 1.96e-09 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 296700 on 6823 degrees of freedom
## Multiple R-squared:  0.4757, Adjusted R-squared:  0.4731
## F-statistic: 182.1 on 34 and 6823 DF,  p-value: < 2.2e-16
```

Model 3 -

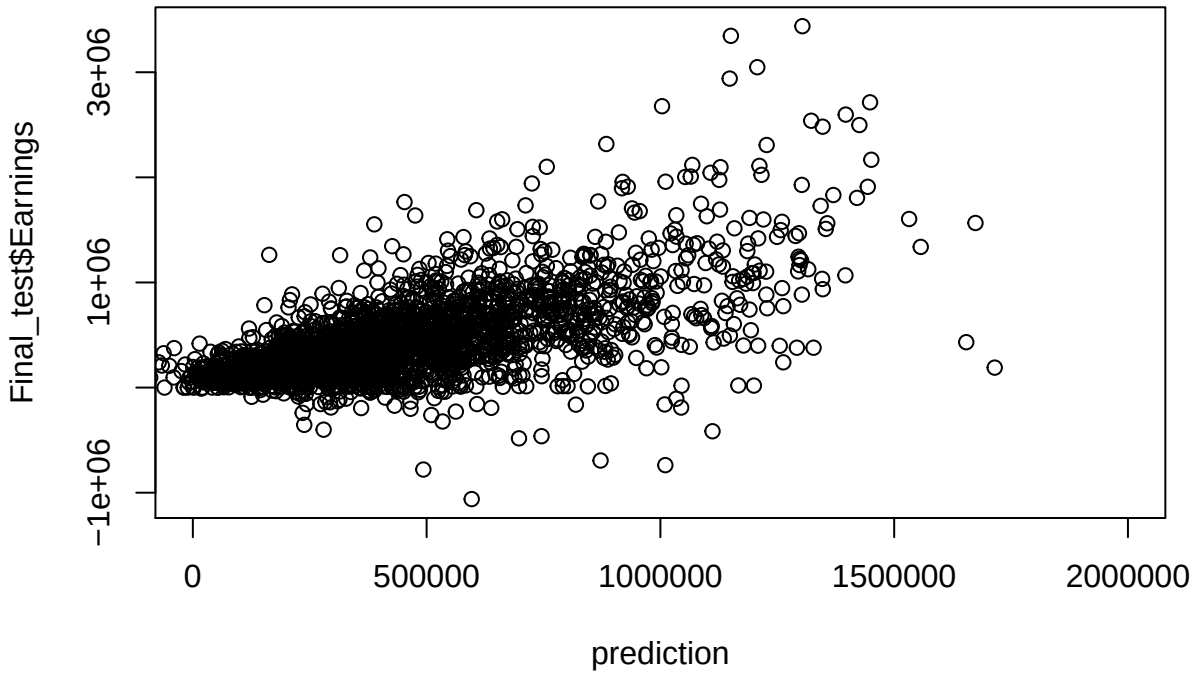
```
# Model 3 adding more independent variables.
model3 <- lm(Earnings~VesselType + Duration + Cargo + Qty + Month +
            DischargeArea + Dwt, Final_train)

#Linear Regression Diagnostics
summary(model3)
```

```
##
## Call:
## lm(formula = Earnings ~ VesselType + Duration + Cargo + Qty +
##     Month + DischargeArea + Dwt, data = Final_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1851055  -129014    -9366   110322  2876851
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.050e+05  2.385e+05   1.698  0.08962 .
## VesselTypeAFRAMAX - SAPCL   1.643e+05  6.657e+04   2.468  0.01362 *
## VesselTypeAFRAMAX LR2 POOL  3.049e+05  3.141e+04   9.708  < 2e-16 ***
## VesselTypeAFRAMAX POOL      6.054e+04  6.841e+04   0.885  0.37626
## VesselTypeCAPE - DRY        5.134e+05  3.193e+05   1.608  0.10795
## VesselTypeCHEMICAL TANKER  -6.964e+05  3.356e+05  -2.075  0.03803 *
## VesselTypeCOBRA             6.487e+04  1.186e+05   0.547  0.58443
## VesselTypeHANDYMAX         -2.282e+05  1.287e+05  -1.773  0.07630 .
## VesselTypeHANDYMAX POOL    -2.285e+05  1.274e+05  -1.794  0.07293 .
## VesselTypeLR1              -4.990e+05  1.808e+05  -2.760  0.00580 **
## VesselTypeLR1 POOL         -3.105e+05  7.376e+04  -4.210 2.59e-05 ***
## VesselTypeLR2              -6.646e+05  1.137e+05  -5.844 5.32e-09 ***
## VesselTypeMID RANGE        -2.658e+05  1.140e+05  -2.331  0.01976 *
## VesselTypeMID RANGE POOL   -1.502e+05  1.067e+05  -1.408  0.15921
## VesselTypePANAMAX          -1.637e+05  8.231e+04  -1.989  0.04673 *
## VesselTypePANAMAX - DRY    -4.094e+05  5.957e+04  -6.873 6.86e-12 ***
## VesselTypePANAMAX POOL     -9.348e+04  7.031e+04  -1.330  0.18371
## VesselTypeULTRAMAX - DRY   -3.787e+05  9.374e+04  -4.040 5.42e-05 ***
## Duration                    1.280e+04  2.293e+02  55.833  < 2e-16 ***
## CargoCHEMICALS             -5.669e+04  1.280e+05  -0.443  0.65788
## CargoCOAL PRODUCTS          6.091e+03  1.078e+05   0.057  0.95494
## CargoCPP                    9.491e+03  1.039e+05   0.091  0.92719
## CargoCRUDE OIL             -7.123e+04  1.051e+05  -0.678  0.49786
## CargoDPP                    6.779e+03  1.040e+05   0.065  0.94805
## CargoDRY CARGO             -5.308e+04  1.787e+05  -0.297  0.76639
## CargoFERTILISER            -2.364e+05  1.786e+05  -1.323  0.18579
## CargoGRAIN PRODUCTS        -1.137e+05  1.235e+05  -0.921  0.35710
## CargoLOGS                  -1.246e+05  1.360e+05  -0.916  0.35956
## CargoMINERALS              -5.026e+03  1.235e+05  -0.041  0.96753
## CargoMOLASSES              -8.339e+04  1.386e+05  -0.601  0.54753
## CargoORE                   -4.203e+04  1.188e+05  -0.354  0.72340
## CargoSALT                  -9.294e+04  1.575e+05  -0.590  0.55510
## CargoSTEEL PRODUCTS        -1.759e+04  1.139e+05  -0.154  0.87726
## CargoVEG OIL                8.550e+04  1.137e+05   0.752  0.45229
## Qty                         1.307e+00  2.095e-01   6.240 4.64e-10 ***
```

```
## Month.L                    -3.214e+04  1.216e+04  -2.644  0.00822 **
## Month.Q                     5.741e+04  1.221e+04   4.703 2.61e-06 ***
## Month.C                     5.373e+04  1.210e+04   4.441 9.10e-06 ***
## Month^4                     3.957e+04  1.216e+04   3.253  0.00115 **
## Month^5                     2.964e+04  1.224e+04   2.422  0.01544 *
## Month^6                    -1.166e+04  1.233e+04  -0.946  0.34423
## Month^7                    -3.807e+04  1.217e+04  -3.129  0.00176 **
## Month^8                     2.247e+03  1.214e+04   0.185  0.85319
## Month^9                     1.371e+04  1.219e+04   1.124  0.26099
## Month^10                   -1.663e+04  1.218e+04  -1.365  0.17216
## Month^11                    8.598e+03  1.229e+04   0.700  0.48422
## DischargeAreaFAR EAST      -2.507e+04  9.285e+04  -0.270  0.78714
## DischargeAreaIOR           -9.887e+04  9.300e+04  -1.063  0.28773
## DischargeAreaLATIN AMERICA  1.044e+05  9.392e+04   1.112  0.26614
## DischargeAreaMED           -6.764e+04  9.232e+04  -0.733  0.46378
## DischargeAreaNORTH AMERICA  1.101e+04  9.245e+04   0.119  0.90516
## DischargeAreaNW EUROPE     -5.771e+04  9.221e+04  -0.626  0.53142
## DischargeAreaOTHERS         2.961e+03  1.091e+05   0.027  0.97835
## DischargeAreaWAF            1.232e+05  9.299e+04   1.325  0.18513
## Dwt                        -4.584e+00  1.767e+00  -2.595  0.00949 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 290000 on 6803 degrees of freedom
## Multiple R-squared:  0.5005, Adjusted R-squared:  0.4966
## F-statistic: 126.2 on 54 and 6803 DF,  p-value: < 2.2e-16
```

Prediction:

```
# Using model 3 to predict the test dataset.
prediction <- predict(model3, Final_test)
```



Linear regression shows how an independent variable may be predictive for a specific outcome, it's difficult to understand which factors are most important, and to evaluate what the prediction will be for a new case.

So we decided to use Random Forest algorithm to get more precise results.

Random Forest:

Random Forest is a type of ensemble learning of Supervised Learning Technique. The idea is to generate multiple models on training dataset and then combining the output rule to generage a strong model which performs very well and does not overfits and balances the Bias-Variance tradeoff too.

We used same training and testing data as with Linear Regression.

Model using random forest algorithm

```
#Divide into training & test datasets
set.seed(40)
split <- sample(nrow(Final), floor(0.7*nrow(Final)))
train <- Final[split,]
test <- Final[-split,]

#Random Forest model
model <- randomForest(Earnings ~ Distance+Duration+Month+LoadArea+DischargeArea+Cargo+Qty
                      +VesselType+Dwt,
                      data=train,
                      ntree=500,
                      mtry=4,
                      importance=TRUE,
                      na.action = na.roughfix,
                      replace=FALSE)

model
```

```
##
## Call:
##  randomForest(formula = Earnings ~ Distance + Duration + Month +      LoadArea + DischargeArea + Carg
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 71495772137
##                     % Var explained: 55.57
```

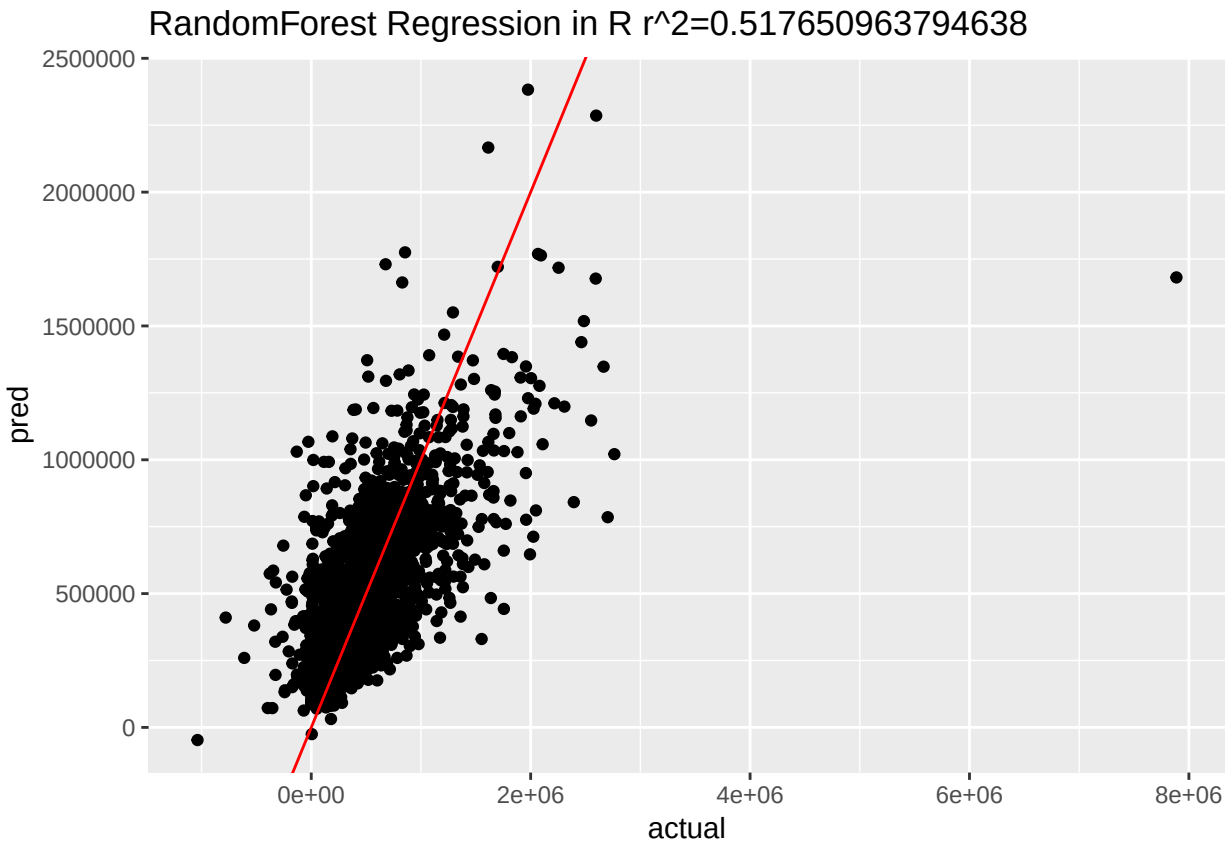
```
(r2 <- rSquared(test$Earnings, test$Earnings - predict(model, test)))
```

```
##          [,1]
## [1,] 0.517651
```

```
(mse <- mean((test$Earnings - predict(model, test))^2))
```

```
## [1] 79049552644
```

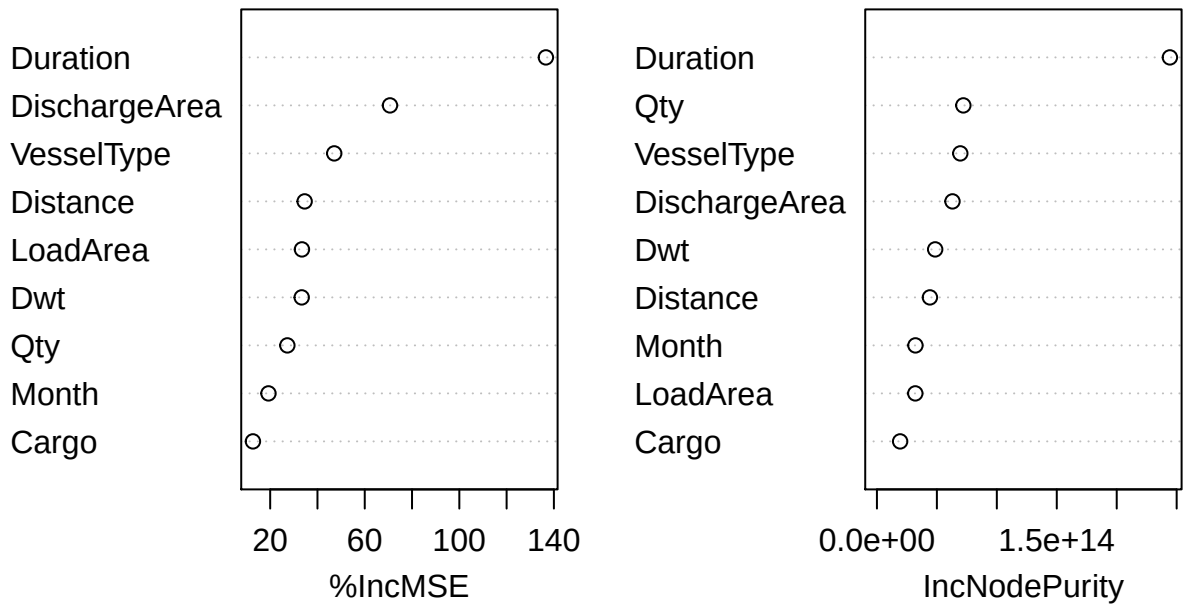
```
p <- ggplot(aes(x=actual, y=pred),
            data=data.frame(actual=test$Earnings, pred=predict(model, test)))
p + geom_point() +
  geom_abline(color="red") +
  ggtitle(paste("RandomForest Regression in R r^2=", r2, sep=""))
```



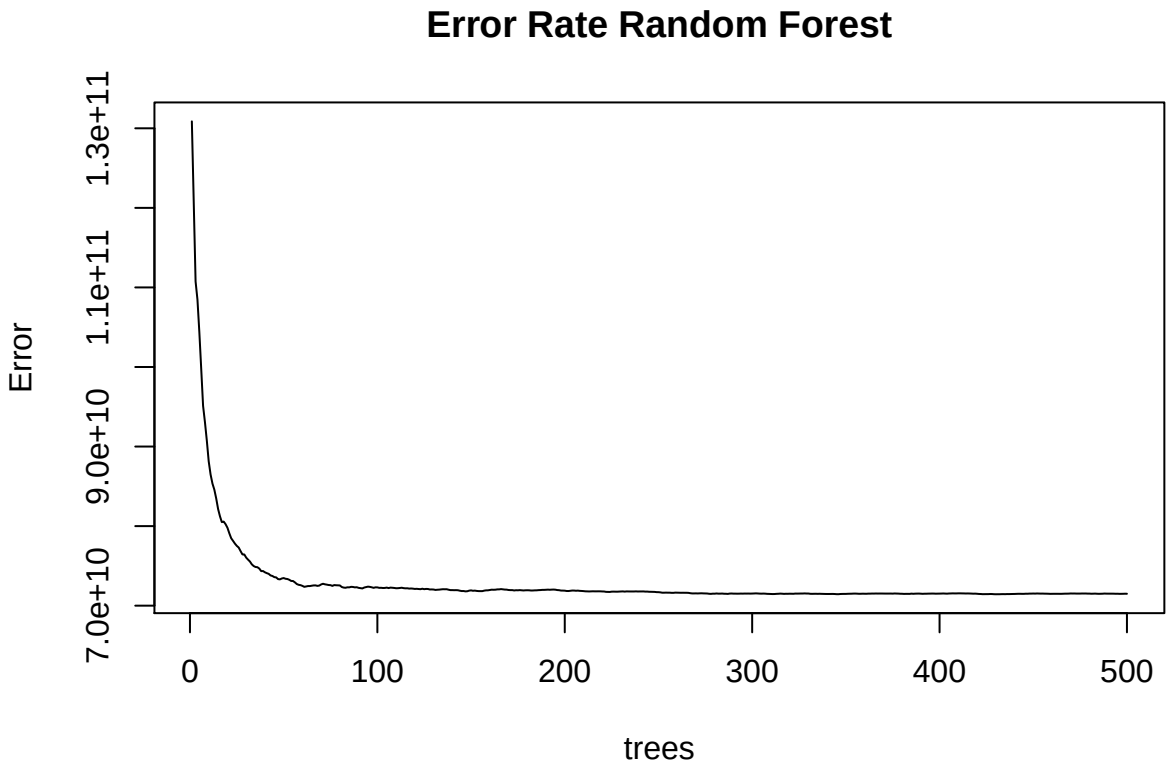

```
#Variable Importance
rn <- round(importance(model), 2)
rn[order(rn[,1], decreasing=TRUE),]
```

```
##               %IncMSE IncNodePurity
## Duration       136.61  2.443405e+14
## DischargeArea   70.68  6.302016e+13
## VesselType      47.09  6.952904e+13
## Distance        34.58  4.418900e+13
## LoadArea        33.43  3.199699e+13
## Dwt             33.33  4.856998e+13
## Qty             27.26  7.205658e+13
## Month           19.28  3.212207e+13
## Cargo           12.70  1.936047e+13
```

```
#Variale Importance Plot
varImpPlot(model, main="")
title(main="Variable Importance Random Forest")
```




```r
#Plot the error rate
plot(model, main="")
title(main="Error Rate Random Forest")
```



Now after seeing the results of Random Forest we decided to remove outliers and see if the results are improving.

So Following are the different models where we removed outliers one by one and checked the results.

Remove outliers and apply Random Forest

Model 1

```
#remove outliers - Duration

x1 <- quantile(Final$Duration,c(0.01,0.99))
Final <- Final[Final$Duration >=x1[1] & Final$Duration<=x1[2],]

#############################################################################################
#Divide into training & test datasets
set.seed(666)
split <- sample(nrow(Final), floor(0.7*nrow(Final)))
train <- Final[split,]
test <- Final[-split,]

#Random Forest model
model1 <- randomForest(Earnings ~ Distance+Duration+Month+LoadArea+DischargeArea+Cargo+Qty
                       +VesselType+Dwt,
                       data=train,
                       ntree=500,
                       mtry=4,
                       importance=TRUE,
                       na.action = na.roughfix,
                       replace=FALSE)

model1
```

```
##
## Call:
##  randomForest(formula = Earnings ~ Distance + Duration + Month +      LoadArea + DischargeArea + Carg
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 64943902783
##                     % Var explained: 53.88
```

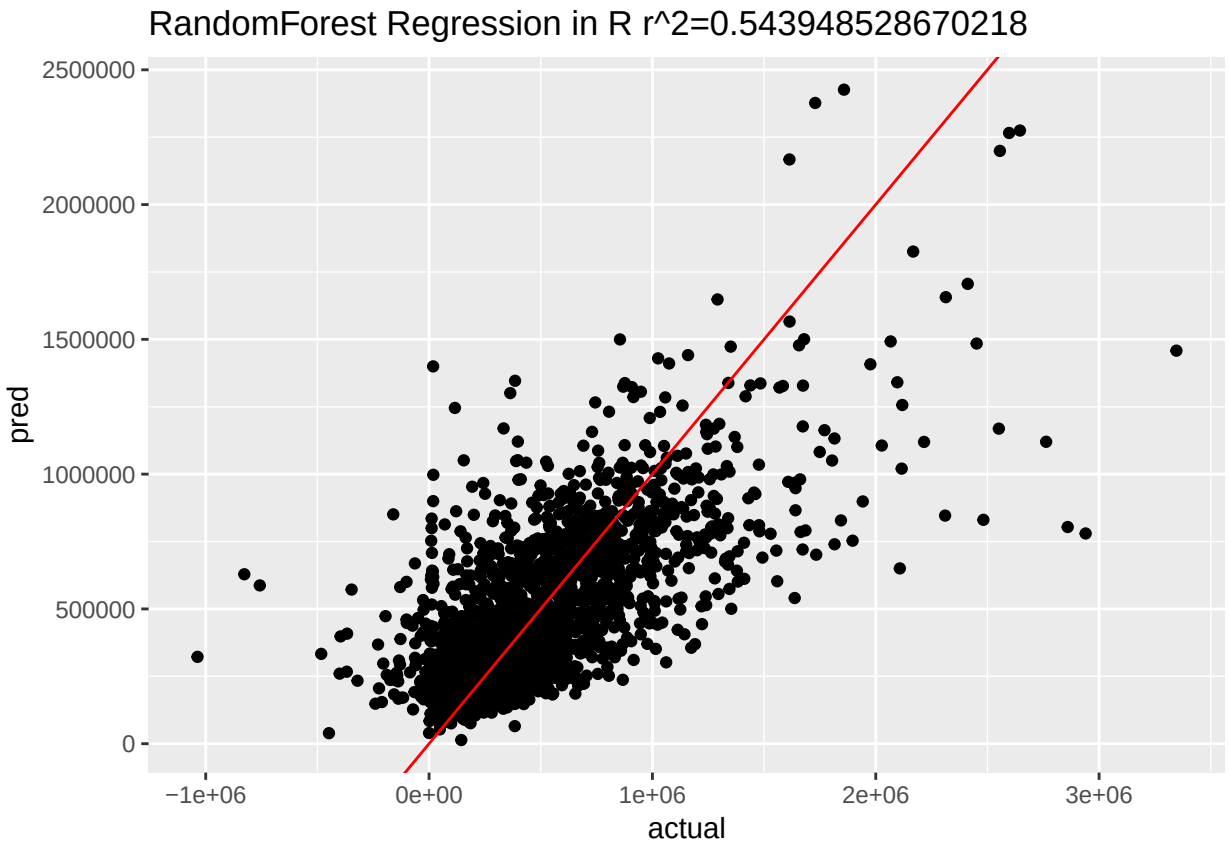
```
(r21 <- rSquared(test$Earnings, test$Earnings - predict(model1, test)))
```

```
##           [,1]
## [1,] 0.5439485
```

```
(mse1 <- mean((test$Earnings - predict(model1, test))^2))
```

```
## [1] 65536144326
```

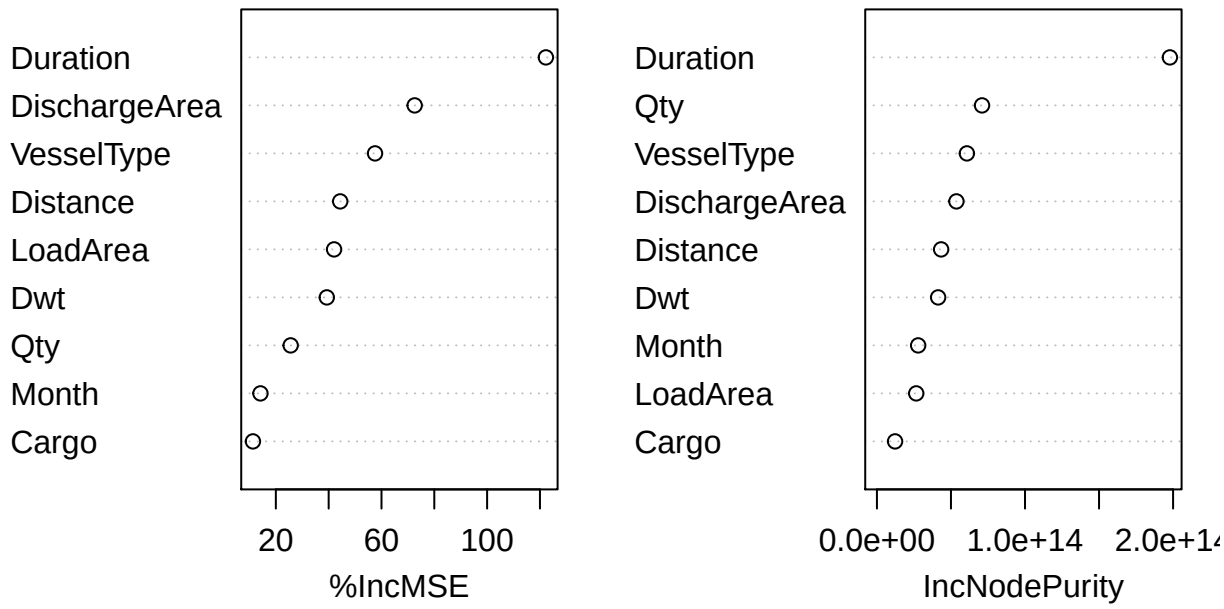
```
p <- ggplot(aes(x=actual, y=pred),
            data=data.frame(actual=test$Earnings, pred=predict(model1, test)))
p + geom_point() +
  geom_abline(color="red") +
  ggtitle(paste("RandomForest Regression in R r^2=", r21, sep=""))
```



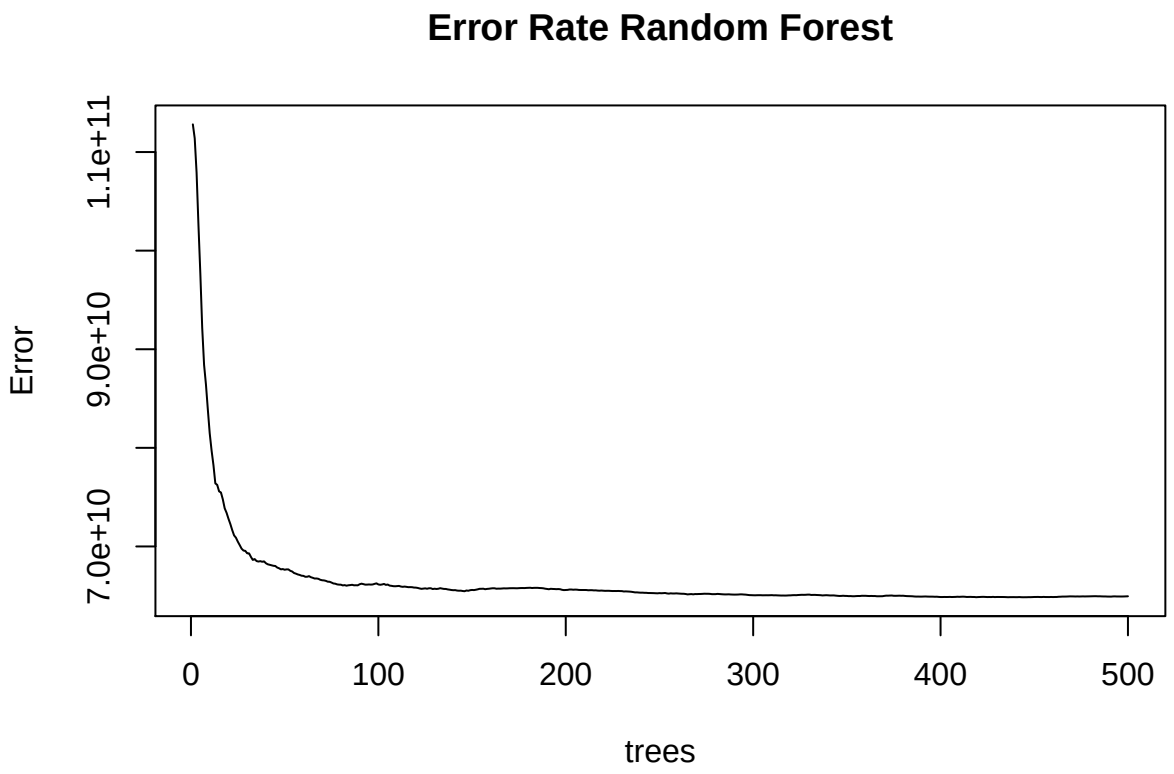

```
#Variable Importance
rn1 <- round(importance(model1), 2)
rn1[order(rn1[,1], decreasing=TRUE),]
```

```
##               %IncMSE IncNodePurity
## Duration       122.25  1.978349e+14
## DischargeArea   72.56  5.368892e+13
## VesselType      57.56  6.074917e+13
## Distance        44.38  4.336935e+13
## LoadArea        42.05  2.654277e+13
## Dwt             39.30  4.133581e+13
## Qty             25.61  7.096495e+13
## Month           14.18  2.784283e+13
## Cargo           11.33  1.224655e+13
```

```
#Variale Importance Plot
varImpPlot(model1, main="")
title(main="Variable Importance Random Forest")
```




```r
#Plot the error rate
plot(model1, main="")
title(main="Error Rate Random Forest")
```




```
#Test it on the test dataset
outcome1 <- predict(model1, test)
```

Model 2

```
#remove outliers - Duration and Distance
x2 <- quantile(Final$Distance,c(0.01,0.99))
Final <- Final[Final$Distance >=x2[1] & Final$Distance<=x2[2],]
#################################################################################################
#Divide into training & test datasets
set.seed(666)
split <- sample(nrow(Final), floor(0.7*nrow(Final)))
train <- Final[split,]
test <- Final[-split,]

#Random Forest model
model2 <- randomForest(Earnings ~ Distance+Duration+Month+LoadArea+DischargeArea+Cargo+Qty
                       +VesselType+Dwt,
                       data=train,
                       ntree=500,
                       mtry=4,
                       importance=TRUE,
                       na.action = na.roughfix,
                       replace=FALSE)

model2
```

```
##
## Call:
##  randomForest(formula = Earnings ~ Distance + Duration + Month +      LoadArea + DischargeArea + Carg
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 66393692477
##                     % Var explained: 55.07
```

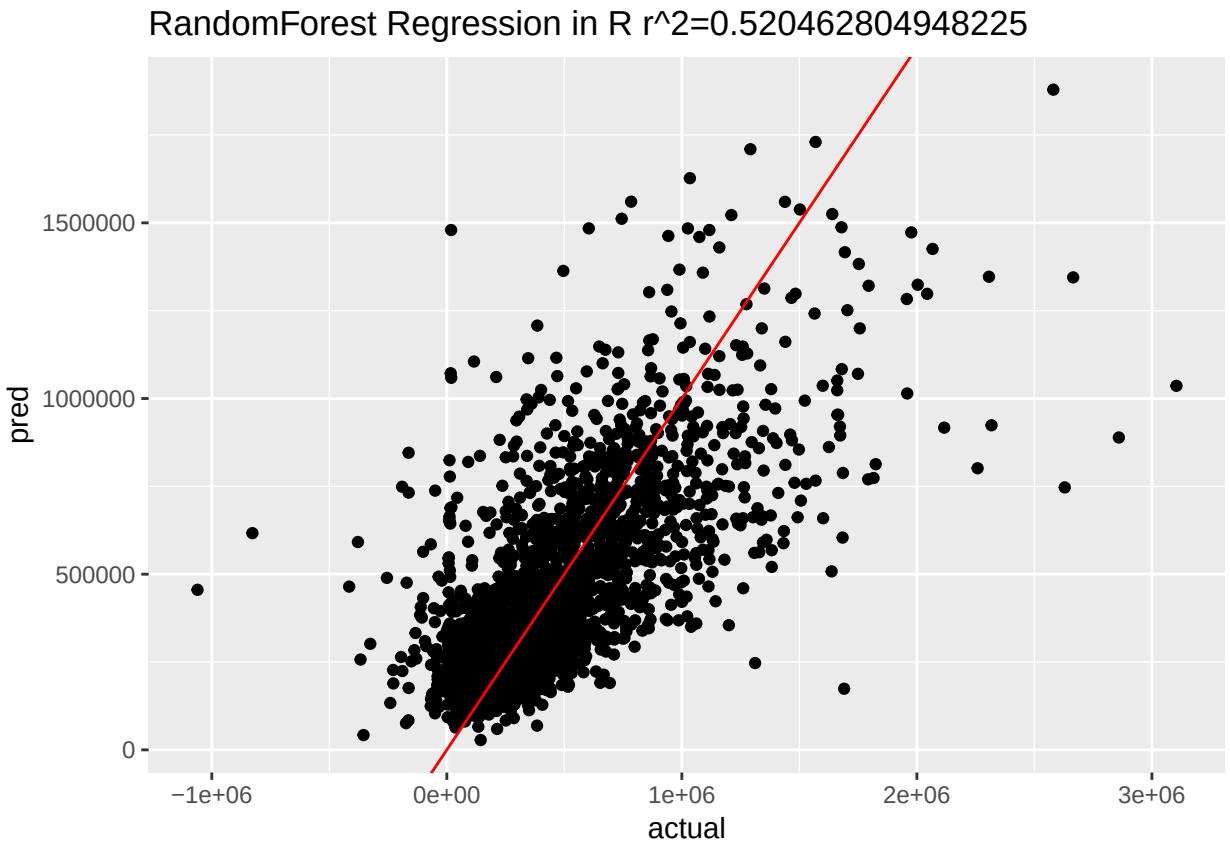
```
(r22 <- rSquared(test$Earnings, test$Earnings - predict(model2, test)))
```

```
##           [,1]
## [1,] 0.5204628
```

```
(mse2 <- mean((test$Earnings - predict(model2, test))^2))
```

```
## [1] 62250269995
```

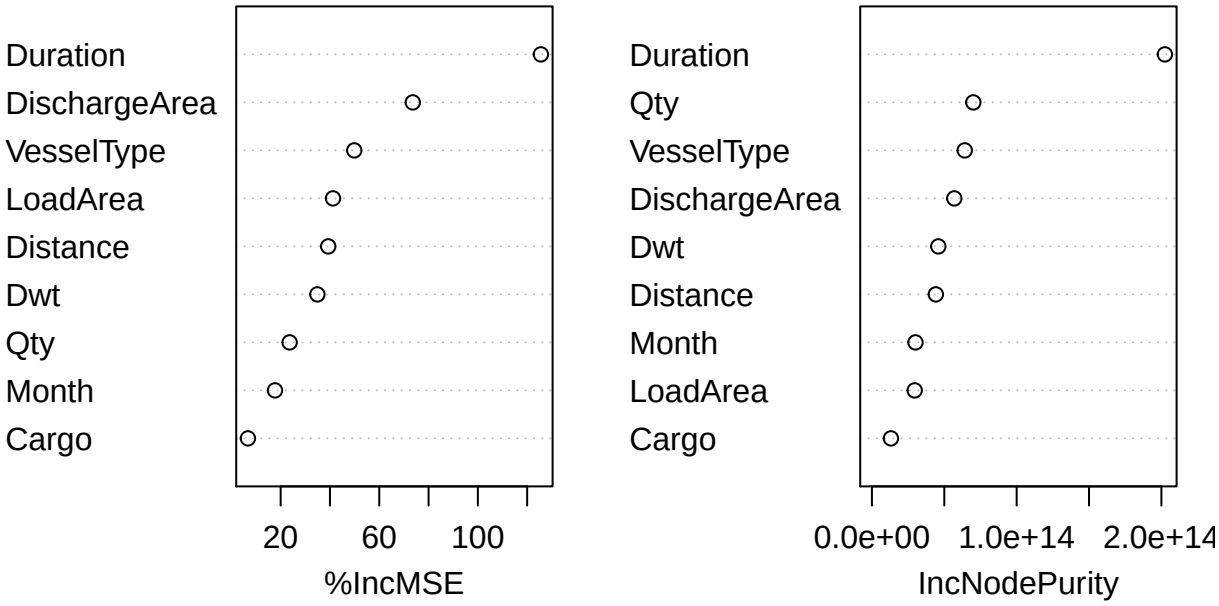
```
p <- ggplot(aes(x=actual, y=pred),
            data=data.frame(actual=test$Earnings, pred=predict(model2, test)))
p + geom_point() +
  geom_abline(color="red") +
  ggtitle(paste("RandomForest Regression in R r^2=", r22, sep=""))
```



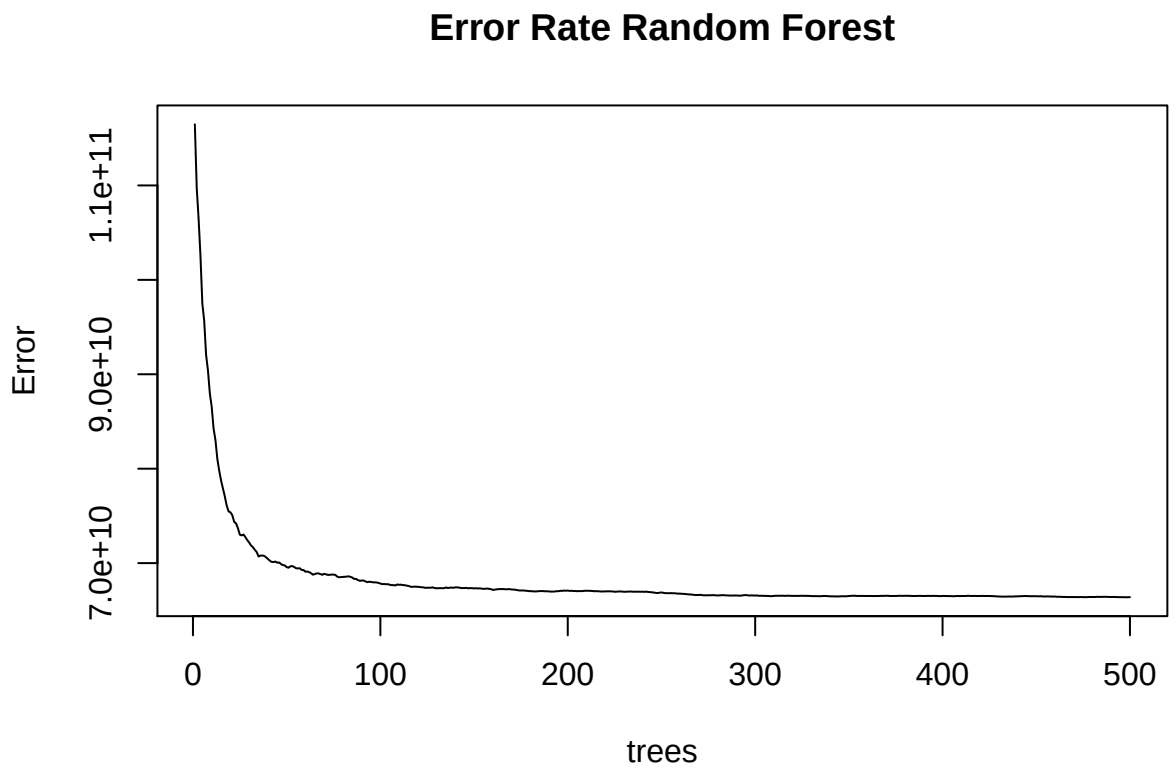

```
#Variable Importance
rn2 <- round(importance(model2), 2)
rn2[order(rn2[,1], decreasing=TRUE),]
```

```
##               %IncMSE IncNodePurity
## Duration       125.56  2.024113e+14
## DischargeArea   73.60  5.691918e+13
## VesselType      49.86  6.412334e+13
## LoadArea        41.23  2.952749e+13
## Distance        39.29  4.417215e+13
## Dwt             34.90  4.587973e+13
## Qty             23.65  7.005938e+13
## Month           17.70  3.008974e+13
## Cargo            6.73  1.311778e+13
```

```
#Variale Importance Plot
varImpPlot(model2, main="")
title(main="Variable Importance Random Forest")
```




```
#Plot the error rate
plot(model2, main="")
title(main="Error Rate Random Forest")
```




```
#Test it on the test dataset
outcome2 <- predict(model2, test)
```

Model 3

```
#remove outliers - Duration, Distance and Quantity
x3 <- quantile(Final$Qty,c(0.01,0.99))
Final <- Final[Final$Qty >=x3[1] & Final$Qty<=x3[2],]
#############################################################################################
#Divide into training & test datasets
set.seed(666)
split <- sample(nrow(Final), floor(0.7*nrow(Final)))
train <- Final[split,]
test <- Final[-split,]

#Random Forest model
model3 <- randomForest(Earnings ~ Distance+Duration+Month+LoadArea+DischargeArea+Cargo+Qty
                       +VesselType+Dwt,
                       data=train,
                       ntree=500,
                       mtry=4,
                       importance=TRUE,
                       na.action = na.roughfix,
                       replace=FALSE)

model3
```

```
##
## Call:
##  randomForest(formula = Earnings ~ Distance + Duration + Month +      LoadArea + DischargeArea + Carg
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 63192754504
##                     % Var explained: 51.79
```

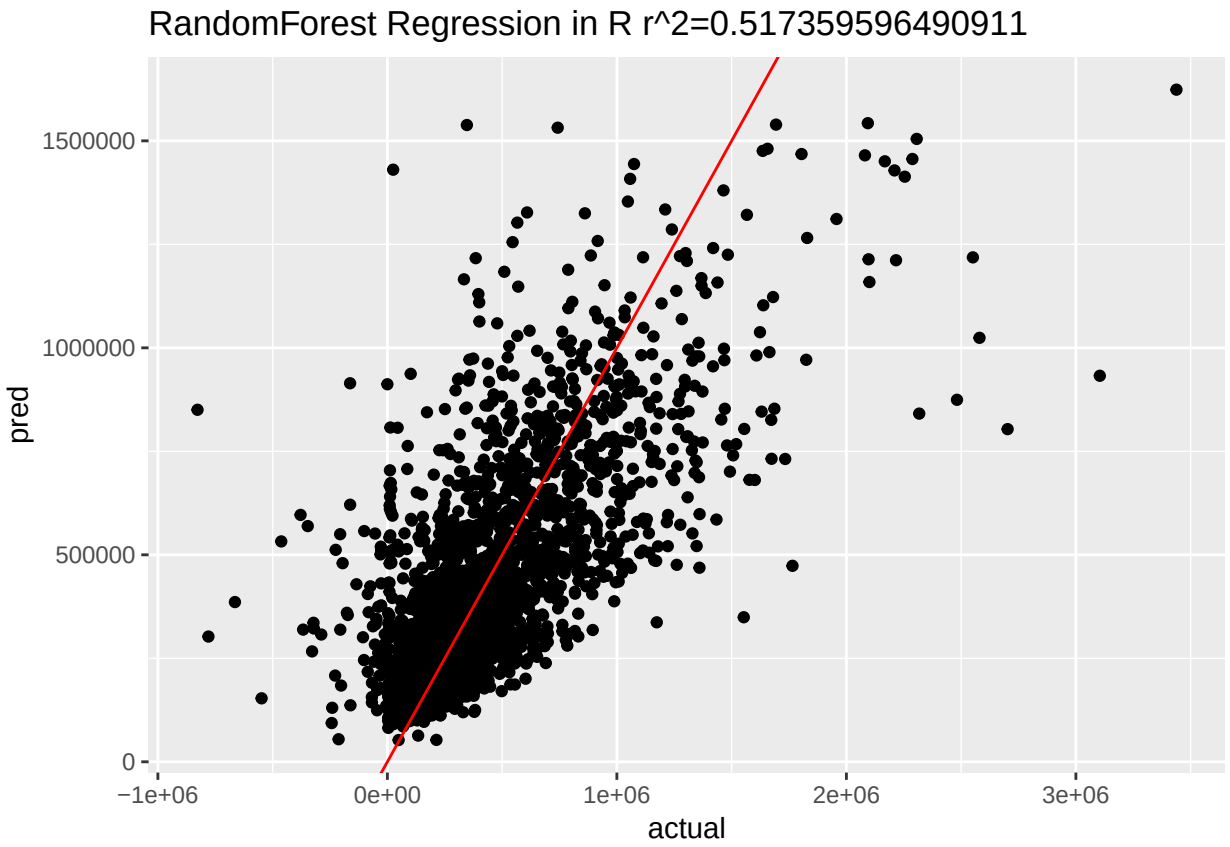
```
(r23 <- rSquared(test$Earnings, test$Earnings - predict(model3, test)))
```

```
##           [,1]
## [1,] 0.5173596
```

```
(mse3 <- mean((test$Earnings - predict(model3, test))^2))
```

```
## [1] 63743712376
```

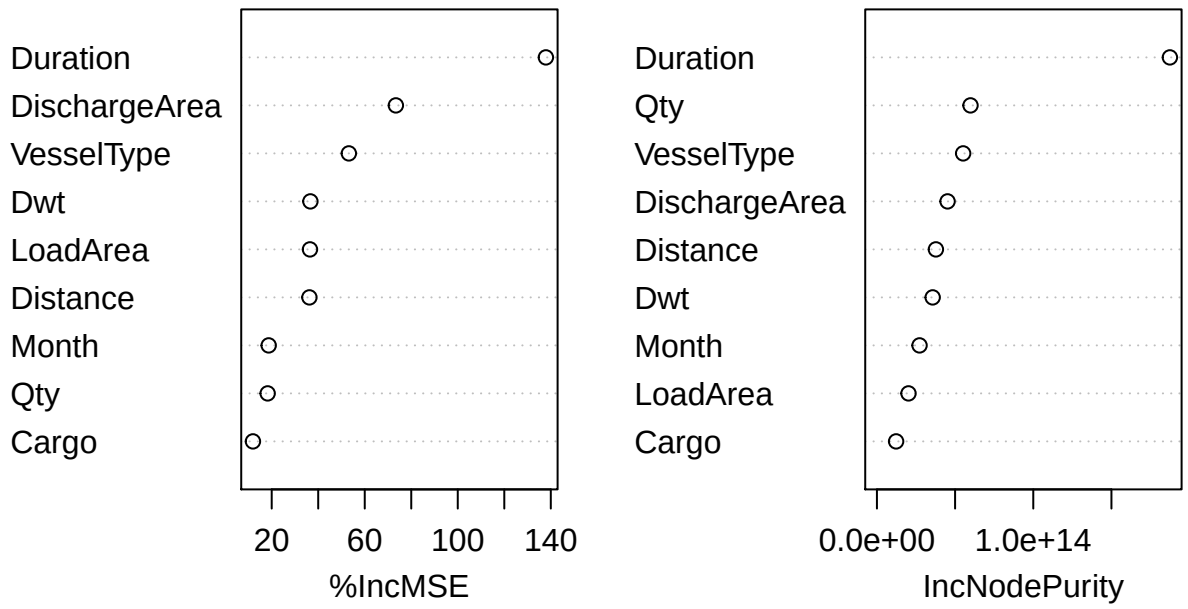
```
p <- ggplot(aes(x=actual, y=pred),
            data=data.frame(actual=test$Earnings, pred=predict(model3, test)))
p + geom_point() +
  geom_abline(color="red") +
  ggtitle(paste("RandomForest Regression in R r^2=", r23, sep=""))
```



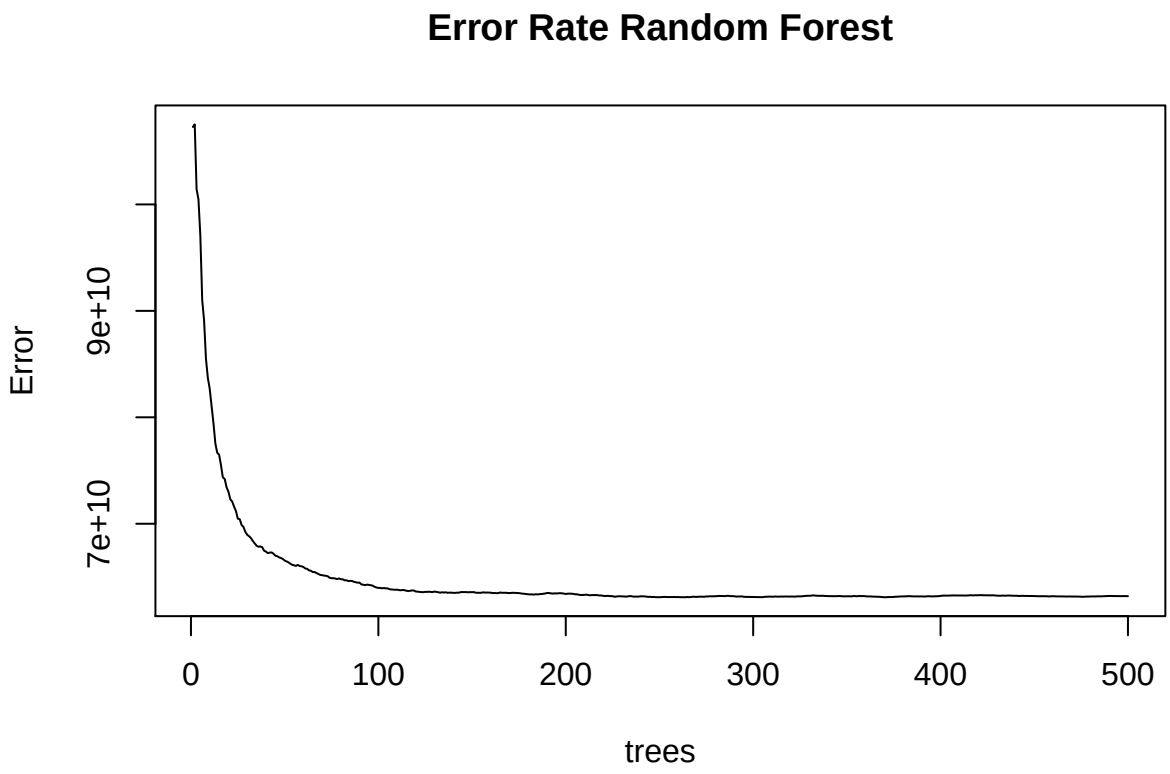

```
#Variable Importance
rn3 <- round(importance(model3), 2)
rn3[order(rn3[,1], decreasing=TRUE),]
```

```
##               %IncMSE IncNodePurity
## Duration       137.89  1.873848e+14
## DischargeArea   73.38  4.517815e+13
## VesselType      53.17  5.515326e+13
## Dwt             36.63  3.564726e+13
## LoadArea        36.45  2.019504e+13
## Distance        36.22  3.771585e+13
## Month           18.71  2.726255e+13
## Qty             18.26  5.985302e+13
## Cargo           11.93  1.226655e+13
```

```
#Variale Importance Plot
varImpPlot(model3, main="")
title(main="Variable Importance Random Forest")
```




```
#Plot the error rate
plot(model3, main="")
title(main="Error Rate Random Forest")
```




```
#Test it on the test dataset
outcome3 <- predict(model3, test)
```

Model 4

```
#remove outliers - Duration, Distance, Quantity and Dwt
x4 <- quantile(Final$Dwt,c(0.01,0.99))
Final <- Final[Final$Dwt >=x4[1] & Final$Dwt<=x4[2],]

#############################################################################################
#Divide into training & test datasets
set.seed(666)
split <- sample(nrow(Final), floor(0.7*nrow(Final)))
train <- Final[split,]
test <- Final[-split,]

#Random Forest model
model4 <- randomForest(Earnings ~ Distance+Duration+Month+LoadArea+DischargeArea+Cargo+Qty
                       +VesselType+Dwt,
                       data=train,
                       ntree=500,
                       mtry=4,
                       importance=TRUE,
                       na.action = na.roughfix,
                       replace=FALSE)

model4
```

```
##
## Call:
##  randomForest(formula = Earnings ~ Distance + Duration + Month +      LoadArea + DischargeArea + Carg
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 62710020224
##                     % Var explained: 53.48
```

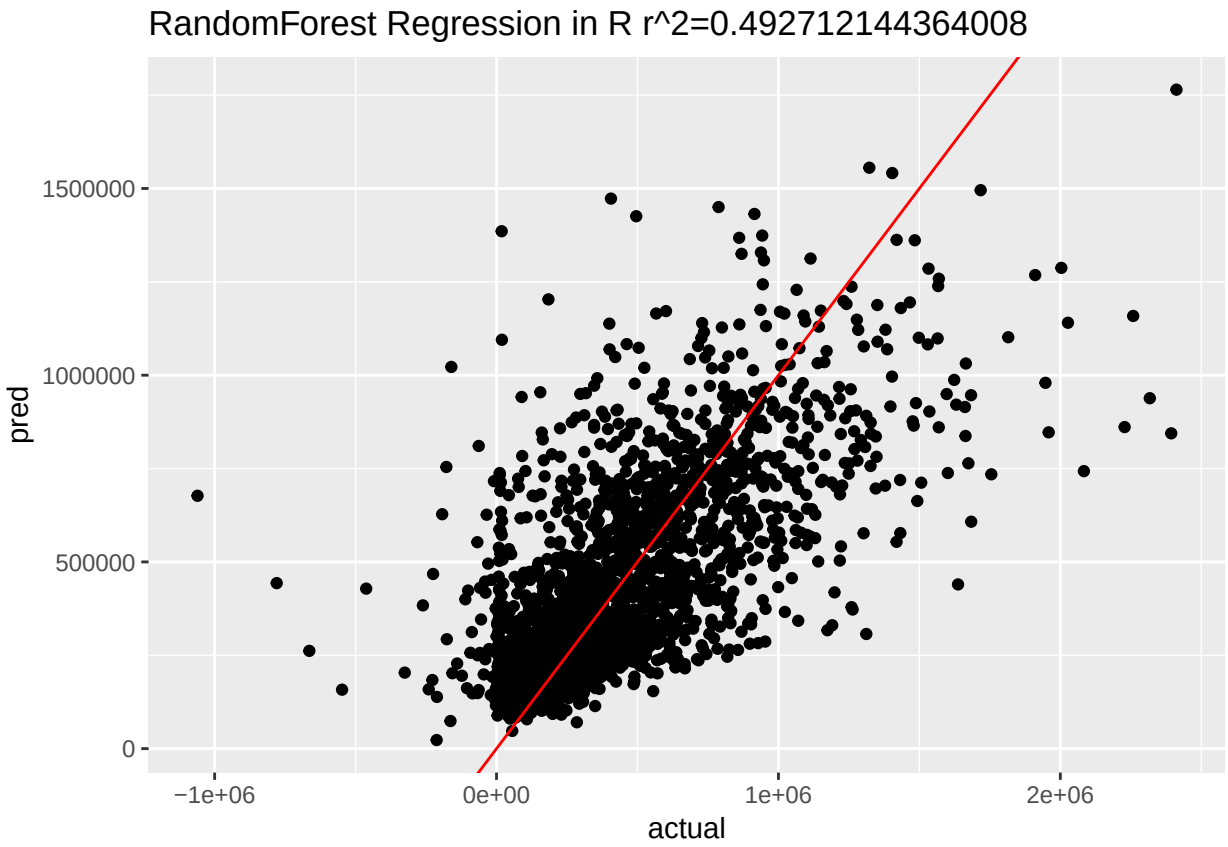
```
(r24 <- rSquared(test$Earnings, test$Earnings - predict(model4, test)))
```

```
##           [,1]
## [1,] 0.4927121
```

```
(mse4 <- mean((test$Earnings - predict(model4, test))^2))
```

```
## [1] 59026144548
```

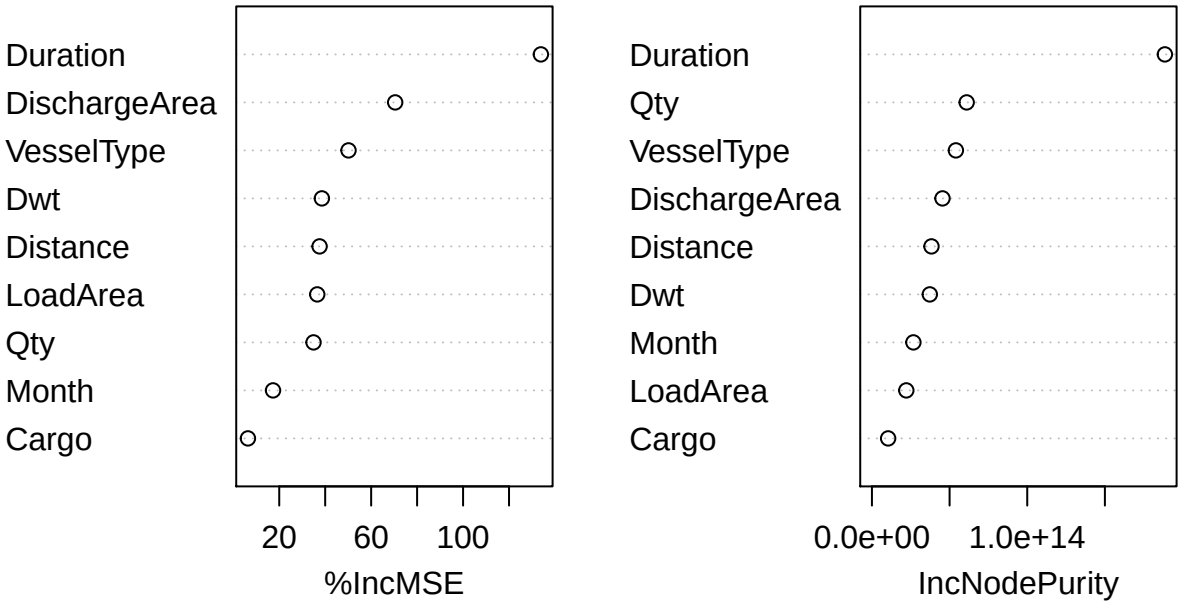
```
p <- ggplot(aes(x=actual, y=pred),
            data=data.frame(actual=test$Earnings, pred=predict(model4, test)))
p + geom_point() +
  geom_abline(color="red") +
  ggtitle(paste("RandomForest Regression in R r^2=", r24, sep=""))
```



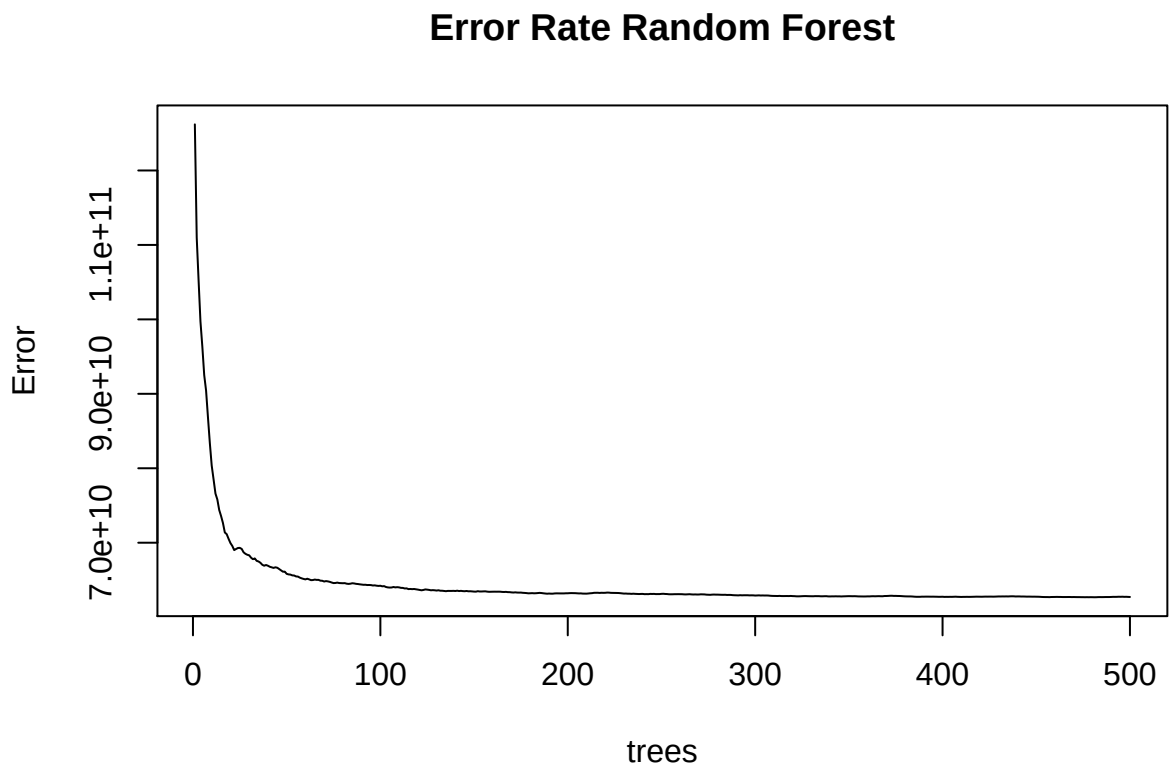

```
#Variable Importance
rn4 <- round(importance(model4), 2)
rn4[order(rn4[,1], decreasing=TRUE),]
```

```
##               %IncMSE IncNodePurity
## Duration       133.86  1.886295e+14
## DischargeArea   70.47  4.544126e+13
## VesselType      50.15  5.406034e+13
## Dwt             38.56  3.725173e+13
## Distance        37.56  3.834057e+13
## LoadArea        36.54  2.213149e+13
## Qty             34.92  6.099887e+13
## Month           17.30  2.669699e+13
## Cargo            6.38  1.043990e+13
```

```
#Variale Importance Plot
varImpPlot(model4, main="")
title(main="Variable Importance Random Forest")
```




```
#Plot the error rate
plot(model4, main="")
title(main="Error Rate Random Forest")
```




```
#Test it on the test dataset
outcome4 <- predict(model4, test)
```

Model 5

```
#remove outliers - Duration, Distance, Quantity, Dwt and Earnings
Final$Earnings <- as.numeric(Final$Earnings)
x5 <- quantile(Final$Earnings,c(0.01,0.99))
Final <- Final[Final$Earnings >=x5[1] & Final$Earnings<=x5[2],]

#################################################################################################
#Divide into training & test datasets
set.seed(666)
split <- sample(nrow(Final), floor(0.7*nrow(Final)))
train <- Final[split,]
test <- Final[-split,]

#Random Forest model
model5 <- randomForest(Earnings ~ Distance+Duration+Month+LoadArea+DischargeArea+Cargo+Qty
                       +VesselType+Dwt,
                       data=train,
                       ntree=500,
                       mtry=4,
                       importance=TRUE,
                       na.action = na.roughfix,
                       replace=FALSE)

model5
```

```
##
## Call:
##  randomForest(formula = Earnings ~ Distance + Duration + Month +      LoadArea + DischargeArea + Carg
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 46376475198
##                     % Var explained: 52.71
```

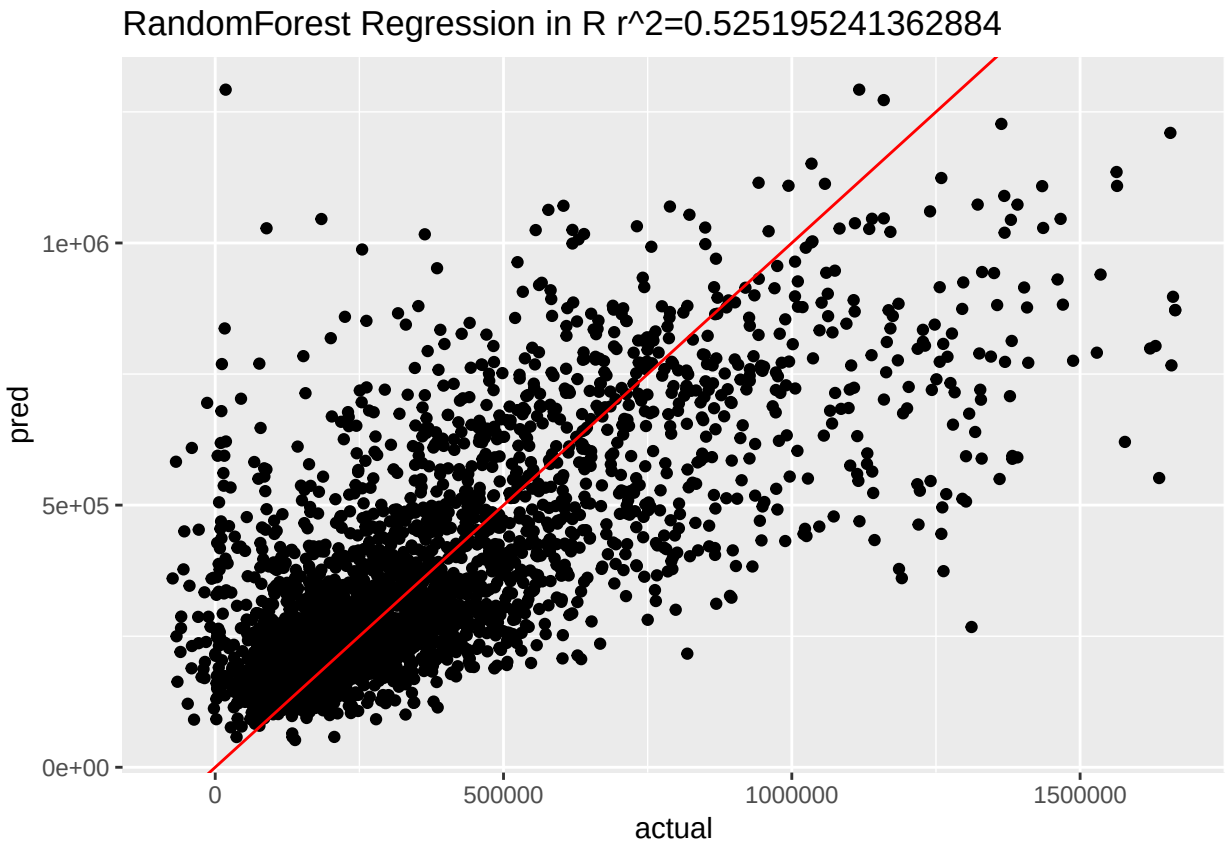
```
(r25 <- rSquared(test$Earnings, test$Earnings - predict(model5, test)))
```

```
##           [,1]
## [1,] 0.5251952
```

```
(mse5 <- mean((test$Earnings - predict(model5, test))^2))
```

```
## [1] 44424881277
```

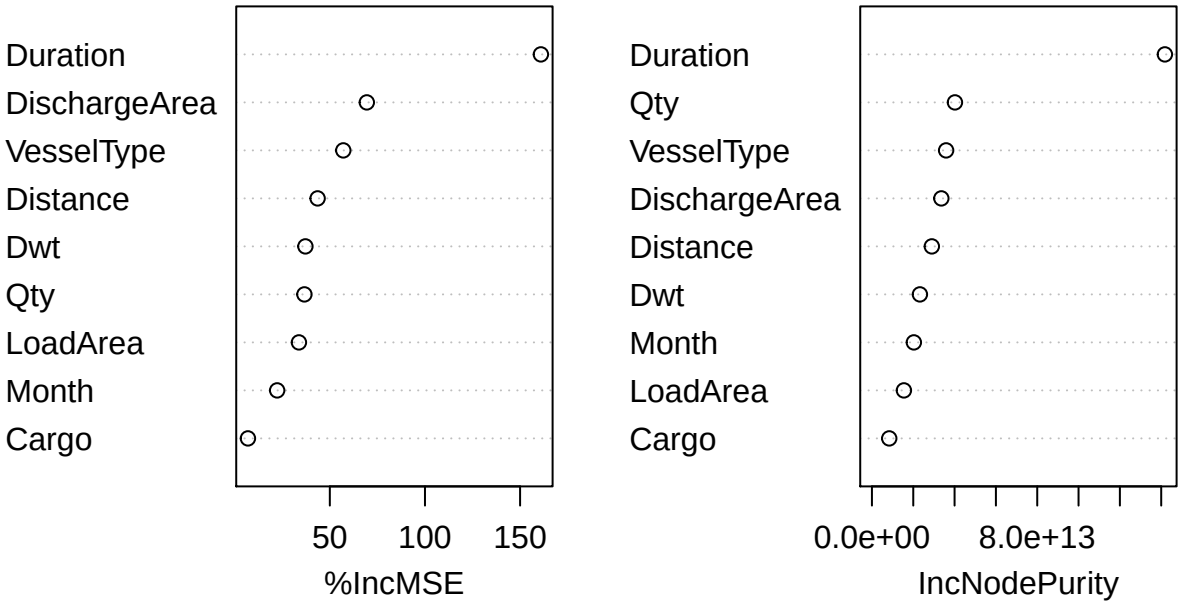
```
p <- ggplot(aes(x=actual, y=pred),
            data=data.frame(actual=test$Earnings, pred=predict(model5, test)))
p + geom_point() +
  geom_abline(color="red") +
  ggtitle(paste("RandomForest Regression in R r^2=", r25, sep=""))
```



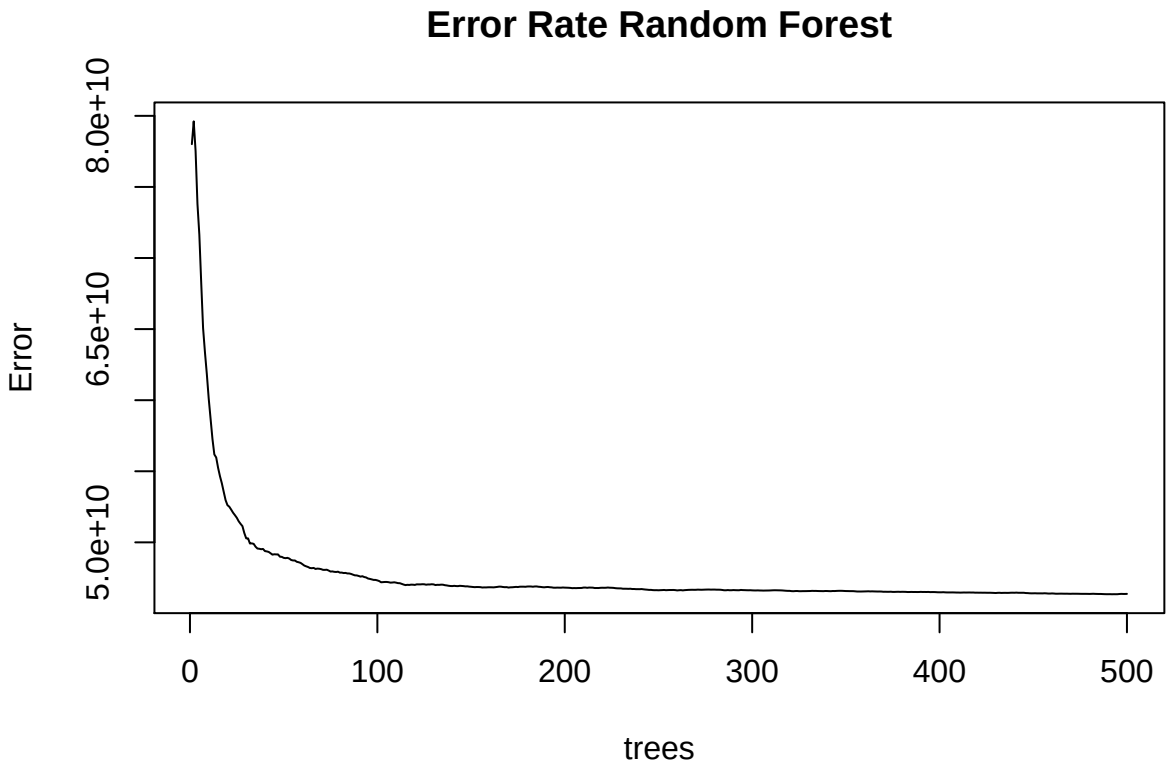

```
#Variable Importance
rn5 <- round(importance(model5), 2)
rn5[order(rn5[,1], decreasing=TRUE),]
```

```
##               %IncMSE IncNodePurity
## Duration       160.92  1.417815e+14
## DischargeArea   69.43  3.359447e+13
## VesselType      57.08  3.590998e+13
## Distance        43.58  2.898442e+13
## Dwt             37.17  2.322205e+13
## Qty             36.65  4.017476e+13
## LoadArea        33.76  1.547067e+13
## Month           22.33  2.028619e+13
## Cargo            6.96  8.374893e+12
```

```
#Variale Importance Plot
varImpPlot(model5, main="")
title(main="Variable Importance Random Forest")
```




```
#Plot the error rate
plot(model5, main="")
title(main="Error Rate Random Forest")
```




```
#Test it on the test dataset
outcome5 <- predict(model5, test)
```

The final model shows improved results by removing outliers from all the variables.

Conclusions:

1. Our Random Forest model shows improved results over Linear Regression model.

2. From our models we can say that Earnings can be predicted using vessel type, loading and discharge ports, cargo type, deadweight tonnage, cargo lift etc. independent variables.

3. While we have been able to improve the results, this may not be enough for the predictions as Weather related information is not provided.

4. If the information about intermittent stops for loading and discharge for a voyage is provided then it will be helpful to get more accurate predictions.

Recommendations:

1. Including weather data for each voyage for further analysis will be helpful to get more precise predictions as it is directly related to fuel consumptions.

2. Information about intermittent stops for loading and discharging will help in building more accurate model.

3. To improve the accuracy in the prediction we can try using different algorithms like Extreme gradient boosting.

4. Comparison of competitor's business can provide more ideas to improve the business.