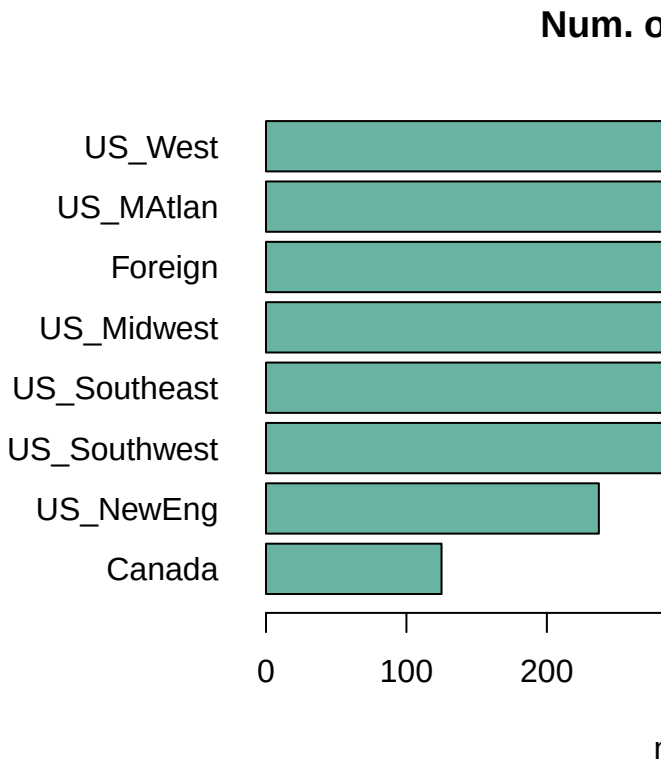
New_final_project

Haokun Zhang, Zhang Lu, Jonathan

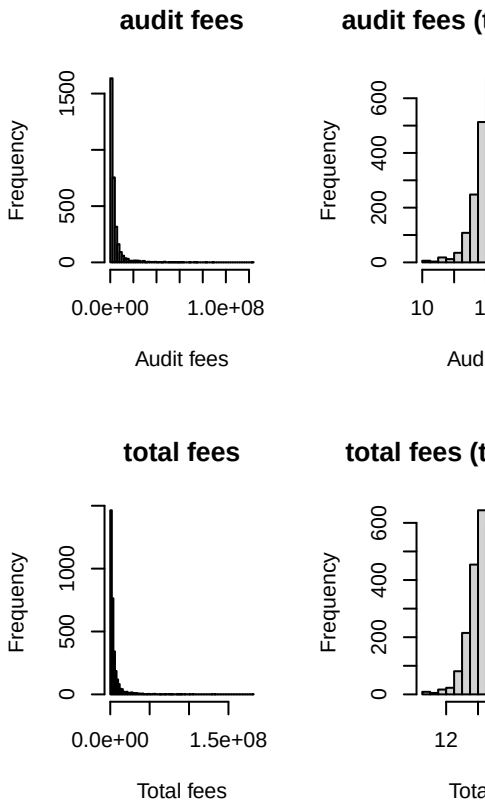
2023/04/21

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

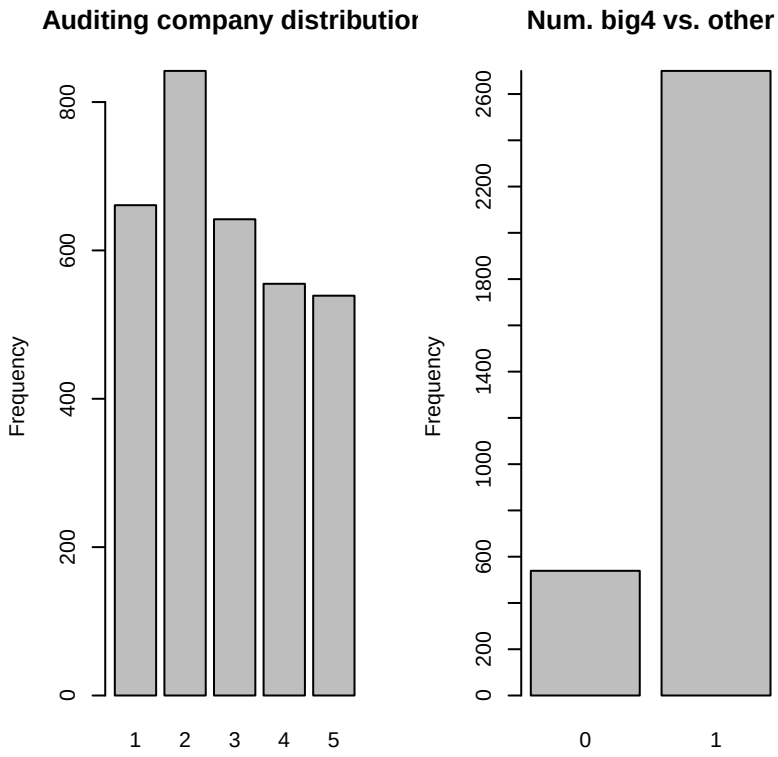
Data visualization(EDA)



Plot 1: plot the number distribution of companies in different regions

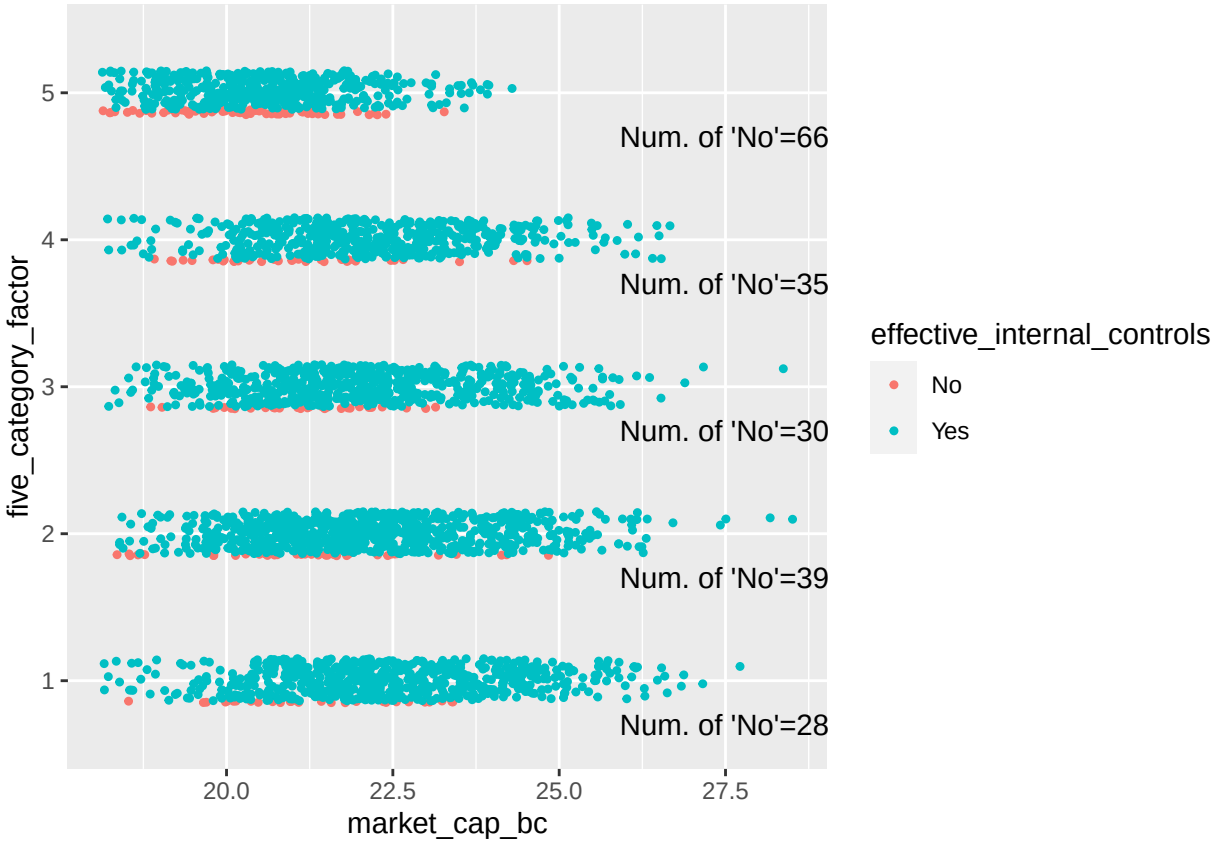


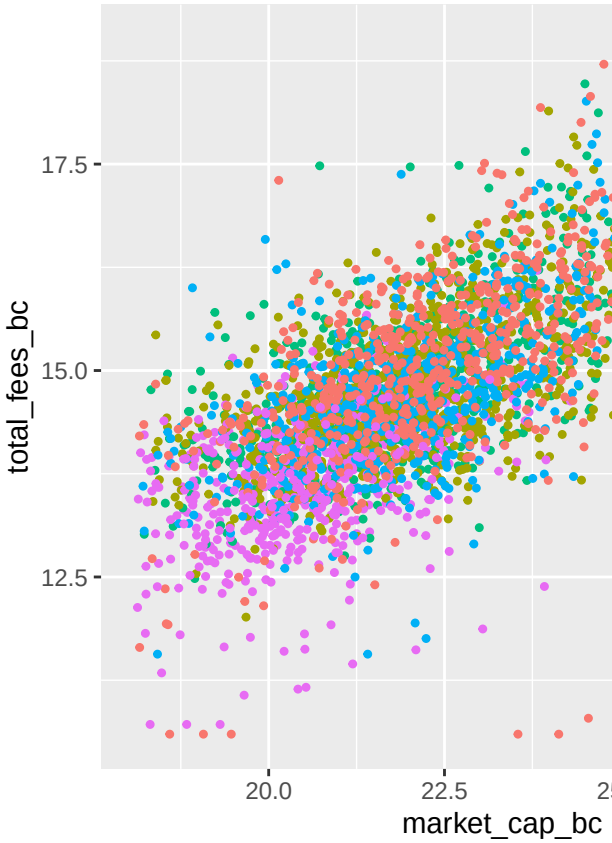
Plot 2: Use eight plots to display the effect of transformation on fee related variables



Plot 3: Use three plots to display the categorical data

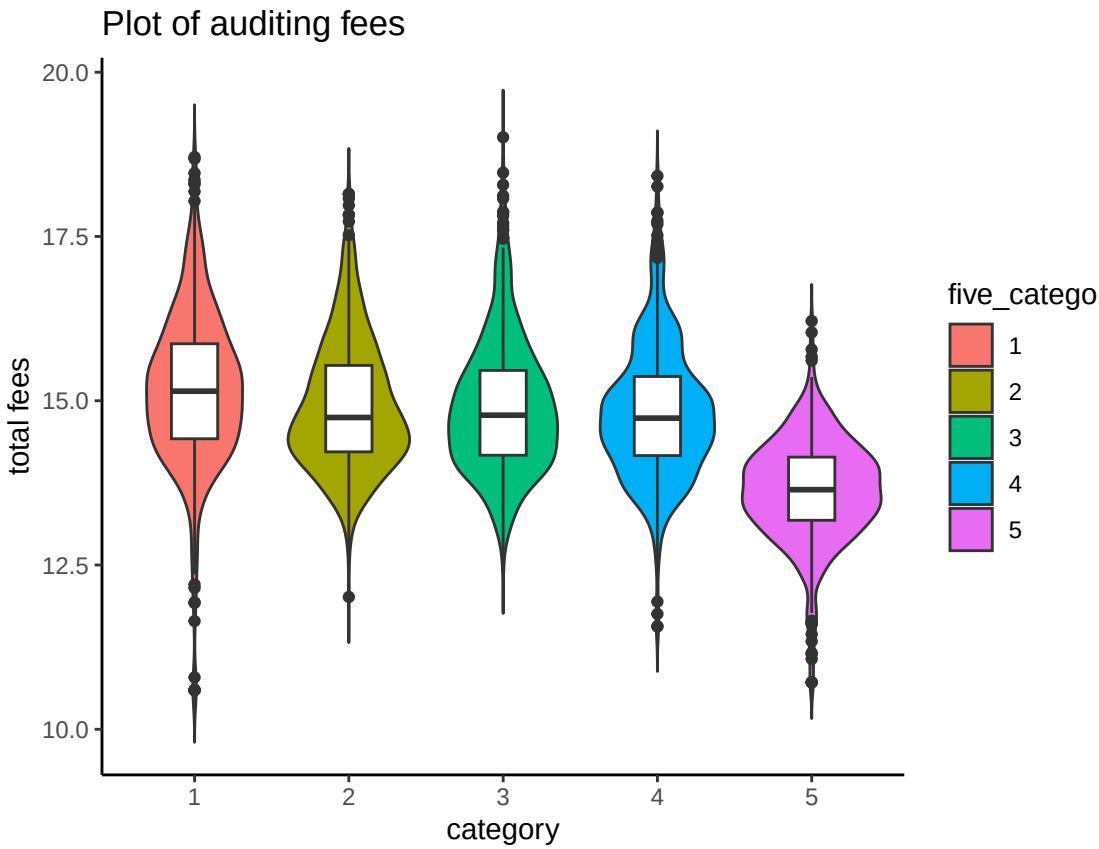
Plot 4. Plot the transformed company market cap, total auditing fees, and effective internal control


five_category_factor
5
4
3
2
1
Num. of 'No'=66
Num. of 'No'=35
Num. of 'No'=30
Num. of 'No'=39
Num. of 'No'=28
20.0
22.5
25.0
27.5
market_cap_bc
effective_internal_controls
No
Yes



Plot 5: Plot the transformed company market cap vs. total auditing fees

```
## [1] 0.6904368
```



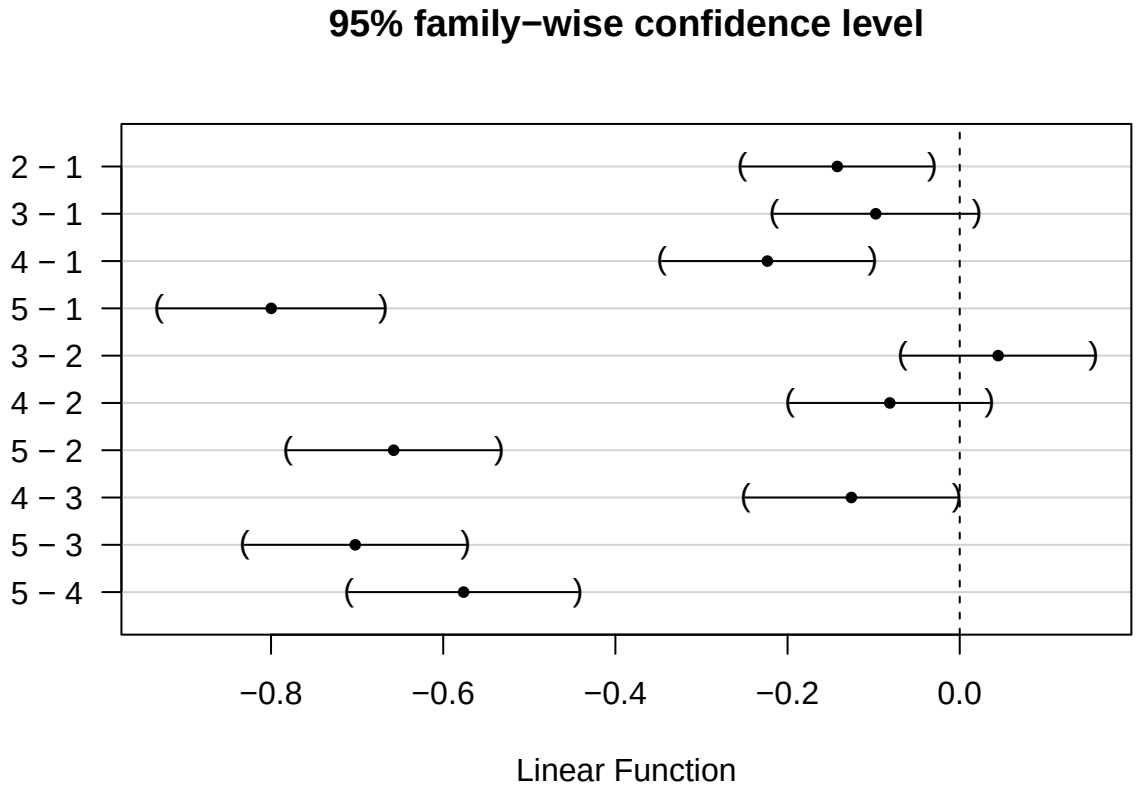
Plot 6: Plot the auditing fees

Perform a statistical test here to compare big4 vs non big4 when considering

```
## Anova Table (Type III tests)
##
## Response: total_fees_bc
##                                  Sum Sq   Df  F value    Pr(>F)
## (Intercept)                      101.36    1 168.8601 < 2.2e-16 ***
## five_category_factor              12.93    4   5.3853 0.0002547 ***
## market_cap_bc                    406.51    1 677.1899 < 2.2e-16 ***
## five_category_factor:market_cap_bc  20.26    4   8.4396 9.028e-07 ***
## Residuals                       1938.31 3229
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = total_fees_bc ~ five_category_factor + market_cap_bc,
##     data = df3)
##
```

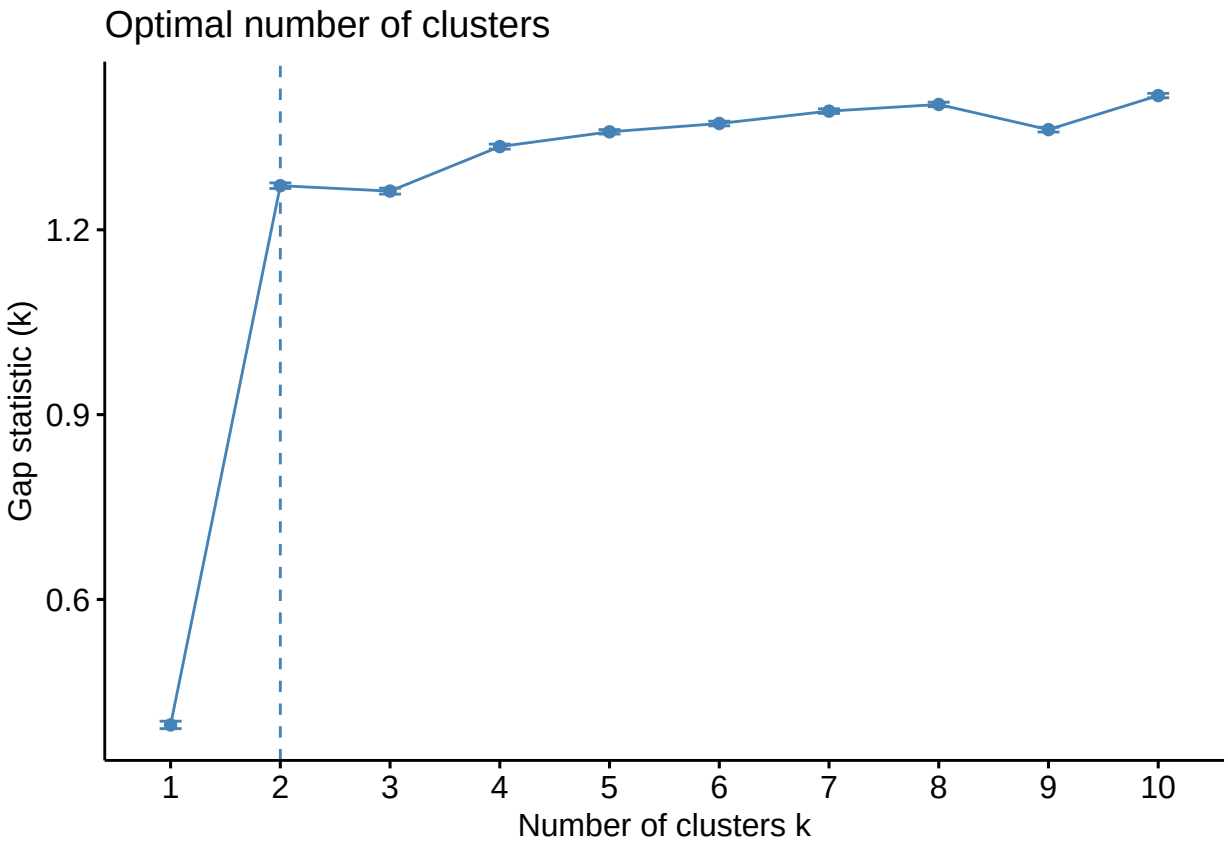
```
## Linear Hypotheses:
##            Estimate Std. Error t value Pr(>|t|)
## 2 - 1 == 0 -0.14223    0.04051  -3.511  0.00407 **
## 3 - 1 == 0 -0.09759    0.04327  -2.255  0.15902
## 4 - 1 == 0 -0.22345    0.04490  -4.977  < 1e-04 ***
## 5 - 1 == 0 -0.79979    0.04787 -16.707  < 1e-04 ***
## 3 - 2 == 0  0.04465    0.04080   1.094  0.80866
## 4 - 2 == 0 -0.08122    0.04256  -1.908  0.31152
## 5 - 2 == 0 -0.65756    0.04504 -14.599  < 1e-04 ***
## 4 - 3 == 0 -0.12586    0.04512  -2.790  0.04197 *
## 5 - 3 == 0 -0.70220    0.04712 -14.904  < 1e-04 ***
## 5 - 4 == 0 -0.57634    0.04885 -11.798  < 1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```



Find the optimal number of clustering

```
## Clustering Gap statistic ["clusGap"] from call:
## clusGap(x = pca_data, FUNcluster = kmeans, K.max = 10, B = 50,     nstart = 25)
## B=50 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
##  --> Number of clusters (method 'firstmax'): 2
##          logW    E.logW       gap      SE.sim
##  [1,] 9.644472 10.040814 0.3963421 0.006042067
##  [2,] 8.538367  9.809951 1.2715833 0.004570689
##  [3,] 8.392705  9.655512 1.2628073 0.004856802
```

```
##  [4,] 8.186603  9.521637 1.3350341 0.004076445
##  [5,] 8.103228  9.462249 1.3590217 0.003586380
##  [6,] 8.036768  9.409289 1.3725207 0.003795454
##  [7,] 7.975004  9.367624 1.3926195 0.003741049
##  [8,] 7.926054  9.329508 1.4034540 0.003674083
##  [9,] 7.935022  9.297599 1.3625769 0.003852537
## [10,] 7.856779  9.274600 1.4178218 0.003624395
```

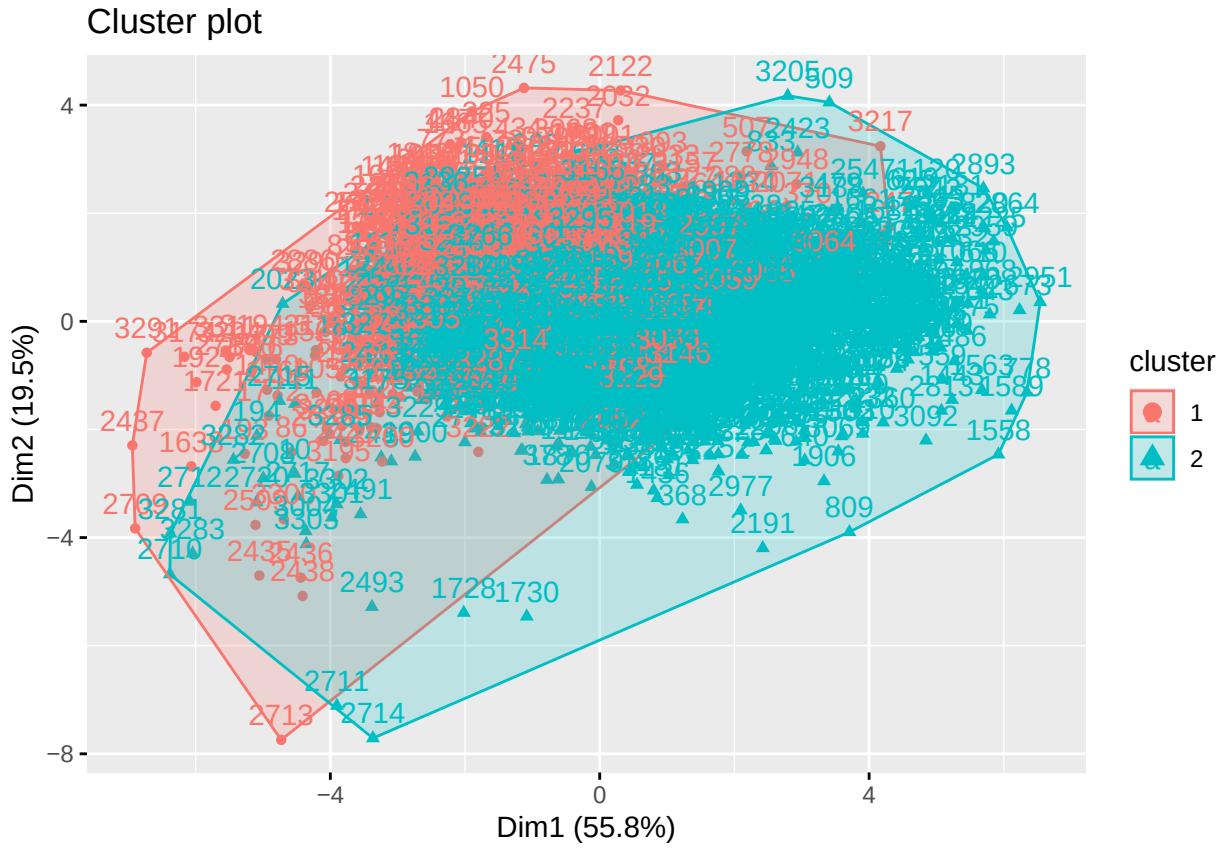


The optimal number of clusters is 2

```
##   cluster audit_fees_bc total_fees_bc market_cap_bc market_fee_ratio assets_log
## 1       1      14.32741      14.46416      21.14987         6.685713   21.03010
## 2       2      14.68505      14.84692      22.20437         7.357452   22.40376
##   revenue_trans earnings_trans
## 1      17.70979      -18.25516
## 2      21.19880       19.27055
```

```
##   audit_fees_bc total_fees_bc market_cap_bc market_fee_ratio assets_log
## 1      14.01025      14.26595      22.26620         8.000254   21.90575
## 2      13.31298      13.31298      18.78857         5.475583   19.59625
## 3      13.70766      13.70766      20.98602         7.278362   19.78954
## 4      13.98102      13.98568      19.59019         5.604515   19.08812
## 5      11.46850      11.60027      20.21906         8.618788   20.03306
## 6      13.88246      13.94147      22.14374         8.202271   21.59282
##   revenue_trans earnings_trans cluster
## 1      20.91361      -18.43784       1
```

```
## 2       19.42208       16.02850       2
## 3       19.33714       18.36820       2
## 4       19.67223      -17.38389       1
## 5       19.62253       16.85774       2
## 6       20.61072       18.13880       2
```



The above: How to explain it

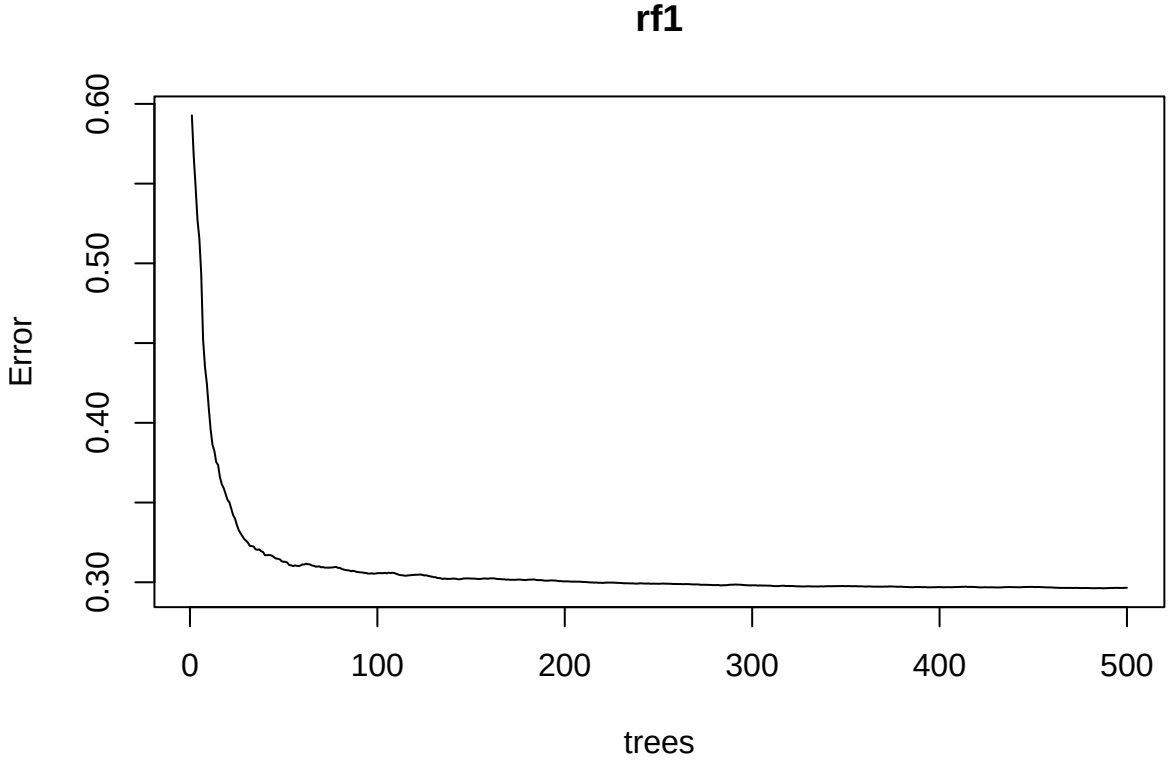
```
##
## Call:
## summary.resamples(object = res)
##
## Models: lm, knn, rf, cart, gbm
## Number of resamples: 5
##
## MAE
##           Min.    1st Qu.     Median       Mean    3rd Qu.       Max. NA's
## lm   0.5008859 0.5039455 0.5103137 0.5104453 0.5112804 0.5258009    0
## knn  0.4204386 0.4625901 0.4651704 0.4579766 0.4680718 0.4736120    0
## rf   0.4143817 0.4168010 0.4310460 0.4298915 0.4402179 0.4470108    0
## cart 0.4920306 0.5171260 0.5248796 0.5210956 0.5344083 0.5370334    0
## gbm  0.4119789 0.4233534 0.4325425 0.4311753 0.4339765 0.4540254    0
##
## RMSE
##           Min.    1st Qu.     Median       Mean    3rd Qu.       Max. NA's
## lm   0.6225688 0.6350685 0.6363976 0.6391081 0.6485709 0.6529345    0
## knn  0.5469420 0.5935824 0.5992334 0.5924718 0.6025545 0.6200466    0
```

```
## rf   0.5319624 0.5396446 0.5548445 0.5536457 0.5694439 0.5723329    0
## cart 0.6348013 0.6702136 0.6805042 0.6704935 0.6824631 0.6844852    0
## gbm  0.5415868 0.5419930 0.5505434 0.5560186 0.5630800 0.5828899    0
##
## Rsquared
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm   0.6588095 0.6663138 0.6782996 0.6789494 0.6849828 0.7063414    0
## knn  0.7085653 0.7133757 0.7235648 0.7261870 0.7331347 0.7522945    0
## rf   0.7414917 0.7511280 0.7692888 0.7608828 0.7702860 0.7722196    0
## cart 0.6245685 0.6311240 0.6405180 0.6475274 0.6660796 0.6753470    0
## gbm  0.7428768 0.7483911 0.7550303 0.7571090 0.7555773 0.7836695    0
```

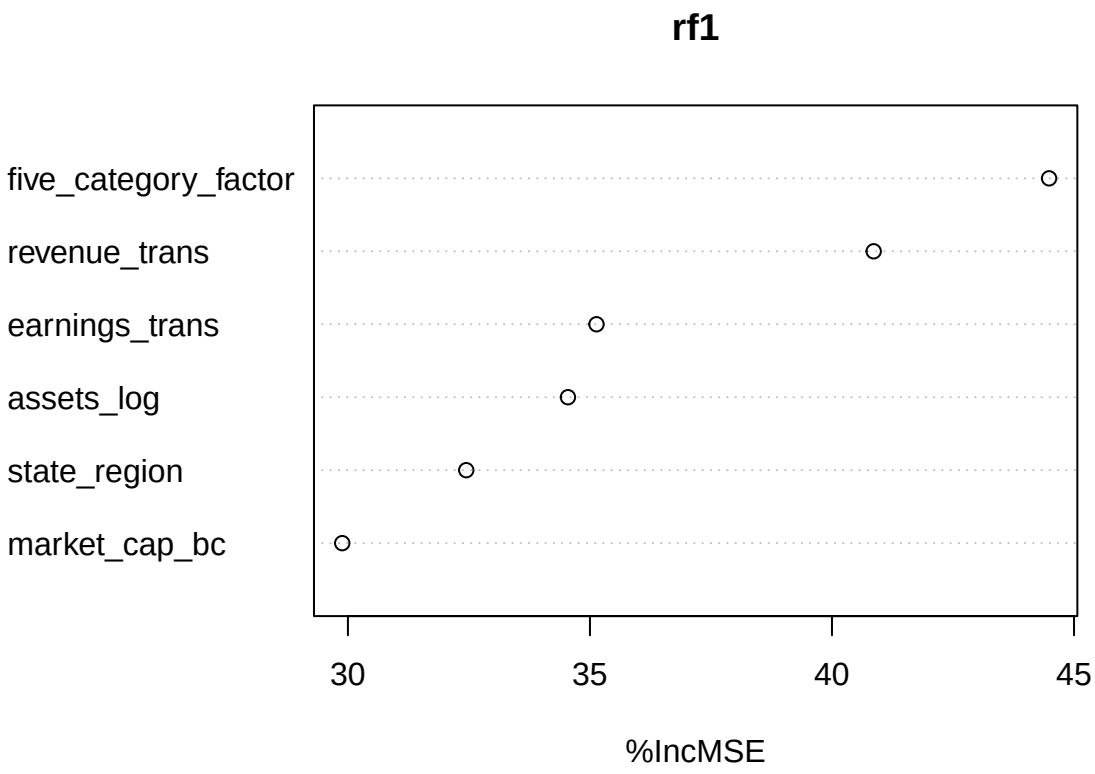
Depend on the above figure, we can see that random forest has the best performance, which has the lowest MAE, lowest RMSE and highest R-square.

So we choose Random Forest model to do the prediction of Total auditing fee based on other variables.

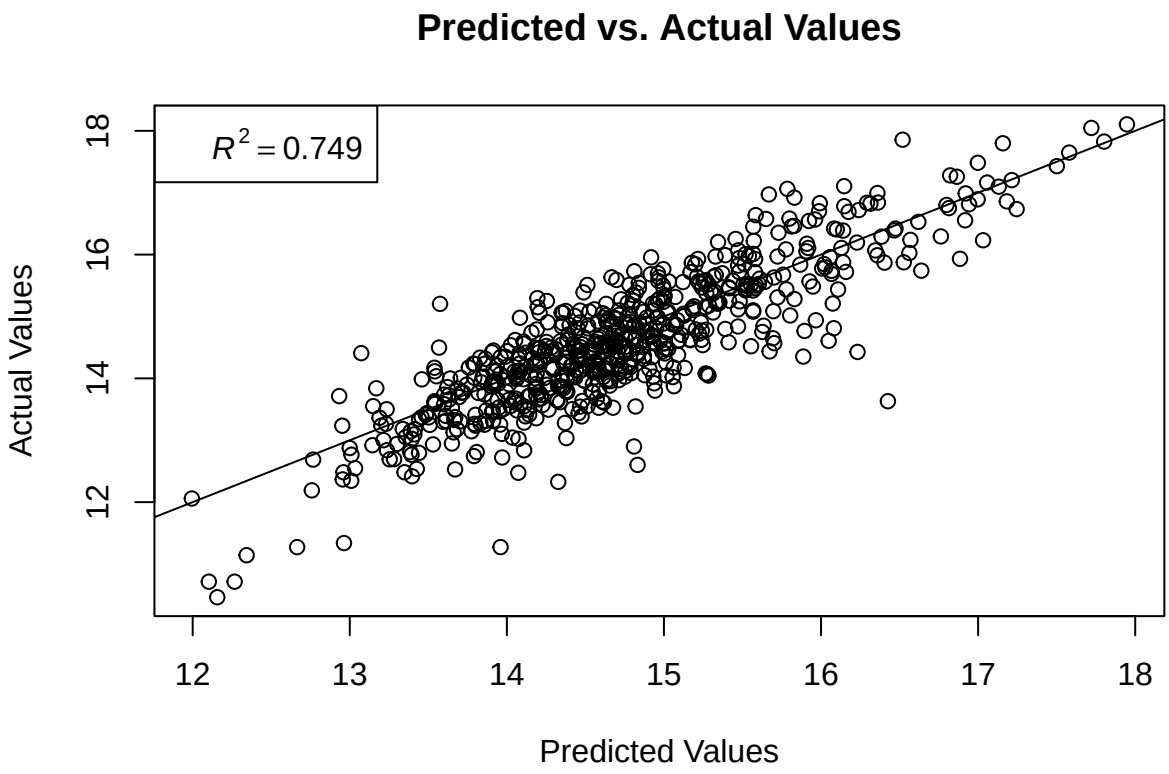
```
##
## Call:
##  randomForest(formula = total_fees_bc ~ five_category_factor +      state_region + market_cap_bc + as
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.2965624
##                     % Var explained: 76.34
```



```
## [1] 0.5824461
```



the explanation of plot(rf1): the explanation of varImpPlot(rf1, type=1):



the explanation of predict v.s. actual values

We add the diagonal line for estimated regression line here.