Predicting Financial Conditions of Companies through 10-K reports and Balance Sheets

CME Assignment I

Rishabh Patil | 2021A7PS0464H

Contents


```
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

1 About the dataset

The dataset is a compilation of various factors from the financial statements of various companies from the 10-K reports and balance sheets. All the relevant data columns that MAY affect the market cap value are presented.

Data Preview:

```
data <- read.csv("Raw Data/Financial Statements.csv")
head(data)
```

```
##   Year Company Category Market.Cap.in.B.USD. Revenue Gross.Profit Net.Income
## 1 2022    AAPL       IT              2066.94  394328       170782      99803
## 2 2021    AAPL       IT              2913.28  365817       152836      94680
## 3 2020    AAPL       IT              2255.97  274515       104956      57411
```

```
## 4 2019    AAPL       IT              1304.76  260174        98392      55256
## 5 2018    AAPL       IT               748.54  265595       101839      59531
## 6 2017    AAPL       IT               868.87  229234        88186      48351
##   Earning.Per.Share EBITDA Share.Holder.Equity Cash.Flow.from.Operating
## 1            6.1100 130541                50672                   122151
## 2            5.6100 120233                63090                   104038
## 3            3.2800  77344                65339                    80674
## 4            2.9700  76477                90488                    69391
## 5            2.9800  81801               107147                    77434
## 6            2.3025  71501               134047                    64225
##   Cash.Flow.from.Investing Cash.Flow.from.Financial.Activities Current.Ratio
## 1                   -22354                            -110749        0.8794
## 2                   -14545                             -93353        1.0746
## 3                    -4289                             -86820        1.3636
## 4                    45896                             -90976        1.5401
## 5                    16066                             -87876        1.1329
## 6                   -46446                             -17974        1.2761
##   Debt.Equity.Ratio      ROE     ROA     ROI Net.Profit.Margin
## 1            2.3695 196.9589 28.2924 66.6994           25.3096
## 2            1.9768 150.0713 26.9742 54.9839           25.8818
## 3            1.7208  87.8664 17.7256 35.0054           20.9136
## 4            1.1940  61.0645 16.3230 30.3113           21.2381
## 5            1.0685  55.5601 16.2775 29.6348           22.4142
## 6            0.8630  36.0702 12.8826 20.9082           21.0924
##   Free.Cash.Flow.per.Share Return.on.Tangible.Equity Number.of.Employees
## 1                   1.3146                  196.9589              164000
## 2                   1.3261                  150.0713              154000
## 3                   1.0183                   87.8664              147000
## 4                  -0.0388                   61.0645              137000
## 5                   0.7414                   55.5601              132000
## 6                   0.0330                   36.0702              123000
##   Inflation.Rate.in.US.
## 1                8.0028
## 2                4.6979
## 3                1.2336
## 4                1.8122
## 5                2.4426
## 6                2.1301
```

The data presented is panel data, that is, it is a combination of both time series and cross sectional data. Therefore traditional regression models cannot be used, but some tweaked versions of it are to be used.

We need to use the plm package for panel data regression.

```
# install.packages('plm')
library(plm)
```

```
## Warning: package 'plm' was built under R version 4.2.3
```

```
pdata = pdata.frame(data, index = c("Company", "Year"))
head(pdata)
```

```
##           Year Company Category Market.Cap.in.B.USD. Revenue Gross.Profit
```

```
## AAPL-2009 2009    AAPL       IT               189.80   42905        17222
## AAPL-2010 2010    AAPL       IT               296.89   65225        25684
## AAPL-2011 2011    AAPL       IT               376.40  108249        43818
## AAPL-2012 2012    AAPL       IT               500.61  156508        68662
## AAPL-2013 2013    AAPL       IT               504.79  170910        64304
## AAPL-2014 2014    AAPL       IT               647.36  182795        70537
##           Net.Income Earning.Per.Share EBITDA Share.Holder.Equity
## AAPL-2009       8235            0.3243  12474               31640
## AAPL-2010      14013            0.5411  19412               47791
## AAPL-2011      25922            0.9886  35604               76615
## AAPL-2012      41733            1.5775  58518              118210
## AAPL-2013      37037            1.4200  55756              123549
## AAPL-2014      39510            1.6125  60449              111547
##           Cash.Flow.from.Operating Cash.Flow.from.Investing
## AAPL-2009                    10159                   -17434
## AAPL-2010                    18595                   -13854
## AAPL-2011                    37529                   -40419
## AAPL-2012                    50856                   -48227
## AAPL-2013                    53666                   -33774
## AAPL-2014                    59713                   -22579
##           Cash.Flow.from.Financial.Activities Current.Ratio Debt.Equity.Ratio
## AAPL-2009                                 663        2.7425            0.0000
## AAPL-2010                                1257        2.0113            0.0000
## AAPL-2011                                1444        1.6084            0.0000
## AAPL-2012                               -1698        1.4958            0.0000
## AAPL-2013                              -16379        1.6786            0.1373
## AAPL-2014                              -37549        1.0801            0.3164
##               ROE     ROA     ROI Net.Profit.Margin Free.Cash.Flow.per.Share
## AAPL-2009 26.0272 17.3365 26.0272           19.1936                   0.3550
## AAPL-2010 29.3214 18.6385 29.3214           21.4841                   0.2857
## AAPL-2011 33.8341 22.2753 33.8341           23.9466                   0.6278
## AAPL-2012 35.3041 23.7033 35.3041           26.6651                   0.3394
## AAPL-2013 29.9776 17.8923 26.3592           21.6705                   0.1363
## AAPL-2014 35.4201 17.0420 28.1142           21.6144                   0.3032
##           Return.on.Tangible.Equity Number.of.Employees Inflation.Rate.in.US.
## AAPL-2009                   26.4052               36800               -0.3555
## AAPL-2010                   30.0013               49400                1.6400
## AAPL-2011                   35.9115               63300                3.1568
## AAPL-2012                   36.9806               76100                2.0693
## AAPL-2013                   31.4425               84400                1.4648
## AAPL-2014                   38.4380               97000                1.6222
```

1.1 Model fitting:

We'll start our regression model with Pooling method panel regression. In **Pooled OLS Regression**, we treat each row as a new data point, i.e: we remove the time variance and company specifics from the data, and run a linear regression model on the data we have.

1.1.1 Dependent and Independent Variables:

Here `Market Cap` is the ***Dependent Variable*** and rest all are ***Independent Variables*** barring `Year`, `Company Type` and `Company`.

Reasons for selecting the independent variable:

1. Financial Performance: Companies are assessed on their market cap, and it is considered as a good indicator for the their performance.
2. Investors Point Of View: Investors see this as a good indicator to make informed investment stratergies.

Segregating them for model fitting:

```
dep <- c(names(pdata))[5:23]
Y <- c(names(pdata))[4]
```

###***Pooled OLS Regression:***

$$Y_{it} = \beta_0 + \sum_{k=1}^{n} \beta_k (X_k)_{it}$$

```
pooledmethod <- plm(Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
    Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
    Cash.Flow.from.Operating + Cash.Flow.from.Investing +
    Cash.Flow.from.Financial.Activities +
    Current.Ratio + ROE + ROA + ROI + Net.Profit.Margin + Free.Cash.Flow.per.Share +
    Return.on.Tangible.Equity + Inflation.Rate.in.US. + Debt.Equity.Ratio +
    Number.of.Employees, data = pdata, model = "pooling")
summary(pooledmethod)
```

```
## Pooling Model
##
## Call:
## plm(formula = Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
##     Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
##     Cash.Flow.from.Operating + Cash.Flow.from.Investing + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + ROE + ROA + ROI + Net.Profit.Margin + Free.Cash.Flow.per.Share +
##     Return.on.Tangible.Equity + Inflation.Rate.in.US. + Debt.Equity.Ratio +
##     Number.of.Employees, data = pdata, model = "pooling")
##
## Unbalanced Panel: n = 12, T = 8-15, N = 160
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -722.429 -104.457  -11.288   70.383  931.734
##
## Coefficients:
##                                    Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)                     -3.3340e+01  4.4671e+01 -0.7463 0.4567142
## Revenue                         -1.4314e-03  8.8592e-04 -1.6157 0.1084170
## Gross.Profit                     8.0234e-03  2.1823e-03  3.6766 0.0003360
## Net.Income                       2.1778e-02  4.9784e-03  4.3745 2.361e-05
## Earning.Per.Share                4.6224e+00  3.2778e+00  1.4102 0.1606912
## EBITDA                          -2.3665e-03  4.4922e-03 -0.5268 0.5991645
## Share.Holder.Equity             -1.4242e-03  6.1577e-04 -2.3129 0.0221879
## Cash.Flow.from.Operating        -1.6627e-03  1.5672e-03 -1.0609 0.2905485
```

```
## Cash.Flow.from.Investing            -3.3714e-04  2.1763e-03 -0.1549 0.8771131
## Cash.Flow.from.Financial.Activities -4.0425e-03  2.8346e-03 -1.4261 0.1560566
## Current.Ratio                        5.0623e+01  1.4862e+01  3.4062 0.0008601
## ROE                                  1.2391e-01  5.9031e-01  0.2099 0.8340496
## ROA                                 -7.9165e+00  4.7632e+00 -1.6620 0.0987498
## ROI                                  4.1989e-02  1.9583e-01  0.2144 0.8305398
## Net.Profit.Margin                   -3.8174e+00  2.9805e+00 -1.2808 0.2023795
## Free.Cash.Flow.per.Share             3.6642e+00  1.7736e+00  2.0660 0.0406729
## Return.on.Tangible.Equity           -4.0931e-02  1.6977e-01 -0.2411 0.8098358
## Inflation.Rate.in.US.               -5.7433e-01  1.0010e+01 -0.0574 0.9543278
## Debt.Equity.Ratio                   -1.1753e+01  1.1662e+01 -1.0078 0.3153013
## Number.of.Employees                  3.3846e-04  1.8519e-04  1.8276 0.0697325
##
## (Intercept)
## Revenue
## Gross.Profit                        ***
## Net.Income                          ***
## Earning.Per.Share
## EBITDA
## Share.Holder.Equity                 *
## Cash.Flow.from.Operating
## Cash.Flow.from.Investing
## Cash.Flow.from.Financial.Activities
## Current.Ratio                       ***
## ROE
## ROA                                 .
## ROI
## Net.Profit.Margin
## Free.Cash.Flow.per.Share            *
## Return.on.Tangible.Equity
## Inflation.Rate.in.US.
## Debt.Equity.Ratio
## Number.of.Employees                 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    46554000
## Residual Sum of Squares: 6975800
## R-Squared:      0.85016
## Adj. R-Squared: 0.82982
## F-statistic: 41.8054 on 19 and 140 DF, p-value: < 2.22e-16
```

According to above summary, `Gross Profit`, `Net Income`, `Share.Holder.Equity` and `Current.Ratio` & `Free.Cash.Flow.per.Share` are significant in Pooled OLS Method.

1.1.2 *Fixed Effect Method:*

```
femethod <- plm(Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
    Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
    Cash.Flow.from.Operating + Cash.Flow.from.Investing +
↪   Cash.Flow.from.Financial.Activities +
    Current.Ratio + ROE + ROA + ROI + Net.Profit.Margin + Free.Cash.Flow.per.Share +
```

```
    Return.on.Tangible.Equity + Inflation.Rate.in.US. + Debt.Equity.Ratio +
    Number.of.Employees, data = pdata, model = "within")
summary(femethod)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
##     Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
##     Cash.Flow.from.Operating + Cash.Flow.from.Investing + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + ROE + ROA + ROI + Net.Profit.Margin + Free.Cash.Flow.per.Share +
##     Return.on.Tangible.Equity + Inflation.Rate.in.US. + Debt.Equity.Ratio +
##     Number.of.Employees, data = pdata, model = "within")
##
## Unbalanced Panel: n = 12, T = 8-15, N = 160
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -627.6915  -75.4975   -8.4776   56.0140  831.3425
##
## Coefficients:
##                                      Estimate  Std. Error t-value  Pr(>|t|)
## Revenue                           -0.00471871  0.00228844 -2.0620 0.0412176
## Gross.Profit                       0.00998796  0.00500888  1.9941 0.0482551
## Net.Income                         0.02308152  0.00583379  3.9565 0.0001248
## Earning.Per.Share                  1.97002936  3.36110691  0.5861 0.5588156
## EBITDA                             0.00094730  0.00659223  0.1437 0.8859621
## Share.Holder.Equity               -0.00171668  0.00096493 -1.7791 0.0775822
## Cash.Flow.from.Operating          -0.00024678  0.00156648 -0.1575 0.8750687
## Cash.Flow.from.Investing          -0.00171203  0.00214387 -0.7986 0.4260074
## Cash.Flow.from.Financial.Activities -0.00570943  0.00298006 -1.9159 0.0575930
## Current.Ratio                     30.23659747 24.09081859  1.2551 0.2117082
## ROE                               -0.10119349  0.61256663 -0.1652 0.8690484
## ROA                               -8.22958255  5.79983378 -1.4189 0.1583298
## ROI                                0.01394018  0.18879591  0.0738 0.9412542
## Net.Profit.Margin                 -2.88705796  3.05607324 -0.9447 0.3465805
## Free.Cash.Flow.per.Share           2.29798007  1.80072749  1.2761 0.2041989
## Return.on.Tangible.Equity          0.01082397  0.16732563  0.0647 0.9485225
## Inflation.Rate.in.US.             -7.39021459 10.63555059 -0.6949 0.4883930
## Debt.Equity.Ratio                 -9.29603565 14.59865125 -0.6368 0.5254007
## Number.of.Employees                0.00080495  0.00034975  2.3015 0.0229680
##
## Revenue                           *
## Gross.Profit                      *
## Net.Income                        ***
## Earning.Per.Share
## EBITDA
## Share.Holder.Equity               .
## Cash.Flow.from.Operating
## Cash.Flow.from.Investing
## Cash.Flow.from.Financial.Activities .
## Current.Ratio
## ROE
```

```
## ROA
## ROI
## Net.Profit.Margin
## Free.Cash.Flow.per.Share
## Return.on.Tangible.Equity
## Inflation.Rate.in.US.
## Debt.Equity.Ratio
## Number.of.Employees                 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    26946000
## Residual Sum of Squares: 5806400
## R-Squared:      0.78452
## Adj. R-Squared: 0.73441
## F-statistic: 24.7192 on 19 and 129 DF, p-value: < 2.22e-16
```

From here, the significant variables are: `Net Income`, `Gross Profit`, `Revenue`, `Number Of Employees`

Normalising the data

```
dep
```

```
##  [1] "Revenue"                          "Gross.Profit"
##  [3] "Net.Income"                       "Earning.Per.Share"
##  [5] "EBITDA"                           "Share.Holder.Equity"
##  [7] "Cash.Flow.from.Operating"         "Cash.Flow.from.Investing"
##  [9] "Cash.Flow.from.Financial.Activities" "Current.Ratio"
## [11] "Debt.Equity.Ratio"                "ROE"
## [13] "ROA"                              "ROI"
## [15] "Net.Profit.Margin"                "Free.Cash.Flow.per.Share"
## [17] "Return.on.Tangible.Equity"        "Number.of.Employees"
## [19] "Inflation.Rate.in.US."
```

```
# install.packages('dplyr')
pdata[is.na(pdata)] <- 0
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plm':
##
##     between, lag, lead
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
normalpdata <- pdata %>%
    mutate_at(setdiff(dep, c("Debt.Equity.Ratio", "Number.of.Employees")),
        ~(scale(.) %>%
            as.vector))
head(normalpdata)
```

```
##           Year Company Category Market.Cap.in.B.USD.    Revenue Gross.Profit
## AAPL-2009 2009    AAPL       IT               189.80 -0.3630216   -0.4838469
## AAPL-2010 2010    AAPL       IT               296.89 -0.1171711   -0.2807708
## AAPL-2011 2011    AAPL       IT               376.40  0.3567299    0.1544199
## AAPL-2012 2012    AAPL       IT               500.61  0.8882934    0.7506412
## AAPL-2013 2013    AAPL       IT               504.79  1.0469286    0.6460553
## AAPL-2014 2014    AAPL       IT               647.36  1.1778396    0.7956385
##            Net.Income Earning.Per.Share      EBITDA Share.Holder.Equity
## AAPL-2009 -0.20814335     -0.0742109882 -0.29230944          -0.4720675
## AAPL-2010  0.08942449     -0.0497814540 -0.02892807          -0.1735179
## AAPL-2011  0.70273971      0.0006438992  0.58575501           0.3592909
## AAPL-2012  1.51700850      0.0670025370  1.45561964           1.1281704
## AAPL-2013  1.27516382      0.0492550664  1.35076819           1.2268612
## AAPL-2014  1.40252368      0.0709464194  1.52892455           1.0050055
##           Cash.Flow.from.Operating Cash.Flow.from.Investing
## AAPL-2009              -0.38942253               -0.4848157
## AAPL-2010              -0.08041395               -0.2637845
## AAPL-2011               0.61313377               -1.9039225
## AAPL-2012               1.10129847               -2.3859927
## AAPL-2013               1.20422807               -1.4936564
## AAPL-2014               1.42572817               -0.8024709
##           Cash.Flow.from.Financial.Activities Current.Ratio Debt.Equity.Ratio
## AAPL-2009                           0.4566272    0.42575375            0.0000
## AAPL-2010                           0.4859147   -0.01449119            0.0000
## AAPL-2011                           0.4951348   -0.25707145            0.0000
## AAPL-2012                           0.3402166   -0.32486628            0.0000
## AAPL-2013                          -0.3836393   -0.21480505            0.1373
## AAPL-2014                          -1.4274393   -0.57515324            0.3164
##                ROE      ROA       ROI Net.Profit.Margin
## AAPL-2009 0.3036092 1.085608 0.1514793         0.4110380
## AAPL-2010 0.3771777 1.233521 0.1867549         0.5819483
## AAPL-2011 0.4779587 1.646678 0.2350787         0.7656928
## AAPL-2012 0.5107879 1.808905 0.2508200         0.9685392
## AAPL-2013 0.3918325 1.148749 0.1550345         0.5958570
## AAPL-2014 0.5133785 1.052151 0.1738277         0.5916709
##           Free.Cash.Flow.per.Share Return.on.Tangible.Equity
## AAPL-2009              0.010522076                0.01961814
## AAPL-2010              0.006025810                0.05231507
## AAPL-2011              0.028221664                0.10605255
## AAPL-2012              0.009509930                0.11577316
## AAPL-2013             -0.003667439                0.06541893
## AAPL-2014              0.007161231                0.12902432
##           Number.of.Employees Inflation.Rate.in.US.
```

```
## AAPL-2009                36800           -1.32038471
## AAPL-2010                49400           -0.30182511
## AAPL-2011                63300            0.47239248
## AAPL-2012                76100           -0.08269826
## AAPL-2013                84400           -0.39125214
## AAPL-2014                97000           -0.31091073
```

trying standardized data(not recommended as we lose meaning):

```
femethod2 <- plm(Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
    Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
    Cash.Flow.from.Operating + Cash.Flow.from.Investing +
↪   Cash.Flow.from.Financial.Activities +
    Current.Ratio + ROE + ROA + ROI + Net.Profit.Margin + Free.Cash.Flow.per.Share +
    Return.on.Tangible.Equity + Inflation.Rate.in.US. + Debt.Equity.Ratio +
    Number.of.Employees, data = normalpdata, model = "within")
summary(femethod2)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
##     Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
##     Cash.Flow.from.Operating + Cash.Flow.from.Investing + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + ROE + ROA + ROI + Net.Profit.Margin + Free.Cash.Flow.per.Share +
##     Return.on.Tangible.Equity + Inflation.Rate.in.US. + Debt.Equity.Ratio +
##     Number.of.Employees, data = normalpdata, model = "within")
##
## Unbalanced Panel: n = 12, T = 9-15, N = 161
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -627.6167  -69.6376   -7.2944   55.8018  831.7061
##
## Coefficients:
##                                      Estimate  Std. Error t-value  Pr(>|t|)
## Revenue                           -4.2124e+02  2.0700e+02 -2.0349 0.0438898
## Gross.Profit                       4.1314e+02  2.0820e+02  1.9844 0.0493196
## Net.Income                         4.4846e+02  1.1302e+02  3.9679 0.0001192
## Earning.Per.Share                  1.7206e+01  2.9758e+01  0.5782 0.5641308
## EBITDA                             2.1635e+01  1.7319e+02  0.1249 0.9007786
## Share.Holder.Equity               -9.1632e+01  5.2049e+01 -1.7605 0.0806778
## Cash.Flow.from.Operating          -6.9846e+00  4.2668e+01 -0.1637 0.8702250
## Cash.Flow.from.Investing          -2.7718e+01  3.4646e+01 -0.8000 0.4251456
## Cash.Flow.from.Financial.Activities -1.1519e+02  6.0298e+01 -1.9103 0.0582988
## Current.Ratio                      5.0048e+01  3.9922e+01  1.2536 0.2122233
## ROE                               -4.3480e+00  2.7366e+01 -0.1589 0.8740087
## ROA                               -7.2060e+01  5.0935e+01 -1.4147 0.1595363
## ROI                                1.3252e+00  1.7591e+01  0.0753 0.9400647
## Net.Profit.Margin                 -3.7105e+01  4.0792e+01 -0.9096 0.3647026
## Free.Cash.Flow.per.Share           3.5721e+01  2.7688e+01  1.2901 0.1993000
## Return.on.Tangible.Equity          1.2442e+00  1.8362e+01  0.0678 0.9460820
## Inflation.Rate.in.US.             -1.4140e+01  2.0783e+01 -0.6804 0.4974791
```

```
## Debt.Equity.Ratio                    -9.2392e+00  1.4566e+01 -0.6343 0.5269946
## Number.of.Employees                   7.9414e-04  3.4857e-04  2.2783 0.0243428
##
## Revenue                              *
## Gross.Profit                         *
## Net.Income                           ***
## Earning.Per.Share
## EBITDA
## Share.Holder.Equity                  .
## Cash.Flow.from.Operating
## Cash.Flow.from.Investing
## Cash.Flow.from.Financial.Activities .
## Current.Ratio
## ROE
## ROA
## ROI
## Net.Profit.Margin
## Free.Cash.Flow.per.Share
## Return.on.Tangible.Equity
## Inflation.Rate.in.US.
## Debt.Equity.Ratio
## Number.of.Employees                  *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    26960000
## Residual Sum of Squares: 5825300
## R-Squared:      0.78393
## Adj. R-Squared: 0.73407
## F-statistic: 24.8238 on 19 and 130 DF, p-value: < 2.22e-16
```

little to no effect on significant variables when standardized.

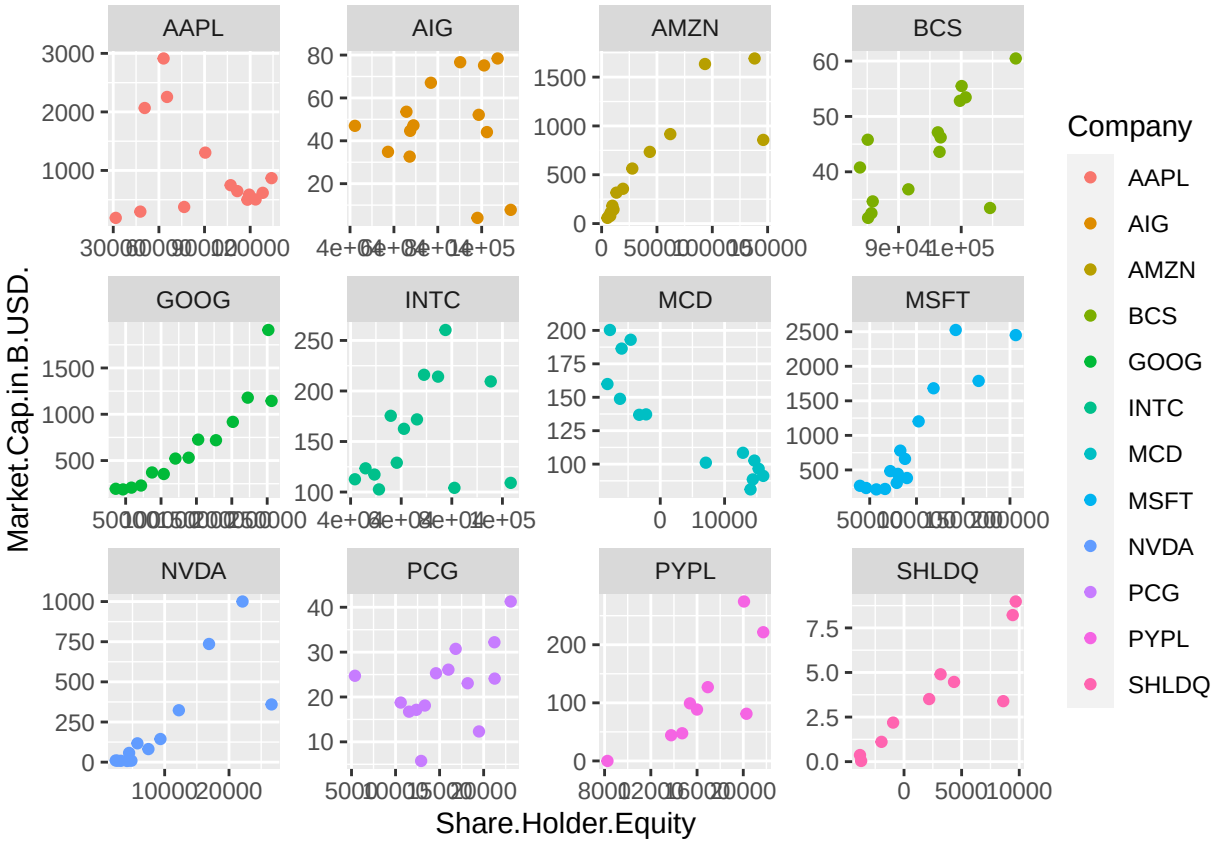
####**Plotting the results:**

```
pdata[is.na(pdata)] <- 0

library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
ggplot(data = pdata, aes(x = Share.Holder.Equity, y = Market.Cap.in.B.USD.,
    color = Company)) + geom_point() + facet_wrap(~Company, nrow = 3,
    scale = "free")
```

```
## Warning: Combining variables of class <pseries> and <factor> was deprecated in ggplot2
## 3.4.0.
## i Please ensure your variables are compatible before plotting (location:
##   `join_keys()`)
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



using log scale for percentage change:

```
ggplot(data = pdata, aes(x = log(Share.Holder.Equity), y = log(Market.Cap.in.B.USD.),
    color = Company)) + geom_point() + facet_wrap(~Company, nrow = 3)
```

```
## Warning in log(Share.Holder.Equity): NaNs produced

## Warning in log(Share.Holder.Equity): NaNs produced

## Warning: Removed 11 rows containing missing values (`geom_point()`).
```

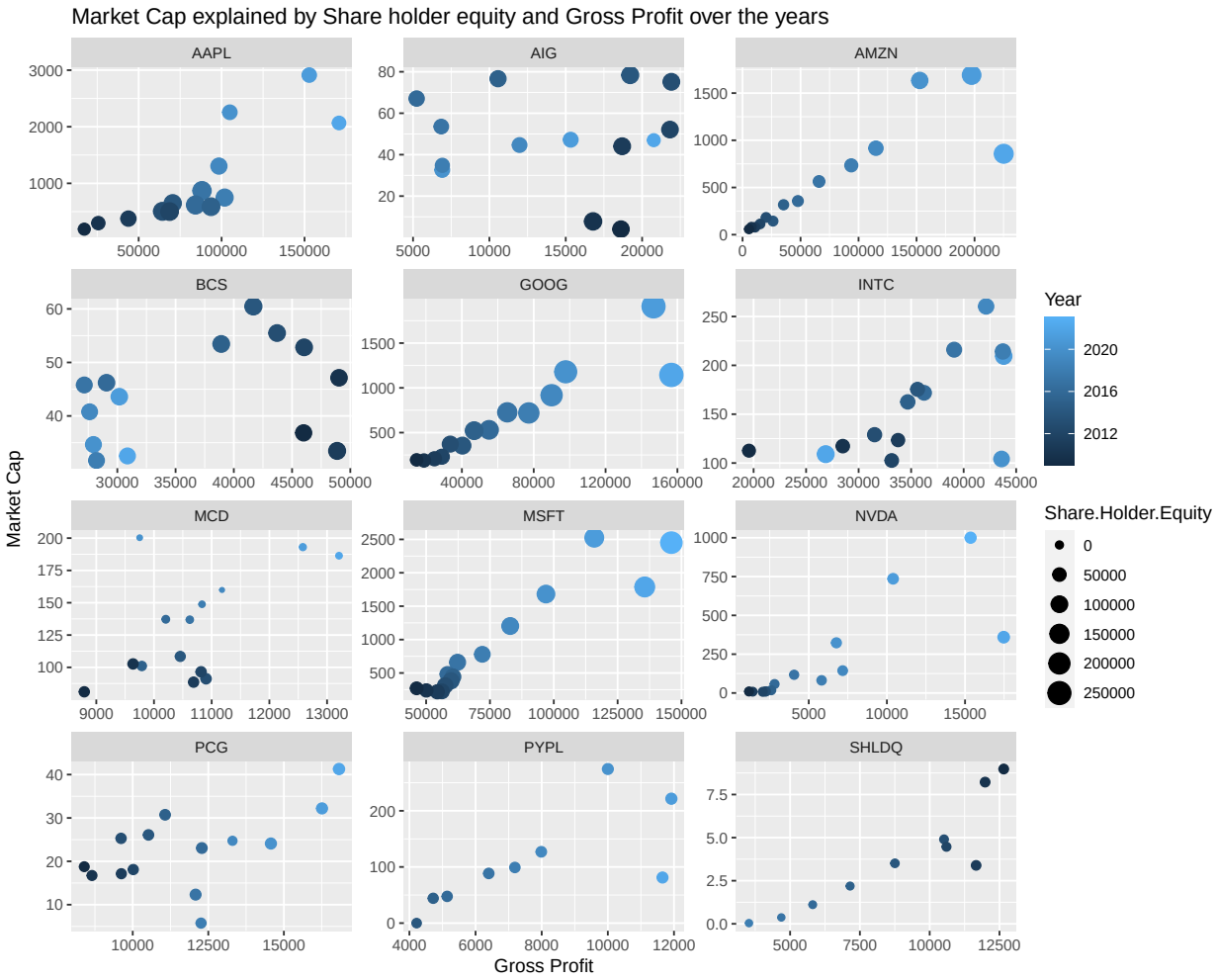
For

```
# install.packages('tidyverse')
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'purrr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.3

## Warning: package 'forcats' was built under R version 4.2.3

## Warning: package 'lubridate' was built under R version 4.2.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
```

```
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## v readr     2.1.4
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::between() masks plm::between()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks plm::lag(), stats::lag()
## x dplyr::lead()    masks plm::lead()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
data[is.na(data)] <- 0
data %>%
    ggplot(aes(x = Gross.Profit, y = Market.Cap.in.B.USD., size = Share.Holder.Equity,
        color = Year)) + geom_point() + facet_wrap(~Company,
    nrow = 4, scale = "free") + labs(title = "Market Cap explained by Share holder equity
    ↪  and Gross Profit over the years",
    x = "Gross Profit", y = "Market Cap")
```




```r
# install.packages('tidyverse')
library(tidyverse)
data[is.na(data)] <- 0
```

```
data %>%
    ggplot(aes(x = Net.Income, y = Market.Cap.in.B.USD., size = Gross.Profit,
        color = Year)) + geom_point() + facet_wrap(~Company,
    nrow = 4, scale = "free") + labs(title = "Market Cap explained by Net Income and
    ↪  Gross Profit over the years",
    x = "Net Income", y = "Market Cap")
```

Random Effect:

```
# remethod <- plm(Market.Cap.in.B.USD.~
#
↪   Revenue+Gross.Profit+Net.Income+Share.Holder.Equity+Cash.Flow.from.Operating+Cash.Flow.from.Investi
# summary(remethod)
```

Here the variables are too many for the given data set to perform Random Effects regression, hence we remove some variables with high p-values from FE.

```
remethod <- plm(Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
    Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
    Cash.Flow.from.Financial.Activities + Current.Ratio + Free.Cash.Flow.per.Share +
```

```
    Number.of.Employees, data = pdata, model = "random")
summary(remethod)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
##     Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
##     Cash.Flow.from.Financial.Activities + Current.Ratio + Free.Cash.Flow.per.Share +
##     Number.of.Employees, data = pdata, model = "random")
##
## Unbalanced Panel: n = 12, T = 9-15, N = 161
##
## Effects:
##                    var  std.dev share
## idiosyncratic 44474.14   210.89 0.998
## individual      103.07    10.15 0.002
## theta:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01027 0.01584 0.01584 0.01546 0.01584 0.01694
##
## Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -738.68  -81.28   -5.53   -0.01   70.13  997.53
##
## Coefficients:
##                                      Estimate  Std. Error z-value  Pr(>|z|)
## (Intercept)                        -6.9190e+01  3.7617e+01 -1.8393 0.0658652
## Revenue                            -7.2894e-04  8.2344e-04 -0.8852 0.3760263
## Gross.Profit                        7.6776e-03  1.9524e-03  3.9324 8.412e-05
## Net.Income                          1.6105e-02  4.4411e-03  3.6264 0.0002874
## Earning.Per.Share                  -5.3306e-01  2.8502e+00 -0.1870 0.8516404
## EBITDA                             -2.4791e-03  3.6877e-03 -0.6723 0.5014171
## Share.Holder.Equity                -1.6171e-03  5.6304e-04 -2.8722 0.0040766
## Cash.Flow.from.Financial.Activities -3.7489e-03  1.7837e-03 -2.1017 0.0355765
## Current.Ratio                       2.7184e+01  1.1352e+01  2.3946 0.0166373
## Free.Cash.Flow.per.Share            1.4720e+00  1.5691e+00  0.9382 0.3481578
## Number.of.Employees                 2.0891e-04  1.6644e-04  1.2552 0.2094136
##
## (Intercept)                        .
## Revenue
## Gross.Profit                       ***
## Net.Income                         ***
## Earning.Per.Share
## EBITDA
## Share.Holder.Equity                **
## Cash.Flow.from.Financial.Activities *
## Current.Ratio                      *
## Free.Cash.Flow.per.Share
## Number.of.Employees
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Total Sum of Squares:    46029000
## Residual Sum of Squares: 7612000
## R-Squared:      0.83463
## Adj. R-Squared: 0.8236
## Chisq: 757.467 on 10 DF, p-value: < 2.22e-16
```

Here we find that `Gross Profit`, `Net Income`, `Current Ratio`, `Share.Holder.Equity`, `Cash.Flow.from.Financial.Activi` and `Current Ratio` are significant.

1.1.3 RE vs FE

using the Hausman Test to check which one is better for our data. Fisrt run a FE model on restricted varaibles:

```
femthod2 <- plm(Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
    Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
    Cash.Flow.from.Financial.Activities + Current.Ratio + Free.Cash.Flow.per.Share +
    Number.of.Employees, data = pdata, model = "within")
summary(femethod2)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
##     Net.Income + Earning.Per.Share + EBITDA + Share.Holder.Equity +
##     Cash.Flow.from.Operating + Cash.Flow.from.Investing + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + ROE + ROA + ROI + Net.Profit.Margin + Free.Cash.Flow.per.Share +
##     Return.on.Tangible.Equity + Inflation.Rate.in.US. + Debt.Equity.Ratio +
##     Number.of.Employees, data = normalpdata, model = "within")
##
## Unbalanced Panel: n = 12, T = 9-15, N = 161
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -627.6167  -69.6376   -7.2944   55.8018  831.7061
##
## Coefficients:
##                                      Estimate  Std. Error t-value  Pr(>|t|)
## Revenue                           -4.2124e+02  2.0700e+02 -2.0349 0.0438898
## Gross.Profit                       4.1314e+02  2.0820e+02  1.9844 0.0493196
## Net.Income                         4.4846e+02  1.1302e+02  3.9679 0.0001192
## Earning.Per.Share                  1.7206e+01  2.9758e+01  0.5782 0.5641308
## EBITDA                             2.1635e+01  1.7319e+02  0.1249 0.9007786
## Share.Holder.Equity               -9.1632e+01  5.2049e+01 -1.7605 0.0806778
## Cash.Flow.from.Operating          -6.9846e+00  4.2668e+01 -0.1637 0.8702250
## Cash.Flow.from.Investing          -2.7718e+01  3.4646e+01 -0.8000 0.4251456
## Cash.Flow.from.Financial.Activities -1.1519e+02  6.0298e+01 -1.9103 0.0582988
## Current.Ratio                      5.0048e+01  3.9922e+01  1.2536 0.2122233
## ROE                               -4.3480e+00  2.7366e+01 -0.1589 0.8740087
## ROA                               -7.2060e+01  5.0935e+01 -1.4147 0.1595363
## ROI                                1.3252e+00  1.7591e+01  0.0753 0.9400647
```

```
## Net.Profit.Margin                    -3.7105e+01  4.0792e+01 -0.9096 0.3647026
## Free.Cash.Flow.per.Share              3.5721e+01  2.7688e+01  1.2901 0.1993000
## Return.on.Tangible.Equity             1.2442e+00  1.8362e+01  0.0678 0.9460820
## Inflation.Rate.in.US.                -1.4140e+01  2.0783e+01 -0.6804 0.4974791
## Debt.Equity.Ratio                    -9.2392e+00  1.4566e+01 -0.6343 0.5269946
## Number.of.Employees                   7.9414e-04  3.4857e-04  2.2783 0.0243428
##
## Revenue                              *
## Gross.Profit                         *
## Net.Income                           ***
## Earning.Per.Share
## EBITDA
## Share.Holder.Equity                  .
## Cash.Flow.from.Operating
## Cash.Flow.from.Investing
## Cash.Flow.from.Financial.Activities .
## Current.Ratio
## ROE
## ROA
## ROI
## Net.Profit.Margin
## Free.Cash.Flow.per.Share
## Return.on.Tangible.Equity
## Inflation.Rate.in.US.
## Debt.Equity.Ratio
## Number.of.Employees                  *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    26960000
## Residual Sum of Squares: 5825300
## R-Squared:      0.78393
## Adj. R-Squared: 0.73407
## F-statistic: 24.8238 on 19 and 130 DF, p-value: < 2.22e-16
```

```
phtest(femethod2, remethod)
```

```
##
##  Hausman Test
##
## data:  Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +  ...
## chisq = 149.48, df = 10, p-value < 2.2e-16
## alternative hypothesis: one model is inconsistent
```

The p-value is significant, i.e p-value <0.05, therefore we use the prior, Fixed Effects model for our data

1.1.4 pooled vs FE

```
pFtest(femethod, pooledmethod)
```

```
##
##  F test for individual effects
##
## data:  Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +  ...
## F = 2.362, df1 = 11, df2 = 129, p-value = 0.01086
## alternative hypothesis: significant effects
```

```
pFtest(femethod2, pooledmethod)
```

```
##
##  F test for individual effects
##
## data:  Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +  ...
## F = 2.5677, df1 = 10, df2 = 130, p-value = 0.007244
## alternative hypothesis: significant effects
```

p-value <0.05 : therefore reject Null, and alt is : Fixed Effect Therefore we go ahead with fixed effect.

```
summary(fixef(femethod))
```

```
##        Estimate Std. Error t-value Pr(>|t|)
## AAPL   -27.2138   176.4359 -0.1542  0.87766
## AIG    164.9517   132.7986  1.2421  0.21645
## AMZN   196.9892   106.2741  1.8536  0.06608 .
## BCS    -83.1089   170.6574 -0.4870  0.62709
## GOOG    28.8531   178.2109  0.1619  0.87163
## INTC  -156.0812   101.4499 -1.5385  0.12637
## MCD   -167.6060   139.4827 -1.2016  0.23171
## MSFT   -76.2292   139.6485 -0.5459  0.58610
## NVDA   144.8175   129.1716  1.1211  0.26432
## PCG      6.7491    70.8521  0.0953  0.92426
## PYPL   138.8054    90.7541  1.5295  0.12860
## SHLDQ -145.0637   105.9125 -1.3697  0.17317
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We'll describe each coefficient later on after variable selection

2 Variable Selection

2.1 LASSO for variable selection

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.2.3
```

```
## Loading required package: Matrix
```

```
## Warning: package 'Matrix' was built under R version 4.2.3
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-8
```

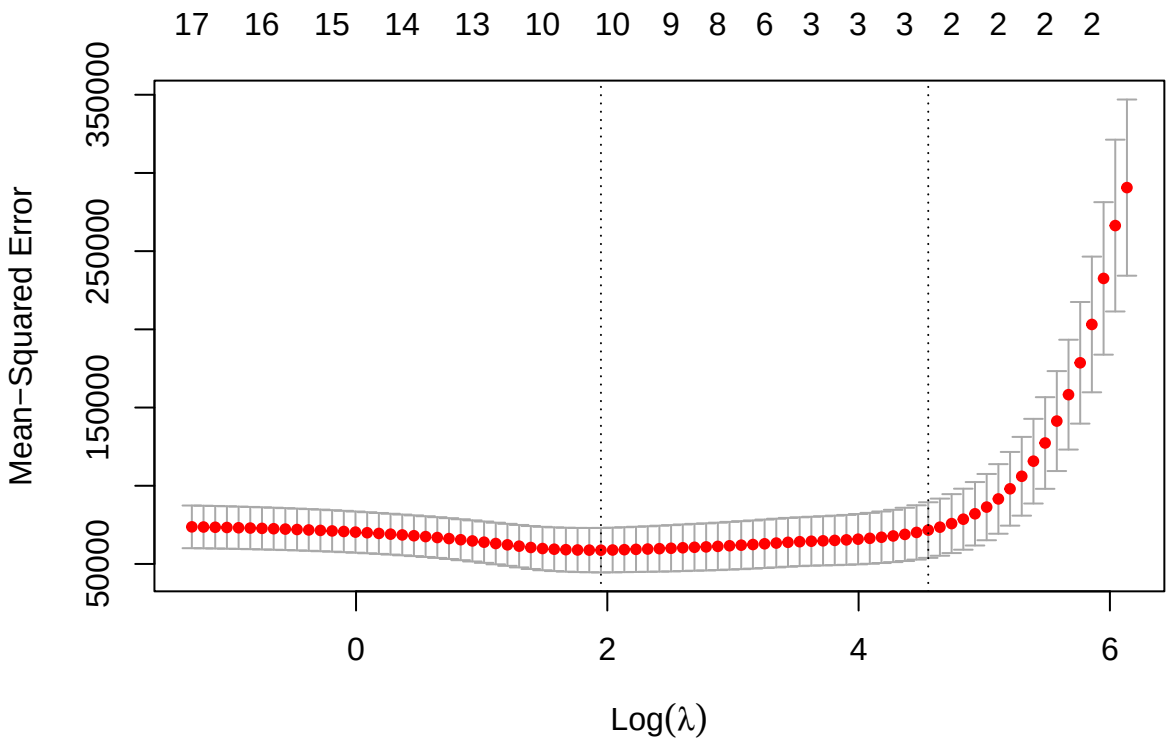
```
formula <- Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
    Share.Holder.Equity + Cash.Flow.from.Operating + Cash.Flow.from.Investing +
    Cash.Flow.from.Financial.Activities + Current.Ratio + ROE +
    ROA + ROI + Net.Profit.Margin + Free.Cash.Flow.per.Share +
    Return.on.Tangible.Equity + Inflation.Rate.in.US. + Debt.Equity.Ratio +
    Number.of.Employees
lasso_model <- cv.glmnet(model.matrix(formula, data = pdata),
    pdata$Market.Cap.in.B.USD., alpha = 1)
# summary(lasso_model)
lasso_selected_variables <- coef(lasso_model, s = "lambda.min") %>%
    as.matrix() %>%
    as.logical() %>%
    colnames()
lasso_selected_variables <- names(lasso_selected_variables[lasso_selected_variables !=
    0])
lasso_selected_variables
```

```
## NULL
```

```
X <- model.matrix(formula, data = pdata)
y <- as.numeric(pdata$Market.Cap.in.B.USD.)

lasso_model <- cv.glmnet(X, y, alpha = 1, family = "gaussian")
best_lambda <- lasso_model$lambda.min
final_lasso_model <- glmnet(X, y, alpha = 1, family = "gaussian",
    lambda = best_lambda)
summary(final_lasso_model)
```

```
##           Length Class     Mode
## a0         1     -none-    numeric
## beta      18     dgCMatrix S4
## df         1     -none-    numeric
## dim        2     -none-    numeric
## lambda     1     -none-    numeric
## dev.ratio  1     -none-    numeric
## nulldev    1     -none-    numeric
## npasses    1     -none-    numeric
## jerr       1     -none-    numeric
## offset     1     -none-    logical
## call       6     -none-    call
## nobs       1     -none-    numeric
```

```r
coef_fe_lasso <- coef(final_lasso_model)
plot(lasso_model)
```




```r
print(coef_fe_lasso)
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                                                s0
## (Intercept)                          -6.005103e+01
## (Intercept)                           .
## Revenue                               .
## Gross.Profit                          5.305109e-03
## Net.Income                            1.484947e-02
## Share.Holder.Equity                  -9.948362e-04
## Cash.Flow.from.Operating              .
## Cash.Flow.from.Investing              .
## Cash.Flow.from.Financial.Activities -2.745923e-03
## Current.Ratio                         3.115969e+01
## ROE                                   .
## ROA                                  -2.511428e+00
## ROI                                   .
## Net.Profit.Margin                    -1.079933e+00
## Free.Cash.Flow.per.Share              8.039784e-01
## Return.on.Tangible.Equity             .
## Inflation.Rate.in.US.                 .
```

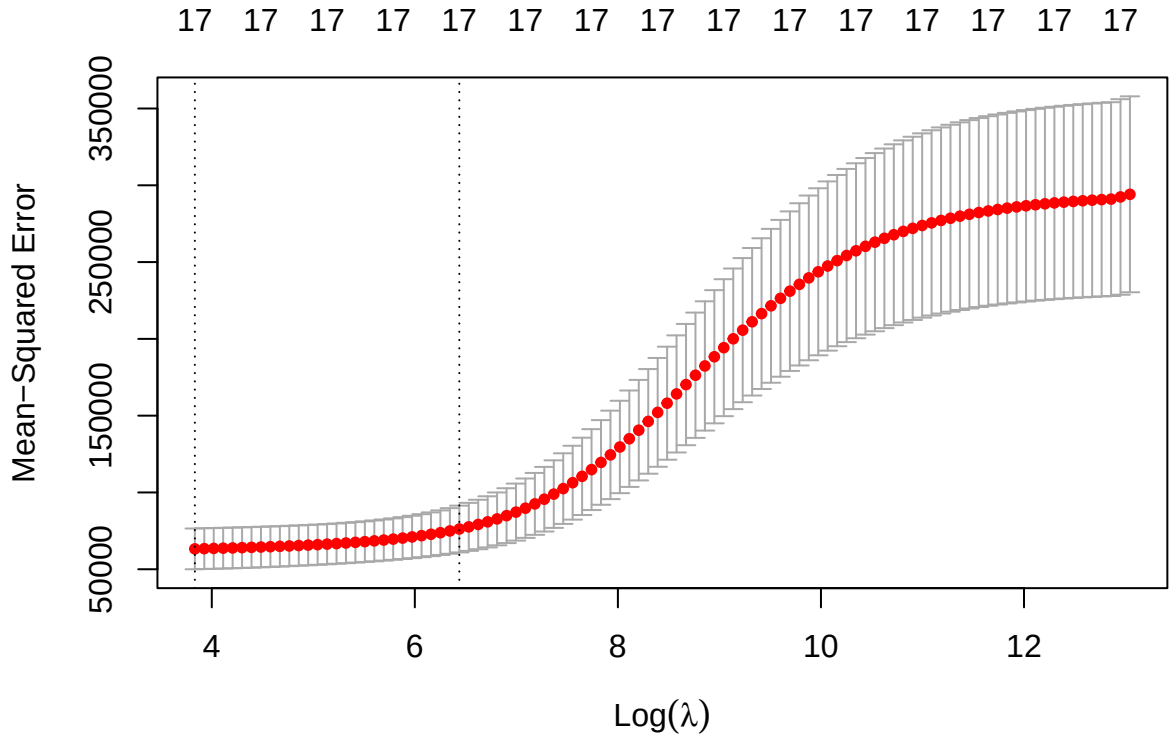
```
## Debt.Equity.Ratio                    -5.523973e+00
## Number.of.Employees                   1.545113e-04
```

##Ridge:

```
ridge_model <- cv.glmnet(X, y, alpha = 0, family = "gaussian")
best_lambda_r <- ridge_model$lambda.min
final_ridge_model <- glmnet(X, y, alpha = 0, family = "gaussian",
    lambda = best_lambda_r)
print(summary(final_ridge_model))
```

```
##           Length Class     Mode
## a0         1     -none-    numeric
## beta      18     dgCMatrix S4
## df         1     -none-    numeric
## dim        2     -none-    numeric
## lambda     1     -none-    numeric
## dev.ratio  1     -none-    numeric
## nulldev    1     -none-    numeric
## npasses    1     -none-    numeric
## jerr       1     -none-    numeric
## offset     1     -none-    logical
## call       6     -none-    call
## nobs       1     -none-    numeric
```

```
plot(ridge_model)
```



```r
lambda_value_r = 0.1
coef_r <- coef(final_ridge_model, s = lambda_value_r)
coef_r
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                                                s1
## (Intercept)                          -7.930208e+01
## (Intercept)                           .
## Revenue                               3.267047e-04
## Gross.Profit                          3.782433e-03
## Net.Income                            1.045018e-02
## Share.Holder.Equity                  -6.905203e-04
## Cash.Flow.from.Operating              1.024461e-03
## Cash.Flow.from.Investing             -1.665572e-03
## Cash.Flow.from.Financial.Activities -5.463816e-03
## Current.Ratio                         3.330747e+01
## ROE                                   2.451588e-01
## ROA                                  -2.485422e+00
## ROI                                   2.006664e-03
## Net.Profit.Margin                    -8.568682e-01
## Free.Cash.Flow.per.Share              6.308430e-01
## Return.on.Tangible.Equity             2.258136e-02
## Inflation.Rate.in.US.                 5.522381e+00
## Debt.Equity.Ratio                    -9.599196e+00
## Number.of.Employees                   1.988910e-04
```

2.2 StepWise:

```r
stepwise_model <- step(lm(formula, data = pdata))
```

```
## Start:  AIC=1757.76
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Operating + Cash.Flow.from.Investing +
##     Cash.Flow.from.Financial.Activities + Current.Ratio + ROE +
##     ROA + ROI + Net.Profit.Margin + Free.Cash.Flow.per.Share +
##     Return.on.Tangible.Equity + Inflation.Rate.in.US. + Debt.Equity.Ratio +
##     Number.of.Employees
##
##                                       Df Sum of Sq     RSS    AIC
## - ROI                                  1        22 7099638 1755.8
## - Inflation.Rate.in.US.                1       247 7099863 1755.8
## - ROE                                  1       293 7099909 1755.8
## - Cash.Flow.from.Investing             1       841 7100457 1755.8
## - Return.on.Tangible.Equity            1       898 7100515 1755.8
## - Cash.Flow.from.Operating             1     32934 7132550 1756.5
## - Debt.Equity.Ratio                    1     40001 7139617 1756.7
## - Net.Profit.Margin                    1     54153 7153770 1757.0
## - Cash.Flow.from.Financial.Activities  1     54973 7154590 1757.0
## <none>                                             7099617 1757.8
```

```
## - Free.Cash.Flow.per.Share             1     92975 7192592 1757.8
## - ROA                                  1    113822 7213439 1758.3
## - Number.of.Employees                  1    187535 7287152 1760.0
## - Revenue                              1    199827 7299443 1760.2
## - Share.Holder.Equity                  1    358033 7457649 1763.7
## - Current.Ratio                        1    584391 7684007 1768.5
## - Gross.Profit                         1    732141 7831757 1771.6
## - Net.Income                           1   1292437 8392054 1782.7
##
## Step:  AIC=1755.76
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Operating + Cash.Flow.from.Investing +
##     Cash.Flow.from.Financial.Activities + Current.Ratio + ROE +
##     ROA + Net.Profit.Margin + Free.Cash.Flow.per.Share + Return.on.Tangible.Equity +
##     Inflation.Rate.in.US. + Debt.Equity.Ratio + Number.of.Employees
##
##                                       Df Sum of Sq     RSS    AIC
## - Inflation.Rate.in.US.                1       246 7099885 1753.8
## - ROE                                  1       292 7099930 1753.8
## - Cash.Flow.from.Investing             1       858 7100496 1753.8
## - Return.on.Tangible.Equity            1       903 7100541 1753.8
## - Cash.Flow.from.Operating             1     32996 7132634 1754.5
## - Debt.Equity.Ratio                    1     39990 7139629 1754.7
## - Cash.Flow.from.Financial.Activities  1     54971 7154609 1755.0
## - Net.Profit.Margin                    1     55345 7154983 1755.0
## <none>                                             7099638 1755.8
## - Free.Cash.Flow.per.Share             1     93594 7193232 1755.9
## - ROA                                  1    115318 7214957 1756.3
## - Number.of.Employees                  1    187617 7287256 1758.0
## - Revenue                              1    200170 7299809 1758.2
## - Share.Holder.Equity                  1    358803 7458441 1761.7
## - Current.Ratio                        1    585296 7684934 1766.5
## - Gross.Profit                         1    732619 7832258 1769.6
## - Net.Income                           1   1292604 8392242 1780.7
##
## Step:  AIC=1753.76
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Operating + Cash.Flow.from.Investing +
##     Cash.Flow.from.Financial.Activities + Current.Ratio + ROE +
##     ROA + Net.Profit.Margin + Free.Cash.Flow.per.Share + Return.on.Tangible.Equity +
##     Debt.Equity.Ratio + Number.of.Employees
##
##                                       Df Sum of Sq     RSS    AIC
## - ROE                                  1       234 7100119 1751.8
## - Cash.Flow.from.Investing             1       878 7100763 1751.8
## - Return.on.Tangible.Equity            1       919 7100804 1751.8
## - Cash.Flow.from.Operating             1     33403 7133288 1752.5
## - Debt.Equity.Ratio                    1     39797 7139681 1752.7
## - Cash.Flow.from.Financial.Activities  1     54920 7154805 1753.0
## - Net.Profit.Margin                    1     56143 7156028 1753.0
## <none>                                             7099885 1753.8
## - Free.Cash.Flow.per.Share             1     93348 7193232 1753.9
## - ROA                                  1    118422 7218307 1754.4
## - Number.of.Employees                  1    188242 7288127 1756.0
```

```
## - Revenue                                1    200023 7299908 1756.2
## - Share.Holder.Equity                    1    370155 7470040 1760.0
## - Current.Ratio                          1    586466 7686351 1764.5
## - Gross.Profit                           1    755959 7855844 1768.0
## - Net.Income                             1   1292617 8392502 1778.7
##
## Step:  AIC=1751.77
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Operating + Cash.Flow.from.Investing +
##     Cash.Flow.from.Financial.Activities + Current.Ratio + ROA +
##     Net.Profit.Margin + Free.Cash.Flow.per.Share + Return.on.Tangible.Equity +
##     Debt.Equity.Ratio + Number.of.Employees
##
##                                     Df Sum of Sq     RSS    AIC
## - Return.on.Tangible.Equity          1       805 7100924 1749.8
## - Cash.Flow.from.Investing           1       851 7100969 1749.8
## - Cash.Flow.from.Operating           1     34485 7134603 1750.5
## - Debt.Equity.Ratio                  1     52753 7152872 1751.0
## - Cash.Flow.from.Financial.Activities  1     54828 7154947 1751.0
## - Net.Profit.Margin                  1     56672 7156791 1751.0
## <none>                                           7100119 1751.8
## - Free.Cash.Flow.per.Share           1     93643 7193762 1751.9
## - ROA                                1    122772 7222891 1752.5
## - Number.of.Employees                1    188114 7288233 1754.0
## - Revenue                            1    215751 7315870 1754.6
## - Share.Holder.Equity                1    393628 7493746 1758.5
## - Current.Ratio                      1    624564 7724683 1763.3
## - Gross.Profit                       1    782887 7883006 1766.6
## - Net.Income                         1   1368346 8468465 1778.1
##
## Step:  AIC=1749.79
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Operating + Cash.Flow.from.Investing +
##     Cash.Flow.from.Financial.Activities + Current.Ratio + ROA +
##     Net.Profit.Margin + Free.Cash.Flow.per.Share + Debt.Equity.Ratio +
##     Number.of.Employees
##
##                                     Df Sum of Sq     RSS    AIC
## - Cash.Flow.from.Investing           1       849 7101773 1747.8
## - Cash.Flow.from.Operating           1     33980 7134904 1748.6
## - Cash.Flow.from.Financial.Activities  1     54875 7155799 1749.0
## - Net.Profit.Margin                  1     57416 7158340 1749.1
## - Debt.Equity.Ratio                  1     57544 7158467 1749.1
## <none>                                           7100924 1749.8
## - Free.Cash.Flow.per.Share           1     92889 7193812 1749.9
## - ROA                                1    122212 7223135 1750.5
## - Number.of.Employees                1    187317 7288240 1752.0
## - Revenue                            1    217293 7318217 1752.6
## - Share.Holder.Equity                1    393472 7494396 1756.5
## - Current.Ratio                      1    623931 7724855 1761.3
## - Gross.Profit                       1    789345 7890268 1764.8
## - Net.Income                         1   1380038 8480962 1776.4
##
## Step:  AIC=1747.81
```

```
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Operating + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + ROA + Net.Profit.Margin + Free.Cash.Flow.per.Share +
##     Debt.Equity.Ratio + Number.of.Employees
##
##                                      Df Sum of Sq     RSS    AIC
## - Cash.Flow.from.Operating            1     34182 7135954 1746.6
## - Net.Profit.Margin                   1     56584 7158357 1747.1
## - Debt.Equity.Ratio                   1     56695 7158467 1747.1
## <none>                                            7101773 1747.8
## - Free.Cash.Flow.per.Share            1     93091 7194864 1747.9
## - ROA                                 1    128295 7230068 1748.7
## - Cash.Flow.from.Financial.Activities 1    159183 7260956 1749.4
## - Number.of.Employees                 1    193703 7295475 1750.1
## - Revenue                             1    220829 7322601 1750.7
## - Share.Holder.Equity                 1    395519 7497291 1754.5
## - Current.Ratio                       1    623681 7725453 1759.4
## - Gross.Profit                        1    806645 7908417 1763.1
## - Net.Income                          1   1647049 8748821 1779.4
##
## Step:  AIC=1746.58
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + ROA + Net.Profit.Margin + Free.Cash.Flow.per.Share +
##     Debt.Equity.Ratio + Number.of.Employees
##
##                                      Df Sum of Sq     RSS    AIC
## - Free.Cash.Flow.per.Share            1     66180 7202134 1746.1
## - Net.Profit.Margin                   1     66768 7202723 1746.1
## - Debt.Equity.Ratio                   1     77829 7213784 1746.3
## <none>                                            7135954 1746.6
## - ROA                                 1    121188 7257142 1747.3
## - Cash.Flow.from.Financial.Activities 1    154824 7290779 1748.0
## - Number.of.Employees                 1    199852 7335806 1749.0
## - Revenue                             1    226341 7362296 1749.6
## - Share.Holder.Equity                 1    417095 7553049 1753.7
## - Current.Ratio                       1    634897 7770852 1758.3
## - Gross.Profit                        1    775723 7911677 1761.2
## - Net.Income                          1   1674855 8810809 1778.5
##
## Step:  AIC=1746.07
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + ROA + Net.Profit.Margin + Debt.Equity.Ratio +
##     Number.of.Employees
##
##                                      Df Sum of Sq     RSS    AIC
## - Debt.Equity.Ratio                   1     84430 7286564 1745.9
## - ROA                                 1     85344 7287478 1746.0
## <none>                                            7202134 1746.1
## - Net.Profit.Margin                   1    107188 7309322 1746.4
## - Cash.Flow.from.Financial.Activities 1    179248 7381382 1748.0
## - Number.of.Employees                 1    183276 7385410 1748.1
## - Revenue                             1    239985 7442119 1749.3
```

```
## - Share.Holder.Equity                   1    405545 7607679 1752.9
## - Current.Ratio                         1    620008 7822142 1757.4
## - Gross.Profit                          1    842186 8044320 1761.9
## - Net.Income                            1   1609747 8811881 1776.5
##
## Step:  AIC=1745.94
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + ROA + Net.Profit.Margin + Number.of.Employees
##
##                                       Df Sum of Sq     RSS    AIC
## - ROA                                  1     74045 7360609 1745.6
## <none>                                             7286564 1745.9
## - Net.Profit.Margin                    1     97038 7383602 1746.1
## - Cash.Flow.from.Financial.Activities  1    177975 7464539 1747.8
## - Revenue                              1    200582 7487146 1748.3
## - Number.of.Employees                  1    234549 7521114 1749.0
## - Share.Holder.Equity                  1    462693 7749258 1753.8
## - Current.Ratio                        1    667437 7954002 1758.0
## - Gross.Profit                         1    759873 8046437 1759.9
## - Net.Income                           1   1735793 9022357 1778.3
##
## Step:  AIC=1745.57
## Market.Cap.in.B.USD. ~ Revenue + Gross.Profit + Net.Income +
##     Share.Holder.Equity + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + Net.Profit.Margin + Number.of.Employees
##
##                                       Df Sum of Sq     RSS    AIC
## <none>                                             7360609 1745.6
## - Cash.Flow.from.Financial.Activities  1    178701 7539311 1747.4
## - Number.of.Employees                  1    225255 7585864 1748.4
## - Revenue                              1    231030 7591640 1748.5
## - Net.Profit.Margin                    1    368220 7728829 1751.4
## - Share.Holder.Equity                  1    409505 7770114 1752.3
## - Current.Ratio                        1    594154 7954763 1756.1
## - Gross.Profit                         1    804534 8165144 1760.3
## - Net.Income                           1   1661815 9022424 1776.3
```

```
print(stepwise_model)
```

```
##
## Call:
## lm(formula = Market.Cap.in.B.USD. ~ Revenue + Gross.Profit +
##     Net.Income + Share.Holder.Equity + Cash.Flow.from.Financial.Activities +
##     Current.Ratio + Net.Profit.Margin + Number.of.Employees,
##     data = pdata)
##
## Coefficients:
##                          (Intercept)                               Revenue
##                           -4.851e+01                            -1.666e-03
##                         Gross.Profit                            Net.Income
##                            7.633e-03                             1.823e-02
##                  Share.Holder.Equity  Cash.Flow.from.Financial.Activities
```

```
##                           -1.571e-03                           -3.267e-03
##                         Current.Ratio                    Net.Profit.Margin
##                             4.323e+01                           -5.752e+00
##                   Number.of.Employees
##                             3.686e-04
```

From the above analysis we find that the best model has :

```
names(coef(stepwise_model))[2:9]
```

```
## [1] "Revenue"                              "Gross.Profit"
## [3] "Net.Income"                           "Share.Holder.Equity"
## [5] "Cash.Flow.from.Financial.Activities" "Current.Ratio"
## [7] "Net.Profit.Margin"                    "Number.of.Employees"
```

Buliding a fixed effects model on this:

```
new_formula_fe <- as.formula(paste("Market.Cap.in.B.USD. ~",
    paste(names(coef(stepwise_model))[2:9], collapse = " + ")))
fe_constrained <- plm(new_formula_fe, data = pdata, model = "within")
summary(fe_constrained)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = new_formula_fe, data = pdata, model = "within")
##
## Unbalanced Panel: n = 12, T = 9-15, N = 161
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -610.4545  -72.8252   -1.5654   61.1696  866.8318
##
## Coefficients:
##                                      Estimate  Std. Error t-value  Pr(>|t|)
## Revenue                           -0.00363174  0.00209584 -1.7328   0.08531
## Gross.Profit                       0.00947308  0.00439632  2.1548   0.03288
## Net.Income                         0.02232434  0.00361231  6.1801 6.492e-09
## Share.Holder.Equity               -0.00149108  0.00087021 -1.7135   0.08882
## Cash.Flow.from.Financial.Activities -0.00370252  0.00193650 -1.9120   0.05791
## Current.Ratio                     21.60976435 22.08436261  0.9785   0.32950
## Net.Profit.Margin                 -5.40869746  2.33398315 -2.3174   0.02192
## Number.of.Employees                0.00058720  0.00028999  2.0249   0.04477
##
## Revenue                           .
## Gross.Profit                      *
## Net.Income                        ***
## Share.Holder.Equity               .
## Cash.Flow.from.Financial.Activities .
## Current.Ratio
## Net.Profit.Margin                 *
## Number.of.Employees               *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    26960000
## Residual Sum of Squares: 6091500
## R-Squared:      0.77405
## Adj. R-Squared: 0.74361
## F-statistic: 60.3802 on 8 and 141 DF, p-value: < 2.22e-16
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
# final_fe_model<-stepAIC(initial_fe_model,direction='backward')
# normal stepAIC function won't work for panel data
```

```
# install.packages('pglm')
library(pglm)
```

```
## Warning: package 'pglm' was built under R version 4.2.3

## Loading required package: maxLik

## Warning: package 'maxLik' was built under R version 4.2.3

## Loading required package: miscTools

## Warning: package 'miscTools' was built under R version 4.2.3

##
## Please cite the 'maxLik' package as:
## Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood estimation in R. C
##
## If you have questions, suggestions, or comments regarding the 'maxLik' package, please use a forum o
## https://r-forge.r-project.org/projects/maxlik/
```

```
# null_model <- pglm(pdata$Market.Cap.in.B.USD.~
# 1,data=pdata,family=gaussian) final_fwd_fe_model <-
# stepAIC(initial_fe_model,direction = 'forward')
```

2.3 Random Forest Method

using tree-based decision making through random forest estimation

The following variables are being selcted.

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.2.3
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
rf_model <- randomForest(formula, data = pdata)
rf_var_importance <- importance(rf_model)
rf_threshold <- 2500000  # Adjust the threshold as needed
rf_selected_variables <- rownames(rf_var_importance)[rf_var_importance[,
    "IncNodePurity"] > rf_threshold]
rf_selected_variables
```

```
## [1] "Revenue"                                "Gross.Profit"
## [3] "Net.Income"                             "Cash.Flow.from.Operating"
## [5] "Cash.Flow.from.Financial.Activities"
```

2.4 Information Criteria:

```
models <- list()
for (i in 1:100) {
    predictors <- sample(dep, size = sample(1:5, 1))
    formula_IC <- as.formula(paste("Market.Cap.in.B.USD. ~",
        paste(predictors, collapse = "+")))
    model <- pglm(formula_IC, model = "within", data = pdata,
        family = gaussian)
    models[[i]] <- model
}
aic_values <- numeric(100)
bic_values <- numeric(100)

for (i in 1:100) {
    aic_values[i] <- AIC(models[[i]])
    bic_values[i] <- BIC(models[[i]])
}
```

```
best_model_index_aic <- which.min(aic_values)
best_model_index_bic <- which.min(bic_values)

best_model_aic <- models[[best_model_index_aic]]
# bic doesn't work for panel data and hence NULL is
# repturned best_model_bic<-models[[best_model_index_bic]]

summary(best_model_aic)
```

```
## ---------------------------------------------
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 150 iterations
## Return code 4: Iteration limit exceeded (iterlim)
## Log-Likelihood: -1097.526
## 7  free parameters
## Estimates:
##                       Estimate Std. error t value  Pr(> t)
## (Intercept)         -2.355e+01  3.992e+01  -0.590   0.5552
## ROI                 -2.072e-02  1.802e-01  -0.115   0.9084
## Share.Holder.Equity -1.783e-03  6.275e-04  -2.841   0.0045 **
## Net.Income           1.764e-02  1.789e-03   9.865  < 2e-16 ***
## Gross.Profit         6.555e-03  9.083e-04   7.217 5.32e-13 ***
## sd.id                9.451e+01  4.927e-01 191.831  < 2e-16 ***
## sd.idios             2.110e+02  1.213e+01  17.393  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---------------------------------------------
```

```
# summary(best_model_bic)
```

Therefore the variables selected are:

```
coef(best_model_aic)
```

```
##         (Intercept)                 ROI Share.Holder.Equity          Net.Income
##       -23.553795707        -0.020724677        -0.001782593         0.017644624
##        Gross.Profit               sd.id            sd.idios
##         0.006555258        94.514984473       211.049541894
```

2.5 Out Of Sample Methodology

(Cross Validation)

```
# library(caret) library(glmnet)
# pdata$Market.Cap.in.B.USD.<-
# as.factor(pdata$Market.Cap.in.B.USD.)  set.seed(123)
# train_index<-createDataPartition(data$Market.Cap.in.B.USD.,p=0.8,list=FALSE)
# training_data<-pdata[train_index,]
# testing_data<-pdata[-train_index,] model_results<-list()
```

```
# num_folds<-5 for(i in 1:100){
# predictors<-sample(names(training_data)[2:201],size=sample(1:5,1))
# training_data$intercept<-1
# x<-as.matrix(training_data[,c('intercept'),drop=FALSE])


# training data is NULL since CV doesn't work for panel
# data:



# formula<-as.formula(paste('Market.Cap.in.B.USD.
# ~',paste(colnames(x)[-1],collapse='+')))
# outcome_class<-'twoClass' outcome_levels
# <-levels(training_data$Market.Cap.in.B.USD. )
#
↪  model<-cv.glmnet(x,as.numeric(training_data$Market.Cap.in.B.USD.),family='gaussian',type.measure
# = 'class',nfolds=num_folds)
# model_results[[i]]<-model_results } best_model_cv <-NULL
# best_auc<-0 for(i in 1:100){ model<-model_results[[i]]
# perf<-max(model$cvm) if(perf>best_auc){ best_auc<-perf
# best_model<-model } } print(best_model)
```

#Interpreting the coefficients

We will take the results of the random Forest methodology: therefor the coefficients are:

```
rf_selected_variables
```

```
## [1] "Revenue"                                "Gross.Profit"
## [3] "Net.Income"                             "Cash.Flow.from.Operating"
## [5] "Cash.Flow.from.Financial.Activities"
```

The Revenue is the total amount procured by the company. Gross Profit is the accounting profit, i.e inflow - outflow Net Income is the inward CASH flow(not the inventory) Cash Flow from Operating and Financial Activities ~ self explanatory

With these variables we can predict the market cap of the given company

We know for our Fixed Effects model, the generalised formula is :

$$\mathbf{Y}_{it} = \beta_{\mathbf{0}} + \mathbf{X_{it}}\beta_{\mathbf{i}} + \mathbf{c_i} + \epsilon_{\mathbf{i}} \text{for ith compant at t time instance.} \mathbf{c_i} : \text{group specific intercept}$$

Now the finalised model:

```
new_formula_rf <- as.formula(paste("Market.Cap.in.B.USD. ~",
    paste(rf_selected_variables, collapse = " + ")))
new_model_rf <- plm(new_formula_rf, data = pdata, model = "within")
```

class c_i terms:

```
fe_coeff <- fixef(new_model_rf)
fe_coeff
```

```
##     AAPL      AIG     AMZN      BCS     GOOG     INTC      MCD     MSFT
## -319.829  -93.320  107.469 -212.460 -121.852 -257.216  -22.742  -79.946
##     NVDA      PCG     PYPL    SHLDQ
##  126.583  -25.337   45.309  -16.809
```

3 Checking the assumptions:

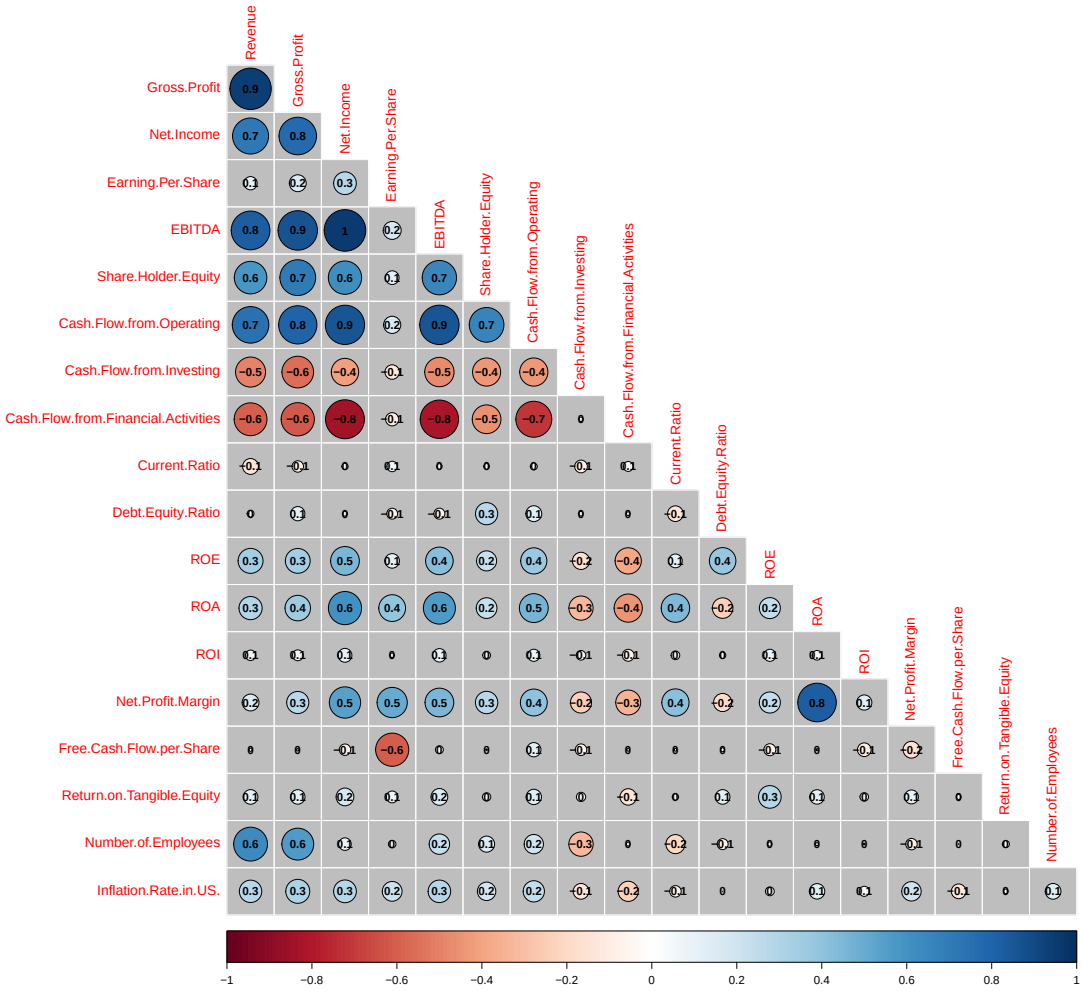
##homoscedasticity The correlation matrix:

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.2.3
```

```
## corrplot 0.92 loaded
```

```
corrplot(corr = cor(pdata[dep]), addCoef.col = "black", number.cex = 0.8,
    number.digits = 1, diag = FALSE, bg = "grey", outline = "black",
    addgrid.col = "white", mar = c(1, 1, 1, 1), type = "lower")
```




```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.2.3
```

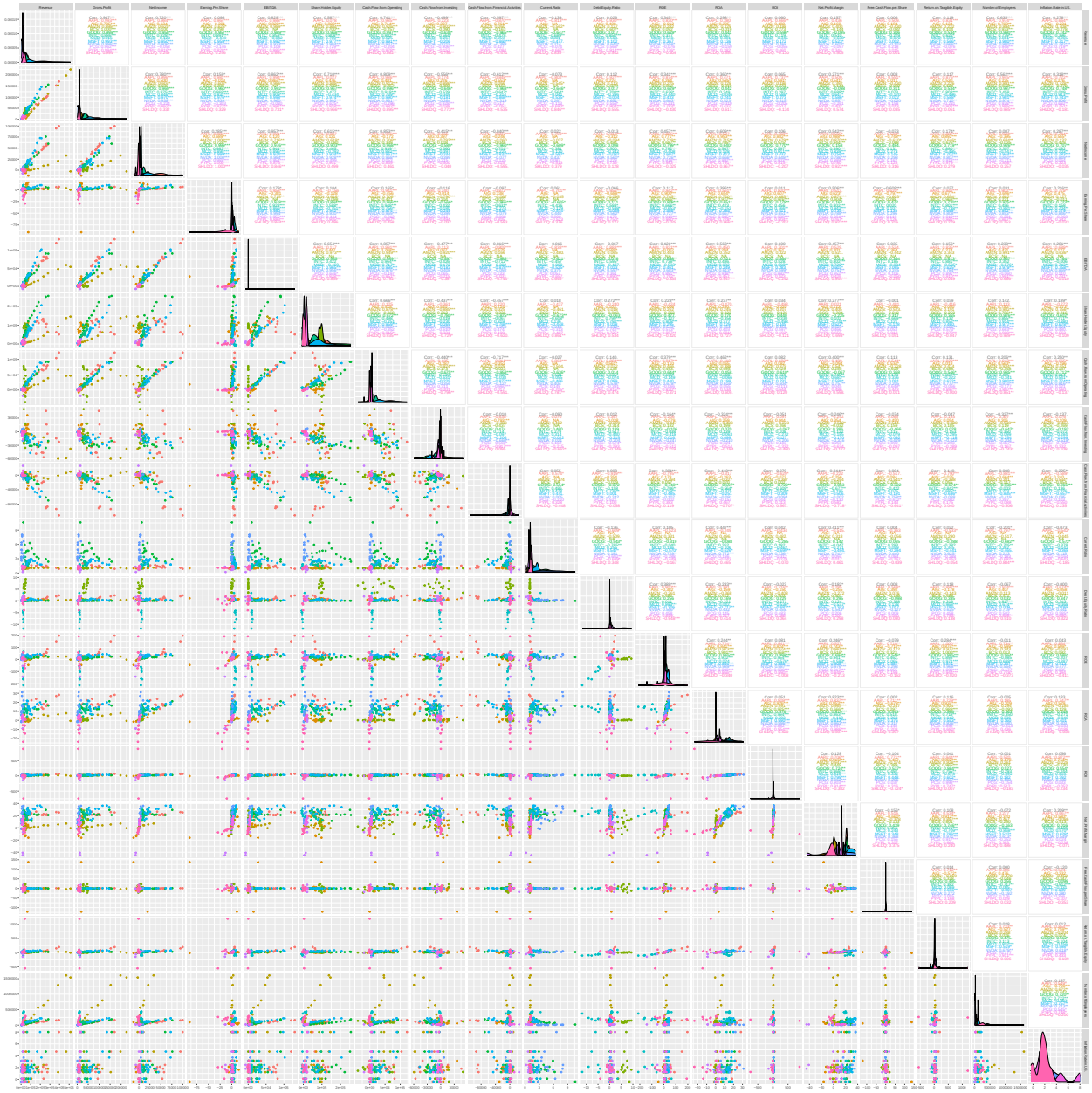
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
GGally::ggpairs(pdata[dep], ggplot2::aes(colour = pdata$Company))
```

```
## Warning in cor(x, y): the standard deviation is zero
```

```
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
```

```
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
```



TO check for homoscedasticity of variables

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.2.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.2.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
pdata$Market.Cap.in.B.USD. <- as.numeric(as.character(pdata$Market.Cap.in.B.USD.))
bptest(formula, data = pdata, studentize = F)
```

```
##
##  Breusch-Pagan test
##
## data:  formula
## BP = 184.94, df = 17, p-value < 2.2e-16
```

Here, the obtained p-value : <0.05. Therefore we can conclude that H0: Homoscedasticity exists, is false. Therefore the data is heteroscedastic in nature.

Checking for our constrained model that we got from variable selection: **Using random Forest estimates:**

```
bptest(new_formula_rf, data = pdata)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  new_formula_rf
## BP = 48.168, df = 5, p-value = 3.282e-09
```

Checking AIC selected variables:

```
new_formula_aic <- as.formula(paste("Market.Cap.in.B.USD. ~",
    paste(names(coef(best_model_aic))[2:5], collapse = " + ")))
new_fe_model_aic <- plm(new_formula_aic, data = pdata, model = "within")
bptest(new_formula_aic, data = pdata)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  new_formula_aic
## BP = 46.712, df = 4, p-value = 1.75e-09
```

To correct heteroskedasticity, we take the squareroot of the independent variable:

```
pdata_sq <- pdata
pdata_sq$Market.Cap.in.B.USD. <- sqrt(pdata$Market.Cap.in.B.USD.)
bptest(new_formula_aic, data = pdata_sq)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  new_formula_aic
## BP = 7.5661, df = 4, p-value = 0.1088
```

```
bptest(new_formula_rf, data = pdata_sq)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  new_formula_rf
## BP = 6.6944, df = 5, p-value = 0.2444
```

->p-value>0.05 for both model variables, therefore homoscedasticity is assumed in both models.

Now modifying the formula:

```
new_formula_aic <- as.formula(paste("sqrt(pdata$Market.Cap.in.B.USD.) ~",
    paste(names(coef(best_model_aic))[2:5], collapse = " + ")))
new_formula_rf <- as.formula(paste("sqrt(pdata$Market.Cap.in.B.USD.) ~",
    paste(rf_selected_variables, collapse = " + ")))
```

```
bptest(formula, data = pdata)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  formula
## BP = 50.673, df = 17, p-value = 3.319e-05
```

Checking for outliers:

```
new_fe_model <- plm(new_formula_rf, data = pdata, model = "within",
    effect = "individual")
residuals <- residuals(new_fe_model)
standardized_residuals <- residuals/sqrt(var(residuals))
standardized_residuals
```

```
##    AAPL-2009    AAPL-2010    AAPL-2011    AAPL-2012    AAPL-2013    AAPL-2014
##  0.701762552  0.998104640  0.335831493 -0.567655954 -0.260615797 -0.001320181
##    AAPL-2015    AAPL-2016    AAPL-2017    AAPL-2018    AAPL-2019    AAPL-2020
## -1.624361603 -0.777975460  0.142928043 -1.953000467  0.567140244  3.135214673
##    AAPL-2021    AAPL-2022     AIG-2009     AIG-2010     AIG-2011     AIG-2012
##  1.202786497 -1.898838680 -0.585714218 -0.972914544 -1.253769646 -0.173768665
##     AIG-2013     AIG-2014     AIG-2015     AIG-2016     AIG-2017     AIG-2018
## -0.058597401  0.094340708  0.691854097  0.914447873  0.947757582  0.304036308
##     AIG-2019     AIG-2020     AIG-2021     AIG-2022    AMZN-2009    AMZN-2010
##  0.207601101  0.613243608 -0.267141245 -0.461375559 -0.999508074 -0.731185431
##    AMZN-2011    AMZN-2012    AMZN-2013    AMZN-2014    AMZN-2015    AMZN-2016
## -0.829267688 -0.451678768  0.071191126 -0.424098136  0.649694534  0.450584580
##    AMZN-2017    AMZN-2018    AMZN-2019    AMZN-2020    AMZN-2021    AMZN-2022
##  1.192890984  0.672992060  0.726086987  1.668659378 -0.103309555 -1.893051998
##     BCS-2009     BCS-2010     BCS-2011     BCS-2012     BCS-2013     BCS-2014
## -0.569617181 -0.523152778 -0.832550021  0.131803389  0.221458077  0.305608611
##     BCS-2015     BCS-2016     BCS-2017     BCS-2018     BCS-2019     BCS-2020
```

```
##  0.306201015  0.312472900  0.629788015  0.019245546  0.185269067  0.158875164
##     BCS-2021     BCS-2022    GOOG-2009    GOOG-2010    GOOG-2011    GOOG-2012
## -0.128139624 -0.217262182  0.098862098 -0.174869735 -0.266412661 -0.249676451
##    GOOG-2013    GOOG-2014    GOOG-2015    GOOG-2016    GOOG-2017    GOOG-2018
##  0.510702013  0.106785429  0.804654950  0.367929078  1.422577514 -0.009900942
##    GOOG-2019    GOOG-2020    GOOG-2021    GOOG-2022    INTC-2009    INTC-2010
##  0.201466910  0.642856473 -0.750375901 -2.704598777  0.543626975 -0.082812417
##    INTC-2011    INTC-2012    INTC-2013    INTC-2014    INTC-2015    INTC-2016
## -0.303123398 -0.349705205  0.053273535  0.226110944  0.264549108  0.310084145
##    INTC-2017    INTC-2018    INTC-2019    INTC-2020    INTC-2021    INTC-2022
##  0.640583466 -0.228008924  0.204682489 -1.283318966 -0.107806220  0.111864468
##     MCD-2009     MCD-2010     MCD-2011     MCD-2012     MCD-2013     MCD-2014
## -0.471749972 -0.231136266 -0.480868480 -0.372979904 -0.453179857 -0.180662291
##     MCD-2015     MCD-2016     MCD-2017     MCD-2018     MCD-2019     MCD-2020
## -0.198057973  0.105825142  0.106824304  0.179419113  0.281993852  0.797101437
##     MCD-2021     MCD-2022    MSFT-2009    MSFT-2010    MSFT-2011    MSFT-2012
##  0.464173128  0.453297765 -0.769385338 -1.413587354 -1.918185786 -1.587404761
##    MSFT-2013    MSFT-2014    MSFT-2015    MSFT-2016    MSFT-2017    MSFT-2018
## -1.211825073 -0.820558237  0.069493641 -0.084433594  0.579955016  1.018854781
##    MSFT-2019    MSFT-2020    MSFT-2021    MSFT-2022    MSFT-2023    NVDA-2009
##  1.105719413  1.917990051  2.703098772 -0.612433233  1.022701701 -1.569379496
##    NVDA-2010    NVDA-2011    NVDA-2012    NVDA-2013    NVDA-2014    NVDA-2015
## -1.621757165 -1.666480353 -1.739433352 -1.682482433 -1.602806072 -1.404380734
##    NVDA-2016    NVDA-2017    NVDA-2018    NVDA-2019    NVDA-2020    NVDA-2021
## -0.561377789  0.164492072 -0.440336290  0.203176735  1.806558773  3.944182002
##    NVDA-2022    NVDA-2023     PCG-2009     PCG-2010     PCG-2011     PCG-2012
##  1.363909721  4.806114381 -0.082739656 -0.145783221 -0.148376836 -0.137722862
##     PCG-2013     PCG-2014     PCG-2015     PCG-2016     PCG-2017     PCG-2018
##  0.085506094  0.038185807  0.165720007 -0.088107765 -0.430368058 -0.221582429
##     PCG-2019     PCG-2020     PCG-2021     PCG-2022    PYPL-2014    PYPL-2015
##  0.425887378  0.216769392  0.117674231  0.204937917 -2.168637004 -0.532035736
##    PYPL-2016    PYPL-2017    PYPL-2018    PYPL-2019    PYPL-2020    PYPL-2021
## -0.511832944  0.073419076  0.131283046  0.457561212  1.692691140  1.115800641
##    PYPL-2022   SHLDQ-2009   SHLDQ-2010   SHLDQ-2011   SHLDQ-2012   SHLDQ-2013
## -0.258249430  0.136522694  0.110837447 -0.124428230  0.157941709  0.062964430
##   SHLDQ-2014   SHLDQ-2015   SHLDQ-2016   SHLDQ-2017   SHLDQ-2018
##  0.061830420  0.023702907 -0.079217785 -0.085581914 -0.264571680
```

```
new_fe_model_aic <- plm(new_formula_aic, data = pdata, model = "within",
    effect = "individual")
residuals <- residuals(new_fe_model_aic)
standardized_residuals <- residuals/sqrt(var(residuals))
standardized_residuals
```

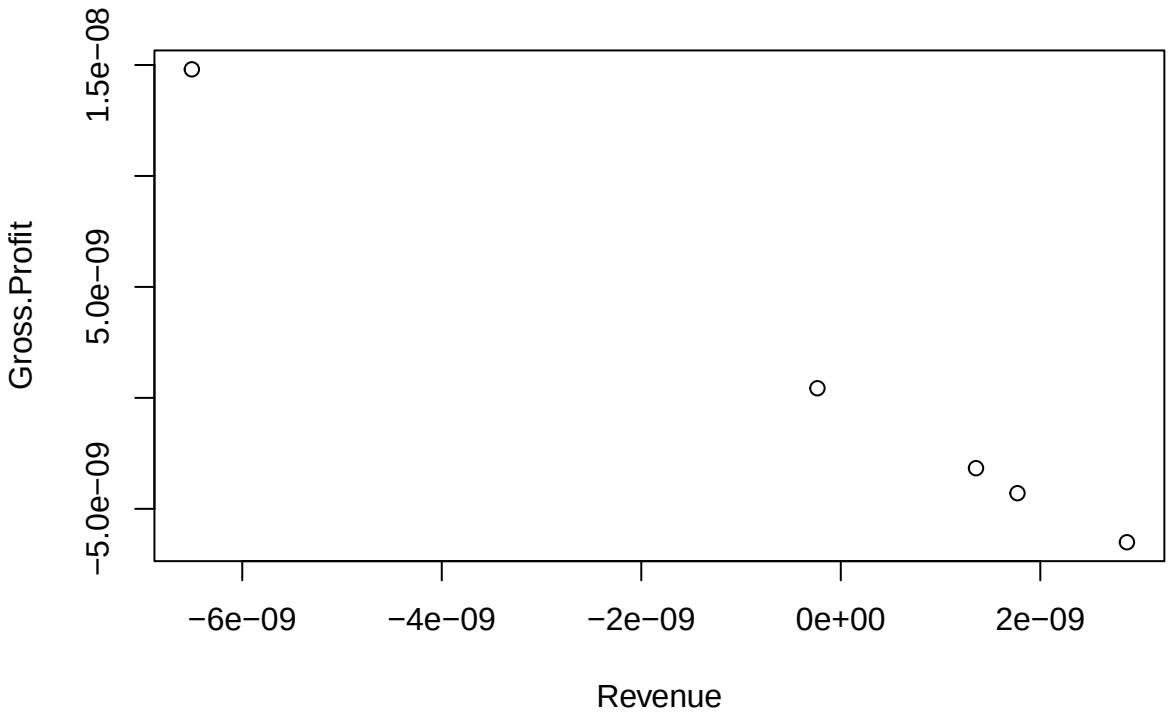
```
##   AAPL-2009   AAPL-2010   AAPL-2011   AAPL-2012   AAPL-2013   AAPL-2014
##  0.69686826  0.95604640  0.22224717 -0.74800415 -0.28205708  0.09399341
##   AAPL-2015   AAPL-2016   AAPL-2017   AAPL-2018   AAPL-2019   AAPL-2020
## -1.80084131 -0.84852849  0.02731491 -1.68099614  0.86814667  3.33840635
##   AAPL-2021   AAPL-2022    AIG-2009    AIG-2010    AIG-2011    AIG-2012
##  1.09543918 -1.93803519 -0.28497640 -0.92143929 -1.16817903 -0.07996609
##    AIG-2013    AIG-2014    AIG-2015    AIG-2016    AIG-2017    AIG-2018
## -0.07612309  0.16272229  0.71992083  0.91709160  0.96560300  0.20824445
##    AIG-2019    AIG-2020    AIG-2021    AIG-2022   AMZN-2009   AMZN-2010
```

```
##  0.04961454  0.55631941 -0.38944275 -0.65938948 -0.93870466 -0.67793243
##   AMZN-2011   AMZN-2012   AMZN-2013   AMZN-2014   AMZN-2015   AMZN-2016
## -0.77010300 -0.41360865  0.12373005 -0.40834159  0.71619524  0.49904866
##   AMZN-2017   AMZN-2018   AMZN-2019   AMZN-2020   AMZN-2021   AMZN-2022
##  1.12389410  0.68045400  0.74757229  1.54857387 -0.36102018 -1.86975770
##    BCS-2009    BCS-2010    BCS-2011    BCS-2012    BCS-2013    BCS-2014
## -0.58699705 -0.57290882 -0.78110370  0.18079345  0.13913984  0.36414656
##    BCS-2015    BCS-2016    BCS-2017    BCS-2018    BCS-2019    BCS-2020
##  0.33964139  0.32659673  0.66984802  0.07514714  0.13298522  0.14301512
##    BCS-2021    BCS-2022   GOOG-2009   GOOG-2010   GOOG-2011   GOOG-2012
## -0.16482446 -0.26547945  0.02278193 -0.27592485 -0.34630065 -0.32388407
##   GOOG-2013   GOOG-2014   GOOG-2015   GOOG-2016   GOOG-2017   GOOG-2018
##  0.45320647  0.06863086  0.78776230  0.37612787  1.49837584 -0.02032180
##   GOOG-2019   GOOG-2020   GOOG-2021   GOOG-2022   INTC-2009   INTC-2010
##  0.26219220  0.68739202 -0.72023199 -2.46980614  0.55186114 -0.11073829
##   INTC-2011   INTC-2012   INTC-2013   INTC-2014   INTC-2015   INTC-2016
## -0.30306535 -0.39899872  0.05848335  0.26814415  0.19871745  0.31910541
##   INTC-2017   INTC-2018   INTC-2019   INTC-2020   INTC-2021   INTC-2022
##  0.67799702 -0.20371118  0.22634892 -1.28979695 -0.13944743  0.14510049
##    MCD-2009    MCD-2010    MCD-2011    MCD-2012    MCD-2013    MCD-2014
## -0.44937928 -0.21764201 -0.46669948 -0.36294949 -0.44158131 -0.16285462
##    MCD-2015    MCD-2016    MCD-2017    MCD-2018    MCD-2019    MCD-2020
## -0.22725848  0.15164371  0.10137741  0.16876326  0.26013365  0.76509803
##    MCD-2021    MCD-2022   MSFT-2009   MSFT-2010   MSFT-2011   MSFT-2012
##  0.43804164  0.44330698 -0.79398427 -1.41936335 -1.98468749 -1.57936955
##   MSFT-2013   MSFT-2014   MSFT-2015   MSFT-2016   MSFT-2017   MSFT-2018
## -1.24024065 -0.83339660  0.12978220 -0.11551816  0.39927738  1.22034934
##   MSFT-2019   MSFT-2020   MSFT-2021   MSFT-2022   MSFT-2023   NVDA-2009
##  1.15896310  2.01185038  2.69301055 -0.60971736  0.96304447 -1.55805233
##   NVDA-2010   NVDA-2011   NVDA-2012   NVDA-2013   NVDA-2014   NVDA-2015
## -1.61264419 -1.65965259 -1.73406029 -1.67445910 -1.59781416 -1.39234293
##   NVDA-2016   NVDA-2017   NVDA-2018   NVDA-2019   NVDA-2020   NVDA-2021
## -0.55231198  0.15684943 -0.43445937  0.20246856  1.80191796  3.89025858
##   NVDA-2022   NVDA-2023    PCG-2009    PCG-2010    PCG-2011    PCG-2012
##  1.29455162  4.86975078 -0.07797098 -0.13710981 -0.13722847 -0.11751202
##    PCG-2013    PCG-2014    PCG-2015    PCG-2016    PCG-2017    PCG-2018
##  0.09105989  0.04528887  0.17443881 -0.07986380 -0.41132999 -0.16664502
##    PCG-2019    PCG-2020    PCG-2021    PCG-2022   PYPL-2014   PYPL-2015
##  0.48004741  0.05371296  0.11551252  0.16759963 -2.14504328 -0.53745038
##   PYPL-2016   PYPL-2017   PYPL-2018   PYPL-2019   PYPL-2020   PYPL-2021
## -0.49720919  0.06826814  0.16355742  0.44494119  1.60283903  1.12715436
##   PYPL-2022  SHLDQ-2009  SHLDQ-2010  SHLDQ-2011  SHLDQ-2012  SHLDQ-2013
## -0.22705729  0.13817549  0.11440651 -0.12744487  0.17301736  0.05948827
##  SHLDQ-2014  SHLDQ-2015  SHLDQ-2016  SHLDQ-2017  SHLDQ-2018
##  0.05491006  0.01724896 -0.12937353 -0.02938893 -0.27103932
```

3.1 Multicolinearity test:

Note: VIF cannot be used for panel data also for panel data multicollinearity test isn't relevant.

```
vcov_fe <- vcovHC(new_fe_model)
plot(vcov_fe)
```



3.2 Autocorrelation:

Wooldridge test:

```
pbgtest(new_fe_model)
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  new_formula_rf
## chisq = 56.902, df = 9, p-value = 5.277e-09
## alternative hypothesis: serial correlation in idiosyncratic errors
```

Since p-value <5% there is either autocorrelation or serial correlation in error term.

```
pbgtest(new_fe_model_aic)
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  new_formula_aic
## chisq = 59.352, df = 9, p-value = 1.787e-09
## alternative hypothesis: serial correlation in idiosyncratic errors
```

Since p-value <5% there is either autocorrelation or serial correlation in error term. But the p-value is higher for AIC selected variables.

Durbin-Watson Test:

```
pdwtest(new_formula_rf, data = pdata, model = "within")
```

```
##
##  Durbin-Watson test for serial correlation in panel models
##
## data:  new_formula_rf
## DW = 1.0854, p-value = 1.497e-09
## alternative hypothesis: serial correlation in idiosyncratic errors
```

Since p-value <5% there is autocorrelation in error term.

```
pdwtest(new_formula_aic, data = pdata, model = "within")
```

```
##
##  Durbin-Watson test for serial correlation in panel models
##
## data:  new_formula_aic
## DW = 1.0868, p-value = 2.375e-09
## alternative hypothesis: serial correlation in idiosyncratic errors
```

p-value>1% so we can assume no autocorrelation. Using a lagged model:

```
pdata$lagy <- lag(pdata$Market.Cap.in.B.USD.)
fixed_effects_model_with_lag_rf <- plm(sqrt(pdata$Market.Cap.in.B.USD.) ~
    Revenue + Gross.Profit + Net.Income + Cash.Flow.from.Operating +
        Cash.Flow.from.Financial.Activities + sqrt(lagy) + factor(Company),
    data = pdata, model = "within")
```

Now testing this model:

```
pdwtest(sqrt(pdata$Market.Cap.in.B.USD.) ~ Revenue + Gross.Profit +
    Net.Income + Cash.Flow.from.Operating + Cash.Flow.from.Financial.Activities +
    sqrt(lagy) + factor(Company), data = pdata, model = "within")
```

```
##
##  Durbin-Watson test for serial correlation in panel models
##
## data:  sqrt(pdata$Market.Cap.in.B.USD.) ~ Revenue + Gross.Profit + Net.Income +  ...
## DW = 1.2502, p-value = 5.563e-07
## alternative hypothesis: serial correlation in idiosyncratic errors
```

This did increase the p-value significantly.

For the AIC selected Variables

```
fixed_effects_model_with_lag_aic <- plm(sqrt(pdata$Market.Cap.in.B.USD.) ~
    Gross.Profit + Net.Income + Share.Holder.Equity + Earning.Per.Share +
        Revenue + sqrt(lagy) + factor(Company), data = pdata,
    model = "within")
pdwtest(sqrt(pdata$Market.Cap.in.B.USD.) ~ Gross.Profit + Net.Income +
    Share.Holder.Equity + Earning.Per.Share + Revenue + sqrt(lagy) +
    factor(Company), data = pdata, model = "within")
```

```
##
##  Durbin-Watson test for serial correlation in panel models
##
## data:  sqrt(pdata$Market.Cap.in.B.USD.) ~ Gross.Profit + Net.Income +  ...
## DW = 1.3245, p-value = 4.742e-06
## alternative hypothesis: serial correlation in idiosyncratic errors
```

##Normality of Error Term: Shapiro-Wilk Test:

```
residuals_panel <- residuals(new_fe_model)
shapiro.test(residuals_panel)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals_panel
## W = 0.90938, p-value = 1.918e-08
```

Since p-value is < 0.01, we accept null hyp. that the residuals are not normally distributed.

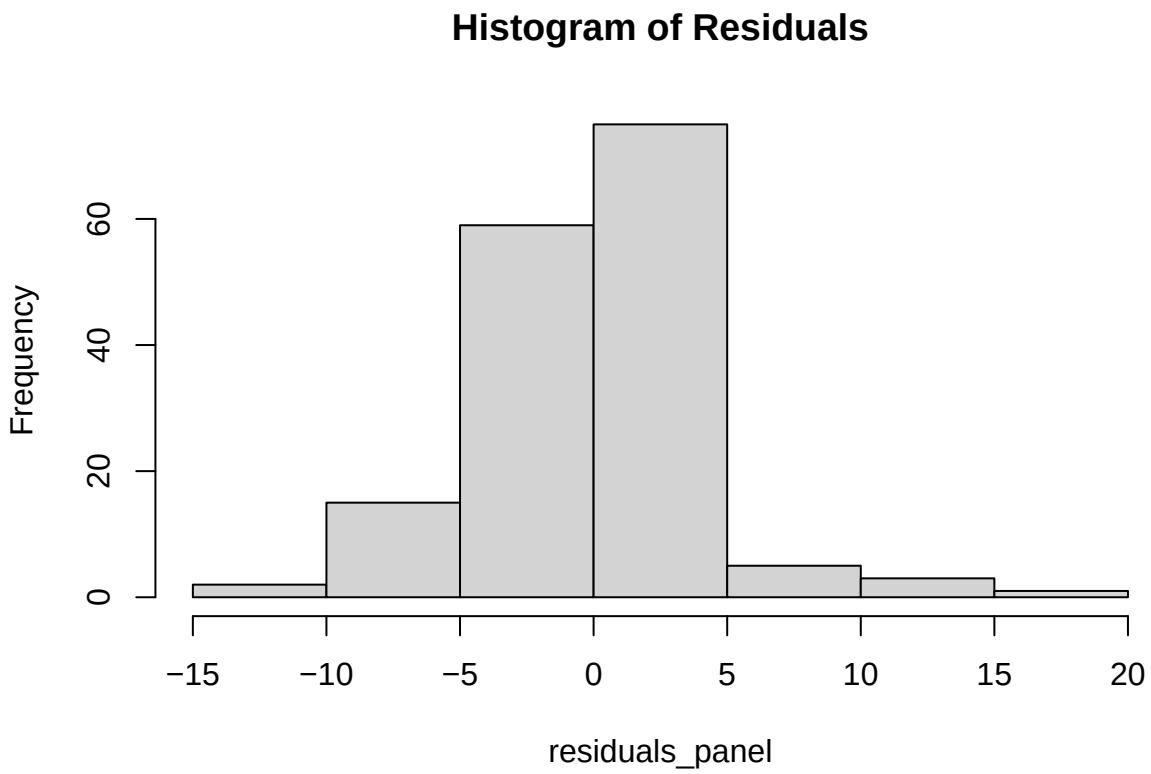
Checking our lagged model:

```
residuals_panel <- residuals(fixed_effects_model_with_lag_aic)
shapiro.test(residuals_panel)
```

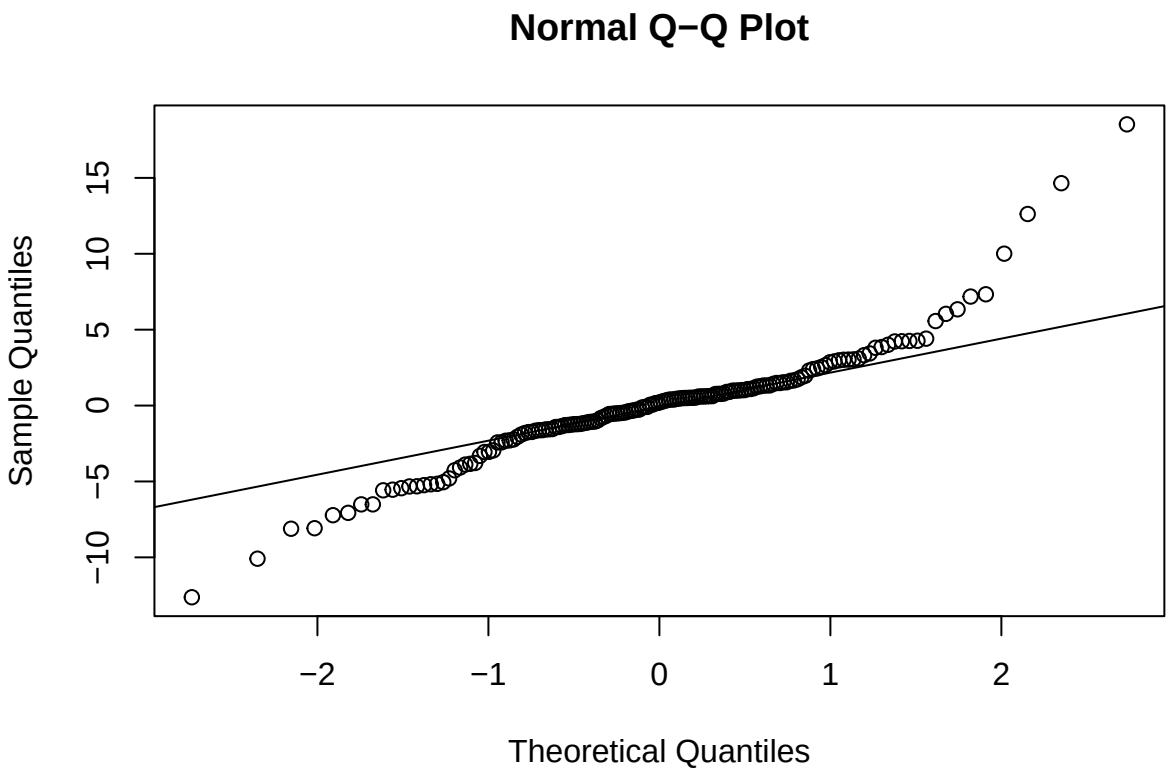
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals_panel
## W = 0.91342, p-value = 3.714e-08
```

P-value is increased but not significantly

```
hist(residuals_panel, main = "Histogram of Residuals")
```




```
qqnorm(residuals_panel)
qqline(residuals_panel)
```



using a log Output scale:

```
pdata$Market.Cap.in.B.USD.[pdata$Market.Cap.in.B.USD. == 0] <- 0.001
pdata$lagy[pdata$lagy == 0] <- 0.001
fixed_eff_lag_log_rf <- plm(log(sqrt(pdata$Market.Cap.in.B.USD.)) ~
    Revenue + Gross.Profit + Net.Income + Cash.Flow.from.Operating +
        Cash.Flow.from.Financial.Activities + log(sqrt(lagy)) +
        factor(Company), data = pdata, model = "within")
residuals_panel <- residuals(fixed_eff_lag_log_rf)
shapiro.test(residuals_panel)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals_panel
## W = 0.63021, p-value < 2.2e-16
```

We don't see an increase in p-value.

```
fixed_eff_lag_log_aic <- plm(log(sqrt(pdata$Market.Cap.in.B.USD.)) ~
    Gross.Profit + Net.Income + +Share.Holder.Equity + Earning.Per.Share +
        Revenue + log(sqrt(lagy)) + factor(Company), data = pdata,
    model = "within")
residuals_panel <- residuals(fixed_eff_lag_log_aic)
shapiro.test(residuals_panel)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals_panel
## W = 0.60135, p-value < 2.2e-16
```

Final Model:

$$\log(\sqrt{\mathbf{Y}_{it}}) = \log(\sqrt{\mathbf{Y}_{(i-1)t}}) + \beta_{\mathbf{0}} + \mathbf{X_{it}}\beta_{\mathbf{i}} + \mathbf{c_i} + \epsilon_{\mathbf{i}}$$

Coefficients:

```
print(fixef(fixed_eff_lag_log_aic))
```

```
##    AAPL     AIG    AMZN     BCS    GOOG    INTC     MCD    MSFT    NVDA     PCG
## 1.80923 1.17192 1.79166 1.23189 1.93970 1.62843 1.69010 2.01027 1.46542 1.09575
##    PYPL   SHLDQ
## 1.20282 0.24114
```

```
summary(fixed_eff_lag_log_aic)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = log(sqrt(pdata$Market.Cap.in.B.USD.)) ~ Gross.Profit +
##     Net.Income + +Share.Holder.Equity + Earning.Per.Share + Revenue +
##     log(sqrt(lagy)) + factor(Company), data = pdata, model = "within")
##
## Unbalanced Panel: n = 12, T = 9-15, N = 160
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -5.121812 -0.113151  0.041112  0.181029  1.446008
##
## Coefficients:
##                         Estimate  Std. Error t-value Pr(>|t|)
## Gross.Profit        -1.0501e-06  1.2824e-05 -0.0819 0.934856
## Net.Income           3.9918e-06  6.4553e-06  0.6184 0.537325
## Share.Holder.Equity  1.1633e-06  2.3128e-06  0.5030 0.615754
## Earning.Per.Share    1.5213e-02  5.9885e-03  2.5403 0.012151 *
## Revenue              2.3187e-06  5.3651e-06  0.4322 0.666265
## log(sqrt(lagy))      2.3357e-01  7.4216e-02  3.1472 0.002009 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    61.909
## Residual Sum of Squares: 46.894
## R-Squared:      0.24254
## Adj. R-Squared: 0.15186
## F-statistic: 7.5782 on 6 and 142 DF, p-value: 4.5841e-07
```