Final Project

Andre Pimenta, Allie Buller, Aaron Berman

May 22, 2019

The first step in in the data science process is to determine what question you want to evaluate. One of the members of our group was particularly interested in stock trends, and the other two members were interested in learning more about the topic, so we decided to focus on the task of predicting yearly returns. More specifically, we wanted to find a model that predicts the change in stock price from one year to the next.

After defining a problem you want to solve, the next step is finding a dataset that you can get meaningful information from. We found a dataset from quandl, which contains yearly stock data in a csv format. To begin working with the file, download the data from https://drive.google.com/file/d/1Z1eI8qwAeTxUuo7eJy9mQrIs1Tsn4H9y/view?usp=sharing. Next, input the data into R as a data table. In our data frame, the entities (rows) represent yearly stocks, and the attributes represent different information about the stocks, including date information, identifying information, and financial information.

```
Data <- as.data.table(read_csv("/Users/andrepimenta/Downloads/ARYearly.csv"))
```

Some of the attributes that will come up include the following:

ROE: Return on Equity - How much equity is required to generate a certain amount of net income? ROA: Return on Assets - How much in assets is required to generate a certain amount of income? FCF: Free Cash Flow - What is a company's discretionary cash flow each year? ROIC: Return on Invested Capital: How much in income for all its investors does a company generate with all its capital? P/E Ratio: Price-to-Earnings Ratio - Ratio for valuing a company that measures its current share price relative to its per-share earnings PB: Price-to-Book - A ratio of the share price of a publicly-traded company to its book value per share, which is the company's total asset value less the value of its liabilities DPS: Dividend Per Share - The sum of declared dividends issued by a company for every share outstanding https://breakingintowallstreet.com/biws/kb/financial-statement-analysis/roic-vs-roe-and-roe-vs-roa/ https://breakingintowallstreet.com/biws/kb/financial-statement-analysis/free-cash-flow-example/ https://www.investopedia.com/terms/p/price-earningsratio.asp https://financial-dictionary.thefreedictionary.com/P-B+Ratio https://www.investopedia.com/terms/d/dividend-per-share.asp

As part of the preprocessing stage, categorize each entity based on its market cap. According to Investopedia, market capitalization, or market cap, is the total dollar market value of a company's outstanding shares. To calculate a company's market cap, multiply their shares outstanding by the current market price per share. As is a typical practice, we will use market cap to indicate each company's size. For more information on market capitalization and classification, please read https://www.investopedia.com/terms/m/marketcapitalization.asp.

This step will be important later on, when we group by market cap. In order to create a market cap column, you will need to use the cut function in R. RDocumentation explains that "cut divides the range of x into intervals and codes the values in x according to which interval they fall. The leftmost interval corresponds to level one, the next leftmost to level two and so on." In our case, we will be dividing the "marketcap" into the numeric intervals provided by the "levels" vector. These have been respectively labeled based on the corresponding capitalization group: nano, micro, small, mid, and large. These cutoffs and labels are based on industry norms outlined in the article above.

We will then use the mutate function to create a new column in our dataframe, entitled "capGroup", which contains the label of the market cap group corresponding to the entity.

```
# classify different market caps
levels <- c(0, 50e6, 300e6, 2e9, 10e9, Inf)
labels <- c("nano", "micro", "small", "mid", "large")
Data <- Data %>% mutate(capGroup = cut(marketcap, levels, labels = labels))
```

We want to make a few changes to the dataframe. We want a numeric column, "year", since we are concerned with yearly changes. Since the datatype of "calendardate" is datetime, we can extract the year. Next, we want to select the attributes that we think will be useful later. We chose "roe", "roa", "fcf", "roic", "pe", "pb", "calendardate", "year", "ticker", "price", "dps", "marketcap", and "capGroup". We are choosing these features because they should have a relationship with returns. This makes sense economically, as companies that are more profitable, as indicated by return on equity, return on assets, and return on invested capital, companies with high free cash flow, and companies with high value, as indicated by the price-to-book and price-to-earning ratios, should generally outperform other companies.

Next, we want to create an attribute that calculates the actual yearly return to compare to the predicted yearly return. The formula for yearly return is current share price plus dividend per share, divided by the price of the previous year, minus 1. We create a column "prevPrice" that is the price corresponding to the stock with the same ticker, from the year prior. This is used in the yearly return calculation. We will also calculate the next yearly return by taking the yearly return of the company's stock for the following year. This next yearly return is what we are going to compare against what we predict with our model.

We also need to account for extreme values. Replace any extreme value with the farthest non-outlier value. This technique is refered to as winsorizing, which you can read about here: https://www.statisticshowto.datasciencecentral.com/winsorize/. Winsorizing makes our model more robust to outliers.

```
Data <- Data %>%
  mutate(year = as.numeric(format(as.Date(calendardate, format="%Y-%m-%d"), "%Y"))) %>%
  select(roe, roa, fcf, roic, pe, pb, calendardate, year, ticker, price, dps, marketcap, capGroup) %>%
  arrange(ticker, desc(as.Date(Data$calendardate, format="%d/%m/%Y"))) %>%
  group_by(ticker) %>%
  mutate(prevPrice = shift(price, n=-1)) %>%
  mutate(yearlyRet = (price + dps) / prevPrice - 1) %>%
  mutate(NextYearlyRet = shift(yearlyRet)) %>%
  filter(year > 1998 & year < 2018) %>% # not many datapoints outside this range
  filter(price > 1) # filter out penny stocks
Data <- na.omit(as.data.table(Data))

# winsorize regressors
Wins <- function(x, left, right) {
  q <- quantile(x, c(left,right), type = 5)
  indx <- findInterval(x, q,left.open = TRUE)
  x[indx == 0] <- q[[1]]
  x[indx == 2] <- q[[2]]
  x
}

# clean data with winsorizing, not removing but changing outliers to farthest non outlier value
Xfactors <- colnames(Data)[c(1:6)]
Data <- Data[, c(.(capGroup=capGroup, ticker = ticker, NextYearlyRet=NextYearlyRet), lapply(.SD, functi
```

We then want to find which variables might be correlated to the yearly return variable, "NextYearlyRet" that we just calculated. First, transform the data to get the z score. Each z score is calculated by sutracting the mean of the feature from the value, and dividing by its standard deviation. This ensures each scaled feature has a center of 0 and a standard deviation of 1. Using these scaled values, we want to find the correlation between each variable and the yearly return. From here, we find the mean correlation per feature over the years in order to determine which features to use in our model. If the p-value is significant, this means that that variable is significantly related to the yearly return. As we should be able to see from the table, the relationship between each variable and our yearly return are significant. This is indicated by p-values close to 0. This tells us that these variable would be good to use in our linear regression, since we want to use

variables that have a relationship with yearly returns.

Notice that we performed the Spearman correlation, instead of a typical Pearson correlation. This is because Spearman correlations capture relationships that are not necessarily strictly linear. Rather, it measure monotonic relationships. Since the relationships between our variables may be monotonic, but not linear, we wanted to still capture these as potential predictors in our model. A good resource to learn more aoubt monotonic relationships and Spearman's correlation is: https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php.

```
# Z-scoring
Data <- Data[, paste0("z", Xfactors) := lapply(.SD, function(x) as.vector(scale(as.numeric(x)))),.SDcols

# correlation of feature to return per year
ZX_factors <- paste0("z", Xfactors)
corrs <- Data[, lapply(.SD, function(x) cor(NextYearlyRet, x, method = "spearman")),.SDcols = ZX_factors

# mean correlation per feature over the years -- shows which variables might be correlated with returns
model <- lm(as.matrix(corrs) ~ 1)
model %>% tidy()
```

```
## # A tibble: 6 x 6
##   response term        estimate std.error statistic    p.value
##   <chr>    <chr>          <dbl>     <dbl>     <dbl>      <dbl>
## 1 zroe     (Intercept)   0.0960    0.0221      4.35 0.000384
## 2 zroa     (Intercept)   0.114     0.0245      4.63 0.000207
## 3 zfcf     (Intercept)   0.124     0.0195      6.38 0.00000523
## 4 zroic    (Intercept)   0.0785    0.0208      3.78 0.00137
## 5 zpe      (Intercept)   0.0750    0.0228      3.30 0.00401
## 6 zpb      (Intercept)  -0.0588    0.0266     -2.21 0.0401
```

```
# Spearman correlation of each factor to returns
coeftest(model, vcov = NeweyWest(model, lag = 1, prewhite = F))
```

```
##
## t test of coefficients:
##
##                     Estimate Std. Error t value  Pr(>|t|)
## zroe:(Intercept)    0.095997   0.017233  5.5707 2.747e-05 ***
## zroa:(Intercept)    0.113655   0.020449  5.5580 2.821e-05 ***
## zfcf:(Intercept)    0.124121   0.018494  6.7116 2.713e-06 ***
## zroic:(Intercept)   0.078499   0.017688  4.4379 0.0003177 ***
## zpe:(Intercept)     0.075023   0.019913  3.7675 0.0014098 **
## zpb:(Intercept)    -0.058829   0.027266 -2.1576 0.0447218 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using the features that we found to be correlated above, create a model to predict "NextYearlyRet". Begin with a multiple linear regression model, since we want to model the relationship between multiple predictors and our response variable, yearly return. The data frame generated provides the least-squares estimate, standard error, t-statistic, and p-value for each predictor variable. For more information on multiple linear regressions and their output, read: http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm

Our null hypothesis is that there is no relationship between yearly regression and any of the predictors. This is the same as saying that the coefficient for each variable is equal to 0. In our model, all of the variables are significant, as indicated by the p-values of the t tests. This means we reject the null hypothesis for each, since the coefficient for each variable is significantly different from 0.

Look to see if any of our variables are related to each other. If they are, we would need to add interaction

terms to our model. Return on equity, return on assets, and return on invested capital are all somewhat related, as shown by dot size in the correlation graph below.

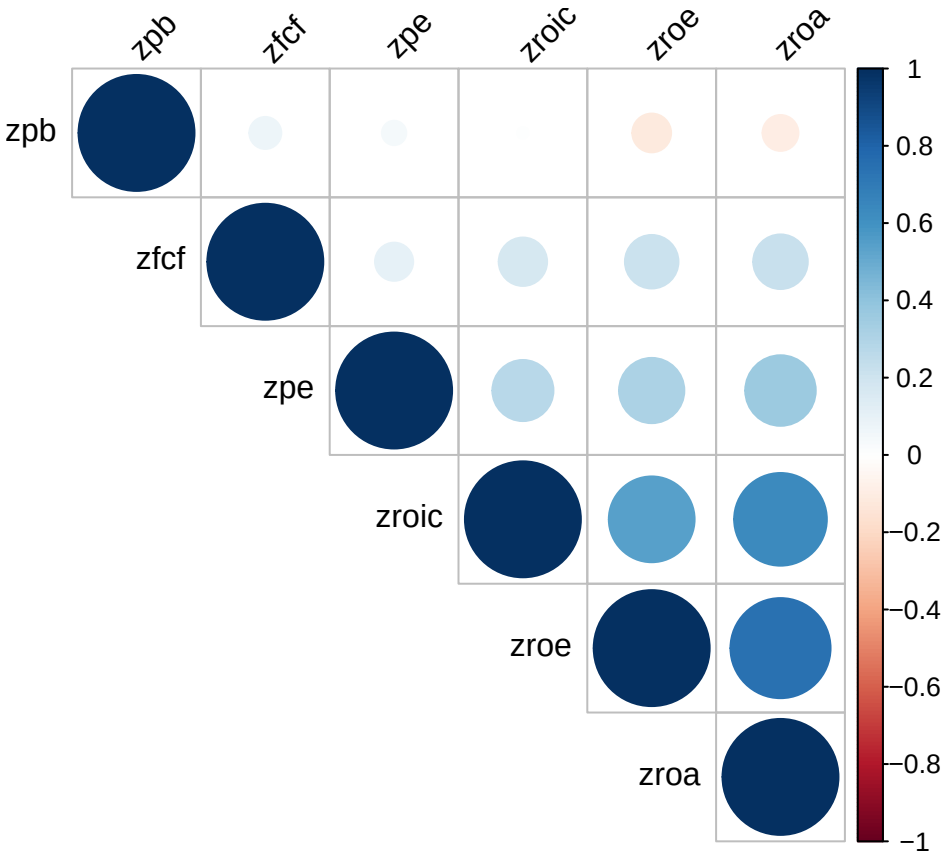
```
#Multiple linear regression
mlr1 <- lm(NextYearlyRet ~ zroe + zroa + zfcf + zroic + zpe + zpb, data=Data)
mlr1 %>% tidy()
```

```
## # A tibble: 7 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  0.138     0.00252    54.9   0.
## 2 zroe        -0.0149    0.00384    -3.89  1.00e- 4
## 3 zroa         0.0147    0.00423     3.46  5.36e- 4
## 4 zfcf         0.00204   0.00260     0.784 4.33e- 1
## 5 zroic        0.00408   0.00332     1.23  2.19e- 1
## 6 zpe         -0.00687   0.00273    -2.52  1.18e- 2
## 7 zpb         -0.0410    0.00257   -16.0   2.77e-57
```

```
# correlation matrix to look for possibly redundant factors
featuresDF <- Data %>% select(11:16)

corr1 <- round(cor(featuresDF),2)
corr2 <- rcorr(as.matrix(featuresDF))

# ROE, ROA, and ROIC all somewhat correlated
corrplot(corr2$r, type="upper", order="hclust",
         tl.col = "black", tl.srt = 45,
         p.mat = corr2$P, sig.level = 0.01, insig = "blank")
```



Now that we know there is a relationship between return on equity, return on assets, and return on invested capital, add interaction terms to the model. Interaction terms are added as products. http://www.sthda.com/english/articles/40-regression-analysis/164-interaction-effect-in-multiple-regression-essentials/ provides a good example and explanation of adding interaction terms to your model, as well as the R code to do so.

Determine which model is better – the original model, or the interaction model. We will use an ANOVA test to compare the two models. Our null hypothesis is that our second model does not perform better than the first model. We will use the ANOVA to test this hypothesis by checking if there is in fact a difference between the two models. If the statistic, which in this case is the F-statistic, is significant, we will reject the null hypothesis. As the results show, the statistic is significant and positive, which means that the second model performs significanlty better than the original model.

```
# Model with interatctions for the correlated variables (zroe,zroic,zroa)
mlr2 <- lm(NextYearlyRet ~ zroe*zroic*zroa + zfcf + zpe + zpb, data=Data)

# zfcf and interactions with zroa are not significant
mlr2 %>% tidy()
```

```
## # A tibble: 11 x 5
##    term            estimate std.error statistic  p.value
##    <chr>              <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)      0.131     0.00300    43.5   0.
##  2 zroe            -0.0149    0.00448    -3.33  8.63e- 4
##  3 zroic            0.00107   0.00438     0.243 8.08e- 1
##  4 zroa             0.0161    0.00438     3.69  2.27e- 4
##  5 zfcf             0.00182   0.00261     0.697 4.86e- 1
##  6 zpe             -0.00556   0.00276    -2.01  4.44e- 2
##  7 zpb             -0.0463    0.00281   -16.5   7.04e-61
##  8 zroe:zroic       0.0182    0.00388     4.68  2.88e- 6
##  9 zroe:zroa        0.00303   0.00223     1.36  1.75e- 1
## 10 zroic:zroa       0.00235   0.00235     1.00  3.17e- 1
## 11 zroe:zroic:zroa  0.00755   0.00132     5.71  1.15e- 8
```

```
# Interaction model is better than the previous one
anova(mlr1, mlr2)
```

```
## Analysis of Variance Table
##
## Model 1: NextYearlyRet ~ zroe + zroa + zfcf + zroic + zpe + zpb
## Model 2: NextYearlyRet ~ zroe * zroic * zroa + zfcf + zpe + zpb
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1  83505 44167
## 2  83501 44145  4    21.974 10.391 2.063e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Continue improving our model. From the p-values of the t-statistic in the interaction model, we see that free cash flow and the interactions with return on assets are not significant. Create a new model, this time removing these insignificant variables. Now, we want to perform an ANOVA between the previous model and this one, again with the null hypothesis that there is no improvement. As was the case between the first and second model, the p-value of the new model is significant, allowing us to reject the null hypothesis. Thus, there is evidence that the newest model, accounting for the insignificant variables and interactions, is better than the previous models.

```
# New model removing insignificant variables and interactions
mlr3 <- lm(NextYearlyRet ~ zroe*zroic + zroa + zpe + zpb, data=Data)
mlr3 %>% tidy()
```

```
## # A tibble: 7 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  0.136     0.00268     50.6  0.
## 2 zroe        -0.0137    0.00385     -3.55 3.81e- 4
## 3 zroic        0.00777   0.00365      2.13 3.32e- 2
## 4 zroa         0.0153    0.00422      3.62 2.94e- 4
## 5 zpe         -0.00718   0.00273     -2.63 8.63e- 3
## 6 zpb         -0.0424    0.00264    -16.1  5.78e-58
## 7 zroe:zroic   0.00411   0.00171      2.41 1.60e- 2
```

```
# New model is better than the previous
anova(mlr2, mlr3)
```

```
## Analysis of Variance Table
##
## Model 1: NextYearlyRet ~ zroe * zroic * zroa + zfcf + zpe + zpb
## Model 2: NextYearlyRet ~ zroe * zroic + zroa + zpe + zpb
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1  83501 44145
## 2  83505 44164 -4   -19.231 9.0938 2.431e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now that we have tuned our model, it is time to visualize our results. We will do this by making residual plots. A residual is the difference between observed value and predicted value. In our case, it is the difference between the follow year's actual yearly return and that predicted by the model. We plot the residuals on the y-axis and the independent variable on the x-axis. If the points are randomly spread around the x-axis, this indicates that the linear model is a good fit for our data. If this does not make sense, you can gain a basic understanding of residuals here: https://stattrek.com/regression/residual-analysis.aspx.

Examine the relationships between the residuals and different variables. Consider grouping by year and cap group to see if these have any impact. We begin by plotting the residuals by year. As the graph indicates, there is a large skew in the residual for 2008-2009. This makes sense, as the stock market crashed during this time. Thus, it makes sense why there would be a large difference between observed and predicted values, since this was an erratic time for the stock market.

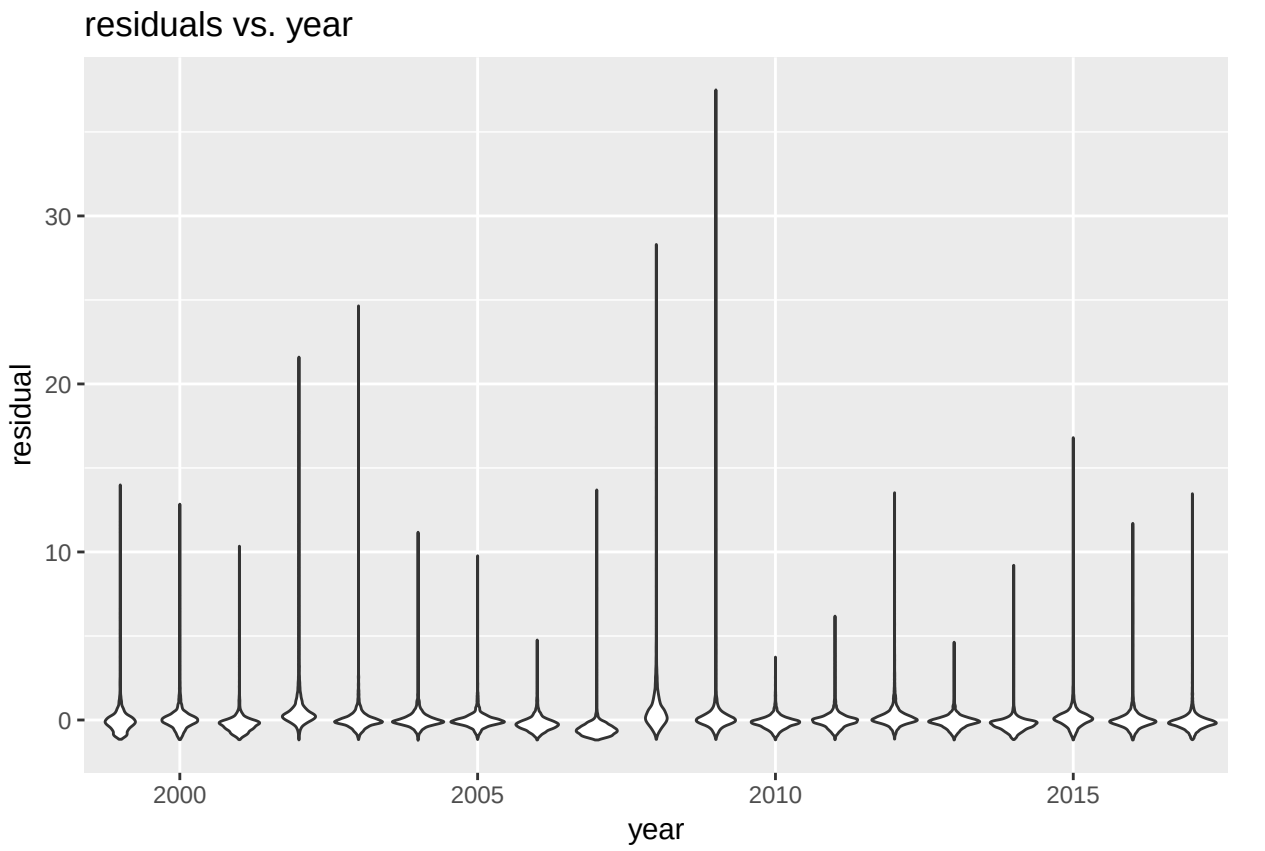
Look at residual vs return on equity, since this is one of our primary predictors. Using the cap groups we created in our preprocessing step, we can get a better understanding of the relationship between market cap and our model. While the points belonging to small, mid, and large cap companies seem to lie around the horizontal axis, there appears to be large variance in the residuals for nano and micro cap stocks. This indicates that we need to group by market cap in order to get a better understanding of the information. It is pretty customary to group by market cap when analyzing investments, since there are big difference between stocks of really small and really large companies. It makes more sense to compare stocks whose companies are similarly sized in order to make predictions about the performance of the stocks.

The third graph shows how the residuals depend on the cap group. The larger the company, the better the residual plot is. This makes sense, since the smallest companies may be less predictable overall, and thus their stocks would be less predictable in turn. Our final plot demonstrates this relationship, showing that it is best to group by cap group, since the relationship between return on equity and next year's return is different depending on the group.

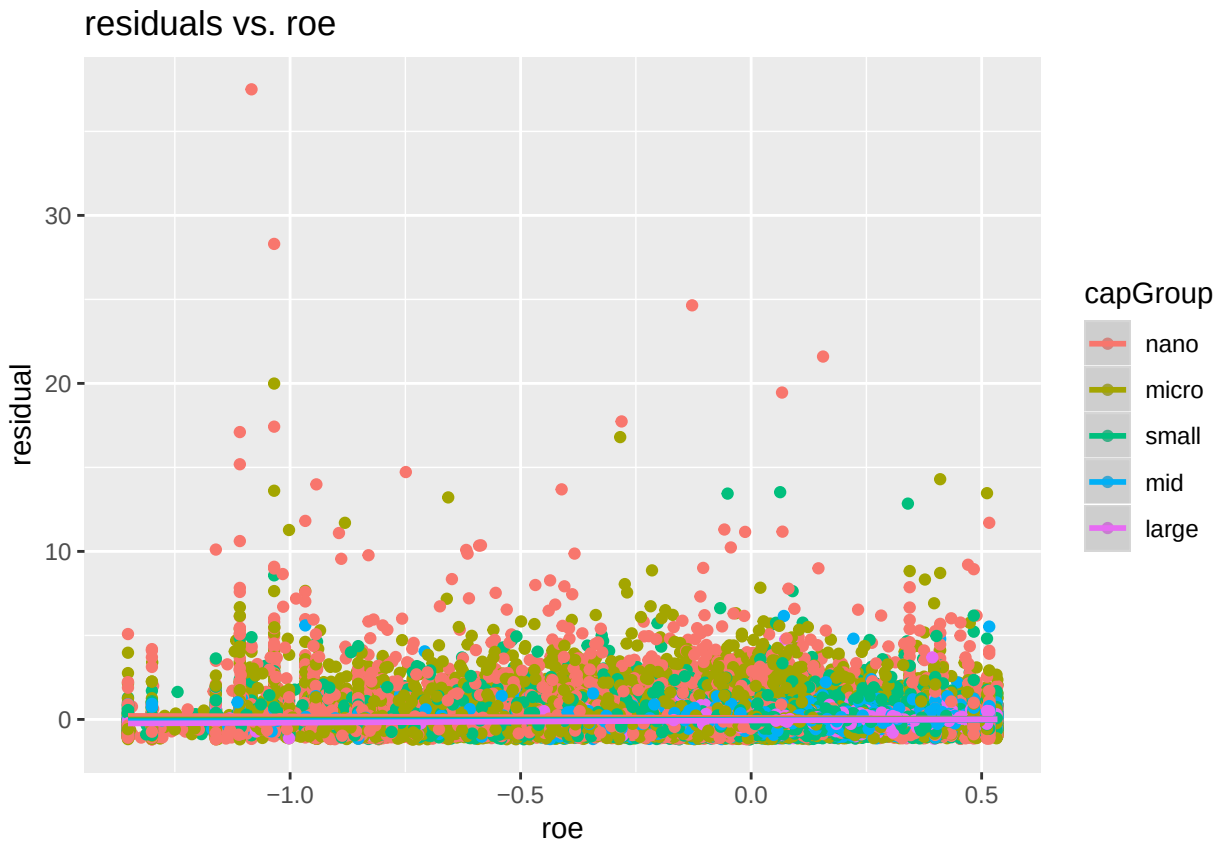
```
augmented <- mlr3 %>% augment()

Data <- as.data.frame(Data)
merged <- merge(Data, augmented, by="row.names")
```

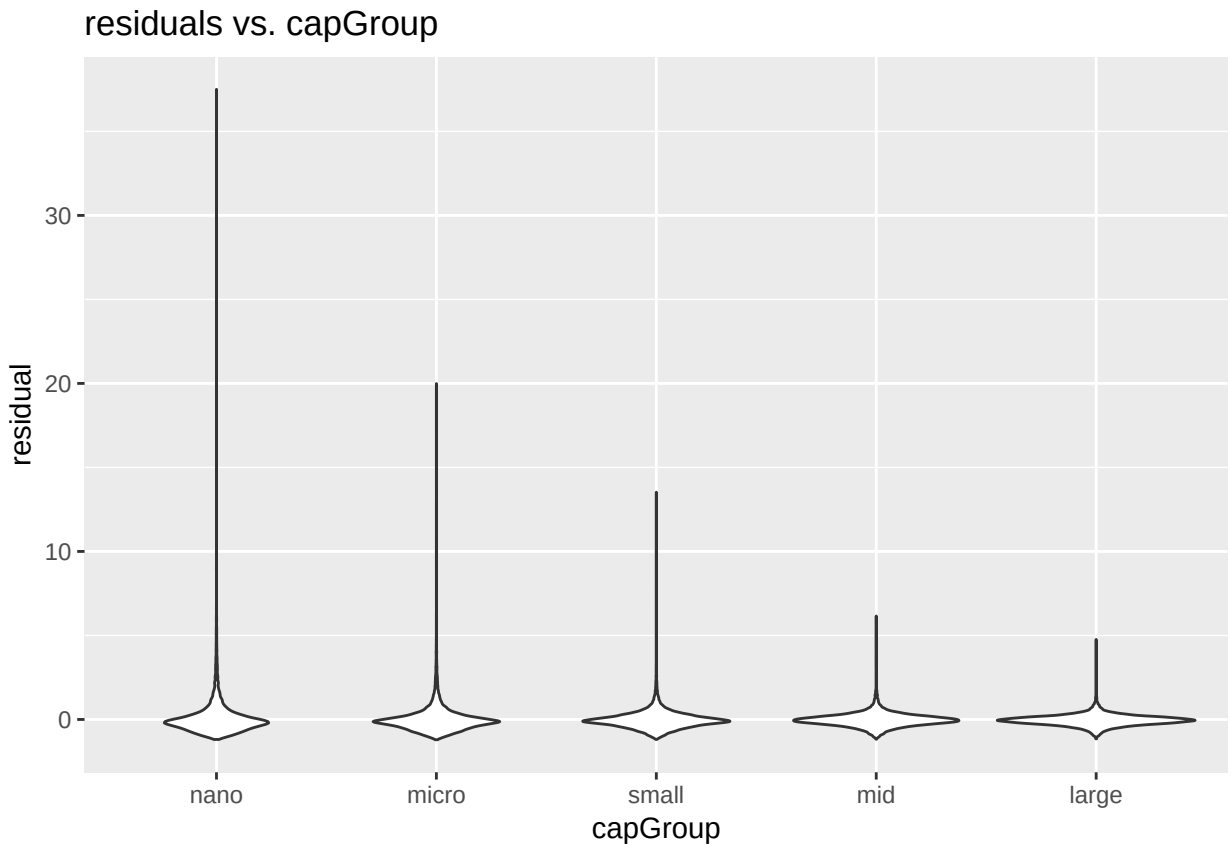
```
# Shows how the residuals are extremely bad in 2008-2009 (stock market crashed)
merged %>%
  ggplot(mapping=aes(x=factor(year), y=.resid)) +
  geom_violin() +
  labs(title="residuals vs. year ",
       x = "year",
       y = "residual") +
  scale_x_discrete(breaks=seq(1980, 2015, 5))
```



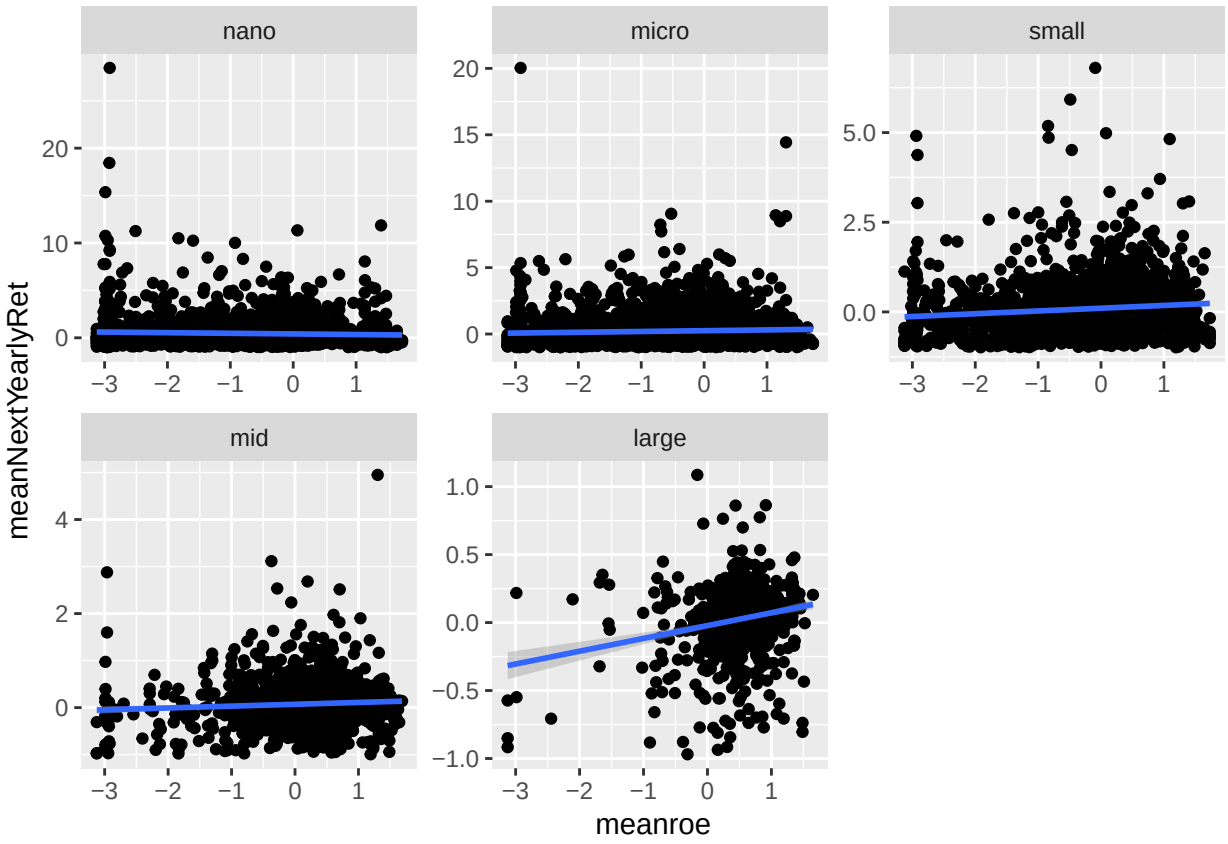

```
# Residuals are worst for nano and micro stocks - shows the need to group by market cap
merged %>%
  ggplot(mapping=aes(x=roe, y=.resid, color=capGroup)) +
  geom_point() +
  geom_smooth(method=lm) +
  labs(title="residuals vs. roe ",
       x = "roe",
       y = "residual")
```




```
# Shows how the residuals depend on capGroup
merged %>%
  ggplot(mapping=aes(x=factor(capGroup), y=.resid)) +
  geom_violin() +
  labs(title="residuals vs. capGroup ",
       x = "capGroup",
       y = "residual")
```




```
#Shows that the trend between zroe and nextYearlyRet depends on capGroup
merged %>%
  group_by(capGroup, ticker) %>%
  summarise(meanNextYearlyRet = mean(NextYearlyRet.x), meanroe = mean(zroe.x)) %>%
  ggplot(aes(x=meanroe, y = meanNextYearlyRet)) +
    facet_wrap(~capGroup, scales = "free") +
    geom_point() +
    geom_smooth(method=lm)
```



Having seen that there may be different relationship depending on cap group, check to see if this is a predictor in the model. As before, our null hypothesis is that the coefficient for cap group will not be significantly different from 0. The p-values for each cap group is significant, so we will include it in our model. Remove insignificant terms, as we did before.

Graph the residuals. Although there is still some variance in the points, the residual suggests that this model is better than the previous ones, now that we have accounted for cap group. The ANOVA results suggest the same.

```
# Regression using market cap as an independent variable
mlr4 <- lm(NextYearlyRet ~ (zroe*zroic*zroa + zfcf + zpe + zpb + capGroup), data=Data)
mlr4 %>% tidy()
```

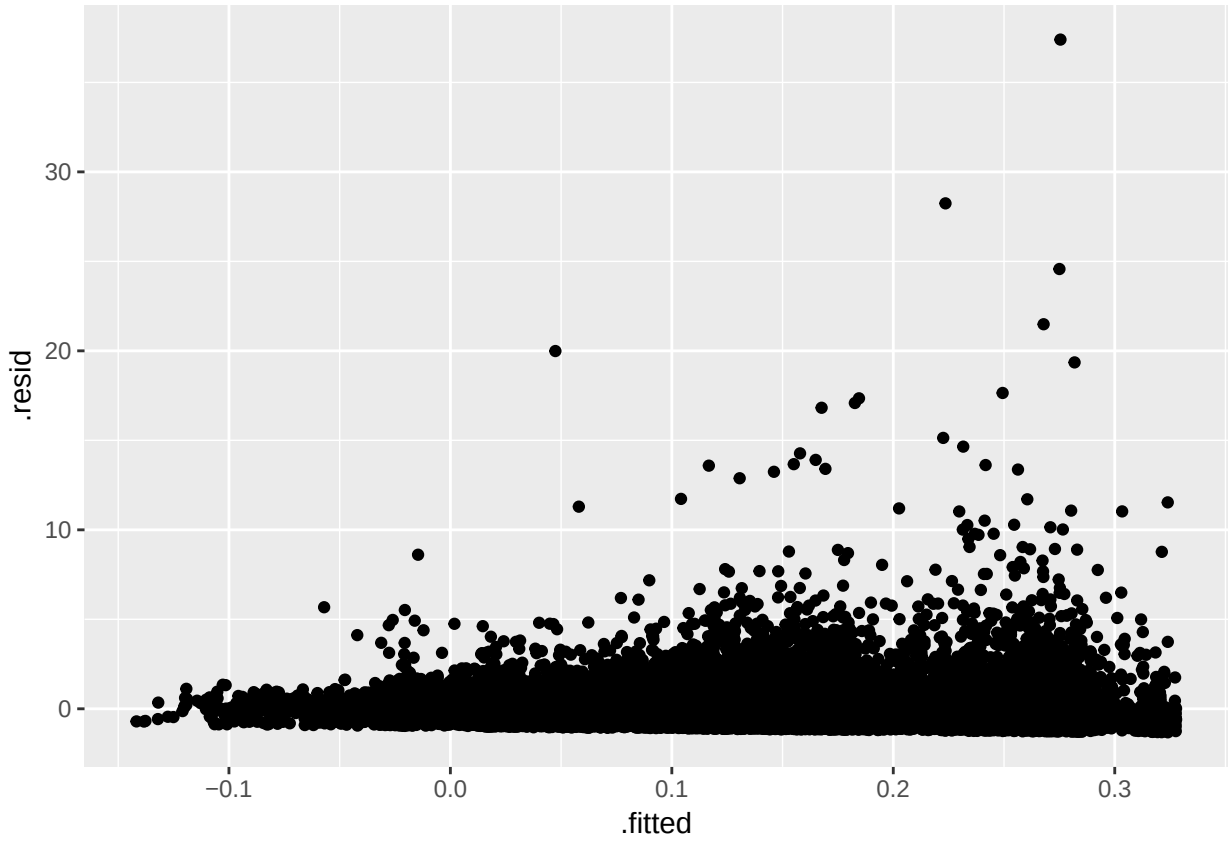
```
## # A tibble: 15 x 5
##    term             estimate std.error statistic   p.value
##    <chr>               <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)      0.260      0.00705    36.9   2.41e-295
##  2 zroe            -0.0120     0.00447    -2.70  7.00e-  3
##  3 zroic            0.000456   0.00437     0.104 9.17e-  1
##  4 zroa             0.0311     0.00443     7.01  2.41e- 12
##  5 zfcf             0.0304     0.00347     8.76  1.98e- 18
##  6 zpe             -0.00243    0.00276    -0.880 3.79e-  1
##  7 zpb             -0.0324     0.00288   -11.3   2.34e- 29
##  8 capGroupmicro   -0.105      0.00805   -13.0   1.27e- 38
##  9 capGroupsmall   -0.149      0.00822   -18.2   1.24e- 73
## 10 capGroupmid     -0.176      0.00977   -18.0   4.01e- 72
## 11 capGrouplarge   -0.249      0.0138    -18.1   2.67e- 73
## 12 zroe:zroic       0.0132     0.00388     3.40  6.84e-  4
```

```
## 13 zroe:zroa        0.00176    0.00223     0.788 4.31e-  1
## 14 zroic:zroa       0.00340    0.00234     1.45  1.47e-  1
## 15 zroe:zroic:zroa  0.00601    0.00132     4.54  5.50e-  6
```

```
# Remove insignificant interaction term
mlr5 <- lm(NextYearlyRet ~ (zroe*zroic + zfcf + zpe + zpb + capGroup), data=Data)
mlr5 %>% tidy()
```

```
## # A tibble: 11 x 5
##    term           estimate std.error statistic   p.value
##    <chr>             <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)    0.259      0.00683    37.9   9.75e-312
##  2 zroe           0.00478    0.00323     1.48  1.39e-  1
##  3 zroic          0.0133     0.00342     3.88  1.03e-  4
##  4 zfcf           0.0312     0.00347     9.01  2.20e- 19
##  5 zpe           -0.000424   0.00270    -0.157 8.75e-  1
##  6 zpb           -0.0314     0.00270   -11.6   4.02e- 31
##  7 capGroupmicro -0.101      0.00802   -12.5   4.59e- 36
##  8 capGroupsmall -0.142      0.00810   -17.5   3.61e- 68
##  9 capGroupmid   -0.167      0.00965   -17.3   3.32e- 67
## 10 capGrouplarge -0.242      0.0137    -17.7   6.03e- 70
## 11 zroe:zroic     0.00231    0.00170     1.36  1.74e-  1
```

```
#Residuals vs. Fitted values
broom::augment(mlr5) %>%
  ggplot(aes(x=.fitted, y=.resid)) +
  geom_point()
```



```r
#F-test showing that the new regression is best
anova(mlr3,mlr4)
```

```
## Analysis of Variance Table
##
## Model 1: NextYearlyRet ~ zroe * zroic + zroa + zpe + zpb
## Model 2: NextYearlyRet ~ (zroe * zroic * zroa + zfcf + zpe + zpb + capGroup)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1  83505 44164
## 2  83497 43896  8    268.01 63.724 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(mlr4, mlr5)
```

```
## Analysis of Variance Table
##
## Model 1: NextYearlyRet ~ (zroe * zroic * zroa + zfcf + zpe + zpb + capGroup)
## Model 2: NextYearlyRet ~ (zroe * zroic + zfcf + zpe + zpb + capGroup)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1  83497 43896
## 2  83501 43935 -4   -39.306 18.692 2.268e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```