edgar\_partc

Gibran Makyanie

04/05/2020

rm(list = ls())  
knitr::opts\_knit$set(root.dir = '/Volumes/Buku Gibran/edgar')  
knitr::opts\_chunk$set(eval = FALSE)  
library(tidyverse)  
library(edgar)  
library(XML)  
library(lubridate)  
library(tm)  
library(RSQLite)  
library(tidytext)  
library(udpipe)  
library(rvest)

## Warning: package 'xml2' was built under R version 3.6.2

library(readxl)  
library(qdap)  
library(sentimentr)  
library(textfeatures)  
library(BatchGetSymbols)

## Warning: package 'BatchGetSymbols' was built under R version 3.6.2

library(lubridate)  
library(DataExplorer)  
library(gridExtra)  
library(stm)

# Data Preparation

conn <- dbConnect(RSQLite::SQLite(), "/Volumes/Buku Gibran/edgar/edgar.db")

documents\_dataset <- dbGetQuery(conn, 'SELECT master\_index.cik, company\_name, gics\_sector, form\_type, date\_filed, year\_filed, accession\_number, cleaned\_noun, price\_adjusted\_ratio FROM master\_index INNER JOIN sp500 ON sp500.cik = master\_index.cik') %>% mutate(date\_filed = as.Date(date\_filed, origin="1970-01-01")) %>% mutate(cik = as.factor(cik), company\_name = as.factor(company\_name), gics\_sector = as.factor(gics\_sector), form\_type = as.factor(form\_type), year\_filed = as.factor(year\_filed), accession\_number = as.factor(accession\_number))  
   
  
documents\_10q <- documents\_dataset %>% filter(form\_type == '10-Q')  
save(documents\_10q, file = '/Volumes/Buku Gibran/edgar/temp/documents\_10q.rda')  
load('/Volumes/Buku Gibran/edgar/temp/documents\_10q.rda')  
  
rm(documents\_10q)  
  
documents\_10k <- documents\_dataset %>% filter(form\_type == '10-K')  
save(documents\_10k, file = '/Volumes/Buku Gibran/edgar/temp/documents\_10k.rda')  
  
  
documents\_10k <- documents\_10k %>% na.omit()  
load('/Volumes/Buku Gibran/edgar/temp/documents\_10k.rda')  
rm(documents\_dataset)

The aim of topic modelling is to discover themes or topics assumed to have in a corpus of documents. The STM package is used to guide the process of topic modelling to discover and estimate relationship of topics to the meta-data of the corpus. This section can be classified into: Initial Exploration (Unsupervised Topic Modelling), Model and Review (Supervised Topic Modelling), and Estimating Effects.

# Initial Exploration (Unsupervised Approach)

There are two main goals of initial exploration; - Identify appropriate K (kappa) number of topics - Check whether another iteration of data cleaning is required to achieve a reasonable set of topics.

By setting K = 0, the stm function will automatically select the number of topics, although there is no guarantee of achieving optimum, it is claimed to be a good start as it has a good computational advantage since it only need to run once. The heuristics of selecting K is to use the K suggested by the unsupervised algorithm, and define a set of neighbourhood of 5, including the proposed K, with an interval of 2. The neighbourhood of K is then reviewed by plotting its diognistic values.

## Pre-process Text and Building Unsupervised Model

set.seed(2107)  
  
# ----- Corpus Preparation  
processed <- textProcessor(sample\_10k$cleaned\_noun,  
 metadata = sample\_10k,  
 customstopwords = c("net","product","service","margin","volume","revenue","inventory"),  
 stem = F)  
  
threshold <- round(1/100 \* length(processed$documents),0)  
  
out\_10k <- prepDocuments(processed$documents,   
 processed$vocab,  
 processed$meta,  
 lower.thresh = threshold)  
  
# ----- STM model fitting  
stm\_unsupervised\_10k <- stm(documents = out\_10k$documents,   
 vocab = out\_10k$vocab,  
 K = 0,  
 prevalence = NULL,  
 max.em.its = 150,  
 data = out\_10k$meta,  
 reportevery = 5,  
 sigma.prior = 0.7,  
 init.type = "Spectral")  
  
save(stm\_unsupervised\_10k, file = '/Volumes/Buku Gibran/edgar/temp/stm\_unsupervised\_10k.rda')

## Evaluate Model Performance

By using the summary function on the unsupervised stm model, we can review the proposed topics including its top 7 words along arranged by FREX (overall frequency and how exlcusive the words to that topic), lift weights (higher weight when less frequent in other topics), and probablity of the word belong to the topic. Although it sounds less complex than FREX and lift, I would argue that the word-topic probability is what we are looking for to identify words that ‘cannot make their mind to which topic they belong’, or simply put, stopwords.

# ----- Review performance  
topic\_summary\_unsupervised <- summary(stm\_unsupervised\_10k)  
plot(stm\_unsupervised\_10k) # plot the topic model  
topicQuality(stm\_unsupervised\_10k,documents = out\_10k$documents) # review topic semantic-coherence  
  
# ----- Review word frequency to identify potential stopwords  
  
unsupervised\_k <- length(topic\_summary\_unsupervised$topicnums)  
top\_topic\_words <- c()  
for (i in 1:unsupervised\_k){  
 top\_topic\_words <- c(top\_topic\_words, topic\_summary\_unsupervised$prob[i,])  
}  
  
data.frame(word = top\_topic\_words) %>%  
 group\_by(word) %>%  
 summarise(count =n()) %>%  
 arrange(desc(count)) %>%  
 top\_n(50) %>%  
 mutate(word = factor(word, word)) %>%  
 ggplot(aes(x = reorder(word, count), y = count)) + geom\_bar(stat="identity") + coord\_flip() + labs(title = 'Occurance of Highest Probable Words Across Topics',subtitle = paste('a result of unsupervised stm with', unsupervised\_k, 'number of topics'), x ='Word') + scale\_y\_continuous(breaks=c(2,4,6,8,10))  
  
  
  
# ----- Review FREX words for every topic  
topic\_proportions <- colMeans(stm\_unsupervised\_10k$theta)  
  
unsupervised\_frex <- data.frame()  
for(i in 1:length(topic\_summary\_unsupervised$topicnums)){  
  
 row\_here <- tibble(topicnum= topic\_summary\_unsupervised$topicnums[i],  
 # topic\_label = topic\_labels[i],  
 proportion = 100\*round(topic\_proportions[i],4),  
 frex\_words = paste(topic\_summary\_unsupervised$frex[i,1:7],  
 collapse = ","))  
 unsupervised\_frex <- rbind(row\_here,unsupervised\_frex)  
}  
  
  
unsupervised\_frex %>%  
 arrange(desc(proportion))

# Modelling (Supervised Approach)

## Deciding on K number of topics

# ----- SearchK  
expected\_K <- c(unsupervised\_k - 6 , unsupervised\_k -4 ,unsupervised\_k - 2, unsupervised\_k, unsupervised\_k + 2 ,unsupervised\_k+4, unsupervised\_k+6) #unsupervised\_k is 64  
sk\_result <- searchK(out\_10k$documents,out\_10k$vocab, expected\_K)  
save(sk\_result, file = '/Volumes/Buku Gibran/edgar/temp/sk\_result.rda')  
  
plot(sk\_result)

## Build Model with optimum K

set.seed(2107)  
  
# ----- Corpus Preparation  
processed <- textProcessor(documents\_10k$cleaned\_noun,  
 metadata = documents\_10k,  
 customstopwords = c("net","product","service","margin","volume","revenue","inventory"),  
 stem = FALSE)  
  
threshold <- round(1/100 \* length(processed$documents),0)  
  
out\_10k <- prepDocuments(processed$documents,   
 processed$vocab,  
 processed$meta,  
 lower.thresh = threshold)  
save(out\_10k, file = '/Volumes/Buku Gibran/edgar/temp/out\_10k.rda')  
  
# ----- STM model fitting  
stm\_supervised\_10k <- stm(documents = out\_10k$documents,   
 vocab = out\_10k$vocab,  
 K = 70,  
 prevalence = ~ factor(gics\_sector) + s(year\_filed),  
 max.em.its = 150,  
 data = out\_10k$meta,  
 reportevery = 5,  
 sigma.prior = 0.7,  
 init.type = "Spectral")  
  
save(stm\_supervised\_10k, file = '/Volumes/Buku Gibran/edgar/temp/stm\_supervised\_10k.rda')  
  
topic\_summary\_supervised <- summary(stm\_supervised\_10k)

## Evaluate Supervised Model Performance

# ----- Review performance  
plot(stm\_supervised\_10k) # plot the topic model  
topicQuality(stm\_supervised\_10k,documents = out\_10k$documents) # review topic semantic-coherence  
  
  
# ----- Review word frequency to identify potential stopwords  
top\_topic\_words <- c()  
supervised\_k <- length(topic\_summary\_supervised$topicnums)  
  
for (i in 1:supervised\_k){  
 top\_topic\_words <- c(top\_topic\_words, topic\_summary$prob[i,])  
}  
  
data.frame(word = top\_topic\_words) %>%  
 group\_by(word) %>%  
 summarise(count =n()) %>%  
 arrange(desc(count)) %>%  
 top\_n(50) %>%  
 mutate(word = factor(word, word)) %>%  
 ggplot(aes(x = reorder(word, count), y = count)) + geom\_bar(stat="identity") + coord\_flip() + labs(title = 'Occurance of Highest Probable Words Across Topics',subtitle = paste('a result of supervised stm with', supervised\_k, 'number of topics'), x ='Word')  
  
  
# ----- Review FREX words for every topic  
topic\_proportions <- colMeans(stm\_supervised\_10k$theta)  
  
supervised\_frex <- data.frame()  
for(i in 1:length(topic\_summary\_supervised$topicnums)){  
  
 row\_here <- tibble(topicnum= topic\_summary\_supervised$topicnums[i],  
 # topic\_label = topic\_labels[i],  
 proportion = 100\*round(topic\_proportions[i],4),  
 frex\_words = paste(topic\_summary\_supervised$frex[i,1:7],  
 collapse = ","))  
 supervised\_frex <- rbind(row\_here,supervised\_frex)  
}  
rm(row\_here)  
  
supervised\_frex %>%  
 arrange(desc(proportion)) %>%  
 filter(topicnum == 68)

## Estimating Effect: How Topic affect Stock Price Change?

# ----- All topics effect  
convergence <- as.data.frame(stm\_supervised\_10k$theta)  
colnames(convergence) <- paste0("topic",1:70)  
  
regression\_data <- cbind(out\_10k$meta,convergence) %>% na.omit() %>% select(-c(cleaned\_noun, accession\_number, date\_filed, cik, year\_filed, company\_name, gics\_sector, form\_type))  
  
topic\_regression <- lm(price\_adjusted\_ratio ~ . ,data = regression\_data)  
  
  
topic\_regression\_summary <- head(as.data.frame(summary(topic\_regression)$coefficients) %>%  
 tibble::rownames\_to\_column() %>%  
 mutate(absolute\_t\_value = abs(`t value`)) %>%  
 arrange(desc(absolute\_t\_value)) , 10) %>%  
 mutate(rowname=factor(rowname, levels=rowname)) %>%  
 mutate(significance = case\_when(`Pr(>|t|)` <= 0.001 ~ 'significant\*\*\*', `Pr(>|t|)` <= 0.01 ~ 'significant\*\*', `Pr(>|t|)` <= 0.05 ~ 'significant\*', TRUE ~ 'not significant')) %>%  
 mutate(r\_squared = paste('Multiple R-squared:',as.character(round(summary(topic\_regression)$r.squared,3))))  
  
ggplot(topic\_regression\_summary, aes(x = reorder(rowname, absolute\_t\_value), y = absolute\_t\_value, fill=significance)) + geom\_bar(stat = "identity") +  
 labs(title ='Top 10 Topic Features for predicting Stock Price Change after 10-K Filings', y = 'Absolute t-value', x = 'Features') + coord\_flip()  
  
  
# ----- Topic effect singualar  
library(stargazer)  
topic68\_regression <- lm(price\_adjusted\_ratio ~ topic68,data = regression\_data)  
topic44\_regression <- lm(price\_adjusted\_ratio ~ topic44,data = regression\_data)  
topic4\_regression <- lm(price\_adjusted\_ratio ~ topic4,data = regression\_data)  
stargazer::stargazer(topic68\_regression,topic44\_regression,topic4\_regression,type = "text")  
  
rm(topic\_regression, topic\_regression\_summary, topic68\_regression, topic44\_regression, topic4\_regression, regression\_data)  
  
  
# ----- Plot most significant topics in cloud words  
cloud(stm\_supervised\_10k, topic = 68, type = c("model"), max.words = 100)  
cloud(stm\_supervised\_10k, topic = 44, type = c("model"), max.words = 100)  
cloud(stm\_supervised\_10k, topic = 4, type = c("model"), max.words = 100)

## Estimate Effects: Topics Proportions over time

load('/Volumes/Buku Gibran/edgar/temp/stm\_supervised\_10k.rda')  
load('/Volumes/Buku Gibran/edgar/temp/out\_10k.rda')  
  
out\_10k$meta$year\_filed <- as.numeric(out\_10k$meta$year\_filed)  
  
effects\_10k <- estimateEffect(~ factor(gics\_sector) + s(year\_filed),  
 stmobj = stm\_supervised\_10k,  
 metadata = out\_10k$meta,  
 uncertainty = 'None')  
  
  
  
  
convergence <- as.data.frame(stm\_supervised\_10k$theta)  
colnames(convergence) <- paste0("topic",1:70)  
  
topics\_of\_interest <- c(68,44,4)  
topic\_labels <- c("Crisis","Hospitality & Travel", "Financial Terms")  
  
for (t in 1:length(topics\_of\_interest)){  
   
plot(effects\_10k, covariate = "year\_filed",  
 topics = topics\_of\_interest[t],  
 model = stm\_supervised\_10k, method = "continuous",  
 xaxt='n',  
 xlab="Year Filed",  
 main = paste('Topic', topics\_of\_interest[t],':',topic\_labels[t]),  
 printlegend = FALSE,  
 linecol = "black",  
 labeltype = "none")  
  
 axis(1,at=seq(from=1,   
 to= length(unique(out\_10k$meta$year\_filed)),  
 by=1), labels= c('2009','2010','2011','2012','2013','2014','2015','2016','2017','2018','2019'))  
   
}  
  
out\_10k$meta$year\_filed <- as.factor(out\_10k$meta$year\_filed)

# Evaluate Additive Predictability of Topics

## Model Fitting with Topic Features as addition

load(file = '/Volumes/Buku Gibran/edgar/temp/model\_data\_10k.rda')  
# ----- Adding significant topic variables to Exhaustive Model  
regression\_data\_full <- cbind(out\_10k$meta,convergence) %>% select(cik, company\_name, accession\_number, gics\_sector, year\_filed, topic68, topic44, topic4) %>% na.omit() %>% left\_join(model\_data\_10k %>% select(-c(year\_filed, gics\_sector, company\_name)), by= 'accession\_number') %>% ungroup() %>% drop\_na()  
  
accession\_number <- regression\_data\_full$accession\_number  
year\_filed <- regression\_data\_full$year\_filed  
predicted\_ratio <- regression\_data\_full$predicted\_ratio  
regression\_data\_full <- regression\_data\_full %>% select(-c(year\_filed, accession\_number, predicted\_ratio, cik))  
  
model\_10k\_with\_topic <- lm(price\_adjusted\_ratio ~., data = regression\_data\_full, na.action=na.exclude)  
  
summary(model\_10k\_with\_topic)  
  
regression\_data\_full$new\_predicted\_ratio <- stats::predict(model\_10k\_with\_topic, newdata = regression\_data\_full %>% select(-c(price\_adjusted\_ratio)))  
regression\_data\_full$year\_filed <- year\_filed # add back year\_filed as to help grouping  
regression\_data\_full$accession\_number <- accession\_number  
regression\_data\_full$prev\_predicted\_ratio <- predicted\_ratio  
  
  
save(model\_10k\_with\_topic, file = '/Volumes/Buku Gibran/edgar/temp/model\_10k\_with\_topic.rda')  
save(regression\_data\_full, file = '/Volumes/Buku Gibran/edgar/temp/regression\_data\_full.rda')  
  
load(file = '/Volumes/Buku Gibran/edgar/temp/model\_10k\_with\_topic.rda')  
load(file = '/Volumes/Buku Gibran/edgar/temp/regression\_data\_full.rda')

## Model Evaluation

# ----- Evaluate Model  
local\_df <- head(as.data.frame(summary(model\_10k\_with\_topic)$coefficients) %>%  
 tibble::rownames\_to\_column() %>%  
 mutate(absolute\_t\_value = abs(`t value`)) %>%  
 arrange(desc(absolute\_t\_value)) , 1000) %>%  
 mutate(rowname=factor(rowname, levels=rowname)) %>%  
 mutate(significance = case\_when(`Pr(>|t|)` <= 0.001 ~ 'significant\*\*\*', `Pr(>|t|)` <= 0.01 ~ 'significant\*\*', `Pr(>|t|)` <= 0.05 ~ 'significant\*', TRUE ~ 'not significant')) %>%  
 mutate(r\_squared = paste('Multiple R-squared:',as.character(round(summary(model\_10k\_with\_topic)$r.squared,3)))) %>%  
 mutate(category = case\_when(grepl("company\_name", rowname, fixed = TRUE) ~ 'Company Feature', grepl("topic", rowname, fixed = TRUE) ~ 'Topic Feature', TRUE ~ 'Sentiment Feature'))  
  
  
feature\_category <- c('Topic Feature','Sentiment Feature', 'Company Feature')  
  
model\_df <- data.frame()  
for(v in 1:length(feature\_category)) {  
 top\_n <- local\_df %>% filter(category == feature\_category[v]) %>% arrange(desc(absolute\_t\_value))  
 top\_n <- top\_n[1:10,]  
 model\_df <- bind\_rows(model\_df, top\_n) %>% drop\_na()  
}  
  
  
ggplot(model\_df, aes(x = reorder(rowname, absolute\_t\_value), y = absolute\_t\_value, fill=significance)) + geom\_bar(stat = "identity") +  
 labs(title ='Model Descriptives: Top 10 features', subtitle = 'groupings on feature type',y = 'Absolute t-value', x = 'Features', caption = paste0('R-squared = ', as.character(round(summary(model\_10k\_with\_topic)$r.squared,3)) )) + coord\_flip() + ylim(0,11) +  
 facet\_wrap(~category, nrow = 4, scales = "free")  
  
rm(top\_n, model\_df, local\_df)

rsq <- function (x, y) cor(x, y) ^ 2 # setup R-squared calculation  
  
# ----- Company Level  
company\_agg <- regression\_data\_full %>%  
 group\_by(year\_filed, company\_name, gics\_sector) %>%  
 summarise(actual\_ratio = mean(price\_adjusted\_ratio),  
 prev\_predicted\_ratio = mean(prev\_predicted\_ratio),  
 new\_predicted\_ratio = mean(new\_predicted\_ratio))  
  
company\_agg\_rsquared\_prev <- round(rsq(company\_agg$actual\_ratio, company\_agg$prev\_predicted\_ratio), 3) # calculate R-squared between actual and predicted  
company\_agg\_rsquared\_new <- round(rsq(company\_agg$actual\_ratio, company\_agg$new\_predicted\_ratio), 3) # calculate R-squared between actual and predicted  
  
   
company\_agg\_actual <- company\_agg %>% select(-c(prev\_predicted\_ratio, new\_predicted\_ratio)) %>% mutate(price\_adjusted\_ratio = actual\_ratio, group = 'Actual') %>% select(-actual\_ratio)  
company\_agg\_predicted\_prev <- company\_agg %>% select(-c(actual\_ratio, new\_predicted\_ratio)) %>% mutate(price\_adjusted\_ratio = prev\_predicted\_ratio, group = 'Model wihtout Topics') %>% select(-prev\_predicted\_ratio)  
company\_agg\_predicted\_new <- company\_agg %>% select(-actual\_ratio, prev\_predicted\_ratio) %>% mutate(price\_adjusted\_ratio = new\_predicted\_ratio, group = 'Model with Topics') %>% select(-new\_predicted\_ratio)  
   
company\_agg <- bind\_rows(company\_agg\_actual, company\_agg\_predicted\_prev, company\_agg\_predicted\_new)  
   
# ----- Plot Actual vs Model  
ggplot(company\_agg, aes(x = year\_filed, y = price\_adjusted\_ratio, color = gics\_sector)) + geom\_line(aes(group = company\_name)) + coord\_cartesian(ylim=c(-50,50)) + facet\_wrap(~group, ncol = 4, scales = "free") + labs(title ='Model Performance: Actual vs Models', subtitle = 'groupings on company Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared from ', company\_agg\_rsquared\_prev,' to ', company\_agg\_rsquared\_new))  
  
rm(company\_agg, company\_agg\_rsquared\_prev, company\_agg\_rsquared\_new, company\_agg\_actual, company\_agg\_predicted\_prev, company\_agg\_predicted\_new)

# ----- GICS Level  
gics\_agg <- regression\_data\_full %>%  
 group\_by(year\_filed, gics\_sector) %>%  
 summarise(actual\_ratio = mean(price\_adjusted\_ratio),  
 prev\_predicted\_ratio = mean(prev\_predicted\_ratio),  
 new\_predicted\_ratio = mean(new\_predicted\_ratio))  
  
gics\_agg\_rsquared\_prev <- round(rsq(gics\_agg$actual\_ratio, gics\_agg$prev\_predicted\_ratio), 3) # calculate R-squared between actual and predicted  
gics\_agg\_rsquared\_new <- round(rsq(gics\_agg$actual\_ratio, gics\_agg$new\_predicted\_ratio), 3) # calculate R-squared between actual and predicted  
  
   
gics\_agg\_actual <- gics\_agg %>% select(-c(prev\_predicted\_ratio, new\_predicted\_ratio)) %>% mutate(price\_adjusted\_ratio = actual\_ratio, group = 'Actual') %>% select(-actual\_ratio)  
gics\_agg\_predicted\_prev <- gics\_agg %>% select(-c(actual\_ratio, new\_predicted\_ratio)) %>% mutate(price\_adjusted\_ratio = prev\_predicted\_ratio, group = 'Model wihtout Topics') %>% select(-prev\_predicted\_ratio)  
gics\_agg\_predicted\_new <- gics\_agg %>% select(-actual\_ratio, prev\_predicted\_ratio) %>% mutate(price\_adjusted\_ratio = new\_predicted\_ratio, group = 'Model with Topics') %>% select(-new\_predicted\_ratio)  
   
gics\_agg <- bind\_rows(gics\_agg\_actual, gics\_agg\_predicted\_prev, gics\_agg\_predicted\_new)  
   
# ----- Plot Actual vs Model  
ggplot(gics\_agg, aes(x = year\_filed, y = price\_adjusted\_ratio, color = gics\_sector)) + geom\_line(aes(group = gics\_sector)) + coord\_cartesian(ylim=c(-10,10)) + facet\_wrap(~group, ncol = 4, scales = "free") + labs(title ='Model Performance: Actual vs Models', subtitle = 'groupings on GICS Sector Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared from ', gics\_agg\_rsquared\_prev,' to ', gics\_agg\_rsquared\_new))  
  
rm(gics\_agg, gics\_agg\_rsquared\_prev, gics\_agg\_rsquared\_new, gics\_agg\_actual, gics\_agg\_predicted\_prev, gics\_agg\_predicted\_new)

# ----- Market Level  
market\_agg <- regression\_data\_full %>%  
 group\_by(year\_filed) %>%  
 summarise(actual\_ratio = mean(price\_adjusted\_ratio),  
 prev\_predicted\_ratio = mean(prev\_predicted\_ratio),  
 new\_predicted\_ratio = mean(new\_predicted\_ratio))  
  
market\_agg\_rsquared\_prev <- round(rsq(market\_agg$actual\_ratio, market\_agg$prev\_predicted\_ratio), 3) # calculate R-squared between actual and predicted  
market\_agg\_rsquared\_new <- round(rsq(market\_agg$actual\_ratio, market\_agg$new\_predicted\_ratio), 3) # calculate R-squared between actual and predicted  
  
   
market\_agg\_actual <- market\_agg %>% select(-c(prev\_predicted\_ratio, new\_predicted\_ratio)) %>% mutate(price\_adjusted\_ratio = actual\_ratio, group = 'Actual') %>% select(-actual\_ratio)  
market\_agg\_predicted\_prev <- market\_agg %>% select(-c(actual\_ratio, new\_predicted\_ratio)) %>% mutate(price\_adjusted\_ratio = prev\_predicted\_ratio, group = 'Model wihtout Topics') %>% select(-prev\_predicted\_ratio)  
market\_agg\_predicted\_new <- market\_agg %>% select(-actual\_ratio, prev\_predicted\_ratio) %>% mutate(price\_adjusted\_ratio = new\_predicted\_ratio, group = 'Model with Topics') %>% select(-new\_predicted\_ratio)  
   
market\_agg <- bind\_rows(market\_agg\_actual, market\_agg\_predicted\_prev, market\_agg\_predicted\_new)  
  
# ----- Plot Actual vs Model  
ggplot(market\_agg, aes(x = year\_filed, y = price\_adjusted\_ratio, group =1)) + geom\_line() + coord\_cartesian(ylim=c(-10,10)) + facet\_wrap(~group, ncol = 4, scales = "free") + labs(title ='Model Performance: Actual vs Models', subtitle = 'groupings on Market Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared from ', market\_agg\_rsquared\_prev,' to ', market\_agg\_rsquared\_new))  
  
rm(market\_agg, market\_agg\_rsquared\_prev, market\_agg\_rsquared\_new, market\_agg\_actual, market\_agg\_predicted\_prev, market\_agg\_predicted\_new)