ACCT 420: Topic modeling and anomaly detection

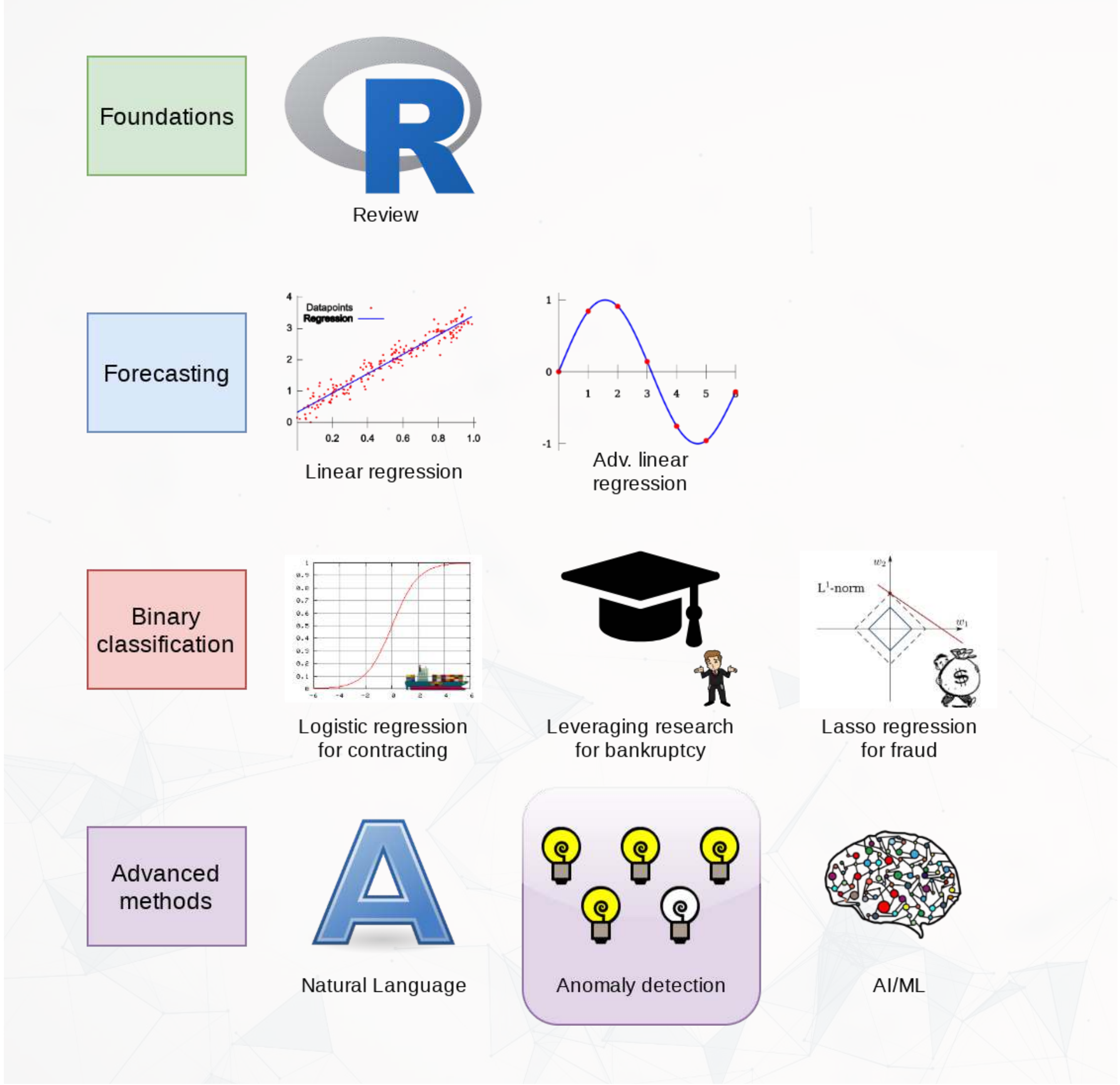
Dr. Richard M. Crowley

rcrowley@smu.edu.sg

https://rmc.link/

Front Matter

Learning objectives



- **Theory:**
  - NLP
  - Anomaly detection
- **Application:**
  - Understand annual report readability
  - Examine the *content* of annual reports
  - Group firms on content
  - Fill in missing data
- **Methodology:**
  - ML/AI (LDA, k-means, KNN)
  - Dimensionality reduction: UMAP

Group project tip #1

> For reading large files, readr is your friend

```r
library(readr)  # or library(tidyverse)
df <- read_csv("really_big_file.csv.zip")

# OR

df <- read_csv("really_big_file.csv.gz")
```

- It can read directly from zip and gzip files!
  - Like those that you can export from WRDS
  - Good for saving disk space
- It can write directly to zip and gzip files too!

Group project tip #2

> For saving intermediary results, `saveRDS()` + `readRDS()` your friend

```r
saveRDS(really_big_object, "big_df.rds")

# Later on...
df <- readRDS("big_df.rds")
```

- You can neatly save processed data, finished models, and more
  - This is particularly helpful if you want to work on something later or distribute data or results to teammates
  - As an added bonus, RDS files are compressed, taking less space on disk than csv files

> If you look at the code file for this lesson, you'll see this used extensively

Sets of documents (corpus)

Importing sets of documents (corpus)

- I will use the readtext package for this example
  - Importing all 6,933 annual reports from 2021
- Other options include using
  - purrr and `df_map()`
  - tm and `VCorpus()`
  - `{textreadr}` and `read_dir()`

```r
library(readtext)
library(quanteda)
library(quanteda.textstats)
# Needs ~6.5GB RAM
corp <- corpus(readtext("/media/Scratch/Data/10-K/2021/*.txt"))
```

Corpus summary

```
summary(corp)
```

```
                            Text Types Tokens Sentences
1   0000002178-21-000034.txt  3906  42087      1352
2   0000002969-21-000055.txt  4848  57425      1863
3   0000003499-21-000005.txt  3413  32839       989
4   0000003570-21-000039.txt  5092  70180      1725
5   0000004127-21-000058.txt  4417  40081      1106
6   0000004281-21-000049.txt  5351  71989      2119
7   0000004457-21-000040.txt  3107  22717       785
8   0000004904-21-000010.txt  7444 160570      4711
9   0000004962-21-000013.txt  5805  82050      2155
10  0000004969-21-000009.txt  3406  35469       960
11  0000004977-21-000047.txt  5782  91119      2928
12  0000005513-21-000015.txt  5953 108414      3193
13  0000006201-21-000014.txt  5870 127350      3423
14  0000006281-21-000294.txt  4794  56351      1631
```

Running readability across the corpus

```r
# Uses ~20GB of RAM...  Break corp into chunks if RAM constrained
corp_FOG <- textstat_readability(corp, "FOG")
corp_FOG %>%
  head() %>%
  html_df()
```

document	FOG
0000002178-21-000034.txt	21.11264
0000002969-21-000055.txt	22.01396
0000003499-21-000005.txt	21.81568
0000003570-21-000039.txt	24.91956
0000004127-21-000058.txt	23.87785
0000004281-21-000049.txt	22.83374

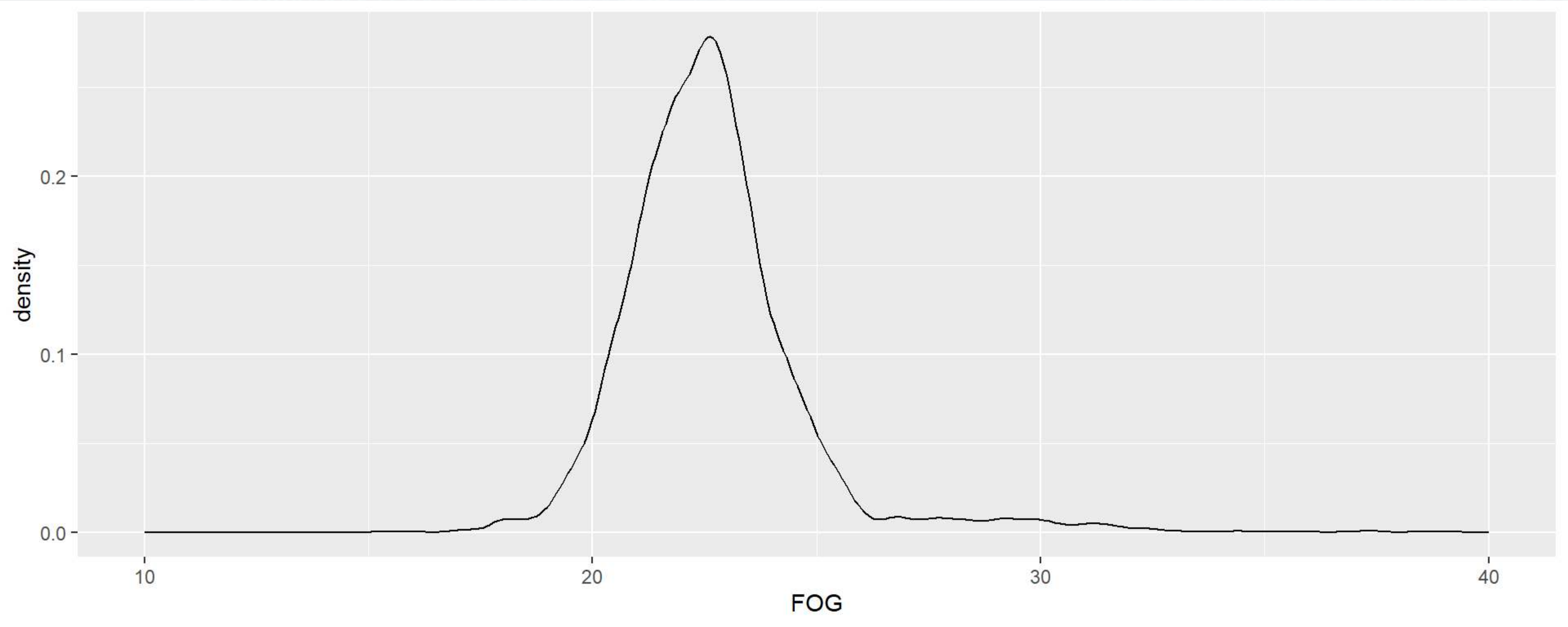
> Recall that Micorsoft's annual report had a Fog index of 20.88

Readability across documents

```
summary(corp_FOG$FOG)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  15.14   21.46   22.47   22.75   23.44  130.85       4 
```

```
ggplot(corp_FOG, aes(x=FOG)) + geom_density() + xlim(c(10,40))
```



Are certain industries' filings more readable?

- Since the SEC has their own industry code (SIC), we'll use that
- SIC codes are 4 digits
  - The first two digits represent the industry
  - The third digit represents the business group
  - The fourth digit represents the specialization
- Example: Microsoft is SIC 7372
  - 73: Business services
  - 737: Computer programming, data processing, and other computer related services
  - 7372: Prepackaged software

Are certain industries' filings more readable?

- Construct a data set of industries mapped to filings

```r
df_SIC <- read.csv('../../Data/Session_8-Filings2021.csv') %>%
  select(accession, regsic) %>%
  mutate(accession=paste0(accession, ".txt")) %>%
  rename(document=accession) %>%
  mutate(industry = case_when(
    regsic >=0100 & regsic <= 0999 ~ "Agriculture",
    regsic >=1000 & regsic <= 1499 ~ "Mining",
    regsic >=1500 & regsic <= 1799 ~ "Construction",
    regsic >=2000 & regsic <= 3999 ~ "Manufacturing",
    regsic >=4000 & regsic <= 4999 ~ "Utilities",
    regsic >=5000 & regsic <= 5199 ~ "Wholesale Trade",
    regsic >=5200 & regsic <= 5999 ~ "Retail Trade",
    regsic >=6000 & regsic <= 6799 ~ "Finance",
    regsic >=7000 & regsic <= 8999 ~ "Services",
    regsic >=9100 & regsic <= 9999 ~ "Public Admin" )) %>%
  group_by(document) %>%
  slice(1) %>%
  ungroup()
```

- Merge the industry data with the readability data

```r
corp_FOG <- corp_FOG %>% left_join(df_SIC)
```

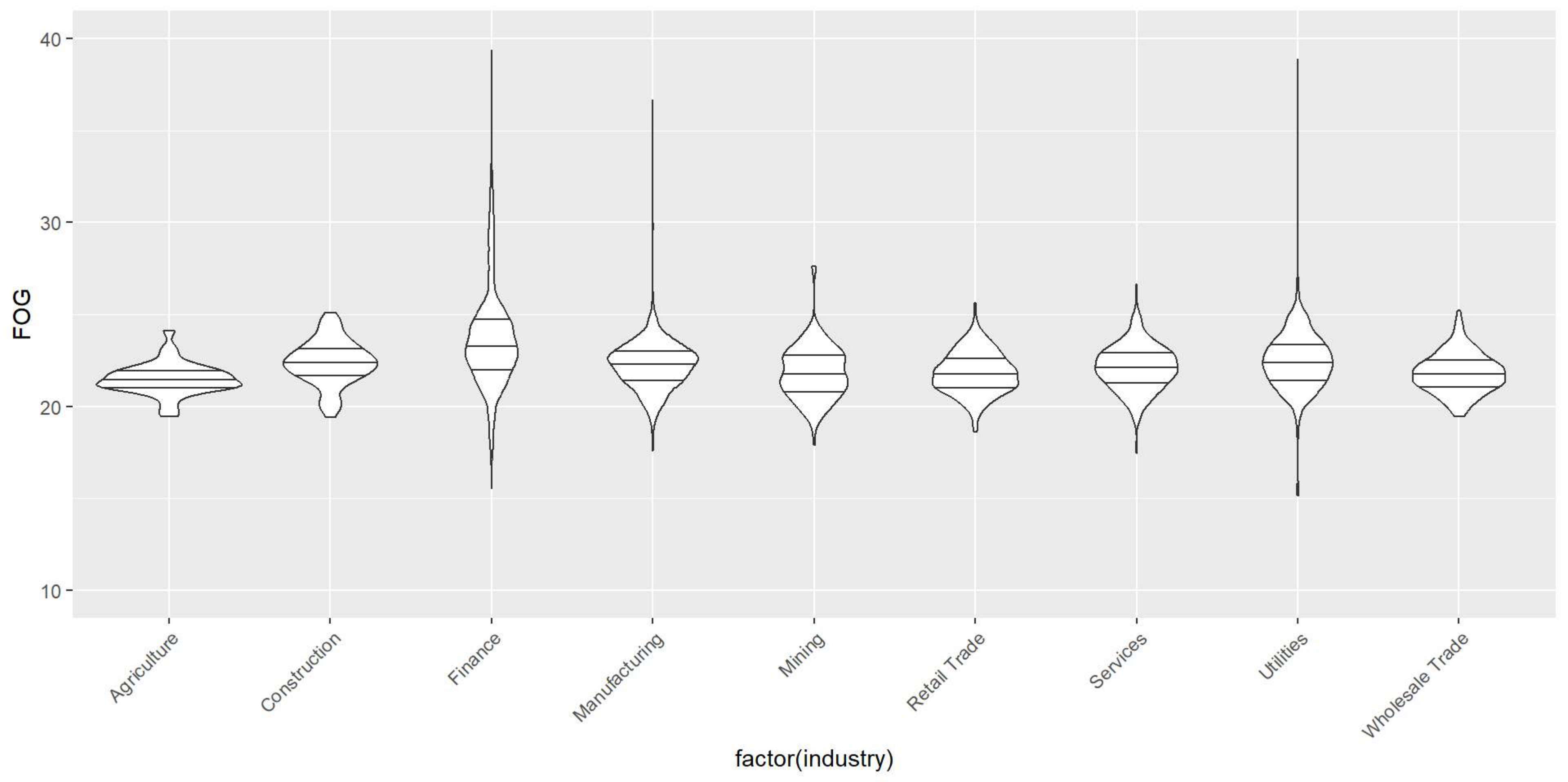
Are certain industries' filings more readable?

```r
corp_FOG %>%
  head() %>%
  html_df()
```

document	FOG	regsic	industry
0000002178-21-000034.txt	21.11264	5172	Wholesale Trade
0000002969-21-000055.txt	22.01396	2810	Manufacturing
0000003499-21-000005.txt	21.81568	6798	Finance
0000003570-21-000039.txt	24.91956	4924	Utilities
0000004127-21-000058.txt	23.87785	3674	Manufacturing
0000004281-21-000049.txt	22.83374	3350	Manufacturing

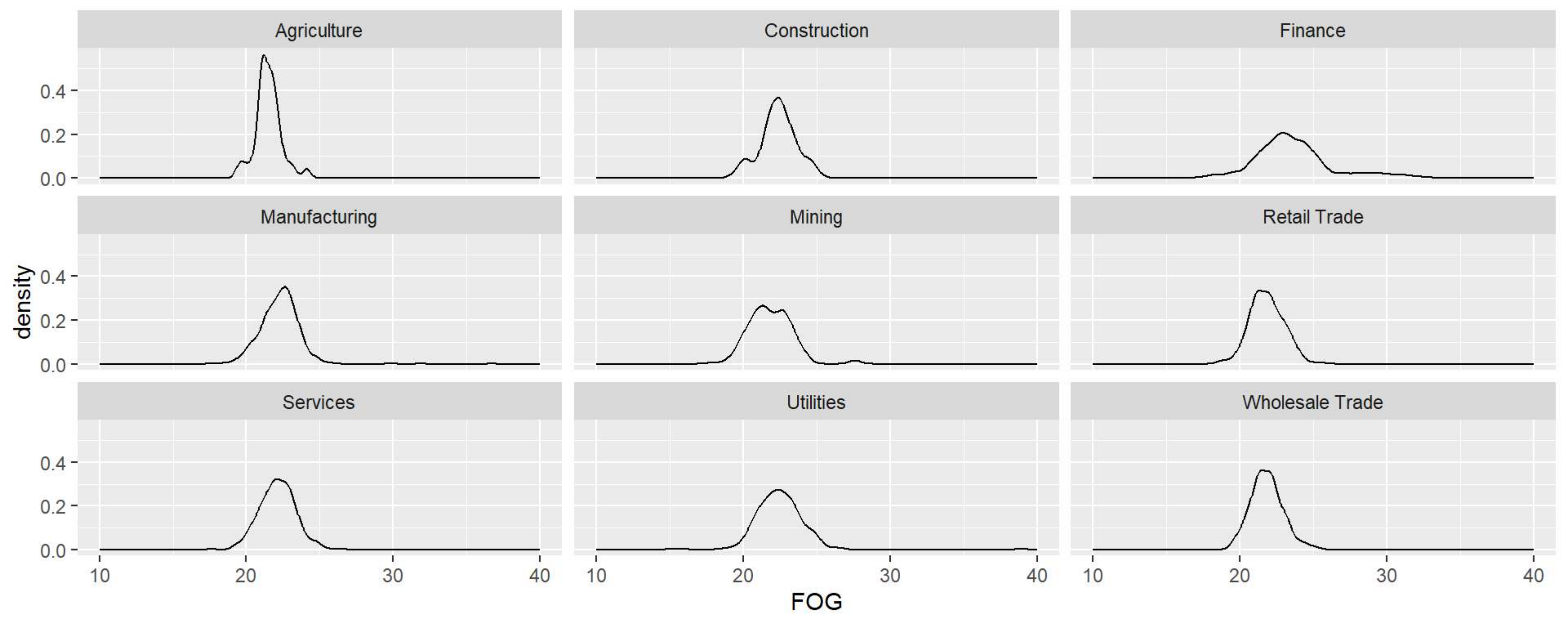
Are certain industries' filings more readable?

```r
ggplot(corp_FOG[!is.na(corp_FOG$industry),], aes(x=factor(industry), y=FOG)) +
  geom_violin(draw_quantiles = c(0.25, 0.5, 0.75)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + ylim(c(10, 40))
```



Are certain industries' filings more readable?

```r
ggplot(corp_FOG[!is.na(corp_FOG$industry),], aes(x=FOG)) +
  geom_density() + facet_wrap(~industry) + xlim(c(10, 40))
```



quanteda bonus: References across text (Global warming)

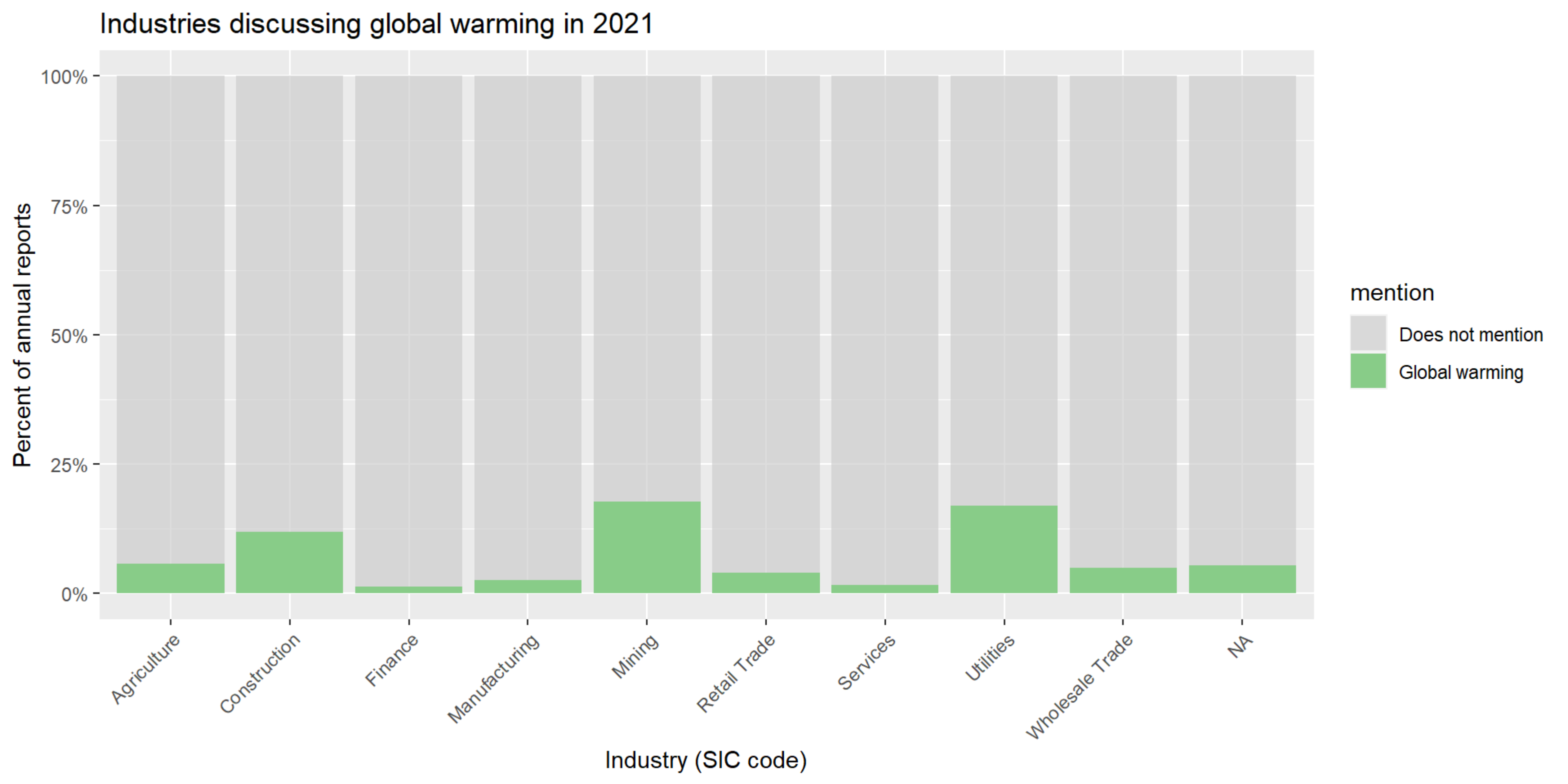
```r
corp_tokens <- tokens(corp)  # This takes a couple hours to run

# kwic() is very fast to run though
kwic(corp_tokens, pattern = phrase("global warming"), window = 3) %>%
  as.tibble() %>%
  mutate(text=paste(pre,keyword,post)) %>%
  left_join(select(df_SIC, document, industry), by = c("docname" = "document")) %>%
  select(docname, text) %>%
  sample_n(100) %>%
  datatable(options = list(pageLength = 5), rownames=F)
```

Show 5 entries Search:

docname	industry	text
0001477932-21-001405.txt	Retail Trade	its name to Global Warming Solutions , Inc
0001477932-21-001405.txt	Retail Trade	outstanding stock of Global Warming Technologies , Inc
0001537028-21-000041.txt	Mining	and contribute to global warming and other environmental
0001692115-21-000008.txt	Utilities	the effects of global warming and overall climate
0001493152-21-007032.txt	Utilities	. Some attribute global warming to increased levels

Showing 1 to 5 of 100 entries Previous 1 2 3 4 5 … 20 Next

quanteda bonus: Mentions by industry



quanteda bonus: References across text (COVID-19)

```r
corp_tokens <- tokens(corp)  # This takes a couple hours to run

# kwic() is very fast to run though
kwic(corp_tokens, pattern = phrase(c("COVID-19", "coronavirus")), window = 3) %>%
  as.tibble() %>%
  mutate(text=paste(pre,keyword,post)) %>%
  left_join(select(df_SIC, document, industry), by = c("docname" = "document")) %>%
  select(docname, text) %>%
  sample_n(100) %>%
  datatable(options = list(pageLength = 5), rownames=F)
```

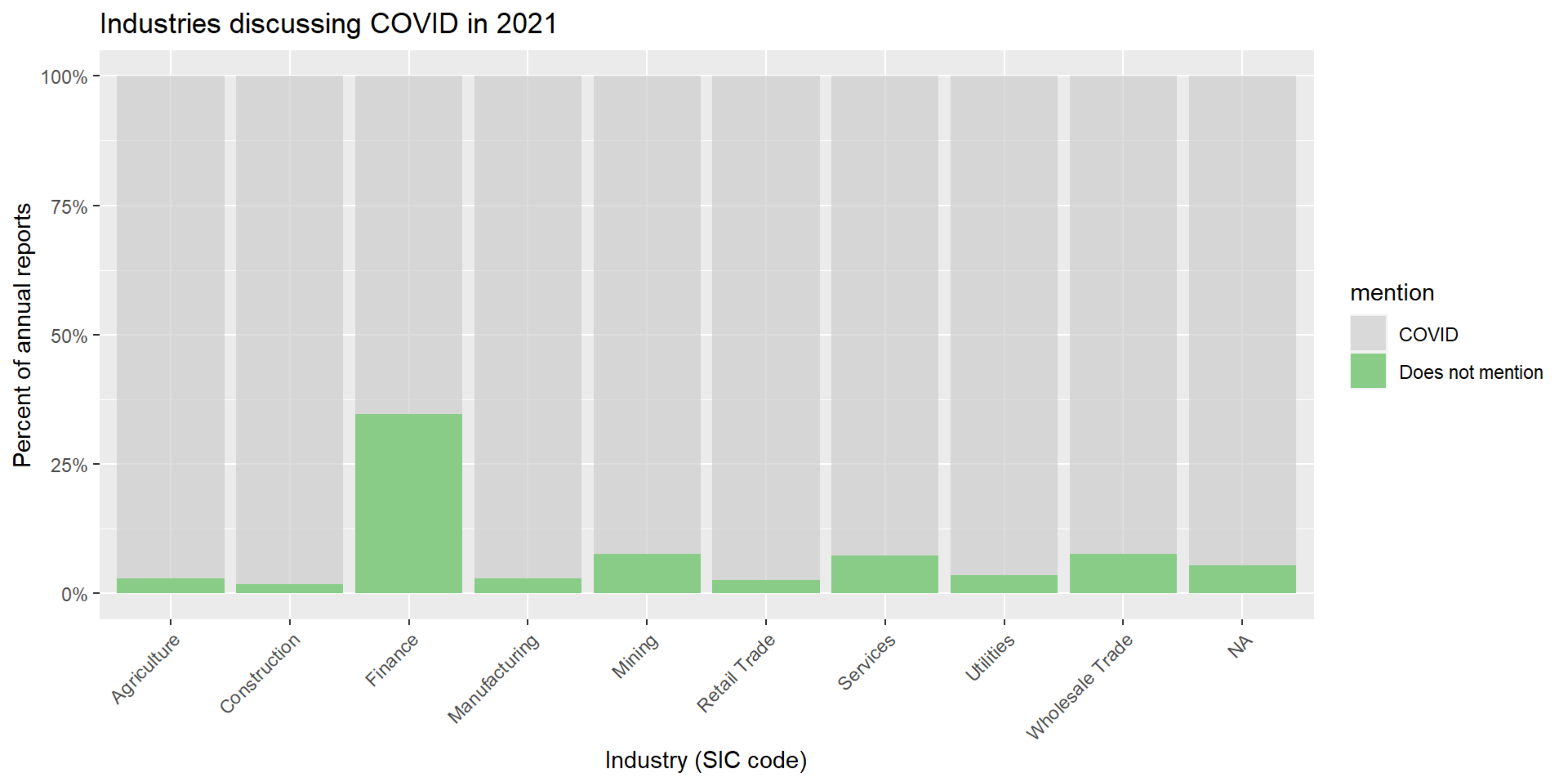
Show 5 entries Search:

docname	industry	text
0001493152-21-005827.txt	Manufacturing	impact of the COVID-19 pandemic on the
0000885508-21-000016.txt	Finance	result of the COVID-19 pandemic , Stratus
0001564590-21-011800.txt	Finance	spread of the COVID-19 virus had an
0001558370-21-003823.txt	Manufacturing	. As the COVID-19 pandemic continues to
0001564590-21-009604.txt	Services	declines due to COVID-19 and the failure

Showing 1 to 5 of 100 entries

Previous 1 2 3 4 5 … 20 Next

quanteda bonus: Mentions by industry



Going beyond simple text measures

What's next

- Armed with an understanding of how to process unstructured data, all of the sudden the amount of data available to us is expanding rapidly
- To an extent, anything in the world can be viewed as data, which can get overwhelming pretty fast
- We'll require some better and newer tools to deal with this

Problem: What do firms discuss in annual reports?

- This is a hard question to answer – our sample has 317,759,360 words in it!
  - 22.1 days for the "world's fastest reader", per this source
  - 315.2 days for a standard speed reader (700wpm)
  - **882.7 days** for an average reader (250wpm)

Solutions

1. We could read a small sample of them.
   - Imprecise, risks missing out on some types of discussion
   - We need a second computer process to apply our findings to the rest of the documents
2. Have a computer read all of them!

Recall the topic variable from session 6

- Topic was a set of 31 variables indicating *how much* a given topic was discussed
- This measure was created by making a machine read every annual report
  - The computer then used a technique called LDA to process these reports' content into topics

> This is our end goal, but we'll work our way up

Term document matrices (TDM)

- Before we begin, we'll need a matrix of word counts per document
- We'll create something called a *sparse matrix* for this
- A sparse matrix is a matrix that only lists values that aren't 0

> Think about the structure of a matrix where rows are document names and columns are individual words. How much of this matrix will be 0s?

Making a TDM

- In quanteda, use `dfm()`
- Useful additions:
  - We can pipe the output of `dfm()` to `dfm_remove()` to remove stopwords
    - You can use `remove=stopwords()` for a simple list
    - We can use SMART like last week: `remove=stopwords(source='smart')`
    - The `stopwords()` function is provided by the stopwords package, and actually supports over 50 languages, including Chinese, English, Hindi, and Malay
    - For other languages: `remove=stopwords("zh", source="stopwords-iso")`
    - With `remove=c(...)`, You can supply a list of stop words to remove
  - We can remove particularly frequent or infrequent terms with `dfm_trim()`
- We can preprocess our `tokens()` output as well
  - Pass it to `tokens_wordstem()` for stemming
    - Ex.: *cod*e, *cod*ing, and *cod*er would all become *cod*
  - `tokens()` has the options `remove_punct=T` and `remove_numbers=T` too

Making a TDM

```r
# Simplest way
tdm <- dfm(corp_tokens)

# With stopwords
tdm <- dfm(corp_tokens) %>%
       dfm_remove(stopwords(source='smart'))

# With stopwords and stemming -> Used in next slides
# 683M elements in the output
corp_tokens2 <- tokens(corp_tokens, remove_punct=TRUE, remove_numbers=TRUE) %>%
  tokens_wordstem()
tdm <- dfm(corp_tokens2) %>%
  dfm_remove(stopwords(source='smart'))
  dfm_trim(min_termfreq=10, termfreq_type = "count")
```

```r
# adding industry to the tdm
docs <- docnames(corp)
docs <- data.frame(document=docs)
docs <- docs %>% left_join(df_SIC)
docvars(tdm, field="industry") <- docs$industry
```

What words matter by industry?

```r
topfeatures(tdm, n=5, groups="industry")
```

```
$Agriculture
compani       $    oper financi    year
   9223    8862    6112    5829    5317

$Construction
      $    oper compani financi million
  14917   11563   11447   11268   10931

$Finance
   loan compani    busi       $ financi
 466138  450468  424405  423439  360063

$Manufacturing
product compani       $  includ financi
 690259  536844  498176  411262  362766
```

> This isn’t very informative

TF-IDF

- Words counts are not very informative
- Knowing the words that show up frequently in one group *but not in the others* would be much more useful
- This is called TF-IDF
  - **T**erm **F**requency-**I**nverse **D**ocument **F**requency
- Think of it roughly as:

$$\frac{\text{How many times a word is in the document}}{\text{How many documents the word is in}}$$

- We can easily calculate TF-IDF using `dfm_tfidf()` from quanteda
  - The options we'll specify are used to match a more standard output

The actual TF-IDF equation we'll use

$$\frac{f_{w,d}}{f_d} \cdot -\log_2\left(\frac{n_w}{N}\right)$$

- w represents 1 word
- d represents 1 document
- $f_{w,d}$ is the number of times w appears in d
- f_d is the number of times any word appears in d
- n_w is the number of documents with w at least once
- N is the number of documents

What words matter by industry?

```r
tfidf_mat <- dfm_tfidf(tdm, base=2, scheme_tf="prop")
topfeatures(tfidf_mat, n=5, groups=industry)
```

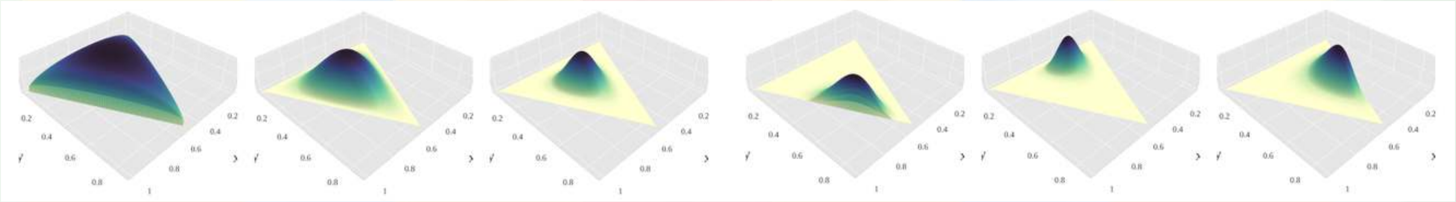
```
$Agriculture
  cannabi       prc   avocado       yew      uspb 
0.2668476 0.2599917 0.2108610 0.1990909 0.1921867 

$Construction
 homebuild 2020-12-31 2019-12-31       home  ck1723866 
 0.4848714  0.2985789  0.2360784  0.2351432  0.2049281 

$Finance
   mortgag      fargo         ab 2020-12-31 2019-12-31 
 22.799752  14.987289  13.155708  11.641575   7.004365 

$Manufacturing
    clinic        fda      trial 2020-12-31    patient 
 12.176848   8.397263   8.002860   6.812589   6.555764 
```

> These terms are often more meaningful

Moving on to LDA

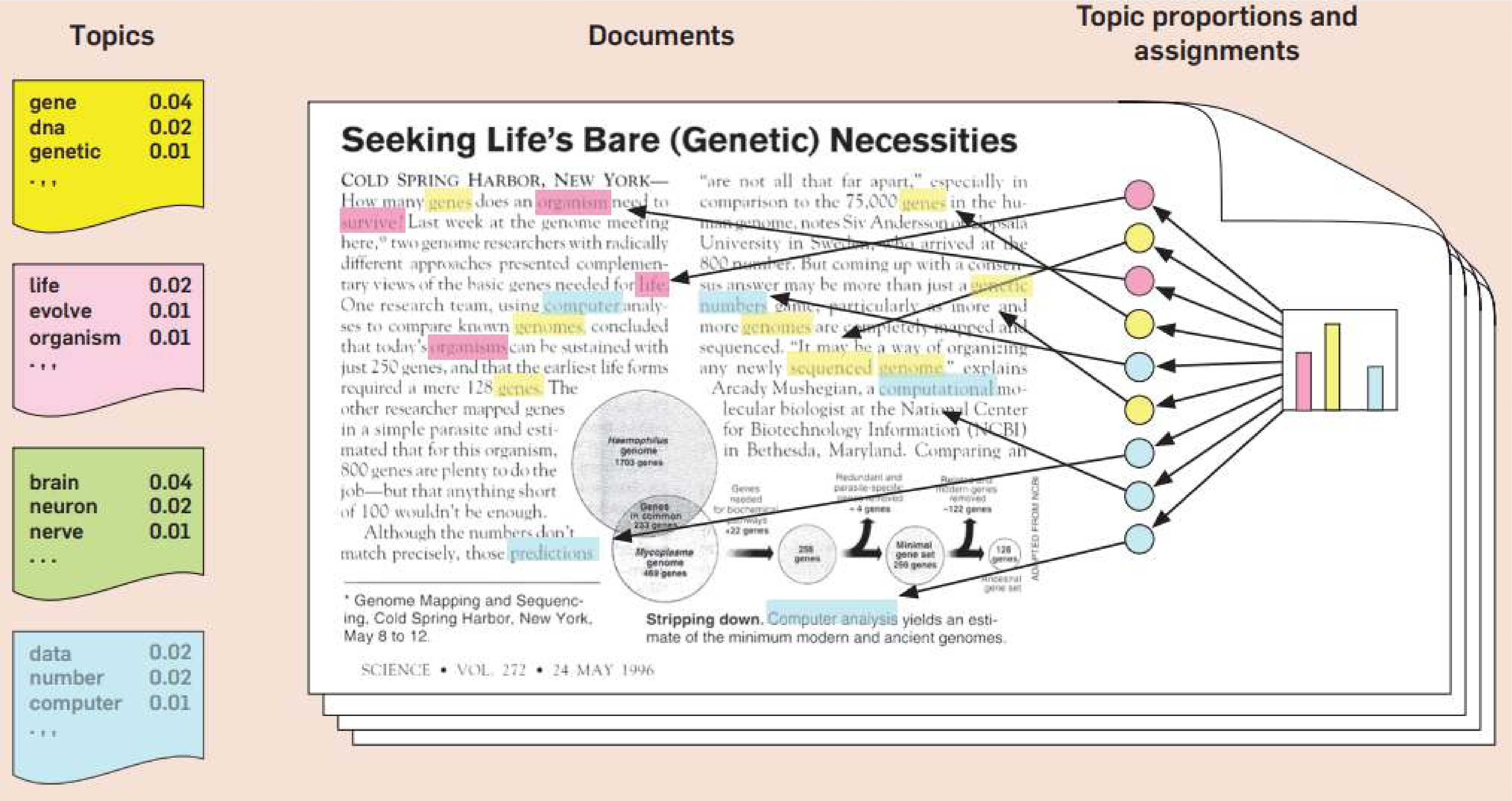
What is LDA?

- **L**atent **D**irichlet **A**llocation
- One of the most popular methods under the field of *topic modeling*
- LDA is a Bayesian method of assessing the content of a document
- LDA assumes there are a set of topics in each document, and that this set follows a *Dirichlet* prior for each document
    - Words within topics also have a *Dirichlet* prior



> More details from the creator

An example of LDA



How does it work?

1. Reads all the documents
    - Calculates counts of each word within the document, tied to a specific ID used across all documents
2. Uses variation in words within and across documents to infer topics
    - By using a Gibbs sampler to simulate the underlying distributions (MCMC method)

- It's a bit complicated mathematically, but it boils down to a system where generating a document follows a couple rules:
    1. Topics in a document follow a multinomial/categorical distribution
    2. Words in a topic follow a multinomial/categorical distribution

What type of Algorithm is LDA?

- Because of the distributional assumptions (which include priors), this is Bayesian
- Because of the way a Gibbs sampler approximates the distributions, this is machine learning

Implementations in R

- There are at least four good implementations of LDA in R
  1. stm: A bit of a tweak on the usual LDA model that plays nicely with quanteda and also has an associated `{stmBrowser}` package for visualization (on Github)
  2. `{lda}`: A somewhat rigid package with difficult setup syntax, but it plays nicely with the great LDAvis package for visualizing models. Supported by quanteda.
  3. `{topicmodels}`: An extensible topic modeling framework that plays nicely with quanteda
  4. mallet: An R package to interface with the venerable MALLET Java package, capable of more advanced topic modeling

Implementing a topic model in STM

```r
# quanteda's conversion for the stm package
out <- convert(tdm, to = 'stm')
# quanteda's conversion for the lda package
# out <- convert(tdm, to = 'lda')
# quanteda's conversion for the topicmodels package
# out <- convert(tdm, to = 'topicmodels')
```

- Creates a list of 3 items:
  - `out$documents`: Index number for each word with count/document
  - `out$vocab`: Words and their index numbers
  - `out$meta` a data frame of information from the corpus (`industry`)

```r
out$documents[[1]][,386:390]
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]
[1,] 23097 23101 23124 23144 23153
[2,]     2     2     1     3    89
```

```r
out$vocab[c(out$documents[[1]][,386:390][1,])]
```

```
[1] "consult"  "consum"   "consumpt" "contamin" "content"
```

Running the model

- We will use the `stm()` function from the stm package
  - It has a lot of options that you can explore to tweak the model
  - The most important is `K`, the number of topics we want. I'll use 10 for simplicity, but often we need more to neatly categorize the text
    - `K=100` is a popular choice when we are using the output of LDA as an input to another model
    - The model we used in Session 6 had `K=31`, as that captures the most restatements in-sample

```r
library(stm)
topics <- stm(out$documents, out$vocab, K=10)
```

> What this looks like while running

LDA model

```r
labelTopics(topics)
```

```
Topic 1 Top Words:
     Highest Prob: 2020-12-31, 2019-12-31, 2020-01-01, 2018-12-31, 2019-01-01, 2018-01-01, decemb
     FREX: nnn:operatingleasememb, vtr:seniorshousingcommunitiesmemb, fcpt:olivegardenmemb,
wpc:realestatesubjecttooperatingleasememb, exc:exelongenerationcollcmemb,
ess:unencumberedapartmentcommunitiesmemb, kim:shoppingcentermemb
     Lift: adc:seniorunsecureddebtmemb, aegco, aep:amortizationofdeferredcostsmemb,
aep:changesinfundedstatusmemb, aep:excessaditthatisnotsubjecttoratenormalizationrequirementsmemb,
aep:ohiopowercomemb, aep:publicservicecoofoklahomamemb
     Score: 2020-12-31, 2019-12-31, 2020-01-01, 2018-12-31, 2019-01-01, 2018-01-01, nnn:operatingleasememb
Topic 2 Top Words:
     Highest Prob: servic, loan, mortgag, exhibit, report, bank, nation
     FREX: corelog, pentalpha, dbtca, dbntc, lnr, ncmslt, cwcapit
     Lift: ikb, -1122, #39, #41, 2013-c10, 2013-c11, 2013-c12
     Score: mortgag, pentalpha, dbtca, dbntc, fargo, cwcapit, corelog
Topic 3 Top Words:
```

- `Highest prob` is a straightforward measure to interpret
  - The words with the highest probability of being chosen in the topic

Applying our topic model to our data

```r
out$meta$industry <- factor(out$meta$industry)

doc_topics = data.frame(document=names(out$documents),
                        industry=out$meta$industry,
                        topic=1,
                        weight=topics$theta[,1])
for (i in 2:10) {
  temp = data.frame(document=names(out$documents),
                    industry=out$meta$industry,
                    topic=i,
                    weight=topics$theta[,i])
  doc_topics = rbind(doc_topics, temp)
}
# Proportional topics (%)
doc_topics <- doc_topics %>%
  group_by(document) %>%
  mutate(topic_prop = weight / sum(weight)) %>%
  ungroup()
```

```r
# Manually label topics
topic_labels = data.frame(topic = 1:10,
  topic_name = c('XBRL', 'Banking', 'Services', 'Equity',
                 'Investment', 'Energy', 'R&D',
                 'Compensation', 'Financial', 'Debt'))

doc_topics <- doc_topics %>% left_join(topic_labels)
```

A nice visualization of our STM model

- Using LDAvis via `{STM}`'s `toLDAvis()` function
  - Need LDAvis and servr installed to run
  - Note: LDAvis scrambles the topic numbers (e.g., topic 1 is LDAvis' topic 9)

```r
# Code to generate LDAvis
toLDAvis(topics, out$documents, R=10)
```

> Click to view

- Using `{stmBrowser}`'s `stmBrowser()` function
  - Install from github, not CRAN

```r
# code to generate stmBrowser
stmBrowser(topics, data=data.frame(text=names(out$documents),
                                   industry=out$meta$industry),
           c('industry'), text='text')
```

> Click to view

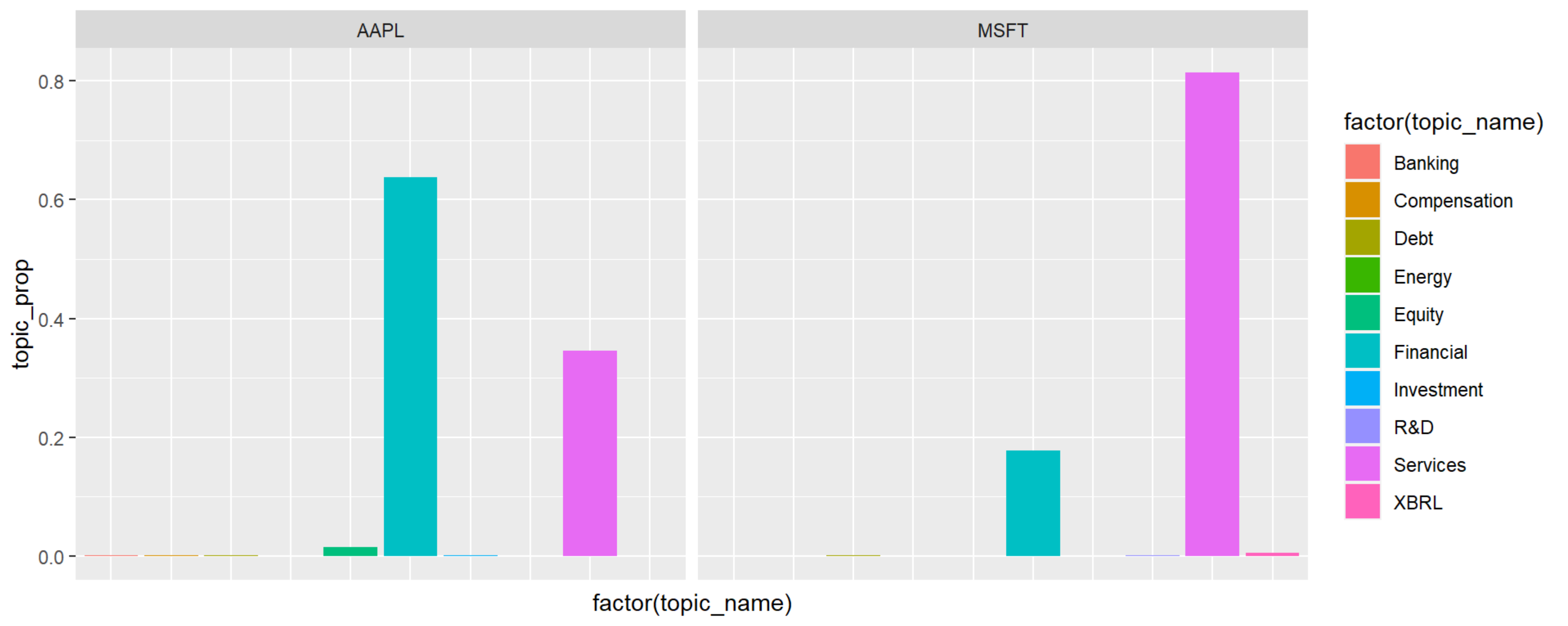
Topic content of the Microsoft 10-K

```r
doc_topics %>% filter(document=='0001564590-21-039151.txt')
```

```
# A tibble: 10 × 6
   document                 industry topic    weight topic_prop topic_name
   <chr>                    <fct>    <dbl>     <dbl>      <dbl> <chr>
 1 0001564590-21-039151.txt Services     1 0.00488    0.00488   XBRL
 2 0001564590-21-039151.txt Services     2 0.0000168  0.0000168 Banking
 3 0001564590-21-039151.txt Services     3 0.814      0.814     Services
 4 0001564590-21-039151.txt Services     4 0.000219   0.000219  Equity
 5 0001564590-21-039151.txt Services     5 0.000164   0.000164  Investment
 6 0001564590-21-039151.txt Services     6 0.0000879  0.0000879 Energy
 7 0001564590-21-039151.txt Services     7 0.00116    0.00116   R&D
 8 0001564590-21-039151.txt Services     8 0.000330   0.000330  Compensation
 9 0001564590-21-039151.txt Services     9 0.177      0.177     Financial
10 0001564590-21-039151.txt Services    10 0.00158    0.00158   Debt
```

Topic content of the Microsoft 10-K versus Apple

```r
doc_topics %>%
  filter(document=='0001564590-21-039151.txt' |
         document=='0000320193-21-000105.txt') %>%
  mutate(Company=ifelse(document=='0001564590-21-039151.txt', 'MSFT','AAPL')) %>%
  ggplot(aes(x=factor(topic_name), y=topic_prop, fill=factor(topic_name))) +
  geom_col() + facet_wrap(~Company) +
  theme(axis.text.x=element_blank(),axis.ticks.x = element_blank())
```



Topic content by industry

```r
doc_topics %>%
  group_by(industry, topic) %>%
  mutate(topic_prop = mean(topic_prop)) %>%
  slice(1) %>%
  ungroup() %>%
  ggplot(aes(x=factor(topic_name), y=topic_prop, fill=factor(topic_name))) +
  geom_col() + facet_wrap(~industry) +
  theme(axis.text.x=element_blank(),axis.ticks.x = element_blank())
```

What we have accomplished?

- We have created a measure of the content of annual reports
  - This gives us some insight as to what is discussed in *any* annual report from 2021 by looking at only 10 numbers as opposed to having to read the whole document
    - We can apply it to other years as well, though it will be a bit less accurate if new content is discussed in those years
  - We can use this measure in a variety of ways
    - Some forecasting related, such as building in firm disclosure into prediction models
    - Some forensics related, such as our model in Session 6

Consider

> How might we leverage LDA (or other topic modeling methods) to improve and simplify analytics?

- What other contexts or data could we use LDA on?
- What other problems can we solve with LDA?

Clustering without known groups

Problem: Classifying companies based on disclosure

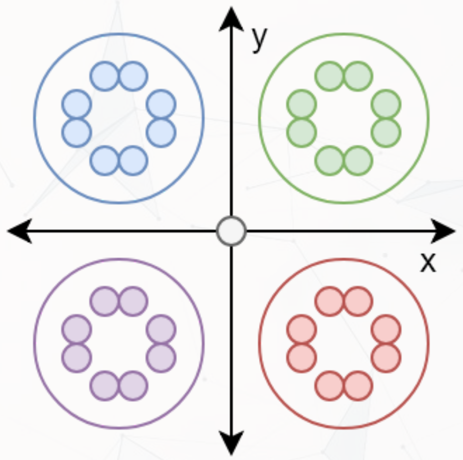
- While industry code is one classification of firms, it has a number of drawbacks:
  1. The classification system is old and perhaps misses new industries
  2. It relies on self-reporting
  3. Firms' classifications rarely change, even when firms themselves change

> We'll build a different classification system, based on what they discuss in their annual reports

Clustering

- One important aspect of detecting anomalies is determining groups in the data
  - We call this *clustering*
- If we find that a few elements of our data don't match the usual groups in the data, we can consider this to be an anomaly
  - Similar to the concept of outliers, but taking into account *multiple variables* simultaneously

- The grey dot is at the mean of both the x and y dimensions
  - it isn't an outlier
- But there are 4 clear clusters… and it doesn't belong to any!



One clustering approach: k-means

$$\min_{C_k} \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

- Minimizes the sum of squared distance between points within groups
- Technically this is a machine learning algorithm, despite its simplicity
- You need to specify the number of groups you want

- Pros:
  - Very fast to run
  - Simple interpretation

- Cons
  - Simple algorithm
  - Need to specify k, the number of clusters

Prepping data

- We will need data to be in a matrix format, with…
  - 1 row for each observation
  - 1 column for each variable we want to cluster by
- Since our data is currently in a long format, we'll recast this with tidyr

```r
library(tidyr)
wide_topics <- spread(doc_topics[,c(1,2,5,6)], topic_name, topic_prop)
# Note: dropping XBRL here
mat <- wide_topics[,3:11]

mat[,1:6] %>% head(n=3) %>% html_df(highlight_cols = c())
```

Banking	Compensation	Debt	Energy	Equity	Financial
0.0000806	0.0007570	0.0045723	0.6573965	0.0012891	0.2155210
0.0000057	0.0000372	0.0000445	0.1259373	0.0000565	0.8653703
0.0372616	0.0004645	0.0083611	0.1996815	0.0501601	0.0380236

Calculating k-means

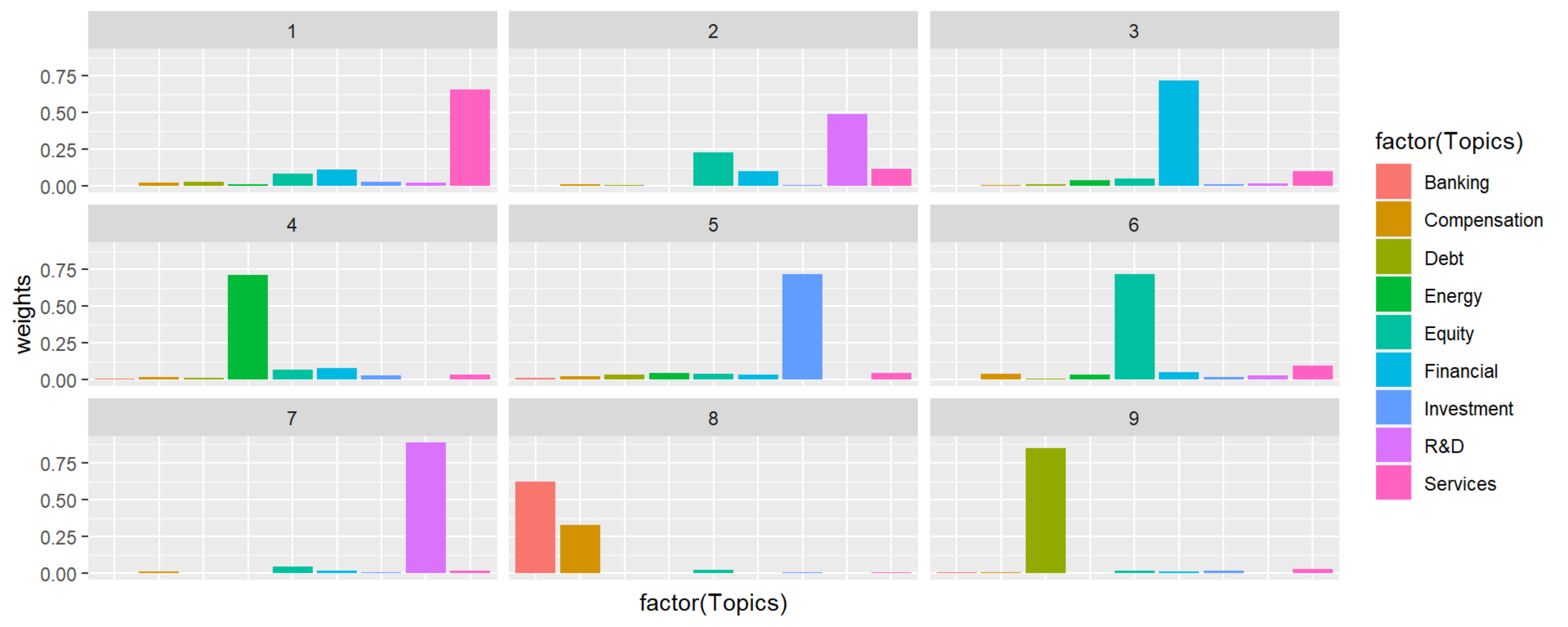
```r
set.seed(6845868)
clusters <- kmeans(mat, 9)

# Add clusters back into our data
wide_topics$kmean <- clusters$cluster

clusters$cluster %>% head()
```

```
[1] 4 3 5 4 3 3
```

- The algorithm tells us group numbers for each observation
- The numbers themselves are arbitrary
  - The clustering (observations sharing a group number) is what matters
- Note: `kmeans()` is built into R – no packages needed

Visualizing the clusters

```r
cbind(as.data.frame(clusters$center), data.frame(kmean=1:9)) %>%
  gather("Topics","weights",-kmean) %>%
  ggplot(aes(x=factor(Topics), y=weights, fill=factor(Topics))) +
  geom_col() +
  facet_wrap(~kmean) +
  theme(axis.text.x=element_blank(),axis.ticks.x = element_blank())
```



Improving our visualization

- There is a relatively new method (2018), UMAP, that is significantly better
  - UMAP stands for **U**niform **M**anifold **A**pproximation and **P**rojection for Dimension Reduction
  - We will use it to reduce 68 dimensions down to 2
  - It is useful for plotting 2 dimensional representations of high dimensional data by maintaining *local* distance structures
    - It also maintains distances *globally*, mostly
  - It is computationally efficient
  - It is based on solid mathematical theory
    - Reimannian manifolds and geodesic distance

There is also t-SNE (**t**-distributed **S**tochastic **N**eighbor **E**mbedding) from 2008, but it is inferior for 2 reasons: 1) it is more computationally costly than UMAP and 2) it is a bit misleading, as it only maintains distance locally *but not globally*. There is an even more outdated method (PCA), which struggles on higher dimensional data like our 10 topics.

Implementing UMAP

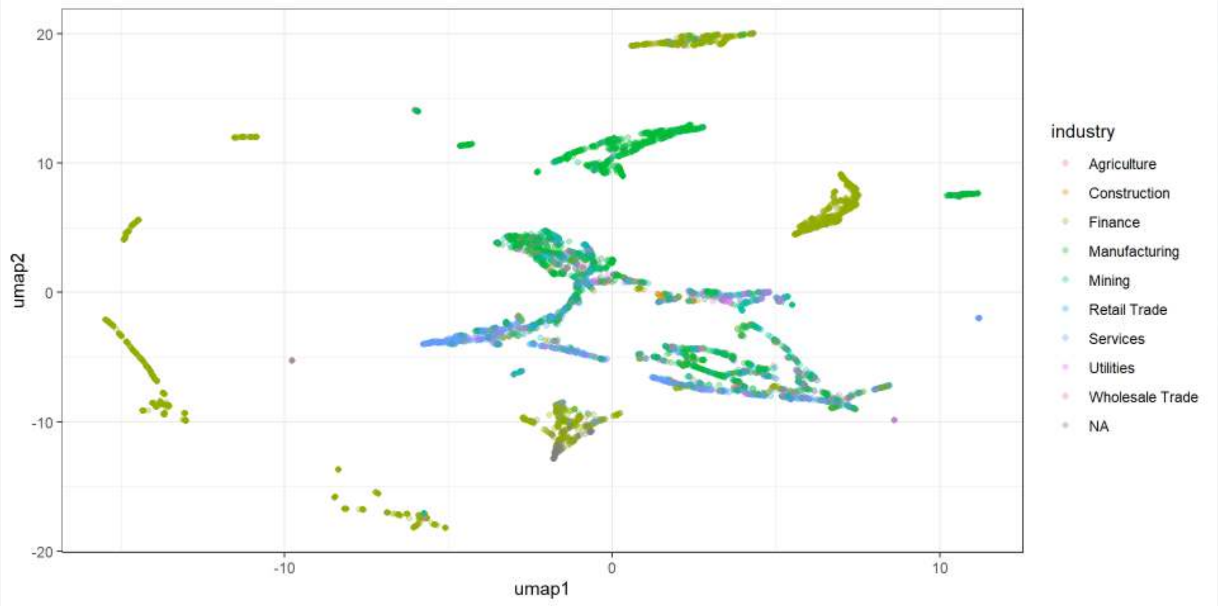
```r
library(uwot)

# Build the UMAP model
umap_train <- umap(mat, ret_model = TRUE)
# Extract coordinates
umap_coords <- umap_train$embedding %>% as.data.frame()
colnames(umap_coords) <- c('umap1', 'umap2')
# Merge coordinates into our data frame
wide_topics <- cbind(wide_topics, umap_coords)
```

- We will use the uwot package to implement UMAP
- Our goal is to extract the locations that each document should be placed at in a 2D space
- The `umap()` function builds the model
- The `umap_train$embedding` object contains the needed coordinates
- Then we just add these back into our data

Visualizing with UMAP: k-means

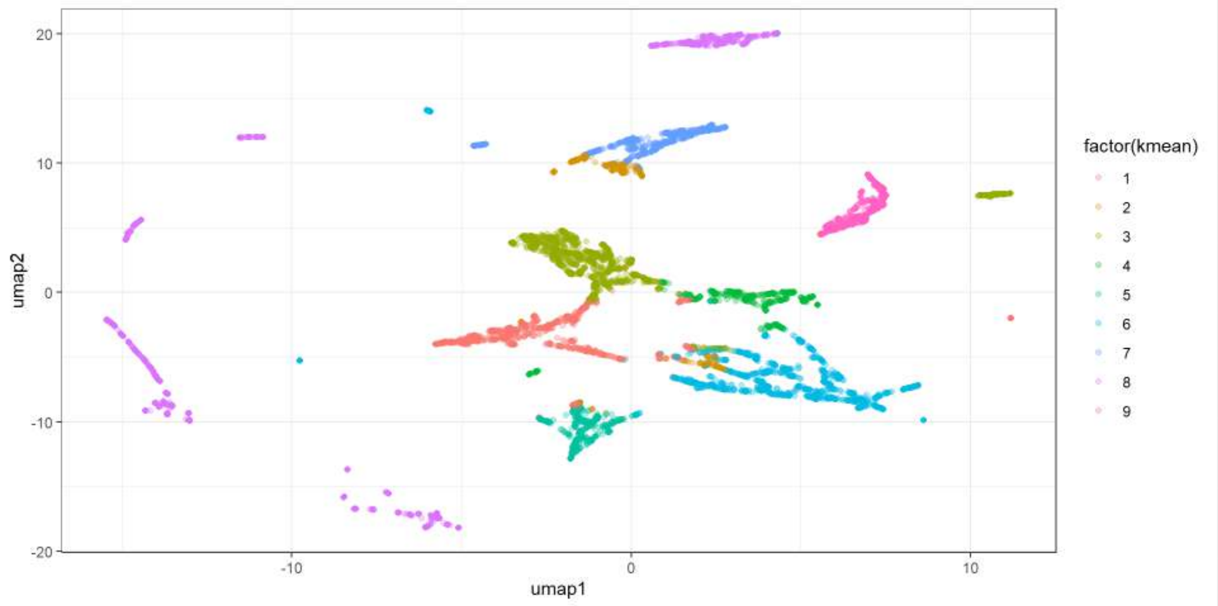
Colored by SIC codes

```r
ggplot(wide_topics,
       aes(x = umap1, y = umap2,
           color = industry)) +
  geom_point(alpha = 0.3) +
  theme_bw()
```



Colored by kmeans

```r
ggplot(wide_topics,
       aes(x = umap1, y = umap2,
           color = factor(kmean))) +
  geom_point(alpha = 0.3) +
  theme_bw()
```

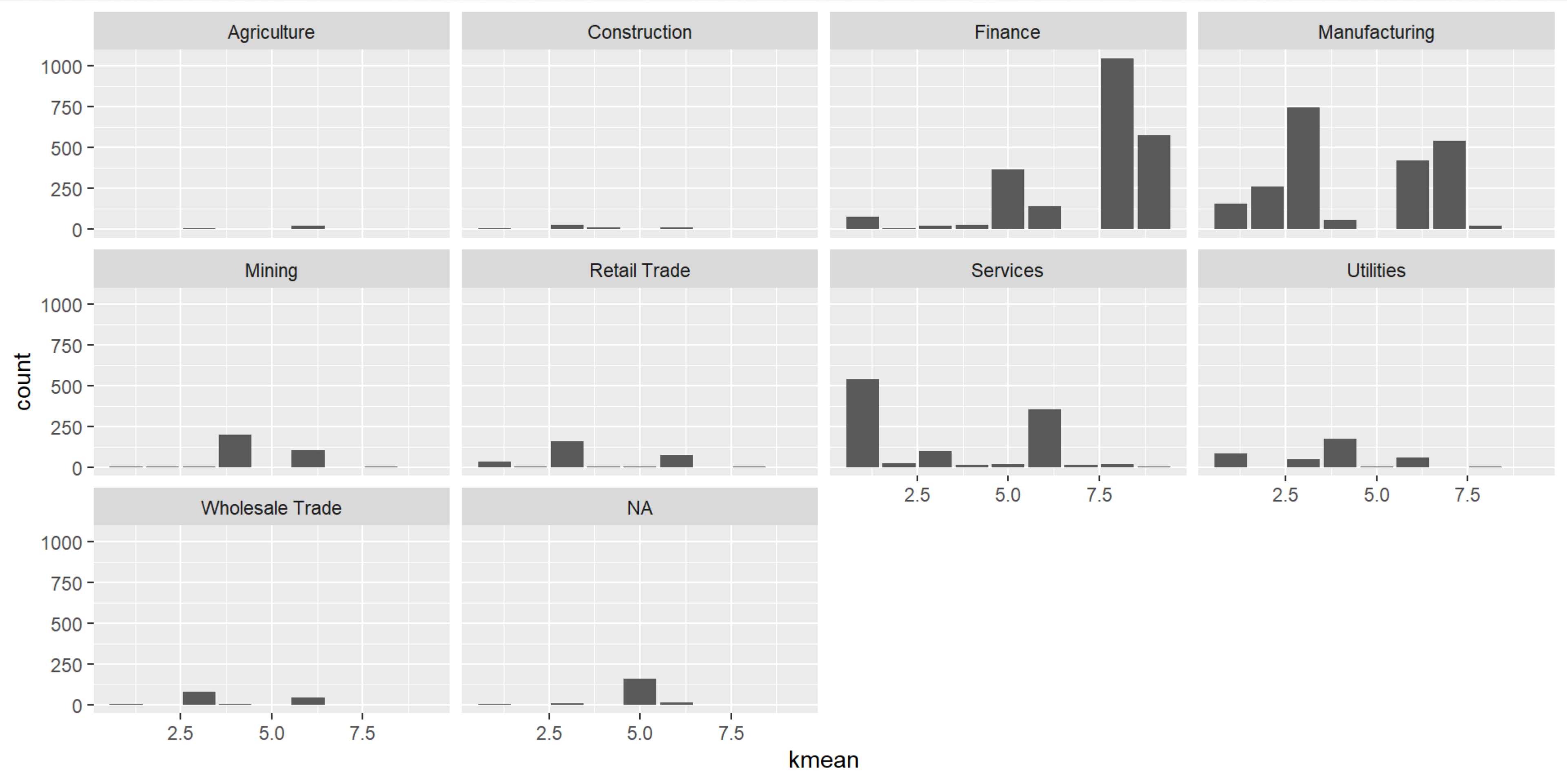


Why are these graphs different?

- Possibly due to…
  - Data: 10-K disclosure content doesn't fully capture industry inclusion
  - LDA: The measure is noisy – it needs more data
  - SIC code: The measure doesn't cleanly capture industry inclusion
    - Some firms are essentially misclassified
- Recall, SIC covers Agriculture, Forestry and Fishing; Mining; Construction; Manufacturing; Transportation, Communications, Electric, Gas, and Sanitary Services; Wholesale Trade; Retail Trade; Finance, Insurance, and Real Estate; Services; Public Administration

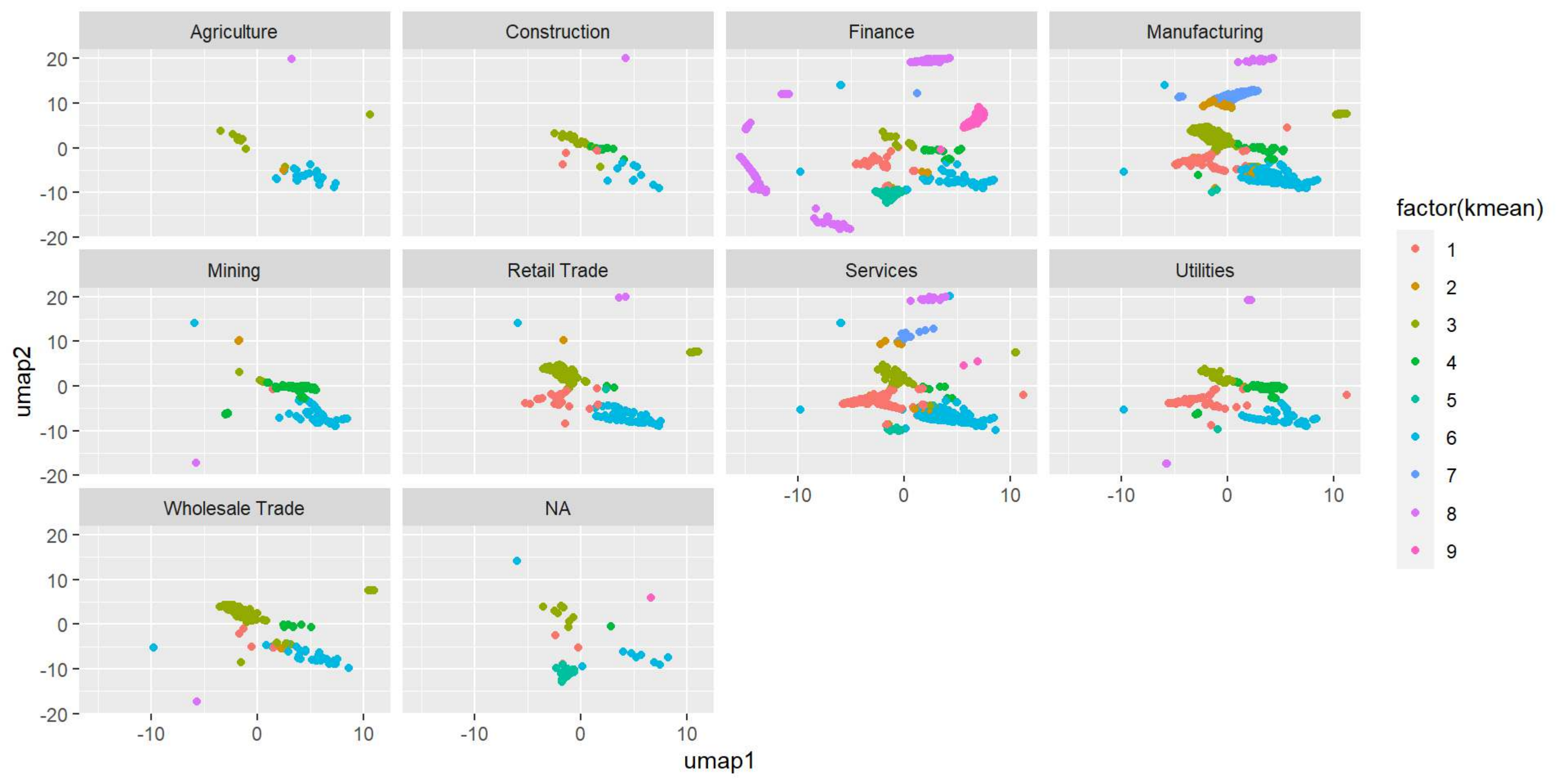
How related are clusters and industries?

```r
ggplot(wide_topics, aes(x=kmean)) + geom_bar() + facet_wrap(~factor(industry))
```



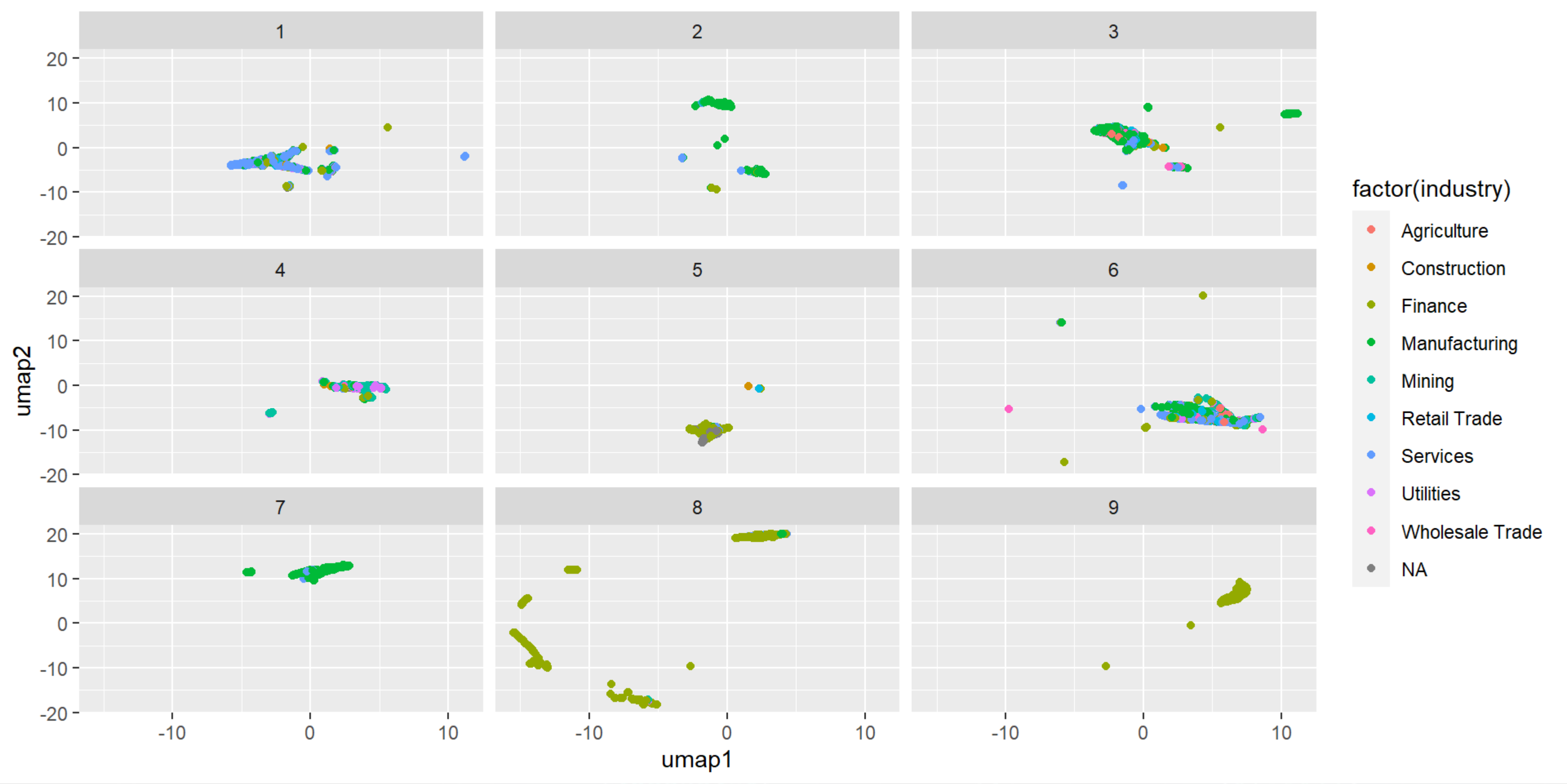
How related are clusters and industries?

```r
ggplot(wide_topics, aes(x=umap1, y=umap2, color=factor(kmean))) + geom_point() +
  facet_wrap(~factor(industry))
```



How related are clusters and industries?

```r
ggplot(wide_topics, aes(x=umap1, y=umap2, color=factor(industry))) + geom_point() +
  facet_wrap(~factor(kmean))
```



Looking for anomalies

Looking for anomalies

- k-means minimizes the distance from a central point
- We can look for the firms that are farthest from said point!

```r
wide_topics$dist <- sqrt(rowSums(abs(mat - fitted(clusters))))  # Distance from center
wide_topics[,c(1,2,4,8,16)] %>% arrange(desc(dist)) %>% slice(1:5) %>% html_df()
```

document	industry	Compensation	Financial	dist
0001104659-21-044134.txt	Finance	0.9995782	2.8e-06	1.156605
0001104659-21-043939.txt	Finance	0.9995650	3.4e-06	1.156594
0001140361-21-010304.txt	Finance	0.9995592	3.7e-06	1.156589
0001193125-21-098450.txt	Services	0.9995478	3.4e-06	1.156579
0001193125-21-102380.txt	Finance	0.9995064	3.9e-06	1.156544

> We are a blank check company incorporated on ________ as a Cayman Islands exempted company for the purpose of effecting a merger, share exchange, asset acquisition, share purchase, reorganization or similar business combination with one or more businesses or entities (a “Business Combination”).
>
> — **All 5 files…**

- They are used for SPACs (e.g., Grab)

Looking for anomalies (ignoring finance firms)

```r
wide_topics[,c(1,2,4,8,16)] %>%
  filter(industry!="Finance") %>%
  arrange(desc(dist)) %>%
  mutate(id=1:n()) %>%
  select(id,everything()) %>%
  slice(1:7) %>%
  html_df()
```

id	document	industry	Compensation	Financial	dist
1	0001193125-21-098450.txt	Services	0.9995478	3.40e-06	1.156579
2	0001193125-21-092793.txt	Manufacturing	0.9988654	1.19e-05	1.155990
3	0001193125-21-100874.txt	Services	0.9963810	1.17e-05	1.153839
4	0001140361-21-010411.txt	Services	0.9778598	2.05e-05	1.153371
5	0001104659-21-031725.txt	Manufacturing	0.9770098	2.31e-05	1.152625
6	0001213900-21-013228.txt	Manufacturing	0.9944066	5.28e-05	1.152517
7	0001213900-21-010315.txt	Services	0.9941140	4.19e-05	1.151873

- All: Yet more SPACs, just with the wrong industry in their filings…
- How many SPACs are there?

```r
wide_topics[,c(1,2,4,8,16)] %>% filter(Compensation > 0.9, dist > 1.1) %>% nrow()
```

```
[1] 307
```

Looking for anomalies (ignoring high compensation discussion)

```r
wide_topics[,c(1,2,4,8,16)] %>%
  filter(industry!="Finance", Compensation < 0.5) %>%
  arrange(desc(dist)) %>%
  mutate(id=1:n()) %>%
  select(id,everything()) %>%
  slice(1,2,3,8,9,10) %>%
  html_df()
```

id	document	industry	Compensation	Financial	dist
1	0001731122-21-000373.txt	Construction	0.4180281	0.0405295	1.0988544
2	0001628280-21-000722.txt	Construction	0.0004643	0.2485153	1.0749207
3	0001654954-21-004244.txt	Services	0.0120396	0.2998639	1.0438606
8	0001410578-21-000612.txt	Construction	0.0092240	0.3515138	1.0067509
9	0001564590-21-009825.txt	Services	0.0003849	0.1270724	0.9992496
10	0001712923-21-000017.txt	Services	0.0101671	0.0004910	0.9990207

- 1: Sustainable homebuilder
- 2: Largest US homebuilder (4-7 are similar companies)
- 3: A bankrupt, regional lessor of 12 aircraft
- 8: Contracting services for automotive and energy firms; data center operation
- 9: A timeshare firm spun off from Hilton
- 10: A complex IPO-related entity with no actual operations

What we have accomplished

- We have created a classification of firms into discrete groups based on their disclosure content of their 10-K filings
  - The classification accounts for how similar each firm's content is to other firms' content
- We have used this classification to identify 10 firms which have non-standard accounting disclosures for their SIC code classification

> Text based industry classification using 10-Ks has been shown to be quite viable, such as in work by Hoberg and Phillips.

Consider

> What else could we use clustering to solve?

- Where in business would we like to group something, but we don’t know the groups?

Filling in missing data

Problem: Missing data

- You may have noticed that some of the `industry` measure was `NA`
- What if we want to assign an industry to these firms based on the content of their 10-K filings?

Using k-means

- One possible approach we could use is to fill based on the category assigned by k-means
- However, as we saw, k-means and SIC code don’t line up perfectly…
  - So using this classification will definitely be noisy

A better approach with KNN

- KNN, or **K**-**N**earest **N**eighbors is a *supervised* approach to clustering
- Since we already have industry classifications for most of our data, we can use that structure to inform our assignment of the missing industry codes
- The way the model uses the information is by letting the nearest labeled points “vote” on what the point should be
  - Points are defined by 10-K content in our case

Implementing KNN in R

- We'll use the caret package for this, as it will allow us to use k-fold cross validation to select a model
  - The same technique we used for LASSO and xgboost

```r
train <- wide_topics %>% filter(!is.na(industry))
label <- wide_topics %>% filter(is.na(industry))
```

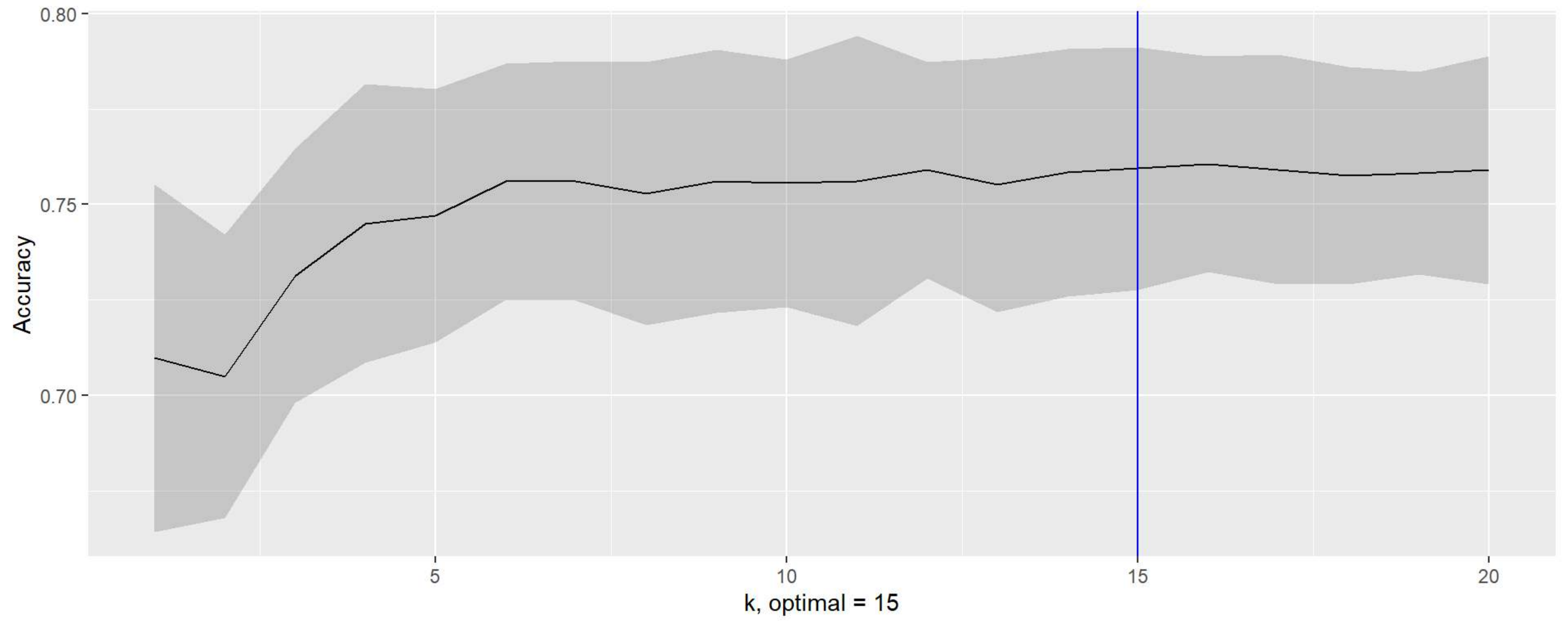
```r
library(caret)
trControl <- trainControl(method='cv', number=20)
tout <- train(industry ~ .,
      method = 'knn',
      tuneGrid = expand.grid(k=1:20),
      trControl = trControl,
      metric = "Accuracy",
      data = train[,c(2:11)])
```

Implementing KNN in R

```
tout
k-Nearest Neighbors

6742 samples
   9 predictor
   9 classes: 'Agriculture', 'Construction', 'Finance', 'Manufacturing', 'Mining', 'Retail Trade', 'Services',
'Utilities', 'Wholesale Trade'

No pre-processing
Resampling: Cross-Validated (20 fold)
Summary of sample sizes: 6403, 6404, 6406, 6404, 6405, 6407, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.7097283  0.6108027
   2  0.7049892  0.6046680
```

KNN performance as we increase k

```r
ggplot(tout$results, aes(x=k, y=Accuracy)) +
  geom_line() +
  geom_ribbon(aes(ymin=Accuracy - AccuracySD*1.96,
                  ymax=Accuracy + AccuracySD*1.96), alpha=0.2) +
  geom_vline(xintercept=15, color="blue") +
  xlab("k, optimal = 15")
```



Using KNN to fill in industry

1. CAPITAL SOUTHWEST CORP: “closed-end, non-diversified investment company”
   - SIC missing, but clearly finance ✓
2. Rayonier Inc: It is a timberland REIT, but it used to be a paper manufacturer
   - SIC is 6798 (finnace) for 1 entity, missing for another, but clearly finance x
3. AMERIPRISE CERTIFICATE COMPANY: Financial certificate firm
   - SIC missing, but clearly finance ✓
4. Callaway Golf: Golf equipment
   - SIC 3949 (in manufacturing) ✓
5. Quest Management, Inc.: No operations, but used to do marketing for fitness equipment
   - No SIC, but it would fall under services ✓
6. MSC Income Fund: “Closed end management investment company”
   - SIC missing, but clearly finance ✓

```r
label$industry_pred <- predict(tout,
                               label)
label[,c("document",
         "industry_pred")] %>%
  head %>% html_df
```

document	industry_pred
0000017313-21-000075.txt	Finance
0000052827-21-000035.txt	Manufacturing
0000820027-21-000014.txt	Finance
0000837465-21-000003.txt	Manufacturing
0001017386-21-000166.txt	Services
0001047469-21-000783.txt	Finance

Recap

Today, we:

1. Processed a set of 6,933 annual reports from 2021 to examine their readability
2. Examined the content discussed in annual reports in 2021
3. Examined the natural groupings of content across firms
    - This doesn’t necessarily match up well with SIC codes
    - There are some firms that don’t quite fit with others in their industry (as we algorithmically identified)
4. Filled in missing industry data using KNN, and were correct in 5 of 6 checked entries ✓

End Matter

Wrap up

- Keep working on the project – you have a lot of tools you can use already
  - And you will learn 1 more next week!
- Survey on the class session at this QR code:

Packages used for these slides

- caret
- cluster
- DT
- downlit
- kableExtra
- knitr
- quanteda and stopwords
- quarto
- readtext
- revealjs
- stm and `{stmBrowser}`
- tidyr
- tidyverse
- uwot