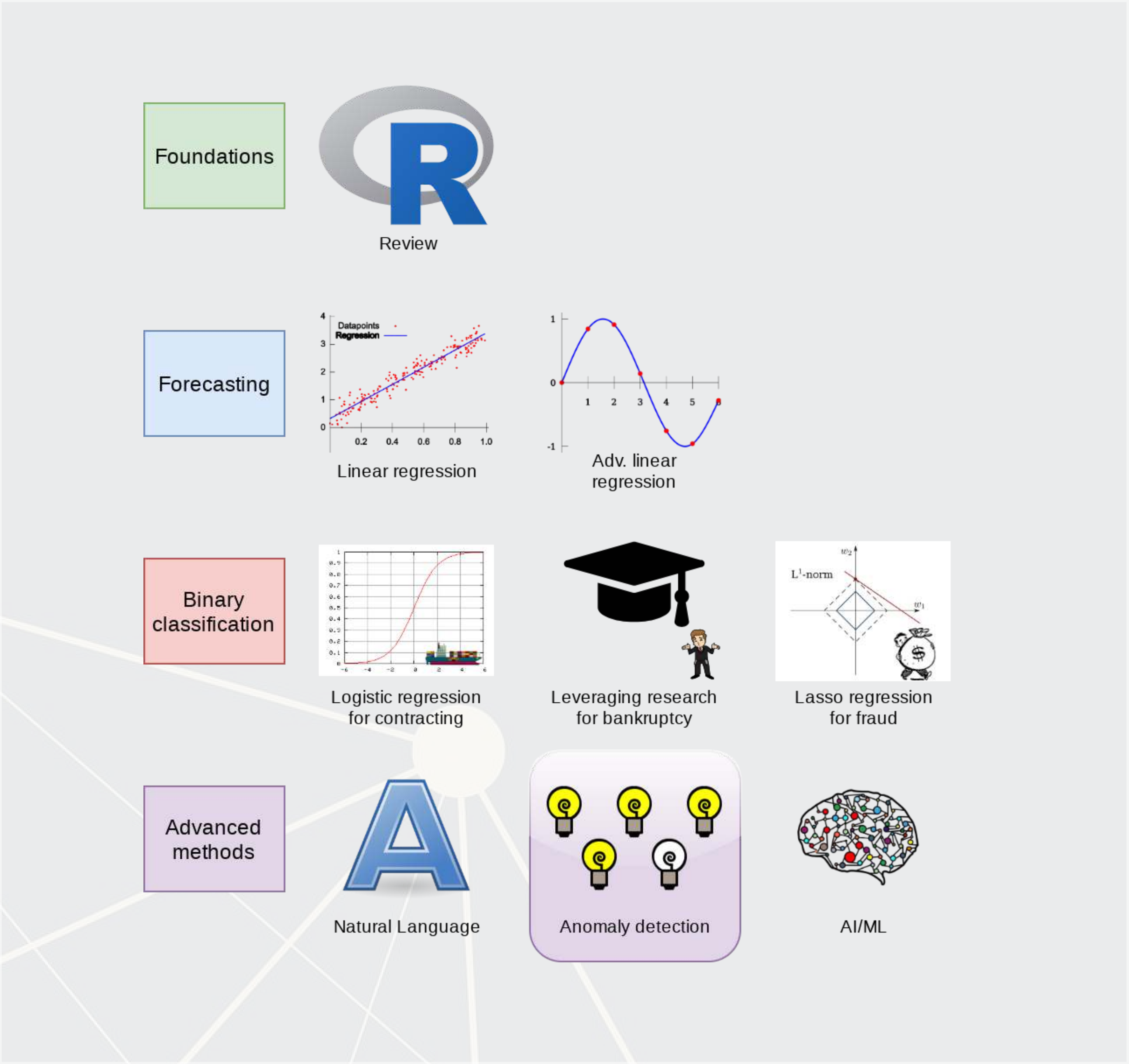
ACCT 420: Topic modeling and anomaly detection

Session 8

Dr. Richard M. Crowley
rcrowley@smu.edu.sg
http://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)
  - ML/AI (k-means, t-SNE)
  - More ML/AI (KNN)

Datacamp

- One last chapter: What is Machine Learning
  - Just the first chapter is required
  - You are welcome to do more, of course
- This is the last *required* chapter on Datacamp

Group project

> For reading large files, `readr` is your friend

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

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

Group project

For saving intermediary results, `saveRDS()` + `readRDS()` is 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

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 6,000 annual reports from 2014
- Other options include using
  - `purrr` and `df_map()`
  - `tm` and `VCorpus()`
  - `textreadr` and `read_dir()`

```r
library(readtext)
library(quanteda)
# Needs ~1.5GB
corp <- corpus(readtext("/media/Scratch/Data/Parser2/10-K/2014/*.txt"))
```

Corpus summary

```r
summary(corp)
```

```
##                          Text Types Tokens Sentences
## 1   0000002178-14-000010.txt  2929  22450       798
## 2   0000003499-14-000005.txt  2710  23907       769
## 3   0000003570-14-000031.txt  3866  55142      1541
## 4   0000004187-14-000020.txt  2902  26959       934
## 5   0000004457-14-000036.txt  3050  23941       883
## 6   0000004904-14-000019.txt  3408  30358      1119
## 7   0000004904-14-000029.txt   370   1308        40
## 8   0000004904-14-000031.txt   362   1302        45
## 9   0000004904-14-000034.txt   358   1201        42
## 10  0000004904-14-000037.txt   367   1269        45
## 11  0000004977-14-000052.txt  4859  73718      2457
## 12  0000005513-14-000008.txt  5316  91413      2918
## 13  0000006201-14-000004.txt  5377 113072      3437
## 14  0000006845-14-000009.txt  3232  28186       981
## 15  0000007039-14-000002.txt  2977  19710       697
## 16  0000007084-14-000011.txt  3912  46631      1531
## 17  0000007332-14-000004.txt  4802  58263      1766
## 18  0000008868-14-000013.txt  4252  62537      1944
## 19  0000008947-14-000068.txt  2904  26081       881
## 20  0000009092-14-000004.txt  3033  25204       896
```

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
0000002178-14-000010.txt	21.03917
0000003499-14-000005.txt	20.36549
0000003570-14-000031.txt	22.24386
0000004187-14-000020.txt	18.75720
0000004457-14-000036.txt	19.22683
0000004904-14-000019.txt	20.51594

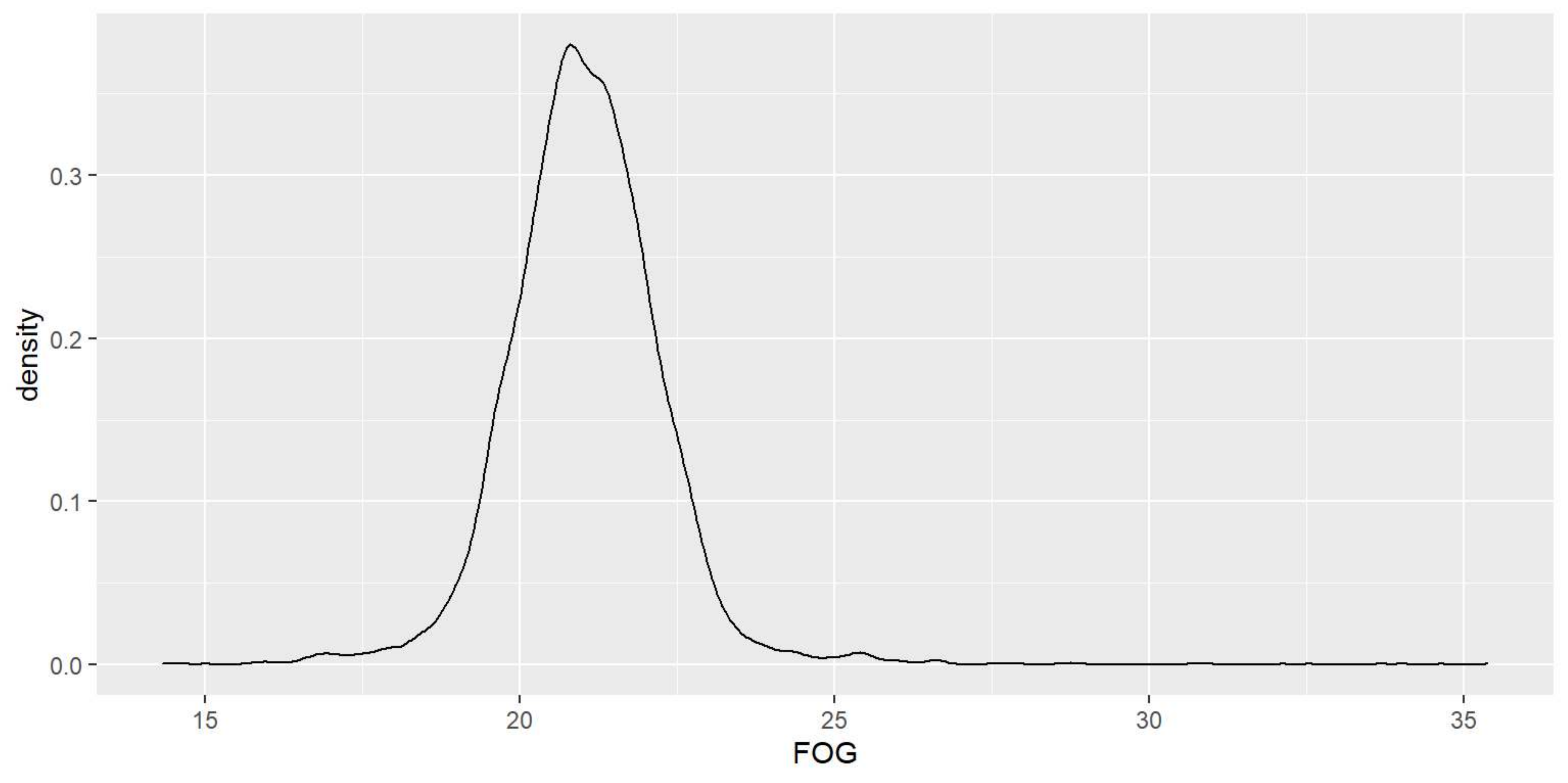
Recall that Citi's annual report had a Fog index of 21.63

Readability across documents

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.33   20.32   21.01   21.05   21.75   35.37
```

```r
ggplot(corp_FOG, aes(x=FOG)) + geom_density()
```



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: Citigroup is SIC 6021
  - 60: Depository institution
  - 602: Commercial bank
  - 6021: National commercial bank

Are certain industries' filings more readable?

- Merge in SIC code by group

```r
df_SIC <- read.csv('../../Data/Filings2014.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()
corp_FOG <- corp_FOG %>% left_join(df_SIC)
```

```
## Joining, by = "document"
```

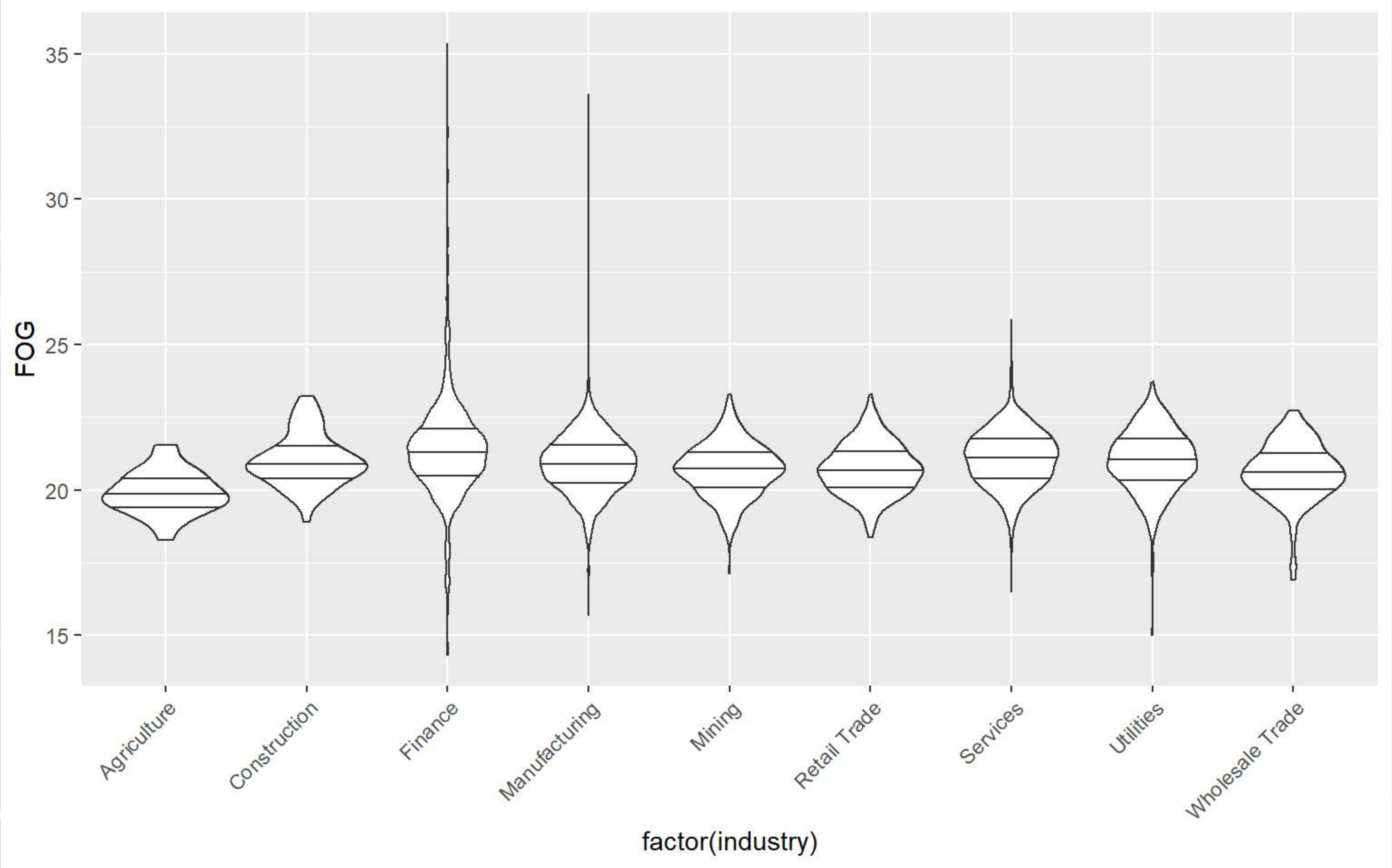
Are certain industries’ filings more readable?

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

document	FOG	regsic	industry
0000002178-14-000010.txt	21.03917	5172	Wholesale Trade
0000003499-14-000005.txt	20.36549	6798	Finance
0000003570-14-000031.txt	22.24386	4924	Utilities
0000004187-14-000020.txt	18.75720	4950	Utilities
0000004457-14-000036.txt	19.22683	7510	Services
0000004904-14-000019.txt	20.51594	4911	Utilities

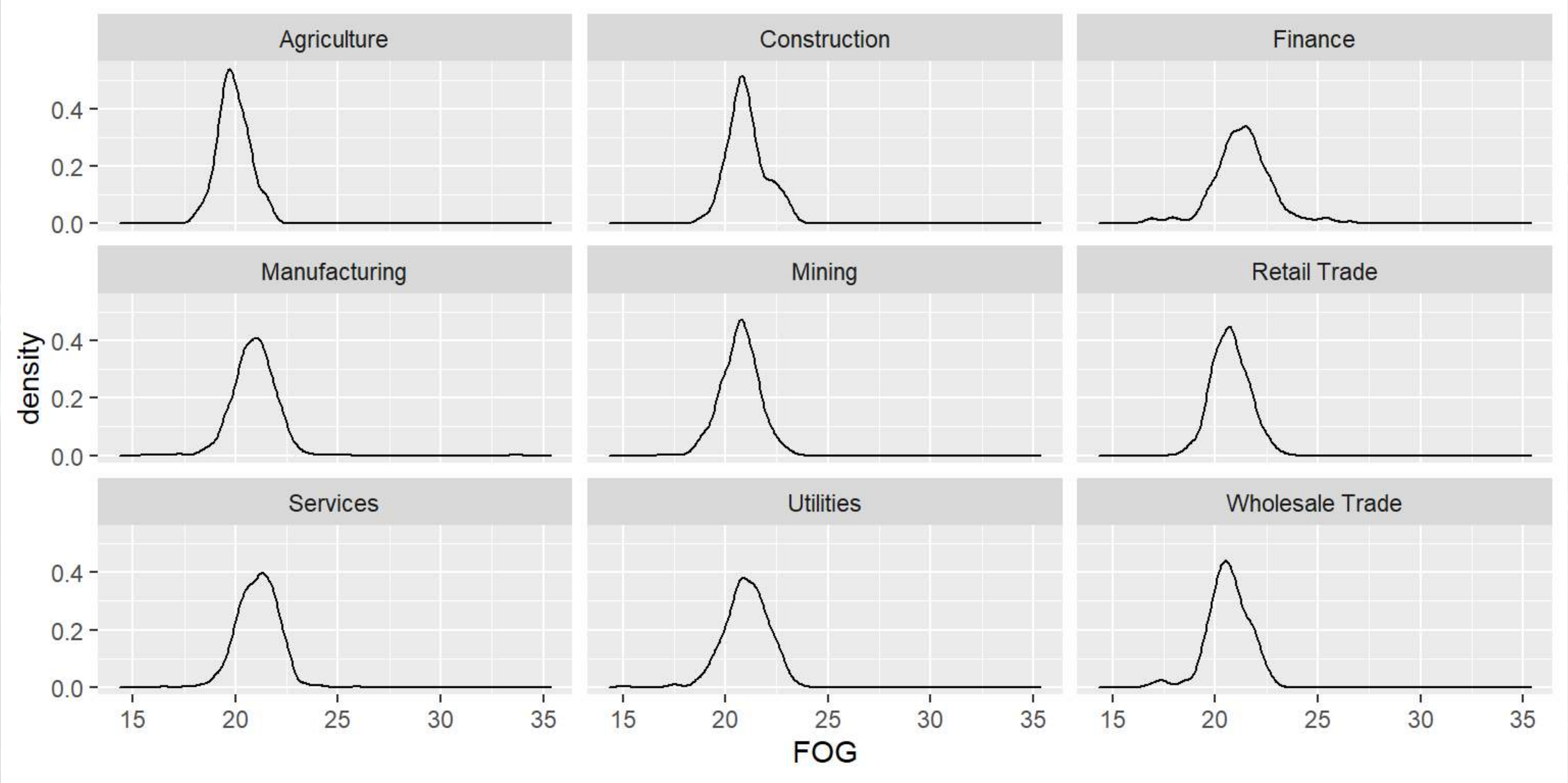
Are certain industries’ filings more readable?

```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))
```



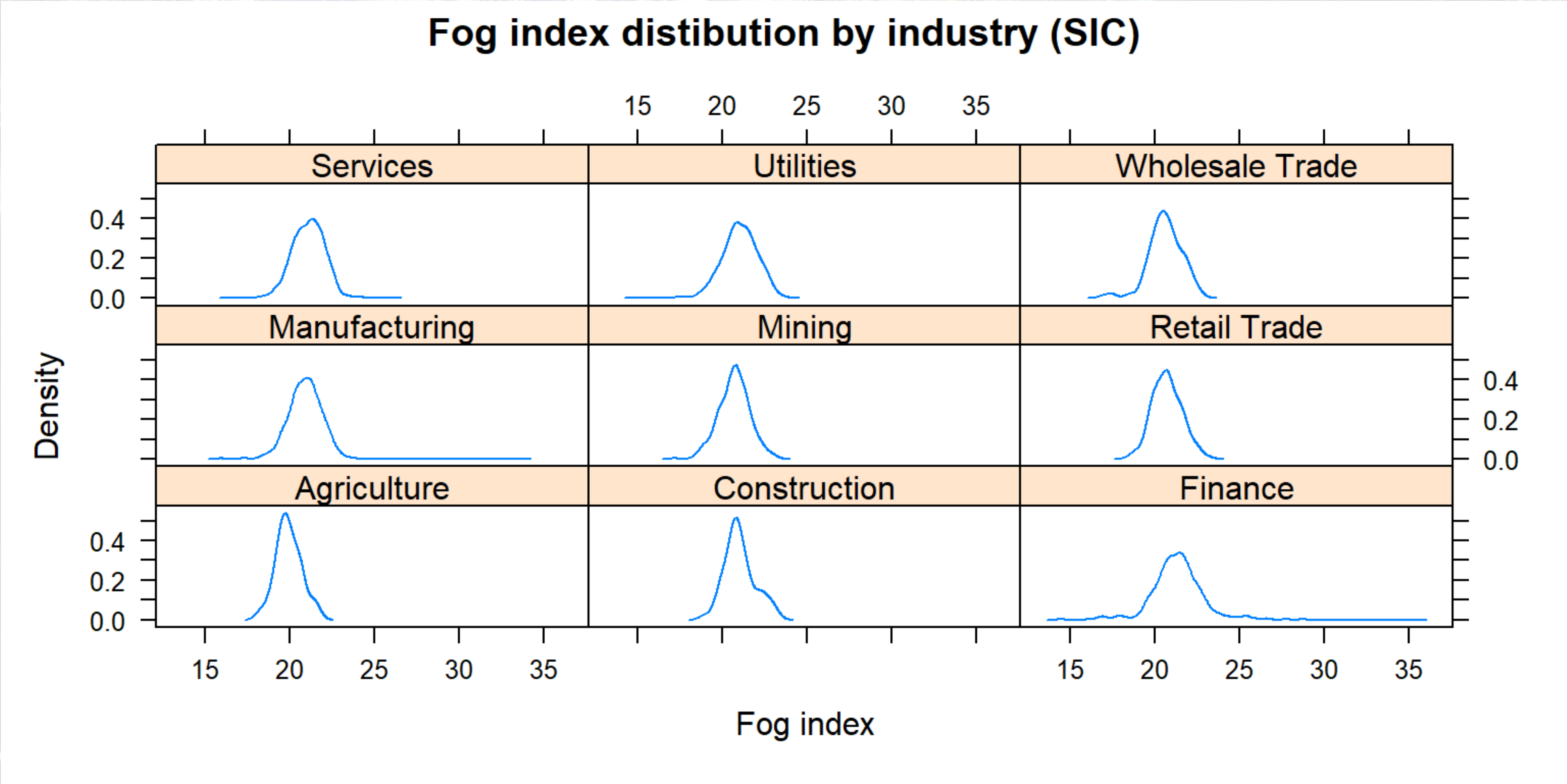
Are certain industries' filings more readable?

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



Are certain industries' filings more readable?

```r
library(lattice)
densityplot(~FOG | industry,
            data=corp_FOG,
            plot.points=F,
            main="Fog index distibution by industry (SIC)",
            xlab="Fog index",
            layout=c(3,3))
```



`quanteda` bonus: Finding references across text

```r
kwic(corp, phrase("global warming")) %>% mutate(text=paste(pre,keyword,post)) %>%
  select(docname, text) %>% datatable(options = list(pageLength = 5), rownames=F)
```

Show entries

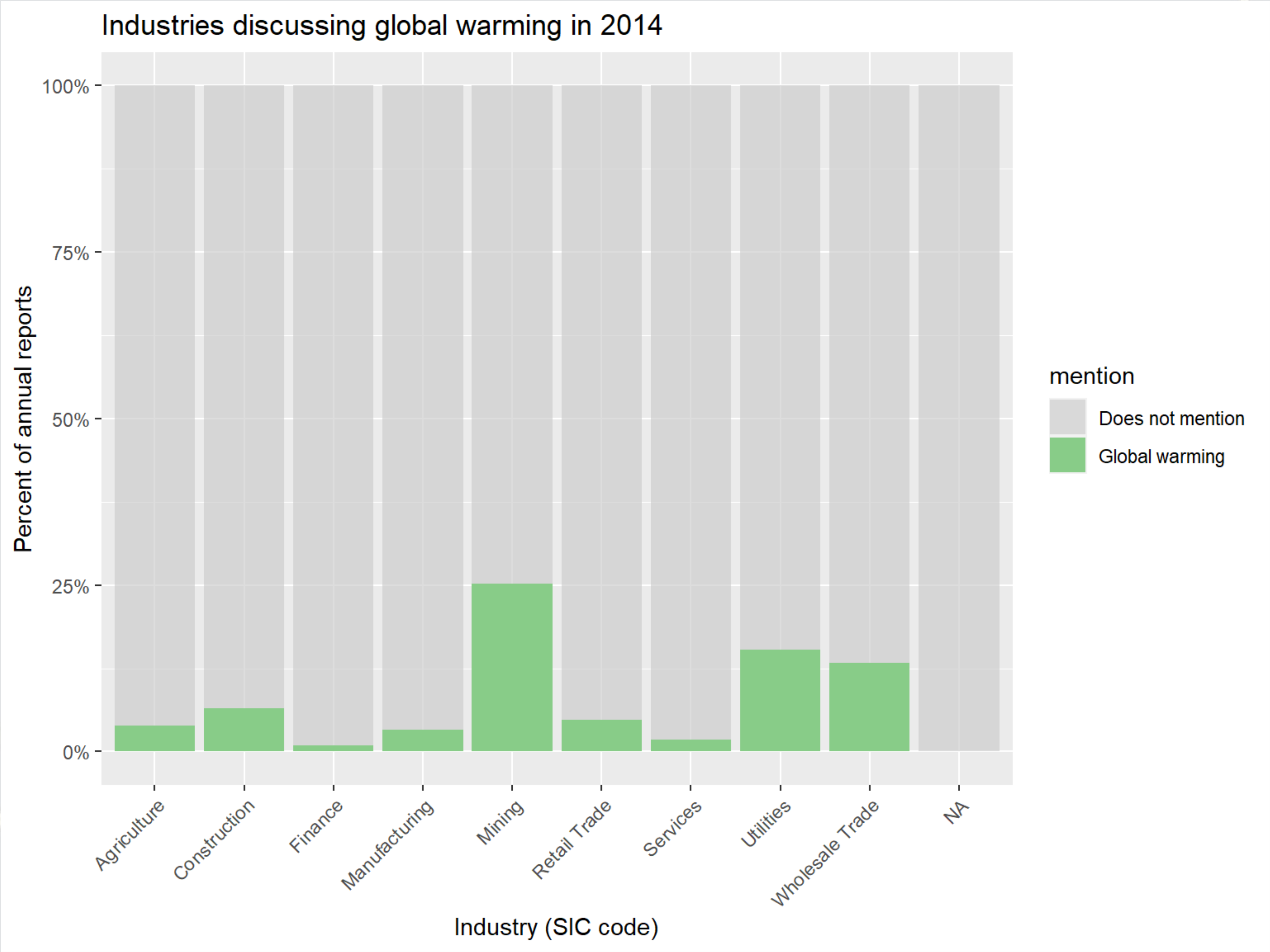
Search:

docname	industry	text
0000003499-14-000005.txt	Finance	. Potentially adverse consequences of global warming could similarly have an impact
0000004904-14-000019.txt	Utilities	nuisance due to impacts of global warming and climate change . The
0000008947-14-000068.txt	Manufacturing	timing or impact from potential global warming and other natural disasters ,
0000029915-14-000010.txt	Manufacturing	human activities are contributing to global warming . At this point ,
0000029915-14-000010.txt	Manufacturing	probability and opportunity of a global warming trend on UCC specifically .

Showing 1 to 5 of 310 entries

Previous 1 2 3 4 5 … 62 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 104,690,796 words in it!
  - 69.8 hours for the “world’s fastest reader”, per this source
  - 103.86 **days** for a standard speed reader (700wpm)
  - 290.8 days for an average reader (250wpm)
- We could read a small sample of them?
- Or… 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 options:
    - `stem=TRUE`, Code similar words as the same
      - Ex.: *cod*e, *cod*ing, and *cod*er would all become *cod*
        - Helps with the curse of dimensionality
    - `remove=c(...)`, You can supply a list of stop words to remove
      - You can use `remove=stopwords()` for a simple list
      - The `stopwords()` function is provided by the `stopwords` package, and actually supports over 50 languages, including Chinese, English, Hindi, and Malay
      - We can use SMART like last week: `remove=stopwords(source='smart')`
      - For other languages, use `remove=stopwords("zh", source="stopwords-iso")`

Making a TDM

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

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

# With stopwords
tdm <- dfm(corp,
           remove=stopwords(source='smart'))

# With stopwords and stemming -> Used in next slides
# 2.6B elements in the matrix
tdm <- dfm(corp,
           stem=TRUE,
           remove=stopwords(source='smart'),
           remove_punct=TRUE,
           remove_numbers=TRUE) %>%
  dfm_trim(min_termfreq=10, termfreq_type = "count")
```

What words matter by industry?

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

```
## $`Wholesale Trade`
## compani    oper million financi product
##   30371   20340   18085   17552   17300
##
## $Finance
## compani    loan financi  decemb million
##  438185  392164  299978  286791  274376
##
## $Utilities
##    oper million compani financi  includ
##  112038  107322  101971   79010   76604
##
## $Services
## compani    oper million financi  servic
##  222276  145506  138397  131881  120817
##
## $Manufacturing
## compani product million    oper financi
##  434805  368900  275829  240181  231687
##
## $Mining
```

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
  - `quanteda`'s default options are a bit odd

The actual TF-IDF equation we'll use

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

- 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")
```

```
## $`Wholesale Trade`
##   graybar  grainger       oil   million     bottl
## 0.3140485 0.2899255 0.2187512 0.2184815 0.2122642
##
## $Finance
##        ab   mortgag depositor      loan      reit
##  9.863862  7.414096  6.192815  5.109854  5.046502
##
## $Utilities
##      gas      fcc  pipelin   energi aircraft
## 2.005220 1.484092 1.227766 1.164767 1.020255
##
## $Services
##     game   client   casino  million  softwar
## 2.394468 1.760647 1.635549 1.496073 1.404740
##
## $Manufacturing
##   clinic      fda    trial     drug  patient
## 7.057913 5.487707 3.949705 3.935010 3.799611
##
## $Mining
```

These are more meaningful

What words matter for banks?

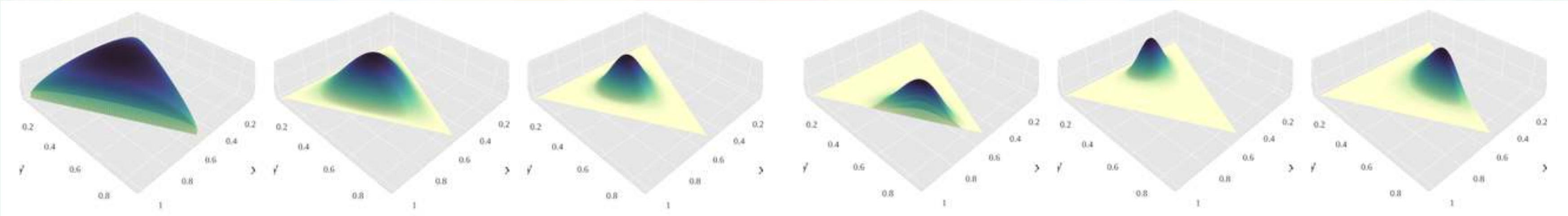
```r
topfeatures(tfidf_mat, n=20, groups="industry")$Finance
```

```
##          ab     mortgag   depositor        loan        reit       trust
##    9.863862    7.414096    6.192815    5.109854    5.046502    4.394811
##     reinsur      truste       estat      tenant    instruct partnership
##    3.809024    3.607591    3.188824    3.100092    2.970419    2.697215
##        real     million        pool        fdic   residenti    bancorp
##    2.506670    2.482285    2.287610    2.238533    2.149133    2.074819
##     obligor        rmbs
##    2.055811    2.055453
```

Moving on to LDA

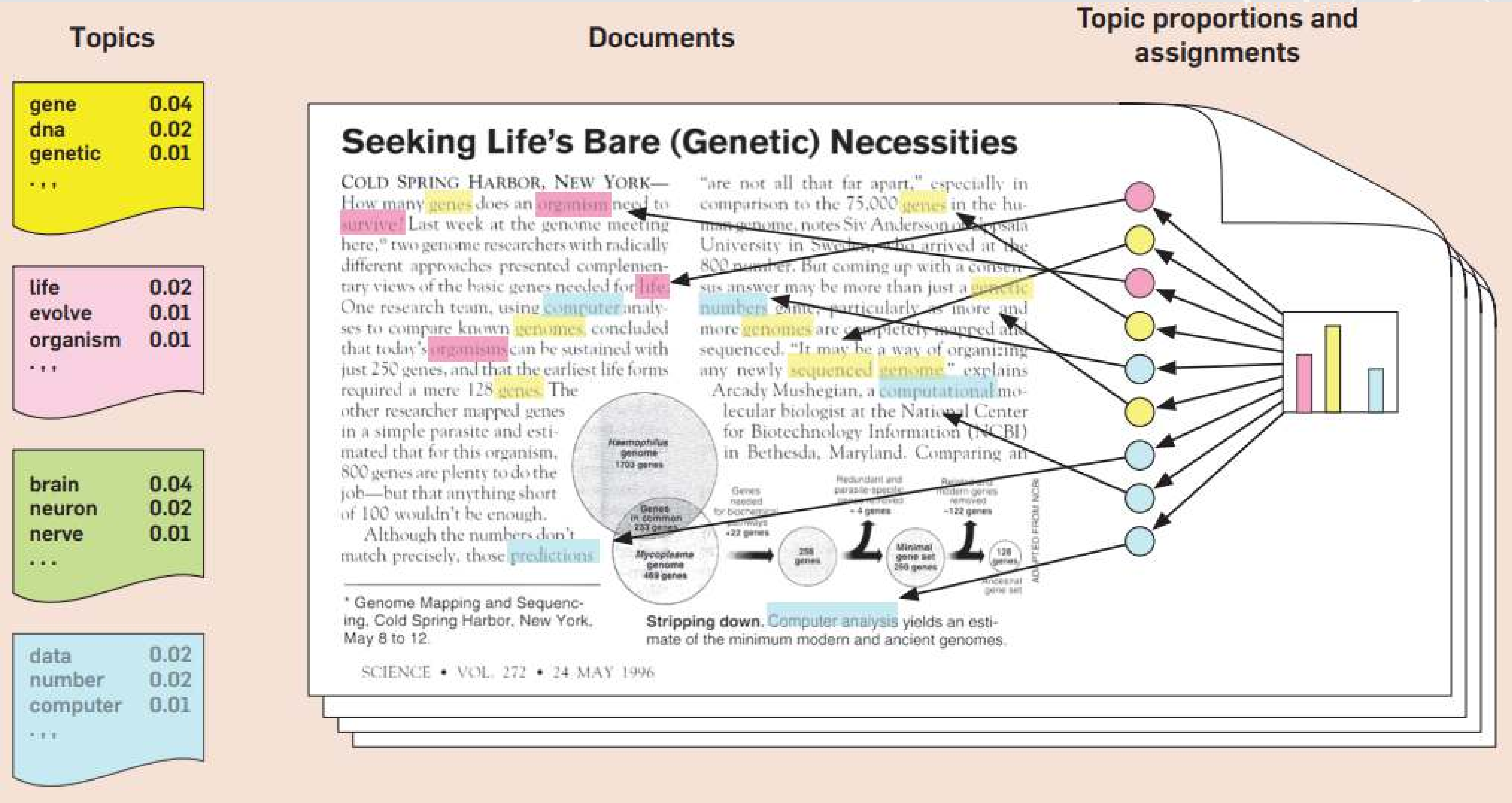
What is LDA?

- **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
     - An MCMC method

- It’s quite complicated in the background, 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

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,] 14590 14593 14598 14614 14625
## [2,]     1     1    38     3     1
```

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

```
## [1] "earlier"  "earliest" "earn"     "earthen"  "eas"
```

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 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: properti, oper, million, decemb, compani, interest, leas
##       FREX: ffo, efih, efh, tenant, hotel, casino, guc
##       Lift: aliansc, baluma, change-of-ownership, crj700s, directly-reimburs, escena, hhmk
##       Score: reit, hotel, game, ffo, tenant, casino, efih
## Topic 2 Top Words:
##       Highest Prob: compani, stock, share, common, financi, director, offic
##       FREX: prc, asher, shaanxi, wfoe, eit, hubei, yew
##       Lift: aagc, abramowitz, accello, akash, alix, alkam, almati
##       Score: prc, compani, penni, stock, share, rmb, director
## Topic 3 Top Words:
##       Highest Prob: product, develop, compani, clinic, market, includ, approv
##       FREX: dose, preclin, nda, vaccin, oncolog, anda, fdas
##       Lift: 1064nm, 12-001hr, 25-gaug, 2ml, 3shape, 503b, 600mg
##       Score: clinic, fda, preclin, dose, patent, nda, product
## Topic 4 Top Words:
##       Highest Prob: invest, fund, manag, market, asset, trade, interest
##       FREX: uscf, nfa, unl, uga, mlai, bno, dno
##       Lift: a-1t, aion, apx-endex, bessey, bolduc, broyhil, buran
##       Score: uscf, fhlbank, rmbs, uga, invest, mlai, ung
## Topic 5 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('Real Estate', 'Management', 'Product',
                                         'Investment', 'Services', 'Financing',
                                         'Service2', 'Insurance', 'Industrial',
                                         'Utility'))

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

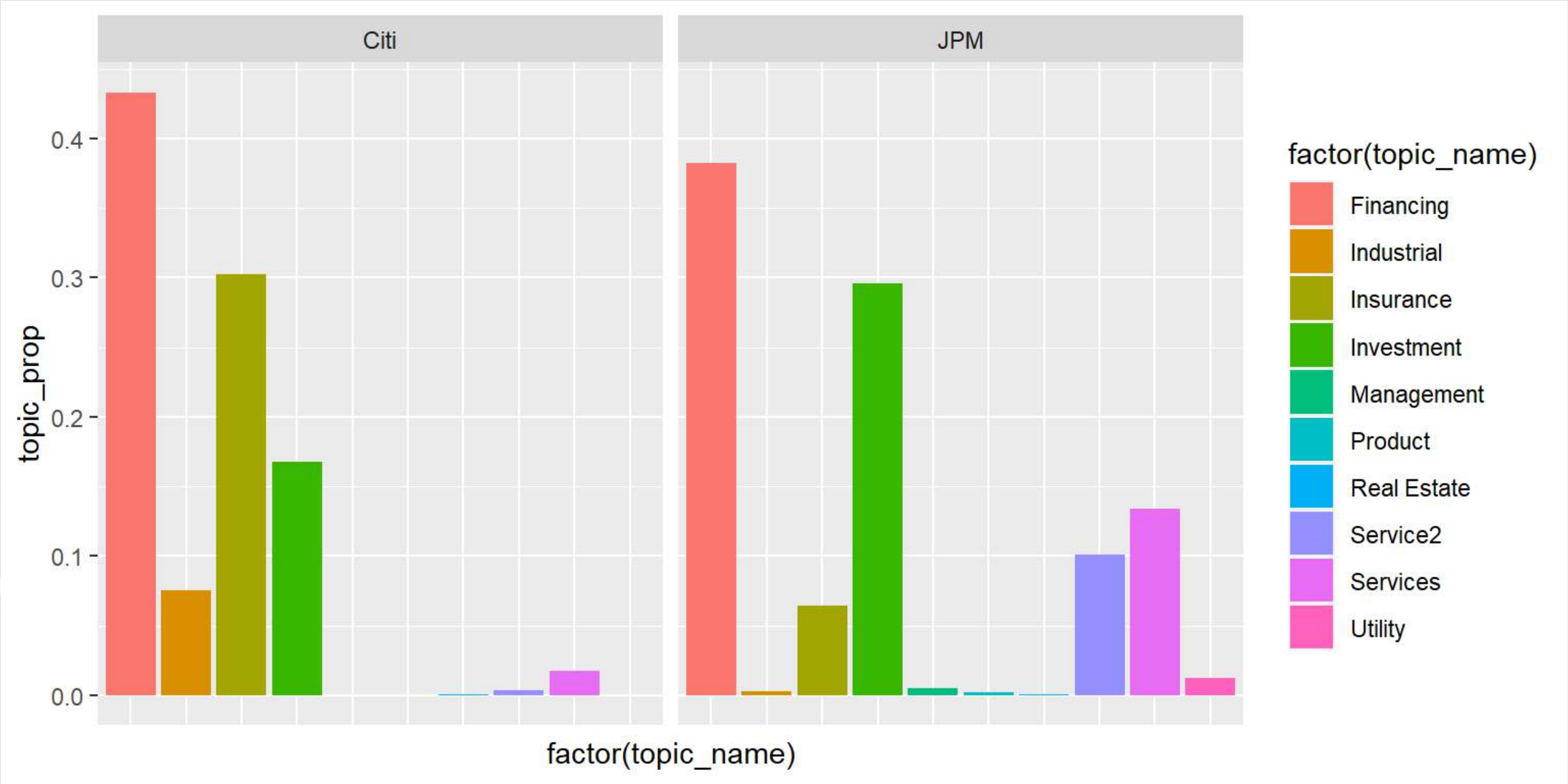
Topic content of the Citi 10-K

```r
doc_topics %>% filter(document=='0001104659-14-015152.txt')
```

```
## # A tibble: 10 x 6
##    document                 industry topic    weight topic_prop topic_name
##    <chr>                    <fct>    <dbl>     <dbl>      <dbl> <chr>
##  1 0001104659-14-015152.txt Finance      1 0.000316   0.000316  Real Estate
##  2 0001104659-14-015152.txt Finance      2 0.0000594  0.0000594 Management
##  3 0001104659-14-015152.txt Finance      3 0.0000153  0.0000153 Product
##  4 0001104659-14-015152.txt Finance      4 0.168      0.168     Investment
##  5 0001104659-14-015152.txt Finance      5 0.0172     0.0172    Services
##  6 0001104659-14-015152.txt Finance      6 0.433      0.433     Financing
##  7 0001104659-14-015152.txt Finance      7 0.00332    0.00332   Service2
##  8 0001104659-14-015152.txt Finance      8 0.303      0.303     Insurance
##  9 0001104659-14-015152.txt Finance      9 0.0755     0.0755    Industrial
## 10 0001104659-14-015152.txt Finance     10 0.0000558  0.0000558 Utility
```

Topic content of the Citi 10-K versus JPMorgan

```r
doc_topics %>%
  filter(document=='0001104659-14-015152.txt' |
           document=='0000019617-14-000289.txt') %>%
  mutate(Company=ifelse(document=='0001104659-14-015152.txt', 'Citi','JPM')) %>%
  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())
```

A nice visualization of our STM model

- Using LDAvis via `package:STM`'s `toLDAvis()` function
  - Need `LDAvis` and `servr` installed to run

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

Click to view

- Using `stmBrowser`'s `stmBrowser()` function
  - Install from github

```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

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 2014 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

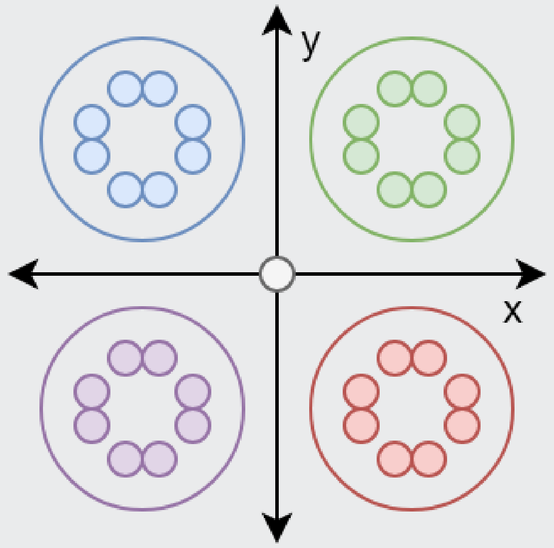
- While industry code is one classification of firms, it has a number of drawbacks:
  1. The classification system is old and perhaps misses new industries
  2. It relies on self-reporting
  3. Firms' classifications rarely change, even when firms themselves change

> 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`

```
library(tidyr)
wide_topics <- spread(doc_topics[,c(1,2,5,6)], topic_name, topic_prop)
mat <- wide_topics[,3:12]

mat[,1:6] %>% head() %>% html_df()
```

Financing	Industrial	Insurance	Investment	Management	Product
0.0105862	0.1578543	0.1088631	0.0004632	0.1161191	0.0002101
0.0467173	0.0059438	0.0235389	0.0005284	0.0801189	0.0001432
0.0069105	0.0351987	0.0003661	0.0201215	0.0023672	0.0000186
0.0870371	0.8271759	0.0003259	0.0003334	0.0206444	0.0000485
0.0036086	0.2680866	0.2677154	0.0008808	0.0026448	0.0000949
0.0000976	0.5299432	0.0001593	0.0007533	0.0009532	0.0000318

Calculating k-means

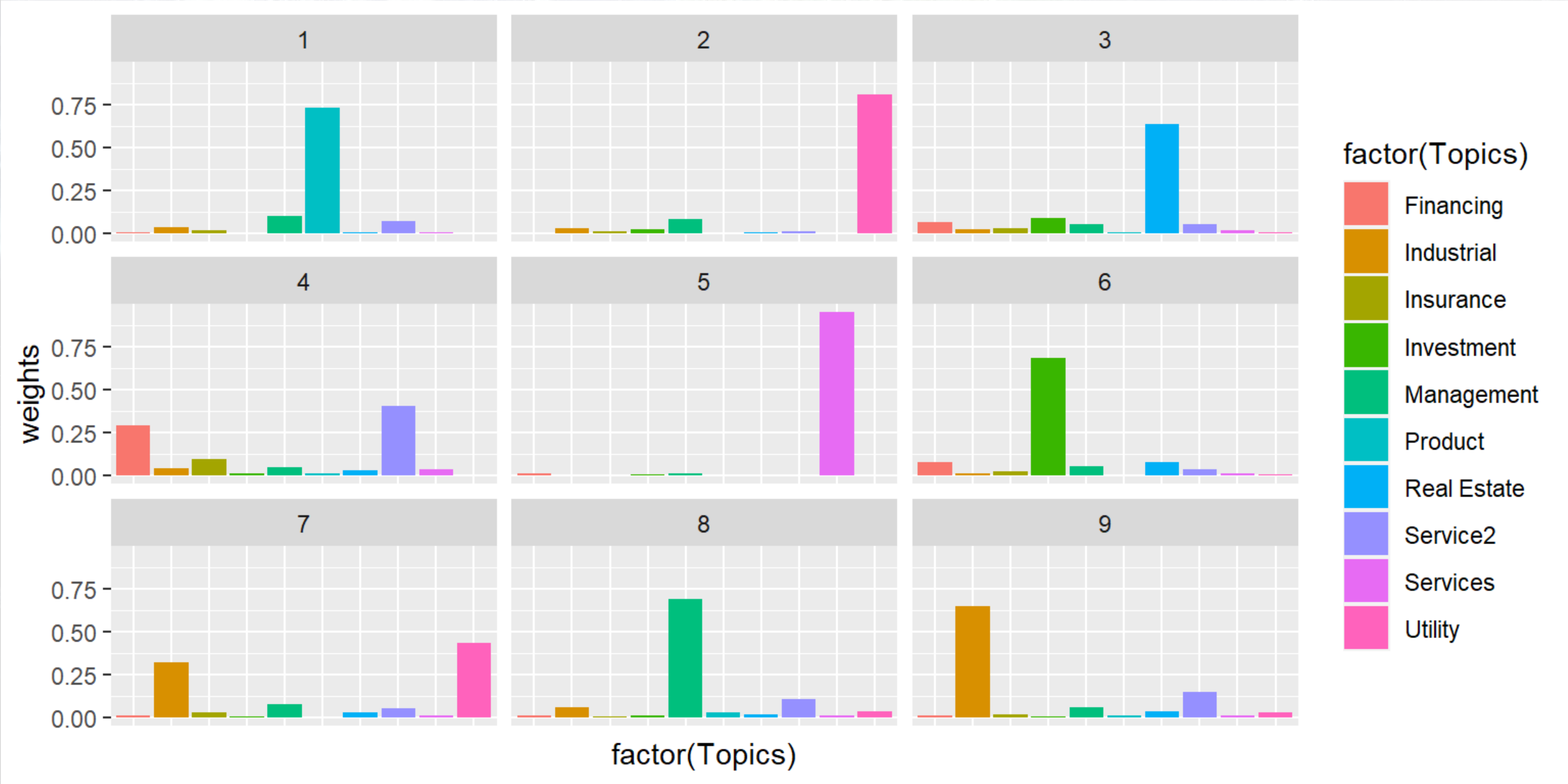
```r
set.seed(6845868)
clusters <- kmeans(mat, 9)
clusters$cluster %>% head()
```

```
## [1] 7 3 2 9 4 7
```

- The algorithm tells us group numbers for each observation
- The numbers themselves are arbitrary
  - The clustering (observations sharing a group number) is what matters

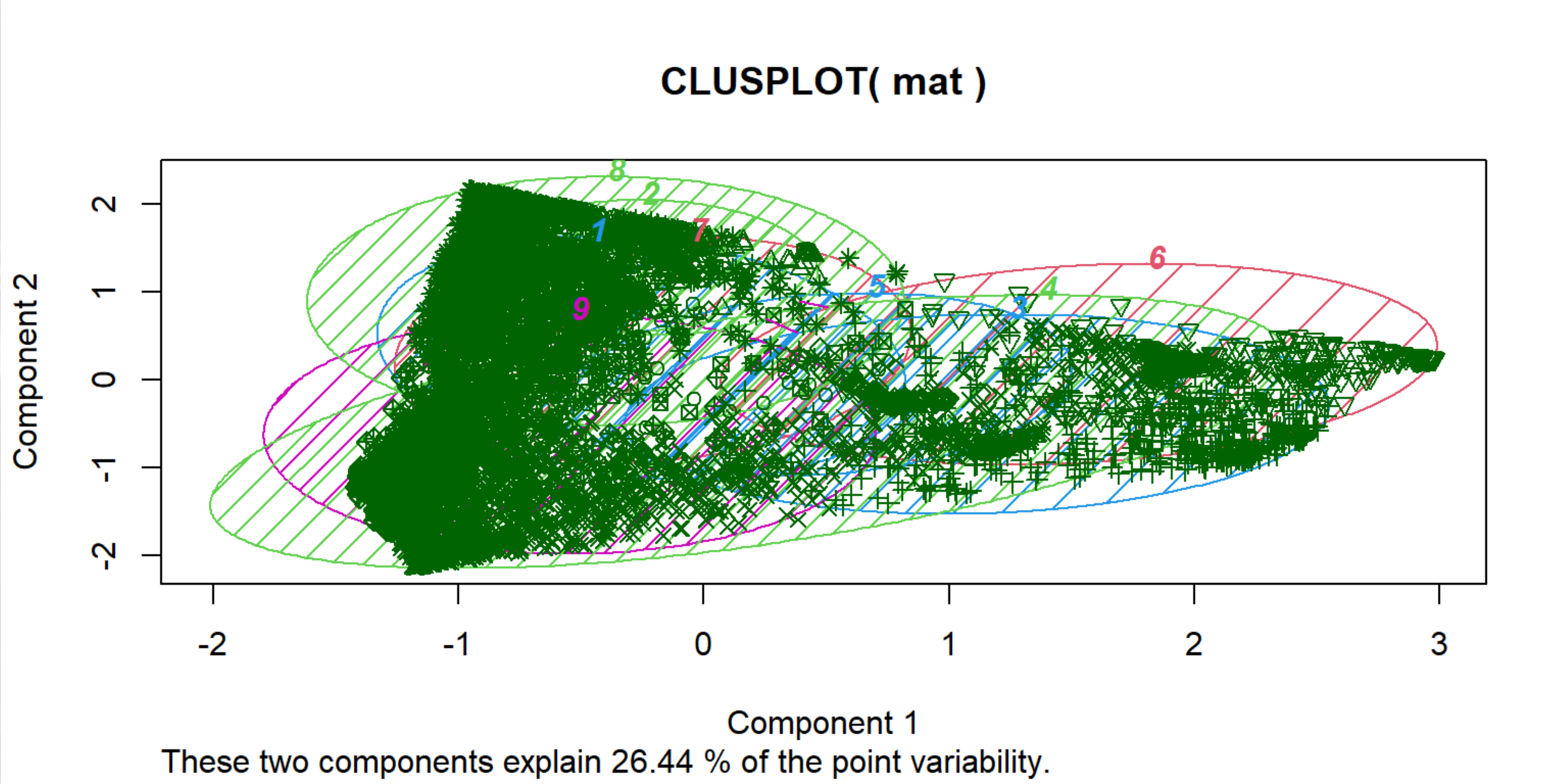
Visualizing the clusters

```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())
```



Visualizing k-means with PCA

```r
library(cluster) # Uses PCA (principle component analysis)
clusplot(mat, clusters$cluster, color=TRUE, shade=TRUE,
         labels=4)
```



Improving our visualization

- The PCA based map is really unreadable
  - This is usually the case, unless you have only a few dimensions to the data
- There is a relatively new method (2008), t-SNE, that is significantly better
  - **t**-distributed **S**tochastic **N**eighbor **E**mbedding
  - A machine learning algorithm designed to explain machine learning algorithms
    - It maintains neighbor relationships while reducing dimensions
  - It takes a much longer time to run than PCA, however
  - Implemented efficiently in R in the `Rtsne` package

Visualizing with t-SNE: Running t-SNE

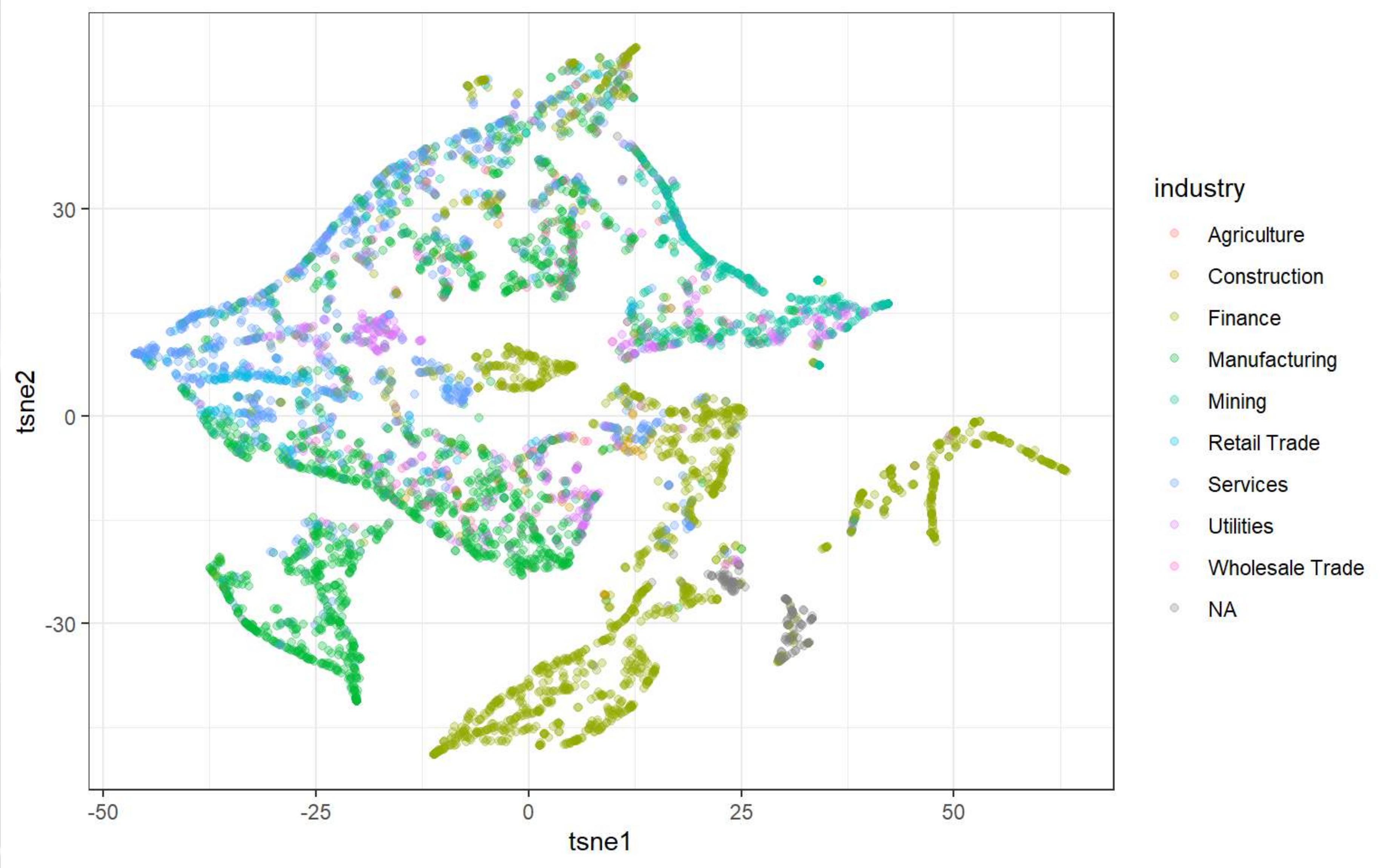
```r
library(Rtsne)
dups <- which(duplicated(mat))
wide_nodup <- wide_topics[-dups,]
wide_nodup$kmean <- clusters$cluster[-dups]

#This is slow (it's O(n log(n))).  Original model was O(n^2) though
tsne_data <- Rtsne(mat[-dups,])

wide_nodup <- wide_nodup %>%
  mutate(tsne1 = tsne_data$Y[, 1], tsne2 = tsne_data$Y[, 2])
```

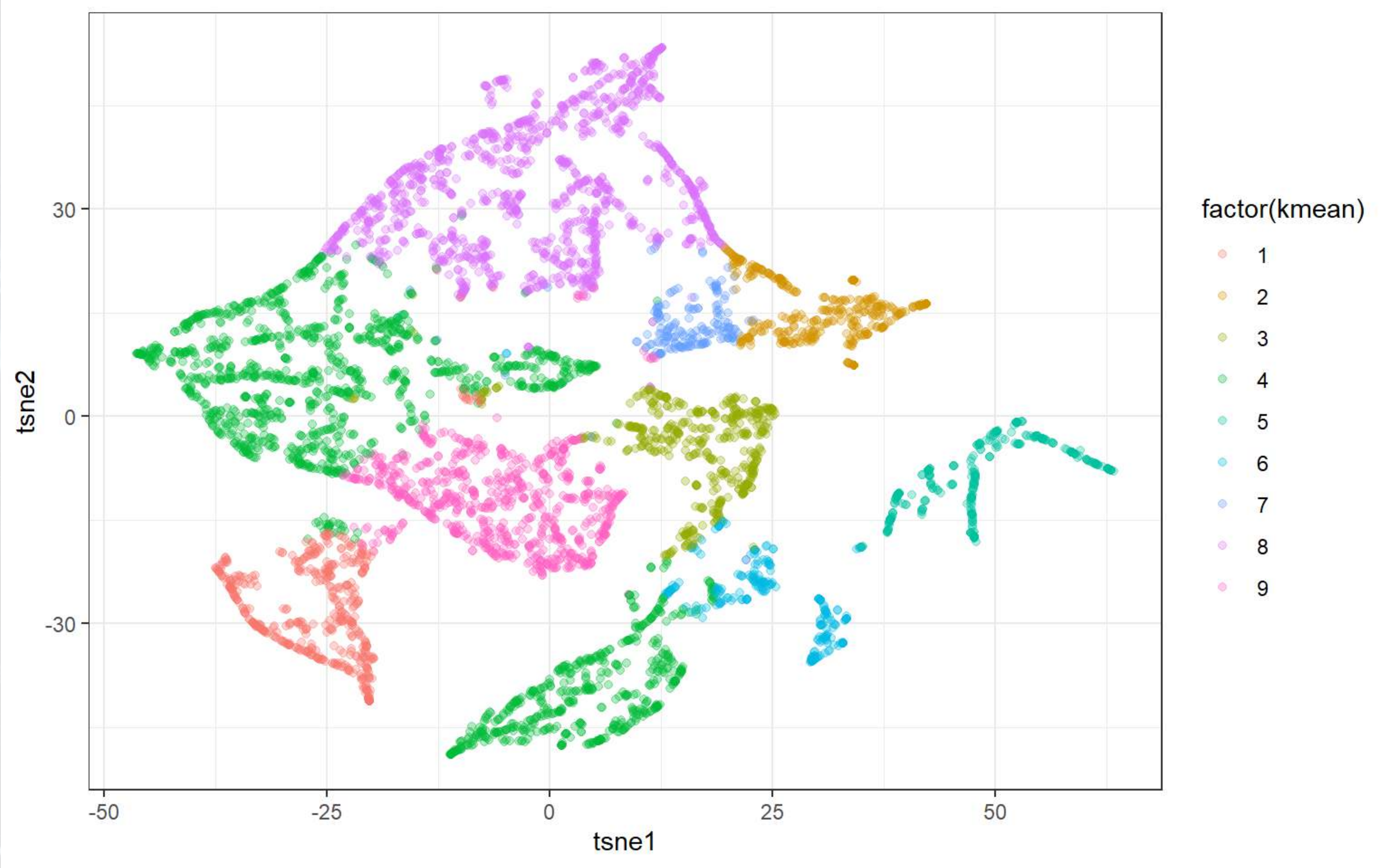
Visualizing with t-SNE: Industries

```r
ggplot(wide_nodup, aes(x = tsne1, y = tsne2, colour = industry)) +
    geom_point(alpha = 0.3) + theme_bw()
```



Visualizing with t-SNE: k-means

```r
ggplot(wide_nodup, aes(x = tsne1, y = tsne2, colour = factor(kmean))) +
    geom_point(alpha = 0.3) + theme_bw()
```

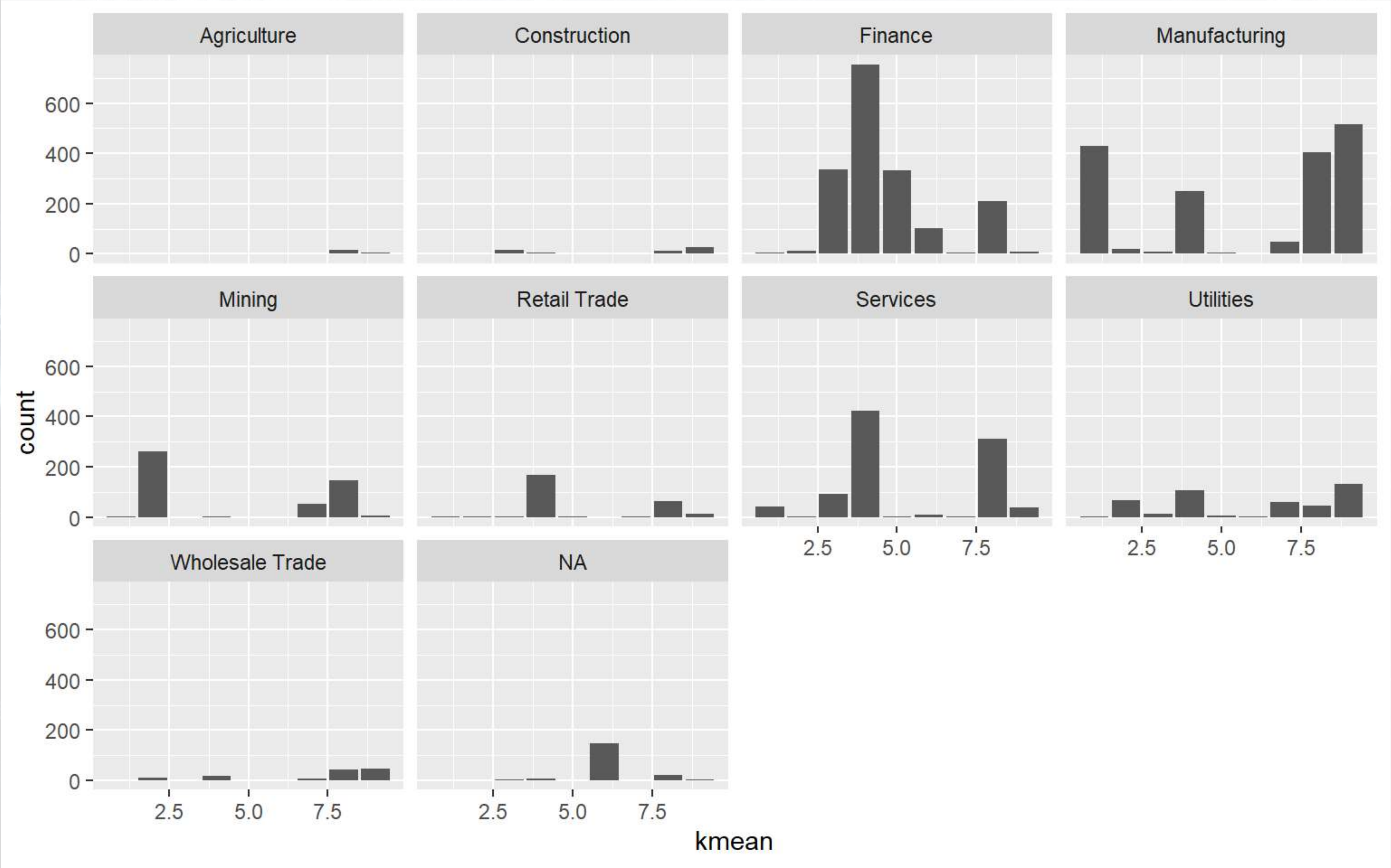


Why are these graphs different?

- 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

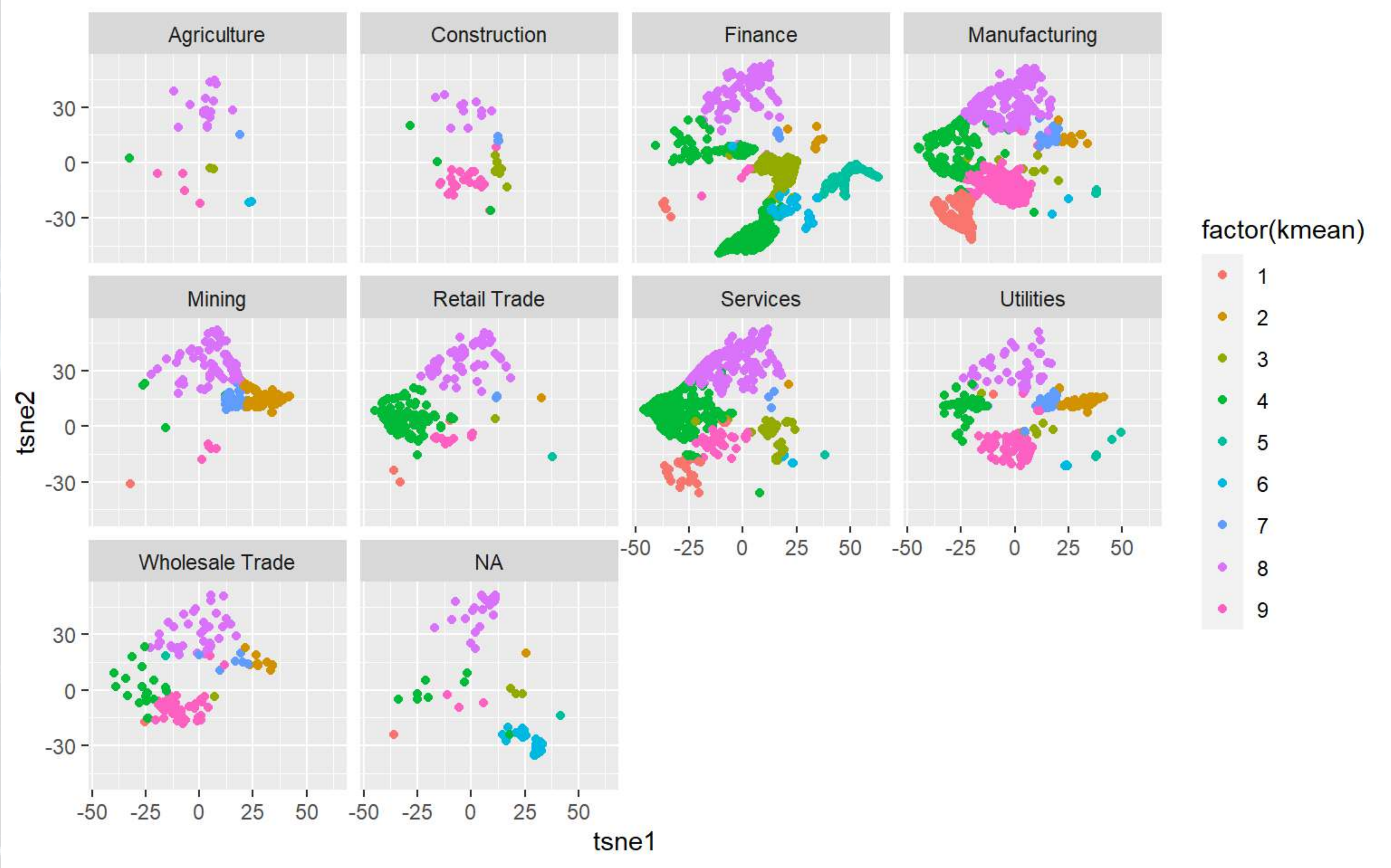
How related are clusters and industries?

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



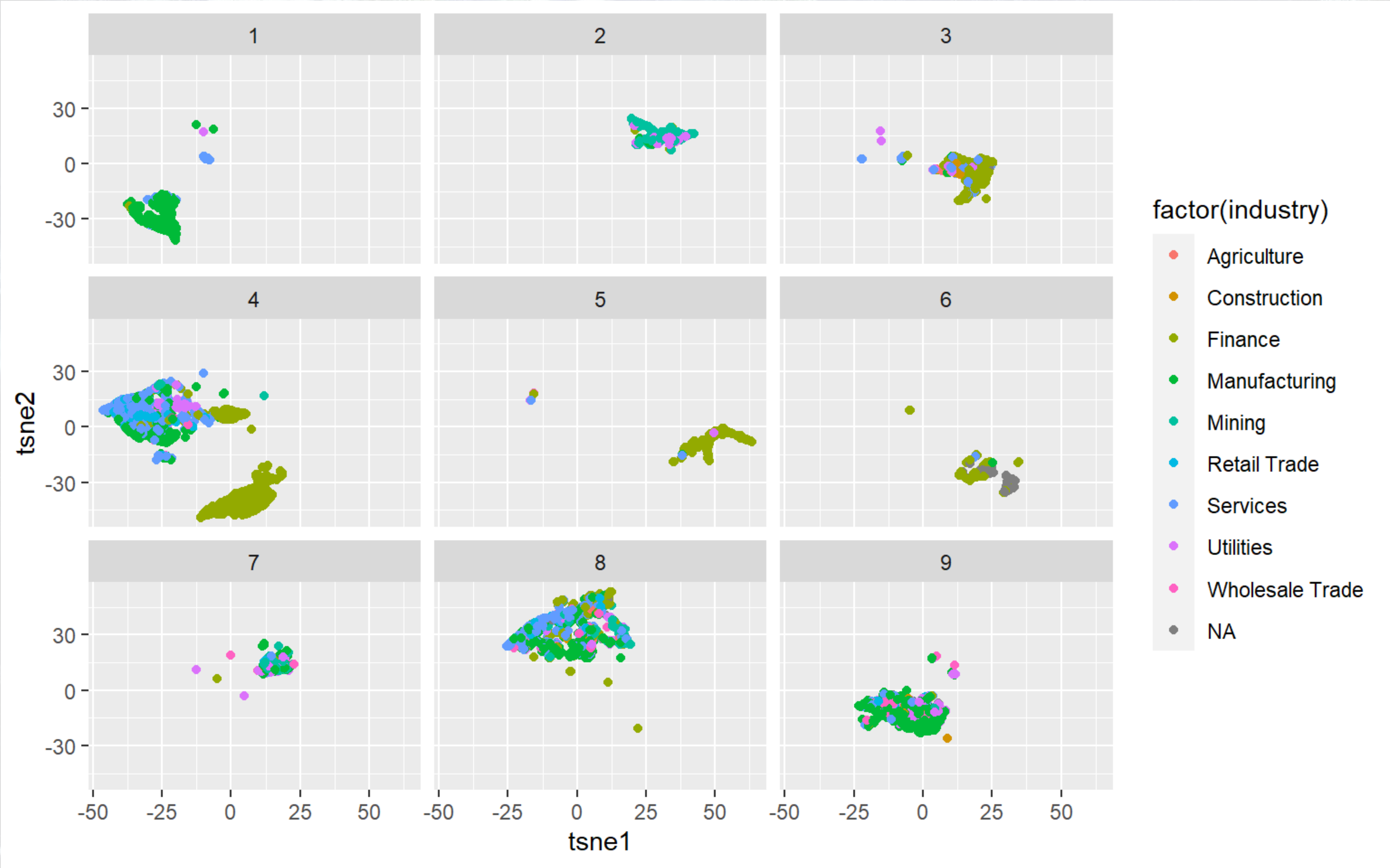
How related are clusters and industries?

```r
ggplot(wide_nodup, aes(x=tsne1, y=tsne2, color=factor(kmean))) + geom_point() +
  facet_wrap(~factor(industry))
```



How related are clusters and industries?

```r
ggplot(wide_nodup, aes(x=tsne1, y=tsne2, color=factor(industry))) + geom_point() +
  facet_wrap(~factor(kmean))
```



Great examples of t-SNE usage

- Visualizing handwritten numbers
- Visualizing Wikipedia articles
    - The full blog post, which is a great read about visualizing high-dimensional data

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$dist <- sqrt(rowSums(mat - fitted(clusters))^2)
wide_topics$dist <- sqrt(rowSums(abs(mat - fitted(clusters))))
wide_topics[,c(1,2,3,5,13)] %>% arrange(desc(dist)) %>% slice(1:5) %>% html_df()
```

document	industry	Financing	Insurance	dist
0001171500-14-000007.txt	Finance	0.0003177	0.9972499	1.341244
0001193125-14-077320.txt	Finance	0.0001725	0.9794704	1.337283
0001095073-14-000008.txt	Finance	0.0002339	0.9916079	1.337031
0000356130-14-000052.txt	Finance	0.0002991	0.9845263	1.334780
0000021175-14-000021.txt	Finance	0.0036298	0.9875105	1.333963

- 5 standard insurance companies
  - SIC codes lump banks and insurance together, but they are actually very different industries
  - E.g.: Platinum Underwriters Holdings

Looking for anomalies

id	industry	Insurance	Product	Real Estate	Service2	Services	dist
1	Services	0.5161854	0.2641739	0.1112599	0.0136804	0.0764400	1.252719
2	Services	0.4154754	0.2778976	0.1109746	0.1116213	0.0814478	1.185233
7	Services	0.5878499	0.1535348	0.0006138	0.1822722	0.0231219	1.123097
6	Services	0.3184271	0.2718329	0.2489164	0.0520256	0.1037725	1.128411
8	Retail Trade	0.3603968	0.1876330	0.0854220	0.0934442	0.0894848	1.119306

- 1-4, 9-10: Healthcare services + real estate (1: HCS Holdings)
- 7: Healthcare and insurance management (Magellan Health Services)
- 6 & 8: Healthcare management (Select Medical Holdings & Omnicare)

id	industry	Investment	Real Estate	Service2	Services	Utility	dist
5	Utilities	0.1768971	0.1143861	0.2481198	0.4017117	0.053171	1.144542

- 5: Partnership for TV/internet/telco in 2 Sourthern US rural areas
  - Northland Cable Properties Eight Ltd. Ptr.

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[!is.na(wide_topics$industry),]
label <- wide_topics[is.na(wide_topics$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[,-1])
saveRDS(tout, '../../Data/corp_knn.rds')
```

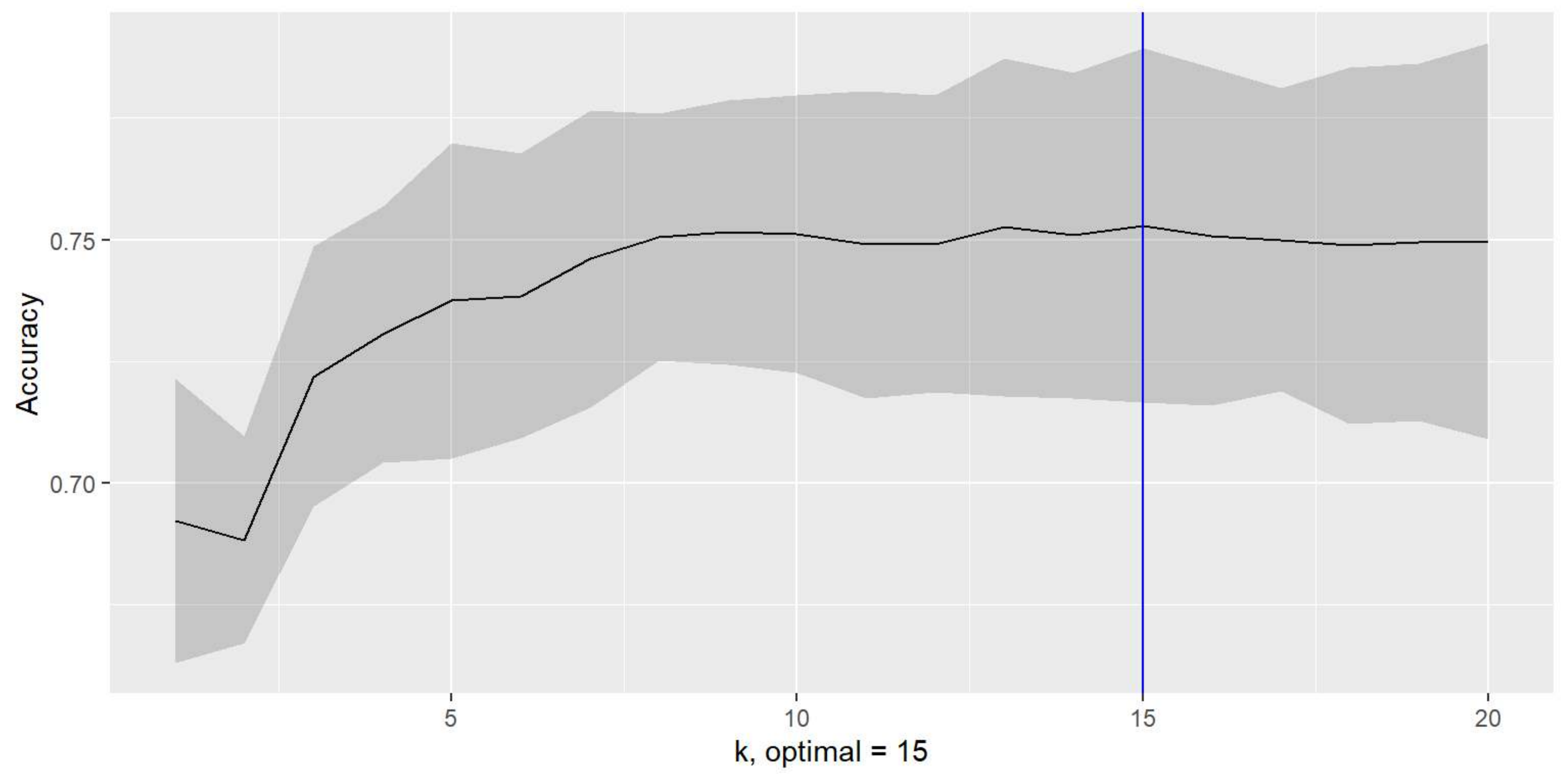
Implementing KNN in R

```r
tout
```

```
## k-Nearest Neighbors
##
## 5804 samples
##   10 predictor
##    9 classes: 'Agriculture', 'Construction', 'Finance', 'Manufacturing', 'Mining', 'Retail Trade', 'Services', 'Utilities',
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 5226, 5222, 5223, 5224, 5223, 5226, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.6922669  0.6037548
##    2  0.6883222  0.5984635
##    3  0.7219205  0.6397779
##    4  0.7305403  0.6495724
##    5  0.7374387  0.6581581
##    6  0.7384702  0.6592123
##    7  0.7460449  0.6686815
##    8  0.7505306  0.6741651
##    9  0.7515604  0.6753179
```

KNN performance

```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")
```



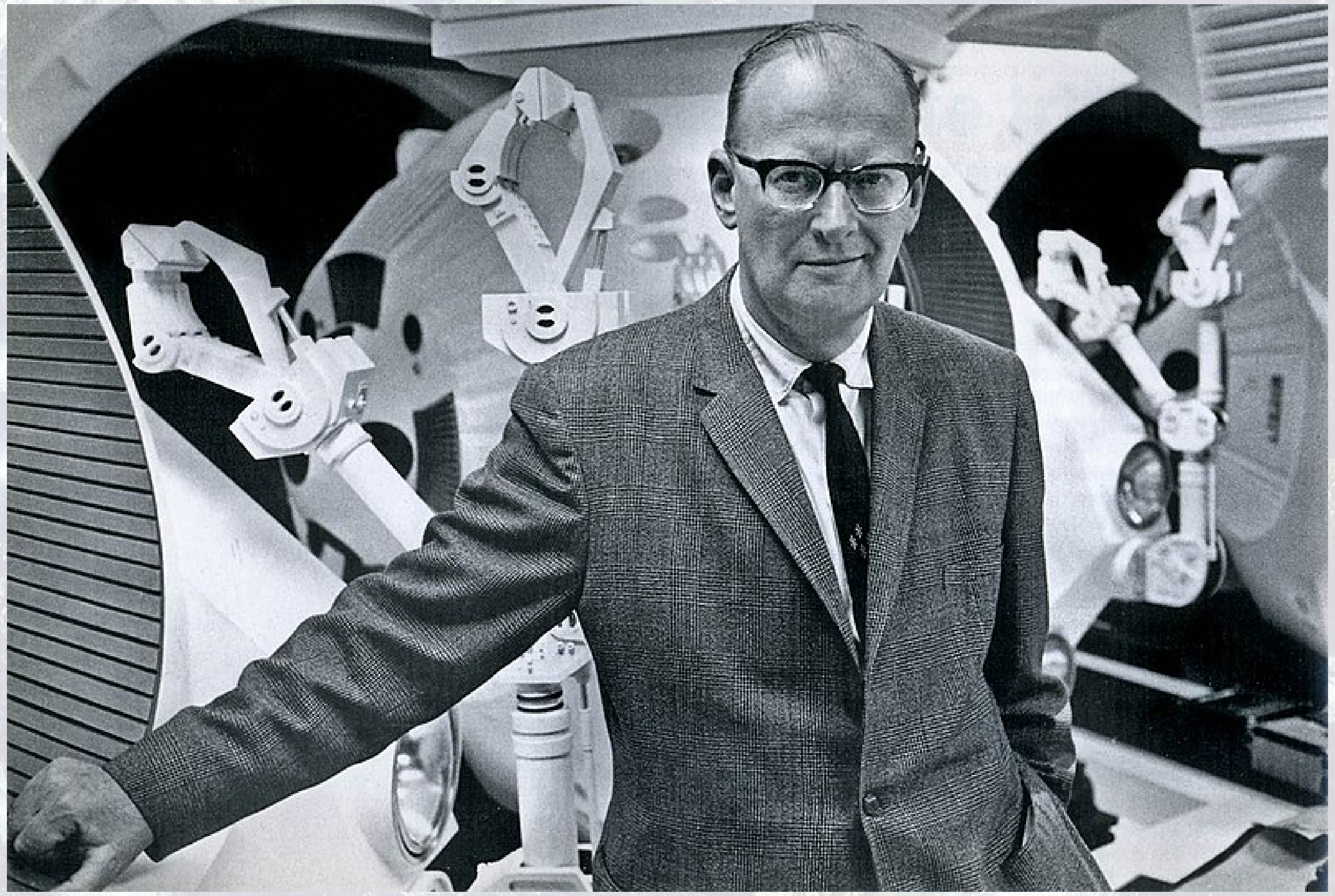
Using KNN to fill in industry

1. American Capital: Asset manager and private equity
   - SIC missing, but clearly finance ✓
2. Ameriprise Certificate Co: Investment company
   - SIC missing, but clearly finance ✓
3. Callaway Golf: Golf equipment
   - SIC 3949 ✓
4. Everest Fund L P: Speculative trading of commodity futures
   - SIC 6221 ✓
5. Bank of Nova Scotia: Joint with Scotiabank Covered Bond Guarantor Limited Partnership
   - SIC 6022 ✓
6. Teucrium Commodity Trust: Commodity funds
   - SIC 6221 ✓

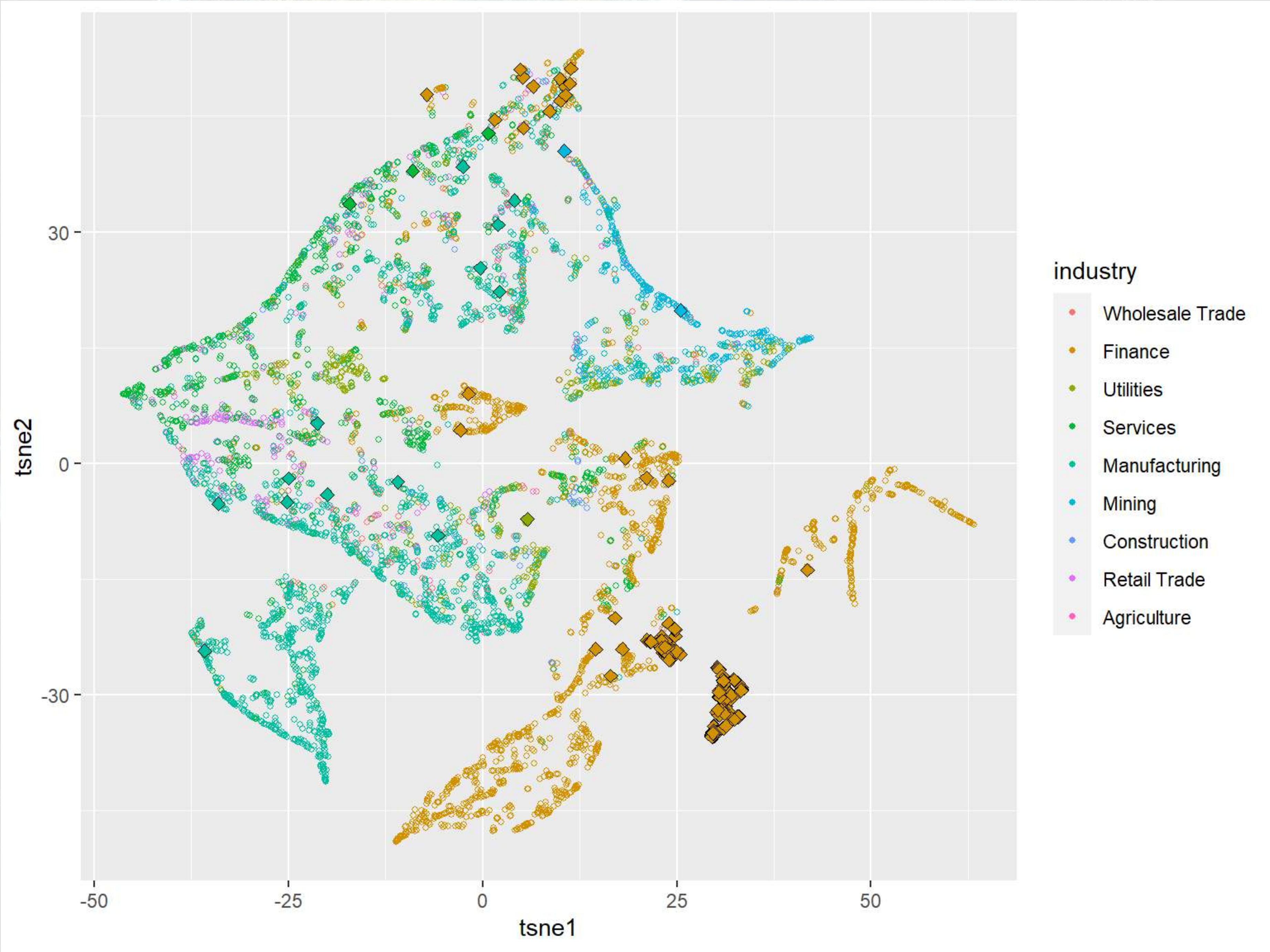
```r
label$industry_pred <- predict(tout,
                               label)
label[,c("document",
         "industry_pred")] %>%
  head %>% html_df
```

document	industry_pred
0000817473-14-000010.txt	Finance
0000820027-14-000025.txt	Finance
0000837465-14-000002.txt	Manufacturing
0000837919-14-000002.txt	Finance
0000891092-14-000570.txt	Finance
0000891092-14-002078.txt	Finance

> “Any sufficiently advanced technology is indistinguishable from magic.” – Sir Arthur Charles Clarke



Bonus: t-SNE on KNN



Bonus: t-SNE on KNN (code)

```r
ts_wt <- wide_nodup %>% left_join(label[,c("document","industry_pred")])

ts_wt <- ts_wt %>%
  mutate(tsne1 = tsne_data$Y[, 1], tsne2 = tsne_data$Y[, 2])

# Force consistent factor values
inds <- unique(ts_wt$industry)
ts_wt$industry <- factor(ts_wt$industry, inds)
ts_wt$industry_pred <- factor(ts_wt$industry_pred, inds)

# Replicate default ggplot colors
ggplotColours <- function(n = 6, h = c(0, 360) + 15){
  if ((diff(h) %% 360) < 1) h[2] <- h[2] - 360/n
  hcl(h = (seq(h[1], h[2], length = n)), c = 100, l = 65)
}
```

```r
ggplot() +
  scale_shape_identity() + # Allow for more plot point options
  geom_point(data=ts_wt[!is.na(ts_wt$industry),],
             aes(x=tsne1, y=tsne2, color=industry, shape=1), size=1) +
  geom_point(data=ts_wt[!is.na(ts_wt$industry_pred),], aes(x=tsne1, y=tsne2,
             fill=industry_pred, shape=23, stroke=0.5), size=2) +
  guides(fill = "none") +
  scale_color_manual(values=ggplotColours(n = 9), labels=inds, drop=FALSE) +
  scale_fill_manual(values=ggplotColours(n = 9), labels=inds, drop=FALSE)
```

Recap

Today, we:

1. Processed a set of 6,000 annual reports from 2014 to examine their readability
2. Examined the content discussed in annual reports in 2014
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 all 6 checked entries ✓

End matter

For next week

- For next week:
  - Datacamp
    - Do the assigned chapter on machine learning
  - Keep working on the group project

Packages used for these slides

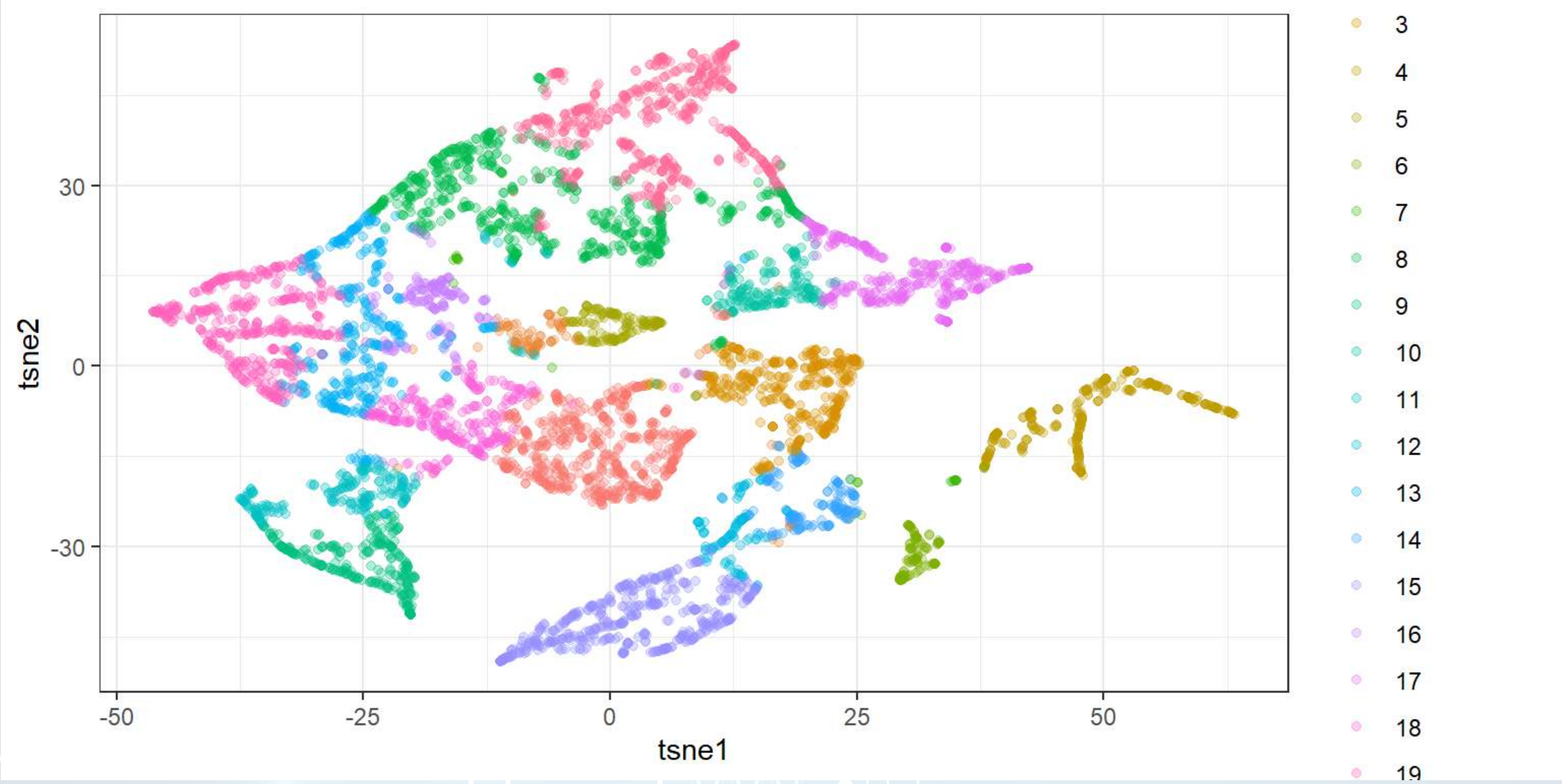
- `caret`
- `cluster`
- `DT`
- `kableExtra`
- `knitr`
- `lattice`
- `quanteda` and `stopwords`
- `readtext`
- `revealjs`
- `Rtsne`
- `stm` and `stmBrowser`
- `tidyr`
- `tidyverse`
  - `dplyr`, `magrittr`, `readr`

Custom code

```r
library(knitr)
library(kableExtra)
html_df <- function(text, cols=NULL, col1=FALSE, full=F) {
  if(!length(cols)) {
    cols=colnames(text)
  }
  if(!col1) {
    kable(text,"html", col.names = cols, align = c("l",rep('c',length(cols)-1))) %>%
      kable_styling(bootstrap_options = c("striped","hover"), full_width=full)
  } else {
    kable(text,"html", col.names = cols, align = c("l",rep('c',length(cols)-1))) %>%
      kable_styling(bootstrap_options = c("striped","hover"), full_width=full) %>%
      column_spec(1,bold=T)
  }
}
```

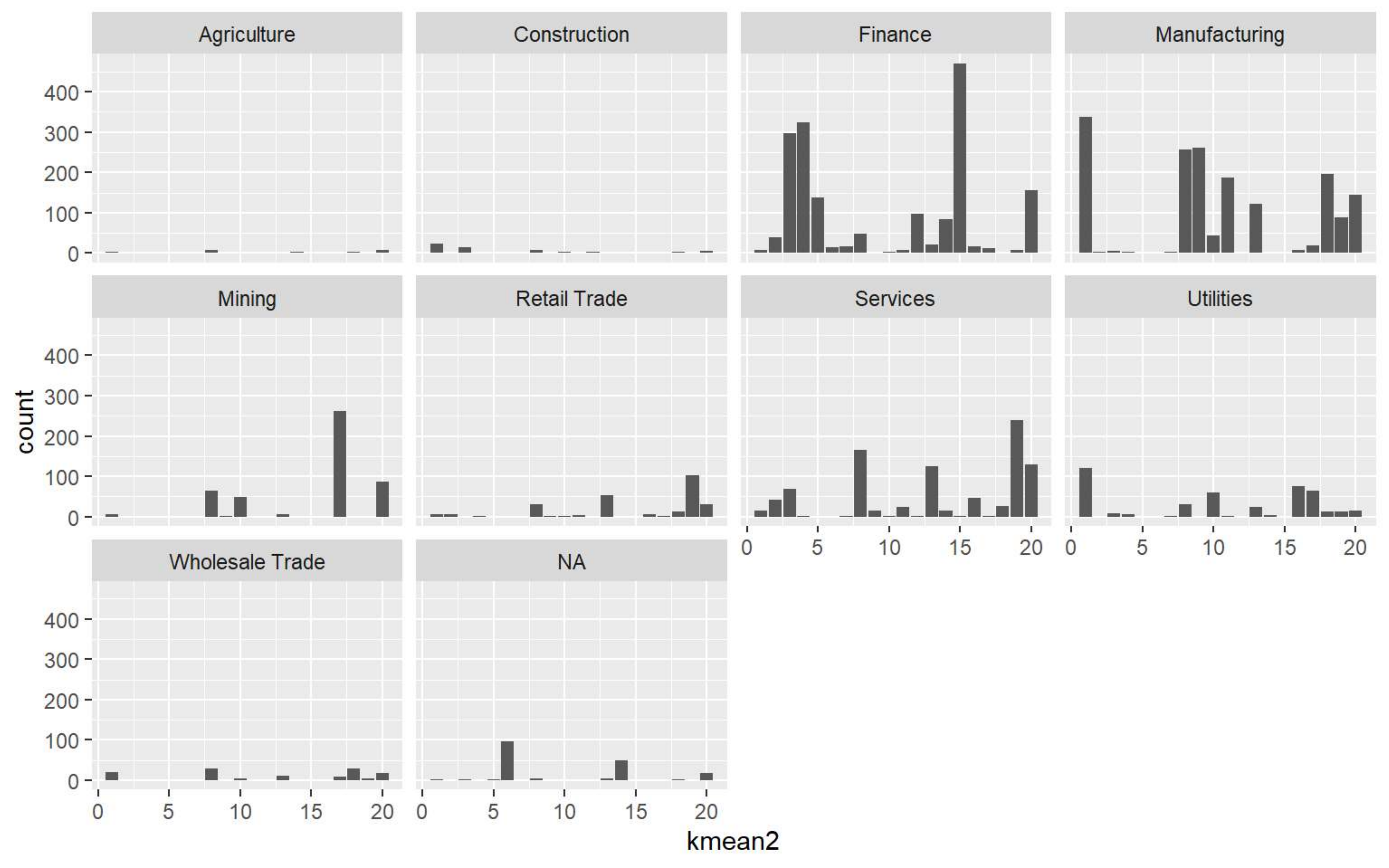
Using more clusters

```r
wide_nodup$kmean2 <- clusters$cluster[-dups]
ggplot(wide_nodup, aes(x = tsne1, y = tsne2, colour = factor(kmean2))) +
    geom_point(alpha = 0.3) + theme_bw()
```



Using more clusters

```r
ggplot(wide_nodup, aes(x=kmean2)) + geom_bar() + facet_wrap(~factor(industry))
```



Using more clusters

```r
ggplot(wide_nodup, aes(x=tsne1, y=tsne2, color=factor(kmean2))) + geom_point() +
  facet_wrap(~factor(industry))
```