Quanteda

SESSION 5

Next steps

Formulate hypotheses

- What would I like to show

Simplify hypotheses

- CIA documents
  - “CIA has 7 ways to obfuscate information and they can co-occur” <- NO
  - “CIA has 3 main ways to obfuscate information. There are an additional 6 ways that could occur within those.”

Operationalization

Simplify more

- Hypothesis 1: The CIA has 3 main methods of obfuscation.
- Operationalization: To test this, I manually code 10% of my documents to be one of the three categories. I train a Naïve Bayesian Classifier to categorize the documents into those three types. I then apply the trained model on a separate set of XX documents that were not used in training. The accuracy of the model is XX%.
- Hypothesis 2: There are 6 subclasses of tools the CIA could use that could co-occur with one another.
- Operationalization: I run a topic model with XX topics to determine each topic's proportion of a document. Based on the tokens for each topic, I identify the one most closely related to my subclasses and find that subclasses 2 and 4 tend to co-occur while subclass 5 mainly appears by itself.

When it doesn’t work

- Topic models take all tokens so maybe will not exact topics desired
- Also, could get topics for broader class
  - Options: Python – seeded topic models
  - Different approach?
- Dictionaries?
  - User-defined list of words for each subclass
  - Frequency of words in subclass as indicator of subclass
- Hierarchical classifications?

Assumption 3: Why frequencies?

Zipf's law

- Frequency of word inversely proportional to rank in a frequency table

Words and phrases with highest frequency reflect important concerns in every communication

- If document concerns "taxation" it should mention "taxation"* more frequently

* Or a synonym of taxation, or part of the word taxation**

** such as tax, taxing, taxes, taxable

Meta concerns

Data structure

- Shape of dataset
  - How to identify unique observations
- Size of dataset
  - Strings compared to numbers
  - Repeated column observations

Metadata

Data about data

Higher level data in a sense

- Might include document number, author, data source, date

With text data includes all document related information

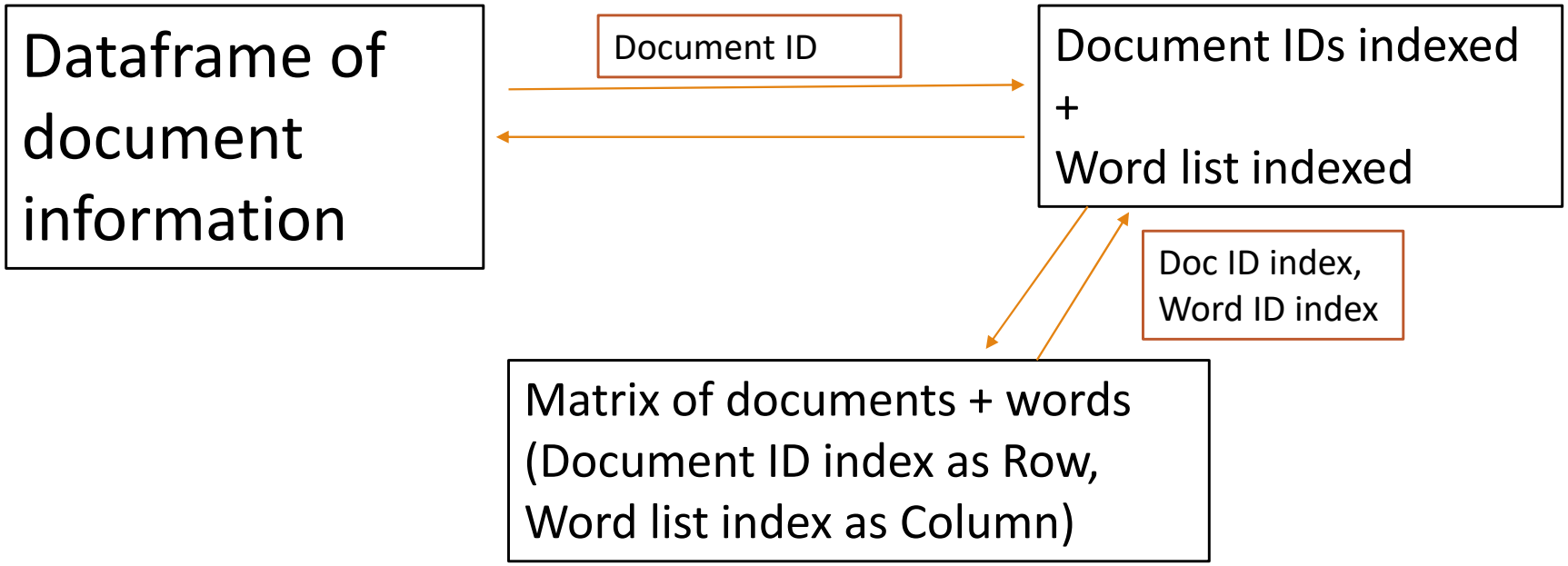
Where is metadata stored with document frequencies?

Text data

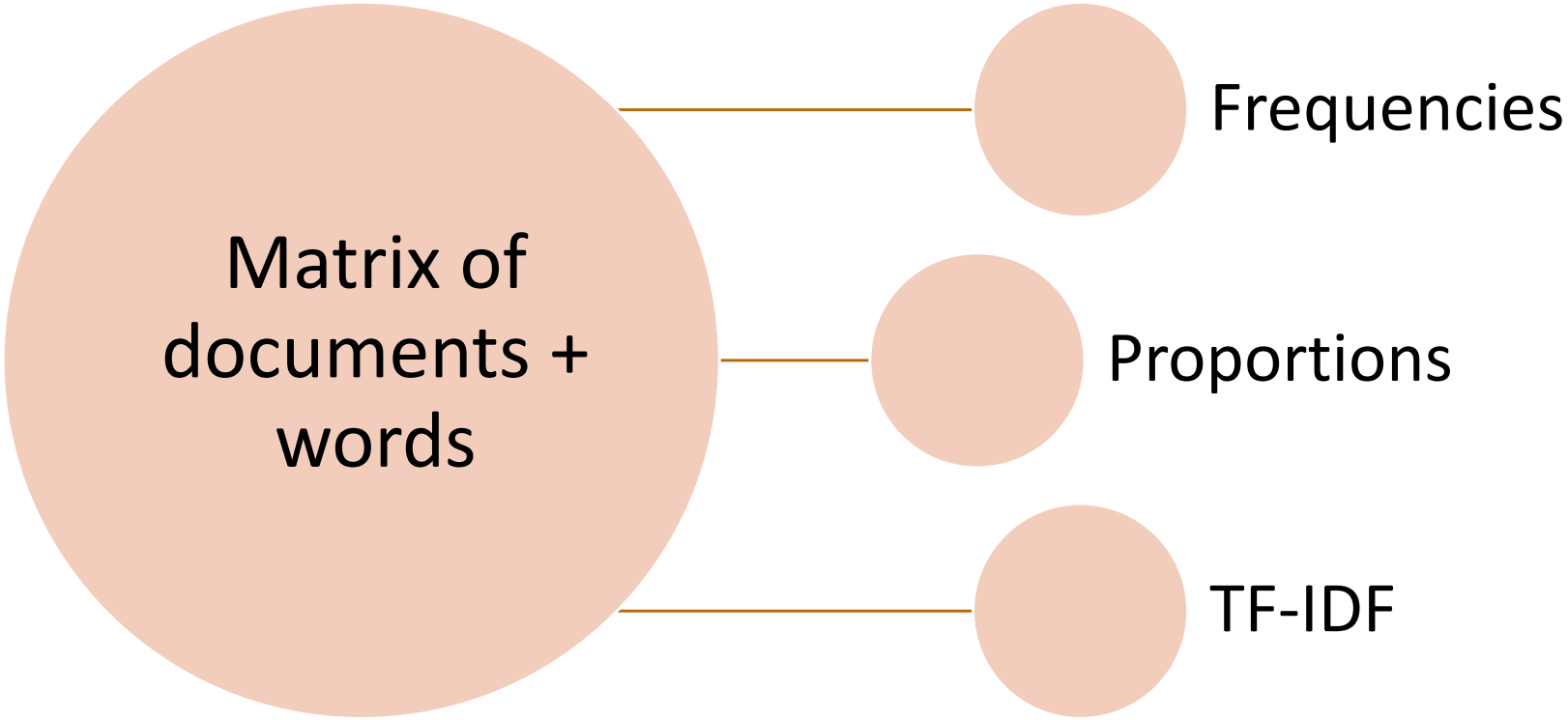
Different objects for different types of data

- First object = dataframe
  - Single observation per document
  - Metadata or information specific to document
- Second objects = word list and document list
  - All words in all documents (corpus) and list of documents
- Why? Third object = matrix of # times word appears in document
  - Number of times word occurs in each document
  - Sparse matrix

Storing text data



Easy transformations



Concepts

Corpus
- Collection of textual documents for analysis

Document
- Individual text

Tokens
- Any word
- Token count = total words

Key words
- Words with special attributes, meanings, or frequencies

Documents

Many possibilities

- Sentences
- Paragraphs/Stanzas
- Pages
- Document (speech, report, manifesto, poem)
- All documents by group

Tokenize

Process of taking elements of text and splitting them based on some character

- Spaces for words
- Punctuation marks for sentences

More concepts

Stemming

- Removing suffixes off words
  - Run: run, running, runs, runner
  - Hope that it's correct most of the time

Lemmas

- Grouping together inflected forms of a word
  - Run: run, running, runs, ran

Stop words

- Natural language words with little meaning

Which words are important

How do we decide which words to use?

Depends on context and research question

Many different possibilities

Cover a few basics to reduce number of features

Comparison with numeric data

Trained to focus on variables that are important in numeric data

Universe of cross-national data = universe of corpus words

- Political variables: # legislators in legislative branch, PR v. presidential system
- Economic variables: GDP, Unemployment, trade, etc.
- Social variables: literacy, mortality, ethnicity, etc
- Idiosyncratic variables: # pets, # restaurants

Not each type of variable helpful

First decisions

Remove punctuation?

Remove numbers?

Convert all to lower case?

- “The” and “the” or “We” and “we” the same
- What about “Rose” and “rose”? Or “Bill” and “bill”?

Stemming

Removing suffixes of words to get at root

- Example: “meet” for “meet”, “meeting”, “meetings”
- “propos” for “proposal”, “propose”, “proposed”

Reduce number of features

- Meet, meeting and meetings all have similar meanings

Lemmas

Dictionary form of word

More computationally intensive

Attempts to figure out context

Quanteda does not do

- Need own dictionary

Stop words

Words with very little meaning

Most common words in language:

- “a”, “the”, “and”, “or”, “could”
- Offer little insight into text
- BUT Federalist papers attribution

Also, words that do not vary much across documents

- E.g., State Department reports: embassy, ambassador, state

Dictionaries

List of similar words that constitute a broader topic

Pre-defined lexicon

Example: Sentiment analysis

- Define a list of features that convey positive sentiment
- Define list of features that convey negative sentiment

Document feature matrix

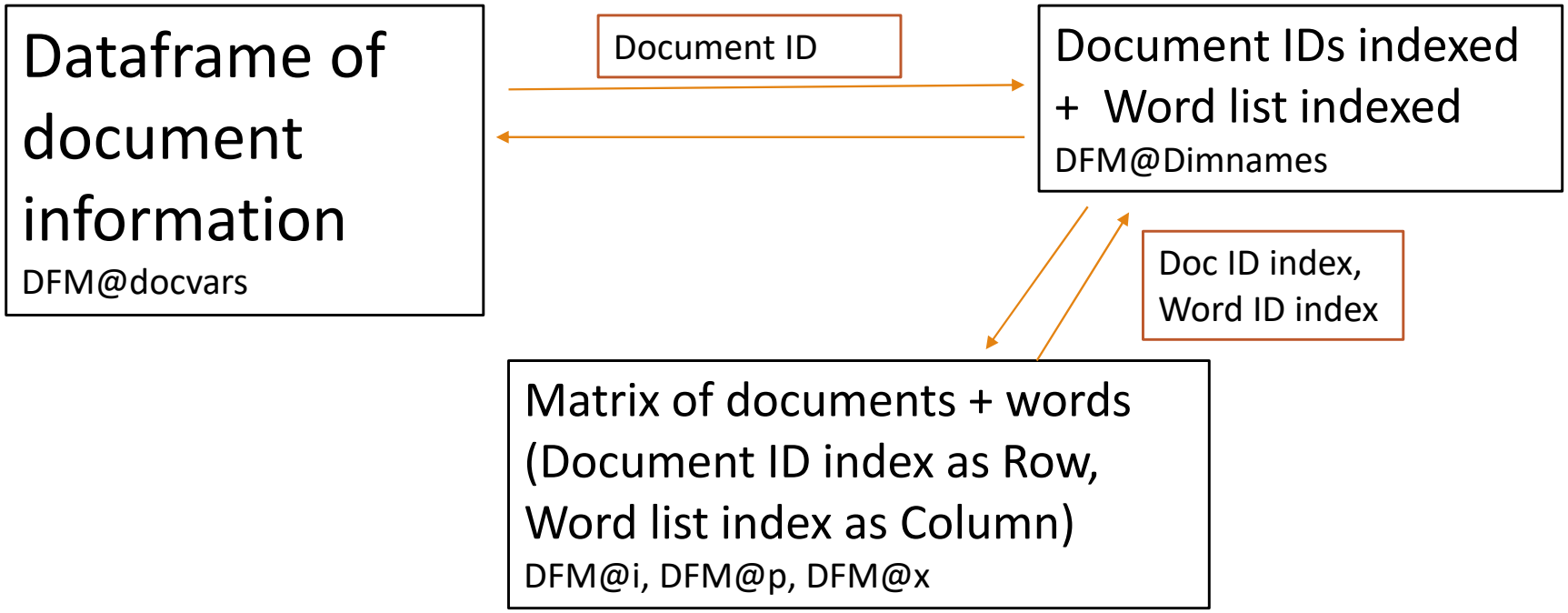
Quanteda converts corpus into document feature matrix (DFM) object

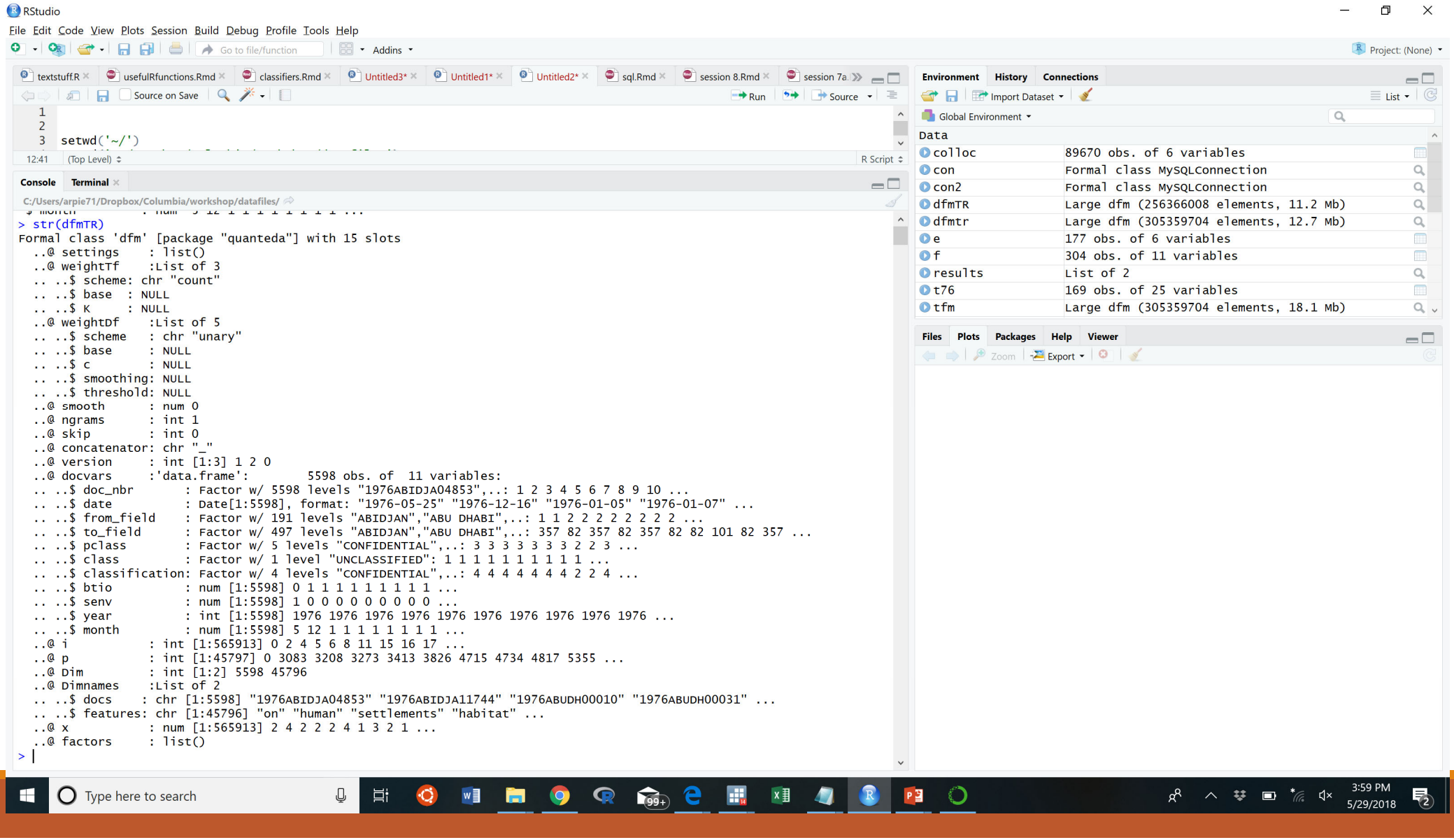
Dataframe -> information about the documents

Word lists+document lists -> link dataframe to bag of words

Bag of words or document feature matrix -> feature's frequency in each document

Document feature matrix




RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
textstuff.R
usefulRfunctions.Rmd
classifiers.Rmd
Untitled3*
Untitled1*
Untitled2*
sql.Rmd
session 8.Rmd
session 7a
Source on Save
Run
Source
1
2
3
setwd('~/')
12:41
(Top Level)
R Script
Console
Terminal
C:/Users/arpie71/Dropbox/Columbia/workshop/datafiles/
> str(dfmTR)
Formal class 'dfm' [package "quanteda"] with 15 slots
..@ settings : list()
..@ weightTf :List of 3
.. ..$ scheme: chr "count"
.. ..$ base : NULL
.. ..$ K : NULL
..@ weightDf :List of 5
.. ..$ scheme : chr "unary"
.. ..$ base : NULL
.. ..$ c : NULL
.. ..$ smoothing: NULL
.. ..$ threshold: NULL
..@ smooth : num 0
..@ ngrams : int 1
..@ skip : int 0
..@ concatenator: chr "_"
..@ version : int [1:3] 1 2 0
..@ docvars :'data.frame': 5598 obs. of 11 variables:
.. ..$ doc_nbr : Factor w/ 5598 levels "1976ABIDJA04853",..: 1 2 3 4 5 6 7 8 9 10 ...
.. ..$ date : Date[1:5598], format: "1976-05-25" "1976-12-16" "1976-01-05" "1976-01-07" ...
.. ..$ from_field : Factor w/ 191 levels "ABIDJAN","ABU DHABI",..: 1 1 2 2 2 2 2 2 2 2 ...
.. ..$ to_field : Factor w/ 497 levels "ABIDJAN","ABU DHABI",..: 357 82 357 82 357 82 82 101 82 357 ...
.. ..$ pclass : Factor w/ 5 levels "CONFIDENTIAL",..: 3 3 3 3 3 3 3 2 2 3 ...
.. ..$ class : Factor w/ 1 level "UNCLASSIFIED": 1 1 1 1 1 1 1 1 1 1 ...
.. ..$ classification: Factor w/ 4 levels "CONFIDENTIAL",..: 4 4 4 4 4 4 4 2 2 4 ...
.. ..$ btio : num [1:5598] 0 1 1 1 1 1 1 1 1 1 ...
.. ..$ senv : num [1:5598] 1 0 0 0 0 0 0 0 0 0 ...
.. ..$ year : int [1:5598] 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 ...
.. ..$ month : num [1:5598] 5 12 1 1 1 1 1 1 1 1 ...
..@ i : int [1:565913] 0 2 4 5 6 8 11 15 16 17 ...
..@ p : int [1:45797] 0 3083 3208 3273 3413 3826 4715 4734 4817 5355 ...
..@ Dim : int [1:2] 5598 45796
..@ Dimnames :List of 2
.. ..$ docs : chr [1:5598] "1976ABIDJA04853" "1976ABIDJA11744" "1976ABUDH00010" "1976ABUDH00031" ...
.. ..$ features: chr [1:45796] "on" "human" "settlements" "habitat" ...
..@ x : num [1:565913] 2 4 2 2 2 4 1 3 2 1 ...
..@ factors : list()
>
Environment
History
Connections
Import Dataset
List
Global Environment
Data
colloc 89670 obs. of 6 variables
con Formal class MySQLConnection
con2 Formal class MySQLConnection
dfmTR Large dfm (256366008 elements, 11.2 Mb)
dfmtr Large dfm (305359704 elements, 12.7 Mb)
e 177 obs. of 6 variables
f 304 obs. of 11 variables
results List of 2
t76 169 obs. of 25 variables
tfm Large dfm (305359704 elements, 18.1 Mb)
Files
Plots
Packages
Help
Viewer
Zoom
Export
Type here to search
3:59 PM
5/29/2018

Summary statistics

Center of data

Mean:

$$X = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Median: Midpoint of variable

If N is odd: $X_{\frac{(n+1)}{2}}$

If N is even: $\frac{1}{2}X_{(\frac{n}{2})} + \frac{1}{2}X_{(\frac{n}{2}+1)}$

Spread of data

Range : [min(X), max(X)]

Quantiles

- 25^{th} percentile = lower quartile
- 50^{th} percentile = median
- 75^{th} percentile = upper quartile

Interquartile range (IQR)

Measure of dispersion

- Difference between 75^{th} and 25^{th} percentiles

Used for box-whisker plots

- Represent whiskers:
  - Upper whisker: within 1.5*IQR of upper quantile
    - Upper quantile + 1.5*IQR
  - Lower whisker: within 1.5*IQR of lower quantile
    - Lower quantile – 1.5*IQR

Standard deviation

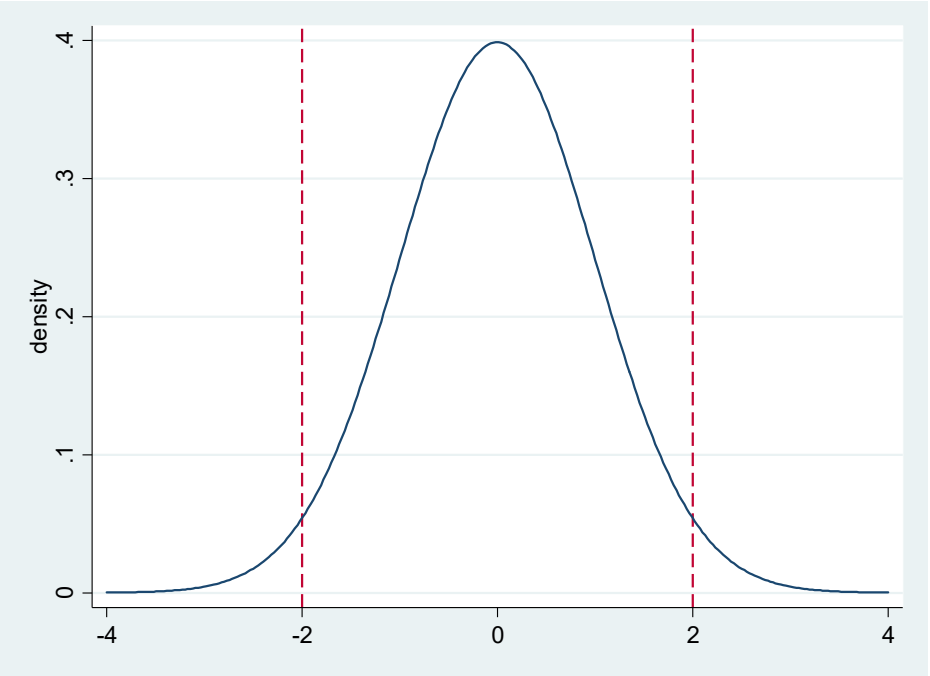
How far are data points away from their mean, on average?

$$s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

Variance = S_x^2

Normal Distribution

95% of observations within
2 standard deviations



Equivalents to sum, count, means?

More complicated

Almost like 3-dimensional data

- Document-feature as unique identifier
- But words themselves can be compared

Sum, count

Number of times feature appears

- In document
  - sum(dfm[1, ‘word’])
- In corpus
  - sum(dfm[, ‘word’])

Number of documents feature appears in

- sum(dfm[,’word’]>0)

Multiple columns

```
> sum(dfmTR[,"on"])
[1] 270
> sum(dfmTR[,c("on","the")])
[1] 2238
```

Does not work with multiple columns

Why?

Need to use apply for multiple columns

```
apply(dfmTR[,c("on","the")]>0, 2, sum )
```

```
on      the
270     1968
```

Quanteda shortcuts

of documents feature appears in:

- docfreq(dfmTR[,c("on","the")])

# of times feature appers

- topfeatures(dfmTR[,c('on','the')])

Both

- textstat_frequency(dfmTR[,c('on','the')])

Top features

Word	Count	Word	Count
the	1968	secretary	124
to	1007	soviet	92
of	994	president	70
and	728	soviets	68
that	659	position	63
in	648	might	62
a	498	oil	62
he	368	agreement	60
be	338	israel	59
was	291	problem	54

Text frequencies

Textstat_frequency*(DFM)*

Tables of most frequently used words

- Like topfeatures, but gives number of occurrences and number of documents word occurs in
- Can also ask for features by group

Different DFM measures

Proportion -> Document's share of feature's total mentions

Boolean -> Does the word appear in document?

Tf-idf -> Term frequency*inverse document frequency

- # times feature appears in doc * log(total doc/docs feature is in)

What do they have in common?

Different DFM measures

Proportion -> Document's share of feature's total mentions

Boolean -> Does the word appear in document?

Tf-idf -> Term frequency*inverse document frequency

- # times feature appears in doc * log(total doc/docs feature is in)

What do they have in common?

Do not depend on other features