Text Visualization Workshop

Yale School of Public Health

Josemari Feliciano

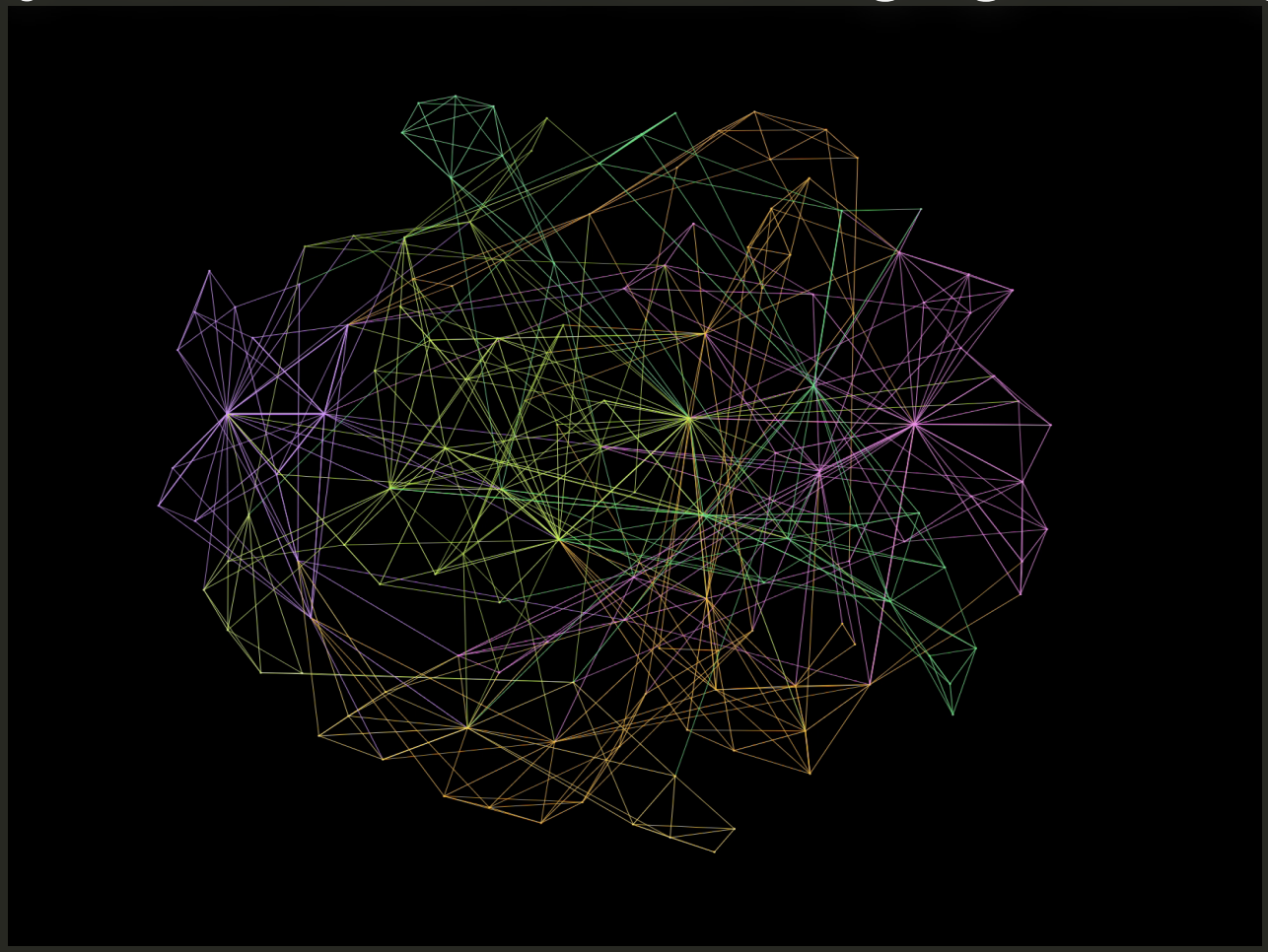
MPH in Biostatistics Candidate

2020/02/26

Text Visualization

- Developing a skillset in text visualization (text viz) is increasingly important as text data becomes more readily available in digital form.
- Tutorials on text viz tend to focus on politics using advanced tools in R such as dplyr and tidyr. The goal of this workshop is to impart you with a skillset using basic tools.
- One of the most common forms of text viz is word cloud.

My Interest: Networks-Based Language Modeling



DISCOVER DATASETS IN THE LIBRARY CATALOG

ROWENA GRIEM, TACHTORN MEIER, YUKARI SUGIYAMA

DID YOU KNOW YALE UNIVERSITY LIBRARY ACQUIRES DATASETS FOR ITS COLLECTION?

Datasets are collections of related information that can be manipulated by computer to identify patterns. As digital scholarship evolves in academia, there is a growing importance and increasing acquisition of datasets at Yale University Library. YUL has over 10,000 datasets ranging from statistical data to linguistics corpora, GIS data, and image datasets. While most of them are online datasets, some are available in CD-ROMs and hard drives.
Here are samples of some popular datasets:

Text datasets
Digital scholar lab
Linguistic Data Consortium collection

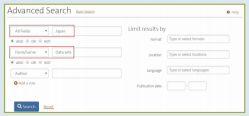
Image datasets
Digital scholar lab
Vogue archive dataset

Geospatial datasets
Geographic data for Lao PDR
IndiaMap 2011

Numeric datasets
Gallup poll of the Islamic world
ICPSR collection

HOW TO SEARCH FOR DATASETS IN THE LIBRARY CATALOG

① Searching for datasets is as easy as searching for books and journals. Go to Quicksearch (search.library.yale.edu). Enter a keyword search term or other search term in Books+, limiting by Form/Genre: "Data sets."



② You can also find or filter your search to a different type of datasets by combining "Data sets" with additional Form/Genres such as:

- Biostatistics
- Census data
- Death registers
- Geospatial data
- Images
- Judicial statistics
- Medical statistics
- Raster data
- Statistics
- Text corpora
- Vector data
- Vital statistics

③ Although most of our datasets are freely available to the Yale University community, some datasets are restricted by licensing agreement and/or agreement of terms of use, which would require consulting with library staff.

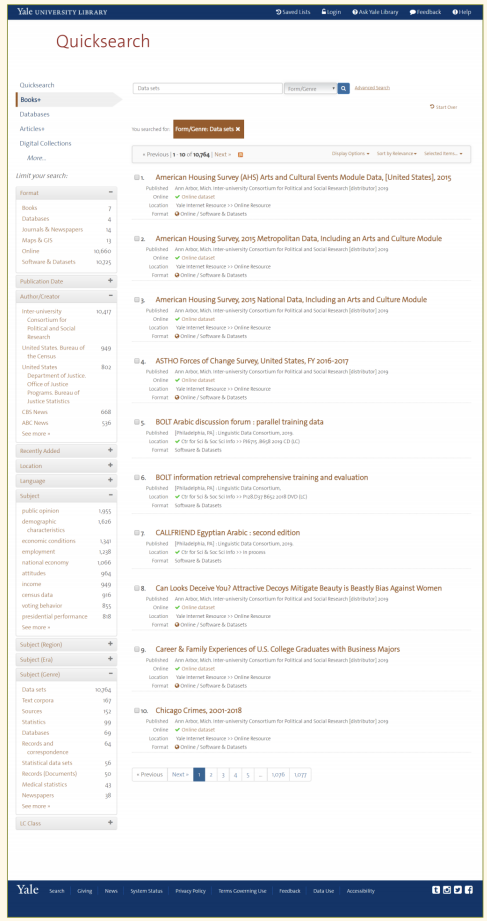
COMING SOON!

Datasets will soon be a separate format allowing you to easily and quickly use facets to narrow your search with one click.

NEED SUPPORT?

Contact researchdata@yale.edu with questions on using datasets and research data.
Check out YUL's StatLab and the Digital Humanities Lab websites for information on workshops, office hours, reference materials, and other resources.

- StatLab: http://statlab.yale.edu/
- Digital Humanities Lab: https://dhlab.yale.edu/



Library Resources

- Our library has tools on text analysis:

https://library.medicine.yale.edu/research-data/data-tools-software/textanalysis

- If you know advanced R (e.g. dplyr), I highly recommend the Silge and Robinson electronic book.

Relevant R Knowledge

- In R, we refer to this data structure as a data frame:

year	text
12	Dancing happily
14	Where did you go? Sad.

- Suppose `text_data` is the name of the said data frame.

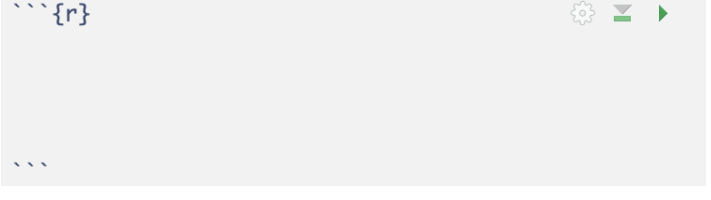
- **First goal:** Develop skills to access and manipulate data frames. It is fairly straight forward.

Running R Scripts in RStudio

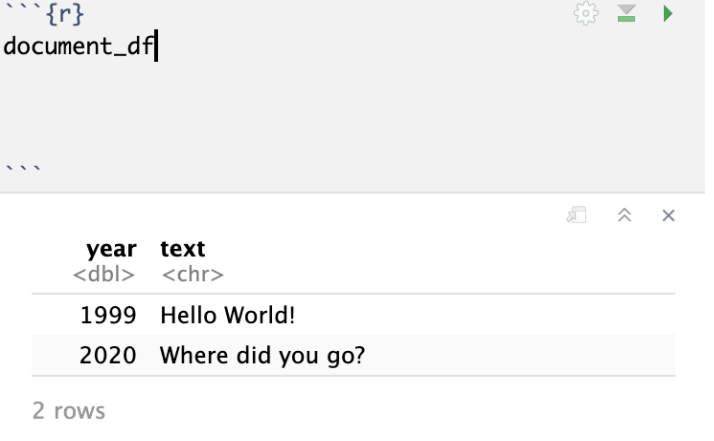
This slide will show you what a chunk container looks like where you are able to run scripts in RStudio.

The image below is an empty chunk in RStudio.

You may run any code you include by clicking the green button to the right.



The image below shows you the output after running the code with the name of an existing data frame.



Overriding a column of variables

The code below demonstrates a way to transform and override a column of variables:

```
text_data$year <- text_data$year + 2000
```

Before running code:

year	text
12	Dancing happily
14	Where did you go? Sad.

After running code:

year	text
2012	Dancing happily
2014	Where did you go? Sad.

Creating a new variable using current variables

- We will create a variable called `num_char` to store the number of characters for each text using the `nchar()` function. For simplicity, you may think of **functions** as tools to perform certain tasks.

```
text_data$num_char <- nchar(text_data$text)
```

You may examine the documentation for nchar by running the function name with a preceding `?` symbol:

```
?nchar
```

After running the script, RStudio will display the following documentation:

Creating a new variable. Revisited.

- We want to create a column with variable name 'num_char' to store the number of characters. `nchar()` can help us!

```
text_data$num_char <- nchar(text_data$text)
```

Before running code:

year	text
2012	Dancing happily
2014	Where did you go? Sad.

After running code:

year	text	num_char
2012	Dancing happily	15
2014	Where did you go? Sad.	22

Text Processing Considerations: Symbols

- Text processing often involves text matching that is case sensitive.
- To get around this, we can take advantage of `gsub()` which I argue is one of the best (underutilized) tools. The lab portion will give you more resources on regular expressions, what we typically use for pattern matching.

```
text_data$text <- gsub(pattern = "[[:punct:]]",
                       replacement = "",
                       x = text_data$text)
```

Before running code:

year	text	num_char
2012	Dancing happily	15
2014	Where did you go? Sad.	22

After running code:

year	text	num_char
2012	Dancing happily	15
2014	Where did you go Sad	22

Text Processing Considerations: Case Sensitivity

To get around this, we can take advantage of the `tolower()` function. This is particularly important as sentiment analysis workflows tend to rely on exact word matching to match words with their assigned sentiments.

```
text_data$text <- tolower(text_data$text)
```

Before running code:

year	text	num_char
2012	Dancing happily	15
2014	Where did you go Sad	22

After running code:

year	text	num_char
2012	dancing happily	15
2014	where did you go sad	22

Tokenization

- Tokenization is the process of breaking down text data into smaller components.
- Individual words can be the tokens. Pairs of words can be the tokens. Sentences can be the tokens. You get to define this.

A typical representation after tokenization:

year	text
2012	dancing
2012	happily
2014	where
2014	did
2014	you
2014	go

- We started with a 2x2 data frame.
  - With tokenization, we now have a 6x2.
  - Computing Nightmare: high dimensional issue.

Basic Sentiment Analysis

- There are preexisting lexicons you may use to assign sentiment to your text data.
- Here's a preview of the first five rows of the Bing sentiment lexicon:

word	sentiment
2-faces	negative
abnormal	negative
abolish	negative
abominable	negative
abominably	negative

Source: Minqing Hu and Bing Liu, Mining and summarizing customer reviews, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004), Seattle, Washington, USA, Aug 22-25, 2004.

Embedding Sentiments into your Data

- You may use the `merge()` function to combine your current data with the sentiments data:

```
##      text year sentiment
## 1 dancing 2012      <NA>
## 2     did 2014      <NA>
## 3      go 2014      <NA>
## 4 happily 2012  positive
## 5     sad 2014  negative
## 6   where 2014      <NA>
## 7     you 2014      <NA>
```

`

More Tools to Explore

- `wordcloud` package to make word clouds.
- `topicmodels` package to perform latent Dirichlet allocation for topic modelling.
- In terms of deep learning tools, recurrent neutral networks (RNNs) are widely used.
- **We now proceed to the lab portion of the workshop.**