



IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY

information technology & management

529 Advanced Data Analytics

November 8, 10 2016

Weekly 12 Presentation

Week 12 Topic: Agenda

- ◆ Discussion Topic Review – Sentiment Analysis

Set up for Week 12 Assignment – due 11/19:

- ◆ Streaming API example
- ◆ Full Text Corpus
- ◆ Tweeter Mood + Stock Market

Setting up for Week 13 Assignment – due 12/1:

- ◆ Setting up Big Data + Hadoop

Week 12 Topic:

Week 11 Assignment – due 11/12

Use your sentiment analysis scoring function to determine the sentiment of your chosen topic for the past week:

- 1) Collect sampling of tweets per day (7 days total)
- 2) If performing Trump and/or Clinton analysis, collect for (11/5 ~ 11/11)
- 3) Determine the daily mood.
- 4) Illustrate with trends and charts.

Week 12 Topic:

Continuing with our Trump example

Make sure to remove all emojis etc. with `corpus <- iconv(corpus, 'UTF-8', 'ASCII', sub='')`. Then, create corpus to reset format `corpus=Corpus(VectorSource(corpus))`

The screenshot shows the RStudio interface with the following components:

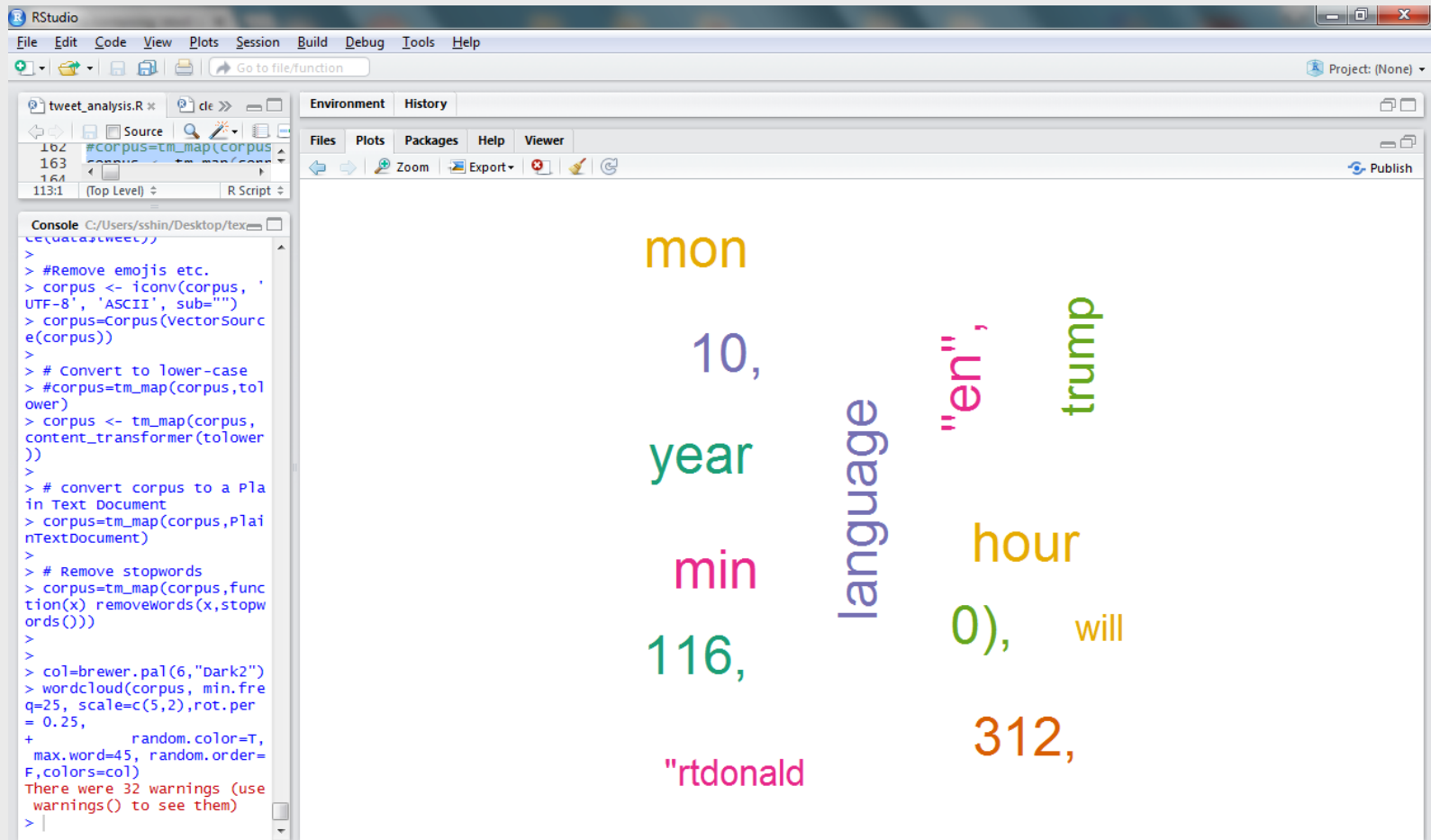
- Source Editor:** Contains a script named `tweet_analysis.R` with the following code:


```
106 donald=do.call(rbind,lapply(1:length(lats), function(i) searchTwitter('Donald+Trump',
112:1 t",
                                lang="en",n=N,resultType="recen
                                geocode=paste(lats[i],lons[i],past
                                e0(s,"mi"),sep=","))))
```
- Console:** Displays the execution output, showing multiple "Rate limited" warnings and a final warning message:


```
warning messages:
1: In doRppAPICall("search/tweets", n, params = params, retryonRateLimit = retryonRateLimit, :
  2000 tweets were requested but the API can only return 1364
2: In (function (..., deparse.level = 1) :
  number of columns of result is not a multiple of vector length (arg 27)
> length(donald)
[1] 60000
> write.csv(donald, "donald.csv",row.names = F)
>
```
- Environment:** Lists the objects in the global environment:
 - `donaldlon`: num [1:6000] NA NA NA NA NA NA ...
 - `donaldtext`: List of 6
 - `dtm`: List of 6
 - `dtms`: List of 6
 - `favoritecou...`: num [1:6000] 0 0 0 0 0 0 0 0 ...
 - `favorited`: logi [1:6000] FALSE FALSE FALSE...
 - `freg`: Named num [1:5] 2763 2388 2220 ...
 - `freq`: Named num [1:336] 1549 595 493 ...
 - `good_text`: chr [1:2010] "a+" "abound" "abo...
 - `isretweet`: logi [1:6000] TRUE TRUE FALSE T...
 - `key`: "midndxApMszw5P2RmyjnoRXsA"
 - `lats`: num [1:30] 38.9 40.7 37.8 39 37...
 - `Local`: List of 395
 - `lons`: num [1:30] -77 -74 -122 -106 -1...
- Plots:** Displays a word cloud visualization of the text data, with prominent words including "will", "new", "language", "year", "hour", "meta", "52", "10", "311", "rdonald", and "line".

Week 12 Topic:

Trump word cloud of 2000 tweets



Week 12 Topic: Streaming API

<http://stackoverflow.com/questions/37961374/how-to-collect-tweets-related-to-stock-market-analysis/37961891#37961891>

```
#Import the necessary methods from tweepy library  
from tweepy.streaming import StreamListener  
from tweepy import OAuthHandler  
from tweepy import Stream
```

```
#Variables that contains the user credentials to access Twitter API  
access_token = "Acces_Token"  
access_token_secret = "Access_Token_Secret"  
consumer_key = "Consumer_Key"  
consumer_secret = "Consumer_Secret"
```

```
f=open("C:\\Users\\Surya's\\Desktop\\tweet.txt","r+")  
#This is a basic listener that just prints received tweets to stdout.
```

Week 12 Topic:

Streaming API (cont.)

```
class StdOutListener(StreamListener):
```

```
def on_data(self, data): f.write(data) return True
```

```
def on_error(self, status): f.write(status) if __name__ == '__main__':
```

```
#This handles Twitter authentication and the connection to Twitter Streaming API l =  
StdOutListener()
```

```
auth = OAuthHandler(consumer_key, consumer_secret)
```

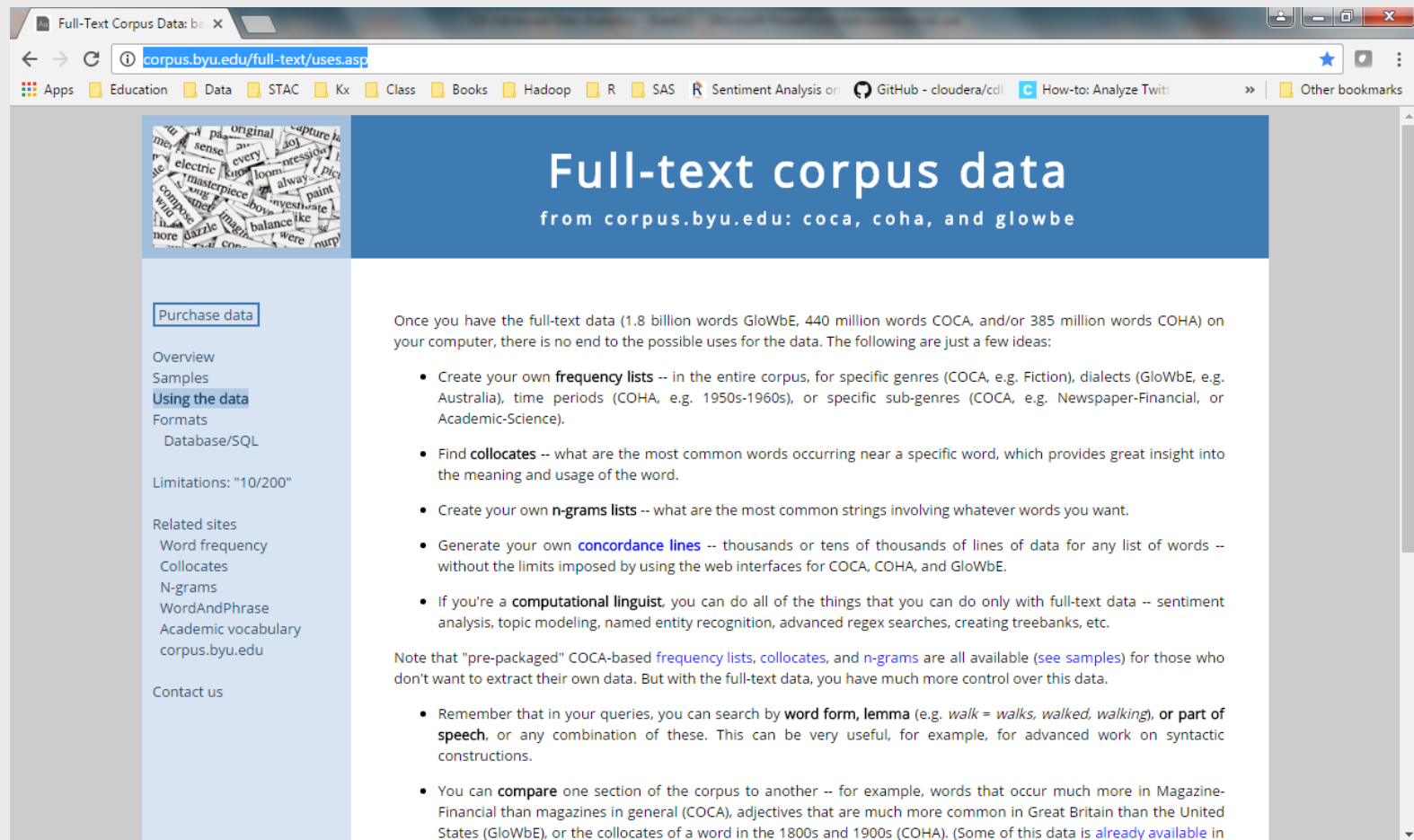
```
auth.set_access_token(access_token, access_token_secret)
```

```
stream = Stream(auth, l)
```

```
#This line filter Twitter Streams to capture data by the keywords: 'python', 'javascript', 'ruby'  
stream.filter(track=['#Mourinho', '#MUFC'])
```

Week 12 Topic: Full Text Corpus

<http://corpus.byu.edu/full-text/uses.asp>



The screenshot shows a web browser window with the address bar displaying corpus.byu.edu/full-text/uses.asp. The browser's bookmark bar includes links for Apps, Education, Data, STAC, Kx, Class, Books, Hadoop, R, SAS, Sentiment Analysis on GitHub - cloudera/cdl, How-to: Analyze Twitter, and Other bookmarks. The website's header features a word cloud on the left and a blue banner with the text "Full-text corpus data from corpus.byu.edu: coca, coha, and glowbe". A left sidebar contains navigation links: Purchase data, Overview, Samples, Using the data (highlighted), Formats, Database/SQL, Limitations: "10/200", Related sites, Word frequency, Collocates, N-grams, WordAndPhrase, Academic vocabulary, corpus.byu.edu, and Contact us. The main content area has a paragraph explaining the data and a list of five ideas for using it. The list includes creating frequency lists, finding collocates, creating n-grams lists, generating concordance lines, and applying computational linguistics. A note mentions that pre-packaged data is available for those who don't want to extract their own. A final list item suggests comparing sections of the corpus.

Full-text corpus data

from corpus.byu.edu: coca, coha, and glowbe

Purchase data

Overview
Samples
Using the data
Formats
Database/SQL
Limitations: "10/200"
Related sites
Word frequency
Collocates
N-grams
WordAndPhrase
Academic vocabulary
corpus.byu.edu
Contact us

Once you have the full-text data (1.8 billion words GloWbE, 440 million words COCA, and/or 385 million words COHA) on your computer, there is no end to the possible uses for the data. The following are just a few ideas:

- Create your own **frequency lists** -- in the entire corpus, for specific genres (COCA, e.g. Fiction), dialects (GloWbE, e.g. Australia), time periods (COHA, e.g. 1950s-1960s), or specific sub-genres (COCA, e.g. Newspaper-Financial, or Academic-Science).
- Find **collocates** -- what are the most common words occurring near a specific word, which provides great insight into the meaning and usage of the word.
- Create your own **n-grams lists** -- what are the most common strings involving whatever words you want.
- Generate your own **concordance lines** -- thousands or tens of thousands of lines of data for any list of words -- without the limits imposed by using the web interfaces for COCA, COHA, and GloWbE.
- If you're a **computational linguist**, you can do all of the things that you can do only with full-text data -- sentiment analysis, topic modeling, named entity recognition, advanced regex searches, creating treebanks, etc.

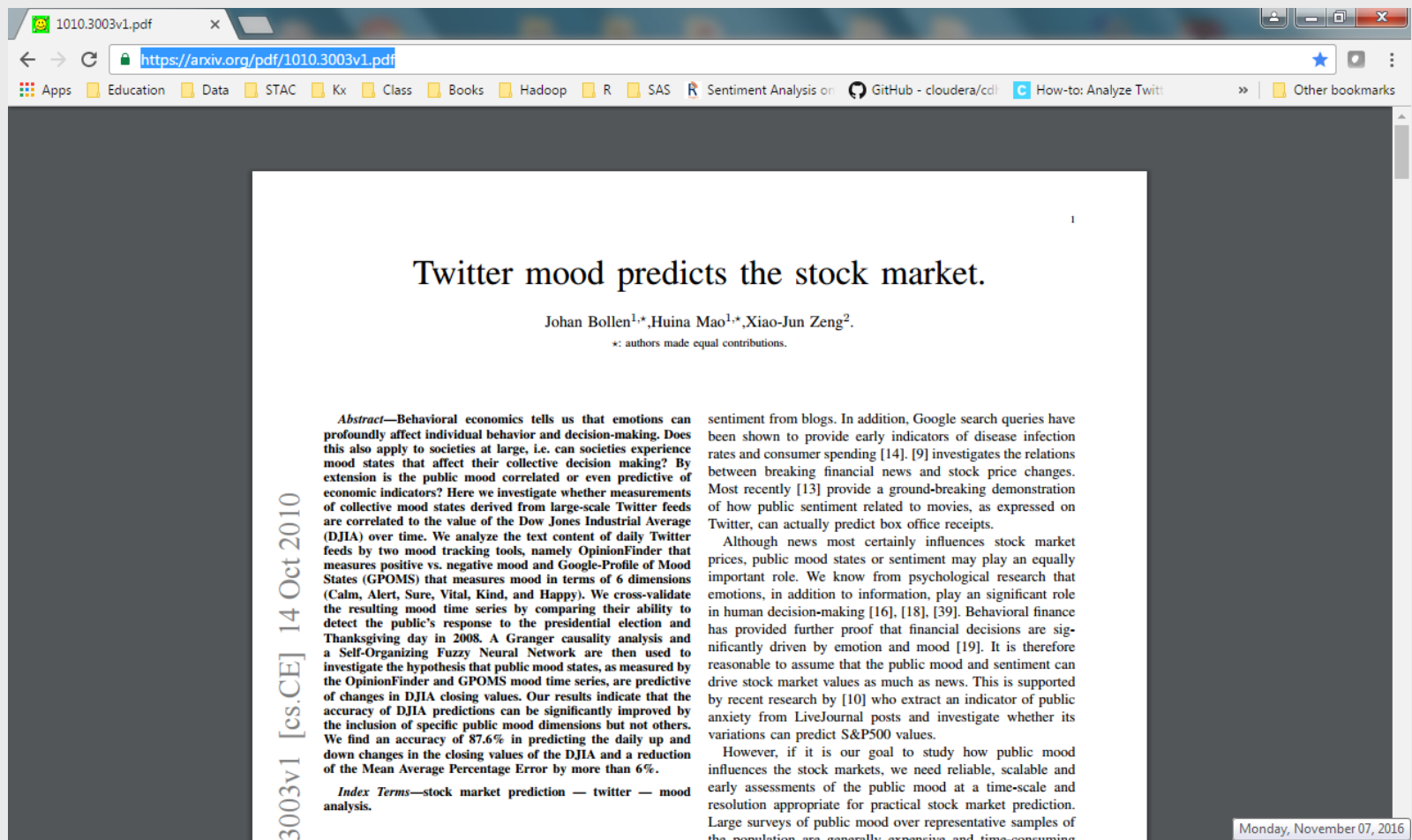
Note that "pre-packaged" COCA-based **frequency lists**, **collocates**, and **n-grams** are all available ([see samples](#)) for those who don't want to extract their own data. But with the full-text data, you have much more control over this data.

- Remember that in your queries, you can search by **word form**, **lemma** (e.g. *walk* = *walks*, *walked*, *walking*), or **part of speech**, or any combination of these. This can be very useful, for example, for advanced work on syntactic constructions.
- You can **compare** one section of the corpus to another -- for example, words that occur much more in Magazine-Financial than magazines in general (COCA), adjectives that are much more common in Great Britain than the United States (GloWbE), or the collocates of a word in the 1800s and 1900s (COHA). (Some of this data is [already available](#) in

Week 12 Topic:

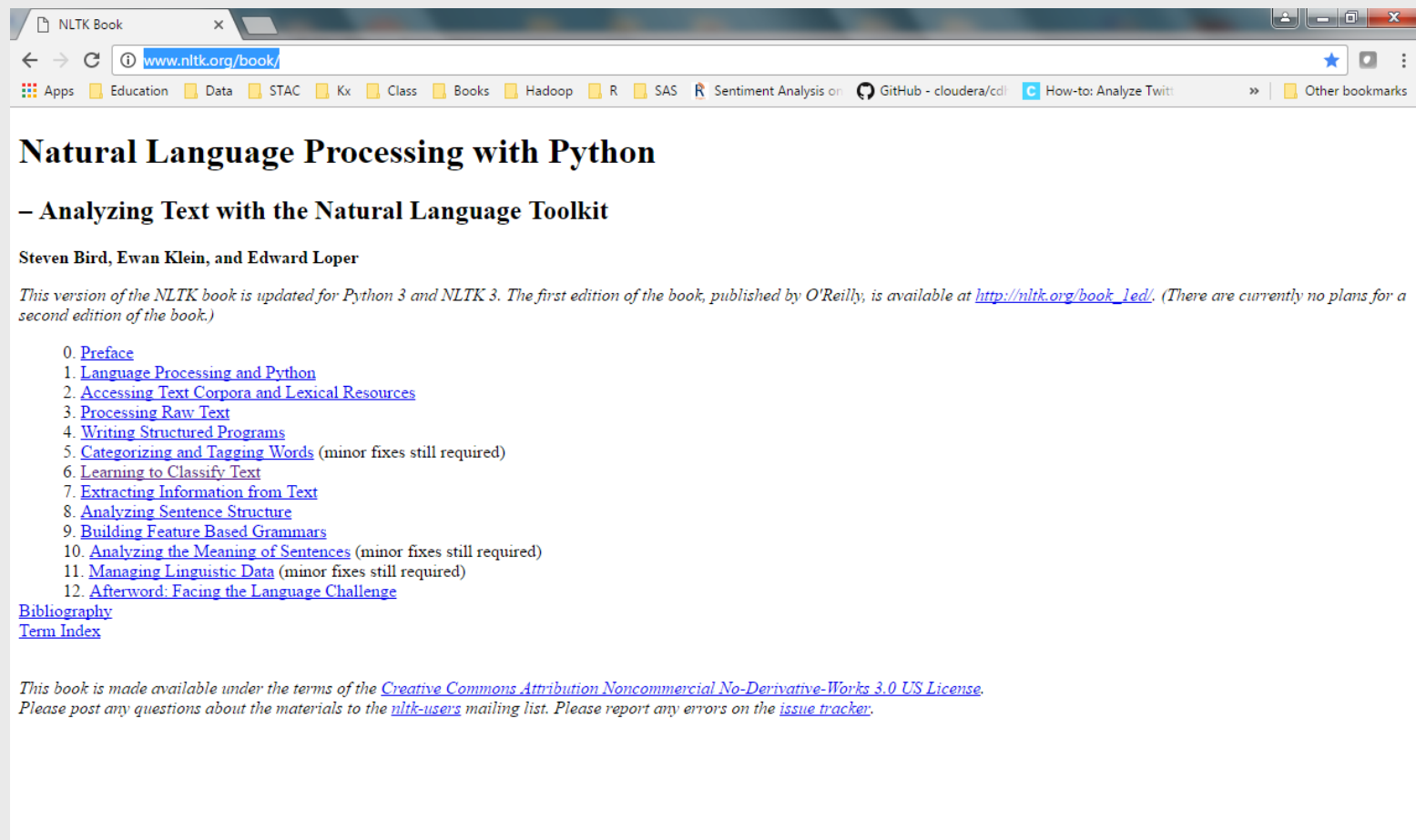
Tweeter Mood + Stock Market

<https://arxiv.org/pdf/1010.3003v1.pdf>



Week 12 Topic: NLTK

<http://www.nltk.org/book/>

A screenshot of a web browser displaying the NLTK Book website. The browser's address bar shows 'www.nltk.org/book/'. The page title is 'Natural Language Processing with Python'. Below the title is the subtitle '– Analyzing Text with the Natural Language Toolkit' and the authors 'Steven Bird, Ewan Klein, and Edward Loper'. A paragraph of text states: 'This version of the NLTK book is updated for Python 3 and NLTK 3. The first edition of the book, published by O'Reilly, is available at http://nltk.org/book_1ed/. (There are currently no plans for a second edition of the book.)'. A list of 12 numbered links follows: 0. Preface, 1. Language Processing and Python, 2. Accessing Text Corpora and Lexical Resources, 3. Processing Raw Text, 4. Writing Structured Programs, 5. Categorizing and Tagging Words (minor fixes still required), 6. Learning to Classify Text, 7. Extracting Information from Text, 8. Analyzing Sentence Structure, 9. Building Feature Based Grammars, 10. Analyzing the Meaning of Sentences (minor fixes still required), 11. Managing Linguistic Data (minor fixes still required), 12. Afterword: Facing the Language Challenge. Below the list are links for 'Bibliography' and 'Term Index'. At the bottom, a paragraph states: 'This book is made available under the terms of the Creative Commons Attribution Noncommercial No-Derivative-Works 3.0 US License. Please post any questions about the materials to the nltk-users mailing list. Please report any errors on the issue tracker.'

NLTK Book

← → ↻ www.nltk.org/book/

Apps Education Data STAC Kx Class Books Hadoop R SAS Sentiment Analysis on GitHub - cloudera/cdl How-to: Analyze Twit Other bookmarks

Natural Language Processing with Python

– Analyzing Text with the Natural Language Toolkit

Steven Bird, Ewan Klein, and Edward Loper

This version of the NLTK book is updated for Python 3 and NLTK 3. The first edition of the book, published by O'Reilly, is available at http://nltk.org/book_1ed/. (There are currently no plans for a second edition of the book.)

- 0. [Preface](#)
- 1. [Language Processing and Python](#)
- 2. [Accessing Text Corpora and Lexical Resources](#)
- 3. [Processing Raw Text](#)
- 4. [Writing Structured Programs](#)
- 5. [Categorizing and Tagging Words](#) (minor fixes still required)
- 6. [Learning to Classify Text](#)
- 7. [Extracting Information from Text](#)
- 8. [Analyzing Sentence Structure](#)
- 9. [Building Feature Based Grammars](#)
- 10. [Analyzing the Meaning of Sentences](#) (minor fixes still required)
- 11. [Managing Linguistic Data](#) (minor fixes still required)
- 12. [Afterword: Facing the Language Challenge](#)

[Bibliography](#)
[Term Index](#)

This book is made available under the terms of the [Creative Commons Attribution Noncommercial No-Derivative-Works 3.0 US License](#). Please post any questions about the materials to the [nltk-users](#) mailing list. Please report any errors on the [issue tracker](#).

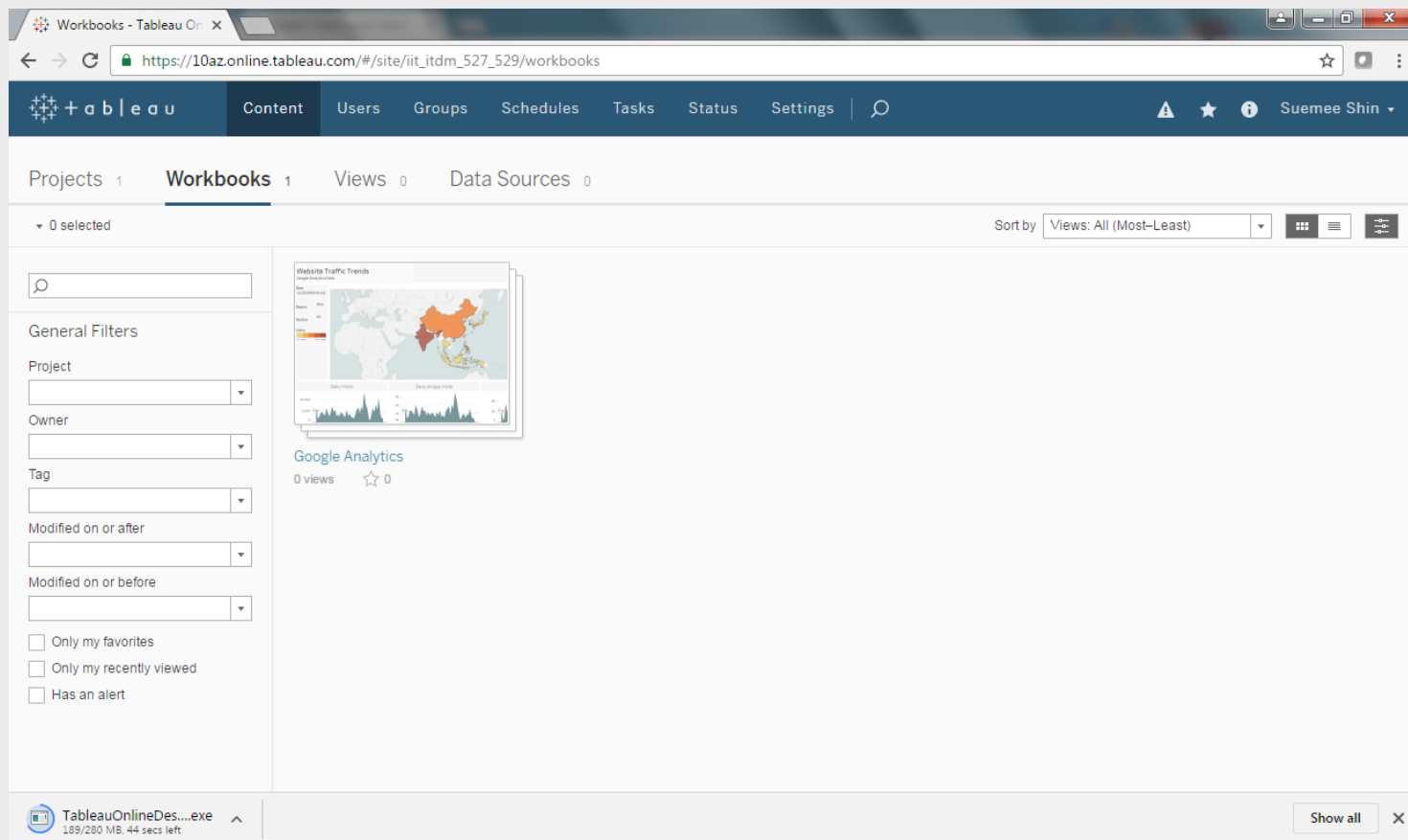
Week 12 Topic:

Continuing w Tableau

- ◆ **Save the data as csv file and import it to Tableau**
- ◆ The map below shows the tweets that I was able to reverse geocode. The size is proportional to the number of favorites each tweet got. In the interactive map, we can hover over each circle and read the tweet, the address it was tweeted from, and the date and time it was posted.

Week 12 Topic: Tableau Install

◆ Follow directions and install the trial version:



Week 12 Topic:

Watch the Getting started videos


◆ Class exercise submissions will be publications from Tableau

Tableau Online Download

www.tableau.com/products/online/thanks-dl

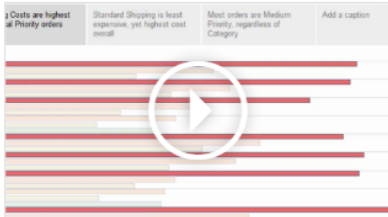
Use Tableau Desktop to connect to data and build dashboards.

Share and collaborate with Tableau Online.



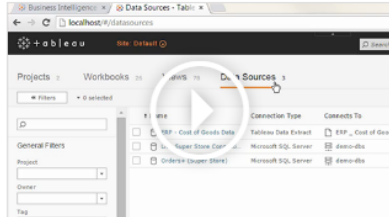
23:05 MIN

Getting Started



2:50 MIN

Publishing to Online



4:55 MIN

Online Administrative Overview

Become a Tableau master with more than 10 hours of free video content through Tableau Learning.

EXPLORE LEARNING →

Thursday, October 27, 2016