Big Data: Data Wrangling Boot Camp
Python Sentiment Analysis

Chuck Cartledge, PhD

28 January 2017

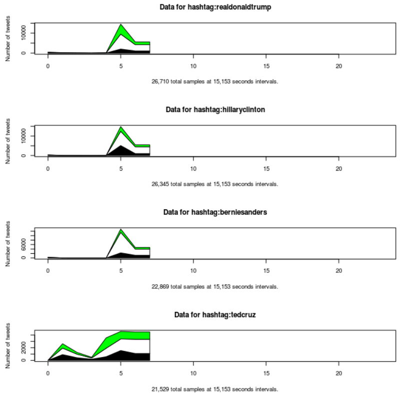
Table of contents I

What are we going to cover?

- Look to the future
- Talk briefly about sentiment analysis
- Address the polyglot of computer languages
- Talk about our sentiment analysis system
- **Data wrangle** tweets using Python

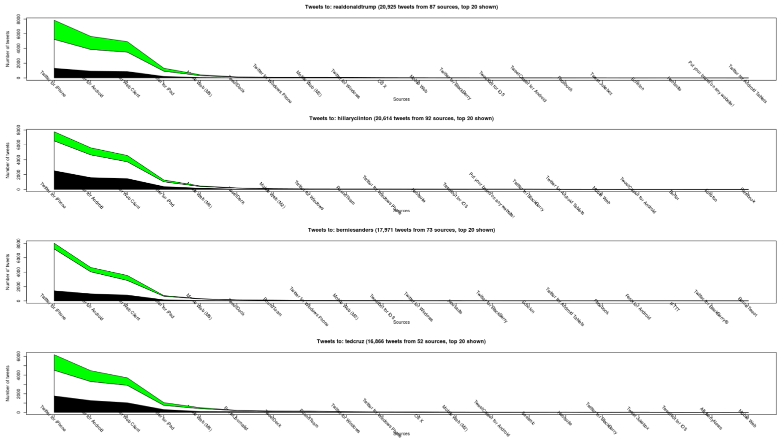
Things that we will be doing.

1. **Data wrangle** tweets using python
2. Conduct sentiment analysis on tweets
3. Look at the sentiments in different ways

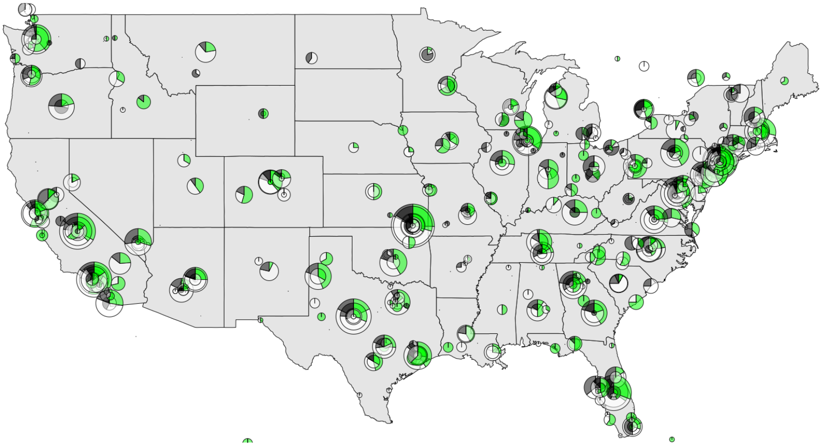


Sentiments over time.

Sentiment by sending device



Sentiment by geographic location

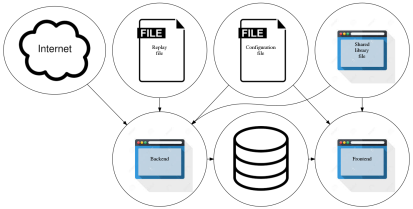


How we'll get to the images

We'll walk before we run.

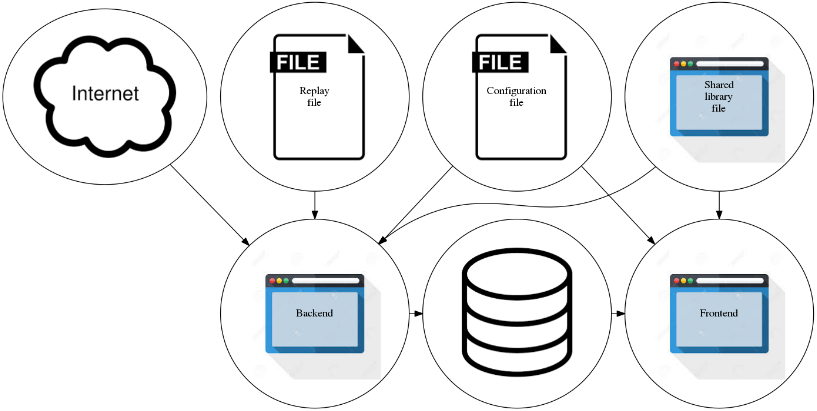
- Start with a replay file
- **Data wrangle** using the library file
- Go live and download live tweets

Data flows through the backend, into the database, out the frontend.



The software design document (attached) contains lots of details.

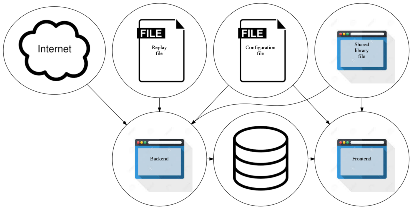
Same image.



The software design document (attached) contains lots of details.

What some of the files do:

- Replay file – previously recorded tweets
- Configuration file – directives used by both backend and frontend script files
- Shared library file – routines that are common to backend and frontend script files
- Backend file – script file that populates the database from the replay file, or the Internet
- Frontend file – script file that extracts data from the database and presents results



A working definition

"Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Sentiment analysis is widely applied to reviews and social media for a variety of applications, ranging from marketing to customer service."

W. Staff [3]

More formal definitions

"The field of opinion mining and sentiment analysis is well-suited to various types of intelligence applications. Indeed, business intelligence seems to be one of the main factors behind corporate interest in the field."

Pang and Lee [2]

"Sentiment analysis, also called opinion mining, is the field of study that analyzes peoples opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes."

Liu [1]

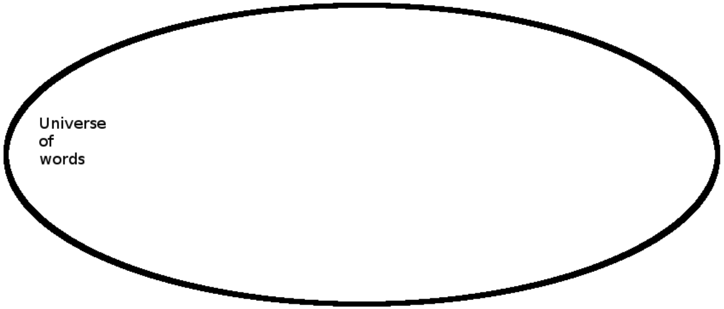
Our approach to sentiment analysis

We will:

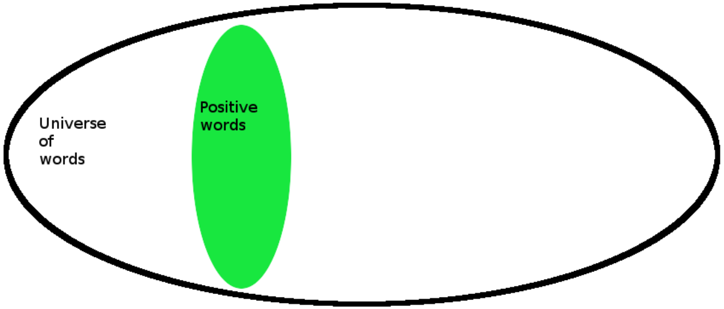
1. Search the “twitterverse” for tweets using specific hashtags
2. Tokenize each tweet
3. **Data wrangle** each token
4. Remove all stop words from the tokens
5. Count number of positive and negative tokens
6. Compute the positive, negative, or neutral sentiment for the tokens
7. Display the results

Our approach is language agnostic.

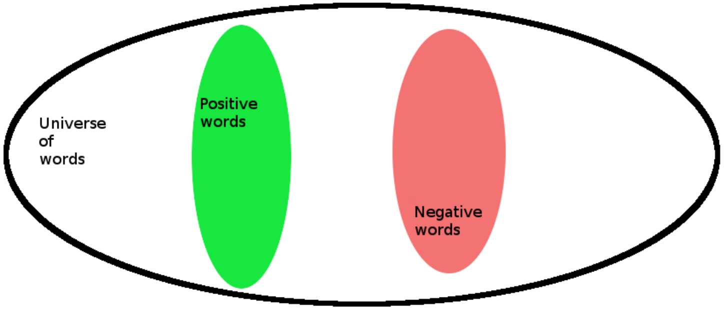
A “Universe” of words



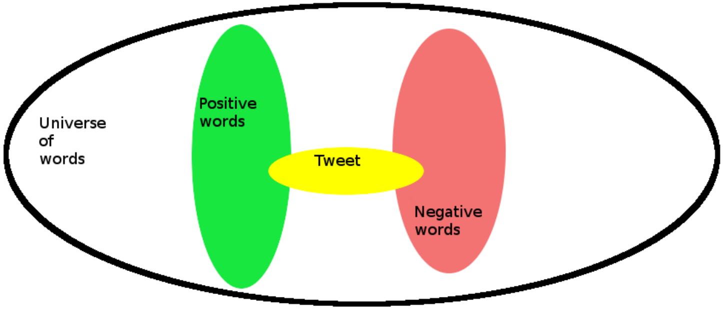
Some words are “Positive”



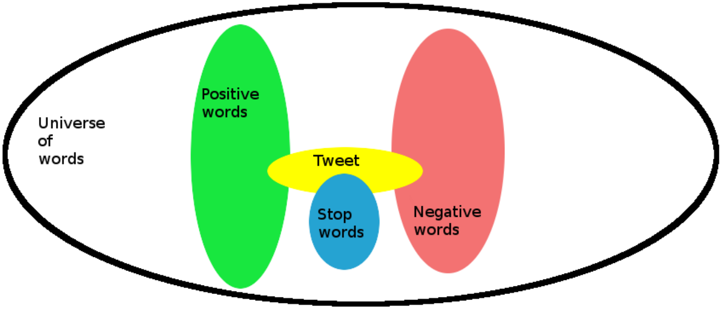
Some words are “Negative”



A tweet will/may have positive and negative words



Some words we don't care about



Mechanically this is what we are doing

The steps are:

1. Break the tweet into tokens
2. Remove stop words from the tokens
3. Compute the percentage of remaining tweet tokens that are positive
4. Compute the percentage of remaining tweet tokens that are negative
5. Classify the tokens as positive, negative, or neutral

Mathematically this is what we are doing

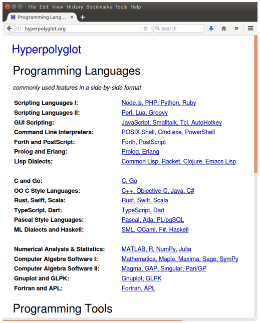
The steps are:

$$\begin{aligned} tokens &= \{\text{words in tweet}\} \\ tokensLessStop &= tokens - stopWords \\ positivePart &= positiveWords \cap tokensLessStop \\ negativePart &= negativeWords \cap tokensLessStop \end{aligned}$$

$$classification = \begin{cases} positive, & \begin{array}{l} \text{if} positiveThreshold \leq \frac{positivePart}{tokensLessStop} \\ \text{AND} \\ \frac{negativePart}{tokensLessStop} < negativeThreshold \end{array} \\ negative, & \begin{array}{l} \text{if} negativeThreshold \leq \frac{negativePart}{tokensLessStop} \\ \text{AND} \\ \frac{positivePart}{tokensLessStop} < positiveThreshold \end{array} \\ neutral, & \text{otherwise} \end{cases}$$

Comparing a known language with Python and R

A specific language was not a requirement for this boot camp. Python and R used in boot-camp. Too many languages to compare to Python and R.


File Edit View History Bookmarks Tools Help

Programming Lang...

hyperpolyglot.org

Search

Hyperpolyglot

Programming Languages

commonly used features in a side-by-side format

Scripting Languages I:	Node.js, PHP, Python, Ruby
Scripting Languages II:	Perl, Lua, Groovy
GUI Scripting:	JavaScript, Smalltalk, Tcl, AutoHotkey
Command Line Interpreters:	POSIX Shell, Cmd.exe, PowerShell
Forth and PostScript:	Forth, PostScript
Prolog and Erlang:	Prolog, Erlang
Lisp Dialects:	Common Lisp, Racket, Clojure, Emacs Lisp
C and Go:	C, Go
OO C Style Languages:	C++, Objective-C, Java, C#
Rust, Swift, Scala:	Rust, Swift, Scala
TypeScript, Dart:	TypeScript, Dart
Pascal Style Languages:	Pascal, Ada, PL/pgSQL
ML Dialects and Haskell:	SML, OCaml, F#, Haskell
Numerical Analysis & Statistics:	MATLAB, R, NumPy, Julia
Computer Algebra Software I:	Mathematica, Maple, Maxima, Sage, SymPy
Computer Algebra Software II:	Magma, GAP, Singular, Pari/GP
Gnuplot and GLPK:	Gnuplot, GLPK
Fortran and APL:	Fortran, APL

Programming Tools

`http://hyperpolyglot.org/`

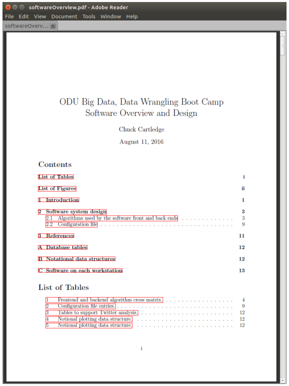
Use Hyperploygit.org as a cross reference.

A software design document

The document contains:

1. Overall system design
2. Algorithms used through out the system
3. Details about the database tables
4. Details about the database tables

The file is attached.


softwareOverview.pdf - Adobe Reader
File Edit View Document Tools Window Help
softwareOverv...

ODU Big Data, Data Wrangling Boot Camp
Software Overview and Design

Chuck Cartledge

August 11, 2016

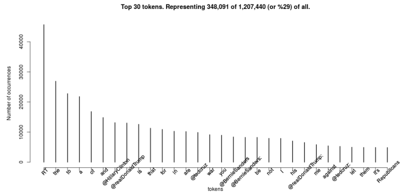
Contents

List of Tables

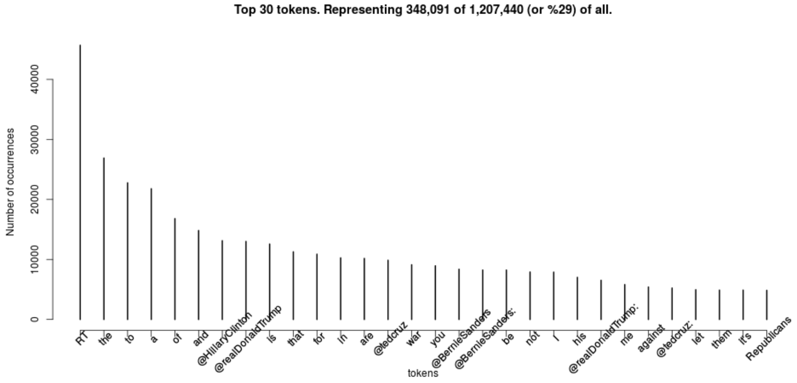
i

What will happen to the tweets when we do nothing.

A histogram of how often a token occurs.

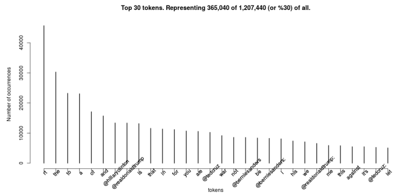


Same image.

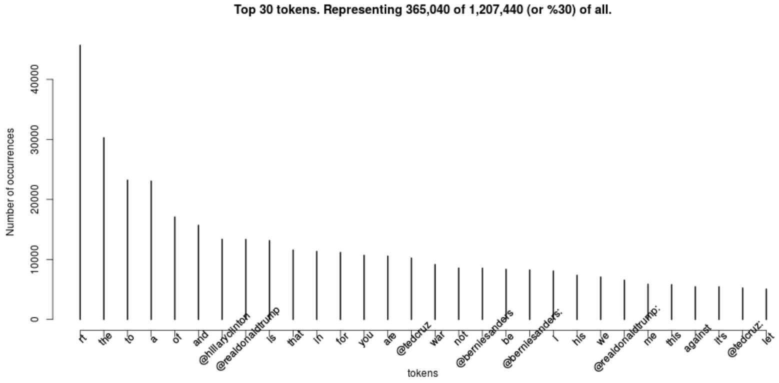

Top 30 tokens. Representing 348,091 of 1,207,440 (or %29) of all.
Number of occurrences
0
10000
20000
30000
40000
tokens
RT
the
to
a
of
and
@HillaryClinton
@realDonaldTrump
is
that
for
in
are
@tedcruz
war
you
@BernieSanders
@BernieSanders:
be
not
I
his
@realDonaldTrump:
me
against
@tedcruz:
let
them
it's
Republicans

What will happen to the tweets when we make everything the same case

Changing case is easy, unless they are emojois.

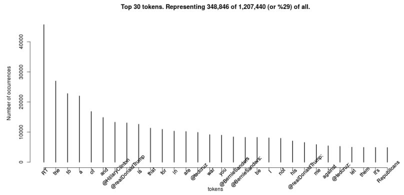


Same image.

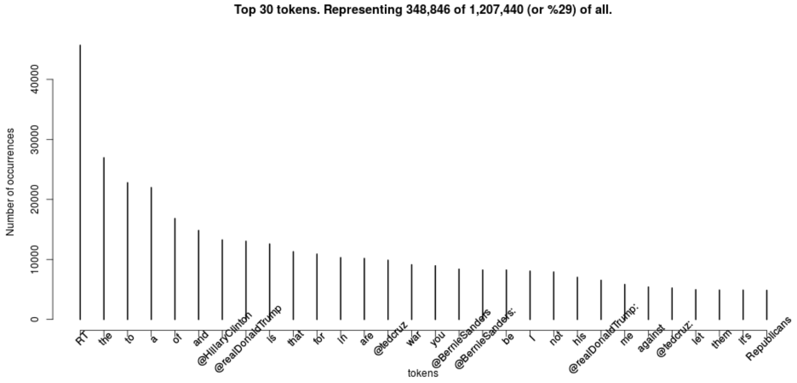

Top 30 tokens. Representing 365,040 of 1,207,440 (or %30) of all.
Number of occurrences
0
10000
20000
30000
40000
tokens
rt
the
to
a
of
and
@hillaryclinton
@realdonaldtrump
is
that
in
for
you
are
@tedcruz
war
not
@berniesanders
be
@berniesanders:
i
his
we
@realdonaldtrump:
me
this
against
it's
@tedcruz:
let

What will happen to the tweets we remove non-ASCII

We need to talk about ASCII and the rest of the character systems.

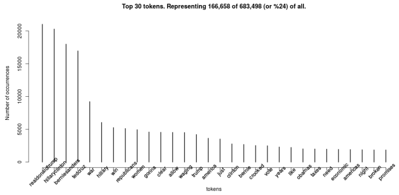


Same image.

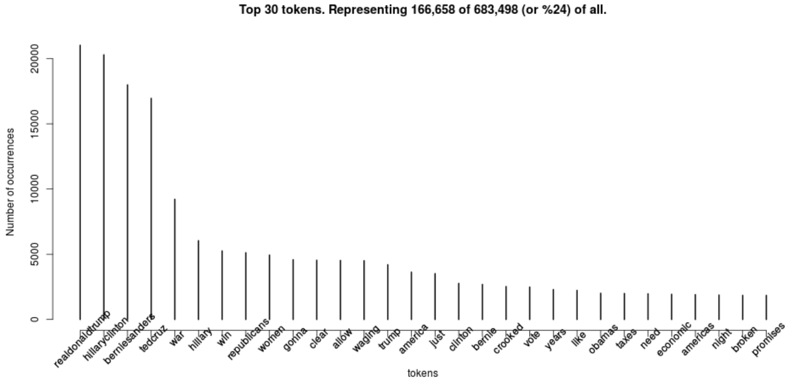

Top 30 tokens. Representing 348,846 of 1,207,440 (or %29) of all.
Number of occurrences
0
10000
20000
30000
40000
tokens
RT
the
to
a
of
and
@HillaryClinton
@realDonaldTrump
is
that
for
in
are
@tedcruz
war
you
@BernieSanders
@BernieSanders:
be
I
not
his
@realDonaldTrump:
me
against
@tedcruz:
let
them
it's
Republicans

What will happen to the tweets when we remove "stop words"

Stop words are words/tokens that have no use in whatever we are doing. Stop words are domain specific.
And we change case, remove non-ASCII, remove punctuation, etc.



Same image.

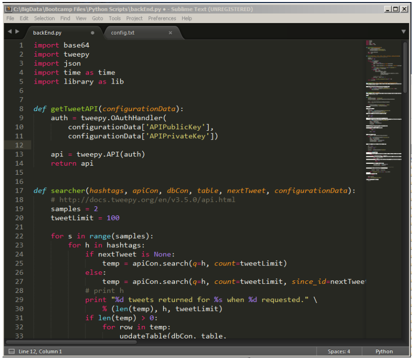

Top 30 tokens. Representing 166,658 of 683,498 (or %24) of all.
Number of occurrences
0
5000
10000
15000
20000
tokens
realdonaldtrump
hillaryclinton
berniesanders
tedcruz
war
hillary
win
republicans
women
gonna
clear
allow
waging
trump
america
just
clinton
bernie
crooked
vote
years
like
obamas
taxes
need
economic
americas
night
broken
promises

A complete IDE

A complete Python integrated develoment environment (IDE)

1. Text editor
2. Python console
3. Extensible
4. Tabbed display for multiple files

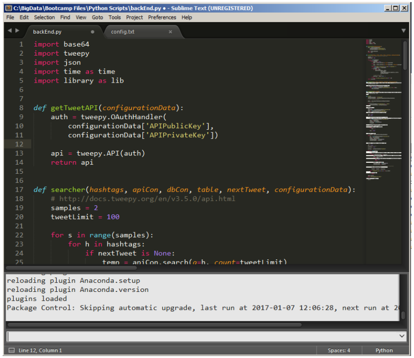
We incorporate the Anaconda plugin for additional functionality.


C:\BigData\Bootcamp Files\Python Scripts\backEnd.py • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
backEnd.py
config.txt
Line 12, Column 1
Spaces: 4
Python

```
import base64
import tweepy
import json
import time as time
import library as lib


def getTweetAPI(configurationData):
    auth = tweepy.OAuthHandler(
        configurationData['APIPublicKey'],
        configurationData['APIPrivateKey'])

    api = tweepy.API(auth)
    return api


def searcher(hashtags, apiCon, dbCon, table, nextTweet, configurationData):
    # http://docs.tweepy.org/en/v3.5.0/api.html
    samples = 2
    tweetLimit = 100

    for s in range(samples):
        for h in hashtags:
            if nextTweet is None:
                temp = apiCon.search(q=h, count=tweetLimit)
            else:
                temp = apiCon.search(q=h, count=tweetLimit, since_id=nextTwee
            # print h
            print "%d tweets returned for %s when %d requested." \
                % (len(temp), h, tweetLimit)
            if len(temp) > 0:
                for row in temp:
                    updateTable(dbCon, table,
```

Editor

- "Smart" editor
- CTRl + O to open a file
- CTRl + S to save a file
- CTRl + B to run a file
- Multiple files can be opened at once


C:\BigData\Bootcamp Files\Python Scripts\backEnd.py • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
backEnd.py
config.txt

```python
import base64
import tweepy
import json
import time as time
import library as lib


def getTweetAPI(configurationData):
    auth = tweepy.OAuthHandler(
        configurationData['APIPublicKey'],
        configurationData['APIPrivateKey'])

    api = tweepy.API(auth)
    return api


def searcher(hashtags, apiCon, dbCon, table, nextTweet, configurationData):
    # http://docs.tweepy.org/en/v3.5.0/api.html
    samples = 2
    tweetLimit = 100

    for s in range(samples):
        for h in hashtags:
            if nextTweet is None:
                temp = apiCon.search(q=h, count=tweetLimit)
```

```
reloading plugin Anaconda.setup
reloading plugin Anaconda.version
plugins loaded
Package Control: Skipping automatic upgrade, last run at 2017-01-07 12:06:28, next run at 2
```

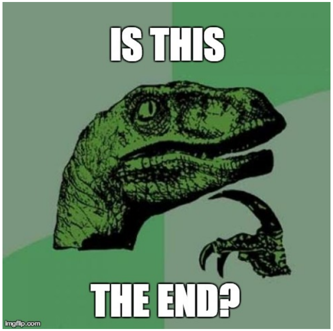
The Anaconda plugin enhances the editor, and adds "lint" and style functions.

Q & A time.

Q: Do you know what the death rate around here is?
A: One per person.

What have we covered?

- A preview of today's activities
- An overview of the sublime IDE



Next: Hands on analysing tweets with Python.

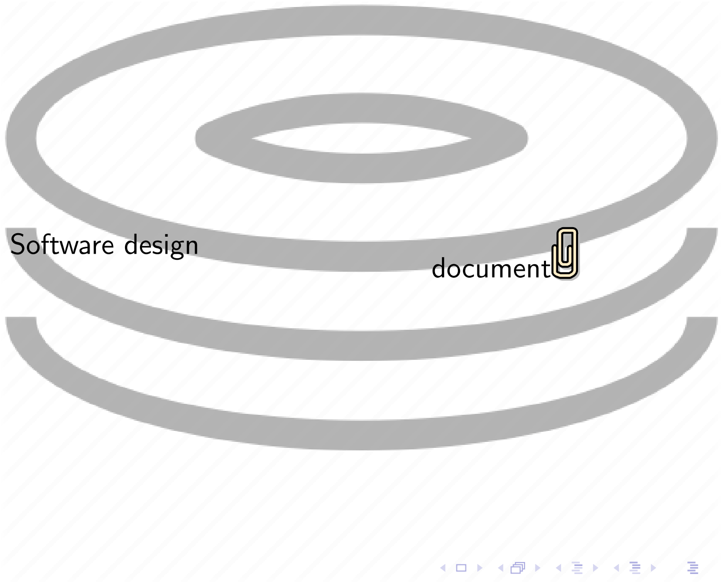
References I

[1] Bing Liu, Sentiment analysis and opinion mining, 2012.

[2] Bo Pang and Lillian Lee, Opinion mining and sentiment analysis, 2008.

[3] Wikipedia Staff, Sentiment analysis, `https://en.wikipedia.org/wiki/Sentiment_analysis`, 2016.

Files of interest



1. Software design document