Project 2

March 22, 2012

Next Few Weeks

Sun	Mon	Tues	Wed	Thurs	Fri	Sat
3/18	3/19	3/20	3/21	3/22	3/23	3/24
		HW2-4 Due HW2-5 Out		**Project 2 Out**		
3/25	3/26	3/27	3/28	3/29	3/30	3/31
Spring Break! Wooooooo!						
4/1	4/2	4/3	4/4	4/5	4/6	4/7
		HW2-5 Due		**Proposal Due**		
4/8	4/9	4/10	4/11	4/12	4/13	4/14
				Project 2 Due		

Today: Project 2

- Data Sources
  - Project Gutenberg
  - English Dictionary
  - Twitter Data
  - Debate Transcripts
- Project 2 Description
- Example Project 2 Proposal

http://www.gutenberg.org/

Project Gutenberg

Project Gutenberg

search book catalog

- Book Search
- Catalog
- Bookshelves

search site

- Main Page
- Categories
- News
- Contact Info

donate

Project Gutenberg needs your donation!

Free eBooks by Project Gutenberg

From Project Gutenberg, the first producer of free ebooks.

Mobile Site · Book search · Bookshelves by topic · Top downloads · Recently added · Report errors

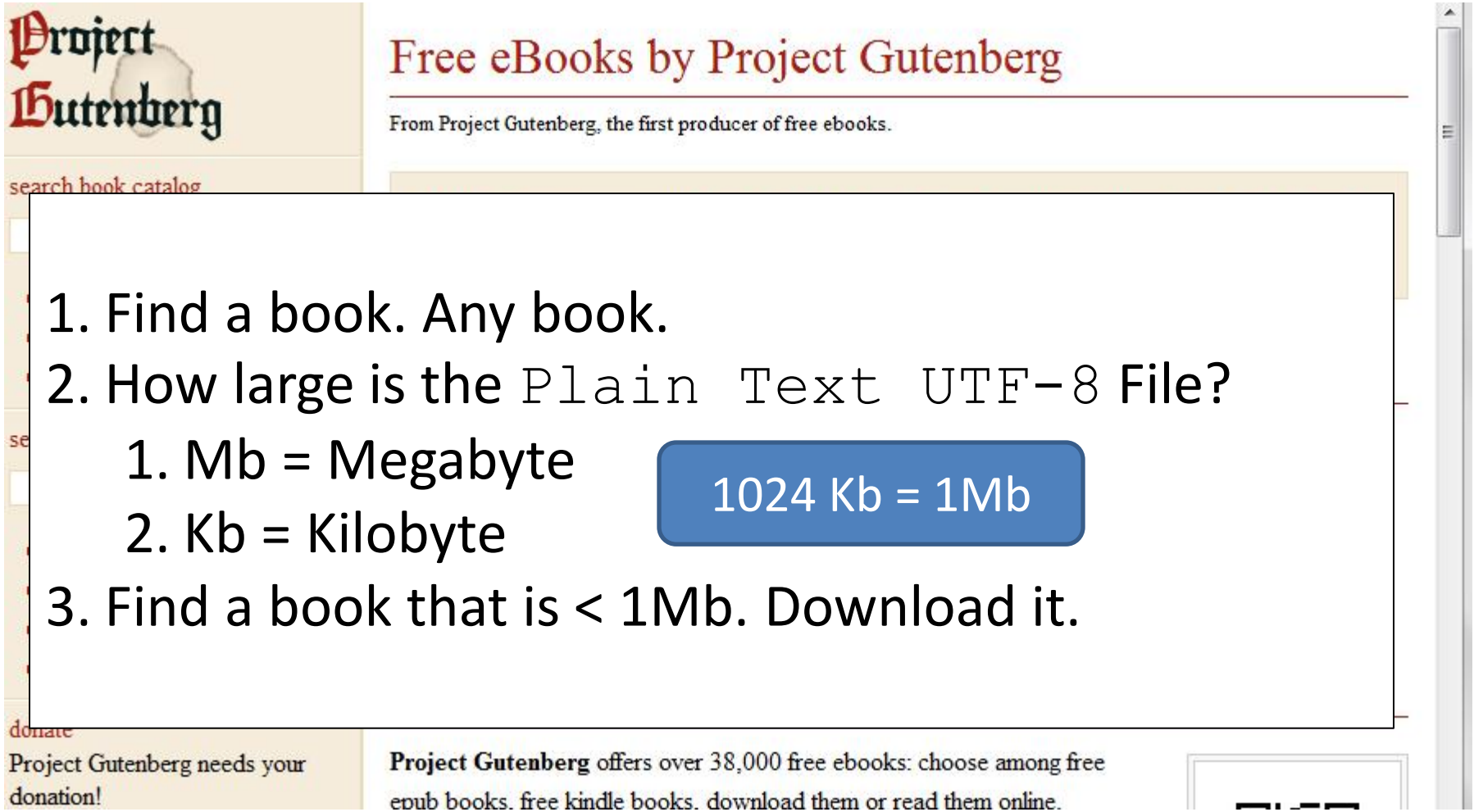
Some of Our Latest Books

Welcome

Project Gutenberg offers over 38,000 free ebooks: choose among free epub books, free kindle books, download them or read them online.

http://www.gutenberg.org/

Project Gutenberg



1. Find a book. Any book.
2. How large is the `Plain Text UTF-8` File?
    1. Mb = Megabyte
    2. Kb = Kilobyte
3. Find a book that is < 1Mb. Download it.

1024 Kb = 1Mb

http://www.gutenberg.org/

Project Gutenberg

Look at the function
`removeLicenseFromProjectGutenberg`
in `DataImport.py`

Today: Project 2

- Data Sources
  - Project Gutenberg
  - English Dictionary
  - Twitter Data
  - Debate Transcripts
- Project 2 Description
- Example Project 2 Proposal

http://www.mso.anu.edu.au/~ralph/OPTED/

Webster's Unabridged Dictionary

The Online Plain Text English Dictionary

OPTED is a public domain English word list dictionary, based on the public domain portion of "The Project Gutenberg Etext of Webster's

1. According to the homepage, what does each line contain?
2. What letter is the **smallest** file?
    1. Mb = Megabyte
    2. Kb = Kilobyte
3. Click on it. Right-click and select `View Page Source...`

1024 Kb = 1Mb

Second computer generated version.

A(1.1M) | B(1005k) | C(1.6M) | D(1M) | E(809k) | F(784k) | G(564k) | H(686k) | I(833k) | J(172k) | K(172k) | L(637k) | M(931k) | N(343k) | O(466k) | P(1.5M) | Q(147k) | R(931k) | S(2.1M) | T(1005k) | U(343k) | V(343k) | W(490k) | X(49k) | Y(74k) | Z(74k)

http://www.mso.anu.edu.au/~ralph/OPTED/

Webster's Unabridged Dictionary

The Online Plain Text English Dictionary

OPTED is a public domain English word list dictionary, based on the public domain portion of "The Project Gutenberg Etext of Webster's Unabridged Dictionary" which is in turn based on the 1913 US Webster's Unabridged Dictionary. (See Project Gutenburg)

This version has been extensively stripped down and set out as one definition per line. All the Gutenburg EText tags and formatting have been removed by computer. Version 0.03 is a new processing of v0.47 of the websters dictionary and it has considerably fewer errors. Also the definition limit of 255 chars has been removed to give full justice of some of the more majestic of the originals. Some important errors in the parts-of-speech fields have been corrected and a lot of inflections/ alternatives and plurals that were missed due to software bugs in v0.01 and 0.02 are now included properly.

Look at the function
`getWebsterDictionary`
in `DataImport.py`

and process them in some way on their own machine. The only usage conditions are that if the material is redistributed, the content (not the formatting) remain in the public domain (ie free) and that the content be easily accessible in non-encoded plain text format at no cost to the end user. The origin of the content should also be acknowledged, including OPTED, Project Gutenburg and the 1913 edition of Webster's Unabridged Dictionary. If the material is to be included in commercial products, Project Gutenburg should be contacted first. There are no restrictions for personal or research uses of this material.

OPTED v0.03 by Letter(size)

Second computer generated version:

A(1.1M) | B(1005k) | C(1.6M) | D(1M) | E(809k) | F(784k) | G(564k) | H(686k) | I(833k) | J(172k) | K(172k) | L(637k) | M(931k) | N(343k) | O(466k) | P(1.5M) | Q(147k) | R(931k) | S(2.1M) | T(1005k) | U(343k) | V(343k) | W(490k) | X(49k) | Y(74k) | Z(74k)

Today: Project 2

- Data Sources
  - Project Gutenberg
  - English Dictionary
  - Twitter Data
  - Debate Transcripts
- Project 2 Description
- Example Project 2 Proposal

Twitter Data

- Collecting tweets **every day** since Jan 23, 2011
  - A little over 1 million tweets a day.

Twitter Data

- Collecting tweets **every day** since Jan 23, 2011
  - A little over 1 million tweets a day.
    - Each tweet should be from the US
    - Each tweet should have geographical data
      - Most tweets are from phones

Twitter Data

- Collecting tweets **every day** since Jan 23, 2011
  - A little over 1 million tweets a day.
    - Each tweet should be from the US
    - Each tweet should have geographical data
      - Most tweets are from phones
  - Currently over 107 million tweets
  - Over 25 Gigabytes (Gb)

1024x1024 Kb = 1024 Mb = 1 Gb

Twitter Data

- `Tweets_With_Elect_Mar20.txt`
  - Tuesday tweets that contain the regex

    `'[eE][lL][eE][cC][tT]'`
  - There are 1,043 of them

Twitter Data

- `Tweets_With_Elect_Mar20.txt`
  - Tuesday tweets that contain the regex

    `'[eE][lL][eE][cC][tT]'`
  - There are 1,043 of them

1. Open the file and look at it.
2. Column information is listed in `TextDataSrcs.pdf`

Twitter Data

1, 934, ‘‘Twitter for iPhone’’, ‘‘Bart Simpson’’, 32, 98453, 87.45984, ‘‘United States’’, ‘‘’’, 0, false, false, ‘‘Don’t have a cow, man!’’

Twitter Data

1, 934, ‘‘Twitter for iPhone’’, ‘‘Bart Simpson’’, 32, 98453, 87.45984, ‘‘United States’’, ‘‘’’, 0, false, false, ‘‘Don’t have a cow, man!’’

Look at the function
`twitterExample`
in `DataImport.py`

Twitter Data

1, 934, ‘‘Twitter for iPhone’’, ‘‘Bart Simpson’’, 32, 98453, 87.45984, ‘‘United States’’, ‘‘’’, 0, false, false, ‘‘Don’t have a cow, man!’’

Look at the function
`twitterExample`
in `DataImport.py`

Question: Are all tweets about the election?

Getting Twitter Data

- Currently over 107 million tweets
  - **BUT** you want to get tweets that *match* a regular expression

Getting Twitter Data

- Currently over 107 million tweets
  - **BUT** you want to get tweets that *match* a regular expression
  - **Professor Reiss** will help us here.

Look at `TwitterForm.txt`

Today: Project 2

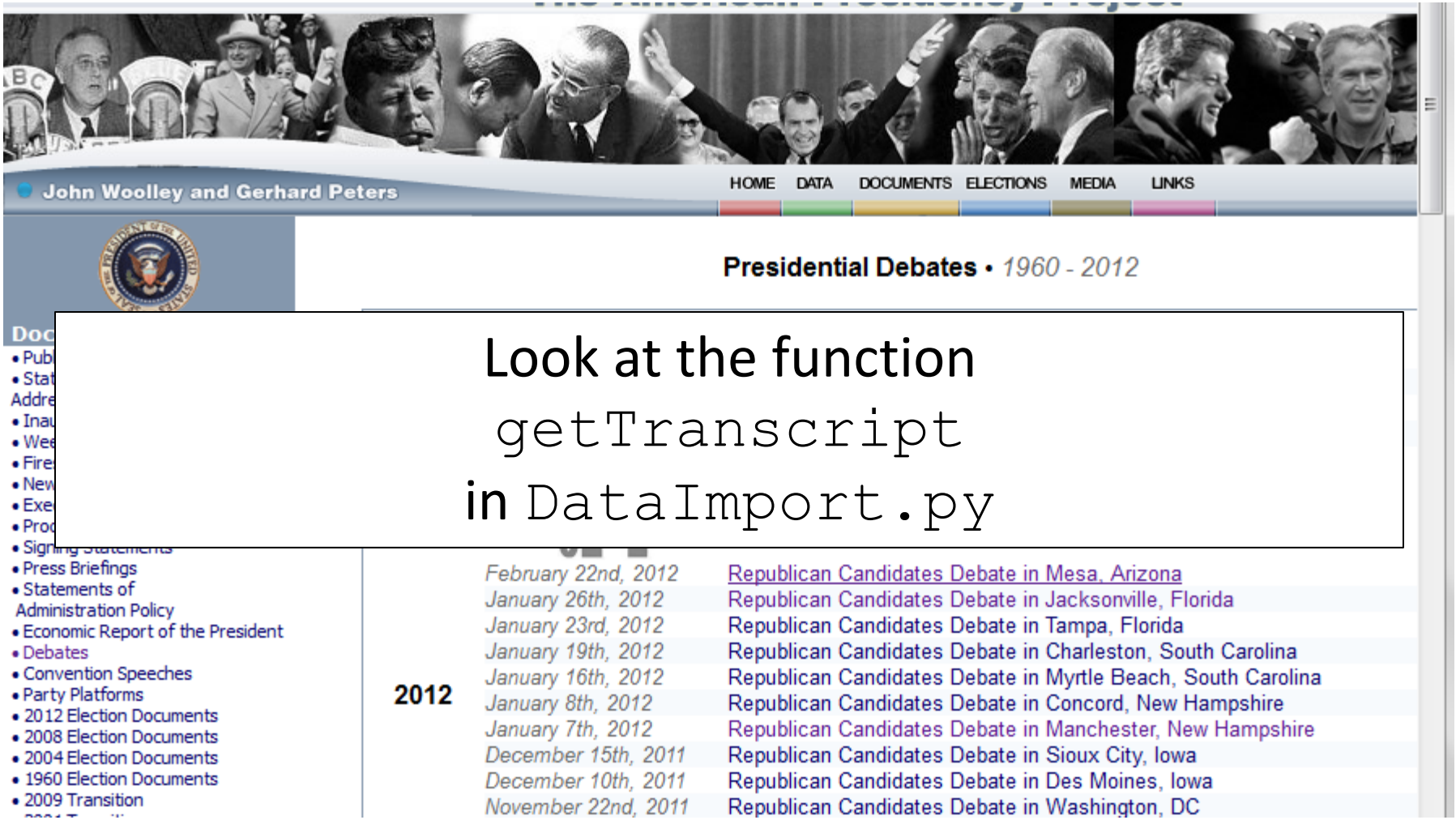
- Data Sources
  - Project Gutenberg
  - English Dictionary
  - Twitter Data
  - Debate Transcripts
- Project 2 Description
- Example Project 2 Proposal

http://www.presidency.ucsb.edu/

The American Presidency Project

http://www.presidency.ucsb.edu/

The American Presidency Project



Look at the function
`getTranscript`
in `DataImport.py`

Today: Project 2

- Data Sources
  - Project Gutenberg
  - English Dictionary
  - Twitter Data
  - Debate Transcripts
- Project 2 Description
- Example Project 2 Proposal

Project 2 Description

(Independent) Pose a computational question based on textual data. You may use your own data source, or choose from the data sources we discussed in class:

1. **Project Gutenberg:** `http://www.gutenberg.org/`
2. **Dictionary:** `http://www.mso.anu.edu.au/~ralph/OPTED/`
3. **Twitter Data:** Email Professor Reiss using the form on the course website with a regular expression to get all tweets that contain the regular expression you specify.
4. **American Presidency Project Debates:** `http://www.presidency.ucsb.edu/debates.php`

For your project you must present a testable hypothesis, carry out the required analyses, report your findings in a clear and understandable way, and discuss your results.

To present your results, you may report descriptive summary statistics (such as count, mean, median, standard deviation, etc.). You will also be able to import your results into Excel to analyze basic trends (via color formatting and plotting). However, you will **not** be graded on any Excel work beyond presenting your results in a clear manner.

Project 2 Description

(Independent) Pose a computational question based on textual data. You may use your own data source, or choose from the data sources we discussed in class:

1. **Project Gutenberg:** `http://www.gutenberg.org/`
2. **Dictionary:** `http://www.mso.anu.edu.au/~ralph/OPTED/`
3. **Twitter Data:** Email Professor Reiss using the form on the course website with a regular expression to get all tweets that contain the regular expression you specify.
4. **American Presidency Project Debates:** `http://www.presidency.ucsb.edu/debates.php`

For your project you must present a testable hypothesis, carry out the required analyses, report your findings in a clear and understandable way, and discuss your results.

To present your results, you may report descriptive summary statistics (such as count, mean, median, standard deviation, etc.). You will also be able to import your results into Excel to analyze basic trends (via color formatting and plotting). However, you will **not** be graded on any Excel work beyond presenting your results in a clear manner.

Proposal (1/2): Project Description

Project Description (`YourName_Proj2_Proposal.txt`)

Write a concise (1 page) description of the project you would like to execute. This description should include the following parts:

1. **Background:** put your project idea in context.
2. **Claim:** the specific hypothesis you plan to test (which is a statement, *not* a question).
3. **Data:** a one-sentence description of your data source.
4. **Programming Elements:** a few sentences describing the problems you will need to write Python functions for.
5. **Potential Roadblocks:** a list of potential obstacles.
6. **Project Modifications:** address the following two scenarios:
   - **Backup Plan:** suppose your project is much harder than you anticipate. What parts of the project would you change to still get somewhat interesting results?
   - **Increasing Degree of Difficulty:** suppose your project is much easier than you anticipate. What ways would you extend the project?

Proposal (1/2): Project Description

Project Description (`YourName_Proj2_Proposal.txt`)

Write a concise (1 page) description of the project you would like to execute. This description should include the following parts:

1. **Background:** put your project idea in context.
2. **Claim:** the specific hypothesis you plan to test (which is a statement, *not* a question).
3. **Data:** a one-sentence description of your data source.
4. **Programming Elements:** a few sentences describing the problems you will need to write Python functions for.
5. **Potential Roadblocks:** a list of potential obstacles.
6. **Project Modifications:** address the following two scenarios:
   - **Backup Plan:** suppose your project is much harder than you anticipate. What parts of the project would you change to still get somewhat interesting results?
   - **Increasing Degree of Difficulty:** suppose your project is much easier than you anticipate. What ways would you extend the project?

Proposal (2/2): Skeleton Code

Skeleton Code (`YourName_Proj2_Proposal.py`)

Write a Python file that contains an outline of the code you anticipate writing. This file should compile! It should include the following:

1. **Comments** at the top of the file describing what the program does.
2. **Functions** that you will write (of course, you might change this later).
3. **Function descriptions** (in triple quotes) that describe (1) what the function does, (2) what the inputs are, and (3) what the outputs are.
4. **Some lines of code and comments** that help describe what the functions will ontain.

Don't get too wrapped up in the details here - the goal of the Skeleton Code is to provide you with an outline of what you have to program.

Proposal (2/2): Skeleton Code

Skeleton Code (`YourName_Proj2_Proposal.py`)

Write a Python file that contains an outline of the code you anticipate writing. This file should compile! It should include the following:

1. **Comments** at the top of the file describing what the program does.
2. **Functions** that you will write (of course, you might change this later).
3. **Function descriptions** (in triple quotes) that describe (1) what the function does, (2) what the inputs are, and (3) what the outputs are.
4. **Some lines of code and comments** that help describe what the functions will ontain.

Don't get too wrapped up in the details here - the goal of the Skeleton Code is to provide you with an outline of what you have to program.

Project 2 Rubric

Category	# Points Earned
Proposal	15
Degree of Difficulty	15
Execution	25
Code Quality	15
Website	15
Analysis	15
TOTAL	100

Today: Project 2

- Data Sources
  - Project Gutenberg
  - English Dictionary
  - Twitter Data
  - Debate Transcripts
- Project 2 Description
- Example Project 2 Proposal

Anna Ritz
Project 2 Proposal

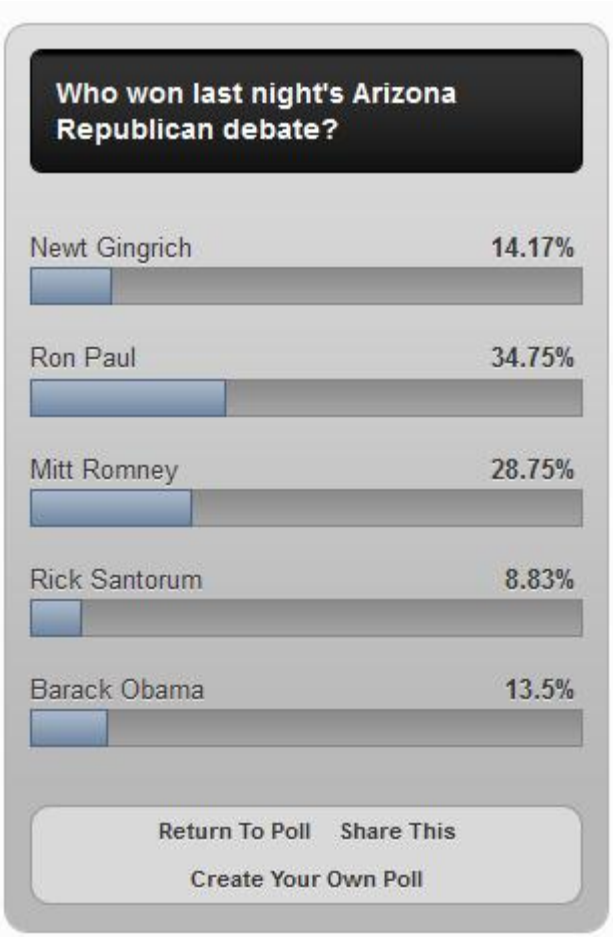
Background: After each debate, there's lots of talk about who "won" it, i.e.

`http://www.washingtonpost.com/blogs/the-fix/post/arizona-republican-debate-winners-and-losers/2012/02/22/gIQAsKkVUR_blog.html`

I will define the "winner" as the person who received applause the most frequently during the debate.

Claim: I claim that in the AZ debate, Romney "won" and Santorum "lost" – that is, Romney received applause the most and Santorum received applause the least.

....



http://blogs.phoenixnewtimes.com/valleyfever/2012/02/who_won_last_nights_arizona_re.php

Skeleton Code

Skeleton Code

```
# Anna Ritz
# Project 2
# Skeleton Code

## This program assesses how "popular" each republican candidate is
## by counting the number of [applause] tags in a
## Republican debate.

# CONSTANT VARIABLES (will NEVER change values) are in ALL CAPS
# If you put these variables OUTSIDE all functions, then you
# can access them in ANY function.
CANDIDATES = ['GINGRICH','PAUL','ROMNEY','SANTORUM']
DEBATE_FILE = 'AZDebate.txt'

def assessPopularity():
    '''Assesses the popularity of the candidates in the AZ debate.
    INPUTS: none
    OUTPUTS: none'''

    # Step 1: Read the debate file
    myString = readFile()

    # Step 2: For each candidate, assess popularity
    for cand in CANDIDATES:
        countApplause(cand,myString)

    return

def readFile():
    '''Reads DEBATE_FILE and returns a string.
    INPUTS: none
    OUTPUTS: String of the debate'''

    return '' # returns an empty string for now.

def countApplause(candidate,debateString):
    '''Assesses the popularity of the candidate in the debateString.
    INPUTS: candidate (String) - name of candidate
```

Anna Ritz
Project 2 Proposal

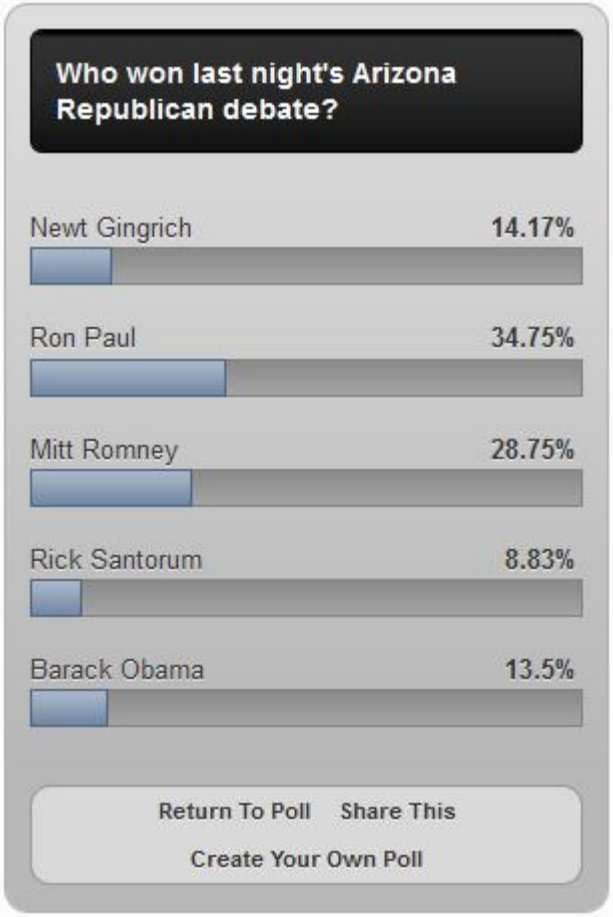
...

Claim: I claim that in the AZ debate, Romney "won" and Santorum "lost" – that is, Romney received applause the most and Santorum received applause the least.

....

Backup Plan: ???

Increasing Degree of Difficulty: ???



http://blogs.phoenixnewtimes.com/valleyfever/2012/02/who_won_last_nights_arizona_re.php

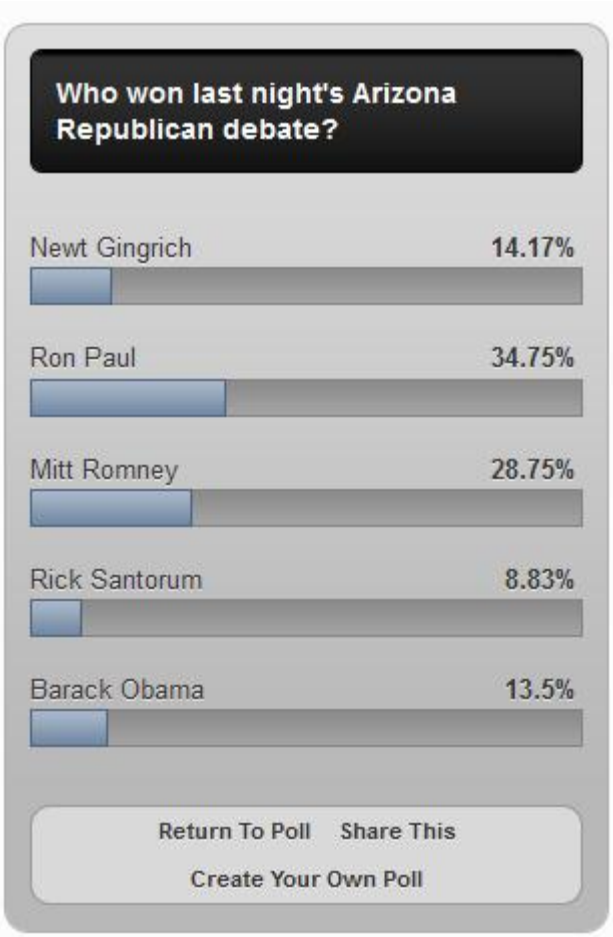
Anna Ritz
Project 2 Proposal

...

Claim: I claim that in the AZ debate, Romney "won" and Santorum "lost" – that is, Romney received applause the most and Santorum received applause the least.

....

Backup Plan: ???



http://blogs.phoenixnewtimes.com/valleyfever/2012/02/who_won_last_nights_arizona_re.php

Today: Project 2

- Data Sources
  - Project Gutenberg
  - English Dictionary
  - Twitter Data
  - Debate Transcripts
- Project 2 Description
- Example Project 2 Proposal

Next Few Weeks

Sun	Mon	Tues	Wed	Thurs	Fri	Sat
3/18	3/19	3/20	3/21	3/22	3/23	3/24
		HW2-4 Due HW2-5 Out		**Project 2 Out**		
3/25	3/26	3/27	3/28	3/29	3/30	3/31
Spring Break! Wooooooo!						
4/1	4/2	4/3	4/4	4/5	4/6	4/7
		HW2-5 Due		**Proposal Due**		
4/8	4/9	4/10	4/11	4/12	4/13	4/14
				Project 2 Due		