Explained Python text co-occurance calculator.

This is an explained example of using python to find word co-occurrences in text (example uses Jane Austen novels). For example, you can find that the word "pride" co-occurs with "darcy" more than with "elizabeth".

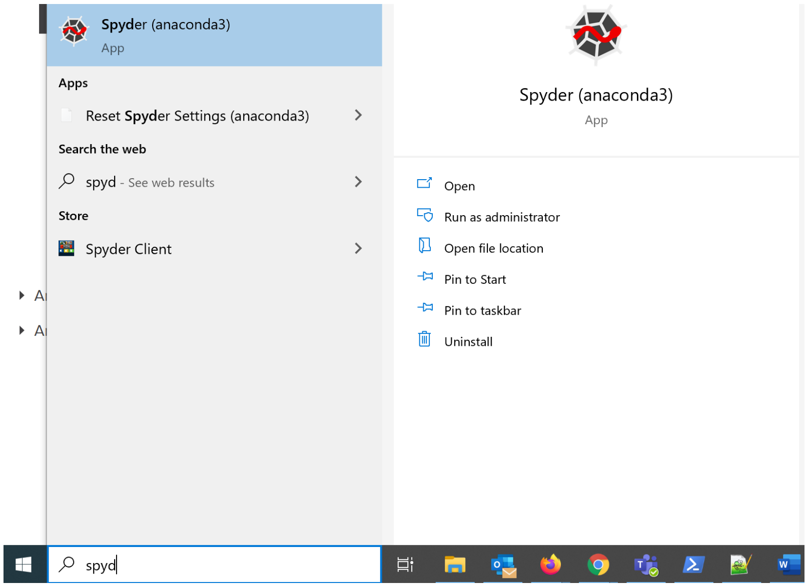
Instructions

1. Download the files in this repository into a new directory (unzipping it if needed). Do this by using the green button with the down arrow that says "Code".

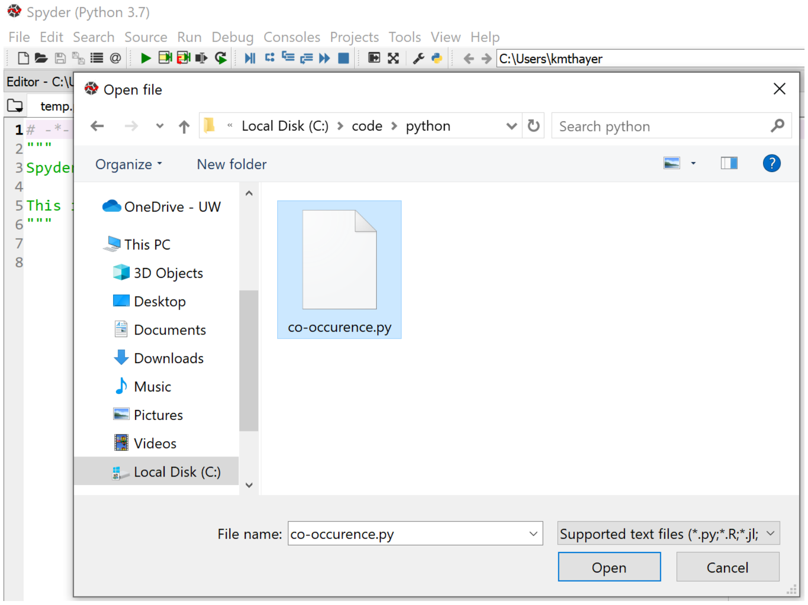
2. Download and install the Anaconda Distribution, which comes with Python and a code editor (called Spyder). Or find or use some other python code editor:

https://www.anaconda.com/distribution/

3. Open the Spyder code editor (screenshot shows opening Spyder on Windows



4. In Spyder, choose the "file" menu, and select "open." Then open the "co-occurence.py" in the new directory you made from the zip file

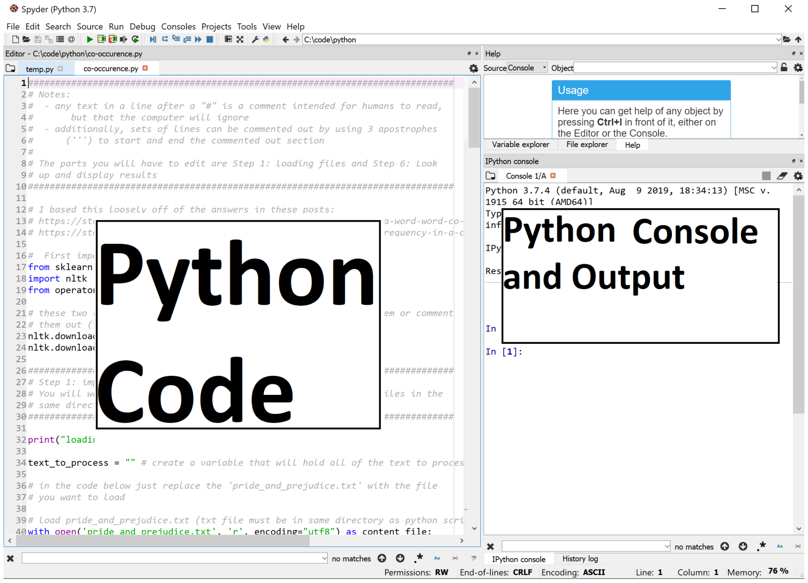


```
# -*-
"""
Spyde

This
"""
```

File name: co-occurence.py

Supported text files (*.py;*.R;*.jl;

5. Now you should see the file open in Spyder like below (though without the giant labels I stuck on the screenshot). There is an area where the code file is on the left, and an area on the right called the "console" or "terminal" where the results of the code will appear, and where you can run additional commands if you want.

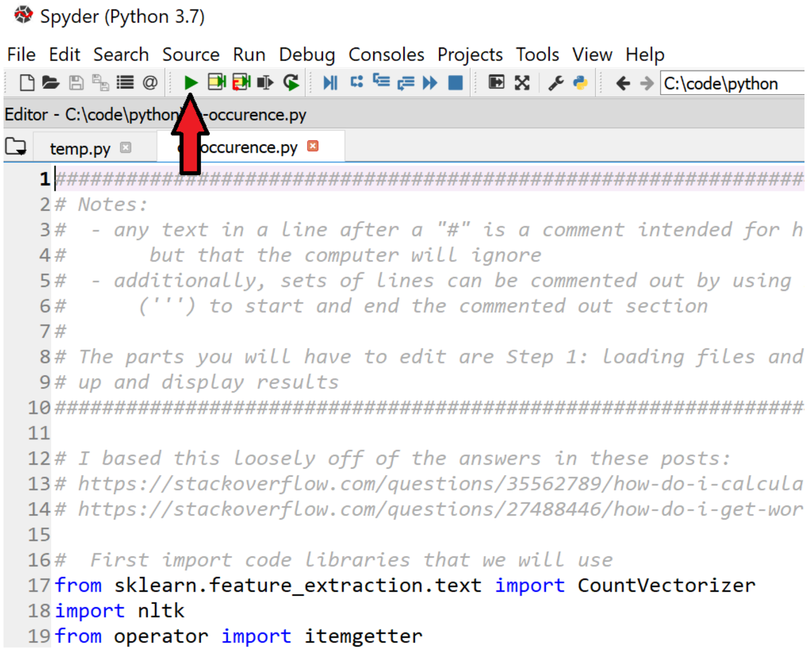


Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

```
##################################################################################
# Notes:
#  - any text in a line after a "#" is a comment intended for humans to read,
#        but that the computer will ignore
#  - additionally, sets of lines can be commented out by using 3 apostrophes
#       (''') to start and end the commented out section
#
# The parts you will have to edit are Step 1: loading files and Step 6: Look
# up and display results
##################################################################################

# I based this loosely off of the answers in these posts:
# https://st                                                  a-word-word-co-
# https://st                                                  requency-in-a-c

#  First imp
from sklearn
import nltk
from operato

# these two                                                   em or comment
# them out (
nltk.downloa
nltk.downloa

############                                                  #############
# Step 1: im
# You will w                                                  iles in the
# same direc
############                                                  #############

print("loadi

text_to_process = "" # create a variable that will hold all of the text to proces

# in the code below just replace the 'pride_and_prejudice.txt' with the file
# you want to load

# load pride_and_prejudice.txt (txt file must be in same directory as python scri
with open('pride_and_prejudice.txt', 'r', encoding="utf8") as content_file:
```

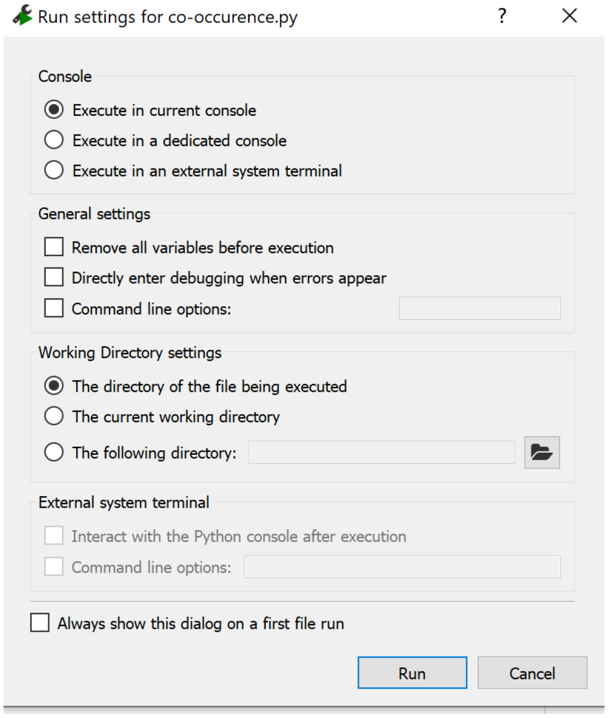
```
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.
1915 64 bit (AMD64)]
Typ
inf

IPy

Res


In

In [1]:
```

6. To run the Python file, click on the little green triangle at the top left.


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\code\python
Editor - C:\code\python\co-occurence.py
temp.py
co-occurence.py

```
##################################################################
# Notes:
#  - any text in a line after a "#" is a comment intended for h
#       but that the computer will ignore
#  - additionally, sets of lines can be commented out by using
#      (''') to start and end the commented out section
#
# The parts you will have to edit are Step 1: loading files and
# up and display results
##################################################################

# I based this loosely off of the answers in these posts:
# https://stackoverflow.com/questions/35562789/how-do-i-calcula
# https://stackoverflow.com/questions/27488446/how-do-i-get-wor

#  First import code libraries that we will use
from sklearn.feature_extraction.text import CountVectorizer
import nltk
from operator import itemgetter
```

7. The first time you run a program, it will ask you some options. The defaults should be fine, but double check that under "Working Directory settings" it has "The directory of the file being executed" selected.



Run settings for co-occurence.py

Console

● Execute in current console
○ Execute in a dedicated console
○ Execute in an external system terminal

General settings

□ Remove all variables before execution
□ Directly enter debugging when errors appear
□ Command line options:

Working Directory settings

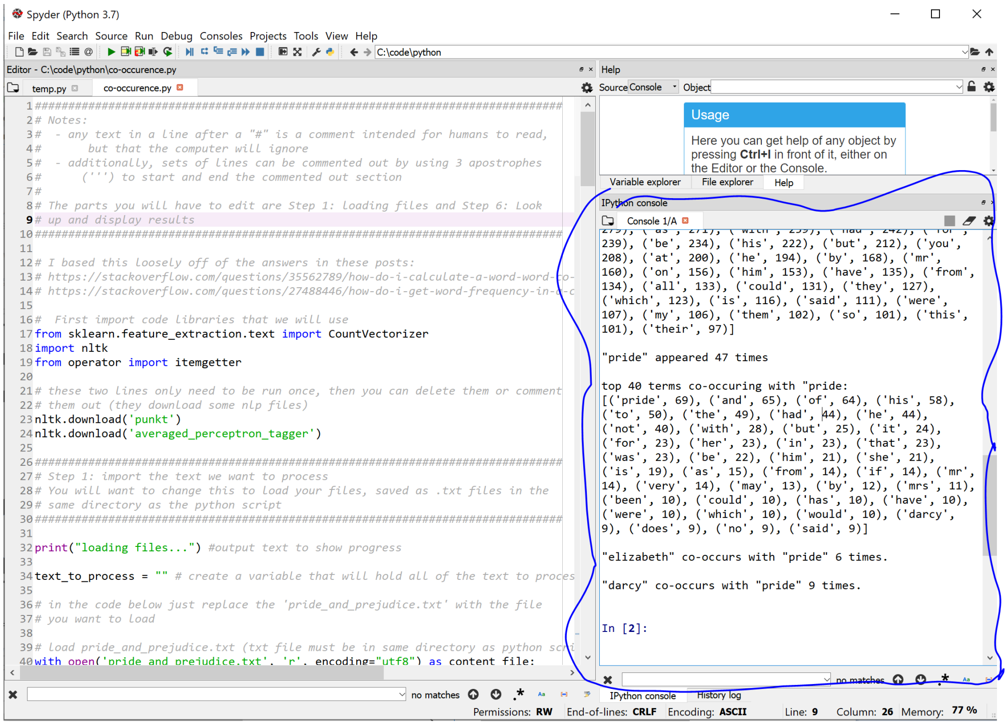
● The directory of the file being executed
○ The current working directory
○ The following directory:

External system terminal

□ Interact with the Python console after execution
□ Command line options:

□ Always show this dialog on a first file run

8. After it runs, you should see the output of the script in the console panel on the right (you can scroll up and down). Note that for the top 40 terms co-occuring with pride, "pride" co-occurs with itself 69 times, with "and" 65 times, with "of" 64 times, etc.



```python
#######################################################################
# Notes:
#  - any text in a line after a "#" is a comment intended for humans to read,
#        but that the computer will ignore
#  - additionally, sets of lines can be commented out by using 3 apostrophes
#       (''') to start and end the commented out section
#
# The parts you will have to edit are Step 1: loading files and Step 6: Look
# up and display results
#######################################################################

# I based this loosely off of the answers in these posts:
# https://stackoverflow.com/questions/35562789/how-do-i-calculate-a-word-word-co-
# https://stackoverflow.com/questions/27488446/how-do-i-get-word-frequency-in-a-c

#  First import code libraries that we will use
from sklearn.feature_extraction.text import CountVectorizer
import nltk
from operator import itemgetter

# these two lines only need to be run once, then you can delete them or comment
# them out (they download some nlp files)
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

#######################################################################
# Step 1: import the text we want to process
# You will want to change this to load your files, saved as .txt files in the
# same directory as the python script
#######################################################################

print("loading files...") #output text to show progress

text_to_process = "" # create a variable that will hold all of the text to proces

# in the code below just replace the 'pride_and_prejudice.txt' with the file
# you want to load

# load pride_and_prejudice.txt (txt file must be in same directory as python scri
with open('pride_and_prejudice.txt', 'r', encoding="utf8") as content_file:
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

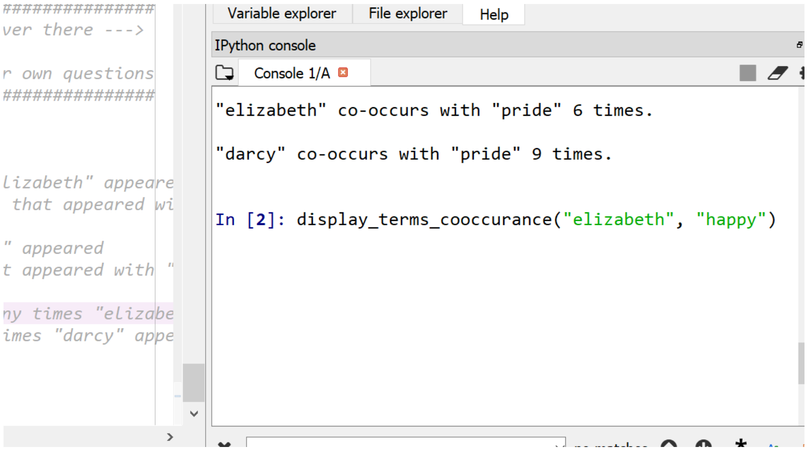
```
239), ('be', 234), ('his', 222), ('but', 212), ('you',
208), ('at', 200), ('he', 194), ('by', 168), ('mr',
160), ('on', 156), ('him', 153), ('have', 135), ('from',
134), ('all', 133), ('could', 131), ('they', 127),
('which', 123), ('is', 116), ('said', 111), ('were',
107), ('my', 106), ('them', 102), ('so', 101), ('this',
101), ('their', 97)]

"pride" appeared 47 times

top 40 terms co-occuring with "pride:
[('pride', 69), ('and', 65), ('of', 64), ('his', 58),
('to', 50), ('the', 49), ('had', 44), ('he', 44),
('not', 40), ('with', 28), ('but', 25), ('it', 24),
('for', 23), ('her', 23), ('in', 23), ('that', 23),
('was', 23), ('be', 22), ('him', 21), ('she', 21),
('is', 19), ('as', 15), ('from', 14), ('if', 14), ('mr',
14), ('very', 14), ('may', 13), ('by', 12), ('mrs', 11),
('been', 10), ('could', 10), ('has', 10), ('have', 10),
('were', 10), ('which', 10), ('would', 10), ('darcy',
9), ('does', 9), ('no', 9), ('said', 9)]

"elizabeth" co-occurs with "pride" 6 times.

"darcy" co-occurs with "pride" 9 times.


In [2]:
```

9. Let me explain the file a little bit:

- Any text after a "#" in a line is a comment, meant for humans to read, and Python will ignore them. Additionally three apostrophes (''') can mark the start and end of a set of lines to be comments that Python will ignore
- After loading some code libraries at the top, I divided the code into six sections:
  - Step 1: import the text we want to process
  - Step 2: split words into sentences here
  - Step 3: clean up data (optional and currently disabled by commenting out all the lines)
  - Step 4: find the co-occurance matrix for all words in the text
  - Step 5: define custom functions to help retrieve and display information
  - Step 6: Look up and display results in the console
- The only sections you will be required to modify are Step 1 and Step 6.
  - Step 1 modifications: You will need to get whatever text you want to process, saved as txt files in the same directory as the python file. Then you will modify the line of code where it has "pride_and_prejudice.txt" and replace that with the name of your file. If you want to load multiple txt files, there is an example commented out for how to load emma.txt and add that one as well.
  - Step 2 modifications: I have several function calls to answer specific questions about words in Pride and Prejudice. You will want to modify and copy these lines of code to ask about whatever particular words you want to ask about, and you can change how many words to return in places where I asked for the top 40 words.
- There are several places where I have added optional code changes you can make:
  - Step 3: You can delete the three apostraphes at the start and end to enable that code cleaning step
  - Step 4: You can uncomment what is currently line 142 to only count each word once per sentence even if it occurs multiple times in a sentence.

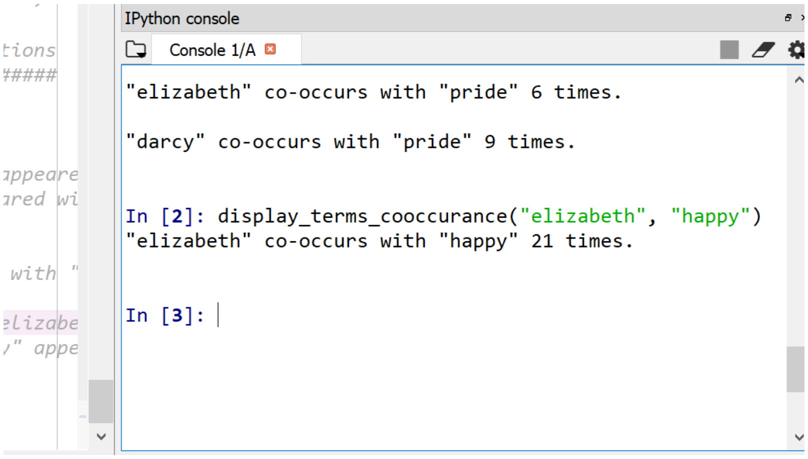
  - Step 5: In the definition of the get_cooccurances_sorted function, you can use an alternate line to get proportional co-occurance based on a term (number of co-occurances divided by number of times the other term appears). This finds more unique words that co-occur with what you are looking for.
- I left comments through most of the code, trying to explain what it does, and sometimes how, so you can look through it and try to understand and modify it yourself if you want to.
- If and when you try modifying things, back it up! It is easy to break a program and not know how to undo it. The easiest way I know to back things up is to just email myself the file as I work on it. Then if you ever get lost, you can download one of those working versions and start again from there.

10. If you want to run additional commands in the console, say like asking more questions about occurances and co-occurances, you can type (or paste) them directly into the console, like this:



```
"elizabeth" co-occurs with "pride" 6 times.

"darcy" co-occurs with "pride" 9 times.


In [2]: display_terms_cooccurance("elizabeth", "happy")
```

11. Then press enter, and the results will display there (see below).



```
"elizabeth" co-occurs with "pride" 6 times.

"darcy" co-occurs with "pride" 9 times.


In [2]: display_terms_cooccurance("elizabeth", "happy")
"elizabeth" co-occurs with "happy" 21 times.


In [3]:
```

12. Note on the console vs. the code file. If you want to just look at values and try things out, that can be good to do on the console, but it is easy to lose which commands you ran. If you want to have lines of code that you remember and can run again, put those in the python file so they run every time you run the file.