Textual Analysis & Introduction to Python

Feb. 21, 2013

Before we leave Excel...

- Pick and choose ways to increase Degree of Difficulty.
- Display your data in different ways (graph, table, etc.)
- Remember to answer your claim
 - My hypothesis is correct/incorrect because ...
 - X% of the time, my hypothesis was correct...
- Be sure to explain what obstacles you encountered
- Look at the rubric!







Alphabetical index of all words in a text

Word	Page Numbers
Apple	4,7,10,27
Banana	77,110,130
Carrot	50,101
Date	9

- Before computers, was a *huge* pain.
- What texts might have had concordances?

- Before computers, was a *huge* pain.
- What texts might have had concordances?
 - The Bible
 - The Quran
 - The Vedas
 - Shakespeare

- Before computers, was a *huge* pain.
- What texts might have had concordances?
 - The Bible
 - The Quran
 - The Vedas
 - Shakespeare

Not a "New" Problem: First Bible Concordance completed in 1230

How long would the King James Bible take us?

- 783,137 words

http://agards-bible-timeline.com/q10_bible-facts.html

How long would the King James Bible take us?
 – 783,137 words

800,000 * (3 min. to look up word and put page #) = 2,400,000 minutes = 40,000 hours = 1,667 days = 4.5 years

http://agards-bible-timeline.com/q10_bible-facts.html

How long would the King James Bible take us?
 – 783,137 words

800,000 * (3 min. to look up word and put page #) = 2,400,000 minutes = 40,000 hours = 1,667 days = 4.5 years

Takes 70 hours to read the King James Bible aloud

http://agards-bible-timeline.com/q10_bible-facts.html

Strong's Concordance

- Concordance of the King James Bible
- Published in 1890 by James Strong

ANT

Prv 6: 6 Go to the a, thou sluggard; consider her H5244

ANTICHRIST

- 1Jn2:18 as ye have heard that a shall come, even
22 is a, that denieth the Father and the Son.
4: 3 this is that *spirit* of a, whereof ye haveG500
G500
- 2Jn 7 in the flesh. This is a deceiver and an a. G500

ANTICHRISTS

1Jn 2:18 are there many a; whereby we know that G500

ANTIOCH

- Act 6: 5 Parmenas, and Nicolas a proselyte of A: G491 11:19 and Cyprus, and A, preaching the word G490 20 they were come to A, spake unto the G490 22 Barnabas, that he should go as far as A. G490 26 brought him unto A. And it came to pass, G490 26 disciples were called Christians first in A. G490 27 came prophets from Jerusalem unto A. G490
 - 13: 1 church that was at A certain prophets 14 they came to A in Pisidia, and went
 - 14:19 certain Jews from A and Iconium, who 21 again to Lystra, and to Iconium, and A, 26 And thence sailed to A, from whence
 - 20 And mence saled to A, from whence
 - 15:22 their own company to A with Paul and

Jas	1:21 Wherefore lay a all filthiness and	G659
API	ELLES	
Ro	16:10 Salute A approved in Christ. Salute them	G559
API	es	
1Ki	10:22 and silver, ivory, and a, and peacocks.	H6971
2Ch	9:21 and silver, ivory, and a, and peacocks.	H6971
API	HARSACHITES	
Ezr	5: 6 companions the A, which were on this	H671
	6: 6 companions the A, which are beyond the	H671
API	HARSATHCHITES	

Ezr 4: 9 the Dinaites, the A, the Tarpelites, the H671

APHARSITES

Ezr 4: 9 the Tarpelites, the A, the Archevites, the H670

APHEK

G490 Jos 12:18 The king of A, one; the king of Lasharon, H663 13: 4 unto A, to the borders of the Amorites: G490 H663 19:30 Ummah also, and A, and Rehob: twenty G490 H663 1Sa 4: 1 and the Philistines pitched in A. G490 H663 G490 29: 1 all their armies to A: and the Israelites H663 1Ki 20:26 and went up to A, to fight against Israel. H663 G490



Wikipedia

http://www.christianbook.com/reader/?item_no=563788

From Concordance to Word Frequency

Suppose our text has 1000 words total.

Word	Page Numbers	# of Occurrences	Word Frequency
Apple	4,7,10,27	4	4/1000
Banana	77,110,130	3	3/1000
Carrot	50,101	2	2/1000
Date	9	1	1/1000

Google Ngrams

- Click on the "Ngrams" url on the class website
- <u>ngram</u>: a set of *n* words
 - "hello" is a 1-gram
 - "hello there" is a 2-gram

Google Ngrams

- Click on the "Ngrams" url on the class website
- <u>ngram</u>: a set of *n* words
 - "hello" is a 1-gram
 - "hello there" is a 2-gram
- Click on "About Google Books Ngram Viewer" for more information
- Question: what is the data source here?



The Wizard of OZ

• About 40 Books, written by 7 different authors



http://www.ssc.wisc.edu/~zzeng/soc357/OZ.pdf

The Wizard of OZ

• About 40 Books, written by 7 different authors



http://www.ssc.wisc.edu/~zzeng/soc357/OZ.pdf

The Wizard of OZ

• About 40 Books, written by 7 different authors



http://www.ssc.wisc.edu/~zzeng/soc357/OZ.pdf

The Federalist Papers

- 85 articles written in 1787 to promote the ratification of the US Constitution
- In 1944, Douglass Adair guessed authorship
 - Alexander Hamilton (51)
 - James Madison (26)
 - John Jay (5)
 - 3 were a collaboration
- Corroborated in 1964 by a computer analysis



Wikipedia

http://pages.cs.wisc.edu/~gfung/federalist.pdf











Makita Cordless Chain Saw, \$270



Python: A Programming Language Free!



9poundhammer.blogspot.com





Break

- If you're on a laptop: install Python 2.6.6 (url)
- Everyone: Open IDLE (Python GUI)



- **Expressions** are *inputs* that Python evaluates
 - Expressions return an *output*
 - Like using a calculator

- **1.** Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

- **Expressions** are *inputs* that Python evaluates
 - Expressions return an *output*
 - Like using a calculator
- Type the expressions below

after '>>>' and hit Enter



- **1.** Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

• Assignments do not have an output, they are stored in memory.

- 1. Expressions
- **2. Assignments**a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

- Assignments do not have an output, they are stored in memory.
 - We've done this kind of thing in Excel



We have *assigned* the number 1 to cell A1.

- **2.** Assignmentsa) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

- Assignments do not have an output, they are stored in memory.
 - We've done this kind of thing in Excel



We have *assigned* the number 1 to cell A1.



Let's rename cell A1 to x.

- **2.** Assignmentsa) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

- Assignments do not have an output, they are stored in memory.
 - We've done this kind of thing in Excel



Let's rename cell A1 to x.

- . Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

- Assignments do not have an output, they are stored in memory.
 - We've done this kind of thing in Excel



- . Expressions
- Assignments
 a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

- Assignments do not have an output, they are stored in memory.
 - We've done this kind of thing in Excel

– We can now use x in expressions!

>>> x+1	Men	Memory	
2 (x+2)*3	Variable Name	Value	
9	x	1	

- . Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

• You can name your variables

anything

>>> jadrian = 100
>>> myNumber = 12345
>>> noninteger = 4.75

- Well, almost anything
 - No spaces, operators, punctuation, number in the first position
- Variables usually start with a lowercase letter and, if useful, describe something about the value.

- Assignments
 a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

• Try this: >>> 3/2

- 1. Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

- Try this: >>> 3/2
- There are two types of numbers in Python.
 The type() function is useful.

```
>>> type(3/2)
<type 'int'>
>>> type(1.5)
<type 'float'>
```

- 1. Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

There are two types of numbers in Python.
 The type() function is useful.

```
>>> type(3/2)
<type 'int'>
>>> type(1.5)
<type 'float'>
```

• Floats are *decimals*.

>>> 3.0/2.0 1.5

- 1. Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

There are two types of numbers in Python.
 The type() function is useful.

```
>>> type(3/2)
<type 'int'>
>>> type(1.5)
<type 'float'>
```

• Floats are *decimals*.

>>> 3.0/2.0 1.5

General Rule: Expressions for a particular type will *output* that same type!

- 1. Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

• **Strings** are sequences of characters, surrounded by single quotes.

>>> 'hi'

'hi'

- >>> myString = 'hi there'
- >>> myString
- 'hi there'

- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

• **Strings** are sequences of characters, surrounded by single quotes.

>>> 'hi'

'hi'

- >>> myString = 'hi there'
- >>> myString

```
'hi there'
```

• The + operator concatenates

General Rule: Expressions for a particular type will *output* that same type!

- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

• **Strings** are sequences of characters, surrounded by single quotes.

>>> 'hi'

'hi'

```
>>> myString = 'hi there'
```

```
>>> myString
```

```
'hi there'
```

• The + operator concatenates

```
>>> endString = ' class!'
>>> myString + endString
'hi there class!'
>>> newString = myString + endString
>>> newString
'hi there class!'
```

- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

• Lists are an ordered collection of items

```
>>> [5,10,15]
[5, 10, 15]
>>> myList = [5,10,15]
>>> myList
[5, 10, 15]
>>> stringList = ['hi','there','class']
>>> stringList
['hi', 'there', 'class']
```

- 1. Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

• Lists are an ordered collection of items

```
>>> [5,10,15]
[5, 10, 15]
>>> myList = [5,10,15]
>>> myList
[5, 10, 15]
>>> stringList = ['hi','there','class']
>>> stringList
['hi', 'there', 'class']
```

• The + operator concatenates

```
>>> myList + stringList
[5, 10, 15, 'hi', 'there', 'class']
```

- Assignments
 a) Variables
 - a) variable
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

 To get an element from a list, use the expression >>> myList[i] where i is the index.

• List indices start at 0!



- 1. Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

To get a range of elements from a list, use the expression >>> myList[i:j] where i is the start index (inclusive) and j is the end index (exclusive).

```
>>> myList
[5, 4, 15]
>>> myList[0:2]
[5, 4]
>>> myList[1:3]
[4, 15]
>>> newList = [2,5,29,1,9,59,3]
>>> newList
[2, 5, 29, 1, 9, 59, 3]
>>> newList[2:6]
[29, 1, 9, 59]
```

Expressions
 Assignments

 Assignments
 Variables

 Types
 Integers
 Floats
 Strings

d) Lists

• Indexing and ranges also work on Strings.

```
>>> myString
'hi there'
>>> myString[0]
'h'
>>> myString[5]
'e'
>>> myString[6]
'r'
>>> myString[0:6]
'hi the'
```

- 1. Expressions
- 2. Assignments
 - a) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

• Remember what assignments do

Memory		
Variable Name	Value	
x	1	
jadrian	100	
myNumber	12345	
noninteger	4.75	
myString	'hi there'	
endString	` class!'	
myList	[5,4,15]	
stringList	['hi','there','class']	
newList	[2,5,29,1,9,59,3]	

1. Expressions

- **2.** Assignmentsa) Variables
- 3. Types
 - a) Integers
 - b) Floats
 - c) Strings
 - d) Lists

Class Review

1. Expressions

- Evaluate input and returns some output (calculator)
- 2. Assignments: <variable> = <expression>
 - Store the value of the expression in the variable instead of outputting the value.
 - There is *always* an equals sign in an assignment
 - Variables can be named many things

3. Types

- Integers vs. Floats (Decimals)
- Strings in single quotes
- Lists are sets of other types
- We can index into Strings & Lists
 - Indexed starting at 0!

General Rule: Expressions for a particular type will *output* that same type!