More Summary Statistics

March 5, 2013

The Big Picture

Overall Goal

Build a Concordance of a text

- *Locations* of words
- *Frequency* of words

Today: Summary Statistics

- Administrative stuff
- Nitpicky Python details
- A new kind of statement
- Count the number of words in Moby Dick
- Compute the average word length of Moby Dick
- Find the longest word in Moby Dick
- Get the vocabulary size of Moby Dick

Administrative Stuff

- Homeworks are shifting back half a week
- TA office hours are probably changing
- Code you submit as homework must work!
  - (You will lose points for errors or incorrect answers)
- Homework must be done in Python 2.6.6

The Big Picture

Overall Goal

Build a Concordance of a text

- *Locations* of words
- *Frequency* of words

Today: Summary Statistics

- Administrative stuff
- Nitpicky Python details
- A new kind of statement
- Count the number of words in Moby Dick
- Compute the average word length of Moby Dick
- Find the longest word in Moby Dick
- Get the vocabulary size of Moby Dick

Literals vs. Variables

"How does Python know what's a variable?"

- A **literal** is a piece of data that we give directly to Python
  - `'hello'` is a string (`str`) literal
  - So are `"hey there"` and `'''wazzup'''`
  - `5` is an integer (`int`) literal
  - `32.8` is a floating-point (`float`) literal

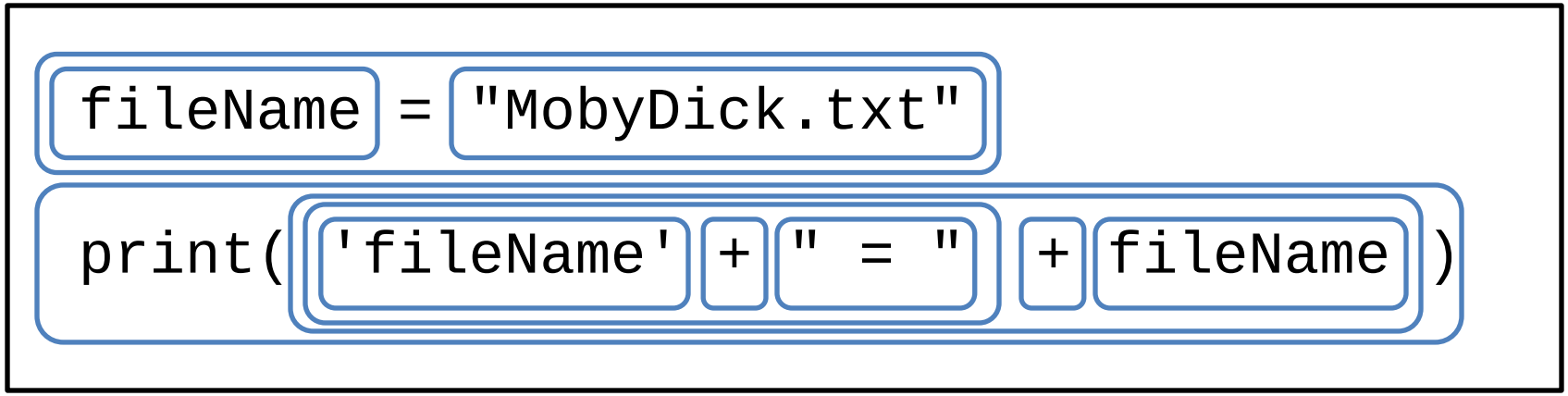
Literals vs. Variables

"How does Python know what's a variable?"

- Variable names are made up of:
  - Letters (uppercase and lowercase)
  - Numbers (but only after the first letter)
  - Underscores
- Names for functions and types follow the same rules
- Anything else must be a literal or operator!

Literals vs. Variables

"How does Python know what's a variable?"



The Big Picture

Overall Goal

Build a Concordance of a text

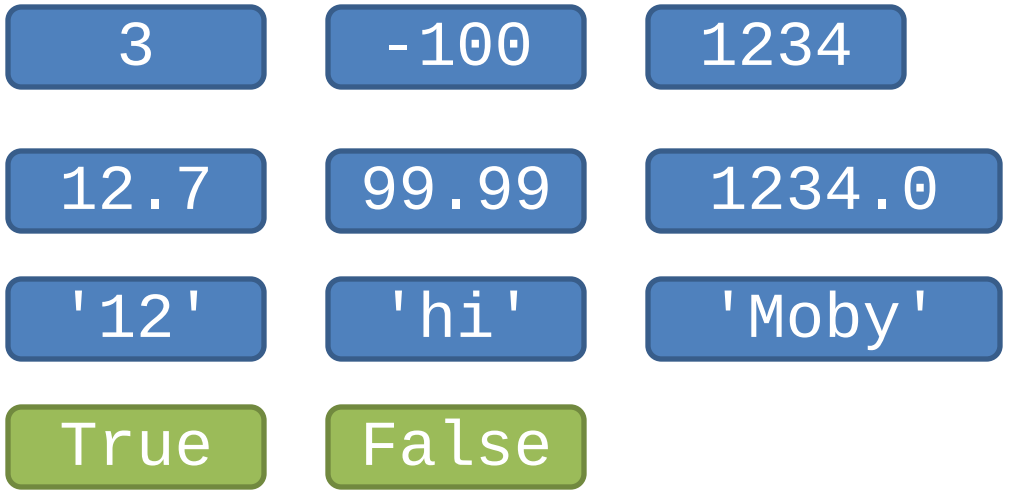
- *Locations* of words
- *Frequency* of words

Today: Summary Statistics

- Administrative stuff
- Nitpicky Python details
- A new kind of statement
- Count the number of words in Moby Dick
- Compute the average word length of Moby Dick
- Find the longest word in Moby Dick
- Get the vocabulary size of Moby Dick

Review: Basic Types

- Integers
- Floats
- Strings
- Booleans



New Type: Booleans

- Either `True` or `False`
  - Note the capitalization

```
>>> x = True
>>> x
True
>>> y = False
>>> y
False
```

New Type: Booleans

- Either `True` or `False`
  - Note the capitalization
- New Operators

Remember

Numerical Operators		
Operator	**Example**	**Result**
Sum	`1 + 2`	3
Difference	`1 - 2`	`-1`

New Type: Booleans

- Either `True` or `False`
  - Note the capitalization
- New Operators

Remember

Numerical Operators		
Operator	**Example**	**Result**
Sum	`1 + 2`	3
Difference	`1 - 2`	-1

Boolean Operators		
Operator	**Example**	**Result**
Equality	`1 == 2`	
Inequality	`1 != 2`	
Less Than	`1 < 2`	
Less Than or Equal To	`1 <= 2`	
Greater Than	`1 > 2`	
Greater Than or Equal To	`1 >= 2`	

New Type: Booleans

- Either `True` or `False`
  - Note the capitalization
- New Operators

Remember

Numerical Operators		
Operator	**Example**	**Result**
Sum	`1 + 2`	`3`
Difference	`1 - 2`	`-1`

Boolean Operators		
Operator	**Example**	**Result**
Equality	`1 == 2`	`False`
Inequality	`1 != 2`	`True`
Less Than	`1 < 2`	`True`
Less Than or Equal To	`1 <= 2`	`True`
Greater Than	`1 > 2`	`False`
Greater Than or Equal To	`1 >= 2`	`False`

New Type: Booleans

- Either `True` or `False`
  - Note the capitalization
- New Operators
- These are **expressions**
- Assignments have only **one** equals sign.

Boolean Operators		
Operator	**Example**	**Result**
Equality	`1 == 2`	`False`
Inequality	`1 != 2`	`True`
Less Than	`1 < 2`	`True`
Less Than or Equal To	`1 <= 2`	`True`
Greater Than	`1 > 2`	`False`
Greater Than or Equal To	`1 >= 2`	`False`

Boolean Types

Last Boolean Operators: `and`, `or` and `not`

Boolean Operators			
Operator	**Examples**		**Result**
`and`	`(4<5) and (6<3)`		
`or`	`(4<5) or (6<3)`		
`not`	`not(4<5)`		

Boolean Types

Last Boolean Operators: `and`, `or` and `not`

Boolean Operators			
Operator	**Examples**		**Result**
`and`	`(4<5) and (6<3)`	`True and False`	
`or`	`(4<5) or (6<3)`	`True or False`	
`not`	`not(4<5)`	`not(True)`	

Boolean Types

Last Boolean Operators: `and`, `or` and `not`

Boolean Operators			
Operator	**Examples**		**Result**
`and`	`(4<5) and (6<3)`	`True and False`	`False`
`or`	`(4<5) or (6<3)`	`True or False`	`True`
`not`	`not(4<5)`	`not(True)`	`False`

Boolean Types

Last Boolean Operators: `and`, `or` and `not`

Boolean Operators			
Operator	**Examples**		**Result**
`and`	`(4<5) and (6<3)`	`True and False`	`False`
`or`	`(4<5) or (6<3)`	`True or False`	`True`
`not`	`not(4<5)`	`not(True)`	`False`

More Examples		**Result**
`(4<5) and ((6<3) or (5==5))`		
`(5==4) or (not(6<3))`		

Boolean Types

Last Boolean Operators: `and`, `or` and `not`

Boolean Operators			
Operator	**Examples**		**Result**
`and`	`(4<5) and (6<3)`	`True and False`	`False`
`or`	`(4<5) or (6<3)`	`True or False`	`True`
`not`	`not(4<5)`	`not(True)`	`False`

More Examples		**Result**
`(4<5) and ((6<3) or (5==5))`	`True and (False or True)`	
`(5==4) or (not(6<3))`		

Boolean Types

Last Boolean Operators: `and`, `or` and `not`

Boolean Operators			
Operator	**Examples**		**Result**
`and`	`(4<5) and (6<3)`	`True and False`	`False`
`or`	`(4<5) or (6<3)`	`True or False`	`True`
`not`	`not(4<5)`	`not(True)`	`False`

More Examples		**Result**
`(4<5) and ((6<3) or (5==5))`	`True and (False or True)`	`True`
`(5==4) or (not(6<3))`		

Boolean Types

Last Boolean Operators: `and`, `or` and `not`

Boolean Operators			
Operator	**Examples**		**Result**
`and`	`(4<5) and (6<3)`	`True and False`	`False`
`or`	`(4<5) or (6<3)`	`True or False`	`True`
`not`	`not(4<5)`	`not(True)`	`False`

More Examples		**Result**
`(4<5) and ((6<3) or (5==5))`	`True and (False or True)`	`True`
`(5==4) or (not(6<3))`	`False or not(False)`	

Boolean Types

Last Boolean Operators: `and`, `or` and `not`

Boolean Operators			
Operator	**Examples**		**Result**
`and`	`(4<5) and (6<3)`	`True and False`	`False`
`or`	`(4<5) or (6<3)`	`True or False`	`True`
`not`	`not(4<5)`	`not(True)`	`False`

More Examples		**Result**
`(4<5) and ((6<3) or (5==5))`	`True and (False or True)`	`True`
`(5==4) or (not(6<3))`	`False or not(False)`	`True`

Review: Statements

- Expression Statements
- Assignment Statements
- `For` Statements
- `If` Statements

Gives you output

Stores a value in a variable

"For each element in `myList`, do something"

If A is true, then do something, otherwise do something else

Boolean Statements (If Stmts)

- "If something's true, do A"

```
def compare(x, y):
        if x > y:
                print(x, ' is greater than ', y)
```

Boolean Statements (If Stmts)

- "If something's true, do A, otherwise, do B"

```
def compare(x, y):
        if x > y:
                print(x, ' is greater than ', y)
        else:
                print(x, ' is less than or equal to ', y)
```

Boolean Statements (If Stmts)

- "If something's true, do A, otherwise, check something else; if that's true, do B, otherwise, do C"

```
def compare(x, y):
	if x > y:
		print(x, ' is greater than ', y)
	else:
		if x < y:
			print(x, ' is less than ', y)
		else:
			print(x, ' is equal to ', y)
```

Review: Other Things

- Lists (a type of **data structure**)

```
[0,1,2]
['hi','there']
['hi',0.0]
[1,2,3,4,5,True,False,`true','one]
```

Review: Other Things

- Lists (a type of **data structure**)

```
[0,1,2]  ['hi','there']  ['hi',0.0]
[1,2,3,4,5,True,False,`true','one]
```

- Files (an **object** that we can open, read, close)

```
myFile = open(fileName,`r')
```

The Big Picture

Overall Goal

Build a Concordance of a text

- *Locations* of words
- *Frequency* of words

Today: Summary Statistics

- Administrative stuff
- Nitpicky Python details
- A new kind of statement
- Count the number of words in Moby Dick
- Compute the average word length of Moby Dick
- Find the longest word in Moby Dick
- Get the vocabulary size of Moby Dick

A Shortcut to List Length

Preloaded Functions		
`len`	List	Integer

```
>>> len([3, 47, 91, -6, 18])

>>> uselessList = ['contextless', 'words']
>>> len(uselessList)

>>> creature = 'woodchuck'
>>> len(creature)
```

Python Functions

Preloaded Functions		
`len`	List OR String	Integer

Python Functions

Preloaded Functions		
`len`	List OR String	Integer
`float`	Number (as an Integer, Float, or String)	Float
`int`	Number (as an Integer, Float, or String)	Integer
`str`	Integer, Float, String, or List	String

These functions *cast* a variable of one type to another type

Python Functions

Preloaded Functions		
`len`	List OR String	Integer
`float`	Number (as an Integer, Float, or String)	Float
`int`	Number (as an Integer, Float, or String)	Integer
`str`	Integer, Float, String, or List	String
`range`	Two Integers 1. Start Index (Inclusive) 2. End Index (Exclusive)	List of Integers

These functions *cast* a variable of one type to another type

The Big Picture

Overall Goal

Build a Concordance of a text

- *Locations* of words
- *Frequency* of words

Today: Summary Statistics

- Administrative stuff
- Nitpicky Python details
- A new kind of statement
- Count the number of words in Moby Dick
- Compute the average word length of Moby Dick
- Find the longest word in Moby Dick
- Get the vocabulary size of Moby Dick

Compute the Average Word Length of Moby Dick

```
def avgWordLengthInMobyDick():
    '''Gets the average word length in MobyDick.txt'''




    return avg
```

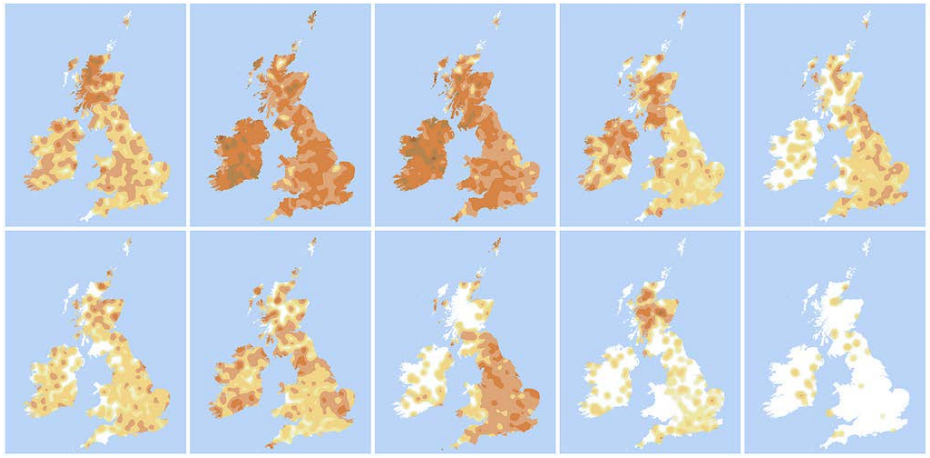
Compute the Average Word Length of Moby Dick

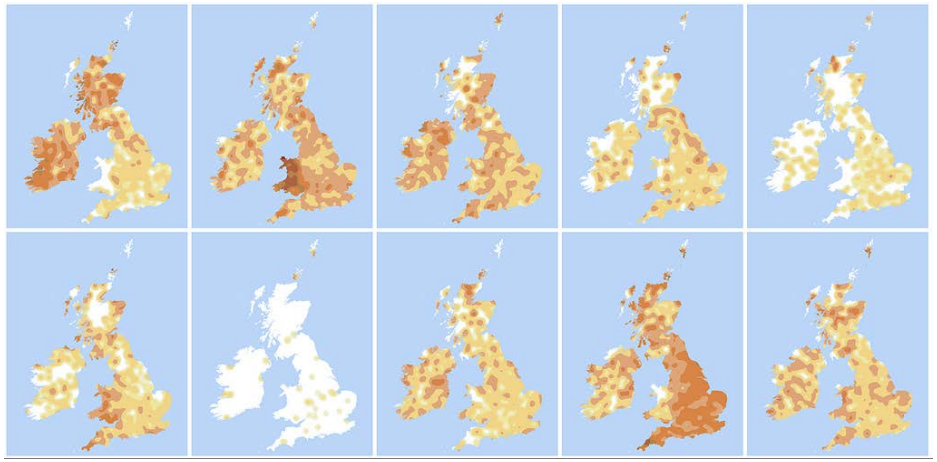
```
def avgWordLengthInMobyDick():
    '''Gets the average word length in MobyDick.txt'''
    myList = readMobyDick()
    s = 0
    for word in myList:
        s = s + len(word)
    avg = s/float(len(myList))
    return avg
```

Is our Program Accurate?

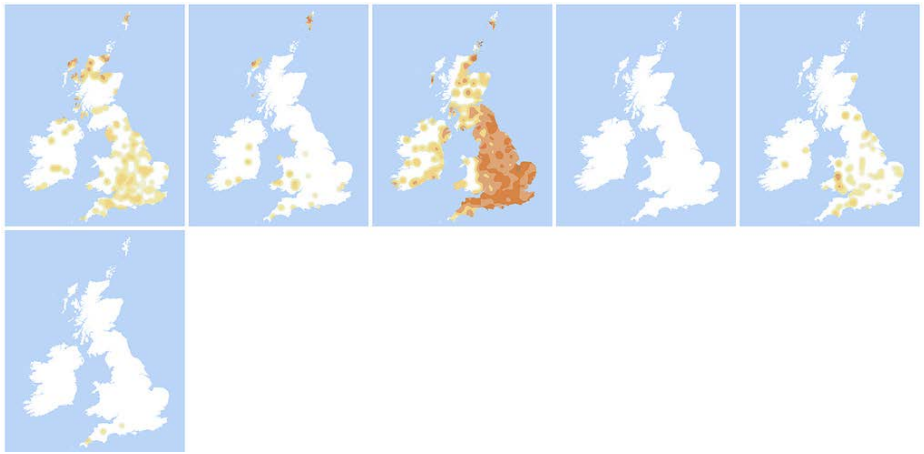
```
>>> MDList = readMobyDick()
>>> MDList[0:99]
['CHAPTER', '1', 'Loomings', 'Call', 'me', 'Ishmael.',
'Some', 'years', 'ago--never', 'mind', 'how', 'long',
'precisely--', 'having', 'little', 'or', 'no', 'money', 'in',
'my', 'purse,', 'and', 'nothing', 'particular', 'to',
'interest', 'me', 'on', 'shore,', 'I', 'thought', 'I',
'would', 'sail', 'about', 'a', 'little', 'and', 'see', 'the',
'watery', 'part', 'of', 'the', 'world.', 'It', 'is', 'a',
'way', 'I', 'have', 'of', 'driving', 'off', 'the', 'spleen',
'and', 'regulating', 'the', 'circulation.', 'Whenever', 'I',
'find', 'myself', 'growing', 'grim', 'about', 'the',
'mouth;', 'whenever', 'it', 'is', 'a', 'damp,', 'drizzly',
'November', 'in', 'my', 'soul;', 'whenever', 'I', 'find',
'myself', 'involuntarily', 'pausing', 'before', 'coffin',
'warehouses,', 'and', 'bringing', 'up', 'the', 'rear', 'of',
'every', 'funeral', 'I', 'meet;', 'and']
```

Break





Alphabet Maps of the UK and Ireland (url on site)



The Big Picture

Overall Goal

Build a Concordance of a text

- *Locations* of words
- *Frequency* of words

Today: Summary Statistics

- Administrative stuff
- Nitpicky Python details
- A new kind of statement
- Count the number of words in Moby Dick
- Compute the average word length of Moby Dick
- Find the longest word in Moby Dick
- Get the vocabulary size of Moby Dick

Get the Longest Word in Moby Dick

```
def getLongestWordInMobyDick():
    '''Returns the longest word in MobyDick.txt'''




    return longestword
```

Get the Longest Word in Moby Dick

```
def getLongestWordInMobyDick():
    '''Returns the longest word in MobyDick.txt'''
    myList = readMobyDick()
    longestword = ""
    for word in myList:
        if len(word) > len(longestword):
            longestword = word
    return longestword
```

Get the Longest Word in Moby Dick

```
def getLongestWordInMobyDick():
    '''Returns the longest word in MobyDick.txt'''
    myList = readMobyDick()
    longestword = ""
    for word in myList:
        if len(word) > len(longestword):
            longestword = word
    return longestword
```

Is our program accurate?

The Big Picture

Overall Goal

Build a Concordance of a text

- *Locations* of words
- *Frequency* of words

Today: Summary Statistics

- Administrative stuff
- Nitpicky Python details
- A new kind of statement
- Count the number of words in Moby Dick
- Compute the average word length of Moby Dick
- Find the longest word in Moby Dick
- Get the vocabulary size of Moby Dick

Boolean Expressions on Strings

Boolean Operators on Strings		
Operator	**Example**	**Result**
Equality	`‘a’ == ‘b’`	`False`
Inequality	`‘a’ != ‘b’`	`True`
Less Than	`'a' < 'b'`	`True`
Less Than or Equal To	`'a' <= 'b'`	`True`
Greater Than	`'a' > 'b'`	`False`
Greater Than or Equal To	`'a' >= 'b'`	`False`

Writing a `vocabSize` Function

```
def vocabSize():
    myList = readMobyDick()
    uniqueList = noReplicates()
    return len(uniquelist)
```

Writing a `vocabSize` Function

```
def vocabSize():
    myList = readMobyDick()
    uniqueList = noReplicates()
    return len(uniquelist)
```

```
def noReplicates(wordList):
    '''takes a list as
argument, returns a list free
of replicate items.  slow
implementation.'''
```

```
def isElementOf(myElement,myList):
    '''takes a string and a list
and returns True if the string is
in the list and False
otherwise.'''
```

What does `slow implementation` mean?

- Re-comment the lines and run `vocabSize()`
  - Hint: Ctrl-C (or Command-C) will abort the call.

What does `slow implementation` mean?

- Re-comment the lines and run `vocabSize()`
  - Hint: Ctrl-C (or Command-C) will abort the call.

- There's a *faster* way to write `noReplicates()`
  - What if we can sort the list?

```
['a','a','a','and','and','at',…,'zebra']
```