Textual Analysis & Introduction to Python
Project Proposal Evaluations
Handling data-selection nicely

- `FILTER(range, condition1, cond2, ...)`
- Returns a filtered version of the source range, returning only rows or columns which meet the specified conditions.
- `condition1`: a column or row containing true/false values, or an array formula
Data-loading

• If you’re loading just a few columns from something with LOTS of data...
• ...the `QUERY()` function may be a big help.
Why averaging averages is bad

- In one year, UC Berkeley hired 30 faculty. 23 were male, 7 female, even though the applicant pool was 60% female.
- Sounds like bias!
- [numbers made up for this example, but this actually happened, with somewhat different numbers]
Which departments were biased?

- UCB looked at the data *per department* to see which ones hired unfairly

- NONE DID!
A 3-hire example

<table>
<thead>
<tr>
<th></th>
<th>Dept 1</th>
<th>Dept 2</th>
<th>Dept 3</th>
<th>Total</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>10</td>
<td>20</td>
<td>12</td>
<td>32</td>
<td>13.6% M</td>
</tr>
<tr>
<td>Female</td>
<td>200</td>
<td>1</td>
<td>2</td>
<td>203</td>
<td>86.4% F</td>
</tr>
<tr>
<td>Hired</td>
<td>1 F</td>
<td>1M</td>
<td>1M</td>
<td>235</td>
<td>67% male</td>
</tr>
</tbody>
</table>

- Every department’s hiring ratio was fair
- The average ratio looked unfair!
- Conclusion: averaging ratios can lead to inconsistent results.
Today’s Class

• Intro to text analysis problems
• Intro to Python
Get Full Credit on Project 1

• Use sheets for intermediate results, interactivity
• Display your data in different ways (graph, table, etc.)
• Follow the rules: only data, parameters, labels, and formulas.
• Use “notes” on cells to explain anything out of the ordinary
• Try to avoid “hand work” and use formulas instead!
Project 1 (cont.)

• Remember to address your claim
  – My hypothesis is correct/incorrect because ...
  – X% of the time, my hypothesis was correct...

• Be sure to explain what obstacles you encountered

• Have a “Discussion and Conclusion” section (or something similarly named) where you reflect on things like “Is this data too unreliable for me to trust the conclusion?” and “Was a threshold of 80% really reasonable?”, and “Did eliminating countries that lacked data for any single year make the analysis worthless?”

• Look at the rubric!
Intermediate Results

Put your raw data on its own sheet and refer to it using a formula when you do your analysis on other sheets.
Intermediate Results

Use a new sheet when the current sheet already has a table with some meaningful data in it. Don’t lose that data.
Interactivity

Use **data validation**, probably with a list (pull down), to add some interactive component.

See ACT1-4 for an example using MATCH and OFFSET.
Presentation

Try out different ways to present your data. Make a chart from your final results. If you’re not sure what to do, ask a TA or me.
Address Your Hypothesis

Remember to relate your results back to the hypothesis. Was it true? Was it true in some cases? Why might it have been false?
Obstacles and Reflection

What was difficult? What are the limitations of the analysis you did?

You must reflect on your project and what you learned.
Check the rubric \textit{before you submit}
A brief review of things you didn’t know you’d learned

• In a spreadsheet, there are many types of data
• Numbers (start with +/- or a digit)
• Strings (nondigit-start, or start with ‘ ’)
• Formulas (start with = )
• Ranges (B2, B2:B4, B2:D5)
• Errors (#N/A)
• Blanks
What shows up in a cell

- If a formula evaluates to a number or string, that number or string
- If it evaluates to a range, the value in the first cell of that range ...sometimes
  - If you write =A1:A6, you get A1
  - If you write =OFFSET(A1:A6, 0, 0), Gsheets fills in adjacent cells; excel just fills in one cell
- If evaluation leads to an error, then #N/A
- Mostly, we never notice any of this
- In Python, the rules have greater consistency, and because results aren’t instantly visible, knowing the rules matters more
Textual Analysis

Define Problem

Find Data

Write a set of instructions

Computer

Solution
Textual Analysis

1. Define Problem
2. Find Data
3. Write a set of instructions
4. Computer
5. Solution

ACTACGTCGACTACGATCAC
GATCGCGCAGTCGATTTT
ACGATCAGCTACGATCGATC
TACGATCGTATGCTGTGATCG
Textual Analysis

Define Problem

Find Data

Write a set of instructions

Build a Concordance of a text
  • Locations of words
  • Frequency of words

Solution

Computer

ACTACGTGCACTACGATCAC
GATCGCGCGATCACGTATTT
ACGATCAGCTACGATCGATC
TACGATCGTAGCTGTGATCG

ACTACGTGCACTACGATCAC
GATCGCGCGATCACGTATTT
ACGATCAGCTACGATCGATC
TACGATCGTAGCTGTGATCG
Concordances

Alphabetical index of all words in a text

<table>
<thead>
<tr>
<th>Word</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>4,7,10,27</td>
</tr>
<tr>
<td>Banana</td>
<td>77,110,130</td>
</tr>
<tr>
<td>Carrot</td>
<td>50,101</td>
</tr>
<tr>
<td>Date</td>
<td>9</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Concordances

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

http://en.wikipedia.org/wiki/Concordance_(publishing)
Concordances

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

http://en.wikipedia.org/wiki/Concordance_(publishing)
Concordances

• 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

http://en.wikipedia.org/wiki/Concordance_(publishing)
Concordances

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

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

• 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
Concordances

• 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
From Concordance to Word Frequency

Suppose our text has 1000 words total.

<table>
<thead>
<tr>
<th>Word</th>
<th>Page Numbers</th>
<th># of Occurrences</th>
<th>Word Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>4,7,10,27</td>
<td>4</td>
<td>4/1000</td>
</tr>
<tr>
<td>Banana</td>
<td>77,110,130</td>
<td>3</td>
<td>3/1000</td>
</tr>
<tr>
<td>Carrot</td>
<td>50,101</td>
<td>2</td>
<td>2/1000</td>
</tr>
<tr>
<td>Date</td>
<td>9</td>
<td>1</td>
<td>1/1000</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Google Ngrams

- Google “Google n-grams”
- **ngram**: 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?
Textual Analysis

Define Problem

Write a set of instructions

Find Data

Computer

Solution

Build a Concordance of a text

- Locations of words
- Frequency of words

- Word frequencies across time

ACTACGTCGACTACGATCAC
GATCGCGCGATCACGTATTT
ACGATCAGCTACGATCGATC
TACGATCGTAGCTGTGATCG

ACTACGTCGACTACGATCAC
GATCGCGCGATCACGTATTT
ACGATCAGCTACGATCGATC
TACGATCGTAGCTGTGATCG
The Wizard of OZ

• About 40 Books, written by 7 different authors

Lyman Frank Baum

Ruth Plumly Thompson

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

• About 40 Books, written by 7 different authors

Lyman Frank Baum
(1856-1919)

Ruth Plumly Thompson

Published in 1921

http://www.ssc.wisc.edu/~zzeng/soc357/OZ.pdf
The Wizard of Oz

• About 40 Books, written by 7 different authors

Lyman Frank Baum (1856-1919)

Ruth Plumly Thompson

Published in 1921

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

Textual Analysis

Define Problem

Find Data

Write a set of instructions

Solution

Build a Concordance of a text

• *Locations* of words

• *Frequency* of words

• Word frequencies across time
  • Determine authorship

ACTACGTCGACTACGATCAC
GATCGCGCGATCACGTATTT
ACGATCAGCTACGATCGATC
TACGATCGTAGCTGTGATCG

ACTACGTCGACTACGATCAC
GATCGCGCGATCACGTATTT
ACGATCAGCTACGATCGATC
TACGATCGTAGCTGTGATCG
Stoppard’s Arcadia

**Bernard**: Yes. One of my colleagues believed he had found an unattributed short story by D. H. Lawrence, and he analysed it on his home computer, most interesting, perhaps you remember the paper?

**Valentine**: Not really. But I often sit with my eyes closed and it doesn't necessarily mean I'm awake.

**Bernard**: Well, by comparing sentence structures and so forth, this chap showed that there was a ninety per cent chance that the story had indeed been written by the same person as *Women in Love*. To my inexpressible joy, one of your maths mob was able to show that on the same statistical basis there was a ninety per cent chance that Lawrence also wrote the *Just William* books and much of the previous day's *Brighton and Hove Argus*. 
Textual Analysis

Define Problem

Find Data

Write a set of instructions

Solution

Build a Concordance of a text

• Locations of words
• Frequency of words

• Word frequencies across time
  • Determine authorship
  • Count labels to determine liberal media bias

Computer

ACTACGTTCGACTACGATCAGCTACGATCGATCAC
GATCGCGCGATCACGTATTT
ACGATCAGCTACGATCGATC
TACGATCGTAGCTGTGATCG

CSCI 0931 - Intro. to Comp. for the Humanities and Social Sciences
How are we going to analyze texts?

Excel

Numerical Data
How are we going to analyze texts?

Excel

Numerical Data

Textual Data

firehow.com
How are we going to analyze texts?

Makita Cordless Chain Saw, $270

Textual Data
How are we going to analyze texts?

Python: A Programming Language
Free!

Textual Data

9poundhammer.blogspot.com
Textual Analysis

Define Problem

Find Data

Write a set of instructions

Solution

Build a Concordance of a text
• **Locations** of words
• **Frequency** of words
• Word frequencies across time
  • Determine authorship
  • Count labels to determine liberal media bias

Python

ACTACGTGCAGACTACGATCAC
GATCGCGCGATCAGTATTT
ACGATCAGCTACGATCGATC
TACGATCGTAGCTGTGATCG
Break

- Everyone: Open IDLE (Python GUI)
“Python”
“Python”

• A language for giving the computer instructions. It has syntax and semantics.
“Python”

• A language for giving the computer instructions. It has syntax and semantics.

• Might say “write a Python program”, meaning “write instructions in the Python language”
“Python”

• **A language** for giving the computer instructions. It has syntax and semantics.

• Might say “write a Python program”, meaning “write instructions in the Python language”

• There is an **interpreter** (e.g., IDLE) that takes Python instructions and executes them with the CPU, etc.
Introduction to Python

• **Expressions** are *inputs* that Python evaluates
  – Expressions return an *output*
  – Like using a calculator
Introduction to Python

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

Type the expressions below after ‘```’ and hit Enter

```
>>> 4+2
6
>>> 4-2
2
>>> 4*2
8
>>> 4/2
2
```
Introduction to Python

- **Assignments** do not have an output, they are stored in memory.
Introduction to Python

• **Assignments** do not have an output, they are stored in memory.
  – We’ve done this kind of thing in Excel

![Excel screenshot](image)

We have assigned the number 1 to cell A1.

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

• **Assignments** do not have an output, they are stored in memory.
  – We’ve done this kind of thing in Spreadsheets

We have assigned the number 1 to cell A1.

Let’s rename cell A1 to x.
Introduction to Python

• **Assignments** do not have an output, they are stored in memory.
  – We’ve done this kind of thing in Spreadsheets

```python
>>> x = 1
```

Let’s rename cell A1 to x.

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

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

  - We’ve done this kind of thing in Spreadsheets

```python
>>> x = 1
```

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1</td>
</tr>
</tbody>
</table>
Introduction to Python

• **Assignments** do not have an output, they are *stored in memory*.
  – We’ve done this kind of thing in Spreadsheets

  ```
  >>> x = 1
  >>> x+1
  2
  >>> (x+2) * 3
  9
  ```

  – We can now use `x` in expressions!

<table>
<thead>
<tr>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Name</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>x</td>
</tr>
</tbody>
</table>

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

• You can name your variables anything
  
  
  >>> steve = 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.

CSCI 0931 - Intro. to Comp. for the Humanities and Social Sciences
Choices

• Why are *those* the rules for names?
• Someone thought about it and made a choice
• Usually based on years of experience
• Many choices seem crazy...
  – Until one day you see they’re obviously correct
Introduction to Python

• Try this:  

```python
>>> 3/2
```
Introduction to Python

• Try this:  

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

```python
>>> 3/2
>>> type(3/2)
<type 'int'>
>>> type(1.5)
<type 'float'>
```
Introduction to Python

• Try this:  

```python
>>> 3/2
```

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

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

• Floats are numbers that display with decimal points.

```python
>>> 3.0/2.0
1.5
```
**Introduction to Python**

- **Try this:** `3/2`

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

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

- **Floats are decimals.**

```python
>>> 3.0/2.0
1.5
```

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

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

```
>>> 'hi'
'hi'
>>> myString = 'hi there'
>>> myString
'hi there'
```
Introduction to Python

• **Strings** are sequences of characters, surrounded by single quotes.
  
  ```python
  >>> 'hi'
  'hi'
  >>> myString = 'hi there'
  >>> myString
  'hi there'
  ```

• The + operator concatenates

General Rule: Expressions for a particular type will **output** that same type!
• **Strings** are sequences of characters, surrounded by single quotes.

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

• The + operator concatenates

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

• **Lists** are an ordered collection of items

```python
>>> [5,10,15]
[5, 10, 15]
>>> myList = [5,10,15]
>>> myList
[5, 10, 15]
>>> stringList = ['hi','there','class']
>>> stringList
['hi', 'there', 'class']
```
Introduction to Python

- **Lists** are an ordered collection of items

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

- Individual items are *elements*
- The + operator concatenates

```python
>>> myList + stringList
[5, 10, 15, 'hi', 'there', 'class']
```
Introduction to Python

• To get an element from a list, use the expression `myList[i]` where `i` is the index. Often spoken: “myList sub i”

• List indices start at 0!

```python
>>> myList[0]
5
>>> myList[1]
10
>>> myList[2]
15
```

• What does `myList[1] = 4` do?
Introduction to Python

• 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]
```
Introduction to Python

- **Indexing** and **ranges** also work on Strings.

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

• Remember what assignments do

<table>
<thead>
<tr>
<th>Memory</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Name</td>
<td>Value</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>steve</td>
<td>100</td>
</tr>
<tr>
<td>myNumber</td>
<td>12345</td>
</tr>
<tr>
<td>noninteger</td>
<td>4.75</td>
</tr>
<tr>
<td>myString</td>
<td>'hi there'</td>
</tr>
<tr>
<td>endString</td>
<td>' class!'</td>
</tr>
<tr>
<td>myList</td>
<td>[5, 4, 15]</td>
</tr>
<tr>
<td>stringList</td>
<td>['hi', 'there', 'class']</td>
</tr>
<tr>
<td>newList</td>
<td>[2, 5, 29, 1, 9, 59, 3]</td>
</tr>
</tbody>
</table>
Python So Far (to be updated/refined!)

1. Expressions
   - Evaluate *input* and returns some *output* (calculator)

2. Variable 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
   - List assignments: `<listvar>[<index>] = <expression>`

3. Types
   - Integers vs. Floats (Decimals)
   - Strings in single quotes
   - Lists are sets of other types
   - We can index into Strings & Lists
   - General Rule: Expressions for a particular type will *output* that same type!