Cultural Analytics

ENGL 64.05

Fall 2019

Prof. James E. Dobson

Key Python Data Types

- Integers
- Floating Point Numbers
- Strings
- Lists
- Dictionaries

Dictionaries

- Mapping from keys to values.
  - Downey gives us an example of a translation table:

    ```
    eng2sp['one'] = 'uno'
    ```

  - We can count the frequency of words or phrases in a text by making our values integers:

    ```
    vocab['evil'] = 2
    ```

  - Other kinds of dictionaries?
- Incredibly useful for storing certain kinds of data.
- As we can have a list-of-lists, you can place dictionaries within a list.

Creating and Using Dictionaries

```
In [1]:  # Create a new dictionary
         vocab=dict()
```

```
In [2]:  # key and value pairs
         # key = 'the' (string)
         # value = 1 (integer)

         vocab['the'] = 1
```

```
In [3]:  # display the keys
         vocab.keys()
```

Out[3]: dict_keys(['the'])

```
In [4]:  # display the values
         vocab.values()
```

Out[4]: dict_values([1])

```
In [5]:  # loop and process both
         for key, value in vocab.items():
             print(key,value)
```

the 1

```
In [6]:  # now update the dictionary
         vocab['the'] = vocab['the'] + 1
```

```
In [7]:  # That's a little confusing, so we can also use += 1 logic to
         # increment our counter
         vocab['the'] += 1
```

```
In [8]:  # loop and process both
         for key, value in vocab.items():
             print(key,"->",value)
```

the -> 3

```
In [9]:   # Lists can have duplicate items because items are not keys.
          # Dictionaries guarantee a single key
          parts_of_speech=dict()
          parts_of_speech['pronouns'] = ['she','he','they','zir']
```

```
In [10]:  for pronoun in parts_of_speech['pronouns']:
              print(pronoun,"filled out the form")
```

```
she filled out the form
he filled out the form
they filled out the form
zir filled out the form
```
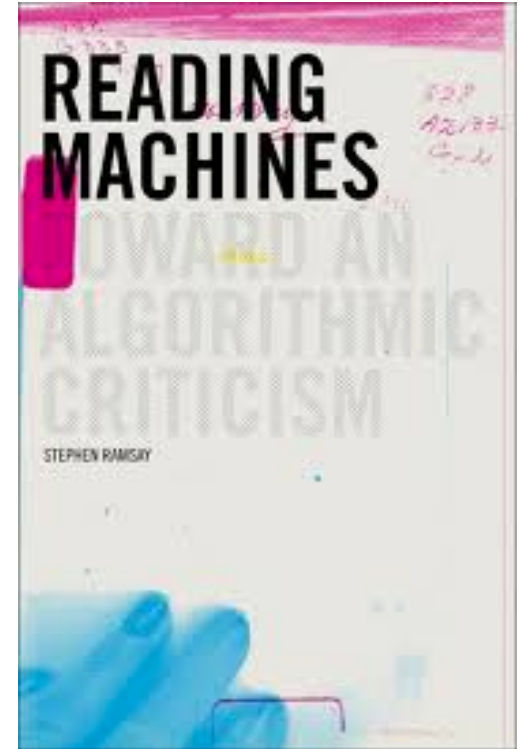
```
In [11]:  if 'zir' in parts_of_speech['pronouns']:
              print("recognized pronoun")
          else:
              print("unrecognized pronoun")
```

```
recognized pronoun
```

Dictionary or List?

- How do we decide when to use a dictionary instead of a list?

Stephen Ramsay
Reading Machines: Toward an Algorithmic Criticism

Algorithmic Criticism

Stephen Ramsay writes: "If algorithmic criticism is to have a central tenet, it is this: that the narrowing constraints of computational logic—the irreducible tendency of the computer toward enumeration, measurement, and verification—is fully compatible with the goals of criticism…For while it is possible, and is some cases useful, to confine algorithmic procedures to the scientific realm, such procedures can be made to conform to the methodological project of *inventio* without transforming the nature of computation or limiting the rhetorical range of critical inquiry. This is possible because critical reading practices already contain elements of the algorithmic" (16).

Reading as a Deformance

- Ramsay argues that all readings involve playful manipulation of the text.

- Russian formalists and others suggest that the goal of interpretation is to make the text strange, to *defamiliarize* the text.

- He's advocated an interpretive practice of screwing around or a "screwmeneutics."

- Human or machine reading is a processing of deformation, an estranging rendering of the text that is both creative and critical.

Entropic Poem

Richard Hovey, "To Prof. C. F. Richardson"

```
In [1]:  hovey_poem = """
         TO PROF. C. F. RICHARDSON
         (For the dedication of a book.)
         SUCH as the seashore gathers from the sea —
         Shells whose glad opal sunlight makes more glad,
         And dead men's bones by bitter seaweed clad —
         Teacher and friend, these songs I send to thee.
         Gay things and ghastly mingled, seem to me
         Here are alike; the merry and the sad,
         The trivial and tragic, good and bad,
         For so I find the ways of life to be.
         Evil and good are woven upon the loom
         Of fate in such inextricable wise
         That no man may be bold to judge and say,
         "This thing is good, that evil," till the day
         When God shall blazon on regenerate skies
         The justice of His pardon and His doom.
         """
```

```
In [2]:  vocab = dict()
         for word in hovey_poem.split():
             word = word.lower()
             if word not in vocab:
                 vocab[word] = hovey_poem.lower().count(word)
                 hovey_poem.count(word)
                 print(vocab[word],word)
```

TO PROF. C. F. RICHARDSON
(For the dedication of a book.)
SUCH as the seashore gathers from the sea —
Shells whose glad opal sunlight makes more glad,
And dead men's bones by bitter seaweed clad —
Teacher and friend, these songs I send to thee.
Gay things and ghastly mingled, seem to me
Here are alike; the merry and the sad,
The trivial and tragic, good and bad,
For so I find the ways of life to be.
Evil and good are woven upon the loom
Of fate in such inextricable wise
That no man may be bold to judge and say,
"This thing is good, that evil," till the day
When God shall blazon on regenerate skies
The justice of His pardon and His doom.

```
5 to            1 these         1 man
1 prof.         1 songs         1-May
1 c.            30 i            2 be
2 f.            1 send          1 bold
1 richardson    1 thee.         1 judge
1 (for          1 gay           1 say,
13 the          1 things        1 "this
1 dedication    1 ghastly       2 thing
5 of            1 mingled,      5 is
46 a            1 seem          1 good,
1 book.)        3 me            1 evil,"
2 such          1 here          1 till
3 as            2 are           1 day
1 seashore      1 alike;        1 when
1 gathers       1 merry         1 god
1 from          1 sad,          1 shall
3 sea           1 trivial       1 blazon
2 —             1 tragic,       8 on
1 shells        3 good          1 regenerate
1 whose         1 bad,          1 skies
2 glad          2 for           1 justice
1 opal          3 so            3 his
1 sunlight      1 find          1 pardon
1 makes         1 ways          1 doom.
1 more          1 life
1 glad,         1 be.
9 and           2 evil
1 dead          1 woven
1 men's         1 upon
1 bones         1 loom
1 by            1 fate
1 bitter        6 in
1 seaweed       1 inextricable
1 clad          1 wise
1 teacher       2 that
1 friend,       1 no
```
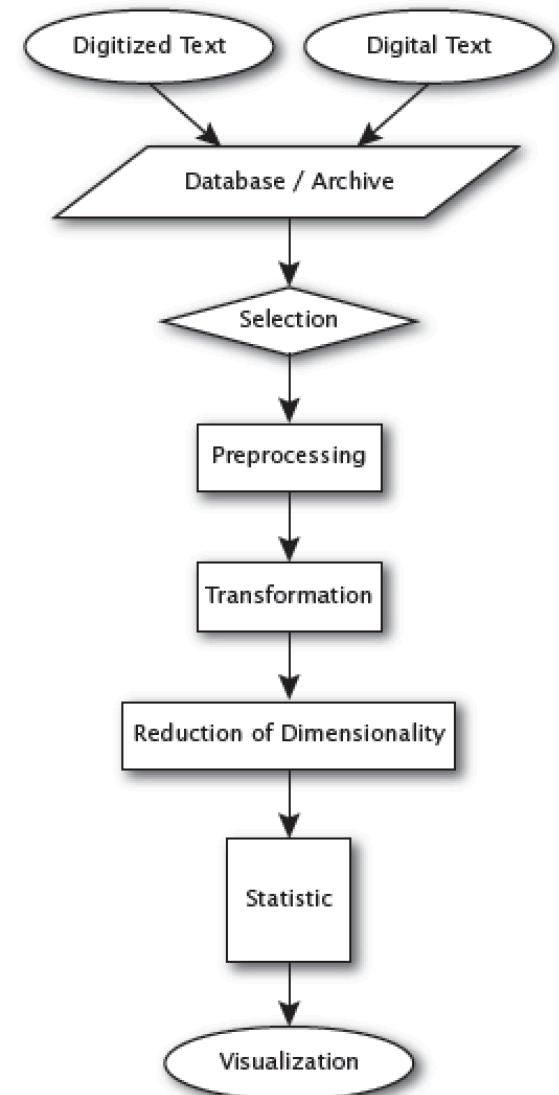
"Protocols, Methods, Workflows:
Digital Ways of Reading"
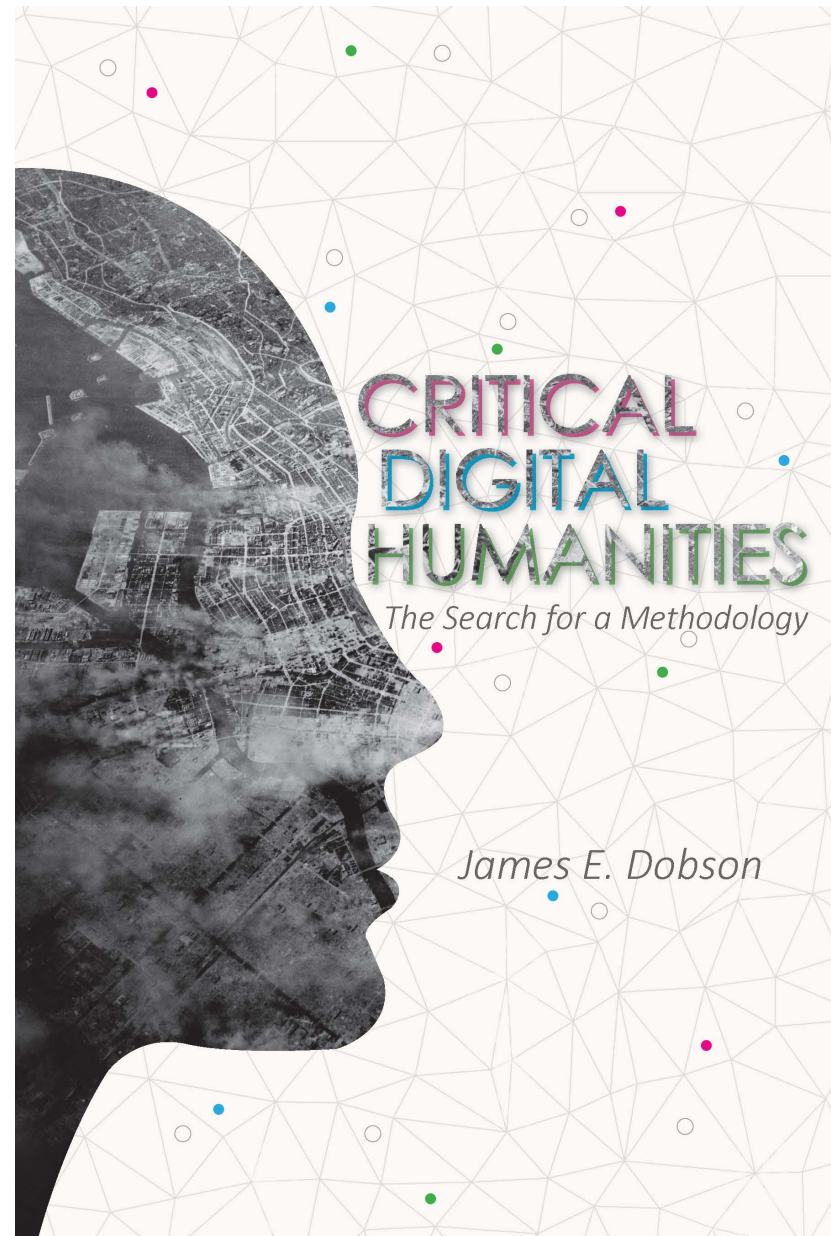
Workflows as the Object of Analysis

"Stating that the tools have a material existence enables them to hide and obscure the underlying code and algorithms. If one is going to engage with the artifacts of the digital humanities, then the artifacts themselves need to be open and available for researchers to examine and understand. A critical digital humanities takes as its proper object the entire workflow used in computational analysis. No areas of inquiry, from the origins of the text and data to the history and ideology of the algorithms used, can remain closed to the gaze of the critic" (27).

Digitized Text → Database / Archive ← Digital Text
Database / Archive → Selection → Preprocessing → Transformation → Reduction of Dimensionality → Statistic → Visualization

Understanding Parameters

```
collocations(self, num=20, window_size=2)
Print collocations derived from the text,
ignoring stopwords.
:seealso: find_collocations
:param num: The maximum number of collocations
to print.
:type num: int
:param window_size: The number of tokens
spanned by a collocation
    (default=2)
:type window_size: int
```

Understanding Choices

Collocated terms:

n't know; New York; Cape Palmas; Methodist Church; shall forget; Amanda Smith; Bishop Taylor; Holy Ghost; never forget; Sierra Leone; camp meeting; Praise Lord; said "Lord; shall never; Mrs. Smith; God bless; Sister Smith; Sabbath School; could get; Lord helped

Preprocessed collocated terms:

cape palmas; amanda smith; new york; camp meeting; shall forget; praise lord; methodist church; sierra leone; bishop taylor; holy ghost; sister smith; never forget; god bless; shall never; mrs. smith; lord help; lord helped; two weeks; thank god; miss drake

Homework #1

- Three parts in three major Jupyter cells.
    1. Patterns in Titles
    2. Entropic Poem
    3. Type/Token ratio
- Each major cell block some completion or modification AND a short critical response to the execution.
- Download from JupyterHub.
- Upload to Canvas by Friday 11:59PM
- Make sure to save!

Next Class

- Matthew Jockers, "Measures of Lexical Variety" (examples written in R).

- Clair Lemercier and Clari Zalc, "Sources and Samples."