DSC 201: Data Analysis & Visualization

Advanced Python and Classes

Dr. David Koop
Color != Wavelength

[Diagram showing relative energy density vs. wavelength with peaks for yellow and brown colors.]

[via M. Meyer]
Color Blindness

Normal

Protaganopia

Deuteranopia

Tritanopia

[via M. Meyer]
Categorical vs. Ordered

- **Hue** has *no implicit ordering*: use for categorical data
- Saturation and luminance do: use for ordered data

![Color Wheel Diagram](Munzner (ill. Maguire), 2014)
Continuous Colormap

US EPA Regional Oxidant Model -- Midwest Ozone (ppbv): June 26, 1987, 18:00

[Bergman et al., 1995]
Segmented Colormap

US EPA Regional Oxidant Model -- Midwest
Ozone (ppbv): June 26, 1987, 18:00

[Bergman et al., 1995]
Don't Use Rainbow Colormaps

Which has a discontinuity?

[M. Bussonnier]
Other Colormaps Work Better

Which has a discontinuity?

[M. Bussonnier]
Multiform

[Improvise, Weaver, 2004]
Small Multiples & Brushing

• Same encoding, but different data in each view (e.g. SPLOM)

[http://bl.ocks.org/mbostock/4063663]
Coordinated Views

Nasdaq 100 Index 1985/11/01-2012/06/29

Yearly Performance (radius: fluctuation/index ratio, color: gain/loss)

Days by Gain/Loss

Quarters

Day of Week

Days by Fluctuation(%)
Superimposed Line Charts

D. Koop, DSC 201, Fall 2018

[M. Bostock, http://bl.ocks.org/mbostock/3884955]
Restaurant locations are derived from the New York City Department of Health and Mental Hygiene database. Due to the limitations of the Health Department's database, some restaurants could not be placed.

By JEREMY WHITE

Source: New York City Department of Health and Mental Hygiene

© 2013 The New York Times Company

Filtering & Aggregation

[J. White, New York Times]
Assignment 2

• Link
• Due Monday, October 15
• Visualizing Hurricane Data using Tableau and Altair
• Tasks:
  - Statistics
  - Bar chart: Number of hurricanes over the years
  - Scatterplot: Windspeed versus pressure
  - Geographic Visualization: Paths of hurricanes
Map Projection

[P. Foresman, Wikimedia]
Flattening the Sphere?

- Central Meridian (selected by mapmaker)
- Great distortion at high latitudes
- Examples of two rhumb lines (direction true between any two points)
- Equator touches cylinder if cylinder is tangent
- Reasonably true shapes and distances within 15 degrees of Equator

[USGS Map Projections]
Lambert Conformal Conic Projection

Two standard parallels
(selected by mapmaker)

Large-scale map sheets can be joined at edges if they have the same standard parallels and scales
Choropleth Map: What are Marks and Channels?

[M. Ericson, New York Times]
Choropleth Map

• Data: geographic geometry data & one quantitative attribute per region
• Tasks: trends, patterns, comparisons
• How: area marks from given geometry, color hue/saturation/luminance
• Scalability: thousands of regions

• Design choices:
  - Colormap
  - Region boundaries (level of summarization)
Problem?

2008 Popular Vote

- Obama: 68 million votes
- McCain: 59 million votes
Problem?

2008 Popular Vote

- Obama: 68 million
- McCain: 59 million

Amount of red and blue shown on map

- Obama: 850,000 mi²
- McCain: 2,150,000 mi²

[M. Ericson, New York Times]
Adding Saturation

[Map showing color-coded data with various saturation levels for different regions of the United States.]

[Source: Washington Post, 2018]
Aggregation: 2016 Election by Precinct

[Interactive Version, NYTimes] [R. Rohla and Washington Post, 2018]
Aggregation: 2016 Election by State

[D. Koop, DSC 201, Fall 2018]
Aggregation: 2016 Election by Country

[Image: Map of the United States showing election results by country, with different color codes for Clinton and Trump margins.]
Size Encoding

[M. Ericson, New York Times]
Dasymetric Dot Density
Glyphs: xkcd's Map

2016 Election Map
Each figure represents 250,000 votes

Votes are distributed by state as accurately as possible while keeping national totals correct.
Location within each state is approximate.

[Image of the 2016 Election Map]
Cartograms

US Presidential Election 2016
Results mapped at county level showing the candidate with the largest vote share in each area

Overall result:
Trump
62,979,636 votes (46.1%)
306 electoral votes

Clinton
69,844,610 votes (48.2%)
232 electoral votes

Other candidates
7,804,213 votes (5.7%)

Reference map

Gridded population cartogram: areas resized according to the total number of people living there (Alaska and Hawaii not included)

Map by Benjamin Hennig
www.viewsfromtheworld.net

B. Hennig
House Races: Map?

House Race Ratings by the Cook Political Report

- 183 solid Democratic seats
- 145 solid Republican seats
- 7 vacancies

- 218 seats needed for House majority

[New York Times, 2018]
House Races: Cartogram?

District totals by category

189

[FiveThirtyEight, 2018]
# Maps Aren't Always Best: Close House Races

<table>
<thead>
<tr>
<th><strong>12 Lean Democratic</strong></th>
<th><strong>31 Tossups</strong></th>
<th><strong>25 Lean Republican</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ-02 Open (McSally)</td>
<td>CA-10 Denham</td>
<td>AR-02 Hill</td>
</tr>
<tr>
<td>CA-49 Open (Issa)</td>
<td>CA-25 Knight</td>
<td>CA-50 Hunter</td>
</tr>
<tr>
<td>CO-06 Coffman</td>
<td>CA-39 Open (Royce)</td>
<td>FL-15 Open (Ross)</td>
</tr>
<tr>
<td>IA-01 Blum</td>
<td>CA-45 Walters</td>
<td>FL-16 Buchanan</td>
</tr>
<tr>
<td>KS-03 Yoder</td>
<td>CA-48 Rohrabacher</td>
<td>GA-06 Handel</td>
</tr>
<tr>
<td>MI-11 Open (Trott)</td>
<td>FL-26 Curbelo</td>
<td>GA-07 Woodall</td>
</tr>
<tr>
<td>MN-02 Lewis</td>
<td>FL-27 Open (Ros-Lehtinen)</td>
<td>IL-13 Davis</td>
</tr>
<tr>
<td>MN-03 Paulsen</td>
<td>IL-06 Roskam</td>
<td>IL-14 Hultgren</td>
</tr>
<tr>
<td>NV-03 Open (Rosen)</td>
<td>IL-12 Bost</td>
<td>MO-02 Wagner</td>
</tr>
<tr>
<td>NJ-11 Open (Frelinghuysen)</td>
<td>IA-03 Young</td>
<td>MT-AL Gianforte</td>
</tr>
<tr>
<td>PA-07 Vacant (formerly Dent)</td>
<td>KS-02 Open (Jenkins)</td>
<td>NE-02 Bacon</td>
</tr>
<tr>
<td>VA-10 Comstock</td>
<td>KY-06 Barr</td>
<td>NY-24 Katko</td>
</tr>
<tr>
<td></td>
<td>ME-02 Poliquin</td>
<td>NY-27 Collins</td>
</tr>
<tr>
<td></td>
<td>MI-08 Bishop</td>
<td>NC-02 Holding</td>
</tr>
<tr>
<td></td>
<td>MN-01 Open (Walz)</td>
<td>OH-12 Balderson</td>
</tr>
<tr>
<td></td>
<td>MN-08 Open (Nolan)</td>
<td>PA-10 Perry</td>
</tr>
<tr>
<td></td>
<td>NJ-03 MacArthur</td>
<td>PA-16 Kelly</td>
</tr>
<tr>
<td></td>
<td>NJ-07 Lance</td>
<td>SC-01 Open (Sanford)</td>
</tr>
<tr>
<td></td>
<td>NM-02 Open (Pearce)</td>
<td>TX-23 Hurd</td>
</tr>
<tr>
<td></td>
<td>NY-19 Faso</td>
<td>TX-31 Carter</td>
</tr>
<tr>
<td></td>
<td>NY-22 Tenney</td>
<td>VA-05 Open (Garrett)</td>
</tr>
<tr>
<td></td>
<td>NC-09 Open (Pittenger)</td>
<td>WA-03 Herrera Beutler</td>
</tr>
<tr>
<td></td>
<td>NC-13 Budd</td>
<td>WA-05 McMorris Rodgers</td>
</tr>
<tr>
<td></td>
<td>OH-01 Chabot</td>
<td>WV-03 Vacant (formerly Jenkins)</td>
</tr>
<tr>
<td></td>
<td>PA-01 Fitzpatrick</td>
<td>WI-01 Open (Ryan)</td>
</tr>
<tr>
<td></td>
<td>TX-07 Culberson</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TX-32 Sessions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UT-04 Love</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VA-02 Taylor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VA-07 Brat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WA-08 Open (Reichert)</td>
<td></td>
</tr>
</tbody>
</table>

[New York Times, 2018]
Python
Tuple, List, Dictionary, or Set?

\[ [1, 2, "abc"] \]
Tuple, List, Dictionary, or Set?

{"a", 1, 2}
Tuple, List, Dictionary, or Set?

{}
Tuple, List, Dictionary, or Set?

\[(1 + (3 + 4))\]
Mutable or Immutable?

- set
- string
- tuple
- key of a dictionary item (the k's in \{k1: v1, k2: v2, ...\})
Sets

- Sets are like dictionaries but without any values:
  - \( s = \{\text{'MA}', \text{'RI}', \text{'CT}', \text{'NH'}\} \); \( t = \{\text{'MA}', \text{'NY}', \text{'NH'}\} \)
  - {} is an empty dictionary, \( \text{set}() \) is an empty set
- Adding values: \( s.\text{add('ME')} \)
- Removing values: \( s.\text{discard('CT')} \)
- Exists: "CT" in \( s \)
- Union: \( s \mid t \) => \( \{\text{'MA}', \text{'RI}', \text{'CT}', \text{'NH}', \text{'NY'}\} \)
- Intersection: \( s \& t \) => \( \{\text{'MA}', \text{'NH'}\} \)
- Exclusive-or (xor): \( s \^ t \) => \( \{\text{'RI}', \text{'CT}', \text{'NY'}\} \)
- Difference: \( s - t \) => \( \{\text{'RI}', \text{'CT'}\} \)
Nesting Containers

• Can have lists inside of lists, tuples inside of tuples, dictionaries inside of dictionaries

• Can also have dictionaries inside of lists, tuples inside of dictionaries, ...

• \[
    d = \{
        "Brady": [(2015, 4770, 36), (2014, 4109, 33)],
        "Luck": [(2015, 1881, 15), (2014, 4761, 40)],
        ...
    \}

• JavaScript Object Notation (JSON) looks very similar for literal values; Python allows variables in these types of structures
Nesting Code

• Can have loops inside of loops, if statements inside of if statements

• Careful with variable names:

  \[
  l = \{1: 3, 4: 5, 9: 12\}
  \]
  
  \[
  \text{for } i \text{ in range}(100):
  \]
  
  \[
  \quad \text{square} = i ** 2
  \]
  
  \[
  \quad \text{max_val} = l[\text{square}]
  \]
  
  \[
  \text{for } i \text{ in range}(\text{max_val}):
  \]
  
  \[
  \quad \text{print}(i)
  \]

• Strange behavior, likely unintended, but Python won't complain!
None

- Like null in other languages
- Used as a placeholder when no value exists
- The value returned from a function that doesn't return a value

```python
def f(name):
    print("Hello,", name)
    v = f("Patricia") # v will have the value None
```

- Also used when you need to create a new list or dictionary:

```python
def add_letters(s, d=None):
    if d is None:
        d = {}
    d.update(count_letters(s))
```

- Looks like `d={}` would make more sense, but that causes issues
- `None` serves as a sentinel value in `add_letters`
is and ==

- == does a normal equality comparison
- is checks to see if the object is the exact same object
- Common style to write statements like if d is None: ...

- Weird behavior:
  - a = 4 - 3
    a is 1 # True
  - a = 10 ** 3
    a is 1000 # False
  - a = 10 ** 3
    a == 1000 # True

- Python caches common integer objects

- Generally, avoid is unless writing is None
Objects

- `d = dict()` # construct an empty dictionary object
- `l = list()` # construct an empty list object
- `s = set()` # construct an empty set object
- `s = set([1,2,3,4])` # construct a set with 4 numbers

Calling methods:
- `l.append('abc')`
- `d.update({'a': 'b'})`
- `s.add(3)`

The method is tied to the object preceding the dot (e.g. `append` modifies `l` to add `'abc'`)
Python Modules

• Python module: a file containing definitions and statements

• Import statement: like Java, get a module that isn't a Python built-in
  
  ```python
  import collections
d = collections.defaultdict(list)
d[3].append(1)
  ```

• Import ... as <shorter-name>

  ```python
  import collections as c
  ```

• From...import...: don't need to refer to the module

  ```python
  from collections import defaultdict
d = defaultdict(list)
d[3].append(1)
  ```
Other Collections

- `collections.defaultdict`: specify a default value for any item in the dictionary (instead of `KeyError`)
- `collections.OrderedDict`: keep entries ordered according to when the key was inserted
  - `dict` objects are ordered in Python 3.7 but `OrderedDict` has some other features (equality comparison, reversed)
- `collections.Counter`: counts hashable objects, has a `most_common` method
Iterators

• Remember `range, values, keys, items`?
• They return **iterators**: objects that traverse containers, only need to know how to get the next element
• Given iterator `it, next(it)` gives the next element
• `StopIteration` exception if there isn't another element
• Generally, we don't worry about this as the for loop handles everything automatically
• …but you cannot index or slice an iterator
• `d.values()[0]` will not work!
• If you need to index or slice, construct a list from an iterator
• `list(d.values())[0]` or `list(range(100))[-1]`
• In general, this is slower code so we try to avoid creating lists
List Comprehensions

- Shorthand for transformative or filtering for loops

  ```python
  squares = []
  for i in range(10):
      squares.append(i**2)
  squares = [i**2 for i in range(10)]
  ```

- Equivalent code, just moved the loop inside of list definition

- Advantages: concise, readable

- Filtering:

  ```python
  squares = []
  for i in range(10):
      if i % 3 != 1:
          squares.append(i ** 2)
  squares = [i**2 for i in range(10) if i % 3 != 1]
  ```

- If clause **follows** the for clause
Dictionary Comprehensions

• Similar idea, but allow dictionary construction
• Could use lists:
  - names = dict([(k, v) for k,v in ... if ...])
• Native comprehension:
  - names = {"Al": ["Smith", "Brown"], "Beth": ["Jones"]}
  first_counts = {k: len(v) for k,v in names.items()}
• Could do this with a for loop as well
Exceptions

• errors but potentially something that can be addressed

• try-except-else-finally:
  - except clause runs if exactly the error(s) you wish to address happen
  - else clause will run if no exceptions are encountered
  - finally: always runs (even if the program is about to crash)

• Can have multiple except clauses

• can also raise exceptions using the raise keyword

• (and define your own)