

---

# CS 111: Program Design I

Lecture 25: Predictive policing concluded;  
plotting, heatmaps!

---

Robert H. Sloan & Richard Warner

University of Illinois at Chicago

November 26, 2019



---

# **PREDICTIVE POLICING (CONTINUED)**

# Co-Arrests

- Generally: co-arrested = arrested together
- Chicago—Two types:
  - X and Y arrested together, Y is murdered later.
    - “A first degree link refers to a relationship between a subject and an individual with whom the subject was previously co-arrested who later became a homicide victim.”
  - “X and Y arrested together, and Y later arrested with the murder victim Z.
    - A second degree link refers to a relationship in which a subject was co-arrested with another person who, in turn, was co-arrested with a later homicide victim.”

# The Underlying Theory

- The more connections you have to co-arrested individuals the more likely you will commit a violent crime or be a victim of one.
- “A series of research studies found that gun violence—just like an infectious disease—can be transmitted from person to person in social networks:
  - i.e., exposure to gun violence not only can lead to a host of negative psychological and cognitive outcomes but also increases the risk of individuals becoming gunshot victims themselves.
  - Furthermore, individuals who associate with a greater number of gunshot victims are at an extremely elevated risk of being victims themselves.”
    - Papachristos and Michael Sierra-Arévalo, *Policing the Connected World*

# Just Like Infectious Disease?

- For infectious disease, we have a well developed theory of causal transmission.
- Do we have the same for the “infectious” transmission of crime?
  - No. The claim is that crime spreads over a network *in a way similar to* the way that infectious diseases spread over a network.
  - There is little explanation of *why* crime spreads that way.

# What You Get

- “These studies suggest that the diffusion of gun violence might occur through person-to-person interactions, in a process akin to the epidemiological transmission of a blood-borne pathogen (eg, HIV). Contagion via social ties, then, may be a critical mechanism in explaining why neighborhoods matter when modeling the diffusion of crime and, perhaps more important, why certain individuals become subjects of gun violence while others exposed to the same high-risk environments do not.”
  - Green, Horel, Papachristos, “Modeling Contagion Through Social Networks to Explain and Predict Gunshot Violence in Chicago, 2006 to 2014”

---

# Chicago's Predictive Success

- The Strategic Subject List algorithm “identified less than 1% of homicide victims (3 out of 405).”
- Later versions of the software improve its accuracy, but no numbers are available.

# Nonetheless . . .

- “A component of the IIT/CPD predictive policing is an algorithm that automatically determines the probability that a prior offender will be involved in a future shooting or homicide in an upcoming period. ***In CPD’s Custom Notifications program, persons having the highest probability of future violence are personally visited by a police commander, a social worker, and a person from the community (such as a person who lost a loved one to violence)***, to illustrate to the at-risk individual the impact of violence on the community.”
  - Miles Wernick, Case Study: Smart Policing, 2016.



---

# And . . .

- “Social network analysis has become extremely popular and important in recent years. Monitoring real-time updates on selected Facebook, Twitter, and other social media can provide law enforcement officials with immediate information on crimes.”
  - Walter Perry et. al., *Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations*

---

# The Need to Limit State Power

- The state has a monopoly on the legitimate use of force.
- Power corrupts.
- So, traditionally, the ability of the state to examine your life has been limited—limited by law *and by technological limitations*.
- The technological limitations have changed.
- How should we respond?

# A Hurdle

- When “it comes to policing, the ordinary rules of democratic governance seem to evaporate. Policing officials decide for themselves how to enforce the law. The rules . . . often are not public. Even more rarely are they adopted with public input . . . we try to fix things after the fact, after they go wrong: with civilian review boards, inspectors general, and especially with review by the courts”
  - Friedman, *Unwarranted: Policing Without Permission*

---

# Why This Matters

- Privacy is an essential protection against the government's abuse of power.
- And—related and also more generally—*privacy in public* matters greatly in a world in which others obtain and use vast amounts of information about us.
- We need to be able to trust that others use information about us only in appropriate, selective ways.

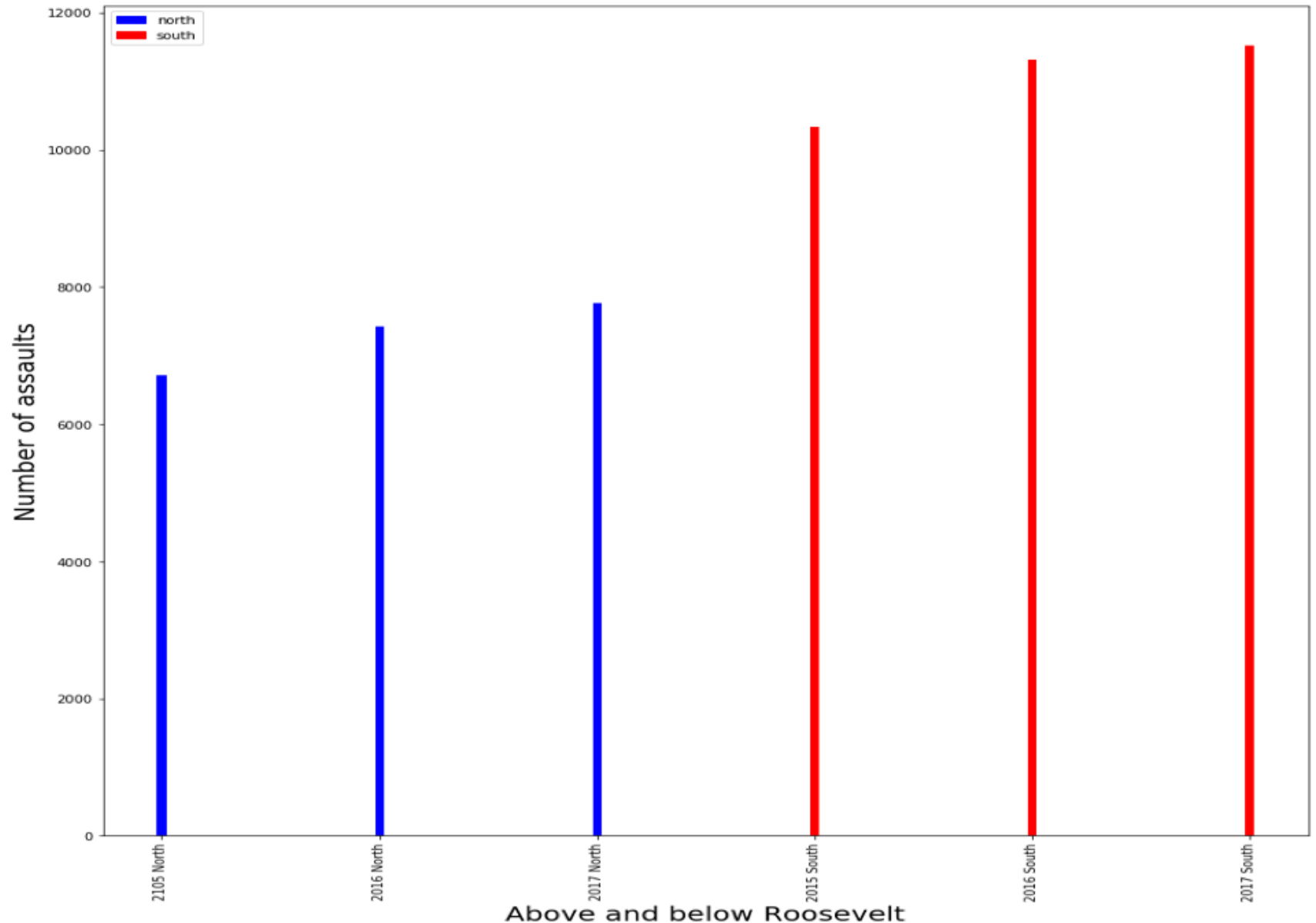
# From The Crimes Database You Are Using

An assault is an intentional act creating an apprehension imminent harmful or offensive contact.

Where would send more police?

North of Roosevelt, or South of Roosevelt?

What assumptions underlie the graph?



# Crime Is Predictable

- “There is a strong body of evidence to support the theory that crime is predictable (in the statistical sense)—mainly because criminals tend to operate in their comfort zone. That is, they tend to commit the type of crimes that they have committed successfully in the past, generally close to the same time and location.”
  - Perry et al., *Predictive Policing*.

# PredPol

- “Using only three data points – crime type, crime location and crime date/time – PredPol’s powerful software provides each law enforcement agency with customized crime predictions for the places and times that crimes are most likely to occur. PredPol pinpoints small areas, depicted in 500 feet by 500 feet boxes on maps – that are automatically generated for each shift of each day. The algorithms used by PredPol . . . are based on the observation that certain crime types tend to cluster in time and space.”
  - <http://www.predpol.com/how-predpol-works/>

# Types of Predictions

- Predict crimes: predict *places and times* with an increased risk of crime. This is PrePol.
- Predict offenders: predict *people* at high risk for committing a *future crime*.
- Predict victims: identify *groups and individuals likely to become victims*.
- Predict perpetrators: Match *people* with *past crimes*.



# Errors and the Myth of the All-Seeing Eye

- An early American Calvinist meeting house painted the all-seeing eye of God on the front of its pulpit. It was a reminder of God's surveillance powers:
  - “God is the witness of [a person's] inmost self and the sure observer of his heart and the listener to his tongue. For the spirit of the LORD fills the world, is all-embracing, and knows what man says.” *Book of Wisdom*.
- Predictive policing is *not* an all-seeing eye.

---

# Two Reasons

- Its predictions are *statistical generalizations*.
  - “Although much news coverage promotes the meme that predictive policing is a crystal ball, these algorithms predict the risk of future events, not the events themselves.” Perry et al., *Predictive Policing*.
- Human judgement—with all of its potential biases—is essential in the process of prediction.

---

# Roles of Human Judgment

“Even with the most complete software suites, humans must **[1]** find and collect relevant data, **[2]** preprocess the data so they are suitable for analysis, **[3]** design and conduct analyses in response to ever-changing crime conditions, **[4]** review and interpret the results of these analyses and exclude erroneous findings, **[5]** analyze the integrated findings and make recommendations about how to act on them, **[6]** and take action to exploit the findings and **[7]** assess the impact of those actions.”

Perry et al., *Predictive Policing*.

# Responding to Modern Crime

- Modern criminals often use sophisticated computer techniques to commit tax fraud, launder money, and commit other crimes. Detecting and prosecuting these crimes can require equally sophisticated techniques. Police should have those techniques?
  - A. Yes
  - B. No

# Power Struggles

Technology	Power struggle	Law
Encryption	Individuals versus businesses and governments	4 <sup>th</sup> and 5 <sup>th</sup> Amendments (government only)
	Businesses and governments versus hackers	Contractual prohibitions against encryption
Crawler/scrapper	Businesses versus businesses	Computer Fraud and Abuse Act
	Governments versus hackers	Contractual prohibitions against crawlers/scrappers
Data analysis	Individuals versus businesses	Computer Fraud and Abuse Act
	Businesses versus businesses	Copyright
	Governments versus hackers	Contractual prohibitions
		Notably absent: substantive, relevant privacy law



# **SOME COURSE LOGISTICS**

# Announcement: Do the course evaluation

- We'll make it worth your while
- Introducing two possible course themes
  - Privacy
  - Collective action problem
- If at least 70% of class completes course evaluation by deadline, will add 1 point extra credit to *all*
  - *We never find out who did/didn't complete it*
  - In past, profs get early warning if response rate *very* low (< 40%?) and final response rate *after responses closed*

# Announcement: ~~Winter~~ Final Exam is Coming

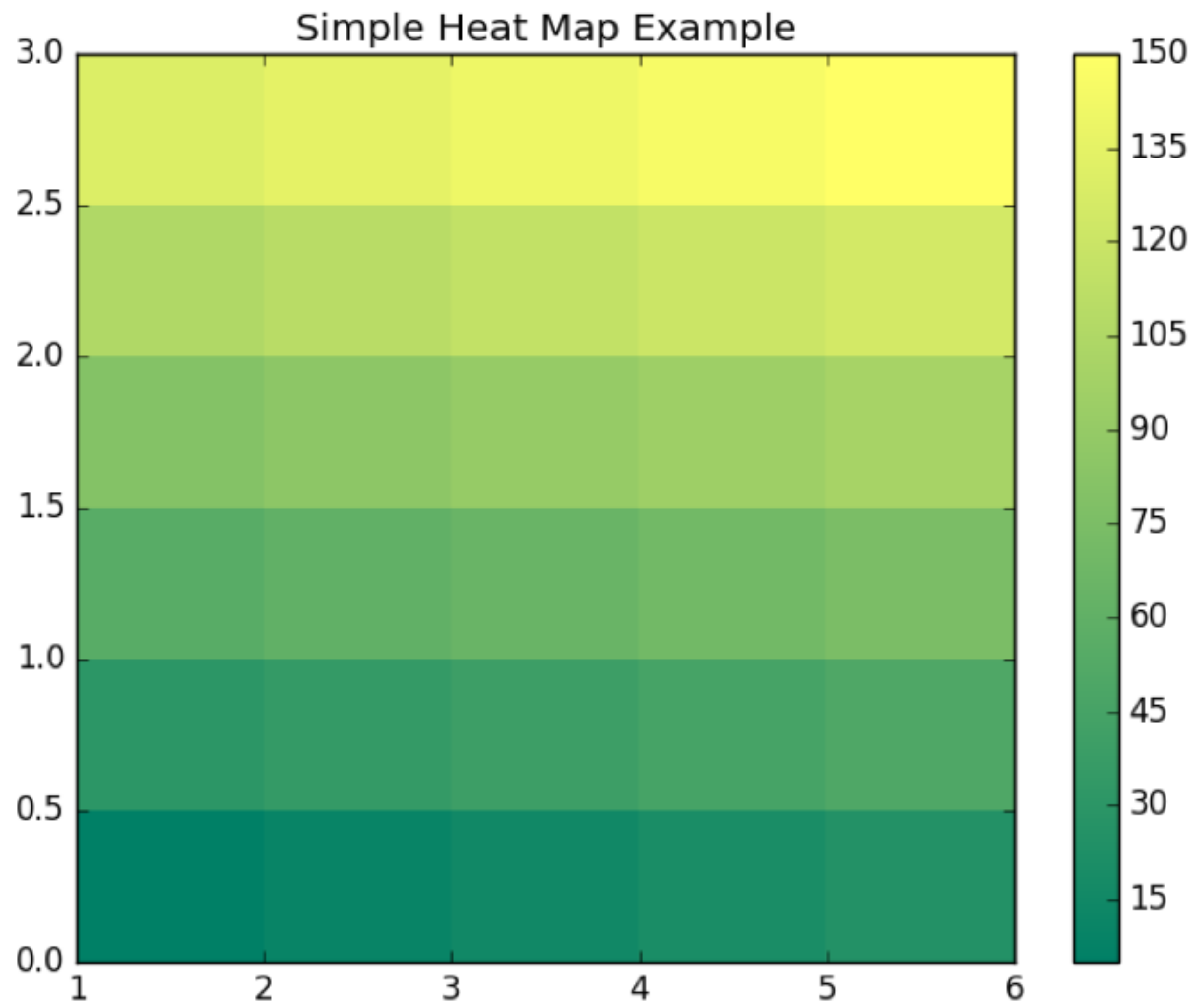
- Most of you are doing really well
- A small handful of you are in trouble with requirement: *You must pass the exam part of the course to pass the exam*
  - Meaning average of 2 midterms and final exam (weighted equal to both midterms taken together)
- For all: Will use  $\geq 67$  final exam to replace any lower midterm exams
  - But two midterms in the 30s and final of 50 will fail this course even with good labs and projects



---

# **HEAT MAPS, CHOROPLETH MAPS**





```
import matplotlib.pyplot as plt

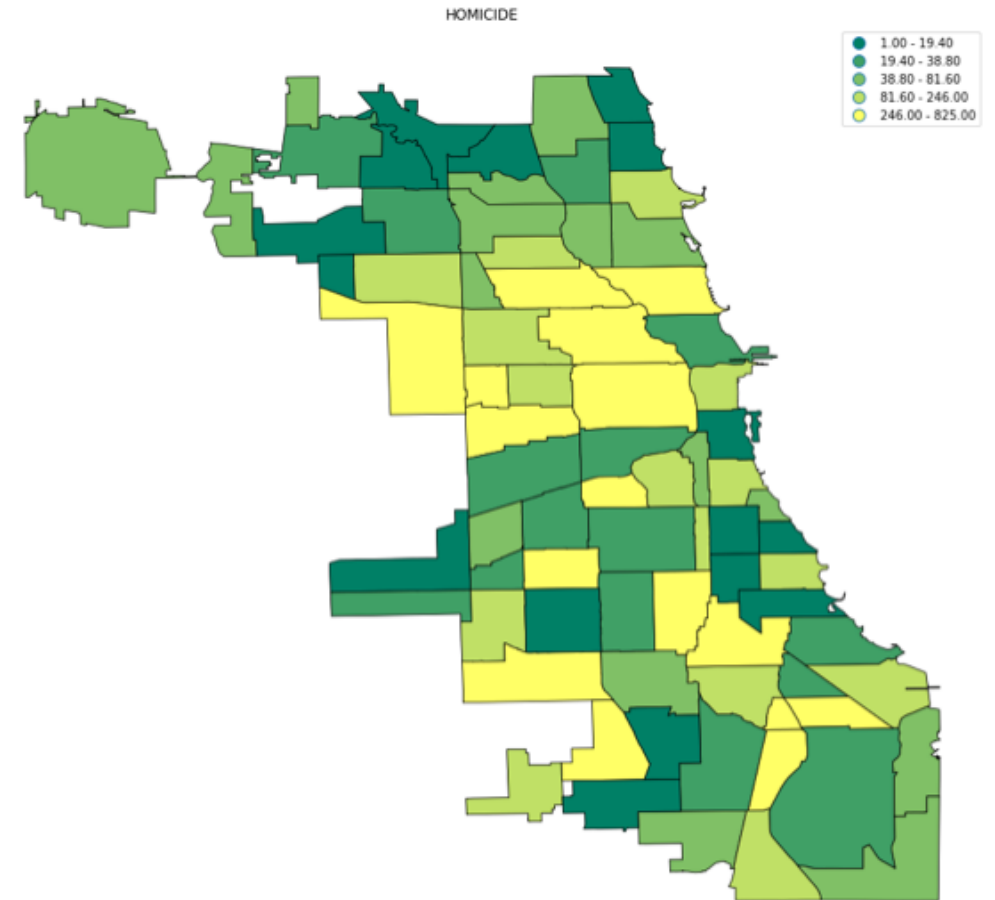
x = [1, 2, 3, 4, 5, 6]
y = [0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0]
intensity = [
    [5, 10, 15, 20, 25],
    [30, 35, 40, 45, 50],
    [55, 60, 65, 70, 75],
    [80, 85, 90, 95, 100],
    [105, 110, 115, 120, 125],
    [130, 135, 140, 145, 150]
]
plt.pcolor(x, y, intensity) # Creates heatmap, origin lower left corner
plt.summer() # Color scheme: Green (low) to Yellow (hi)
plt.colorbar() # Adds colorbar so we know what values mean
plt.title("Simple Heat Map Example")
```

# Heat maps in Python

- Everything we need is in `matplotlib.pyplot`, so
  - `import matplotlib.pyplot as plt`
- `plt.pcolor(C)` creates heat map from array `C` of color values
  - Row 0 displayed at *bottom* of figure
  - Each row displayed left to right
  - So `C[0][0]` in lower left corner
- `plt.pcolor(X, Y, C)` has lists of numbers for X- and Y-axes

# Heat map → Choropleth map

**Choropleth** (map) (from Greek *χῶρος* "area/region" and *πλῆθος* "multitude"): thematic map where areas colored/shaded proportionally to measurement of statistical variable displayed on map. Technically distinct from heat maps (not on maps!)



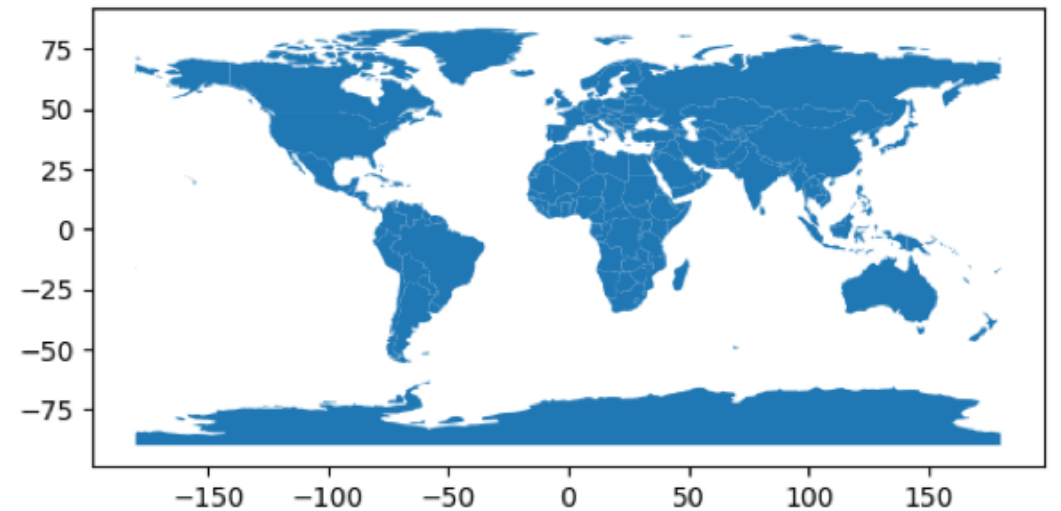
# Making Choropleth Maps in Python

- Use **geopandas** module
- Just like pandas but using geodata from “shape files”
  - Lots can be downloaded from web; we’ll use Chicago communities
  - Also some provided with geopandas; e.g., after `import geopandas as gpd`, `gpd.datasets.get_path('naturalearth_lowres')`
  - gives world map

# Example

```
import geopandas as gpd
import matplotlib
```

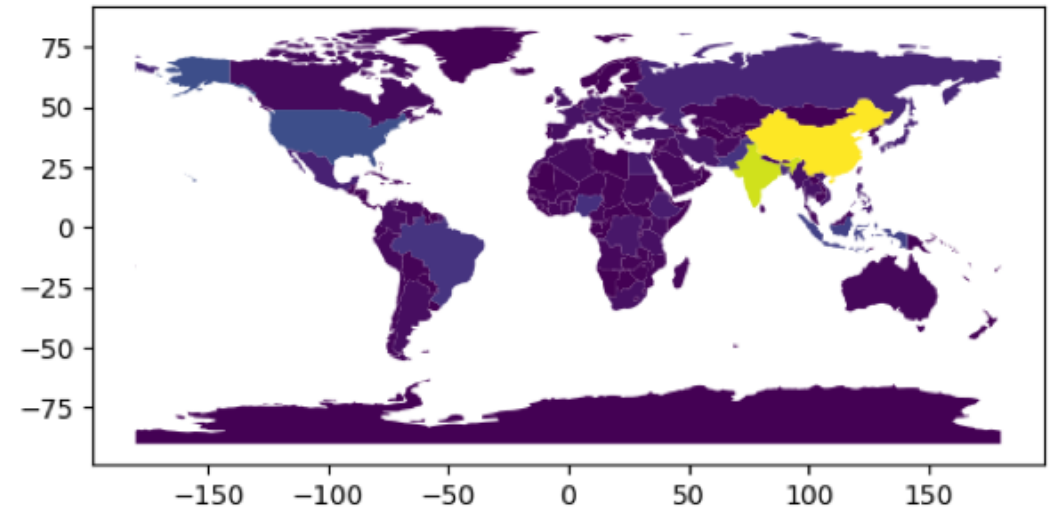
```
world_df = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
# Check data so we can see we have GEOdataframe
type(world_df)
world_df.head()
world_df.plot() # just shape, no data
```



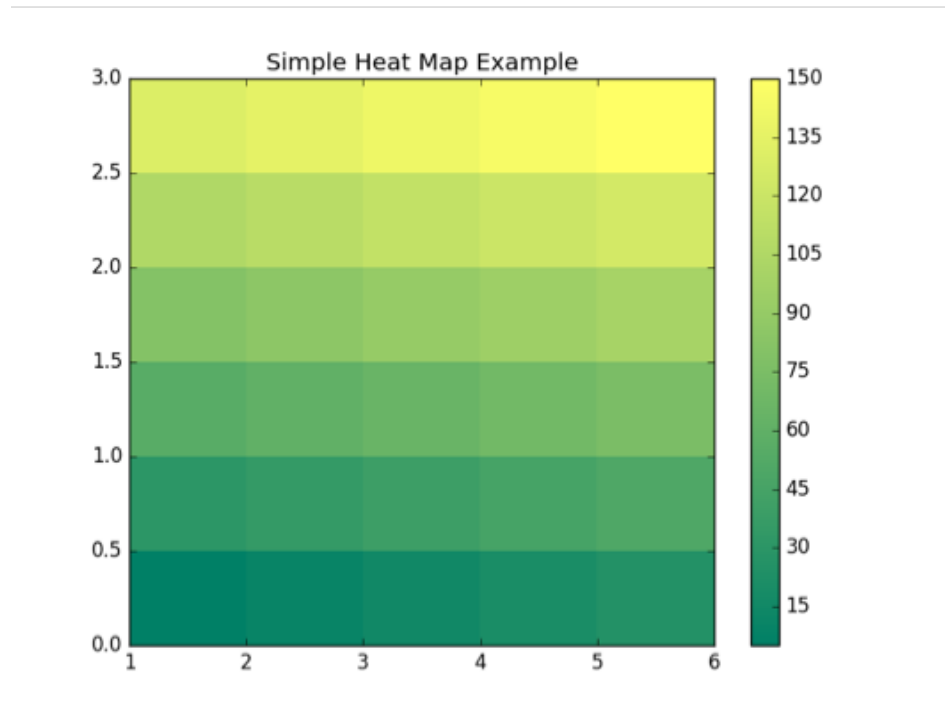


# From plotting (empty) map to choropleth

- Geodataframe must have column of data we want to display on map
- Creating column of crime rate main part of assignment
- world example come with one column of data we can plot: 'pop\_est'. (estimated population)
- `world_df.plot(column='pop_est')`



# What do the colors mean? Colorbar in heatmap; legend=True in chloropleth



- `plt.colorbar()`
  - places colorbar at right of plot
  - Gives key for what colors correspond to what matrix values
- Same thing from optional argument `legend=True` inside  
`geo_df.plot(column='...', legend=True)`



# To set the colors (“colormap”)

```
plt.viridis()# viridis colormap
```

```
plt.summer() # summer colormap
```

- etc.

- Use matplotlib colormap from the Perceptually Uniform Sequential\*, Sequential, or Sequential(2) group

\*The favorite of the politically correct visualization police

# Reasonable Colormap choices

From

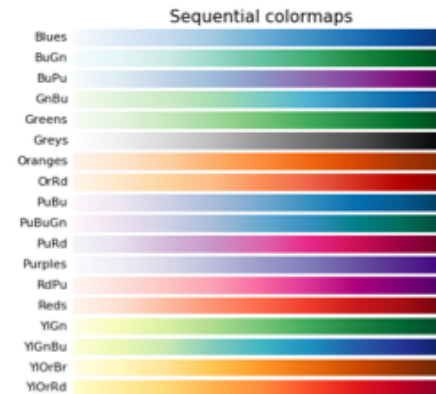
[http://matplotlib.org/examples/color/colormaps\\_reference.html](http://matplotlib.org/examples/color/colormaps_reference.html)

## color example code: colormaps\_reference.py

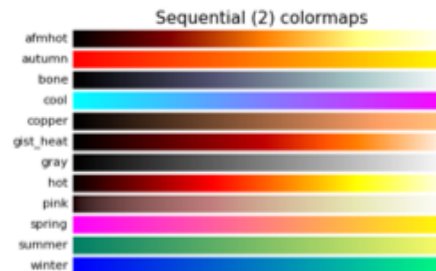
(Source code)



(png, hires.png, pdf)



(png, hires.png, pdf)



---

Drawing graphs

# **MATPLOTLIB MODULE HIGHLIGHTS**

## A Picture is Worth 1000 Excel cells

Year, Annual anomaly, Lower 95% confidence interval, Upper 95% confidence interval

1880, -0.4700088, -0.672646261, -0.267371339

1881, -0.3568788, -0.560588343, -0.153169257

1882, -0.3726612, -0.575728173, -0.169594227

1883, -0.448443, -0.650803864, -0.246082136

1884, -0.5897538, -0.790478088, -0.389029512

1885, -0.6636546, -0.86307244, -0.46423676

1886, -0.6439392, -0.842606641, -0.445271759

1887, -0.7616232, -0.959851596, -0.563394804

1888, -0.5166342, -0.713950039, -0.319318361

1889, -0.4717926, -0.674798269, -0.268786931

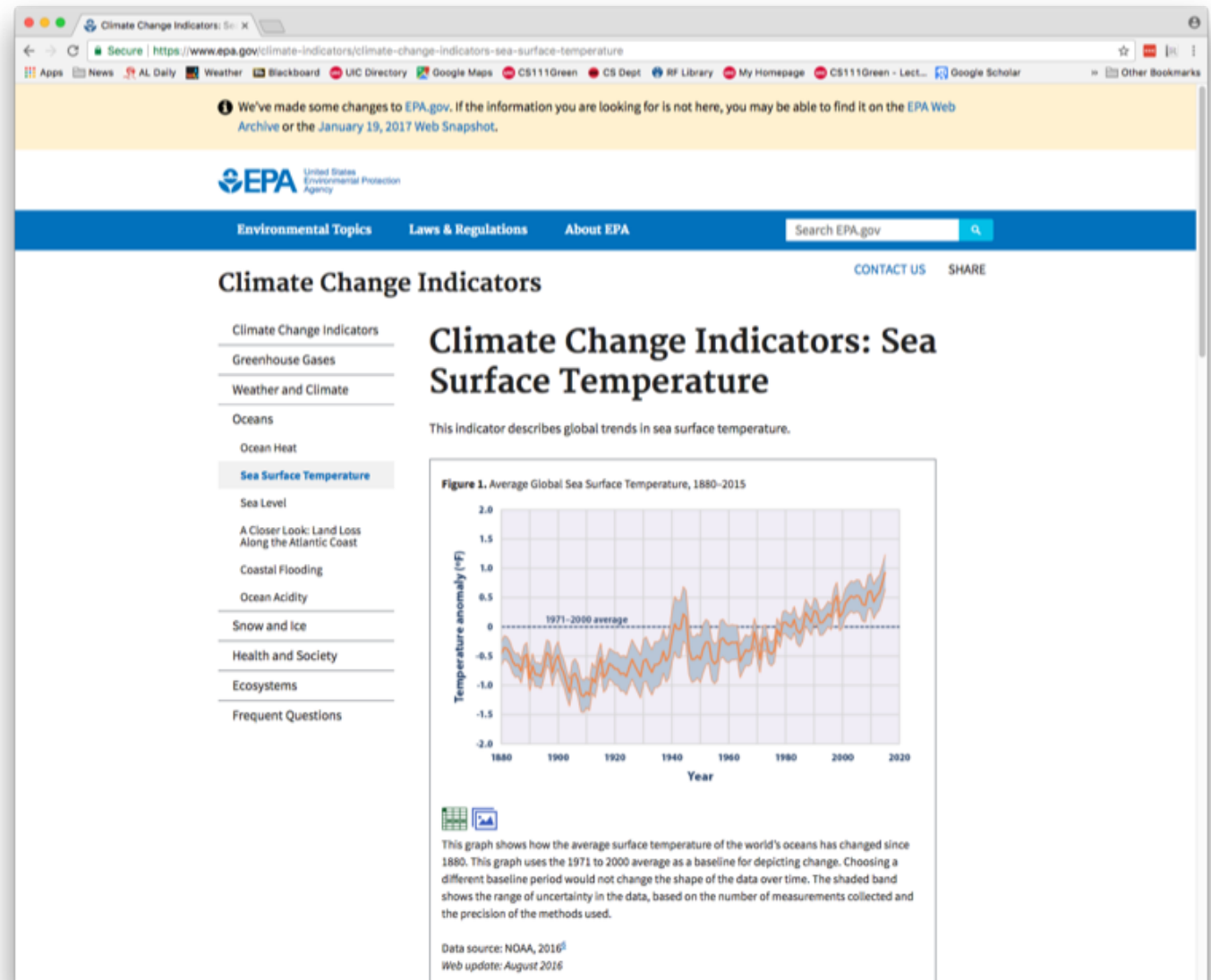
1890, -0.8875836, -1.093269328, -0.681897872

1891, -0.6603264, -0.864035943, -0.456616857

1892, -0.8173098, -1.021385617, -0.613233983

1893, -0.8148276, -1.019876951, -0.609778249

*And so on*



<https://www.epa.gov/climate-indicators/climate-change-indicators-sea-surface-temperature>

---

# matplotlib: A module for drawing graphs

```
import matplotlib.pyplot as plt
```

- matplotlib is super commonly used module for 2-D graphics in Python.
  - There are others, but matplotlib by far most widely used
  - Style taken from MATLAB



# Functions in plt that might be nice to use: plot

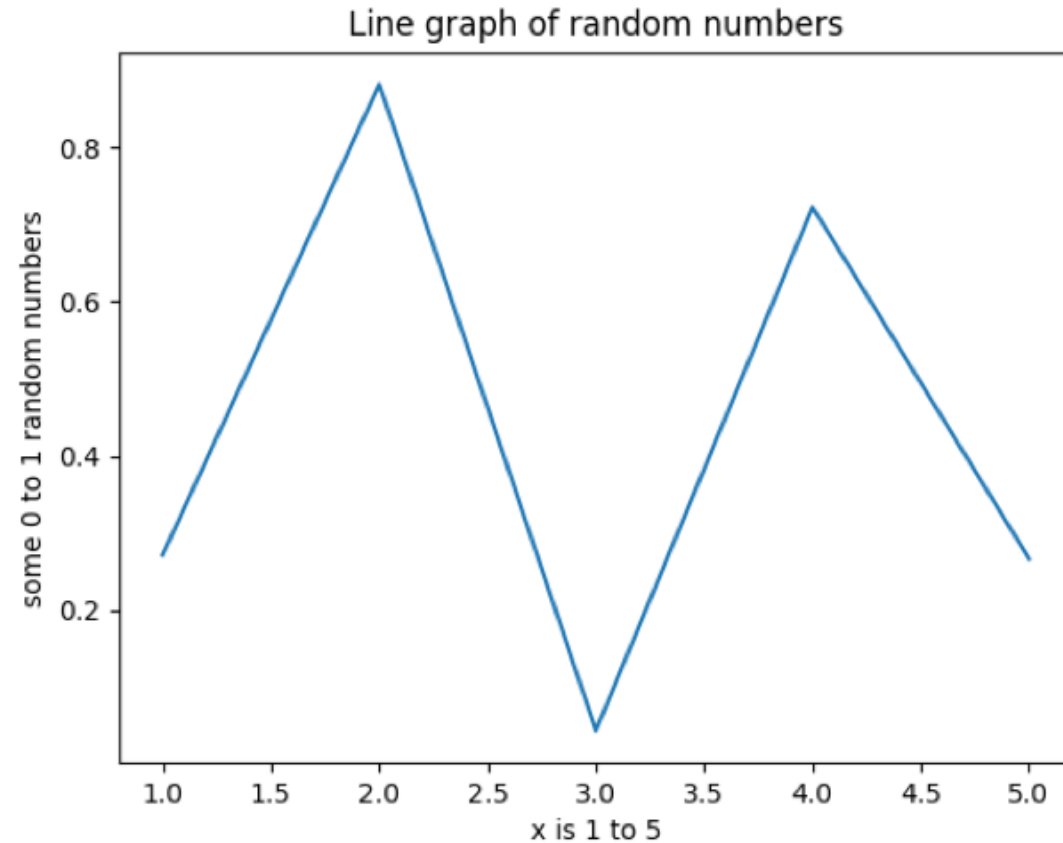
- `plt.plot(ls)` with one list input parameter: Makes a line graph assuming x's are `range(len(ls))`, i.e., 0, 1, ..., `len(ls) - 1`
- With 2: x vs. y
- Already seen: Can set x and/or y-axis label, and the title
- Will need title again for last project
  
- **Important after CS 111! Exact details of *when* and *where* plot appears *depend heavily on whether you are using Spyder, console, Jupyter, or other, and on your settings***
- (Spyder things generally just appear)

## Basic demo code

```
import matplotlib.pyplot as plt
import random

# simple plotting demo of plain line graph

x = [1, 2, 3, 4, 5]
yline = [] # y values will go here
for i in range(len(x)):
    yline.append(random.random())
plt.ylabel('some 0 to 1 random numbers')
plt.xlabel('x is 1 to 5')
plt.title('Line graph of random numbers')
plt.plot(x, yline)
```



---

# You probably want plots in their own window

- Anybody having plots show up inside the lower-right console window and having trouble saving them out?

# To always get graphs in their own window

- Spyder preferences (under Python menu on Mac, on Windows maybe under Tools?)
  - Then: IPython Console → Graphics → Graphics Backend → Backend: "automatic"
- *Requires you to restart Spyder (once) to start working*
  - In theory there's also command can give at iPython prompt for this, but Prof. Sloan couldn't figure it out, and it's *definitely not* what's stated in official Spyder tutorial (`%matplotlib qt`), which gives error
- And `plt.figure(1)`, `plt.figure(2)`, etc. starts new plots
  - Until then, `plt.whatever()` keeps adding to current plot

# More specific styling things

- Described in Zybooks 15.2 and 15.3, like adding legends to describe what different lines on multi-line line graph are, making lines different colors and styles (dotted, dashed, solid) that you choose instead of matplotlib choosing automagically
  - Nobody in their right mind memorizes this stuff unless working on graphs as full-time job; we wouldn't ask exam questions about it
  - But do need to know where to find it if lab/project asks you to plot something with a green dashed line

---

# A few examples

- If last argument to plot is string, that's the format
- 'b-'      Solid blue line; matplotlib default
- 'r--'     Red dashes
- 'bs'      Blue squares
- 'g^'      Green triangles