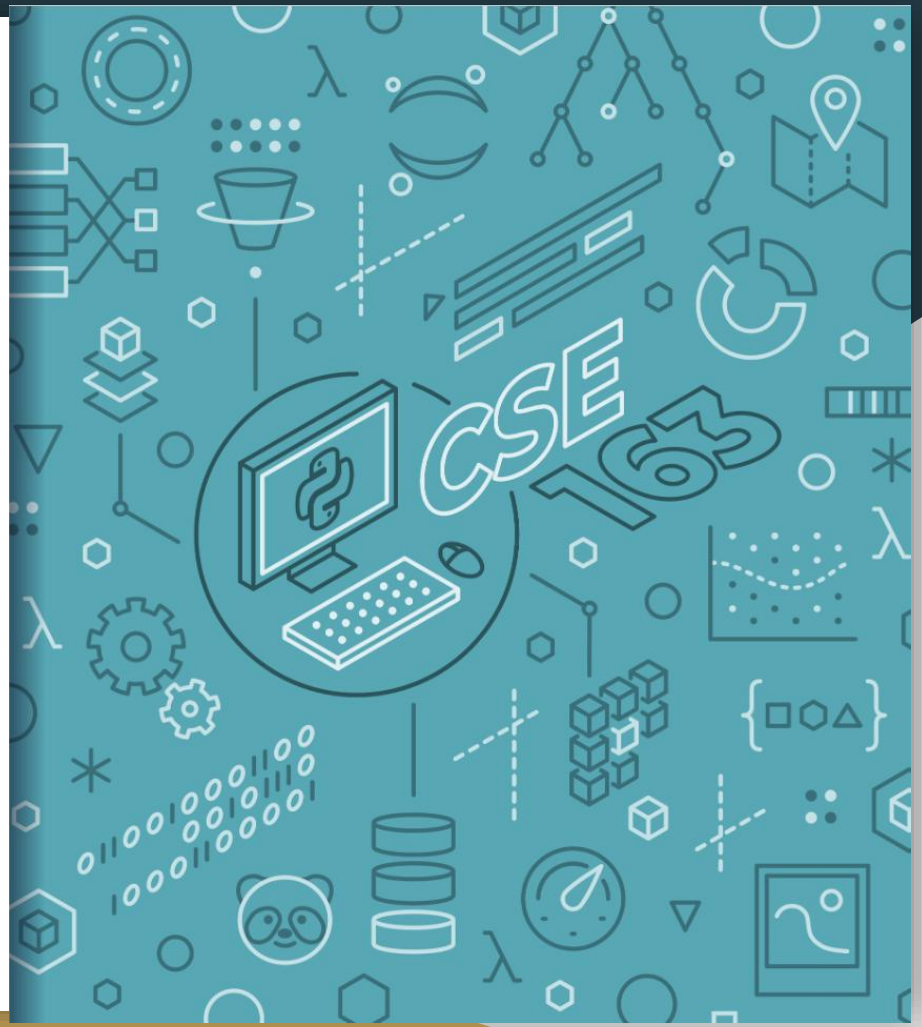
CSE 163
Dissolve & Joins

Adrian Salguero
Spring 2026

💭 Icebreaker (discuss with neighbors): ‘
What is your favorite snack?
Add to our Slido!

slido.com
#cse163



Announcements

- **Take Home Assessment 5: Mapping** released on Tuesday
- **Reading Assignment 4** due tonight at 11:59pm on Canvas!
- **Project Part 2** due May 14th at 11:59pm!
  - EDA/Milestone
  - Group Projects: only one person needs to submit but add your teammates to your submission using these instructions
- **No Lesson Quiz for Today!**

Geopandas - pandas with geo data

- GeoDataFrame and GeoSeries

```
import geopandas as gpd

df = gpd.read_file('data_file.shp')

df.plot(column='some_col', legend=True)
plt.savefig('plot.png')
```

Matplotlib

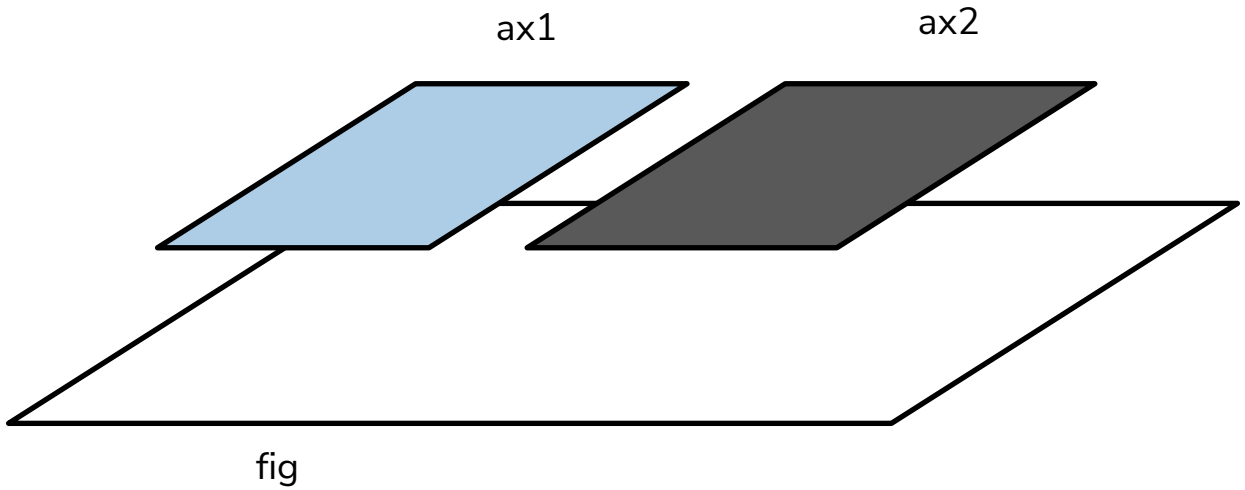
- Default plots gets plotted on a global figure (think like a painting canvas)
  - *Plots would overwrite each other*
  - *Extra work to plot multiple graph information on the same set of axes*

```
fig, ax = plt.subplots(1)
<plot1>(ax=ax)
<plot2>(ax=ax)
fig.savefig('plot_together.png')
```

```
fig, [ax1, ax2] = plt.subplots(1, 2) # width, height
<plot1>(ax=ax1)
<plot2>(ax=ax2)
fig.savefig('plot_separate.png')
```

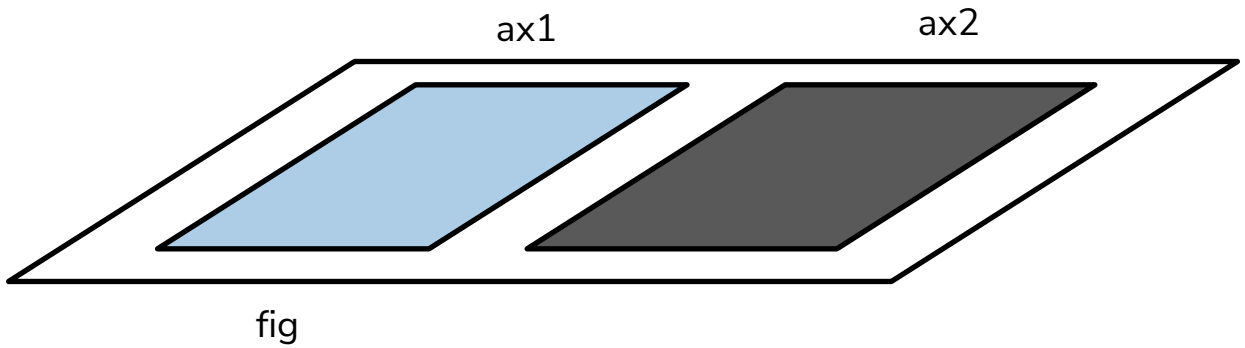
Matplotlib

- Default plots gets plotted on a global figure (think like a painting canvas)
  - *Plots would overwrite each other*
  - *Extra work to plot multiple graph information on the same set of axes*



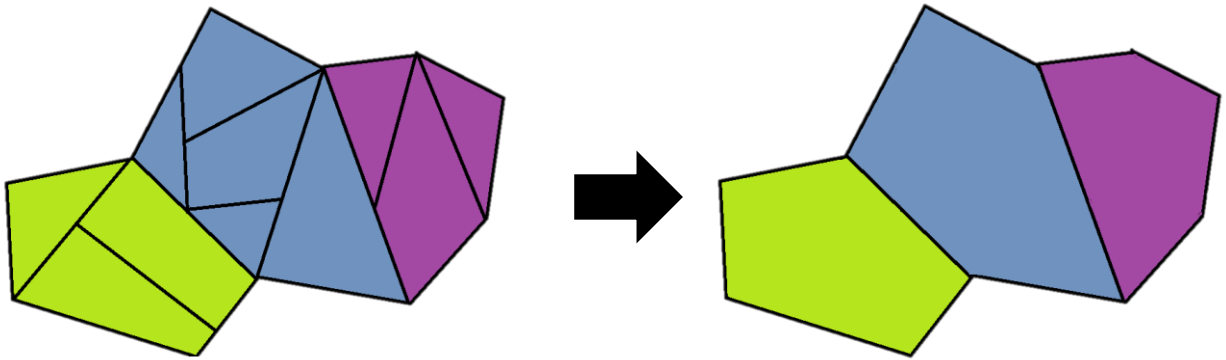
Matplotlib

- Default plots gets plotted on a global figure (think like a painting canvas)
  - *Plots would overwrite each other*
  - *Extra work to plot multiple graph information on the same set of axes*



Dissolve

- **Exactly the same as a *groupby* for the regular columns**
  - *For the geometry columns, overlays all of the geometries*
- Options for the *aggfunc*
  - *‘first’, ‘last’, ‘min’, ‘max’, ‘sum’, ‘mean’, ‘median’*



Join

```
tas.merge(grading, left_on='ta_id',
          right_on='grader_id')
```

ta_name	ta_id
Ryan	1
James	2
Nicole	3

grader_id	student_name
2	Flora
3	Paul
1	Wen
3	Andrew

ta_name	ta_id	grader_id	student_name
Ryan	1	1	Wen
James	2	2	Flora
Nicole	3	3	Paul
Nicole	3	3	Andrew

Types of Join: what if rows don't line up?

```
left.merge(right, left_on = 'lcol', right_on = 'rcol', how='type'
```

- ***Inner (default)***: Both values must be present
- ***Left***: If a value from the *left* has no match, add *NaNs*
- ***Right***: If a value from the *right* has no match, add *NaNs*
- ***Outer***: If a value from *either* table has no match, add *NaNs*

Hurricane Florence

In [9]:
```
affected_states = gpd.sjoin(country, florence, how='inner', op='intersects')


fig, ax = plt.subplots(1, figsize=(20, 10))

country.plot(ax=ax, color='#EEEEEE', edgecolor='#FFFFFF')
affected_states.plot(ax=ax, edgecolor='#FFFFFF')
florence.plot(ax=ax, color='#000000', markersize=10)
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f071b8466a0>
```