



# *Collidea*

---

*Construction site collision research*

# BACKGROUND

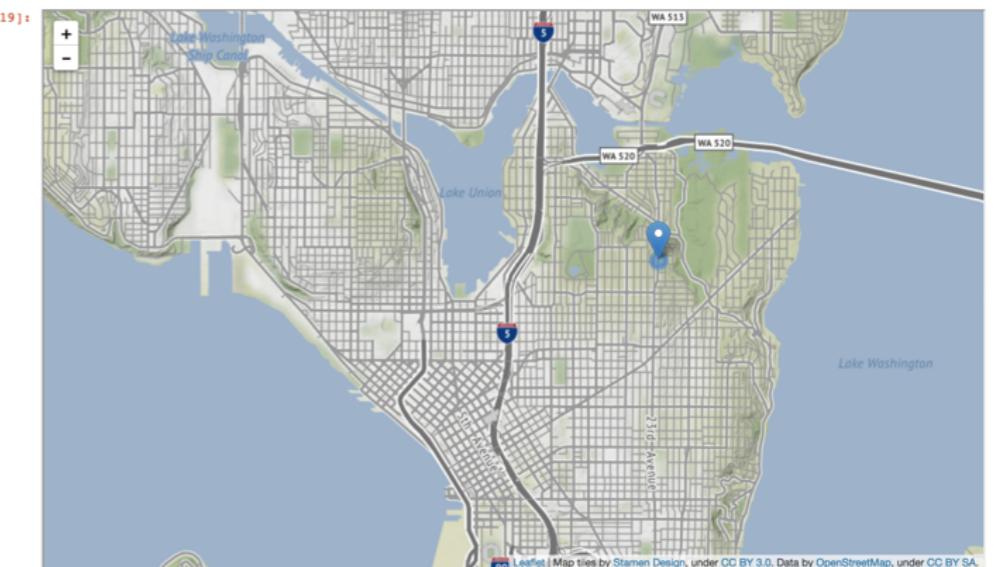
---

- Combine two datasets from Seattle Open Data portal
  - **Building Permits**
  - **Collisions**
- Map and compare collisions surrounding building permit sites before, during, and after construction
- Potential research questions and use cases include:
  - Collision frequency
  - Collision types (pedestrian, cyclists, vehicles)
  - Building permit characteristics

# FINAL PRODUCT

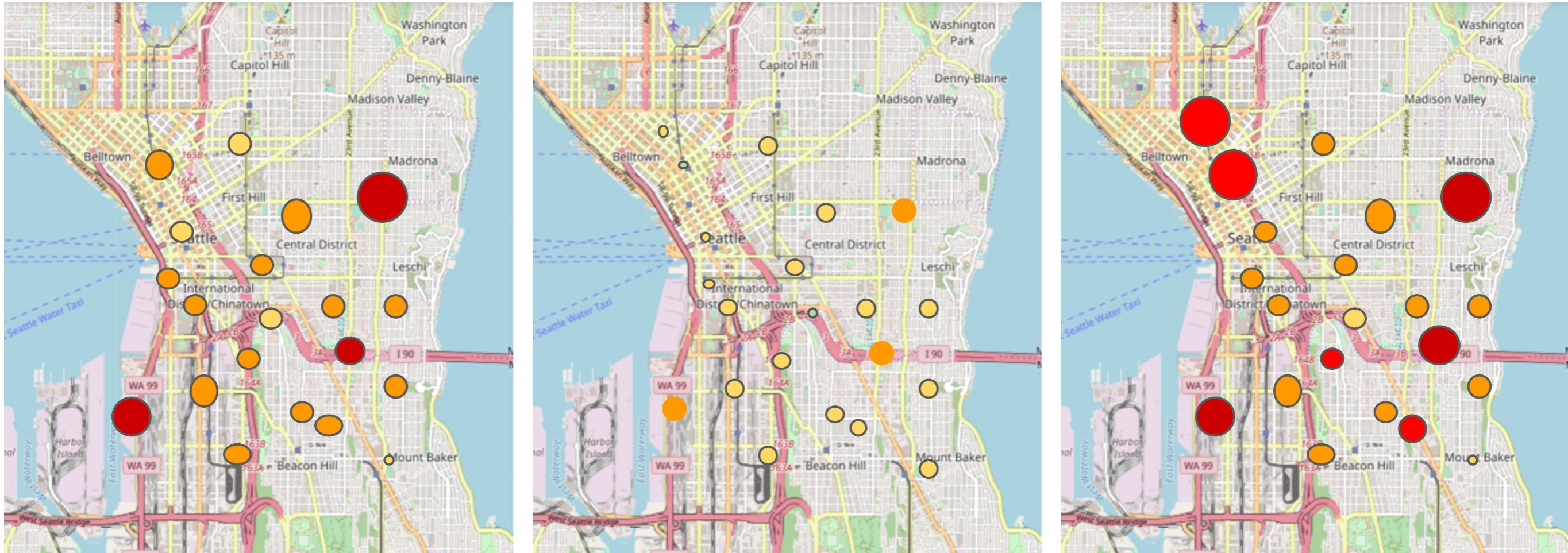
---

- Link collisions and buildings based on location
- Map locations of building permits
- Use different sized circles to represent number of collisions
- Maps displayed within Jupyter notebook
- Interactive widgets to filter data and then update dynamic map



# MOCK-UP OF OUR FINAL PRODUCT

---



Location of point indicates location of building site

Color and size of point indicates volume of nearby collisions in the specified time period (before/during/after construction)

User can interact to select options such as collisions involving pedestrians, type of building

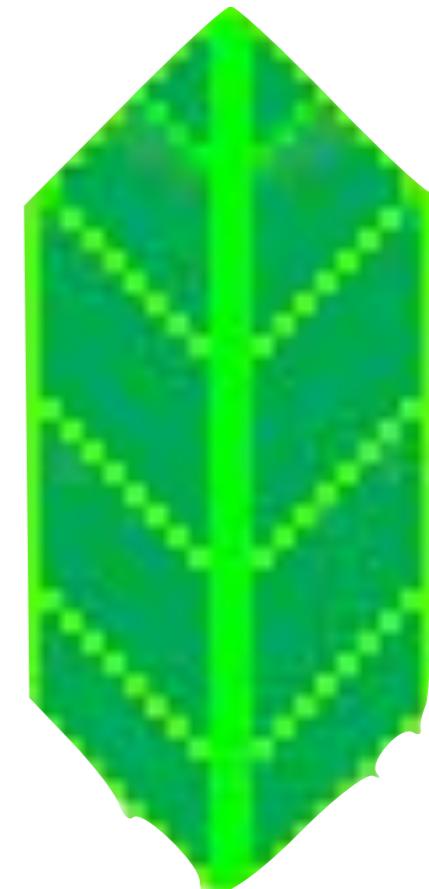
# FOLIUM - HOW IT WORKS

---

- Folium uses Leaflet.js library to generate interactive maps in Python
- Leaflet creates interactive maps with Javascript -- widely used by Facebook, Washington Post, Craigslist etc.
- Folium displays Leaflet maps in Python, can embed directly in Jupyter notebook

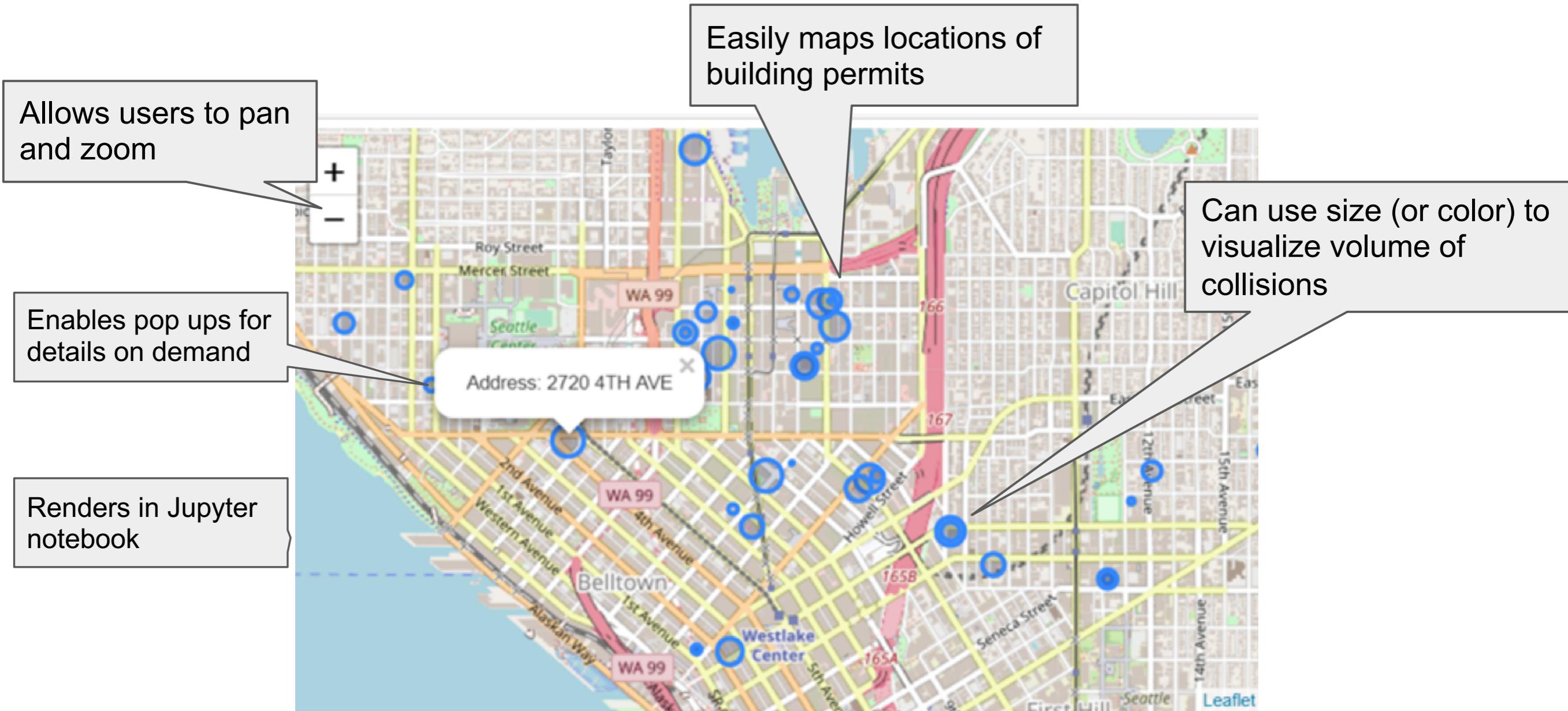


an open-source JavaScript library  
for mobile-friendly interactive maps



# PROS: IT HAS THE FUNCTIONALITY WE NEED

---



# PROS: IT IS INTUITIVE AND EASY TO LEARN AND INTERPRET

.....

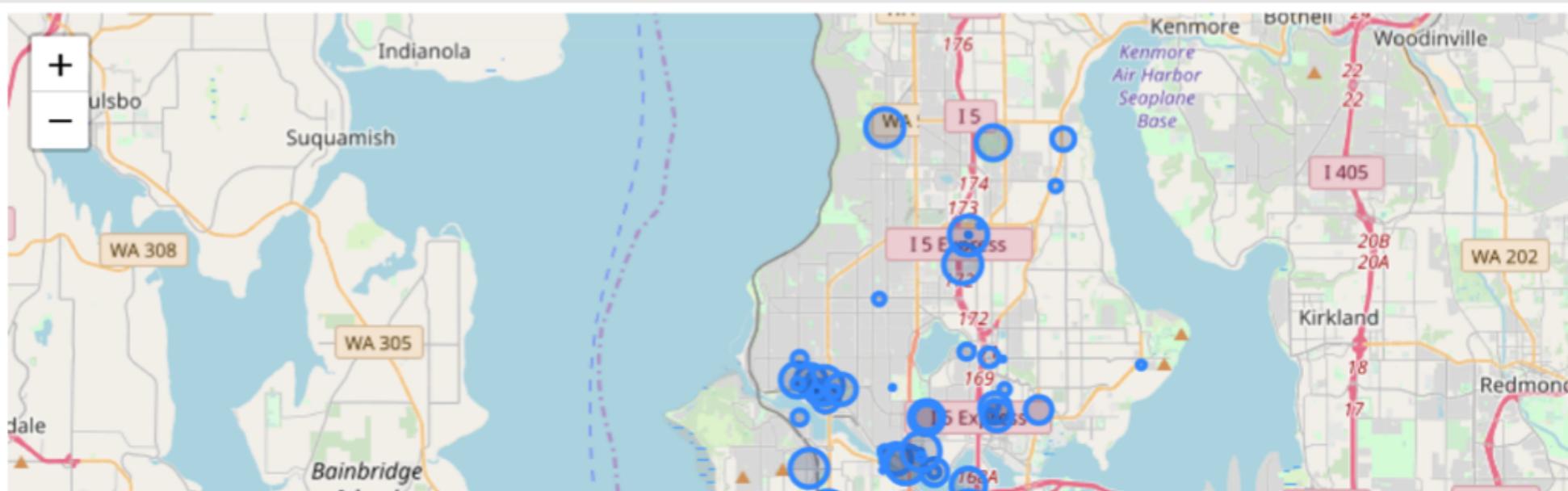
Provides js functionality in python environment, allows for integration with other python packages like numpy and pandas

Intuitive naming for features and parameters

Highly readable code enables humans to easily interpret code functionality

```
[131]: def create_map(data):
    my_map = folium.Map(location=[np.mean(data['Latitude']), np.mean(data['Longitude'])], zoom_start = 11)
    for index, row in data.iterrows():
        folium.CircleMarker(
            location = [row['Latitude'], row['Longitude']],
            radius=row['Accident_Count'],
            fill = True,
            popup = folium.Popup(str("Address: " + row['Address'])),
            fill_color='#132b5e'
        ).add_to(my_map)
    return my_map

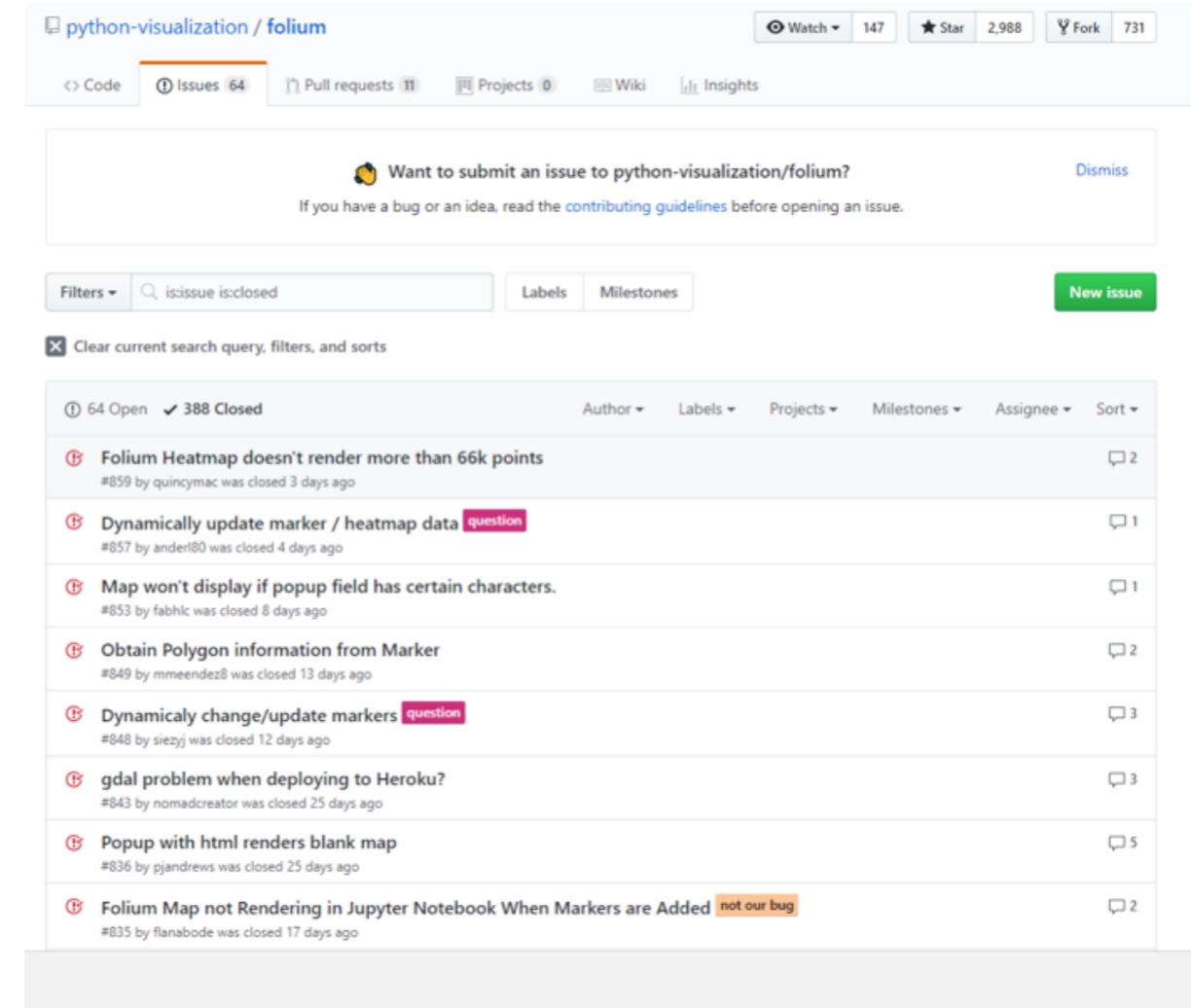
create_map(commercial_permits)
```



# PROS: IT HAS A ROBUST SUPPORT COMMUNITY

---

- Great documentation with examples
- Community actively posting and responding to issues, as recently as this week.

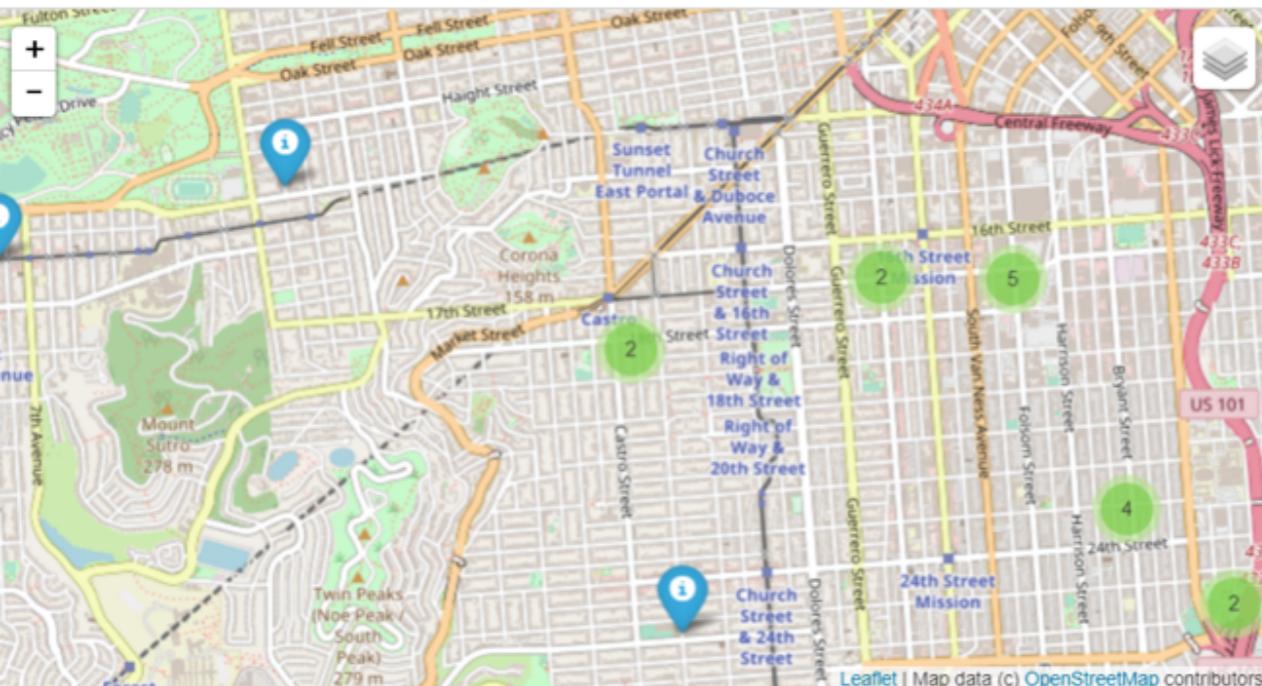


# PROS: LOTS OF EXAMPLES AND ANSWERS ARE AVAILABLE

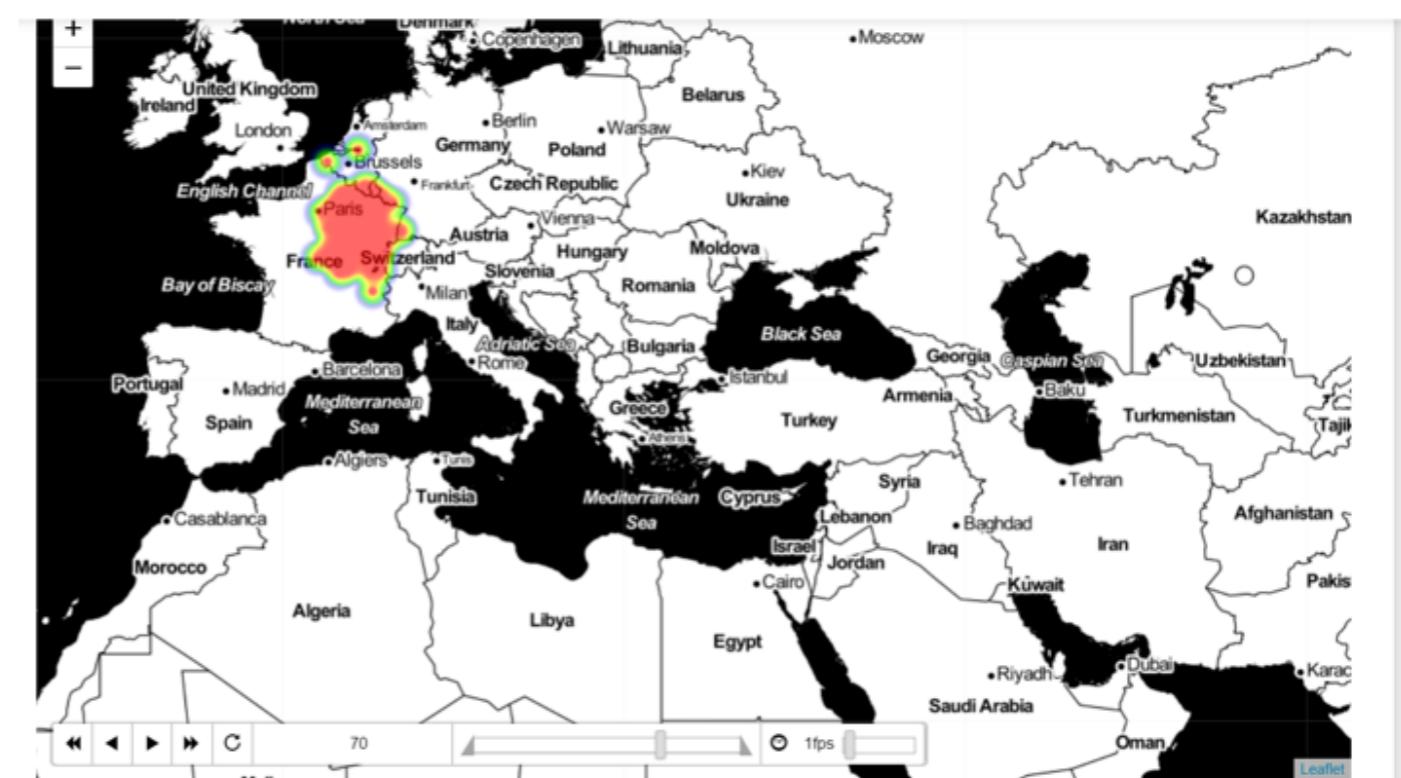
---

- Useful tutorials are available for interesting features
- It's easy to find answers to many questions via google search

Automatically cluster data when data points are close



Automatically update maps using a slider



# CONS: INSTALLATION REQUIRED

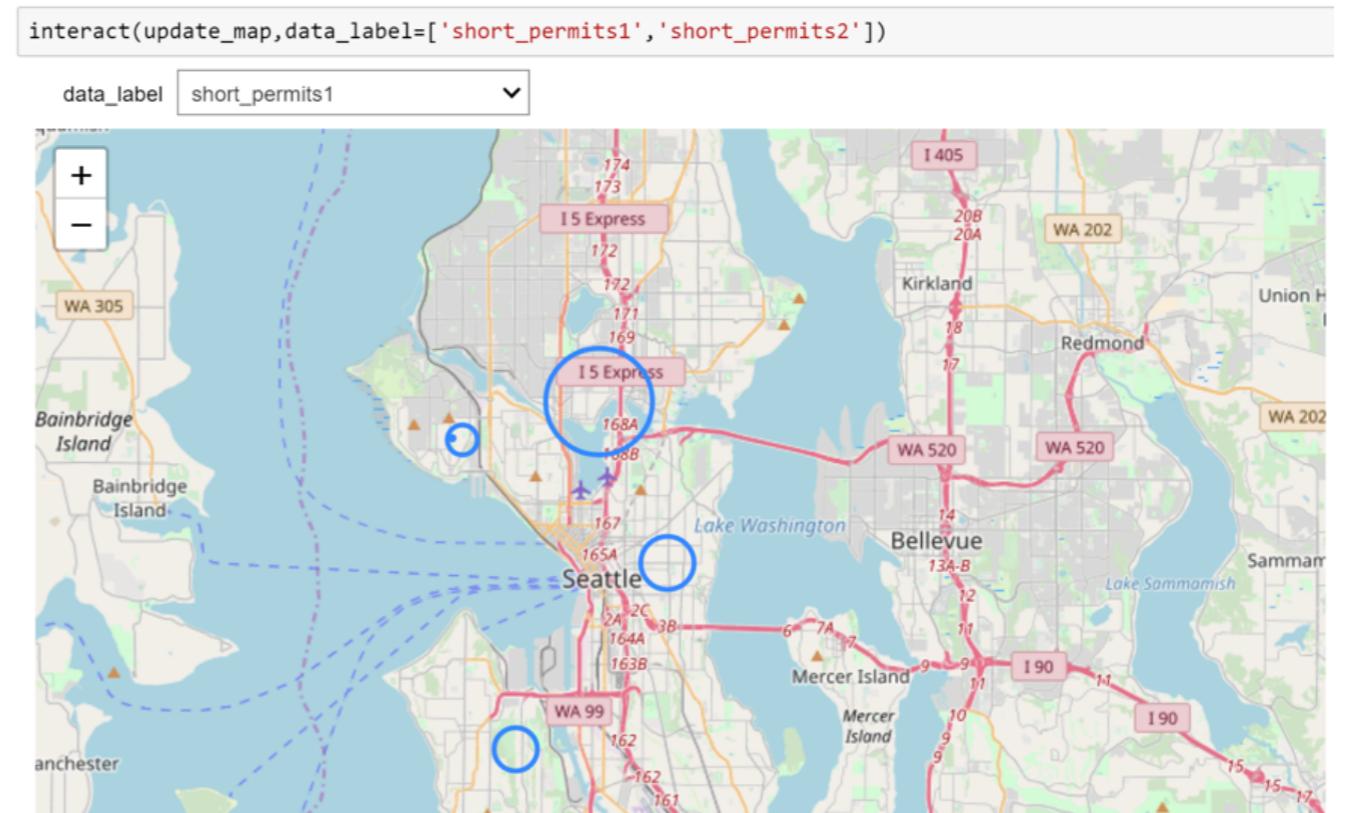
---

```
Ian@bear MINGW64 ~
$ pip install folium
Collecting folium
  Downloading https://files.pythonhosted.org/packages/07/37/45fb3699ed23caa0011f8b90d9cad94445eddc656b601e6268090de35f5/folium-0.5.0.tar.gz (79kB)
Collecting branca (from folium)
  Downloading https://files.pythonhosted.org/packages/c3/fe/c140768eaf9d14ef6e844d0e9679b7085d87ceaa578020fd6fed679d74c/branca-0.2.0-py3-none-any.whl
```

# CONS: INTERACTIVITY

---

- No built-in interactivity
  - Must use ipython interactive widgets
  - Requires map rendering at each change
  - Jupyter only renders maps with up to ~2500 points, which takes ~8 sec.



## CONS: AESTHETICS

---

- Google Maps vs. OpenStreetMap
- Not a lot of decoration control
  - HTML formatting no longer enabled for output
  - Markers best option for our use cases
- No syncing of side-by-side maps
  - Branca allows side-by-side Folium maps without iFrames
  - Leaflet.Sync plug-in does not enable dragging sync



WHY FOLIUM?

---



matplotlib

Leaflet



# WHY NOT BOKEH?

---



- *Poor documentation.*
- *Additional API access required for map integration.*
- *Learning curve.*

**matplotlib**  
The matplotlib logo features the word "matplotlib" in a bold, dark blue sans-serif font. To the right of the text is a circular icon containing a stylized sunburst or radar chart design with orange, green, and yellow segments.

**Leaflet**  
The Leaflet logo features the word "Leaflet" in a light gray, lowercase, sans-serif font. To the right of the text is a small green leaf icon.



# WHY NOT MATPLOTLIB?

---

**matplotlib**



- *Mapping requires use of shape files.*
- *Inability to connect with Google Maps API.*



# WHY NOT LEAFLET?

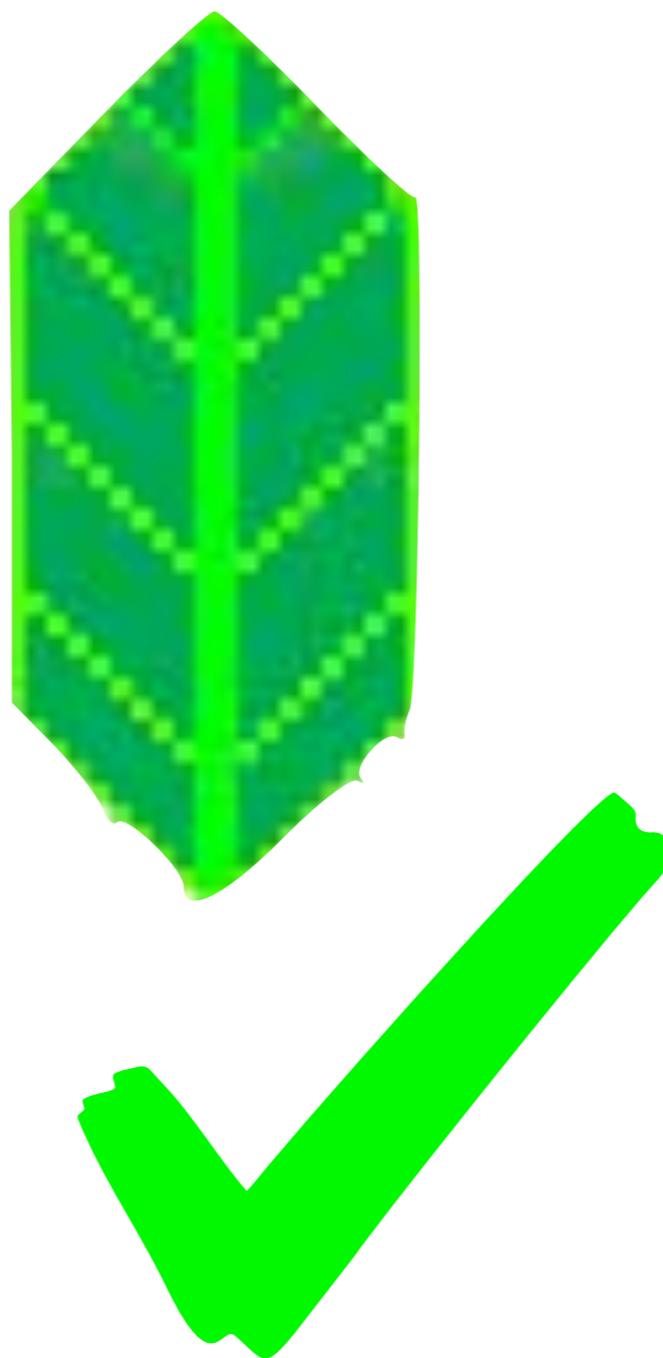
---



- Required an additional Python module.
- Folium does the heavy lifting and gives good results with less code. Performs all Leaflet function as a Python wrapper.

# FOLIUM MADE THE MOST SENSE

---



---

*Thank You!*