2021

ART350 Data Visualization

Semester project

Air Watch
Bay Area
Learn more about air quality in the San Francisco Bay Area and report air pollution. Use our interactive tool to know how air pollution affects your community, especially pollution from oil refineries.

Figure 1. Airwatch

The semester project will allow you to combine multiple skills and insights you gained in this course towards a comprehensive data visualization task. Importantly, you will have direct access to a unique collection of environmental data sourced directly from sensors in the San Francisco Bay Area and curated on the Air Watch Bay Area platform.

Air Watch Bay Area

Air Watch Bay Area is a collaboration between the Fair Tech Collective and communities adjacent to oil refineries in the Bay Area. Since the 1990s, residents of these "fenceline communities" have fought for more information about hazardous pollutants in the air they breathe. In the 2010s, new regulations and new technologies made real time air quality data available, but not easy to access or understand. Air Watch Bay Area attempts to consolidate the data and make it meaningful. You can read more about the project here.

Environmental Data - Particulate Matter

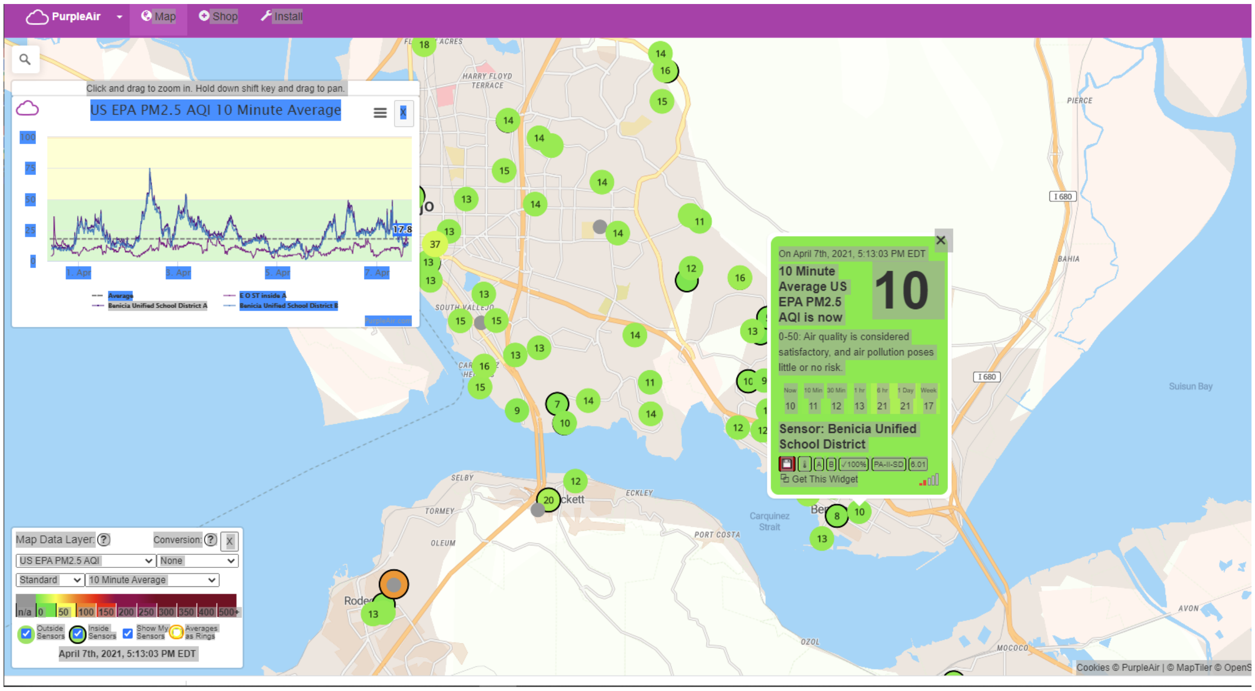
Particulate matter (PM) are small particles in the air. PM comes from a variety of sources, including motor vehicles, oil refineries, power plants, and wildfires. Exposure to PM can cause cardiovascular problems and respiratory symptoms; it has also been shown to worsen the effects of the COVID-19 virus.

PM is classified by its size. $PM_{2.5}$, for example, refers to particles with a diameter less than 2.5 microns. $PM_{2.5}$ and PM_{10} are regulated by the US National Ambient Air Quality Standards. PM concentrations--the mass of particles in a volume of air--are measured in micrograms per cubic meter.

PurpleAir

PurpleAir is a company that creates prosumer grade real-time air quality monitoring devices. PurpleAir devices measure PM1.0, PM2.5 and PM10 particles. Moreover, the organization makes the data (shared with the platform) available to the public in real time.

More information on PurpleAir is available here: https://www2.purpleair.com/
Additional details are discussed here: https://www2.purpleair.com/community/faq



Click and drag to zoom in. Hold down shift key and drag to pan.

On April 7th, 2021, 5:13:03 PM EDT

10 Minute Average US EPA PM2.5 AQI is now 10

0-50: Air quality is considered satisfactory, and air pollution poses little or no risk.

Now	10 Min	30 Min	1 hr	6 hr	1 Day	Week
10	11	12	13	21	21	17

Sensor: Benicia Unified School District

A B ✓100% PA-II-SD 6.01

Get This Widget

Map Data Layer: US EPA PM2.5 AQI

Conversion: None

Standard

10 Minute Average

☑ Outside Sensors ☑ Inside Sensors ☑ Show My Sensors ☐ Averages as Rings

April 7th, 2021, 5:13:03 PM EDT

Figure 2: PurpleAir map of the Benicia area.

Primary goal of this final assignment

The goal is to use all your data exploration and visualization skills to explore several datasets with PurpleAir particulate matter readings from Benicia and to create data intelligent and salient visualizations that describe the sensor data - and by extension help in interpreting what produced the sensor data in the first place.

Accessing sample code

The *DataVis2021_final.ipynb* file in UBlearns (assignments) has all the code you need to complete this assignment.

Accessing the datasets

Several datasets from Benicia, a North Bay region of the San Francisco Bay Area, are available on our GitHub data repository:
https://github.com/realtechsupport/DataNarratives/tree/master/AirWatch

You can access the files in the following manner:

```
BeniciaMiddleSchool = 'Benicia_Middle_School_1_2019.csv'
# other sites...
import pandas
source = BeniciaMiddleSchool
data = pandas.read_csv("https://raw.githubuser.../AirWatch/" + source)
data.head()
```

The result (column names and first row) will look something like this:

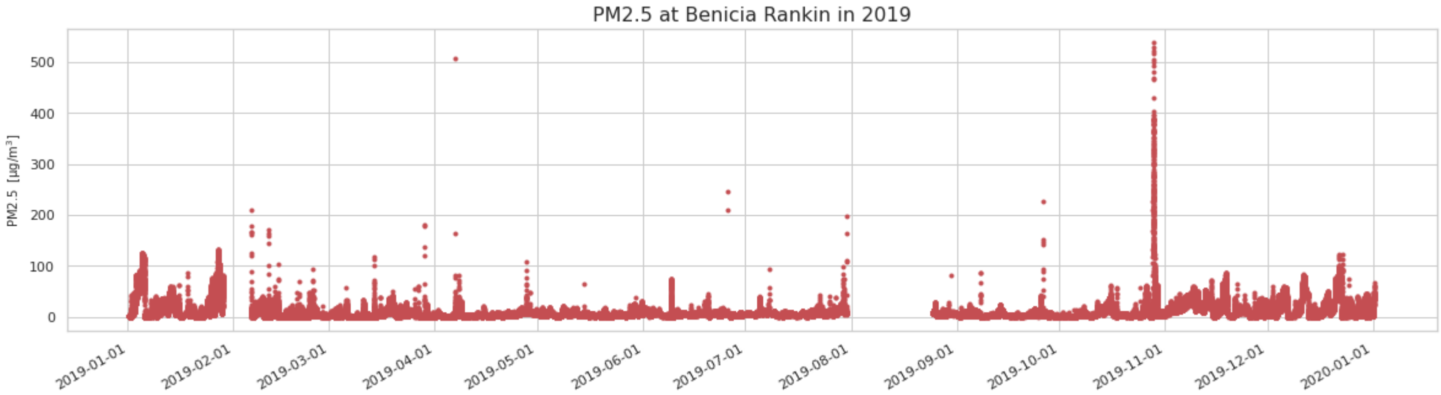
time	label	PM2.5	PM10	humidity	temperature	id	lat	long
2019-10-03 14:05:50	Benicia Middle School 1	4.45	4.90	23.0	76.0	39745.0	38.069053	-122.172851

Sensor readings are recorded every 2 minutes approximately. Label is the name of the location where the sensor resides. PM2.5 is the measurement of fine inhalable particulate matter, with diameters that are generally 2.5 micrometers and smaller. PM10 is the measurement of inhalable particles with diameters between 2.5 and 10 micrometers in diameter. Units for both PM2.5 and PM10 are indicated in micrograms / m^3. Humidity is indicated as a percentage (0 - 100). Temperature is indicated in Fahrenheit. The id is the sensor's unique identifier. Latitude and Longitude indicate location coordinates (-122.17 is 122.17 West)

Part 1 - time series sensor readings from a single location

You can plot time versus sensor measurements very easily with matplotlib:

```
fig, ax = plt.subplots()
x = data['time']
y = data['PM2.5']
ax.plot(x, y, 'r.')
```



The simple scatter plot already reveals some interesting details. Compare the various datasets to find events and patterns, and use different plotting approaches (Seaborn, for example) to visualize the data.

Part 2a - combining sensor data from different locations

You can display the data from multiple locations in different ways. The following code snippet creates a Seaborn histogram of the sensor readings from all the locations and filters them to show only those above a threshold:

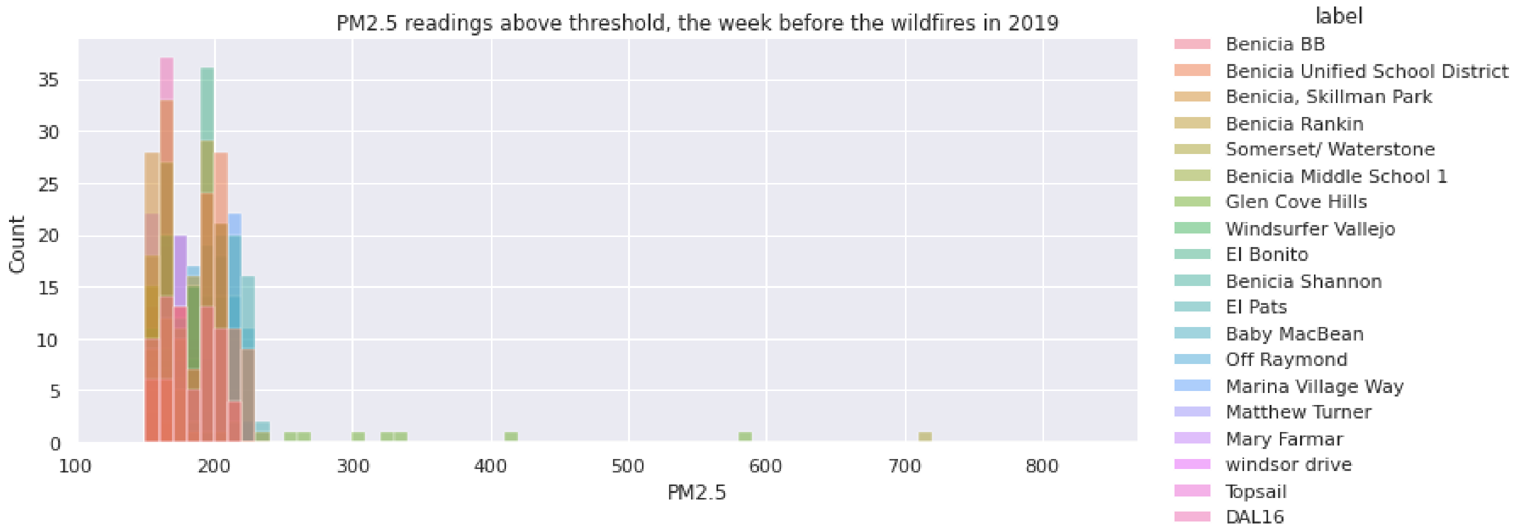
```
beforewildfiredata = pandas.read_csv("https://raw.githubusercontent.com/ …..”)

threshold = 150
temp_b = beforewildfiredata[beforewildfiredata['PM2.5'] > threshold]

g = seaborn.displot(temp_b, x='PM2.5', hue='label', binwidth = 10)

g.fig.set_figwidth(20)
g.fig.set_figheight(4)

plt.xlim(100, 1000)
plt.title('PM2.5 readings above threshold, week before the wildfires in 2019')
plt.show()
```



Part 2b - mapping sensor data to sensor locations

Geopandas and geoplot make it easy to combine location-specific information with sensor readings. The following code snippet, for example, outlines how to create a density map of sensor readings around the PurpleAir sensor locations in the study area:

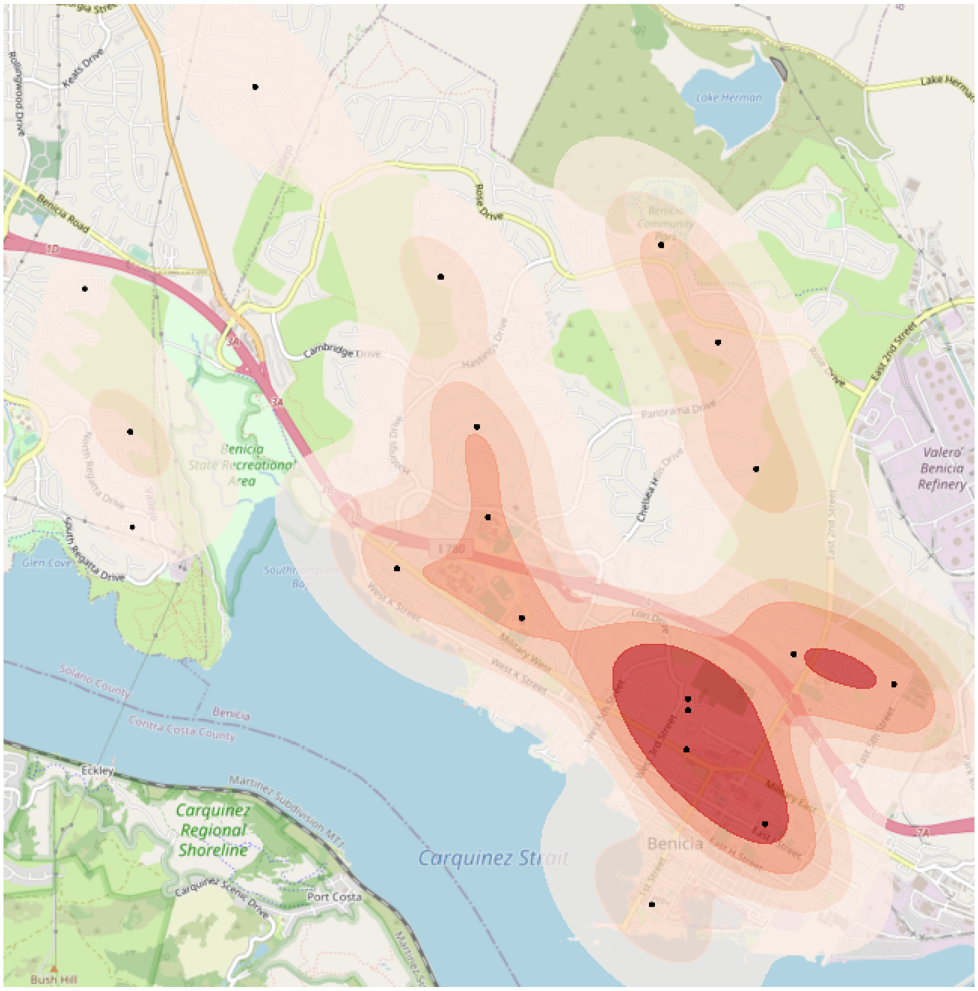
```
source = pandas.read_csv("somefile.csv”)
threshold = 150

beforewildfiredata_th = beforewildfiredata[beforewildfiredata['PM2.5'] >
      threshold]

bwdata_th = geopandas.GeoDataFrame(beforewildfiredata_th,
      geometry=geopandas.points_from_xy(beforewildfiredata_th.longitude,
      beforewildfiredata_th.latitude))

ax = geoplot.kdeplot(bwdata_th, projection=gcrs.WebMercator(), figsize=(12,12),
      shade=True, cmap='Reds', shade_lowest=False, n_levels=6, alpha=0.7)

geoplot.pointplot(bwdata_th, s=2, color='black', alpha=0.5, ax=ax)
geoplot.webmap(bwdata_th, ax=ax)
```



PurpleAir sensors in Benicia showing frequency of sensor readings above 150 ug/m^3 one week after wildfires in 2019. The black dots are the locations of the sensors.

What differences can you detect before and after the wildfire? You will need to ingest the files

'week_after_wildfires_2020.csv' and 'week_before_wildfires_2020.csv'

in order to address that question.

Include a short text on your approach and observations in the jupyter notebook that you submit (see UBlearns notebook example for the template).

Extra Credit: design an enticing overview site

If you click on ‘Benicia’ from the main page http://www.airwatchbayarea.org/ you land at this site:

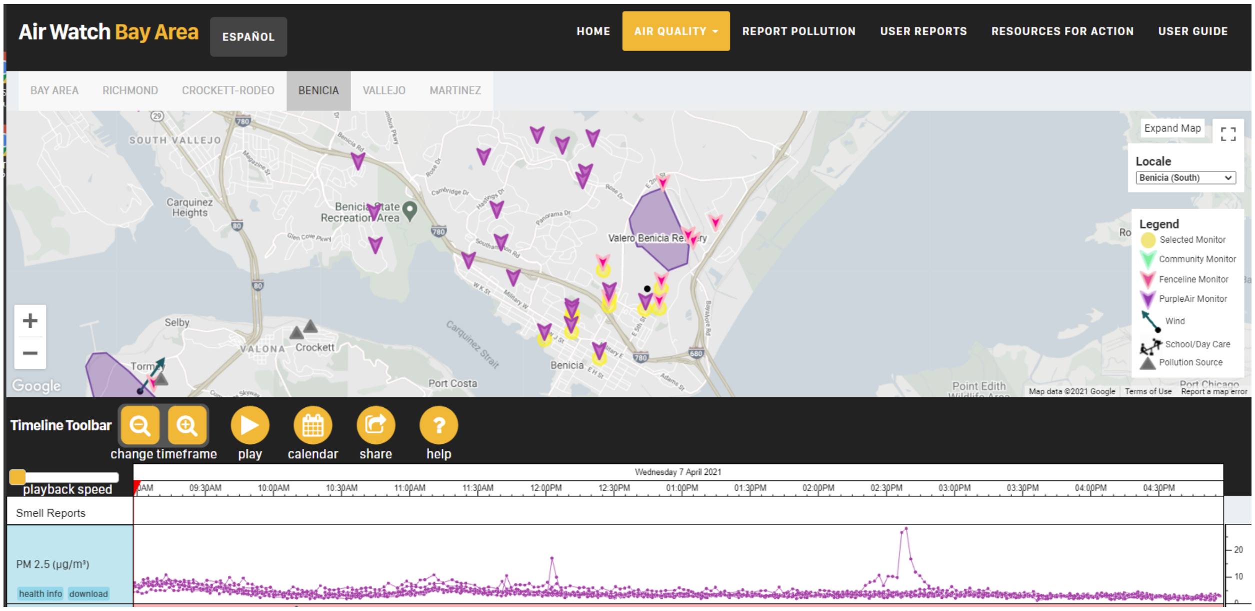


Figure 3. Airwatch overview site for Benicia

How would you improve the design of this page? Create a (non-functional, image only) mock-up in your favorite graphics editing environment that *visually incorporates your suggested improvements*.

DELIVERABLES

1) Time series plots from two or more locations in the study area
2a) Graph or histogram of the combined sensor data.
2b) Maps or graphs of sensor readings before and after the wildfires.
3) Design for a landing page (extra credit)

POINTS Max 42 (12 points for each 1, 2a, 2b and 6 extra credits for 3)

DEADLINE Thursday, May 13th, noon.

FORMAT and DELIVERY

1) Create a link to a single jupyter notebook with all code, visualizations and descriptions.
2) Save the landing page design as a jpeg: ‘firstname_lastname_ART350_final_landingpage.jpeg’

Send the link to your jupyter notebook to: marcbohlen@gmail.com before the deadline.
Include the landing page design for extra credit.