


A background image of a mangrove landscape. In the foreground, there are dark, silty waters with numerous mangrove roots (pneumatophores) protruding from the water. In the mid-ground, there are lush green mangrove trees with dense foliage. The sky is bright and slightly hazy. The overall scene is a natural, coastal environment.

OCN 390: Field Methods


Week 11

Data Analysis in Python



Group Project Check-ins

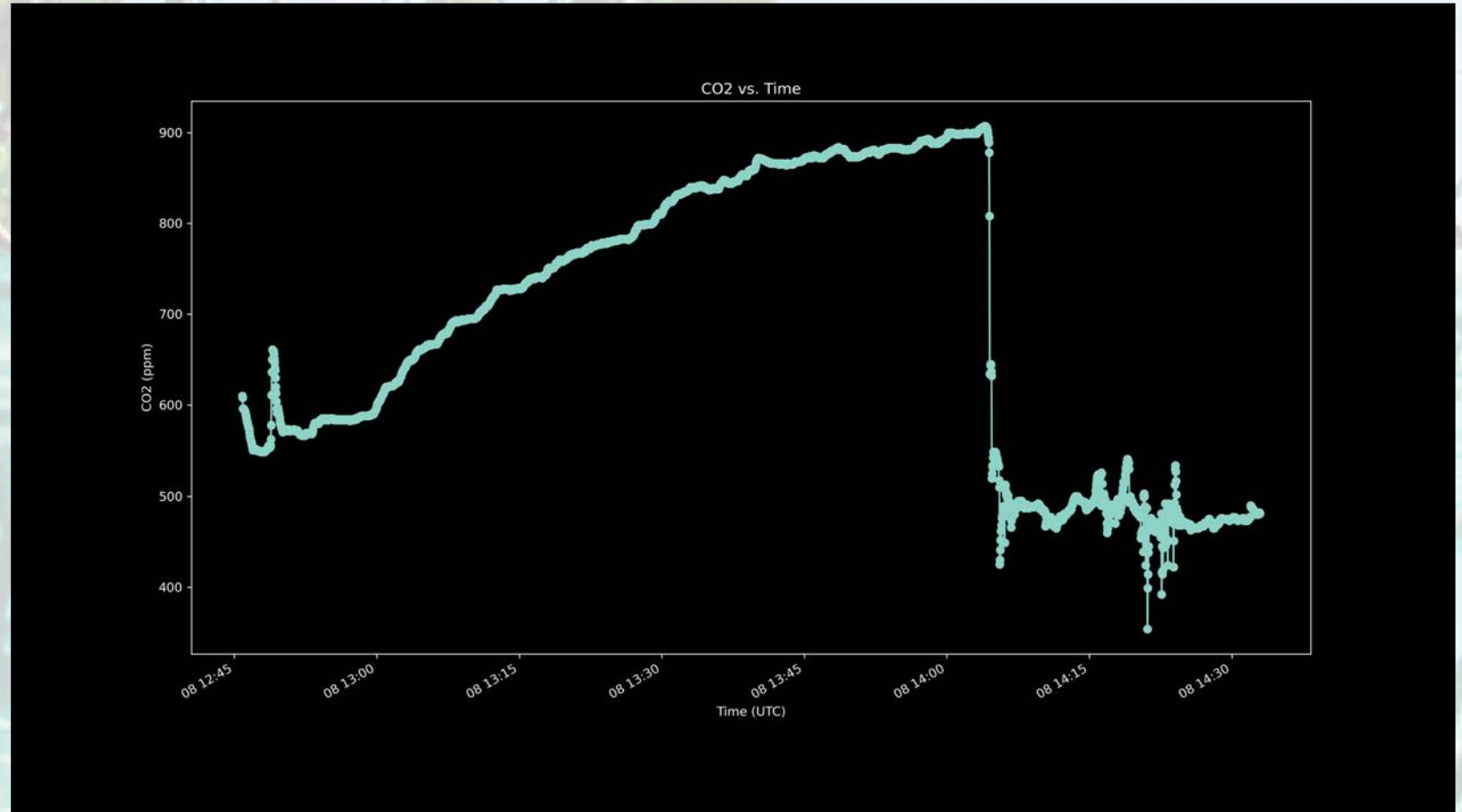
What are current plans? How will you analyze and visualize data? Have you already collected data? Analyzed it? Started Story Map?

A photograph of a mangrove forest with clear turquoise water and green trees. A white bird is perched on a branch on the right. The text 'Python for Earth Science' is overlaid in the center-left.

Python for Earth Science

Figure Markup

- What features do you see in this figure?



K-30 test

- Plugged K-30 into power in DL 114 @ 12:45:47 pm (EST). Me + one other student in room
- 2nd student (3rd person in room) arrives at 12:48.

3rd @ 12:48.

4th @ 12:52

5th @ 12:54

6th @ 12:55

7th @ 12:56

8th @ 12:56

9th @ 12:57

10th @ 12:58

11th @ 12:58

12th 13th @ 13:00

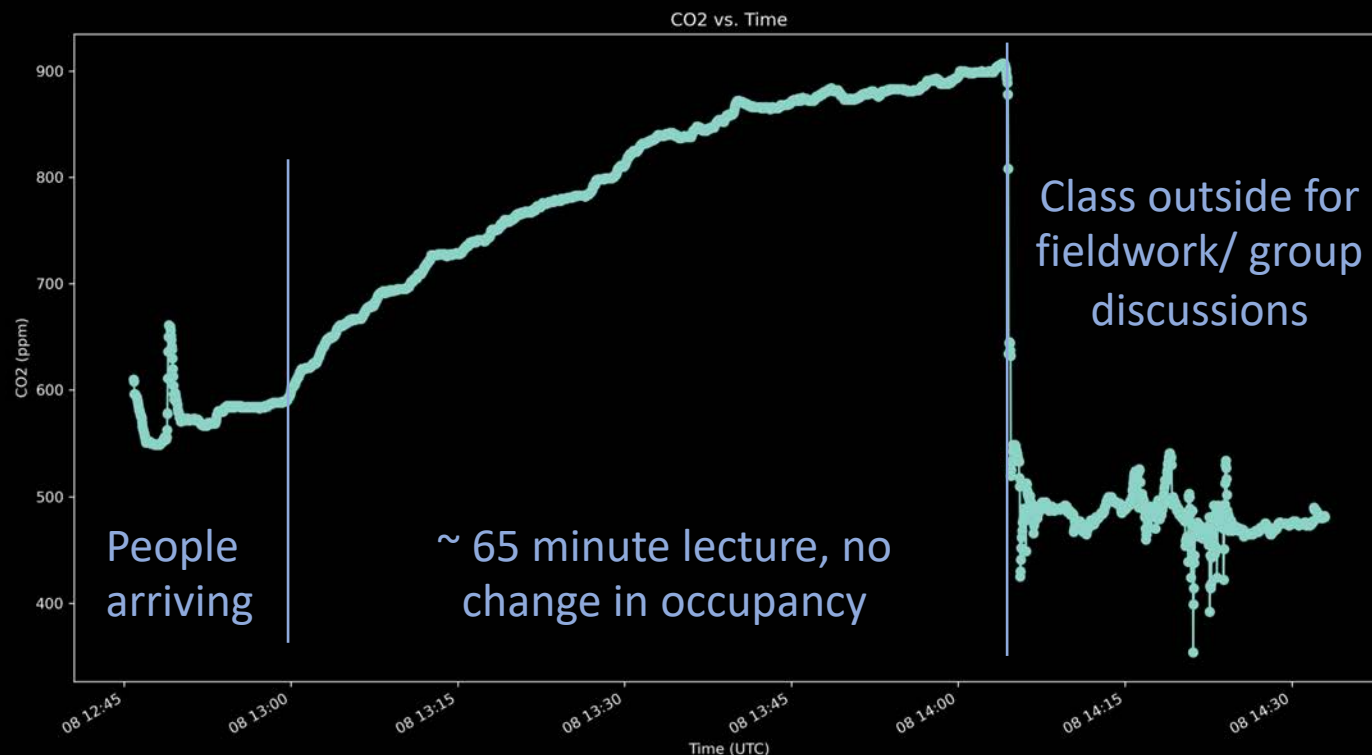
Even on Zoom

Class end @ 1400

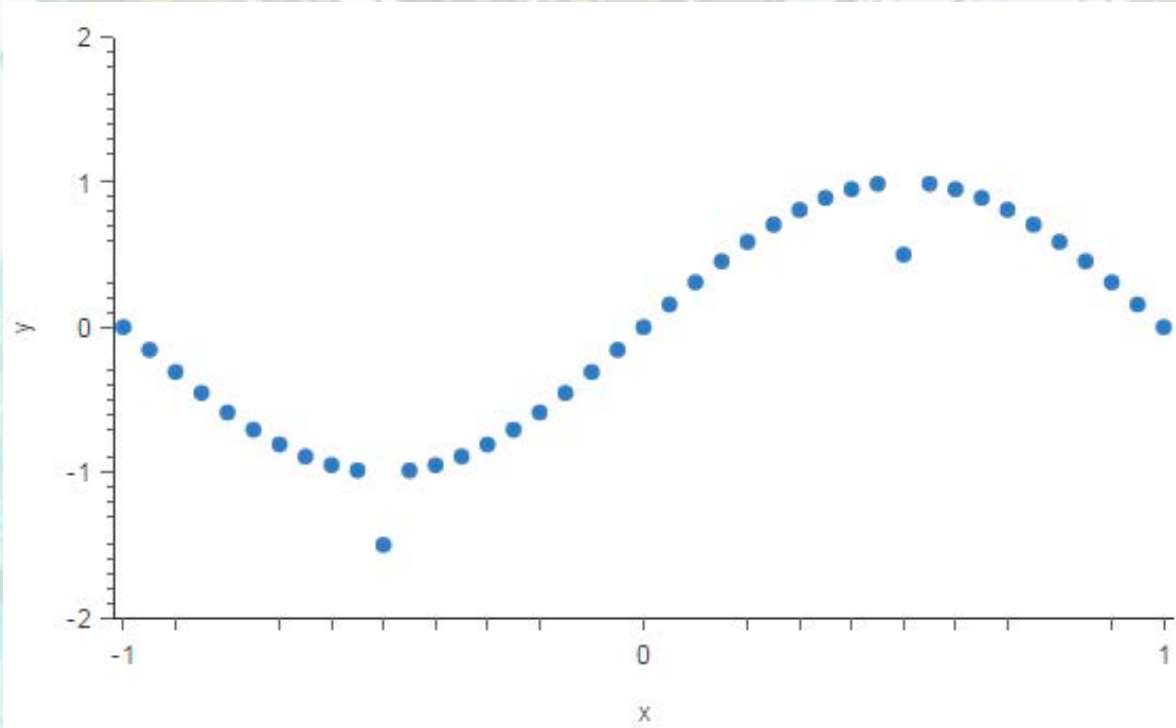
outside by 1405

power off @ 1432

8 Mar 2020



Outlier Detection

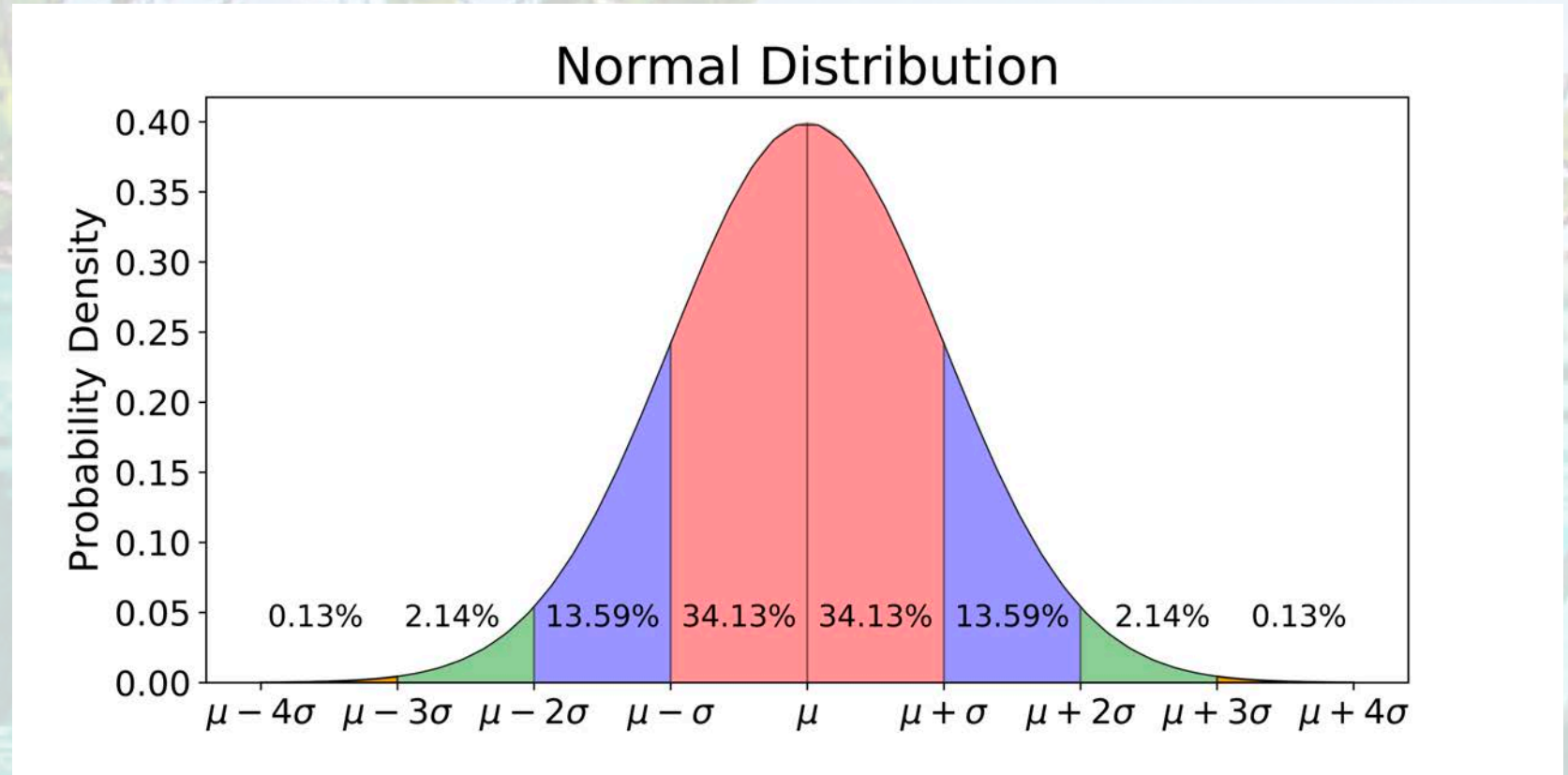


<https://www.kdnuggets.com/2017/01/3-methods-deal-outliers.html>

- Simply plotting all of your data without describing it is insufficient
- How do we know if a point counts as an outlier (as opposed to being an actual, natural deviation)?

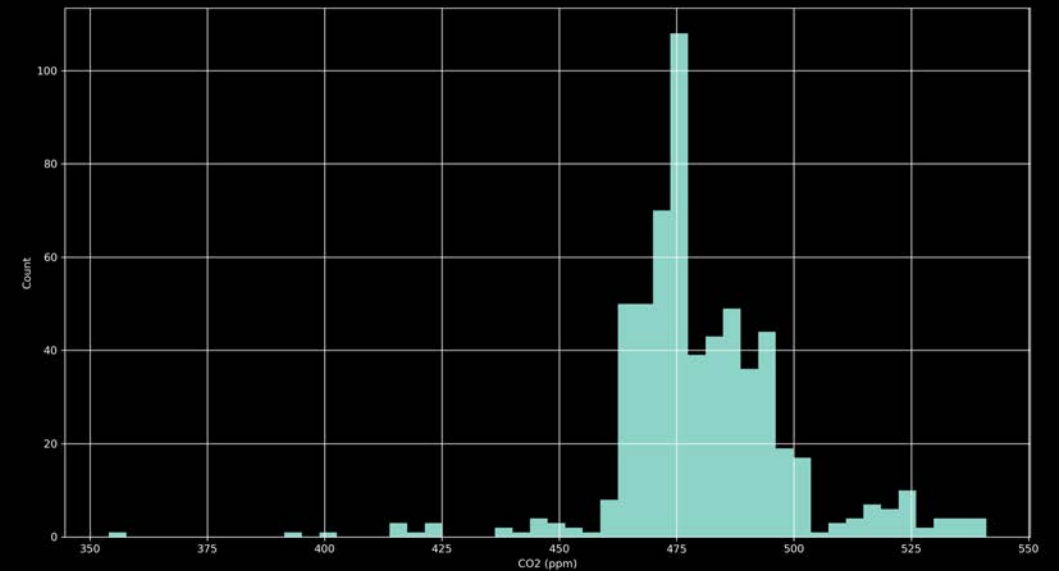
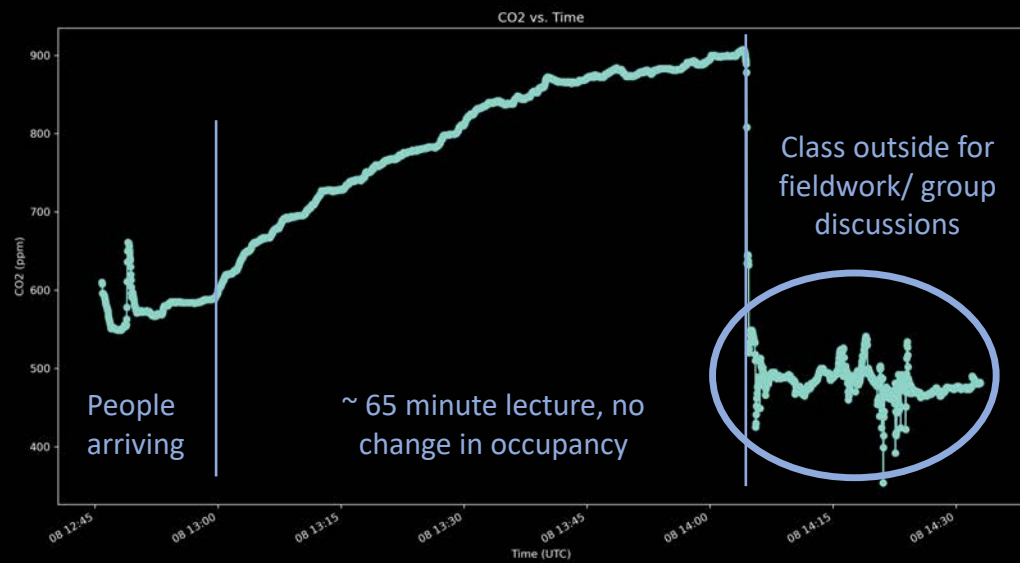
One method: local standard deviation

- if point is > 2 standard deviations from (local) mean, remove
- why 2 x standard dev?
- μ = mean
- σ = standard dev

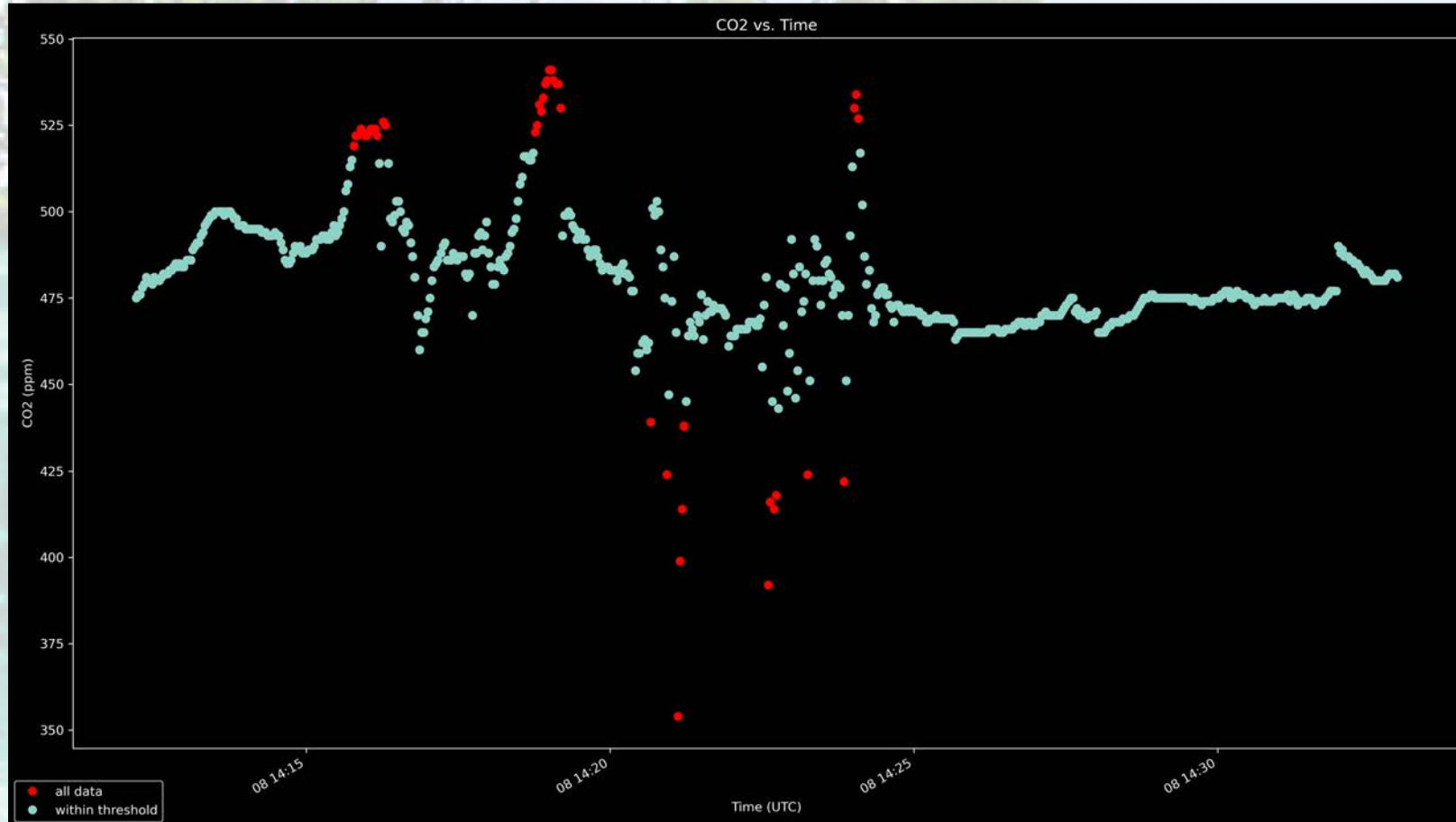


<https://towardsdatascience.com/understanding-the-68-95-99-7-rule-for-a-normal-distribution-b7b7cbf760c2>

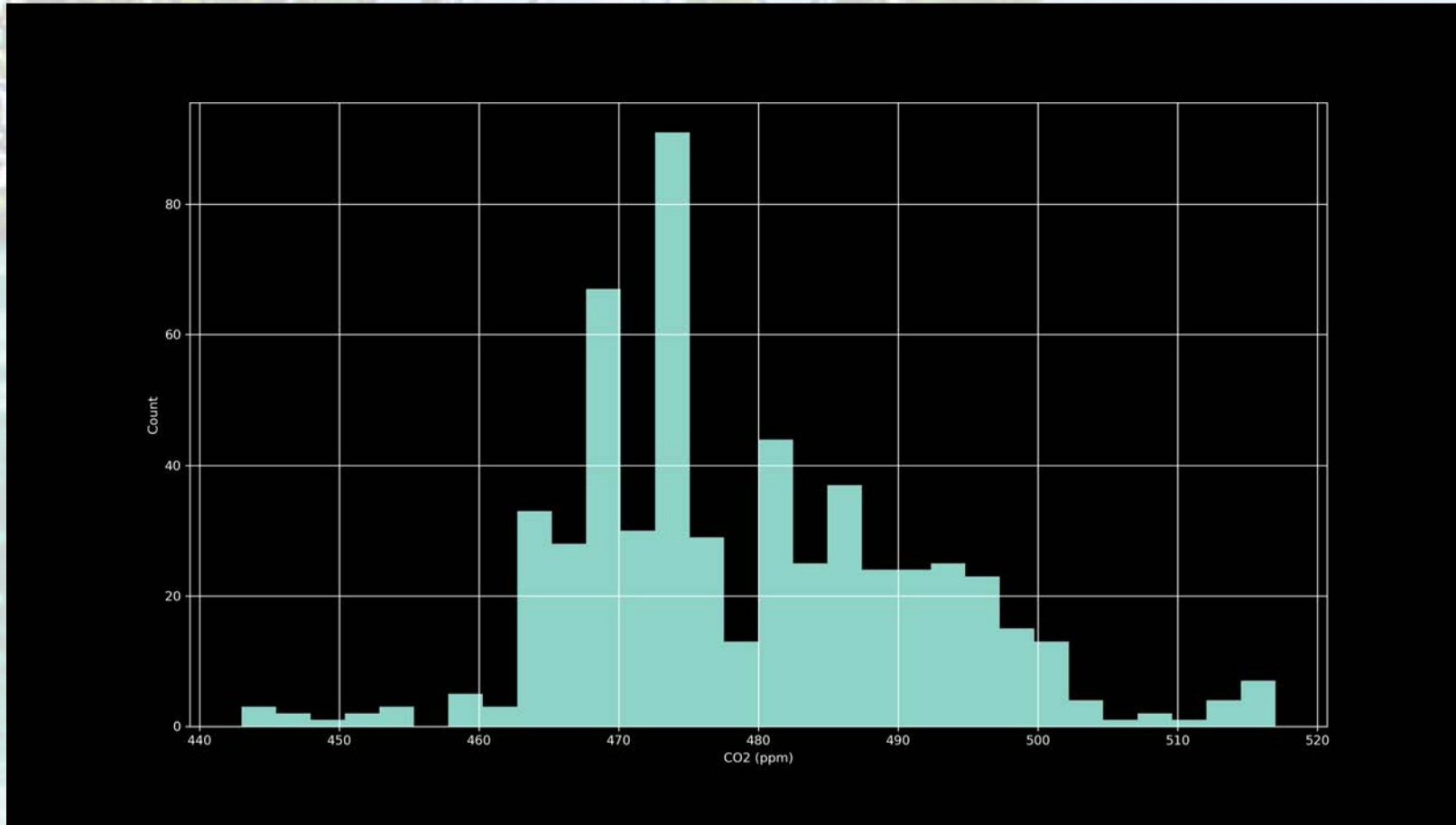
Distribution of our outdoor data



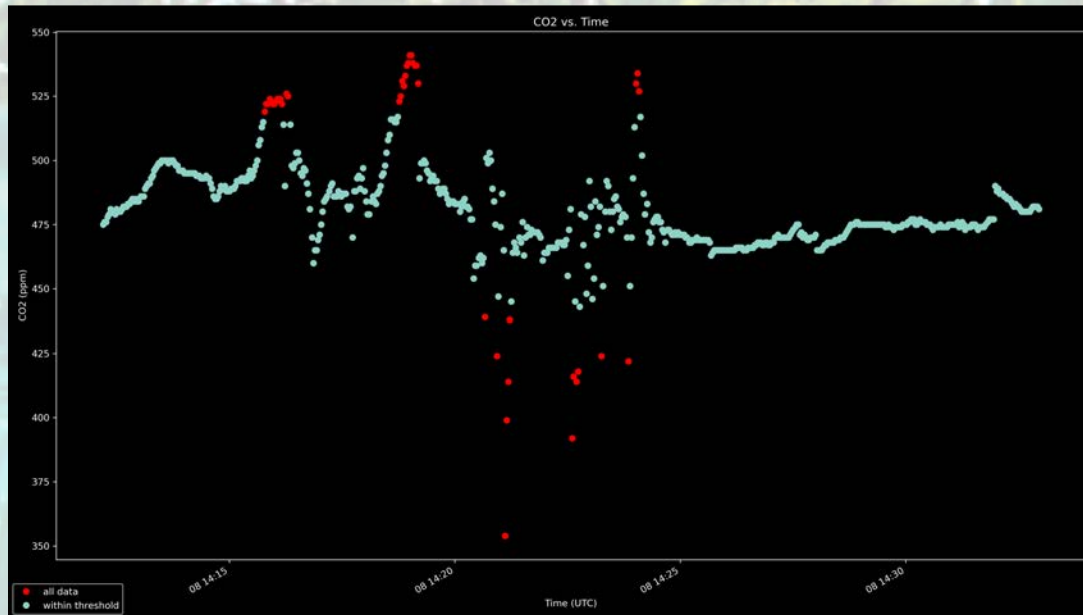
Identifying data for further processing



Histogram after outlier removal



Notes of Caution



1. It *could* be “good” data so chosen approach merits thoughtfulness
2. NEVER remove data without justifying having done so. It could look like (or actually be) improper manipulation.

Python vs. Jupyter

- Python is the actual coding language:

```
df_C02 = pd.read_csv(C02_filename)
elapsed_time_sec = df_C02.iloc[:, 0] # insensitive to column name
C02 = df_C02.iloc[:, 1]
```

- Jupyter is the interface, which provides places for input (like code, other text) and output (like tables, figures, results)
- Python works without Jupyter and vice versa, but it's a powerful combo

Getting started

Before executing any analysis in Python, save the file from the K-30 as a .csv file (rather than .txt) and add a header row of *elapsed time (sec),CO2 (ppm)*

```
[2]: # Read in data
df_C02 = pd.read_csv('2021-03-15_C02SensorUpload.csv')

# Time when I noted that C02 logging began *** in UTC ***
# Year, month, day, hour (24 hr format), minute, second
starttime_est = datetime.datetime(2021, 3, 8, 12, 45, 47)
df_C02.head()
```

```
[2]:
```

	elapsed time (sec)	CO2 (ppm)
0	2	0
1	4	610
2	6	608
3	8	596
4	10	596

Why Python in the first place?

- (or R, Octave, MATLAB, ...)
- Next portion of lecture courtesy of Earth Lab at the University of Colorado, Boulder
<https://www.earthdatascience.org/about/>
- Specific resources:
 - <https://www.earthdatascience.org/courses/intro-to-earth-data-science/open-reproducible-science/get-started-open-reproducible-science/>
 - <https://www.youtube.com/watch?v=NGF00kdbZmk> (NOVA)
 - <https://www.earthdatascience.org/courses/intro-to-earth-data-science/open-reproducible-science/jupyter-python/>

Elizabeth Iorns
SCIENCE EXCHANGE

NOVA



0:48 / 14:59



Why Python

- Free
- Reproducible analyses
- Relatively easy to share (Jupyter notebooks, code, etc.) because others can install software
- Great user community, eager to help (lots of good online assistance)

Why Git (Hub) ?

- Build on others' projects
- Collaborate
- Track changes in code

The screenshot shows a GitHub repository page. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. Below this, a green banner promotes learning Git and GitHub without code, with a 'Read the guide' button. The repository name 'uncw-ocean-field-methods / ocn390_2021spring' is displayed, along with options to Unwatch, Star (0), and Fork (0). A secondary navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is active, showing a file tree with folders like Analysis, Documentation, Lecture-PDFs, and Sensor-Dev, and a file named .aitianore. Each item has a description and a commit date. On the right, the 'About' section states 'No description, website, or topics provided.' and the 'Releases' section states 'No releases published' with a link to 'Create a new release'.

github.com/uncw-ocean-field-methods/ocn390_2021spring

Search or jump to... Pull requests Issues Marketplace Explore

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

uncw-ocean-field-methods / ocn390_2021spring

Unwatch 1 Star 0 Fork 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

	SUPScientist Add lecture, update notebook with new data	3d09681 14 days ago 27 commits
	Analysis	Add lecture, update notebook with new data 14 days ago
	Documentation	Add files via upload 2 months ago
	Lecture-PDFs	Add lecture, update notebook with new data 14 days ago
	Sensor-Dev	update main CO2 datalogger script to print out time elapsed 28 days ago
	.aitianore	Update repo with aitianore and aitianored directories 2 months ago

About

No description, website, or topics provided.

Readme

Releases

No releases published

Create a new release

How to code (for data analysis, sensor development, etc.)

- We haven't gone far enough to fairly call this even a crash course in Python or Arduino
- When taking the next steps in your projects (or careers), you may choose/have to code
- A couple suggestions:
 1. Google
 2. (Free!) online coding classes/tutorials
 3. Classes here

A screenshot of a Google search interface. The search bar contains the text "python calculate standard deviation with nan". Below the search bar, there are tabs for "All", "Videos", "Images", "News", "Shopping", and "More". The search results show "About 744,000 results (0.50 seconds)". The first result is from NumPy's documentation, titled "numpy.nanstd — NumPy v1.20 Manual". The snippet describes the `nanstd` function: "Compute the standard deviation along the specified axis, while ignoring NaNs. Returns the standard deviation, a measure of the spread of a distribution, of the non-NaN array elements." The second result is from a website "het.as.utexas.edu", titled "numpy.nanstd — NumPy v1.9 Manual", with a similar snippet.

python calculate standard deviation with nan

All Videos Images News Shopping More Settings Tools

About 744,000 results (0.50 seconds)

numpy.nanstd. Compute the **standard deviation** along the specified axis, while ignoring **NaNs**. Returns the **standard deviation**, a measure of the spread of a distribution, of the non-**NaN** array elements. Jan 31, 2021

[https://numpy.org/doc/stable/reference/generated/numpy.nanstd — NumPy v1.20 Manual](https://numpy.org/doc/stable/reference/generated/numpy.nanstd.html)

About featured snippets Feedback

[https://het.as.utexas.edu/Numpy/reference/generated/numpy.nanstd — NumPy v1.9 Manual](https://het.as.utexas.edu/Numpy/reference/generated/numpy.nanstd.html)

numpy.nanstd. Compute the **standard deviation** along the specified axis, while ignoring **NaNs**. Returns the **standard deviation**, a measure of the spread of a distribution, of the non-**NaN** array elements.

Recommendations (not requirements) for your projects

- Create a folder called **OCN390 Spring 2021 Project**
- Collect data in whatever way makes the most sense to you, given your project needs and goals
- Immediately save all output (e.g., K30_0000.txt, any GPX files, field notes) to a subfolder called **Field Study *2021-03-29* Original Data**
- Create a different subfolder called **Analysis** and COPY (don't cut and paste) data and code for subsequent analysis into that

For example...

- DOWNTOWN.csv and WRIGHTSVILLE.csv files are both derived from K30_0000.txt
- Saved as different files so that we can easily take statistics of each, plot, etc. separately using associated .ipynb files

Name	
▼	Analysis
📄	2021-03-08_CO2SensorTestWalk_DOWNTOWN.csv
📄	2021-03-08_CO2SensorTestWalk_WRIGHTSVILLE.csv
📄	CO2_sensor_test_walk DOWNTOWN.gpx
📄	CO2_sensor_test_walk WRIGHTSVILLE.gpx
📄	Final_Analysis_Aggregate.ipynb
📄	K30_interpret DOWNTOWN.ipynb
📄	K30_interpret WRIGHTSVILLE.ipynb
▼	Field Study 2021-03-25 Original Data
📄	CO2_sensor_test_walk.gpx
📄	Field Journal.jpeg
📄	K30_0000.txt

Your data, your responsibility!!!

Lost/overwritten data not an excuse for not submitting a complete project. Store data in the cloud (OneDrive/Sharepoint), email copies out to teammates, create your own GitHub project page repository, etc.

Next week

- Keep working on Story Maps. Story Maps (team assignment; **deadline Monday, Apr. 12, 12:00 pm**) . Send to me if you want feedback by Monday, Apr. 5, 12:00 pm.
- Re-quiz: similar questions to last time + question about reproducibility
- Overview of Final Report