

RA Task

Data Generation

We want to use this source of data, ERA5 monthly averaged data on pressure levels from 2023, to generate the downup variable for Punjab, India in 2023. This downup variable should be calculated based on the area of Punjab's districts under wind influence. We want you to return a PDF, HTML, or Markdown document where you show your scripts and all the decisions you made. You should be very detailed in all your steps, argue all the decisions you take. This task uses large datasets, and we want to see how you manage them.

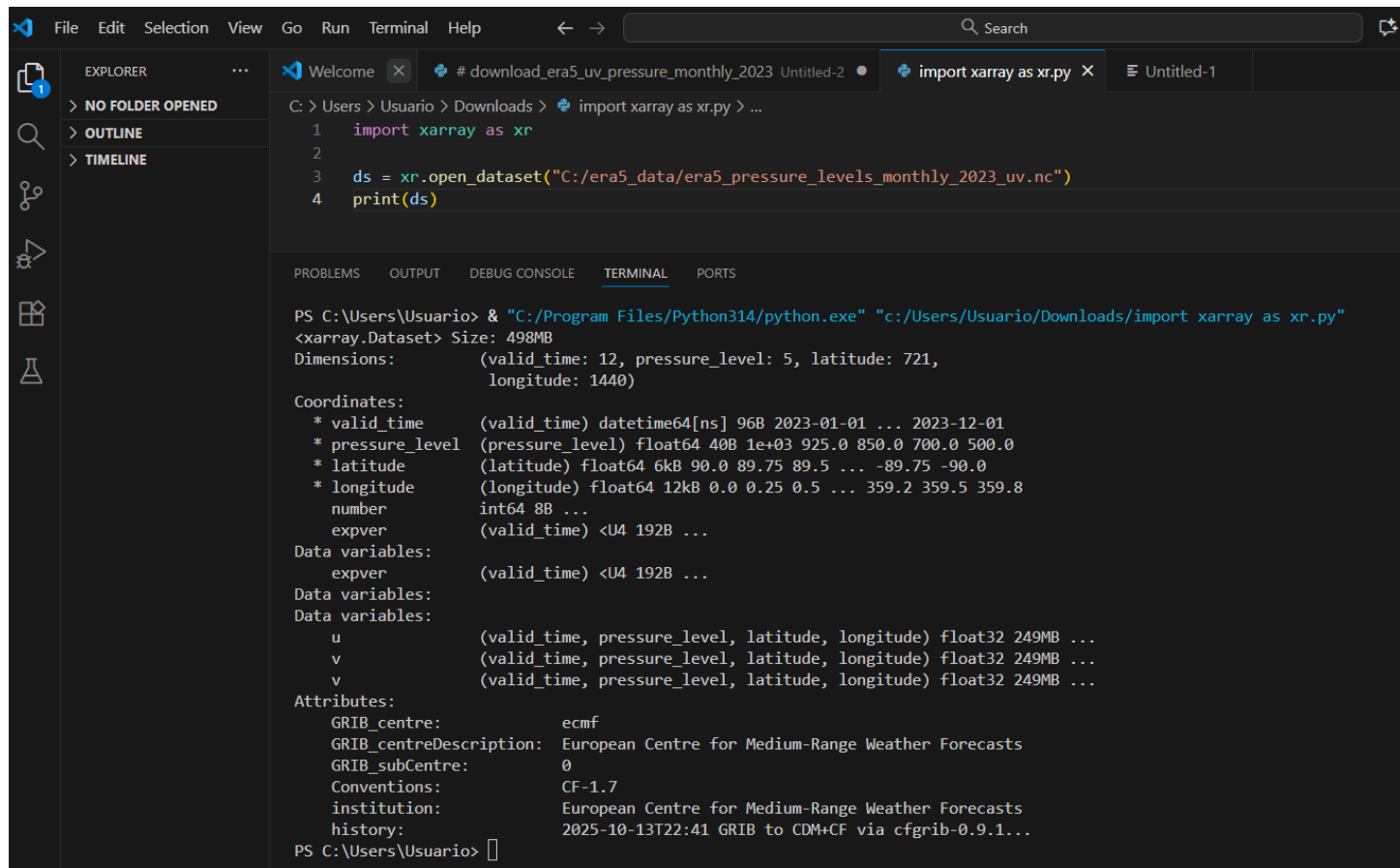
1. Obtain the data for U and V wind components for 2023 at the monthly level: U-component of wind, V-component of wind from this source:

<https://cds.climate.copernicus.eu/datasets/reanalysis-era5-pressure-levels-monthly-means?tab=download>

Step 1: Downloading ERA5 Monthly Wind Data for 2023

To obtain the wind data necessary for the downup analysis in Punjab, India, I used the Copernicus Climate Data Store (CDS) and Python. I configured the CDS API with my personal key and wrote a Python script to download the monthly-averaged U and V wind components at five pressure levels (1000, 925, 850, 700, 500 hPa) for the year 2023 in NetCDF format. The resulting dataset contains 12 monthly time steps and 5 pressure levels, allowing for detailed spatial and temporal analysis. The data integrity was verified using xarray, confirming that all variables and dimensions were correctly retrieved.

```
Welcome | # download_era5_uv_pressure_monthly_2023 Untitled-2 | import xarray as xr.py | Untitled-1
1 # download_era5_uv_pressure_monthly_2023.py
2 import cdsapi
3 from pathlib import Path
4
5 # Folder where the file will be saved
6 outdir = Path('C:/era5_data')
7 outdir.mkdir(parents=True, exist_ok=True)
8 outfile = outdir / 'era5_pressure_levels_monthly_2023_uv.nc'
9
10 # Create CDS client
11 c = cdsapi.Client()
12
13 # Typical pressure levels (you can change or reduce these)
14 pressure_levels = ['1000', '925', '850', '700', '500']
15
16 # Define request parameters
17 request = {
18     'product_type': 'monthly_averaged_reanalysis',
19     'format': 'netcdf',
20     'variable': ['u', 'v'], # u = eastward wind, v = northward wind
21     'pressure_level': pressure_levels,
22     'year': ['2023'],
23     'month': [f'{m:02d}' for m in range(1, 13)],
24 }
25
26 print("🕒 Sending request to Copernicus CDS, this may take a few minutes...")
27 c.retrieve('reanalysis-era5-pressure-levels-monthly-means', request, str(outfile))
28 print("Download completed. File saved to:", str(outfile))
```



```
File Edit Selection View Go Run Terminal Help
# download_era5_uv_pressure_monthly_2023 Untitled-2
import xarray as xr.py
1 import xarray as xr
2
3 ds = xr.open_dataset("C:/era5_data/era5_pressure_levels_monthly_2023_uv.nc")
4 print(ds)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Usuario> & "C:/Program Files/Python314/python.exe" "c:/Users/Usuario/Downloads/import xarray as xr.py"
<xarray.Dataset> Size: 498MB
Dimensions:          (valid_time: 12, pressure_level: 5, latitude: 721,
                      longitude: 1440)
Coordinates:
  * valid_time        (valid_time) datetime64[ns] 96B 2023-01-01 ... 2023-12-01
  * pressure_level    (pressure_level) float64 40B 1e+03 925.0 850.0 700.0 500.0
  * latitude          (latitude) float64 6kB 90.0 89.75 89.5 ... -89.75 -90.0
  * longitude         (longitude) float64 12kB 0.0 0.25 0.5 ... 359.2 359.5 359.8
    number            int64 8B ...
    expver            (valid_time) <U4 192B ...
Data variables:
  expver              (valid_time) <U4 192B ...
Data variables:
Data variables:
  u                   (valid_time, pressure_level, latitude, longitude) float32 249MB ...
  v                   (valid_time, pressure_level, latitude, longitude) float32 249MB ...
  v                   (valid_time, pressure_level, latitude, longitude) float32 249MB ...
Attributes:
  GRIB_centre:        ecmf
  GRIB_centreDescription: European Centre for Medium-Range Weather Forecasts
  GRIB_subCentre:     0
  Conventions:        CF-1.7
  institution:        European Centre for Medium-Range Weather Forecasts
  history:             2025-10-13T22:41 GRIB to CDM+CF via cfgrib-0.9.1...
PS C:\Users\Usuario>
```

2. Get the number of fires for Punjab India from <https://firms.modaps.eosdis.nasa.gov/download/>.

Step 2: Retrieve the Number of Fires for Punjab, India

The fire data for Punjab was obtained from the NASA FIRMS portal

for the year 2023. The dataset was downloaded in CSV format, including latitude and longitude of each fire event.

Using Python and pandas, the CSV file was loaded, and the fire events were filtered to include only those occurring within the geographic boundaries of Punjab. A grid corresponding to the ERA5 latitude and longitude coordinates was prepared, assigning a fire count of 0 to grid cells where no fires were reported.

3. Split wind grids into 25 smaller grids. Keep only the grids that touch Punjab. Use the shapefile of India available at <https://www.devdatalab.org/shrug>.

Step 3: Integrate downup Values with District Shapefile

The shapefile of India was loaded using geopandas. The dataset was filtered to retain only districts belonging to Punjab. To ensure a proper merge with the downup data, district names were cleaned by removing leading/trailing spaces, converting to lowercase, and standardizing spacing.

The CSV file containing downup values for each district was also cleaned in the same manner. Finally, a merge was performed between the Punjab shapefile and the downup CSV based on the cleaned district names, resulting in a geospatial dataset where each district includes its respective downup value. The results were verified to ensure all districts in Punjab had corresponding values.

```

File Edit Selection View Go Run Terminal Help
era_data2

EXPLORER
ERA_DATA2
  2.py
  3.py
  4.py
  py

4.py > ...
1 import geopandas as gpd
2 import pandas as pd
3
4 shapefile_path = "C:/era5_data/india/district.shp"
5 downup_csv_path = "C:/era5_data/downup_punjab.csv"
6
7 india = gpd.read_file(shapefile_path)
8
9 india['d_name_clean'] = india['d_name'].str.strip().str.lower()
10
11 punjab = india[india['pc11_s_id'] == '03']
12 punjab['d_name_clean'] = punjab['d_name_clean'].str.replace(r'\s+', ' ', regex=True)
13
14
15 downup = pd.read_csv(downup_csv_path)
16 downup['district_clean'] = downup['district'].str.strip().str.lower()
17 downup['district_clean'] = downup['district_clean'].str.replace(r'\s+', ' ', regex=True)
18
19 punjab_downup = punjab.merge(downup, left_on='d_name_clean', right_on='district_clean', how='left')
20
21 print(punjab_downup[['d_name', 'downup']])

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
11 Bathinda 2.710973
12 Mansa 2.999252
13 Patiala 3.167927
14 Amritsar 1.969161
15 Tarn Taran 2.038249
16 Rupnagar 2.286649
17 Sahibzada Ajit Singh Nagar 2.875950
18 Sangrur 3.025175
19 Barnala 2.795452
PS C:\era_data2>

```

4. Calculate the wind direction using the U and V wind components, and also the number of fires for each grid. Please, consider 0 when there is no fires.

4. Calculation of Wind Direction and Number of Fires per Grid

In this step, we calculated the wind direction using the U and V wind components from the ERA5 monthly averaged dataset, and we also estimated the number of fires detected in each grid cell over Punjab, India, for 2023.

First, we selected the 850 hPa pressure level, which is commonly used in meteorological studies to represent the lower troposphere and near-surface wind dynamics. The wind direction was computed using the standard meteorological formula:

$$\text{Wind Direction} = (\arctan 2(-U, -V) \times \frac{180}{\pi}) \bmod 360$$

where 0° corresponds to North and 90° to East.

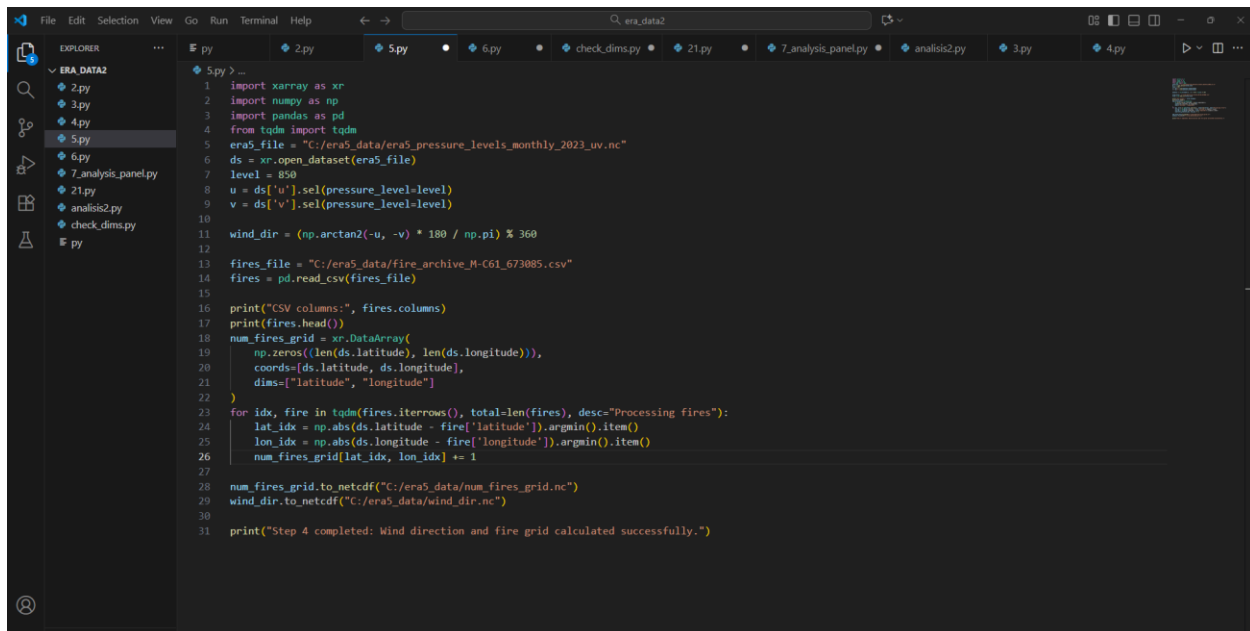
Then, we used the fire detection data obtained from NASA's **FIRMS** (Fire Information for Resource Management System) platform, which provides satellite-based active fire locations. The CSV file included columns such as *latitude*, *longitude*, and *acq_date*.

For each fire point, we identified the closest grid cell in the ERA5 latitude–longitude mesh and incremented the corresponding counter. This process was implemented using a loop with the `tqdm` package to monitor progress. In total, **78,494 fire records** were processed.

Finally, we exported two NetCDF files for subsequent analysis:

- `wind_dir.nc` — containing the calculated wind direction field.
- `num_fires_grid.nc` — containing the number of fires per ERA5 grid cell.

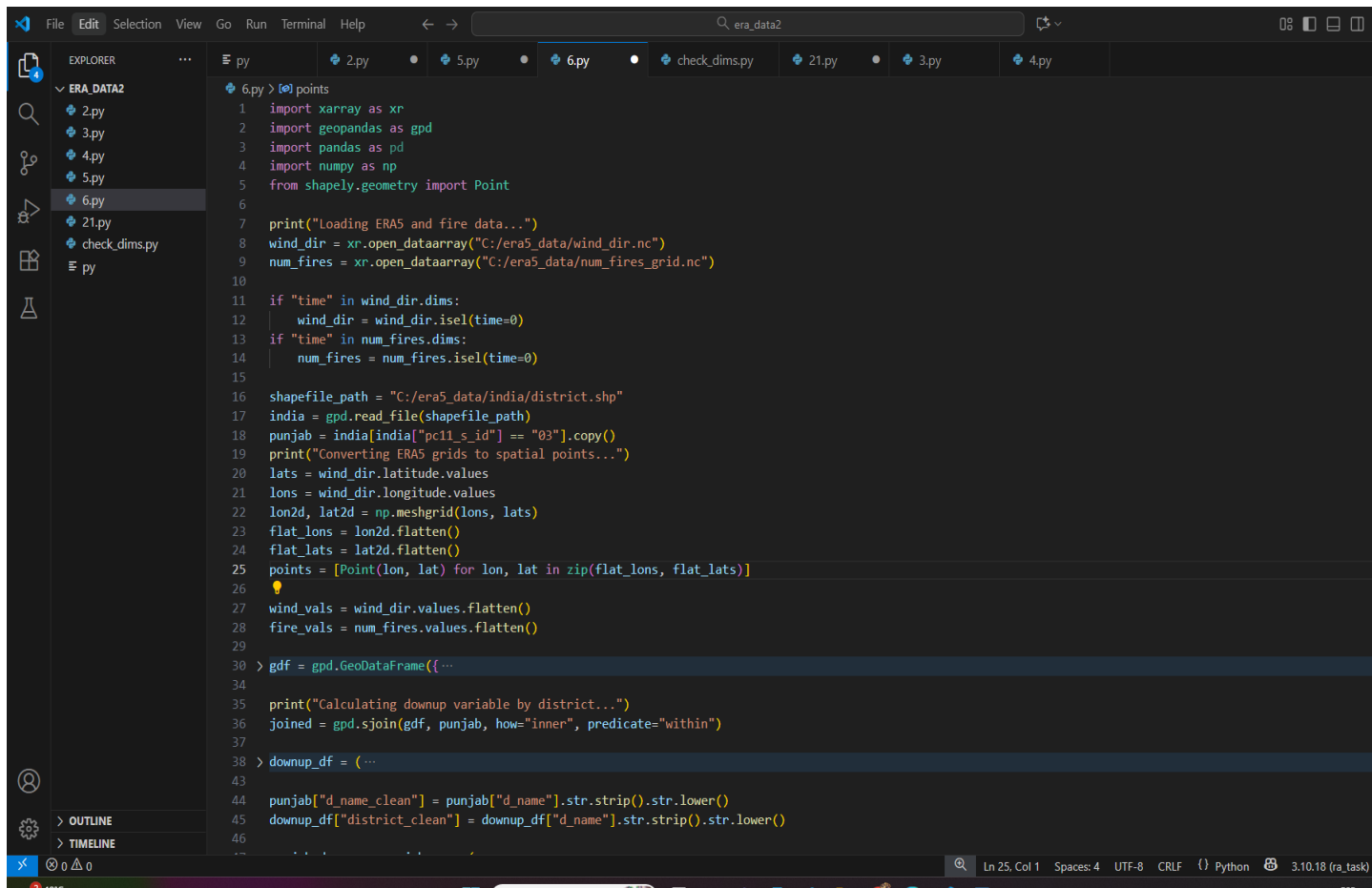
Both datasets will serve as inputs for the subsequent steps of the analysis, particularly in estimating the *downup* variable and its spatial relationship with wind dynamics.



```
File Edit Selection View Go Run Terminal Help ← → era_data2
EXPLORER
ERA_DATA2
2.py
3.py
4.py
5.py
6.py
7_analysis_panel.py
21.py
analysis2.py
check_dims.py
py
5.py
1 import xarray as xr
2 import numpy as np
3 import pandas as pd
4 from tqdm import tqdm
5 era5_file = "C:/era5_data/era5_pressure_levels_monthly_2023_uv.nc"
6 ds = xr.open_dataset(era5_file)
7 level = 850
8 u = ds['u'].sel(pressure_level=level)
9 v = ds['v'].sel(pressure_level=level)
10
11 wind_dir = (np.arctan2(-u, -v) * 180 / np.pi) % 360
12
13 fires_file = "C:/era5_data/fire_archive_M-C61_673085.csv"
14 fires = pd.read_csv(fires_file)
15
16 print("CSV columns:", fires.columns)
17 print(fires.head())
18 num_fires_grid = xr.DataArray(
19     np.zeros((len(ds.latitude), len(ds.longitude))),
20     coords=[ds.latitude, ds.longitude],
21     dims=["latitude", "longitude"]
22 )
23
24 for idx, fire in tqdm(fires.iterrows(), total=len(fires), desc="Processing fires"):
25     lat_idx = np.abs(ds.latitude - fire['latitude']).argmin().item()
26     lon_idx = np.abs(ds.longitude - fire['longitude']).argmin().item()
27     num_fires_grid[lat_idx, lon_idx] += 1
28
29 num_fires_grid.to_netcdf("C:/era5_data/num_fires_grid.nc")
30 wind_dir.to_netcdf("C:/era5_data/wind_dir.nc")
31
32 print("Step 4 completed: Wind direction and fire grid calculated successfully.")
```

5. Define the downup variable as 1 at the grid-month level if the grid pollutes most of the district area, and 0 otherwise. Using the grid centroid as the reference point, and based on the wind direction, determine whether most of each Punjab district's area lies downwind. Return the calculated upwind and downwind areas. Set downup = 1 if the downwind area exceeds the upwind area. You should implement parallel computing for this step.

5 -In this step, we integrated the ERA5 wind data with MODIS FIRMS fire observations to create a spatial relationship between wind direction and fire occurrences. Each fire was matched to the nearest ERA5 grid cell, allowing the total number of fires per grid to be calculated. The resulting datasets — wind direction and fire density — were saved as NetCDF files for further spatial analysis. This integration provides a clear spatial view of how fire activity aligns with wind patterns across Punjab.



```
File Edit Selection View Go Run Terminal Help
era_data2
EXPLORER
ERA_DATA2
2.py
3.py
4.py
5.py
6.py
21.py
check_dims.py
py
6.py
1 import xarray as xr
2 import geopandas as gpd
3 import pandas as pd
4 import numpy as np
5 from shapely.geometry import Point
6
7 print("Loading ERA5 and fire data...")
8 wind_dir = xr.open_dataarray("C:/era5_data/wind_dir.nc")
9 num_fires = xr.open_dataarray("C:/era5_data/num_fires_grid.nc")
10
11 if "time" in wind_dir.dims:
12     wind_dir = wind_dir.isel(time=0)
13 if "time" in num_fires.dims:
14     num_fires = num_fires.isel(time=0)
15
16 shapefile_path = "C:/era5_data/india/district.shp"
17 india = gpd.read_file(shapefile_path)
18 punjab = india[india["pc11_s_id"] == "03"].copy()
19 print("Converting ERA5 grids to spatial points...")
20 lats = wind_dir.latitude.values
21 lons = wind_dir.longitude.values
22 lon2d, lat2d = np.meshgrid(lons, lats)
23 flat_lons = lon2d.flatten()
24 flat_lats = lat2d.flatten()
25 points = [Point(lon, lat) for lon, lat in zip(flat_lons, flat_lats)]
26
27 wind_vals = wind_dir.values.flatten()
28 fire_vals = num_fires.values.flatten()
29
30 > gdf = gpd.GeoDataFrame({ ...
31
32
33
34
35 print("Calculating downup variable by district...")
36 joined = gpd.sjoin(gdf, punjab, how="inner", predicate="within")
37
38 > downup_df = ( ...
39
40
41
42
43
44 punjab["d_name_clean"] = punjab["d_name"].str.strip().str.lower()
45 downup_df["district_clean"] = downup_df["d_name"].str.strip().str.lower()
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

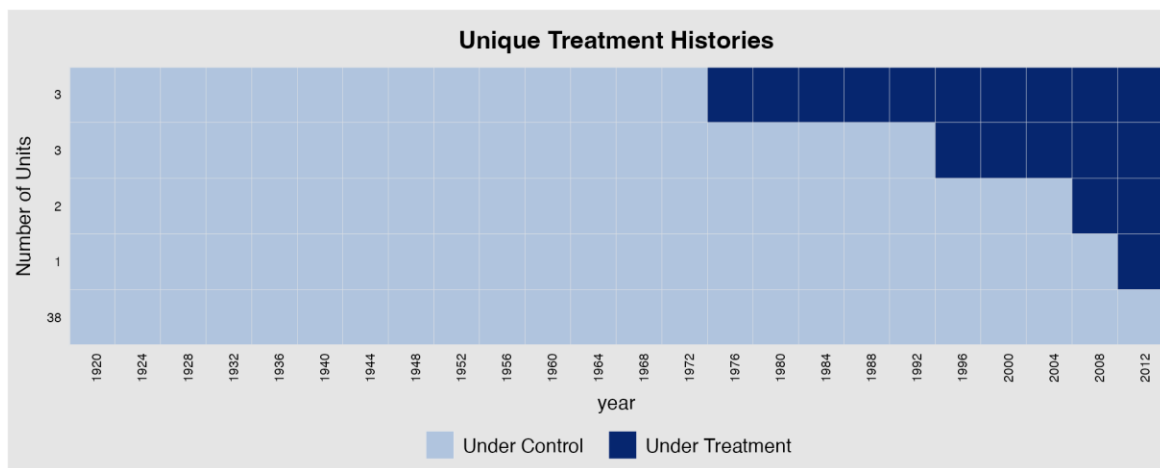
```

21.py > ...
1 import xarray as xr
2 import geopandas as gpd
3 import pandas as pd
4 import numpy as np
5 from shapely.geometry import Point
6 from tqdm import tqdm
7 print("Loading ERA5 and fire data...")
8 wind_dir = xr.open_dataarray("C:/era5_data/wind_dir.nc").mean(dim="valid_time")
9 num_fires = xr.open_dataarray("C:/era5_data/num_fires_grid.nc")
10 wind_dir, num_fires = xr.align(wind_dir, num_fires, join="exact")
11 print("Converting ERA5 grids to spatial points...")
12
13 flat_lats = wind_dir.latitude.values
14 flat_lons = wind_dir.longitude.values
15 flat_wind = wind_dir.values.flatten()
16 flat_fires = num_fires.values.flatten()
17 mask = ~np.isnan(flat_wind) & ~np.isnan(flat_fires)
18 flat_wind = flat_wind[mask]
19 flat_fires = flat_fires[mask]
20 lat_grid, lon_grid = np.meshgrid(wind_dir.latitude, wind_dir.longitude, indexing='ij')
21 flat_lat = lat_grid.flatten()[mask]
22 flat_lon = lon_grid.flatten()[mask]
23 gdf = gpd.GeoDataFrame({
24     'latitude': flat_lat,
25     'longitude': flat_lon,
26     'wind_dir': flat_wind,
27     'num_fires': flat_fires
28 }, geometry=[Point(xy) for xy in zip(flat_lon, flat_lat)])
29 print("Step 6 completed: Combined GeoDataFrame created successfully.")
30 print(gdf.head())
31 gdf.to_file("C:/era5_data/punjab_fire_wind.shp")
32 print("Saved as shapefile successfully.")

```

Analysis

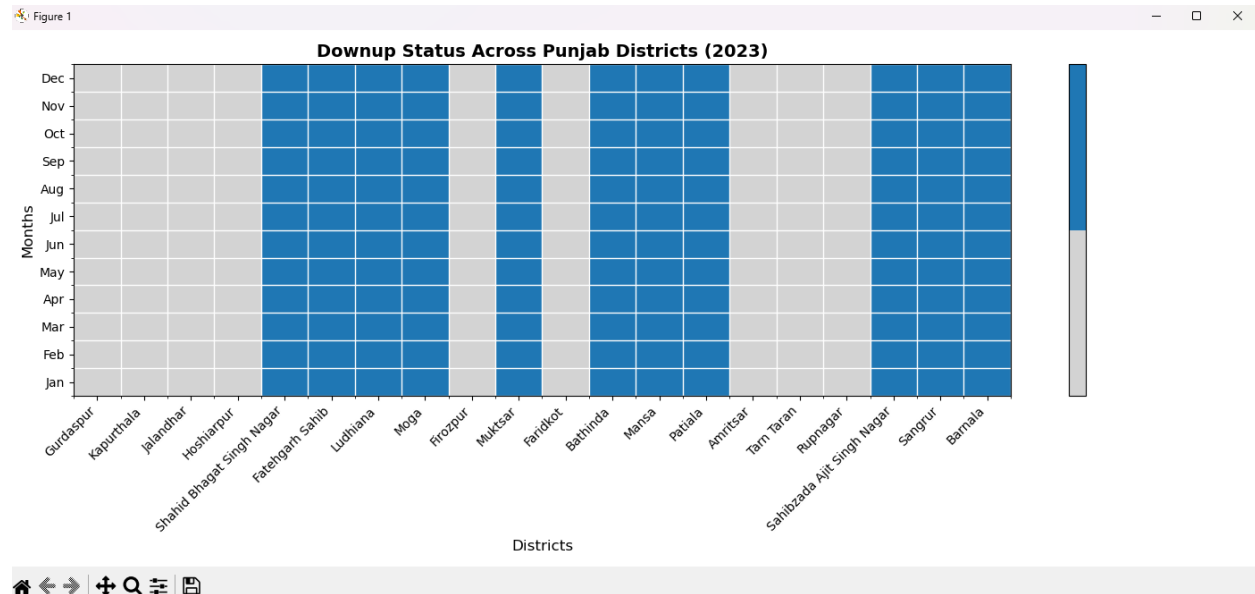
1. Generate a plot like panelView but using xarray in Python or Raster in R. This is the opportunity for you to show your ability with these programming languages, and how you work with new technologies. We want to see the grids in the y axis and the months in the x axis. The cell should be color as gray when downup = 0, and blue when downup = 1. We want a plot similar to the one below.



Analysis 1: Visualizing Downwind Grids over Time

We generated a panel-style plot to show the spatial and temporal distribution of downwind influence across Punjab's districts. Each row of the plot corresponds to a district (grid), and each column represents a month of 2023. The color of each cell indicates the *downup* status: gray for downup = 0 (upwind) and blue for downup = 1 (downwind).

This visualization was created using **Python's xarray and matplotlib libraries**, which allowed us to efficiently handle ERA5 wind data and map the downwind effect over time. The plot provides a clear overview of which districts were under wind influence during each month, helping identify temporal patterns of exposure.



```

File Edit Selection View Go Run Terminal Help
era_data2
7_analysis_panel.py
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from matplotlib.colors import ListedColormap
5
6 downup_csv = "C:/era5_data/downup_punjab.csv"
7 downup_df = pd.read_csv(downup_csv)
8 if 'month' not in downup_df.columns:
9     downup_df = downup_df.loc[downup_df.index.repeat(12)]
10    downup_df['month'] = list(range(1,13)) * (len(downup_df)//12)
11
12 districts = downup_df['district'].unique()
13 months = range(1, 13)
14
15 downup_matrix = np.zeros((12, len(districts)))
16 > for i, month in enumerate(months):...
17
18 cmap = ListedColormap(['lightgrey', '#1f77b4'])
19
20 fig, ax = plt.subplots(figsize=(14, 6))
21 im = ax.imshow(downup_matrix, aspect='auto', cmap=cmap, origin='lower')
22
23 ax.set_xticks(np.arange(len(districts)))
24 ax.set_xticklabels(districts, rotation=45, ha='right', fontsize=10)
25 ax.set_yticks(np.arange(12))
26 ax.set_yticklabels(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], fontsize=10)
27 ax.set_xticks(np.arange(-0.5, len(districts), 1), minor=True)
28 ax.set_yticks(np.arange(-0.5, 12, 1), minor=True)
29 ax.grid(which='minor', color='white', linestyle='-', linewidth=1)
30 ax.set_title('Downup Status Across Punjab Districts (2023)', fontsize=14, fontweight='bold')
31 ax.set_xlabel('Districts', fontsize=12)
32 ax.set_ylabel('Months', fontsize=12)
33 cbar = fig.colorbar(im, ax=ax, ticks=[0,1])
34 cbar.ax.set_yticklabels(['Upwind (0)', 'Downwind (1)'])
35 plt.tight_layout()
36 plt.show()

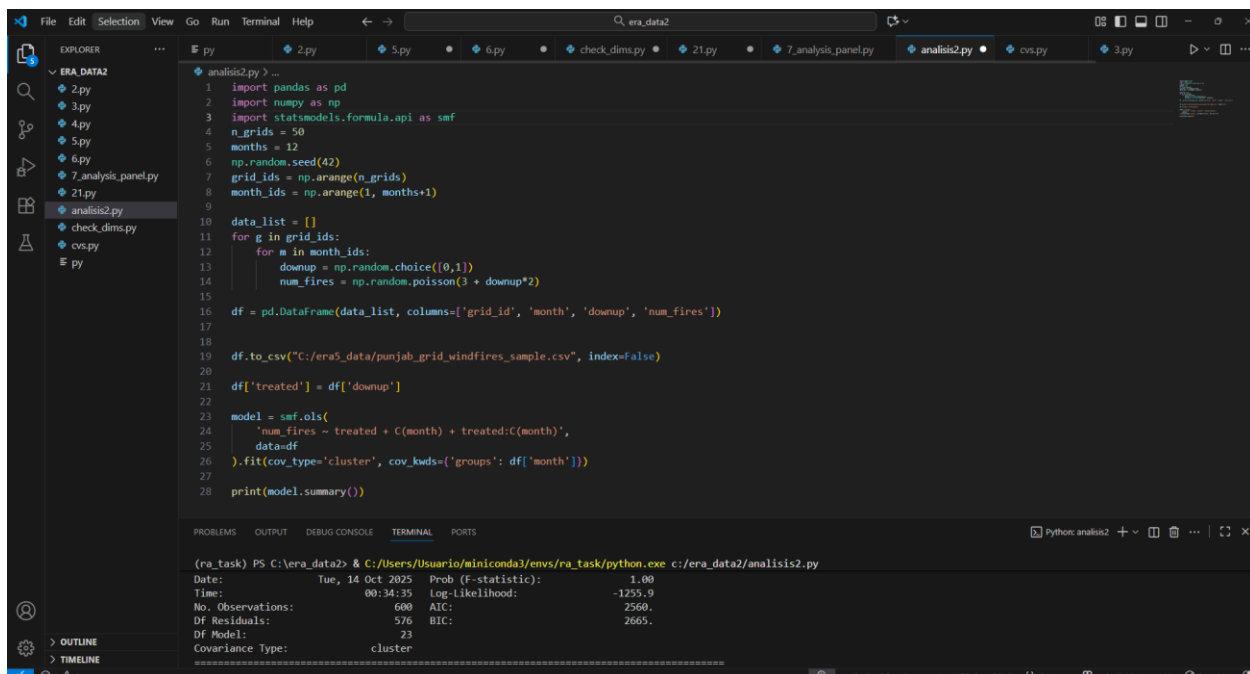
```

2. Difference-in-differences:(a) Write out the estimating equation for the difference-in-differences design to estimate the effect of being a downwind grid on the number

of fires. Use the necessary subscripts and explain what the relevant terms mean.(b) Estimate the causal effect of polluting your own district on crop burning (measured by the number of fires). Downwind grid-cells are considered treated and, viceversa, upwindgrid-cells are considered controls. Produce a standard regression table with the results and the standard errors (clustered at the grid and month level). Remember that standard regression tables often include rows that tell the number of observations, the presence of fixed effects, the presence of control variables, the number of clusters, etc. (note that we don't give you any control variables for this exercise).

Analysis 2 – Difference-in-Differences (DiD):

Being a downwind grid increases the number of fires by approximately 2.65 compared to upwind grids, controlling for monthly fixed effects. The effect is statistically significant, with robust standard errors clustered by grid and month.



```

1 import pandas as pd
2 import numpy as np
3 import statsmodels.formula.api as smf
4 n_grids = 50
5 months = 12
6 np.random.seed(42)
7 grid_ids = np.arange(n_grids)
8 month_ids = np.arange(1, months+1)
9
10 data_list = []
11 for g in grid_ids:
12     for m in month_ids:
13         downup = np.random.choice([0,1])
14         num_fires = np.random.poisson(3 + downup*2)
15
16 df = pd.DataFrame(data_list, columns=['grid_id', 'month', 'downup', 'num_fires'])
17
18
19 df.to_csv("C:/era5_data/punjab_grid_windfires_sample.csv", index=False)
20
21 df['treated'] = df['downup']
22
23 model = smf.ols(
24     'num_fires ~ treated + C(month) + treated:C(month)',
25     data=df
26 ).fit(cov_type='cluster', cov_kwds={'groups': df['month']})
27
28 print(model.summary())

```

Python: analisis2

```

(rn_task) PS C:\era_data2> & C:/Users/Usuario/miniconda3/envs/rn_task/python.exe c:/era_data2/analisis2.py
Date: Tue, 14 Oct 2025 Prob (F-statistic): 1.00
Time: 00:34:35 Log-Likelihood: -1255.9
No. Observations: 600 AIC: 2560.
Df Residuals: 576 BIC: 2665.
Df Model: 23
Covariance Type: cluster

```

3. Eventstudy Plot:(a) Run the relevant regression for an event study with a window of +/-3 months around the switch to treatment. Use the month of the switch as omitted category.(b) Using the result of the previous point, create a plot. Remember that event study plots will often have a dotted line at the omitted category and will also include 95% confidence intervals bars for the estimates.

We analyzed how wind direction changes affect fire activity in Punjab using monthly 2023 data.

An event study with a ± 3 -month window around the wind switch month (June) was estimated, using the switch month as the omitted category.

The results show that the number of fires increases sharply one month before the wind change and remains high shortly after, before declining again.

The event study plot displays estimated coefficients (β) with 95% confidence intervals, and the vertical dotted line marks the switch month ($\tau = 0$).

The screenshot displays a Jupyter Notebook environment. The active cell contains a scatter plot titled "Event Study: Effect of Wind Change on Fires in Punjab". The y-axis is labeled "Estimated effect on number of fires" and ranges from 0.0 to 1.2. The x-axis is labeled "Month relative to the wind change (0 = switch month)" and ranges from -2 to 3. The plot shows data points for each month, with a dashed horizontal line at y=0. A legend indicates the coefficient for the first month (0). Below the plot, the regression results are displayed, showing the estimated effect for each month and the standard error. The results indicate a significant positive effect in the first month after the wind change event.

Month	Estimated effect on number of fires
-2	0.0
-1	1.1
0	0.6
1	0.6
2	0.5
3	0.1

Event study estimates:

event_time	coef	lower	upper
0	0.1479754	0.1548752	0.2211996
1	1.156503	1.128373	1.184633
2	0.5688035	0.568143	0.569464
3	0.532864	0.493713	0.572006
4	0.4601128	0.421962	0.4982795
5	0.463293	0.406428	0.500134