

Weather & Corona

Michele Coscia

First Year Project #1

February 3rd, 2022

Outline

- Logistics
- Project Background
- Kickoff Q&A

Logistics

Skills for Project 1

- Geospatial data wrangling
- Choropleths
- Multiple hypothesis testing
- Multivariate regression

Work Organization: GIT

- Study well the Github tutorial from previous lecture
- Follow best practices
- Remember: code & docs, no data (use .gitignore)

Project Folder Suggestion

- Data
 - Raw (Original, immutable)
 - Interim (Intermediate datasets)
 - Processed (Final datasets for the analysis)
- Notebooks (Polished “start to finish” code for report)
- References (Data documentation, manuals, literature)
- Reports (Your outputs for the project: the final handout, slides, etc)
 - Figures
- Code (Working directory with all the code organized in logical subfolders)

Exercises

- Doing the exercise = working on the project
- The tasks will help you with your report
- Follow them to get the basics down so you have more time for the interesting stuff

Hand-in

- gitlog.txt
- code.zip
 - No data, unless it's 3rd party
- report.pdf
 - Formatting instructions on LearnIT

Report

- 1) Introduction (context and motivation)
- 2) Data (with cleaning steps, including missing data)
- 3) Results & Discussion
- 4) Limitations (short-coming(s) of your methodology/data)
- 5) Concluding remarks and future work
- 6) Disclosure statement (optional)

Tasks

- 0) Data filtering & cleaning
- 1) Single variable analysis
- 2) Associations
- 3) Map visualization
- 4) Open question

Exam

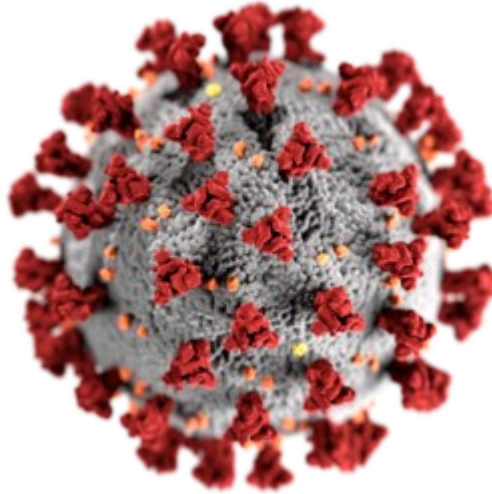
- 30 minutes per group
- Taken as a group all together
- Project Presentation
- Questions: on project AND on theory
- March 8th & 10th
- Only feedback, no intermediate points (sorry)

Lecture Plan

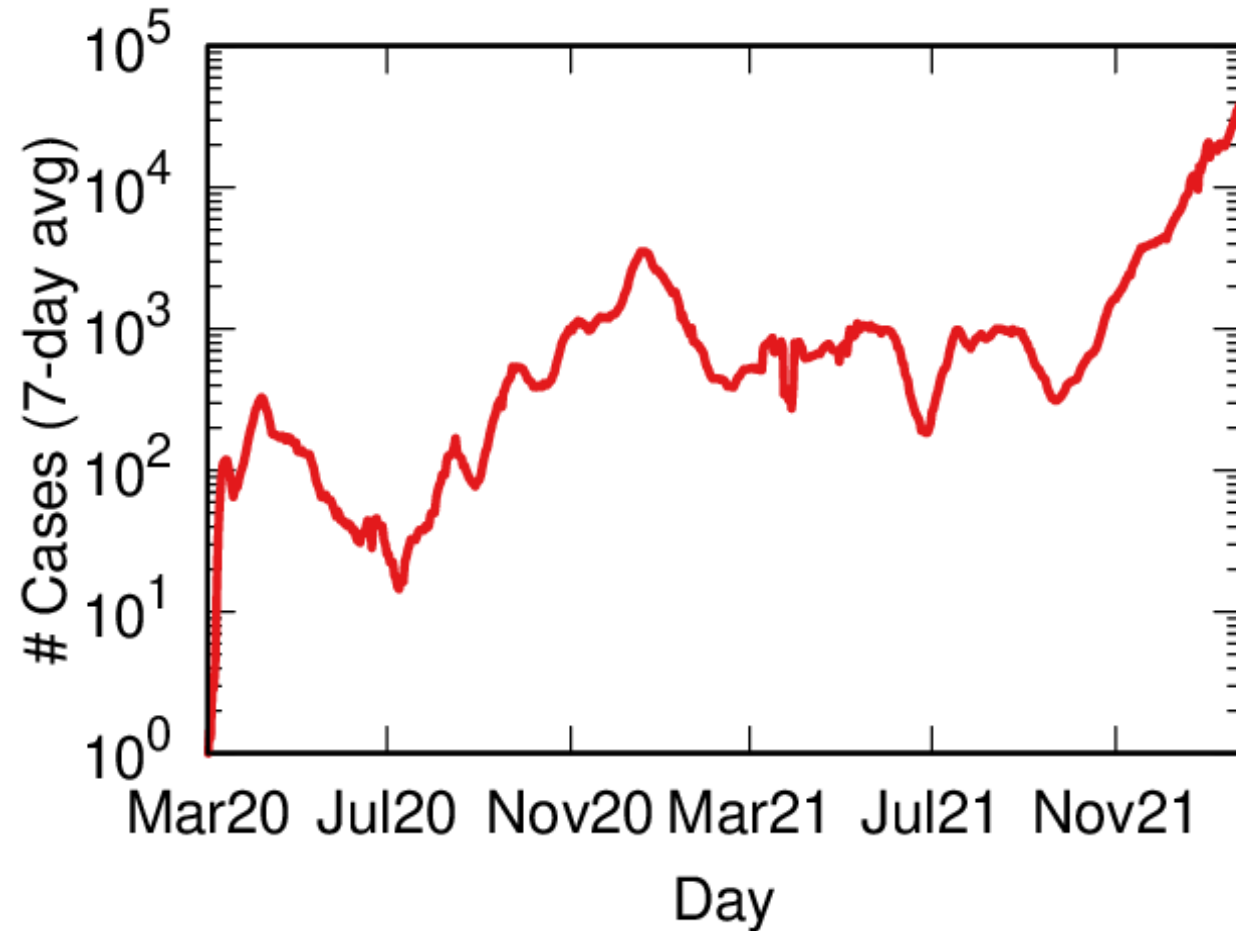
- 1) **(Today) Intro**
- 2) (February 10th 8th) Geospatial Basics
- 3) (February 15th) Estimating Associations
- 4) (February 17th) Multivariate Regression
- 5) (February 22nd) Interventions
- 6) (February 24th) Project Run Through
- 7) (March 1st) Q&A – Open Supervision
- 8) (March 3rd) Q&A – Open Supervision

Project Background

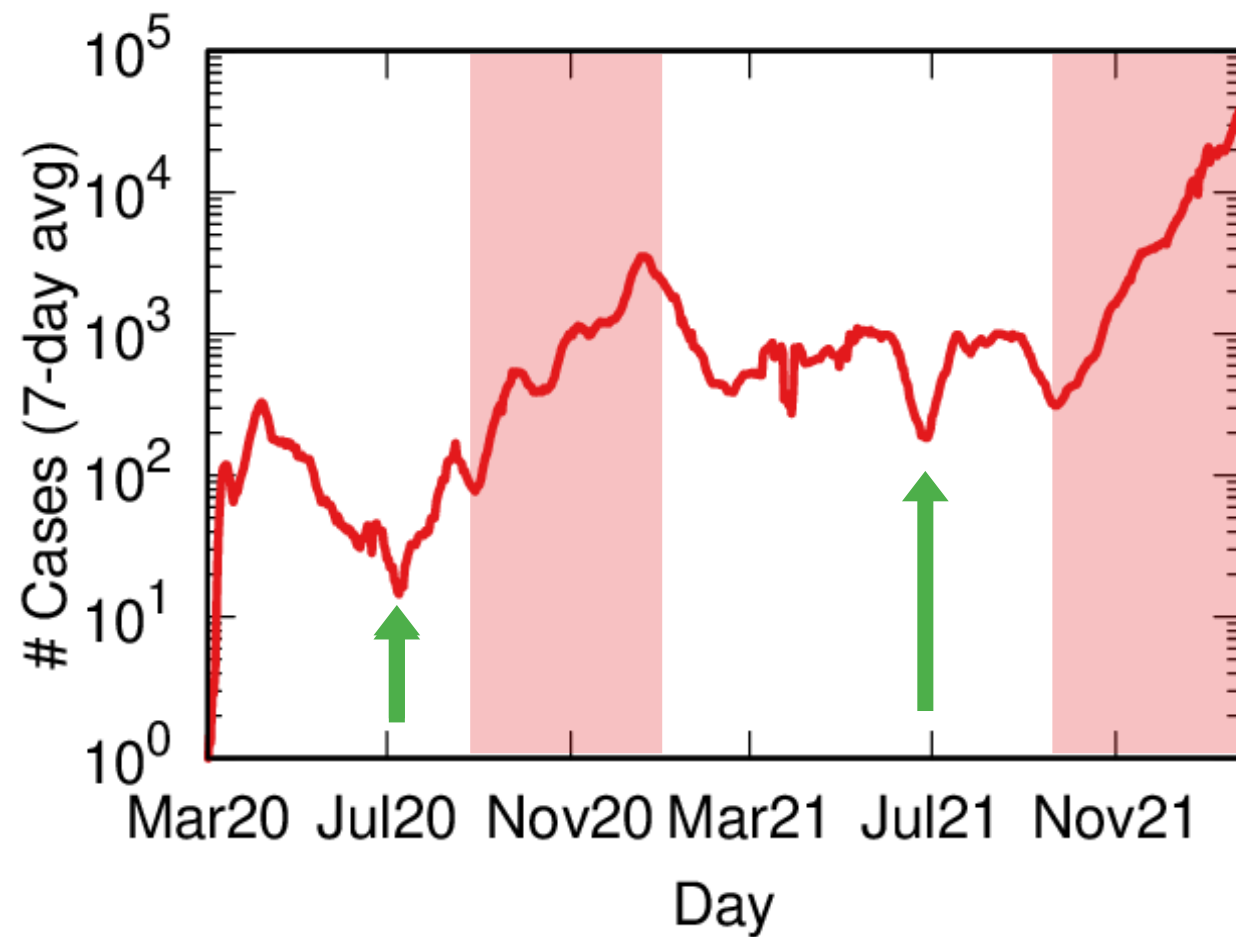
No Need to Introduce you



Huge Disruptions



Mmmh...



Seasonal Diseases

- Cold
- Flu
- Covid?

Let's discuss:
what could be the cause?

Congratulations, you have a working hypothesis
for your task 4 :-)

Quick Look at Weather Data

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The file explorer shows a directory with files named pr01_e01, pr01_e02, pr02_e98, and pr02_e99. The code editor shows a Python script that loads weather data from a CSV file, checks for missing values, filters data by country (iso3166-2), and calculates summary statistics by region. The output of the script is a table with 28 columns and 5 rows of data.

```
[1]: # We start by loading the libraries we're going to use
import pandas as pd

[2]: # Here we store the paths to the data files
# Remember to change this in case your folder structure is different
weather_df = pd.read_csv("../data/raw/weather/weather.csv")

# First task: the number of observation and of variables is the shape of the dataframe
print(weather_df.shape)

(20220, 9)

[3]: # Second task: check for missing values. We start by asking if there's any NA
print(weather_df.isna().any().any()) # There aren't any! We're so lucky, this never happens normally :-))

# Now we check if there's a variable that needs special treatment
# In this case, temperature is reported in kelvins, but we'd like it in celsius instead
# Careful if you run this cell more than once!
weather_df["TemperatureAboveGround"] = weather_df["TemperatureAboveGround"] - 273.15

# Third task, filter the dataframe to contain only the data bout the country of interest
# We do so by checking the iso3166-2 code of the country: it needs to start with the iso3166 code!
weather_df = weather_df[weather_df["iso3166-2"].str.startswith("DK")]

# Fourth task: summary statistics by region
# Standard pandas groupby here...
weather_by_region = weather_df.groupby(by = "iso3166-2").agg(["min", "mean", "median", "max"])
weather_by_region
```

		RelativeHumiditySurface				SolarRadiation				Surfacepressure			Totalprecipitation			UVIndex					
		min	mean	median	max	min	mean	median	max	min	mean	...	median	max	min	mean	median	max	min	mean	media
iso3166-2																					
DK-81	46.166826	81.055723	82.495994	98.183444	0.000000	6.998678e+06	4.519083e+06	2.407046e+07	2.342712e+06	2.424567e+06	...	0.000765	0.018153	0.0	14.137697	9.979405	44.536232	0.977651	4.789387	4.44991	
DK-82	49.033149	81.566104	83.703552	98.192610	0.005224	6.660075e+06	3.965236e+06	2.392633e+07	2.342463e+06	2.420853e+06	...	0.000803	0.025882	0.0	14.285644	9.993017	44.405486	0.804547	4.128445	3.82101	
DK-83	50.835916	80.914819	82.252746	96.980902	0.000000	6.788604e+06	4.503614e+06	2.403996e+07	2.346761e+06	2.424560e+06	...	0.000773	0.024844	0.0	15.077168	10.867238	43.914882	1.167244	4.548929	4.23694	
DK-84	50.049442	78.592708	79.835315	96.115837	85.014818	6.912021e+06	4.686215e+06	2.283164e+07	2.359255e+06	2.428423e+06	...	0.000835	0.041373	0.0	14.931567	11.106236	43.076212	0.839867	4.317130	3.91872	
DK-85	50.719733	79.364675	80.494326	96.581381	270.216521	7.163466e+06	4.990226e+06	2.349346e+07	2.355432e+06	2.428640e+06	...	0.000696	0.026545	0.0	15.076703	11.327366	43.986360	1.219587	4.478090	4.19229	

5 rows x 28 columns

```
[4]: # To plot, we use pandas built-in support for matplotlib
# Any data series can be plotted by calling ".plot.<plottype>()"
```

Q&A