

---

**Visualising the spread of COVID-19 across the World**

# **COVID-19 VISUALISATIONS ACROSS DIFFERENT BRANCHES OF SOCIETY**

[X19147996@STUDENT.NCIRL.IE](mailto:X19147996@STUDENT.NCIRL.IE)

[X19180080@STUDENT.NCIRL.IE](mailto:X19180080@STUDENT.NCIRL.IE)

[X19182732@STUDENT.NCIRL.IE](mailto:X19182732@STUDENT.NCIRL.IE)

[X19194137@STUDENT.NCIRL.IE](mailto:X19194137@STUDENT.NCIRL.IE)

**HITESH MADHUKAR PATIL**

**VIKAS KISHANRAO THAMKE**

**VISHAL SHAKYA**

**AMANDEEP SINGH**

REDDIT DATASET

STOCK MARKET DATASET

TWITTER DATASET

COVID-19 WORLD DATASET

---

# **WHAT IS THE AIM?**

**TO PLOT A VISUALISATION THAT  
CORRELATES THE WORLD-WIDE COVID-19  
SPREAD TO THE VARIATIONS IN STOCK  
MARKETS AND INTERACTIONS ON  
DIFFERENT SOCIAL MEDIA PLATFORMS**

---

---

# **WHAT DATA IS SELECTED?**

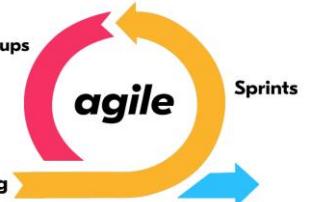
- 1. Social Media Platform - TWITTER — VISHAL SHAKYA**
  - 2. Social Media Platform - REDDIT — HITESH MADHUKAR PATIL**
  - 3. Stock Market - 7 Stocks from different sectors — VIKAS KISHANRAO THAMKE**
  - 4. COVID-19 World-wide infection spread — AMANDEEP SINGH**
-

# Environment Setup

- Project Methodology: Agile Sprint
- Planning Tool: Microsoft Planner
- Cloud: AWS EC2 Debian Buster
- Databases:
  - Unstructured: MongoDB
  - Structured: PostgreSQL
- Repository: GitHub



PostgreSQL



# Dataset Selection

- Yahoo Finance
  - Legitimacy Verification
- Stock Market
  - **Airline:** American Airline
  - **Automotive:** General Motors
  - **Pharmaceutical:** Johnson & Johnson
  - **Construction:** Jacobs Engineering Group
  - **Chemicals:** Albemarle Corporation
  - **Entertainment:** The Walt Disney Company
  - **Telecommunication:** AT & T



# Web Scrapping

- Programming Language: Python
- Automation: Selenium Using Chromium
- Webpage Parsing: BeautifulSoup, Regex



**Selenium**



# Mongo dB Data

Data in json format ready to upload on  
Mongo dB

```
Started: General Motors
{'Date': 'Apr 24 2020', 'Values': {'Open': '21.71', 'High': '22.24', 'Low': '21.54', 'Close': '21.95', 'Adj Close': '21.95',
'Volume': '11990500'}}}
{'Date': 'Apr 23 2020', 'Values': {'Open': '21.55', 'High': '22.06', 'Low': '21.45', 'Close': '21.52', 'Adj Close': '21.52',
'Volume': '10673000'}}}
{'Date': 'Apr 22 2020', 'Values': {'Open': '21.65', 'High': '21.78', 'Low': '21.07', 'Close': '21.30', 'Adj Close': '21.30',
'Volume': '9553600'}}}
{'Date': 'Apr 21 2020', 'Values': {'Open': '21.27', 'High': '21.89', 'Low': '20.98', 'Close': '21.24', 'Adj Close': '21.24',
'Volume': '13507300'}}}
{'Date': 'Apr 20 2020', 'Values': {'Open': '21.72', 'High': '22.64', 'Low': '21.44', 'Close': '22.38', 'Adj Close': '22.38',
'Volume': '15912400'}}}
{'Date': 'Apr 17 2020', 'Values': {'Open': '21.92', 'High': '22.54', 'Low': '21.83', 'Close': '22.48', 'Adj Close': '22.48',
'Volume': '17166300'}}}
{'Date': 'Apr 16 2020', 'Values': {'Open': '21.64', 'High': '21.65', 'Low': '20.56', 'Close': '20.87', 'Adj Close': '20.87',
'Volume': '11501700'}}}
{'Date': 'Apr 15 2020', 'Values': {'Open': '21.92', 'High': '22.27', 'Low': '21.47', 'Close': '21.66', 'Adj Close': '21.66',
'Volume': '12760900'}}}
{'Date': 'Apr 14 2020', 'Values': {'Open': '23.60', 'High': '23.77', 'Low': '22.67', 'Close': '22.98', 'Adj Close': '22.98',
'Volume': '12654700'}}}
```

# PostgreSQL Data

Data in structured form fetched from PostgreSQL

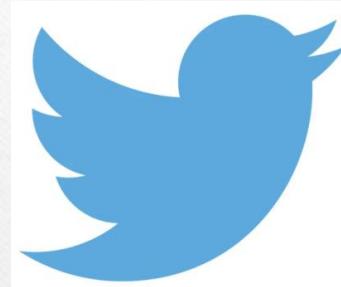
	date_time	gm_open	gm_close	johnson_open	johnson_close	jacobs_open	jacobs_close	albemarle_open	albemarle_close	disney_open	disney_close	at
0	2019-10-28	37.21	36.64	128.62	129.18	94.33	94.41	62.75	63.74	131.45	130.53	
1	2019-10-29	37.60	38.21	129.70	129.12	93.80	94.29	63.45	63.03	130.50	129.48	
2	2019-10-30	38.26	37.91	133.30	132.84	94.48	94.84	62.86	62.67	129.69	129.60	
3	2019-10-31	37.62	37.16	132.38	132.04	94.53	93.58	62.31	60.74	129.53	129.92	
4	2019-11-01	37.21	37.97	132.05	131.20	94.25	96.07	61.46	64.49	130.99	132.75	

# Visualizations

- Python Library: Seaborn



# TWITTER



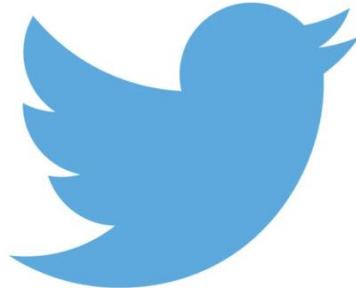
---

SOCIAL MEDIA PLATFORM

Vishal Shakya  
x19182732@student.ncirl.ie

# DATA COLLECTION

---

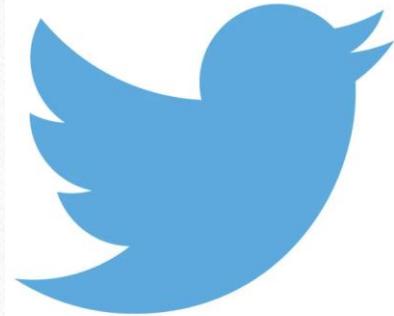


- Data returned was only 7 days prior to Today's date.
- Limit of 180 queries per 15 minutes.
- Hashtags related to COVID-19.
- Data from Kaggle with the IDs.
- IDs from 22nd January, 2020 till 13<sup>th</sup> April, 2020.
- Collected number of tweets related to that particular Hashtag.

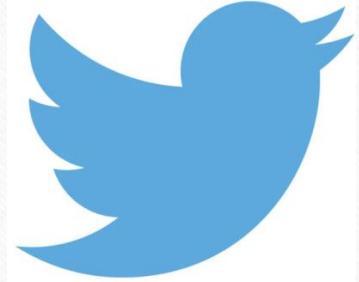
# DATA PROCESSING

---

- Collected unstructured data.
- Uploaded it to MongoDB.
- MongoDB to PostgreSQL.
- Visualizations.



# VISUALIZATIONS



## • HASHTAGS WITHIN THE TWEETS

#CORONAVIRUS

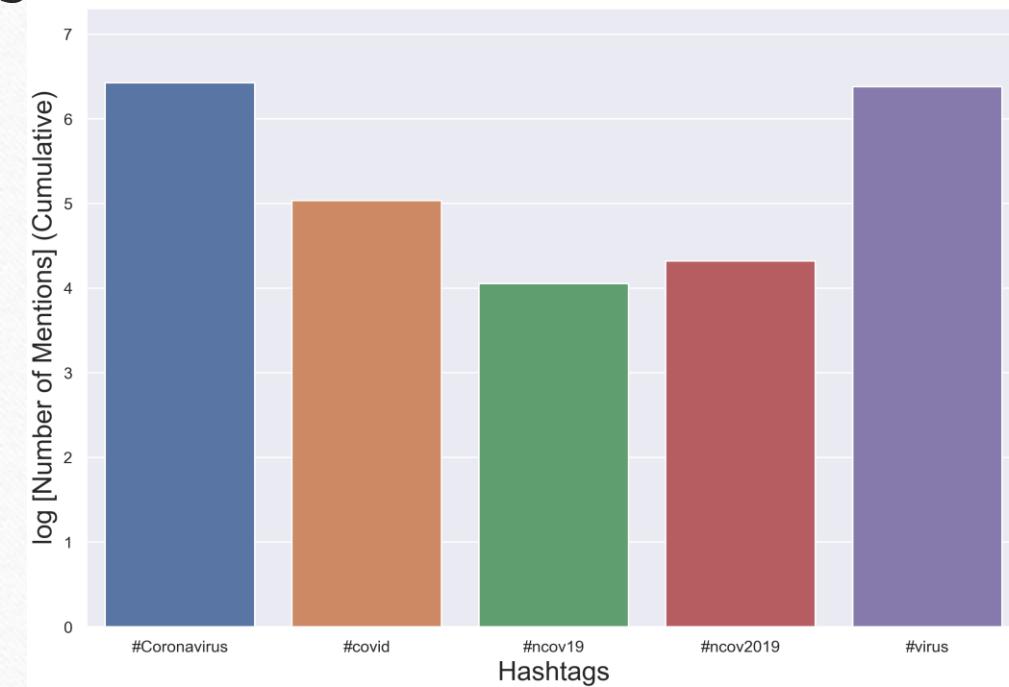
#COVID

#NCOV19

#NCOV2019

#VIRUS

TWITTER





# Reddit

Reddit post scraping for DAP on coronavirus pandemic, Covid19,  
Worldnews, Lockdown

# Web scraping and APIs

- ▶ fetched data **PRAW library**.
- ▶ Unable to fetch data older than **7 days**.
- ▶ Archive files.
- ▶ Web scraping on **old.reddit.com**.
- ▶ Web scraping on **Parsehub** software.
- ▶ **Json → MongoDB → Postgres**

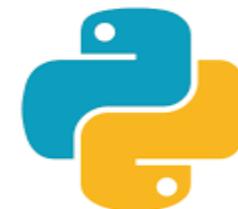
**Python + PRAW**

**Streaming  
Reddit (p.3)**



# Stored data in Json

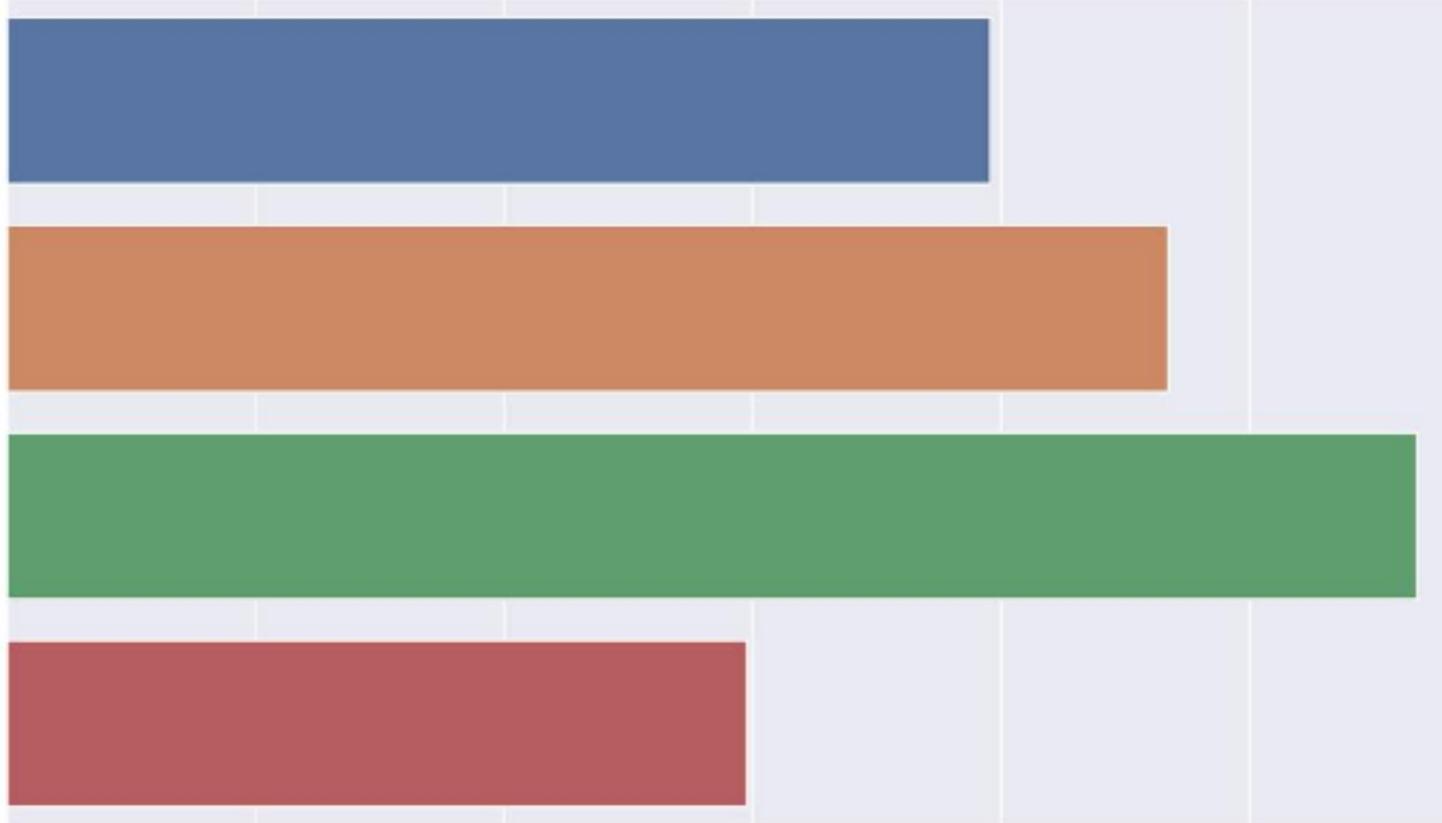
- ▶ Uploaded unstructured data on **MangoDB**
- ▶ Uploaded data to **Postgres**
- ▶ Data transformed on **Python**



# REDDIT

Categories

Contro Posts



---

**Visualising the spread of COVID-19 across the World**

# **COVID-19 WORLD MAP**

---

# WHAT IS THE AIM?

- To plot a **World map that shows the total number of confirmed cases of COVID-19 for each country.**
  - **The total number of confirmed cases can then be plotted with the respective dates of infection detection to create a visual timeline of the spread.**
  - **Ultimately, an animated map that changes colour with increasing days/date showing the spread of COVID-19 across the world.**
-

---

# WHAT DATA IS NEEDED?

- **Dataset should have:**
    1. **Country-wise world-wide distribution of total number of confirmed cases**
    2. **Corresponding dates for animating spread of infection**
    3. **Information regarding the latitudes/longitudes to plot the countries on a map**
-

---

# WHERE IS THE DATA COMING FROM?

- **Data Source: CoronaVirus API that uses data supplied by Johns Hopkins University**
  - **Johns Hopkins University's CoronaVirus Database - only trustworthy data source, other than WHO, that supplies World-wide infection data**
  - **Advantages: trustworthy data source; no limit to API request calls; absolutely free; open-source, i.e., trusted; updates daily; all data requirements satisfied;**
  - **Disadvantages: data is available only from Jan 22<sup>nd</sup> 2020 - mainly because Johns Hopkins University started data collection from Jan 22<sup>nd</sup>**
-

# WHAT DOES THE DATA LOOK LIKE?

	Country	CountryCode	Province	City	CityCode	Lat	Lon	Confirmed	Deaths	Recovered	Active	Date
0	Afghanistan	AF	Nan	Nan	Nan	33.94	67.71	0	0	0	0	2020-01-22 00:00:00+00:00
1	Afghanistan	AF	Nan	Nan	Nan	33.94	67.71	0	0	0	0	2020-01-23 00:00:00+00:00
2	Afghanistan	AF	Nan	Nan	Nan	33.94	67.71	0	0	0	0	2020-01-24 00:00:00+00:00
3	Afghanistan	AF	Nan	Nan	Nan	33.94	67.71	0	0	0	0	2020-01-25 00:00:00+00:00
4	Afghanistan	AF	Nan	Nan	Nan	33.94	67.71	0	0	0	0	2020-01-26 00:00:00+00:00

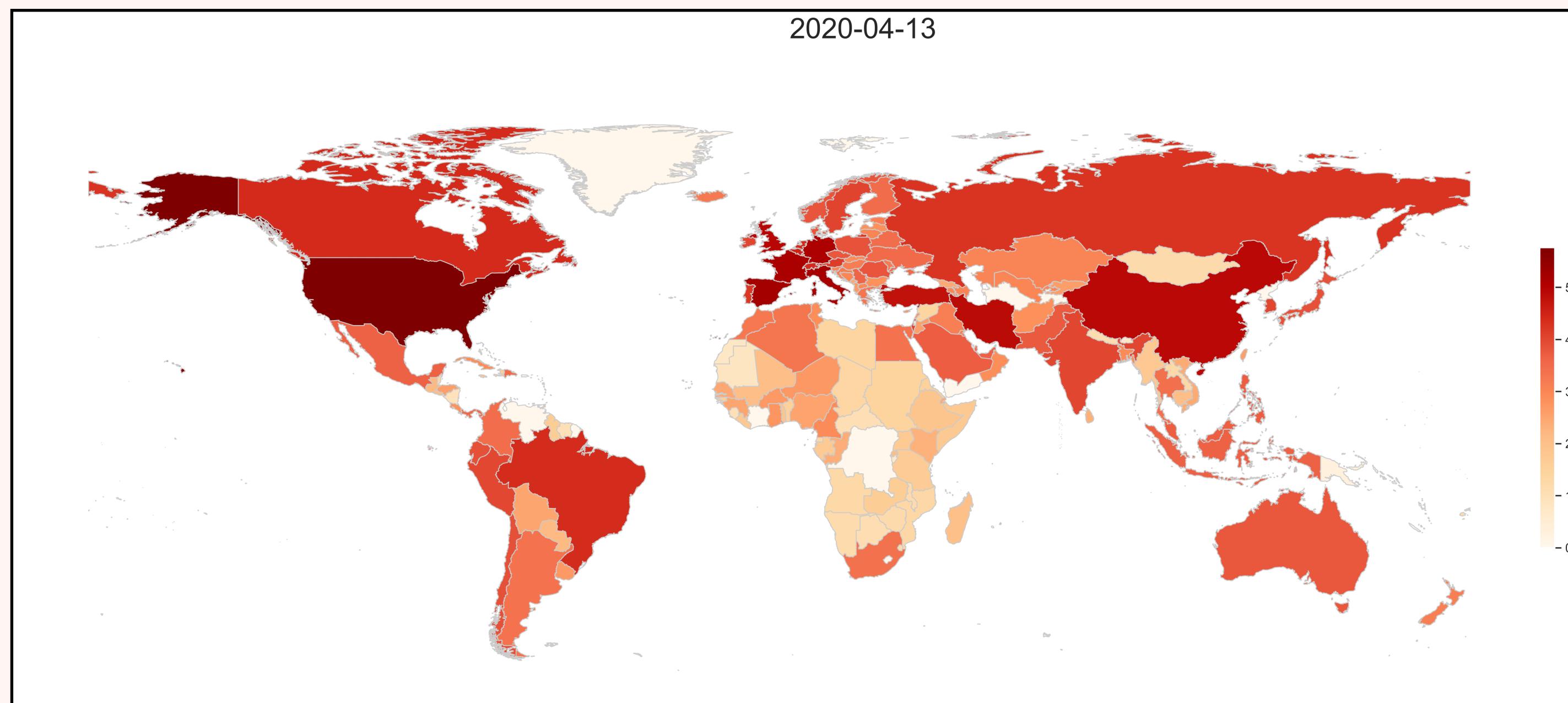
Raw data

	Country	Date	Confirmed	Deaths	Recovered	Active	geometry	log_Confirmed	log_Deaths
0	Aruba	2020-01-22 00:00:00	0.0	0.0	0.0	0.0	POLYGON ((-69.8822326660156 12.411099243165, ...	0.0	0.0
1	Aruba	2020-01-23 00:00:00	0.0	0.0	0.0	0.0	POLYGON ((-69.8822326660156 12.411099243165, ...	0.0	0.0
2	Aruba	2020-01-24 00:00:00	0.0	0.0	0.0	0.0	POLYGON ((-69.8822326660156 12.411099243165, ...	0.0	0.0
3	Aruba	2020-01-25 00:00:00	0.0	0.0	0.0	0.0	POLYGON ((-69.8822326660156 12.411099243165, ...	0.0	0.0
4	Aruba	2020-01-26 00:00:00	0.0	0.0	0.0	0.0	POLYGON ((-69.8822326660156 12.411099243165, ...	0.0	0.0

After cleaning and transforming

# HOW TO PLOT A WORLD MAP?

- Using libraries Geopandas + Seaborn
- Sample:



# HOW TO USE MAPS TO VISUALISE INFECTION GROWTH?

