**README file:**

**A Broad Analysis of Cook County During Corona Crisis**

**Usage**

**Script files**

* + Chicago\_suntimes\_webscraping\_and\_sentiment\_analysis.ipynb
  + Social\_distancing\_limit\_to\_cook.ipynb
  + Medical\_nursing\_survey\_data\_load.html
  + Medical\_archive\_data\_cleansing\_load.sql
  + Data\_normaliztion\_load\_to\_tables\_DDL\_DML.sql
  + Snowflake\_DM\_DDL\_Final.sql
  + Snowflake\_DM\_DML\_Final.sql
  + Tableaux\_Visualization\_queries.ipynb
  + Visualization\_social\_distancing\_data.twbx
  + Medical\_Examiner\_Covid.twbx
  + SurveyandNusringData.twbx

**Steps:**

1. **News Web Scraping:** Execute the Jupiter notebook called ***“Chicago\_suntimes\_webscraping\_and\_sentiment\_analysis.ipynb***” that collects all news articles from Chicago Suntimes published between Jan 1 and April 30. All these news articles are saved in a CSV file called “Chicago\_suntimes.csv” with the columns “Body”, “Date” and “Title”. The Jupiter notebook further calculates, for each date, the percentage of news that are related to Covid by identifying whether the news article (body + title), after making all words lowercase, contains any of the words 'covid', 'coronavirus', 'pandemic', 'virus', 'covid-19' or ‘covid19’. Finally, the Jupiter notebook assigns, for each day, an average polarity score between -1 and 1 to the news articles using the TextBlob natural language processing package (<https://textblob.readthedocs.io/en/dev/>). The results are stored in a CSV called “Chicago\_suntimes\_date\_perc\_score.csv” with the columns “Date”, “Percentage Covid” and “Polarity Score”. The results are also used to plot the figure called “plot\_perc\_score.png”, which is then annotated in Viewer on a MacBook to get the figure “plot\_perc\_score\_annotated.png”.
2. **Google Mobility Data:** The Global mobility data, which shows visits to various categories (Grocery\_pharmacy, Parks, Residential, Transit, Retail\_recreation, Workplace) around the world from Febr 15 until the present, was downloaded and opened in Vim editor in bash. All entries not for Cook county were deleted and labeled according to the categories were added. The resulting csv file was opened in Numbers (the equivalent of Excel on a Macbook) and a plot of the Mobility in the different categories from Febr 15 until April 11 was created.

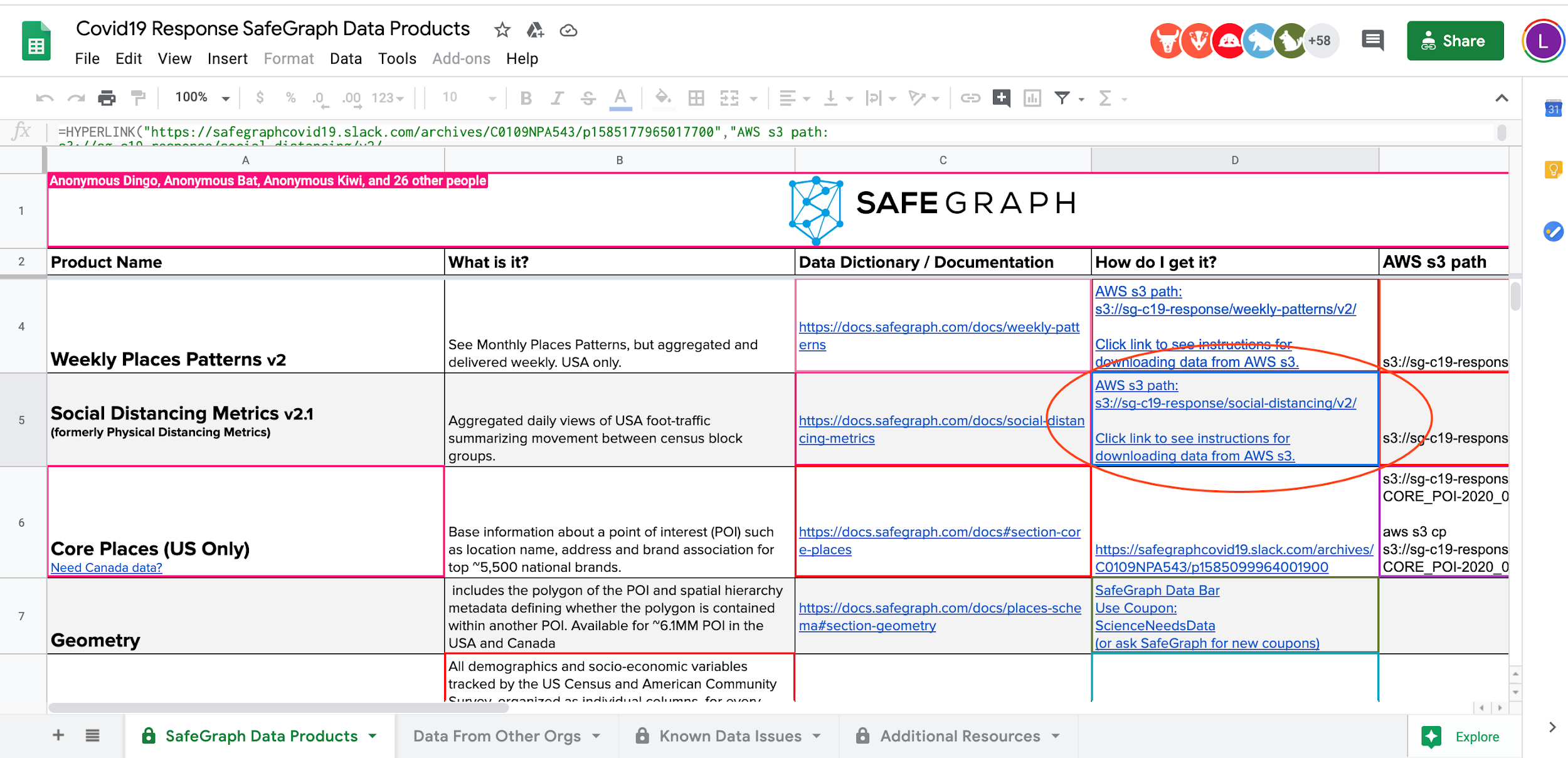
Reference : <https://www.gstatic.com/covid19/mobility/Global_Mobility_Report.csv?cachebust=c050b74b9ee831a7>

1. **Medical Examiner & Nursing Data:** Execute the python script “***Medical\_nursing\_survey\_data\_load.html”***which reads the medical examiner denormalized data, cook county nursing data, performs the data cleaning like handling special characters and load the data to mysql.
2. **Survey Data:** We prepared the survey for COVID 19 using google forms and circulated to our surroundings for response. After the data collection, we utilized Open Refine to clean the data especially standardizing Latinx ethnic group entries and correcting the user entries errors related to city,news sources and created a csv file. Used **‘Load Data Local Inpath’** Sql command to read the file and load the data to table directly. Here is the link of the survey

<https://docs.google.com/forms/d/e/1FAIpQLSfCCd99BhZ0Eg_x66dX4fLXZnGXXY7v657-fG0j6LP04rS9GA/viewform>

1. **SAFEGRAPH Social Distancing Data:** The Safegraph Database is a huge database that aggregates GPS pings from millions of phones across the US. It has, among others, datasets tracking foot traffic to thousands of points of interest (restaurants, cafes, malls etc) in the US, as well as a dataset tracking how long people stay at home, how far they travel etc. Due to time restrictions, only the latter data set was used.

In order to get access to the data, Laura Tociu joined the Slack of Safegraph and got access to a Google doc with all the links:



The data is organized in terms of Census Block Groups (CBG), which are small areas delimited by polygons, used by SafeGraph to perform their data collection. The Census Block dataset is available here:

<https://www.safegraph.com/open-census-data>

The crucial file from above has the name “ cbg\_geographic\_data.csv” and contains the latitude and longitude of the center of the polygon enclosing the CBG area.

Since our dataset is based on ZIP codes, it was necessary to map the CBG codes to ZIP codes. This was done using the latitude and longitudes of the CBG’s, and a dataset (<https://www.kaggle.com/joeleichter/us-zip-codes-with-lat-and-long>) downloaded from Kaggle that maps ZIP codes to a latitude / longitude pair.

Due to the size of the Social Distancing data (~ 60 GB), and the fact that we were only looking for Cook county ZIP Codes, the first step in data cleaning was to truncate the file “datasets\_5391\_8097\_zip\_lat\_long.csv” to only keep the rows corresponding to the Illinois ZIP codes, 60000 - 63000 (it was easy to do after ordering by ZIP code).

Then, to further minimize code execution time, we looked at the minimum and maximum latitudes in the truncated “datasets\_5391\_8097\_zip\_lat\_long.csv” file. We ordered and deleted all rows in the file “cbg\_geographic\_data.csv” that were far from those latitudes. Then we looked at minimum and maximum longitudes in the “datasets\_5391\_8097\_zip\_lat\_long.csv” file and further removed entries in “cbg\_geographic\_data.csv” that were far from those longitudes. The end result was a smaller file “datasets\_5391\_8097\_zip\_lat\_long.csv” that contains only Illinois ZIP codes, and a much smaller file “cbg\_geographic\_data.csv” that contains only the CBG codes with geographic location inside or very close to Illinois.

All the above was done manually in Vim. To even further reduce execution time, only the Cook county ZIP codes were extracted from the file “Location\_DataWorld.csv” and saved in a file “cook\_zipcodes.csv”, also manually

Finally, a Jupyter notebook, **“Social\_distancing\_limit\_to\_cook.ipynb”,** was run to match CBG codes to the ZIP codes in Cook county by calculating, for each CBG latitude and longitude, the closest ZIP code latitude and longitude. Subsequently, social distancing metrics we thought could be useful were extracted from the SAFEGRAPH database for each CBG code in Cook county, and the ZIP code was added as another column. Since there are many CBG codes per ZIP code, the metrics were aggregated per ZIP code. The latter two steps, even though they are included in the “**Social\_distancing\_limit\_to\_cook.ipynb**” notebook, were run on the Midway cluster as separate Python scripts and took > 24 hours to run.

The result of the social distancing data cleaning is a CSV file called Social\_distancing\_data\_Cook\_febr\_april.csv” that contains data for 170 ZIP codes in Cook county. Some data was missing or looked anomalous in the SAFEGRAPH files so the Python script was only successful in cleaning data for 170 out of about 220 ZIP codes in Cook county. The social distancing metrics in the file “Social\_distancing\_data\_Cook\_febr\_april.csv” are:

In order to visualize Social distancing data better in Tableau, a series of queries were performed in the Python notebook called “Tableaux\_visualization\_queries.ipynb”. The results of the queries, Average\_delivery\_behavior\_devices\_per\_10\_days.csv,Average\_full\_time\_work\_behavior\_devices\_per\_10\_days.csv” and “Average\_perc\_time\_home\_per\_10\_days.csv” were manually concatenated into the file “Social\_distancing\_per\_10\_days.csv”, which was used in Tableaux to visualize the data using a Metric filter.

**Data Transformation and Normalization:**  Execute sql file **“Data\_normaliztion\_load\_to\_tables\_DDL\_DML.sql”** to perform transformation, data cleanup of medical archive, nursing and survey data, normalization of all the data. Unique primary keys, handling of nullable columns and defaults were handled as part of this process. The data is normalized upto 3NF. Execute sql **“Medical\_archive\_data\_cleansing\_load.sql**” first to load the medical archive,nursing and survey data to a staging layer followed by above normalization sql file.

**OLAP Dimensional Data** - Execute the DDL “Snowflake\_DM\_DDL\_Final.sql” to create the Snowflake dimensional tabels and execute “Snowflake\_DM\_DML\_Final.sql” to load the data to dimensional tables. Dimensional tables use the data from the above normalization transactional database.

**BI Visualization**: Execute below tableau files to visualize patterns related to the Social distancing data, socio-economic and/or race correlation on COVID cases in Cook County based on zip codes, survey and nursing home observations

* + Visualization\_social\_distancing\_data.twbx
  + Medical\_Examiner\_Covid.twbx
  + SurveyandNusringData.twbx

### **Authors**

Cristal Garcia

Karlos Dodson

Laura Tociu

Surendiran Rangaraj