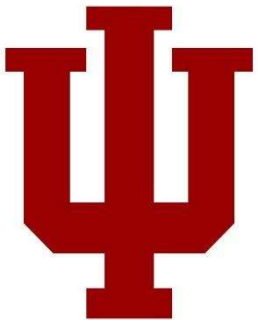


United States Post COVID-19 Pandemic Travel Recovery

Joel Klein

Luddy School of Engineering, Indiana University

INFO 535: Management, Access, & Use of Big Data

Dr. Inna Kouper

April 20, 2022

Introduction

On January 20, 2020, the first reported COVID-19 case reached the United States of America. As of April 11, 2022, nearly two and a half years later, 982 thousand deaths and 80.2 million confirmed cases summarize the ravaging pandemic's impact on American lives. With confirmed cases now plummeting to on average less than 29 thousand and deaths to a mere 500 daily, the United States seems to be recovering to "normalcy" with the workforce majority returning to in-person work and traveling steadily rising. There remain portions of the general population choosing to stay home despite low daily case and death counts and 66.3 percent of the eligible population fully vaccinated (receiving primary vaccine series).

Analyzing daily trip data since January 2019 published by the Bureau of Transportation illustrates a nationwide travel recovery since the beginning of the pandemic. Several big data technologies and capabilities such as APIs, virtual machines, cloud services, distributed storage, Python, SQL, and Tableau support developing a nimble and robust pipeline to extract, ingest, clean, preserve, and share publicly available data to gain deeper travel recovery insights. In addition to assessing recovery nationwide, this research implements a big data pipeline to uncover the relationships between vaccination rates, metropolitan class (urban/rural), political results, demographics, and weather patterns on trip recovery at a U.S. county level. The resulting pipeline implementation detects the direct relationship between monthly vaccination rates and travel throughout 2021. The primary geographic areas remaining under pre-pandemic travel levels with daily average travelers in February 2022 ranging between 95 to 99% of levels two years prior are associated with higher metropolitan, older, and democratic populations.

Background

The project objective is to build an analytics dashboard tracking travel levels before, during, and after the COVID-19 pandemic while gaining insight on the driving factors halting travel recovery in specific areas of the country.

Although the Bureau of Transportation reports travel statistics directly on their website, there is no readily available dashboard which compares current travel rates directly to pre-COVID rates. In addition, there is also no single source which highlights the relationship between metro status, political voting, demographics, and weather - all effects shown to be associated with higher vaccination rates, case counts, and general apprehension of COVID-19 - on travel recovery. The resulting dashboard demonstrates the recovery rates over time by vaccination rate, average age, and weather patterns, as well as compares travel recovery of metro counties vs. non-metro counties, and democratic vs. republican counties respectively.

This project involves an extensive data management plan to extract, transform, load, and integrate six publicly available data sources (table 1) together before feeding the dashboard. Documentation for each source is listed in the appendix.

Source	**Description**	**Frequency**	**Update cadence**
Bureau of Transportation (BTS)	U.S. Trips by Distance - County Level	Daily	Daily
Center for Disease Control (CDC)	U.S. Vaccination Counts & Rates - County Level	Daily	Daily
National Center for Health Statistics (NCHS)	Urban-Rural Classifications - County Level	N/A	Last Updated: 2013
U.S. Census Bureau	Age/Sex Population Breakdown - County Level	Yearly	Last Updated: 2020
Harvard Dataverse	Presidential Election Returns 2000-2020 - County Level	N/A	Last Updated: June 2021
National Centers for Environmental Information (NCEI)	Monthly Temperature/Precipitation - County Level	Monthly	Monthly

Table 1: Project Data Sources Summary

A suite of tools including Google Cloud Platform services like Cloud Functions for data extraction and ingestion, BigQuery for distributed data storage and modeling, and Cloud Scheduler for batch scheduling data updates, Python for data cleaning, SQL for data extraction, and Tableau Server for visualization and preservation execute the big data pipeline end-to-end. Experience using these technologies in conjunction with telling a data story are high value skills in the data science job market today. The methodology section details the application of these technologies further for producing the product.

Methodology

There are numerous steps within the data pipeline to extract, transform, store, model, analyze, preserve, and publish/share data. The main steps involve configuring a project within a cloud servicing platform (Google Cloud Platform), creating a database and storage model, extracting, transforming, and loading static public data, extracting, transforming, loading, and updating daily batch public data, modeling data, connecting data to Tableau, building a dashboard, and publishing the dashboard (figure 1).

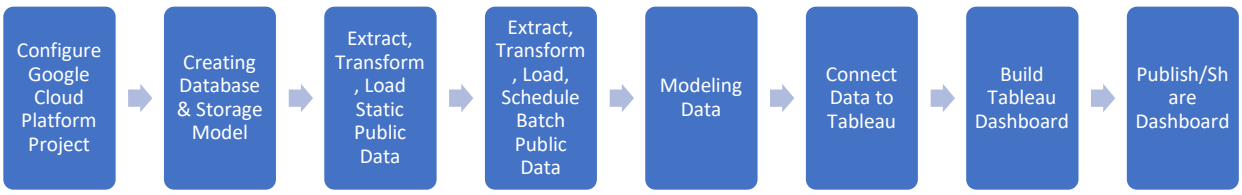


Figure 1: Project Pipeline Steps

Google Cloud Platform (GCP) is Google's cloud computing services collection running on Google hardware available for compute, storage, and application development. This project leverages a suite of GCP services including Google BigQuery - a distributed data warehouse storage solution, Google Cloud Functions - a serverless compute service for running functions in a fully managed environment requiring little virtual machine configuration, and Google Cloud Scheduler - a fully managed CRON job scheduling service. To start, the user must start and

provision a project within Google Cloud. The step-by-step instructions for building a GCP project are in appendix part 2.

Once the project is set up, the next step is to extract static (do not update) public data sources and load them into Google BigQuery. Google BigQuery is a serverless, highly scalable, and cost-effective cloud data warehouse intended for business intelligence and light data analysis. Google BigQuery is well suited for big data projects as it is built on Google's distributed file system called Colossus. BigQuery benefits from Colossus' sophisticated, highly parallelized, and scalable infrastructure. Data ingested into BigQuery is stored across numerous Colossus clusters with significant disks enabling data replication and recovery, while reducing risk for single failure points when queries executed on the data. A data warehouse storage model is suitable since each public data source is relational and stored with a common county identifier (FIPS) for integration. Each static data source (*Urban-Rural Classifications - County Level* [2]*, Age/Sex Population Breakdown - County Level* [3]*, Presidential Election Returns 2000-2020 - County Level* [4]) can be downloaded directly from the respective government website specified in appendix part 3 or extracted from this projects GitHub repository (appendix part 1) in the data folder. These data sources are manually ingested as they will not update in the future and the steps for ingesting each static data (appendix part 4) are only needed once.

The remaining three data sources are updated daily (*U.S. Trips by Distance - County Level* [5]*, U.S. Vaccination Counts & Rates - County Level* [6]) or monthly (*Monthly Temperature/Precipitation - County Level* [7]). A key business intelligence pillar is providing the most updated data to the end user with low latency. Rather than manually extracting and ingesting these data sets into BigQuery, this project uses Cloud Functions to automate the data extraction and ingestion process. A Cloud Function can be provisioned to execute several general-purpose programming languages over a specific memory and instance allocation. There is no need for managing and provisioning servers or infrastructure and the cloud functions are run in a virtual environment with a few simple steps (appendix 5). Since each data source is available via

application programming interface (API), a user can write a script on a python runtime to extract each source from the API into a specified environment, transform the data, and write the output into BigQuery. The complete steps are outlined in appendix part 5 along with the python script and environment requirements for the Cloud Function. Each Cloud Function can then be scheduled either daily or monthly via a cron job using Google's Cloud Scheduler (steps in appendix 5).

After each raw data source is ingested into BigQuery, the last step in preparing the data for use in Tableau is to clean and integrate each source together into a single modeled table. Each source requires a series of transformations. The demographic data is limited to the most recent year; the election results are aggregated to identify the 2020 election top party and percentage of votes for the Democratic party; the trip data adds features for calculating post vs. pre covid travel statistics; the rural urban status codes are mapped to actual descriptions. After the transformations are applied and the final features are selected for each source, the data is integrated together by the common FIPS key. This script is then saved and scheduled within BigQuery to update daily. The full scheduled SQL script is available in the GitHub repository in the code folder. The steps to schedule the script are outlined in appendix 6.

The final step in the pipeline is the Tableau dashboard build and deploy. Tableau's integration to connect with several data source types including BigQuery is a significant competitive advantage over other business intelligence and visualization software. By creating a new data source in Tableau, authenticating, and connecting with the BigQuery data warehouse within the GCP project the data is available within Tableau for the dashboard build (steps highlighted in appendix 8). The final tableau workbook is available within the dashboards folder in the project GitHub repository. In practice, dashboards are published on an organization's Tableau Server along with their data extracts (on a schedule). However, Tableau Server is a paid subscription-based service unavailable to Indiana University students. Instead, the dashboard is published and hosted on Tableau Public, a free public platform for community members to publish dashboards using public data. The link to the dashboard is available in appendix part 1.

Results

In December 2019, two months prior to the COVID-19 onset, 20.25% of Americans on average stayed at home daily. During the pandemic height just one year later, that percentage jumped as high as 29%. The almost nine percentage point increase equates to an additional 28 million Americans choosing to stay home during the pandemic. The only major region still hovering slightly below normal levels was the southeastern U.S. except for Florida (figure 3). The states most heavily impacted during this time include New York, California, Oregon, and Washington likely due to stricter stay-at-home mandates.

As of the most recent month of full available data, February 2022, the percentage of Americans staying home has rebounded to 20.54% just slightly higher than pre-pandemic levels.

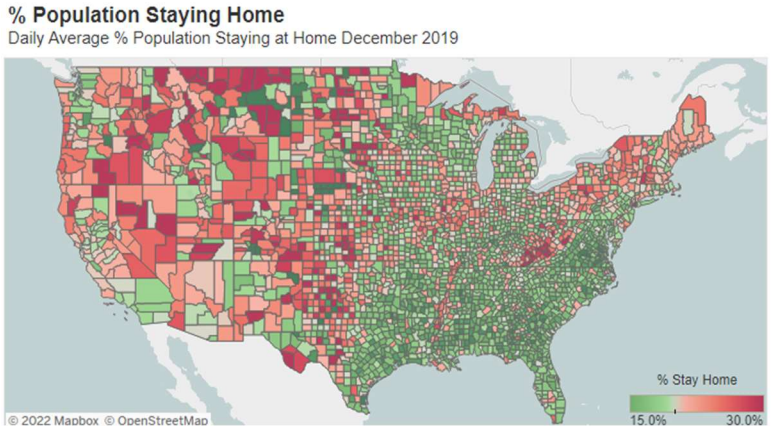


Figure 2: % U.S. Population Staying Home (December 2019)

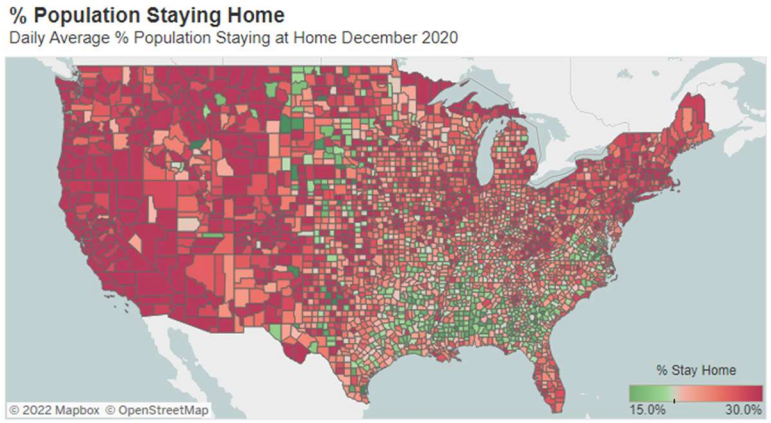


Figure 3: % U.S. Population Staying Home (December 2020)

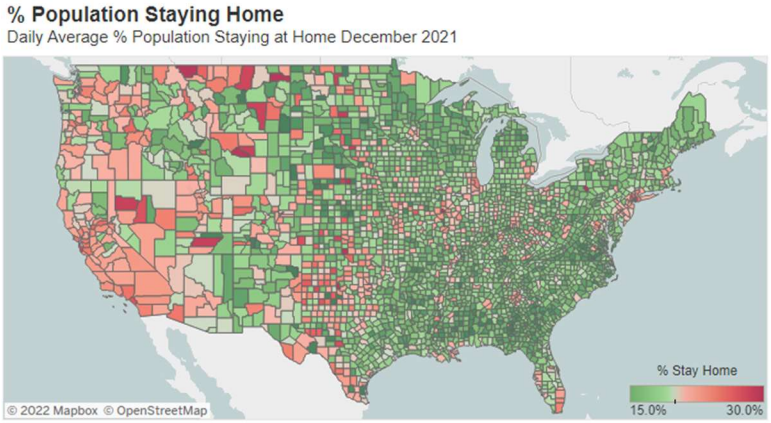


Figure 4: % U.S. Population Staying Home (December 2021)

A major factor influencing the travel recovery included the introduction and distribution of vaccinations across the United States. There is a direct relationship over time between the percentage of Americans fully vaccinated and the percentage of Americans staying home. This relationship is likely also heavily influenced by the time of year and weather patterns (the vaccinations were fully rolled out in December 2020). From January to April 2021 as vaccination rates increased to about 20% (and monthly COVID cases decreased drastically), traveling rapidly rebounded near normal levels (figure 5). There were slight travel decreases again in subsequent summer months as COVID cases began to climb in the U.S. which have since recovered.

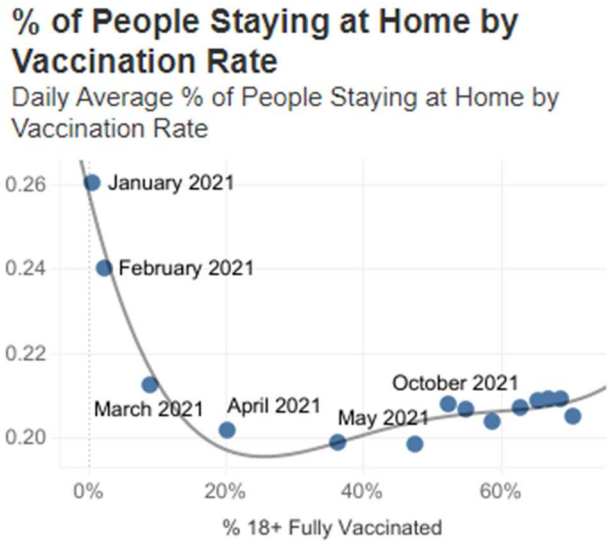


Figure 5: % U.S. Population Staying Home by Vaccination Rate

The main benefit from the custom data pipeline implementation is the ability to assess at a more granular level, which types of counties have yet to recover in daily travel. A major indicator of travel recovery is the metropolitan class (figure 6). Metro counties are at a 98% daily average travel recovery in February 2022 compared to two years prior while non-metro counties surpassed pre-pandemic rates at 102%. This may potentially be

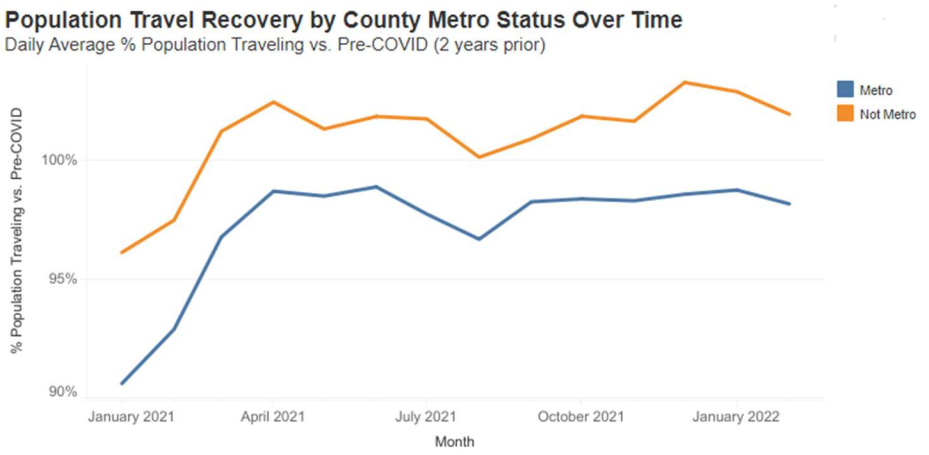


Figure 6: Population Travel Recovery by County Metro Status Over Time

influenced by the population migration out of metropolitan areas to non-metro areas throughout the pandemic explained by W. Frey at *Brookings Institution* [1].

Another known relationship is the difference in vaccination rates by political party. Counties more democratic (as determined by 2020 Presidential election results) statistically have higher vaccination rates (figure 7). The travel recovery for democratic counties is also lower than republican counties with February 2022 daily population traveling rates at 98% and 100% respectively. Overall, the flexible and robust dashboard tool highlights the associations between vaccination rates, weather, metro status, demographics, and political party on travel recovery.

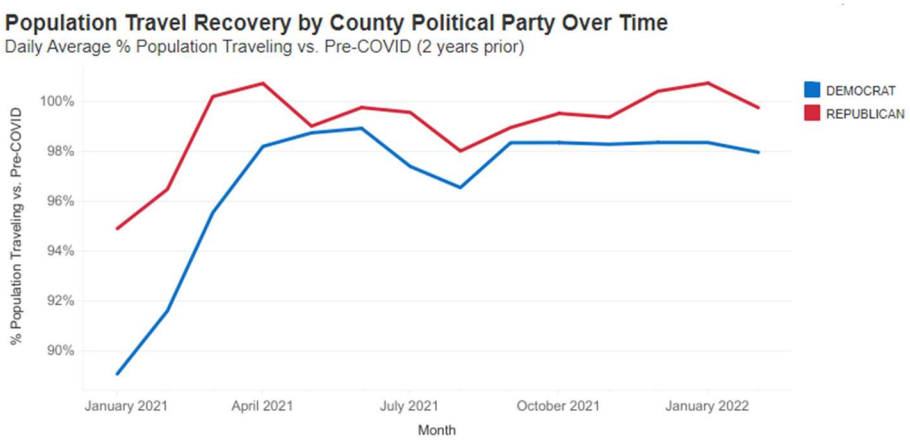


Figure 7: Population Travel Recovery by County Political Party Over Time

Discussion

There are clear signs indicating changes in travel recovery by the features analyzed. It is surprising to see such a large gap in travel recovery between metro and non-metro areas. It is quite difficult from this simple analysis, however, to identify which features are contributing the heaviest to the travel recovery rates. Most of the predictive features are associated with each other which limits a univariate analysis ability to identify the true effect of vaccination rate, weather, metro status, demographics, and political party on travel recovery alone. Fitting a non-linear machine learning algorithm such as XGBoost (which handles predictor interaction) plus a Shapley additive explanations (SHAP) analysis, a state-of-the-art machine learning explainability process based on game theory, would be able to isolate each predictor effect while accounting for the interaction.

There are other potential data sources which may add incremental value if included in the scope of this research such as gas prices, cases and deaths, mandates, and population changes. If this project were a machine learning model, the data scientist would likely consider modeling the travel patterns at a more granular level such as daily to include the full effect of more granular weather data like temperature and precipitation.

Creating the end dashboard employed numerous big data skills learned throughout the course to implement the fully reproducible end-to-end analysis pipeline. Several data sources with differing types and update cadences were ingested into a data warehouse sitting on top of a distributed file system using cloud computing services. Several modules and labs practicing the using Google Cloud Platform and its services like BigQuery increased confidence planning the data lifecycle and executing the pipeline. The pipeline also integrated more commonly used data science tools like Python, SQL, and Tableau for data processing and analysis and GitHub for version control.

The most challenging roadblocks encountered throughout the project implementation were more administrative tasks. The first hurdle occurred when provisioning the proper environment to enable running the Cloud Function endpoint for ingesting the data. The appropriate package versions required adding to the requirements file to set up the python runtime environment to execute the data extraction and ingestion. The other administrative hurdle involved setting up the proper security and IAM roles within Google Cloud Scheduler to invoke the Cloud Function using an HTTP event trigger. After additional documentation research, the proper IAM role setup enabled the CRON job to invoke the function and update the data on the defined cadence.

Conclusion

Cloud computing services enable end-to-end data pipelines to extract, transform, store, model, analyze, preserve, and publish/share data from several sources. Integrating six publicly available data sources from the Bureau of Transportation (BTS), Center for Disease Control (CDC), National Center for Health Statistics (NCHS), U.S. Census Bureau, Harvard Dataverse, and National Centers for Environmental Information (NCEI) identify the primary geographic areas in the United States remaining under pre-pandemic travel levels. Higher fully vaccinated, metropolitan, older, and democratic counties are associated with higher travel recovery levels. The end product (scheduled data pipeline and dashboard) enables continuous tracking for these travel measures. Over the next few months heading into the summer as travel recovery continues to rise throughout the U.S., it will be informative to assess if the patterns and relationships will remain.

References

[1] W. Frey. (April 2022). New census data shows a huge spike in movement out of big metro areas during pandemic, *Brookings Institution.*

[2] National Center for Health Statistics. (June 2017). NCHS Urban-Rural Classification Scheme for Counties, *CDC*.

[3] U.S. Department of Commerce. (February 2020). Review Guide for the Population Estimates Challenge Program, *United States Census Bureau Economics and Statistics Administration.*

[4] MIT Election Data and Science Lab. (2018). County Presidential Election Returns 2000-2020, *Harvard Dataverse*, V10.

[5] Bureau of Transportation Statistics. (February 2022). Trips by Distance. *U.S. Department of Transportation*.

[6] *CDC*. (April 2022). COVID-19 Vaccinations in the United States, County.

[7] National Oceanic and Atmospheric Administration. (October 2018). nClimDiv County Temperature-Precipitation, *National Centers for Environmental Information*.

[8] B. Chamblee. (November 2021). How to Upload Data to Google BigQuery Using Python: In 3 Steps, *Towards Data Science.*

[9] Google. (March 2022). Ingesting New Datasets into BigQuery.

Appendix 1: Project links

Project Github Repository:

https://github.iu.edu/joeklein/final-project-i535

Google Cloud Platform Project:

https://console.cloud.google.com/home/dashboard?project=upbeat-stratum-347321

Tableau Dashboards on Tableau Public:

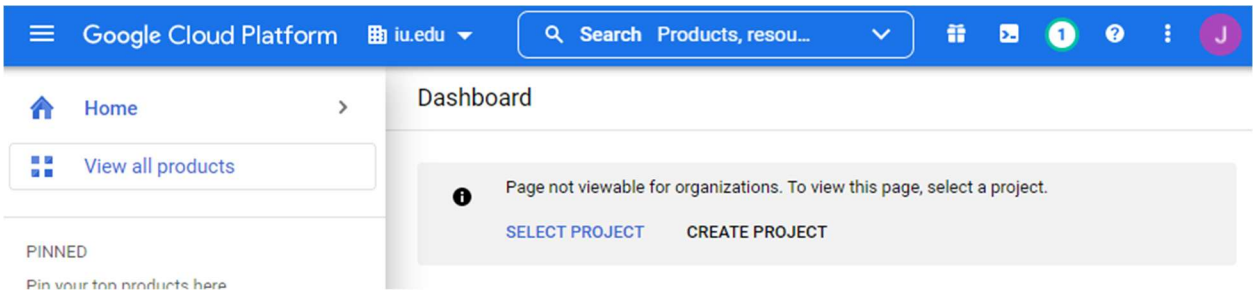
https://public.tableau.com/app/profile/joel.klein

Appendix 2: Data Sources

Source	Description	URL	Documentation URL	Frequency	Update cadence	BigQuery Dataset ID
Bureau of Transportation (BTS)	U.S. Trips by Distance - County Level	https://data.bts.gov/Research-and-Statistics/Trips-by-Distance/w96p-f2qv	https://data.bts.gov/Research-and-Statistics/Trips-by-Distance/w96p-f2qv	Daily	Daily	bts_trips
Center for Disease Control (CDC)	U.S. Vaccination Counts & Rates - County Level	https://data.cdc.gov/Vaccinations/COVID-19-Vaccinations-in-the-United-States-County/8xkx-amqh	https://data.cdc.gov/Vaccinations/COVID-19-Vaccinations-in-the-United-States-County/8xkx-amqh	Daily	Daily	cdc_vax_rates
National Center for Health Statistics (NCHS)	Urban-Rural Classifications - County Level	https://www.cdc.gov/nchs/data_access/urban_rural.htm#Data_Files_and_Documentation	https://www.cdc.gov/nchs/data_access/urban_rural.htm	N/A	Last Updated: 2013	county_urban_rural_codes_2013
U.S. Census Bureau	Age/Sex Population Breakdown - County Level	https://www2.census.gov/programs-surveys/popest/datasets/2010-2020/counties/asrh/	https://www2.census.gov/programs-surveys/popest/FTP2_Key.xlsx	Yearly	Last Updated: 2020	census_demographics_est_2020
Harvard Dataverse	Presidential Election Returns 2000-2020 - County Level	https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/VOQCHQ	https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/VOQCHQ	N/A	Last Updated: June 2021	county_pres_election_results_2000_2020
National Centers for Environmental Information (NCEI)	Monthly Temperature/Precipitation - County Level	https://www.ncei.noaa.gov/pub/data/cirs/climdiv/	https://www.ncei.noaa.gov/pub/data/cirs/climdiv/county-readme.txt	Monthly	Monthly	ncei_weather_data

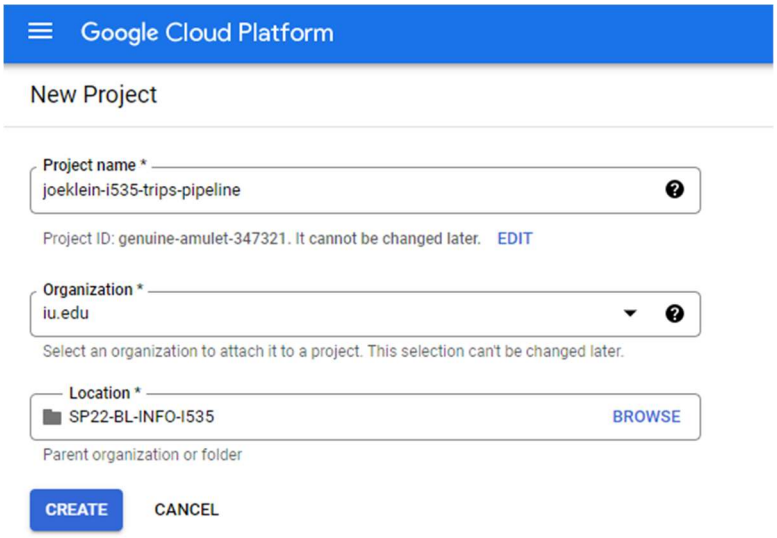
Appendix 3: Google Cloud Platform Project Setup Steps

1. Login to https://console.cloud.google.com/ using IU credentials
2. Select CREATE PROJECT:



3. Set up the project with:

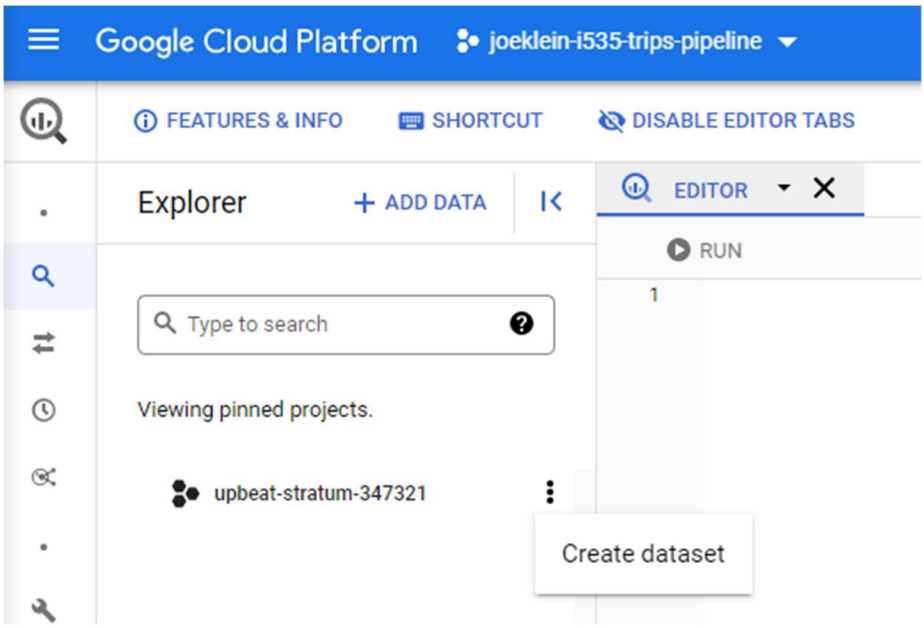
   Project name: joeklein-i535-trips-pipeline
   Project ID: upbeat-stratum-347321
   Organization: iu.edu
   Location: SP22-BL-INFO-I535



4. Click CREATE to create the project

Appendix 4: Uploading Static Data Sources to Google BigQuery Steps

1. Navigate to BigQuery within GCP Project
2. Under the project id click 3 dots and Create dataset



3. Input the following:

   Dataset ID: census_demographics_est_2020

   Data location: us (multiple regions in the United States)

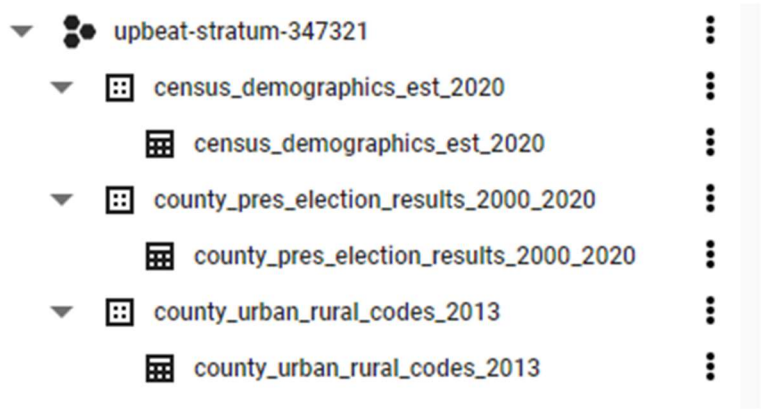
4. Click CREATE DATASET
5. Repeat steps 3 and 4 for the following Dataset IDs:

   county_pres_election_results_2000_2020

   county_urban_rural_codes_2013

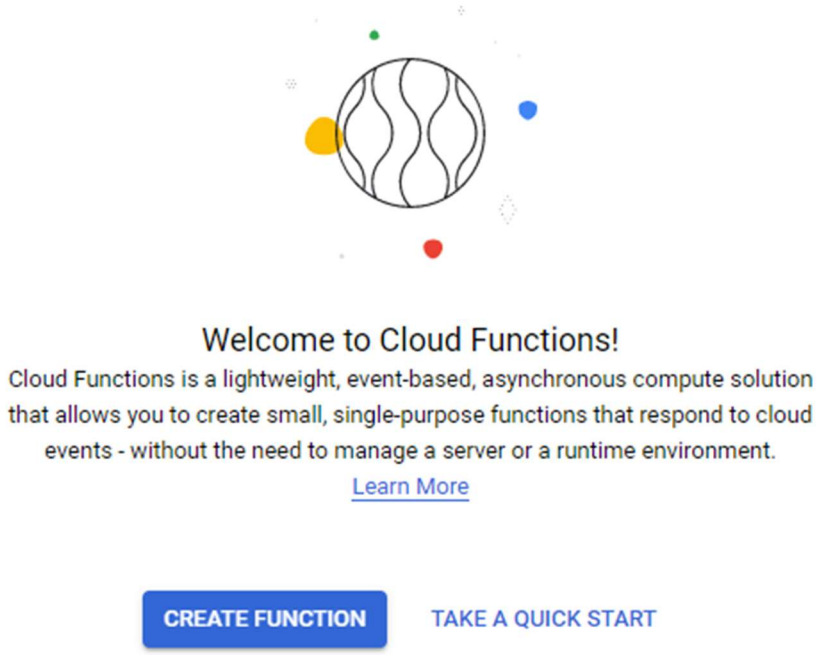
6. For each dataset, hit 3 dots and select Create table

7. Create table from Upload and browse to select the appropriate file from the Github repo: https://github.iu.edu/joeklein/final-project-i535/tree/master/static_data
8. Enter the Dataset ID as the Table name
9. Select Auto detect for Schema
10. Keep all other values and click CREATE TABLE
11. After repeated for each of the 3 sources, BigQuery Explorer should resemble:



Appendix 5: Uploading Daily/Monthly Data Sources to Google BigQuery Using Google Cloud Function & Scheduling with Google Cloud Scheduler

1. Navigate to Google Cloud Functions
2. Select CREATE FUNCTION

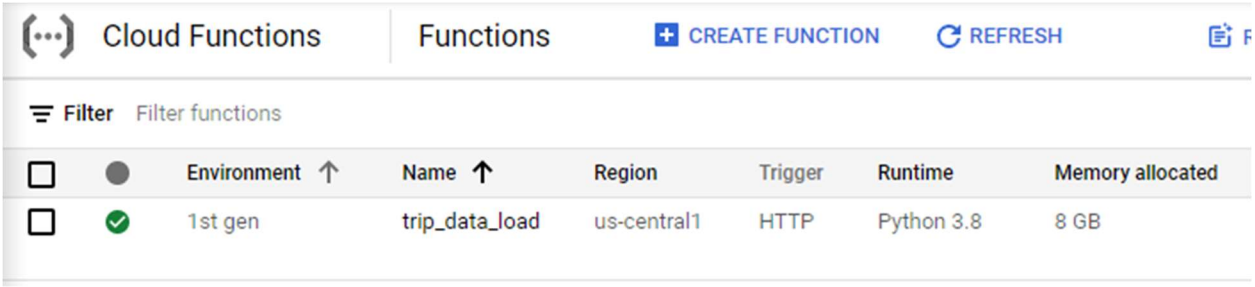


3. Enable the required APIs by selecting ENABLE
4. Insert the following:
    a. Basics:
        i. Function name: trip_data_load
    b. Trigger:
        i. Trigger type: HTTP
        ii. HTTP Authentication: Allow unauthenticated invocations
    c. Runtime, build, connections and security settings
        i. Memory allocated: 8GB

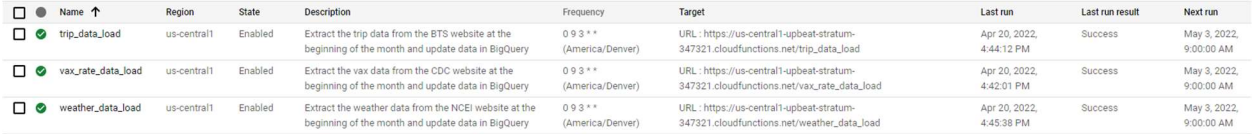
ii. Timeout: 540 seconds

iii. Maximum number of instances: 5000

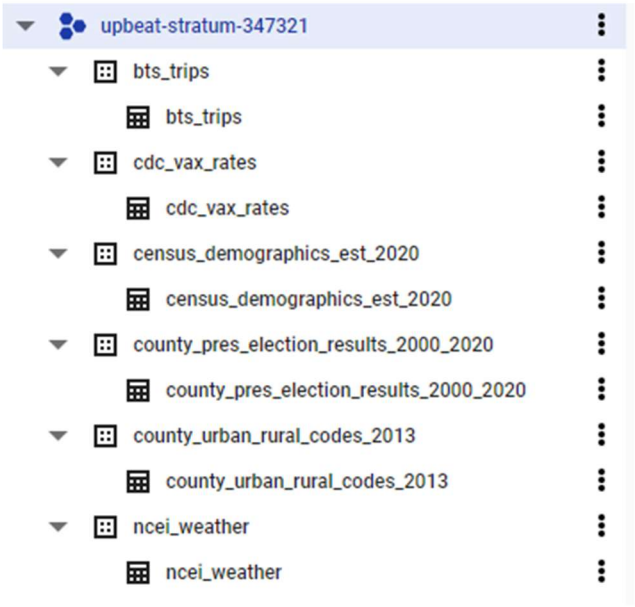
5. Select NEXT
6. Set the following:
    a. Runtime: Python 3.8
    b. Entry point: http_validation
7. Copy the requirement.txt file from https://github.iu.edu/joeklein/final-project-i535/tree/master/code/pt01_data_load/requirements.txt into requirement.txt
8. Copy the requirement.txt file from https://github.iu.edu/joeklein/final-project-i535/tree/master/code/pt01_data_load/trip_data_load.py into main.py
9. Click DEPLOY
10. Once Active the Cloud Functions should have a green check circle



11. To run the function, under Actions, click 3 dots, and select Test function
12. On the next page hit the blue TEST THE FUNCTION button
13. This function should take about a minute to run
14. Expect Output message to read: Data pulled and loaded
15. Repeat this process to create two more functions: vax_rate_data_load and weather_data_load
16. After testing each function the explorer in BigQuery looks like the following:



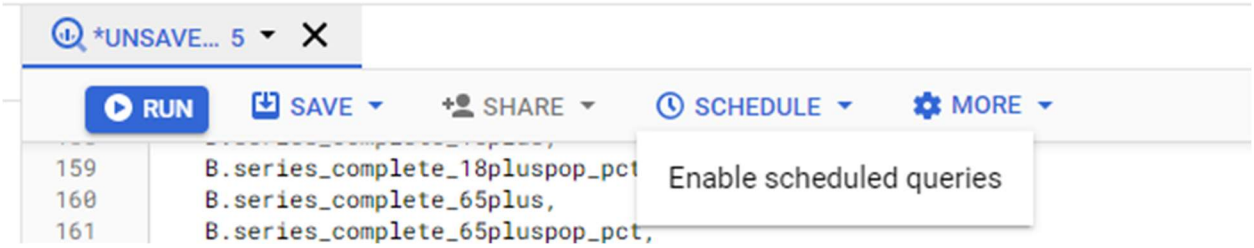
Name	Region	State	Description	Frequency	Target	Last run	Last run result	Next run
trip_data_load	us-central1	Enabled	Extract the trip data from the BTS website at the beginning of the month and update data in BigQuery	0 9 3 * * (America/Denver)	URL : https://us-central1-upbeat-stratum-347321.cloudfunctions.net/trip_data_load	Apr 20, 2022, 4:44:12 PM	Success	May 3, 2022, 9:00:00 AM
vax_rate_data_load	us-central1	Enabled	Extract the vax data from the CDC website at the beginning of the month and update data in BigQuery	0 9 3 * * (America/Denver)	URL : https://us-central1-upbeat-stratum-347321.cloudfunctions.net/vax_rate_data_load	Apr 20, 2022, 4:42:01 PM	Success	May 3, 2022, 9:00:00 AM
weather_data_load	us-central1	Enabled	Extract the weather data from the NCEI website at the beginning of the month and update data in BigQuery	0 9 3 * * (America/Denver)	URL : https://us-central1-upbeat-stratum-347321.cloudfunctions.net/weather_data_load	Apr 20, 2022, 4:45:38 PM	Success	May 3, 2022, 9:00:00 AM



17. Navigate to Google Cloud Scheduler
18. Click CREATE JOB
19. Enter the following (for scheduling BTS Trips ETL cloud function):
    a. Name: trip_data_load
    b. Region: us-central1
    c. Frequency: 0 9 3 * *
    d. Timezone: Mountain Daylight Time (MDT)
    e. Target type: HTTP
    f. URL: https://us-central1-upbeat-stratum-347321.cloudfunctions.net/trip_data_load
    g. HTTP method: POST
    h. HTTP headers: no headers
20. Click CREATE
21. To test the function select it and click RUN NOW
22. After completing this process for each cloud function and testing each job, the cloud scheduler pane should resemble:

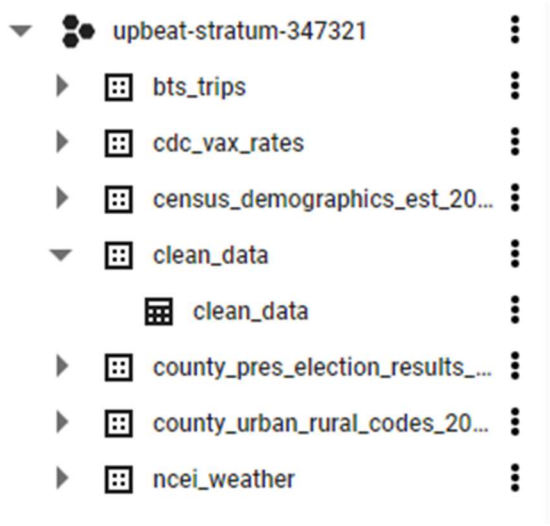
Appendix 6: Modeling Data in Google BigQuery Steps

1. Navigate to Google BigQuery
2. Copy the SQL code from https://github.iu.edu/joeklein/final-project-i535/tree/master/code/pt02_data_prep/clean_data.sql into the editor
3. Click the SCHEDULE button and Enable schedule queries. Enable the API.



4. After 2-3 minutes click the SCHEDULE button and Create new scheduled query
5. Enter the following information:
    a. Name: Clean Data ETL
    b. Repeats: Monthly On the 7
    c. Schedule end time: 5/8/22, 12:00 AM
    d. Check Set a destination table for query results
    e. Dataset: clean_data (create a new data set)
    f. Table Id: clean_data
    g. Destination table write preference: Overwrite table
    h. Data location: US
    i. Check Send email notifications
6. Click SAVE
7. This will initiate the data transfer. Note: you may need to hit RETRY if the API is not enabled yet.
8. After the transfer is complete you should see the following:

9. The new modeled table should appear within explorer in BigQuery:

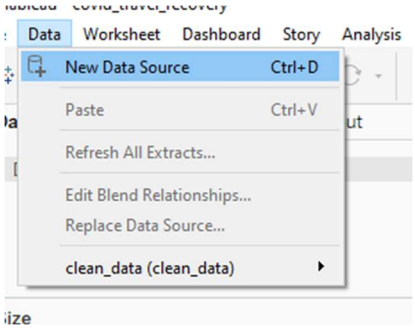


Appendix 7: Tableau Desktop Install

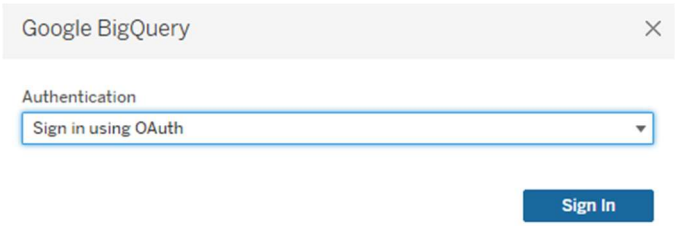
1. Navigate to https://www.tableau.com/academic/students
2. Click the FREE STUDENT LICENSE
3. Fill in the form
4. Follow the prompts to complete the download
5. When downloaded open Tableau Desktop

Appendix 8: Connecting Tableau to Google BigQuery Steps

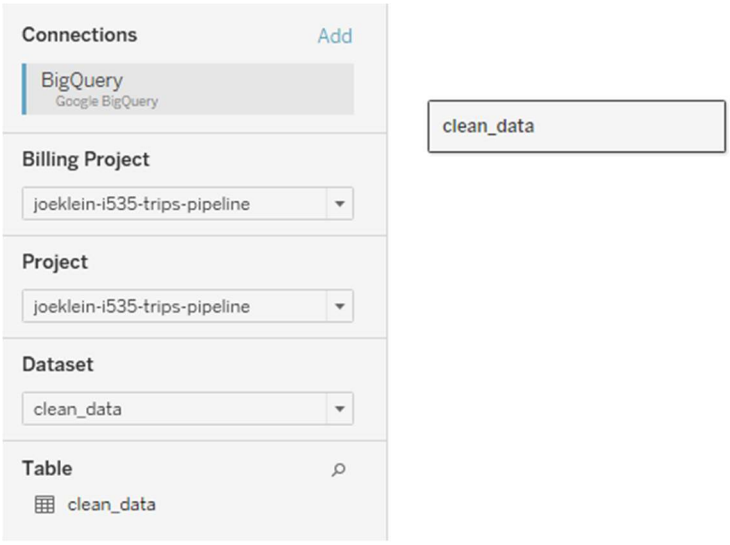
1. Open Tableau Desktop
2. Navigate to the Data tab and select New Data Source



3. Under To a Server, select Google BigQuery
4. Authenticate via OAuth and follow the prompts



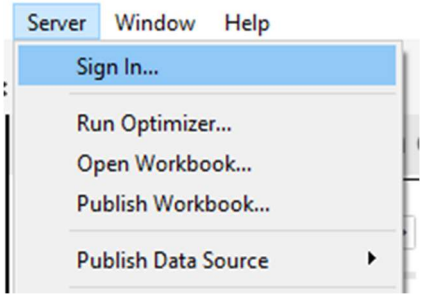
5. Select the inputs for Billing Project, Project, and Dataset as shown in the screenshot below:



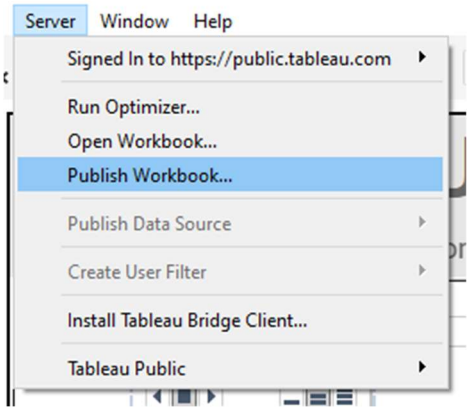
6. Change the Connection type to Extract and store the tableau .hyper file in a specific location
7. Proceed to build the dashboard

Appendix 9: Publishing Tableau Dashboards to Tableau Public

1. Open the Tableau workbook from the project Github repository: https://github.iu.edu/joeklein/final-project-i535/tree/master/dashboard/covid_travel_recovery.twb
2. Sign up for an account on Tableau Public: https://public.tableau.com/en-us/s/
3. Sign into Tableau Public by clicking Server in Tableau Desktop:



4. Click Connect
5. Sign into Tableau Public using the account you just set up
6. Publish Workbook by clicking Server in Tableau Desktop:



7. Enter the workbook name.
8. Repeat this process for each dashboard in the Tableau workbook