Covid-19 Vaccination Across the World

TEAM – 7

Kavya Nadella
Liping Shao
Dinesh Rane
Swetha Sethumadhavan

Introduction:

Vaccination is a critical tool in preventing the spread of COVID-19, reducing the severity of illness, and preventing hospitalizations and deaths. **70.6%** of the world population has received at least one dose of a COVID-19 vaccine. **13.53 billion** doses have been administered globally, and **20,499** are now administered each day. **32.8%** of people in low-income countries have received at least one dose.

Some countries however may have achieved higher vaccination rates, while others may face challenges, including vaccine supply issues and vaccination willingness. Despite significant achievement in vaccination development and distribution, there are critical gaps in vaccination coverage, and efficiency across different regions and countries.

This project aims to perform analysis on the coverage and effectiveness of COVID-19 vaccination across different regions and countries globally. This can play a crucial role in optimizing vaccination efforts, enhancing public health outcomes, and contributing valuable insights for current and future pandemic responses.

This project is worked on to achieve following goals:

- Vaccination Coverage Analysis
  - Which countries have highest vaccination rate, and which have the lowest?
  - Which countries had the most recent covid cases?
- Vaccination brands Analysis
  - Which vaccination brands are used most across the countries?

System implementation:

1. Input Dataset:

For this project, the dataset is obtained from this GitHub URL:
https://github.com/owid/covid-19-data/tree/master/public/data/vaccinations

The dataset has various information related to covid vaccination across different countries, but the scope of this project includes the usage of following csv files:

- locations.csv
- vaccinations.csv
- vaccinations-by-manufacturer.csv

locations.csv:

This csv file has following information:

- location: name of the country (or region within a country).
- iso_code: ISO 3166-1 alpha-3 – three-letter country codes.
- vaccines: list of vaccines administered in the country up to the current date.
- last_observation_date: date of the last observation in our data.
- source_name: name of our source for data collection.
- source_website: web location of our source. It can be a standard URL if numbers are consistently reported on a given page; otherwise it will be the source for the last data point.

location	iso_code	vaccines	last_observation_date	source_name	source_website
Afghanistan	AFG	CanSino, Covaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnil	2022-06-01	World Health Organization	https://covid19.who.int/
Albania	ALB	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V	2022-05-22	World Health Organization	https://covid19.who.int/
Algeria	DZA	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V	2022-05-01	World Health Organization	https://covid19.who.int/
Andorra	AND	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	2022-05-22	World Health Organization	https://covid19.who.int/
Angola	AGO	Oxford/AstraZeneca	2022-05-22	World Health Organization	https://covid19.who.int/
Anguilla	AIA	Oxford/AstraZeneca, Pfizer/BioNTech	2022-05-27	World Health Organization	https://ais.paho.org/imm/IM_DosisAdmin-
Antigua and Barbuda	ATG	Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V	2022-05-30	Ministry of Health	https://covid19.gov.ag
Argentina	ARG	CanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V	2022-06-06	Ministry of Health	https://covidstats.com.ar/
Armenia	ARM	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, Sputnik Light, Sputnik V	2022-05-15	World Health Organization	https://covid19.who.int/
Aruba	ABW	Pfizer/BioNTech	2022-06-07	Government of Aruba	https://www.government.aw
Australia	AUS	Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech	2022-06-07	Government of Australia via CovidBaseAU	https://covidbaseau.com/
Austria	AUT	Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech	2022-06-03	European CDC	https://www.ecdc.europa.eu/en/publicatic
Azerbaijan	AZE	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V	2022-06-02	Government of Azerbaijan	https://koronavirusinfo.az

vaccinations.csv:

This csv file has following information:

- location: name of the country (or region within a country).
- iso_code: ISO 3166-1 alpha-3 – three-letter country codes.
- date: date of the observation.
- total_vaccinations: total number of doses administered. For vaccines that require multiple doses, each individual dose is counted. If a person receives one dose of the vaccine, this metric goes up by 1. If they receive a second dose, it goes up by 1 again. If they receive a third/booster dose, it goes up by 1 again.
- total_vaccinations_per_hundred: total_vaccinations per 100 people in the total population of the country.
- daily_vaccinations_raw: daily change in the total number of doses administered. It is only calculated for consecutive days. This is a raw measure provided for data checks and transparency, but we strongly recommend that any analysis on daily vaccination rates be conducted using daily_vaccinations instead.

- daily_vaccinations: new doses administered per day (7-day smoothed). For countries that don't report data on a daily basis, we assume that doses changed equally on a daily basis over any periods in which no data was reported. This produces a complete series of daily figures, which is then averaged over a rolling 7-day window. An example of how we perform this calculation can be found here.
- daily_vaccinations_per_million: daily_vaccinations per 1,000,000 people in the total population of the country.
- people_vaccinated: total number of people who received at least one vaccine dose. If a person receives the first dose of a 2-dose vaccine, this metric goes up by 1. If they receive the second dose, the metric stays the same.
- people_vaccinated_per_hundred: people_vaccinated per 100 people in the total population of the country.
- people_fully_vaccinated: total number of people who received all doses prescribed by the initial vaccination protocol. If a person receives the first dose of a 2-dose vaccine, this metric stays the same. If they receive the second dose, the metric goes up by 1.
- people_fully_vaccinated_per_hundred: people_fully_vaccinated per 100 people in the total population of the country.
- total_boosters: total number of COVID-19 vaccination booster doses administered (doses administered beyond the number prescribed by the initial vaccination protocol)
- total_boosters_per_hundred: total_boosters per 100 people in the total population of the country.
- daily_people_vaccinated: daily number of people receiving a first COVID-19 vaccine dose (7-day smoothed).
- daily_people_vaccinated_per_hundred: daily_people_vaccinated per 100 people in the total population of the country.

location	iso_code	date	total_vacci	people_vaccina	people_fully_	total_boos	daily_vaccinations_	daily_vaccina	total_vaccinations	people_vaccinated_pe	people_fully_vaccina	total_boosters	daily_vaccinat	daily_people_vac	daily_people_vaccinated_
Afghanistan	AFG	2021-02-22	0	0					0.0	0.0					
Afghanistan	AFG	2021-02-23						1367					34	1367	0.003
Afghanistan	AFG	2021-02-24						1367					34	1367	0.003
Afghanistan	AFG	2021-02-25						1367					34	1367	0.003
Afghanistan	AFG	2021-02-26						1367					34	1367	0.003
Afghanistan	AFG	2021-02-27						1367					34	1367	0.003
Afghanistan	AFG	2021-02-28	8200	8200				1367	0.02	0.02			34	1367	0.003
Afghanistan	AFG	2021-03-01						1580					40	1580	0.004
Afghanistan	AFG	2021-03-02						1794					45	1794	0.005
Afghanistan	AFG	2021-03-03						2008					50	2008	0.005
Afghanistan	AFG	2021-03-04						2221					56	2221	0.006
Afghanistan	AFG	2021-03-05						2435					61	2435	0.006
Afghanistan	AFG	2021-03-06						2649					66	2649	0.007
Afghanistan	AFG	2021-03-07						2862					72	2862	0.007
Afghanistan	AFG	2021-03-08						2862					72	2862	0.007
Afghanistan	AFG	2021-03-09						2862					72	2862	0.007
Afghanistan	AFG	2021-03-10						2862					72	2862	0.007

<u>vaccinations_by_manufacturer.csv:</u>

This csv file has following information:

- location: name of the country (or region within a country).
- date: date of the observation.

- vaccine: brand of vaccine used in the country.
- total_vaccinations: total number of doses administered. For vaccines that require multiple doses, each individual dose is counted. If a person receives one dose of the vaccine, this metric goes up by 1. If they receive a second dose, it goes up by 1 again. If they receive a third/booster dose, it goes up by 1 again.

vaccinations-by-manufacturer

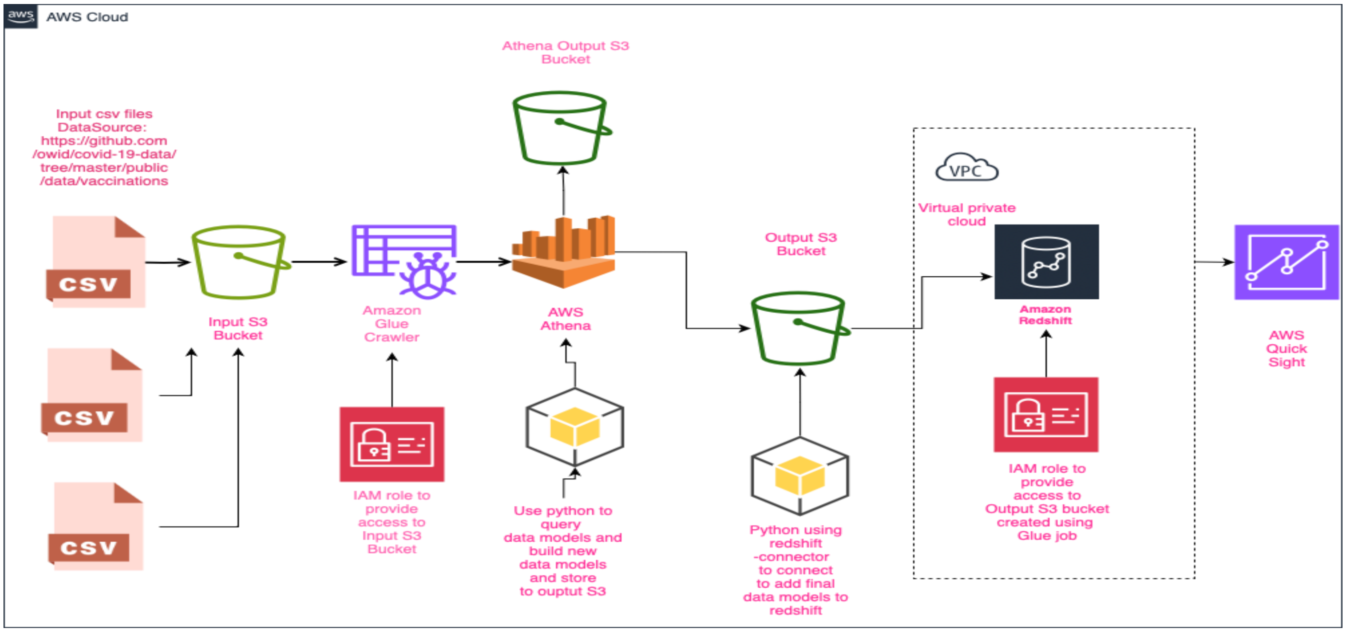
location	date	vaccine	total_vaccinations
Argentina	2020-12-29	Moderna	2
Argentina	2020-12-29	Oxford/AstraZeneca	7
Argentina	2020-12-29	Pfizer/BioNTech	1
Argentina	2020-12-29	Sinopharm/Beijing	3
Argentina	2020-12-29	Sputnik V	20484
Argentina	2020-12-30	Moderna	2
Argentina	2020-12-30	Oxford/AstraZeneca	7
Argentina	2020-12-30	Pfizer/BioNTech	1
Argentina	2020-12-30	Sinopharm/Beijing	3
Argentina	2020-12-30	Sputnik V	40588
Argentina	2020-12-31	Moderna	2
Argentina	2020-12-31	Oxford/AstraZeneca	7
Argentina	2020-12-31	Pfizer/BioNTech	1
Argentina	2020-12-31	Sinopharm/Beijing	3
Argentina	2020-12-31	Sputnik V	43394
Argentina	2021-01-01	Moderna	2

2. System Architecture:

This project is implemented using Amazon web services. AWS is chosen because of the following reasons:

- AWS provides scalable and cost-effective cloud computing solutions.
- It is one of the best functional cloud providers.
- AWS provides the most secure and flexible cloud computing environments.
- It is 'Pay as you use' model and is more reliable.

System architecture at a high level is shown below:



The main components of this architecture are as follows:

1. Amazon S3
    a. S3 is a service of AWS that is used to store, protect and retrieve huge data.
    b. It can be used to store ‘Big Data’ that is in any format like json, csv, text files etc.
2. Amazon Glue Crawler
    a. Glue Crawler is a component of the Amazon Glue service which is mainly used to build data catalogs that can be used to create data lakes.
    b. Crawlers can integrate with other AWS services such as S3, Athena, Redshift, EMR etc., to provide a comprehensive data processing and analysis solution.
3. AWS Athena
    a. Athena is a simple interactive query service which is serverless and can be used to query data in S3 buckets using standard SQL commands.
4. Amazon Redshift
    a. Amazon Redshift is a completely managed data warehouse offered as a cloud service by Amazon.
    b. It is very easy to process and analyze huge terabytes of data as it can be configured to have various cluster sizes and cluster properties according to the data that has to be processed.
    c. It is a fast, scalable and cost-effective solution.
5. Amazon QuickSight
    a. Amazon Quick Sight is a machine learning powered AWS service to analyze data and build visualizations of data that is stored in Redshift, Athena, S3 etc.
    b. It has a very high performance and is highly interactive.

6. IAM
    a. IAM is the Identity and Access management service of AWS.
    b. It helps in creating fine grained security rules that can enable access within various AWS services and also various roles within the organization.

3. Implementation Steps:

1. Input CSV files are loaded to input S3 buckets.
2. Crawl the input CSV files in S3 buckets to build relational data model in Athena.
3. Connect to Athena through Jupyter notebook using boto3 client and query the data to store it in pandas data frame. The intermediate results of Athena can also be stored in S3 buckets which can later be used if there is any requirement.
4. The final data frame objects of step-3 are stored to an output S3 bucket
5. Connect to Amazon Redshift using redshift-connector and build final data models on Redshift using copy command
6. Query data from Redshift in Quick Sight for analysis

Experiments:

1. Input CSV files are loaded to input S3 buckets. A bucket can be created by providing the information as shown in Image 1.
    - 3 input buckets are created to store the 3 csv files:
        - locations.csv - stored in location
        - vaccinations.csv - stored in vaccinations
        - vaccinations_by_manufacturer - stored in vaccinations_by_manufacturer

2. Separate buckets are created so that crawlers can build data models which can be viewed from Athena.

Image 1:

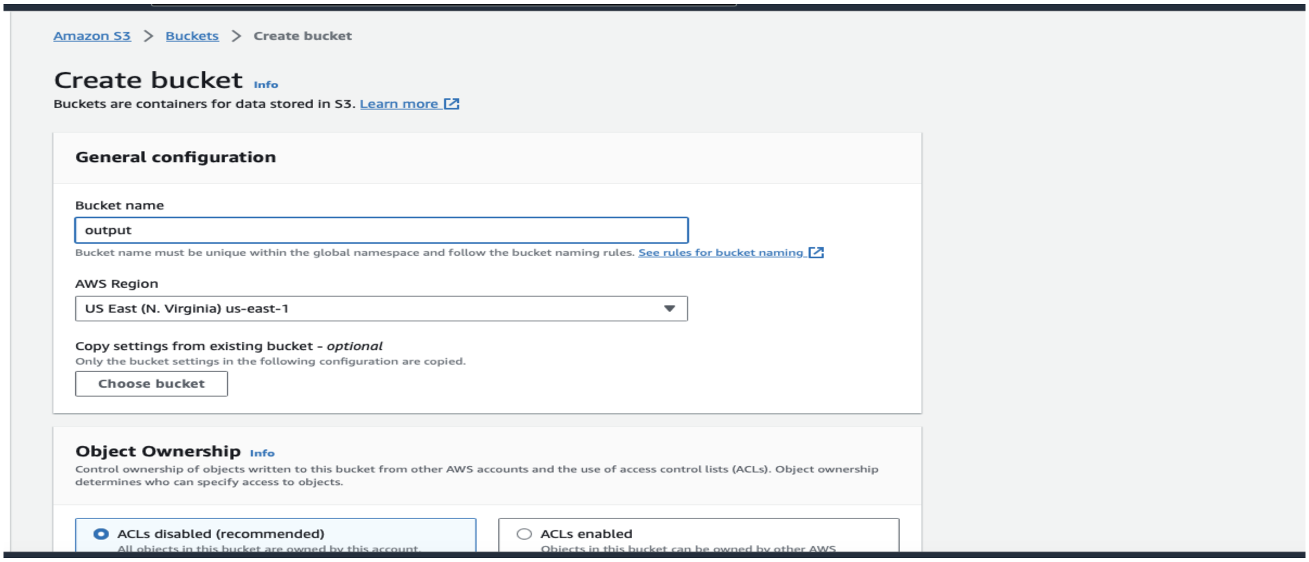
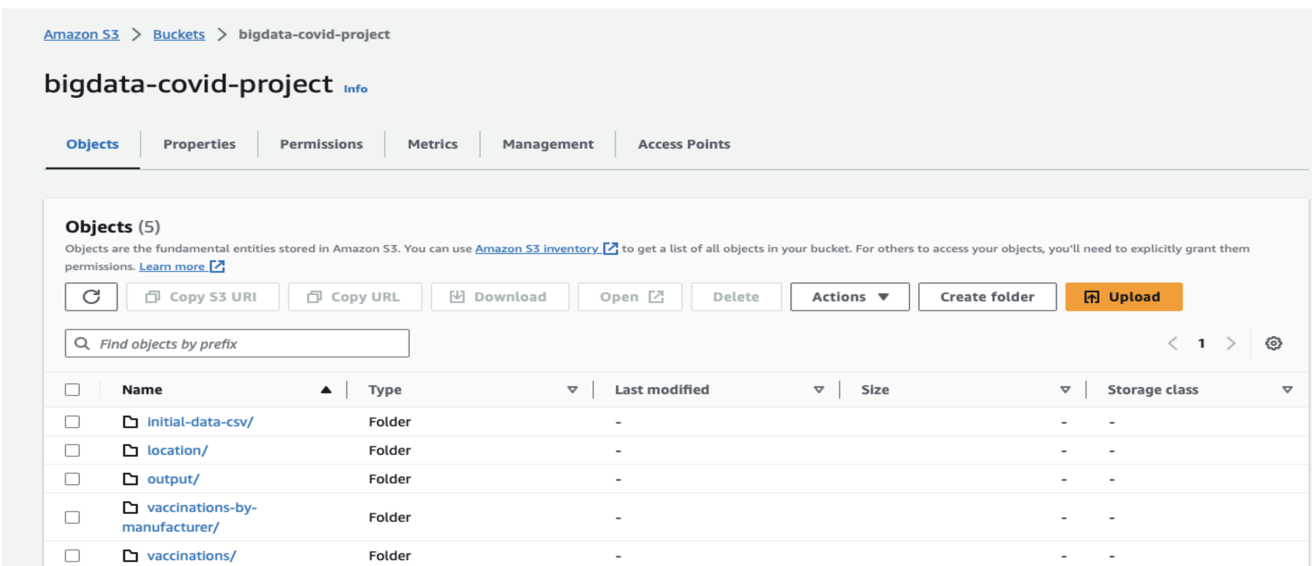


Image 2:



2. Crawl the input CSV files in S3 buckets to build relational data model in Athena.
    - Three crawlers are created by using the create crawler option as seen in the below Image 1.
    - While creating the crawler, the input s3 bucket should be selected and for the crawler to be able to access s3 bucket, IAM rule should be created, and the name given for this role is bigdata-covid-project-s3-glue-role as seen in Image 2.
    - Finally, Image 3 shows the data models built using crawler which are available by clicking on the Data Catalog option

Image 1:

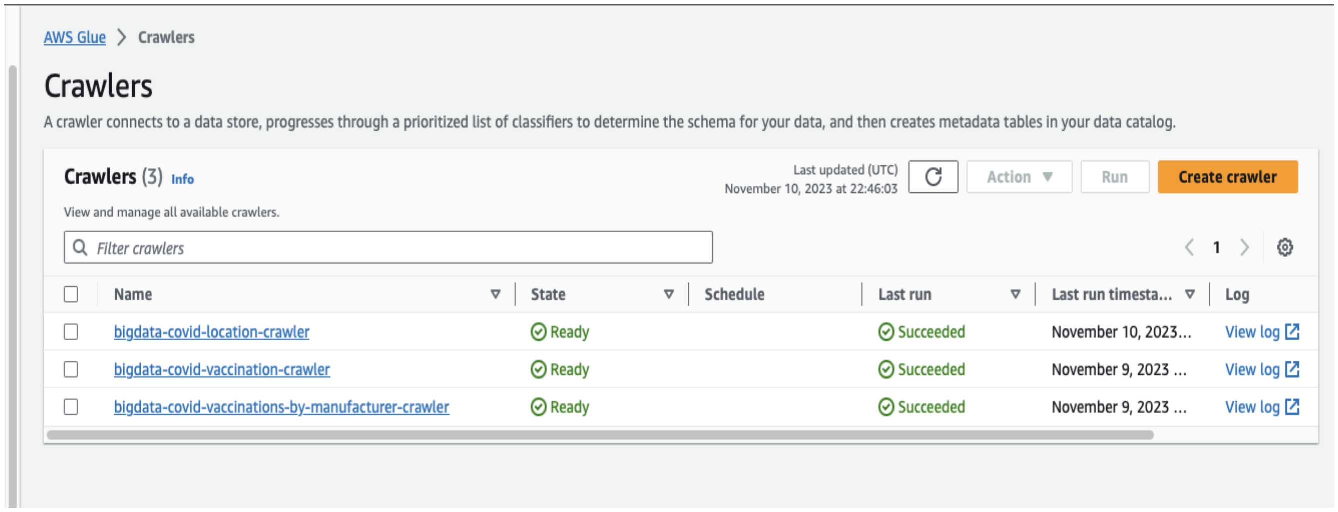


Image 2:

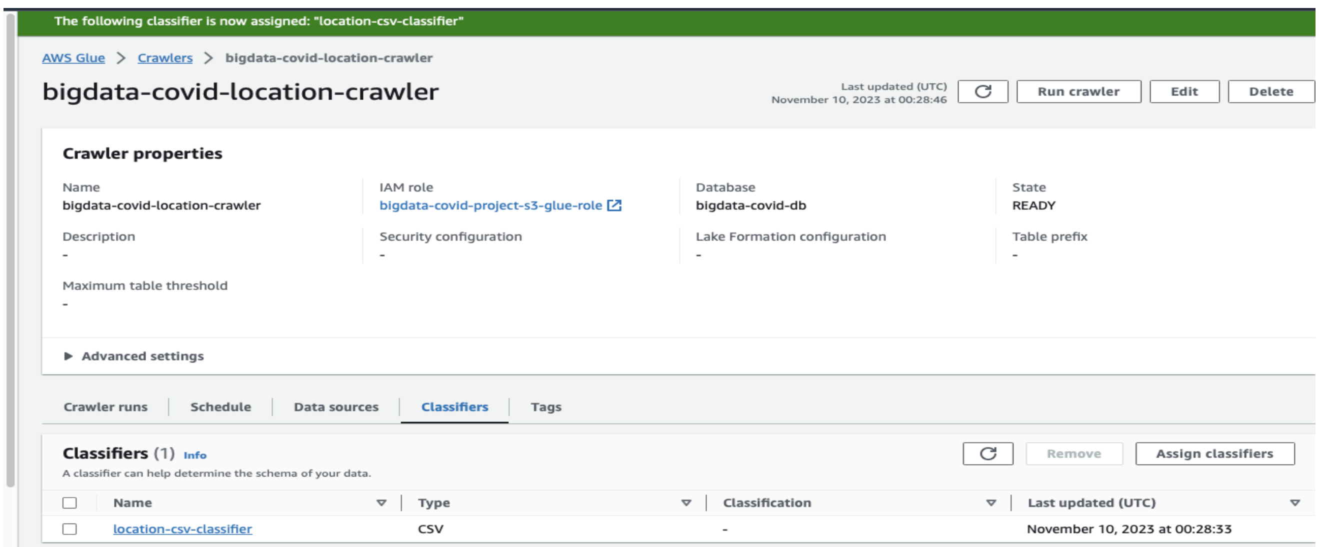
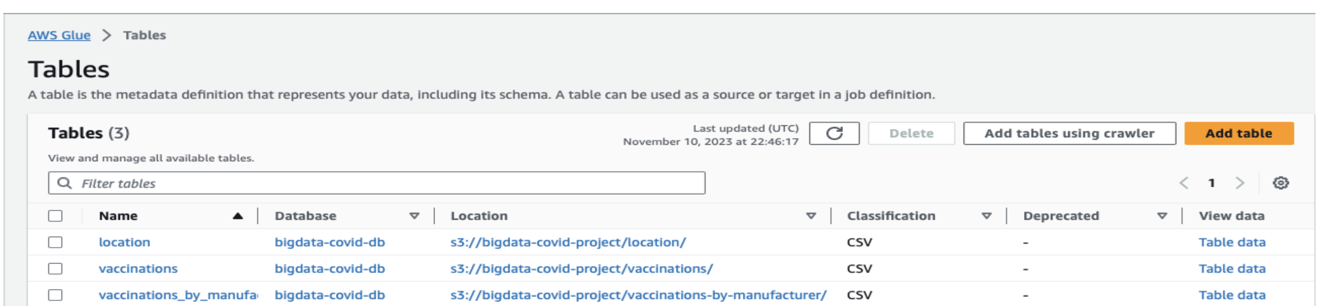


Image 3:



3. Connect to Athena through Jupyter notebook using boto3 client and query the data to store it in pandas data frame. The intermediate results of Athena can also be stored in S3 buckets which can later be used if there is any requirement.

- The models created from the above step 2 through Glue crawler can be accessed from Athena as shown in Image 1 by providing the Database name which is bigdata-covid-db that was created while creating the crawler.
- However, these data models are not the final data models that should be loaded on to the Redshift cluster and require pre-processing to build final data models. So, we connect to Athena using boto3 client with Jupyter notebook and this requires providing the boto3 client with our client id and client secret along with the input and the output s3 bucket locations as shown in Image 2. The data preprocessing can be seen in Image 3 & Image 4.

Image 1(Shows the data models built using Glue crawler in Athena):

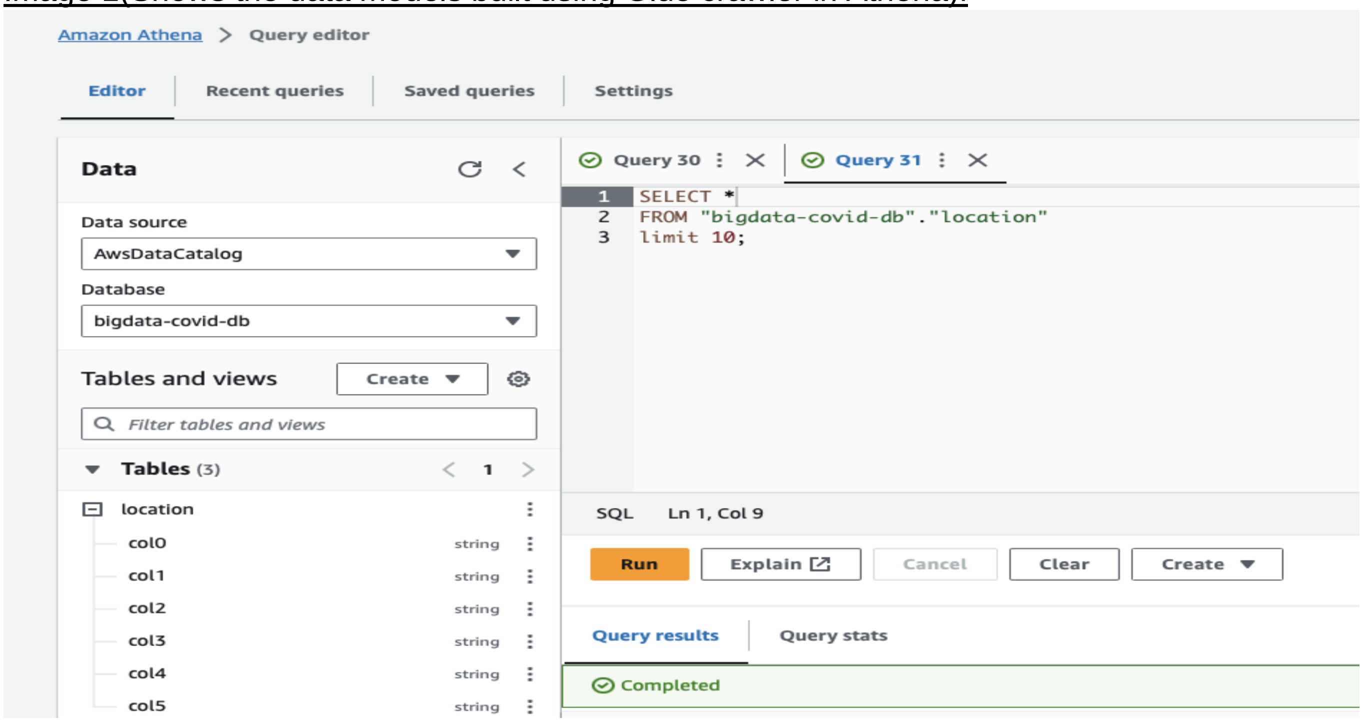


Image 2:

jupyter bigdata-covid-athena-s3-pandas Last Checkpoint: 11/10/2023 (unsaved changes)

File Edit View Insert Cell Kernel Help | Not Connected | Trusted | Python 3 (ipykernel)

In [214]:
```
import pandas as pd
from io import StringIO
import boto3
import time

AWS_ACCESS_KEY = ""
AWS_ACCESS_SECRET = ""
AWS_REGION = "us-east-1"
SCHEMA_NAME = "bigdata-covid-db"
S3_STAGING_DIR = "s3://bigdata-covid-project-athena/athena-output-data/"
S3_BUCKET_NAME = "bigdata-covid-project-athena"
S3_OUTPUT_DIRECTORY = "athena-output-data"
```

In [215]:
```
athena_client = boto3.client("athena",
                             aws_access_key_id = AWS_ACCESS_KEY,
                             aws_secret_access_key = AWS_ACCESS_SECRET,
                             region_name = AWS_REGION,
                             )
```

In [216]:
```
#querying athena and storing into pandas dataframe
Dict = {}
def download_and_load_query_results(client: boto3.client,
                                    query_response: Dict
                                   )-> pd.DataFrame:
    while True:
        try:
            client.get_query_results(
            QueryExecutionId = query_response["QueryExecutionId"])
            break
        except Exception as err:
            if "not yet finished" in str(err):
                time.sleep(0.001)
            else:
                raise err
    temp_file_location: str = "athena_query_results.csv"
    s3_client = boto3.client(
    "s3",
    aws_access_key_id=AWS_ACCESS_KEY,
    aws_secret_access_key=AWS_ACCESS_SECRET,
    region_name=AWS_REGION,
    )
    s3_client.download_file(
    S3_BUCKET_NAME,
    f"{S3_OUTPUT_DIRECTORY}/{query_response['QueryExecutionId']}.csv",
    temp_file_location,
    )
    return pd.read_csv(temp_file_location)
response = athena_client.start_query_execution(
    QueryString="SELECT * FROM vaccinations",
    QueryExecutionContext={"Database": SCHEMA_NAME},
    ResultConfiguration={
        "OutputLocation": S3_STAGING_DIR,
        "EncryptionConfiguration": {"EncryptionOption": "SSE_S3"},
    },
)
response
```

Image 3:

In [221]:
```
#Fixing the column names in location table by using iloc function in pandas.This is a data cleaning step.
#get the first row
new_column_name = location.iloc[0]
#remove the first row and store in the same data frame
location = location[1:]
#assign location column names with new column name stored above
location.columns = new_column_name
location.head()
```

Out[221]:

	location	iso_code	vaccines	last_observation_date	source_name	source_website
1	Afghanistan	AFG	CanSino, Covaxin, Johnson&Johnson, Moderna, Ox...	2022-06-01	World Health Organization	https://covid19.who.int/
2	Albania	ALB	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ...	2022-05-22	World Health Organization	https://covid19.who.int/
3	Algeria	DZA	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...	2022-05-01	World Health Organization	https://covid19.who.int/
4	Andorra	AND	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	2022-05-22	World Health Organization	https://covid19.who.int/
5	Angola	AGO	Oxford/AstraZeneca	2022-05-22	World Health Organization	https://covid19.who.int/

In [222]:
```
#Fetching only location and vaccines from location and storing in new data frame
location_and_vaccines = location[['location','vaccines']]
location_and_vaccines.head()
```

Out[222]:

	location	vaccines
1	Afghanistan	CanSino, Covaxin, Johnson&Johnson, Moderna, Ox...
2	Albania	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ...
3	Algeria	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
4	Andorra	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
5	Angola	Oxford/AstraZeneca

Image 4:

In [223]:
```python
#Fetching only location and last_observation_date from location and storing in new data frame
location_and_last_observation_date = location[['location','last_observation_date']]
location_and_last_observation_date['last_observation_date'] = pd.to_datetime(location_and_last_observation_dat
location_and_last_observation_date.head()
```

```
/var/folders/0n/rbsnm7dj6r567xfgpjyvkx440000gn/T/ipykernel_85427/2641038401.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#r
eturning-a-view-versus-a-copy
  location_and_last_observation_date['last_observation_date'] = pd.to_datetime(location_and_last_observation_
date['last_observation_date'])
```

Out[223]:

	location	last_observation_date
1	Afghanistan	2022-06-01
2	Albania	2022-05-22
3	Algeria	2022-05-01
4	Andorra	2022-05-22
5	Angola	2022-05-22

In [224]:
```python
#Fetching only vaccine and total_vaccinations from vaccinations_by_manufacturer and storing in new data frame
vaccinations_by_manufacturer_and_total_vaccinations = vaccinations_by_manufacturer[['vaccine','total_vaccinati

vaccinations_by_manufacturer_and_total_vaccinations.head()
```

Out[224]:

	vaccine	total_vaccinations
0	Moderna	2
1	Oxford/AstraZeneca	7
2	Pfizer/BioNTech	1
3	Sinopharm/Beijing	3
4	Sputnik V	20484

In [225]:
```python
#Fetching only location and daily_vaccinations from vaccinations and storing in new data frame
location_and_daily_vaccinations = vaccinations[['location','daily_vaccinations']]
location_and_daily_vaccinations.head()
```

Out[225]:

	location	daily_vaccinations
0	Afghanistan	NaN
1	Afghanistan	1367.0
2	Afghanistan	1367.0

4. The final data frame objects of step-3 are stored in an output S3 bucket.
    - After step 3, the final data models will be available in the output S3 bucket as shown in Image 1.
    - We convert the Athena data models from step 3 to our final data model using csv buffers as seen in Image 2, Image 3 using StringIO function and writing these final data models to output S3 bucket using put_object method of boto3 client.

Image 1:

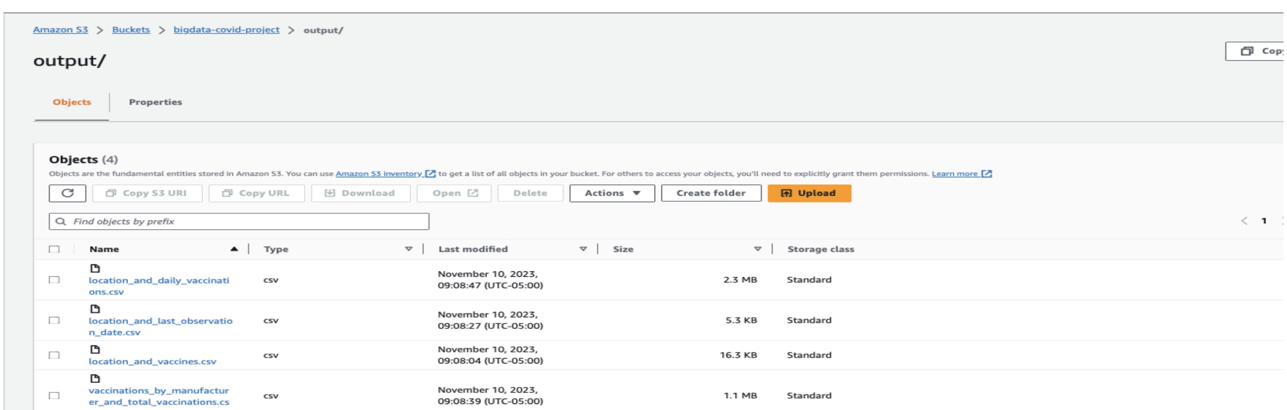


Image 2:

```
In [228]: #Currently we have all the required data from all three tables in following 4 dataframes
          #location_and_vaccines, location_and_last_observation_date, vaccinations_by_manufacturer_and_total_vaccination
          #Storing all these to an S3 bucket
          bucket = "bigdata-covid-project"
          csv_buffer = StringIO()
          csv_buffer

Out[228]: <_io.StringIO at 0x120a371c0>

In [229]: #store the dataframes into buffers which can be then stored to S3
          location_and_vaccines.to_csv(csv_buffer)
          csv_buffer.getvalue()
```


```
putnik V"\n195,Taiwan,"Medigen, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech"\n196,Tajikistan,"Moderna, Oxf
ord/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V"\n197,Tanzania,"Johnson&Johnson, Pfizer/BioNTech, Sino
pharm/Beijing"\n198,Thailand,"Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac"\n19
9,Timor,"Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac"\n200,Togo,Oxford/AstraZeneca\n201,Tokelau,Pfizer/Bio
NTech\n202,Tonga,Oxford/AstraZeneca\n203,Trinidad and Tobago,"Johnson&Johnson, Oxford/AstraZeneca, Pfizer/B
ioNTech, Sinopharm/Beijing"\n204,Tunisia,"CanSino, Covaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, P
fizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V"\n205,Turkey,"Pfizer/BioNTech, Sinova
c, Turkovac"\n206,Turkmenistan,"EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2
001"\n207,Turks and Caicos Islands,Pfizer/BioNTech\n208,Tuvalu,Oxford/AstraZeneca\n209,Uganda,"Johnson&John
son, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac"\n210,Ukraine,"Johnson&Johnson, Moderna, Oxford/
AstraZeneca, Pfizer/BioNTech, Sinovac"\n211,United Arab Emirates,"Oxford/AstraZeneca, Pfizer/BioNTech, Sino
pharm/Beijing, Sinopharm/Wuhan, Sputnik V"\n212,United Kingdom,"Moderna, Oxford/AstraZeneca, Pfizer/BioNTec
h"\n213,United States,"Johnson&Johnson, Moderna, Pfizer/BioNTech"\n214,Uruguay,"Oxford/AstraZeneca, Pfizer/
BioNTech, Sinovac"\n215,Uzbekistan,"Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik Light, S
putnik V, ZF2001"\n216,Vanuatu,Oxford/AstraZeneca\n217,Venezuela,"Abdala, Sinopharm/Beijing, Sinovac, Sober
ana02, Sputnik Light, Sputnik V"\n218,Vietnam,"Abdala, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinoph
arm/Beijing, Sputnik V"\n219,Wales,"Moderna, Oxford/AstraZeneca, Pfizer/BioNTech"\n220,Wallis and Futuna,Mo
derna\n221,Yemen,"CanSino, Covaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinophar
m/Beijing, Sinovac, Sputnik Light, Sputnik V"\n222,Zambia,"Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/B
eijing"\n223,Zimbabwe,"Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V"\n'
```


```
In [230]: #Create S3 object and storing the buffers into the S3
          ACCESS_KEY = ""
          ACCESS_SECRET = ""
          REGION = "us-east-1"
          s3 = boto3.client(
              's3',
              aws_access_key_id=ACCESS_KEY,
              aws_secret_access_key=ACCESS_SECRET,
              region_name=REGION,
              )
          #csv_buffer.seek(0)
          s3.put_object(Bucket=bucket, Body=csv_buffer.getvalue(), Key='output/location_and_vaccines.csv')
```


```
Out[230]: {'ResponseMetadata': {'RequestId': 'HAF5YFGM1EE5AYZ4',
            'HostId': 'C7d0s8efv04SEnjQjHEvVoFvg92dgU55tMMxC9tKmMviUQKOMulgXET/yJRNFY5Remzl97UaOwU=',
            'HTTPStatusCode': 200,
            'HTTPHeaders': {'x-amz-id-2': 'C7d0s8efv04SEnjQjHEvVoFvg92dgU55tMMxC9tKmMviUQKOMulgXET/yJRNFY5Remzl97UaOwU
          =',
             'x-amz-request-id': 'HAF5YFGM1EE5AYZ4',
             'date': 'Fri, 10 Nov 2023 14:08:04 GMT',
             'x-amz-server-side-encryption': 'AES256',
```


Image 3:

```
In [231]: #load location_and_last_observation_date to s3
          csv_buffer_1 = StringIO()
          csv_buffer_1
          location_and_last_observation_date.to_csv(csv_buffer_1)
          csv_buffer_1.getvalue()
          s3.put_object(Bucket=bucket, Body=csv_buffer_1.getvalue(), Key='output/location_and_last_observation_date.csv'
```


```
Out[231]: {'ResponseMetadata': {'RequestId': '3KB5TDPACDF0E084',
            'HostId': 'tQLG5u40VsfoG+9PQPUYblGQoJHQP6lQj60imrYngEEC+3WTXRzetq1GkE4fC62uaXp7y3YqSuc=',
            'HTTPStatusCode': 200,
            'HTTPHeaders': {'x-amz-id-2': 'tQLG5u40VsfoG+9PQPUYblGQoJHQP6lQj60imrYngEEC+3WTXRzetq1GkE4fC62uaXp7y3YqSuc
          =',
             'x-amz-request-id': '3KB5TDPACDF0E084',
             'date': 'Fri, 10 Nov 2023 14:08:27 GMT',
             'x-amz-server-side-encryption': 'AES256',
             'etag': '"b09172979633b6179e69453456d55dea"',
             'server': 'AmazonS3',
             'content-length': '0'},
            'RetryAttempts': 0},
           'ETag': '"b09172979633b6179e69453456d55dea"',
           'ServerSideEncryption': 'AES256'}
```


```
In [232]: inations_by_manufacturer_and_total_vaccinations to s3
          _2 = StringIO()
          _2
          ns_by_manufacturer_and_total_vaccinations.to_csv(csv_buffer_2)
          .getvalue()
          ect(Bucket=bucket, Body=csv_buffer_2.getvalue(), Key='output/vaccinations_by_manufacturer_and_total_vaccinatio
```


```
Out[232]: {'ResponseMetadata': {'RequestId': 'D3H6G2ZZJDWH9V2D',
            'HostId': 'yjOOYEXkAaYT4aP1SsnIHiL/7atyjdm3Rlyg62572pGhQSfyF6ldbjrqqoGDcFgUOqTPBsCX9RE=',
            'HTTPStatusCode': 200,
            'HTTPHeaders': {'x-amz-id-2': 'yjOOYEXkAaYT4aP1SsnIHiL/7atyjdm3Rlyg62572pGhQSfyF6ldbjrqqoGDcFgUOqTPBsCX9RE
          =',
             'x-amz-request-id': 'D3H6G2ZZJDWH9V2D',
             'date': 'Fri, 10 Nov 2023 14:08:39 GMT',
             'x-amz-server-side-encryption': 'AES256',
             'etag': '"5c7dbf0903114024e795d16b133cb7c4"',
             'server': 'AmazonS3',
             'content-length': '0'},
            'RetryAttempts': 0},
           'ETag': '"5c7dbf0903114024e795d16b133cb7c4"',
           'ServerSideEncryption': 'AES256'}
```


```
In [233]: #load location_and_daily_vaccinations to s3
          csv_buffer_3 = StringIO()
          csv_buffer_3
          location_and_daily_vaccinations.to_csv(csv_buffer_3)
          csv_buffer_3.getvalue()
          s3.put_object(Bucket=bucket, Body=csv_buffer_3.getvalue(), Key='output/location_and_daily_vaccinations.csv')
```


```
Out[233]: {'ResponseMetadata': {'RequestId': '9F1ZMX207477Z29B',
            'HostId': 'OONjrbIHFlZXDWsdg/nkjWVyUcBXzfVIfzCvbQNMjhthqIqSt71nR4GaBnjNrxLrORMqwqghVpo=',
            'HTTPStatusCode': 200,
            'HTTPHeaders': {'x-amz-id-2': 'OONjrbIHFlZXDWsdg/nkjWVyUcBXzfVIfzCvbQNMjhthqIqSt71nR4GaBnjNrxLrORMqwqghVpo
          =',
             'x-amz-request-id': '9F1ZMX207477Z29B',
```

5. Connect to Amazon Redshift using redshift-connector and build final data models on Redshift using copy command.
    - Amazon Redshift cluster is created using create cluster option as shown in Image 1 and Image 2, which is the data lake that contains all the final data models
    - The final data models from step 4 can be used to extract the schema using Pandas get_schema method as shown in Image 3.
    - Once we have the schema ready for all the 4 final data models, Redshift connector can be used to create tables on Redshift after which the tables are loaded with the data from the final output S3 bucket using copy command as seen in Image 4.
    - The final 4 data models i.e., location_and_vaccines, location_and_last_observation_date, location_and_daily_vaccinations, vaccinations_by_manufacturer_and_total_vaccinations can be seen from redshift query editor as seen in Image 5 & Image 6.

Image 1:

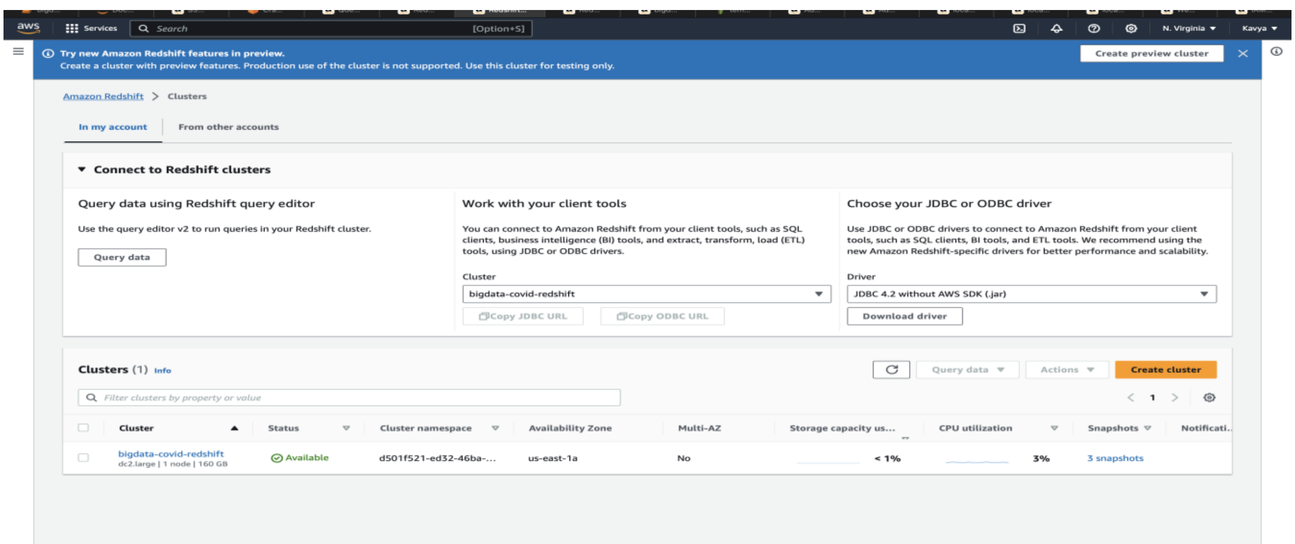


Image 2:

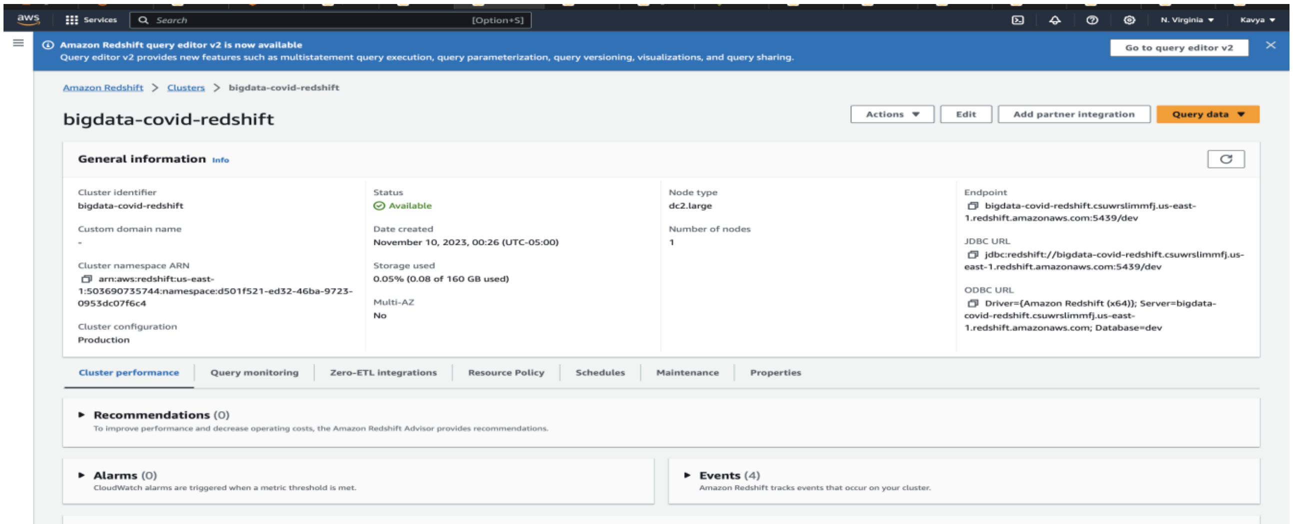


Image 3:

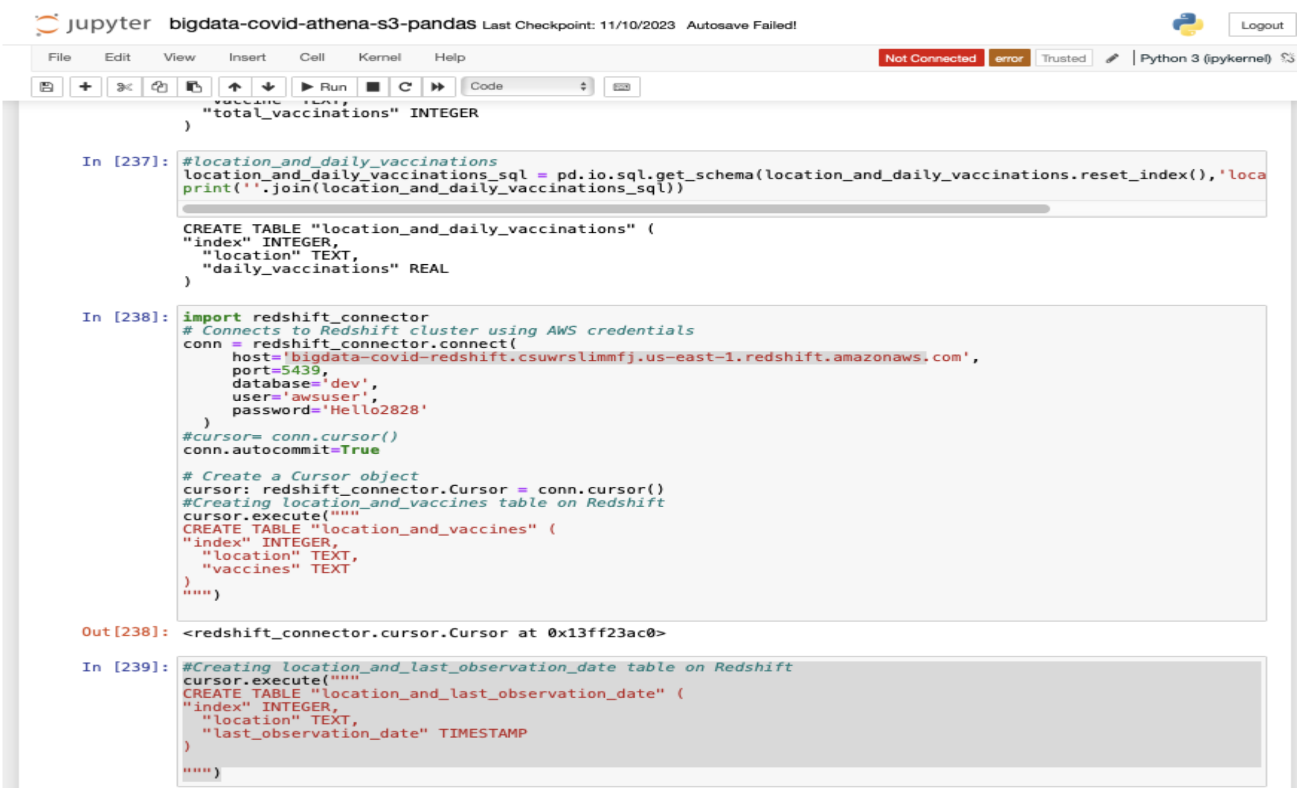


Image 4:

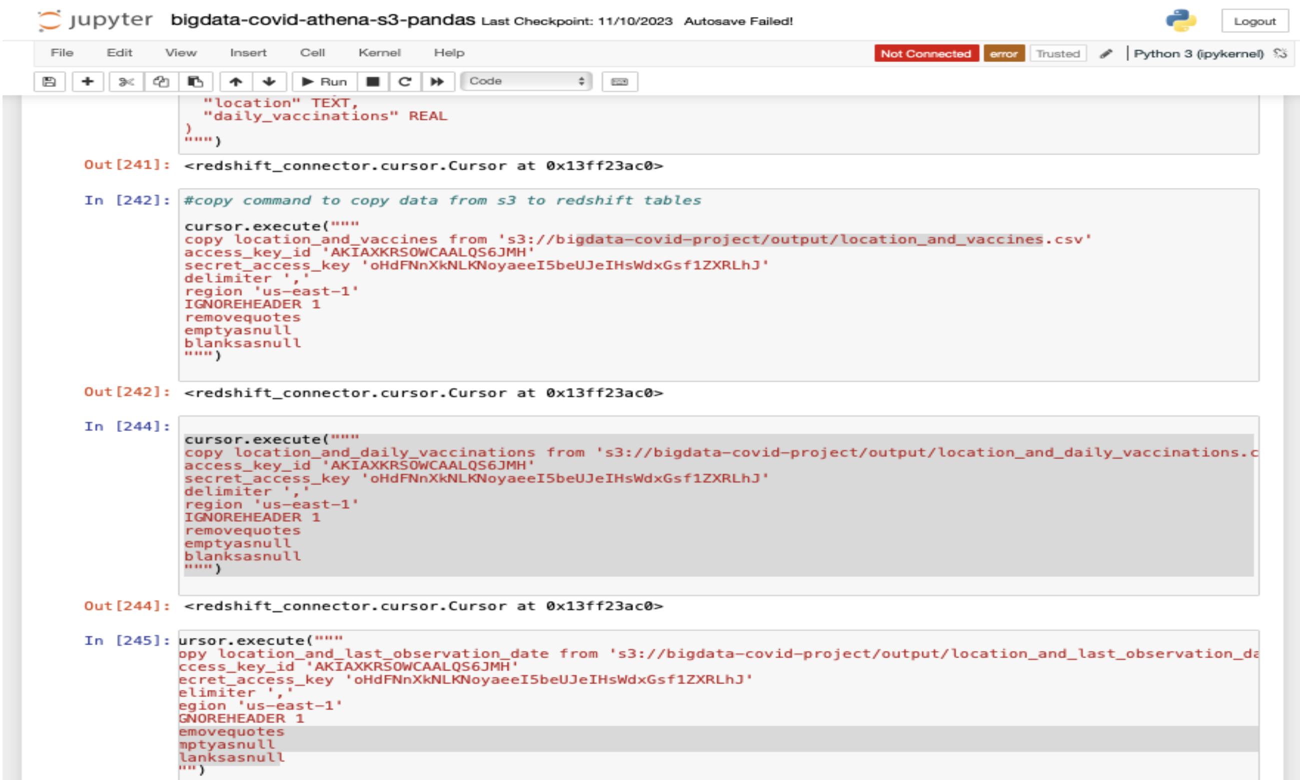


Image 5:

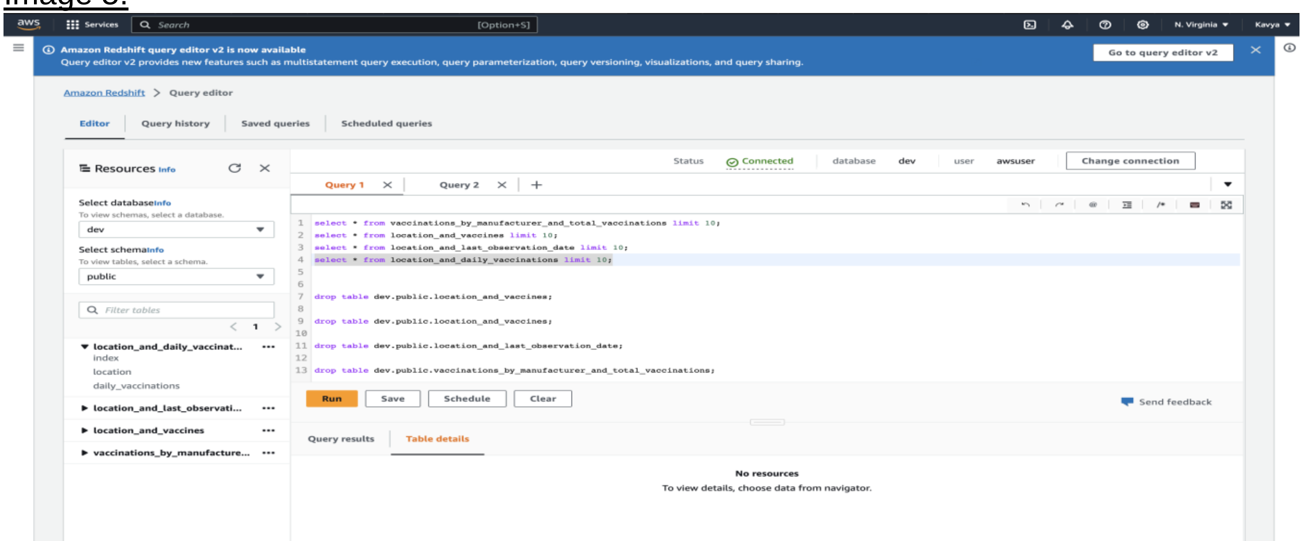
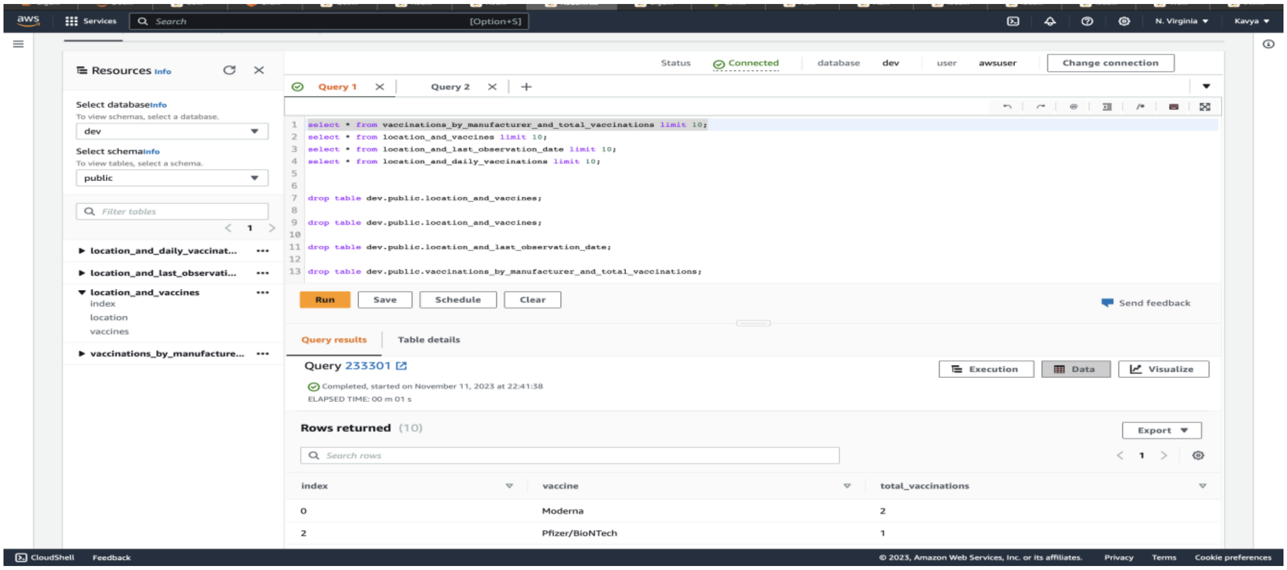


Image 6:



6. Query data from Redshift in Quick Sight for analysis.
    - Amazon Quick Sight account is created as shown in Image 1, which is a great service for analyzing the data in the Redshift cluster
    - Quick Sight is used to plot different histograms, pie charts etc., with the data and useful insights are drawn which can be seen in the results section.

Image 1:

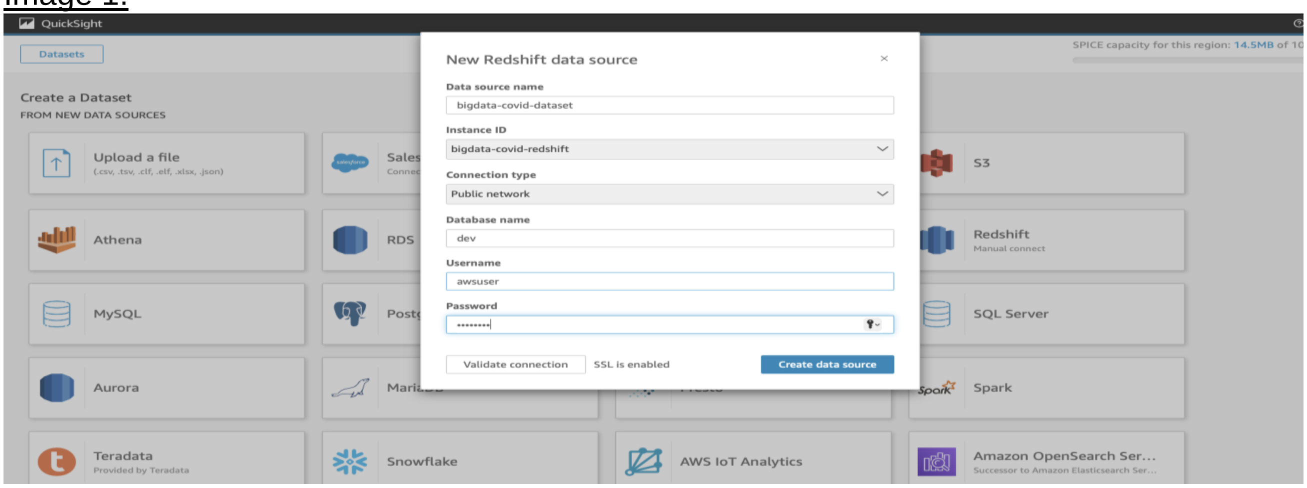
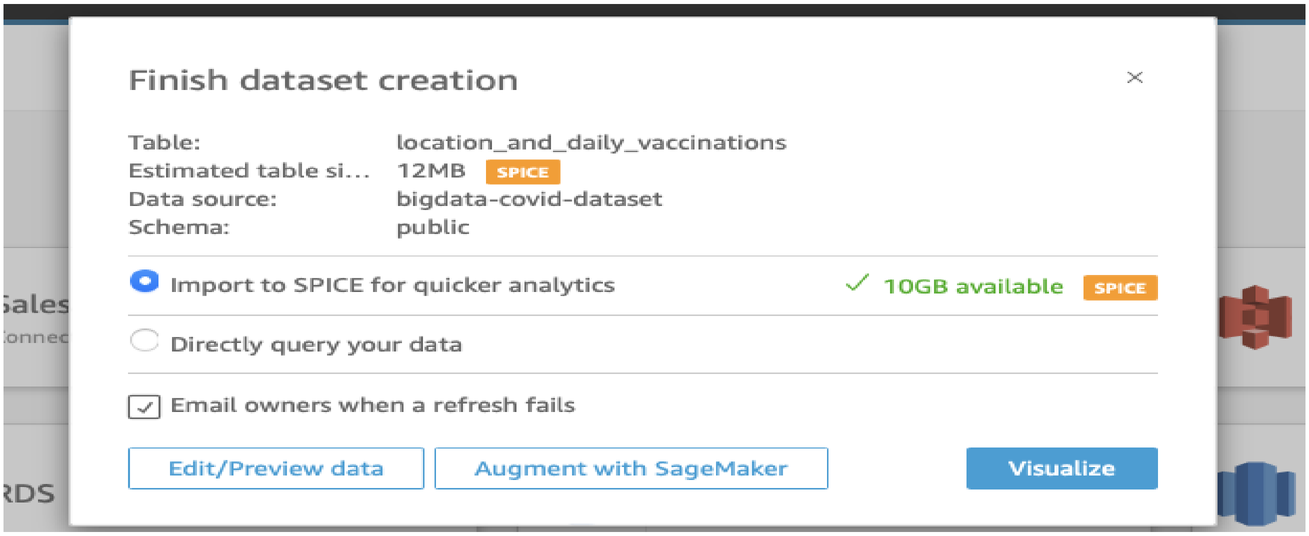


Image 2:



Results:

- Various histograms and charts are plotted to draw valuable insights from the final data models as shown in the images below.

Some of the insights from the data are:

- **Top 3 vaccine brands** for total total_vaccinations are: **Pfizer/BioNTech** with 504,896,030,676, **Moderna** with 149,294,846,161 and **Oxford/AstraZeneca** with 57,405,253,353
- Total **total_vaccinations count is 767,258,055,838**.
- **Top 3 vaccine combinations** for total count of records are:
  - **Oxford/AstraZeneca** with 19 countries.
  - **Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech** with 19 countries.
  - **CanSino, Covaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V** with 16 countries.
- There are **totally 10 vaccine brands** that are used across all over the globe.
- **Pfizer/BioNTech** is the top brand of vaccine that is used and **covaxin** is the brand that is having a smaller number of vaccination count and is the least used vaccine.
- **Asia, China, India, Europe and North America** are the **top 5** countries with **more vaccination count** and had more exposure towards vaccination.
- **Pitcaim, Tokelau, Niue, Monteserrat, Falkland Islands** are the countries with **the least vaccination count** and such countries should be prepared

for such pandemic outbreaks and should be educated about the importance of vaccination.

Image 1: [locations and vaccines chart]

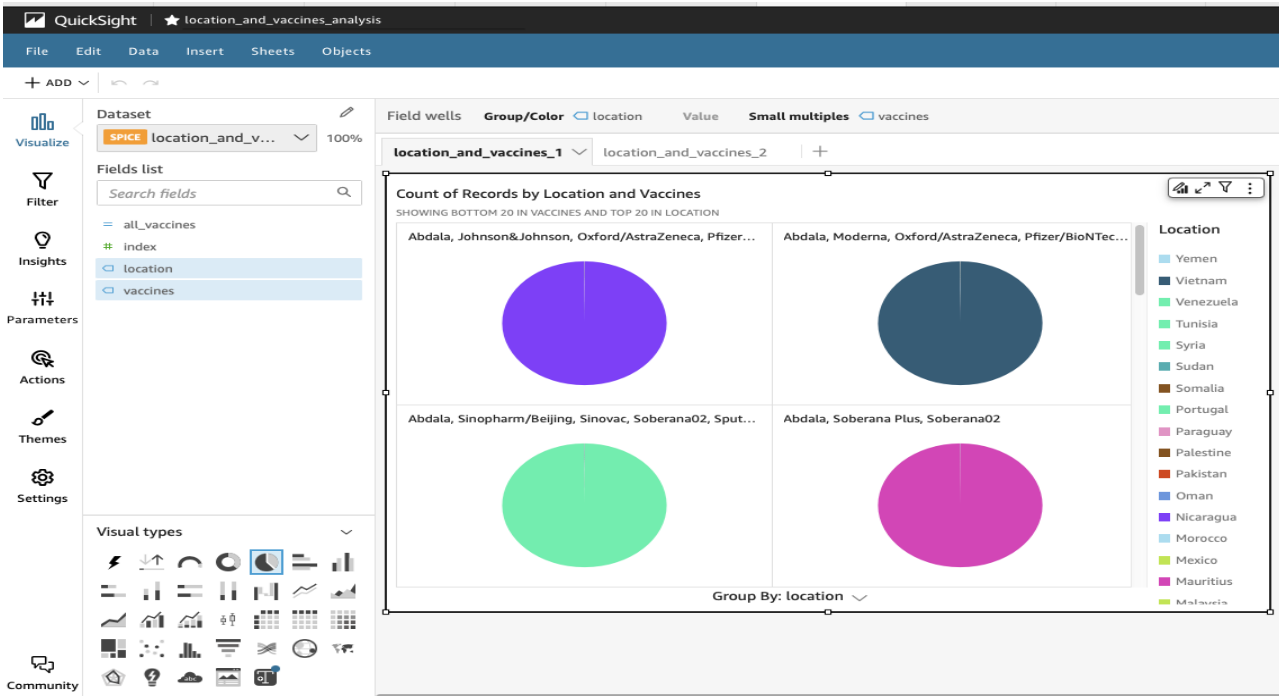


Image 2: [count of records by vaccines and locations chart]

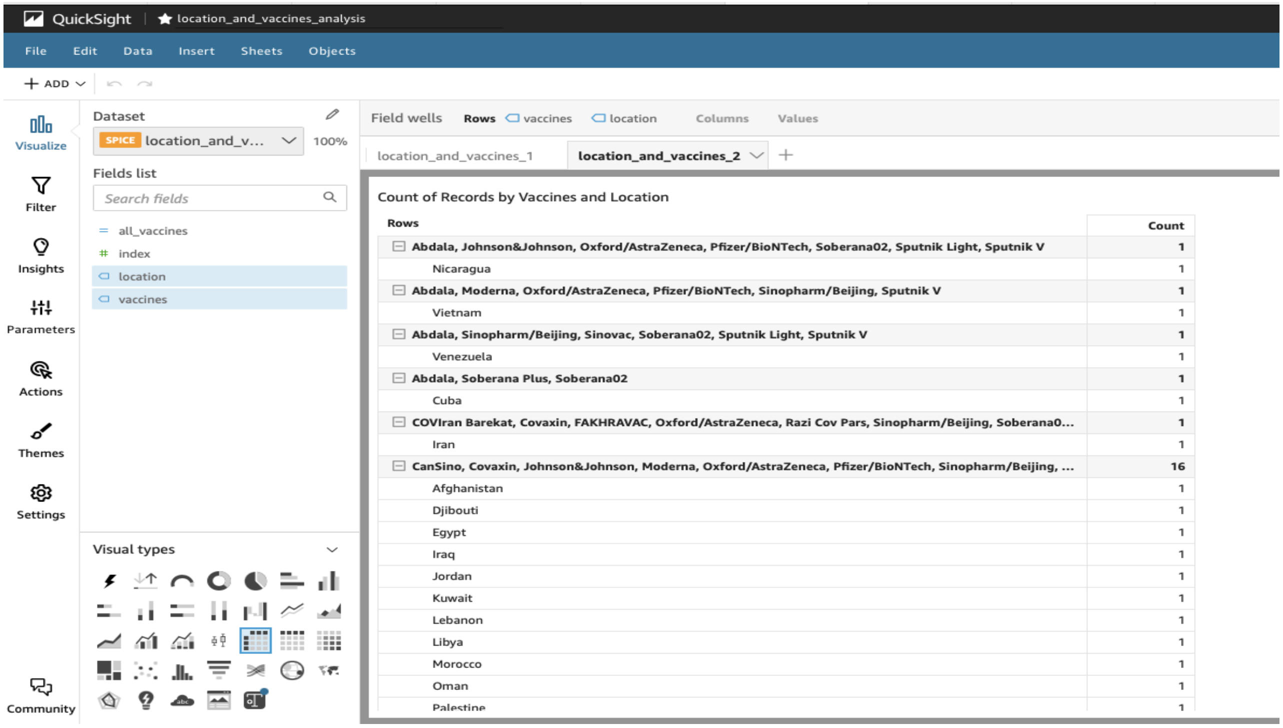


Image 3: [vaccination by manufacturer count]

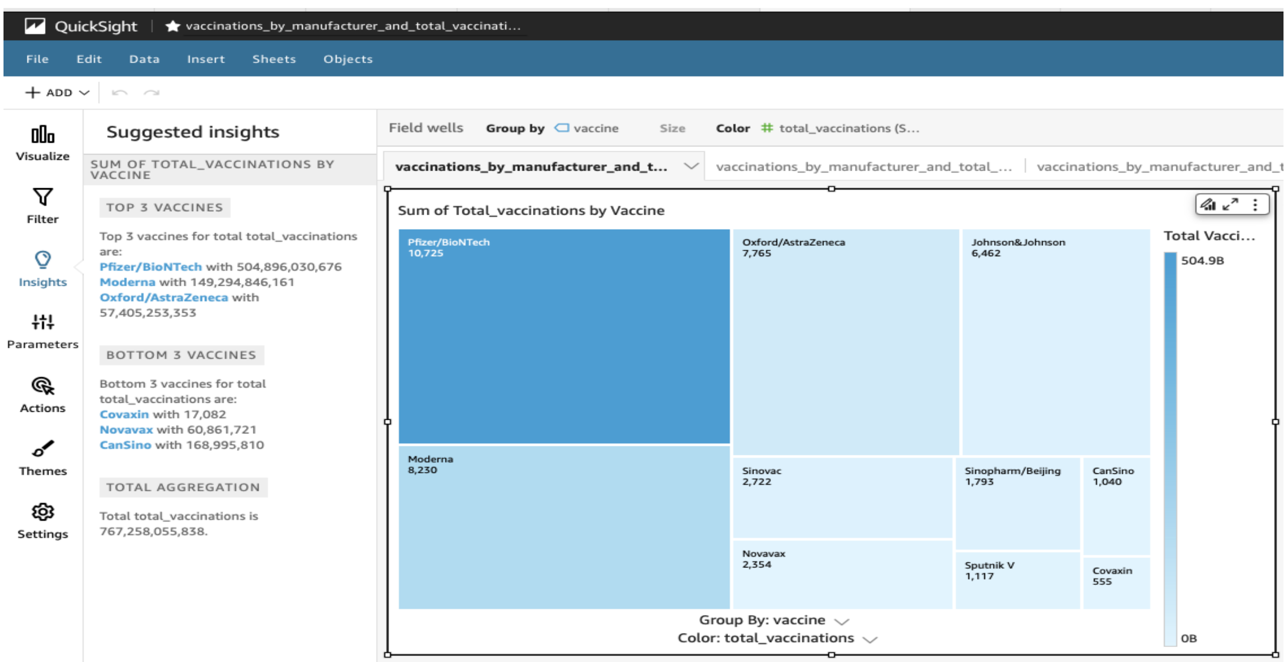


Image 4: [Pie chart representing vaccinations count by manufacturer]

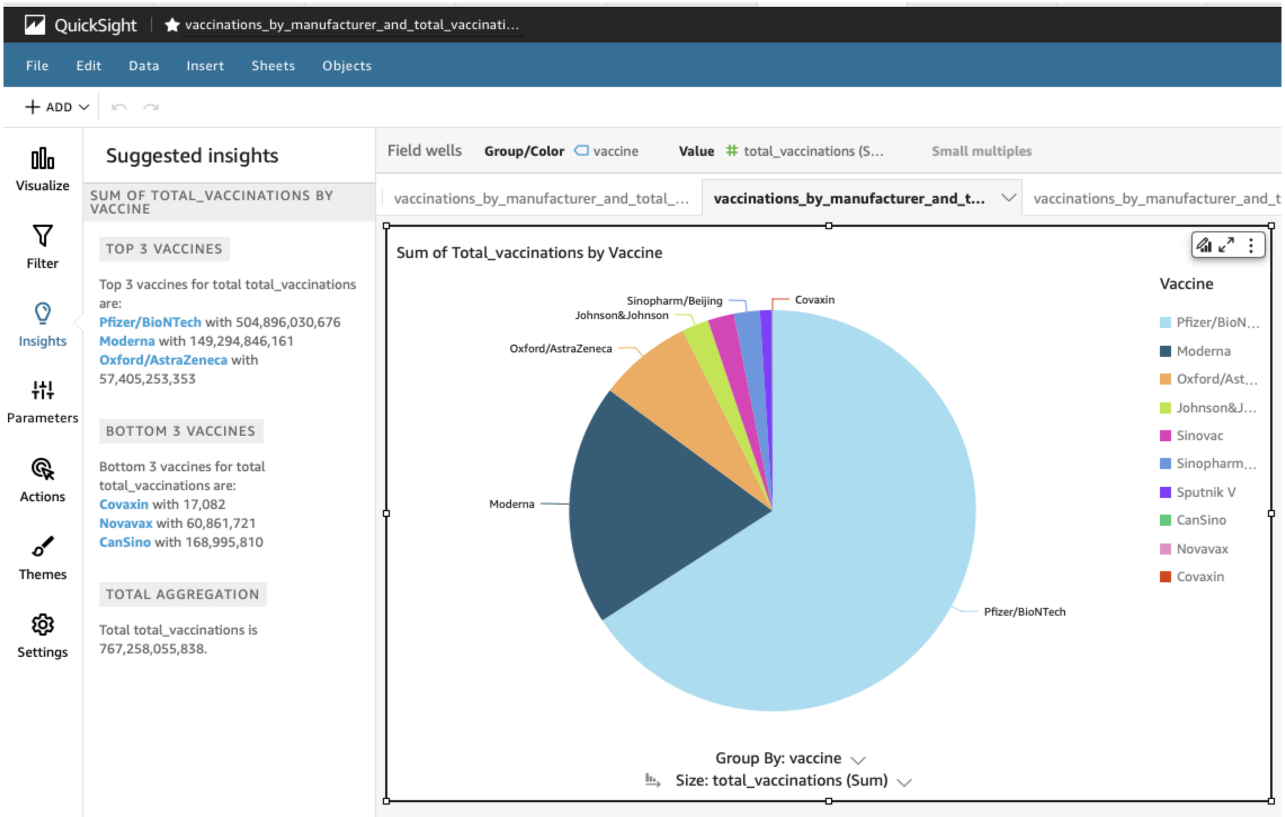


Image 5: [Table representation of vaccinations count by manufacturer]

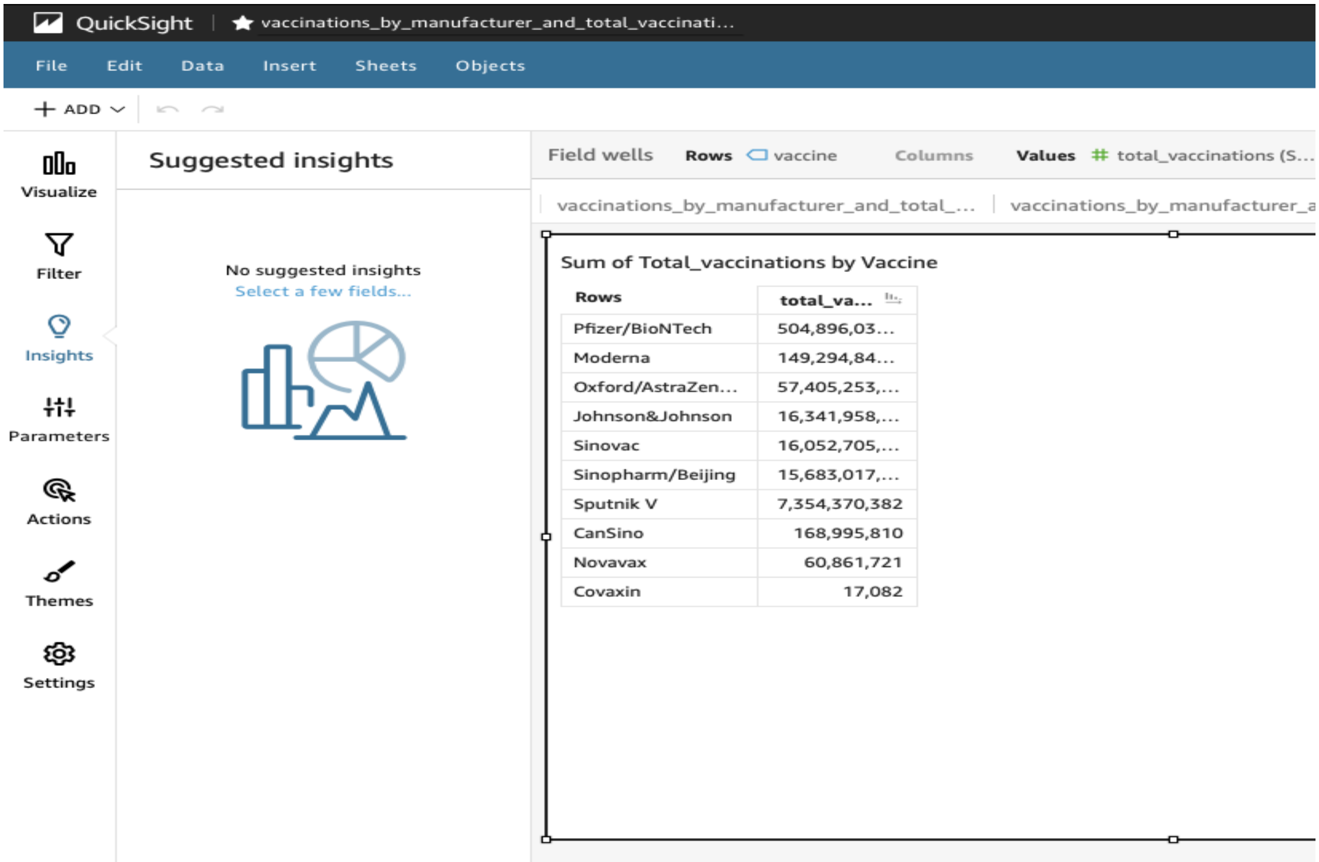


Image 6: [representing vaccinations count by manufacturer in descending order]

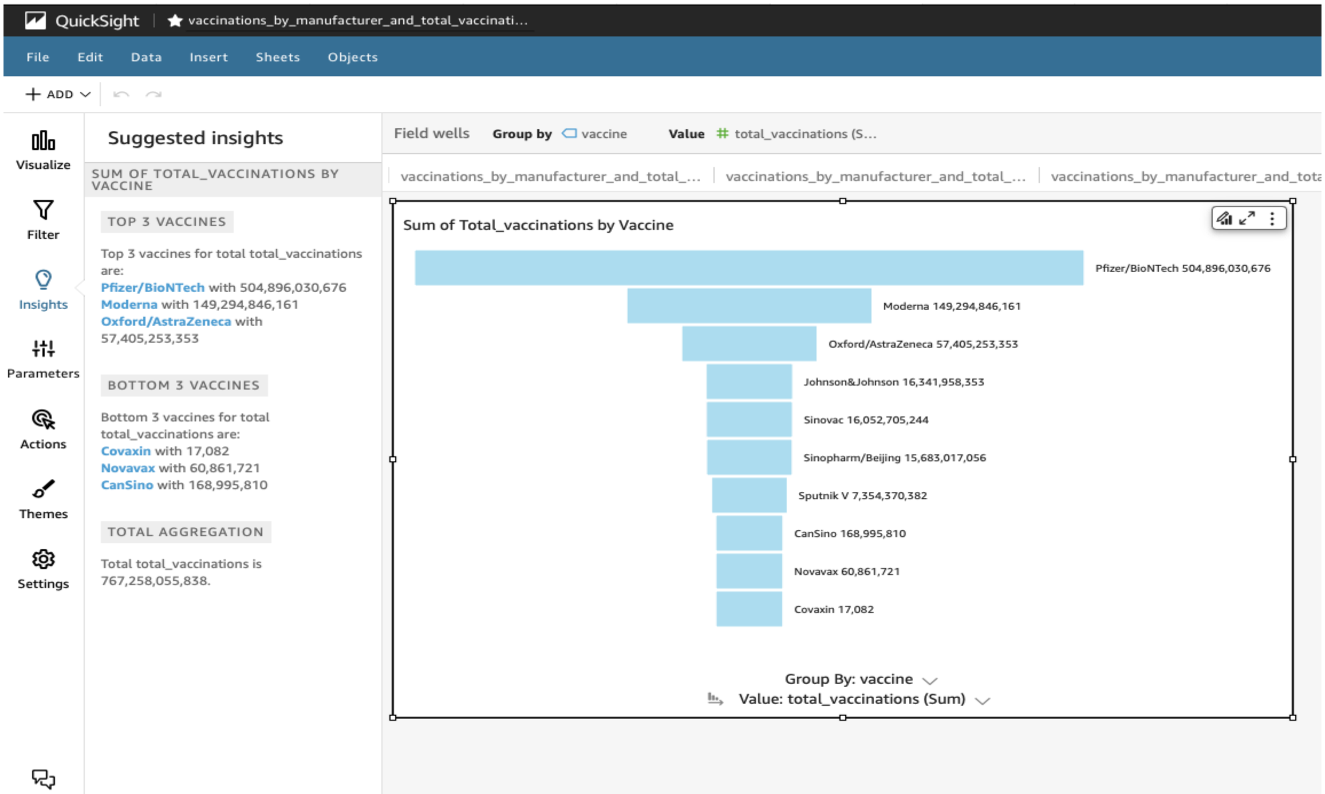


Image 7: [count of records by last observation date]

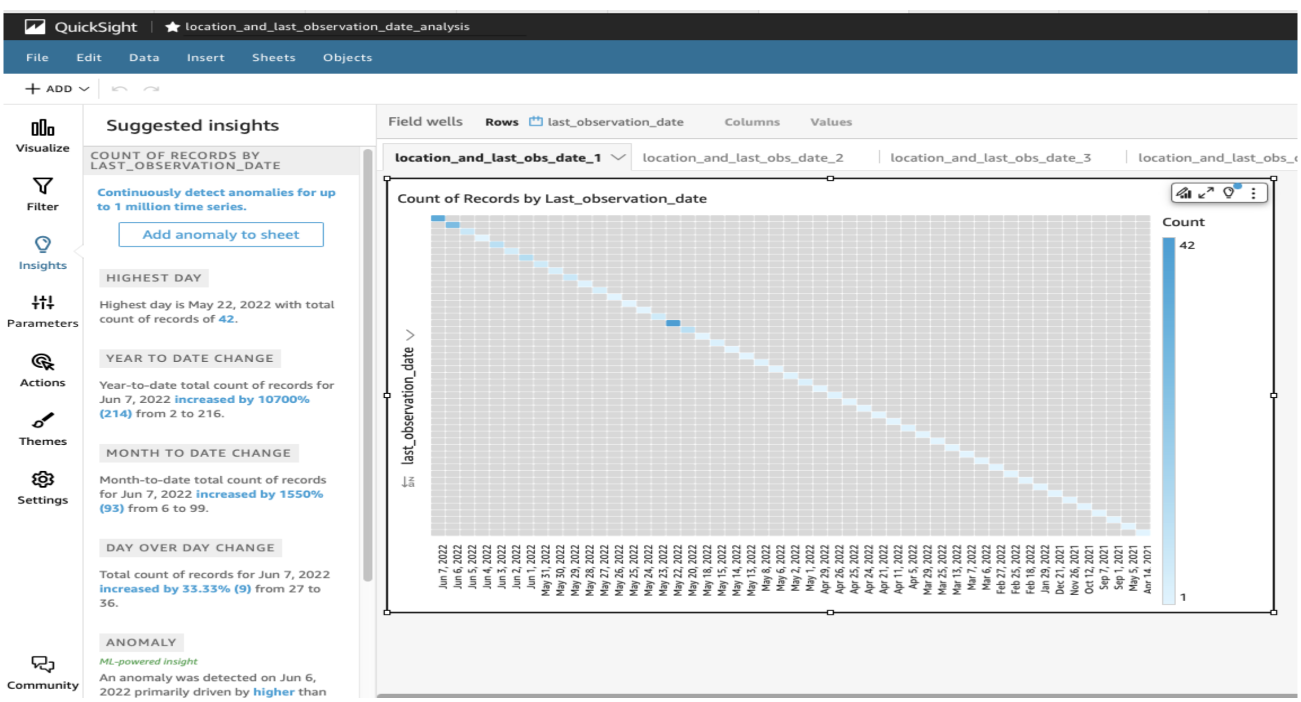


Image 8: [count of records by last observation date]

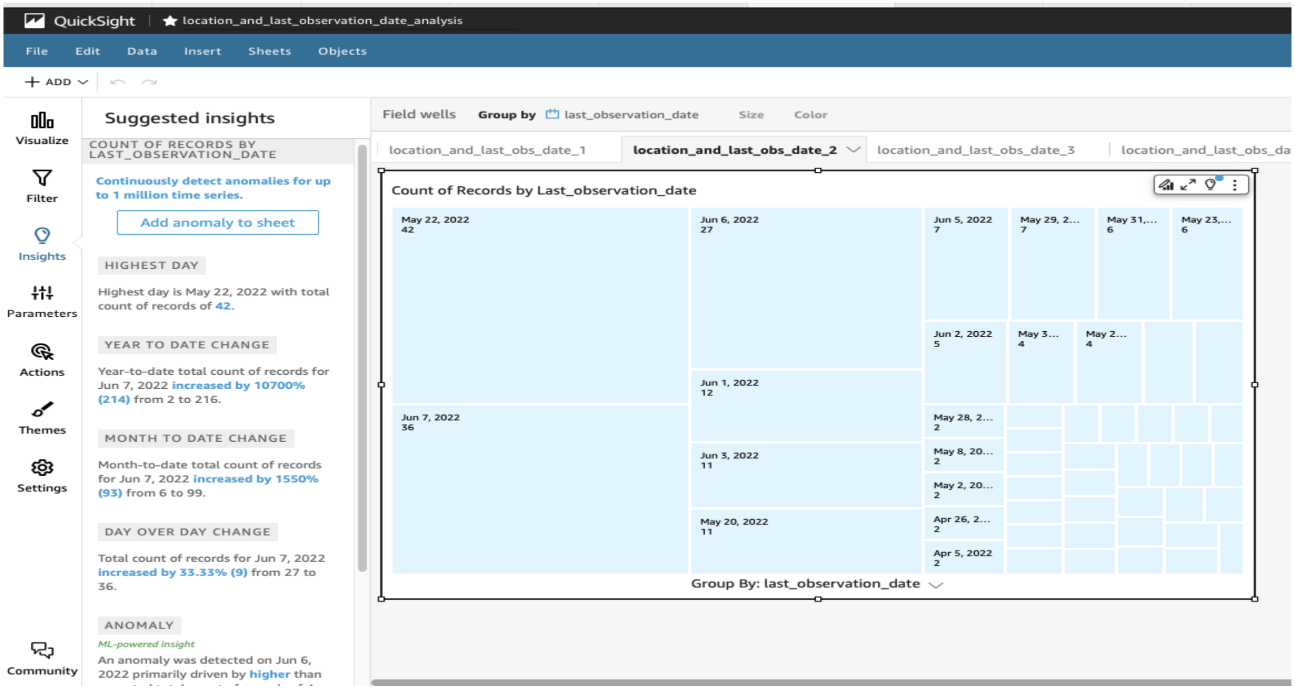


Image 9: [count of records by last observation date]

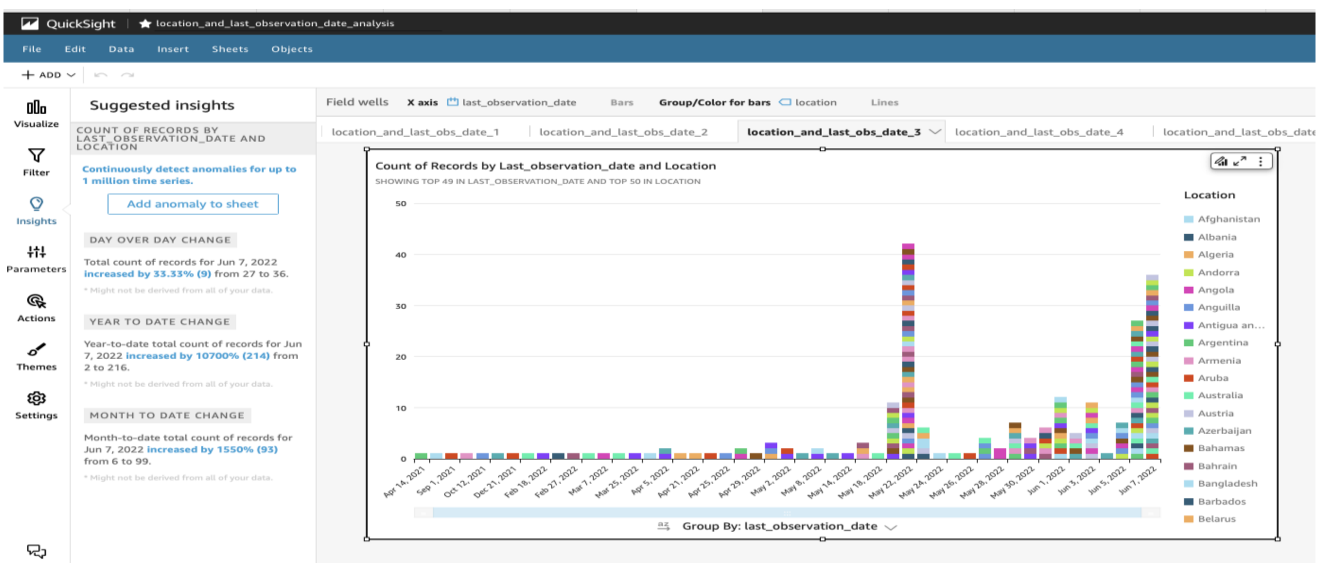


Image 10: [count of records by last observation date]

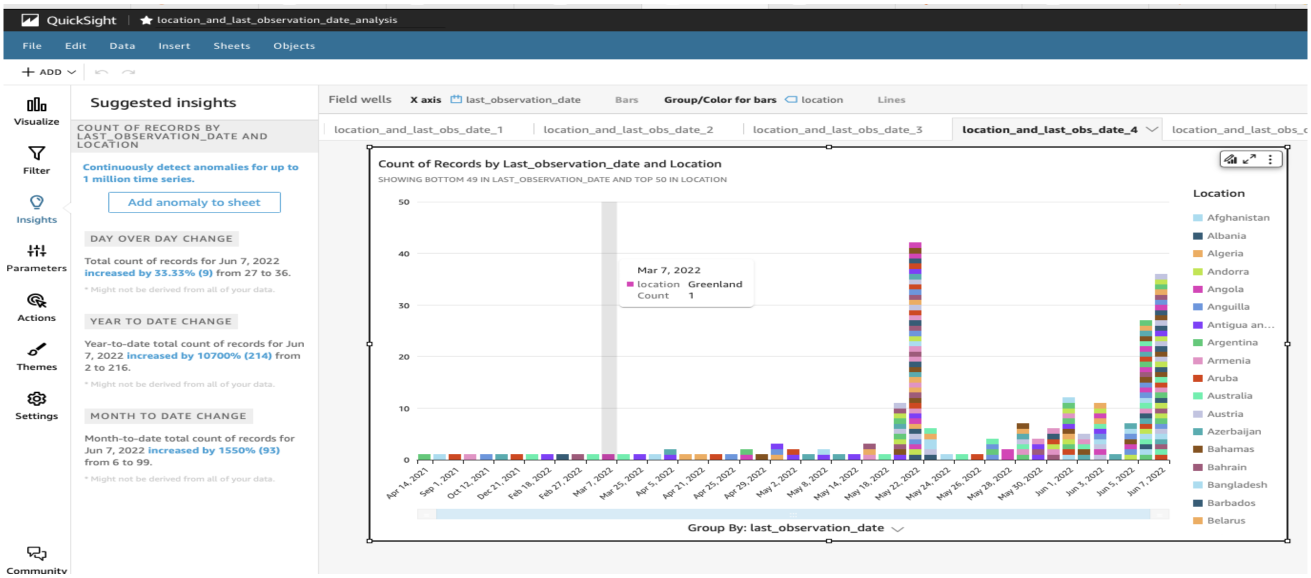


Image 11: [histogram representing country wise total vaccination count]

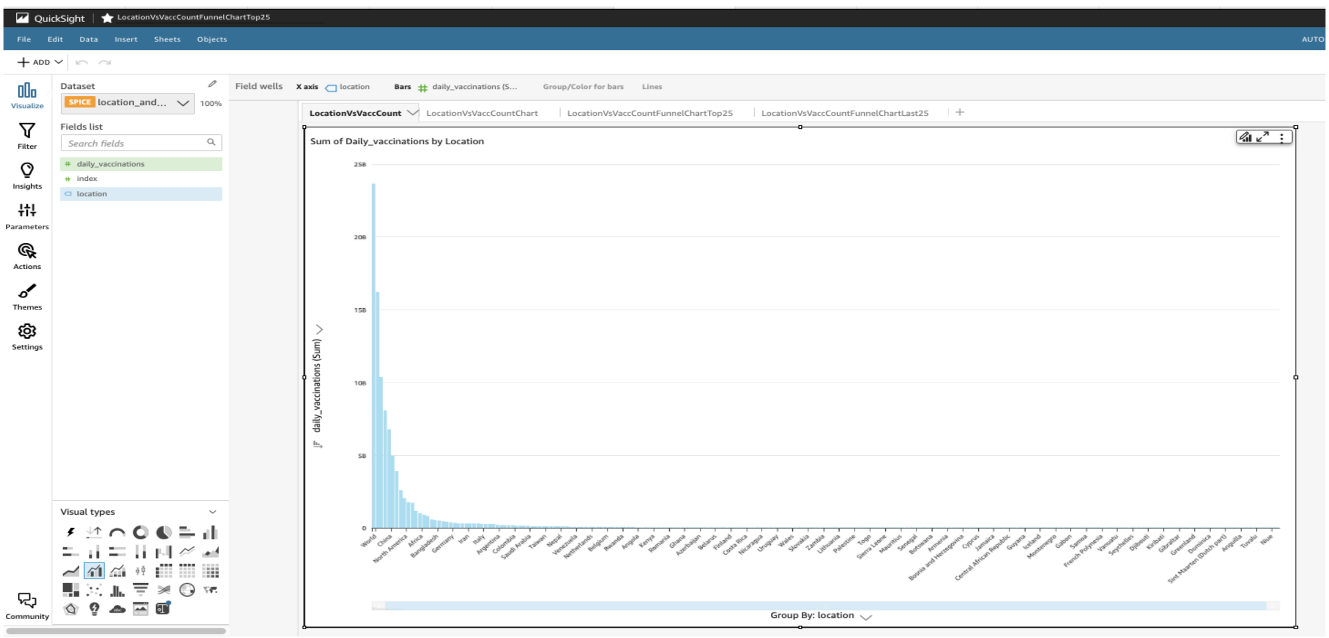
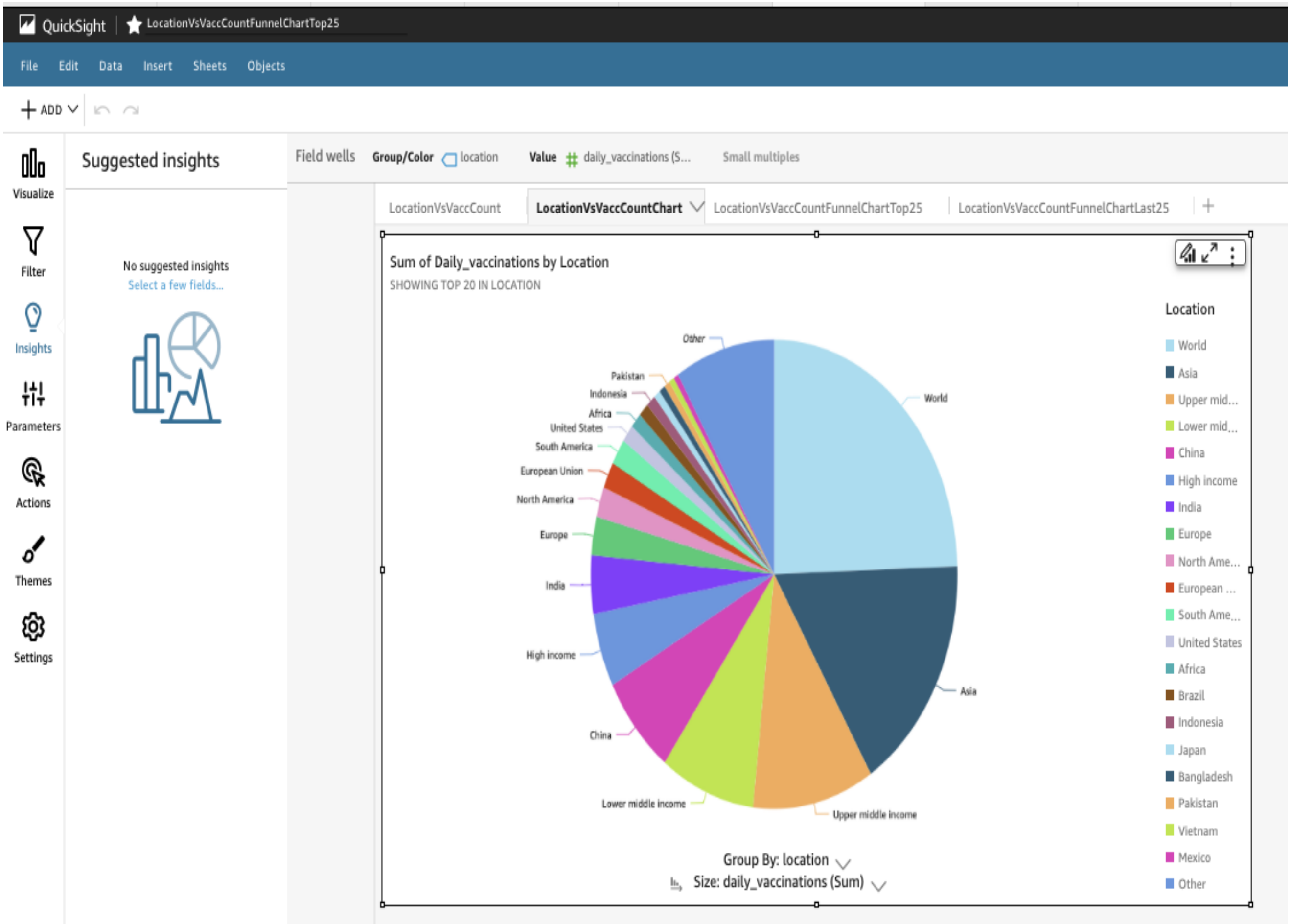


Image 12: [pie chart representing country wise total vaccination count]



<u>Image 13: [waterfall chart representing top countries with highest total vaccination count]</u>

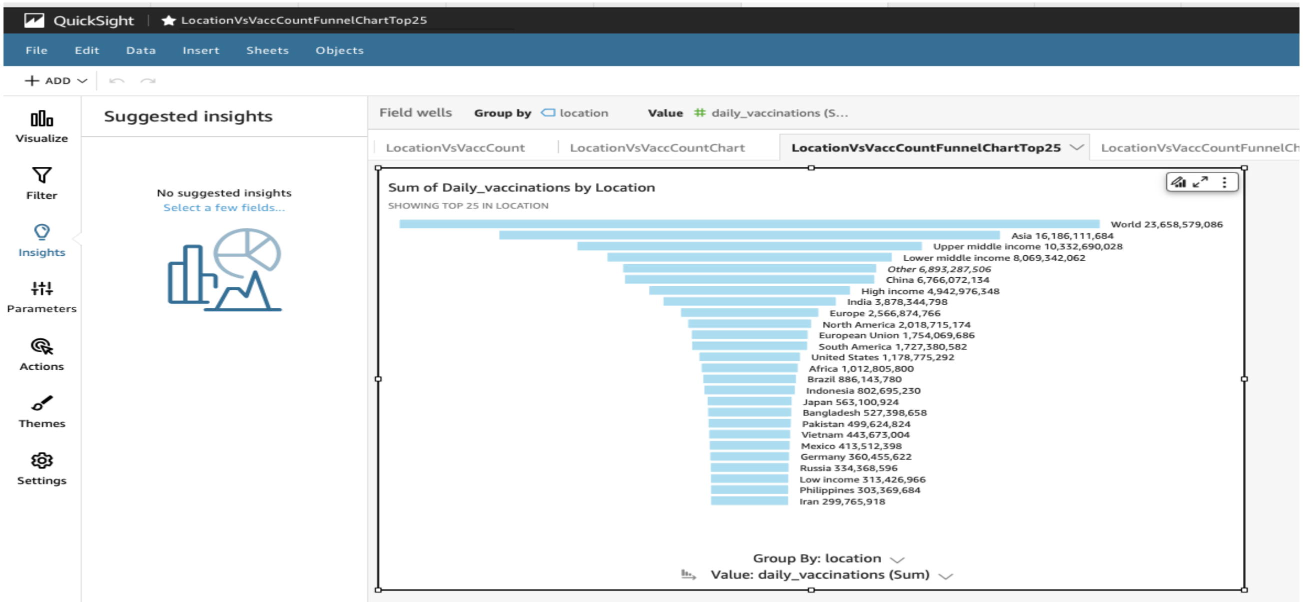
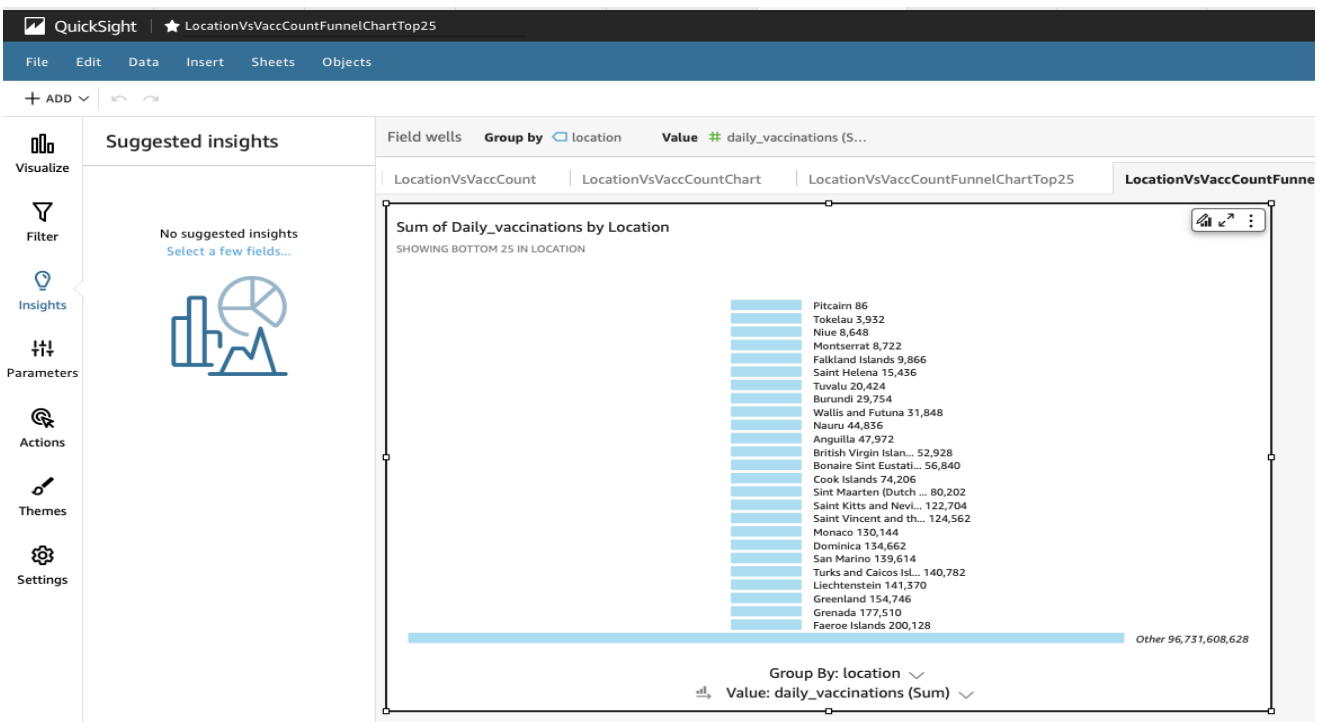


Image 14: [waterfall chart representing top countries with lowest total vaccination count]



Conclusion:

In conclusion, our comprehensive analysis of COVID vaccination across diverse countries in this project has unveiled critical insights into the global response to the pandemic. Through meticulous examination of vaccination rates and distribution strategies, our analysis provides a nuanced understanding of the statistics about vaccination rate across various countries, vaccination brands used in different regions etc.

Key findings indicate substantial variations in vaccination coverage among nations, prompting the need for tailored strategies to address unique challenges. Notably, some regions showcased remarkable success in achieving high vaccination rates, underscoring the importance of effective communication, transparent information dissemination while there is a need to awaken certain countries on how to respond in such pandemic outbreaks.

Our project contributes valuable insights to the global discourse on COVID vaccination, emphasizing the need for collaborative efforts, knowledge exchange, and innovative solutions. The identified patterns and challenges serve as a foundation for informed decision-making by policymakers, health practitioners, and researchers worldwide.

Looking forward, our insights suggest potential areas for further research, including the long-term effectiveness of vaccination campaigns, the impact of emerging variants, and the equity in vaccine access. By addressing these critical gaps in knowledge, future research endeavors can build upon our findings and contribute to the ongoing global effort to mitigate the impact of the COVID-19 pandemic.

In essence, our project illuminates the intricate tapestry of COVID vaccination on a global scale, emphasizing the necessity of adaptive strategies and collective action. As we navigate the evolving landscape of this unprecedented health crisis, our commitment to understanding and addressing the complexities of vaccination across borders remains paramount for a resilient and united global response.