Final Project
Big Data Management Concepts
Covid-19 Vaccine Distribution
Anuj Mahajan (anujmaha) - Fall 2023

Section 1 – Introduction

For this project I have used the data published by US Government for Covid – 19 Vaccine distribution for the last 1 year [1]. The data is available at covid - 19 vaccine distribution data. This data consist of different attributes like total number of vaccines distributed, number of vaccines administered, Number of vaccines distributed for people having age > 18, age > 50 and age > 65. Through this data, we can analyze what are the current trends going on in covid-19 vaccine distribution in each US State and hence can analyze the spread of covid-19 as well.

The following concepts of the big data from the course are used throughout the project.

- Data Ingestion and Storage
- Virtualization
- Cloud Computing
- Data Lifecycle & Pipelines
- Data Processing and Analytics
- AI Fairness

During this project, I downloaded the data from CDC website and kept it in the MongoDB Atlas which acted as a storage service for the project. Then I read all the data through python inbuilt libraries. I incorporated Jetstream VM to run my python code on the data. After that, I set up the data pipeline where I

transformed the data and performed the in-depth analysis and at the end published the data and code on cloud (GCP) and GitHub.

Section 2 – Background

The COVID-19 vaccine distribution dataset was chosen due to its critical significance in understanding the dynamics and effectiveness of the immunization program that was implemented throughout the United States during the historic worldwide pandemic. The use of vaccines became apparent as a vital tactic to stop the COVID-19 virus from spreading and protect public health as the world struggled with its problems.

The information under review offers a detailed look at the weekly distribution of COVID-19 vaccinations, revealing details about the precise distribution among various demographic groups and vaccine producers in addition to general distribution trends. The information enables a detailed examination of the immunization campaigns by exploring the age-specific distributions (above 12, above 18, and above 65) as well as the donations of significant pharmaceutical corporations like Pfizer, Johnson & Johnson and Moderna.

The potential for this investigation to reveal trends, contradictions, and effective vaccination distribution tactics makes it very fascinating. Comprehending the distribution of vaccinations throughout distinct age cohorts and the functions fulfilled by diverse vaccine producers is crucial for assessing the efficacy of public health initiatives, pinpointing opportunities for enhancement, and providing guidance for further immunization drives. All things considered, the dataset is a useful resource for decision-makers, and the general public to understand the complex terrain of COVID-19 vaccine distribution in the US.

Section 3 – Methodology

Below are the steps involved in the process of addressing the problem at hand

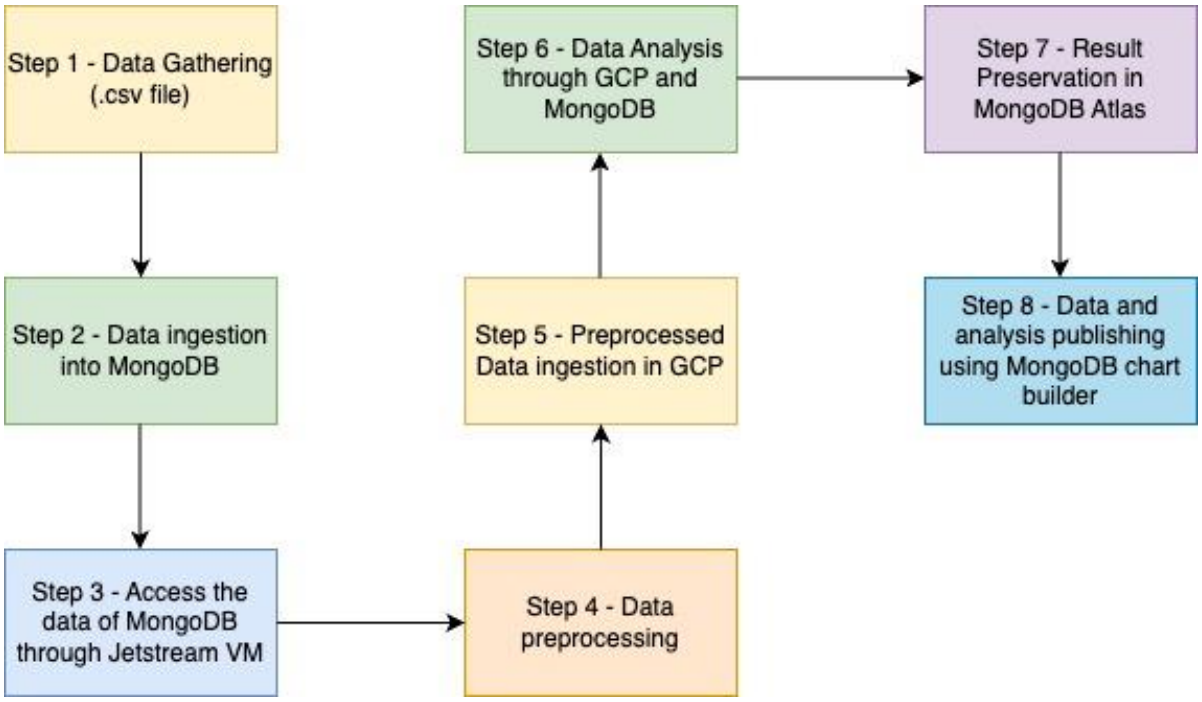


Figure 1 : Methodology

Step 1 - Data Gathering

- In this step, the required data has been collected from the CDC website. The data gathered is then stored into MongoDB atlas for future use.

Step 2 - Data Ingestion into MongoDB

- To store the data into MongoDB I have created the cluster with free tier in MongoDB. I have created the cluster in us-east-1 region.
- Under this cluster, I then created the database ‘Covid’ and under that I created a collection ‘Vaccine’.
- Through MongoDB Compass, I imported the .csv file downloaded in step 1 of Figure 1 to ingest all the data into NoSQL MongoDB collection.

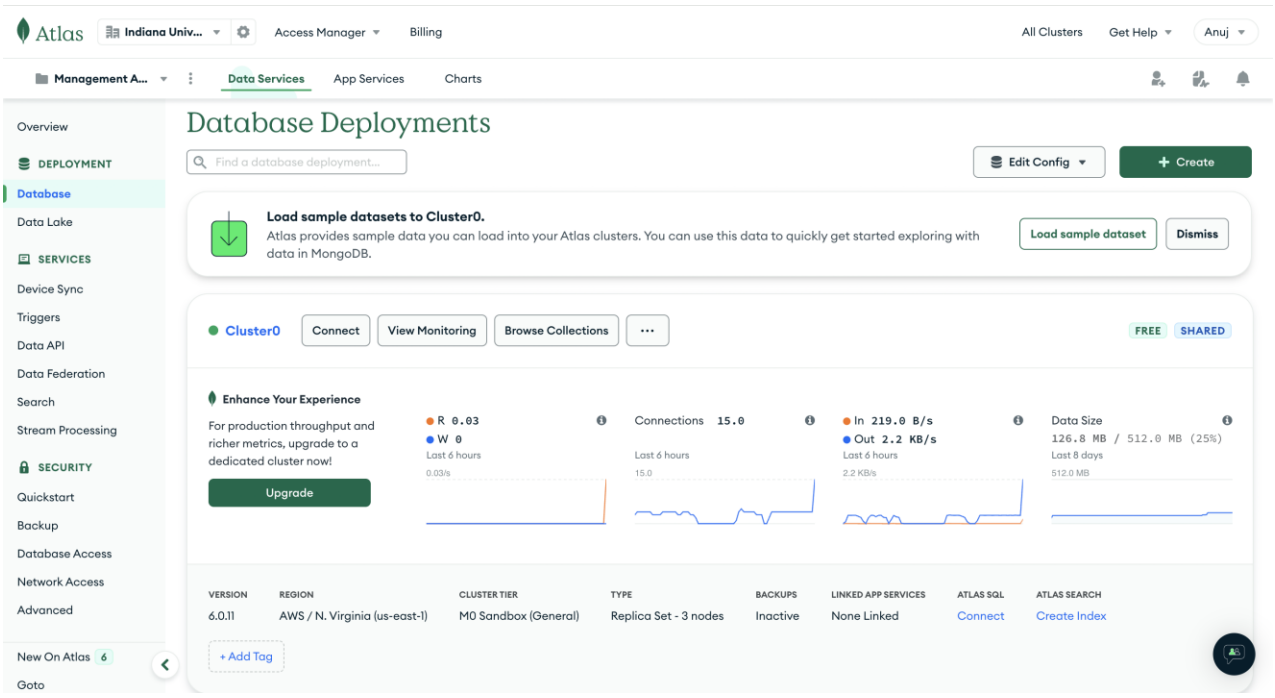


Figure 2 : MongoDB Cluster

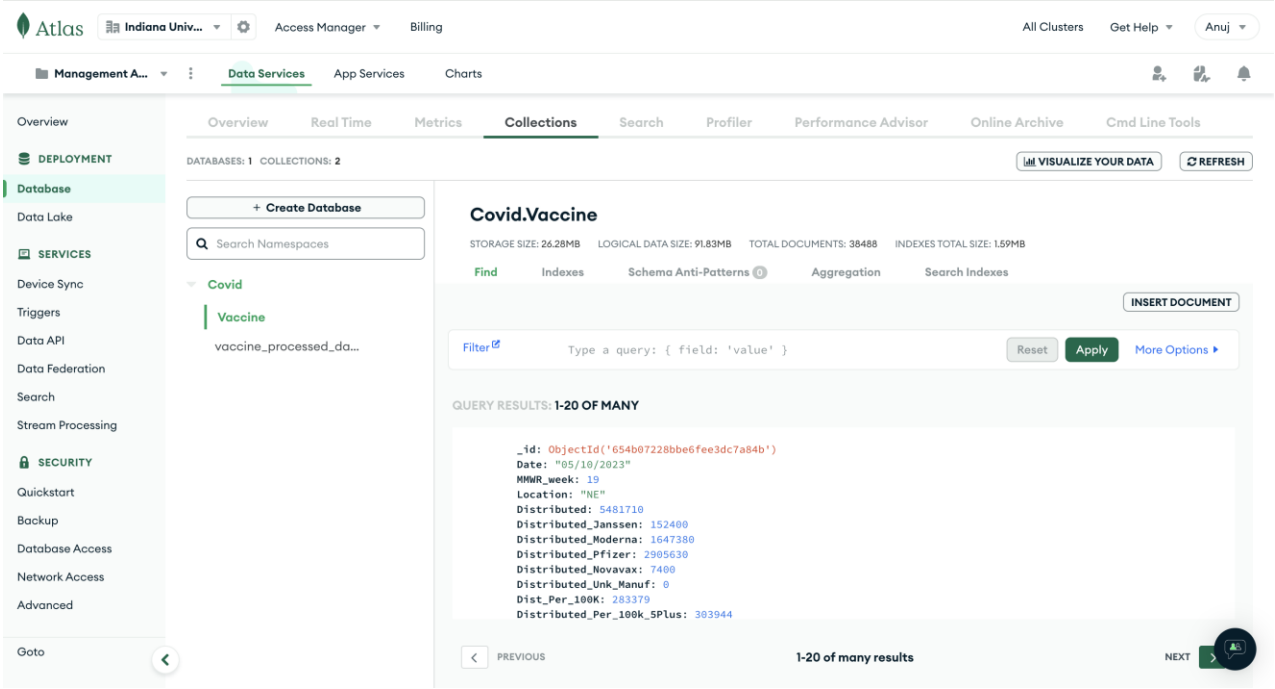


Figure 3 : MongoDB collections

Step 3 – Access the data on Jetstream VM

- To make use of virtualization, I incorporated Jetstream VM as we used throughout the course.
- Here, inside the VM I created the docker container and ran my jupyter notebook inside it the same way we did it while completing the pyspark assignment.
- I ran the docker-compose up command to start the docker container and create my jupyter notebook inside it where I fetched the data, processed it and stored it on the MongoDB [5].

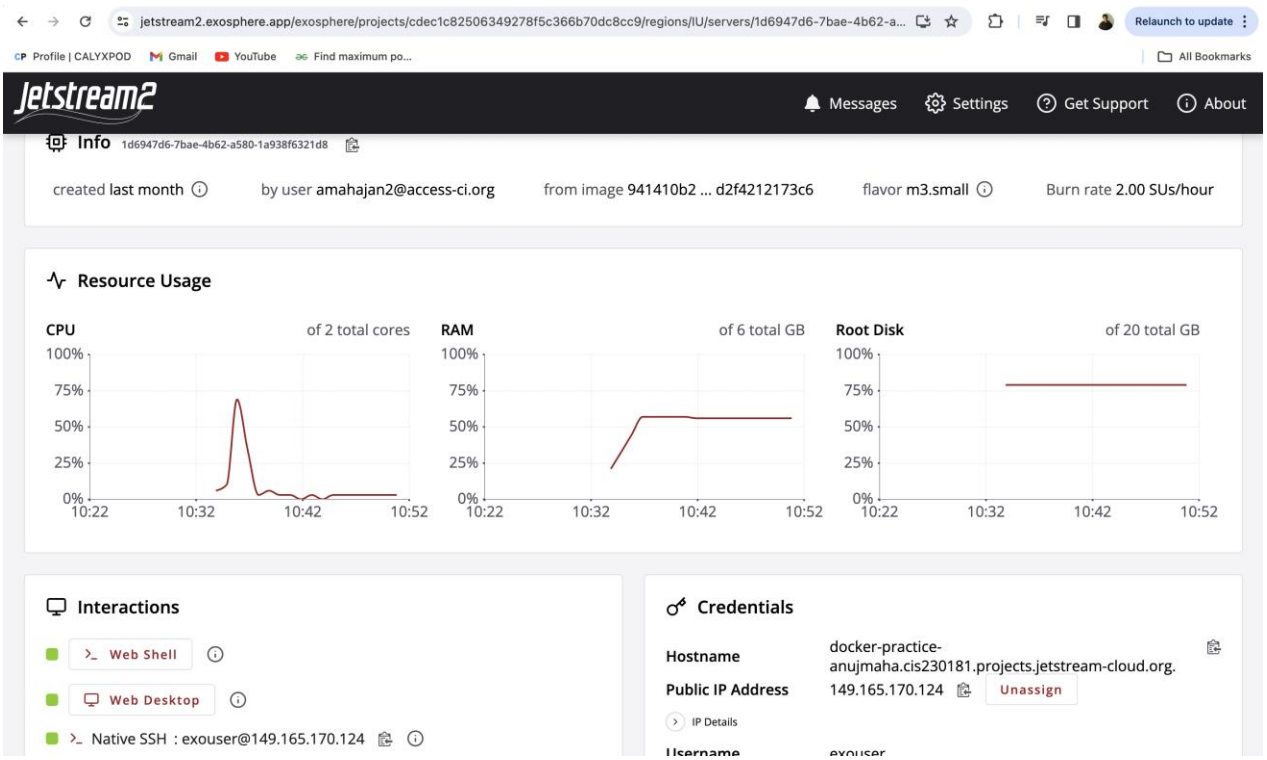


Figure 4 : Jetstream VM Instance

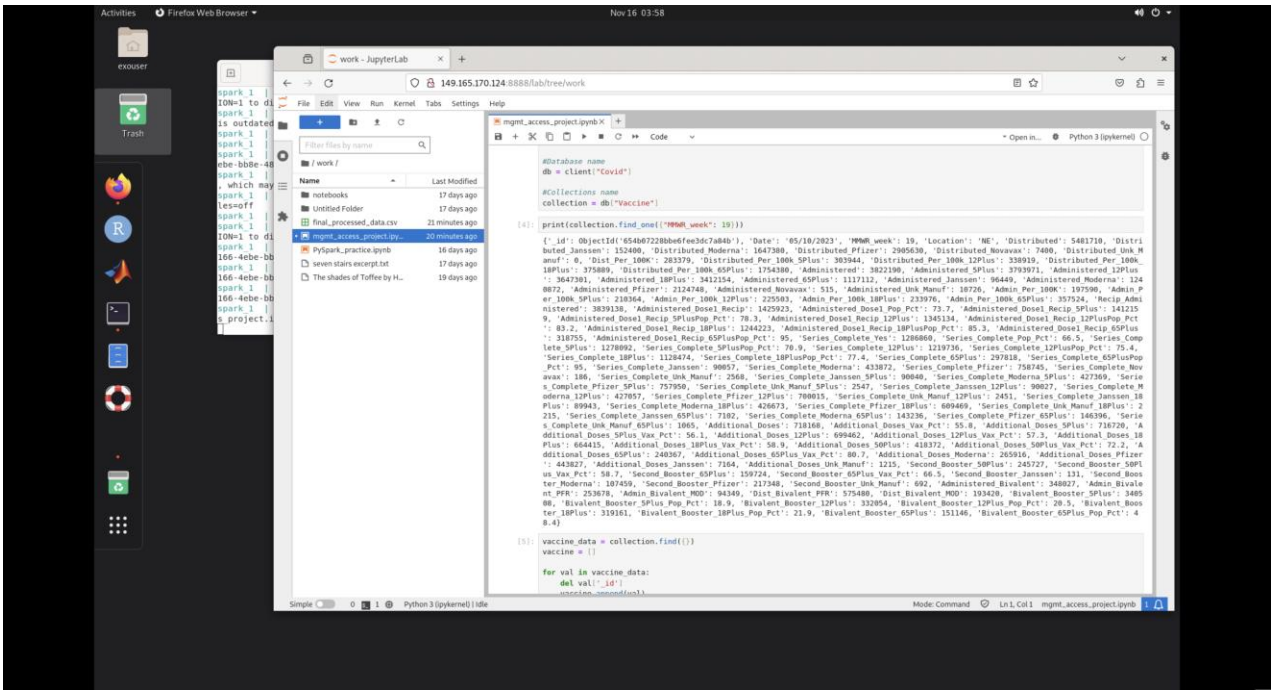


Figure 5 : Jupyter Notebook running inside Jetstream VM on Docker

- To fetch the data from MongoDB into python, I have used the pymongo library offered by python [2]. I connected my MongoDB cluster and connection using the URL provided my MongoDB Compass and fetched the data

```
In [28]:
!pip install pymongo

Requirement already satisfied: pymongo in /Users/anujmahajan/opt/anaconda3/lib/python3.9/site-packages (4.6.0)
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in /Users/anujmahajan/opt/anaconda3/lib/python3.9/site-packages (from pymongo) (2.4.2)
```

```
In [29]:
import matplotlib.pyplot as plt
import pandas as pd
import plotly.express as px
import seaborn as sns
```

```
In [30]:
from pymongo import MongoClient

client = MongoClient("mongodb+srv://anujmaha:anujmaha@cluster0.vsqmzh9.mongodb.net/")

#Database name
db = client["Covid"]

#Collections name
collection = db["Vaccine"]
```

Figure 6 : MongoDB Connection Python code

In [32]:

```
vaccine_data = collection.find({})
vaccine = []

for val in vaccine_data:
    del val['_id']
    vaccine.append(val)

vaccine_data = pd.DataFrame(vaccine)
vaccine_data.head(10)
```

Out[32]:

	Date	MMWR_week	Location	Distributed	Distributed_Janssen	Distributed_Moderna	Distributed_Pfizer	Distributed_Novavax	Distributed_Unk_Manuf
0	05/10/2023	19	TX	80813315	2667600	25478340	43584365	119500.0	0
1	05/10/2023	19	WA	25606455	777800	7783800	12807915	41600.0	0
2	05/10/2023	19	SD	2713025	94000	882400	1326385	5600.0	0
3	05/10/2023	19	VA2	9888200	626900	4250780	3780780	12100.0	0
4	05/10/2023	19	MO	15654225	441000	5140460	7967645	22500.0	0
5	05/10/2023	19	CA	121107865	3785200	37596780	63095065	175900.0	0
6	05/10/2023	19	MT	2714785	106500	952100	1267535	7400.0	0
7	05/10/2023	19	MD	22359840	616400	6422360	11814000	40800.0	0
8	05/10/2023	19	HI	4677910	124900	1504020	2369320	4900.0	0
9	05/03/2023	18	NE	5470070	152400	1647380	2905630	7400.0	0

10 rows × 109 columns

Figure 7 : Displaying data fetched from MongoDB

Step 4 – Data Preprocessing

- Data preprocessing is one of the most important part in any data pipeline and data lifecycle.
- This process involves different subparts, like filtering the data, cleaning the data, removing unnecessary data, etc.
- Figure 8 will give you the detail insight about what all steps I took to create the final copy of data

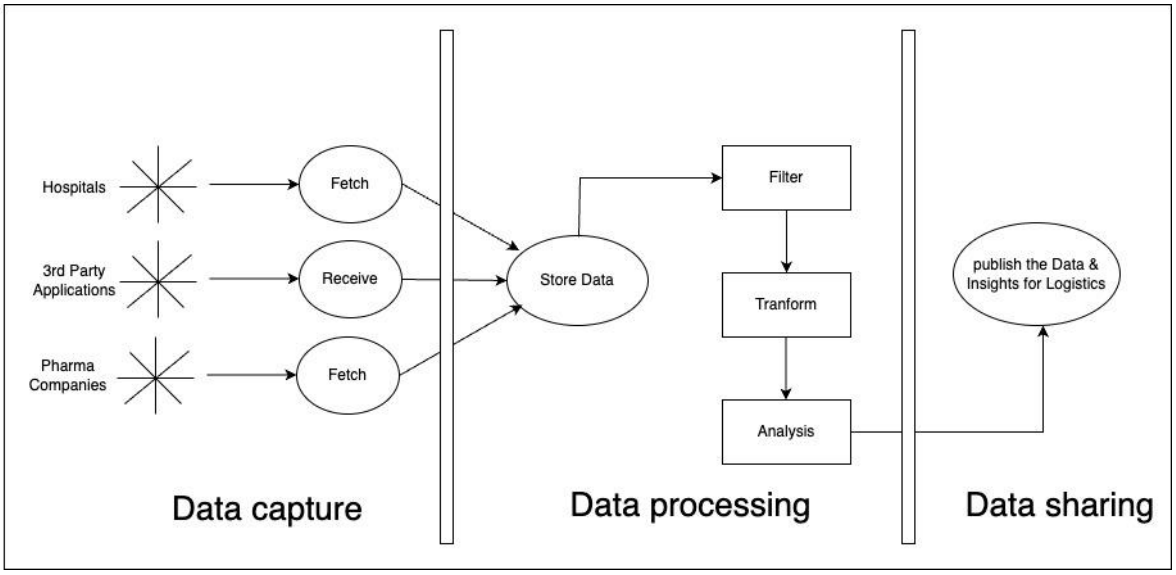


Figure 8 : Data Lifecycle and Pipeline

- Data Cleaning
  - The first task I performed in the data processing was to clean the data.
  - In the overall data there were lot of columns which have almost all the values as Null.
  - I set the threshold of 50% null values and dropped all the columns having null values more than this threshold.

In [33]:

```
null_counts = vaccine_data.isnull().sum()

# For 0 values
zero_counts = (vaccine_data == 0).sum()

# Printing the counts
print("Null Counts:")
# sortedList = sorted(null_counts, reverse = True)
# print(null_counts[::-1][30:60])
sorted_null_counts = null_counts.sort_values(ascending=False)

print(sorted_null_counts[:60])

# print("\nZero Counts:")
# print(zero_counts)
```

```
Null Counts:
Second_Booster                          38385
Bivalent_Booster_5Plus_Pop_Pct          36568
Bivalent_Booster_5Plus                  36568
Bivalent_Booster_65Plus_Pop_Pct         36504
Bivalent_Booster_65Plus                 36504
Bivalent_Booster_18Plus_Pop_Pct         36504
Bivalent_Booster_18Plus                 36504
Bivalent_Booster_12Plus_Pop_Pct         36504
Bivalent_Booster_12Plus                 36504
Admin_Bivalent_MOD                      36312
Admin_Bivalent_PFR                      36312
Dist_Bivalent_MOD                       36312
Dist_Bivalent_PFR                       36312
Administered_Bivalent                   36248
Series_Complete_Novavax                 35808
Administered_Novavax                    35807
Distributed_Novavax                     35800
Additional_Doses_5Plus                  35544
Additional_Doses_5Plus_Vax_Pct          35544
Second_Booster_Unk_Manuf                31907
Second_Booster_Janssen                  31905
Second_Booster_50Plus                   31896
Second_Booster_Pfizer                   31896
Second_Booster_Moderna                  31896
Second_Booster_65Plus_Vax_Pct           31896
Second_Booster_65Plus                   31896
Second_Booster_50Plus_Vax_Pct           31896
Additional_Doses_12Plus                 26456
Additional_Doses_12Plus_Vax_Pct         26456
Series_Complete_Unk_Manuf_5Plus         21020
Series_Complete_Pfizer_5Plus            21016
Series_Complete_Janssen_5Plus           21016
Series_Complete_Moderna_5Plus           21016
```

Figure 9 : Columns with null values count

```
In [34]:
threshold = 16000

# Filter columns based on the threshold
columns_to_keep = null_counts[null_counts <= threshold].index

# Create a new DataFrame with selected columns
filtered_vaccine_data = vaccine_data[columns_to_keep]

# Print the filtered DataFrame
print(filtered_vaccine_data)
print(filtered_vaccine_data.shape[1])
```

```
                Date  MMWR_week Location  Distributed  Distributed_Janssen  \
0         05/10/2023         19       TX     80813315              2667600
1         05/10/2023         19       WA     25606455               777800
2         05/10/2023         19       SD      2713025                94000
3         05/10/2023         19      VA2      9888200               626900
4         05/10/2023         19       MO     15654225               441000
...              ...        ...      ...          ...                  ...
38483     12/14/2020         51       MD         1950                    0
38484     12/14/2020         51       AR         3900                    0
38485     12/14/2020         51       WI         9750                    0
38486     12/14/2020         51       IA         1950                    0
38487     12/14/2020         51       AL         6825                    0

       Distributed_Moderna  Distributed_Pfizer  Distributed_Unk_Manuf  \
0                 25478340            43584365                      0
1                  7783800            12807915                      0
2                   882400             1326385                      0
3                  4250780             3780780                      0
4                  5140460             7967645                      0
```

Figure 10 : Dataframe after removing null values columns

- Data Transformation
  - After the data cleaning I performed the data transformation.
  - The data I had at hand had all the states/territories in USA and for each state I had multiple entries for all 52 weeks of year. So, I grouped all the data based on the states and weeks.

```
In [37]:
grouped_data = filtered_vaccine_data.groupby(['Location', 'MMWR_week'], as_index=False)
result = grouped_data.sum()
```

Figure 11 : Grouping based on Week and Location

  - After this, I created a new feature for my dataset to analyze what is the percentage of population impacted is above 65 age and named it as ‘Age_65Plus_percentage’. Along with that I created one new feature to detect if for that specific week and for that specific location, does this impacted population of age > 65 is greater than the mean for that specific location and named this feature as ‘majority_old_people_impacted’.

Figure 12 : Creating new features

- To check the quality, bias and fairness of this new grouped data I incorporated the use of aif360 [3].
- I followed the same process as we did in the assignment of AI Fairness. I made use of BinaryLabelDatasetMetric, Reweighing and I used the previously created new feature ‘Age_65Plus_percentage’ as the protected attribute.
- I divided the dataset into training and testing data with 70-30 distribution.
- At start, I received the fairness score of –0.19, which means the privileged group was receiving 19% more favorable results.
- To mitigate this bias, I used reweighing functionality and transformed the data so that it should not contain any bias.

```
In [40]:
from aif360.datasets import GermanDataset
from aif360.metrics import BinaryLabelDatasetMetric
from aif360.algorithms.preprocessing import Reweighing
from aif360.datasets import StandardDataset

In [41]:
processed_dataframe_bkp = processed_dataframe.copy(deep=True)
processed_dataframe.drop(columns=['Location'], inplace=True)

In [42]:
dataset = StandardDataset(processed_dataframe,
                          label_name='majority_old_people_impacted',
                          favorable_classes=[1],
                          protected_attribute_names=['Age_65Plus_percentage'],
                          privileged_classes=[lambda x: x < 275])

WARNING:root:Missing Data: 1 rows removed from StandardDataset.

In [43]:
dataset_train, dataset_test = dataset.split([0.7], shuffle=True)

In [44]:
privileged_groups = [{'Age_65Plus_percentage': 1}]
unprivileged_groups = [{'Age_65Plus_percentage': 0}]

In [45]:
metric_orig_train = BinaryLabelDatasetMetric(dataset_train,
                                             unprivileged_groups=unprivileged_groups,
                                             privileged_groups=privileged_groups)

print("Difference in mean outcomes between unprivileged and privileged groups = %f" % metric_orig_train.mean_dif

Difference in mean outcomes between unprivileged and privileged groups = -0.199863

In [46]:
RW = Reweighing(unprivileged_groups=unprivileged_groups,
                privileged_groups=privileged_groups)
dataset_train = RW.fit_transform(dataset_train)

In [47]:
ric_orig_train = BinaryLabelDatasetMetric(dataset_train,
                                          unprivileged_groups=unprivileged_groups,
                                          privileged_groups=privileged_groups)

nt("Difference in mean outcomes between unprivileged and privileged groups = %f" % metric_orig_train.mean_differe

Difference in mean outcomes between unprivileged and privileged groups = 0.000000
```

Figure 13 : Mitigating Bias through aif360

Step 5 – Preprocessed data ingestion into GCP

- After completing the preprocessing and transformation in step 4, I ingested all this data into buckets on GCP known as GCS to make it available publicly.
- In bigQuery, I then created one database and table [4].
- Through this bucket I read the data into this bigQuery table.
- I ran the queries using bigQuery and generated some insights for logistics.

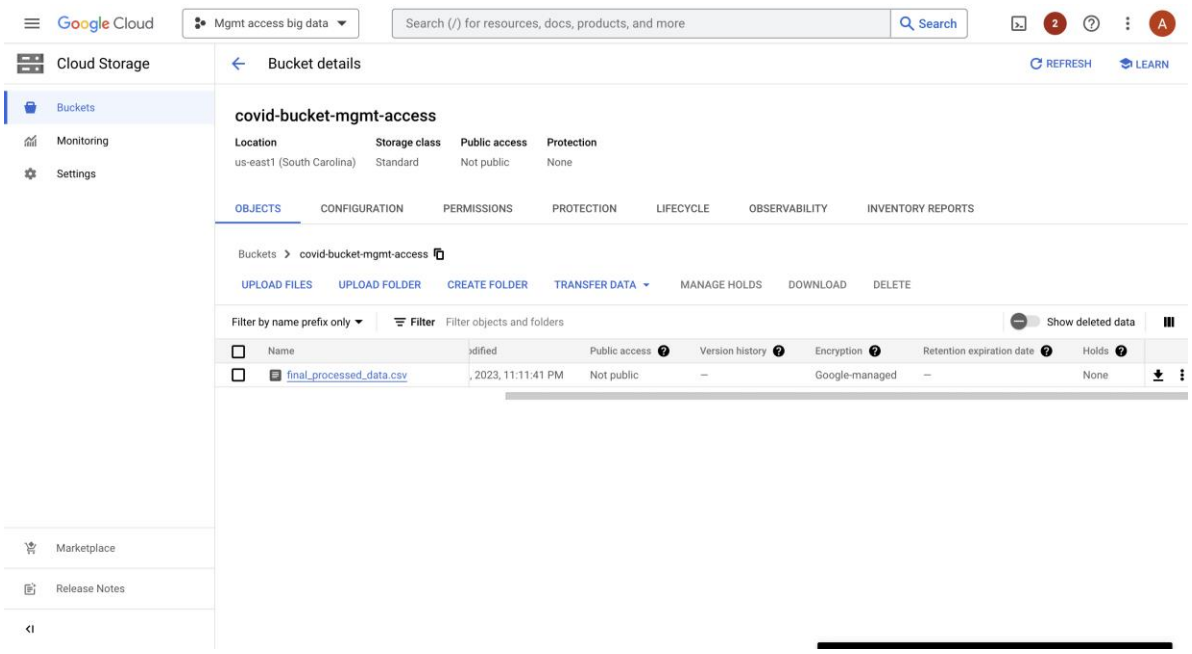


Figure 14 : GCP bucket with processed data

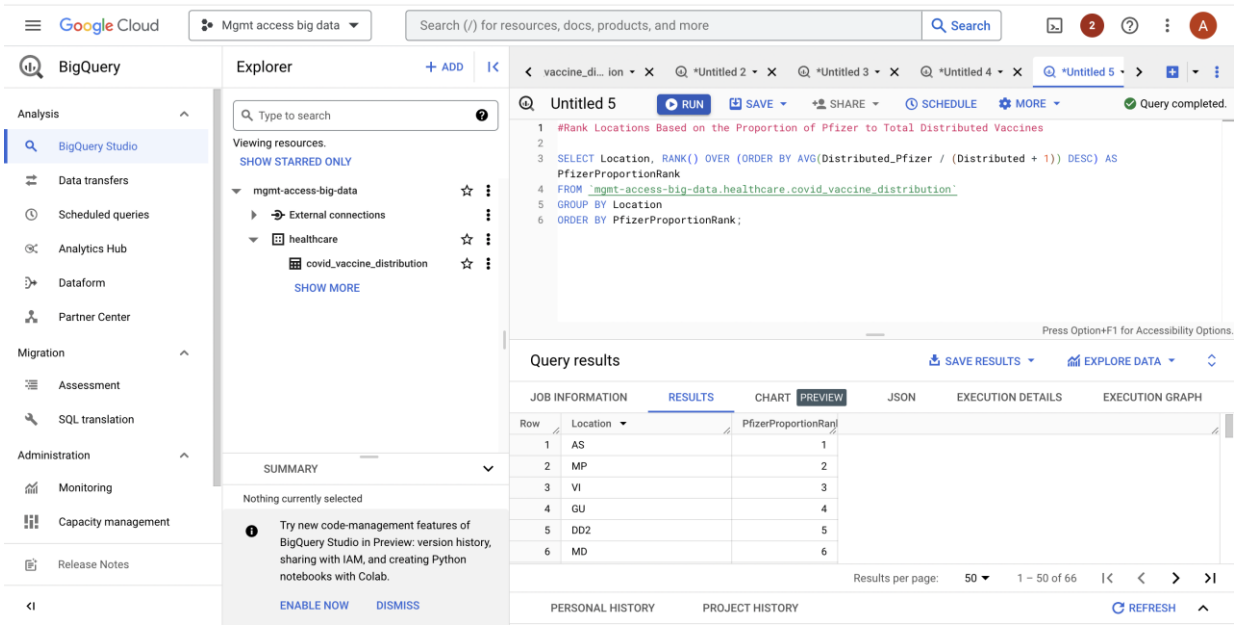


Figure 15 : GCP BigQuery with diagnostics

Step 6, 7, 8 will be discussed in results as they are analysis.

Technologies used

Figure 16 : Technology Stack

Section 4 – Result

- As a result of all the steps I had taken till now, I got the preprocessed data and then I ran the analysis on that data to gain some insights about the vaccine distribution
- Below are the questions I tried to answer through the analysis

MongoDB Chart Builder Analysis [6]

Analysis 1

Demographic vaccine distribution analysis

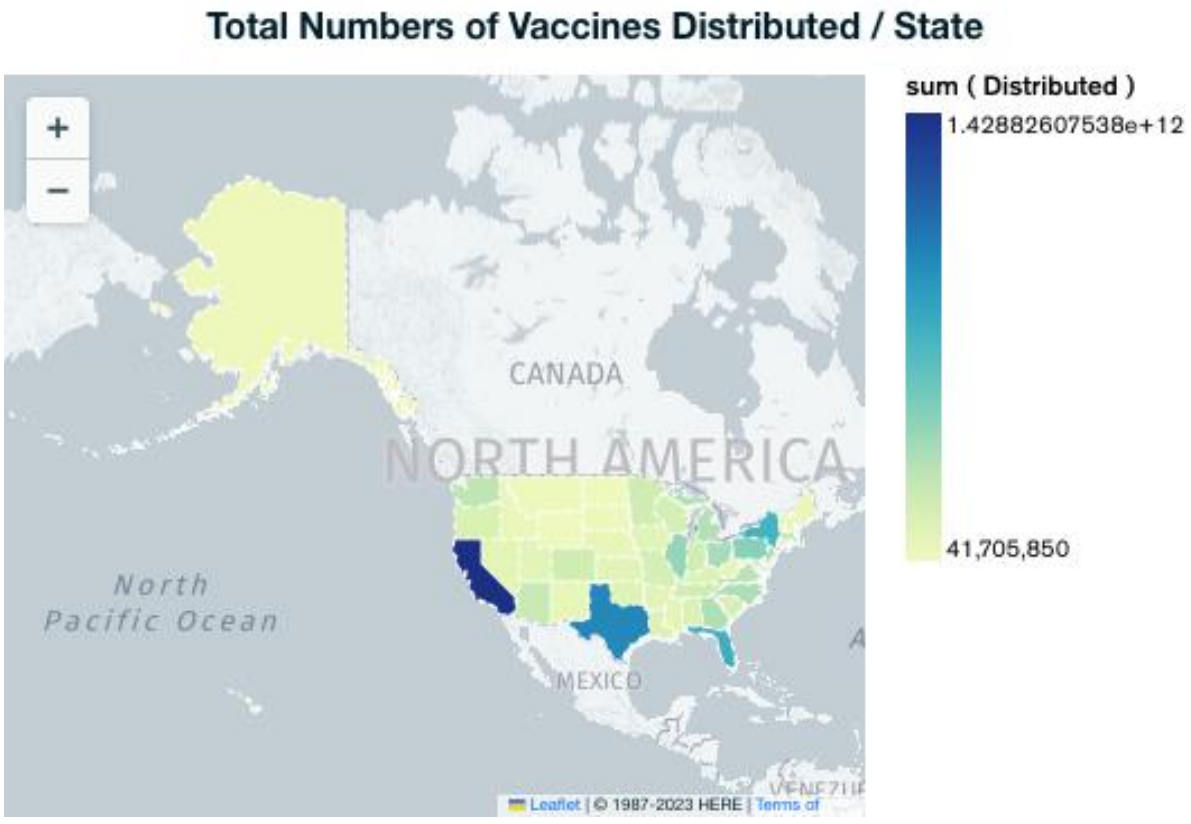


Figure 17 : Demographic Vaccine Distribution

- Figure 17 shows the total distribution throughout the year in different US states.
- This helps us to analyze which states has high vaccine requirements and which states has the relatively low vaccine requirements

Analysis 2

What number of doses are administered each week ?

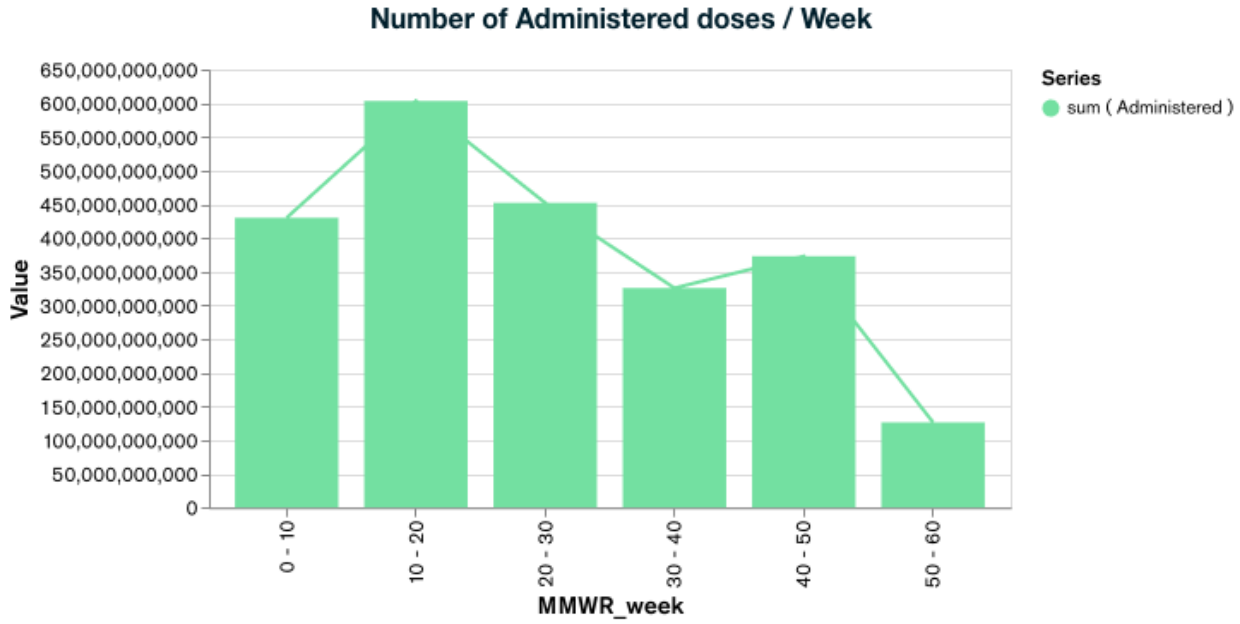


Figure 18 : Weekly Administered Doses

- Figure 18 shows the number of doses administered by US government each week.
- This tells us which weeks had the highest number of administered doses.
- From the figure 18 we can analyze that the week 10-20 had the highest dose administration which implies that there are highest number of doses distributed in same time frame. On the other hand, week 50-60 had the lowest dose administration.

Analysis 3

What is the additional dose distribution for age > 18, age > 50 and age > 65 ?

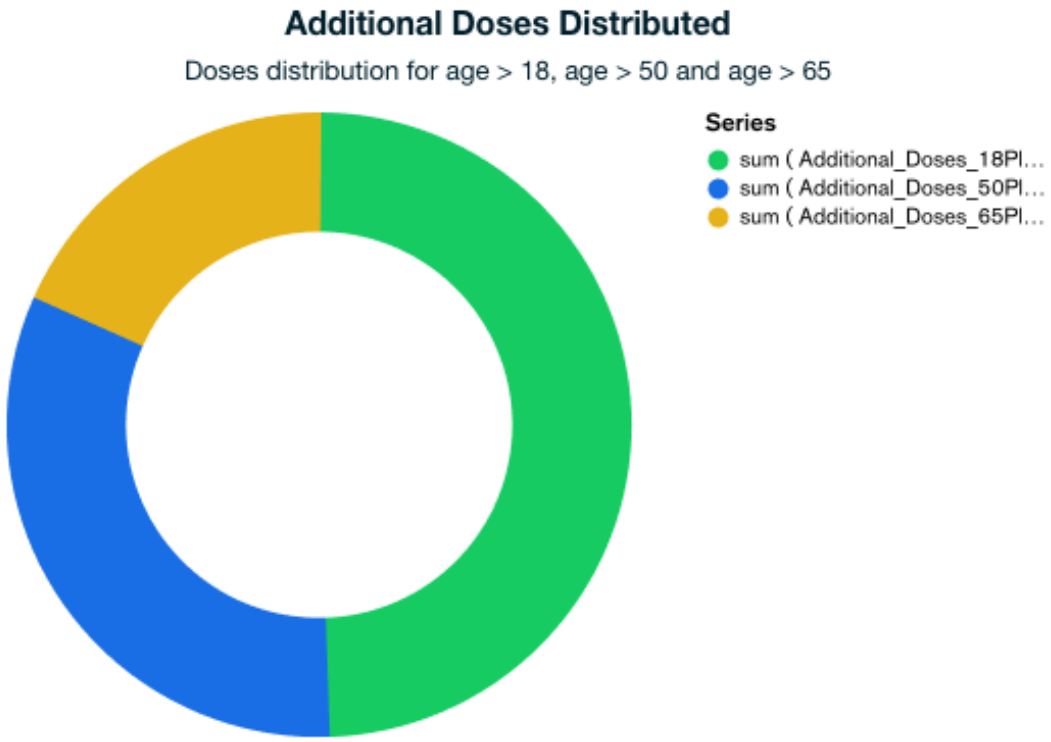


Figure 19 : Additional Dose Distribution for different age groups

- Figure 19 shows the number of additional doses distributed for different age group.
- We can infer from the pie chart that almost half of the additional doses were distributed for the people above age 18 and below 50 followed by the people between age group 50 to 65 which accounts for 32 %.
- This implies that the old population (age > 65), received exceedingly small number of additional doses in last year as compared to the adults.

Analysis 4

What is the weekly total vaccine distribution and the contribution of different companies in that distribution ?

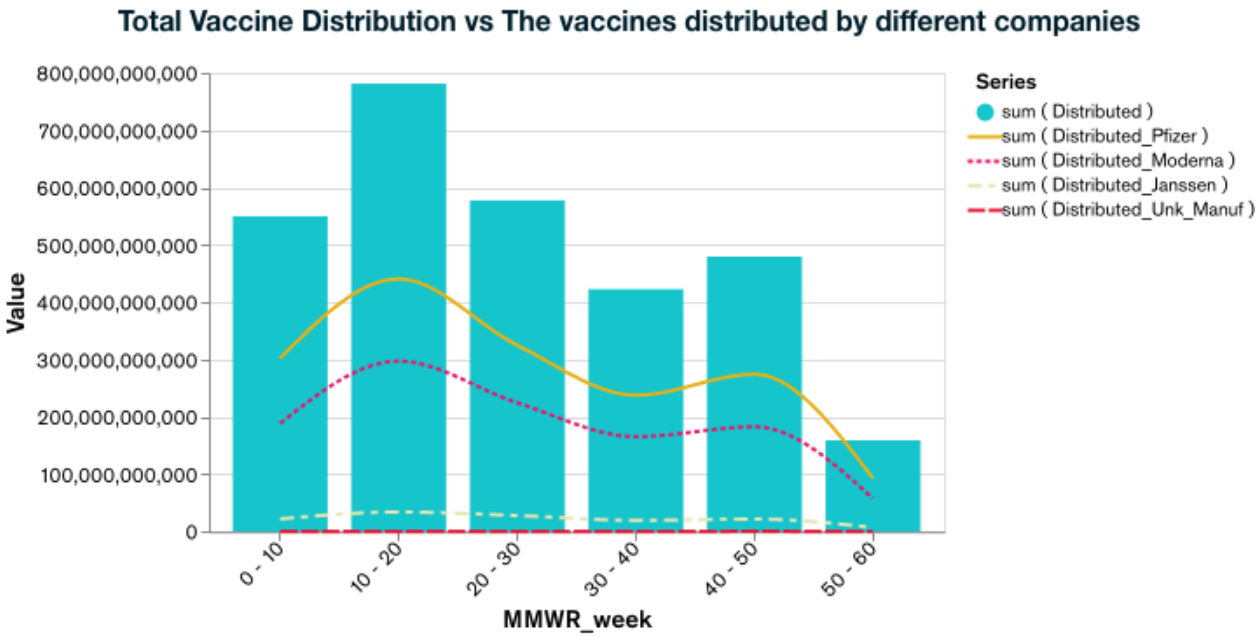


Figure 20 : Distribution for different manufacturers

- Figure 20 shows the weekly total distribution of vaccines. Along with that, it shows which manufacturer distributed how many vaccines out of those total vaccines.
- We can clearly infer from the figure 20 that in all the weeks Pfizer is the leading vaccine distributor in USA followed by the Moderna.
- Johnson & Johnson and Unknown Manufacturers contributed the least in vaccine distribution throughout the year.

Analysis 5

What is the statewise total vaccine distribution vs statewise total vaccines administered?

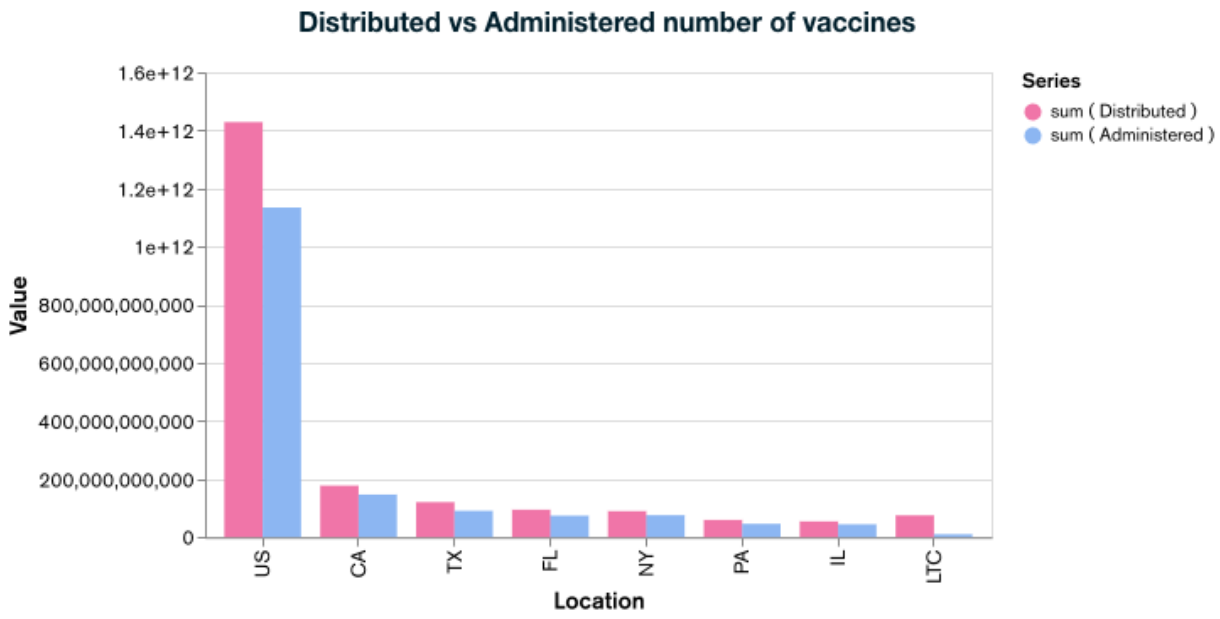


Figure 21 : Vaccines Distributed vs Vaccines Administered

- Figure 21 shows the statewise vaccines distributed and at the same time the vaccines administered.
- We can see that the administered vaccines are less than the total distributed vaccines.
- This shows that, US Government must bridge a gap to administer all the vaccines they had distributed.

Jupyter Notebook Analysis

Analysis 1

Which 10 states have the highest vaccine distribution count throughout the year?

Top 10 states - Highest distribution

In [50]:
```
statewise_distribution = processed_dataframe.groupby('Location')['Distributed'].sum().reset_index()

statewise_sorted_distribution = statewise_distribution.sort_values(by='Distributed', ascending=False)

statewise_sorted_distribution = statewise_sorted_distribution[1:11]

fig = px.bar(statewise_sorted_distribution, x='Location', y='Distributed', title='Top 10 States - Total Vaccines
fig.update_layout(xaxis=dict(tickangle=-45), yaxis=dict(type='linear'))
fig.show()
```

Figure 22 : Code for Top 10 states with highest distribution

- I performed the group by on Location and the Distributed (number of vaccines) columns.
- This provided me with the state wise total number of vaccine distribution count.

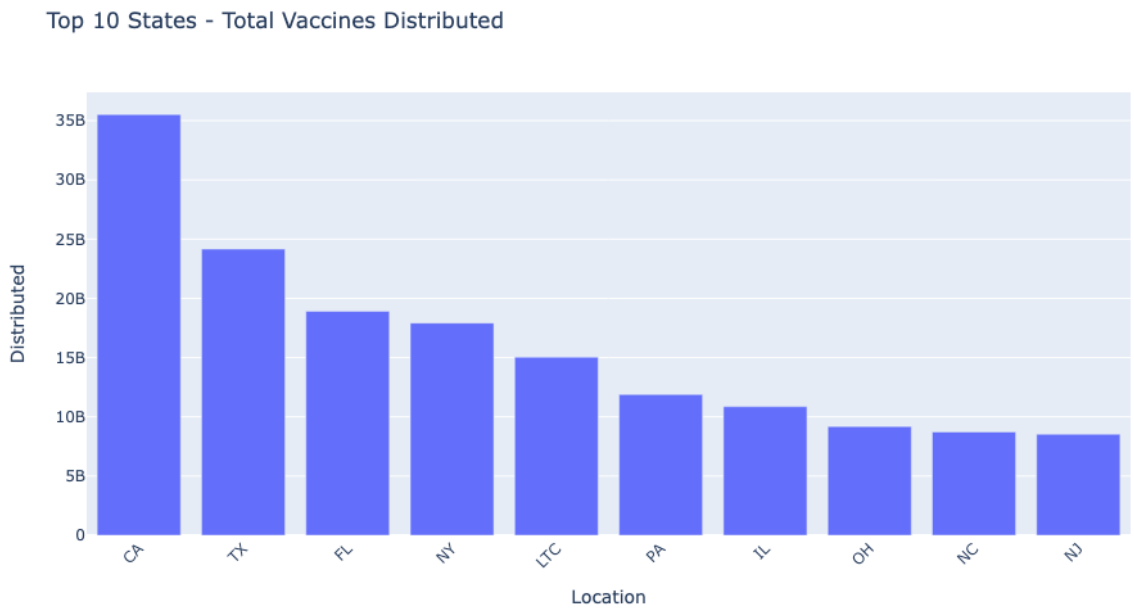


Figure 23 : Top 10 states with highest distribution

- From the figure 23 we can infer that ‘California’ had the highest vaccine distribution followed by the Texas and Florida.
- This states that California is one the most populated states and it had most covid-19 patients as compared to other states.

- While the states like North Carolina and New Jersey has relatively less vaccine distribution which states that these states might have a smaller number of covid-19 patients as compared to other states like California and Texas

Analysis 2

What is the weekly vaccine distribution count throughout the USA ?

Weekwise Vaccine Distribution

In [51]:
```
df_grouped = processed_dataframe.groupby('MMWR_week')['Distributed'].sum().reset_index()

fig = px.line(df_grouped, x='MMWR_week', y='Distributed',
              title='Total Vaccines Distributed by Week',
              labels={'Distributed': 'Total Vaccines Distributed'})
fig.update_layout(xaxis=dict(tickangle=-45), yaxis=dict(type='linear'))
fig.show()

```

Figure 24 : Code for Weekly Vaccine Distribution

- I performed the group by on MMWR_week and the Distributed (number of vaccines) columns.
- This provided me with the weekly total number of vaccine distribution count across USA.

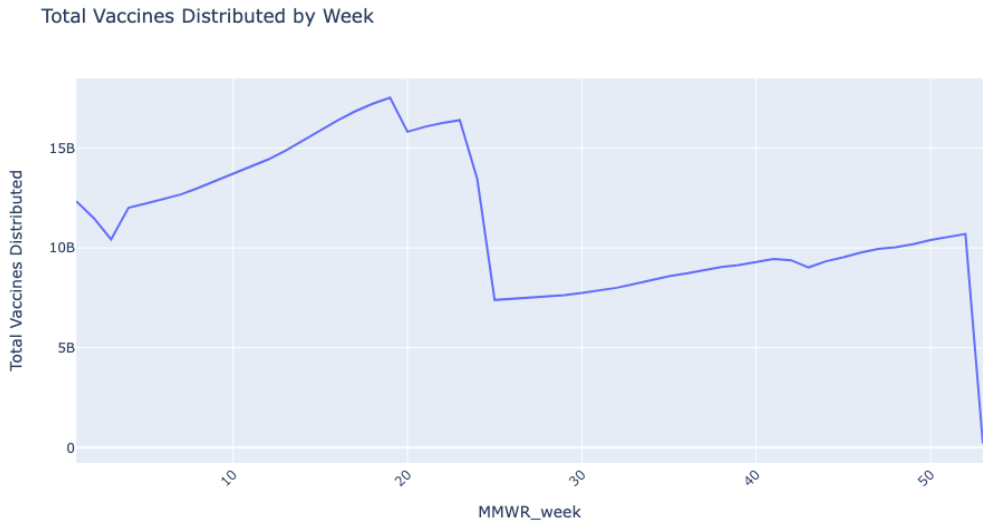


Figure 25 : Weekly Vaccine Distribution

- From figure 25 we can infer which weeks had a peak vaccine distribution.
- This information will help researchers to gather important insight about which period had peak covid-19 spread.
- From the line graph, we can easily infer that the week 19 had the highest vaccine distribution while weeks 25 and 53 had the lowest recorded vaccine distribution.

Analysis 3

What proportion of the total vaccine distribution in week 19 (Highest distribution week) does each company contribute?

Week 19 (highest distribution) with different vaccine providers percentage

In [52]:
```
filtered_df = processed_dataframe[processed_dataframe['MMWR_week'] == 19]

sum_values = filtered_df[['Distributed_Janssen', 'Distributed_Moderna', 'Distributed_Pfizer', 'Distributed_Unk_Ma

fig = px.pie(names=sum_values.index, values=sum_values.values, title='Vaccine Distribution for Week 19')
fig.show()
```

Figure 26 : Code for week 19 distributions for different Manufacturers

- As, from previous analysis we are already aware about the week which has highest distribution of vaccine, I decided to analyze which company is contributing highest in distribution for that week.
- In Data set I had 4 different vaccine distributors, Johnson & Johnson, Pfizer, Moderna and Unknown Manufacturer.
- I filtered the data for these 4 columns for week 19 and summed them up to get the percentage of each one throughout the total distribution

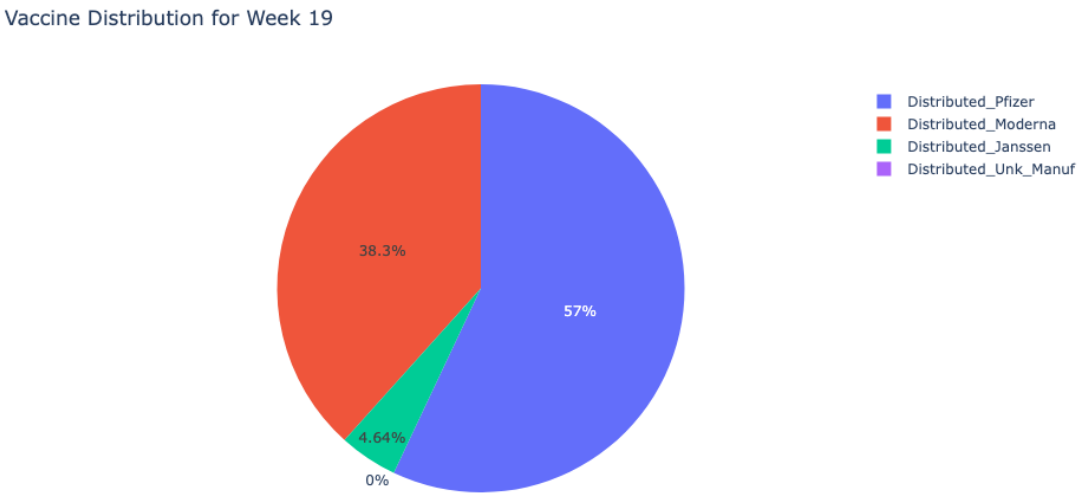


Figure 27 : Week 19 distributions for different Manufacturers

- Above figure shows the contribution of different manufacturers in covid-19 vaccine distribution for week 19.
- We can infer from figure that Pfizer is the leading distributor in week 19, which contributed around 57% of the total distribution followed by Moderna and then Johnson & Johnson for 38.3 % and 4.64 % respectively.
- One key thing to notice here is that there is nothing distributed from any other distributor apart from these 3 as Unknown Manufacturer contributes 0 %.

Analysis 4

In which weeks unknown manufacturer distributed the vaccines ?

```
In [53]:
df_grouped = processed_dataframe.groupby('MMWR_week')['Distributed_Unk_Manuf'].sum().reset_index()

fig = px.line(df_grouped, x='MMWR_week', y='Distributed_Unk_Manuf',
              title='Number of Vaccines distributed by Unknown Manufacturer',
              labels={'Distributed': 'Total Vaccines Distributed'})


fig.add_shape(type='line',
              x0=43, x1=43, y0=0, y1=df_grouped['Distributed_Unk_Manuf'].max(),
              line=dict(color='red', width=2))

fig.add_shape(type='line',
              x0=46, x1=46, y0=0, y1=df_grouped['Distributed_Unk_Manuf'].max(),
              line=dict(color='red', width=2))

fig.update_layout(xaxis=dict(tickangle=-45, title='Week'),
                  yaxis=dict(type='linear', title='Total Vaccines Distributed'))


fig.show()
```

Figure 28 : Code for weekly distributions by Unknown Manufacturer

- In previous analysis we had seen that the unknown manufacturer contributed 0% in distribution.
- So, I checked in which weeks unknown manufacturer had also distributed the vaccines.
- For this I grouped data by MMWR_week and then by using ‘Dsitributed_Unk_Manuf’. This gave me the weekly distribution made by Unknown Manufacturer.

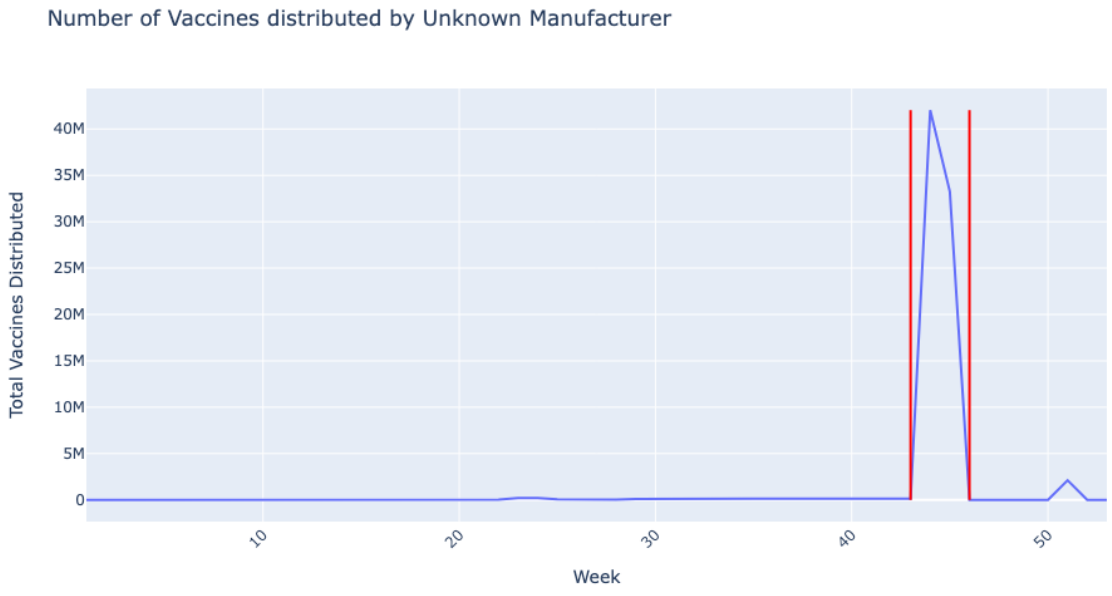


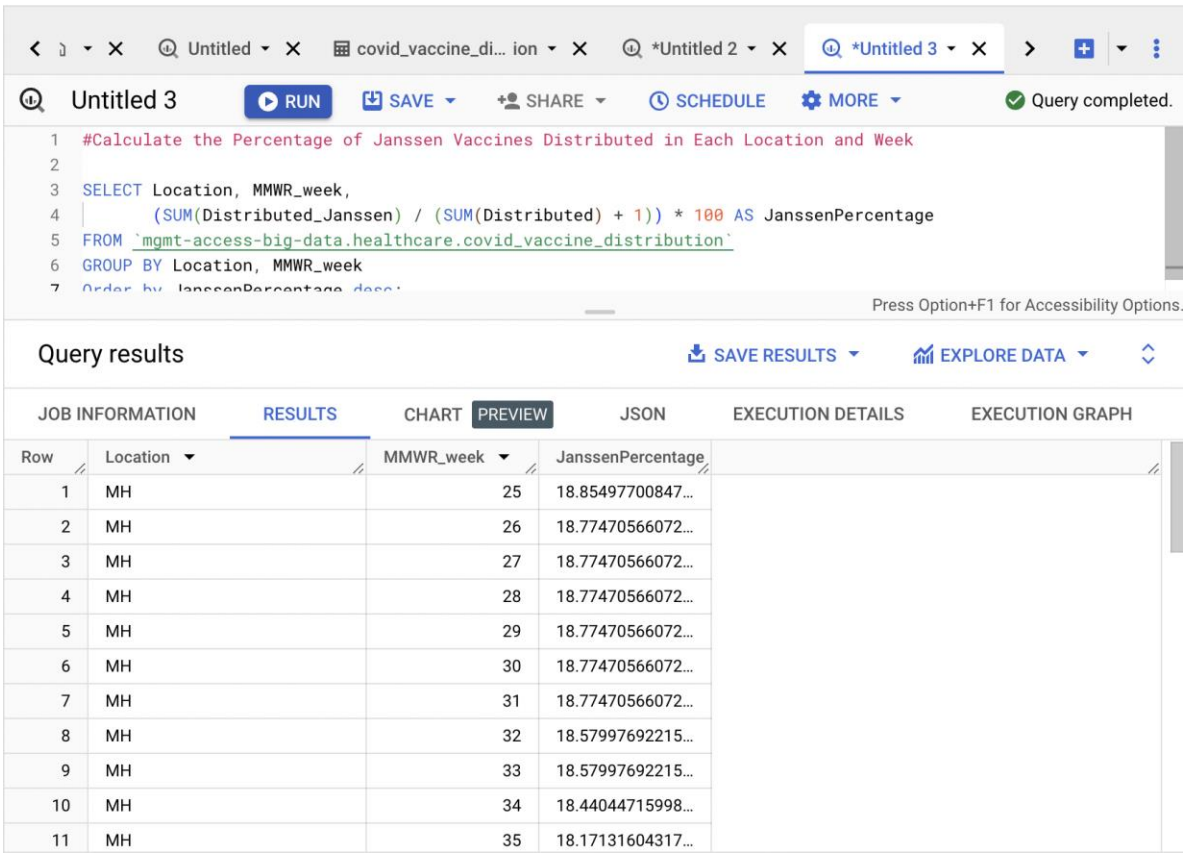
Figure 29 : Weekly distributions by Unknown Manufacturer

- From the analysis I have found that weeks 44 and 45 had seen a peak in distribution via some unknown manufacturers apart from the standard 3, which are Moderna, Pfizer and J&J.
- This showcase that, in these 2 weeks, there might be shortage of vaccines at 3 other manufacturers which promoted some unknown manufacturers to distribute the expected load.
- This also, shows the domination of Pfizer, Moderna and J&J throughout the year as we can see only unknown manufacturers distributing vaccines in 2-3 weeks.

BigQuery Analysis

Analysis 1

Which state/Territory has the highest distribution of Johnson & Johnson Vaccines and how much did it contribute relative to the total distribution?



Row	Location	MMWR_week	JanssenPercentage
1	MH	25	18.85497700847...
2	MH	26	18.77470566072...
3	MH	27	18.77470566072...
4	MH	28	18.77470566072...
5	MH	29	18.77470566072...
6	MH	30	18.77470566072...
7	MH	31	18.77470566072...
8	MH	32	18.57997692215...
9	MH	33	18.57997692215...
10	MH	34	18.44044715998...
11	MH	35	18.17131604317...

Figure 30 : BigQuery implementation for J&J distribution

- Figure 30 shows us the territory ‘MH (Marshall Islands)’ has the highest distribution of J&J vaccines as compared to all other states/territories
- It also highlights which weeks in that specific territory had the highest J & J distribution.
- This analysis helps researcher to compute which state is relying more on which manufacturer

Analysis 2

What are the ranks of states/territories based on the Pfizer distribution?

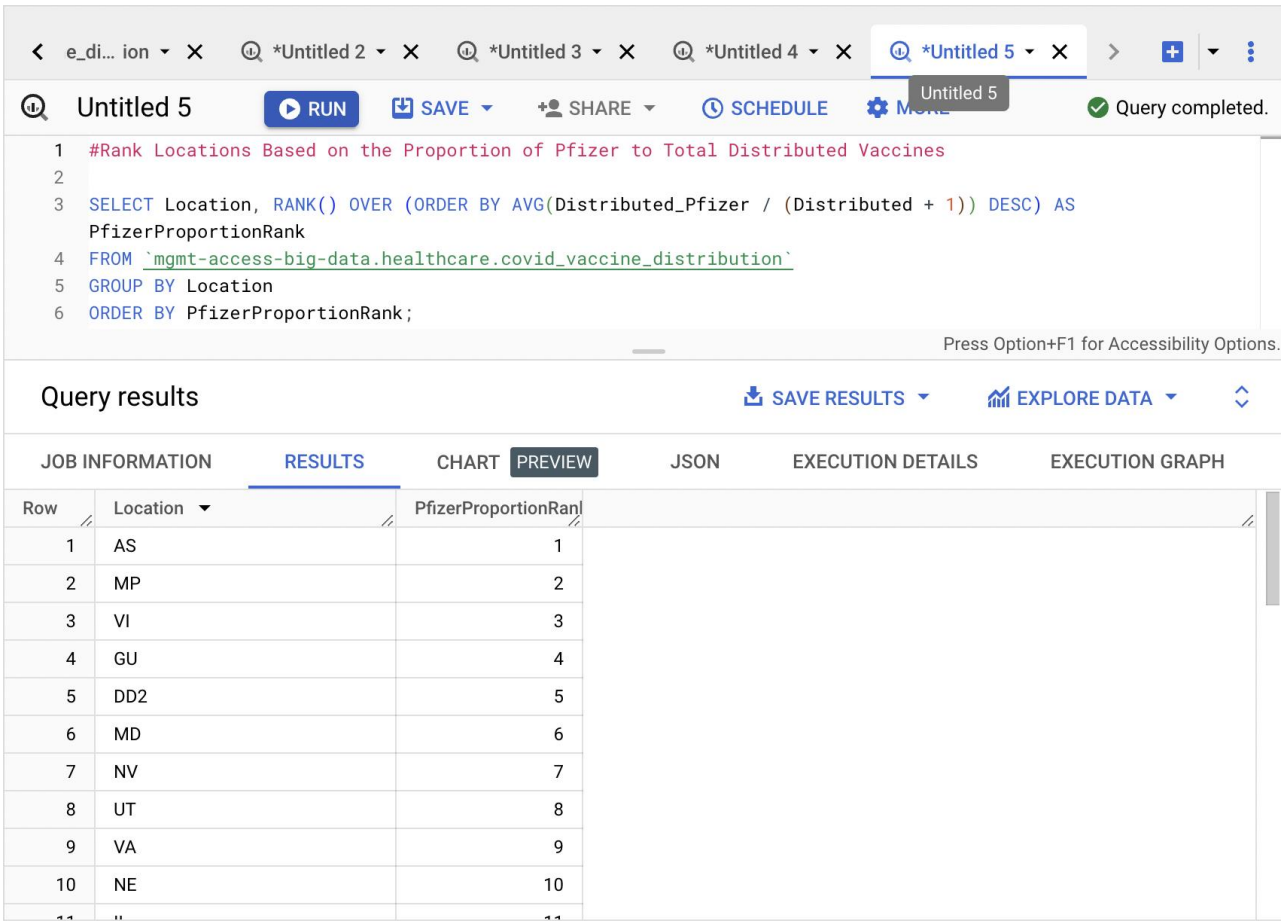


Figure 31 : BigQuery implementation for state ranking based on distribution

- Figure 31 shows us which states/territories had the highest amount of Pfizer distribution.
- The territory ‘AS (American Samoa)’ has the highest distribution percentage for Pfizer as compared to any other states/territories.
- This allows us to monitor which demographic area is relying more on Pfizer and which is not.

After all the analysis I published the analysis results on MongoDB dashboard.

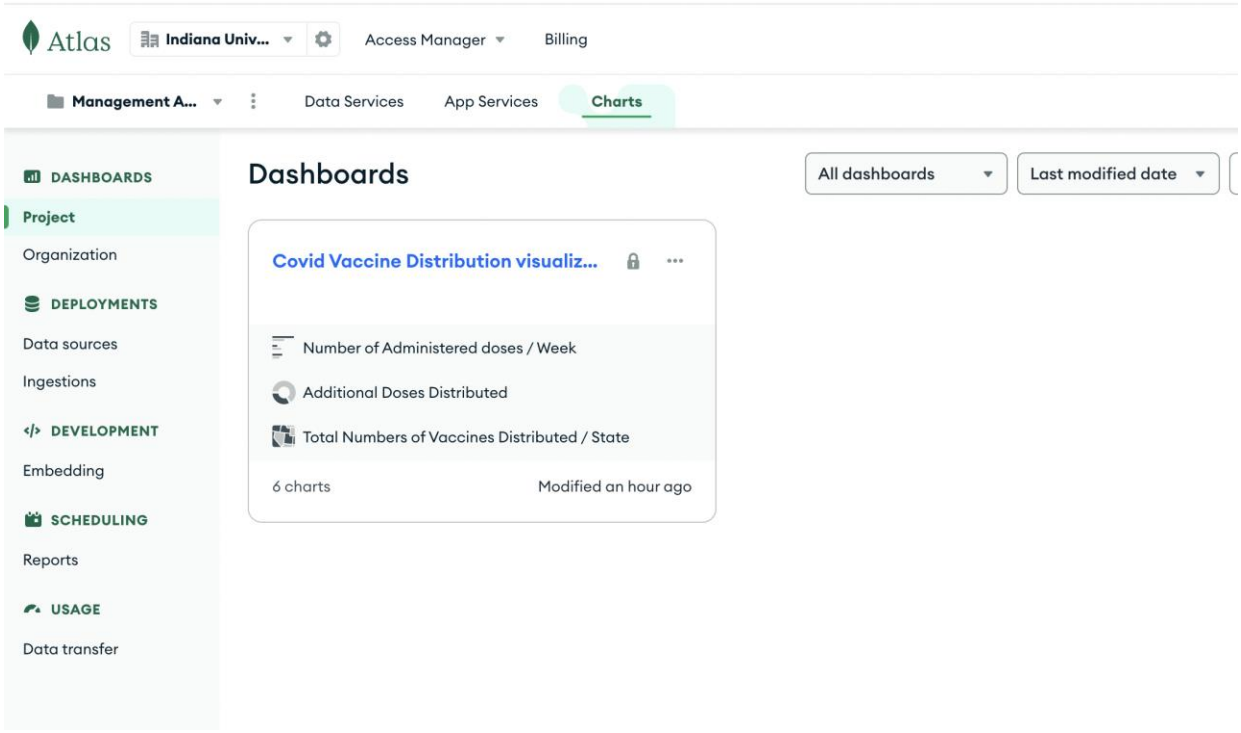


Figure 32 : Published MongoDB Chart Dashboard

Section 5 – Discussions

I used this dataset to analyze what is the vaccine distribution trend in USA for the last year. To reach the final result, I implemented the data pipeline where I gathered the data, ingested it into NoSQL storage and then performed data cleaning and transformation to generate information from it. Initially The data consisted of 38K rows and 109 features, but after data preprocessing and cleaning I had 4K records with the 76 features. This was a significant reduction in data where I grouped all the duplicate records to have a single unique record and removed all the unnecessary columns. I employed multiple technologies to test out the knowledge on big data concepts which includes GCP, Jetstream VM, MongoDB, BigQuery, Docker, etc.

Below are some of the major observations I made during the analysis on data

- There are a total of 65 states and territories in this dataset. Throughout all the states and territories of the USA, California received the highest number of vaccine distribution. It is then followed by Texas and Florida. The

California had the highest vaccine requirements which states that California is the state where USA had seen the major outburst of covid-19 as compared to other states and territories.

- Throughout the year, we can see the trend of vaccine distribution in the USA. From the first week the distribution kept on increasing till it reached its peak value in week 19. After that, the distribution decreased slightly for the next 4-5 weeks till week 25. After week 25, there was again a progressive increase till week 52 and then it dropped again for week 53. So, the USA has seen a good amount of covid-19 outbreak in week 19.
- Pfizer is the leading distributor for Covid-19 Vaccine in the USA. Out of total vaccine distribution Pfizer accounts for almost 57%. It is then followed by Moderna, which accounted for 38%. This is then followed by Johnson & Johnson which accounted for 4.5%. Moreover, there are unknown manufacturers involved in vaccine distribution, but their contribution is negligible and Pfizer, Moderna and Johnson & Johnson lead the vaccine distribution.
- As per the data about additional dose distribution, the majority of additional doses were distributed in the population having age between 18 and 50 years. This accounted for almost 50% of total additional dose distribution. It is followed by distribution for the population between age 50 and 65. And finally, the least number of doses were distributed in the population above age 65. The additional dose requirement is inversely proportional to age.

Challenges Faced

- Detecting bias in the dataset and mitigating that bias is something new concept I tried to make the dataset fair. The aif360 was relatively new to me, and I struggled while deciding on the protected attributes and how those variables will be compared to the target variable to detect the bias. I majorly struggled with deciding the protected variable and its use case in detecting bias.
- Along with this, I was very new to GCP. I wanted to put my preprocessed data on Google Cloud Storage on the bucket so that anyone could access it.

And after that I wanted to fetch that data in BigQuey to run the diagnostics. To navigate through the different services in GCP was the biggest challenge for me. Though I had gone through the assignment provided during the course on the same topic, due to being relatively new to the platform I faced challenges during smooth implementation of data ingestion on bucket and setting up BigQuery

- One major challenge I faced is data preprocessing. I had created some new features like 'Age_65Plus_percentage'. To create this new feature, I had to logically think through the different columns and then produce correct implementation in order to get the exact values I am expecting.

Section 6 – Conclusion

As I conclude the project of Covid-19 Vaccine distribution, this project has given me a lot of insights on how the vaccines distribution has happened in USA in last year. Throughout the project I have implemented different Big Data concepts which include Data pipelining, Virtualization, Cloud concepts and Data Quality maintenance. As a conclusion, this project has provided meaningful information on distribution trends in different states and territories, contribution of different manufacturers and distribution of doses in different age groups. From the analysis, we can conclude that California has the highest need of vaccines whenever the same pandemic will happen in the future. Moreover, Pfizer is the most reliable manufacturer which has a huge production capacity and can be utilized in the future for the same purposes. The People with age > 65 are less likely to affect in the later stages of pandemic, as they need relatively a smaller number of additional doses as compared to other age groups. Based on all this data and analysis, researcher can predict the trend in the future if same pandemic happens, which will eventually help us to fast pace things and will help us to get things under control easily. Moreover, with the help of this data, different companies can get an idea about demographics and can run their campaign

accordingly. The Vaccine manufacturer can easily cater the need in future as they already have insights about where and how much quantity is required.

Section 7 – References

1. https://data.cdc.gov/browse?category=Vaccinations
2. https://pymongo.readthedocs.io/en/stable/atlas.html
3. https://nbviewer.org/github/IBM/AIF360/tree/master/examples/
4. https://www.cloudskillsboost.google/focuses/3692?catalog_rank=%7B%22rank%22%3A1%2C%22num_filters%22%3A0%2C%22has_search%22%3Atrue%7D&parent=catalog&search_id=26980459
5. https://iu.instructure.com/courses/2169303/assignments/15072899
6. https://www.mongodb.com/docs/charts/