

**Aayush Keshari**

**CS 5165**

**PROJECT 5: Exploring COVID-19 Data using Databricks**

**Links**

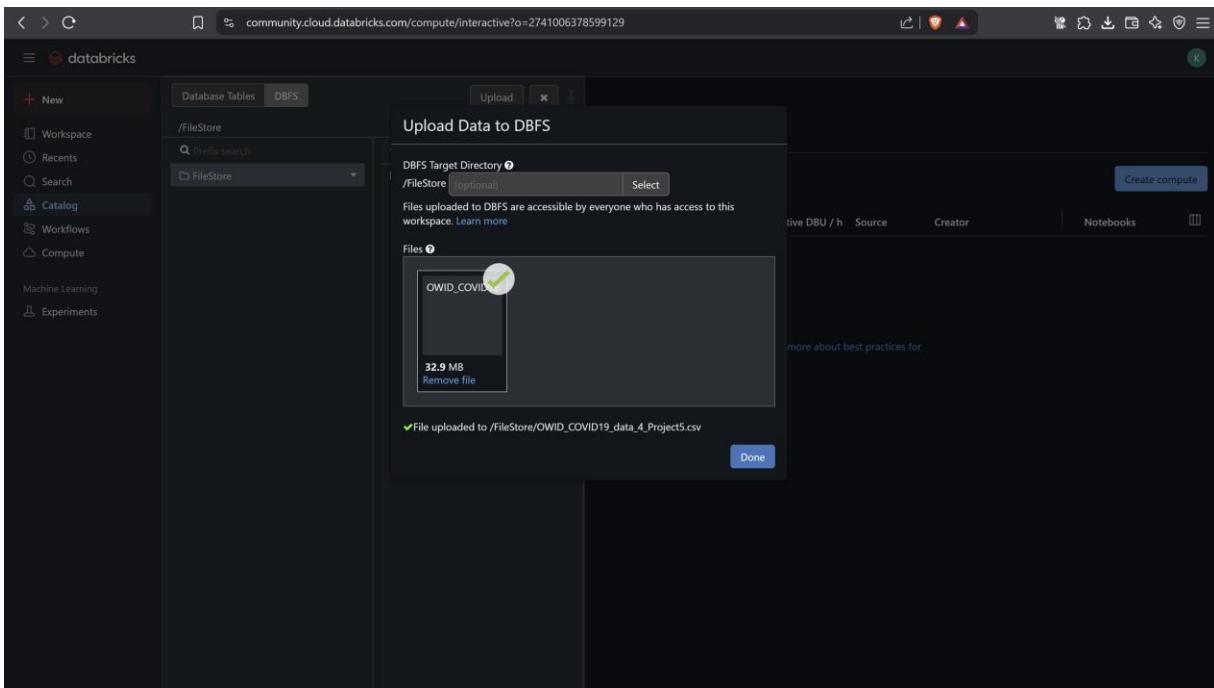
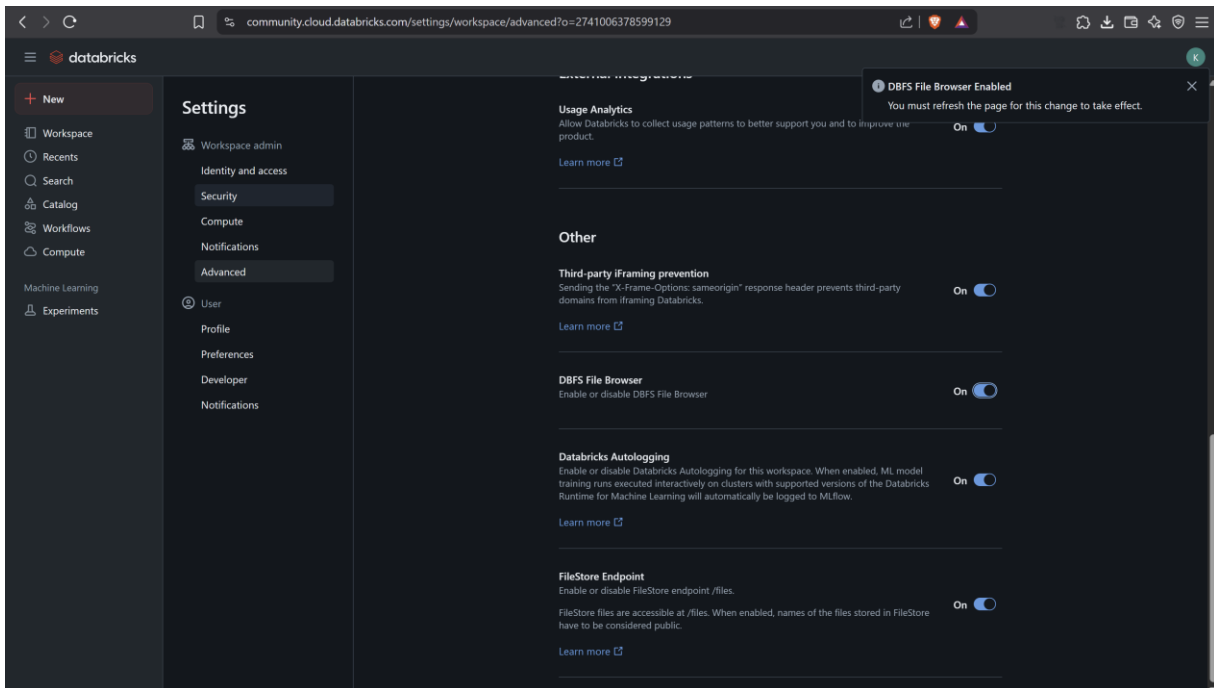
Databricks Notebook Link:

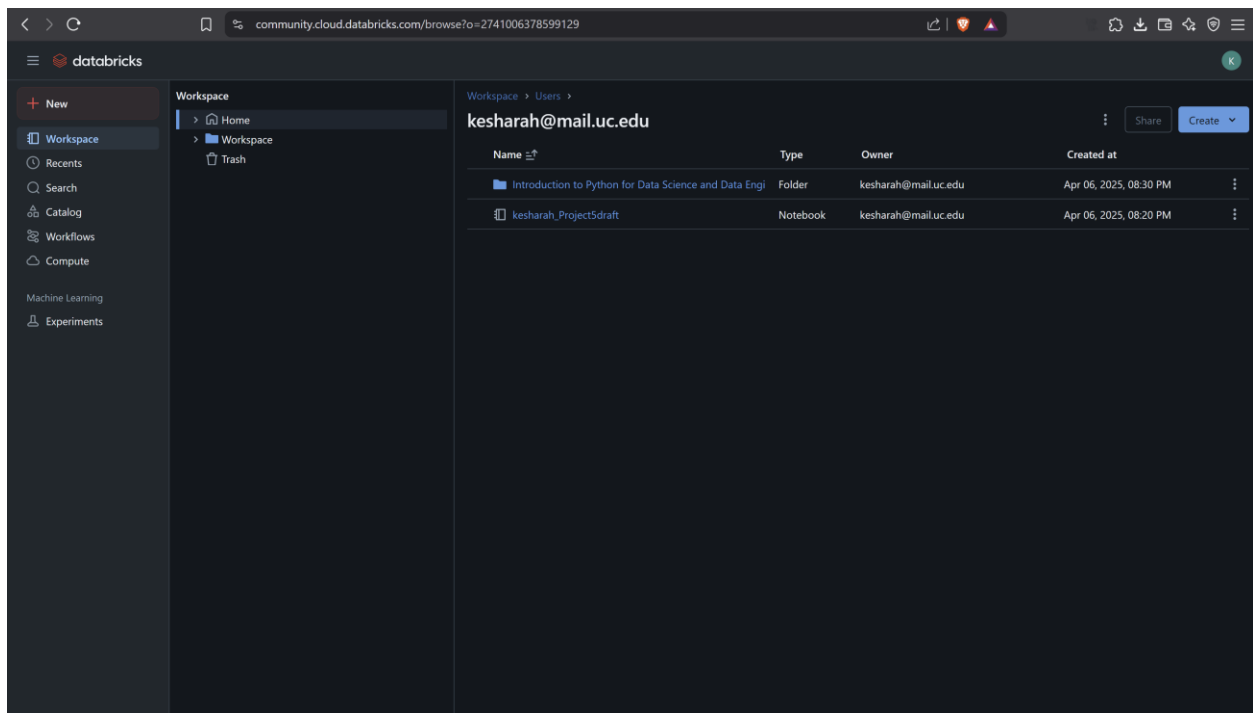
<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bfc/2741006378599129/3779699637522574/1037910492077287/latest.html>

GitHub Repository Link:

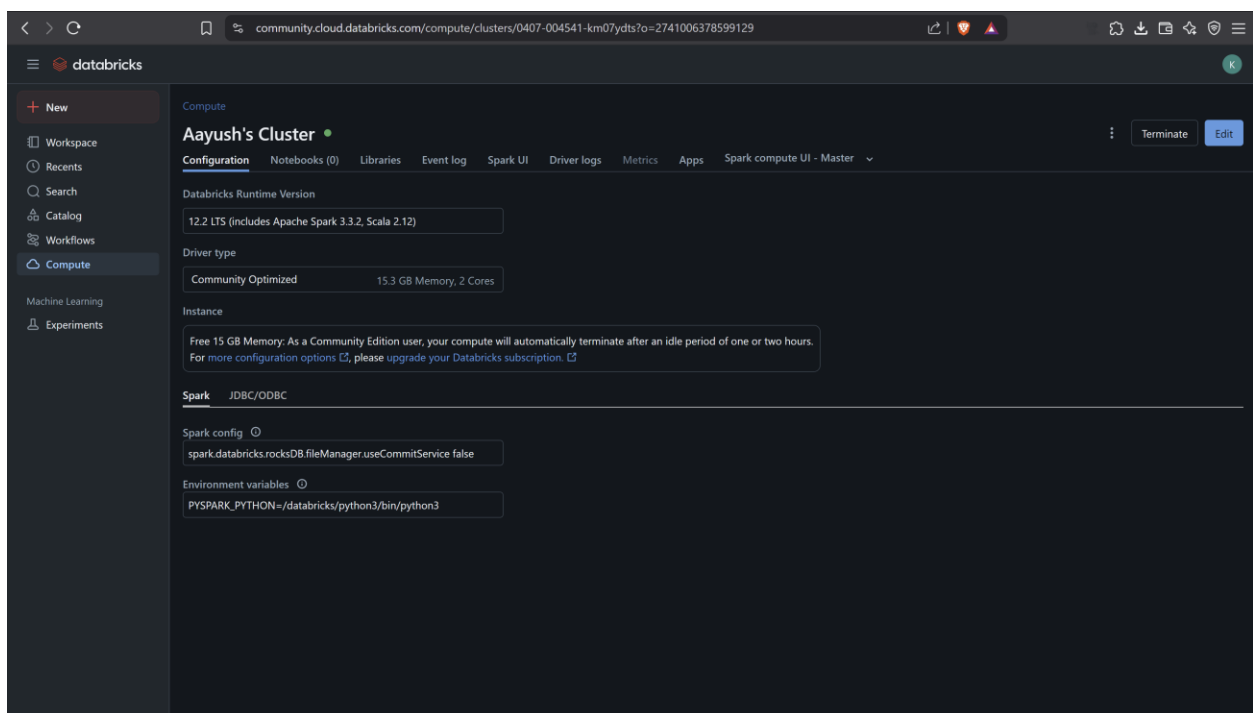
<https://github.com/aayushkeshari/Project-5-Databricks>

# 1. Install Databricks and upload the data.

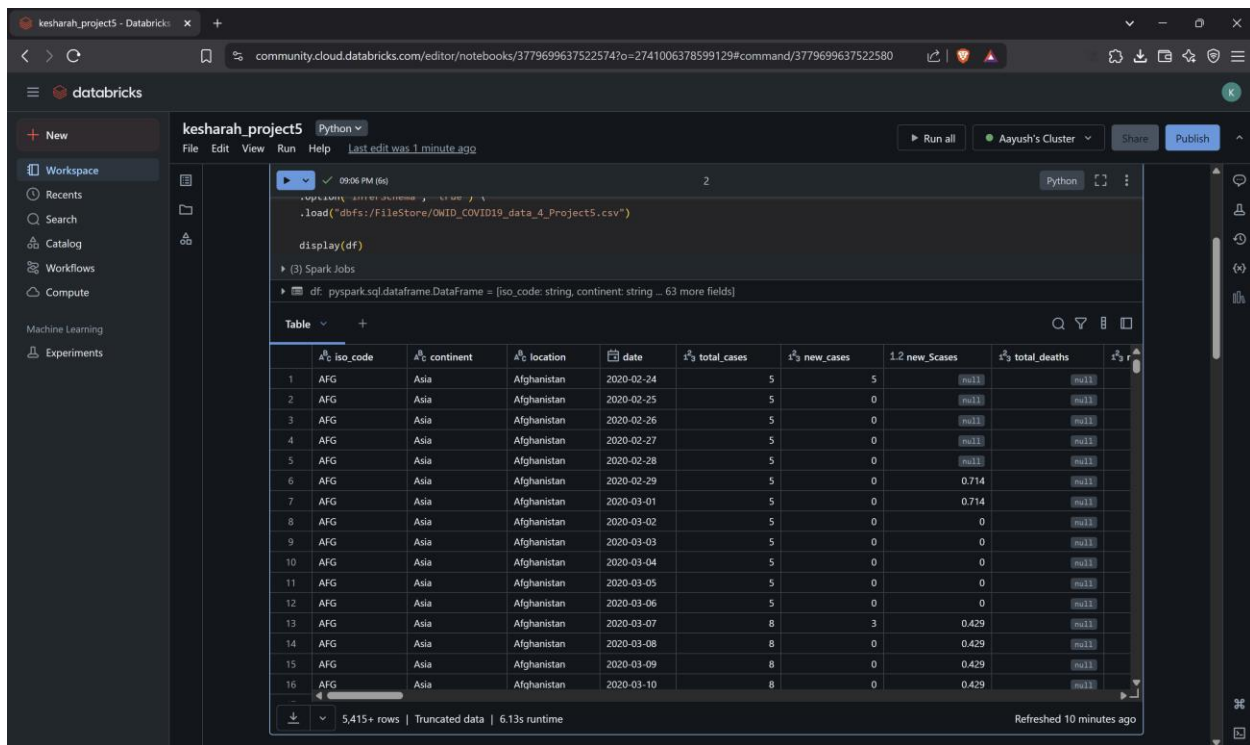




2. Create a Databricks cluster and load the data file using Pandas or PySpark as needed.



You can see the entire file uploaded here.

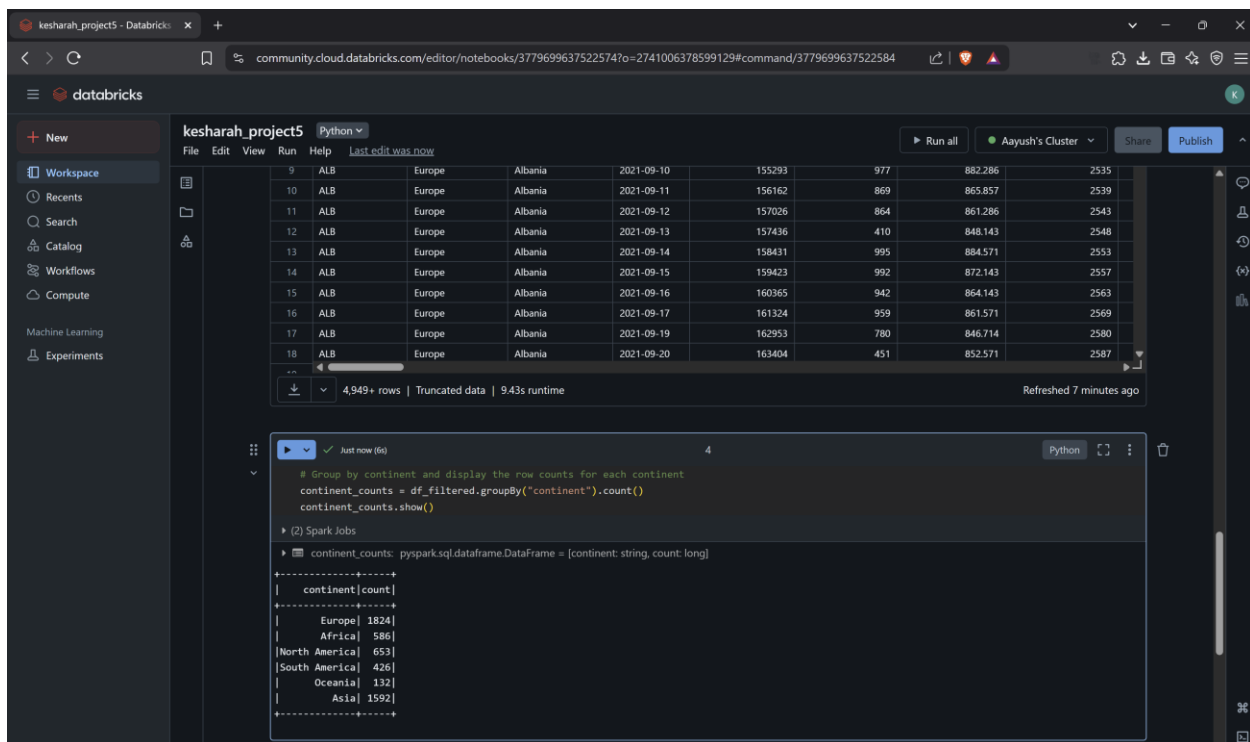


The screenshot shows a Databricks workspace for 'kesharah\_project5'. A notebook cell has been executed, loading a CSV file from 'dbfs:/FileStore/OWID\_COVID19\_data\_4\_Project5.csv'. The data is displayed as a table with 16 rows and 10 columns. The columns are: iso\_code, continent, location, date, total\_cases, new\_cases, new\_scases, total\_deaths, and two additional columns with null values. The data represents COVID-19 statistics for Afghanistan from February to March 2020.

	iso_code	continent	location	date	total_cases	new_cases	new_scases	total_deaths	
1	AFG	Asia	Afghanistan	2020-02-24	5	5	null	null	
2	AFG	Asia	Afghanistan	2020-02-25	5	0	null	null	
3	AFG	Asia	Afghanistan	2020-02-26	5	0	null	null	
4	AFG	Asia	Afghanistan	2020-02-27	5	0	null	null	
5	AFG	Asia	Afghanistan	2020-02-28	5	0	null	null	
6	AFG	Asia	Afghanistan	2020-02-29	5	0	0.714	null	
7	AFG	Asia	Afghanistan	2020-03-01	5	0	0.714	null	
8	AFG	Asia	Afghanistan	2020-03-02	5	0	0	null	
9	AFG	Asia	Afghanistan	2020-03-03	5	0	0	null	
10	AFG	Asia	Afghanistan	2020-03-04	5	0	0	null	
11	AFG	Asia	Afghanistan	2020-03-05	5	0	0	null	
12	AFG	Asia	Afghanistan	2020-03-06	5	0	0	null	
13	AFG	Asia	Afghanistan	2020-03-07	8	3	0.429	null	
14	AFG	Asia	Afghanistan	2020-03-08	8	0	0.429	null	
15	AFG	Asia	Afghanistan	2020-03-09	8	0	0.429	null	
16	AFG	Asia	Afghanistan	2020-03-10	8	0	0.429	null	

The table footer indicates 5,415+ rows, truncated data, and a 6.13s runtime. It was refreshed 10 minutes ago.

3. Filter Records and display the row count by continent.



The screenshot shows the same Databricks workspace. A new notebook cell has been executed, filtering the data by continent and displaying the row counts. The output shows a table with 2 rows and 2 columns: continent and count. The counts are: Europe 1824, Africa 586, North America 653, South America 426, Oceania 132, and Asia 1592.

continent	count
Europe	1824
Africa	586
North America	653
South America	426
Oceania	132
Asia	1592

The table footer indicates 4,949+ rows, truncated data, and a 9.43s runtime. It was refreshed 7 minutes ago.

#### 4. Create Month and Year Columns and display the total record count.

The screenshot shows a Databricks workspace with a notebook named 'kesharah\_project5'. The notebook contains the following Python code:

```
continent_counts: pyspark.sql.dataframe.DataFrame = [continent: string, count: long]
continent|count|
-----+-----+
|Europe|1824|
|Africa|586|
|North America|653|
|South America|426|
|Oceania|132|
|Asia|1592|
-----+-----+

from pyspark.sql.functions import month, year

# Step 1: Create Month and Year columns
df_filtered = df_filtered.withColumn("month", month("date"))
df_filtered = df_filtered.withColumn("year", year("date"))

# Step 2: Filter for September and October 2021
df_filtered = df_filtered.filter(
    (F.col("year") == 2021) &
    (F.col("month").isin([9, 10]))
)

# Step 3: Display the total record count
total_count = df_filtered.count()
print(f"Total record count for September and October 2021: {total_count}")
```

The output shows the total record count for September and October 2021: 5213.

#### 5. Calculate Averages and display the row count by continent by month.

The screenshot shows a Databricks workspace with a notebook named 'kesharah\_project5'. The notebook contains the following Python code:

```
.withColumn("month", month("date")) \
.filter((F.col("year") == 2021) & (F.col("month").isin([9, 10])))

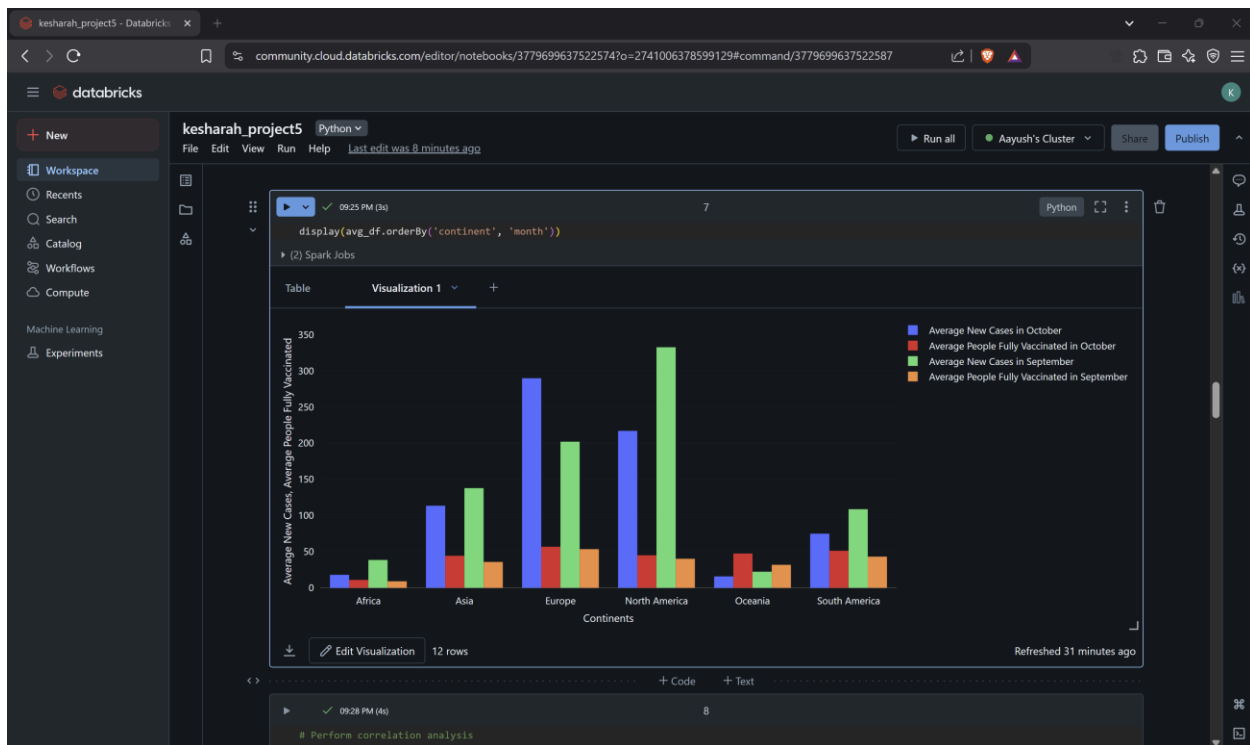
# Calculate averages by continent and month
# Group by 'continent' and 'month' and calculate averages for the specified metrics
avg_df = df_filtered.groupBy('continent', 'month').agg(
    F.mean('people_fully_vaccinated_phun').alias('average_people_fully_vaccinated'),
    F.mean('new_cases_pail').alias('average_new_cases'),
    F.mean('excess_mortality').alias('average_excess_mortality')
).orderBy('continent', 'month')

# Display the results
avg_df.show()
```

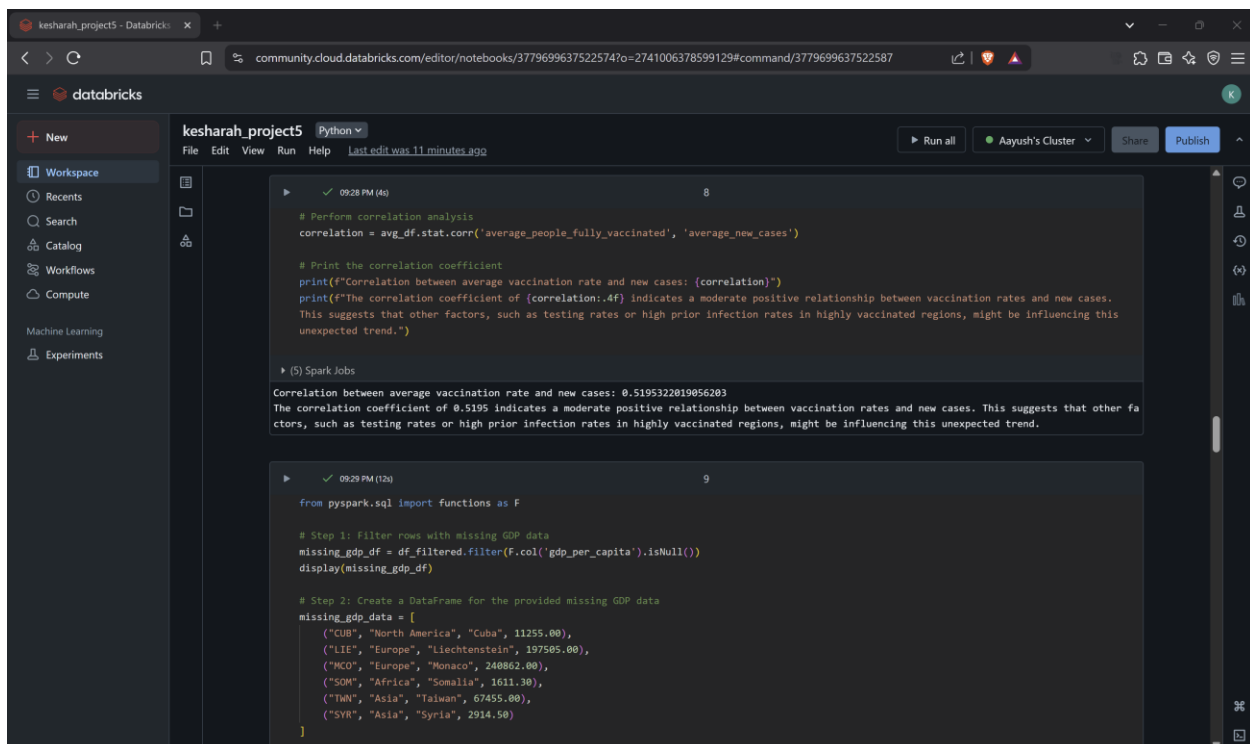
The output shows the row count by continent by month:

continent	month	average_people_fully_vaccinated	average_new_cases	average_excess_mortality
Africa	9	9.050997067448682	38.56174506172839	25.88
Africa	10	10.979402173913043	18.04095400238949	9.0975
Asia	9	35.818799999999996	137.81524184397162	45.19416666666667
Asia	10	44.35388982035929	113.52837542896364	46.09333333333334
Europe	9	53.51533724340176	202.05234275362315	12.174639175257733
Europe	10	56.79475073313782	289.7618099579243	12.14
North America	9	48.22337078651684	332.61750579710144	74.689
North America	10	45.033721973094174	217.01673913043487	null
Oceania	9	31.851630434782614	22.318063636363636	-6.7
Oceania	10	47.44908163265306	15.83905232581394	-13.0
South America	9	43.167231404958684	108.70364066852369	9.0475
South America	10	51.30851528384279	74.9803790322581	7.786666666666668

## 6. Plotting a Bar Chart.



## 7. Run Correlation Analysis.



## 8. Fill missing GDP (PPP) per capita.

The screenshot shows a Databricks notebook interface for 'kesharah\_project5'. The code cell contains the following Python code:

```
F.coalesce(F.col("gdp_per_capita"), F.col("gdp_per_capita_new"))
).drop("gdp_per_capita_new") # Drop the new column after merging

# Step 5: Display the updated DataFrame to verify missing GDP data is filled
display(df_with_gdp)
```

The output shows 12 Spark jobs completed. Below the code, a table view displays the DataFrame with the following columns: iso\_code, continent, location, date, total\_cases, new\_cases, new\_Scapes, total\_deaths, and a truncated column. The table contains 15 rows of data for Andorra (AND) from 2021-09-01 to 2021-09-14.

	iso_code	continent	location	date	total_cases	new_cases	new_Scapes	total_deaths	
1	AND	Europe	Andorra	2021-09-01	15046	13	4.571	130	
2	AND	Europe	Andorra	2021-09-02	15052	6	5.143	130	
3	AND	Europe	Andorra	2021-09-03	15055	3	4.286	130	
4	AND	Europe	Andorra	2021-09-04	15055	0	4.286	130	
5	AND	Europe	Andorra	2021-09-05	15055	0	4.286	130	
6	AND	Europe	Andorra	2021-09-06	15069	14	5.286	130	
7	AND	Europe	Andorra	2021-09-07	15070	1	5.286	130	
8	AND	Europe	Andorra	2021-09-08	15070	0	3.429	130	
9	AND	Europe	Andorra	2021-09-09	15078	8	3.714	130	
10	AND	Europe	Andorra	2021-09-10	15083	5	4	130	
11	AND	Europe	Andorra	2021-09-11	15083	0	4	130	
12	AND	Europe	Andorra	2021-09-12	15083	0	4	130	
13	AND	Europe	Andorra	2021-09-13	15096	13	3.857	130	
14	AND	Europe	Andorra	2021-09-14	15099	3	4.143	130	
15									

1,431 rows | 11.73s runtime

The screenshot shows a Databricks notebook interface for 'kesharah\_project5'. The top table view displays data for GDP per capita with the following columns: iso\_code, continent, location, and 1.2 gdp\_per\_capita\_new. The table contains 6 rows of data for various countries.

	iso_code	continent	location	1.2 gdp_per_capita_new
1	CUB	North America	Cuba	11255
2	LIE	Europe	Liechtenstein	197505
3	MCO	Europe	Monaco	240862
4	SOM	Africa	Somalia	1611.3
5	TWN	Asia	Taiwan	67455
6	SYR	Asia	Syria	2914.5

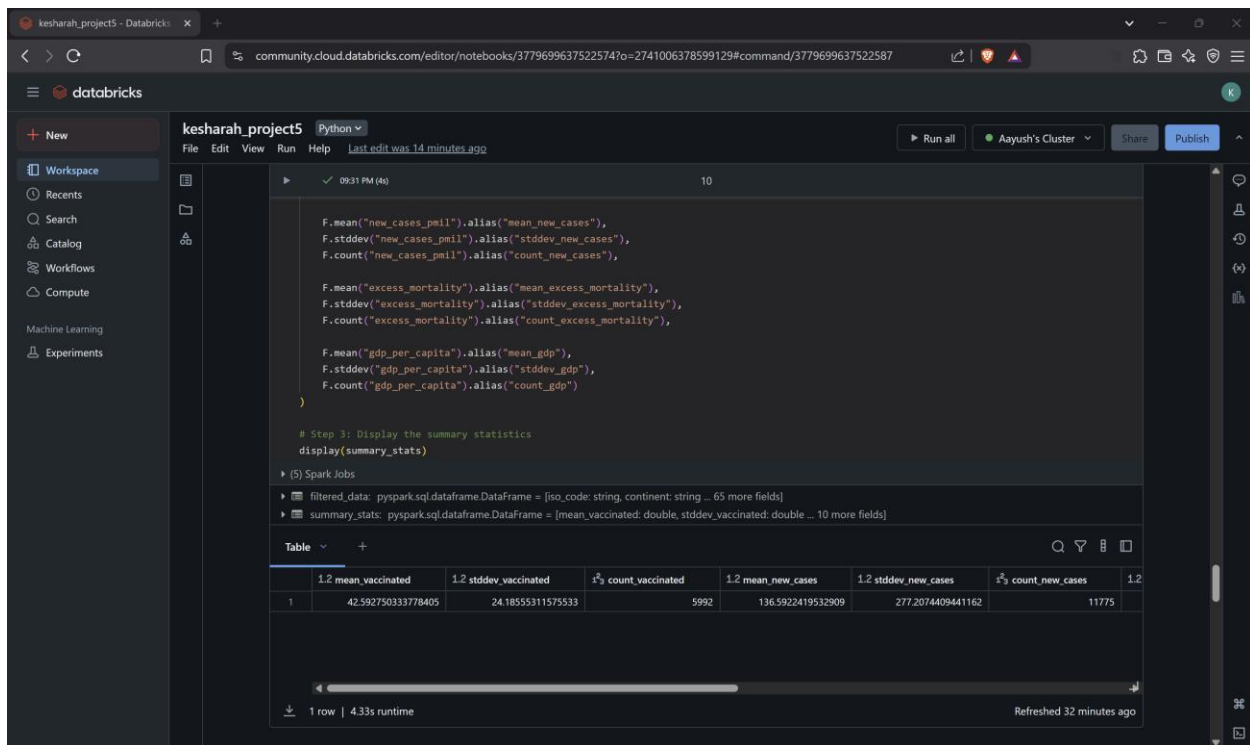
6 rows | 11.73s runtime

The bottom table view displays a DataFrame with the following columns: iso\_code, continent, location, date, total\_cases, new\_cases, new\_Scapes, total\_deaths, and a truncated column. The table contains 15 rows of data for Afghanistan (AFG) from 2021-09-01 to 2021-09-14.

	iso_code	continent	location	date	total_cases	new_cases	new_Scapes	total_deaths	
1	AFG	Asia	Afghanistan	2021-09-01	153260	40	76.857	7123	
2	AFG	Asia	Afghanistan	2021-09-02	153306	46	69.143	7127	
3	AFG	Asia	Afghanistan	2021-09-03	153375	69	59.286	7127	
4	AFG	Asia	Afghanistan	2021-09-04	153395	20	55.429	7128	
5	AFG	Asia	Afghanistan	2021-09-05	153423	28	55.714	7132	
6	AFG	Asia	Afghanistan	2021-09-06	153534	111	55.143	7141	
7	AFG	Asia	Afghanistan	2021-09-07	153626	92	58	7144	
8	AFG	Asia	Afghanistan	2021-09-08	153736	110	68	7151	
9	AFG	Asia	Afghanistan	2021-09-09	153840	104	76.286	7157	
10	AFG	Asia	Afghanistan	2021-09-10	153962	122	83.857	7164	
11	AFG	Asia	Afghanistan	2021-09-11	153982	20	83.857	7167	
12	AFG	Asia	Afghanistan	2021-09-12	153990	8	81	7167	
13	AFG	Asia	Afghanistan	2021-09-13	154094	104	80	7169	
14	AFG	Asia	Afghanistan	2021-09-14	154180	86	79.143	7171	
15									

5,164+ rows | Truncated data | 11.73s runtime

## 9. Create Summary/Descriptive Statistics Table.



The screenshot shows a Databricks notebook interface for 'kesharah\_project5'. The code calculates summary statistics for three variables: new\_cases\_pm1, excess\_mortality, and gdp\_per\_capita. It uses F.mean, F.stddev, and F.count functions. The results are displayed in a table with 1 row and 7 columns.

```
Python
```

```
F.mean("new_cases_pm1").alias("mean_new_cases"),
F.stddev("new_cases_pm1").alias("stddev_new_cases"),
F.count("new_cases_pm1").alias("count_new_cases"),

F.mean("excess_mortality").alias("mean_excess_mortality"),
F.stddev("excess_mortality").alias("stddev_excess_mortality"),
F.count("excess_mortality").alias("count_excess_mortality"),

F.mean("gdp_per_capita").alias("mean_gdp"),
F.stddev("gdp_per_capita").alias("stddev_gdp"),
F.count("gdp_per_capita").alias("count_gdp")
)

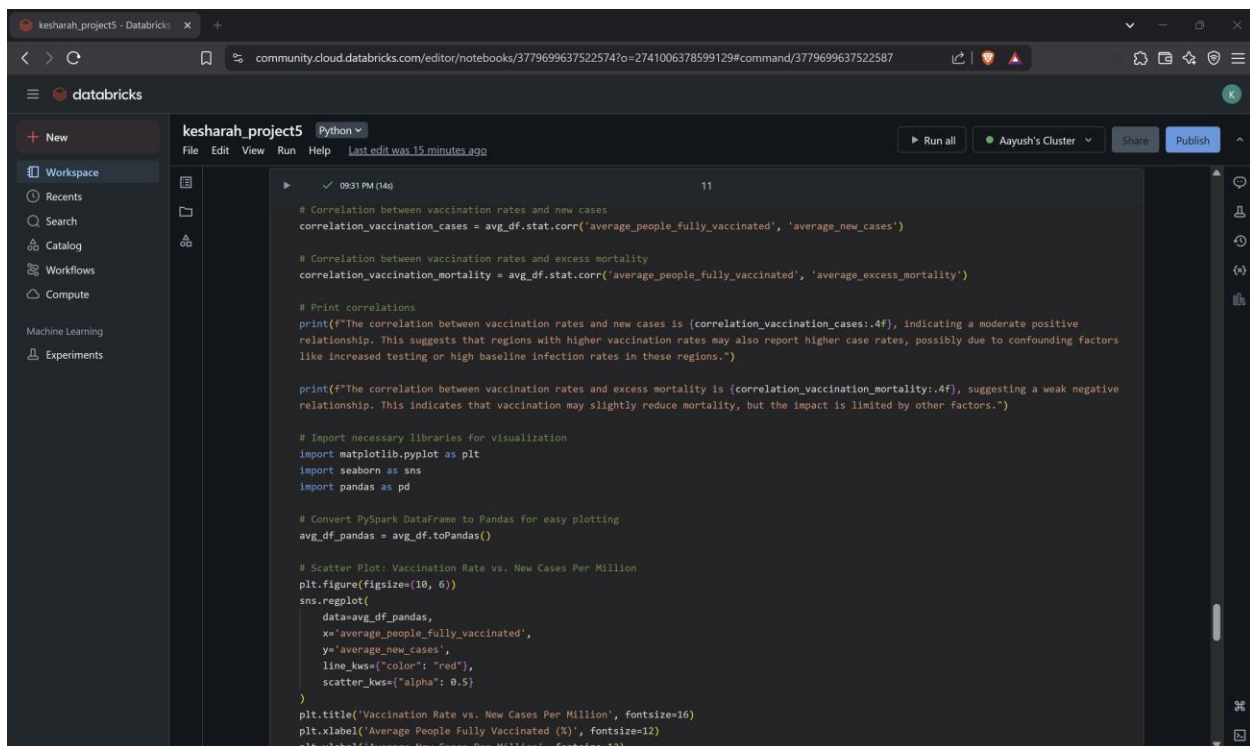
# Step 3: Display the summary statistics
display(summary_stats)
```

Table

	1.2 mean_vaccinated	1.2 stddev_vaccinated	1.2 count_vaccinated	1.2 mean_new_cases	1.2 stddev_new_cases	1.2 count_new_cases
1	42.592750333778405	24.18555311575533	5992	136.5922419532909	277.2074409441162	11775

1 row | 4.33s runtime

## 10. Reporting Results.



The screenshot shows a Databricks notebook interface for 'kesharah\_project5'. The code calculates correlations between vaccination rates and new cases, and between vaccination rates and excess mortality. It also imports necessary libraries for visualization (matplotlib, seaborn, pandas) and creates a scatter plot titled 'Vaccination Rate vs. New Cases Per Million'.

```
Python
```

```
# Correlation between vaccination rates and new cases
correlation_vaccination_cases = avg_df.stat.corr('average_people_fully_vaccinated', 'average_new_cases')

# Correlation between vaccination rates and excess mortality
correlation_vaccination_mortality = avg_df.stat.corr('average_people_fully_vaccinated', 'average_excess_mortality')

# Print correlations
print(f"The correlation between vaccination rates and new cases is {correlation_vaccination_cases:.4f}, indicating a moderate positive relationship. This suggests that regions with higher vaccination rates may also report higher case rates, possibly due to confounding factors like increased testing or high baseline infection rates in these regions.")

print(f"The correlation between vaccination rates and excess mortality is {correlation_vaccination_mortality:.4f}, suggesting a weak negative relationship. This indicates that vaccination may slightly reduce mortality, but the impact is limited by other factors.")

# Import necessary libraries for visualization
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Convert PySpark DataFrame to Pandas for easy plotting
avg_df_pandas = avg_df.toPandas()

# Scatter Plot: Vaccination Rate vs. New Cases Per Million
plt.figure(figsize=(10, 6))
sns.regplot(
    data=avg_df_pandas,
    x='average_people_fully_vaccinated',
    y='average_new_cases',
    line_kws={"color": "red"},
    scatter_kws={"alpha": 0.5}
)

plt.title('Vaccination Rate vs. New Cases Per Million', fontsize=16)
plt.xlabel('Average People Fully Vaccinated (%)', fontsize=12)
plt.ylabel('Average New Cases Per Million', fontsize=12)
```



+

new

Workspace

Recents

Search

Catalog

Workflows

Compute

Machine Learning

Experiments

kesharah\_project5

Python

File Edit View Run Help

Last edit was 16 minutes ago

Run all

Aayush's Cluster

Share

Publish

09:31 PM (14s)

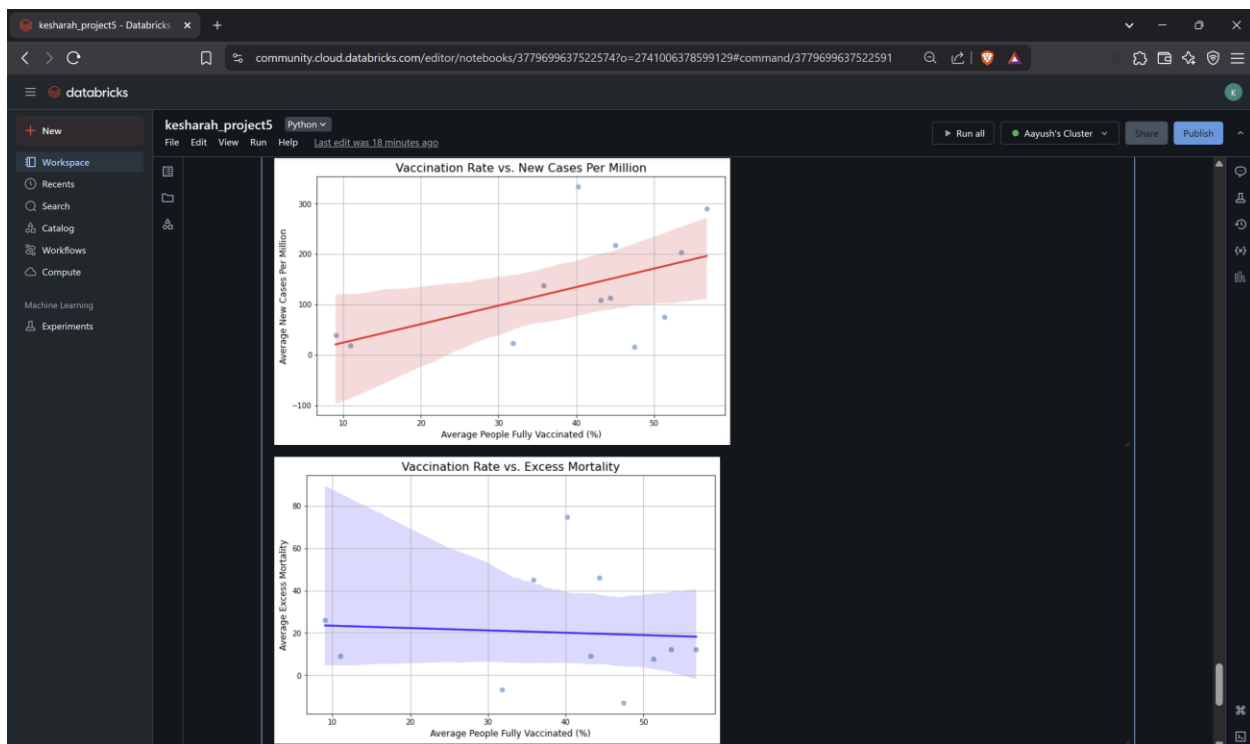
11

```
data=avg_df_pandas,
x='average_people_fully_vaccinated',
y='average_new_cases',
line_kws={"color": "red"},
scatter_kws={"alpha": 0.5}
)
plt.title('Vaccination Rate vs. New Cases Per Million', fontsize=16)
plt.xlabel('Average People Fully Vaccinated (%)', fontsize=12)
plt.ylabel('Average New Cases Per Million', fontsize=12)
plt.grid(True)
plt.show()

# Scatter Plot: Vaccination Rate vs. Excess Mortality
plt.figure(figsize=(10, 6))
sns.regplot(
    data=avg_df_pandas,
    x='average_people_fully_vaccinated',
    y='average_excess_mortality',
    line_kws={"color": "blue"},
    scatter_kws={"alpha": 0.5}
)
plt.title('Vaccination Rate vs. Excess Mortality', fontsize=16)
plt.xlabel('Average People Fully Vaccinated (%)', fontsize=12)
plt.ylabel('Average Excess Mortality', fontsize=12)
plt.grid(True)
plt.show()
```

(14) Spark Jobs

The correlation between vaccination rates and new cases is 0.5195, indicating a moderate positive relationship. This suggests that regions with higher vaccination rates may also report higher case rates, possibly due to confounding factors like increased testing or high baseline infection rates in these regions.  
The correlation between vaccination rates and excess mortality is -0.0940, suggesting a weak negative relationship. This indicates that vaccination may slightly reduce mortality, but the impact is limited by other factors.



## Findings

The analysis indicates a **moderate positive correlation ( $r = 0.5195$ )** between COVID-19 vaccination rates and the number of new cases per million during the months of September and October 2021. This observation suggests that regions with higher vaccination coverage may simultaneously report increased case counts. While this relationship appears counterintuitive, it is likely influenced by confounding variables such as elevated testing rates, higher population densities, or historically elevated baseline infection levels in areas with robust vaccination initiatives. Moreover, vaccination may induce behavioral adaptations such as decreased adherence to masking and social distancing protocols particularly in populations that perceive themselves as protected, further complicating the observed association.

Conversely, the **weak negative correlation ( $r = -0.0940$ )** between vaccination rates and excess mortality is consistent with the anticipated protective effect of vaccines against severe disease outcomes and death. Although the magnitude of this association is relatively modest, it provides evidence that vaccination contributes to a reduction in mortality. It is important to acknowledge, however, that other factors such as the quality of healthcare systems, demographic characteristics, and the virulence of circulating variants also play significant roles in shaping mortality trends. These findings underscore the importance of a multifaceted public health strategy, wherein vaccination serves as a cornerstone for mitigating disease severity but must be integrated with additional interventions to effectively control case incidence.