

## Project 5

Name: Jash Dedhia

M-ID: M15047151 (dedhiaja)

Date: November 17, 2024

Github Link: <https://github.com/dedhiaja/Project-5>

2248 Courses

Home

Announcements

Assignments

Discussions

Grades

People

Files

Syllabus

Quizzes

Modules

BigBlueButton

Collaborations

Chat

Echo360

Google Drive

RedShelf Course Materials

Follett Discover

Library Resources

Media Gallery

Zoom

TopHat

Microsoft OneDrive

Search entries or author...

All

Sort

View Split Screen

Expand Threads

RK Rashmi Kansakar AUTHOR | TEACHER

Created Nov 12 8:12pm | Posted Nov 12 8:12pm

### Clarification for Project 5: Databricks

- **Reminder:** Limit the data set to only September and October 2021 for all questions (including the last three questions).
- **Filter:** Exclude rows where the continent column has null values.
- **Data Sources:** It appears that some are using the data set from the website, while others are using the attached CSV.
  - The CSV contains columns named *people\_fully\_vaccinated\_phun* and *new\_cases\_pmil*.
  - The website data contains *people\_fully\_vaccinated\_per\_hundred* and *new\_cases\_per\_million*.
  - You may use either data set for the assignment.
- **Missing GDP PPP Per Capita from the CSV file:** Please use the following values for countries with missing GDP PPP per capita:

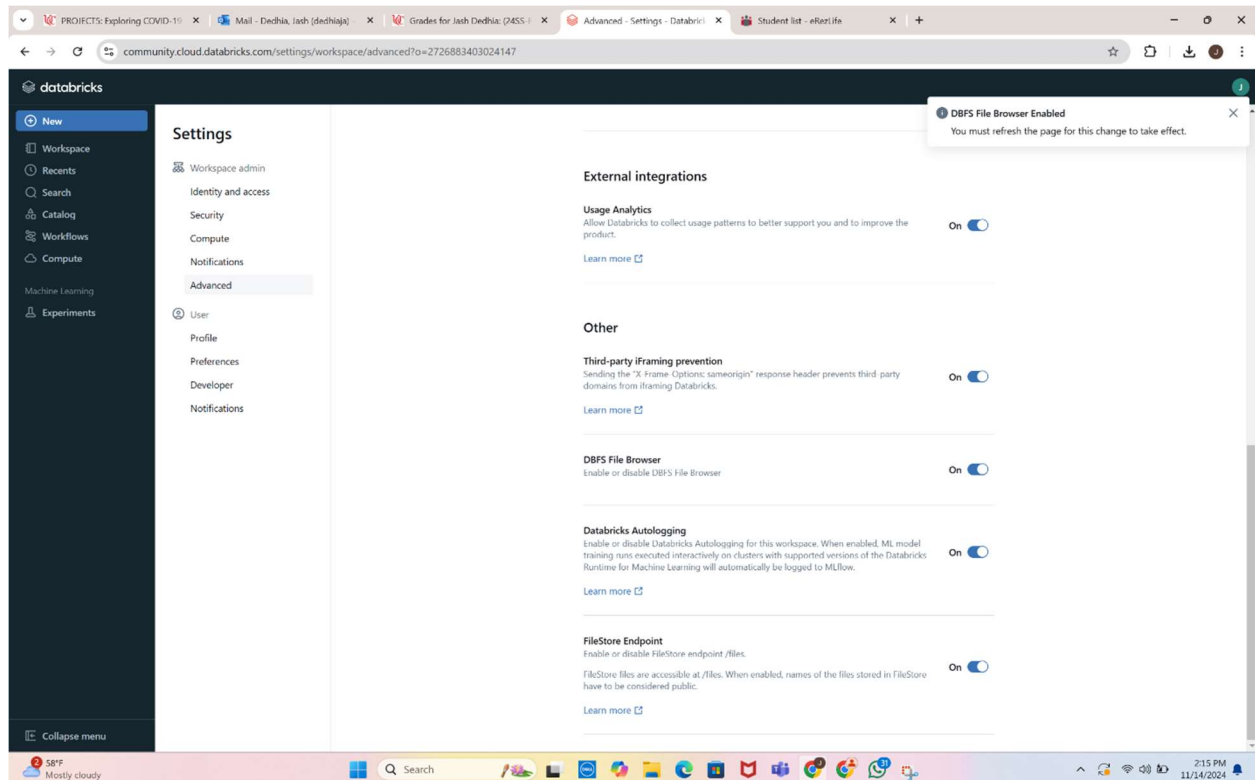
iso_code	continent	location	gdp_per_capita
CUB	North America	Cuba	\$ 11,255.00
LIE	Europe	Liechtenstein	\$ 197,505.00
MCO	Europe	Monaco	\$ 240,862.00
SOM	Africa	Somalia	\$ 1,611.30
TWN	Asia	Taiwan	\$ 67,455.00
SYR	Asia	Syria	\$ 2,914.50

This topic is closed for comments.

61°F Sunny 12:44 PM 11/17/2024

Following professor's instruction that for all the questions the data set has to be limited to September and October 2021. (Screenshot attached above)

1.



The screenshot shows the Databricks Settings page for a workspace. The left sidebar contains navigation links: New, Workspace, Recents, Search, Catalog, Workflows, Compute, Machine Learning, and Experiments. The main content area is titled 'Settings' and includes sections for Workspace admin, Identity and access, Security, Compute, Notifications, and Advanced. The 'Advanced' section is currently selected, showing options for External integrations, Other, DBFS File Browser, Databricks Autologging, and FileStore Endpoint. A notification at the top right states 'DBFS File Browser Enabled' and 'You must refresh the page for this change to take effect.'

**Settings**

- Workspace admin
- Identity and access
- Security
- Compute
- Notifications
- Advanced

**External integrations**

- Usage Analytics**  
Allow Databricks to collect usage patterns to better support you and to improve the product. **On**  
[Learn more](#)

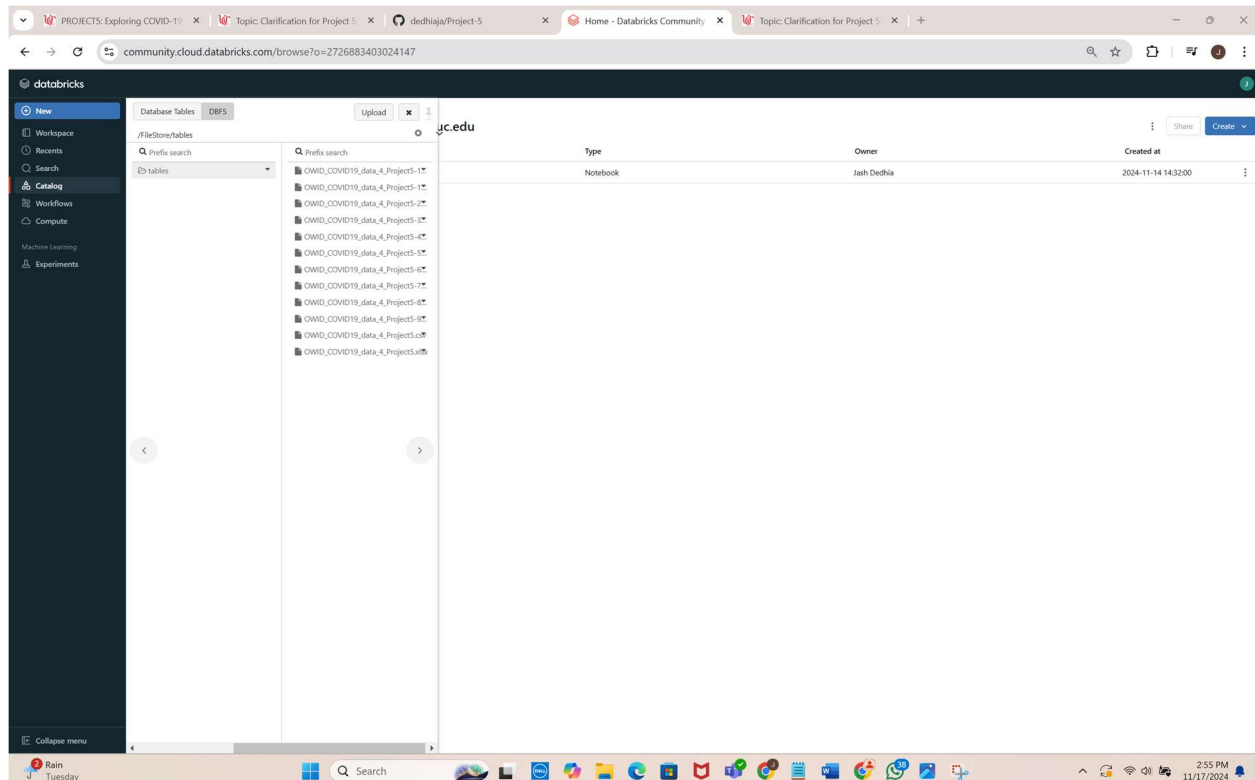
**Other**

- Third-party iFraming prevention**  
Sending the "X-Frame-Options: sameorigin" response header prevents third-party domains from iFraming Databricks. **On**  
[Learn more](#)

**DBFS File Browser**  
Enable or disable DBFS File Browser. **On**

**Databricks Autologging**  
Enable or disable Databricks Autologging for this workspace. When enabled, ML model training runs executed interactively on clusters with supported versions of the Databricks Runtime for Machine Learning will automatically be logged to MLflow. **On**  
[Learn more](#)

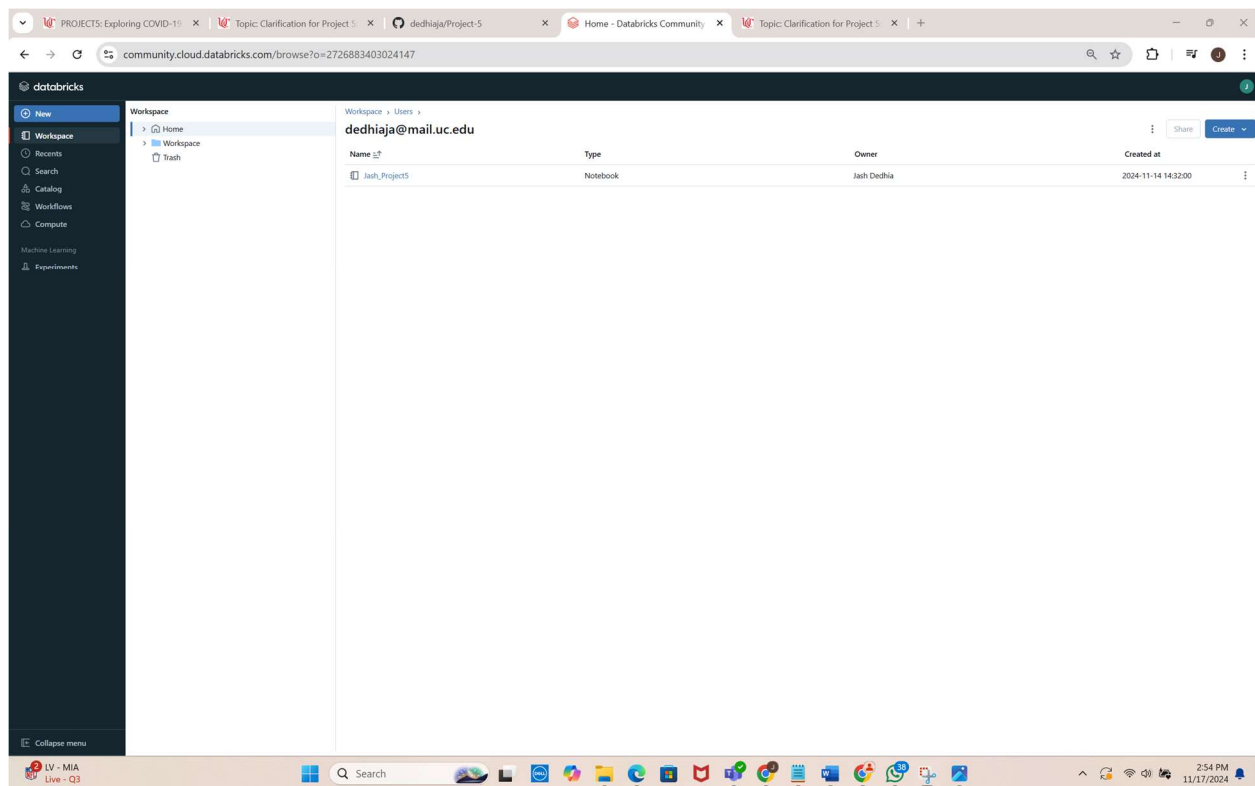
**FileStore Endpoint**  
Enable or disable FileStore endpoint /files. **On**  
FileStore files are accessible at /files. When enabled, names of the files stored in FileStore have to be considered public.  
[Learn more](#)



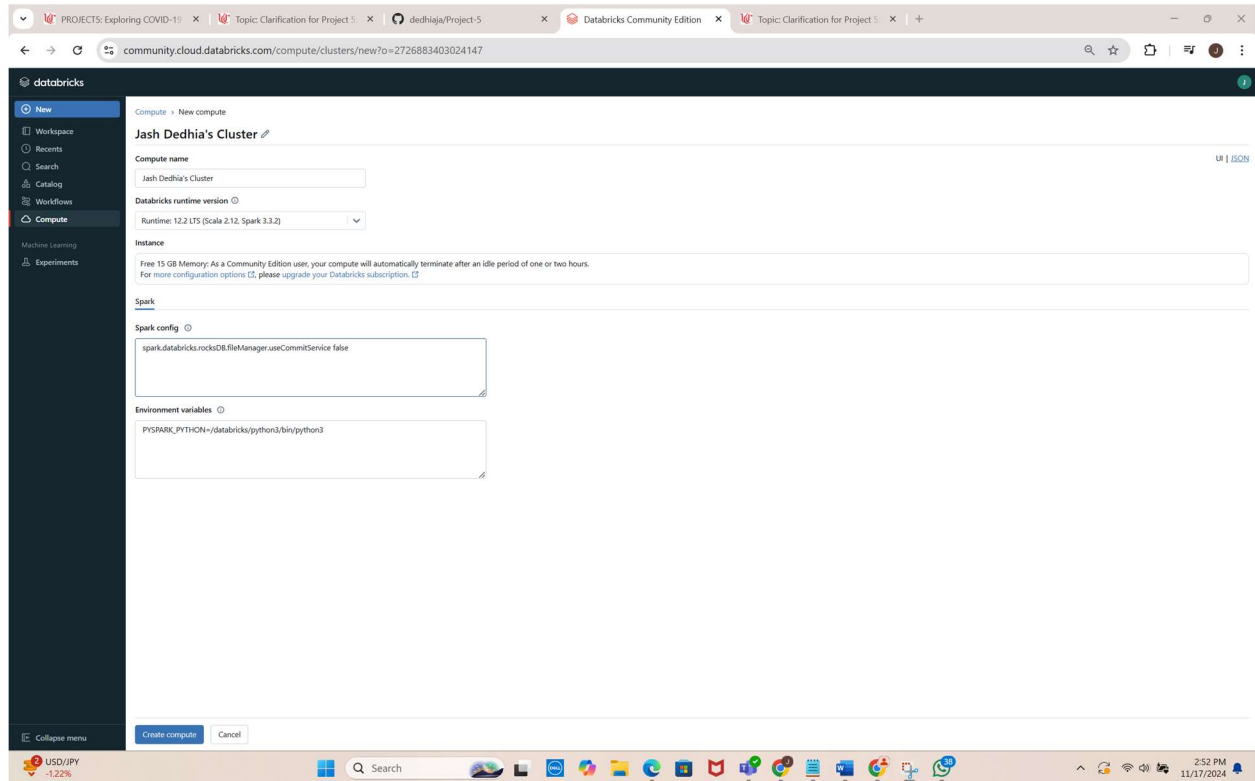
The screenshot shows the Databricks Catalog page. The left sidebar is the same as the previous screenshot. The main content area is titled 'Catalog' and shows a list of tables. The 'Database Tables' tab is selected, displaying a list of tables with columns for Name, Type, Owner, and Created at. The tables are listed in a table format.

**Catalog**

Name	Type	Owner	Created at
OWID_COVID19_data_4_Project5-1*	Notebook	Jash Dedhia	2024-11-14 14:32:00
OWID_COVID19_data_4_Project5-1*			
OWID_COVID19_data_4_Project5-2*			
OWID_COVID19_data_4_Project5-3*			
OWID_COVID19_data_4_Project5-4*			
OWID_COVID19_data_4_Project5-5*			
OWID_COVID19_data_4_Project5-6*			
OWID_COVID19_data_4_Project5-7*			
OWID_COVID19_data_4_Project5-8*			
OWID_COVID19_data_4_Project5-9*			
OWID_COVID19_data_4_Project5-10*			
OWID_COVID19_data_4_Project5-11*			



2.



This shows that the entire file is loaded:

The screenshot shows a Databricks workspace with a notebook named 'Jash\_Project5'. The notebook is running a SQL query that selects data from a table named 'OMID\_COVID19\_data\_4\_Project5\_csv'. The query result is displayed as a table with 10 columns: iso\_code, continent, location, date, total\_cases, new\_cases, new\_Scapes, total\_deaths, and new\_deaths. The table contains 10 rows of data for Afghanistan, spanning from 2020-03-07 to 2020-03-20. The status bar indicates that the result is stored as a \_sql\_ddf and can be used in other Python cells.

	iso_code	continent	location	date	total_cases	new_cases	new_Scapes	total_deaths	new_deaths
13	AFG	Asia	Afghanistan	2020-03-07	8	3	0.429	null	null
14	AFG	Asia	Afghanistan	2020-03-08	8	0	0.429	null	null
15	AFG	Asia	Afghanistan	2020-03-09	8	0	0.429	null	null
16	AFG	Asia	Afghanistan	2020-03-10	8	0	0.429	null	null
17	AFG	Asia	Afghanistan	2020-03-11	11	3	0.857	null	null
18	AFG	Asia	Afghanistan	2020-03-12	11	0	0.857	null	null
19	AFG	Asia	Afghanistan	2020-03-13	11	0	0.857	null	null
20	AFG	Asia	Afghanistan	2020-03-14	14	3	0.857	null	null
21	AFG	Asia	Afghanistan	2020-03-15	20	6	1.714	null	null
22	AFG	Asia	Afghanistan	2020-03-16	25	5	2.429	null	null
23	AFG	Asia	Afghanistan	2020-03-17	26	1	2.571	null	null
24	AFG	Asia	Afghanistan	2020-03-18	26	0	2.143	null	null
25	AFG	Asia	Afghanistan	2020-03-19	26	0	2.143	null	null
26	AFG	Asia	Afghanistan	2020-03-20	24	-2	1.857	null	null

This shows that the records are now just narrowed down for September and October 2021:

The screenshot shows a Databricks workspace with a notebook named 'Jash\_Project5'. The notebook is running a Python script that filters the data for September and October 2021. The script uses the following code:

```
from pyspark.sql import functions as F
from pyspark.sql.functions import to_date, month, year

# File location and type
file_location = "/FileStore/tables/OMID_COVID19_data_4_Project5_csv"
file_type = "csv"

# CSV options
infer_schema = "true"
first_row_is_header = "true"
delimiter = ","

# Load the CSV file into a DataFrame
df = spark.read.format(file_type) \
    .option("inferSchema", infer_schema) \
    .option("header", first_row_is_header) \
    .option("sep", delimiter) \
    .load(file_location)

# Convert the 'date' column to Datatype if necessary
df = df.withColumn("date", to_date("date", "yyyy-MM-dd"))

# Filter for September and October 2021, exclude null/empty continents, and ensure relevant columns have positive values
df_filtered = df.filter(
    (F.col('continent').isNotNull()) &
    (F.col('continent') != '') &
    (F.col('people_fully_vaccinated_phum') > 0) &
    (F.col('new_cases_phum') > 0) &
    (year(F.col('date')) == 2021) &
    (month(F.col('date')).isin([9, 10]))
)

# Display the Filtered DataFrame to confirm
display(df_filtered)
```

The script also includes a Spark job status bar indicating that the job is running and the data is being displayed.

	1.2 new_Sdeaths	1.2 total_cases_phum	1.2 new_cases_phum	1.2 new_Scapes_phum	1.2 total_deaths_phum	1.2 new_deaths_phum	1.2 new_Sdeaths_phum	1.2 reproduction_rate	1.2 iso_pater
1	2.143	3914.606	0.226	0.721	182.074	0.025	0.054	0.97	
2	2.714	3921.384	0.351	0.968	182.551	0.075	0.068	0.98	
3	2.371	51295.644	341.811	298.55	870.339	1.044	0.895	1.23	
4	2.714	51080.553	296.909	296.81	871.901	1.392	0.945	1.19	
5	2.429	52783.199	174.734	306.605	876.804	1.392	1.193	1.12	
6	1.571	52940.775	252.526	290.967	878.196	1.167	1.241	1.08	

3.

The screenshot shows a Databricks notebook titled "Jash\_Project5" in Python. The notebook is running on a cluster named "Jash Dedhia's Cluster". The first code cell (12:49 PM (3)) contains the following code:

```
# Group by continent and display the row counts for each continent
continent_counts = df_filtered.groupBy("continent").count()
continent_counts.show()
```

The output of this cell is a table with 6 rows, showing the count of records for each continent:

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

The second code cell (12:49 PM (3)) contains the following code:

```
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]))
)
```

The notebook interface includes a sidebar with navigation options like "New", "Workspace", "Recents", "Search", "Catalog", "Workflows", "Compute", "Machine Learning", and "Experiments". The bottom status bar shows the time as 12:55 PM on 11/17/2024.

4.

The screenshot shows the same Databricks notebook "Jash\_Project5" with additional code cells. The first code cell (12:49 PM (3)) is the same as in the previous screenshot, showing the continent counts table.

The second code cell (12:49 PM (3)) contains the same code as in the previous screenshot, creating month and year columns and filtering for September and October 2021.

The third code cell (12:49 PM (3)) contains the following code:

```
# 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 of this cell is a text message: "Total record count for September and October 2021: 5213".

The fourth code cell (12:49 PM (3)) contains the following code:

```
# Calculate the average values for the selected metrics by continent and month
avg_df = df_filtered.groupBy("continent", "month").agg(
    F.mean("people_fully_vaccinated_phun").alias("average_people_fully_vaccinated"),
    F.mean("new_cases_pm1").alias("average_new_cases"),
    F.mean("excess_mortality").alias("average_excess_mortality")
)
avg_df.show()
```

The notebook interface is consistent with the previous screenshot, showing the same sidebar and status bar.

5.

The screenshot shows a Databricks workspace with a notebook titled "Jash\_Project5". The notebook contains the following Python code:

```
df_filtered = df.filter(F.col('continent').isNotNull()) \
    .withColumn("year", year("date")) \
    .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_phun').alias('average_new_cases'),
    F.mean('excess_mortality').alias('average_excess_mortality')
).orderBy('continent', 'month')

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

Below the code, the output of the `show()` command is displayed as a table with 12 rows and 5 columns:

continent	month	average_people_fully_vaccinated	average_new_cases	average_excess_mortality
Africa	9	9.050997867448602	18.56174506172839	25.88
Africa	10	10.979482173913043	18.04095400238949	9.0975
Asia	9	35.818799999999996	137.81524184397162	45.19416666666667
Asia	10	44.35308982035929	113.52837542896364	46.093333333333334
Europe	9	53.51533724340176	202.05234275362315	12.174639175257733
Europe	10	56.79475073313782	289.7618099579243	12.14
North America	9	40.22337078651684	352.61750579710144	74.689
North America	10	45.035721970994174	217.01079913043487	null
Oceania	9	31.851639434703514	122.51806363636363	-6.7
Oceania	10	47.44008163265306	15.839052325581394	-13.0
South America	9	43.167231404958684	108.70364066852369	9.0475
South America	10	51.30851528384279	74.9003790322581	7.786666666666668

6.

The screenshot shows the same Databricks workspace, but the notebook now displays a bar chart visualization. The chart is titled "Bar Chart" and shows the "Average New Cases, Average People Fully Vaccinated" for September and October across five continents: Africa, Asia, Europe, North America, and South America. The legend indicates four data series: October Average New Cases (blue), October Average People Fully Vaccinated (red), September Average New Cases (green), and September Average People Fully Vaccinated (purple).

The chart shows that Europe has the highest average new cases in both months, while North America has the highest average people fully vaccinated. The chart is refreshed 58 minutes ago.

New

Workspace

Recents

Search

Catalog

Workflows

Compute

Machine Learning

Experiments

Jash\_Project5Python

FileEditViewRunHelpLast edit was now

Run allJash Dedhia's ClusterSharePublish

Continents

DownloadEdit Visualization12 rowsRefreshed 1 hour ago

Just now (3s)11

# Perform correlation analysis  
correlation = avg\_df.stat.corr('average\_people\_fully\_vaccinated', 'average\_new\_cases')  
  
# Print the correlation coefficient  
print(f"Correlation between average vaccination rate and new cases: {correlation}")  
print(f"The correlation coefficient of {correlation:.4f} indicates a moderate positive relationship between vaccination rates and new cases. This suggests that other factors, such as testing rates or high prior infection rates in highly vaccinated regions, might be influencing this unexpected trend.")  
  
(5) Spark Jobs  
Correlation between average vaccination rate and new cases: 0.5195322019056203  
The correlation coefficient of 0.5195 indicates a moderate positive relationship between vaccination rates and new cases. This suggests that other factors, such as testing rates or high prior infection rates in highly vaccinated regions, might be influencing this unexpected trend.

12:49 PM (2d)12Python

# Define missing GDP data as a dictionary  
gdp\_data = {  
 "CUB": 11295.00,  
 "LIE": 197585.00,  
 "MCO": 248062.00,  
 "SOM": 1611.30,  
 "THU": 67455.00,  
 "SYR": 2914.50  
}  
  
# Create a DataFrame from the dictionary  
from pyspark.sql.types import StructType, StructField, StringType, FloatType  
  
schema = StructType([  
 StructField("iso code", StringType(), True),

[illegible]



9.

The screenshot shows a Databricks notebook interface with the following content:

```

from pyspark.sql import functions as F
from pyspark.sql.functions import year, month

# Step 1: Filter the Data
# Ensure data is limited to September and October 2021, and rows with valid continent values
filtered_data = df_with_gdp.filter(
    (F.col("continent").isNotNull()) &
    (F.col("year") == 2021) &
    (F.col("month").isin([9, 10]))
)

# Step 2: Calculate descriptive statistics
summary_stats = filtered_data.agg(
    F.mean("people_fully_vaccinated_phum").alias("mean_vaccinated"),
    F.stddev("people_fully_vaccinated_phum").alias("stddev_vaccinated"),
    F.count("people_fully_vaccinated_phum").alias("count_vaccinated"),
    F.mean("new_cases_gmll").alias("mean_new_cases"),
    F.stddev("new_cases_gmll").alias("stddev_new_cases"),
    F.count("new_cases_gmll").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)

```

The output shows a table with 8 columns: 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, 1.2 mean\_excess\_mortality, and 1.2 stddev\_excess\_mortality. The table contains 1 row of data.

10.

The screenshot shows a Databricks notebook interface with the following content:

```

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_mortality:.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()

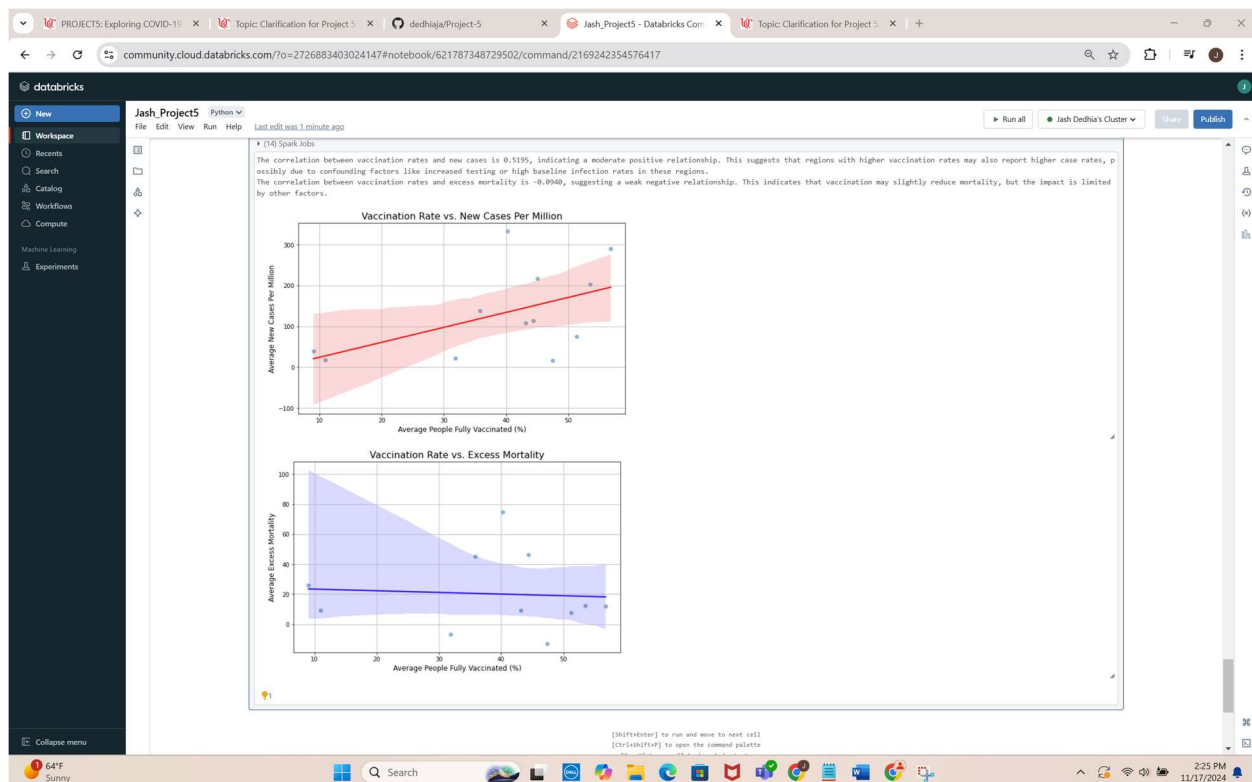
# Scatter Plot: Vaccination Rate vs. New Cases Per Million
plt.figure(figsize=(10, 6))
sns.regplot(
    data=avg_df_pandas,
    x="average_people_fully_vaccinated",
    y="average_new_cases",
    line_kws={"color": "red"},
    scatter_kws={"alpha": 0.5}
)
plt.title("Vaccination Rate vs. New Cases Per Million", fontsize=16)
plt.xlabel("Average People fully Vaccinated (M)", 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 (M)", fontsize=12)
plt.ylabel("Average Excess Mortality", fontsize=12)
plt.grid(True)
plt.show()

```

The output shows a text-based correlation result: "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 analysis reveals a **moderate positive correlation (0.5195)** between vaccination rates and new COVID-19 cases per million. This suggests that regions with higher vaccination rates may also report more cases during September and October 2021. However, this counterintuitive trend can be attributed to external factors, such as increased testing rates, higher population densities, or previously high baseline infection levels in regions with strong vaccination campaigns. Additionally, vaccination may encourage behavioral changes like reduced mask usage or social distancing, particularly in regions where people perceive themselves as protected. These factors likely confound the direct relationship between vaccination rates and case counts.

In contrast, the **weak negative correlation (-0.0940)** between vaccination rates and excess mortality aligns with expectations that vaccines reduce severe outcomes and deaths. While the effect appears small, it indicates that vaccination may help mitigate mortality, even if the direct impact on case numbers is less clear. Factors such as healthcare quality, demographic differences, and variant severity likely also influence mortality rates, underscoring the importance of comprehensive pandemic responses beyond vaccination campaigns alone. These findings suggest that vaccination plays a critical role in reducing the severity of COVID-19 but must be complemented by other measures to manage overall case rates.