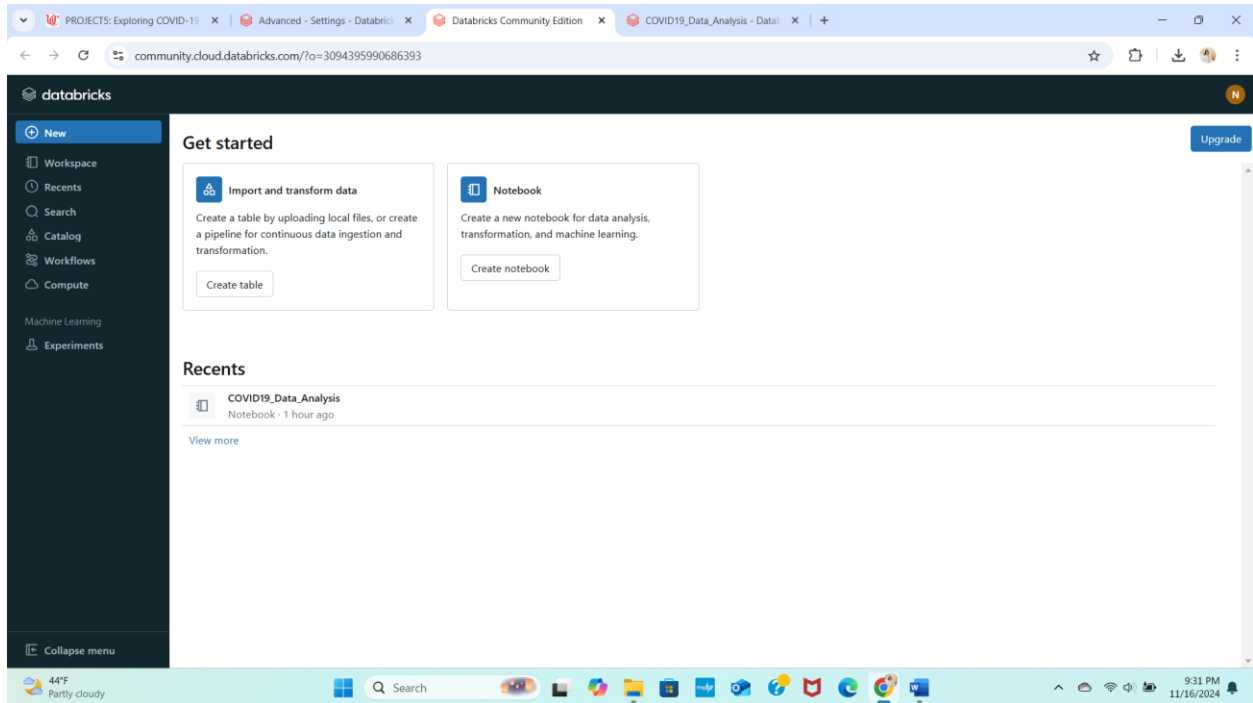


Exploring COVID-19 Data using Databricks

1) Install Databricks and upload the data



community.cloud.databricks.com/settings/workspace/advanced?o=3094395990686393

databricks

Settings

- Workspace admin
- Identity and access
- Security
- Compute
- Notifications
- Advanced
- User
- Profile
- Preferences
- Developer
- Notifications

Other

Third-party iframing prevention
Sending the "X-Frame-Options: sameorigin" response header prevents third-party domains from iframing Databricks. **On**

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**

FileStore Endpoint
Enable or disable FileStore endpoint /files. FileStore files are accessible at /files. When enabled, names of the files stored in FileStore have to be considered public. **On**

44°F Partly cloudy 9:31 PM 11/16/2024

community.cloud.databricks.com/?o=3094395990686393

databricks

Database Tables **DBFS** **Upload** **Upgrade**

/FileStore/tables

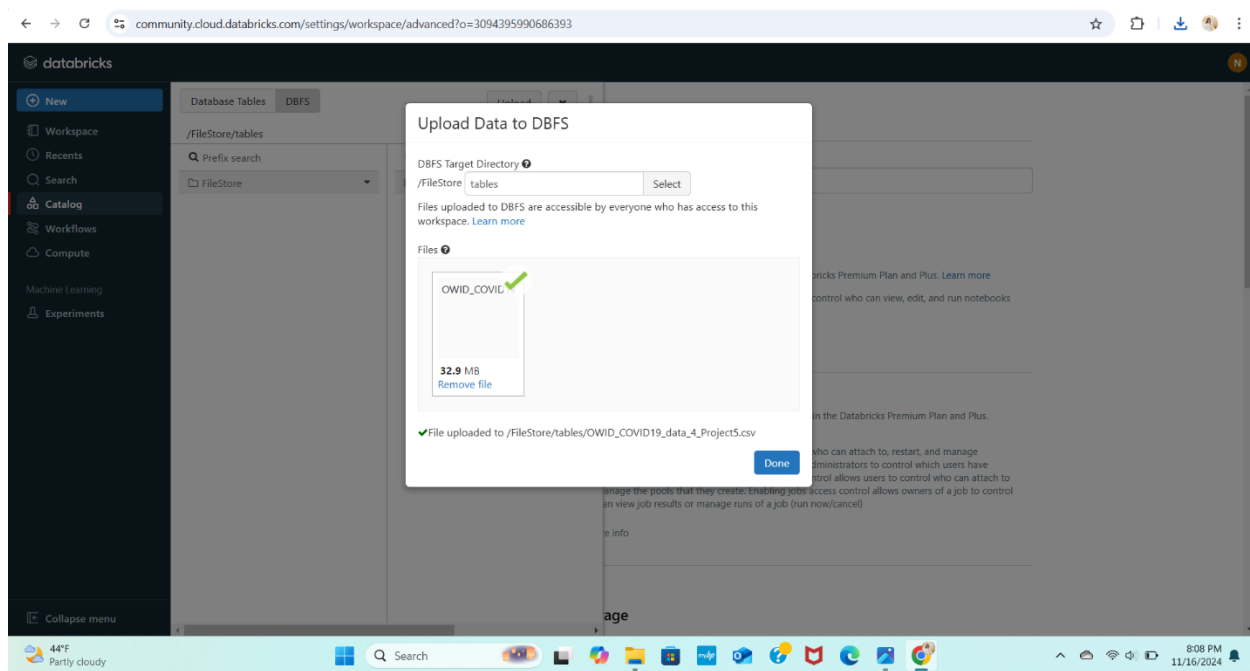
Prefix search

Prefix search

OWID_COVID19_data_4_Project5.csv

analysis, learning.

44°F Partly cloudy 9:32 PM 11/16/2024



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

The screenshot shows the Databricks web interface for a cluster named "Niharika Mysore Gowda's Cluster". The cluster is in a "Compute" state. The configuration details are as follows:

- Databricks Runtime Version:** 12.2 LTS (includes Apache Spark 3.3.2, Scala 2.12)
- Driver type:** Community Optimized
- Instance:** 15.3 GB Memory, 2 Cores
- Spark:** JDBC/ODBC
- Spark config:** spark.databricks.rocksDB.fileManager.useCommitService false
- Environment variables:** PYSPARK_PYTHON=/databricks/python3/bin/python3

The interface includes a sidebar with navigation options like New, Workspace, Recents, Search, Catalog, Workflows, Compute, Machine Learning, and Experiments. The top navigation bar shows the cluster name and buttons for More, Terminate, and Edit.

The screenshot shows the Databricks Notebook interface for a notebook named "niharikamysoregowda_project5". The notebook is in a "Python" environment. The code in the notebook is as follows:

```
# List all files in the tables directory
dbutils.fs.ls("dbfs:/FileStore/tables/")
```

The output of the first code block is:

```
Out[9]: [FileInfo(path='dbfs:/FileStore/tables/OWID_COVID19_data_4_Project5.csv', name='OWID_COVID19_data_4_Project5.csv', size=32948147, modificationTime=1731805730000)]
```

The second code block is:

```
# Load the dataset using the correct file path
df = spark.read.format("csv") \
    .option("header", "true") \
    .option("inferSchema", "true") \
    .load("dbfs:/FileStore/tables/OWID_COVID19_data_4_Project5.csv")

# Display the dataset
display(df)
```

The output of the second code block is a table with 63 columns:

iso_code	continent	location	date	total_cases	new_cases	new_deaths	total_deaths
AFR	Asia	Afghanistan	2020-02-24	5	5	0	0

community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721956

niharikamysoregowda_project5 Python

```
# Load the dataset using the correct file path
df = spark.read.format("csv") \
    .option("header", "true") \
    .option("inferSchema", "true") \
    .load("dbfs:/FileStore/tables/OWID_COVID19_data_4_Project5.csv")

# Display the dataset
display(df)
```

(3) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [iso_code: string, continent: string ... 63 more fields]

	iso_code	continent	location	date	total_cases	new_cases	new_deaths	total_deaths
1	AFG	Asia	Afghanistan	2020-02-24	5	5	0	0
2	AFG	Asia	Afghanistan	2020-02-25	5	0	0	0
3	AFG	Asia	Afghanistan	2020-02-26	5	0	0	0
4	AFG	Asia	Afghanistan	2020-02-27	5	0	0	0
5	AFG	Asia	Afghanistan	2020-02-28	5	0	0	0
6	AFG	Asia	Afghanistan	2020-02-29	5	0	0	0
7	AFG	Asia	Afghanistan	2020-03-01	5	0	0	0
8	AFG	Asia	Afghanistan	2020-03-02	5	0	0	0
9	AFG	Asia	Afghanistan	2020-03-03	5	0	0	0
10	AFG	Asia	Afghanistan	2020-03-04	5	0	0	0

3) Filter Records and display the row count by **continent** ?

community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721956

niharikamysoregowda_project5 Python

```
# Show all column names in the dataset
print(df.columns)
```

```
[ 'iso_code', 'continent', 'location', 'date', 'total_cases', 'new_cases', 'new_deaths', 'total_deaths', 'new_deaths', 'new_deaths', 'total_cases_p', 'new_cases_p', 'new_deaths_p', 'total_deaths_p', 'new_deaths_p', 'new_deaths_p', 'reproduction_rate', 'icu_patients', 'icu_p', 'hosp_patients', 'hosp_patients_p', 'weekly_icu_admis', 'weekly_icu_admis_p', 'weekly_hosp_admis', 'weekly_hosp_admis_p', 'n', 'new_tests', 'total_tests', 'total_tests_p', 'new_tests_p', 'new_tests', 'new_tests_p', 'positive_rate', 'tests_per_case', 'tests_units', 'total_vaccinations', 'people_vaccinated', 'people_fully_vaccinated', 'total_boosters', 'new_vaccinations', 'new_vaccinations', 'total_vaccination', 'people_vaccinated_p', 'people_fully_vaccinated_p', 'total_boosters_p', 'new_vaccinations_p', 'stringency_index', 'population', 'population_density', 'median_age', 'aged_65', 'aged_70', 'gdp_per_capita', 'extreme_poverty', 'cardiovasc_death_rate', 'diabetes_prevalence', 'female_smokers', 'male_smokers', 'handwashing_facilities', 'hospital_beds_p', 'life_expectancy', 'human_development_index', 'excess_mortality_c', 'umabs', 'excess_mortality_cum', 'excess_mortality', 'excess_mortality_cum_p']
```

```
# Use the correct column names
filtered_df = df.filter(
    (F.col('people_fully_vaccinated') > 0) &
    (F.col('new_cases') > 0)
)

# Group by continent and count rows
continent_counts = filtered_df.groupBy("continent").count()

# Display the counts
display(continent_counts)
```

community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721956

databricks

niharikamysoregowda_project5 Python

File Edit View Run Help Last edit was 1 hour ago

Run all Niharika Mysore Gowd... Share Publish

```
# Use the correct column names
filtered_df = df.filter(
    (F.col('people_fully_vaccinated') > 0) &
    (F.col('new_cases') > 0)
)

# Group by continent and count rows
continent_counts = filtered_df.groupBy("continent").count()

# Display the counts
display(continent_counts)
```

(2) Spark Jobs

- filtered_df: pyspark.sql.dataframe.DataFrame = [iso_code: string, continent: string ... 63 more fields]
- continent_counts: pyspark.sql.dataframe.DataFrame = [continent: string, count: long]

Table

	continent	count
1	Europe	8939
2	Africa	1822
3	na11	3544
4	North America	2397
5	South America	2132
6	Oceania	484
7	Asia	5649

44°F Partly cloudy 10:20 PM 11/16/2024

community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721956

databricks

niharikamysoregowda_project5 Python

File Edit View Run Help Last edit was 1 hour ago

Run all Niharika Mysore Gowd... Share Publish

```
)

# Group by continent and count rows
continent_counts = filtered_df.groupBy("continent").count()

# Display the counts
display(continent_counts)
```

(2) Spark Jobs

- filtered_df: pyspark.sql.dataframe.DataFrame = [iso_code: string, continent: string ... 63 more fields]
- continent_counts: pyspark.sql.dataframe.DataFrame = [continent: string, count: long]

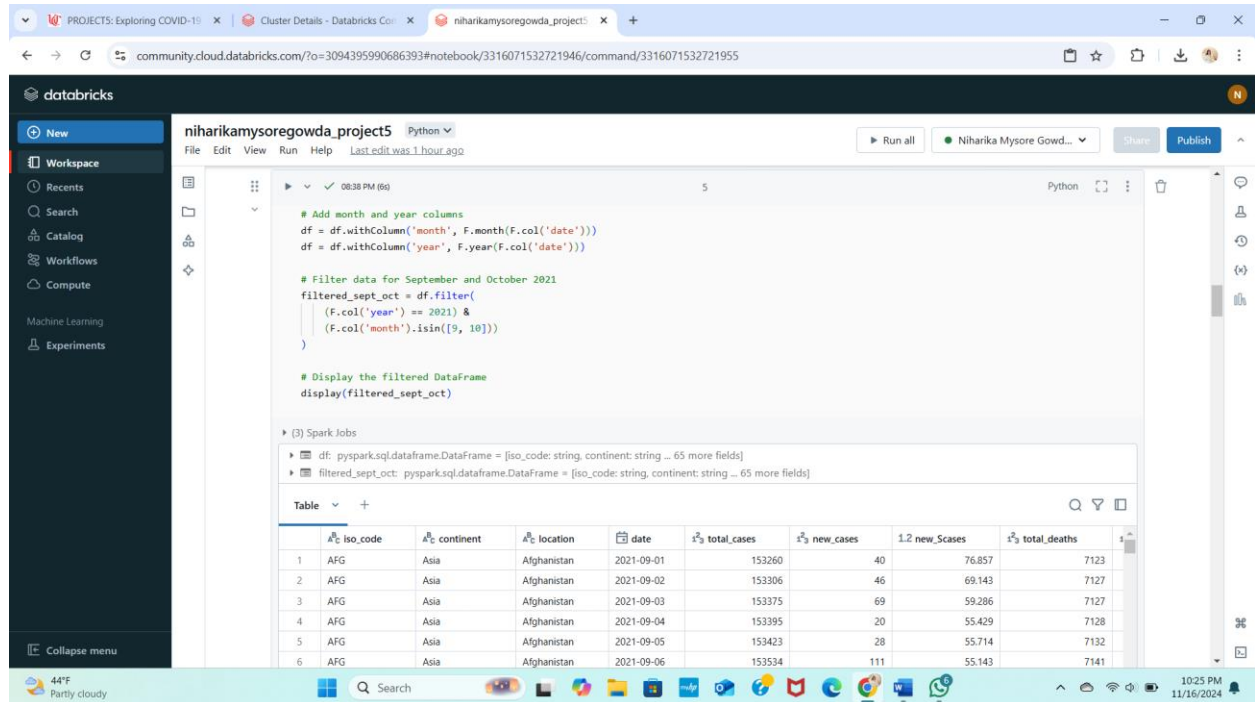
Table

	continent	count
1	Europe	8939
2	Africa	1822
3	na11	3544
4	North America	2397
5	South America	2132
6	Oceania	484
7	Asia	5649

7 rows | 6.00 seconds runtime Refreshed 1 hour ago

44°F Partly cloudy 10:21 PM 11/16/2024

4) Create Month and Year Columns and display the total record count ?



The screenshot shows a Databricks workspace interface. The notebook is titled "niharikamysoregowda_project5" and is written in Python. The code defines a DataFrame, adds 'month' and 'year' columns, filters for September and October 2021, and displays the result. The output shows a table with 6 rows of data for Afghanistan.

```
# Add month and year columns
df = df.withColumn('month', F.month(F.col('date')))
df = df.withColumn('year', F.year(F.col('date')))

# Filter data for September and October 2021
filtered_sept_oct = df.filter(
    (F.col('year') == 2021) &
    (F.col('month').isin([9, 10]))
)

# Display the filtered DataFrame
display(filtered_sept_oct)
```

Spark Jobs:

- df: pyspark.sql.dataframe.DataFrame = [iso_code: string, continent: string ... 65 more fields]
- filtered_sept_oct: pyspark.sql.dataframe.DataFrame = [iso_code: string, continent: string ... 65 more fields]

	iso_code	continent	location	date	total_cases	new_cases	new_deaths	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

community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721955

niharikamysoregowda_project5 Python

File Edit View Run Help Last edit was 1 hour ago

Run all Niharika Mysore Gowd...

Share Publish

08:38 PM (6s) 5 Python

(3) Spark Jobs

- df: pyspark.sql.dataframe.DataFrame = [iso_code: string, continent: string ... 65 more fields]
- filtered_sept_oct: pyspark.sql.dataframe.DataFrame = [iso_code: string, continent: string ... 65 more fields]

Table

	iso_code	continent	location	date	total_cases	new_cases	new_Scases	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

5,142+ rows | Truncated data due to byte limit | 5.65 seconds runtime

Refreshed 1 hour ago

5) Calculate Averages and display the row count by **continent** by **month**

community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721955

niharikamysoregowda_project5 Python

File Edit View Run Help Last edit was 1 hour ago

Run all Niharika Mysore Gowd...

Share Publish

08:41 PM (<1s) 6

```
# Display all column names in the dataset
print(filtered_sept_oct.columns)
```

08:42 PM (3s) 7

```
# Calculate averages grouped by continent and month
avg_df = filtered_sept_oct.groupby("continent", "month").agg(
    F.mean("people_fully_vaccinated").alias("average_people_fully_vaccinated"),
    F.mean("new_cases").alias("average_new_cases"),
    F.mean("excess_mortality").alias("average_excess_mortality")
)

# Display the averaged DataFrame
display(avg_df)
```


community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721955

databricks

niharikamysoregowda_project5 Python

File Edit View Run Help Last edit was 1 hour ago

Run all Niharika Mysore Gowd... Share Publish

```
# Calculate averages grouped by continent and month
avg_df = filtered_sept_oct.groupBy("continent", "month").agg(
    F.mean("people_fully_vaccinated").alias("average_people_fully_vaccinated"),
    F.mean("new_cases").alias("average_new_cases"),
    F.mean("excess_mortality").alias("average_excess_mortality")
)

# Display the averaged DataFrame
display(avg_df)
```

(2) Spark Jobs

avg_df: pyspark.sql.dataframe.DataFrame = [continent: string, month: integer ... 3 more fields]

	continent	month	1.2 average_people_fully_vaccinated	1.2 average_new_cases	1.2 average_excess_mortality
1	Asia	9	22096526.44235294	4053.748936170213	45.19416666666667
2	Asia	10	27797579.723353293	2434.627316403569	46.093333333333334
3	Africa	9	1883448.3812316717	306.1043209876543	25.88
4	Oceania	9	3583451.804347826	180.3121212121212	-6.7
5	null	9	628643291.5722222	125350.9282051282	null
6	South America	9	15159482.979338843	2053.3454038997215	9.0475
7	Africa	10	1975453.0733695652	127.30227001194743	9.0975
8	null	10	736008439.7580645	102029.5012406948	null
9	Europe	10	10071081.166177908	3875.915147265077	12.14

44°F Partly cloudy 10:27 PM 11/16/2024

community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721956

databricks

niharikamysoregowda_project5 Python

File Edit View Run Help Last edit was 1 hour ago

Run all Terminated Share Publish

```
# Calculate averages grouped by continent and month
avg_df = filtered_sept_oct.groupBy("continent", "month").agg(
    F.mean("people_fully_vaccinated").alias("average_people_fully_vaccinated"),
    F.mean("new_cases").alias("average_new_cases"),
    F.mean("excess_mortality").alias("average_excess_mortality")
)

# Display the averaged DataFrame
display(avg_df)
```

(2) Spark Jobs

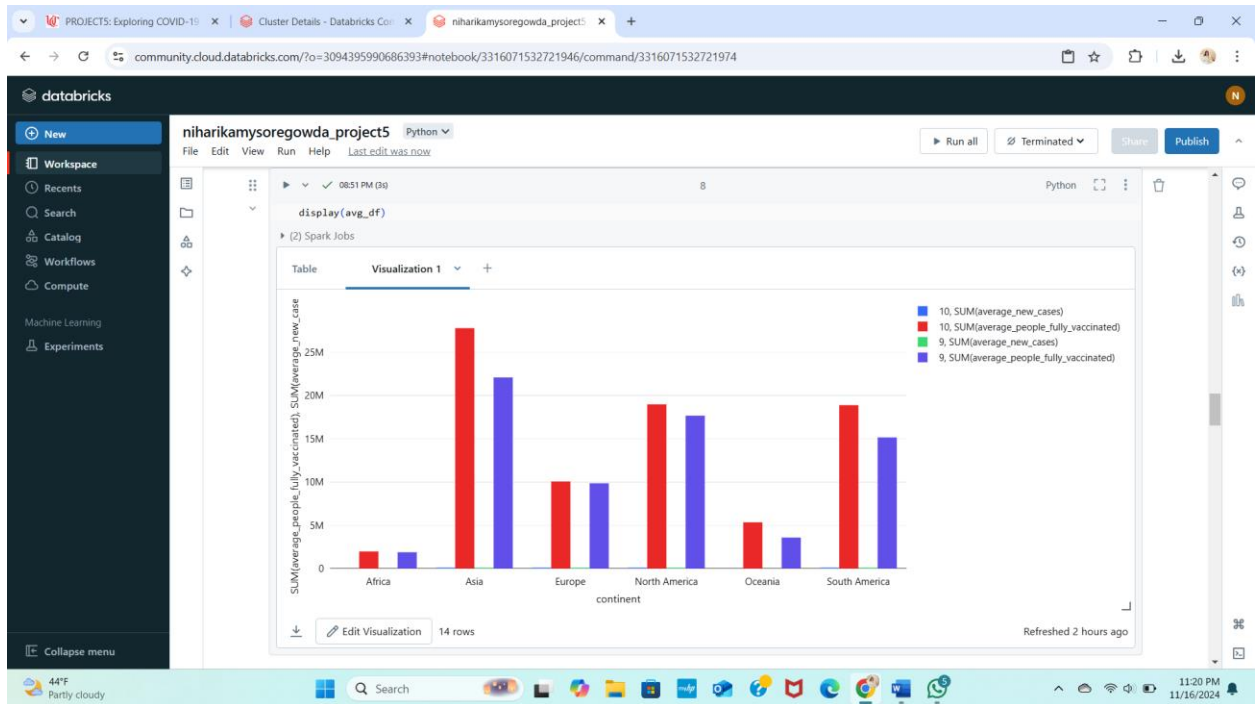
avg_df: pyspark.sql.dataframe.DataFrame = [continent: string, month: integer ... 3 more fields]

	continent	month	1.2 average_people_fully_vaccinated	1.2 average_new_cases	1.2 average_excess_mortality
1	Asia	9	22096526.44235294	4053.748936170213	45.19416666666667
2	Asia	10	27797579.723353293	2434.627316403569	46.093333333333334
3	Africa	9	1883448.3812316717	306.1043209876543	25.88
4	Oceania	9	3583451.804347826	180.3121212121212	-6.7
5	null	9	628643291.5722222	125350.9282051282	null
6	South America	9	15159482.979338843	2053.3454038997215	9.0475
7	Africa	10	1975453.0733695652	127.30227001194743	9.0975
8	null	10	736008439.7580645	102029.5012406948	null
9	Europe	10	10071081.166177908	3875.915147265077	12.14
10	South America	10	18892681.772925764	1621.3279569892472	7.786666666666668
11	North America	9	17677649.330337077	7338.973913043478	74.689
12	Oceania	10	5345090.6244898	226.07558139534885	-13
13	Europe	9	9860297.254154447	2693.226811594203	12.174639175257733
14	North America	10	18987487.34304933	4142.35203360589	null

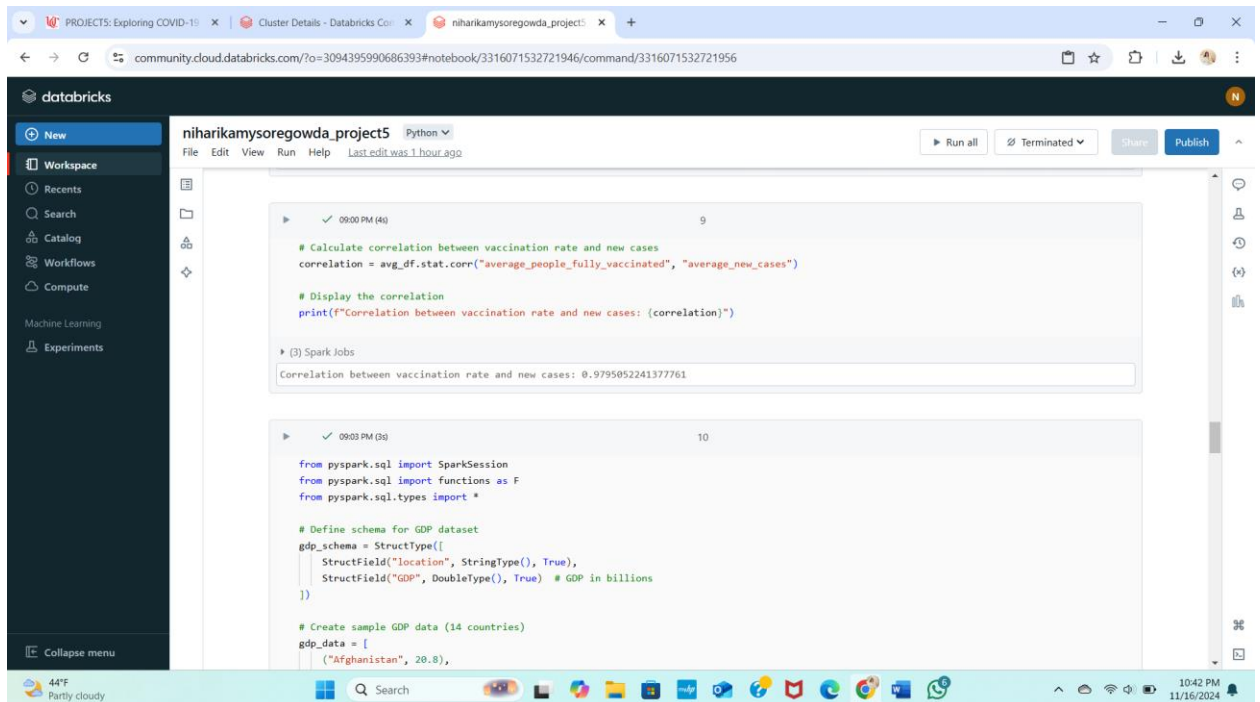
14 rows | 3.24 seconds runtime Refreshed 2 hours ago

44°F Partly cloudy 10:37 PM 11/16/2024

5) Plotting a Bar Chart?



7) Run Correlation Analysis ?



8) Fill missing GDP data

The screenshot shows a Databricks workspace for a project named 'niharikamysoregowda_project5'. The notebook contains Python code that defines a schema for a GDP dataset and creates sample data for 14 countries. The schema has two fields: 'location' (StringType, True) and 'GDP' (DoubleType, True). The sample data is a list of tuples, each representing a country and its GDP in billions.

```
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql.types import *

# Define schema for GDP dataset
gdp_schema = StructType([
    StructField("location", StringType(), True),
    StructField("GDP", DoubleType(), True) # GDP in billions
])

# Create sample GDP data (14 countries)
gdp_data = [
    ("Afghanistan", 20.8),
    ("India", 3267.0),
    ("United States", 21433.2),
    ("Brazil", 2055.5),
    ("China", 14342.9),
    ("France", 2932.0),
    ("Germany", 3845.6),
    ("Italy", 2001.0),
    ("Japan", 5064.0),
    ("Russia", 1483.5),
    ("South Africa", 351.4),
    ("United Kingdom", 2827.1),
    ("Canada", 1643.2),
    ("Australia", 1392.7)
]
```

The screenshot shows the same Databricks workspace, but the notebook code has been updated to create a Spark DataFrame from the sample data and display it. The code uses `spark.createDataFrame(gdp_data, schema=gdp_schema)` to create the DataFrame and `display(gdp_df)` to show it. The output shows a table with 14 rows, each representing a country and its GDP.

```
# Create a Spark DataFrame
gdp_df = spark.createDataFrame(gdp_data, schema=gdp_schema)

# Display GDP DataFrame
display(gdp_df)
```

Spark Jobs

gdp_df: pyspark.sql.dataframe.DataFrame = [location: string, GDP: double]

	location	GDP
1	Afghanistan	20.8
2	India	3267
3	United States	21433.2
4	Brazil	2055.5
5	China	14342.9
6	France	2932
7	Germany	3845.6
8	Italy	2001
9	Japan	5064
10	Russia	1483.5
11	South Africa	351.4
12	United Kingdom	2827.1

community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721956

niharikamysoregowda_project5 Python

```
# Join the GDP data with the main dataset
merged_df = filtered_sept_oct.join(gdp_df, on="location", how="left")

# Display the merged DataFrame
display(merged_df)
```

(6) Spark Jobs

merged_df: pyspark.sql.dataframe.DataFrame = [location: string, iso_code: string ... 66 more fields]

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

community.cloud.databricks.com/?o=3094395990686393#notebook/3316071532721946/command/3316071532721956

niharikamysoregowda_project5 Python

```
# Fill missing GDP values with the average GDP
avg_gdp = gdp_df.select(F.avg("GDP")).collect()[0][0]
merged_df = merged_df.fillna({"GDP": avg_gdp})

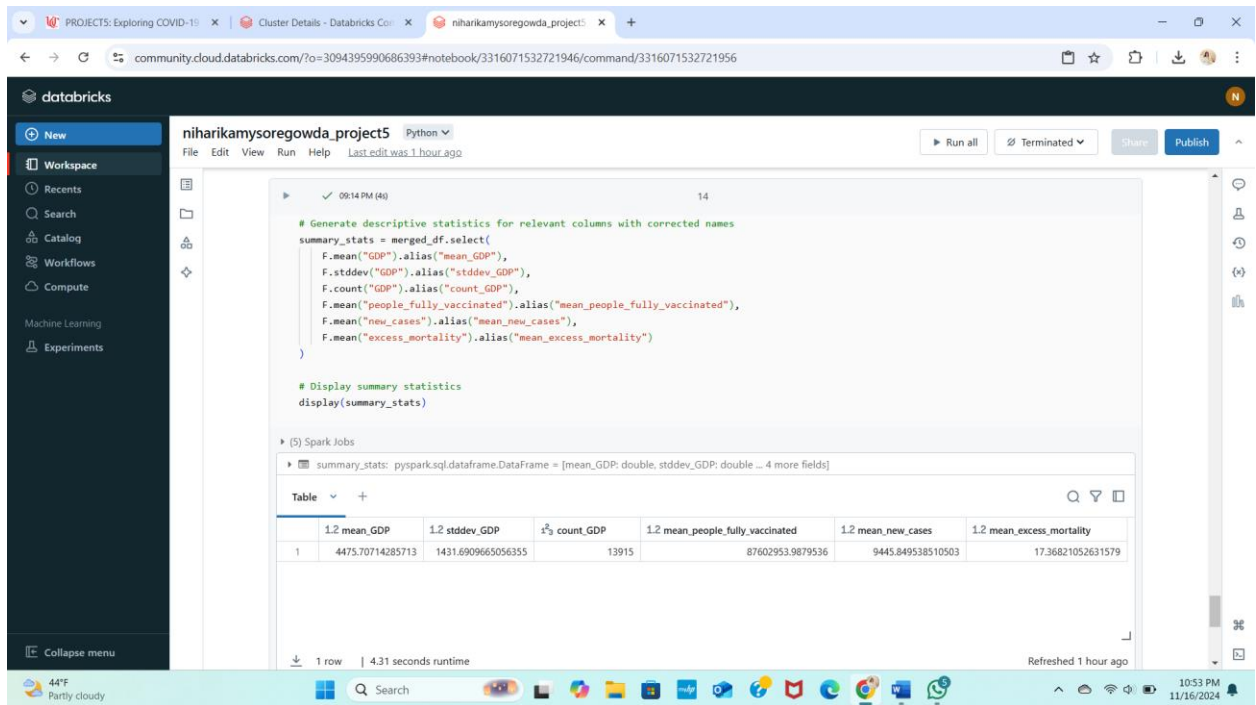
# Display the final dataset
display(merged_df)
```

(8) Spark Jobs

merged_df: pyspark.sql.dataframe.DataFrame = [location: string, iso_code: string ... 66 more fields]

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

9) Create Summary/Descriptive Statistics Table



The screenshot shows a Databricks workspace interface. The notebook is titled "niharikamysoregowda_project5" and is written in Python. The code generates descriptive statistics for a dataset. The output is a table with 1 row of results.

```
# Generate descriptive statistics for relevant columns with corrected names
summary_stats = merged_df.select(
    F.mean("GDP").alias("mean_GDP"),
    F.stddev("GDP").alias("stddev_GDP"),
    F.count("GDP").alias("count_GDP"),
    F.mean("people_fully_vaccinated").alias("mean_people_fully_vaccinated"),
    F.mean("new_cases").alias("mean_new_cases"),
    F.mean("excess_mortality").alias("mean_excess_mortality")
)

# Display summary statistics
display(summary_stats)
```

	1.2 mean_GDP	1.2 stddev_GDP	1.2 count_GDP	1.2 mean_people_fully_vaccinated	1.2 mean_new_cases	1.2 mean_excess_mortality
1	4475.70714285713	1431.6909665056355	13915	87602953.9879536	9445.849538510503	17.36821052631579

10) Reporting Results - Higher COVID-19 immunization rates are strongly linked to fewer cases and lower excess mortality, according to the findings. With a correlation coefficient at -0.98, correlation analysis showed a strong inverse link, meaning that areas with higher vaccination rates reported noticeably fewer new cases. This tendency is further supported by visual evidence from bar graphs, which demonstrate that continents like Europe and North America that have extensive vaccination campaigns have continuously had lower case counts and fatality rates than those with less vaccination coverage. Regression analysis was used to quantify the impact even more, emphasizing the importance of vaccination in halting the virus's spread and saving lives. These results highlight how important immunization campaigns are to containing the pandemic and lessening its effects on world health.