

Name : Piyush Jangade

PRN : 202401110040

Roll No : ES7-45

Div : ES7

Batch : ES72

# This is the dataset that i have imported from the kaggle



```
import kagglehub  
|  
# Download latest version  
path = kagglehub.dataset_download("imdevskp/corona-virus-report")  
  
print("Path to dataset files:", path)
```

## Covid 19 Dataset

**this is my Google Colab link :**

<https://colab.research.google.com/drive/1qGvoNixwBb8WVgBaF0A83B3xogwNKPVe?usp=sharing>

## 1. countries having 0 new deaths but past deaths are more than 5000



```
import pandas as pd

# Load the dataset
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')

# Filter countries with 0 new deaths but total deaths > 5000
result = df[
    (df['New deaths'] == 0) &
    (df['Deaths'] > 5000)
][['Country/Region', 'Deaths', 'New deaths']]

# Sort by highest total deaths
result = result.sort_values('Deaths', ascending=False)

print("Countries with 0 new deaths but total deaths > 5,000:")
print(result)
```



Countries with 0 new deaths but total deaths > 5,000:

	Country/Region	Deaths	New deaths
157	Spain	28432	0

## 2. Calculate the mean, median, and standard deviation of 'Active' cases.



```
import pandas as pd
country_wise_df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
active_mean = country_wise_df['Active'].mean()
active_median = country_wise_df['Active'].median()
active_std = country_wise_df['Active'].std()
print(f"Mean of Active cases: {active_mean}")
print(f"Median of Active cases: {active_median}")
print(f"Standard Deviation of Active cases: {active_std}")
```



```
Mean of Active cases: 34001.935828877
Median of Active cases: 1600.0
Standard Deviation of Active cases: 213326.1733714289
```

### 3. List all countries that have a "Deaths / 100 Cases" ratio greater than 5%

```
[ ] import pandas as pd
country_wise_df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
high_death_rate_countries = country_wise_df[country_wise_df['Deaths / 100 Cases'] > 5]
print(high_death_rate_countries[['Country/Region', 'Deaths / 100 Cases']])
```

	Country/Region	Deaths / 100 Cases
3	Andorra	5.73
14	Barbados	6.36
16	Belgium	14.79
32	Canada	7.68
34	Chad	8.13
36	China	5.37
51	Ecuador	6.82
52	Egypt	5.03
61	France	13.71
73	Guyana	5.14
77	Hungary	13.40

#### 4. Create a NumPy array of "New cases" and find the maximum and minimum value.

```
import pandas as pd
import numpy as np
country_wise_df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
new_cases_array = country_wise_df['New cases'].to_numpy()
max_new_cases = np.max(new_cases_array)
min_new_cases = np.min(new_cases_array)
print(f"Maximum new cases reported by a country: {max_new_cases}")
print(f"Minimum new cases reported by a country: {min_new_cases}")
```

```
➞ Maximum new cases reported by a country: 56336
Minimum new cases reported by a country: 0
```

#### 5. Find the top 5 countries with the highest recovery rate (Recovered / 100 Cases).

```
[ ] import pandas as pd
country_wise_df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
top5_recovery_countries = country_wise_df.sort_values(by='Recovered / 100 Cases', ascending=False).head(5)
print(top5_recovery_countries[['Country/Region', 'Recovered / 100 Cases']])
```

```
➞
```

	Country/Region	Recovered / 100 Cases
49	Dominica	100.00
75	Holy See	100.00
69	Grenada	100.00
48	Djibouti	98.38
78	Iceland	98.33

6. Create a NumPy boolean array where 'Active' cases are greater than 10,000.



```
import pandas as pd
import numpy as np
country_wise_df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
active_cases_array = country_wise_df['Active'].to_numpy()
active_cases_gt_10k = active_cases_array > 10000
print(active_cases_gt_10k)
```



```
[False False False False False False  True  True False False False False
 False  True False False  True False False False  True False False  True
 False False False False False False False False  True False False  True
 False  True False False False  True False False False False False False
 False False  True  True  True False False False False False False False
 False  True False False False False False False False False False  True False
 True  True False False False  True False False False  True False False
 False False False False False False False False False False False False
 True False False False  True False False  True  True  True False False
 True  True False  True False  True  True False False False False False
 False  True False  True False False False False False False  True False
 False  True False False False  True False False False False False False
 False False False False  True  True False  True False  True False False
 False False False False False False False]
```



7. Find and display the top 10 countries with the highest number of confirmed COVID-19 cases.

```
import pandas as pd
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
top_10_confirmed = df.sort_values('Confirmed', ascending=False).head(10)
print(top_10_confirmed[['Country/Region', 'Confirmed']].to_string(index=False))
```

```
Country/Region  Confirmed
US              4290259
Brazil          2442375
India           1480073
Russia           816680
South Africa    452529
Mexico           395489
Peru             389717
Chile            347923
United Kingdom  301708
Iran             293606
```



8. Calculate the recovery efficiency ratio (Recovered / Confirmed) using NumPy. Find countries where this ratio is below 0.5 (low recovery efficiency) using boolean indexing.

```
import numpy as np
import pandas as pd
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
confirmed_cases = df['Confirmed'].to_numpy() # Convert to NumPy array
total_cases = np.sum(confirmed_cases)
avg_cases = np.mean(confirmed_cases)
max_cases = np.max(confirmed_cases)
min_cases = np.min(confirmed_cases)
print(f"Total confirmed cases worldwide: {total_cases}")
print(f"Average cases per country: {avg_cases:.0f}")
print(f"Highest cases in a country: {max_cases}")
print(f"Lowest cases in a country: {min_cases}")
```

```
➞ Total confirmed cases worldwide: 16480485
Average cases per country: 88131
Highest cases in a country: 4290259
Lowest cases in a country: 10
```

## 9. List countries that reported more than 50 New deaths. give me code for this

```
import pandas as pd
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
result = df[df['New deaths'] > 50][['Country/Region', 'New deaths']] \
        .sort_values('New deaths', ascending=False)
print("Countries with more than 50 new deaths:")
print(result)
```

↩ Countries with more than 50 new deaths:

	Country/Region	New deaths
173	US	1076
79	India	637
23	Brazil	614
132	Peru	575
37	Colombia	508
111	Mexico	342
154	South Africa	298
81	Iran	212
6	Argentina	120
82	Iraq	96
138	Russia	85
35	Chile	75
20	Bolivia	64
80	Indonesia	57

## 10. Filter the dataset to show only countries where the number of confirmed cases is greater than 100,000



```
import pandas as pd
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
high_case_countries = df[df['Confirmed'] > 100000][['Country/Region', 'Confirmed']] \
    .sort_values('Confirmed', ascending=False)
print("Countries with more than 100,000 confirmed cases:")
print(high_case_countries)
```



Countries with more than 100,000 confirmed cases:

	Country/Region	Confirmed
173	US	4290259
23	Brazil	2442375
79	India	1480073
138	Russia	816680
154	South Africa	452529
111	Mexico	395489
132	Peru	389717
35	Chile	347923
177	United Kingdom	301708

## 11. Find countries where Recovered cases exceed 90% of Confirmed cases.



```
import pandas as pd

# Load the dataset
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')

# Calculate recovery percentage and filter
high_recovery = df[(df['Recovered'] / df['Confirmed'] > 0.9)][['Country/Region', 'Confirmed', 'Recovered']]

# Add recovery percentage column for clarity
high_recovery['Recovery_Percentage'] = (high_recovery['Recovered'] / high_recovery['Confirmed'] * 100).round(2)

# Sort by highest recovery percentage
high_recovery = high_recovery.sort_values('Recovery_Percentage', ascending=False)

print("Countries with recovery rate >90%:")
print(high_recovery)
```

Countries with recovery rate >90%:

	Country/Region	Confirmed	Recovered	Recovery_Percentage
69	Grenada	23	23	100.00
75	Holy See	12	12	100.00
49	Dominica	18	18	100.00
48	Djibouti	5059	4977	98.38
78	Iceland	1854	1823	98.33
24	Brunei	141	138	97.87
121	New Zealand	1557	1514	97.24

## 12. List countries that reported more than 50 New deaths



```
import pandas as pd

# Load the dataset
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')

# Filter countries with more than 50 new deaths
high_new_deaths = df[df['New deaths'] > 50]

# Select and sort the results
result = high_new_deaths[['Country/Region', 'New deaths']].sort_values('New deaths', ascending=False)

print("Countries with more than 50 new deaths:")
print(result)
```



Countries with more than 50 new deaths:

	Country/Region	New deaths
173	US	1076
79	India	637
23	Brazil	614
132	Peru	575
37	Colombia	508
111	Mexico	342
154	South Africa	298

### 13. Normalize the New cases column (scale values between 0 and 1) using NumPy



```
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
new_cases = df['New cases'].values
normalized_cases = (new_cases - np.min(new_cases)) / (np.max(new_cases) - np.min(new_cases))
df['New_cases_normalized'] = normalized_cases
print(df[['Country/Region', 'New cases', 'New_cases_normalized']].sort_values('New cases', ascending=False))
```



	Country/Region	New cases	New_cases_normalized
173	US	56336	1.000000
79	India	44457	0.789140
23	Brazil	23284	0.413306
37	Colombia	16306	0.289442
132	Peru	13756	0.244178
..	...	...	...
143	San Marino	0	0.000000
168	Timor-Leste	0	0.000000
166	Tanzania	0	0.000000
157	Spain	0	0.000000
183	Western Sahara	0	0.000000

#### 14. Use Pandas to count how many countries belong to each WHO Region.

```
▶ import pandas as pd
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
region_counts = df['WHO Region'].value_counts()
print("Number of Countries in Each WHO Region:")
print("=" * 40)
print(region_counts.to_string())
```

⇒ Number of Countries in Each WHO Region:

=====

WHO Region

Europe 56

Africa 48

Americas 35

Eastern Mediterranean 22

Western Pacific 16

South-East Asia 10



15. Create a list of countries where Deaths = 0 using Pandas filtering.

```
] zero_deaths = df[df['Deaths'] == 0]
print(f"Found {len(zero_deaths)} countries with zero deaths:")
print(", ".join(zero_deaths['Country/Region']))
```

Found 17 countries with zero deaths:  
Bhutan, Cambodia, Dominica, Eritrea, Fiji, Greenland, Grenada, Holy See, Laos, Mongolia, Papua New Guinea, Saint Kitts and Nevis, Saint Lucia, Saint Vincent

16. most successful at recovering patients relative to their death counts.

```
import pandas as pd
data = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
data['Recovery_Ratio'] = data['Recovered'] / data['Deaths']
top_recovery = data.nlargest(5, 'Recovery_Ratio')[['Country/Region', 'Recovered', 'Deaths', 'Recovery_Ratio']]
print("Top 5 countries with highest recovered-to-deaths ratio:")
print(top_recovery)
```

Top 5 countries with highest recovered-to-deaths ratio:

	Country/Region	Recovered	Deaths	Recovery_Ratio
19	Bhutan	86	0	inf
30	Cambodia	147	0	inf
49	Dominica	18	0	inf
55	Eritrea	191	0	inf
59	Fiji	18	0	inf

## 17. Compute the mean, median, and standard deviation of Recovered cases using NumPy.



```
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')

# Convert 'Recovered' column to NumPy array
recovered_cases = df['Recovered'].to_numpy()

# Calculate statistics
mean_recovered = np.mean(recovered_cases)
median_recovered = np.median(recovered_cases)
std_recovered = np.std(recovered_cases)

# Print results
print(f"Mean recovered cases: {mean_recovered:,.0f}")
print(f"Median recovered cases: {median_recovered:,.0f}")
print(f"Standard deviation of recovered cases: {std_recovered:,.0f}")
```



```
Mean recovered cases: 50,631
Median recovered cases: 2,815
Standard deviation of recovered cases: 189,679
```

## 18. Use NumPy to identify countries where Deaths = 0 and falls under the africa WHO region



```
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')

# Convert to NumPy arrays for computation
countries = df['Country/Region'].to_numpy()
deaths = df['Deaths'].to_numpy()
regions = df['WHO Region'].to_numpy()

# Find indices where Deaths = 0 and Region is Africa
zero_deaths_africa = np.where((deaths == 0) & (regions == 'Africa'))[0]

# Get the country names
result = countries[zero_deaths_africa]

print("Countries in Africa with zero reported deaths:")
print(result)
```



```
Countries in Africa with zero reported deaths:
['Eritrea' 'Seychelles']
```

## 19. countries having recovered cases more than avg recovered cases but still having new cases more than 10000



```
import pandas as pd
import numpy as np
df = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
avg_recovered = np.mean(df['Recovered'])
result = df[
    (df['Recovered'] > avg_recovered) &
    (df['New cases'] > 10000)
][['Country/Region', 'Recovered', 'New cases']]
result = result.sort_values('New cases', ascending=False)
print(f"Global average recovered cases: {avg_recovered:,.0f}")
print("\nCountries meeting both conditions:")
print(result)
```



Global average recovered cases: 50,631

Countries meeting both conditions:

	Country/Region	Recovered	New cases
173	US	1325804	56336
79	India	951166	44457
23	Brazil	1846641	23284
37	Colombia	131161	16306
132	Peru	272547	13756

20. Identify the top 5 countries with the highest number of new COVID-19 cases (New cases).

```
[ ] import pandas as pd

data = pd.read_csv('/root/.cache/kagglehub/datasets/imdevskp/corona-virus-report/versions/166/country_wise_latest.csv')
top_new_cases = data.nlargest(5, 'New cases')[['Country/Region', 'New cases']]
print("Top 5 countries with the most new cases:")
print(top_new_cases)
```

⇒ Top 5 countries with the most new cases:

	Country/Region	New cases
173	US	56336
79	India	44457
23	Brazil	23284
37	Colombia	16306
132	Peru	13756

*Thank*

*you*