

Name :- Shirsath Pranav Hiranman

PRN : 202401050001

Roll No : CC-04

Course :- Essential Data Science (EDS)

THEORY ACTIVITY NO.1

I'll first **formulate 20 problem statements** based on a **COVID-19 dataset** (assuming the dataset includes fields like Date, Country, Confirmed, Deaths, Recovered, Active, etc.).

Then I'll show how you would apply **Numpy and Pandas methods** to solve them.

---

## ✂ Problem Statements + Solutions

**Dataset columns assumed:**

`['Date', 'Country', 'Confirmed', 'Deaths', 'Recovered', 'Active']`

1. Find the country with the highest number of confirmed cases.

```
import pandas as pd
```

```
import numpy as np
```

```
# Assume df is the DataFrame
```

```
country_max_confirmed = df.groupby('Country')['Confirmed'].max().idxmax()
```

2. Find the total number of deaths globally.

```
total_deaths = df['Deaths'].sum()
```

---

### 3. Calculate the mortality rate (deaths/confirmed cases) for each country.

```
df['Mortality Rate'] = df['Deaths'] / df['Confirmed']
```

---

### 4. Find the country with the highest mortality rate.

```
highest_mortality_country = df.groupby('Country')['Mortality Rate'].mean().idxmax()
```

---

### 5. Find the top 5 countries with the most active cases.

```
top5_active =  
df.groupby('Country')['Active'].sum().sort_values(ascending=False).head(5)
```

---

### 6. Find the daily new confirmed cases for each country.

```
df['New Confirmed'] =  
df.groupby('Country')['Confirmed'].diff().fillna(0)
```

---

### 7. Find the first date when a country crossed 1000 confirmed cases.

```
threshold_crossed = df[df['Confirmed'] >  
1000].groupby('Country')['Date'].min()
```

---

### 8. Calculate the recovery rate (Recovered/Confirmed) for each country.

```
df['Recovery Rate'] = df['Recovered'] / df['Confirmed']
```

---

### 9. Find the country with the highest recovery rate.

```
highest_recovery_country = df.groupby('Country')['Recovery Rate'].mean().idxmax()
```

---

### 10. Find the day with the highest number of global new cases.

```
global_daily_new_cases =  
df.groupby('Date')['Confirmed'].sum().diff().fillna(0)  
date_highest_new_cases = global_daily_new_cases.idxmax()
```

---

### 11. Find the cumulative confirmed cases worldwide over time.

```
global_cumulative_cases =  
df.groupby('Date')['Confirmed'].sum().cumsum()
```

---

### 12. Identify countries where the number of recovered cases has surpassed active cases.

```
recovered_more_than_active = df[df['Recovered'] >  
df['Active']]['Country'].unique()
```

---

13. Find the average number of deaths per day globally.

```
avg_deaths_per_day = df.groupby('Date')['Deaths'].sum().mean()
```

---

14. Find the standard deviation of daily new cases globally.

```
std_new_cases = global_daily_new_cases.std()
```

---

15. Find countries with zero active cases.

```
countries_zero_active = df[df['Active'] == 0]['Country'].unique()
```

---

16. Find the country with the fastest growth in confirmed cases over 7 days.

```
df['7d Growth'] = df.groupby('Country')['Confirmed'].diff(periods=7)
fastest_growth_country = df.groupby('Country')['7d Growth'].max().idxmax()
```

---

17. Plot the trend of confirmed cases for the top 3 most affected countries.

```
import matplotlib.pyplot as plt

top3_countries =
df.groupby('Country')['Confirmed'].max().sort_values(ascending=False)
).head(3).index
for country in top3_countries:
    country_data = df[df['Country'] == country]
    plt.plot(country_data['Date'], country_data['Confirmed'],
label=country)

plt.legend()
plt.title('Confirmed Cases Trend - Top 3 Countries')
plt.xlabel('Date')
plt.ylabel('Confirmed Cases')
plt.show()
```

---

18. Calculate the total confirmed cases per continent (assuming continent info available).

```
# If there's a 'Continent' column
continent_cases = df.groupby('Continent')['Confirmed'].sum()
```

---

19. Identify the countries where the death toll doubled in less than 7 days.

```
df['Deaths Shifted'] = df.groupby('Country')['Deaths'].shift(7)
df['Death Growth'] = df['Deaths'] / df['Deaths Shifted']
countries_double_death = df[df['Death Growth'] >=
2]['Country'].unique()
```

---

20. Find the moving average (7 days) of confirmed cases globally.

```
global_confirmed = df.groupby('Date')['Confirmed'].sum()
global_moving_avg = global_confirmed.rolling(window=7).mean()
```

---

## Notes:

- **Pandas** helps in `groupby()`, `diff()`, `rolling()`, `shift()`, `fillna()`, `mean()`, `std()`, etc.
- **Numpy** helps with direct operations if needed like `np.mean()`, `np.std()` etc., on arrays extracted from Pandas.
- **Visualization** like in problem 17 uses `Matplotlib`.
- These examples assume the dataset is **clean and well-formatted** (e.g., no missing dates).