covid19

May 16, 2025

[4]:

```python
import pandas as pd
df = pd.read_csv('covid19_data.csv')
```

[8]:

```python
df.head()     # Shows first 5 rows
df.info()     # Summary of columns and data types
df.describe() # Statistical summary
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30397 entries, 0 to 30396
Data columns (total 67 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   iso_code                                    30397 non-null  object
 1   continent                                   27039 non-null  object
 2   location                                    30397 non-null  object
 3   date                                        30397 non-null  object
 4   total_cases                                 30383 non-null  float64
 5   new_cases                                   30383 non-null  float64
 6   new_cases_smoothed                          30288 non-null  float64
 7   total_deaths                                30383 non-null  float64
 8   new_deaths                                  30382 non-null  float64
 9   new_deaths_smoothed                         30287 non-null  float64
 10  total_cases_per_million                     30383 non-null  float64
 11  new_cases_per_million                       30383 non-null  float64
 12  new_cases_smoothed_per_million              30288 non-null  float64
 13  total_deaths_per_million                    30382 non-null  float64
 14  new_deaths_per_million                      30381 non-null  float64
 15  new_deaths_smoothed_per_million             30286 non-null  float64
 16  reproduction_rate                           13065 non-null  float64
 17  icu_patients                                4004 non-null   float64
 18  icu_patients_per_million                    4004 non-null   float64
 19  hosp_patients                               2533 non-null   float64
 20  hosp_patients_per_million                   2533 non-null   float64
 21  weekly_icu_admissions                       0 non-null      float64
 22  weekly_icu_admissions_per_million           0 non-null      float64
 23  weekly_hosp_admissions                      0 non-null      float64
 24  weekly_hosp_admissions_per_million          0 non-null      float64
 25  total_tests                                 5344 non-null   float64
 26  new_tests                                   4968 non-null   float64
 27  total_tests_per_thousand                    5344 non-null   float64
 28  new_tests_per_thousand                      4968 non-null   float64
 29  new_tests_smoothed                          7055 non-null   float64
 30  new_tests_smoothed_per_thousand             7055 non-null   float64
 31  positive_rate                               6719 non-null   float64
 32  tests_per_case                              6608 non-null   float64
 33  tests_units                                 7156 non-null   object
 34  total_vaccinations                          6761 non-null   float64
 35  people_vaccinated                           6548 non-null   float64
 36  people_fully_vaccinated                     6452 non-null   float64
 37  total_boosters                              4399 non-null   float64
 38  new_vaccinations                            5636 non-null   float64
 39  new_vaccinations_smoothed                   15534 non-null  float64
 40  total_vaccinations_per_hundred              6761 non-null   float64
 41  people_vaccinated_per_hundred               6548 non-null   float64
 42  people_fully_vaccinated_per_hundred         6452 non-null   float64
 43  total_boosters_per_hundred                  4399 non-null   float64
 44  new_vaccinations_smoothed_per_million       15534 non-null  float64
 45  new_people_vaccinated_smoothed              14908 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred  14908 non-null  float64
 47  stringency_index                            13358 non-null  float64
 48  population_density                          25364 non-null  float64
 49  median_age                                  22016 non-null  float64
 50  aged_65_older                               22016 non-null  float64
 51  aged_70_older                               22016 non-null  float64
 52  gdp_per_capita                              22016 non-null  float64
 53  extreme_poverty                             10298 non-null  float64
 54  cardiovasc_death_rate                       23690 non-null  float64
 55  diabetes_prevalence                         23690 non-null  float64
 56  female_smokers                              16994 non-null  float64
 57  male_smokers                                16994 non-null  float64
 58  handwashing_facilities                      8620 non-null   float64
 59  hospital_beds_per_thousand                  18668 non-null  float64
 60  life_expectancy                             27038 non-null  float64
 61  human_development_index                     22016 non-null  float64
 62  population                                  30396 non-null  float64
 63  excess_mortality_cumulative_absolute        742 non-null    float64
 64  excess_mortality_cumulative                 742 non-null    float64
 65  excess_mortality                            742 non-null    float64
 66  excess_mortality_cumulative_per_million     742 non-null    float64
dtypes: float64(62), object(5)
memory usage: 15.5+ MB
```

```
[8]: iso_code                                       0
     continent                                   3358
```

```
location                                        0
date                                            0
total_cases                                    14
                                            …
population                                      1
excess_mortality_cumulative_absolute        29655
excess_mortality_cumulative                 29655
excess_mortality                            29655
excess_mortality_cumulative_per_million     29655
Length: 67, dtype: int64
```

[19]:

```
import pandas as pd

# Load dataset
df = pd.read_csv("covid19_data.csv")  # replace with your filename

# Show initial info
print(df.info())

# Step 1: Drop rows with missing critical values (e.g., Date, Country,␣
 ↪Confirmed cases)
df = df.dropna(subset=['Date', 'Country', 'Confirmed'])

# Step 2: Convert 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Step 3: Filter for specific countries
countries_of_interest = ['Kenya', 'USA', 'India']
df = df[df['Country'].isin(countries_of_interest)]

# Step 4: Handle missing numeric values
# Option A: Fill with 0
df[['Confirmed', 'Recovered', 'Deaths']] = df[['Confirmed', 'Recovered',␣
 ↪'Deaths']].fillna(0)

# Option B: Interpolate missing values (optional)
# df[['Confirmed', 'Recovered', 'Deaths']] = df[['Confirmed', 'Recovered',␣
 ↪'Deaths']].interpolate()

# Final check
print(df.head())
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30397 entries, 0 to 30396
Data columns (total 67 columns):
 #   Column                                      Non-Null Count  Dtype
```

```
---  ------                                      --------------  -----
 0   iso_code                                    30397 non-null  object
 1   continent                                   27039 non-null  object
 2   location                                    30397 non-null  object
 3   date                                        30397 non-null  object
 4   total_cases                                 30383 non-null  float64
 5   new_cases                                   30383 non-null  float64
 6   new_cases_smoothed                          30288 non-null  float64
 7   total_deaths                                30383 non-null  float64
 8   new_deaths                                  30382 non-null  float64
 9   new_deaths_smoothed                         30287 non-null  float64
 10  total_cases_per_million                     30383 non-null  float64
 11  new_cases_per_million                       30383 non-null  float64
 12  new_cases_smoothed_per_million              30288 non-null  float64
 13  total_deaths_per_million                    30382 non-null  float64
 14  new_deaths_per_million                      30381 non-null  float64
 15  new_deaths_smoothed_per_million             30286 non-null  float64
 16  reproduction_rate                           13065 non-null  float64
 17  icu_patients                                4004 non-null   float64
 18  icu_patients_per_million                    4004 non-null   float64
 19  hosp_patients                               2533 non-null   float64
 20  hosp_patients_per_million                   2533 non-null   float64
 21  weekly_icu_admissions                       0 non-null      float64
 22  weekly_icu_admissions_per_million           0 non-null      float64
 23  weekly_hosp_admissions                      0 non-null      float64
 24  weekly_hosp_admissions_per_million          0 non-null      float64
 25  total_tests                                 5344 non-null   float64
 26  new_tests                                   4968 non-null   float64
 27  total_tests_per_thousand                    5344 non-null   float64
 28  new_tests_per_thousand                      4968 non-null   float64
 29  new_tests_smoothed                          7055 non-null   float64
 30  new_tests_smoothed_per_thousand             7055 non-null   float64
 31  positive_rate                               6719 non-null   float64
 32  tests_per_case                              6608 non-null   float64
 33  tests_units                                 7156 non-null   object
 34  total_vaccinations                          6761 non-null   float64
 35  people_vaccinated                           6548 non-null   float64
 36  people_fully_vaccinated                     6452 non-null   float64
 37  total_boosters                              4399 non-null   float64
 38  new_vaccinations                            5636 non-null   float64
 39  new_vaccinations_smoothed                   15534 non-null  float64
 40  total_vaccinations_per_hundred              6761 non-null   float64
 41  people_vaccinated_per_hundred               6548 non-null   float64
 42  people_fully_vaccinated_per_hundred         6452 non-null   float64
 43  total_boosters_per_hundred                  4399 non-null   float64
 44  new_vaccinations_smoothed_per_million       15534 non-null  float64
 45  new_people_vaccinated_smoothed              14908 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred  14908 non-null  float64
 47  stringency_index                          13358 non-null  float64
 48  population_density                        25364 non-null  float64
 49  median_age                                22016 non-null  float64
 50  aged_65_older                             22016 non-null  float64
 51  aged_70_older                             22016 non-null  float64
 52  gdp_per_capita                            22016 non-null  float64
 53  extreme_poverty                           10298 non-null  float64
 54  cardiovasc_death_rate                     23690 non-null  float64
 55  diabetes_prevalence                       23690 non-null  float64
 56  female_smokers                            16994 non-null  float64
 57  male_smokers                              16994 non-null  float64
 58  handwashing_facilities                    8620 non-null   float64
 59  hospital_beds_per_thousand                18668 non-null  float64
 60  life_expectancy                           27038 non-null  float64
 61  human_development_index                   22016 non-null  float64
 62  population                                30396 non-null  float64
 63  excess_mortality_cumulative_absolute      742 non-null    float64
 64  excess_mortality_cumulative               742 non-null    float64
 65  excess_mortality                          742 non-null    float64
 66  excess_mortality_cumulative_per_million   742 non-null    float64
dtypes: float64(62), object(5)
memory usage: 15.5+ MB
None
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/tmp/ipykernel_416/1215905071.py in ?()
      6 # Show initial info
      7 print(df.info())
      8
      9 # Step 1: Drop rows with missing critical values (e.g., Date, Country,␣
 ↪Confirmed cases)
---> 10 df = df.dropna(subset=['Date', 'Country', 'Confirmed'])
     11
     12 # Step 2: Convert 'Date' column to datetime format
     13 df['Date'] = pd.to_datetime(df['Date'])

/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/pandas/
 ↪core/frame.py in ?(self, axis, how, thresh, subset, inplace, ignore_index)
   6403             ax = self._get_axis(agg_axis)
   6404             indices = ax.get_indexer_for(subset)
   6405             check = indices == -1
   6406             if check.any():
-> 6407                 raise KeyError(np.array(subset)[check].tolist())
   6408             agg_obj = self.take(indices, axis=agg_axis)
   6409
   6410         if thresh is not no_default:
```

```
KeyError: ['Date', 'Country', 'Confirmed']
```

[23]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Optional: for better-looking plots
sns.set(style='whitegrid')

df = pd.read_csv("covid19_data.csv")  # Replace with your actual file name

# Remove whitespace from column names (just in case)
df.columns = df.columns.str.strip()

# Drop rows with missing critical values
df = df.dropna(subset=['Date', 'Country', 'Confirmed', 'Deaths'])

# Convert 'Date' to datetime
df['Date'] = pd.to_datetime(df['Date'])

# Filter selected countries
countries = ['Kenya', 'USA', 'India']
df = df[df['Country'].isin(countries)]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/tmp/ipykernel_416/1360106656.py in ?()
     10 # Remove whitespace from column names (just in case)
     11 df.columns = df.columns.str.strip()
     12
     13 # Drop rows with missing critical values
---> 14 df = df.dropna(subset=['Date', 'Country', 'Confirmed', 'Deaths'])
     15
     16 # Convert 'Date' to datetime
     17 df['Date'] = pd.to_datetime(df['Date'])

/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/pandas/
 ↪core/frame.py in ?(self, axis, how, thresh, subset, inplace, ignore_index)
   6403                 ax = self._get_axis(agg_axis)
   6404                 indices = ax.get_indexer_for(subset)
   6405                 check = indices == -1
   6406                 if check.any():
-> 6407                     raise KeyError(np.array(subset)[check].tolist())
   6408                 agg_obj = self.take(indices, axis=agg_axis)
   6409
   6410             if thresh is not no_default:
```

```
KeyError: ['Date', 'Country', 'Confirmed', 'Deaths']
```

[22]:

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/tmp/ipykernel_416/1265528812.py in ?()
      3 # Remove whitespace from column names (just in case)
      4 df.columns = df.columns.str.strip()
      5
      6 # Drop rows with missing critical values
----> 7 df = df.dropna(subset=['Date', 'Country', 'Confirmed', 'Deaths'])
      8
      9 # Convert 'Date' to datetime
     10 df['Date'] = pd.to_datetime(df['Date'])

/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/pandas/
 ↪core/frame.py in ?(self, axis, how, thresh, subset, inplace, ignore_index)
   6403             ax = self._get_axis(agg_axis)
   6404             indices = ax.get_indexer_for(subset)
   6405             check = indices == -1
   6406             if check.any():
-> 6407                 raise KeyError(np.array(subset)[check].tolist())
   6408             agg_obj = self.take(indices, axis=agg_axis)
   6409
   6410         if thresh is not no_default:

KeyError: ['Date', 'Country', 'Confirmed', 'Deaths']
```

[24]:

```python
plt.figure(figsize=(12, 6))
for country in countries:
    country_data = df[df['Country'] == country]
    grouped = country_data.groupby('Date')['Confirmed'].sum()
    plt.plot(grouped.index, grouped.values, label=country)

plt.title("Total Confirmed Cases Over Time")
plt.xlabel("Date")
plt.ylabel("Total Cases")
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
```

```
Cell In[24], line 2
      1 plt.figure(figsize=(12, 6))
----> 2 for country in countries:
      3     country_data = df[df['Country'] == country]
      4     grouped = country_data.groupby('Date')['Confirmed'].sum()

NameError: name 'countries' is not defined
```

```
<Figure size 1200x600 with 0 Axes>
```

[25]:

```
plt.figure(figsize=(12, 6))
for country in countries:
    country_data = df[df['Country'] == country].groupby('Date')['Confirmed'].
 ↪sum().diff().fillna(0)
    plt.plot(country_data.index, country_data.values, label=country)

plt.title("Daily New Cases Comparison")
plt.xlabel("Date")
plt.ylabel("New Cases")
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[25], line 2
      1 plt.figure(figsize=(12, 6))
----> 2 for country in countries:
      3     country_data = df[df['Country'] == country].
 ↪groupby('Date')['Confirmed'].sum().diff().fillna(0)
      4     plt.plot(country_data.index, country_data.values, label=country)

NameError: name 'countries' is not defined
```

```
<Figure size 1200x600 with 0 Axes>
```

[ ]: