**Phase 1: Problem Definition and Design Thinking**

**Problem Definition:**

The project aims to conduct an in-depth analysis of COVID-19 vaccine data, focusing on aspects such as vaccine efficacy, distribution, and adverse effects. The ultimate goal is to provide actionable insights that can assist policymakers and health organizations in optimizing their vaccine deployment strategies. This multifaceted project involves several critical steps, including data collection, pre-processing, exploratory data analysis, statistical analysis, and data visualization.

**Design Thinking:**

This project embodies a thorough examination of COVID-19 vaccine data, emphasizing vaccine effectiveness, distribution strategies, and adverse effects. Its primary objective is to offer valuable insights to support decision-making for health organizations and policymakers. The project unfolds in a structured manner:

**1. Data Collection:** The initial step involves gathering comprehensive COVID-19 vaccine data from a variety of sources, including government agencies, public health organizations, and research institutions. This data encompasses critical details related to vaccine efficacy, distribution patterns, and adverse effects.

**2. Data Preprocessing:** Subsequent to data collection, rigorous preprocessing occurs. This phase includes cleansing and preparing the data, ensuring consistency, and addressing missing or irregular entries. The objective is to create a clean, homogeneous dataset for analysis.

**3. Exploratory Data Analysis (EDA):** EDA is instrumental in uncovering patterns and relationships within the data. By employing data visualization techniques and statistical analysis, the project identifies insights that can inform further investigation.

**4. Statistical Analysis:** The statistical analysis phase aims to validate hypotheses generated during EDA. Rigorous statistical tests are executed to determine statistically significant relationships between variables of interest.

**5. Data Visualization:** The insights derived from the statistical analysis are translated into visual representations, making complex data more accessible. Various types of plots and charts are generated, including time series plots, bar charts, and error bars.

**6. Insight Generation:** The project's insights are diligently examined, searching for patterns, variations, and correlations within the data. These insights can lead to recommendations or actionable strategies for addressing and mitigating the COVID-19 situation. Additionally, a system for continuous monitoring and analysis is considered, recognizing the dynamic nature of the COVID-19 landscape.

The findings from this comprehensive project will be disseminated to policymakers and health organizations through various channels, including peer-reviewed journals, conference presentations, and policy reports. The project's potential impacts include enhancing vaccine deployment strategies, mitigating COVID-19 cases, hospitalizations, and deaths, and addressing vaccine equity issues. It is poised to be a significant contributor in the global fight against COVID-19.

**PHASE-2: Document Submission**

In this phase, the focus shifts towards compiling the project documentation, which encompasses various stages of problem definition, design thinking, data preprocessing, and visualization.

**Phase 3: Data Preprocessing and Visualization**

**Introduction:**

This project's core objective is to conduct a comprehensive analysis of COVID-19 vaccination data, providing valuable insights into vaccine efficacy, distribution, and adverse effects. The COVID-19 pandemic has had a profound impact worldwide, and understanding the data can significantly contribute to informed decision-making.

**Given Dataset:**

The dataset was sourced from a variety of reliable sources, and the data preprocessing and visualization phases were initiated to make the data analysis-ready.

**Data Preprocessing:**

The preprocessing phase is crucial to ensure the data's integrity, reliability, and usability. It involves handling missing values, removing outliers, and transforming data. Steps included:

**1. Importing Libraries:** The project initiated by importing necessary libraries for data analysis and visualization.

**2. Loading the Dataset**: The dataset was loaded and examined to gain insights into its structure, including its dimensions, data types, and missing values.

**3. Data Cleaning:** This phase aimed to address missing values, remove irrelevant data, and ensure the dataset's consistency and quality.

**4. Handling Missing Data:** A crucial step involved handling missing data through imputation and careful data inspection.

**5. Data Transformation**: Converting data into a suitable format for analysis was performed, including converting data types.

**Data Visualization:**

Data visualization enhances the understanding of the dataset's patterns and relationships. It included:

1. Total Vaccination per Country: Visualizing the distribution of total vaccinations by country.

2. Total Vaccinations and People Vaccinated: Examining the relationship between total vaccinations and people vaccinated.

3. Total Vaccinations by ISO Code: Analyzing the total vaccinations based on ISO codes.

4. Daily Vaccinations by ISO Code: Visualizing daily vaccination trends based on ISO codes.

**Conclusion:**

The data preprocessing phase successfully prepared the dataset for advanced analysis. Data cleaning, handling of missing data, and transformation are essential steps to ensure the data's quality.

Data visualization provided insights into various aspects of the data, offering a clear perspective on the distribution of vaccinations across countries and their correlations. These initial findings pave the way for more in-depth analysis, hypothesis validation, and actionable conclusions.

**Phase 4: Data Analysis and Model Building**

**Aim:**

The project's primary aim is to analyse COVID-19 vaccination data comprehensively, gaining insights into vaccine efficacy, distribution, and adverse effects. This phase focuses on exploratory data analysis (EDA), statistical analysis, and model building.

**Exploratory Data Analysis (EDA):**

EDA is a foundational step that involves initial data inspection, data cleaning, and the identification of patterns and trends within the dataset. Key aspects include:

**1. Data Inspection:** An initial overview of the dataset's structure and characteristics, including dimensions, data types, and missing values.

**2. Data Cleaning:** The process of addressing missing values, eliminating irrelevant data, and ensuring data consistency.

**3. Handling Missing Data:** Strategies for handling missing data, which can impact the overall analysis.

**4. Data Transformation:** Conversion of data into suitable formats for analysis.

**Statistical Analysis:**

Statistical analysis is a vital component of data analysis, encompassing various techniques and tests to answer specific questions:

**1. Hypothesis Testing:** The use of statistical tests to compare different

variables, such as investigating significant differences in infection rates between regions.

**2. Time-Series Analysis:** Analyzing data over time to identify trends, seasonal variations, or the effects of public health measures. Time-dependent patterns are captured through models like ARIMA.

**Model Building:**

Model building is an essential phase in data analysis, as it allows for insights and predictions. Key components include:

**1. Data Splitting**: Division of the dataset into training and testing data.

**2. Scaling Data:** Ensuring data is scaled appropriately for modeling.

**3. Regression Analysis:** Performing linear regression and evaluating model performance.

**4. Evaluation**: Assessing the model's performance, such as the coefficient of determination (R-squared).

**5. Visualization:** Employing data visualization techniques to understand relationships between variables.

**Conclusion:**

The project has reached a critical phase, with a focus on data analysis and model building. The EDA phase has provided essential insights, laying the foundation for statistical analysis and model development.

Statistical analysis is employed to validate hypotheses, while model building enables predictions and insights into vaccine efficacy, distribution, and adverse effects. Data visualization enhances the understanding of these relationships, setting the stage for informed decisions and strategies.

**Instructions on how to replicate the analysis and generate visualizations using IBM Cognos and Python:**

**Step 1:** Data Loading and Preprocessing in python:

1. Import necessary libraries:

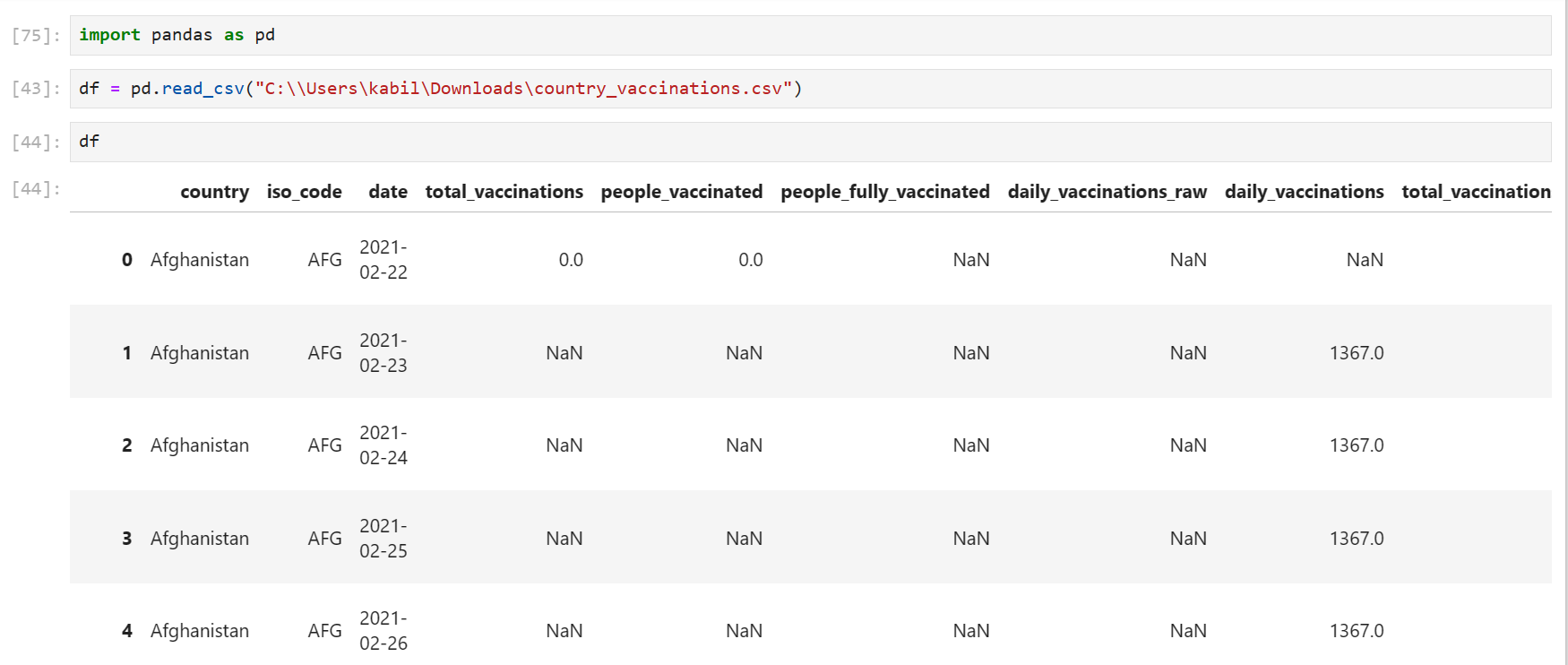
- Open the Python environment (e.g., Jupyter Notebook).

- Start a new Python script or notebook.

2. Load the dataset:

- Use the **pd.read\_csv()** function from the Pandas library to load the dataset.

- This code reads the dataset into a Pandas DataFrame named **df**.



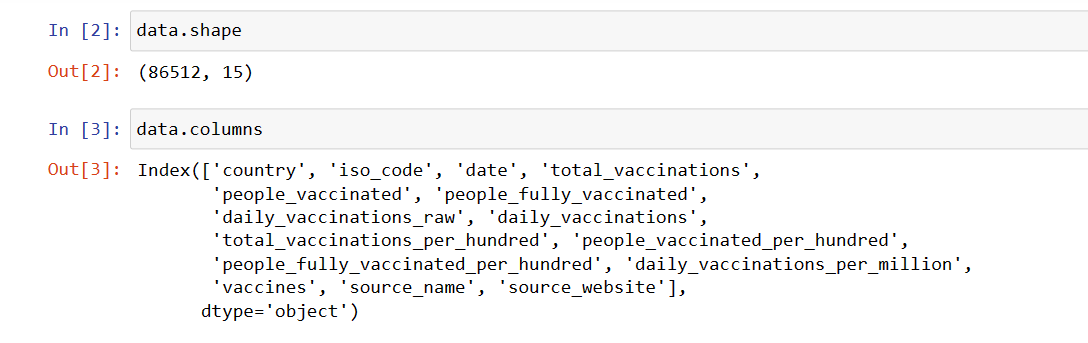
3. Explore basic dataset information:

- Use the **.info()** method to display information about the dataset, including data types and missing values.

- Use **.tail()** to see the last few rows of the dataset.

- Use **.columns** to list the column names.

- Use **.shape** to get the number of rows and columns.



4. Data cleaning:

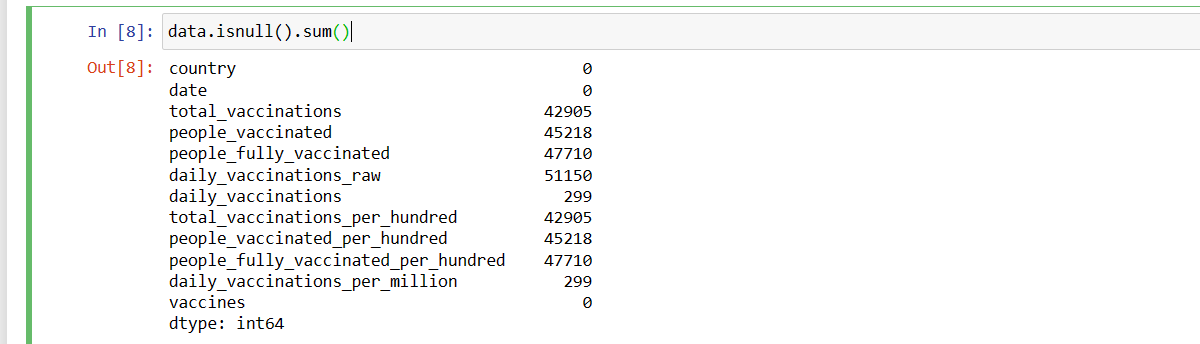
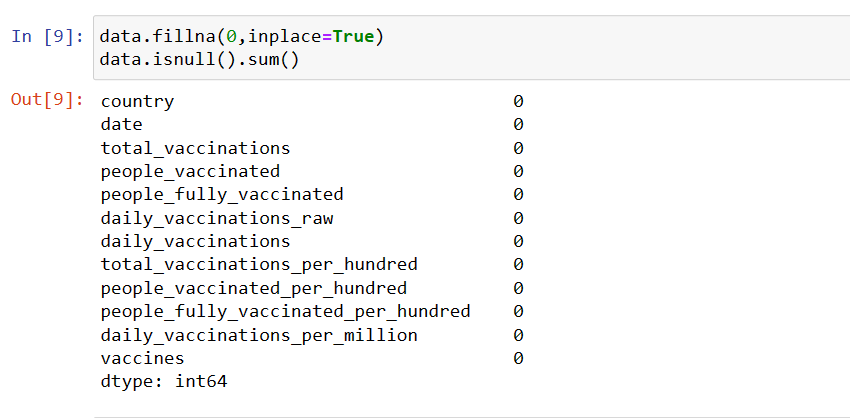
- Remove unnecessary columns ("source\_name" and "source\_website") using **.drop()**.

- Use **.describe()** and **.apply()** to show descriptive statistics with formatted numbers.

- Check for missing values using **.isnull().sum()**.

- Replace missing values with 0 using **.fillna()**.

- Adjust data types for specific columns using **.astype()**.

5. Save the preprocessed data:

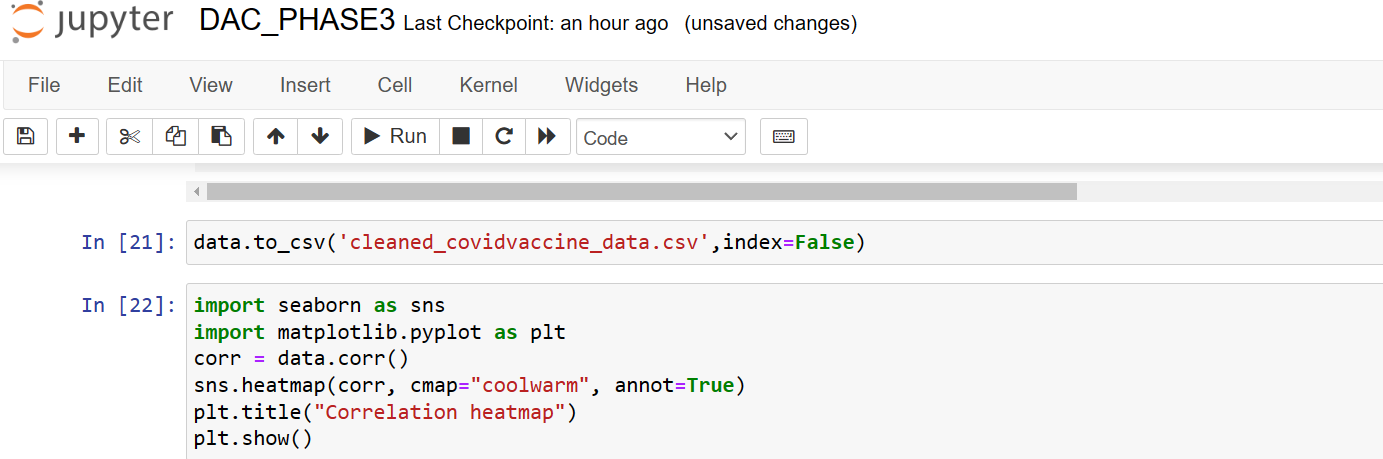
- Use **.to\_csv()** to save the preprocessed DataFrame to a new CSV file.

6. Now, run the script to load, preprocess, and save the data.

**Step 2:** Exploratory Analysis and Visualization

1. Import visualization libraries:

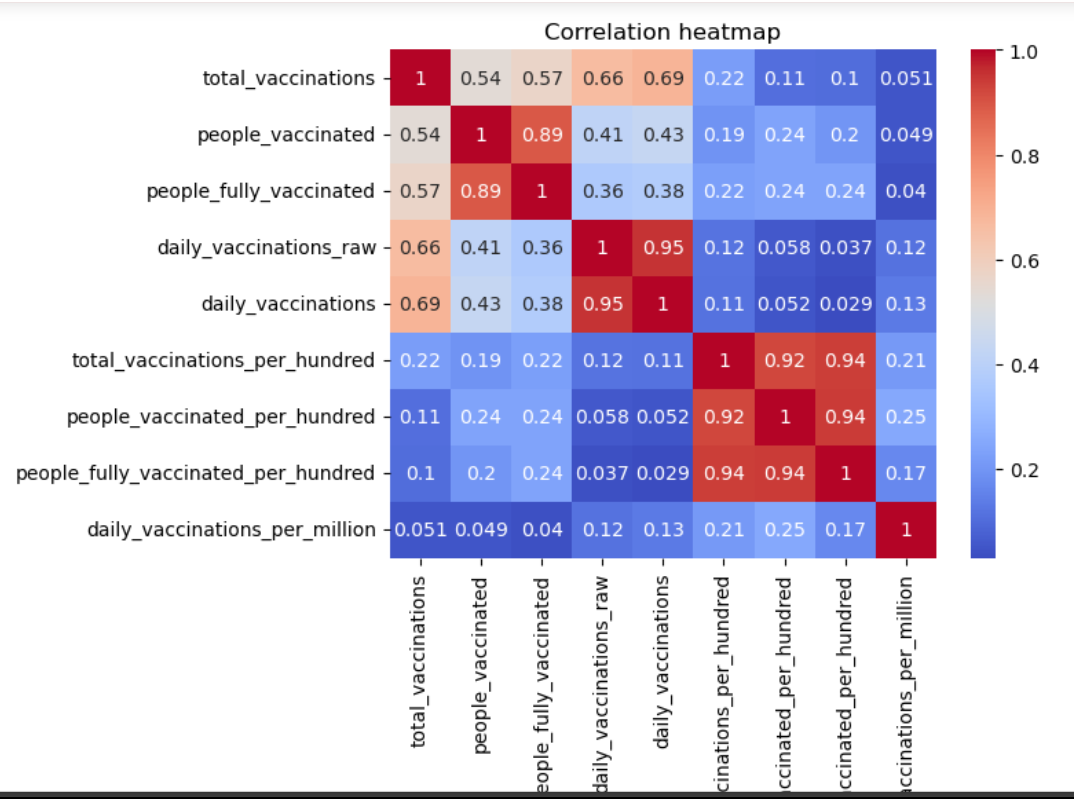
- Import libraries like Seaborn, Matplotlib, and Pandas for exploratory data analysis (EDA) and visualization.



2. Set up plotting style and configurations:

- Use Seaborn for styling with **sns.set\_style()** to choose a style (e.g., 'darkgrid').

- Adjust font size, figure size, and face color for plots using **matplotlib.rcParams**.



3. Explore data statistics:

- Set and reset options to format numeric values for better readability using **pd.set\_option()** and **pd.reset\_option()**.

4. Calculate and display statistics:

- Compute and display statistics like mean, min, max, and correlation between numeric columns using Pandas and NumPy functions.



5. Import necessary libraries for statistical analysis:

- Import Pandas, NumPy, and SciPy for statistical analysis.

6. Define a variable and perform a one-sample t-test:

- Set up variables (e.g., 'total\_vaccinations' and 'expected\_mean').

- Perform a one-sample t-test to check if 'total\_vaccinations' significantly differs from the 'expected\_mean'.

- Print the test results and check for statistical significance using a chosen alpha level.

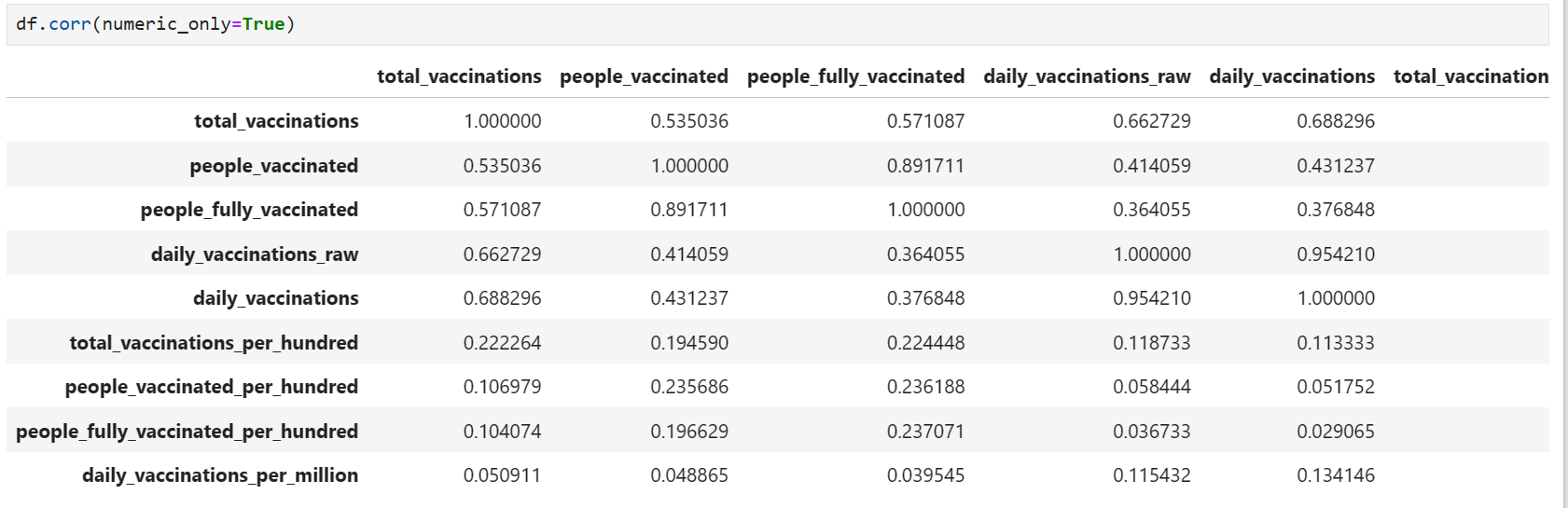


7. Calculate descriptive statistics:

- Calculate descriptive statistics (mean, median, standard deviation, variance) for 'total\_vaccinations' using Pandas functions.

8. Conduct correlation analysis:

- Calculate and display the correlation matrix between numeric columns in the dataset using **.corr()**.

****

9. Explore the 'country' column:

- Examine the distribution of unique country values using **.value\_counts()**.

- List the unique countries and count using **.country**.

- Get the total number of unique countries using **.nunique()**.

10. Explore 'people\_fully\_vaccinated':

- Find the minimum and maximum values of the 'people\_fully\_vaccinated' column using **.min()** and **.max()**.

11. Explore date information:

- Determine the minimum and maximum dates in the 'date' column to understand the time frame of the dataset.

12. Plot the number of daily vaccinations over time:

- Create a line plot using Seaborn to visualize the trend in the number of daily vaccinations over time.

- Customize the plot's appearance (e.g., title) using Matplotlib functions.

13. Run the script to perform exploratory analysis and visualize the dataset.

Simply run each section of code in the Python environment, following these step-by-step instructions, to process and analyze the data effectively.

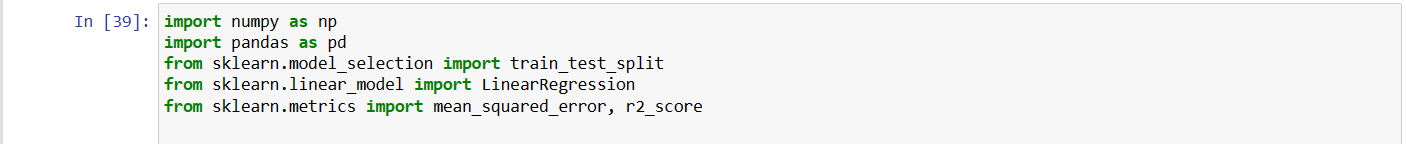
**Step 4: Model Building**

**Step 4.1: Importing Libraries**

1. Import the necessary libraries for model building:

- Ensure you have libraries such as scikit-learn (sklearn), pandas, numpy, and any other relevant libraries.

- You might need specific functions or modules depending on the model choice.

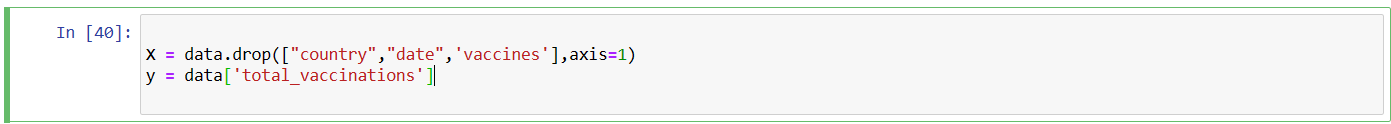


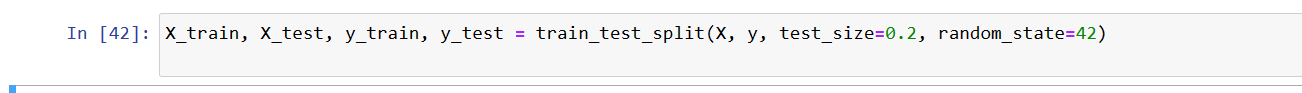
**Step 4.2: Splitting Data**

1. Split the dataset into training and testing data:

- Divide the data into two subsets: one for training the model and one for testing its performance.

- Use scikit-learn's `**train\_test\_split**` function for this purpose.



2. Specify the target variable (dependent variable) and features (independent variables): 

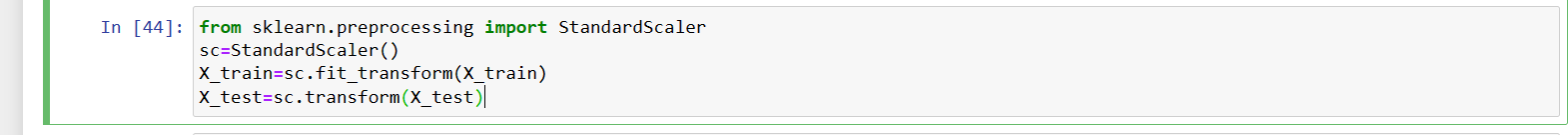
- Identify the variables you want to predict (target) and the variables you'll use for prediction (features).

**Step 4.3: Scaling Data**

1. Scale the dataset:

- Scaling is a crucial preprocessing step to ensure that all features have similar scales.

- Use methods like Min-Max scaling, Standardization, or Robust scaling based on the data.

****

**Step 4.4: Choosing a Model**

1. Decide the type of model based on the problem:

- Determine whether you're dealing with a regression or classification problem.

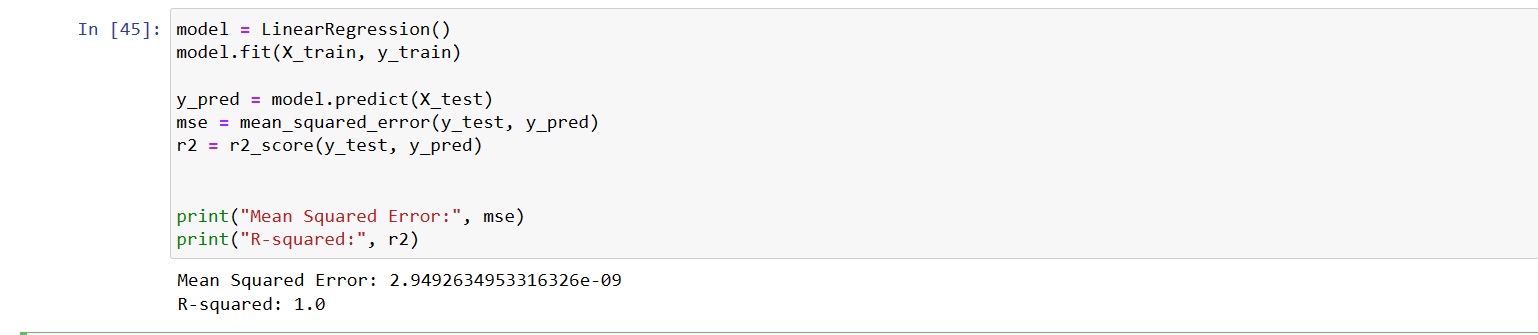
- Consider whether you prioritize model interpretability or high accuracy.

2. Choose a specific model algorithm:

- Select the model type from a wide range available in scikit-learn (e.g., Linear Regression, Decision Trees, Random Forest, Support Vector Machines, etc.).

- Based on the problem's characteristics, choose the most appropriate model.

Here we are performing simple **linear regression** in our dataset

****

**Step 4.5: Training the Model**

1. Train the selected model:

- Fit the model to the training data using the `fit` method.

- Provide the training features and target variables as input.

- For example: `model.fit(X\_train, y\_train)`.

**Step 4.6: Evaluating Model Performance**

1. Evaluate the model's performance:

- Calculate metrics relevant to the problem, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R^2), or classification metrics like accuracy and F1-score.

- Use scikit-learn functions like `predict` to generate predictions and then calculate performance metrics.

- For regression problems: `y\_pred = model.predict(X\_test)`

- For classification problems: `y\_pred = model.predict(X\_test)`

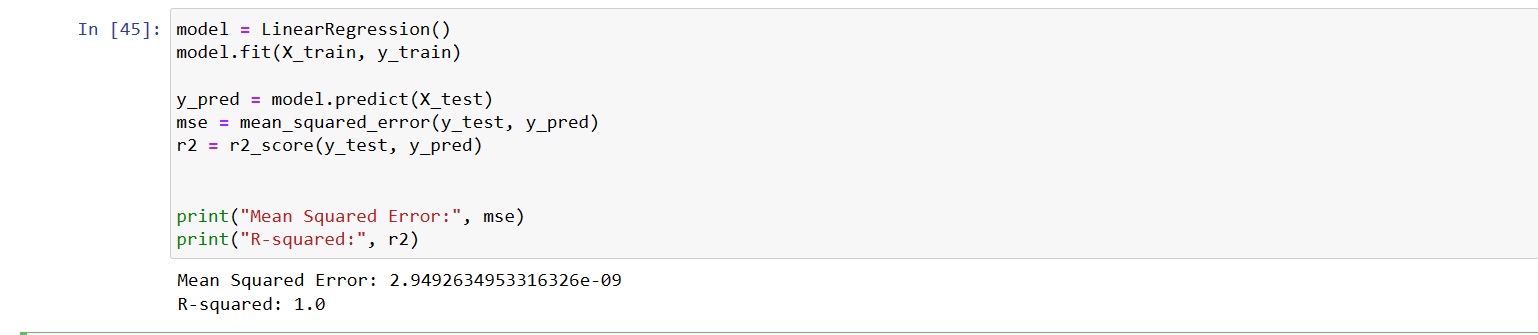
2. Assess model accuracy and fit:

- Check the R-squared (R^2) value; closer to 1 indicates better model fit for regression.

- Analyze other relevant metrics based on the problem type.

3. Interpret results:

- A high R^2 and low errors (MAE, MSE) indicate a well-fitting model.

- Analyze the model's strengths and limitations based on the context of the project. ****

**Step 5: Visualization using IBM Cognos Analytics**

**Step 5.1: Load the Dataset:**

1. Make sure you have the dataset you want to visualize.

2. Open IBM Cognos Analytics and create a new project or report where you want to perform the visualizations.

3. Load the dataset into the project or report.

**Step 5.2: Visualize the Number of Daily Vaccinations by Date:**

1. In IBM Cognos Analytics, navigate to the project or report where the dataset is loaded.

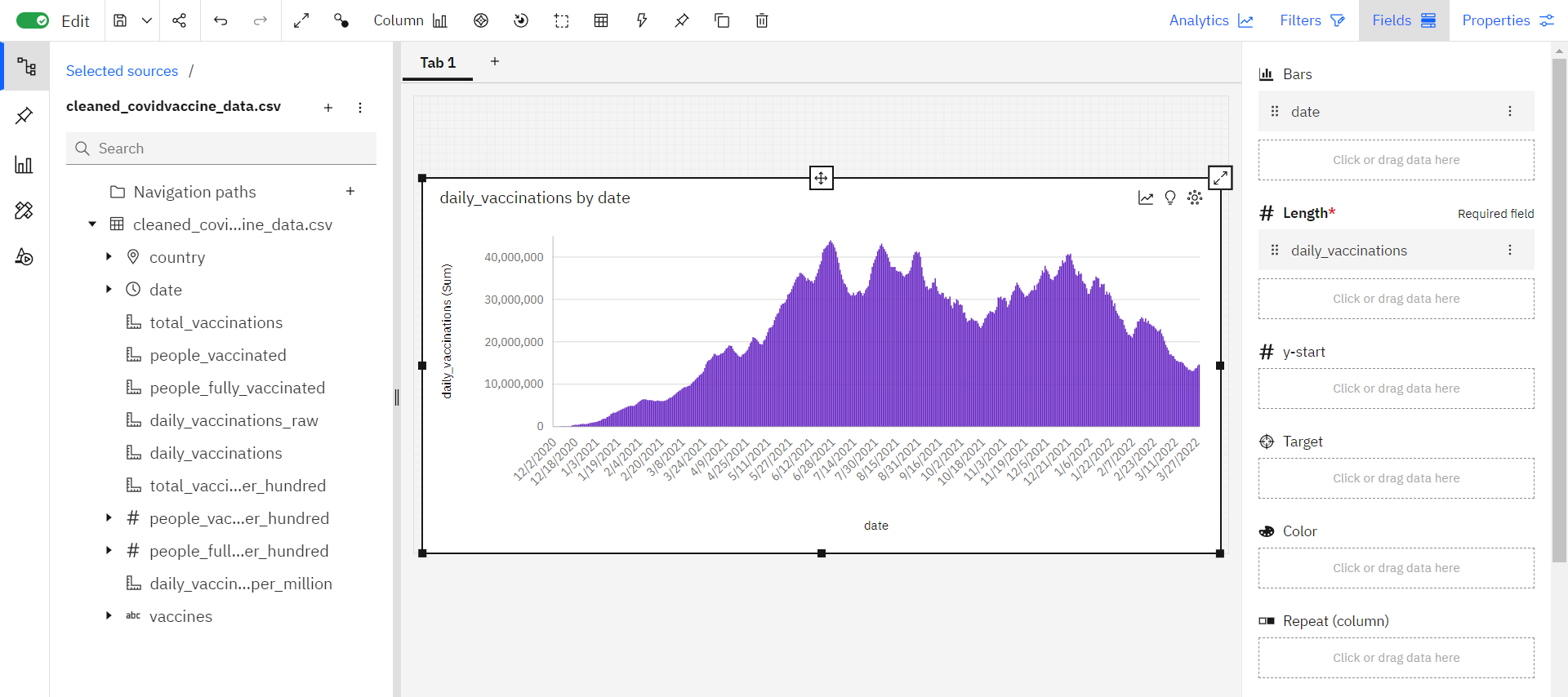
2. Choose the "Number of Daily Vaccinations by Date" as the specific visualization you want to create.

3. Set the appropriate visualization type, such as line chart or time series plot, to represent daily vaccination data.

4. Configure the X-axis to represent the date and the Y-axis for daily vaccinations.

5. Customize the appearance, labels, and legends as needed.

6. Generate the visualization to display the trends in daily vaccinations over time.



**Step 5.3: Visualize Total Vaccinations by Vaccine:**

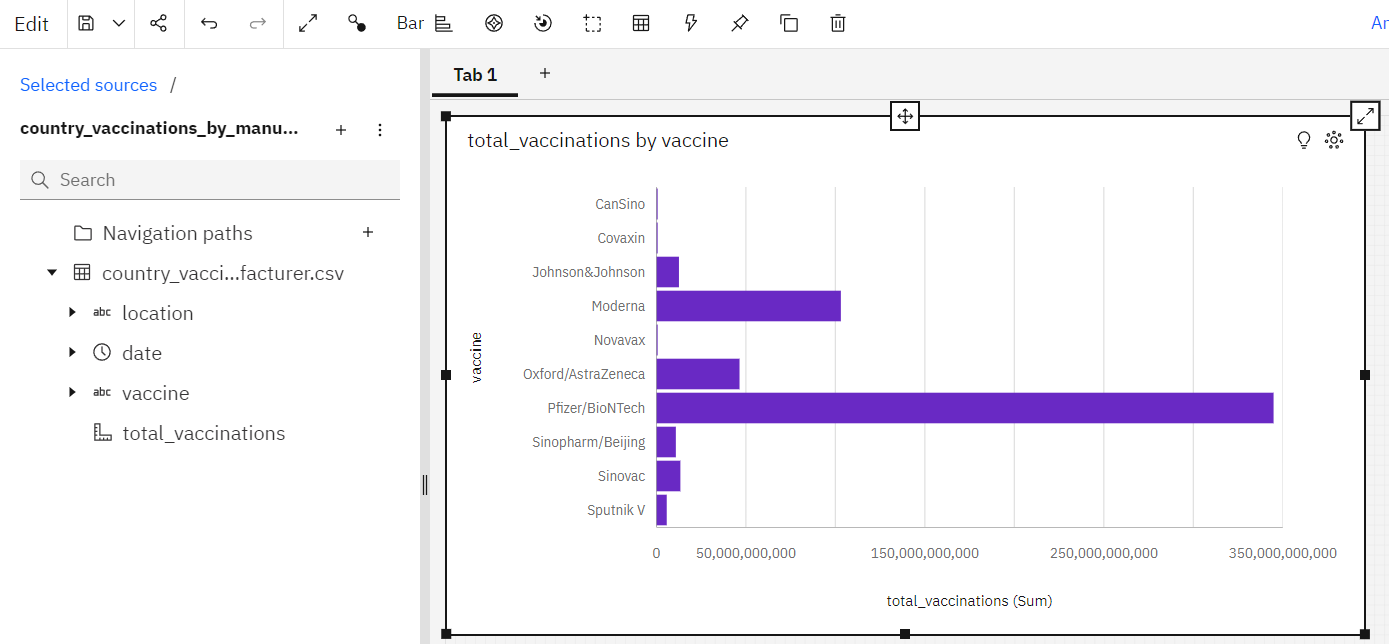
1. In the IBM Cognos project or report, select the "Total Vaccinations by Vaccine" visualization.

2. Choose a suitable chart type (e.g., bar chart or pie chart) to display the distribution of total vaccinations by vaccine.

3. Configure the chart to use vaccines as categories on the X-axis and the total vaccinations as data points.

4. Customize labels, colors, and other chart settings to enhance readability.

5. Create the visualization to show how different vaccines contribute to total vaccinations.

****

**Step 5.4: Visualize Total Vaccinations and People Vaccinated**

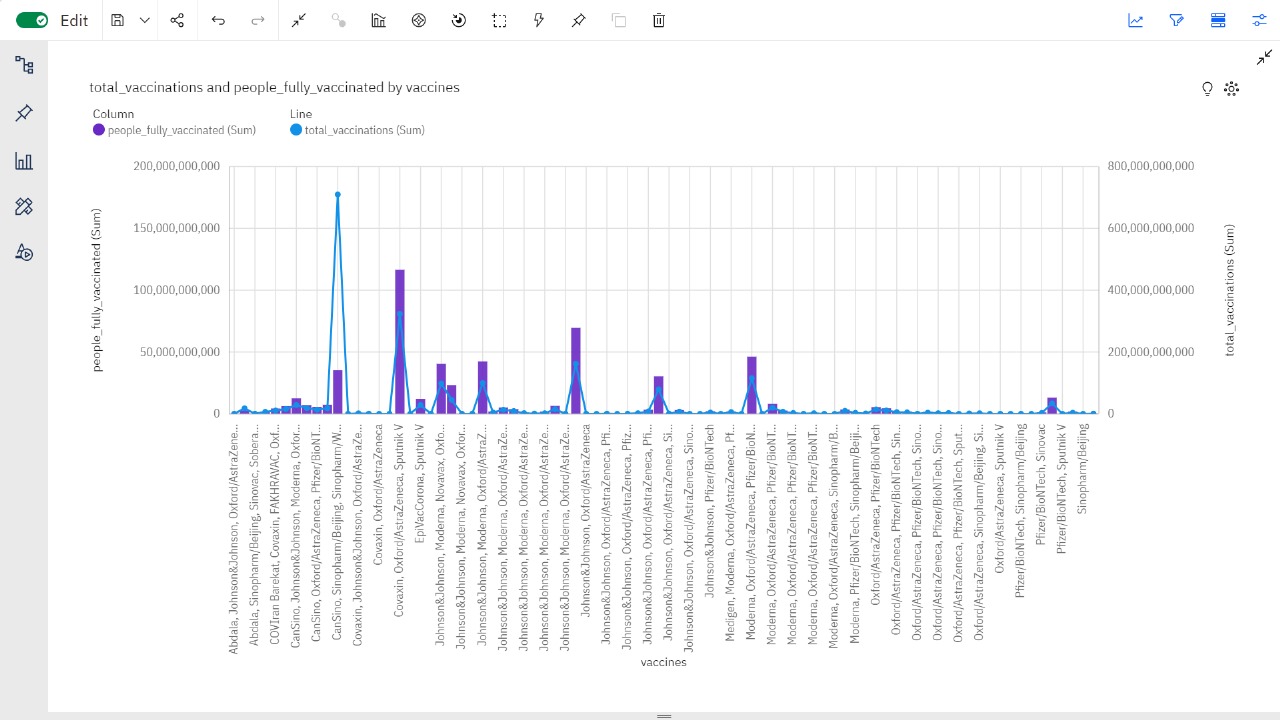
1. Within the project or report in IBM Cognos Analytics, pick the "Total Vaccinations and People Vaccinated" visualization.

2. Decide on the chart type that best represents the relationship between total vaccinations and people vaccinated (e.g., scatter plot, line chart, or bar chart).

3. Set the X-axis for total vaccinations and the Y-axis for people vaccinated.

4. Fine-tune the chart appearance and labels for clarity.

5. Generate the visualization to illustrate how total vaccinations relate to people vaccinated.



**Step 5.5: Visualize Country vs. People Fully Vaccinated**

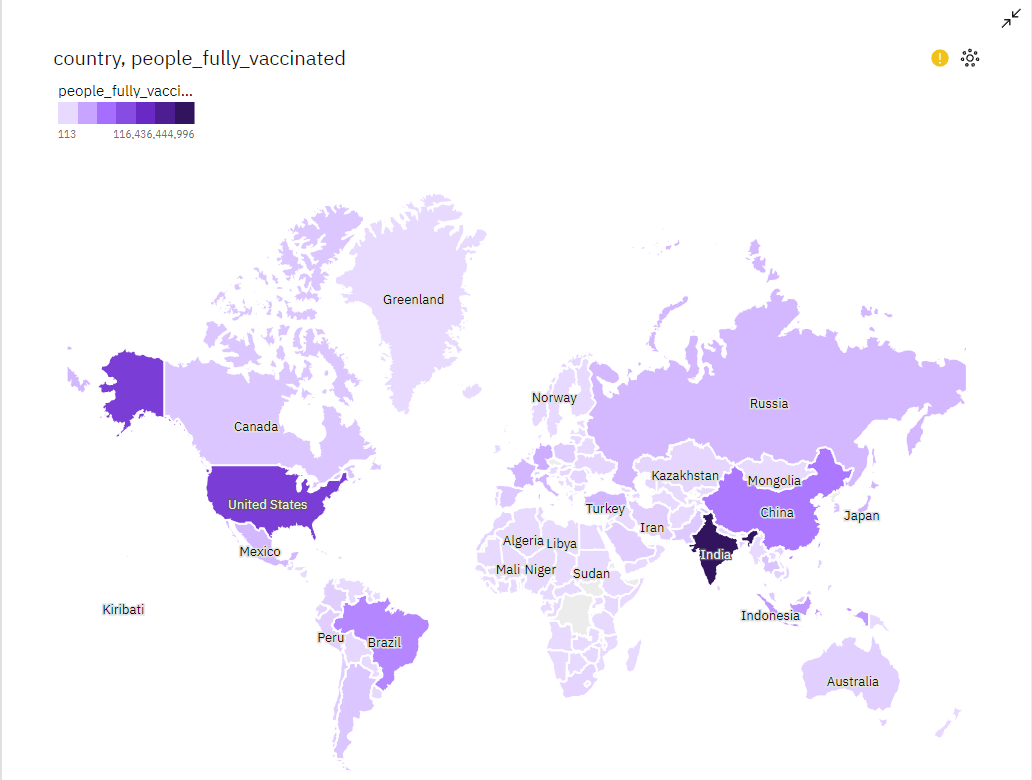
1. In the IBM Cognos project or report, choose the "Country vs. People Fully Vaccinated" visualization.

2. Select a chart type that effectively compares countries and the number of people who are fully vaccinated (e.g., bar chart or stacked bar chart).

3. Configure the X-axis for countries and the Y-axis for the number of fully vaccinated individuals.

4. Customize labels, legends, and colors for better interpretation.

5. Create the visualization to demonstrate the status of people fully vaccinated in different countries.

****

The visualizations are now ready to provide insights based on the dataset. We can use these visualizations to explore trends, patterns, and relationships within the data and communicate the findings effectively. Adjust the visualizations as needed to convey the most relevant information for the audience.