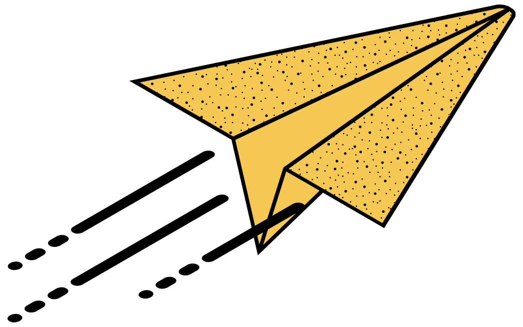
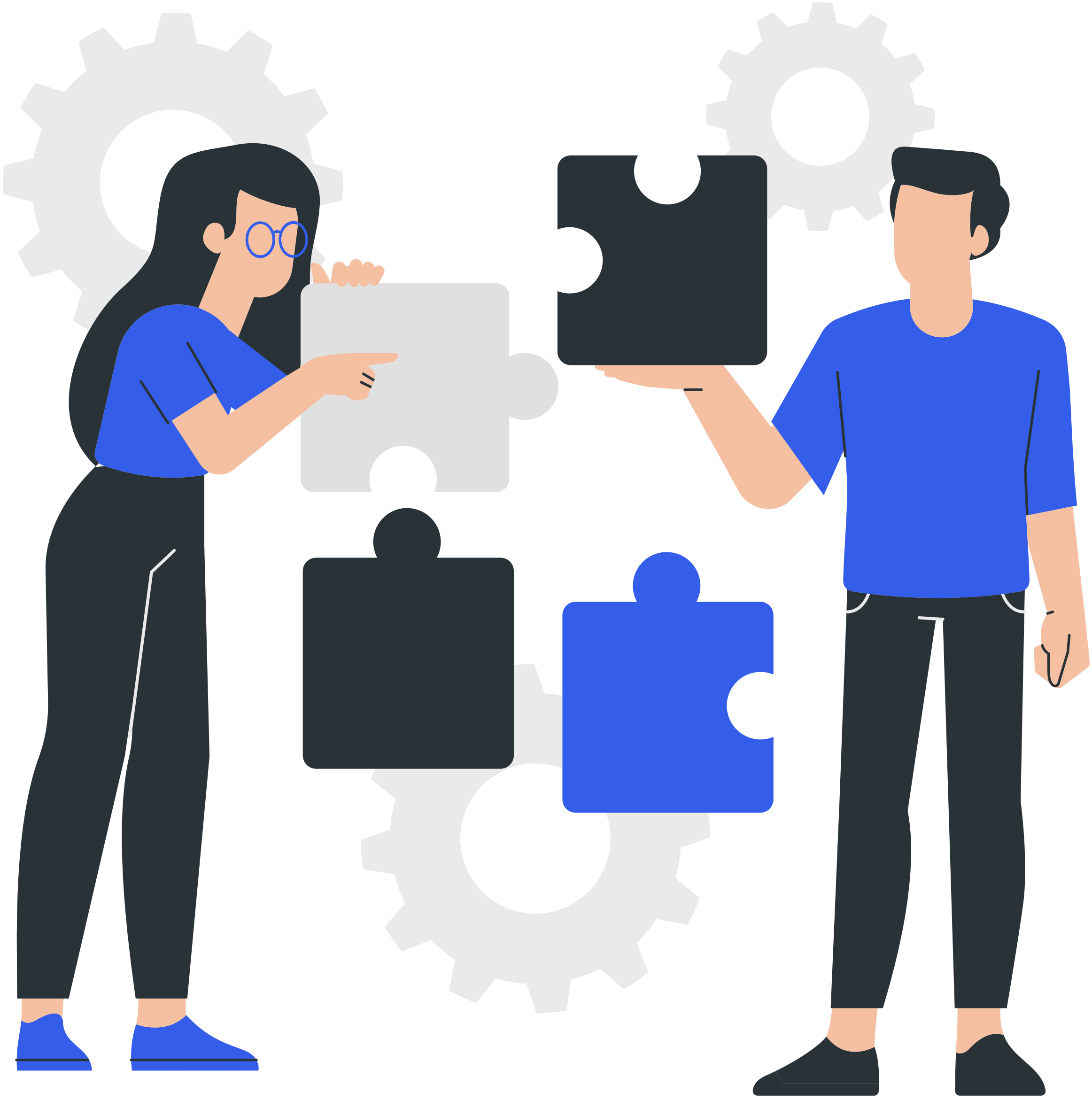


DATA PREPARATION



The First Step to Accurate Data Analysis.

Noorhan Yasin
2024-MS-CDS-005

TABLE OF CONTENTS

1. Introduction
2. Programming Language
3. Data Preparation Steps
4. Conclusion
5. Key Takeaways

DATA PREPARATION

VS

DATA PREPROCESSING

WHAT IS DATA PREPARATION?

- **Definition**: Data preparation is the first and crucial step in the data science pipeline.
- **Purpose**: It involves cleaning, transforming, and organizing raw data into a usable form for analysis or modeling.

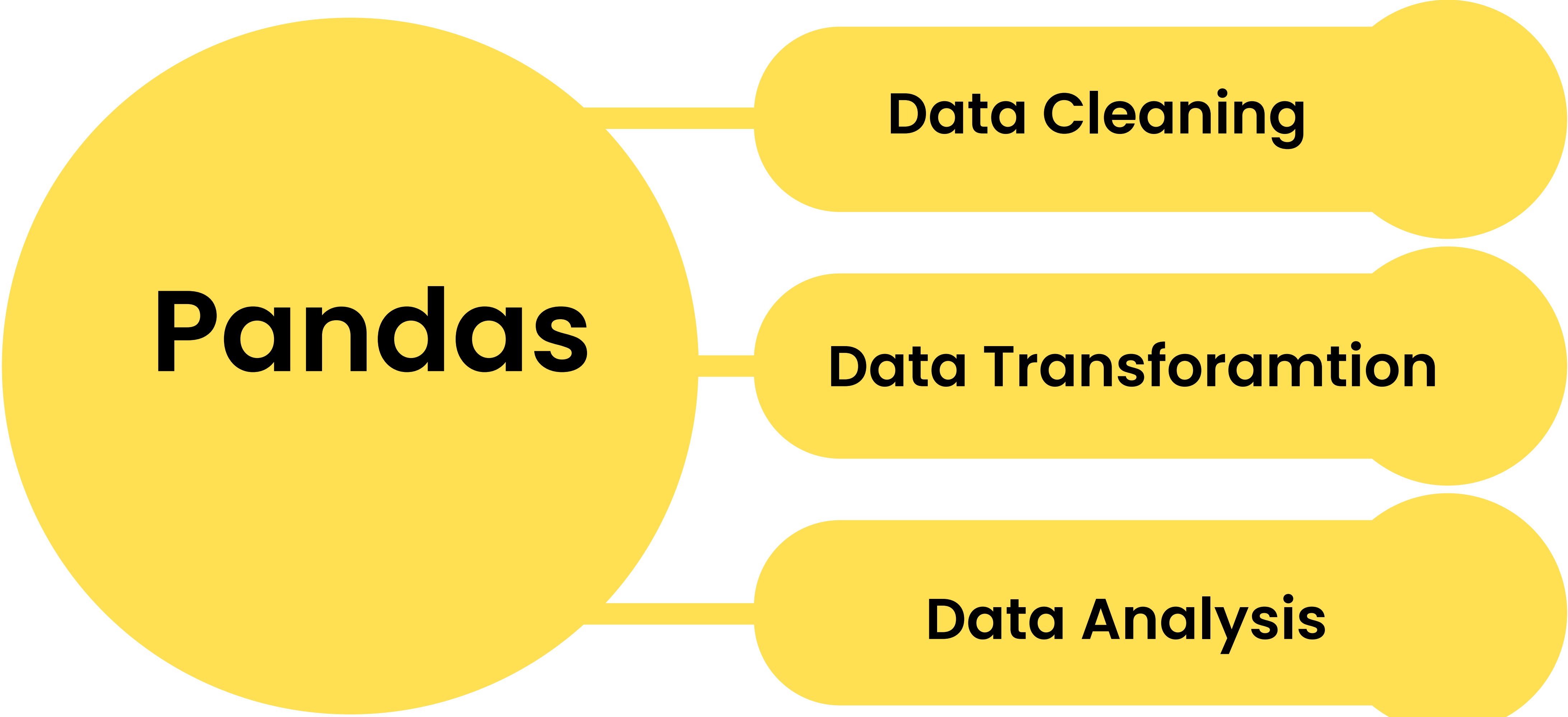
WHAT IS DATA PREPARATION?

- **Includes:**
- Handling missing values
- Data type conversions
- Data cleaning and removal of duplicates

BASICS

Series	DataFrame
Operations are typically applied to individual columns (single data points).	Operations can be performed across multiple columns (like filtering or aggregating data)

DATA PREPARATION IN PYTHON



Pandas

Data Cleaning

Data Transforamtion

Data Analysis

Use of Pandas in Data Preparation

Pandas is one of the most widely used libraries in Python for data manipulation and preparation. It offers powerful tools and data structures, particularly the DataFrame, which is ideal for handling structured data in the form of tables or spreadsheets.

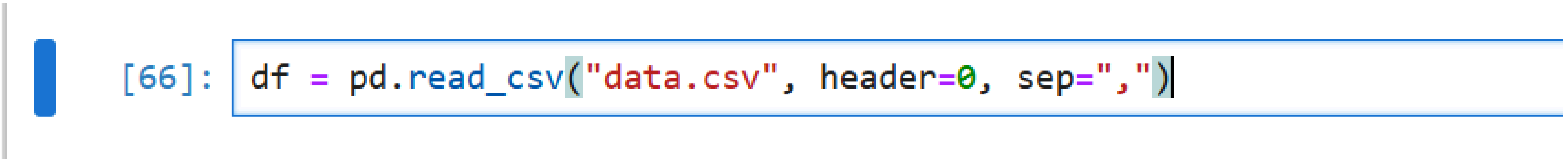
CAR DATA SET

Car	Model	Volume	Weight	CO2
Toyoty	Aygo	1000	790	99
Mitsubishi	Space Star	1200	1160	95
Skoda	Citigo	1000	929	95
Mini	Cooper	1500	1140	105
VW	Up!	1000	929	105
Skoda	Fabia	1400	1109	90
Mercedes	A-Class	1500	1365	92
Ford	Fiesta	1500	1112	98

1. DATA LOADING AND IMPORTING

Pandas allows you to easily load data from various file formats, such as CSV, Excel, SQL databases, and more.

- **Function:** pd.read_csv() / pd.read_excel()



```
df = pd.read_csv("data.csv", header=0, sep=",")
```

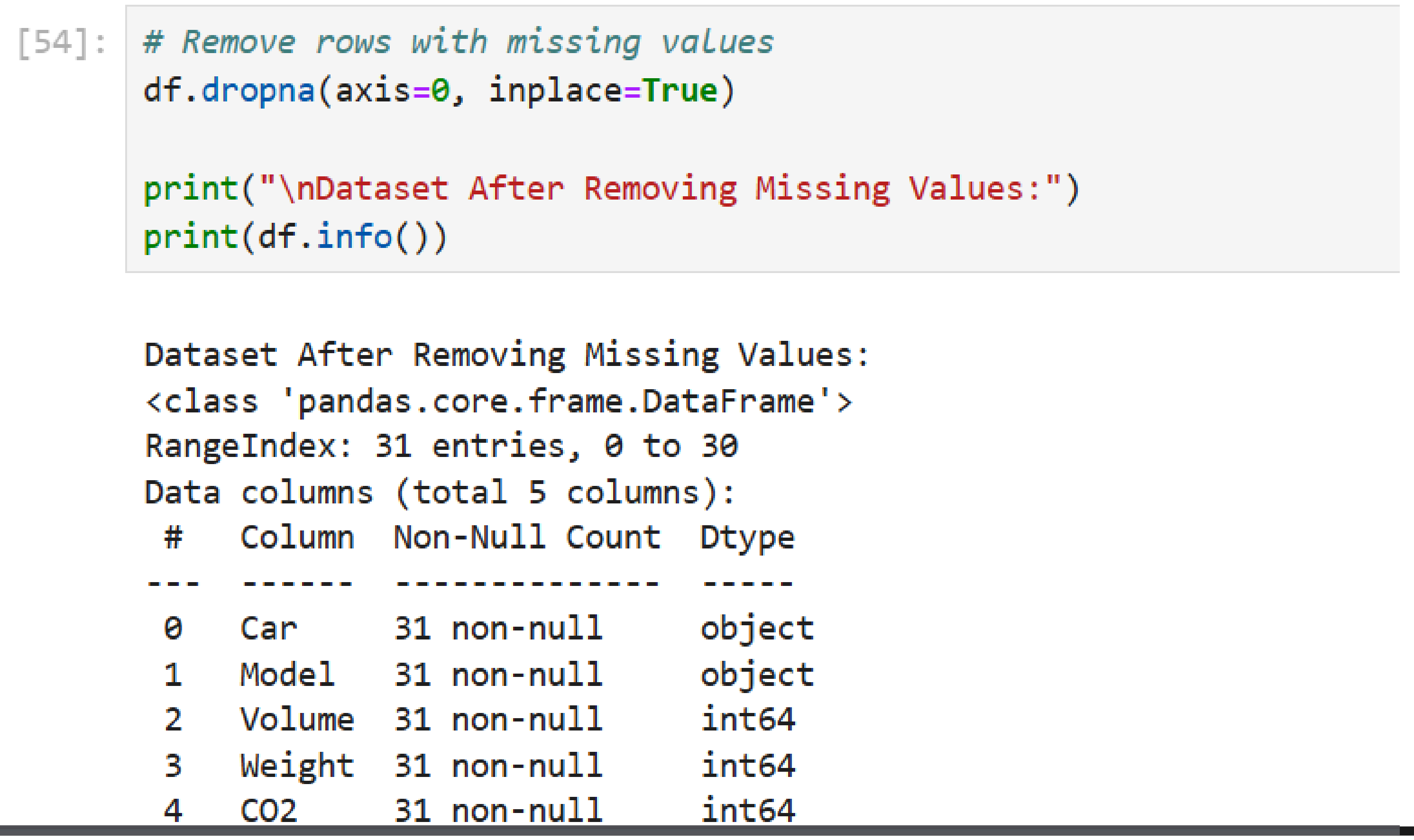
2. DATA CLEANING

Data cleaning is one of the most essential aspects of data preparation. With Pandas, we can easily identify and clean unwanted, missing, or corrupted data.

Handling Missing Data:

- Drop missing values: df.dropna() removes rows with missing data.
- Fill missing values: df.fillna() fills missing data with a specific value or method (e.g., forward fill).

2. DATA CLEANING



[54]:

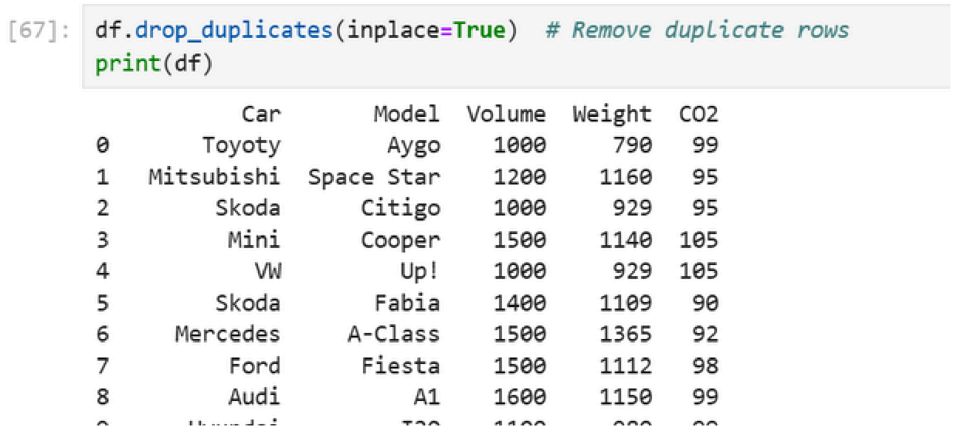
```
# Remove rows with missing values
df.dropna(axis=0, inplace=True)

print("\nDataset After Removing Missing Values:")
print(df.info())
```

```
Dataset After Removing Missing Values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Car     31 non-null     object
 1   Model   31 non-null     object
 2   Volume  31 non-null     int64 
 3   Weight  31 non-null     int64 
 4   CO2     31 non-null     int64 
```

3. HANDLING DUPLICATE DATA

Pandas makes it easy to identify and remove duplicate rows from the dataset.

- **Function:** df.drop_duplicates()



```
[67]: df.drop_duplicates(inplace=True)  # Remove duplicate rows
      print(df)
```

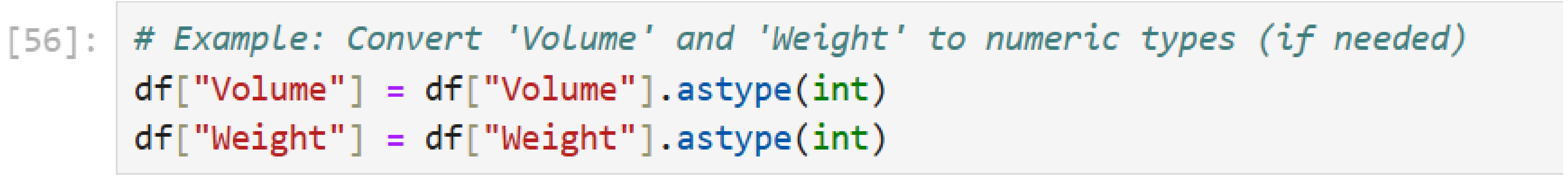
```
          Car       Model  Volume  Weight  CO2
0      Toyoty        Aygo    1000     790   99
1  Mitsubishi  Space Star    1200    1160   95
2       Skoda      Citigo    1000     929   95
3        Mini      Cooper    1500    1140  105
4          VW         Up!    1000     929  105
5       Skoda       Fabia    1400    1109   90
6    Mercedes     A-Class    1500    1365   92
7        Ford      Fiesta    1500    1112   98
8        Audi          A1    1600    1150   99
```

4.DATA TRANSFORMATION

Pandas allows you to transform data by modifying existing columns, creating new features, or changing data types. These operations are essential for preparing the data for analysis or modeling.

- **Changing Data Types:** Convert columns to the appropriate data type (e.g., from object to float or integer).
- **Function:** df.astype()

4.DATA TRANSFORMATION

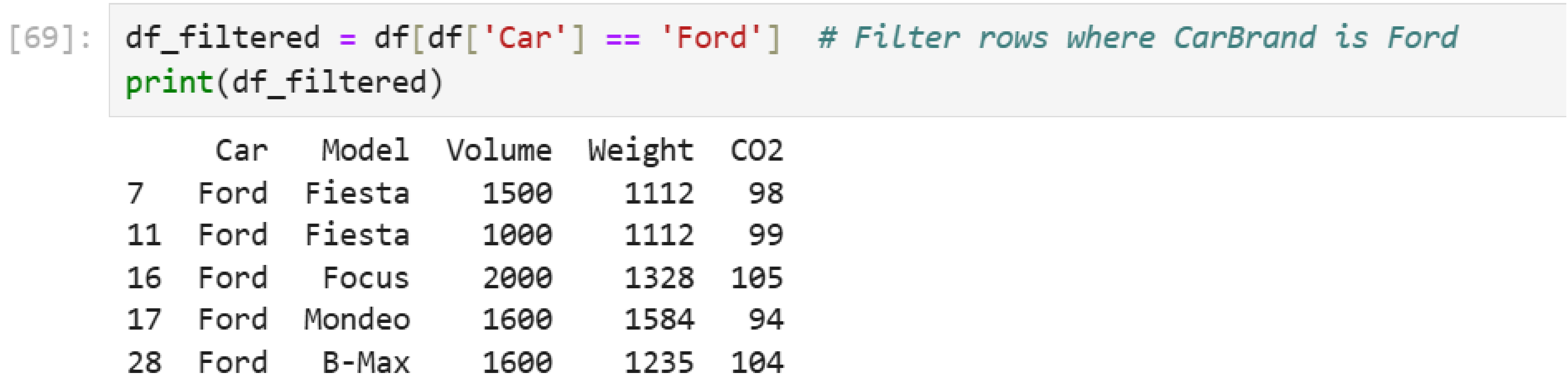


[56]:

```python
# Example: Convert 'Volume' and 'Weight' to numeric types (if needed)
df["Volume"] = df["Volume"].astype(int)
df["Weight"] = df["Weight"].astype(int)
```

5. FILTERING AND SELECTING DATA

Pandas allows you to filter and select rows or columns based on specific conditions.



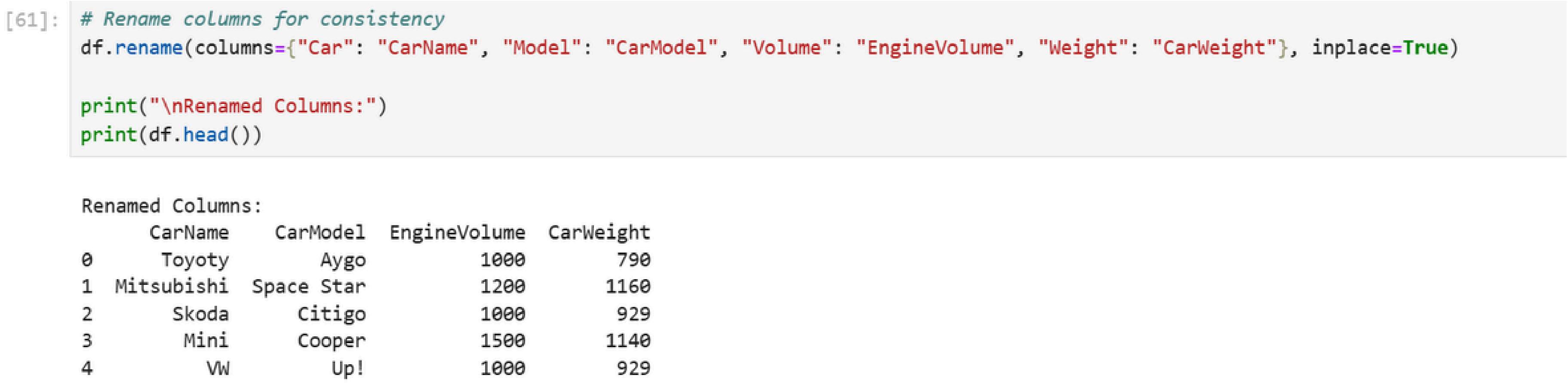
```
[69]: df_filtered = df[df['Car'] == 'Ford']  # Filter rows where CarBrand is Ford
      print(df_filtered)
```

```
     Car   Model  Volume  Weight  CO2
7   Ford  Fiesta    1500    1112   98
11  Ford  Fiesta    1000    1112   99
16  Ford   Focus    2000    1328  105
17  Ford  Mondeo    1600    1584   94
28  Ford   B-Max    1600    1235  104
```

6. RENAMING COLUMNS

Often, raw data may have inconsistent or unclear column names. Pandas makes it easy to rename columns to ensure clarity.

- **Function:** df.rename()



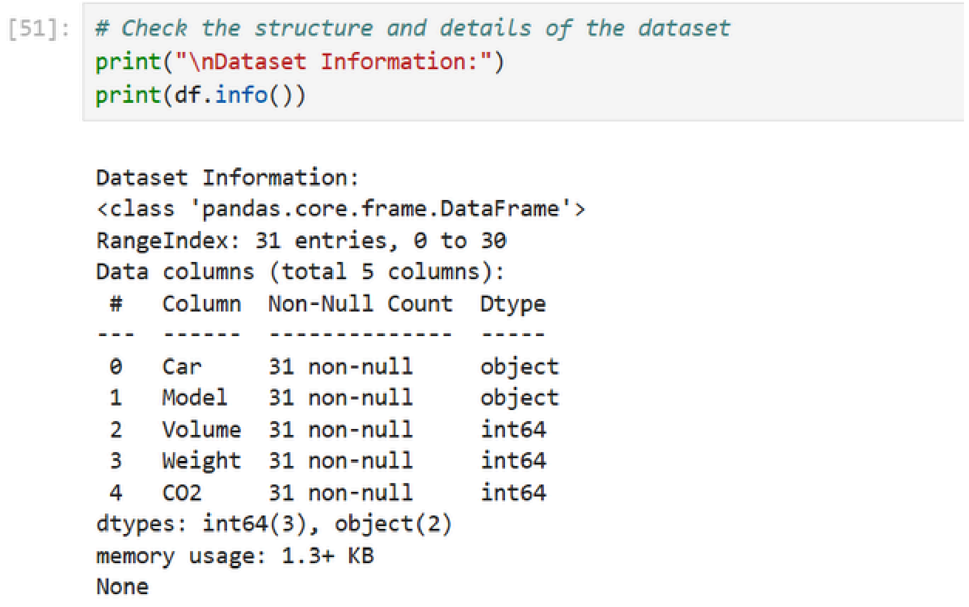
```
[61]: # Rename columns for consistency
      df.rename(columns={"Car": "CarName", "Model": "CarModel", "Volume": "EngineVolume", "Weight": "CarWeight"}, inplace=True)

      print("\nRenamed Columns:")
      print(df.head())
```

```
Renamed Columns:
      CarName    CarModel  EngineVolume  CarWeight
0      Toyoty        Aygo          1000        790
1  Mitsubishi  Space Star          1200       1160
2       Skoda      Citigo          1000        929
3        Mini      Cooper          1500       1140
4          VW         Up!          1000        929
```

7.SUMMARIZING DATA

Pandas provides methods to summarize data, which is useful for obtaining insights from the data before analysis.

Function: df.info()



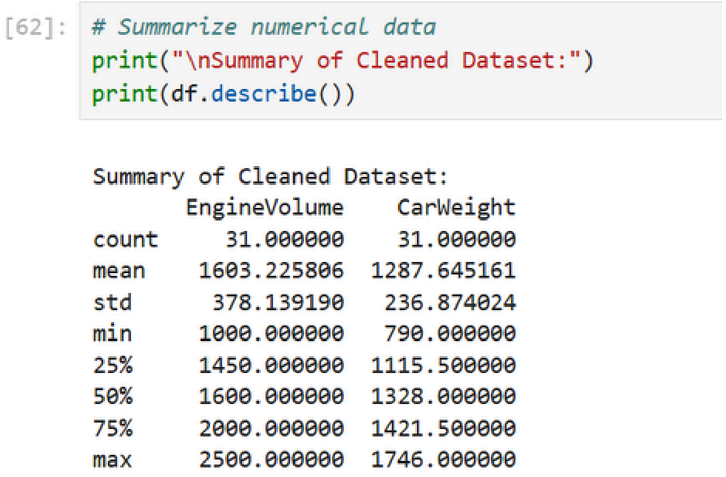
```
# Check the structure and details of the dataset
print("\nDataset Information:")
print(df.info())
```

```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Car     31 non-null     object
 1   Model   31 non-null     object
 2   Volume  31 non-null     int64 
 3   Weight  31 non-null     int64 
 4   CO2     31 non-null     int64 
dtypes: int64(3), object(2)
memory usage: 1.3+ KB
None
```

8.STATISTICAL SUMMARY

Pandas provides methods to summarize data, which is useful for obtaining insights from the data before analysis.

Function: df.describe()



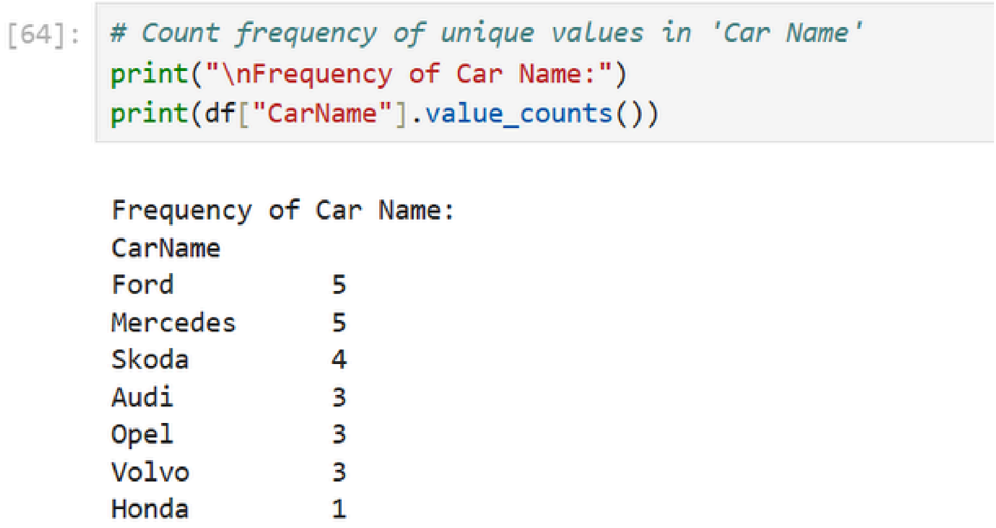
```
[62]: # Summarize numerical data
      print("\nSummary of Cleaned Dataset:")
      print(df.describe())
```

```
Summary of Cleaned Dataset:
       EngineVolume    CarWeight
count     31.000000    31.000000
mean    1603.225806  1287.645161
std      378.139190   236.874024
min     1000.000000   790.000000
25%     1450.000000  1115.500000
50%     1600.000000  1328.000000
75%     2000.000000  1421.500000
max     2500.000000  1746.000000
```

9.CHECKING DATA DISTRIBUTION

An optional step involves visualizing the distribution of numerical data or analyzing the frequency of categorical variable.

Function: value_counts())



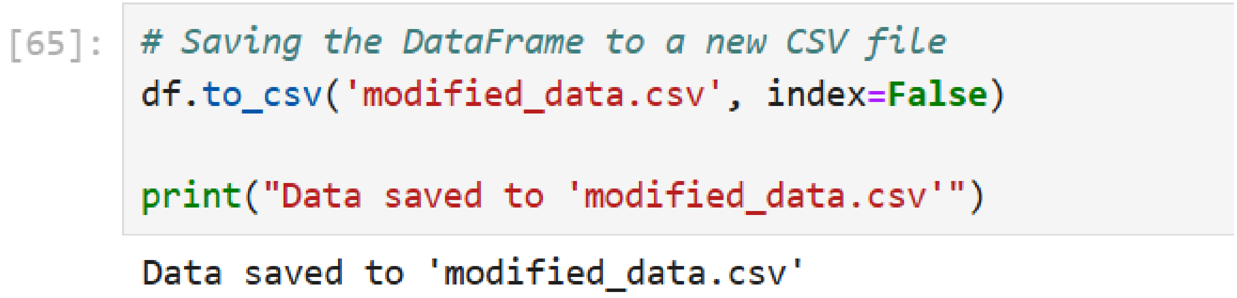
```
[64]: # Count frequency of unique values in 'Car Name'
      print("\nFrequency of Car Name:")
      print(df["CarName"].value_counts())
```

```
Frequency of Car Name:
CarName
Ford          5
Mercedes      5
Skoda         4
Audi          3
Opel          3
Volvo         3
Honda         1
```

10. SAVING THE PREPARED DATA

Once the data is cleaned, transformed, and ready for analysis, you can save it to various formats (e.g., CSV, Excel) for further use.

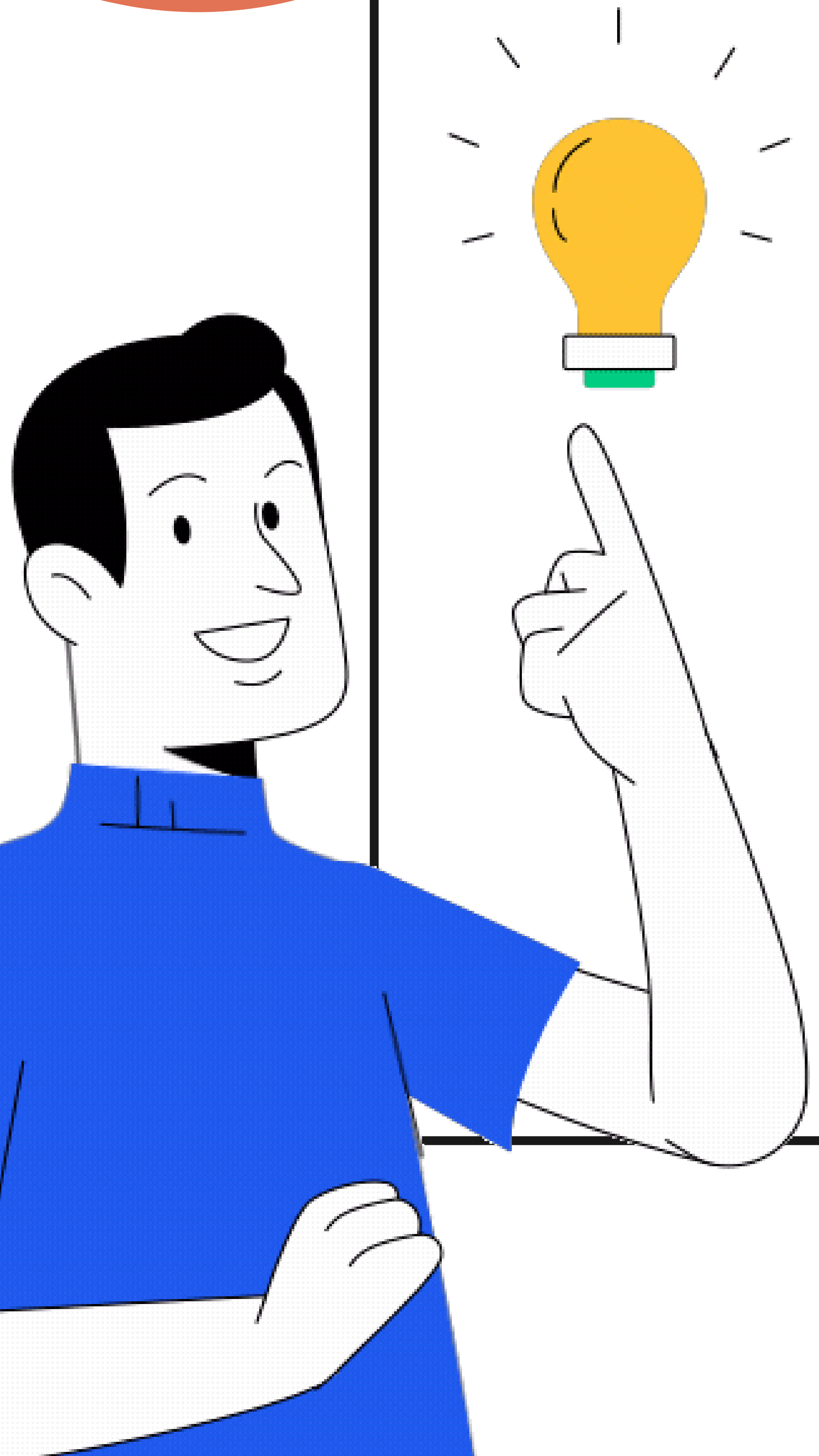
- **Function:** df.to_csv() / df.to_excel()



```
[65]: # Saving the DataFrame to a new CSV file
      df.to_csv('modified_data.csv', index=False)

      print("Data saved to 'modified_data.csv'")

      Data saved to 'modified_data.csv'
```

CONCLUSION

CONCLUSION



- **Data Preparation is Key:** Proper data preparation is a vital process in data science. It ensures that data is in the best format for further analysis, machine learning, and visualization.
- **Iterative Process:** Data preparation may require revisiting steps based on the complexity and issues encountered in the dataset.

KEY TAKEAWAYS

Data Cleaning

- Removing or imputing missing data, correcting data types.

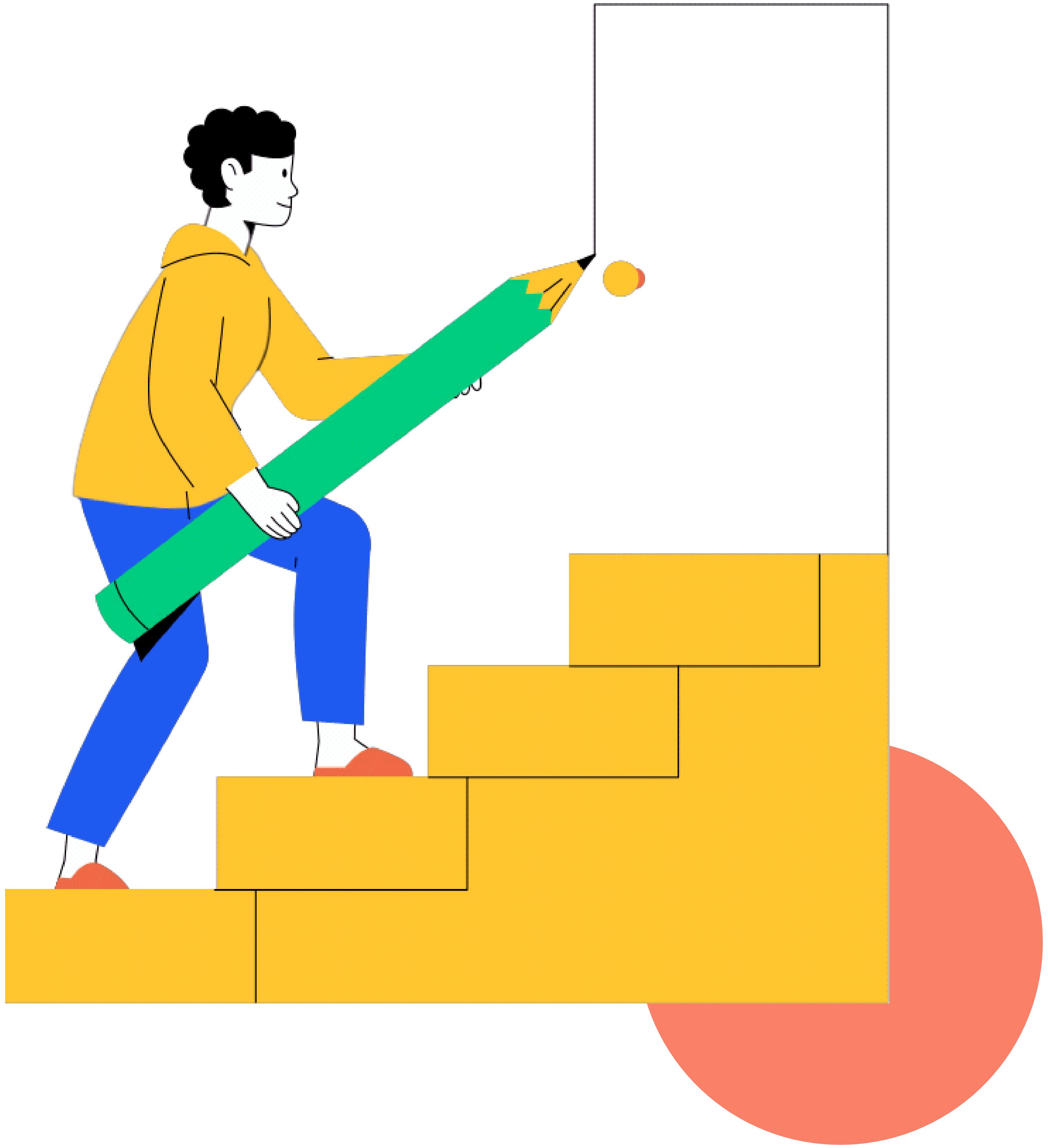
Data Transformation

- Aggregating, renaming, and organizing data into a usable format.

Data Summary

- Descriptive statistics help understand the dataset before deeper analysis.

THANK YOU!



REFERENCES

- **Kaggle** - www.kaggle.com
- **Towards Data Science** - www.towardsdatascience.com
- **Pandas Documentation** - https://pandas.pydata.org
- **DataCamp** - www.datacamp.com
- **Analytics Vidhya** - www.analyticsvidhya.com
- **GeeksforGeeks** - www.geeksforgeeks.org
- **Medium (Data Science Section)** - https://medium.com/topic/data-science