Data Preprocessing & Exploratory Analysis

This presentation will cover the process of **Data Acquisition & Wrangling** using **Python** and **Key Libraries** including **Pandas**, **NumPy**, **Matplotlib**, **Seaborn**, and **Scikit-learn (SKLearn)**.

In this project, I used **Pandas** to organize & explore the data easily. **NumPy** helped me with math calculations and fixing missing values. I checked the **Skewness** and **Correlation** of the data using **Seaborn** to create visualizations that showed how different variables relate to each other.

To handle **missing values** & **outliers**, I used **Scikit-learn** & its **SimpleImputer** tool to fill in gaps. I also looked for outliers with **charts** from **Matplotlib** and **Seaborn**.

Overall, this project brought all these tools together to help me analyze the data, check for skewness and correlation, fix missing values, and manage outliers, leading to some really valuable insights!

by Debasis Baidya

Importing of the necessary Libraries

Pandas

To load & organize the datasets, making it easy to explore the data structure and perform operations like merging and filtering.

Library 1

Numpy

It helped with mathematical operations, allowing me to compute statistics and handle missing values efficiently.

Library 2

Matplotlib

Used it to create visualizations, such as bar charts and histograms, to better understand the data distributions and trends.

Library 3

Seaborn

For generating more advanced visualizations, helping me check for skewness & visualize correlations between different variables.

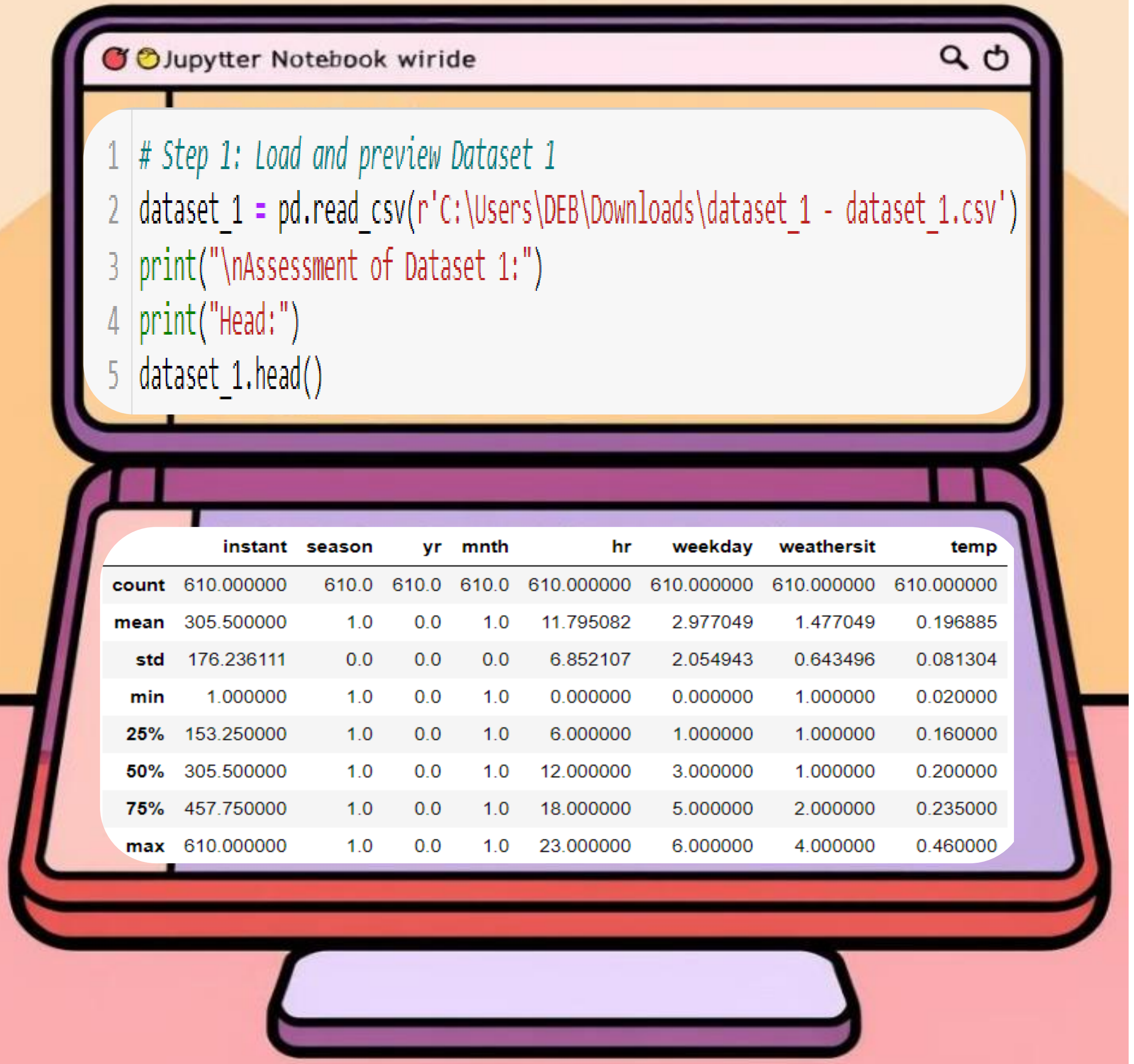
Library 4

Scikit-learn (Sklearn)

For preprocessing the data, specifically using the **SimpleImputer** to fill in missing values and to identify and manage outliers effectively.

Library 5

Load and Preview Dataset 1



```
# Step 1: Load and preview Dataset 1
dataset_1 = pd.read_csv(r'C:\Users\DEB\Downloads\dataset_1 - dataset_1.csv')
print("\nAssessment of Dataset 1:")
print("Head:")
dataset_1.head()
```

	instant	season	yr	mnth	hr	weekday	weathersit	temp
count	610.000000	610.0	610.0	610.0	610.000000	610.000000	610.000000	610.000000
mean	305.500000	1.0	0.0	1.0	11.795082	2.977049	1.477049	0.196885
std	176.236111	0.0	0.0	0.0	6.852107	2.054943	0.643496	0.081304
min	1.000000	1.0	0.0	1.0	0.000000	0.000000	1.000000	0.020000
25%	153.250000	1.0	0.0	1.0	6.000000	1.000000	1.000000	0.160000
50%	305.500000	1.0	0.0	1.0	12.000000	3.000000	1.000000	0.200000
75%	457.750000	1.0	0.0	1.0	18.000000	5.000000	2.000000	0.235000
max	610.000000	1.0	0.0	1.0	23.000000	6.000000	4.000000	0.460000

1

Loaded Dataset 1

Imported the CSV file and displayed the head, tail, shape, column names, and data types.

2

Analyzed Dataset 1

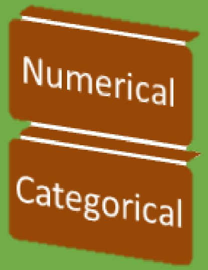
Examined the dataset's structure, including null values and unique value counts.

3

Descriptive Statistics

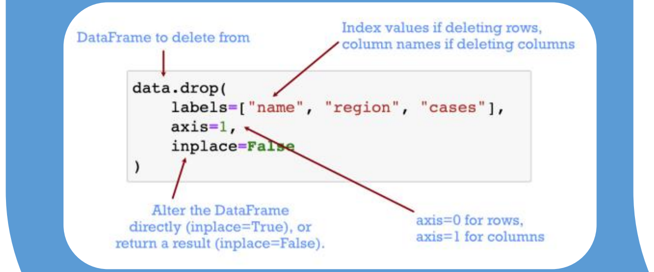
Compute the mean, median, and mode for the numerical columns in Dataset 1.

Data Preprocessing for Dataset 1



Identify Data Types

Determined the appropriate data types for each column in Dataset 1



DataFrame to delete from

Index values if deleting rows, column names if deleting columns

```
data.drop(
    labels=["name", "region", "cases"],
    axis=1,
    inplace=False
)
```

Alter the DataFrame directly (inplace=True), or return a result (inplace=False).

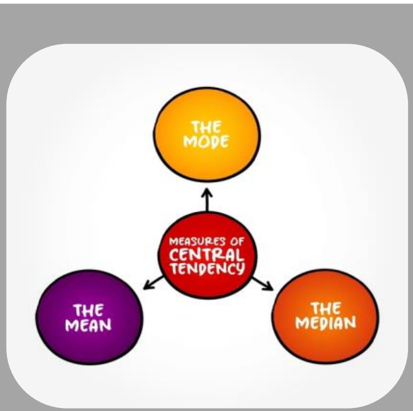
axis=0 for rows, axis=1 for columns

Dropping Columns

Didn’t drop any columns in dataset_1 as all columns appeared important for analysis

Handle Null Values

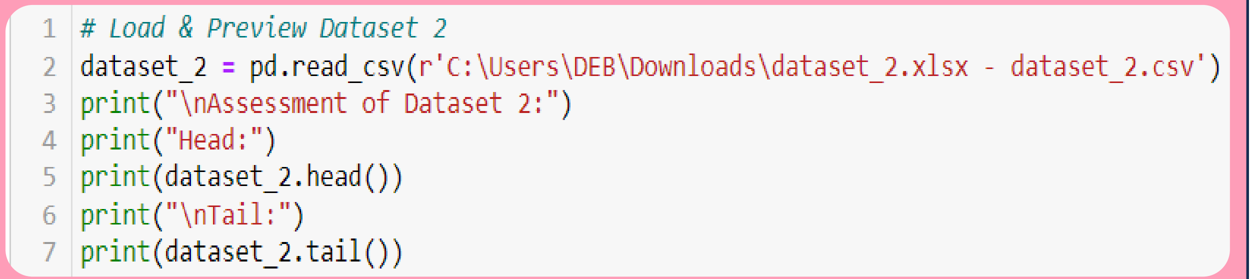
Imputed numerical columns by importing the SimpleImputer method from sklearn.impute



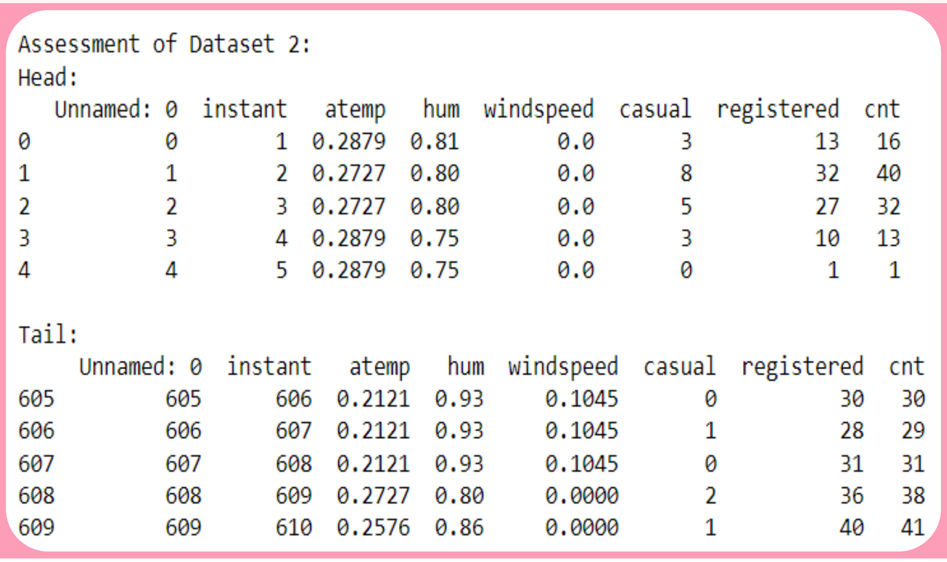
Central tendency for Dataset 1

Computed the mean, median, and mode for the numerical columns in Dataset 1 from the Pandas Library

Exploring of Dataset 2

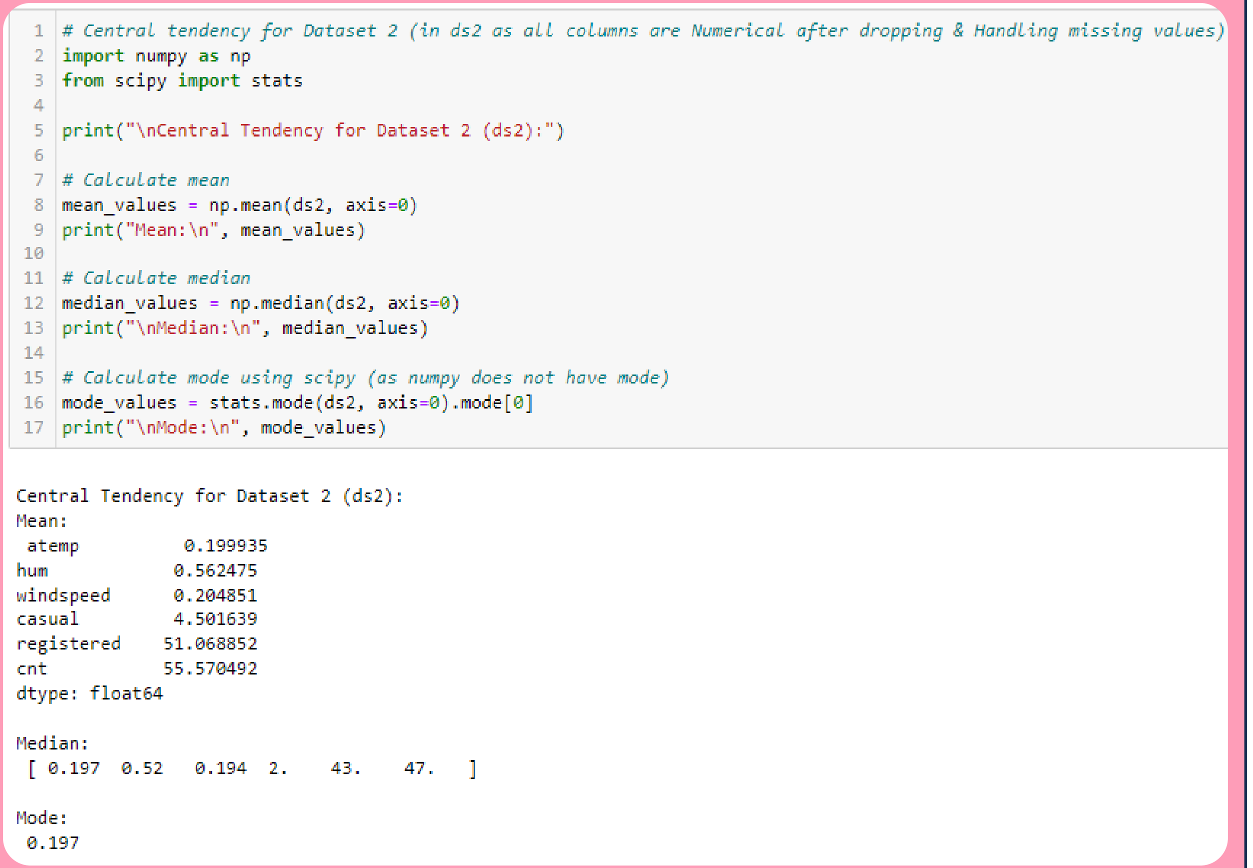


```python
# Load & Preview Dataset 2
dataset_2 = pd.read_csv(r'C:\Users\DEB\Downloads\dataset_2.xlsx - dataset_2.csv')
print("\nAssessment of Dataset 2:")
print("Head:")
print(dataset_2.head())
print("\nTail:")
print(dataset_2.tail())
```



```
Assessment of Dataset 2:
Head:
   Unnamed: 0  instant   atemp   hum  windspeed  casual  registered  cnt
0           0        1  0.2879  0.81        0.0       3          13   16
1           1        2  0.2727  0.80        0.0       8          32   40
2           2        3  0.2727  0.80        0.0       5          27   32
3           3        4  0.2879  0.75        0.0       3          10   13
4           4        5  0.2879  0.75        0.0       0           1    1

Tail:
     Unnamed: 0  instant   atemp   hum  windspeed  casual  registered  cnt
605         605      606  0.2121  0.93     0.1045       0          30   30
606         606      607  0.2121  0.93     0.1045       1          28   29
607         607      608  0.2121  0.93     0.1045       0          31   31
608         608      609  0.2727  0.80     0.0000       2          36   38
609         609      610  0.2576  0.86     0.0000       1          40   41
```



```python
# Central tendency for Dataset 2 (in ds2 as all columns are Numerical after dropping & Handling missing values)
import numpy as np
from scipy import stats

print("\nCentral Tendency for Dataset 2 (ds2):")

# Calculate mean
mean_values = np.mean(ds2, axis=0)
print("Mean:\n", mean_values)

# Calculate median
median_values = np.median(ds2, axis=0)
print("\nMedian:\n", median_values)

# Calculate mode using scipy (as numpy does not have mode)
mode_values = stats.mode(ds2, axis=0).mode[0]
print("\nMode:\n", mode_values)
```

```
Central Tendency for Dataset 2 (ds2):
Mean:
 atemp          0.199935
hum            0.562475
windspeed      0.204851
casual         4.501639
registered    51.068852
cnt           55.570492
dtype: float64

Median:
 [ 0.197  0.52   0.194  2.     43.     47.   ]

Mode:
 0.197
```

1 Loading and Previewing

Imported the csv file and displayed the head, tail, shape, column names, and data types.

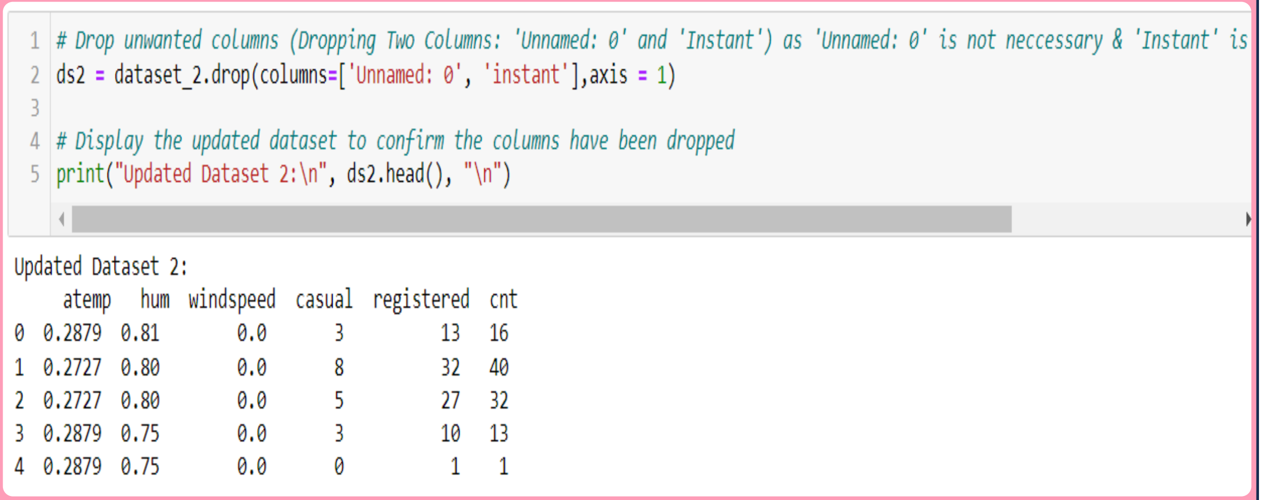
2 Analyzing Dataset 2

Examined the dataset's structure, including null values and unique value counts.

3 Descriptive Statistics

Computed the mean, median, and mode for the numerical columns in Dataset 2.

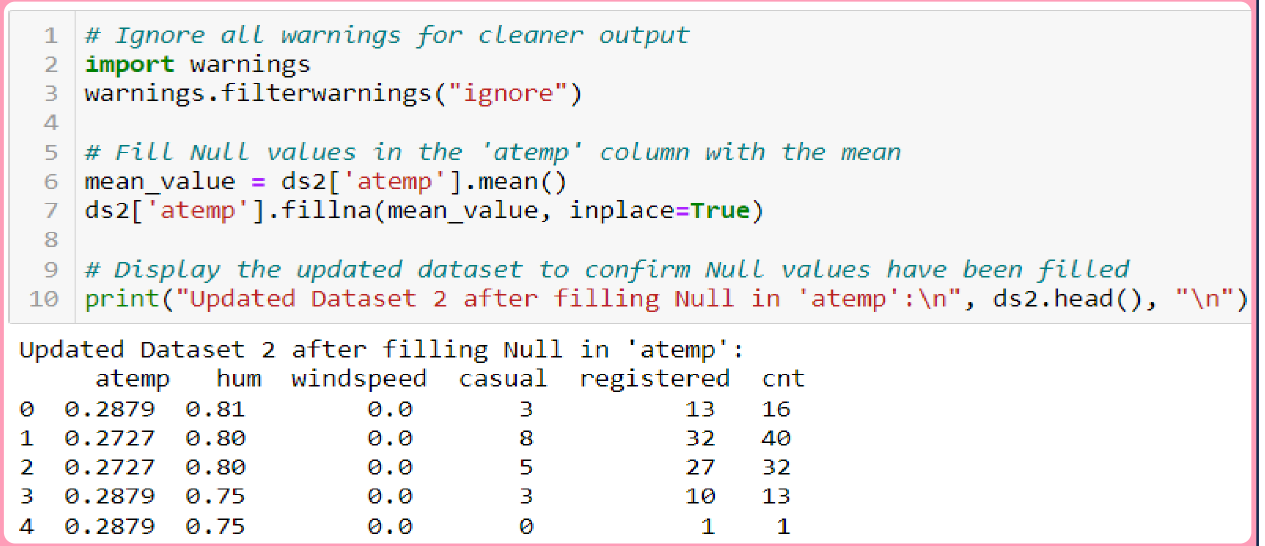
Data Preprocessing for Dataset 2



```python
# Drop unwanted columns (Dropping Two Columns: 'Unnamed: 0' and 'Instant') as 'Unnamed: 0' is not neccessary & 'Instant' is
ds2 = dataset_2.drop(columns=['Unnamed: 0', 'instant'],axis = 1)

# Display the updated dataset to confirm the columns have been dropped
print("Updated Dataset 2:\n", ds2.head(), "\n")
```

```
Updated Dataset 2:
     atemp   hum  windspeed  casual  registered  cnt
0  0.2879  0.81        0.0       3          13   16
1  0.2727  0.80        0.0       8          32   40
2  0.2727  0.80        0.0       5          27   32
3  0.2879  0.75        0.0       3          10   13
4  0.2879  0.75        0.0       0           1    1
```

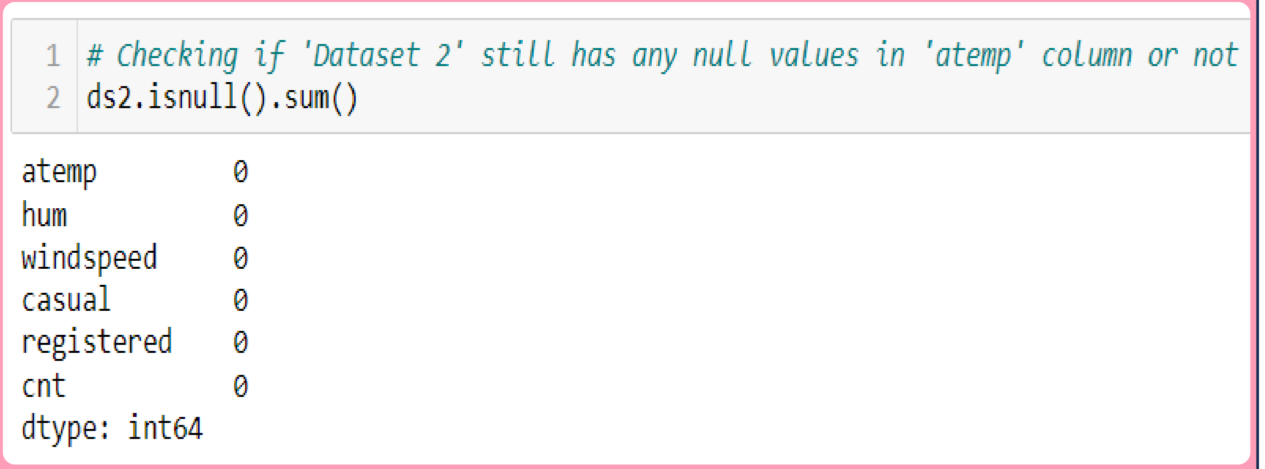


```python
# Ignore all warnings for cleaner output
import warnings
warnings.filterwarnings("ignore")

# Fill Null values in the 'atemp' column with the mean
mean_value = ds2['atemp'].mean()
ds2['atemp'].fillna(mean_value, inplace=True)

# Display the updated dataset to confirm Null values have been filled
print("Updated Dataset 2 after filling Null in 'atemp':\n", ds2.head(), "\n")
```

```
Updated Dataset 2 after filling Null in 'atemp':
     atemp   hum  windspeed  casual  registered  cnt
0  0.2879  0.81        0.0       3          13   16
1  0.2727  0.80        0.0       8          32   40
2  0.2727  0.80        0.0       5          27   32
3  0.2879  0.75        0.0       3          10   13
4  0.2879  0.75        0.0       0           1    1
```



```python
# Checking if 'Dataset 2' still has any null values in 'atemp' column or not
ds2.isnull().sum()
```

```
atemp         0
hum           0
windspeed     0
casual        0
registered    0
cnt           0
dtype: int64
```

Dropping Unwanted Columns

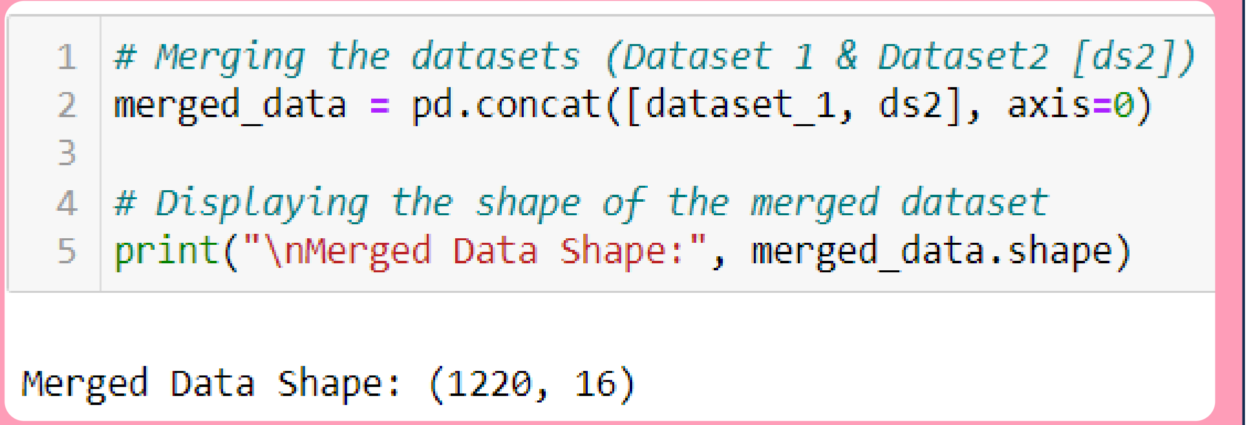
Removed the 'Unnamed: 0' and 'instant' columns from Dataset 2.

Handling Missing Values

Filled the 'atemp' column with the mean value.

Central Tendency

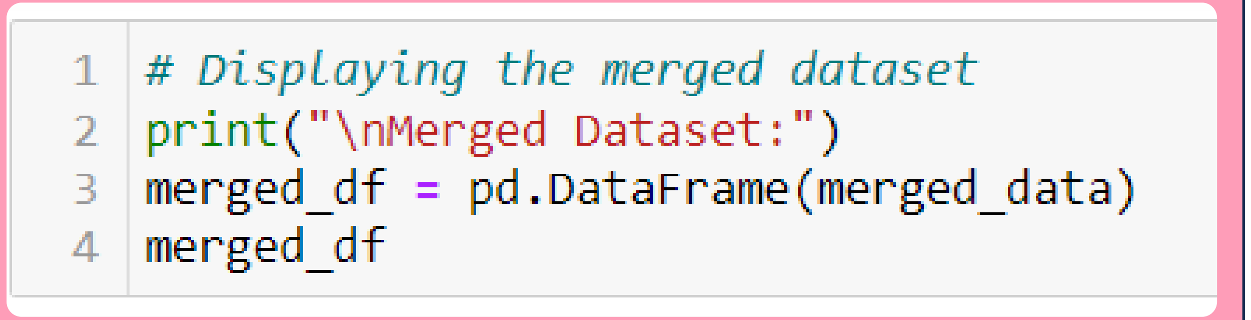
Calculated the mean, median, and mode for the numerical columns in Dataset 2.



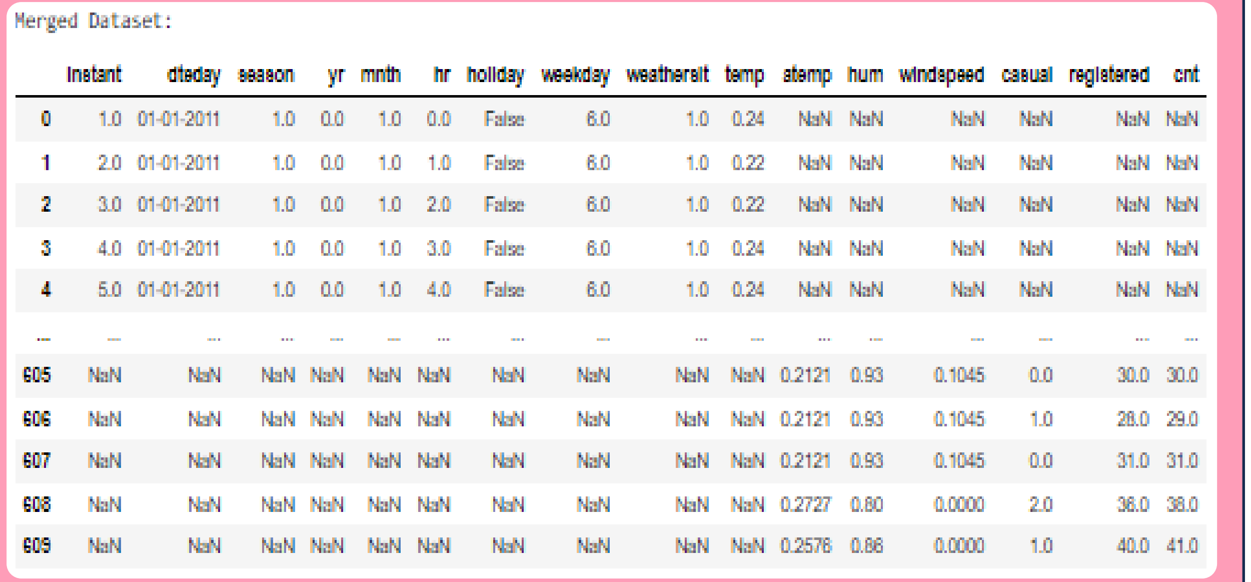
```
# Merging the datasets (Dataset 1 & Dataset2 [ds2])
merged_data = pd.concat([dataset_1, ds2], axis=0)

# Displaying the shape of the merged dataset
print("\nMerged Data Shape:", merged_data.shape)
```

```
Merged Data Shape: (1220, 16)
```



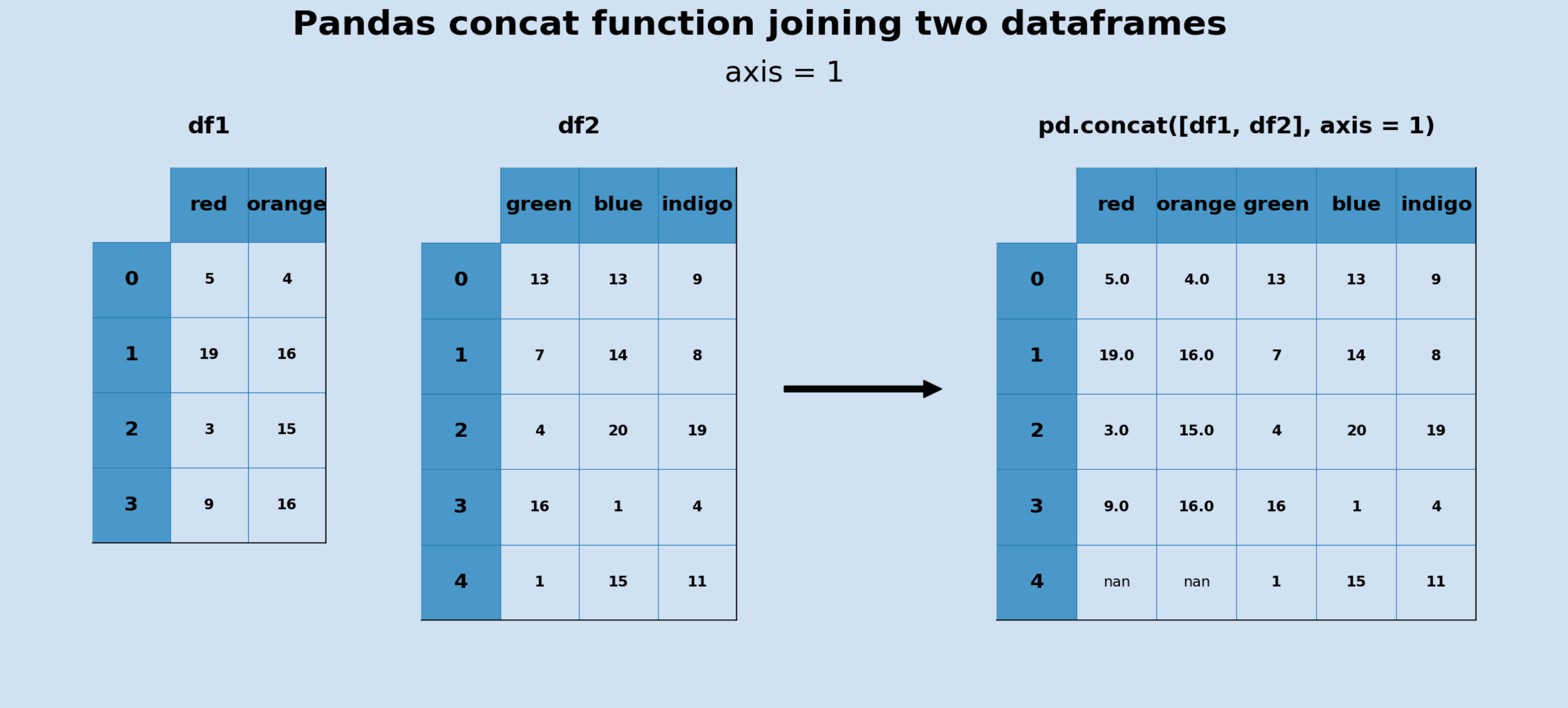
```
# Displaying the merged dataset
print("\nMerged Dataset:")
merged_df = pd.DataFrame(merged_data)
merged_df
```



```
Merged Dataset:
```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1.0	01-01-2011	1.0	0.0	1.0	0.0	False	6.0	1.0	0.24	NaN	NaN	NaN	NaN	NaN	NaN
1	2.0	01-01-2011	1.0	0.0	1.0	1.0	False	6.0	1.0	0.22	NaN	NaN	NaN	NaN	NaN	NaN
2	3.0	01-01-2011	1.0	0.0	1.0	2.0	False	6.0	1.0	0.22	NaN	NaN	NaN	NaN	NaN	NaN
3	4.0	01-01-2011	1.0	0.0	1.0	3.0	False	6.0	1.0	0.24	NaN	NaN	NaN	NaN	NaN	NaN
4	5.0	01-01-2011	1.0	0.0	1.0	4.0	False	6.0	1.0	0.24	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
605	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.2121	0.93	0.1045	0.0	30.0	30.0
606	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.2121	0.93	0.1045	1.0	28.0	29.0
607	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.2121	0.93	0.1045	0.0	31.0	31.0
608	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.2727	0.80	0.0000	2.0	36.0	38.0
609	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.2576	0.86	0.0000	1.0	40.0	41.0

Merging of Dataset 1 & Dataset 2



Pandas concat function joining two dataframes

axis = 1

df1

	red	orange
0	5	4
1	19	16
2	3	15
3	9	16

df2

	green	blue	indigo
0	13	13	9
1	7	14	8
2	4	20	19
3	16	1	4
4	1	15	11

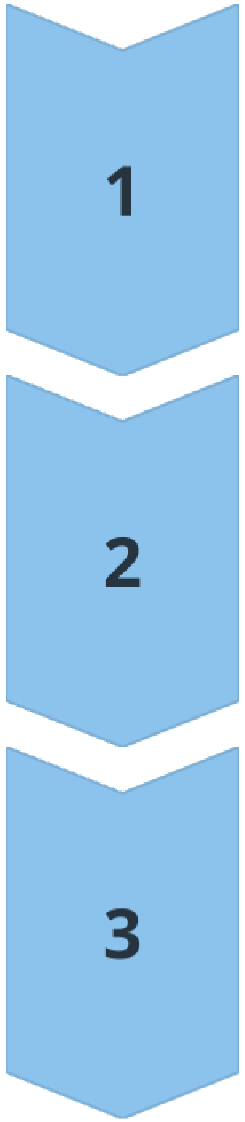
pd.concat([df1, df2], axis = 1)

	red	orange	green	blue	indigo
0	5.0	4.0	13	13	9
1	19.0	16.0	7	14	8
2	3.0	15.0	4	20	19
3	9.0	16.0	16	1	4
4	nan	nan	1	15	11

Combining Datasets

Concatenated Dataset 1 and the cleaned Dataset 2 (ds2) into a single merged dataset. This involves aligning the columns of both datasets based on shared features or key variables. The resulting merged dataset will provide a comprehensive view of the combined data, allowing for further analysis and insights.

Load and Preview Dataset 3



Loading Dataset 3

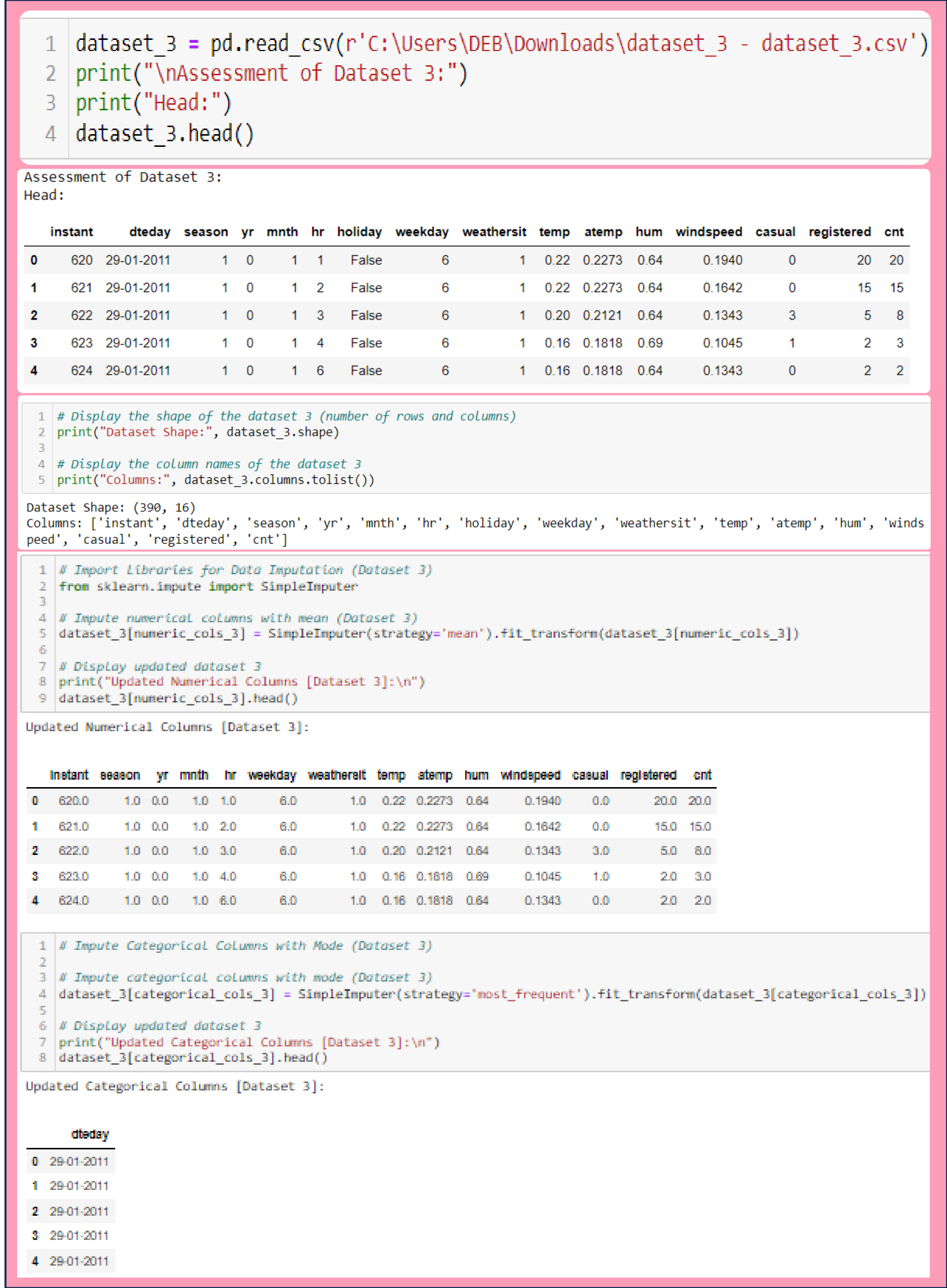
Imported the CSV file and displayed the head, tail, shape, and data types, describe.

Analyzing Dataset 3

Examined the dataset's structure and identified any missing values.

Prepared for Merging

Cleaned and preprocessed Dataset 3 to prepare it for combining with the already merged datasets (Dataset 1 & 2).



```
dataset_3 = pd.read_csv(r'C:\Users\DEB\Downloads\dataset_3 - dataset_3.csv')
print("\nAssessment of Dataset 3:")
print("Head:")
dataset_3.head()
```

```
Assessment of Dataset 3:
Head:
```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	620	29-01-2011	1	0	1	1	False	6	1	0.22	0.2273	0.64	0.1940	0	20	20
1	621	29-01-2011	1	0	1	2	False	6	1	0.22	0.2273	0.64	0.1642	0	15	15
2	622	29-01-2011	1	0	1	3	False	6	1	0.20	0.2121	0.64	0.1343	3	5	8
3	623	29-01-2011	1	0	1	4	False	6	1	0.16	0.1818	0.69	0.1045	1	2	3
4	624	29-01-2011	1	0	1	6	False	6	1	0.16	0.1818	0.64	0.1343	0	2	2

```
# Display the shape of the dataset 3 (number of rows and columns)
print("Dataset Shape:", dataset_3.shape)

# Display the column names of the dataset 3
print("Columns:", dataset_3.columns.tolist())
```

```
Dataset Shape: (390, 16)
Columns: ['instant', 'dteday', 'season', 'yr', 'mnth', 'hr', 'holiday', 'weekday', 'weathersit', 'temp', 'atemp', 'hum', 'winds
peed', 'casual', 'registered', 'cnt']
```

```
# Import Libraries for Data Imputation (Dataset 3)
from sklearn.impute import SimpleImputer

# Impute numerical columns with mean (Dataset 3)
dataset_3[numeric_cols_3] = SimpleImputer(strategy='mean').fit_transform(dataset_3[numeric_cols_3])

# Display updated dataset 3
print("Updated Numerical Columns [Dataset 3]:\n")
dataset_3[numeric_cols_3].head()
```

```
Updated Numerical Columns [Dataset 3]:
```

	instant	season	yr	mnth	hr	weekday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	620.0	1.0	0.0	1.0	1.0	6.0	1.0	0.22	0.2273	0.64	0.1940	0.0	20.0	20.0
1	621.0	1.0	0.0	1.0	2.0	6.0	1.0	0.22	0.2273	0.64	0.1642	0.0	15.0	15.0
2	622.0	1.0	0.0	1.0	3.0	6.0	1.0	0.20	0.2121	0.64	0.1343	3.0	5.0	8.0
3	623.0	1.0	0.0	1.0	4.0	6.0	1.0	0.16	0.1818	0.69	0.1045	1.0	2.0	3.0
4	624.0	1.0	0.0	1.0	6.0	6.0	1.0	0.16	0.1818	0.64	0.1343	0.0	2.0	2.0

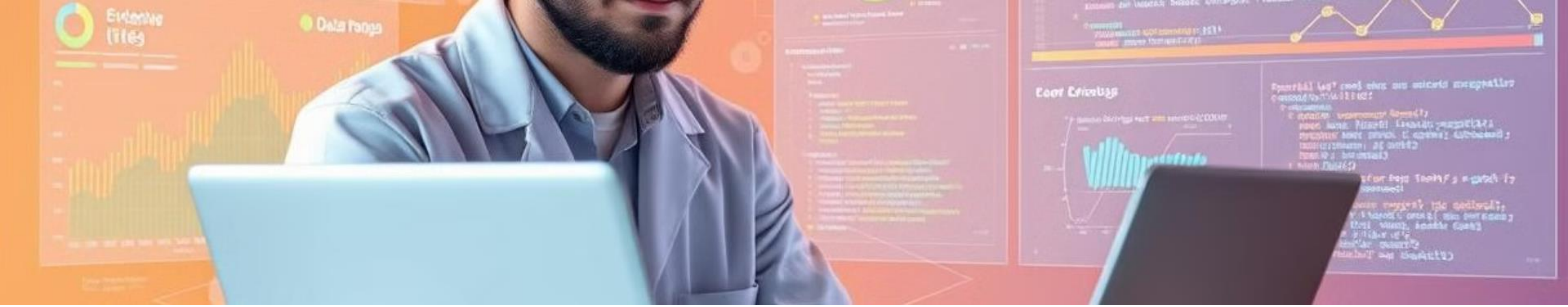
```
# Impute Categorical Columns with Mode (Dataset 3)

# Impute categorical columns with mode (Dataset 3)
dataset_3[categorical_cols_3] = SimpleImputer(strategy='most_frequent').fit_transform(dataset_3[categorical_cols_3])

# Display updated dataset 3
print("Updated Categorical Columns [Dataset 3]:\n")
dataset_3[categorical_cols_3].head()
```

```
Updated Categorical Columns [Dataset 3]:
```

	dteday
0	29-01-2011
1	29-01-2011
2	29-01-2011
3	29-01-2011
4	29-01-2011



Data Preprocessing for Dataset 3

Imputing Numerical Columns

Filled missing values in numerical columns with the mean.

Imputing Categorical Columns

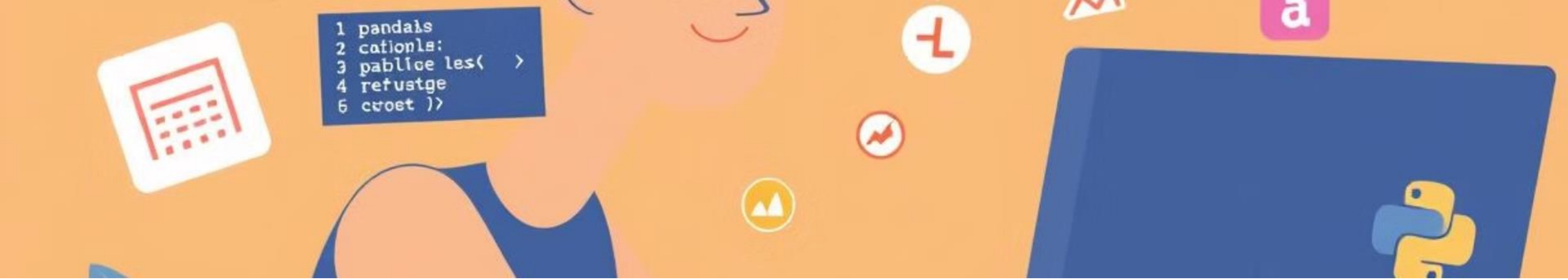
Filled missing values in categorical columns with the mode.

Central Tendency

Computed the mean, median, and mode for the numerical columns in Dataset 3.

Combining Cleaned Datasets

Dataset 1	Dataset 2	Dataset 3
Cleaned and preprocessed	Cleaned and preprocessed	Cleaned & preprocessed
Ready to be merged with Dataset 2	Merged Dataset 1 with Dataset 2	Merged All to make Final Dataset



```
pandals
cationls:
pablice les(   >
refustge
cwoet ])
```

Handling Missing Values and Outliers

01

Missing Value Imputation

Filled missing values in numerical columns with the mean and categorical columns with the mode.

02

Outlier Detection

Identified and handled outliers in the numerical columns using the interquartile range (IQR) method.

03

Skewness Analysis

Assessed the skewness of the numerical columns in the final dataset.

04

Correlation Visualization

Generated a correlation matrix to explore relationships between the variables.

I hope you found this presentation helpful and informative. Please don't hesitate to contact me with any questions or feedback you may have. I am more than eager to hear your thoughts and suggestions.

Summary

I have explored the key aspects of data preprocessing and exploratory analysis, covering steps like data loading, cleaning, merging, and handling missing values and outliers. This analysis provides a solid foundation for further data exploration and model building.

Future Enhancements

I am open to exploring additional enhancements in the future, including model selection and training. If you have any specific areas you'd like me to delve into or explore further, please let me know.

Q&A

I am available to answer any questions you may have. Feel free to reach out to me for any questions or clarifications.