

APPLIED DATA SCIENCE

ASSIGNMENT NO : 1

TITLE : PYTHON FOR DATA HANDLING

NAME: Sandip Dattatray Jadhav

ROLL NO : 82

CLASS : ECE

GITHUB LINK : https://github.com/sandipjadhav87/Applied_Data_Science/tree/main

DATASET LINK: <https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand>

CODE:

```
# =====  
# APPLIED DATA SCIENCE  
# Assignment 1 – Python for Data Handling  
# Dataset: Hotel Booking Demand (Kaggle)  
# =====  
  
# 1. Import Libraries  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# -----  
# 2. Load Dataset (500 Records Already Selected)  
# -----  
df = pd.read_csv("hotel_500_records.csv")  
  
print("Dataset Loaded Successfully")  
print("="*100)
```

3. Clean Column Names

```
# -----  
df.columns = df.columns.str.strip().str.lower()
```

```
print("Columns after cleaning:")  
print(df.columns.tolist())  
print("="*100)
```

4. Dataset Exploration

```
# -----  
print("First 5 Records:")  
print(df.head())
```

```
print("\nDataset Shape:", df.shape)
```

```
print("\nDataset Information:")  
df.info()  
print("="*100)
```

5. Check Missing Values and Zero Values

```
# -----  
print("Missing Values:")  
print(df.isnull().sum())
```

```
numerical_cols = df.select_dtypes(include=np.number).columns
```

```
print("\nZero Values in Numerical Columns:")  
print((df[numerical_cols] == 0).sum())  
print("="*100)
```

6. Remove Duplicate Records

```
print("Duplicate Records:", df.duplicated().sum())  
df.drop_duplicates(inplace=True)  
print("Duplicates Removed")  
print("="*100)
```

7. Handle Missing Values

```
df[numerical_cols] = df[numerical_cols].fillna(df[numerical_cols].mean())
```

```
cat_cols = df.select_dtypes(include=['object']).columns
```

```
for col in cat_cols:
```

```
    df[col] = df[col].fillna(df[col].mode()[0])
```

```
print("Missing Values Handled")
```

```
print("="*100)
```

8. Feature Engineering – Total Guests

```
df['total_guests'] = df['adults'] + df['children']
```

```
print("Total Guests Column Created")
```

```
print("="*100)
```

9. Statistical Measures

```
print("ADR Statistics")
```

```
print("Mean:", df['adr'].mean())
```

```
print("Median:", df['adr'].median())
```

```
print("Mode:", df['adr'].mode()[0])
```

```
print("Skewness:", df['adr'].skew())
```

```
print("="*100)
```

10. Basic Visualization

```
plt.figure(figsize=(6,4))
```

```
sns.histplot(df['adr'], kde=True)
```

```
plt.title("Distribution of ADR")
```

```
plt.show()
```

```

plt.figure(figsize=(6,4))

sns.countplot(x='hotel', data=df)

plt.title("Hotel Type Distribution")

plt.show()

print("Preprocessing Completed Successfully")

```

OUTPUT:

```

PS C:\Users\Dell\Desktop\ADS> & "C:/Program Files/Python311/python.exe" c:/Users/Dell/Desktop/ADS/data_handling.py
Dataset Shape: (500, 10)
   hotel  is_canceled  lead_time  ...  agent  company  adr
0  Resort Hotel      0      342  ...   NaN     NaN    0.0
1  Resort Hotel      0      737  ...   NaN     NaN    0.0
2  Resort Hotel      0       7  ...   NaN     NaN   75.0
3  Resort Hotel      0      13  ...  304.0     NaN   75.0
4  Resort Hotel      0      14  ...  240.0     NaN   98.0

[5 rows x 10 columns]
PS C:\Users\Dell\Desktop\ADS> & "C:/Program Files/Python311/python.exe" c:/Users/Dell/Desktop/ADS/data_handling.py
Dataset Loaded Successfully
=====
Columns after cleaning:
['hotel', 'is_canceled', 'lead_time', 'arrival_date_month', 'adults', 'children', 'country', 'agent', 'company', 'adr']

```

First 5 Records:

| | hotel | is_canceled | lead_time | ... | agent | company | adr |
|---|--------------|-------------|-----------|-----|-------|---------|------|
| 0 | Resort Hotel | 0 | 342 | ... | NaN | NaN | 0.0 |
| 1 | Resort Hotel | 0 | 737 | ... | NaN | NaN | 0.0 |
| 2 | Resort Hotel | 0 | 7 | ... | NaN | NaN | 75.0 |
| 3 | Resort Hotel | 0 | 13 | ... | 304.0 | NaN | 75.0 |
| 4 | Resort Hotel | 0 | 14 | ... | 240.0 | NaN | 98.0 |

[5 rows x 10 columns]

Dataset Shape: (500, 10)

Dataset Information:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   hotel                  500 non-null   object
1   is_canceled            500 non-null   int64
2   lead_time              500 non-null   int64
3   arrival_date_month     500 non-null   object
4   adults                 500 non-null   int64
5   children               500 non-null   float64
6   country                499 non-null   object
7   agent                  455 non-null   float64
8   company                7 non-null     float64
9   adr                   500 non-null   float64
dtypes: float64(4), int64(3), object(3)
memory usage: 39.2+ KB

```

```
Missing Values:
```

```
hotel          0
is_canceled    0
lead_time      0
arrival_date_month 0
adults         0
children       0
country        1
agent          45
company        493
adr            0
dtype: int64
```

```
Zero Values in Numerical Columns:
```

```
is_canceled    398
lead_time      23
adults         0
children       445
agent          0
company        0
Duplicates Removed
```

```
=====
Missing Values Handled
```

```
Duplicates Removed
```

```
=====
Missing Values Handled
```

```
=====
Total Guests Column Created
```

```
=====
ADR Statistics
```

```
Mean: 115.17229437229437
```

```
Median: 112.0
```

```
Mode: 123.0
```

```
Skewness: -0.05463829989468755
=====
```

INTERPRETATION:

Data Cleaning

- Column names converted into lowercase format.
 - Duplicate booking records removed.
 - Missing values replaced using mean (numerical) and mode (categorical).

Data Exploration

- Dataset contains hotel booking details like hotel type, cancellation status, lead time, number of guests and ADR.
 - Dataset shape confirms 500 rows and selected columns.
 - Data types verified using `df.info()`.

Zero Value Analysis

- Checked zero values in numerical columns to detect unrealistic entries.

Feature Engineering

- Created new column `total_guests` from adults and children.
 - Helps analyze booking size.

Statistical Analysis

- Calculated mean, median, mode and skewness of ADR.
 - Helps understand price distribution pattern.