

```
---- 1. IMPORTS ----

import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from IPython.display import display

pd.set_option('display.max_columns', 200)

sns.set(style="whitegrid")

# ---- 2. LOAD DATA ----

# CHANGE THIS TO YOUR UPLOADED FILE

# UPLOAD 'python first.csv' TO COLAB SESSION AND THEN UPDATE THE PATH
# BELOW

FILE = ("C:\\\\Users\\\\akash\\\\Downloads\\\\python first.csv")      # <-- REPLACE WITH
# THE CORRECT PATH TO YOUR UPLOADED FILE

# df = pd.read_excel(FILE)      # <-- Use this if Excel
df = pd.read_csv(FILE)

print("Dataset Loaded Successfully!")

print("Shape:", df.shape)

df.head()

# ---- 3. BASIC DATA UNDERSTANDING ----

print("---- INFO ----")

df.info()

print("\n---- DESCRIBE NUMERIC ----")

display(df.describe().T)
```

```

print("\n---- DESCRIBE CATEGORICAL ----")
display(df.describe(include=['object']).T)

# ---- 4. MISSING VALUES ----

missing = df.isnull().sum().sort_values(ascending=False)
missing_percent = (missing / len(df) * 100).round(2)
missing_table = pd.concat([missing, missing_percent], axis=1)
missing_table.columns = ['missing_count', 'missing_percent']

print("\n---- MISSING VALUE SUMMARY ----")
display(missing_table)

print("\nDuplicate rows:", df.duplicated().sum())

# ---- 5. VALUE COUNTS FOR CATEGORICAL COLUMNS ----

cat_cols = df.select_dtypes(include=['object', 'category']).columns.tolist()

for c in cat_cols:
    print(f"\nTop categories in: {c}")
    print(df[c].value_counts(dropna=False).head(10))

# ---- 6. CORRELATION HEATMAP ----

num_cols = df.select_dtypes(include=[np.number]).columns.tolist()

plt.figure(figsize=(12,10))
sns.heatmap(df[num_cols].corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()

# ---- 7. PAIRPLOT (Sampled to avoid overload) ----

if len(df) > 500:
    df_sample = df.sample(500, random_state=1)

```

```

else:
    df_sample = df

sns.pairplot(df_sample[num_cols], diag_kind="kde")
plt.show()

# ---- 8. DISTRIBUTIONS – HISTOGRAM + BOXPLOT ----

for c in num_cols:
    fig, axes = plt.subplots(1, 2, figsize=(12,4))

    # Histogram
    axes[0].hist(df[c].dropna(), bins=30)
    axes[0].set_title(f"Histogram: {c}")

    # Boxplot
    sns.boxplot(x=df[c], ax=axes[1])
    axes[1].set_title(f"Boxplot: {c}")

    plt.tight_layout()
    plt.show()

# ---- 9. CATEGORICAL vs NUMERIC (Boxplots) ----

for cat in cat_cols:
    for num in num_cols:
        plt.figure(figsize=(10,4))

        top_vals = df[cat].value_counts().nlargest(8).index
        sns.boxplot(x=df[cat].apply(lambda x: x if x in top_vals else "Other"),
                    y=df[num])

        plt.xticks(rotation=45)

```

```

plt.title(f"{num} by {cat}")
plt.tight_layout()
plt.show()

# ---- 10. GROUPED SUMMARY (First categorical column) ----

if cat_cols:

    grp = df.groupby(cat_cols[0])[num_cols].agg(['mean','median','count'])

    display(grp.head(20))

# ---- 11. OBSERVATIONS SECTION (WRITE YOUR NOTES HERE) ----

observations = """"

OBSERVATIONS / FINDINGS:

```

1. Data Types & Structure:

- The dataset contains numeric and categorical columns.
- No. of rows: {}
- No. of columns: {}.

2. Missing Values:

- Columns with highest missing values identified above.
- Suggest: impute numeric with median, categorical with mode.

3. Correlation Insights:

- Positive correlations:
 - * (mention highly correlated columns)
- Negative correlations:
 - * (mention inverse relationships)

4. Distributions:

- Skew detected in: (list numeric columns with long tails)
- Outliers detected in: (boxplot results)

5. Key Category Insights:

- Top categories for each categorical variable identified.
- Some categories dominate the dataset.

6. Overall Summary:

- Data is clean/moderate/messy.
- Next steps: Feature engineering / modeling / deeper analysis.

```
""".format(df.shape[0], df.shape[1])
```

```
print(observations)
```

```
# ---- 12. OPTIONAL: SAVE CLEAN SAMPLE & MISSING VALUE CSV ----
```

```
os.makedirs("eda_outputs", exist_ok=True)
```

```
df.sample(100).to_csv("eda_outputs/sample_100_rows.csv", index=False)
```

```
missing_table.to_csv("eda_outputs/missing_summary.csv")
```

```
print("Output files saved in eda_outputs/")
```