DATA ANALYSIS ON LARGE DATASET

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Settings
pd.set_option('display.max_columns', None)
sns.set(style="whitegrid")

# 1. CREATE DATASET IF NOT PRESENT
file_name = "dataset.csv"
if not os.path.exists(file_name):
    print("Dataset not found. Creating sample dataset...")

    np.random.seed(42)
    data = {
        "Age": np.random.randint(18, 60, 500),
        "Salary": np.random.randint(20000, 120000, 500),
        "Experience": np.random.randint(0, 40, 500),
        "Score": np.random.normal(70, 10, 500),
        "Department": np.random.choice(["HR", "IT", "Sales", "Finance"], 500)
    }
    df = pd.DataFrame(data)
```

```
    # Introduce missing values
    df.loc[10:20, "Salary"] = np.nan
    df.loc[30:35, "Department"] = np.nan
    df.to_csv(file_name, index=False)
    print("Sample dataset created as dataset.csv")
```

 # 2. LOAD DATASET

```
df = pd.read_csv(file_name)
print("\nDataset Loaded Successfully")
print("Shape:", df.shape)
print(df.head())
```

 # 3. DATA INFO

```
print("\nDataset Info:")
print(df.info())
```

 # 4. HANDLE MISSING VALUES

```
num_cols = df.select_dtypes(include=np.number).columns
cat_cols  = df.select_dtypes(include="object").columns
df[num_cols] = df[num_cols].fillna(df[num_cols].mean())
for col in cat_cols:
    df[col] = df[col].fillna(df[col].mode()[0])
print("\nMissing values handled.")
```

 # 5. REMOVE DUPLICATES

```
df.drop_duplicates(inplace=True)
print("Duplicates removed.")
```

 # 6. REMOVE OUTLIERS (IQR)

```
Q1 = df[num_cols].quantile(0.25)
Q3 = df[num_cols].quantile(0.75)
IQR = Q3 - Q1
df = df[~((df[num_cols] < (Q1 - 1.5 * IQR)) |
       (df[num_cols] > (Q3 + 1.5 * IQR))).any(axis=1)]
print("Outliers removed.")
print("Final shape:", df.shape)
```

 # 7. STATISTICAL SUMMARY

```
print("\nStatistical Summary:")
print(df.describe())
```

 # 8. VISUALIZATIONS

```
# Histogram
df[num_cols].hist(figsize=(10, 7), bins=20)
plt.suptitle("Feature Distributions")
plt.show()
```

```
# Boxplot
plt.figure(figsize=(10, 6))
sns.boxplot(data=df[num_cols])
plt.title("Boxplot of Numerical Features")
plt.xticks(rotation=45)
plt.show()

# Heatmap
corr_matrix = df[num_cols].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```

 # 9. INSIGHTS

```
high_corr = corr_matrix.abs().unstack().sort_values(ascending=False)
high_corr = high_corr[high_corr < 1]
print("\nTop Correlated Feature Pairs:")
print(high_corr.head(10))
```

 # 10. SAVE OUTPUTS

```
df.to_csv("cleaned_dataset.csv", index=False)
with open("DA_Report.txt", "w") as file:
    file.write("DATA ANALYSIS REPORT\n")
    file.write("====================\n\n")
```

```
    file.write(f"Final Dataset Shape: {df.shape}\n\n")
    file.write("Statistical Summary:\n")
    file.write(str(df.describe()))
    file.write("\n\nCorrelation Matrix:\n")
    file.write(str(corr_matrix))

print("\nCleaned dataset and report generated successfully!")
```

OUTPUT

```
Dataset Loaded Successfully
Shape: (500, 5)
   Age  Salary  Experience      Score Department
0   56 81476.0          13 67.645433         HR
1   46 64811.0          29 71.021289         HR
2   32 56208.0          34 75.510677    Finance
3   25 40150.0          20 70.388776    Finance
4   38 91180.0          36 70.432717         HR
```

Dataset Info:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 500 entries, 0 to 499

Data columns (total 5 columns):

#	Column	Non-Null Count	Dtype
0	Age	500 non-null	int64
1	Salary	489 non-null	float64
2	Experience	500 non-null	int64
3	Score	500 non-null	float64
4	Department	494 non-null	object

dtypes: float64(2), int64(2), object(1)

memory usage: 19.7+ KB

None

Missing values handled.

Duplicates removed.

Outliers removed.

Final shape: (496, 5)

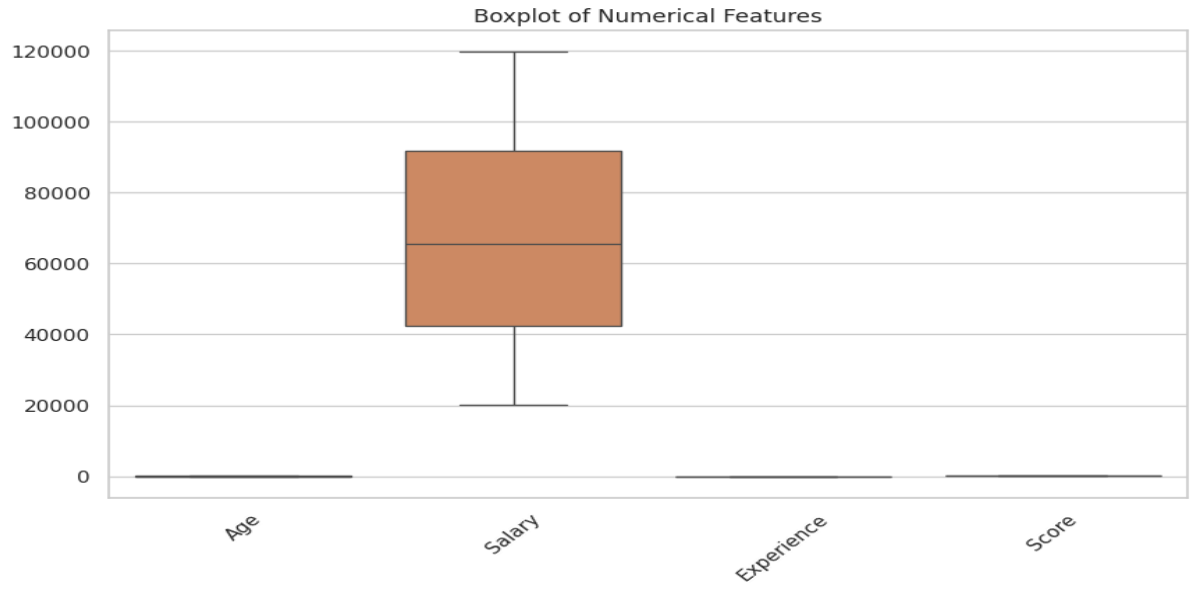
Statistical Summary:

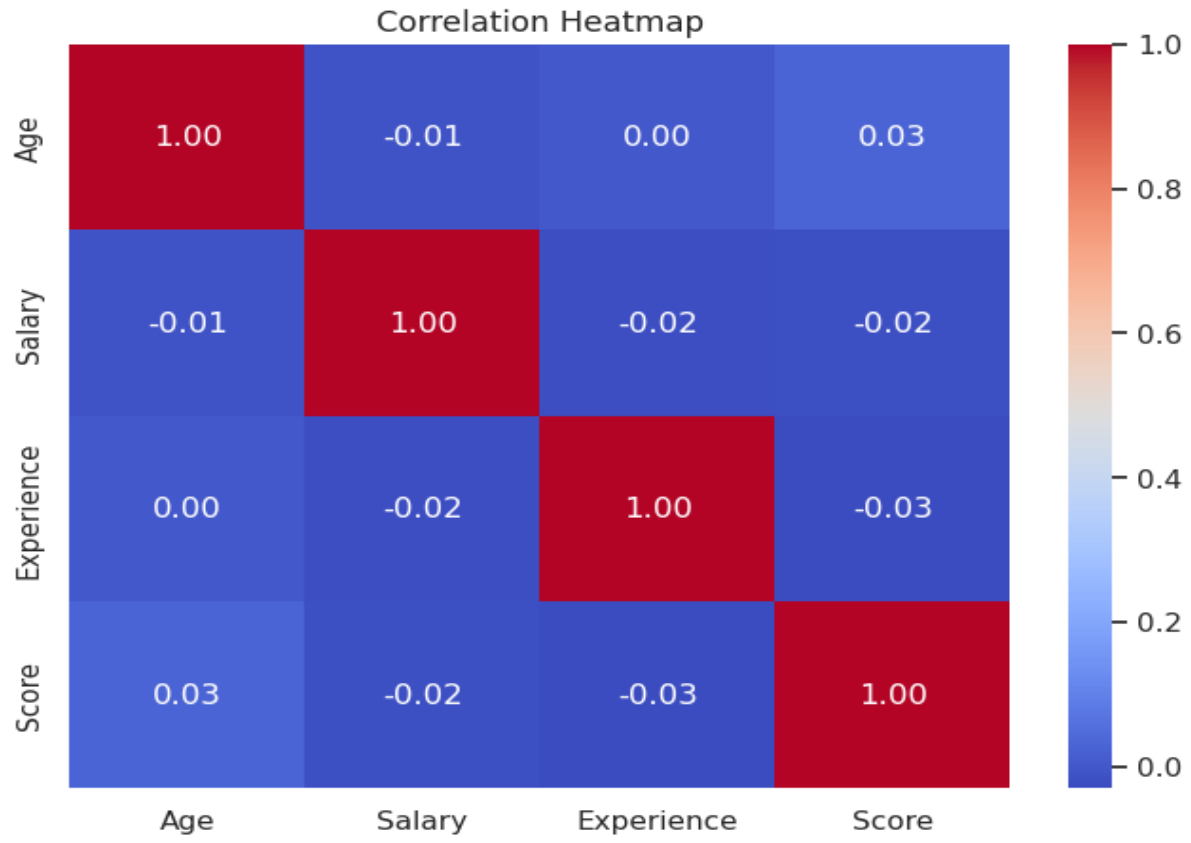
	Age	Salary	Experience	Score
count	496.000000	496.000000	496.000000	496.000000
mean	39.272177	66669.340363	20.481855	70.032838
std	12.182970	28531.744442	11.681102	10.241025
min	18.000000	20055.000000	0.000000	45.314571
25%	29.000000	42534.250000	10.000000	63.414936

50% 41.000000 65565.000000 22.000000 70.454376

75% 50.000000 91861.500000 31.000000 77.403336

max 59.000000 119835.000000 39.000000 96.292126





Top Correlated Feature Pairs:

Age Score 0.030835

Score Age 0.030835

Experience Score 0.029078

Score Experience 0.029078

Salary Experience 0.022714

Experience Salary 0.022714

Score Salary 0.020118

Salary Score 0.020118

Age Salary 0.011853

Salary Age 0.011853

dtype: float64

Cleaned dataset and report generated successfully!

Conclusion

This project successfully demonstrated data analysis on a large dataset using Python libraries such as Pandas, NumPy, Matplotlib, and Seaborn. The dataset was cleaned by handling missing values, removing duplicates, and eliminating outliers to improve data quality. Statistical summaries helped in understanding the central tendencies and variability of features. Visualizations like histograms, boxplots, and heatmaps revealed important patterns, distributions, and correlations. These insights support better understanding of relationships among variables. The cleaned dataset and generated report can be effectively used for decision-making or as a foundation for further machine learning model development.