In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
# Load the dataset
file_path = r'C:/Users/HP/Downloads/SampleSuperstore.csv'  # Replace with your a
data = pd.read_csv(file_path)
```

In [ ]:
```
data.head()
```

In [ ]:
```
# Check for missing values
missing_values = data.isnull().sum()
print("Missing Values:\n", missing_values)

# Fill missing values with mean, median, or placeholders
for column in data.columns:
    if data[column].isnull().any():
        if data[column].dtype in ['int64', 'float64']:
            data[column].fillna(data[column].mean(), inplace=True)  # or use mea
        else:
            data[column].fillna('Unknown', inplace=True)  # Placeholder for cate
```

In [ ]:
```
# Remove duplicates
data.drop_duplicates(inplace=True)
```

In [ ]:
```
# Using IQR to detect outliers for numerical columns
numerical_cols = data.select_dtypes(include=['float64', 'int64']).columns

for column in numerical_cols:
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    # Define outlier condition
    outlier_condition = (data[column] < (Q1 - 1.5 * IQR)) | (data[column] > (Q3
    # Remove outliers
    data = data[~outlier_condition]
```

In [ ]:
```
# Check the data types of the columns
print(data.dtypes)

# Select only numeric columns for correlation
numeric_data = data.select_dtypes(include=[np.number])

# Calculate the correlation matrix
correlation_matrix = numeric_data.corr()
print("Correlation Matrix:\n", correlation_matrix)
```

In [ ]:
```
# One-hot encode categorical columns
data_encoded = pd.get_dummies(data, drop_first=True)

# Now calculate the correlation matrix on the encoded data
correlation_matrix_encoded = data_encoded.corr()
print("Correlation Matrix with Encoded Data:\n", correlation_matrix_encoded)
```

In [ ]:
```
# Summary statistics
summary_stats = data.describe()
print("Summary Statistics:\n", summary_stats)

# Correlation matrix
correlation_matrix = data.corr()
print("Correlation Matrix:\n", correlation_matrix)
```

Plot histograms for numerical features

data.hist(bins=30, figsize=(15, 10)) plt.tight_layout() plt.show()

In [ ]:
```
# Boxplots for continuous variables
plt.figure(figsize=(15, 10))
for i, column in enumerate(numerical_cols):
    plt.subplot(3, 3, i + 1)
    sns.boxplot(y=data[column])
    plt.title(column)
plt.tight_layout()
plt.show()
```

In [ ]:
```
# Heatmap for correlation matrix
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', square=T
plt.title('Correlation Heatmap')
plt.show()
```

In [ ]:
```
# Save the cleaned dataset
data.to_csv('cleaned_sample_superstore.csv', index=True)
```

In [ ]:
```
# Save summary statistics to a CSV file
summary_stats.to_csv('summary_statistics.csv')
```

In [ ]:
```

```

In [ ]:
```

```