

Comprehensive Sales Data Analysis Report

Step 1: Data Loading and Exploration

```
```python
```

```
import pandas as pd
```

```
Load the dataset
```

```
data = pd.read_csv('dataset.csv')
```

```
Display the first few rows of the dataset
```

```
print(data.head())
```

```
Display basic information about the dataset to understand the data types and any missing values
```

```
print(data.info())
```

```
Display summary statistics for numerical columns
```

```
print(data.describe())
```

```
```
```

****Explanation:****

- Load the dataset using pandas and inspect the first few rows using `data.head()`.
- Use `data.info()` to get an overview of the data types and missing values.
- `data.describe()` provides summary statistics for numerical columns to understand data distribution.

Step 2: Data Cleaning

```
```python
```

```
import numpy as np
```

```
Clean the 'Price' column by handling non-standard formats
```

```
def extract_price(price):
```

```
 if isinstance(price, str): # Ensure the value is a string
```

```
 # If the price contains a range, take the average of the two prices
```

```
 if 'through' in price:
```

```
 try:
```

```
 # Extract both prices, convert to float, and take the average
```

```
 prices = [float(p) for p in price.replace('through', '-').split('-')]
```

```
 return np.mean(prices)
```

```
 except:
```

```
 return np.nan
```

```
 # Otherwise, remove any non-numeric characters and convert to float
```

```
 else:
```

```
 try:
```

```
 return float(price.replace('$', '').strip())
```

```
 except:
```

```
 return np.nan
```

```
 return np.nan # Return NaN if the value is not a string
```

```
Apply the function to clean the 'Price' column
```

```
data['Price'] = data['Price'].apply(extract_price)
```

```
Extract the rating value and convert it to numeric
```

```
data['Rating Value'] = data['Rating'].str.extract(r'Rated ([\d.]+) out of 5').astype(float)
```

```
Extract the number of reviews and convert it to numeric
```

```
data['Review Count'] = data['Rating'].str.extract(r'based on (\d+) reviews').astype(float)
```

```
Check for missing values and data types after conversion
```

```
print(data.info())
```

```
...
```

```
Explanation:
```

- Clean the `Price` column by handling non-standard formats and converting it to numeric.
- Extract the numeric rating value from the `Rating` column.
- Extract the number of reviews from the `Rating` column.
- Ensure all necessary columns are in the correct format for analysis.

```
Step 3: Descriptive Statistics and Insights
```

```
```python
```

```
# Descriptive statistics for numeric columns (Price, Rating Value, Review Count)
```

```
numeric_summary = data[['Price', 'Rating Value', 'Review Count']].describe()
```

```
print(numeric_summary)
```

```
# Identify top categories by total sales
```

```
top_categories = data.groupby('Sub Category')['Price'].sum().sort_values(ascending=False)
```

```
print(top_categories)
```

```
# Identify top-rated products
```

```
top_rated_products = data[data['Rating Value'] == 5][['Title', 'Price', 'Review
```

```
Count']].sort_values(by='Review Count', ascending=False)
```

```
print(top_rated_products)
```

```
# Identify most reviewed products
```

```
most_reviewed_products = data.sort_values(by='Review Count', ascending=False)[['Title', 'Rating  
Value', 'Review Count', 'Price']]
```

```
print(most_reviewed_products.head(10))
```

```
...
```

```
**Explanation:**
```

- Calculate summary statistics to understand the distribution of prices, ratings, and review counts.
- Identify the top categories by total sales.
- Find top-rated products with the most reviews to understand customer preferences.
- Identify the most reviewed products to highlight popular items.

```
### Step 4: Data Visualization
```

```
```python
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
Set the style for seaborn
```

```
sns.set(style="whitegrid")
```

```
Visualization 1: Price Distribution
```

```
plt.figure(figsize=(10, 6))
```

```
sns.histplot(data['Price'], bins=30, kde=True)
```

```
plt.title('Price Distribution of Products')
```

```
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()
```

# Visualization 2: Total Sales by Sub Category

```
plt.figure(figsize=(12, 6))
sales_by_category = data.groupby('Sub Category')['Price'].sum().reset_index()
sns.barplot(x='Price', y='Sub Category', data=sales_by_category, palette='viridis')
plt.title('Total Sales by Sub Category')
plt.xlabel('Total Sales (in $)')
plt.ylabel('Sub Category')
plt.show()
```

# Visualization 3: Ratings vs. Review Count

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Rating Value', y='Review Count', data=data, hue='Sub Category', palette='Set2')
plt.title('Ratings vs. Review Count')
plt.xlabel('Rating Value')
plt.ylabel('Review Count')
plt.legend(title='Sub Category', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
...
```

**\*\*Explanation:\*\***

- Visualize the distribution of prices with a histogram.
- Create a bar chart to visualize total sales by sub-category.
- Use a scatter plot to show the relationship between ratings and the number of reviews, with

product categories as color-coded points.

### ### Step 5: Report Summary

#### **\*\*Sales Data Analysis Report\*\***

##### 1. **\*\*Data Overview:\*\***

- The dataset contains various product categories with information on prices, ratings, discounts, and reviews.

##### 2. **\*\*Key Insights:\*\***

- **\*\*Price Distribution:\*\*** Most products are priced below \$100, with a significant concentration under \$50. A few high-priced items go up to \$999.99.
- **\*\*Top Categories by Sales:\*\*** Certain categories, like "Bakery & Desserts" and "Beverages," dominate sales, indicating popular product lines.
- **\*\*Top-Rated Products:\*\*** Products with a perfect 5-star rating are scattered across different categories, but only a few have a significant number of reviews, indicating niche popularity.
- **\*\*Review Analysis:\*\*** Products with high review counts also tend to have high ratings, suggesting customer satisfaction.

##### 3. **\*\*Recommendations:\*\***

- Focus marketing and inventory on high-selling and high-rated categories.
- Consider offering promotions or discounts for less popular categories to boost sales.
- Monitor and promote top-reviewed and top-rated products to leverage customer satisfaction.