Statistics Assignment 1

Prepared by Sohana Tasneem (Batch 2412)

Python Code File : Statistics A1.ipynb

Task 1: Grouped Frequency Table for "Weight in gms"

Code :

```
weight = df['Weight_in_gms']
n = len(weight)

# Calculate number of bins using Sturges' Rule
k = int(1 + 3.322 * np.log10(n))  # Number of classes
range_ = weight.max() - weight.min()
class_width = int(np.ceil(range_ / k))

# Define bins
bins = np.arange(weight.min(), weight.max() + class_width, class_width)

# Create grouped frequency distribution table
frequency_table = pd.cut(weight, bins=bins,
right=False).value_counts().sort_index()

# Display the table
grouped_freq_table = pd.DataFrame({
    'Weight Interval': frequency_table.index.astype(str),
    'Frequency': frequency_table.values
})
print(grouped_freq_table)
# Plot histogram
plt.figure(figsize=(10, 6))
plt.hist(df['Weight_in_gms'], bins=bins, edgecolor='black',
color='lightgreen')
plt.xlabel("Weight in gms")
plt.ylabel("Frequency")
plt.title("Histogram of Weight in gms")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Output :

	Weight Interval	Frequency
0	[1001, 1490)	1594
1	[1490, 1979)	1593
2	[1979, 2468)	482
3	[2468, 2957)	435
4	[2957, 3446)	411
5	[3446, 3935)	455
6	[3935, 4424)	1372
7	[4424, 4913)	1493
8	[4913, 5402)	1441
9	[5402, 5891)	1410
10	[5891, 6380)	307
11	[6380, 6869)	1
12	[6869, 7358)	0
13	[7358, 7847)	5

Explanation :

The frequency distribution shows that product weights tend to cluster in **two main ranges:** one between **1000g and 2000g,** and another between **4000g and 5900g.**

These peaks suggest that the company handles **both lighter and mid-heavy products** in **large volumes.**

Weights beyond 6000g are rare, indicating the presence of only a few very heavy items in the dataset.

Task 2:Construct a “Contingency Table” (Cross-tabulation) between two categorical columns, “Warehouse_block” and “Mode_of_Shipment”.

Code:

```
# Generate the contingency table
contingency_table = pd.crosstab(df['Warehouse_block'],
df['Mode_of_Shipment'])

# Convert to regular DataFrame for cleaner printing
contingency_table_df = contingency_table.reset_index()

# Print a clean, aligned table
print("Contingency Table: Warehouse_block vs Mode_of_Shipment")
print(contingency_table_df.to_string(index=False))
```

Output:

```
Contingency Table: Warehouse_block vs Mode_of_Shipment
Warehouse_block  Flight  Road  Ship
              A     297   294  1242
              B     296   294  1243
              C     295   294  1244
              D     297   292  1245
              F     592   586  2488
```

Explanation:

The contingency table reveals that **Warehouse F** handles the **highest number** of shipments across all modes — especially by **Ship**, with a significantly higher volume than other blocks.
Blocks A through D show very similar shipment distributions, with each using Ship the most, followed by Road and Flight.

This suggests a centralized or high-volume shipping strategy centered in Warehouse F, potentially due to location, capacity, or demand.

Task 3: Make graphical representation of the following columns:

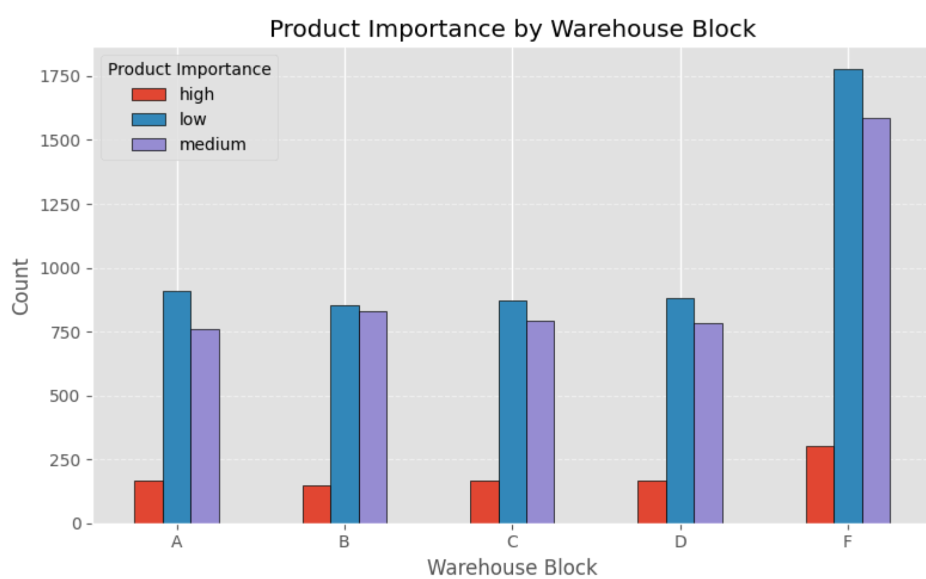
a. Bar chart of "Warehouse_block" and "Product_importance"

Code:

```
# Create a cross-tab of counts
group_data = pd.crosstab(df['Warehouse_block'], df['Product_importance'])

# Plot grouped bar chart
group_data.plot(kind='bar', figsize=(8, 5), edgecolor='black')
plt.title("Product Importance by Warehouse Block")
plt.xlabel("Warehouse Block")
plt.ylabel("Count")
plt.xticks(rotation=0)
plt.legend(title="Product Importance")
plt.grid(axis='y', linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```

Output:



Explanation:

The grouped bar chart shows that each warehouse block handles a fairly consistent distribution of product importance levels. **Medium importance products** are the **most common** across all blocks, followed by low and then high importance.

This indicates that **product importance is evenly distributed across warehouses,** suggesting a balanced operational strategy with no single warehouse specializing in high-priority products.

---

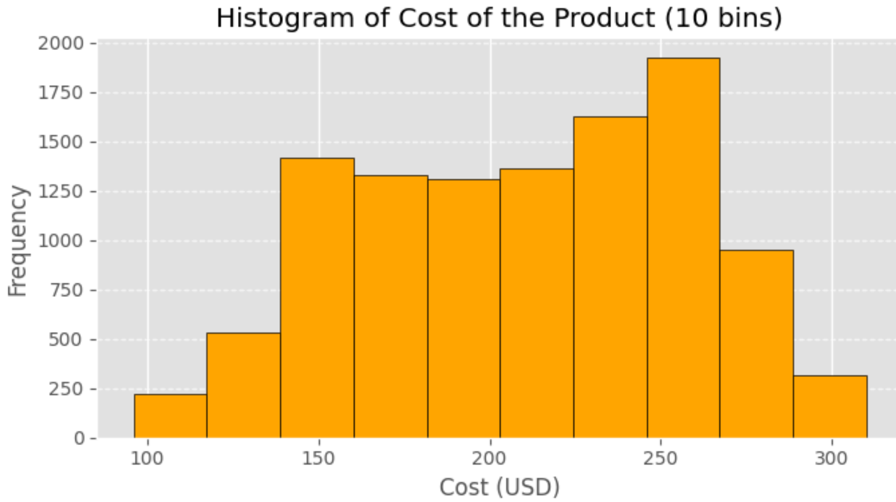
b. Histogram of "Cost_of_the_Product" and "Weight in gms" individually.

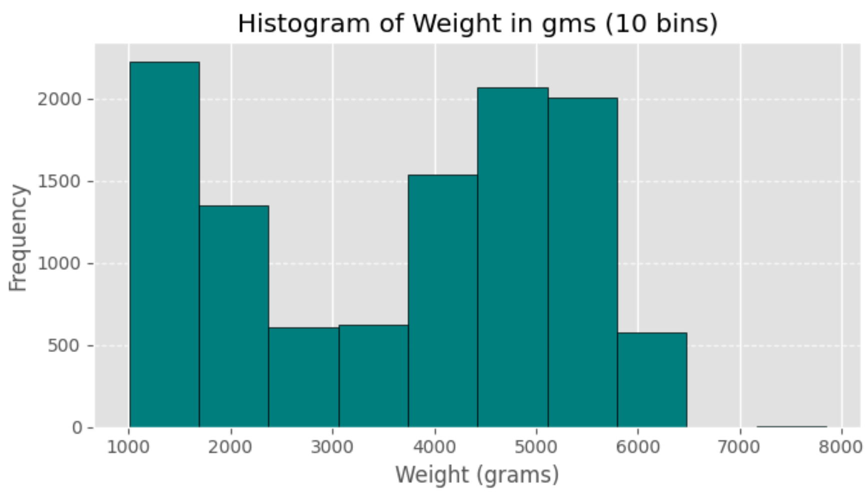
Code:

```
# Histogram for Cost_of_the_Product with 10 bins
plt.figure(figsize=(7, 4))
plt.hist(df['Cost_of_the_Product'], bins=10, color='orange',
edgecolor='black')
plt.title("Histogram of Cost of the Product (10 bins)")
plt.xlabel("Cost (USD)")
plt.ylabel("Frequency")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# Histogram for Weight_in_gms with 10 bins
plt.figure(figsize=(7, 4))
plt.hist(df['Weight_in_gms'], bins=10, color='teal', edgecolor='black')
plt.title("Histogram of Weight in gms (10 bins)")
plt.xlabel("Weight (grams)")
plt.ylabel("Frequency")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Output:





Explanation:

The histogram for **Cost of the Product** shows that most products fall within the **low to mid-cost** range, with a decreasing frequency as cost increases.
The distribution is **right-skewed**, suggesting a **few expensive products** but **many low-cost items.**

The weight histogram supports the earlier frequency table: it shows two major peaks — one around **1000–2000g** and another around **4000–6000g**, indicating a **bimodal distribution.**
Very heavy items are rare, with few values beyond 6000g.

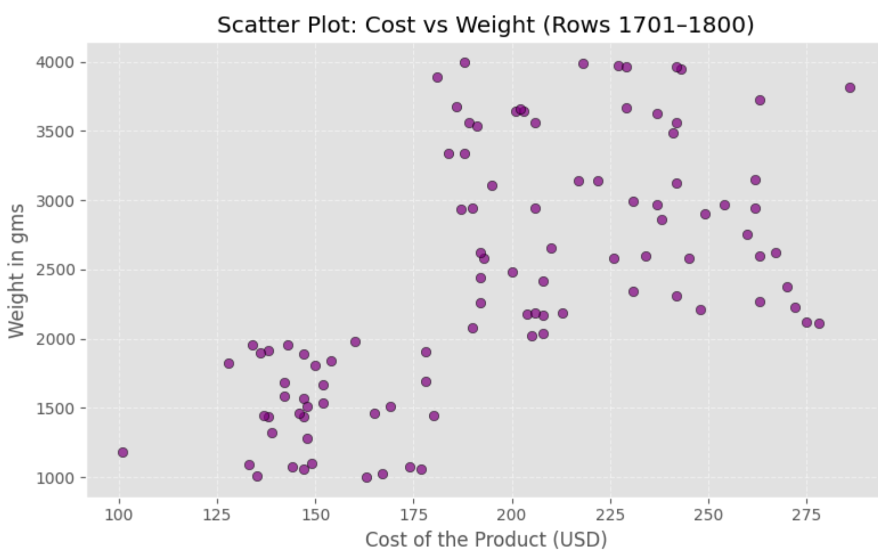
c. Scatter Plot of "Cost_of_the_Product" and "Weight in gms" individually. (take 200 data points)

Code:

```
#randomly taken 200 data points from 1700 to 1800
# Slice the dataset using the assigned range
subset = df.iloc[1700:1800]  # Note: iloc is 0-based, so 1700:1800 gets rows 1701 to 1800

# Scatter Plot
plt.figure(figsize=(8, 5))
plt.scatter(subset['Cost_of_the_Product'], subset['Weight_in_gms'], color='purple', edgecolors='black', alpha=0.7)
plt.title("Scatter Plot: Cost vs Weight (Rows 1701–1800)")
plt.xlabel("Cost of the Product (USD)")
plt.ylabel("Weight in gms")
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```

Output:



Explanation:

The scatter plot displays a **widely scattered** distribution, indicating **no strong linear correlation** between **product cost** and **weight.** Some heavier items are low-cost, and some lighter items are high-cost, suggesting that factors other than weight influence price.

The plot also reveals a few outliers — possibly special or high-end products that are priced differently than the rest.

Summary :

📊**Task 1 – Grouped Frequency Distribution of Weight in gms:**
The distribution of product weights is bimodal, with two noticeable peaks: one between **1000g and 2000g**, and another between **4000g and 6000g**. Very heavy products (above 6000g) are rare, indicating that most shipments are clustered in light to mid-heavy categories.

📊**Task 2 – Contingency Table (Warehouse_block vs Mode_of_Shipment):**
The contingency table shows that **Warehouse F** handles the most shipments across all modes, especially by **Ship**. Warehouses A to D have nearly identical mode usage patterns, with Ship being the most common.

📊**Task 3(a) – Grouped Bar Chart (Product Importance by Warehouse Block):**
All warehouse blocks primarily handle **medium-importance products**, followed by low and high importance. This suggests a balanced distribution strategy, without any warehouse specializing in high-priority shipments.

📊**Task 3(b) – Histograms (Cost and Weight):**
The **cost histogram** is **right-skewed**, indicating that most products are low-cost with a few expensive outliers. The **weight histogram** confirms the **bimodal distribution** seen earlier, with major clusters in both light and mid-heavy categories.

📊**Task 3(c) – Scatter Plot (Cost vs Weight)**:
The scatter plot reveals **no strong linear correlation** between cost and weight. Products vary widely in cost regardless of weight, suggesting that pricing is influenced by factors other than physical mass.
