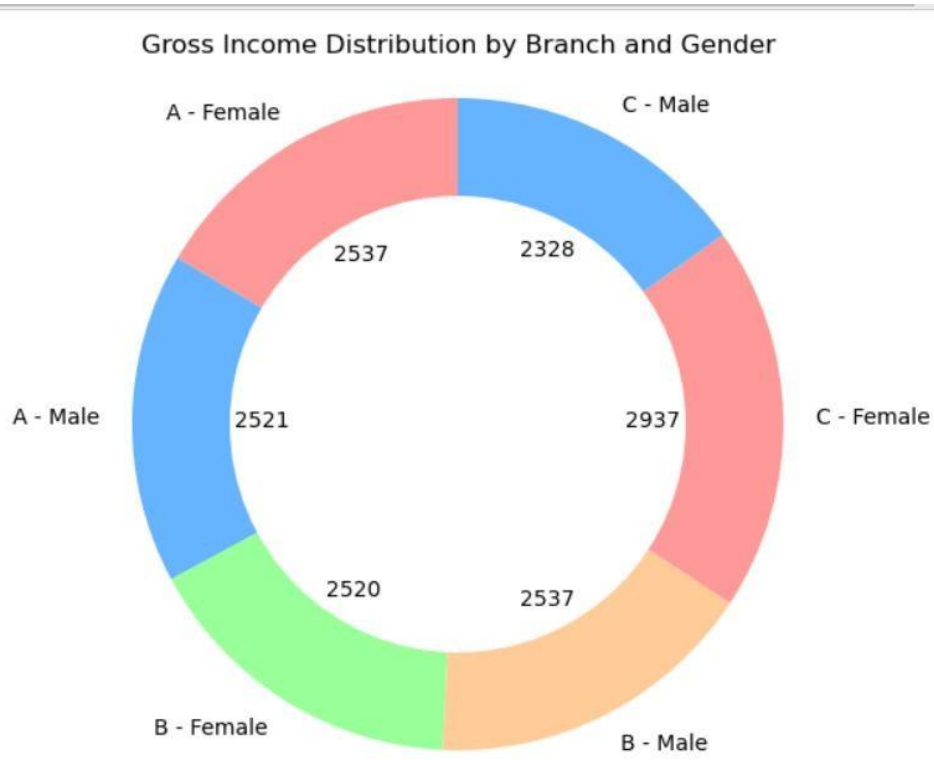
Assignment-2

Source Code:

```
#Donut Chart
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("C:\\Users\\yogap\\Downloads\\supermarket_sales - Sheet1.csv")
df = pd.DataFrame(data)
# Grouping data by branch and gender and summing up gross income
grouped_data = df.groupby(['Branch', 'Gender'])['gross income'].sum().reset_index()
# Plotting the donut chart
plt.figure(figsize=(8, 6))
colors = ['#ff9999','#66b3ff','#99ff99','#ffcc99']
total_income = grouped_data['gross income'].sum()
plt.pie(grouped_data['gross income'], labels=grouped_data.apply(lambda x: f"{x['Branch']} - {x['Gender']}", axis=1), autopct=lambda p: '{:.0f}'.format(p * total_income / 100), startangle=90, colors=colors)
plt.title('Gross Income Distribution by Branch and Gender')
plt.gca().add_artist(plt.Circle((0,0),0.70,fc='white'))
plt.axis('equal')
plt.show()
```

OUTPUT:



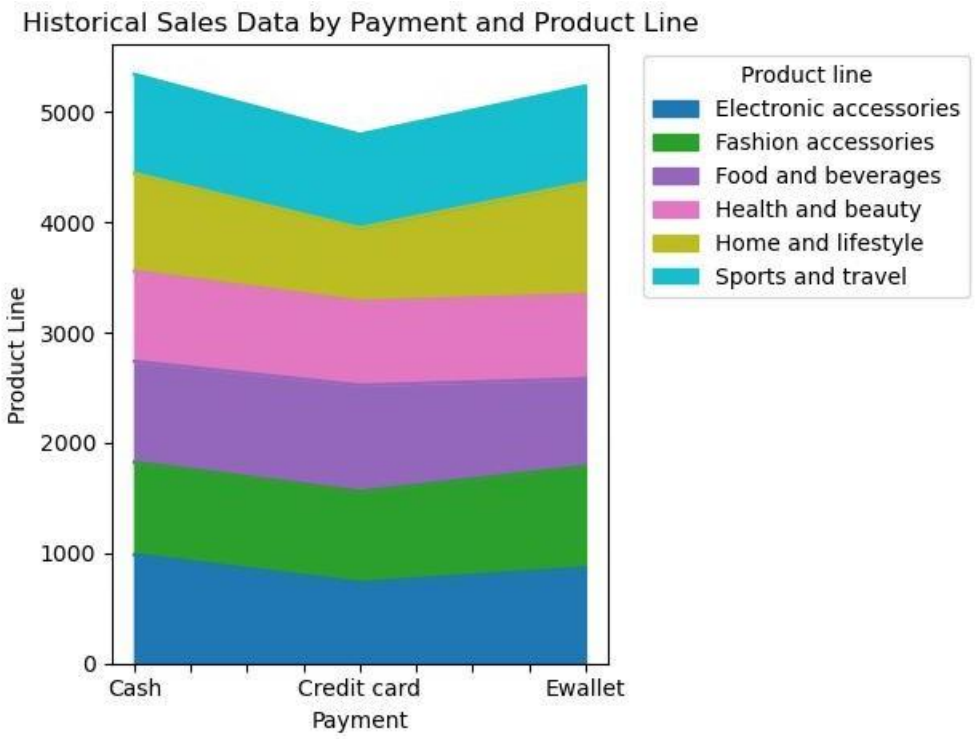
Source Code:

#Area Chart

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("C:\\Users\\yogap\\Downloads\\supermarket_sales - Sheet1.csv")
df = pd.DataFrame(data)

# Grouping data by branch and product line and summing up gross income
grouped_data = df.groupby(['Payment', 'Product line'])['gross income'].sum().reset_index()

# Pivot the data for visualization
pivot_data = grouped_data.pivot(index='Payment', columns='Product line', values='gross income').fillna(0)

# Plotting the area chart
plt.figure(figsize=(10, 6))
```

```
pivot_data.plot.area(stacked=True, cmap='tab10')
plt.title('Historical Sales Data by Payment and Product Line')
plt.xlabel('Payment')
plt.ylabel('Product Line')
plt.xticks(rotation=0)
plt.legend(title='Product line', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

OUTPUT:



Source Code:

```
#Text Table
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("C:\\Users\\yogap\\Downloads\\supermarket_sales - Sheet2.csv")
```

```
df = pd.DataFrame(data)

# Convert DataFrame to tabular format with pipe format
table = tabulate(df, headers='keys', tablefmt='plain', showindex=False)

# Print the table
print(table)
```

OUTPUT:

Branch	City	Customer type	Gender	Product line	Date	Total	Payment	Rating	Quanti ty
A	Yangon	Member	Female	Health and beauty	1/5/2019	548.971	Ewallet	9.1	7
C	Naypyitaw	Normal	Female	Electronic accessories	3/8/2019	80.22	Cash	9.6	5
A	Yangon	Normal	Male	Home and lifestyle	3/3/2019	340.526	Credit card	7.4	7
A	Yangon	Member	Male	Health and beauty	1/27/2019	489.048	Ewallet	8.4	8
A	Yangon	Normal	Male	Sports and travel	2/8/2019	634.379	Ewallet	5.3	7
C	Naypyitaw	Normal	Male	Electronic accessories	3/25/2019	627.616	Ewallet	4.1	7
A	Yangon	Member	Female	Electronic accessories	2/25/2019	433.692	Ewallet	5.8	6
C	Naypyitaw	Normal	Female	Home and lifestyle	2/24/2019	772.38	Ewallet	8	10
A	Yangon	Member	Female	Health and beauty	1/10/2019	76.146	Credit card	7.2	

Source Code:

```
#Highlighted table
import pandas as pd
# Sample data
data = pd.read_csv("C:\\Users\\yogap\\Downloads\\supermarket_sales - Sheet2.csv")
# Convert data to DataFrame
df = pd.DataFrame(data)
# Define function to highlight maximum value in each column
def highlight_max(s):
    is_max = s == s.max()
    return ['background-color: yellow' if v else '' for v in is_max]
# Apply highlight function to the DataFrame
highlighted_df = df.style.apply(highlight_max)
```

Display the highlighted table

highlighted_df

OUTPUT:

Out[28]:

	Branch	City	Customer type	Gender	Product line	Date	Total	Payment	Rating	Quantity
0	A	Yangon	Member	Female	Health and beauty	1/5/2019	548.971500	Ewallet	9.100000	7
1	C	Naypyitaw	Normal	Female	Electronic accessories	3/8/2019	80.220000	Cash	9.600000	5
2	A	Yangon	Normal	Male	Home and lifestyle	3/3/2019	340.525500	Credit card	7.400000	7
3	A	Yangon	Member	Male	Health and beauty	1/27/2019	489.048000	Ewallet	8.400000	8
4	A	Yangon	Normal	Male	Sports and travel	2/8/2019	634.378500	Ewallet	5.300000	7
5	C	Naypyitaw	Normal	Male	Electronic accessories	3/25/2019	627.616500	Ewallet	4.100000	7
6	A	Yangon	Member	Female	Electronic accessories	2/25/2019	433.692000	Ewallet	5.800000	6
7	C	Naypyitaw	Normal	Female	Home and lifestyle	2/24/2019	772.380000	Ewallet	8.000000	10
8	A	Yangon	Member	Female	Health and beauty	1/10/2019	76.146000	Credit card	7.200000	2
9	B	Mandalay	Member	Female	Food and beverages	2/20/2019	172.746000	Credit card	5.900000	3
10	B	Mandalay	Member	Female	Fashion accessories	2/6/2019	60.816000	Ewallet	4.500000	4

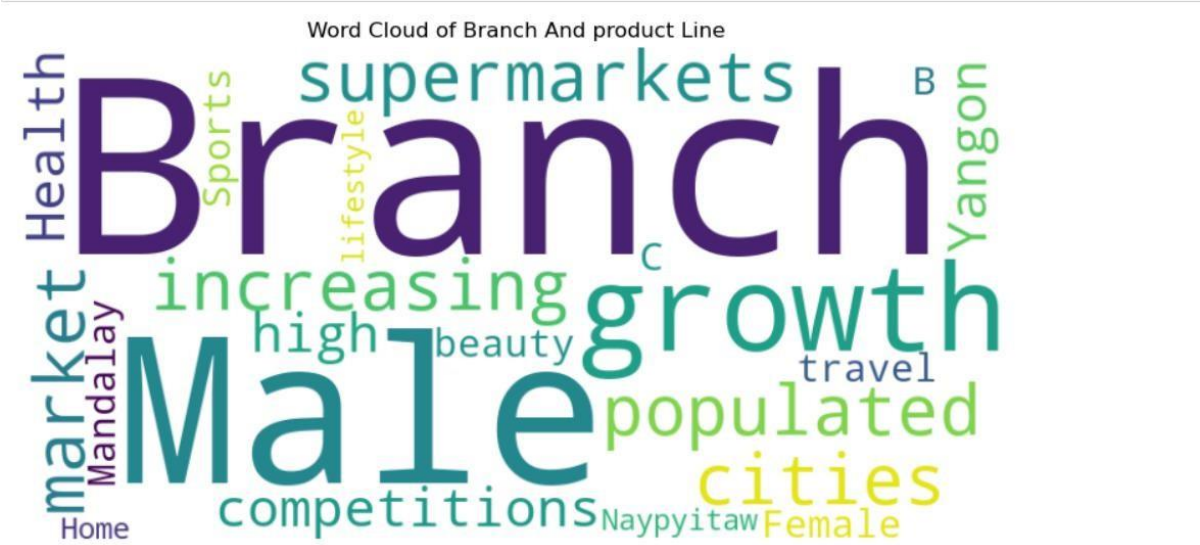
Source Code:

#WordCloud

```
from wordcloud import WordCloud

import matplotlib.pyplot as plt

# Description

description = "The growth of supermarkets in most populated cities is increasing and market competitions are also high."

# Sample data with updated 'City' data

data = {

    'Branch': ['Branch A', 'Branch B', 'Branch C'],

    'Gender': ['Male', 'Female', 'Male'],

    'City': ['Yangon', 'Mandalay', 'Naypyitaw'],  # Updated 'City' data

    'Product line': ['Health and beauty', 'Sports and travel', 'Home and lifestyle']

}

# Combine text from description and columns into a single string

text = description + ' ' + ' '.join(' '.join(str(value) for value in row) for row in zip(*data.values()))

# Generate word cloud

wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)
```

```
# Plot word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Branch And product Line')
plt.show()
```

OUTPUT:

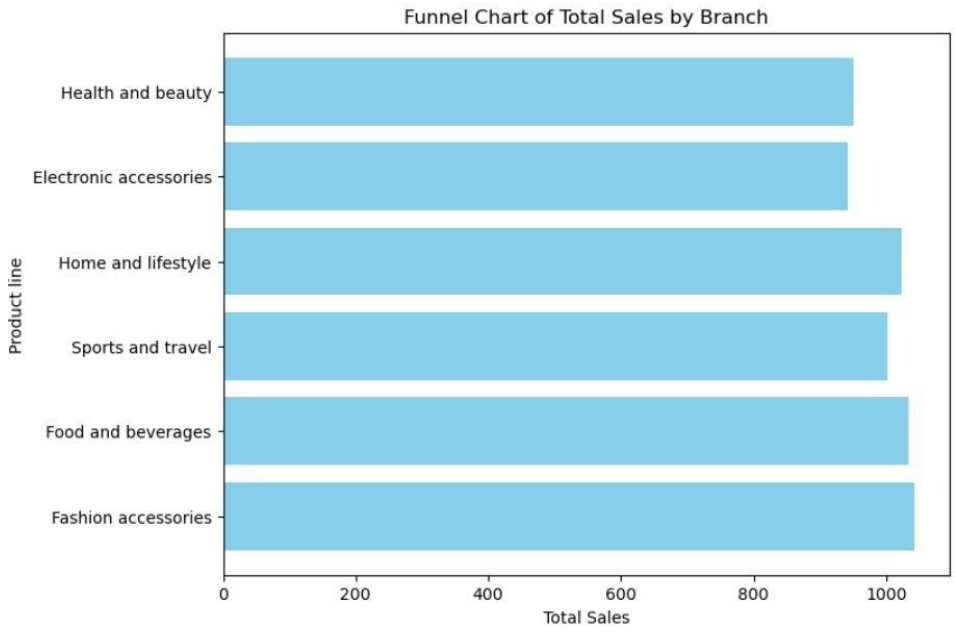


Source Code:

```
#Funnel Chart
import matplotlib.pyplot as plt
# Provided sample data
data = pd.read_csv("C:\\Users\\yogap\\Downloads\\supermarket_sales - Sheet1.csv")
df = pd.DataFrame(data)
# Plotting the funnel chart
plt.figure(figsize=(8, 6))
plt.barh(data['Product line'], data['Total'], color='skyblue')
plt.xlabel('Total Sales')
plt.ylabel('Product line')
```

```
plt.title('Funnel Chart of Total Sales by product Line')
plt.gca().invert_yaxis()  # Invert y-axis to show the top-down funnel shape
plt.show()
```

OUTPUT:



Source Code :

```
#Waterfall Chart
import matplotlib.pyplot as plt
# Sample data
data = {
   'Product line': ['Health and beauty', 'Sports and travel', 'Home and lifestyle', 'Fashion accessories', 'Electronic accessories', 'Total Costs', 'Net Profit'],
   'Amount': [100000, 60000, 30000, 90000, -20000, -50000, 40000]
}
# Calculate cumulative sum
data['Cumulative'] = [sum(data['Amount'][:i+1]) for i in range(len(data['Amount']))]
```

```
# Plot waterfall chart
plt.figure(figsize=(10, 6))
plt.bar(data['Product line'], data['Amount'], color='b', alpha=0.5, align='center')
plt.plot(data['Product line'], data['Cumulative'], color='orange', marker='o')
plt.title('Waterfall Chart of Product Line and Costs')
plt.xlabel('Product line')
plt.ylabel('Amount')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

OUTPUT: