Food Delivery cost Analysis

In [132…
```
import pandas as pd
df=pd.read_csv(r"C:\Users\PMLS\OneDrive\Desktop\MAKE IT\PYTHON FOR DATA ANALYSIS\1s
df.head()
```

Out[132…

	Order ID	Customer ID	Restaurant ID	Order Date and Time	Delivery Date and Time	Order Value	Delivery Fee	Payment Method	Discounts and Offers
0	1	C8270	R2924	2024-02-01 01:11:52	2024-02-01 02:39:52	1914	0	Credit Card	5% on App
1	2	C1860	R2054	2024-02-02 22:11:04	2024-02-02 22:46:04	986	40	Digital Wallet	10%
2	3	C6390	R2870	2024-01-31 05:54:35	2024-01-31 06:52:35	937	30	Cash on Delivery	15% New User
3	4	C6191	R2642	2024-01-16 22:52:49	2024-01-16 23:38:49	1463	50	Cash on Delivery	NaN
4	5	C6734	R2799	2024-01-29 01:19:30	2024-01-29 02:48:30	1992	30	Cash on Delivery	50 off Promo

In [133…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Order ID                1000 non-null   int64
 1   Customer ID             1000 non-null   object
 2   Restaurant ID           1000 non-null   object
 3   Order Date and Time     1000 non-null   object
 4   Delivery Date and Time  1000 non-null   object
 5   Order Value             1000 non-null   int64
 6   Delivery Fee            1000 non-null   int64
 7   Payment Method          1000 non-null   object
 8   Discounts and Offers    815 non-null    object
 9   Commission Fee          1000 non-null   int64
 10  Payment Processing Fee  1000 non-null   int64
 11  Refunds/Chargebacks     1000 non-null   int64
dtypes: int64(6), object(6)
memory usage: 93.9+ KB
```

Data Cleaning

In [134…

```
df["Order Date and Time"]=pd.to_datetime(df["Order Date and Time"])
df["Delivery Date and Time"]=pd.to_datetime(df["Delivery Date and Time"])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Order ID                1000 non-null   int64
 1   Customer ID             1000 non-null   object
 2   Restaurant ID           1000 non-null   object
 3   Order Date and Time     1000 non-null   datetime64[ns]
 4   Delivery Date and Time  1000 non-null   datetime64[ns]
 5   Order Value             1000 non-null   int64
 6   Delivery Fee            1000 non-null   int64
 7   Payment Method          1000 non-null   object
 8   Discounts and Offers    815 non-null    object
 9   Commission Fee          1000 non-null   int64
 10  Payment Processing Fee  1000 non-null   int64
 11  Refunds/Chargebacks     1000 non-null   int64
dtypes: datetime64[ns](2), int64(6), object(4)
memory usage: 93.9+ KB
```

In [135…

```
 #Extract the month from the 'Order Date and Time' column
df['Order Month'] = df['Order Date and Time'].dt.strftime('%B')
```

In [136…

```
# Extract the day name from the 'Order Date and Time' column
df['Order Day'] = df['Order Date and Time'].dt.strftime('%A')
```

In [137…

```
# Extract the time from the 'Order Date and Time' column
df['Order Time'] = df['Order Date and Time'].dt.strftime('%I:%M %p')
```

```
# Function to categorize time of day
def categorize_time(hour):
    if 5 <= hour < 12:
        return 'Morning'
    elif 12 <= hour < 17:
        return 'Afternoon'
    elif 17 <= hour < 21:
        return 'Evening'
    else:
        return 'Night'

# Apply function to categorize based on time
df['Order Period'] = df['Order Date and Time'].dt.hour.apply(categorize_time)
```

In [138…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype         
---  ------                  --------------  -----         
 0   Order ID                1000 non-null   int64         
 1   Customer ID             1000 non-null   object        
 2   Restaurant ID           1000 non-null   object        
 3   Order Date and Time     1000 non-null   datetime64[ns]
 4   Delivery Date and Time  1000 non-null   datetime64[ns]
 5   Order Value             1000 non-null   int64         
 6   Delivery Fee            1000 non-null   int64         
 7   Payment Method          1000 non-null   object        
 8   Discounts and Offers    815 non-null    object        
 9   Commission Fee          1000 non-null   int64         
 10  Payment Processing Fee  1000 non-null   int64         
 11  Refunds/Chargebacks     1000 non-null   int64         
 12  Order Month             1000 non-null   object        
 13  Order Day               1000 non-null   object        
 14  Order Time              1000 non-null   object        
 15  Order Period            1000 non-null   object        
dtypes: datetime64[ns](2), int64(6), object(8)
memory usage: 125.1+ KB
```

In [140…

```
def extract(value):
    a=str(value).split(" ")
    return a[0]

df["Discounts and Offers"]=df["Discounts and Offers"].apply(extract)
df.head()
```

Out[140…

	Order ID	Customer ID	Restaurant ID	Order Date and Time	Delivery Date and Time	Order Value	Delivery Fee	Payment Method	Discounts and Offers
0	1	C8270	R2924	2024-02-01 01:11:52	2024-02-01 02:39:52	1914	0	Credit Card	5%
1	2	C1860	R2054	2024-02-02 22:11:04	2024-02-02 22:46:04	986	40	Digital Wallet	10%
2	3	C6390	R2870	2024-01-31 05:54:35	2024-01-31 06:52:35	937	30	Cash on Delivery	15%
3	4	C6191	R2642	2024-01-16 22:52:49	2024-01-16 23:38:49	1463	50	Cash on Delivery	nan
4	5	C6734	R2799	2024-01-29 01:19:30	2024-01-29 02:48:30	1992	30	Cash on Delivery	50

In [141…

```
def removep(value):
    if"%" in value:
        a=value.replace("%","")
        return float(a)
    else:
        return float(value)

df["Discounts and Offers"]=df["Discounts and Offers"].apply(removep)
df.head()
```

Out[141…

	Order ID	Customer ID	Restaurant ID	Order Date and Time	Delivery Date and Time	Order Value	Delivery Fee	Payment Method	Discounts and Offers
0	1	C8270	R2924	2024-02-01 01:11:52	2024-02-01 02:39:52	1914	0	Credit Card	5.0
1	2	C1860	R2054	2024-02-02 22:11:04	2024-02-02 22:46:04	986	40	Digital Wallet	10.0
2	3	C6390	R2870	2024-01-31 05:54:35	2024-01-31 06:52:35	937	30	Cash on Delivery	15.0
3	4	C6191	R2642	2024-01-16 22:52:49	2024-01-16 23:38:49	1463	50	Cash on Delivery	NaN
4	5	C6734	R2799	2024-01-29 01:19:30	2024-01-29 02:48:30	1992	30	Cash on Delivery	50.0

In [142…

```
df.loc[(df["Discounts and Offers"] <= 15),"Discounts and Offers" ] = (df["Discounts
```

In [164…

```
df["Dicounts and Offers"]= df["Discounts and Offers"].fillna(0)
df.head()
```

Out[164...

	Order ID	Customer ID	Restaurant ID	Order Date and Time	Delivery Date and Time	Order Value	Delivery Fee	Payment Method	Discounts and Offers
0	1	C8270	R2924	2024-02-01 01:11:52	2024-02-01 02:39:52	1914	0	Credit Card	95.70
1	2	C1860	R2054	2024-02-02 22:11:04	2024-02-02 22:46:04	986	40	Digital Wallet	98.60
2	3	C6390	R2870	2024-01-31 05:54:35	2024-01-31 06:52:35	937	30	Cash on Delivery	140.55
3	4	C6191	R2642	2024-01-16 22:52:49	2024-01-16 23:38:49	1463	50	Cash on Delivery	NaN
4	5	C6734	R2799	2024-01-29 01:19:30	2024-01-29 02:48:30	1992	30	Cash on Delivery	50.00

Cost and Profitability Analysis

In [144...
```
df["Costs"] = df["Delivery Fee"] + df['Discounts and Offers'] + df["Payment Process
```

In [145...
```
df["Revenue"]=df["Commission Fee"]
```

In [146...
```
df["Profit"] = df["Revenue"] - df['Costs']
df.head()
```

Out[146…

	Order ID	Customer ID	Restaurant ID	Order Date and Time	Delivery Date and Time	Order Value	Delivery Fee	Payment Method	Discounts and Offers
0	1	C8270	R2924	2024-02-01 01:11:52	2024-02-01 02:39:52	1914	0	Credit Card	95.70
1	2	C1860	R2054	2024-02-02 22:11:04	2024-02-02 22:46:04	986	40	Digital Wallet	98.60
2	3	C6390	R2870	2024-01-31 05:54:35	2024-01-31 06:52:35	937	30	Cash on Delivery	140.55
3	4	C6191	R2642	2024-01-16 22:52:49	2024-01-16 23:38:49	1463	50	Cash on Delivery	NaN
4	5	C6734	R2799	2024-01-29 01:19:30	2024-01-29 02:48:30	1992	30	Cash on Delivery	50.00

In [147…
```
df["Profit"].sum()
```

Out[147…
```
-17975.85
```

In [148…
```
df["Costs"].sum()
```

Out[148…
```
121773.85
```

In [149…
```
df["Revenue"].sum()
```

Out[149…
```
126990
```

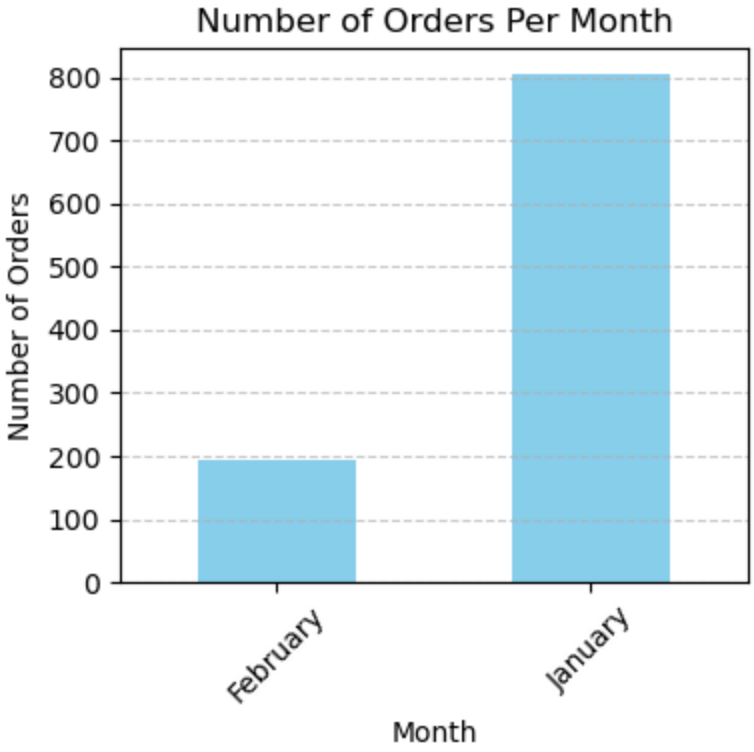
In [150…
```
cost_dist = df[["Delivery Fee", "Payment Processing Fee", "Discounts and Offers"]].
cost_dist
```

Out[150…
```
Delivery Fee             28620.00
Payment Processing Fee   29832.00
Discounts and Offers     74289.85
dtype: float64
```

Visualization

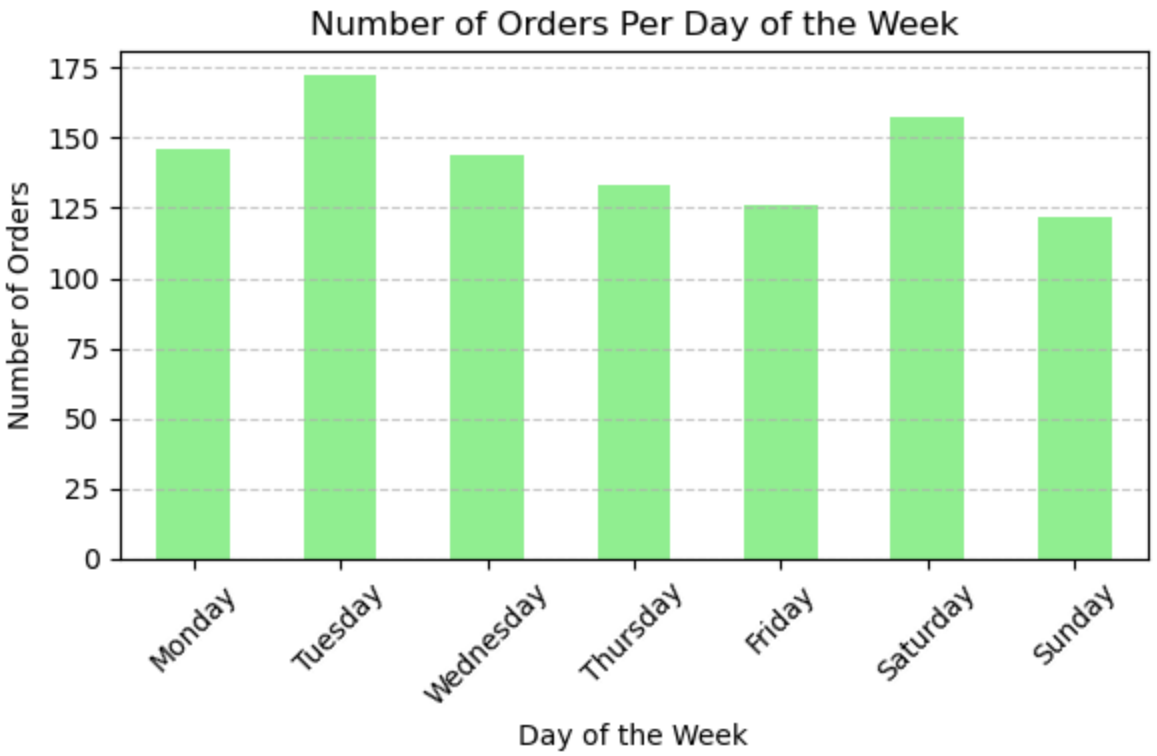
In [151…
```
import matplotlib.pyplot as plt

# Aggregate the number of orders by month
monthly_orders = df['Order Month'].value_counts().sort_index()
```

```
# Plot the number of orders per month
plt.figure(figsize=(4,4))
monthly_orders.plot(kind='bar', color='skyblue')
plt.title('Number of Orders Per Month')
plt.xlabel('Month')
plt.ylabel('Number of Orders')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()

# Show the plot
plt.show()
```



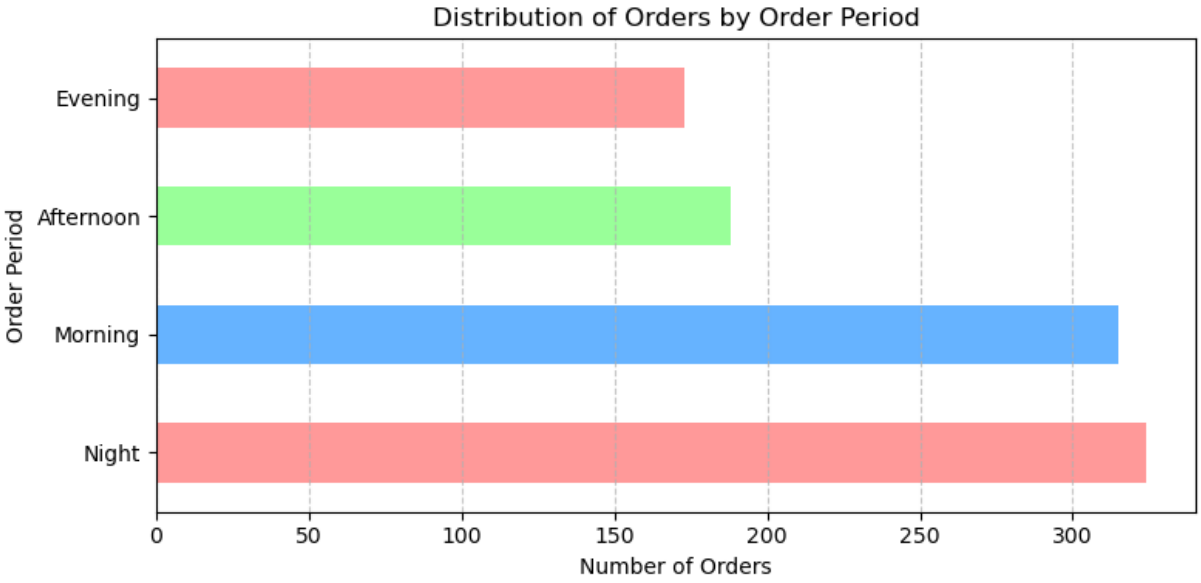
In [152…

```
# Aggregate the number of orders by day of the week
daily_orders = df['Order Day'].value_counts().reindex(['Monday', 'Tuesday', 'Wednes

# Plot the number of orders per day of the week
plt.figure(figsize=(6,4))
daily_orders.plot(kind='bar', color='lightgreen')
plt.title('Number of Orders Per Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Orders')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

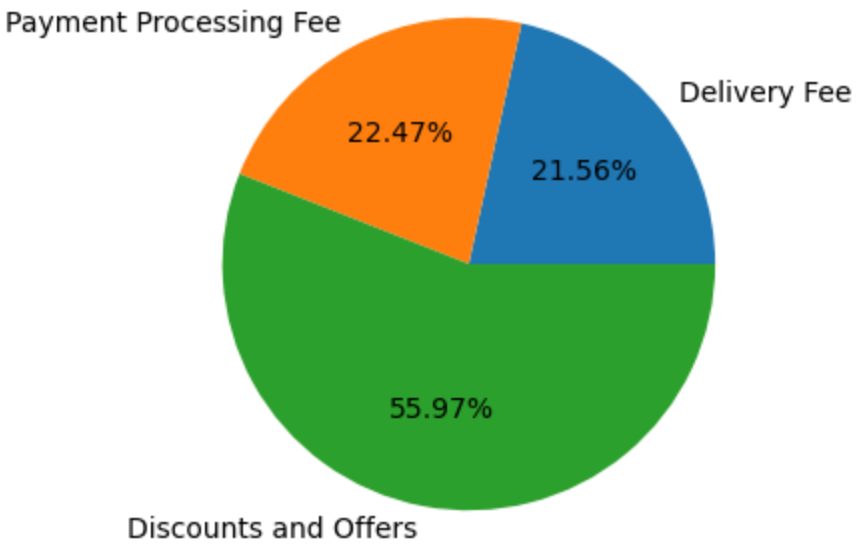


In [153…

```
# Count the number of orders for each order period (e.g., Morning, Evening, Night)
order_period_counts = df['Order Period'].value_counts()

plt.figure(figsize=(8, 4))
order_period_counts.plot(kind='barh', color=['#ff9999','#66b3ff','#99ff99'])
plt.title('Distribution of Orders by Order Period')
plt.xlabel('Number of Orders')
plt.ylabel('Order Period')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



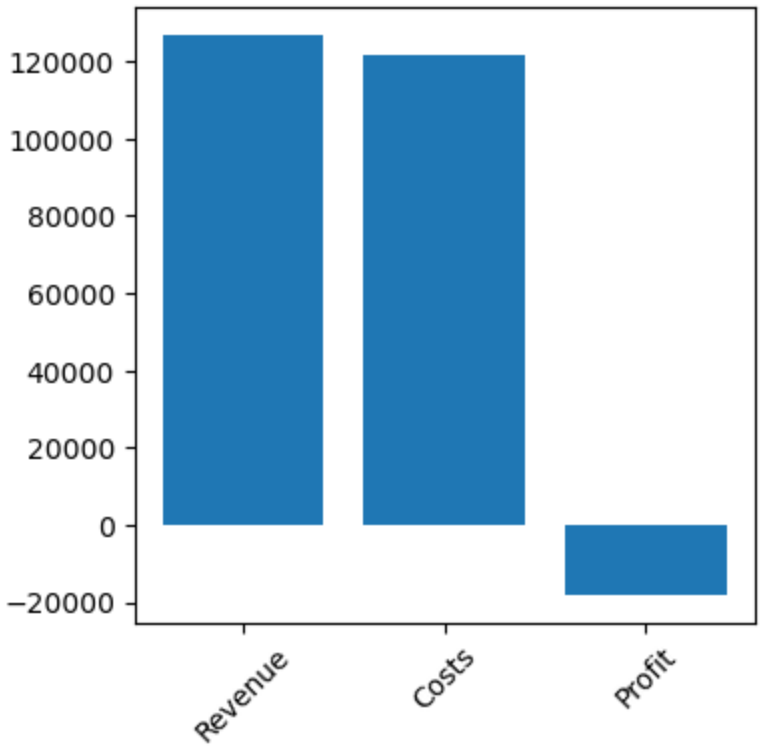
In [154…
```
# Distribution of cost
plt.figure(figsize = (4,4))
plt.pie(cost_dist, labels = cost_dist.index, autopct = "%1.2f%%")
plt.show()
```



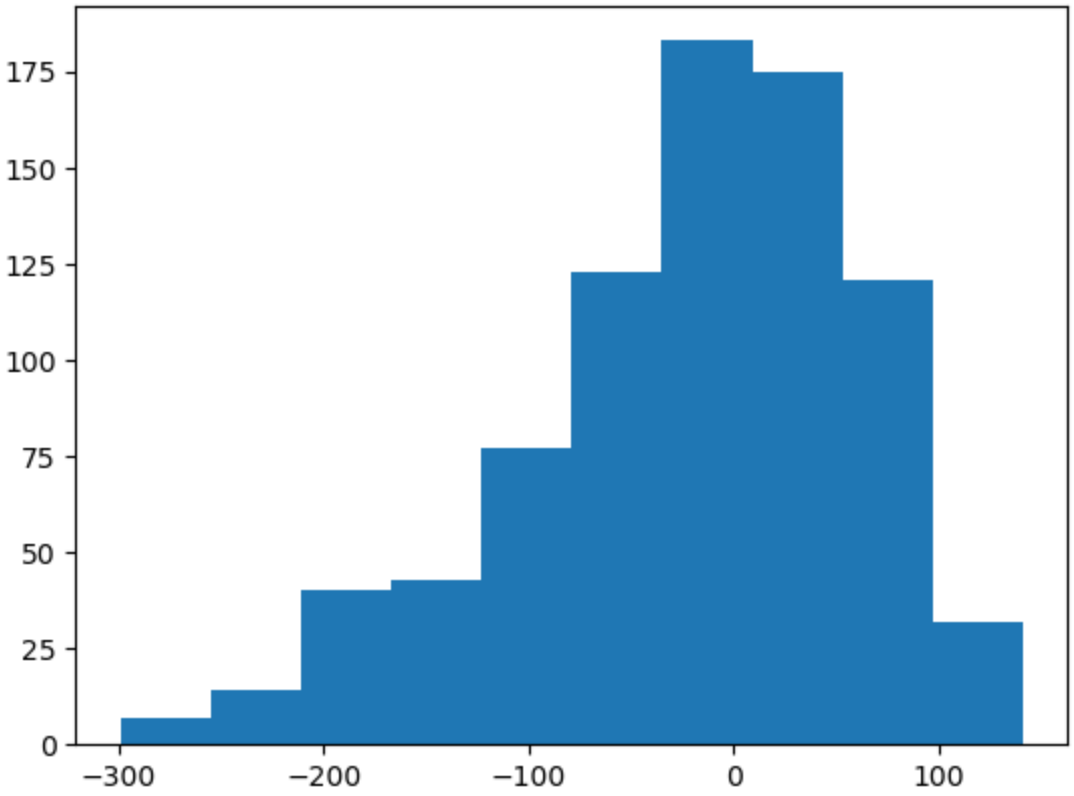
In [155…
```
Data = df[["Revenue", "Costs","Profit"]].sum()
Data
```

Out[155…
```
Revenue    126990.00
Costs      121773.85
Profit     -17975.85
dtype: float64
```

In [156…
```
plt.figure(figsize = (4,4))
plt.bar(Data.index, Data)
plt.xticks(rotation = 45)
plt.show()
```

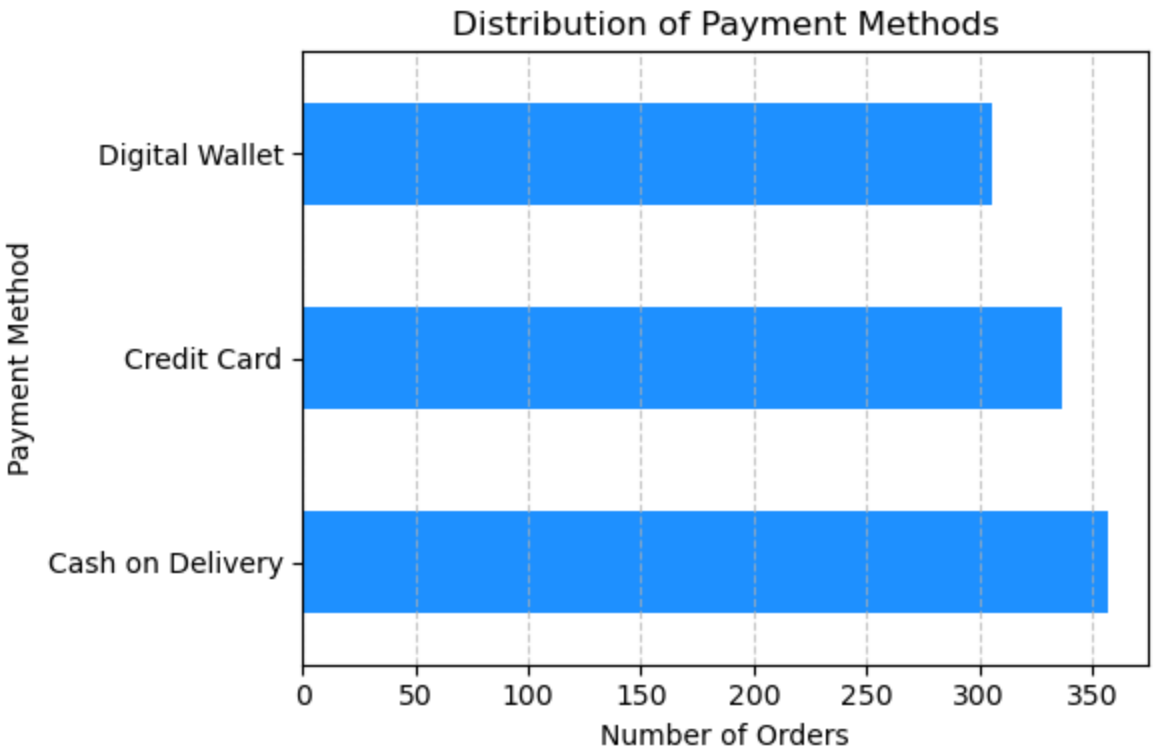


In [157…
```
plt.hist(df["Profit"])
plt.show()
```



In [158…
```
# Count the occurrences of each payment method
payment_methods = df['Payment Method'].value_counts()

# Plot the distribution of payment methods as a horizontal bar chart
```

```
plt.figure(figsize=(6, 4))
payment_methods.plot(kind='barh', color='dodgerblue')
plt.title('Distribution of Payment Methods')
plt.xlabel('Number of Orders')
plt.ylabel('Payment Method')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



In [159…

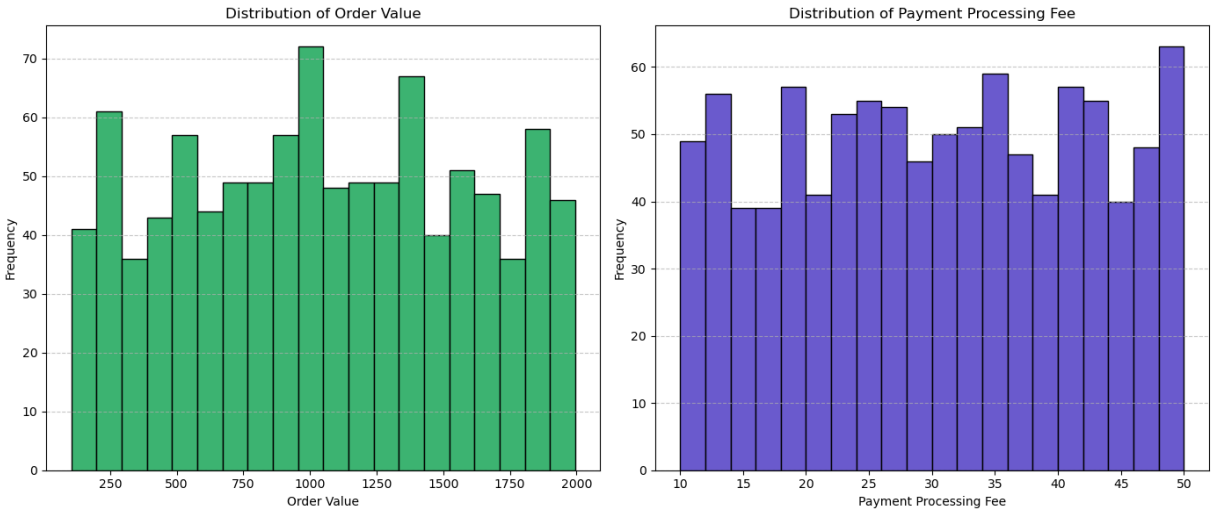
```
# Plot the distribution of discounts and offers using a histogram
plt.figure(figsize=(6, 4))
plt.hist(df['Discounts and Offers'], bins=20, color='lightgreen', edgecolor='black'
plt.title('Distribution of Discounts and Offers')
plt.xlabel('Discounts and Offers')
plt.ylabel('Frequency')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

In [160…

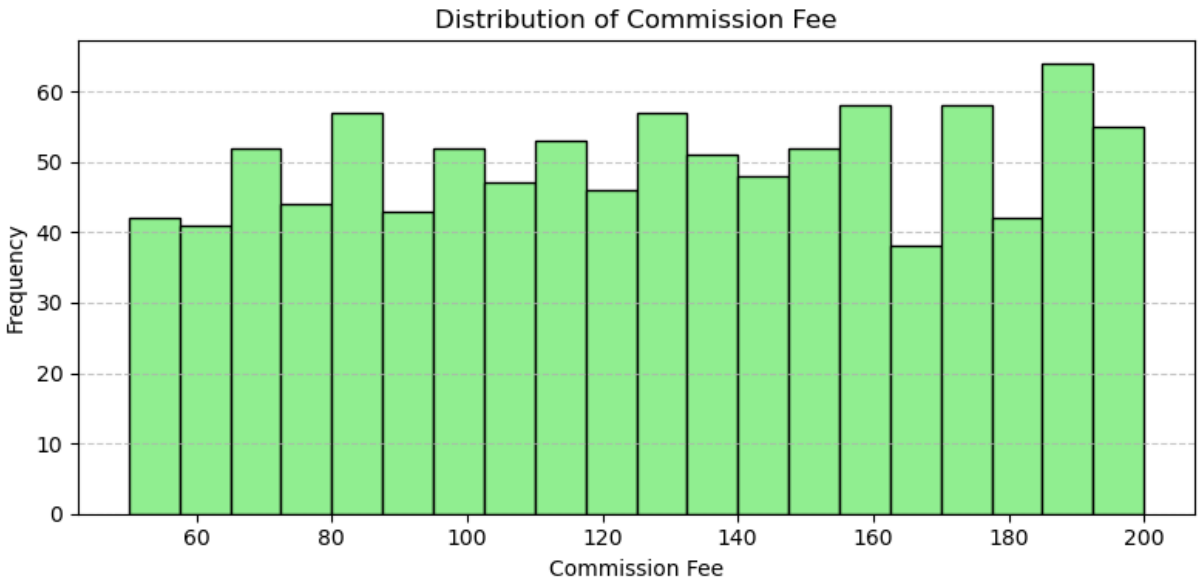
```
# Plot the distribution of the delivery fee using a histogram
plt.figure(figsize=(6, 4))
plt.hist(df['Delivery Fee'], bins=20, color='lightgreen', edgecolor='black')
plt.title('Distribution of Delivery Fee')
plt.xlabel('Delivery Fee')
plt.ylabel('Frequency')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

In [161…

```
# Plot the distribution of order value using a histogram
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)  # First subplot for Order Value
plt.hist(df['Order Value'], bins=20, color='mediumseagreen', edgecolor='black')
plt.title('Distribution of Order Value')
plt.xlabel('Order Value')
plt.ylabel('Frequency')
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.subplot(1, 2, 2)  # Second subplot for Payment Processing Fee
plt.hist(df['Payment Processing Fee'], bins=20, color='slateblue', edgecolor='black
plt.title('Distribution of Payment Processing Fee')
plt.xlabel('Payment Processing Fee')
plt.ylabel('Frequency')
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()

# Show the plots
plt.show()
```



In [162…
```
# Plot the distribution of commission fees using a histogram
plt.figure(figsize=(8, 4))
plt.hist(df['Commission Fee'], bins=20, color='lightgreen',edgecolor='black')
plt.title('Distribution of Commission Fee')
plt.xlabel('Commission Fee')
plt.ylabel('Frequency')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



In [173…
```
output_file_path = r"C:\Users\PMLS\OneDrive\Desktop\MAKE IT\PYTHON FOR DATA ANALYSI
df.to_csv(output_file_path, index=False)
print(f"File saved as {output_file_path}")
```

```
File saved as C:\Users\PMLS\OneDrive\Desktop\MAKE IT\PYTHON FOR DATA ANALYSIS\1st py
thon project\New folder\output_file.csv
```

In [ ]: