

```
In [7]: import pandas as pd
Finance = pd.read_csv("C:/Users/Abhinav/Desktop/DataAnalysisCourseMaterials/DataAnalysis/Finance")
```

Out[7]:

	January	February	March	April	May	June	July	August	September
Month									
Food	3728.0	3536.0	3200.0	1700.0	8076.0	5563.0	5718.0	7020.0	4220.0
Party and Travel	2430.0	2454.0	2935.0	NaN	1450.0	2000.0	1400.0	2840.0	2630.0
Netflix/Amazon prime/Spotify	650.0	650.0	650.0	2018.0	NaN	NaN	NaN	900.0	NaN
Theatre	NaN	NaN	NaN	NaN	580.0	1030.0	NaN	NaN	NaN
Current bill/home expenses	1100.0	58.0	NaN	NaN	370.0	6671.0	311.0	NaN	1330.0
Mobile recharge	595.0	784.0	307.0	785.0	1880.0	249.0	730.0	20.0	NaN
Clothing/Personal	1362.0	2568.0	730.0	3323.0	1000.0	5403.0	NaN	NaN	980.0
Cash withdrawal	NaN	NaN	NaN	NaN	NaN	2000.0	NaN	340.0	NaN
Gift	4376.0	4104.0	4104.0	2389.0	500.0	6544.0	605.0	120.0	NaN
Electronic/Homelab	NaN	NaN	NaN	2200.0	150.0	NaN	NaN	NaN	NaN
Fuel/bike maintenance	650.0	500.0	540.0	625.0	910.0	1524.0	400.0	400.0	400.0
Transportation/RW Pass	112.0	1075.0	705.0	700.0	1815.0	1205.0	1175.0	1866.0	1200.0
Cricket/football nets/swimming/Eco-park	400.0	348.0	NaN	NaN	125.0	NaN	NaN	NaN	NaN
Shoes/Slippers/socks	NaN	2000.0	288.0	800.0	480.0	NaN	NaN	NaN	2490.0
IELTS/GRE/Academic	14700.0	NaN	NaN	NaN	NaN	NaN	10000.0	NaN	NaN
Cosmetics/Body Care/haircut	NaN	830.0	4500.0	1808.0	NaN	1327.0	800.0	1738.0	1240.0
Books	NaN	NaN	NaN	950.0	NaN	NaN	NaN	NaN	NaN
Medical expenses	NaN	NaN	NaN	2763.0	NaN	NaN	NaN	NaN	NaN
Donation	NaN	NaN	NaN	NaN	NaN	NaN	1000.0	300.0	NaN
Trip/Holiday	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Income Tax	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5000.0	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Total	30103.0	18907.0	17959.0	20061.0	17336.0	33516.0	22139.0	20544.0	14640.0

```
In [12]: Finance.fillna(0, inplace=True)
Finance
```

Out[12]:

	January	February	March	April	May	June	July	August	Septem
Month									
Food	3728.0	3536.0	3200.0	1700.0	8076.0	5563.0	5718.0	7020.0	42
Party and Travel	2430.0	2454.0	2935.0	0.0	1450.0	2000.0	1400.0	2840.0	26
Netflix/Amazon prime/Spotify	650.0	650.0	650.0	2018.0	0.0	0.0	0.0	900.0	
Theatre	0.0	0.0	0.0	0.0	580.0	1030.0	0.0	0.0	
Current bill/home expenses	1100.0	58.0	0.0	0.0	370.0	6671.0	311.0	0.0	13
Mobile recharge	595.0	784.0	307.0	785.0	1880.0	249.0	730.0	20.0	
Clothing/Personal	1362.0	2568.0	730.0	3323.0	1000.0	5403.0	0.0	0.0	9
Cash withdrawal	0.0	0.0	0.0	0.0	0.0	2000.0	0.0	340.0	
Gift	4376.0	4104.0	4104.0	2389.0	500.0	6544.0	605.0	120.0	
Electronic/Homelab	0.0	0.0	0.0	2200.0	150.0	0.0	0.0	0.0	
Fuel/bike maintenance	650.0	500.0	540.0	625.0	910.0	1524.0	400.0	400.0	4
Transportation/RW Pass	112.0	1075.0	705.0	700.0	1815.0	1205.0	1175.0	1866.0	12
Cricket/football nets/swimming/Eco-park	400.0	348.0	0.0	0.0	125.0	0.0	0.0	0.0	
Shoes/Slippers/socks	0.0	2000.0	288.0	800.0	480.0	0.0	0.0	0.0	24
IELTS/GRE/Academic	14700.0	0.0	0.0	0.0	0.0	0.0	10000.0	0.0	
Cosmetics/Body Care/haircut	0.0	830.0	4500.0	1808.0	0.0	1327.0	800.0	1738.0	12
Books	0.0	0.0	0.0	950.0	0.0	0.0	0.0	0.0	
Medical expenses	0.0	0.0	0.0	2763.0	0.0	0.0	0.0	0.0	
Donation	0.0	0.0	0.0	0.0	0.0	0.0	1000.0	300.0	
Trip/Holiday	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Income Tax	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5000.0	
NaN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Total	30103.0	18907.0	17959.0	20061.0	17336.0	33516.0	22139.0	20544.0	146

In [16]: Finance.size

Out[16]: 276

In [17]: Finance.shape

Out[17]: (23, 12)

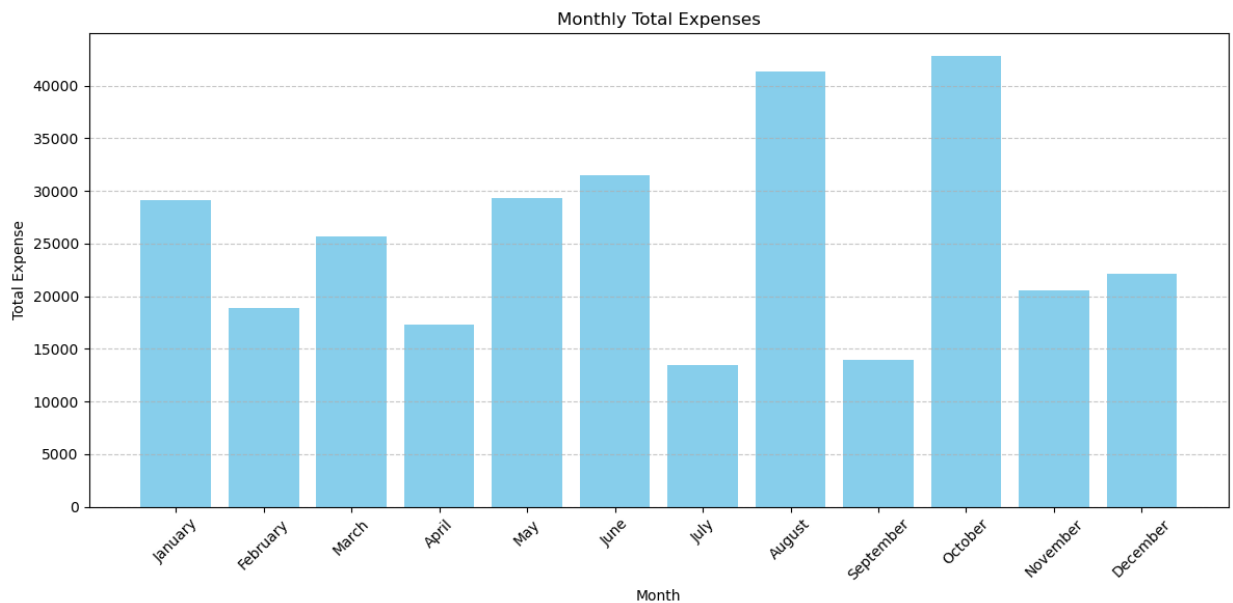
In [13]: Finance.describe()

Out[13]:

	January	February	March	April	May	June	.
count	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000
mean	2617.652174	1644.086957	1561.652174	1744.434783	1507.478261	2914.434783	1925.130
std	6769.151670	3957.295598	3839.404057	4128.327119	3844.963859	7039.130401	4971.400
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
50%	112.000000	348.000000	0.000000	700.000000	150.000000	249.000000	0.000
75%	1231.000000	1537.500000	717.500000	1913.000000	955.000000	2000.000000	900.000
max	30103.000000	18907.000000	17959.000000	20061.000000	17336.000000	33516.000000	22139.000

```
In [31]: # Calculate the total expenses for each month
monthly_total_expenses = df.drop(columns='Month').sum(axis=1)

# Plot a bar plot for monthly total expenses
plt.figure(figsize=(12, 6))
plt.bar(df['Month'], monthly_total_expenses, color='skyblue')
plt.xlabel('Month')
plt.ylabel('Total Expense')
plt.title('Monthly Total Expenses')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



```
In [83]: import matplotlib.pyplot as plt

# Data for total expenses for each month
```

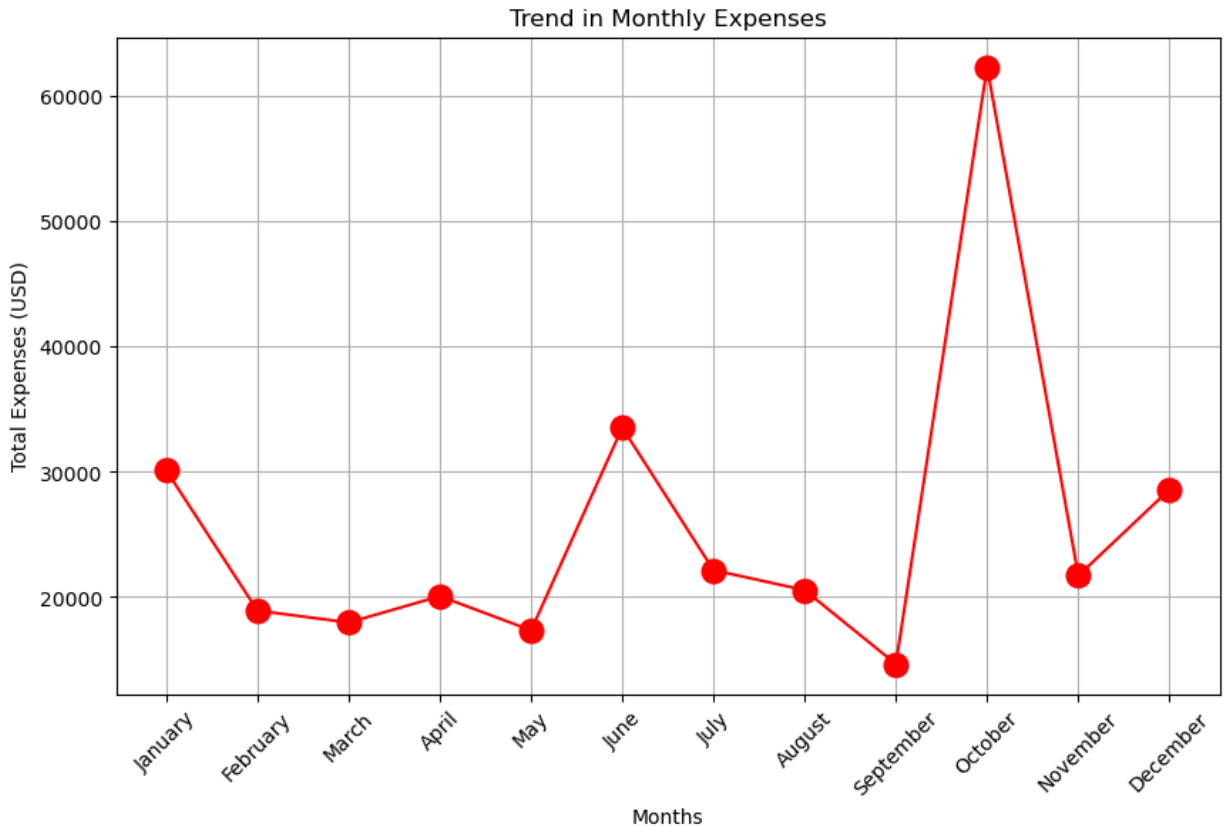
```

months = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"]
total_expenses = [30103.0, 18907.0, 17959.0, 20061.0, 17336.0, 33516.0, 22139.0, 20544.0, 15000.0, 62000.0, 22000.0, 29000.0]

# Create a Line plot
plt.figure(figsize=(10, 6))
plt.plot(months, total_expenses, marker='o', linestyle='-', color='r', markersize=12)
plt.xlabel('Months')
plt.ylabel('Total Expenses (USD)')
plt.title('Trend in Monthly Expenses')
plt.xticks(rotation=45)
plt.grid(True)

# Show the plot
plt.show()

```



In [32]: `import matplotlib.pyplot as plt`

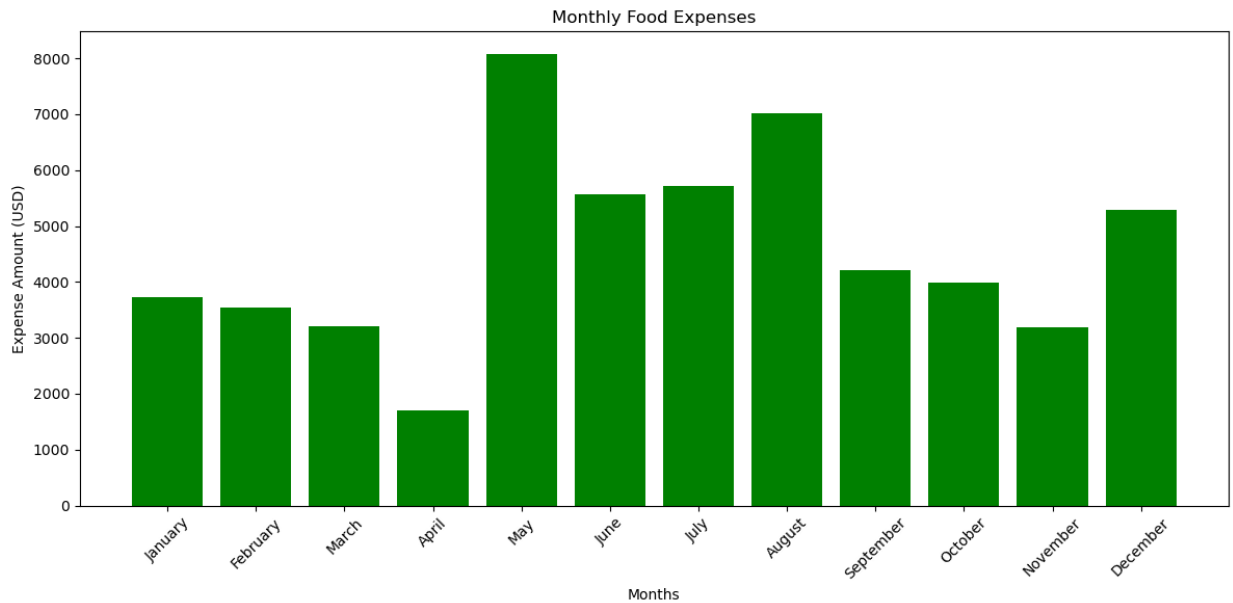
```

months = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"]
food_expenses = [3728.0, 3536.0, 3200.0, 1700.0, 8076.0, 5563.0, 5718.0, 7020.0, 4220.0, 3983.0, 3728.0, 3536.0]

# Creating a bar plot
plt.figure(figsize=(12, 6))
plt.bar(months, food_expenses, color='Green')
plt.xlabel('Months')
plt.ylabel('Expense Amount (USD)')
plt.title('Monthly Food Expenses')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()

```

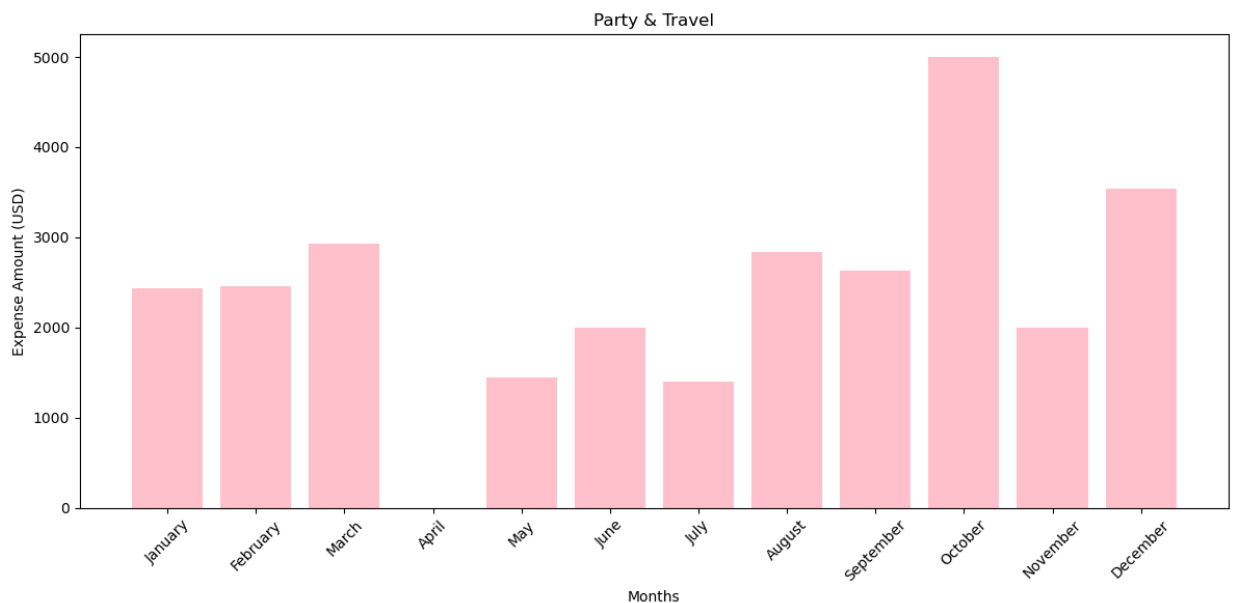


```
In [33]: import matplotlib.pyplot as plt

months = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"]
Expenses = [2430.0, 2454.0, 2935.0, 0.0, 1450.0, 2000.0, 1400.0, 2840.0, 2632.0, 5000.0, 2000.0, 3500.0]

# Creating a bar plot
plt.figure(figsize=(12, 6))
plt.bar(months, Expenses, color='Pink')
plt.xlabel('Months')
plt.ylabel('Expense Amount (USD)')
plt.title('Party & Travel')
plt.xticks(rotation=45)
plt.tight_layout()

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



```
In [90]: import pandas as pd
import matplotlib.pyplot as plt

Finance = {
```

```

"Month": ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"],
"Food": [3728.0, 3536.0, 3200.0, 1700.0, 8076.0, 5563.0, 5718.0, 7020.0, 4220.0, 3728.0, 3536.0, 3200.0],
"Party and Travel": [2430.0, 2454.0, 2935.0, 0.0, 1450.0, 2000.0, 1400.0, 2840.0, 2430.0, 2454.0, 2935.0, 0.0],
"Netflix/Amazon prime/Spotify": [650.0, 650.0, 650.0, 2018.0, 0.0, 0.0, 0.0, 0.0, 650.0, 650.0, 650.0, 2018.0],
"Theatre": [0.0, 0.0, 0.0, 0.0, 580.0, 1030.0, 0.0, 0.0, 0.0, 0.0, 0.0, 86.0],
"Current bill/home expenses": [100.0, 58.0, 0.0, 0.0, 370.0, 6671.0, 311.0, 0.0, 100.0, 58.0, 0.0, 0.0],
"Mobile recharge": [595.0, 784.0, 307.0, 785.0, 1880.0, 249.0, 730.0, 20.0, 480.0, 595.0, 784.0, 307.0],
"Clothing/Personal": [1362.0, 2568.0, 730.0, 3323.0, 1000.0, 5403.0, 989.0, 2000.0, 1362.0, 2568.0, 730.0, 3323.0],
"Cash withdrawal": [0.0, 0.0, 0.0, 0.0, 2000.0, 0.0, 340.0, 6000.0, 130.0, 0.0, 0.0, 0.0],
"Gift": [4376.0, 4104.0, 4104.0, 2389.0, 500.0, 6544.0, 605.0, 120.0, 150.0, 0.0, 0.0, 86.0],
"Electronic/Homelab": [0.0, 0.0, 0.0, 2200.0, 150.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
"Fuel/bike maintenance": [650.0, 500.0, 540.0, 625.0, 910.0, 1524.0, 400.0, 400.0, 650.0, 500.0, 540.0, 625.0],
"Transportation/RW Pass": [112.0, 1075.0, 705.0, 700.0, 1815.0, 1205.0, 1175.0, 1815.0, 112.0, 1075.0, 705.0, 700.0],
"Cricket/football nets/swimming/Eco-park": [400.0, 348.0, 0.0, 0.0, 125.0, 0.0, 0.0, 0.0, 400.0, 348.0, 0.0, 0.0],
"Shoes/Slippers/socks": [0.0, 2000.0, 288.0, 800.0, 480.0, 0.0, 0.0, 0.0, 0.0, 2000.0, 288.0, 800.0],
"IELTS/GRE/Academic": [14700.0, 0.0, 0.0, 0.0, 10000.0, 0.0, 0.0, 19000.0, 0.0, 0.0, 0.0, 0.0],
"Cosmetics/Body Care/haircut": [0.0, 830.0, 4500.0, 1808.0, 0.0, 1327.0, 800.0, 1700.0, 0.0, 830.0, 4500.0, 1808.0],
"Books": [0.0, 0.0, 0.0, 950.0, 0.0, 0.0, 0.0, 0.0, 950.0, 0.0, 0.0, 0.0],
"Medical expenses": [0.0, 0.0, 2763.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
"Donation": [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1000.0, 300.0, 0.0, 500.0, 0.0, 0.0],
"Trip/Holiday": [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 9056.0],
"Income Tax": [0.0, 0.0, 5000.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
}

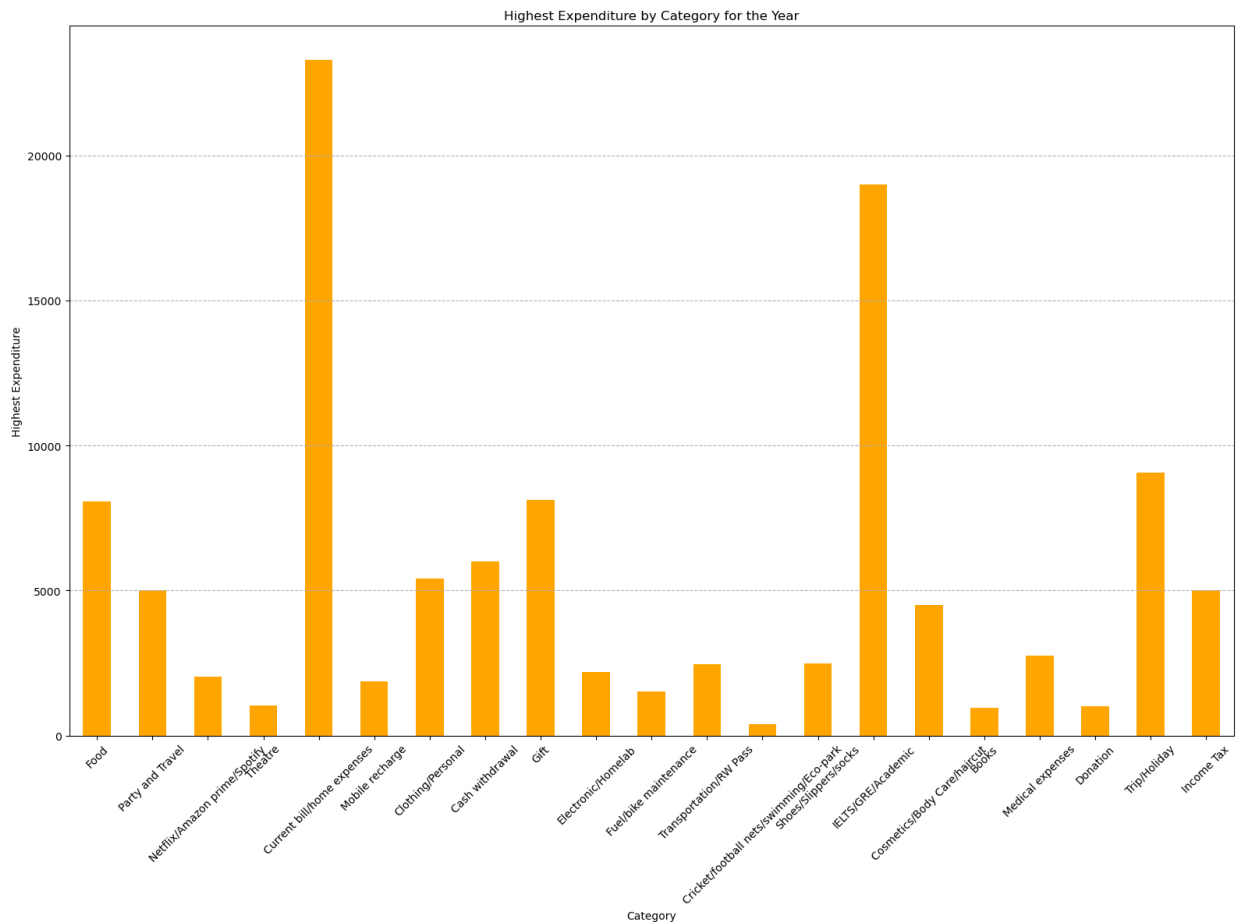
# Create a DataFrame from the data
df = pd.DataFrame(Finance)

# Calculate the highest expenditure for each category for the entire year
category_highest_expenditure = df.drop(columns="Month").max()

# Create a bar plot to visualize the highest expenditures for each category
plt.figure(figsize=(16, 12))
category_highest_expenditure.plot(kind="bar", color="orange")
plt.xlabel('Category')
plt.ylabel('Highest Expenditure')
plt.title('Highest Expenditure by Category for the Year')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=1.0)

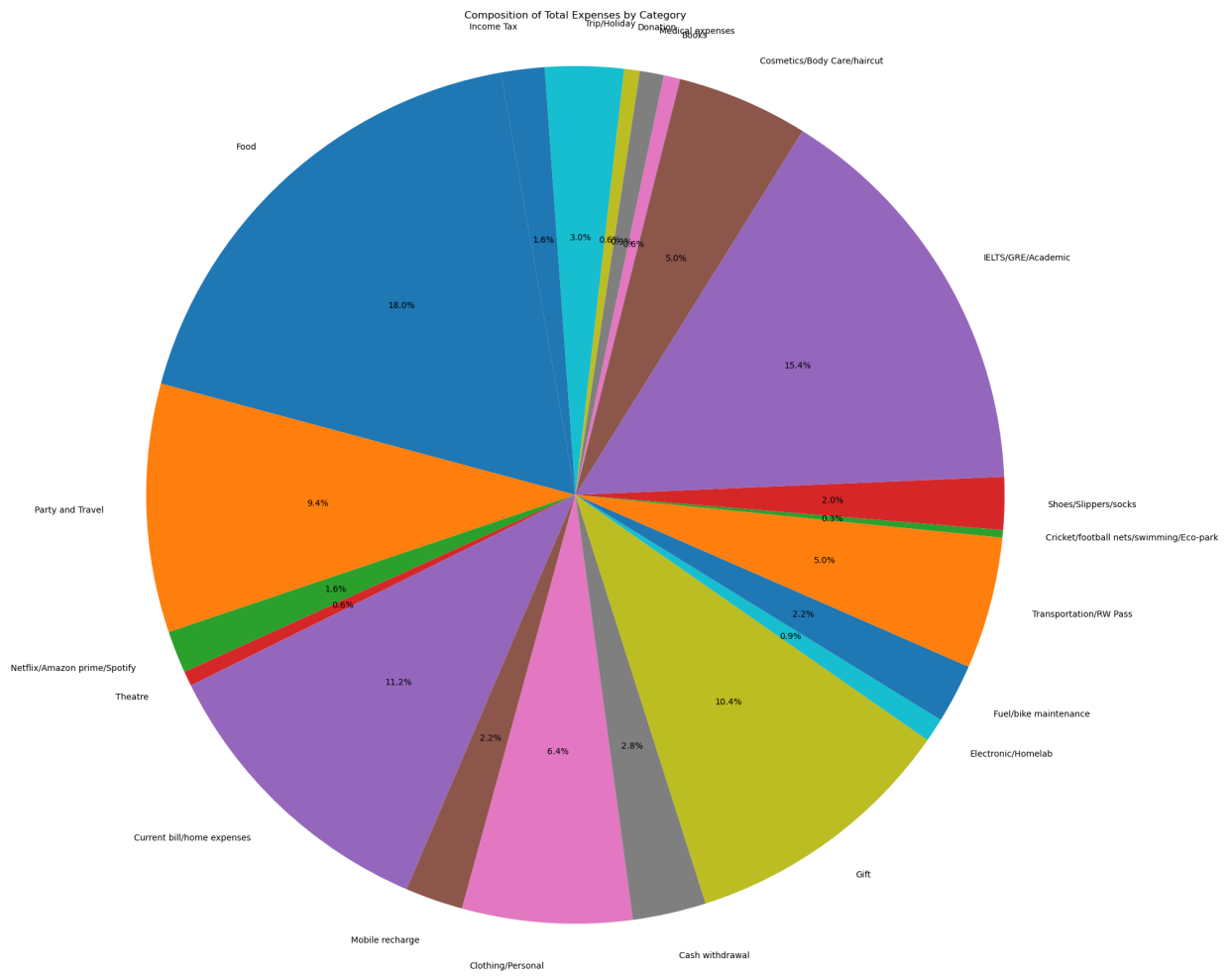
# Show the plot
plt.tight_layout()
plt.show()

```

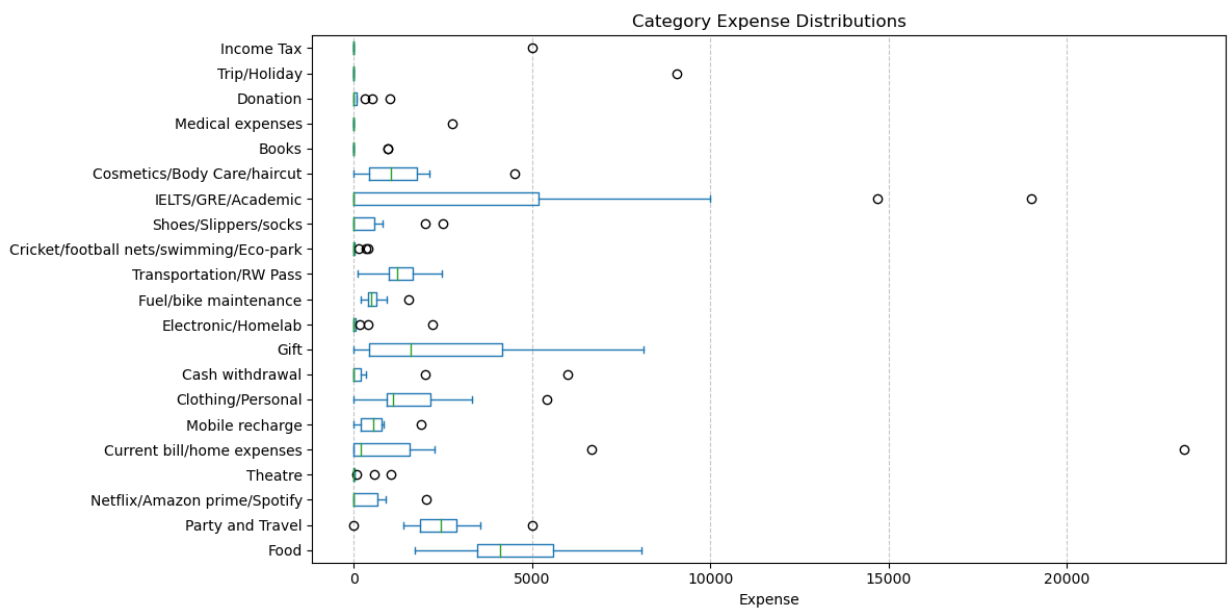


```
In [86]: # Calculate the total expenses for each category for the year
total_expenses = df.drop(columns='Month').sum()

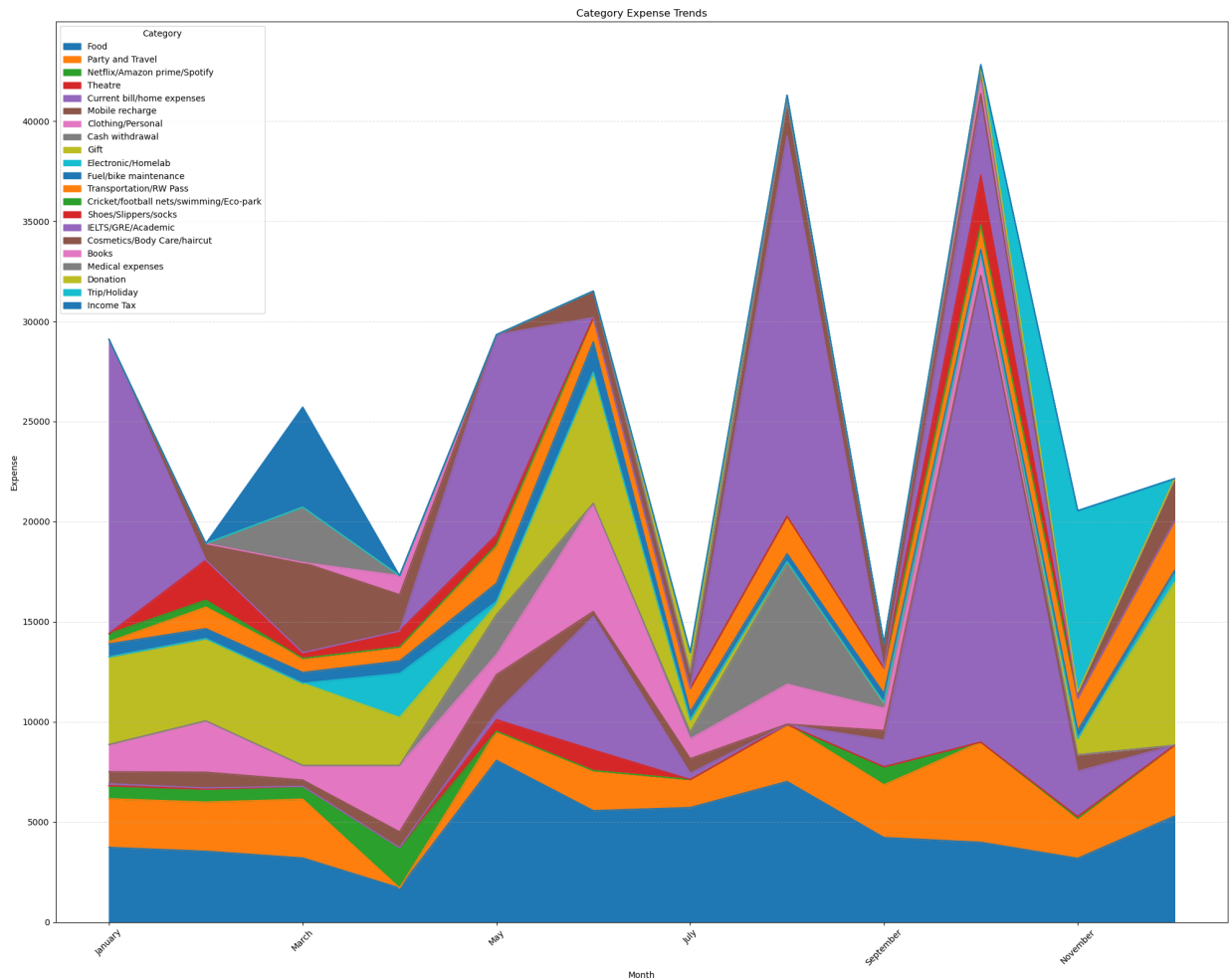
# Plot a pie chart for total expenses
plt.figure(figsize=(24,20))
plt.pie(total_expenses, labels=total_expenses.index, autopct='%1.1f%%', startangle=100)
plt.title('Composition of Total Expenses by Category')
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



```
In [66]: # Plot a box plot for category distributions
df.drop(columns='Month').plot(kind='box', vert=False, figsize=(12, 6))
plt.xlabel('Expense')
plt.title('Category Expense Distributions')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```




```
In [89]: # Plot a stacked area chart for category trends
df.set_index('Month').plot(kind='area', stacked=True, figsize=(20, 16))
plt.xlabel('Month')
plt.ylabel('Expense')
plt.title('Category Expense Trends')
plt.xticks(rotation=45)
plt.legend(title='Category', loc='upper left')
plt.grid(axis='y', linestyle='--', alpha=0.4)
plt.tight_layout()
plt.show()
```



```
In [ ]:
```