What will you learn ?

Here you will learn how to plot different types of Graphs in Matplotlib Library and also different basic types of parameters used to plot the graphs,with the Visual Insights.

Topics covered:

a) Bar Graph
b) Line Plot Graph
c) Pie Chart Graph
d) Box Plot Graph
e) Histogram Graph
f) Violin Plot Graph
g) Stem Plot Graph
h) Stack Plot Graph
i) Step Plot Graph
j) Scatter Plot Graph
k) Subplots and Savefig

Dataset : Github link

https://github.com/tech-dheer-98 (https://github.com/tech-dheer-98)

Importing the Librabries

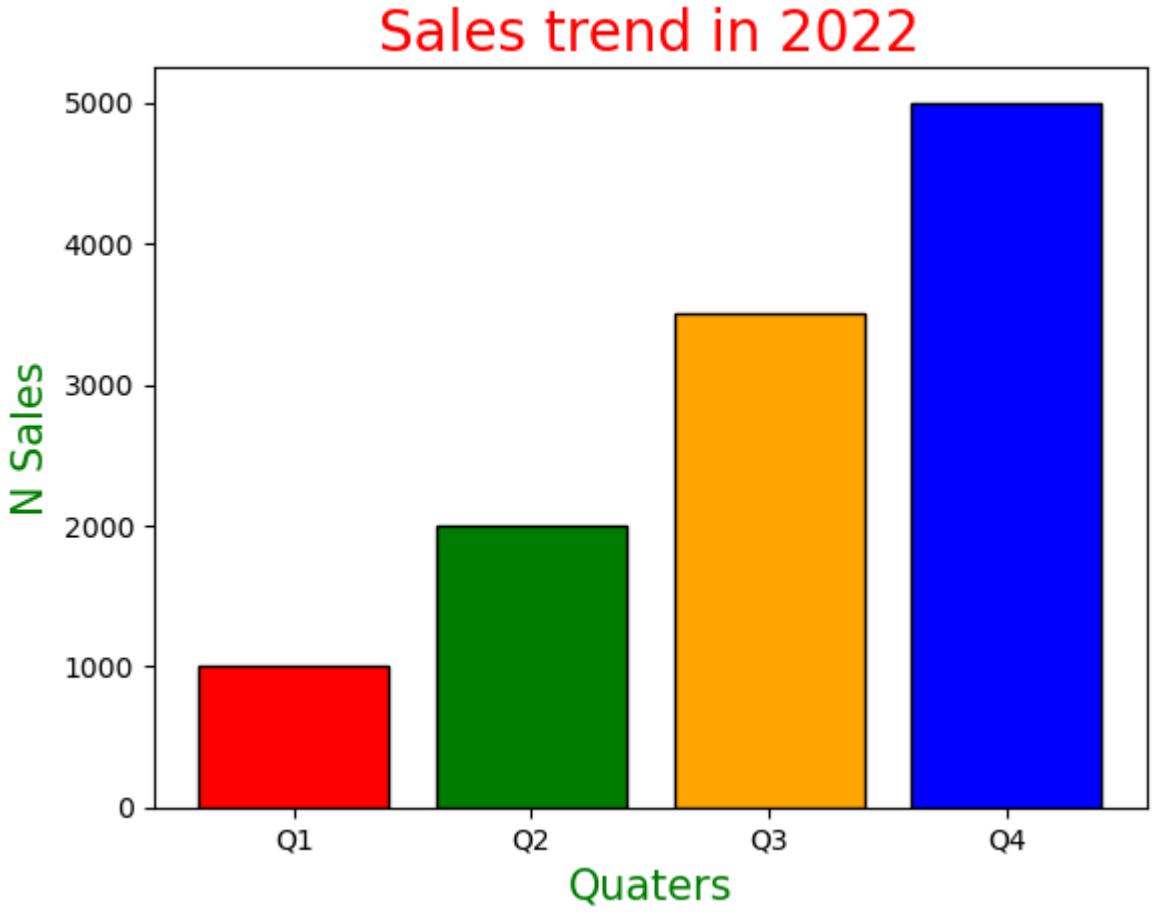
In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
```

a) Bar Chart :

A bar chart is a graphical representation of data that uses rectangular bars to compare discrete categories.

In [2]:
```
# Example of Bar Chart: Plot a graph Showing the Sales trend in 2022 Quater wise.
```

In [3]:
```
X=[1000,2000,3500,5000]
Y=["Q1","Q2","Q3","Q4"]
colors=["red","green","orange","blue"]

plt.bar(Y,X,color=colors,edgecolor="black")
plt.title("Sales trend in 2022",fontsize=20,color="Red")
plt.xlabel("Quaters",fontsize=15,color="green")
plt.ylabel("N Sales",fontsize=15,color="green")

plt.show()
```



Visual Insights :

From the above Bar Graph ,we can conclude that the Sales in 2022 was exponentially increasing Quater wise.

In [4]:
```
# Dataset:Importing the "Expense" Dataset.
```

In [5]:
```
data=pd.read_excel("expense3.xlsx")
data.head()  # Reading the firt 5 values in Dataset.
```

Out[5]:

	Date	Category	Sub-Category	Amount	Payment Mode
0	2023-01-01	Grocery	Grocery	30	Cash
1	2023-01-02	Food	Restaurant	890	UPI
2	2023-01-04	123	Zomato	257	NaN
3	2023-01-06	Essentials	Diary	120	UPI
4	2023-01-06	Essentials	Perfume	1500	Cash

In [6]:
```
data['Payment Mode'].value_counts() # Checking for the Different Values
```

Out[6]:
```
UPI     16
Cash     8
Card     1
Name: Payment Mode, dtype: int64
```

In [7]:
```
data.isnull().sum()  # Checking for Null Values , if there are null values remove it
```

Out[7]:
```
Date             0
Category         0
Sub-Category     0
Amount           0
Payment Mode     4
dtype: int64
```

In [8]:
```
data.shape # Validating the Shape before dropping values
```

Out[8]:
```
(29, 5)
```

In [9]:
```
data=data.dropna() #Dropping the Null Values from the Table
```

In [10]:
```
data.isnull().sum()  # Revalidating the null values has been dropped
```

Out[10]:
```
Date             0
Category         0
Sub-Category     0
Amount           0
Payment Mode     0
dtype: int64
```

In [11]:
```
data.shape  # Revalidating the Shape after Dropping the NaN values
```

Out[11]:
```
(25, 5)
```

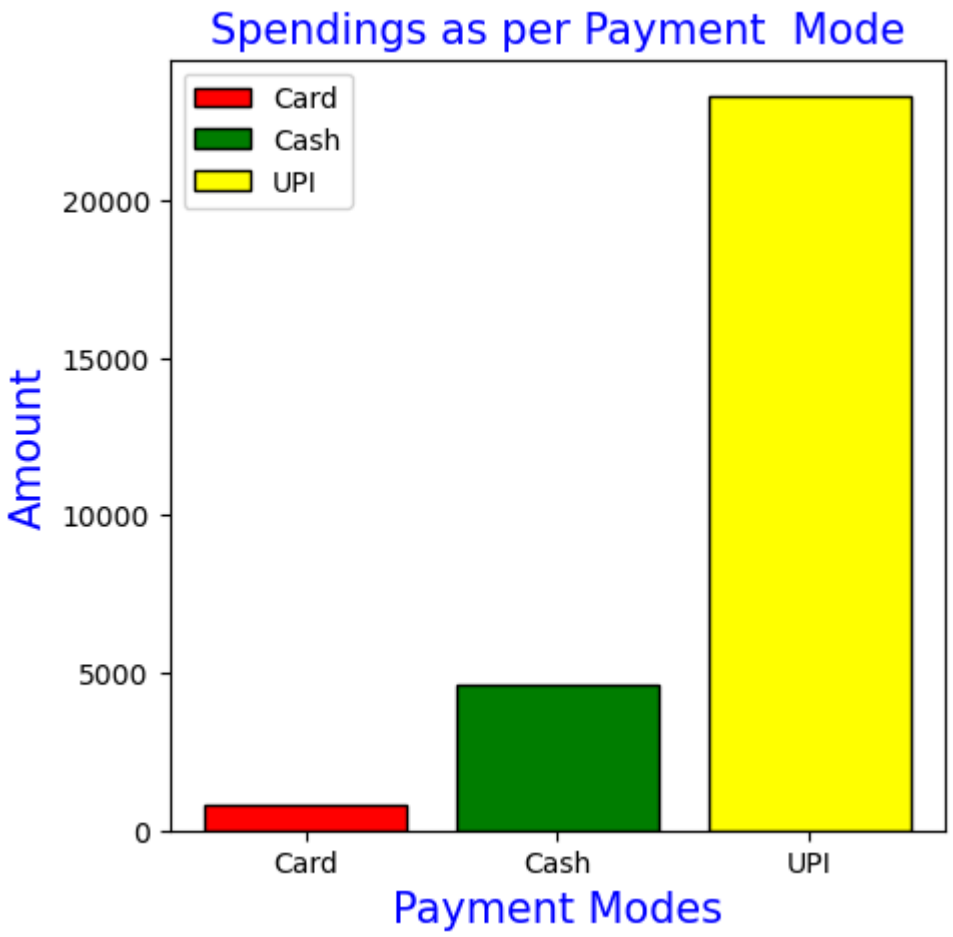
By using the group by function to get the sum of all the Amount wrt Payment methods

In [12]:
```
grouped_by=data.groupby("Payment Mode")["Amount"].sum()
```

In [13]:
```
print(grouped_by) # checking the group by function grouped
```

```
Payment Mode
Card      780
Cash     4637
UPI     23305
Name: Amount, dtype: int64
```

In [14]:
```
plt.figure(figsize=(5,5),dpi=100) # Scaling the Graph
colors=["red","green","yellow"]   # Providing Different colors to the Graph
labels=["Card","Cash","UPI"]      # Labelling the Bars

# Defining the Title and Lables of the Graph
plt.title("Spendings as per Payment  Mode",size=15,color="blue")
plt.xlabel("Payment Modes",size=15,color="blue")
plt.ylabel("Amount",size=15,color="blue")

#Plotting the bar graph
plt.bar(grouped_by.index,grouped_by.values,color=colors,label=labels,edgecolor="black")

plt.legend() # Dispalying the labels in the plt.bar()
plt.show()
```



Visual Insights :

From above bar chart we can conclude that "UPI" Payments have been used extensively.

b) Line Plot Graph:

A line graph, also known as a line chart or curve chart, is a graphical representation of data points connected by straight lines.

In [15]:
```
# Dataset:Importing the "Expense" Dataset.
```

In [16]:
```
data.head(2) # Checking the Dataset
```

Out[16]:

	Date	Category	Sub-Category	Amount	Payment Mode
0	2023-01-01	Grocery	Grocery	30	Cash
1	2023-01-02	Food	Restaurant	890	UPI

In [17]:
```
data.isnull().sum() # Checking for the Null Values
```

Out[17]:
```
Date            0
Category        0
Sub-Category    0
Amount          0
Payment Mode    0
dtype: int64
```

In [18]:
```
data.Category.value_counts() # Checking the Different Values
```

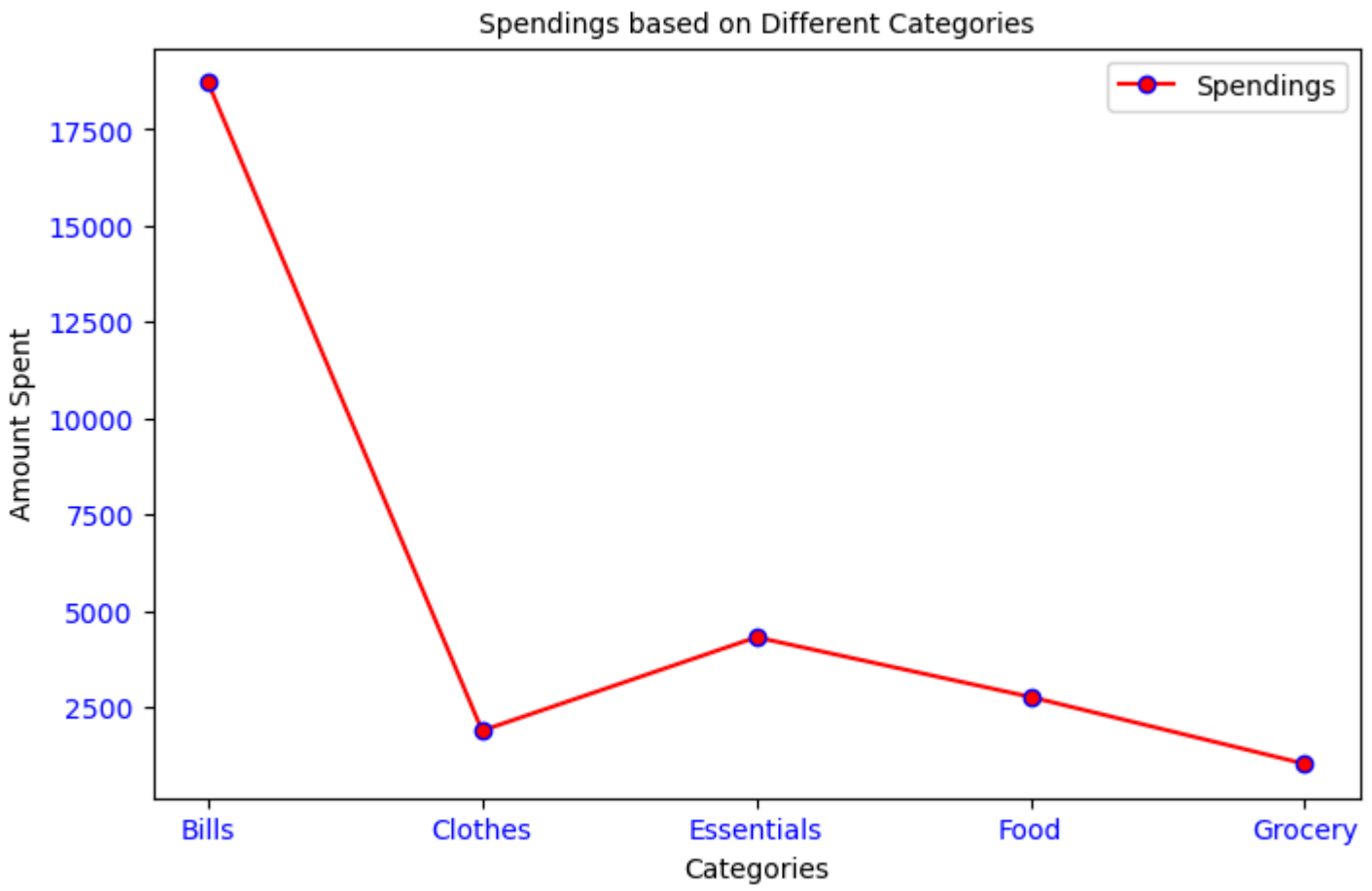
Out[18]:
```
Grocery      8
Food         8
Essentials   5
Bills        3
Clothes      1
Name: Category, dtype: int64
```

In [19]:
```
grouped_by_new=data.groupby("Category")["Amount"].sum() # Using the Group by clause
```

In [20]:
```
print(grouped_by_new) # This shows the Spendings/Amount based on Category basis
```

```
Category
Bills         18724
Clothes        1890
Essentials     4315
Food           2765
Grocery        1028
Name: Amount, dtype: int64
```

In [21]:
```
#Scaling the Graph Size
plt.figure(figsize=(8,5),dpi=100)

# Defining the Graphical Paramenters
plt.plot(grouped_by_new.index,grouped_by_new.values,label="Spendings",color="red",marker='o',linestyle='-',markere
plt.title("Spendings based on Different Categories",color="black",size=10)
plt.xlabel("Categories",color="black",size=10)
plt.ylabel("Amount Spent",color="black",size=10)

#This helps us define x and y scale as per user input choice.
plt.xticks(color="blue")
plt.yticks(color="blue")

plt.legend()

plt.show()
```



Visual Insights :

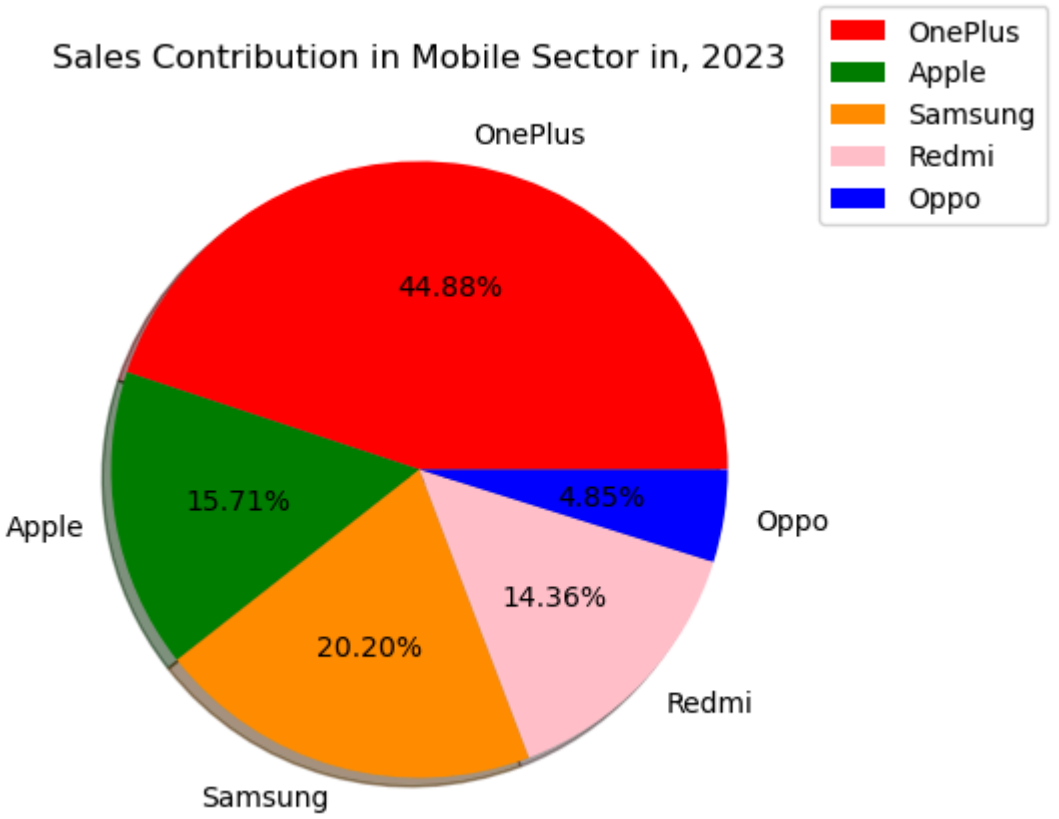
From above line chart we can conclude that maximum spending are seen in Bills Category followed by the Essentials.

c) Pie Chart Graph:

A pie chart is a type of graph that represents data in a circular form. It's also known as a circle chart.

In [22]:
```
# Pie chart Example:
#Highlight the Sales Trend into Mobile Retails  domains where,
#competiors are (Oneplus,Apple,Samsung,Redmi and Oppo) , show their % contribution in year 2023.
```

In [23]:
```
plt.figure(figsize=(8,5),dpi=100)
brands=["OnePlus","Apple","Samsung","Redmi","Oppo"]
Popularity=[10000,3500,4500,3200,1080] # This total numbers should be 100%

color=["red","green","darkorange","pink","blue"]
ex=[0,0,0,0,0] # This explodes the particlular brand from a pie.

#Plotting the Pie Chart#
#explode: It is used to remove  a pie from a chart
#shadow: It is used to give a 3-D effect to a pie chart.
#autopct: It is used to give % contribution in the pie chart
#startangle: It guides the Pie chart from which angle to start.
plt.title("Sales Contribution in Mobile Sector in, 2023",size=12,color="black")
plt.pie(Popularity,labels=brands,explode=ex,colors=color,shadow=True,autopct="%.2f%%",startangle=0)

# Adjusting legend position
plt.legend(loc=3, bbox_to_anchor=(1, 0.8))
#Use bbox_to_anchor: when you get the operlapping of the chart with Legend

plt.show()
```



Visual Insights :

From above Pie Chart we conculde that major Sales is contributed from Oneplus brand contributing to 44.88% of overall Sales,followed by Samsung brand which is contributing 20.20% of overall Sales.

In [24]:
```
# Dataset:Importing the "Expense" Dataset.
```

In [25]:
```
data.head(2)
```

Out[25]:

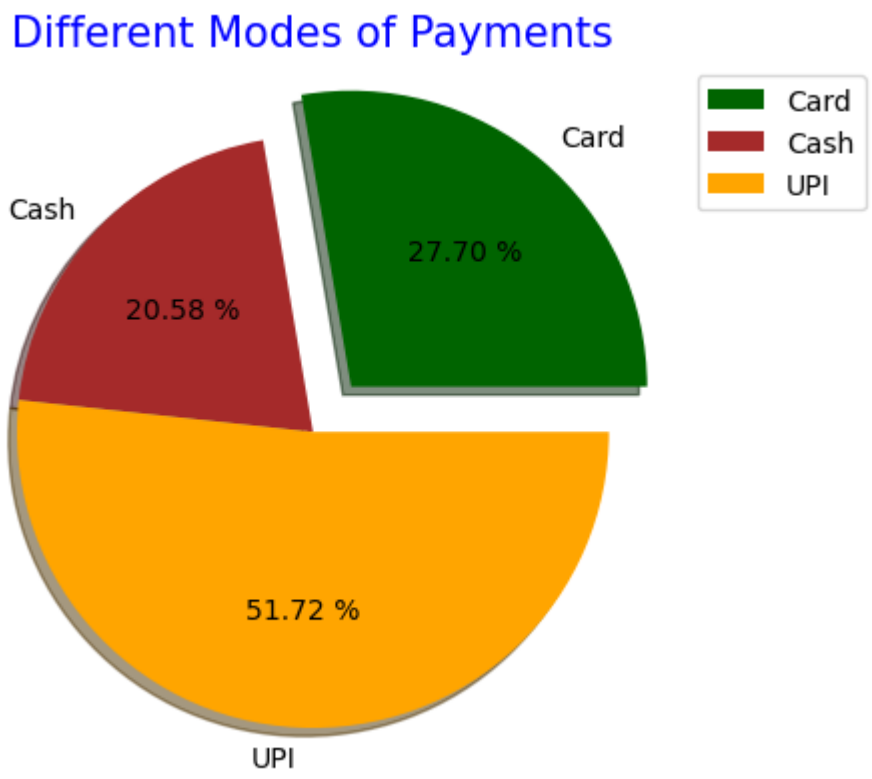
	Date	Category	Sub-Category	Amount	Payment Mode
0	2023-01-01	Grocery	Grocery	30	Cash
1	2023-01-02	Food	Restaurant	890	UPI

In [26]:
```
grouped_new=data.groupby('Payment Mode')["Amount"].mean()
```

In [27]:
```
print(grouped_new)
```

```
Payment Mode
Card     780.0000
Cash     579.6250
UPI     1456.5625
Name: Amount, dtype: float64
```

In [28]:
```
colors=["darkgreen","brown","orange"]
ex=[0.2,0,0]
plt.pie(grouped_new.values,labels=grouped_new.index,colors=colors,autopct="%.2f %%",explode=ex,shadow=True)
plt.title("Different Modes of Payments",color="blue",size=15)

plt.legend(loc=0, bbox_to_anchor=(1,1))
plt.show()
```



Visual Insights :

From the above pie chart , we can conclude that the average Expenses of UPI mode is highest , contributing to 51.72% of total Expenses.

In [ ]:
```

```

d) Box Plot Graph:

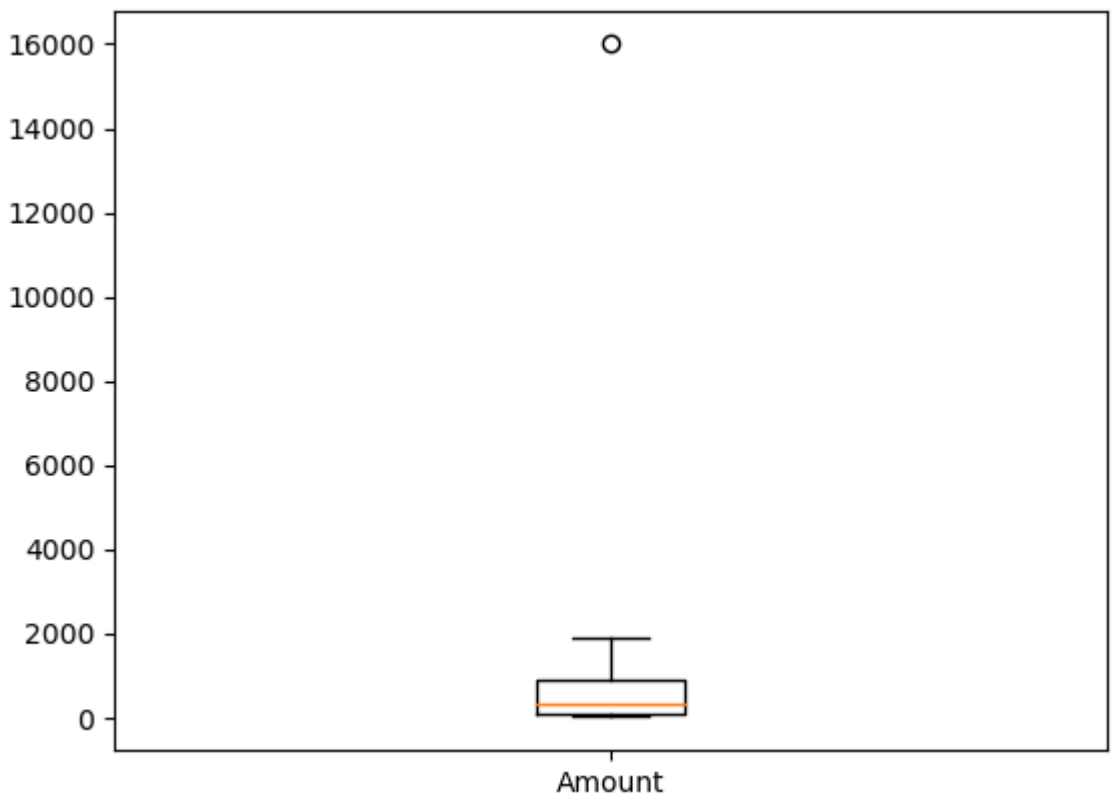
A box plot, also known as a box-and-whisker plot, is a graph that summarizes a set of data. It shows the distribution of a data set based on its five-number summary: minimum, first quartile (Q1), median, third quartile (Q3), and maximum.

Short Explanation:

1. Box Plot contains: Min Fence,Q1@25% ,Median,Q3@75%,Max Fence.
2. Q1=(25/100)*(n+1) ... This is how percentile is calculated ,where n= datavalues, Eg : data=[1,2,3,4,5,6,7,8], here n=8
3. Q3=(75/100)*(n+1) ... This is how percentile is calculated ,where n= datavalues, Eg : data=[1,2,3,4,5,6,7,8], here n=8
4. Min/Lower Fence= Q1-1.5(IQR)... Formula to calculate LF,IQR(Inter Quartile Range)
5. Max/Upper Fence= Q3+1.5(IQR)... Formula to calculate UF,IQR(Inter Quartile Range)
6. IQR is the range between Q1 and Q3 ,IQR=Q3-Q1

In [29]:
```
#Dataset:Importing the "Expense" Dataset.
```

In [30]:
```
# Plot the Box Plot

plt.boxplot(data["Amount"],labels=["Amount"])


plt.show()
```



Visual Insights:

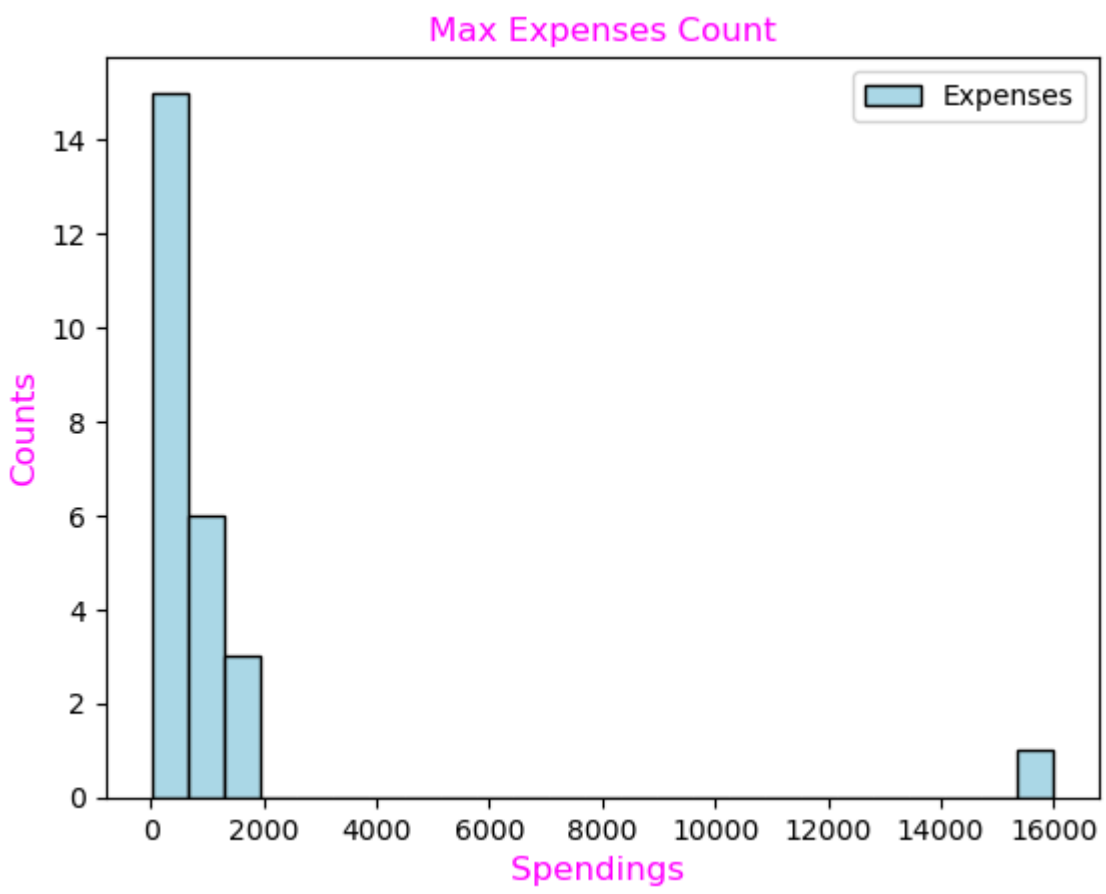
Here we can see that one amount 16k is the outlier

e) Histogram Graph:

A histogram is a column chart that displays frequency data. It's a statistical graph that uses plotted bars to represent the distribution of a continuous dataset.

In [31]:
```
#Dataset:Importing the "Expense" Dataset.
```

In [32]:
```
# Plotting the Histogram

plt.hist(data.Amount,bins=len(data.Amount),color="lightblue",edgecolor="black",label="Expenses")
plt.title("Max Expenses Count",color="magenta",size=12)
plt.xlabel("Spendings ",color="magenta",size=12)
plt.ylabel("Counts",color="magenta",size=12)
plt.legend()

plt.show()
```



Visual Insights :

From the above histogram chart we can conclude that the maximum frequency count of Spendings lie under 2000.

f) Violin Plot Graph :

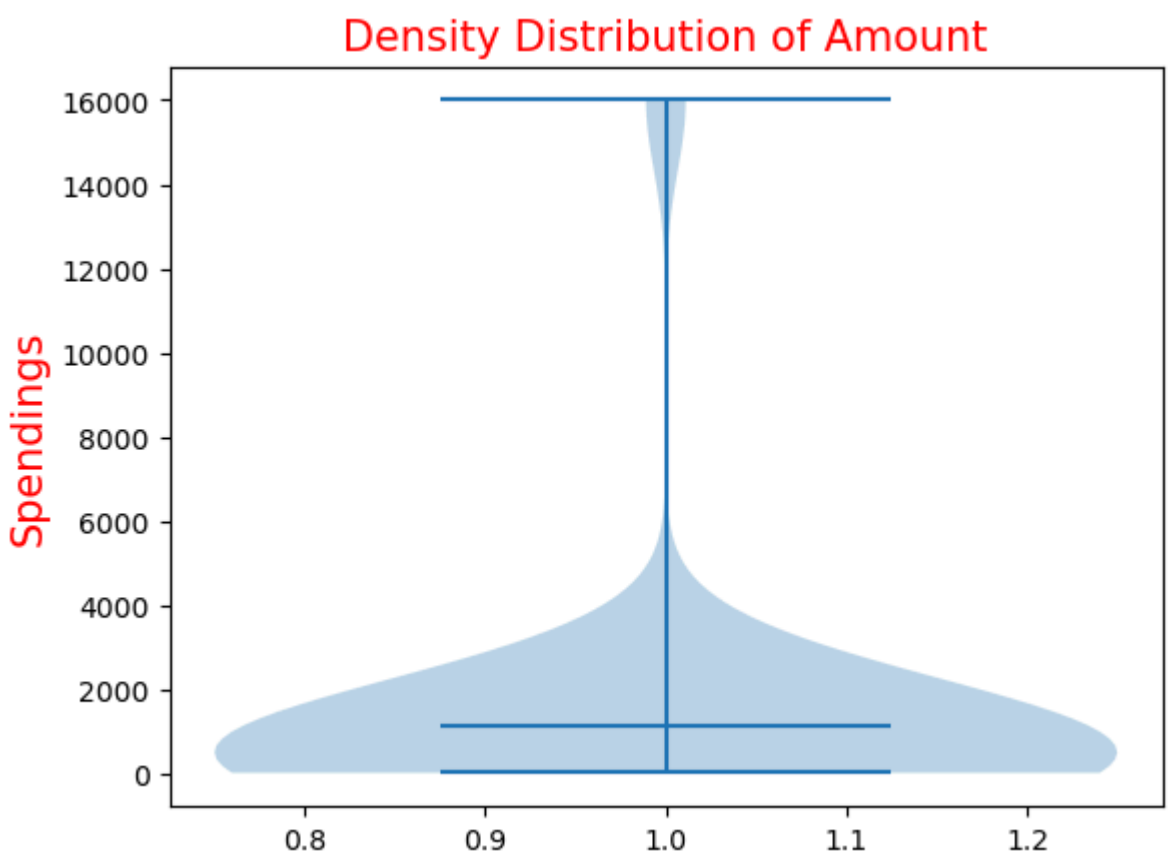
A violin plot depicts distributions of numeric data for one or more groups using density curves. The width of each curve corresponds with the approximate frequency of data points in each region.

In [33]:
```
#Dataset:Importing the "Expense" Dataset.
data.head(2)
```

Out[33]:

	Date	Category	Sub-Category	Amount	Payment Mode
0	2023-01-01	Grocery	Grocery	30	Cash
1	2023-01-02	Food	Restaurant	890	UPI

In [34]:
```
plt.violinplot(data["Amount"],vert=True,showmeans=True)
plt.title("Density Distribution of Amount",size=15,color="red")
plt.ylabel("Spendings",size=15,color="red")
plt.show()
```



Visual Insights :

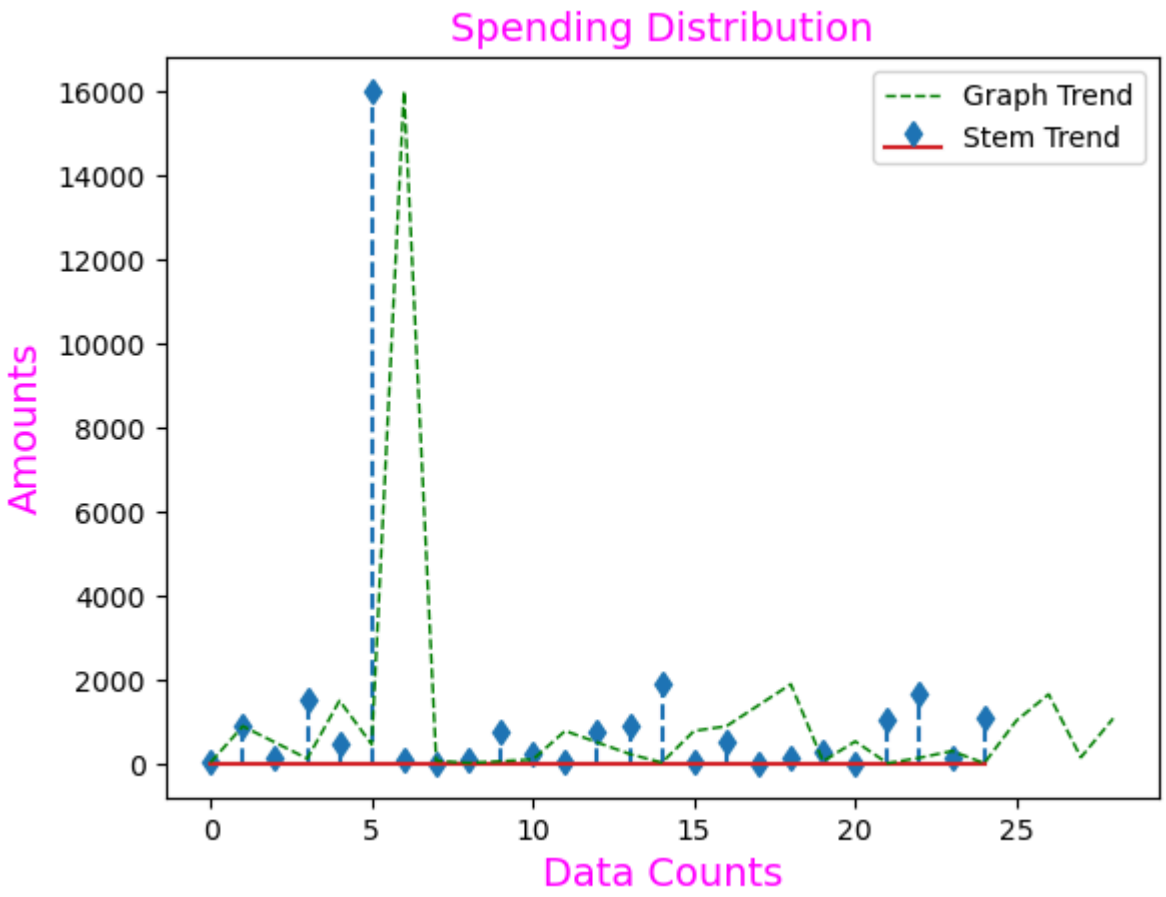
This Violin Graph highlights how the Density of Spendings is distributed in the dataset. Here we can conclude that maximum spread of area of data lies under 2000.

g) Stem Plot Graph:

A stem plot, also known as a stem-and-leaf display, is a graphical representation of quantitative data. It is a two-column diagram that organizes data and shows its frequency distribution.

In [35]:
```
#Dataset:Importing the "Expense" Dataset.
```

In [36]:
```
plt.stem(data.Amount,bottom=0,linefmt="--",markerfmt="d",orientation="vertical",label="Stem Trend")
plt.plot(data.Amount,linewidth=1,color="green",linestyle="--",label="Graph Trend")
plt.title("Spending Distribution",size=14,color="Magenta")
plt.ylabel("Amounts",size=14,color="magenta")
plt.xlabel("Data Counts",size=14,color="magenta")
plt.legend()

plt.show()
```



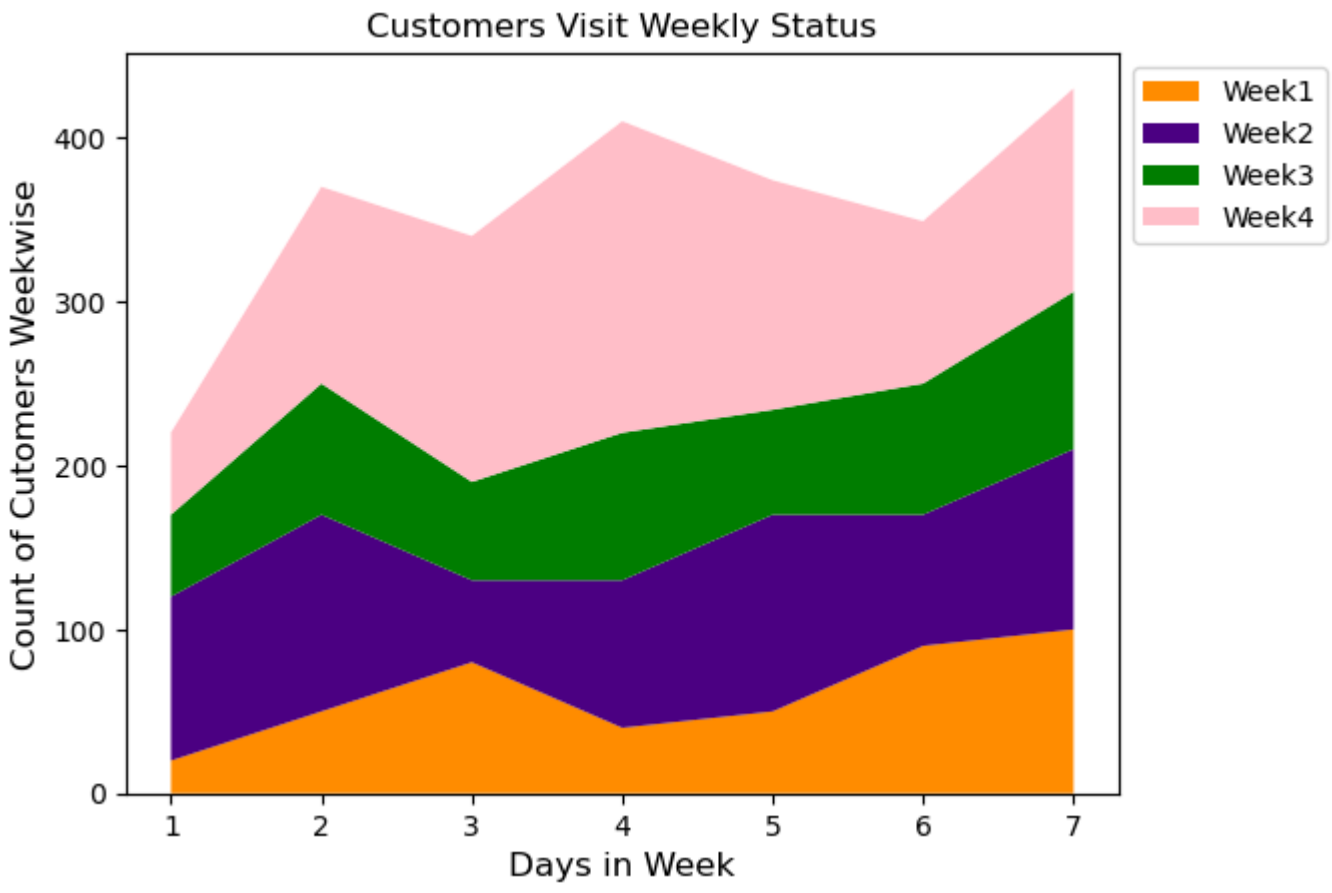
Visual Insights:

Here we conclude from the stem plot ,that maximum data count Spendings, lie under the range of 2000.

h) Stack Plot Graph:

A stack plot is a linear data plot that's stacked vertically, with each plot stacked on top of another. Stack plots are typically used to generate cumulative plots.

In [37]:
```
# Example : Suppose we want to Calculate the No of Customers visiting in a resturant on weekly basis

Days=[1,2,3,4,5,6,7]

Week1=[20,50,80,40,50,90,100]
Week2=[100,120,50,90,120,80,110]
Week3=[50,80,60,90,64,80,96]
Week4=[50,120,150,190,140,99,124]

label=["Week1","Week2","Week3","Week4"]
color=["darkorange","indigo","green","pink"]

plt.stackplot(Days,Week1,Week2,Week3,Week4,labels=label,colors=color)
plt.title("Customers Visit Weekly Status",size=12,color="black")
plt.xlabel("Days in Week",size=12,color="black")
plt.ylabel("Count of Cutomers Weekwise",size=12,color="black")

plt.legend(loc=2, bbox_to_anchor=(1,1))
plt.show()

```



Visuals Insights:

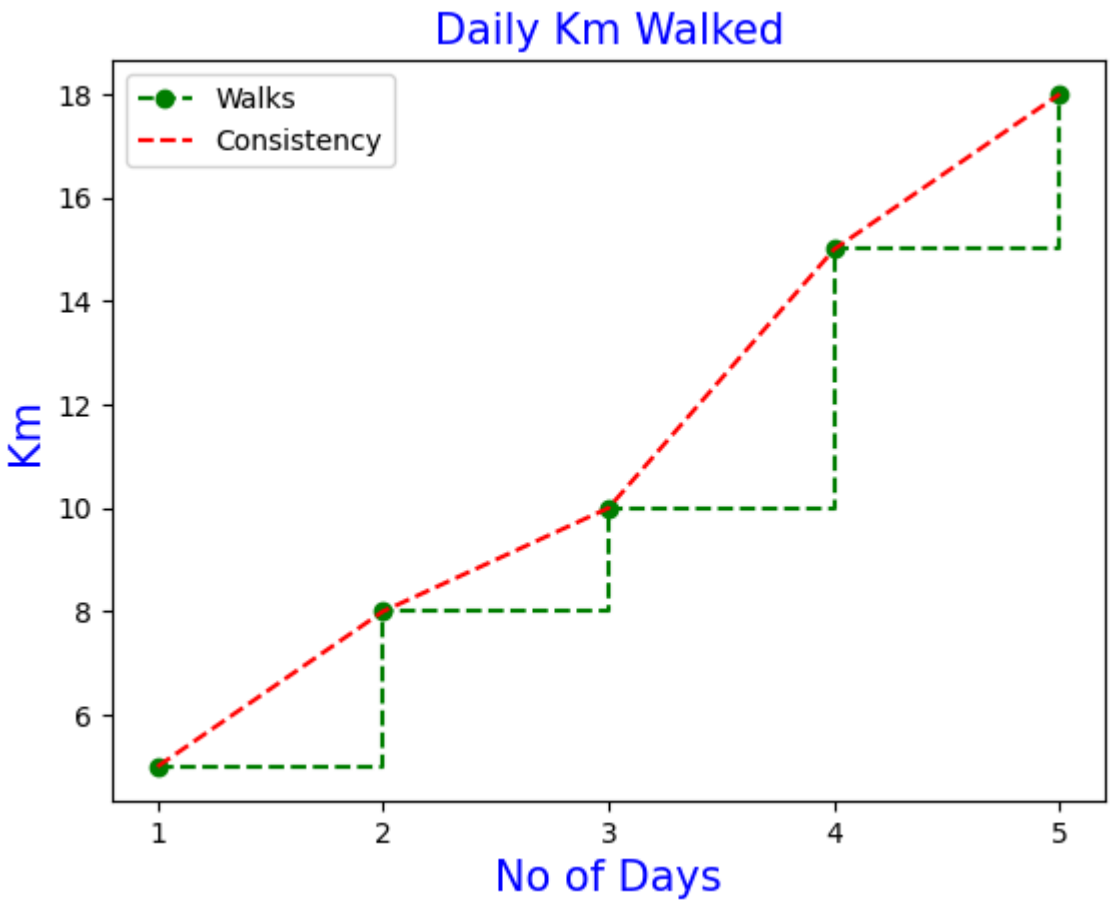
1. Here we can conclude that Week 4 has the highest numbers of cutomers visiting the Resturant.
2. Also we can conclude that Day 7 (Sunday) in a Week seem to be most crowded.

i) Step Plot Graph :

A step plot, also known as a step line plot or stair plot, is a data visualization style that shows data points as a series of horizontal and vertical steps.

In [38]:
```
#Example:Here we will take a data of Daily Walk in Km for 5 Consecutive Days.
```

In [39]:
```
x=[1,2,3,4,5]
y=[5,8,10,15,18]

plt.step(x,y,marker='o',color='green',label='Walks',linestyle='--',where="post")
plt.xlabel("No of Days",color="blue",size=15)
plt.ylabel("Km",color='blue',size=15)
plt.title("Daily Km Walked",color='blue',size=15)
plt.plot(x,y,linestyle='--',label="Consistency",color='red')
plt.xticks([1,2,3,4,5])

plt.legend()

plt.show()
```



Visual Insights :

From the above Step Graph plotted with Line Graph we can conclude that there is consistent increase in the Trend of Walking.

In [40]:
```
##Dataset:Importing the "Expense" Dataset.

data.head(2)
```

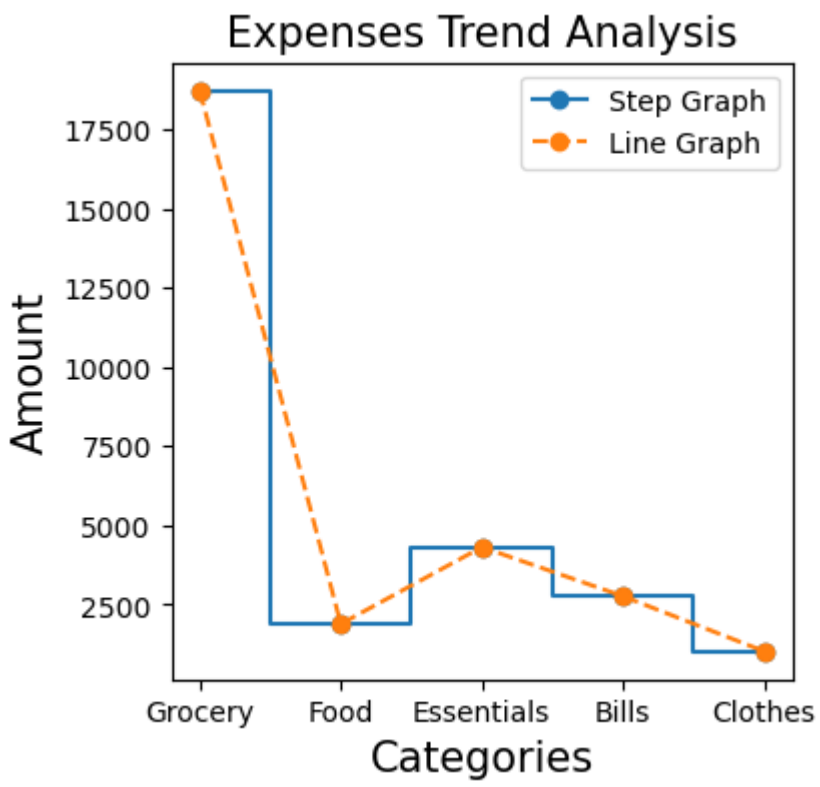
Out[40]:

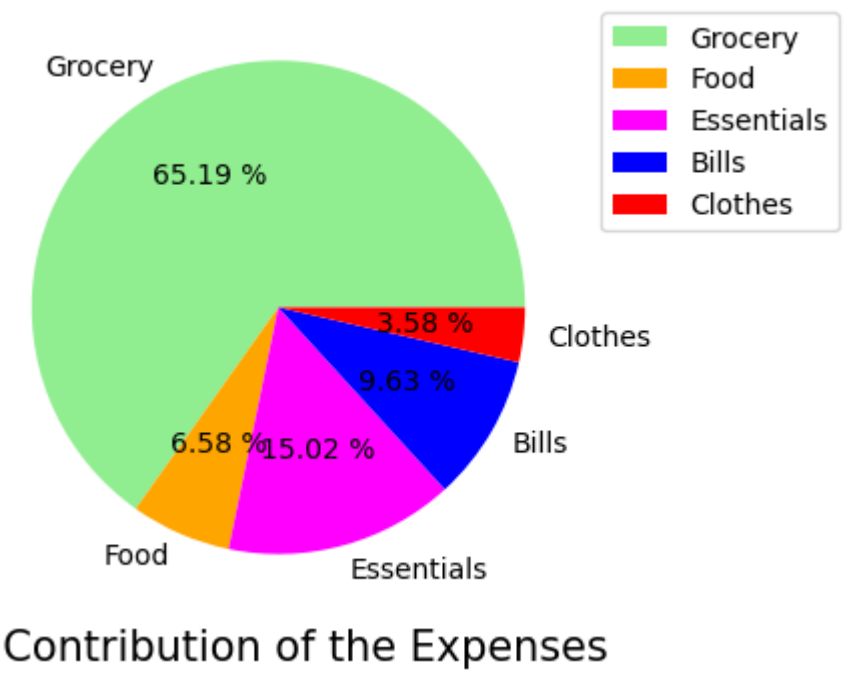
	Date	Category	Sub-Category	Amount	Payment Mode
0	2023-01-01	Grocery	Grocery	30	Cash
1	2023-01-02	Food	Restaurant	890	UPI

In [41]:
```
step_group=data.groupby(["Category"])["Amount"].sum()
step_group
```

Out[41]:
```
Category
Bills         18724
Clothes        1890
Essentials     4315
Food           2765
Grocery        1028
Name: Amount, dtype: int64
```

In [42]:
```
# Plotting the Step Graph with Line Graph to check the sum of Expenses Trend.
plt.figure(figsize=(4,4),dpi=100)
plt.step(data.Category.unique(),step_group,marker="o",where='mid',label="Step Graph")
plt.plot(data.Category.unique(),step_group,marker="o",linestyle='--',label="Line Graph")
plt.xlabel("Categories",size=15,color='black')
plt.ylabel("Amount",size=15,color='black')
plt.title("Expenses Trend Analysis",size=15,color='black')
plt.legend()

plt.show()
# Plotting the Pie chart to check the Percentage Contribution

plt.figure(figsize=(4,4),dpi=100)

plt.pie(step_group,labels=data.Category.unique(),autopct="%.2f %%",colors=["lightgreen","orange","magenta","blue",
plt.xlabel("% Contribution of the Expenses",color="black",size=15)

plt.legend(loc=2, bbox_to_anchor=(1,1))

plt.show()
```





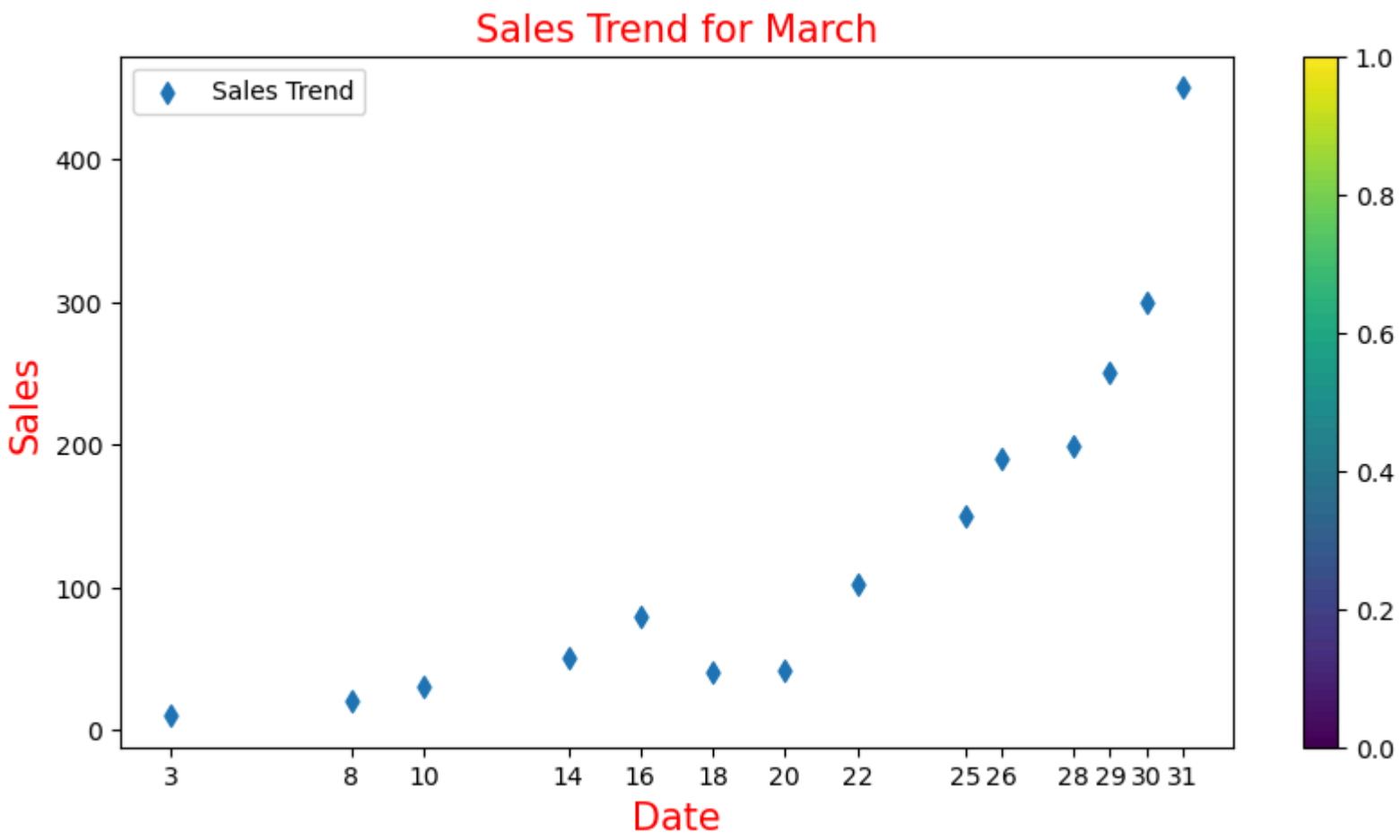
Visual Insights :

1)From the above Step Graph plotted with the Line Graph we can conclude that , the sum of Expenses of Grocery is Higher for a given month.
2)From the Pie Chart we can also conclude that 65.19% of Expenses is utilized on Grocery and 15.02% is utilized by Essentials for a given month.

j) Scatter Plot Graph:

A scatter plot is a mathematical diagram that uses Cartesian coordinates to display values for two variables for a set of data. It is also known as a scatterplot, scatter graph, scatter chart, scattergram, or scatter diagram.

In [43]:
```
# Example : Consider the Highest Sales in a March month from the ordered food.

x=[3,8,10,14,16,18,20,22,25,26,28,29,30,31] # Days on which the Resturant received orders in march month
y=[10,20,30,50,80,40,42,102,150,190,199,250,300,450] #Orders Corresponding to that day

plt.figure(figsize=(10,5),dpi=100) #Size of the Graph
plt.scatter(x,y,label="Sales Trend",marker="d")
plt.title("Sales Trend for March",size=15,color="Red")
plt.xlabel("Date",size=15,color="Red")
plt.ylabel("Sales",size=15,color="Red")
plt.xticks(x)

plt.colorbar() # Displays the Bar of color
plt.legend()
plt.show()
```



Visual Insights :

From the above Scatter Chart we can conclute that,Sales of the given month is suddenly increasing after 20th of March Month

k) Subplots:

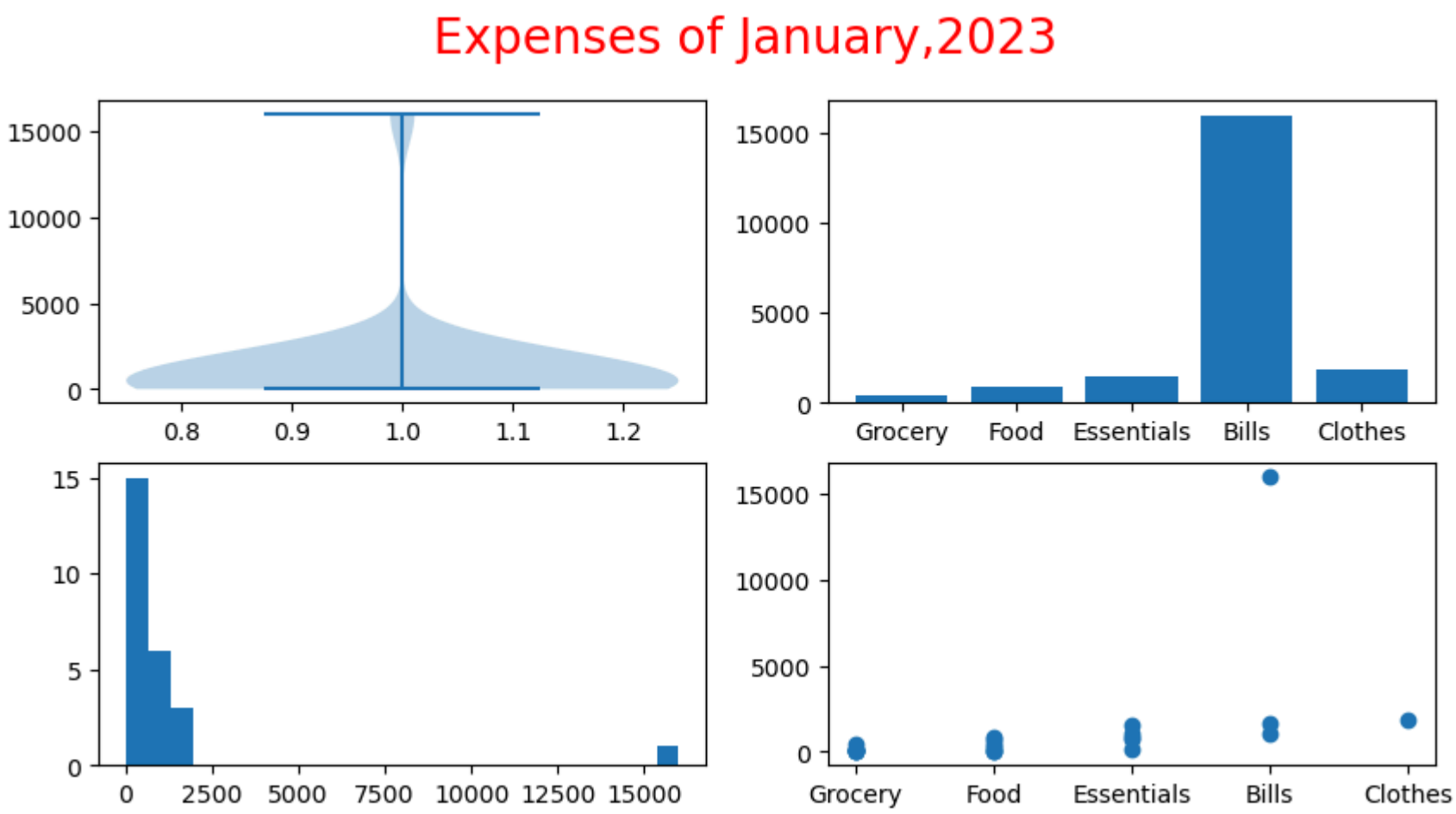
In Python, subplots are a way to create multiple plots on the same figure. This can be useful for comparing different data sets or for creating more complex visualizations.

In [44]:
```
##Dataset:Importing the "Expense" Dataset.
data.head(2)
```

Out[44]:

	Date	Category	Sub-Category	Amount	Payment Mode
0	2023-01-01	Grocery	Grocery	30	Cash
1	2023-01-02	Food	Restaurant	890	UPI

In [45]:
```
# Plotting the Subplots with the expense dataset

#Defining the dimensions of the Graph for all 4 :
plt.figure(figsize=(10,5),dpi=100)


#Plotting the Violin Graph
plt.subplot(2,2,1) # Parameters are rows,columns, chart number
plt.violinplot(data["Amount"])

# Plotting the Common Title to all
plt.suptitle("Expenses of January,2023",size=20,color="red")


# Plotting the Bar Graph
plt.subplot(2,2,2) # Parameters are rows,columns, chart number
plt.bar(data["Category"],data.Amount)

# Plotting the Histogram
plt.subplot(2,2,3) # Parameters are rows,columns, chart number
plt.hist(data.Amount,bins=25)


# Plotting the Step Plot
plt.subplot(2,2,4) # Parameters are rows,columns, chart number
plt.scatter(data["Category"],data.Amount)


# Saving the Graph
plt.savefig("Expenses_2023.png")
plt.show()
```



Visual Insights:

1)Violin chart explains that the Expenses is widely spread under 2000Rs.
2)From the bar Chart we can conclude that bills contribute the highest Expenses.
3)From the Histogram chart , we get to know there is a outlier in the graph.
4)From the scatter chart ,we can conclude that the Outlier is present in the Bills Expenses.