Tip

In [1]:

```
import seaborn as sns
import matplotlib.pyplot as plt
tips=sns.load_dataset("tips")
tips.head()
```

Out[1]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [2]:

```
## Check the Dimension of the Dataset
tips.shape
```

Out[2]:

```
(244, 7)
```

In [4]:

```
## Statistical Analysis
## In statistical analysis, first, we use the df.describe() which will give
tips.describe()
```

Out[4]:

	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

The above table shows the count, mean, standard deviation, min, 25%, 50%, 75%, and max values for each column.

In [5]:
```
tips.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype   
---  ------      --------------  -----   
 0   total_bill  244 non-null    float64 
 1   tip         244 non-null    float64 
 2   sex         244 non-null    category
 3   smoker      244 non-null    category
 4   day         244 non-null    category
 5   time        244 non-null    category
 6   size        244 non-null    int64   
dtypes: category(4), float64(2), int64(1)
memory usage: 7.4 KB
```

In [6]:
```
## Checking for missing values
tips.isnull().sum()
```

Out[6]:
```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

In [7]:
```
tips.columns
```

Out[7]:
```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtyp
e='object')
```

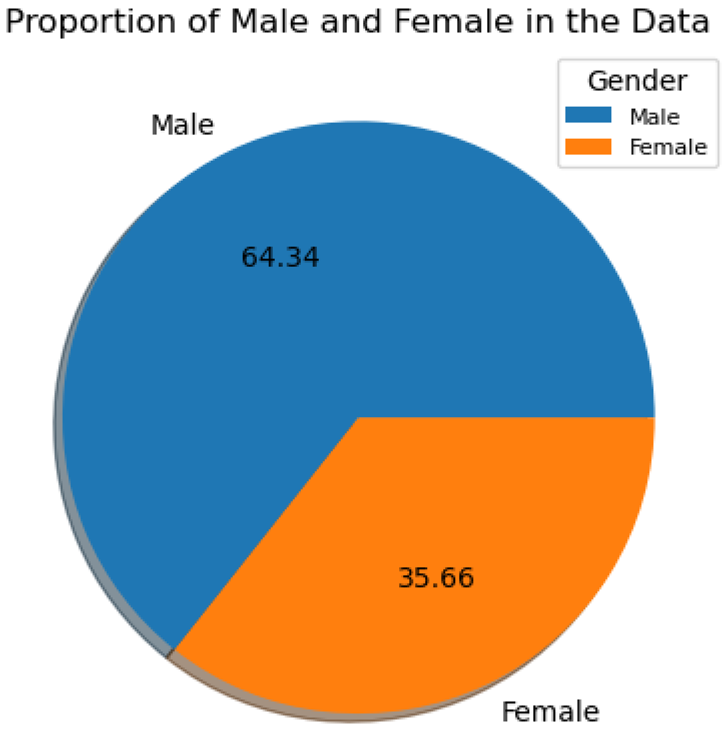
In [8]:
```
tips["sex"].unique()
```

Out[8]:
```
['Female', 'Male']
Categories (2, object): ['Male', 'Female']
```

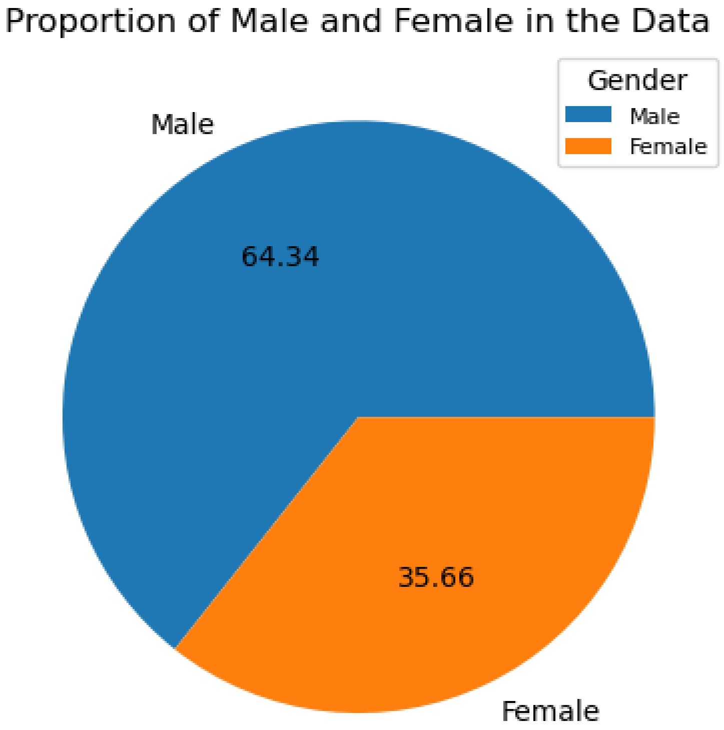
In [9]:
```
tips["sex"].value_counts()
```

Out[9]:
```
sex
Male      157
Female     87
Name: count, dtype: int64
```

In [22]:
```
plt.pie(tips["sex"].value_counts(),labels=["Male","Female"],autopct="%0.02
plt.title("Proportion of Male and Female in the Data")
plt.legend(title="Gender",loc="upper right",fontsize=8)
plt.show()
```



In [27]:
```
plt.pie(tips["sex"].value_counts(),labels=["Male","Female"],autopct="%0.02
plt.title("Proportion of Male and Female in the Data")
plt.legend(title="Gender",loc="upper right",fontsize=8)
plt.show()
```



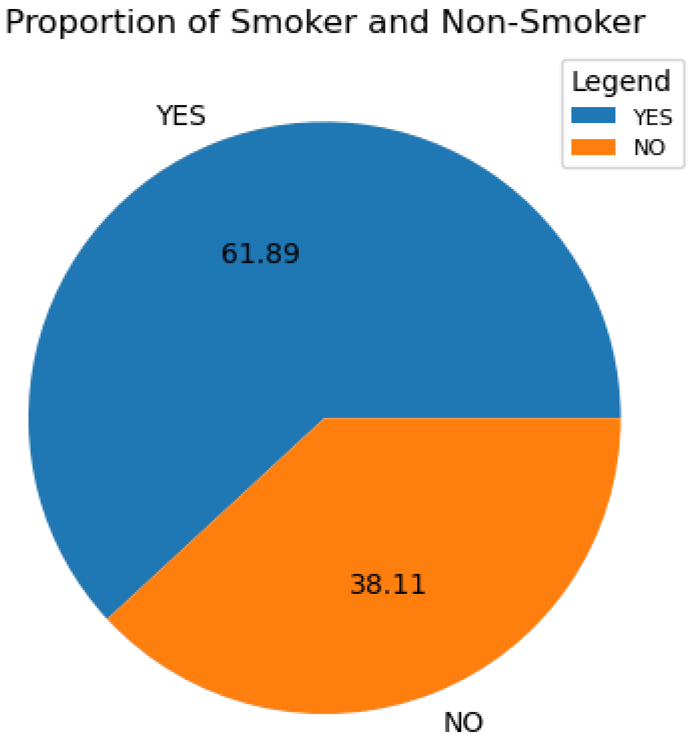
In [23]:
```
tips["smoker"].unique()
```

Out[23]:
```
['No', 'Yes']
Categories (2, object): ['Yes', 'No']
```

In [24]:
```
tips["smoker"].value_counts()
```

Out[24]:
```
smoker
No     151
Yes     93
Name: count, dtype: int64
```

In [26]:
```
plt.pie(tips["smoker"].value_counts(),labels=["YES","NO"],autopct="%0.02f"
plt.title("Proportion of Smoker and Non-Smoker")
plt.legend(title="Legend",loc="upper right",fontsize=8)
plt.show()
```



In [28]:
```
tips["day"].unique()
```

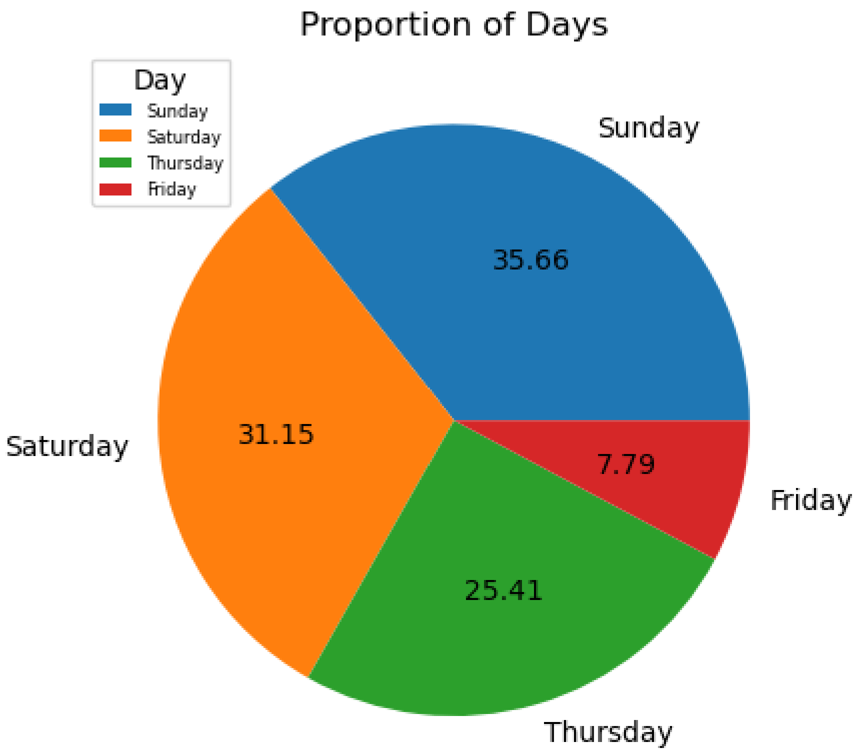
Out[28]:
```
['Sun', 'Sat', 'Thur', 'Fri']
Categories (4, object): ['Thur', 'Fri', 'Sat', 'Sun']
```

In [29]:
```
tips["day"].value_counts()
```

Out[29]:
```
day
Sat     87
Sun     76
Thur    62
Fri     19
Name: count, dtype: int64
```

In [34]:

```
plt.pie(tips["day"].value_counts(),labels=["Sunday","Saturday","Thursday","
plt.title("Proportion of Days")
plt.legend(title="Day",loc="upper left",fontsize=6)
plt.show()
```



In [35]:

```
tips["time"].unique()
```

Out[35]:

```
['Dinner', 'Lunch']
Categories (2, object): ['Lunch', 'Dinner']
```

In [36]:

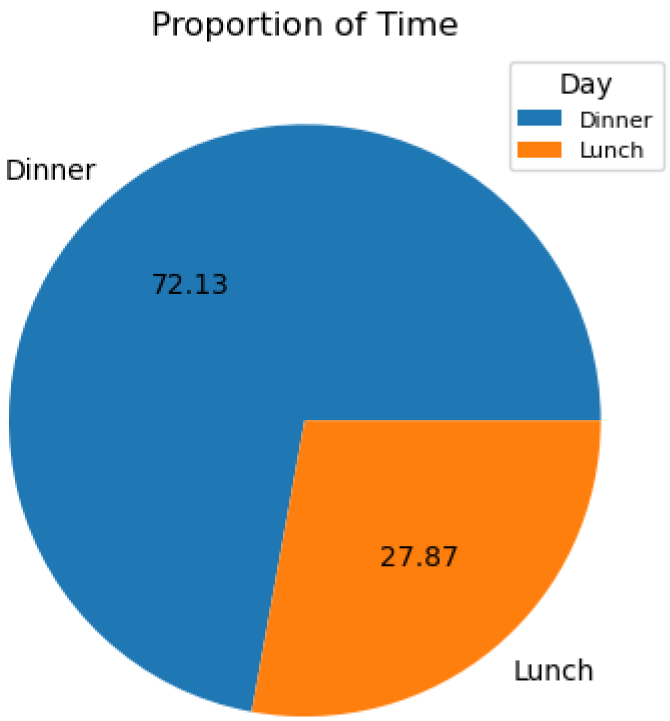
```
tips["time"].value_counts()
```

Out[36]:

```
time
Dinner    176
Lunch      68
Name: count, dtype: int64
```

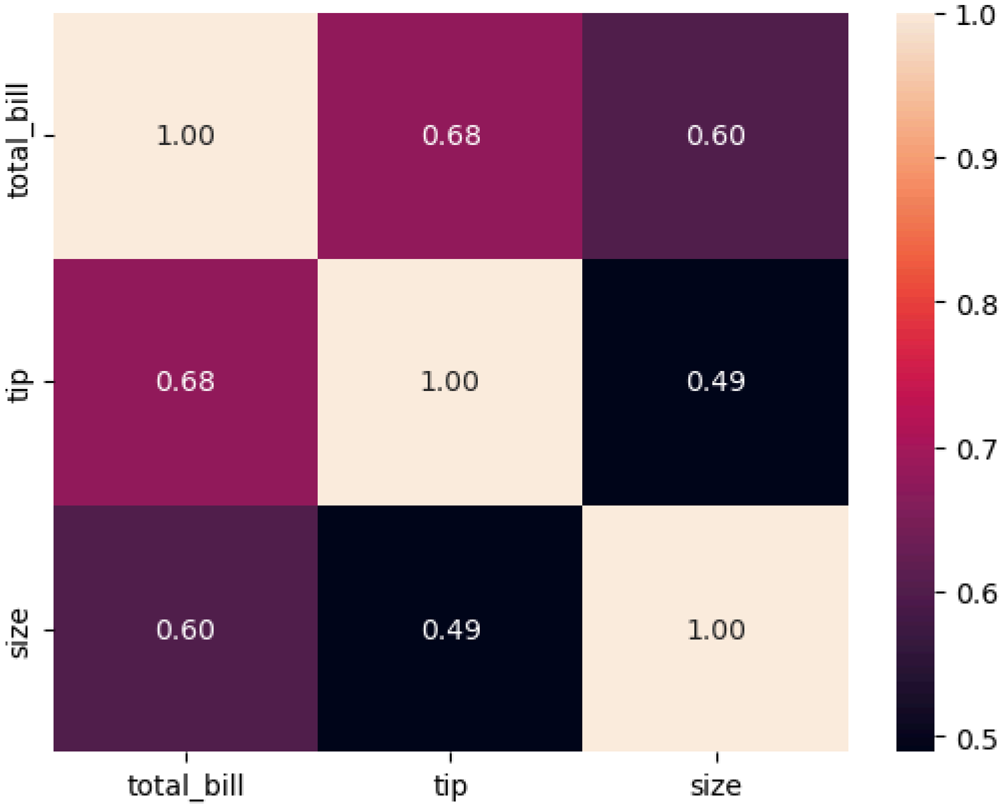
In [37]:

```
plt.pie(tips["time"].value_counts(),labels=["Dinner","Lunch"],autopct="%0.
plt.title("Proportion of Time")
plt.legend(title="Day",loc="upper right",fontsize=8)
plt.show()
```

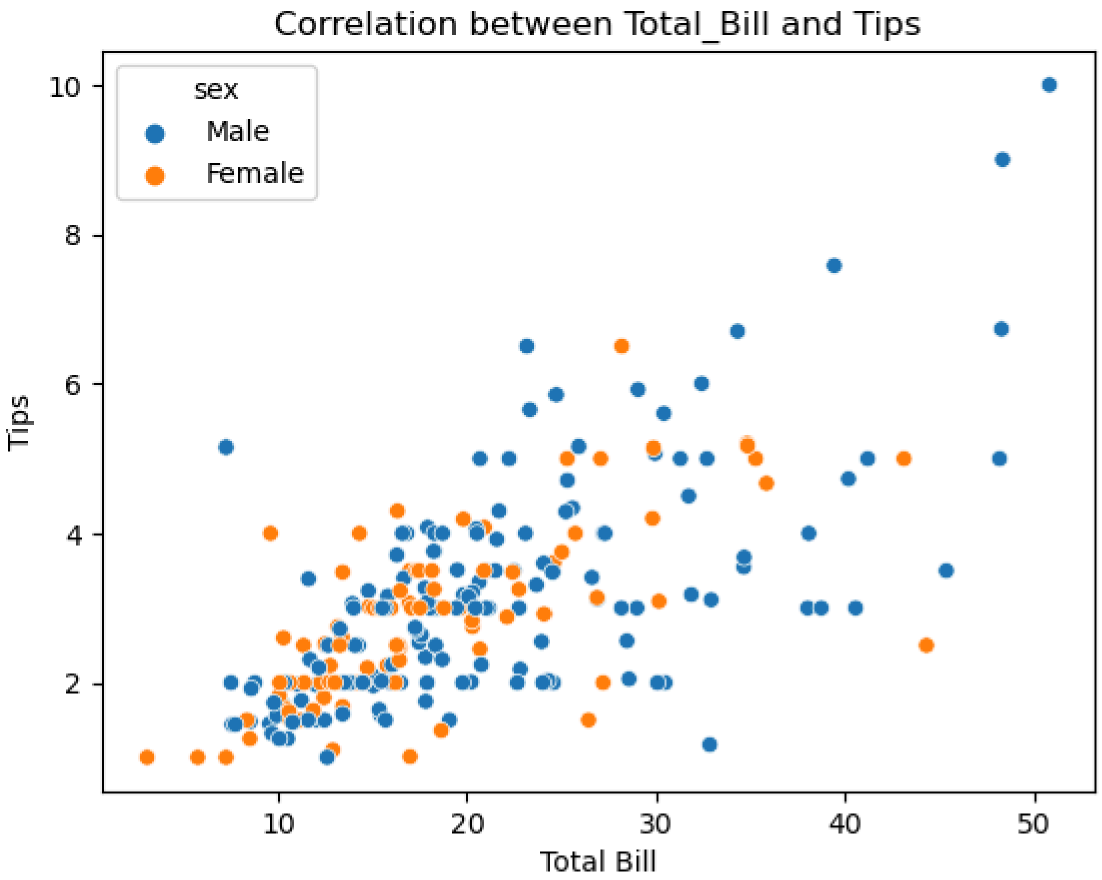


Correlation Matrix

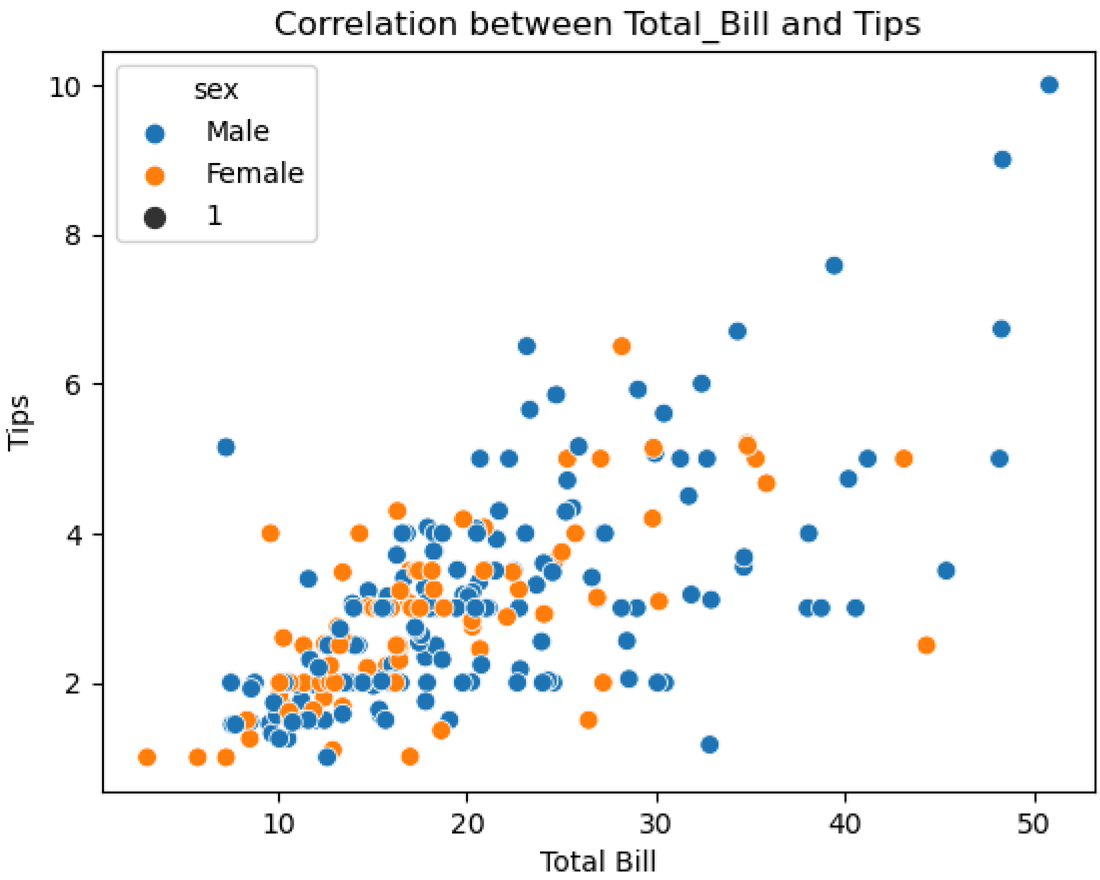
In [41]:
```
df=tips[["total_bill","tip","size"]]

corr=df.corr()
plt.figure(dpi=100)
sns.heatmap(df.corr(),annot=True,fmt="0.2f")
plt.show()
```



```
In [47]: plt.figure(dpi=100)
         sns.scatterplot(x="total_bill",y="tip",hue="sex",sizes=(50,100),data=tips)
         plt.xlabel("Total Bill")
         plt.ylabel("Tips")
         plt.title("Correlation between Total_Bill and Tips")
         plt.show()
```

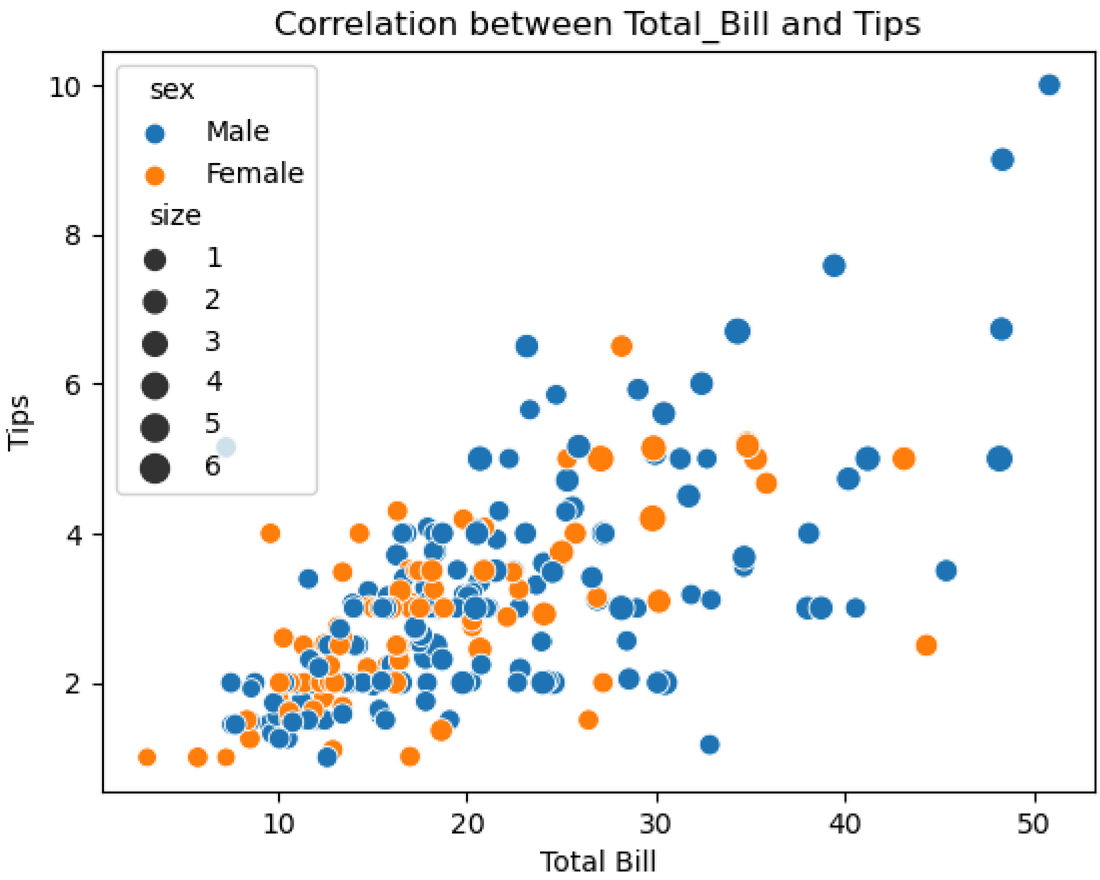


```
In [51]: plt.figure(dpi=100)
         sns.scatterplot(x="total_bill",y="tip",hue="sex",sizes=(50,100),size=1,dat
         plt.xlabel("Total Bill")
         plt.ylabel("Tips")
         plt.title("Correlation between Total_Bill and Tips")
         plt.show()
```



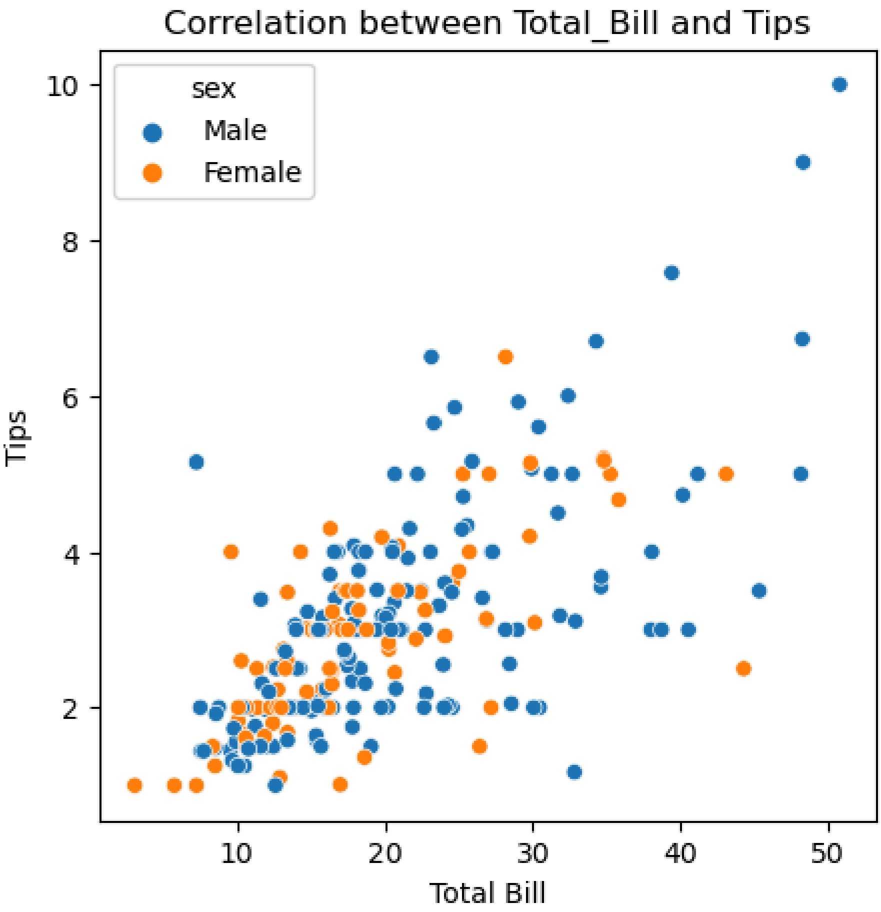
In [53]:

```
plt.figure(dpi=100)
sns.scatterplot(x="total_bill",y="tip",hue="sex",sizes=(50,100),size="size
plt.xlabel("Total Bill")
plt.ylabel("Tips")
plt.title("Correlation between Total_Bill and Tips")
plt.show()
```



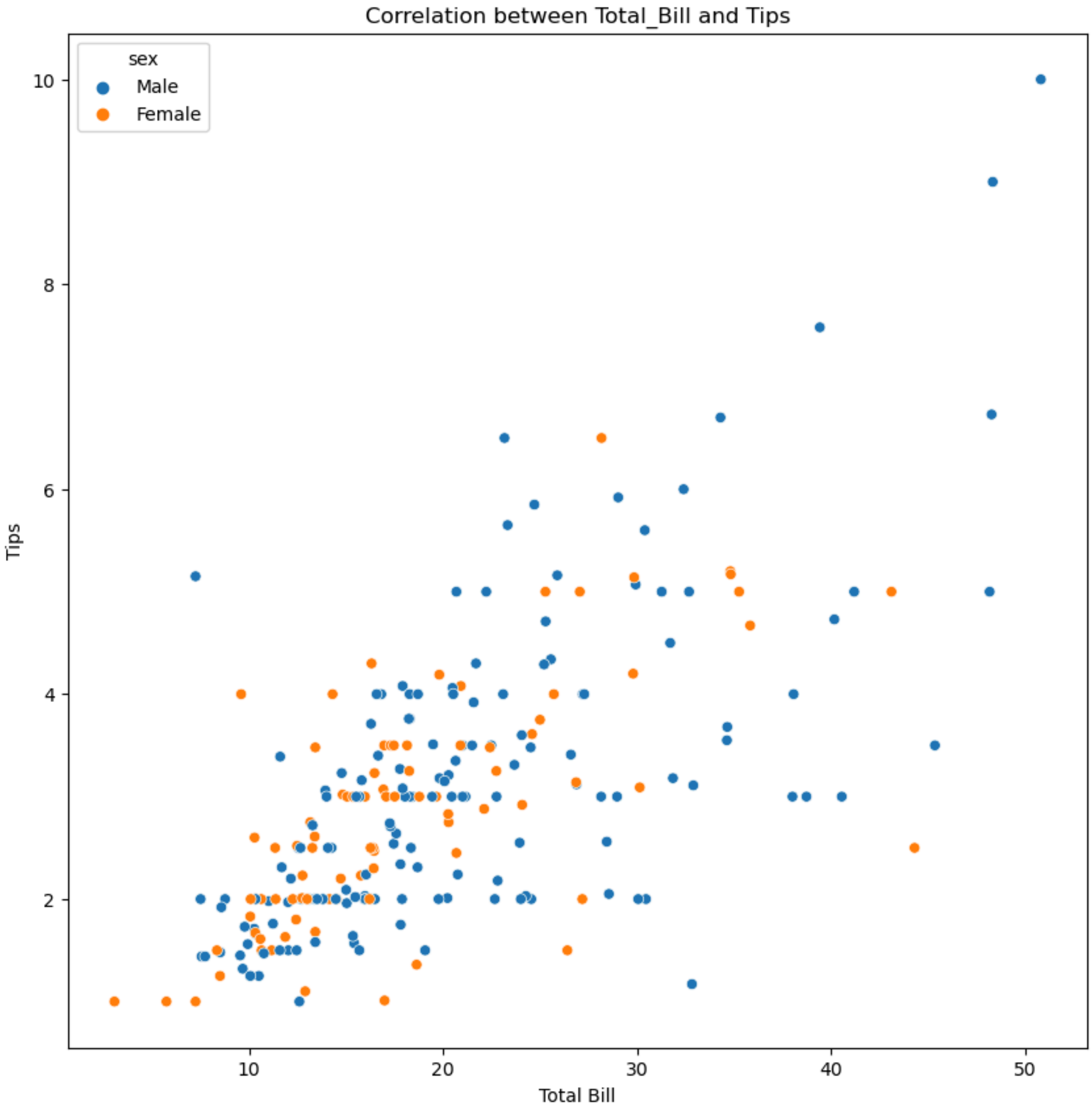
In [56]:

```
plt.figure(figsize=(5,5))
sns.scatterplot(x="total_bill",y="tip",hue="sex",sizes=(50,100),data=tips)
plt.xlabel("Total Bill")
plt.ylabel("Tips")
plt.title("Correlation between Total_Bill and Tips")
plt.show()
```



In [58]:

```
plt.figure(figsize=(10,10))
sns.scatterplot(x="total_bill",y="tip",hue="sex",sizes=(50,100),data=tips)
plt.xlabel("Total Bill")
plt.ylabel("Tips")
plt.title("Correlation between Total_Bill and Tips")
plt.show()
```



In [ ]: