In [114]:

```
import seaborn as sns # data visualization
import pandas as pd # functions and reading the files
import numpy as np
```

In [115]:

```
S = pd.read_csv(r"C:\Users\Vishal\Downloads\Serve.csv")
```

In [116]:

```
S.head() ##reading the data columns and rows
```

Out[116]:

| | Amount | Tip | Gender | Smoker | Day | Time | Partysize |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2.0 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3.0 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3.0 |
| **3** | 23.68 | 3.31 | Mal | No | Sun | Dinner | 2.0 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4.0 |

In [ ]:

```
"""
Finding the days apart from weekends
"""
```

In [117]:

```
S.Day.unique()
```

Out[117]:

```
array(['Sun', 'Sat', 'Thur', 'Fri', 'Friday', 'Thurs', 'San', nan, 'S'],
      dtype=object)
```

In [ ]:

```
"""
Eating time of people
"""
```

In [118]:

```
S.Time.unique()
```

Out[118]:

```
array(['Dinner', 'Lunch', 'LD', 'L', 'D'], dtype=object)
```

In [ ]:

```
"""
Finding the last row of the data
"""
```

In [119]:

```
S.tail()
```

Out[119]:

| | Amount | Tip | Gender | Smoker | Day | Time | Partysize |
|---|---|---|---|---|---|---|---|
| **322** | 19.09 | 3.00 | Female | No | Thur | Dinner | 3.0 |
| **323** | 23.90 | 3.98 | Male | Yes | Thur | Dinner | 3.0 |
| **324** | 25.89 | 3.89 | Female | No | NaN | Lunch | 3.0 |
| **325** | 20.09 | 2.96 | Female | No | Sun | Dinner | 4.0 |
| **326** | 14.56 | 2.78 | Female | Yes | Sat | Lunch | 2.0 |

In [ ]:

```
"""
Count of lunches and dinners served
"""
```

In [120]:

```
S.Time.value_counts()
```

Out[120]:

```
Dinner    212
Lunch     110
L           3
LD          1
D           1
Name: Time, dtype: int64
```

In [ ]:

```
"""
Data cleaning
"""
```

In [122]:

```
S["Time"] = S["Time"].replace(['l'],'Lunch') #replacing the value of l with lunch
```

In [123]:

```
S["Time"] = S["Time"].replace(['D'],'Dinner') #replacing the D value with dinner
```

In [124]:

```
S.Time.value_counts()
```

Out[124]:

```
Dinner    213
Lunch     110
L           3
LD          1
Name: Time, dtype: int64
```

In [ ]:

```
"""
Counting the nan values
"""
```

In [125]:

```
S['Day'].isna().sum()
```

Out[125]:

```
2
```

In [ ]:

```
"""
dropping all the nan values

"""
```

In [126]:

```
S2 = S.dropna()
S2
```

Out[126]:

| | Amount | Tip | Gender | Smoker | Day | Time | Partysize |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2.0 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3.0 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3.0 |
| **3** | 23.68 | 3.31 | Mal | No | Sun | Dinner | 2.0 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **321** | 19.76 | 2.21 | Male | No | Sat | Dinner | 3.0 |
| **322** | 19.09 | 3.00 | Female | No | Thur | Dinner | 3.0 |
| **323** | 23.90 | 3.98 | Male | Yes | Thur | Dinner | 3.0 |
| **325** | 20.09 | 2.96 | Female | No | Sun | Dinner | 4.0 |
| **326** | 14.56 | 2.78 | Female | Yes | Sat | Lunch | 2.0 |

317 rows × 7 columns

In [127]:

```
S2.Day.value_counts()
```

Out[127]:

```
Sat       106
Sun        91
Thur       77
Fri        34
Thurs       4
S           3
Friday      1
San         1
Name: Day, dtype: int64
```

In [ ]:

```
"""
replacing the values and making it equal
"""
```

In [128]:

```
S2["Day"] = S2["Day"].replace(['Thurs'],'Th')
S2["Day"] = S2["Day"].replace(['S'],'Sun')
S2["Day"] = S2["Day"].replace(['Friday'],'Fri')
S2["Day"] = S2["Day"].replace(['San'],'Sun')
```

```
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\83213700.py:1: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Day"] = S2["Day"].replace(['Thurs'],'Th')
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\83213700.py:2: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Day"] = S2["Day"].replace(['S'],'Sun')
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\83213700.py:3: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Day"] = S2["Day"].replace(['Friday'],'Fri')
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\83213700.py:4: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Day"] = S2["Day"].replace(['San'],'Sun')
```

In [129]:

```
S2.Day.value_counts()
```

Out[129]:

```
Sat     106
Sun      95
Thur     77
Fri      35
Th        4
Name: Day, dtype: int64
```

In [130]:

```
S2["Time"] = S2["Time"].replace(['LD'],'Lunch')
```

```
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\1637844370.py:1: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Time"] = S2["Time"].replace(['LD'],'Lunch')
```

In [131]:

```
S2.Amount.value_counts()
```

Out[131]:

```
19.65    4
10.07    3
8.88     3
13.42    3
20.69    2
        ..
10.65    1
12.43    1
24.08    1
11.69    1
20.09    1
Name: Amount, Length: 286, dtype: int64
```

In [132]:

```
S2.applymap(np.isreal)
```

Out[132]:

| | Amount | Tip | Gender | Smoker | Day | Time | Partysize |
|---|---|---|---|---|---|---|---|
| **0** | True | True | False | False | False | False | True |
| **1** | True | True | False | False | False | False | True |
| **2** | True | True | False | False | False | False | True |
| **3** | True | True | False | False | False | False | True |
| **4** | True | True | False | False | False | False | True |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **321** | True | True | False | False | False | False | True |
| **322** | True | True | False | False | False | False | True |
| **323** | True | True | False | False | False | False | True |
| **325** | True | True | False | False | False | False | True |
| **326** | True | True | False | False | False | False | True |

317 rows × 7 columns

In [133]:

```
S2[~S2.applymap(np.isreal).all(1)]
```

Out[133]:

| | Amount | Tip | Gender | Smoker | Day | Time | Partysize |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2.0 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3.0 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3.0 |
| **3** | 23.68 | 3.31 | Mal | No | Sun | Dinner | 2.0 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **321** | 19.76 | 2.21 | Male | No | Sat | Dinner | 3.0 |
| **322** | 19.09 | 3.00 | Female | No | Thur | Dinner | 3.0 |
| **323** | 23.90 | 3.98 | Male | Yes | Thur | Dinner | 3.0 |
| **325** | 20.09 | 2.96 | Female | No | Sun | Dinner | 4.0 |
| **326** | 14.56 | 2.78 | Female | Yes | Sat | Lunch | 2.0 |

317 rows × 7 columns

In [134]:

```
S2.Time.value_counts()
```

Out[134]:

```
Dinner    206
Lunch     108
L           3
Name: Time, dtype: int64
```

In [ ]:

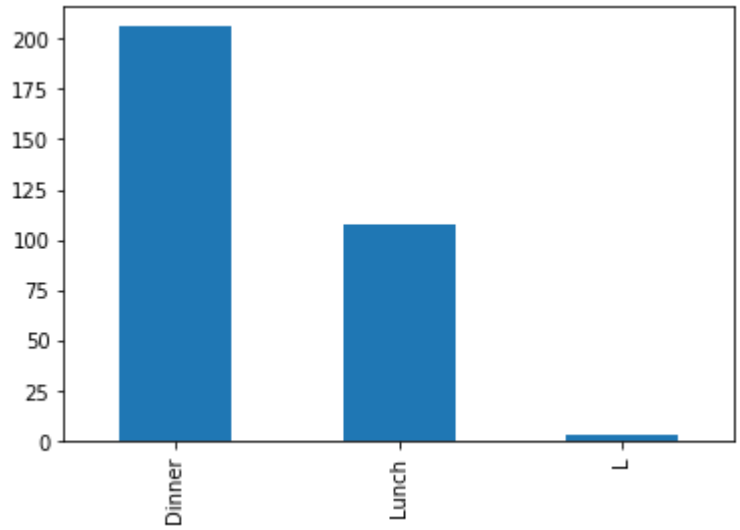
```
"""
People pay on dinner as compared to that of lunch, as the value of dinner is more than lunc
       """
```

In [135]:

```
S2.Time.value_counts().plot(kind = "bar")
```

Out[135]:

```
<AxesSubplot: >
```




In [136]:

```
"""
Contribution
Dinner = 65%
Lunch = 35%"""
```

Out[136]:

```
'\nContribution\nDinner = 65%\nLunch = 35%'
```

In [137]:

```
S2.Time.value_counts(normalize = True)
```

Out[137]:

```
Dinner    0.649842
Lunch     0.340694
L         0.009464
Name: Time, dtype: float64
```

In [138]:

```
"""
Finding tip
"""
```

Out[138]:

```
'\nFinding tip\n'
```

In [139]:

```
S2.groupby(["Time"])["Tip"].mean()
```

Out[139]:

```
Time
Dinner    3.077233
L         2.450000
Lunch     2.701296
Name: Tip, dtype: float64
```

In [140]:

```
"""Plotting"""
```

Out[140]:

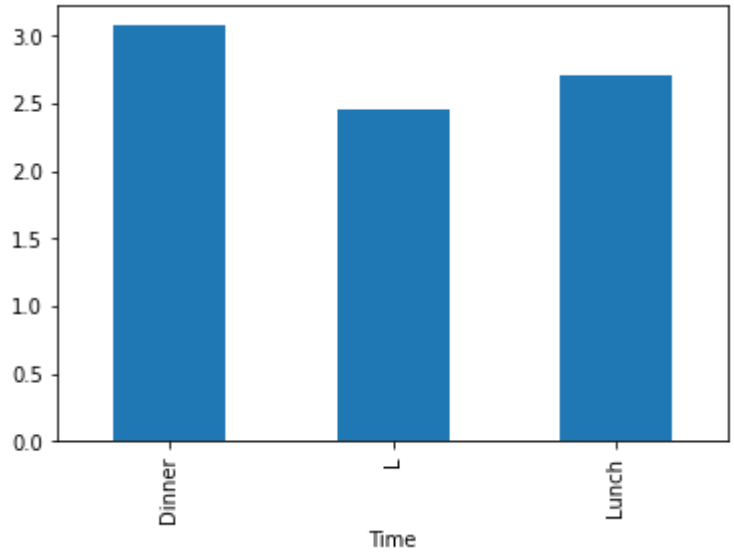
```
'Plotting'
```

In [141]:

```
S2.groupby(["Time"])["Tip"].mean().plot.bar()
```

Out[141]:

```
<AxesSubplot: xlabel='Time'>
```




In [142]:

```
S2.Smoker.value_counts()
```

Out[142]:

```
No     181
Yes    129
N        5
Y        2
Name: Smoker, dtype: int64
```

In [143]:

```
S2["Smoker"] = S2["Smoker"].replace(['N'],'No')
S2["Smoker"] = S2["Smoker"].replace(['Y'],'Yes')
```

```
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\966997748.py:1: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Smoker"] = S2["Smoker"].replace(['N'],'No')
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\966997748.py:2: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Smoker"] = S2["Smoker"].replace(['Y'],'Yes')
```

In [144]:

```
S2.Smoker.value_counts()
```

Out[144]:

```
No     186
Yes    131
Name: Smoker, dtype: int64
```

In [ ]:

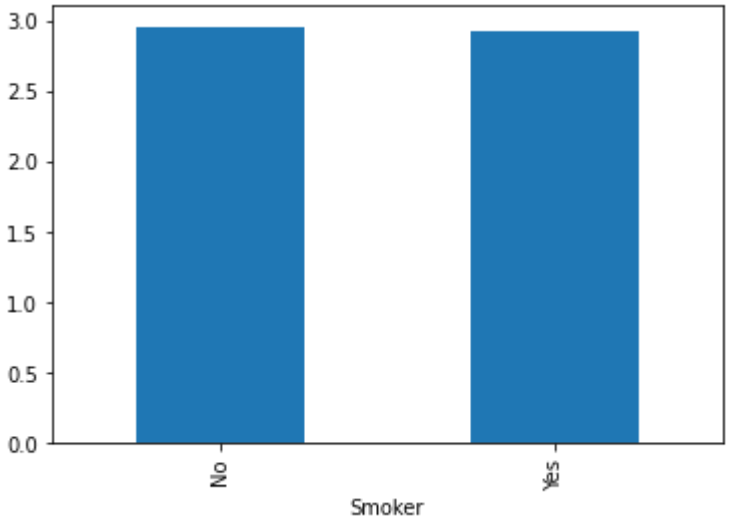
```
"""
Checking tips rate of somkers Vs Non Smokers
"""
```

In [145]:

```
S2.groupby(["Smoker"])["Tip"].mean().plot.bar()
```

Out[145]:

```
<AxesSubplot: xlabel='Smoker'>
```




In [ ]:

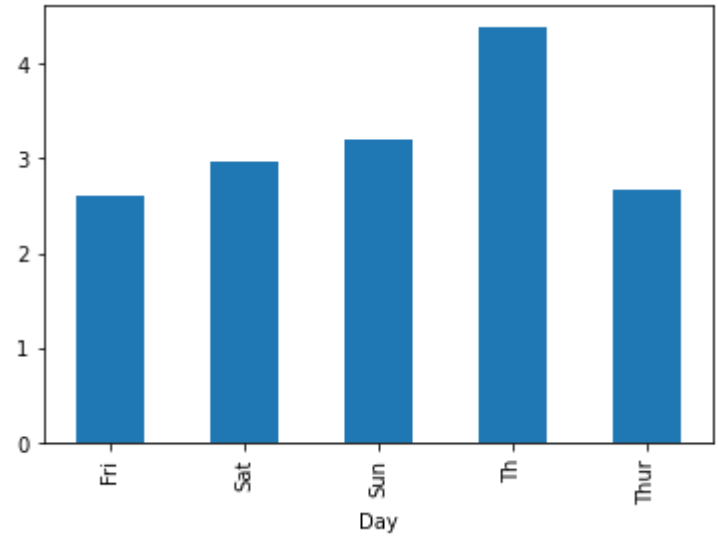
```
"""Maximum tip day"""
```

In [146]:

```
S2.groupby(["Day"])["Tip"].mean().plot.bar()
```

Out[146]:

```
<AxesSubplot: xlabel='Day'>
```




In [147]:

```
S2.Gender.value_counts()
```

Out[147]:

```
Male      187
Female    126
M           2
Mal         1
F           1
Name: Gender, dtype: int64
```

In [148]:

```
S2["Gender"] = S2["Gender"].replace(['M'],'Male')
S2["Gender"] = S2["Gender"].replace(['Mal'],'Male')
S2["Gender"] = S2["Gender"].replace(['F'],'Female')
```

```
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\2345461062.py:1: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Gender"] = S2["Gender"].replace(['M'],'Male')
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\2345461062.py:2: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Gender"] = S2["Gender"].replace(['Mal'],'Male')
C:\Users\Vishal\AppData\Local\Temp\ipykernel_22540\2345461062.py:3: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  S2["Gender"] = S2["Gender"].replace(['F'],'Female')
```

In [149]:

```
S2.Gender.value_counts()
```

Out[149]:

```
Male      190
Female    127
Name: Gender, dtype: int64
```

In [ ]:

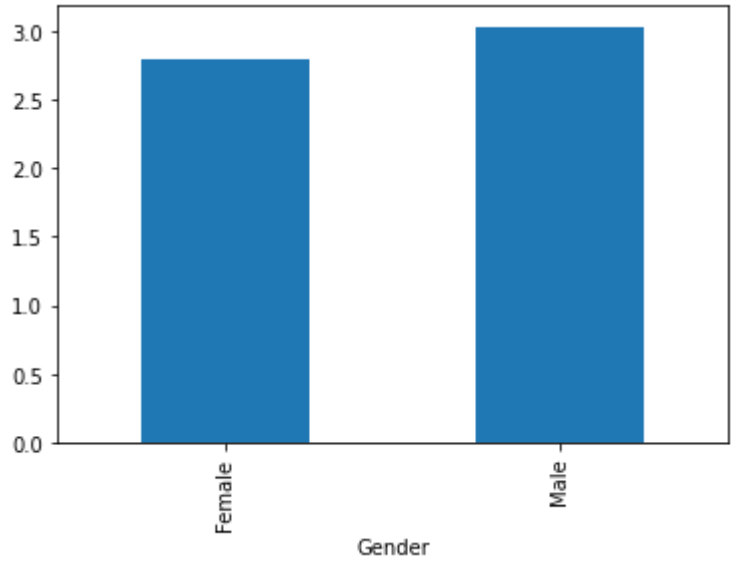
```
"""Gender giving more tip
"""
```

In [150]:

```
S2.groupby(["Gender"])["Tip"].mean().plot.bar()
```

Out[150]:

```
<AxesSubplot: xlabel='Gender'>
```




In [ ]:

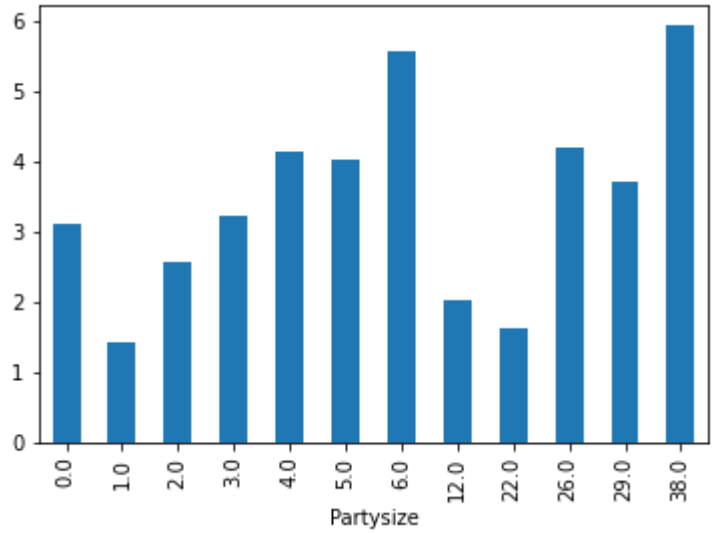
```
"""Party paying more tip"""
```

In [151]:

```
S2.groupby(["Partysize"])["Tip"].mean().plot.bar()
```

Out[151]:

```
<AxesSubplot: xlabel='Partysize'>
```




In [ ]:

```
"""Gender wise distribution on tip on the basis of smoker and non smoker"""
```

In [152]:

```
S2.groupby(['Smoker','Gender'])['Tip'].mean()
```

Out[152]:

```
Smoker  Gender
No      Female    2.800127
        Male      3.069626
Yes     Female    2.803333
        Male      2.997349
Name: Tip, dtype: float64
```

In [153]:

```
S2.groupby(['Smoker','Gender'])['Tip'].mean().unstack()
```

Out[153]:

| Gender | Female | Male |
|---|---|---|
| **Smoker** | | |
| **No** | 2.800127 | 3.069626 |
| **Yes** | 2.803333 | 2.997349 |

In [ ]:

```
"""Which day do smokers and non-smokers tip more generously on average?"""
```

In [154]:

```
S2.groupby(['Smoker','Day'])['Tip'].mean().unstack()
```

Out[154]:

| Day | Fri | Sat | Sun | Th | Thur |
|---|---|---|---|---|---|
| **Smoker** | | | | | |
| **No** | 2.720667 | 3.017358 | 3.216308 | 4.350 | 2.571961 |
| **Yes** | 2.527000 | 2.932075 | 3.144000 | 4.425 | 2.855000 |

In [ ]:

```

```

In [155]:

```
S2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 317 entries, 0 to 326
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Amount     317 non-null    float64
 1   Tip        317 non-null    float64
 2   Gender     317 non-null    object 
 3   Smoker     317 non-null    object 
 4   Day        317 non-null    object 
 5   Time       317 non-null    object 
 6   Partysize  317 non-null    float64
dtypes: float64(3), object(4)
memory usage: 19.8+ KB
```

In [156]:

```
S2 = S2.astype({'Amount':'float'})
print(S2.dtypes)
```

```
Amount       float64
Tip          float64
Gender        object
Smoker        object
Day           object
Time          object
Partysize    float64
dtype: object
```

In [ ]:

```
"""Make a new column to track the percentage of tips compared to the total bill."""
```

In [157]:

```
S2['tip_pct'] = S2.Tip / S2.Amount
```

In [158]:

```
S2.head()
```

Out[158]:

| | Amount | Tip | Gender | Smoker | Day | Time | Partysize | tip_pct |
|---|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2.0 | 0.059447 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3.0 | 0.160542 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3.0 | 0.166587 |
| **3** | 23.68 | 3.31 | Male | No | Sun | Dinner | 2.0 | 0.139780 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4.0 | 0.146808 |

In [ ]:

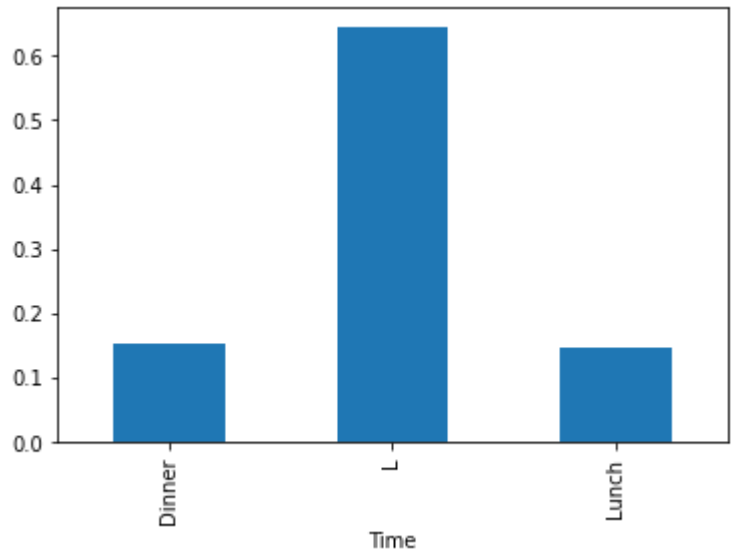
```
"""When is the tip percentage the highest?"""
```

In [159]:

```
S2.groupby(["Time"])["tip_pct"].mean().plot.bar()
```

Out[159]:

```
<AxesSubplot: xlabel='Time'>
```




In [ ]:

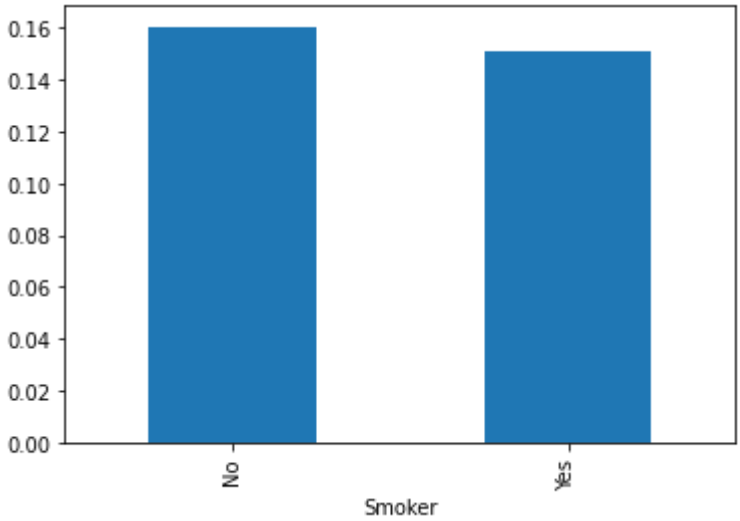
```
"""When is the tip percentage the highest?"""
```

In [160]:

```
S2.groupby(["Smoker"])["tip_pct"].mean().plot.bar()
```

Out[160]:

```
<AxesSubplot: xlabel='Smoker'>
```




In [ ]:

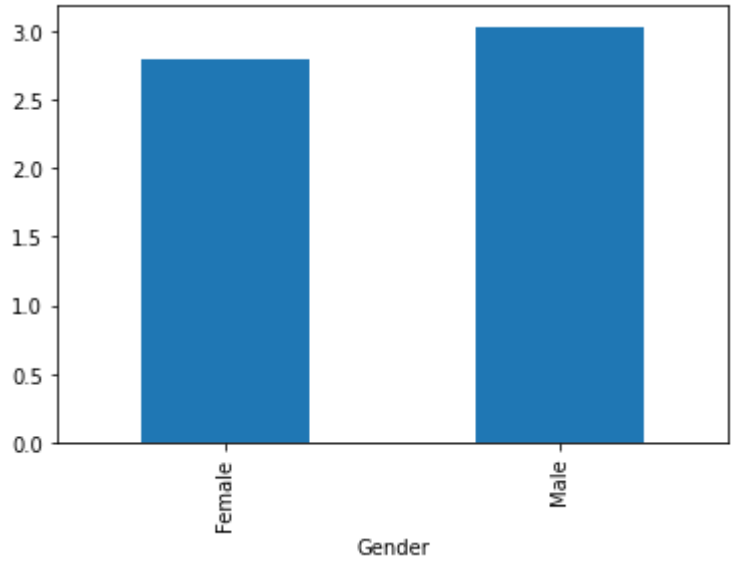
```
"""Which gender's percentage of tips is higher?"""
```

In [161]:

```
S2.groupby(["Gender"])["Tip"].mean().plot.bar()
```

Out[161]:

```
<AxesSubplot: xlabel='Gender'>
```




In [ ]:

```
"""Let's illustrate the connection between the cost and the tips."""
```

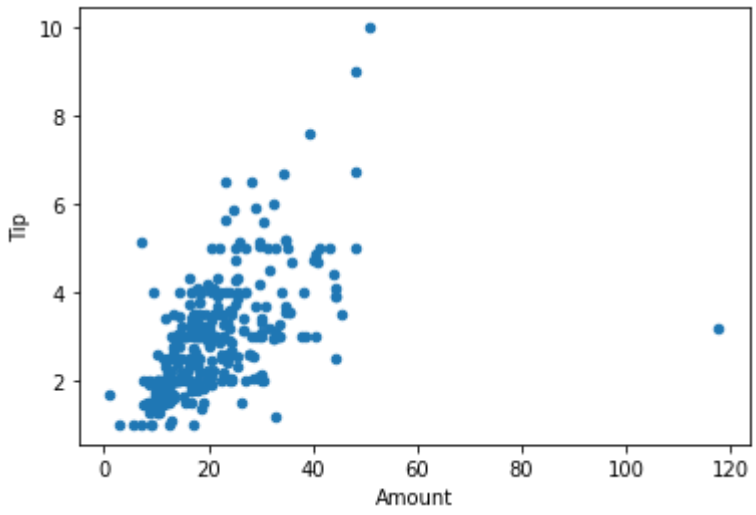
In [162]:

```
S2.plot.scatter(x = "Amount", y = 'Tip')
```

```
C:\Users\Vishal\AppData\Local\Programs\Python\Python310\lib\site-packages\pa
ndas\plotting\_matplotlib\core.py:1114: UserWarning: No data for colormappin
g provided via 'c'. Parameters 'cmap' will be ignored
  scatter = ax.scatter(
```

Out[162]:

```
<AxesSubplot: xlabel='Amount', ylabel='Tip'>
```



In [ ]:

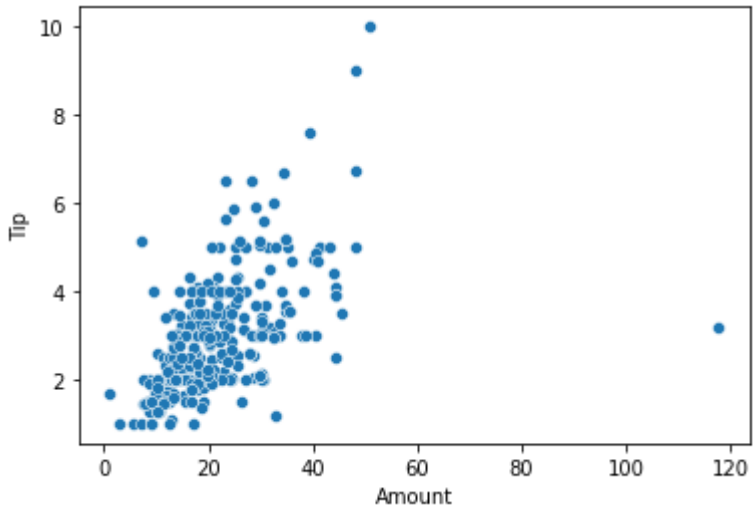
```
"""According to this graph, the tip likewise rises as the overall bill does.
There are a few outliers in the graph above as well.
a better view of this graph using Seaborn"""
```

In [163]:

```
sns.scatterplot(x = "Amount" , y = 'Tip', data = S2)
```

Out[163]:

```
<AxesSubplot: xlabel='Amount', ylabel='Tip'>
```




In [ ]:

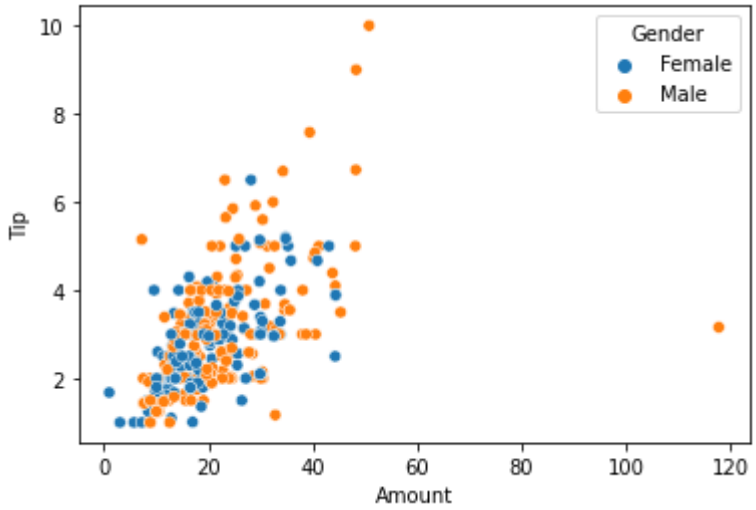
```
"""adding the impact of column gender to the data to better understand it"""
```

In [164]:

```
sns.scatterplot(x = "Amount" , y = 'Tip', data = S2, hue = 'Gender')
```

Out[164]:

```
<AxesSubplot: xlabel='Amount', ylabel='Tip'>
```



In [ ]:

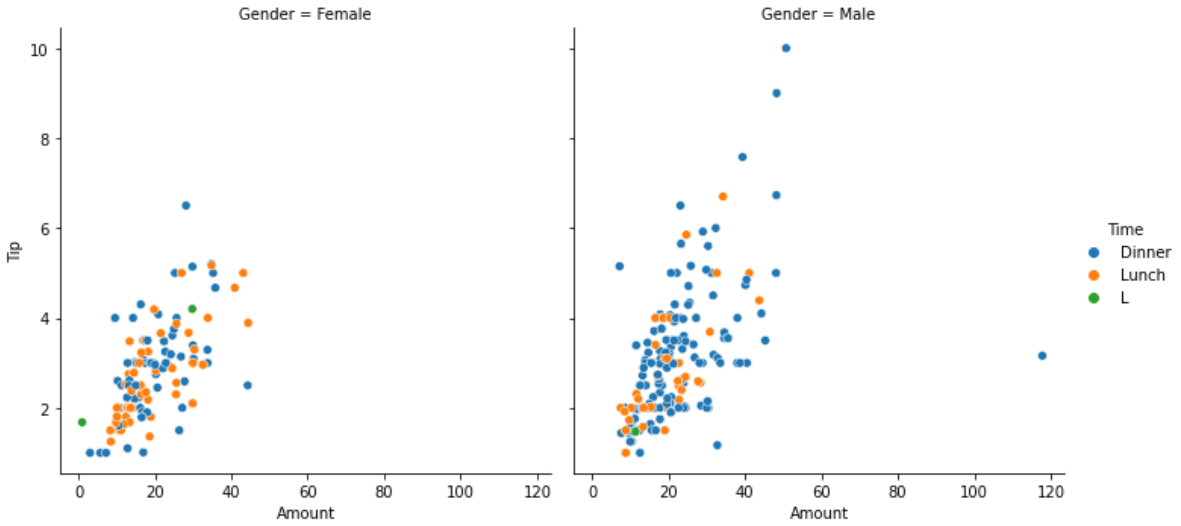
```
"""dividing the time on the graph based on the gender column"""
```

In [165]:

```
sns.relplot(x = 'Amount', y = 'Tip', data = S2, col = 'Gender', hue = 'Time')
```

Out[165]:

```
<seaborn.axisgrid.FacetGrid at 0x2676fe3dc30>
```




In [ ]:

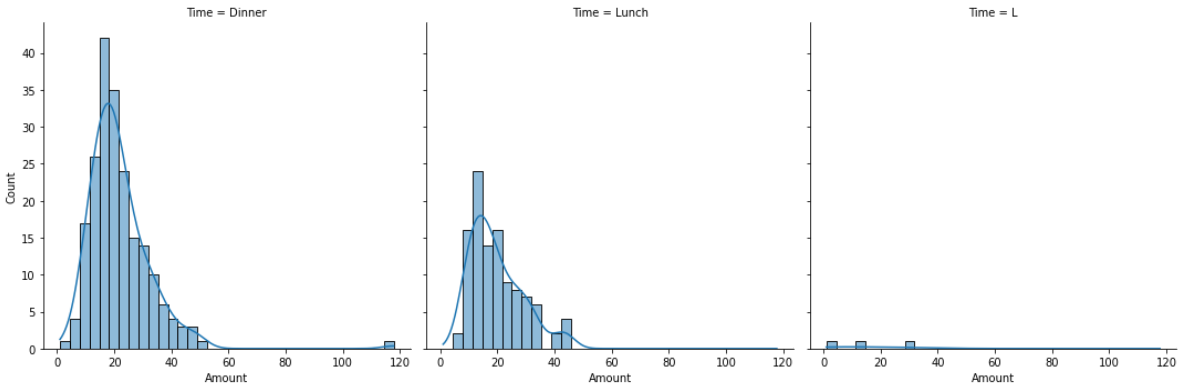
```
"""Let's examine the amount by time distribution.
Distribution plot"""
```

In [167]:

```
sns.displot(data = S2, x = 'Amount' , col = 'Time', kde = True)
```

Out[167]:

```
<seaborn.axisgrid.FacetGrid at 0x26770f5ca90>
```



In [168]:

```
"""Evidently, more customers pay at supper than at lunch; perhaps the restaurant could conc
```

Out[168]:

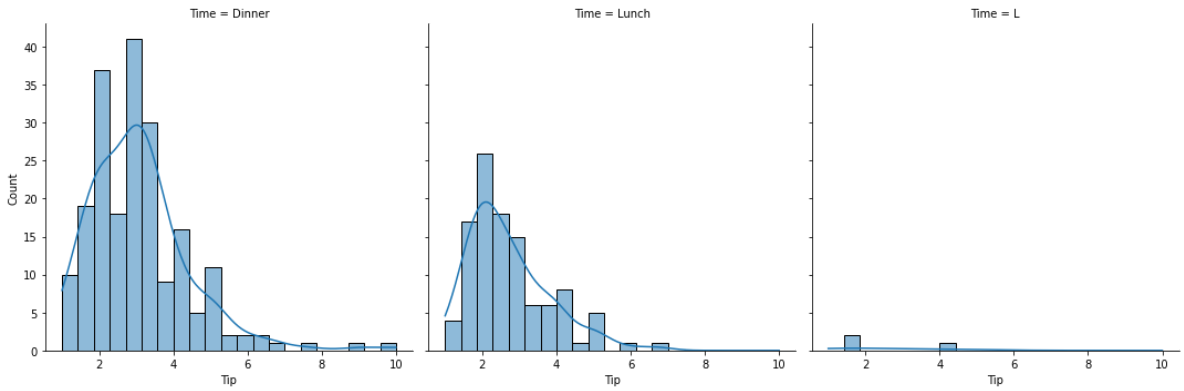
```
'Evidently, more customers pay at supper than at lunch; perhaps the restaura
nt could concentrate more on lunch?'
```

In [169]:

```
sns.displot(data = S2, x = 'Tip' , col = 'Time', kde = True) #distribution plot
```

Out[169]:

```
<seaborn.axisgrid.FacetGrid at 0x267738ba050>
```




In [ ]:

```
"""The graph is left tailed"""
```

In [ ]:

```
"""generating a distribution and connection for all numerical values according to gender"""
```

In [171]:

```
sns.pairplot(data = S2, hue = 'Gender')
```

Out[171]:

```
<seaborn.axisgrid.PairGrid at 0x26773d4dfc0>
```