Data Visualization using seaborn

a. Frequency Distribution - Categorical Variables

- countplot
- catplot

b. Distribution of the Numerical Variable**

- distplot(histogram)
- kdeplot
- boxplot
- violinplot

c. Relationship between 2 Numerical Variables

- lineplot
- scatterplot
- relplot
- lmplot
- heatmap
- pairplot
- facetgrid

d. Relationship between Numerical and Categorical Variables

- pointplot
- barplot
- boxplot
- violinplot
- swarmplot
- catplot
  - facetgrid

Importing required libraries

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
%matplotlib inline
```

```
C:\Users\Ramya\Anaconda3\lib\site-packages\statsmodels\tools\_testing.py:19: FutureWarning: pandas.ut
il.testing is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

Loading dataset

In [2]:

```
tips=sns.load_dataset('tips')
df=pd.DataFrame(tips)
df.head()
```

Out[2]:

	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

Preprocessing and Exploratory Data Analysis

checking for missing data

In [3]:

```
df.isnull().sum()
```

```
Out[3]: total_bill    0
        tip           0
        sex           0
        smoker        0
        day           0
        time          0
        size          0
        dtype: int64
```

Viewing the descriptive statistics of the dataset

```
In [4]: df.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

Get a numerical summary for 'tip'

```
In [5]: df.tip.describe()
```

```
Out[5]: count    244.000000
        mean       2.998279
        std        1.383638
        min        1.000000
        25%        2.000000
        50%        2.900000
        75%        3.562500
        max       10.000000
        Name: tip, dtype: float64
```

Five Number Summary For bill and tip

```
In [6]: bill = df.total_bill

        print("Maximum Bill = ",np.max(bill))
        print("Minimum Bill = ",np.min(bill))
        print("Standard Deviation = ",np.std(bill))
        print("Median =",np.median(bill))
        print("Mean = ",np.mean(bill))
```

```
Maximum Bill =  50.81
Minimum Bill =  3.07
Standard Deviation =  8.88415057777113
Median = 17.795
Mean =  19.785942622950824
```

```
In [7]: tip = df.tip
        print("Maximum Tip = ",np.max(tip))
        print("Minimum Tip = ",np.min(tip))
        print("Standard Deviation = ",np.std(tip))
        print("Median =",np.median(tip))
        print("Mean = ",np.mean(tip))
```

```
Maximum Tip =  10.0
Minimum Tip =  1.0
Standard Deviation =  1.3807999538298958
Median = 2.9
Mean =  2.9982786885245902
```

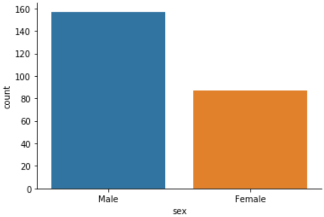
Exploratory Data Analysis

Explore if there is any dependency between the variable "Tip" and rest of the variables.

In [8]:

```
sns.countplot(x='sex', data=tips)
sns.despine() # no top and right axes spine

print(tips.sex.value_counts())
```

```
Male      157
Female     87
Name: sex, dtype: int64
```

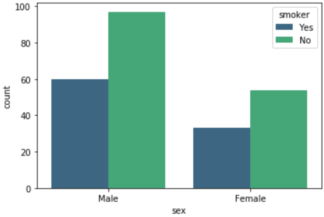


In [9]:

```
sns.countplot(x='sex', data=tips, hue='smoker', palette='viridis')
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f0ded0508>
```

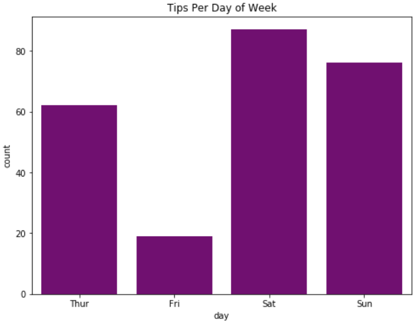


In [10]:

```
plt.figure(figsize=(8,6))
plt.title("Tips Per Day of Week")
sns.countplot(x=tips['day'],color = 'purple')
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f0dfa6b48>
```

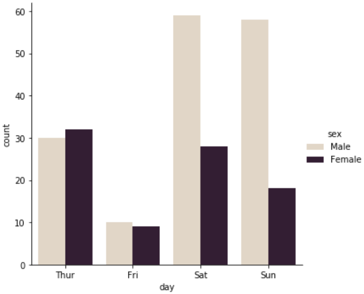


In [11]:

```
sns.catplot(x='day', data=tips, hue='sex', palette='ch:.25', kind='count')
```

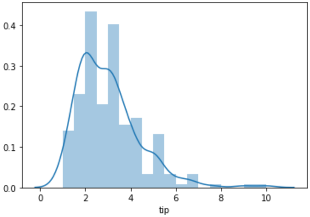
Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x13f0dff5dc8>
```

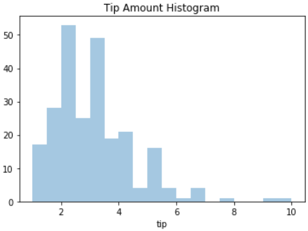


In [12]:
```
sns.distplot(df['tip'])
```

Out[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x13f0e075808>
```



In [13]:
```
g = sns.distplot(tips.tip,kde=False)
g.set_title('Tip Amount Histogram');
```

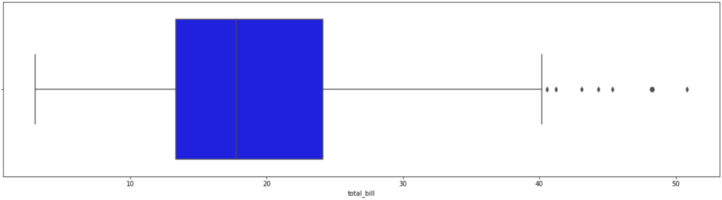


Find out the outliers for "Total Bill" and "Tip"

outliers in bill column

In [14]:
```
plt.figure(figsize=(20,5))

sns.boxplot(x=bill, color='b')
```

Out[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x13f0e1e7ec8>
```



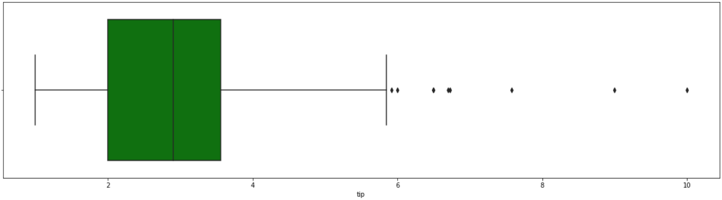
outliers in tip column

In [15]:

```
plt.figure(figsize=(20,5))

sns.boxplot(x=tip, color='g')
```

Out[15]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f0be78ec8>
```



IQR value

In [16]:

```
bill_tip = pd.DataFrame(df,columns=['total_bill','tip','size'])

print(bill_tip)

print("IQR For Total Bill : ",stats.iqr(bill))
print("IQR For Tip : ",stats.iqr(tip))
```

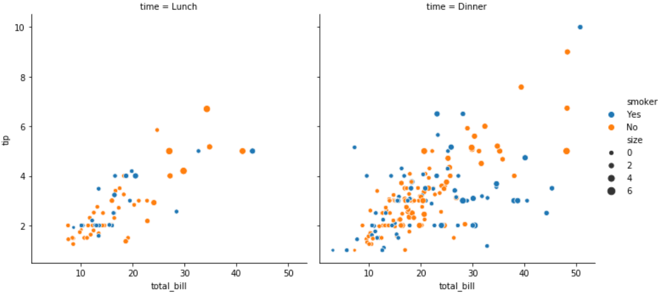
```
     total_bill   tip  size
0         16.99  1.01     2
1         10.34  1.66     3
2         21.01  3.50     3
3         23.68  3.31     2
4         24.59  3.61     4
..          ...   ...   ...
239       29.03  5.92     3
240       27.18  2.00     2
241       22.67  2.00     2
242       17.82  1.75     2
243       18.78  3.00     2

[244 rows x 3 columns]
IQR For Total Bill :  10.779999999999998
IQR For Tip :  1.5625
```

In [17]:

```
sns.relplot(x='total_bill',y='tip',data=df,col='time',hue='smoker',size='size')
```

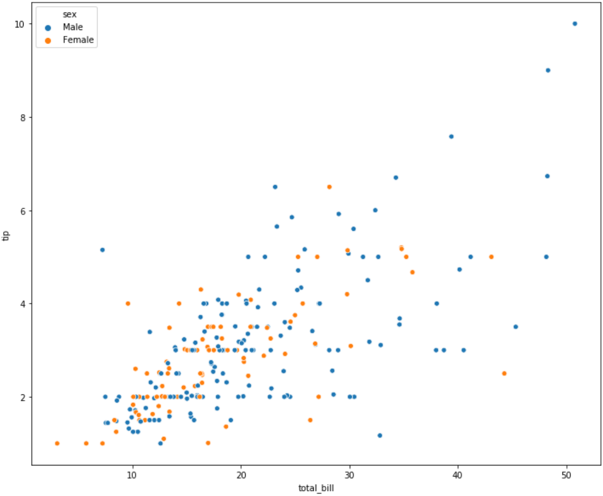
Out[17]:

```
<seaborn.axisgrid.FacetGrid at 0x13f0be788c8>
```



plots based on lunch and dinner based on whether a person is a smoker or not.Can see a linaer pattern ie as total bill increases tip also increases. relplot and lmplot are used for visualizing linear relationships.

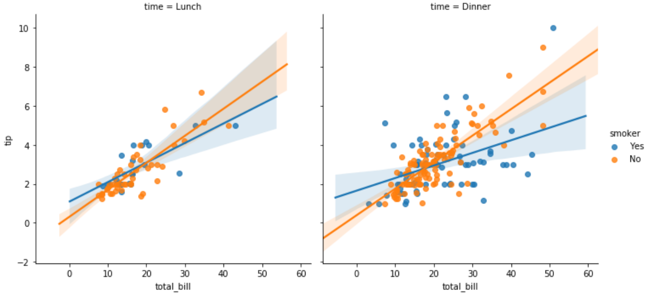
```
In [18]: plt.figure(figsize=(12,10))
         sns.scatterplot(data=df,x="total_bill",y="tip",hue="sex");
```



Can see a linaer pattern ie as total bill increases tip also increases.

```
In [19]: sns.lmplot(x='total_bill',y='tip',data=df,col='time',hue='smoker')
```

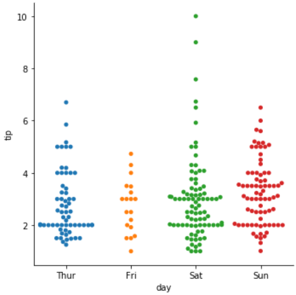
```
Out[19]: <seaborn.axisgrid.FacetGrid at 0x13f0e681188>
```



If bill amount is less,smokers give more tip than non smokers but they give less tips if the bill amount is high.So it is better to serve a non smoker if the bill amount is high.

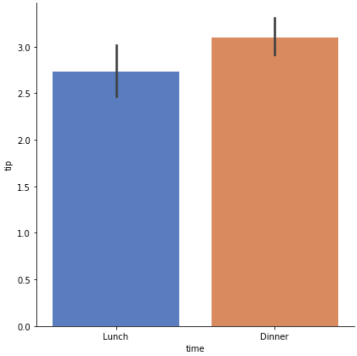
```
In [20]: sns.catplot(x='day',y='tip',data=df,kind='swarm')
```

```
Out[20]: <seaborn.axisgrid.FacetGrid at 0x13f0e925d08>
```



```
In [21]: sns.catplot(x="time", y="tip", data=df, height=6, kind="bar", palette="muted")
```

```
Out[21]: <seaborn.axisgrid.FacetGrid at 0x13f0e995588>
```

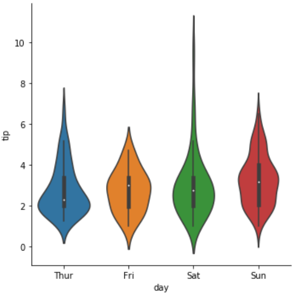


In [22]:

```
sns.catplot(x='day',y='tip',data=df,kind='violin')
```

Out[22]:

```
<seaborn.axisgrid.FacetGrid at 0x13f0e9e3d88>
```

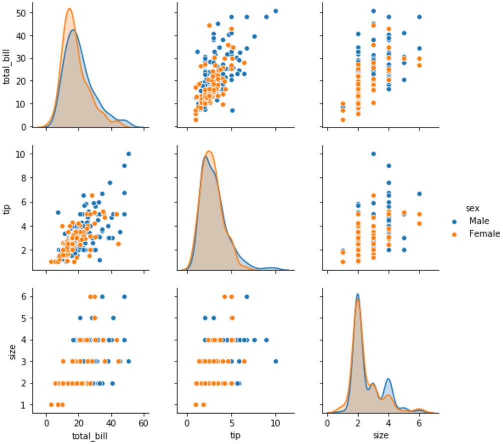


In [23]:

```
sns.pairplot(df, hue='sex')
```

Out[23]:

```
<seaborn.axisgrid.PairGrid at 0x13f0ea64208>
```



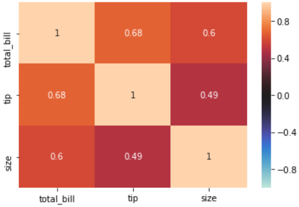
Correlation Matrix

In [24]:

```
corr_matrix=df.corr()
ax=sns.heatmap(data=corr_matrix,annot=True,vmax=1,vmin=-1,center=0)
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
```

Out[24]:

```
(3.0, 0.0)
```



Converting categorical variables into numerical values so that the machine learning model can understand.

In [25]:

```
from sklearn.preprocessing import LabelEncoder
labelencoder_df=LabelEncoder()
df['sex']=labelencoder_df.fit_transform(df['sex'])
df['smoker']=labelencoder_df.fit_transform(df['smoker'])
df['day']=labelencoder_df.fit_transform(df['day'])
df['time']=labelencoder_df.fit_transform(df['time'])
df.head()
```

Out[25]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	0	0	2	0	2
1	10.34	1.66	1	0	2	0	3
2	21.01	3.50	1	0	2	0	3
3	23.68	3.31	1	0	2	0	2
4	24.59	3.61	0	0	2	0	4

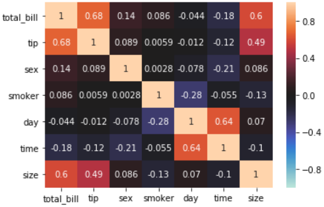
Creating heatmap including these numerical values

In [26]:

```
corr_matrix=df.corr()
ax=sns.heatmap(data=corr_matrix,annot=True,vmax=1,vmin=-1,center=0)
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5 , top - 0.5)
```

Out[26]:

```
(7.0, 0.0)
```



Each square shows the correlation between the variables on each axis. Correlation ranges from -1 to +1. Values closer to zero means there is no linear trend between the two variables. The close to 1 the correlation is the more positively correlated they are; that is as one increases so does the other and the closer to 1 the stronger this relationship is. A correlation closer to -1 is similar, but instead of both increasing one variable will decrease as the other increases. The diagonals are all 1/dark green because those squares are correlating each variable to itself (so it's a perfect correlation). For the rest the larger the number and darker the color the higher the correlation between the two variables. The plot is also symmetrical about the diagonal since the same two variables are being paired together in those squares.