In [1]:
```
from IPython.display import Image
Image(filename='Logo.PNG', height=340, width=900)
```

DATA VISUALIZATION USING SEABORN

1. Seaborn is a statistical plotting library
2. It has statistical graphs like distibution plots, heatmap, etc.
3. It has a variety of graph styles
4. It works in consonance with PANDAS and MATPLOTLIB

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid')
```

In [3]:
```
tips = sns.load_dataset('tips')
```

In [4]:
```
tips.head()
```

Out[4]:

	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 [5]:
```
tips.describe()
```

Out[5]:

	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

In [6]:
```
tips.describe(include='all')
```

Out[6]:

	total_bill	tip	sex	smoker	day	time	size
count	244.000000	244.000000	244	244	244	244	244.000000
unique	NaN	NaN	2	2	4	2	NaN
top	NaN	NaN	Male	No	Sat	Dinner	NaN
freq	NaN	NaN	157	151	87	176	NaN
mean	19.785943	2.998279	NaN	NaN	NaN	NaN	2.569672
std	8.902412	1.383638	NaN	NaN	NaN	NaN	0.951100
min	3.070000	1.000000	NaN	NaN	NaN	NaN	1.000000
25%	13.347500	2.000000	NaN	NaN	NaN	NaN	2.000000

	total_bill	tip	sex	smoker	day	time	size
50%	17.795000	2.900000	NaN	NaN	NaN	NaN	2.000000
75%	24.127500	3.562500	NaN	NaN	NaN	NaN	3.000000
max	50.810000	10.000000	NaN	NaN	NaN	NaN	6.000000

1. DIST PLOT

- Dist plot is used to show the distribution of a univariate set of observations (one variable)

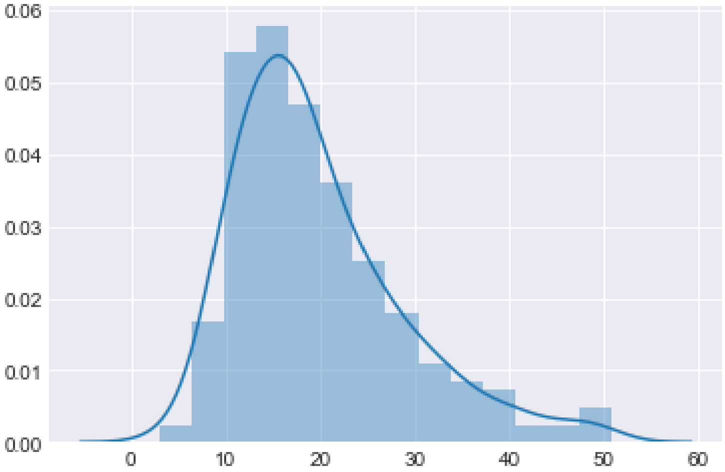
In [7]:

```
sns.distplot(tips['total_bill'])
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:170
6: FutureWarning: Using a non-tuple sequence for multidimensional ind
exing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In
the future this will be interpreted as an array index, `arr[np.array
(seq)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[7]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21e19a84080>
```

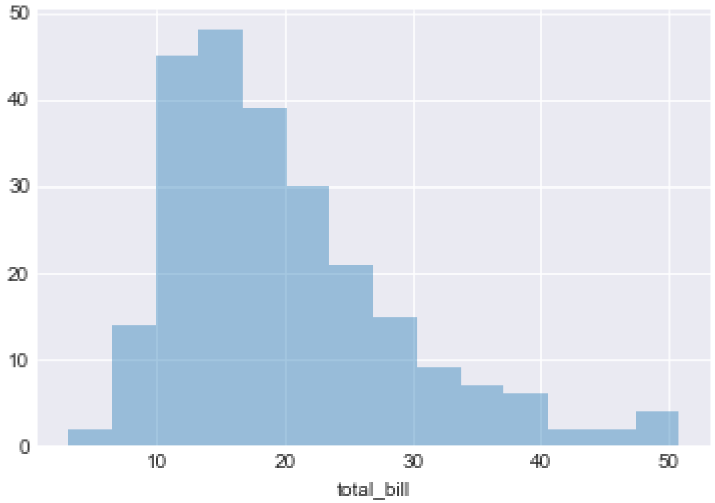


total_bill

In [8]:
```
sns.distplot(tips['total_bill'], kde=False)  # Removing the Kernel Dens
ity Line
```

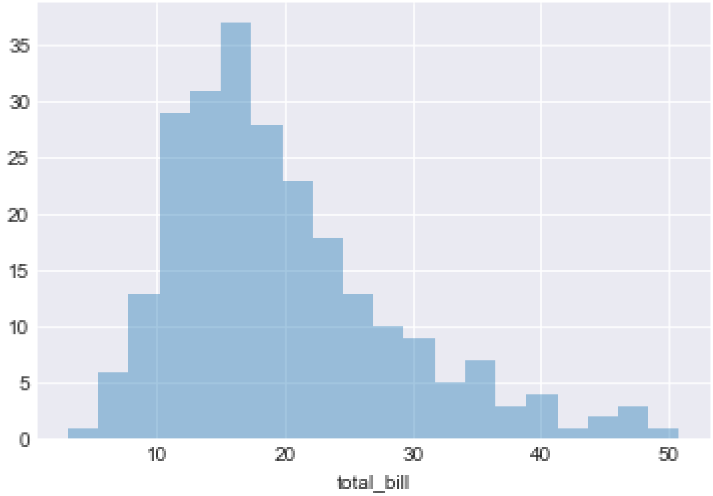
```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21e19aadc18>
```



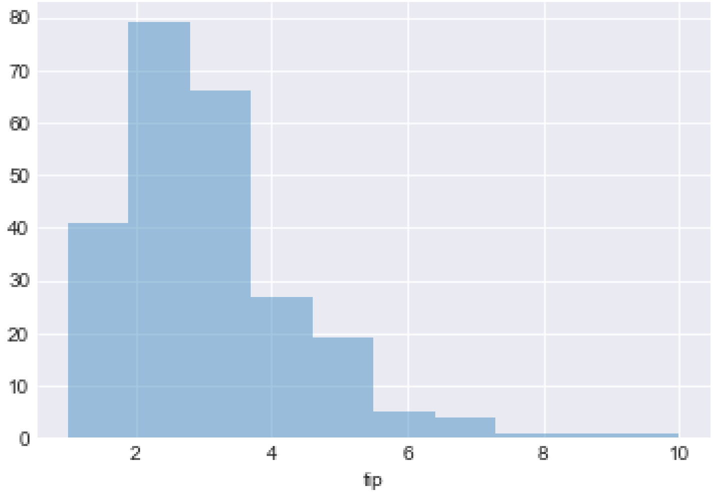
In [9]:
```
sns.distplot(tips['total_bill'], kde=False, bins = 20)
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21e1bbb8550>
```



```
In [10]: sns.distplot(tips['tip'], kde=False, bins = 10)

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x21e1bc6b9b0>
```



2. JOINTPLOT

- **Joint plot is used to show the distribution of a bivariate set of observations (two variables)**

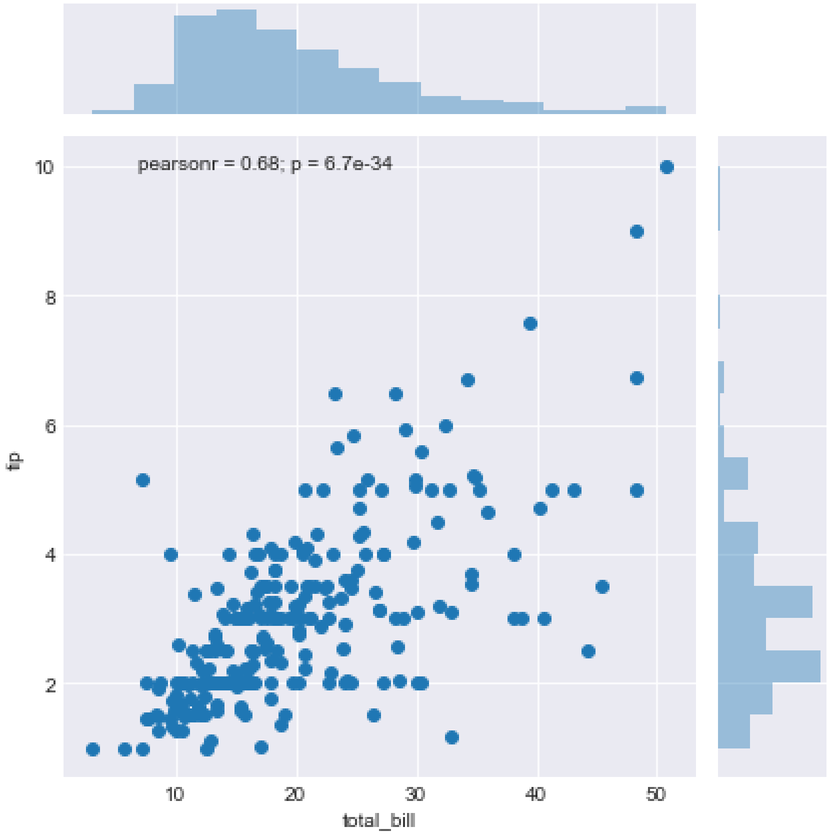
- **It is basically two Dist Plots created**

- **It also shows the scatter plot between the two variables**

```
In [11]: sns.jointplot(x='total_bill', y='tip', data = tips)
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

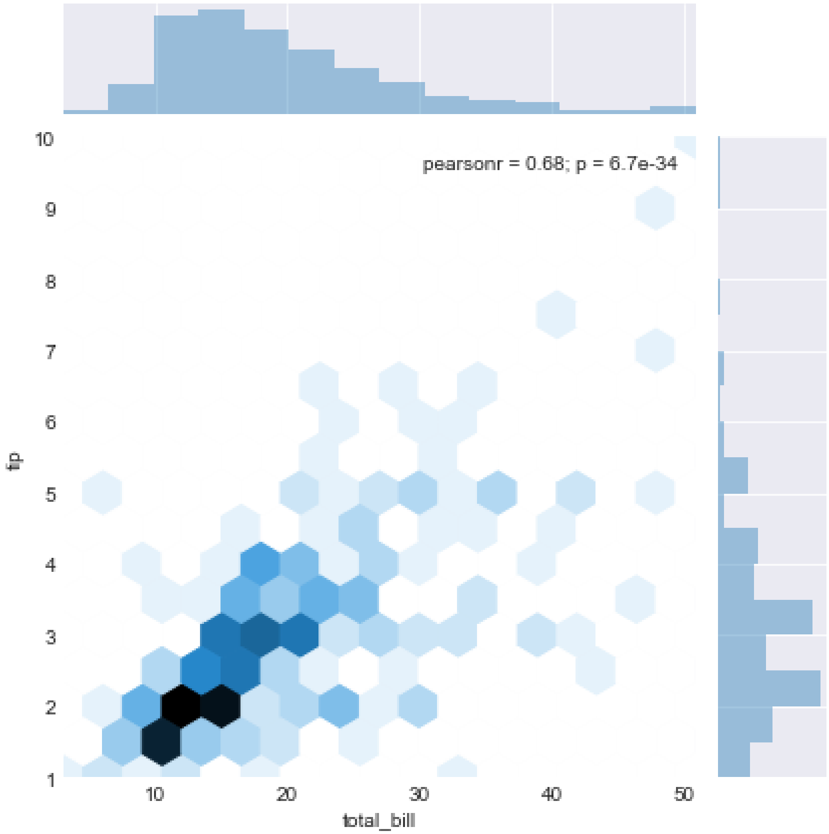
```
Out[11]: <seaborn.axisgrid.JointGrid at 0x21e1bd075f8>
```



In [12]:
```
sns.jointplot(x='total_bill', y='tip', data = tips, kind='hex')
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

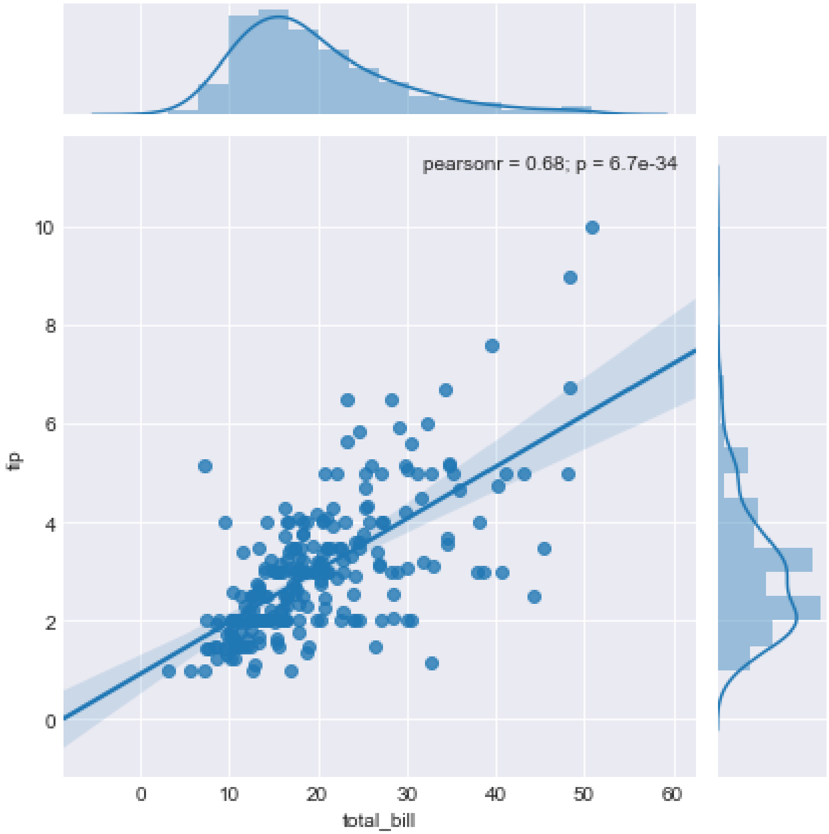
Out[12]:
```
<seaborn.axisgrid.JointGrid at 0x21e1be869e8>
```



```
In [13]: sns.jointplot(x='total_bill', y='tip', data = tips, kind='reg')
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

```
Out[13]: <seaborn.axisgrid.JointGrid at 0x21e1bd68ac8>
```

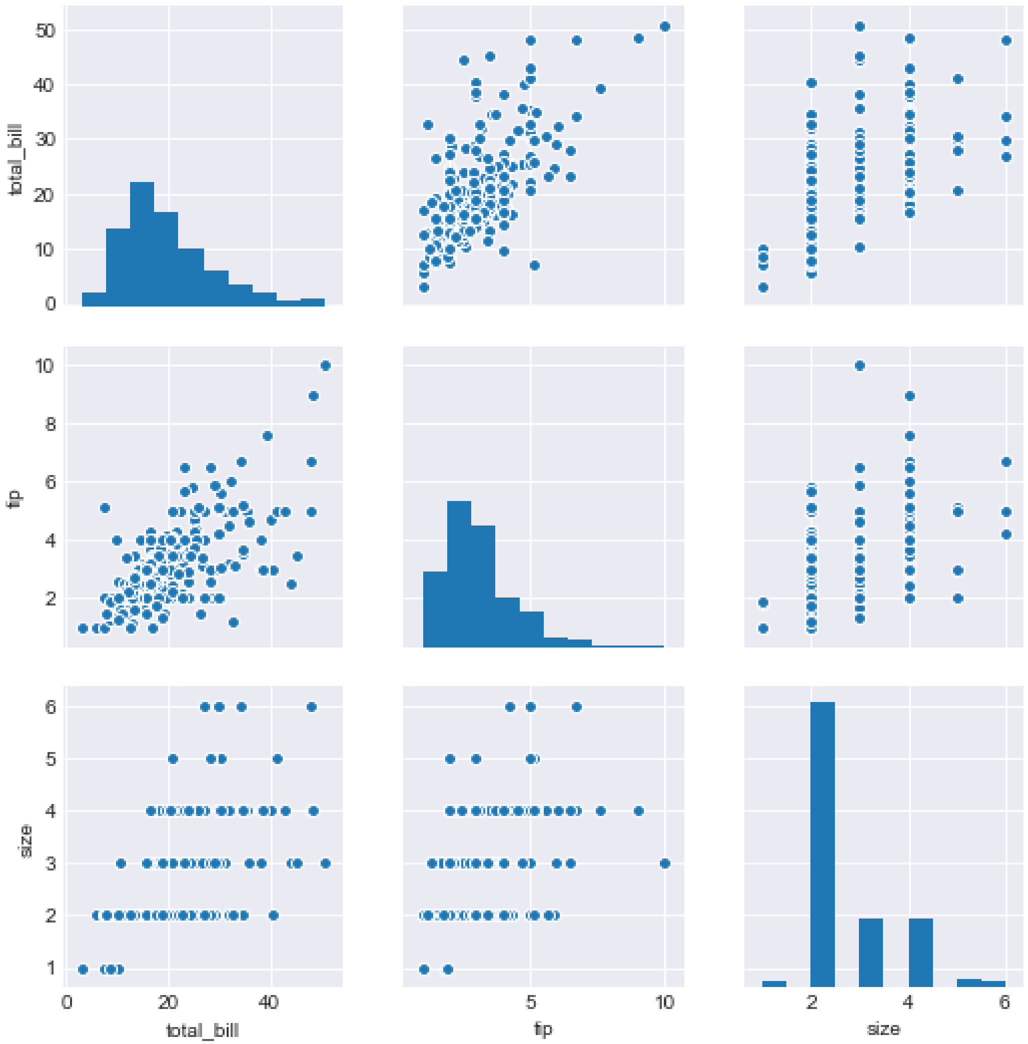


3. PAIRPLOT

- It plots pairwise relationships across an entire data frame at least for the numerical columns.

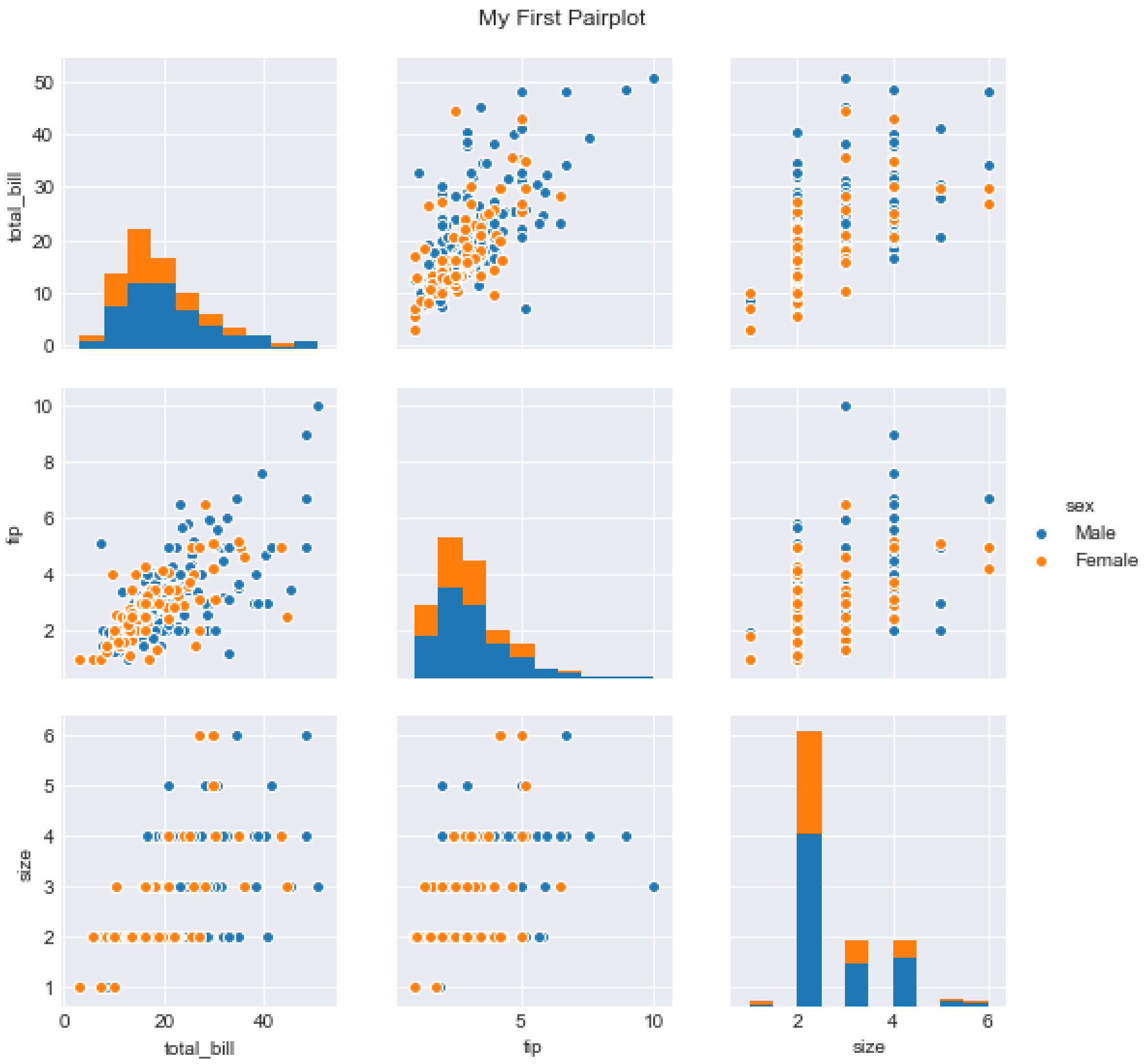
In [14]:
```
sns.pairplot(tips)
```

Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x21e1bd49b38>
```

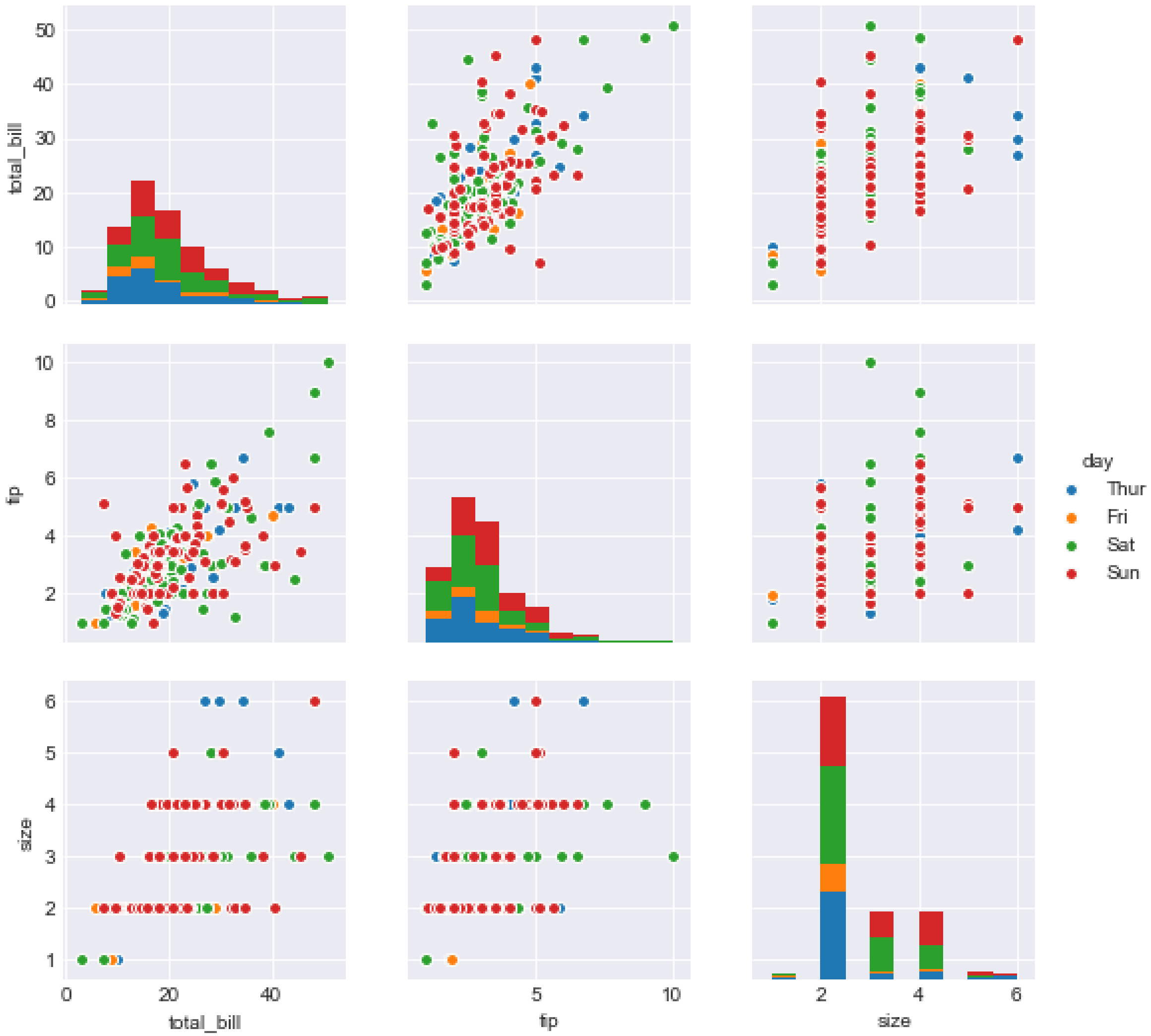


```
In [15]: sns.pairplot(tips, hue='sex')  #Add hue for Categorical Variable
         plt.suptitle(t='My First Pairplot',y=1.02)
```

```
Out[15]: Text(0.5,1.02,'My First Pairplot')
```



```
In [16]: sns.pairplot(tips, hue='day')
Out[16]: <seaborn.axisgrid.PairGrid at 0x21e1d627ac8>
```



PLOTTING CATEGORICAL DATA

4. BARPLOT

- bar plot is just a general plot that allows you to aggregate the categorical data based off of some function and by default that's the mean

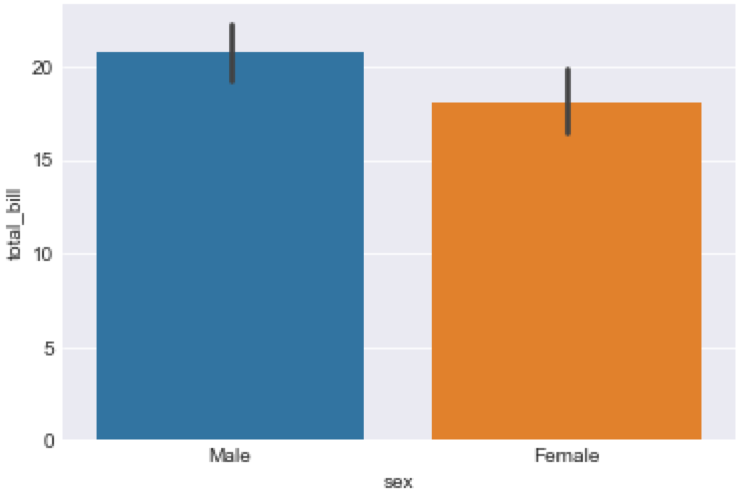
In [17]:

```
sns.barplot(x='sex', y='total_bill', data=tips)
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21e1e54b5f8>
```



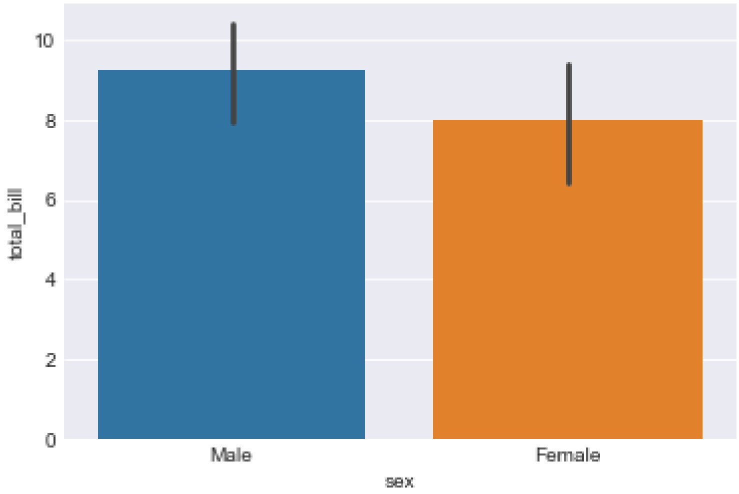
In [18]:

```
sns.barplot(x='sex', y='total_bill', data=tips, estimator=np.std)
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21e1e54b668>
```

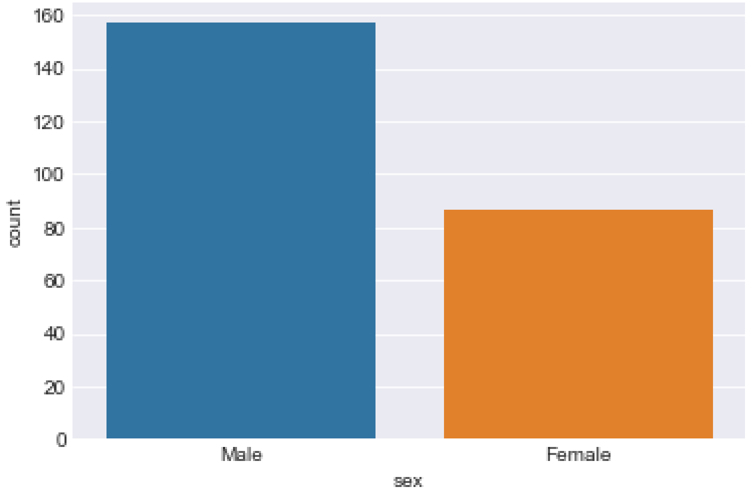


5. COUNTPLOT

- Countplot is a kind of barplot in the which the estimator is the count

```
In [19]: sns.countplot(x='sex', data=tips)
```

```
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x21e1e938240>
```

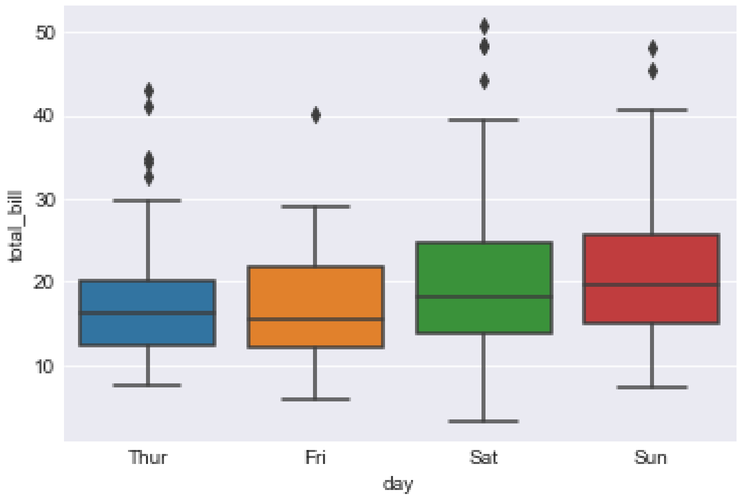


6. BOXPLOT

- are used to show that distribution of categorical data
- And it shows the distribution of quantitative data in a way that hopefully facilitates comparisons between the variables.

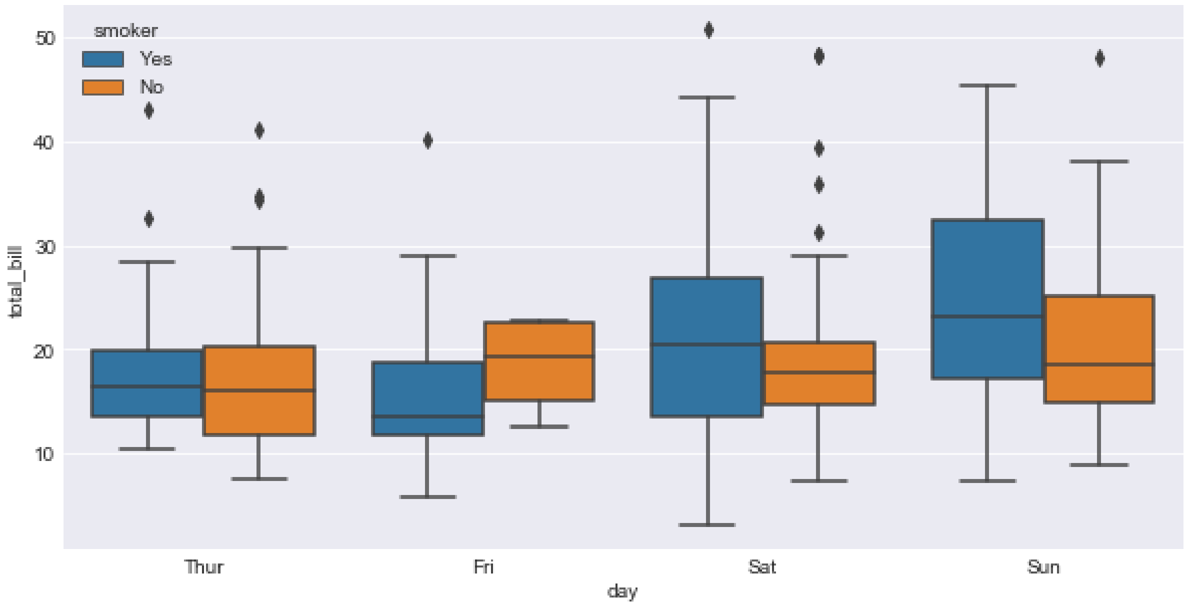
```
In [20]: sns.boxplot(x='day', y='total_bill', data=tips)
```

```
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x21e1e92a6d8>
```



In [21]:
```
plt.figure(figsize=(10,5))
sns.boxplot(x='day', y='total_bill', data=tips, hue='smoker')
```

Out[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21e1e934a90>
```



7. VIOLIN PLOT

- are used to show that distribution of categorical data
- And it shows the distribution of quantitative data in a way that hopefully facilitates comparisons between the variables.
- It shows us a little more information that the box plot in the sense that it also indicates the distribution of the variable

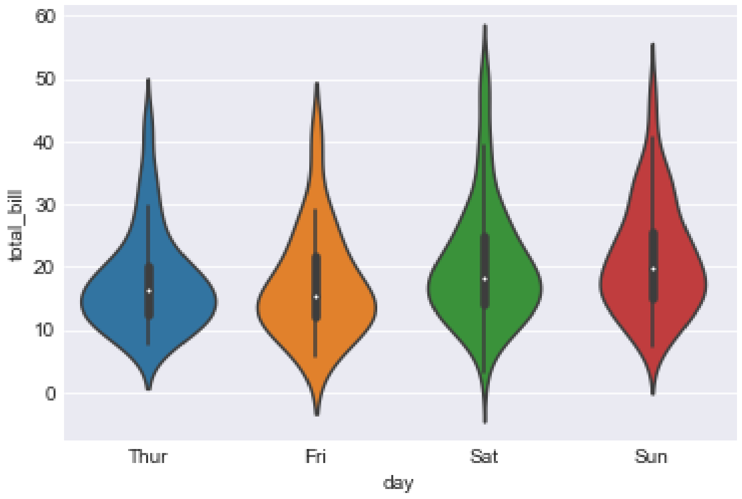
In [22]:

```
sns.violinplot(x='day', y='total_bill', data=tips)
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21e1ea1af60>
```



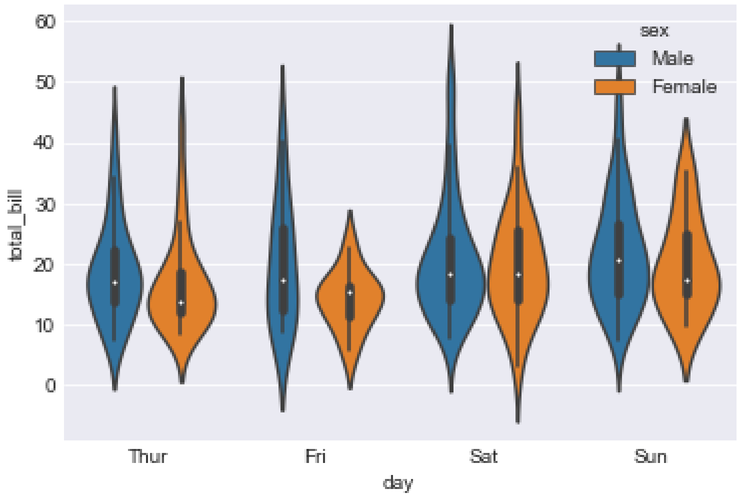
In [23]:

```
sns.violinplot(x='day', y='total_bill', data=tips, hue='sex')
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21e1eb09ba8>
```



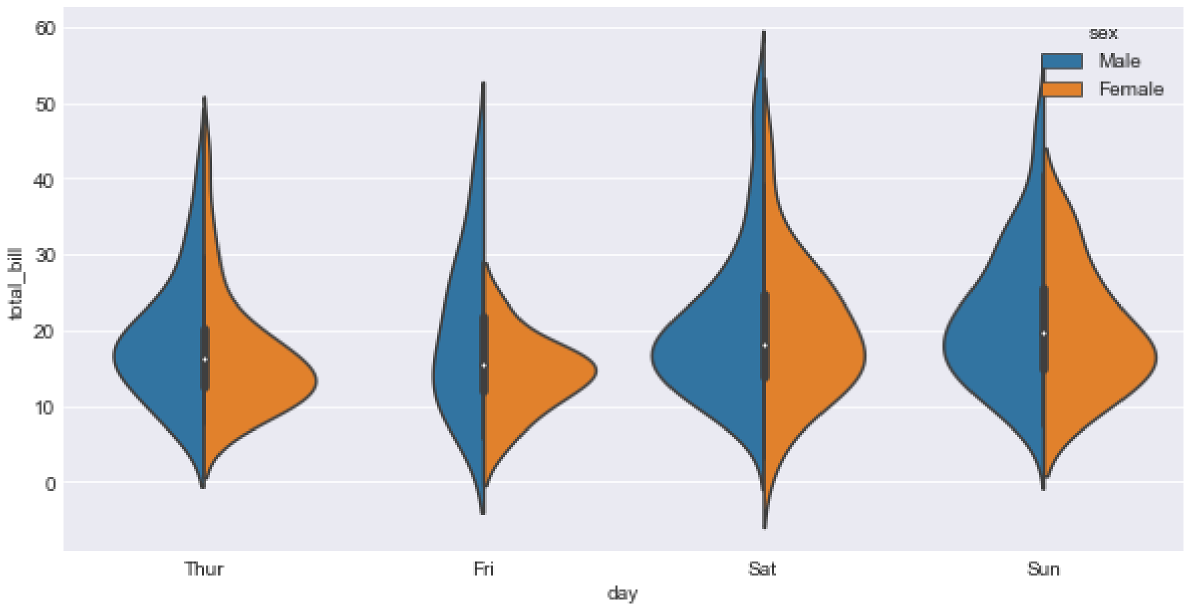
In [24]:

```
plt.figure(figsize=(10,5))
sns.violinplot(x='day', y='total_bill', data=tips, hue='sex', split=Tru
e)
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21e1ead0320>
```



8. HEATMAP

- Needs the data to be in matrix format

In [25]:

```
tips.corr()
```

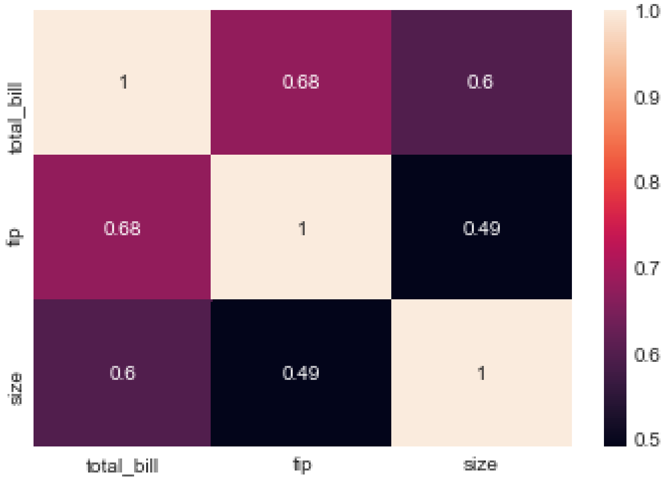
Out[25]:

	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315
tip	0.675734	1.000000	0.489299
size	0.598315	0.489299	1.000000

In [26]:

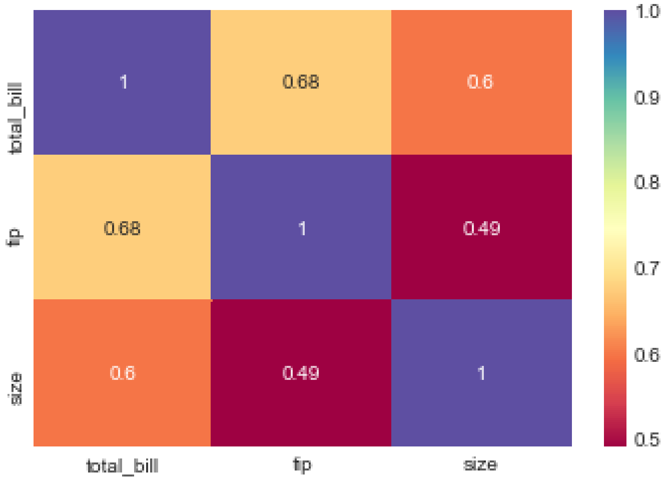
```
sns.heatmap(tips.corr(), annot=True)
```

```
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x21e1fd330b8>
```



```
In [27]: sns.heatmap(tips.corr(), annot=True, cmap='Spectral')
```

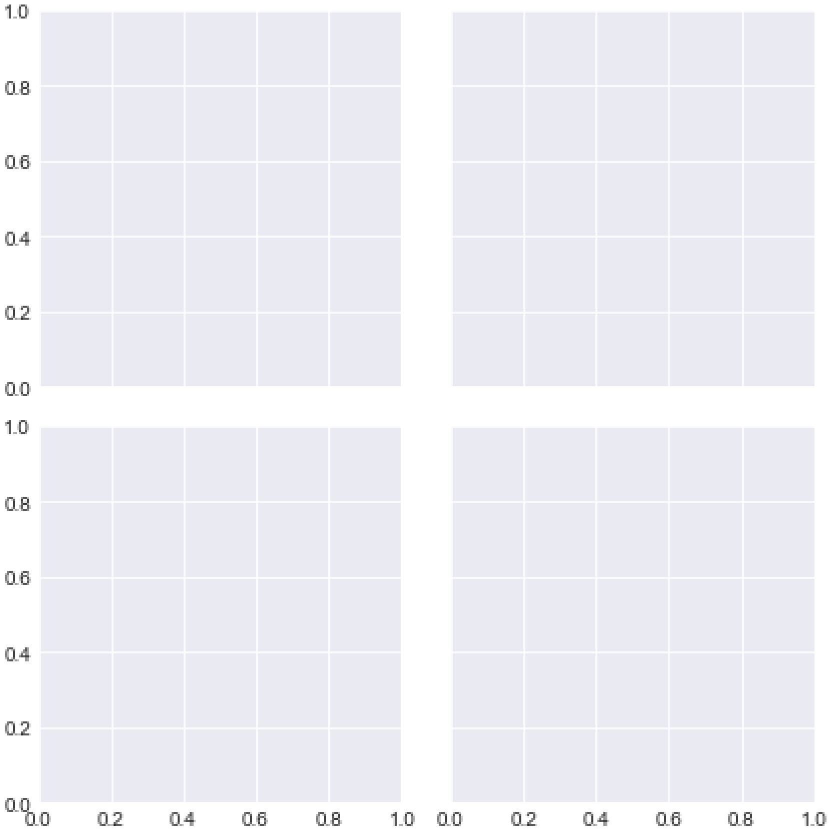
```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x21e1fd558d0>
```



9. FACETGRID

In [28]:
```
sns.FacetGrid(data=tips, col='time', row='smoker')  #CREATES A BLANK GR
ID
```

Out[28]:
```
<seaborn.axisgrid.FacetGrid at 0x21e1ff5fef0>
```

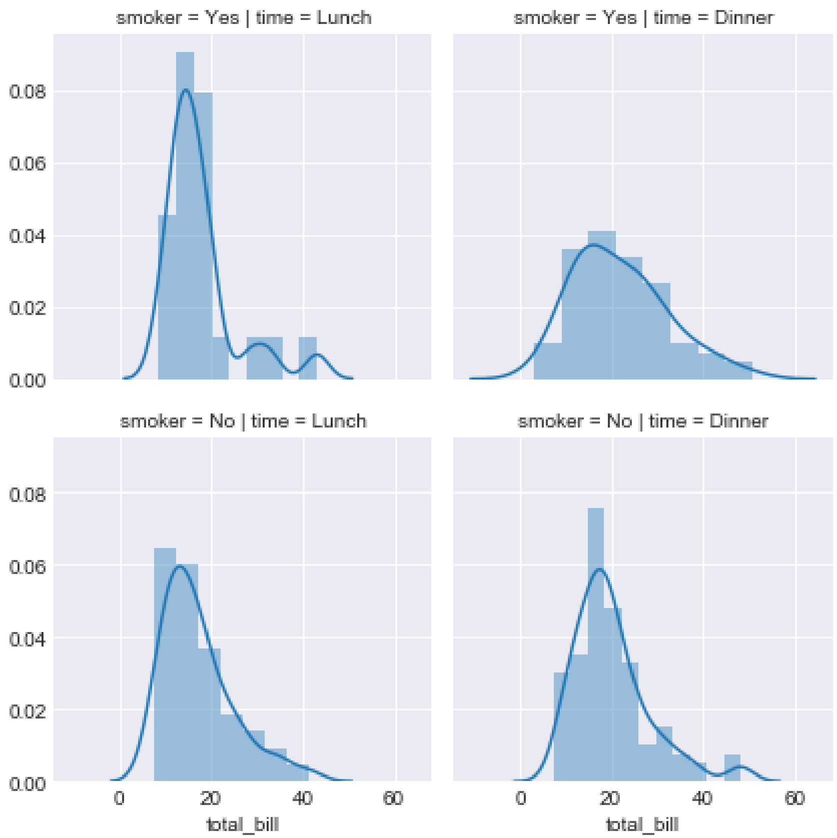


In [29]:
```
g=sns.FacetGrid(data=tips, col='time', row='smoker')
g.map(sns.distplot,'total_bill')
```

```
C:\Users\Shivani\Anaconda3\lib\site-packages\scipy\stats\stats.py:1706:
FutureWarning: Using a non-tuple sequence for multidimensional indexing
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
re this will be interpreted as an array index, `arr[np.array(seq)]`, wh
```

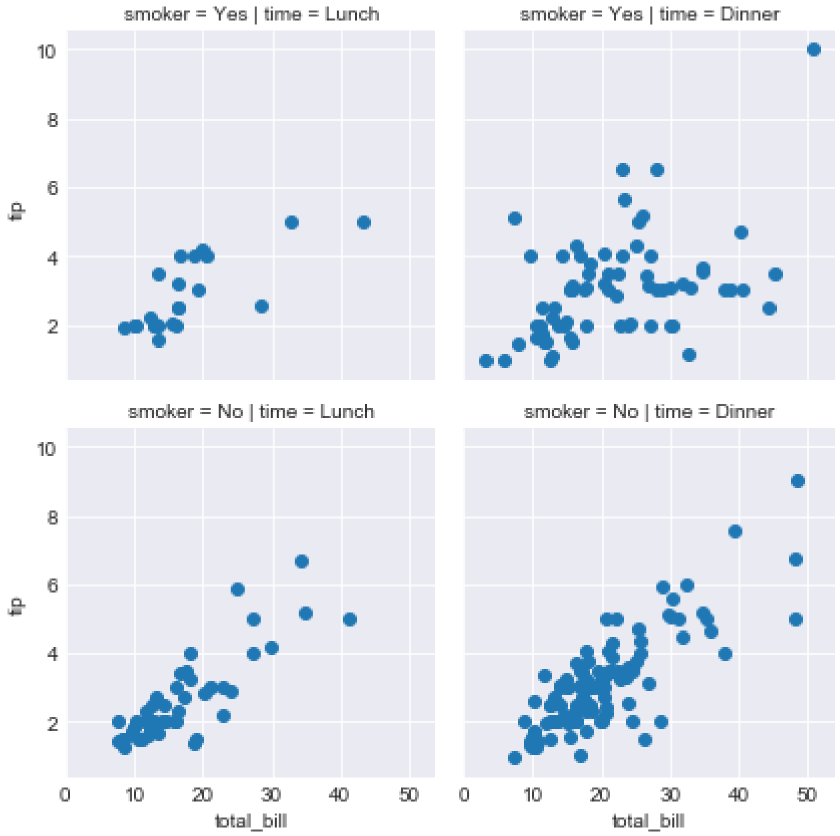
```
ich will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[29]: `<seaborn.axisgrid.FacetGrid at 0x21e1fecbba8>`



In [30]:
```
g=sns.FacetGrid(data=tips, col='time', row='smoker')
g.map(plt.scatter,'total_bill', 'tip')
```

Out[30]: `<seaborn.axisgrid.FacetGrid at 0x21e201dfef0>`


smoker = Yes | time = Lunch
smoker = Yes | time = Dinner
smoker = No | time = Lunch
smoker = No | time = Dinner
tip
tip
total_bill
total_bill
0
10
20
30
40
50
2
4
6
8
10