Data Analytics & Machine Learning Using Python

DataRitz Technologies
Enhancing Technology Experience

Session: 10 June 2020

Data Visualization using Seaborn

- Seaborn is an open source, licensed python library providing high level API for visualiazing the data using Python Programming Language
- Seaborn helps resolving two major problems faced by Matplotlib
  - Default Matplotlib Parameters
  - Working with DataFrames

Important Features of Seaborn

- Visualizing Univariate and Bivariate data
- Visualizing Linear regression models
- Plotting statistical Time-Series Data
- Seaborn works well with Numpy & Pandas Data Structures
- It has built in themes for styling matplotlib graphics

In [1]:

```
# importing Libraries

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

In [5]:

```
print(sb.get_dataset_names(),end=" ")
```

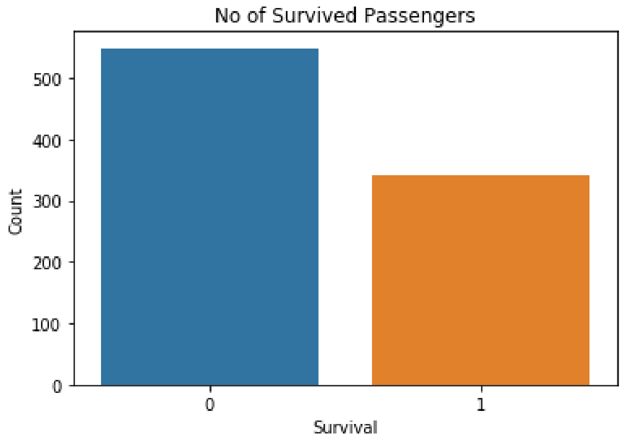
```
C:\Users\Shashank Shekhar\Anaconda3\lib\site-packages\seaborn\utils.py:384: UserWarning: No parser
was explicitly specified, so I'm using the best available HTML parser for this system ("lxml"). Th
is usually isn't a problem, but if you run this code on another system, or in a different virtual
environment, it may use a different parser and behave differently.

The code that caused this warning is on line 384 of the file C:\Users\Shashank Shekhar\Anaconda3\l
ib\site-packages\seaborn\utils.py. To get rid of this warning, pass the additional argument 'featu
res="lxml"' to the BeautifulSoup constructor.

  gh_list = BeautifulSoup(http)

['anscombe', 'attention', 'brain_networks', 'car_crashes', 'diamonds', 'dots', 'exercise', 'flight
s', 'fmri', 'gammas', 'geyser', 'iris', 'mpg', 'planets', 'tips', 'titanic']
```

In [14]:
```
data = pd.read_csv('titanic.csv')
data.head(10)
```

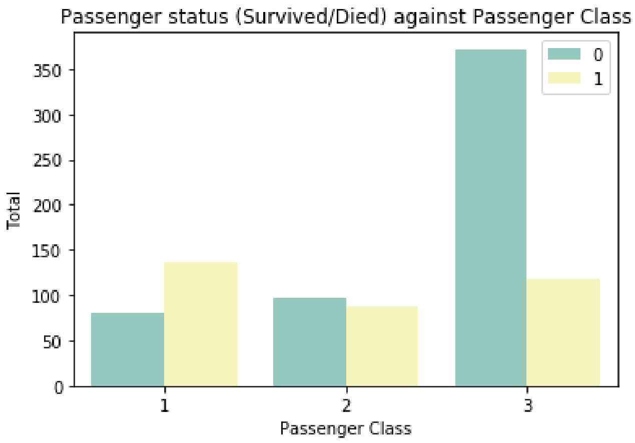
Out[14]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	aliv
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	n
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	ye
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	ye
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	ye
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	n
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	NaN	Queenstown	n
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	E	Southampton	n
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	NaN	Southampton	n
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	NaN	Southampton	ye
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	NaN	Cherbourg	ye

In [15]:
```
sb.countplot(data['survived'])
plt.title("No of Survived Passengers")
plt.xlabel('Survival')
plt.ylabel('Count')
plt.show()
```



In [27]:
```
3survive = sb.countplot(x = 'pclass', hue = 'survived', palette = 'Set3', data = data)
survive.set(title = 'Passenger status (Survived/Died) against Passenger Class', xlabel = 'Passenger
plt.legend(loc='upper right')
plt.show()
```



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

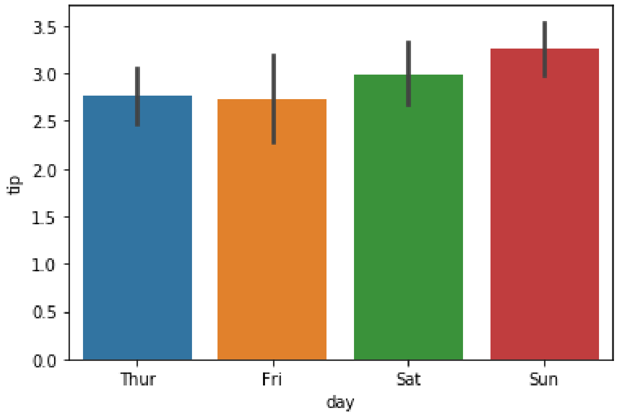
Out[3]:

	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
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

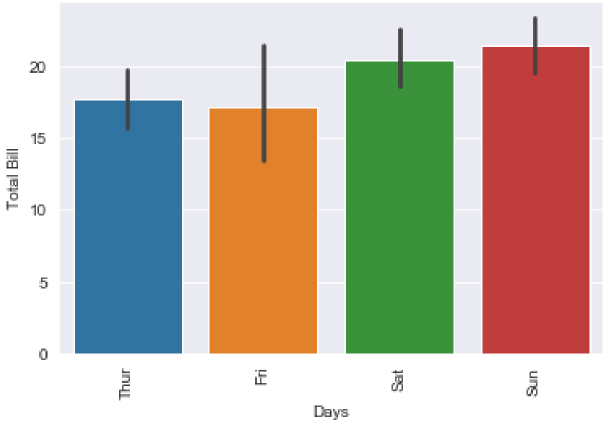
In [30]:
```
# bar graph plot 1
sns.barplot(x='day',y='tip',data=tips)
```

Out[30]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1af66fa88>
```



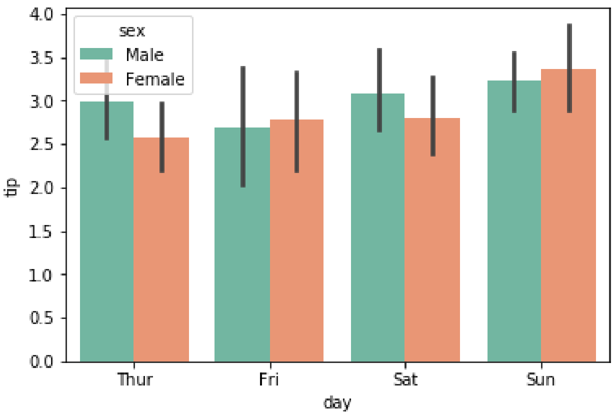
In [138]:
```
# bar graph plot 2
sns.barplot(x='day',y='total_bill',data=tips)
plt.xlabel('Days')
plt.ylabel('Total Bill')
plt.xticks(rotation=90)
```

Out[138]:
```
(array([0, 1, 2, 3]), <a list of 4 Text xticklabel objects>)
```



In [38]:
```
# bar graph with hue and palette
sns.barplot(x='day',y='tip',hue='sex',data=tips,palette='Set2')
```

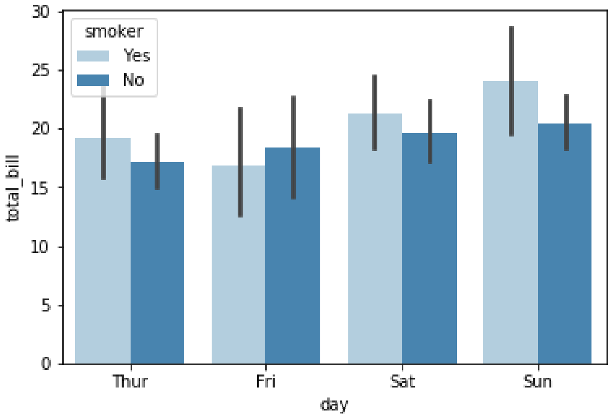
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1af8e64c8>
```



In [39]:
```
sns.barplot?
```

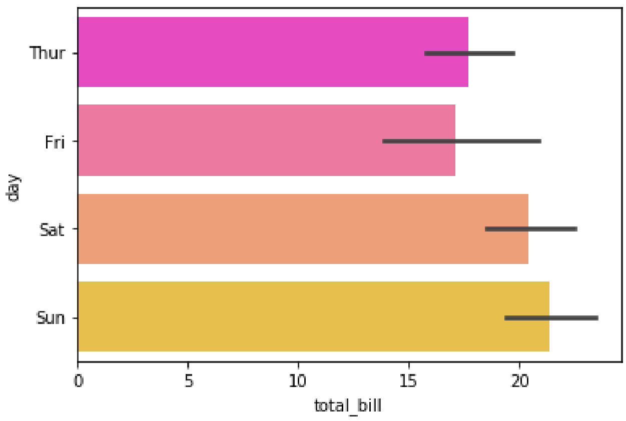
In [49]:
```
# bar graph with hue_smoker and palette
sns.barplot(x='day',y='total_bill',data=tips,hue='smoker',palette='Blues')
```

Out[49]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1af46ce08>
```



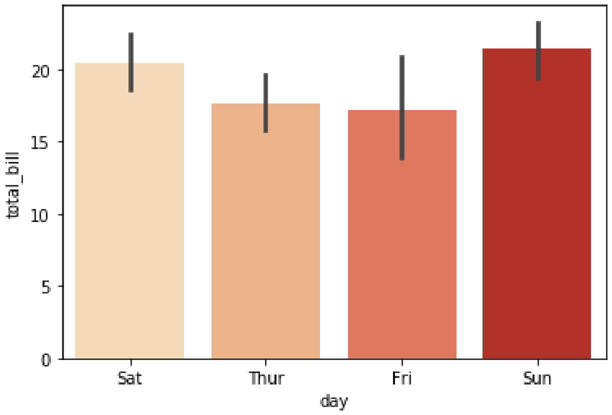
In [50]:
```
# horizontal bar plot
sns.barplot(x='total_bill',y='day',data=tips,palette='spring')
```

Out[50]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1aebabcc8>
```



```
In [60]: # bar plot order
         sns.barplot(x='day',y='total_bill',data=tips,palette='OrRd',order=['Sat','Thur','Fri','Sun'])
```

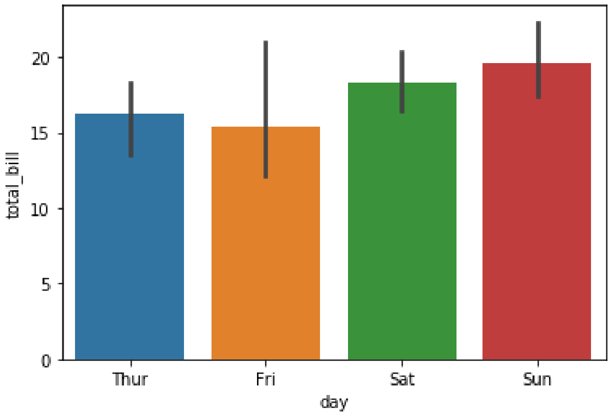
```
Out[60]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b0bc1b08>
```



```
In [61]: sns.barplot?
```

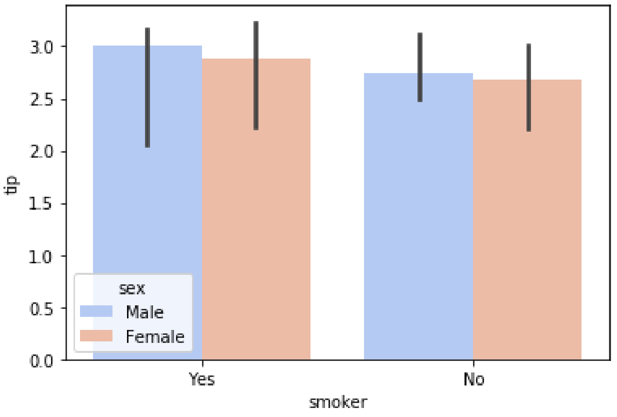
```
In [63]: # bar plot median in place of mean
         from numpy import median
         sns.barplot(x='day',y='total_bill',data=tips,estimator=median)
```

```
Out[63]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b0c36208>
```



```
In [67]: # new bar plot smoker & tip and palette
         sns.barplot(x='smoker',y='tip',data=tips,estimator=median,hue='sex',palette='coolwarm')
```

```
Out[67]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b0ce0388>
```

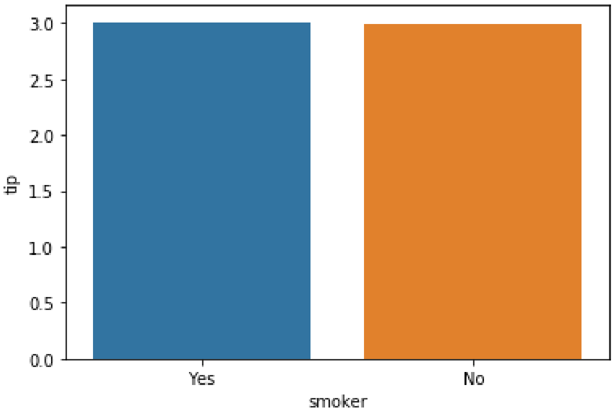


In [6]:

```
# bar plot with attribute CI
sns.barplot(x='smoker',y='tip',data=tips,ci=0)
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e09cd87988>
```

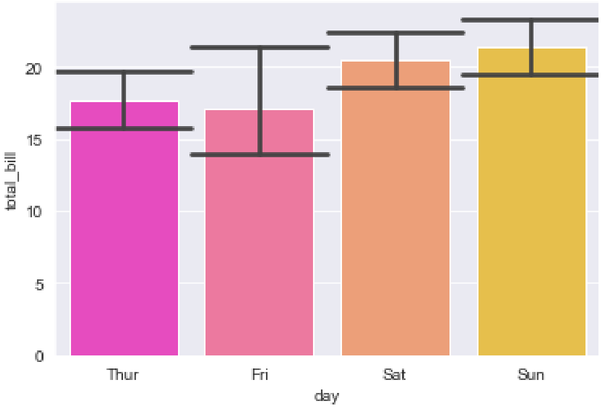


In [104]:

```
# bar plot with capsize
sns.barplot(x='day',y='total_bill',data=tips,palette='spring',capsize=1)
```

Out[104]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b2aaf4c8>
```

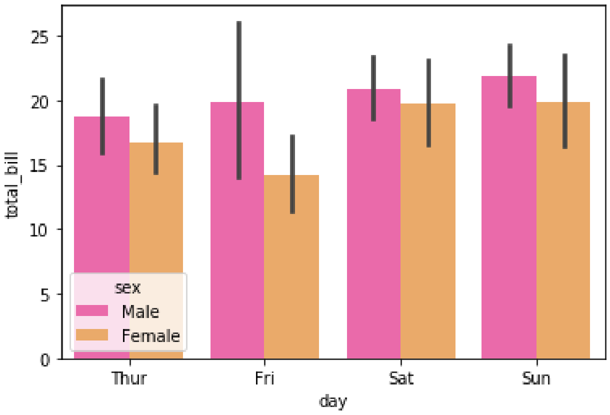


In [80]:

```
# bar plot with capsize & hue
sns.barplot(x='day',y='total_bill',data=tips,palette='spring',capsize=0,hue='sex')
```

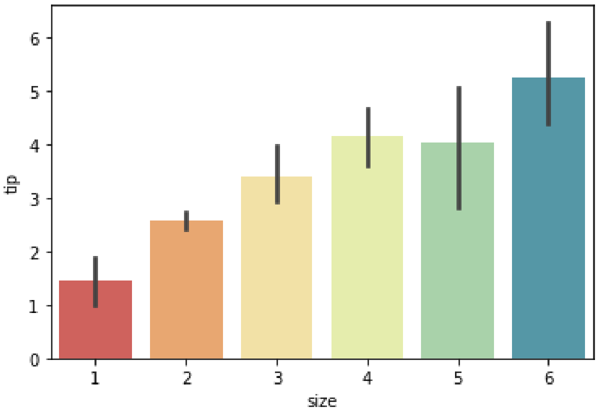
Out[80]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b11a3fc8>
```



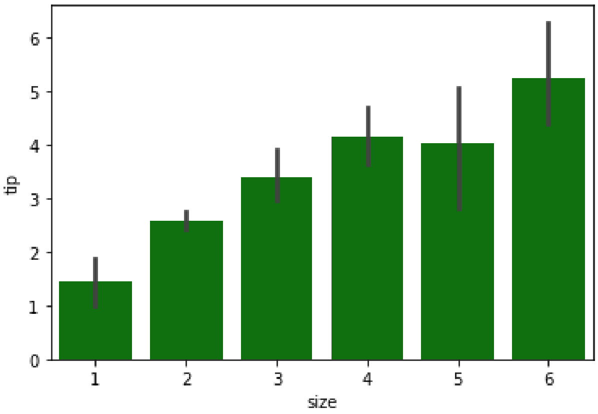
```
In [10]: # new bar plot with size and tip
         sns.barplot(x='size',y='tip',data=tips,palette='Spectral',capsize=0)
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1e09cfcb488>
```



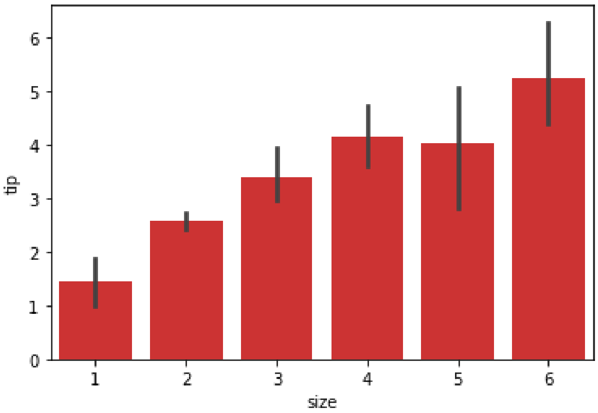
```
In [83]: # bar plot with color instead of palette
         sns.barplot(x='size',y='tip',data=tips,color='green',capsize=0)
```

```
Out[83]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b131ecc8>
```



```
In [88]: # bar plot with saturation
         sns.barplot(x='size',y='tip',data=tips,color='red',capsize=2,saturation=0.6)
```

```
Out[88]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b1632308>
```



Dist Plot

In [89]:
```
#num=np.random.randn(150)
```

In [96]:
```
tips
```

Out[96]:

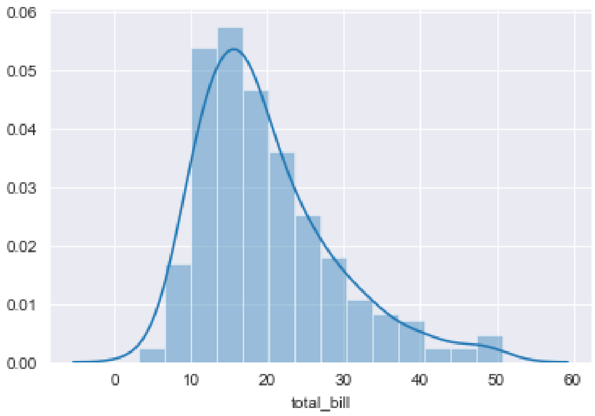
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

In [115]:
```
sns.distplot?
```

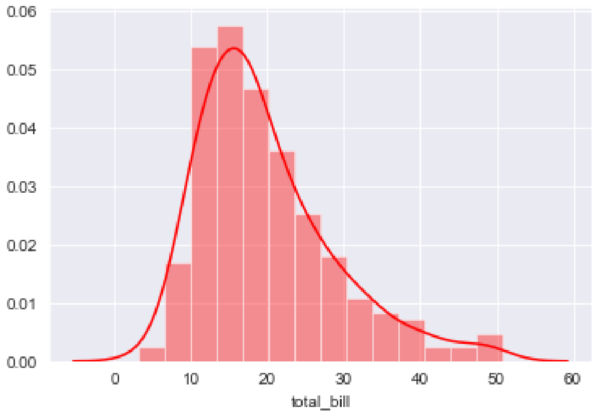
In [97]:
```
# dist plotting
sns.set_style('darkgrid')
sns.distplot(tips['total_bill'])
```

Out[97]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b2786108>
```



In [111]:
```
# dist plot with different color
sns.distplot(tips['total_bill'],color='red')
```

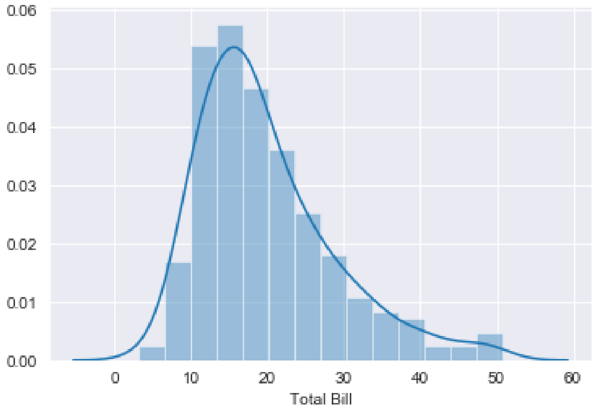
Out[111]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b2c1b148>
```



```
In [100]: # dist plot with name on x-axis
          label_x=pd.Series(tips['total_bill'],name='Total Bill')
```

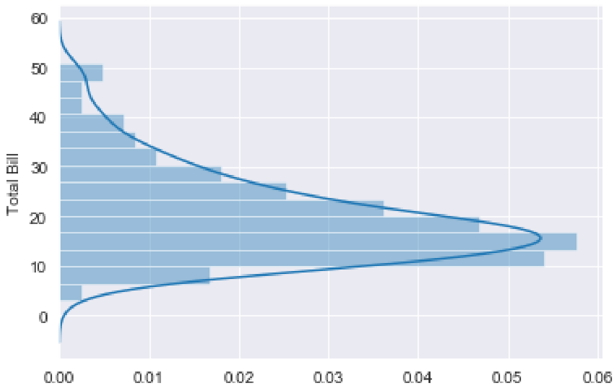
```
In [101]: # plot dist plot
          sns.distplot(label_x)
```

```
Out[101]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b2993d48>
```



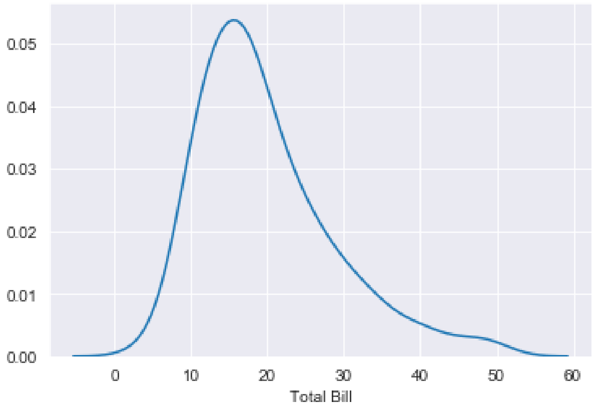
```
In [112]: # dist plot with vertical
          sns.distplot(label_x, vertical=True)
```

```
Out[112]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b2deba08>
```



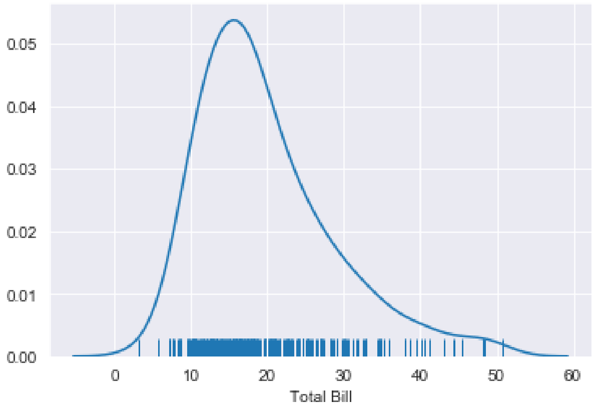
```
In [113]: # dist plot with hist
          sns.distplot(label_x,hist=False)
```

```
Out[113]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b2e84248>
```



In [114]:
```
# dist plot with rug
sns.distplot(label_x, rug=True, hist=False)
```

Out[114]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b2e48e48>
```



Box Plot

In [116]:
```
# load tips
tips
```

Out[116]:

	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
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

In [117]:
```
tips.head()
```

Out[117]:

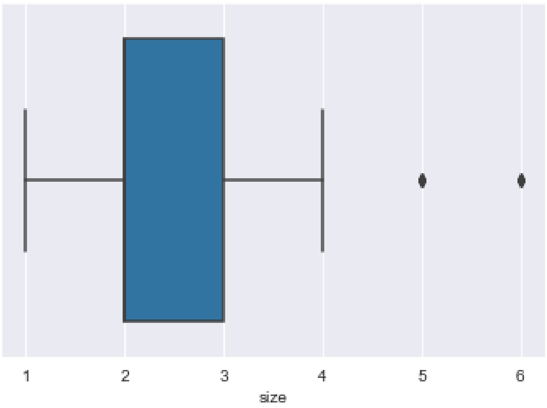
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [118]:

```
# horizontal box plot for size
sns.boxplot(x=tips['size'])
```

Out[118]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b2f74788>
```

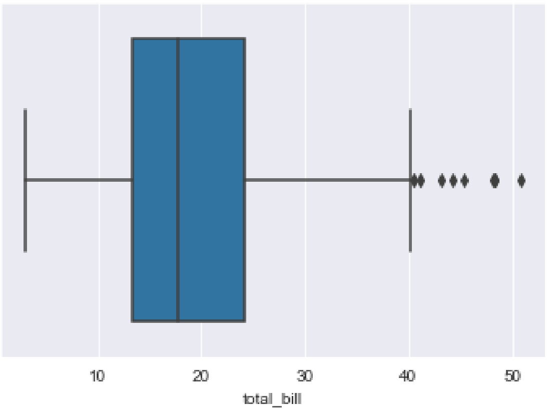


In [119]:

```
# box plot for total_bill
sns.boxplot(x=tips['total_bill'])
```

Out[119]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b2fb0448>
```

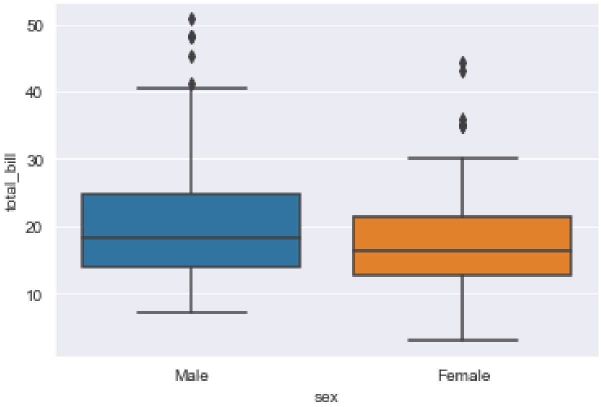


In [120]:

```
# box plot for 2 variable
sns.boxplot(x='sex',y='total_bill',data=tips)
```

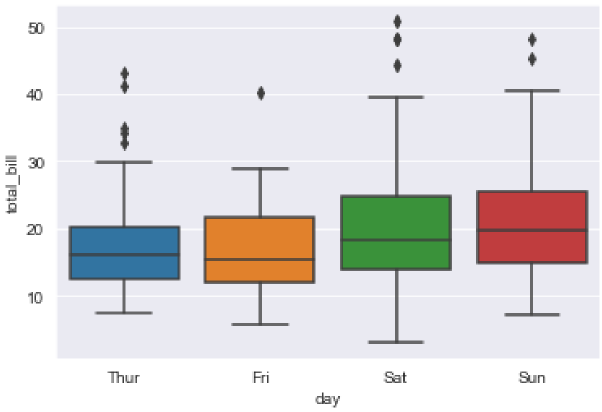
Out[120]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b302ae88>
```



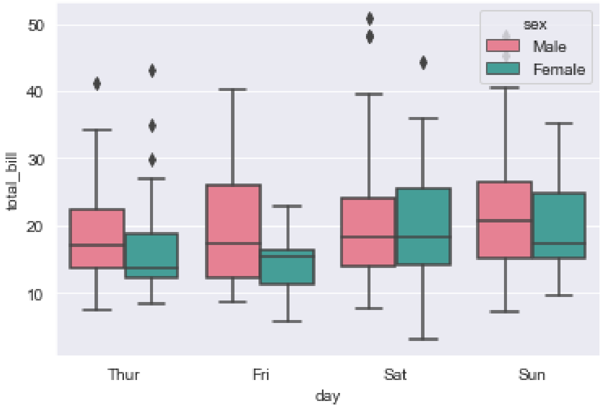
```
In [122]: sns.boxplot(x='day',y='total_bill',data=tips)
```

```
Out[122]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b30c7d88>
```



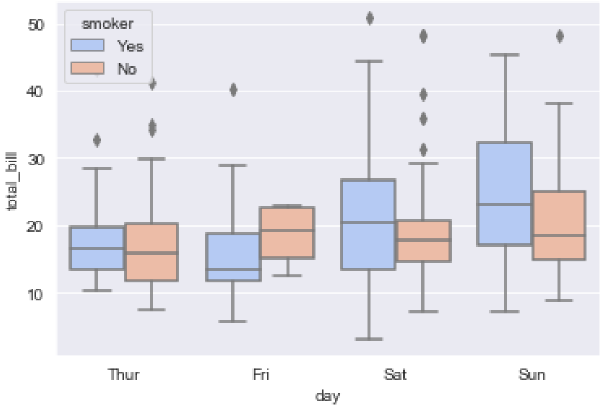
```
In [125]: # box plot with hue
          sns.boxplot(x='day',y='total_bill',data=tips, hue='sex',palette='husl')
```

```
Out[125]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b4245788>
```



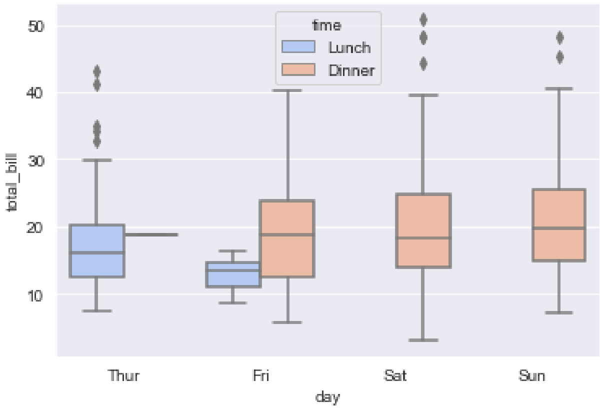
```
In [126]: sns.boxplot(x='day',y='total_bill',data=tips, hue='smoker',palette='coolwarm')
```

```
Out[126]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b43294c8>
```



```
In [127]: # box plot with hue=time
          sns.boxplot(x='day',y='total_bill',data=tips, hue='time',palette='coolwarm')
```

```
Out[127]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b4415608>
```

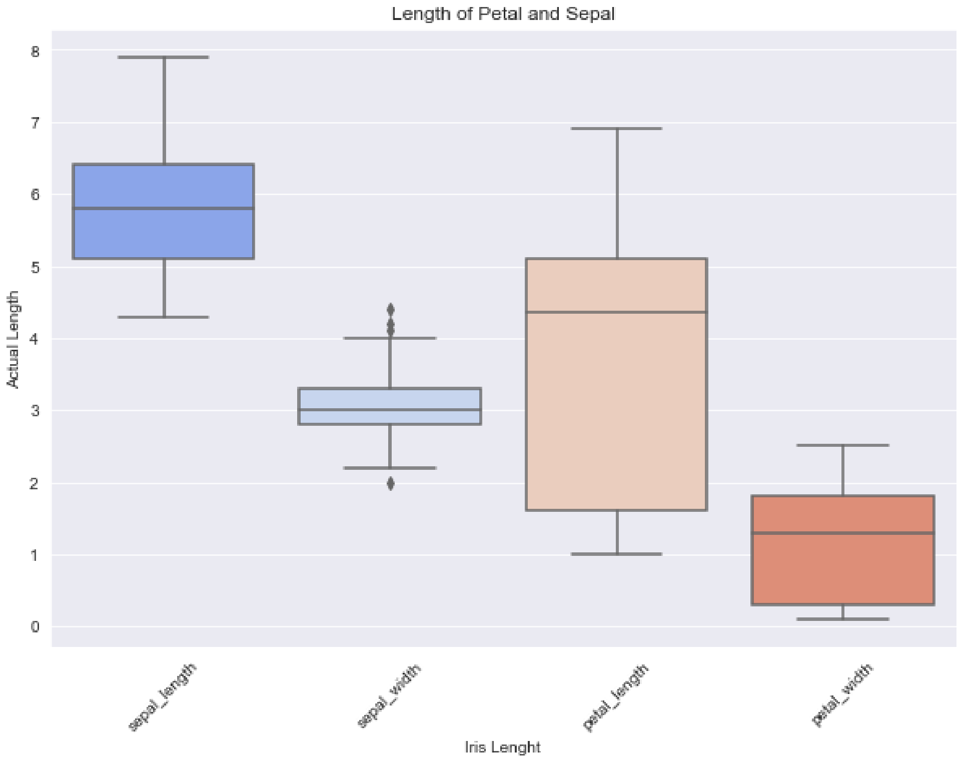


```
In [130]: # Load Iris dataset
          iris=sns.load_dataset('iris')
          iris.head()
```

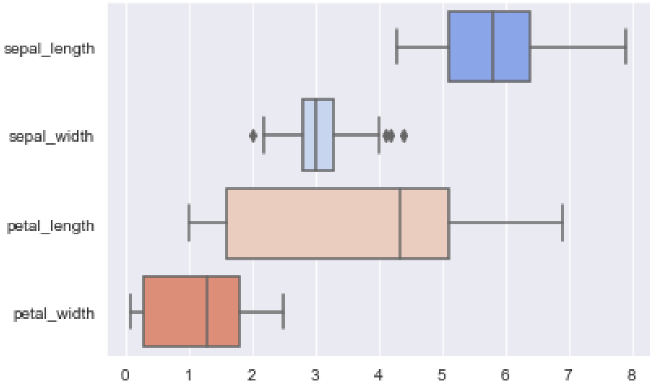
Out[130]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
In [137]: # plot box
          plt.figure(figsize=(10,7))
          sns.boxplot(data=iris,palette='coolwarm')
          plt.xticks(rotation=45)
          plt.xlabel('Iris Lenght')
          plt.ylabel('Actual Length')
          plt.title('Length of Petal and Sepal')

Out[137]: Text(0.5, 1.0, 'Length of Petal and Sepal')
```

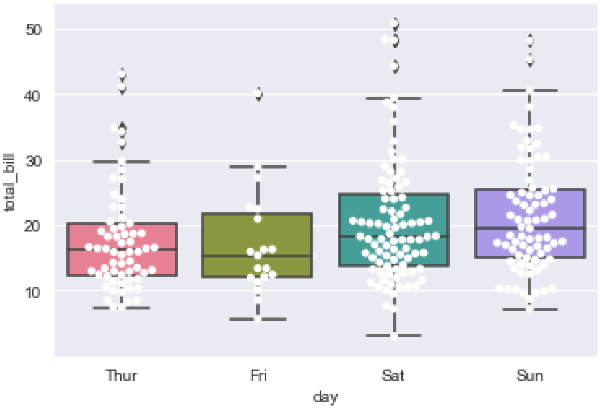


```
In [134]: # box plot with orient
          sns.boxplot(data=iris,palette='coolwarm',orient='horizontal')

Out[134]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b465ff48>
```



In [142]:
```
#tips data and swarm plot
sns.boxplot(x='day',y='total_bill',data=tips,palette='husl')
sns.swarmplot(x='day',y='total_bill',data=tips,color='1')
```

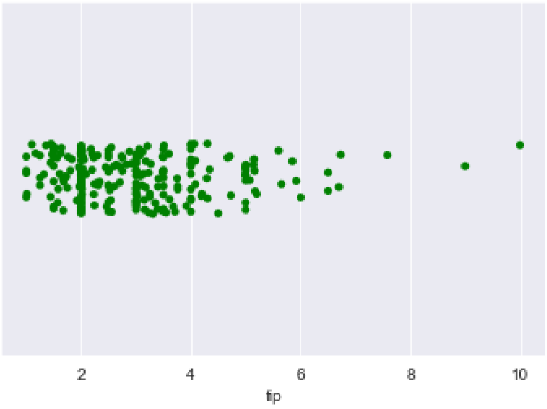
Out[142]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b4d05d48>
```



Strip Plot

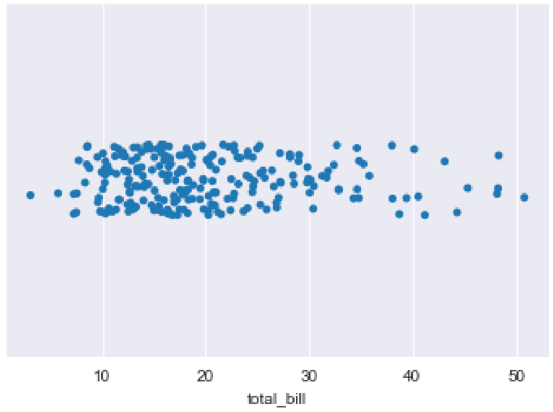
In [144]:
```
sns.stripplot(x=tips['tip'],color='green')
```

Out[144]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b4e0ec48>
```



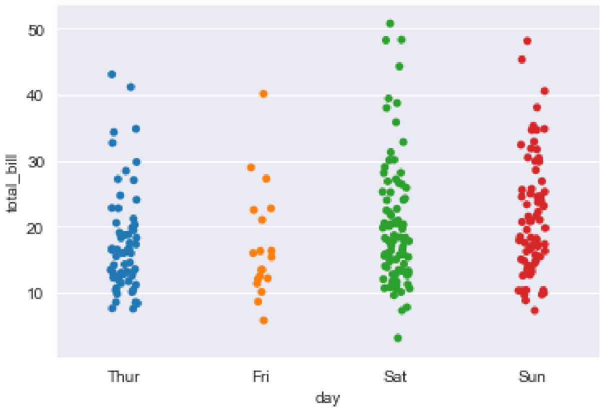
```
In [145]: sns.stripplot(x=tips['total_bill'])
```

```
Out[145]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b4df6f08>
```



```
In [156]: # strip plot for 2 variables
          sns.stripplot(x='day',y='total_bill',data=tips,jitter=1)
```

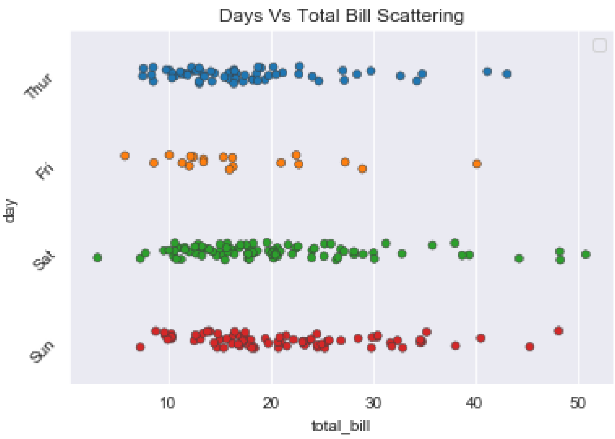
```
Out[156]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b6156ec8>
```



```
In [163]: # horizontal strip plot
          sns.stripplot(x='total_bill',y='day',data=tips,linewidth=0.5)
          plt.xlabel='Total'
          plt.ylabel='Days'
          plt.title("Days Vs Total Bill Scattering")
          plt.yticks(rotation=45)
          plt.legend()
```

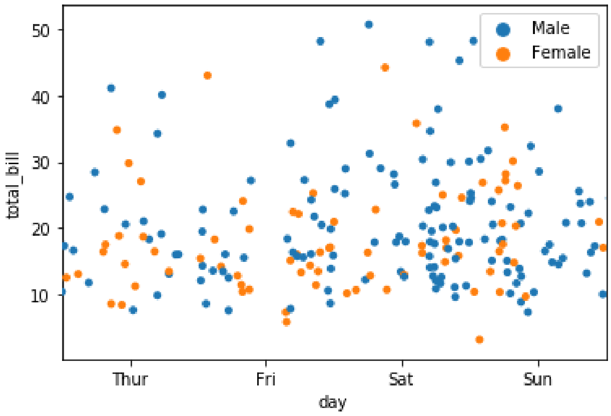
```
No handles with labels found to put in legend.
```

```
Out[163]: <matplotlib.legend.Legend at 0x1e1b6339508>
```



In [11]:
```
# stripplot with hue
sns.stripplot(x='day',y='total_bill',data=tips,hue='sex',jitter='0.9')
plt.legend(loc='upper right')
```

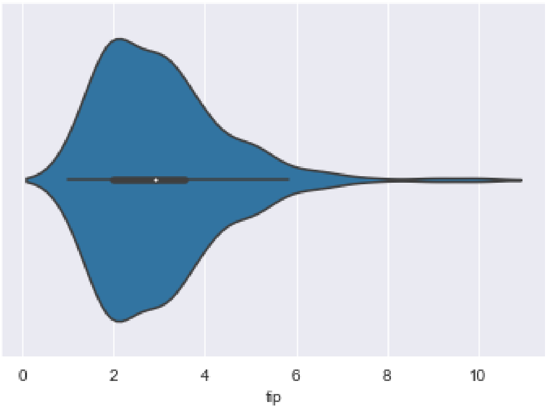
Out[11]:
```
<matplotlib.legend.Legend at 0x1e09d06dec8>
```



Violin Plot

In [173]:
```
sns.violinplot(x=tips['tip'])
```

Out[173]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b8faaac8>
```



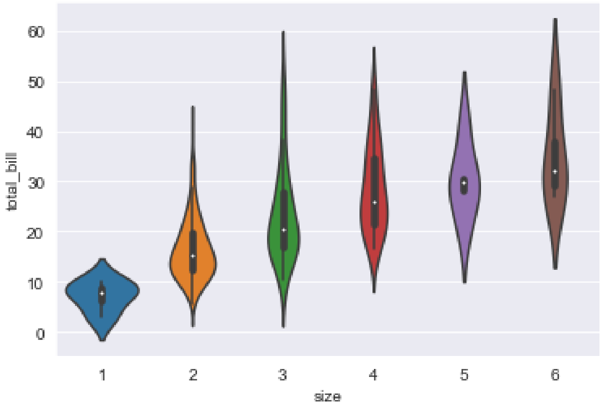
In [172]:
```
tips.head()
```

Out[172]:

	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

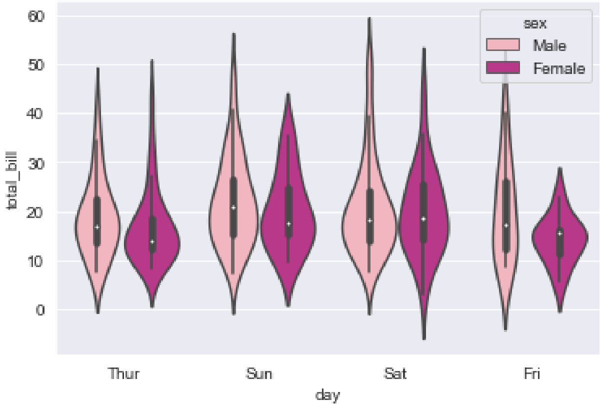
In [174]:
```
sns.violinplot(x='size',y='total_bill',data=tips)
```

Out[174]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b903c288>
```



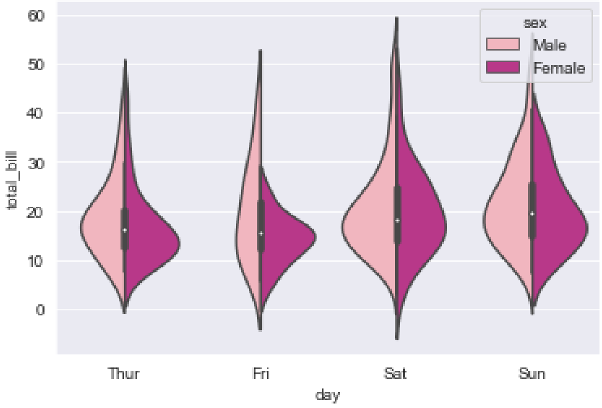
In [180]:
```
sns.violinplot(x='day',y='total_bill',data=tips,order=['Thur','Sun','Sat','Fri'],hue='sex',palette='
```

Out[180]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b9235cc8>
```



In [181]:
```
sns.violinplot(x='day',y='total_bill',data=tips,hue='sex',palette='RdPu',split=True)
```

Out[181]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b91ef808>
```

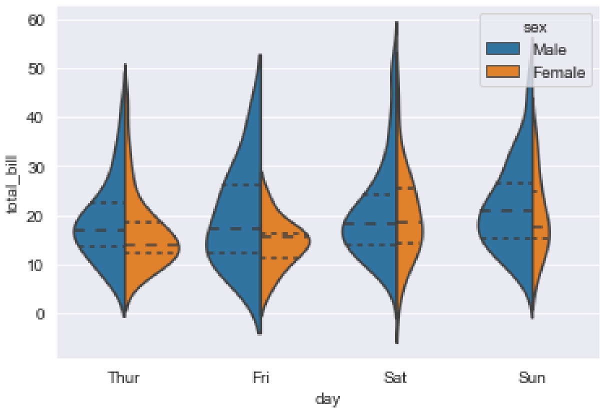


In [186]:

```
sns.violinplot(x='day',y='total_bill',hue='sex',inner='quartile',data=tips,scale='count',split=True)
```

Out[186]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b8c35308>
```

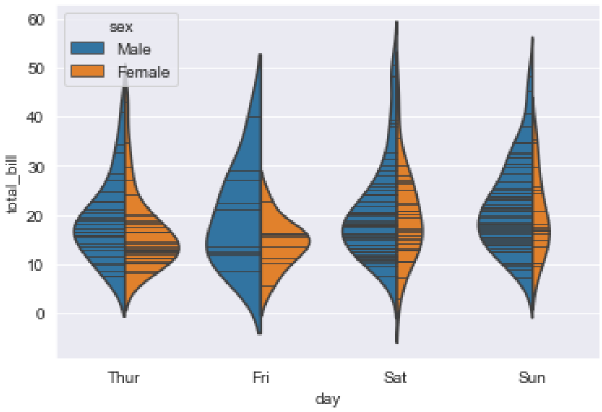


In [187]:

```
sns.violinplot(x='day',y='total_bill',hue='sex',inner='stick',data=tips,scale='count',split=True)
```

Out[187]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b8e06288>
```



KDE - Kernel Density Estimation

In [188]:

```
mean=[0,0]
cov=[[0.2,0],[0,3]]
```

In [199]:

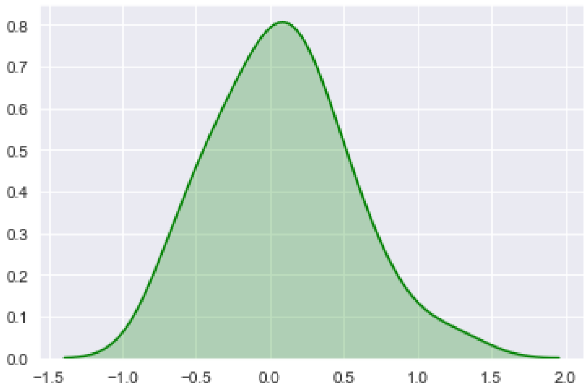
```
x_axis,y_axis=np.random.multivariate_normal(mean,cov,size=40).T
```

In [201]:

```
sns.kdeplot(x_axis,color='green',shade=True)
```

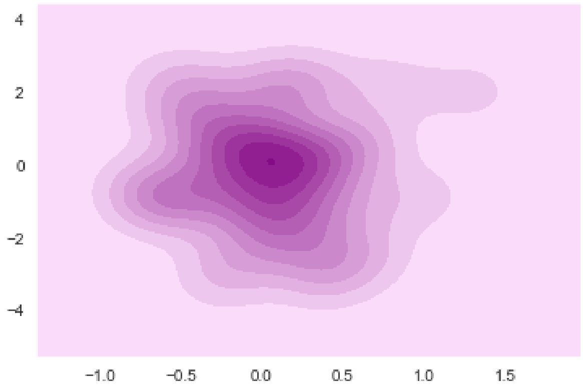
Out[201]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1b6f8c788>
```



```
In [205]: sns.kdeplot(x_axis,y_axis,shade=True,color='purple')
```

```
Out[205]: <matplotlib.axes._subplots.AxesSubplot at 0x1e1b8f29fc8>
```



```
In [ ]:
```