In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
%matplotlib inline
```

In [3]:
```
databases=sns.get_dataset_names()
databases
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\utils.py:384: UserWarning: No parser was explicitly specifi
ed, so I'm using the best available HTML parser for this system ("lxml"). This 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:\ProgramData\Anaconda3\lib\site-packages\seabor
n\utils.py. To get rid of this warning, pass the additional argument 'features="lxml"' to the BeautifulSoup co
nstructor.

  gh_list = BeautifulSoup(http)
```

Out[3]:
```
['anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'exercise',
 'flights',
 'fmri',
 'gammas',
 'geyser',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'tips',
 'titanic']
```

In [4]:
```
tips=sns.load_dataset('tips')
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.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: 7.3 KB
```

In [7]:
```
cols=list(tips['day'].unique())
print(cols)
tips[tips['day']=='Sun'].describe()
```

```
['Sun', 'Sat', 'Thur', 'Fri']
```

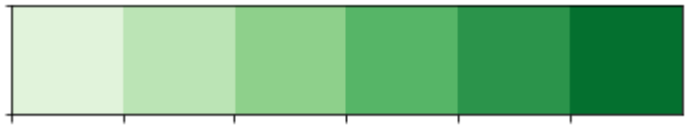
Out[7]:

	total_bill	tip	size
count	76.000000	76.000000	76.000000
mean	21.410000	3.255132	2.842105
std	8.832122	1.234880	1.007341
min	7.250000	1.010000	2.000000
25%	14.987500	2.037500	2.000000
50%	19.630000	3.150000	2.000000
75%	25.597500	4.000000	4.000000
max	48.170000	6.500000	6.000000

In [8]:
```
tips['day'].head()
```

Out[8]:
```
0    Sun
1    Sun
2    Sun
3    Sun
4    Sun
Name: day, dtype: category
Categories (4, object): [Thur, Fri, Sat, Sun]
```

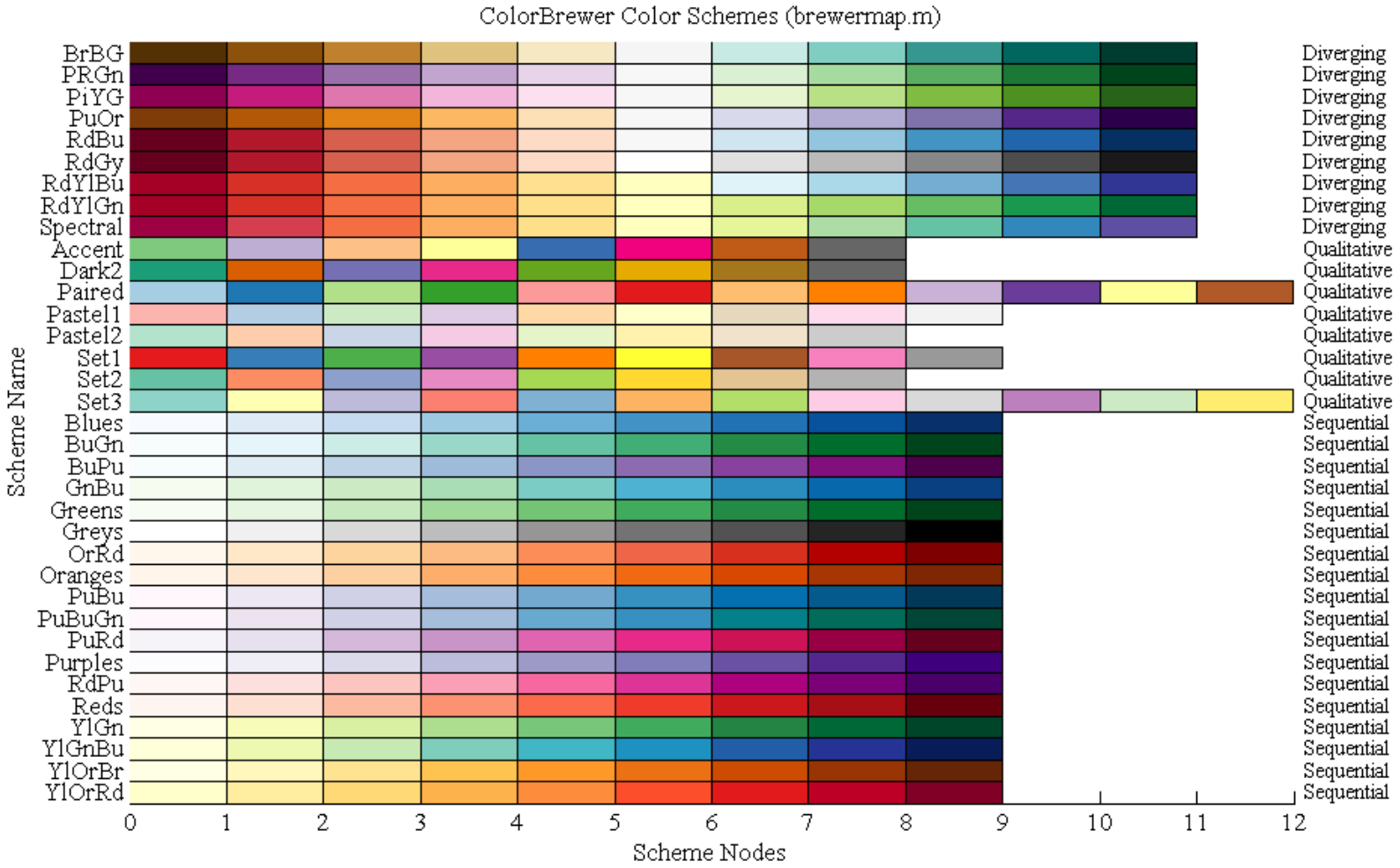
In [9]:
```
current_palette = sns.color_palette()
sns.palplot(sns.color_palette("Greens"))
plt.show()
```



In [10]:
```
current_palette = sns.color_palette()
sns.palplot(sns.color_palette("winter_r"))
plt.show()
```

In [11]:
```
current_palette = sns.color_palette()
sns.palplot(sns.color_palette("autumn"))
plt.show()
```

Other Color Palettes



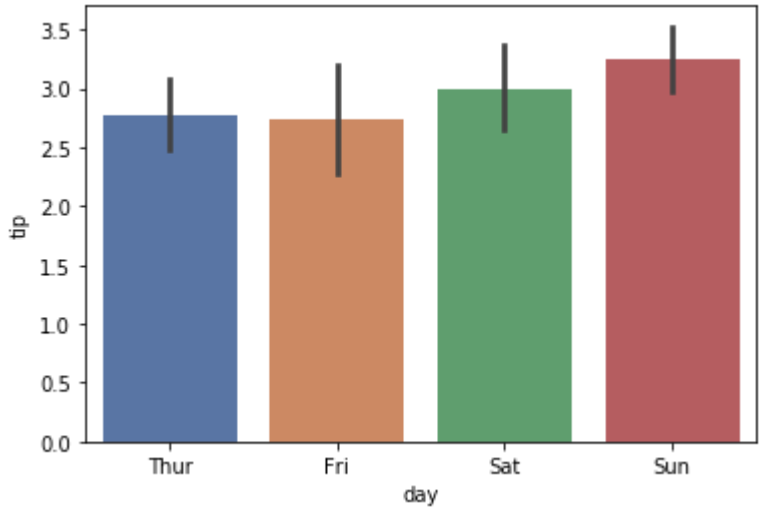
Bar Plots

The barplot() shows the relation between a categorical variable and a continuous variable. The data is represented in rectangular bars where the length the bar represents the proportion of the data in that category. Bar plot represents the estimate of central tendency

In [12]:
```
sns.barplot(x='day',y='tip', data=tips,palette='deep')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

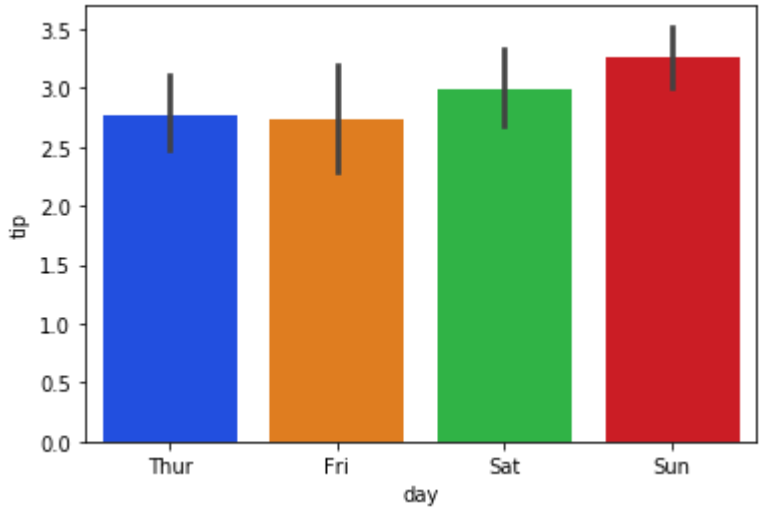
Out[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x299693e4b08>
```



In [13]:
```
sns.barplot(x='day',y='tip', data=tips,palette='bright')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

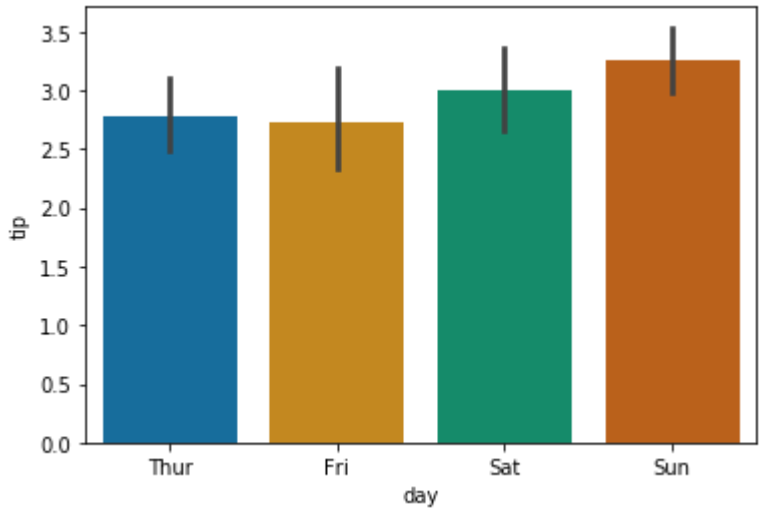
Out[13]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29969465f48>
```



In [14]:
```
sns.barplot(x='day',y='tip', data=tips,palette='colorblind')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

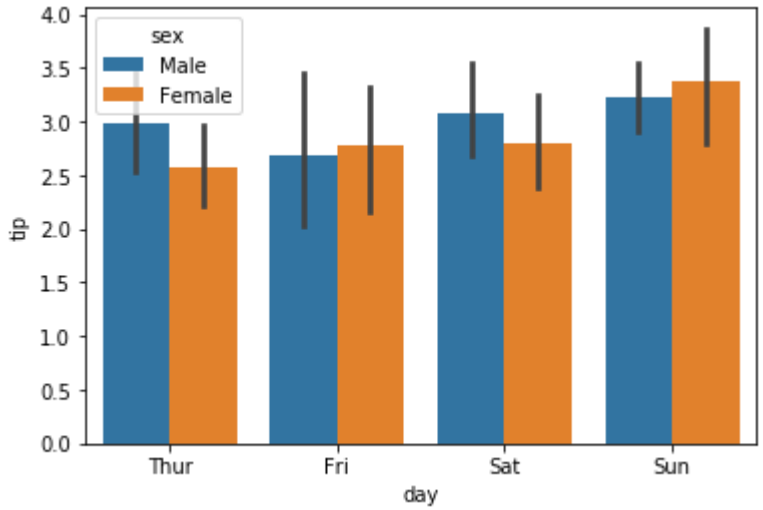
Out[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x299694d9048>
```



In [15]:
```
sns.barplot(x='day',y='tip', data=tips,hue='sex')
```

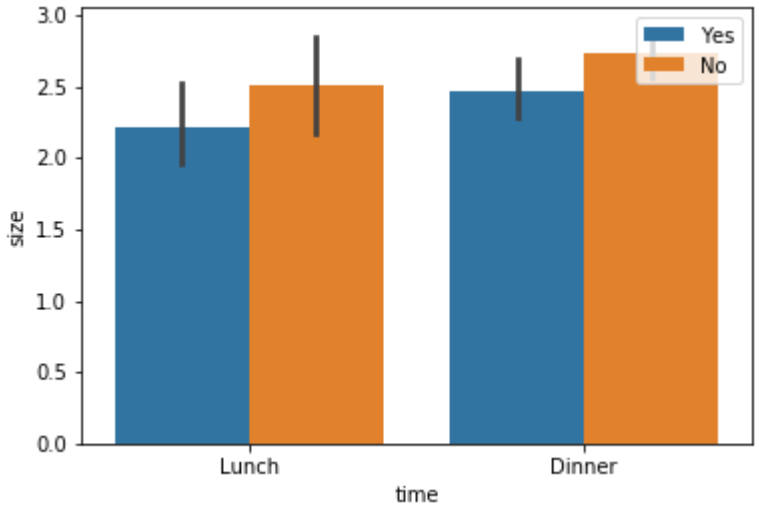
```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this
will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2996927b248>
```



In [16]:
```
graph=sns.barplot(x='time',y='size', data=tips,hue='smoker')
plt.legend(loc='upper right')
plt.show()
```

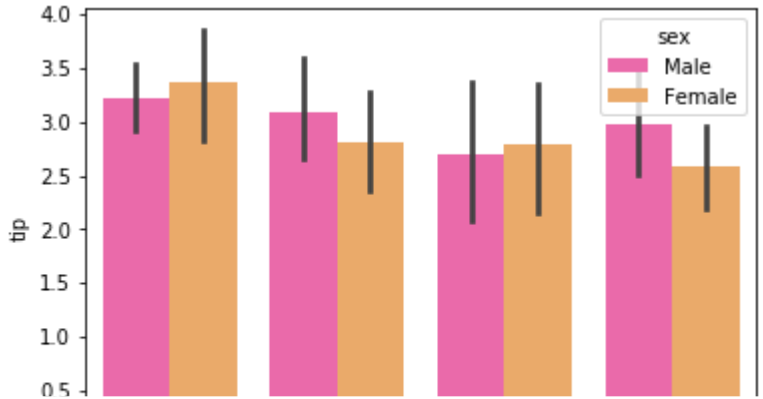
```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this
will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



In [17]:
```
sns.barplot(x='day',y='tip', data=tips,hue='sex',palette='spring',order=['Sun','Sat','Fri','Thur'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing 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[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29968ad07c8>
```

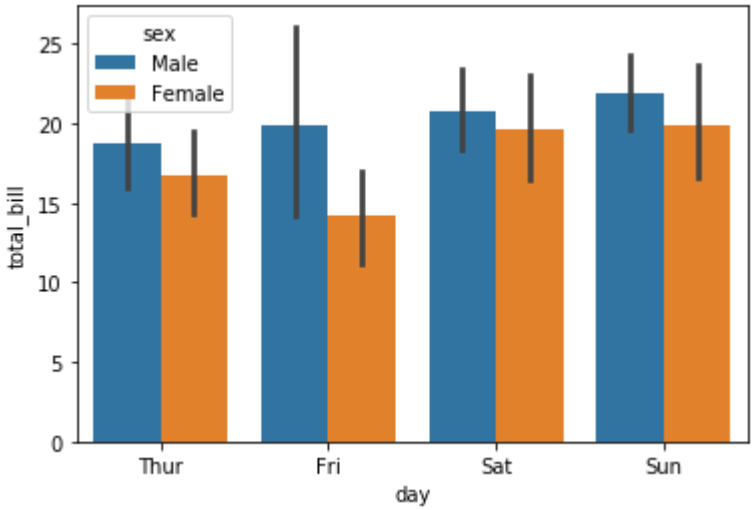


Mean and Median Bar Plots

In [18]:
```
from numpy import mean
sns.barplot(x='day',y='total_bill', data=tips,hue='sex',estimator=mean)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

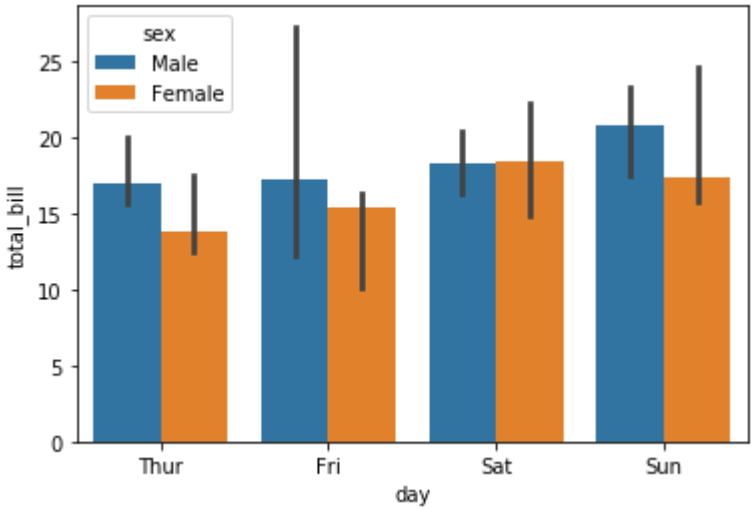
Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29968a6c3c8>
```



In [19]:
```
from numpy import median
sns.barplot(x='day',y='total_bill', data=tips,hue='sex',estimator=median)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

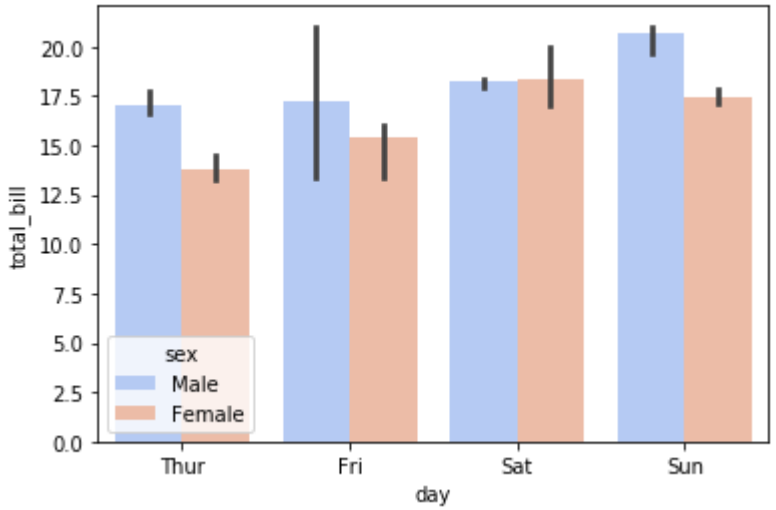
Out[19]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29969522cc8>
```



In [20]:
```
from numpy import median
sns.barplot(x='day',y='total_bill', data=tips,hue='sex',estimator=median,palette='coolwarm',ci=33)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

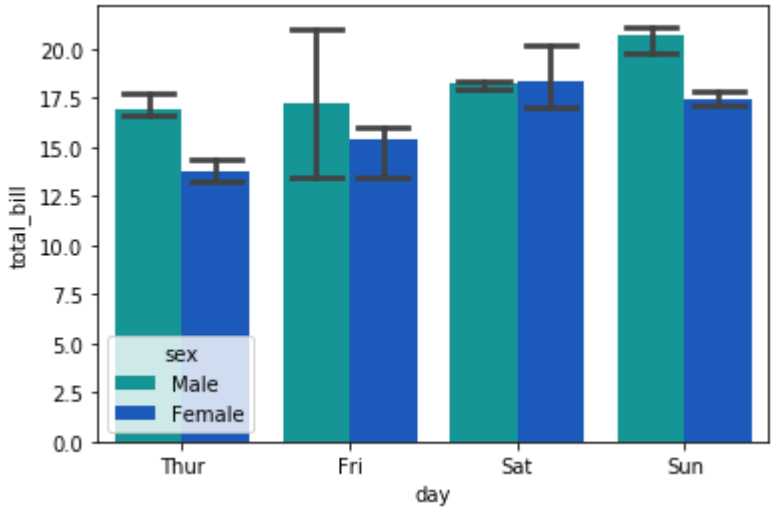
Out[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x299695b88c8>
```



In [21]:
```
from numpy import median
sns.barplot(x='day',y='total_bill', data=tips,hue='sex',estimator=median,palette='winter_r',ci=33,capsize=0.3)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

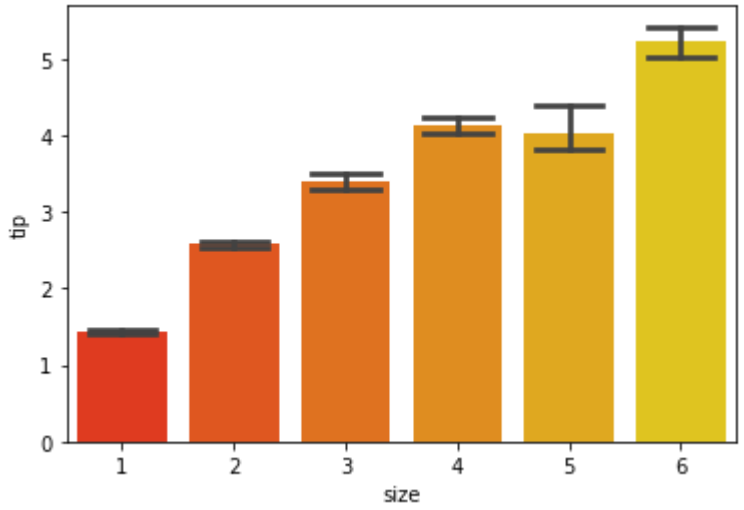
Out[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29969656388>
```



In [22]:
```
sns.barplot(x='size',y='tip', data=tips,palette='autumn',ci=33,capsize=0.6)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29969718148>
```



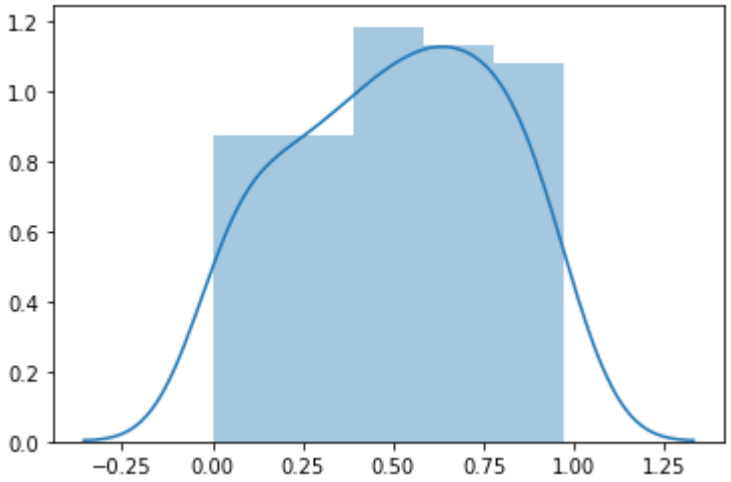
Dist Plots

Function distplot() provides the most convenient way to take a quick look at univariate distribution. This function will plot a histogram that fits the kernel density estimation of the data.

In [23]:
```
data=np.random.rand(100)
sns.distplot(data)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

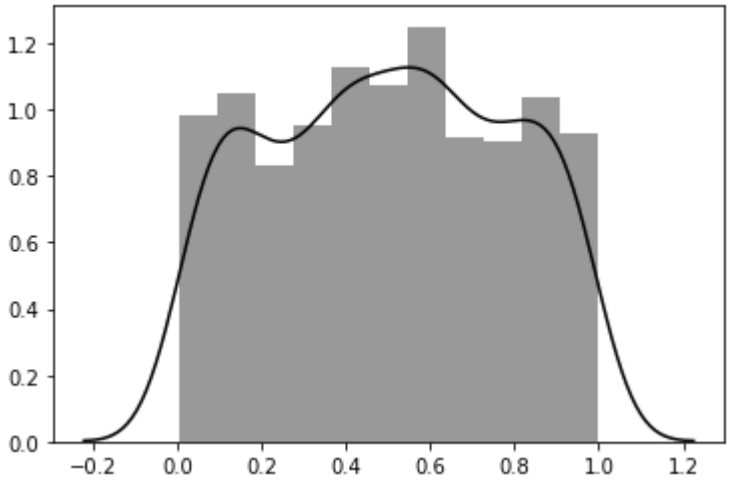
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x299697ac748>



In [24]:
```
data=np.random.rand(1000)
sns.distplot(data,color='k')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x29969833d48>

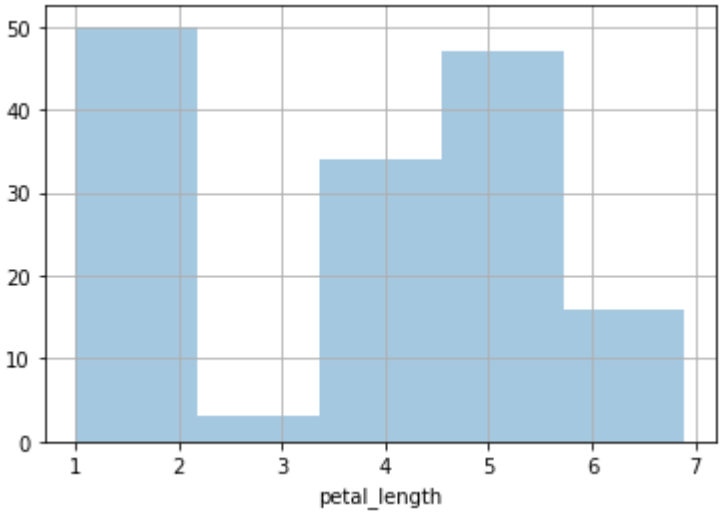


Histograms

Histograms represent the data distribution by forming bins along the range of the data and then drawing bars to show the number of observations that fall in each bin.

In [25]:
```
# kde=False will remove the representation of kernel estimation plot.
data=sns.load_dataset('iris')
#data.info()
sns.distplot(data['petal_length'],kde=False)
plt.grid()
```

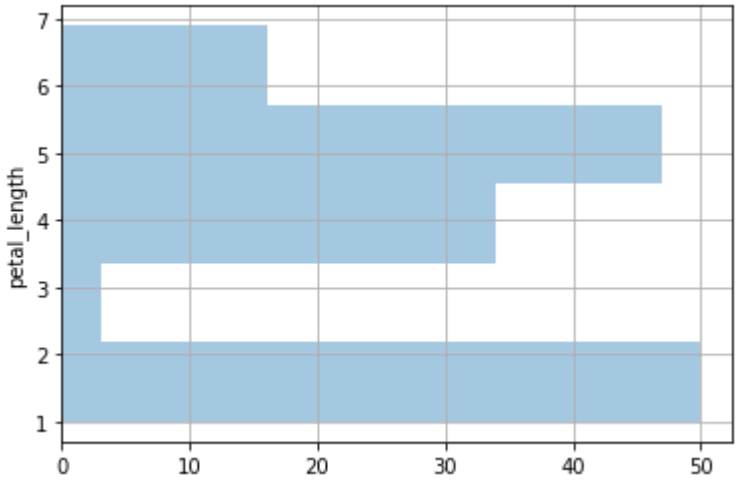
```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this
will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



Kernel Density Estimation (KDE) is a way to estimate the probability density function of a continuous random variable. It is used for non-parametric analysis. Setting the hist flag to False in distplot will yield the kernel density estimation plot.

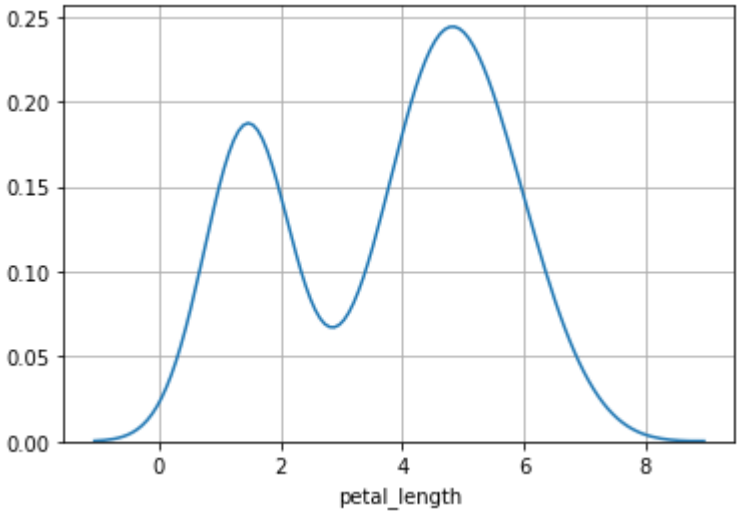
In [26]:
```
data=sns.load_dataset('iris')
#data.info()
sns.distplot(data['petal_length'],kde=False,vertical=True)
plt.grid()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



In [27]:
```
data=sns.load_dataset('iris')
#data.info()
sns.distplot(data['petal_length'],hist=False)
plt.grid()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



Bivariate Distribution

Bivariate Distribution is used to determine the relation between two variables. This mainly deals with relationship between two variables and how one variable is behaving with respect to the other. The best way to analyze Bivariate Distribution in seaborn is by using the jointplot() function.

1.Scatter Plot

visualize the distribution where each observation is represented in two-dimensional plot via x and y axis.

In [28]:
```
df = sns.load_dataset('iris')
sns.jointplot(x='petal_length',y='petal_width',data=df)
plt.show()
# A trend in the plot says that positive correlation exists between the variables under study.
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequenc
e for multidimensional indexing 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 differe
nt result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```