

```
In [1]: #Installation  
#pip install seaborn
```

```
In [2]: #Dependence= pandas,numpy,matplotlib,scipy,statsmodels
```

```
In [3]: import numpy as np  
import matplotlib as plt  
import seaborn as sns  
import pandas as pd  
%matplotlib inline
```

Barplot

A bar plot represents an estimate of central tendency for a numeric variable with the height of each rectangle and provides some indication of the uncertainty around that estimate using error bars. Bar plots include 0 in the quantitative axis range, and they are a good choice when 0 is a meaningful value for the quantitative variable, and you want to make comparisons against it. Parameters: x, y, hue : names of variables in data or vector data, optional Inputs for plotting long-form data. See examples for interpretation. data : DataFrame, array, or list of arrays, optional Dataset for plotting. If x and y are absent, this is interpreted as wide-form. Otherwise it is expected to be long-form. order, hue_order : lists of strings, optional Order to plot the categorical levels in, otherwise the levels are inferred from the data objects. estimator : callable that maps vector -> scalar, optional Statistical function to estimate within each categorical bin. ci : float or "sd" or None, optional Size of confidence intervals to draw around estimated values. If "sd", skip bootstrapping and draw the standard deviation of the observations. If None, no bootstrapping will be performed, and error bars will not be drawn.

n_boot : int, optional Number of bootstrap iterations to use when computing confidence intervals. units : name of variable in data or vector data, optional Identifier of sampling units, which will be used to perform a multilevel bootstrap and account for repeated measures design. orient : "v" | "h", optional Orientation of the plot (vertical or horizontal). This is usually inferred from the dtype of the input variables, but can be used to specify when the "categorical" variable is a numeric or when plotting wide-form data. color : matplotlib color, optional Color for all of the elements, or seed for a gradient palette. palette : palette name, list, or dict, optional Colors to use for the different levels of the hue variable. Should be something that can be interpreted by color_palette(), or a dictionary mapping hue levels to matplotlib colors. saturation : float, optional Proportion of the original saturation to draw colors at. Large patches often look better with slightly desaturated colors, but set this to 1 if you want the plot colors to perfectly match the input color spec. errcolor : matplotlib color Color for the lines that represent the confidence interval. errwidth : float, optional Thickness of error bar lines (and caps). capsize : float, optional Width of the "caps" on error bars. dodge : bool, optional When hue nesting is used, whether elements should be shifted along the categorical axis. ax : matplotlib Axes, optional Axes object to draw the plot onto, otherwise uses the current Axes. kwargs : key, value mappings Other keyword arguments are passed through to plt.bar at draw time.

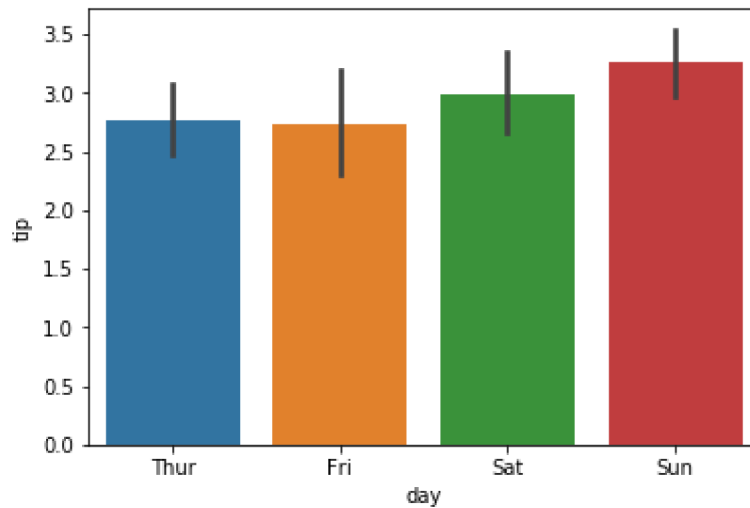
```
In [4]: tips=sns.load_dataset('tips')
```

```
In [5]: print(tips.columns)
```

```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtype='object')
```

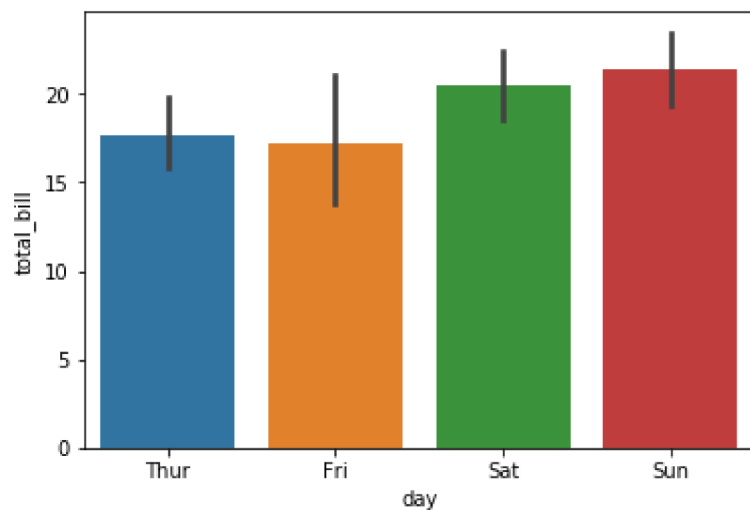
```
In [6]: sns.barplot(x="day",y="tip",data=tips)
```

```
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x1b813c30278>
```



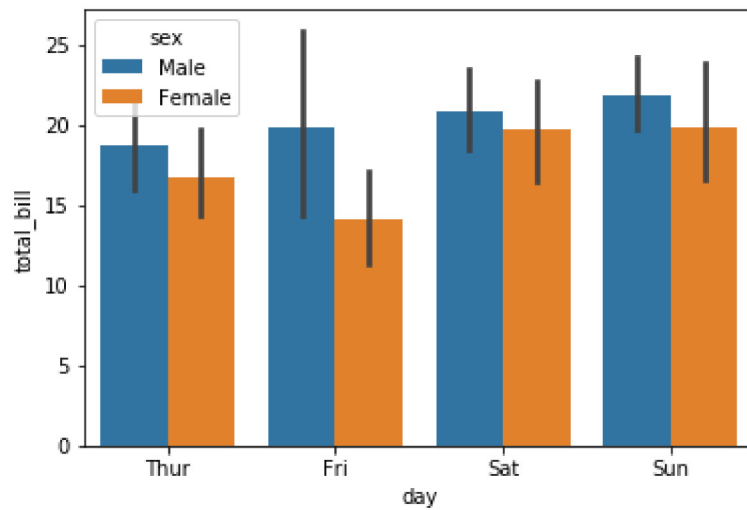
```
In [7]: sns.barplot(x="day",y="total_bill",data=tips)
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x1b8177f1860>
```



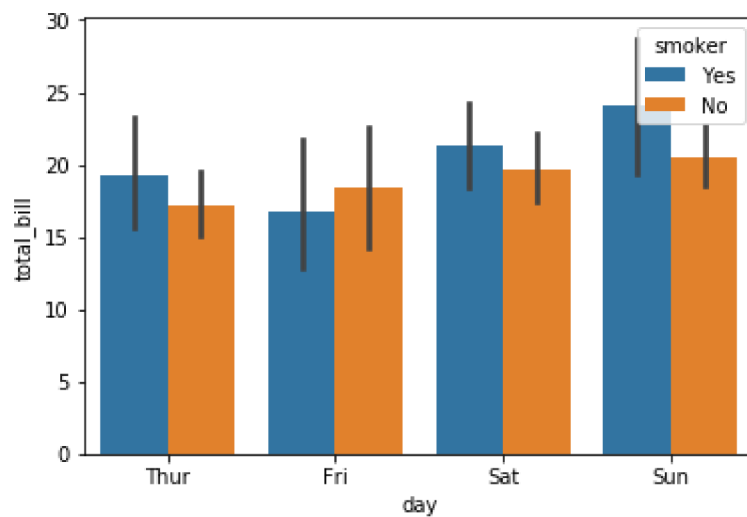
```
In [8]: sns.barplot(x="day",y="total_bill",hue="sex",data=tips)
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1b818826438>
```



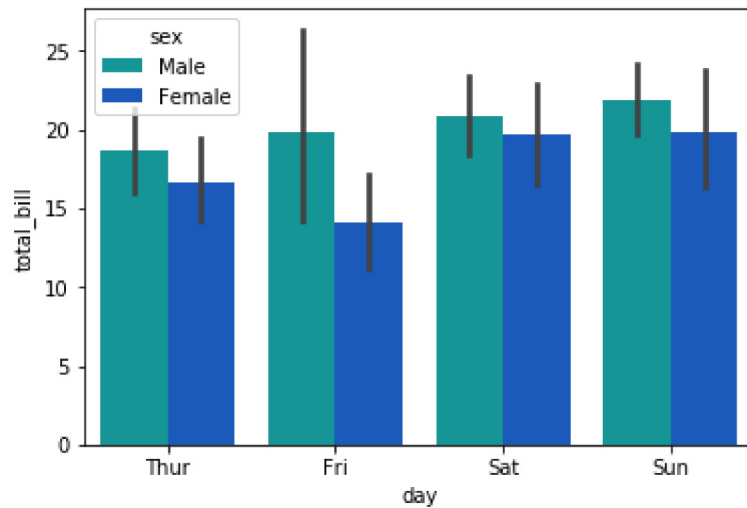
```
In [9]: sns.barplot(x="day",y="total_bill",hue="smoker",data=tips)
```

```
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1b8188b9b70>
```



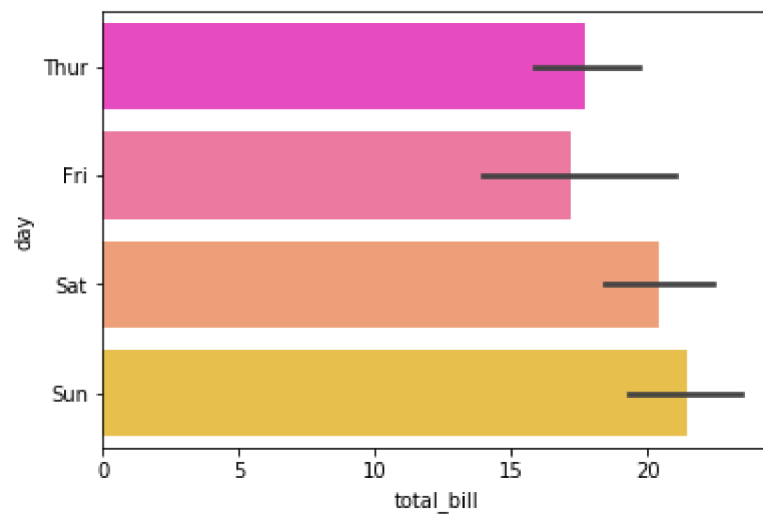
```
In [10]: #Change the color  
sns.barplot(x="day",y="total_bill",hue="sex",data=tips,palette="winter_r")
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1b81892b358>



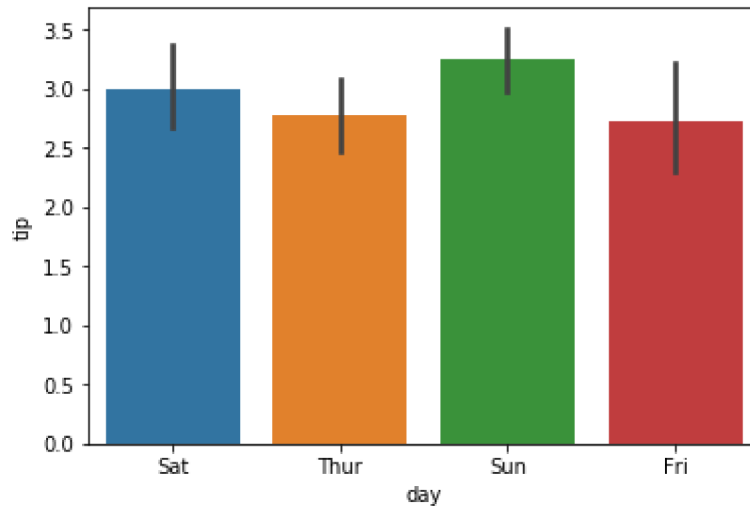
```
In [11]: sns.barplot(x="total_bill",y="day",data=tips,palette="spring")
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1b8189a2550>



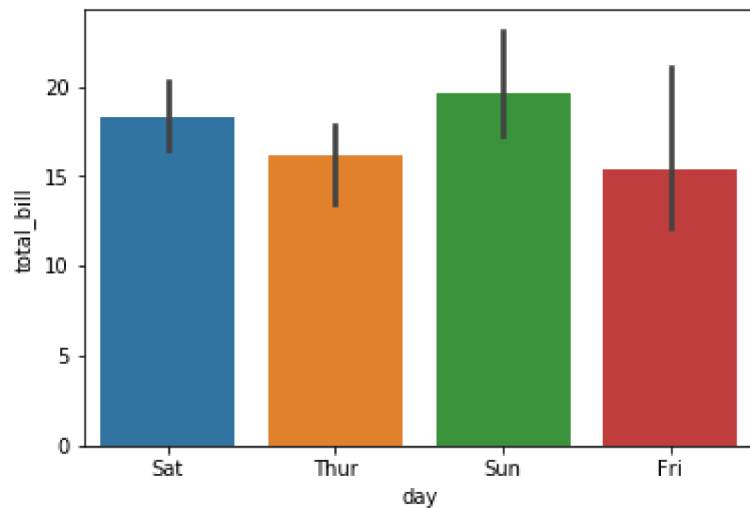
```
In [12]: sns.barplot(x="day",y="tip",data=tips,order=['Sat','Thur','Sun','Fri'])
```

```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x1b818a0f5c0>
```



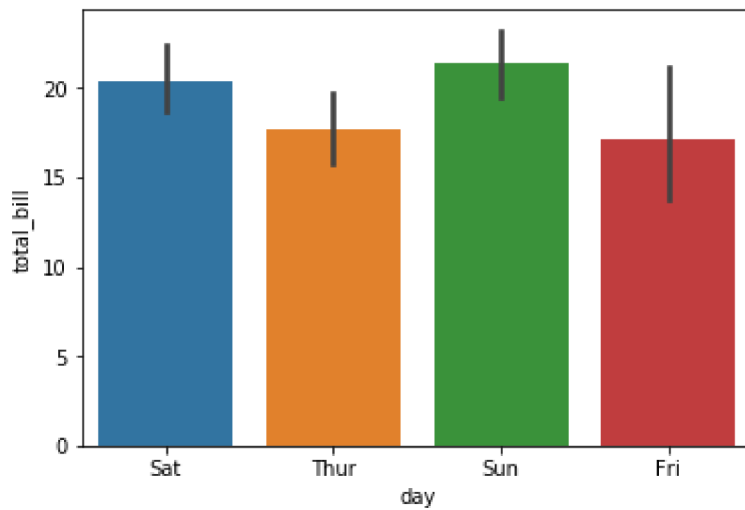
```
In [13]: from numpy import median
sns.barplot(x="day",y="total_bill",data=tips,estimator=median,order=['Sat','Thur','Sun','Fri'])
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1b818a58b38>
```



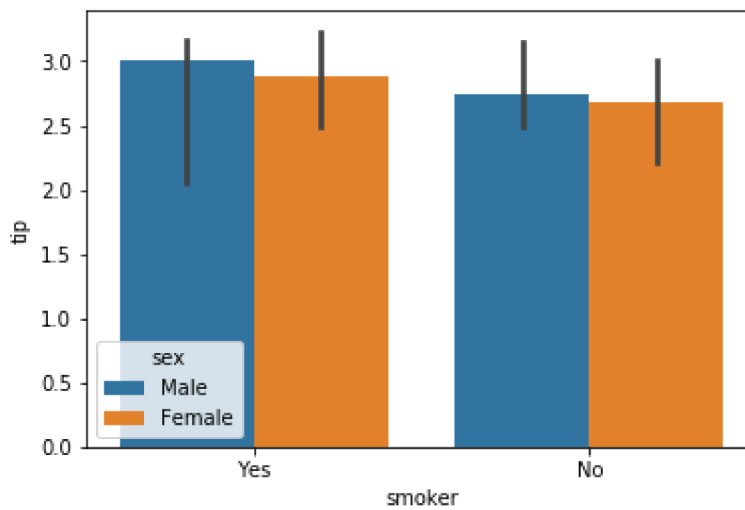
```
In [14]: sns.barplot(x="day",y="total_bill",data=tips,order=['Sat','Thur','Sun','Fri'])
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1b818ae8518>
```



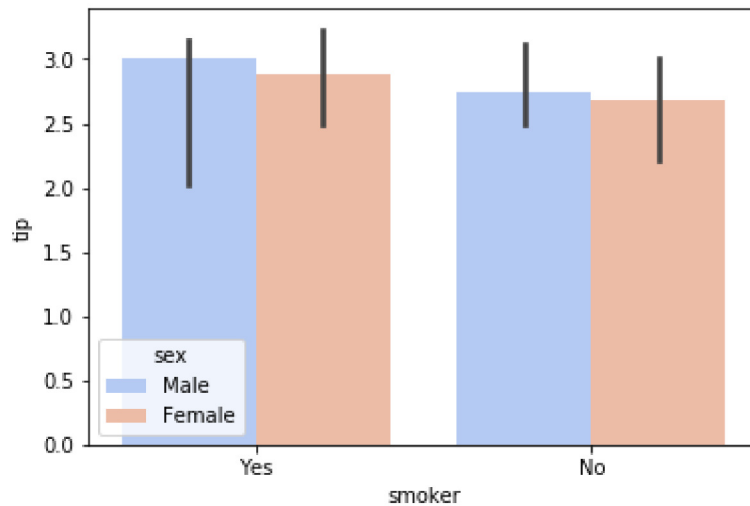
```
In [15]: sns.barplot(x="smoker",y="tip",data=tips,estimator=median,hue="sex")
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1b818b2fa20>
```



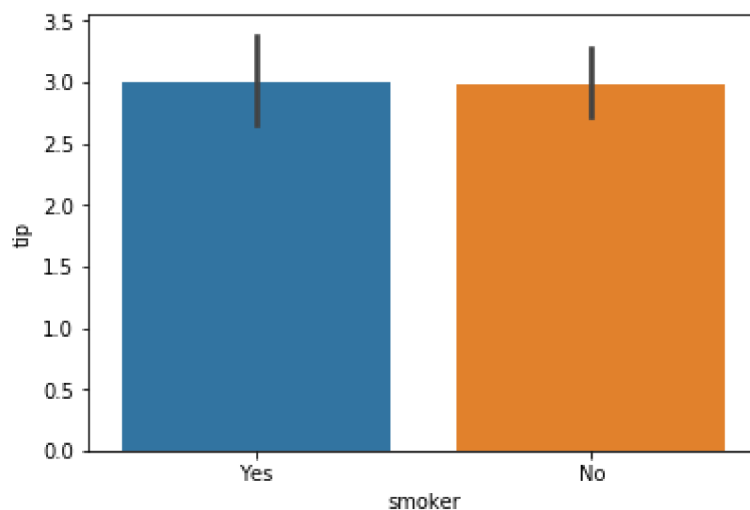
```
In [16]: sns.barplot(x="smoker",y="tip",data=tips,estimator=median,hue="sex",palette="coolwarm")
```

```
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1b818b8cac8>
```



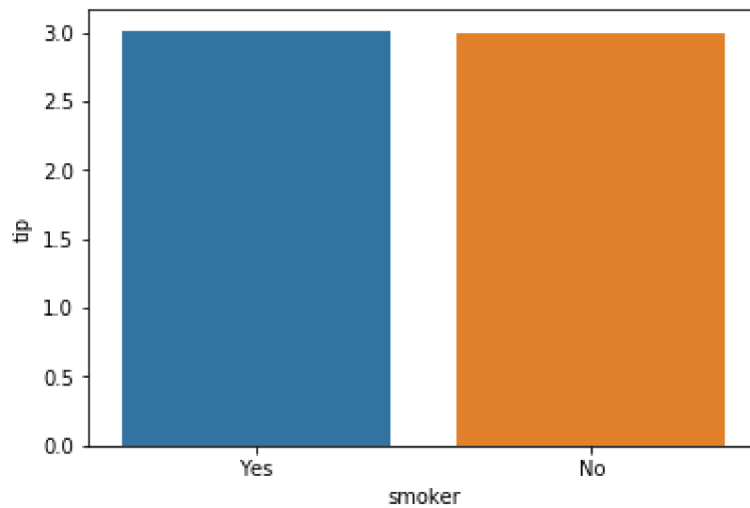
```
In [18]: #Black Bar is known as Error bar,, where we can set it's value using "ci" attribute  
sns.barplot(x="smoker",y="tip",data=tips,ci=99)
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1b8191ac320>
```



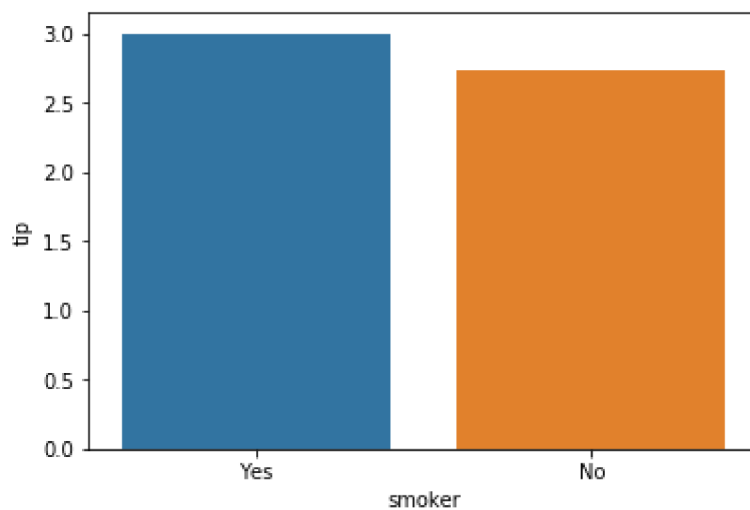
```
In [21]: sns.barplot(x="smoker",y="tip",data=tips,ci=0)
```

```
Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x1b81946aa20>
```



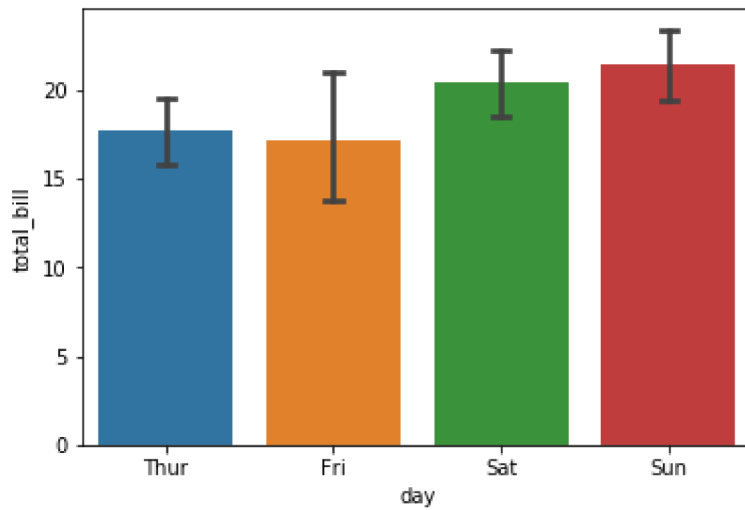
```
In [22]: sns.barplot(x="smoker",y="tip",data=tips,ci=0,estimator=median)
```

```
Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x1b81ae1d278>
```



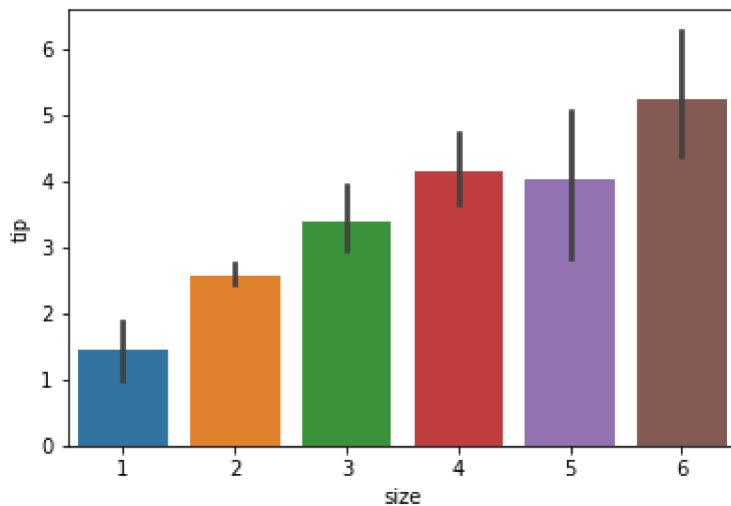

```
In [25]: #Two row at the end of the error bars are known as caps.Setting the cap size using "capsize" attribute.  
sns.barplot(x='day',y='total_bill',data=tips,capsize=0.2)
```

```
Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x1b81a9ec358>
```



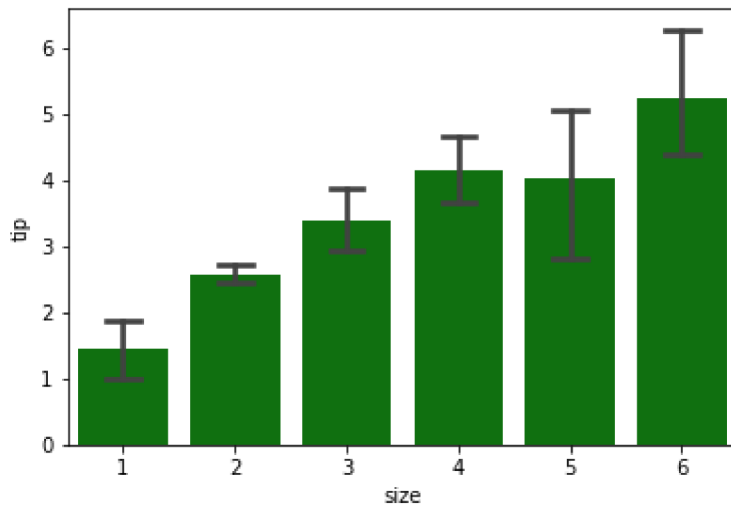
```
In [27]: sns.barplot(x="size",y="tip",data=tips)
```

```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x1b81abaedd8>
```



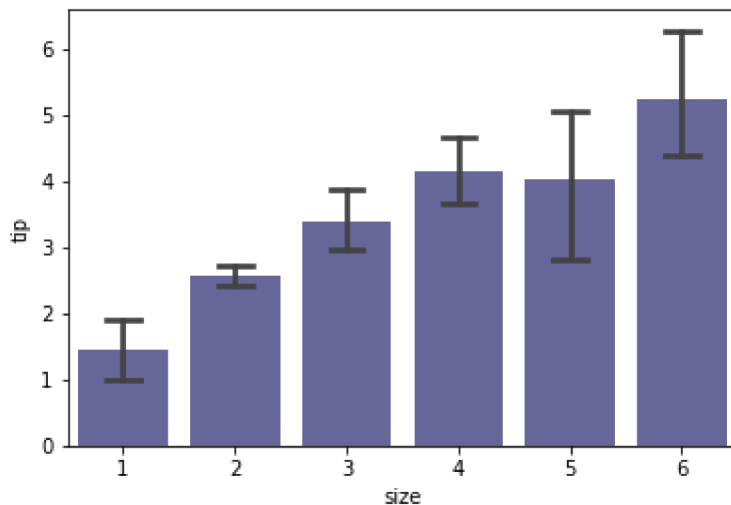
```
In [29]: # Passing "colour" attribute we can specify a same colour to all the bars  
sns.barplot(x="size",y="tip",data=tips,capsize=0.3,color="green")
```

```
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x1b81a8d58d0>
```



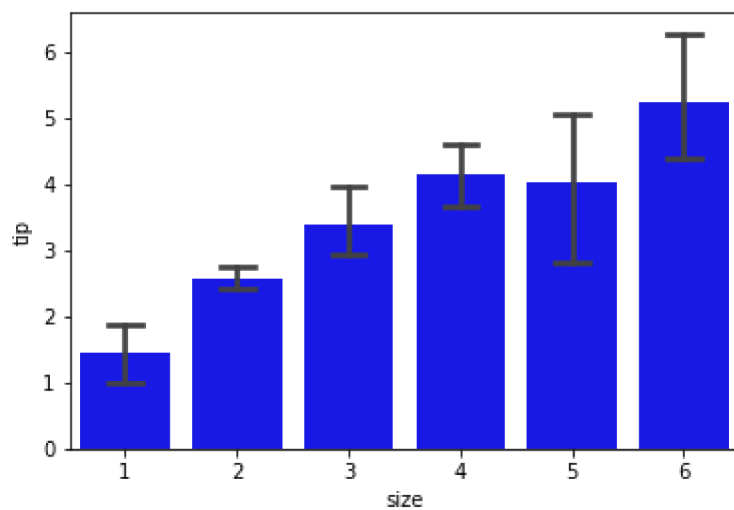
```
In [30]: # Adding Saturation  
sns.barplot(x="size",y="tip",data=tips,capsize=0.3,color="blue",saturation=0.2  
)
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x1b81aec31d0>
```



```
In [31]: sns.barplot(x="size",y="tip",data=tips,capsize=0.3,color="blue",saturation=0.8)
```

```
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1b81aad91d0>
```



```
In [ ]:
```