

```
In [4]: import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

load the dataset

```
In [5]: sns.get_dataset_names()
```

```
Out[5]: ['anagrams',
'anscombe',
'attention',
'brain_networks',
'car_crashes',
'diamonds',
'dots',
'exercise',
'flights',
'fmri',
'gammas',
'geyser',
'iris',
'mpg',
'penguins',
'planets',
'tips',
'titanic']
```

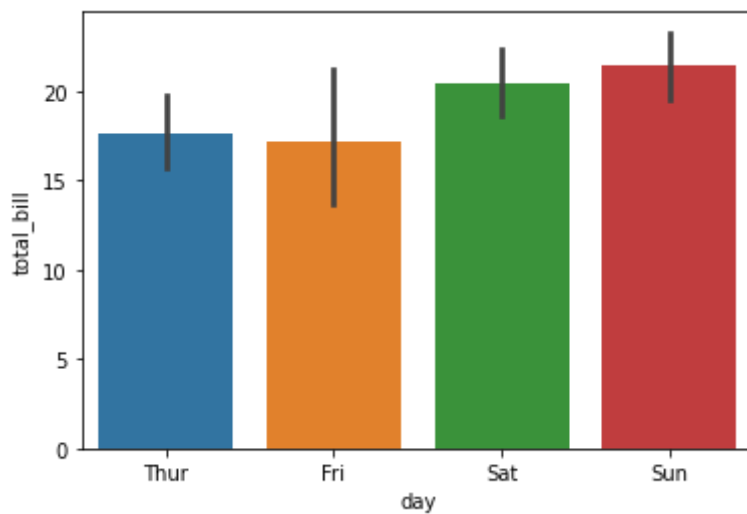
```
In [13]: tips = sns.load_dataset('tips')
tips.head()
```

```
Out[13]:
```

	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 [16]: sns.barplot(x = 'day', y = 'total_bill', data = tips)
```

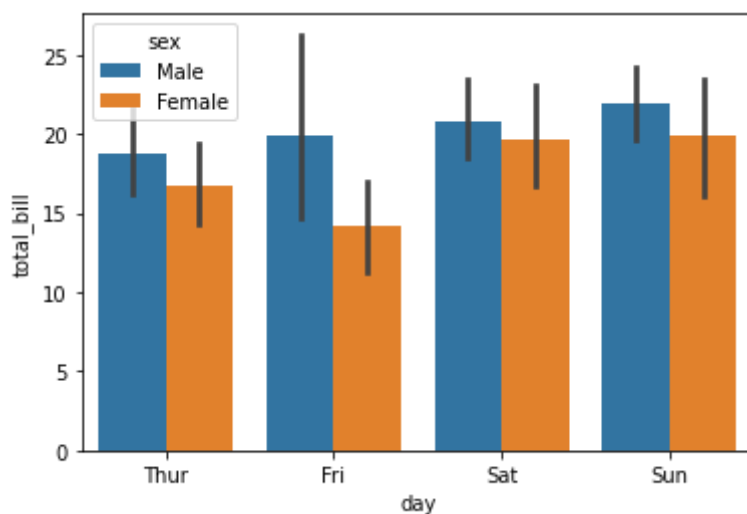
```
Out[16]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



Hue Attribute

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

```
Out[18]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



Palette Attribute

using the palette function to draw plots

Different type of palettes

1. deep
2. light
3. winter_r
4. spring

i'm going to demonstrate each sparately inshallah

```
In [24]: sns.color_palette('deep',10)
```

Out[24]:

```
In [35]: sns.palplot(sns.light_palette('green',10))
```



```
In [25]: sns.color_palette('winter_r',10)
```

Out[25]:

```
In [26]: sns.color_palette('spring',10)
```

Out[26]:

Creating your own custom palette

create palette as an array

```
In [48]: custom = ['red','blue','green']
```

set the palette as the current palette

```
In [49]: sns.set_palette(custom)
```

```
In [50]: sns.palplot(sns.color_palette())
```

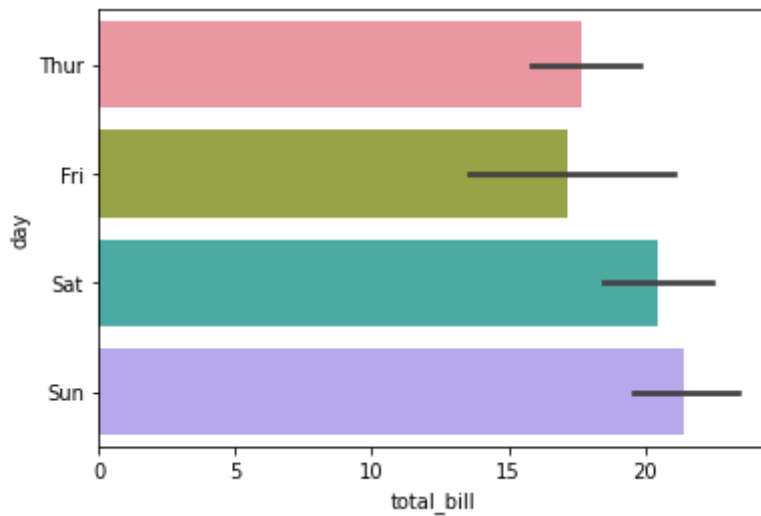


Horizontals Bar Plots

x = total_bill and y = day

```
In [51]: sns.barplot(x = 'total_bill', y = 'day', data = tips)
```

```
Out[51]: <AxesSubplot:xlabel='total_bill', ylabel='day'>
```

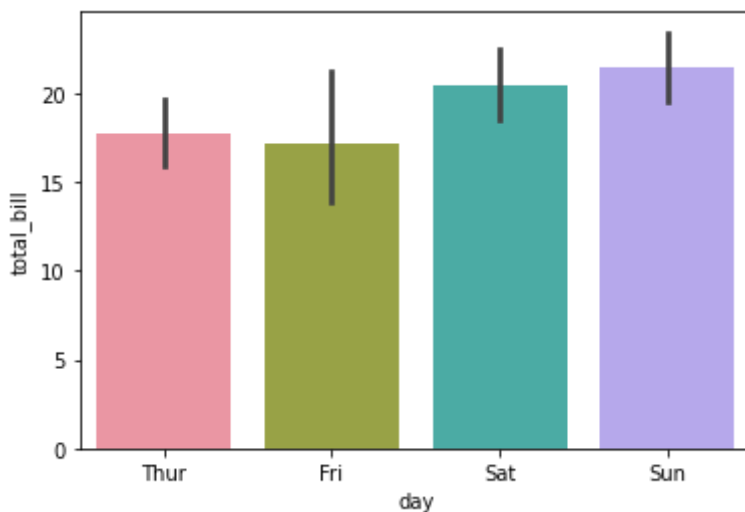


order Attribute

x = day and y = tips

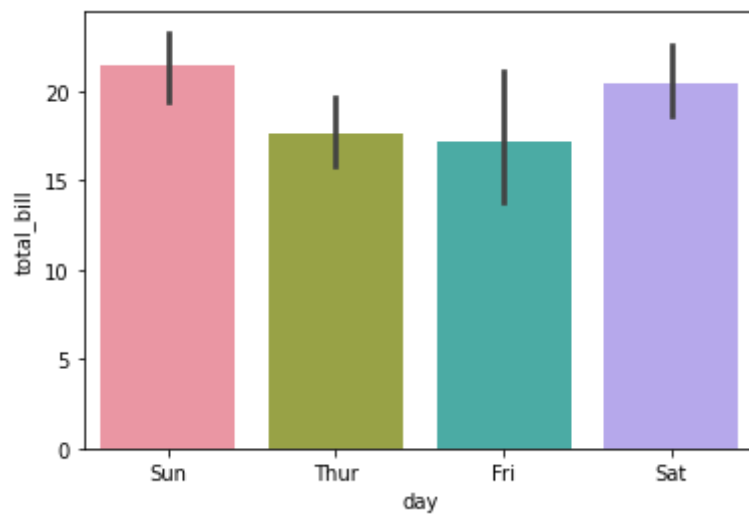
```
In [58]: sns.barplot(x = 'day', y = 'total_bill', data = tips)
```

```
Out[58]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



```
In [60]: sns.barplot(x = 'day', y = 'total_bill', order = ['Sun', 'Thur', 'Fri', 'Sat'], d
```

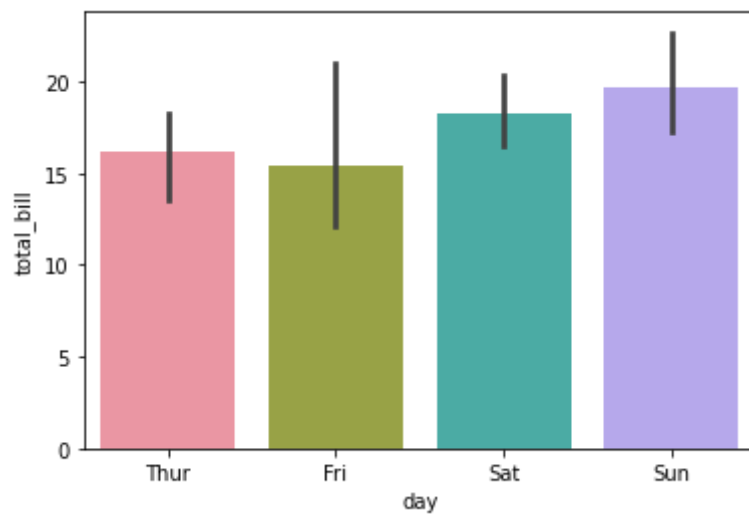
```
Out[60]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



part 2 of bar plotting in seaborn

```
In [61]: sns.barplot(x = 'day', y = 'total_bill', estimator = np.median, data = tips)
```

```
Out[61]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



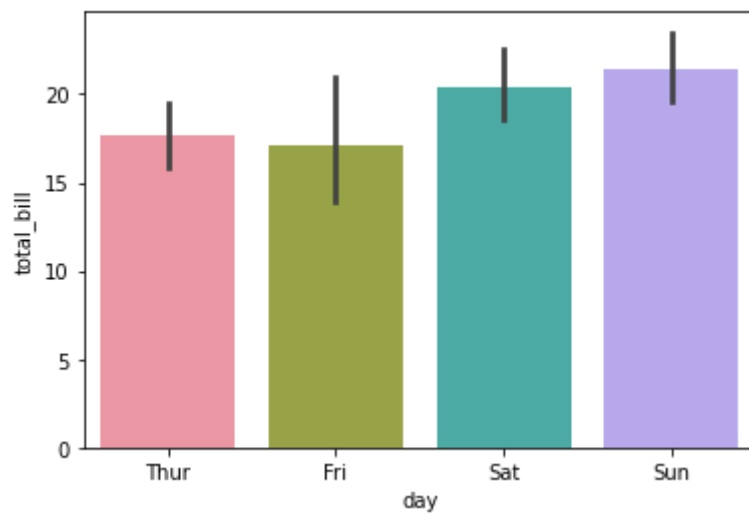
```
In [63]: tips = sns.load_dataset('tips')
tips.head()
```

```
Out[63]:
```

	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

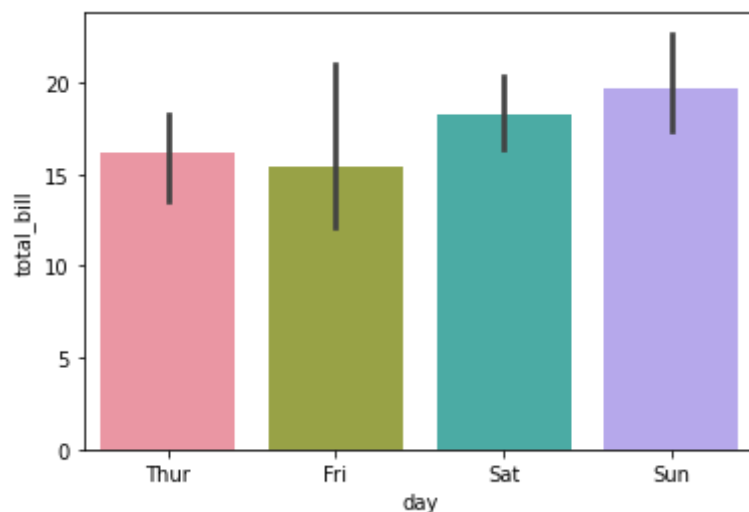
```
In [64]: sns.barplot(x = 'day', y = 'total_bill', data = tips)
```

```
Out[64]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



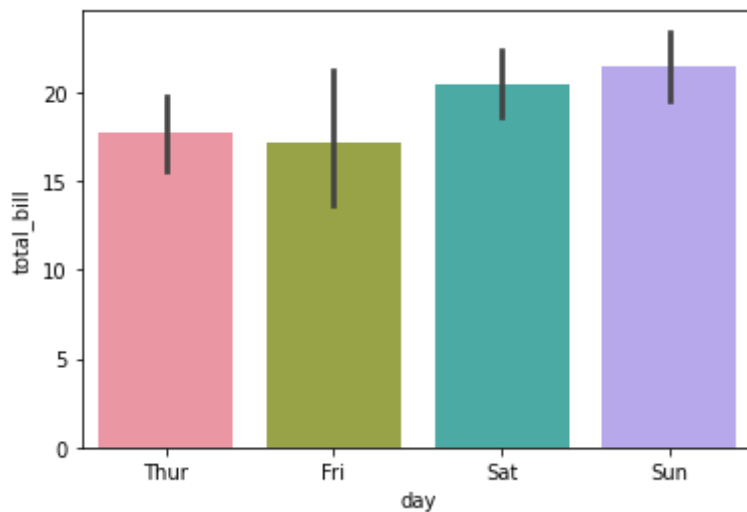
```
In [65]: sns.barplot(x = 'day', y = 'total_bill', estimator = np.median, data = tips)
```

```
Out[65]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



```
In [66]: sns.barplot(x = 'day', y = 'total_bill', data = tips)
```

```
Out[66]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



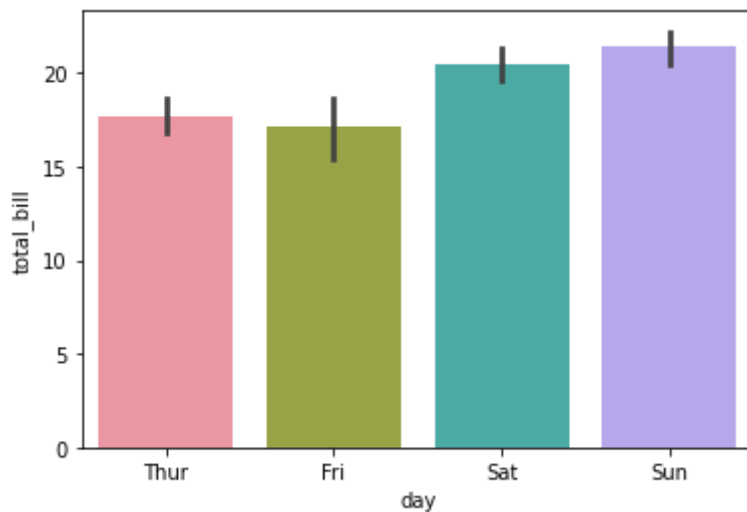
Confidence Intervals

the black bars are the confidence intervals

in short a confidence intervals is a range of values in which we fairly confident that the statistical measure lies in

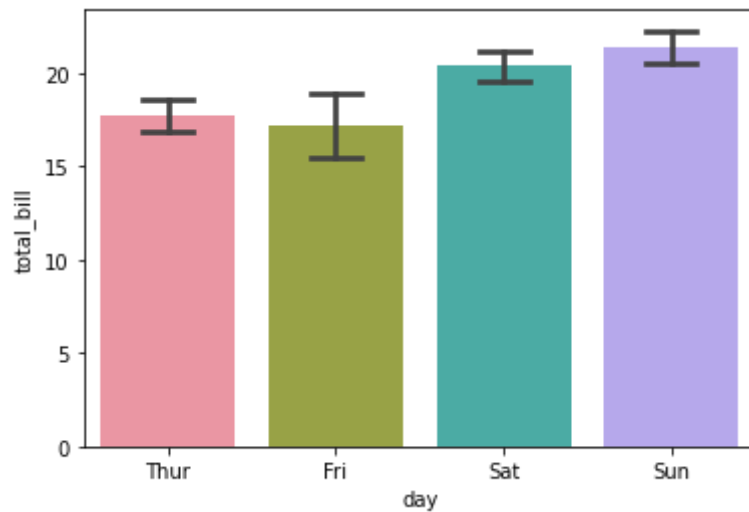
```
In [67]: sns.barplot(x = 'day', y = 'total_bill', ci = 60, data = tips)
```

```
Out[67]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



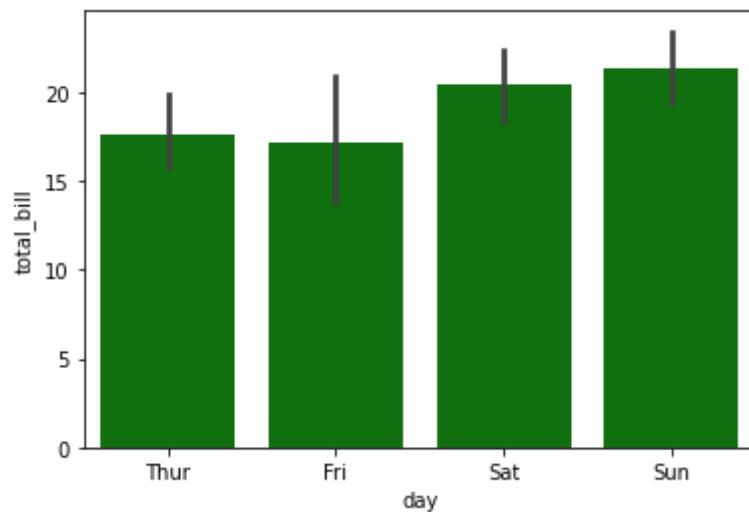
```
In [68]: sns.barplot(x = 'day', y = 'total_bill', ci = 60, capsize = 0.3, data = tips)
```

```
Out[68]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



```
In [69]: sns.barplot(x = 'day', y = 'total_bill', color = 'green', data = tips)
```

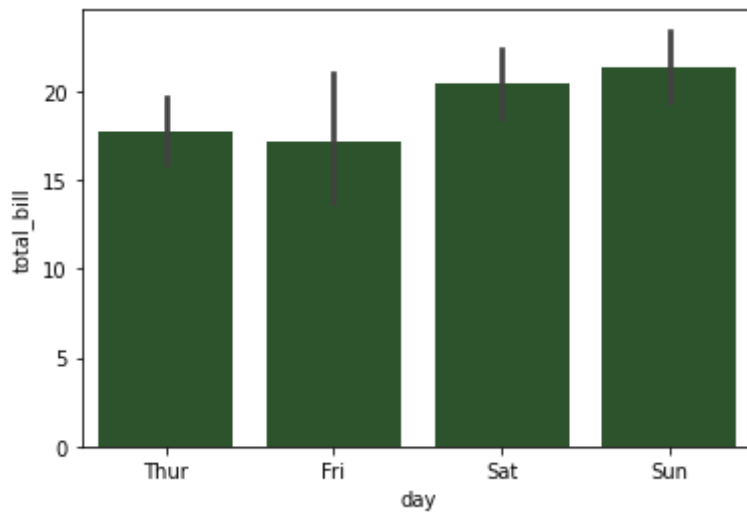
```
Out[69]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



Saturation Level

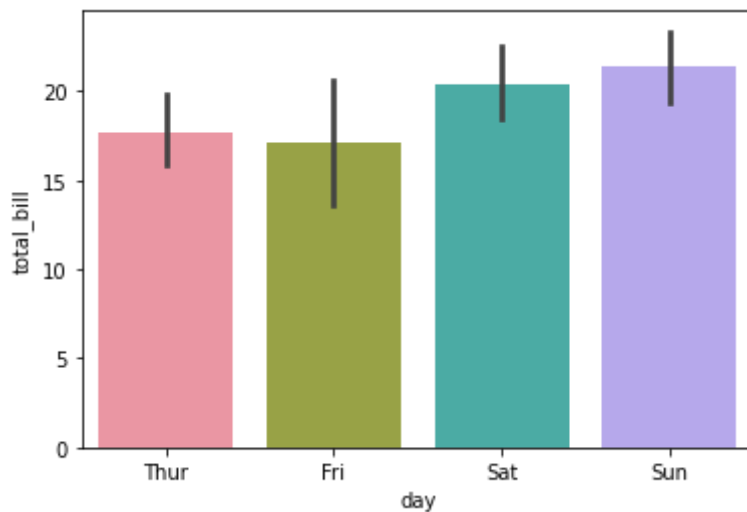

```
In [73]: sns.barplot(x = 'day', y = 'total_bill', color = 'green', saturation = 0.3, data = tips)
```

```
Out[73]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



Saving plots in file

```
In [74]: sns_plot = sns.barplot(x = 'day', y = 'total_bill', data = tips)
```



```
In [75]: fig = sns_plot.get_figure()
```

```
In [78]: fig.savefig(fname = 'plot.png')
```

```
In [ ]:
```

In []: