

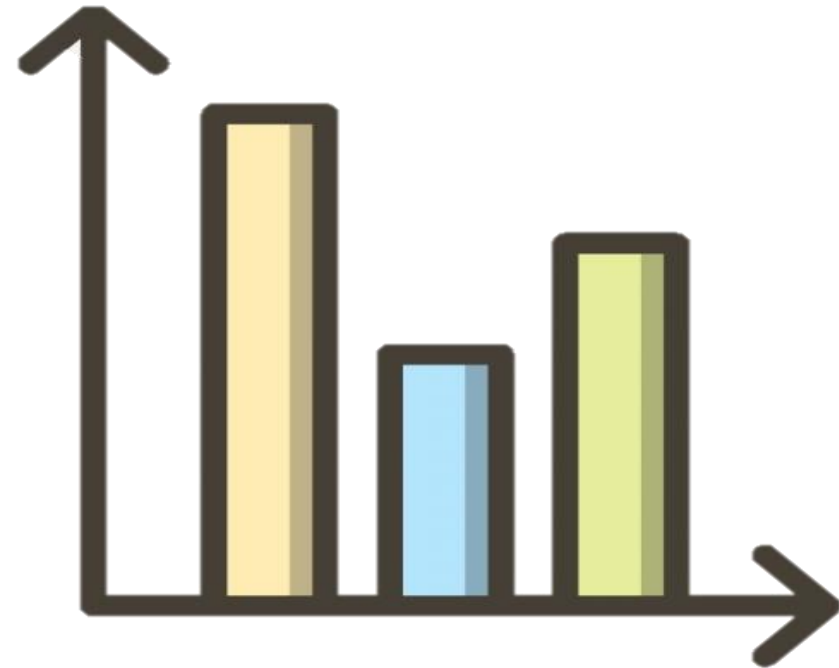
SESSIONS 5

Bar Plot

Data Science Program

Outline

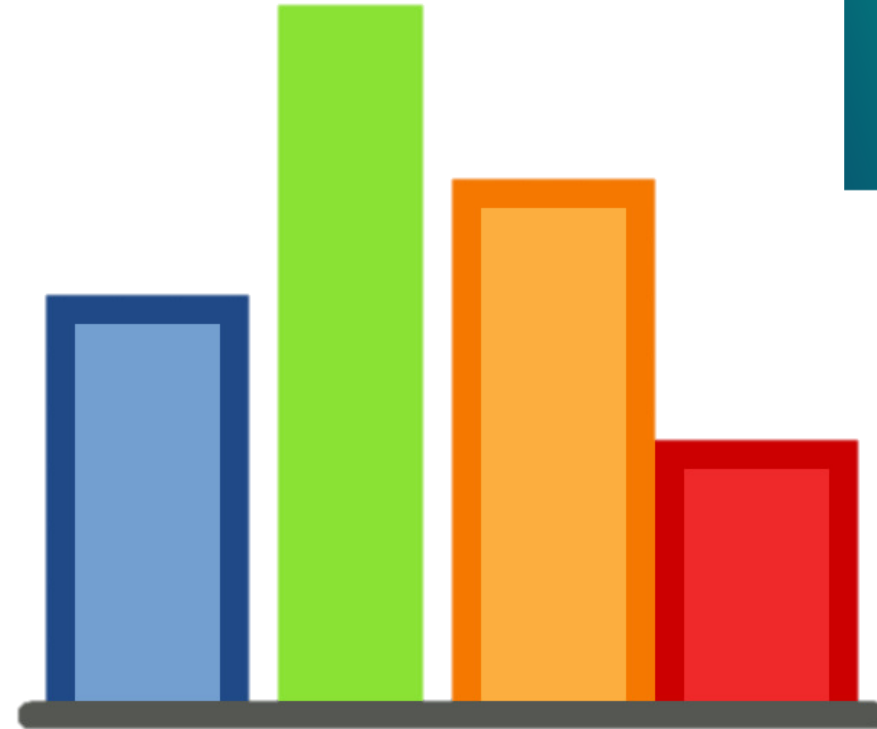
- What is Bar Plot?
- When to Use Bar Plot?
- When to Avoid Bar Plot?
- Create Bar Plot using Matplotlib, Seaborn, and Pandas



What is Bar Plot?

What is Bar Plot?

- A bar chart or bar plot is a chart that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent.
- The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column chart.
- A bar graph shows comparisons among discrete categories. One axis of the chart shows the specific categories being compared, and the other axis represents a measured value.
- Some bar graphs present bars clustered in groups of more than one, showing the values of more than one measured variable.



What is Bar Plot?

- Bar Plot shows the distribution of data over several groups.
- It is commonly confused with a histogram which only takes numerical data for plotting. It helps in comparing multiple numeric values.
- Bar charts are among the most frequently used chart types. As the name suggests a bar chart is composed of a series of bars illustrating a variable's development.
- Given that bar charts are such a common chart type, people are generally familiar with them and can understand them easily.



When to Use Bar Plot?

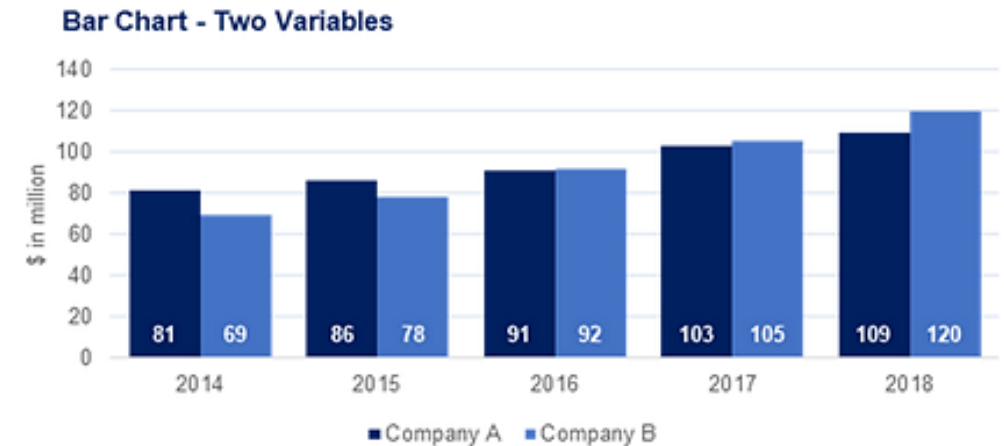
When to Use Bar Plot?

- Bar Plot is used when to compare between several groups.
- Bar charts are nice but limited. We have to consider the type of data we want to visualize and the number of variables that will be added to the chart.
- Bar charts are great when we want to track the development of one or two variables over time.
- For example, one of the most frequent applications of bar charts incorporate presentations is to show how a company's total revenues have developed during a given period.



When to Use Bar Plot?

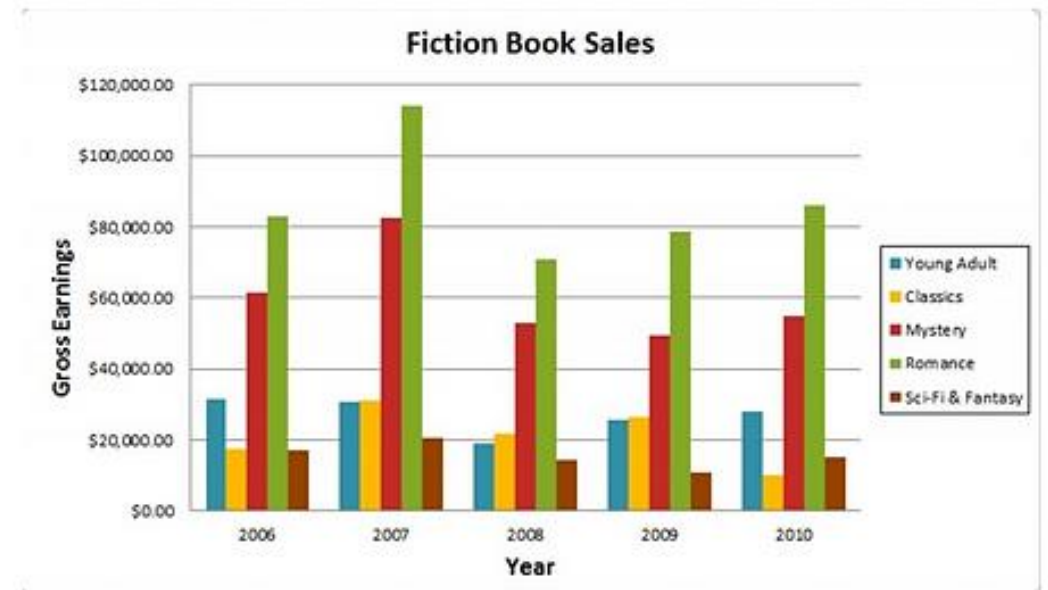
- A bar chart can be used to make both a year-on-year comparison and a monthly breakdown.
- Moreover, bar charts can be pretty intuitive when we compare the development of two numerical variables over time.
- Let's say we would like to compare the revenues of two companies in the timeframe between 2014 and 2018.



When to Avoid Bar Plot?

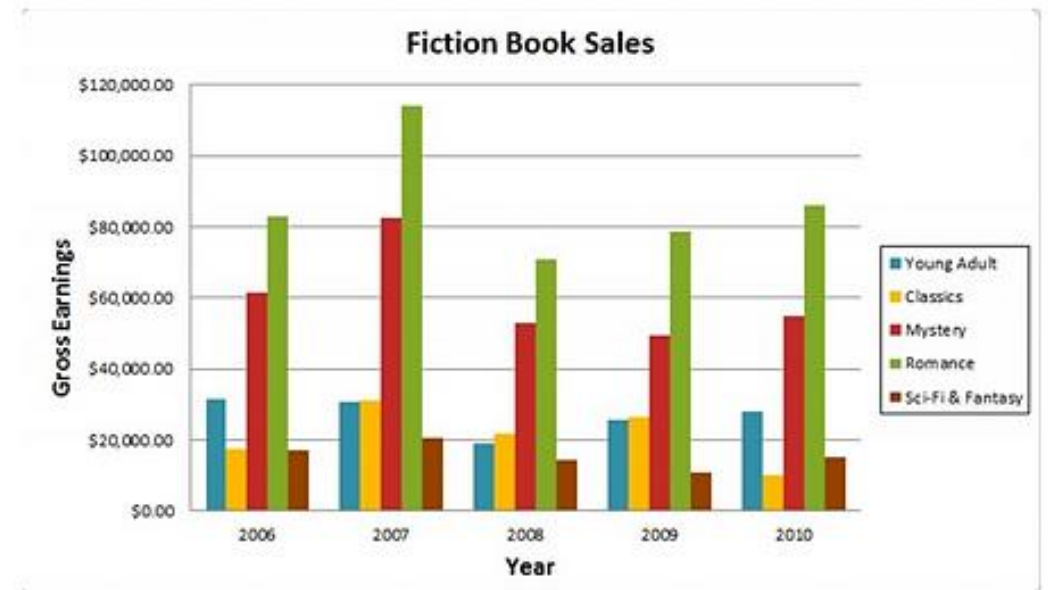
When to Avoid Bar Plot?

- **Simple bar charts** are far from ideal in situations when we have several variables and all of them are part of a whole.
- Such as the case in the Fiction Book Sales chart in this slide, there were five categories: Young adult; classics; mystery; romance; and Sci-fi. These account for all fiction books. Meaning, their sum gives us the total volume of the Fiction book sales market.
- Do you get any of this information with this bar chart?



When to Avoid Bar Plot?

- It simply shows us **multiple lines** and one has to start making calculations on their own to understand how numbers developed over time.
- And if they have to do that, why bother even creating a chart in the first place? **We are better off showing the data in a table format,** right?



Create Bar Plot using Matplotlib

Create Bar Plot using Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

```
[1]: # Import Matplotlib & Seaborn
import matplotlib.pyplot as plt
import seaborn as sns

# Import Tips Dataset from seaborn
tips = sns.load_dataset("tips")
tips.head(3)
```

```
[1]:
```

	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]: tips_by_day = tips[['tip', 'day']].groupby(['day']).mean()
tips_by_day
```

```
[3]:
```

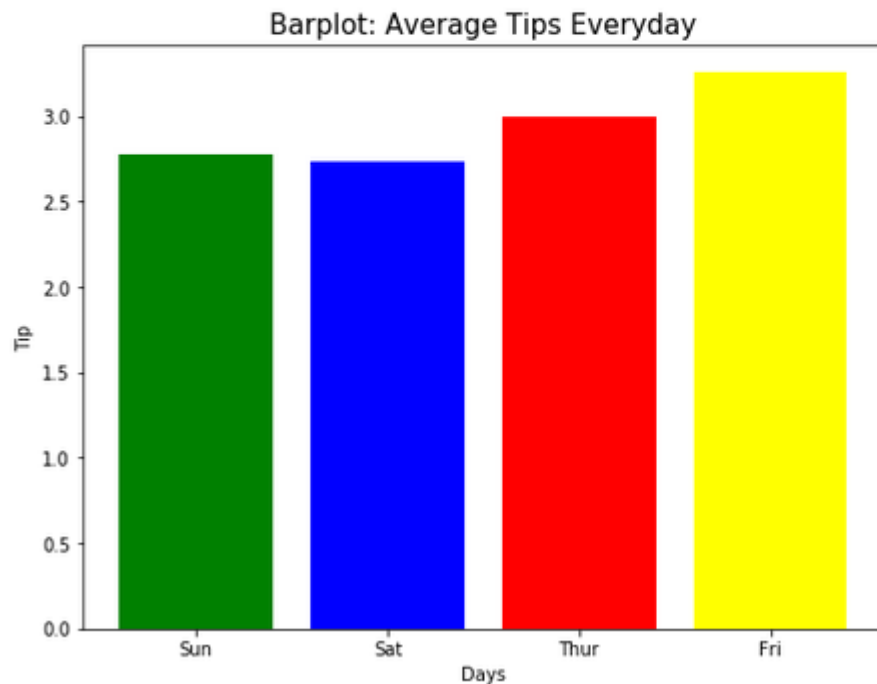
	tip
day	
Thur	2.771452
Fri	2.734737
Sat	2.993103
Sun	3.255132

Create Bar Plot using Matplotlib

```
[6]: x = tips['day'].unique().tolist()
     y = tips_by_day['tip']

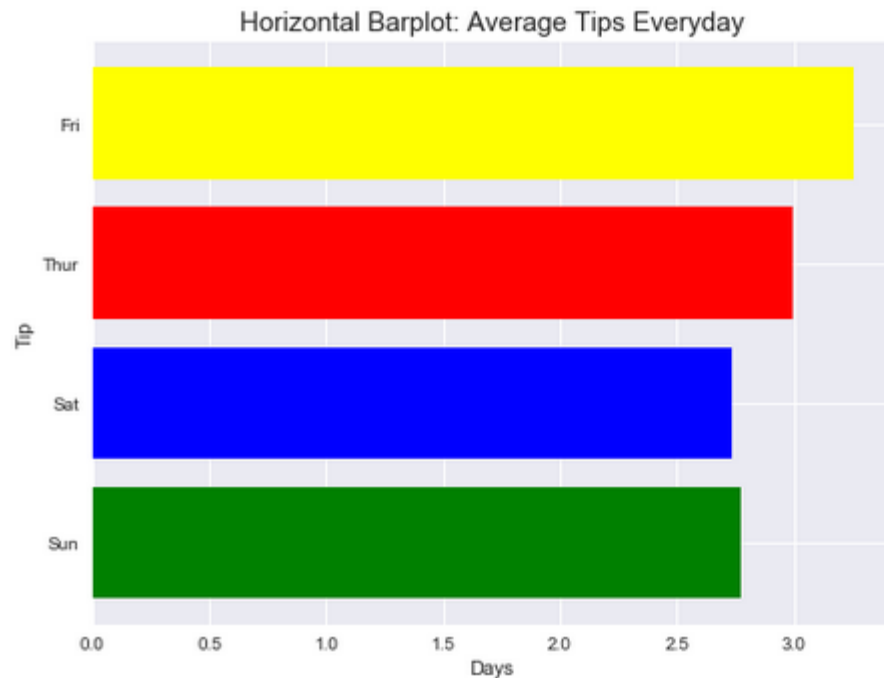
     plt.figure(figsize=(8,6))
     plt.bar(x, y, color = ['green', 'blue', 'red', 'yellow'])
     plt.title('Barplot: Average Tips Everyday', size=15)
     plt.xlabel('Days')
     plt.ylabel('Tip')
     plt.show()
```

figure size
create barplot in matplotlib
Title
X Label
Y Label



Create Bar Plot using Matplotlib

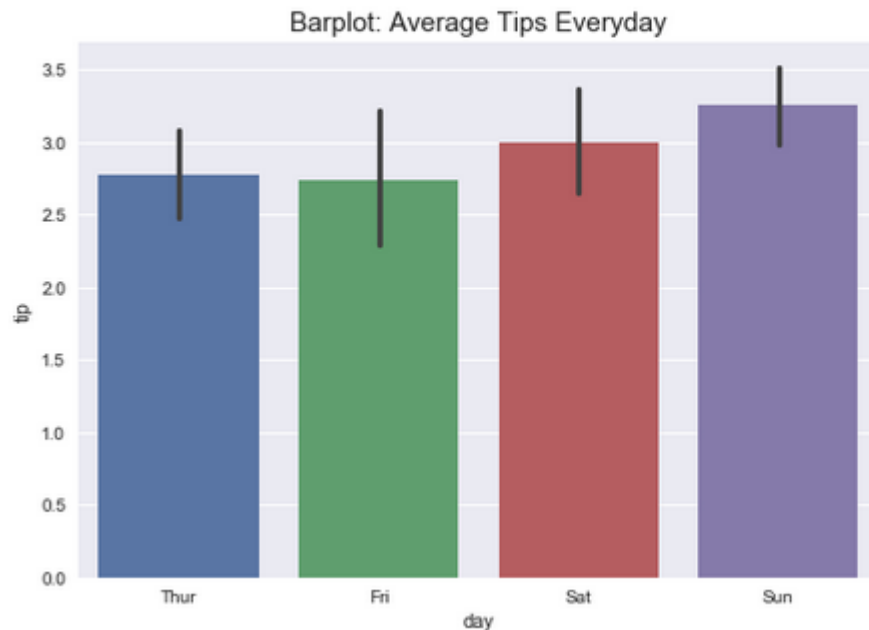
```
[20]: plt.style.use('seaborn')           # change style
      plt.figure(figsize=(8,6))         # figure size
      plt.barh(x, y, color = ['green', 'blue', 'red', 'yellow']) # create horizontal barplot
      plt.title('Horizontal Barplot: Average Tips Everyday', size=15) # Title
      plt.xlabel('Days')                # X Label
      plt.ylabel('Tip')                  # Y Label
      plt.grid(True)                    # add grid
      plt.savefig('Barplot_TipsEveryday.png') # saving plot
      plt.show()
```



Create Bar Plot using Seaborn

Create Bar Plot using Seaborn

```
[8]: sns.barplot(data=tips, x="day", y="tip")           # create bar plot using seaborn
plt.title('Barplot: Average Tips Everyday', size=15)  # add title
plt.show()
```

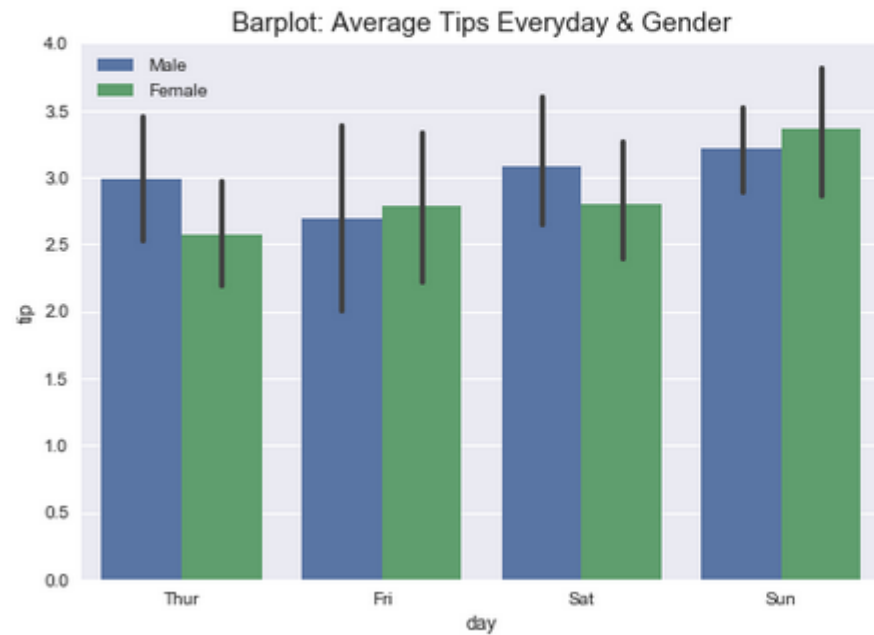


Seaborn is a Python data visualization library based on matplotlib.

It provides a high-level interface for drawing attractive and informative statistical graphics.

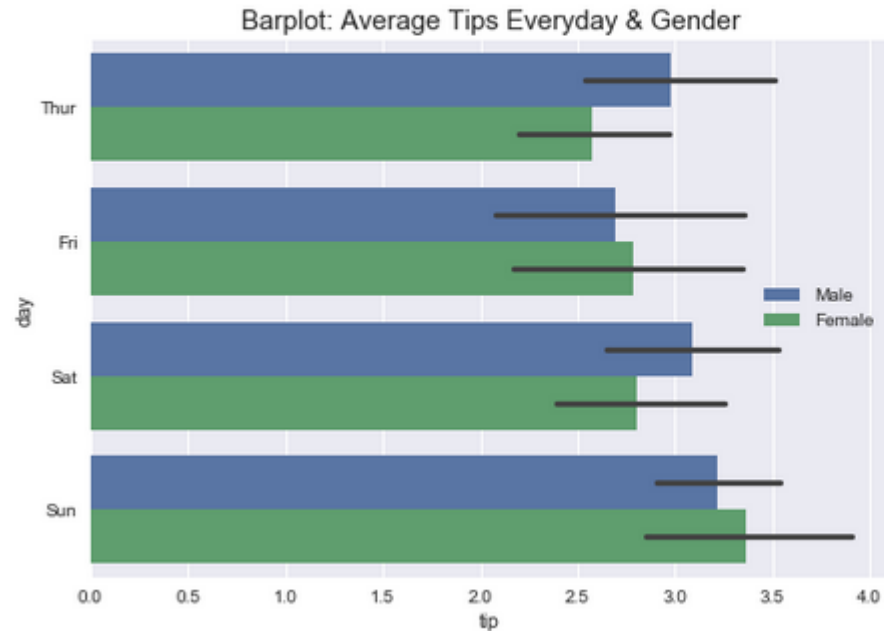
Create Bar Plot using Seaborn

```
[11]: sns.barplot(data=tips, x="day", y="tip", hue='sex')           # create bar plot using seaborn
      plt.title('Barplot: Average Tips Everyday & Gender', size=15) # add title
      plt.legend(loc=0)
      plt.show()
```



Create Bar Plot using Seaborn

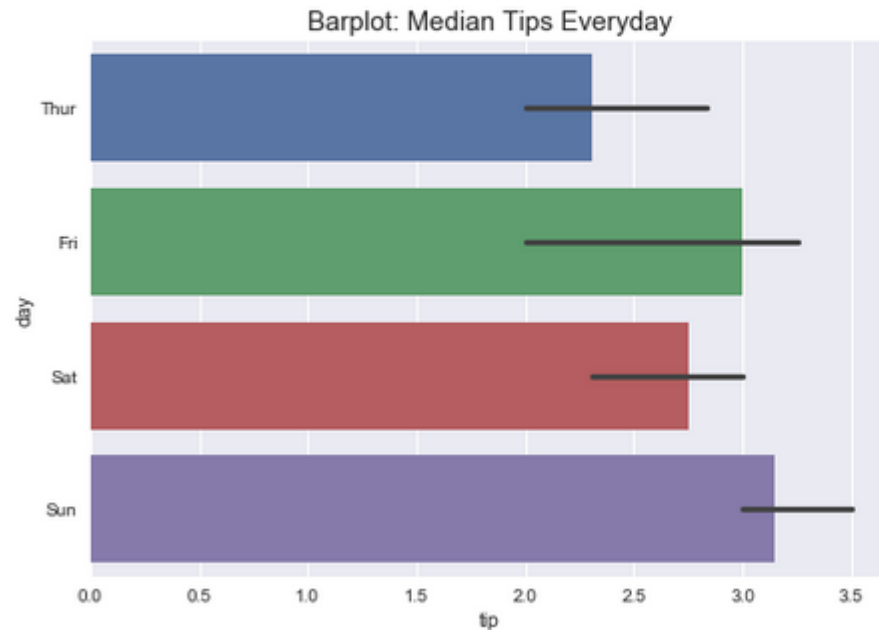
```
[12]: sns.barplot(data=tips, x="tip", y="day", hue='sex')           # create bar plot using seaborn
plt.title('Barplot: Average Tips Everyday & Gender', size=15)    # add title
plt.legend(loc=0)
plt.show()
```



Create Bar Plot using Seaborn

```
[13]: import numpy as np

sns.barplot(data=tips, x="tip", y="day", estimator=np.median) # create bar plot using seaborn
plt.title('Barplot: Median Tips Everyday', size=15)           # add title
plt.show()
```



Create Bar Plot using Pandas

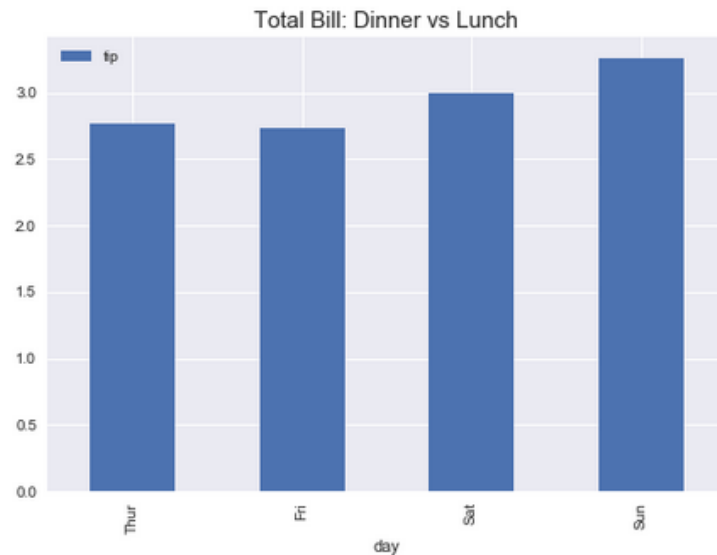
Create Bar Plot using Pandas

```
[17]: tips_by_day = tips[['tip', 'day']].groupby(['day'], as_index=False).mean()
tips_by_day
```

```
[17]:
```

	day	tip
0	Thur	2.771452
1	Fri	2.734737
2	Sat	2.993103
3	Sun	3.255132

```
[19]: tips_by_day.plot.bar(x='day', y='tip') # create bar plot using pandas
plt.title('Total Bill: Dinner vs Lunch', size=15) # add title
plt.show()
```



Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Reference

- 365datascience, "Choosing the right chart: Selecting among 14 chart types", <https://365datascience.com/chart-types-and-how-to-select-the-right-one/>
- Badreesh Shetty, "Data Visualization using Matplotlib", <https://towardsdatascience.com/data-visualization-using-matplotlib-16f1aae5ce70>
- Wikipedia, "Bar chart", https://en.wikipedia.org/wiki/Bar_chart