



BATCH : B150 Data Science
LESSON : **Data Visualization**
DATE : 28.04.2023
SUBJECT : **Intro & Matplotlib**

ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ



Data Visualization With Python





Eğitim Programı

- Data Visualization introsu
- Matplotlib
- Seaborn
- Plotly
- Visualization project



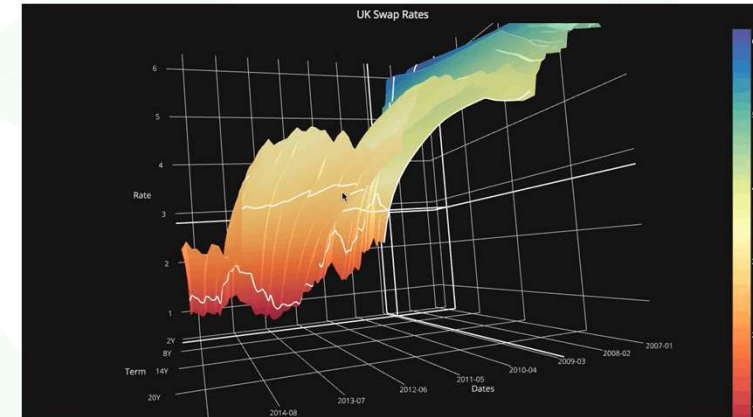
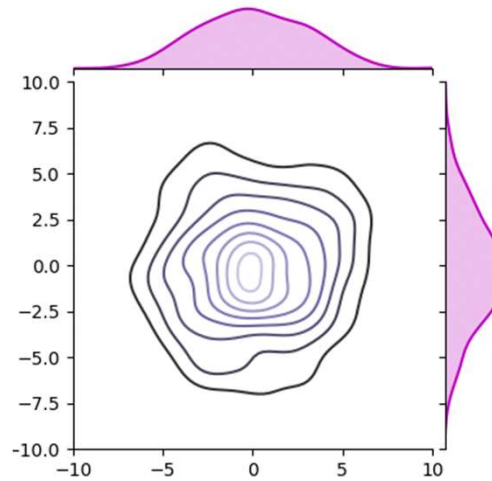
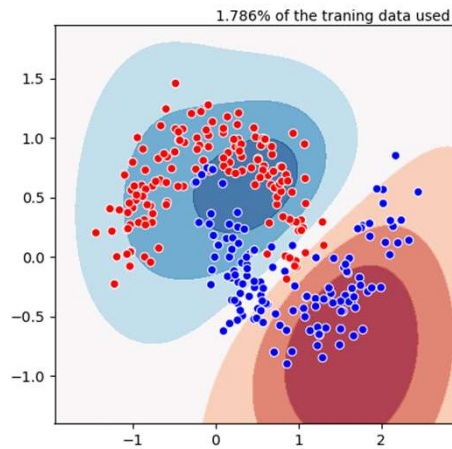
Kursun Kapsamı

matplotlib



seaborn

plotly





Kursun Kapsamı

matplotlib



seaborn

plotly

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
import pandas as pd
```

Create figure, axes, subplots

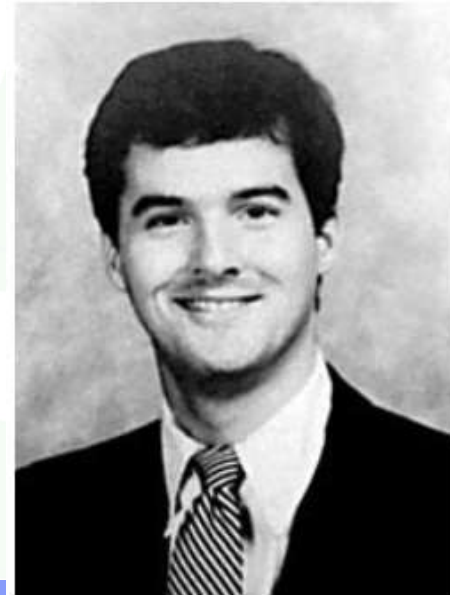
Built on matplotlib and
can be used together with it

No need to import matplotlib
or seaborn



Matplotlib & Seaborn

- Python'da veriyi görselleştirmek için kullanılır.
- 2002 yılında John Hunter tarafından matlap tarzında bir arayüz oluşturmak amacıyla bir proje olarak başlatıldı.
- İlk sürümü 2003 yılında yayınlandı.
- Geliştirilerek seaborn kütüphanesi ortaya çıkmıştır.





Matplotlib & Seaborn

FEATURES	MATPLOTLIB	SEABORN
Functionality	<p>It is utilized for making basic graphs. Datasets are visualised with graphs styles.</p> <ul style="list-style-type: none">• Bar graphs,• Histograms,• Pie charts,• Scatter plots,• Lines <p>and so on.</p>	<p>Seaborn contains a number of patterns and plots for data visualization. It uses fascinating themes. It helps in compiling whole data into a single plot.</p>
Syntax	<p>It uses comparatively complex and lengthy syntax.</p>	<p>It uses comparatively simple syntax which is easier to learn and understand.</p>

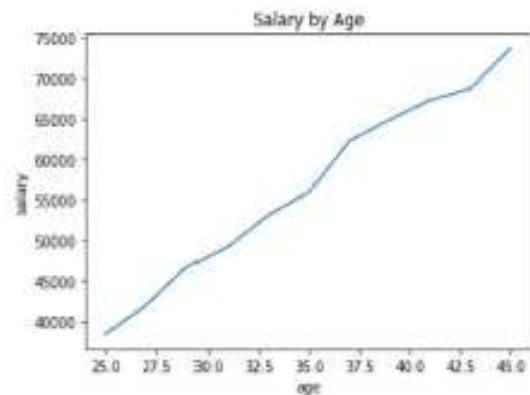


Two Methods

Functional Method

```
plt.plot(age, salary)
plt.xlabel("age")
plt.ylabel("salary")
plt.title("Salary by Age")

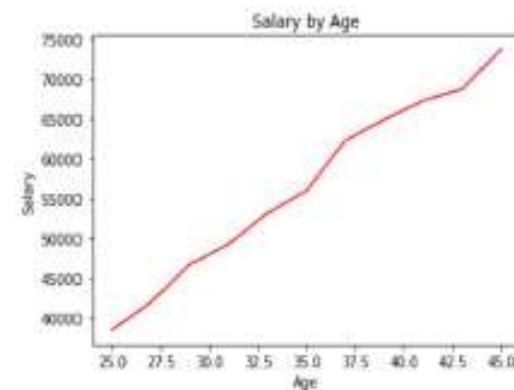
plt.show()
```



Object Oriented

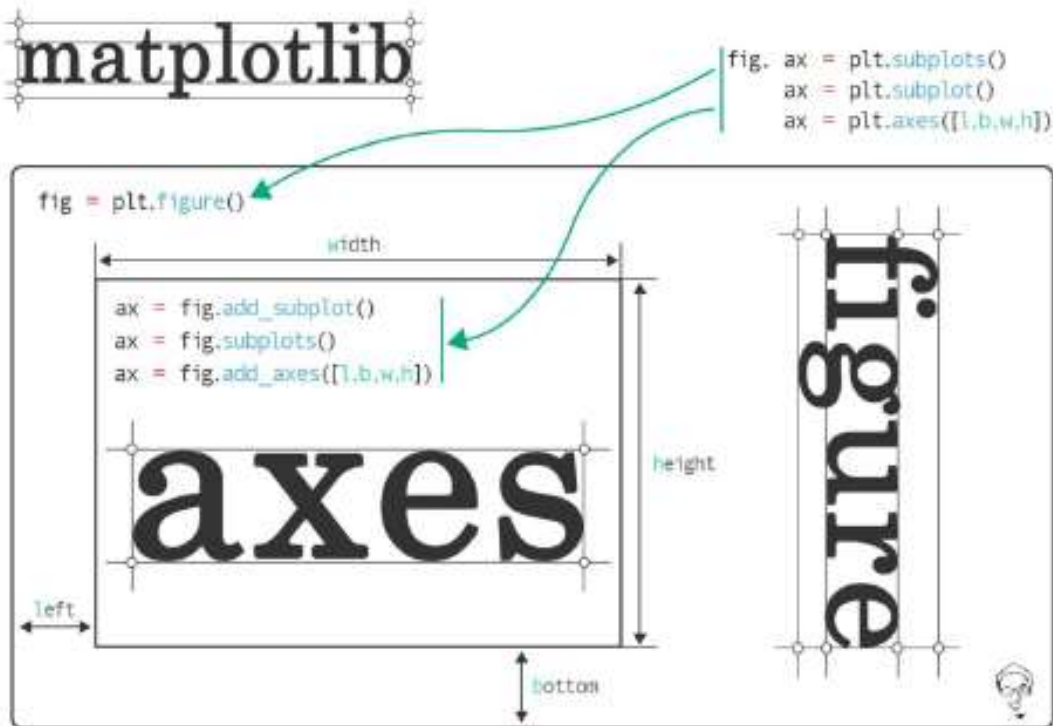
```
fig, ax = plt.subplots()

ax.plot(age, salary, "r")
ax.set_xlabel("Age")
ax.set_ylabel("Salary")
ax.set_title("Salary by Age")
```





Axis – Axes – Figure ?



How do you describe
figure & axes?

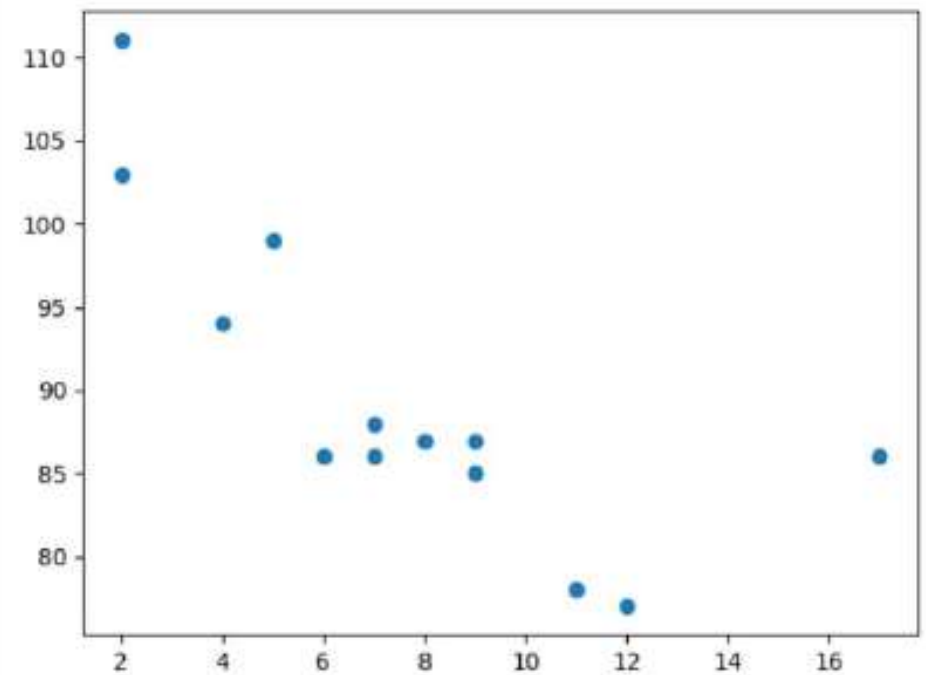


Scatter Plot

```
import matplotlib.pyplot as plt
import numpy as np

x= np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y= np.array([99,86,87,88,111,86,
            103,87,94,78,77,85,86])

plt.scatter(x, y)
plt.show()
```



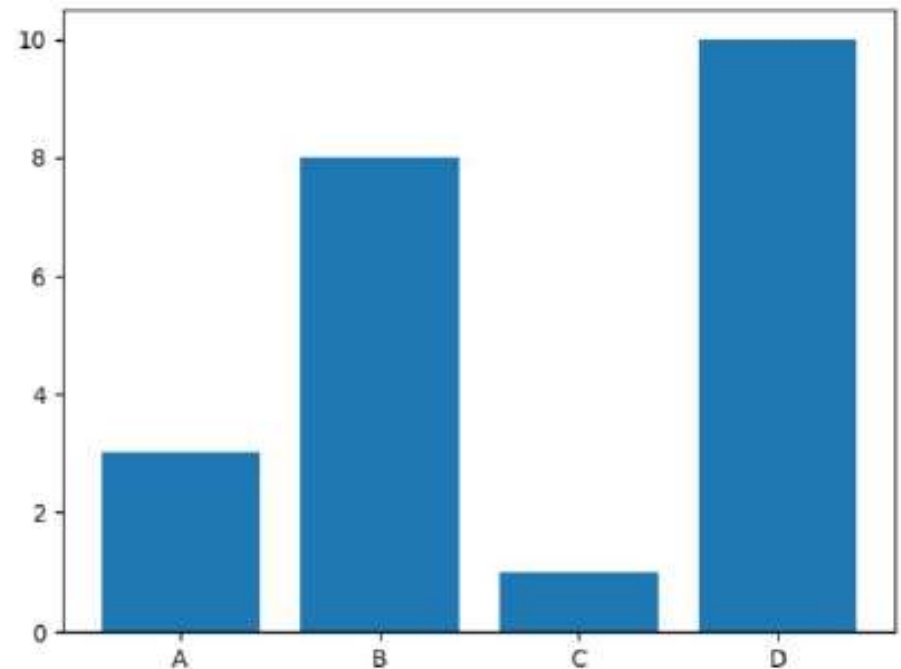


Bar Chart

```
import matplotlib.pyplot as plt
import numpy as np

x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x,y)
plt.show()
```



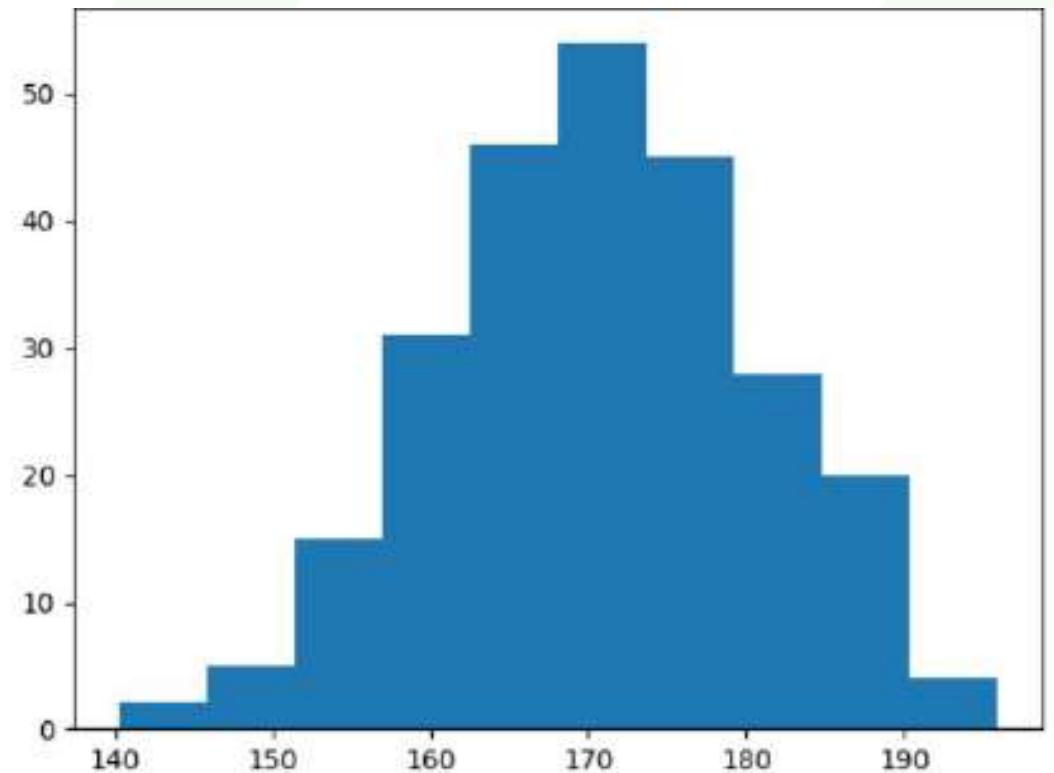


Histogram

```
import matplotlib.pyplot as plt
import numpy as np

x = np.random.normal(170, 10, 250)

plt.hist(x)
plt.show()
```



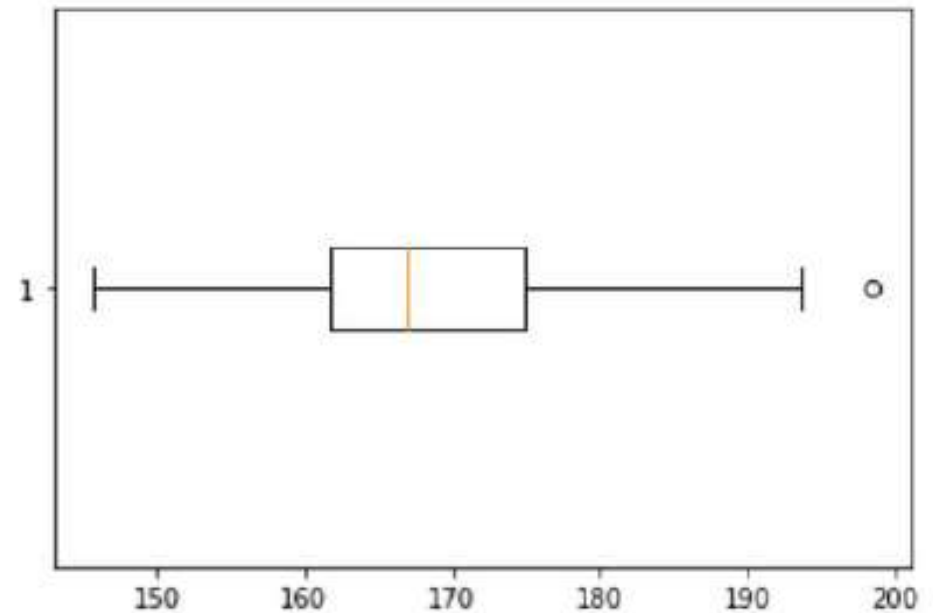


Box Plot

```
import matplotlib.pyplot as plt
import numpy as np

x = np.random.normal(170, 10, 250)

plt.boxplot(x,)
plt.show()
```

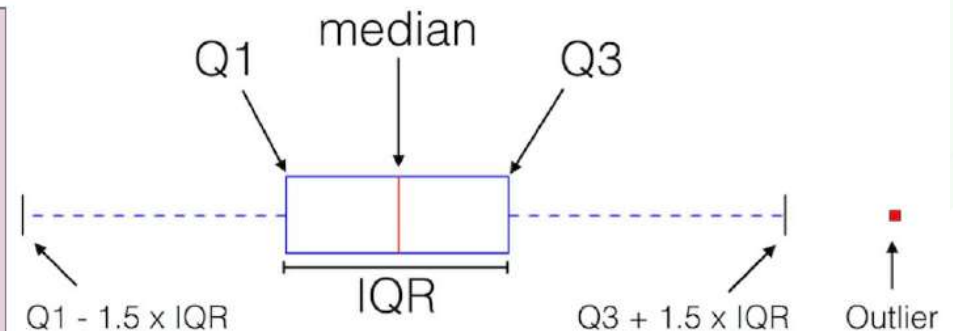




Box Plot

A box plot is a method for graphically depicting groups of numerical data through their quartiles.

A box plot generally shows **median**, **25th and 75th percentiles**, and **outliers**.

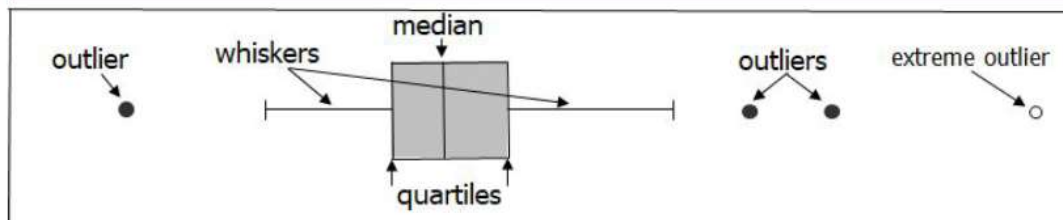


Q1: *Quartile 1*, or median of the *left* data subset after dividing the original data set into 2 subsets via the median (25% of the data points fall below this threshold)

Q3: *Quartile 3*, median of the *right* data subset (75% of the data points fall below this threshold)

IQR: *Interquartile-range*, $Q3 - Q1$

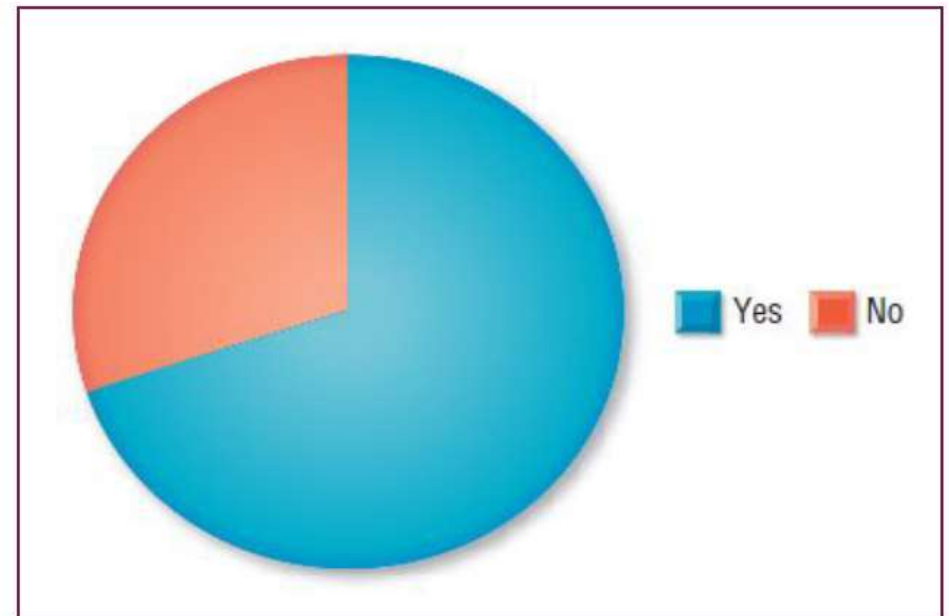
Outliers: Data points are considered to be outliers if
value $< Q1 - 1.5 \times IQR$ or
value $> Q3 + 1.5 \times IQR$





Pie Chart

- ▶ Often used with nominal and ordinal variables.
- ▶ Circle cut into “pie slices” that add up to 100%.
- ▶ Each pie slice represents an attribute for the variable.





Seaborn Plot Types

Distributions Plots

- **kdeplot**
- **rugplot**
- **displot**
- **histplot**

Categorical Plots

- **barplot**
- **countplot**
- **boxplot**
- **swarmplot**
- **violinplot**

Comparison Plots

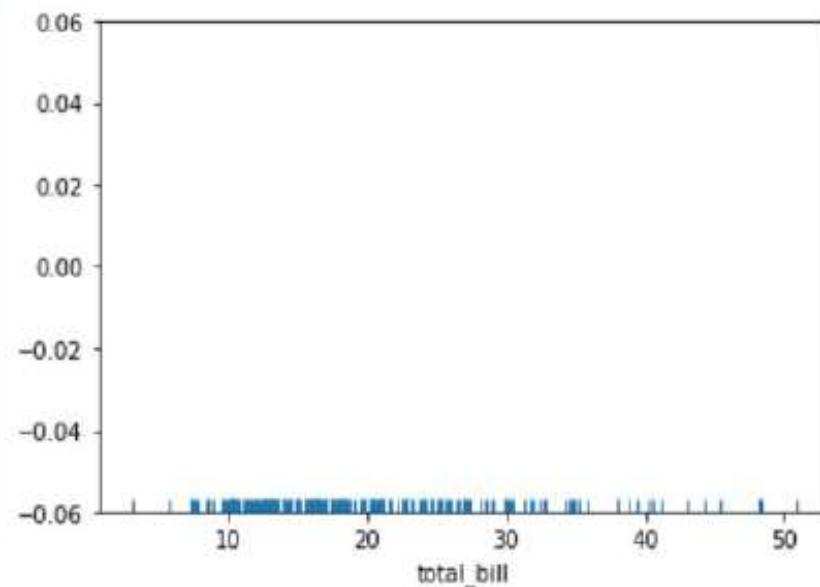
- **jointplot**
- **pairplot**
- **catplot**
- **matrix plot**
- **grid plot**



Distribution Plots - rugplot

```
import seaborn as sns
tips = sns.load_dataset("tips")
sns.rugplot(x='total_bill',
data=tips)
sns.rugplot(tips['total_bill'])
```

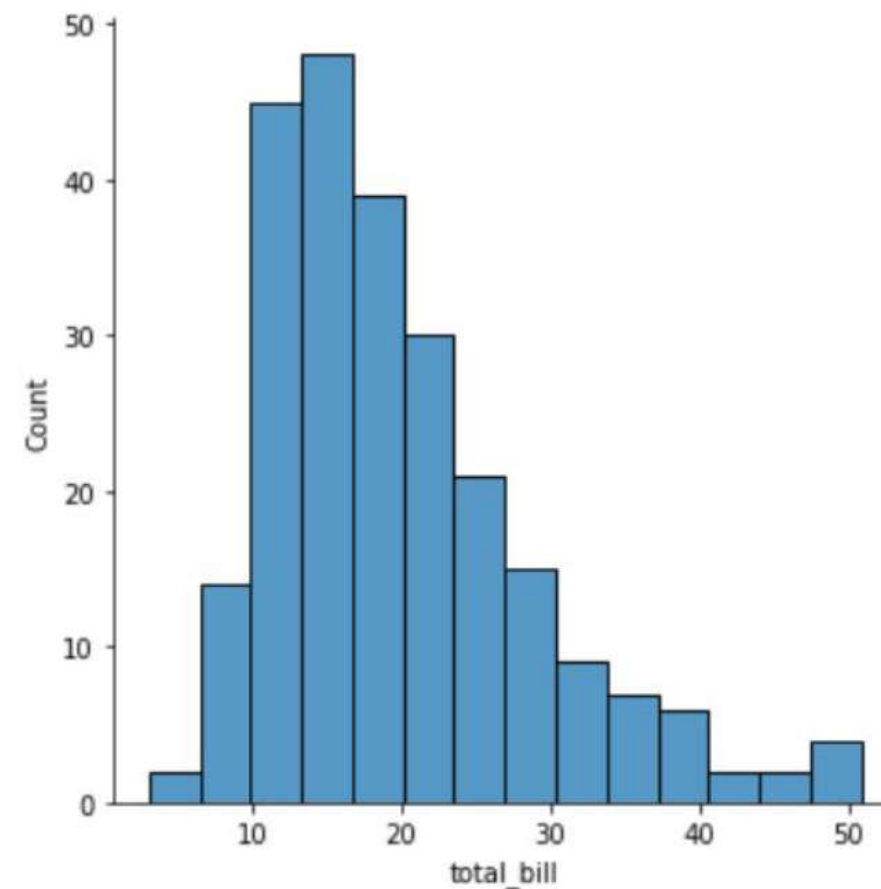
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4





Distribution Plots - displot

```
import seaborn as sns
tips = sns.load_dataset("tips")
# Don't use distplot
sns.displot(x='total_bill',
data=tips)
sns.displot(tips['total_bill'])
```



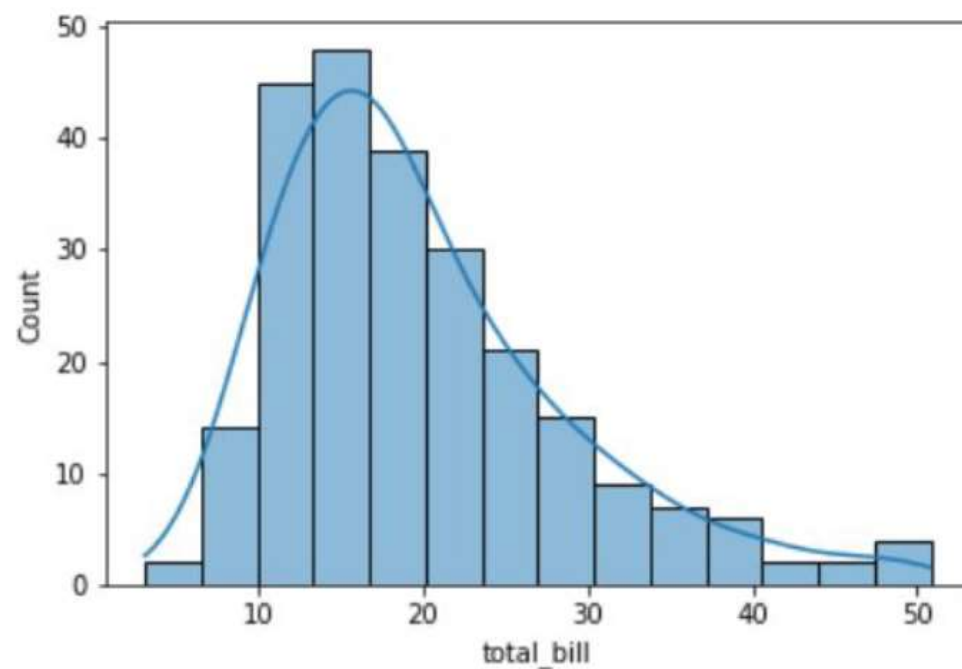
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4



Distribution Plots - histplot

```
import seaborn as sns
tips = sns.load_dataset("tips")
sns.histplot(x='total_bill',
             data=tips,
             kde=True)
sns.histplot(tips['total_bill'])
```

	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

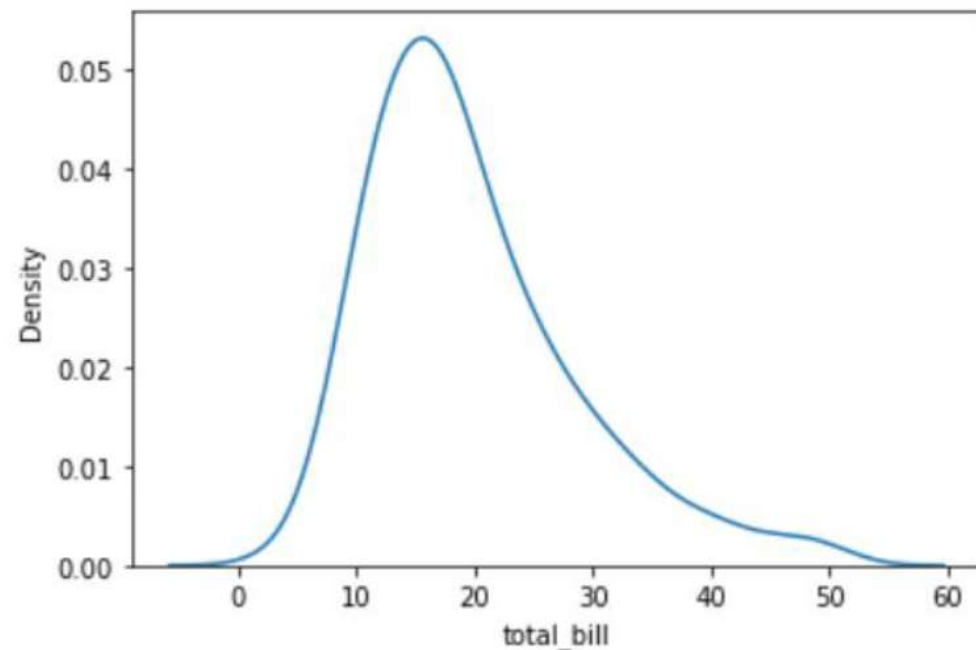




Distribution Plots - kde

```
import seaborn as sns
tips = sns.load_dataset("tips")
sns.kdeplot(x='total_bill',
data=tips)
sns.kdeplot(tips['total_bill'])
```

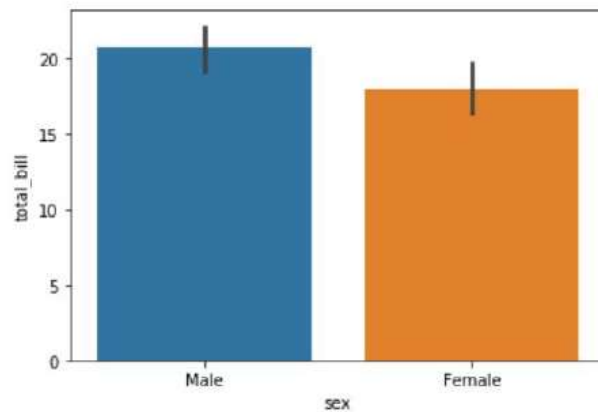
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4



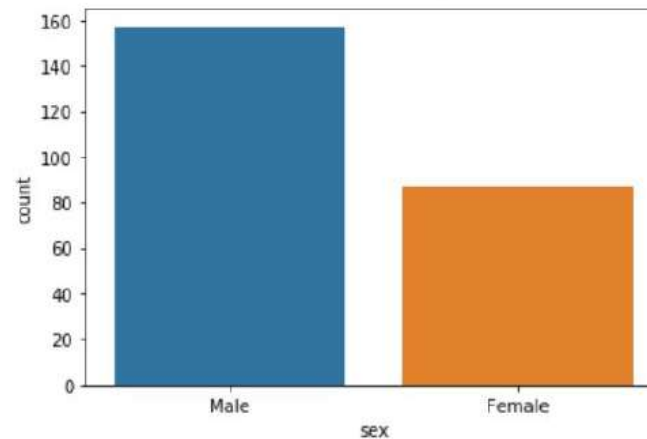


Categorical Plots

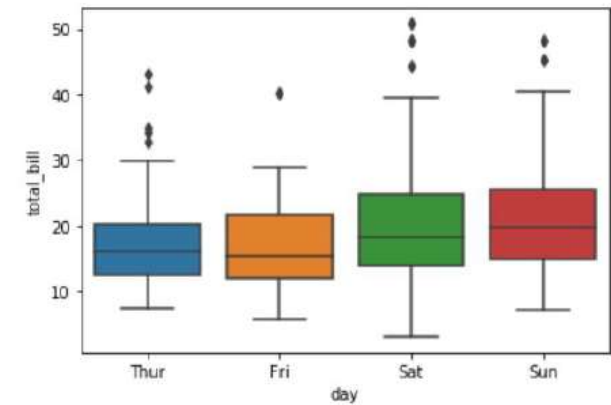
barplot



countplot



boxplot

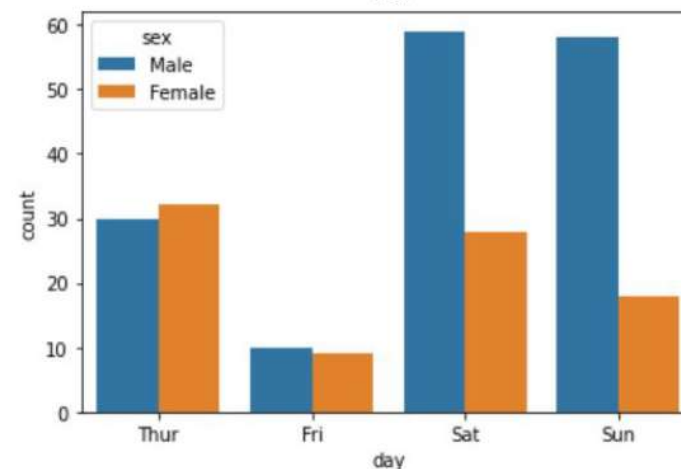
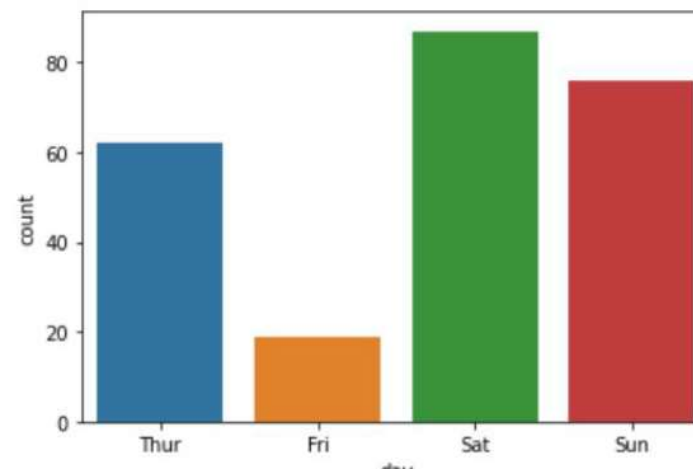




Categorical Plots - countplot

```
import seaborn as sns
tips = sns.load_dataset("tips")
sns.countplot(x='day', data=tips)
sns.countplot(tips['day'])
```

	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

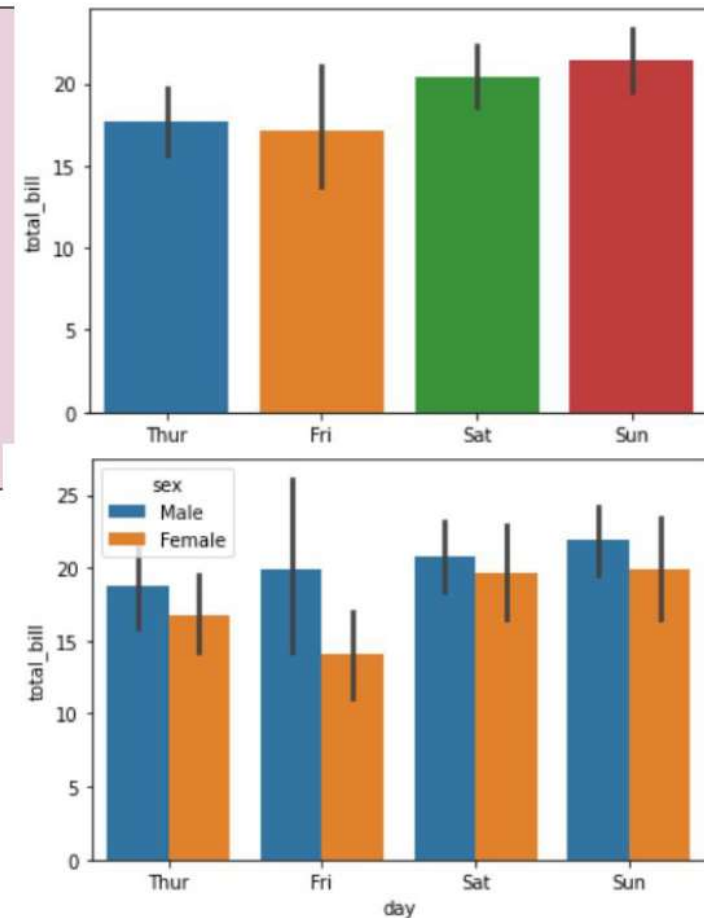




Categorical Plots - barplot

```
import seaborn as sns
tips = sns.load_dataset("tips")
sns.barplot(x='day', y="total_bill",
data=tips)
sns.barplot(x='day', y="total_bill",
data=tips, hue='sex')
```

	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

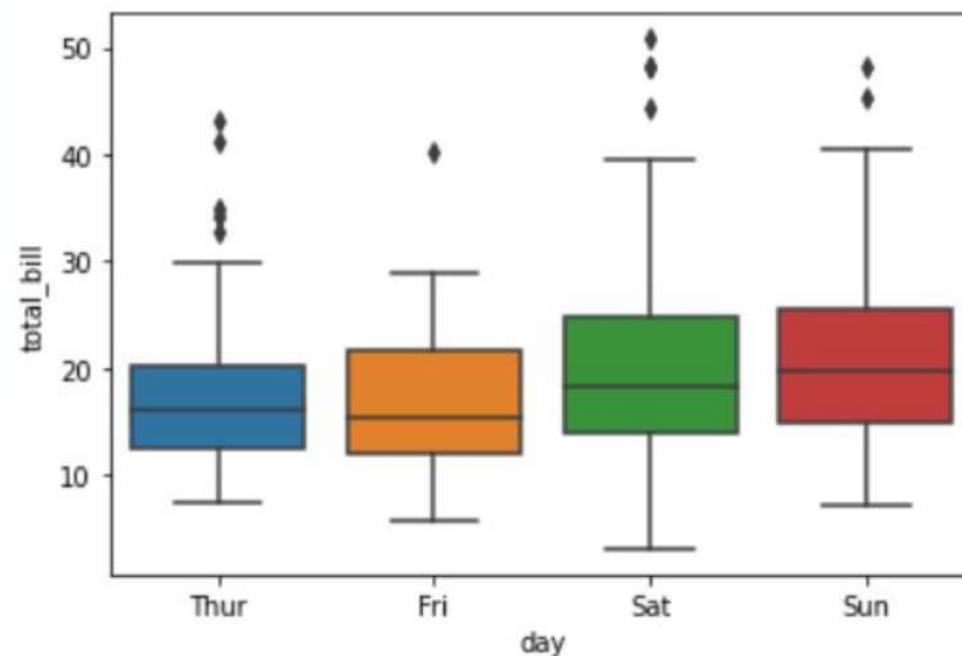




Categorical Plots - boxplot

```
import seaborn as sns
tips = sns.load_dataset("tips")
sns.boxplot(x='day',
            y="total_bill", data=tips)
```

	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



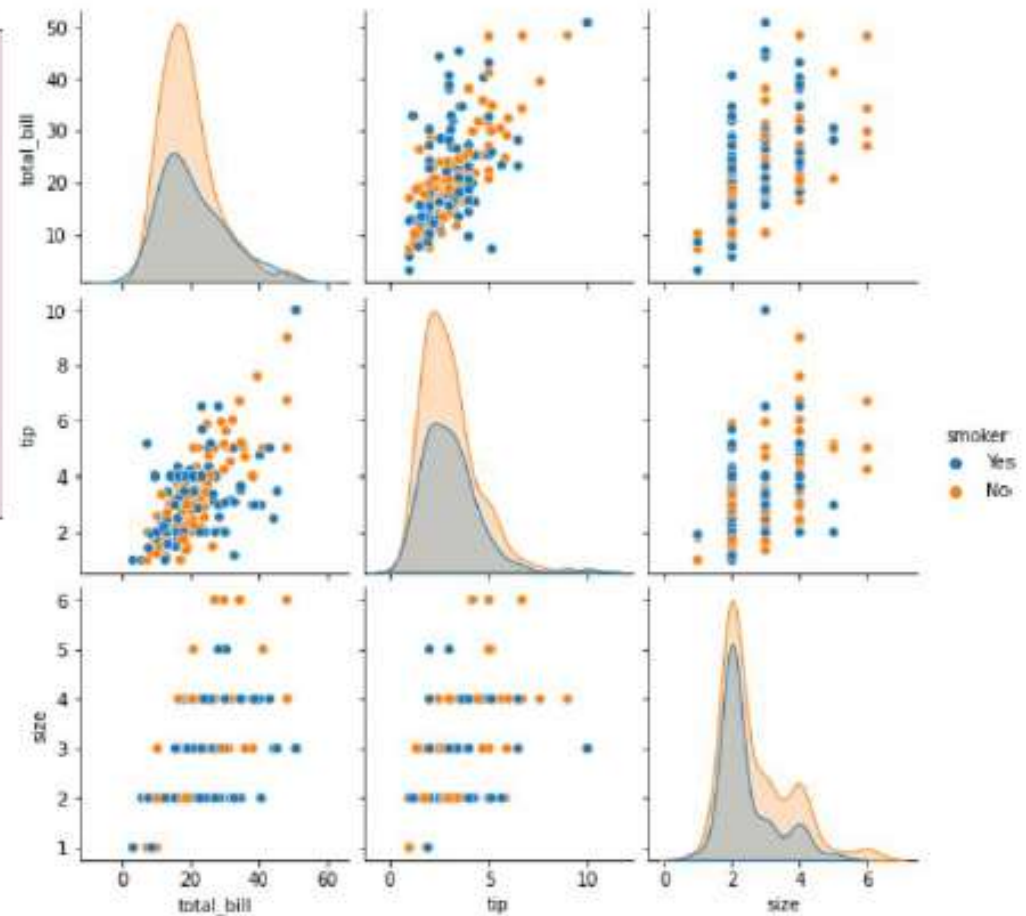


Comparison Plots - pairplot

```
import seaborn as sns
tips = sns.load_dataset("tips")

sns.pairplot(tips, hue="smoker")

# corner=True
```





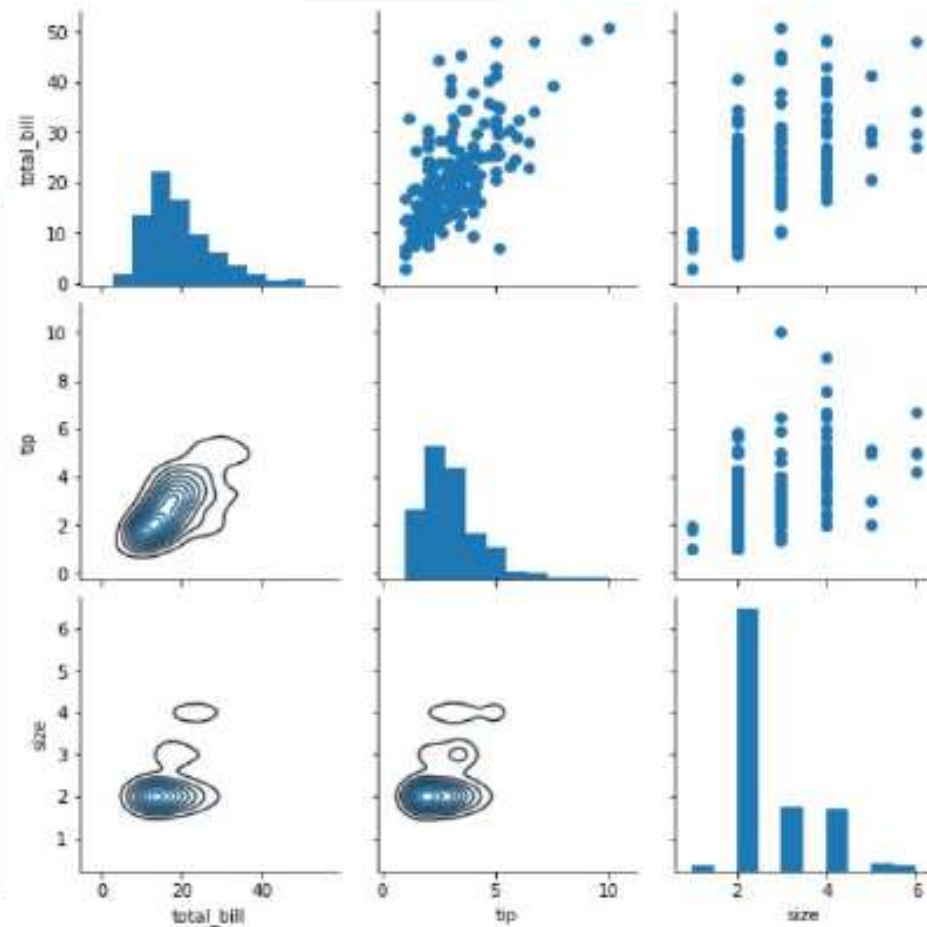
Grids

PairGrid

```
import seaborn as sns
tips = sns.load_dataset("tips")

g = sns.PairGrid(tips)
# g = g.map(sns.scatterplot)

g = g.map_upper(sns.scatterplot)
g = g.map_diag(sns.histplot)
g = g.map_lower(sns.kdeplot)
```





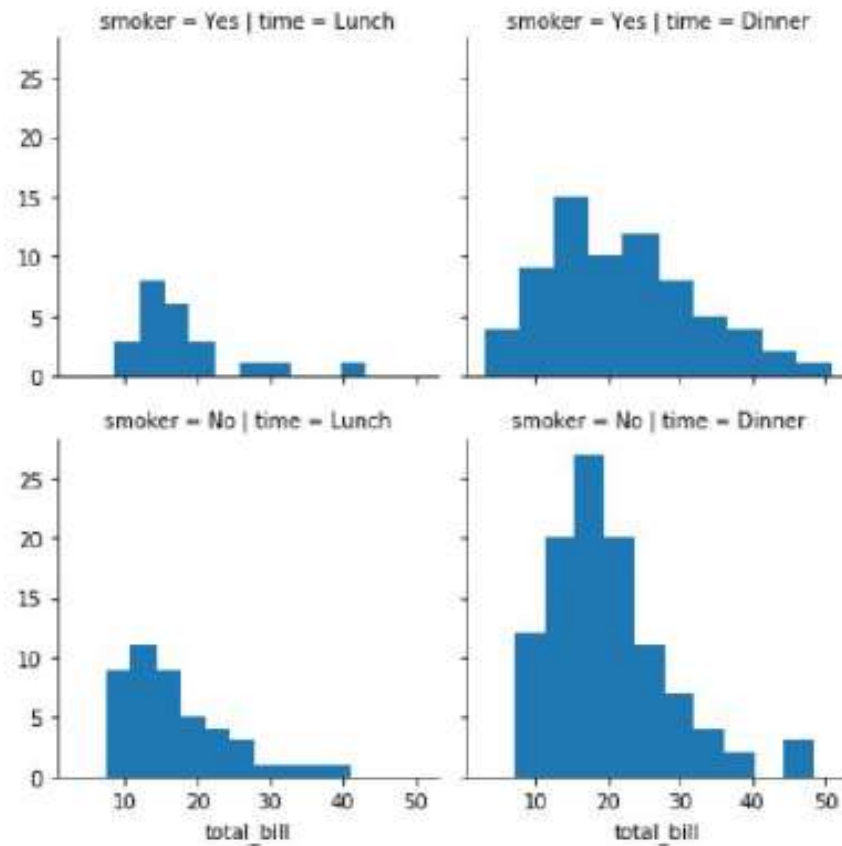
Grids

FacetGrid

```
import seaborn as sns
tips =
sns.load_dataset("tips")

g=sns.FacetGrid( data = tips,
col="time", row="smoker")

g=g.map(plt.hist,"total_bill")
```

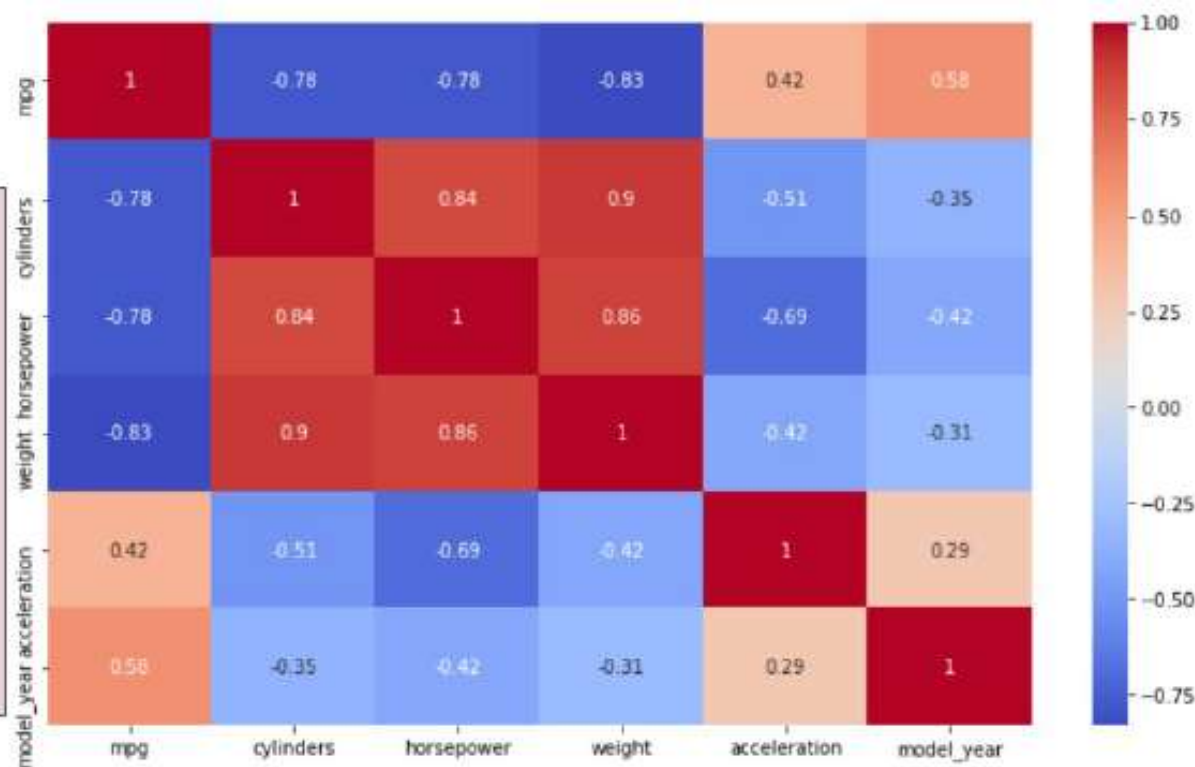




Matrix Plots

heatmap

```
import seaborn as sns  
  
sns.heatmap( df.corr() )
```



Coffee break...

00:00

