Data Visualization

Step-1

```
import seaborn as sns
import matplotlib.pyplot as plt
```

Step-2 Load Dataset

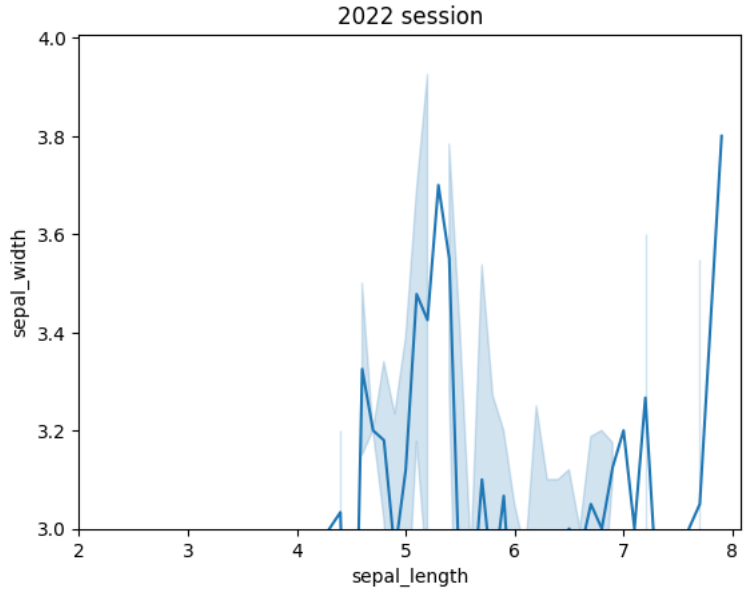
```
iris = sns.load_dataset("iris")
iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Step-3 Plot a Graph

```
sns.lineplot(x="sepal_length",y="sepal_width", data=iris)
plt.xlim(2)
plt.ylim(3)
plt.title("2022 session")
```

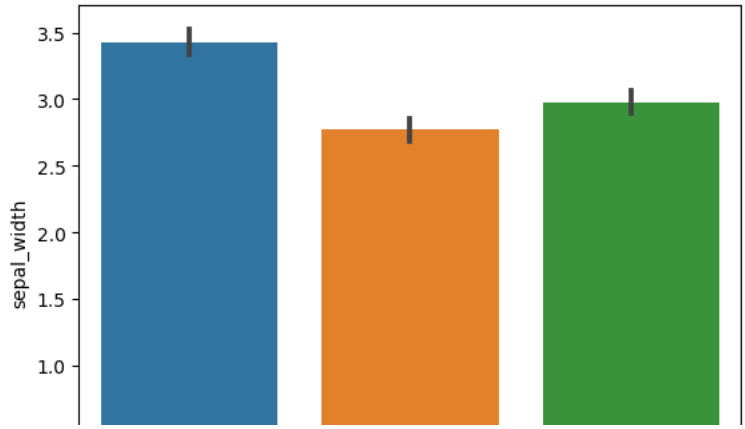
```
Text(0.5, 1.0, '2022 session')
```



Bar plot

```
sns.barplot(x="species",y="sepal_width", data=iris)
```

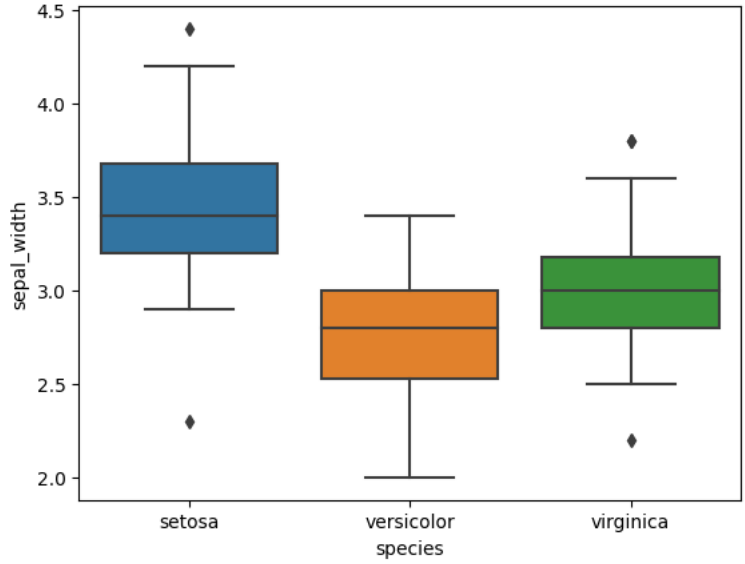
```
<Axes: xlabel='species', ylabel='sepal_width'>
```



Box Plot


```
sns.boxplot(x="species",y="sepal_width", data=iris)
```

```
<Axes: xlabel='species', ylabel='sepal_width'>
```



Scatter Plot

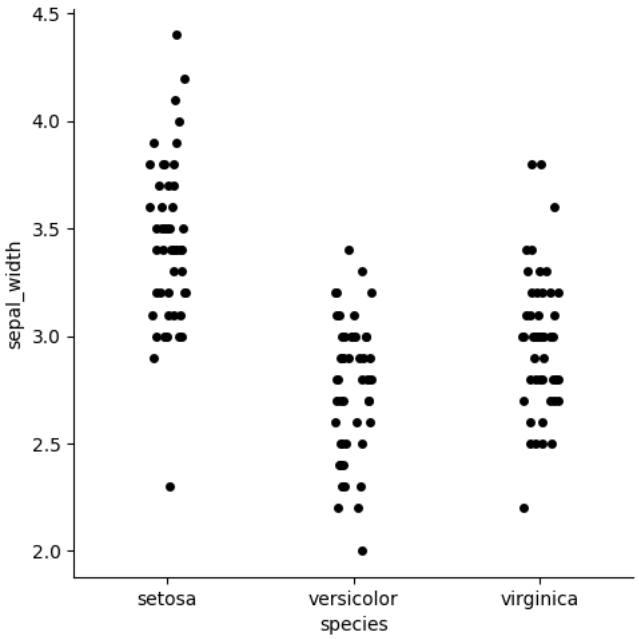
```
sns.scatterplot(x="sepal_length",y="sepal_width", data=iris)
```

```
<Axes: xlabel='sepal_length', ylabel='sepal_width'>
```

CatPlot

```
sns.catplot(x="species",y="sepal_width", data=iris,color="black")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f498e9af2b0>
```




```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="dark")

# Simulate data from a bivariate Gaussian
n = 10000
mean = [0, 0]
cov = [(2, .4), (.4, .2)]
rng = np.random.RandomState(0)
x, y = rng.multivariate_normal(mean, cov, n).T

# Draw a combo histogram and scatterplot with density contours
f, ax = plt.subplots(figsize=(6, 6))
sns.scatterplot(x=x, y=y, s=5, color=".15")
sns.histplot(x=x, y=y, bins=50, pthresh=.1, cmap="mako")
sns.kdeplot(x=x, y=y, levels=5, color="w", linewidths=1)
```

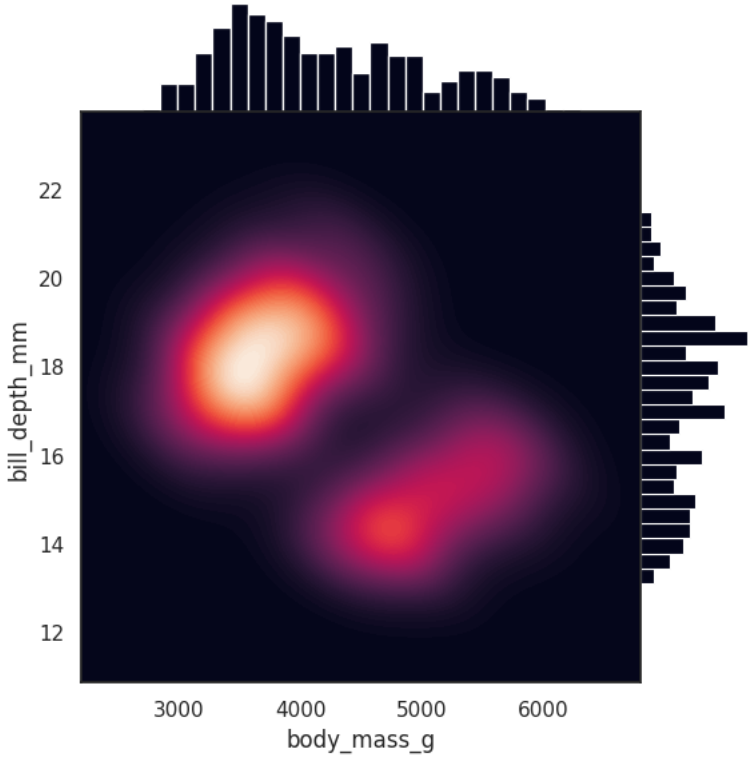
```
<Axes: >
```


```
import seaborn as sns
sns.set_theme(style="white")

df = sns.load_dataset("penguins")

g = sns.JointGrid(data=df, x="body_mass_g", y="bill_depth_mm", space=0)
g.plot_joint(sns.kdeplot,
             fill=True, clip=((2200, 6800), (10, 25)),
             thresh=0, levels=100, cmap="rocket")
g.plot_marginals(sns.histplot, color="#03051A", alpha=1, bins=25)
```

```
<seaborn.axisgrid.JointGrid at 0x7f498e91f430>
```

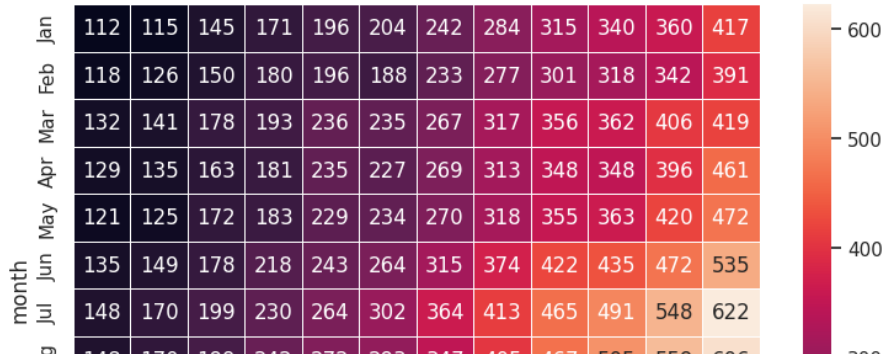



```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme()

# Load the example flights dataset and convert to long-form
flights_long = sns.load_dataset("flights")
flights = flights_long.pivot("month", "year", "passengers")

# Draw a heatmap with the numeric values in each cell
f, ax = plt.subplots(figsize=(9, 6))
sns.heatmap(flights, annot=True, fmt="d", linewidths=.5, ax=ax)
```

```
<ipython-input-19-fd553bdfde69>:7: FutureWarning: In a future version of pandas all argu
  flights = flights_long.pivot("month", "year", "passengers")
<Axes: xlabel='year', ylabel='month'>
```



Data Visualization

Step-1

```
import seaborn as sns
import matplotlib.pyplot as plt
```

Step-2 Load Dataset

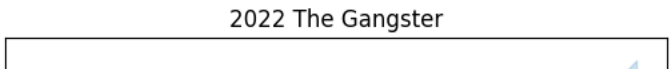
```
flights = sns.load_dataset("flights")
flights.head()
```

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121

Step-3 Plot a Graph

```
sns.lineplot(x="MONTH",y="passengers", data=flights)
plt.xlim(2)
plt.ylim(3)
plt.title("2022 The Gangster")
```

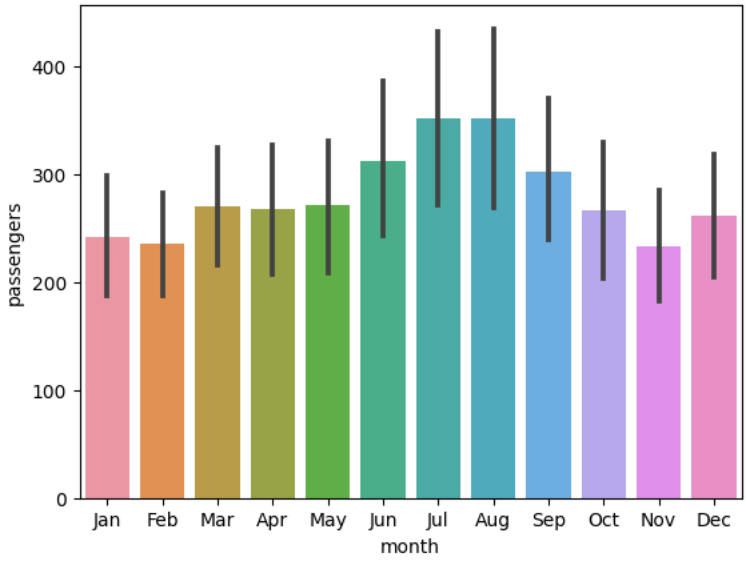
```
Text(0.5, 1.0, '2022 The Gangster')
```



BarPlot

```
sns.barplot(x="month",y="passengers", data=flights)
```

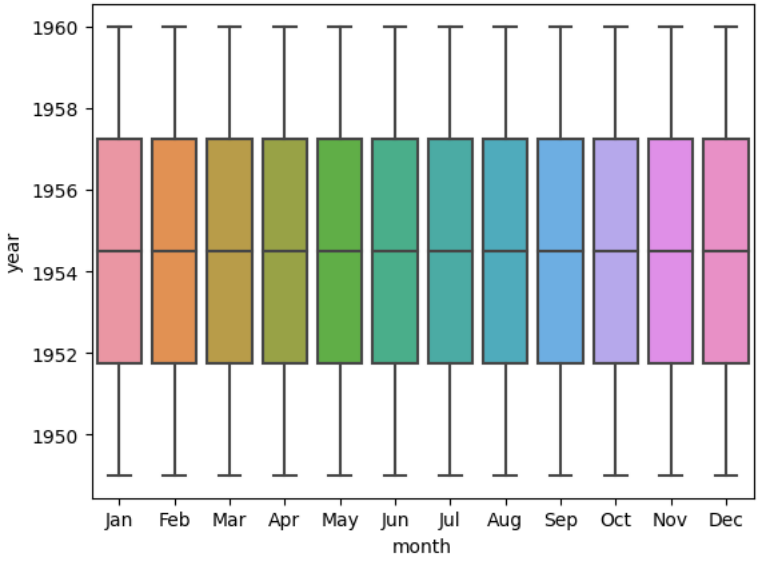
```
<Axes: xlabel='month', ylabel='passengers'>
```



BoxPlot

```
sns.boxplot(x="month",y="year", data=flights)
```

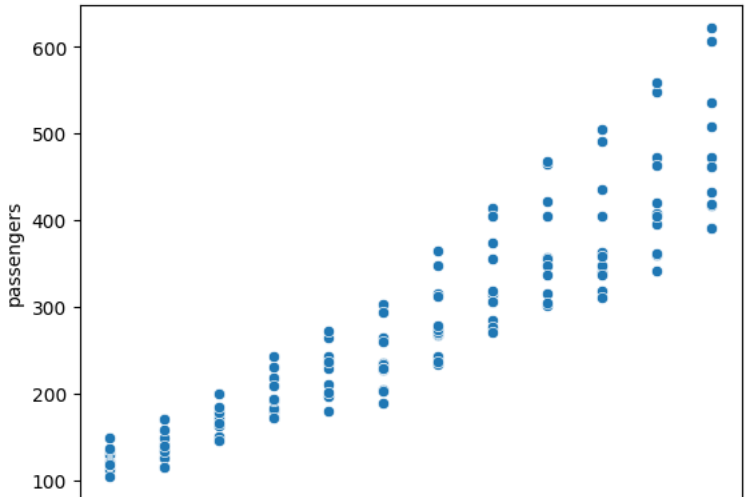
```
<Axes: xlabel='month', ylabel='year'>
```



Scatter Plot

```
sns.scatterplot(x="year",y="passengers", data=flights)
```

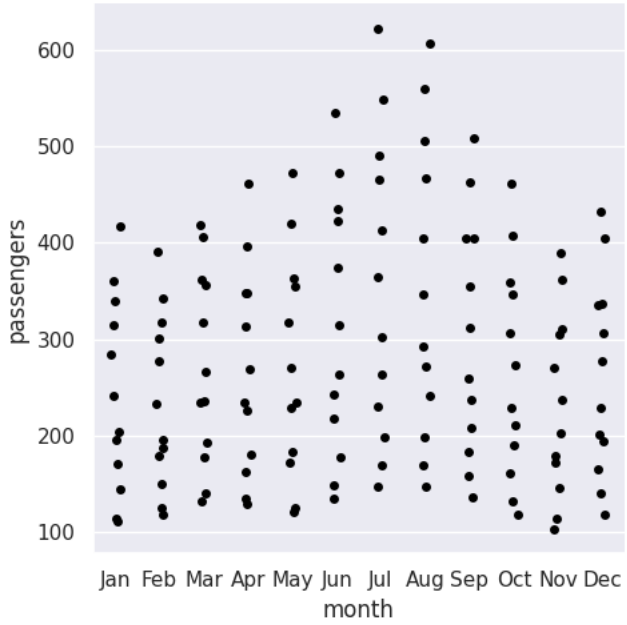
```
<Axes: xlabel='year', ylabel='passengers'>
```



CatPlot

```
sns.catplot(x="month",y="passengers", data=flights,color="black")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f498993b3a0>
```




```
import plotly.express as px
fig= px.scatter(iris, x="sepal_width", y="sepal_length", color="species", size="petal_length",hover_data=['petal_width'])
fig.show()
```

```
 import plotly.express as px
gapminder= px.data.gapminder()
fig=px.scatter(gapminder, x="gdpPercap", y="lifeExp", animation_frame="year", animation_group="country", size="pop", color="continent", hover
fig.show()
```