# **→** DATA VISUALIZATION

#### **▼ STEP 1 IMPORT LIBRARIES**

import seaborn as sns
import matplotlib.pyplot as plt

#### ▼ STEP 2 LOAD DATA SET

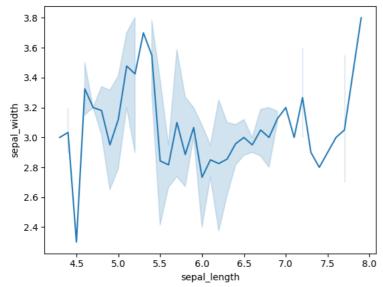
iris = sns.load\_dataset("iris")
iris.head()

	sepal_length	sepal_width	petal_length	petal_width	species	1
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	

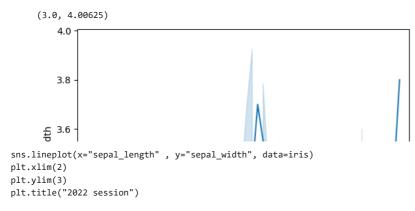
## **▼** STEP-3 PLOT A GRAPH

sns.lineplot(x="sepal\_length" , y="sepal\_width", data=iris)

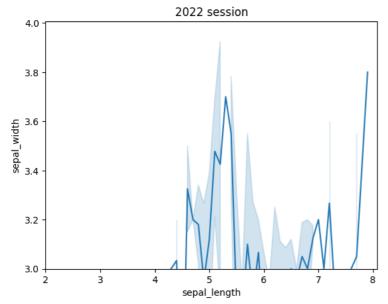
<Axes: xlabel='sepal\_length', ylabel='sepal\_width'>



sns.lineplot(x="sepal\_length" , y="sepal\_width", data=iris)
plt.xlim(2)
plt.ylim(3)

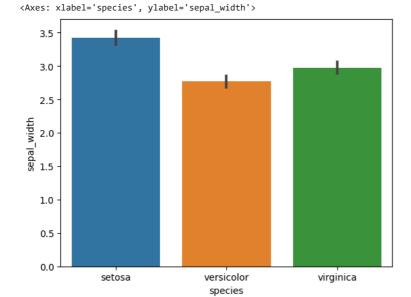


#### Text(0.5, 1.0, '2022 session')



## **→** BAR PLOT

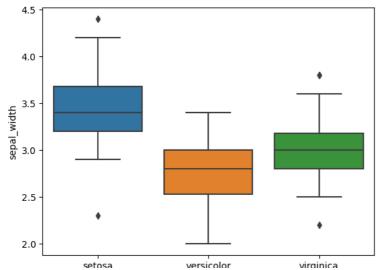
sns.barplot(x="species" , y="sepal\_width", data=iris)



## **→** BOX PLOT

sns.boxplot(x="species" , y="sepal\_width", data=iris)

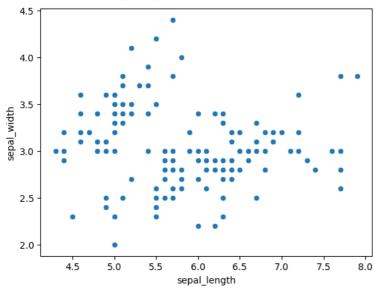
<Axes: xlabel='species', ylabel='sepal\_width'>



## **▼** SCATTER PLOT

sns.scatterplot(x="sepal\_length" , y="sepal\_width", data=iris)

<Axes: xlabel='sepal\_length', ylabel='sepal\_width'>



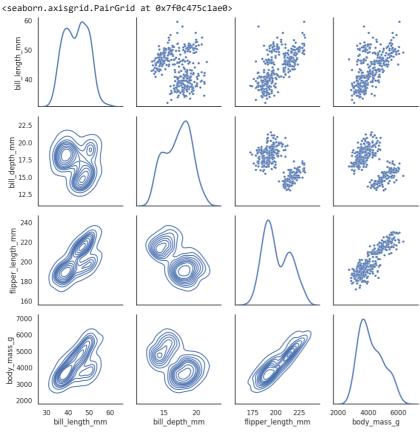
# ▼ CAT PLOR

sns.catplot(x="species" , y="sepal\_width", data=iris , color="gold")

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

df = sns.load_dataset("penguins")

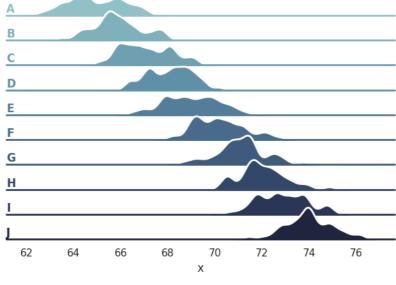
g = sns.PairGrid(df, diag_sharey=False)
g.map_upper(sns.scatterplot, s=15)
g.map_lower(sns.kdeplot)
g.map_diag(sns.kdeplot, lw=2)
```



```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="white", rc={"axes.facecolor": (0, 0, 0, 0)})

# Create the data
rs = np.random.RandomState(1979)
x = rs.randn(500)
g = np.tile(list("ABCDEFGHIJ"), 50)
df = pd.DataFrame(dict(x=x, g=g))
m = df.g.map(ord)
```

```
df["x"] += m
# Initialize the FacetGrid object
pal = sns.cubehelix palette(10, rot=-.25, light=.7)
g = sns.FacetGrid(df, row="g", hue="g", aspect=15, height=.5, palette=pal)
# Draw the densities in a few steps
g.map(sns.kdeplot, "x",
         bw_adjust=.5, clip_on=False,
          fill=True, alpha=1, linewidth=1.5)
g.map(sns.kdeplot, "x", clip_on=False, color="w", lw=2, bw_adjust=.5)
# passing color=None to refline() uses the hue mapping
g.refline(y=0, linewidth=2, linestyle="-", color=None, clip_on=False)
# Define and use a simple function to label the plot in axes coordinates
def label(x, color, label):
      ax = plt.gca()
      ax.text(0, .2, label, fontweight="bold", color=color,
                   ha="left", va="center", transform=ax.transAxes)
g.map(label, "x")
# Set the subplots to overlap
g.figure.subplots adjust(hspace=-.25)
# Remove axes details that don't play well with overlap
g.set titles("")
g.set(yticks=[], ylabel="")
g.despine(bottom=True, left=True)
        /usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:118: UserWarning: Tigh
           self._figure.tight_layout(*args, **kwargs)
        /usr/local/lib/python 3.10/dist-packages/seaborn/axis grid.py: 118: \ User Warning: \ Tight and the packages and the packages and the packages are also become a supplied to the packages and the packages are also become an extension of the packages are also become a supplied to the packages are also become an extension of the packages. \\
           self._figure.tight_layout(*args, **kwargs)
        /usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:118: UserWarning: Tigh
           self._figure.tight_layout(*args, **kwargs)
        /usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:118: UserWarning: Tigh
           self._figure.tight_layout(*args, **kwargs)
        /usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:118: UserWarning: Tigh
           self._figure.tight_layout(*args, **kwargs)
        <seaborn.axisgrid.FacetGrid at 0x7f0c41fed7e0>
           D
           G
```



#### **→** HEAT MAP

```
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-23-fd553bdfde69>:7: FutureWarning: In a future version of pandas al flights = flights\_long.pivot("month", "year", "passengers") <Axes: xlabel='year', ylabel='month'>

Jan	112	115	145	171	196	204	242	284	315	340	360	417		- 600
Feb	118	126	150	180	196	188	233	277	301	318	342	391		
Mar	132	141	178	193	236	235	267	317	356	362	406	419		- 500
Apr	129	135	163	181	235	227	269	313	348	348	396	461		
May	121	125	172	183	229	234	270	318	355	363	420	472		
듔	135	149	178	218	243	264	315	374	422	435	472	535		- 400
month Jul Jun	148	170	199	230	264	302	364	413	465	491	548	622		
Aug	148	170	199	242	272	293	347	405	467	505	559	606		- 300
Sep	136	158	184	209	237	259	312	355	404	404	463	508		
Oct	119	133	162	191	211	229	274	306	347	359	407	461	- 20	- 200
Nov	104	114	146	172	180	203	237	271	305	310	362	390		
Dec	118	140	166	194	201	229	278	306	336	337	405	432		
	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960		

4

year

✓ 2s completed at 10:38

×