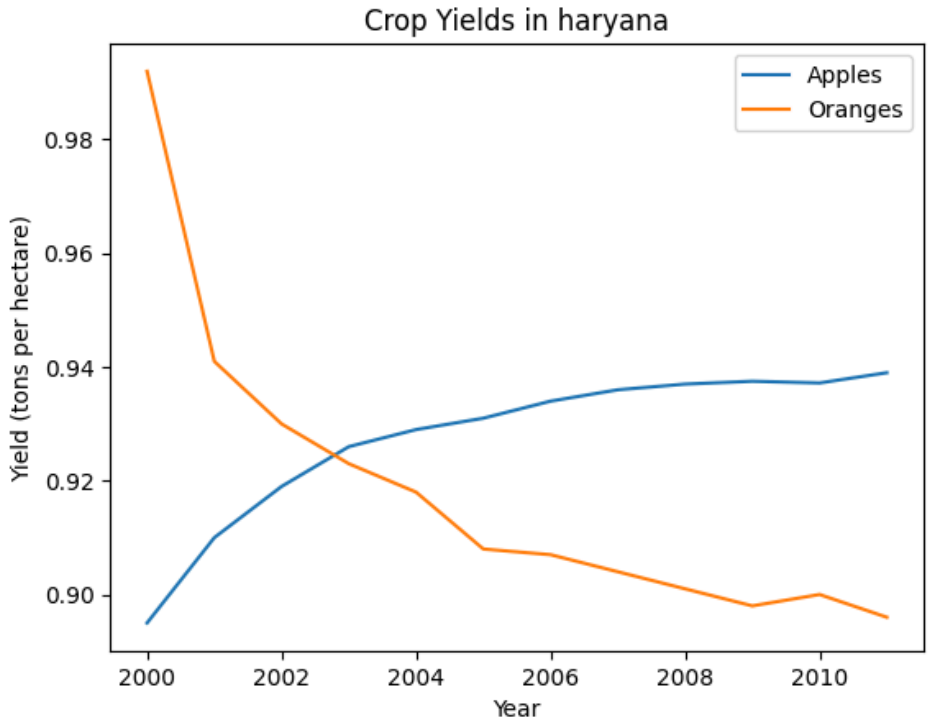
INTRO TO MATPLOTLIB

```
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
```

```
years= range(2000,2012)
apples = [0.895,0.91,0.919,0.926,0.929,0.931,0.934,0.936,0.937,0.9375,0.9372,0.939]
oranges=[0.992,0.941,0.930,0.923,0.918,0.908,0.907,0.904,0.901,0.898,0.9,0.896]
```

```
plt.plot(years, apples)
plt.plot(years, oranges)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in haryana")
plt.legend(["Apples", "Oranges"])
```

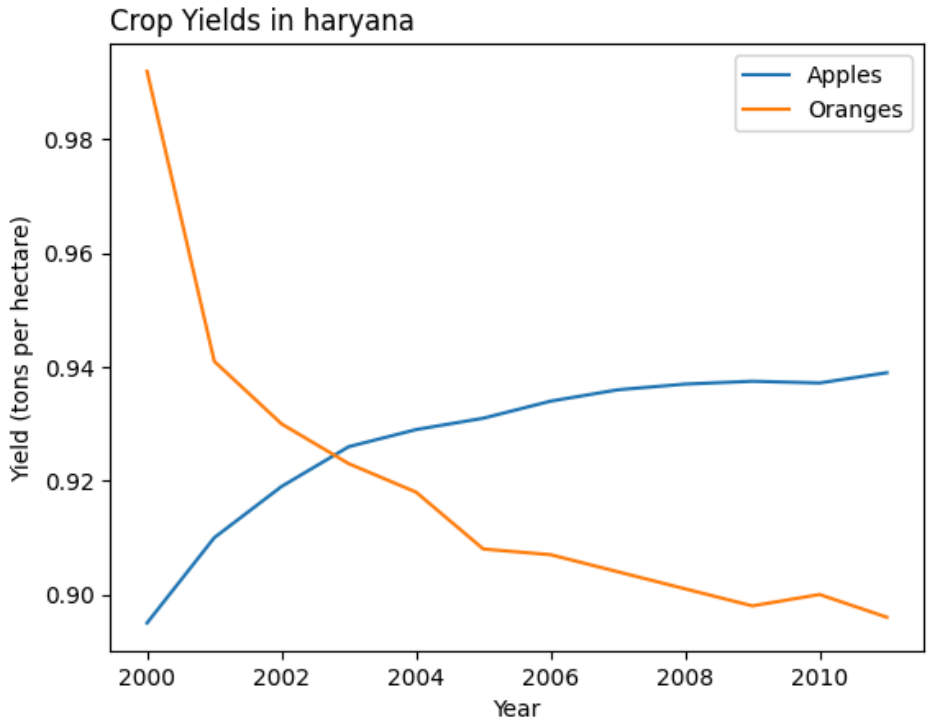
```
<matplotlib.legend.Legend at 0x7f0ce30dad10>
```



Start coding or generate with AI.

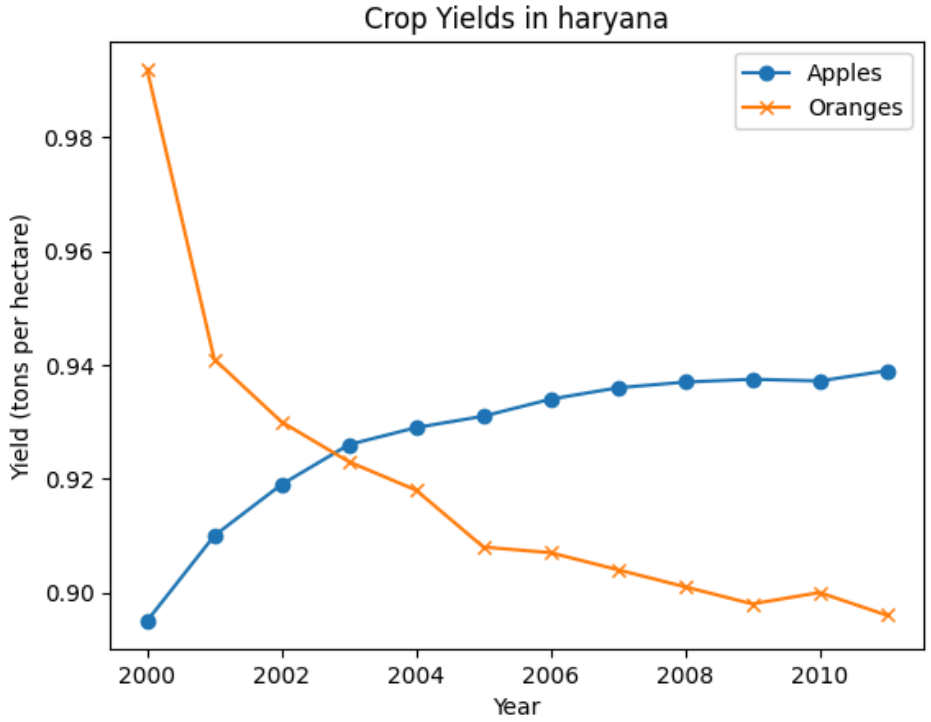
```
plt.plot(years, apples)
plt.plot(years, oranges)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in haryana", loc="left") #position left of the title
plt.legend(["Apples", "Oranges"])
```

```
<matplotlib.legend.Legend at 0x7f0ce4fc5750>
```




```
plt.plot(years, apples, marker='o')
plt.plot(years, oranges, marker='x')
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in haryana")
plt.legend(["Apples", "Oranges"])
```

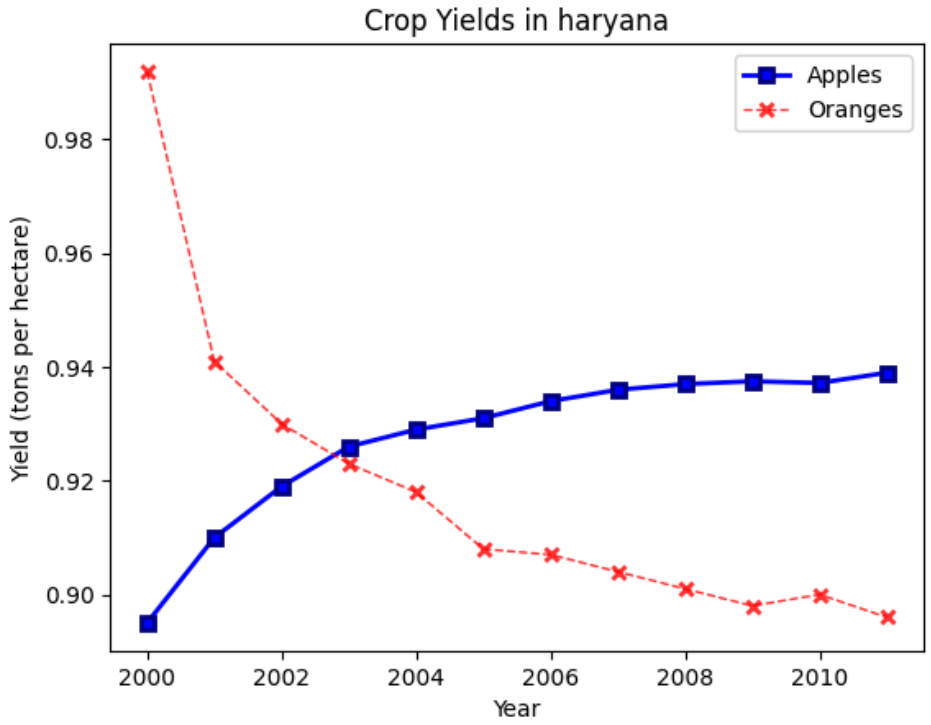
```
<matplotlib.legend.Legend at 0x7f0d258f8250>
```




```
plt.plot(years, apples, marker='s', c='b', ls='-', lw=2, ms=6, mew=2, mec='navy')
plt.plot(years, oranges, marker='x', c='r', ls='--', lw=1, ms=6, mew=2, alpha=0.8)
```

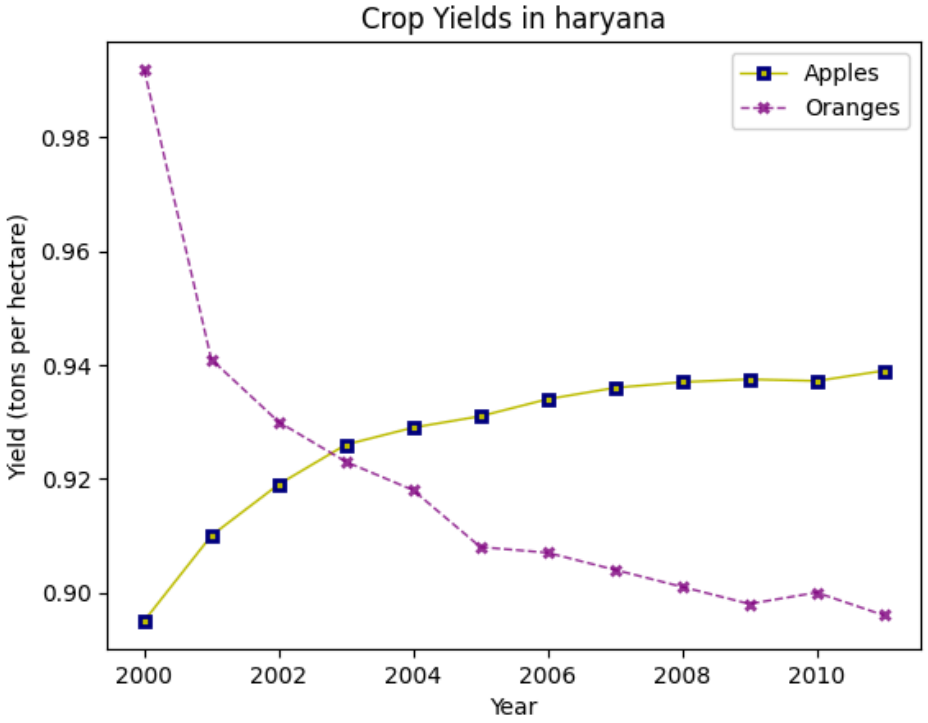
```
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in haryana")
plt.legend(["Apples", "Oranges"])
```

```
<matplotlib.legend.Legend at 0x7f0ce0df9c50>
```



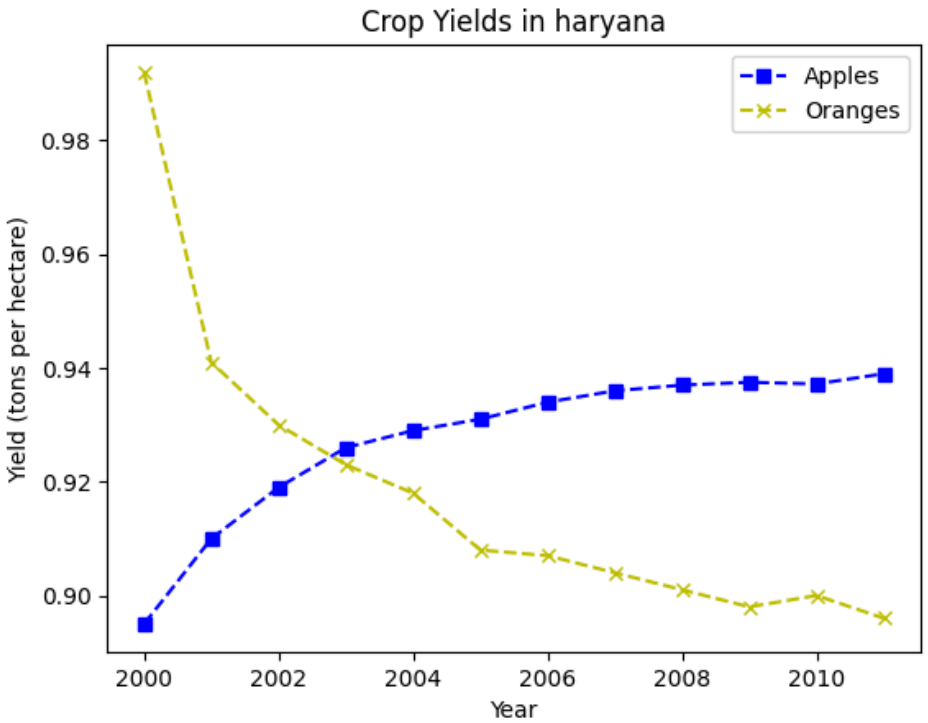

```
plt.plot(years, apples, marker='s', c='y', ls='-', lw=1, ms=5, mew=2, mec='navy')
plt.plot(years, oranges, marker='x', c='purple', ls='--', lw=1, ms=5, mew=2, alpha=0.8)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in haryana")
plt.legend(["Apples", "Oranges"])
```

```
<matplotlib.legend.Legend at 0x7f0ce0e67cd0>
```




```
plt.plot(years, apples,"s--b")
plt.plot(years, oranges, "x--y")
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in haryana")
plt.legend(["Apples", "Oranges"])
```

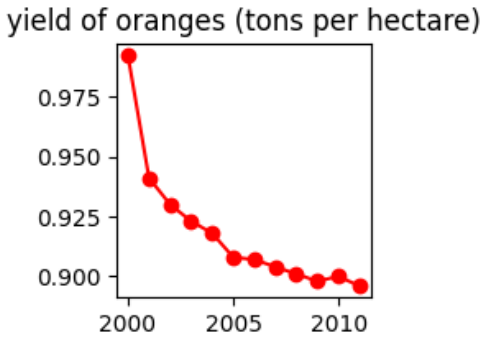
```
<matplotlib.legend.Legend at 0x7f0ce0df1250>
```



Double-click (or enter) to edit

```
plt.figure(figsize=(2,2))

plt.plot(years, oranges, 'o-r')
plt.title("yield of oranges (tons per hectare)")
```

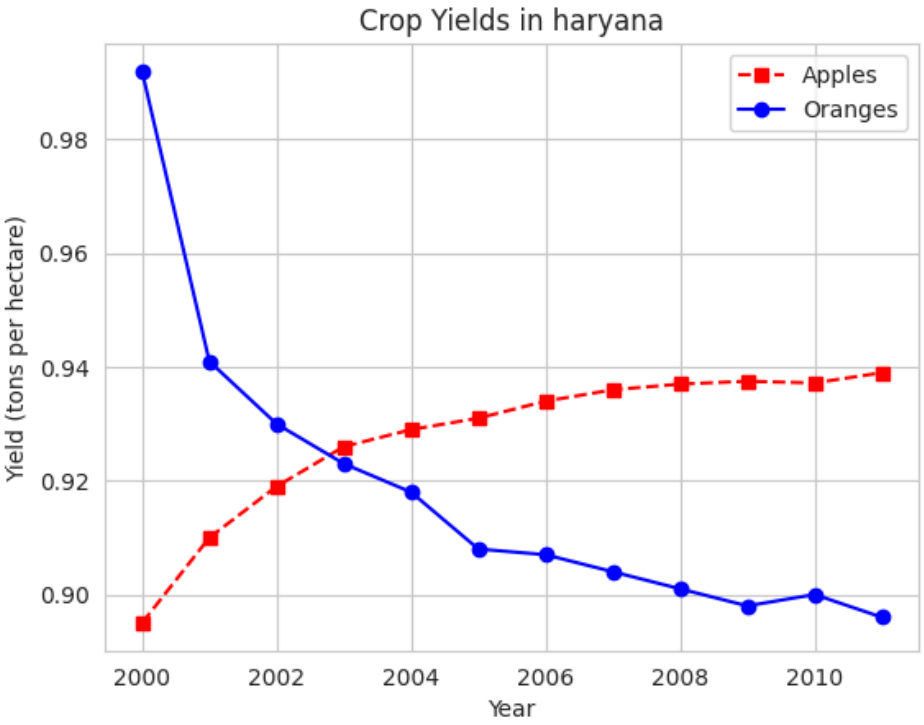
```
Text(0.5, 1.0, 'yield of oranges (tons per hectare)')
```



SEABORN

```
sns.set_style("whitegrid")
plt.plot(years, apples, 's--r')
plt.plot(years, oranges, 'o-b')
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in haryana")
plt.legend(["Apples", "Oranges"])
```

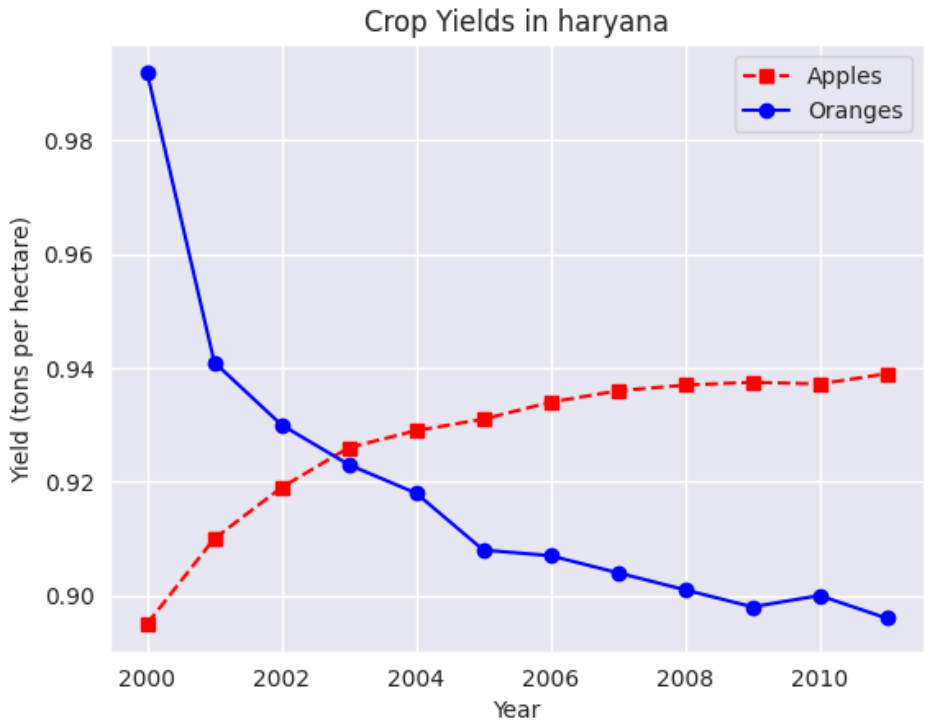
```
<matplotlib.legend.Legend at 0x7f0ce0d41050>
```




```
sns.set_style("darkgrid")
plt.plot(years, apples, 's--r')
plt.plot(years, oranges, 'o-b')
plt.xlabel('Year')
```

```
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in haryana")
plt.legend(["Apples", "Oranges"])
```

```
<matplotlib.legend.Legend at 0x7f0ce0e77d90>
```




```
# plt.rcParams['font.size'] = 20
# plt.rcParams['figure.figsize'] = (6,4)
# plt.rcParams['figure.facecolor'] = '#00000001'
# plt.plot(years, apples, 's--r')
# plt.plot(years, oranges, 'o-b')
# plt.xlabel('Year')
# plt.ylabel('Yield (tons per hectare)')
# plt.title("Crop Yields in haryana")
# plt.legend(["Apples", "Oranges"])
```

scatter plot

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

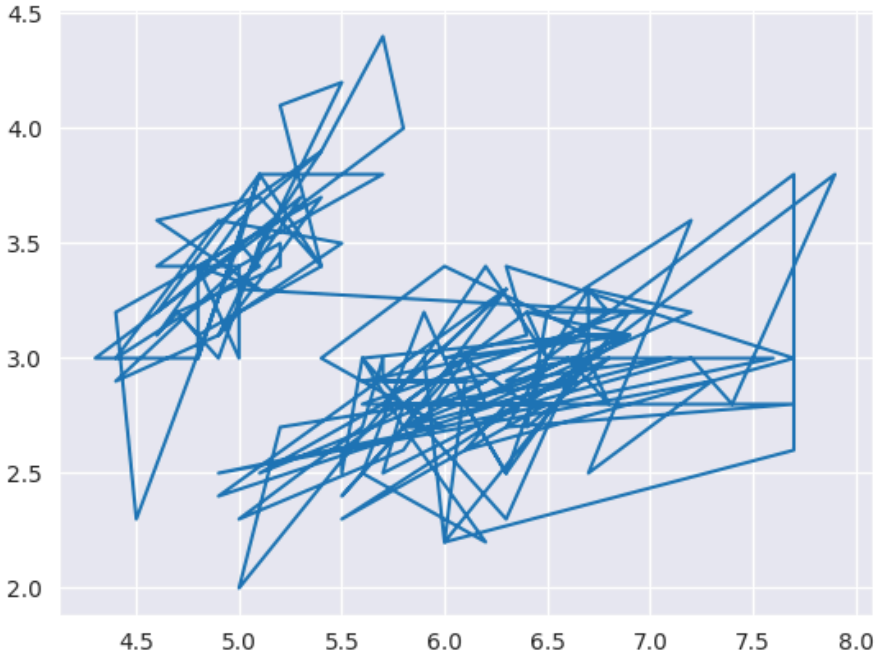
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
flower_df.species.unique()
```

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

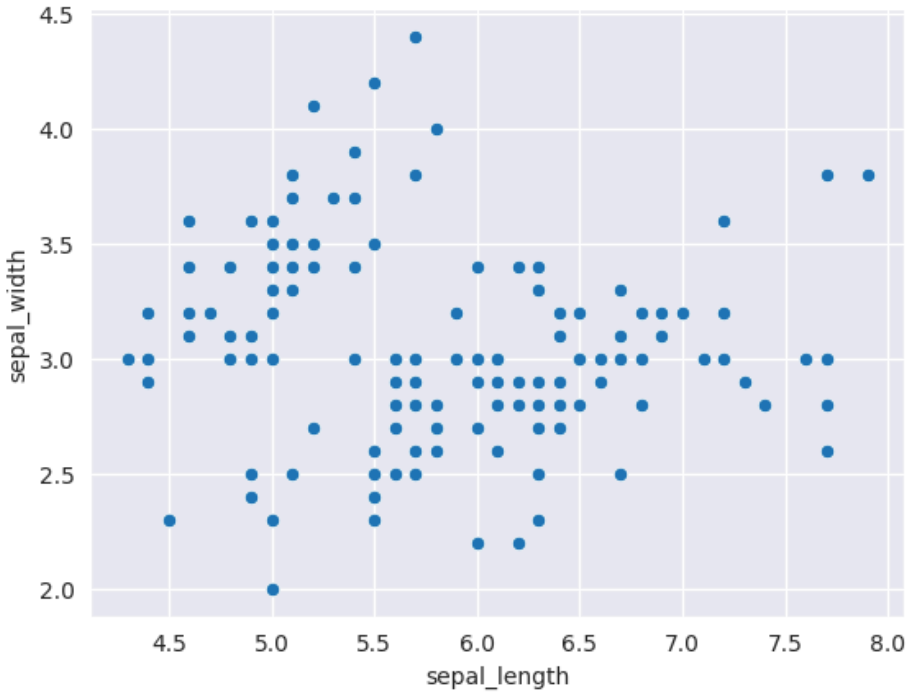
```
plt.plot(flower_df.sepal_length, flower_df.sepal_width)
```

```
[<matplotlib.lines.Line2D at 0x7f0ce0beaf50>]
```



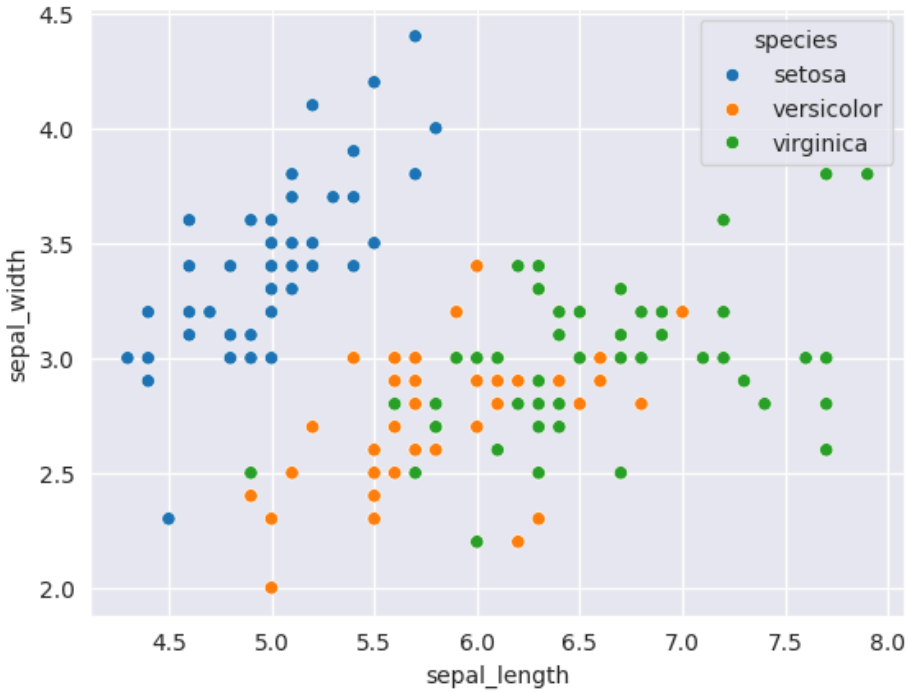

```
sns.scatterplot(x=flower_df.sepal_length, y=flower_df.sepal_width)
```

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



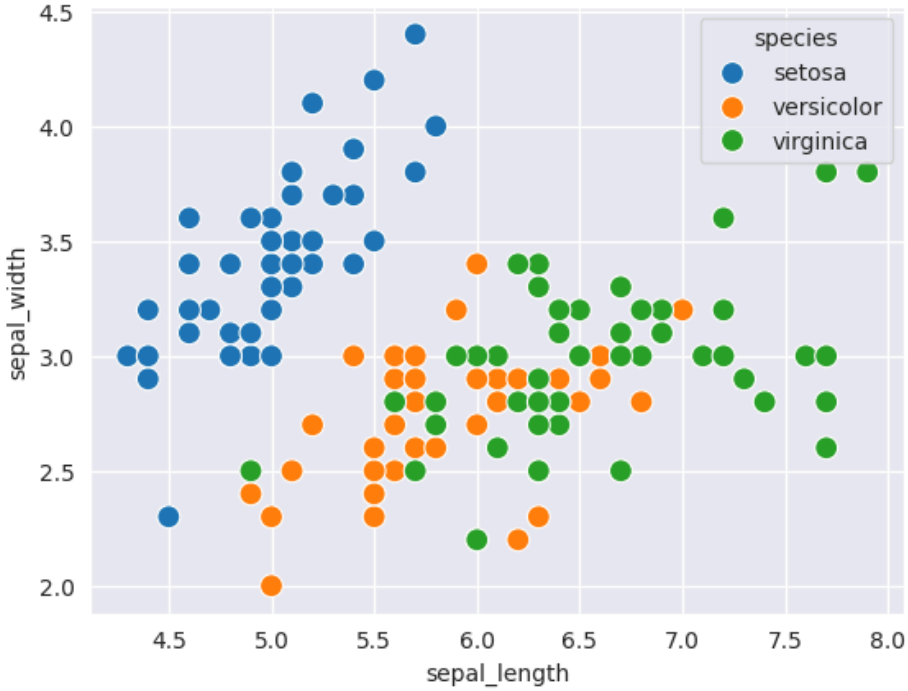

```
sns.scatterplot(x=flower_df.sepal_length, y=flower_df.sepal_width, hue=flower_df.species)
```

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



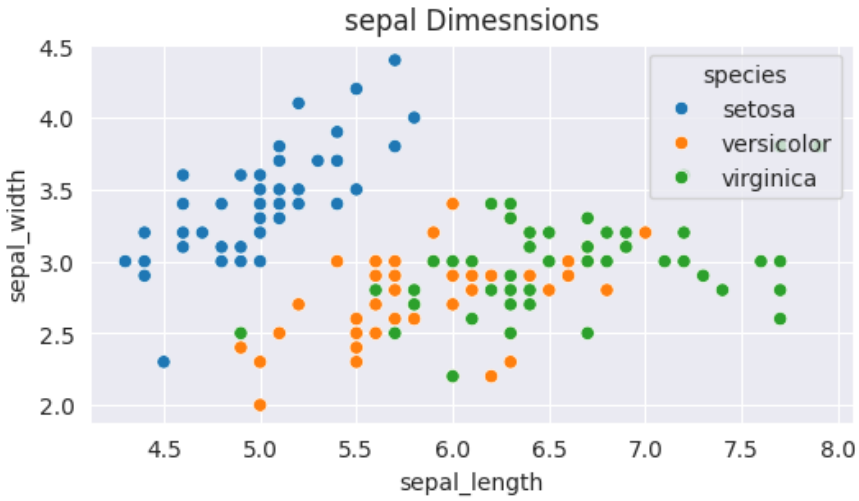

```
sns.scatterplot(x=flower_df.sepal_length, y=flower_df.sepal_width, hue=flower_df.species, s=100)
```

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




```
plt.figure(figsize=(6, 3))
plt.title('sepal Dimesnsions')
sns.scatterplot(x=flower_df.sepal_length, y=flower_df.sepal_width, hue=flower_df.species)
```

```
<Axes: title={'center': 'sepal Dimesnsions'}, xlabel='sepal_length', ylabel='sepal_width'>
```



Histogram

```
flower_df.sepal_width
```

	sepal_width
0	3.5
1	3.0
2	3.2
3	3.1
4	3.6
...	...
145	3.0
146	2.5
147	3.0
148	3.4
149	3.0

150 rows × 1 columns

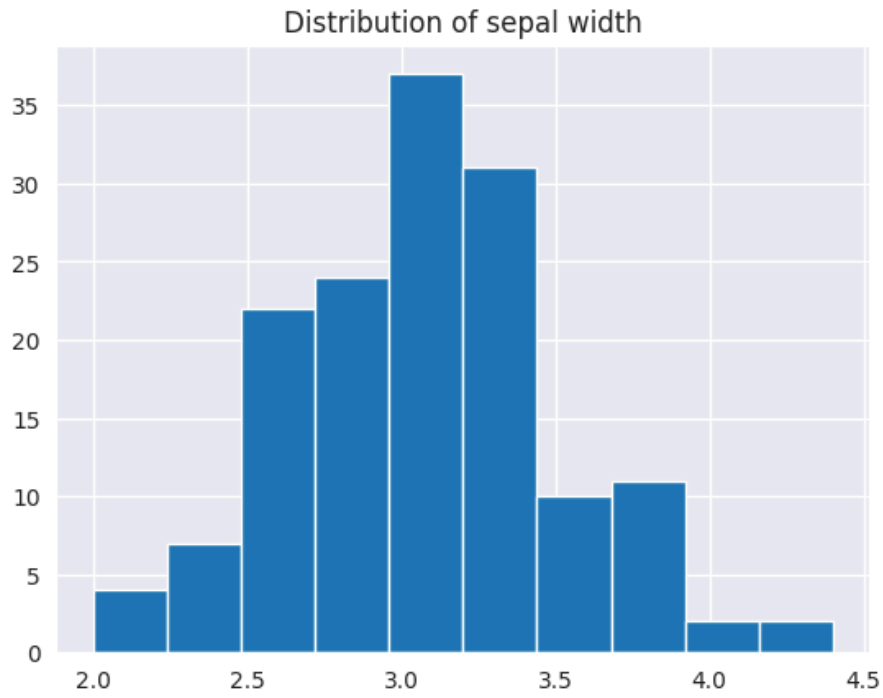
dtype: float64

```
flower_df["sepal_width"].describe().reset_index()
```

	index	sepal_width
0	count	150.000000
1	mean	3.057333
2	std	0.435866
3	min	2.000000
4	25%	2.800000
5	50%	3.000000
6	75%	3.300000
7	max	4.400000

```
plt.title("Distribution of sepal width")
plt.hist(flower_df.sepal_width)
```

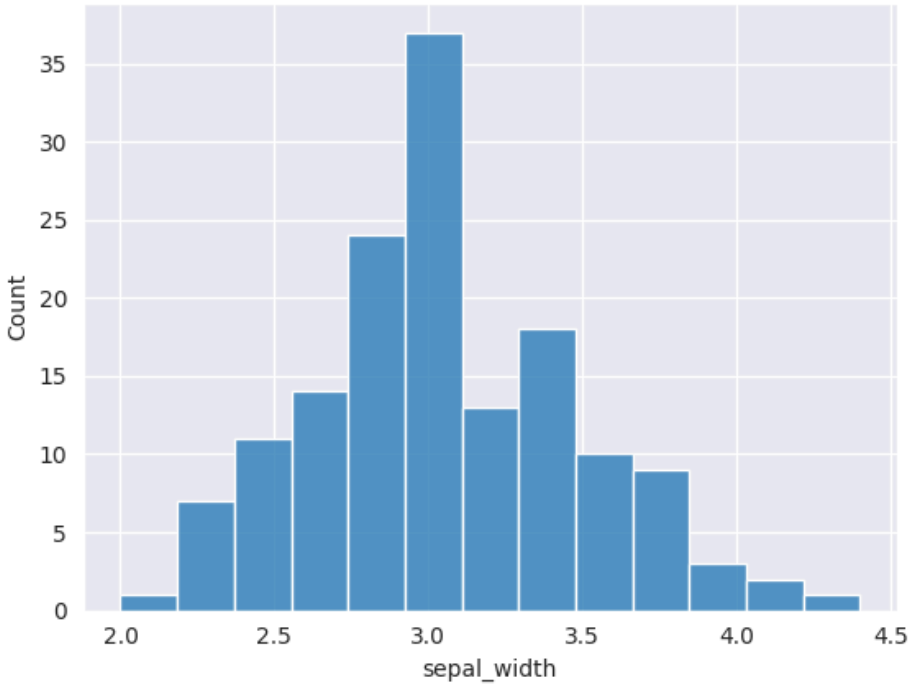
```
(array([ 4.,  7., 22., 24., 37., 31., 10., 11.,  2.,  2.]),
 array([2.  , 2.24, 2.48, 2.72, 2.96, 3.2 , 3.44, 3.68, 3.92, 4.16, 4.4 ]),
 <BarContainer object of 10 artists>)
```




```
Start coding or generate with AI.
```

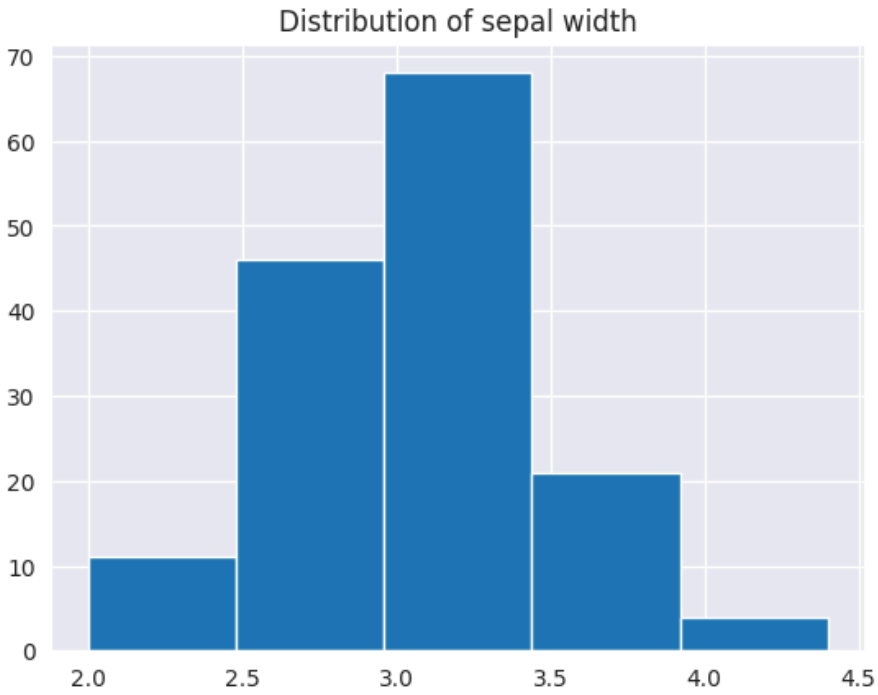
```
sns.histplot(flower_df.sepal_width)
```

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



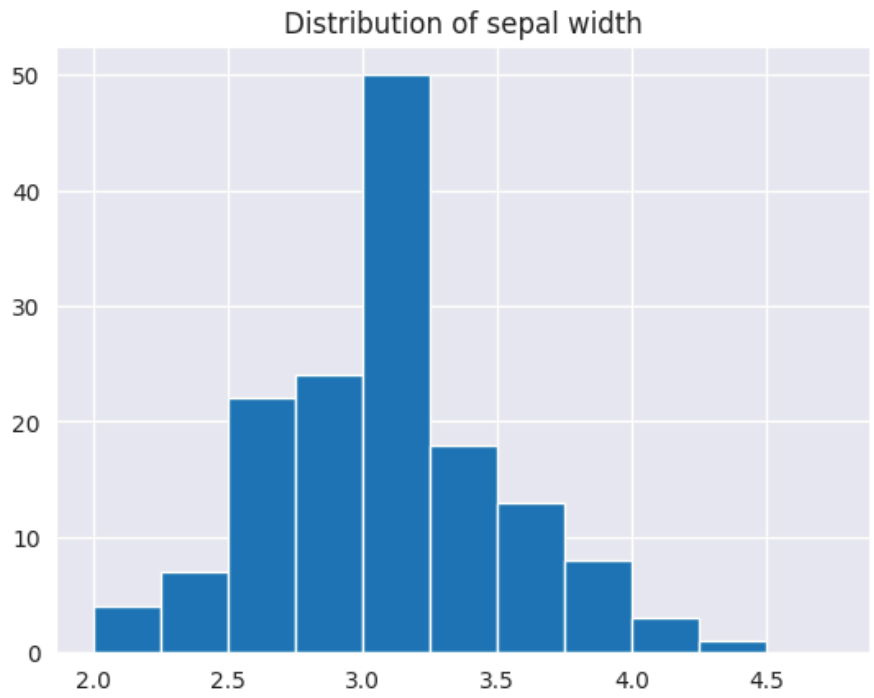

```
plt.title("Distribution of sepal width")
plt.hist(flower_df.sepal_width,bins=5)
```

```
(array([11., 46., 68., 21.,  4.]),
 array([2.  , 2.48, 2.96, 3.44, 3.92, 4.4 ]),
 <BarContainer object of 5 artists>)
```



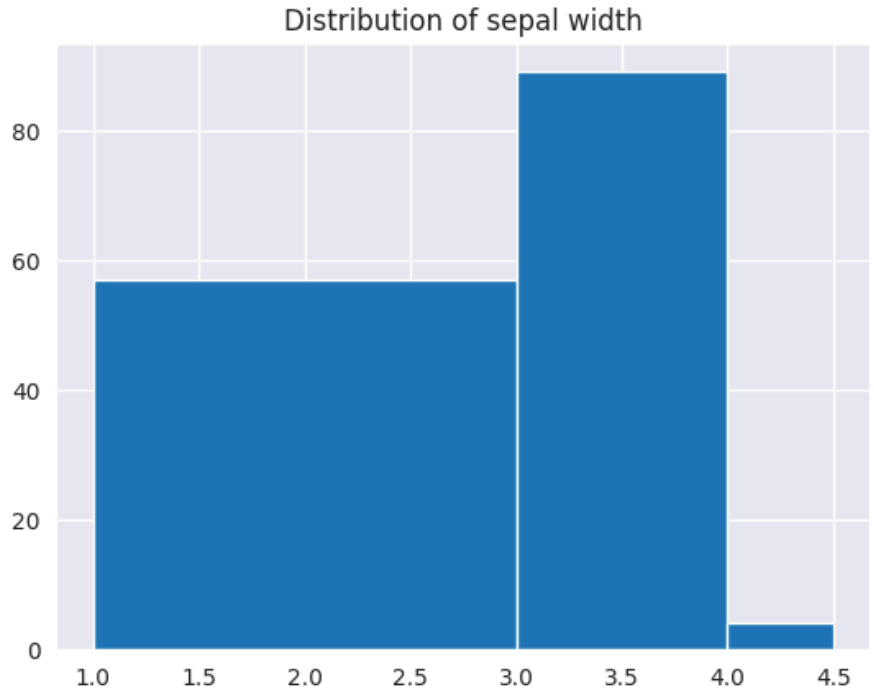

```
plt.title("Distribution of sepal width")
plt.hist(flower_df.sepal_width, bins=np.arange(2,5, 0.25))
```

```
(array([ 4.,  7., 22., 24., 50., 18., 13.,  8.,  3.,  1.,  0.]),
 array([2.  , 2.25, 2.5 , 2.75, 3.  , 3.25, 3.5 , 3.75, 4.  , 4.25, 4.5 ,
        4.75]),
 <BarContainer object of 11 artists>)
```




```
plt.title("Distribution of sepal width")
plt.hist(flower_df.sepal_width,bins=[1,3,4,4.5])
```

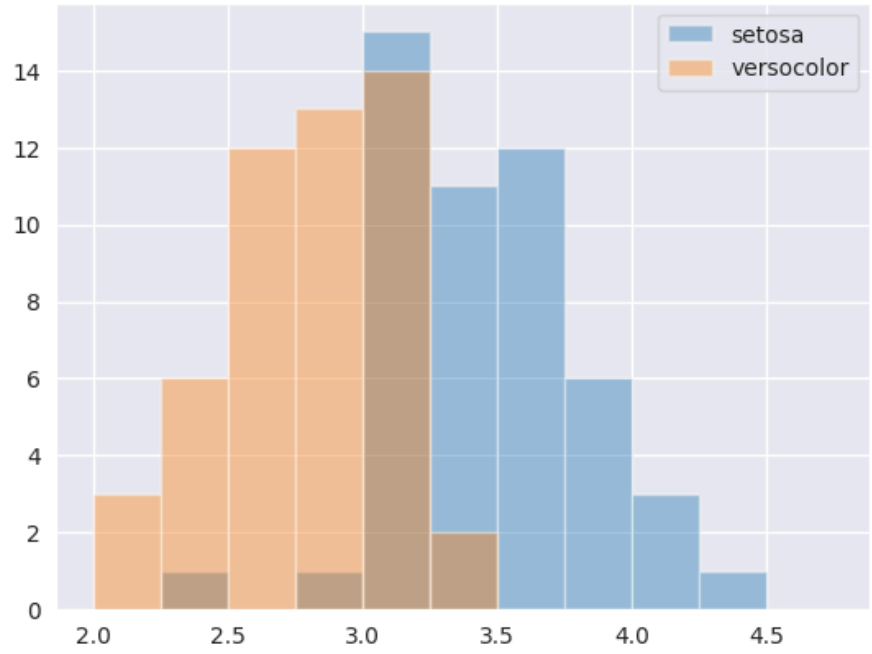
```
(array([57., 89.,  4.]),
 array([1. , 3. , 4. , 4.5]),
 <BarContainer object of 3 artists>)
```



Multiple histograms

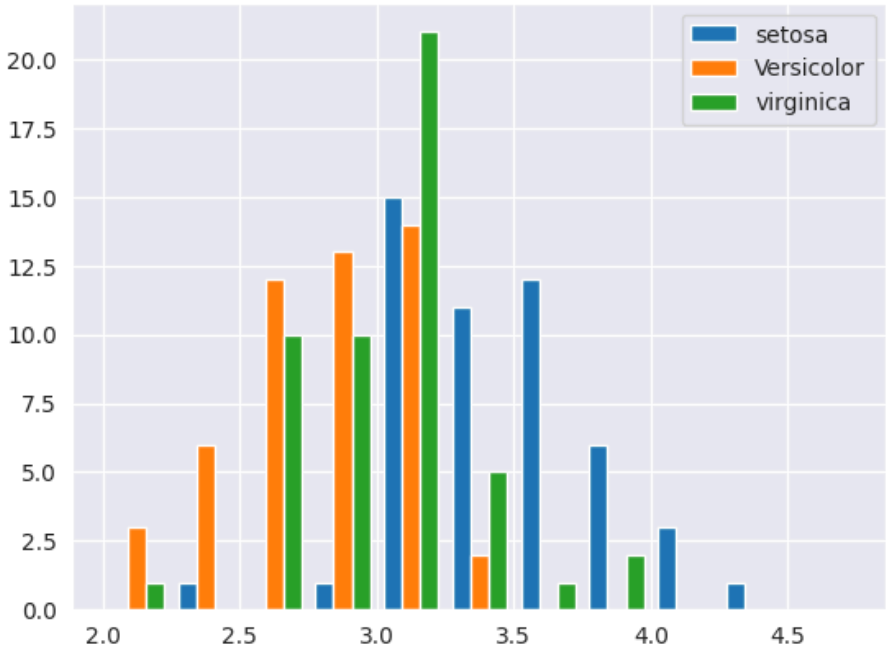
```
setosa_df = flower_df[flower_df.species == "setosa"]
versicolor_df = flower_df[flower_df.species == "versicolor"]
virginica_df = flower_df[flower_df.species == "virginica"]
# print(setosa_df)
# print(versicolor_df)
# print(virginica_df)


plt.hist(setosa_df.sepal_width, alpha=0.4, bins=np.arange(2,5,0.25))
plt.hist(versicolor_df.sepal_width, alpha=0.4, bins=np.arange(2,5,0.25))
plt.legend(["setosa", "versocolor"])
```

```
<matplotlib.legend.Legend at 0x7f0cdf508a90>
```



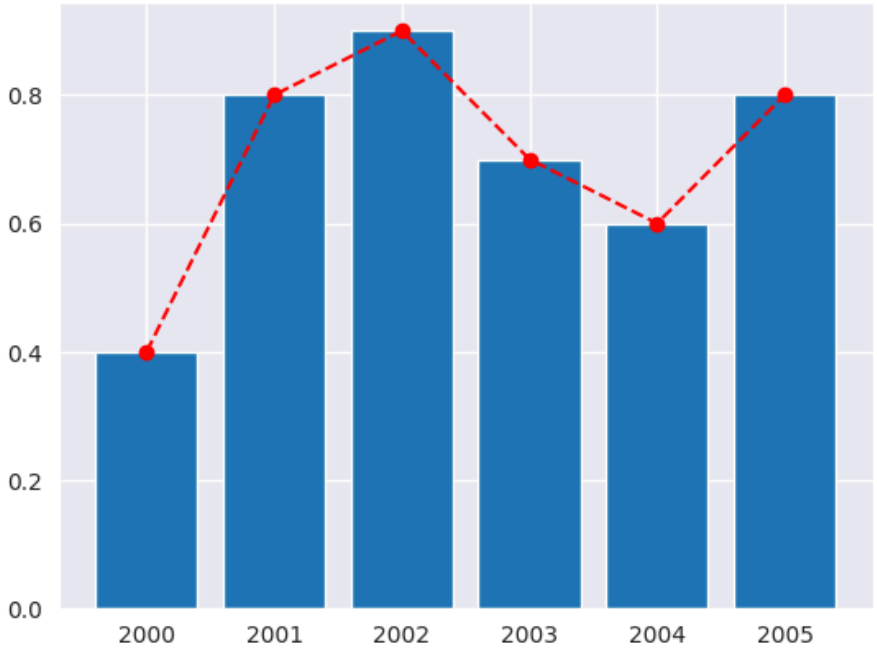

```
plt.hist([setosa_df.sepal_width,
         versicolor_df.sepal_width,
         virginica_df.sepal_width],
         bins = np.arange(2,5,0.25),
         stacked = False)
plt.legend(['setosa', 'Versicolor', 'virginica'])
```

```
<matplotlib.legend.Legend at 0x7f0cdf569f10>
```



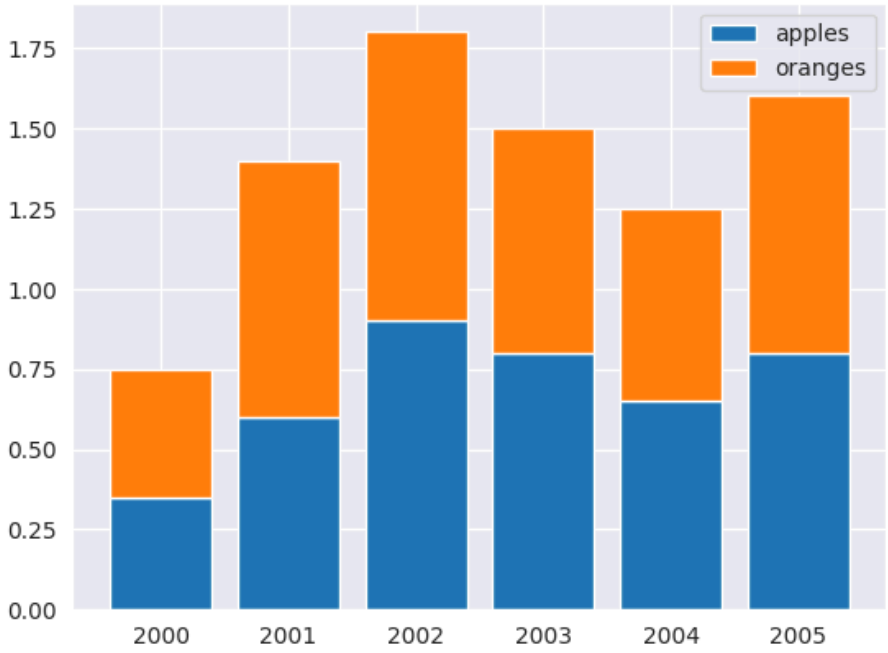

```
years = range(2000, 2006)
apples = [0.35, 0.6, 0.9, 0.8, 0.65, 0.8]
oranges = [0.4, 0.8, 0.9, 0.7, 0.6, 0.8]

plt.bar(years, oranges)
plt.plot(years, oranges, 'o--r')
```

```
[<matplotlib.lines.Line2D at 0x7f0cdf5c56d0>]
```




```
plt.bar(years, apples)
plt.bar(years, oranges, bottom=apples)
plt.legend(["apples", "oranges"])
```

```
<matplotlib.legend.Legend at 0x7f0cdf450890>
```




```
tips_df = sns.load_dataset("tips")
tips_df
```

	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
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

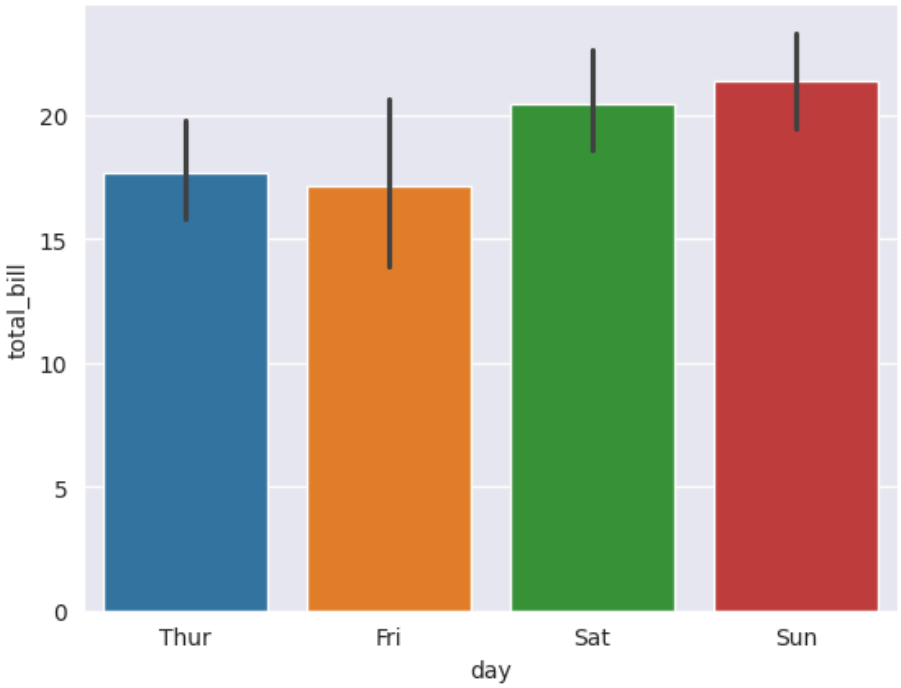
244 rows × 7 columns

```
tips_df.day.unique()
```

```
['Sun', 'Sat', 'Thur', 'Fri']
Categories (4, object): ['Thur', 'Fri', 'Sat', 'Sun']
```

```
sns.barplot(x='day', y='total_bill', data=tips_df, hue='day')
```

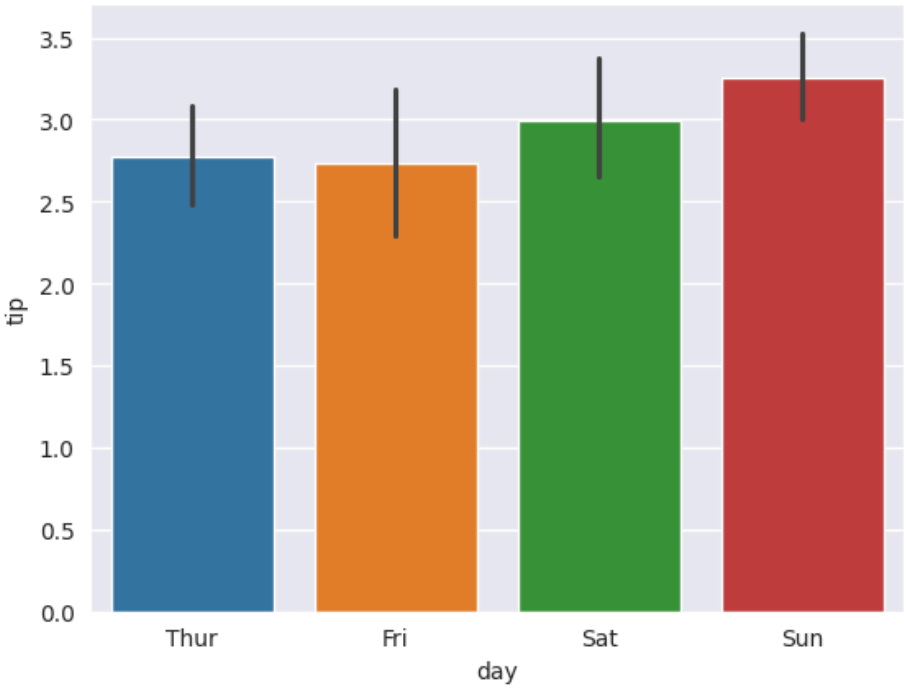
```
<Axes: xlabel='day', ylabel='total_bill'>
```




```
Start coding or generate with AI.
```

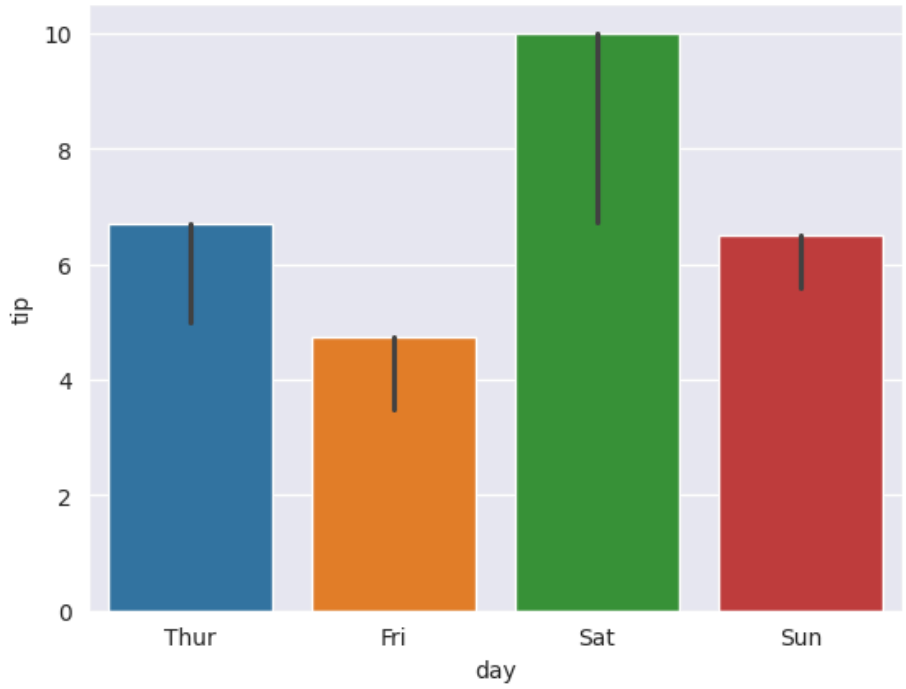
```
sns.barplot(x='day', y='tip', data=tips_df, hue='day')
```

```
<Axes: xlabel='day', ylabel='tip'>
```



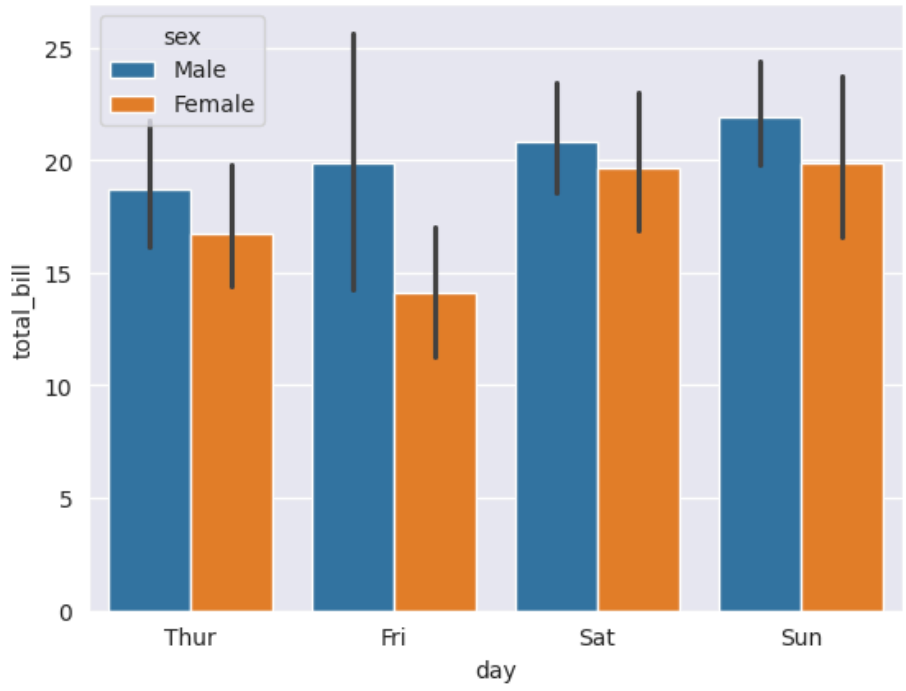

```
sns.barplot(x='day', y='tip', data=tips_df, hue='day',estimator="max")
```

```
<Axes: xlabel='day', ylabel='tip'>
```



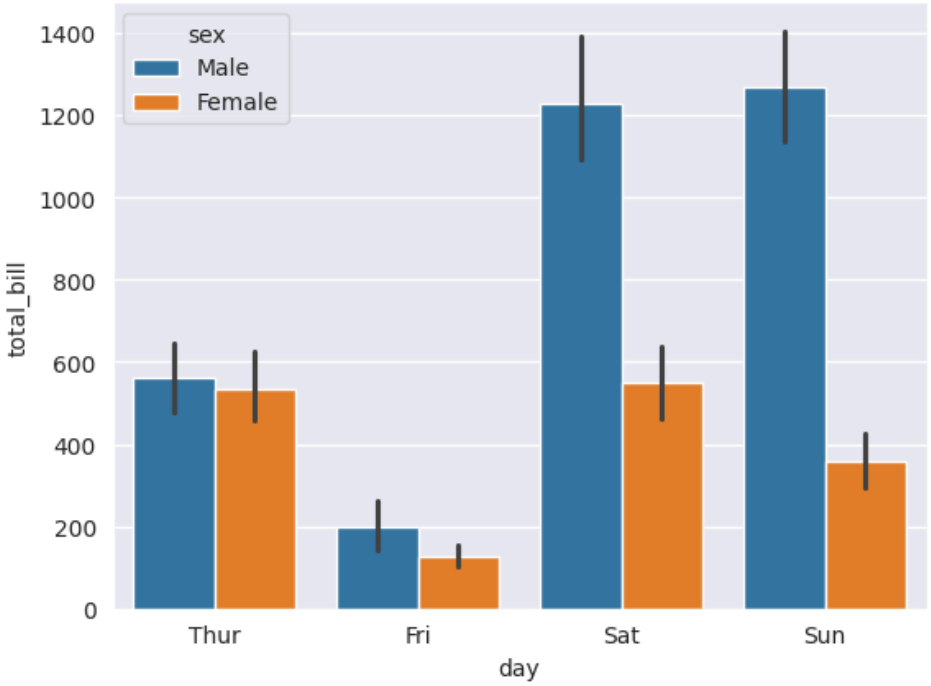

```
sns.barplot(x='day', y='total_bill', data=tips_df, hue='sex')
```

```
<Axes: xlabel='day', ylabel='total_bill'>
```




```
sns.barplot(x='day', y='total_bill', data=tips_df, hue='sex',estimator="sum")
```

```
<Axes: xlabel='day', ylabel='total_bill'>
```

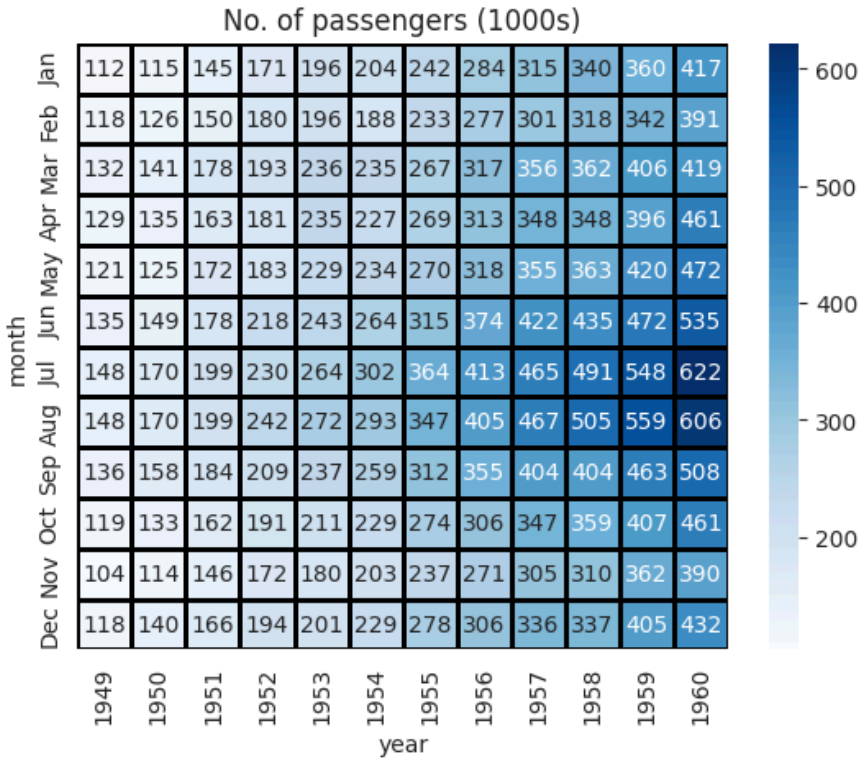


HEATMAP

1. it provides info of data wrt x and y axis like how many passengers in every year in every month

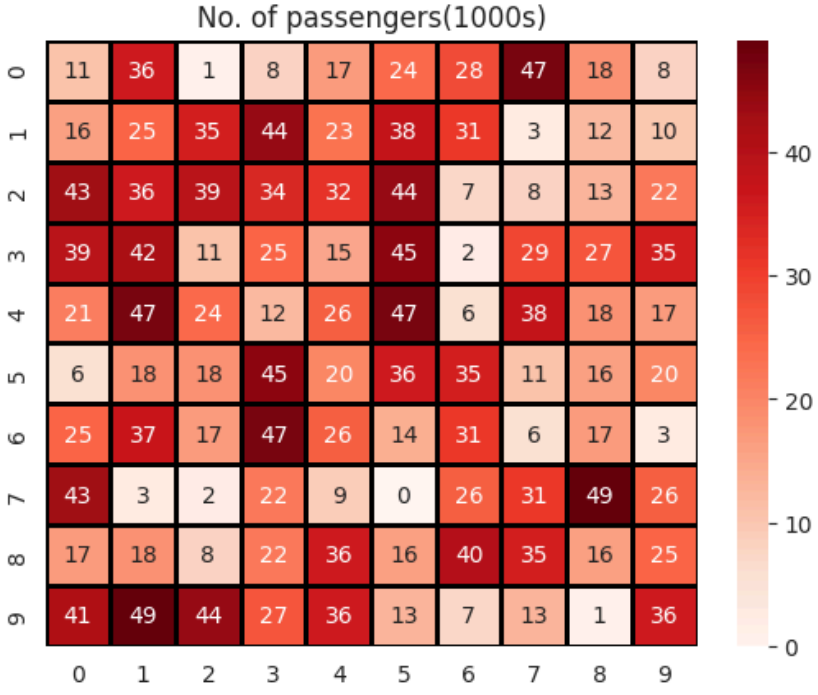
```
df=sns.load_dataset("flights").pivot(index="month", columns="year", values="passengers")
# print(df)
plt.title("No. of passengers (1000s)")
sns.heatmap(df, fmt='d',linecolor='black',linewidth=2, annot=True, cmap='Blues')
```

```
<Axes: title={'center': 'No. of passengers (1000s)'}, xlabel='year', ylabel='month'>
```




```
plt.title("No. of passengers(1000s)")
sns.heatmap(np.random.randint(0,50, size=(10,10)), fmt="d", linecolor="black", linewidth=2,annot=True, cmap="Reds
```

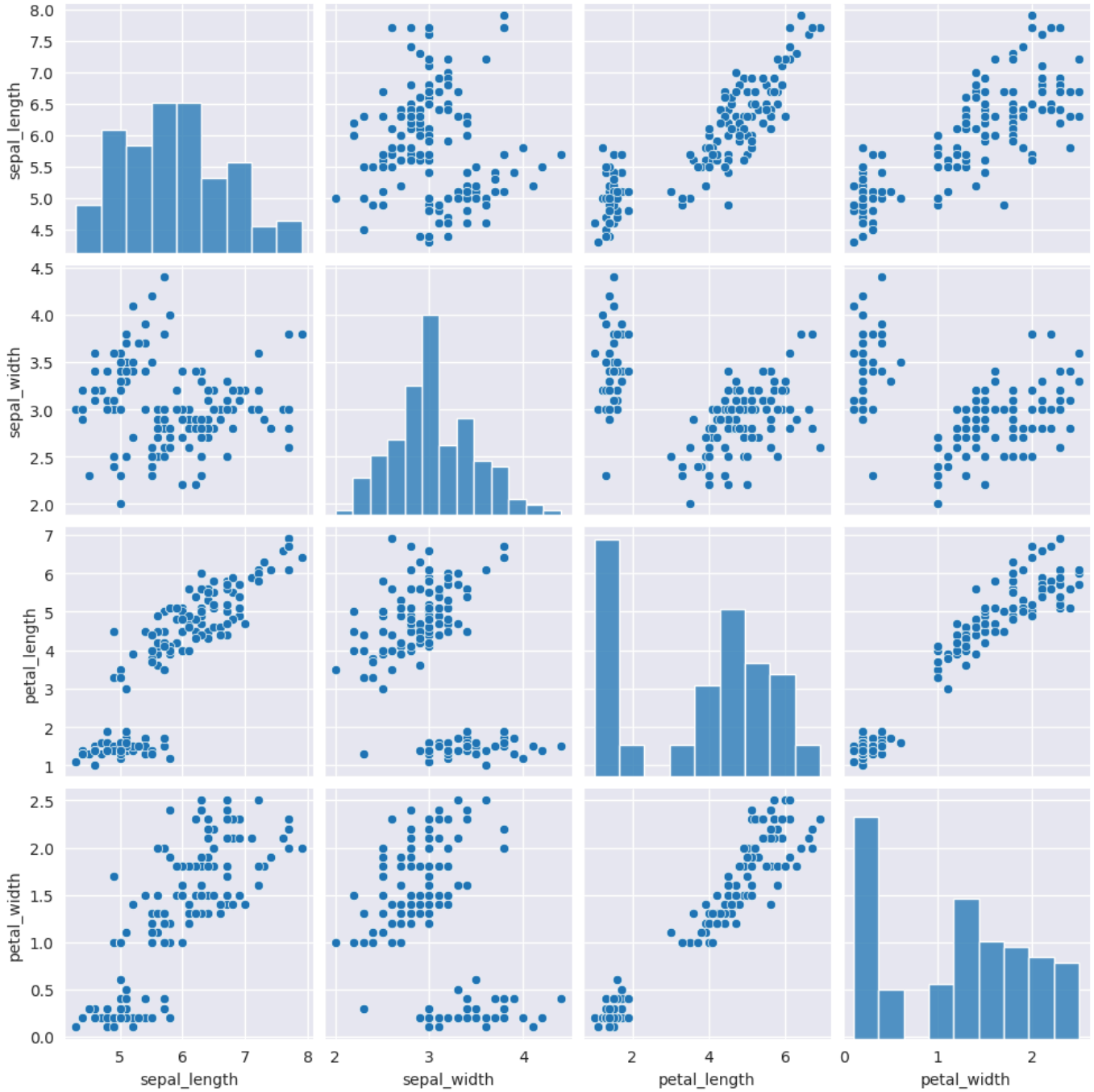
```
<Axes: title={'center': 'No. of passengers(1000s)'}>
```



Pair plots with seaborn

```
sns.pairplot(flower_df)
```

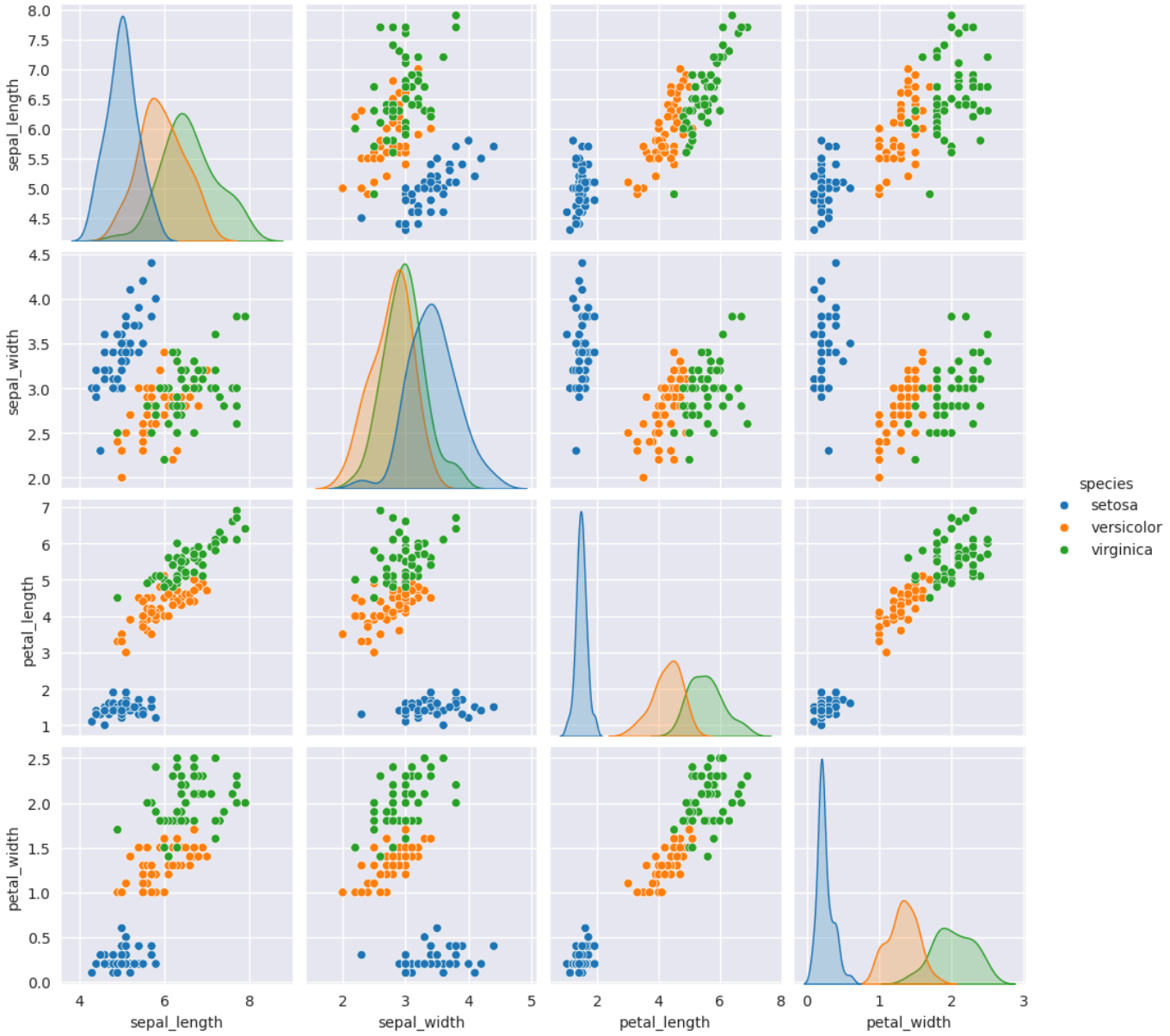
```
<seaborn.axisgrid.PairGrid at 0x7f0cda2df610>
```




```
Start coding or generate with AI.
```

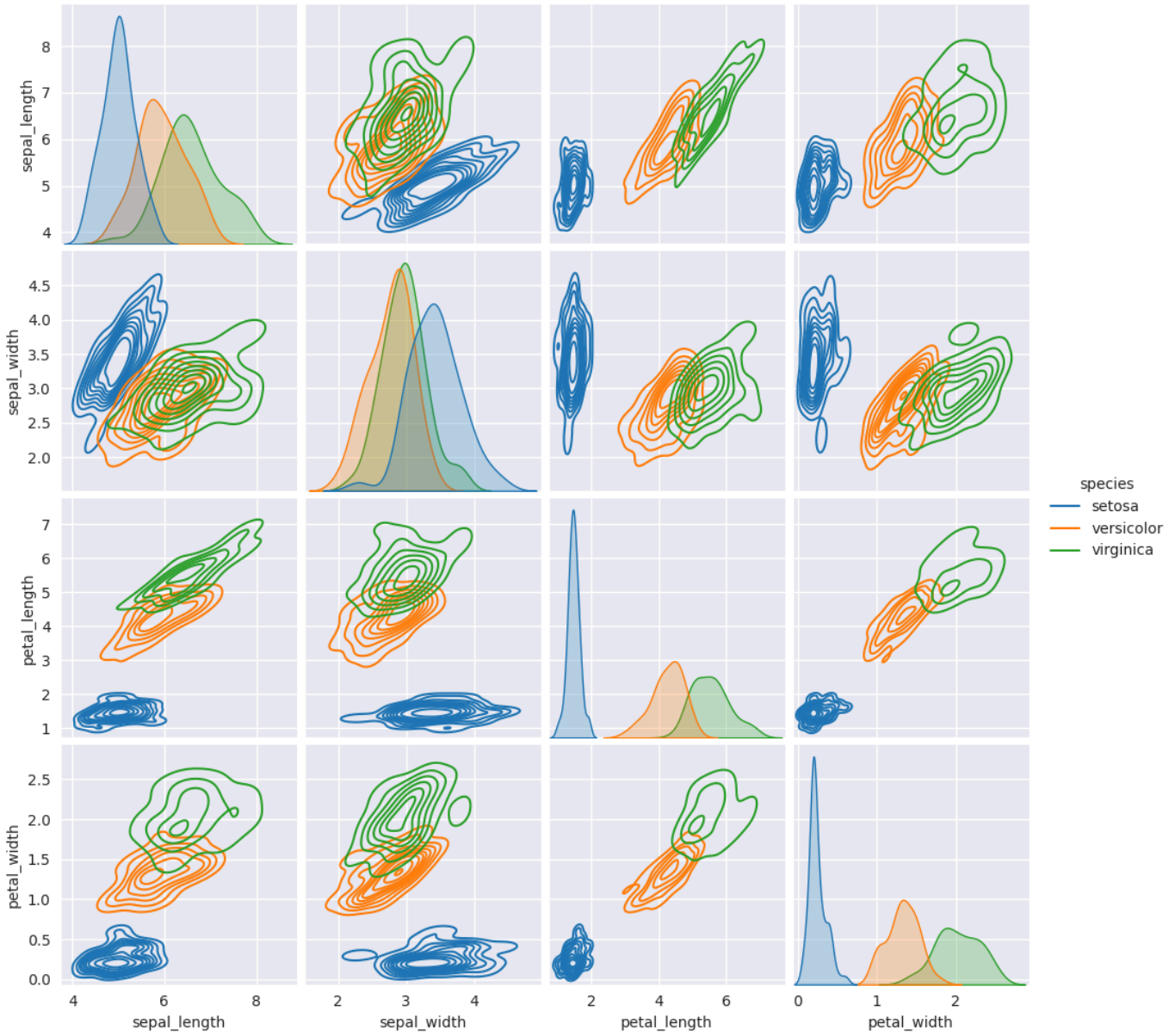
```
sns.pairplot(flower_df,hue="species")
```

```
<seaborn.axisgrid.PairGrid at 0x7f0cd594e750>
```



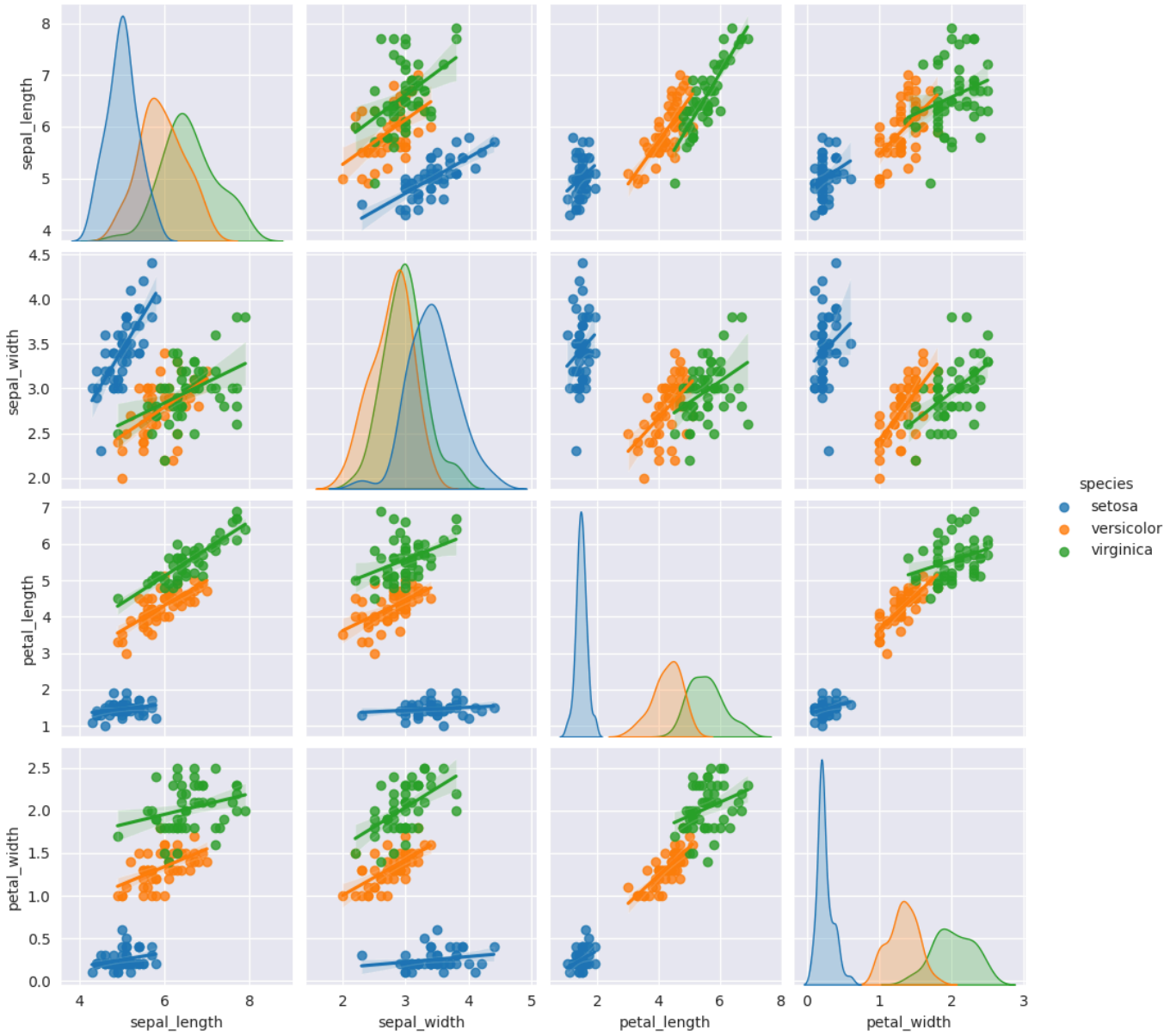

```
sns.pairplot(flower_df,hue="species",kind="kde")
```

```
<seaborn.axisgrid.PairGrid at 0x7f0cd52c85d0>
```



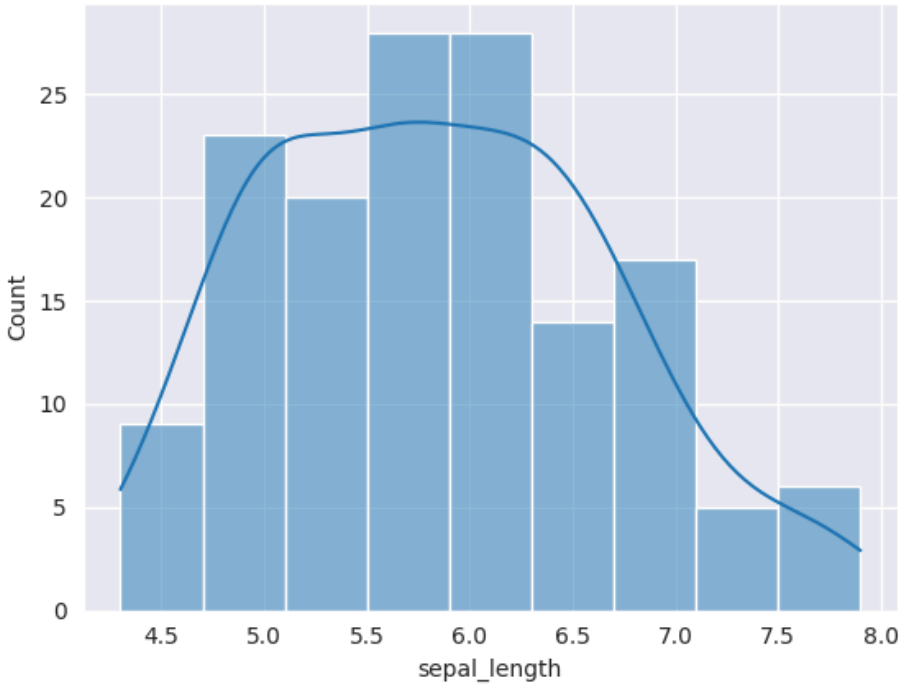

```
sns.pairplot(flower_df,hue="species",kind="reg")
```

```
<seaborn.axisgrid.PairGrid at 0x7f0cd490c610>
```



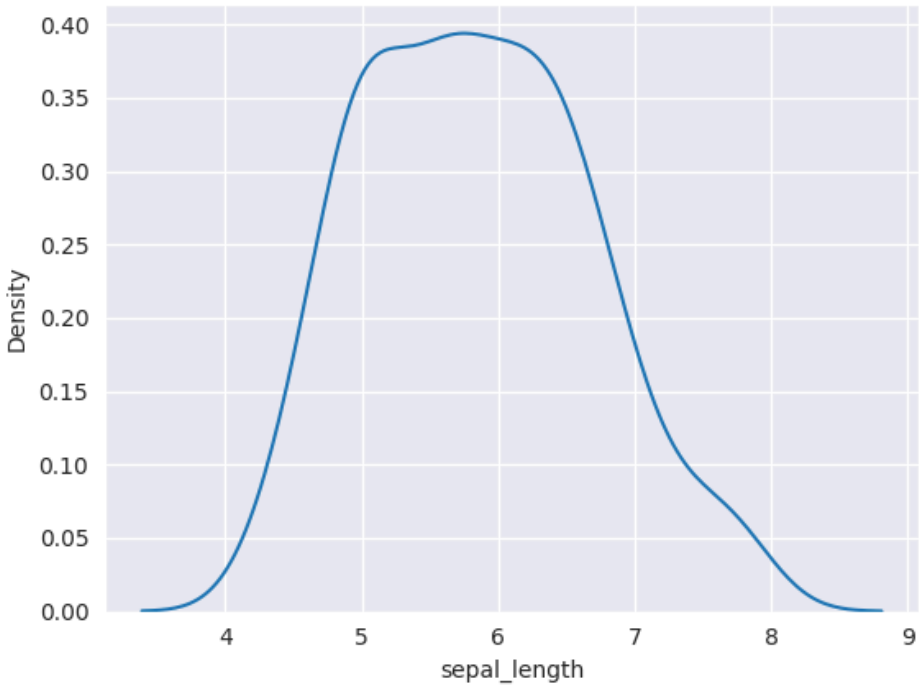

```
sns.histplot(flower_df.sepal_length, kde=True)
```

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



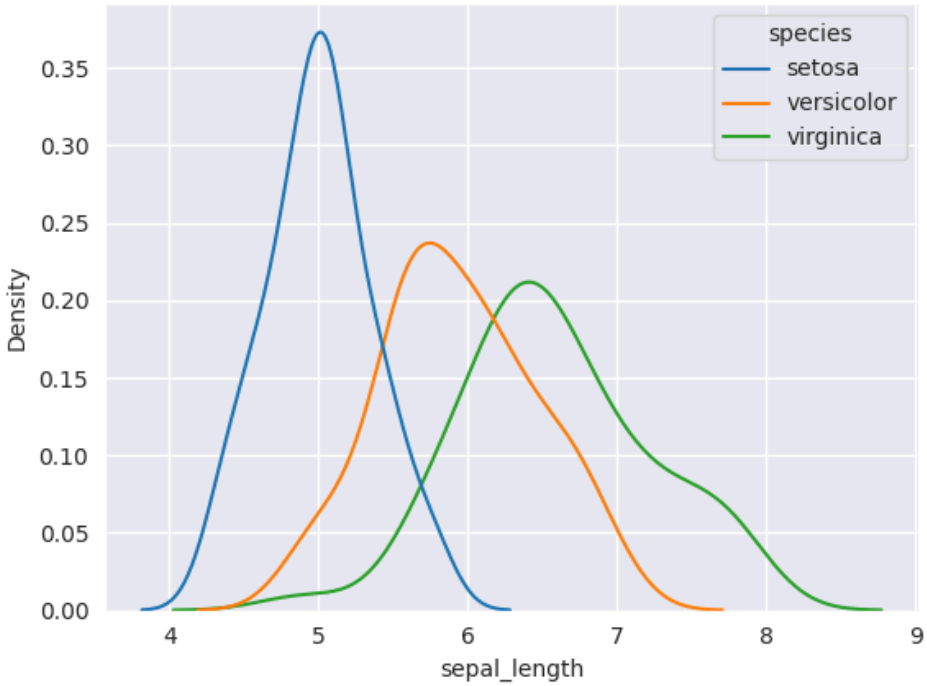

```
sns.kdeplot(data=flower_df.sepal_length)
```

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



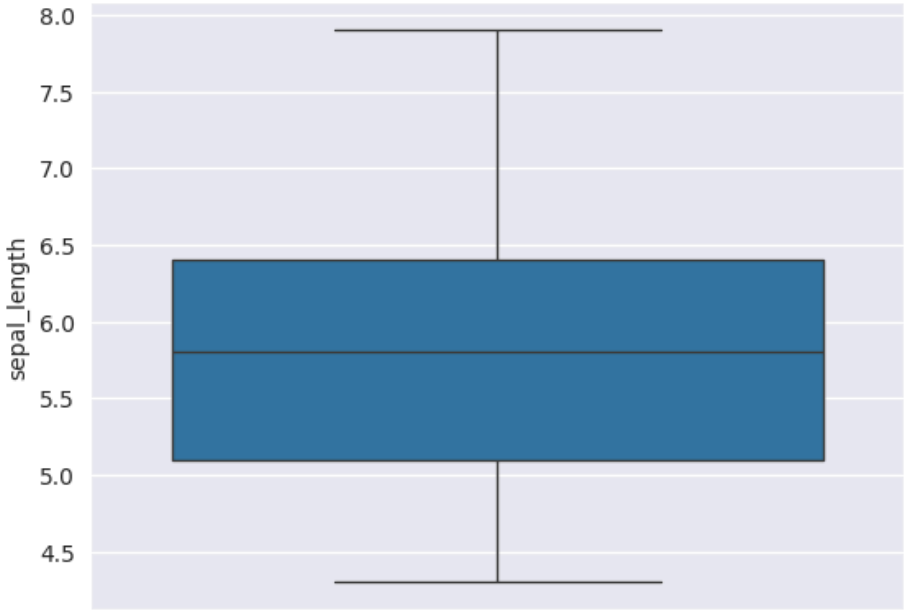

```
sns.kdeplot(x="sepal_length", data=flower_df, hue="species")
```

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

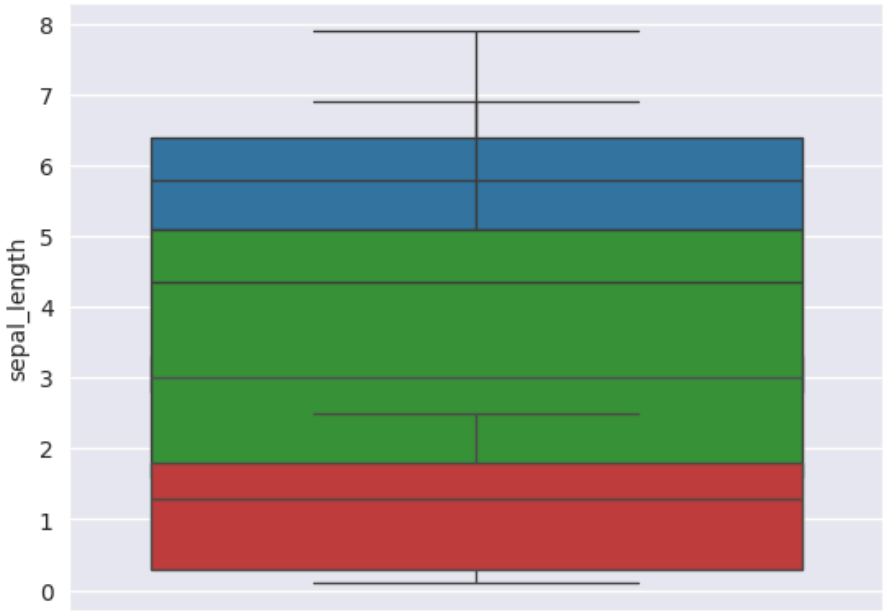



```
sns.boxplot(flower_df.sepal_length)
```

```
<Axes: ylabel='sepal_length'>
```

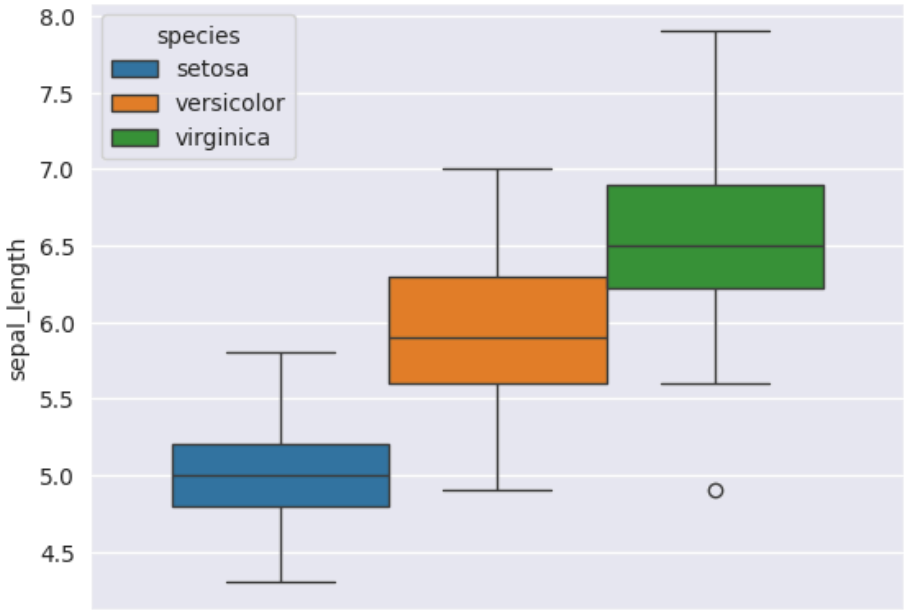



```
for i in flower_df.columns :
  if i != "species":
    sns.boxplot(flower_df[i])
```




```
sns.boxplot(y="sepal_length",data=flower_df,hue="species")
```

```
<Axes: ylabel='sepal_length'>
```



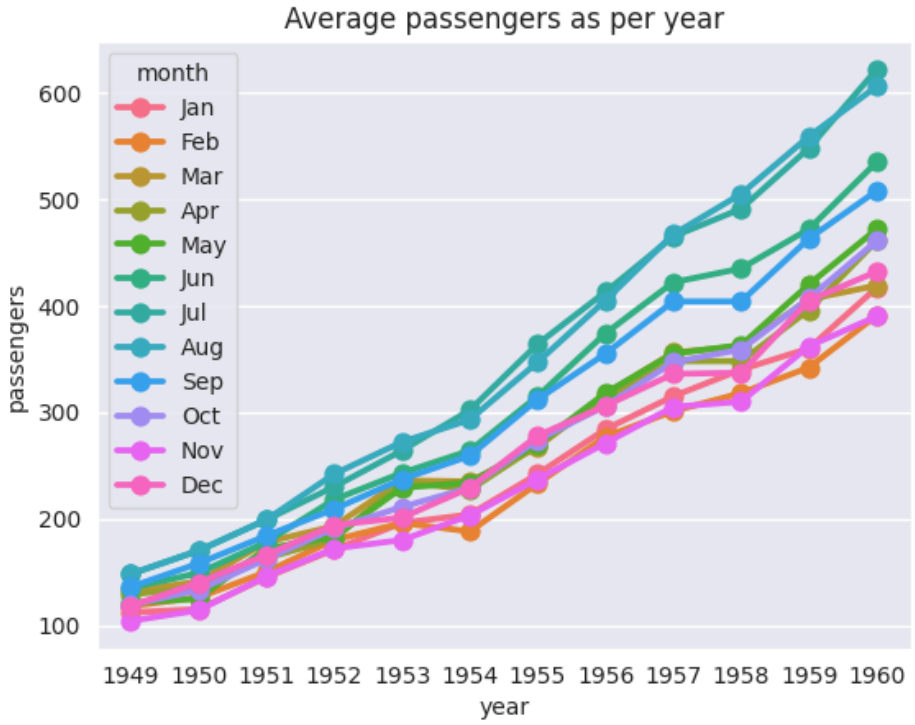
Point plot

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

```
plt.title("Average passengers as per year")
```

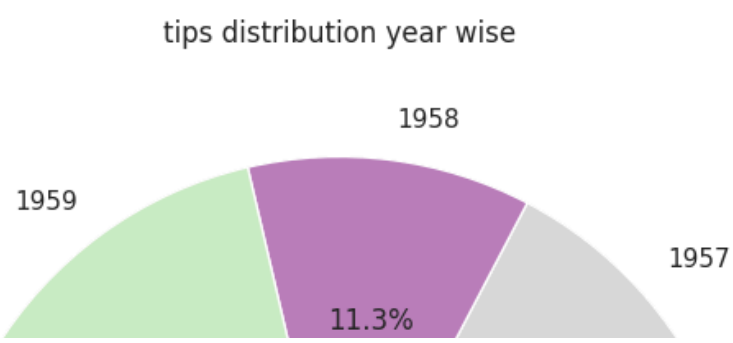
```
sns.pointplot(x="year", data=df, y="passengers", hue="month")
```

```
<Axes: title={'center': 'Average passengers as per year'}, xlabel='year', ylabel='passengers'>
```



pie chart

```
sns.set_theme(style="darkgrid")

colors = sns.color_palette('Set3', 12)
plt.figure(figsize=(8,8))
plt.pie(df.groupby("year")["passengers"].mean().astype("int"),
        labels=df.year.unique(),
        colors = colors, autopct="%1.1f%%",
        startangle=220, wedgeprops={"width" : 0.6})
plt.title("tips distribution year wise")
plt.show()
```


tips distribution year wise
1958
1959
1957
11.3%