Colab Link: https://colab.research.google.com/drive/1RqDs5J208uIifDHUvfKiw8A_r6JouVG0?usp=sharing

```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: matplotlib>=2.2 in /usr/local/lib/python3.7/dis
Requirement already satisfied: scipy>=1.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: pandas>=0.23 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/d
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /us
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/d
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packa
```

```
import seaborn as sns
```

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

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

```
type(iris)
```

```
pandas.core.frame.DataFrame
```

```
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


```
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
```

```
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
iris.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

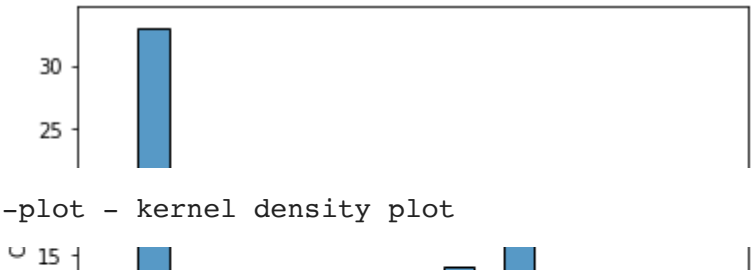
```
iris.describe(include="object")
```

	species
count	150
unique	3
top	setosa
freq	50

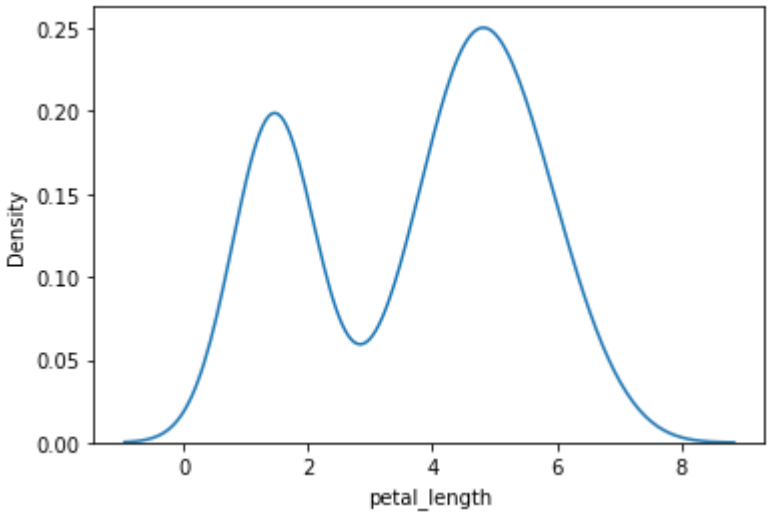
```
iris["species"].unique()
```

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

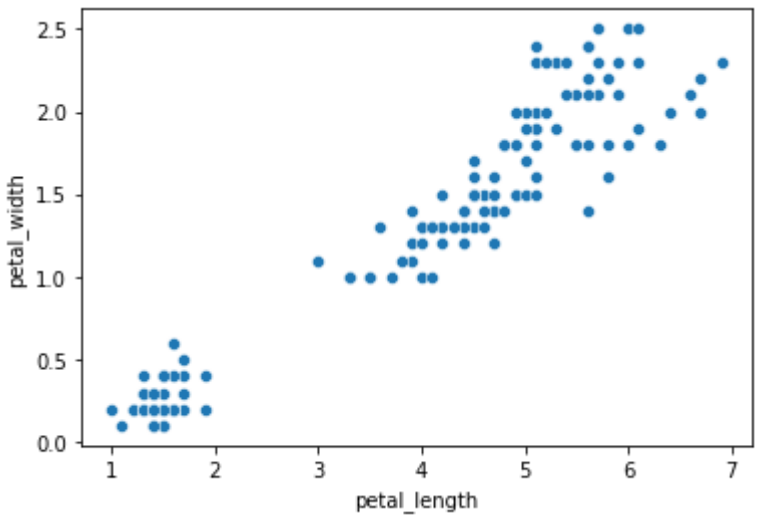
```
sns.histplot(iris["petal_length"], bins=20)
plt.show()
```



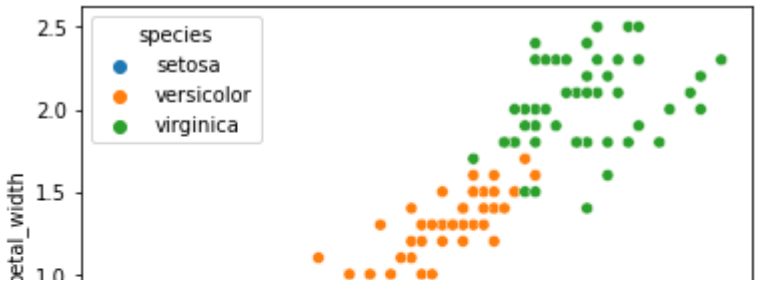

```
# kde-plot - kernel density plot
sns.kdeplot(iris["petal_length"])
plt.show()
```



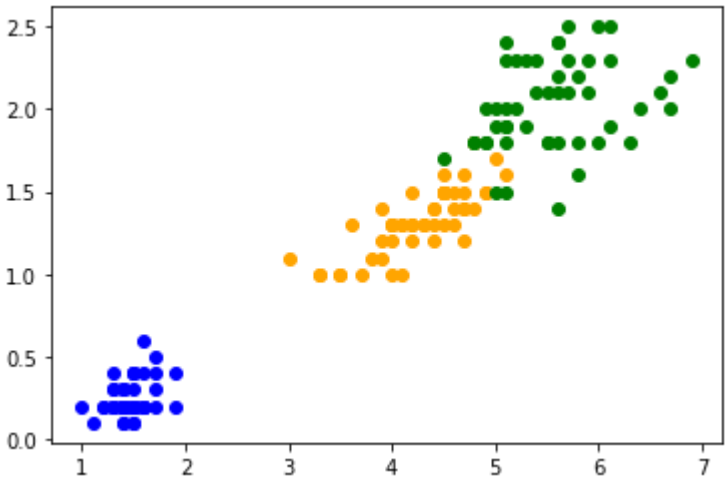

```
sns.scatterplot(x=iris["petal_length"],y=iris["petal_width"])
plt.show()
```



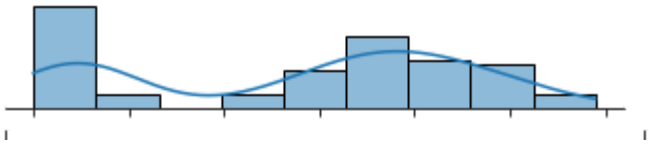

```
y="petal_width", data=iris, hue="species")
```



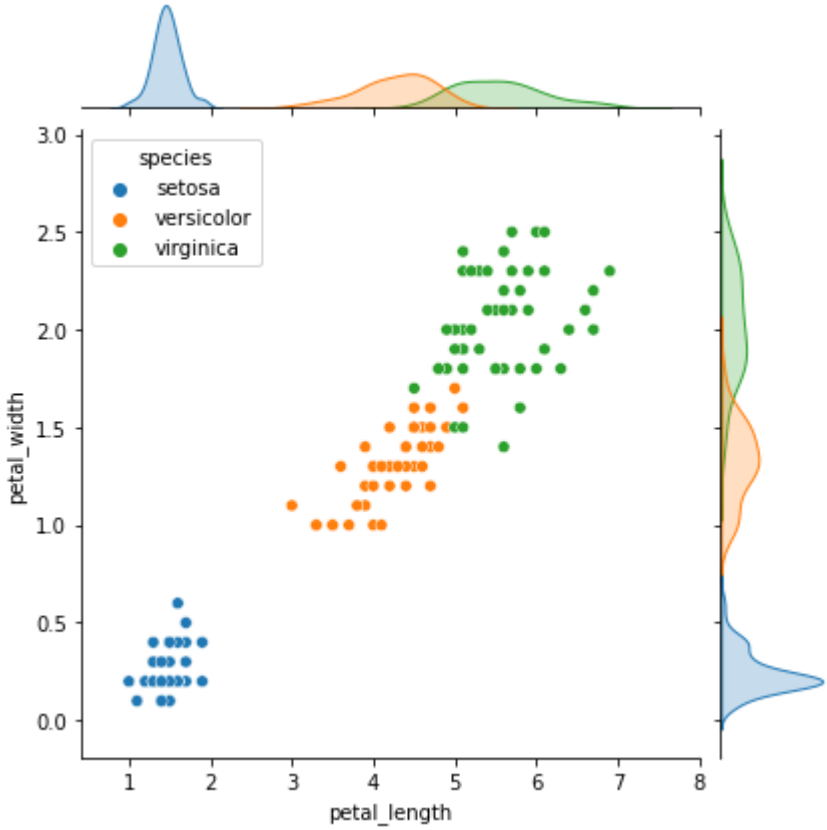

```
setosa = iris.loc[iris['species'] == 'setosa']
versicolor = iris.loc[iris['species'] == 'versicolor']
virginica = iris.loc[iris['species'] == 'virginica']

plt.scatter(x=setosa['petal_length'], y=setosa['petal_width'], c = 'blue')
plt.scatter(x=versicolor['petal_length'], y=versicolor['petal_width'], c = 'orange'
plt.scatter(x=virginica['petal_length'], y=virginica['petal_width'], c = 'green')

plt.show()
```




```
sns.jointplot(x="petal_length", y="petal_width", data=iris, kind="reg")
plt.show()
```



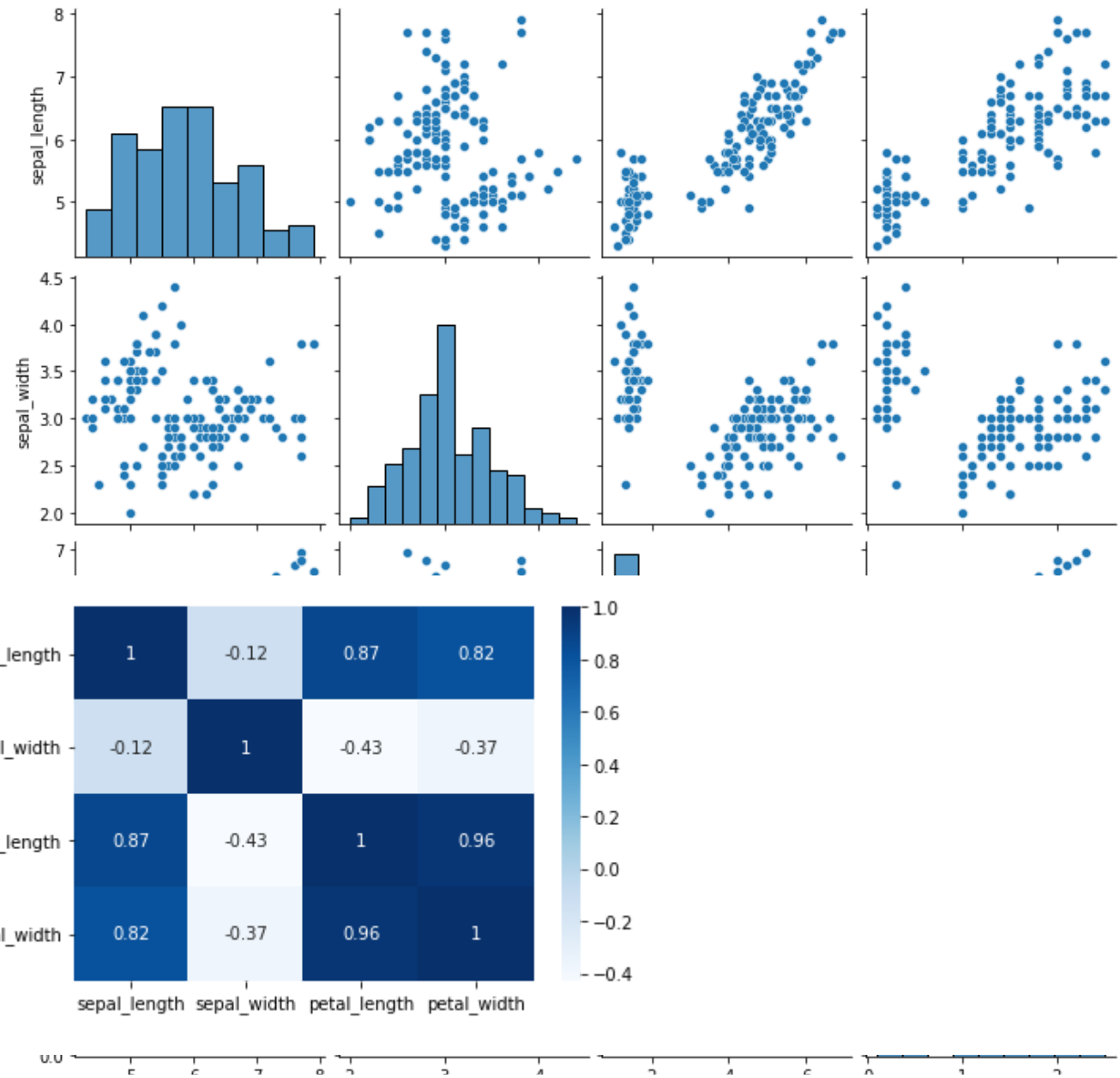
```
sns.jointplot(x="petal_length", y="petal_width", data=iris, hue="species")
plt.show()
```



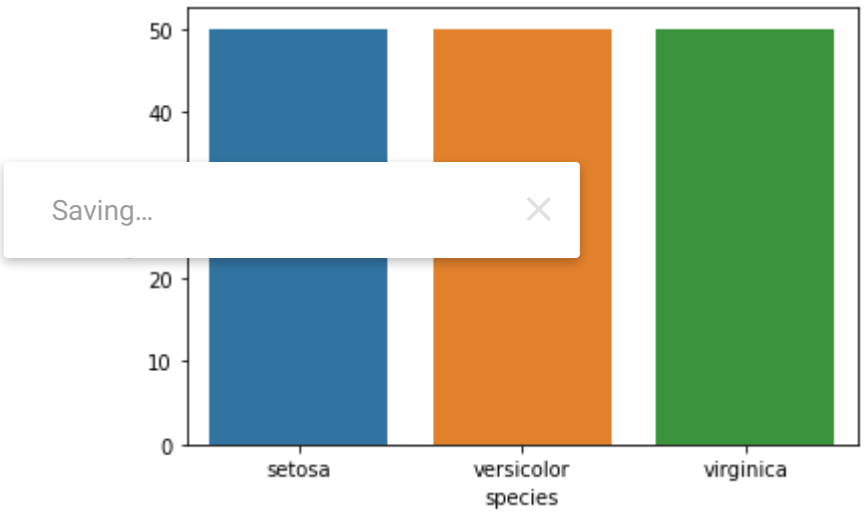

```
sns.pairplot(data=iris)
plt.plot()
```

Saving...

[ ]

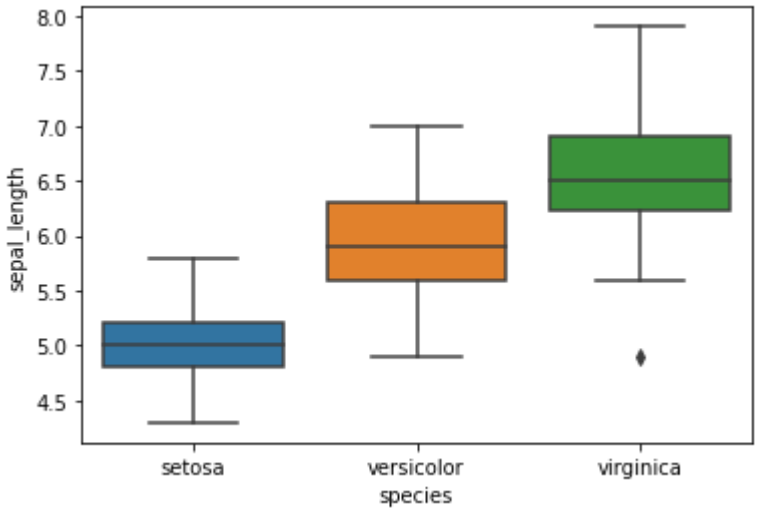



```
# categorical data


sns.countplot(x="species", data=iris)
plt.show()
```

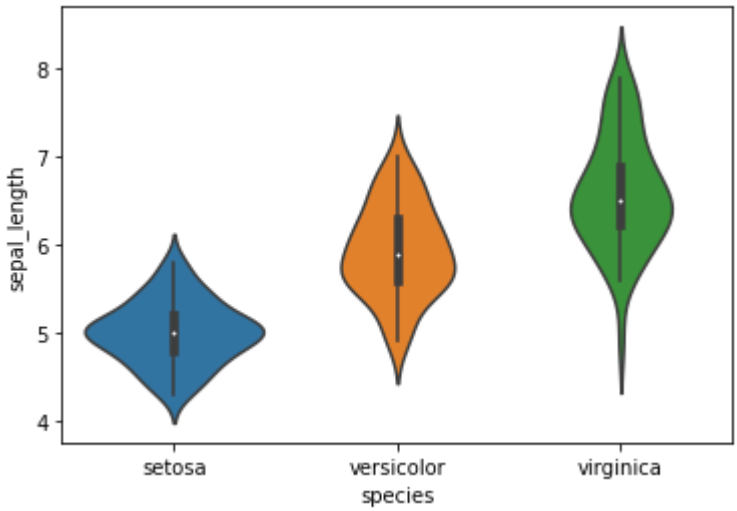


Saving...

```
sns.boxplot(x="species", y="sepal_length", data=iris)
plt.show()
```




```
sns.violinplot(x="species", y="sepal_length", data=iris)
plt.show()
```

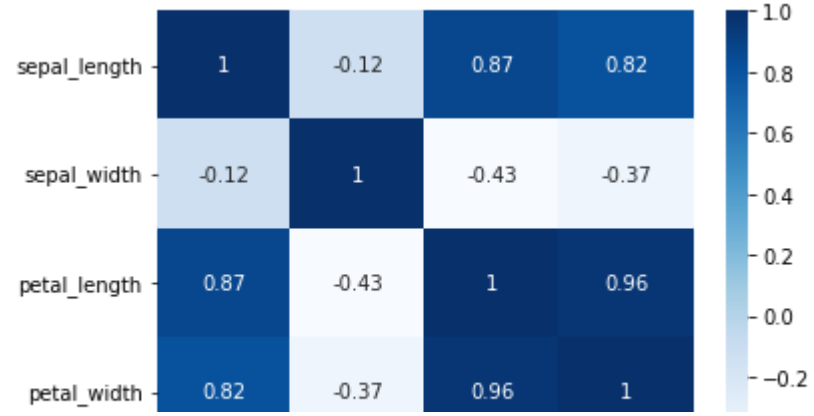



```
iris.corr()
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
			-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000


```
sns.heatmap(iris.corr(), cmap="Blues", annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ae85dd10>
```



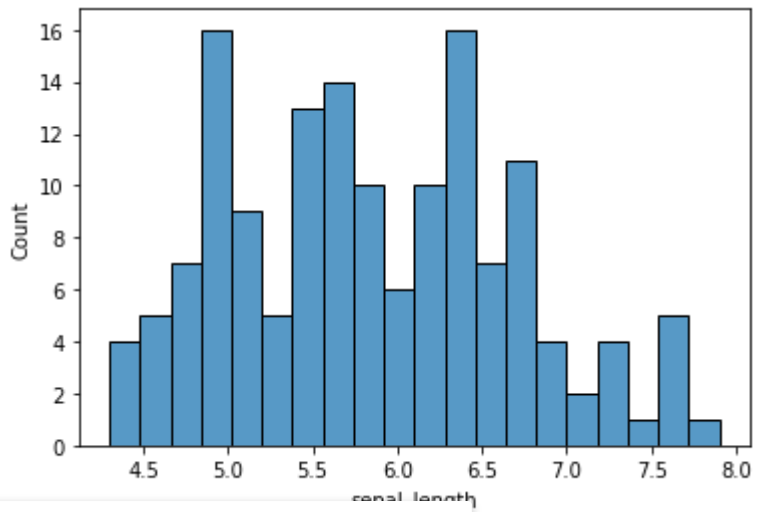

```
plt.colormaps()
```

```
['Accent',
 'Accent_r',
 'Blues',
 'Blues_r',
 'BrBG',
 'BrBG_r',
 'BuGn',
 'BuGn_r',
 'BuPu',
 'BuPu_r',
 'CMRmap',
 'CMRmap_r',
 'Dark2',
 'Dark2_r',
 'GnBu',
 'GnBu_r',
 'Greens',
 'Greens_r',
 'Greys',
 'Greys_r',
 'OrRd',
 'OrRd_r',
 'Oranges',
 'Oranges_r',
 'PRGn',
 'PRGn_r',
 'Paired',
 'Paired_r',
 'Pastel1',
 'Pastel1_r',
 'Pastel2',
 'Pastel2_r',
 'PiYG',
 'PuBuGn',
 'PuBuGn_r',
 'PuBu_r',
 'PuOr',
 'PuOr_r',
 'PuRd',
 'PuRd_r',
 'Purples',
 'Purples_r',
```

```
        'RdBu',
        'RdBu_r',
        'RdGy',
        'RdGy_r',
        'RdPu',
        'RdPu_r',
        'RdYlBu',
        'RdYlBu_r',
        'RdYlGn',
        'RdYlGn_r',
        'Reds',
        'Reds_r',
        'Set1',
        'Set1_r',
```

```
# choosing the right visualisation/plot


# What is the right chart to use to find dependency between
# one Continuous Variable and Categorical variable?


# 1D viz (univariate)- one variable -- num  (hist, kde, boxplot), categorical-count
# 2D viz (bivariate)- two variables -- num-num (scatterplot, jointplot), cat-cat (?
# 2D viz - multiple variables -- pairplot, heatmap
# Multi-dimensional viz (multivariate) -- scatter plot with hue, num-num-categorica


# 1D plots


sns.histplot(iris["sepal_length"], bins=20)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ae8e4650>
```

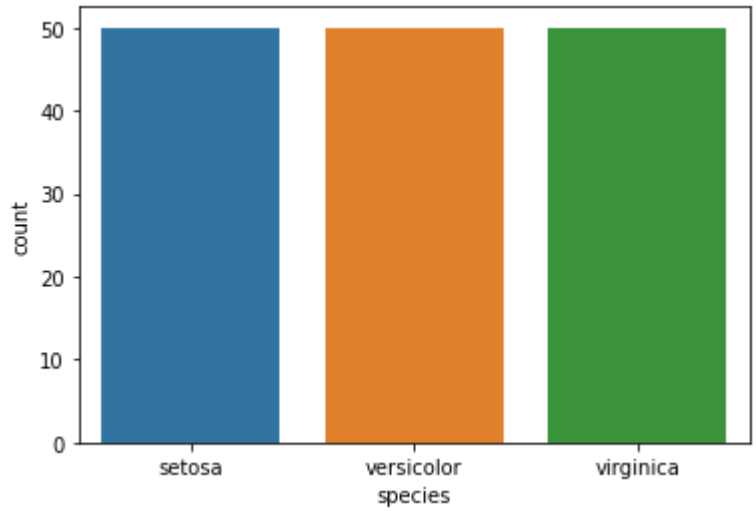



```
sns.boxplot(y="sepal_length", data=iris)
```

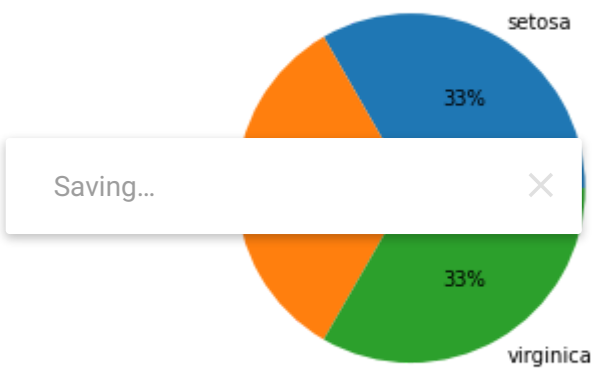
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ae589a90>
```


```
sns.countplot(x="species", data=iris)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ae4d1ed0>
```



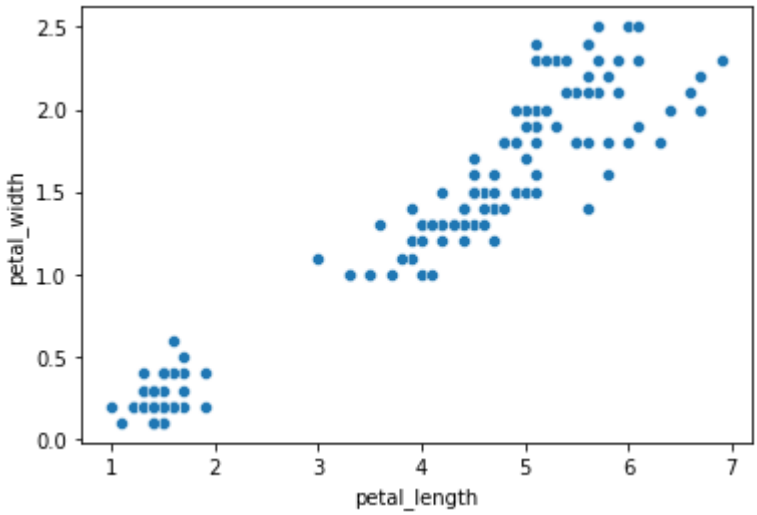

```
n_setosa = iris[iris['species'] == 'setosa'].shape[0]
n_versicolor = iris[iris['species'] == 'versicolor'].shape[0]
n_virginica = iris[iris['species'] == 'virginica'].shape[0]

data = [n_setosa, n_versicolor, n_virginica]
labels=['setosa','versicolor','virginica']

plt.pie(data,
        labels=labels,
        autopct='%.0f%%') # To show the portions in %ages

plt.show()
```




```
# 2D dimensionals
```

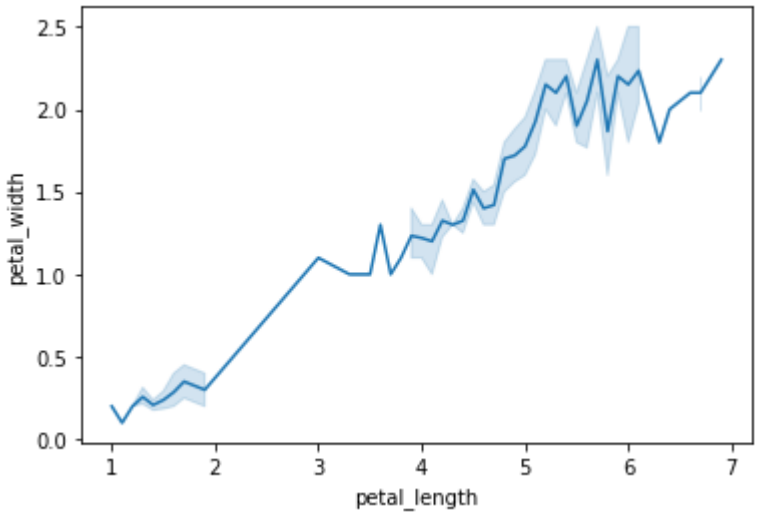
```
sns.scatterplot(x="petal_length", y="petal_width", data=iris)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ae405810>
```




```
sns.lineplot(x="petal_length", y="petal_width", data=iris)
```

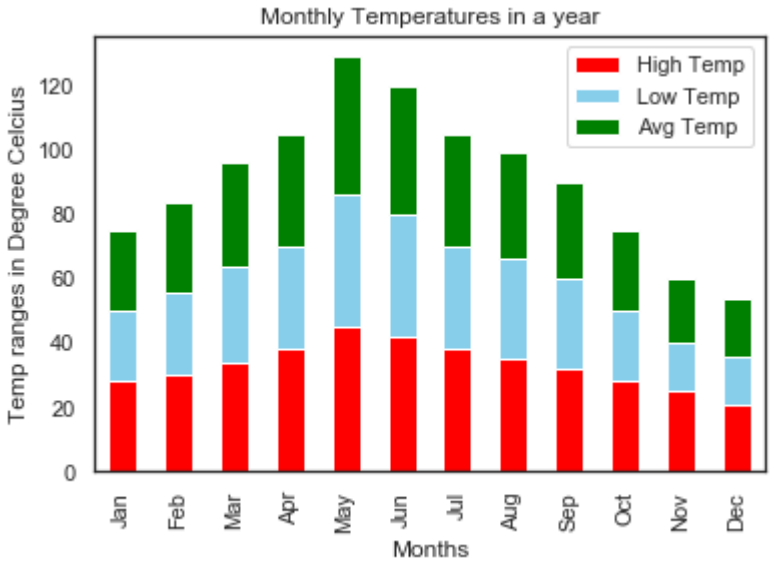
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ae484650>
```




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

<matplotlib.axes._subplots.AxesSubplot at 0x7f14ae491f50>

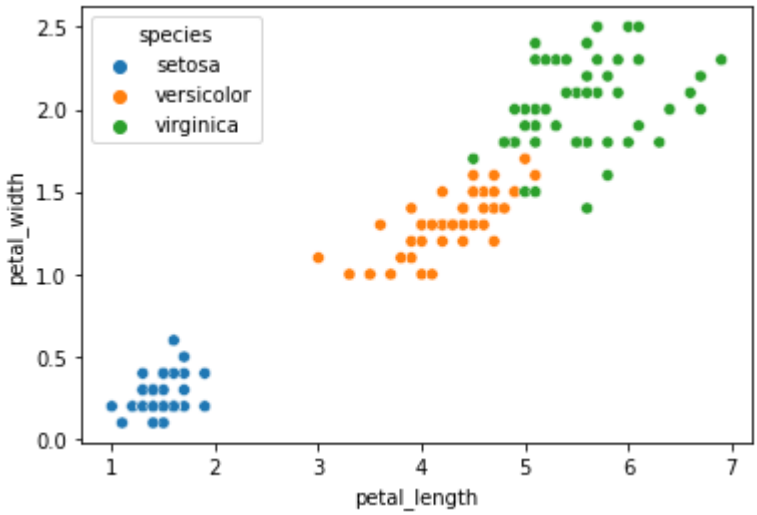
two categorical features




```
# 3D dimensionl plots
```

```
sns.scatterplot(x ='petal_length', y ='petal_width' , data= iris, hue='species')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ae8f5210>
```




```
sns.lineplot(x ='petal_length', y ='petal_width' , data= iris, hue='species')
```

Saving...

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ae956ad0>
```


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

```
tips.shape
```

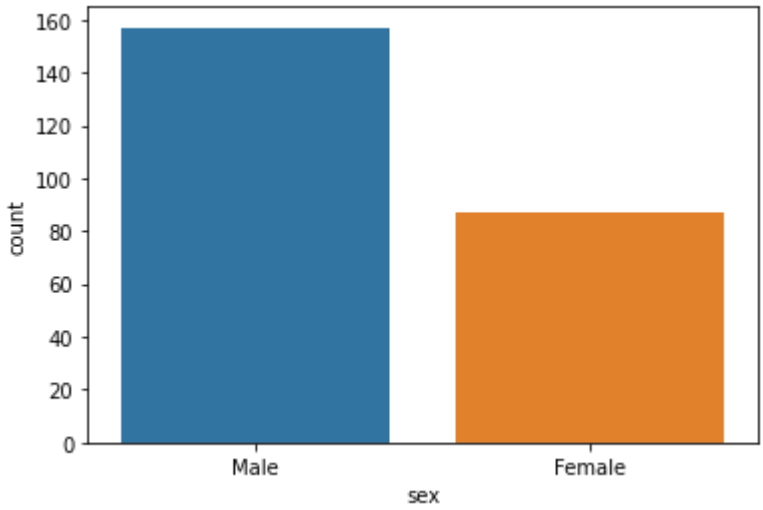
```
(244, 7)
```

```
tips.head()
```

	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

```
sns.countplot(x = tips["sex"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14b60de150>
```



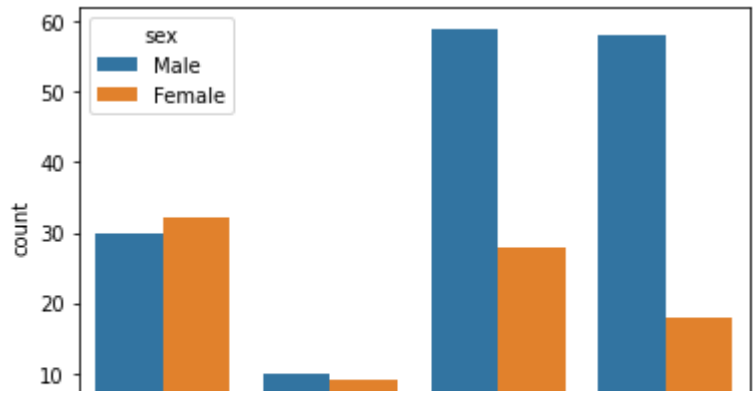

```
tips["day"].unique()
```

Saving...

```
]
ur', 'Fri', 'Sat', 'Sun']
```

```
sns.countplot(x="day", data=tips, hue="sex")
```

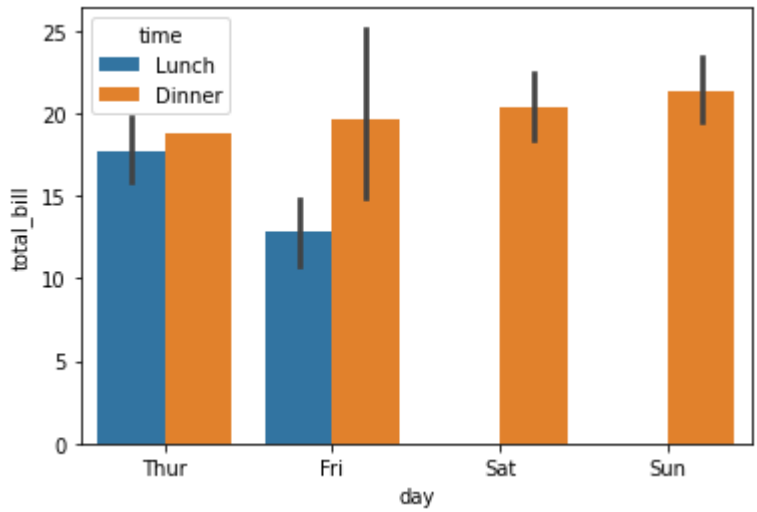
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14b345af90>
```




```
# two categoricals, and one numerical - day, time, total_bill
```

```
sns.barplot(x="day", y="total_bill", hue="time", data=tips, estimator=np.mean)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14b36c04d0>
```




```
# titanic dataset
```

```
# regular expressions (text processing) - Wednesday, Friday
```

✓ 8s completed at 23:53

Saving...