EDA

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

import seaborn as sns

df = pd.read_csv("data.csv")
df.head()
```

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S

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

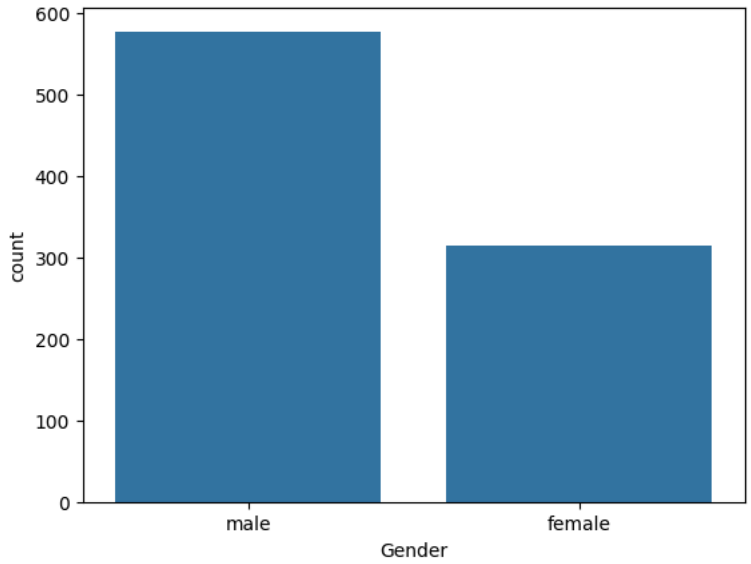
```
df["Gender"].value_counts()
```

Gender	count
male	577
female	314

dtype: int64

```
sns.countplot(x="Gender" , data = df )
```

```
<Axes: xlabel='Gender', ylabel='count'>
```



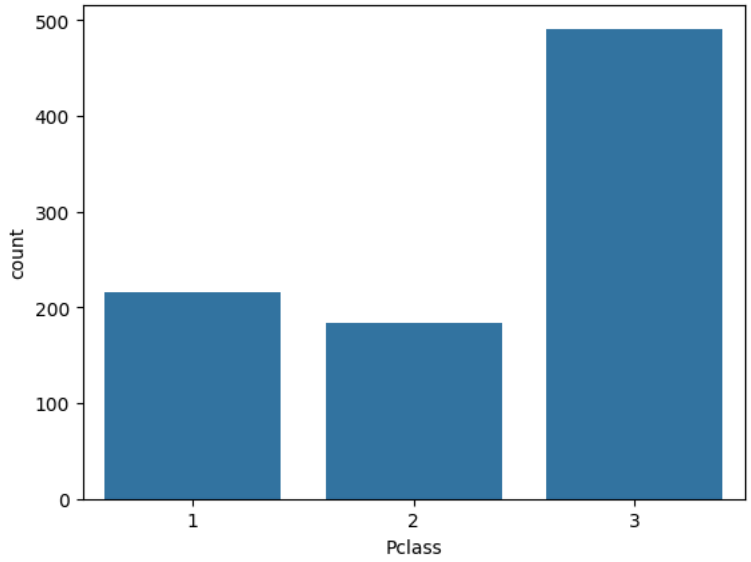

```
df["Pclass"].value_counts()
```

Pclass	count
3	491
1	216
2	184

dtype: int64

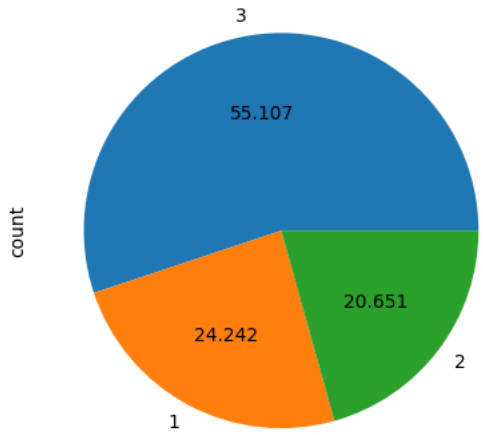
```
sns.countplot(x="Pclass", data=df)
```

```
<Axes: xlabel='Pclass', ylabel='count'>
```




```
df["Pclass"].value_counts().plot(kind="pie" , autopct = "%.3f") #piechart
```

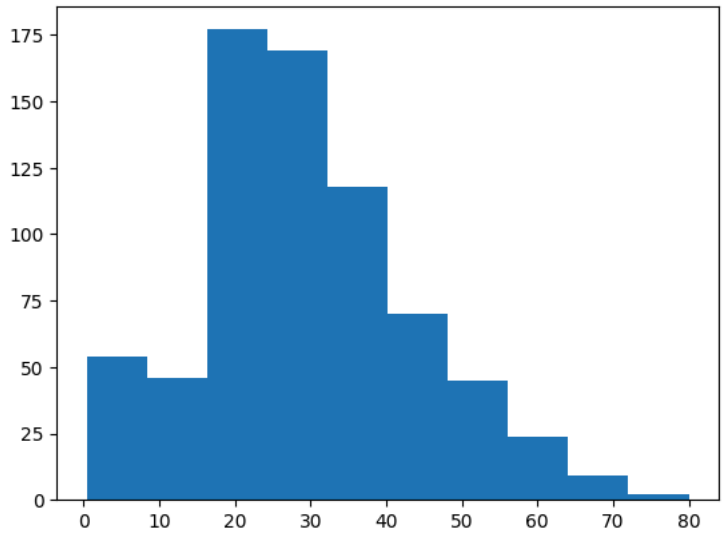
```
<Axes: ylabel='count'>
```




```
df.head()
```

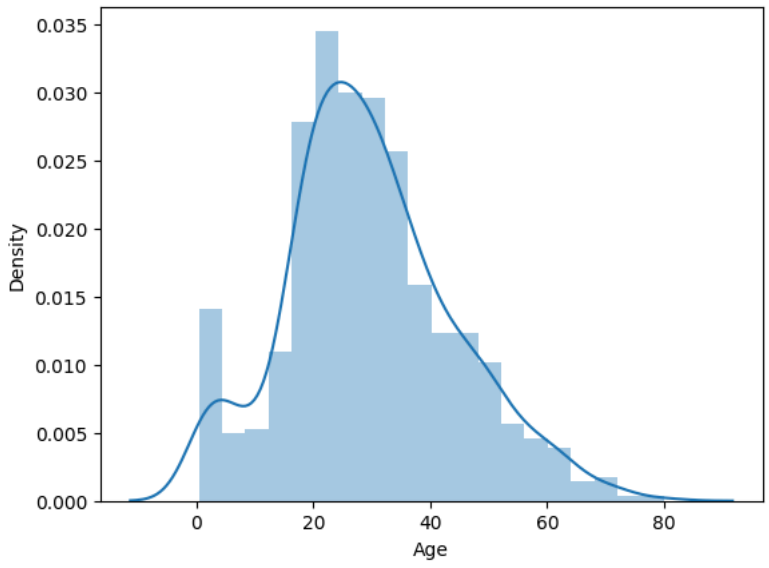
	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S

```
plt.hist(df["Age"] , bins = 10)
plt.show()
```



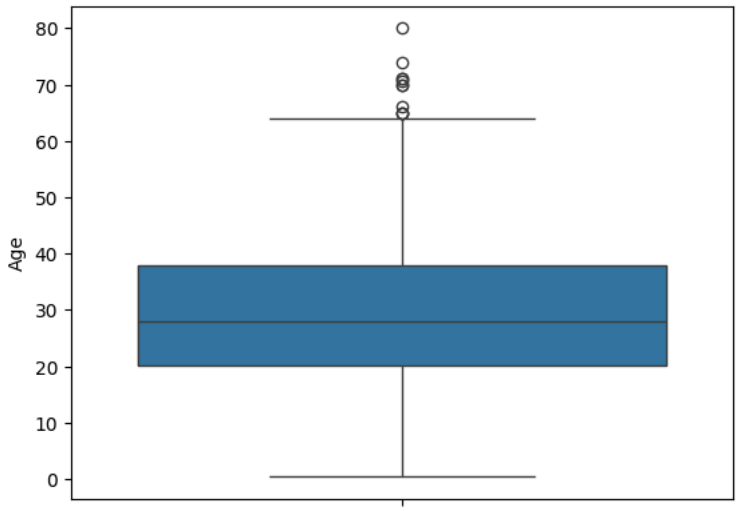

```
sns.distplot(df["Age"])
plt.show()
```

```
<ipython-input-36-6ed014a30d79>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df["Age"])
```



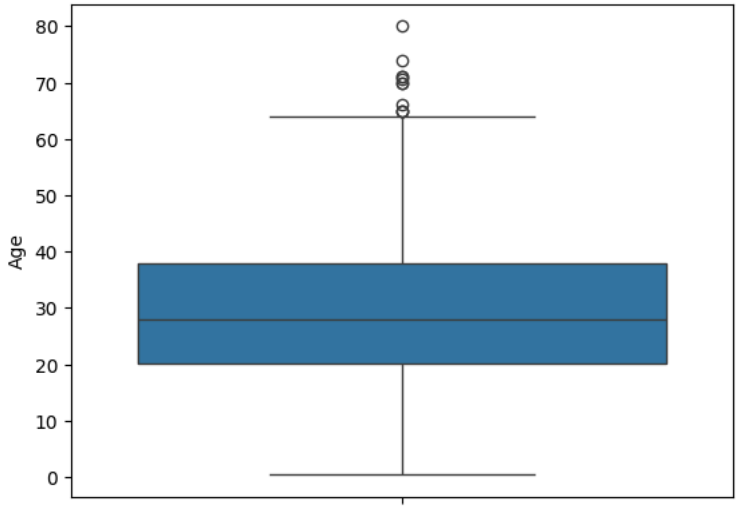

```
sns.boxplot(df["Age"])
```

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



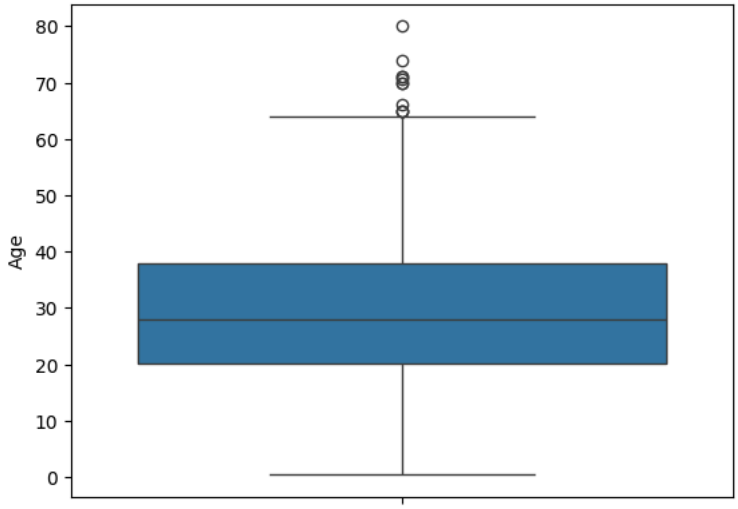

```
sns.boxplot(df["Age"])
```

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



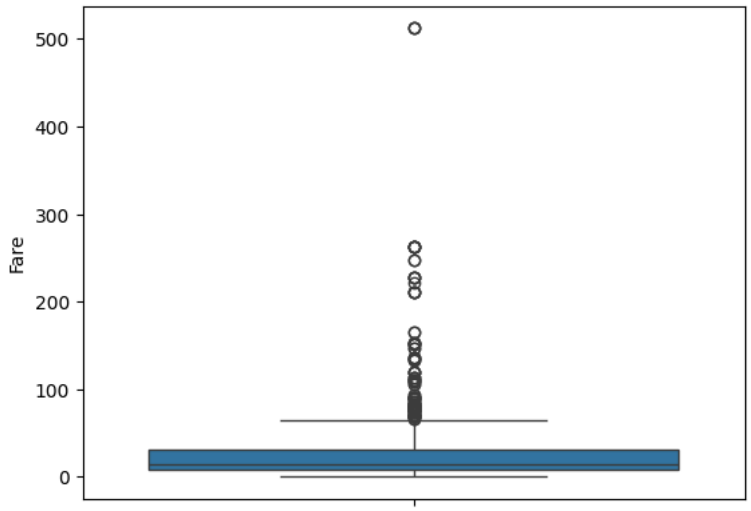

```
sns.boxplot(df["Age"])
```

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




```
sns.boxplot(df["Fare"])
```

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



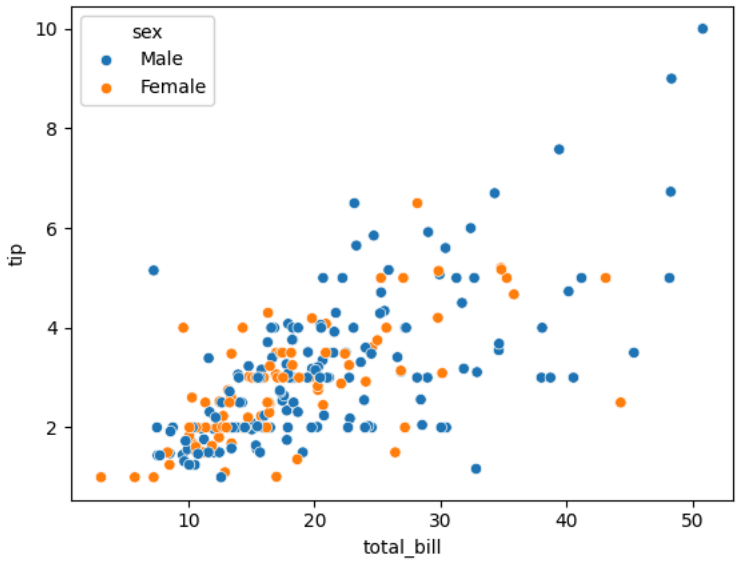

```
titanic = pd.read_csv("data.csv")
flights = sns.load_dataset("flights")
iris = sns.load_dataset("iris")
tips = sns.load_dataset("tips")
```

```
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

```
#scatterplot
```

```
sns.scatterplot(x=tips["total_bill"] , y = tips["tip"] , hue=tips["sex"])
plt.show()
```



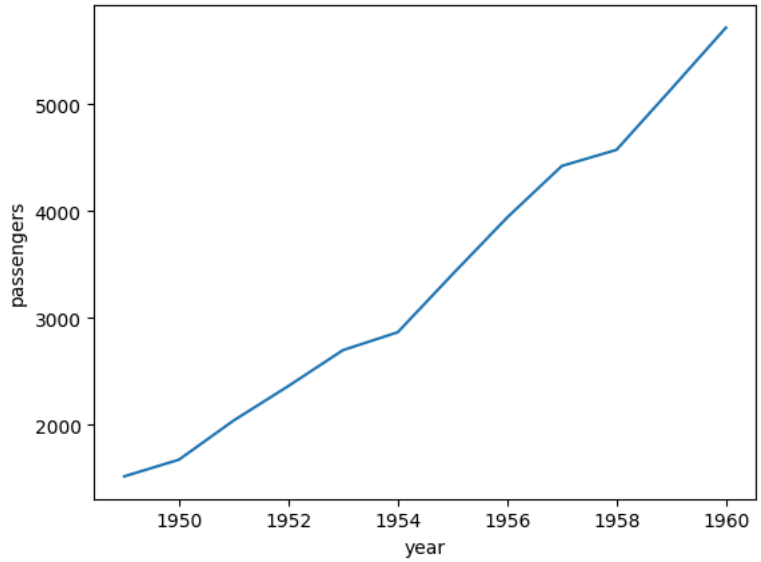

```
flights.head()
```

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

```
flights_group=flights.groupby("year")["passengers"].sum().reset_index()
```

```
sns.lineplot(x=flights_group["year"] , y=flights_group["passengers"])
```

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

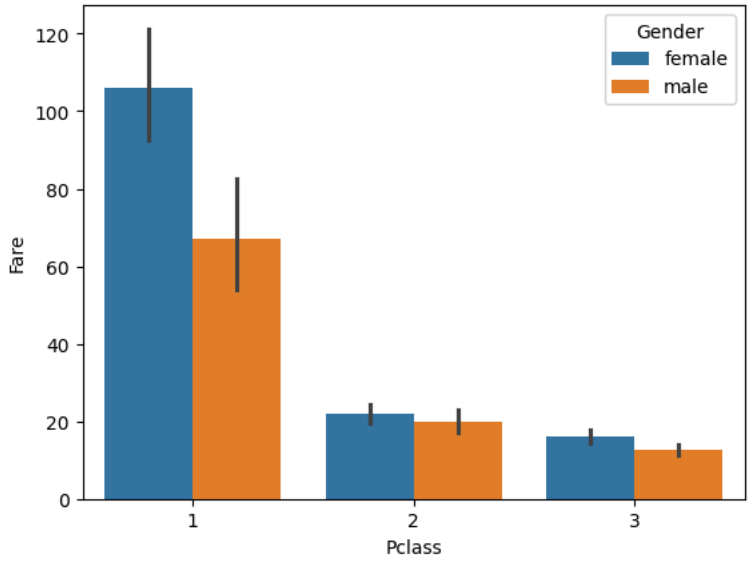



```
titanic.head(3)
```

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C

```
sns.barplot(x=titanic["Pclass"] , y=titanic["Fare"] , hue=titanic["Gender"])
```

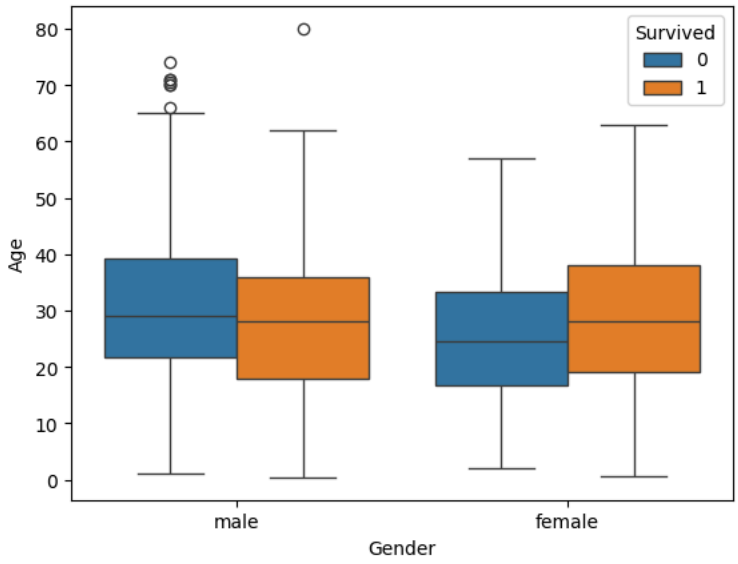
```
<Axes: xlabel='Pclass', ylabel='Fare'>
```



Double-click (or enter) to edit

```
sns.boxplot(x=titanic["Gender"] , y = titanic["Age"] , hue=titanic["Survived"])
```

```
<Axes: xlabel='Gender', ylabel='Age'>
```




```
titanic[titanic["Survived"]==1]["Age"]
```

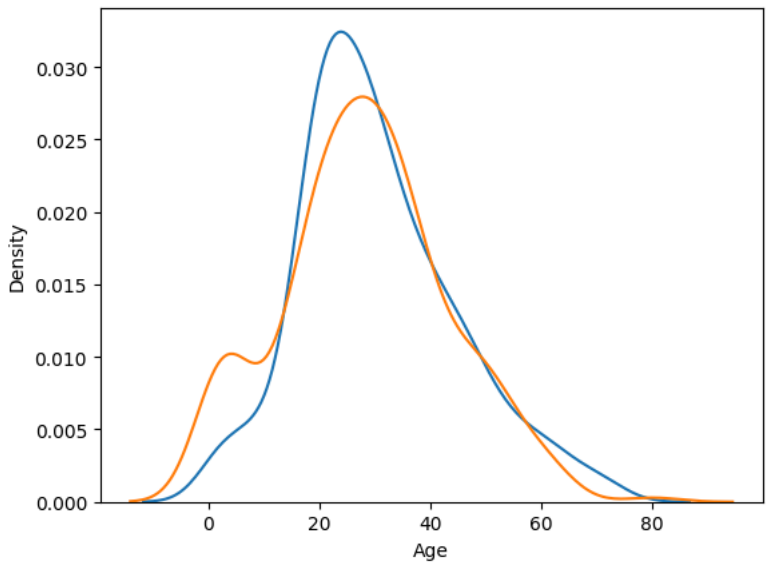
	Age
1	38.0
2	26.0
3	35.0
8	27.0
9	14.0
...	...
875	15.0
879	56.0
880	25.0
887	19.0
889	26.0

342 rows × 1 columns

dtype: float64

```
sns.distplot(titanic[titanic["Survived"]==0]["Age"] , hist = False)
sns.distplot(titanic[titanic["Survived"]==1]["Age"] , hist=False)
```

```
<ipython-input-53-42adc82d2495>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(titanic[titanic["Survived"]==0]["Age"] , hist = False)
<ipython-input-53-42adc82d2495>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(titanic[titanic["Survived"]==1]["Age"] , hist=False)
<Axes: xlabel='Age', ylabel='Density'>
```




```
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["species"].value_counts()
```

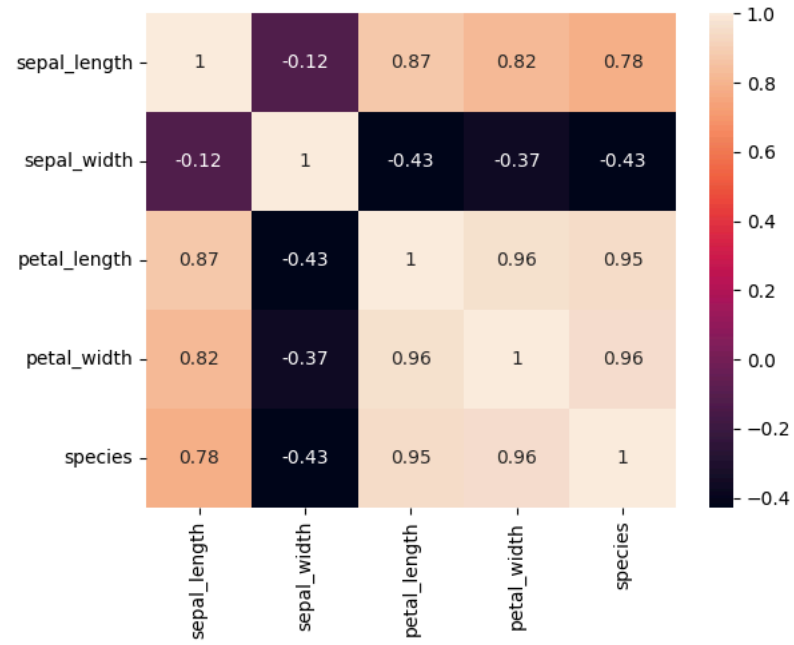
species	count
setosa	50
versicolor	50
virginica	50

dtype: int64

```
from sklearn.preprocessing import LabelEncoder
label_encoder=LabelEncoder()
iris["species"] = label_encoder.fit_transform(iris["species"])
```

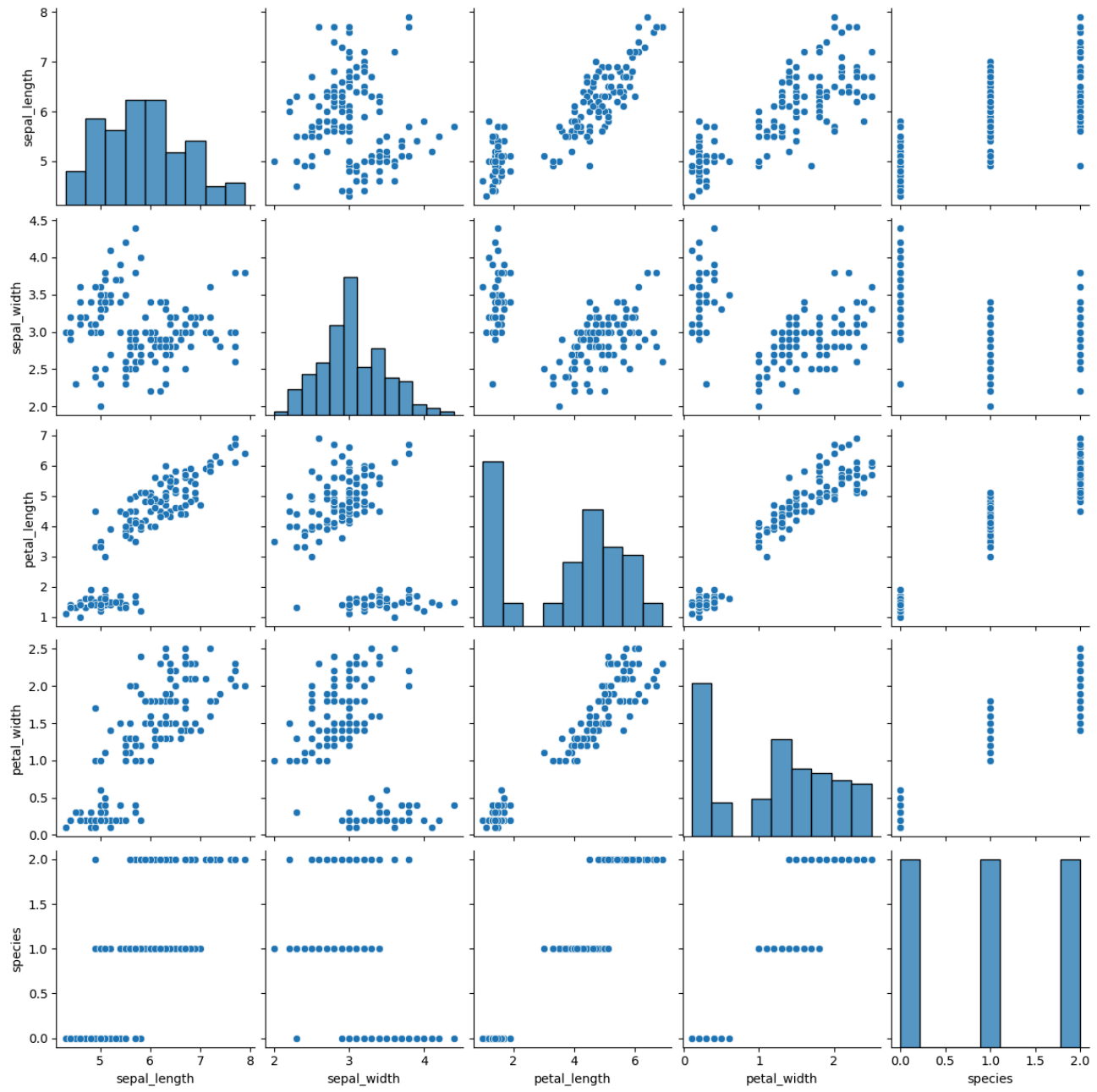
```
corr = iris.corr()
corr
sns.heatmap(corr , annot=True)
```

```
<Axes: >
```




```
sns.pairplot(iris) #multivariate
```

```
<seaborn.axisgrid.PairGrid at 0x7cb227371ad0>
```




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

```
athletes = pd.read_csv("athlete_events.csv")
regions = pd.read_csv("noc_regions.csv")
```

```
athletes.head()
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
		Gunnar							1920					Football	

```
regions.head()
```

	NOC	region	notes
0	AFG	Afghanistan	NaN
1	AHO	Curacao	Netherlands Antilles
2	ALB	Albania	NaN
3	ALG	Algeria	NaN
4	AND	Andorra	NaN

```
df = pd.merge(athletes,regions,on="NOC")
```

```
df.head()
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
		Edgar												Tug-Of-	

```
df.shape
```

```
(270767, 17)
```

```
df.duplicated().sum()
```

```
1385
```

```
df.drop_duplicates(inplace=True)
```

```
df.isnull().sum()
```

	0
ID	0
Name	0
Sex	0
Age	9303
Height	58726
Weight	61437
Team	0
NOC	0
Games	0
Year	0
Season	0
City	0
Sport	0
Event	0
Medal	229619
region	21
notes	264347

dtype: int64

```
df["Medal"].unique() ## check null values in medal because there two many null
```

```
array([nan, 'Gold', 'Bronze', 'Silver'], dtype=object)
```

```
df["Medal"].fillna("No_Medal" , inplace=True) # fill null values qwith no_medal
```

```
<ipython-input-75-28f7feccfab0>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are sett

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df


  df["Medal"].fillna("No_Medal" , inplace=True) # fill null values qwith no_medal
```

```
df.head()
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	No_Meda
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	No_Meda
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	No_Meda
		Edgar												Tug-Of-	

```
df["Season"].value_counts()
```

Season	count
Summer	220818
Winter	48564

dtype: int64

```
summer = df[df["Season"]=="Summer"]
winter = df[df["Season"]=="Winter"]
```

```
summer.shape
```

```
(220818, 17)
```

```
winter.shape
```

```
(48564, 17)
```

```
medal_count_summer = summer.groupby(["NOC","Medal"]).size().reset_index(name="count")
```

```
medal_count_summer
```

	NOC	Medal	count
0	AFG	Bronze	2
1	AFG	No_Medal	124
2	AHO	No_Medal	73
3	AHO	Silver	1
4	ALB	No_Medal	63
...	...	...	...
579	ZAM	Silver	1
580	ZIM	Bronze	1
581	ZIM	Gold	17
582	ZIM	No_Medal	287
583	ZIM	Silver	4

584 rows × 3 columns

```
medal_pivot_summer = medal_count_summer.pivot(index="NOC",columns="Medal",values="count").fillna(0)
```

```
medal_pivot_summer
```

Medal NOC	Bronze	Gold	No_Medal	Silver
AFG	2.0	0.0	124.0	0.0
AHO	0.0	0.0	73.0	1.0
ALB	0.0	0.0	63.0	0.0
ALG	8.0	5.0	522.0	4.0
AND	0.0	0.0	53.0	0.0
...	...	...	...	...
YEM	0.0	0.0	32.0	0.0
YMD	0.0	0.0	5.0	0.0
YUG	92.0	130.0	1659.0	161.0
ZAM	1.0	0.0	181.0	1.0
ZIM	1.0	17.0	287.0	4.0

229 rows × 4 columns

```
medal_pivot_summer = medal_pivot_summer.astype(int) ## decimal to int change krenge
```

```
medal_pivot_summer
```

Medal NOC	Bronze	Gold	No_Medal	Silver
AFG	2	0	124	0
AHO	0	0	73	1
ALB	0	0	63	0
ALG	8	5	522	4
AND	0	0	53	0
...	...	...	...	...
YEM	0	0	32	0
YMD	0	0	5	0
YUG	92	130	1659	161
ZAM	1	0	181	1
ZIM	1	17	287	4

229 rows × 4 columns

```
medal_pivot_summer["Total_Medal"] = medal_pivot_summer[["Gold","Silver","Bronze"]].sum(axis=1)
```

```
medal_pivot_summer
```

Medal NOC	Bronze	Gold	No_Medal	Silver	Total_Medal
AFG	2	0	124	0	2

```
medal_pivot_summer.sort_values(by=["Gold","Silver","Bronze"] , ascending=False).head(40)
```

Medal NOC	Bronze	Gold	No_Medal	Silver	Total_Medal
ALG	8	5	322	4	17
AND	0	0	53	0	0
USA	1197	2472	9813	1333	5002
URS	596	838	2559	635	2063
YEM	0	0	32	0	0
GBR	620	635	8792	729	1984
GER	640	592	5747	538	1779
YUG	92	136	1659	161	389
ITA	454	518	6724	474	1446
FRA	587	463	8809	567	1617
ZIM	1	17	287	4	22
HUN	363	432	4951	328	1123
SWE	358	354	4920	396	1108
AUS	510	342	5787	452	1304
GDR	227	339	1261	277	843
CHN	258	334	3414	317	909