

carsandflowers

April 14, 2024

```
[ ]: import numpy as np
import pandas as pd
import csv
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import pathlib as path
```

```
[ ]: df=pd.read_csv("cars.csv", encoding='unicode escape')
```

```
[ ]: df.shape
```

```
[ ]: (5, 12)
```

```
[ ]: df.head()
```

```
[ ]:
```

	Model	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	\
0	Mazda RX4	21.1	6	160	110	3.91	2.621	16.46	0	1	
1	Mazda RX4 Wag	21.4	6	160	110	3.91	2.875	17.02	0	1	
2	Datsun 710	22.8	4	108	93	3.85	2.321	18.61	1	1	
3	Hornet 7 Drive	21.2	6	258	110	3.08	3.215	19.44	1	0	
4	Hornet Sportabout	18.7	8	360	175	3.15	3.441	17.02	0	0	

	GEAR	CARB
0	4	4
1	4	4
2	4	1
3	3	1
4	3	2

```
[ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 12 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Model    5 non-null         object
```

```

1  MPG      5 non-null    float64
2  CYL      5 non-null    int64
3  DISP     5 non-null    int64
4  HP       5 non-null    int64
5  DRAT     5 non-null    float64
6  WT       5 non-null    float64
7  QSEC     5 non-null    float64
8  VS       5 non-null    int64
9  AM       5 non-null    int64
10 GEAR     5 non-null    int64
11 CARB     5 non-null    int64
dtypes: float64(4), int64(7), object(1)
memory usage: 608.0+ bytes

```

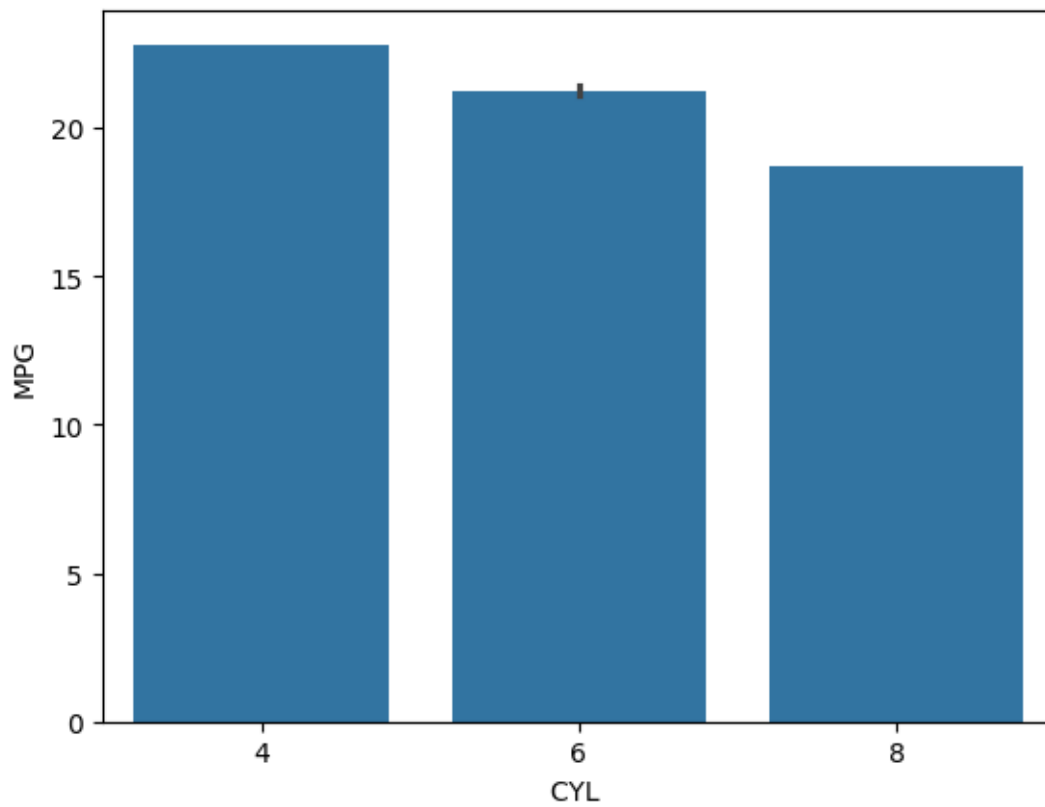
Abbreviation of columns along with their meaning
 MODEL=MODEL,
 MPG=MILES PER GALLON,
 CYL=CYLINDER,
 DISP-DISPLACEMENT IN QUBIC INCH,
 DRAT=REAL AXIS RATIO,
 WT=WEIGHT*1000 POUND,
 QSEC=1/4 MILE TIME(TIME TAKEN TO COMPLETE QUARTER MILE),
 VS=TYPE OF ENGINE, 0-'V'Engine 1-Straight Engine ,
 AM=mode of transmission, 0-Automatic 1-Manual,
 GEAR=Forward gear,
 CARB= No of carborator

```
[ ]: df.columns
```

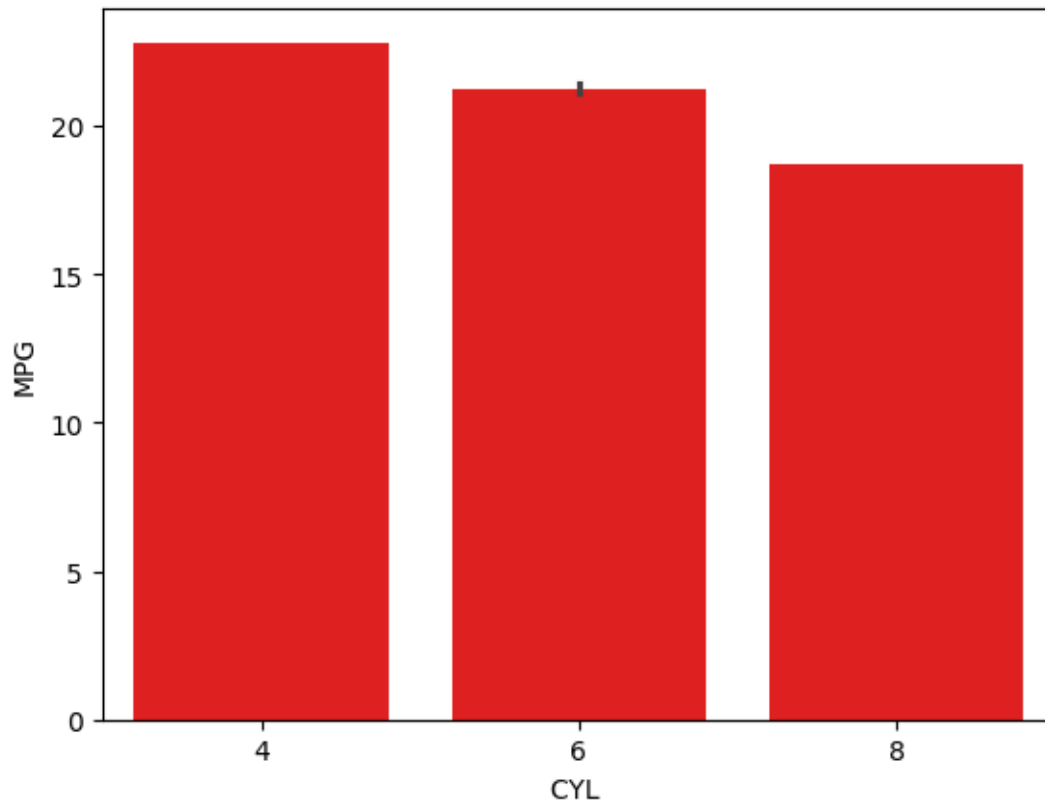
```
[ ]: Index(['Model ', 'MPG', 'CYL ', 'DISP', 'HP ', 'DRAT', 'WT', 'QSEC', 'VS',
          'AM', 'GEAR', 'CARB '],
          dtype='object')
```

#BARPLOT

```
[ ]: res=sns.barplot(x='CYL ',y='MPG',data=df)
plt.show()
```



```
[ ]: res=sns.barplot(x='CYL ',y='MPG',data=df,color='red')  
plt.show()
```

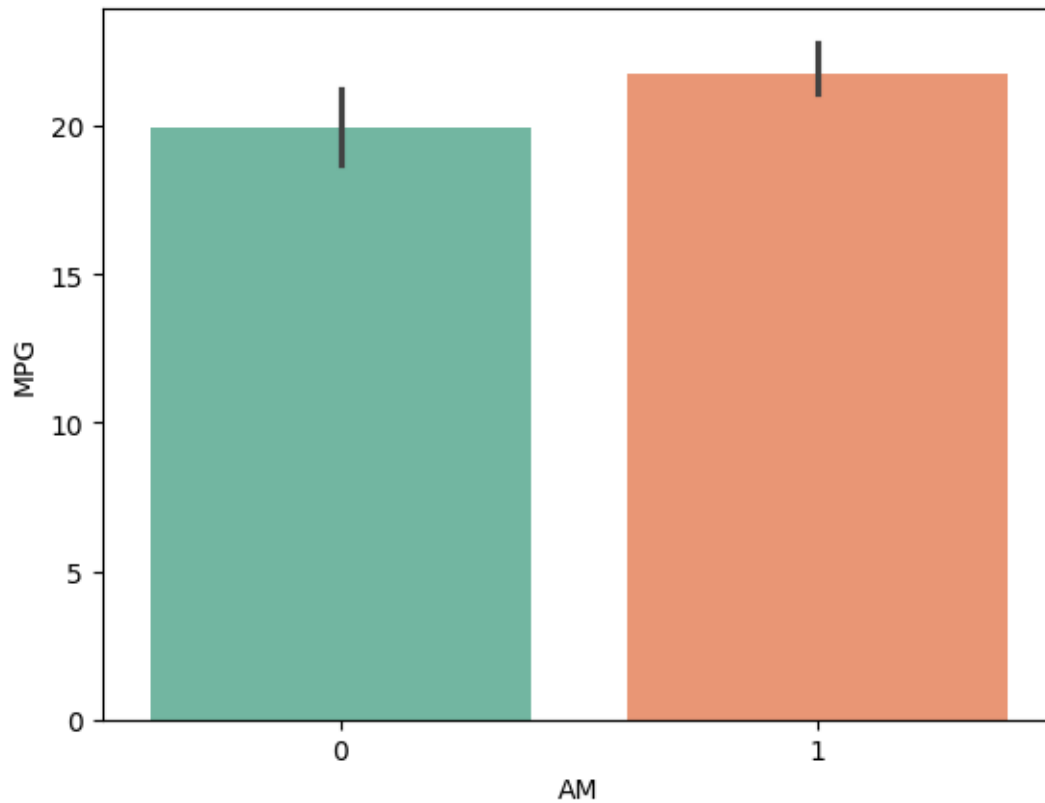


```
[ ]: res=sns.barplot(x='AM',y='MPG',data=df,palette='Set2')
plt.show()
```

<ipython-input-12-747238b8f088>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
res=sns.barplot(x='AM',y='MPG',data=df,palette='Set2')
```



```
[ ]: df.columns
```

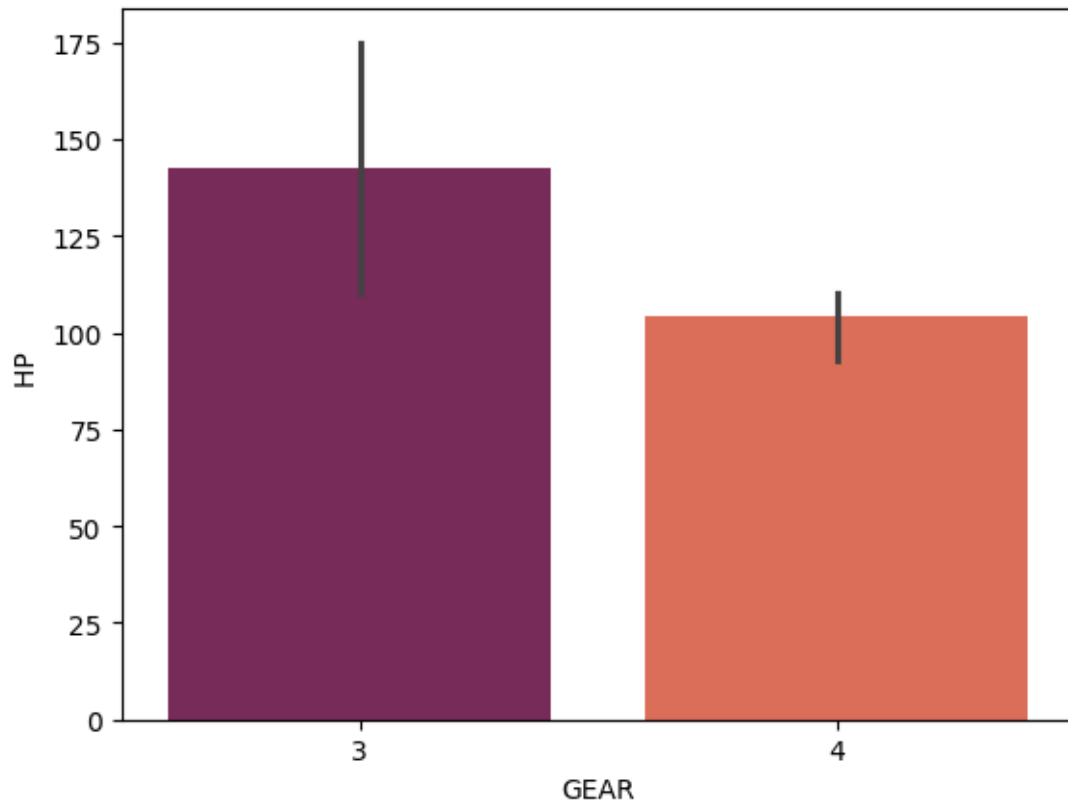
```
[ ]: Index(['Model ', 'MPG', 'CYL ', 'DISP', 'HP ', 'DRAT', 'WT', 'QSEC', 'VS',  
          'AM', 'GEAR', 'CARB '],  
         dtype='object')
```

```
[ ]: res=sns.barplot(x='GEAR',y='HP ',data=df,palette='rocket')  
plt.show()
```

<ipython-input-14-6dfcee706a7b>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
res=sns.barplot(x='GEAR',y='HP ',data=df,palette='rocket')
```



##CountPlot

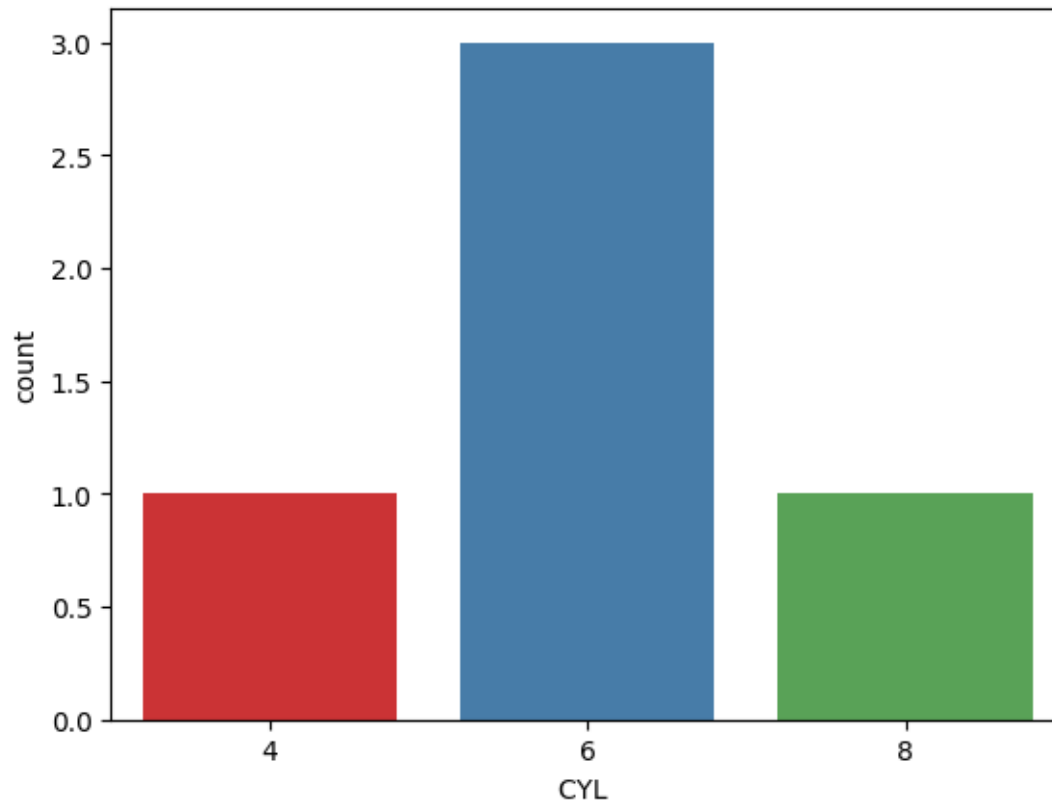
```
[ ]: sns.countplot(x='CYL ',data=df,palette='Set1')
```

<ipython-input-15-46007f7f7728>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(x='CYL ',data=df,palette='Set1')
```

```
[ ]: <Axes: xlabel='CYL ', ylabel='count'>
```



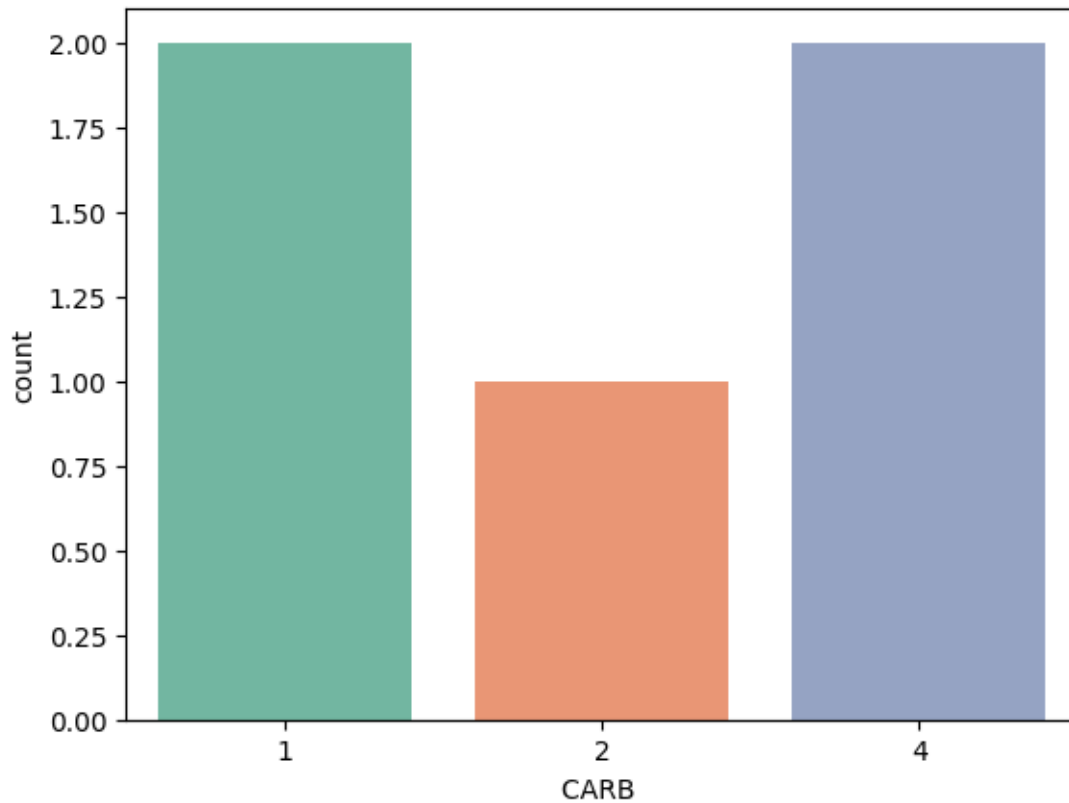
```
[ ]: sns.countplot(x='CARB ',data=df,palette='Set2')
```

<ipython-input-19-ae0d4e075072>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(x='CARB ',data=df,palette='Set2')
```

```
[ ]: <Axes: xlabel='CARB ', ylabel='count'>
```



#Horizontal countplot

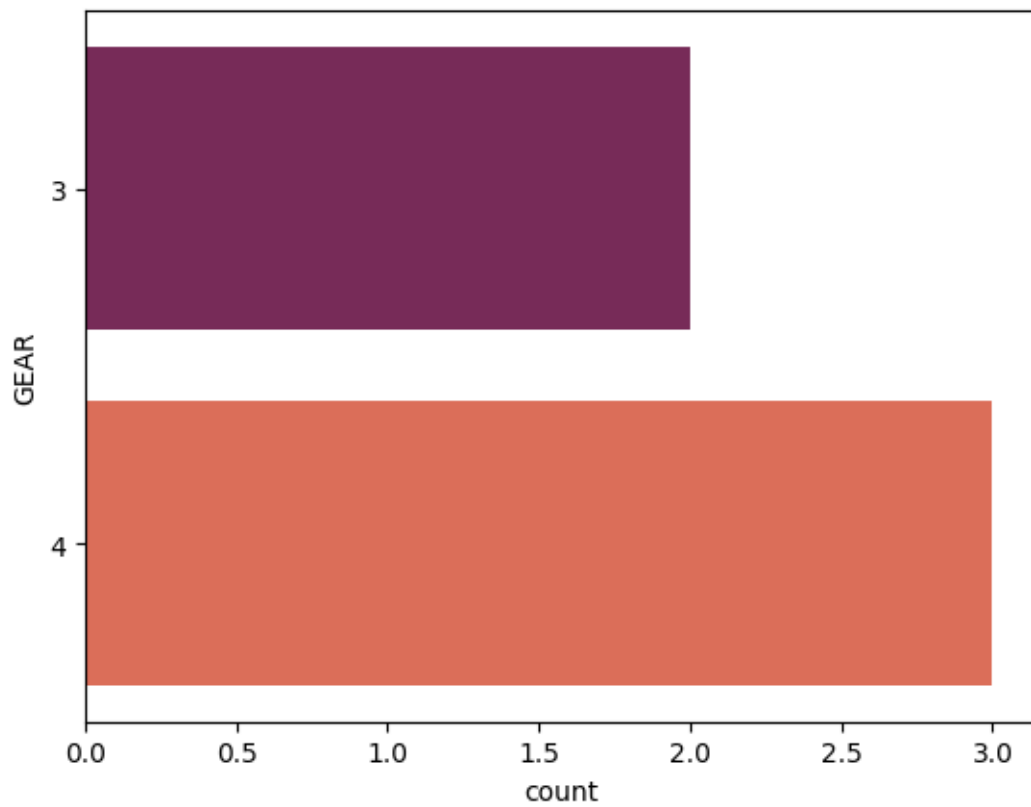
```
[ ]: sns.countplot(y='GEAR',data=df,palette='rocket')
```

<ipython-input-21-5123069fdb7>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(y='GEAR',data=df,palette='rocket')
```

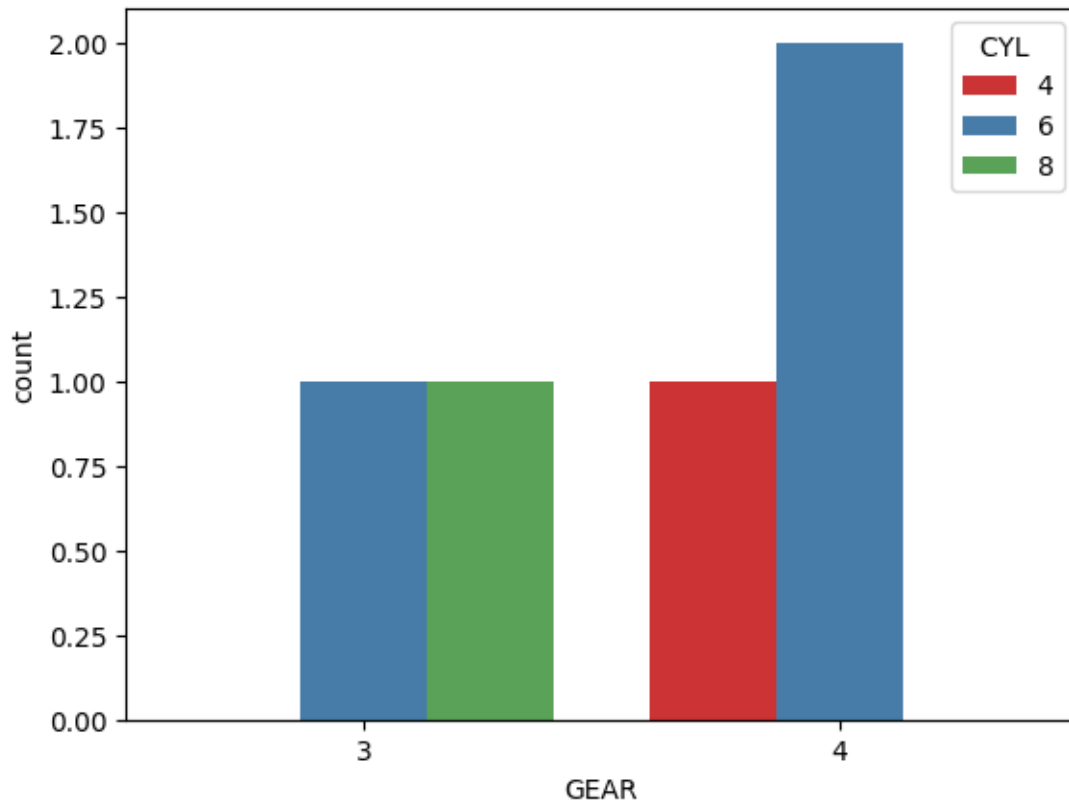
```
[ ]: <Axes: xlabel='count', ylabel='GEAR'>
```

#Grouped Countplot

```
[ ]: sns.countplot(x='GEAR',hue='CYL ',data=df,palette='Set1')
```

```
[ ]: <Axes: xlabel='GEAR', ylabel='count'>
```



#Distplot

```
[ ]: sns.distplot(df.MPG,bins=5,color='green')
```

<ipython-input-23-40d484c8eb31>:2: UserWarning:

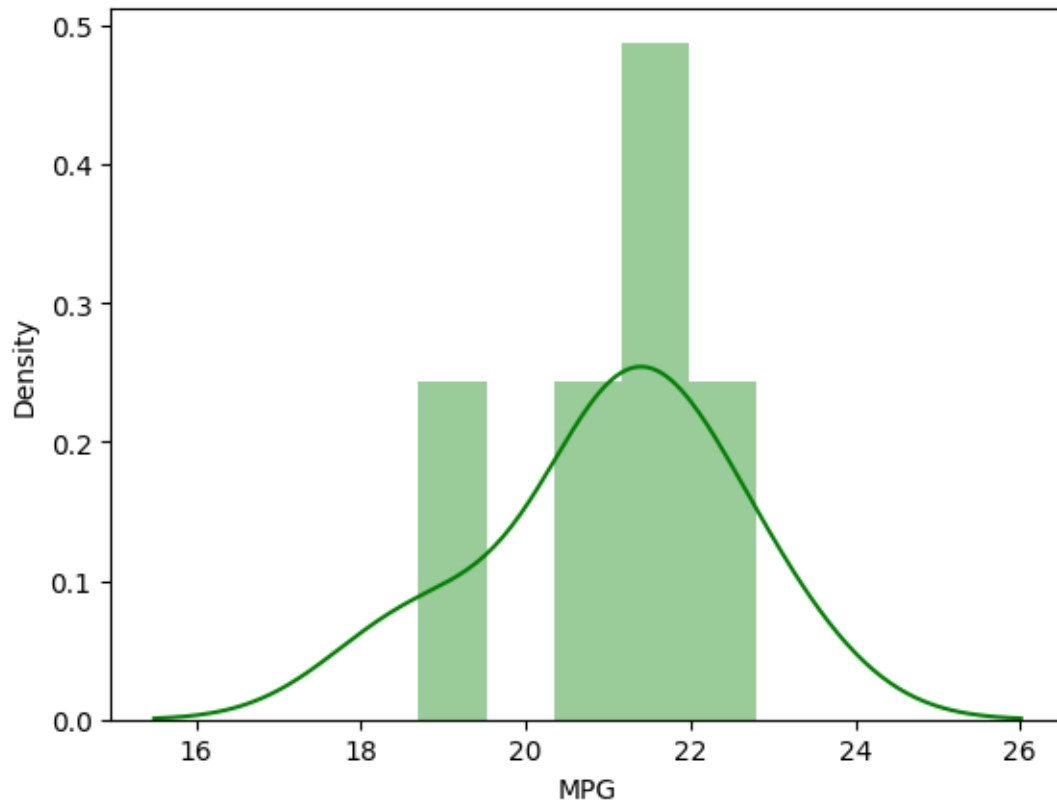
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `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.MPG,bins=5,color='green')
```

```
[ ]: <Axes: xlabel='MPG', ylabel='Density'>
```



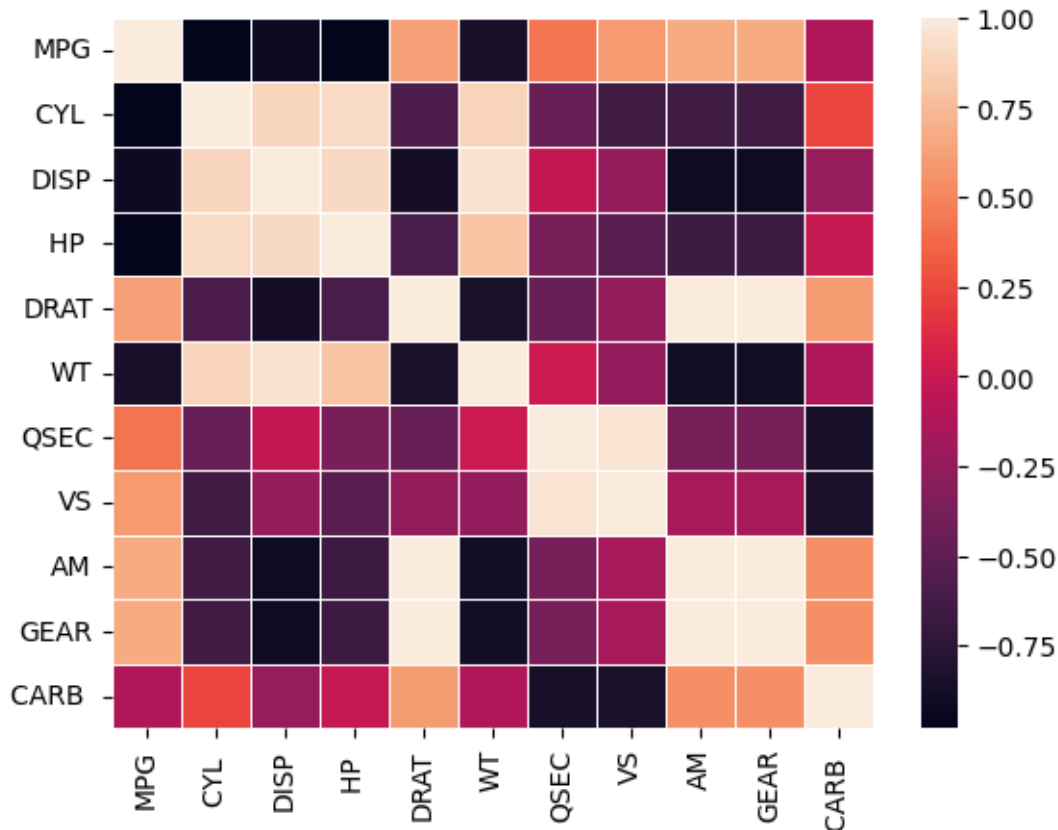
#HEATMAP FOR CORELATION

```
[ ]: #heatmap
sns.heatmap(df.corr(),cbar=True,linewidth=0.5)
```

<ipython-input-24-c996b3f7d381>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
sns.heatmap(df.corr(),cbar=True,linewidth=0.5)
```

```
[ ]: <Axes: >
```



```
[ ]: iris=sns.load_dataset('iris')
iris.head()
```

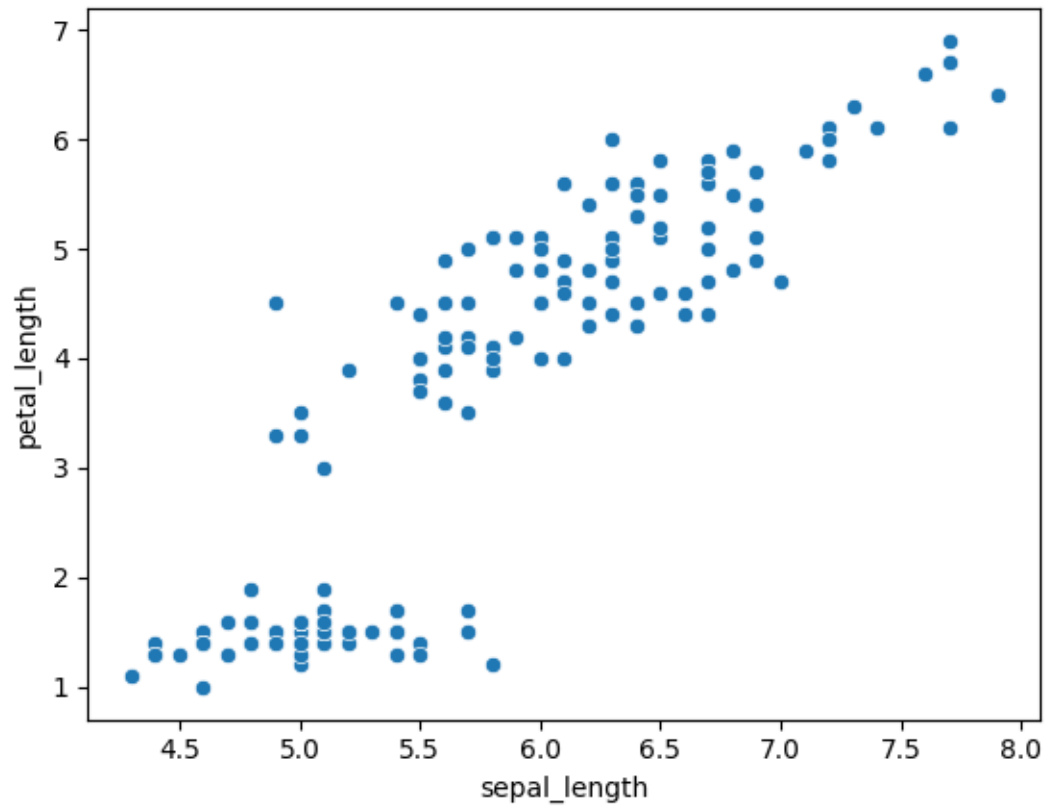
```
[ ]:   sepal_length  sepal_width  petal_length  petal_width  species
0         5.1         3.5         1.4         0.2  setosa
1         4.9         3.0         1.4         0.2  setosa
2         4.7         3.2         1.3         0.2  setosa
3         4.6         3.1         1.5         0.2  setosa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

```
[ ]: iris['species'].value_counts()
```

```
[ ]: setosa      50
versicolor    50
virginica     50
Name: species, dtype: int64
```

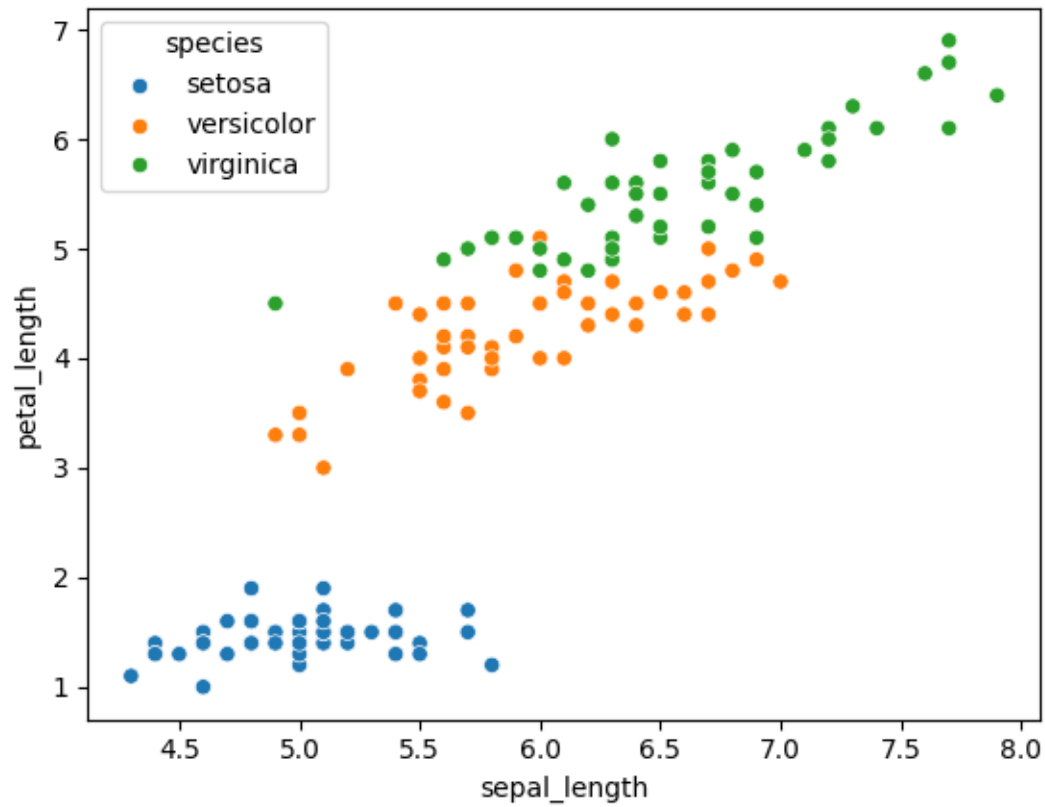
```
[ ]: sns.scatterplot(x='sepal_length',y='petal_length',data=iris )
```

```
[ ]: <Axes: xlabel='sepal_length', ylabel='petal_length'>
```



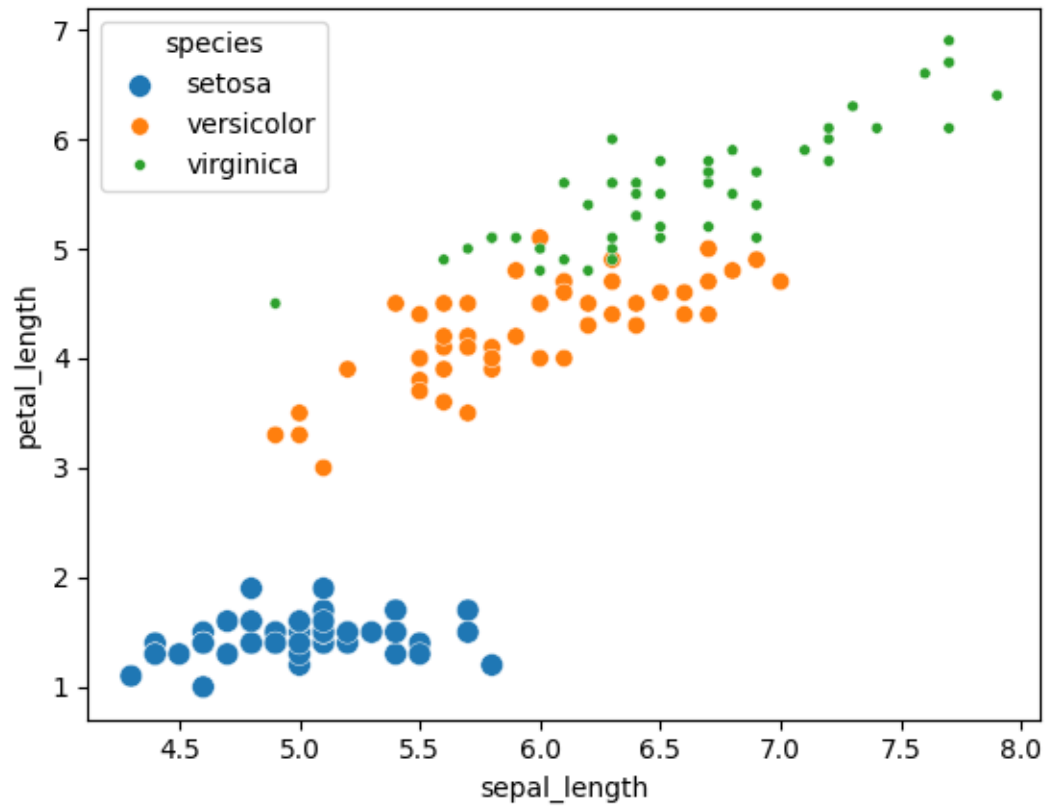
```
[ ]: sns.scatterplot(x='sepal_length',y='petal_length',data=iris,hue='species')
```

```
[ ]: <Axes: xlabel='sepal_length', ylabel='petal_length'>
```



#SCATTERPLOT

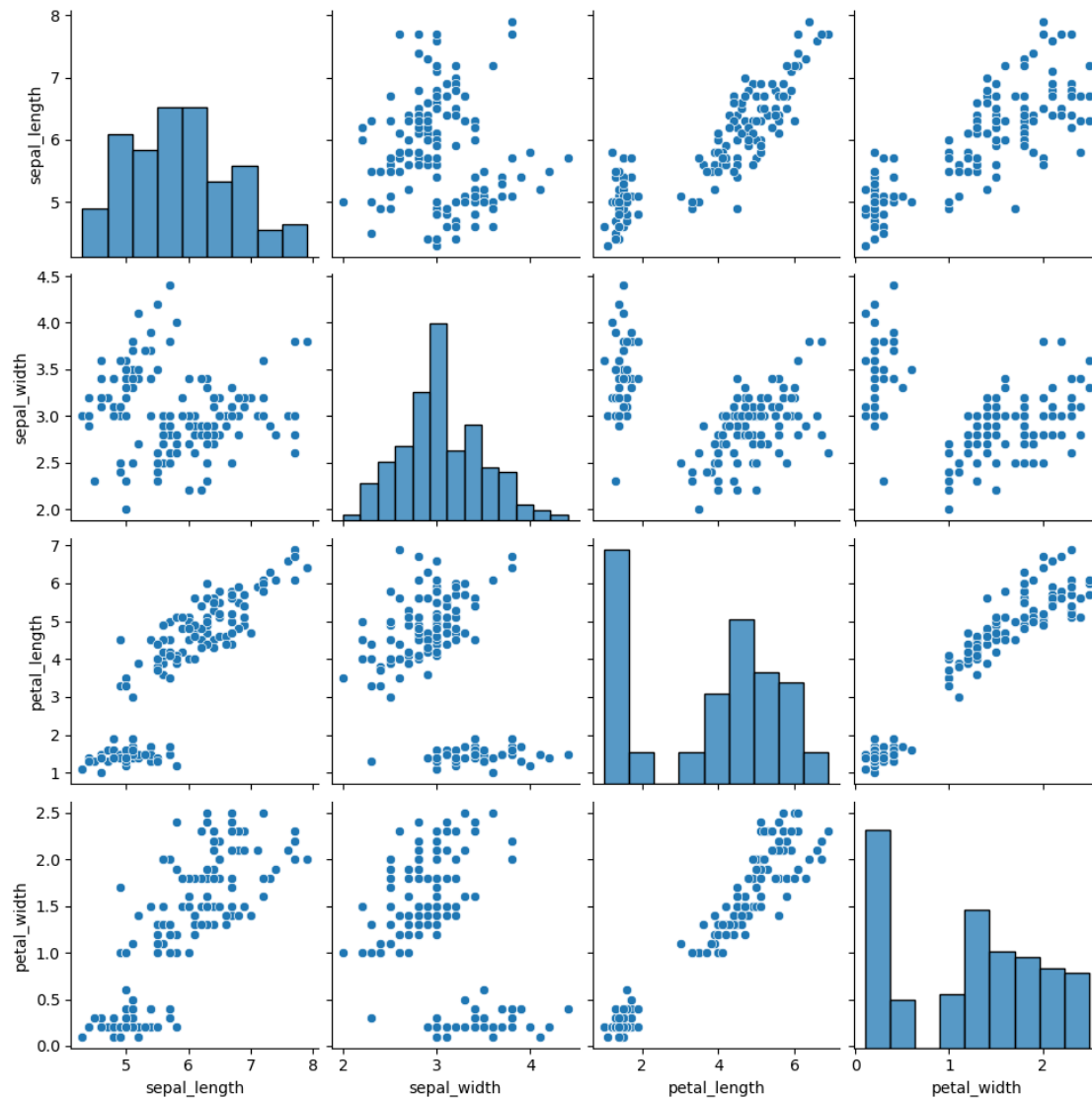
```
[ ]: sns.  
      ↳scatterplot(x='sepal_length',y='petal_length',data=iris,hue='species',size='species')  
[ ]: <Axes: xlabel='sepal_length', ylabel='petal_length'>
```



```
#PIARPLOT
```

```
[ ]: sns.pairplot(iris)
```

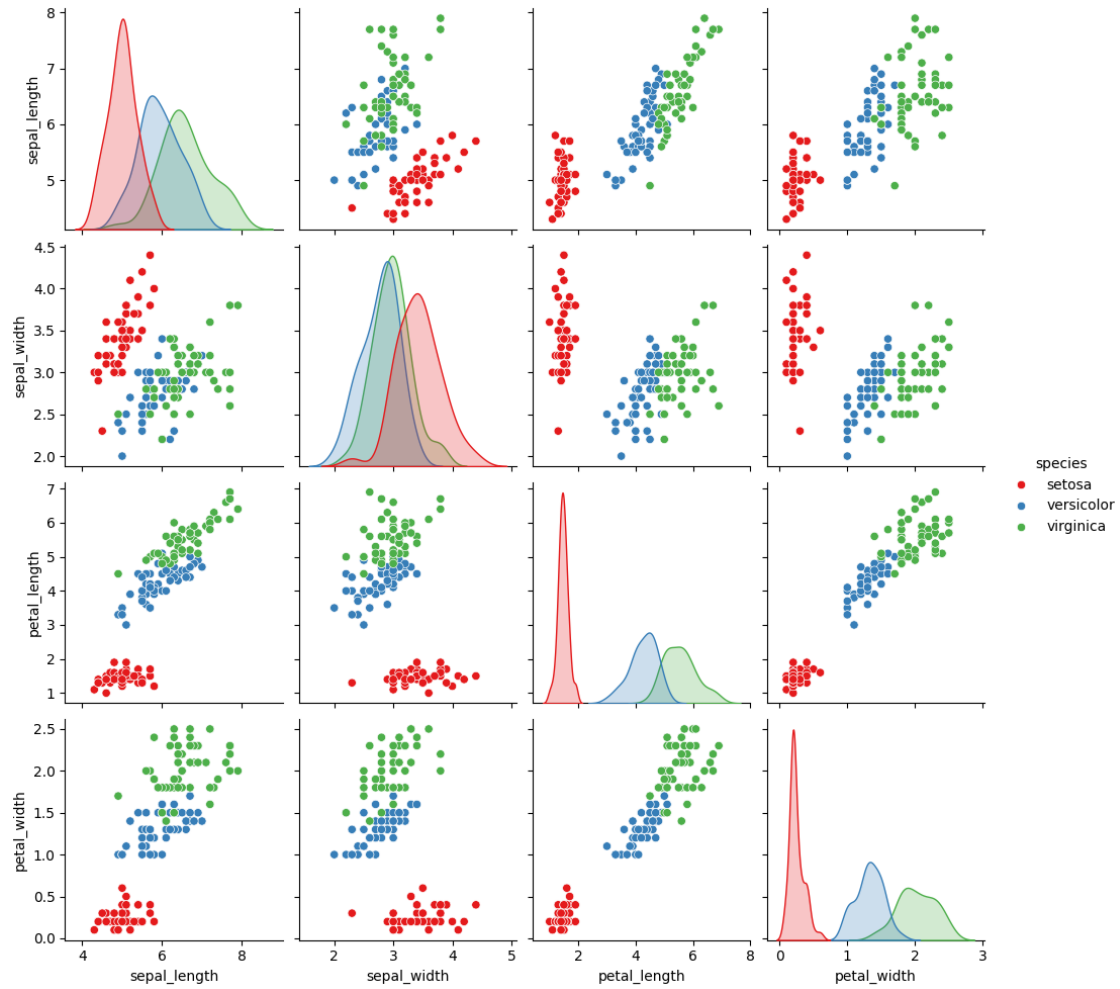
```
[ ]: <seaborn.axisgrid.PairGrid at 0x7dbf1a0e9a50>
```



#PAIRPLOT

```
[ ]: sns.pairplot(iris,hue='species',palette='Set1')
```

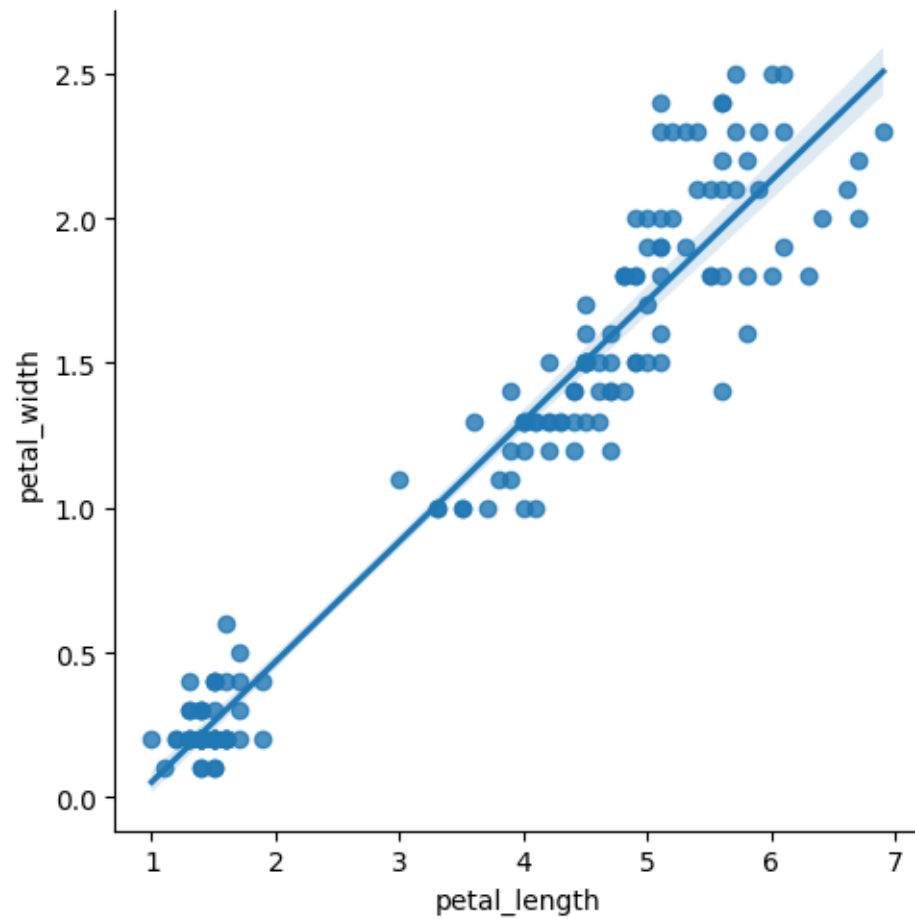
```
[ ]: <seaborn.axisgrid.PairGrid at 0x7dbf1458b8b0>
```

#LMPLLOT

```
[ ]: sns.lmplot(x='petal_length',y='petal_width',data=iris)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7dbf0cfabd60>
```



#BOXPLOT

```
[ ]: sns.boxplot(x='species',y='sepal_width',data=iris)
```

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

