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

```
data =pd.read_csv("/content/Iris.csv")
```

```
data.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Next steps: Generate code with data View recommended plots

```
data.tail()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

```
data.shape
```

```
(150, 6)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

```
data.describe()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

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

```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

```
data['SepalLengthCm'].max()
```

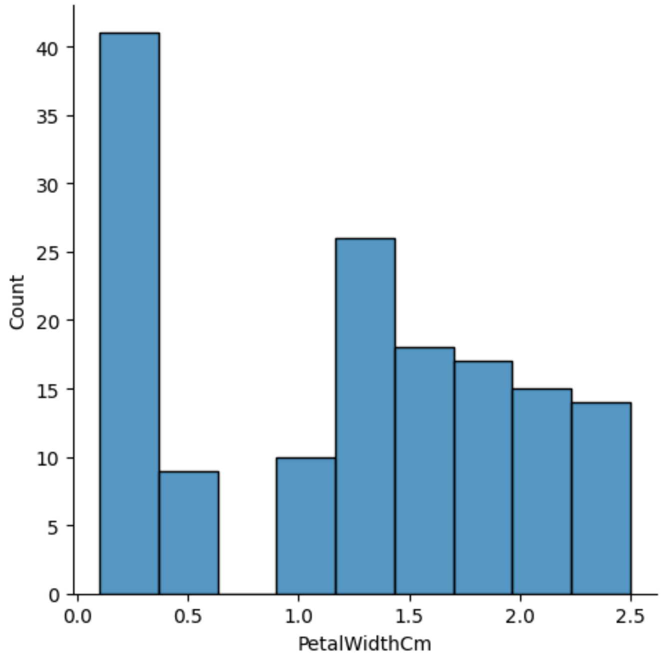
```
7.9
```

```
data.columns
```

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

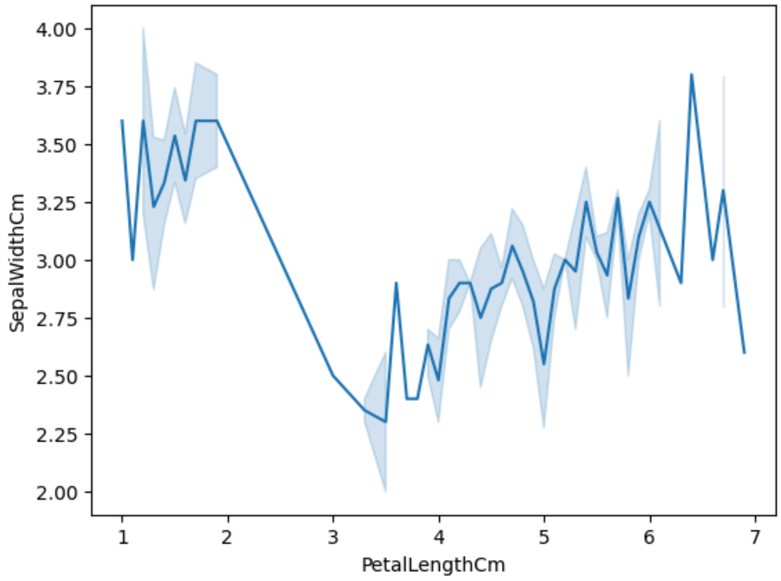
```
sns.displot(data['PetalWidthCm'])
```

```
<seaborn.axisgrid.FacetGrid at 0x7f1b86cb6200>
```



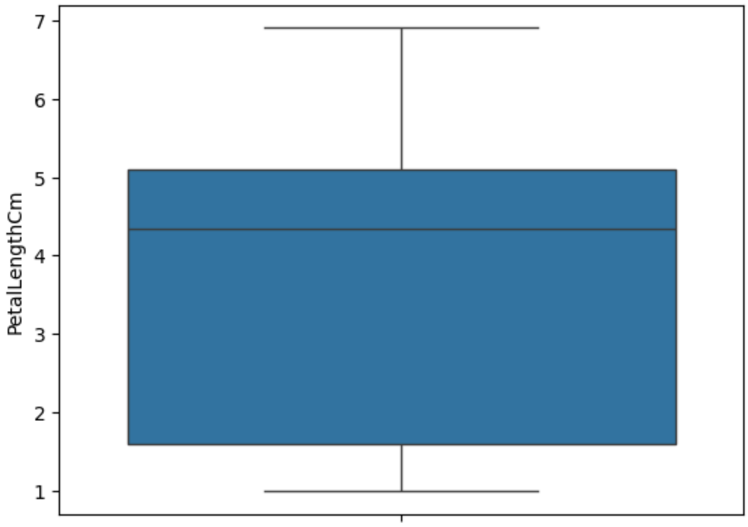
```
sns.lineplot(x=data['PetalLengthCm'],y=data['SepalWidthCm'])
```

```
<Axes: xlabel='PetalLengthCm', ylabel='SepalWidthCm'>
```



```
sns.boxplot(data['PetalLengthCm'])
```

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



```
numeric_columns=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']
```

```
mean=data[numeric_columns].mean()
mean
```

```
SepalLengthCm    5.843333
SepalWidthCm     3.054000
PetalLengthCm    3.758667
PetalWidthCm     1.198667
dtype: float64
```

```
median=data[numeric_columns].median()
median
```

```
SepalLengthCm    5.80
SepalWidthCm     3.00
PetalLengthCm    4.35
PetalWidthCm     1.30
dtype: float64
```

```
mode=data[numeric_columns].mode()
mode
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.0	3.0	1.5	0.2

```
x=data.drop('Species',axis=1)
```

```
y=data['Species']
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=12)
```

```
from sklearn import svm
```

```
model=svm.SVC(kernel='linear')
```

```
model=model.fit(x_train,y_train)
```

```
predict=model.predict(x_test)
```

```
from sklearn import metrics
```

```
metrics.accuracy_score(predict,y_test)
```

```
1.0
```

```
from sklearn import svm
```

```
model=svm.SVC(kernel='poly')
```

```
Model=model.fit(x_train,y_train)
```

```
predict=Model.predict(x_test)
```

```
predict
```

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

```
metrics.accuracy_score(predict,y_test)
```

```
0.9666666666666667
```

Start coding or generate with AI.