jupyter IRS EDA _Riya Last Checkpoint: 1 minute ago

File Edit View Run Kernel Settings Help

Not Trusted

Code

JupyterLab Python 3 (ipykernel)

[13]:
```
#for dtata extraction, manipulation and analysis
import pandas as pd
import numpy as py

# data visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Stats
from scipy.stats import mode
from scipy import stats

# train-test split
from sklearn.model_selection import train_test_split
```

[14]:
```
## Data Gathering
df = pd.read_json(r"C:\Users\riyac\Downloads\iris.json")
df
```

[14]:

	sepalLength	sepalWidth	petalLength	petalWidth	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

[15]:
```
df.shape # check for shape
```

[15]:
```
(150, 5)
```

[16]:
```
df.size # check for size
```

[16]:
```
750
```

[17]:
```
df.info ## check for size
```

[17]:
```
<bound method DataFrame.info of      sepalLength  sepalWidth  petalLength  petalWidth    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
..           ...         ...          ...         ...        ...
145          6.7         3.0          5.2         2.3  virginica
146          6.3         2.5          5.0         1.9  virginica
147          6.5         3.0          5.2         2.0  virginica
148          6.2         3.4          5.4         2.3  virginica
149          5.9         3.0          5.1         1.8  virginica

[150 rows x 5 columns]>
```

[18]:
```
df.describe()  ## check for stats
```

[18]:

	sepalLength	sepalWidth	petalLength	petalWidth
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

[21]:
```
df.isna().sum() # there  is no null value
```

[21]:
```
sepalLength    0
sepalWidth     0
petalLength    0
petalWidth     0
species        0
dtype: int64
```

[19]:
```
df.dtypes # check for datatype
```

[19]:
```
sepalLength    float64
sepalWidth     float64
petalLength    float64
petalWidth     float64
species         object
dtype: object
```

Analysis for SepalLength

Analysis for sepalLength

[23]:
```
df['sepalLength'].mean() # check  mean for col =  sepallength
```

[23]:
```
5.843333333333334
```

[27]:
```
df['sepalLength'].median() ## check  median for col =  sepallength
```

[27]:
```
5.8
```

[29]:
```
Mode_sepallength=mode(df["sepalLength"])  ## check mode for col =  sepallength
Mode_sepallength
```

[29]:
```
ModeResult(mode=5.0, count=10)
```

[47]:
```
df['sepalLength'].var()
```

[47]:
```
0.6856935123042505
```

[49]:
```
df['sepalLength'].std()
```

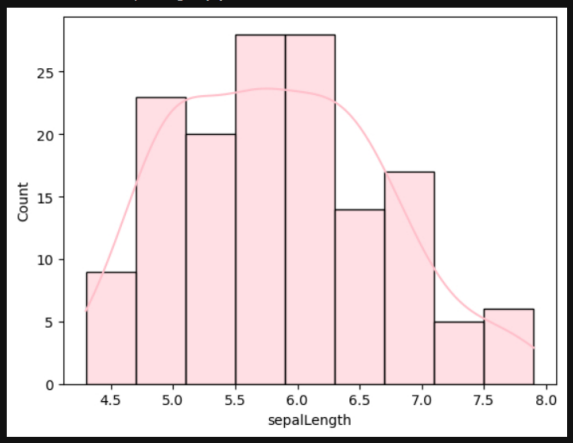
[49]:
```
0.8280661279778629
```

[51]:
```
df['sepalLength'].skew()
```

[51]:
```
0.3149109566369728
```

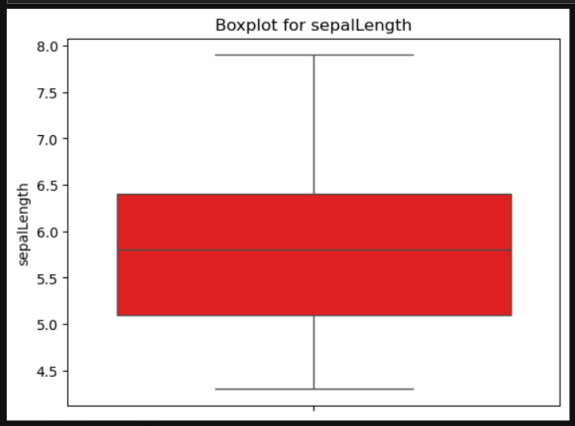
[59]:
```
sns.histplot(df["sepalLength"], kde = True, color = "pink")
```

[59]:
```
<Axes: xlabel='sepalLength', ylabel='Count'>
```



[61]:
```
def Checking_and_Handling_Of_Outliers(df, col):
    sns.boxplot(df[col], color = "Red")
    plt.title(f"Boxplot for {col}")
    plt.show()

    q1 = df[col].quantile(0.25)
    q3 = df[col].quantile(0.75)

    iqr = q3 - q1

    LowerTail = q1 - 1.5*iqr
    UpperTail = q3 + 1.5*iqr

    print(f"25% Quantile q1 = {q1}\n75% Quantile q3 = {q3}\nIQR = {iqr}\n")
    print("-"*80)
    print(f"Lower Tail = {LowerTail}\nUpper Tail = {UpperTail}")
    print("-"*80)

    # Checking for Outliers
    Outliers = df[(df[col] < LowerTail) | (df[col] > UpperTail)]
    print("\nOutliers :\n",Outliers)
    print("-"*80)
    ##Handling of Outliers :
    df.loc[df[col] < LowerTail, col] = LowerTail # all outliers less than lowertail, assigned by lowertail value
    df.loc[df[col] > UpperTail, col] = UpperTail # all outliers greater than uppertail, assigned by uppertail value
    print("After handling of Outliers data:\n")
    print(df.head())
```

[74]:
```
Checking_and_Handling_Of_Outliers(df,'sepalLength')
```




```
25% Quantile q1 = 5.1
75% Quantile q3 = 6.4
IQR = 1.3000000000000007
```

```
--------------------------------------------------------------------------------
Lower Tail = 3.1499999999999986
Upper Tail = 8.350000000000001
--------------------------------------------------------------------------------

Outliers :
 Empty DataFrame
Columns: [sepalLength, sepalWidth, petalLength, petalWidth, species]
Index: []
--------------------------------------------------------------------------------
After handling of Outliers data:

   sepalLength  sepalWidth  petalLength  petalWidth 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
```

Analysis for sepalwidth

[31]:
```
df['sepalWidth'].mean()  ## check  mean for col = sepalwidth
```

[31]:
```
3.0573333333333337
```

[33]:
```
df['sepalWidth'].median()  ## check  median for col = sepalwidth
```

[33]:
```
3.0
```

[35]:
```
Mode_sepalWidth=mode(df["sepalWidth"])  ## ## check  mode for col = sepalwidth
Mode_sepalWidth
```

[35]:
```
ModeResult(mode=3.0, count=26)
```

[83]:
```
df['sepalWidth'].var()
```

[83]:
```
0.1899794183445188
```

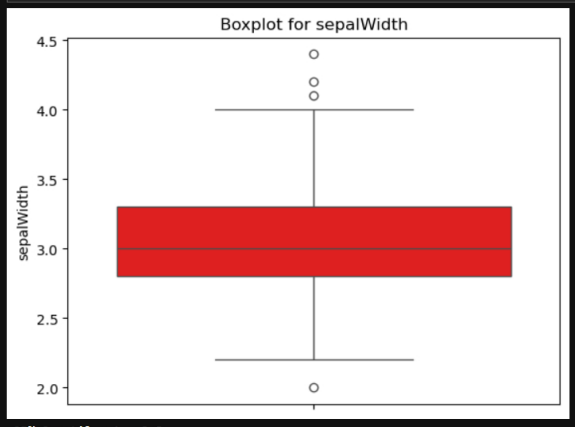
[85]:
```
df['sepalWidth'].std()
```

[85]:
```
0.435866284936698
```

[87]:
```
df['sepalWidth'].skew()
```

[87]:
```
0.31896566471359966
```

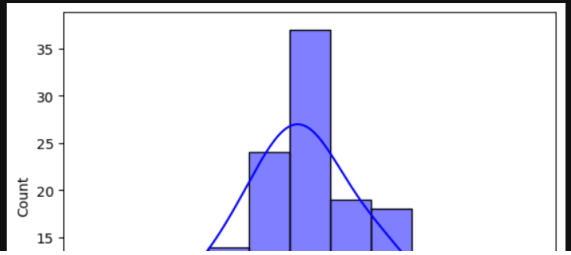
[91]:
```
Checking_and_Handling_Of_Outliers(df,'sepalWidth')
```

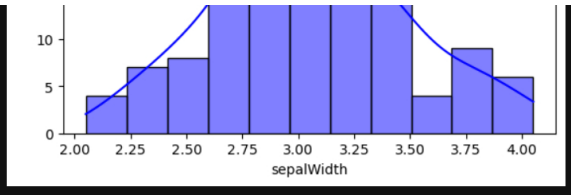



```
25% Quantile q1 = 2.8
75% Quantile q3 = 3.3
IQR = 0.5

--------------------------------------------------------------------------------
Lower Tail = 2.05
Upper Tail = 4.05
--------------------------------------------------------------------------------

Outliers :
     sepalLength  sepalWidth  petalLength  petalWidth     species
15           5.7         4.4          1.5         0.4      setosa
32           5.2         4.1          1.5         0.1      setosa
33           5.5         4.2          1.4         0.2      setosa
60           5.0         2.0          3.5         1.0  versicolor
--------------------------------------------------------------------------------
After handling of Outliers data:

   sepalLength  sepalWidth  petalLength  petalWidth 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
```

[106]:
```
sns.histplot(df["sepalWidth"], kde = True, color = "blue")
```

[106]:
```
<Axes: xlabel='sepalWidth', ylabel='Count'>
```





Analysis for petallength

[37]:
```
df['petalLength'].mean()  # check mean for col = petallength
```

[37]:
```
3.7580000000000005
```

[39]:
```
df['petalLength'].median() # check median for col = petallength
```

[39]:
```
4.35
```

[41]:
```
Mode_petalLength=mode(df["petalLength"])  # check mode  for col = petallength
Mode_petalLength
```

[41]:
```
ModeResult(mode=1.4, count=13)
```

[70]:
```
df['petalLength'].var()
```

[70]:
```
3.1162778523489942
```

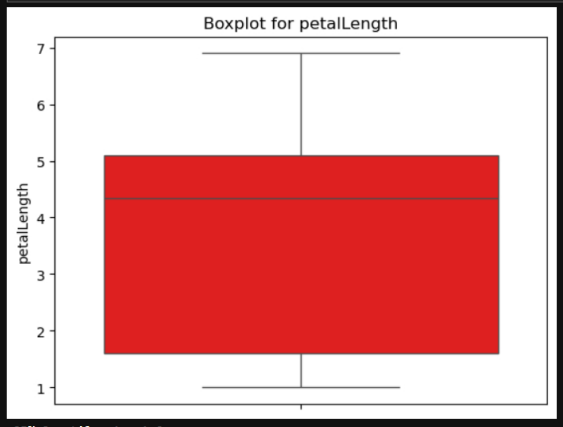
[68]:
```
df['petalLength'].std()
```

[68]:
```
1.7652982332594667
```

[72]:
```
df['petalLength'].skew()
```

[72]:
```
-0.2748841797510126
```

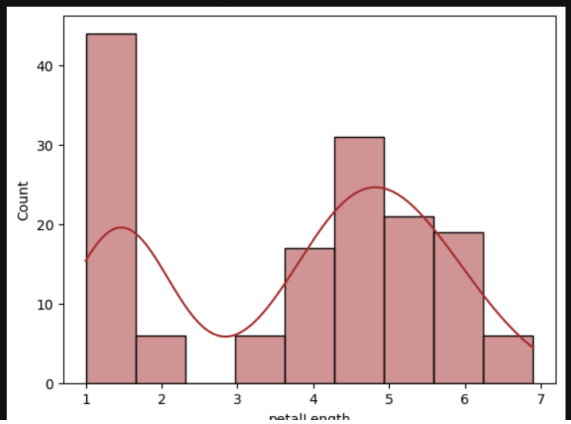
[96]:
```
Checking_and_Handling_Of_Outliers(df,'petalLength')
```




```
25% Quantile q1 = 1.6
75% Quantile q3 = 5.1
IQR = 3.4999999999999996

--------------------------------------------------------------------------------
Lower Tail = -3.649999999999999
Upper Tail = 10.349999999999998
--------------------------------------------------------------------------------

Outliers :
 Empty DataFrame
Columns: [sepalLength, sepalWidth, petalLength, petalWidth, species]
Index: []
--------------------------------------------------------------------------------
After handling of Outliers data:

   sepalLength  sepalWidth  petalLength  petalWidth 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
```

[108]:
```
sns.histplot(df["petalLength"], kde = True, color = "brown")
```

[108]:
```
<Axes: xlabel='petalLength', ylabel='Count'>
```



Analysis for petalwidth

[43]:
```
df['petalWidth'].mean()  # # check mean for col = petalwidth
```

[43]:
```
3.7580000000000005
```

[45]:
```
df['petalWidth'].median()  # # check median for col = petalwidth
```

[45]:
```
1.3
```

[49]:
```
Mode_petalWidth=mode(df["petalWidth"])   # # check mode  for col = petalwidth
Mode_petalWidth
```

[49]:
```
ModeResult(mode=0.2, count=29)
```

[38]:
```
df['petalLength'].var()
```

[38]:
```
3.1162778523489942
```

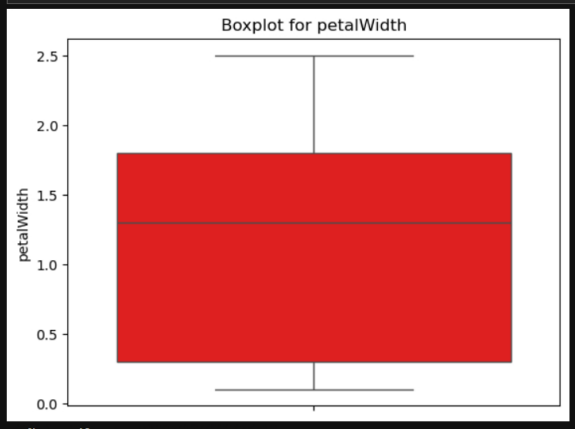
[40]:
```
df['petalLength'].std()
```

[40]:
```
1.7652982332594667
```

[42]:
```
df['petalLength'].skew()
```

[42]:
```
-0.2748841797510126
```

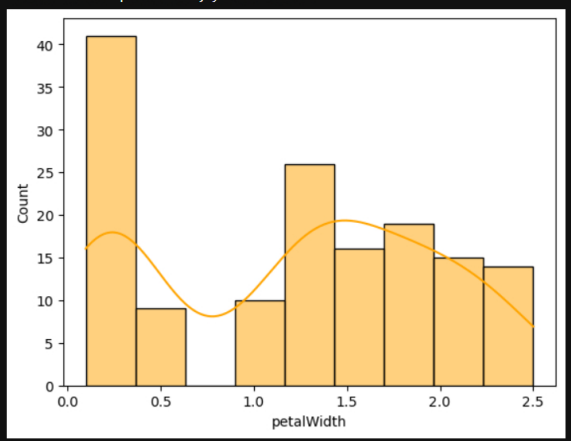
[104]:
```
Checking_and_Handling_Of_Outliers(df,'petalWidth')
```




```
25% Quantile q1 = 0.3000000000000004
75% Quantile q3 = 1.8
IQR = 1.5

--------------------------------------------------------------------------------
Lower Tail = -1.95
Upper Tail = 4.05
--------------------------------------------------------------------------------

Outliers :
 Empty DataFrame
Columns: [sepalLength, sepalWidth, petalLength, petalWidth, species]
Index: []
--------------------------------------------------------------------------------
After handling of Outliers data:

   sepalLength  sepalWidth  petalLength  petalWidth 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
```

[110]:
```
sns.histplot(df["petalWidth"], kde = True, color = "orange")
```

[110]:
```
<Axes: xlabel='petalWidth', ylabel='Count'>
```



[120]:
```
df["petalWidth"].max() - df["sepalLength"].min()
```

[120]:
```
-1.7999999999999998
```

[124]:
```
# Bivariate Analysis - covariance between petalWidth and sepallength column
covariance = df.iloc[:, [0,2]].cov()
covariance
```

[124]:

	sepalLength	petalLength
sepalLength	0.685694	1.274315
petalLength	1.274315	3.116278

[130]:
```
q1 = df["sepalLength"].quantile(0.25)
q2 = df["sepalLength"].quantile(0.50)
q3 = df["sepalLength"].quantile(0.75)
```

[132]:
```
iqr = q3 - q1
iqr
```

[132]: 1.3000000000000007

[134]:
```
LowerTail = q1 - 1.5*iqr
LowerTail
```

[134]: 3.1499999999999986

[136]:
```
UpperTail = q3 + 1.5*iqr
UpperTail
```

[136]: 8.350000000000001

[140]:
```
outliers = df[(df["sepalLength"] < LowerTail) & (df["sepalLength"] > UpperTail)]
outliers
```

[140]:

	sepalLength	sepalWidth	petalLength	petalWidth	species

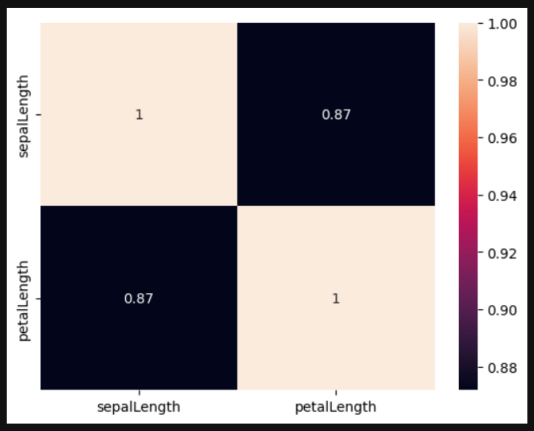
[144]:
```
# Bivariate Analysis - corr between sepalLength and petalLength column

corr = df.iloc[:, [0,2]].corr()
corr
```

[144]:

	sepalLength	petalLength
sepalLength	1.000000	0.871754
petalLength	0.871754	1.000000

[148]:
```
sns.heatmap(corr, annot=True)
```

[148]: <Axes: >



Analysis for species column

[174]:
```
df["species"].nunique()
```

[174]: 3

[176]:
```
df["species"].unique()
```

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

[178]:
```
# ecoding

df["species"].replace({'setosa' : 1, 'versicolor' :2 , 'virginica':3}, inplace = True)
```

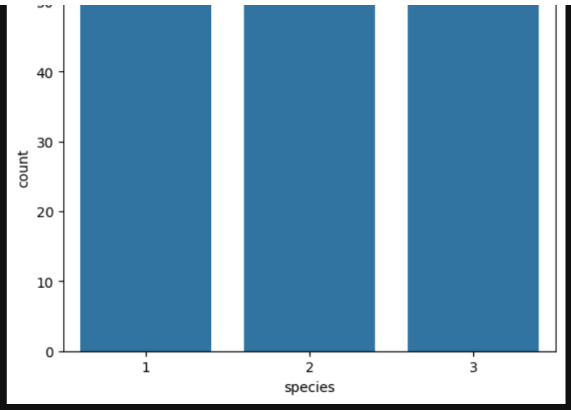
```
C:\Users\riyac\AppData\Local\Temp\ipykernel_19568\4080611295.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df["species"].replace({'setosa' : 1, 'versicolor' :2 , 'virginica':3}, inplace = True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_19568\4080611295.py:3: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a
future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('futur
e.no_silent_downcasting', True)`
  df["species"].replace({'setosa' : 1, 'versicolor' :2 , 'virginica':3}, inplace = True)
```

[180]:
```
df["species"].value_counts()
```

[180]:
```
species
1    50
2    50
3    50
Name: count, dtype: int64
```

[184]:
```
sns.countplot(x = df["species"])
```

[184]: <Axes: xlabel='species', ylabel='count'>



[186]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   sepalLength  150 non-null    float64
 1   sepalWidth   150 non-null    float64
 2   petalLength  150 non-null    float64
 3   petalWidth   150 non-null    float64
 4   species      150 non-null    int64
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

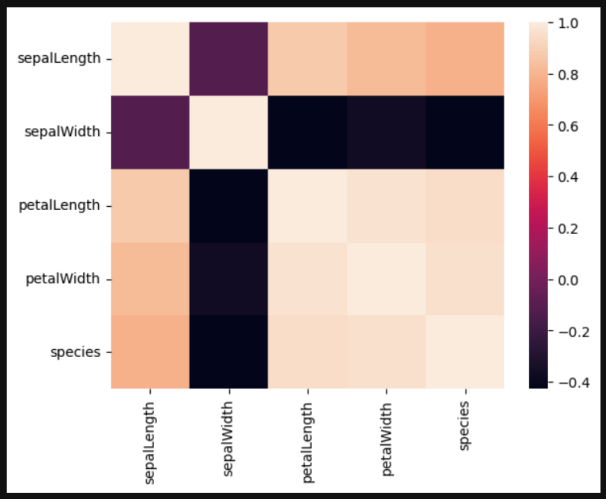
[188]:
```
df.corr()
```

[188]:

	sepalLength	sepalWidth	petalLength	petalWidth	species
sepalLength	1.000000	-0.118719	0.871754	0.817941	0.782561
sepalWidth	-0.118719	1.000000	-0.427873	-0.365317	-0.426572
petalLength	0.871754	-0.427873	1.000000	0.962865	0.949035
petalWidth	0.817941	-0.365317	0.962865	1.000000	0.956547
species	0.782561	-0.426572	0.949035	0.956547	1.000000

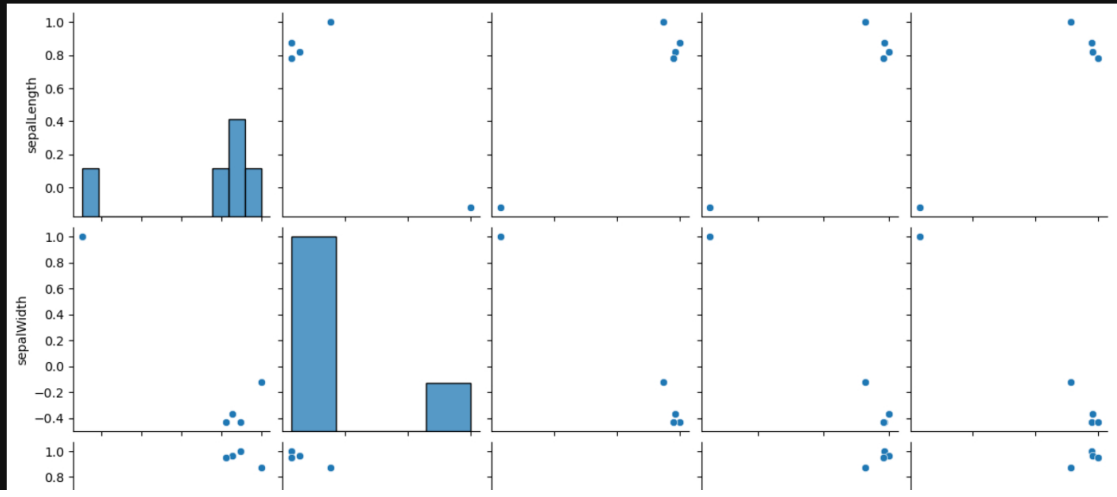
[190]:
```
sns.heatmap(df.corr())
```

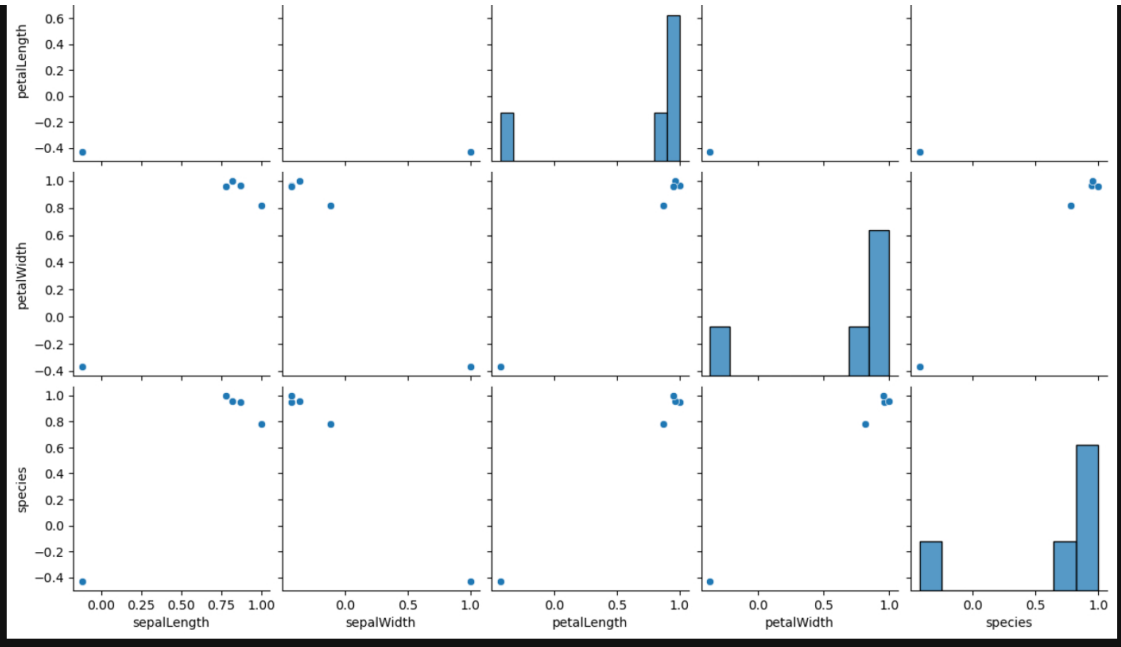
[190]:
```
<Axes: >
```



[192]:
```
sns.pairplot(df.corr())
```

[192]:
```
<seaborn.axisgrid.PairGrid at 0x2300c066060>
```





train test split

[194]:
```
x = df.iloc[:, :5]
y = df["sepalLength"]
```

[196]:
```
y.head()
```

[196]:
```
0    5.1
1    4.9
2    4.7
3    4.6
4    5.0
Name: sepalLength, dtype: float64
```

[198]:
```
xtrain, xtest, ytrain, ytest = train_test_split(x,y, test_size=0.3)
```

[200]:
```
xtrain.shape
```

[200]:
```
(105, 5)
```

[202]:
```
xtest.shape
```

[202]:
```
(45, 5)
```

[204]:
```
ytrain.shape
```

[204]:
```
(105,)
```

[206]:
```
ytest.shape
```

[206]:
```
(45,)
```

[ ]: