```
from google.colab import drive # Mount Google Drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

```
import pandas as pd
df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/iris.csv")
```

Discriptive Statistics

```
df.mean() #Mean value
```

```
<ipython-input-3-2bc51fe1e1eb>:1: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it
will default to False. In addition, specifying 'numeric_only=None' is
deprecated. Select only valid columns or specify the value of
numeric_only to silence this warning.
  df.mean() #Mean value
```

```
sepal.length    5.843333
sepal.width     3.057333
petal.length    3.758000
petal.width     1.199333
dtype: float64
```

```
df.head()
```

	sepal.length	sepal.width	petal.length	petal.width	variety
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

```
df.count()
```

```
sepal.length    150
sepal.width     150
petal.length    150
petal.width     150
variety         150
dtype: int64
```

```
df.sum()
```

```
sepal.length                                                  876.5
sepal.width                                                   458.6
petal.length                                                  563.7
petal.width                                                   179.9
```

```
variety         SetosaSetosaSetosaSetosaSetosaSetosaSetosaSeto...
dtype: object
```

```
df.median()
```

```
<ipython-input-7-6d467abf240d>:1: FutureWarning: The default value of
numeric_only in DataFrame.median is deprecated. In a future version,
it will default to False. In addition, specifying 'numeric_only=None'
is deprecated. Select only valid columns or specify the value of
numeric_only to silence this warning.
  df.median()
```

```
sepal.length    5.80
sepal.width     3.00
petal.length    4.35
petal.width     1.30
dtype: float64
```

```
df.mode()
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.0	3.0	1.4	0.2	Setosa
1	NaN	NaN	1.5	NaN	Versicolor
2	NaN	NaN	NaN	NaN	Virginica

```
df.std()
```

```
<ipython-input-9-ce97bb7eaef8>:1: FutureWarning: The default value of
numeric_only in DataFrame.std is deprecated. In a future version, it
will default to False. In addition, specifying 'numeric_only=None' is
deprecated. Select only valid columns or specify the value of
numeric_only to silence this warning.
  df.std()
```

```
sepal.length    0.828066
sepal.width     0.435866
petal.length    1.765298
petal.width     0.762238
dtype: float64
```

```
df.min()
```

```
sepal.length       4.3
sepal.width        2.0
petal.length       1.0
petal.width        0.1
variety         Setosa
dtype: object
```

```
df.max()
```

```
sepal.length          7.9
sepal.width           4.4
petal.length          6.9
petal.width           2.5
variety         Virginica
dtype: object
```

```
df['sepal.length'].abs()
```

```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
      ...
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: sepal.length, Length: 150, dtype: float64
```

```
df.prod()
```

```
<ipython-input-13-7c1ba80be652>:1: FutureWarning: The default value of
numeric_only in DataFrame.prod is deprecated. In a future version, it
will default to False. In addition, specifying 'numeric_only=None' is
deprecated. Select only valid columns or specify the value of
numeric_only to silence this warning.
  df.prod()
```

```
sepal.length    2.257440e+114
sepal.width      1.390618e+72
petal.length     3.522857e+76
petal.width      5.945429e-12
dtype: float64
```

```
df.cumsum()
```

```
     sepal.length  sepal.width  petal.length  petal.width  \
0             5.1          3.5           1.4          0.2
1            10.0          6.5           2.8          0.4
2            14.7          9.7           4.1          0.6
3            19.3         12.8           5.6          0.8
4            24.3         16.4           7.0          1.0
..            ...          ...           ...          ...
145         851.6        446.7         543.0        171.9
146         857.9        449.2         548.0        173.8
147         864.4        452.2         553.2        175.8
148         870.6        455.6         558.6        178.1
149         876.5        458.6         563.7        179.9
```

```
                                                  variety
0                                                  Setosa
1                                           SetosaSetosa
2                                    SetosaSetosaSetosa
3                             SetosaSetosaSetosaSetosa
4                      SetosaSetosaSetosaSetosaSetosa
..                                                   ...
145  SetosaSetosaSetosaSetosaSetosaSetosaSetosaSeto...
146  SetosaSetosaSetosaSetosaSetosaSetosaSetosaSeto...
147  SetosaSetosaSetosaSetosaSetosaSetosaSetosaSeto...
148  SetosaSetosaSetosaSetosaSetosaSetosaSetosaSeto...
149  SetosaSetosaSetosaSetosaSetosaSetosaSetosaSeto...

[150 rows x 5 columns]
```

```
df['petal.length'].cumprod()
```

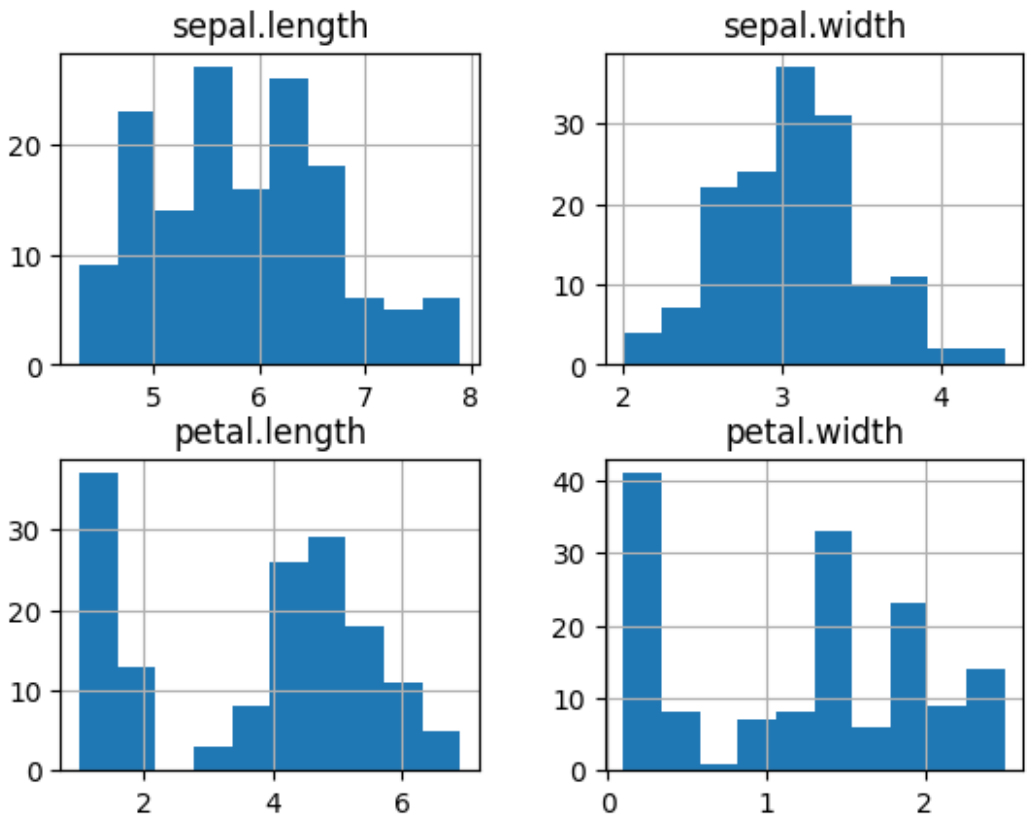
```
0      1.400000e+00
1      1.960000e+00
2      2.548000e+00
3      3.822000e+00
4      5.350800e+00
           ...
145    4.919916e+73
146    2.459958e+74
147    1.279178e+75
148    6.907562e+75
149    3.522857e+76
Name: petal.length, Length: 150, dtype: float64
```

```
df.describe()
```

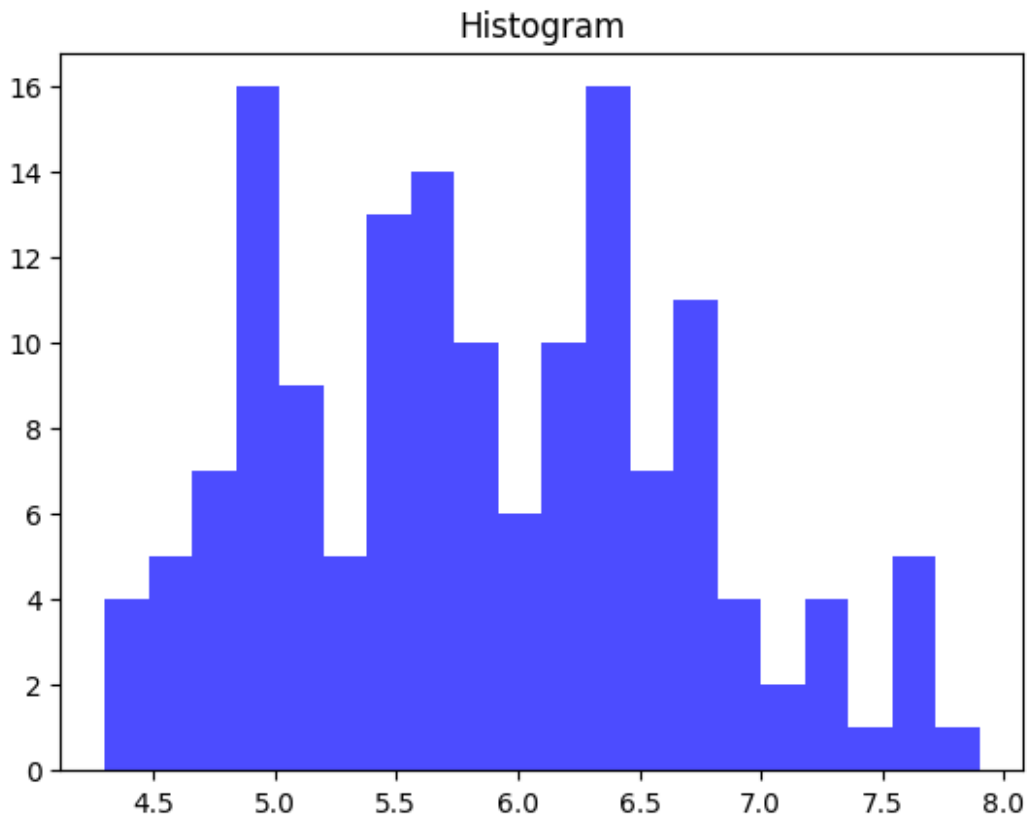
```
       sepal.length  sepal.width  petal.length  petal.width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
min        4.300000     2.000000      1.000000     0.100000
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
```

Explorartive Data Analysis

```
import matplotlib.pyplot as plt
df.hist() #Histogram
plt.show()
```

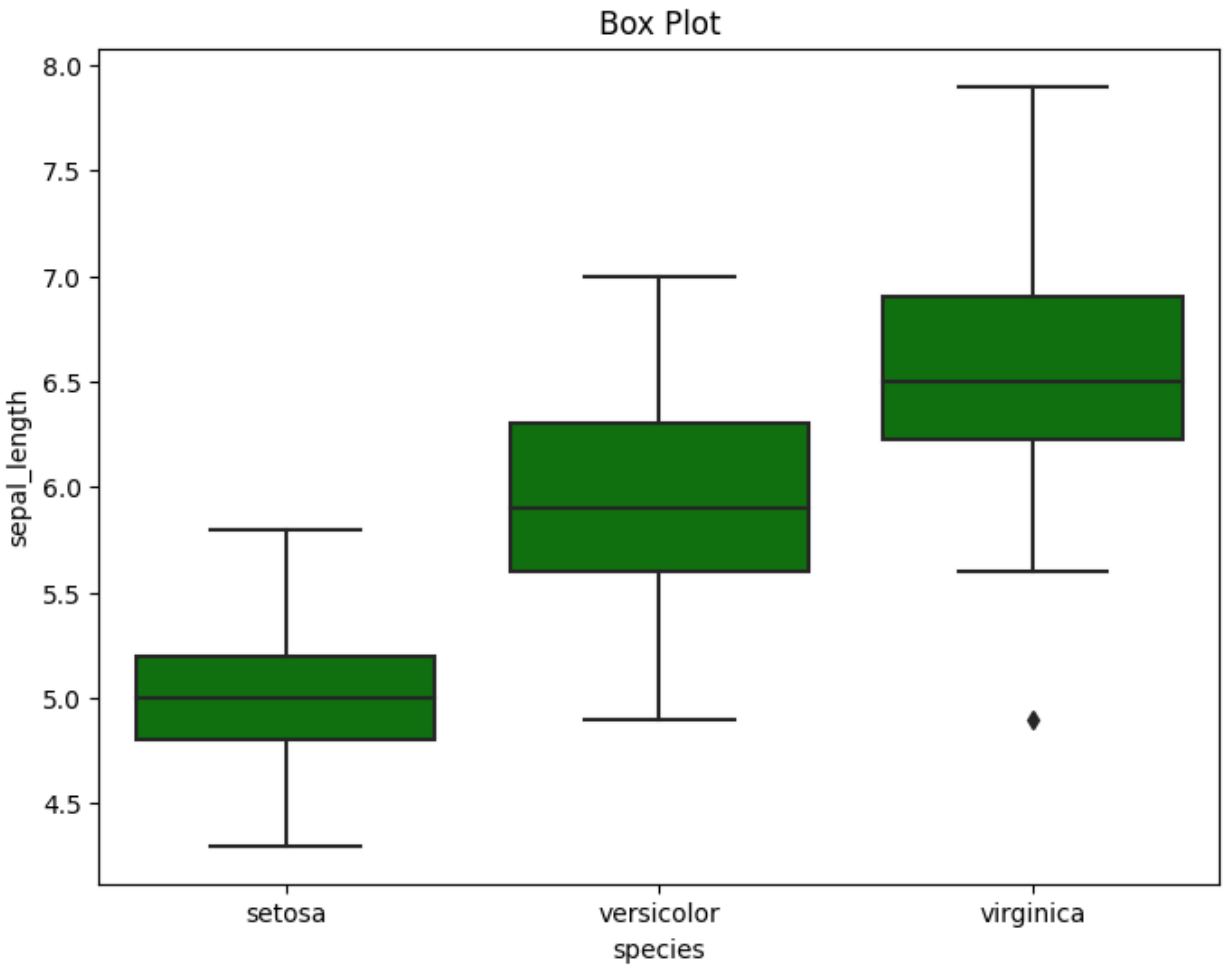



```
import matplotlib.pyplot as plt
data = df['sepal.length']
plt.hist(data,bins=20,color='blue',alpha=0.7)
plt.title('Histogram')
plt.show()
```



Histogram :It was found that sepal length is higher in 4.9 and 6.4

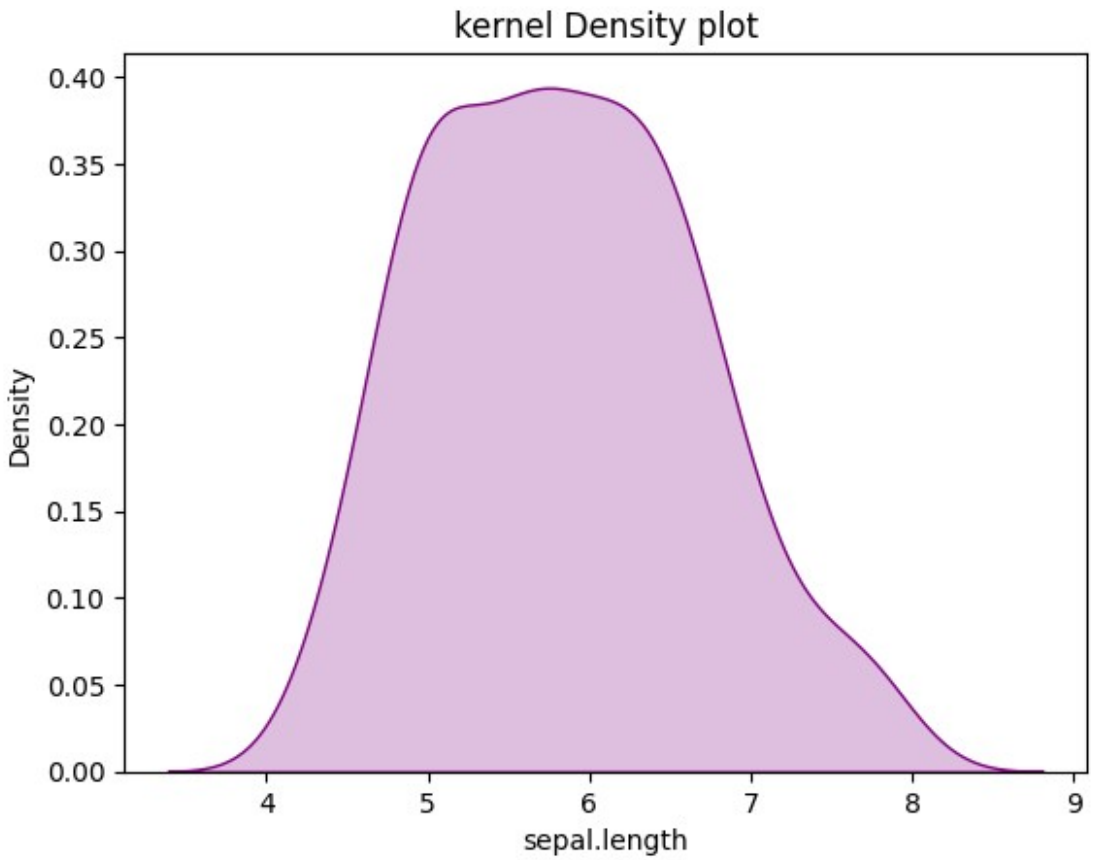
```
import seaborn as sns
import matplotlib.pyplot as plt
iris = sns.load_dataset('iris')
plt.figure(figsize=(8, 6))
sns.boxplot(x='species', y='sepal_length', data=iris, color='green')
plt.title('Box Plot')
plt.show()
```



It was found that there is no Outliers in setosa and versicolor and 1 outlier in virginica

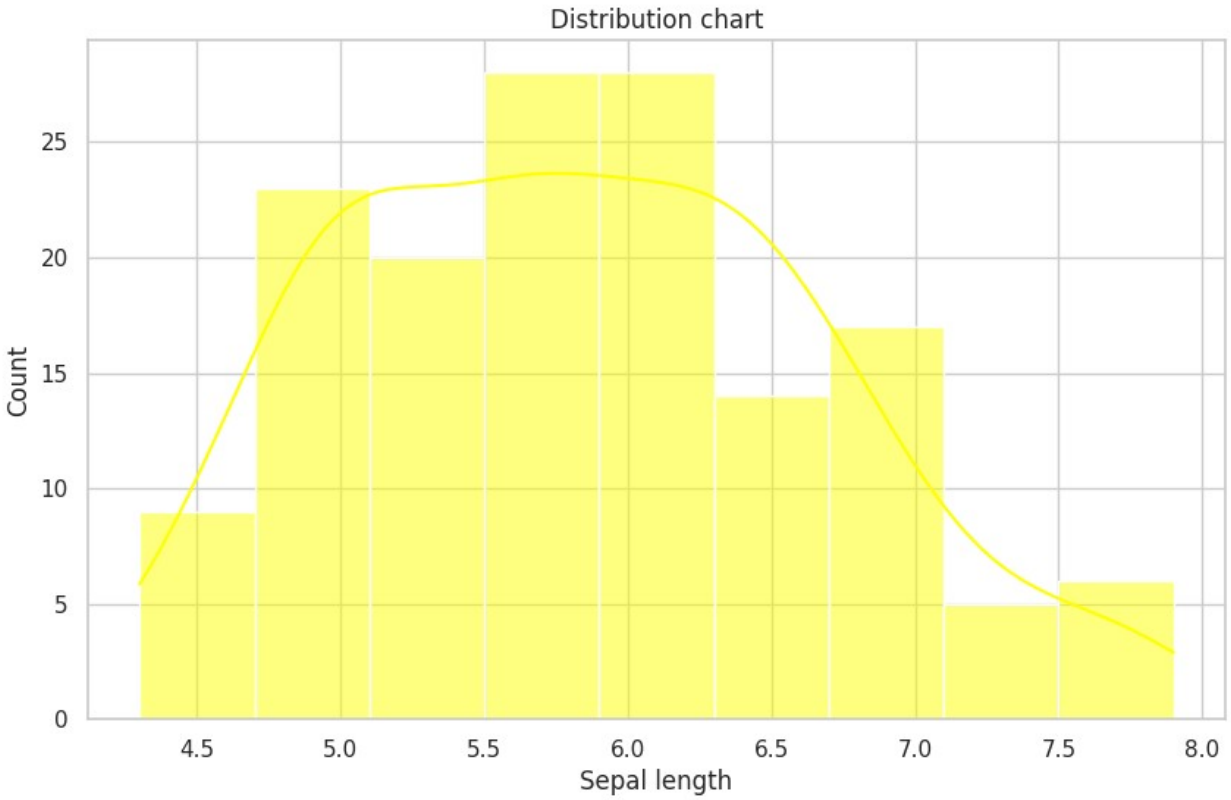
```
sns.kdeplot(data,shade=True,color='purple')
plt.title('kernel Density plot')
plt.show()
```

```
<ipython-input-20-dea7aa9d1642>:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data,shade=True,color='purple')
```



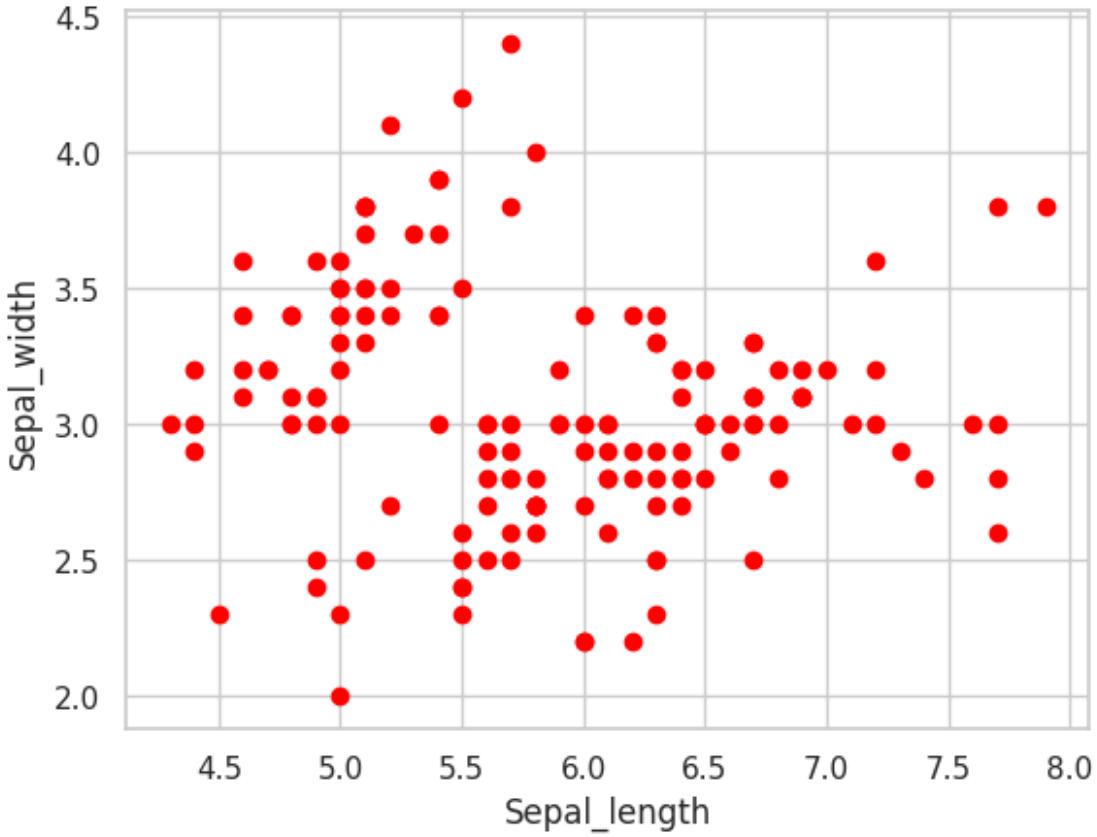
Density Chart:sepal.length has higher value in the range between 5 and 7

```
sns.set(style='whitegrid')
plt.figure(figsize=(10,6))
sns.histplot(data=df,x='sepal.length',kde=True,color='yellow')
plt.xlabel('Sepal length')
plt.title('Distribution chart')
plt.show()
```



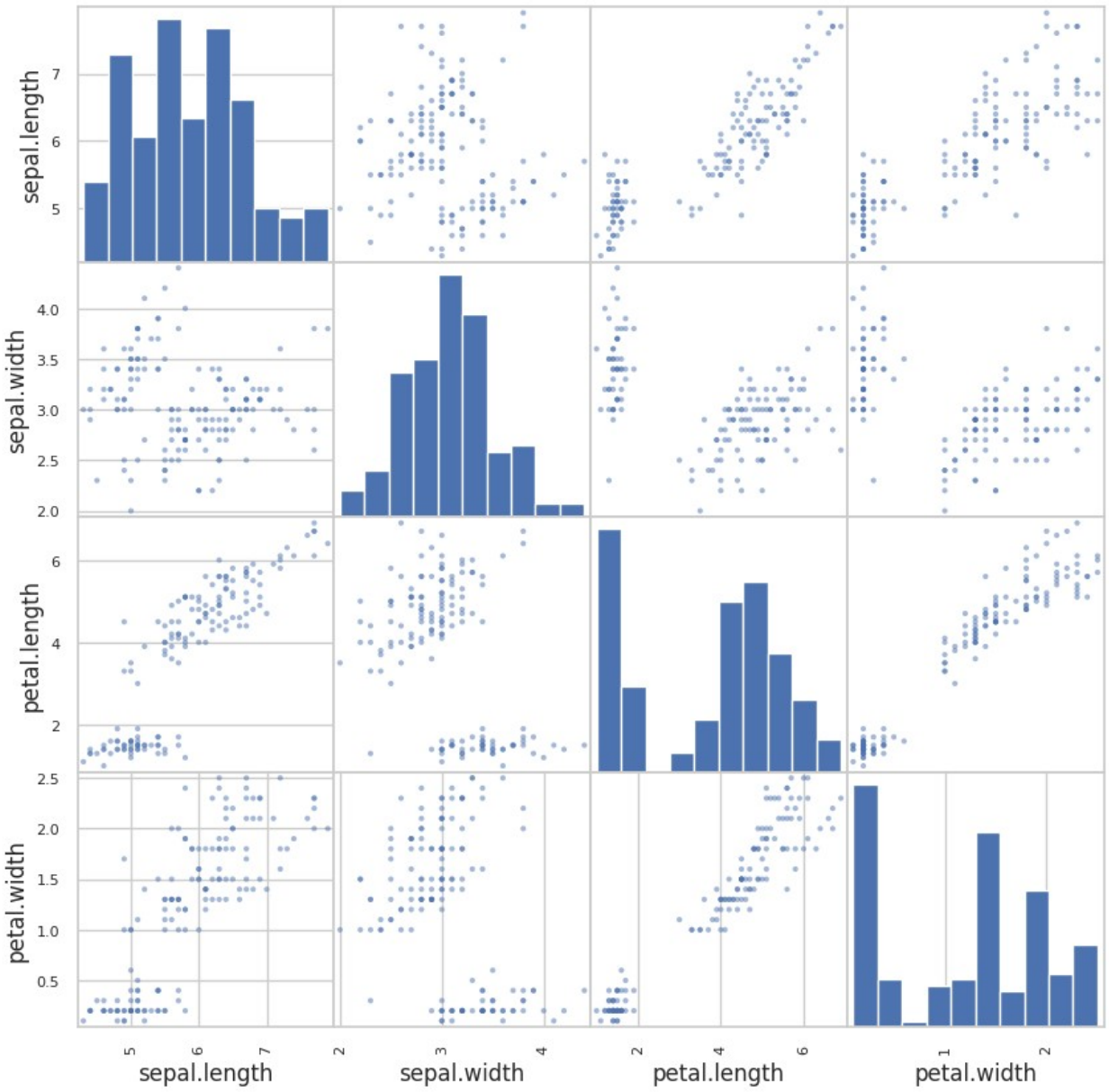
Distribution Chart:sepal.length has higher value in between 5.5 and 6.3

```
data1=df['sepal.length']
data2=df['sepal.width']
plt.scatter(data1,data2,color='red',marker='o')
plt.xlabel('Sepal_length')
plt.ylabel('Sepal_width')
plt.show()
```

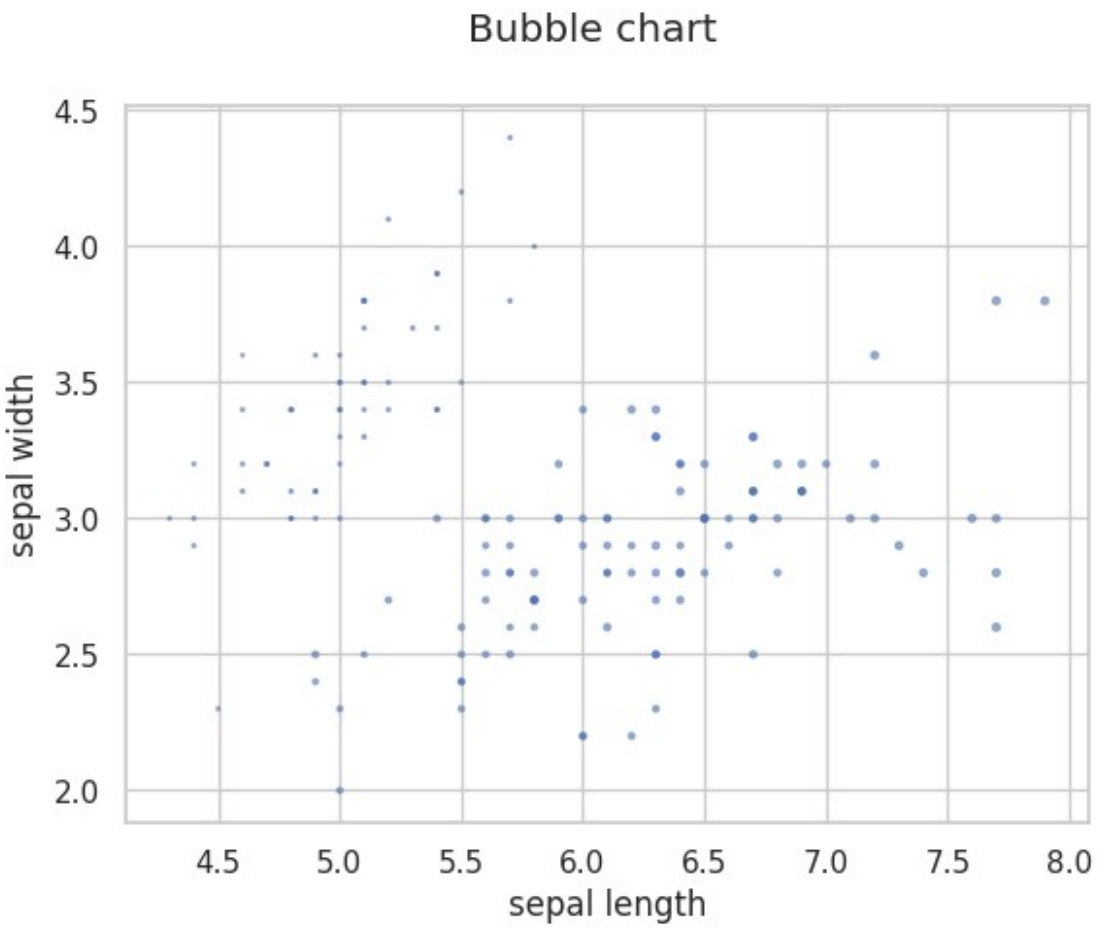



```
from pandas.plotting import scatter_matrix
scatter_matrix(df,figsize=(10,10))
plt.suptitle('Scatter Matrix')
plt.show()
```

Scatter Matrix



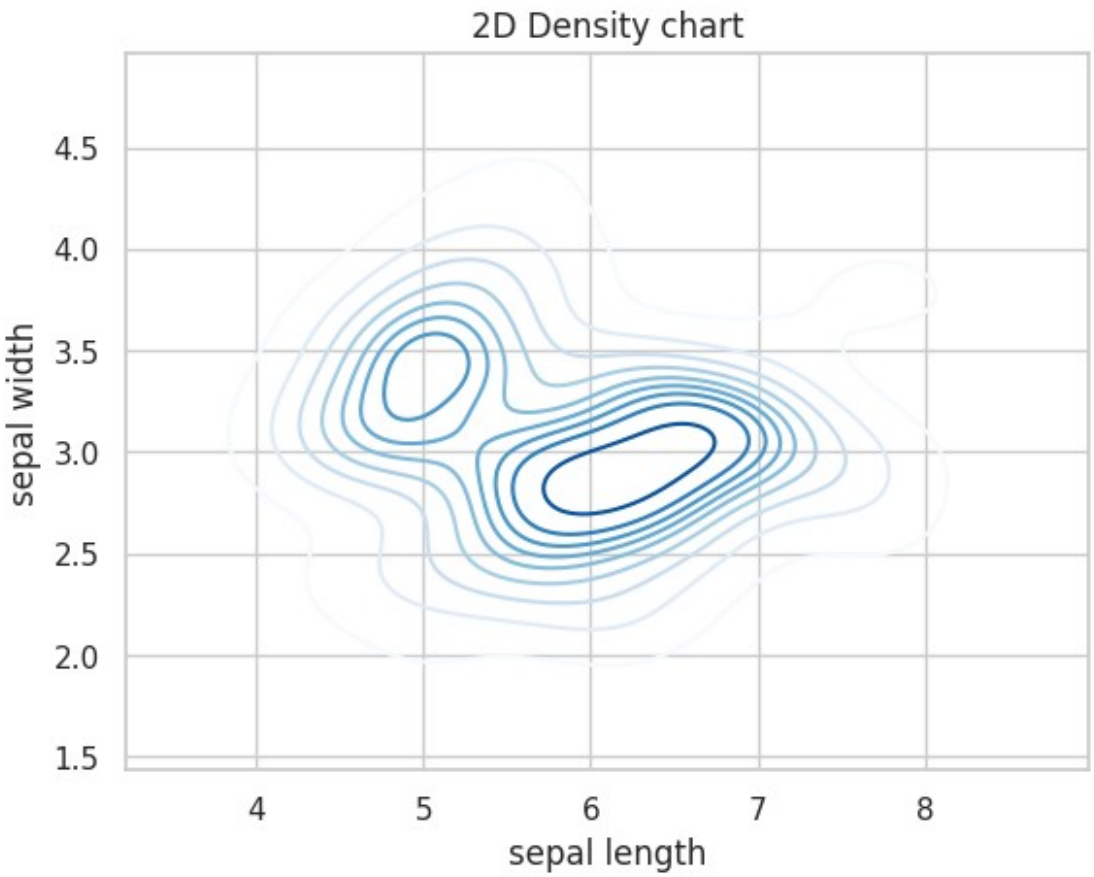

```
import matplotlib.pyplot as plt

plt.scatter(df['sepal.length'], df['sepal.width'],
s=df['petal.length'], alpha=0.5)
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.suptitle('Bubble chart')
plt.show()
```



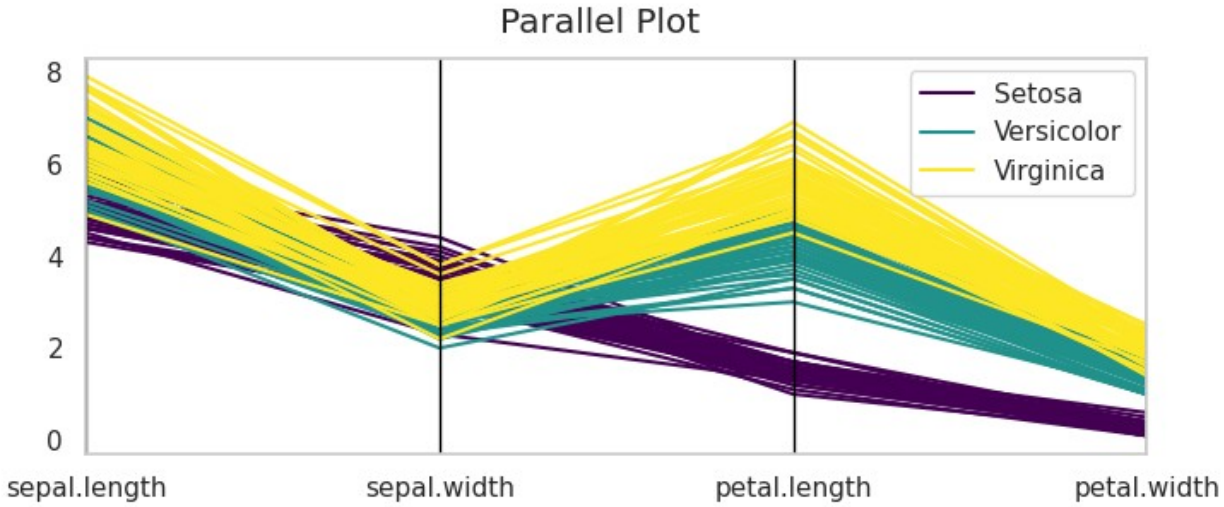
Bubble Chart:sepal length and sepal width are used as the x and y coordinates for the data points, respectively.petal.length specifies the size of each bubble. Bigger values of 'petal.length' will result in larger bubbles, while smaller values will result in smaller bubbles.

```
sns.kdeplot(data=df,x='sepal.length',y='sepal.width',cmap='Blues',Shad
e=True) #to create a Kernel Density Estimation (KDE) plot
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.title('2D Density chart')
plt.show()
```

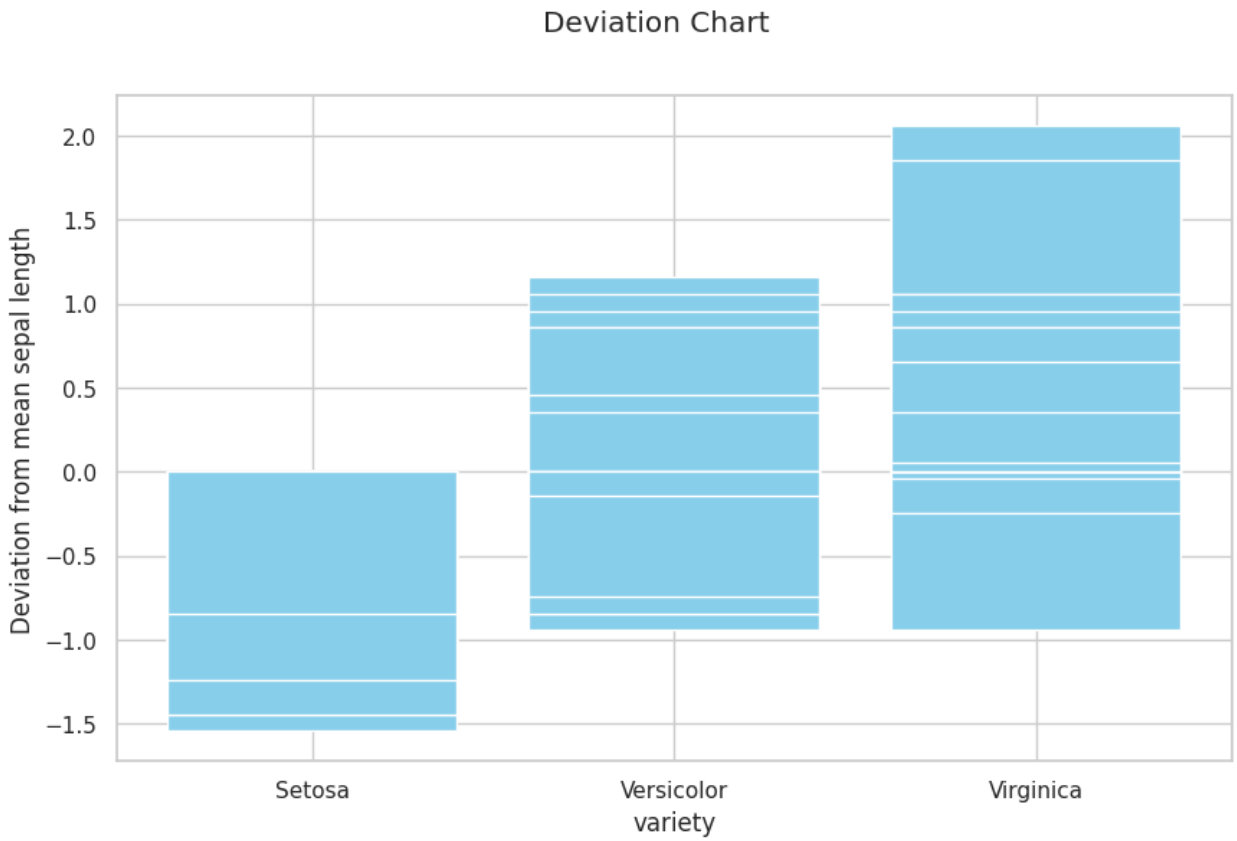
```
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:1185:
UserWarning: The following kwargs were not used by contour: 'Shade'
  cset = contour_func(
```




```
plt.figure(figsize=(8,3))
pd.plotting.parallel_coordinates(df,'variety',colormap='viridis')
plt.title=('parallel plot')
plt.suptitle('Parallel Plot')
plt.show()
```



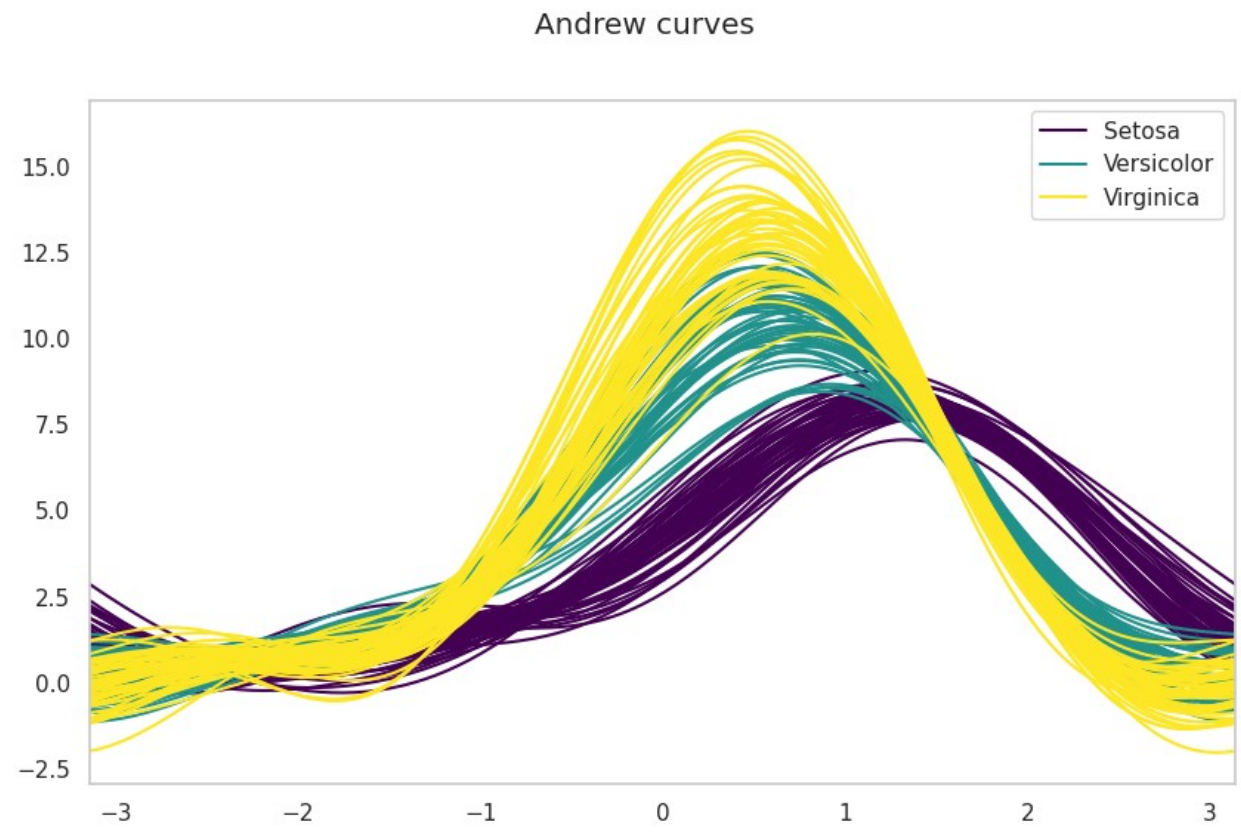
```
reference=df['sepal.length'].mean()
df['deviation']=df['sepal.length'] - reference
plt.figure(figsize=(10,6))
plt.bar(df['variety'],df['deviation'],color='skyblue')
plt.xlabel('variety')
plt.ylabel('Deviation from mean sepal length')
plt.suptitle('Deviation Chart')
plt.show()
```




```
import matplotlib.pyplot as plt

reference = df['sepal.length'].mean()
df['deviation'] = df['sepal.length'] - reference

plt.figure(figsize=(10, 6))
bars = plt.bar(df['variety'], df['deviation'], color='lightcoral')

plt.xlabel('Variety')
plt.ylabel('Deviation from mean sepal length')
plt.suptitle('Deviation Chart')


for i, deviation in enumerate(df['deviation']):
```

```
    plt.text(i, deviation, f'{deviation:.2f}', ha='center',
va='bottom')

plt.show()
```

```
from pandas.plotting import andrews_curves
plt.figure(figsize=(10,6))
andrews_curves(df,'variety',colormap='viridis')
plt.suptitle('Andrew curves')
plt.show()
```




```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 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   variety       150 non-null    object
 5   deviation     150 non-null    float64
dtypes: float64(5), object(1)
memory usage: 7.2+ KB
```