AIM

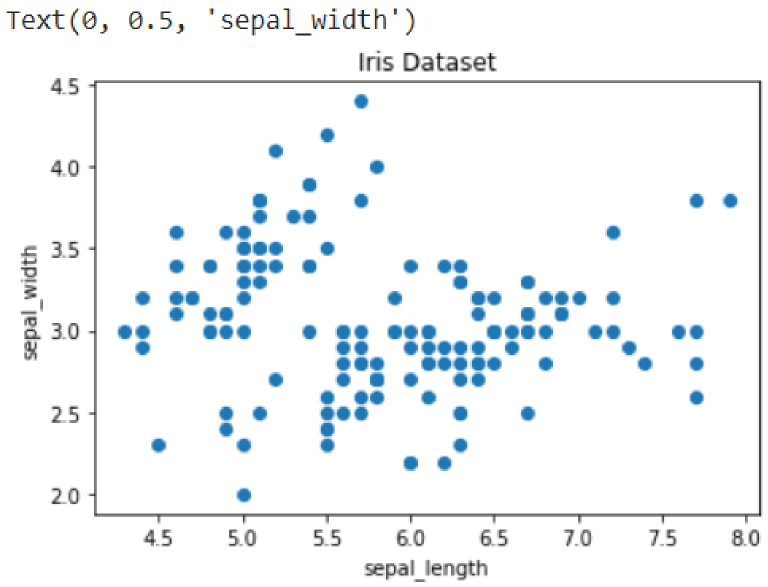
Programs using matplotlib / plotly / bokeh / seaborn for data visualisation.

Dataset used: iris.csv

Programming Code:

```
import pandas as pd
iris = pd.read_csv('/content/iris.csv')

import matplotlib.pyplot as plt
fig, ax = plt.subplots()

# scatter the sepal_length against the sepal_width
ax.scatter(iris['sepal.length'], iris['sepal.width'])
# set a title and labels
ax.set_title('Iris Dataset')
ax.set_xlabel('sepal_length')
ax.set_ylabel('sepal_width')
```

OUTPUT:

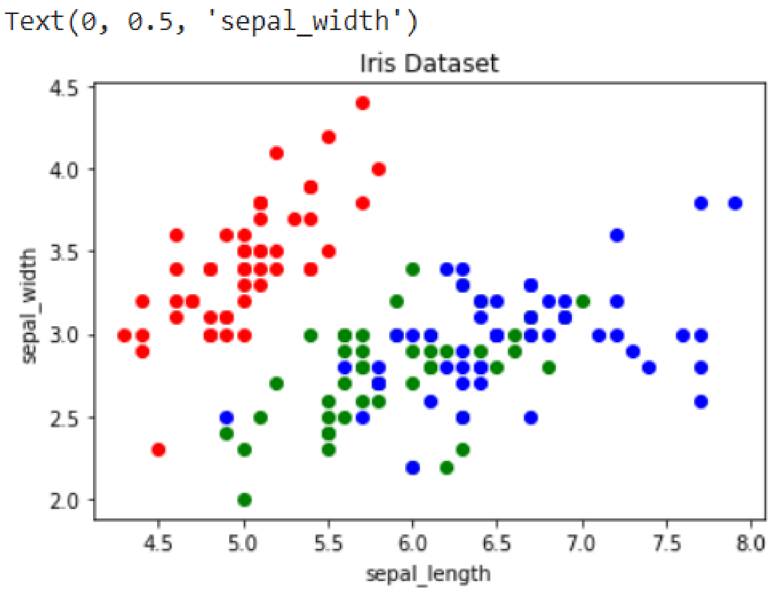


```
Text(0, 0.5, 'sepal_width')
```

Programming Code:

```
#matplotlib plot with diferent colors for Iris flower varities
fig, ax = plt.subplots()
colors = {'Setosa':'r', 'Versicolor':'g', 'Virginica':'b'}

for i in range(len(iris['sepal.length'])):
    ax.scatter(iris['sepal.length'][i], iris['sepal.width'][i],co
lor=colors[iris['variety'][i]])
ax.set_title('Iris Dataset')
ax.set_xlabel('sepal_length')
ax.set_ylabel('sepal_width')
```

OUTPUT:

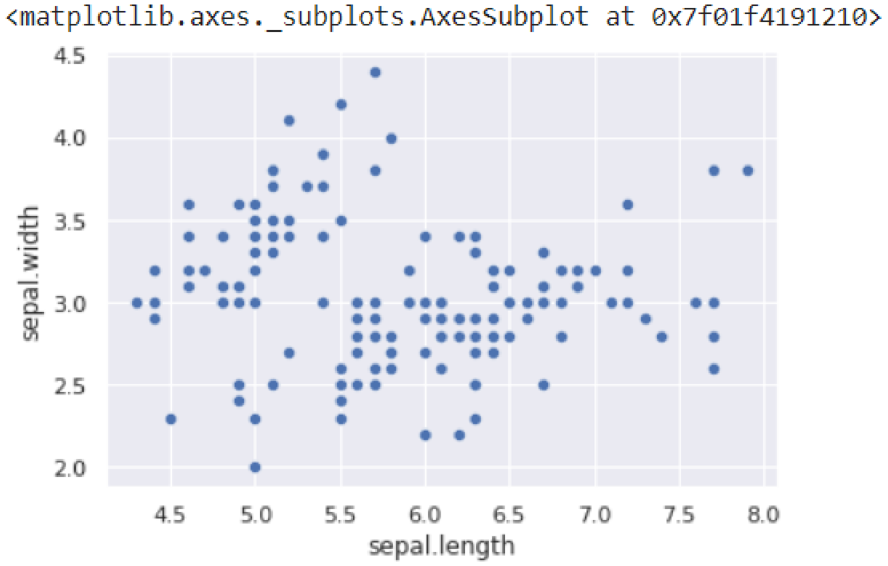


```
Text(0, 0.5, 'sepal_width')
```

Programming Code:

```
import seaborn as sns
sns.scatterplot(x='sepal.length', y='sepal.width', data=iris)
```

OUTPUT:

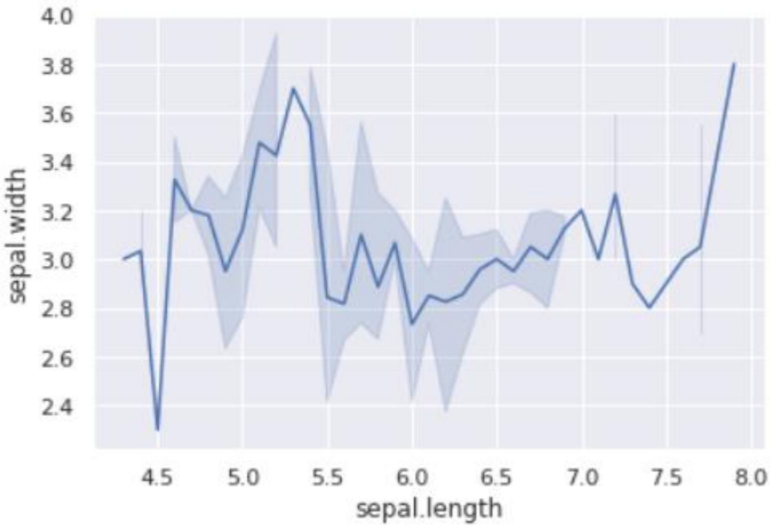


```
<matplotlib.axes._subplots.AxesSubplot at 0x7f01f4191210>
```

Programming Code:

```
sns.lineplot(x="sepal.length", y="sepal.width", data=iris)
plt.show()
```

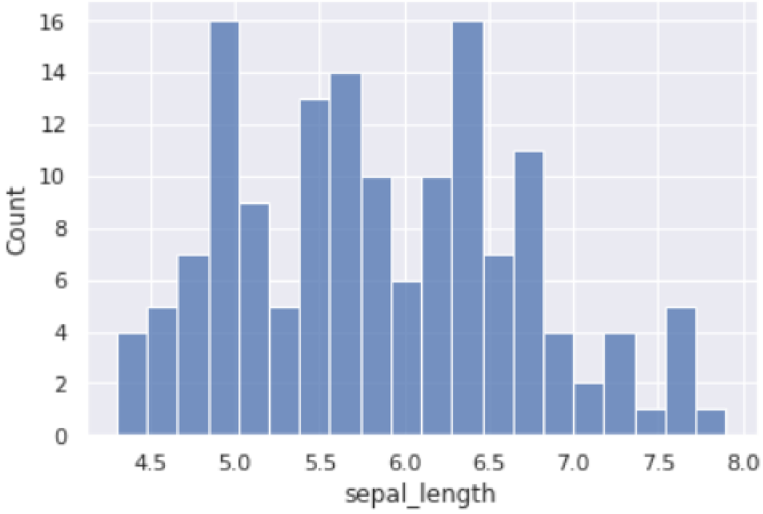
OUTPUT:



Programming Code:

```
#seaborn histogram plot
sns.set(style="darkgrid")
df = sns.load_dataset("iris")


sns.histplot(data=df, x="sepal_length",bins=20)
plt.show()
```

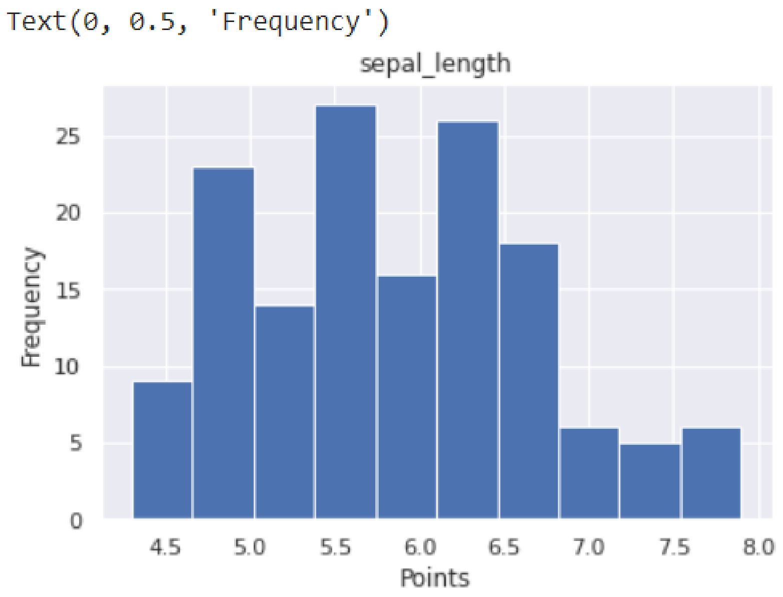
OUTPUT:



Programming Code:

```
#matplotlib histogram plot
iris_feat = iris.iloc[:,:-1]
iris_species = iris.iloc[:,-1]
fig, ax = plt.subplots()
# plot histogram
ax.hist(iris_feat['sepal.length'])
# set title and labels
ax.set_title('sepal_length')
ax.set_xlabel('Points')
ax.set_ylabel('Frequency')
```

OUTPUT:

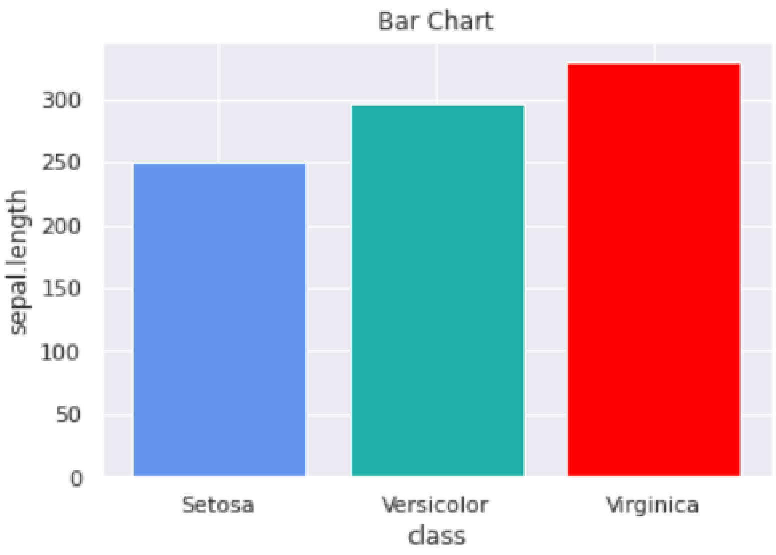


```
Text(0, 0.5, 'Frequency')
```

Programming Code:

```
#Bar chart using Matplotlib
df = iris.groupby('variety')['sepal.length'].sum().to_frame().reset_index()
#Creating the bar chart
plt.bar(df['variety'],df['sepal.length'],color = ['cornflowerblue','lightseagreen','red'])
#Adding the aesthetics
plt.title('Bar Chart')
plt.xlabel('class')
plt.ylabel('sepal.length')
#Show the plot
plt.show()
```

OUTPUT:

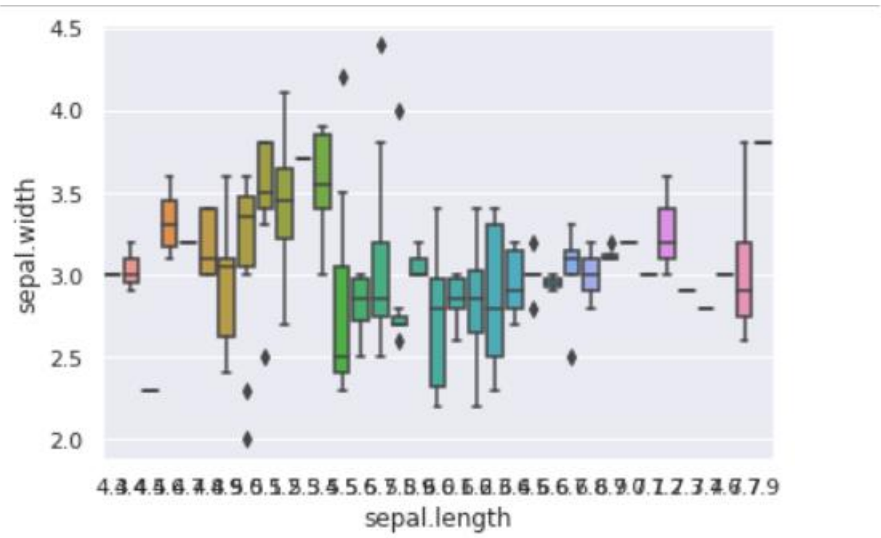


Programming Code:

```
import seaborn as sns
sns.boxplot('sepal.length', 'sepal.width', data=iris)
```

OUTPUT:

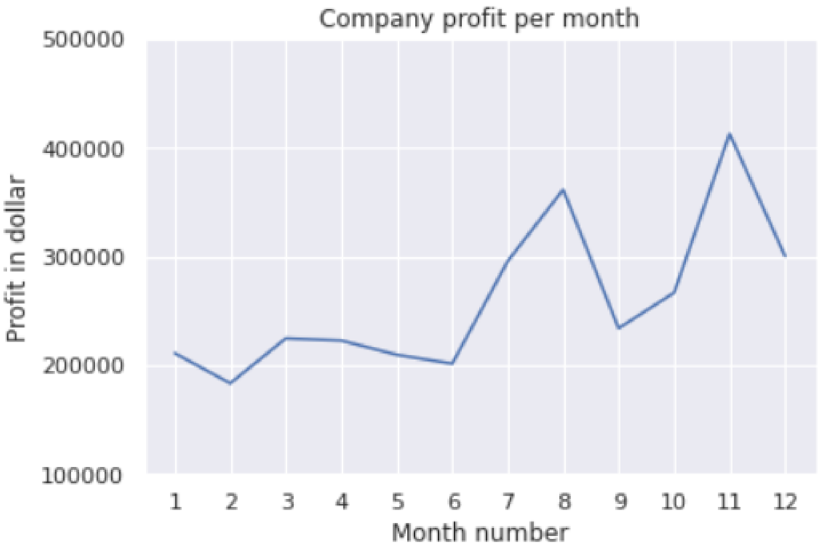
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will
result in an error or misinterpretation.
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f01ef6944d0>
```



Programming Code:

```
import pandas as pd
iris = pd.read_csv('/content/company_sales_data.csv')


#Line plot with matplotlib
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("company_sales_data.csv")
profitList = df ['total_profit'].tolist()
monthList  = df ['month_number'].tolist()
plt.plot(monthList, profitList, label = 'Month-
wise Profit data of last year')
plt.xlabel('Month number')
plt.ylabel('Profit in dollar')
plt.xticks(monthList)
plt.title('Company profit per month')
plt.yticks([100000, 200000, 300000, 400000, 500000])
plt.show()
```

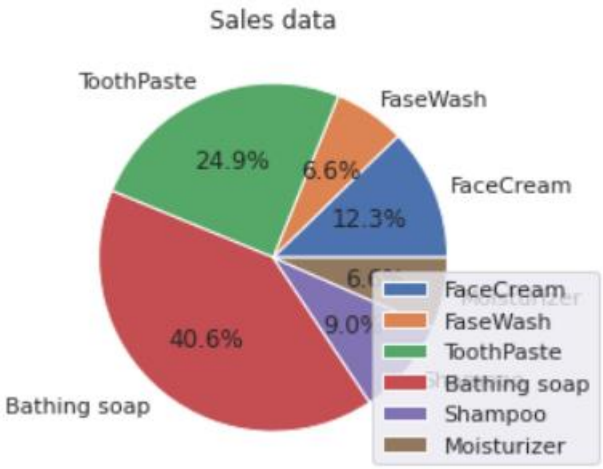
OUTPUT:



Programming Code:

```
monthList  = df ['month_number'].tolist()

labels = ['FaceCream', 'FaseWash', 'ToothPaste', 'Bathing soap',
'Shampoo', 'Moisturizer']
salesData   = [df ['facecream'].sum(), df ['facewash'].sum(), df
['toothpaste'].sum(),
         df ['bathingsoap'].sum(), df ['shampoo'].sum(), df ['moi
sturizer'].sum()]
plt.axis("equal")
plt.pie(salesData, labels=labels, autopct='%1.1f%%')
plt.legend(loc='lower right')
plt.title('Sales data')
plt.show()
```

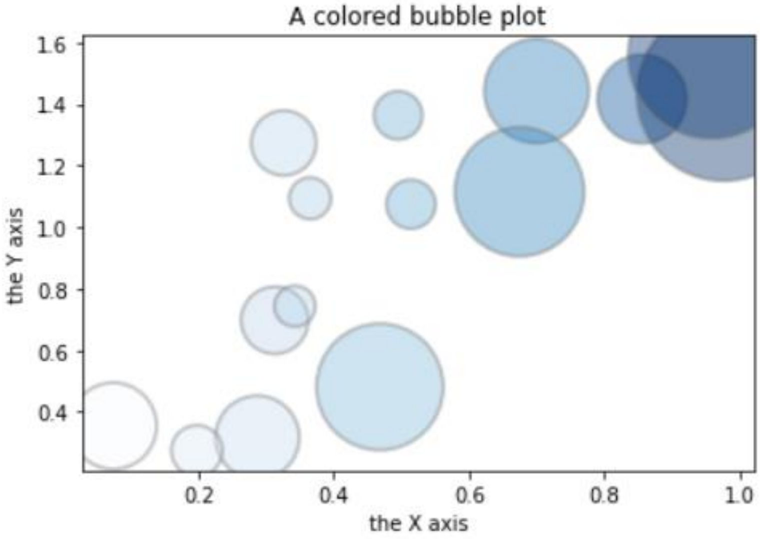
OUTPUT:



Programming Code:

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

# create data
x = np.random.rand(15)
y = x+np.random.rand(15)
z = x+np.random.rand(15)
z=z*z

# Change color with c and transparency with alpha.
# I map the color to the X axis value.
plt.scatter(x, y, s=z*2000, c=x, cmap="Blues", alpha=0.4, edgecol
ors="grey", linewidth=2)

# Add titles (main and on axis)
plt.xlabel("the X axis")
plt.ylabel("the Y axis")
plt.title("A colored bubble plot")

# Show the graph
plt.show()
```

OUTPUT:


A colored bubble plot
the Y axis
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.4
0.6
0.8
1.0
the X axis