Visualization

August 4, 2025

1 Review

1.1 Line Chart

[1]:
```
from matplotlib import pyplot as plt
```

[ ]:
```
x = [1, 2, 3]
y = [1, 4, 9]

plt.plot(x,y, color = 'b', linestyle = "-", linewidth = 2.5)
plt.show()
```

1.2 Scatter Plot and Bubble Chart

In many cases this is the least aggregated representation of your data. Displays relationship between two numerical variables.

[3]:
```
import pandas as pd
df = pd.read_csv('iris.csv')
df
```

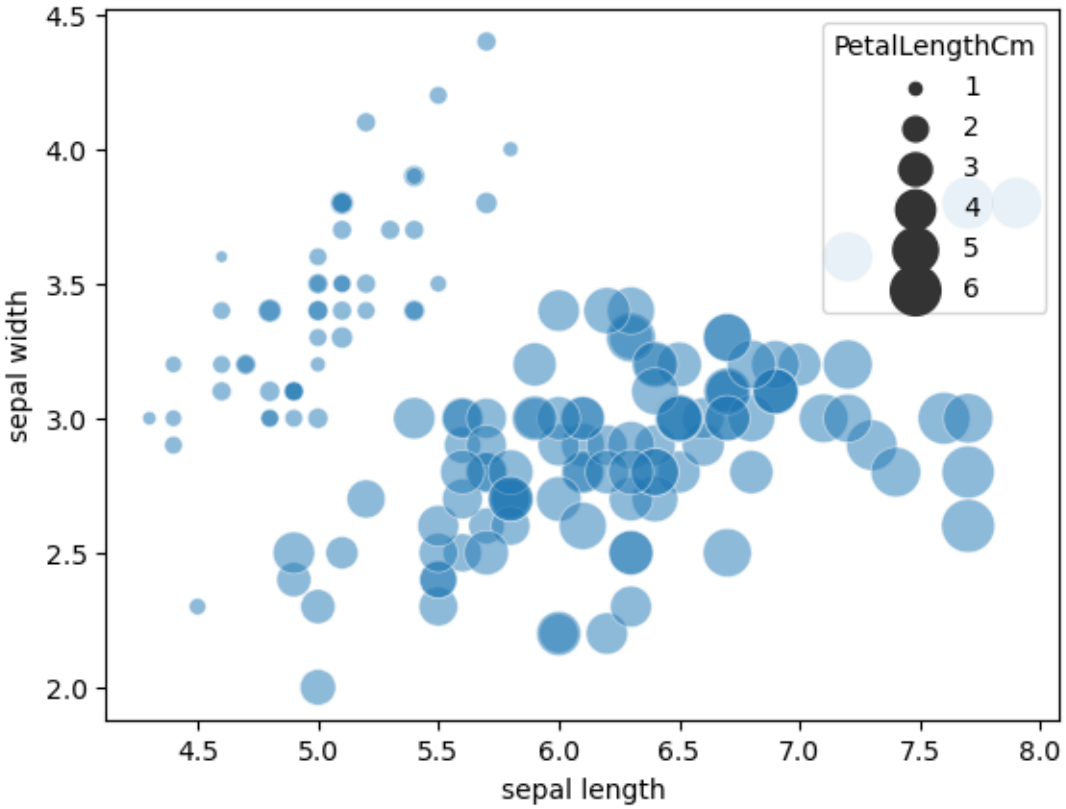
[3]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
..	…	…	…	…	…
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

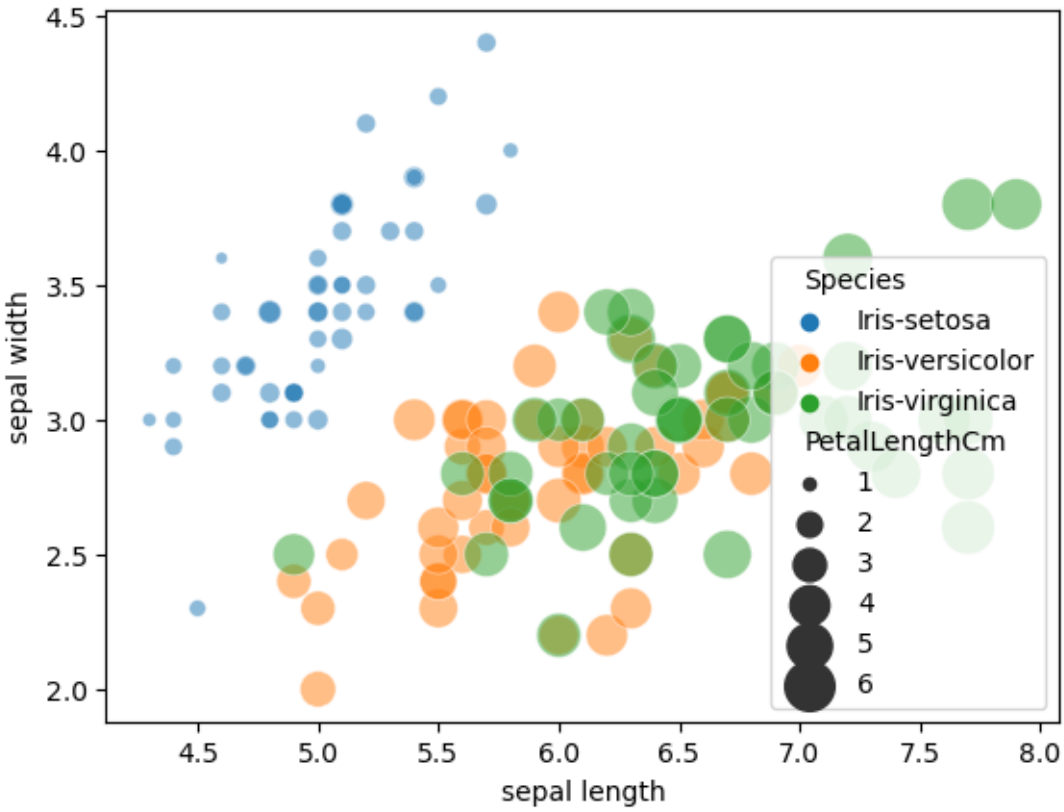
[150 rows x 5 columns]

[4]:
```

```

```
import seaborn as sns
sns.scatterplot(data=df, x="SepalLengthCm", y="SepalWidthCm",␣
 ↪size="PetalLengthCm",
                sizes=(20, 400), # Adjust the 'sizes' range
                alpha=0.5 ) # transparency level

plt.xlabel("sepal length")
plt.ylabel("sepal width")
plt.show()
```

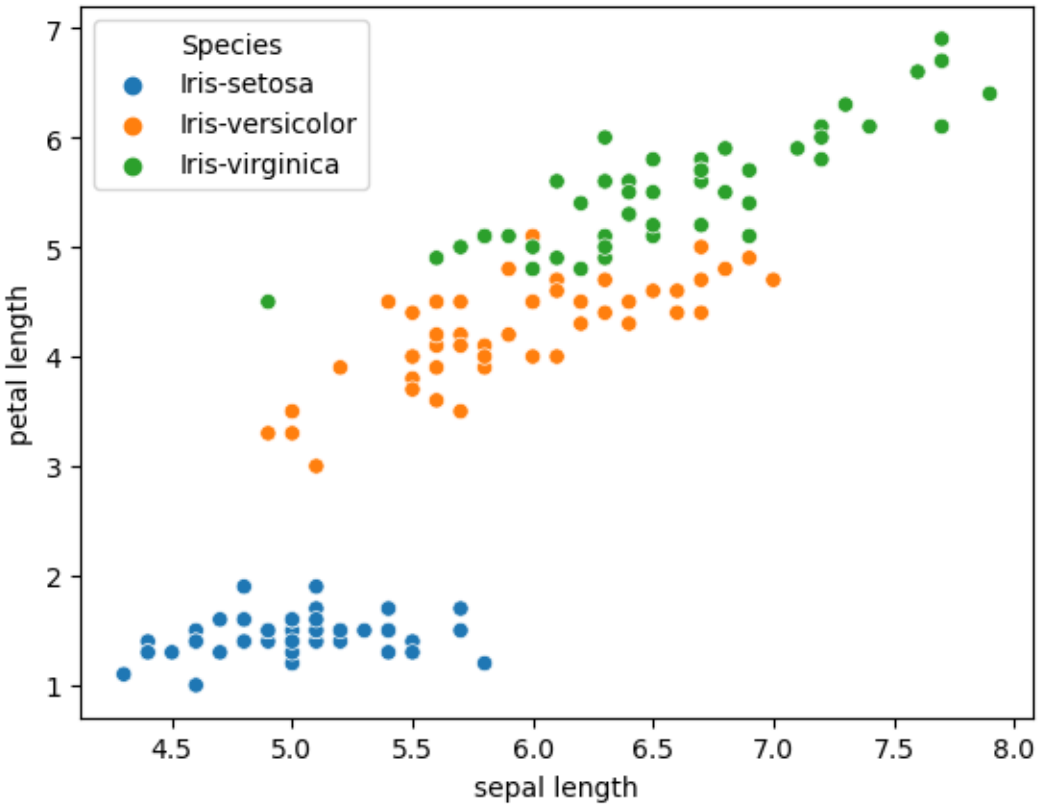


[6]:
```
import seaborn as sns
sns.scatterplot(data=df, x="SepalLengthCm", y="SepalWidthCm",␣
 ↪size="PetalLengthCm",
                sizes=(20, 400), # Adjust the 'sizes' range
                alpha=0.5,
                hue = "Species"
               ) # transparency level

plt.xlabel("sepal length")
plt.ylabel("sepal width")
plt.show()
```



1.2.1 Exploring different variables for better decision making

[7]:
```
sns.scatterplot(data=df, x="SepalLengthCm", y="PetalLengthCm", hue="Species")
plt.xlabel("sepal length")
plt.ylabel("petal length")
plt.show()
```



2 Distribution Plot

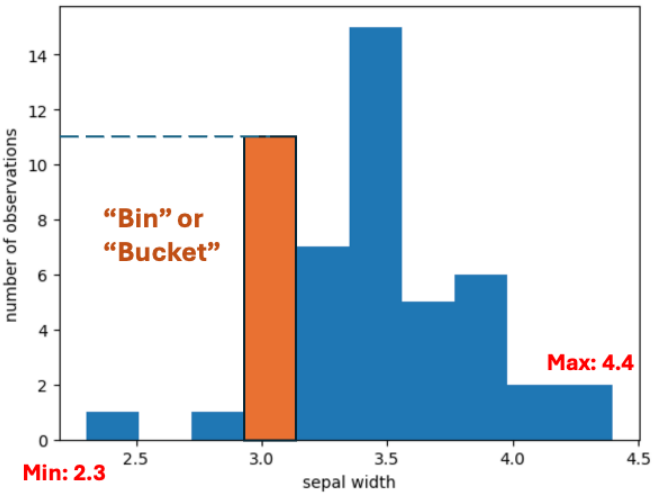
A distribution plot is a great way to visualize how a variable (like sepal length, price, or number_of_reviews) is spread across your dataset.

2.1 Distribution Plot: Histogram

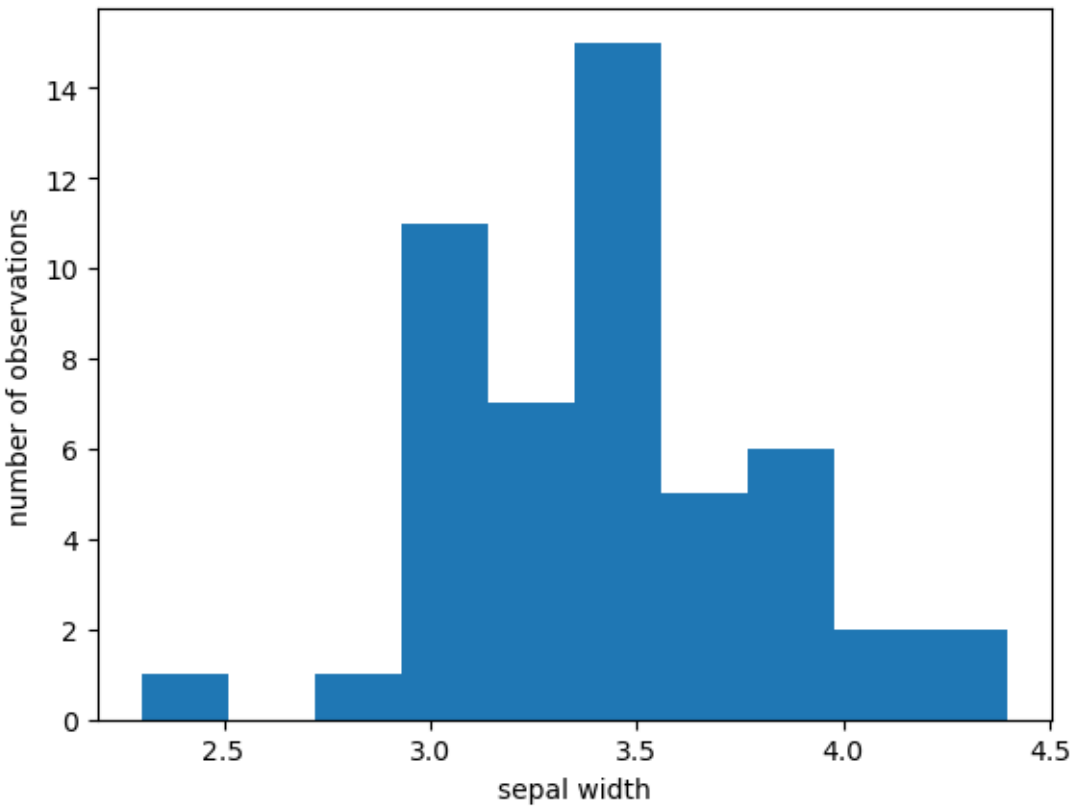
Histogram is an estimate of the probability distribution of a quantitative variable. The observed values are placed into different bins and the frequency of observations in each of those bins is calculated. A histogram is a graphical display of data using bars of different heights. Taller bars show that more data falls in that range.

It can detect possible outliers.

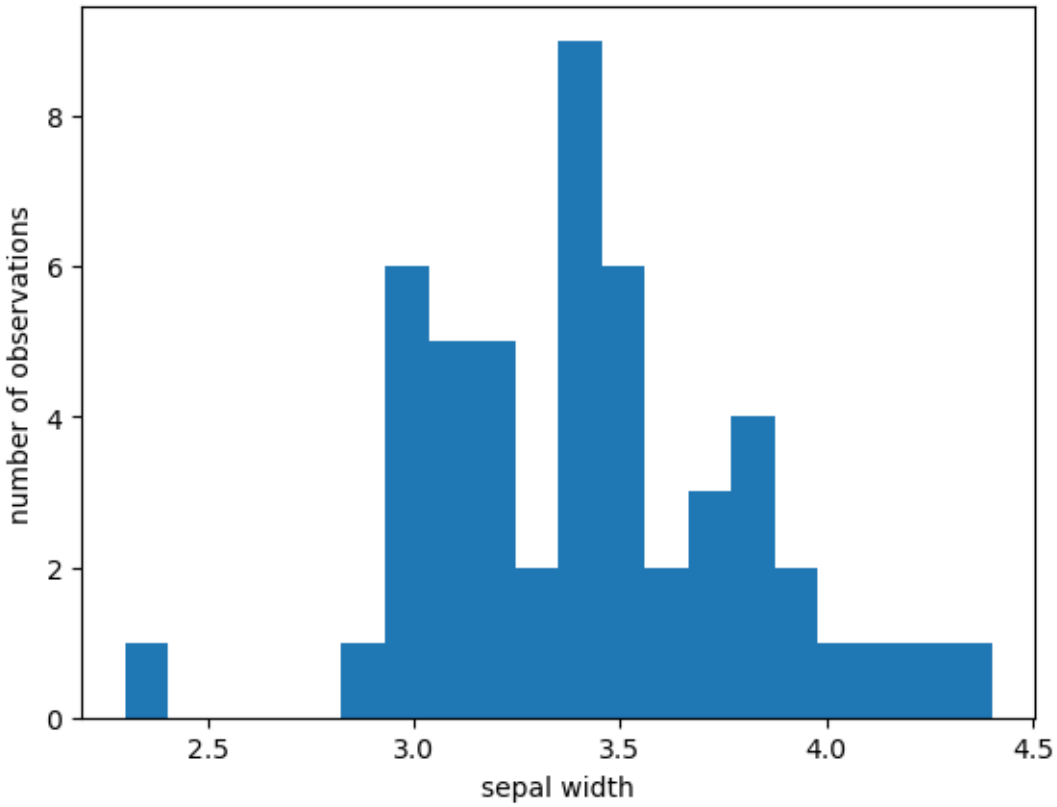
For this example, let's examine the distribution of the sepal width of all "Iris-Setosa"s.



[9]:
```
index = df["Species"] == "Iris-setosa"  # generate logical values on whether␣
↪Setosa
setosa = df.loc[index, :]
plt.hist(setosa["SepalWidthCm"])  # give the frequency of observations in each␣
↪bin
plt.xlabel("sepal width")
plt.ylabel("number of observations")
plt.show()
```



[10]:
```
num_bins = 20                       # default = 10
plt.hist(setosa["SepalWidthCm"], num_bins)  # give the frequency of␣
 ↪observations in each bin
plt.xlabel("sepal width")
plt.ylabel("number of observations")
#plt.grid(True)
plt.show()
```

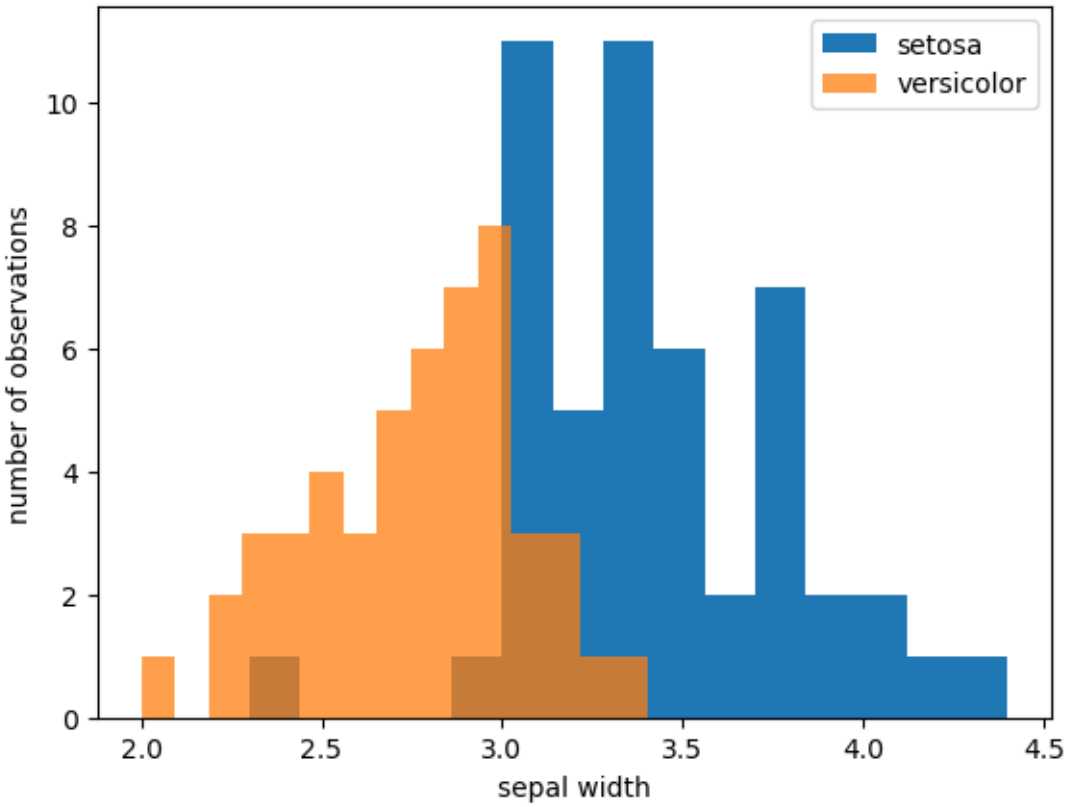


In statistics, an outlier is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses.

2.1.1 Overlaid Histogram

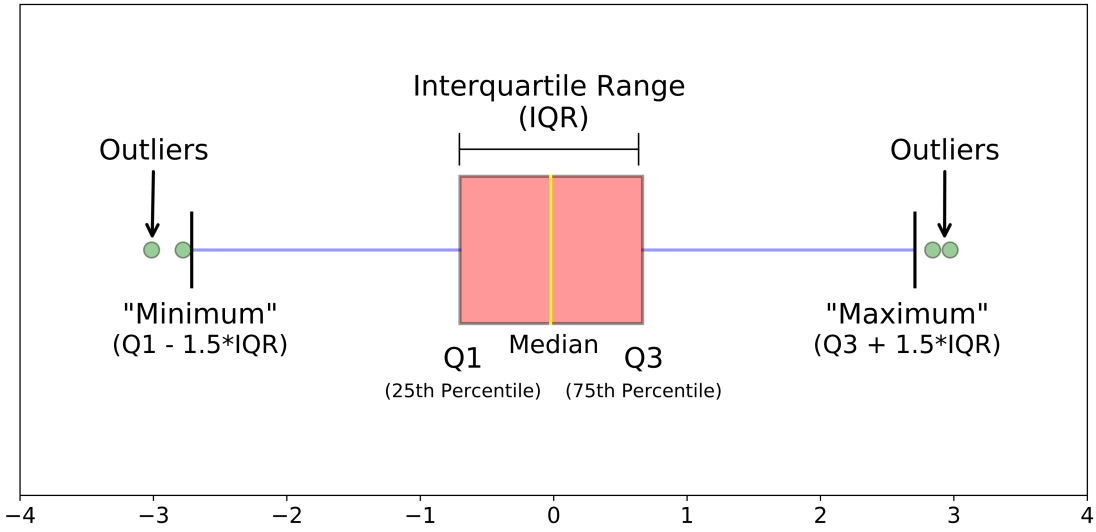
If you are looking to compare two (or more) distributions, use an overlaid histogram. Some additional care needs to be taken with these plots to ensure that they remain clear and easy to read, especially when more than two distributions are visualized. In this example, we will compare the distributions of sepal width of all "Iris-setosa"s and sepal width of all "Iris-versicolor"s.

[15]:
```
versicolor = df.loc[df["Species"] == "Iris-versicolor" , :]

num_bins = 15
plt.hist(setosa["SepalWidthCm"], num_bins, label = "setosa")
plt.hist(versicolor["SepalWidthCm"], num_bins, alpha = 0.75, label =␣
 ↪"versicolor")

plt.xlabel("sepal width")
plt.ylabel("number of observations")
plt.legend()
plt.show()
```



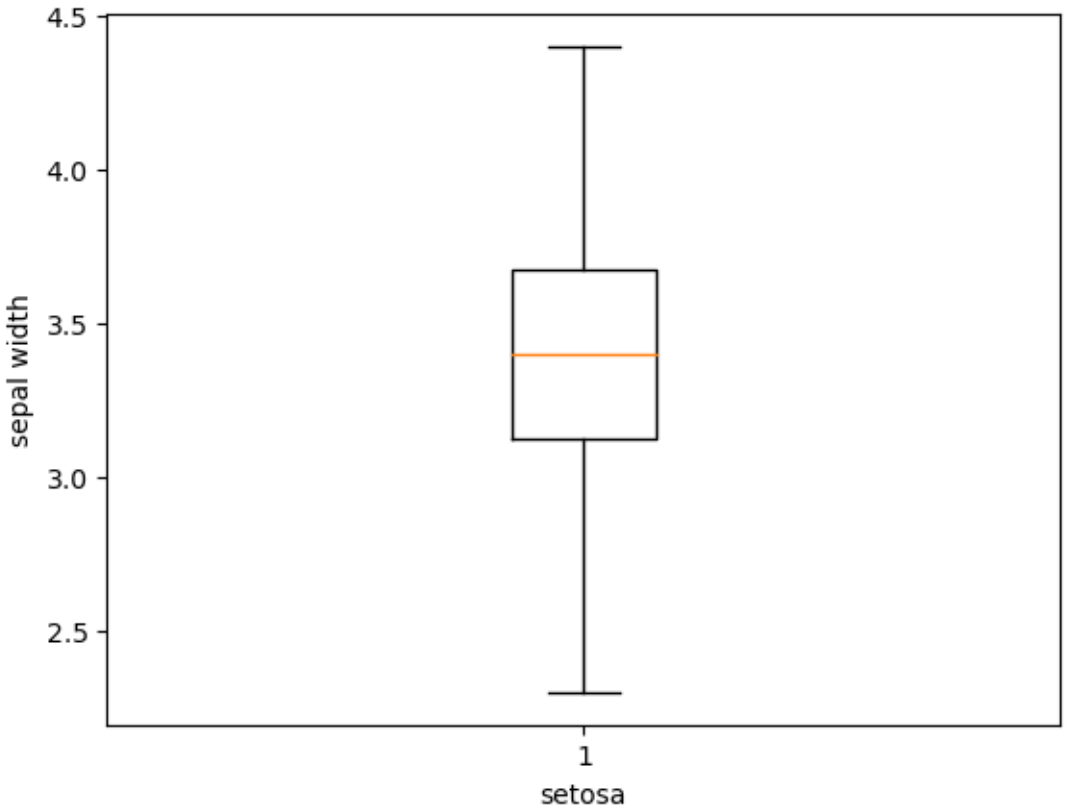
2.2 Distribution Plot: Box plot

A box plot used for graphically depicting groups of numerical data through their quartiles. Outliers can be plotted as individual points. The spacings between the different parts of the box indicate the degree of dispersion (spread) and skewness in the data, and show outliers. The basic explanations for boxplot are provided in the following figure:



Next, let's examine the distribution of the sepal width of all "Iris-setosa"s.

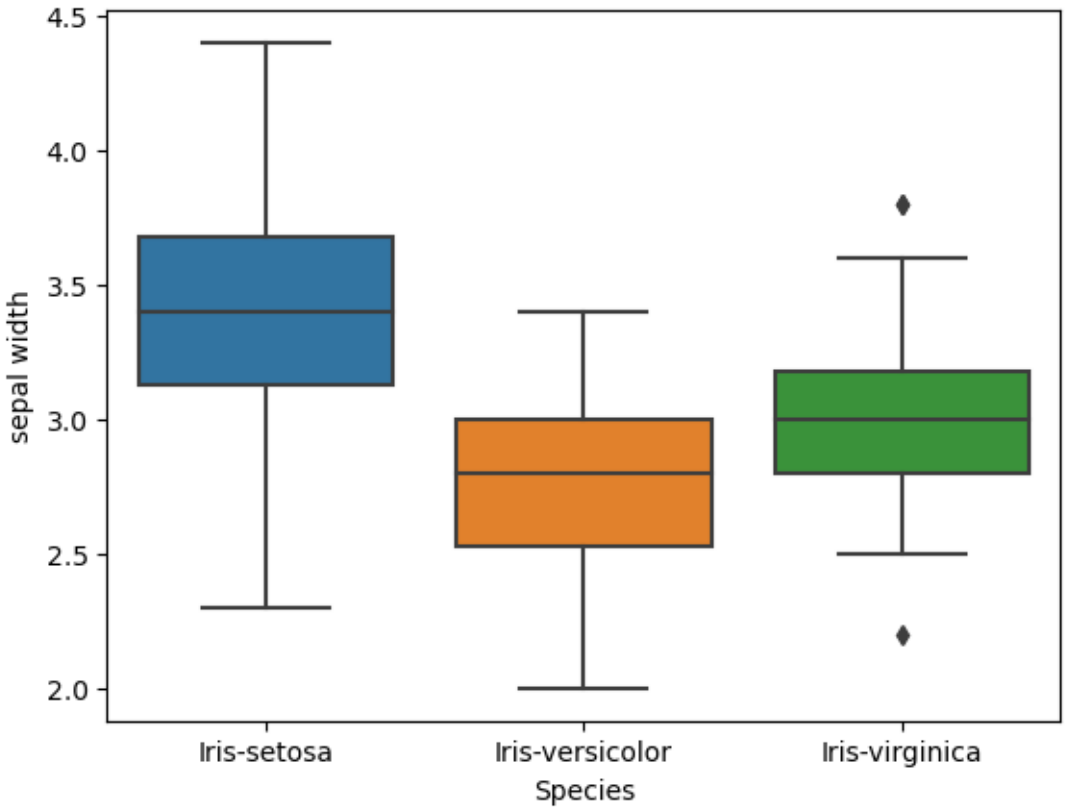
[19]:
```
plt.boxplot(setosa["SepalWidthCm"])
plt.ylabel("sepal width")
plt.xlabel("setosa")
plt.show()
```



2.2.1 Side-by-Side box plots

If you are looking to compare two (or more) distributions, you can use a side-by-side box plots. In this example, we will compare the distributions of sepal width of all "Iris-setosa"s, sepal width of all "Iris-versicolor"s, and that of all "Iris-virginica".

[20]:
```
sns.boxplot(data=df, x='Species', y='SepalWidthCm' )
plt.xlabel('Species')
plt.ylabel('sepal width')
plt.show()
```



2.3 Take-home Visualization Practice: Bike Sharing Systems

Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return back has become automatic. Through these systems, user is able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which is composed of over 500 thousands bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.

Apart from interesting real world applications of bike sharing systems, the characteristics of data being generated by these systems make them attractive for the research. Opposed to other transport services such as bus or subway, the duration of travel, departure and arrival position is explicitly recorded in these systems. This feature turns bike sharing system into a virtual sensor network that can be used for sensing mobility in the city. Hence, it is expected that most of important events in the city could be detected via monitoring these data.

Data Description We will be using the daily version of the Capital Bikeshare System dataset from the UCI Machine Learning Repository. This data set contains information about the daily count of bike rental checkouts in Washington, D.C.'s bikeshare program between 2011 and 2012. It also includes information about the weather and seasonal/temporal features for that day (like whether it was a weekday).

- **day**: Day of the record (relative to day 1:2011-01-01)
- **season**: Season (1:winter, 2:spring, 3:summer, 4:fall)
- **weekday**: Day of the week (0=Sunday, 6=Saturday)
- **workingday**: If day is neither weekend nor holiday is 1, otherwise is 0.
- **weathersit**:
  - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
  - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
- **temp**: Normalized temperature in Celcius
- **windspeed**: Normalized wind speed
- **casual**: Count of checkouts by casual/non-registered users
- **registered**: Count of checkouts by registered users
- **cnt**: Total checkouts

[ ]:
```
import pandas as pd
daily = pd.read_csv('day.csv')
daily.head()
```

Questions:

1. **Understand Trends.** Generate a line chart to show the checkouts over time by using `day` column as the x-axis and `cnt` column as the y-axis. Label the x-axis as 'Day', and y-axis as 'Check Outs'. What can you conclude?
2. **Explore Relationships.** We will plot the daily count of bikes that were checked out by casual users against the temperature. Color the points to be '#539cab'. Set the transparency to be 0.7. Be sure to include appropriate labels for x-axis and y-axis. What insight can you get?
3. **Explore Relationships with Multidimensional Information.** We will plot the daily count of bikes that were checked out by casual users against the temperature. The color of each point will be set according to whether it is a working day. Set the transparency to be 0.7. Be sure to include appropriate labels for x-axis and y-axis. Change the legend to whether it is a working day. What additional insights can you get?

4. **Examine Distributions.** Let's first build a histogram of the registered bike checkouts with the number of bins as 10. Set appropriate labels. Also set the title to be "Distribution of Registered Check Outs".

5. **Compare Distributions.** We now compare the distributions of registered and casual checkouts. To make the figure easy to understand, additional to the histogram we made for the previous question, we will set the transparency of the casual one to 0.8 and the number of bins to 15. Set appropriate labels.

6. How do the temperatures change across the seasons? You need to choose the type of visualization that best serves this purpose. What are the mean and median temperatures?

7. In this question, we are exploring AI-assistance tools in python programming and visualization. By providing proper prompts to generative AI tools, generate a line chart showing average bike rentals for each day of the week. This can reveal weekly usage patterns and peak days. Interact with generative AI tools, ask questions about anything you find unclear, and try to understand the generated codes.