Intro to pandas

Installing Pandas

In [1]:
```
# !pip install pandas
```

Import pandas

In [2]:
```
import pandas as pd
```

Read data from a csv file using pandas

df = pd.read_csv(r"\File Name.csv")
for example we have :

In [3]:
```
df = pd.read_csv(r'C:\Users\RSKALA\jupyter_notebooks\csv files\iris.csv')
```

Show the first 5 rows of dataframe

In [4]:
```
df.head()
```

Out[4]:

	c1	c2	c3	c4	label
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

You can adjust the number of rows you want to see

In [5]:
```
1 df.head(10)
```

Out[5]:

	c1	c2	c3	c4	label
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
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa

More info about our dataframe

A quick overview of the entire data-frame, such as the Datatypes of the columns, number of Null/Non-Null values in the columns and so on.

In [6]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   c1      150 non-null    float64
 1   c2      150 non-null    float64
 2   c3      150 non-null    float64
 3   c4      150 non-null    float64
 4   label   150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Filtering data in dataframe

for example we want to sort the data in our df based on 'c1' values and check the first 10 rows

In [7]:
```
df.sort_values('c1')[['c1','c2','c3','c4','label']].head(10)
```

Out[7]:

	c1	c2	c3	c4	label
13	4.3	3.0	1.1	0.1	Iris-setosa
42	4.4	3.2	1.3	0.2	Iris-setosa
38	4.4	3.0	1.3	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
41	4.5	2.3	1.3	0.3	Iris-setosa
22	4.6	3.6	1.0	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
47	4.6	3.2	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa

Check number of rows and columns of dataframe

as we can see in our df there are 150 rows and 5 columns

In [8]:
```
df.shape
```

Out[8]:
```
(150, 5)
```

df to numpy arrays

save all columns of df except 'label' as x and 'label' column as y

In [9]:
```
x = df.iloc[:,:-1].to_numpy()
y = df.iloc[:,-1].to_numpy()

```

Intro to matplotlib

installing matplotlib

In [10]:
```
# !pip install matplotlib
```

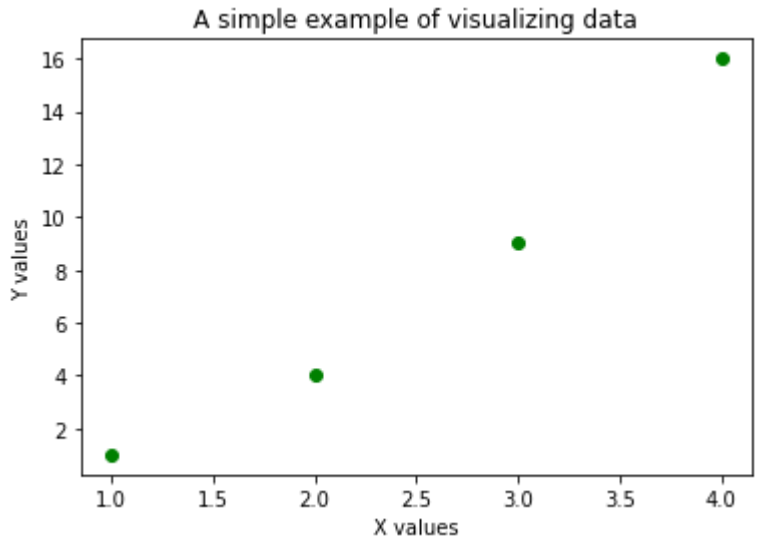
import matplotlib

In [11]:
```
import matplotlib.pyplot as plt
```

Assume we have some points with specific x and y. we want to show them as indivisual points.

In [12]:
```
x = [1, 2, 3, 4]
y = [1, 4, 9, 16]
plt.scatter(x,y,c='green')
plt.xlabel("X values")
plt.ylabel("Y values")
plt.title('A simple example of visualizing data')
```

Out[12]:
```
Text(0.5, 1.0, 'A simple example of visualizing data')
```



Now consider the previous x and y values, we want to plot a line that passes through these points. plot function in matplotlib package interpolate points between these points.

In [13]:
```
1 plt.plot(x,y,c='green')
2 plt.xlabel("X values")
3 plt.ylabel("Y values")
4 plt.title('A simple example of visualizing a line passes the data')
```

Out[13]: Text(0.5, 1.0, 'A simple example of visualizing a line passes the data')