Visualizing Data in Python

VANESSA M. CUEVAS ARROYO A01634064

The notebook aims to undertand the content of the cartwheel data set.

Acknowledgments

- Data from https://www.coursera.org/ from the course "Understanding and Visualizing Data with Python" by University of Michigan

Data management is a crucial component to statistical analysis and data science work.

This notebook will show you how to import, view, undertand, and manage your data using the Pandas data processing library, i.e., the notebook will demonstrates how to read a dataset into Python, and obtain a basic understanding of its content.

Note that **Python** by itself is a general-purpose programming language and does not provide high-level data processing capabilities. The **Pandas** library was developed to meet this need. **Pandas** is the most popular Python library for data manipulation, and we will use it extensively in this course. **Pandas** provides high-performance, easy-to-use data structures and data analysis tools.

The main data structure that **Pandas** works with is called a **Data Frame**. This is a two-dimensional table of data in which the rows typically represent cases and the columns represent variables (e.g. data used in this tutorial). Pandas also has a one-dimensional data structure called a **Series** that we will encounter when accesing a single column of a Data Frame.

Pandas has a variety of functions named `read_xxx` for reading data in different formats. Right now we will focus on reading `csv` files, which stands for comma-separated values. However the other file formats include `excel`, `json`, and `sql`.

There are many other options to `read_csv` that are very useful. For example, you would use the option `sep='\t'` instead of the default `sep=','` if the fields of your data file are delimited by tabs instead of commas. See here for the full documentation for `read_csv`.

Acknowledgments

- The dataset used in this tutorial is from https://www.coursera.org/ from the course "Understanding and Visualizing Data with Python" by University of Michigan

When working with a new dataset, one of the most useful things to do is to begin to visualize the data. By using **tables**, **histograms**, **boxplots**, **scatter plots** and other visual tools, we can get a better idea of what the data may be trying to tell us, and we can gain insights into the data that we may have not discovered otherwise.

In this notebook will use the Seaborn data processing library, which is a higher-level interface to **Matplotlib** that can be used to simplify many visualization tasks

The **Seaborn** provides visualisations tools that will allow to explore data from a graphical perspective.

Acknowledgments

- Data from https://www.coursera.org/ from the course "Understanding and Visualizing Data with Python" by University of Michigan

Importing libraries

```
# Import the packages that we will be using
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Importing data

```
# Define where you are running the code: colab or local
RunInColab          = False     # (False: no  | True: yes)

# If running in colab:
if RunInColab:
    # Mount your google drive in google colab
    from google.colab import drive
    drive.mount('/content/drive')

    # Find location
    #!pwd
    #!ls
    #!ls "/content/drive/My Drive/Colab 
Notebooks/MachineLearningWithPython/"

    # Define path del proyecto
    Ruta            = "/content/drive/My Drive/Colab 
Notebooks/MachineLearningWithPython/"

else:
    # Define path del proyecto
    Ruta            = "datasets/cartwheel/cartwheel.csv"

# url string that hosts our .csv file


# Read the .csv file and store it as a pandas Data Frame
df  = pd.read_csv(Ruta)
```

Exploring the content of the data set

Get a general 'feel' of the data

```
df.head (5)

   ID   Age Gender  GenderGroup Glasses  GlassesGroup  Height  
Wingspan  \
0   1  56.0      F            1       Y             1    62.0   
61.0   
1   2  26.0      F            1       Y             1    62.0   
60.0   
2   3  33.0      F            1       Y             1    66.0   
64.0   
3   4  39.0      F            1       N             0    64.0   
63.0   
4   5  27.0      M            2       N             0    73.0   
75.0   

   CWDistance Complete  CompleteGroup  Score  
0          79        Y            1.0      7  
1          70        Y            1.0      8  
2          85        Y            1.0      7  
3          87        Y            1.0     10  
4          72        N            0.0      4  
```

Frequency tables

The `value_counts()` method can be used to determine the number of times that each distinct value of a variable occurs in a data set. In statistical terms, this is the "frequency distribution" of the variable. The `value_counts()` method produces a table with two columns. The first column contains all distinct observed values for the variable. The second column contains the number of times each of these values occurs. Note that the table returned by `value_counts()` is actually a **Pandas** data frame, so can be further processed using any Pandas methods for working with data frames.

```
# Number of times that each distinct value of a variable occurs in a 
data set
df.value_counts()
```

```
ID   Age   Gender  GenderGroup  Glasses  GlassesGroup  Height  Wingspan  
CWDistance  Complete  CompleteGroup  Score
1    56.0  F       1            Y        1             62.00   61.0      
79          Y         1.0            7        1
26   28.0  M       2            N        0             75.00   76.0      
```

```
111         Y         1.0            10       1
28  25.0  M       2            Y        1             76.00   73.0
107         Y         1.0            8        1
29  32.0  F       1            Y        1             63.00   60.0
75          Y         1.0            8        1
30  38.0  F       1            Y        1             61.50   61.0
78          Y         1.0            7        1
31  27.0  F       1            Y        1             62.00   60.0
72          Y         1.0            8        1
32  33.0  F       1            Y        1             65.30   64.0
91          Y         1.0            7        1
33  38.0  F       1            N        0             64.00   63.0
86          Y         1.0            10       1
34  27.0  M       2            N        0             77.00   75.0
100         Y         1.0            8        1
35  24.0  F       1            N        0             67.80   62.0
98          Y         1.0            9        1
36  27.0  M       2            N        0             68.00   66.0
74          Y         1.0            5        1
37  25.0  F       1            Y        1             65.00   64.5
92          Y         1.0            6        1
38  26.0  F       1            N        0             61.50   59.5
90          Y         1.0            9        1
39  31.0  M       2            Y        1             73.00   74.0
72          Y         1.0            9        1
40  30.0  M       2            Y        1             69.50   66.0
96          Y         1.0            6        1
41  23.0  F       1            N        0             70.40   71.0
66          Y         1.0            4        1
42  26.0  M       2            Y        1             73.50   72.0
115         Y         1.0            6        1
43  28.0  F       1            Y        1             72.50   72.0
81          Y         1.0            10       1
44  26.0  F       1            Y        1             72.00   72.0
92          Y         1.0            8        1
45  30.0  F       1            Y        1             66.00   64.0
85          Y         1.0            7        1
46  39.0  F       1            N        0             64.00   63.0
87          Y         1.0            10       1
47  27.0  M       2            N        0             78.00   75.0
72          N         0.0            7        1
48  24.0  M       2            N        0             79.50   75.0
82          N         0.0            8        1
27  24.0  M       2            N        0             78.40   71.0
92          Y         1.0            7        1
25  23.0  F       1            Y        1             65.00   63.0
67          N         0.0            3        1
2   26.0  F       1            Y        1             62.00   60.0
70          Y         1.0            8        1
24  26.0  M       2            N        0             69.00   71.0
```

```
63          Y        1.0           5        1
3   33.0  F       1            Y        1               66.00   64.0
85          Y        1.0           7        1
4   39.0  F       1            N        0               64.00   63.0
87          Y        1.0           10       1
5   27.0  M       2            N        0               73.00   75.0
72          N        0.0           4        1
6   24.0  M       2            N        0               75.00   71.0
81          N        0.0           3        1
7   28.0  M       2            N        0               75.00   76.0
107         Y        1.0           10       1
8   22.0  F       1            N        0               65.00   62.0
98          Y        1.0           9        1
9   29.0  M       2            Y        1               74.00   73.0
106         N        0.0           5        1
10  33.0  F       1            Y        1               63.00   60.0
65          Y        1.0           8        1
11  30.0  M       2            Y        1               69.50   66.0
96          Y        1.0           6        1
12  28.0  F       1            Y        1               62.75   58.0
79          Y        1.0           10       1
13  25.0  F       1            Y        1               65.00   64.5
92          Y        1.0           6        1
14  23.0  F       1            N        0               61.50   57.5
66          Y        1.0           4        1
15  31.0  M       2            Y        1               73.00   74.0
72          Y        1.0           9        1
16  26.0  M       2            Y        1               71.00   72.0
115         Y        1.0           6        1
17  26.0  F       1            N        0               61.50   59.5
90          N        0.0           10       1
18  27.0  M       2            N        0               66.00   66.0
74          Y        1.0           5        1
19  23.0  M       2            Y        1               70.00   69.0
64          Y        1.0           3        1
20  24.0  F       1            Y        1               68.00   66.0
85          Y        1.0           8        1
21  23.0  M       2            Y        1               69.00   67.0
66          N        0.0           2        1
22  29.0  M       2            N        0               71.00   70.0
101         Y        1.0           8        1
23  25.0  M       2            N        0               70.00   68.0
82          Y        1.0           4        1
49  28.0  M       2            N        0               77.80   76.0
99          Y        1.0           9        1
dtype: int64
```

```
# Proportion of each distinct value of a variable occurs in a data set
df.value_counts(normalize=True)
```

```
df
```

```
    ID   Age Gender  GenderGroup Glasses  GlassesGroup  Height  
Wingspan  \
0    1  56.0      F            1       Y             1   62.00  
61.0   
1    2  26.0      F            1       Y             1   62.00  
60.0   
2    3  33.0      F            1       Y             1   66.00  
64.0   
3    4  39.0      F            1       N             0   64.00  
63.0   
4    5  27.0      M            2       N             0   73.00  
75.0   
5    6  24.0      M            2       N             0   75.00  
71.0   
6    7  28.0      M            2       N             0   75.00  
76.0   
7    8  22.0      F            1       N             0   65.00  
62.0   
8    9  29.0      M            2       Y             1   74.00  
73.0   
9   10  33.0      F            1       Y             1   63.00  
60.0   
10  11  30.0      M            2       Y             1   69.50  
66.0   
11  12  28.0      F            1       Y             1   62.75  
58.0   
12  13  25.0      F            1       Y             1   65.00  
64.5   
13  14  23.0      F            1       N             0   61.50  
57.5   
14  15  31.0      M            2       Y             1   73.00  
74.0   
15  16  26.0      M            2       Y             1   71.00  
72.0   
16  17  26.0      F            1       N             0   61.50  
59.5   
17  18  27.0      M            2       N             0   66.00  
66.0   
18  19  23.0      M            2       Y             1   70.00  
69.0   
19  20  24.0      F            1       Y             1   68.00  
66.0   
20  21  23.0      M            2       Y             1   69.00  
67.0   
21  22  29.0      M            2       N             0   71.00  
70.0   
22  23  25.0      M            2       N             0   70.00  
```

```
68.0
23  24  26.0      M            2       N             0   69.00
71.0
24  25  23.0      F            1       Y             1   65.00
63.0
25  26  28.0      M            2       N             0   75.00
76.0
26  27  24.0      M            2       N             0   78.40
71.0
27  28  25.0      M            2       Y             1   76.00
73.0
28  29  32.0      F            1       Y             1   63.00
60.0
29  30  38.0      F            1       Y             1   61.50
61.0
30  31  27.0      F            1       Y             1   62.00
60.0
31  32  33.0      F            1       Y             1   65.30
64.0
32  33  38.0      F            1       N             0   64.00
63.0
33  34  27.0      M            2       N             0   77.00
75.0
34  35  24.0      F            1       N             0   67.80
62.0
35  36  27.0      M            2       N             0   68.00
66.0
36  37  25.0      F            1       Y             1   65.00
64.5
37  38  26.0      F            1       N             0   61.50
59.5
38  39  31.0      M            2       Y             1   73.00
74.0
39  40  30.0      M            2       Y             1   69.50
66.0
40  41  23.0      F            1       N             0   70.40
71.0
41  42  26.0      M            2       Y             1   73.50
72.0
42  43  28.0      F            1       Y             1   72.50
72.0
43  44  26.0      F            1       Y             1   72.00
72.0
44  45  30.0      F            1       Y             1   66.00
64.0
45  46  39.0      F            1       N             0   64.00
63.0
46  47  27.0      M            2       N             0   78.00
75.0
47  48  24.0      M            2       N             0   79.50
```

```
75.0
48  49  28.0      M            2       N             0   77.80
76.0
49  50  30.0      F            1       N             0   74.60
NaN
50  51   NaN      M            2       N             0   71.00
70.0
51  52  27.0      M            2       N             0     NaN
71.5

    CWDistance Complete  CompleteGroup  Score
0           79        Y            1.0      7
1           70        Y            1.0      8
2           85        Y            1.0      7
3           87        Y            1.0     10
4           72        N            0.0      4
5           81        N            0.0      3
6          107        Y            1.0     10
7           98        Y            1.0      9
8          106        N            0.0      5
9           65        Y            1.0      8
10          96        Y            1.0      6
11          79        Y            1.0     10
12          92        Y            1.0      6
13          66        Y            1.0      4
14          72        Y            1.0      9
15         115        Y            1.0      6
16          90        N            0.0     10
17          74        Y            1.0      5
18          64        Y            1.0      3
19          85        Y            1.0      8
20          66        N            0.0      2
21         101        Y            1.0      8
22          82        Y            1.0      4
23          63        Y            1.0      5
24          67        N            0.0      3
25         111        Y            1.0     10
26          92        Y            1.0      7
27         107        Y            1.0      8
28          75        Y            1.0      8
29          78        Y            1.0      7
30          72        Y            1.0      8
31          91        Y            1.0      7
32          86        Y            1.0     10
33         100        Y            1.0      8
34          98        Y            1.0      9
35          74        Y            1.0      5
36          92        Y            1.0      6
37          90        Y            1.0      9
38          72        Y            1.0      9
```

```
39          96        Y             1.0      6
40          66        Y             1.0      4
41         115        Y             1.0      6
42          81        Y             1.0     10
43          92        Y             1.0      8
44          85        Y             1.0      7
45          87        Y             1.0     10
46          72        N             0.0      7
47          82        N             0.0      8
48          99        Y             1.0      9
49          71        Y             1.0      9
50         101        Y             NaN      8
51         103        Y             1.0     10
```

Note that the `value_counts()` method excludes missing values. We confirm this below by adding up observations to your data frame with some missing values and then computing `value_counts()` and comparing this to the total number of rows in the data set, which is 28. This tells us that there are 28 - (21+6) = 1 missing values for this variable (other variables may have different numbers of missing values).

```
# Total number of observations
```

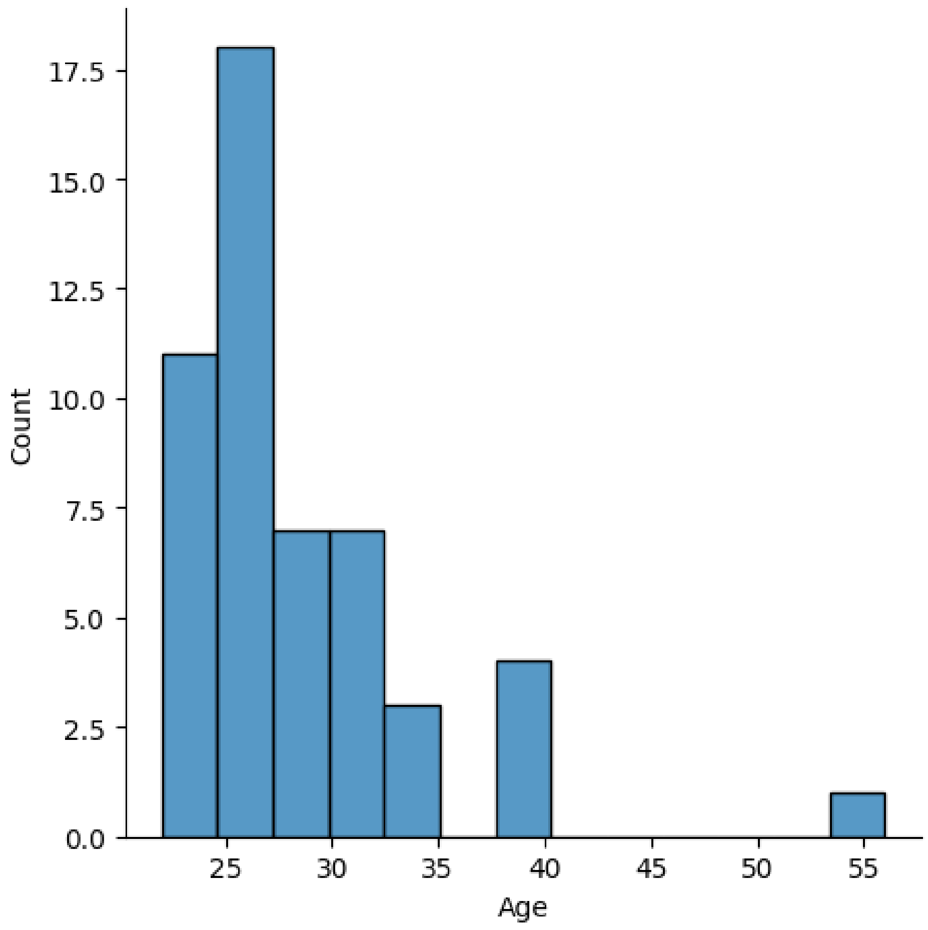
```
# Total number of null observations
```

```
# Total number of counts (excluding missing values)
```

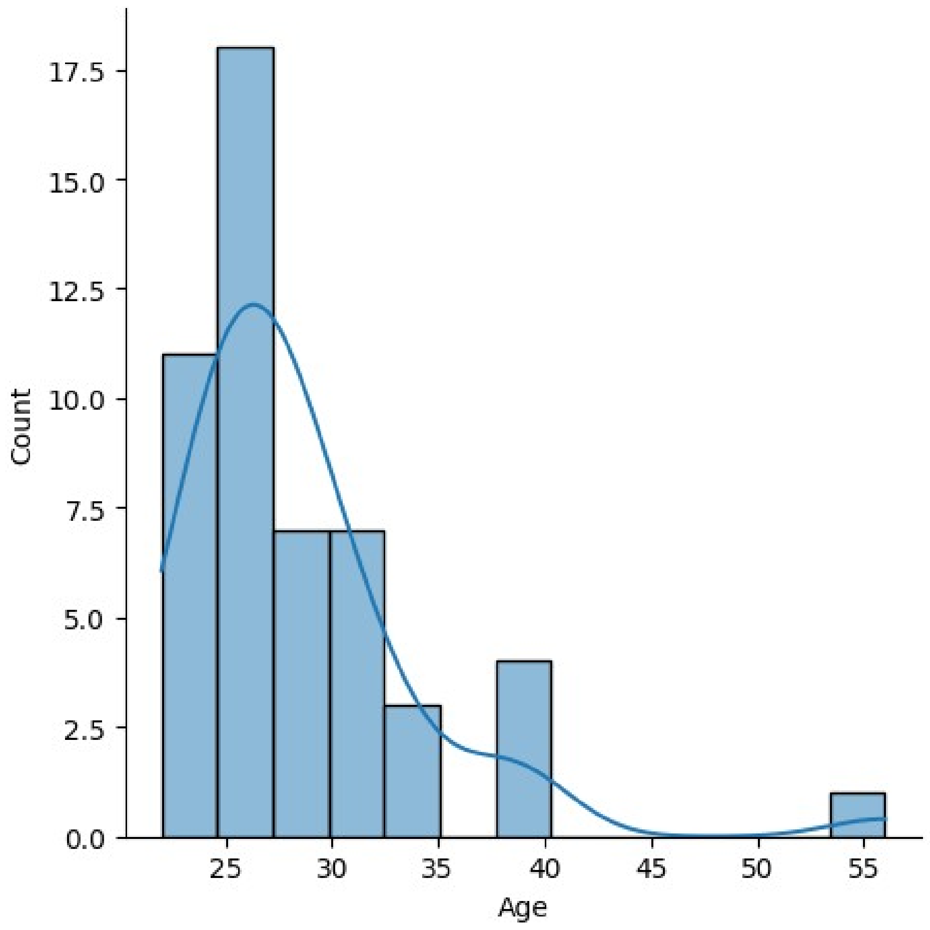
Histogram

It is often good to get a feel for the shape of the distribution of the data.

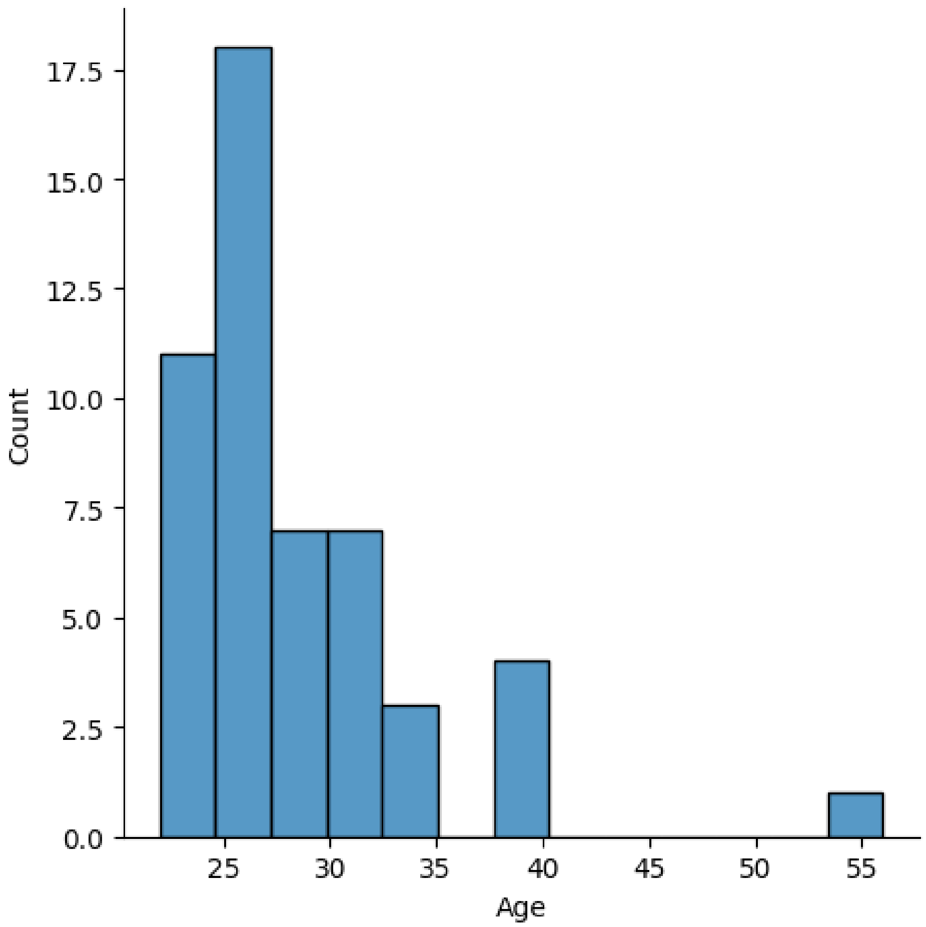
```
# Plot histogram of the total bill only

sns.displot(df["Age"],kde = False)
plt.show()
```



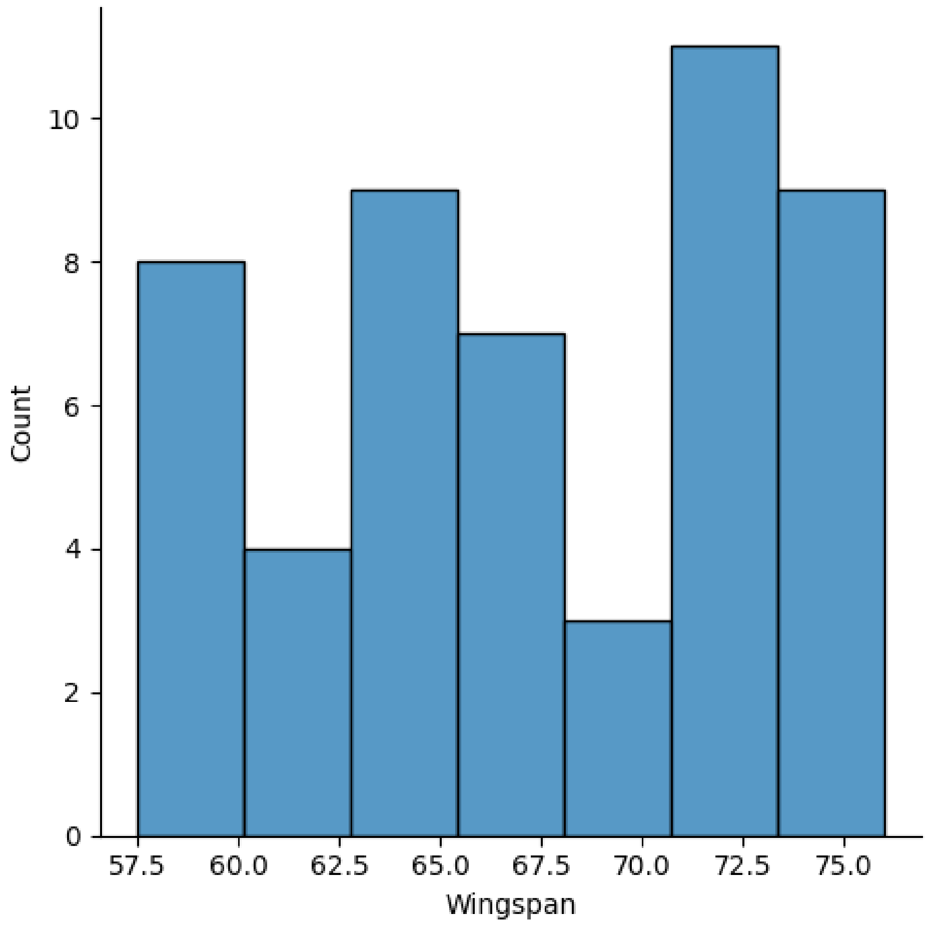
```
# Plot distribution of the tips only
sns.displot(df["Age"],kde = True)
plt.show()
```



```
# Plot histogram of both the Age and the Wingspan
sns.displot(df["Age"], kde = False)
sns.displot(df["Wingspan"], kde = False)

plt.show()
```



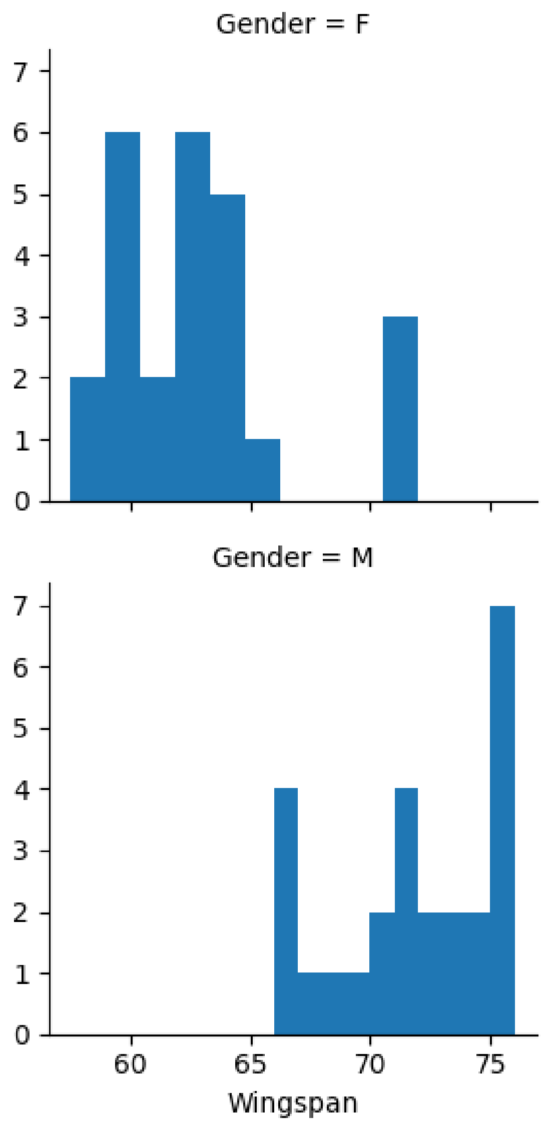
Count
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
25
30
35
40
45
50
55
Age



Histograms plotted by groups

While looking at a single variable is interesting, it is often useful to see how a variable changes in response to another. Thus, we can create a histograms of one quantitative variable grouped by another categorical variables.

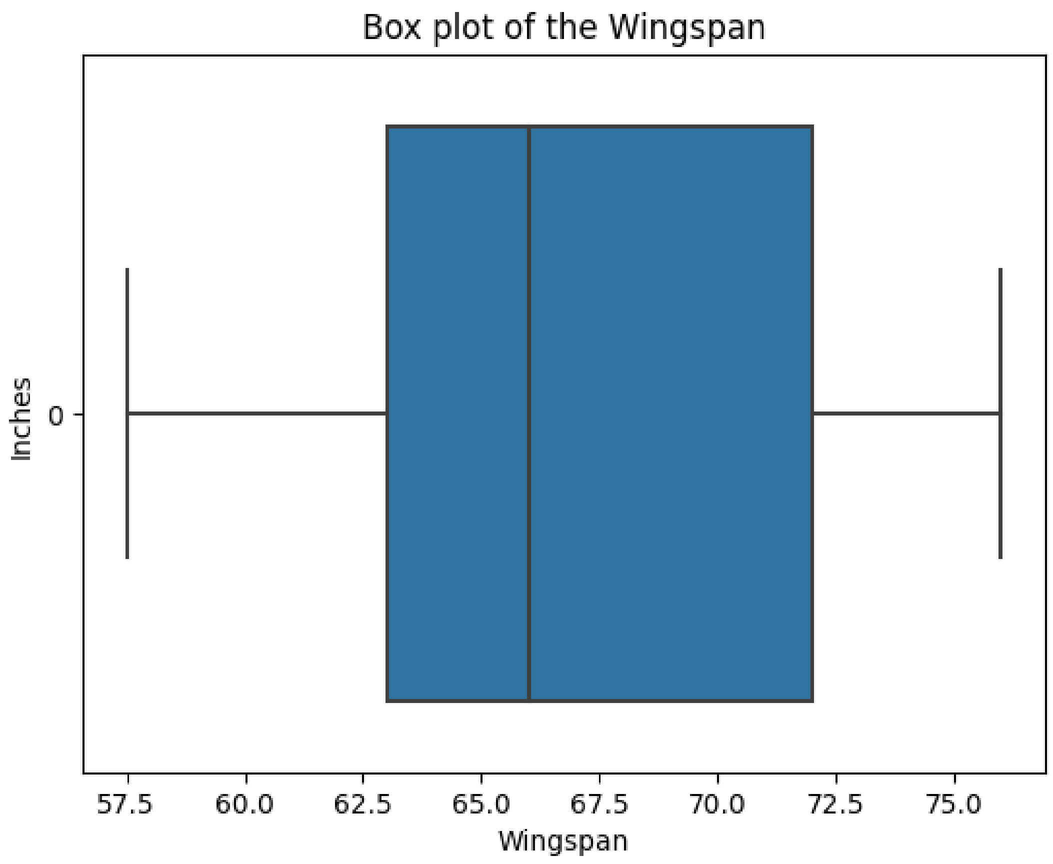
```
# Create histograms of the "Wingspan" grouped by "Gender"


g = sns.FacetGrid(df, row = "Gender")
g = g.map(plt.hist, "Wingspan")

plt.show()
```



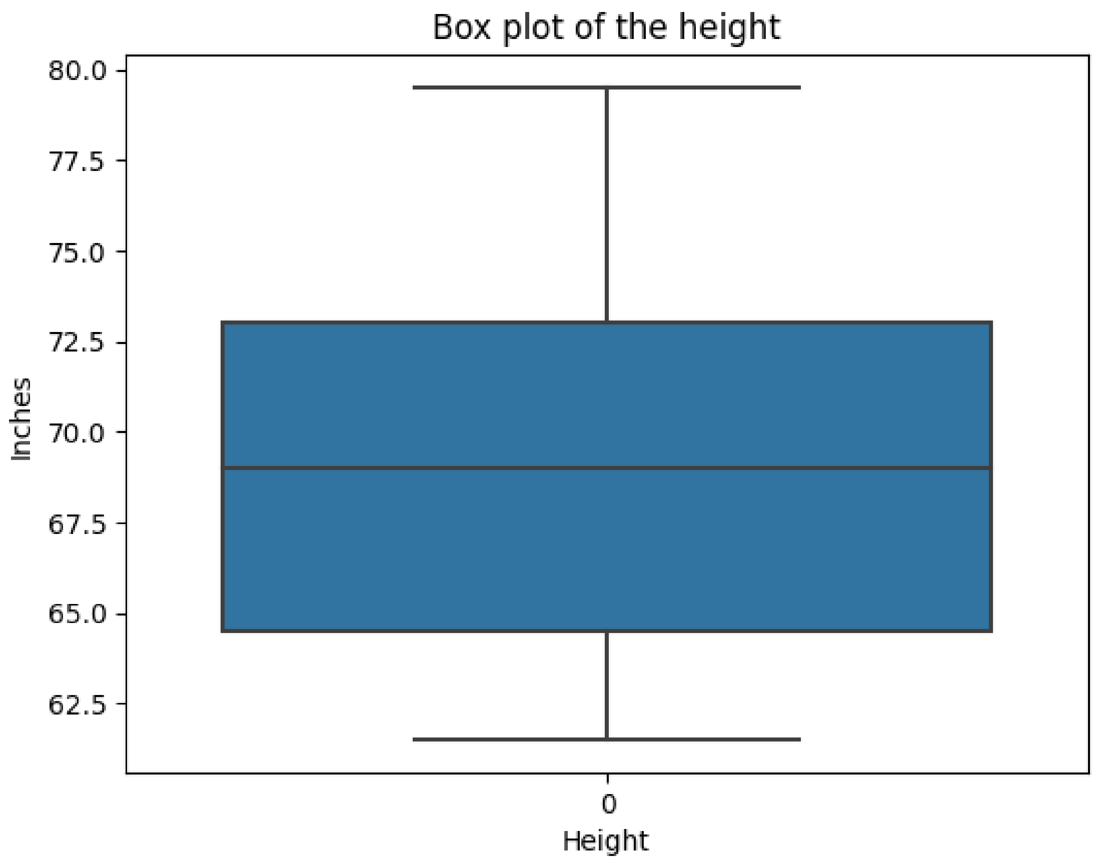
Boxplots

Boxplots do not show the shape of the distribution, but they can give us a better idea about the center and spread of the distribution as well as any potential outliers that may exist. Boxplots and Histograms often complement each other and help an analyst get more information about the data

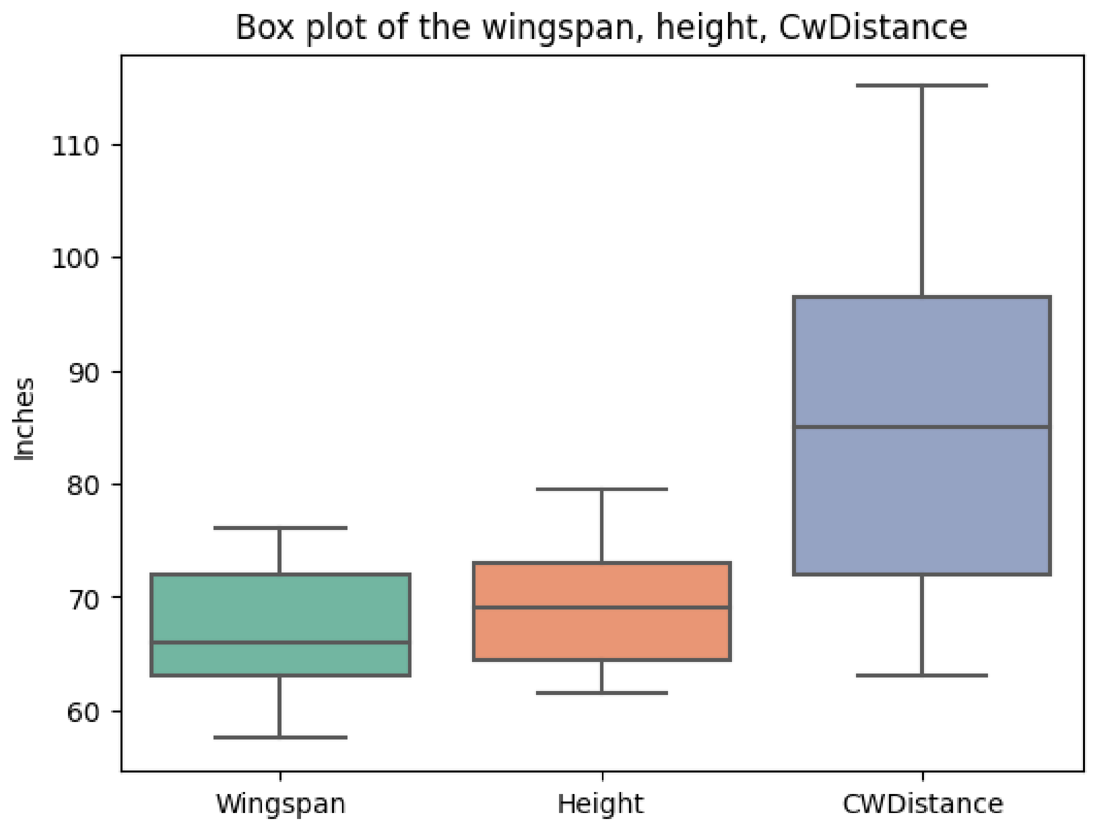
```
# Create the boxplot of the "total bill" amounts
sns.boxplot(df["Wingspan"], orient= "h").set_title("Box plot of the
Wingspan")
plt.xlabel("Wingspan")
plt.ylabel("Inches")
plt.show()
```



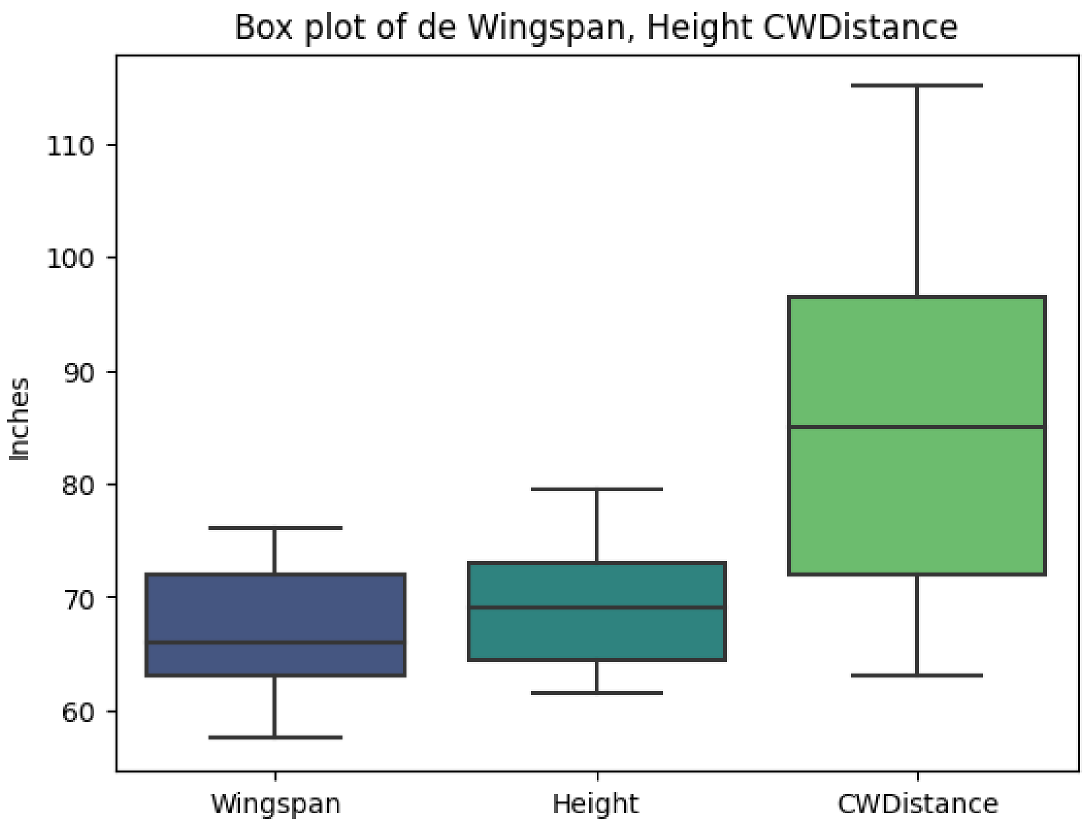
```
# Create the boxplot of the "tips" amounts
sns.boxplot(df["Height"]).set_title("Box plot of the height")
plt.xlabel("Height")
plt.ylabel("Inches")
plt.show()
```



```
# Create the boxplots of the "Wingspan" and of the "Height" amounts
x = df.loc[:,["Wingspan", "Height", "CWDistance"]]
x2bp = sns.boxplot(data=x, orient= "v",palette= "Set2")
x2bp.set_title("Box plot of the wingspan, height, CwDistance")
plt.ylabel("Inches")
plt.show()
```



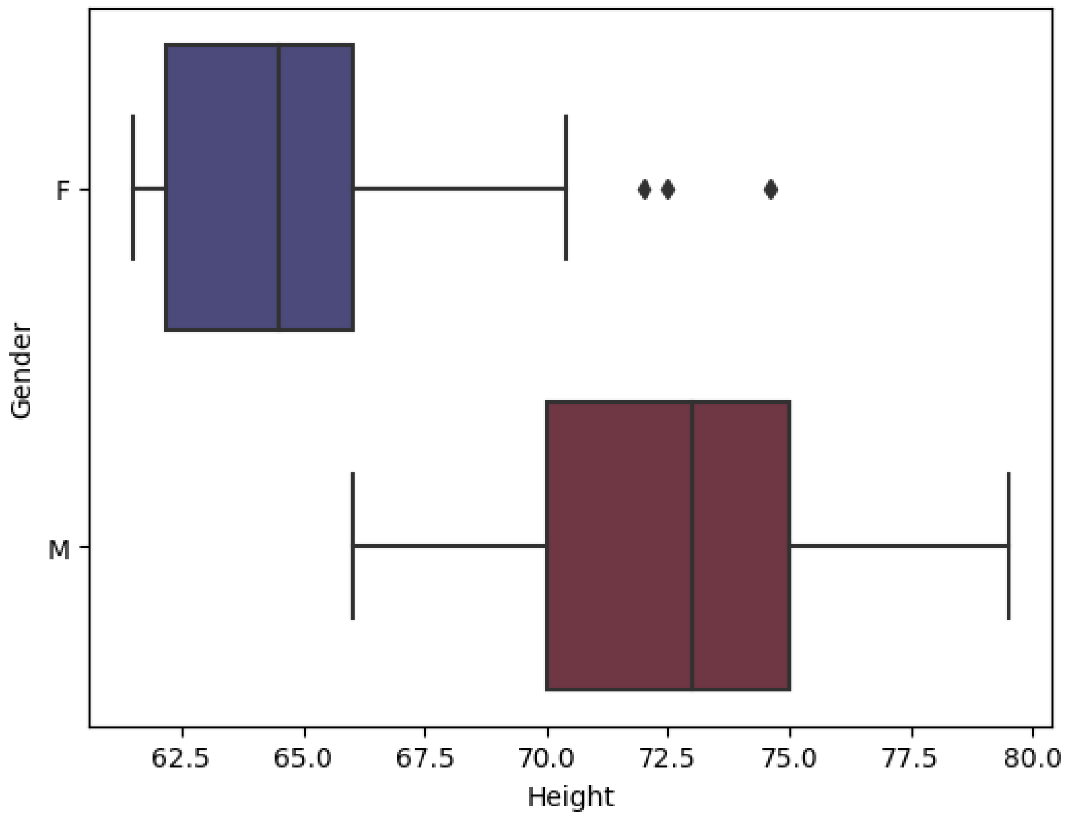
```
# Create the boxplots of the "Wingspan" and of the "tips" amounts

X = df.loc[:, ["Wingspan", "Height", "CWDistance"]]
x2bp = sns.boxplot(data=X, orient="v", palette="viridis")
x2bp.set_title("Box plot of de Wingspan, Height CWDistance")
plt.ylabel("Inches")
plt.show()
```



Boxplots plotted by groups

While looking at a single variable is interesting, it is often useful to see how a variable changes in response to another. Thus, we can create a side-by-side boxplots of one quantitative variable grouped by another categorical variables.

```
# Create side-by-side boxplots of the "Height" grouped by "Gender"

sns.boxplot(x = df["Height"], y = df["Gender"], palette = "icefire")
plt.show()
```



Histograms and boxplots plotted by groups

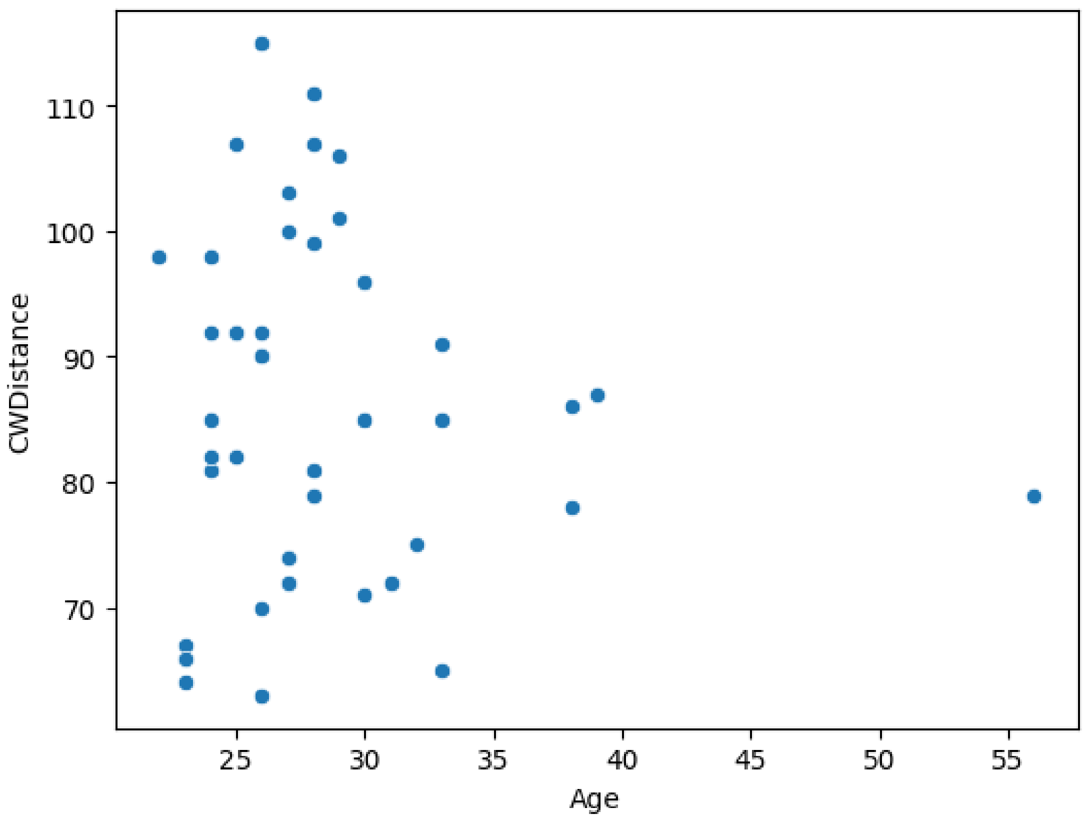
We cal also create both boxplots and histograms of one quantitative variable grouped by another categorical variables

```
# Create a boxplot and histogram of the "tips" grouped by "Gender"
```

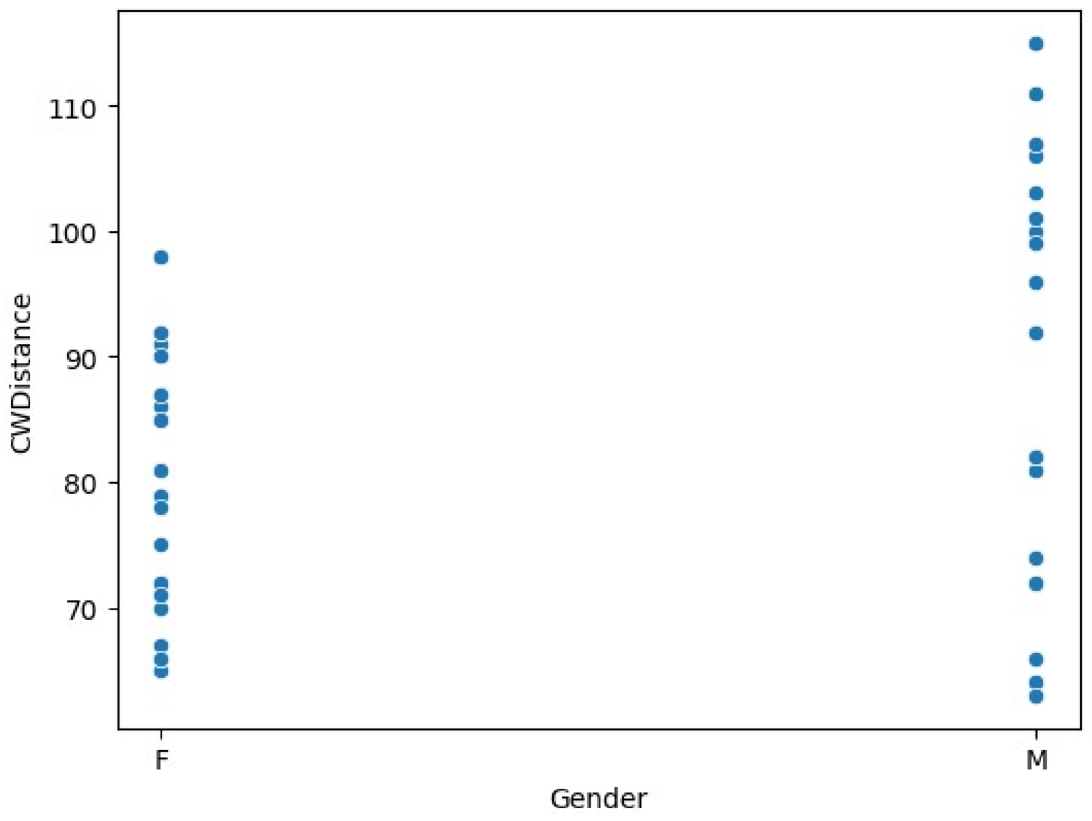
Scatter plot

Plot values of one variable versus another variable to see how they are correlated

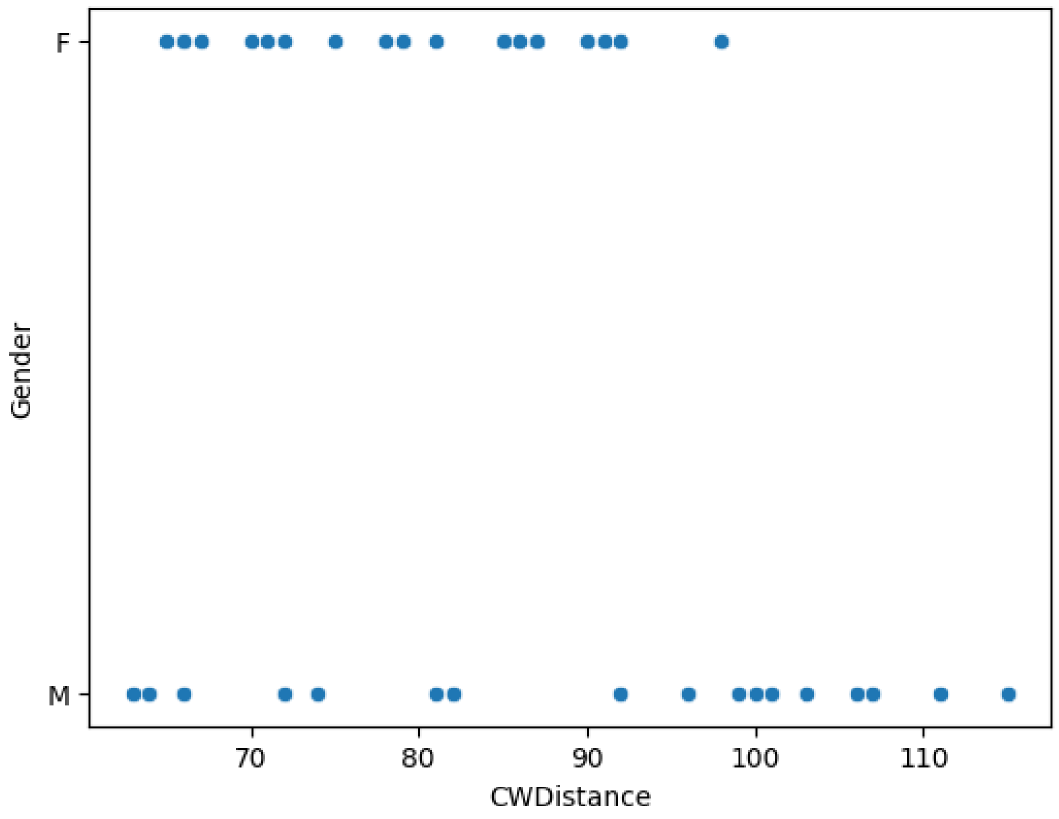
```
# scatter plot between two variables
sns.scatterplot(data = df, y = "CWDistance", x = "Age")
plt.show()
```



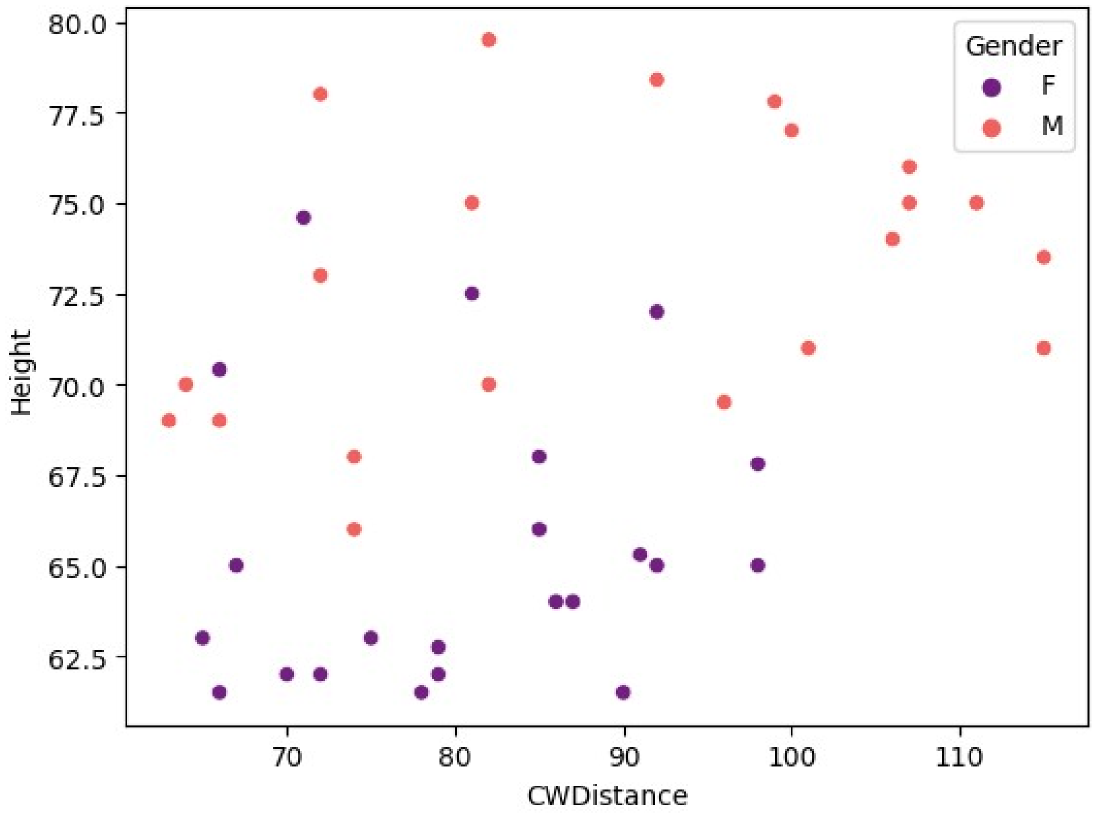
```
# scatter plot between two variables (one categorical)

sns.scatterplot(data = df, y = "CWDistance", x = "Gender")
plt.show()
```



```
# scatter plot between two variables (one categorical)

sns.scatterplot(data = df, x = "CWDistance", y = "Gender")
plt.show()
```



```
# scatter plot between two variables grouped according to a
categorical variable
sns.scatterplot(data = df, x = "CWDistance", y = "Height", hue =
"Gender", palette = "magma")
plt.show()
```



```
# scatter plot between two variables grouped according to a
categorical variable and with size of markeers
```

Final remarks

- Visualizing your data using **tables**, **histograms**, **boxplots**, **scatter plots** and other tools is essential to carry put analysis and extract conclusions
- There are several ways to do the same thing
- The **Seaborn** package provides visualisations tools that allow to explore data from a graphical perspective

Activity: work with the iris dataset

Repeat this tutorial with the iris data set and respond to the following inquiries

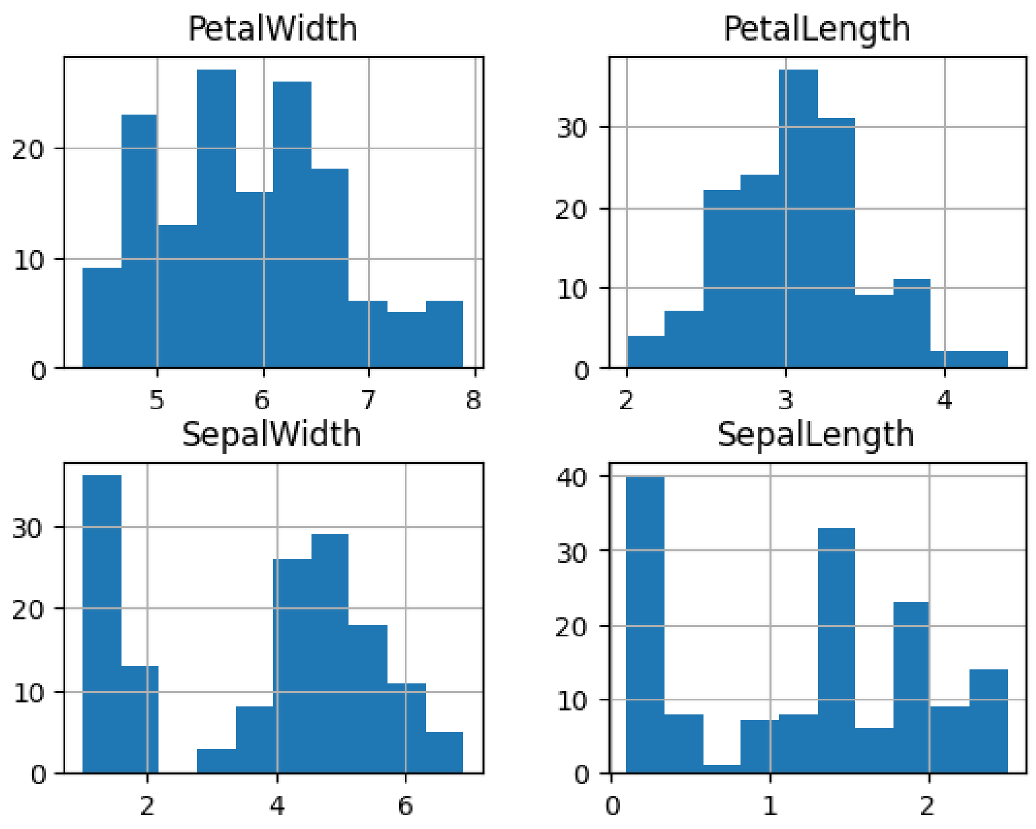
1. Plot the histograms for each of the four quantitative variables
2. Plot the histograms for each of the quantitative variables
3. Plot the boxplots for each of the quantitative variables

4. Plot the boxplots of the petal width grouped by type of flower

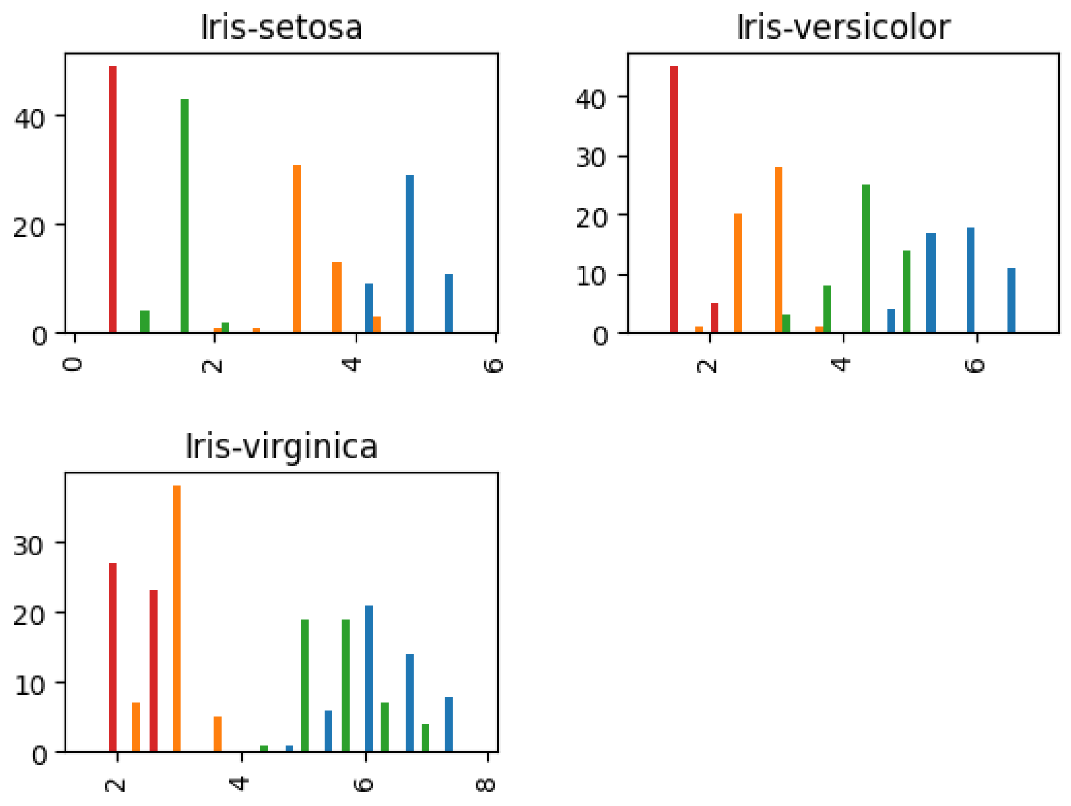
5. Plot the boxplots of the setal length grouped by type of flower

6. Provide a description (explaination from your observations) of each of the quantitative variables

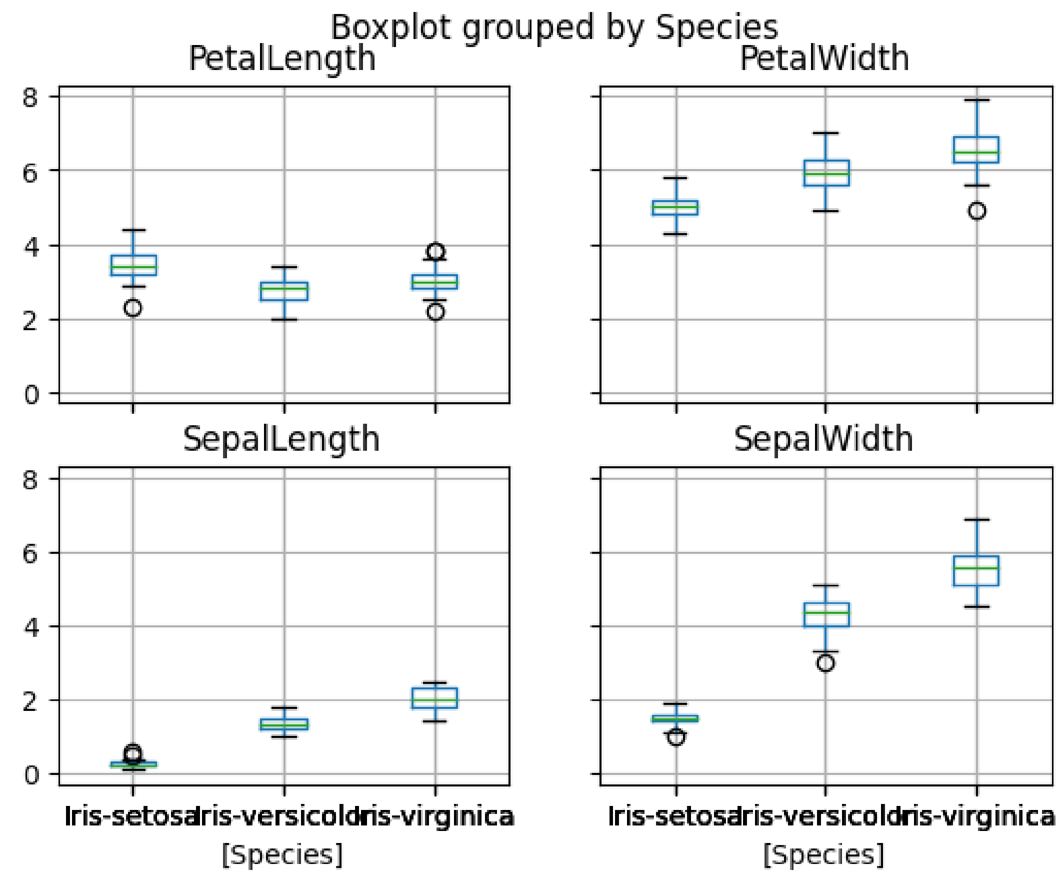
```
dfIris= pd.read_csv("datasets/iris/iris.csv")
dfIris.columns = ["PetalWidth", "PetalLength", "SepalWidth",
"SepalLength", "Species"]

#Plot the histograms for each of the four quantitative variables
dfIris.hist()
plt.show()
```



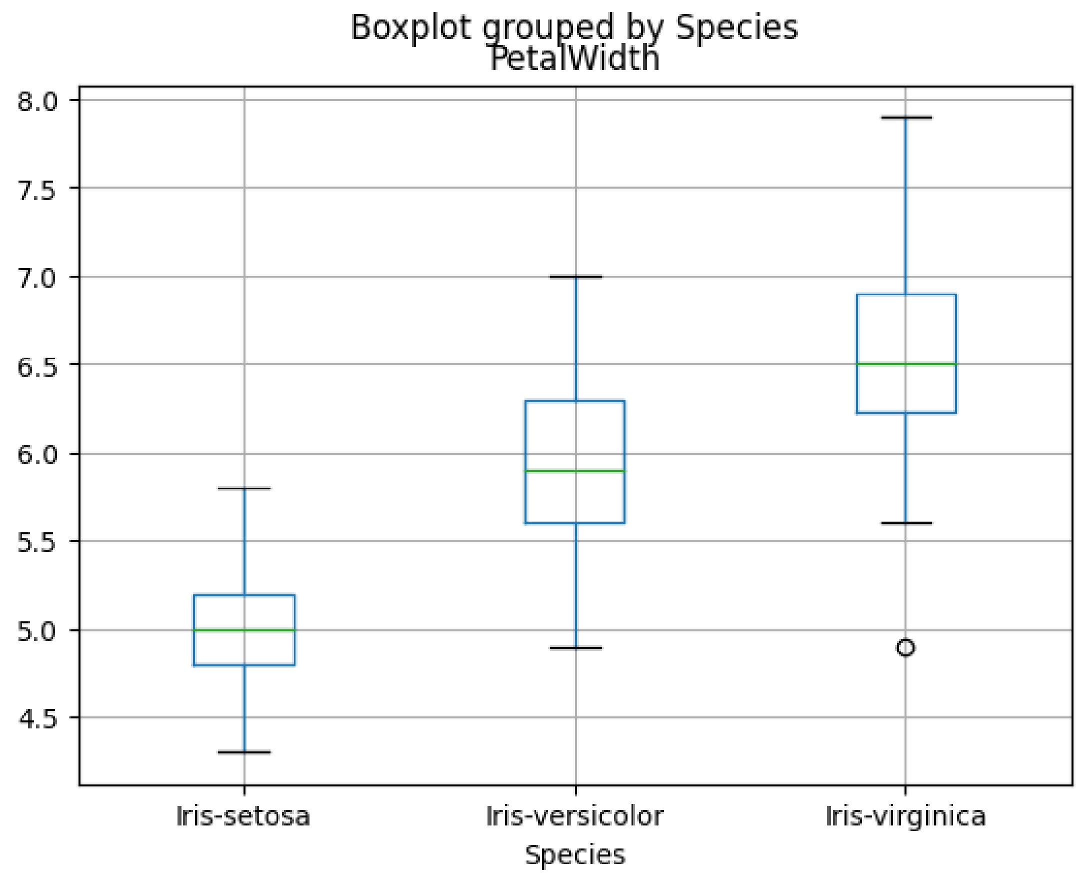
```
#Plot the histograms for each of the quantitative variables, grouped
by the species
dfIris.hist(by = "Species")
plt.show()
```



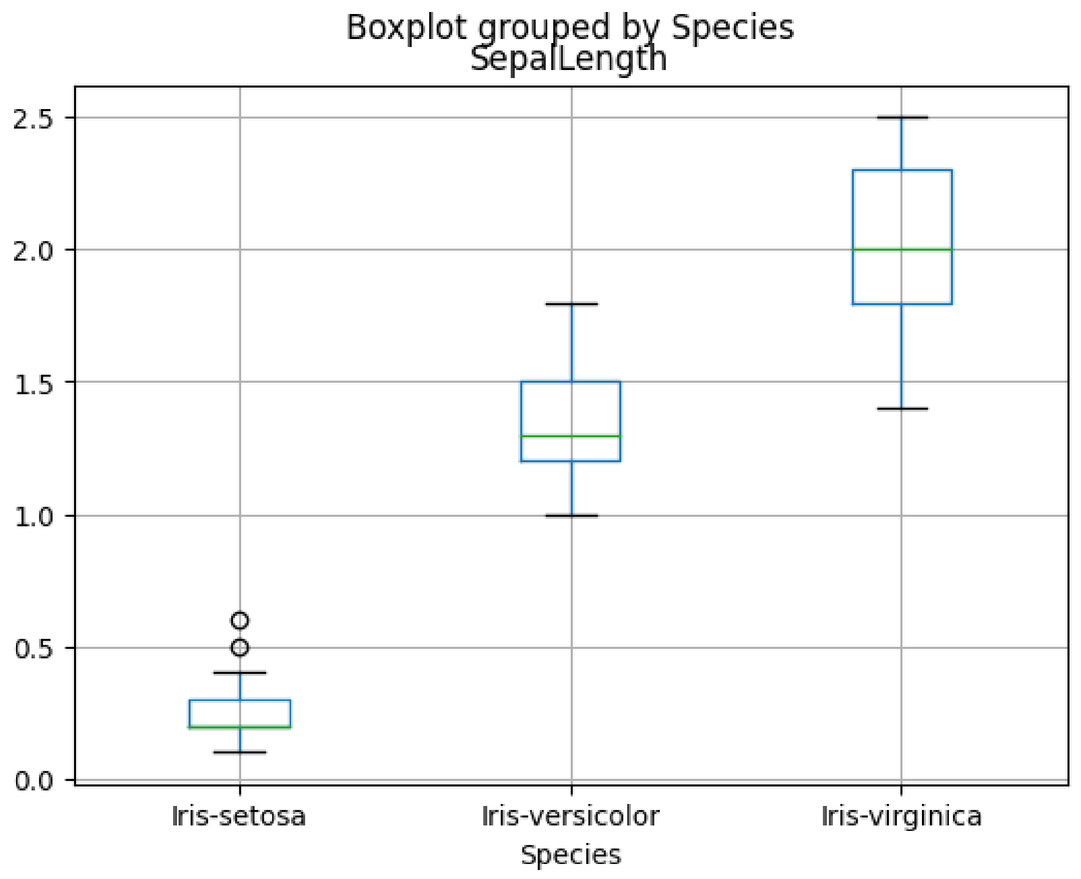
```
#Plot the boxplots for each of the quantitative variables, grouped by
the species
dfIris.boxplot(by = "Species")
plt.show()
```



```
#Plot the boxplots of the petal width grouped by type of flower
dfIris.boxplot(column = "PetalWidth", by = "Species")
plt.show()
```



```
#Plot the boxplots of the setal length grouped by type of flower
dfIris.boxplot(column = "SepalLength", by = "Species")
plt.show()
```


Boxplot grouped by Species
SepalLength
2.5
2.0
1.5
1.0
0.5
0.0
Iris-setosa
Iris-versicolor
Iris-virginica
Species