**Numpy and MatPlotLib**

Exercise 1

Read the data from 'student_grades_3.csv'. The file contains the results of the November exams of some students. Use loadtxt() and create 3 integer arrays to calculate respectively the max, min and mean of each subject.

Note the first line contains headers, use 'skiprows' as a parameter for the loadtxt() function.

student_grades_3.csv | student_grades_2.txt

```
1  Python;Linux;Routing & Switching
2  11;10;16
3  12;14;10
4  8;19;11
5  13;15;18
```

Analyse the data via numpy as shown in the screenshot below:

```
Grades analysis Python:
max 20    min 3  mean   11.05
Grades analysis Linux:
max 19    min 2  mean   11.5
Grades analysis Routing & Switching:
max 18    min 5  mean   13.1
```

Exercise 2

We collected the data of our module exams in 4 different text files (points_python.txt, points_networks.csv, points_web.txt and linx.csv)

The files contain the grades of 100 students for each exam out of 20.

Now we want to calculate the total score for each student, but due to Corona **not** every student has taken each exam!

Load all the data in 4 different arrays in Python and combine them in 1 'total' array.

Print the result to the output screen:

```
[[ 1.   13.75]
 [ 2.    1.5 ]
 [ 3.    8.75]
 [ 4.   11.5 ]
 [ 5.   10.5 ]
```

Next, calculate the marks for each student on 100%.

Tip: you can multiply a dimension of an array using the following syntax:
array[ :,dimension number] *= multiplier

Print the result to the output screen:

```
[[ 1.   68.75]
 [ 2.    7.5 ]
 [ 3.   43.75]
 [ 4.   57.5 ]
 [ 5.   52.5 ]
```

Finally, print the following output:

```
November 2021 Exam results:
---------------------------
The highest score this exam period is: 91.25%
The lowest score this exam period is: 7.5%
```

Exercise 3

A sensor outside is collecting data. We have a small text file (Temperatures.txt) with some data points (days and temperature). Create 2 arrays based on the measurement data and visualize the data using Matplotlib.

You can experiment with the different visualization options. In this example we used a basic plot based on 2 arrays.

Temperatures.txt

```
12,18
13,19
14,20
15,21
16,22
17,22
18,21
19,21
```

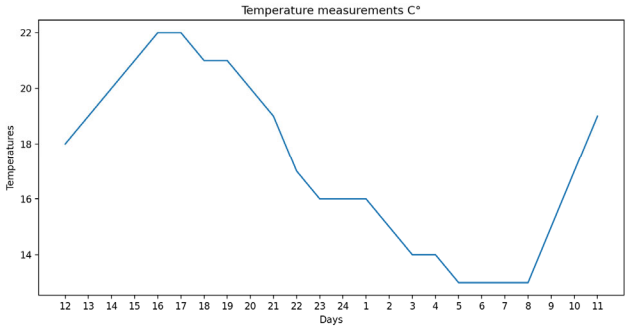
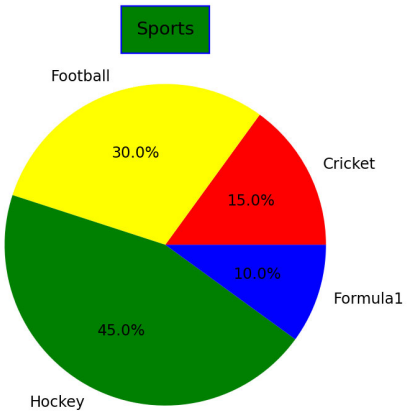



Figure out for yourself how to rotate the X axis in matplotlib!

Exercise 4:

Find out yourself how to create a pie chart.

labels = 'Cricket', 'Football', 'Hockey', 'Formula1'

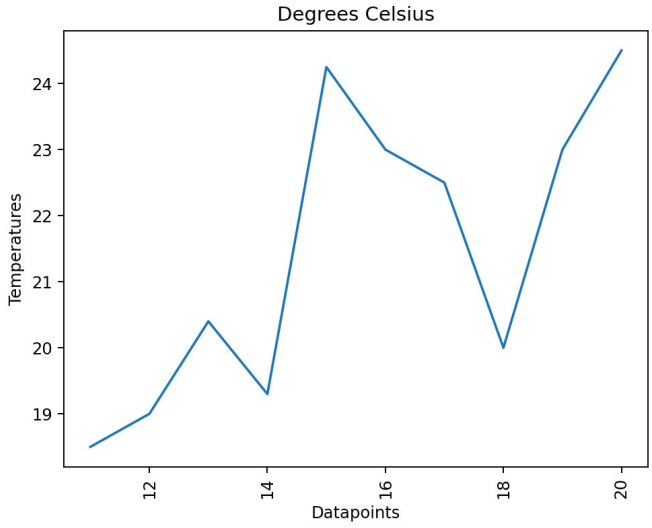



Exercise 5

A sensor collecting data unfortunately collects data in Fahrenheit. We however want to present the data to our user in degrees Celsius.

Write a function Convert with 1 parameter (the file name you're going to convert) which loads the import file and converts the measurement data from degrees Fahrenheit to degrees Celsius. Use the following formula: °C = (°F -32) *5 / 9. Our function returns an array with the converted data.

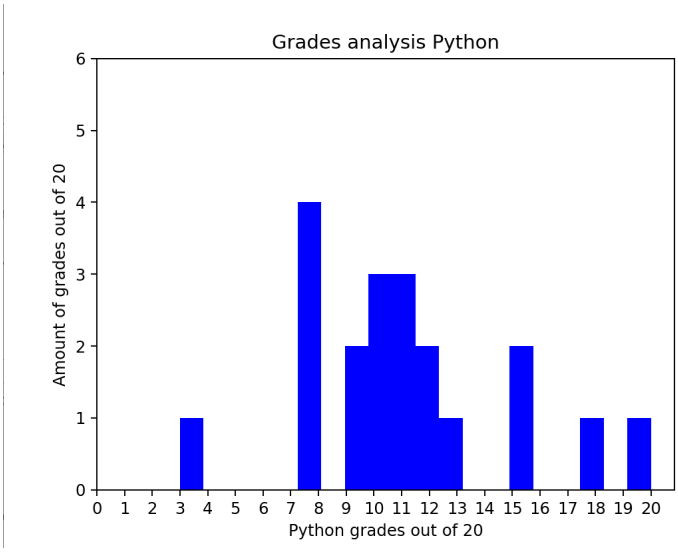
Next, plot the information using a line graph as displayed in the example below:



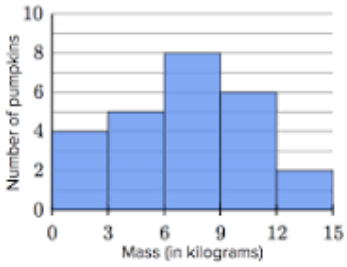
Exercise 6 extension of exercise 1

Plot an histogram with the grades of **Python** which you have extracted in exercise 1.

Use hist(), bins, xticks() and yticks() to plot the following graph.




What are bins in a histogram?




A histogram displays numerical data by grouping data into "bins" of equal width. Each bin is plotted as **a bar whose height corresponds to how many data points are in that bin**. Bins are also sometimes called "intervals", "classes", or "buckets".