iscte

INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

UPskill


<Welcome To>
python™

1.

Overall Program Content

Web development with Python	Hours
Work skills development	50
Python Programming Introduction	**150**
Web Programming Introduction (html/css)	100
Databases Concepts and Structures	50
Web Servers Programming	150
Web services development	150
Total	650

iscte
INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

Python programming Introduction Content

1. Course Introduction

- Why Python?
- Python Applications
- Installation Tools
- Building your code catalog
- Useful websites

Python programming Introduction Content

2. Data types/outputs/inputs
3. Operators
4. Functions and Modules

Python programming Introduction Content

5. Conditional statements and expression
6. Loops
7. Work with standard Library and Modules

Python programming Introduction Content

8. Data structure in python
9. List,
10. Tuple,
11. Dictionaries,
12. Set

Python programming Introduction Content

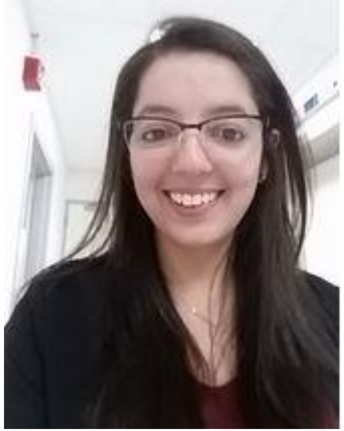
13. Files
14. Functions and Modules
15. Classes
16. Introduction to Numpy
17. Introduction to Pandas

Python programming Introduction Content

18. Introduction to matplotlib for data visualization
19. Data Preprocessing

100% Loaded

Our Teachers:



Joseanne Viana (Josi)

Email: jcova1@iscte-iul.pt

Stefan Postolache

Email:stefanpostolache@edu.ulisboa.pt

Hamed Farkhari

Email:Hamed_Farkhari@iscte-iul.pt

Schedule

Days/modules		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	12-Oct	Joseanne																		
2	13-Oct																			
3	14-Oct																			
4	15-Oct																			
5	16-Oct																			
6	19-Oct						Hamed													
7	20-Oct																			
8	21-Oct																			
9	22-Oct																			
10	23-Oct																			
11	26-Oct																			
12	27-Oct												Stefan							
13	28-Oct																			
14	29-Oct																			
15	30-Oct																			
16	2-Nov															Joseanne				
17	3-Nov																			
18	4-Nov																			
19	5-Nov																			
20	6-Nov																	Hamed		
21	9-Nov																			

<Let’s get started >

Contents

Statistics

Mean / Average

Arithmetic Mean (AM)
Geometric Mean (GM)
Harmonic Mean (HM)

$$\mathrm{AM}(x_1, \ldots, x_n) = \frac{1}{n}(x_1 + \cdots + x_n)$$

$$\mathrm{GM}(x_1, \ldots, x_n) = \sqrt[n]{|x_1 \times \cdots \times x_n|}$$

$$\mathrm{HM}(x_1, \ldots, x_n) = \frac{n}{\frac{1}{x_1} + \cdots + \frac{1}{x_n}}$$

$$\mathrm{HM}\left(\frac{1}{x_1}, \ldots, \frac{1}{x_n}\right) = \frac{1}{\mathrm{AM}(x_1, \ldots, x_n)}$$

$$\mathrm{GM}\left(\frac{1}{x_1}, \ldots, \frac{1}{x_n}\right) = \frac{1}{\mathrm{GM}(x_1, \ldots, x_n)}$$

Weighted arithmetic mean

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

$$\min \leq \mathrm{HM} \leq \mathrm{GM} \leq \mathrm{AM} \leq \max$$

Mean Example

For these values 4, 36, 45, 50, 75 calculate mean?

Arithmetic Mean (AM)

$$\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

$$\frac{4+36+45+50+75}{5} = \frac{210}{5} = 42$$

Geometric Mean (GM)

$$\bar{x} = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} = \left(x_1 x_2 \cdots x_n\right)^{\frac{1}{n}}$$

$$(4 \times 36 \times 45 \times 50 \times 75)^{\frac{1}{5}} = \sqrt[5]{24\,300\,000} = 30.$$

Harmonic Mean (HM)

$$\bar{x} = n\left(\sum_{i=1}^{n} \frac{1}{x_i}\right)^{-1}$$

$$\frac{5}{\frac{1}{4}+\frac{1}{36}+\frac{1}{45}+\frac{1}{50}+\frac{1}{75}} = \frac{5}{\frac{1}{3}} = 15.$$

Mean
Median
Mode

First Sort the values
Then find median!

Comparison of common averages of values { 1, 2, 2, 3, 4, 7, 9 }

Type	Description	Example	Result
Arithmetic mean	*Sum of values of a data set divided by number of values*	(1+2+2+3+4+7+9) / 7	**4**
Median	*Middle value separating the greater and lesser halves of a data set*	1, 2, 2, **3**, 4, 7, 9	**3**
Mode	*Most frequent value in a data set*	1, **2**, **2**, 3, 4, 7, 9	**2**

Mean
Median
Mode
Example

```python
import statistics
from scipy import stats
import numpy as np

a = [4, 36, 45, 50, 75]
b = [1, 2, 2, 3, 4, 7, 9]
c = [6, 3, 9, 6, 6, 5, 9, 9, 3, 1]
```

```python
print(statistics.mean(a))
print(np.mean(b))
print(np.mean(c))
```

```
42
4.0
5.7
```

```python
print(statistics.mode(a))
print(stats.mode(b))
print(stats.mode(c))
```

```
4
ModeResult(mode=array([2]), count=array([2]))
ModeResult(mode=array([6]), count=array([3]))
```

```python
print(statistics.median(a))
print(statistics.median(b))
print(np.median(c))
```

```
45
3
6.0
```

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA
emprego digital

Variance
Standard deviation

Standard deviation (SD)

The measure of the amount of variation or dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

σ , sigma for the population standard deviation

μ , the population mean $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

s , the sample standard deviation

$\bar{x}$, a sample mean

Variance (Var)

it measures how far a set of numbers is spread out from their average value.
The average of the **squared** differences from the Mean

$$\sigma^2,\ s^2,\ \mathrm{Var}(X)$$

$$StandardDeviation = \sqrt{variance}$$

https://www.mathsisfun.com/data/standard-deviation-formulas.html

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA
emprego digital

Sampling

Why Take a Sample?

Mostly because it is easier and cheaper.

Imagine you want to know what the whole country thinks ...
you can't ask millions of people, so instead you ask maybe
1,000 people.

To find out information about the population (such as mean
and standard deviation), we do not need to look
at **all** members of the population; we only need a sample.

But when we take a sample, we lose some accuracy.

Variance
SD
Population Example

Population standard deviation example

Example: Sam has 20 Rose Bushes.

The number of flowers on each bush is

9, 2, 5, 4, 12, 7, 8, 11, 9, 3, 7, 4, 12, 5, 4, 10, 9, 6, 9, 4

Work out the Standard Deviation.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

The handy Sigma Notation says to sum up as many terms as we want:

start at this value

go to this value

what to sum

$$\sum_{n=1}^{4} n = 1+2+3+4 = 10$$

Sigma Notation

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Example: 9, 2, 5, 4, 12, 7, 8, 11, 9, 3, 7, 4, 12, 5, 4, 10, 9, 6, 9, 4

The mean is:

$$\frac{9+2+5+4+12+7+8+11+9+3+7+4+12+5+4+10+9+6+9+4}{20}$$

$$= \frac{140}{20} = \mathbf{7}$$

So:

$$\mu = 7$$

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

Variance
SD
Population Example

Example (continued):

$(9 - 7)^2 = (2)^2 = \mathbf{4}$

$(2 - 7)^2 = (-5)^2 = \mathbf{25}$

$(5 - 7)^2 = (-2)^2 = \mathbf{4}$

$(4 - 7)^2 = (-3)^2 = \mathbf{9}$

$(12 - 7)^2 = (5)^2 = \mathbf{25}$

$(7 - 7)^2 = (0)^2 = \mathbf{0}$

$(8 - 7)^2 = (1)^2 = \mathbf{1}$

... etc ...

And we get these results:

4, 25, 4, 9, 25, 0, 1, 16, 4, 16, 0, 9, 25, 4, 9, 9, 4, 1, 4, 9

Example (continued):

$$\sum_{i=1}^{N}(x_i - \mu)^2$$

Which means: Sum all values from $(x_1-7)^2$ to $(x_N-7)^2$

We already calculated $(x_1-7)^2=4$ etc. in the previous step, so just sum them up:

$= 4+25+4+9+25+0+1+16+4+16+0+9+25+4+9+9+4+1+4+9 = \mathbf{178}$

Example (concluded):

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

$\sigma = \sqrt{(8.9)} = \mathbf{2.983...}$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

Variance
SD
Sample Example

Sample Standard Deviation example

Example: Sam has **20** rose bushes, but only counted the flowers on **6 of them**!

The "population" is all 20 rose bushes,

and the "sample" is the 6 bushes that Sam counted the flowers of.

Let us say Sam's flower counts are:

9, 2, 5, 4, 12, 7

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$

The important change is "N-1" instead of "N"

The symbols also change to reflect that we are working on a sample instead of the whole population:

- The mean is now $\bar{x}$ (for sample mean) instead of **μ** (the population mean),
- And the answer is **s** (for Sample Standard Deviation) instead of **σ**.

But that does not affect the calculations. **Only N-1 instead of N changes the calculations.**

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA
emprego digital

Variance
SD
Sample Example

Sample Standard Deviation example

Example 2: Using sampled values 9, 2, 5, 4, 12, 7

The mean is (9+2+5+4+12+7) / 6 = 39/6 = 6.5

So:

$$\bar{x} = 6.5$$

Example 2 (continued):

$(9 - 6.5)^2 = (2.5)^2 = 6.25$

$(2 - 6.5)^2 = (-4.5)^2 = 20.25$

$(5 - 6.5)^2 = (-1.5)^2 = 2.25$

$(4 - 6.5)^2 = (-2.5)^2 = 6.25$

$(12 - 6.5)^2 = (5.5)^2 = 30.25$

$(7 - 6.5)^2 = (0.5)^2 = 0.25$

Example 2 (concluded):

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

$s = \sqrt{(13.1)} =$ **3.619...**

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

Variance
SD
Statistics vs NumPy

```
import statistics
import numpy as np
```

Entire Population

```
a = [9, 2, 5, 4, 12, 7, 8, 11, 9, 3, 7, 4, 12, 5, 4, 10, 9, 6, 9, 4]
b = [9, 2, 5, 4, 12, 7]
```

a Sample

```
print(np.mean(a))
print(np.mean(b))
```

```
7.0
6.5
```

```
print(np.std(a))               # standard deviation of an entire population
print(statistics.pstdev(a))    # standard deviation of an entire population
print(statistics.stdev(a))
```

```
2.9832867780352594
2.9832867780352594
3.0607876523260447
```

Wrong answers: stdev(a) , std(b)

```
print(np.std(b))
print(statistics.stdev(b))   # standard deviation of a Sample
```

```
3.304037933599835
3.6193922141707713
```

```
print(np.var(a))                 # Variance of an entire population
print(statistics.pvariance(a)) # Variance of an entire population
print(statistics.variance(a))
```

```
8.9
8.9
9.368421052631579
```

Wrong answers: variance(a) , var(b)

```
print(np.var(b))
print(statistics.variance(b)) # Variance of a  Sample
```

```
10.916666666666666
13.1
```

for Entire Population
numpy.std(a)
statistics.pstdev(a)
numpy.var(a)
statistics.pvariance(a)

for a Sample
statistics.stdev(b)
statistics.variance(b)

Quartile

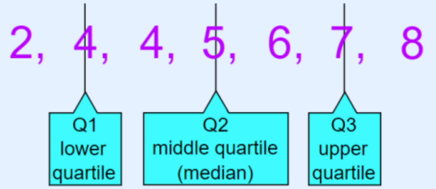
Quartiles are the values that divide a list of numbers into quarters

- ✓ *Put the list of numbers in order*
- ✓ *Then cut the list into four equal parts*
- ✓ *The Quartiles are at the "cuts"*

Example: 5, 7, 4, 4, 6, 2, 8

Put them in order: 2, 4, 4, 5, 6, 7, 8

Cut the list into quarters:



And the result is:

- Quartile 1 (Q1) = **4**
- Quartile 2 (Q2), which is also the Median, = **5**
- Quartile 3 (Q3) = **7**

https://www.mathsisfun.com/data/quartiles.html

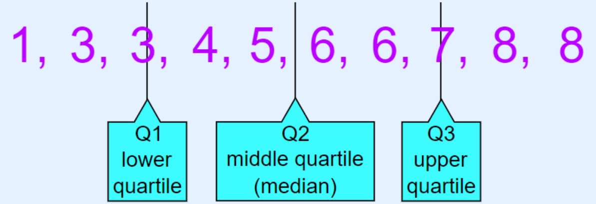
Quartile

Sometimes a "cut" is between two numbers ...
the Quartile is the average of the two numbers

Example: 1, 3, 3, 4, 5, 6, 6, 7, 8, 8

The numbers are already in order

Cut the list into quarters:



In this case Quartile 2 is half way between 5 and 6:

Q2 = (5+6)/2 = **5.5**

And the result is:

- Quartile 1 (Q1) = **3**
- Quartile 2 (Q2) = **5.5**
- Quartile 3 (Q3) = **7**

Quartile

5 Methods to calculate Quartiles

when the desired quantile lies between two data points $i < j$

Linear

$i + (j - i) * fraction$

where `fraction` is the fractional part of the index surrounded by i and j

Lower

i

Higher

j

Nearest

i or j whichever is nearest

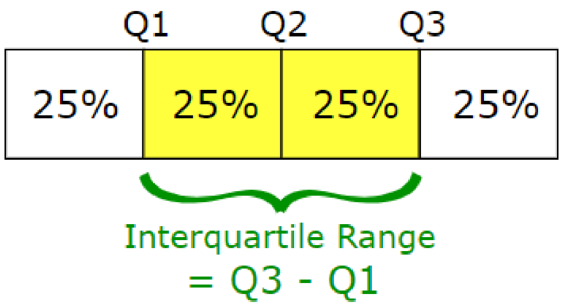
Midpoint

$(i + j) / 2$

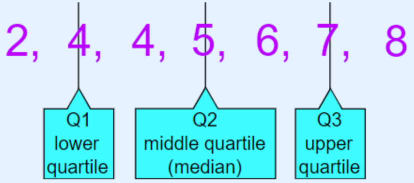
IQR

Interquartile Range

The "Interquartile Range" is from Q1 to Q3



Example:



The **Interquartile Range** is:

Q3 − Q1 = 7 − 4 = **3**

Data Visualization

Matplotlib

Plot
Simple Example

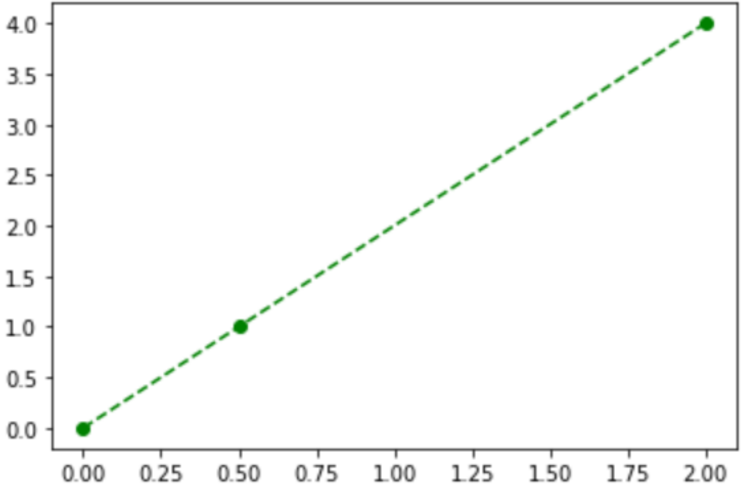
```
import matplotlib.pyplot as plt
```

```
plt.plot?
```

Plot y versus x as lines and/or markers.

for points A(0, 0) , B(0.5, 1) , C(2, 4)

```
x = [0, 0.5, 2]
y = [0, 1, 4]

plt.plot(x, y, 'go--');

# plt.plot(x, y, color='green', marker='o', linestyle='dashed');
```



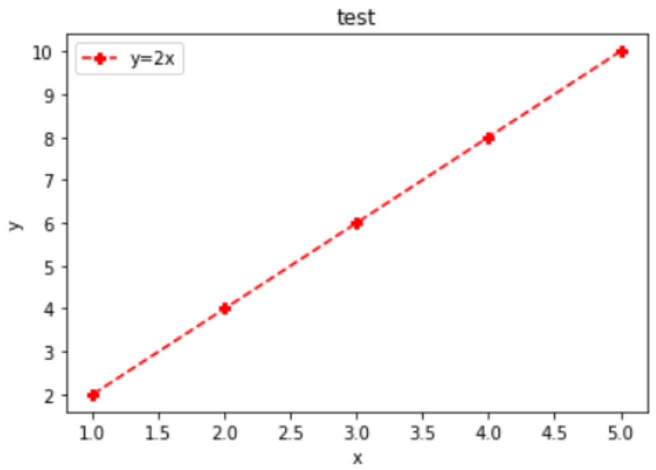
Plot
Legend
Title
xlabel
ylabel

```
import matplotlib.pyplot as plt
```

```
x=[1,2,3,4,5]
y=[2,4,6,8,10]
```

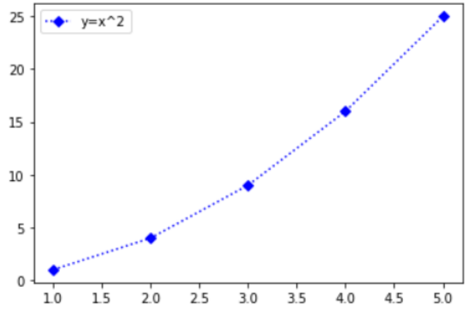
```
plt.plot(x,y,'r--P',label='y=2x')
plt.legend()
plt.title('test')
plt.xlabel('x')
plt.ylabel('y')
```

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

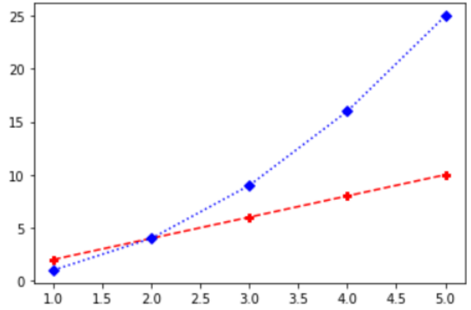



```
x1=[1,2,3,4,5]
y1=[1,4,9,16,25]
```

```
plt.plot(x1,y1,'b:D' , label='y=x^2');
plt.legend();
```




```
plt.plot(x,y,'r--P',label='y=2x');
plt.plot(x1,y1,'b:D' , label='y=x^2');
```



iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

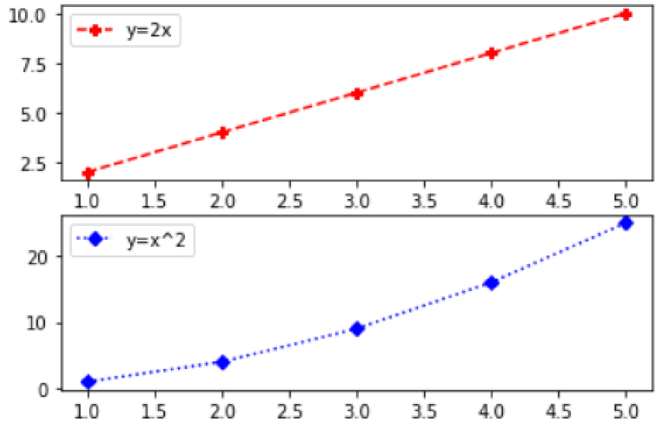
Plot
Subplot
Legend
Savefig

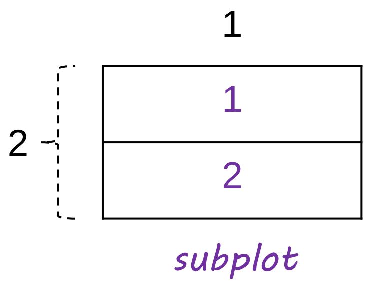
```python
import matplotlib.pyplot as plt
```

```python
x=[1,2,3,4,5]
y=[2,4,6,8,10]
```

```python
x1=[1,2,3,4,5]
y1=[1,4,9,16,25]
```

```python
plt.subplot(211)
plt.plot(x,y,'r--P',label='y=2x')
plt.legend()

plt.subplot(212)
plt.plot(x1,y1,'b:D' , label='y=x^2')
plt.legend()

plt.savefig('D:/plot.png')
```

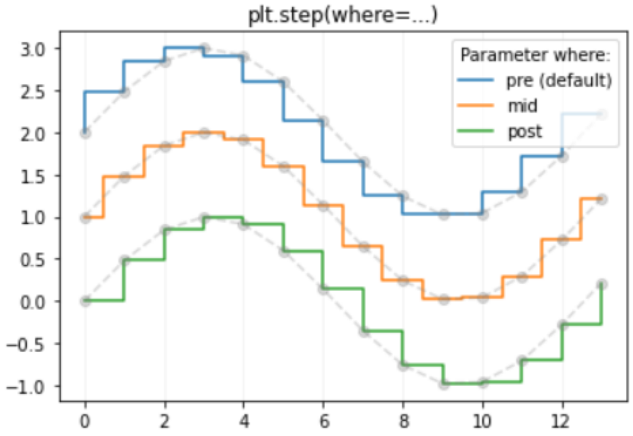




Plot
Step
Grid
Legend
Title

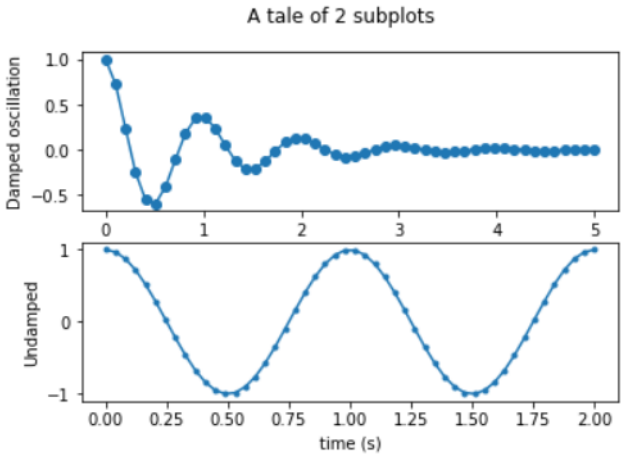
```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(14)
y = np.sin(x / 2)

plt.step(x, y + 2, label='pre (default)')
plt.plot(x, y + 2, 'o--', color='grey', alpha=0.3)

plt.step(x, y + 1, where='mid', label='mid')
plt.plot(x, y + 1, 'o--', color='grey', alpha=0.3)

plt.step(x, y, where='post', label='post')
plt.plot(x, y, 'o--', color='grey', alpha=0.3)

plt.grid(axis='x', color='0.95')
plt.legend(title='Parameter where:')
plt.title('plt.step(where=...)')
plt.show()
```

Function



Plot
Subplots
Suptitle
Set_xlabel
Set_ylabel

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

x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)

y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)

fig, (ax1, ax2) = plt.subplots(2, 1)
fig.suptitle('A tale of 2 subplots')

ax1.plot(x1, y1, 'o-')
ax1.set_ylabel('Damped oscillation')

ax2.plot(x2, y2, '.-')
ax2.set_xlabel('time (s)')
ax2.set_ylabel('Undamped')

plt.show()
```

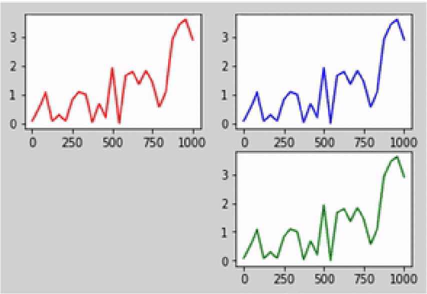


State-based vs Object-oriented

Subplot vs Subplots

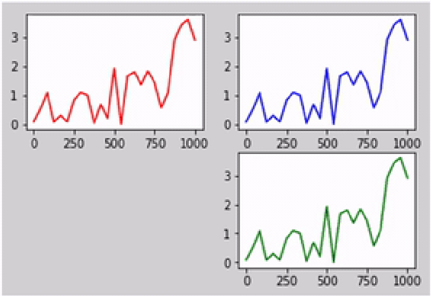
State-based vs *Object-oriented*

STATE BASED



```
plt.figure(facecolor='lightgrey')
plt.subplot(2,2,1)
plt.plot(data_x, data_y, 'r-')
plt.subplot(2,2,2)
plt.plot(data_x, data_y, 'b-')
plt.subplot(2,2,4)
plt.plot(data_x, data_y, 'g-')
```

OBJECT ORIENTED



```
fig, ax = plt.subplots(2,2)
fig.set_facecolor('lightgrey')
ax[0,0].plot(data_x, data_y, 'r-')
ax[1,0].plot(data_x, data_y, 'b-')
fig.delaxes(ax[1,0])
ax[1,1].plot(data_x, data_y, 'g-')
# ... complete!
```

1	2
3	4

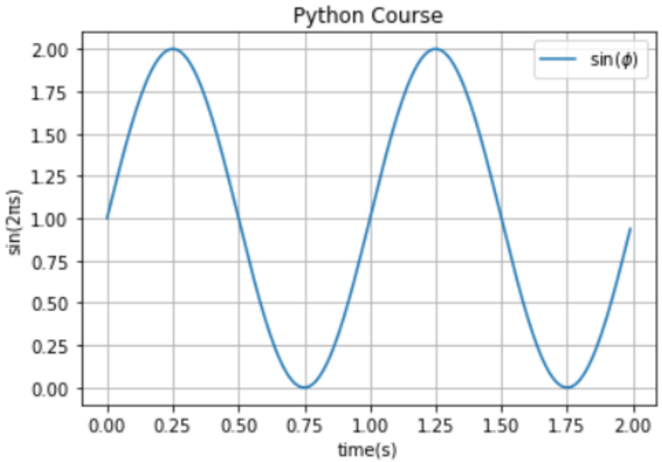
subplot() vs *subplots()*

STATE BASED

OBJECT ORIENTED

Plot
Subplots
Set(title, xlabel, ylabel)
Grid
Legend
Savefig

```python
import matplotlib.pyplot as plt
import numpy as np

# Data for plotting
x = np.arange(0.0, 2.0, 0.01)
y = 1 + np.sin(2 * np.pi * x)

fig, ax = plt.subplots()
ax.plot(x, y, label='sin($\phi$)')

ax.set(xlabel='time(s)', ylabel='sin(2\u03C0s)', title='Python Course')
ax.grid()
ax.legend()
fig.savefig("test.png")
plt.show()
```



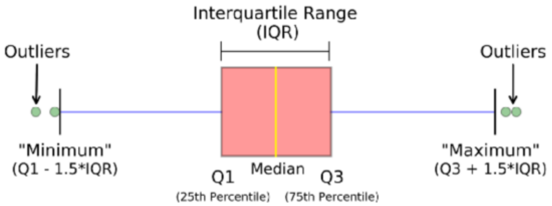
https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html
https://pythonforundergradengineers.com/unicode-characters-in-python.html

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA
emprego digital

Boxplot

Boxplots are a standardized way of displaying the distribution of data based on a five number summary

Minimum > = (Q1 - 1.5 * IQR)
first quartile (Q1)
Median (Q2)
third quartile (Q3)
Maximum < = (Q3 + 1.5 * IQR)



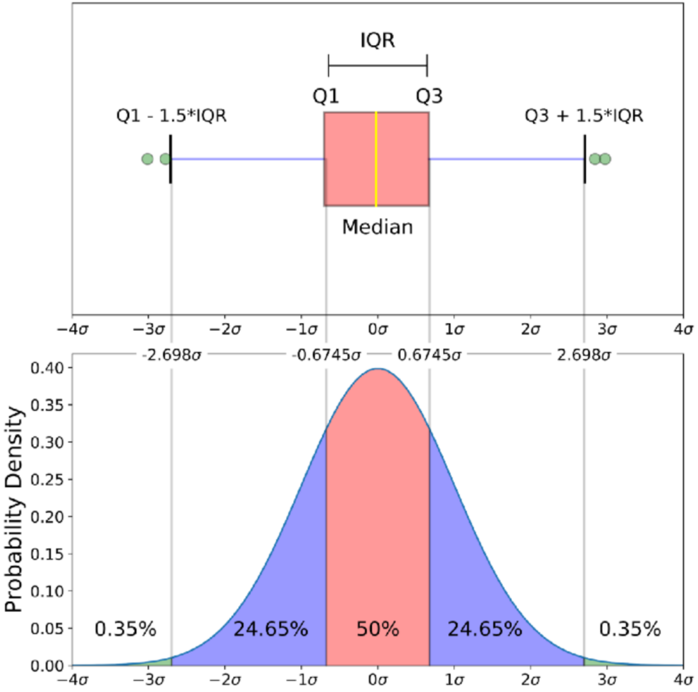
1. Tell you the values of your outliers
2. Identify if data is symmetrical
3. Determine how tightly data is grouped
4. See if your data is skewed

https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA emprego digital

Boxplot

Boxplot on a Normal Distribution



$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

PDF for a Normal Distribution

mean (μ) of 0
standard deviation (σ) of 1

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

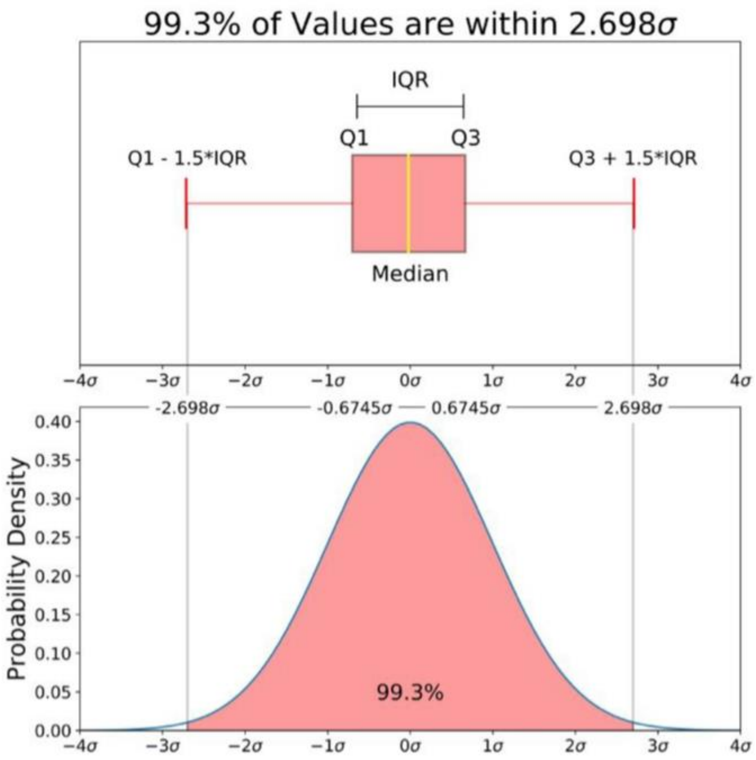
PDF for a Normal Distribution

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA
emprego digital

Boxplot

Boxplot on a Normal Distribution

By removing outliers, we have access to 99.3% of data on a normal distribution



Boxplot Example

```python
import numpy as np
```

```python
import matplotlib.pyplot as plt
# from matplotlib import pyplot as plt
```

```python
data=np.array([-10 , -5 , -2 , -1 , 0 , 1 , 2 , 3 , 4])
```

```python
q1 = np.quantile(data, .25)
q1
```

```
-2.0
```

```python
q2 = np.quantile(data, .50)
q2
```

```
0.0
```

```python
q3 = np.quantile(data, .75)
q3
```

```
2.0
```

```python
iqr = q3 - q1
iqr
```

```
4.0
```

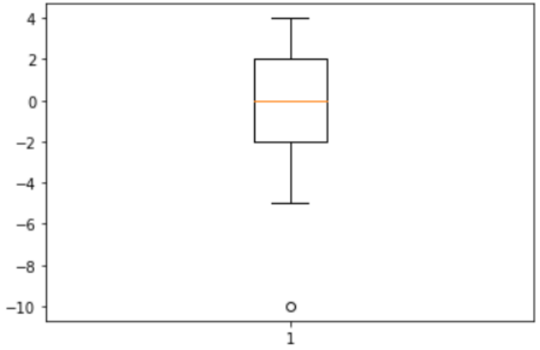
```python
lv = q1 - 1.5 * iqr
lv
```

```
-8.0
```

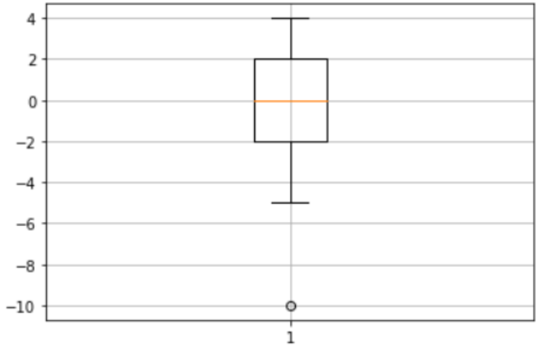
```python
hv = q3 + 1.5 * iqr
hv
```

```
8.0
```

```python
plt.boxplot(data);
```




```python
plt.boxplot(data);
plt.grid()
```



iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

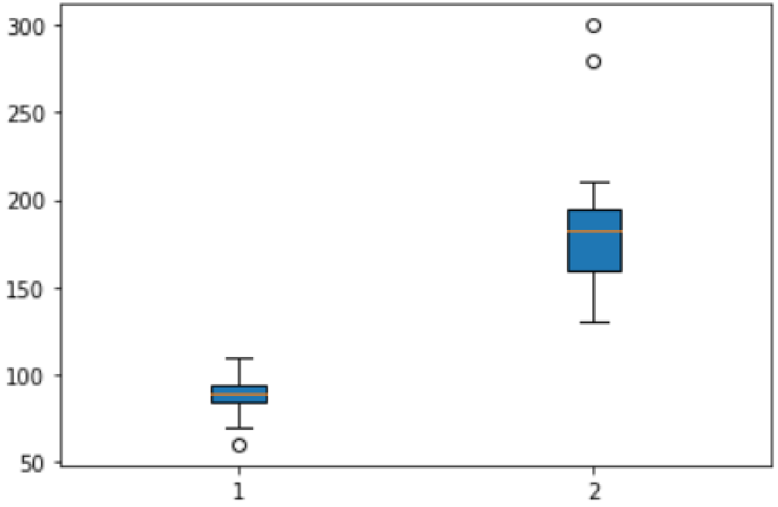
Boxplot Example

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

```
class1= np.array([60,70,80,83,85,87,88,89,90,92,94,95,97,100,110])
```

```
class2 = np.array([130,143,150,158,160,170,175,182,185,188,190,200,210,280,300])
```

```
plt.boxplot([class1,class2],patch_artist=True);
```



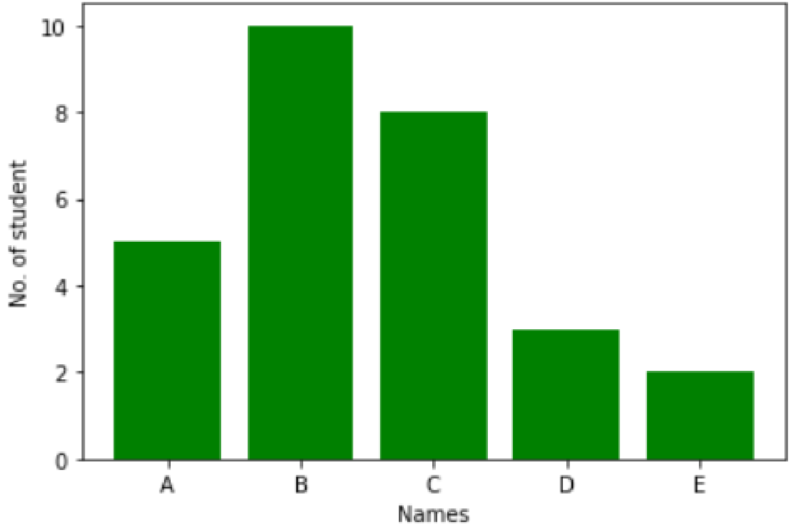
Bar

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

```
names = ['A', 'B', 'C', 'D', 'E']
```

```
values = [ 5, 10,  8,  3 , 2]
```

```
plt.bar(names, values, color='green');
plt.xlabel('Names');
plt.ylabel('No. of student');
```

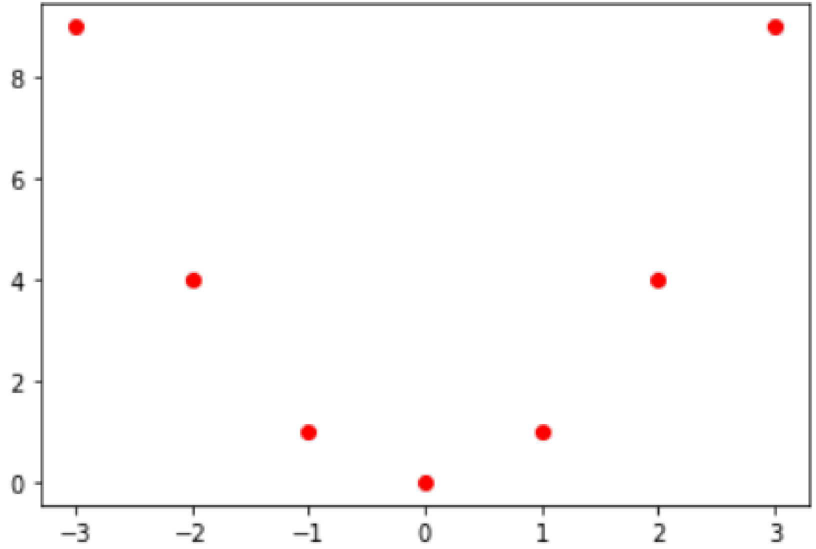


Scatter

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

```
x=np.array([-3,-2 ,-1 , 0 , 1 , 2 , 3 ])
y=np.array([ 9, 4 , 1 , 0 , 1 , 4 , 9])
```

```
plt.scatter(x, y,  c='r');
```

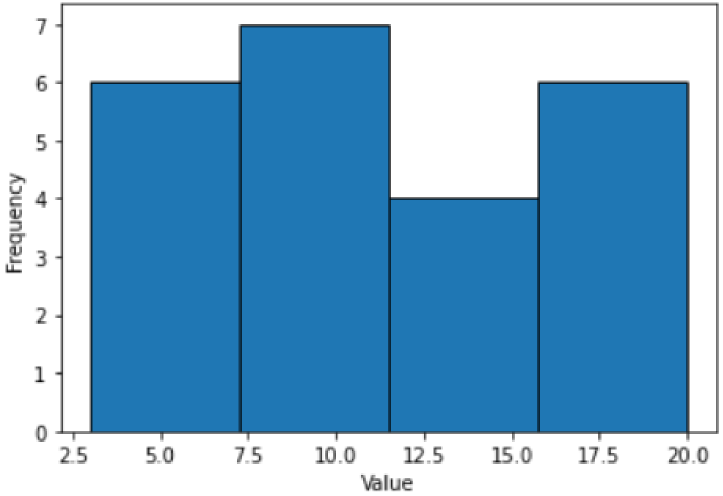


Histogram

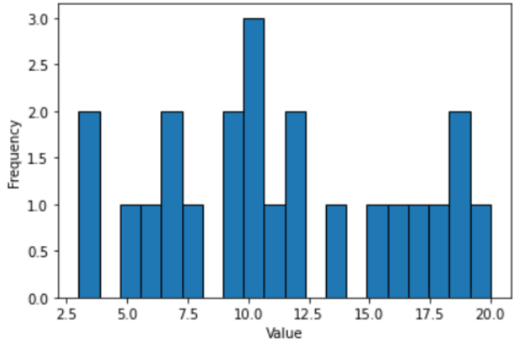
```python
import numpy as np
import matplotlib.pyplot as plt
```

```python
data = np.array([3,3,5,6,7,7,8,9,9,10,10,10,11,12,12,14,15,16,17,18,19,19,20])
```

```python
plt.hist(data, bins=4, edgecolor='black');
plt.xlabel('Value');
plt.ylabel('Frequency');
```




```python
plt.hist(data, bins=20, edgecolor='black');
plt.xlabel('Value');
plt.ylabel('Frequency');
```

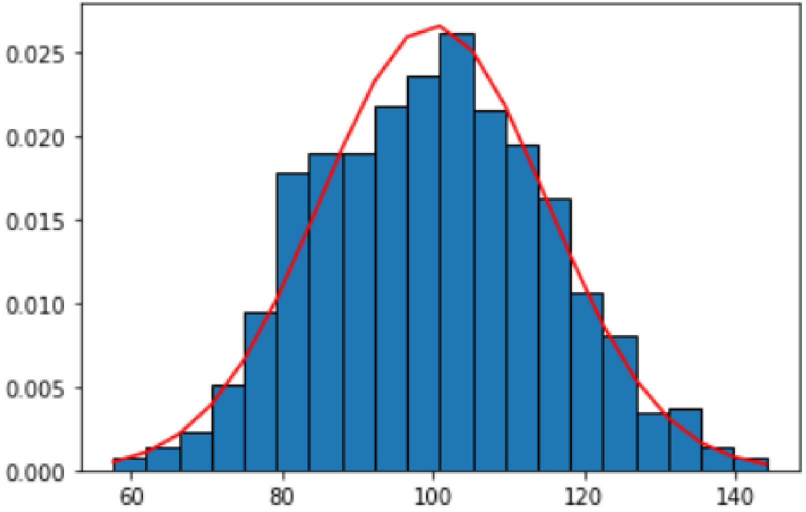


Histogram

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
```

```
x = 100 + 15 * np.random.randn(1000)
```

```
n, bins, _ = plt.hist(x, bins=20, edgecolor='black', density=1)
y = norm.pdf(bins, 100, 15)
plt.plot(bins, y, 'r');
```

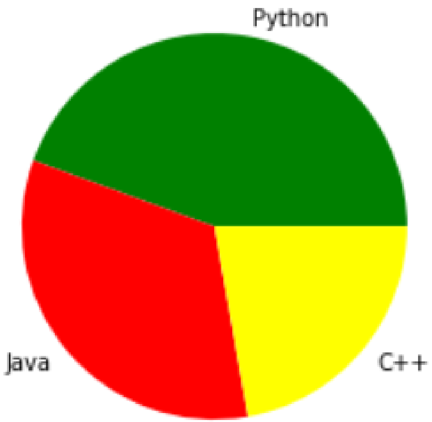


Pie

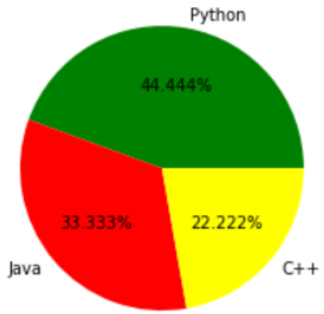
```
import matplotlib.pyplot as plt
```

```
l = ['Python', 'Java' , 'C++']
s = [  200    ,  150    ,  100]
c = [ 'green', 'red', 'yellow']
```

```
plt.pie(s,  labels=l, colors=c);
```




```
plt.pie(s,  labels=l, colors=c, autopct='%1.3f%%');
```

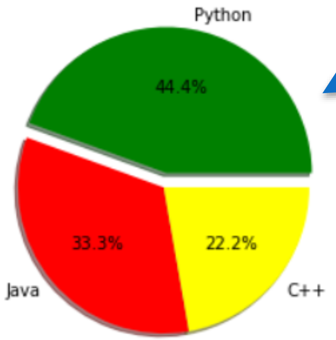



```
plt.pie(s,  labels=l, colors=c, startangle=120 );
```

Rotate

Pie

```
plt.pie(s,  labels=l, colors=c, explode=(0.1, 0, 0), shadow=True, autopct='%1.1f%%');
```



Data Visualization with Pandas

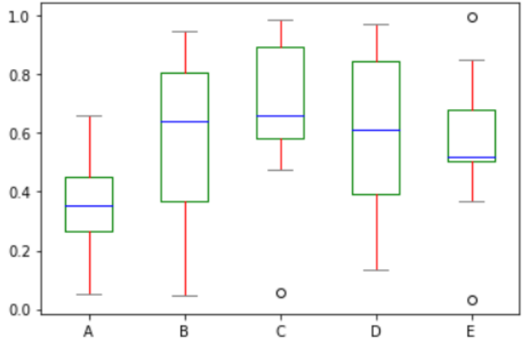
Pandas
Box

```python
import pandas as pd
```

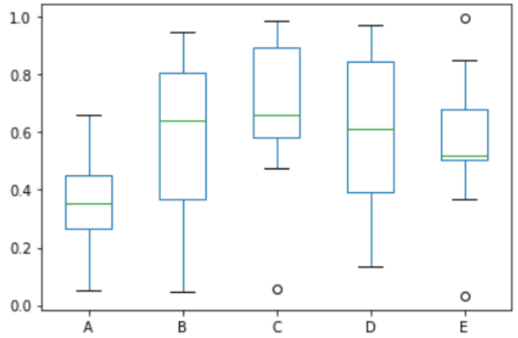
```python
df = pd.DataFrame(np.random.rand(10, 5),
                  columns=['A', 'B', 'C', 'D', 'E'])
df
```

	A	B	C	D	E
0	0.049709	0.045436	0.055949	0.843779	0.512288
1	0.660786	0.840155	0.724785	0.132184	0.850772
2	0.050459	0.669042	0.627813	0.286994	0.032594
3	0.248601	0.601896	0.568547	0.573618	0.524015
4	0.356679	0.829488	0.474301	0.548752	0.503438
5	0.548138	0.290860	0.984472	0.343178	0.369197
6	0.323869	0.739252	0.952468	0.651361	0.717576
7	0.385569	0.162749	0.972111	0.906281	0.573732
8	0.475793	0.608820	0.683542	0.841382	0.502442
9	0.347030	0.949040	0.636254	0.970489	0.995102

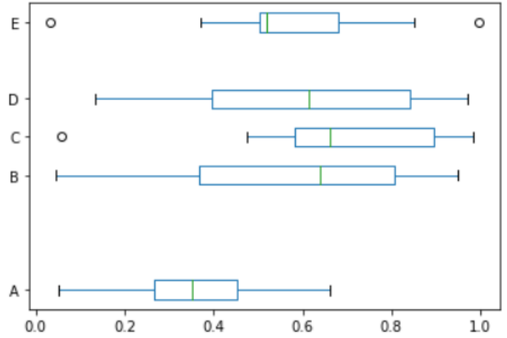
```python
df.plot.box(color={'boxes': 'Green', 'whiskers': 'red',
                   'medians': 'Blue', 'caps': 'Gray'});
```




```python
df.plot.box();
```



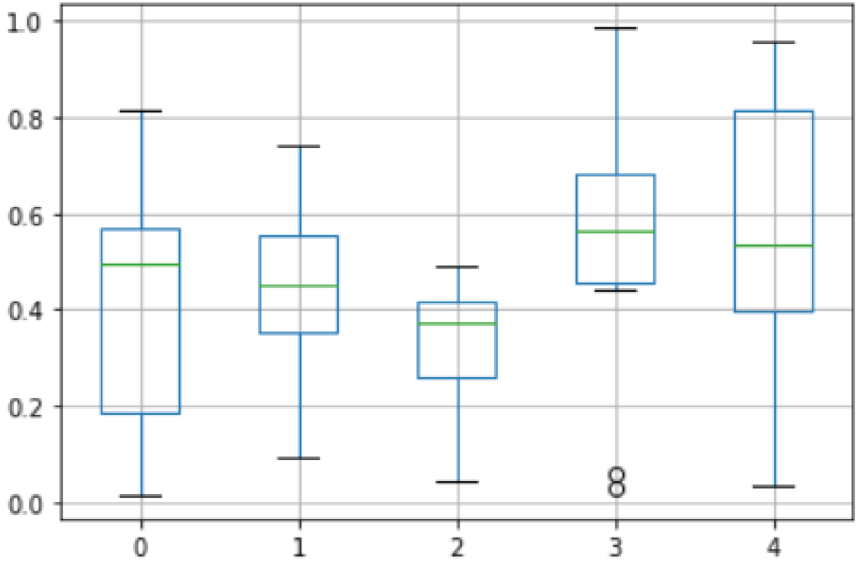

```python
df.plot.box(vert=False, positions=[1, 4, 5, 6, 8]);
```



Pandas
Boxplot

```
import pandas as pd
```

```
df = pd.DataFrame(np.random.rand(10, 5))
df.boxplot();
```



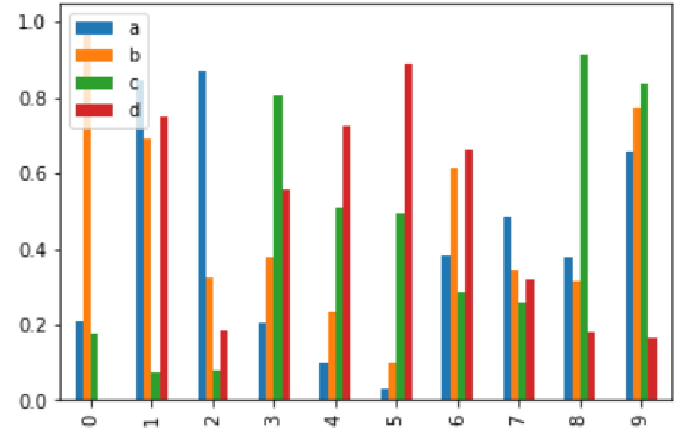
Pandas
Bar
Barh

```python
import pandas as pd
```

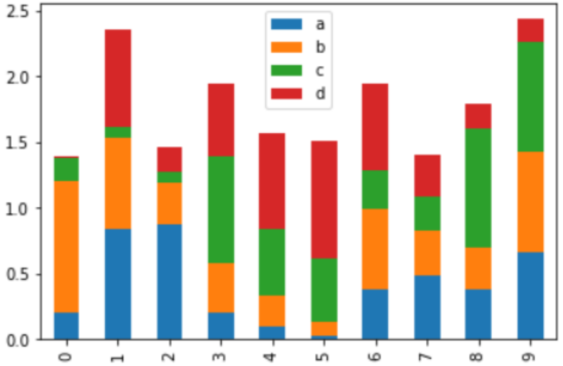
```python
df = pd.DataFrame(np.random.rand(10, 4), columns=['a', 'b', 'c', 'd'])
df
```

	a	b	c	d
0	0.208853	0.999368	0.177955	0.004178
1	0.844905	0.692018	0.074666	0.749286
2	0.871019	0.324711	0.080421	0.184144
3	0.204348	0.379936	0.809383	0.554804
4	0.097902	0.235168	0.507640	0.725698
5	0.032330	0.096933	0.492393	0.891947
6	0.380896	0.613993	0.288315	0.664387
7	0.485372	0.342866	0.256060	0.318199
8	0.380383	0.317800	0.911732	0.178845
9	0.658787	0.773167	0.835882	0.168122

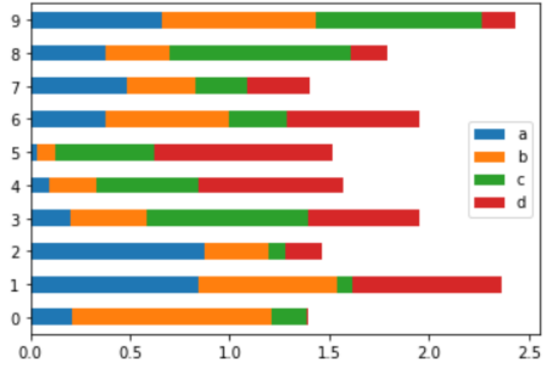
```python
df.plot.bar();
```




```python
df.plot.bar(stacked=True);
```




```python
df.plot.barh(stacked=True);
```



iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

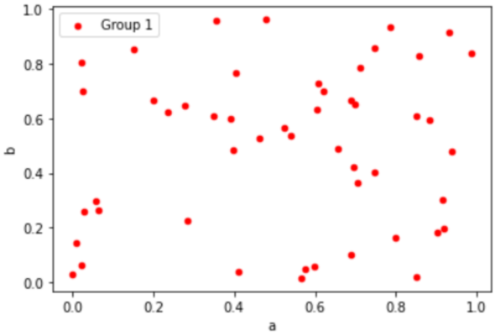
Pandas Scatter

```python
import pandas as pd
import numpy as np
```

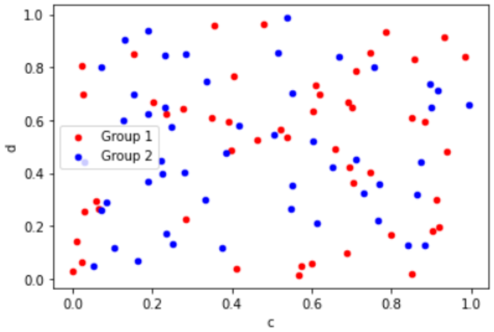
```python
df = pd.DataFrame(np.random.rand(50, 4), columns=['a', 'b', 'c', 'd'])
df.head()
```

	a	b	c	d
0	0.390322	0.597285	0.232865	0.173990
1	0.913780	0.302225	0.051365	0.048679
2	0.882417	0.593372	0.332704	0.300124
3	0.057654	0.296669	0.728719	0.323799
4	0.745395	0.857029	0.188868	0.366672

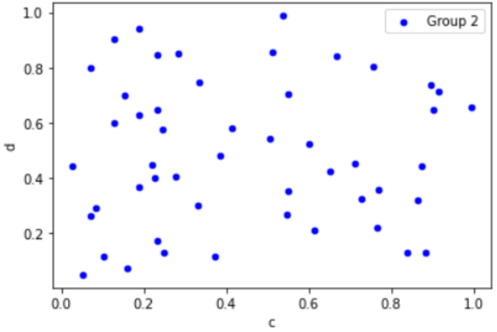
```python
df.plot.scatter(x='a', y='b', color='red', label='Group 1');
```




```python
k=df.plot.scatter(x='a', y='b', color='red', label='Group 1')
df.plot.scatter(x='c', y='d', color='blue', label='Group 2',ax=k);
```




```python
df.plot.scatter(x='c', y='d', color='blue', label='Group 2');
```



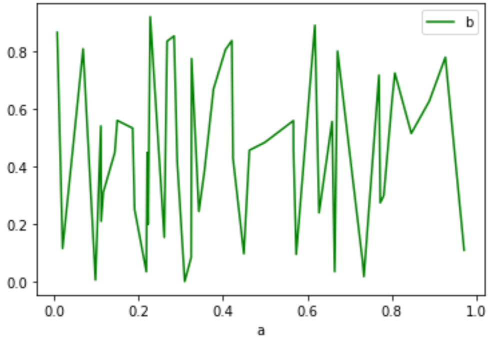
Pandas
Plot (kind = …)

```
import numpy as np
import pandas as pd
```

```
df = pd.DataFrame(np.random.rand(50, 3), columns=['a', 'b', 'c'])
df['c']*=200
df.sort_values(by = 'a', inplace=True)
df.head(7)
```

	a	b	c
10	0.007162	0.867517	195.458007
9	0.019932	0.116770	39.280149
23	0.068505	0.810289	91.971103
15	0.097792	0.006978	145.376167
36	0.110587	0.542228	106.221738
18	0.111375	0.211138	161.600937
31	0.116303	0.311410	13.561107

```
df.plot(kind='line', x= 'a', y='b' , color='green');
```




```
kind : str
    The kind of plot to produce:

    - 'line' : line plot (default)
    - 'bar' : vertical bar plot
    - 'barh' : horizontal bar plot
    - 'hist' : histogram
    - 'box' : boxplot
    - 'kde' : Kernel Density Estimation plot
    - 'density' : same as 'kde'
    - 'area' : area plot
    - 'pie' : pie plot
    - 'scatter' : scatter plot
    - 'hexbin' : hexbin plot.
```

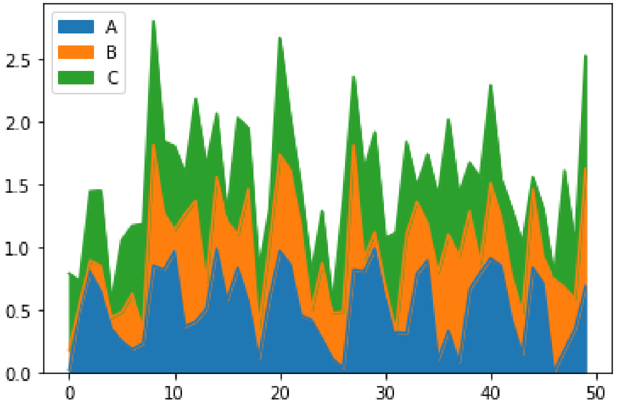
Pandas
Area

```
import numpy as np
import pandas as pd
```

```
df = pd.DataFrame(np.random.rand(50, 3),columns=['A', 'B', 'C'])
df.head(7)
```

	A	B	C
0	0.393983	0.831335	0.934014
1	0.891985	0.953771	0.052773
2	0.081741	0.386546	0.766863
3	0.739023	0.967765	0.043960
4	0.072500	0.872009	0.205749
5	0.713897	0.390277	0.075670
6	0.849453	0.796737	0.862341

```
df.plot(kind='area');
```



Pandas Scatter style

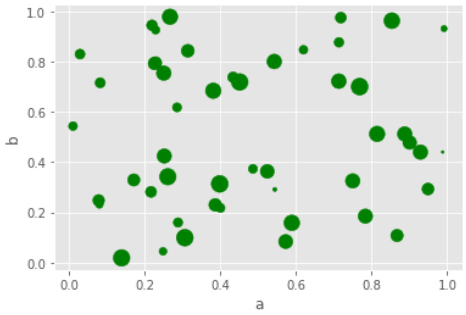
```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
plt.style.use('ggplot')
df = pd.DataFrame(np.random.rand(50, 3), columns=['a', 'b', 'c'])
df['c']*=200
df.head()
```

	a	b	c
0	0.750055	0.325310	141.280214
1	0.125275	0.028598	6.777477
2	0.713286	0.722529	147.216246
3	0.768355	0.700825	192.918051
4	0.783541	0.184801	139.875161

```
df.plot(kind='scatter', x='a', y='b', s='c' , color='green');
```




```
plt.style.available
```

```
['Solarize_Light2',
 '_classic_test_patch',
 'bmh',
 'classic',
 'dark_background',
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'seaborn',
 'seaborn-bright',
 'seaborn-colorblind',
 'seaborn-dark',
 'seaborn-dark-palette',
 'seaborn-darkgrid',
 'seaborn-deep',
 'seaborn-muted',
 'seaborn-notebook',
 'seaborn-paper',
 'seaborn-pastel',
 'seaborn-poster',
 'seaborn-talk',
 'seaborn-ticks',
 'seaborn-white',
 'seaborn-whitegrid',
 'tableau-colorblind10']
```

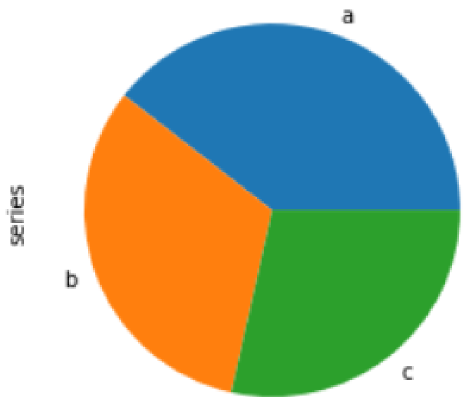
Pandas Series Pie

```
import pandas as pd
import numpy as np
```

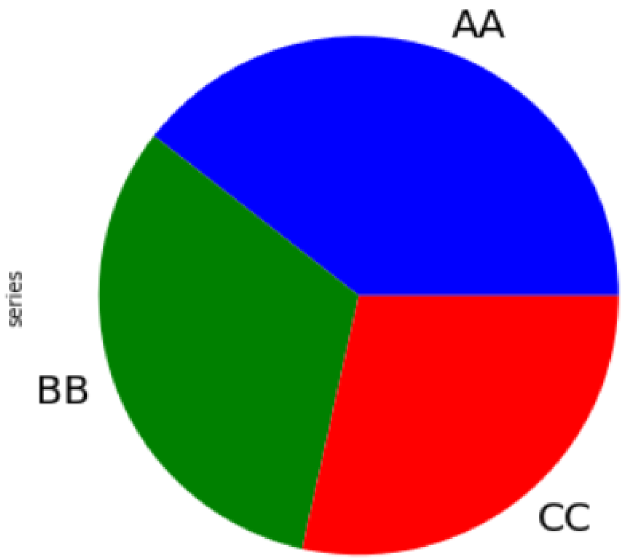
```
s = pd.Series(3 * np.random.rand(3),
              index=['a', 'b', 'c'],
              name='series')
s
```

```
a    1.635387
b    1.329396
c    1.181540
Name: series, dtype: float64
```

```
s.plot.pie();
```




```
s.plot.pie(labels=['AA', 'BB', 'CC'],
           colors=['b', 'g', 'r'],
           fontsize=20,
           figsize=(6, 6));
```



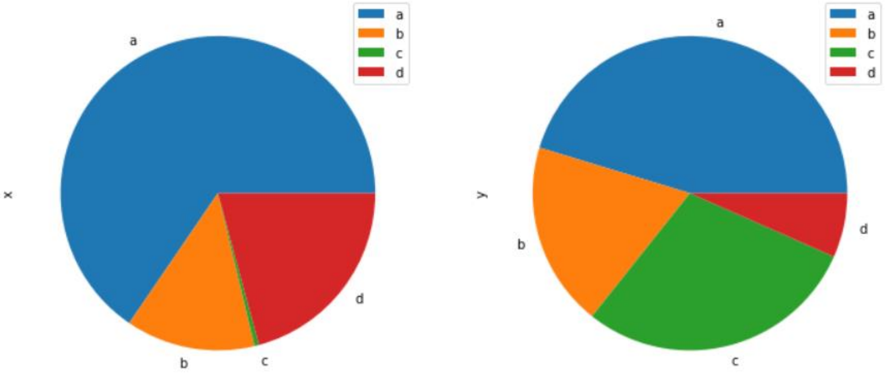
iscte INSTITUTO UNIVERSITÁRIO DE LISBOA emprego digital

Pandas DataFrames
Pie

```
df = pd.DataFrame(3 * np.random.rand(4, 2),
                  index=['a', 'b', 'c', 'd'],
                  columns=['x', 'y'])
df
```

	x	y
a	2.280157	2.398253
b	0.462583	1.005063
c	0.015842	1.538840
d	0.724033	0.350281

```
df.plot.pie(subplots=True, figsize=(12, 8));
```



Meshgrid

The numpy.meshgrid function is used to create a rectangular grid out of two given one-dimensional arrays representing the Cartesian indexing or Matrix indexing.

```
import numpy as np
```

```
x = np.array([1,2,3,4])
y = np.array([7,8])
```

```
a,b = np.meshgrid(x, y)
```

```
a
```

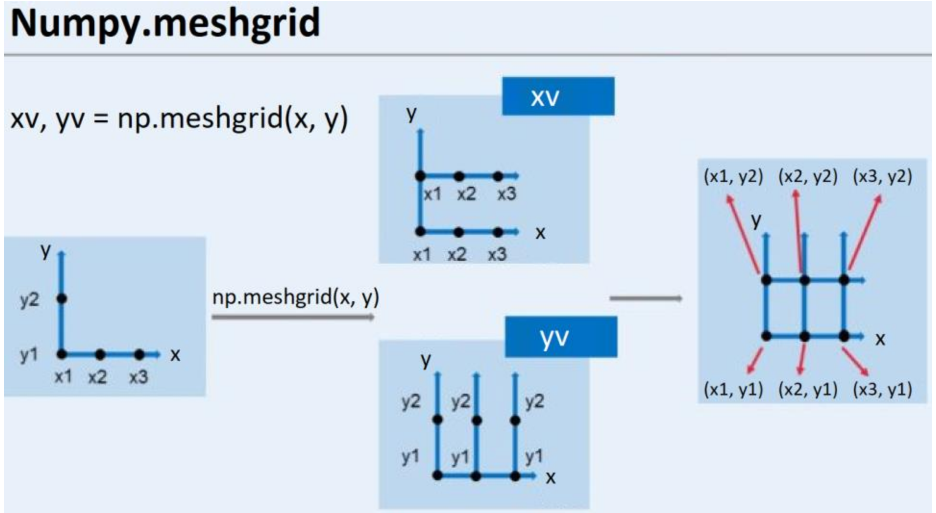
```
array([[1, 2, 3, 4],
       [1, 2, 3, 4]])
```

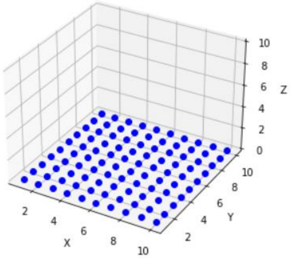
```
b
```

```
array([[7, 7, 7, 7],
       [8, 8, 8, 8]])
```

```
a.shape
```

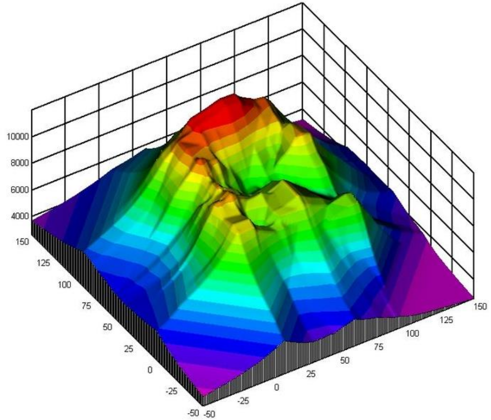
```
(2, 4)
```

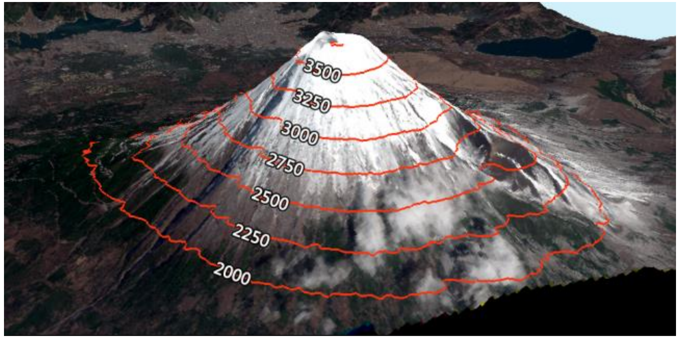


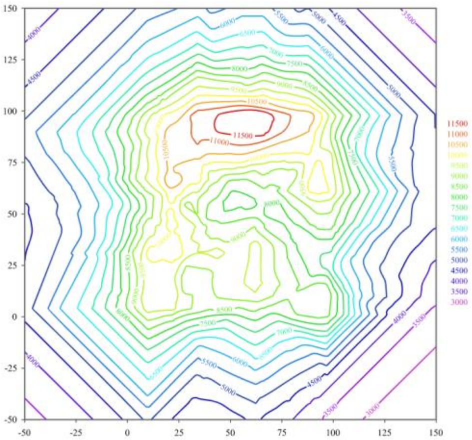


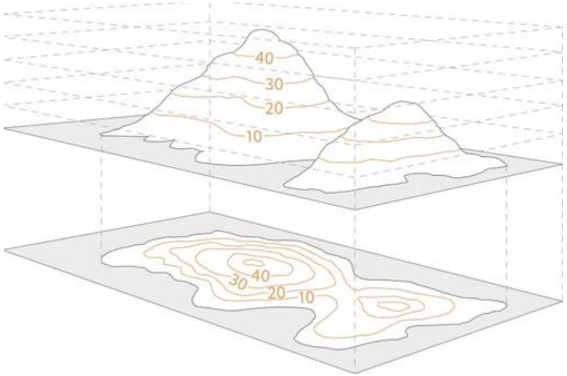
Contour?

What is the Contour? (TOPOGRAPHIC MAPS)









3D Plot

mpl_toolkits.mplot3d
Plot_surface
Contour

Contour plots (sometimes called Level Plots) are a way to show a three-dimensional surface on a two-dimensional plane.

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

```
x = np.arange(-10, 10, 0.5)
y = np.arange(-10, 10, 0.5)
X, Y = np.meshgrid(x, y)
Z = X ** 2 + Y **2
```

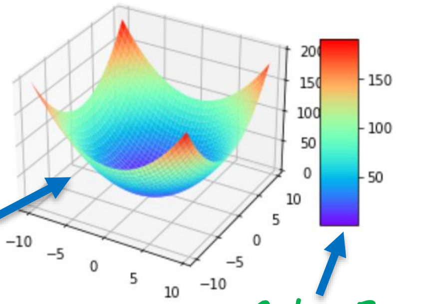
Create an empty 3D figure

Plot 3D surface

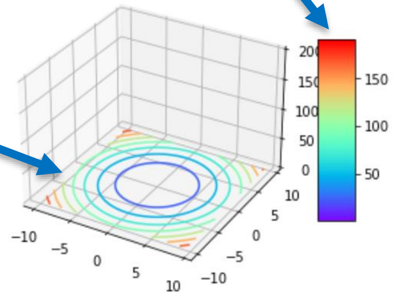
Contour

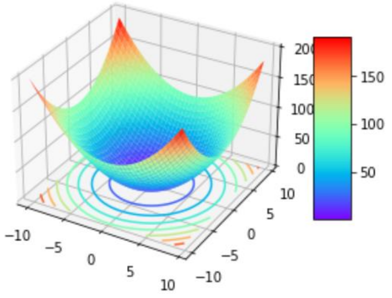
Color bar

```
fig = plt.figure(figsize=(5, 5))
ax = fig.gca(projection='3d')

s = ax.plot_surface(X, Y, Z, cmap=plt.cm.rainbow)

cset = ax.contour(X, Y, Z, zdir='z', offset=0, cmap=plt.cm.rainbow)

fig.colorbar(s, shrink=0.5, aspect=5);
```

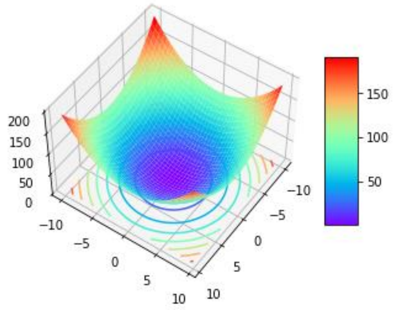


Color Bar






```
ax.view_init(50, 35)
fig
```

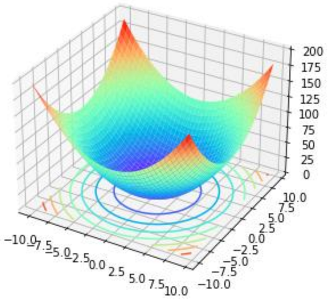


iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

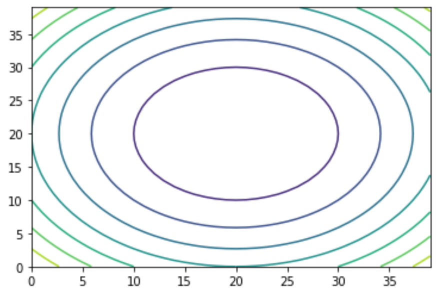
emprego digital

Contour
Contourf

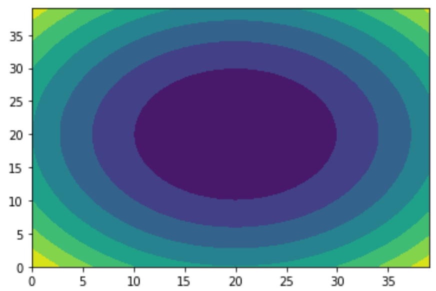
contour and contourf draw contour lines and filled contours, respectively. Except as noted, function signatures and return values are the same for both versions.



```
plt.contour(Z);
```



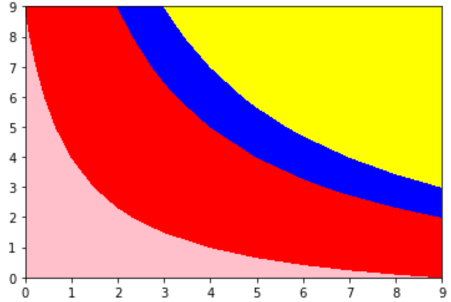
```
plt.contourf(Z);
```



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

```
x = np.arange(1, 11)
y = x.reshape(-1, 1)
h = x*y
```

```
cs = plt.contourf(h, levels=[10, 30, 40], colors=['r', 'b'], extend='both')
cs.cmap.set_over('yellow')
cs.cmap.set_under('pink')
cs.changed()
```



“

- *Make it work*
- *Make it Right*
- *Make it Fast*

O futuro profissional começa aqui

iscte
INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

UPskill