Achal Gupta

A00258772



Python Packages

NumPy, SciPy, Matplotlib

Table Of COntents

[Introduction 2](#_Toc530772373)

[Pycharm 2](#_Toc530772374)

[Features of Pycharm 2](#_Toc530772375)

[Installing Packages 3](#_Toc530772376)

[Importing Packages 4](#_Toc530772377)

[NumPy – Numerical Python 4](#_Toc530772378)

[Properties of NumPy 4](#_Toc530772379)

[Creating Arrays 5](#_Toc530772380)

[One Dimensional Array 5](#_Toc530772381)

[Arrays with Zeros 5](#_Toc530772382)

[Arrays with ones 6](#_Toc530772383)

[Creating 1D arrays – Arrange 6](#_Toc530772384)

[Creating 2D arrays – resahape 7](#_Toc530772385)

[Creating 3D arrays 7](#_Toc530772386)

[Identifying dimensions of array – ndim 8](#_Toc530772387)

[Identifying shape of the array – shape 9](#_Toc530772388)

[Identifying Size of the array – size 9](#_Toc530772389)

[Basic Operations 10](#_Toc530772390)

[Slicing 12](#_Toc530772391)

[Boolean Arrays 12](#_Toc530772392)

[Universal Mathematical Functions – floor, sqrt 13](#_Toc530772393)

[SciPy – Scientific Python 14](#_Toc530772394)

[Real World Scientific Domains 14](#_Toc530772395)

[Why do we need SciPy? 14](#_Toc530772396)

[Linear Algebra 15](#_Toc530772397)

[Statistics 16](#_Toc530772398)

[Constants 17](#_Toc530772399)

[Matplotlib 19](#_Toc530772400)

[Benefits of Data Visualization 19](#_Toc530772401)

[Advantages of Matplotlib 19](#_Toc530772402)

[Steps to create a Plot 20](#_Toc530772403)

[Some important plots. 21](#_Toc530772404)

[Real World example – Video Games Data 28](#_Toc530772405)

[Data Source – 28](#_Toc530772406)

[Importing data and defining arrays 29](#_Toc530772407)

[Conclusion 30](#_Toc530772408)

# Introduction

Python is a high level general purpose programming language created by Guido van Rossum and first released in 1991. Python is based on the philosophy that focus on code reliability, using appropriate indentation. Python features a dynamic type system and automatic memory management system. It supports various programming approaches like object-oriented programming, imperative programming, functional and procedural. Python is an interpreter-based language and is available for many operating systems. Both Python and CPython, the reference implementation of Python is managed by a non-profit organization Python Software Foundation (Wikipedia, 2018).

# PyCharm

PyCharm is the IDE (Integrated Development Environment) used for programming in Python. It is developed by a Czech company Jetbeans. PyCharm is a cross-platform tool supporting windows, macOS and Linux. The community edition of the software is released under Apache License while there is a Professional version with additional features which is under a proprietary license (Wikipedia, 2018).

# Features of PyCharm



(jetbeans, 2018)

# Installing Packages

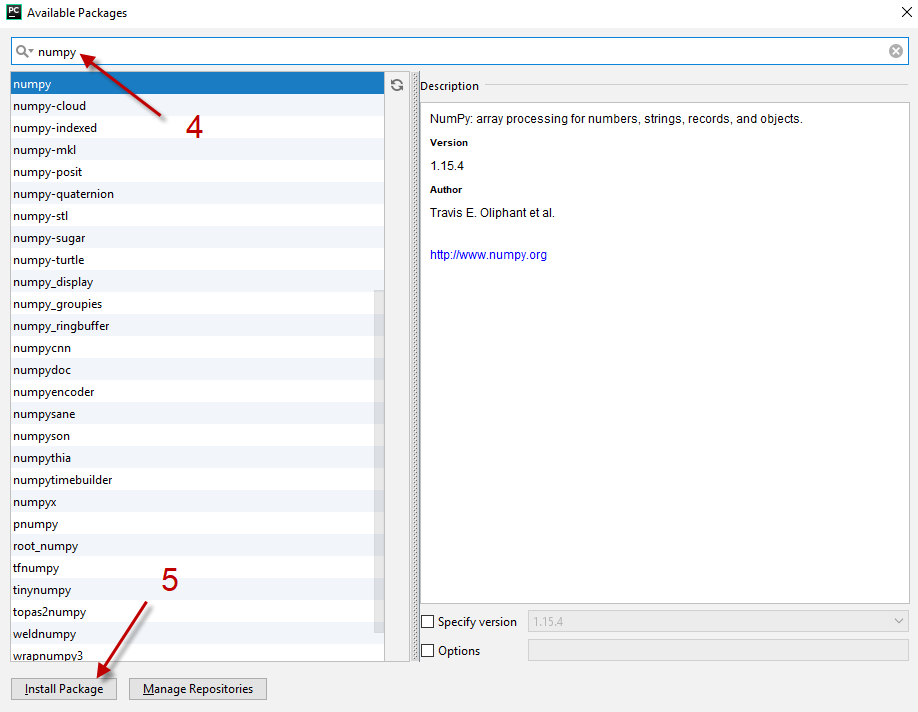
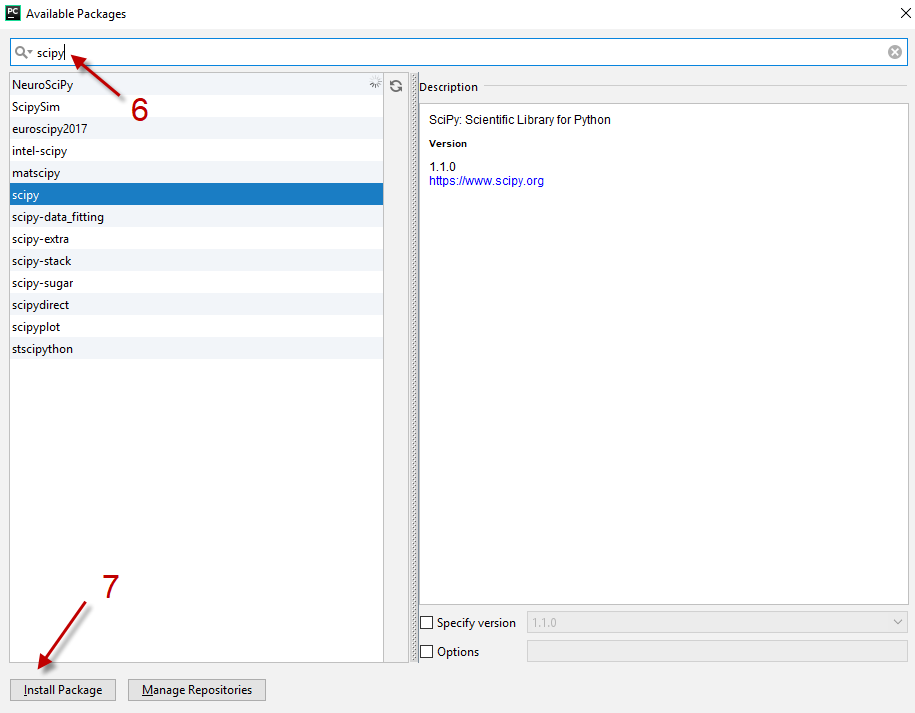
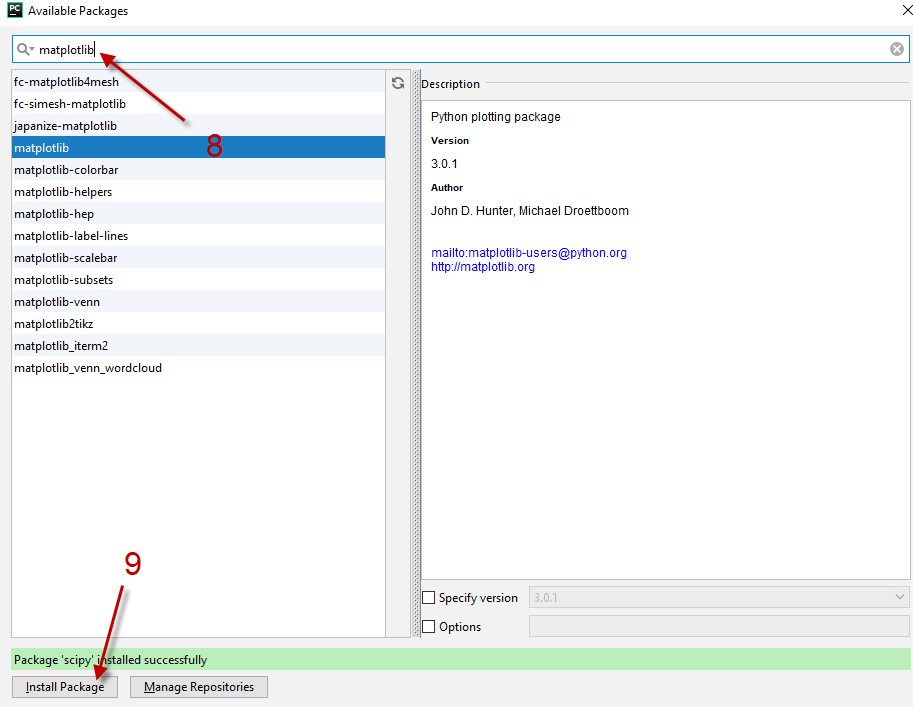
**Step 1:** Go to File -> Settings from the menu.

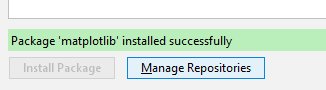
**Step 2:** Expand the Project tab and Go to Project Interpreter

**Step 3:** Click on ‘+’ Button to install various packages into the project

**Step 4:**

Search for various libraries like NumPy, SciPy and matplotlib and click on Install Package for each of the package as shown in figures below.





Importing Modules

# Importing Packages

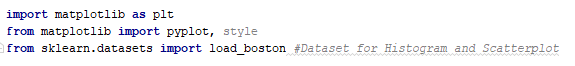
After installing packages, we need to import packages into the python before we can go ahead and use various functionalities. To import packages in python we use “import” command. Below command will import all the sub packages in NumPy library and an alias of np will be given.



Below command will import only three sub package of SciPy library i.e. linalg, stats and constants



Below command will import matplotlib package and its sub package pyplot and style, which will be used to create graphs. Also, we are importing a free dataset called “load\_boston” to generate some meaningful graphs.

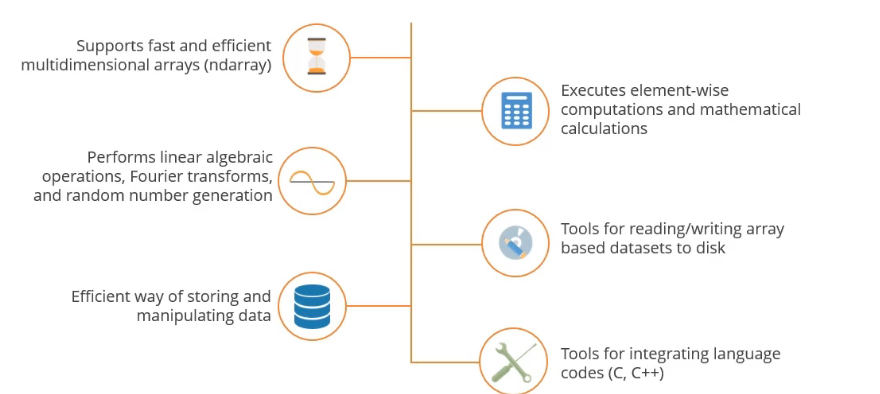


# NumPy – Numerical Python

Numerical Python is the foundational package for mathematical computing in Python.

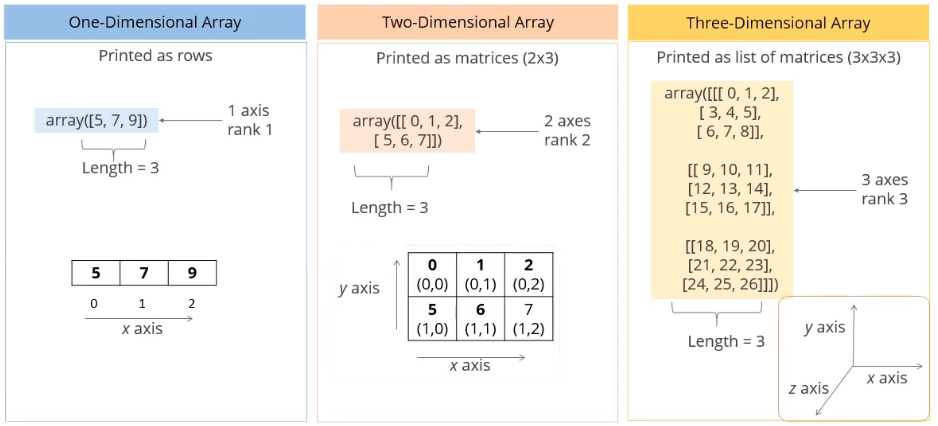
NumPy is used to create arrays and perform various operations on arrays. Mathematical functions cannot be applied on lists and hence arrays are used for this purpose.

## Properties of NumPy



(Simplilearn, 2018)

Arrays act as primary container for the data. Arrays can me one dimensional, two dimensional or even multidimensional.



(Simplilearn, 2018)

## Creating Arrays

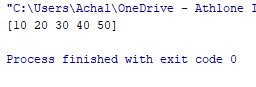
Below code explores various types of arrays in Python.

## One Dimensional Array

Code: This will create and print a one dimensional array.



Output: Displays a one dimensional array

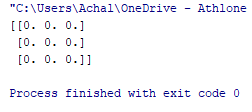


## Arrays with Zeros

Code: This will create a two dimensional array and fill 0 as the values for each element



Output: Displays a 2-D array with 0 values

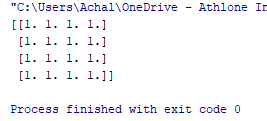


## Arrays with ones

Code: This will create an array and fill 1 in values for each element



Output: Displays the arrays with 1 as value for all elements

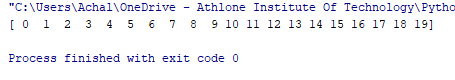


## Creating 1D arrays – Arrange

Code: arrange function will create a 1-D array with argument as number of elements. E.g. in below code 20 will create elements from 0 to 19

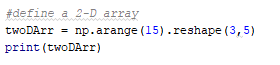


Output: Displays the array created using arrange function.

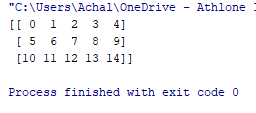


## Creating 2D arrays – resahape

Code: First we will create a 1-D array using arrange and reshape function is used to transform 1-D array into 2-D array



Output: Displays the 2-D array with 3 rows and 5 columns



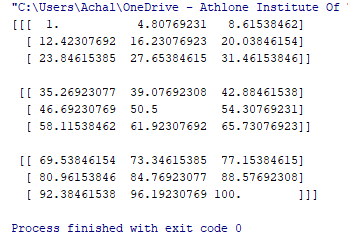
## Creating 3D arrays

Code: Linepace function will create a 1-D array values starting from value of first argument to the value of second argument. Third argument tells the total number of elements e.g. in code below a 1-D array will be create starting from 1 to 100 and having total 27 values.

After creating 1-D array we will reshape the array into a 3-D array with 3X3X3 format.



Output: Displays a 3-D array created using above code.

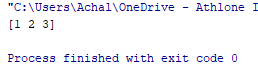


## Identifying dimensions of array – ndim

Code: ndim function will display the number of dimension of the array. Below code will create an array value of dimensions of all the three arrays define in previous code i.e. oneDArr, twoDArr, threeDArr

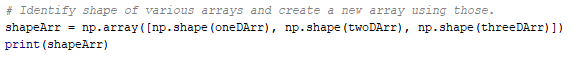


Output: Displays the array of value of number of dimensions of three arrays, we can see that first array has a dimension of 1, second has a dimension of 2 and third has a dimension of 3



## Identifying shape of the array – shape

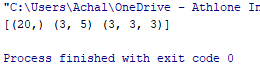
Code: shape function will display the shape of the array. Below code will create a list of values of the shape of arrays for the three arrays defined in previous code.



Output: Displays the shape of three arrays. 1-D array is displayed as (20,) which signifies only one dimension of 20 elements.

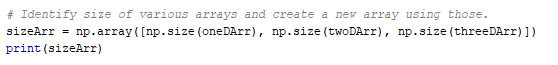
2-D array is displayed as (3,5) which means array shape is like a matrix with 3 rows and 5 columns

3-D array is displayed as (3,3,3) which signifies presence of three dimensions in the array with total 27 elements.

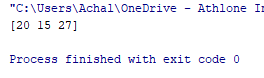


## Identifying Size of the array – size

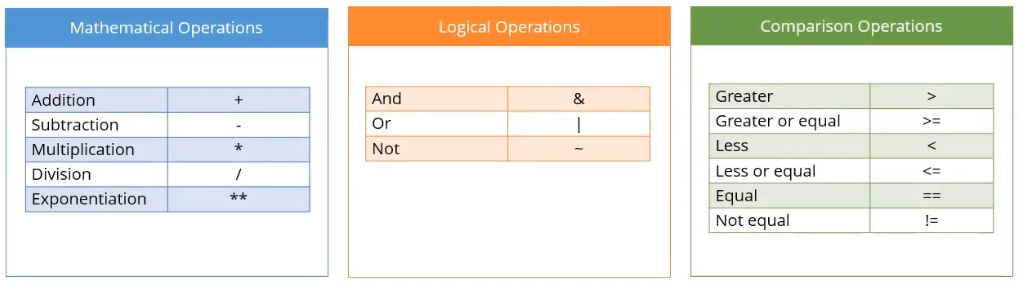
Code: size function displays the size or number of elements in the array. Below code will create an array of value of number of elements for the three arrays defined in previous code.



Output: Displays the array of size of three arrays.



## Basic Operations



(Simplilearn, 2018)

### Mathematical Multiplication

Given below is the hourly rates for all days of a week Mr. A worked. Also given number of hours he worked each day in that week.

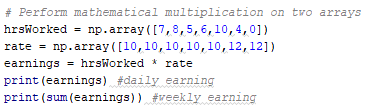
Weekday Hourly rate – 10 Euros / hour

Weekend Hourly rate – 12 Euros / hour

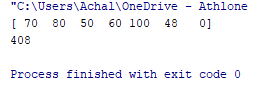
Hours worked per day in on week (Mon to Sun) – [ 7, 8, 5, 6, 10, 4, 0]

Calculate daily and weekly earnings of Mr. A?

Code: Defines two arrays with number of hours worked per day in a week and rate per hour of the day. Daily Earning per day is calculated by multiplying both arrays. And weekly earning is calculated by adding up all daily earnings.



Output: Displays array with 7 days of daily earning and weekly earnings of 408 Euros.



Array [70, 80, 50, 60, 100, 48, 0] displays daily earnings in Euros for Mr. A and 408 Euros is the weekly earning of Mr. A.

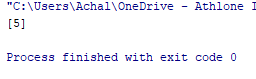
### Comparison Operations

Identify day number where Mr. A worked more than 8 hours in the week?

Code: Define an array with values 1 to 7 to identify day of the week. Then we will print weekday with hours worked greater than 8



Output: Displays the week day where number of hours worked is greater than 8



On day number 5 or Friday Mr. A worked for more than 8 hours.

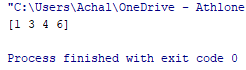
### Logical Operations

Identify day number where Mr. A worked between 0 to 4 hours.

Code: Print the wee days where hours worked is between 0 and 8 hrs



Output: Displays an array with week day where hours worked is between 0 and 8

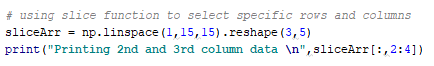


Hence on day numbers 1,3,5 and 6 i.e. Monday, Wednesday, Thursday, Saturday Mr. A worked between 0 and 8 hours.

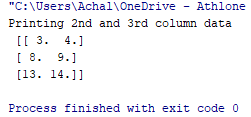
## Slicing

Slicing is the method of selecting specific rows.

Code: Define a 2D array with 3 rows and 5 columns an element ranging from 1 to 14. Then select all rows with 2 and 3rd column.



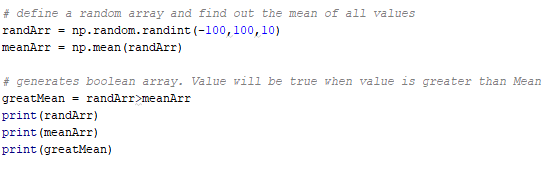
Output: Displays 2nd and 3rd column with all rows of the 2D array



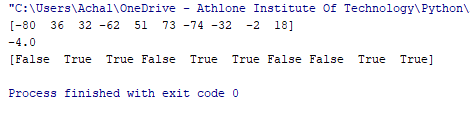
## Boolean Arrays

Boolean array is the array with Boolean data type values.

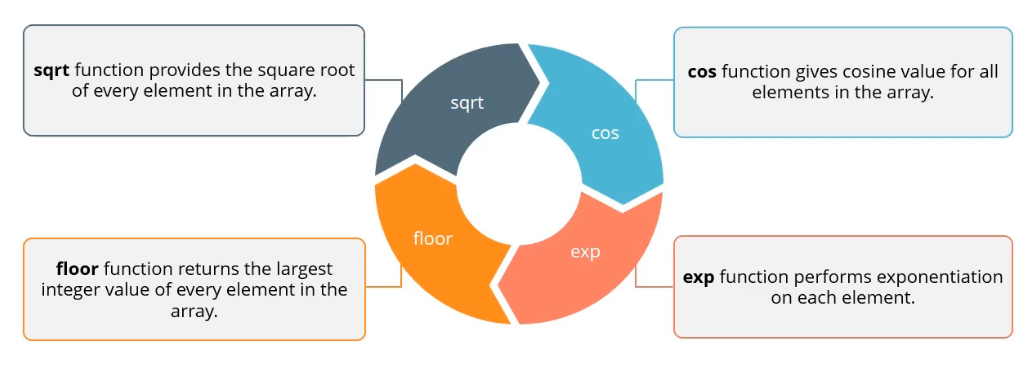
Code: Defines a random array and then generate the mean of all values of the array. Defines a Boolean array which is TRUE when value is greater than mean.



Output: Displays the generated random array, mean of the array and the Boolean array where TRUE is the value greater than mean.



## Universal Mathematical Functions – floor, sqrt

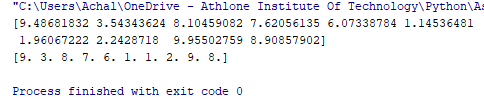


(Simplilearn, 2018)

Code: Defines a random array of 10 elements and then apply floor function on all values of the array. Floor function returns the nearest smallest integer value.



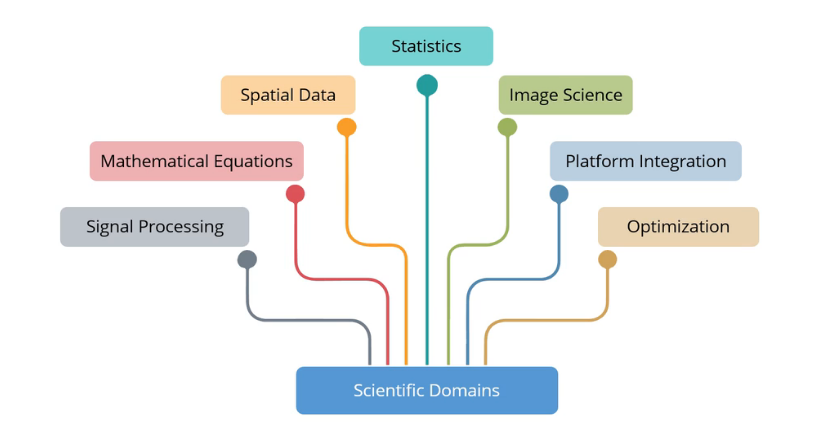
Output: Displays both random array and the values when floor function is applied.



# SciPy – Scientific Python

Extension of NumPy for advanced scientific and mathematical algorithms and functions.

## Real World Scientific Domains



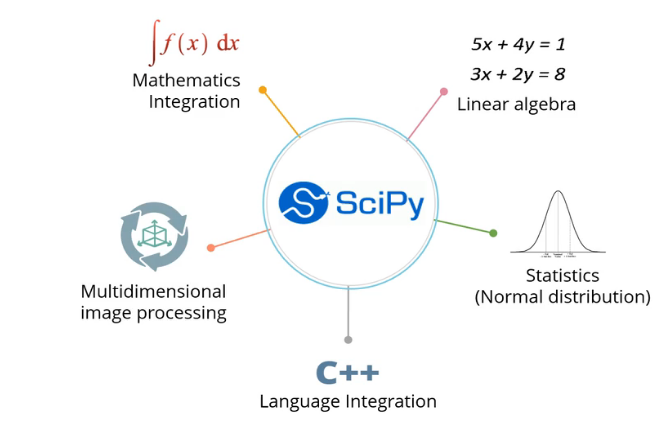
(Simplilearn, 2018)

* Statistics – Domain build hypo and predict outcome
* Spatial Data – Spatial Data Structures – Spatial objects – points, lines, rectangles, volumes, time
  + Rivers, crops, county, countries
  + Retrieve records based on spatial properties
* Image Science – This domain covers image processing – understand image properties and extract values from there.
* Mathematical Equations – Bigger or Complex problems like Differentiation and Integration#
* Platform Integration – reuse code written in various coding platforms like C/C++
* Signal Processing – Process Analog/Digital Signals – Read signals from electronic sensors
* Optimization – chose right value from set of given values – improve overall performance of the process

## Why do we need SciPy?

SciPy deals with above mentioned scientific domains, it has inbuilt packages for:

* Mathematical Integration – fftpack, integrate
* Linear Algebra - linalg
* Multidimensional Image Processing - ndimage
* Statistics - stats
* Language Integration – weave,

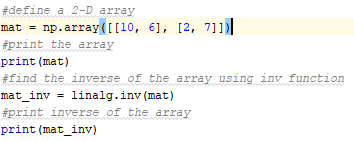


(Simplilearn, 2018)

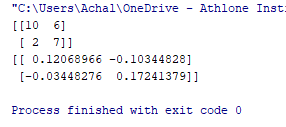
## Linear Algebra

### Inverse of Matrix – Matrix must be a square matrix

Code: Defines a random array and calculate its inverse using inv function of SciPy library.



Output: Diplayus both the array and its inverse



### Solving Linear Equation

2x + 3y + z = 21

-x + 5y + 4z = 9

3x + 2y + 9z = 6

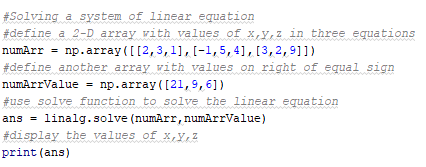
Solving above equation will give following result

x = 4.95

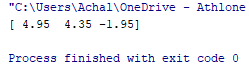
y = 4.35

z = -1.95

Code: Defines two arrays based on the constant coefficients of the multiple linear equation. Then uses solve function to generate the values of x,y and z.



Output: Displays value of x, y and z as retrieved from solve function.



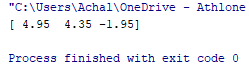
## Statistics

### Random Variables

Code: Defines a random variates with mean of 0 and standard deviation of 1.



Output: Displays the array generated using rvs function.

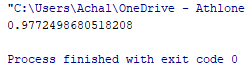


### Cumulative Distribution Function (CDF)

Code: Defines an array of normally distributed values and calculate the cumulative density of the array using cumulative distribution function or cdf.



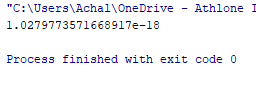
Output: Displays the cumulative frequency of the array.



### Probability Distribution Function (PDF)

Code: Defines an array of normally distributed values and calculate the probability density of the array using probability distribution function or pdf. 

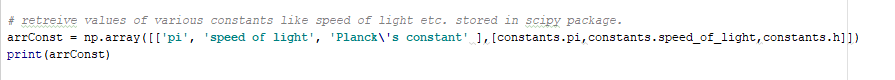
Output: Displays the probability density of the array.



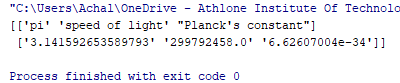
## Constants

SciPy package stores values of various constants used in scientific calculations. We can retrieve their values and use them in various calculations

Code: Defines an array with value pi, speed of light and Planck's constant as retrieved from constants sub package of SciPy



Output: Displays the 2D array with name and value of various constants.



# Matplotlib

Data Visualization – Technique to represent data in a pictorial or graphical format. To analyse data visually, identify patterns and trends easily

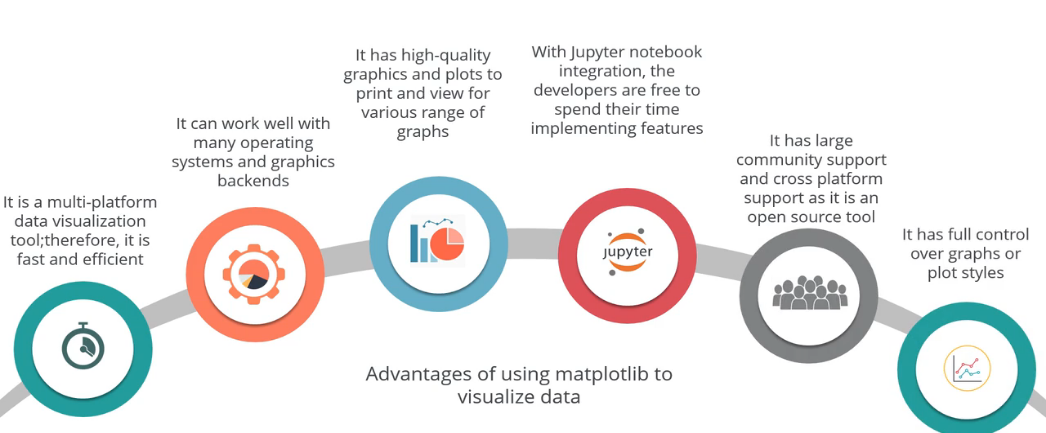
## Benefits of Data Visualization



(Simplilearn, 2018)

* Easily explore new patterns and reveals hidden patterns using visually appealing charts and figures.
* Simplifies the complex quantitative information using tables and other visuals.
* Identify relationships between data points and variables by visualizing clusters.
* Analyse and explore big data
* Identify areas of improvement easily using visuals and drilldown and drill through functionalities.

## Advantages of Matplotlib



(Simplilearn, 2018)

* Multi-platform data visualization tool built on NumPy and SciPy packages and hence is very fast and efficient.
* Can work in many operating systems and graphic backends
* High quality graphics and plots to print and view various graphs
* Large community support
* Full control over graphs or plot styles

## Steps to create a Plot



(Simplilearn, 2018)

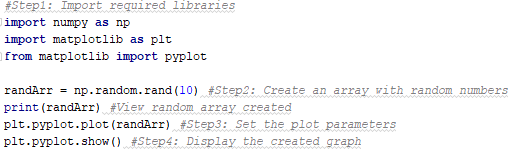
Step 1: Import the required libraries using import keyword

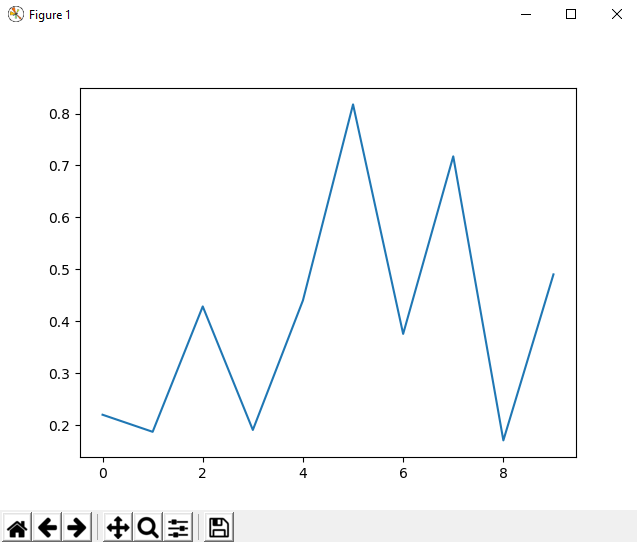
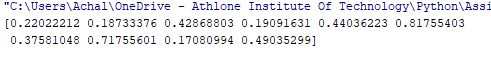
Step 2: Import required dataset

Step 3: Set various plot parameters like X axis values, Y-axis values, X label, Y label, title etc.

Step 4: Show the graph

Code: Defines a random array and plot it on the line chart.



Output: Displays the generated random array and corresponding graph.

## Some important plots.

### Multiple Line Charts

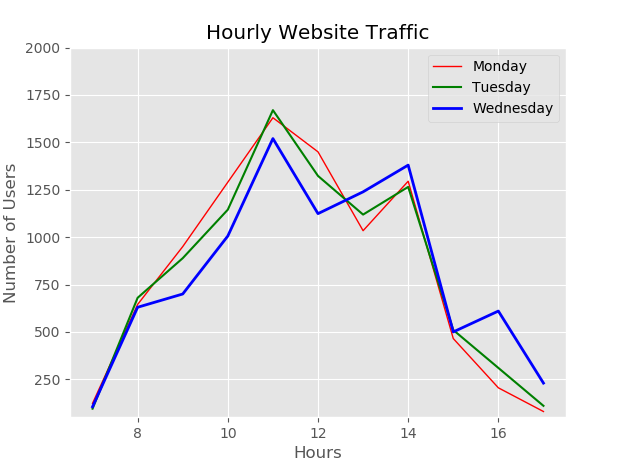
No. of users visiting a website per hour for the busiest 10 hours of three days was collected.

Below code defines three arrays with website traffic data of 10 hours for Monday, Tuesday and Wednesday

Code: Using the three arrays a multiple line chart is plotted with various prameters.



Output: Displays the Hourly website traffic for three days.



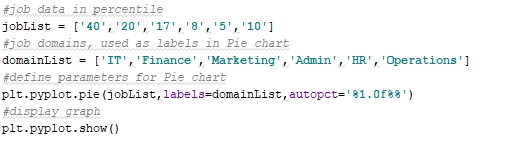
From the graph we can easily visualize that there is a surge in visitors around noon on all three days. Thus if a promotion need to be displayed on the website it will be a good to time to display the add as the traffic is maximum during noon.

### Pie Chart

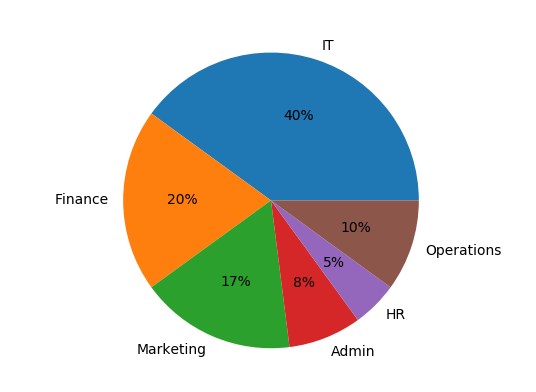
Pie chart is used to visualize relative percent of each category of data from the overall percentage.

Below data shows the percentile of jobs in various sectors like IT, HR, Finance etc.

Code: Defines an array of percentiles and name of sectors and then plot it on pie chart.



Output: Displays the pie chart with percentage of each sector.



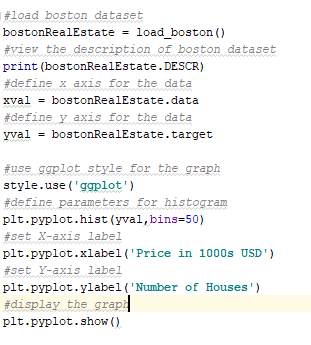
We can easily see that out of overall job openings 40% belongs to IT sector, while HR sector contributes only 5% of overall job openings. Finance is second hottest sector after IT with 20% of jobs followed by Marketing (17%) and Operations (10%). Admin jobs comprise of only 8% of the overall jobs.

### Histogram

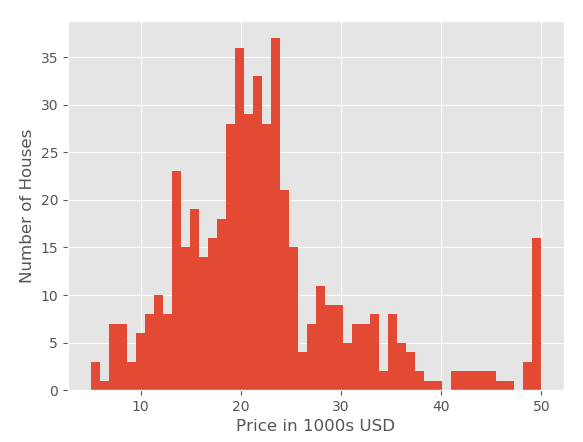
A histogram is the representation of distribution of data. In histogram we define number of bins and values are counted for each bin. Each bar represents the count of values belonging to a bin.

We are using the freely available Boston dataset which we imported in previous steps. Using the histogram we will identify the distribution of prices of houses in Boston.

Code: Loads the boston dataset, describe it and the plot Price of houses on an histogram.



Output: Displays the histogram where x axis is the Price in 1000s USD.



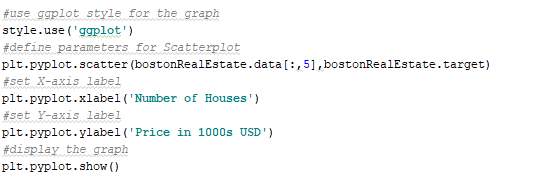
From the above graph we can visualize that majority of Houses in Boston have a price ranging from 18000 USD to 25000 USD.

### Scatterplot

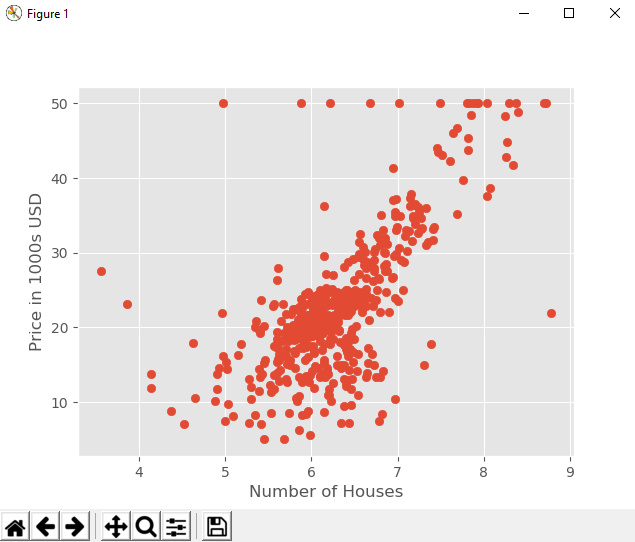
Scatter plot is the plot of values of two variables on cartesian coordinates. Scatterplots are used to identify the relation between two variables and identify any outliers or exceptions to the general trend.

Again, we will be using the Boston data set and plot prices of house vs number of houses on a scatterplot.

Code: Plots a scatter plot of Number of houses and prices in 1000s USD of Boston.



Output: Displays the scatterplot of Number of houses and prices in 1000s USD of Boston.



The observation from Histogram is confirmed using scatter plot as we can see that most of the houses are concentrated between 18000 USD to 250000 USD price range.

# Real World example – Video Games Data

## Data Source –

This dataset covers a list of games having sales of more than 1,00,000 copies.

Below are the details of various columns

* Rank - Rank of sales
* Name - The name of the game
* Platform – Platform for which the game was released
* Year - Year of release of game
* Genre - Genre of the game
* Publisher - Publisher details of the game
* NA\_Sales - Sales in North American region (in millions)
* EU\_Sales - Sales in European region (in millions)
* JP\_Sales - Sales in Japan (in millions)
* Other\_Sales - Sales in the rest of the world (in millions)
* Global\_Sales - Total worldwide sales (in millions).

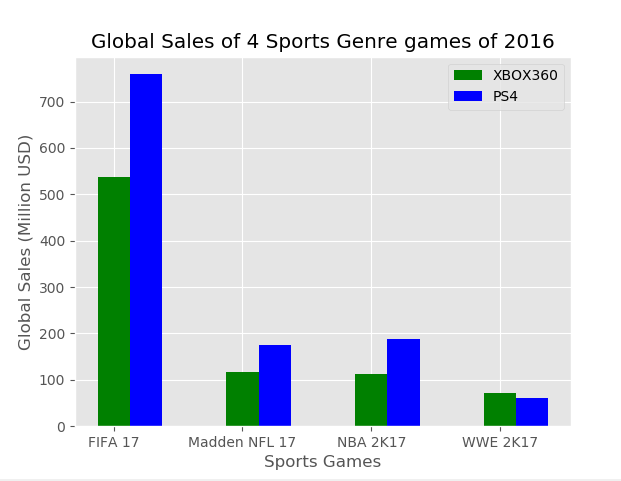
<https://www.kaggle.com/gregorut/videogamesales>

We will compare the global sales of four major Sports Genre games released in 2016 on two platforms XBOX 360 and PS4.

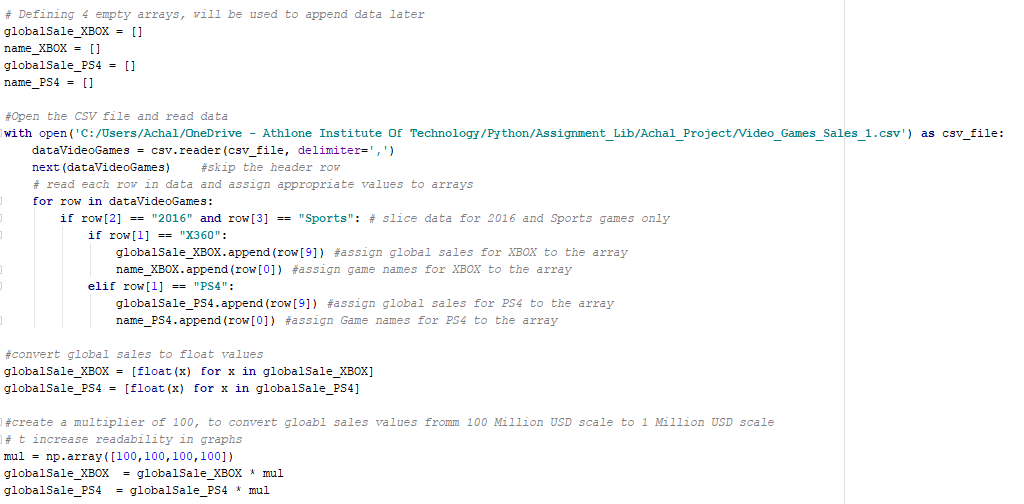
To analyse the above we will use multiple bar graphs to visualize and compare the data.

First, let's have a look at the output. From the below graph we can easily visualize following observations

* Overall sales of sports games of PS4 is much higher than XBOX 360 globally.
* For FIFA 17 there is a very huge difference in sales between XBOX 360 and PS4, hence we can assume that gameplay experience of FIFA 17 is higher in PS4 when it comes to FIFA 17.
* Only WWE 2k17 sales are higher for XBOX 360 than PS4, which can be due to various factors like better gameplay experience of XBOX for WWE or may be due to the launch of special promotions and offer by Microsoft for WWE.



## Importing data and defining arrays



First part of the code defines 4 empty arrays which will be utilized to store various values of Global sales for both XBOX and PS4.

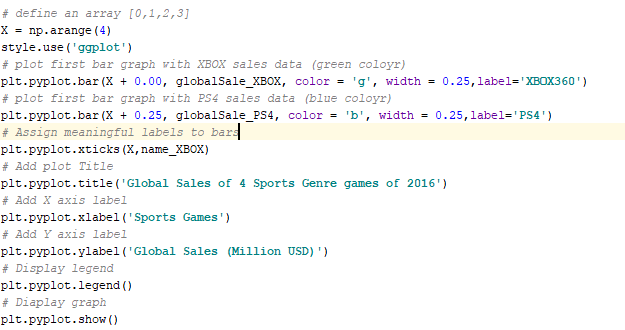
Next step is to read csv file and store the data of csv in a dataset called dataVideGames.

Then for each row of dataset we will store the values of Game name and Global Sale of only 2016, Sports Genre games. We will store values for XBOX and PS4 in two separate arrays.

Since the values read from a csv file are all integer values, we need to convert Global Sales value for both XBOX and PS4 into float values.

Also, to visualize better we will change the scale of global sales from 100 Million USD to 1 Million USD by multiplying values in both global sales array by 100.

Now that we are ready with the arrays we will plot them on the graph.



First, we will define a reference array for 4 Sports games (X array) as we cannot plot string values (Game names) on X axis. We will later assign appropriate labels to each reference point in the graph.

Global sales of XBOX is first plotted in green colour and then bar plot of PS4 Global sales is added to the figure with a width of (+0.25) on the scale.

Next using xticks function we will assign appropriate game names to the 4 reference points.

Define other parameters like Title, X axis label and Y axis label and we are good to show the graph along with the legend.

# Conclusion

The three packages (NumPy, SciPy, Matplotlib) provides a very wide range of function for performing various mathematical and scientific calculations on arrays of dataset and visualize them to find new and hidden patterns in the data. Thus, we can say that Python offers a complete toolset to access, manipulate, transform and visualize complex datasets.

# References

jetbeans. (2018). *Pycharm*. Retrieved from www.jetbeans.com: https://www.jetbrains.com/pycharm/

Simplilearn. (2018). *Data Science with Python*. Retrieved from www.simplilearn.com: https://lms.simplilearn.com/#/

Wikipedia. (2018, November 6). *PyCharm*. Retrieved from www.wikipedia.org: https://en.wikipedia.org/wiki/PyCharm

Wikipedia. (2018, November 2). *Python (programming language)*. Retrieved from www.wikipedia.org: https://en.wikipedia.org/wiki/Python\_(programming\_language)