What is Numerical Calculation?

By: Iman Khani Jazani

- *AI and Data Specialist, Business Developer | AiHum*
- *Adjunct Professor | Sharif University of Technology*
- *Senior Data Scientist, Community builder | Adin*

Last Lecture

- Short Presentations
- Review the last lecture
- Data story telling
- Visualization in Python!
- Application of Programming in the Digital Age!

Today

- Short Presentations
- Random number generation
- Sampling from distribution
- Introduction to linear algebra with Numpy
- Introduction to SciPy
- Application of Programming in the Digital Age!

Together

Send your feedback about the class whenever you want!

Contact me

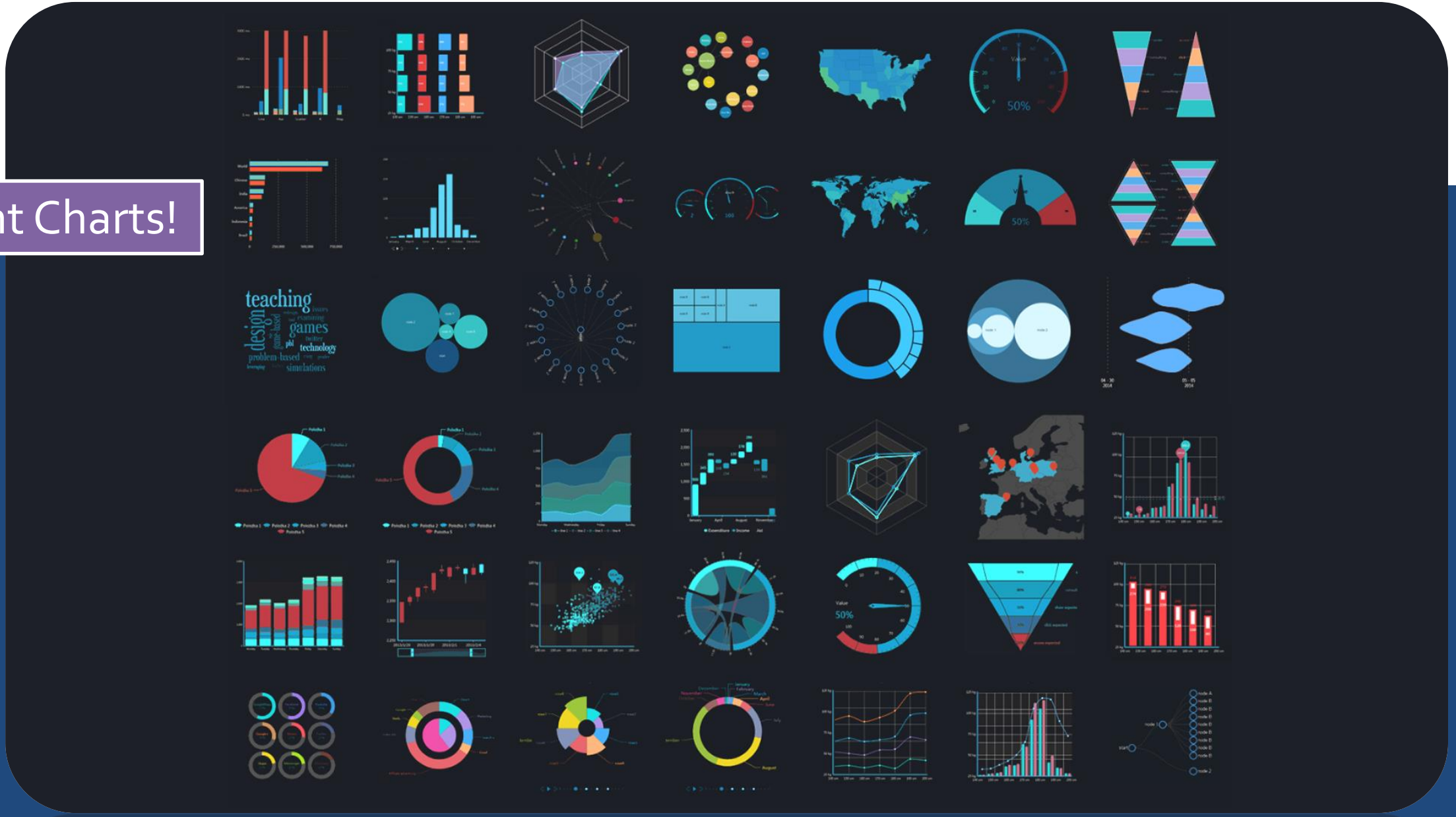
- Gmail: ImanKhaniJazani@gmail.com
- LinkedIn: https://www.linkedin.com/in/ImanKhaniJazani/
- Telegram: @IKJ1992

Short Presentations

Review the Last Lecture

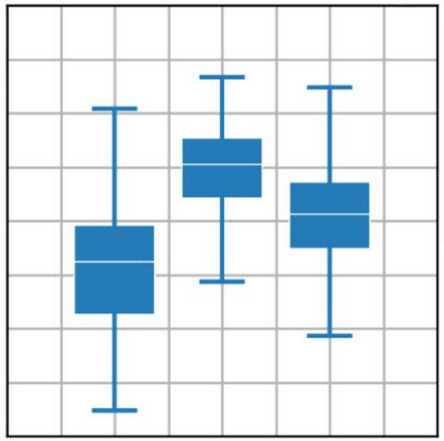
DST Procedure

Different Charts!



Let’s go to matplotlib.org!

matplotlib Plot types Examples Tutorials Reference User guide Develop Releases



boxplot(X)

Matplotlib: Visualization with Python

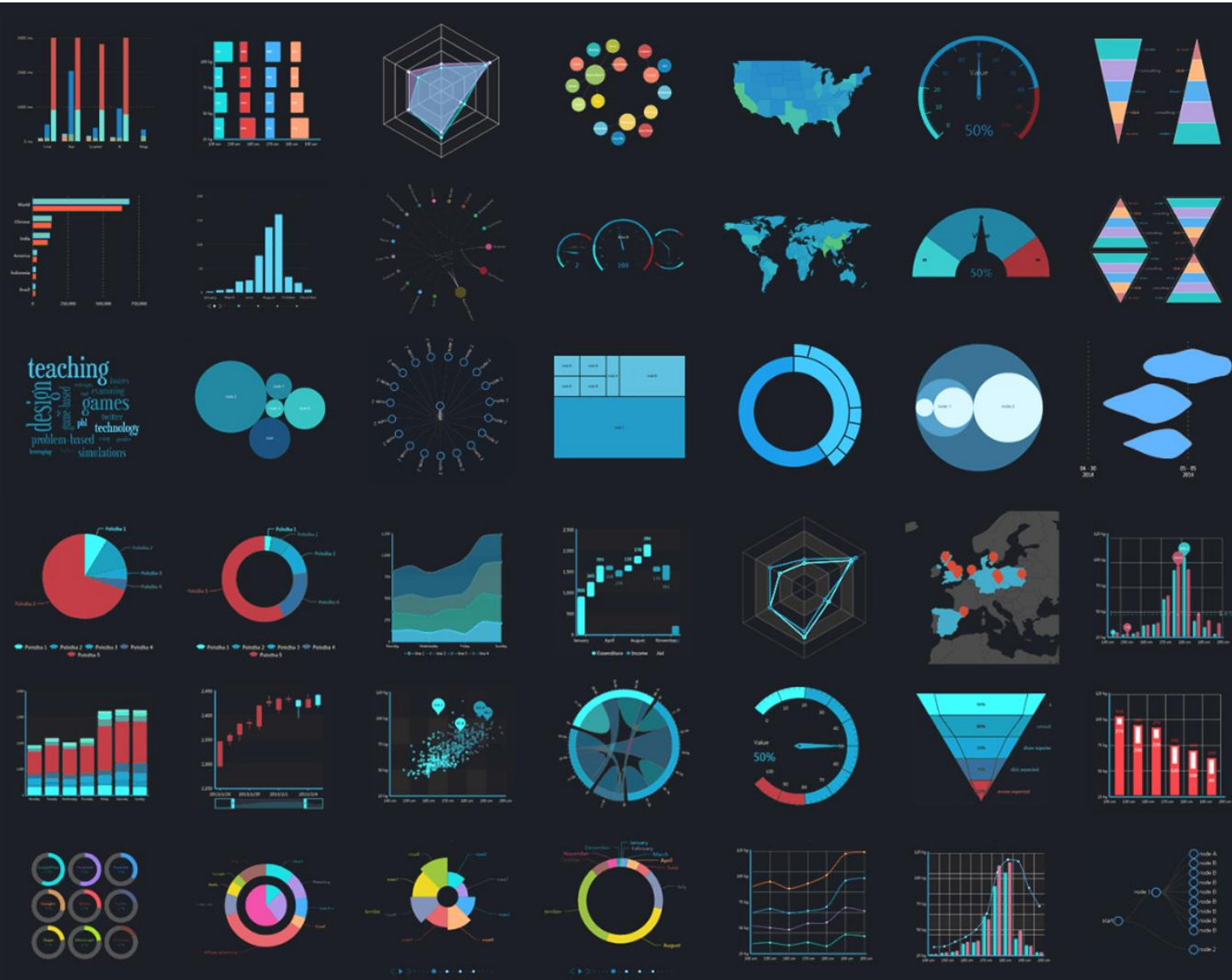
Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

Try Matplotlib (on Binder) →

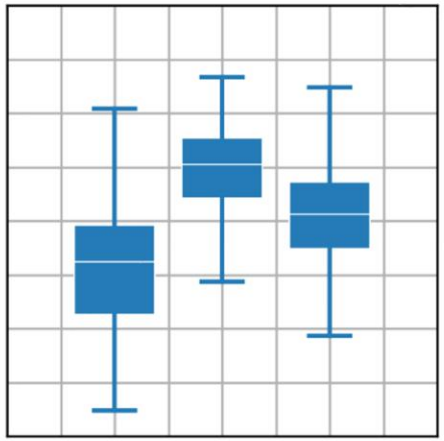
DST Procedure

Different Charts!



Let’s go to matplotlib.org!

matplotlib Plot types Examples Tutorials Reference User guide Develop Releases



boxplot(X)

Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

Try Matplotlib (on Binder) →

PyLab

LEARN MATPLOTLIB

absolute beginners

Matplotlib Tutorial

- Matplotlib - Home
- Matplotlib - Introduction
- Matplotlib - Environment Setup
- Matplotlib - Anaconda distribution
- Matplotlib - Jupyter Notebook
- Matplotlib - Pyplot API
- Matplotlib - Simple Plot
- Matplotlib - PyLab module

Matplotlib - PyLab module

Previous Page Next Page

PyLab is a procedural interface to the Matplotlib object-oriented plotting library. Matplotlib is the whole package; matplotlib.pyplot is a module in Matplotlib; and PyLab is a module that gets installed alongside Matplotlib.

PyLab is a convenience module that bulk imports matplotlib.pyplot (for plotting) and NumPy (for Mathematics and working with arrays) in a single name space. Although many examples use PyLab, it is no longer recommended.

Basic Plotting

Plotting curves is done with the plot command. It takes a pair of same-length arrays (or sequences) –

```
from numpy import *
```

Thanks to Prof. Shahshahani for his talk in the meeting about Math!

Random Number Generation

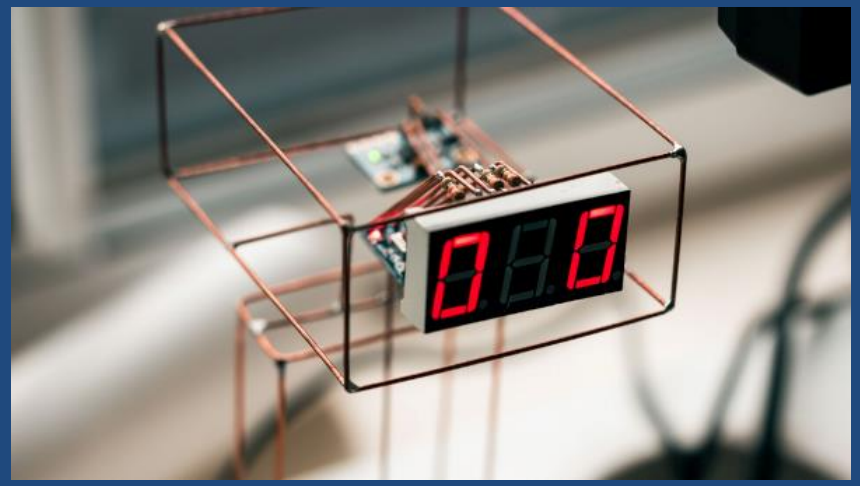
History!

- People have been using **random numbers** for millennia, so the concept isn't new. From the lottery in ancient Babylon, to roulette tables in Monte Carlo, to dice games in Vegas, the goal is to leave the end result up to random chance.

- Generating large quantities of random numbers requires great deal of time and work. Thanks to human ingenuity, we have more powerful tools and methods at our disposal.

True Random Number Generator

- A true random number generator (TRNG), also known as a hardware random number generator (HRNG), does not use a computer algorithm. Instead, it uses an external unpredictable physical variable such as radioactive decay of isotopes or airwave static to generate random numbers.



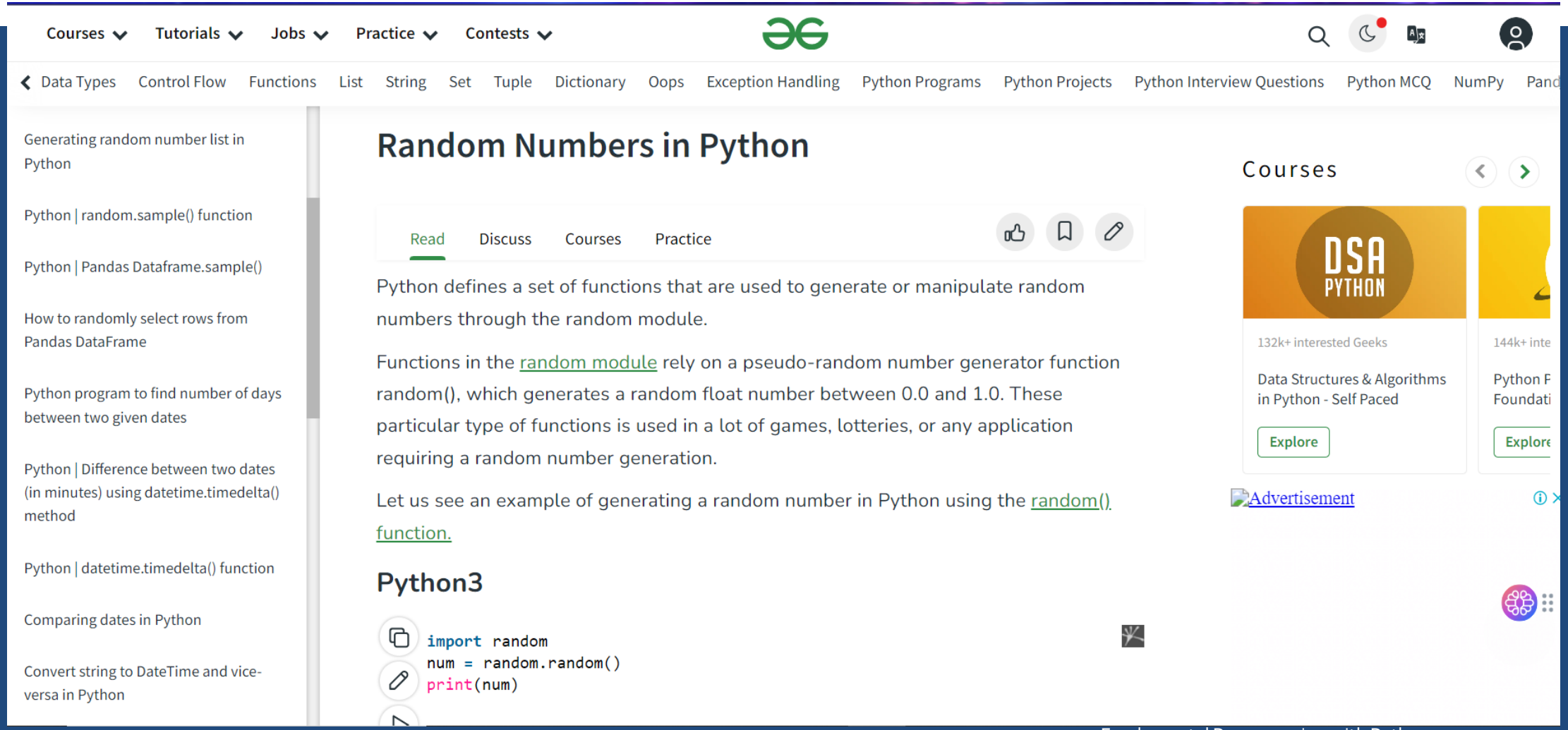
Pseudo-Random Number

- As an alternative to "true" random numbers, the **second method** of generating random numbers involves computational algorithms that can produce apparently random results.
- Why apparently random? Because the end results obtained are in fact completely determined by an initial value also known as the **seed** value or **key**. Therefore, if you knew the key value and how the algorithm works, you could reproduce these seemingly random results.

Pseudo-Random number generator algorithm

1. **Accept** some initial input number, that is a seed or key.
2. **Apply** that seed in a sequence of mathematical operations to generate the result. That result is the random number.
3. **Use** that resulting random number as the seed for the next iteration.
4. **Repeat** the process to emulate randomness.

Let’s go to Python!



Sampling from Distribution

What is distribution?

A **probability distribution** is a mathematical function that describes the probability of different possible values of a variable. Probability distributions are often depicted using graphs or probability tables.

Example: Probability distribution

We can describe the probability distribution of one coin flip using a probability table:

Outcome	Probability
Heads	Tails
.5	.5

Different types of distribution

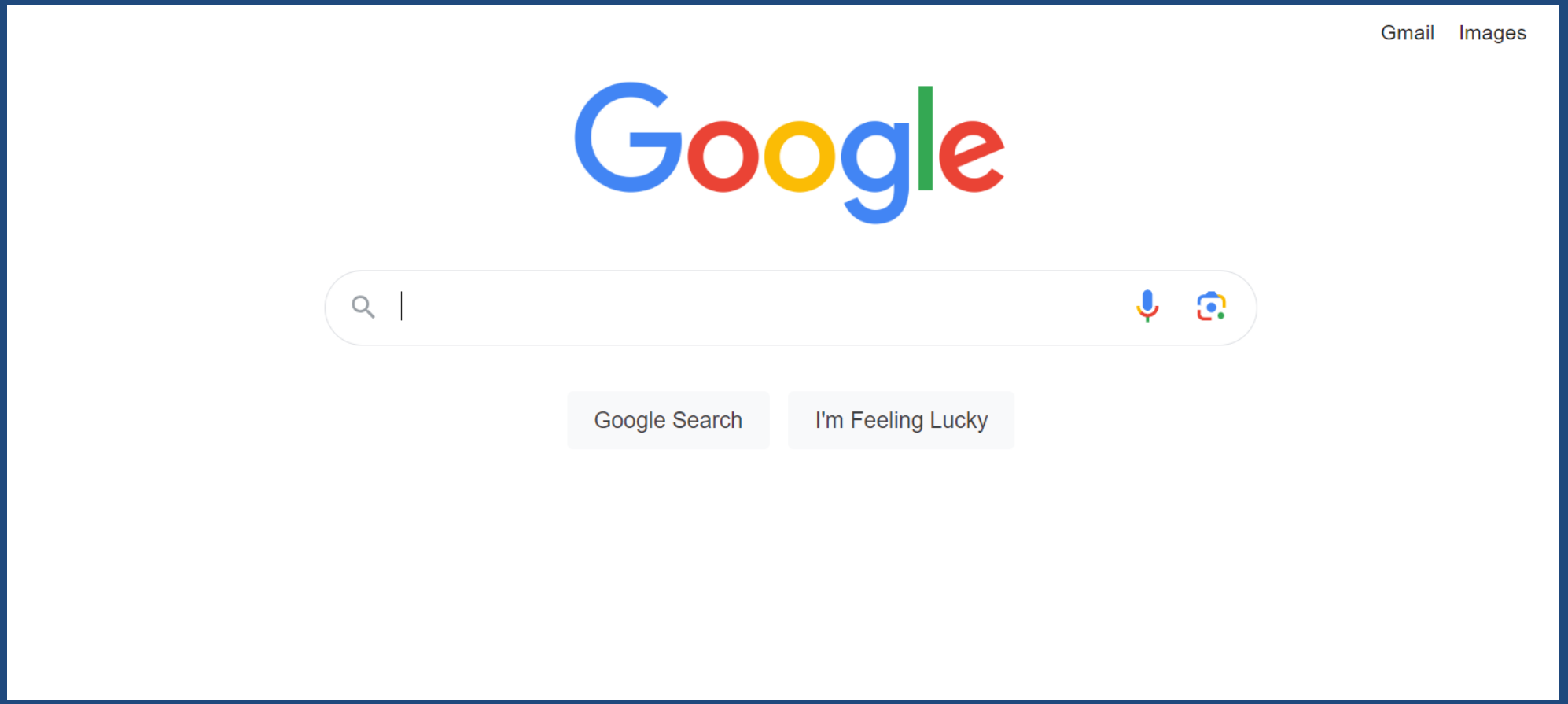
- Bernoulli Distribution
- Uniform Distribution
- Binomial Distribution
- Normal or Gaussian Distribution
- Exponential Distribution
- Poisson Distribution

What is sampling?

- A **sample** is a subset of individuals from a larger population. **Sampling** means selecting the group that you will actually collect data from in your research. For example, if you are researching the opinions of students in your university, you could survey a sample of 100 students.

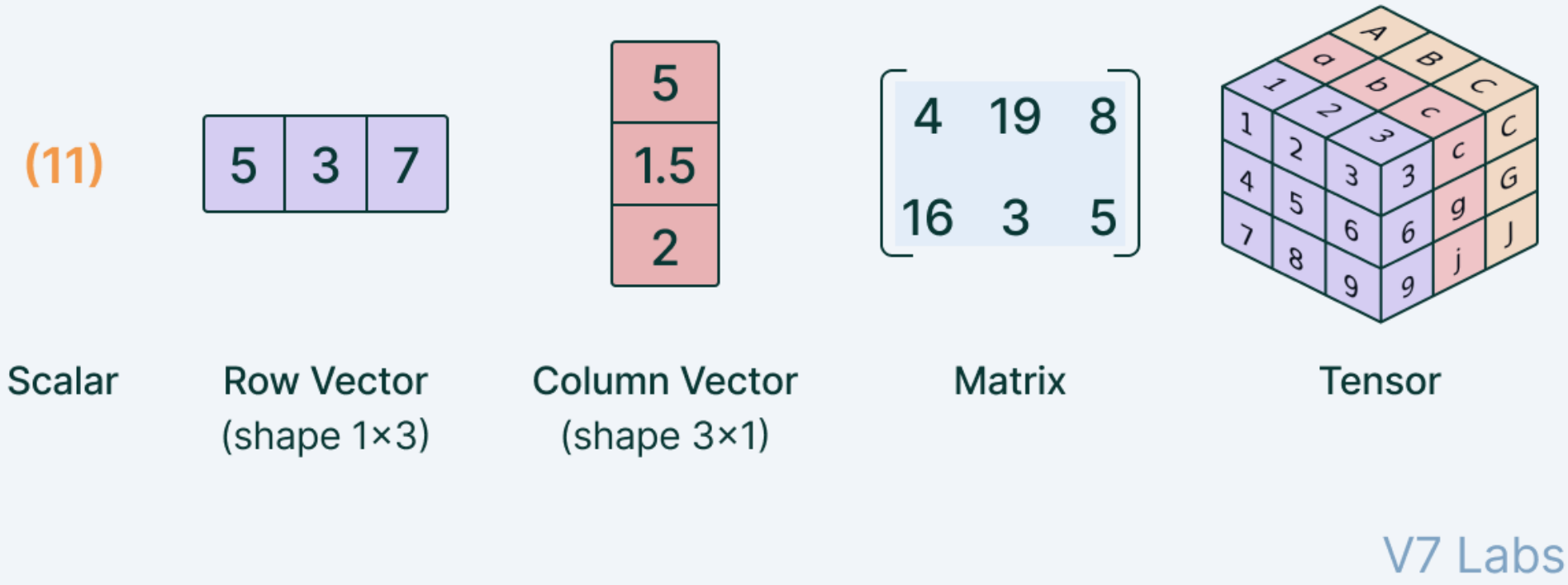
What is a sample in Bernoulli distribution?

Sampling with Python? Let’s go!



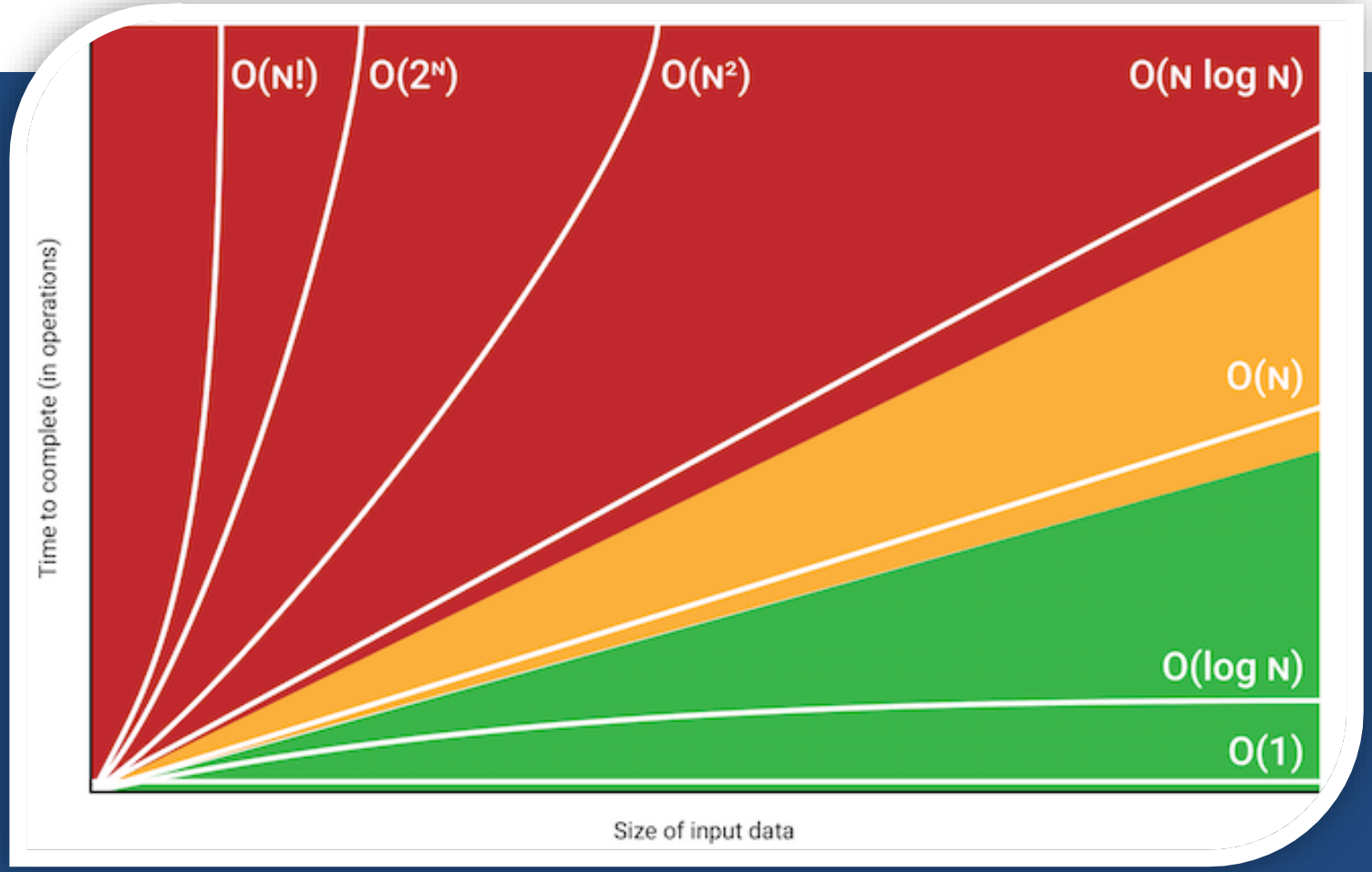
Introduction to linear algebra with Numpy

What is vector? matrix? tensor?

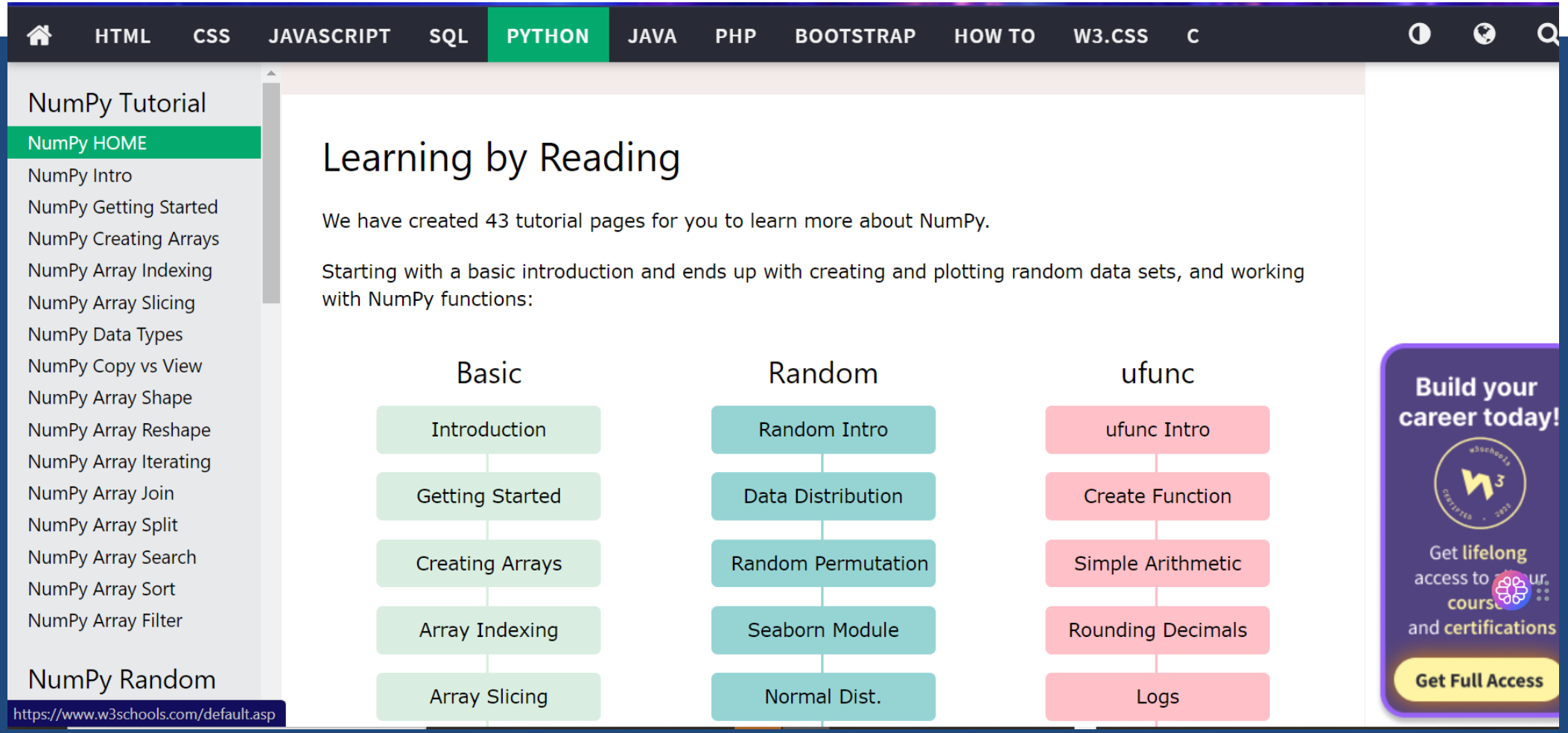


Let’s go to implement in python and time complexity!

- scalar * vector
- vector + vector
- matrix + matrix
- matrix * matrix

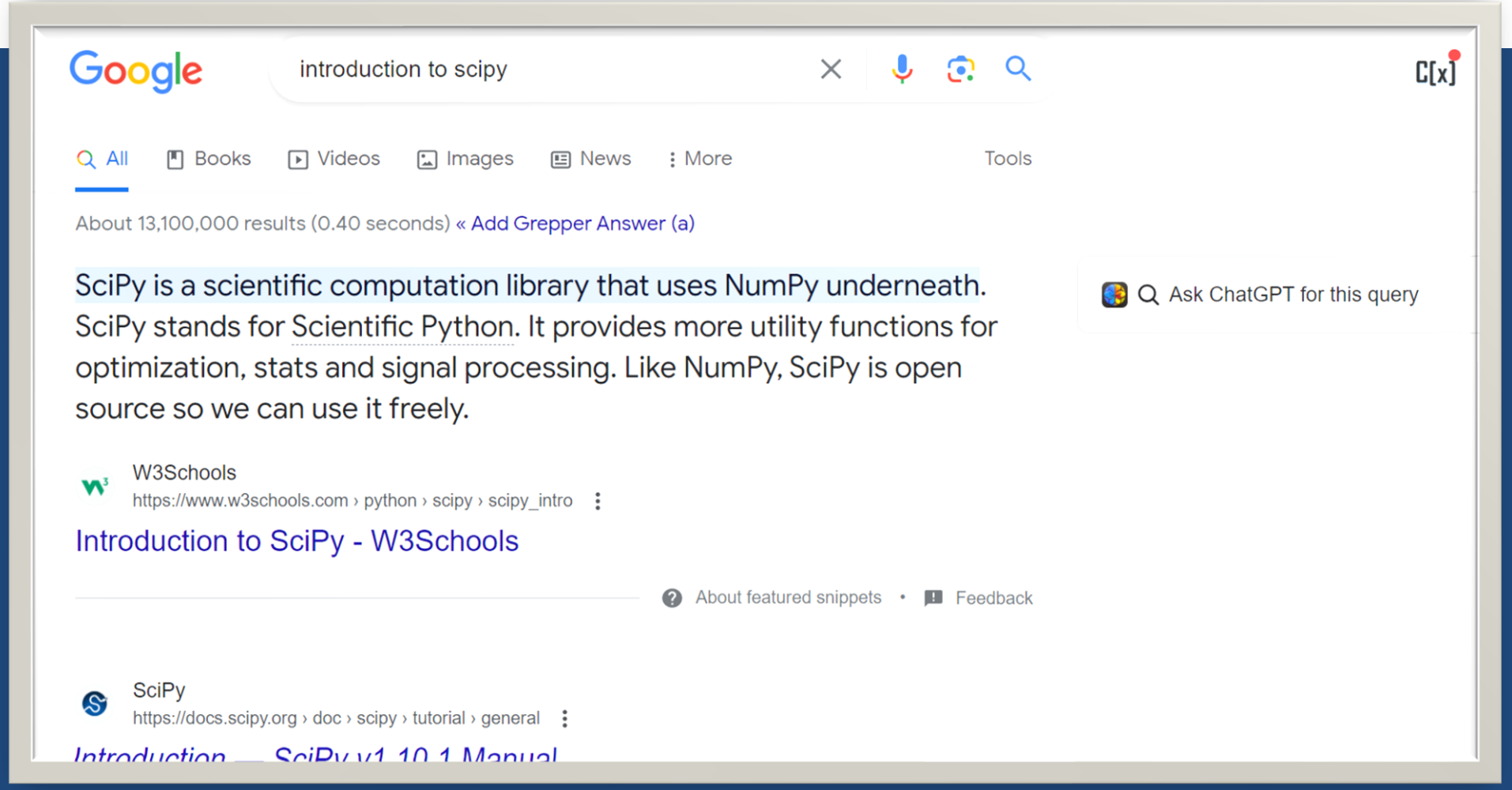


Let’s go to Numpy



Introduction to SciPy

Let's go!


Google
introduction to scipy
All
Books
Videos
Images
News
More
Tools
About 13,100,000 results (0.40 seconds) « Add Grepper Answer (a)
SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely.
Ask ChatGPT for this query
W3Schools
https://www.w3schools.com › python › scipy › scipy_intro
Introduction to SciPy - W3Schools
About featured snippets
Feedback
SciPy
https://docs.scipy.org › doc › scipy › tutorial › general

Solve linear equation

Edit & Run

```
import numpy as np
from scipy import linalg

# The linear algebra system which is given as

# 3x + 2y = 2

# x - y = 4

# 5y + z = -1

#We need to find values of x,y and z for which all these equations are zero

# Creating the input array
a = np.array([[3, 2, 0], [1, -1, 0], [0, 5, 1]])

# Providing the solution Array
b = np.array([[2], [4], [-1]])

# Solve the linear algebra
x = linalg.solve(a, b)

# Printing the result
print(x)

# Checking the result
np.dot(a, x) == b
```

What is interpolation?

Google

interpolation in python

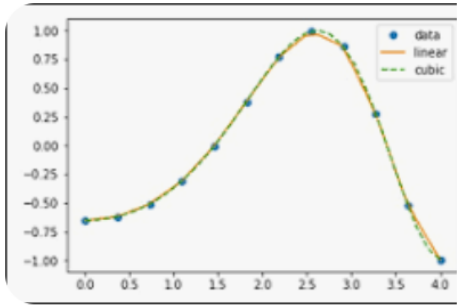
All Videos Images Books News More Tools SafeSearch on

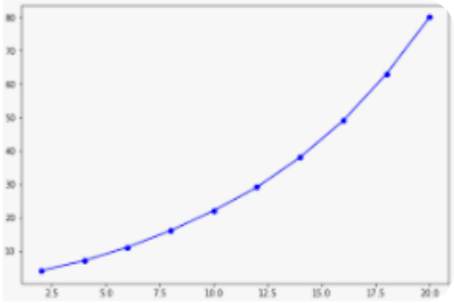
Linear Trilinear Bicubic Natural neighbor Kriging Multivariate IDW

« Add Grepper Answer (a)

Interpolation in Python is a technique used to estimate unknown data points between two known data points. In Python, Interpolation is a technique mostly used to impute missing values in the data frame or series while preprocessing data.

Khordad 11, 1400 AP





Analytics Vidhya
https://www.analyticsvidhya.com › blog › 2021/06 › po...

Interpolation Techniques Guide & Benefits | Data Analysis

About featured snippets • Feedback

Interpolation with SciPy

```
from scipy.interpolate import interp1d
import numpy as np

xs = np.arange(10)
ys = 2*xs + 1

interp_func = interp1d(xs, ys)

newarr = interp_func(np.arange(2.1, 3, 0.1))

print(newarr)
```

Application of Programming in the Digital Age!

Is it ethical!?

Monitoring

Lecture Resources

- https://www.guru99.com/scipy-tutorial.html
- https://www.geeksforgeeks.org/numpy-tutorial/
- https://www.w3schools.com/python/numpy/default.asp