

# Data Science Tools and Python Programming

Pertemuan 2

Pengajar:

Dian Ade Kurnia, M.Kom  
Rusnanda Farhan  
Rusnandi Fikri  
Rika Sahriana

# Tujuan

Membahas Data science tools dengan menjelaskan perangkat dan teknik yang berkaitan dengan keterampilan dasar dalam ilmu computer, matematika, dan statistik untuk melakukan tugas-tugas yang umumnya terkait dengan data science.

# Pada topik ini akan mempelajari

01

## Introduction

Pengenalan Tools Data  
Science

02

## Python

Bahasa Pemrograman  
Python

03

## IDE

Integrated Development  
Environment

04

## Python Library

Dasar-Dasar library  
Python untuk proyek  
data science

# Data Science Tools



# Mengapa Python



- Bahasa pemrograman tingkat tinggi
- Penulisan kode/sintaks lebih sederhana dan tersedia banyak library
- Bersifat *open-source* dan *cross-platform*
- Diluncurkan oleh Guido Van Rosum pada tahun 1991.

## Data Professional

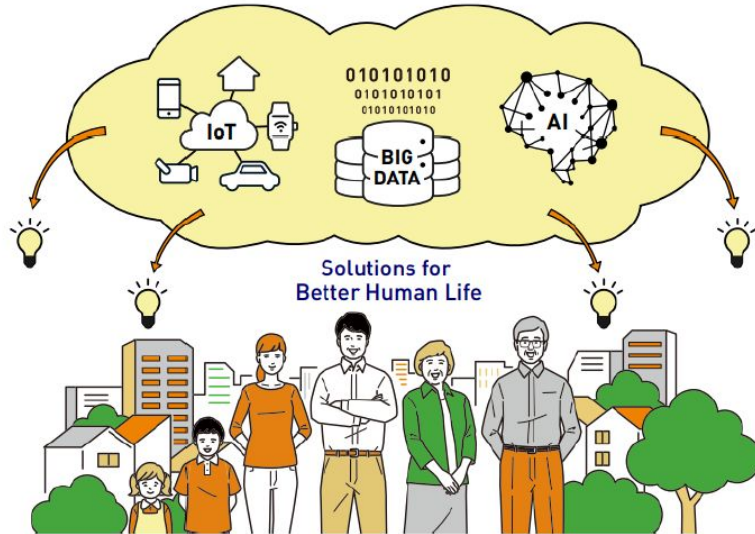


- Data Analyst
- Data Engineer
- Data Scientist
- Business Intelligence
- ML Engineer

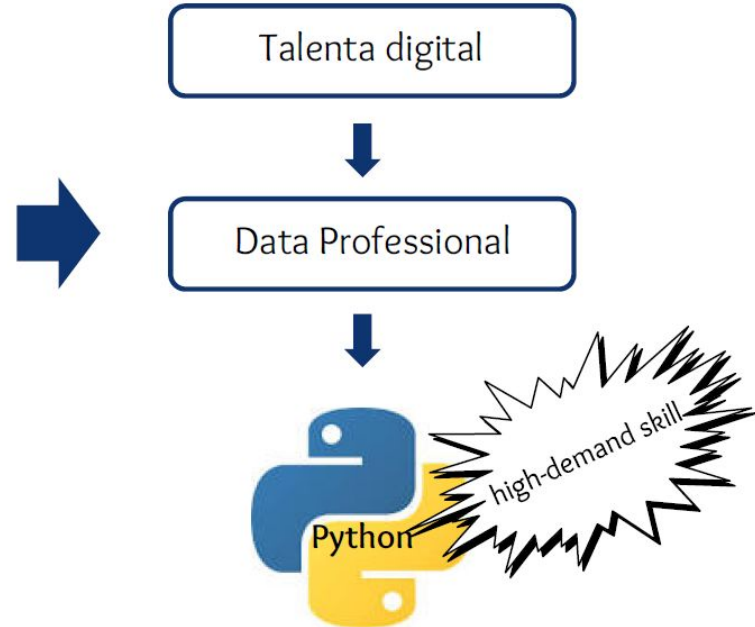


- Cocok untuk pemula
- Sederhana tapi *powerful*
- *High-demand skill*

# Mengapa Python



[https://www.japan.go.jp/abenomics/\\_userdata/abenomics/pdf/society\\_5.0.pdf](https://www.japan.go.jp/abenomics/_userdata/abenomics/pdf/society_5.0.pdf)

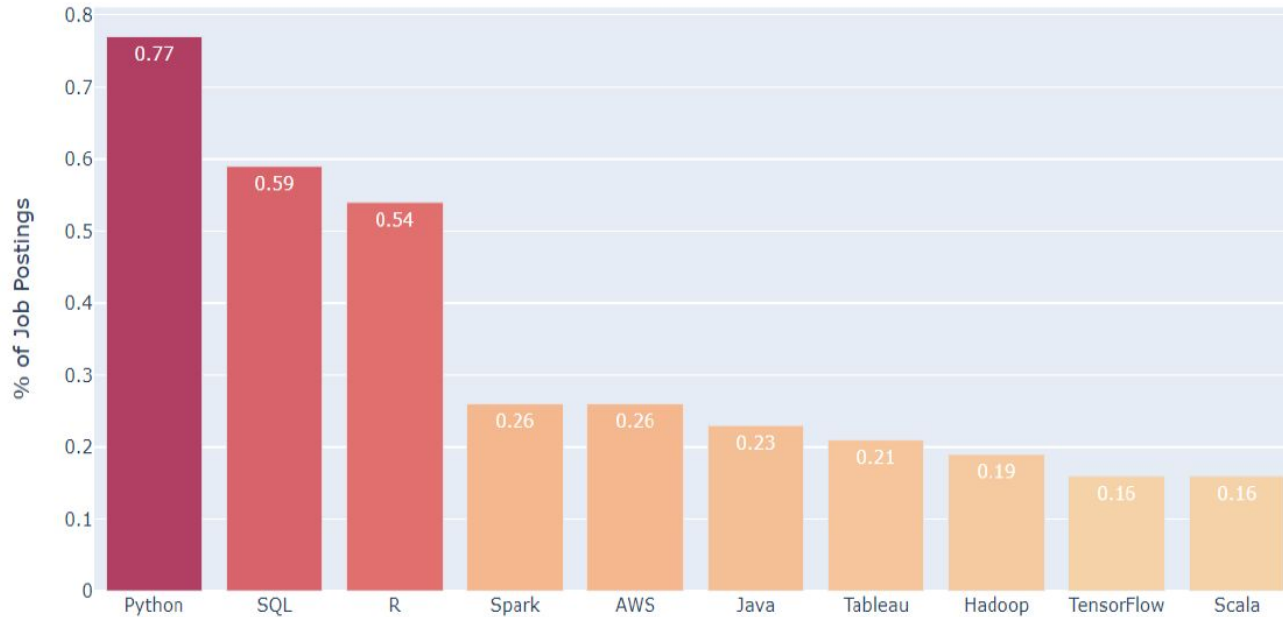


# Mengapa Python

Nama <i>Tools</i> (Alat)	Jumlah Pengguna
Python	86,7%
SQL	42,1%
R	23,9%
C++	21,4%
Java	18,8%
C	18,5%
JavaScript	16,7%
MATLAB	12,4%
Other	10,9%
Bash	9,9%

# Python menjadi yang Pertama dalam Daftar Keahlian yang paling dibutuhkan

10 Most In-Demand Data Science Skills in 2021



(sumber: <https://towardsdatascience.com>)



# Python digunakan pada Youtube

*"Google runs millions of lines of Python code. The front-end server that drives youtube.com and YouTubes APIs is primarily written in Python, and it serves millions of requests per second!"*

— Dylan Trotter, Youtube Engineer, 2017

<https://opensource.googleblog.com/2017/01/grumpy-go-running-python.html>



# Python digunakan pada Quora

The Quora logo, featuring the word "Quora" in a bold, red, serif font.

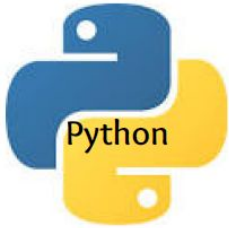
*“We decided that Python was fast enough for most of what we need to do (since we push our performance-critical code to backend servers written in C++ whenever possible). As far as typechecking, we ended up writing very thorough unit tests which are worth writing anyway, and achieve most of the same goals.”*  
— Adam D’Angelo, CEO Quora, 2014

<https://www.quora.com/Why-did-Quora-choose-Python-for-its-development>

# Python digunakan pada Beberapa Industri



# Penerapan Python pada Data Science



## Data Exploration

- Scraping, crawling, data mining
- Coding, query

## Data Pre-Processing

- Seleksi fitur, statistika deskriptif, *class balancing*, visualisasi data
- Transformasi fitur: *Categorical encoding*, *binning*

## Data Cleansing

- Menangani nilai kosong (*missing values*), menghapus baris terduplikasi
- *Data formating*, menangani data pencilan (*outliers*)

## Data Modeling

- Melatih data dengan algoritma *machine learning*
- Melakukan klasifikasi, regresi, prediksi, klasterisasi

# Memulai Python

- Python adalah bahasa *interpreter*, yang dapat mengurangi siklus *edit-test-debug* karena tidak memerlukan langkah kompilasi
- Untuk menjalankan Python, Anda memerlukan *runtime/interpreter environment* untuk mengeksekusi kode:
  - Mode interaktif: Setiap perintah yang Anda tulis akan langsung ditafsirkan dan segera dieksekusi sehingga bisa langsung melihat hasilnya → **IPython**
  - Mode skrip: Anda memasukkan satu set kode Python ke dalam format `.py`, program dijalankan baris demi baris



# Konsep Ipyhton: REPL Environment

**Read**

- Proses membaca code

**Eval**

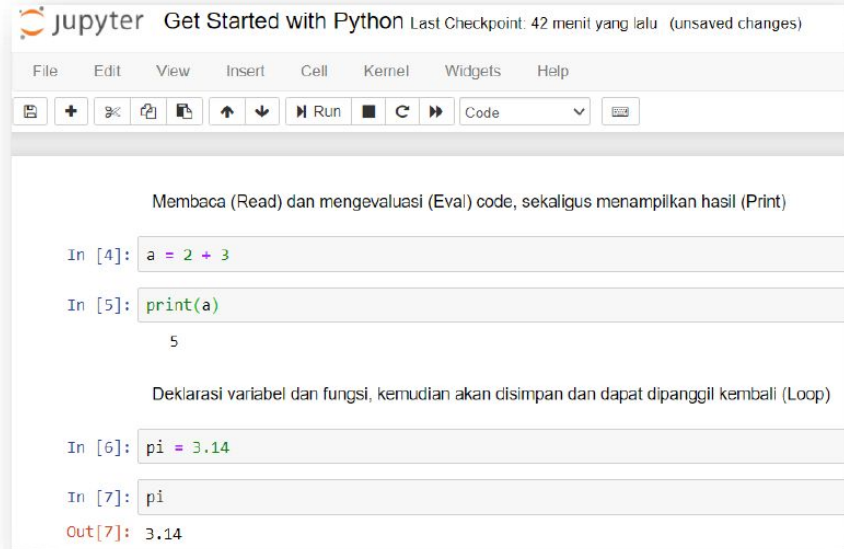
- Proses evaluasi (eksekusi) code

**Print**

- Proses menampilkan hasil (*output*)

**Loop**

- Pengulangan proses R-E-P



The screenshot shows a Jupyter Notebook window titled "jupyter Get Started with Python Last Checkpoint: 42 menit yang lalu (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and undo/redo. The notebook content area displays the following:

Membaca (Read) dan mengevaluasi (Eval) code, sekaligus menampilkan hasil (Print)

```
In [4]: a = 2 + 3
```

```
In [5]: print(a)
```

5

Deklarasi variabel dan fungsi, kemudian akan disimpan dan dapat dipanggil kembali (Loop)

```
In [6]: pi = 3.14
```

```
In [7]: pi
```

```
Out[7]: 3.14
```



# Pilihan Development Environment

Pilih *Development Environment* yang paling mudah dan nyaman:

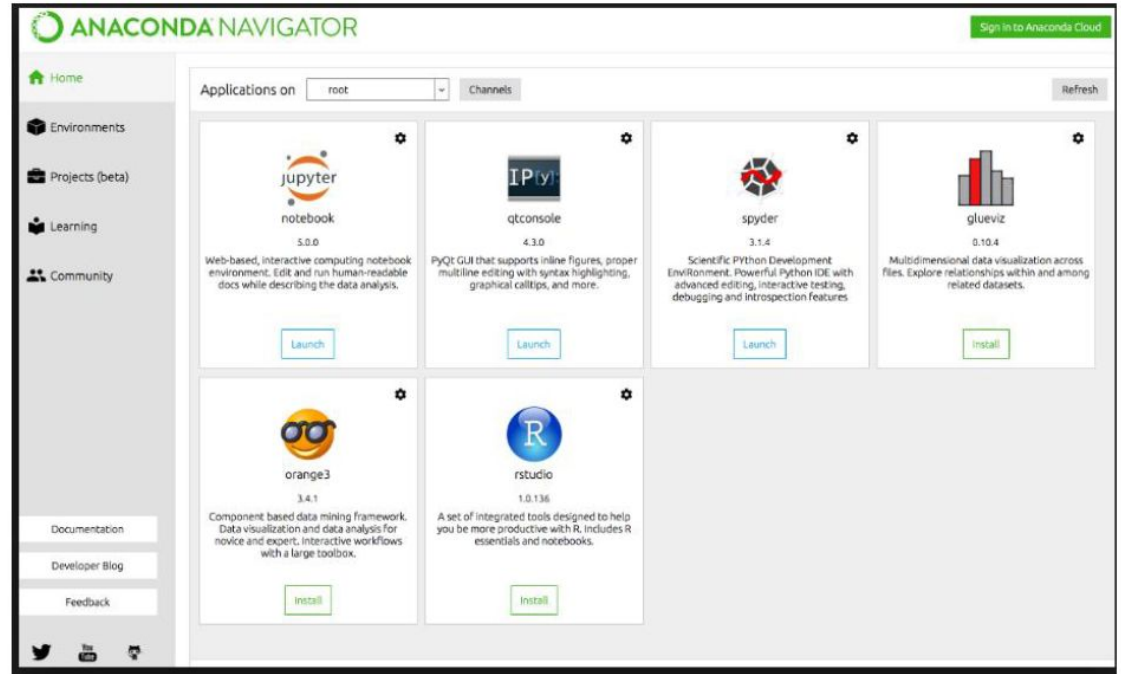
- Anaconda Distribution (<https://www.anaconda.com/distribution/>)
  - Python, Conda, lebih dari 1000 library data science
- Miniconda (<https://docs.conda.io/en/latest/miniconda.html>)
  - Python interpreter, Conda
- Jupyter Notebook (<https://jupyter.org/>)
- Python installer (<https://www.python.org/downloads/>).
- Google Colaboratory (<https://colab.research.google.com/>).
- Notebooks Azure (<https://notebooks.azure.com/>)

# Anaconda Distribution



Anaconda Navigator

Sebuah aplikasi *dashboard interface*  
pada paket Anaconda Distribution



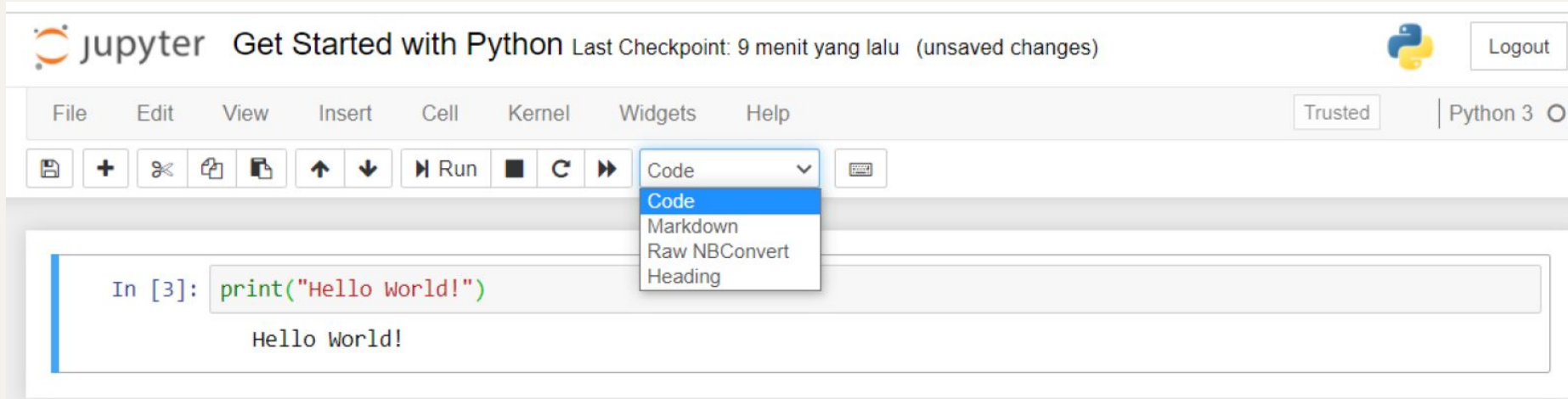


# Jupyter Notebook



- Lingkungan pemrograman interaktif berbasis web yang mendukung berbagai bahasa pemrograman termasuk Python
- Banyak digunakan oleh peneliti dan akademisi untuk pemodelan matematika, pembelajaran mesin, analisis statistik, dan untuk pengajaran pemrograman

# Jupyter Notebook



- Skrip dapat ditulis dalam bentuk:
  - *Code* : Algoritma dan formula matematis
  - *Markdown/Heading* : Teks deskripsi, penjelasan code
  - *Raw NBConvert* : Konversi format yang berbeda
- Hasil dapat diketahui langsung setelah menjalankan perintah *Run*

# Google Colaboratory



The screenshot displays the Google Colaboratory web interface. At the top, the browser address bar shows the URL `colab.research.google.com/notebooks/intro.ipynb#scrollTo=gJr_9dXGpJ05`. Below the address bar, the Google Colaboratory logo is followed by the text "Selamat Datang di Colaboratory". A menu bar contains the options: File, Edit, Lihat, Sisipkan, Runtime, Fitur, and Bantuan. On the right side of the menu bar, there are links for "Bagikan" (Share) and a settings icon. A left sidebar titled "Daftar isi" (Table of contents) lists: Memulai (Getting started), Ilmu data (Data science), Machine learning, Referensi Lainnya (Other references), Contoh Machine Learning (Machine Learning examples), and Bagian (Sections). The main workspace shows a code cell with the following content:

```
+ Kode + Teks Salin ke Drive
mencetak hasilnya.

seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day

86400
```

Below the code cell, a text box explains: "Untuk mengeksekusi kode dalam sel di atas, pilih kode tersebut dengan mengkliknya, kemudian tekan tombol putar di sebelah kiri kode atau gunakan pintasan keyboard 'Command/Ctrl+Enter'. Untuk mengedit kode, cukup klik sel dan mulai pengeditan."

- Skrip dapat ditulis dalam bentuk:
  - Code : Algoritma dan formula matematis
  - Teks: Teks deskripsi, penjelasan code
- Dapat digunakan pada <https://colab.research.google.com/> dan hasil dapat diketahui langsung setelah menialankan perintah *Run*

# Hello World !

## Bahasa C

```
#include <stdio.h>

int main() {
printf("Hello World!");
return 0;
}
```

## Bahasa Python

```
print("Hello World!")
```

- Lebih sederhana
- Tidak ada kurung kurawal {..}
- Tidak perlu titik koma ;

# Tipe Data Python

- float – bilangan riil
- int – bilangan bulat (integer)
- str – string, teks
- bool – True or False

```
In [1]: height = 1.84
```

```
In [2]: tall = True
```

- Masalah
  - Terlalu banyak data masukan untuk tipe data yang sama
  - Tidak nyaman

```
In [3]: height1 = 1.84
```

```
In [4]: height2 = 1.79
```

```
In [5]: height3 = 1.82
```

```
In [6]: height4 = 1.90
```

- Solusi → Python List

# Python List [a,b,c]

- Koleksi nilai-nilai
- Dapat mengandung beberapa tipe data berbeda

```
In [7]: [1.84, 1.79, 1.82, 1.90, 1.80]
```

```
Out[7]: [1.84, 1.79, 1.82, 1.90, 1.80]
```

```
In [8]: height = [1.84, 1.79, 1.82,  
1.90, 1.80]
```

```
In [9]: height
```

```
Out[9]: [1.84, 1.79, 1.82, 1.90, 1.80]
```

```
In [10]: famz = ["Abe", 1.84, "Beb",  
1.79, "Cory", 1.82, "Dad", 1.90]
```

```
In [11]: famz
```

```
Out[11]: ["Abe", 1.84, "Beb", 1.79,  
"Cory", 1.82, "Dad", 1.90]
```

```
["Abe", 1.84]
```

```
["Beb", 1.79]
```

```
["Cory", 1.82]
```

```
["Dad", 1.90]
```

# Python List

```
In [1]: height = [1.84, 1.79, 1.82, 1.90, 1.80]
```

```
In [2]: height
```

```
Out[2]: [1.84, 1.79, 1.82, 1.90, 1.80]
```

```
In [3]: weight = [66.5, 60.3, 64.7, 89.5, 69.8]
```

```
In [4]: weight
```

```
Out[4]: [66.5, 60.3, 64.7, 89.5, 69.8]
```

**Problem!**

```
In [5]: weight / height ** 2
```

```
TypeError: unsupported operand type(s) for ** or pow(): 'list' and 'int'
```



# Solusi : NumPy

- Library dasar untuk perhitungan saintifik (*scientific computing*) dengan Python (<https://numpy.org/>)
- Alternatif untuk Python List: Numpy Array untuk  $n$ -dimensi
- Mudah digunakan dan bersifat *open source*
- Jika library belum terpasang, tuliskan perintah instalasi:

```
pip install numpy
```

- Kemudian impor:

```
import numpy as np
```

```
In [6]: import numpy as np
```

```
In [7]: np_height = np.array(height)
```

```
In [8]: np_height
```

```
Out[8]: array([1.84, 1.79, 1.82, 1.9, 1.8])
```

```
In [9]: np_weight = np.array(weight)
```

```
In [10]: np_weight
```

```
Out[10]: array([66.5, 60.3, 64.7, 89.5, 69.8])
```

```
In [11]: bmi = np_weight / np_height ** 2
```

```
In [12]: bmi
```

```
Out[12]: array([19.64201323, 18.81963734, 19.53266514, 24.79224377, 21.54320988])
```



# NumPy



- Pengolahan data dapat berupa bermacam-macam bentuk dan formatnya: dokumen, gambar, video, suara, angka, atau teks
- Ketika data-data tersebut diproses, tidak secara mentah-mentah dibaca sebagai video atau audio. Tetapi sudah dilakukan transformasi ke dalam bentuk array atau *matrix of number*
- Array dengan minimal dua dimensi akan membentuk matriks dan dapat menggunakan NumPy

```
import numpy as np  
np.<TAB>
```

# Numpy

- NumPy juga dapat digunakan untuk membuat array berdimensi- $n$

```
In [13]: import numpy as np
```

```
In [14]: np_height = np.array([1.84, 1.79, 1.82, 1.9, 1.8])
```

```
In [15]: np_weight = np.array([66.5, 60.3, 64.7, 89.5, 69.8])
```

```
In [16]: type(np_height)
Out[16]: numpy.ndarray
```

```
In [16]: type(np_weight)
Out[16]: numpy.ndarray
```

*ndarray = n-dimensional array*

```
In [17]: np_2d = np.array([[1, 2, 3, 4, 5],
                           [6, 7, 8, 9, 10]])
```

```
In [18]: np_2d
Out[18]: array([[1, 2, 3, 4, 5],
                [6, 7, 8, 9, 10]])
```

```
In [19]: np_2d.shape
Out[19]: (2, 5)
```

Array berdimensi 2 baris 5 kolom → Matriks  $M_{2 \times 5}$

$$M = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \end{bmatrix}$$

# SciPy

- SciPy (dibaca “Sigh Pie”) merupakan library yang bersifat *open source* dan tersedia di <https://www.scipy.org/>
- SciPy dibangun untuk untuk bekerja dengan NumPy array dan menyediakan kumpulan algoritma numerik, termasuk pemrosesan sinyal, optimasi, statistika, dan library Matplotlib untuk visualisasi data.
- Jika library belum terpasang, tuliskan perintah instalasi:  

```
pip install scipy
```

# Pandas

- Pandas (Panel Data) merupakan library popular di Python yang digunakan untuk *data structure* dan *data analysis*
- Bersifat *open source* dan tersedia di <https://pandas.pydata.org/>
- Pandas sangat berkaitan dengan NumPy
- Jika library belum terpasang, tuliskan perintah instalasi:  

```
pip install pandas
```
- Kemudian impor:  

```
import pandas as pd
```

## *Data Wrangling / Data Munging*

- *Reshaping* (mengubah bentuk data)
- *Joining* (menggabungkan data)
- *Splitting* (pemisahan data)
- *Time-series analysis* (data berkala)

## *Data Cleansing*

- Membersihkan data tidak lengkap (*Error*)
- Menangani data pencilan (*outliers*)
- Menghapus data duplikat

# Representasi data di Pandas

- Terdapat 2 *data objects*: *Series* dan *DataFrame*
- *Series* → Data berbentuk 1 dimensi

```
In [13]: np.array([1, 2, 3, 4, 5])
```

```
Out[13]: array([1, 2, 3, 4, 5])
```

- *DataFrame* → Data berbentuk 2 dimensi atau lebih

```
In [14]: np.array([[1, 2], [3, 4]])
```

```
Out[14]: array([[1, 2],  
                [3, 4]])
```

Kolom: Fitur / atribut

	Negara	Populasi	Area	Ibukota
IN	Indonesia	250	123456	Jakarta
MA	Malaysia	25	3456	KL
SI	Singapura	15	456	Singapura
JP	Jepang	60	5678	Tokyo
TH	Thailand	45	678	Bangkok

Baris: sampel

# Pandas

- Pandas dapat mengimpor data dari berbagai format: *comma-separated value* (CSV), file teks, Microsoft Excel, database SQL, dan format HDF5
- Unduh dataset: <http://bit.ly/TabDataset>
- CSV file → DataFrame

```
import pandas as pd
```

Tab.csv

```
,Negara,Populasi,Area,Ibukota  
IN,Indonesia,250,123456,Jakarta  
MA,Malaysia,25,3456,KL  
SI,Singapura,15,456,Singapura  
JP,Jepang,60,5678,Tokyo  
TH,Thailand,45,678,Bangkok
```

```
In [1]: Tab = ...    # deklarasi tabel
```

```
In [2]: Tab
```

	Negara	Populasi	Area	Ibukota
IN	Indonesia	250	123456	Jakarta
MA	Malaysia	25	3456	KL
SI	Singapura	15	456	Singapura
JP	Jepang	60	5678	Tokyo
TH	Thailand	45	678	Bangkok



# Pandas

```
In [3]: import pandas as pd
```

```
In [4]: Tab = pd.read_csv("Tab.csv")
```

```
In [5]: Tab
```

```
Out[5]:
```

	Unnamed: 0	Negara	Populasi	Area	Ibukota
0	IN	Indonesia	250	123456	Jakarta
1	MA	Malaysia	25	3456	KL
2	SI	Singapura	15	456	Singapura
3	JP	Jepang	60	5678	Tokyo
4	TH	Thailand	45	678	Bangkok

```
In [6]: Tab["Negara"] # akses kolom
```

```
Out[6]:
```

```
0    Indonesia
1    Malaysia
2    Singapura
3     Jepang
4    Thailand
Name: Negara, dtype: object
```

```
In [7]: Tab.Ibukota # akses kolom
```

```
Out[7]:
```

```
0    Jakarta
1         KL
2    Singapura
3     Tokyo
4    Bangkok
Name: Ibukota, dtype: object
```

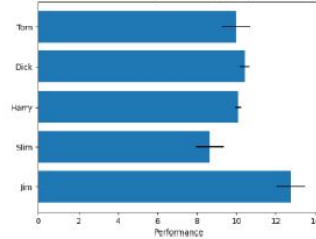
# Matplotlib

- Matplotlib adalah library Python untuk visualisasi data dengan dua dimensi
- Bersifat *open source* dan tersedia di <https://matplotlib.org/>
- Matplotlib berkaitan dengan NumPy dan Pandas
- Jika library belum terpasang, tuliskan perintah instalasi:

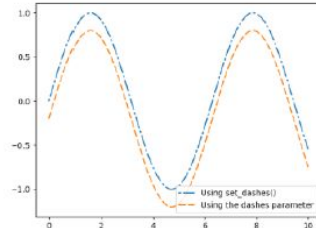
```
pip install matplotlib
```

- Kemudian impor:

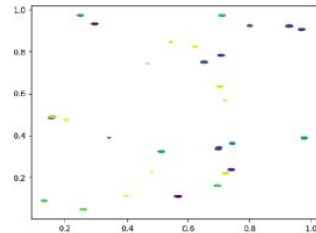
```
import matplotlib.pyplot as plt
```



*bar chart*



*Line chart*



*Scatter plot*



# Matplotlib

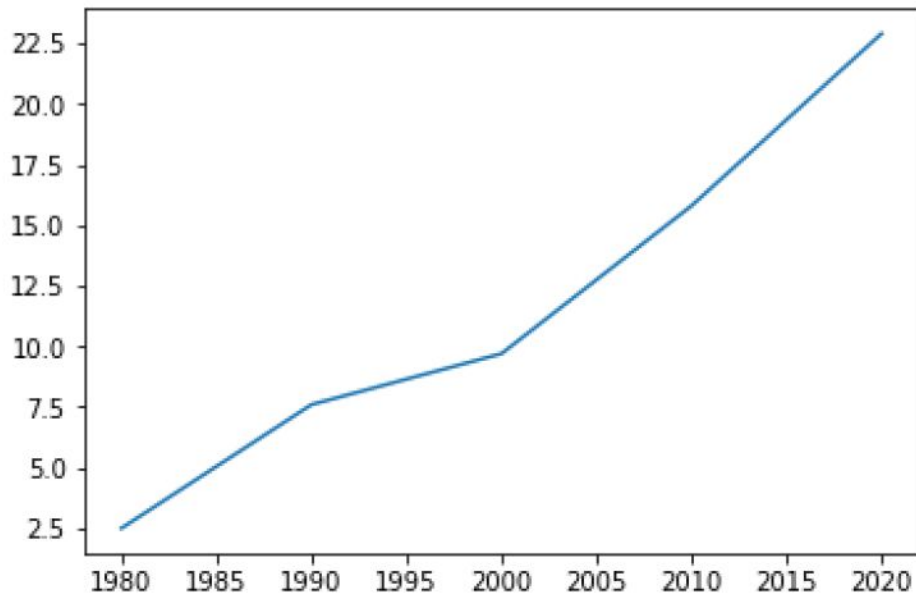
```
In [1]: import matplotlib.pyplot as plt
```

```
In [2]: year = [1980, 1990, 2000, 2010, 2020]
```

```
In [3]: price = [2.5, 7.6, 9.7, 15.8, 22.9]
```

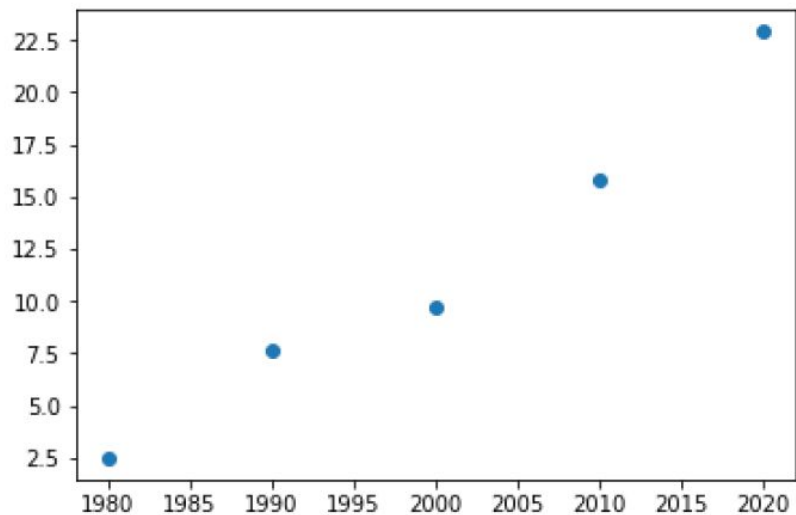
```
In [4]: plt.plot(year, price)
```

```
In [5]: plt.show()
```

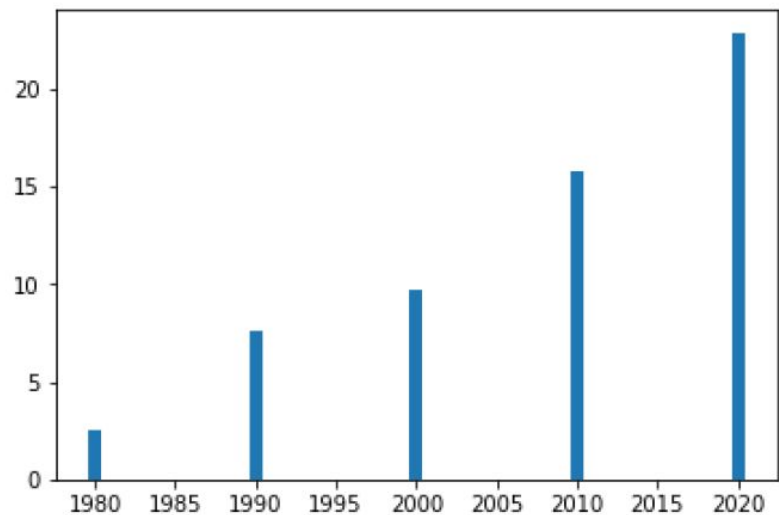


# Matplotlib

```
In [6]: plt.scatter(year, price)
```



```
In [7]: plt.bar(year, price)
```



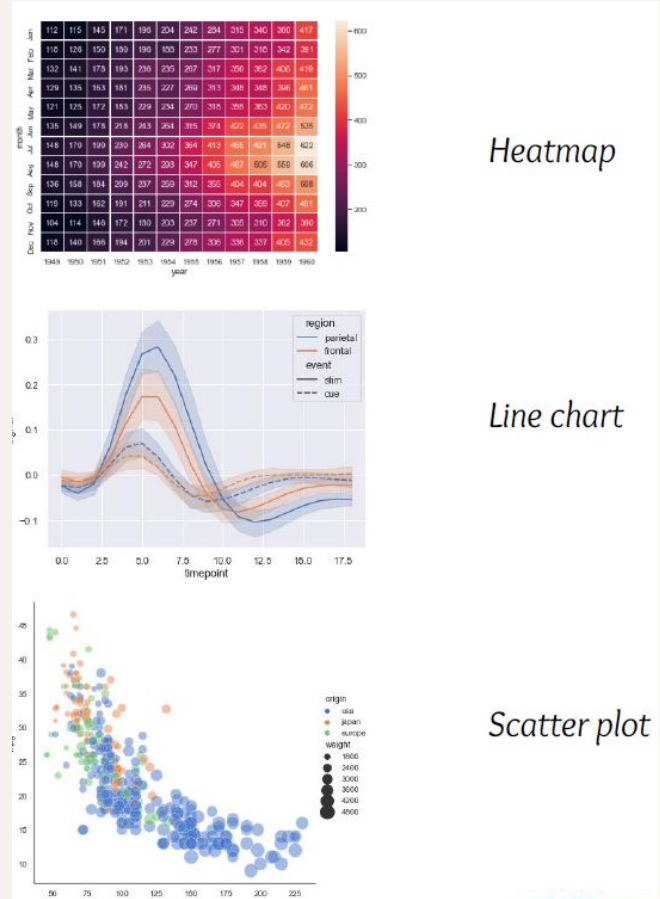
# Seaborn

- Seaborn adalah library visualisasi data Python (serupa dengan Matplotlib) yang menyediakan *high-level interface* untuk menggambar grafik statistika yang menarik dan informatif
- Library ini bersifat *open source* dan tersedia di <https://seaborn.pydata.org/>
- Jika library belum terpasang, tuliskan perintah instalasi:

```
pip install seaborn
```

- Kemudian impor:

```
import seaborn as sns
```



# Scikit-Learn

- Scikit-learn adalah library untuk mempraktikkan *machine learning* dan membuat model
- Bersifat *open source* dan tersedia di <https://scikit-learn.org/>
- Scikit-learn diawali dari project SciPy (*Scientific Python*) yang berisi fungsi-fungsi matematis
- Jika library belum terpasang, tuliskan perintah instalasi:  

```
pip install sklearn
```
- Kemudian impor:  

```
import sklearn
```

## Classification

---

- Support Vector Machines
- Decision Tree
- Random Forest
- Neural Network
- Nearest neighbors

## Clustering

---

- K-Means Clustering
- Hierarchical Clustering

## Model Selection

---

- Cross validation
- Metrics

# Ringkasan

Pada topik ini, kita sudah mempelajari:

- Keunggulan Python sebagai Tools dalam proyek data science
- *Development environment* Python yang bervariasi, baik yang bersifat *offline (local computer)* maupun berbasis web (Jupyter Notebook / Google Colaboratory)
- Dasar-dasar library Python untuk proyek data science:
  - NumPy → library untuk *numerical computation*
  - SciPy → library untuk perhitungan statistika dan matematis
  - Pandas → library untuk analisis dan manipulasi data
  - Matplotlib → library untuk visualisasi data
  - Seaborn → library untuk visualisasi data dengan *high-level interface*
  - Scikit-learn → library untuk mempraktikkan *machine learning* dan pemodelan