

BÀI TẬP THỰC HÀNH MÔN THỐNG KÊ MÁY TÍNH

- ❖ Bài tập được thiết kế theo từng lab, mỗi lab là 3 tiết có sự hướng dẫn của GV.
- ❖ Cuối mỗi buổi thực hành, sinh viên nộp lại phần bài tập mình đã thực hiện cho GV hướng dẫn.
- ❖ Những câu hỏi mở rộng/khó giúp sinh viên trau dồi thêm kiến thức của môn học. Sinh viên phải có trách nhiệm nghiên cứu, tìm câu trả lời nếu chưa thực hiện xong trong giờ thực hành.

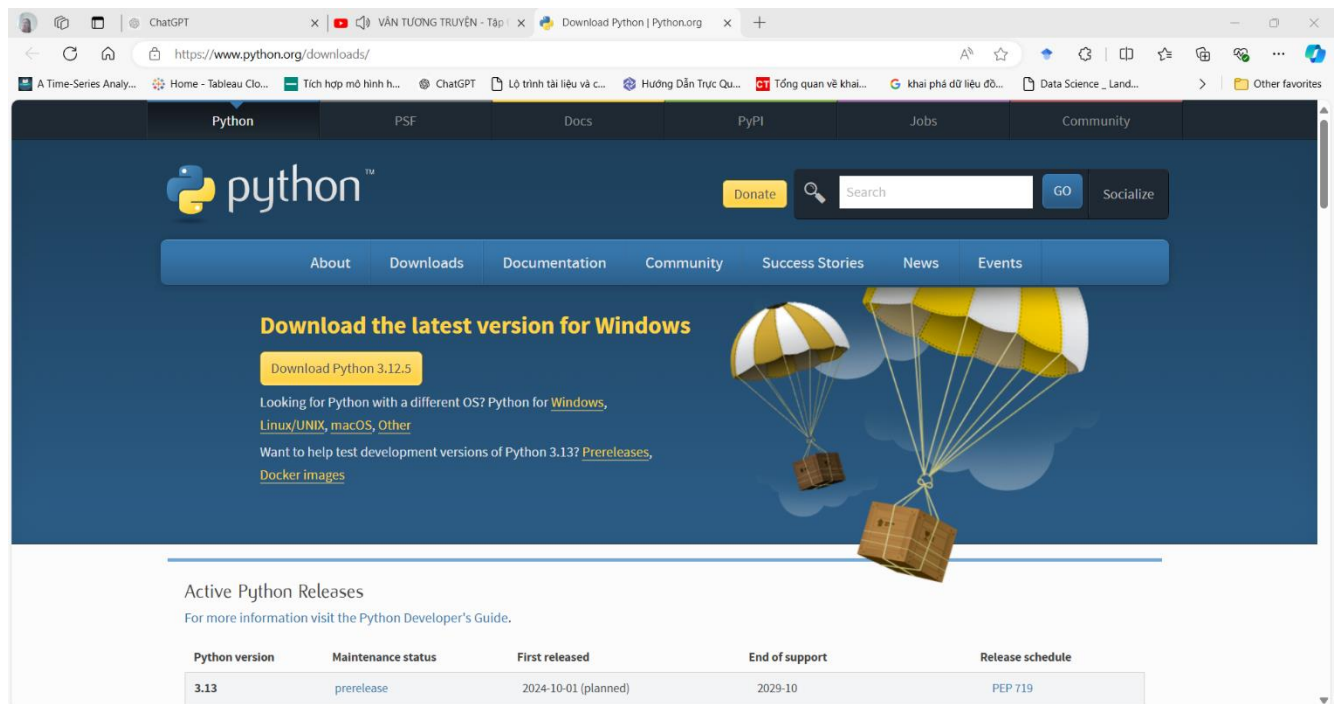
Lab 1. LÀM QUEN VỚI PYTHON

Nội dung:

1. Download Python
2. Cài đặt Python
3. Làm quen với Python
4. Các IDE cho Python
5. Các package quan trọng sử dụng trong thống kê
6. Bài tập

1. Download Python

Để download Python, bạn truy cập địa chỉ: [python.org/downloads/](https://www.python.org/downloads/) Nhấn vào nút **Download Python 3.12.5** để download phiên bản mới nhất của Python.



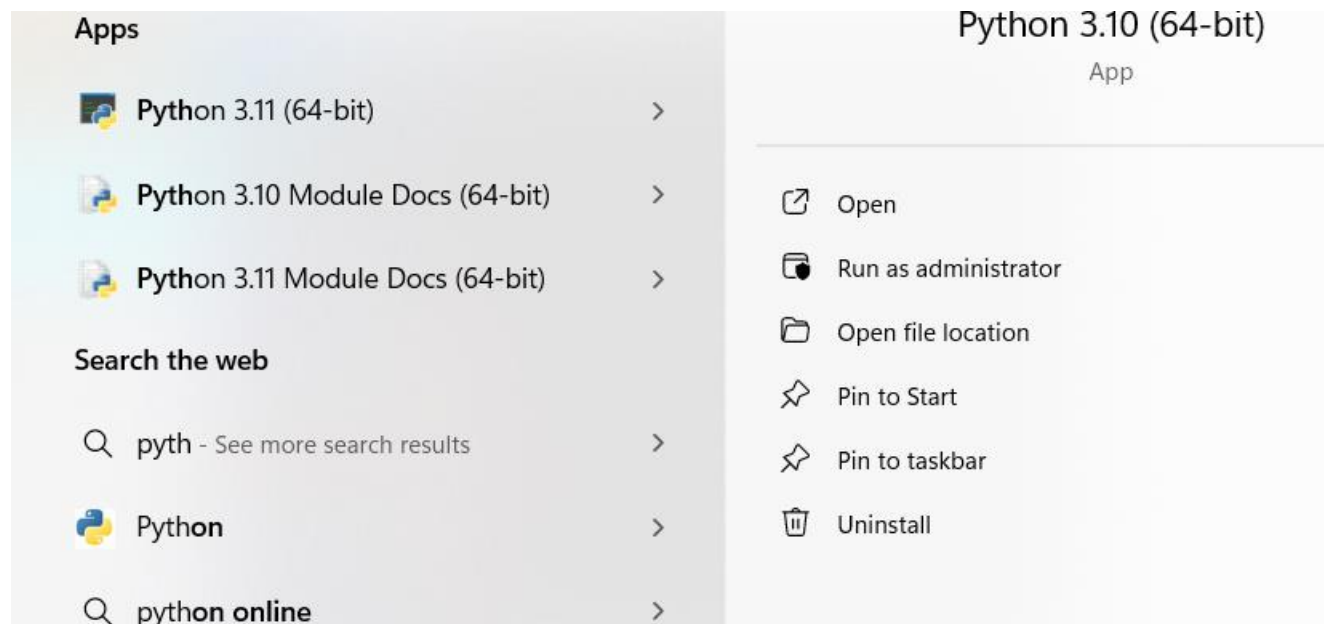
Sau khi download xong bạn có 1 file **python-3.12.5.exe**

2. Cài đặt Python

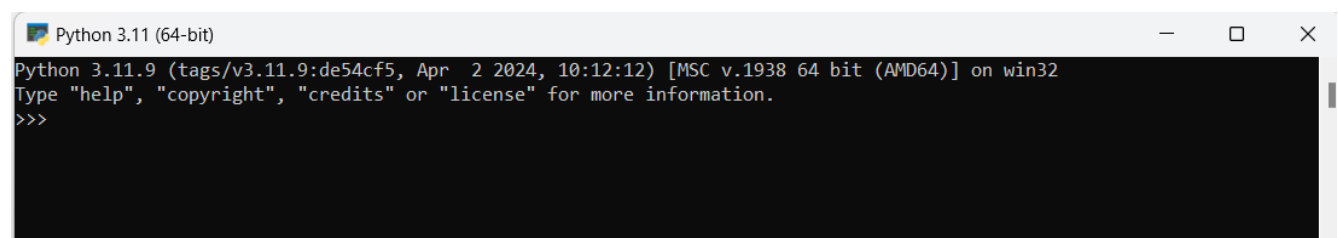
Thực thi file bạn download được ở bước trên để bắt đầu cài đặt. Chọn “**Customize Installation**” để bạn có thể tùy chọn vị trí **Python** sẽ được cài đặt. Thực hiện theo các bước để hoàn thành việc cài đặt.

3. Làm quen với Python

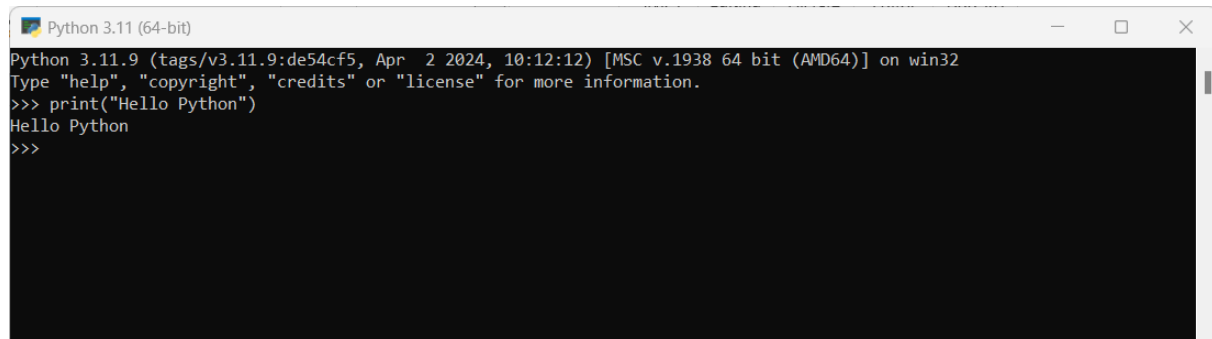
Vào mục tìm kiếm của Window gõ chữ “Python”, sẽ xuất hiện IDLE (Python 3.12 64- bit), nhấn chọn vào IDLE trên.



Chương trình “Python Shell” đã được thực thi, nó là một chương trình giúp bạn viết mã Python. Dưới đây là hình ảnh của Python Shell:



Nhập vào một đoạn code: `print("Hello Python")` và nhấn Enter

A screenshot of a Python 3.11 (64-bit) shell window. The title bar reads "Python 3.11 (64-bit)". The window content shows the Python version and build information: "Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32". It then prompts the user to type "help", "copyright", "credits" or "license" for more information. The user has entered the command `>>> print("Hello Python")`, and the shell has responded with `Hello Python` and a new prompt `>>>`.

Sau khi bạn cài đặt xong Python, ta có thêm một công cụ Python Shell, đây là một IDE (Integrated Development Environment) giúp bạn viết mã Python. Nếu bạn không muốn sử dụng Python Shell bạn có thể sử dụng một IDE khác.

4. Các IDE cho Python

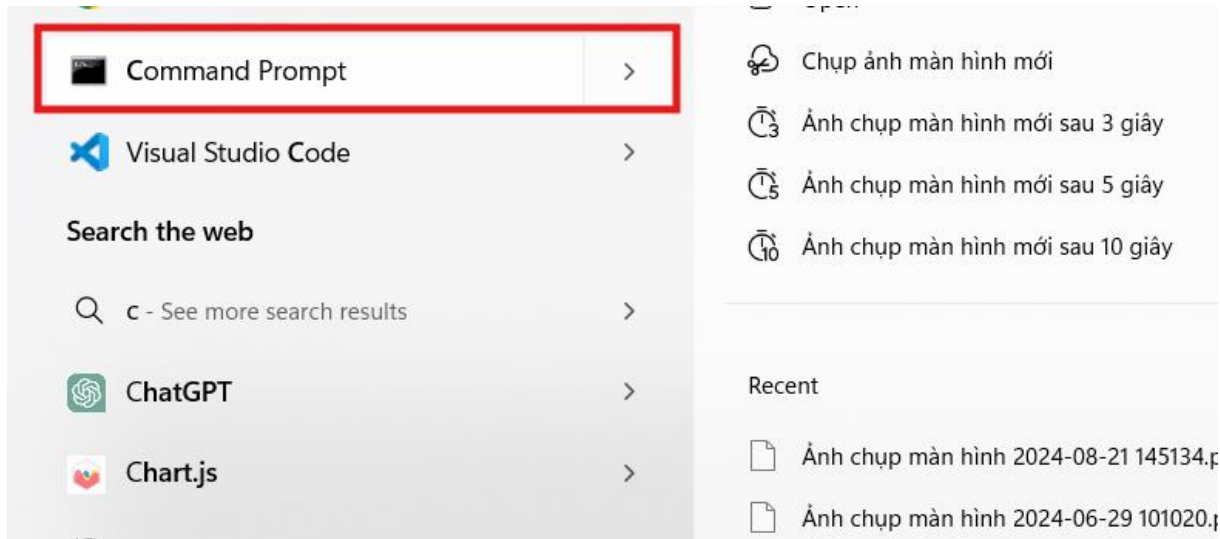
Một số IDE giúp bạn lập trình **Python**:

- PyCharm
- Anaconda
- **Jupyter Notebook**
-

Hướng dẫn cài đặt Jupyter Notebook:

Sau khi cài đặt xong Python 3.7, vào **Command Prompt** gõ lệnh: **pip install jupyter**

Nếu chương trình không nhận biết được lệnh trên thì gõ lệnh **py -m pip install jupyter**



1. **Khởi động Jupyter Notebook:** Ở command prompt, nhập vào câu lệnh dưới đây, server sẽ được khởi động, và có thể xác nhận việc hiển thị giao diện của Jupyter Notebook ở browser.

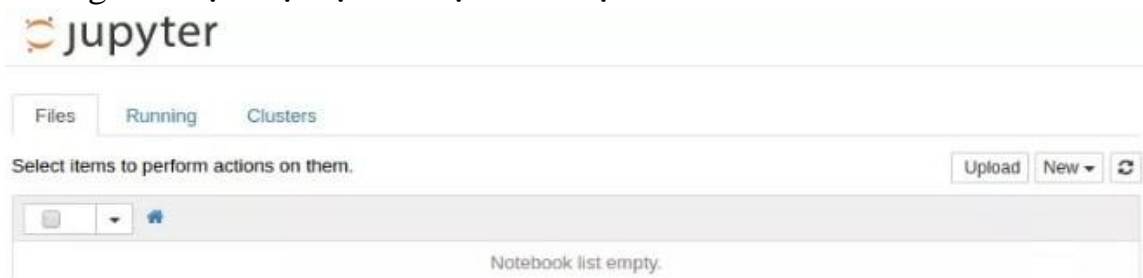
```
jupyter notebook
```

Nếu chương trình không nhận biết được lệnh trên thì gõ lệnh **py -m jupyter notebook**

Mặc định thì Jupyter Notebook sẽ sử dụng cổng 8888, tuy nhiên cũng có thể chỉ định cổng khác bằng tham số **--port**. Xem ví dụ dưới:

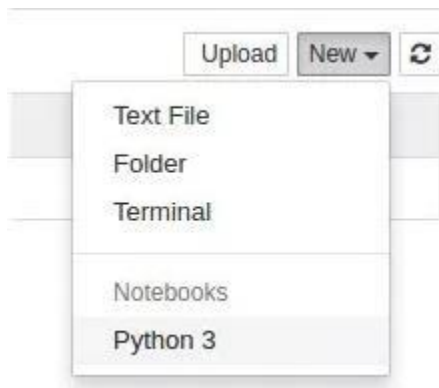
```
jupyter notebook --port 9000
```

Sau khi khởi động, màn hình dưới đây sẽ hiển thị. Ở màn hình này, danh sách các file trong thư mục hiện tại sẽ được hiển thị.

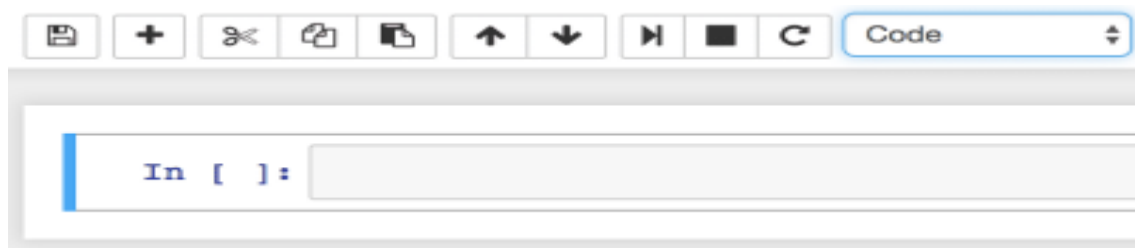


Home page của Jupyter Notebook

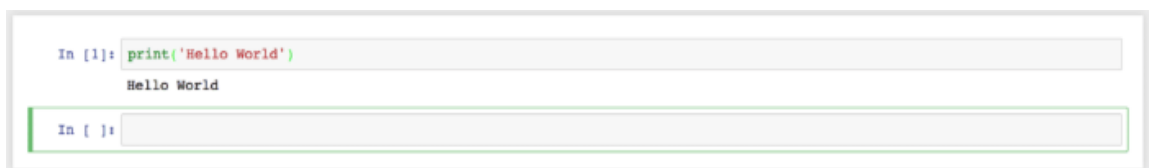
2. **Cách mở một Notebook mới:** Click vào button 「New」 ở góc bên phải, rồi lựa chọn 「Python 3」 để có thể mở một Notebook mới.



3. **Làm việc với Notebook:** Một notebook bao gồm nhiều cell (ô). Khi tạo mới một notebook, bạn luôn được tạo sẵn một cell rỗng đầu tiên.



Cell trên có kiểu là “Code”, điều đó có nghĩa là bạn có thể gõ code Python vào cell này. Để thực thi code, bạn có thể nhấn nút Run cell hoặc nhấn phím Ctrl + Enter.



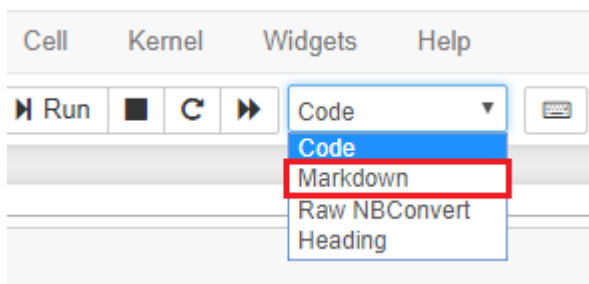
Kết quả được hiển thị tại ô bên dưới. Một cell rỗng sẽ được tạo sau khi bạn thực thi code. Hãy gõ tiếp một đoạn code Python dưới đây để thử nghiệm:

```
In [1]: print('Hello World')
Hello World

In [2]: i = 1
while i <= 10:
    print(i)
    i = i + 1
1
2
3
4
5
6
7
8
9
10

In [ ]:
```

Bạn có thể chuyển loại cell từ **Code** thành **Markdown** để viết những đoạn văn bản giải thích code của bạn. Để chuyển đổi, bạn click vào ComboBox **Code** và chọn **Markdown** như hình:



Sau khi chuyển, hãy nhập ngay một đoạn **Markdown** sau để thử nghiệm

```
# This is a headline
## Sub headline

**Text**

More Text
```

Bạn cũng nhấn nút Run cell hoặc nhấn Ctrl + Enter để xem kết quả

```
In [1]: print('Hello World')
Hello World

In [2]: i = 1
while i <= 10:
    print(i)
    i = i + 1
1
2
3
4
5
6
7
8
9
10

This is a headline

Sub headline

Text

More Text

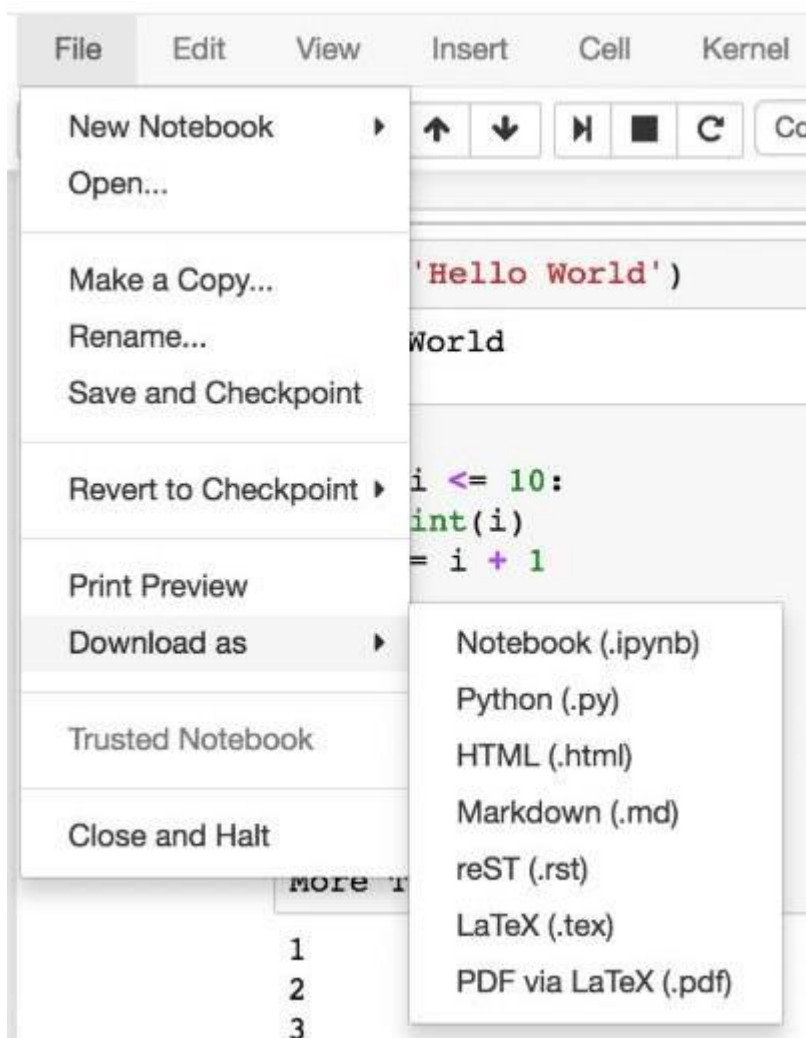
In [ ]:
```

Nếu bạn muốn chỉnh sửa đoạn **Markdown** vừa thực thi thì chỉ việc click vào kết quả vừa xuất hiện và bạn sẽ được chuyển sang chế độ chỉnh sửa.

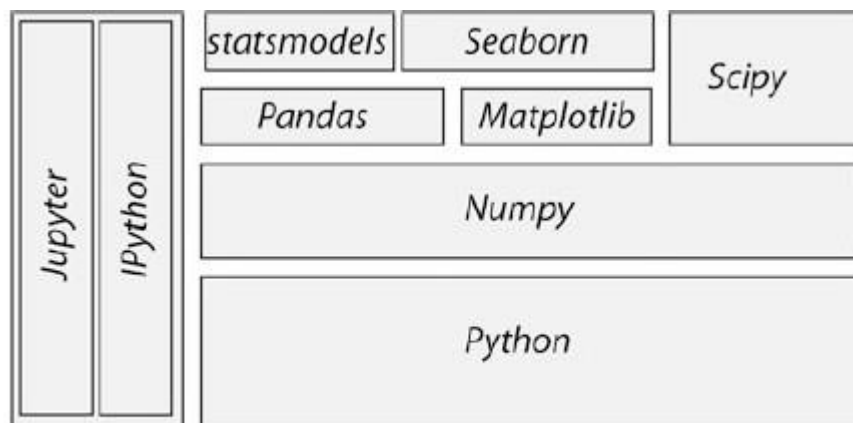
4. **Checkpoint:** Một trong những chức năng cực hay của Jupyter Notebook là Checkpoints. Bằng cách tạo các Checkpoints lưu trạng thái hiện tại của notebook, Jupyter Notebook cho phép bạn có thể quay lại thời điểm tạo Checkpoints để kiểm tra hoặc hoàn tác trước đó.

Để tạo Checkpoint, chọn **File -> Save and Checkpoint**. Nếu bạn muốn xem lại các Checkpoints trước đó thì chọn **File -> Revert to Checkpoint**.

5. **Chức năng Export notebook:** Jupyter Notebook cho phép bạn export notebook của bạn ra một vài loại file như: PDF, HTML, Python(.py),...Để làm được điều đó, bạn chọn **File -> Download as:**



2. Các package quan trọng sử dụng trong thống kê:



The structure of the most important *Python* packages for statistical applications

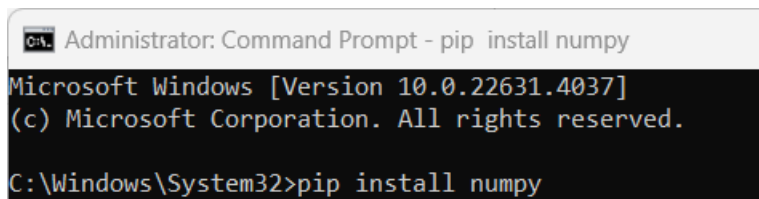
- **numpy**: dùng cho các kiểu dữ liệu vector và array
- **scipy**: dùng cho các thuật toán cơ bản trong thống kê
- **matplotlib**: dùng để vẽ các dạng đồ thị
- **seaborn**: dùng để vẽ các dạng đồ thị
- **pandas**: dùng cho các Dataframe (giống 1 bảng gồm các dòng và các cột)
- **statsmodels**: dùng để mô hình hóa thống kê và phân tích nâng cao ví dụ như phân tích hồi quy và phân tích phương sai.

Hướng dẫn cài đặt các package này: vào Command Prompt của Window gõ lệnh: **pip install <tên gói>**

Nếu chương trình không nhận biết được lệnh trên thì gõ lệnh

py -m pip install <tên gói>

Ví dụ: pip install numpy



```
C:\> Administrator: Command Prompt - pip install numpy
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\System32>pip install numpy
```

3. Bài tập:

➤ **Kiểu dữ liệu: Tuple, List, Array và DataFrames**

+ **Tuple()**: một tập hợp các kiểu dữ liệu khác nhau, không thể sửa đổi khi đã tạo.

Ví dụ:

```
In [1]: import numpy as np

In [2]: myTuple = ('abc', np.arange(0,3,0.2), 2.5)

In [3]: myTuple[2]
Out[3]: 2.5
```

+ **List[]**: các phần tử trong list có thể được cập nhật. Vì vậy, list thường được sử dụng cho các item cùng kiểu dữ liệu chẳng hạn kiểu dữ liệu số, chuỗi. Chú ý: phép cộng list là “+”

Ví dụ:

```
In [4]: myList = ['abc', 'def', 'ghij']

In [5]: myList.append('klm')

In [6]: myList
Out[6]: ['abc', 'def', 'ghij', 'klm']

In [7]: myList2 = [1,2,3]

In [8]: myList3 = [4,5,6]

In [9]: myList2 + myList3
Out[9]: [1, 2, 3, 4, 5, 6]
```

+ **Array []**: vectors và matrices, dùng để thao tác với kiểu dữ liệu dạng số, được định nghĩa trong package numpy. Phép toán ‘+’, ‘.dot’ dùng để cộng, nhân các phần tử trong mảng lại với nhau.

Ví dụ:

```
In [10]: myArray2 = np.array(myList2)
```

```
In [11]: myArray3 = np.array(myList3)
```

```
In [12]: myArray2 + myArray3
```

```
Out [12]: array([5, 7, 9])
```

```
In [13]: myArray2.dot(myArray3)
```

```
Out [13]: 32
```

+ **DataFrame**: cấu trúc dữ liệu sử dụng cho dữ liệu thống kê, được định nghĩa trong package **pandas**.

- DataFrame là cấu trúc dữ liệu 2 chiều, có gắn nhãn với các cột có thể giống hoặc khác kiểu dữ liệu, giống như một bảng dữ liệu gồm các dòng và các cột.

Ví dụ: tạo 1 DataFrame với 3 cột có tên là “Time,” “x,” và “y”:

```
import numpy as np
import pandas as pd

t = np.arange(0,10,0.1)
x = np.sin(t)
y = np.cos(t)

df = pd.DataFrame({'Time':t, 'x':x, 'y':y})
```

- Trong pandas, các dòng được xử lý thông qua các chỉ số và cột thông qua tên của chúng.
- Để lấy dữ liệu cột tên “Time”, bạn có hai cách sau:

```
df.Time  
df['Time']
```

- Nếu bạn muốn lấy dữ liệu hai cột cùng một lúc, bạn thực hiện như sau:

```
data = df[['Time', 'y']]
```

- Để hiển thị dòng 5 dòng đầu tiên hoặc 5 dòng cuối cùng của DataFrame, sử dụng:

```
data.head()  
data.tail()
```

- Để lấy dữ liệu từ dòng 5 đến dòng 10, sử dụng:

```
data[4:10]
```

- Để lấy dữ liệu đồng thời 2 cột “Time” và “y”, dòng 5 đến dòng 10, sử dụng: Hoặc có thể sử dụng:

```
df[['Time', 'y']][4:10]  
df.loc[4:10, ['Time', 'y']]
```

➤ Đọc dữ liệu từ file text vào DataFrame:

Bạn có thể dễ dàng đọc vào một file **.csv** bằng cách sử dụng hàm **read_csv** và được trả về 1 dataframe.

Bạn cũng có thể dùng hàm **read_csv** để đọc **1 file text** và cũng được trả về 1 dataframe.

Tuy nhiên, bạn cũng sẽ phải lưu ý một vài tham số của hàm **read_csv** như:

- **encoding**: chỉ định encoding của file đọc vào. Mặc định là utf-8.

```
df = pd.read_csv('example.txt', encoding='utf-16')
```

- **sep**: thay đổi dấu ngăn cách giữa các cột. Mặc định là dấu phẩy (‘,’)

df = pd.read_csv('example.txt', sep=';') # Dấu phân cách là dấu chấm phẩy

- **header**: chỉ định file đọc vào có header (tiêu đề của các cột) hay không. Mặc định là infer.

df = pd.read_csv('example.txt', header=0) # Dòng đầu tiên là header

df=pd.read_csv('example.txt', header=None) # Không có header

- **index_col**: chỉ định chỉ số cột nào là cột chỉ số (số thứ tự). Mặc định là None.

df = pd.read_csv('example.txt', index_col=0) # Cột đầu tiên làm chỉ số của DataFrame

- **n_rows**: chỉ định số bản ghi sẽ đọc vào. Mặc định là None – đọc toàn bộ.

df = pd.read_csv('example.txt', nrows=100) # Chỉ đọc 100 dòng đầu tiên

Ví dụ:

Đọc dữ liệu từ file **babies.txt** vào **DataFrame**:

```
#Đọc dữ liệu từ file babies.txt vào DataFrame:
import pandas as pd
import numpy as np
data=pd.read_csv('babies.txt',sep='\t')
print(data)
```

	bwt	smoke
0	120	0
1	113	0
2	128	1
3	123	0
4	108	1
...
1231	113	0
1232	128	0
1233	130	1
1234	125	0
1235	117	0

[1236 rows x 2 columns]

Tạo 1 DataFrame tên là **df_data** gồm tất cả các dòng dữ liệu, các **cột được đặt tên là: bwt và smoke** (nếu file dữ liệu đã có header thì lệnh trên sẽ đặt lại tên header)

```
import pandas as pd

# Đọc dữ liệu từ file babies.txt với khoảng trắng làm dấu phân cách
data = pd.read_csv('babies.txt', delim_whitespace=True)

# Chọn các cột mong muốn (ở đây có thể không cần vì tất cả các cột đều hữu ích)
df_data = data[['bwt', 'smoke']]

# In dữ liệu ra màn hình
print(df_data)
```

	bwt	smoke
0	120	0
1	113	0
2	128	1
3	123	0
4	108	1
...
1231	113	0
1232	128	0
1233	130	1
1234	125	0
1235	117	0

[1236 rows x 2 columns]

Tạo 1 DataFrame tên là **df_cohutthuoc** gồm các dòng dữ liệu có cột **smoke=1**

```
#Tạo 1 DataFrame tên là df_cohutthuoc gồm các dòng dữ liệu có cột smoke=1
import pandas as pd

# Đọc dữ liệu từ file babies.txt với khoảng trắng làm dấu phân cách
data = pd.read_csv('babies.txt', delim_whitespace=True)

# Lọc các dòng có cột smoke = 1
df_cohutthuoc = data[data['smoke'] == 1]

# In ra DataFrame mới chứa các dòng có cột smoke = 1
print(df_cohutthuoc)
```

	bwt	smoke
2	128	1
4	108	1
9	143	1
11	144	1
12	141	1
...
1224	143	1
1225	113	1
1226	109	1
1227	103	1
1233	130	1

[484 rows x 2 columns]

Tạo 1 DataFrame tên là **df_khonghutthuoc** gồm các dòng dữ liệu có cột **smoke=0**

```
# Lọc các dòng có cột smoke = 0
df_cohutthuoc = data[data['smoke'] == 0]

# In ra DataFrame mới chứa các dòng có cột smoke = 0
print(df_cohutthuoc)
```

	bwt	smoke
0	120	0
1	113	0
3	123	0
5	136	0
6	138	0
...
1230	132	0
1231	113	0
1232	128	0
1234	125	0
1235	117	0

[742 rows x 2 columns]

Tạo một mảng tên là **arr_cohutthuoc** lấy dữ liệu từ cột **bwt** của DataFrame **df_cohutthuoc**

```
# Lọc các dòng có cột smoke = 1
df_cohutthuoc = data[data['smoke'] == 1]

# Tạo mảng arr_cohutthuoc lấy dữ liệu từ cột bwt của df_cohutthuoc
arr_cohutthuoc = df_cohutthuoc['bwt'].to_numpy()

# In ra mảng
print(arr_cohutthuoc)
```

```
[128 108 143 144 141 110  92 115 119 115 103 114 114 134
 145 108 124 122 101 128 104 137 103 133  91 153  99 114
  87 120 107 119 103  91  95 141 100 115  94 101 112 128
 113 129 118 133 116 113 131 121 122 101 113 131  96 142
  98 150 119 101 113  97 115 121 117 110 130 140 111 154
 154 150  99 117 130  81 124 125 115 104 119 123 141 129
  98 136 121  91  85 106 109  98 101  71 124  93 101 100
 109 120 103 123 104 122 116 129 133 122 133 130 106 121
 107 129 145 102 129 135 104 126 127  98 131  99 115 102
 116 144 120 116 112 132 146 119 100 118 129 122 117 144
 109 102  99 128 101 109 117  88  95 119 127 107 126  98
 125 132  69 114 123 129 114 119 119 131 114 110 103 117
 108 123 113  93 130 111  97 107 105 133 161 115 127 128
 126  98 103 117 115 118 144  85 130 117 135 115 123 154
 108 103 127 107 106 152 136 123  93 109 120 129 125  96
  99  97 126 119 117 131 118 109 131 134 128  86 115 141]
```

Tạo một mảng tên là **arr_khonghutthuoc** lấy dữ liệu từ cột **bwt** của DataFrame **df_khonghutthuoc**

```
# Lọc các dòng không có cột smoke = 1
df_khonghutthuoc = data[data['smoke'] == 0]

# Tạo mảng arr_khonghutthuoc lấy dữ liệu từ cột bwt của df_khonghutthuoc
arr_khonghutthuoc = df_khonghutthuoc['bwt'].to_numpy()

# In ra mảng
print(arr_khonghutthuoc)
```

✓ 0.0s

```
[120 113 123 136 138 132 120 140 114 115 144 105 137 122 131 146 125 122
  93 130 119 113 134 107 128 129 110 111 155 110 122 115 102 143 146 124
 145 106  75 107 124  97 142 130 156 120 127 121 120 149 129 139 138 138
 131 128 134 114  92 135 125 128 105 119 116 133 155 126 129 137 125 134
 118 131 121 131 118 152 121 117 112 109 132 117 128 117 134 127 122 147
 120 144 136 102 126 126 115 127 119 123 105 134 144 111 125 135 134 129
 121 138 136 120 134 112 132 136  96 124 113 137 133 107 136 130  90 123
 137 101 142 124 151 109 131 127 117 150  85 128 105 107 119 134 117 115
  93 125  93 129 126  85 173 111 126 122 141 142 113 149 128 125 114 116
 125 110 138 142 102 140 133 127 152 143 131 113 131 148 137 117 115 132
 119 132  80 111 143 136 110 108 106 149 135 110 121 142 104 138 112 131
 116 140 120 139 131 111 110 105  93 104 120 118 114 116 129 107  88 122
 106 135 107 126 116 124 123  98 110 101  96 100 154 127 126 127 129 132
 127 145 136 121 121 120 118 127 132 102 118 102 163 132 116 138 139 132
 131 115 119 125 123 120 140 120 146 122 128 135 116 129 116 138 123 113
 132 120 114 130 142 127  85 123 112  78 107 136 100 123 124 104 133 118]
```

➤ **Đọc dữ liệu từ file excel vào DataFrame:**

Để đọc dữ liệu từ file excel vào DataFrame, dùng hàm
read_excel

```
import pandas as pd
import numpy as np
data = pd.read_excel('18_M&M.xls')
print(data)
```

	Red	Orange	Yellow	Brown	Blue	Green
0	0.751	0.735	0.883	0.696	0.881	0.925
1	0.841	0.895	0.769	0.876	0.863	0.914
2	0.856	0.865	0.859	0.855	0.775	0.881
3	0.799	0.864	0.784	0.806	0.854	0.865
4	0.966	0.852	0.824	0.840	0.810	0.865
5	0.859	0.866	0.858	0.868	0.858	1.015
6	0.857	0.859	0.848	0.859	0.818	0.876
7	0.942	0.838	0.851	0.982	0.868	0.809
8	0.873	0.863	NaN	NaN	0.803	0.865
9	0.809	0.888	NaN	NaN	0.932	0.848
10	0.890	0.925	NaN	NaN	0.842	0.940
11	0.878	0.793	NaN	NaN	0.832	0.833
12	0.905	0.977	NaN	NaN	0.807	0.845