Statistiques Descriptives
(Rappel – TD1)

Abdellah Madane

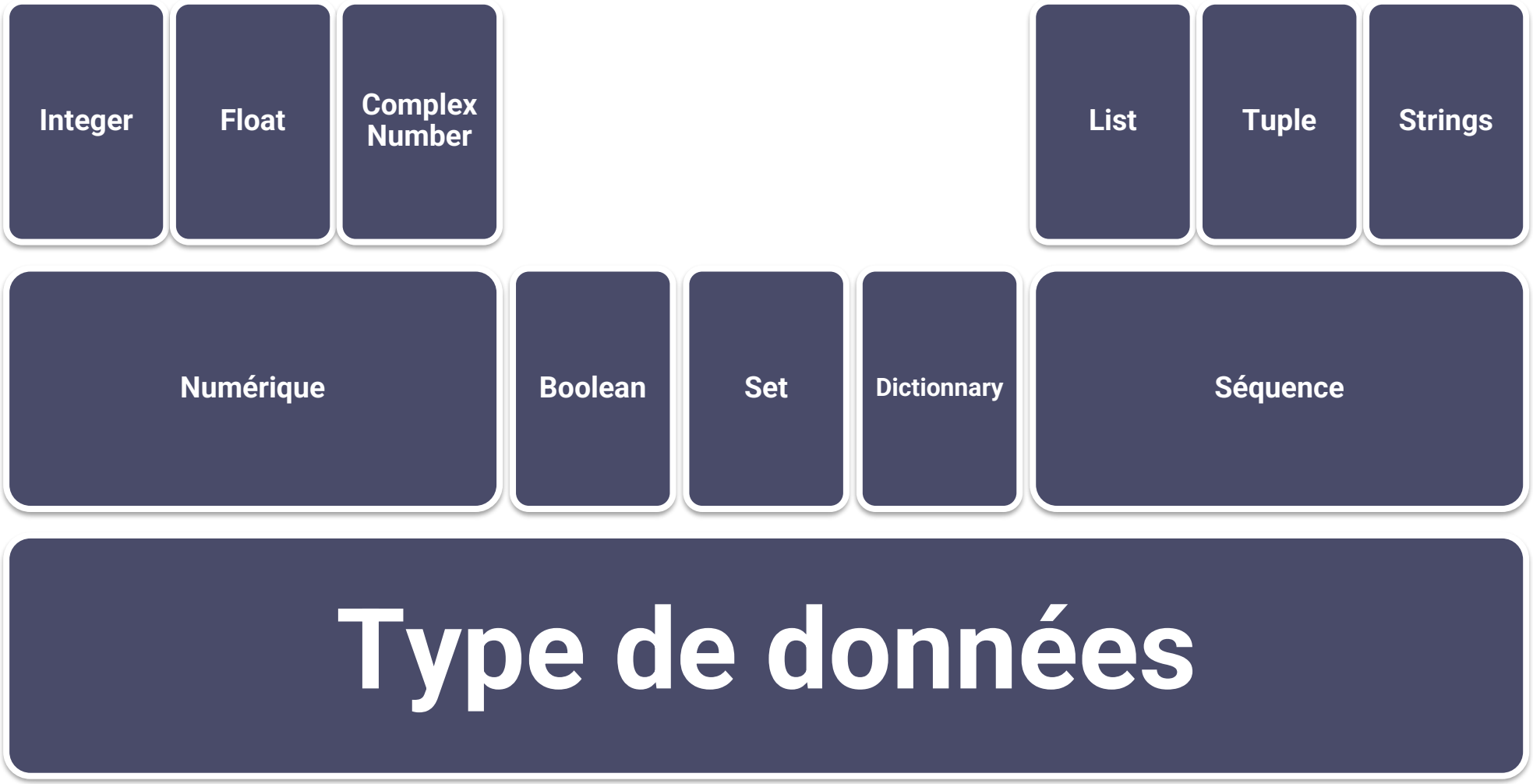
madane@lipn.univ-paris13.fr

2023/2024

Quick facts

- Python est un langage de programmation de **haut niveau**, **interprété**, avec une syntaxe très lisible.
- Milliers de librairies (appelées modules) disponibles pour à peu près n'importe quelle application
- Langage le plus utilisé pour l'analyse de données et l'apprentissage (aux côtés de R, Matlab, Scala...)
- Itératif, orienté objet.
- IPortable/multi-plateforme : ce qui signifie qu'il peut être exécuté sur différents systèmes d'exploitation tels que Windows, MacOS et Linux.
- Dynamiquement typé : ce qui signifie que le type des variables est déterminé à l'exécution et non déclaré explicitement.

Types de données



Principaux Opérateurs

Arithmétiques
+
-
*
**
/
//
%

Relationnels
==
!=
<
<=
>
>=

Booléens
and
or
not

Structures conditionnelles

SI	Si .... Sinon	Si .... Sinon si .... Sinon
if condition : instuction 1 instuction 2	**if condition :** instuction 1 instuction 2 **else:** instuction 1	**if condition :** instuction 1 instuction 2 **elif condition :** instuction 1 . . . . . . **else :** instuction 1

Structures conditionnelles

```python
nombre = 5  # Vous pouvez changer cette valeur pour tester d'autres cas

if nombre > 0:
    print("Le nombre est positif.")
elif nombre < 0:
    print("Le nombre est négatif.")
else:
    print("Le nombre est zéro.")
```

Les boucles

Les boucles sont des instructions qui permettent de répéter un bloc d'instructions plusieurs fois.

- **Boucle déterministe « for » :** répétition en précisant la valeur initale et la valeur finale d'increment.
- **Boucle non déterministe « while » :** Définition d'une condition d'arrêt pour interrompre la boucle.

Boucle « for »	Boucle « while »
for x in « suite de valeur » : instuction 1 instuction 2 . .	**while condition :** instuction 1 instuction 2 . .

Les boucles

Boucle « for » classique : Cette boucle est utilisée pour parcourir des éléments d'une séquence (comme une liste ou une chaîne).

```
nombres = [1, 2, 3, 4, 5]
for num in nombres:
    print(num)
```

Boucle « for » avec range : range() est souvent utilisé avec la boucle for pour répéter une action un certain nombre de fois.

```
for i in range(5):  # 0, 1, 2, 3, 4
    print(i)
```

Boucle while : Cette boucle continue de s'exécuter tant qu'une condition est vraie.

```
compteur = 0
while compteur < 5:
    print(compteur)
    compteur += 1
```

Boucle avec enumerate : enumerate() est utile lorsque vous souhaitez obtenir l'indice et la valeur de chaque élément d'une séquence.

```
mots = ["pomme", "banane", "cerise"]
for indice, mot in enumerate(mots):
    print(f"Indice: {indice}, Mot: {mot}")
```

Les fonctions

- Une fonction est **une tâche partielle** du programme globale qui peut être subdivisé en petites unités.
- Une fonction est **réutilisable**. Elle permet donc un gain de temps et sa réutilisation dans différents endroits du programme.
- Une fonction doit être définie avant son appel.

Structure d'une fonction

```
def NomFonction ( Les paramètres ) :
        Instruction 1
        Instruction 2
        .
        .
        .
```

- Les fonctions qui n'utilisent pas "return" sont des fonctions qui n'ont pas de valeur. Elles sont appelées aussi "procédure". Elles ne peuvent pas être utilisées dans une formule ou comme paramètre d'une autre fonction.

Les fonctions

```python
# Sans utiliser de fonctions:
# Calculer la surface d'un cercle de rayon 5
from math import pi
rayon = 5
aire_cercle_1 = pi * rayon * rayon

# Calculer la surface d'un cercle de rayon 10
rayon = 10
aire_cercle_2 = pi * rayon * rayon

print(aire_cercle_1, aire_cercle_2)
```

```python
# En utilisant une fonction:
from math import pi
def aire_cercle(rayon):
    return pi * rayon * rayon

# Calculer la surface de deux cercles de rayon 5 et 10 respectivement
aire_1 = aire_cercle(5)
aire_2 = aire_cercle(10)

print(aire_1, aire_2)
```

Les structures de données

Liste

- **Collection ordonnée et modifiable** d'éléments.
- 0 est l'indice du premier élément de la liste.
- -1 est l'indice du dernier élément de la liste.
- La fonction « len » permet de connaître la longueur d'une liste.

L = [22, "A", 0.7], len(L) = 3

- Des méthodes à savoir : L.copy() , L.append(x) , L.index(x), L.insert(i, x), L.pop(i), L.remove(x)

Les structures de données

Tuple

- **Collection ordonnée et non modifiable** d'éléments
- Structure de données non modifiable que ce soit les valeurs ou la taille.
- On peut dire qu'un tuple est une liste non modifiable.
- P = tuple(range(1,5)) -> p = (1, 2, 3, 4) , P[0] = 1
- L = list(range(1,5)) -> L = [1, 2, 3, 4]

Les structures de données

Ensemble

- **Collection non ordonnée et non indexée** d'éléments **uniques**.
- set **ou** {}
- E = set((1,2,3,2,5,3)) → E = {1,2,3,5}
- Parcours : for x in E
- Méthodes à savoir : E.add(x) , E.remove() , E.discard() , E.clear, E.isdisjoint(F)
- Opérateurs entre sets : **&** (intersection) , | (union), **-** (exclusion), **^** (Différence symétrique), <, >

.

Les structures de données

Dictionnaire

- **Collection non ordonnée, modifiable** de couples (clé non modifiable, valeur modifiable) permettant un accès à la valeur si on fournit la clé.
- On peut le voir comme une liste dans laquelle l'accès à un élément se fait par un code au lieu d'un indice.
- La clé est une chaîne de caractère mais la valeur est de n'importe quel type.
- D = {'clé1':valeur1, … , 'clé N' : valeur N }

```
dictionnaire = {"prénom": "Jean", "nom": "Dupont", "âge": 30}

fruits_quantitée = {"pommes": 10, "bananes": 5, "cerises": 7}
```

- Les méthodes à savoir : D.keys() , D.items() , D.values()

Les matrices

- En Python, une matrice peut être représentée comme une liste de listes, où chaque liste interne représente une ligne de la matrice. Voici un exemple simple de création d'une matrice 3x3 en Python :

```
matrice = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
```

Les modules/bibliothèques

Numpy

- Calcul sur des objets multidimensionnels : vecteurs, matrices, ...
- NumPy peut être utilisé pour créer un array (ndarrays) et manipuler les données à l'aide de plusieurs fonctions mathématiques.
- Contrairement aux listes Python, les éléments d'un array sont homogènes.
- Utilisation similaire à Matlab
- Utilise des libraires en C compilées d'algèbre linéaire performant
- Compatible avec de très nombreuse libraires de data science/machine learning/deep learning : pandas, scikit-learn, Tensorflow, MXNet, etc.

Les modules/bibliothèques

Pandas

- La bibliothèque pandas est utile pour traiter les données structurées comme les données stockées dans des tableaux, comme les fichiers CSV, les feuilles de calcul Excel ou les tableaux de bases de données.
- Les données sont traitées sous forme d'une abstraction appelée DataFrame : données organisées par colonnes nommées (table relationnelle).
- Une donnée sous forme de colonne ou ligne (vecteur) a pour type ‘Series’.
- https://pandas.pydata.org/pandas-docs/stable/reference/frame.html
- https://pandas.pydata.org/pandas-docs/stable/reference/series.html

Les modules/bibliothèques

Pandas

Columns (Axis 1)

	t	EGT_SEL	FLIGHT_MOD	FMV_SEL	HPTC_SEL	LPTC_SEL	N1_SEL	N2_ACTSEL	OIL_P	OIL_TEMP	...	T25_SEL	T3_SEL	VBV_SEL	VIB_CN1	VIB_CN2	VIB_TN1	VIB_
1	15/09/2011 14:25:58.125	NaN	NaN	NaN	NaN	NaN	NaN	0	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
2	15/09/2011 14:25:58.375	NaN	NaN	NaN	NaN	NaN	NaN	0	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
3	15/09/2011 14:25:58.625	NaN	NaN	NaN	NaN	NaN	NaN	0	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
4	15/09/2011 14:25:58.875	NaN	NaN	NaN	NaN	NaN	NaN	0	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
5	15/09/2011 14:25:59.125	NaN	NaN	NaN	NaN	NaN	0	0	647.998	NaN	...	NaN	NaN	0	0	0	0	

Index

5 rows × 26 columns

Valeurs manquantes

Données

Rows (Axis 0)