



# Utiliser R et Python pour le traitement de données :

## exploration des avantages de Python en matière de visualisation

Rencontres R 2023

Mickaël Carlos

[mickael.carlos@makina-corpus.com](mailto:mickael.carlos@makina-corpus.com)

# Conclusion



Pas spécialement d'avantage net en terme de visualisation,  
Python et R c'est trop bien!!!



Merci pour votre attention, des questions ?

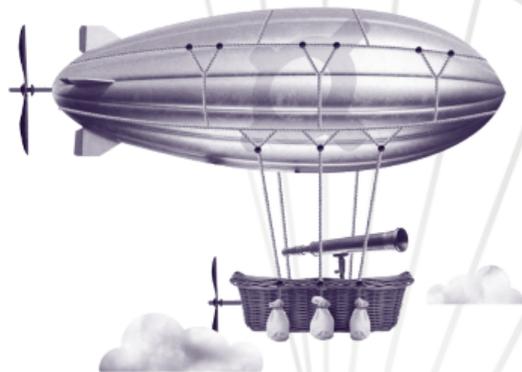
# Présentation : Mickaël Carlos

- Data scientist
- Developpeur Python/Django/IA
- Notebookiste (Jupyter Notebook)
- ( Docteur en Astro-physique, Astro-chimie )



# Présentation : Makina Corpus

- Société de services numériques
- Applications innovantes en utilisant des logiciels libres et données ouvertes
- Dev App Web
- Formation
- Audit
- Expertise : SIG, Data science, Drupal, Gestion de l'eau, App mobile, Python/Django



# Historique des faits

- Fortran (Calcul tensoriel de physique et chimie quantique)
- Matlab (ETL) Physique quantique et Astrophysique
- Python (Un peu de tout)
  - Calcul
  - Visualisation
  - Application Web
  - Machine learning et Deep Learning

# Qu'est-ce que je fais là...

- Les conférences scientifiques me manquent!!!
- Makina Corpus est sponsor des Rencontres R.
- Je vais vous parler de ce que j'aime en python.

# Que choisir pour développer ?

⇒ Python et pourquoi :

- Haut niveau et proche de « langage parler » et facilement partageable
- Open source ( ≠ Matlab)
- Très grosse communauté ⇒ Grande chance de pouvoir s'inspirer, utiliser, modifier, optimiser un script déjà existant...
- Les possibilités en IA!!!

⇒ Autres possibilités :

- Industrialiser un workflow
- Créer des interfaces plutôt ergonomique
- Faciliter de connexion avec le reste du monde informatique

# Que développer en Python ?

- Pas tellement de limite dans ce que l'on peut développer en Python.
- Grâce à la communauté de plus en plus de bibliothèques permettent de plus en plus de choses dans tous les domaines, même les domaines niches...
- Scipy-stack, Scikit (learn, image, network), Django, Flask, Pytorch, Tensorflow (Keras), Shapely, Geopandas, Astropy... Parmi les plus connus
- Mais aussi Faker, Wikipedia, Beautiful Soup, ...
- Lorsque c'est nécessaire pour le calcul certaines bibliothèques sont implémentées en C.

# Historique des faits

Mon choix d'un langage de programmation : **Une passion opportuniste**

# Par quoi commencer???

- Personne ne commence à coder en se laissant porter par le vent...
- Tutoriel
- Choisir souvent une distribution Python
- Anaconda (Python et R)

# Anaconda

- Interface ET ligne de commande...
- Plusieurs IDE, (Pycharm, VScode, Rstudio,...)
- Jupyter Notebook
- Conda

# Conda : Qu'est-ce que c'est ?

- Gestionnaire de bibliothèque(s), d'environnement(s) virtuel(s)

Installation classique de bibliothèque python

« pip install bibli »

Avec conda

« conda install bibli »

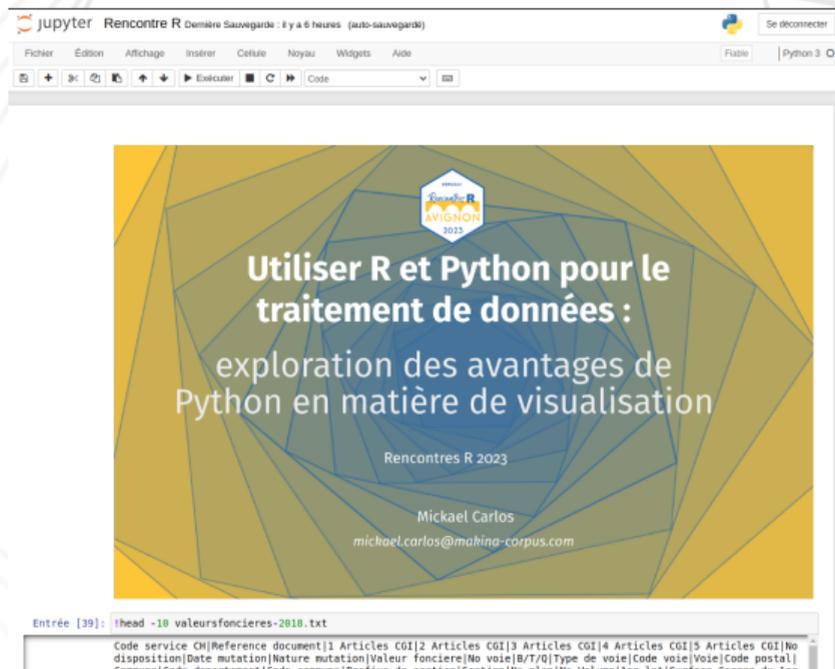
# Pour conda alors ?

- Packages installés par conda sont optimisés pour s'adapter aux mieux à notre machine.
- Amélioration des performances.
- Unité des dependances d'un projet

# Jupyter Notebook?

- Permet de coder dans plusieurs langages :
    - Python
    - R
    - Julia
    - C++
  - Permet d'enrober le code dans du texte quelque peu formaté si écrit en Markdown
  - on peut y insérer du Latex
  - ...
- Installation d'Anaconda

# Jupyter Notebook



The screenshot shows a Jupyter Notebook window titled "Rencontre R Dernière Sauvegarde : il y a 6 heures (auto-sauvegardé)". The interface includes a menu bar with options like "Fichier", "Édition", "Affichage", "Insérer", "Cellule", "Noyau", "Widgets", and "Aide". Below the menu is a toolbar with icons for file operations and a dropdown menu set to "Code".

The main content area displays a presentation slide with a yellow and blue geometric background. The slide text reads:

**Utiliser R et Python pour le traitement de données :**  
exploration des avantages de Python en matière de visualisation

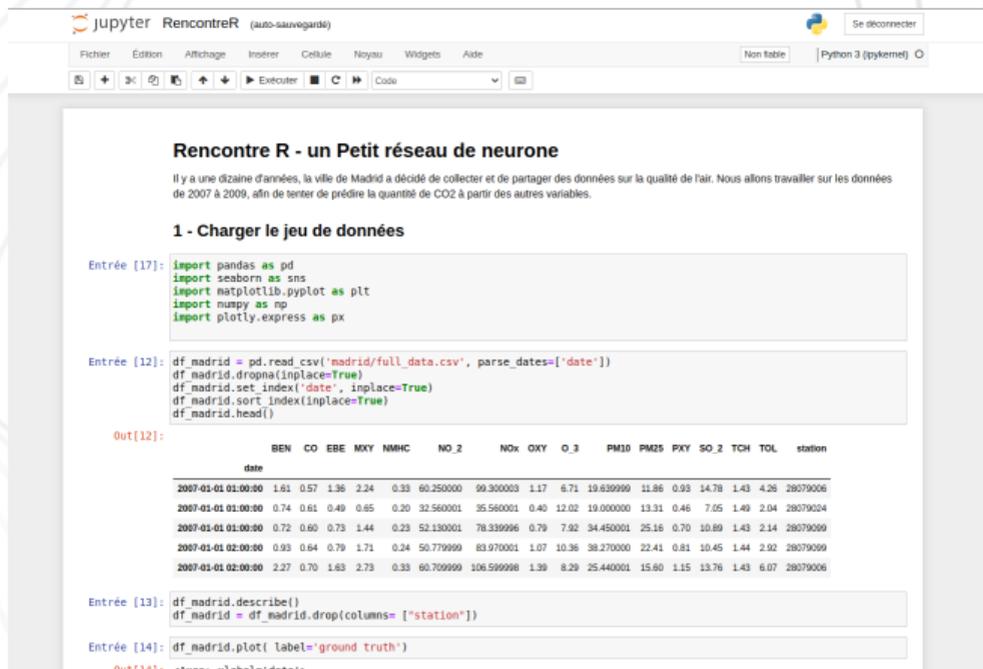
Rencontres R 2023

Mickaël Carlos  
mickaël.carlos@makina-corpus.com

Below the slide, a terminal window shows the command `!head -10 valeursfoncieres-2018.txt` and its output, which is a header line from a CSV file:

```
Code service CH|Reference document|1 Articles CGI|2 Articles CGI|3 Articles CGI|4 Articles CGI|5 Articles CGI|No disposition|Date mutation|Nature mutation|Valeur fonciere|No voie|B/T/O|Type de voie|Code voie|Voie|Code postal|Commune|Code departement|Code commune|Prefixe de section|Section|No plan|No Volume|Etat|Surface Carrez du Ter
```

# Jupyter Notebook : Mon Notebook



Jupyter RencontreR (auto-sauvegardé) Se déconnecter

Fichier Édition Affichage Insérer Cellule Noyau Widgets Aide Non fiable Python 3 (ipykernel) Q

↳ ↶ ↷ ▶ Exécuter ⏏ ⏪ ⏩ Code ⏏

## Rencontre R - un Petit réseau de neurone

Il y a une dizaine d'années, la ville de Madrid a décidé de collecter et de partager des données sur la qualité de l'air. Nous allons travailler sur les données de 2007 à 2009, afin de tenter de prédire la quantité de CO2 à partir des autres variables.

### 1 - Charger le jeu de données

Entrée [17]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import plotly.express as px
```

Entrée [12]:

```
df_madrid = pd.read_csv('madrid/full_data.csv', parse_dates=['date'])
df_madrid.dropna(inplace=True)
df_madrid.set_index('date', inplace=True)
df_madrid.sort_index(inplace=True)
df_madrid.head()
```

Out[12]:

	BEN	CO	EBE	MXV	NMHC	NO_2	NOx	OXY	O_3	PM10	PM25	PXY	SO_2	TCH	TOL	station
2007-01-01 01:00:00	1.61	0.57	1.36	2.24	0.33	60.250000	99.300000	1.17	6.71	19.639999	11.86	0.93	14.78	1.43	4.26	28079006
2007-01-01 01:00:00	0.74	0.61	0.49	0.65	0.20	32.560001	35.560001	0.40	12.02	19.000000	13.31	0.46	7.05	1.49	2.04	28079024
2007-01-01 01:00:00	0.72	0.60	0.73	1.44	0.23	52.130001	78.336996	0.79	7.92	34.450001	25.16	0.70	10.89	1.43	2.14	28079099
2007-01-01 02:00:00	0.93	0.64	0.79	1.71	0.24	50.779999	83.970001	1.07	10.36	38.270000	22.41	0.81	10.45	1.44	2.92	28079099
2007-01-01 02:00:00	2.27	0.70	1.63	2.73	0.33	60.709999	106.599998	1.39	8.29	25.440001	15.60	1.15	13.76	1.43	6.07	28079006

Entrée [13]:

```
df_madrid.describe()
df_madrid = df_madrid.drop(columns=["station"])
```

Entrée [14]:

```
df_madrid.plot(label='ground truth')
```

Out[14]:

```
df_madrid.plot(label='data')
```

# Visualisation : Cartographie

Même en Cartographie nous avons la chance de pouvoir compter sur une pléthore de librairie de vont de la librairie statique avec Matplotlib/Cartopy, aux interfaces dynamiques et interactives avec Plotly, Keplergl, Folium, ...

### Définition de la fonction

```
1 plt.figure(figsize=(10,10)) # créer une figure
2 ax = plt.axes(projection=ccrs.PlateCarree())
3 # choisir un type de projection
4 ax.set_extent([-10, 30, 30, 70])
5 ax.coastlines() # afficher les lignes de cotes
6
7 plt.show() # afficher la carte
```

# Visualisation

Cartographie : Matplotlib/Cartopy



# Visualisation

Cartographie : Matplotlib/Cartopy



# Visualisation

Cartographie : Matplotlib/Cartopy



Basé sur la librairie Javascript Leaflet, Folium permet une interaction plus poussée avec la carte avec notamment la possibilité de « Dessiner » sur la carte.

```
tileurl = 'https://api.mapbox.com/v4/mapbox.satellite/{z}/{x}/{y}@2x.png?access_token=' + str(token)
token = "*****"

m = folium.Map(
    location=[43.5749251, 1.4083081],
    zoom_start=5,
    max_zoom=22,
    tiles=tileurl, # 'OpenStreetMap', # tiles
    attr='MapBox'
)

fgs = folium.FeatureGroup(name="Markers", control=True)
for stations in stations_positions.itertuples():
    popup_text = f"Station de {stations.nom}"
    marker = folium.Marker(
        location=[stations.geometry.coords[0][1], stations.geometry.coords[0][0]],
        popup=popup_text).add_to(m)

Draw(export=True).add_to(m)
m
```

# Visualisation

## Cartographie : Folium

Basé sur la librairie Javascript Leaflet, Folium permet une interaction plus poussée avec la carte avec notamment la possibilité de « Dessiner » sur la carte.



# Interfaçage

Pour aller encore plus loin dans la visualisation et la personnalisation, Python dispose de nombreuses bibliothèques comme Streamlit qui permettent de faire des interfaces entières à partir de code Python.

En reprenant le dernier exemple avec des stations météo on pourrait rapidement construire une interface qui permettent de récupérer la position Gps d'une adresse (Geocoding) pour ensuite calculer quel est la station Météo la plus proche.

Mon Streamlit

# Conclusion

Deep Learning, Application Web complexe (gestion d'utilisateur etc...)

Pas spécialement d'avantage net en terme de visualisation

Python et R c'est trop bien!!!

# Merci pour votre attention