



Projet Python :

Les arbres de Grenoble



L'emblématique arbre de Venon

Source : francebleu.fr

Objectifs du module

L'objectif de ce projet est de vous plonger, tête la première dans de l'analyse de données, sans aucun pré-requis technique. A l'issue de ce projet, vous aurez vu les bases de la **manipulation de données grâce à Python**.

Nous sommes conscients qu'il existe parmi vous une disparité de connaissances relatives au monde informatique, à la programmation et à l'analyse de données. Certains d'entre vous ont probablement déjà codé (régulièrement ou non) au cours de leur vie personnelle ou professionnelle (python, matlab, R ou autre) alors que d'autres découvrent ce monde aujourd'hui.

Il est donc tout à fait normal que le projet à suivre paraisse facile à certaines personnes et difficile pour d'autres. Tant pis, et tant mieux ! Nous vous encourageons autant que possible à vous entraider, à transmettre, et à collaborer. Prenez votre temps au Campus Numérique comme un **espace disponible vous permettant de vous auto-former et d'expérimenter à plusieurs**. N'hésitez pas à approfondir vos connaissances et à aller chercher plus loin que ce qui est strictement attendu de vous si vous en sentez le besoin ou le désir.

Le rôle des formateurs au cours de ce projet (et de la formation de manière plus générale) sera celui d'**accompagnateur méthodologique**. Nous n'aurons probablement pas la réponse à toutes vos questions.

Modalités

- Durée du projet : 5 jours.
- Ce projet sera réalisé en autonomie
- L'entraide est (très) fortement encouragée !

Description du projet

- Ce projet s'insère dans le cadre de l'**Open Data**, et en particulier des données libérées par la Ville de Grenoble. Ce jeu de données contient des informations relatives aux arbres situés sur le territoire de la ville : numéro d'identification, date de plantation, genre botanique, espèce, etc. Ce jeu de données est disponible à cette adresse : <http://data.metropolegrenoble.fr/ckan/dataset/les-arbres-de-grenoble>
- Vous explorerez les données via un **Jupyter notebook**, particulièrement adapté pour le calcul scientifique et la Data Science. Jupyter est un projet open-source et gratuit.



Installation de l'environnement Anaconda

Objectifs de l'activité

- Mettre en place l'environnement de développement scientifique avec Anaconda



ANACONDA DISTRIBUTION STARTER GUIDE

See full documentation for Anaconda Distribution
docs.anaconda.com/anaconda/

BEFORE YOU START

**Why do I need
Anaconda Distribution?**

Installing Python in a terminal is no joy. Many scientific packages require a specific version of Python to run, and it's difficult to keep them from interacting with each other. It is even harder to keep them updated. Anaconda Distribution makes getting and maintaining these packages quick and easy.

**What is
Anaconda Distribution?**

It is an open source, easy-to-install high performance Python and R distribution, with the conda package and environment manager and collection of 1,000+ open source packages with free community support.

Then what is Miniconda?

It's Anaconda Distribution without the collection of 1,000+ open source packages. With Miniconda you install only the packages you want with the conda command,

```
conda install PACKAGENAME
```

Example: `conda install anaconda-navigator`

Consignes

- Pour Linux :
 - Télécharger <https://www.anaconda.com/distribution/#linux>
 - Choisir Python 3.7 => 64-bit (x86) installer (506MB)
 - Consignes d'installation :
<https://docs.anaconda.com/anaconda/install/linux/>
 - Pour ouvrir un terminal à l'endroit où vous avez téléchargé Anaconda
 - Click droit => Open Terminal Here
 - Vous pouvez alors exécuter les instructions de la doc
- Pour Windows
 - Idem mais avec l'interface graphique !

Itération 1

Exploration préliminaire du dataset "Arbres"

Objectifs de l'activité

- Installer Anaconda

- Nous nous Intéresserons dans un premier temps à quelques **structures de données internes à Python** - *built-in data structures* - sans ajout de librairie spécialisée (type *pandas*) et aux opérations de bases (**boucles** et **conditions**)
- Ouverture et lecture du contenu d'un fichier ".csv "
- Exploration préliminaire des données.

Compétences

- Ouvrir et lire le contenu d'un fichier .csv
- Utiliser des boucles for
- Utiliser des structures conditionnelles
- Manipuler des listes et tuples

Consignes

1. Avant d'explorer les données, pensez à **ranger votre arborescence de fichier**. Vous allez très rapidement accumuler de nombreux dossiers et fichiers, pas question de tout mettre en b@r#*l sur le bureau ! Vous y perdriez beaucoup de temps et d'énergie assez rapidement. Au sein de votre répertoire de travail, créez un dossier spécifique à ce projet ayant une structure :

```

Projet_arbre
|_ data
    |_ arbres.csv
    |_ notebooks
        |_ arbres.nbpy
  
```

2. Avant même de commencer à coder, il est très important d'observer les données. Ouvrez donc vos données dans un éditeur de texte (vscode, sublime text, etc.) ou même dans le *bash* (*cat* ou *nano* par exemple). Ou même dans Excel 2000 ! Regardez les données à traiter droit dans les yeux !
3. A l'aide du notebook, vous écrirez maintenant du code (uniquement à l'aide du module csv de la librairie standard dans un premier temps) permettant de :
 - ☐ Lire les deux premières ligne du csv des arbres de Grenoble (module csv, voir doc officielle). Que représentent elles ?
 - ☐ Extraire l'information concernant l'année de plantation pour l'ensemble des lignes (*rows*) et stocker ces informations dans une variable de type *List*
 - ☐ Afficher les 50 premières lignes et les 50 dernières lignes de cette liste.
 - ☐ Pour combien d'arbres manque-t-il l'information concernant la date de plantation ?
 - ☐ Extraire l'ensemble des lignes (*rows*) et stocker ces informations dans une variable de type *List*
 - ☐ Combien d'arbres sont recensés dans ce jeu de données ?
 - ☐ Combien d'arbres ont été plantés en moyenne chaque année ?
 - ☐ Afficher tous les arbres ayant été plantés l'année de votre naissance.
 - ☐ Combien d'arbres ont été plantés depuis votre naissance ?
 - ☐ Quel Maire a planté le plus d'arbres à Grenoble ?
 - ☐ Utilisez Matplotlib pour tracer l'histogramme représentant le nombre d'arbres plantés par année.

- ❑ Récupérez maintenant l'information concernant le genre botanique et la stocker conjointement avec l'année de plantation dans un tuple. Les tuples seront stockés dans une liste.
- ❑ Pour combien d'arbres manque-t-il l'information concernant le genre botanique ?
- ❑ Triez les tuples en fonction de l'année de plantation par ordre décroissant.

Note sur les ressources :

Les ressources que vous trouverez sur Internet concernant Python (et plus largement sur les langages de programmation) sont de plusieurs types : **tutoriels** (écrits, vidéos, livres), **how-to** (orientés sur un but précis) et **forums** (stackoverflow par exemple) et **documentation officielle**. Vous serez régulièrement amenés à jongler entre toutes ces sources d'information. Ne négligez pas la documentation officielle! Elle est souvent aride, mais elle est exhaustive. Il est important de s'y confronter régulièrement pour finir par la dompter.

Ressources

- Documentation officielle Python 3
 - <https://docs.python.org/3/library/csv.html>
 - <https://docs.python.org/3/howto/sorting.html>
 - <https://docs.python.org/3/tutorial/introduction.html#lists>
- Documentation W3C
 - https://www.w3schools.com/python/python_datatypes.asp
- Tutoriel par Google :
 - <https://developers.google.com/edu/python>
- Documentation Matplotlib :
 - <https://matplotlib.org/tutorials/index.html>
- Un peu de biologie (désolé pour les vrais biologistes...) :
 - [https://fr.wikipedia.org/wiki/Genre_\(biologie\)](https://fr.wikipedia.org/wiki/Genre_(biologie))

Interlude

The Weird Trees Game

Un second fichier recensant les mêmes arbres a été découvert. Celui-ci s'appelle "weird_trees.csv". Ce fichier a été corrompu de manière aléatoire.



Pour des raisons obscures qui nous échappent - probablement politiques - nous sommes missionnés pour nous assurer que ce fichier redevienne strictement identique au fichier original. Nous ne pouvons pas simplement l'effacer et faire une copie du fichier original.

Repérez les corruptions et réparez !

Ressources

- W3C
 - https://www.w3schools.com/python/python_ref_string.asp

Avertissement préliminaire



Les métiers de Data Analyst et Data Scientist visent à recueillir des données, les traiter et s'en servir pour extraire une information "métier" pertinente.

Le projet sur lequel nous travaillons a un intérêt en terme pédagogique (python) et également un potentiel intérêt réel à l'échelle du territoire de la métropole grenobloise. Cependant, il est important de garder en tête que sans les compétences métiers associées au projet - dans notre cas celles des urbanistes, pépiniéristes, pédologues et biologistes - les analyses effectuées n'auront que peu d'intérêt réel.

LISTE DES ESPECES SUJETTES AUX PATHOGENES

Genre, espèce	Argumentaire
Aesculus	Maladies : chancre bactérien, mineuse
Fraxinus	Maladie : la chalarose
Platanus	Maladie : chancre coloré

LISTE DES ESPECES INVASIVES

A éviter absolument

Genre, espèce
Ailanthus altissima
Robinia pseudoacacia
Acer negundo

LISTE DES ESPECES SUR REPRESENTEES A L'ECHELLE DU TERRITOIRE

Non souhaités par la Métropole dans un soucis de diversification des espèces

Genre, espèce	Argumentaire supplémentaire
Platanus x acerifolia	Chancre coloré (toutes espèces confondues)
Acer platanoides et pseudoplatanus	Sensible aux modifications climatique à l'échelle de la métropole

LISTE DES GENRES ALLERGENES PRINCIPAUX

Espèces anémophiles à éviter de planter à proximité de zones très fréquentées

Genre
Betula
Castanea
Cupressus
Fraxinus
Olea
Platanus