

# Introduction à numpy & pandas

Pourquoi le python, pourquoi ces bibliothèques spécifiques?

# Un historique mouvant

- Fortran (70-90)
- C, C++ (80-...)
- Matlab
- Python (2010, ...)
  - Mais un python en évolution

Une **opportunité** incroyable pour les sciences des données !

- Outils matures & unifiés

# De la programmation orientée *data*

Numpy = un environnement à l'intérieur de python

1. **numpy**: prise en main des outils
  - Cas d'usage : classification bayésienne
  - descente de gradient
2. **Pandas : lien avec les API**

# Passer à un nouveau langage

- Cout faible
  - une fois que vous avez compris la logique générale
- Cout non négligeable:
  - comprendre les forces et les faiblesses du langage
  - Adapter sa manière de programmer (e.g. calculer un décile)

# Pourquoi numpy

- Pour la richesse des opérateurs
  - Développés en C et rapides
- Pour les interfaces riches et efficaces
  - Matplotlib, panda
  - Scikit-learn
- Pour la préparation que ça représente  $\Leftrightarrow$  pytorch, tensorflow

# Le but du cours

- Comprendre la philosophie de l'outil
- Acquérir les bons reflexes
- Consolider vos acquis en python
- Introduire des données dans la programmation avec pandas

# Au-delà du cours

- Premier classifieur
- Descente de gradient
- Analyse de séries temporelles

# Le programme

- Lancement d'un notebook
- Vendredi = Tutos numpy: 1 à 5
  - Opt: classification bayesienne
- Samedi = Pandas + prise en main de l'évaluation
  - Opt: classification bayesienne / gradient

Note: les séries temporelles NE sont VRAIMENT PAS au programme !