

Certificat de spécialisation Analyste de données massives UESTA211 Entreposage et fouille de données

5ème séance

Emmanuel Jakobowicz

ej@stat4decision.com

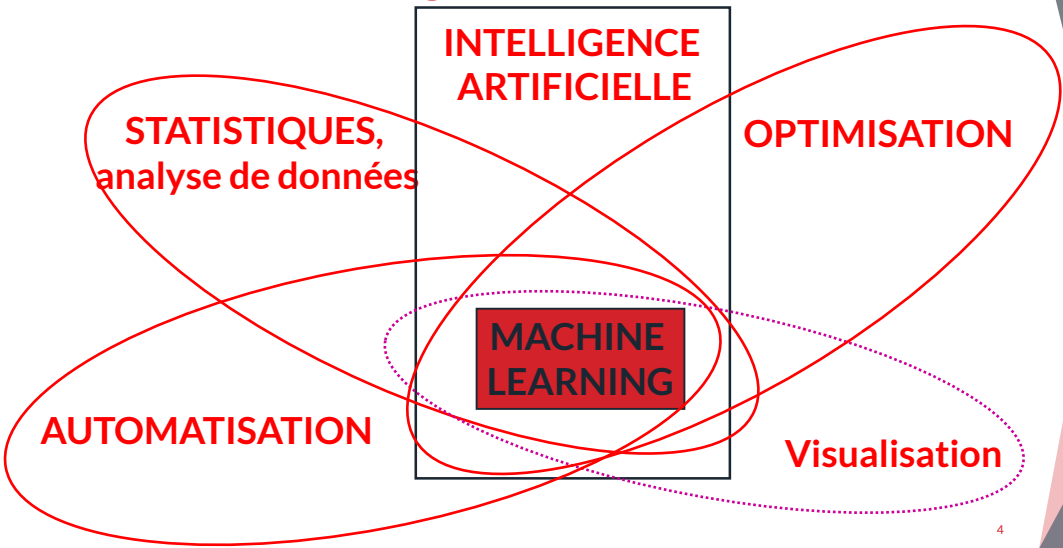
Objectif de la session de 2 jours 23-24 novembre 2017

- ▶ Approfondir les notions d'apprentissage (machine learning)
- ▶ Faire un point sur les outils (R et python)
- ▶ L'apprentissage non supervisé en détails
- ▶ L'apprentissage supervisé
 - ▶ Les principes
 - ▶ Les méthodes de régression
- ▶ Des mises en œuvres sur des données

LE MACHINE LEARNING

3

Le machine learning



4

Définitions

- ▶ Le **machine learning** est un ensemble d'outils statistiques et géométriques et d'algorithmes informatiques qui permettent d'automatiser la construction d'une fonction de prédiction f à partir d'un ensemble d'observations
- ▶ Un **modèle de machine learning** est un procédé algorithmique spécifique qui permet de construire une fonction de prédiction f à partir d'un jeu de données d'apprentissage.
- ▶ La construction de f constitue l'**apprentissage** du modèle
- ▶ Une **prédiction** correspond à l'évaluation de f sur les variables prédictives de x

5

Exemples d'applications

- ▶ Fraud detection.
- ▶ Web search results.
- ▶ Real-time ads on web pages
- ▶ Credit scoring and next-best offers.
- ▶ Prediction of equipment failures.
- ▶ New pricing models.
- ▶ Network intrusion detection.
- ▶ Recommendation Engines
- ▶ Customer Segmentation
- ▶ Text Sentiment Analysis
- ▶ Predicting Customer Churn
- ▶ Pattern and image recognition.
- ▶ Email spam filtering.
- ▶ Financial Modeling
- ▶ ...

Un bon algorithme, c'est quoi ?

- ▶ Déployabilité
- ▶ Robustesse
- ▶ Transparence
- ▶ Proportionnalité
 - ▶ Bénéfice proportionnel au coût de mise en place
- ▶ Performance

Qu'est-ce que le machine learning ?

Définition

- ▶ Le Machine learning c'est la capacité d'apprentissage d'une machine. On l'appelle aussi apprentissage automatique.
- ▶ Le domaine du machine learning consiste en la construction de programmes informatiques qui s'améliorent grâce à l'expérience

9

Définition de l'apprentissage

- ▶ Un algorithme apprend d'une expérience E pour effectuer des actions T dont les performance seront mesurées par P si sa performance mesurée par P s'améliore avec l'expérience E.
- ▶ Imaginons une voiture sans chauffeur :
 - ▶ T : conduire sur une route en utilisant des cameras
 - ▶ P : distance parcourue avant une erreur
 - ▶ E : des images enregistrées lors de l'observation d'un conducteur humain
- ▶ **Exercice** : Définissez T, P et E pour un outil de recommandation, pour un analyse de churn



Les problèmes liés au machine learning

- ▶ Quel algorithme choisir suivant le problème traité ?
- ▶ Quels paramètres expérimentaux choisir afin d'obtenir des résultats satisfaisants ?
- ▶ Combien faut-il d'observations pour l'apprentissage ?
- ▶ Comment et quand ajouter des connaissances a priori dans le modèle ?
- ▶ Comment réduire l'apprentissage a une seule tâche et à un seul problème d'optimisation pour des cas complexes ?
- ▶ ...

11

Les types d'algorithmes de machine learning

- ▶ 3 types :
 - ▶ Apprentissage supervisé
 - ▶ Une cible / variable dépendante
 - ▶ Des prédicteurs
 - ▶ Régression, Decision Tree, Random Forest, KNN, Logistic Regression etc.
 - ▶ Apprentissage non supervisé
 - ▶ Pas de cible à prédire
 - ▶ On rassemble des populations en différents groups / on rassemble les variables en différents groups...
 - ▶ Règles d'association, k-means, ACP...
 - ▶ Apprentissage par renforcement
 - ▶ L'apprentissage permet d'apprendre des décisions spécifiques afin d'arriver à un objectif après plusieurs décisions
 - ▶ Markov Decision Process
 - ▶ Beaucoup de recherches actuellement dans ce domaine

12

Les étapes d'un projet de machine learning

- Dans un projet machine learning / data science, on se concentre généralement sur 5 étapes :
 1. Traduire le problème métier en problème data science
 2. Identifier les données
 3. Préparer les données
 4. Choisir, appliquer et valider un modèle de machine learning
 5. Transposer les résultats en des décisions métiers

Traduire un problème métier en problème de data science

- ▶ C'est une étape centrale
- ▶ C'est la principale source d'incompréhension et de deception
- ▶ La data scientist doit être honnête avec le décideur
- ▶ A cette étape, on identifie aussi les sorties nécessaires à la resolution du problème métier

15

Identification des données

- ▶ Les données sont partout ;-)
- ▶ Malgré l'essor du big data, c'est pas si facile de récupérer des données
- ▶ Les sources sont très variées mais les algorithmes ne traitent pas n'importe quel type de données
- ▶ Première étape :
 - ▶ Identifier les sources (web, CRM, enquêtes, IoT...)
 - ▶ Identifier les formats (valeurs, textes, images articles....)
- ▶ Seconde étape
 - ▶ Le stockage des données
 - ▶ Est-ce que les données doivent être stockées ?
 - ▶ Oui : comment ? (data lake, SQL, Hadoop...)
 - ▶ Non : utilisation d'API ou de scrapping

16

Préparation des données

- ▶ C'est l'étape la plus longue
- ▶ Attention GIGO (garbage in, garbage out)
- ▶ La majorité des algorithmes demandent des données structurées en entrée
- ▶ Première étape :
 - ▶ Structurer les données
- ▶ Deuxième étape :
 - ▶ Transformer les données structurées de manière à ce qu'elles s'adaptent à la modélisation choisie

17

Modéliser

1. Comprendre de manière approfondie le problème et les données
2. Laissez le problème guider la modélisation (sélection d'outils, préparation des données...)
3. Vérifier les hypothèses
4. Améliorer le modèle de manière itérative
5. Faire en sorte que le modèle soit le plus simple possible
6. Rechercher les instabilités dans le modèle

18

Transposer les résultats en décisions

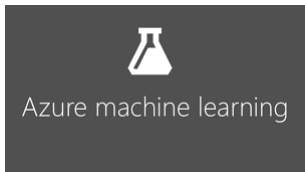
- ▶ Cette étape depend de l'objectif visé :
 - ▶ Prévission automatique : automatique
 - ▶ Compréhension du modèle : Demende la comprehension des résultats et leur traduction en actions
 - ▶ En general, on se trouve dans les deux cas simultanément.

19

Les outils

20

Une offre très large



21

Nous nous concentrons sur deux langages

► R

- Pour un cours introductif :
- <https://www.datacamp.com/courses/free-introduction-to-r>

► Python

- Pour un cours introductif :
- <https://www.datacamp.com/courses/intro-to-python-for-data-science>

22

- ▶ Question centrale de la communauté des data scientists / data miners

Python est un langage créé
par des informaticiens
adapté à la data science

R est un langage créé par
des statisticiens pour des
statisticiens

23

Python vs. R

Python

- ▶ Productivité et lisibilité
- ▶ Programmation et débogage simplifiés
- ▶ Simple de créer des fonctions et d'interfacer
- ▶ Les bibliothèques sont dans l'index PyPi
- ▶ Utilisation de R dans Python avec RPy2

R

- ▶ Analyse de données et graphiques
- ▶ Simplification des fonctions pour les modèles stat
- ▶ Apprentissage initial difficile puis plus simple
- ▶ Nombre de packages impressionnant dans le CRAN
- ▶ Utilisation de Python dans R avec rPython

24

Python vs. R

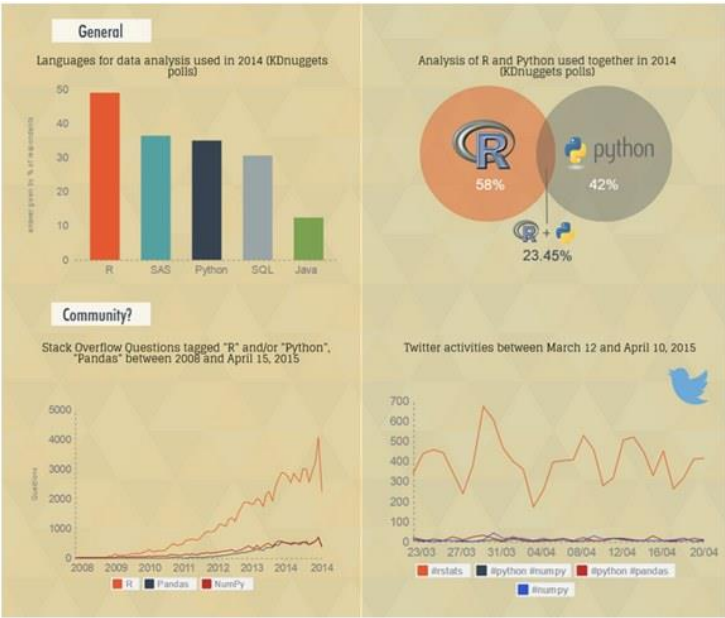
Python

- ▶ Utilisé surtout dans le cadre d'application intégrées dans des apps ou sur des serveurs
- ▶ Malgré des débuts difficiles, Python permet de gérer les données de manière efficace grâce à NumPy ou pandas
- ▶ Pour les graphiques, il existe de bonnes bibliothèques

R

- ▶ Utilisé pour du data mining au quotidien
- ▶ Pour la gestion des données, R est très doué
- ▶ Les graphiques sont la force de R

25



La question à se poser :
Quel objectif ?
Quelles compétences ?

26

Quand utiliser R ? Quand utiliser python ?

- ▶ On peut aboutir aux mêmes résultats avec R et python
- ▶ R va être efficace pour faire des analyses ponctuelles
- ▶ Python va être efficace pour créer des POC, automatiser des traitements
- ▶ Le data scientist se doit d'avoir une certaine connaissance de ces deux outils

27

Qu'est-ce que R ?

- ▶ R est un logiciel de développement scientifique spécialisé dans le calcul et l'analyse statistique
- ▶ R est un langage
- ▶ R est un environnement
- ▶ R est un projet open source
- ▶ R est un logiciel multi-plateforme (Linux, MAC, Windows...)

28

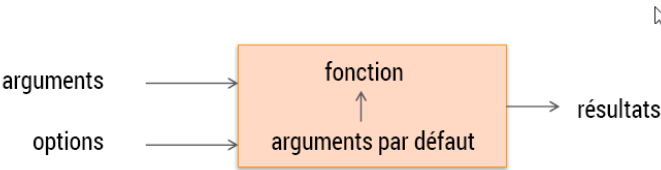
Principales fonctionnalités

- ▶ Gestionnaire de données
 - ▶ Lecture, manipulation (select, delete, update, join...), stockage
- ▶ Algèbre linéaire
 - ▶ Opérations classiques sur les vecteurs, tableaux, matrices
- ▶ Statistiques et analyses de données
 - ▶ Grand nombre de méthodes statistiques d'analyse de données (des plus anciennes aux plus récentes)
- ▶ Moteur de sorties graphiques
 - ▶ Sortie écran ou fichiers
- ▶ Systèmes de modules
 - ▶ Alimenté par la communauté
- ▶ Interface facile avec C/C++, fortran, Java, Python...

29

Comment R travaille ?

- ▶ R est un langage interprété (non compilé)
- ▶ Les variables, les données, les fonctions, les résultats, etc. sont stockés dans la mémoire vive de l'ordinateur sous forme d'objets qui ont chacun un nom
- ▶ L'utilisateur va agir sur ces objets via des opérateurs (arithmétiques, logiques, de comparaison...) et des fonctions
- ▶ Une fonction peut être modélisée de la façon suivante :



30

Installation du logiciel en version desktop

- ▶ Rendez-vous sur le site du CRAN : <http://cran.r-project.org>
- ▶ Téléchargez la version de R adaptée à votre système d'exploitation
- ▶ Dans le cas de Windows, lancez le fichier R-3.x.x-win.exe
- ▶ Suivez les instructions

31

Installation de Rstudio en version desktop

- ▶ Rendez-vous sur le site de rstudio :
<https://www.rstudio.com/products/rstudio/download/>
- ▶ Téléchargez la dernière version
- ▶ Suivez les instructions

32

Paramétrages de l'environnement de travail

- ▶ Par défaut, R lit et écrit dans le répertoire de travail
- ▶ `getwd` permet de connaître le répertoire de travail
`getwd()`
- ▶ Pour changer le répertoire de travail :
 - ▶ Utiliser le menu : Fichier → Changer le répertoire courant
 - ▶ Utiliser la fonction `setwd()`
- ▶ En écrivant le chemin du répertoire de travail
`setwd("C:/travail")`
- ▶ En utilisant `dirname` et `file.choose`
`setwd(dirname(file.choose()))`

33

Sauvegarder son environnement

- ▶ Les données créées en mémoire peuvent être sauvegardées. Les fichiers de données R portent l'extension `.RData`
 - ▶ Fichier → sauver l'environnement de travail : copie toutes les données en mémoire dans un fichier `.Rdata`
 - ▶ En ligne de commande : `save.image(file = "lenomdufichier.RData")`
- ▶ A la fermeture de R, le logiciel vous propose de sauvegarder l'environnement de travail. Si vous répondez "oui", il le ré-ouvrira lors de la session suivante.
- ▶ Le logiciel vous propose également de sauvegarder l'historique des commandes. Ces fichiers portent l'extension `.Rhistory`

34

RStudio

- ▶ Un IDE évolué adapté à R
- ▶ Quelques raccourcis :
 - ▶ Alt + flèche : déplacer la ligne courante
 - ▶ Ctrl + Shift + C : commenter ou décommenter en bloc
 - ▶ Ctrl + D : supprime la ligne courante
 - ▶ Ctrl + I : indente un bloc de code
 - ▶ Tab : fournit l'auto-complétion (y.c. pour les chemins)

35

Créer des objets dans R

- ▶ On peut taper une expression sans assigner sa valeur à l'objet
`(10+2)*5`
- ▶ Un objet peut être créé avec l'opérateur "assigner" `->` ou `<-`
`a <- 15`
`5 -> b` # à éviter
- ▶ Si l'objet existe déjà, sa valeur peut être écrasée
`n <- 10+2`
`n <- 3+rnorm(1)`

36

Lister des objets en mémoire

- ▶ La fonction `ls` permet d'afficher la liste des objets en mémoire

```
names <- "Carmen"; n1 <- 10 ; n2 <- 100 ; m <- 0.5
ls()
```
- ▶ Si l'on veut lister uniquement les objets qui contiennent un caractère donné dans leur nom, on utilisera l'option `pattern`

```
ls(pattern = "m")
```
- ▶ Pour restreindre la liste aux objets dont le nom commence par le caractère en question

```
ls(pattern = "^m")
```
- ▶ La fonction `ls.str` fournit des détails sur les objets en mémoire

```
ls.str()
```

37

Supprimer des objets de l'environnement de travail

- ▶ On utilise la fonction `rm` pour effacer les objets
- ▶ Pour effacer un objet

```
rm(n1)
```
- ▶ Pour effacer tous les objets présents en mémoire

```
rm(list = ls())
```
- ▶ On peut sélectionner certains objets en se servant des options vues pour `ls`

```
rm(list = ls(pattern = "^m"))
```

38

Les différents types d'objets disponibles dans R

Objet	Modes	Plusieurs modes possibles dans le même objet?
Vecteur	Numérique, caractère, complexe ou logique	Non
Facteur	Numérique ou caractère	Non
Tableau	Numérique, caractère, complexe ou logique	Non
Matrice	Numérique, caractère, complexe ou logique	Non
Liste	Numérique, caractère, complexe , logique, fonction, expression...	Oui
Tableau de données	Numérique, caractère, complexe ou logique	Oui
Ts	Numérique, caractère, complexe ou logique	Non

39

Les opérateurs

Opérateurs...					
... arithmétiques		... de comparaison		... logiques	
+	addition	<	inférieur à	!x	NON logique
-	soustraction	>	supérieur à	x & y	ET logique
*	multiplication	<=	inférieur ou égal à	x && y	Idem
/	division	>=	supérieur ou égal à	x y	OU logique
^	puissance	==	égal à	x y	idem
%/%	division entière	!=	différent	xor(x,y)	OU exclusif

40

MANIPULATION DES OBJETS

► **Objet vector :**

- `x[n]` nième élément
- `x[-n]` tous sauf le nième élément
- `x[1:n]` les n premiers éléments
- `x[-(1:n)]` tout sauf les n premiers éléments
- `x[c(1,4,2)]` des éléments spécifiques
- `x["nom"]` l'élément nommé "nom"
- `x[x > 3]` tous les éléments plus grands que 3
- `x[x > 3 & x < 5]` tous les éléments plus grands que 3 et plus petits que 5

41

MANIPULATION DES OBJETS

► **Objet list :**

- `x[i]` le ou les éléments i de la liste (renvoyé(s) sous forme de liste, même principe que pour les vecteurs)
- `x[[i]]` ième élément de la liste
- `x[["nom"]]` l'élément nommé "nom"
- `x$nom` l'élément nommé "nom"

42

MANIPULATION DES OBJETS

- ▶ **Objet matrix :**
 - ▶ `x[i,j]` l'élément de la ligne i, colonne j
 - ▶ `x[i,]` toute la ligne i
 - ▶ `x[,j]` toute la colonne j
 - ▶ `x[,c(1,3)]` les colonnes 1 et 3
 - ▶ `x["nom",]` la ligne nommée "nom"
- ▶ **Objet data.frame :**
 - ▶ **Idem que pour les Matrix + -**
 - ▶ `x[["nom"]]` la colonne nommée "nom"
 - ▶ `x$nom` la colonne nommée "nom"

43

Qu'est-ce qu'un package ?

- ▶ Un package est une librairie additionnelle aux fonctionnalités de base
 - ▶ *FactoMineR est une librairie destinée à réaliser des analyses factorielles (ACP, ACM, AFC...)*
- ▶ Les packages sont créés et maintenus par la communauté des utilisateurs de R
 - ▶ *FactoMineR est un package maintenu par François Husson au LMA² d'Agrocampus Ovest*
- ▶ C'est une famille de fonction
 - ▶ *le package FactoMineR contient 72 fonctions*
- ▶ La fonction `install.packages("nomdupackage")` permet d'installer le package sur l'ordinateur depuis le dépôt sécurisé de packages (CRAN)
 - ▶ `install.packages("FactoMineR")`
- ▶ La fonction `library(nomdupackage)` rend disponibles les fonctions du package dans l'environnement de travail de R
 - ▶ `library(FactoMineR)`

44

45

46

Les ressources sur le web

- ▶ La page web du CRAN (Comprehensive R Archive Network) : <https://cran.r-project.org/>
- ▶ Le R journal propose des articles sur : <http://www.journal.r-project.org/>
- ▶ Sur twitter: Le hashtag #rstats
- ▶ La mailing list officielle: <https://stat.ethz.ch/mailman/listinfo/r-help>
- ▶ Le forum du CIRAD <http://forums.cirad.fr/logiciel-R/>
- ▶ <http://stackoverflow.com/questions/tagged/r>

47

Premier exemple d'utilisation de R

- ▶ Une ACP avec FactoMineR
- ▶ Les étapes :
 - ▶ Lancez Rstudio
 - ▶ Charger les données (données auto)
 - ▶ Charger FactoMineR
 - ▶ Lancer une ACP avec FactoMineR

48

Une alternative : PYTHON

- ▶ Python est un langage de programmation de haut niveau interprété

49

Comment installer Python ?

- ▶ Python est généralement installé en natif sur la plupart des machines
- ▶ On peut installer Python directement depuis le site de Python :
 - ▶ <https://www.python.org/downloads/>



- ▶ On préférera des versions packagées plus spécifiques à la data science : Anaconda

50

Python 2 ou Python 3 ?

- ▶ Python 3 a été introduit en 2008
- ▶ Python 2 est encore la version la plus utilisée
- ▶ Pour python.org :
 - ▶ *Python 2.x is legacy, Python 3.x is the present and future of the language* (mais ça fait déjà 7 ans...)
- ▶ Problème de rétrocompatibilité de Python 3
- ▶ Nous utilisons dans cette formation python 3 car il s'agit de la version la plus récente. Il est simple de travailler en python 2 lorsqu'on maîtrise python 3

51

Les changements

- ▶ Ce qui a changé avec Python 3 :
 - ▶ `print` est maintenant une fonction
 - ▶ Certaines fonctions natives ne renvoient plus des listes (`dict`, `map()`, `filter()`, `range()`, `zip()`)
 - ▶ Quelques changements sur l'ordre des comparaisons
 - ▶ Les entiers sont maintenant divisibles, `1/2` est maintenant un `float`
 - ▶ Des changements importants sur les données textuelles : tout ce qui est du texte est encodé Unicode
 - ▶ Et d'autres petits changements

52

Comment installer Python pour l'analyse de données ?

- ▶ Plusieurs méthodes existent pour installer Python
- ▶ Nous utiliserons une méthode simple : **Anaconda**
- ▶ <https://www.anaconda.com/download/>
- ▶ Que trouve-t-on dans Anaconda :
 - ▶ Plus de 380 bibliothèques déjà installées
 - ▶ Spécialement adaptée à la data science
- ▶ Les principales bibliothèques disponibles dans Anaconda :
 - ▶ lpython
 - ▶ matplotlib
 - ▶ Jupyter notebook
 - ▶ numPy
 - ▶ pandas
 - ▶ scikit-learn
 - ▶ seaborn
 - ▶ ...



**Anaconda est surtout dévolu à Python
mais on peut aussi utiliser R avec
Anaconda**

53

Anaconda

Anaconda 5.0.0 For Windows Installer

Python 3.6 version *

↓ Download

[64-Bit Graphical Installer \(535 MB\)](#) ⓘ
[32-Bit Graphical Installer \(436 MB\)](#)

Python 2.7 version *

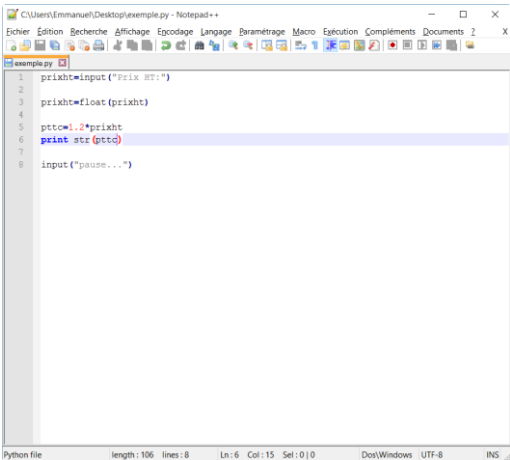
↓ Download

[64-Bit Graphical Installer \(522 MB\)](#) ⓘ
[32-Bit Graphical Installer \(421 MB\)](#)

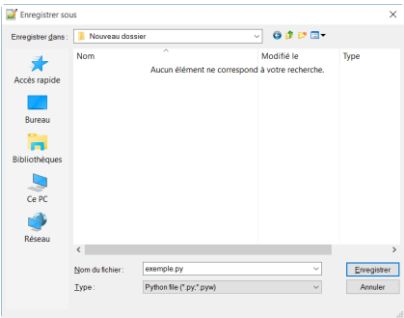
54

Méthode classique pour programmer en Python

- On utilise un éditeur de texte classique (type NotePad++) et on sauvegarde le programme en .py



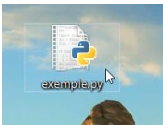
```
1 prixht=input("Prix HT:")
2
3 prixht=float(prixht)
4
5 ptto=1.2*prixht
6 print str(ptto)
7
8 input("pause...")
```



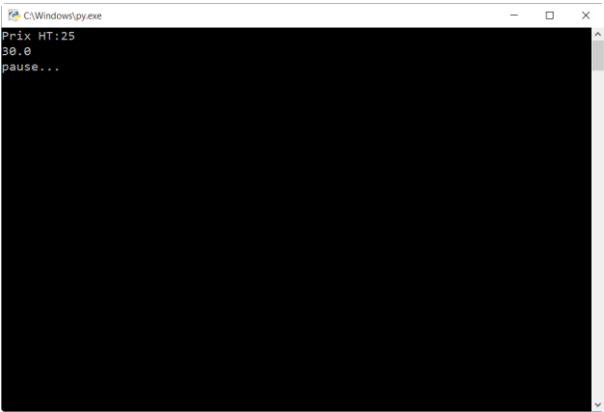
55

Méthode classique pour programmer en Python

- On lance le programme en cliquant dessus



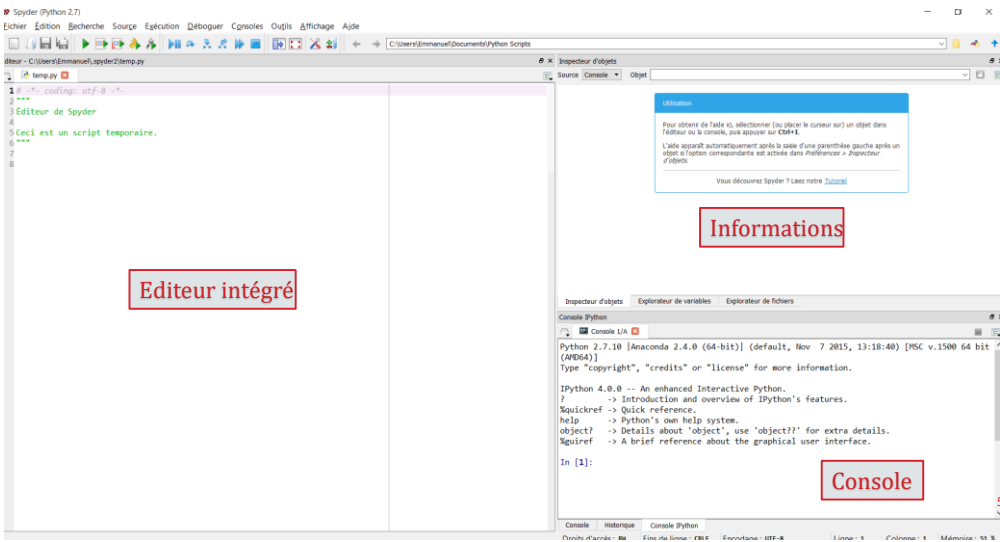
- Le programme s'exécute automatiquement



→ Débogage très difficile,
peu d'interactivité

56

Une approche alternative – utiliser SPYDER



Très proche de RStudio en terme d'ergonomie

Informations

Spyder est basé sur la distribution Anaconda et sur la console améliorée IPython

Console

Une interface pour programmer en Python : iPython

- ▶ L'une des forces de Python est son interpréteur qui permet une réelle interactivité lorsqu'on code en Python (à la différence d'autres langages compilés)
- ▶ L'interpréteur standard de Python est assez pauvre, c'est que ipython trouve sa force
- ▶ Il permet de créer un environnement de développement simple et unifié avec trois axes principaux :
 - ▶ Une ligne de commande vous simplifiant la vie
 - ▶ La possibilité d'avoir plusieurs clients sur le même noyau de travail avec les notebook
 - ▶ Une architecture pour mieux gérer le calcul parallélisé

JUPYTER NOTEBOOK

- ▶ Il s'agit d'une évolution de IPython qui vous permet de coder directement dans un navigateur web avec affichage des graphiques et des sorties sur des pages web
- ▶ Les notebooks travaillent directement sur le kernel ouvert (vous aurez généralement un programme qui tourne en plus)
- ▶ Depuis peu, les notebooks s'appellent Jupyter Notebook afin de se séparer de l'aspect Python sachant qu'on peut lancer des notebook avec du R ou d'autres langages

59

Lancer une session iPython notebook (Jupyter notebook)

```
Invite de commandes - ipython notebook
Microsoft Windows [version 10.0.10240]
(c) 2015 Microsoft Corporation. Tous droits réservés.

C:\Users\Emmanuel>ipython notebook
[I 14:57:17.334 NotebookApp] Serving notebooks from local directory: C:\Users\Emmanuel
[I 14:57:17.335 NotebookApp] 0 active kernels
[I 14:57:17.335 NotebookApp] The IPython Notebook is running at: http://localhost:8888/
[I 14:57:17.335 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[I 14:58:32.020 NotebookApp] Creating new notebook in
[I 14:58:32.707 NotebookApp] Kernel started: a0ee46ef-80a9-4e66-acff-68be13050476
[I 15:00:32.904 NotebookApp] Saving file at /Untitled1.ipynb
```

60

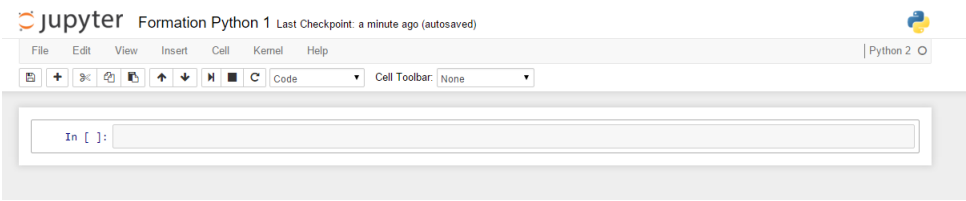
Les notebooks

- La session s'ouvre dans le navigateur de votre machine



61

Jupyter notebook

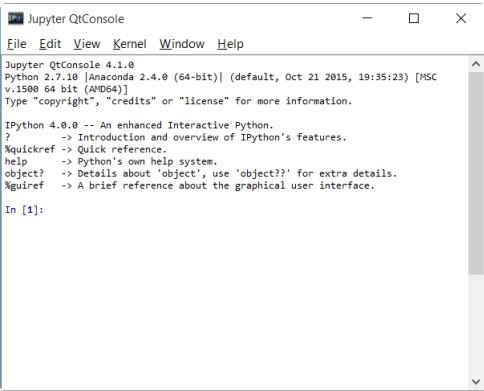


- Une invite de commande Python apparaît et permet de commencer à coder en Python
- On peut renommer ce fichier et ainsi travailler sur un fichier qui se sauvegarde automatiquement
- Pour soumettre une ligne de code, il suffit d'utiliser la combinaison (Alt+Entrée)
- Pour obtenir de l'aide sur une fonction, il suffit d'entre le nom de la fonction suivi de ?
- Pour obtenir les arguments de la fonction on utilise Shift+Tab

62

Les principes de base de Python

- ▶ Pas de compilation (langage interprété)
- ▶ L'indentation est centrale
- ▶ Langage orienté objet, des objets spécifiques sont créés (similaire à R)



```
Jupyter QtConsole
File Edit View Kernel Window Help

Jupyter QtConsole 4.1.0
Python 2.7.10 [Anaconda 2.4.0 (64-bit)] (default, Oct 21 2015, 19:35:23) [MSC
v.1500 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 4.0.0 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%gui? -> A brief reference about the graphical user interface.

In [1]:
```

63

Les bibliothèques Python pour la data science – présentation

64

NumPy

- ▶ NumPy, raccourci pour Numerical Python, est la bibliothèque de référence pour le calcul scientifique
- ▶ La plupart des bibliothèques sont construites en s'appuyant sur NumPy
- ▶ Il est surtout utile pour ses arrays et les possibilités de calculs sur les arrays
- ▶ Il a des outils pour importer et exporter des données dans des arrays
- ▶ Il possède des fonctions d'algèbre linéaire et de génération de nombres aléatoires
- ▶ Il possède des outils pour intégrer du C, du C++, du Fortran... dans Python
- ▶ Les arrays de NumPy sont spécialement adaptés pour du traitement de données (plus que les structures habituelles de Python)

65

SciPy

- ▶ SciPy possède un panel de fonctions de calcul scientifique tels que :
 - ▶ `scipy.integrate` : intégration numérique et équations différentielles
 - ▶ `scipy.linalg` : algèbre linéaire et décomposition de matrices
 - ▶ `scipy.optimize` : algorithme d'optimisation et de recherche de racine
 - ▶ `scipy.signal` : outils de traitement du signal
 - ▶ `scipy.stats` : distributions de probabilités et tests statistiques

66

Pandas

- ▶ C'est la bibliothèque centrale de gestion des données
- ▶ Import / export de données
- ▶ Manipulations avancées

67

matplotlib

- ▶ `matplotlib` est la bibliothèque de référence pour la visualisation des données
- ▶ Elle est bien adaptée pour l'obtention de graphiques pour des publications
- ▶ Elle s'intègre parfaitement avec iPython notamment grâce aux graphiques interactifs

68

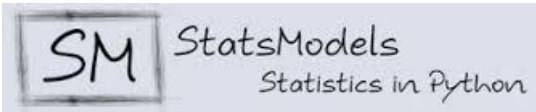
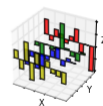
Les autres bibliothèques

► Nous utiliserons aussi :

- scikit-learn
- statsmodels
- Seaborn
- bokeh
- tensor-flow
- nltk
- ...

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



69

Traitement d'un exemple en python

- Faire une ACP avec python
- Charger les données
- Lancer le modèle
- ...

70