ANALYSE DE DONNÉES

Nassim Laga
nassimlaga@gmail.com
Janvier 2019

Plan du cours

- **Introduction au data mining (1h)**
- Apprentissage supervisé et non supervisé
- Focus sur l'apprentissage supervisé + mesures de performance et de validation (2h)
- Focus sur l'apprentissage non supervisé + mesures de performance et de validation (2h)
- Focus sur l'analyse de texte (2h)

INTRODUCTION AU DATA MINING

Nassim Laga
nassimlaga@gmail.com
Janvier 2019

Plan du jour

- Quelques définitions (datascience, apprentissage, machine learning, bigdata…etc.)…pour dire la même chose ?
- Quelques use case
- But du cours
- Le processus d'analyse de données
- L'équation d'un bon data scientist
- Visualisation des données
- Introduction à Python

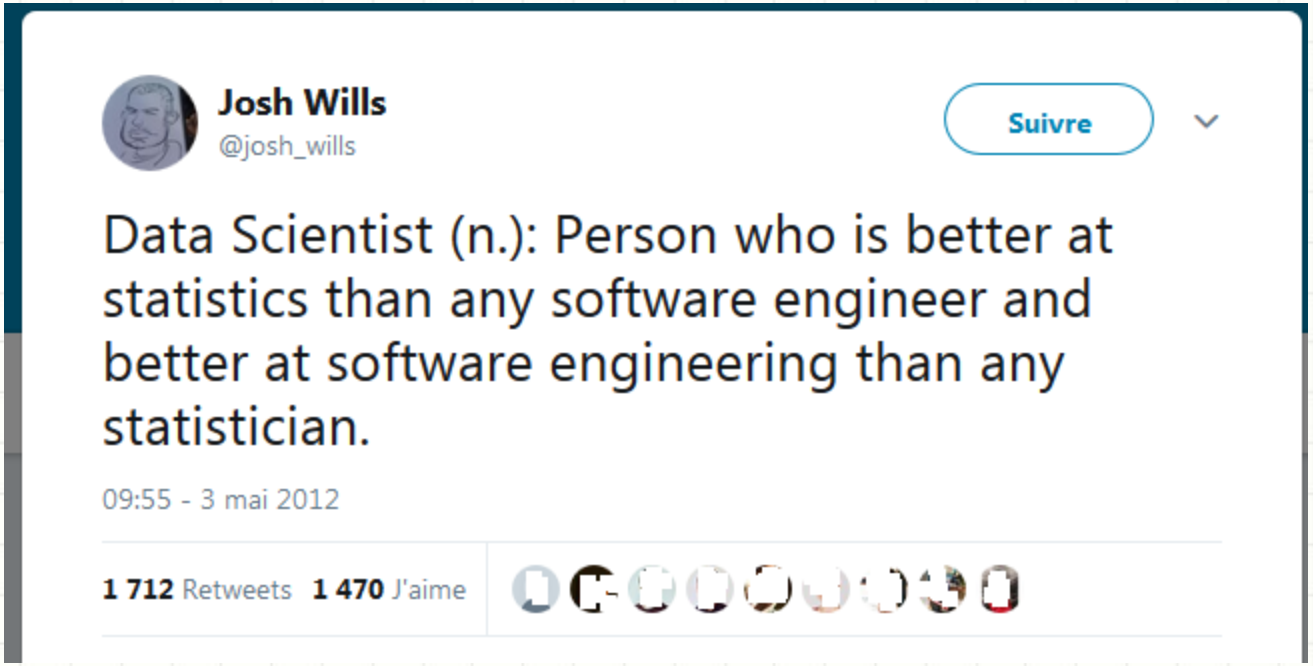
Quelques définitions

- La **statistique** est l'étude d'un phénomène par la collecte de données, leur analyse, leur traitement, l'interprétation des résultats et leur présentation afin de rendre les données compréhensibles par tous.
- Le **« Machine Learning »** (ou apprentissage artificiel, ou apprentissage statistiques) est un sous domaine de l'informatique qui donne au ordinateurs la possibilité d'apprendre sans être explicitement programmés.
- La **« Data science »** est un domaine interdisciplinaire qui utilise des méthodes scientifiques, des processus, des algorithmes et des systèmes pour extraire des connaissances à partir de données sous différentes formes, structurées ou non. … est un « concept pour unifier les statistiques, l'analyse des données, l'apprentissage automatique et leurs méthodes associées »
- Le **« Big data »** incluent généralement des ensembles de données dont la taille dépasse la capacité des outils logiciels couramment utilisés pour capturer, organiser, gérer et traiter les données dans un délai raisonnable.

Quelques définitions… mon point de vue

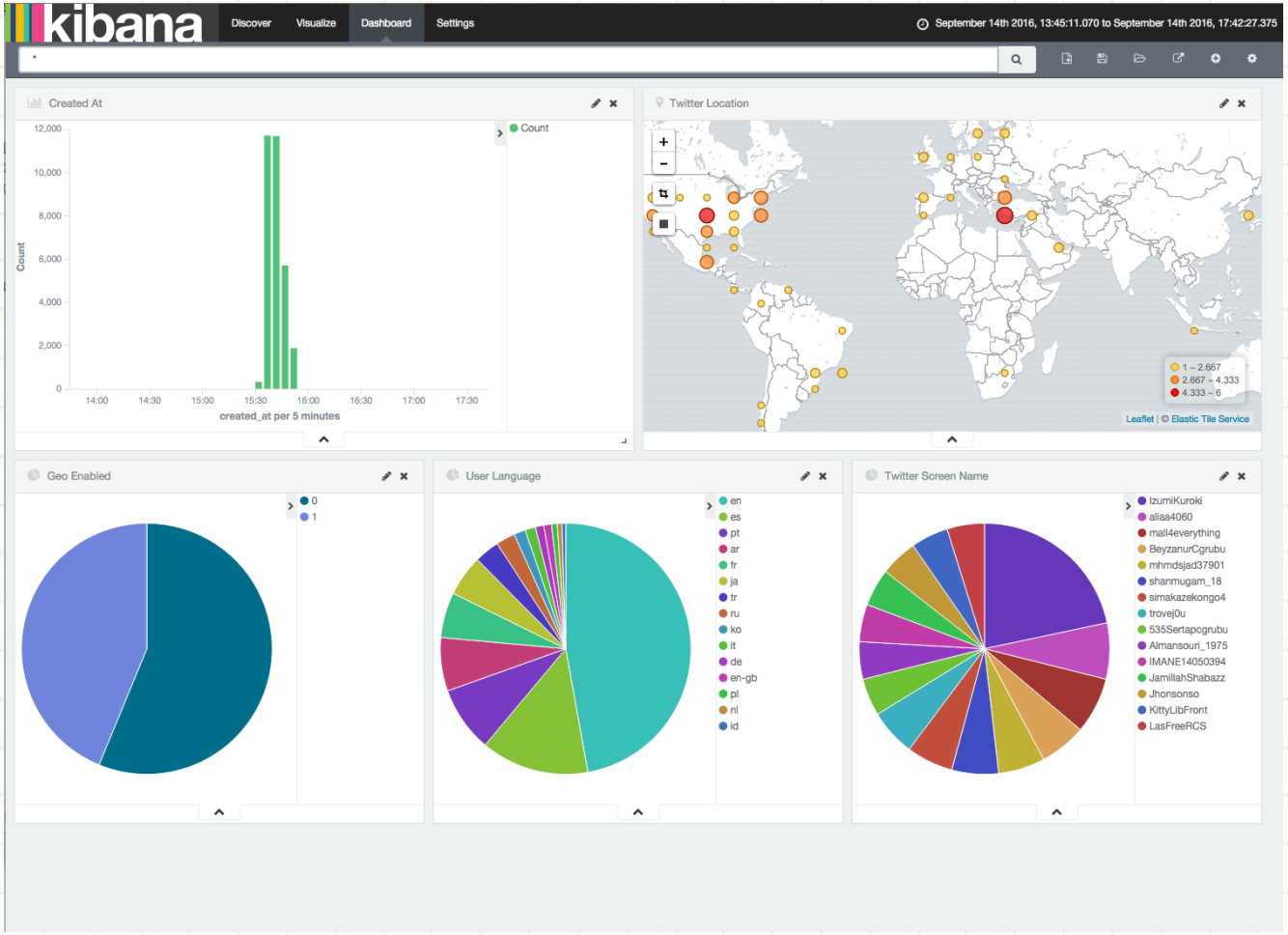
- Les termes sont tous connexes, c'est une évolution des statistiques qui est le terme le plus ancien.
  - 1er évolution vers le machine learning lorsque les informaticiens ont pris un peu pour eux le sujet. Le machine learning sont des statistiques de point de vue informatique ! … ou en d'autre termes, comment faire apprendre les ordinateurs sans qu'ils soient explicitement programmés, en utilisant des concepts statistiques.
  - 2ème « évolution » vers le bigdata, pour adresser l'explosion du volume de données à traiter
    - d'où les technologies de bigdata : hadoop, map/reduce, spark, NoSQL, etc.

Quelques définitions… et la data science alors ?



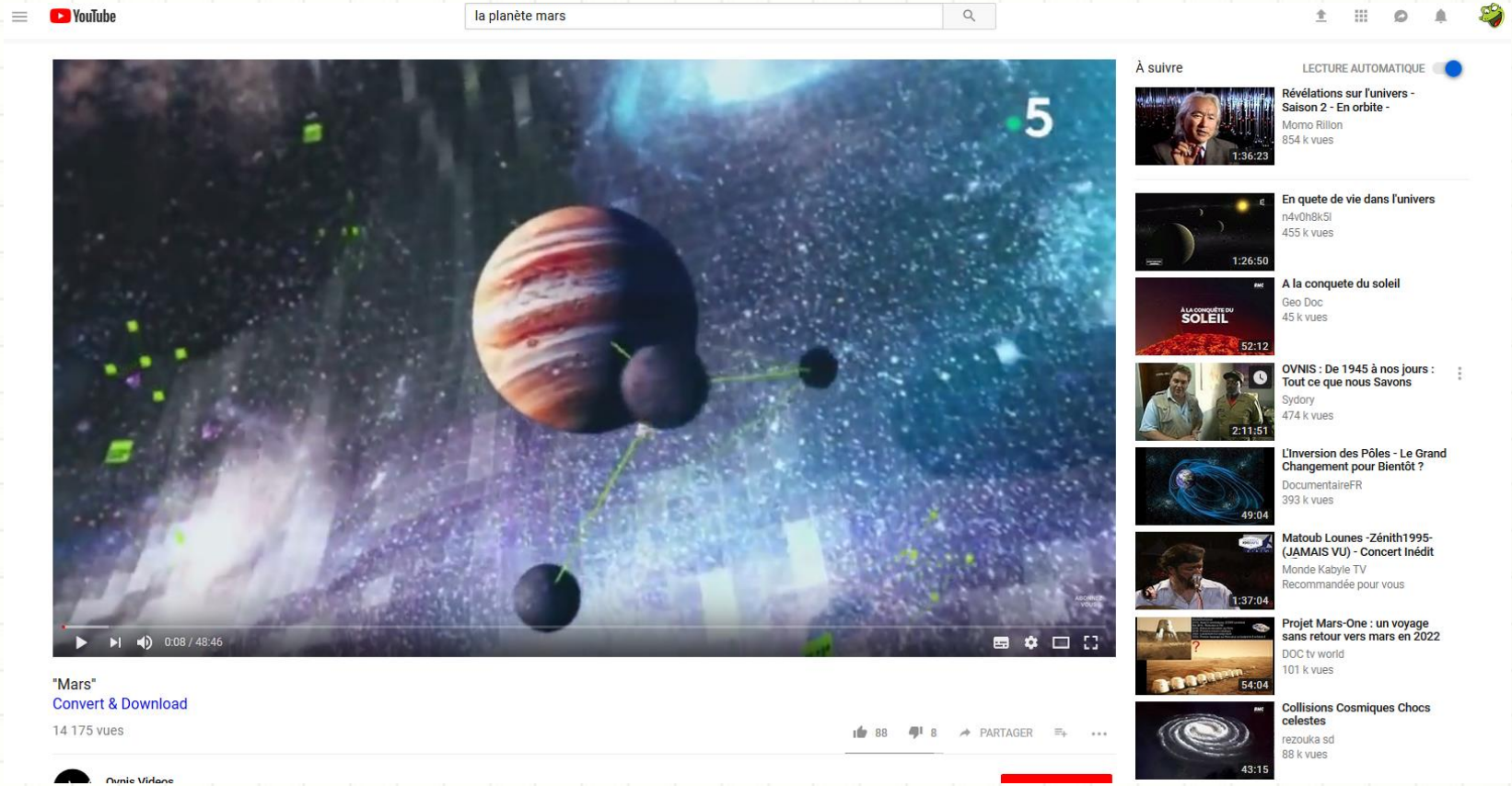
Quelques use case

- Dashboard (peut être personnalisé)



Sous forme d’un logiciel « intelligent »

- Moteur de recommandation



Sous forme d'un logiciel « intelligent »

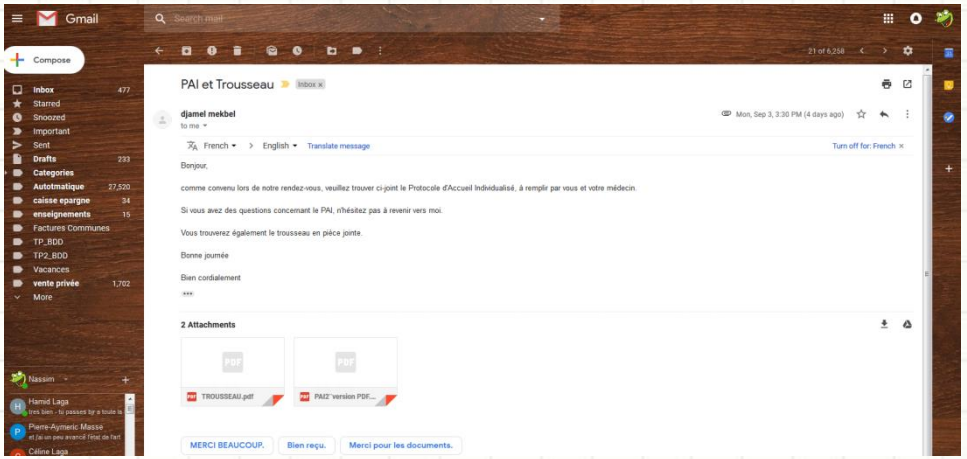
- Moteur de recommandation (publicité)

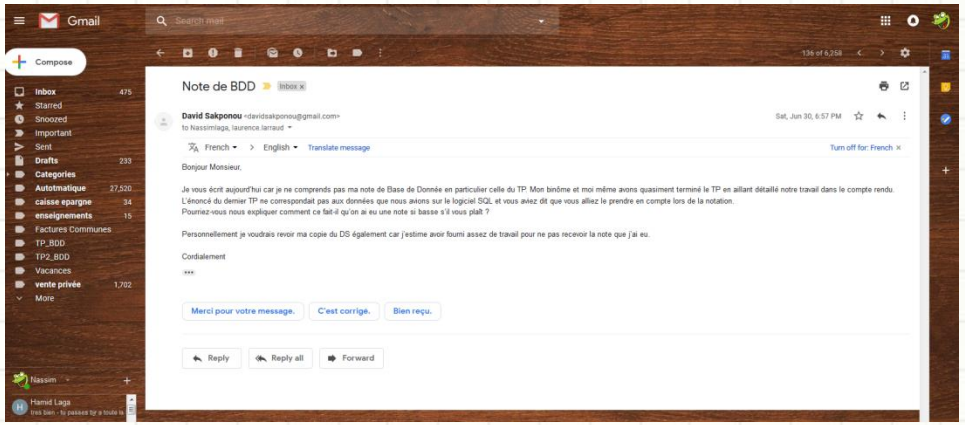
Sous forme d’un logiciel « intelligent »

- Moteur de recherche
  - Indexation d’un volume gigantesque de données: structurée, semi-structurées, et non-structurées

Sous forme d’un logiciel « intelligent »

- Fonction de prédiction intégrée dans un logiciel
  - détecteur de spams
  - classification automatique de documents
  - Détection d’intention dans un texte, et rédaction de réponse automatique

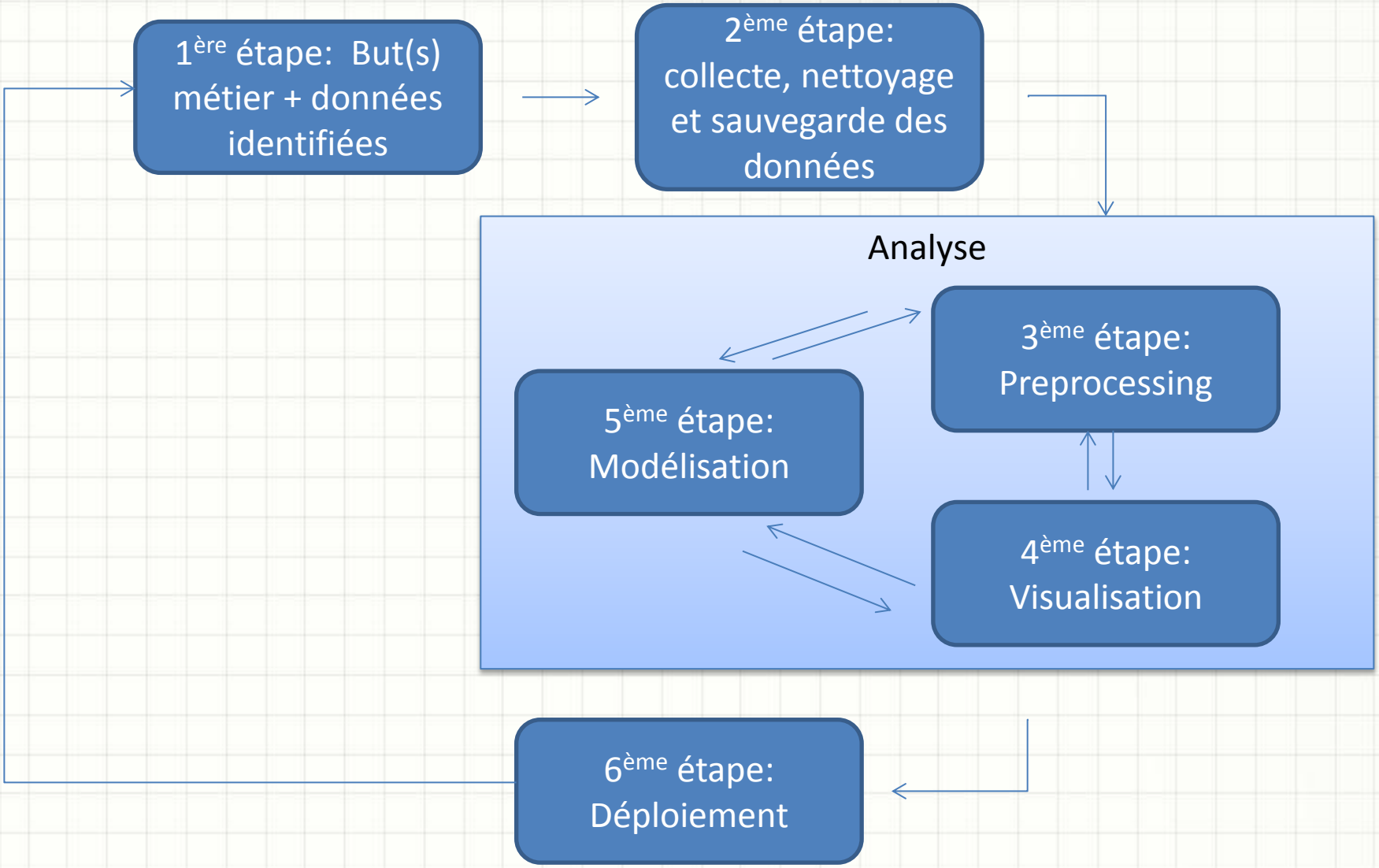




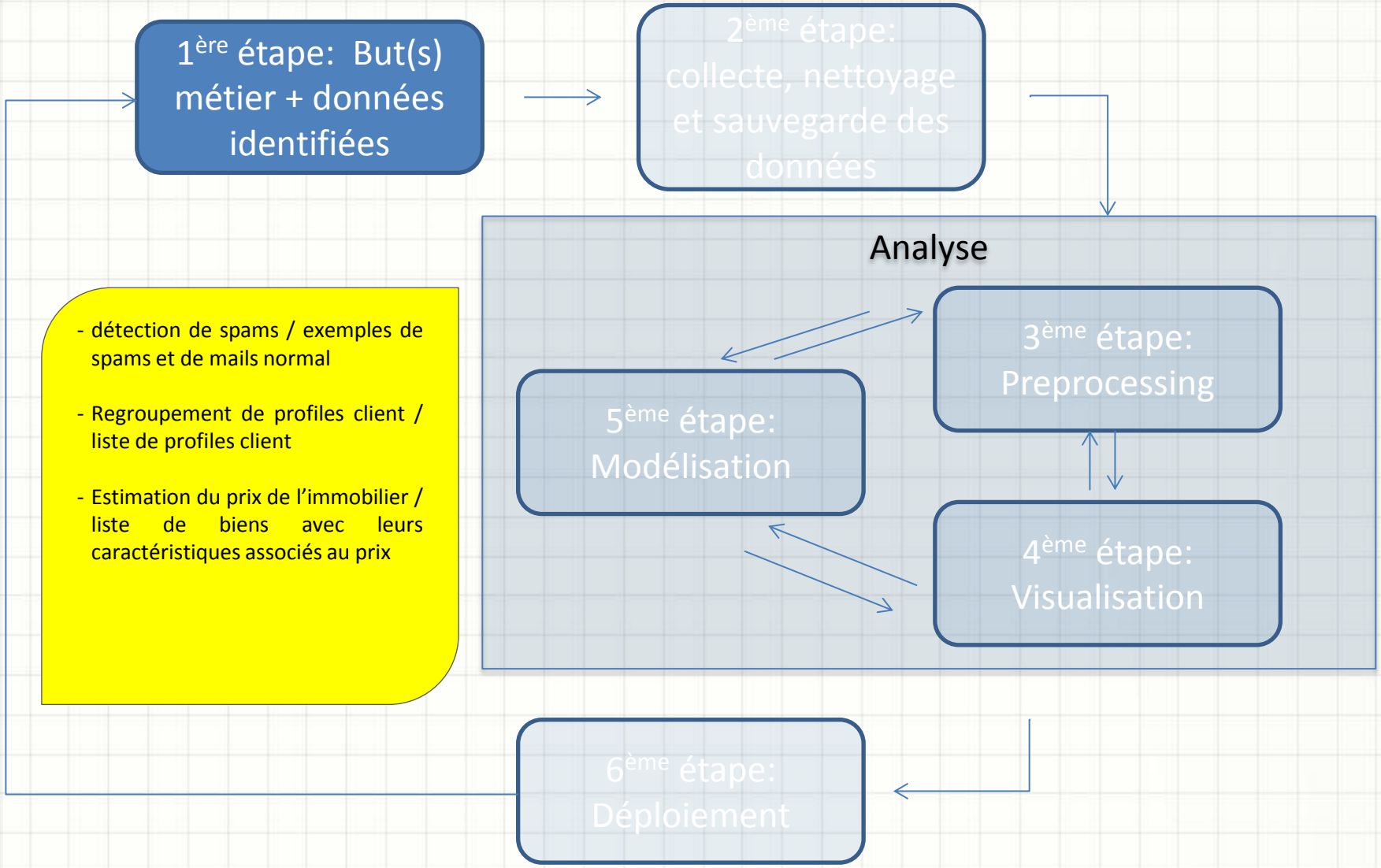
But du cours

- Nous n’allons pas faire du bigdata ☹
- Je suis plus ingénieur ☺ que statisticien ☹
- Par conséquent, nous allons faire du machine learning (ou apprentissage machine en français), en se focalisant plus sur la pratique que la théorie

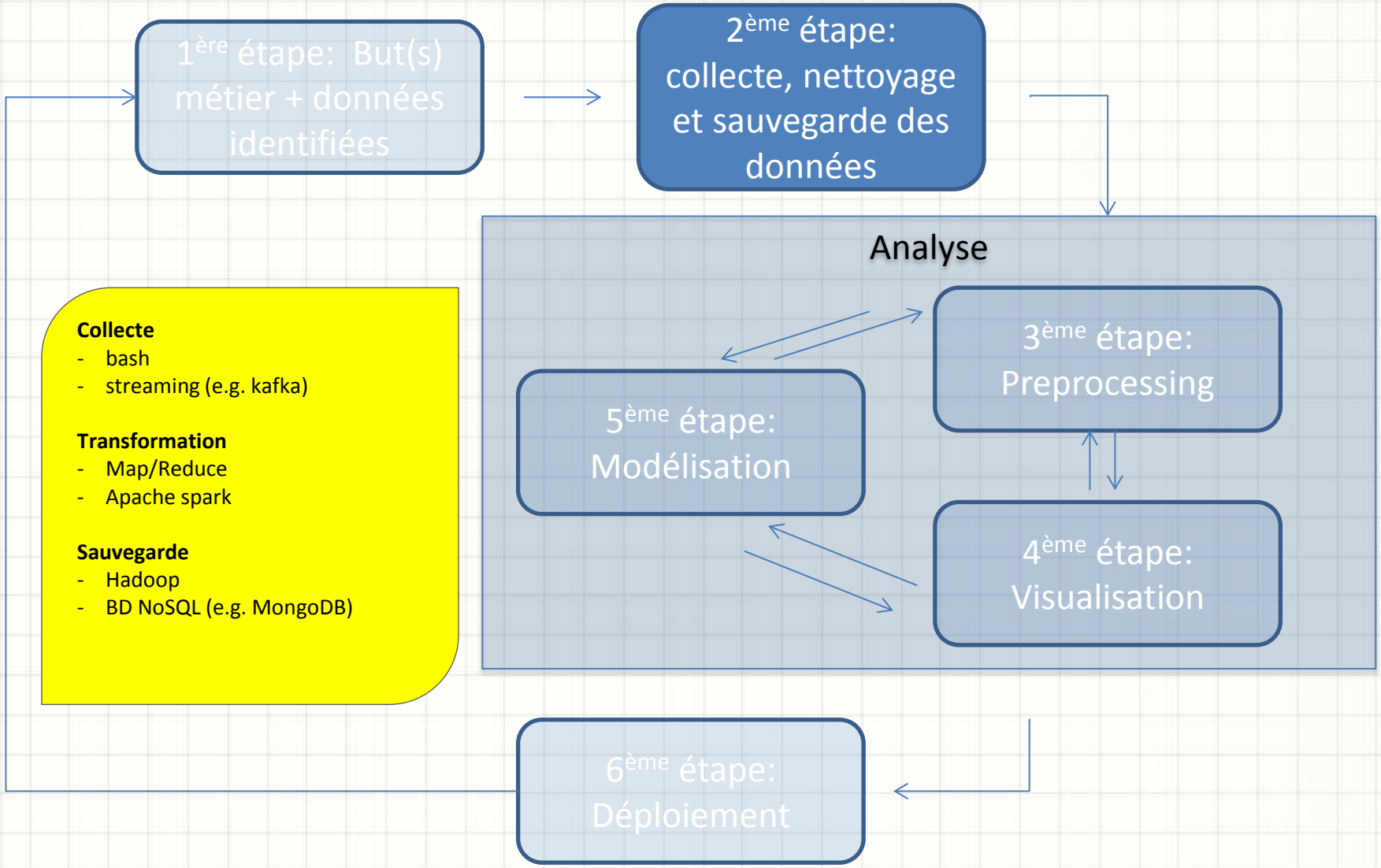
Le processus d'analyse de données



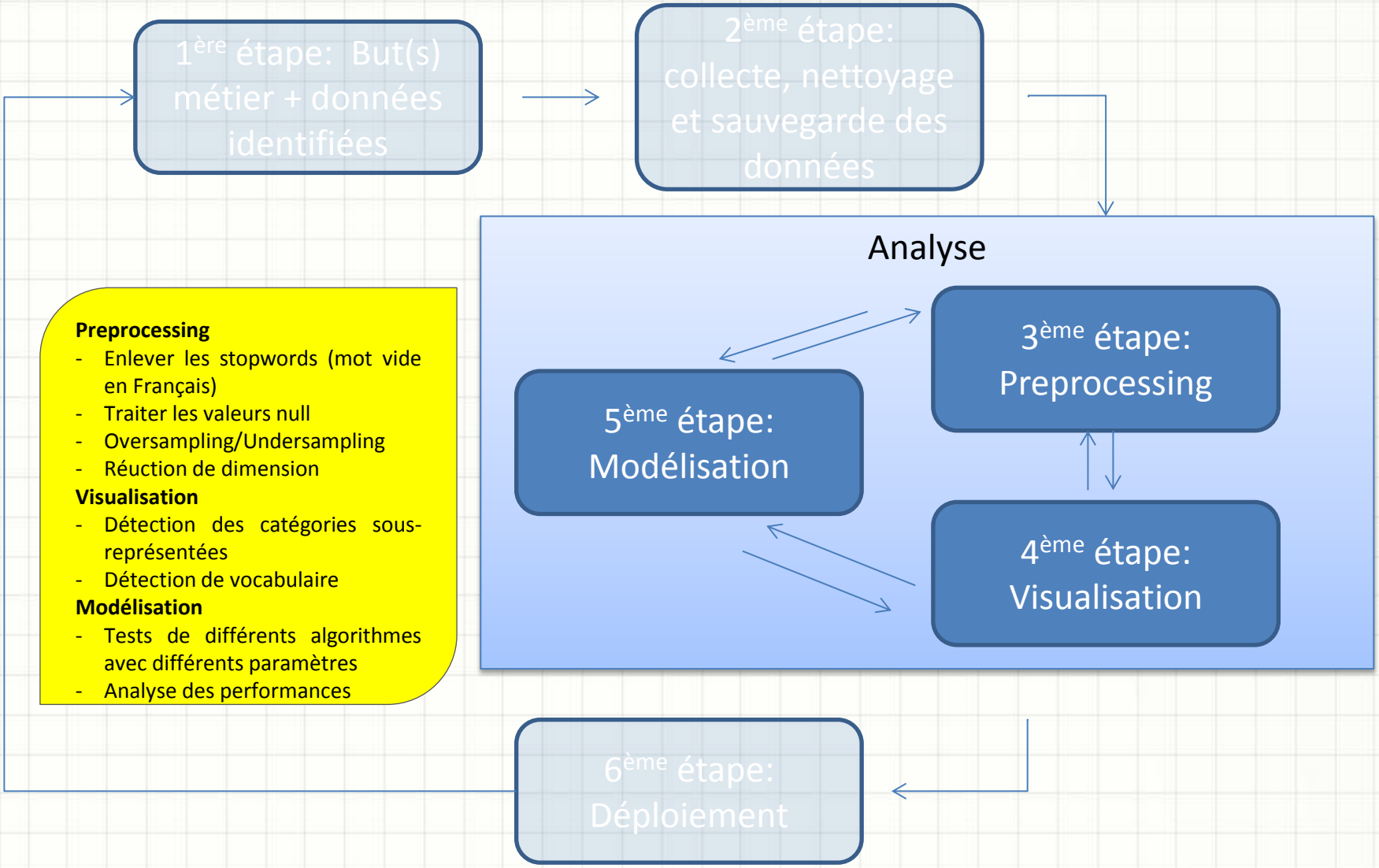
Le processus d'analyse de données



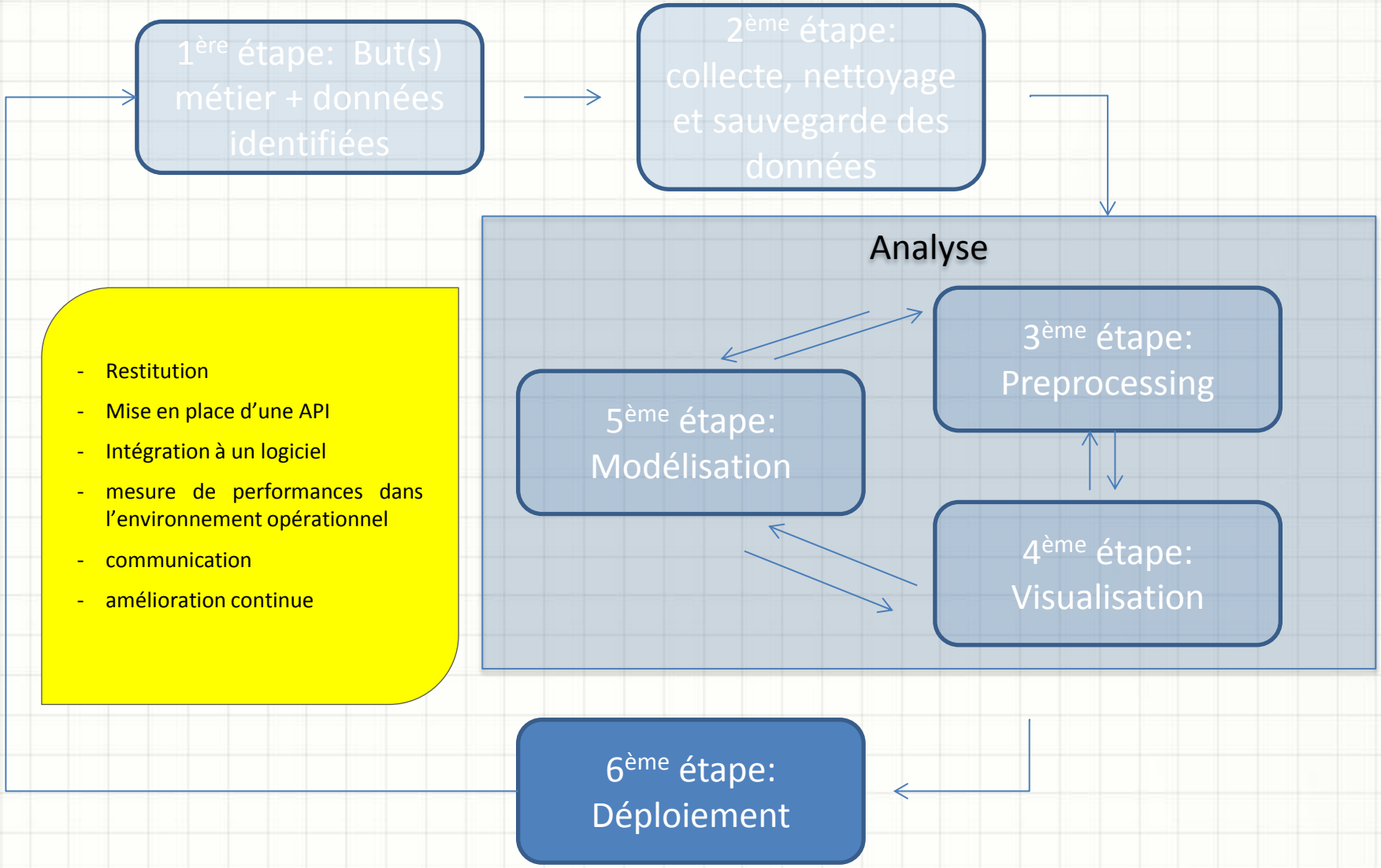
Le processus d'analyse de données



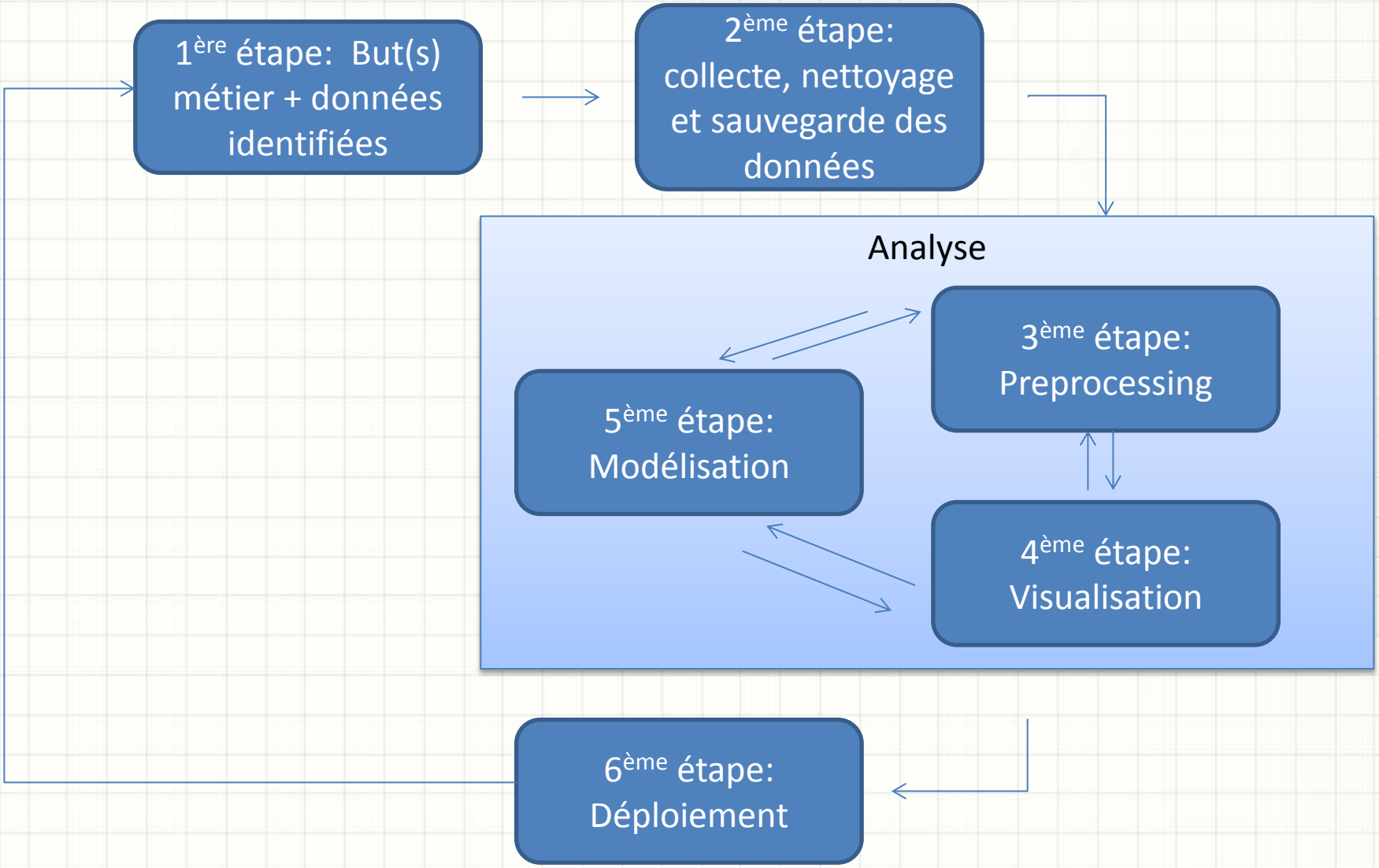
Le processus d'analyse de données



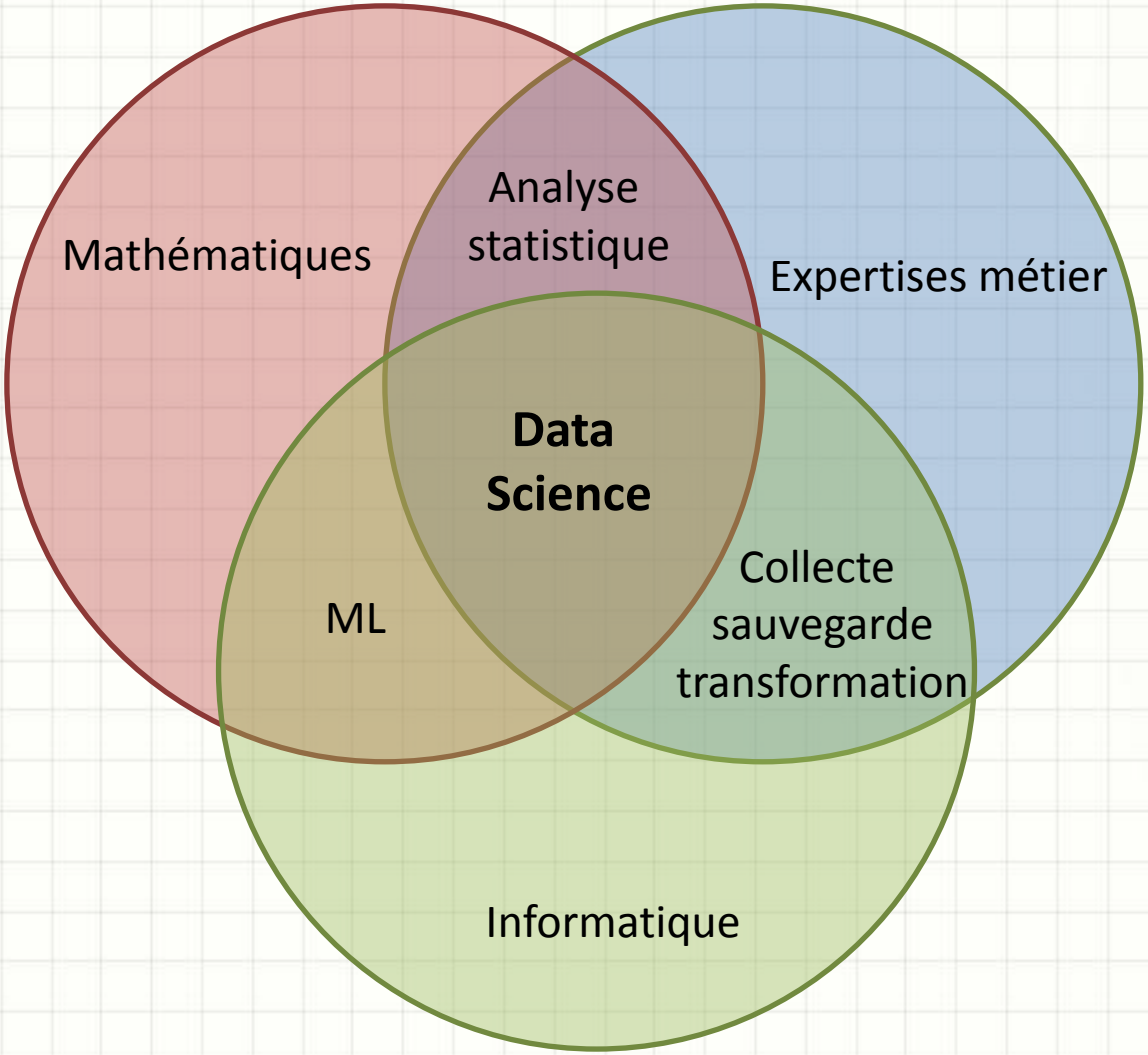
Le processus d'analyse de données



L'équation d'un bon data scientist



L’équation d’un bon data scientist



L’équation d’un bon data scientist

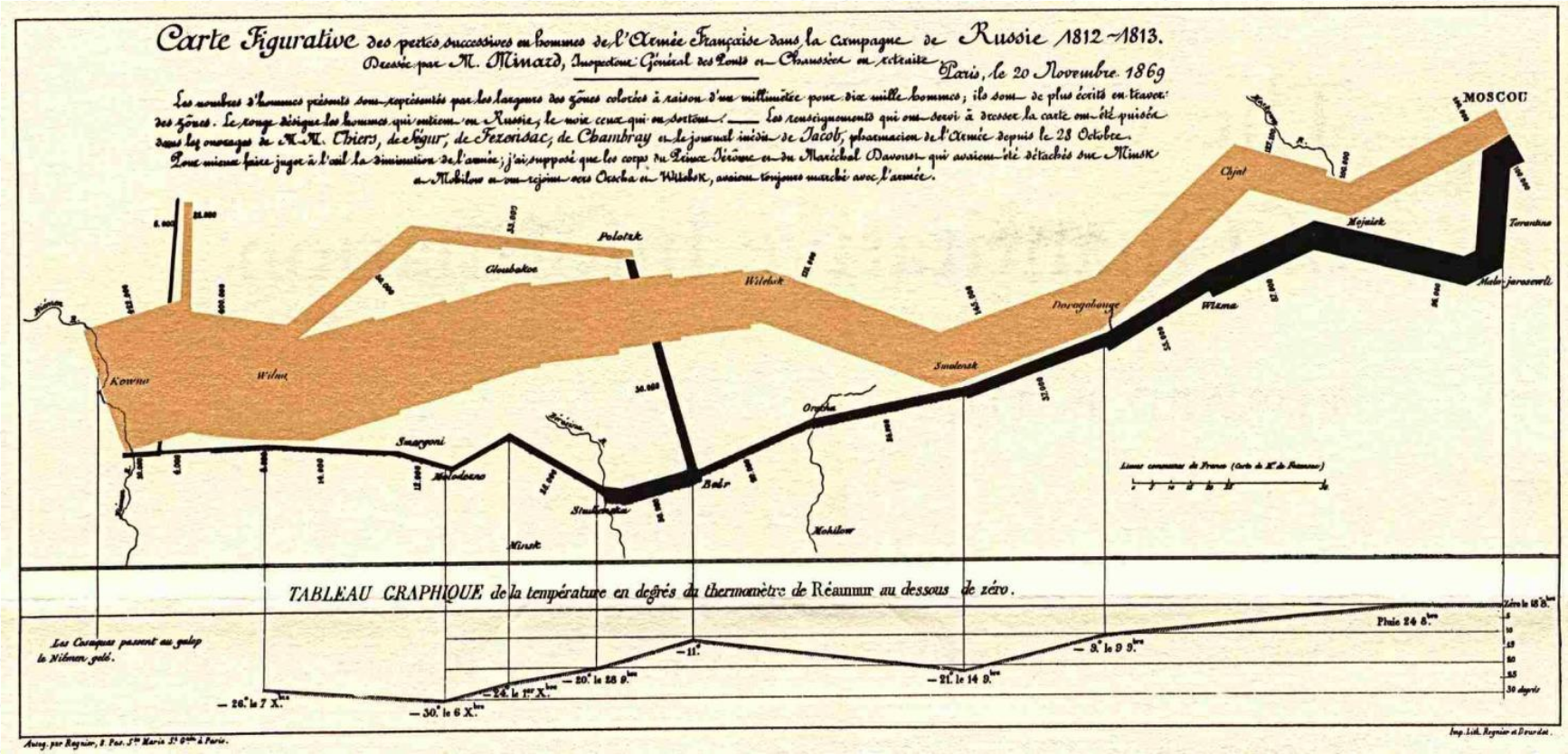
- Josh Wills : “Person who is better at statistics than any software engineer and better at software engineering than any statistician.”

Visualisation des données

- Utilisée à la fois pour
  - L'étape d'exploration de jeux de données et de prétraitement
  - L'étape de restitution des résultats

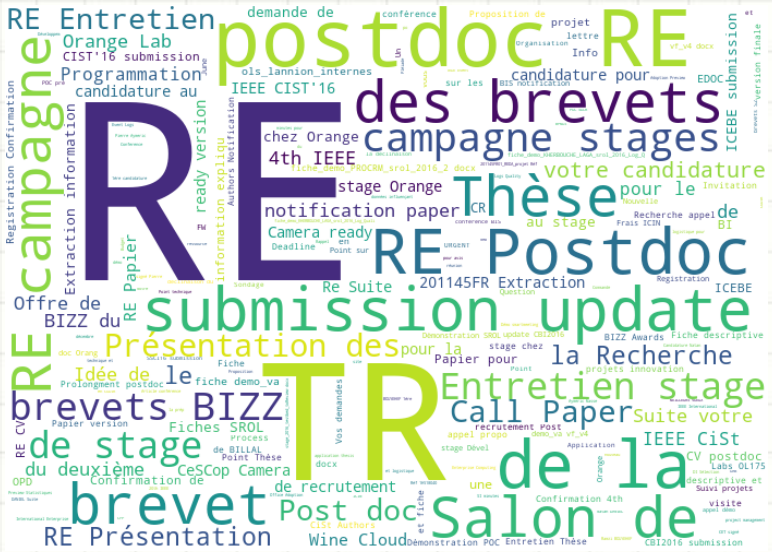
Visualisation des données

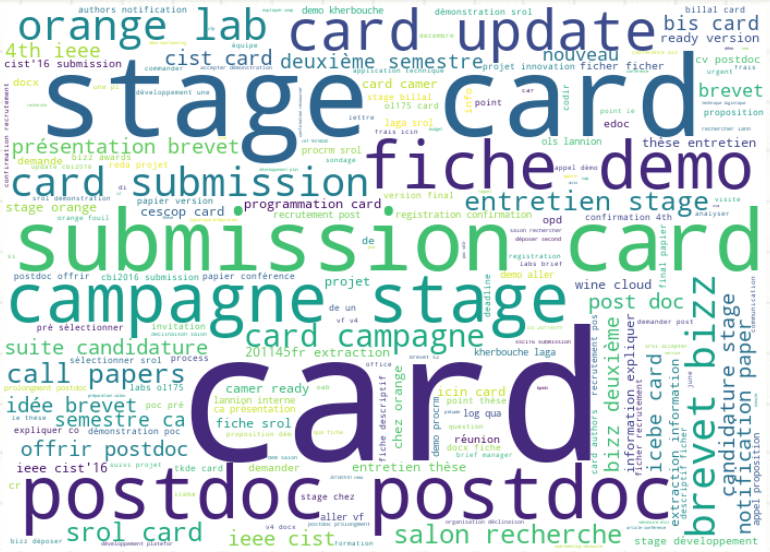
- Charles Joseph Minard, ingénieur civil français, 1969



Visualisation des données

- Prétraitement des données, exemples



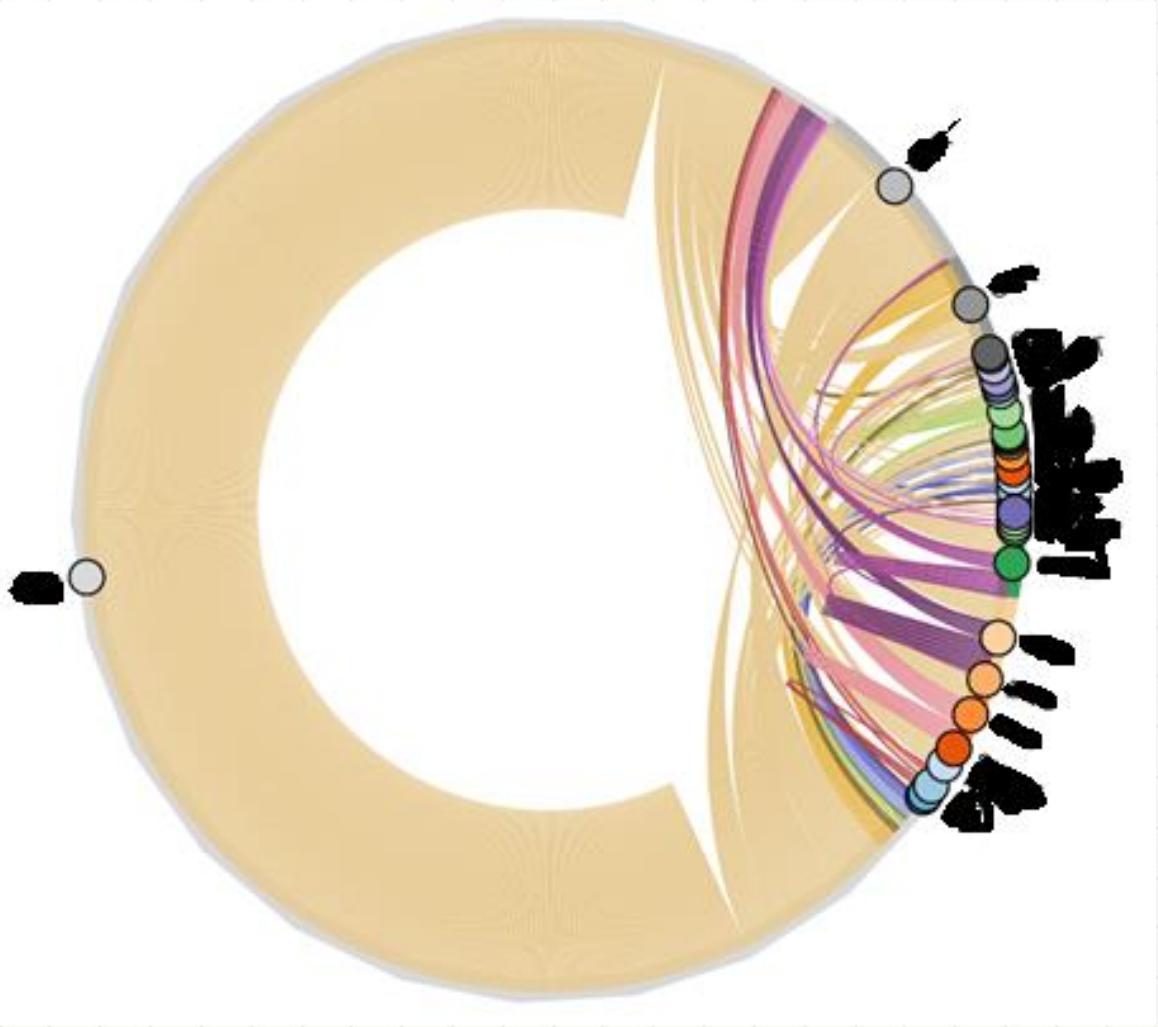


Visualisation des données

- Prétraitement des données, exemples

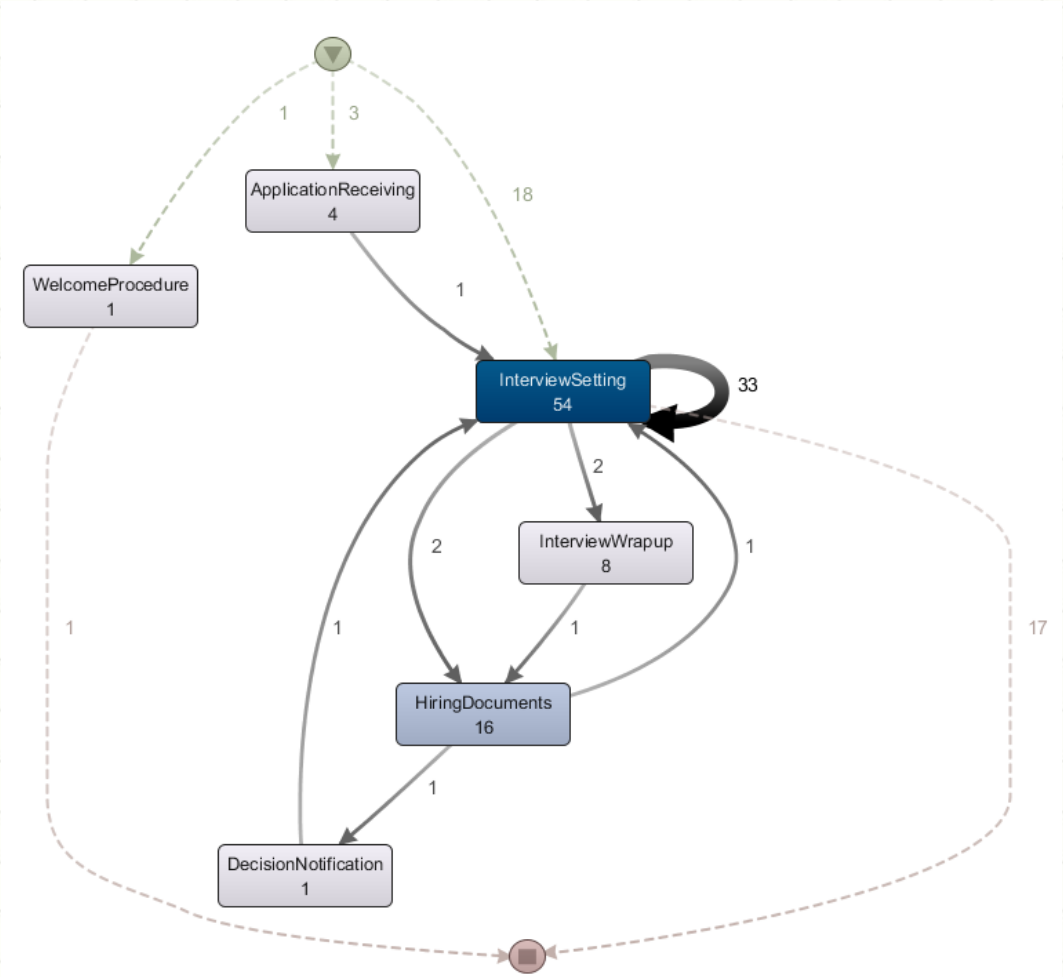
Visualisation des données

- Restitution, exemples



Visualisation des données

- Restitution, exemples



Les langages / outils de la datascience

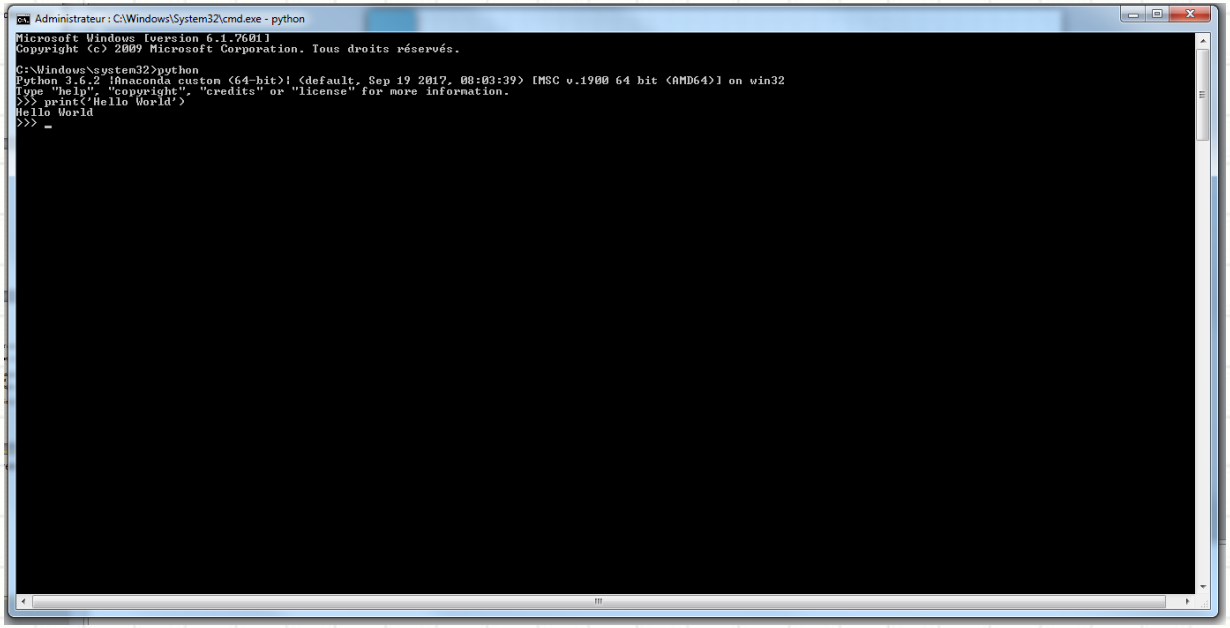
- R
- Python
- Julia
- …

Introduction à python

- Langage interprété
  - Plus lent, mais portable assez facilement d'une machine à une autre
  - Nécessite un interpréteur auquel on soumet un fichier .py
- Comme tout langage évolué il intègre :
  - des types de données simples et évolués, des boucles, des conditions, des procédures et des fonctions
  - la possibilité de programmer en Orienté Objet avec les notions de classes, d'instances, et d'héritage
- Une communauté très active, donnant lieu à de nombreuse librairies très intéressantes
  - Numpy, Scipy, scikit-learn, pandas, etc.

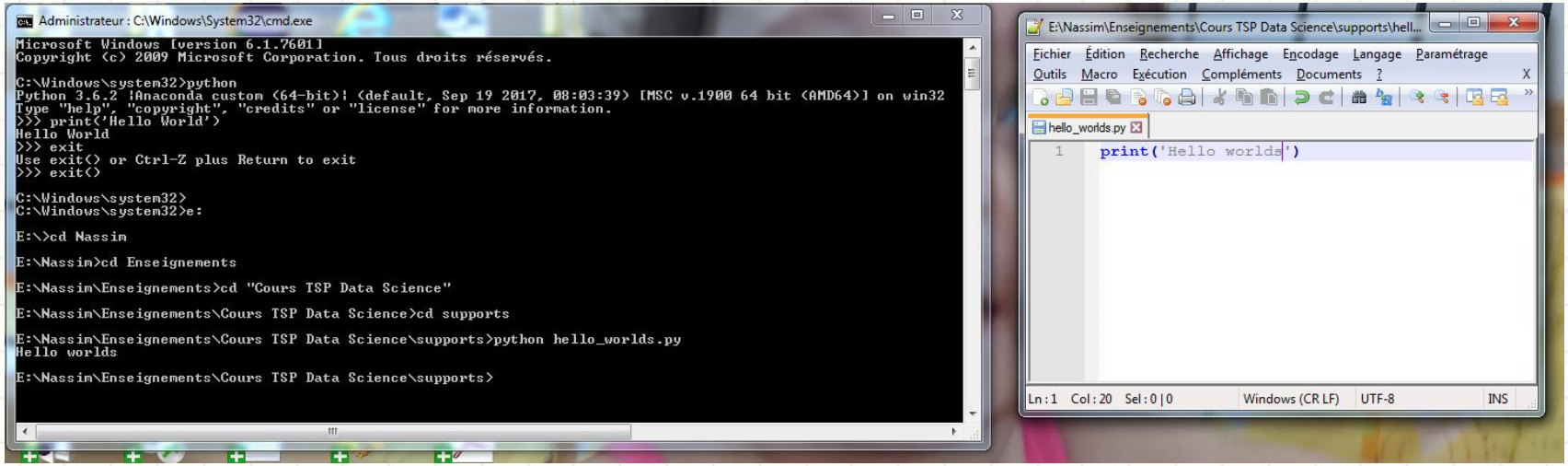
Introduction à python – premiers programmes

- Console interactive



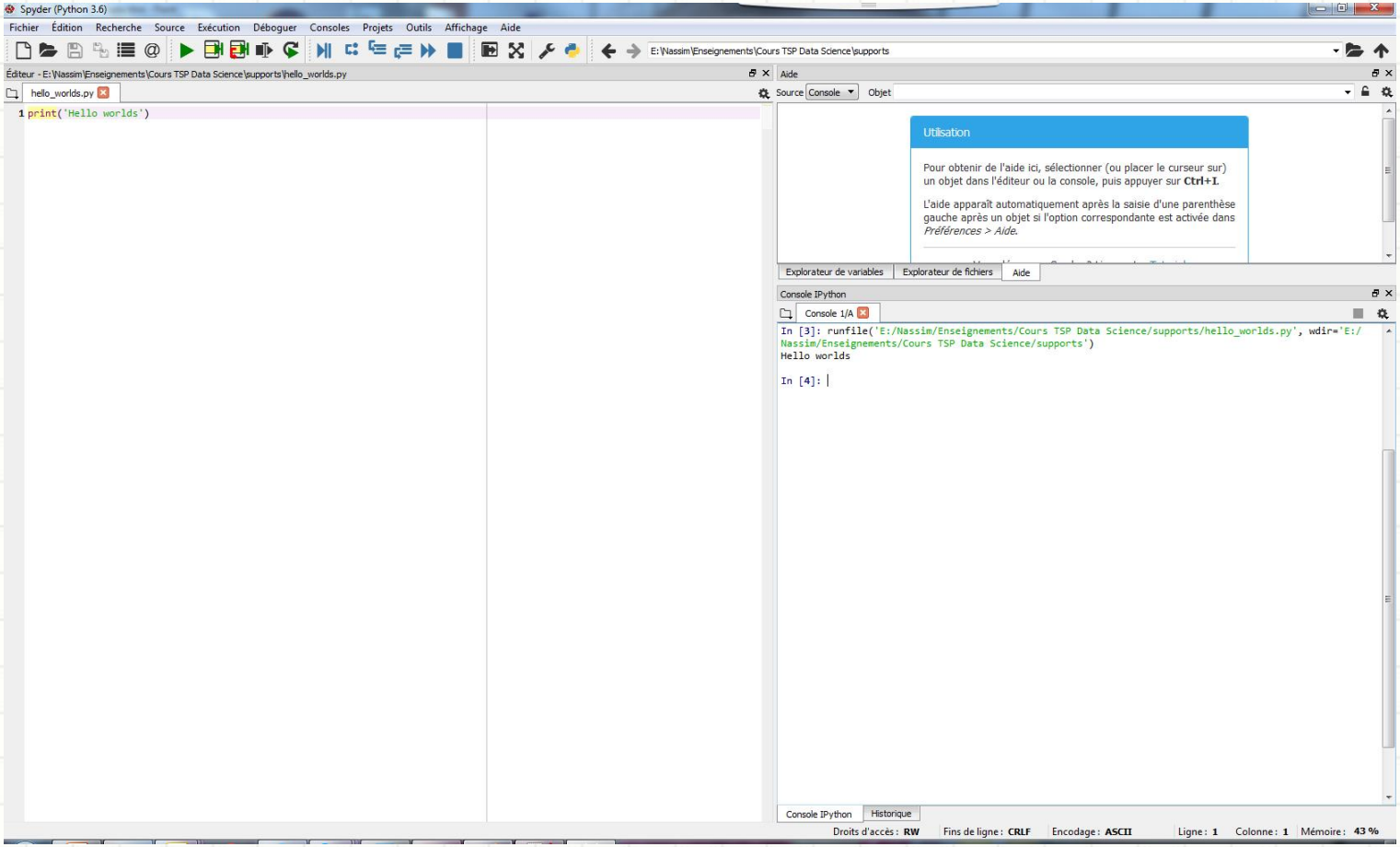
Introduction à python – premiers programmes

- Création d'un fichier .py et exécution à partir de la ligne de commande



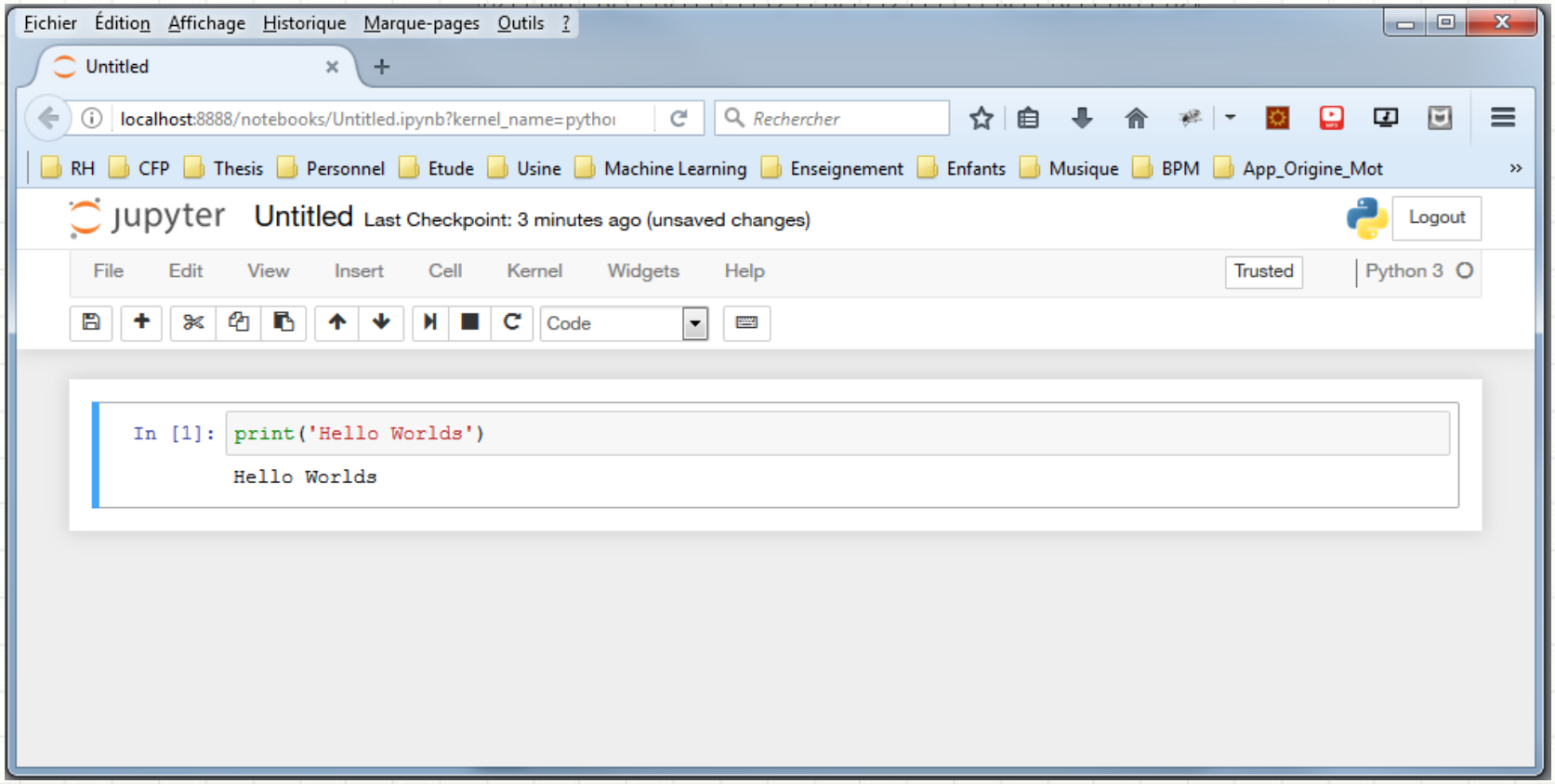
Introduction à python – premiers programmes

- spyder



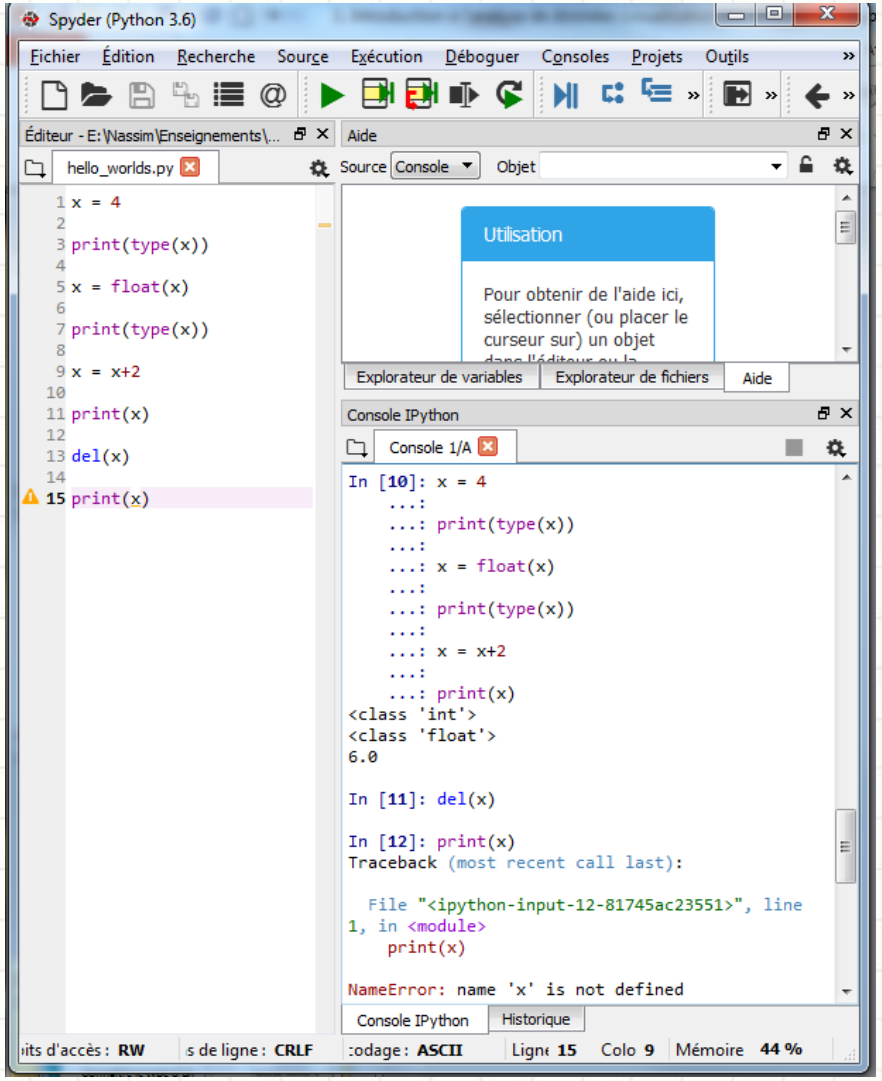
Introduction à python – premiers programmes

- Jupyter



Introduction à python – les opérations de base

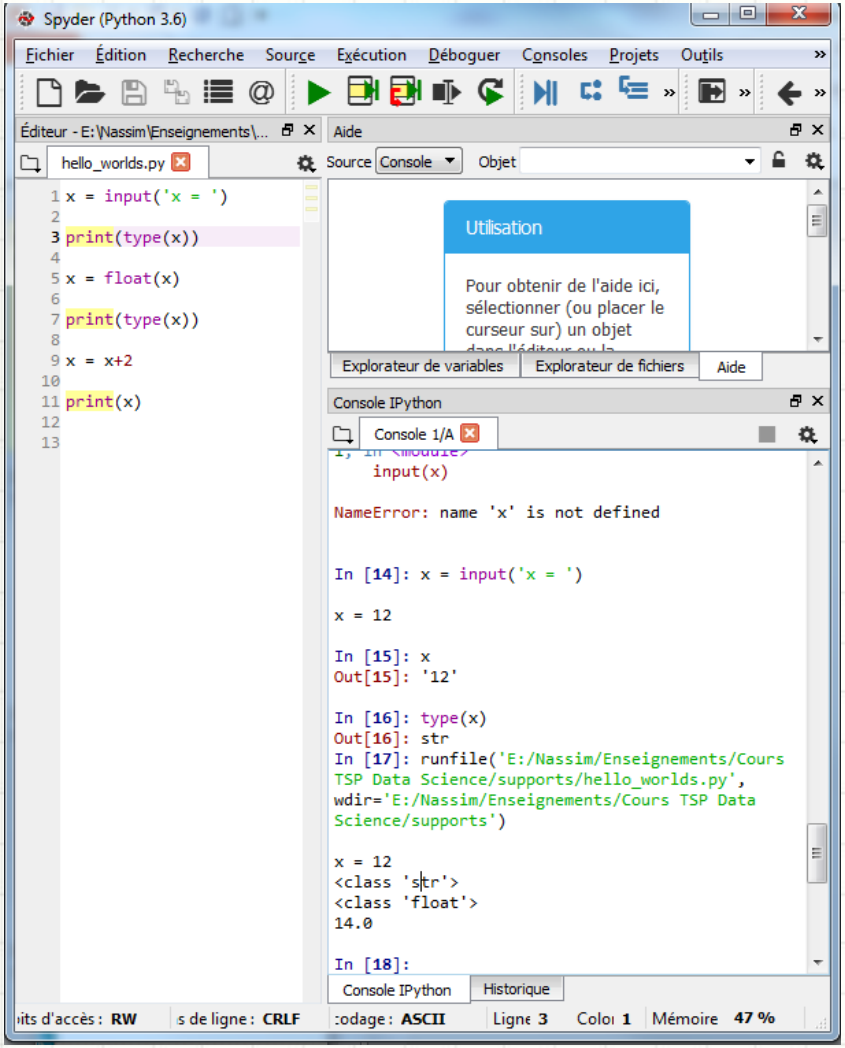
- Affectation
- Typage & Transtypage
- Connaître le type d'un objet/d'une variable
- Calcul
- Libération de la mémoire



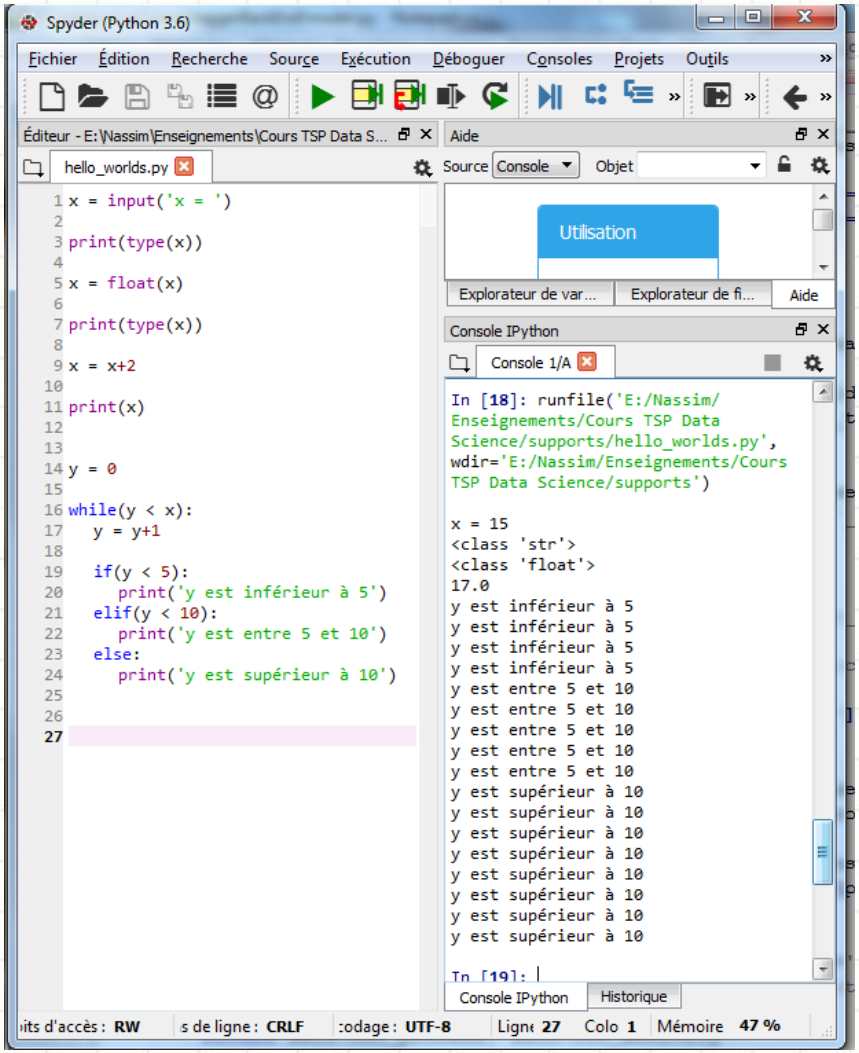
Introduction à python – les types

Types primitifs	Types complexes
int float (double) str Bool (True, False)	list tuple : N-uplet dict set: liste d'éléments uniques file None exception …

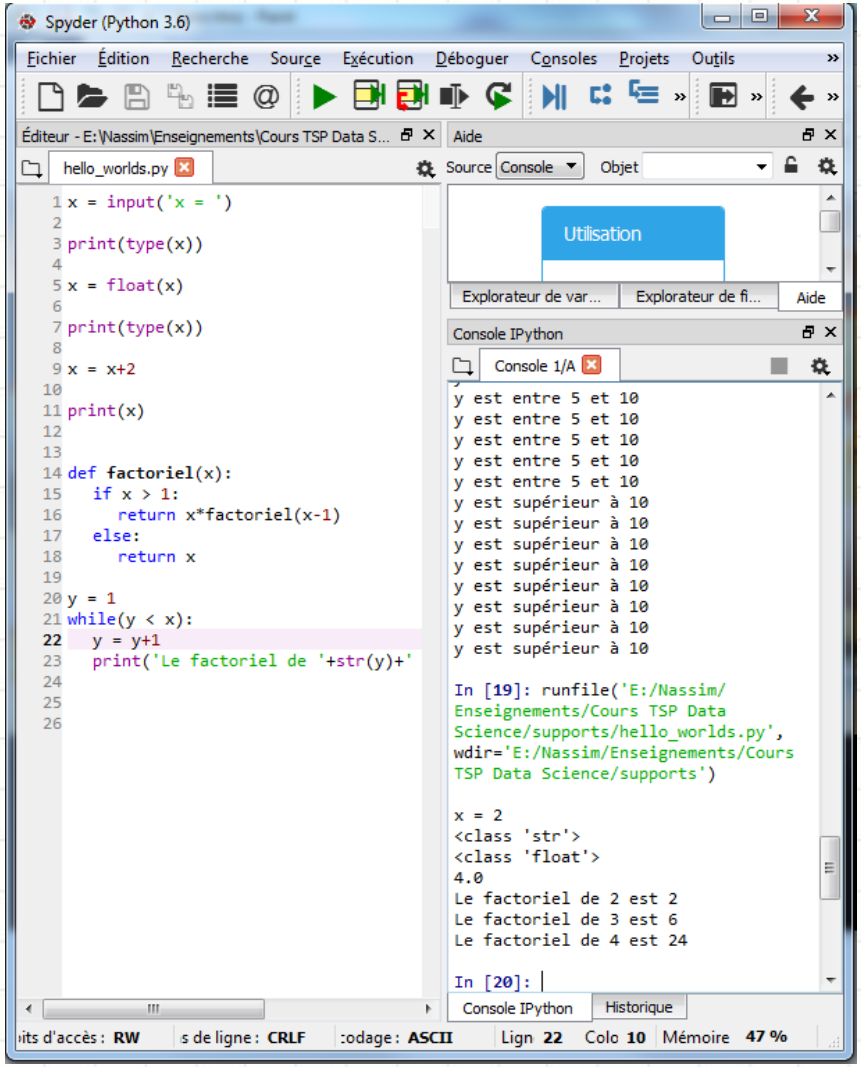
Introduction à python – les entrées/sorties



Introduction à python – structures algorithmiques



Introduction à python – les fonctions



Introduction à python – les listes

- C’est une collection d’éléments, potentiellement identiques, potentiellement de types différents
- Les éléments sont indexés numériquement x[0] récupère le premier élément de la liste
- On connait la taille de la liste grace à la fonction len(x)
- La fonction del(x[i]) supprime l’élément d’index i de la liste x
- x.append(element) ajoute l’élément en fin de liste

```
27 ys = [2, 3, 5, 10, 20, 20]
28 for y in ys:
29     print('Le factoriel de '+str(y)+' est '+str(factoriel(y)))
```

Python et l'analyse de données

- Librairies
  - Numpy
  - SciPy
  - Pandas
  - Scikit-learn
  - matplotlib (visualisation)
  - ggplot (visualisation)
  - Seaborn (visualisation)

Python et l'analyse de données - Pandas

- Une librairie permettant de gérer les données de 2 dimensions, à des fin d'analyse
- Concept issues des outils de statistiques telle que : R, matlab, et SAS
- Deux types de données clefs
  - Series
  - Dataframe

Python et l'analyse de données - Pandas

- Series
  - Structure de données d'une dimension, indexable
  - Plusieurs manière de créer une Series

```
In [18]: #Building a basic pandas Series
         import pandas
         s = pandas.Series(['John', 'Jack', 'Jack', 'Murielle'])
         s

Out[18]: 0        John
         1        Jack
         2        Jack
         3    Murielle
         dtype: object
```

Python et l'analyse de données - Pandas

- Series
  - Structure de données d'une dimension, indexable
  - Plusieurs manière de créer une Series

```
In [20]: #Building a basic pandas Series, specifying index
         import pandas
         s = pandas.Series(['John', 'Jack', 'Jack', 'Murielle'],
                           index = ['p1', 'p2', 'p3', 'p4'])
         s

Out[20]: p1         John
         p2         Jack
         p3         Jack
         p4     Murielle
         dtype: object
```

Python et l'analyse de données - Pandas

- Series
  - Sélection d'une valeur par index

```
In [21]: #Accès aux contenu
         s['p1']

Out[21]: 'John'
```

  - Sélection de plusieurs valeurs

```
In [24]: #Accès aux contenu
         s['p1', 'p3']

         --------------------
         TypeError
         C:\ProgramData\Anaco
           2482
```

X

```
In [23]: #Accès aux contenu
         s[['p1', 'p3']]

Out[23]: p1    John
         p3    Jack
         dtype: object
```

ok

Python et l'analyse de données - Pandas

- Series
  - Sélection par filtre

```
In [52]: #Sélection par filtre
         s = s[s != 'Jack']
         s

Out[52]: 0         John
         3     Murielle
         dtype: object
```

  - Opérations groupées, potentiellement conditionnées

```
In [54]: #Grouped Operations
         s + "blabla"

Out[54]: p1         Johnblabla
         p2         Jackblabla
         p3         Jackblabla
         p4     Murielleblabla
         dtype: object
```

```
In [2]: # grouped operation with conditions
        s[s != 'Jack'] + "blabla"

Out[2]: p1         Johnblabla
        p4     Murielleblabla
        dtype: object
```

Python et l'analyse de données - Pandas

- dataframe
  - Une structure de données à deux dimensions
  - Contient des lignes et des colonnes
  - indexée au niveau des lignes et des colonnes
  - peut être considérée comme un dictionnaire de Series, avec un index partagé

Python et l'analyse de données - Pandas

- Dataframe
  - Plusieurs manière de créer une dataframe
  - à partir d'un dictionnaire

In [5]:
```
#à partir d'un dictionnaire
dict_data = {
    'name':['John', 'Jack', 'Jack', 'Murielle'],
    'sexe':['M', 'M', 'M', 'F'],
    'age':[25, 50, 12, None]
}

df = pandas.DataFrame(dict_data)
df
```

Out[5]:

	age	name	sexe
0	25.0	John	M
1	50.0	Jack	M
2	12.0	Jack	M
3	NaN	Murielle	F

Python et l'analyse de données - Pandas

- Dataframe
  - Plusieurs manière de créer une dataframe
  - à partir fichier csv

```
In [10]: df = pandas.read_csv('Heart.csv')
         df
```

Out[10]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed	No
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0	normal	Yes
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0	reversable	Yes
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0	normal	No
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0	normal	No
5	6	56	1	nontypical	120	236	0	0	178	0	0.8	1	0.0	normal	No
6	7	62	0	asymptomatic	140	268	0	2	160	0	3.6	3	2.0	normal	Yes
7	8	57	0	asymptomatic	120	354	0	0	163	1	0.6	1	0.0	normal	No
8	9	63	1	asymptomatic	130	254	0	2	147	0	1.4	2	1.0	reversable	Yes
9	10	53	1	asymptomatic	140	203	1	2	155	1	3.1	3	0.0	reversable	Yes

Python et l'analyse de données - Pandas

- Dataframe
  - Plusieurs manière de créer une dataframe
  - à partir d'une base de données, à travers une requête SQL
  - …etc.

In [13]:
```
import pandas as pd
from sqlalchemy import *
from sqlalchemy.orm import sessionmaker
from sqlalchemy.types import TIMESTAMP
from sqlalchemy.exc import IntegrityError
from sqlalchemy.sql import select
from sqlalchemy import desc


#Loading data
database_cnf = 'mysql+pymysql://root:@localhost:3306/examen'

engine = create_engine(database_cnf, echo=False)
conn = engine.raw_connection()

df = pd.read_sql(sql='SELECT * FROM `compte`', con=conn)
df
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pymysql\cursors.py
E9)' for column 'VARIABLE_VALUE' at row 480")
  result = self._query(query)
```

Out[13]:

	id	id_client	num_agence	solde
0	1	1	1	22000
1	2	1	1	35000
2	3	2	1	7000

Python et l'analyse de données - Pandas

- Les opérations sur les dataframes
  - Accès par colonnes
  - Accès par ligne

```
In [18]: df[['Age']]
```

Out[18]:

	Age
0	63
1	67
2	67
3	37
4	41
5	56
6	62

```
In [27]: df.loc[[0]]
```

Out[27]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed	No

Python et l'analyse de données - Pandas

- Les opérations sur les dataframes
  - Ajout de colonnes et de lignes

```
In [32]: df['blabla'] = None
         df['doubleAge'] = df['Age'] * 2
         df
```

Out[32]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD	blabla	doubleAge
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed	No	None	126
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0	normal	Yes	None	134
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0	reversable	Yes	None	134
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0	normal	No	None	74
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0	normal	No	None	82
5	6	56															
6	7	62															

```
In [37]: df.loc[304] = [304, 99, 1, 'nonanginal', 138 , 75, 0,0, 173,0 ,0.0 ,1 , 'NaN' ,'normal' , 'No', None, 198]
         df
```

294	295	63	0	asymptomatic	124	197	0	0	136	1	0.0	2	0	normal	Yes	None	126
295	296	41	1	nontypical	120	157	0	0	182	0	0.0	1	0	normal	No	None	82
296	297	59	1	asymptomatic	164	176	1	2	90	0	1.0	2	2	fixed	Yes	None	118
297	298	57	0	asymptomatic	140	241	0	0	123	1	0.2	2	0	reversable	Yes	None	114
298	299	45	1	typical	110	264	0	0	132	0	1.2	2	0	reversable	Yes	None	90
299	300	68	1	asymptomatic	144	193	1	0	141	0	3.4	2	2	reversable	Yes	None	136
300	301	57	1	asymptomatic	130	131	0	0	115	1	1.2	2	1	reversable	Yes	None	114
301	302	57	0	nontypical	130	236	0	2	174	0	0.0	2	1	normal	Yes	None	114
302	303	38	1	nonanginal	138	175	0	0	173	0	0.0	1	NaN	normal	No	None	76
304	304	99	1	nonanginal	138	75	0	0	173	0	0.0	1	NaN	normal	No	None	198

304 rows × 17 columns

Python et l'analyse de données - Pandas

- D'autres opérations avancées:
  - description statistiques: moyenne (mean), min, max, sum, describe, …etc
- Visualisation
  - ggplot

Python - ggplot

- Grammar Graphics Plot
- Créé pour permettre de créer « facilement » des graphe pour visualiser les données et leurs caractèristiques
- Écrit initialement pour R, mais utilisé dans Python
- Les pré-requis
  - matplotlib
  - numpy
  - scipy
  - statmodels
- Installation
  - pip install ggplot

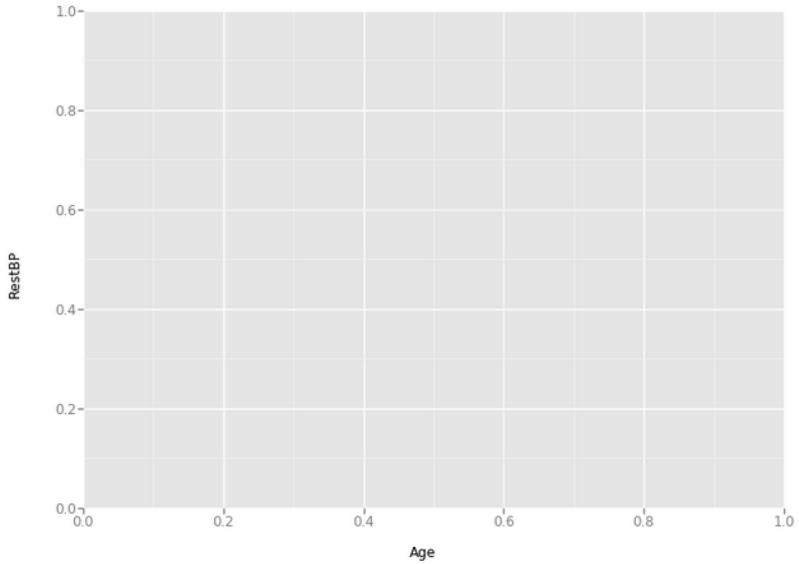
Python - ggplot

- Création d'un graphique avec python ggplot – principe :
  1. Spécification des données et des axes
  2. Description des axes et leur apparence
  3. Ajout de couche sur le graphe

Python - ggplot

- Spécification des données et des axes

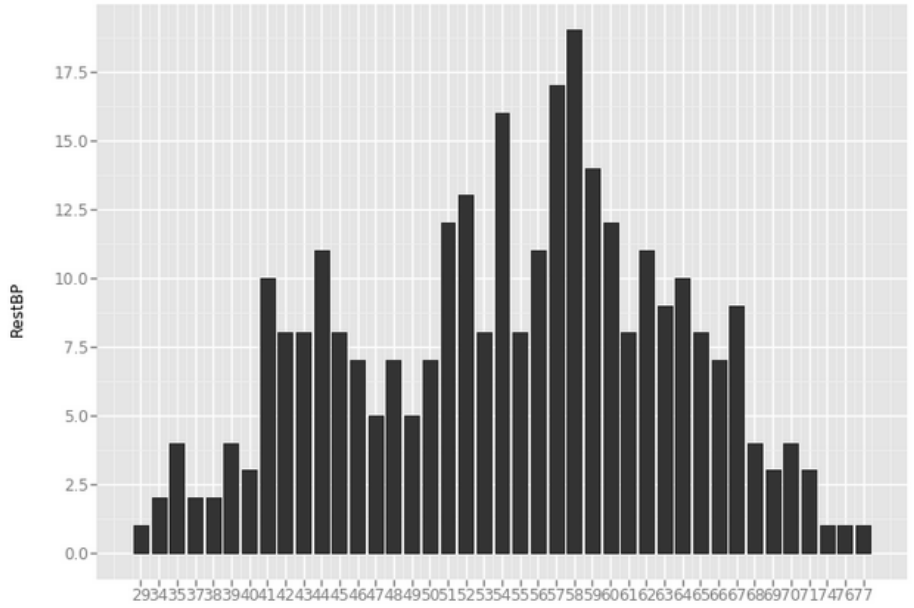
```
In [14]: from ggplot import *
         p = ggplot(df, aes(x='Age', y='RestBP'))
         p
```



Python - ggplot

- Spécification des données et des axes
- Description des axes et leur apparence

```
from ggplot import *
p = ggplot(df, aes(x='Age', y='RestBP'))
p = p + geom_bar()
p
```

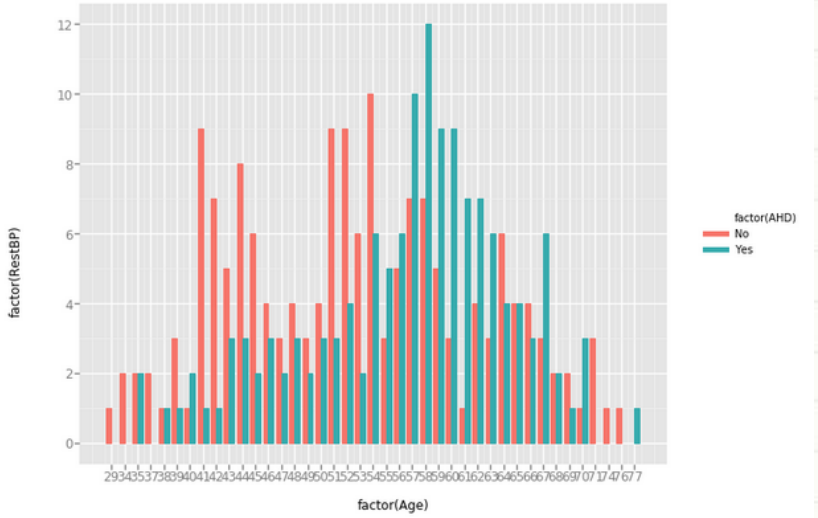


Python - ggplot

- Spécification des données et des axes
- Description des axes et leur apparence
- Ajout de couche sur le graphe

```
In [5]: from ggplot import *
        p = ggplot(aes(x='factor(Age)', y='factor(RestBP)', fill='factor(AHD)'), data=df)
        p = p + geom_bar()
        p
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ggplot\ggplot.py:602: FutureWarning: sort(col
, use sort_values(by=.....)
  fill_levels = self.data[[fillcol_raw, fillcol]].sort(fillcol_raw)[fillcol].unique()
```



Python - ggplot

- Spécification des données et des axes
- Description des axes et leur apparence
- Ajout de couche sur le graphe

```
In [25]: from ggplot import *
         p = ggplot(aes(x='Age', y='MaxHR'), data=df)
         p = p + geom_point()
         p = p + stat_smooth()
         p
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ggplot\stats\stat_smooth.py:77: FutureWarning: sort(
t_values(by=.....)
  smoothed_data = smoothed_data.sort('x')
```