Data Mining

John Samuel
CPE Lyon

Year: 2020-2021
Email: john(dot)samuel(at)cpe(dot)fr

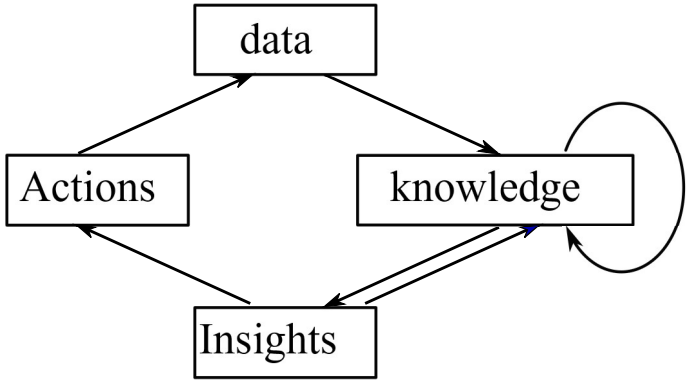
Objectifs

- Cycle de vie des données
- Acquistion, extraction, transformation de données
- Stockage de données
- ETL
- Analyses de données
- Visualisation de données

1. Cycle de vie des données

Cycle de vie des données

1. Données
2. Connaissances
3. Perspectives
4. Actions

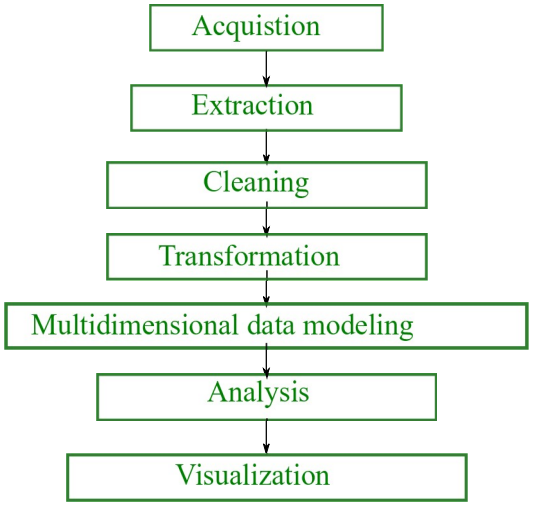


Cycle de vie des données

1. Cycle de vie des données

1.1. Des données à la connaissance

1. Acquisition de données
2. Extraction de données
3. Nettoyage de données
4. Transformation de données
5. Stockage de données
6. Data analysis modeling
7. Analyses de données
8. Visualisation de données

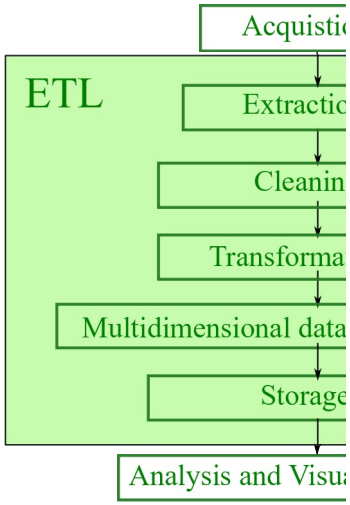


Des données à la connaissance

1.1.1. Acquisition de données

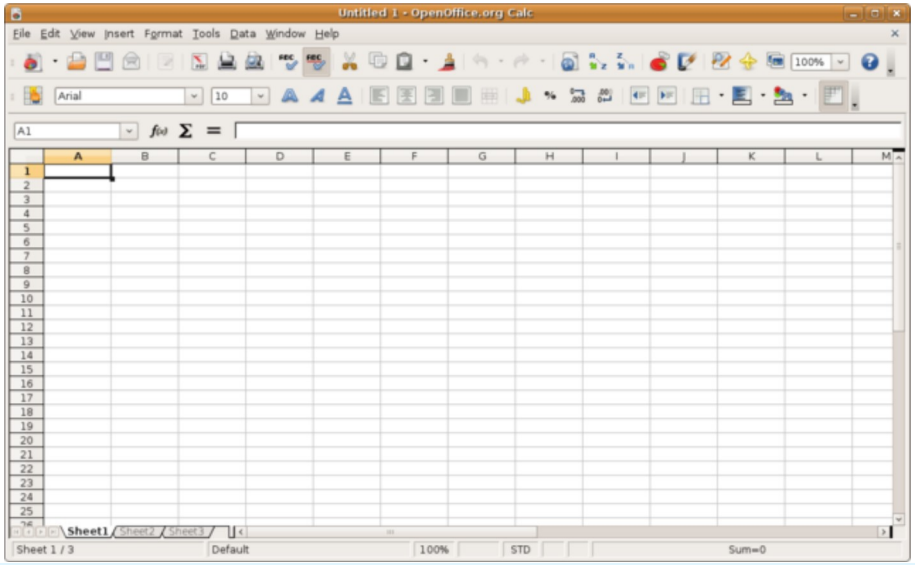
1.1.2. ETL (Extraction Transformation, Loading)

1. Extraction de données
2. Nettoyage de données
3. Transformation de données
4. Chargement des données dans les entrepôts de données



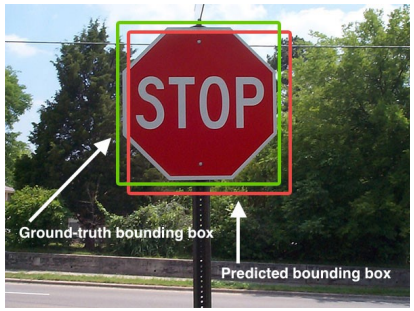
ETL (Extraction, Transforma

1.1.3. Analyses de données

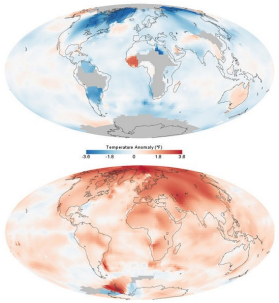


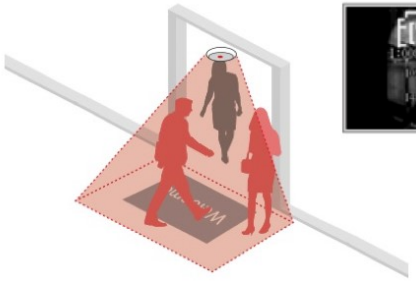
1.1.3. Analyses de données

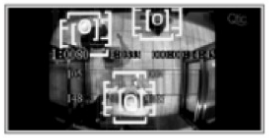
1.1.4. Visualisation de données



1.1.4. Visualisation de données







L'acquisition de données [Lenzerini 2002][Dong 2013]

1. Questionnaires
   - Questionnaires face à face
   - Questionnaires en ligne
2. Capteurs[1]
   - Température, pression, humidité
   - Acoustique, navigation
   - Proximité, capteurs de présence
3. Réseau sociaux
4. Vidéo de surveillance
5. Web
6. Enregistrement

1. https://en.wikipedia.org/wiki/List_of_sensors

Formats de stockage de données

- Fichiers textuelles et binaires
- CSV/TSV
- XML
- JSON
- Média (Images/Audio/Vidéo)

Formats de stockage de données: JSON

```
[
  {
    "languageLabel": "ENIAC coding system",
    "year": "1943"
  },
  {
    "languageLabel": "ENIAC Short Code",
    "year": "1946"
  },
  {
    "languageLabel": "Von Neumann and Goldstine graphing system",
    "year": "1946"
  }
]
```

Formats de stockage de données: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <element>
      <languageLabel>ENIAC coding system</languageLabel>
      <year>1943</year>
   </element>
   <element>
      <languageLabel>ENIAC Short Code</languageLabel>
      <year>1946</year>
   </element>
   <element>
      <languageLabel>Von Neumann and Goldstine graphing system</languageLabel>
      <year>1946</year>
   </element>
</root>
```

Formats de stockage de données: CSV

```
languageLabel,year
ENIAC coding system,1943
ENIAC Short Code,1946
Von Neumann and Goldstine graphing system,1946
```

2.2 Types de stockage des données

1. Bases de données structurées
   - Bases de données relationnelles
   - Bases de données orientées objet
2. Bases de données non-structurées
   - Systèmes de fichiers
   - Systèmes de gestion de contenu (CMS)
   - Collections de documents
3. Bases de données semi-structurées
   - Systèmes de fichiers
   - Bases de données NoSQL

Paris is the capital of France. In 2015,
its population was recorded as 2,206,488

Country
Name Capital

Population
Value Year

```
<xml>
  <country>
    <name>France</name>
    <capital>
      <name>Paris</name>
      <population>
        <value>2,206,488</value>
          <year>2015</year>
      </population>
    </capital>
  </country>
</xml>
```

Unstructured vs. Structured vs. Semi-structured

2.3.1. Propriétés ACID[1]

- **Atomicité**: chaque transaction se fait au complet ou pas du tout
- **Cohérence**: Any transaction must bring database from one valid state to another.
- **Isolation**: Toute transaction doit amener la base de données d'un état valide à un autre.
- **Durabilité**: Indépendamment des pertes de puissance, des plantages, une transaction une fois engagée dans la base de données doit rester dans cet état.

1. https://fr.wikipedia.org/wiki/Propri%C3%A9t%C3%A9s_ACID

2.3.1. Propriétés ACID

- Assurer la validité des bases de données même en cas d'erreurs, de pannes de courant
- Important dans le secteur bancaire

2.2 Types de bases de données

- Bases de données relationnelles
- Base de données orientée objet
- **NoSQL**
- NewSQL

2.3.3. NoSQL

Théorème CAP[1]

Il est impossible sur un système informatique de calcul distribué de garantir en même temps (c'est-à-dire de manière synchrone) les trois contraintes suivantes

- **Cohérence**: tous les nœuds du système voient exactement les mêmes données au même moment
- **Disponibilité**: garantie que toutes les requêtes reçoivent une réponse. Chaque requête reçoit une réponse (non erronée), sans la garantie qu'elle contient l'écriture la plus récente
- **Tolérance au partitionnement**: Le système continue à fonctionner malgré un nombre arbitraire de messages qui sont abandonnés (ou retardés) par le réseau entre les nœuds

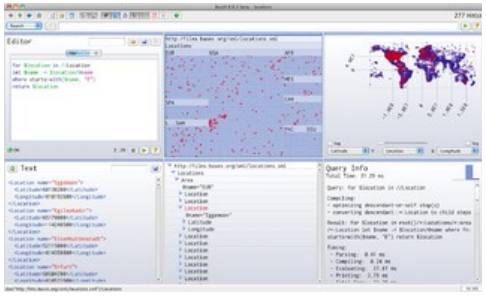
- compromis sur la cohérence
- priorité à la disponibilité et à la rapidité

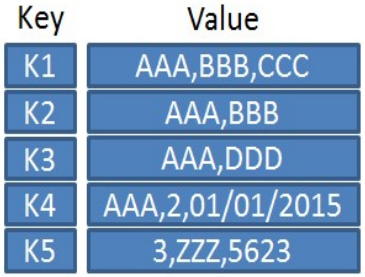
2.3.3. NoSQL

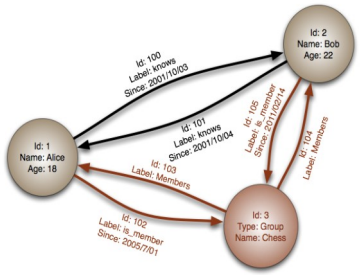
- compromis sur la cohérence
- priorité à la disponibilité et à la rapidité

2.3.3. Types de bases de données NoSQL

- Base de données orientée colonnes
- Base de données orientée documents
- Base de données clé-valeur
- Base de données orientée graphe







Exemple: un tableau dans une base de données relationnelles

num	languageLabel	year
1	ENIAC coding system	1943
2	ENIAC Short Code	1946
3	Von Neumann and Goldstine graphing system	1946

Exemple: base de données orientée colonnes

```
ENIAC coding system:1; ENIAC Short Code:2 Von Neumann and Goldstine graphing system:3

1943:1; 1946:2; 1946:3
```

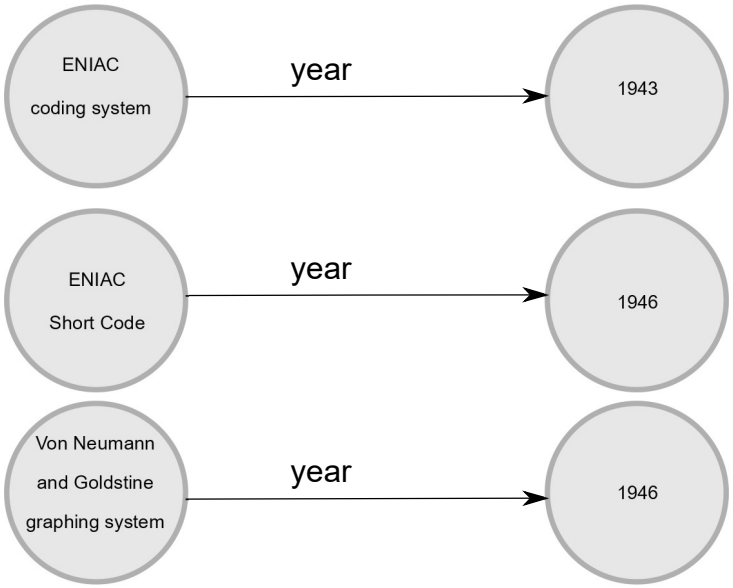
Exemple: base de données orientée documents

```
{
  "languageLabel": "ENIAC coding system",
  "year": "1943"
}

{
  "languageLabel": "ENIAC Short Code",
  "year": "1946"
}

{
  "languageLabel": "Von Neumann and Goldstine graphing system",
  "year": "1946"
}
```

Exemple: base de données orientée clé-valeur

identifiant	languageLabel,year
p1	ENIAC coding system,1943
p2	ENIAC Short Code,1946
p3	Von Neumann and Goldstine graphing system,1946

Exemple: base de données orientée graphe



base de données orientée graphe

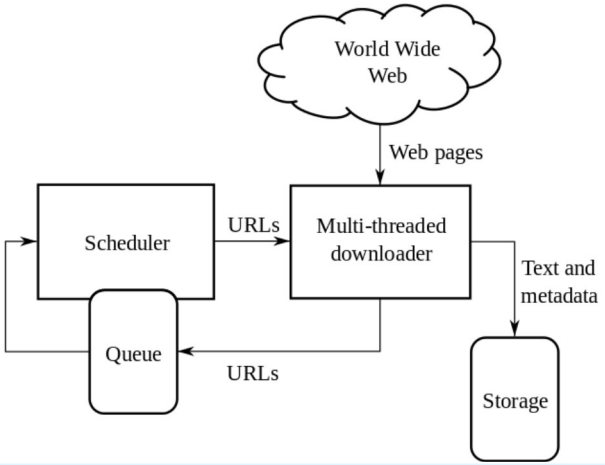
3.1. Techniques d'extraction des données

- Vidage de données (data dumps)
  - Téléchargement de données de vidange complètes
  - Téléchargement de vidanges sélectives de données
- Interrogation périodique des flux de données (par exemple, les blogs, les flux d'informations)
- Flux de données
  - Abonnement aux flux de données (notifications "push")

3.2. Interfaces d'interrogation

- Points terminaux de requête supportant les langues déclaratives
  - SQL
  - SPARQL
- Options de recherche (et de filtrage) manuelle automatisée

3.3. Crawlers pour les pages web



Web crawlers : naviguer dans l'ensemble en utilisant des hyperliens

3.4. Interface de programmation d'applications (API)

- Opérations Web (CRUD) pour manipuler les ressources gérées en externe
  - Create: créer
  - Read: lire
  - Update: mettre à jour
  - Delete: supprimer
- Requiert que les programmeurs développent des wrappers pour l'intégration des services web

3.4. Interface de programmation d'applications (API)

```
import requests
url = "https://api.github.com/users/johnsamuelwrites"

response = requests.get(url)
print(response.json())
```

4.1. Nettoyage de données

- Erreurs de syntaxe
- Erreurs sémantiques
- Erreurs de couverture

4.1.1. Erreurs de syntaxe

- Les erreurs lexicales (par exemple, l'utilisateur a saisi une chaîne de caractères au lieu d'un chiffre)
- Erreurs de format des données (par exemple, ordre du nom de famille, du prénom)
- Erreurs de données irrégulières (par exemple, utilisation de mesures différentes)

4.1.2. Erreurs sémantiques [Abedjan 2016]

- Violation des contraintes d'intégrité
- Erreurs de contradiction
- Erreurs de duplication
- Erreurs de donnée invalide

4.1.3. Erreurs de couverture

- Valeur manquante
- Donnée manquante

See also:

`DataFrame.isna`
Indicate missing values.

`DataFrame.notna`
Indicate existing (non-missing) values.

`DataFrame.fillna`
Replace missing values.

`Series.dropna`
Drop missing values.

`Index.dropna`
Drop missing indices.

Exemple: Pandas

4.2.1. Traitement des erreurs syntaxiques

- Validation à l'aide d'un schéma (par exemple, XSD, JSONP)
- Transformation de données

4.2.1. Traitement des erreurs syntaxiques: XSD

```
<xs:schema attributeFormDefault="unqualified"
      elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="root" type="rootType"/>
  <xs:complexType name="elementType">
    <xs:sequence>
      <xs:element type="xs:string" name="languageLabel"/>
      <xs:element type="xs:short" name="year"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="rootType">
    <xs:sequence>
      <xs:element type="elementType" name="element" maxOccurs="unbounded" minOccurs="
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

4.2.2. Traitement des erreurs sémantiques

- Élimination des doublons à l'aide de techniques telles que la spécification de contraintes d'intégrité comme les dépendances fonctionnelles

num	languageLabel	year
1	ENIAC coding system	1943

$$num \rightarrow languageLabel$$

$$languageLabel \rightarrow year$$

$$num \rightarrow year$$

4.2.3. Traitement des erreurs de couverture

- Techniques d'interpolation
- Utilisation de sources de données externes pour les vérifications croisées

4.2.4. Administrateurs et traitement des erreurs

- Retour d'information des utilisateurs pour correction (par exemple, OpenStreetMap, Wikipedia, etc.)
- Alertes et déclencheurs en cas d'ajout d'informations incohérentes

5.1 Langages de programmation

- Langues des templates
- XSLT
- AWK
- Sed
- Langages de programmation comme PERL

6.1. ETL (Extraction Transformation and Loading)

1. Extraction des données
2. Nettoyage des données
3. Transformation des données
4. Chargement des données dans les entrepôts de données

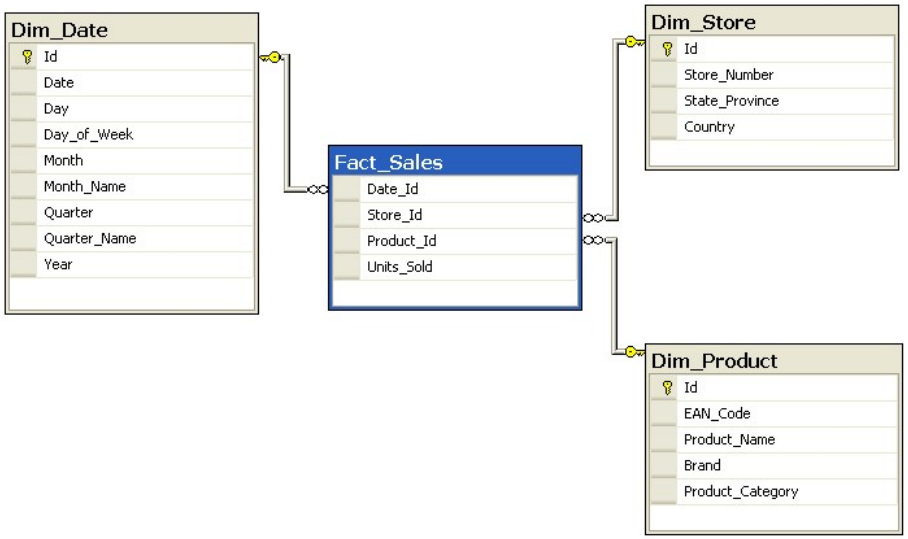
6.2.1. Analyse des données multidimensionnelles

- Analyse des données multidimensionnelles
  - dimensions
    - attributes
    - niveaux
    - hiérarchies
  - faits
    - mesures

6.2.1. Analyse des données multidimensionnelles

- Analyse des données multidimensionnelles
  - dimensions (par exemple,spatio-temporelles dimensions, produits)
    - attributes (par exemple, nom, fabricant, etc.)
    - niveaux (par exemple, jour, mois, trimestre, magasin, ville, pays, etc.)
    - hiérarchies (par exemple, jour-mois-trimestre-année, magasin-ville-pays, etc.)
  - faits
    - mesures (par exemple, le nombre de produits vendus/non vendus)

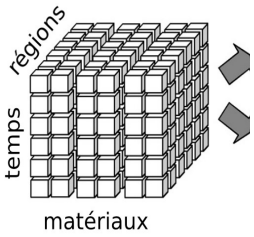
6.2.3. Modèle de données en étoile



Modèle de données en étoile

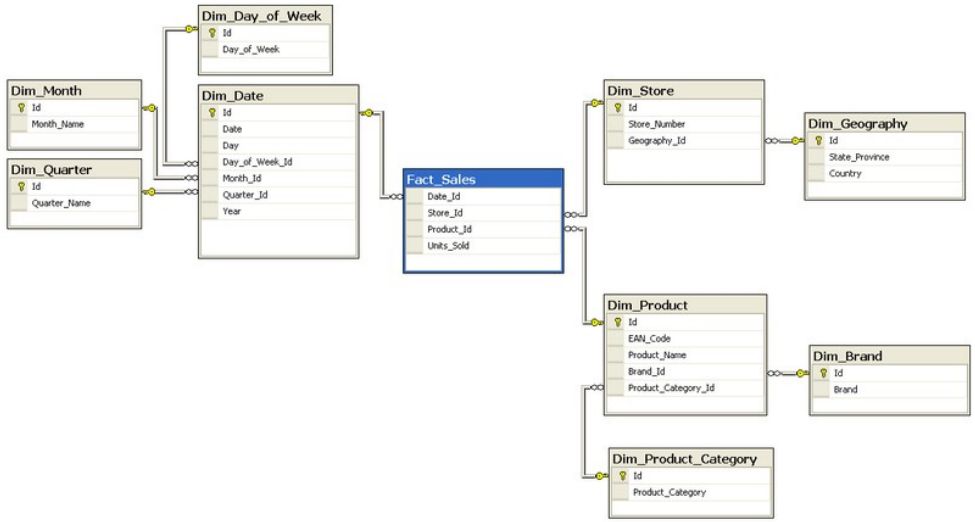
6.2.3. Cubes ou hypercube de données

- Cubes de données pour le traitement analytique en ligne (OLAP)
- Opérations du cube OLAP
  - **Slice**: extraction d'une tranche d'information
  - **Dice**: extraction d'un bloc de données (opération plus générale que le slicing),
  - **Drill up**: synthèse des informations en fonction d'une dimension
  - **Drill down**: opération inverse du drill-up
  - **Pivot**: sélection du couple de dimensions qui formera le résultat de la requête



Mod

6.2.4. Modèle de données en flocon



6.2. ETL: d'une base de données à l'autre

- De : Sources de données
  - Bases de données internes ou externes
  - Services web
- À : Entrepôts de données
  - Entrepôts de données d'entreprise
  - Entrepôts Web

Activités d'analyse des données

1. Récupération des valeurs
2. Filtrer
3. Calculer les valeurs dérivées
4. Trouver l'extremum
5. Trier
6. Déterminer la limite
7. Caractériser la distribution
8. Trouver des anomalies
9. Cluster
10. Corréler
11. Contextualisation

1. https://en.wikipedia.org/wiki/Data_analysis

8. Visualisation des données

8.1.Les variables visuelles [Jacques Bertin]

1. position
2. taille
3. forme
4. valeur
5. couleur
6. orientation
7. texture

1. https://en.wikipedia.org/wiki/Visual_variable

8. Visualisation des données

8.1. Visualisation des données

1. séries temporelles
2. classement
3. partie à l'ensemble
4. écart
5. triage
6. distribution des fréquences
7. corrélation
8. comparaison nominale
9. géographique ou géospatial

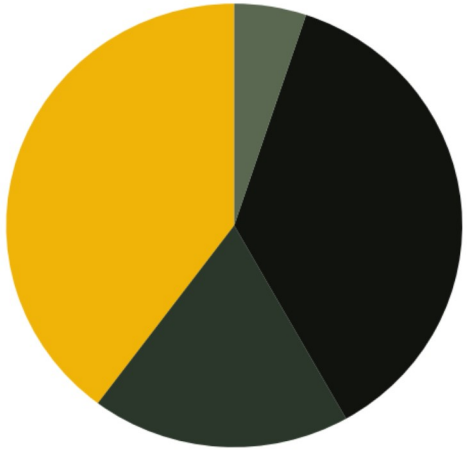
1. https://en.wikipedia.org/wiki/Data_visualization

8.2. Visualisation des données: Exemples

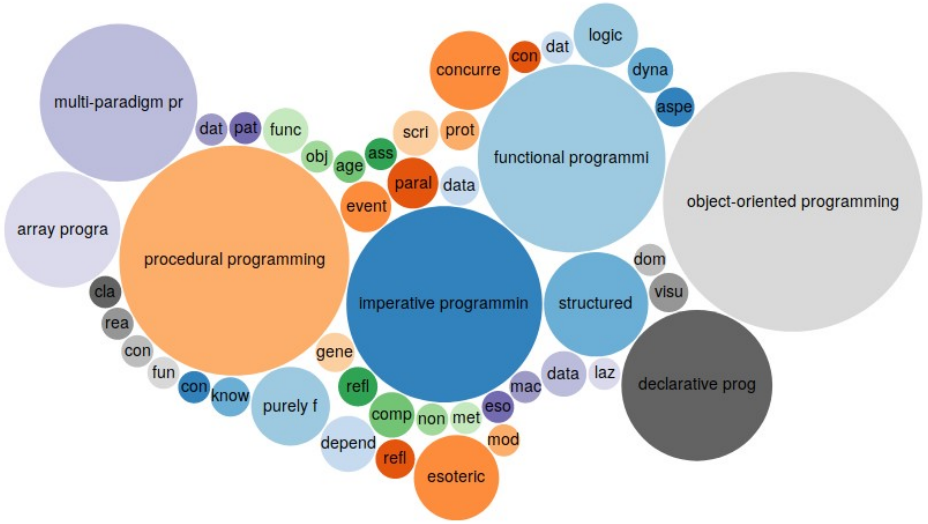
1. Diagramme en bâtons (comparaison nominale)
2. Diagramme circulaire (partie à l'ensemble)
3. Histogramme (distribution des fréquences)
4. Nuage de points (corrélation)
5. Réseaux
6. Graphique linéaire (séries temporelles)
7. Arborescence
8. Diagramme de Gantt
9. Carte thermique/heatmap

1. https://fr.wikipedia.org/wiki/Repr%C3%A9sentation_graphique_de_donn%C3%A9es

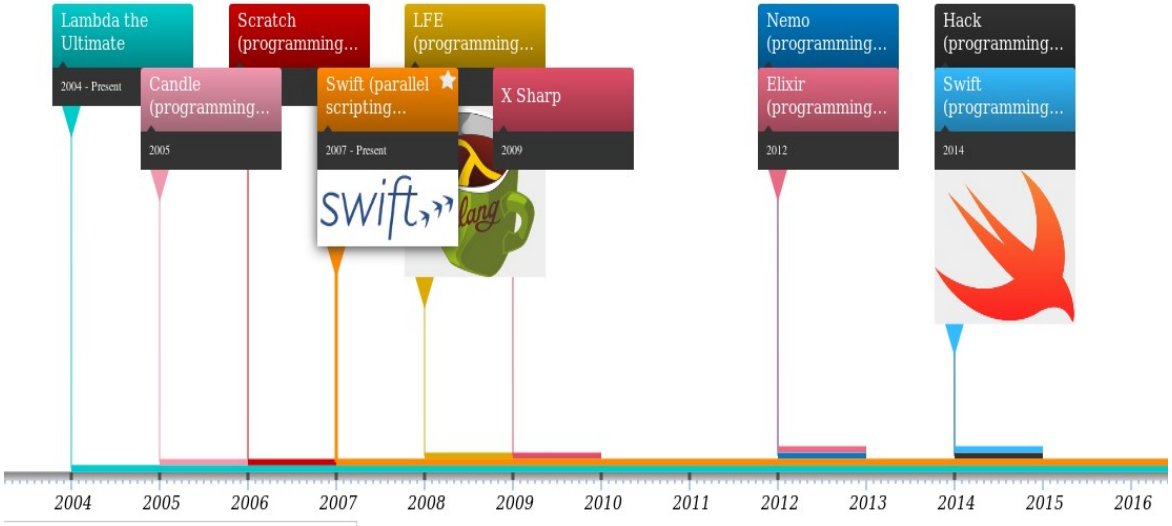
Diagramme circulaire



8. Visualisation des données



Les paradigmes du langage de programmation (diagramme à bulles)



Historique des langages de programmation (Histropedia)

8. Visualisation des données

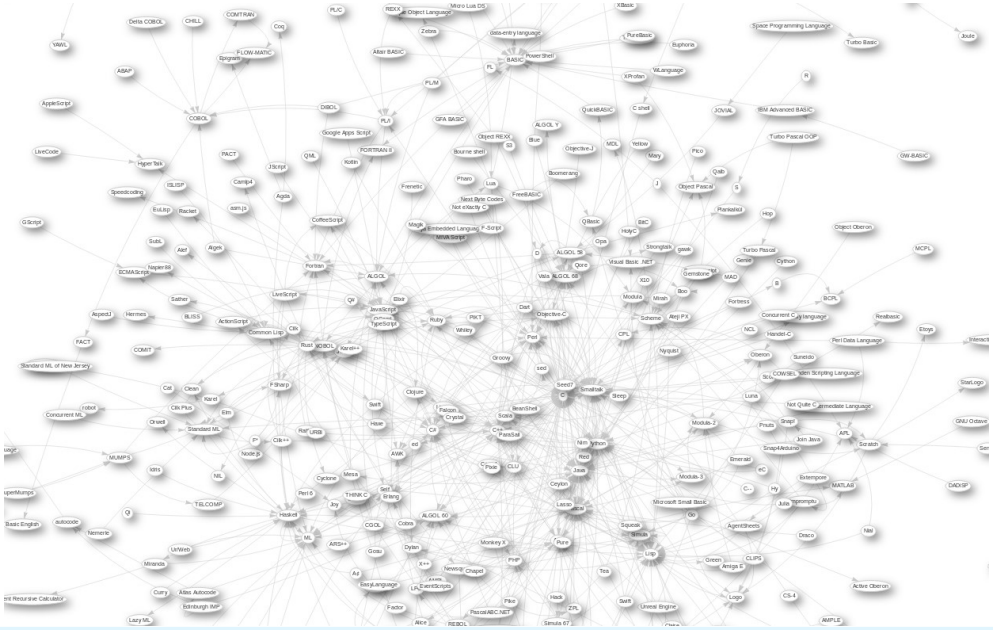
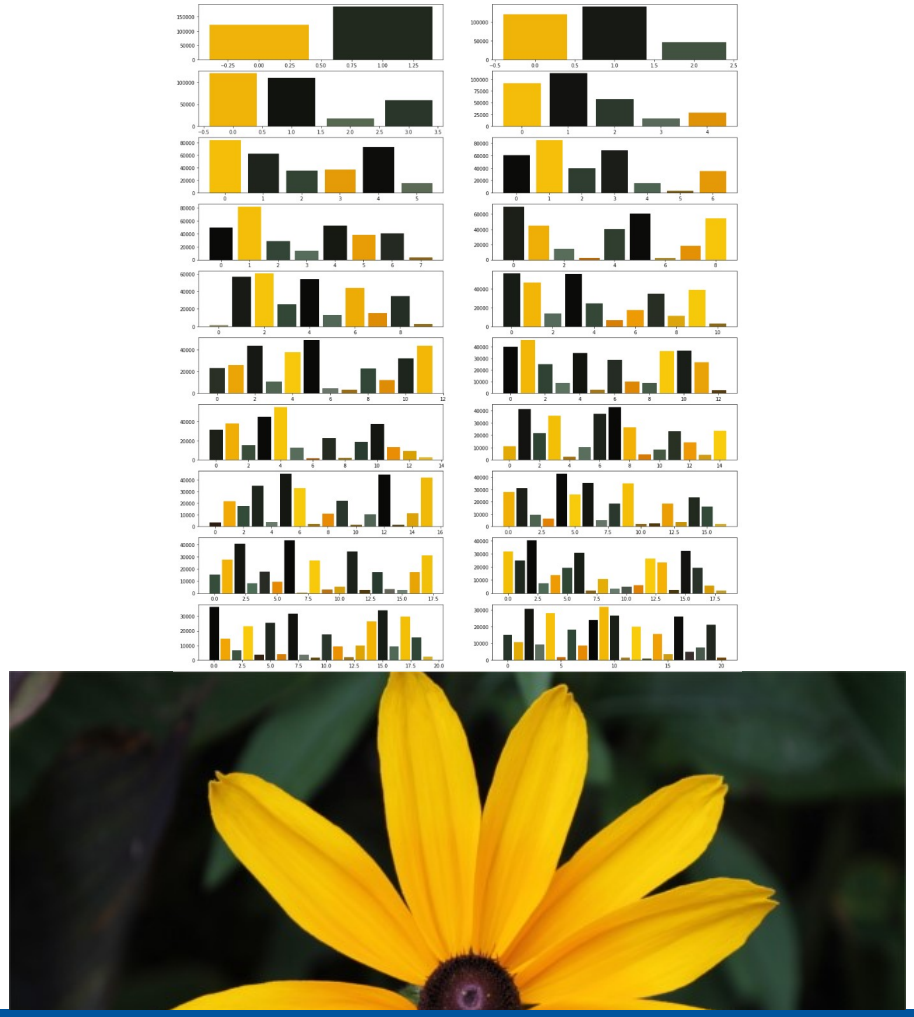
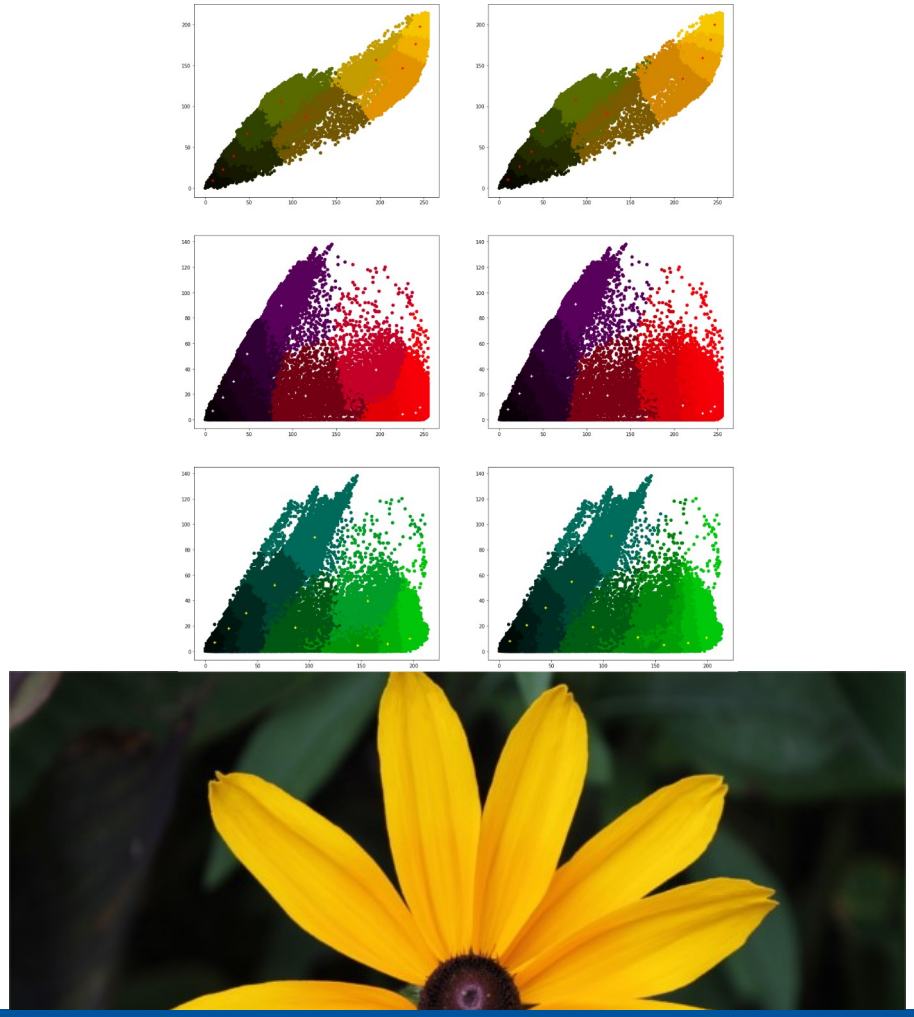


Diagramme d'influence des langages de programmation

k couleurs prédominantes



Diagrammes de dispersion RGB (Comparaison)



Références

Articles de recherche

- **[Abedjan 2016]** Abedjan, Ziawasch, et al. Detecting Data Errors: Where Are We and What Needs to Be Done? VLDB Endowment, 1 Aug. 2016.
- **[Dong 2013]** Dong, Xin Luna, and Divesh Srivastava. “Big Data Integration.” 2013 IEEE 29th International Conference on Data Engineering (ICDE), 2013, pp. 1245–48. IEEE Xplore
- **[Lenzerini 2002]** Lenzerini, Maurizio. “Data Integration: A Theoretical Perspective.” Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Association for Computing Machinery, 2002, pp. 233–246. ACM Digital Library

Références

Couleurs

- Color Tool - Material Design

Images

- Wikimedia Commons