# Analyse exploratoire des données

*EDA*

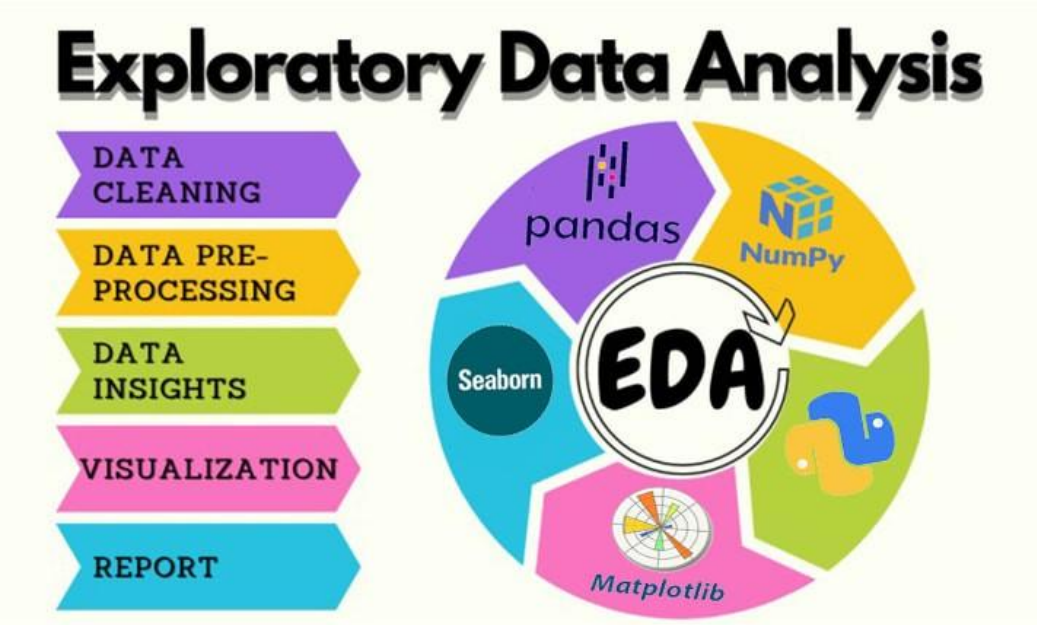
# Introduction à l’EDA

# Qu'est-ce que l'EDA ? Exploratory Data Analysis

L'EDA est un **processus itératif** d'investigation des données pour en comprendre les principales caractéristiques, identifier des tendances, des relations et des anomalies.

L'EDA est une étape cruciale **dans tout projet de Data Science**, car elle permet de poser les bonnes questions et de choisir les méthodes d'analyse et de modélisation les plus appropriées.

# Qu'est-ce que l'EDA ? Exploratory Data Analysis



# Pourquoi faire de l'EDA ?

***Comprendre les données:***

- Identifier les types de variables (catégorielles, numériques, etc.)
- Visualiser les distributions des variables
- Détecter les valeurs manquantes et les outliers

***Identifier des patterns :***

- Découvrir des relations entre les variables
- Identifier des tendances ou des groupes d'observations

***Préparer les données pour la modélisation :***

- Sélectionner les variables pertinentes
- Transformer les données si nécessaire (normalisation, encodage, etc.)
- Détecter des problèmes potentiels (déséquilibre des classes, multicolinéarité, etc.)

# Chargement et exploration des données

# Pandas

Pandas, librairie incontournable pour la manipulation et l'analyse de données en Python

Fonctionne grâce aux ***DataFrame*** Structure de données bidimensionnelle (tableaux) pour organiser et manipuler les données

Fonctions clés:

- **pd.read_csv():** Charger des données depuis un fichier CSV
- **pd.read_excel():** Charger des données depuis un fichier Excel
- Autres fonctions pour lire différents formats de données


pandas Getting started User Guide API reference Development Release notes Search Ctrl + K 2.2 (stable)

- Input/output
- General functions
- Series
- DataFrame
  - pandas.DataFrame
  - pandas.DataFrame.index
  - pandas.DataFrame.columns
  - pandas.DataFrame.dtypes
  - pandas.DataFrame.info
  - pandas.DataFrame.select_dtypes
  - pandas.DataFrame.values
  - pandas.DataFrame.axes
  - pandas.DataFrame.ndim
  - pandas.DataFrame.size
  - pandas.DataFrame.shape
  - pandas.DataFrame.memory_usage
  - pandas.DataFrame.empty
  - pandas.DataFrame.set_flags
  - pandas.DataFrame.astype
  - pandas.DataFrame.convert_dtypes

API reference > DataFrame > pandas.DataFrame

On this page
DataFrame

Show Source

# pandas.DataFrame

*class* pandas.DataFrame(*data=None, index=None, columns=None, dtype=None, copy=None*) # [source]

Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure.

**Parameters:**

**data** : *ndarray (structured or homogeneous), Iterable, dict, or DataFrame*

Dict can contain Series, arrays, constants, dataclass or list-like objects. If data is a dict, column order follows insertion-order. If a dict contains Series which have an index defined, it is aligned by its index. This alignment also occurs if data is a Series or a DataFrame itself. Alignment is done on Series/DataFrame inputs.
If data is a list of dicts, column order follows insertion-order.

**index** : *Index or array-like*

Index to use for resulting frame. Will default to RangeIndex if no indexing information part of input data and no index provided.

**columns** : *Index or array-like*

# Premier aperçu des données

**df.head():** Afficher les premières lignes du ***DataFrame*** pour avoir un aperçu rapide de sa structure et de son contenu

**df.info():** Obtenir des informations sur les colonnes (noms, types de données, nombre de valeurs non nulles)

**df.describe():** Calculer des statistiques descriptives de base pour les colonnes numériques (moyenne, écart-type, quartiles, etc.)

**df.shape:** Obtenir les dimensions du ***DataFrame*** (nombre de lignes et de colonnes)

Documentation de la fonction head :
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.head.html

```
>>> df = pd.DataFrame({'animal': ['alligator', 'bee', 'falcon', 'lion',
...                               'monkey', 'parrot', 'shark', 'whale', 'zebra']})
>>> df
      animal
0  alligator
1        bee
2     falcon
3       lion
4     monkey
5     parrot
6      shark
7      whale
8      zebra
```

```
>>> df.head()
      animal
0  alligator
1        bee
2     falcon
3       lion
4     monkey
```

```
>>> df.head(3)
      animal
0  alligator
1        bee
2     falcon
```

# Identifier les types de variables

**Variables catégorielles** : Représentent des catégories ou des groupes (ex: Type de Pokémon, Arrondissement)

**Variables numériques :** Représentent des quantités mesurables (ex: Attaque, Défense, Prix)

# Identifier les types de variables

**Variables catégorielles** : Représentent des catégories ou des groupes (ex: Type de Pokémon, Arrondissement)

**Variables numériques :** Représentent des quantités mesurables (ex: Attaque, Défense, Prix)

⇒ Le type de variable influence les analyses et les visualisations possibles

# Gérer les valeurs manquantes

Identification: **df.isnull().sum()** pour compter les valeurs manquantes par colonne

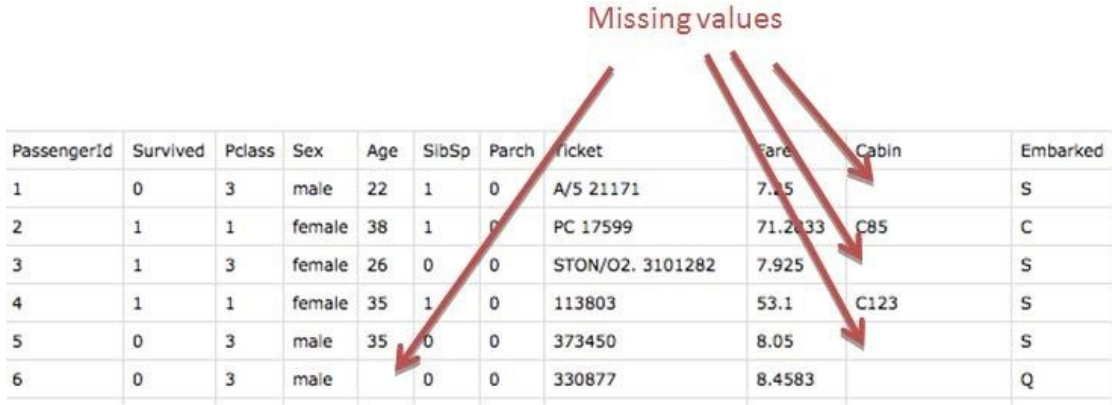
Visualisation: Heatmap pour visualiser la répartition des valeurs manquantes

**Stratégies de traitement :**

- Suppression des lignes ou des colonnes avec trop de valeurs manquantes
- Imputation des valeurs manquantes (moyenne, médiane, etc.)
- Utilisation d'algorithmes plus avancés pour l'imputation

# Exemples

| | First Score | Second Score | Third Score | Fourth Score |
|---|---|---|---|---|
| 0 | 100.0 | 30.0 | 52.0 | 60 |
| 1 | NaN | NaN | NaN | 67 |
| 2 | NaN | 45.0 | 80.0 | 68 |
| 3 | 95.0 | 56.0 | 98.0 | 65 |




| PassengerId | Survived | Pclass | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | male | 22 | 1 | 0 | A/5 21171 | 7.25 | | S |
| 2 | 1 | 1 | female | 38 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 1 | 3 | female | 26 | 0 | 0 | STON/O2. 3101282 | 7.925 | | S |
| 4 | 1 | 1 | female | 35 | 1 | 0 | 113803 | 53.1 | C123 | S |
| 5 | 0 | 3 | male | 35 | 0 | 0 | 373450 | 8.05 | | S |
| 6 | 0 | 3 | male | | 0 | 0 | 330877 | 8.4583 | | Q |

https://miro.medium.com/v2/resize:fit:737/0*z9UFoLa8awMxyTR7.png

# Détecter et traiter les outliers

Les valeurs extrêmes (**outliers**) qui peuvent biaiser les analyses et les modèles

***Méthodes de détection :***

- Visualisation : Boxplots, histogrammes
- Méthodes statistiques : Z-score, IQR (Interquartile Range)

***Stratégies de traitement :***

- Suppression des outliers
- Transformation des données (logarithme, racine carrée, etc.)
- Winsorisation (remplacer les outliers par des valeurs seuils)

# Analyse univariée

# Analyse Univariée

L'**analyse univariée** consiste à étudier une seule variable à la fois pour en comprendre la distribution, les caractéristiques principales et détecter d'éventuelles anomalies.

C'est la première étape pour se familiariser avec chaque variable de votre jeu de données et poser les bases pour des analyses plus complexes.

# Variables numériques

# Statistiques descriptives : les chiffres clés

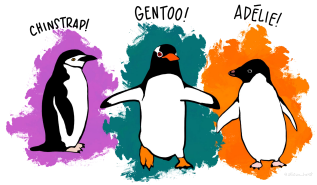
- ***Mesures de tendance centrale :***
  - Moyenne: La valeur "moyenne" de la variable
  - Médiane: La valeur qui sépare les données en deux moitiés égales
  - Mode: La valeur la plus fréquente (pour les variables catégorielles)
- ***Mesures de dispersion :***
  - Étendue: La différence entre la valeur maximale et la valeur minimale
  - Écart-type: Mesure la dispersion des valeurs autour de la moyenne
  - Quartiles: Divisent les données en quatre parties égales

# Visualisation des variables numériques

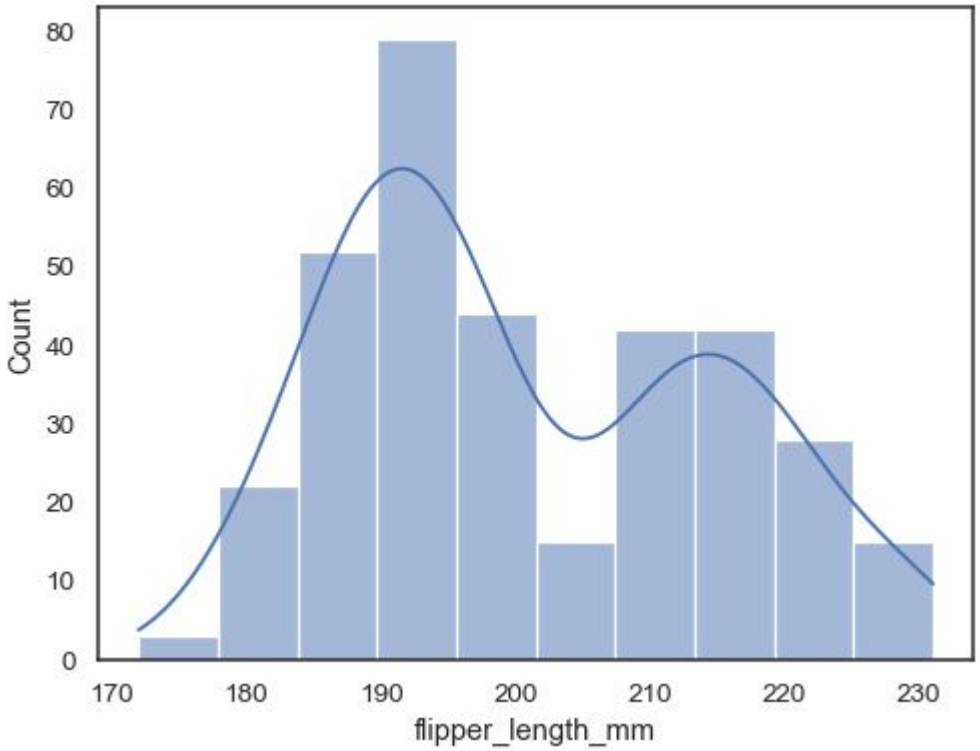
**Histogrammes :** Représentent la distribution d'une variable numérique en regroupant les valeurs en intervalles (bins)

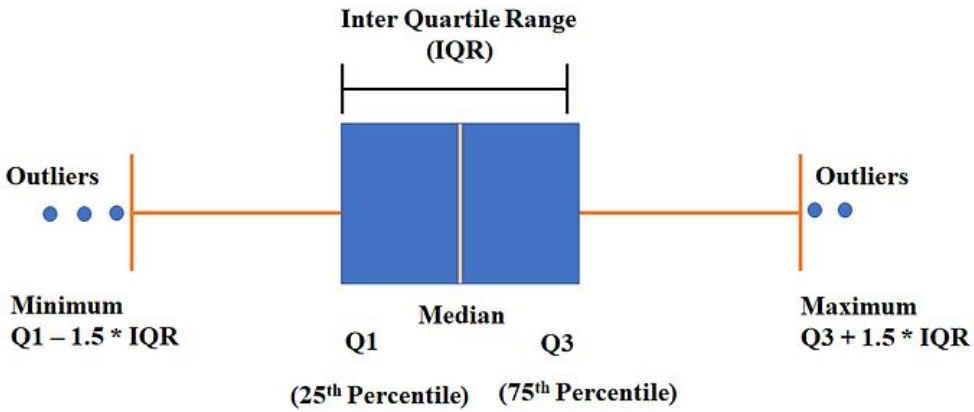
**Boxplots** (diagrammes en boîtes à moustaches): Montrer la médiane, les quartiles, les valeurs extrêmes et d'éventuels outliers

# *Exemple histogramme (penguins dataset)*

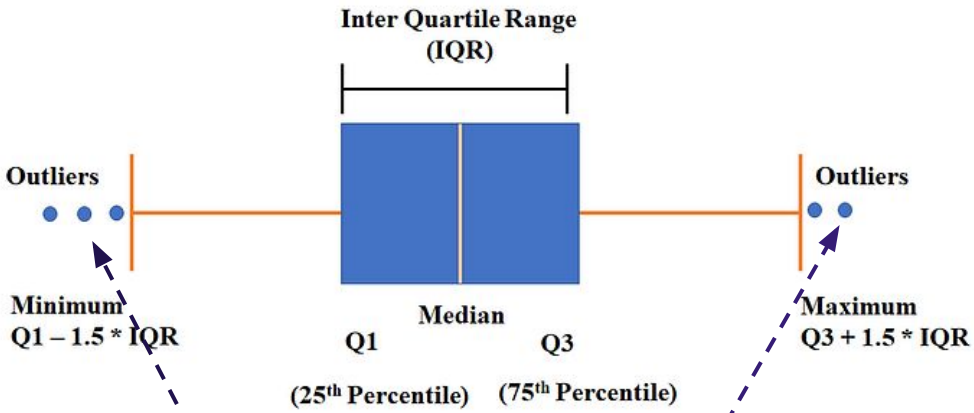



Avec un histogramme, on représente la distribution d'une variable **numérique**




Le boxplot montre la distribution d'une variable **numérique** avec des statistiques clés (quartiles, maximum, médiane...)




Visualisation des valeurs aberrantes ici

Le boxplot montre la distribution d'une variable **numérique** avec des statistiques clés (quartiles, maximum, médiane...)

# Variables catégorielles

# Analyse des variables catégorielles

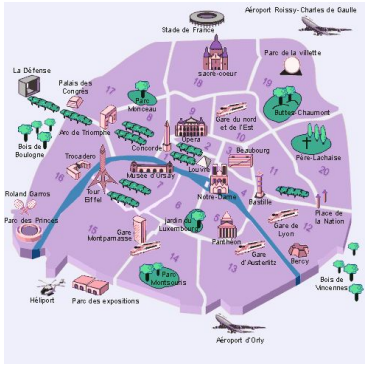
**Fréquences** : Compter le nombre d'occurrences de chaque catégorie

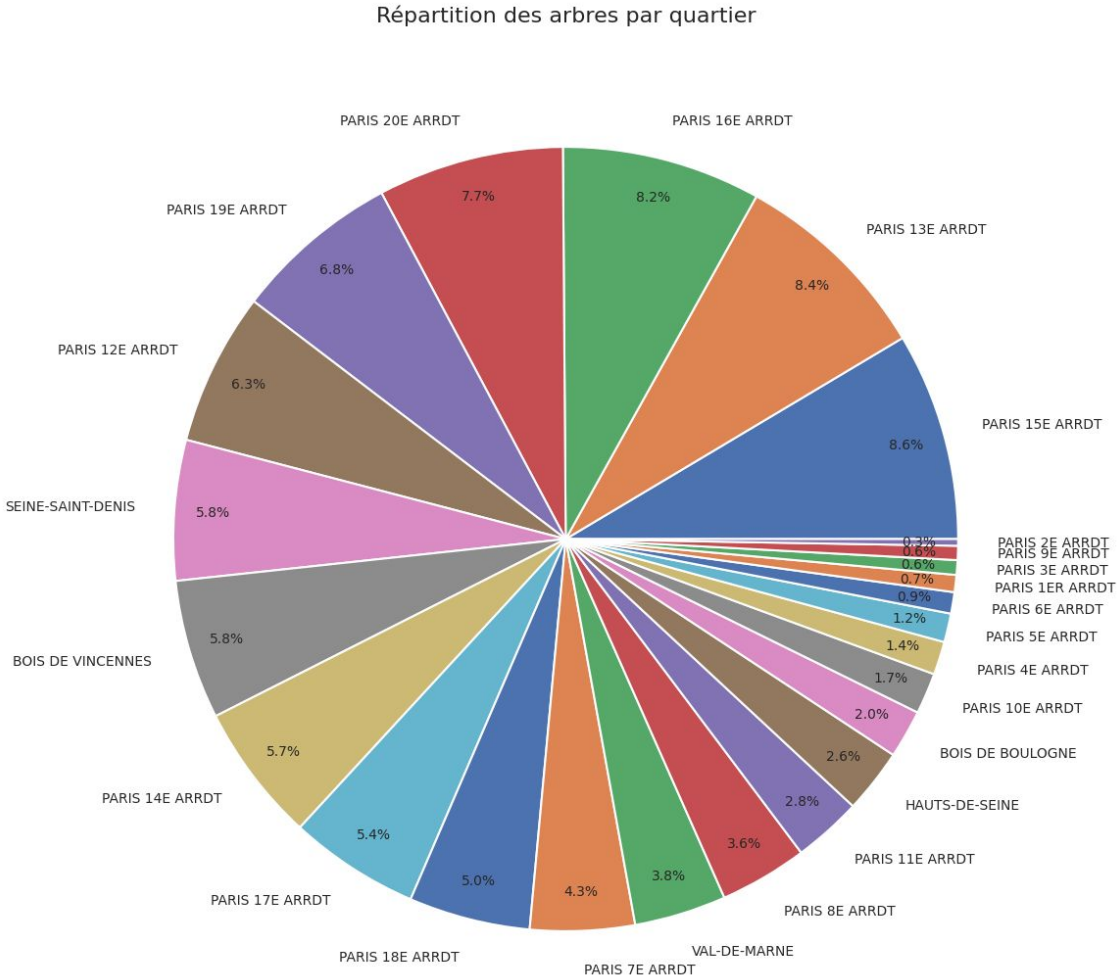
**Proportions** : Calculer la proportion de chaque catégorie par rapport au total

Diagrammes en secteurs (**pie charts**): Visualiser les proportions de chaque catégorie

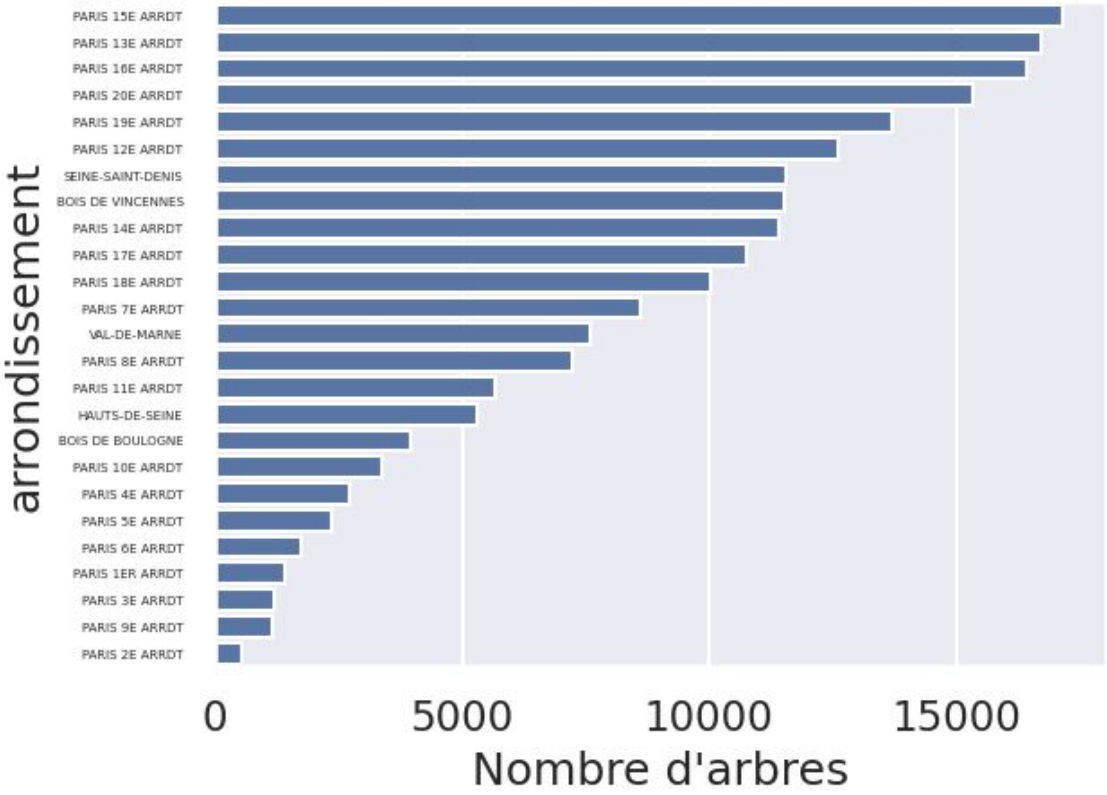
Les **diagrammes en barres :** Visualisent les fréquences ou les proportions des différentes catégories d'une variable catégorielle

Avec un **piechart**, on visualise les proportions de chaque catégorie



Avec un diagramme en barre, on représente comment se répartit une variable **catégorielle**




*Exemple barplot (localisation des arbres de Paris)*

# Analyse bivariée

# Analyse Bivariée

L'**analyse bivariée** étudie la relation entre deux variables pour comprendre comment elles évoluent ensemble, s'il existe une association ou une dépendance entre elles.

Elle permet de découvrir des liens intéressants, de formuler des hypothèses et de préparer le terrain pour des analyses multivariées plus complexes.

# Catégorielles vs. catégorielles

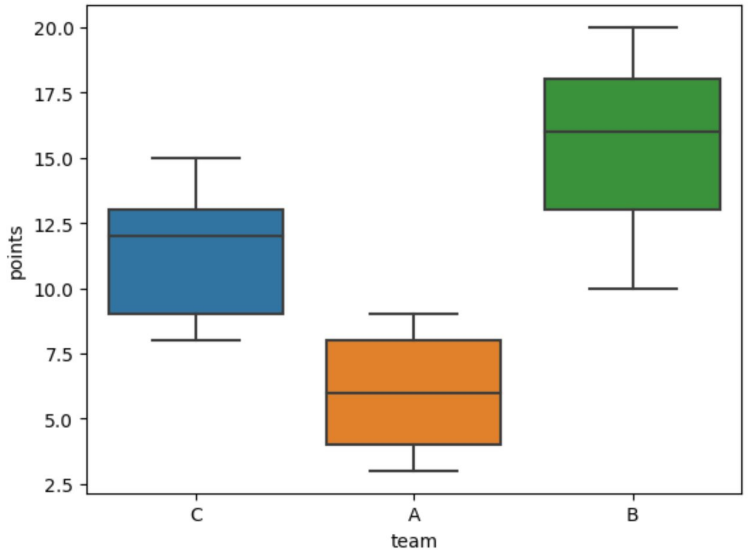
- Tableaux croisés (**contingency tables**): Présentent les fréquences conjointes de deux variables catégorielles

| Âge / Malade | 0 fois | 1 fois | 2 fois | 3 fois | 4 fois |
|---|---|---|---|---|---|
| 20 ≤ âge < 30 ans | 4 individus | 2 individus | 2 individus | 1 individu | 1 individu |
| 30 ≤ âge < 40 ans | 4 | 3 | 3 | 1 | 1 |
| 40 ≤ âge < 50 ans | 7 | 2 | 1 | 0 | 0 |
| 50 ≤ âge < 60 ans | 3 | 2 | 1 | 1 | 1 |
| âge ≥ 60 ans | 0 | 0 | 0 | 1 | 1 |

# Numériques vs. catégorielles

**Boxplots groupés:** Comparent la distribution d'une variable numérique en fonction des catégories d'une variable catégorielle

**Exemple :** On visualise la distribution de points (numérique) dans les sous-groupes (catégorie) d’une classe



# Numériques vs. numériques

**Nuages de points** (scatter plots): Visualisent la relation entre deux variables numériques

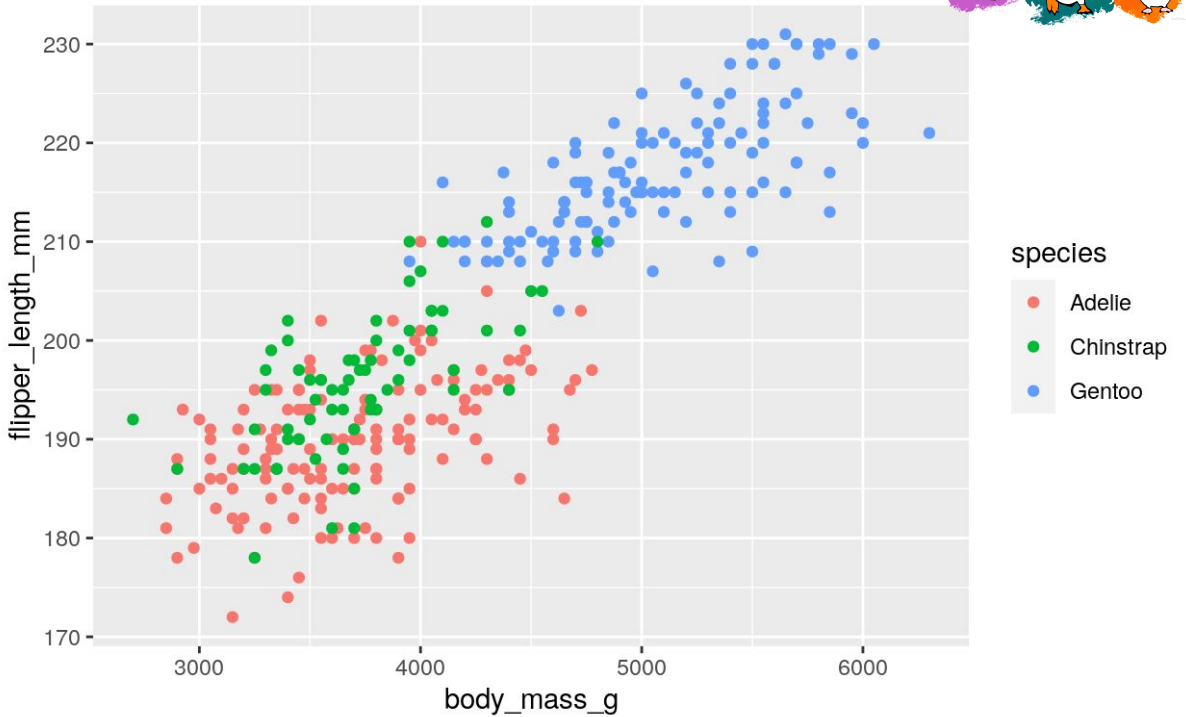
**Corrélation :** Mesure la force et le sens de l'association linéaire entre deux variables numériques

**Matrices de corrélation :** Visualisent les corrélations entre plusieurs variables numériques

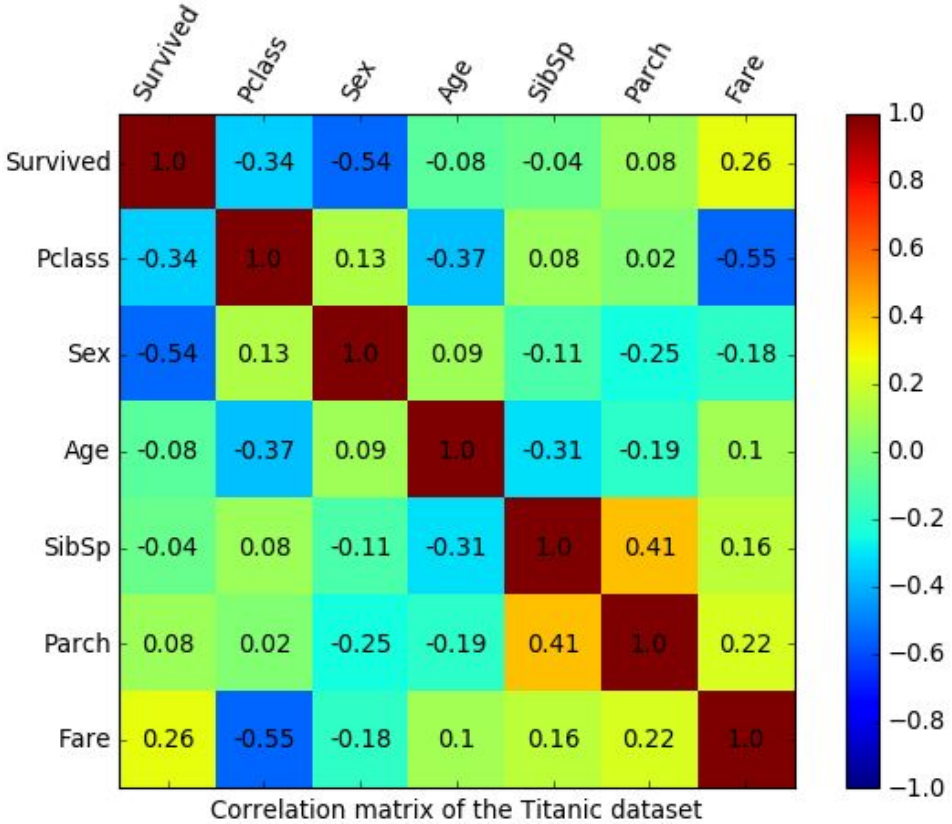
# Nuage de points


Le scatterplot visualise la relation entre 2 variables numériques.

**Exemple :** entre la taille de la nageoire du manchot et son poids (corrélés)



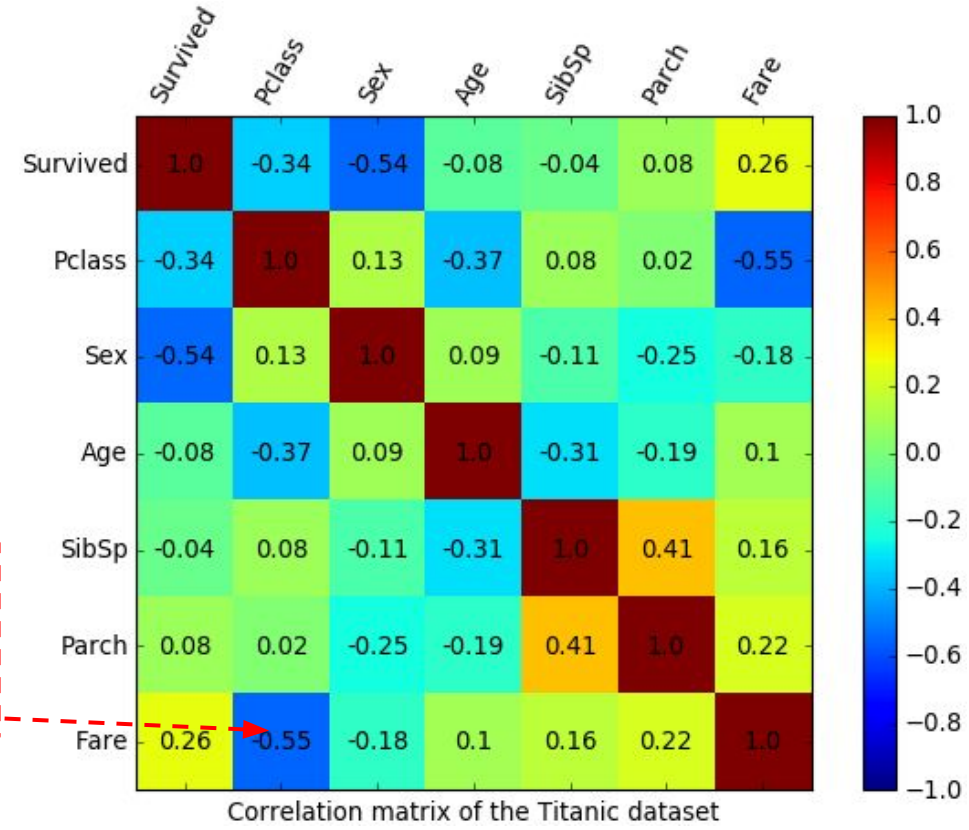
Une **matrice de corrélation** permet de visualiser s'il y a une relation de type linéaire entre deux variables numériques.
Cela peut nous aider à supprimer des **variables redondantes**




Matrice de corrélation (*Titanic dataset*)

Une **matrice de corrélation** permet de visualiser s'il y a une relation de type linéaire entre deux variables numériques.
Cela peut nous aider à supprimer des **variables redondantes**

Deux variables redondantes (Classe dans le navire *Titanic* et Prix du ticket sont corrélés neg.)



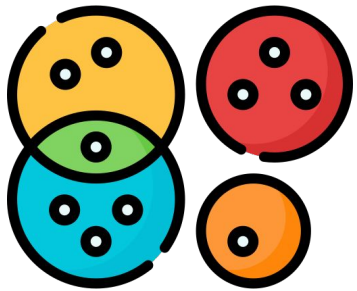

Matrice de corrélation (*Titanic dataset*)

# Analyse multivariée

# Analyse Multivariée

L'**analyse multivariée** étudie simultanément **plusieurs variables** pour comprendre leurs interactions, identifier des groupes d'observations similaires et extraire des informations plus riches.

Elle permet de dépasser les limites de l'analyse bivariée en révélant des structures cachées, des tendances complexes et des relations non linéaires entre les variables.



# Techniques d'analyse multivariée

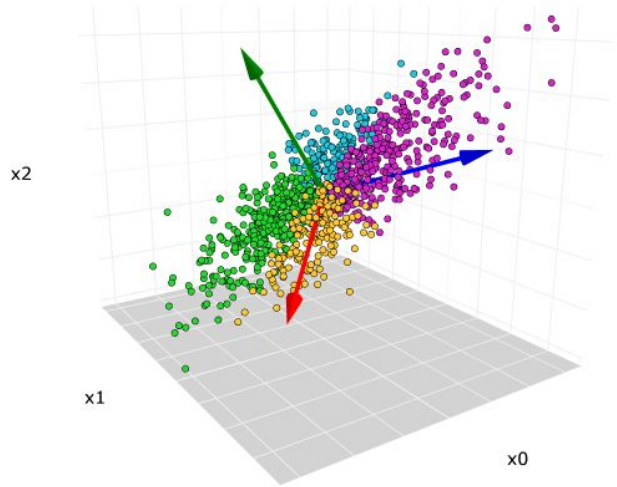
- Analyse en Composantes Principales (**ACP**)
- Analyse Factorielle Discriminante (**AFD**)
- **Clustering** (classification non supervisée)

# Analyse en Composantes Principales (ACP)

Réduit la dimensionnalité des données en identifiant les axes principaux qui expliquent au mieux la variance

Permet de visualiser les données dans un **espace de dimension réduite**

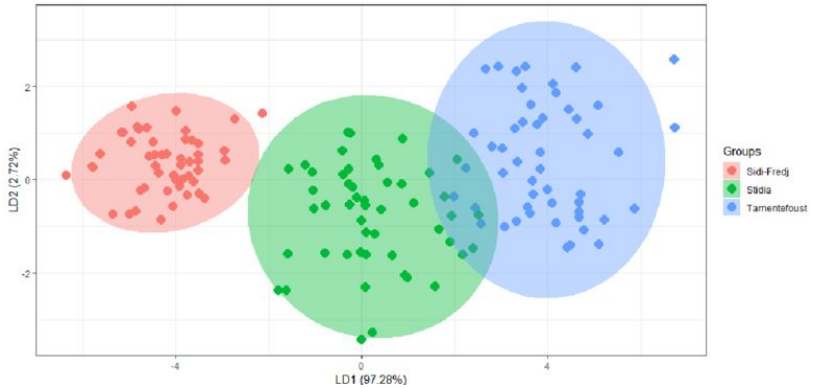
Utile pour identifier les variables les plus importantes et détecter des groupes d'individus similaires



# Analyse Factorielle Discriminante (AFD)

Identifie les **combinaisons linéaires** de variables qui discriminent le mieux entre différents groupes prédéfinis

Utile pour la classification et la prédiction de l'appartenance à un groupe



# Clustering (classification non supervisée)

Regroupe les observations en **clusters homogènes** en fonction de leurs similitudes

Utile pour découvrir des groupes naturels dans les données et segmenter les clients, les produits, etc.