Support Didactique - Séance 1

Fondations Data Science

Formation Machine Learning & MLOps

Objectifs de la Séance

— Comprendre l'importance de l'analyse exploratoire
— Maîtriser l'interprétation des visualisations
— Gérer efficacement les valeurs manquantes
— Identifier et mitiger les biais dans les données
— Créer des features pertinentes et éthiques

1 Philosophie de l'Exploration de Données

Concept Clé

L'analyse exploratoire n'est pas une formalité mais une nécessité :

— **Détecter les problèmes** avant qu'ils ne polluent le modèle
— **Comprendre le domaine** métier à travers les données
— **Éviter les conclusions erronées** causées par des biais cachés
— **Choisir les bonnes techniques** de prétraitement

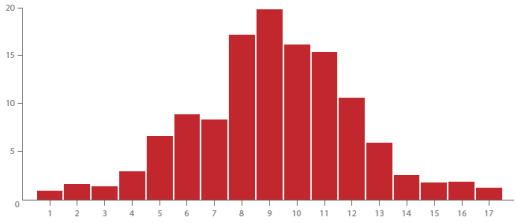
1.1 Les 4 Questions Fondamentales

Question	Technique	Piège à éviter
Quelle est la distribution ?	Histogramme, Boxplot	Confondre asymétrie avec outlier
Y a-t-il des relations ?	Scatterplot, Heatmap	Corrélation $\neq$ causation
Les données sont-elles propres ?	.isnull(), .duplicated()	Supprimer sans analyser le pattern
Existe-t-il des sous-groupes ?	GroupBy, FacetGrid	Conclusions sur petits échantillons

Table 1 – Questions fondamentales pour l'EDA

2 Interprétation des Visualisations

2.1 Histogramme : Lire la Distribution



figureExemple d'histogramme (Âge des passagers)

À analyser :

1. **Forme :** Normale, bimodale, asymétrique ?
2. **Centre :** Moyenne $\neq$ médiane ?
3. **Étalement :** Variance, outliers ?
4. **Gaps :** Problème de collecte ?

Concept Clé

Exemple concret (Âge Titanic) :

- **Asymétrie positive :** Plus de jeunes passagers
- **Pic autour de 20-30 ans :** Voyageurs en âge de travailler
- **Queue longue droite :** Quelques passagers âgés
- **Implication :** Moyenne > Médiane, utiliser médiane pour imputation

2.2 Boxplot : Détecter les Outliers

La règle des 1.5×IQR

$$\text{Outlier inférieur} < Q1 - 1.5 \times IQR \tag{1}$$

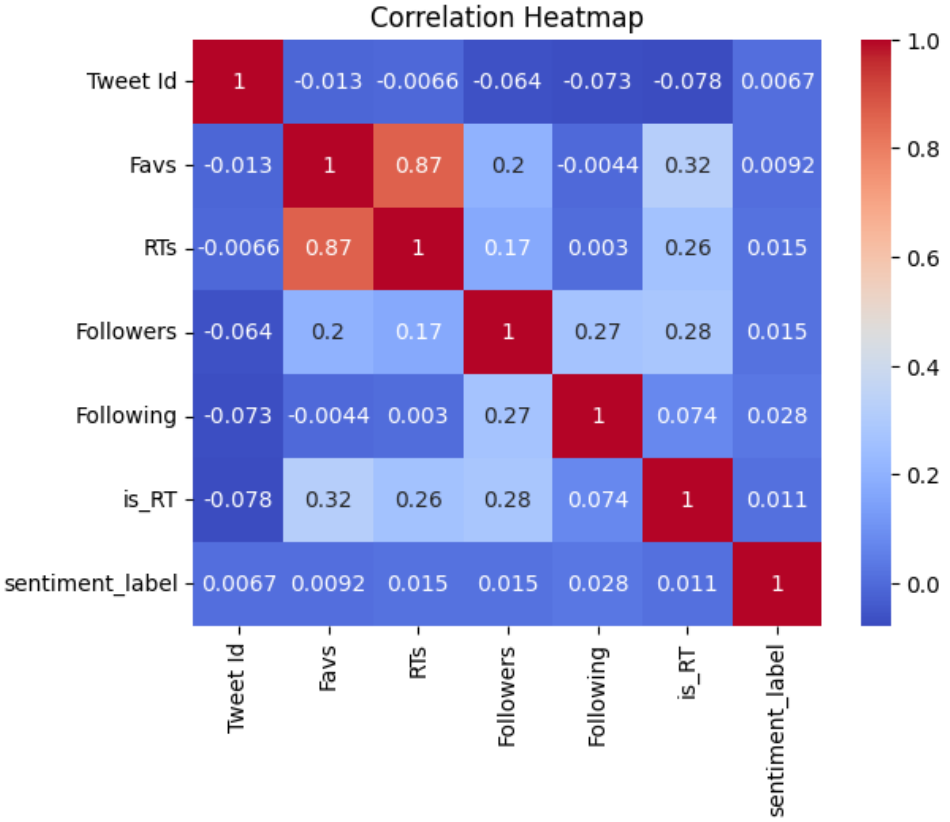
$$\text{Outlier supérieur} > Q3 + 1.5 \times IQR \tag{2}$$

$$\text{avec } IQR = Q3 - Q1 \tag{3}$$

Ne pas supprimer automatiquement !	**Questions à se poser**
Point de données légitime	Est-ce une erreur de mesure ?
Représente une population rare	Vient-il du même processus génératif ?
Information importante	Impacte-t-il la métrique business ?

TABLE 2 – Décision concernant les outliers

2.3 Heatmap de Corrélation



figureExemple de heatmap de corrélations

Valeur	Force	Interprétation
$\|r\| = 1$	Parfaite	Relation linéaire exacte
$\|r\| > 0.7$	Forte	Prédiction possible
$0.3 < \|r\| < 0.7$	Modérée	Relation détectable
$\|r\| < 0.3$	Faible	Peu ou pas de relation

TABLE 3 – Interprétation des coefficients de corrélation

Attention

Avertissements critiques :

1. **Corrélation $\neq$ Causation :** Le chant du coq ne fait pas lever le soleil
2. **Sensibilité aux outliers :** Un outlier peut créer une corrélation illusoire
3. **Relations non-linéaires :** $r = 0$ n'implique pas indépendance
4. **Biais d'agrégation :** Corrélation écologique corrélation individuelle

3 Gestion des Valeurs Manquantes

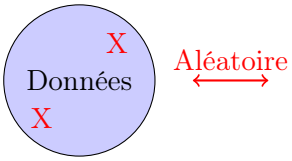
3.1 Classification des Mécanismes

MCAR : Missing Completely At Random

Définition : Absence indépendante des valeurs observées ET non observées

— Exemple : Questionnaire perdu aléatoirement

— Solution : Suppression ou imputation simple

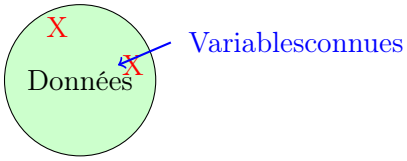


MAR : Missing At Random

Définition : Absence dépendante des valeurs observées

— Exemple : Âge manquant pour jeunes participants (sexe connu)

— Solution : Imputation conditionnelle

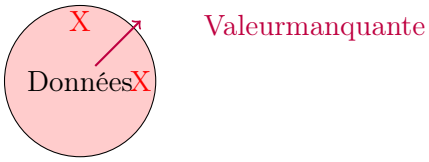


MNAR : Missing Not At Random

Définition : Absence dépendante des valeurs non observées

— Exemple : Salaire manquant pour hauts revenus

— Solution : Modélisation explicite ou flag



3.2 Méthodes d'Imputation Comparées

Méthode	Avantages	Inconvénients	Utilisation
Suppression	Simple, pas de biais ajouté	Perte d'information	MCAR, <5%
Moyenne/Médiane	Rapide, stable	Réduit la variance	Variables numériques
KNN	Utilise similarité entre observations	Comput. coûteux, choix de k	MAR, dataset riche
MICE	Gère interdépendances complexes	Complexe, convergence ?	Données structurées
Flag + Imputation	Préserve l'information du manque	Augmente dimensionnalité	MNAR suspecté

TABLE 4 – Choix de la méthode d'imputation

3.3 Détection de Pattern en Python

```python
import pandas as pd
import numpy as np
from scipy.stats import chi2_contingency
import seaborn as sns

# 1. Pattern simple
missing_age = df['Age'].isnull()
print("Survie quand Age manquant :")
print(df.loc[missing_age, 'Survived'].mean())
print("Survie quand Age pr sent :")
print(df.loc[~missing_age, 'Survived'].mean())

```

```python
# 2. Test statistique (test du chi2)
contingency = pd.crosstab(missing_age, df['Survived'])
chi2, p, dof, expected = chi2_contingency(contingency)
print(f"p-value : {p:.4f}")  # p < 0.05     pattern significatif

# 3. Visualisation du pattern
df['Age_missing'] = missing_age
sns.countplot(data=df, x='Pclass', hue='Age_missing')
plt.title('R partition des valeurs manquantes par Classe')
plt.show()
```

Listing 1 – Analyse des patterns de valeurs manquantes

Concept Clé

Interprétation des résultats :

- **p-value $<$ 0.05 :** Pattern détecté → risque de biais MNAR
- **Différence de survie :** Si significative, le manque n'est pas aléatoire
- **Action :** Créer une variable "Age_was_missing" + imputation conservatrice

4 Types de Biais et Mitigation

4.1 Taxonomie des Biais

Type de biais	Description	Exemple Titanic	Impact
Biais de mesure	Erreur systématique collecte	"Cabin" non-enregistré 3e classe	Variables incomplètes
Biais de sélection	Échantillon non représentatif	Seulement passagers survivants	Généralisation impossible
Biais d'étiquetage	Labels incorrects/incomplets	Survie influencée par normes sociales	Modèle apprend stéréotypes
Biais de confirmation	Chercher ce qu'on veut voir	Focus sur "femmes et enfants d'abord"	Ignorer contre-exemples
Biais algorithmique	Algorithme amplifie les biais	Sur-optimisation sur classe majoritaire	Discrimination

TABLE 5 – Types de biais courants en data science

4.2 Stratégies de Mitigation

Avant la modélisation

1. **Analyse des marges de distribution :**

$$\text{Écart} = \left| \frac{\text{Proportion échantillon} - \text{Proportion population}}{\text{Proportion population}} \right| \tag{4}$$

2. **Pondération des observations :**

$$w_i = \frac{N_{\text{pop}}/K_{\text{pop}}}{N_{\text{sample}}/K_{\text{sample}}} \tag{5}$$

3. **Stratification de l'échantillon :** Garantir représentation minimale

Pendant la modélisation

- **Regularisation :** Pénaliser les coefficients extrêmes
- **Métriques équitables :** F1-score par classe, AUC par sous-groupe
- **Contraintes de fairness :**

$$\text{s.t. } |P(\hat{Y} = 1|A = a) - P(\hat{Y} = 1|A = b)| < \epsilon \tag{6}$$

Après la modélisation

- **Audit des prédictions :** Analyse des erreurs par sous-groupe
- **Interprétabilités :** SHAP values pour comprendre les décisions
- **Monitorage continu :** Détection de dérive dans le temps

4.3 Détection de Biais de Sélection

```python
def detect_selection_bias(df, reference_population):
    """
    Detecte les ecarts de representation entre l'echantillon
    et la population de reference
    """
    bias_report = {}

    for col in ['Pclass', 'Sex', 'AgeGroup']:
        # Calcul des proportions
        sample_prop = df[col].value_counts(normalize=True)
        ref_prop = reference_population[col].value_counts(normalize=True)

        # Calcul  cart  relatif
        common_categories = set(sample_prop.index) & set(ref_prop.index)
        max_bias = 0
        for cat in common_categories:
            bias = abs(sample_prop[cat] - ref_prop[cat]) / ref_prop[cat]
            max_bias = max(max_bias, bias)

        bias_report[col] = {
            'max_relative_bias': max_bias,
            'warning': max_bias > 0.3  # Seuil de 30%
        }

    return bias_report

# Utilisation
reference_data = pd.read_csv('titanic_population_reference.csv')
bias_results = detect_selection_bias(df, reference_data)
print(bias_results)
```

Listing 2 – Détection automatique de biais de représentation

5 Feature Engineering Critique

5.1 Principes de Création Éthique

Concept Clé

Les 5 commandements du feature engineering :

1. **Comprendre avant de créer :** Connaître la signification métier
2. **Documenter la provenance :** Traçabilité des transformations
3. **Éviter la fuite temporelle :** Features calculées sur données futures
4. **Tester l'impact :** Validation croisée pour chaque nouvelle feature
5. **Auditer les biais :** Vérifier l'impact sur les sous-groupes

5.2 Exemples de Features Robustes

```python
# 1. Variables composites (reduisent la colinearite)
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1
df['IsAlone'] = (df['FamilySize'] == 1).astype(int)

# 2. Binning intelligent (bas  sur la distribution)
df['AgeGroup'] = pd.cut(df['Age'],
                        bins=[0, 12, 18, 30, 50, 100],
                        labels=['Child', 'Teen', 'YoungAdult',
                                'Adult', 'Senior'])

# 3. Ratio features (plus robustes que les differences)
df['FarePerPerson'] = df['Fare'] / df['FamilySize']

# 4. Interactions non-biaisees
df['FemaleInFirstClass'] = ((df['Sex'] == 'female') &
                            (df['Pclass'] == 1)).astype(int)

# 5. Encodage target-aware (avec pr caution !)
from category_encoders import TargetEncoder

# UNIQUEMENT sur le training set !
encoder = TargetEncoder()
X_train_encoded = encoder.fit_transform(X_train[['Embarked']], y_train)
```

Listing 3 – Création de features informatives et equitables

6 Checklist de Fin de Session

Validation Technique	Validation Éthique
☐ Valeurs manquantes analysées et traitées	☐ Biais de mesure identifiés
☐ Outliers détectés et justifiés	☐ Biais de sélection quantifiés
☐ Distributions visualisées et comprises	☐ Features créées sans reproduire stéréotypes
☐ Corrélations interprétées	☐ Impact sur sous-groupes évalué
☐ Code documenté et reproductible	☐ Limitations clairement énoncées

Table 6 – Checklist de validation de l'analyse exploratoire

La phrase à retenir :

"Un bon data scientist ne se contente pas de trouver des patterns, il comprend lesquels sont réels, lesquels sont des artefacts, et lesquels sont éthiquement problématiques."

Ressources Complémentaires

- **Visualisation :** "The Visual Display of Quantitative Information" - Tufte
- **Biais statistiques :** "Statistical Fallacies" - Altman et Bland
- **Éthique ML :** "Weapons of Math Destruction" - O'Neil
- **Tools :** Aequitas (audit de fairness), SHAP (interprétabilité)
- **Datasets de test :** UCI ML Repository, Kaggle (avec précaution)

Exercice Pratique

Exercice d'application immédiate :
Créez une fonction Python qui :

1. Prend un DataFrame en entrée
2. Détecte automatiquement le type de mécanisme de valeurs manquantes (MCAR/-MAR/MNAR)
3. Propose une stratégie de traitement adaptée
4. Génère un rapport d'analyse avec visualisations

Bonus : Testez cette fonction sur le dataset Titanic et comparez avec votre approche initiale.

Document préparé pour la Formation Machine Learning & MLOps - Séance 1
Révision : 1.0 | Date : Déc-2025 | Principe pédagogique : "Comprendre pourquoi avant d'apprendre comment"