📌 Méthodologie d'une analyse de données (RNCP Niveau 7)

🏁 1. Définition du problème

Avant de commencer une analyse, il est essentiel de **bien comprendre la question à résoudre**.

- **Objectif** : Que cherche-t-on à analyser ? (ex : Prédire les ventes d'un produit, comprendre le comportement des clients...)
- **Contexte métier** : Qui va utiliser ces résultats ? Pourquoi cette analyse est-elle importante ?
- **Types de données nécessaires** : Besoin de données historiques, en temps réel ? Structurées ou non ?

🔹 **Exemple** : Une entreprise souhaite analyser pourquoi ses ventes baissent.

📊 2. Collecte des données

Une fois la problématique définie, il faut **récupérer les données** nécessaires.

- **Sources de données** :
    - Bases de données SQL / NoSQL
    - Fichiers CSV, Excel, JSON
    - API publiques ou internes
    - Web Scraping (si besoin d'extraction depuis un site web)
- **Vérification des sources** :
    - Données complètes ?
    - Données à jour ?
    - Sources fiables ?

🔹 **Exemple** : On récupère les ventes des 5 dernières années depuis une base de données SQL.

📌 **Commande SQL pour récupérer les données :**

```sql
SELECT * FROM ventes WHERE date >= '2019-01-01';
```

📌 **Commande Python avec Pandas :**

```python
import pandas as pd

df = pd.read_csv('ventes.csv')  # Charger un fichier CSV
df.head()  # Voir les premières lignes
```

🛠️ 3. Préparation et nettoyage des données

Les données brutes sont rarement exploitables immédiatement. Il faut les nettoyer et les préparer.

- **Vérification des valeurs manquantes** → Suppression ou imputation des valeurs nulles
- **Suppression des doublons** → Éviter les erreurs d'analyse
- **Standardisation des formats** → Dates, nombres, textes homogènes
- **Gestion des valeurs aberrantes** → Exclure ou ajuster les valeurs incohérentes

🔹 **Exemple** : Supprimer les transactions dupliquées et remplacer les valeurs manquantes par la moyenne.

📌 **Commande Python avec Pandas :**

```python
# Supprimer les doublons
df.drop_duplicates(inplace=True)

# Remplacer les valeurs manquantes par la moyenne
df.fillna(df.mean(), inplace=True)
```

📈 4. Exploration et visualisation des données

Avant toute modélisation, il faut **comprendre les tendances des données**.

- **Statistiques descriptives** : Moyenne, médiane, écart-type...
- **Visualisations** : Histogrammes, scatter plots, heatmaps...
- **Analyse des corrélations** : Repérer les relations entre les variables

🔹 **Exemple** : Analyser si une hausse des prix impacte les ventes avec un graphique.

📌 **Commande Python avec Matplotlib et Seaborn :**

```
import matplotlib.pyplot as plt
import seaborn as sns

# Visualisation de la distribution des ventes
sns.histplot(df['ventes'], bins=30, kde=True)
plt.title('Distribution des ventes')
plt.show()
```

🔬 5. Modélisation et analyse avancée

Si l'objectif est prédictif, il faut entraîner un modèle.

- **Modèles statistiques** : Régressions linéaires/logistiques
- **Modèles de Machine Learning** : Forêts aléatoires, SVM, réseaux de neurones
- **Tests et validation** : Comparaison des performances avec des métriques (MSE, Accuracy...)

🔹 **Exemple** : Prédire les ventes en fonction des saisons avec une régression linéaire.

📌 **Commande Python avec Scikit-learn :**

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Séparation des données
X = df[['prix', 'publicite']]
y = df['ventes']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

# Entraînement du modèle
model = LinearRegression()
model.fit(X_train, y_train)

# Prédictions
y_pred = model.predict(X_test)

# Évaluation du modèle
mse = mean_squared_error(y_test, y_pred)
print(f'Erreur quadratique moyenne : {mse}')
```

📌 6. Interprétation et recommandations

L'analyse ne sert à rien si on ne sait pas l'expliquer !

- **Synthèse des résultats** : Quels sont les insights clés ?
- **Visualisation des tendances** : Graphiques et dashboards
- **Recommandations basées sur les données** : Actions concrètes à prendre

🔹 **Exemple** : "Nos ventes chutent quand les prix augmentent. Nous recommandons une promotion saisonnière."

📢 7. Présentation des résultats

Un Data Analyst doit savoir **communiquer** ses analyses à un public non technique.

- **Storytelling avec les données** → Raconter une histoire avec les insights
- **Rapport écrit** → Synthèse claire des résultats
- **Présentation visuelle** → Slides, dashboards interactifs (Power BI, Tableau...)

🔹 **Exemple** : Présentation PowerPoint aux managers avec des graphiques clairs.

📌 **Commande pour créer un Dashboard avec Power BI ou Tableau :**

- Importer les données CSV ou SQL.
- Créer des graphiques interactifs (barres, courbes, heatmaps).
- Ajouter des filtres et segments pour affiner l'analyse.

🚀 Conclusion : Suivi et amélioration continue

L’analyse de données est un processus **itératif**.

- **Suivre l’impact des recommandations** → Est-ce que les changements ont amélioré les performances ?
- **Améliorer les modèles** → Ajuster les paramètres, tester de nouvelles approches
- **Automatiser les analyses** → Déploiement sous forme de dashboards, scripts automatisés

🔹 **Exemple** : Mettre à jour l’analyse chaque mois et affiner les modèles en fonction des nouvelles données.

💡 **En suivant cette méthodologie, tu structureras bien tes analyses et éviteras les erreurs lors de ton diplôme !** 🚀