**REPUBLIQUE DEMOCRATIQUE DU CONGO**

**ENSEIGNEMENT SUPERIEUR ET UNIVERSITAIRE**

**INSTITUT SUPERIEUR DE STATISTIQUE**

**ECOLE DOCTORALE DE STATISTIQUE ET INFORMATIQUE (EDSI)**

**B.P.2471**

**LUBUMBASHI**

J0105250



**TRAVAIL PRATIQUE DU COURS DE DATA MINING**

Titulaire du Cours : MBAKI LUZAYISU Efrem

Présenté par :

- PINTO KATENDE Jonathan

- MUSONDA Balthazar

- KHANG MATE ZULBAL

- MUKWIYO MUKALO Patrick

- MPUNDU MWANA NGOY

**2024**

**QUESTION**

**Objectif**

1. **Préparation des Données :**
   * Importer les données et vérifier s'il y a des valeurs manquantes.
   * Nettoyer les données en traitant les valeurs manquantes et en codant les variables catégorielles si nécessaire.
2. **Analyse Exploratoire :**
   * Calculer les statistiques descriptives pour chaque variable et visualiser la répartition des variables
     + Statistique univariée
     + Statistique Bivariée
     + Etude des liaisons et corrélations
3. **Modélisation :**
   * Construire un modèle pour prédire les différentes Catégories en fonction des autres variables
   * Évaluer la performance du modèle.
4. **Visualisation et Interprétation :**
   * Présenter les résultats de manière visuelle (graphes, diagrammes).
   * Interpréter les résultats et proposer des recommandations

**ANALYSE**

Dans le cadre de cette analyse, nous utilisons Python 3.13 comme langage de programmation.

Voici les librairies qui serons utilisés :

**Préparation des données**

#### Installation des packages

%pip install pandas numpy # Manipulation excel  
%pip install --upgrade pip # Gestionnaire des paquets  
%pip install polars # Meilleur alternatif de Pandas  
%pip install matplotlib plotly # Pour la visualisation  
%pip install fastexcel # Dépendance de Polars pour le fichie .xlsx  
%pip install pyarrow # Dépendance de Polars pour le fichie .xlsx  
%pip install openpyxl # Dépendance de Polars pour le fichie .xlsx  
%pip install seaborn # Pour la visualisation  
%pip install scikit-learn # Pour la visualisation  
%pip install statsmodels # Pour la visualisation

%pip install plotnine # Pour la visualisation

%pip install tabulate colorama # Pour la colaration

#### Chargement des packages

import pandas as pd # type: ignore

import numpy as np

import polars as pl

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report

from sklearn.preprocessing import LabelEncoder

import statsmodels.api as sm

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

from plotnine import \* # type: ignore  # noqa: F403

import scipy.stats as ss

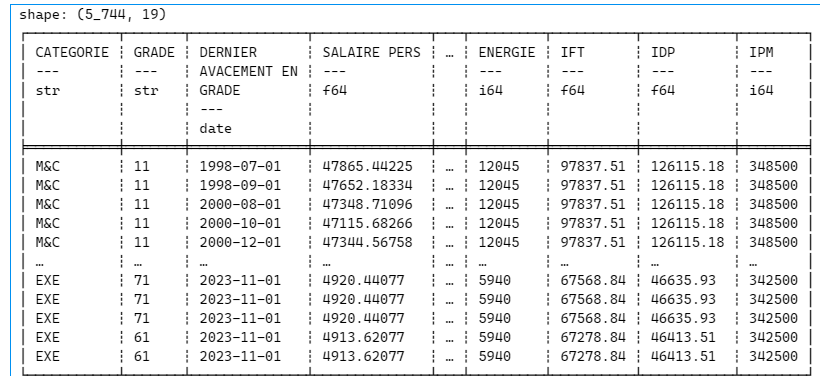
from tabulate import tabulate

from colorama import Fore, Style, Back

#### Importation des données

data\_pl = pl.read\_excel("data.xlsx") # Pour la manipulation avec Polars  
data\_pd = pd.read\_excel("data.xlsx") # Pour la manipulation avec Pandas  
ctx = pl.SQLContext(data=data\_pl, eager=True) # Ecriture en SQL

#### Illustration des données

print(data\_pl)

#### Explication des variables

* **CATEGORIE** : Code ou description de la catégorie professionnelle du personnel
* **GRADE** : Niveau de grade de l’employé au sein de la catégorie
* **DERNIER AVACEMENT EN GRADE** : Date du dernier avancement en grade de l’employé.
* **SALAIRE PERS** : Salaire personnel ou de base de l’employé,
* **CITE** : Code de la cité ou de la localité où réside l’employé
* **SEXE** : Sexe de l’employé
* **ETATCIVIL** : Code de l’état civil de l’employé.
* **INDEPOUSE** : Indicateur pour savoir si l’employé a un conjoint à charge. Par exemple, “1” pour oui et “0” pour non.
* **ENFANT** : Nombre d’enfants à charge de l’employé
* **COTATION** : Code de cotation ou score lié au poste ou à la performance de l’employé.
* **DATENAISSA** : Date de naissance de l’employé.
* **DATEENGAGE** : Date d’engagement de l’employé dans l’entreprise (date d’embauche).
* **TRP** : Indemnité de transport, ou allocation liée aux frais de déplacement.
* **LGT** : Allocation logement de l’employé.
* **SSANTE** : Cotisation ou indemnité pour la sécurité sociale ou assurance santé.
* **ENERGIE** : Allocation ou indemnité énergétique, qui pourrait couvrir des frais d’électricité, de chauffage, etc.
* **IFT** : Allocation ou indemnité pour frais de téléphone.
* **IDP** : Indemnité pour dépendants, ou allocation pour les personnes à charge.
* **IPM** : Indemnité pour prestations médicales, couvrant potentiellement les frais de santé ou d’hospitalisation.

#### Information sur les variables

print(data\_pd.info())

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 5744 entries, 0 to 5743  
Data columns (total 19 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 CATEGORIE 5744 non-null object   
 1 GRADE 5744 non-null int64   
 2 DERNIER AVACEMENT EN GRADE 5742 non-null datetime64[ns]  
 3 SALAIRE PERS 5744 non-null float64   
 4 CITE 5744 non-null object   
 5 SEXE 5744 non-null object   
 6 ETATCIVIL 5678 non-null object   
 7 INDEPOUSE 5534 non-null float64   
 8 ENFANT 5744 non-null int64   
 9 COTATION 2536 non-null float64   
 10 DATENAISSA 5744 non-null datetime64[ns]  
 11 DATEENGAGE 5744 non-null datetime64[ns]  
 12 TRP 5744 non-null float64   
 13 LGT 5744 non-null float64   
 14 SSANTE 5744 non-null float64   
 15 ENERGIE 5744 non-null int64   
 16 IFT 5744 non-null float64   
 17 IDP 5744 non-null float64   
 18 IPM 5744 non-null int64   
dtypes: datetime64[ns](3), float64(8), int64(4), object(4)  
memory usage: 852.8+ KB

Nous constatons que l’analyse est faite sur 5744 individus.

Sur toutes les variables, nous remarquons que la variable **COTATION** a près de 50% des valeurs manquantes. En plus, cette variable cotation n’intéresse que ceux de la catégorie M&C. D’où, elle sera supprimée lors de la modélisation.

#### Création des variables artificielles

En se fiant sur les données que nous disposons, nous constatons que certaines variables artificielles peuvent être créées :

* **L’âge à l’engagement** : la différence entre la date d’engagement et la date de Naissance.
* Après que l’âge à l’engagement soit créé, une autre variable peut être créée sur la base de celle-ci : **tranche d’âge.**

Pour la création des classes, il existe des critères et des méthodes existent du point de vu statistique, entres autres :

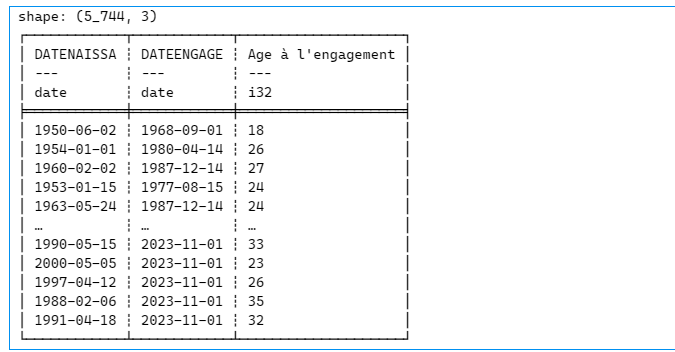
* + [Méthode de LIORZOU](https://hal.science/hal-04064748v1/file/Statistique%20descritive%20appliqu%C3%A9e.pdf)
  + [Méthode de Stem and Leaf](https://mmerevise.co.uk/gcse-maths-revision/stem-and-leaf-diagrams-revision/)

Dans le cadre de cette étude, nous allons nous limiter par une méthode manuelle d’amplitude 5, de la valeur minimale de cette variable jusqu’à sa valeur maximale.

*Création de variable âge*

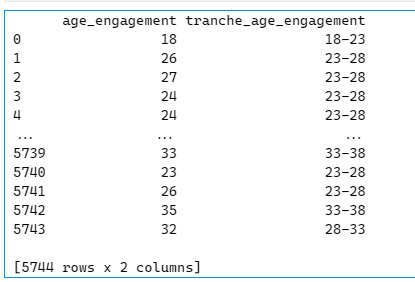
result = data\_pl.select(  
 pl.col('DATENAISSA'),  
 pl.col('DATEENGAGE'),  
 ((pl.col("DATEENGAGE") - pl.col('DATENAISSA')).cast(pl.Duration) / pl.duration(days=365.25)).cast(pl.Int32).alias("Age à l'engagement")  
)  
  
print(result)

Voici comment se présente cette nouvelle variable :



*Création de variable Tranche d’âge*

data\_pd = data\_pl\_new.to\_pandas()  
  
bins = [13, 18, 23, 28, 33, 38, 43, 48, 53, 56]  
labels = ['13-18', '18-23', '23-28', '28-33', '33-38', '38-43', '43-48', '48-53', 'Plus de 53']  
data\_pd['tranche\_age\_engagement'] = pd.cut(data\_pd['age\_engagement'], bins=bins, labels=labels, right=False)  
  
print(data\_pd[['age\_engagement', 'tranche\_age\_engagement']])



**Analyse exploratoire**

Dans le cadre de cette analyse, nous considérons la variable CATEGORIE comme variable cible.

* **Analyse univariée**

Variables Quantitatives

print(data\_pl\_new.select(  
 pl.col('age\_engagement'),  
 pl.col('SALAIRE PERS'),  
 pl.col('ENFANT')  
).describe())

shape: (9, 4)  
┌────────────┬────────────────┬──────────────┬──────────┐  
│ statistic ┆ age\_engagement ┆ SALAIRE PERS ┆ ENFANT │  
│ --- ┆ --- ┆ --- ┆ --- │  
│ str ┆ f64 ┆ f64 ┆ f64 │  
╞════════════╪════════════════╪══════════════╪══════════╡  
│ count ┆ 5744.0 ┆ 5744.0 ┆ 5744.0 │  
│ null\_count ┆ 0.0 ┆ 0.0 ┆ 0.0 │  
│ mean ┆ 25.584436 ┆ 25926.253361 ┆ 2.184018 │  
│ std ┆ 5.77386 ┆ 23616.887196 ┆ 2.524423 │  
│ min ┆ 13.0 ┆ 4705.46265 ┆ 0.0 │  
│ 25% ┆ 21.0 ┆ 5180.02077 ┆ 0.0 │  
│ 50% ┆ 24.0 ┆ 5352.96077 ┆ 1.0 │  
│ 75% ┆ 29.0 ┆ 49967.96363 ┆ 4.0 │  
│ max ┆ 55.0 ┆ 74226.169 ┆ 16.0 │  
└────────────┴────────────────┴──────────────┴──────────┘

Les résultats de l’analyse montrent que :

* Le salaire moyen de cette population est de 25926
* Tandis que près de 50% des individus de cette population ont un salaire de moins de 5180.
* Se basant sur le nombre d’enfants, en moyenne chaque agent a 2 enfants.
* Près de 50% des agents ne dépasse pas un agent.
* Mais dans cette population, il y a au moins un agent qui a 16 enfants.
* Se référant sur l’âge à l’engagement, on constate que l’âge moyen à l’engagement est de 25 ans.
* Près de 75% d’agents avait moins de 29 ans à l’engagement.

Nous constatons également des incohérences sur plusieures résultats, mais il y a une raison à cela qui pourrait être déceler lors de l’analyse bivariée.

Mais également il y a des résultats qui violent les recommandations de l’Etat Congolais :

Nous voyons que l’âge à l’engagement minimal est de 13 ans, or la loi N°16.010 du 15 Juillet 2016 portant code de la famille stipule que « Tout enfant né en République Démocratique du Congo peut être engagé ou maintenue en service pour l’exécution des travaux légers et salubres, que celui qui 16 à moins de 18 ans ». D’où l’âge minimal pour le travail est de 16 ans.

data\_pl\_new.filter(pl.col('age\_engagement') < 16).height

5

On sait voir que l’entreprise a engagé 5 personnes qui ont moins de 16, ce qui est illégal.

**Variables Qualitatives**

1. Tranche d’âge

# Conversion du DataFrame Polars à Pandas

tranche\_counts\_pd = tranche\_counts\_pl.to\_pandas()

plot = (

    ggplot(tranche\_counts\_pd, aes(x='tranche\_age\_engagement', y='effectif', fill='tranche\_age\_engagement'))

    + geom\_bar(stat='identity', show\_legend=False)  # On désactive la légende ici pour simplifier

    + scale\_fill\_manual(values=[

        '#1f77b4',  # bleu

        '#ff7f0e',  # orange

        '#2ca02c',  # vert

        '#d62728',  # rouge

        '#9467bd',  # violet

        '#8c564b',  # brun

        '#e377c2',  # rose

        '#7f7f7f',  # gris

        '#bcbd22'   # jaune-vert

    ])

    + labs(

        title='Distribution des employés par tranche d\'âge',

        x='Tranche d\'âge',

        y='Nombre de personnes'

    )

    + theme\_minimal()

    + theme(

        axis\_text\_x=element\_text(rotation=45, hjust=1),

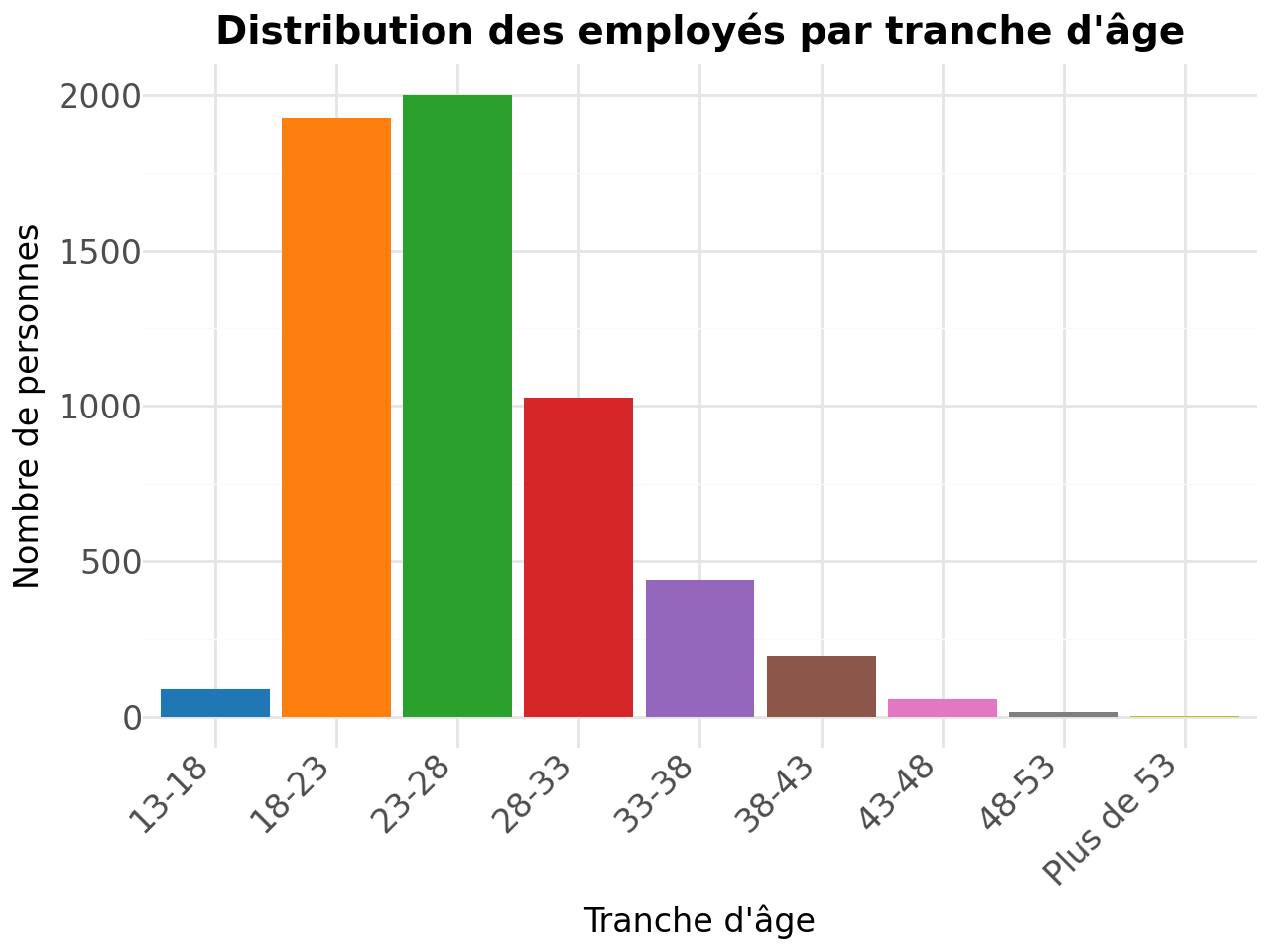
        text=element\_text(size=12),

        plot\_title=element\_text(weight='bold', size=14)

    )

)

plot



Les résultats montrent que :

* La tranche d’âge majoritaire pour les engagements est de 18 à 38 ans
* Mais avec des engagements aussi entre 13 et 18 ans

2. Catégorie

categorie\_counts = data\_pd[['CATEGORIE']].value\_counts().reset\_index()

categorie\_counts.columns = ['CATEGORIE', 'count']

total\_count = categorie\_counts['count'].sum()

categorie\_counts['percentage'] = (categorie\_counts['count'] / total\_count) \* 100

(ggplot(categorie\_counts, aes(x='CATEGORIE', y='percentage', fill='CATEGORIE'))

+ geom\_bar(stat='identity', show\_legend=False)

+ labs(

    title='Distribution par CATEGORIE',

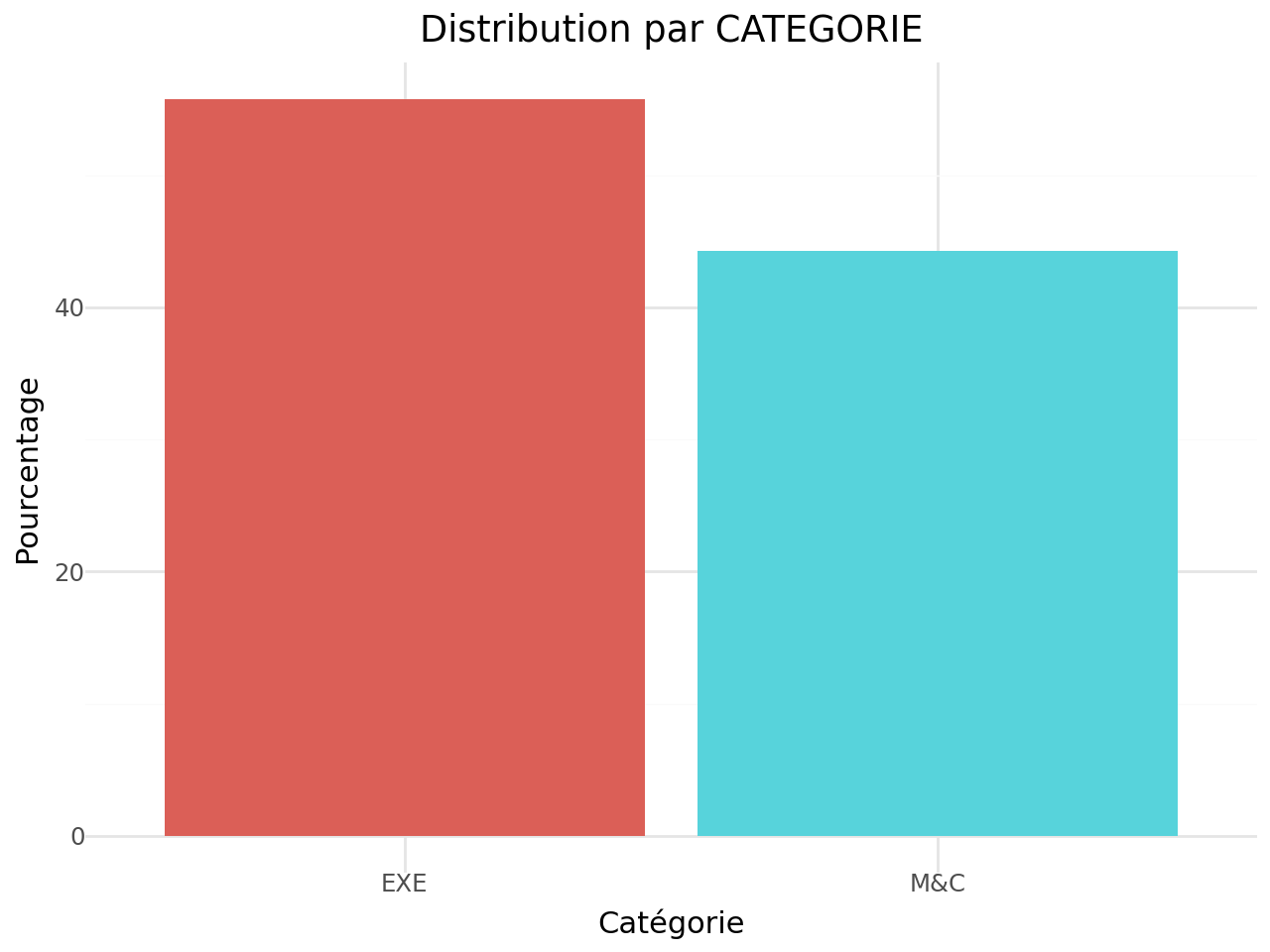
    x='Catégorie',

    y='Pourcentage'

)

+ theme\_minimal()

)



Ce graphique montre que pour cette société, la majorité des employés sont dans la categorie EXE, à hauteur de plus de 55% des agents se trouvant dans cette catégorie.

3. Indepouse (Indicateur pour savoir si l’employé a un conjoint à charge)

indepouse\_counts = data\_pd[['INDEPOUSE']].value\_counts().reset\_index()

indepouse\_counts.columns = ['INDEPOUSE', 'count']

# Calcul des pourcentages

total\_count = indepouse\_counts['count'].sum()

indepouse\_counts['percentage'] = (indepouse\_counts['count'] / total\_count) \* 100

(

    ggplot(indepouse\_counts, aes(x='INDEPOUSE', y='percentage', fill='factor(INDEPOUSE)'))

    + geom\_bar(stat='identity')

    + scale\_fill\_manual(values=['#66c2a5', '#fc8d62'])  # Palette de couleurs attrayantes

    + labs(

        title='Distribution des valeurs de INDEPOUSE en pourcentage',

        x='INDEPOUSE',

        y='Pourcentage'

    )

    + theme\_minimal()

    + theme(

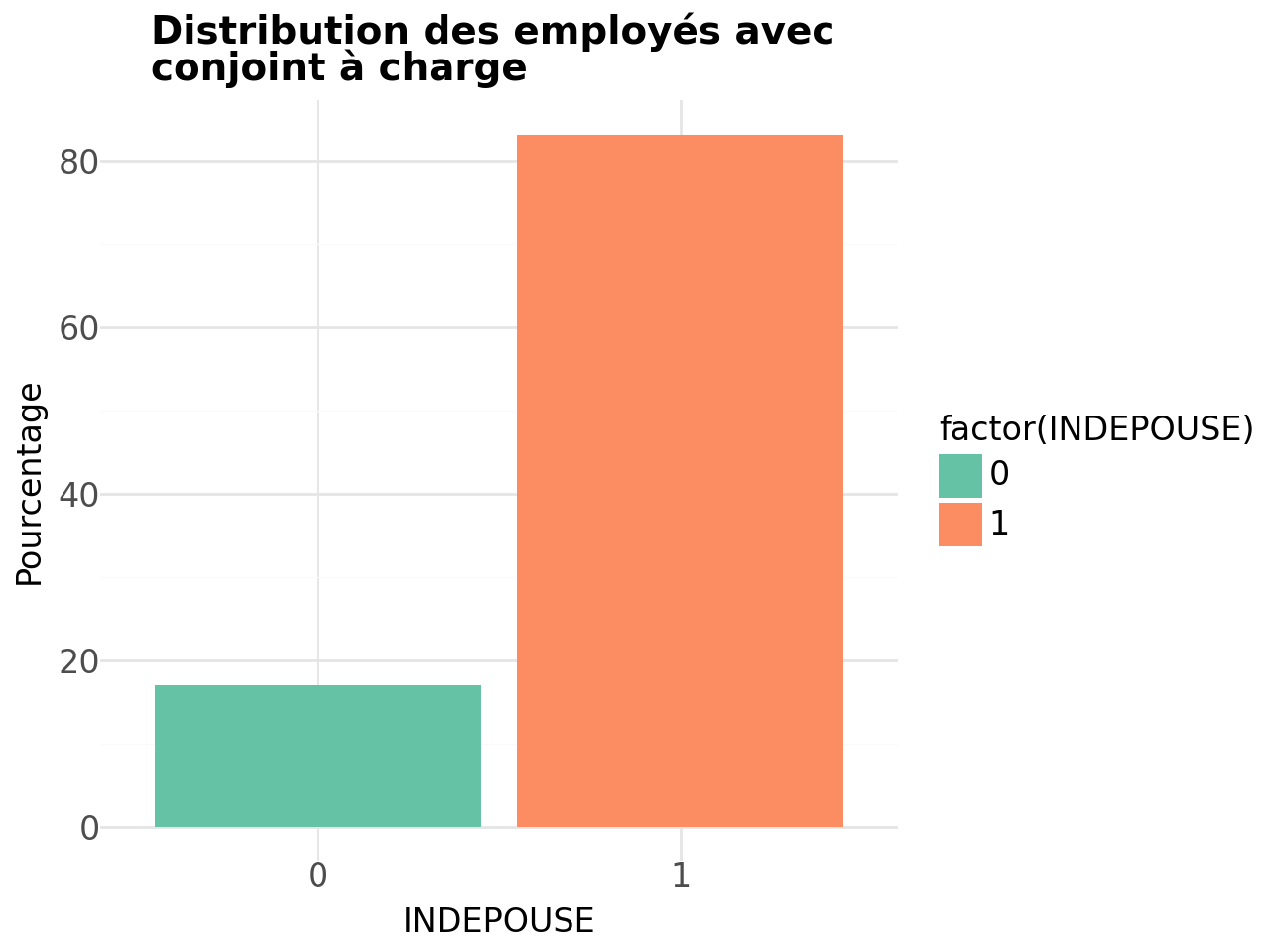
        axis\_text\_x=element\_text(rotation=0, hjust=0.5),

        text=element\_text(size=12),

        plot\_title=element\_text(weight='bold', size=14)

    )

)



Ce graphique montre que plus de 80% des employés ont un conjoint à charge.

1. Sexe

# Conversion des données en DataFrame

sexe\_counts = data\_pd['SEXE'].value\_counts().reset\_index()

sexe\_counts.columns = ['SEXE', 'count']

# Calcul des pourcentages

total\_count = sexe\_counts['count'].sum()

sexe\_counts['percentage'] = (sexe\_counts['count'] / total\_count) \* 100

(

    ggplot(sexe\_counts, aes(x='SEXE', y='percentage', fill='factor(SEXE)'))

    + geom\_bar(stat='identity')

    + scale\_fill\_manual(values=['#a6cee3', '#b2df8a'])  # Palette de couleurs attrayantes

    + labs(

        title='Distribution des employés par SEXE',

        x='SEXE',

        y='Pourcentage'

    )

    + theme\_minimal()

    + theme(

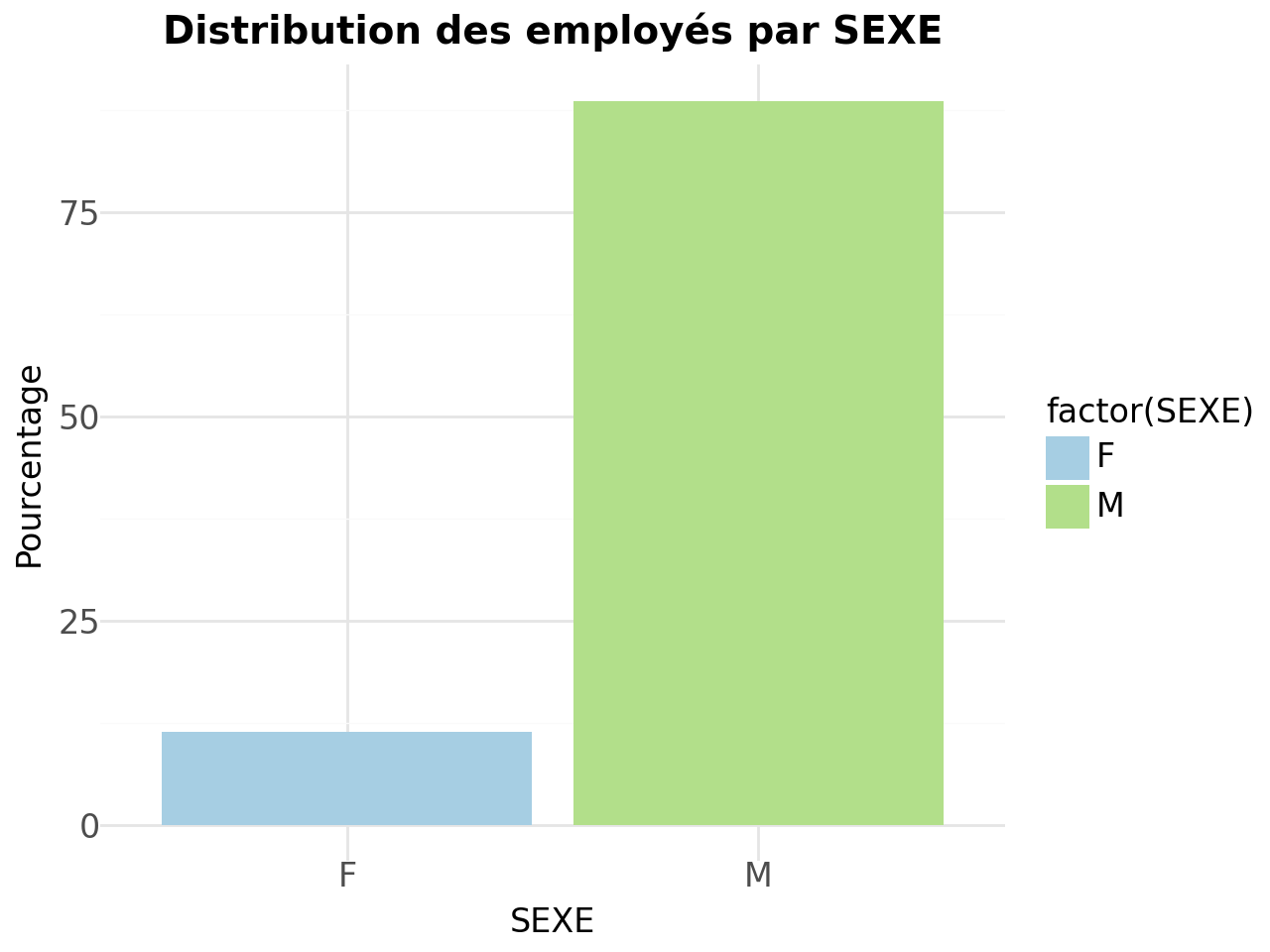
        axis\_text\_x=element\_text(rotation=0, hjust=0.5),

        text=element\_text(size=12),

        plot\_title=element\_text(weight='bold', size=14)

    )

)



Se basant sur le sexe, plus de 85% des employés sont des hommes. Ce qui montre une prédominance des hommes dans cette entreprise.

* **Analyse bivariée**

Categorie vs Sexe

contingency\_table = pd.crosstab(data\_pd['CATEGORIE'], data\_pd['SEXE']).reset\_index()

contingency\_table = pd.melt(contingency\_table, id\_vars='CATEGORIE', var\_name='SEXE', value\_name='count')

(

    ggplot(contingency\_table, aes(x='CATEGORIE', y='count', fill='SEXE'))

    + geom\_bar(stat='identity', position='stack')

    + scale\_fill\_manual(values=['#febd01', '#0db2a2'])

    + labs(

        title='Répartition des employés selon leur CATEGORIE par SEXE',

        x='Catégorie',

        y='Effectif',

        fill='SEXE'

    )

    + theme\_minimal()

    + theme(

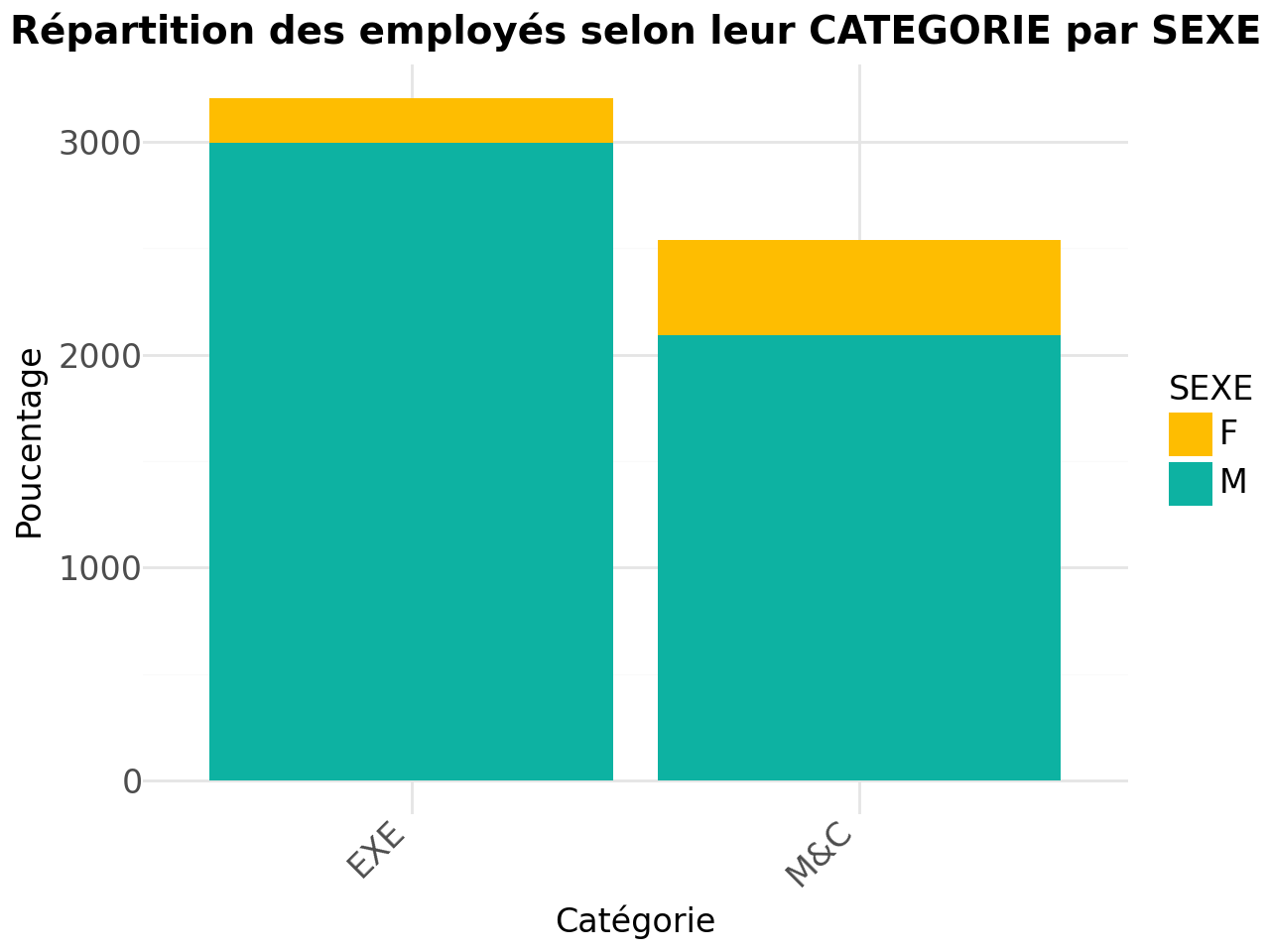
        axis\_text\_x=element\_text(rotation=45, hjust=1),

        text=element\_text(size=12),

        plot\_title=element\_text(weight='bold', size=14)

    )

)



Nous remarquons que dans toutes les catégories confondues, les hommes sont prédominant en termes d’effectif.

Categorie vs Salaire

statistiques\_salaire = data\_pl.group\_by("CATEGORIE").agg([

    pl.col("SALAIRE PERS").count().alias("count"),

    pl.col("SALAIRE PERS").mean().alias("mean"),

    pl.col("SALAIRE PERS").std().alias("std"),

    pl.col("SALAIRE PERS").min().alias("min"),

    pl.col("SALAIRE PERS").quantile(0.25).alias("25%"),

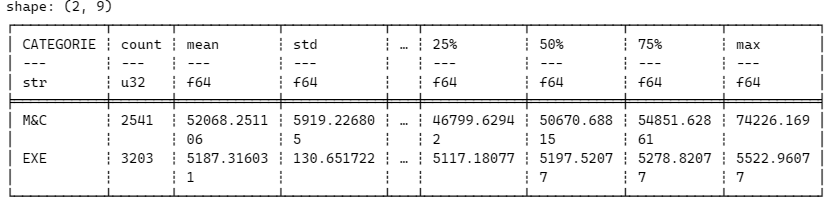
    pl.col("SALAIRE PERS").median().alias("50%"),

    pl.col("SALAIRE PERS").quantile(0.75).alias("75%"),

    pl.col("SALAIRE PERS").max().alias("max")

])

print(statistiques\_salaire)



De ce résultat, on peut voir :

* Le salaire moyen de la catégorie M&C est de 52068.25F, tandis que dans la catégorie EXE le salaire moyen est de 5187.31F ; ce qui largement inférieur à celle de la catégorie M&C.
* Près de 50% des employés dans la catégorie M&C ne dépassent pas le salaire de 50671 ; de même pour ceux qui sont dans la catégorie EXE, ils ne dépassent pas le salaire 5198.

Categorie vs Nombre d'enfant

result = (

    data\_pl\_new.group\_by('CATEGORIE')

    .agg([

        pl.col('ENFANT').mean().alias('mean'),

        pl.col('ENFANT').std().alias('std'),

        pl.col('ENFANT').min().alias('min'),

        pl.col('ENFANT').quantile(0.25).alias('25%'),

        pl.col('ENFANT').median().alias('50%'),  # Médiane

        pl.col('ENFANT').quantile(0.75).alias('75%'),

        pl.col('ENFANT').max().alias('max'),

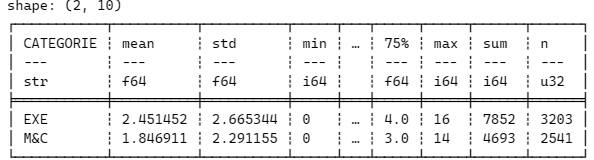
        pl.col('ENFANT').sum().alias('sum'),

        pl.col('ENFANT').count().alias('n')

    ])

)

print(result)



De ce tableau, il résulte comme informations suivantes :

* Dans la catégorie EXE, chaque employé a aux alentours de 3 enfants en moyenne, tandis que dans la catégorie M&C la tendance est autour de deux enfants en moyenne.
* Près de 75% d’employés dans la catégorie EXE ne dépasse pas 4 enfants, avec une exception d’au moins une personne qui a 16 enfants.
* Mais si nous prenons la Catégorie M&C, près de 50% d’employé ont moins d’un enfant ; et près de 75% d’agent ne dépense pas 3 enfants. Il y aussi une exception d’au moins un agent qui a 14 enfants.

De cette analyse, on peut donc dire, les employés qui ont un salaire faible, ont tendance a avoir plus d’enfants comparativement à ceux qui ont un salaire assez élevé.

Categorie vs INDEPOUSE

# Création de la table de contingence avec group\_by et pivot

contingency\_table = (

    data\_pl\_new.group\_by(['CATEGORIE', 'INDEPOUSE'])

    .agg(pl.count().alias('count'))

    .pivot(

        values='count',

        index='CATEGORIE',

        columns='INDEPOUSE'

    )

    .fill\_null(0)

)

# Calcul des pourcentages par rapport à la somme de chaque ligne dans Polars

contingency\_table\_percent = contingency\_table.with\_columns([

    ((pl.col('0') / (pl.col('0') + pl.col('1'))) \* 100).alias('INDEPOUSE\_0'),

    ((pl.col('1') / (pl.col('0') + pl.col('1'))) \* 100).alias('INDEPOUSE\_1')

])

# Configuration des styles pour le graphique

plt.style.use('ggplot')

plt.rcParams['font.family'] = 'DejaVu Sans'

plt.rcParams['axes.edgecolor'] = '#333F4B'

plt.rcParams['axes.linewidth'] = 0.8

plt.rcParams['xtick.color'] = '#333F4B'

plt.rcParams['ytick.color'] = '#333F4B'

plt.rcParams['text.color'] = '#333F4B'

# Création du graphique avec les pourcentages

contingency\_table\_percent[['CATEGORIE', 'INDEPOUSE\_0', 'INDEPOUSE\_1']].to\_pandas().plot(kind='barh', stacked=True, figsize=(8, 4), color=['#007ACC', '#FF5733'], alpha=0.85)

# Configuration des axes et des étiquettes

plt.title('Répartition des Catégories selon INDEPOUSE', fontsize=16, weight='bold', color='#333F4B')

plt.xlabel('Pourcentage', fontsize=14, weight='bold', color='#333F4B')

plt.ylabel('Catégorie', fontsize=14, weight='bold', color='#333F4B')

plt.xticks(fontsize=12)

plt.yticks(fontsize=12)

# Suppression des spines inutiles

ax = plt.gca()

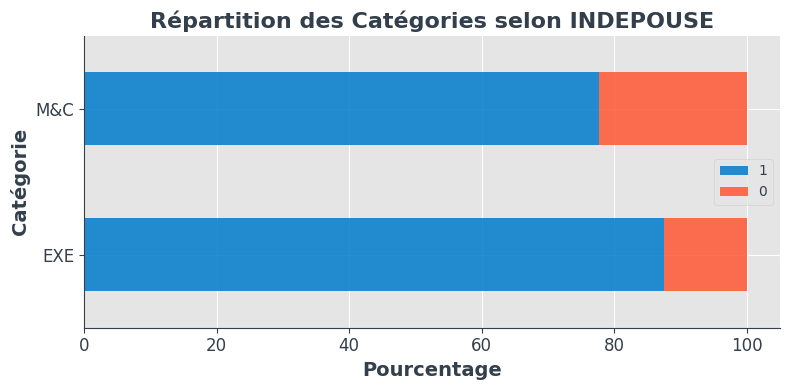
ax.spines['top'].set\_visible(False)

ax.spines['right'].set\_visible(False)

# Affichage du graphique

plt.tight\_layout()

plt.show()



Nous constatons dans chaque catégorie que la proportion des hommes est largement supérieure à celle des femmes. Mais cela est dû à la faible représentativité des femmes par rapport aux hommes au sein de cette société, car les femmes sont moins de 15% dans cette firme.

* **Etude des liaisons et de corrélations**

Sur ce point nous allons essayer de voir le rapprochement qui existe entre différentes variables, car si les variables sont fortement liées cela risquera de créer des problèmes lors de la modélisation. Par exemple le problème de multicolinéarité[[1]](#footnote-1).

* + - 1. Cas des variables Quantitatives

Ici nous faire la corrélation des variables quantitatives de Pearson.

# Calcul de la matrice de corrélation

corr\_matrix = data\_pd[['SALAIRE PERS', 'ENFANT', 'TRP', 'LGT', 'SSANTE', 'ENERGIE', 'IFT', 'IDP', 'IPM', 'age\_engagement']].corr()

# Masque pour afficher uniquement la moitié supérieure de la matrice

mask = np.triu(np.ones\_like(corr\_matrix, dtype=bool))

# Création d'une figure

plt.figure(figsize=(12, 10))

sns.set\_theme(style="white")

# Création de la heatmap avec un design plus sophistiqué

sns.heatmap(

    corr\_matrix,

    mask=mask,  # Application du masque

    annot=True,  # Affiche les valeurs

    fmt=".2f",  # Format des valeurs

    cmap='Spectral',  # Palette de couleurs attrayante

    center=0,  # Centre la palette autour de 0

    cbar\_kws={'shrink': 0.75, 'aspect': 20},  # Configuration de la barre de couleur

    linewidths=0.5,  # Espacement entre les cellules

    linecolor='gray',  # Couleur des lignes

    square=True  # Format carré pour chaque cellule

)

# Personnalisation des axes et du titre

plt.title("Corrélations des variables quantitatives", fontsize=18, weight='bold', color='#333F4B')

plt.xticks(rotation=45, ha='right', fontsize=12, weight='bold')

plt.yticks(rotation=0, fontsize=12, weight='bold')

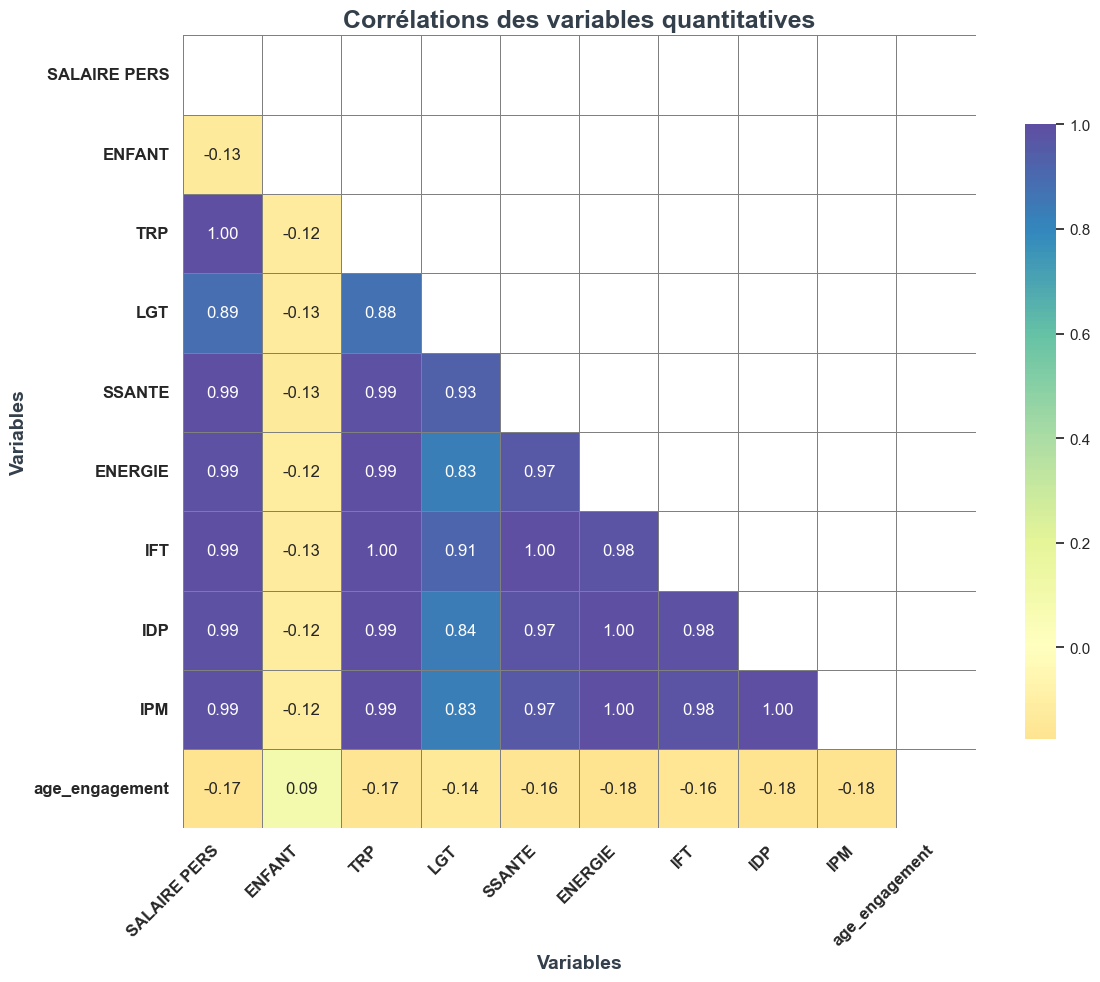
plt.xlabel('Variables', fontsize=14, weight='bold', color='#333F4B')

plt.ylabel('Variables', fontsize=14, weight='bold', color='#333F4B')

# Affichage du graphique

plt.tight\_layout()

plt.show()



Nous remarquons des très fortes corrélations dans le sens positif entre différentes variables, ceci signifie que lorsque la valeur d'une variable augmente, la valeur de l'autre augmente **exactement** de la même manière, de façon proportionnelle.

Or, ces corrélations très fortes ou parfaites ne sont pas bien pour la modélisation, ça créera des problèmes de multicolinéarité.

D’où, certaines variables ne doivent pas figurer dans la modélisation (cfr : les variables en bleu sur le graphique).

* + - 1. Cas des variables Qualitatives

La corrélation dans le cadre des variables qualitative est appelée liaison, mais elle ne peut pas être calculée de la même manière que pour les variables quantitatives, car les méthodes classiques comme le **coefficient de corrélation de Pearson** ne s'appliquent qu'aux données numériques. Cependant, il existe d'autres méthodes adaptées pour mesurer l'association ou la dépendance entre des variables qualitatives :

### 1**. Coefficient de corrélation de Cramér (Cramér's V)**

* Il est utilisé pour mesurer la force de l'association entre deux variables catégorielles.
* Calculé à partir du **test du Chi-carré**, Cramér's V varie entre 0 (aucune association) et 1 (association parfaite).
* Il convient lorsque les variables ont plus de deux catégories.

### 2. **Coefficient Phi (Φ)**

* Utilisé pour mesurer l'association entre deux variables binaires (ayant chacune deux catégories).
* Il s'agit d'un cas particulier de Cramér's V lorsque le tableau de contingence est de dimension 2x2.

### 3. **Test du Chi-carré d'indépendance**

* Permet de vérifier si deux variables qualitatives sont statistiquement indépendantes ou s'il existe une association entre elles.
* Bien qu'il ne donne pas directement un coefficient de corrélation, il permet de déterminer si une relation significative existe entre les variables.

### 4. **Indice de corrélation de Theil (Theil's U ou l'entropie conditionnelle)**

* Utile pour mesurer la dépendance asymétrique entre deux variables catégorielles.
* Contrairement aux méthodes ci-dessus, cet indice peut être asymétrique (la relation d'une variable à l'autre peut ne pas être réciproque).

### 5. **Corrélation de Kendall’s Tau pour les variables ordinales**

* Si vos variables qualitatives sont ordinales (elles ont un ordre mais ne sont pas numériques), vous pouvez utiliser la **corrélation de Kendall’s Tau** pour mesurer la force de l'association ordonnée.

### 6. **Tableaux de contingence**

* En plus des coefficients, il est courant de visualiser les associations entre les variables qualitatives à l'aide de **tableaux croisés** (ou tableaux de contingence). Ces tableaux montrent les fréquences de chaque combinaison possible de catégories entre les variables.

# Étape 1: Créer le tableau de contingence

contingency\_table = pd.crosstab(index=[data\_pd['CATEGORIE'], data\_pd['SEXE']], columns=data\_pd['INDEPOUSE'])

# Étape 2: Afficher le tableau de contingence avec un design amélioré

print(Fore.MAGENTA + Style.BRIGHT + "### Tableau de Contingence ###")

print(Style.RESET\_ALL)

# Format de tableau plus esthétique avec bordures et alignement

print(Fore.CYAN + tabulate(contingency\_table, headers='keys', tablefmt='fancy\_grid', showindex=True, stralign="center"))

print(Style.RESET\_ALL)

print(Fore.MAGENTA + Style.BRIGHT + "-"\*40 + Style.RESET\_ALL)  # Séparateur visuel

# Étape 3: Définir la fonction pour calculer le Cramér's V

def cramers\_v(confusion\_matrix):

    """

    Calcule la statistique de Cramér's V pour mesurer l'association entre les variables catégorielles.

    :param confusion\_matrix: Un tableau de contingence

    :return: La valeur de Cramér's V (entre 0 et 1)

    """

    # Statistique du test du chi-carré et nombre total d'observations

    chi2\_statistic = ss.chi2\_contingency(confusion\_matrix)[0]

    total\_obs = confusion\_matrix.sum().sum()

    # Calcul du phi-squared

    phi2 = chi2\_statistic / total\_obs

    # Nombre de lignes et de colonnes dans le tableau de contingence

    rows, cols = confusion\_matrix.shape

    # Appliquer les corrections pour phi-squared

    phi2\_corrected = max(0, phi2 - ((cols - 1) \* (rows - 1)) / (total\_obs - 1))

    rows\_corr = rows - ((rows - 1) \*\* 2) / (total\_obs - 1)

    cols\_corr = cols - ((cols - 1) \*\* 2) / (total\_obs - 1)

    # Retourner la racine carrée de la valeur corrigée de phi-squared

    return np.sqrt(phi2\_corrected / min((cols\_corr - 1), (rows\_corr - 1)))

# Étape 4: Calcul de Cramér's V

cramers\_v\_value = cramers\_v(contingency\_table)

# Étape 5: Affichage du résultat avec un design plus moderne

print(Fore.GREEN + Style.BRIGHT + "\n### Résultat Cramér's V ###")

print(Fore.GREEN + "-"\*40)  # Ligne de séparation

print(f"{Style.BRIGHT}Cramér's V: {cramers\_v\_value:.4f}{Style.RESET\_ALL}")

print(Fore.GREEN + "-"\*40)  # Ligne de séparation

# Interprétation de la valeur de Cramér's V

if cramers\_v\_value < 0.1:

    print(Fore.YELLOW + Style.BRIGHT + "Faible association entre les variables." + Style.RESET\_ALL)

elif cramers\_v\_value < 0.3:

    print(Fore.CYAN + Style.BRIGHT + "Association modérée entre les variables." + Style.RESET\_ALL)

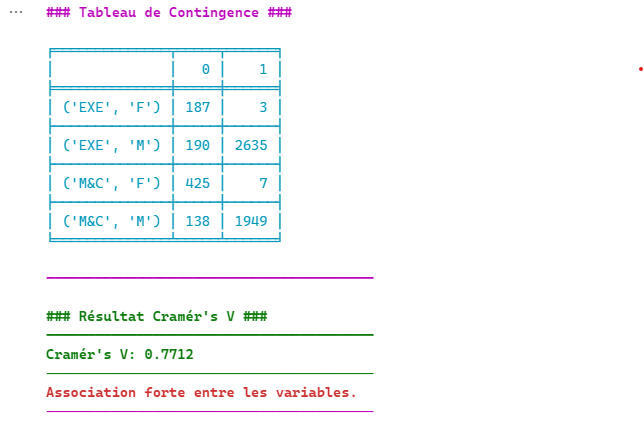
else:

    print(Fore.RED + Style.BRIGHT + "Association forte entre les variables." + Style.RESET\_ALL)

# Réinitialiser les styles à la fin

print(Fore.MAGENTA + "-"\*40 + Style.RESET\_ALL)  # Séparateur final

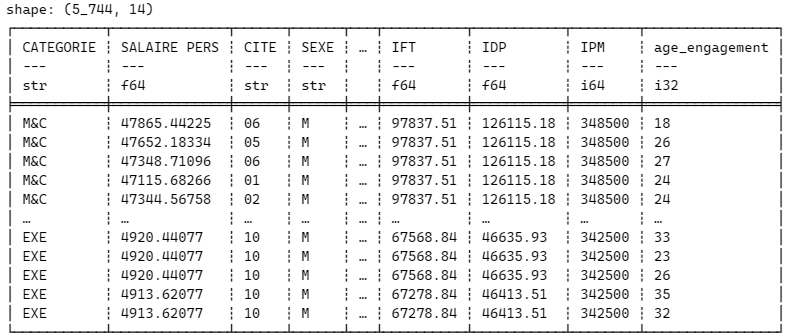
print(Style.RESET\_ALL)



Nous remarquons une association forte entre la catégorie, le sexe et l’INDEPOUSE ; la force de cette association est de 77%.

**Modélisation**

Pour l’instant certaines variables sont à exclure à cause des incohérences sur ces variables. Parmi ces variables, nous excluons : 'DERNIER AVACEMENT EN GRADE', 'COTATION', 'DATENAISSA', 'DATEENGAGE', 'tranche\_age\_engagement', 'GRADE','ETATCIVIL'



#### Sélection des variables pour le modèle

data\_model = data\_pl\_new.select(

    pl.all().exclude(['DERNIER AVACEMENT EN GRADE', 'COTATION', 'DATENAISSA', 'DATEENGAGE', 'tranche\_age\_engagement', 'GRADE','ETATCIVIL' ])

)

# Convertir les données Polars en DataFrame Pandas pour la manipulation

df = data\_model.to\_pandas()

# Identifier les colonnes catégorielles

colonnes\_categoricielles = ["CITE", "SEXE", "INDEPOUSE"]

# Appliquer un encodage LabelEncoder sur ces colonnes

label\_encoders = {}

for col in colonnes\_categoricielles:

    le = LabelEncoder()

    df[col] = le.fit\_transform(df[col])

    label\_encoders[col] = le

# Encodage de la variable cible

df["CATEGORIE"] = LabelEncoder().fit\_transform(df["CATEGORIE"])

# Séparer les données en X et y

X = df.drop(columns=["CATEGORIE"])

y = df["CATEGORIE"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42) # 80% des données sont utilisées pour l’entraînement

# Initialiser et entraîner le modèle de régression logistique

model = LogisticRegression(max\_iter=1000)

model.fit(X\_train, y\_train)

# Prédictions et évaluation

y\_pred = model.predict(X\_test)

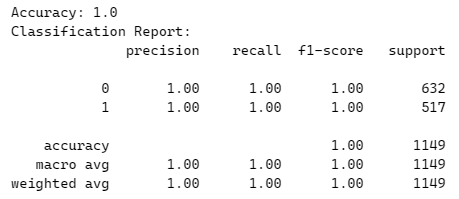
accuracy = accuracy\_score(y\_test, y\_pred)

report = classification\_report(y\_test, y\_pred)

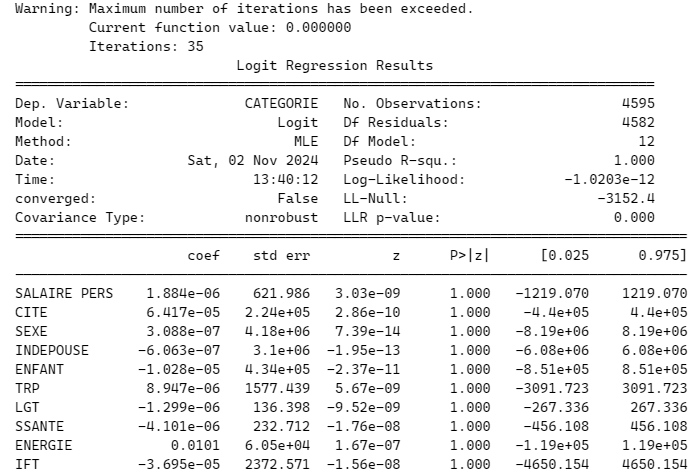
print(f"Accuracy: {accuracy}")

print("Classification Report:")

print(report)



Les résultats de ce modèle paraissent meilleurs alors qu’il y a un problème, soulevé lors de l’analyse bivariées.

Pour mieux voir, les résultats nous utilisons la librairie statmodels :  


Avec ces résultats, aucune variable n’est significative. Or ce problème a été décelé au niveau de l’analyse des corrélations.

Pour ça, on doit identifier les variables à problèmes à l’aide de l’étude de la multicolinéarité.

**Etude de la multicolinéarité**

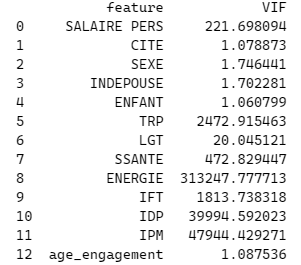
# VIF pour chaque variable

vif\_data = pd.DataFrame()

vif\_data["feature"] = X.columns

vif\_data["VIF"] = [variance\_inflation\_factor(X.values, i) for i in range(X.shape[1])]

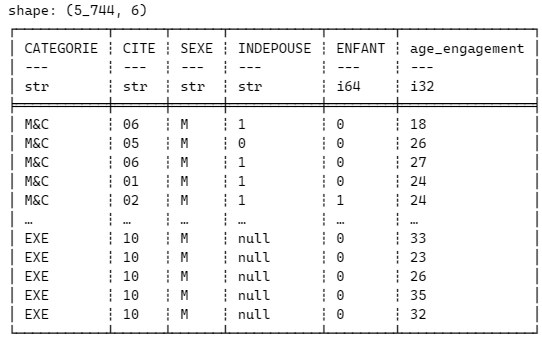
print(vif\_data)



Les variables à éliminer sont celles la valeur du VIF excède 10 ; ce qui confirme notre présomption au niveau de la corrélation.

La meilleure stratégie est d’éliminer une variable après une autre et voir le comportement du modèle. Mais sur base de notre hypothèse sur la corrélation certaines variables ne doivent pas figurer ensemble dans la modélisation. Entre autres : Salaire pers, TRP, LGT, SSANTE, ENERGIE, IFT, IDP, IPM.

Après plusieurs tests de suppression des variables, celles retenues sont :



# Convertir les données Polars en DataFrame Pandas pour la manipulation

df = data\_model.to\_pandas()

# Identifier les colonnes catégorielles

colonnes\_categoricielles = ["CITE", "INDEPOUSE", 'SEXE']

# Appliquer un encodage LabelEncoder sur ces colonnes

label\_encoders = {}

for col in colonnes\_categoricielles:

    le = LabelEncoder()

    df[col] = le.fit\_transform(df[col])

    label\_encoders[col] = le

# Encodage de la variable cible

df["CATEGORIE"] = LabelEncoder().fit\_transform(df["CATEGORIE"])

# Séparer les données en X et y

X = df.drop(columns=["CATEGORIE"])

y = df["CATEGORIE"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialiser et entraîner le modèle de régression logistique

model = LogisticRegression(max\_iter=1000)

model.fit(X\_train, y\_train)

# Prédictions et évaluation

y\_pred = model.predict(X\_test)

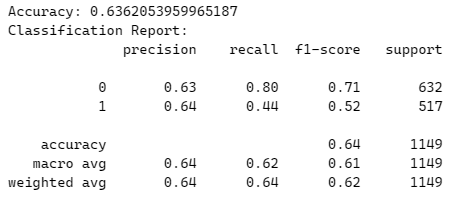
accuracy = accuracy\_score(y\_test, y\_pred)

report = classification\_report(y\_test, y\_pred)

print(f"Accuracy: {accuracy}")

print("Classification Report:")

print(report)



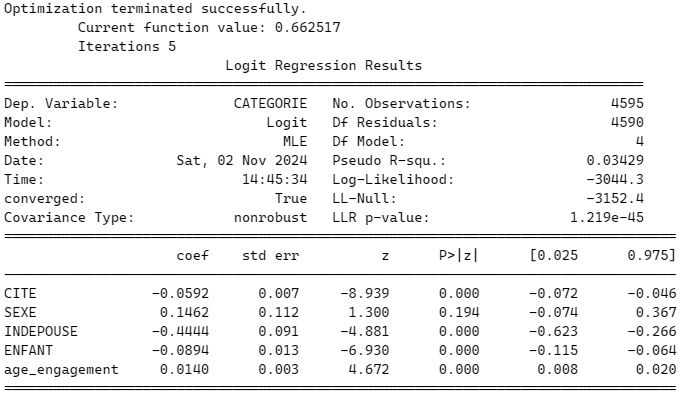
Pour mieux interpréter les résultats, nous utilisons la librairie statmodels :

model = sm.Logit(y\_train, X\_train)

result = model.fit(maxiter=1000)

# Afficher le résumé du modèle

print(result.summary())



On sait maintenant voir que la modèle a une précision de 66% sur les données d’entraînement et une précision de 63% celles de test.

En plus, tous les paramètres sont significatifs pour l’explication du modèle à un niveau de confiance de 95%.

Recommandations

Ces analyses sont faites sur base de la nature des variables, d’autres pistes de recherche sont aussi viables, entre autres :

* Exploration des autres modèles de Machine learning (Analyse discriminante, K-means, …)
* Ajout d’une régularisation : cas de la régression logistique avec régularisation (comme Ridge ou Lasso)
* Réaliser l'AFDM (Analyse Factorielle Des Données Mixtes), afin de détecter les variables les plus corrélées avec la catégorie de l’employé, puis les éliminer.
* Etc.

Code source du travail

1. La **multicolinéarité** est un phénomène statistique qui survient lorsque deux ou plusieurs variables indépendantes dans un modèle de régression sont fortement corrélées. Cela signifie qu'il existe une forte relation linéaire entre ces variables, ce qui complique l'estimation des coefficients de la régression.

   Mathématiquement, c’est si l’une au moins des variables peut s’écrire comme combinaison linéaire des autres variables. [↑](#footnote-ref-1)