December 18 , 2023

Analyse Approfondie des Inégalités Éducatives et de l'Illettrisme

22038 _ 22088 _ 22106

1. BUSINESS UNDERSTANDING

1.1 1. Présentation des Données

Les données exploitées dans le cadre de cette étude découlent d'une enquête menée en 2019 en Mauritanie, spécifiquement de l'Enquête EPCV (Enquête permanente sur les conditions de vie des ménages) . Cette enquête vise à comprendre les dynamiques éducatives au sein des ménages mauritaniens et autres comme la qualité de vie et la santé .

L'objectif principal de cette étude est de comprendre les inégalités éducatives et les niveaux d'illettrisme .

1.2 Les variables

Nous avons examiné plusieurs variables éducatives et socio-économiques pertinentes dans le cadre de cette étude.

Variables Démographiques et Familiales:

- **BN (Nom et Prénoms)**
- **B1 (Lien de parenté avec le chef de ménage)**
- **B2 (Sexe)**
- **B4 (Age)**
- **B5 (Statut matrimonial)**
- **B6 (Contribue aux dépenses du ménage)**

- **B7 (Nationalité)**
- **B7A1 (État de vie du père)**
- **B7B1 (État de vie de la mère)**

Variables Éducatives:

- **C1 (Sait lire ou écrire)**
- **C1AA, C1AB, C1AC (Lire et écrire en Arabe, Français, Anglais)**
- **C2 (Est déjà allé à l'école)**
- **C2A (Raison de non-scolarisation)**
- **C3 (Année d'inscription en 1AF)**
- **C4N, C4C, C4A (Niveau, Classe, Diplôme le plus élevé)**

Variables Financières de l'Éducation:

- **C1D (Frais d'inscription)**
- **C1EM, C1EF, C1EN (Frais de scolarité)**
- **C1FM, C1FF, C1FN (Cours particuliers)**
- **C1G (Frais liés aux livres)**
- **C1GM, C1GF, C1GN (Fournitures): Montant, fréquence, et nombre de fois des frais liés aux fournitures.**
- **C1IM, C1IF, C1IN (Transport): Montant, fréquence, et nombre de fois des frais de transport.**

Variables sur l'Éducation Post-Scolaire:

- **C1J (Non spécifié): Catégorie non spécifiée.**
- **C10, C10_A (Source de financement et lien de parenté): Principale source de financement des études et lien de parenté avec le payeur.**
- **C13, C14, C141 (Formation professionnelle): Suivi de formation professionnelle, type et niveau de formation.**

Variables Géographiques:

- **wilaya, moughataa, commune (Wilaya, Moughataa, Commune): Localisation géographique.**
- **milieu (Milieu de résidence).**

2. Data Preparation & Data Exploration

Pour mener à bien cette étude, nous avons travaillé sur un ensemble de données initial comprenant 60908 observations et 437 variables. Nous avons ensuite procédé à la réduction de ce dataset en un sous-ensemble de variables pertinentes pour notre thème, soit un ensemble final de 44 variables.

2.1 Importations des datasets

In [2]:
```
import pandas as pd

chemin_du_fichier = r'C:/Users/Dell/Desktop/mm.csv'

df = pd.read_csv(chemin_du_fichier)
print(df)
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_5688\873912781.py:5: DtypeWarning: Col
umns (82,114,118,120,124,161,189,190,240,249,253,254,255,261,265,313,355) have mi
xed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv(chemin_du_fichier)
```

```
       US_ORDRE  A7   B0                  BN             B1        B2  B4  \
0             1   1  3.0     الخليفة ولد يحي  Fils ou filles  Masculin  34
1             1   1  2.0  خديجه منت سيد احمد       Epoux (se)   Féminin  60
2             1   1  4.0    سيد محمد ولد يحي  Fils ou filles  Masculin  15
3             1   1  1.0        يحي ولد محمد  Chef de ménage  Masculin  68
4             1   2  5.0    اميلمنين بنت جدو  Fils ou filles   Féminin   6
...         ...  ..  ...                 ...             ...       ...  ..
60903      1078  10  1.0     محمد ولد اميليد  Chef de ménage  Masculin  60
60904      1078  10  2.0      اعويشة منت احمد       Epoux (se)   Féminin  40
60905      1078  10  3.0       مريم منت محمد  Fils ou filles   Féminin  13
60906      1078  10  5.0      اسلام ولد محمد  Fils ou filles  Masculin   5
60907      1078  10  4.0      النانه منت محمد  Fils ou filles   Féminin  10

                        B5  B5ACJ1  B5ACJ2  ...  autrdepscol_20a  inscrip_18a  \
0      Marié(e) (monogame)     0.0     NaN  ...              NaN          NaN
1      Marié(e) (monogame)     NaN     NaN  ...              NaN          NaN
2          Jamais marié(e)     NaN     NaN  ...              NaN          NaN
3      Marié(e) (monogame)     2.0     NaN  ...              NaN          NaN
4                      NaN     NaN     NaN  ...           5000.0          NaN
...                    ...     ...     ...  ...              ...          ...
60903  Marié(e) (monogame)     2.0     NaN  ...              NaN          NaN
60904  Marié(e) (monogame)     NaN     NaN  ...              NaN          NaN
60905      Jamais marié(e)     NaN     NaN  ...              0.0          0.0
60906                  NaN     NaN     NaN  ...              NaN          NaN
60907      Jamais marié(e)     NaN     NaN  ...              0.0          0.0

      livre_18a cotis_18a autrdepscol_18a  vag inscrip_1920a  cotis_1920a  \
0           NaN       NaN             NaN    2           NaN          NaN
1           NaN       NaN             NaN    2           NaN          NaN
2           NaN       NaN             NaN    2           NaN          NaN
3           NaN       NaN             NaN    2           NaN          NaN
4           NaN       NaN             NaN    2           NaN          NaN
...         ...       ...             ...  ...           ...          ...
60903       NaN       NaN             NaN    4           NaN          NaN
60904       NaN       NaN             NaN    4           NaN          NaN
60905       0.0       0.0             0.0    4           0.0          0.0
60906       NaN       NaN             NaN    4           NaN          NaN
60907       0.0       0.0             0.0    4           0.0          0.0

      livre_1920a autrdepscol_1920a
0             NaN               NaN
1             NaN               NaN
2             NaN               NaN
3             NaN               NaN
4             NaN               NaN
...           ...               ...
60903         NaN               NaN
60904         NaN               NaN
60905   1333.3334               0.0
60906         NaN               NaN
60907   1333.3334               0.0

[60908 rows x 437 columns]
```

In [3]:
```
variables_a_conserver = [ 'wilaya', 'moughataa', 'commune', 'milieu'
    ,'BN', 'B1', 'B2', 'B4', 'B5', 'B6', 'B7', 'B7A1', 'B7B1',
    'C1', 'C1AA', 'C1AB', 'C1AC', 'C2', 'C2A', 'C3', 'C4N', 'C4C', 'C4A',
    'C1D', 'C1EM', 'C1EF', 'C1EN', 'C1FM', 'C1FF', 'C1FN', 'C1G', 'C1GM',
    'C1GF', 'C1GN', 'C1IM', 'C1IF', 'C1IN', 'C1J', 'C10', 'C10_A', 'C13', 'C14',
    'C141', 'C15'
```

```
]

data = df[variables_a_conserver]
```

In [4]:
```
data.shape
```

Out[4]:
```
(60908, 44)
```

In [5]:
```
data.dtypes
```

Out[5]:
```
wilaya        object
moughataa     object
commune       object
milieu        object
BN            object
B1            object
B2            object
B4            object
B5            object
B6            object
B7            object
B7A1          object
B7B1          object
C1            object
C1AA          object
C1AB          object
C1AC          object
C2            object
C2A           object
C3            object
C4N           object
C4C          float64
C4A           object
C1D          float64
C1EM         float64
C1EF          object
C1EN         float64
C1FM         float64
C1FF          object
C1FN         float64
C1G          float64
C1GM         float64
C1GF          object
C1GN         float64
C1IM         float64
C1IF          object
C1IN         float64
C1J          float64
C10           object
C10_A         object
C13           object
C14           object
C141          object
C15           object
dtype: object
```

2.2_ Analyse des Valeurs Manquantes

Les valeurs manquantes dans la colonne 'B4' (âge) ont été remplacées par la médiane de cette colonne pour assurer une imputation basée sur une mesure centrale représentative de l'âge dans le dataset. De plus, le type de la colonne 'B4' a été corrigé pour refléter des valeurs entières après l'imputation.

Calcul des proportions de valeurs manquantes pour chaque colonne

Nous avons calculé les proportions de valeurs manquantes pour chaque colonne dans le dataset. Ceci est une étape importante pour évaluer la qualité des données. Selon les résultats obtenus, nous pouvons prendre des mesures appropriées, telles que l'imputation des valeurs manquantes ou l'élimination de certaines colonnes si nécessaire.

In [6]:
```
proportions_valeurs_manquantes = data.isna().mean()
print("Proportions de valeurs manquantes pour chaque colonne :\n")
print(proportions_valeurs_manquantes)
```

```
Proportions de valeurs manquantes pour chaque colonne :

wilaya       0.000000
moughataa    0.000000
commune      0.000000
milieu       0.000000
BN           0.005057
B1           0.005057
B2           0.005057
B4           0.005057
B5           0.314228
B6           0.145826
B7           0.005057
B7A1         0.256715
B7B1         0.256715
C1           0.221088
C1AA         0.436265
C1AB         0.436265
C1AC         0.436265
C2           0.087772
C2A          0.758143
C3           0.329628
C4N          0.329628
C4C          0.544460
C4A          0.341400
C1D          0.734698
C1EM         0.734698
C1EF         0.963535
C1EN         0.963535
C1FM         0.735043
C1FF         0.925379
C1FN         0.925379
C1G          0.734698
C1GM         0.734698
C1GF         0.785792
C1GN         0.785792
C1IM         0.734698
C1IF         0.977359
C1IN         0.977359
C1J          0.734698
C10          0.763151
C10_A        0.792375
C13          0.754696
C14          0.989886
C141         0.005057
C15          0.989886
dtype: float64
```

Imputation des valeurs manquantes dans les colonnes de type float64

Les colonnes de type float64 dans le dataset ont été examinées, et les valeurs manquantes ont été remplacées par zéro (0). Cette stratégie d'imputation peut être appliquée lorsque des valeurs nulles sont appropriées. Par exemple, dans le cas des dépenses de transport, certaines familles ne sont peut-être pas loin de l'école, donc leurs dépenses de transport sont null et la meme maniere pour les autres .e.

In [7]:
```
float_variables = data.select_dtypes(include='float64').columns
data[float_variables] = data[float_variables].fillna(0)
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_5688\1421052707.py:2: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  data[float_variables] = data[float_variables].fillna(0)
```

Supprimer les doublons : nous avons effectué une opération de nettoyage en supprimant les doublons dans le DataFrame data à l'aide de la fonction drop_duplicates(). Ensuite, nous avons affiché la forme du DataFrame après cette opération .

In [8]:
```
data = data.drop_duplicates()
print("Forme du DataFrame après suppression des doublons :", data.shape)
```

```
Forme du DataFrame après suppression des doublons : (60659, 44)
```

2.3_Vérification de la Cohérence des Données

Nous avons spécifiquement ciblé les colonnes 'B4 :AGE' (âge) et 'wilaya' (localisation géographique) pour plusieurs raisons :

1. **'B4 :AGE' (âge) :**
    - L'âge est souvent soumis à des contraintes spécifiques, par exemple, il ne devrait pas contenir de valeurs négatives ou des catégories non valides.
    - Nous voulons nous assurer que toutes les valeurs dans cette colonne sont numériques et représentent des âges valides.
2. **'wilaya' (localisation géographique) :**
    - La colonne 'wilaya' devrait contenir des valeurs correspondant à des régions géographiques spécifiques.
    - Nous voulons nous assurer qu'il n'y a pas de fautes de frappe, de valeurs manquantes ou d'autres incohérences qui pourraient compromettre la qualité des données géographiques.

In [9]:
```
colonnes_a_verifier = ['B4', 'wilaya']
for colonne in colonnes_a_verifier:
    print(f"\nValeurs uniques dans la colonne '{colonne}' :")
    print(data[colonne].unique())
```

```
Valeurs uniques dans la colonne 'B4' :
['34' '60' '15' '68' '6' '13' '9' '48' '36' '26' '14' '18' '12' '40' '2'
 '5' '21' '4' '1' '7' '28' '39' '24' '32' '58' '30' '35' '59' '45' '11'
 '8' '10' '17' '33' '3' '38' '50' '20' '22' '46' '62' '19' '0' '31' '56'
 '70' '25' '53' 'NSP' '16' '41' '23' '65' '52' '44' '69' '49' '27' '64'
 '54' '37' '47' '51' '29' '74' '95+' '61' '43' '42' '79' '75' '72' '67'
 '66' '57' '55' '77' '81' '78' '73' '86' '84' '71' '63' nan '87' '80' '89'
 '76' '90' '91' '85' '83' '82' '88' '92' '93' '94']

Valeurs uniques dans la colonne 'wilaya' :
['Hodh charghy' 'Hodh Gharby' 'Assaba' 'Gorgol' 'Brakna' 'Trarza' 'Adrar'
 'Dakhlett Nouadibou' 'Tagant' 'Guidimagha' 'Tirs-ezemour' 'Inchiri'
 'Nouakchott']
```

Nous avons observé la présence de la valeur 'NSP' dans la colonne 'B4', qui représente l'âge. Pour traiter cette incohérence, nous avons effectué l'étape suivant :

Remplacement des valeurs manquantes : Les valeurs manquantes dans la colonne 'B4' ont été remplacées par la médiane de cette colonne.

In [10]:
```python
data['B4'] = pd.to_numeric(data['B4'], errors='coerce')
median_age = data['B4'].median()
data['B4'].fillna(median_age, inplace=True)
data['B4'] = data['B4'].astype(int)
colonnes_a_verifier = ['B4']
for colonne in colonnes_a_verifier:
    print(f"\nValeurs uniques dans la colonne '{colonne}' :")
    print(data[colonne].unique())
```

```
Valeurs uniques dans la colonne 'B4' :
[34 60 15 68  6 13  9 48 36 26 14 18 12 40  2  5 21  4  1  7 28 39 24 32
 58 30 35 59 45 11  8 10 17 33  3 38 50 20 22 46 62 19  0 31 56 70 25 53
 16 41 23 65 52 44 69 49 27 64 54 37 47 51 29 74 61 43 42 79 75 72 67 66
 57 55 77 81 78 73 86 84 71 63 87 80 89 76 90 91 85 83 82 88 92 93 94]
```

2_4 Traitement des Frais Scolaires

Calcul des Frais Totaux

Nous avons créé une nouvelle colonne 'Frais_Totaux' dans le dataset pour représenter le coût total des frais liés à l'éducation. Cette colonne a été calculée en additionnant les frais d'inscription (C1D) et les frais de scolarité mensuels (C1EM) multipliés par la fréquence (C1EF) et le nombre de fois (C1EN).

La fréquence des frais de scolarité a été cartographiée comme suit :

- 'Mensuel': 1 (par mois)
- 'Trimestrielle': 3 (par trimestre)
- 'Un seul paiement': 1 (paiement unique)

Le résultat du calcul des frais totaux est présenté dans le dataset sous la colonne 'Frais_Totaux'.

In [11]:
```python
import numpy as np
```

```
data['Frais_Totaux'] = np.nan
frequence_mapping = {'Mensuel': 1, 'Trimestrielle': 3, 'Un seul paiement': 1}
data['C1EN'].replace('nan', 0, inplace=True)
for index, row in data.iterrows():
    frais_inscription = row['C1D']
    frais_scolarite_montant = row['C1EM']
    frequence = row['C1EF']
    nombre_de_fois = row['C1EN']
    if frequence in frequence_mapping:
        frais_totaux = frais_inscription + (frais_scolarite_montant * frequence_
    else:
        frais_totaux = frais_inscription + (frais_scolarite_montant * 1 * nombre

    data.at[index, 'Frais_Totaux'] = frais_totaux
```

Nous avons supprimé les colonnes redondantes ('C1D', 'C1EM', 'C1EF', 'C1EN') du dataset, car les informations qu'elles contiennent ont été intégrées dans la nouvelle colonne 'Frais_Totaux'.

Suite à la création de la colonne 'Frais_Totaux', nous avons calculé les valeurs minimale et maximale de cette colonne pour l'année 2020. Ces valeurs permettent d'obtenir un aperçu des coûts totaux des frais éducatifs dans le dataset.

In [12]:
```
data.drop(columns=['C1D', 'C1EM', 'C1EF', 'C1EN'], inplace=True)

min_frais_totaux = data['Frais_Totaux'].min()
max_frais_totaux = data['Frais_Totaux'].max()

print("Valeur minimale des frais totaux  :", min_frais_totaux)
print("Valeur maximale des frais totaux :", max_frais_totaux)
```

```
Valeur minimale des frais totaux  : 0.0
Valeur maximale des frais totaux : 2135000.0
```

Nous avons créé une nouvelle colonne 'Frais_livre_fornuture_Totaux' en calculant les coûts totaux pour les frais liés aux livres et aux fournitures. Les informations nécessaires ont été extraites des colonnes 'C1GM' (Montant des fournitures), 'C1GF' (Fréquence des fournitures), et 'C1GN' (Nombre de fois des fournitures).

Nous avons utilisé une stratégie similaire à celle employée pour les frais totaux précédemment. Les montants, fréquences et nombres de fois ont été combinés pour obtenir les coûts totaux.

Nous avons également supprimé les colonnes redondantes ('C1GM', 'C1GF', 'C1GN') du dataset, puisque les informations qu'elles contiennent ont été consolidées dans la nouvelle colonne 'Frais_livre_fornuture_Totaux'.

In [13]:
```
import numpy as np

colonnes_a_traiter = ('C1G', 'C1GM', 'C1GF', 'C1GN')

data['Frais_livre_fornuture_Totaux'] = np.nan

frequence_mapping = {'Mensuel': 1, 'Trimestrielle': 3, 'Un seul paiement': 1}
```

```
data['C1GN'].replace('nan', 0, inplace=True)

for index, row in data.iterrows():
    montant = row[colonnes_a_traiter[1]]
    frequence = row[colonnes_a_traiter[2]]
    nombre_de_fois = row[colonnes_a_traiter[3]]

    if frequence in frequence_mapping:
        Frais_livre_fornuture_Totaux = montant * frequence_mapping[frequence] *
    else:
        Frais_livre_fornuture_Totaux = montant * 1 * nombre_de_fois

    data.at[index, 'Frais_livre_fornuture_Totaux'] = Frais_livre_fornuture_Totau
colonnes_a_supprimer = [col for col in colonnes_a_traiter[1:]]
data.drop(columns=colonnes_a_supprimer, inplace=True)
```

In [14]:
```
min_Frais_livre_fornuture_Totaux = data['Frais_livre_fornuture_Totaux'].min()
max_Frais_livre_fornuture_Totaux = data['Frais_livre_fornuture_Totaux'].max()
print("Valeur minimale de Frais_Totaux :", min_Frais_livre_fornuture_Totaux)
print("Valeur maximale de Frais_Totaux :", max_Frais_livre_fornuture_Totaux)
```

```
Valeur minimale de Frais_Totaux : 0.0
Valeur maximale de Frais_Totaux : 240000.0
```

3. *Data Visualisation*

3.1 Distribution des Catégories par Wilaya

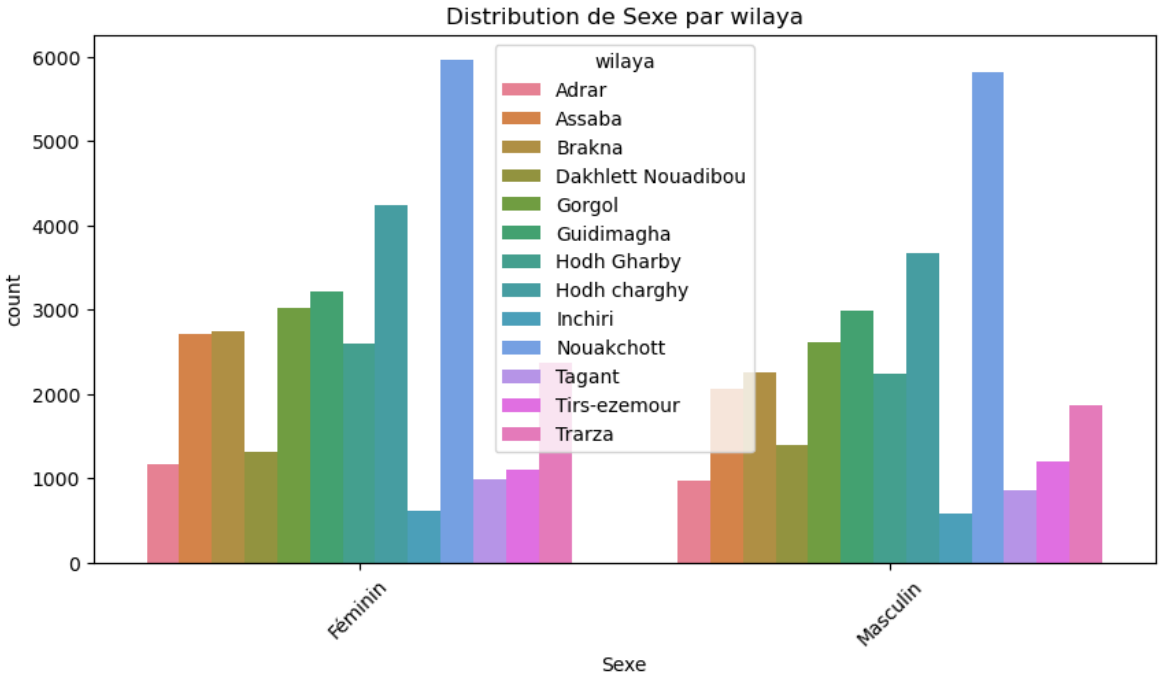
Nous avons utilisé la bibliothèque Seaborn pour créer des graphiques de comptage montrant la distribution des catégories pour les variables suivantes :

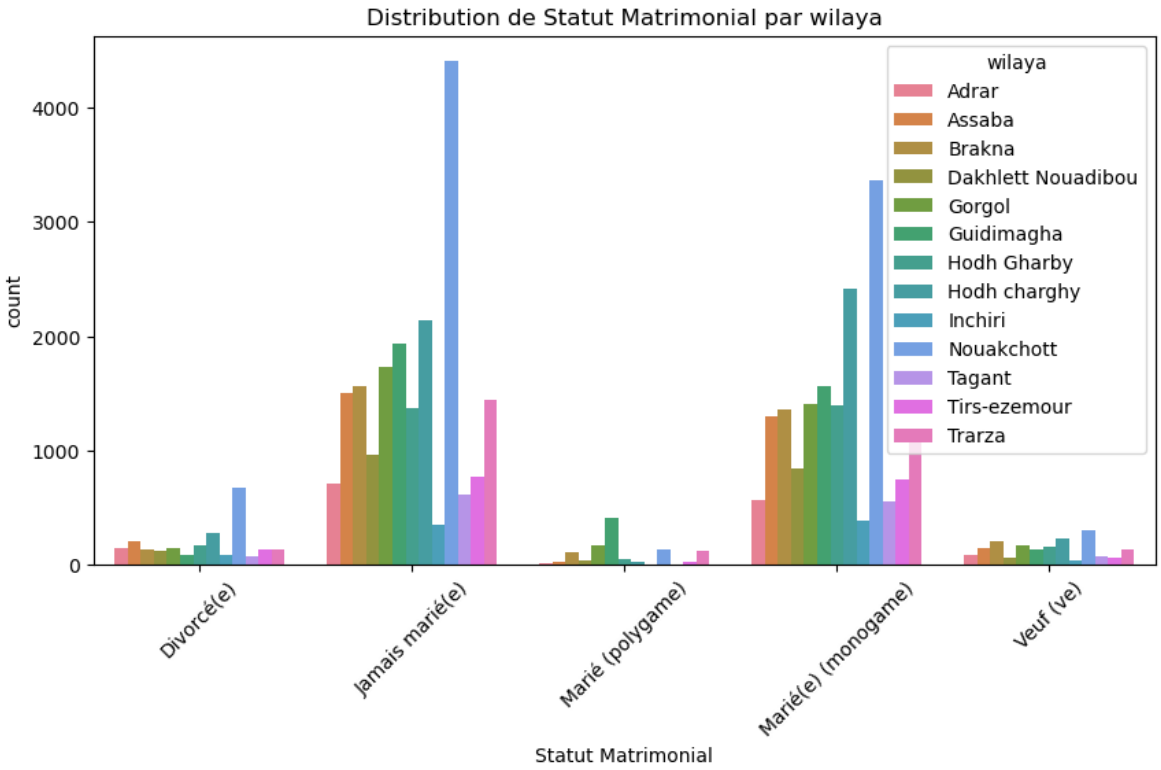
- Sexe
- Statut Matrimonial
- Sait Lire ou Écrire
- Est Allé à l'École
- Niveau Scolaire

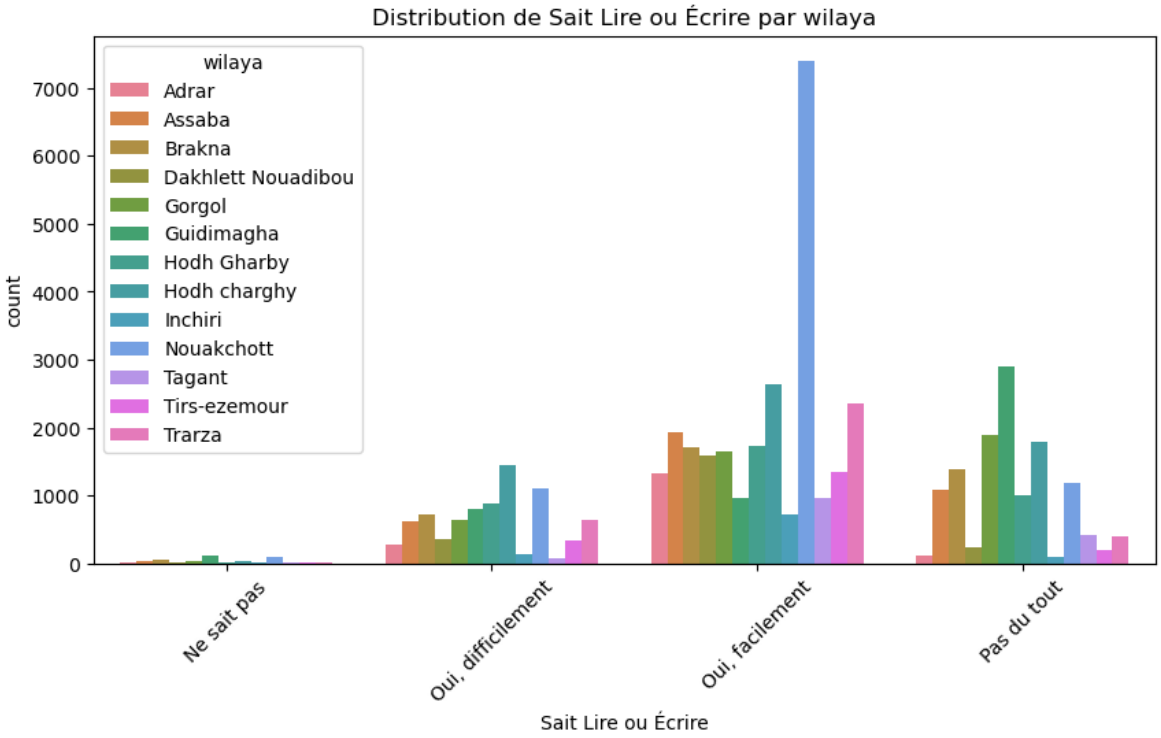
Chaque graphique est coloré par wilaya, offrant ainsi une représentation visuelle de la distribution des catégories dans chaque région. Nous avons également créé des graphiques interactifs avec Bokeh pour explorer davantage la distribution des catégories par wilaya. Ces graphiques interactifs permettent une analyse plus approfondie en affichant la distribution des catégories sous forme de barres empilées. Ces visualisations visuelles fournissent un aperçu rapide des tendances éducatives dans différentes régions, mettant en évidence les variations significatives entre les wilayas. comme vous pouvez zoomer sur ces graphiques pour une visualisation de meilleure qualité. .

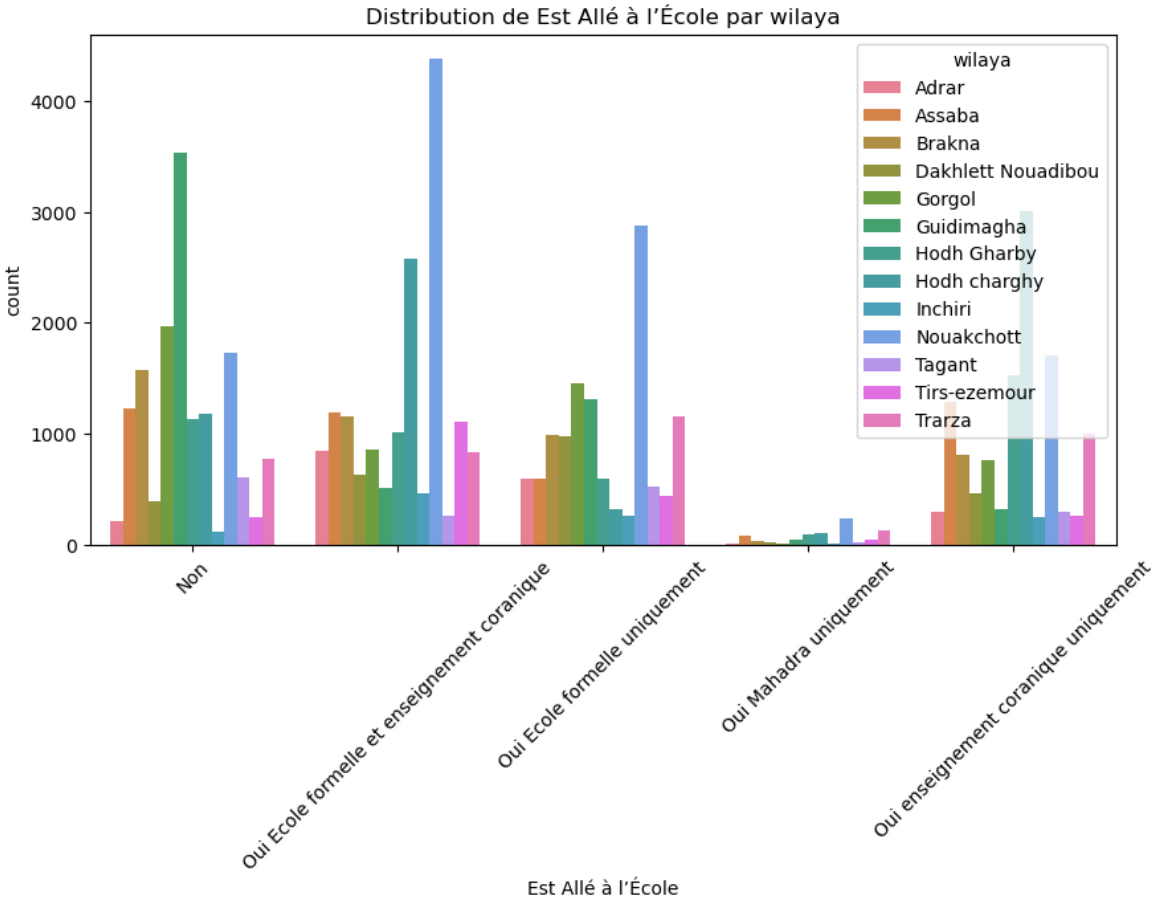
In [15]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

variables_a_conserver = [
    'wilaya', 'moughataa', 'commune', 'milieu',
```

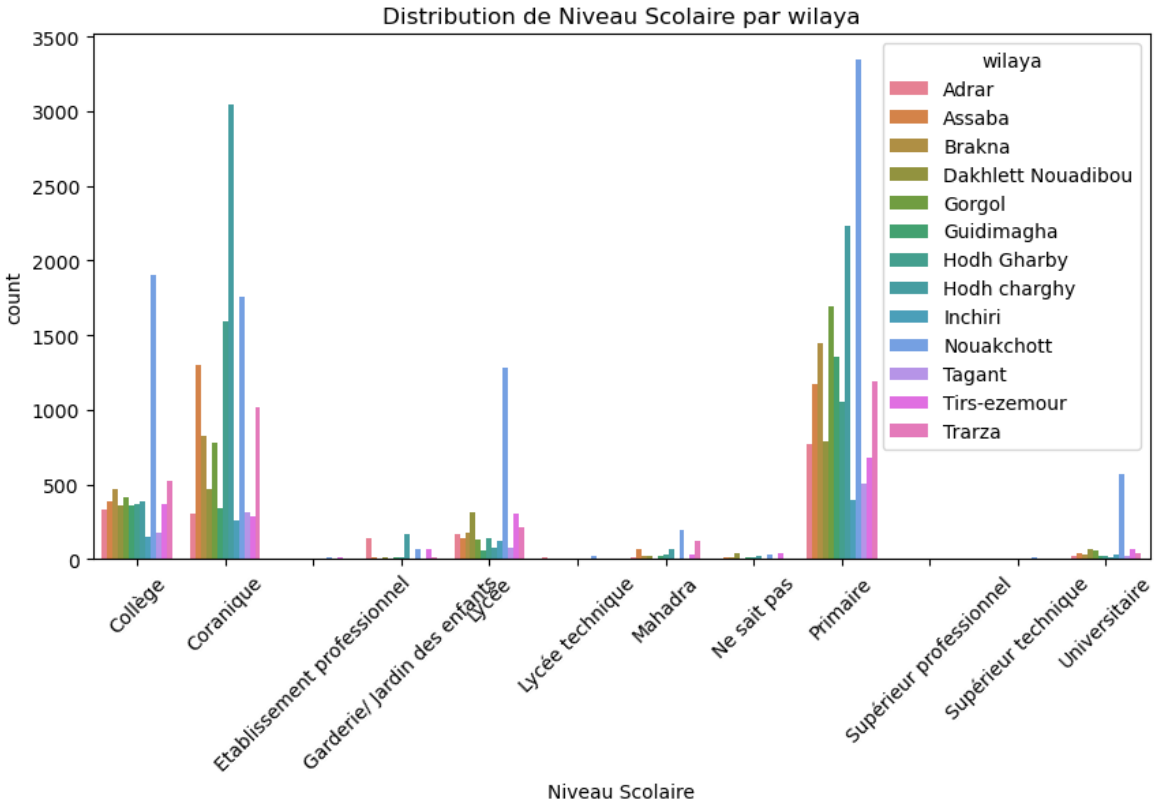
```
    'BN', 'B1', 'B2', 'B4', 'B5', 'B6', 'B7', 'B7A1', 'B7B1',
    'C1', 'C1AA', 'C1AB', 'C1AC', 'C2', 'C2A', 'C3', 'C4N', 'C4C', 'C4A',
    'C1D', 'C1EM', 'C1EF', 'C1EN', 'C1FM', 'C1FF', 'C1FN', 'C1G', 'C1GM',
    'C1GF', 'C1GN', 'C1IM', 'C1IF', 'C1IN', 'C1J', 'C10', 'C10_A', 'C13', 'C14',
    'C141', 'C15'
]

data = df[variables_a_conserver].copy()

for col in variables_a_conserver:
    data[col] = data[col].astype('category')

noms_descriptifs = {
    'B2': 'Sexe',
    'B5': 'Statut Matrimonial',
    'C1': 'Sait Lire ou Écrire',
    'C2': 'Est Allé à l’École',
    'C4N': 'Niveau Scolaire'
}
data = data.rename(columns=noms_descriptifs)

for col in ['Sexe', 'Statut Matrimonial', 'Sait Lire ou Écrire', 'Est Allé à l’É
    plt.figure(figsize=(10, 5))
    sns.countplot(x=col, hue='wilaya', data=data)
    plt.title(f'Distribution de {col} par wilaya')
    plt.xticks(rotation=45)
    plt.show()
```




Distribution de Statut Matrimonial par wilaya
wilaya
Adrar
Assaba
Brakna
Dakhlett Nouadibou
Gorgol
Guidimagha
Hodh Gharby
Hodh charghy
Inchiri
Nouakchott
Tagant
Tirs-ezemour
Trarza
count
0
1000
2000
3000
4000
Divorcé(e)
Jamais marié(e)
Marié (polygame)
Marié(e) (monogame)
Veuf (ve)
Statut Matrimonial


Distribution de Sait Lire ou Écrire par wilaya
wilaya
Adrar
Assaba
Brakna
Dakhlett Nouadibou
Gorgol
Guidimagha
Hodh Gharby
Hodh charghy
Inchiri
Nouakchott
Tagant
Tirs-ezemour
Trarza
count
0
1000
2000
3000
4000
5000
6000
7000
Ne sait pas
Oui, difficilement
Oui, facilement
Pas du tout
Sait Lire ou Écrire





interpretation

Distribution par Sexe :

La Nouakchott présente un pic élevé pour les femmes et les hommes, indiquant une proportion relativement égale des deux sexes. Inchiri a une représentation moins importante en comparaison.

Distribution par Statut Matrimonial :

Nouakchott a le pic le plus élevé dans tous les statuts matrimoniaux, suggérant une diversité importante de situations maritales. Tiris-Zemour a un pic moins élevé pour le divorce. Inchiri a un pic significatif pour les personnes jamais mariées. Adrar présente un pic moins élevé pour la polygamie. Inchiri a un pic moins élevé pour les veufs.

Distribution par Capacité de Lire ou Écrire :

Nouakchott présente un pic élevé pour les personnes capables de lire et écrire facilement. Houdh_chargui a un pic pour les personnes ayant des difficultés à lire et écrire. Gidimagha a un pic pour les personnes incapables de lire et écrire.

Distribution par Fréquentation Scolaire :

Guidimagha a un pic élevé pour les personnes ne fréquentant pas l'école. Nouakchott présente des pics pour les personnes fréquentant l'école, le Coran, uniquement l'école, et uniquement le Mahadhra. Gidimagha a un pic pour les personnes enseignant uniquement le Coran.

Distribution par Niveau Scolaire :

Nouakchott domine dans tous les types de niveaux scolaires, sauf Mahadhra, où Houdh_Chrghi a une plus grande représentation.

Conclusion

Ces observations suggèrent des disparités significatives entre les wilayas pour chaque catégorie étudiée, mettant en évidence des tendances socio-éducatives et démographiques spécifiques à chaque région.

3.2_ la distribution des catégories de niveau scolaire

In [26]:
```
import plotly.express as px

fig = px.pie(data, names='Niveau Scolaire', title='Répartition du Niveau Scolair
             labels={'Niveau Scolaire': 'Catégorie'},
             hover_data=['Niveau Scolaire'],
             color_discrete_sequence=px.colors.qualitative.Set3)
```

```
fig.update_traces(textposition='outside', textinfo='percent+label')

fig.update_layout(
    margin=dict(l=20, r=20, t=20, b=20),
    # Mise à jour des marges pour centrer le graphe
    paper_bgcolor="lightgrey",
    width=800,
    height=500,
)

fig.show()
```

interpretation

1. **Niveau Scolaire null de pourcentage 33% :** Indique une partie importante de la population qui n'a pas suivi l'éducation formelle .
2. **Primaire de pourcentage 27.3% :** Malgré le pourcentage élevé de "null", une proportion significative de personnes ayant terminé l'enseignement primaire suggère que certaines personnes ont suivi une éducation formelle de base.
3. **Coranique : 20.1% :** La présence notable d'individus ayant suivi une éducation coranique indique l'importance culturelle ou religieuse de ce type d'éducation .

4. **Collège de pourcentage 10.2% :** La proportion d'individus ayant suivi un enseignement collégial indique une partie de la population ayant atteint un niveau d'éducation intermédiaire .

5. **Lycée de pourcentage 5.28% :** Un pourcentage relativement bas d'individus ayant terminé le lycée suggère que l'accès ou la poursuite de l'enseignement supérieur pourrait être un défi pour certains.

6. **Universitaire de pourcentage 1.66% :** La faible proportion d'individus ayant atteint un niveau universitaire indique une accessibilité limitée à l'éducation supérieure.

7. **Mahadra de pourcentage 1.02% :** La présence de personnes ayant suivi une éducation dans des mahadras (instituts religieux) peut refléter l'importance de l'éducation religieuse dans certaines communautés.

8. **Jardin des Enfants de pourcentage 0.87% :** La faible proportion d'individus ayant fréquenté des jardins d'enfants peut indiquer des défis potentiels dans l'accès précoce à l'éducation.

9. **Lycée Technique de pourcentage 0.09% :** La faible proportion d'individus ayant suivi une éducation technique au lycée peut refléter des choix de carrière spécifiques ou des possibilités limitées dans ce domaine.

10. **Établissement Professionnel de pourcentage 0.064% :** La présence limitée d'individus ayant suivi une éducation dans des établissements professionnels .

11. **Supérieur Technique de pourcentage 0.03% :** La faible proportion d'individus ayant atteint un niveau supérieur technique indique des défis potentiels dans l'accès à une éducation technique de niveau supérieur.

12. **Supérieur Professionnel de pourcentage 0.0296% :** Une proportion limitée d'individus ayant atteint un niveau supérieur professionnel peut indiquer une accessibilité réduite à ce type d'éducation.

conclusion : Un pourcentage élevé des gens qui n'ont pas de nivau , L'importance de l'éducation coranique et les faibles taux d'accès à l'éducation supérieure .

4 _ Analyses approfondies

4 _1 Analyse des Relations entre Diplôme le plus élevé, État de vie du père et État de vie de la mère

Dans cette analyse, nous explorons la relation entre le Diplôme le plus élevé (variable `C4A`) et l'État de vie du père (variable `B7A1`), ainsi que l'État de vie de la mère (variable `B7B1`). Les étapes suivantes ont été effectuées :

Relation entre Diplôme le plus élevé et État de vie du père (B7A1)

- Création d'une table de contingence pour examiner la distribution des diplômes en fonction de l'état de vie du père.
- Utilisation d'une heatmap pour visualiser la fréquence des combinaisons diplôme/état de vie du père.
- Annotations ajoutées pour une lisibilité accrue.

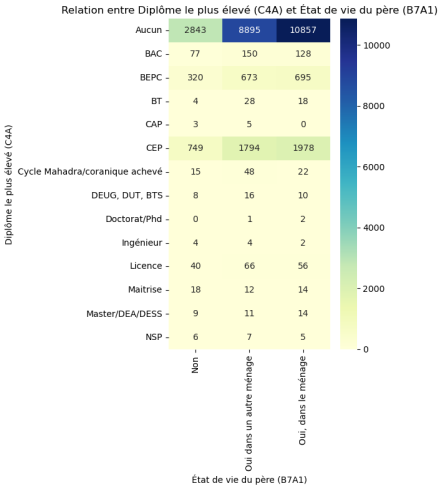
Cette visualisation peut révéler des tendances et corrélations potentielles, comme l'effet de la présence d'un père vivant sur la réussite éducative.

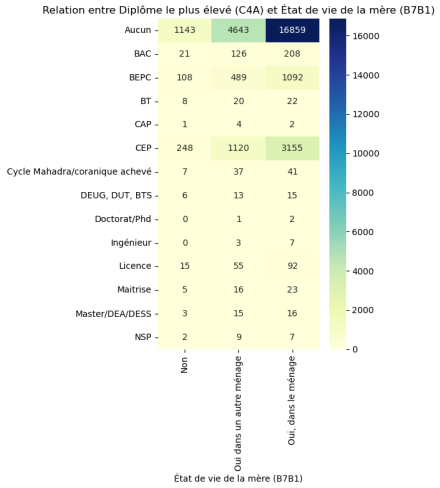
Relation entre Diplôme le plus élevé et État de vie de la mère (B7B1)

- Création d'une table de contingence pour examiner la distribution des diplômes en fonction de l'état de vie de la mère.
- Utilisation d'une heatmap pour visualiser les fréquences des différentes combinaisons diplôme/état de vie de la mère.

Ce graphique fournit des aperçus sur la manière dont le statut de vie de la mère pourrait être corrélé avec la réussite éducative. Il aide à comprendre l'impact potentiel de la présence ou de l'absence de la mère sur les résultats éducatifs.

In [17]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
table_contingence_A1 = pd.crosstab(data['C4A'], data['B7A1'].replace('Ne sait pa
table_contingence_B1 = pd.crosstab(data['C4A'], data['B7B1'].replace('Ne sait pa
plt.figure(figsize=(18, 8))
plt.subplot(1, 3, 1)
sns.heatmap(table_contingence_A1, annot=True, cmap="YlGnBu", cbar=True, fmt='g')
plt.title("Relation entre Diplôme le plus élevé (C4A) et État de vie du père (B7
plt.xlabel("État de vie du père (B7A1)")
plt.ylabel("Diplôme le plus élevé (C4A)")

plt.subplot(1, 3, 2)
plt.axis('off')

plt.subplot(1, 3, 3)
sns.heatmap(table_contingence_B1, annot=True, cmap="YlGnBu", cbar=True, fmt='g')
plt.title("Relation entre Diplôme le plus élevé (C4A) et État de vie de la mère
plt.xlabel("État de vie de la mère (B7B1)")
plt.ylabel("")

plt.tight_layout()
plt.show()
```





Interprétation des Tableaux de Contingence

Relation entre Diplôme le plus élevé (C4A) et État de vie du Père (B7A1)

Le premier diagraleur à gauche met en évidence une corrélation positive entre le diplôme le plus élevé (C4A) et l'état de vie du père (B7A1). Voici quelques observations :

- Les personnes ayant des diplômes plus élevés, tels que le BAC, BEPC et CEP, ont une tendance à avoir un père vivant.
- En particulier, les individus ayant un père vivant sont plus représentés parmi ceux ayant un diplôme élevé.

On peut également noter que les personnes ayant des pères vivants et dans le même ménage ont plus fréquemment des diplômes élevés, par rapport à celles ayant des pères vivants mais dans un autre ménage.

Relation entre Diplôme le plus élevé (C4A) et État de vie de la Mère (B7B1)

Le deuxième de chaleur à droite illustre une corrélation positive entre le diplôme le plus élevé (C4A) et l'état de vie de la mère (quelques observationss points clés :

- Les individus avec des diplômes plus élevés, notamment le B (presque verifie par tous) C, BEPC et CEP, ont tendance à avoir des mères vivantes.
- De manière générale, les personnes ayant des mères vivantes et dans le même ménage présentent plus fréquemment des diplômes élevés, par rapport à celles ayant des mères vivantes mais dans un autre ménage.

Ces observations suggèrent une association positive entre le niveau d'éducation et la présence, ainsi que l'état de vie, des parents dans le ménage. Plus spécifiquement, les individus ayant des diplômes élevés ont tendance à avoir des parents vivants, et cette tendance est renforcée lorsque les parents résident danons si nécessaire.

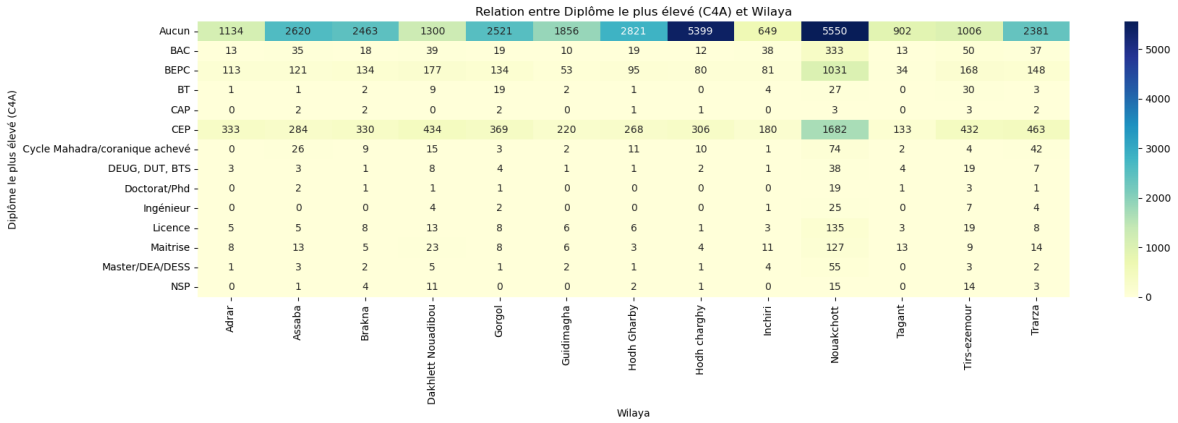
4_2 Analyse de la Relation entre Diplôme le plus élevé et Wilaya

L'analyse de la distribution des diplômes par wilaya revêt une importance stratégique

Identification des Disparités Éducatives

Permet de repérer les disparités éducatives entre les régions, mettant en lumière les zones avec une concentration plus élevée de personnes ayant atteint un certain niveau d'éducation.

In [18]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

table_contingence_wilaya = pd.crosstab(data['C4A'], data['wilaya'].replace('Ne s
plt.figure(figsize=(18, 6))

plt.subplot(1, 1, 1)
sns.heatmap(table_contingence_wilaya, annot=True, cmap="YlGnBu", cbar=True, fmt=
plt.title("Relation entre Diplôme le plus élevé (C4A) et Wilaya")
plt.xlabel("Wilaya")
plt.ylabel("Diplôme le plus élevé (C4A)")

plt.tight_layout()
plt.show()
```



Interprétation du Tableau de Contingence : Diplôme le plus élevé (C4A) et Wilaya

Le tableau de contingence ci-dessus présente la distribution des diplômes les plus élevés (C4A) selon les différentes wilayas.

1. **Aucun diplôme (Aucun):**
    - La wilaya avec le plus grand nombre d'individus sans diplôme est Hodh Gharby, suivie de Nouakchott.
    - Les wilayas avec les nombres les plus faibles d'individus sans diplôme sont Trarza et Inchiri.

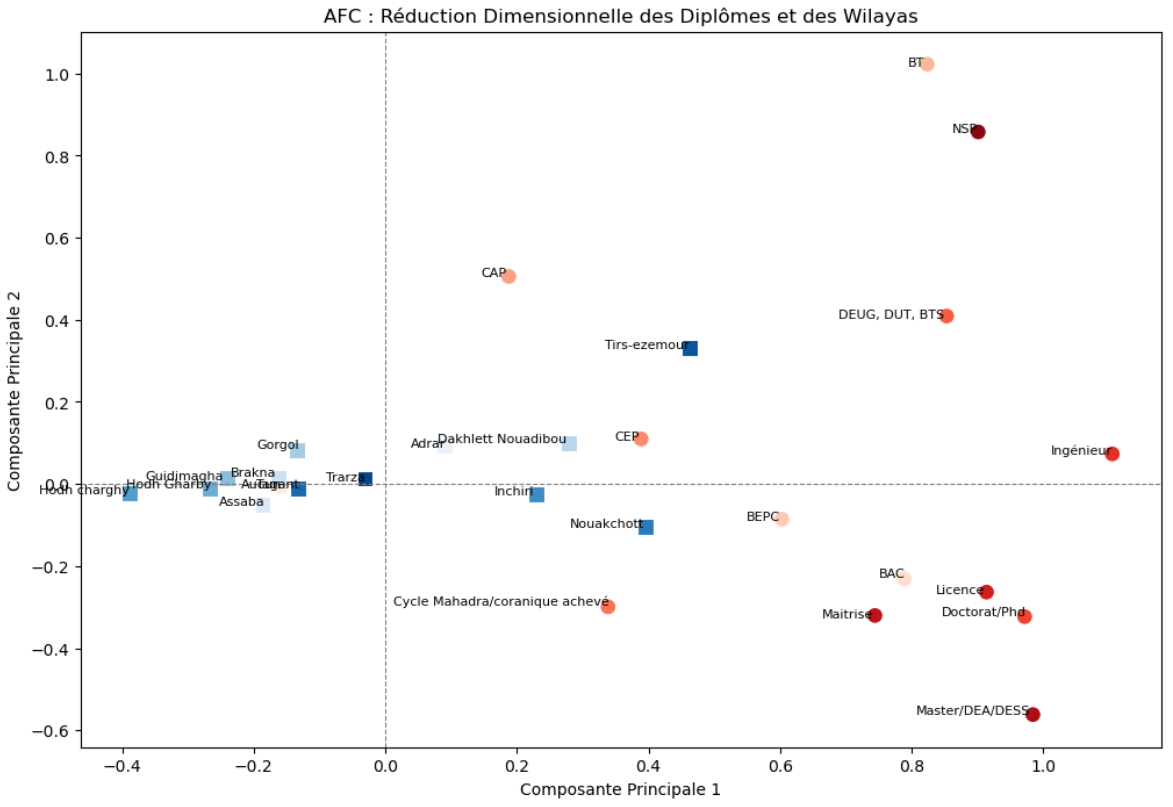
2. **Baccalauréat (BAC):**
    - La majorité des individus avec un BAC se trouvent à Nouakchott.
    - Certaines wilayas ont très peu d'individus avec un BAC.
3. **BEP, CAP, CEP:**
    - Les diplômes de BEP, CAP et CEP sont plus répartis à travers les différentes wilayas.
    - Nouakchott a une concentration plus élevée de ces diplômes par rapport aux autres wilayas.
4. **Cycle Mahadra/coranique achevé:**
    - La plupart des individus avec un cycle Mahadra/coranique achevé se trouvent à Nouakchott.
5. **Enseignement supérieur (DEUG, DUT, BTS, Doctorat, Ingénieur, Licence, Maitrise, Master/DEA/DESS):**
    - La plupart des individus avec des diplômes d'enseignement supérieur se trouvent à Nouakchott.
    - Certains diplômes, tels que DEUG, DUT, BTS, Licence et Maitrise, montrent une distribution plus large sur l'ensemble des wilayas.

Analyse Factorielle des Correspondances (AFC)

On a créé une Analyse Factorielle des Correspondances (AFC) pour étudier la relation entre les diplômes (Diplôme le plus élevé - C4A) et les wilayas Le résultat est une visualisation qui illustre la relation entre les diplômes et les wilayas dans un espace bidimensionnel. On a utilisé des couleurs différentes pour distinguer les diplômes et les wilayas dans le graphique.

In [19]:
```
import prince
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import numpy as np
table_contingence_wilaya = pd.crosstab(data['C4A'], data['wilaya'].replace('Ne s

table_contingence_wilaya.index = table_contingence_wilaya.index.astype(str)
afc = prince.CA(n_components=2)
afc.fit(table_contingence_wilaya)
coordinates_rows = afc.row_coordinates(table_contingence_wilaya)
coordinates_columns = afc.column_coordinates(table_contingence_wilaya)
fig, ax = plt.subplots(figsize=(12, 8))
scatter_rows = sns.scatterplot(
    x=coordinates_rows.iloc[:, 0],
    y=coordinates_rows.iloc[:, 1],
    hue=table_contingence_wilaya.index,
    s=100,
    marker='o',
    palette='Reds',
```

```
    ax=ax,
    legend=False
)

scatter_columns = sns.scatterplot(
    x=coordinates_columns.iloc[:, 0],
    y=coordinates_columns.iloc[:, 1],
    hue=table_contingence_wilaya.columns,
    s=100,
    marker='s',
    palette='Blues',
    ax=ax,
    legend=False
)

plt.axhline(0, color='gray', linestyle='--', linewidth=0.8)
plt.axvline(0, color='gray', linestyle='--', linewidth=0.8)
for i, txt in enumerate(table_contingence_wilaya.index):
    margin = 0.01
    x_adjusted = coordinates_rows.iloc[i, 0] + (np.random.rand() - 0.5) * margin
    y_adjusted = coordinates_rows.iloc[i, 1] + (np.random.rand() - 0.5) * margin
    scatter_rows.annotate(txt, (x_adjusted, y_adjusted), fontsize=8, ha='right',
for i, txt in enumerate(table_contingence_wilaya.columns):
    margin = 0.01
    x_adjusted = coordinates_columns.iloc[i, 0] + (np.random.rand() - 0.5) * mar
    y_adjusted = coordinates_columns.iloc[i, 1] + (np.random.rand() - 0.5) * mar
    scatter_columns.annotate(txt, (x_adjusted, y_adjusted), fontsize=8, ha='righ

plt.title("AFC : Réduction Dimensionnelle des Diplômes et des Wilayas")
plt.xlabel("Composante Principale 1")
plt.ylabel("Composante Principale 2")

plt.show()
```



interpretation

Interprétation de l'Analyse Factorielle des Correspondances (AFC)

L'Analyse Factorielle des Correspondances (AFC) a été réalisée pour étudier la relation entre les diplômes les plus élevés (C4A) et les différentes wilayas en Mauritanie. Voici une interprétation des résultats :

Composantes Principales

- **Composante Principale 1 (CP1) :** Cette composante oppose les wilayas avec une forte proportion d'individus sans diplôme (Hodh Gharby, Nouakchott) aux wilayas avec une faible proportion d'individus sans diplôme (Trarza, Inchiri).
- **Composante Principale 2 (CP2) :** Cette composante oppose les wilayas avec une concentration élevée de diplômes d'enseignement supérieur (Nouakchott) aux wilayas avec une concentration plus équilibrée de divers diplômes (DEUG, DUT, BTS, Licence, Maitrise).

Observations Wilaya par Wilaya

Aucun Diplôme (Aucun)

- Hodh Gharby et Nouakchott ont une forte proportion d'individus sans diplôme.
- Trarza et Inchiri ont une faible proportion d'individus sans diplôme.

Baccalauréat (BAC)

- Nouakchott concentre la majorité des individus avec un BAC.
- Certaines wilayas ont une faible proportion d'individus avec un BAC.

BEP, CAP, CEP

- Ces diplômes sont plus largement répartis dans toutes les wilayas, avec une concentration plus élevée à Nouakchott.

Cycle Mahadra/Coranique Achevé

- La plupart des individus avec un cycle Mahadra/Coranique achevé se trouvent à Nouakchott.

Enseignement Supérieur

- Nouakchott abrite la plupart des individus avec des diplômes d'enseignement supérieur, tandis que d'autres diplômes (DEUG, DUT, BTS, Licence, Maitrise) sont répartis plus équitablement sur l'ensemble des wilayas.

5_ Models creation

Modèle de Prédiction d'Illettrisme

Description du Modèle

Le modèle de prédiction d'illettrisme est basé sur une régression logistique. Il utilise des caractéristiques telles que le sexe, la contribution aux dépenses, la nationalité, l'état des parents (père vivant, mère vivante), la localisation géographique (wilaya, moughataa, commune) et l'âge pour prédire le statut d'illettrisme d'un individu.

In [20]:
```
data.rename(columns={
    'B2': 'Sexe',
    'B4': 'Age',
    'B6': 'Contribution aux dépenses',
    'B7': 'Nationalité',
    'B7A1': 'Père vivant',
    'B7B1': 'Mère vivante'
}, inplace=True)

selected_columns = ['Sait Lire ou Écrire', 'Sexe', 'Contribution aux dépenses',

df = data[selected_columns]
df = df.dropna()
print(df.shape)
```

```
(32114, 10)
```

In [21]:
```
df['Illettrisme'] = df['Sait Lire ou Écrire'].map({
    'Oui, facilement': 0,
    'Oui, difficilement': 0,
    'Pas du tout': 1,
    'Ne sait pas': 1,
    np.nan: 1
})

selected_columns = ['Sait Lire ou Écrire', 'Sexe', 'Contribution aux dépenses',

df = df[selected_columns + ['Illettrisme']].dropna()
print(df.shape)
```

```
(32114, 11)
```

Le modèle a été évalué sur un ensemble de test, fournissant les résultats suivants :

- **Précision du Modèle :** (proportion d'individus correctement classés)
- **Rapport de Classification :** Un rapport détaillé présentant la précision, le rappel et la f1-score pour chaque classe.

In [22]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
import pandas as pd

df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
categorical_columns = ['Sexe', 'Contribution aux dépenses', 'Nationalité', 'Père
numerical_columns = ['Age']
all_columns = categorical_columns + numerical_columns + ['Illettrisme']
selected_columns = list(dict.fromkeys(all_columns))
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_columns),
        ('num', 'passthrough', numerical_columns)
    ]
)

logistic_model = LogisticRegression()

pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('classifier', logistic_model)
])

data_for_model = df[selected_columns].dropna()
X = data_for_model.drop('Illettrisme', axis=1)
y = data_for_model['Illettrisme']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
pipeline.fit(X_train, y_train)

predictions = pipeline.predict(X_test)

accuracy = accuracy_score(y_test, predictions)
classification_report_result = classification_report(y_test, predictions)

print(f"Précision du modèle : {accuracy:.2f}")
print("\nRapport de classification :\n", classification_report_result)
```

```
Précision du modèle : 0.83

Rapport de classification :
              precision    recall  f1-score   support

           0       0.86      0.94      0.89      5003
           1       0.66      0.45      0.54      1420

    accuracy                           0.83      6423
   macro avg       0.76      0.69      0.72      6423
weighted avg       0.81      0.83      0.82      6423
```

```
C:\Users\Dell\AppData\Roaming\jupyterlab-desktop\jlab_server\lib\site-packages\sk
learn\linear_model\_logistic.py:460: ConvergenceWarning:

lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
```

Évaluation du Modèle

La précision du modèle de prédiction d'illettrisme est de 0.83, ce qui indique que le modèle classe correctement 83 % des individus dans l'ensemble de test. Le rapport de classification fournit des détails sur la performance du modèle pour chaque classe.

Résultats du Rapport de Classification :

- **Classe 0 (Non-Illettrisme) :**
  - Précision : 0.86 (86 % des prédictions de la classe 0 sont correctes)
  - Recall : 0.94 (94 % des individus réellement de la classe 0 sont correctement identifiés)
  - F1-score : 0.89 (mesure équilibrée entre la précision et le recall)
- **Classe 1 (Illettrisme) :**
  - Précision : 0.66 (66 % des prédictions de la classe 1 sont correctes)
  - Recall : 0.45 (45 % des individus réellement de la classe 1 sont correctement identifiés)
  - F1-score : 0.54 (mesure équilibrée entre la précision et le recall)

Performance Globale du Modèle :

- **Accuracy (Exactitude) :** 0.83 (83 % des prédictions sont correctes sur l'ensemble de test)
- **Macro AVG :** 0.76 (moyenne des performances des deux classes, utilisée lorsque les classes sont déséquilibrées)
- **Weighted AVG :** 0.81 (moyenne pondérée en fonction du nombre d'instances par classe, utile pour des ensembles de données déséquilibrés)

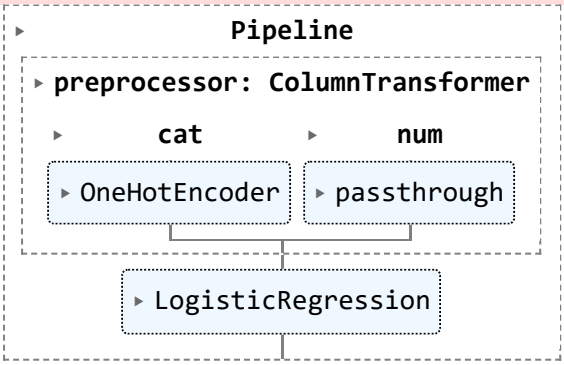
Ces résultats suggèrent que le modèle est généralement performant, mais montre une meilleure prédiction pour la classe 0 (Non-Illettrisme) par rapport à la classe 1 (Illettrisme).

In [23]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
```

```
from sklearn.pipeline import Pipeline
df['Age'] = pd.to_numeric(df['Age'], errors='coerce')

categorical_columns = ['Sexe', 'Contribution aux dépenses', 'Nationalité', 'Père
numerical_columns = ['Age']
all_columns = categorical_columns + numerical_columns + ['Illettrisme']
selected_columns = list(dict.fromkeys(all_columns))

preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_columns),
        ('num', 'passthrough', numerical_columns)
    ])

logistic_model = LogisticRegression()

pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('classifier', logistic_model)
])

data_for_model = df[selected_columns].dropna()
X = data_for_model.drop('Illettrisme', axis=1)
y = data_for_model['Illettrisme']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
pipeline.fit(X_train, y_train)
```

```
C:\Users\Dell\AppData\Roaming\jupyterlab-desktop\jlab_server\lib\site-packages\sk
learn\linear_model\_logistic.py:460: ConvergenceWarning:

lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
```

Out[23]:



Utilisation du Modèle pour la Prédiction

'utilisation du modèle de régression logistique pour effectuer une prédiction sur de nouvelles données. Dans cet exemple, un DataFrame `input_data` est créé avec des informations factices sur un individu, puis ces données sont prétraitées à l'aide du

préprocesseur du pipeline. Enfin, le modèle entraîné est utilisé pour prédire le statut d'illettrisme de l'individu.

1. **Création d'un DataFrame avec des informations sur un individu :**
    - Les caractéristiques telles que le sexe, la contribution aux dépenses, la nationalité, l'état des parents, la localisation géographique et l'âge sont fournies pour un nouvel individu.
2. **Prédiction du statut d'illettrisme avec le modèle entraîné :**
    - Le modèle de régression logistique est utilisé pour prédire si l'individu est illettré ou non en se basant sur les caractéristiques fournies.
3. **Affichage du résultat de la prédiction :**
    - Le résultat de la prédiction est affiché pour indiquer si l'individu est prédit comme illettré (1) ou non illettré (0).

In [24]:
```
input_data = pd.DataFrame({
    'Sexe': ['Masculin'],
    'Contribution aux dépenses': ['Non'],
    'Nationalité': ['Mauritanien'],
    'Père vivant': ['Oui'],
    'Mère vivante': ['Oui'],
    'wilaya': ['Nouakchott'],
    'moughataa': ['Arafat'],
    'commune': ['Arafat'],
    'Age': [23]
})
input_features = pipeline.named_steps['preprocessor'].transform(input_data)
prediction = pipeline.named_steps['classifier'].predict(input_features)
print(f"Prediction Result: {prediction[0]}")
```

```
Prediction Result: 0
```

Commentaire sur le Résultat de la Prédiction

Suite à l'utilisation du modèle pour prédire le statut d'illettrisme d'un individu, le résultat obtenu est affiché ci-dessous.

- **Résultat de la Prédiction :** 0

Ce résultat indique que l'individu est prédit comme non illettré. En termes simples, le modèle estime que l'individu ne présente pas de caractéristiques indiquant un statut d'illettrisme.

In [ ]:

In [ ]:

In [ ]:

🚀 Alhamdoulillah, notre projet a abouti de manière fructueuse !