EEC23-22

April 4, 2025

ANALYSE DES DONNEES ACTIVITE EMPLOI ET CHOMAGE - ENQUETE EMPLOI EN CONTINU Source : https://www.data.gouv.fr/fr/datasets/activite-emploi-et-chomage-enquete-emploi-en-continu

Ce projet s'inscrit dans une démarche personnelle d'analyse de données, visant à valoriser mes compétences dans ce domaine tout en perfectionnant ma pratique. À travers cette étude approfondie, l'objectif est d'explorer l'activité, l'emploi et le chômage en France à partir des données collectées lors de l'enquête emploi en continu (EEC) de 2023 et 2022. Cette approche me permet non seulement de mieux appréhender les dynamiques socio-économiques actuelles, mais aussi de cibler les facteurs déterminants qui influencent l'employabilité et le marché du travail.

L'analyse englobe des variables clés telles que le statut d'activité, le niveau de diplôme, l'âge et le sexe, tout en s'efforçant de décrypter les relations complexes qui les relient. En combinant des outils statistiques, des visualisations avancées et des modèles économétriques, ce travail ambitionne de produire des résultats à la fois clairs, exploitables et pertinents.

Dans une démarche méthodologique rigoureuse, ce projet ne se limite pas à enrichir la compréhension des données ; il aspire également à fournir des pistes de réflexion constructives pour alimenter des initiatives futures en matière d'emploi et de formation professionnelle

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from simpledbf import Dbf5
from scipy.stats import norm
import statsmodels.api as sm
```

[2]:
```
#Importation des fichiers de données
eec23 = pd.read_csv('FD_csv_EEC23/FD_csv_EEC23.csv', delimiter=';')
eec22_dbf = Dbf5('FD_EEC_2022_dbase/FD_EEC_2022.dbf')
eec22 = eec22_dbf.to_dataframe()
```

[3]:
```
#Importation des fichiers descriptifs des modalités des variables
varmod_eec23 = pd.read_csv('FD_csv_EEC23/Varmod_EEC_2023.csv', delimiter=';')
varmod_eec22_dbf = Dbf5('FD_EEC_2022_dbase/varmod_EEC_2022.dbf')
varmod_eec22 = varmod_eec22_dbf.to_dataframe()
```

ANALYSE DES DONNEES ACTIVITE EMPLOI ET CHOMAGE - EEC DE 2023
La description des codes des variables se trouve dans le fichier "varmod_EEC_2023". C'est également le cas pour la base de données de 2022.

[4]:
```
#Vérification des données manquantes
missing_values = eec23.isnull().sum()
print(missing_values[missing_values > 0])
```

```
AAC           173184
ACL_EMPLOI     40045
AISCO2        166930
ANCCHOM       334803
ANCEMPL4      181694
               …
STPLC         181694
TEMP          305550
TPPRED        181694
TRAREF         46282
TXTPPRED      318697
Length: 64, dtype: int64
```

La base de données comporte plusieurs valeurs manquantes. Il est crucial de comprendre la signification de ces valeurs ainsi que les variables affectées avant de les utiliser. Je m'assurerai de le faire progressivement. La base de données contient 83 variables et 348 624 observations.

ANALYSE EXPLOIRATOIRE DES DONNEES

ANALYSE DES LA VARIABLE STATUT D'ACTIVITE [ACTEU] Champ : personnes de 15 ans ou plus (15<=AGE)

[ ]:
```
''' def plot_all_variables(df, varmod_df):
        for column in df.columns:
                if column in varmod_df['COD_VAR'].values:
                        varmod = varmod_df[varmod_df['COD_VAR'] == column]
                        varmod_dict = dict(zip(varmod['COD_MOD'],␣
↪varmod['LIB_MOD']))
                        df[column] = df[column].map(varmod_dict)

                value_counts = df[column].value_counts()
                if value_counts.empty:
                        continue

                total = value_counts.sum()
                pourcentages = value_counts / total * 100

                plt.figure(figsize=(10, 5))
                ax = pourcentages.plot(kind='barh', color='b')
                for i in ax.patches:
```

```
                        ax.text(i.get_x() + i.get_width() / 2, i.get_height() +␣
↪0.5, f"{i.get_height():.2f}%", ha='center', va='bottom')

                plt.title(f"Distribution of {column}")
                ax.set_xticklabels(ax.get_xticklabels(), rotation=0,␣
↪ha='center')
                plt.show()

plot_all_variables(eec23, varmod_eec23) '''
```

Cette fonction permet de visualiser toutes les variables de la base de données en associant les données à leurs libellés contenus dans le fichier varmod. Lors de son exécution, elle donne un premier aperçu des variables susceptibles d'être intéressantes à analyser. Toutefois, les graphiques générés ne peuvent pas être personnalisés individuellement, ce qui peut parfois aboutir à des résultats peu convaincants ou incorrects. Par conséquent, je vais procéder à une analyse individuelle de chaque variable.

L'objectif est de traiter, nettoyer et explorer la base de données afin de mieux la comprendre et d'identifier les différentes variables ainsi que leurs significations.

[6]:
```
#Création d'un dictionnaire pour les modalités de la variable ACTEU
acteu_varmod = varmod_eec23[varmod_eec23['COD_VAR'] == 'ACTEU']
acteu_varmod = acteu_varmod.drop(columns=['TYPE_VAR', 'LONG_VAR'])
acteu_dict = dict(zip(acteu_varmod['COD_MOD'], acteu_varmod['LIB_MOD']))
print(acteu_dict)
```

```
{'1': 'Emploi', '2': 'Chômage', '3': 'Non Inactivité'}
```

Le COD_MOD "3" est actuellement libellé 'Non Inactivité', ce qui est incorrect et ne correspond pas à la description des libellés fournis par l'INSEE dans le fichier PDF (EEC 2023 _ Dictionnaire des codes _ Fichier detail_2024_07_17). De plus, selon la classification du BIT, la population active, qui comprend toutes les personnes âgées de 15 ans ou plus, est divisée en personnes en emploi, personnes au chômage et personnes inactives. Ainsi, il est probable que l'intitulé 'Non Inactivité' soit incorrect et résulte peut-être d'une erreur de saisie.

[7]:
```
# Correction de la modalités de la variable ACTEU
acteu_dict['3'] = 'Inactivité'
print(acteu_dict)
```

```
{'1': 'Emploi', '2': 'Chômage', '3': 'Inactivité'}
```

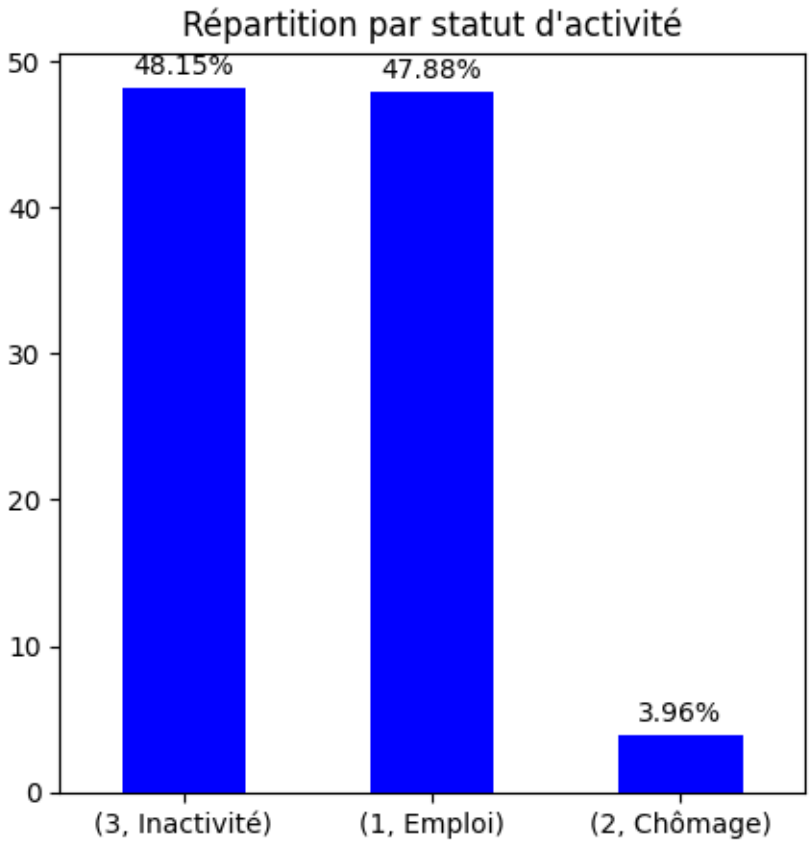
[8]:
```
eec23_acteu = pd.DataFrame()
eec23_acteu['code_mod'] = eec23['ACTEU'].astype(str)
eec23_acteu['mod_lib'] = eec23_acteu['code_mod'].map(acteu_dict)
```

[9]:
```
# Recherche de valeurs manquantes
missing_values = eec23_acteu.isnull().sum()
print(missing_values)
```

```
code_mod    0
```

```
mod_lib     0
dtype: int64
```

[10]:
```
# Répartition des individus par statut d'activité
total = eec23_acteu.value_counts().sum()
pourcentages = eec23_acteu.value_counts() / total * 100

plt.figure(figsize=(5, 5))

ax = pourcentages.plot(kind='bar', color='b')
for i in ax.patches:
    ax.text(i.get_x() + i.get_width() / 2, i.get_height() + 0.5, f"{i.
 ↪get_height():.2f}%", ha='center', va='bottom')

plt.xlabel("")
plt.ylabel("")

plt.title("Répartition par statut d'activité")
ax.set_xticklabels(ax.get_xticklabels(), rotation=0, ha='center')
plt.show()
```



On observe qu'il y a presque autant de personnes occupées(ou en emploi) qu'il y a de personnes inactives. Le taux de chômage est quand à lui relativement faible et d'environ 4%

ANALYSE DE LA VARIABLE EMPLOI ACTUEL OU DERNIER EMPLOI [ACL_EMPLOI]** Champ : personnes en emploi ou ayant travaillé ACTEU = 1(emploi) ou ACTEU =2(chômage),3(Inactif) et AAC(Exercice d'une activité antérieure pour les personnes sans emploi)=1(Oui)

Extraction des codes et libellés des données classe d'emploi.

[11]:
```
acl_emploi_varmod = varmod_eec23[varmod_eec23['COD_VAR'] == 'ACL_EMPLOI']
acl_emploi_varmod = acl_emploi_varmod.drop(columns=['TYPE_VAR', 'LONG_VAR'])
acl_emploi_dict = dict(zip(acl_emploi_varmod['COD_MOD'],␣
 ↪acl_emploi_varmod['LIB_MOD']))
acl_emploi_dict['0'] = acl_emploi_dict.pop(np.nan)
```

Création d'un dictionnaire permettant de faire correspondre les codes aux descriptions adéquates.

Préparation des données pour visualisation

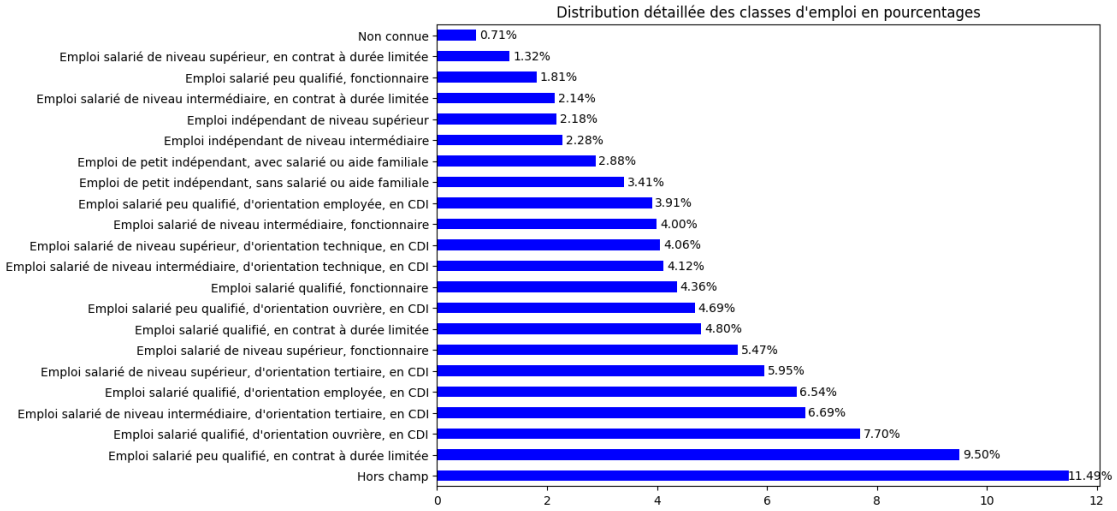
[12]:
```
# Création d'un dataframe vide
eec23_acl_emploi = pd.DataFrame()

# Création d'une colonne 'code_mod' contenant les données codées des modalités
eec23_acl_emploi['code_mod'] = eec23['ACL_EMPLOI'].fillna(0).astype(str)
eec23_acl_emploi['mod_lib'] = eec23_acl_emploi['code_mod'].map(acl_emploi_dict)
```

[13]:
```
# Recherche de valeurs manquantes
missing_values = eec23_acl_emploi.isnull().sum()
print(missing_values)
```

```
code_mod    0
mod_lib     0
dtype: int64
```

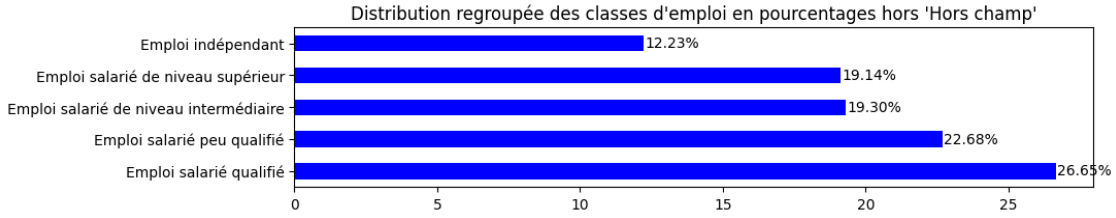
[14]:
```
# Graphique en barres
total = eec23_acl_emploi['mod_lib'].value_counts().sum()
pourcentages = eec23_acl_emploi['mod_lib'].value_counts() / total * 100

plt.figure(figsize=(10, 7))

ax = pourcentages.plot(kind='barh', color='b')
for i in ax.patches:
    ax.text(i.get_width() + 0.4, i.get_y() + i.get_height() / 2, f"{i.
 ↪get_width():.2f}%", ha='center', va='center')

plt.xlabel("")
plt.ylabel("")
```

```
plt.title("Distribution détaillée des classes d'emploi en pourcentages")
plt.show()
```



[15]:
```
# Extraction du premier caractère de codes des modalités afin de les regrouper␣
 ↪en classes
eec23_acl_emploi['code_mod_category'] = eec23_acl_emploi['code_mod'].
 ↪apply(lambda x: x[0] if pd.notnull(x) else x)

# Définition d'un dictionnaire de classes
dictionnaire = {
    'A': 'Emploi salarié de niveau supérieur',
    'B': 'Emploi salarié de niveau intermédiaire',
    'C': 'Emploi salarié qualifié',
    'D': 'Emploi salarié peu qualifié',
    'I': 'Emploi indépendant'
}

# Correspondance
eec23_acl_emploi['classes'] = eec23_acl_emploi['code_mod_category'].
 ↪map(dictionnaire)

# Calcul des pourcentages
total = eec23_acl_emploi['classes'].value_counts().sum()
pourcentages = eec23_acl_emploi['classes'].value_counts() / total * 100

plt.figure(figsize=(10, 2))

ax = pourcentages.plot(kind='barh', color='b')
for i in ax.patches:
```

```
    ax.text(i.get_width() + 1, i.get_y() + i.get_height() / 2, f"{i.get_width():
 ↪.2f}%", ha='center', va='center')


plt.xlabel("")
plt.ylabel("")


plt.title("Distribution regroupée des classes d'emploi en pourcentages hors␣
 ↪'Hors champ'")
plt.show()
```



Dans un premier temps, on observe qu'environ 12 % des personnes enquêtées ne sont pas en emploi et n'ont pas exercé d'activité antérieure.

Par ailleurs, les emplois salariés peu qualifiés en CDD constituent la classe d'emploi la plus recrutée, représentant environ 10 % du total. Cela inclut probablement de la main-d'œuvre ponctuelle (intérim) et possiblement des emplois secondaires. Une analyse plus approfondie, notamment en examinant le nombre d'heures travaillées, pourrait être intéressante. Les emplois salariés peu qualifiés en CDD pourraient représenter un grand nombre de contrats, mais un faible nombre d'heures travaillées.

En regroupant les données, on constate que les emplois salariés qualifiés dominent le marché du travail, représentant environ 27 % des emplois, suivis par les emplois salariés peu qualifiés à hauteur de 23 %. Les emplois salariés de niveau intermédiaire et les emplois salariés de niveau supérieur sont relativement au même niveau, soit 19 % chacun.

Quant aux emplois indépendants, ils recrutent beaucoup moins et représentent environ 12 % du total.

Analyse de la variable "Exercice d'une activité professionnelle régulière antérieure, pour les inactifs, chômeurs et personnes ayant une activité temporaire ou d'appoint autre qu'un emploi informel"[AAC] Champ : personnes sans emploi ACTEU=2 (chômage), 3(Inactif)

[16]:
```
#Création d'un dictionnaire pour les modalités de la variable ACTEU
aac_varmod = varmod_eec23[varmod_eec23['COD_VAR'] == 'AAC']
aac_varmod = aac_varmod.drop(columns=['TYPE_VAR', 'LONG_VAR'])
aac_dict = dict(zip(aac_varmod['COD_MOD'], aac_varmod['LIB_MOD']))
```

```
aac_dict['0'] = aac_dict.pop(np.nan)
print(aac_dict)
```

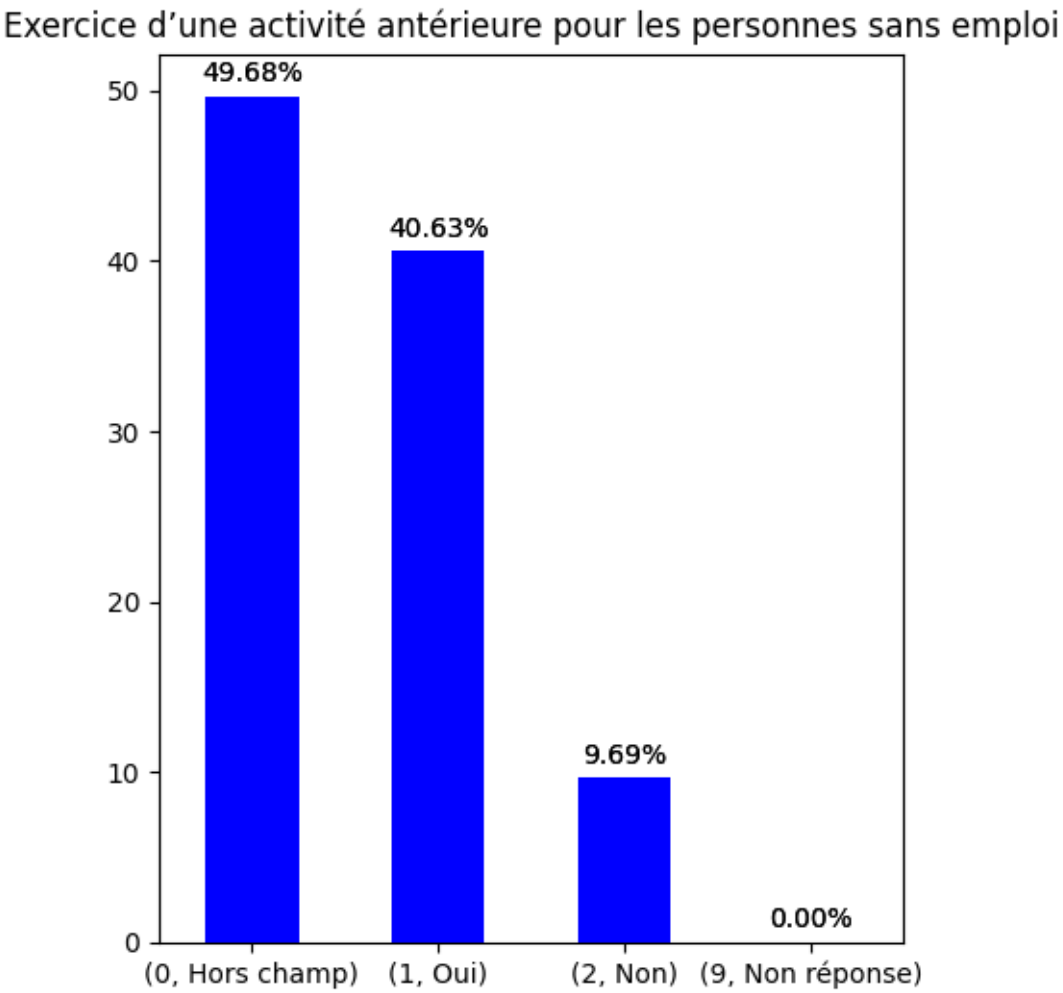
```
{'1': 'Oui', '2': 'Non', '9': 'Non réponse', '0': 'Hors champ'}
```

[17]:
```
eec23_aac = pd.DataFrame()
eec23_aac['code_mod'] = eec23['AAC'].fillna(0).astype(int).astype(str)
eec23_aac['mod_lib'] = eec23_aac['code_mod'].map(aac_dict)
```

[18]:
```
# Recherche de valeurs manquantes
missing_values = eec23_aac.isnull().sum()
print(missing_values)
```

```
code_mod    0
mod_lib     0
dtype: int64
```

[19]:
```
total = eec23_aac.value_counts().sum()
pourcentages = eec23_aac.value_counts() / total * 100

plt.figure(figsize=(5, 6))
ax = pourcentages.plot(kind='bar', color='b')

ax = pourcentages.plot(kind='bar', color='b')
for i in ax.patches:
    ax.text(i.get_x() + i.get_width() / 2, i.get_height() + 0.5, f"{i.
 ↪get_height():.2f}%", ha='center', va='bottom')

plt.xlabel("")
plt.ylabel("")

plt.title("Exercice d'une activité antérieure pour les personnes sans emploi")
ax.set_xticklabels(ax.get_xticklabels(), rotation=0, ha='center')
plt.show()
```



Il est notable que la majorité des personnes sans emploi, soit 41 % des enquêtés, sont en activité ou ont déjà exercé une activité professionnelle. Le faible taux de non-réponse, proche de zéro, est également rassurant et souligne la qualité des données collectées. Cependant, il est inquiétant de constater que 50 %, soit près de la moitié des personnes enquêtées, ne se trouvent pas dans le champ de la variable pourtant assez large. Cela représente-t-il des emplois informels ? Ou est-ce lié à la semaine de référence ?

Il serait intéressant de se pencher sur le profil des personnes sans emploi n'ayant jamais exercé d'activité professionnelle ainsi que sur la durée de leur chômage afin de mieux comprendre les dynamiques en jeu. De plus, des analyses plus approfondies seraient nécessaires pour comprendre pourquoi autant de personnes sortent du champ de la variable.

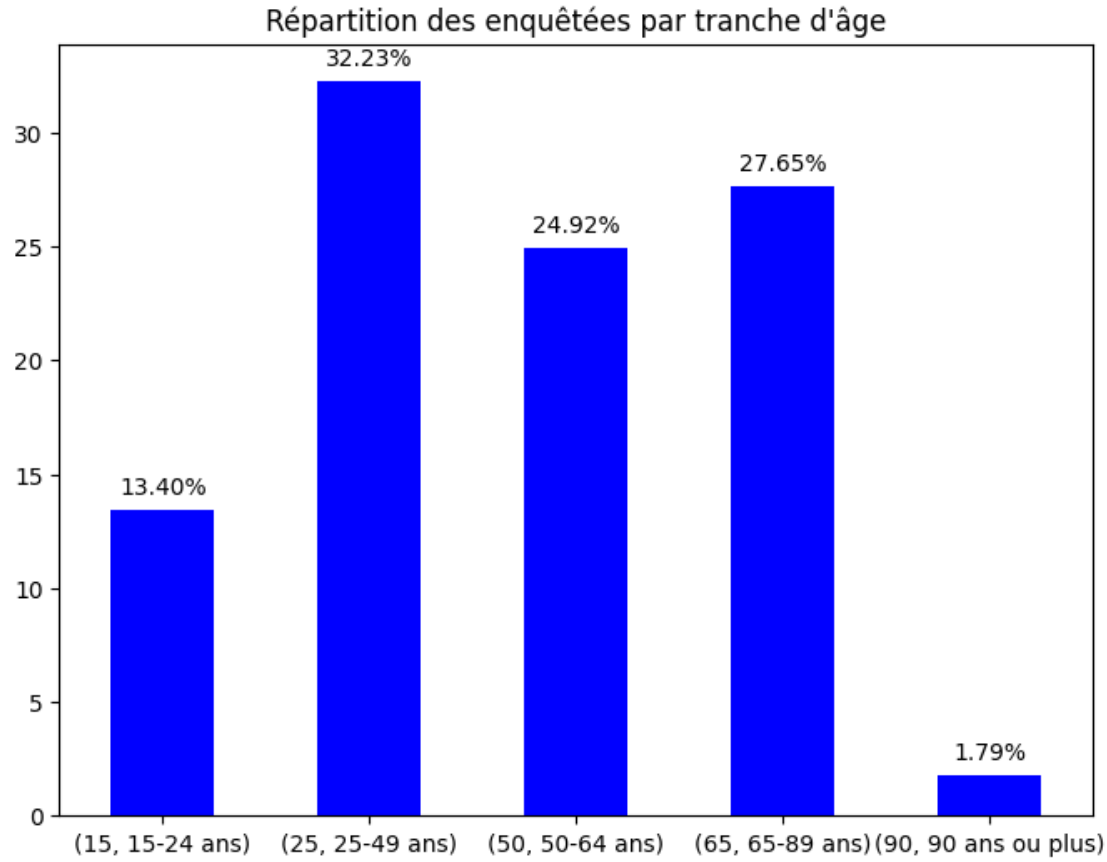
ANALYSE DE LA VARIABLE "AGE EN 6 TRANCHES"[AGE6] Champ : ensemble des personnes

[20]:
```
#Création d'un dictionnaire pour les modalités de la variable ACTEU
age_varmod = varmod_eec23[varmod_eec23['COD_VAR'] == 'AGE6']
age_varmod = age_varmod.drop(columns=['TYPE_VAR', 'LONG_VAR'])
age_dict = dict(zip(age_varmod['COD_MOD'], age_varmod['LIB_MOD']))
print(age_dict)
```

```
{'00': '14 ans ou moins', '15': '15-24 ans', '25': '25-49 ans', '50': '50-64
ans', '65': '65-89 ans', '90': '90 ans ou plus'}
```

[21]:
```
eec23_age = pd.DataFrame()
eec23_age['code_mod'] = eec23['AGE6'].fillna(0).astype(int).astype(str)
eec23_age['mod_lib'] = eec23_age['code_mod'].map(age_dict)
```

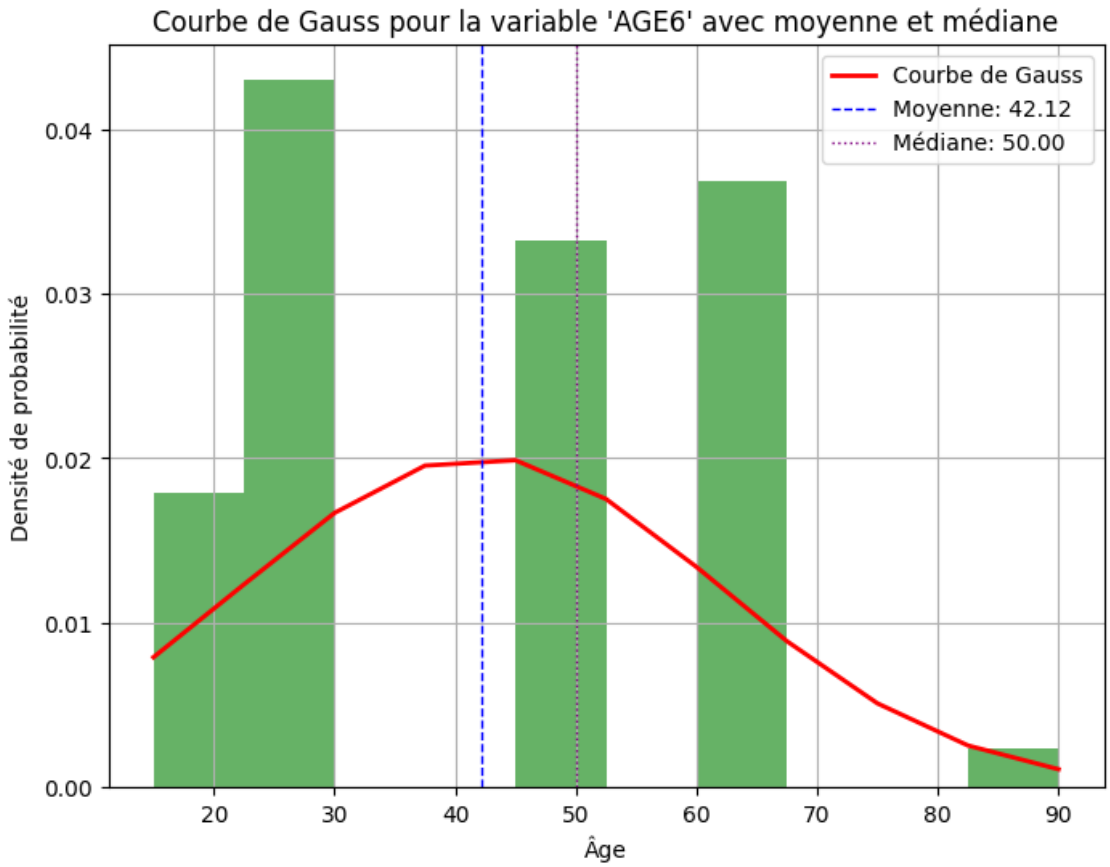
[22]:
```
#Répartition des personnes enquêtées par tranche d'âge

total = eec23_age.value_counts().sum()
pourcentages = eec23_age.value_counts(sort=False) / total * 100

plt.figure(figsize=(8, 6))

ax = pourcentages.plot(kind='bar', color='b')
for i in ax.patches:
    ax.text(i.get_x() + i.get_width() / 2, i.get_height() + 0.5, f"{i.
 ↪get_height():.2f}%", ha='center', va='bottom')

plt.xlabel("")
plt.ylabel("")

plt.title("Répartition des enquêtées par tranche d'âge")
ax.set_xticklabels(ax.get_xticklabels(), rotation=0, ha='center')
plt.show()
```



Ce graphique, bien que complexe à interpréter, nous fournit néanmoins quelques informations cruciales sur la distribution des âges de l'échantillon. On remarque que la classe d'âge la plus représentée regroupe les personnes âgées de 25 à 49 ans. Environ 30 % des personnes enquêtées ont plus de 65 ans. L'âge moyen de la retraite en France étant de 63 ans en 2023 (https://www.retraite.com/dossier-retraite/chiffres-cles-retraite-cnav.html), ces personnes sont probablement déjà à la retraite. Il serait intéressant d'examiner plus en détail comment les données sont réellement réparties au sein de ces classes d'âge. Malheureusement, nous ne disposons pas des âges réels des personnes enquêtées dans ce jeu de données ; les âges sont regroupés et codés par classes.

[23]:
```
int_eec23_age = eec23_age['code_mod'].astype(int)

# Calcul des paramètres de la distribution normale (moyenne et écart-type)
mu, sigma = int_eec23_age.mean(), int_eec23_age.std()
median = int_eec23_age.median()

# Tracé de l'histogramme des âges
plt.figure(figsize=(8, 6))
count, bins, ignored = plt.hist(int_eec23_age, bins=10, density=True, alpha=0.
↪6, color='g')
```

```
# Calcul de la courbe de Gauss
gauss_curve = norm.pdf(bins, mu, sigma)

# Tracé de la courbe de Gauss
plt.plot(bins, gauss_curve, linewidth=2, color='r', label='Courbe de Gauss')

# Ajout des lignes pour la moyenne et la médiane
plt.axvline(mu, color='b', linestyle='dashed', linewidth=1, label=f'Moyenne:␣
 ↪{mu:.2f}')
plt.axvline(median, color='#800080', linestyle='dotted', linewidth=1,␣
 ↪label=f'Médiane: {median:.2f}')


plt.title("Courbe de Gauss pour la variable 'AGE6' avec moyenne et médiane")
plt.xlabel("Âge")
plt.ylabel("Densité de probabilité")
plt.legend()
plt.grid(True)
plt.show()
```



[24]:
```python
# Calcul des statistiques descriptives
statistiques_descriptives = {
    'moyenne': int_eec23_age.mean(),
    'médiane': int_eec23_age.median(),
    'mode': int_eec23_age.mode().values[0],
    'écart_type': int_eec23_age.std(),
    'minimum': int_eec23_age.min(),
    'maximum': int_eec23_age.max(),
    'quantiles': int_eec23_age.quantile([0.25, 0.5, 0.75]).to_dict(),
    'variance': int_eec23_age.var(),
    'skewness': int_eec23_age.skew(),
    'kurtosis': int_eec23_age.kurtosis()
}

# Affichage des statistiques descriptives
for stat, valeur in statistiques_descriptives.items():
    print(f'{stat}: {valeur}')
```

```
moyenne: 42.117998187158655
médiane: 50.0
mode: 25
écart_type: 19.858742148518523
minimum: 15
maximum: 90
quantiles: {0.25: 25.0, 0.5: 50.0, 0.75: 65.0}
variance: 394.3696397213461
skewness: 0.14244757492726018
kurtosis: -1.2562891473200504
```

Afin de pouvoir analyser plus en détail la distribution de la variable 'AGE6', j'ai réalisé un histogramme combiné à une courbe de Gauss. Ce graphique permet de bien illustrer les statistiques descriptives.

L'âge minimum est de 15 ans et l'âge maximum de 90 ans. L'âge moyen étant de 42 ans, cela donne une idée générale de l'âge central de la population étudiée. Avec un âge médian de 50 ans, il est peu probable qu'il y ait des valeurs aberrantes, compte tenu de la proximité entre la moyenne et la médiane. Cela est cohérent avec une kurtosis négative (-1,26), indiquant que la distribution est aplatie par rapport à une distribution normale, ce qui signifie moins de valeurs extrêmes.

L'écart-type de 19,86 ans montre une dispersion importante des âges autour de la moyenne, ce qui indique une grande variation dans les âges des individus. Cela peut être dû au fait que les données sont regroupées par classes, avec des intervalles de 10 à 25 ans entre chaque classe.

L'analyse des quantiles montre que 25 % des individus ont moins de 25 ans, 50 % ont moins de 50 ans (la médiane) et 75 % ont moins de 65 ans. L'échantillon a probablement été conçu en prenant en compte ces paramètres. En mettant cela en parallèle avec l'âge moyen de la retraite en France, nous constatons que l'échantillon contient environ 35 % de personnes potentiellement à la retraite, comme expliqué précédemment.

La variance élevée (394,37) est cohérente avec l'écart-type élevé, indiquant une grande diversité des âges. Cela peut être dû au regroupement des données par classes, comme mentionné plus haut.

Enfin, un skewness positif (0,14) indique que la distribution des âges est légèrement asymétrique et penchée vers la droite, ce qui signifie qu'il y a quelques valeurs élevées influençant la moyenne. Cela est tout à fait normal et cohérent avec le fait que la médiane se trouve légèrement à droite de la moyenne.

ANALYSE DE LA VARIABLE "Diplôme le plus élevé obtenu"[DIP7] Champ : personnes de 15-89 ans (15<=AGE<=89)

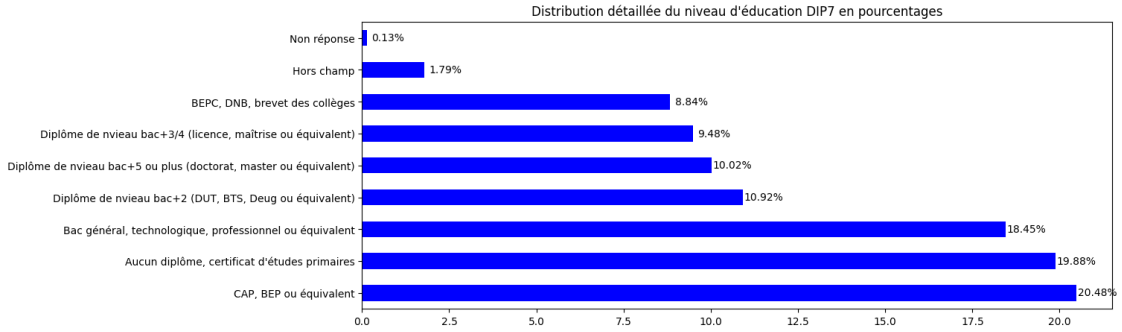
[25]:
```
dip_varmod = varmod_eec23[varmod_eec23['COD_VAR'] == 'DIP7']
dip_varmod = dip_varmod.drop(columns=['TYPE_VAR', 'LONG_VAR'])
dip_dict = dict(zip(dip_varmod['COD_MOD'], dip_varmod['LIB_MOD']))
dip_dict['0'] = dip_dict.pop(np.nan)
```

[26]:
```
eec23_dip = pd.DataFrame()
eec23_dip['code_mod'] = eec23['DIP7'].fillna(0).astype(int).astype(str)
eec23_dip['mod_lib'] = eec23_dip['code_mod'].map(dip_dict)
```

[27]:
```
# Recherche de valeurs manquantes
missing_values = eec23_dip.isnull().sum()
print(missing_values)
```

```
code_mod    0
mod_lib     0
dtype: int64
```

[28]:
```
# Graphique en barres
total = eec23_dip['mod_lib'].value_counts().sum()
pourcentages = eec23_dip['mod_lib'].value_counts() / total * 100

plt.figure(figsize=(13, 5))

ax = pourcentages.plot(kind='barh', color='b')
for i in ax.patches:
    ax.text(i.get_width() + 0.6, i.get_y() + i.get_height() / 2, f"{i.
 ↪get_width():.2f}%", ha='center', va='center')

plt.xlabel("")
plt.ylabel("")

plt.title("Distribution détaillée du niveau d'éducation DIP7 en pourcentages")
plt.show()
```



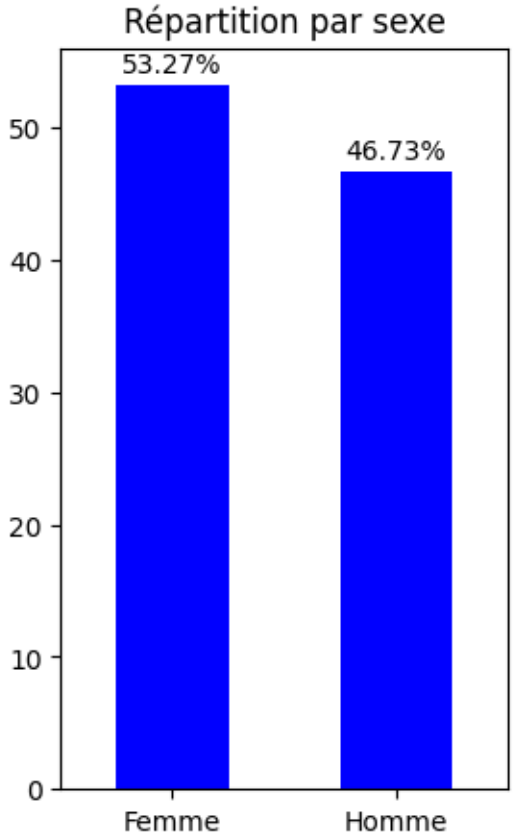
On observe qu'environ 60 % des personnes enquêtées ont un diplôme de niveau Bac ou moins. Les détenteurs de diplômes de niveau Bac+2, Bac+3, et Bac+5 représentent chacun environ 10 % de l'échantillon. Cela reflète un niveau d'études général relativement bas. Ce qui est assez préoccupant. Un niveau d'études général relativement bas peut avoir plusieurs conséquences économiques et sociales. Un faible niveau d'éducation peut limiter les opportunités d'emploi et conduire à une sous-utilisation des compétences. Cela peut également entraîner une plus grande vulnérabilité au chômage, surtout dans les secteurs où les emplois non qualifiés sont en déclin en raison de l'automatisation et de la numérisation. Néamoins il conviendrait d'utiliser la variable de pondération EXTRIAN avant de tirer des conclusions.

Il serait intéressant d'analyser les catégories d'emploi occupées en fonction du niveau d'études. Une telle analyse pourrait révéler des disparités importantes dans les opportunités d'emploi et les conditions de travail, ainsi que les secteurs où des efforts de formation et de développement des compétences pourraient être nécessaires.

ANALYSE DE LA VARIABLE SEXE[SEXE] Champ : ensemble des personnes

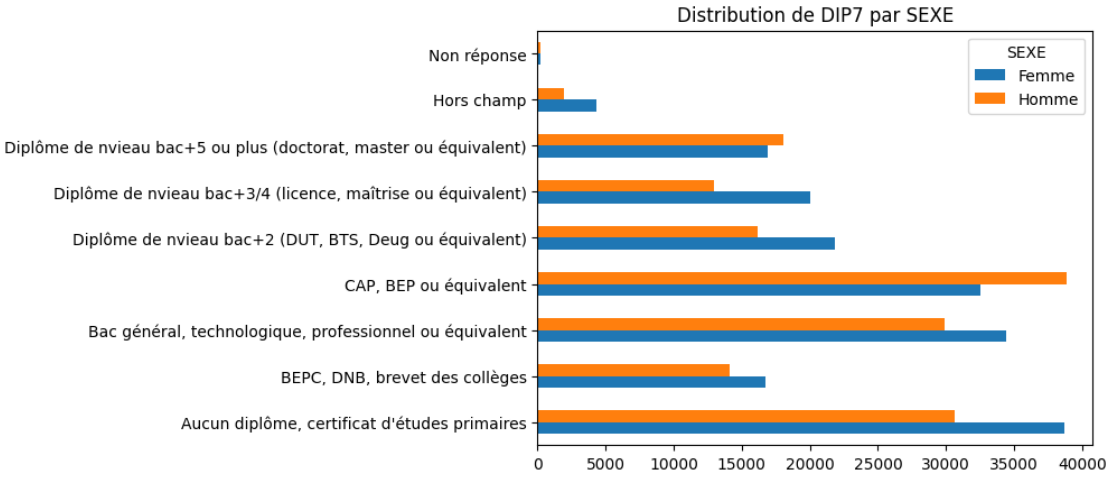
[29]:
```
sexe_dict = {
    '1': 'Homme',
    '2': 'Femme'
}

eec23_sexe = pd.DataFrame()
eec23_sexe['code_mod'] = eec23['SEXE'].astype(str)
eec23_sexe['mod_lib'] = eec23_sexe['code_mod'].map(sexe_dict)


# Graphique en barres
total = eec23_sexe['mod_lib'].value_counts().sum()
pourcentages = eec23_sexe['mod_lib'].value_counts() / total * 100

plt.figure(figsize=(3, 5))
ax = pourcentages.plot(kind='bar', color='b')
for i in ax.patches:
```

```
    ax.text(i.get_x() + i.get_width() / 2, i.get_height() + 0.5, f"{i.
 ↪get_height():.2f}%", ha='center', va='bottom')

plt.xlabel("")
plt.ylabel("")

plt.title("Répartition par sexe")
ax.set_xticklabels(ax.get_xticklabels(), rotation=0, ha='center')
plt.show()
```



[30]:
```
grouped = pd.DataFrame()
#Répartition des individus par sexe et par niveau d'éducation

grouped['DIP7'] = eec23['DIP7'].fillna(0).astype(int).astype(str).map(dip_dict)
grouped['SEXE'] = eec23['SEXE'].astype(str).map(sexe_dict)
eec23_grouped = grouped.groupby(['DIP7', 'SEXE']).size().unstack().fillna(0).
 ↪astype(int)
eec23_grouped.plot(kind='barh')
```

```
plt.title('Distribution de DIP7 par SEXE')
plt.xlabel('')
plt.ylabel('')
plt.show()
```



L'échantillon contient légèrement plus de femmes que d'hommes, représentant respectivement 53,27 % et 46,73 %. Ce déséquilibre pourrait refléter la répartition réelle de la population enquêtée.

Malgré cette domination féminine dans toutes les classes de diplômes, ce qui pourrait être attribué au fait qu'il y ait plus de femmes dans l'échantillon, on observe néanmoins que les hommes sont majoritaires parmi les détenteurs de diplômes de niveau Bac+5 ainsi que les diplômes de CAP, BEP ou équivalent.

Ce phénomène pourrait s'expliquer par plusieurs facteurs socio-économiques. Par exemple, la sur-représentation des hommes parmi les titulaires de Bac+5 pourrait être liée à des choix de filières d'études ou à des inégalités d'accès à l'éducation supérieure. En ce qui concerne les diplômes de CAP, BEP ou équivalent, ces formations techniques et professionnelles sont souvent perçues comme des parcours plus masculins, influençant ainsi la répartition des genres.

Les conséquences possibles de cette situation incluent des disparités salariales et des inégalités d'opportunités professionnelles. Les femmes, malgré leur représentation importante dans l'échantillon, peuvent être désavantagées dans l'accès à certains emplois bien rémunérés ou de haut niveau en raison de ces différences de qualification. Cela peut également affecter l'équilibre des compétences dans certains secteurs économiques et limiter la diversité des perspectives au sein des professions.

Il serait intéressant d'examiner la aussi les catégories d'emploi occupées en fonction du niveau d'études et du genre pour mieux comprendre les dynamiques en jeu et identifier les domaines où des interventions pourraient être nécessaires pour favoriser l'égalité des chances.

L'idée ici est de vérifier les observations précédemment effectuées, je vais réaliser plusieurs analyses à l'aide d'outils statistiques et économétriques. La première étape consistera à construire une matrice de corrélation. Cela permettra d'identifier des relations entre les différentes variables et de

comprendre si elles concordent avec les observations initiales.

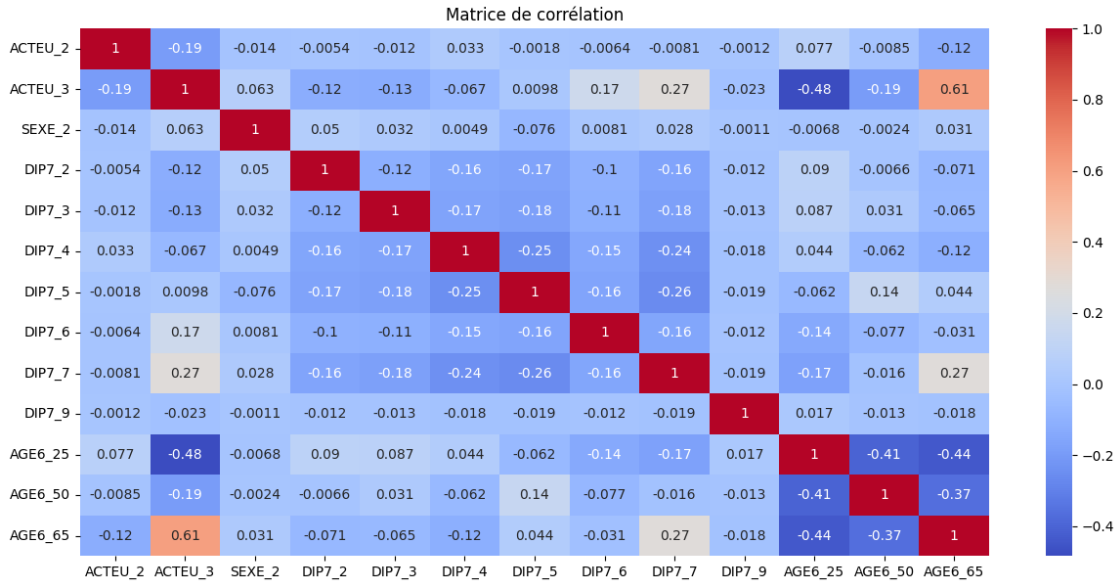
Ensuite, je vais procéder à plusieurs régressions linéaires pour analyser comment certaines variables, telles que le genre, le niveau d'études, l'âge et l'expérience, influencent l'employabilité. En d'autres termes, je cherche à déterminer dans quelle mesure ces facteurs affectent la probabilité d'être employé.

Ces analyses sont essentielles pour confirmer les hypothèses et fournir une base solide pour des recommandations de politique publique. Par exemple, si le niveau d'études a un impact significatif sur l'employabilité, cela pourrait justifier des investissements accrus dans l'éducation et la formation professionnelle. De même, si l'âge ou l'expérience montrent des corrélations fortes, des politiques visant à soutenir les jeunes travailleurs ou les travailleurs expérimentés pourraient être envisagées.

[31]:
```
reg_data = eec23[['AGE6', 'SEXE', 'DIP7', 'ACTEU']]
reg_data = reg_data.dropna()
reg_data = reg_data.astype(int)
encoded_data = pd.get_dummies(reg_data, columns=['ACTEU', 'SEXE', 'DIP7',␣
 ↪'AGE6'], drop_first=True)
encoded_data = encoded_data.astype(int)
print(encoded_data.columns)
```

```
Index(['ACTEU_2', 'ACTEU_3', 'SEXE_2', 'DIP7_2', 'DIP7_3', 'DIP7_4', 'DIP7_5',
       'DIP7_6', 'DIP7_7', 'DIP7_9', 'AGE6_25', 'AGE6_50', 'AGE6_65'],
      dtype='object')
```

[32]:
```
# Calcul de la matrice de corrélation
corr_matrix = encoded_data.select_dtypes(include=[np.number]).corr()

plt.figure(figsize=(15, 7))
# Affichage de la matrice de corrélation sous forme de heatmap avec seaborn
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm")
plt.title('Matrice de corrélation')
plt.show()
```



On observe qu'il n'y a pas de corrélation forte entre les variables. Toutefois, il y a une corrélation modérée (0,61) entre la variable ACTEU_3, qui correspond au fait d'être inactif, et la variable AGE6_65 (personnes âgées de 65 à 89 ans). Cela est logique et concorde avec l'observation initiale. En effet, environ 30 % des personnes enquêtées ont plus de 65 ans. L'âge moyen de la retraite en France étant de 63 ans en 2023, il était probable que ces personnes soient déjà à la retraite et donc inactives. Cette corrélation entre les deux variables est donc explicable.

[33]:
```
# Définir les variables explicatives (X) et la variable cible (y)
X = encoded_data[['ACTEU_3', 'AGE6_25', 'AGE6_50', 'AGE6_65', 'SEXE_2',␣
  ↪'DIP7_2', 'DIP7_3', 'DIP7_4', 'DIP7_5', 'DIP7_6', 'DIP7_7', 'DIP7_9']]
y = encoded_data['ACTEU_2']

# Ajouter une constante pour l'interception (ordonnée à l'origine)
X = sm.add_constant(X)

# Ajuster le modèle de régression linéaire
model = sm.OLS(y, X).fit()

# Résumé des résultats
print(model.summary())
```

OLS Regression Results

Dep. Variable:	ACTEU_2	R-squared:	0.053
Model:	OLS	Adj. R-squared:	0.053
Method:	Least Squares	F-statistic:	1607.
Date:	Fri, 04 Apr 2025	Prob (F-statistic):	0.00
Time:	22:14:39	Log-Likelihood:	80087.

No. Observations:	342370	AIC:	-1.601e+05
Df Residuals:	342357	BIC:	-1.600e+05
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.1062	0.002	69.701	0.000	0.103	0.109
ACTEU_3	-0.0996	0.001	-107.442	0.000	-0.101	-0.098
AGE6_25	-0.0522	0.001	-43.464	0.000	-0.055	-0.050
AGE6_50	-0.0663	0.001	-54.948	0.000	-0.069	-0.064
AGE6_65	-0.0440	0.001	-36.283	0.000	-0.046	-0.042
SEXE_2	0.0018	0.001	2.649	0.008	0.000	0.003
DIP7_2	0.0068	0.001	4.616	0.000	0.004	0.010
DIP7_3	0.0067	0.001	4.714	0.000	0.004	0.010
DIP7_4	0.0270	0.001	20.663	0.000	0.024	0.030
DIP7_5	0.0334	0.001	25.996	0.000	0.031	0.036
DIP7_6	0.0299	0.002	18.503	0.000	0.027	0.033
DIP7_7	0.0515	0.001	38.707	0.000	0.049	0.054
DIP7_9	-0.0194	0.009	-2.182	0.029	-0.037	-0.002

Omnibus:	290514.129	Durbin-Watson:	1.897
Prob(Omnibus):	0.000	Jarque-Bera (JB):	5463332.135
Skew:	4.284	Prob(JB):	0.00
Kurtosis:	20.594	Cond. No.	38.4

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Le coefficient de détermination (R^2_R-squared) indique que le modèle n'explique que 5,3% de la variance de la variable ACTEU_2 (le fait d'être au chômage). Un coefficient de détermination aussi faible indique que le modèle ne capture pas bien les variations des données et qu'il y a beaucoup de facteurs non inclus dans le modèle qui influencent la variable dépendante.

Le manque de corrélation entre les variables observé avec la matrice de corrélation s'explique.

Néanmoins, même avec un coefficient de corrélation faible, les coefficients des variables indépendantes peuvent fournir des informations sur la direction et la force des relations entre les variables.

Par exemple, les variables ACTEU_3, AGE6_25, AGE6_50, AGE6_65, et AGE6_90 ont des coefficients négatifs significatifs, ce qui signifie qu'elles ont une relation négative avec la variable dépendante ACTEU_2. Cela signifie que, toutes choses égales par ailleurs, à mesure que l'âge augmente, la probabilité ou le taux de chômage diminue.

La variable SEXE_2 a un coefficient positif significatif, indiquant une relation positive avec ACTEU_2. Cela signifie que, toutes choses égales par ailleurs, être une femme augmente la probabilité d'être au chômage par rapport à ne pas être une femme (ou être un homme). En d'autres termes, ce résultat suggère qu'il existe une corrélation positive entre le sexe féminin et le chômage. Il est

donc possible que les femmes soient victimes de discrimination à l'embauche, ce qui réduit leurs chances de trouver un emploi par rapport aux hommes. Ou que les femmes sont moins disponibles pour le travail à plein temps compte tenu des responsabilités qu'elles peuvent assumer comme la garde d'enfants par exemple. Cela peut aussi provenir de différences sectorielles.

Il est donc important de souligner que la corrélation observée ne signifie pas nécessairement une relation causale directe. D'autres facteurs non inclus dans le modèle peuvent également influencer cette relation.

Pour améliorer le modèle, il serait intéressant de : - Ajouter des variables importantes telles que le niveau d'éducation, le type d'emploi (ACL_EMPLOI), l'expérience professionnelle (AAC), la disponibilité etc… - Explorer les interactions potentielles entre les variables. - Envisager des modèles non linéaires ou des transformations de variables. - Effectuer une analyse qualitative pour fournir des informations contextuelles. - Utiliser des techniques de validation croisée pour évaluer la robustesse du modèle.

PROJET EN COURS…