



L'Évolution de l'emploi



*Chef d'œuvre - Gnei Shehara MOHAMED
Développeur/se Data P10*

Table des matières

1. Introduction du projet
 - 1.1 Présentation du projet
 - 1.2 Pourquoi
 - 1.3 Les enjeux
 - 1.4 Objectif
 - 1.5 Stack technique
 - 1.6 Gestion de projet
2. Conception et développement de la base de données
 - 2.1 Collecte des données
 - 2.1.1 Les sources des données
 - 2.1.2 Description des données
 - 2.2 Conception de la base de données
 - 2.2.1 Modèle Conceptuel de Données
 - 2.2.2 Modèle Logique de Données
 - 2.2.3 Modèle Physique de Données
 - 2.3 Traitement des données
 - 2.3.1 Préparation ou nettoyage des données
 - 2.3.2 Insertion des données
 - 2.4 Sécurisation et Optimisation de la base de données
3. Analyser et visualiser les données
 - 3.1 Analyse et visualisation (manuelle)
 - 3.2 Application web créée avec streamlit
 - 3.2.1 La démarche du développement une webapp avec streamlit
 - 3.2.2 Les techniques utilisées
 - 3.2.3 L'explication de la source code
4. Ressources
5. Les axes d'amélioration et les difficultés
6. Remerciements

1. Introduction du projet

1.1 Présentation de projet

La marché de l'emploi est important individuellement et globalement car c'est un domaine qui influe sur la vie de chacun et l'économie d'un secteur géographique (pays, région, département et etc.)

Ce projet est basé sur l'évolution du marché de l'emploi et du nombre de chômeurs à travers différents indicateurs. Afin de représenter ces évolutions, j'ai collecté les données sur des sites regroupant des données statistiques sur le marché de l'emploi (par exemple le site de l'INSEE, pôle emploi, etc.). Les indicateurs concernés sont :

- La catégorie socioprofessionnelle
- Le diplôme
- Les caractéristiques de l'emploi
- Le secteur géographique

1.2 Contexte ?

Afin de valider cette formation, j'ai effectué un stage d'un mois dans l'entreprise gouvernementale « France Stratégie » pour le projet « Le métier de demain ». Ce projet est basé sur l'évolution de l'emploi sur les dix prochaines années. En m'inspirant de ce projet j'ai décidé de récupérer les données et de les traiter dans le but de représenter ces données de manière organisée et de suivre l'évolution du marché du travail.

1.3 Les enjeux

Jour après jour la population française augmente, ainsi que les besoins en emploi. Toutefois nous ne savons pas s'il y a suffisamment d'emplois pour satisfaire cette demande. En France il y a environ 11 % de chômeurs selon les données 2020. Chaque mois, il y a 5 000 à 10 000 (en moyenne) personnes par département qui s'inscrivent à pôle emploi.

Cette étude sera utile aux demandeurs d'emploi qui souhaitent analyser le marché de l'emploi et les besoins selon le diplôme, la catégorie socioprofessionnelle ainsi que le secteur géographique.

Nous avons de nombreuses données brutes difficiles à analyser et exploiter en l'état. Toutes ces données doivent être organisées afin de visualiser l'évolution de chaque indicateur.

Après avoir organisé toutes ces données les résultats pourront représenter :

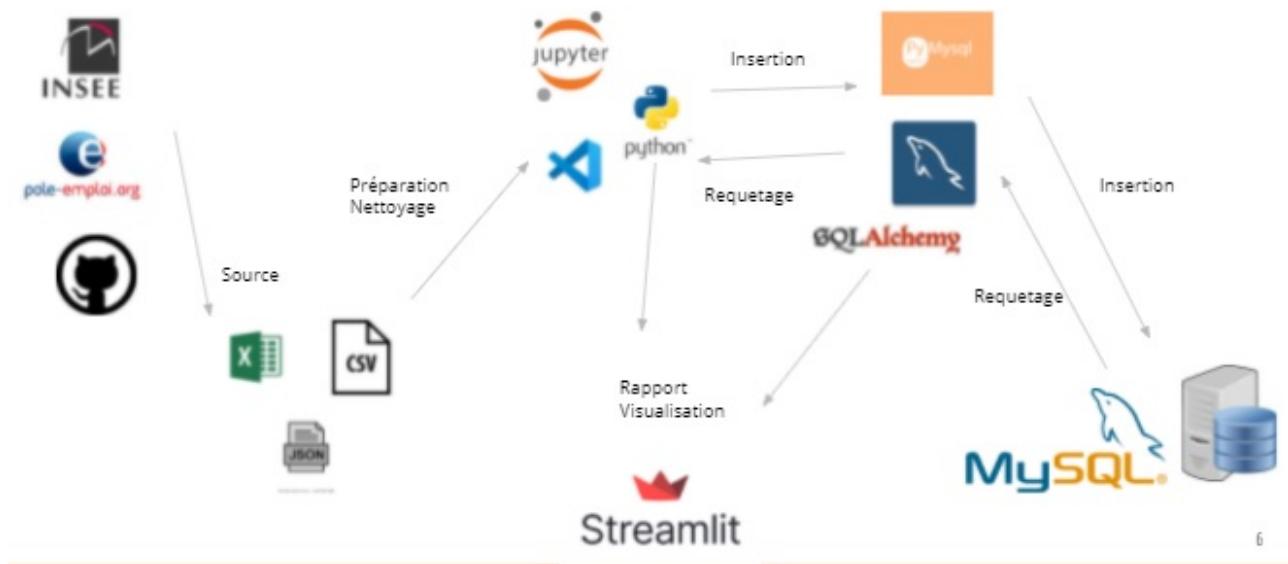
- La proportion d'emploi et de chômeurs par groupe d'âge et de sexe (pour chaque indicateur).
- Les départements ou les régions où il y a le plus de chômeurs.
- Le niveau de salaire par secteur géographique.
- Les catégories en train d'évoluer et en train de disparaître

1.4 Objectifs

Les objectifs de ce projet sont :

- Organisations des données.
- Création d'une base de données pour toutes ces données brutes.
- Présentation des données via une application web dans le but de produire des analyses.
- Veille du marché de l'emploi par indicateur.

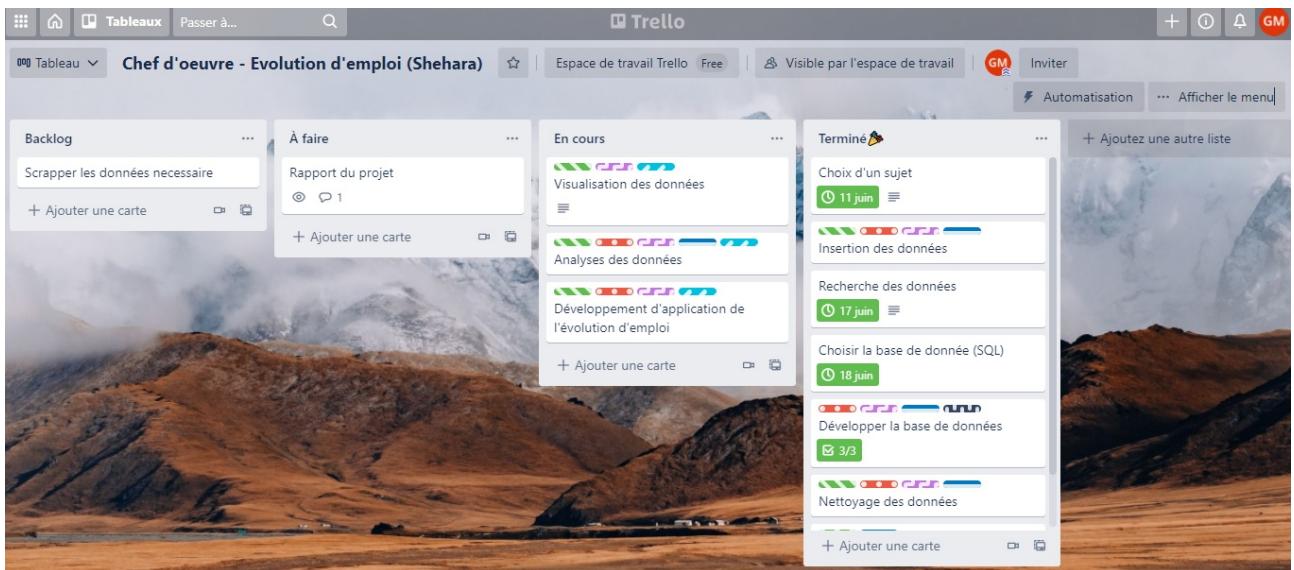
1.5 Stack Technique



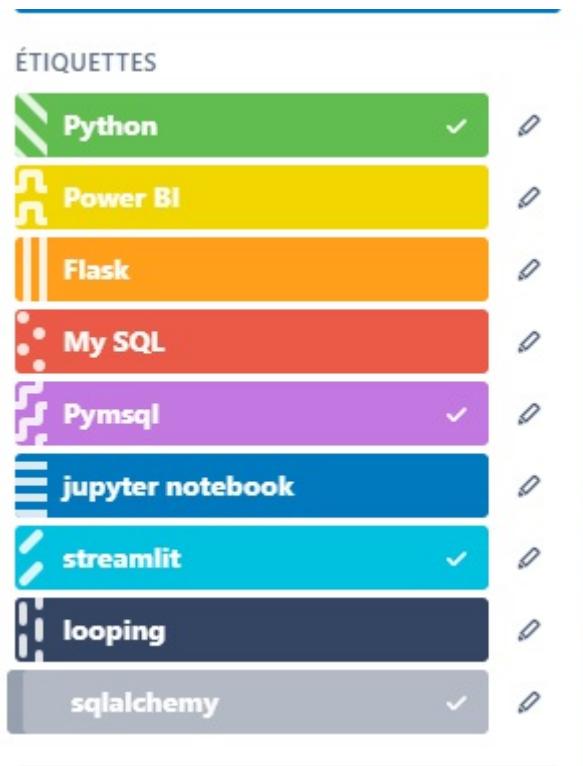
Outil	Utilisation
Python	Langage numérique utilisé pour le traitement des données et le développement de l'application
Jupyter notebook	Un notebook utilise pour le codage python. Ce notebook est exécuté par anaconda
VSCode	IDE pour les scripts python
Mysql	Langage de programmation pour la base de données et son serveur local
MySQL Workbench	IDE pour les codages mysql
Sqlalchemy	Librairie python pour la connexion vers Mysql
Pymysql	Librairie python pour la connexion vers Mysql
Streamlit	Une librairie python pour développer un web app machine learning
Trello	Gestion de projet
Looping	MCD – Conception de la base de donnée

1.6 Gestion de projet

Afin d'organiser mes étapes et les tâches du projet, j'ai utilisé l'outil « Trello » respectant la méthodologie d'agilité.



Chaque étiquette couleur représente la technique utilisée pour chaque étape.

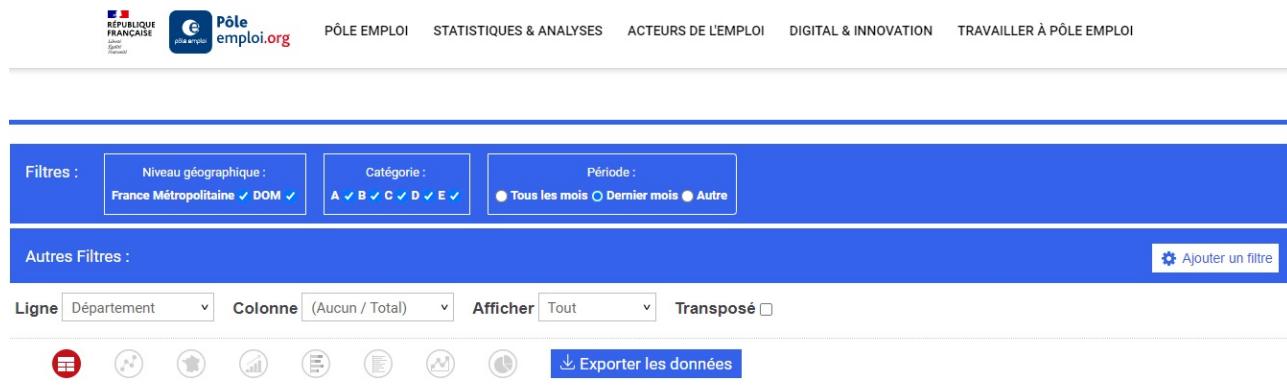


2. Conception et développement de la base de données

2.1 Collecte des données

2.1.1 Les sources des données

Les données des emplois, des chômeurs ou de la population active sont disponibles sur le site internet de l'Insee (Institut National de la Statistique et des Études Économique) et aussi sur le site de pôle emploi en format csv et excel.



The screenshot shows the Pôle emploi.org homepage. At the top, there is a navigation bar with links to 'PÔLE EMPLOI', 'STATISTIQUES & ANALYSES', 'ACTEURS DE L'EMPLOI', 'DIGITAL & INNOVATION', and 'TRAVAILLER À PÔLE EMPLOI'. Below the navigation bar, there are three filter boxes: 'Niveau géographique' (France Métropolitaine ✓ DOM ✓), 'Catégorie' (A ✓ B ✓ C ✓ D ✓ E ✓), and 'Période' (Tous les mois). A blue button 'Ajouter un filtre' is located below these. Underneath, there are dropdown menus for 'Ligne' (Département), 'Colonne' (Aucun / Total), 'Afficher' (Tout), and 'Transposé'. A row of icons follows, including a grid, a scatter plot, a star, a bar chart, a pie chart, a line graph, and a clock. A blue button 'Exporter les données' is at the bottom. The main content area displays a table titled 'Demandeurs d'emploi inscrits en fin de mois à Pôle emploi par Département - France entière - Juin 2021 - Données brutes'. The table has two columns: 'Département' and 'Nombre de demandeurs d'emploi'. The data rows are: Ain 01 (52 970), Aisne 02 (61 360), and Allier 03 (31 040).



The screenshot shows the Institut national de la statistique et des études économiques (Insee) website. The header includes the Insee logo, the text 'Institut national de la statistique et des études économiques', and 'Mesurer pour comprendre'. It also features a search bar and links for 'Menu', 'Contenu', 'Blog', 'Espace presse', 'Aide', and 'English'. The main navigation menu includes 'STATISTIQUES ET ÉTUDES', 'DÉFINITIONS, MÉTHODES ET QUALITÉ', 'SERVICES', and 'L'INSEE ET LA STATISTIQUE PUBLIQUE'. Below the menu, a breadcrumb trail shows 'Accueil > Statistiques et études'. The main content area is titled 'Statistiques et études' and features a sub-section 'Retrouvez ci-dessous les recherches les plus fréquentes' with links to various statistical series like 'Statistiques nationales', 'Statistiques locales', 'Recensement de la population', 'Comptes nationaux annuels', 'Etudes nationales', 'Etudes régionales', 'Conjoncture nationale', and 'Indices et séries chronologiques'. At the bottom, there are buttons for 'MES CRITÈRES' and 'Supprimer tous', and a search result summary '73 résultats de recherche.' with pagination controls 'Afficher par 10' and '1 à 10 / 73'.

2.1.2 Description des données

Comme indiqué dans l'introduction j'ai dû chercher les données pour les quatre indicateurs différents. Les valeurs que j'ai cherchées sont « nombre d'emplois » et « nombre de chômeurs » pour chaque indicateurs.

Voici un format d'un fichier .

Ex : Indicateur – Le fichier de nombre des emplois par diplôme

La première colonne de chaque diplôme représente le nombre d'emplois

A	B	C	D	E	F	G	H	I	J	K
1	T206 : Emploi et part dans l'emploi selon le plus haut diplôme obtenu, par sexe et âge regroupé, en moyenne annuelle									
2	SEXE : Ensemble									
3	AGEREG : 15 - 24 ans									
4										
5	Diplôme le plus élevé obtenu	Diplôme supérieur	Bacca...	Bacca...	CQP...	Brev...	Brev...			
6	Indicateurs	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcentage)	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcentage)	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcentage)	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcentage)	
7	Année									
8	1982	61	1,6	188	4,8	471	12,1	1 368	35,1	425
9	1983	79	2,1	197	5,2	484	12,7	1 398	36,6	414
10	1984	75	2,1	213	6,0	483	13,6	1 351	37,9	387
11	1985	62	1,8	220	6,2	482	13,6	1 374	38,9	365
12	1986	76	2,1	230	6,5	513	14,5	1 392	39,3	359
13	1987	64	1,8	244	7,0	473	13,5	1 407	40,1	326
14	1988	74	2,2	246	7,3	447	13,3	1 348	40,1	296
15	1989	83	2,5	243	7,3	430	12,9	1 341	40,1	289
16	1990	96	3,0	249	7,8	435	13,7	1 252	39,3	265
17	1991	82	2,8	248	8,5	406	14,0	1 138	39,1	240
18	1992	102	3,7	260	9,4	448	16,2	1 028	37,1	219
19	1993	106	4,2	290	11,4	436	17,1	895	35,1	233
20	1994	111	4,7	266	11,3	441	18,7	804	34,2	205
21	1995	125	5,3	287	12,3	479	20,5	744	31,8	209
22	1996	130	5,7	279	12,3	525	23,2	656	29,1	204
23	1997	150	7,0	290	13,6	509	23,9	585	27,5	198
24	1998	150	7,0	289	13,4	546	25,4	571	26,5	203
25	1999									

L'année considérée

Chaque feuille représente les données agrégée par le groupe d'âge et le sexe

Sexe

E – Ensemble (Femme et Homme)

H – Homme

F – Femme

Groupe d'âge

T – Totalité

T15 – Entre 15 et 24 ans

T25 – Entre 25 et 50 ans

T50 – 50 ans et plus

Donc toutes les données téléchargées ont déjà été agrégées par le groupe d'âge et le sexe. Parfois ils ont été séparés par feuille d'un fichier excel comme dans l'exemple au-dessus ou un fichier à part pour chaque groupe d'âge et sexe sur l'image suivante.

Nom	Modifié le	Type	Taille
Demandeurs d'emploi inscrits en fin de mois à Pôle emploi, Sexe _ Femmes, Tranche d'âge _ 50 ans ou...	07/07/2021 17:39	Classeur OpenOffi...	218 Ko
Demandeurs d'emploi inscrits en fin de mois à Pôle emploi, Sexe _ Femmes, Tranche d'âge _ De 25 à 49...	07/07/2021 17:37	Classeur OpenOffi...	236 Ko
Demandeurs d'emploi inscrits en fin de mois à Pôle emploi, Sexe _ Femmes, Tranche d'âge _ Moins de 2...	07/07/2021 17:38	Classeur OpenOffi...	217 Ko
Demandeurs d'emploi inscrits en fin de mois à Pôle emploi, Sexe _ Hommes, Tranche d'âge _ 50 ans ou...	07/07/2021 17:35	Classeur OpenOffi...	217 Ko
Demandeurs d'emploi inscrits en fin de mois à Pôle emploi, Sexe _ Hommes, Tranche d'âge _ De 25 à 49...	07/07/2021 17:34	Classeur OpenOffi...	233 Ko
Demandeurs d'emploi inscrits en fin de mois à Pôle emploi, Sexe _ Hommes, Tranche d'âge _ Moins de 2...	07/07/2021 17:32	Classeur OpenOffi...	216 Ko

Les données disponibles dans les fichiers

Indicateur 1 : Le diplôme

Nom	Type
Groupe d'age	string
Sexe	string
Année	integer
Diplôme	string
Nombre d'emplois (en millier)	float
Nombre de chômeurs (en millier)	float

Indicateur 2 : La catégorie socioprofessionnelle

Nom	Type
Groupe d'age	string
Sexe	string
Année	integer
Catégorie	string
Nombre d'emplois (en millier)	float
Nombre de chômeurs (en millier)	float

Indicateur 3 : La caractéristique de l'emploi

Nom	Type
Groupe d'age	string
Sexe	string
Année	integer
Caractéristique de l'emploi	string
Type caractéristique de l'emploi	string
Nombre d'emplois (en millier)	float

Indicateur 4 : Le secteur géographique (Les données par les départements français)

Nom	Type
Groupe d'age	string
Sexe	string
Année	integer
Département	string
Région	string
Pays	string
Population (en millier)	float
Nombre de chômeurs	float
Salaire moyen (par heure)	float

2.2 Conception de la base de données

Pour ce projet comme les données sont liées les unes aux autres, j'ai choisi de développer une base données relationnelle. Donc j'ai choisi MySQL comme technique pour développer cette base de donnée.

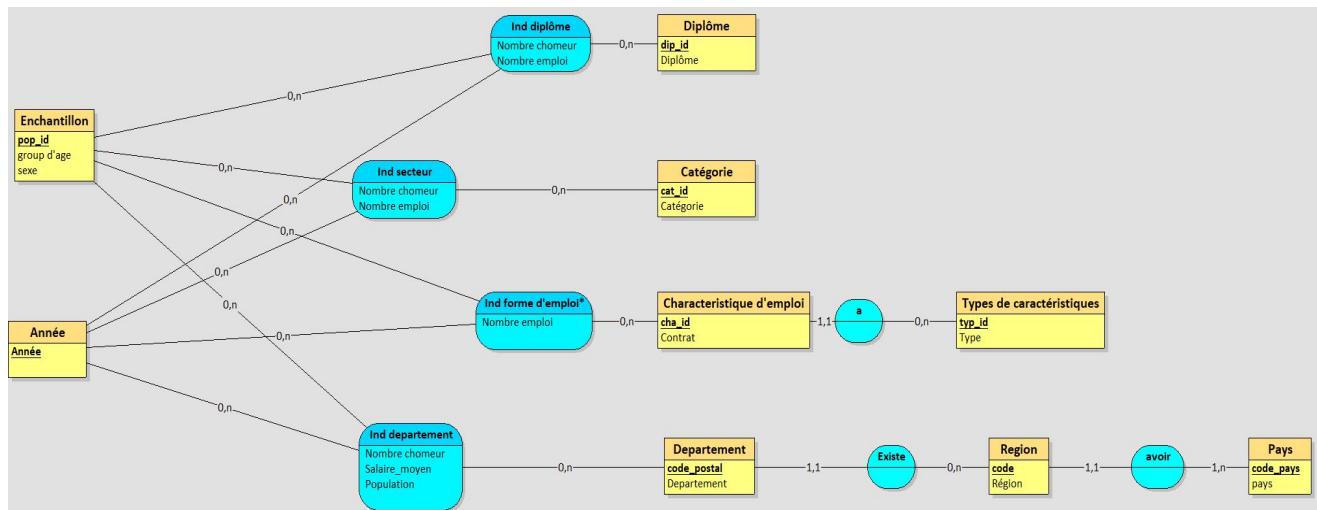
2.2.1 Modèle Conceptuel de Donnée

Comme expliqué dans la description des données des fichiers , chaque fichier représente des données d'un échantillon de la population qui ont une groupe d'âge et un sexe. Donc cette partie de groupe devient une entité principale de ce modèle.

Comme les données sont disponibles pour chaque année, dont l'année sera aussi une entité principale.

Les quatre indicateurs d'emploi aussi seront quatre différentes entités dans ce modèle car les données sont considérées pour chaque indicateur.

Voici le modèle final de la conception de la base de données. (outil : « looping »)



De l'indicateur caractéristique d'emploi, j'ai reçu des données pour trois différentes caractéristiques. Donc j'ai créé une nouvelle entité qui catégorise toutes ces caractéristiques (type_caractéristique).

De même pour les données par départements, elles peuvent être agrégées par région ou pays. Donc j'ai ajouté deux nouvelles entités afin d'avoir plus de types de résultats par secteur géographique.

Comme dans l'image au-dessus, l'entité jaune représente les entités principales. La table bleue représente une association ou une association porteuse. Les données (les chiffres) de nombre d'emploi ou nombre de chômeurs dépendent des entités l'année , l'échantillon et un d'indicateur. Donc ces données deviennent automatiquement les attributs de l'association porteuse de chaque indicateurs. La table dessous explique les relations d'entité avec ses cardinalités

Tables	Relation	Tables	Relation
Pays ↔ Région	One to Many	(Année ou Échantillon) ↔ (Diplôme ou Département ou Caractéristique ou Catégorie)	Many to Many
Région ↔ Département	One to Many		
Caractéristique ↔ Type	One to Many		

2.2.2 Modèle Logique de Données

Après la conception de notre base de données et afin de faciliter la création du modèle physique, dans cette étape on va identifier les clés primaire et les clés étrangères , MLD représentera les liens entre toutes ces entités avec des clés.

➔ Les entités principales

Année (code_année , Année)

Type_caractéristique (typ_id,Type)

Caractéristique_d_emploi (cha_id , Caractéristique, #typ_id)
#typ_id ← Type_caractéristique (typ_id)

Diplôme (dip_id , Diplôme)

Catégorie (cat_id , Catégorie)

Pays (code_pays , Pays)

Echantillon (pop_id , Group_d_age , Sexe)

Région (code_reg, Région, #code_pays)
#code_pays ← Pays (code_pays)

Departement (code_postal , Departement , #code_reg)
#code_reg ← Region (code_reg)

➔ Les association porteuses deviennent les tables avec les clés primaires composées par plusieurs clés étrangères.

Ind_diplôme (#code_année, #pop_id, #dip_id , Nb_emploi , Nb_chomeur)
#dip_id ← Diplôme (dip_id)
#code_année ← Année = (code_année)
#pop_id ← Echantillion(pop_id)

Ind_catégorie (#code_année, #pop_id, #cat_id, Nb_emploi , Nb_chomeur)
#cat_id ← Catégorie (cat_id)
#code_année ← Année = (code_année)
#pop_id ← Echantillion(pop_id)

```
Ind_char_d_emploi (#code_année, #pop_id, #cha_id, Nb_emploi )
# cha_id ← Caractéristique_d_emploi (cha_id)
#code_année ← Année = (code_année)
#pop_id ← Echantillion(pop_id)
```

```
Ind_departement (#code_année, #pop_id, #code_postal, Nb_chomeur,
Salaire_moyen, Population)
# code_postal ← Département (code_postal)
#code_année ← Année = (code_année)
#pop_id ← Échantillon(pop_id)
```

Par exemple prenons la table Ind_diplôme ,

Comme je l'ai expliqué avant « le nombre d'emploi » et « les chômeurs » dépendent des entités d'année, d'échantillon et de diplôme. Donc la table Ind_diplôme prend ces trois clés primaires pour composer sa clé primaire.

2.2.3 Modèle Physique des Données

Utilisant le MCD et le MLD au-dessus, on va développer la base de données de mon projet avec les techniques MySQL serveur et MySQL Workbench.

➔ Création de la base de donnée

```
1 • CREATE DATABASE IF NOT EXISTS emploi;
2 • USE emploi;
-
```

➔ Création des tables principales

(Année, Échantillon (population) , diplôme, caractéristiques d'emploi, catégorie, département, région, pays)

```

4   #####Année#####
5   #####
6   #####
7 • CREATE TABLE année(
8     année INT PRIMARY KEY
9   );
10 #####
11 ######Type characteristic#####
12 #####
13 #####
14 • CREATE TABLE type_char(
15     typ_id VARCHAR(6) PRIMARY KEY,
16     type_caractéristique VARCHAR(100)
17   );
18 #####
19 ######Characteristic d'emploi#####
20 #####
21 #####
22 • CREATE TABLE char_emploi(
23     cha_id VARCHAR(6) PRIMARY KEY,
24     caractéristique VARCHAR(200),
25     typ_id VARCHAR(6),
26     FOREIGN KEY (typ_id) REFERENCES type_char(typ_id)
27   );
28 #####
29   #####Diplôme#####
30   #####
31   #####
32 • CREATE TABLE diplôme(
33     dip_id VARCHAR(6) PRIMARY KEY,
34     diplôme VARCHAR(100)
35   );
36 #####
37   #####Catégorie#####
38   #####
39   #####
40 • CREATE TABLE catégorie(
41     cat_id VARCHAR(6) PRIMARY KEY,
42     catégorie VARCHAR(100)
43   );
44 #####
45   #####Population#####
46   #####
47   #####
48 • CREATE TABLE population(
49     pop_id VARCHAR(6) PRIMARY KEY,
50     group_dage VARCHAR(25),
51     sexe VARCHAR(10)
52   );
53 #####
54   #####Pays#####
55   #####
56   #####
57 • CREATE TABLE pays(
58     code_pays VARCHAR(6) PRIMARY KEY,
59     pays VARCHAR(100)
60   );
61 #####
62   #####Région#####
63   #####
64   #####
65 • CREATE TABLE région(
66     code_reg VARCHAR(6) PRIMARY KEY,
67     region VARCHAR(100),
68     code_pays VARCHAR(6),
69     FOREIGN KEY (code_pays) REFERENCES pays(code_pays)
70   );
71 #####
72   #####Département#####
73   #####
74   #####
75 • CREATE TABLE département(
76     code_postal VARCHAR(5) PRIMARY KEY,
77     département VARCHAR(100),
78     code_reg VARCHAR(6),
79     FOREIGN KEY (code_reg) REFERENCES région(code_reg)
80   );
81 #####

```

→ Les indicateurs (l'association porteuses dans MCD)

```

81
82      ||-----||          111      ||-----||
83      ||##Indicateur characteristics d'emploi##|          112      ||##Indicateur diplôme##|
84      ||-----||          113      ||-----||
85  ●  ○ CREATE TABLE indicateur_chardemploi(          114  ●  ○ CREATE TABLE indicateur_diplome(
86      année int,          115      année int,
87      pop_id VARCHAR(6),          116      pop_id VARCHAR(6),
88      cha_id VARCHAR(6),          117      dip_id VARCHAR(6),
89      nombre_employe FLOAT,          118      nombre_employe FLOAT,
90      FOREIGN KEY (pop_id) REFERENCES population(pop_id),          119      nombre_chomeur FLOAT,
91      FOREIGN KEY (année) REFERENCES année(année),          120      FOREIGN KEY (pop_id) REFERENCES population(pop_id),
92      FOREIGN KEY (cha_id) REFERENCES char_demploi(cha_id),          121      FOREIGN KEY (année) REFERENCES année(année),
93      PRIMARY KEY (pop_id,année,cha_id)          122      FOREIGN KEY (dip_id) REFERENCES diplôme(dip_id),
94  );          123      PRIMARY KEY (pop_id,année,dip_id)
95
96      ||-----||          124      );
97      ||##Indicateur catégorie##|
98      ||-----||          125
99  ●  ○ CREATE TABLE indicateur_catégorie(          126      ||-----||          127      ||##Indicateur département##|
100     année int,          127      ||-----||          128      ||-----||          129  ●  ○ CREATE TABLE indicateur_department(
101     pop_id VARCHAR(6),          128      année int,          130      année int,
102     cat_id VARCHAR(6),          129      pop_id VARCHAR(6),          131      pop_id VARCHAR(6),
103     nombre_employe FLOAT,          130      code_postal VARCHAR(6),          132      code_postal VARCHAR(6),
104     nombre_chomeur FLOAT,          131      population FLOAT,          133      population FLOAT,
105     FOREIGN KEY (pop_id) REFERENCES population(pop_id),          132      nombre_chomeur FLOAT,          134      nombre_chomeur FLOAT,
106     FOREIGN KEY (année) REFERENCES année(année),          133      salaire_moyen FLOAT,          135      FOREIGN KEY (pop_id) REFERENCES population(pop_id),
107     FOREIGN KEY (cat_id) REFERENCES catégorie(cat_id),          134      FOREIGN KEY (année) REFERENCES année(année),          136      FOREIGN KEY (code_postal) REFERENCES département(code_postal),
108     PRIMARY KEY (pop_id,année,cat_id)          135      PRIMARY KEY (pop_id,année,code_postal)
109  );          136
110
111      ||-----||          137
112      ||##Indicateur diplôme##|
113      ||-----||          138
114  ●  ○ CREATE TABLE indicateur_diplome(          139
115      année int,          140
116      pop_id VARCHAR(6),
117      dip_id VARCHAR(6),
118      nombre_employe FLOAT,
119      nombre_chomeur FLOAT,
120      FOREIGN KEY (pop_id) REFERENCES population(pop_id),
121      FOREIGN KEY (année) REFERENCES année(année),
122      FOREIGN KEY (dip_id) REFERENCES diplôme(dip_id),
123      PRIMARY KEY (pop_id,année,dip_id)
124  );
125
126      ||-----||          127
127      ||##Indicateur département##|
128      ||-----||          128
129  ●  ○ CREATE TABLE indicateur_department(          129
130      année int,          130
131      pop_id VARCHAR(6),          131
132      code_postal VARCHAR(6),          132
133      population FLOAT,          133
134      nombre_chomeur FLOAT,          134
135      salaire_moyen FLOAT,          135
136      FOREIGN KEY (pop_id) REFERENCES population(pop_id),          136
137      FOREIGN KEY (année) REFERENCES année(année),          137
138      FOREIGN KEY (code_postal) REFERENCES département(code_postal),          138
139      PRIMARY KEY (pop_id,année,code_postal)
140  );

```

Comme on l'a précisé avant toutes les tables indicateur ont des clés primaires composées avec plusieurs clés étrangères.

Prenons l'exemple de l'indicateur diplôme :

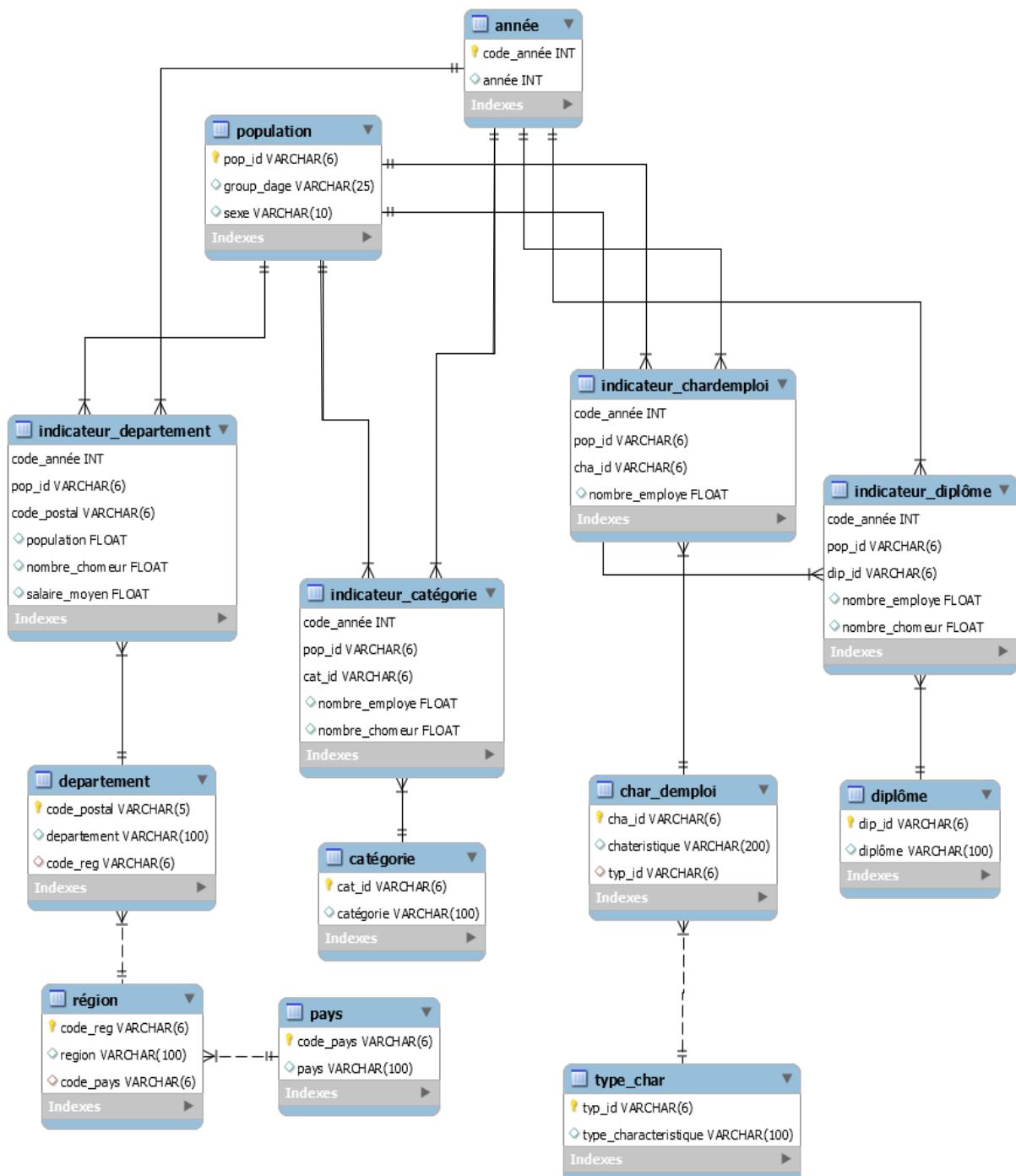
```

" " "
      ||-----||          " "
      ||##Indicateur diplôme##|
      ||-----||          " "
#-----#
1  ●  ○ CREATE TABLE indicateur_diplome(
2      année int,
3      pop_id VARCHAR(6),
4      dip_id VARCHAR(6),
5      nombre_employe FLOAT,
6      nombre_chomeur FLOAT,
7      FOREIGN KEY (pop_id) REFERENCES population(pop_id),
8      FOREIGN KEY (année) REFERENCES année(année),
9      FOREIGN KEY (dip_id) REFERENCES diplôme(dip_id),
10     PRIMARY KEY (pop_id,année,dip_id)
11 );

```

The diagram illustrates the structure of the `CREATE TABLE` statement for the `indicateur_diplome` table. It shows two blue arrows pointing from the table definition to annotations on the right. One arrow points to the section where foreign keys are defined, labeled "Définition des clés étrangères". The other arrow points to the section where the primary key is defined, labeled "Définition de la clé primaire composée".

Voici la dernière version de notre base de donnée après avoir exécuté le codage mysql. (MPD)



2.3 Traitement des données

Après avoir développé la base de donnée, cette étape sera une étape très importante du projet car c'est ici que les données seront nettoyées afin de les insérer dans la base de données. Nous avons des données brutes donc on utilisera le langage de programmation python surtout la librairie pandas pour cette étape.

2.3.1 Préparation ou nettoyage des données

Comme les données de ce projet sont déjà agrégées, j'ai plusieurs fichiers ou plusieurs feuilles des données pour chaque indicateur. Techniquement le but de cette étape sera de transformer les données brutes de l'image 1 à l'image 2.

image 1 :

A	B	C	D	E	F	G	H	I	J	K	L
1 T205 : Emploi et part dans l'emploi selon la catégorie socioprofessionnelle, par sexe et âge regroupé, en moyenne annuelle											
2 SEXE : Homme											
3 AGREG : 15 - 24 ans											
4											
5 Catégorie socioprofessionnelle agrégée en 9 modalités											
6 Indicateurs											
7 Année											
B	1982	102	4,6	36	1,6	8	0,3	236	10,7	148	6,7
9	1983	107	4,9	32	1,5	6	0,3	232	10,6	135	6,2
10	1984	115	5,7	31	1,5	6	0,2	224	11,0	130	6,4
11	1985	104	5,2	27	1,4	2	0,1	224	11,2	121	6,0
12	1986	95	4,8	32	1,6	12	0,6	222	11,1	104	5,2
13	1987	97	4,9	33	1,7	5	0,3	225	11,4	99	5,0
14	1988	85	4,5	39	2,0	6	0,3	216	11,4	90	4,7
15	1989	70	3,7	30	1,6	8	0,4	221	11,7	86	4,6
16	1990	61	3,4	33	1,9	8	0,5	203	11,3	92	5,1
17	1991	56	3,4	23	1,4	9	0,6	200	12,2	75	4,6
18	1992	44	2,8	17	1,1	11	0,7	202	12,9	75	4,8
19	1993	36	2,5	16	1,1	17	1,2	194	13,6	78	5,5
20	1994	38	2,8	15	1,1	17	1,0	156	11,8	73	5,5
21	1995	34	2,6	18	1,4	14	1,3	150	11,3	66	5,0
22	1996	31	2,4	14	1,2	18	0,5	159	11,6	63	4,9
23	1997	31	2,6	15	1,2	15	1,3	136	11,2	63	5,2
24	1998	21	1,8	14	1,2	17	1,4	137	11,3	57	4,7
25	1999	23	1,9	13	1,1	15	1,2	172	14,0	71	5,8
26	2000	21	1,6	16	1,2	23	1,8	194	14,7	79	6,0
27	2001	18	1,4	13	1,0	28	2,1	202	15,2	103	7,8
28	2002	16	1,2	13	1,0	35	2,6	216	16,1	120	8,9
29	2003	20	1,5	13	1,0	40	3,0	213	16,3	137	10,5
30	2004	12	1,0	11	0,8	44	3,4	208	16,0	136	10,4
31	2005	13	1,0	13	1,0	50	3,8	197	15,1	131	10,1
		44

image 2 :

Année	new	Catégorie	nombre_employe	
0	1982	E - T15	Agriculteurs exploitants	117
1	1983	E - T15	Agriculteurs exploitants	122
2	1984	E - T15	Agriculteurs exploitants	128
3	1985	E - T15	Agriculteurs exploitants	117
4	1986	E - T15	Agriculteurs exploitants	109
...	
463	2020	F - T50	Employés	1797
464	2020	H - T	Employés	1737
465	2020	H - T15	Employés	271
466	2020	H - T25	Employés	1039
467	2020	H - T50	Employés	426

Voici les étapes de la préparation des données.

La table Indicateur catégorie (« Ind_catégorie »)

- ✓ J'ai commencé par organiser tous les fichiers dans quatre différents dossiers qui représentent les quatre indicateurs.

	category	19/07/2021 23:58	Dossier de fichiers
	characteristics_emploi	06/07/2021 18:40	Dossier de fichiers
	departement	29/07/2021 19:54	Dossier de fichiers
	diplome	19/08/2021 02:17	Dossier de fichiers

Considérons qu'on va préparer l'indicateur catégorie socioprofessionnelle.

- ✓ Importer les libraires python nécessaires

```
import pandas as pd
import numpy as np
```

- ✓ Lire un fichier excel (Le fichier de nombre d'emploi)

```
#Lire le fichier excel (Emploi)
file_emploi_cat = pd.ExcelFile('../Raw_data/Categorie/T205.xlsx')
file_emploi_cat
```

- ✓ Lire les noms des feuilles s'il existe plusieurs.

```
#Obtenir les nom des feuille
sheets_emploi_cat = file_emploi_cat.sheet_names
sheets_emploi_cat
```

```
['E - T15',
'E - T25',
'E - T50',
'E - T',
'H - T15',
'H - T25',
'H - T50',
'H - T',
'H - T15',
'H - T25',
'H - T50',
'H - T']
```

Ici les feuilles représentent l'échantillon de la population qu'on avait expliqué au chapitre introduction. (pour certains indicateurs il y a plusieurs fichiers au lieu de plusieurs feuilles).

- ✓ On va commencer par consulter une seule feuille.

#Obtenir les données d'un feuille comme un dataframe df_emploi_cat = file_emploi_cat.parse('E - T15')																	
T205 : Emploi et part dans l'emploi selon la catégorie socioprofessionnelle, par sexe et âge regroupé, en moyenne annuelle																	
Indicateurs	Catégorie socioprofessionnelle aggregée en 9 mo...	Agriculteurs exploitants	Artisans, commerçants et chefs d'entreprises	Cadres et professions intellectuelles supérieures	Professions intermédiaires	Employés qualifiés	Employés non qualifiés	Ouvriers qualifiés	Ouvriers non qualifiés	Part dans l'emploi au sens du BIT (en milliers) pourcent..	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en milliers)	
5	Année	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	N	
6	1982	117	3	71	1.8	13	0.3	486	12.5	762	...	543	13.9	509	13.1	1031	2
7	1983	122	3.2	63	1.7	14	0.4	483	12.7	741	...	531	13.9	461	12.1	1044	2
8	1984	128	3.6	68	1.9	10	0.3	466	13.1	707	...	509	14.3	407	11.4	932	2
9	1985	117	3.3	61	1.7	9	0.3	459	13	697	...	534	15.1	391	11.1	922	2

✓ La suppression des lignes inutiles.

```
#Enlever les lignes unutiles
df_emploi_cat.drop(labels=[8,1,2,5,45,46,47],inplace=True)
df_emploi_cat
```

T205 : Emploi et part dans l'emploi selon la catégorie socioprofessionnelle, par sexe et âge regroupé, en moyenne annuelle																	
3	Catégorie socioprofessionnelle agrégée en 9 mo...	Agriculteurs exploitants	NaN	Artisans, commerçants et chefs d'entreprises	NaN	Cadres et professions intellectuelles supérieures	NaN	Professions intermédiaires	NaN	Employés qualifiés	...	Employés non qualifiés	NaN	Ouvriers qualifiés	NaN	Ouvriers non qualifiés	N
4	Indicateurs	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcent...)	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcent...)	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcent...)	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcent...)	Nombre d'emplois au sens du BIT (en milliers)	...	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcent...)	Nombre d'emplois au sens du BIT (en milliers)	Part dans l'emploi au sens du BIT (en pourcent...)	Nombre d'emplois au sens du BIT (en milliers)	Part d l'emp au sens BIT pourcen
6	1982	117	3	71	1.8	13	0.3	486	12.5	762	...	543	13.9	509	13.1	1031	2
7	1983	122	3.2	63	1.7	14	0.4	483	12.7	741	...	531	13.9	461	12.1	1044	2
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

✓ Supprimer les colonnes s'il y a des valeurs nulle. (Cette démarche enlèvera les valeurs en pourcentage qui seront inutiles comme des données car ça pourrait être calculé des que les données sont prêtes à utiliser)

```
#Enlever les columns unutiles
df_emploi_cat.dropna(axis=1,how='any',inplace=True)
df_emploi_cat
```

T205 : Emploi et part dans l'emploi selon la catégorie socioprofessionnelle, par sexe et âge regroupé, en moyenne annuelle																
3	Catégorie socioprofessionnelle agrégée en 9 mo...	Agriculteurs exploitants	Artisans, commerçants et chefs d'entreprises	Cadres et professions intellectuelles supérieures	Professions intermédiaires	Employés qualifiés	Employés non qualifiés	Ouvriers qualifiés	Ouvriers non qualifiés	Autres (militaires du contingent, non renseigné)
4	Indicateurs	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	...	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)	Nombre d'emplois au sens du BIT (en milliers)
6	1982	117	71	13	486	762	543	509	1031	360
7	1983	122	63	14	483	741	531	461	1044	359
8	1984	128	68	10	466	707	509	407	932	338

✓ Définir la première ligne comme les noms de colonne et supprimer la ligne 3 et la ligne 4.

```
#Enlever
labels = df_emploi_cat.iloc[0]
df_emploi_cat.drop(labels=[3,4], inplace = True)
df_emploi_cat
```

T205 : Emploi et part dans l'emploi selon la catégorie socioprofessionnelle, par sexe et âge regroupé, en moyenne annuelle																
6	...	1982	117	71	13	486	762	543	509	1031	360
7	...	1983	122	63	14	483	741	531	461	1044	359
8	...	1984	128	68	10	466	707	509	407	932	338
9	...	1985	117	61	9	459	697	534	391	922	343

✓ Renommer les colonnes

```
#Renommer Les colonnes
df_emploi_cat.columns = labels
df_emploi_cat
```

3	Catégorie socioprofessionnelle agrégée en 9 modalités	Agriculteurs exploitants	Artisans, commerçants et chefs d'entreprises	Cadres et professions intellectuelles supérieures	Professions intermédiaires	Employés qualifiés	Employés non qualifiés	Ouvriers qualifiés	Ouvriers non qualifiés	Autres (militaires du contingent, non renseigné)
6	1982	117	71	13	486	762	543	509	1031	360
7	1983	122	63	14	483	741	531	461	1044	359

- ✓ Renommer la première colonne comme ‘année’.

```
df_emploi_cat.rename(columns = {df_emploi_cat.columns[0]:"année"},inplace = True)
df_emploi_cat
```

3 année	Agriculteurs exploitants	Artisans, commerçants et chefs d'entreprises	Cadres et professions intellectuelles supérieures	Professions intermédiaires	Employés qualifiés	Employés non qualifiés	Ouvriers qualifiés	Ouvriers non qualifiés	Autres (militaires du contingent, non renseigné)	Ensemble
6 1982	117	71	13	486	762	543	509	1031	360	3892

- ✓ Enfin toutes les étapes nécessaires pour nettoyer sont faites, donc on rassemblera toutes ces étapes et définira une fonction.

```
def separation_df(fichier, sheet):
    temp = fichier.parse(sheet)
    temp = pd.DataFrame(temp)
    temp.drop(labels=[0,1,2,5],inplace=True)
    temp = temp.iloc[:, :-1]
    temp.dropna(axis=1,how='any',inplace=True)

    labels = temp.iloc[0]
    temp.drop(labels=[3,4], inplace = True)

    temp.columns = labels
    temp.rename(columns = {temp.columns[0]:"année"},inplace = True)

    temp.reset_index()
    return temp
```

On utilise un boucle pour appliquer cette fonction sur tous les feuilles du dataframe concaténé pour traiter les données

- ✓ Appliquer cette fonction à toutes les feuilles en utilisant une boucle.
Afin d'identifier chaque feuille avec le dernier dataframe concaténé, on ajoutera une nouvelle colonne ‘new’ avec son groupe d’age et son sexe.

```
#Création de variable pour le résultat final
fin_emp_cat = pd.DataFrame()
for i in range (0,len(sheets_emploi_cat)):

    #Utilisation de la fonction
    temp = separation_df(file_emploi_cat,sheets_emploi_cat[i])

    #Ajouté les type de données (age et sexe)
    temp['new'] = sheets_emploi_cat[i]

    #Concatener avec les données précédent
    fin_emp_cat = pd.concat([fin_emp_cat,temp])

fin_emp_cat
```

3 année	Agriculteurs exploitants	Artisans, commerçants et chefs d'entreprises	Cadres et professions intellectuelles supérieures	Professions intermédiaires	Employés qualifiés	Employés non qualifiés	Ouvriers qualifiés	Ouvriers non qualifiés	Autres (militaires du contingent, non renseigné)	Ensemble	new
6 1982	117	71	13	486	762	543	509	1031	360	3892	E-T15
7 1983	122	63	14	483	741	531	461	1044	359	3818	E-T15

- ✓ Sauf les colonnes ‘année’ et ‘new’, le restes des colonnes doit être transformé comme des lignes. Donc on utilise la méthode ‘melt’ pour cette étape.

```
id_vars = ['année','new']
value_vars = [el for el in fin_emp_cat.columns.to_list() if el not in id_vars]
value_vars

['Agriculteurs exploitants',
 'Artisans, commerçants et chefs d'entreprises',
 'Cadres et professions intellectuelles supérieures',
 'Professions intermédiaires',
 'Employés qualifiés',
 'Employés non qualifiés',
 'Ouvriers qualifiés',
 'Ouvriers non qualifiés',
 'Autres (militaires du contingent, non renseigné)',
 'Ensemble']
```

```
fin_emp_cat_melted = pd.melt(fin_emp_cat,value_vars = value_vars, id_vars = id_vars,
                             var_name = "Catégorie", value_name = "nombre_employe")
fin_emp_cat_melted
```

année	new	Catégorie	nombre_employe
0 1982	E-T15	Agriculteurs exploitants	117
1 1983	E-T15	Agriculteurs exploitants	122
2 1984	E-T15	Agriculteurs exploitants	128
3 1985	E-T15	Agriculteurs exploitants	117
4 1986	E-T15	Agriculteurs exploitants	109

Donc nous avons la table indicateur catégorie (« ind_catégorie ») qui est presque prête. En utilisant le dernier résultat on va créer la table d'échantillon (« population ») et la table (« catégorie »).

La table catégorie (« catégorie »)

- ✓ Création de la colonne de nom de catégorie.

```
#On va rassembler les 'Employés qualifiés', 'Employés non qualifiés' comme 'Employés'
#(car il y a que Les employés dans la table chomeur)
drop = ['Employés qualifiés', 'Employés non qualifiés']
catégorie = [el for el in list(fin_emp_cat_melted['Catégorie'].unique()) if el not in drop]
catégorie.insert(4,'Employés')
catégorie
```

```
['Agriculteurs exploitants',
 'Artisans, commerçants et chefs d'entreprises',
 'Cadres et professions intellectuelles supérieures',
 'Professions intermédiaires',
 'Employés',
 'Ouvriers qualifiés',
 'Ouvriers non qualifiés',
 'Autres (militaires du contingent, non renseigné)',
 'Ensemble']
```

- ✓ Création des identifiants pour chaque catégorie.

```
cat_ids = []
for i in range(len(catégorie[:-1])):
    cat_ids.append("cat_"+str(i+1))
cat_ids
```

```
['cat_1', 'cat_2', 'cat_3', 'cat_4', 'cat_5', 'cat_6', 'cat_7', 'cat_8']
```

- ✓ Création de la table catégorie.

```
catégorie = pd.DataFrame({'cat_id':cat_ids,"catégorie":catégorie[:-1]})

##### CATEGORIE #####
catégorie
```

cat_id	catégorie
0 cat_1	Agriculteurs exploitants
1 cat_2	Artisans, commerçants et chefs d'entreprises
2 cat_3	Cadres et professions intellectuelles supérieures
3 cat_4	Professions intermédiaires
4 cat_5	Employés
5 cat_6	Ouvriers qualifiés
6 cat_7	Ouvriers non qualifiés
7 cat_8	Autres (militaires du contingent, non renseigné)

La table d'échantillon de la population (« population »)

- ✓ En utilisant la table indicateur catégorie on va créer la table population.

```
p1 = fin_emp_cat_melted['new'].unique().tolist()

#On enlève Les E - ensemble et T - total d'age de cette liste
p1 = [el for el in p1 if ('E' not in el)]
p1.remove('H - T')
p1.remove('F - T')

pop_id = []

for i in range(0,len(p1)):
    pop_id.append("pop_"+str(i+1))

print(pop_id)
sexes = [el.split(' - ')[0] for el in p1]
ages = [el.split(' - ')[1] for el in p1]

population = pd.DataFrame({"pop_id": pop_id, "group_dage": ages, "sexe": sexes})

##### POPULATION #####
population
```

```
['pop_1', 'pop_2', 'pop_3', 'pop_4', 'pop_5', 'pop_6']

pop_id group_dage sexe
0 pop_1    T15   H
1 pop_2    T25   H
2 pop_3    T50   H
3 pop_4    T15   F
..      ..  ..  ..
```

Comme toute dernière étape de la table d'indicateur on remplacera les noms de catégorie et la population avec ses identifiants. Car ce sont des clés étrangères de la table « ind_catégorie ».

✓ Remplacer les identifiants de la population.

```
#Un dictionnaire population pour faciliter de la création de la table indicateur catégorie
dict_population = dict(zip(pi.population['pop_id']))
dict_population

{'H - T15': 'pop_1',
'H - T25': 'pop_2',
'H - T50': 'pop_3',
'F - T15': 'pop_4',
'F - T25': 'pop_5',
'F - T50': 'pop_6'}

#Remplacer les valeurs 'id' de la population
indicateur_catégorie['new'] = indicateur_catégorie['new'].apply(lambda x: dict_population[x] if x in dict_population.keys() else x)
indicateur_catégorie = indicateur_catégorie[indicateur_catégorie['new'].str.contains('pop')]

#Changer les nom des colonnes par rapport les tables SQL
indicateur_catégorie.rename(columns = {'new':'pop_id', 'Catégorie':'cat_id'}, inplace=True)
indicateur_catégorie
```

✓ Remplacer les identifiants de la catégorie

```
dict_cat = dict()
for el in catégorie.iterrows():
    dict_cat[el[1]['catégorie']] = el[1]['cat_id']

{'Agriculteurs exploitants': 'cat_1',
'Artisans, commerçants et chefs d'entreprises': 'cat_2',
'Cadres et professions intellectuelles supérieures': 'cat_3',
'Professions intermédiaires': 'cat_4',
'Employés': 'cat_5',
'Ouvriers qualifiés': 'cat_6',
'Ouvriers non qualifiés': 'cat_7',
'Autres (militaires du contingent, non renseigné)': 'cat_8'}

#Remplacer catégorie par son code de catégorie (table "catégorie")
indicateur_catégorie['cat_id'] = indicateur_catégorie['cat_id'].apply(lambda x: dict_cat[x])
indicateur_catégorie
```

Enfin la dernière version des tables « catégorie », « ind_catégorie » et « population » sont prêtes à être insérées dans la base de données.

	pop_id	group_dage	sexe	catégorie	cat_id	catégorie
0	pop_1	T15	H		0	Agriculteurs exploitants
1	pop_2	T25	H		1	Artisans, commerçants et chefs d'entreprises
2	pop_3	T50	H		2	Cadres et professions intellectuelles supérieures
3	pop_4	T15	F		3	Professions intermédiaires
4	pop_5	T25	F		4	Employés
5	pop_6	T50	F		5	Ouvriers qualifiés
					6	Ouvriers non qualifiés
					7	Autres (militaires du contingent, non renseigné)

	année	pop_id	cat_id	nombre_employé	nombre_chomeur	
ind_catégorie	156	1982	pop_1	cat_1	102	1
	157	1983	pop_1	cat_1	107	0
	158	1984	pop_1	cat_1	115	1
	159	1985	pop_1	cat_1	104	1

Ce module représente une partie de notre base de données physique. Le reste des fichiers sera traité avec les étapes similaires et enfin toutes les tables seront prêtes pour l'insertion.

2.3.2 Insertion des données

Après avoir nettoyé et préparé les données, nous allons commencer à insérer les données dans la base de données développée. Pour cette étapes nous utilisons le langage de programmation python et sa librairie pymysql dans le but de créer une connexion avec mysql serveur et de travailler depuis python.

Importation des libraires

```
: import pymysql as ps
import pandas as pd
import numpy as np

import os
```

✓ Connexion au serveur mysql

```
#Connection à server mysql
connection = ps.connect(host = 'localhost',
                       user = 'root',
                       password = '████████',
                       database = 'emploi')

try:
    cur = connection.cursor()
    print("Connection successfull")

except mysql.connector.Error as error:
    print("Error in connection: ",error)

Connection successfull
```

Pour se connecter au serveur mysql on utilise la fonction « connect » de « pymysql » et on passera les paramètres le nom de la connexion ou ip, l'utilisateur, le mot de passe et le nom de base de donnée.

✓ On insère les données utilisant la connexion qu'on avait créée. (Ici on considère l'insertion des données de la table diplôme)

```
diplome = pd.read_csv("../inputs/diplome.csv")
diplome

# # Création d'un nouveau record
sql_insert = "INSERT INTO diplome VALUES (%s, %s)"

# Execution de la requete
try:
    for ligne in diplome.iterrows():
        temp = tuple(ligne[1])
        cur.execute(sql_insert,temp)

    # Commiter l'exécution afin de sauvegarder les changements
    connection.commit()

    print("L'exécution de la requête est réussie")

except ps.Error as e:
    print("Une erreur s'est produite lors de l'exécution de la requête: ",e)

L'exécution de la requête est réussie
```

Dans la table diplôme il n'y a que deux colonnes. Donc à la requête d'insertion au lieu des valeur on remplace ‘%s’ qui représente un string.

En lisant la table avec une boucle on remplace ‘%s’ avec chaque valeur d'une ligne de la table.

Enfin en utilisant la même connexion, on insérera toutes les données dans la base de données.

2.4 Sécurisation et Optimisation de la base de données

J'ai crée une base de donnée local afin de faciliter l'accès aux données depuis l'application web (qui sera expliqué au chapitre 4). Actuellement cette base de données est accessible pour tout le monde. Si ces données doivent être protégées par les modifications ou les accès anonymes, je peux créer les utilisateurs depuis le console MySQL où je pourrai gérer les privilèges des utilisateurs.

Pour créer des utilisateurs je vais connecter avec console mysql en tant qu'un administrateur (root). Ensuite je vais créer un utilisateur 'user2' avec un mot de passe et attribuer les privilèges.

```
Anaconda Prompt (anaconda3)
(base) C:\Users\gneis>cd C:\Program Files\MySQL\MySQL Server 8.0\bin
(base) C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.23 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create user 'user2'@'localhost';
Query OK, 0 rows affected (0.04 sec)

mysql> alter user 'user2'@'localhost' identified by 'mdp2';
Query OK, 0 rows affected (0.02 sec)

mysql> grant select on emploi.* to 'user2'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> show grants for 'user2'@'localhost';
+-----+
| Grants for user2@localhost |
+-----+
| GRANT USAGE ON *.* TO `user2`@`localhost` |
| GRANT SELECT ON `emploi`.* TO `user2`@`localhost` |
+-----+
2 rows in set (0.00 sec)

mysql> exit
Bye

(base) C:\Program Files\MySQL\MySQL Server 8.0\bin>
```

Ici j'ai attribué seulement le droits de lire (SELECT) la base données d'emploi. Pour vérifier ca, je vais connecter comme 'user2' et vérifier s'il pourra insérer les données.

```
(base) C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u user2 -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.23 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database   |
+-----+
| emploi     |
| information_schema |
+-----+
2 rows in set (0.00 sec)

mysql> use emploi;
```

```
mysql> SELECT * FROM année;
+-----+
| année |
+-----+
| 1975  |
| 1976  |
| 1977  |
| 1978  |
| 1979  |
+-----+
```

```
mysql> INSERT INTO année VALUES (2021);
ERROR 1142 (42000): INSERT command denied to user 'user2'@'localhost' for table 'année'
mysql>
```

Donc si le 'user2' essaye d'insérer les données, la base de données ne donne pas l'accès à insérer les données.

Afin d'optimiser et de protéger les données, il est mieux d'effectuer un back up de données pour éviter la perte de données ou accéder aux données en cas de serveur base de données en panne. Donc j'utilise une requête mysql afin de sauvegarder les tables en tant que les fichiers au format csv. (cette requête est équivalent à sqldump)

```
19 •  SELECT * FROM année
20    INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/année.csv'
21    FIELDS TERMINATED BY ','
22    OPTIONALLY ENCLOSED BY ""
23    ESCAPED BY ''
24    LINES TERMINATED BY '\n';
```

3. Analyser et visualiser les données

3.1 L'analyse et la visualisation (manuelle)

Après avoir développé notre base de données, on effectuera des analyses et on représentera ces analyses par quelques visualisations dans le but de trouver les informations utiles.

Les outils(libraires) techniques utilisés

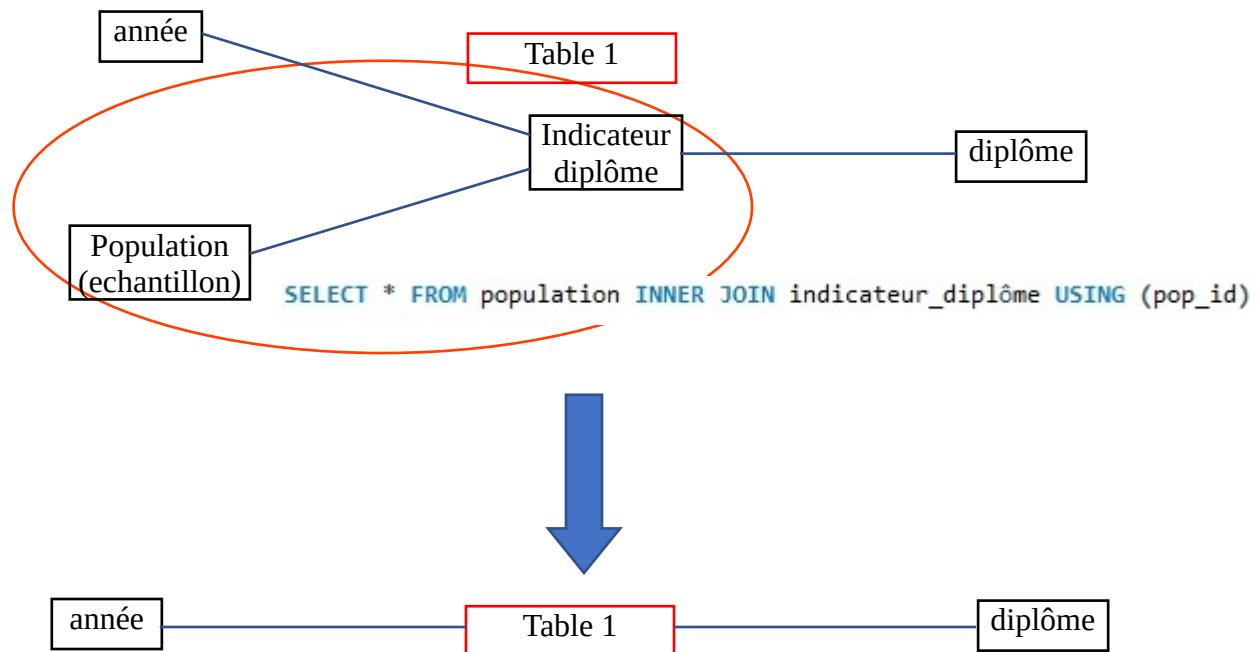
Outils	Utilisation
Python, Python - Pandas	Traitement des données
MySQL	Exécution des requêtes
Python – Matplotlib, Seaborn, Plotly, Folium	Visualisation des graphes
Sqlalchemy, PyMysql	Connexion de la base de données

Voici quelques analyses et ces représentations.

I. L'évolution d'emplois ou de chômeurs par indicateur diplôme.

Le diplôme est un élément très important pour un emploi et ici on regardera comment le diplôme impact sur l'évolution d'emplois.

Comme il y a les tables d'année, de la population, d'indicateur diplôme et de diplôme à rassembler, j'utilise «INNER JOIN » pour récupérer toute les données en commun. Mais ces tables n'ont pas une relation linéaire.



On commence par utiliser « INNER JOIN » entre Population ou Indicateur diplôme (qui devient une sous- requête dans la requête finale) afin de donner une relation linaire entre les tables.
(`Population + Indicateur diplôme = Table 1`)

Ensuite en utilisant la Table on va développer la requête finale avec «INNER JOIN».

* Note : On utilisera la même stratégie pour les autres indicateurs car ils ont le même type de relation.

Comme on a besoin que des données par année et par diplômé, j'utilise « »GROUP BY » pour grouper les données.

On utilise la fonction d'agrégation « SUM » pour avoir la totalité des données numériques. On multiplie les valeur par 1000 parce que ils sont en millier.

La requête finale,

```

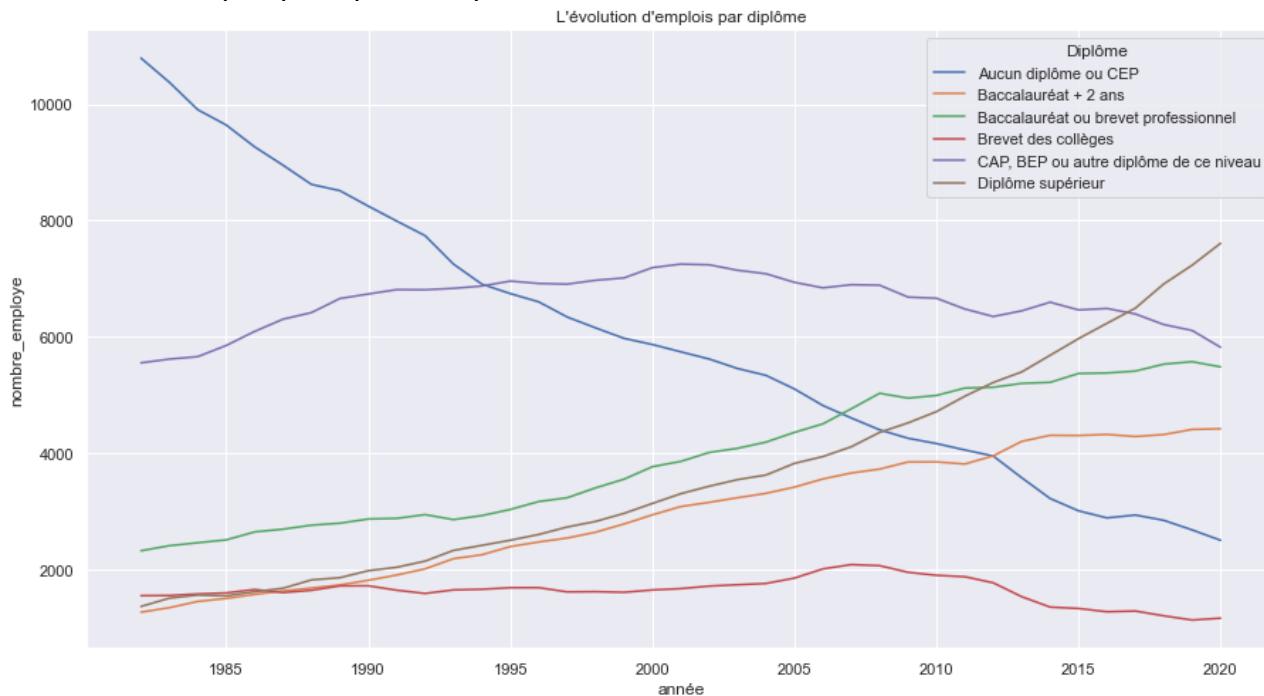
43 •   SELECT a.année, d.diplôme, SUM(p.nombre_employé)*1000 AS nombre_employé ,SUM(p.nombre_chomeur)*1000 AS nombre_chomeur
44      FROM année a
45      INNER JOIN (SELECT * FROM population INNER JOIN indicateur_diplôme USING (pop_id)) p USING (année)
46      INNER JOIN diplôme d USING (dip_id)
47      GROUP BY a.année, d.diplôme
48      ORDER BY a.année;

```

La table des résultats est,

	année	diplôme	nombre_employé	nombre_chomeur
▶	1982	Aucun diplôme ou CEP	10791000	922000
	1982	Baccalauréat + 2 ans	1266000	40000
	1982	Baccalauréat ou brevet professionnel	2322000	141000
	1982	Brevet des collèges	1550000	103000
	1982	CAP, BEP ou autre diplôme de ce niveau	5552000	403000
	1982	Diplôme universitaire	1365000	64000

L'évolution d'emploi par diplôme et par année



Les chômeurs et les emplois par diplôme sur l'année 2019

```

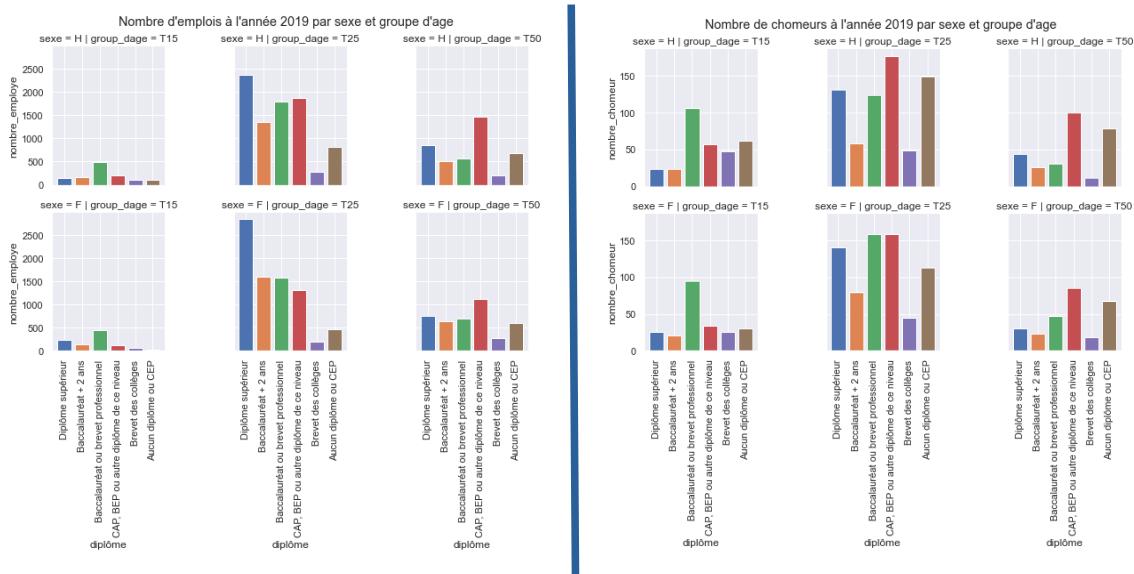
43 •   SELECT a.année, d.diplôme, p sexe, p.group_dage, p.nombre_employé*1000 , p.nombre_chomeur*1000
44      FROM année a
45      INNER JOIN (SELECT * FROM population INNER JOIN indicateur_diplôme USING (pop_id)) p USING (année)
46      INNER JOIN diplôme d USING (dip_id)
47      HAVING a.année = 2019
48      ORDER BY a.année;

```

	année	diplôme	sexe	group_dage	p.nombre_employé*1000	p.nombre_chomeur*1000
▶	2019	Diplôme supérieur	H	T15	151000	23000
	2019	Diplôme supérieur	H	T25	2372000	131000
	2019	Diplôme supérieur	H	T50	862000	44000
	2019	Diplôme supérieur	F	T15	247000	26000
	2019	Diplôme supérieur	F	T25	2835000	140000
	2019	Diplôme supérieur	F	T50	764000	30000
	2019	Baccalauréat + 2 ans	H	T15	168000	23000
	2019	Baccalauréat + 2 ans	H	T25	1357000	58000

Pour trouver des informations de l'année 2019 on ajoute la condition avec une clause

23 « HAVING »



Conclusion : Ces graphiques démontrent une augmentation du taux d'emploi pour les plus diplômés alors que la part des emplois occupés par les non-diplômés ne cessent de diminuer. Il est donc indispensable aujourd'hui d'avoir des diplômes pour augmenter ses chances de trouver un emploi. Avant 1990, il y avait que très peu des gens diplômés jusqu'à BAC+5 et presque 90% gens qui ont du travail sont non diplômés à l'époque, alors que actuellement c'est l'inverse.

II. L'évolution d'agriculteurs et les autres catégories

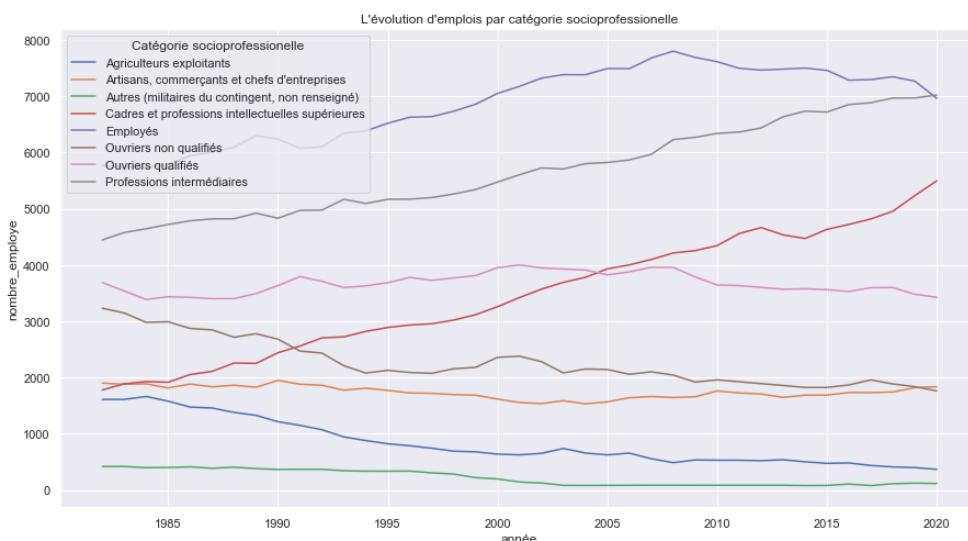
J'utilise la même stratégie de la partie diplôme pour rassembler toutes les tables de l'indicateur catégorie.

L'évolution d'emplois et de chômeurs par année et par catégorie

```

74 •   SELECT a.année, c.catégorie, SUM(i.nombre_employe)*1000 AS nombre_employe ,SUM(i.nombre_chomeur)*1000 AS nombre_chomeur
75     FROM année a
76     INNER JOIN indicateur_catégorie i USING (année)
77     INNER JOIN catégorie c USING (cat_id)
78     GROUP BY a.année, c.catégorie
79     ORDER BY a.année;
    
```

	année	catégorie	nombre_employe	nombre_chomeur
▶	1982	Agriculteurs exploitants	1610000	4000
	1982	Artisans, commerçants et chefs d'entreprises	1899000	34000
	1982	Autres (militaires du contingent, non renseigné)	417000	0
	1982	Cadres et professions intellectuelles supérieures	1778000	54000
	1982	Employés	5768000	393000
	1982	Ouvriers non qualifiés	3233000	343000
	1982	Ouvriers qualifiés	3688000	247000
	1982	Professions intermédiaires	4446000	171000



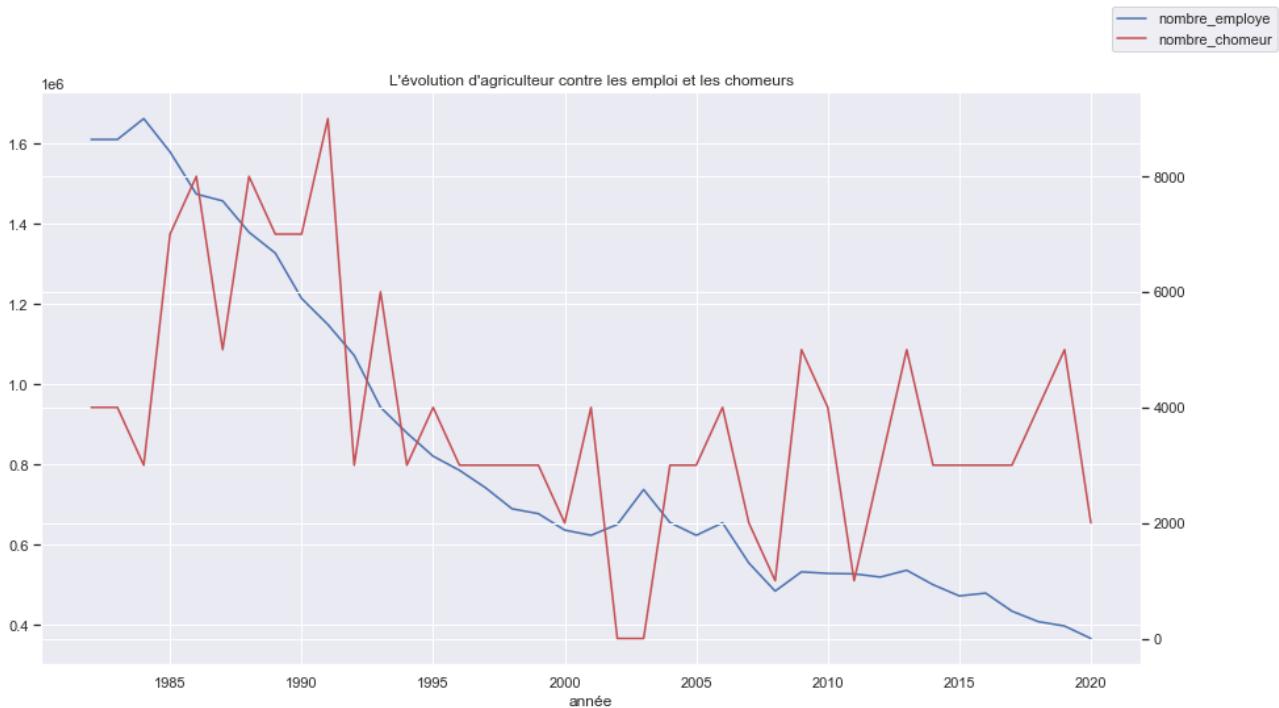
L'évolution de la catégorie des agriculteurs

```

64 •   SELECT a.année, c.catégorie, SUM(i.nombre_employé)*1000 AS nombre_employé ,SUM(i.nombre_chomeur)*1000 AS nombre_chomeur
65      FROM année a
66      INNER JOIN indicateur_catégorie i USING (année)
67      INNER JOIN catégorie c USING (cat_id)
68      GROUP BY a.année, c.catégorie
69      HAVING c.catégorie LIKE '%Agriculteurs%'
70      ORDER BY a.année;
    
```

	année	catégorie	nombre_employé	nombre_chomeur
▶	1982	Agriculteurs exploitants	1610000	4000
	1983	Agriculteurs exploitants	1610000	4000
	1984	Agriculteurs exploitants	1662000	3000
	1985	Agriculteurs exploitants	1579000	7000
	1986	Agriculteurs exploitants	1474000	8000
	1987	Agriculteurs exploitants	1457000	5000
	1988	Agriculteurs exploitants	1379000	8000
	1989	Agriculteurs exploitants	1327000	7000

Pour récupérer les informations des agriculteurs j'utilise une condition avec le syntax « LIKE ». c.catégorie LIKE ‘%Agriculteurs %’ sorte un résultats équivalent à notre recherche. Ici les lignes qui contiennent la chaîne de caractères « agriculteurs ». « % » symbole représente la possibilité d'avoir autres caractères avant ou après le mot.



Conclusion : En consultant les nombres de chômeurs, la courbe reste en position statique. Donc il reste toujours entre 0 à 8000 chômeurs par année. Mais la courbe d'emplois représente une grosse baisse d'emplois dans l'agriculture. La courbe de l'emploi des agriculteurs ne cesse pas de décroître.

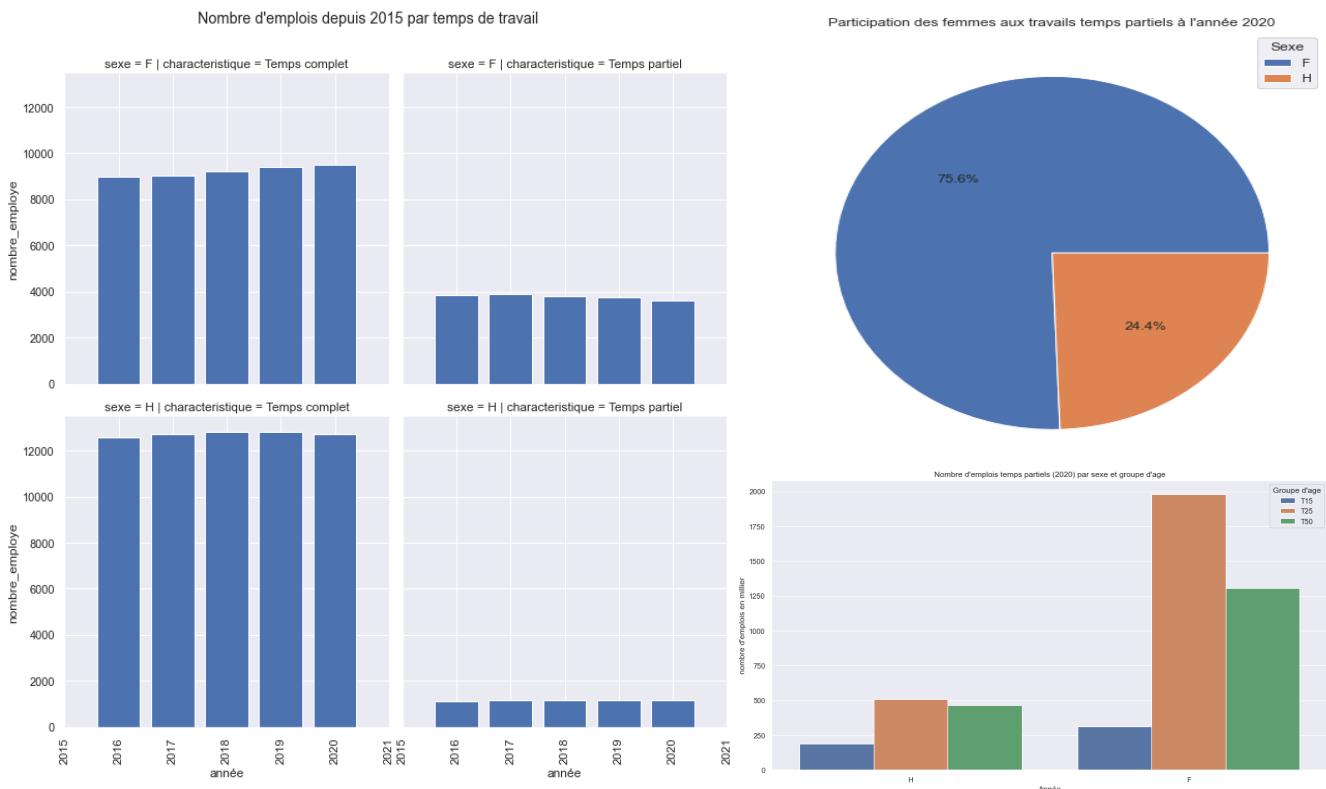
III. Participation des hommes et des femmes par temps de travail (d'indicateur caractéristique d'emploi)

Pour trouver les informations de la caractéristique temps de travail, je fais une recherche des ligne qui contiennent le mot ‘temps’ et je vais grouper ces données par l’année , la caractéristique, le sexe et l’age. Pour cette analyse j’utilise que les données depuis 2015. Donc j’ajoute une deuxième condition qui recherche des informations depuis 2015.

```

72 •   SELECT a.année, p.sex, p.group_dage, c.characteristique, p.nombre_employe FROM année a
73     INNER JOIN (SELECT * FROM population INNER JOIN indicateur_chardemploi USING (pop_id)) p USING (année)
74     INNER JOIN char_demploi c USING (cha_id)
75     INNER JOIN type_char t USING (typ_id)
76     WHERE c.characteristique LIKE 'temp%' AND a.année >2015
77     ORDER BY p.année;
    
```

	année	sex	group_dage	characteristique	nombre_employe
▶	2016	F	T50	Temps partiel	1321
	2016	H	T50	Temps partiel	456
	2016	H	T50	Temps complet	3585
	2016	H	T15	Temps partiel	179
	2016	H	T15	Temps complet	959
	2016	F	T25	Temps complet	5731
	2016	F	T25	Temps partiel	2199
	2016	F	T50	Temps complet	2627
	2016	H	T25	Temps partiel	501



Conclusion : La majorité des emplois proposés sont à temps complet. Concernant les emplois à temps partiels, la majorité des emplois sont occupés par les femmes. Les emplois à temps plein sont occupés majoritairement par les hommes.

IV. Les chômeurs par département ou par région

Notre base de données a déjà les données par département. Pour récupérer les données par région on va simplement grouper les données par région et récupérer la totalité des valeurs.

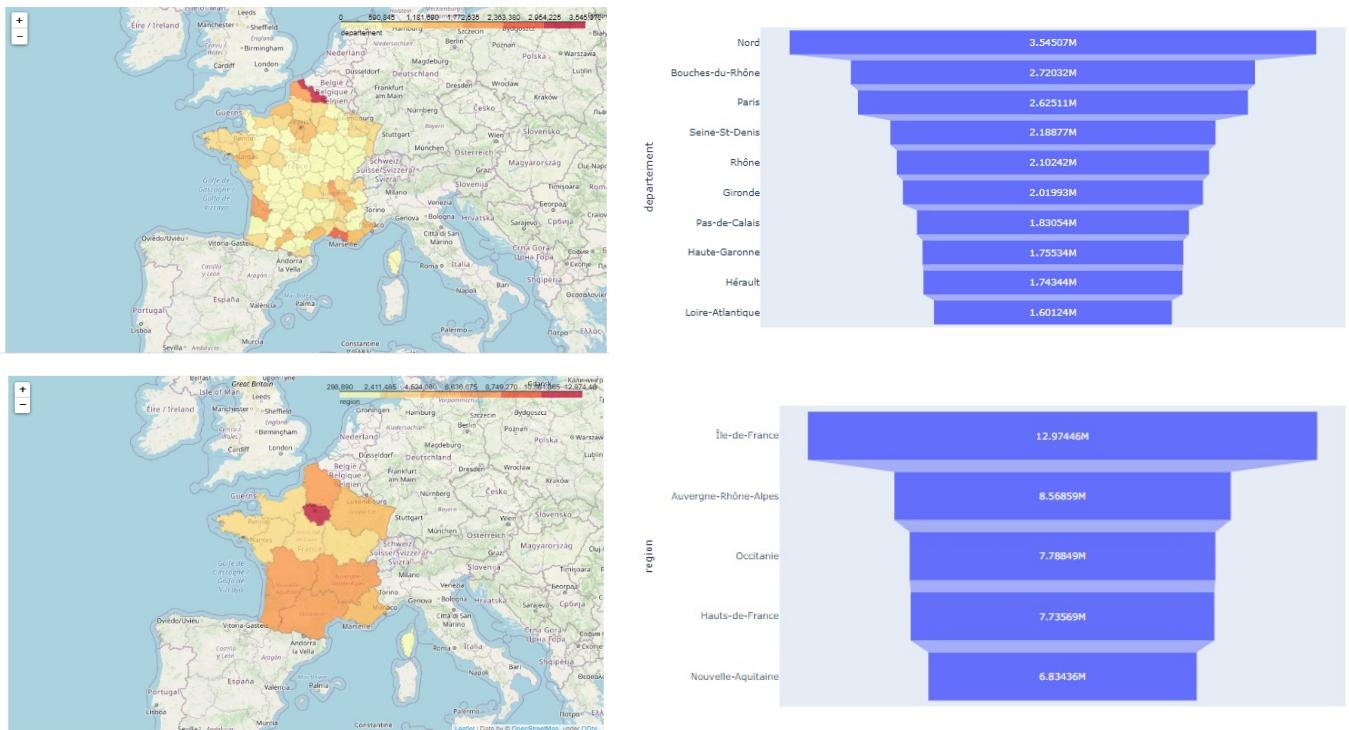
Par département

	année	code_postal	departement	population	nombre_chomeur	salaire_moyen
▶	2018	01	Ain	647634000	593120	9.96
	2018	02	Aisne	533316000	745640	9.23
	2018	03	Allier	337171000	385270	9.07
	2018	04	Alpes-de-Haute-Provence	164068000	219980	9.35
	2018	05	Hauts-Alpes	140698000	181820	9.04
	2018	06	Alpes-Maritimes	1086219000	1389820	10.61
	2018	07	Ardeche	336666000	411010	9.24

Par région

	a.année	r.code_reg	r.region	SUM(i.population*1000) AS population	SUM(i.nombre_chomeur) AS nombre_chomeur	ROUND(AVG(i.salaire_moyen),2) AS salaire_moyen
88	•	a.année,	r.code_reg,	SUM(i.population*1000) AS population,	SUM(i.nombre_chomeur) AS nombre_chomeur,	
89				ROUND(AVG(i.salaire_moyen),2) AS salaire_moyen	FROM année a	
90				INNER JOIN indicateur_departement i USING (année)		
91				INNER JOIN departement d USING (code_postal)		
92				INNER JOIN région r USING (code_reg)		
93				WHERE a.année = 2018		
94				GROUP BY r.region		
95				ORDER BY a.année;		

Les chômeurs par département et région à l'année 2018



Conclusion: Les départements où il y a le plus de chômeurs sont le Nord et Paris. La région comprenant le plus de chômeurs est l'Ile de France.

V. La population sans emplois

Je n'ai pas des données d'emplois par département. Mais je peux comparer la totalité d'emplois par un des indicateurs (sauf département) avec les données de la population qui existe dans les données départemental. En comparant ces données je peux récupérer la population sans emploi (tension d'emploi) de la France.

Cette information nous aidera à trouver si nous avons suffisamment d'emplois pour satisfaire la population. Ici je récupère la population sans compter les enfants. Donc j'ajoute une condition de ne pas compter l'échantillon de la population moins de 15 ans (T0 - Entre 0 à 15 ans).

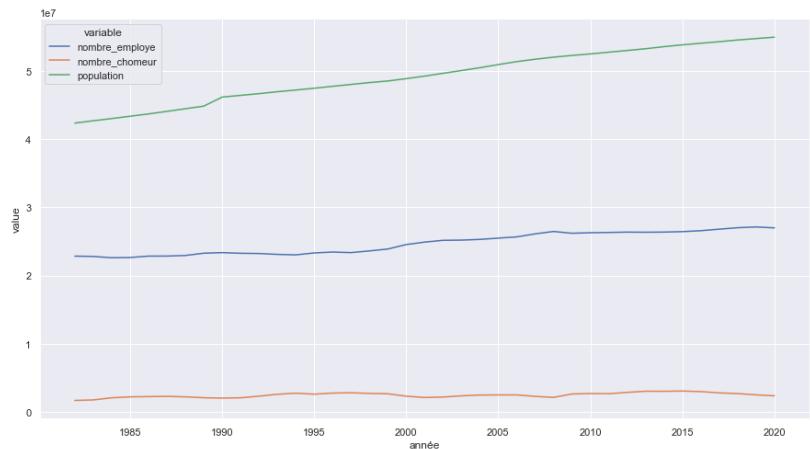
```

38 •   SELECT * FROM
39   (SELECT a.année, SUM(i.nombre_employe)*1000 AS nombre_employe ,SUM(i.nombre_chomeur)*1000 AS nombre_chomeur FROM année
40     INNER JOIN indicateur_diplôme i USING (année)
41     GROUP BY a.année
42     ORDER BY a.année) AS A
43   INNER JOIN
44     (SELECT a.année, SUM(p.population*1000) AS population FROM année a
45       INNER JOIN (SELECT * FROM population INNER JOIN indicateur_departement USING (pop_id)) p USING (année)
46       WHERE p.group_dage != 'T0'
47       GROUP BY a.année
48       ORDER BY a.année) AS B
49   USING (année);

```

	année	nombre_employe	nombre_chomeur	population
▶	1982	22846000	1673000	42368744000
	1983	22805000	1752000	42711608000
	1984	22614000	2060000	43032838000
	1985	22647000	2198000	43374690000
	1986	22850000	2235000	43714523000
	1987	22858000	2272000	44086888000
	1988	22943000	2200000	44476683000
	1989	23278000	2072000	44856201000
	1990	23364000	2017000	46182655000
	1991	23267000	2064000	46439025000

Conclusion : Cette image représente l'évolution de la population éligible pour travailler (vert), de nombre d'emplois (bleu) et de chômeurs (orange). La courbe d'emplois reste toujours plus basse que la population actif. Dont environ 50 % de la population n'ont pas d'emplois.



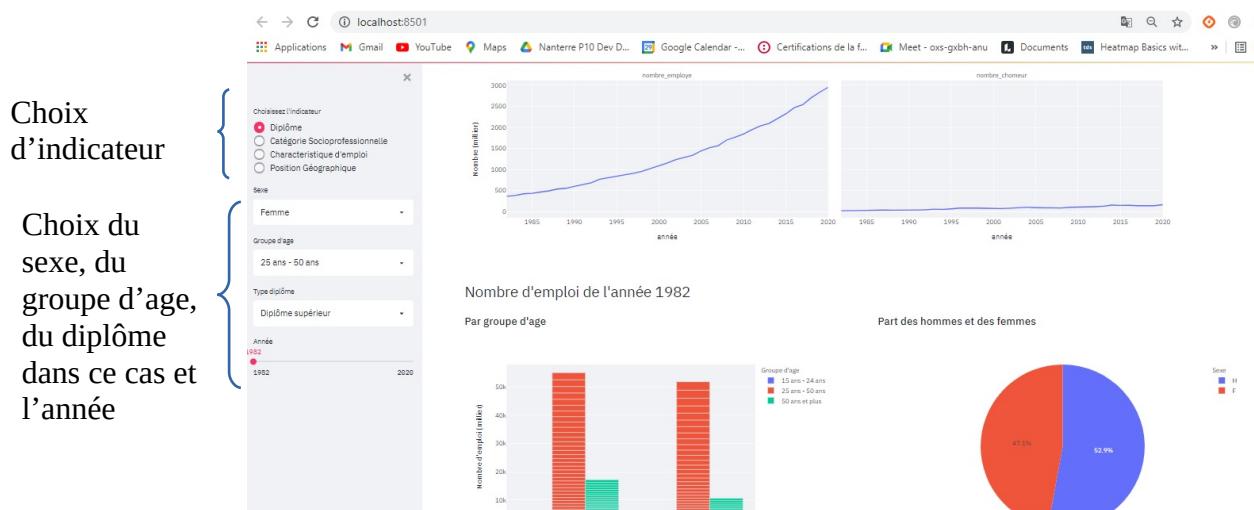
3.2 Application web créée par streamlit

Avec toutes ces données inserées on pourra représenter plusieurs types d'analyses. Afin de faciliter l'analyse complète au lieu de faire une analyse manuelle , j'ai décidé de créer une petite application en utilisant python streamlit. Streamlit est une librairie python pour créer des webapp machine learning.

3.2.1 La démarche du développement de webapp l'évolution de l'emploi par streamlit

La stratégie utilisée pour créer cette webapp :

- Création de quatre interfaces différentes (méthode modularisation) pour les quatre indicateurs et de diriger vers ces interfaces par un simple menu créé par des bouton radio (en haut à gauche dans l'image).
- Laisser le choix de groupe d'âge et le sexe et le type d'indicateur avec les menus déroulants afin de faciliter l'analyse par l'utilisateur de cette application.
- Le choix de l'année considéré est représentée par un slider.



3.2.2 Les techniques utilisées

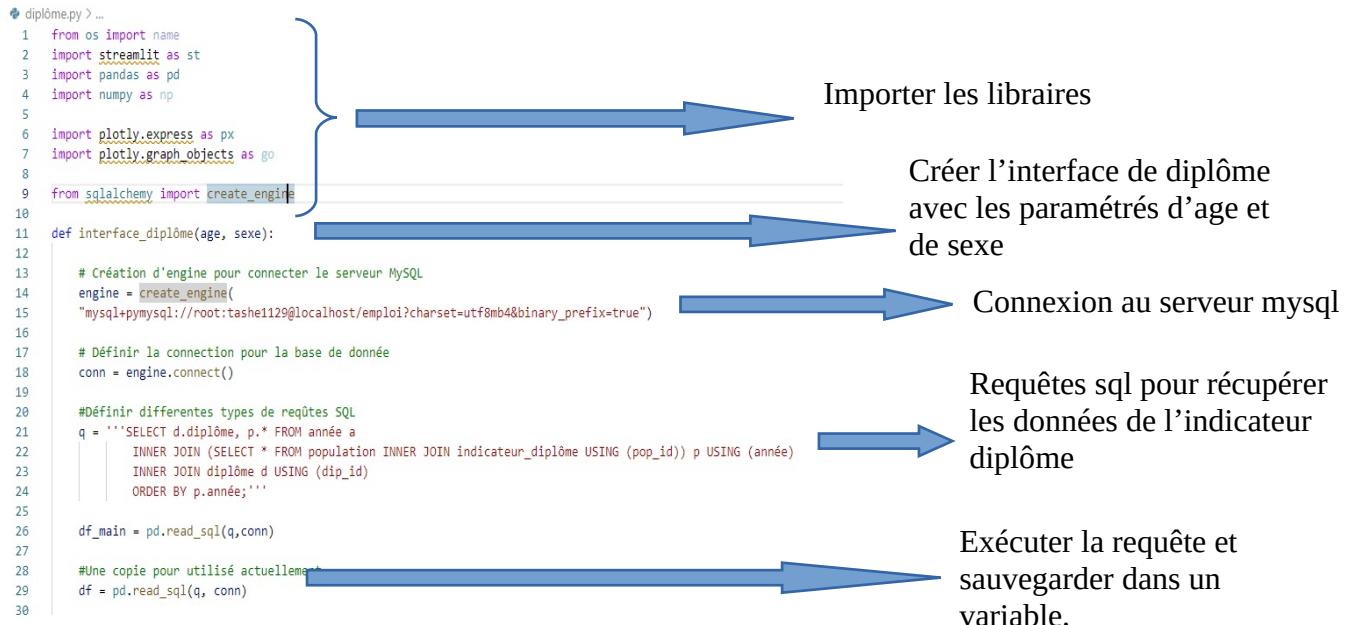
Outil	Utilisation
python streamlit	Développer l'application web
python pandas et numpy	Traitement des données
Sqlalchemy et pymysql	La connexion du serveur mysql
plotly express	Visualisation des données par les graphs
Folium, geopandas	Visualisation par les position géographique

3.2.3 L'explication de la source code

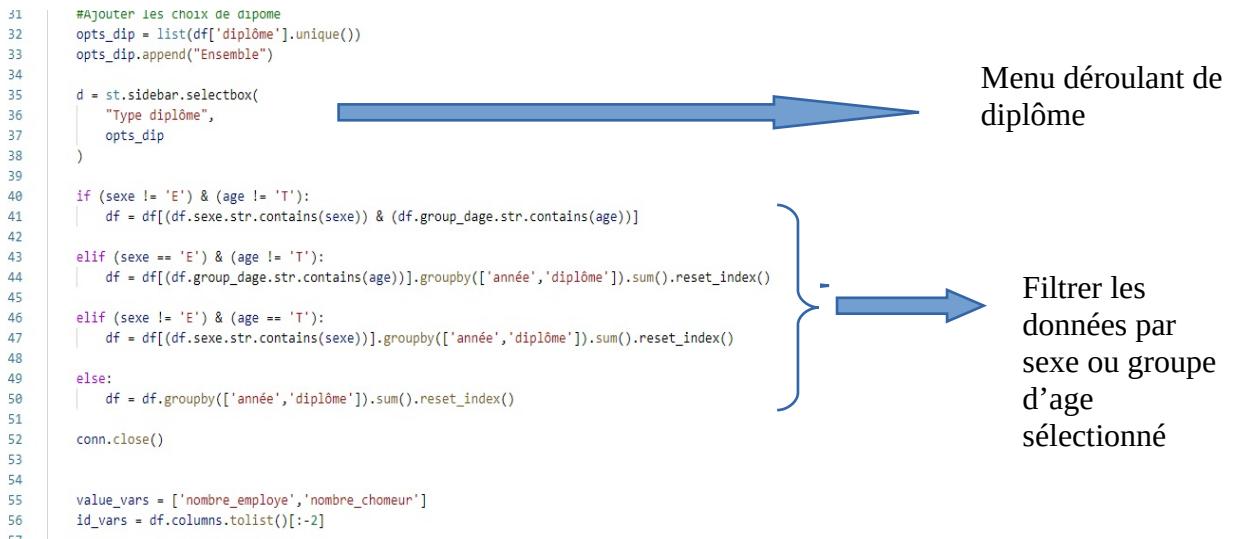
On considère l'indicateur diplôme dont l'interface streamlit de diplôme et voici quelques exemples de la source code.

- ✗ Note : Seules quelques parties de la source code sont présentes. Pour consulter la source code complète diriger le lien https://github.com/SheharaMohamed/chef_doeuvre-levolution_demploi.

- ✓ Importation des librairies et récupération des données de serveur mysql.



- ✓ Filtrer les données selon les choix d'utilisateur



✓ Créer une courbe avec plotly

```
58     if d != "Ensemble":  
59         display_df = pd.melt(df[df.diplôme == d], id_vars = id_vars, value_vars = value_vars,  
60                               var_name = "type_de_nombr", value_name = "nombre")  
61         fig1 = px.line(display_df, x='année', y='nombre', facet_col = "type_de_nombr")  
62         # fig2 = px.line(display_df, x='année', y=['nombre_chomeur'])  
63  
64         fig1.update_layout(showlegend=False,  
65                             title_x=0.5,  
66                             xaxis_title='année',  
67                             yaxis_title='Nombre (millier)',  
68                             width = 1400,  
69                             height = 400)  
70         fig1.for_each_annotation(lambda a: a.update(text=a.text.split("=")[1]))  
71         # fig1.for_each_trace(lambda t: t.update(name=t.name.split("=")[1]))
```

✓ Créer un histogramme et pie chart avec plotly

```
112     fig3 = px.bar(df_temp, x="sexe", y="nombre_employé",  
113                     color='group_dage', barmode='group')  
114  
115     fig3.update_layout(showlegend = True,  
116                         legend=dict(  
117                             title = 'Groupe d\'âge'  
118                         ),  
119                         title_x=0.5,  
120                         xaxis_title='sexe',  
121                         yaxis_title='Nombre d\'emploi (millier)')  
122     fig3.for_each_annotation(lambda a: a.update(text=a.text.split("=")[1]))  
123  
124     fig4 = px.pie(df_temp, names = 'sexe', values = 'nombre_employé') #, color_discrete_sequence=px.colors.sequential  
125  
126     fig4.update_layout(showlegend = True,  
127                         legend = dict(  
128                             title = 'Sexe'  
129                         ))
```

✓ organisation des éléments (les titres, les paragraphes et les graphes)

```
156     st.header("L'évolution annuelle des emplois et des chômeurs")  
157     st.plotly_chart(fig1)  
158  
159     st.header("Nombre d'emploi de l'année {}".format(a))  
160     col_fig3, col_fig4 = st.columns((1,1))  
161     col_fig3.subheader("Par groupe d'âge")  
162     col_fig3.plotly_chart(fig3)  
163     col_fig4.subheader("Part des hommes et des femmes")  
164     col_fig4.plotly_chart(fig4)  
165  
166     st.header("Nombre de chômeur de l'année {}".format(a))  
167     col_fig5, col_fig6 = st.columns((1,1))  
168     col_fig5.subheader("Par groupe d'âge")  
169     col_fig5.plotly_chart(fig5)  
170     col_fig6.subheader("Part des hommes et des femmes")  
171     col_fig6.plotly_chart(fig6)
```

✓ Folium choropleth carte sur streamlit

```
112     #La carte population  
113     m_pop = folium.Map(location=[46.71109 ,1.7191036], zoom_start=5)  
114  
115     folium.Choropleth(  
116         geo_data=display_df,  
117         name="choropleth",  
118         data=display_df,  
119         columns=[code,"population"],  
120         key_on="feature.properties."+code,  
121         threshold_scales = "population",  
122         fill_color="YlOrRd", |  
123         fill_opacity=0.7,  
124         line_opacity=.1,  
125         legend_name=leg,  
126         ).add_to(m_pop).geojson.add_child(  
127             folium.features.GeoJsonTooltip(fields = [nom, 'population'])  
128     )
```

- ✓ La source code principale qui gère toutes les interfaces.

```

1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 import diplôme
5 import catégorie
6 import caractéristique
7 import géographique
8 from modules_emploi import exec
9
10 from sqlalchemy import create_engine
11 # from sqlalchemy.orm import sessionmaker
12 # from sqlalchemy import create_engine, MetaData, Table, Column, Integer, String
13 import pymysql
14
15 st.set_page_config(layout="wide")
16
17 nav = st.sidebar.radio(
18     "Choisissez l'indicateur",
19     ["Diplôme", "Catégorie Socioprofessionnelle", "Caractéristique d'emploi", "Position Géographique"]
20 )
21
22 engine = create_engine(
23     "mysql+pymysql://root:tashe1129@localhost/emploi?charset=utf8mb4&binary_prefix=true"
24 )
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```



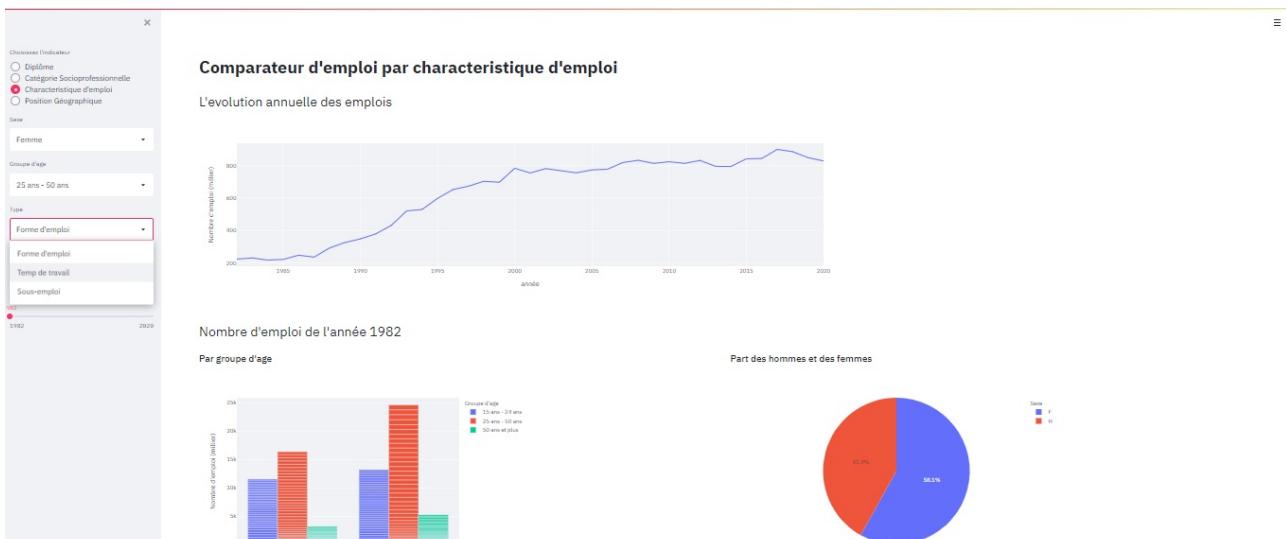
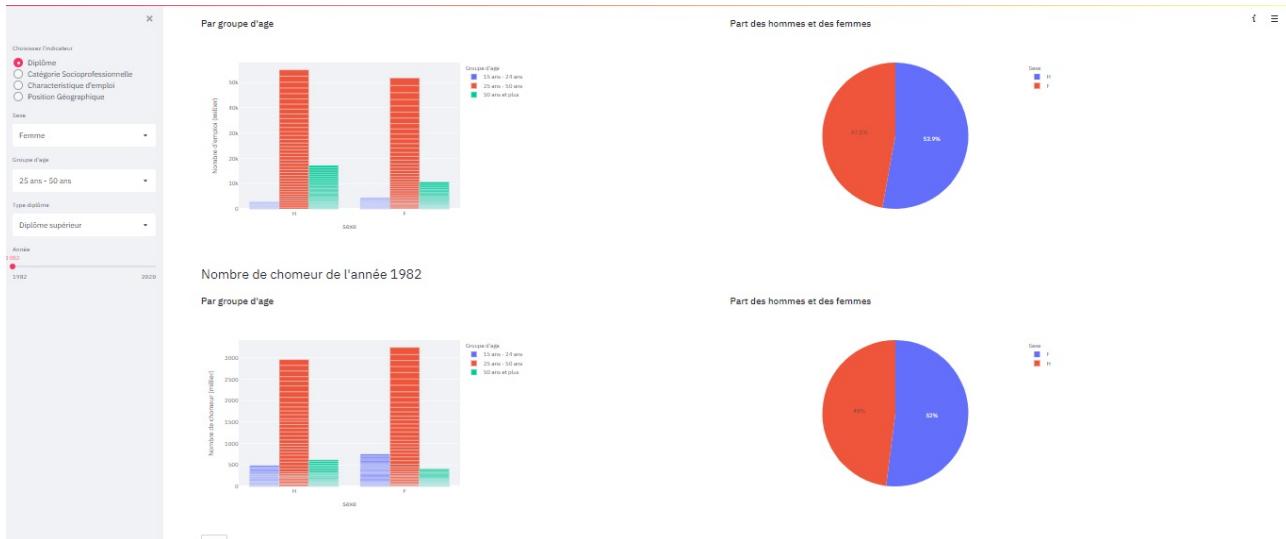
Les quatre indicateurs comme des bouton radio

- ✓ Création des menu déroulants d'age , de sexe et appelle des fonctions d'interfaces.

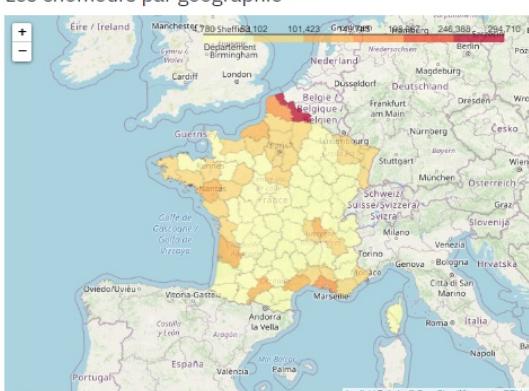
```

25 s = st.sidebar.selectbox(
26     "Sexe",
27     ["Femme", "Homme", "Ensemble"],
28     index = 0
29 )
30
31 sexe = dict(zip(["Femme", "Homme", "Ensemble"],['F','H','E']))
32 age = dict(zip(["0 ans - 14 ans","15 ans - 24 ans", "25 ans - 50 ans", "50 ans et plus", "Ensemble"],['T0','T15','T25',
33
34 if nav == "Diplôme":
35     g = st.sidebar.selectbox(
36         "Groupe d'âge",
37         ["15 ans - 24 ans", "25 ans - 50 ans", "50 ans et plus", "Ensemble"],
38         index = 1
39     )
40     st.title("Comparateur d'emploi par diplôme")
41     diplôme.interface_diplôme(age[g],sexe[s])
42
43 elif nav == "Catégorie Socioprofessionnelle":
44     g = st.sidebar.selectbox(
45         "Groupe d'âge",
46         ["15 ans - 24 ans", "25 ans - 50 ans", "50 ans et plus", "Ensemble"],
47         index = 1
48     )
49     st.title("Comparateur d'emploi par catégorie socioprofessionnelle")
50     catégorie.interface_catégorie(age[g],sexe[s])
51
52 elif nav == "Caractéristique d'emploi":
53
54     g = st.sidebar.selectbox(
55         "Groupe d'âge",
56         ["15 ans - 24 ans", "25 ans - 50 ans", "50 ans et plus", "Ensemble"],
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
302
303
304
305
305
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
13
```

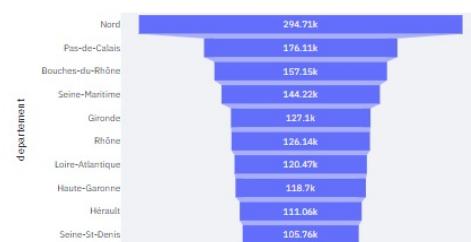
Ci-dessous il y a quelques images de l'application.



➤ Les chomeurs par géographie



Plus touché par les chomeurs



La salaire moyen



4. Ressources

- Les ressources humaines:
 - ✗ Mes formateurs,
 - Manel Boummaiza
 - Josselin Tobblem
 - David Azzaria
 - Nicolas Zanforlini
 - Assia M'Jalled (Prof. Français)
 - ✗ Mon mentor de la formation Augustin Ador – Lead Data Scientiste chez Dataiku
- Les ressources matérielles :
 - ✗ Stack overflow en cas de questions
 - ✗ Python documentation pour toutes types des libraires
 - ✗ Autres recherches internet

5. Les axes d'amélioration et les difficultés

Les axes d'amélioration :

Une analyse pour illustrer l'évolution future du marché de l'emploi - Une prédition utilisant toutes ces données serait pertinente pour conclure ce projet et tirer des informations utiles.

NLP - Un processus NLP mettra ce projet plus en valeur. Il permettra de :

- De scrapper, par indicateur, les avis et commentaires sur les emplois disponible sur linkedin
- De consulter les données sur des potentielles formations

L'apparence de l'application web (streamlit) - Dans ce projet l'application a été crée seulement grâce aux widgets autorisés par streamlit (Streamlit ne peut pas ajouter de menus, ni réinitialiser une page). Mais dans ce projet j'ai pu faire les menus et les nouvelles fenêtres avec les widgets disponibles. S'il est possible d'améliorer l'apparence pour s'adapter aux utilisateurs, ça sera un avantage de le présenter de façon plus intuitive.

Les difficultés :

La durée - J'ai sous-estimé la durée de travail avec le nombre d'étapes qu'on doit faire car l'appréhension des techniques je n'ai pas trouvé difficile. Mais la quantité de travail qu'on devait effectuer était nombreux.

Le choix du projet – Pour faire le choix de mon projet m'a pris quelques semaines. J'avais cette idée de ce projet dès le départ. Mais j'ai pris de temps pour savoir s'il y a des données similaire pour chaque indicateurs et si ces données pouvaient sortir une bonne analyse.

La nettoyage des données – J'ai reçu plus de vingt fichiers comme des données brutes. Et toutes ces données été agrégées par plusieurs attributs. Pour séparer toutes ces données et développer une base de données étaient un vrai défi pour moi.

6. Remerciements

J'aimerais commencer par remercier ma famille pour m'avoir motivé à atteindre mon but de ma vie et ensuite je remercie Simplon pour m'avoir donnée cette opportunité de suivre la formation développeur/se Data.

Ensuite je voudrais remercier mes formateurs pour m'avoir donné les connaissances dans le domaine de data et d'avoir aidé pour ce projet. Je remercie également mon professeur de français d'avoir aidé pour l'avancement de la langue et mon mentor qui m'a suivi et aidé pour mes démarches et avancements pendant toute la durée de la formation.

Je remercie mes collègues qui ont partagé leurs connaissances avec moi et enfin je remercie l'équipe administrative qui m'a aidée à avancer dans mes démarches administratives.