



INSTITUE NATIONAL DES POSTES ET TÉLÉCOMMUNICATION

Projet du 'Web Scraping' Sur le site Avito

Réalisé par :
ADDI Kamal
HOSSAM Hiba

ENCADRÉ PAR :
MR.LAANAYA HICHAM

1^{er} avril 2020

Résumé

Le web scraping (parfois appelé harvesting) est une technique d'extraction du contenu de sites Web, via un script ou un programme, dans le but de le transformer pour permettre son utilisation dans un autre contexte. L'objectif de notre projet est donc d'extraire le contenu des pages du site 'Avito' de façon structurée et faire une annalyse et une visualisation de ces données.

Table des matières

1	Appliquer le 'Web Scraping' sur le site Avito :	1
1.1	Afficher le code source de la page :	1
1.2	Comment extraire les données ?	2
1.3	Le Code Python :	2
2	Exploitation des données :	6
2.1	Analyse des données :	6
2.2	Visualisation :	12

Chapitre 1

Appliquer le 'Web Scraping' sur le site Avito :

1.1 Afficher le code source de la page :

le principe est de chercher les mots clés des balises ou/et des classes dans le code source de la page après avoir inspecter la partie visée(cf. fig. 1.1)

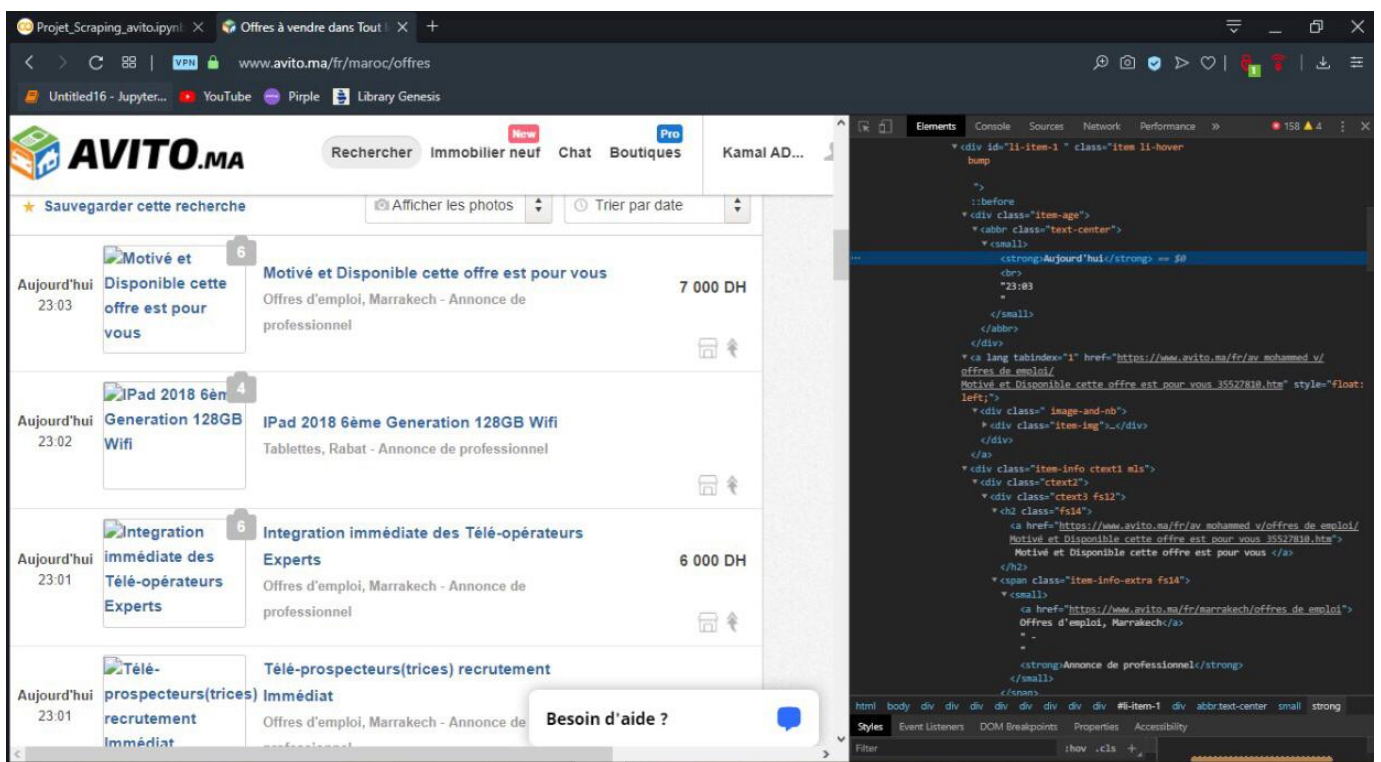


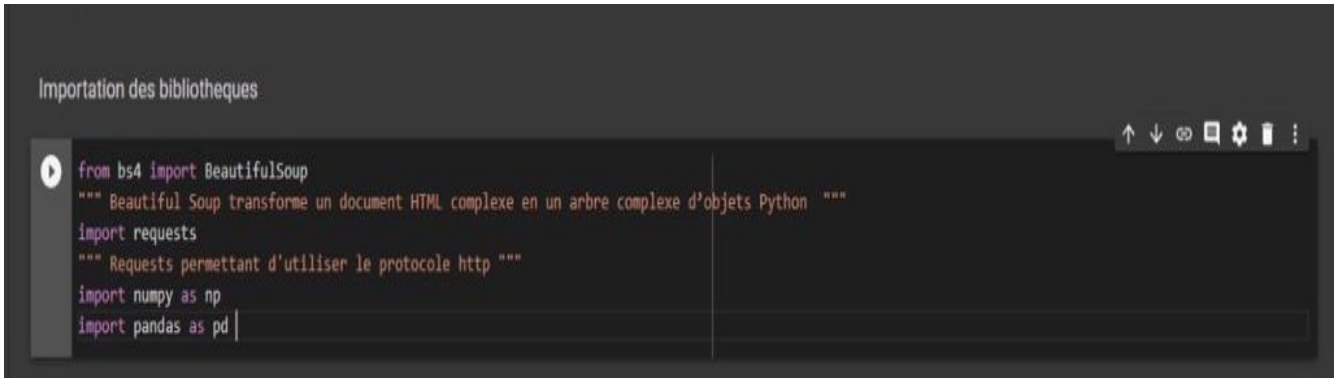
FIGURE 1.1 – le code 'html' de la page

1.2 Comment extraire les données ?

En utilisant des fonctions prédéfinies sur python comme (findAll, find,...) et les mots clés se trouvant dans le code source on peut extraire les données utiles comme (le prix des articles, la date de l'annonce,)

1.3 Le Code Python :

on commence par importer les bibliothèques (cf. fig. 1.2) :



```
Importation des bibliotheques

from bs4 import BeautifulSoup
""" BeautifulSoup transforme un document HTML complexe en un arbre complexe d'objets Python """
import requests
""" Requests permettant d'utiliser le protocole http """
import numpy as np
import pandas as pd
```

FIGURE 1.2 – les bibliothèques

Puis on commence notre code !



```
Scraper les annonces d'offres :

[32] """***** Scraper les annonces d'offres : *****"""
    liste_prices=[]
    liste_titles=[]
    liste_dates=[]
    liste_types=[]
    liste_categories=[]
    liste_villes=[]
    #scraper 250 pages d'annonces d'offres
    for i in range(1,250):
```

FIGURE 1.3 – déclaration des listes

Ces listes vont contenir les données de chaque article dans le site

Après on commence à utiliser les fonctions de la bibliothèque BeautifulSoup : findAll(balise,attrs="class"="nom de la class").text pour extraire le contenu sans oublier le package requests qui permet de comprendre les balises html

```

for i in range(1,250):
    # se connecter au site et obtenir le code source html
    url1 = "https://www.avito.ma/fr/maroc/offres?o={}".format(i)
    info1=requests.get(url1)
    # utiliser le package BeautifulSoup
    # qui "comprend" les balises html
    soup = BeautifulSoup(info1.text)      #mtnt je peux faire un recherche sur mes elements dans l'objet soup

    # scraper les titres
    titles=soup.findAll('h2',attrs={ "class" : "fs14" })
    for t in titles:
        if t.text != None :
            liste_titles.append(t.text)      #ajouter le titre a la liste des titres
        else:
            liste_titles.append(np.nan)

    # scraper les prix
    prices=soup.findAll('span',attrs={ "class" : "price_value" })
    for p in prices:
        prix=p.text
        prix=prix.replace(" ", "")      #supprimer l'espace dans la chaîne de caractère
        if prix!='\xa0':      # c.a.d le prix existe
            liste_prices.append(int(prix))      # convertir le prix en entier et l'ajouter à la liste des prix
        else:
            prix=np.nan      # remplacer les valeurs manquante par nan
            liste_prices.append(prix)

    # scraper les dates

```

FIGURE 1.4

```

# scraper les dates
dates=soup.find('div', {'class' : 'listing listing-thumbs'}).findAll("abbr","class"=="text-center")
for d in dates:
    date = d.find("strong")
    # verifier l'existence de la date (avec != None)
    if date != None:
        # modifier la date d'aujourd'hui en format day-month
        if date.text=="Aujourd'hui":
            date_auj=date.text.replace("Aujourd'hui","30 Mar")
            liste_dates.append(date_auj)
        # modifier la date d'hier en format day-month
        elif date.text=="Hier":
            date_hier=date.text.replace("Hier","29 Mar")
            liste_dates.append(date_hier)
        # les autres dates sont sous le format demander
        else:
            liste_dates.append(date.text)
    # si le champ de la date est null on met la valeur nan
    else:
        liste_dates.append(np.nan)

# scraper le type
types=soup.find('div', {'class' : 'listing listing-thumbs'}).findAll('span',attrs={ "class" : "item-info-extra fs14" })
for t in types:
    t=t.find("strong")
    if t!=None:
        liste_types.append(t.text)      # ici l'annonce est professionnelle
    else:
        # les autres annonces sont particulières
        liste_types.append("Annonce de particulier")

```

FIGURE 1.5

```

# scraper le type
types=soup.find('div', {'class' : 'listing listing-thumbs'}).findAll('span',attrs={ "class" : "item-info-extra fs14" })
for t in types:
    t=t.find("strong")
    if t!=None:
        liste_types.append(t.text)      # ici l'annonce est professionnelle
    else:
        # les autres annonces sont particuliers
        liste_types.append("Annonce de particulier")

# scraper les villes et categories
ville_categ=soup.find('div', {'class' : 'listing listing-thumbs'}).findAll('span',attrs={ "class" : "item-info-extra fs14" })
for vc in ville_categ:
    vc=vc.find("a")
    if vc!=None:
        l1=vc.text.split(",")
        liste_categories.append(l1[-2])      # extraire le categorie
        # extraire la ville
        liste_villes.append(l1[-1].replace(" ",""))
        # replace() remplacer l'espace qui precede la chaîne de caractère ville par "" (on a trouver toutes les villes commencent avec un espace ' ra
    else:
        # si le champ {ville et categorie} est vide on ajoute la valeur nan
        liste_categories.append(np.nan)
        liste_villes.append(np.nan)

```

FIGURE 1.6

On crée le dataframe qui va représenter les informations de chaque article(cf. fig. 1.8) de la manière suivante :

```

+ Code + Texte
# représenter les données dans un dataframe df_offres
import numpy as np
import pandas as pd
liste_offres = ["Offre" for i in range(len(liste_prices))]
ar_offres = np.array([ liste_titles , liste_prices , liste_dates , liste_types , liste_categories , liste_villes , liste_offres ]).T
rows_offres = [ "Titres" , "Prix" , "Dates" , "Types" , "Categories" , "Villes" , "Offres/Demande" ]
ind_offres = [ "article {}".format(i) for i in range(1,8716) ]

df_offres = pd.DataFrame(ar_offres,index=ind_offres,columns =rows_offres)

```

FIGURE 1.7

+ Code + Texte

df_offres.head(20)

	Titres	Prix	Dates	Types	Categories	Villes	Offres/Demande
article 1	Recrutement massive de téléprospecteurs	5000	30 Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre
article 2	Lecteur vitesse CD pour DDJ NUMARK CDN 88 MP3	nan	30 Mar	Annonce de professionnel	Pièces et Accessoires pour véhicules	Rabat	Offre
article 3	Téléopérateur en prise de rendez-vous	5000	30 Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre
article 4	Créez rapidement votre entreprise	3500	30 Mar	Annonce de professionnel	Services	Casablanca	Offre
article 5	Nous recrutons en urgence 20 télé-vendeurs	5000	30 Mar	Annonce de professionnel	Offres d'emploi	Casablanca	Offre
article 6	Domicilier votre entreprise à partir de 120 d...	120	30 Mar	Annonce de professionnel	Services	Casablanca	Offre
article 7	III Call recrute en urgence des télé-opérate...	5000	30 Mar	Annonce de professionnel	Offres d'emploi	Casablanca	Offre
article 8	Téléopérateurs pour des sondages / enquêtes	4500	30 Mar	Annonce de professionnel	Offres d'emploi	Casablanca	Offre
article 9	Prise de rdv	4000	30 Mar	Annonce de professionnel	Offres d'emploi	Casablanca	Offre
article 10	Appartement à Fès	1900	30 Mar	Annonce de particulier	Appartements	Fès	Offre
article 11	Motivé et Disponible cette offre est pour vous	7000	30 Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre
article 12	IPad 2018 6ème Generation 128GB Wifi	nan	30 Mar	Annonce de professionnel	Tablettes	Rabat	Offre
article 13	Integration immédiate des Télé-opérateurs Exp...	6000	30 Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre
article 14	Télé-prospecteurs(trices) recrutement Immédiat	8000	30 Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre

FIGURE 1.8

Pour conserver les données et éviter la mise à jour du site on les enregistre dans un fichier csv nommé 'Annonces-offres.csv' (cf. fig. 1.9)

Enregistrer nos données scraper dans un fichier scv

```

# Enregistrer le dataframe dans un fichier .csv |
df_offres.to_csv ('Annonces_offres.csv', index = False, header=True)

```

FIGURE 1.9

Annances_offres.csv - Excel

Chercher des outils adaptés

Partager

Calibri 11

Standard

Mise en forme conditionnelle

Mettre sous forme de tableau

Styles de cellules

Insérer

Supprimer

Format

Trier et Rechercher et filtrer

Édition

B2

5000

	A	B	C	D	E	F	G	H	I
	Titres	Prix	Dates	Types	Catégories	Villes	Offres/Demande		
2	Recrutement massive de tÃ©lÃ©opÃ©rateurs	5000	30-Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre		
3	Lecteur vitesse CD pour DDJ NUMARK CDN 8 nan		30-Mar	Annonce de professionnel	PIÃ©ces et Accessoires pour vÃ©hicul	Rabat	Offre		
4	TÃ©lÃ©opÃ©rateur en prise de rendez-vous	5000	30-Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre		
5	CrÃ©ez rapidement votre entreprise	3500	30-Mar	Annonce de professionnel	Services	Casablanca	Offre		
6	Nous recrutons en urgence 20 tÃ©lÃ©opÃ©-	5000	30-Mar	Annonce de professionnel	Offres d'emploi	Casablanca	Offre		
7	Domicilier votre entreprise Ã partir de 120 d	120	30-Mar	Annonce de professionnel	Services	Casablanca	Offre		
8	Illit Call recrute en urgence des tÃ©lÃ©opÃ©	5000	30-Mar	Annonce de professionnel	Offres d'emploi	Casablanca	Offre		
9	TÃ©lÃ©opÃ©rateurs pour des sondages / e	4500	30-Mar	Annonce de professionnel	Offres d'emploi	Casablanca	Offre		
10	Prise de rdv	4000	30-Mar	Annonce de professionnel	Offres d'emploi	Casablanca	Offre		
11	Appartement Ã FÃ©s	1900	30-Mar	Annonce de particulier	Appartements	FÃ©s	Offre		
12	MotivÃ© et Disponible cette offre est pour v	7000	30-Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre		
13	IPad 2018 6Ã©me Generation 128GB Wifi	nan	30-Mar	Annonce de professionnel	Tablettes	Rabat	Offre		
14	Integration immÃ©diate des TÃ©lÃ©opÃ©	6000	30-Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre		
15	TÃ©lÃ©opÃ©rateurs(trices) recrutement in	8000	30-Mar	Annonce de professionnel	Offres d'emploi	Marrakech	Offre		
16	ContrÃ´leur Audio Bose T1 ToneMatch	nan	30-Mar	Annonce de professionnel	Image & Son	Rabat	Offre		
17	Camera Document Dymo Mimioview ICD03	3500	30-Mar	Annonce de professionnel	Appareils photo et CamÃ©ras	Rabat	Offre		
18	Marguerites 2 Appartement Haut standing G	947000	30-Mar	Annonce de professionnel	Appartements	Marrakech	Offre		
19	Appartement de 90 m2 Autre secteur	810000	30-Mar	Annonce de professionnel	Appartements	Mohammedia	Offre		
20	Appartement de 55 m2 Autre secteur	495000	30-Mar	Annonce de professionnel	Appartements	Mohammedia	Offre		
21	Appartement de 112 m2 FI Alja	840000	30-Mar	Annonce de professionnel	Appartements	Mohammedia	Offre		

Annances_offres

Prêt

110%

12:27 AM 3/31/2020

FIGURE 1.10 – le fichier csv qui contient les données

C'était pour les offres, on fait le même travail sur les demandes, et on enregistre les données dans un fichier nommé 'Annonces-demandes.csv' !

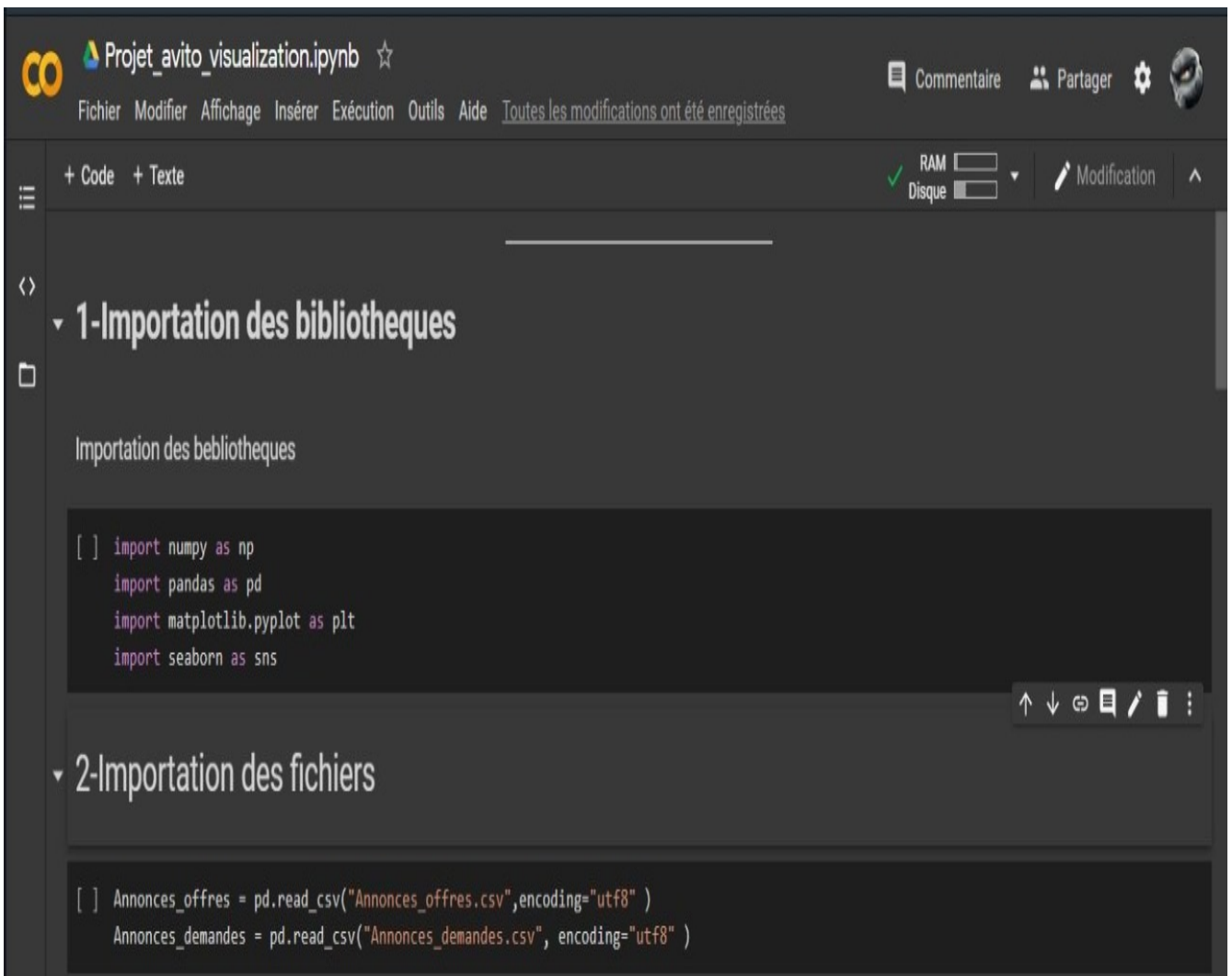
Chapitre 2

Exploitation des données :

Après avoir extraire les données du site Avito, on va les traiter indépendamment du site sous l'aide des fichier csv 'Annonces-offres' et 'Annonces-demandes'

2.1 Analyse des données :

On commence par importer les bibliothèques utiles, puis on fait la lecture des données à partir des fichiers csv enregistrés (cf. fig. 2.1)



The screenshot shows a Jupyter Notebook titled 'Projet_avito_visualization.ipynb'. The interface includes a top bar with navigation links (Fichier, Modifier, Affichage, Insérer, Exécution, Outils, Aide) and a status message 'Toutes les modifications ont été enregistrées'. Below the top bar, there are tabs for '+ Code' and '+ Texte', and a RAM/Disque usage indicator. The notebook content is organized into sections: '1-Importation des bibliotheques' and '2-Importation des fichiers'. The code in section 1 imports numpy, pandas, matplotlib.pyplot, and seaborn. The code in section 2 reads two CSV files: 'Annonces_offres.csv' and 'Annonces_demandes.csv'.

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

[ ] Annonces_offres = pd.read_csv("Annonces_offres.csv",encoding="utf8" )
Annonces_demandes = pd.read_csv("Annonces_demandes.csv", encoding="utf8" )
```

FIGURE 2.1

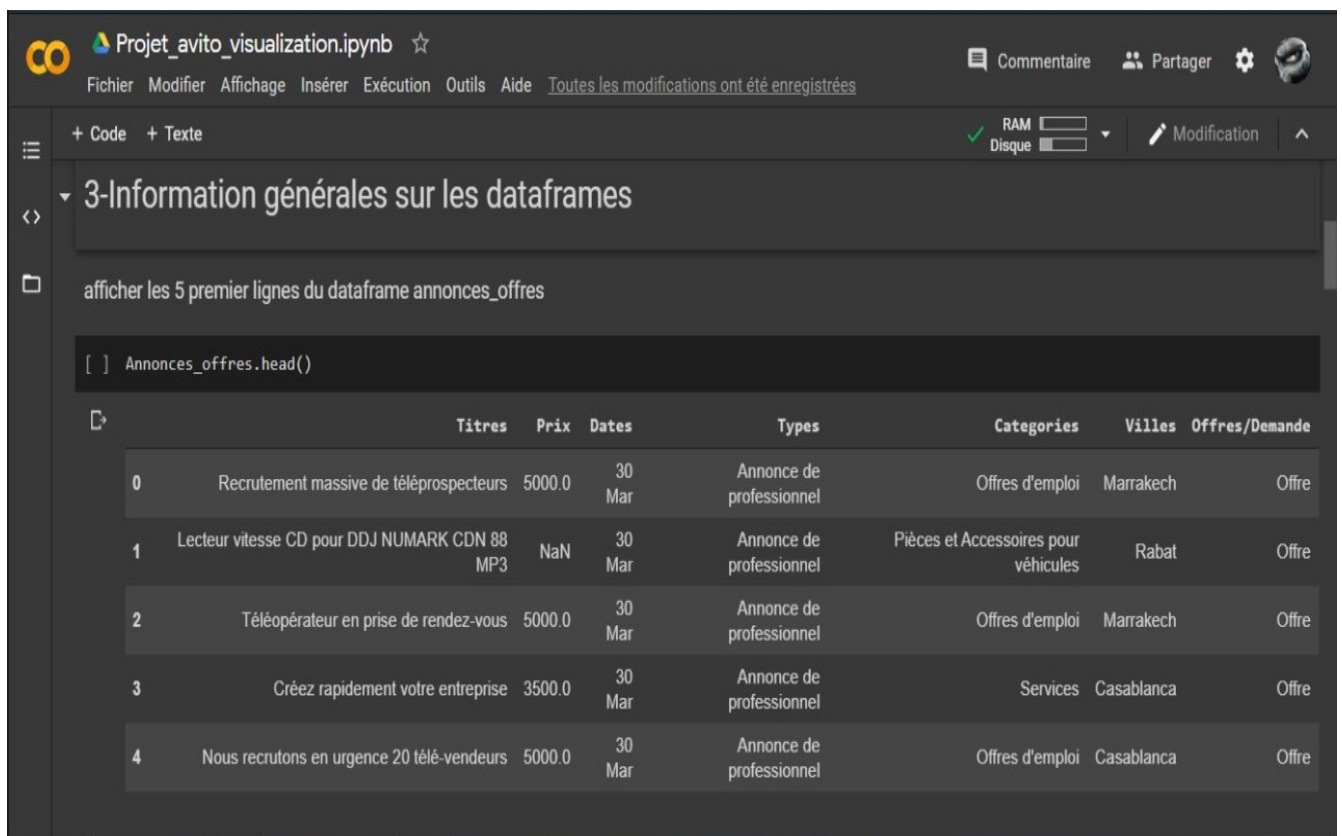


FIGURE 2.2

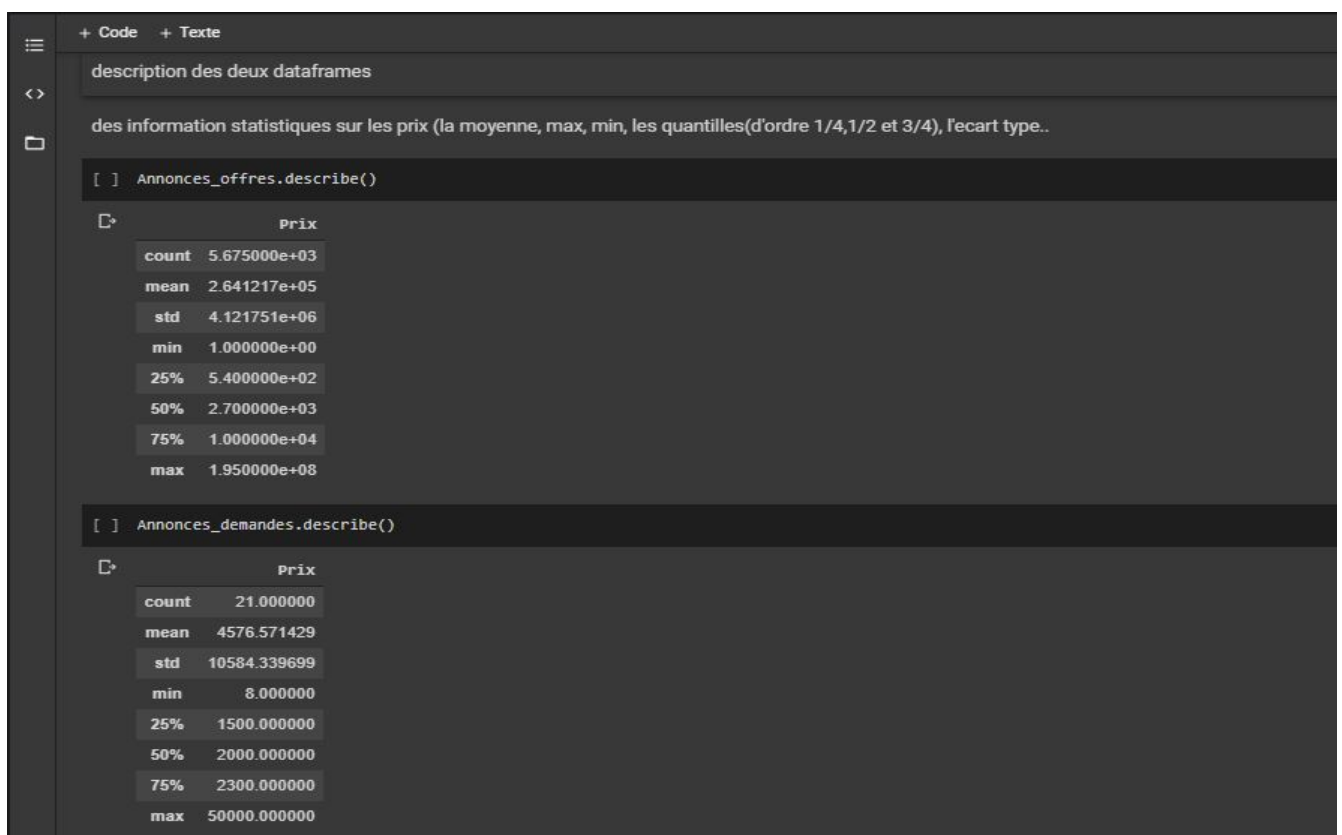


FIGURE 2.3

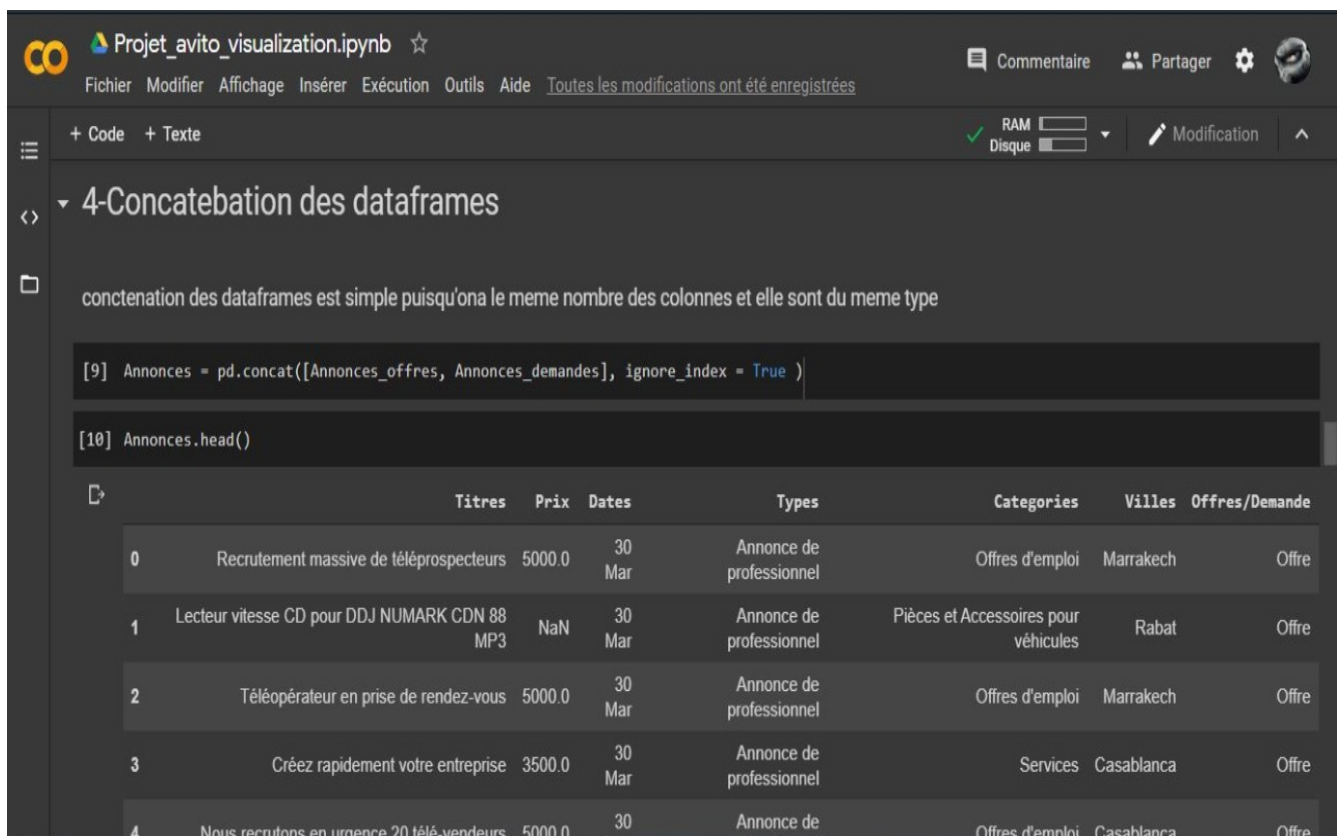


FIGURE 2.4

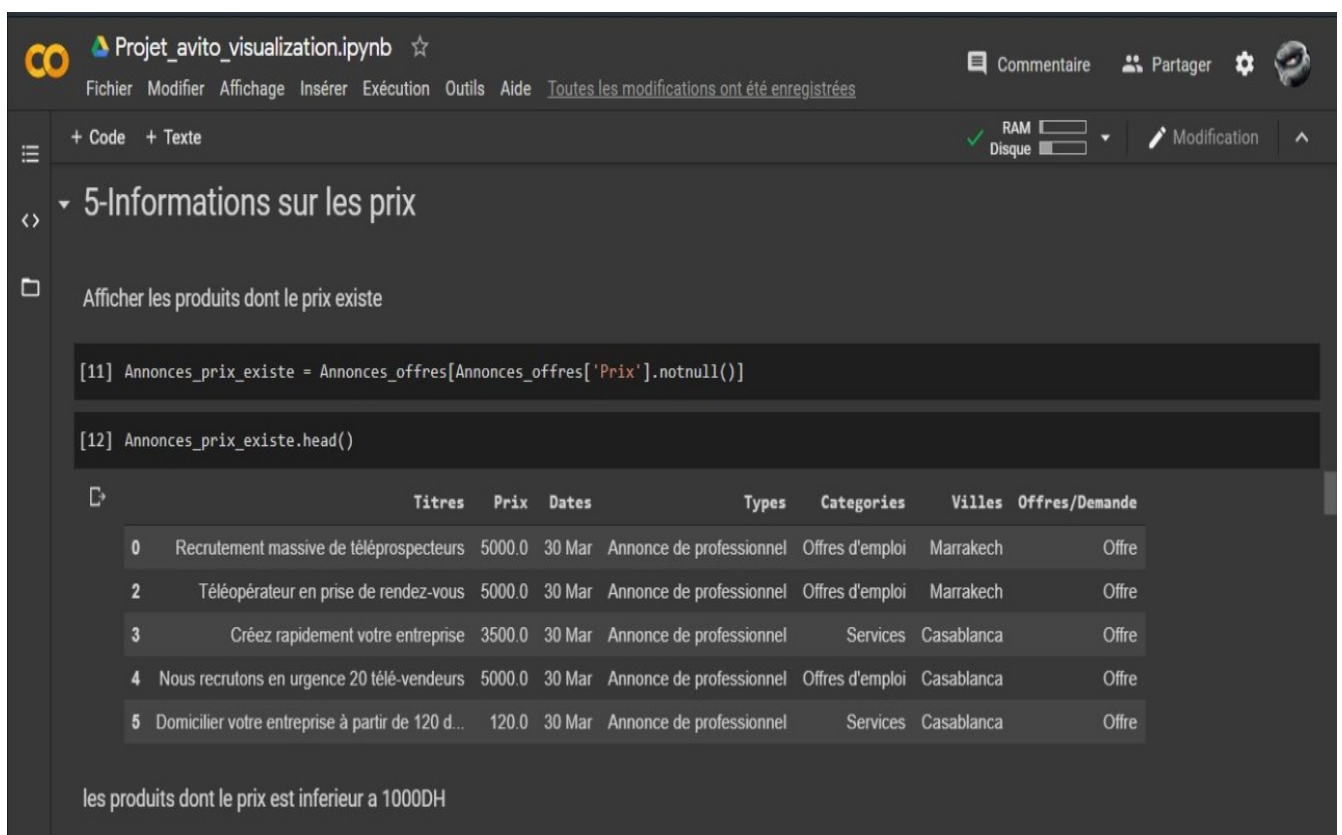


FIGURE 2.5

```

Projet_avito_visualization (1).ipynb
Fichier Modifier Affichage Insérer Exécution Outils Aide Enregistrement...

+ Code + Texte
sommes d'argent en DH par jour

Somme_grouper_par_dates = Annonces.groupby("Dates")["Prix"].sum()

Somme_grouper_par_dates

Dates
1 Déc 270000.0
1 Fév 879463.0
1 Jan 877524.0
1 Mar 3566694.0
10 Fév 9213070.0
...
9 Déc 316548.0
9 Fév 3156734.0
9 Jan 7815221.0
9 Mar 196526880.0
9 Nov 0.0
Name: Prix, Length: 149, dtype: float64

la moyenne par jour

moy = Somme_grouper_par_dates.mean()
print("la moyenne est %.2f DH"%(moy))

la moyenne est 10060315.53 DH

la somme maximale et leur date

somme_max = Somme_grouper_par_dates.max()
date_somme_max = Somme_grouper_par_dates.idxmax() #l'indice de la valeur maximale dans la serie des prix par jour
print("la somme maximale des prix des annonces par jour est :", somme_max)
print("Enregistrer le ", date_somme_max)

la somme maximale des prix des annonces par jour est : 310006819.0
Enregistrer le 10 Mar

```

FIGURE 2.6

```

+ Code + Texte
6-Catégories

le nombre d'annonces de chaque catégories

Serie_categorie = Annonces.groupby("Categories")["Categories"].count().sort_values(ascending=True) #croissante par rapport au nombre de chaque categorie

Serie_categorie.head(10)

Categories
Bateaux 1
Engins Agricole 1
Remorques et Caravanes 1
Autres 1
Magazines 3
Engins BTP 3
Vêtements pour enfant et bébé 4
Voyages et Billetterie 4
Travaux de maison 4
Autre Immobilier 6
Name: Categories, dtype: int64

les 10 categories les plus frequents :

Serie_categorie[-10:]

Categories
Image & Son 256
Services 272
Pièces et Accessoires pour véhicules 292
Appareils photo et Caméras 369
Ordinateurs de bureau 687
Accessoires informatique et Gadgets 736
Ordinateurs portables 759
Appartements 791
Matériels Professionnels 932
Offres d'emploi 1003
Name: Categories, dtype: int64

```

FIGURE 2.7

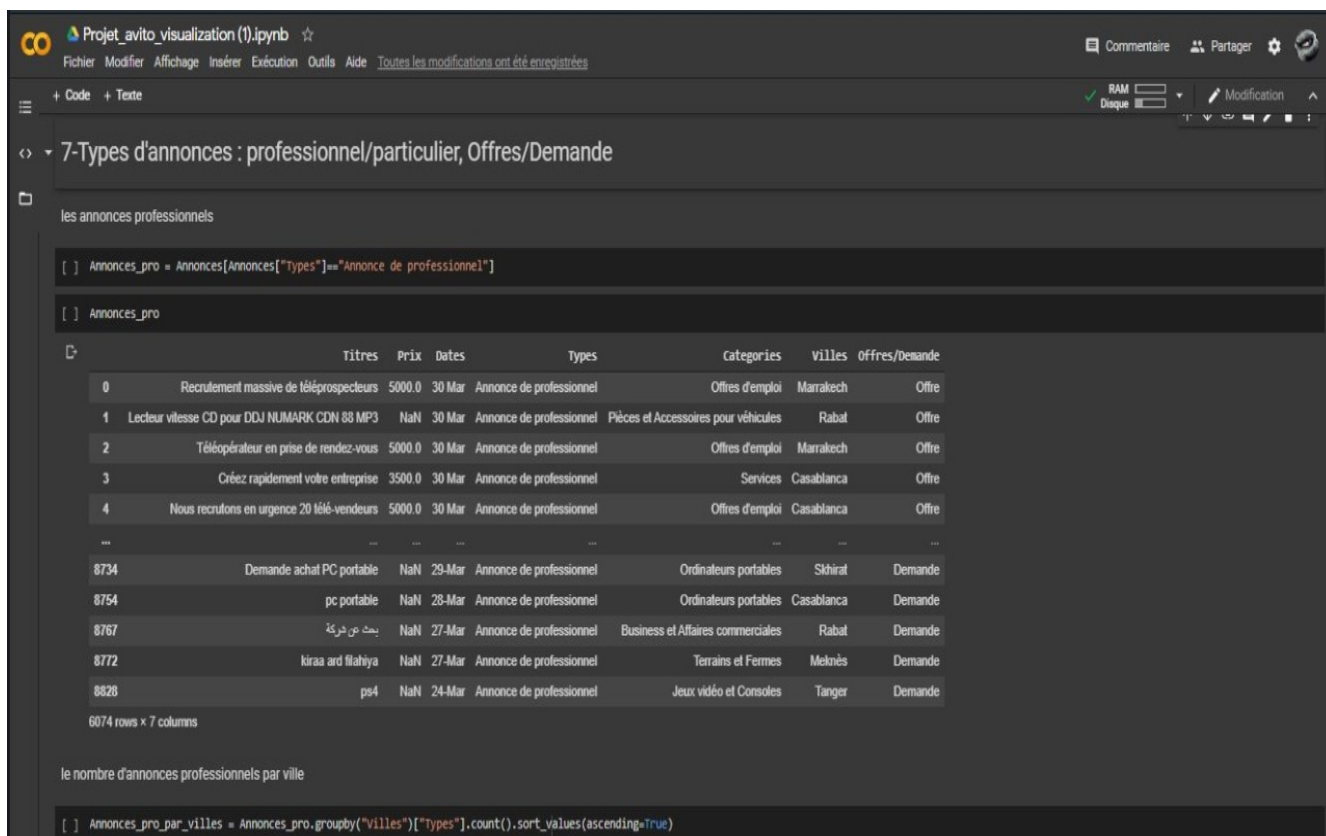


FIGURE 2.8

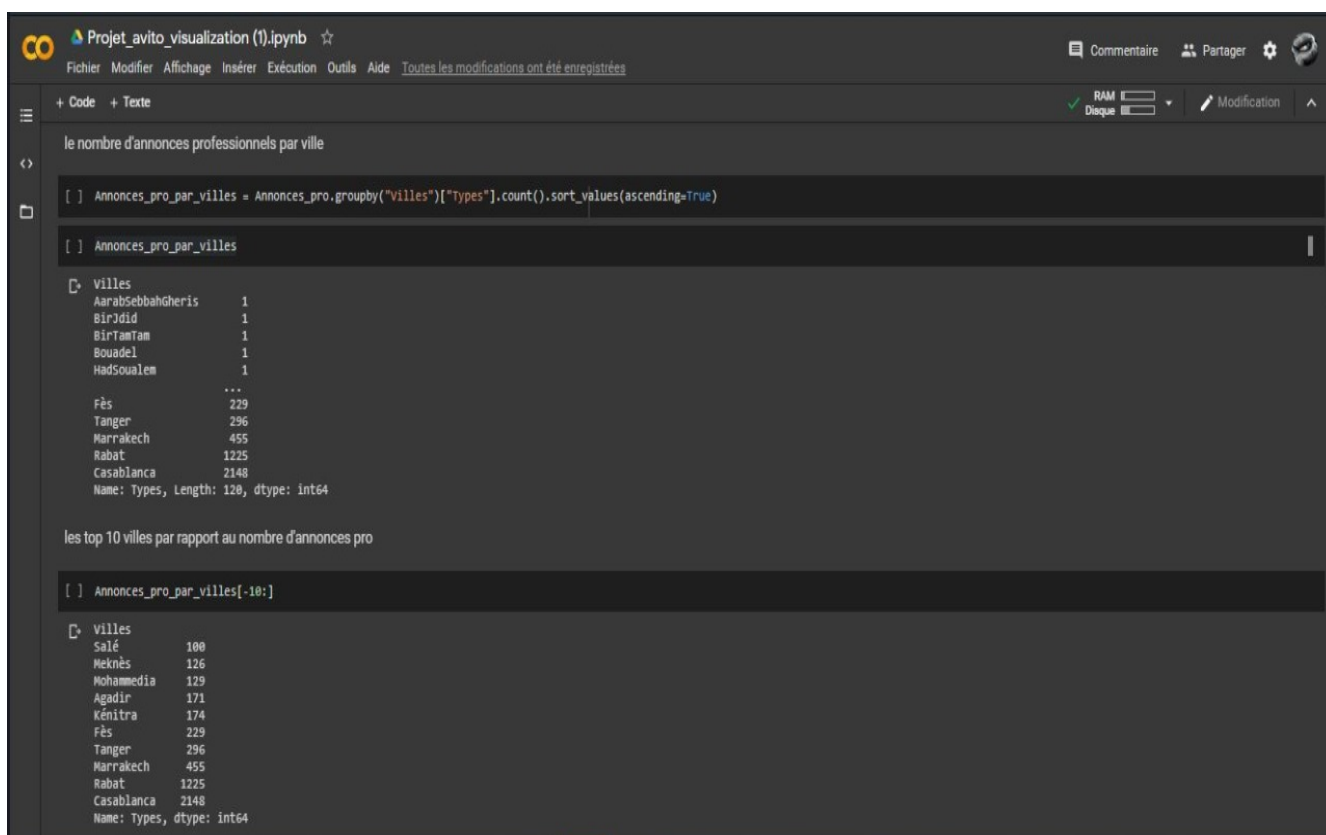


FIGURE 2.9

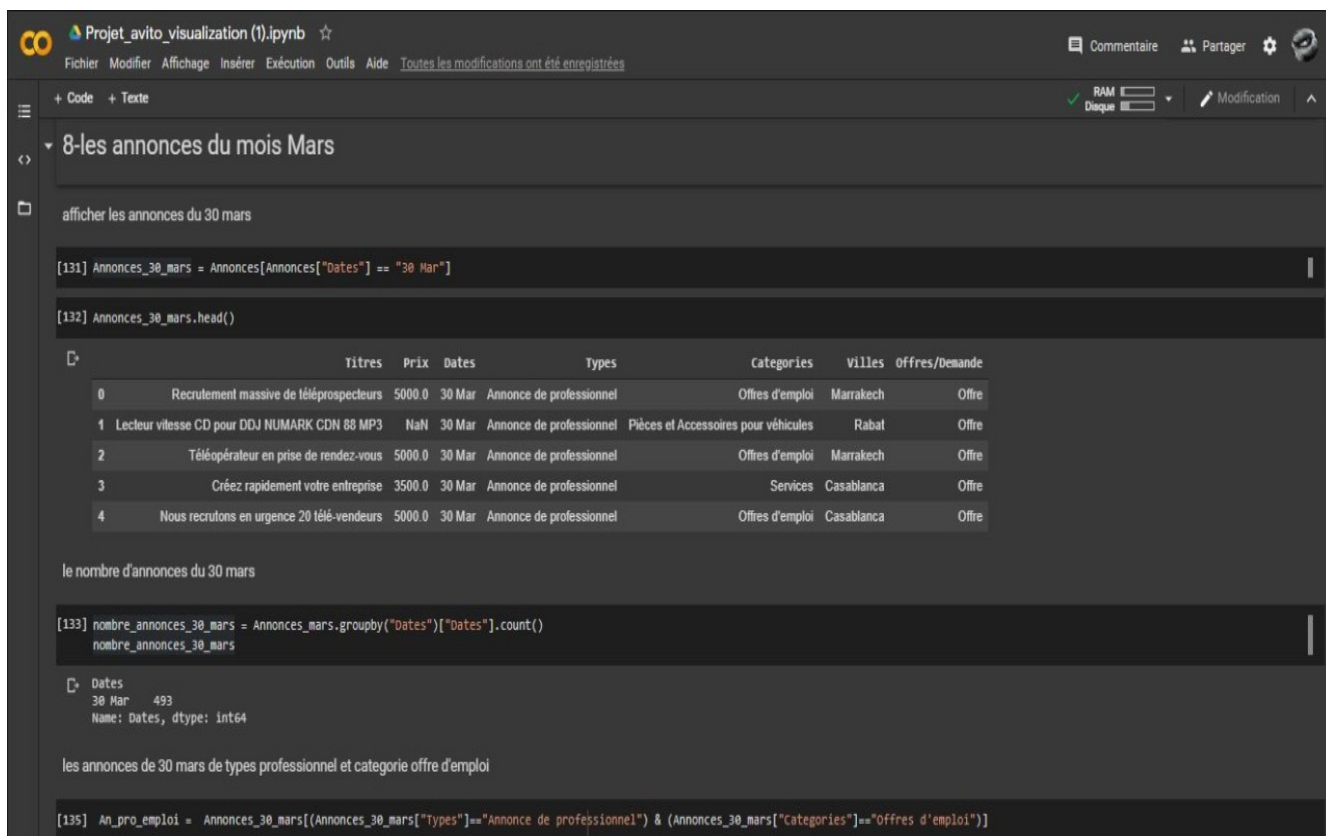


FIGURE 2.10

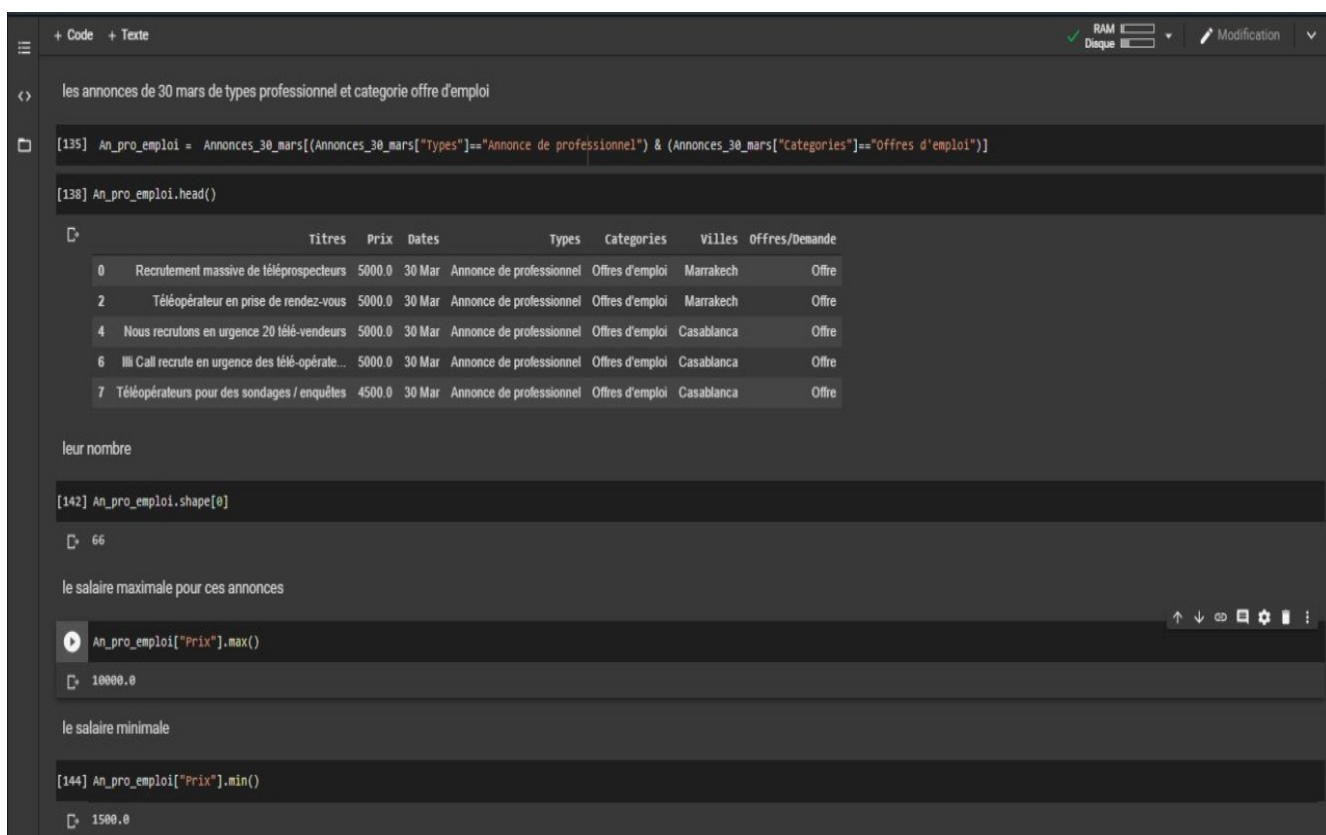


FIGURE 2.11

2.2 Visualisation :

Dans cette partie on va se contenter de metre les résultats après l'exécution du code, puisque c'est la chose la plus importante de cette etude, pour plus de détails veuillez consulter le code Python

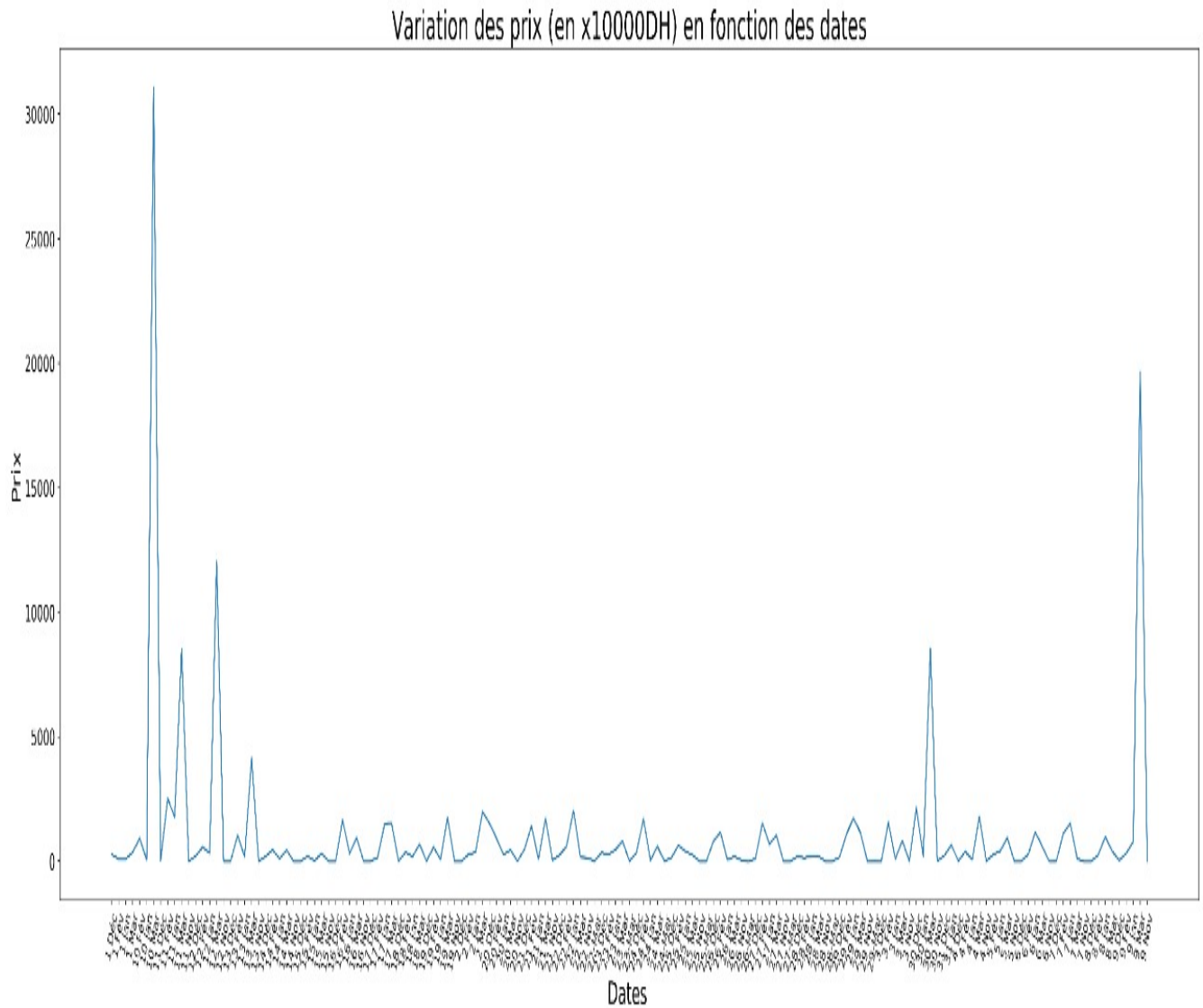


FIGURE 2.12

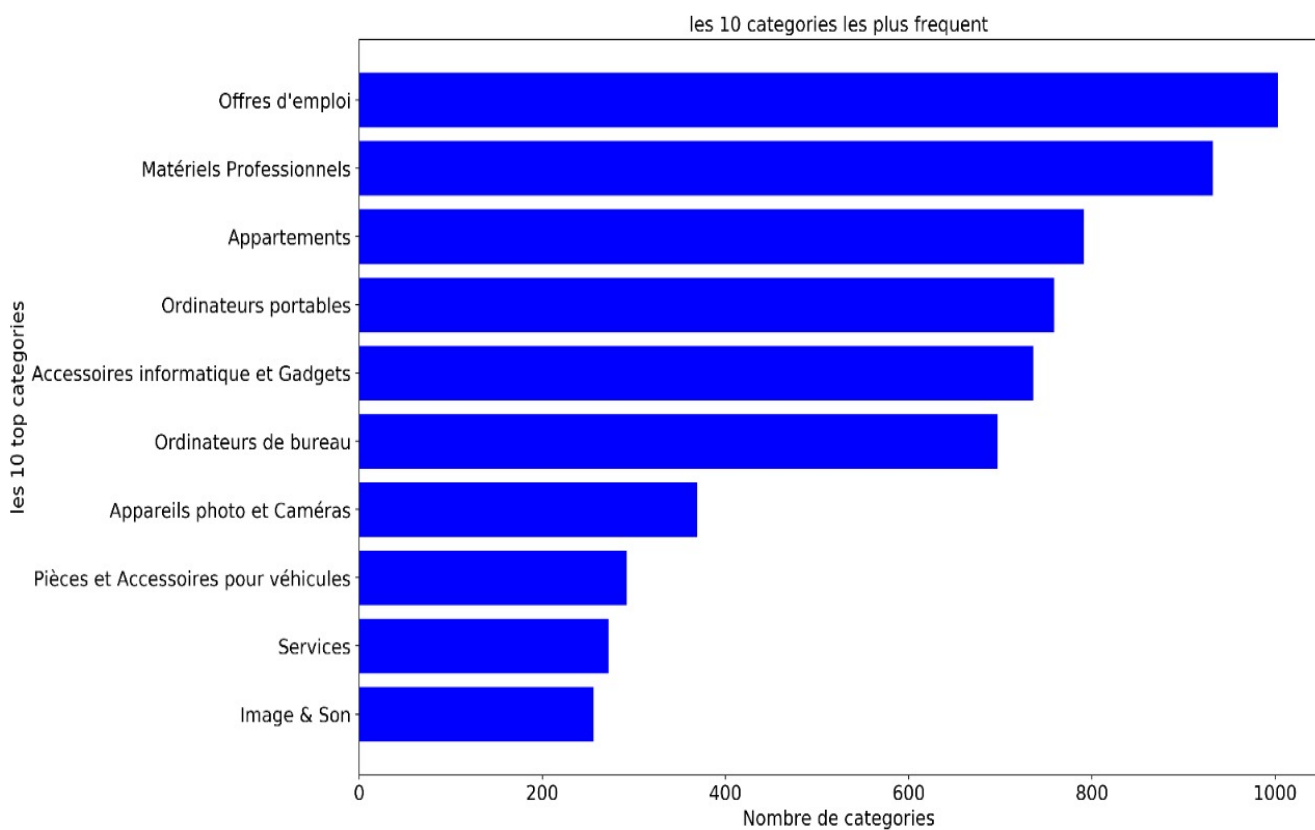


FIGURE 2.13

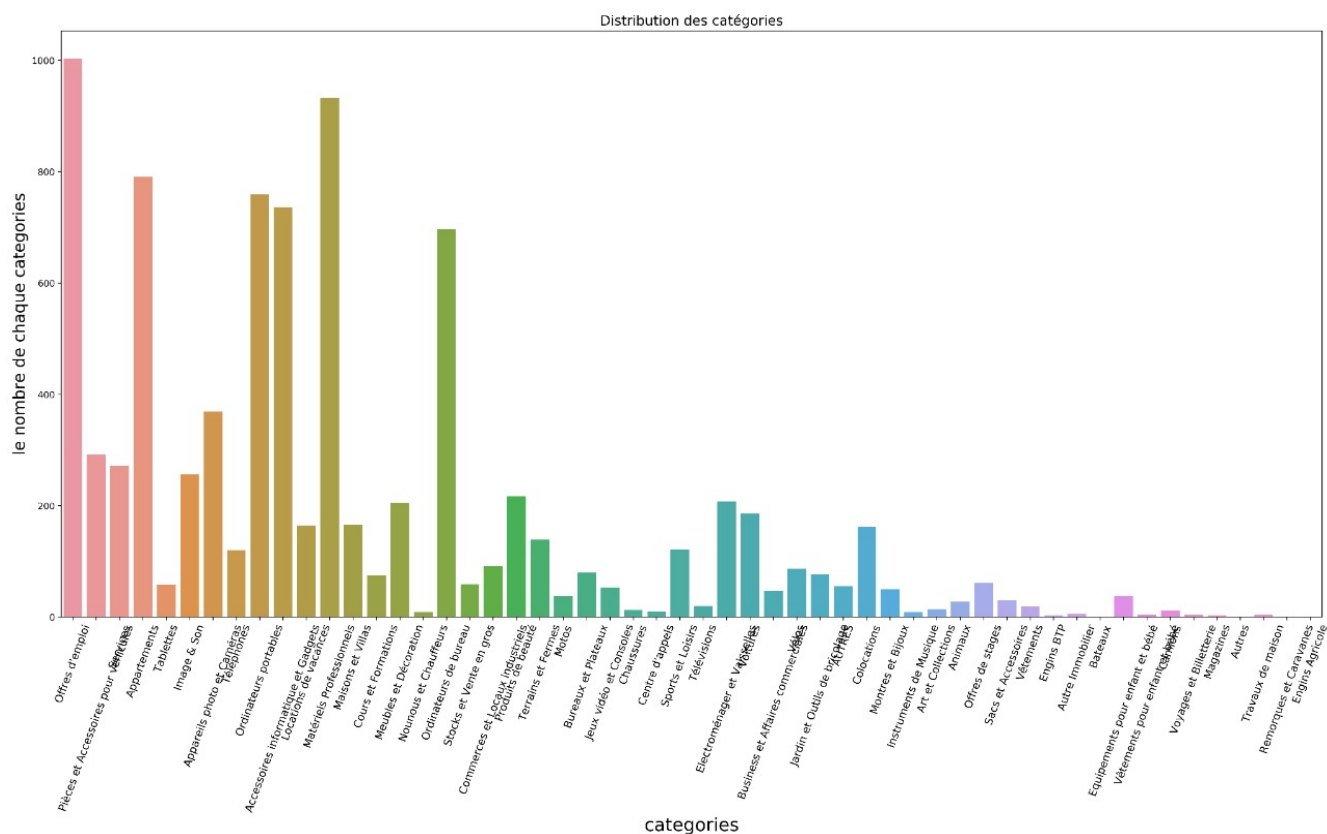


FIGURE 2.14

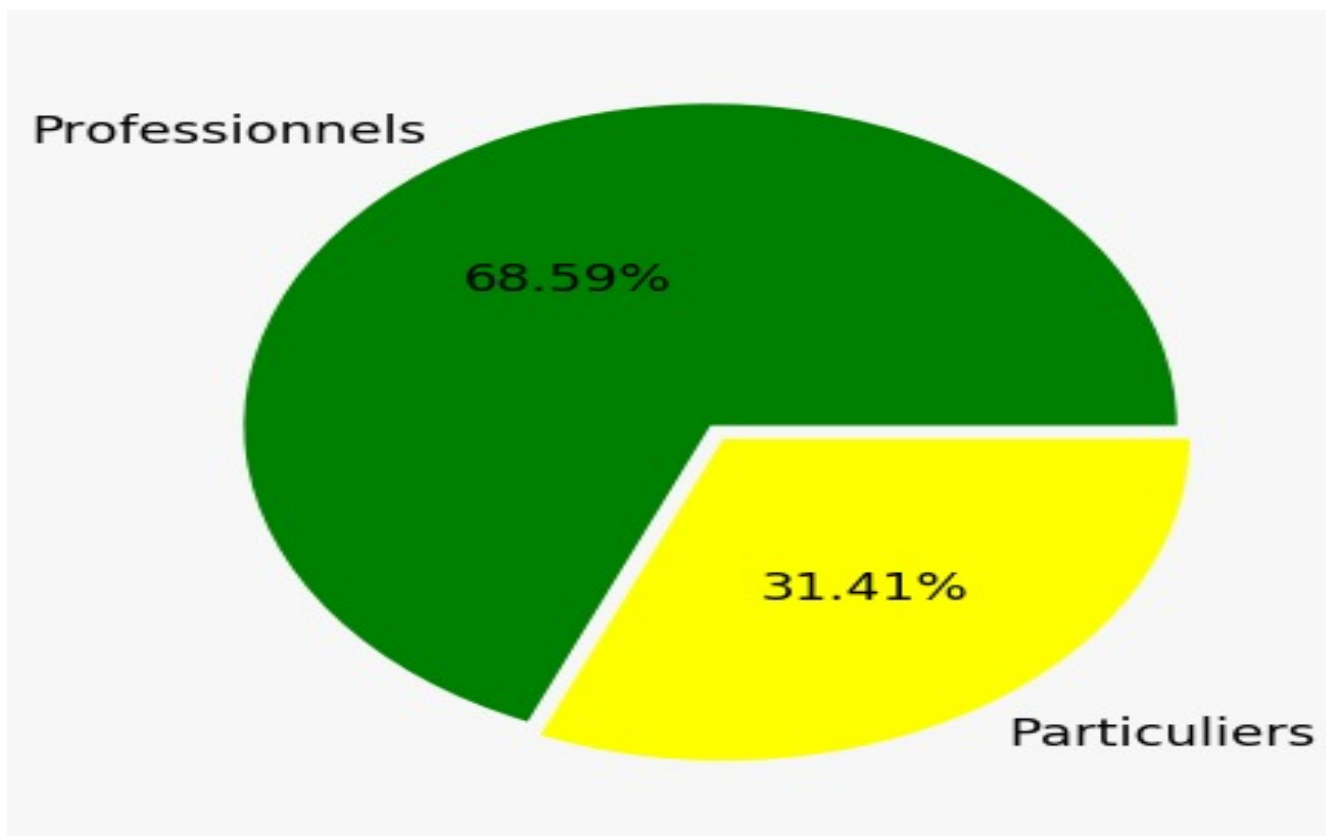


FIGURE 2.17 – Pourcentage des annonces professionnels et particuliers

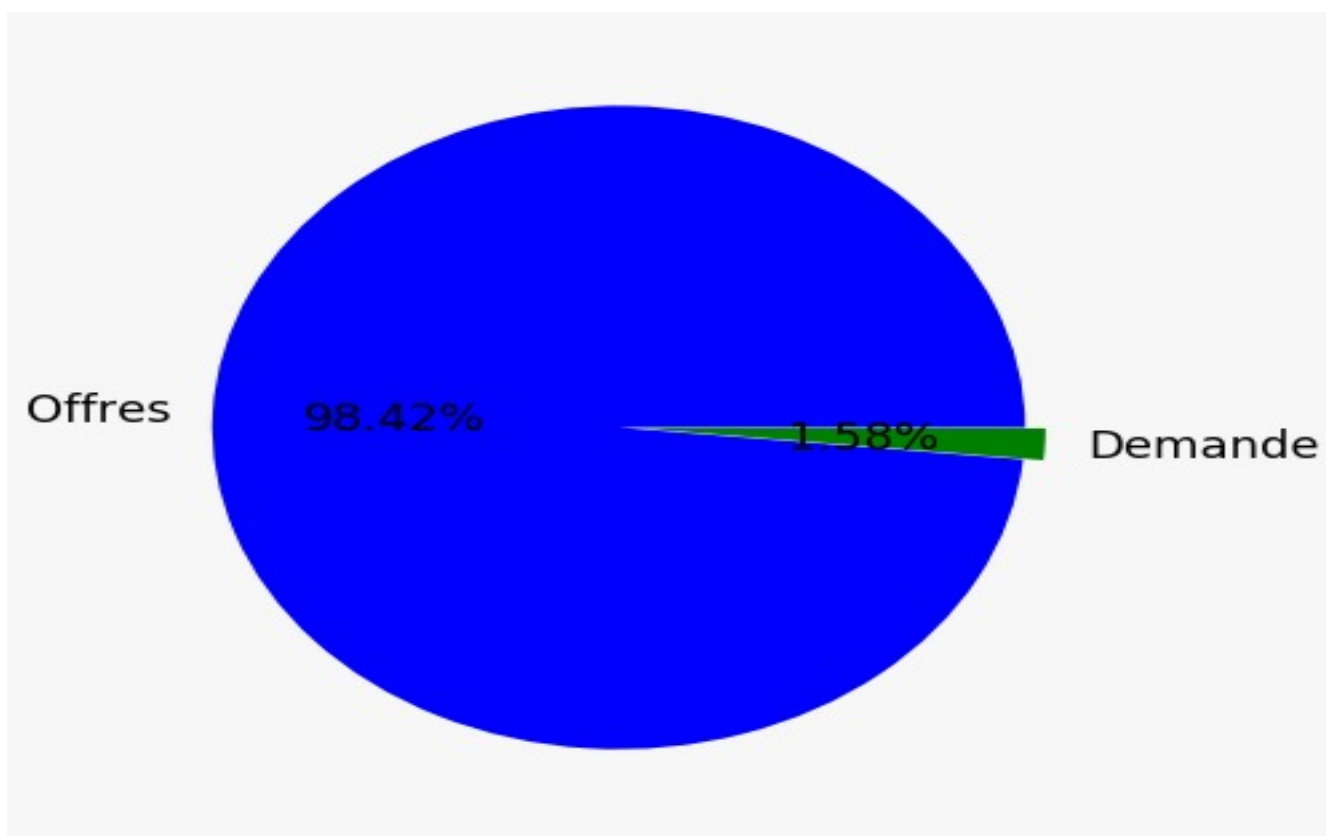


FIGURE 2.18 – Pourcentage de offres et des demandes

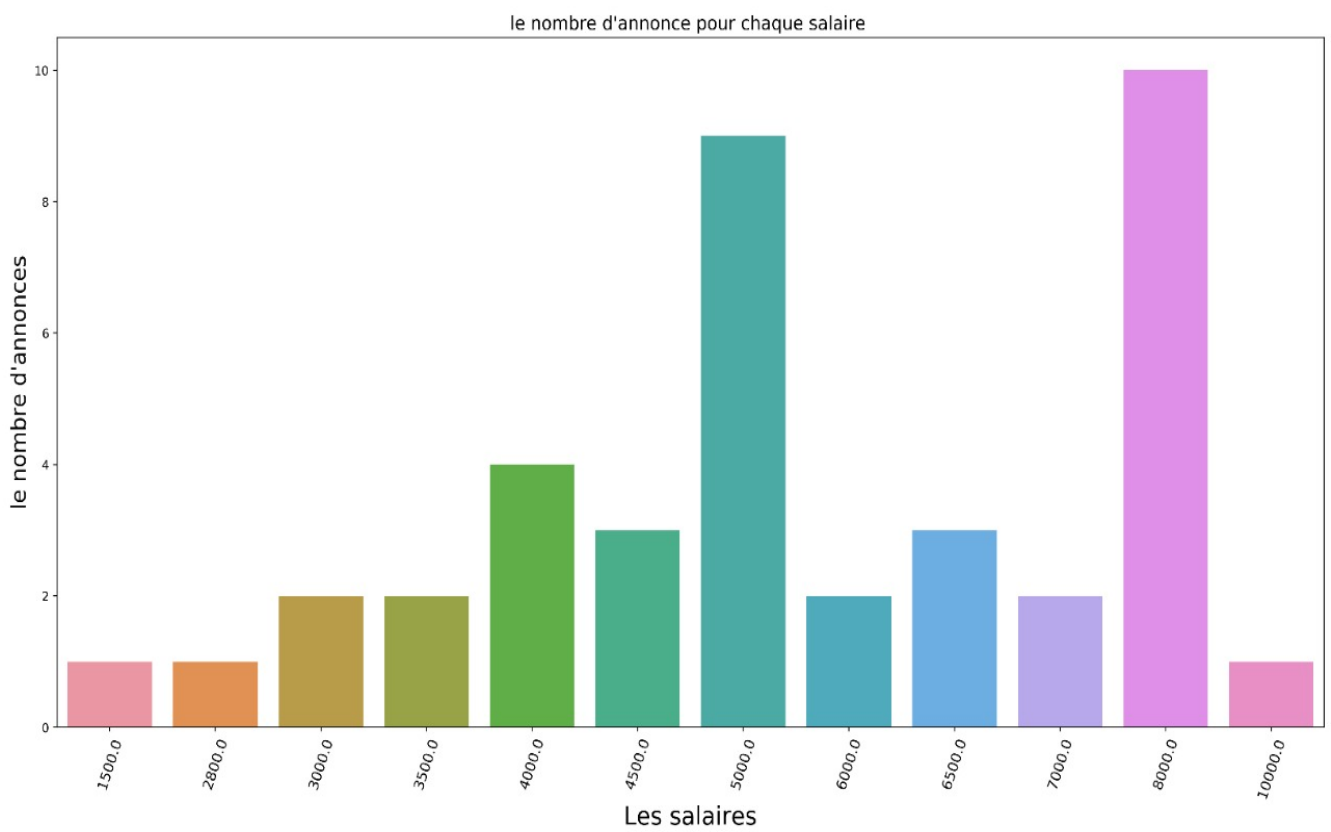


FIGURE 2.19 – le nombre d'annonce en fonction des salaire pour les offres d'emploie

Bibliographie

- [1] Baesens Bart Vanden Broucke Seppe. le livre :practical web scraping for data science : Best practices and examples with python.