

SAE15 :TP-2



CLAIRE Klayce
QOUDAD Bilal

1.Création de listes

Premièrement nous avons créé des listes pour chaque mois dans lesquelles nous avons mis les consos de début et de fin de mois. Nous avons également créé deux listes pour chaque moyen de production, une pour la production totale et une pour la production mensuelle.

```
#####Création des listes
#####
month =
["Jan", "Fev", "Mar", "Avr", "Mai", "Jun", "Jul", "Aou", "Sep", "Oct", "Nov", "Dec
"]
debutmois =
["2020-01-01", "2020-02-01", "2020-03-01", "2020-04-01", "2020-05-01", "2020
-06-01", "2020-07-01", "2020-08-01", "2020-09-01", "2020-10-01", "2020-11-01
", "2020-12-01"]
finmois =
["2020-01-31", "2020-02-29", "2020-03-31", "2020-04-30", "2020-05-31", "2020
-06-30", "2020-07-31", "2020-08-31", "2020-09-30", "2020-10-31", "2020-11-30
", "2020-12-31"]

Consomation = []
consomois = []

fioul=[]
fioulmois=[]

charbon=[]
charbonmois=[]

pompage=[]
pompagemois=[]

prod = []
prodmois = []

nucleaire = []
nucleairemois = []

gaz = []
gazmois = []
```

```

eolien = []
eolienmois = []

solaire = []
solairemois = []

hydraulique = []
hydrauliquemois = []

bioenergies = []
bioenergiesmois = []

```

2.Récupération

On récupère ensuite le contenu de la liste “table” contenant les données du fichier csv et on les incorpore aux liste qui leur correspondent afin de faciliter la mise en place des graphiques

```

#####Récupération des données dans le
fichier CSV#####
for mois in range(len(debutmois)):
    for i in range(len(table)):
#####Récupération des données par
mois#####
        if table[i][2] >= debutmois[mois] and table[i][2] <=
finmois[mois]:
            try:
#####Conversion des données en int si
possible#####
                Consomation.append(int(table[i][4]))
                nucleaire.append(int(table[i][10]))
                gaz.append(int(table[i][9]))

```

```

        eolien.append(int(table[i][11]))
        solaire.append(int(table[i][12]))
        hydraulique.append(int(table[i][13]))
        bioenergies.append(int(table[i][15]))
        fioul.append(int(table[i][7]))
        charbon.append(int(table[i][8]))
        pompage.append(int(table[i][14]))
#####Conversion de la plage des données
de production(7:15)#####
        for k in range(7,16):
            prod.append(int(table[i][k]))
        except:
            pass
#####Somme des données par
mois/production#####
        consomois.append(sum(Consomation))
        prodmois.append(sum(prod))
        nucleairemois.append(sum(nucleaire))
        gazmois.append(sum(gaz))
        eolienmois.append(sum(eolien))
        solairemois.append(sum(solaire))
        hydrauliquemois.append(sum(hydraulique))
        bioenergiesmois.append(sum(bioenergies))
        fioulmois.append(sum(fioul))
        charbonmois.append(sum(charbon))
        pompagemois.append(sum(pompage))

```

3.Affichage

Enfin on crée les graphiques, dans un premier temps le camembert qui utilise les données en fonction du moyen de production et dans un second temps la courbe qui utilise les données en fonction de la production et de la consommation globale

```
#####Affichage des données (repartition
de la prod) avec un
camembert#####
x =
[sum(nucleairemois),sum(gazmois),sum(eolienmois),sum(solairemois),sum(h
ydrauliquemois),sum(bioenergiesmois),]
plt.figure(figsize=(9,9))
plt.pie(x
,labels=["Nucleaire","Gaz","Eolien","Solaire","Hydraulique","Bioenergie
s"],autopct='%1.1f%%',explode=(0.1,0,0,0,0,0),startangle=50,normalize=T
rue)
plt.title('Répartition de la production principale en 2020')
plt.show()

#####Affichage des données
(consomation/production) avec des courbes
#####
plt.plot(month,consomois)
plt.plot(month,prodmois,color='black')
plt.grid(True)
plt.xlabel('Mois')
plt.ylabel('Consomation/Production(MWh)')
plt.title('Evolution de la consomation/production en 2020(MWh)')
plt.legend(['Consomation','Production'])
plt.show()
```

Répartition de la production principale en 2020

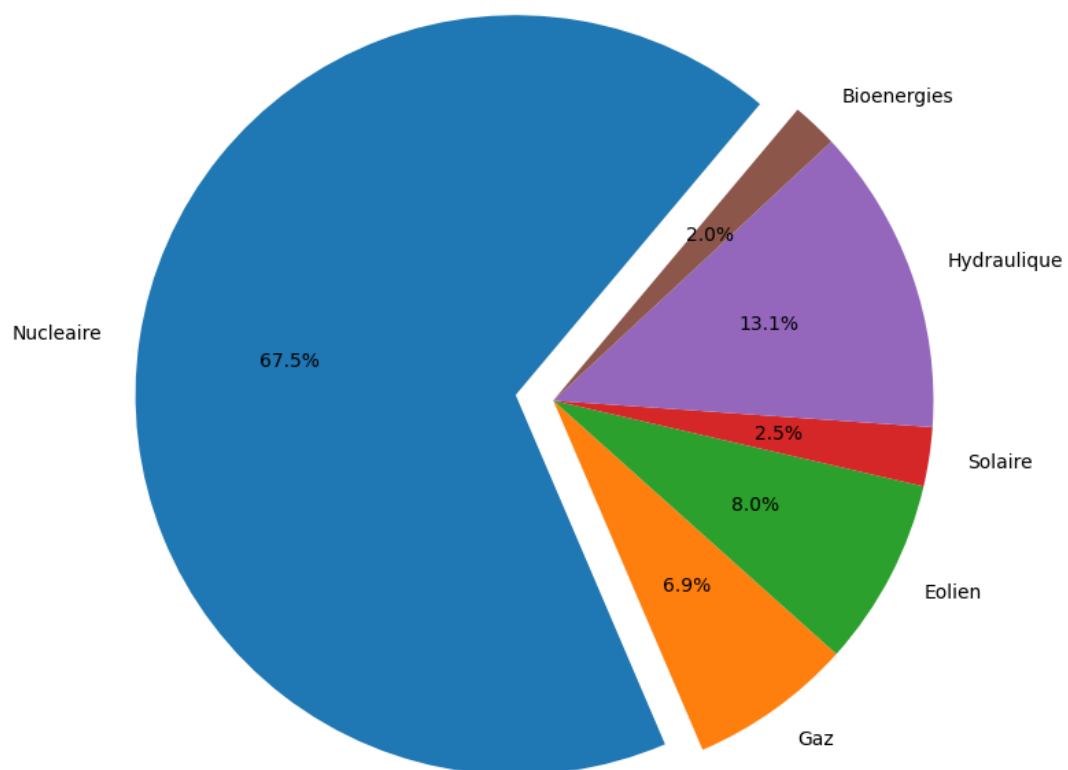


Figure 1

