Analyse statistique

Julian Marques 11A

Tous les graphiques et calculs ont été fait grâce à Python, à l'aide des bibliothèques Matplotlib, Seaborn, Numpy ainsi que Pymysql.

Toutes m'ont servies à réaliser les calculs demandés ainsi qu'à les illustrer sous forme de graphiques, et surtout, tout est géré automatiquement, on se connecte à la base de données, il y a juste à modifier le "user" et le "mot de passe" utilisés pour la base de données en local (ou sur serveur, donc préciser l'adresse) et enfin changer le nom de la base de données utilisée.

Une fois cela fait, il suffit de saisir la requête voulue et en exécutant le programme Python, la requête est exécutée sur la base de données, puis les informations sont stockées dans des listes et ces listes servent à afficher sur des graphiques ou dans le terminal les données retournées.
Le fichier avec le code Python est : AnalyseStatistique.py

1) Extraire de la base de données le nombre de formations par région en 2022, vous obtenez un premier jeu de 21 données notées V. Calculez

La requête utilisée:

```
select count(num_form) nb_form, region_etab_aff from REGION
natural join DEPARTEMENT natural join ETABLISSEMENT natural join
VOEUX natural join STATS where session = 2022 group by
region_etab_aff;
```

La requête SQL ressort cela dans le terminal:

```
+---------+----------------------------+
| nb_form | region_etab_aff            |
+---------+----------------------------+
|    1336 | Auvergne-Rhône-Alpes       |
|     506 | Bourgogne-Franche-Comté    |
|     614 | Bretagne                   |
|     386 | Centre-Val de Loire        |
|      62 | Corse                      |
|      14 | Etranger                   |
|     970 | Grand Est                  |
|     111 | Guadeloupe                 |
|      54 | Guyane                     |
|    1094 | Hauts-de-France            |
|    1871 | Ile-de-France              |
|     215 | La Réunion                 |
|     108 | Martinique                 |
|      39 | Mayotte                    |
|     544 | Normandie                  |
|     989 | Nouvelle-Aquitaine         |
|    1003 | Occitanie                  |
|     674 | Pays de la Loire           |
|      67 | Polynésie française        |
|     744 | Provence Alpes Côte d'Azur |
|       2 | Saint-Martin               |
+---------+----------------------------+
21 rows in set (0.026 sec)
```

V(barre) = 543.0

```
################################
# V barre (la moyenne de V)
################################
res = 0

for i in range(len(nb_form)):
    res += nb_form[i]

moyenne = res / (i+1)
print("\033[1;33m V(barre) : \033[0m", moyenne)
```

La médiane de V = 506

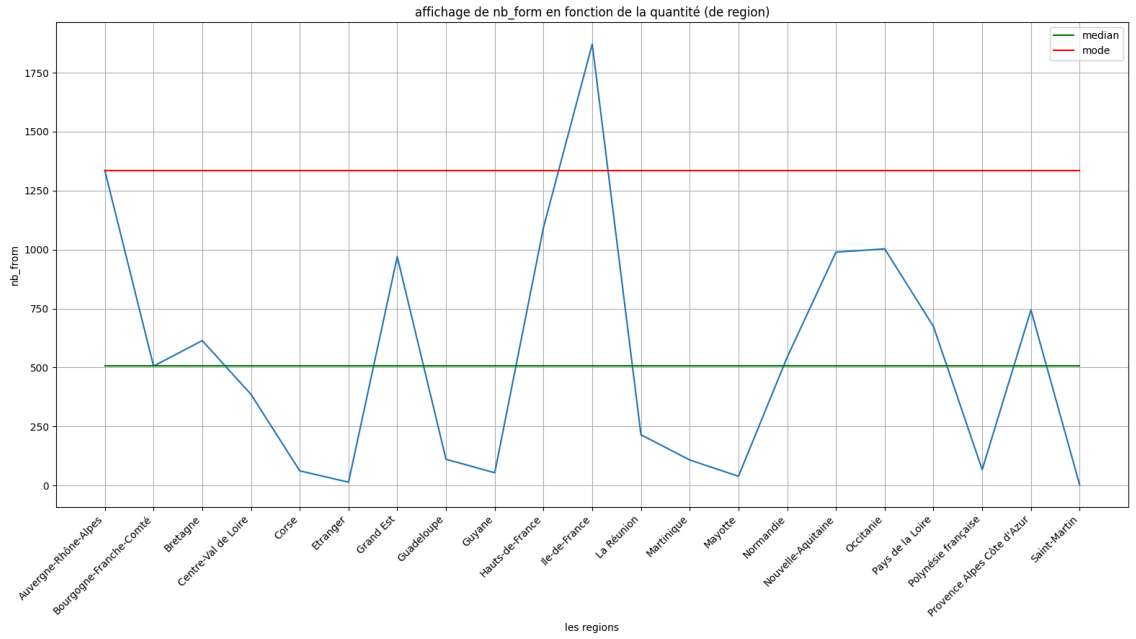
```
################################
# Calcul de la médiane
################################

tri_nb_form = sorted(nb_form)

n = len(tri_nb_form)
if n % 2 == 0:
    median = (tri_nb_form[n // 2 - 1] + tri_nb_form[n // 2]) / 2
else:
    median = tri_nb_form[n // 2]
print("\033[1;32m La median de V est:", median,"\033[0m")
```

Le mode de V = 1336 selon Numpy

```
################################
# Calcul du mode
################################


mode_counter = Counter(nb_form)
mode = mode_counter.most_common(1)[0][0]
print("\033[1;31m le mode de V est:", mode,"\033[0m")
```

Hors il est amodal car la caractéristique d'une distribution statistique ne comporte aucune valeur dominante.

Graphique pour illustrer :



2) Extraire le nombre de candidatures effectuées par région en 2022 lors de la phase principale, ces valeurs constituent une seconde statistique notée E. Vous obtenez ainsi la statistique double (V, E).

La requête utilisée:

```
select nb_form,nb_candi, A.region_etab_aff from
(select count(num_form) nb_form, region_etab_aff
    from REGION natural join DEPARTEMENT natural join
ETABLISSEMENT natural join VOEUX natural join STATS
    where session = 2022 group by region_etab_aff) AS A
JOIN
    (select sum(nb_voe_pp) nb_voe_pp, nb_candi, region_etab_aff
    from REGION natural join DEPARTEMENT natural join
ETABLISSEMENT natural join VOEUX natural join STATS
    where session = 2022 group by region_etab_aff
) AS B ON A.region_etab_aff=B.region_etab_aff;
```

La requête SQL ressort cela dans le terminal:

nb_form	nb_candi	region_etab_aff
1336	1170069	Auvergne-Rhône-Alpes
506	287462	Bourgogne-Franche-Comté
614	434649	Bretagne
386	268465	Centre-Val de Loire
62	14574	Corse
14	17149	Etranger
970	662770	Grand Est
111	26224	Guadeloupe
54	12469	Guyane
1094	743411	Hauts-de-France
1871	2376267	Ile-de-France
215	82451	La Réunion
108	20544	Martinique
39	22970	Mayotte
544	358231	Normandie
989	727834	Nouvelle-Aquitaine
1003	929747	Occitanie
674	459430	Pays de la Loire
67	11038	Polynésie française
744	719372	Provence Alpes Côte d'Azur
2	190	Saint-Martin

21 rows in set (0.053 sec)

Tracez le nuage de points (V, E).

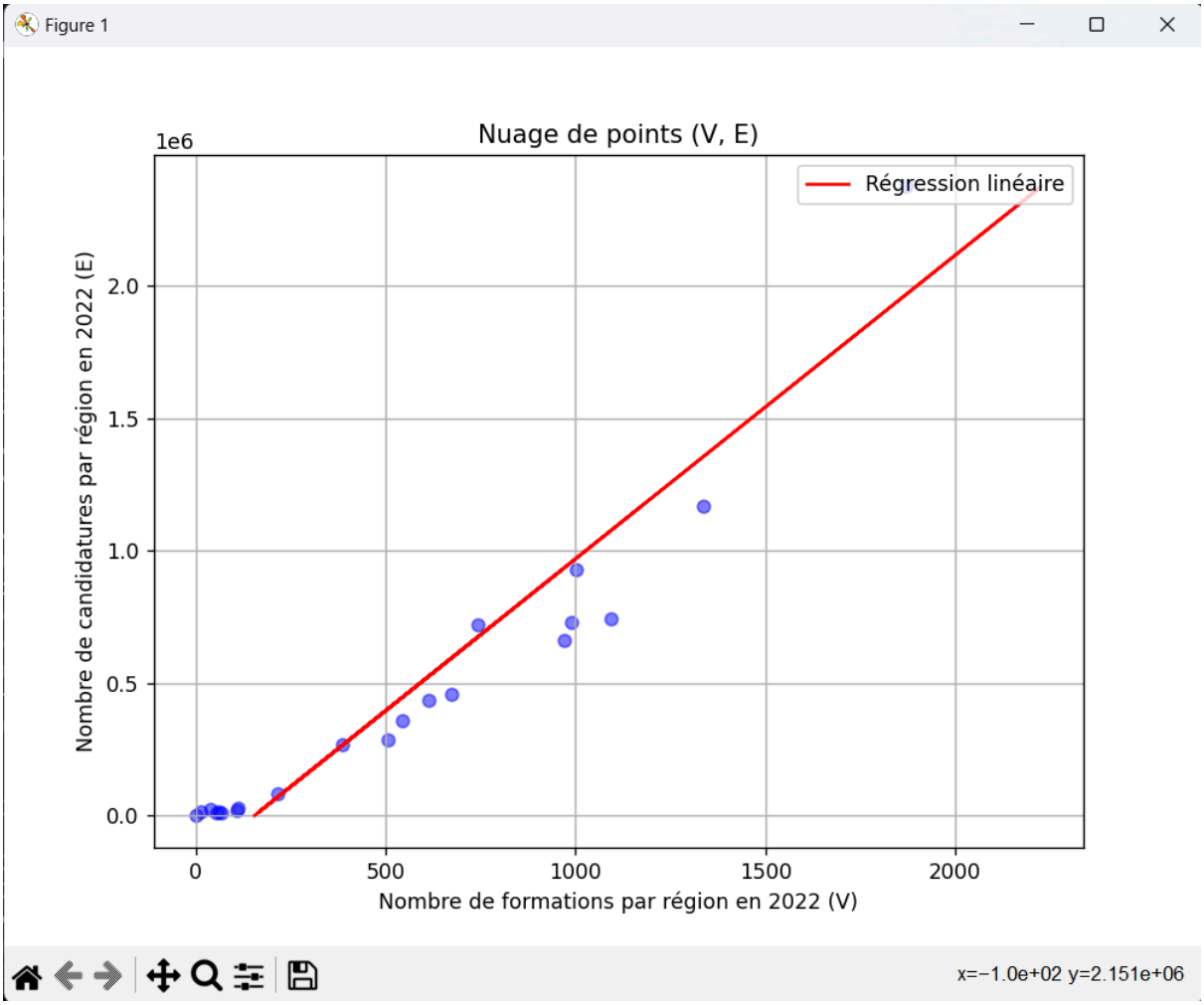


Figure 1
Nuage de points (V, E)
1e6
Régression linéaire
Nombre de candidatures par région en 2022 (E)
2.0
1.5
1.0
0.5
0.0
0
500
1000
1500
2000
Nombre de formations par région en 2022 (V)
x=−1.0e+02 y=2.151e+06

Voyez-vous apparaître (visuellement) une corrélation linéaire entre V et E ?

Oui on peut en apercevoir une (là où il y a le plus gros des points on peut apercevoir vers où elle sera), on peut voir aussi que V et E sont liés car quand V augmente E aussi et vice-versa.

Calculez le coefficient de corrélation ρV,E du couple (V, E)

```
########################################
# courbe de corrélation entre V & E
########################################
correlation = np.corrcoef(v, e)[0, 1]
print("\033[1;33m Corrélation entre e et v : \033[0m", correlation)
coefficients = np.polyfit(e, v, 1)
p = np.poly1d(coefficients)
# plt.plot(p(e), e, color='red', label='Régression linéaire')
```

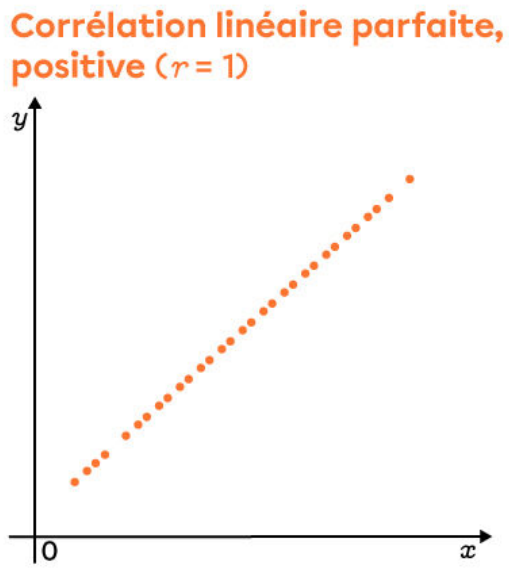
ρV,E = 0.9543248807237141
Calcul fait via Python et numpy
Ci-dessous les calculs exécutés avec le logiciel scilab en calculant tous les composants essentiels

```
Y=[1170069;287462;434649;268465;14574;17149;662770;26224;12469;743411;2376267;
82451;20544;22970;358231;727834;929747;459430;11038;719372;190]
X=[1336;506;614;386;62;14;970;111;54;1094;1871;215;108;39;544;989;1003;674;67;
744;2]


// calcul de la variance de X aka V et Y aka E
VarX= mean((X-mean(X)).^2)
VarY= mean((Y-mean(Y)).^2)

// covariance entre X et Y
CovXY = mean(X.*Y)-mean(X)*mean(Y)

// corrélation entre X et Y
CorXY = CovXY/sqrt(VarX*VarY)

// droite de régression
a = CovXY / VarX
b = mean(Y)-a*mean(X)

R2 = CorXY^2
```

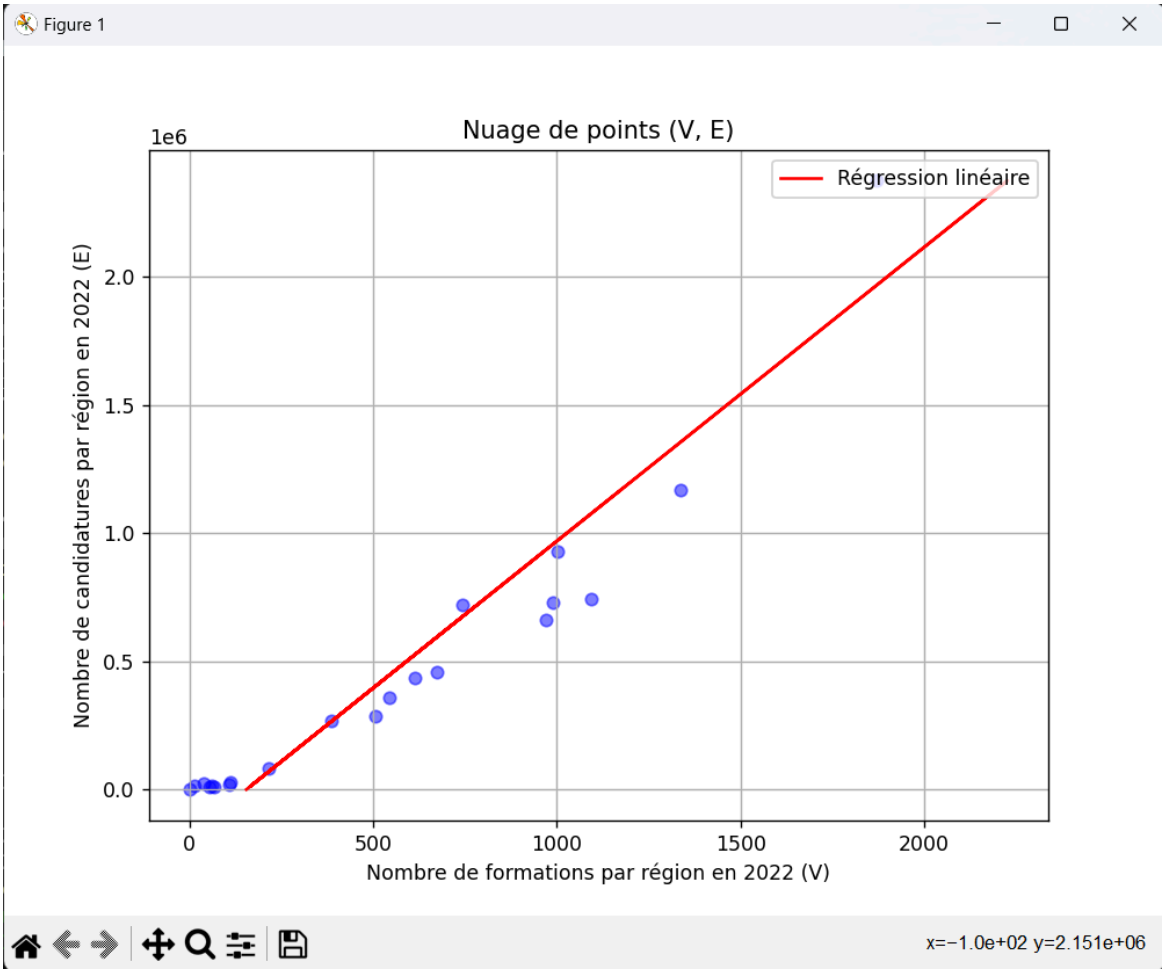
```
// calcul de la variance de X aka V et Y aka E
VarX= mean((X-mean(X)).^2) =   `256406.19`
VarY= mean((Y-mean(Y)).^2) =    `3.073D+11`

// covariance entre X et Y
CovXY = mean(X.*Y)-mean(X)*mean(Y) =    `2.679D+08`

// corrélation entre X et Y
CorXY = CovXY/sqrt(VarX*VarY) =  `0.9543249`

// droite de régression
a = CovXY / VarX =      `1044.7072`
b = mean(Y)-a*mean(X) = `-122260.94`

// Coef de détermination %
R2 = CorXY^2 =      `0.9107360`

// calcul de prédiction
x= `2500`
j = a*x+b
= `2489507.0`
```

Qu’en déduisez-vous ?

Une corrélation de 0.95432 est une corrélation positive entre ces deux ensembles, on sait que si l'un monte l'autre aussi et vice-versa. La corrélation est entre -1 et 1, plus elle est proche de 1 ou de -1 plus le lien linéaire entre les deux variables est parfait et quand elle est vers 0, plus le lien linéaire entre les deux variables est faible. donc pour une corrélation de 0.95 c’est une corrélation qui a un fort lien et sur le graphique des données, on y voit le lien par rapport à la courbe de régression tracée en rouge



Graphique pour illustrer :



3) En vous servant d'un outil mathématique de prédiction vu en cours de statistiques, quel nombre de candidatures minimum anticipez-vous si le

nombre de formations d'une région augmente au-delà de 2500 ? (Détaillez votre démarche et vos calculs)

```
// droite de régression
a = CovXY / VarX =      `1044.7072`
b = mean(Y)-a*mean(X) = `-122260.94`
```

Pour calculer la prédiction pour le nombre de formations de la région qui augmente par exemple vers 2500 formations.

Il suffit de prendre le calcul de la courbe de régression qui est :
= ax+b
ou x= 2500
a et b qui correspond à la droite de régression
= `1044.7071686851034*2500+(-122260.94497696351)`
= `2489506.98`

Pour une augmentation de formation de 2500 formations, nous avons une prédiction de 2489506 candidatures