SAE 2.04 - Exploitation d'une base de données

IUT de Lannion
Nathan Thuault, Eliott Janot
Année universitaire 2023 - 2024

1 - Les données Colleges.csv - Problématique

a - Présentation des données

Le fichier Colleges.csv contient plusieurs séries statistiques sur l'ensemble de toutes les collèges répertoriés dans notre base de données :

- La population est l'ensemble des départements français, représentés par leur dénomination.
- La 1ère variable statistique sur cette population est le nombre d'élèves en classe de segpa pour chaque département.
- La 2ème est le nombre d'élèves en classe d'ulis.
- La 3ème est le nombre d'élèves entre les segpa et les ulis.
- La 4ème est le nombre d'élèves total pour chaque département.
- Les autres variables sont le nombre d'élèves en ulis et en segpa pour chaque classe du collège.

b - Problématique

En utilisant ces données, on va essayer de répondre à la problématique suivante :

> Est-ce que le nombre d'élèves ayant des difficultés en cours permet de définir la longueur du nom du département ?

Notre variable endogène est donc la taille des noms de départements.

2 - Import des données, mise en forme

a - Importer les données en Python

On importe notre vue sous forme de DataFrame avec la commande suivante :

```
CollegesDF = pd.read_csv("Colleges.csv")
```

b - Mise en forme

Dans un premier temps nous nous assurons de ne pas avoir de cases vides dans nos données, puis on transforme notre DataFrame en Array :

```
CollegesDF = CollegesDF.dropna()
CollegesAr = CollegesDF.to_numpy()
```

c - Centrer-réduire

On ne garde que les colonnes de notre tableau qui contiennent des données numériques, on peut alors centrer-réduire ces données :

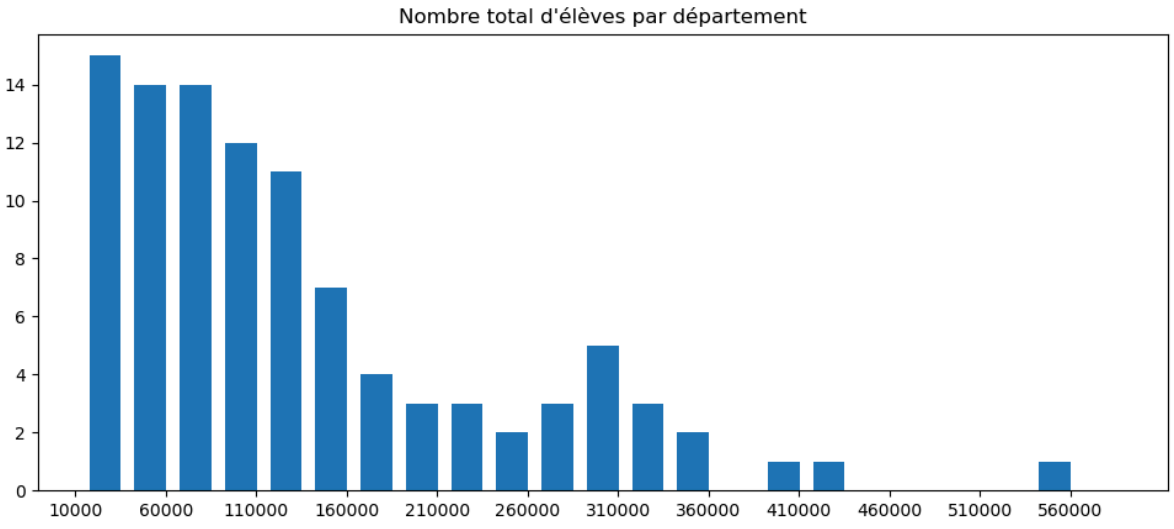
```
def Centreduire(T):
    T = np.array(T, dtype=np.float64)
    (n, p) = T.shape
    res = np.zeros((n, p))
    TMoy = np.mean(T, axis=0)
    TEcart = np.std(T, axis=0)
    for j in range(p):
        res[:, j] = (T[:, j] - TMoy[j]) / TEcart[j]
    return res
```

```
CollegesAr0 = CollegesAr[:, 1:]
CollegesAr0_CR = Centreduire(CollegesAr0)
```

3 - Exploration des données

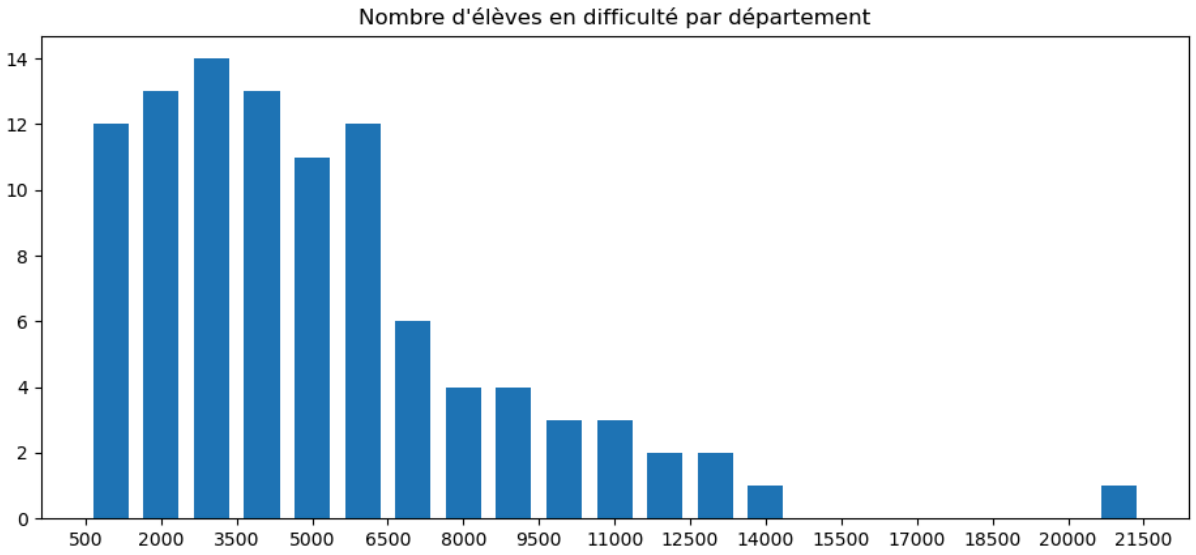
a - Représentations graphiques

On choisit d'étudier les diagrammes en bâtons de nos variables statistiques :

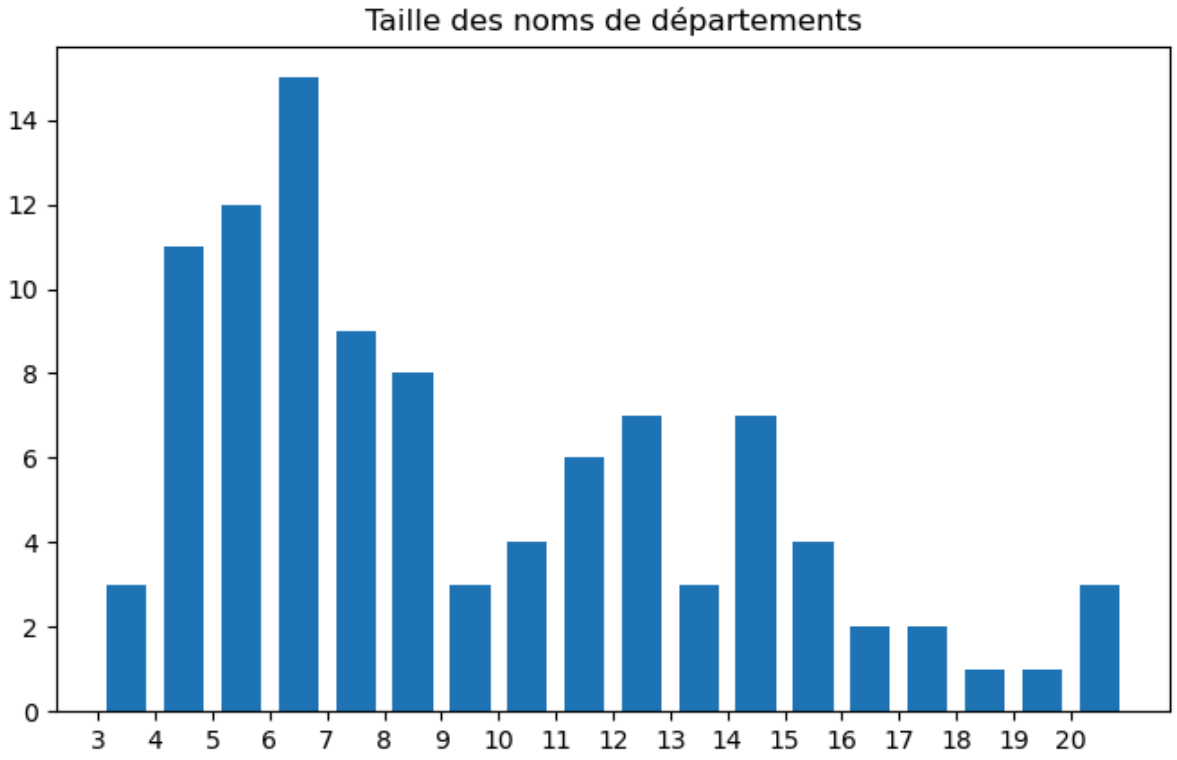


Dans un premier temps nous avons représenté le nombre total d'élèves par département. Nous pouvons voir que la majorité des départements possèdent entre 10 000 et 160 000 collégiens. Ensuite, une plus petite partie de départements

possèdent entre 160 000 et 360 000 collégiens. Finalement, il est important de noter qu'un département, le Nord, possède 561 632 collégiens.



Nous avons maintenant représenté le nombre d'élèves en difficulté par département. Ce graphique est similaire au précédent. En effet, il est assez évident que le nombre total d'élèves et le nombre d'élèves en difficulté sont deux variables fortement corrélées. Encore une fois, le département du Nord majore avec 21 099 élèves en difficulté.



Nous avons maintenant représenté la longueur des noms de département. Les valeurs vont de 3 jusqu'à 21. Nous observons une concentration des longueurs entre 4 et 7 lettres. Cela correspond à la taille moyenne d'un mot en français. Les autres valeurs semblent être réparties de manière plutôt égale.

Afin de réaliser des calculs de moyennes et de variances, nous avons séparé les noms de département en deux sous-groupes. Une première liste contient les noms de départements commençant par une voyelle et la deuxième ceux qui commencent par une consonne. Pour ce faire nous avons implémenté cette fonction :

```
def sepVoyCons(DepAr):
    voyelles = ['A', 'E', 'I', 'O', 'U', 'Y']
    depVoy = []
    depCons = []
    for i in DepAr:
        if i[0] in voyelles:
            depVoy.append(i)
        else:
            depCons.append(i)
    return depVoy, depCons
```

Puis nous obtenons les moyennes et variances avec ces deux commandes :

```
print(np.mean(tailleVoy), np.var(tailleVoy))
print(np.mean(tailleCons), np.var(tailleCons))
```

En moyenne, un nom de département possède 9 caractères et une variance de 20. Cependant, lorsqu'un département commence par une voyelle, il possède en moyenne 7.6 caractères et une variance de 21. Finalement, pour une consonne en premier caractère nous obtenons une longueur moyenne de 9.3 caractères et une variance 19.1.

b - Matrice de Covariance

a - Démarche

Nous allons maintenant calculer la matrice de covariance. Celle-ci nous permet d'obtenir en un seul calcul les corrélations 2 à 2 de toutes nos variables. De plus, si nous utilisons des valeurs centrées-réduites, la variance est égale à 1 et la covariance est égale au coefficient de corrélation.

b - Matrice de covariance

Dans un premier temps nous regroupons toutes nos données utiles dans une même array. Puis, nous pouvons centrer-réduire ces données.

```
Dif6emeAr = CollegesAr0[:,6]
Dif5emeAr = CollegesAr0[:,9]
Dif4emeAr = CollegesAr0[:,12]
Dif3emeAr = CollegesAr0[:,15]

DonneesAr = np.array([tailleDepAr, Dif6emeAr, Dif5emeAr,
Dif4emeAr, Dif3emeAr]).T
DonneesAr = Centreduire(DonneesAr)
```

Ensuite nous utilisons la fonction cov de numpy permettant de calculer la matrice de covariance associée. Nous avons d'abord forcé le typage des données en float pour permettre à la fonction cov de fonctionner correctement.

```
DonneesAr = np.array(DonneesAr, dtype=np.float64)
MatriceCo = np.cov(DonneesAr, rowvar=False, bias=True)
```

Voici la matrice de covariance obtenue :

	0	1	2	3	4
0	1	0.116644	0.115612	0.11557	0.11506
1	0.116644	1	0.99996	0.999853	0.99954
2	0.115612	0.99996	1	0.999939	0.999634
3	0.11557	0.999853	0.999939	1	0.99979
4	0.11506	0.99954	0.999634	0.99979	1

Nous pouvons maintenant plus facilement analyser nos données. Il est important de rappeler qu'en utilisant des valeurs centrées-réduites, la covariance est égale au coefficient de corrélation. De plus, la variance est bien égale 1 et se trouve dans la diagonale. Le reste des valeurs en bleu correspond à la covariance et donc au coefficient de corrélation entre les différents effectifs des classes de collèges par département. Evidemment, ceux-ci sont très fortement corrélés, les élèves ayant validé leur année resteront probablement dans le même collège l'année suivante. Cependant, les valeurs en rouge correspondent aux coefficients de corrélation entre la longueur du nom des départements et le nombre d'élèves en difficulté. Celui-ci est constamment aux alentours de 0.11. Nos variables ne semblent donc pas corrélées.

4 - Régression linéaire multiple

a - Utilisation de la Régression linéaire multiple : comment ?

En choisissant la 1ère variable statistique comme variable endogène et d'autres variables pertinentes comme variables explicatives, la régression linéaire multiple nous permettrait en théorie d'obtenir une estimation de la longueur des noms de départements.

b - Variables explicatives les plus pertinentes

Pour appliquer la régression linéaire multiple, nous devons définir notre modèle en utilisant ces données :

- **Variable endogène (Y)** : Longueur des noms de départements français.
- **Variables explicatives (X1, X2, X3, X4)** : Nombre d'élèves en difficulté par département pour chaque classe du collège (6ème, 5ème, 4ème, 3ème).

Nous allons modéliser cette relation par une équation de la forme :

$$Y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X3 + \beta_4 X4 + \epsilon$$

c - Lien avec la problématique

Cependant selon nos calculs réalisés précédemment, en particulier avec la matrice de covariance, il ne nous semble pas y avoir de corrélation dans notre modèle. Nous avons tout de même choisi ces variables explicatives afin de pouvoir au moins démontrer que notre problématique s'avère erronée.

d - Régression Linéaire Multiple en Python

Voici notre implémentation de la régression linéaire multiple en python :

```
# Créer le modèle de régression linéaire
model = LinearRegression()
# Regroupement des variables explicatives
Y = np.array([Dif6emeAr, Dif5emeAr, Dif4emeAr, Dif3emeAr]).T
# On doit reshape tailleDepAr par soucis algorithmique
tailleDepAr = np.array(tailleDepAr).reshape(-1, 1)
# On utilise des données centrées et réduites
tailleDepAr = Centreduire(tailleDepAr)
Y = Centreduire(Y)

# Entraîner le modèle
```

```
model.fit(tailleDepAr, Y)
# Afficher les coefficients
print("Coefficients (a0, a1, ...): ", model.coef_)
print("Intercept (a0): ", model.intercept_)
```

e - Paramètres, interprétation

Les paramètres obtenus sont :

- a0 = 0.11664369
- a1 = 0.11561167
- a2 = 0.11556983
- a3 = 0.11506035

Le signe du paramètre a0 nous indique comment la longueur du nom de département varie lorsque toutes les variables explicatives sont nulles.
Étant donné que les variables sont centrées-réduites, nous pouvons également comparer l'impact relatif de chaque variable sur le nombre d'élèves en difficulté.
En utilisant l'intercept nous obtenons : -7.37219421e-17, 1.19733589e-16, 2.95963019e-17 et -4.95527900e-17

f - Coefficient de corrélation multiple, interprétation

Pour calculer le coefficient de corrélation multiple nous avons écrit ce code :

```
# Prédictions
Y_pred = model.predict(tailleDepAr)

# Calcul du coefficient de corrélation multiple
ss_total = np.sum((Y - np.mean(Y))**2)
ss_residual = np.sum((Y - Y_pred)**2)
r_squared = 1 - (ss_residual / ss_total)

print("Coefficient de corrélation multiple (R²) : ",
r_squared)
```

Nous obtenons un coefficient de corrélation multiple de 0.01339176971412448.
Un R^2 proche de 1 indique que le modèle expliquerait bien la longueur du nom d'un département à l'aide du nombre d'élèves en difficultés. Ici, avec un coefficient à 0.01 nous sommes très éloignés de 1 et proches de 0. Nous pouvons donc affirmer sans craintes que le modèle ne prouve en aucun cas un lien d'implication entre le nombre d'élèves en difficultés et la longueur d'un nom de département.

5. Conclusions

a - Réponse à la problématique

La régression linéaire multiple nous permet de conclure que le nombre d'élèves en difficulté possède, étonnement, un impact totalement négligeable sur la longueur du nom du département.

b - Argumentation à partir des résultats de la régression linéaire

Comme nous l'avons démontré précédemment, nos différents calculs et coefficients de corrélations sont très bas et souvent quasi égaux à 0. Ceux-ci démontrent mathématiquement l'absence totale de corrélation.

c - Interprétations personnelles

En conclusion, après des heures de recherche intensive et d'analyse méticuleuse de données, il est clair que le nombre d'élèves en difficulté dans un département ne semble pas déterminer la longueur de son nom. Qui aurait cru que le sort orthographique des départements français était indépendant des défis académiques des collégiens français ? Certainement pas nos algorithmes python, qui sont désormais en quête d'un sens perdu dans les confins des statistiques de l'absurde. À quand une étude sur la corrélation entre le nombre de syllabes dans un nom de département et le taux de participation aux cours de mathématiques à l'IUT de Lannion ?