# RAPPORT DE LA SAE 2.04

## 0. Les données

### Présentation des données :

Le fichier export.csv contient plusieurs séries statistiques sur l'ensemble de toutes les formations répertoriées dans Parcoursup :

- La 1re série correspond au nom des formations.
- La 2e correspond à l'année.
- La 3e et la 4e série correspondent à la région et au type d'établissement. (public/privé)
- La 5e et 6e série correspondent au nombre total d'étudiant.e.s admis et au nombre total d'étudiant.e.s boursiers admis.
- La 7e série au taux de réussite des élèves boursiers.
- La 8e et 9e série correspondent à la capacité de la formation et au rang du dernier élève admis.

### Problématique :

Comment visualiser et analyser les taux de réussite des étudiants boursiers dans différentes filières, années, régions et types d'établissements ?

### La Régression Linéaire "comment ?" :

La régression linéaire multiple nous permet d'explorer la relation entre une variable dépendante (la proportion d'étudiants dans chaque formation) et plusieurs variables indépendantes (d'autres informations sur ces formations). En utilisant cette méthode, nous pouvons obtenir des estimations quantitatives de l'impact de chaque variable explicative sur la proportion d'étudiants, et ainsi mieux comprendre les facteurs qui influencent cette proportion dans différentes formations.

### La Régression Linéaire "pourquoi ?" :

La régression linéaire multiple nous permet d'identifier et d'estimer l'impact des différents descripteurs sur le montant des dommages. En analysant les paramètres de la régression, nous pouvons déterminer quels descripteurs ont une influence significative sur le montant des dommages. En comparant ces estimations aux données réelles, nous pourrons répondre à la problématique posée et évaluer la validité de notre modèle dans la prédiction des dommages.

# 1. Import des données, mise en forme

## Importer les données en Python

Après avoir exporté notre vue en .csv nous allons l'importer dans un dataFrame sous python
On définit le séparateur en ; car certaines formations contiennent des virgules dans le texte

```
dataFrame = pd.DataFrame(pd.read_csv("export.csv", sep = ';',))
```

## Mise en forme

On supprime les cases vides (NaN) et on supprime les colonnes qui contiennent du texte, enfin on convertir notre dataFrame en array

```
# drop colonne 0,2,3
dataFrame = dataFrame.drop(dataFrame.columns[[0, 2, 3]], axis=1)
dataFrame.dropna()
array = np.array(dataFrame)[:, 1:]
```

## Centrer-réduire

On exécute la fonction centrer réduire que l'on stock dans une nouvelles variables

```
1 usage
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

array_reduit = centreduire(array)
```

# Choix des variables explicatives

## Démarche

Dans cette partie, nous supprimerons toutes les variables qui ne sont pas pertinentes. Nous commençons par calculer la matrice de covariance.

```
cov = np.cov(array_reduit, rowvar=False)
```

## Matrice de covariance

on obtient la matrice suivante :

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1.00004 | 0.871654 | 0.138303 | 0.677892 | 0.467804 |
| 1 | 0.871654 | 1.00004 | 0.19391 | 0.548033 | 0.375984 |
| 2 | 0.138303 | 0.19391 | 1.00004 | 0.077359 | 0.0657665 |
| 3 | 0.677892 | 0.548033 | 0.077359 | 1.00004 | 0.270678 |
| 4 | 0.467804 | 0.375984 | 0.0657665 | 0.270678 | 1.00004 |

## Variables explicatives les plus pertinentes

Notre objectif est de trouver des variables qui expliquent le mieux possible le pourcentage de boursiers admis.e.s dans les formations, qui se trouve dans la colonne 2 d'array. La colonne 2 de cov donne les coefficients de corrélation du pourcentage de boursier.e.s admis.e.s avec chacune des autres variables/colonnes de type numérique d'array. On va choisir comme variables explicatives celles qui ont le coefficient de corrélation le plus grand (en valeur absolue) avec le pourcentage de boursier.e.s admis.e.s. Les coefficients de corrélation les plus grands en valeur absolue dans la colonne 2 de cov sont : 0.138303 et 0.1939. Ils correspondent aux variables numéros 0 et 1. Les colonnes 0 et 1 d'array correspondent aux :

- total_admis
- total_admis_boursier

On choisit donc ces 2 variables comme variables explicatives.

# 3. Régression linéaire multiple pour export.csv

## Régression linéaire multiple

On fait maintenant la régression linéaire multiple avec la série des pourcentages de boursier.e.s admis.e.s comme variable endogène, et les 2 variables explicatives trouvées ci-dessus.

```
linearRegression = LinearRegression()
linearRegression.fit(X, Y)
a = linearRegression.coef_
b = 0

y_pred = a*X+b
print(y_pred)
```

## Paramètres, interprétation

On attribue à X les variables explicatives 0 et 1 du tableau de covariance et a Y la valeur 2 qui est notre valeur endogène.

```

X = cov[:, [0, 1]]
Y = cov[:, 2]
```

On va donc tenter de prédire le taux de réussite à partir de total admis et de total admis boursier.

```
X: [[1.000043   0.87165398]
 [0.87165398 1.000043  ]
 [0.13830349 0.1939096 ]
 [0.6778918  0.54803295]
 [0.46780389 0.37598439]]

Y: [0.13830349 0.1939096  1.000043   0.07735902 0.06576648]
```

## Coefficient de corrélation multiple, interprétation

On fournit les données “X” et “Y” au modèle de régression linéaire.
linearRegression.coef_ permet de récupérer les coefficients de régression estimés à partir du modèle.
Enfin, on multiplie nos variables endogènes par nos coefficients de régression estimés.

```
linearRegression = LinearRegression()
linearRegression.fit(X, Y)
a = linearRegression.coef_
b = 0

y_pred = a*X+b
print(y_pred)
```

```
[[-2.00185655  1.07230631]
 [-1.74485121  1.23025012]
 [-0.27685185  0.23854705]
 [-1.35698378  0.67418861]
 [-0.93643601  0.46253495]]
```

Interprétation, Les valeurs de "y_pred" représentent les valeurs prédites de "Y" en utilisant les coefficients estimés et les valeurs correspondantes de "X". Chaque ligne dans "y_pred" correspond à une prédiction pour chaque ligne de “X”.

# Conclusion

## Réponse à la problématique :

La problématique était de prédire le taux de réussite en pourcentage des élèves boursiers en fonction du nombre total d'élèves admis  et du nombre total d'élèves boursiers admis . En utilisant une régression linéaire, nous avons pu obtenir des prédictions pour le taux de réussite des élèves boursiers en fonction de ces variables explicatives.

Argumentation à partir des résultats de la régression linéaire :
Les résultats de la régression linéaire fournissent des coefficients pour chaque variable indépendante dans le modèle. Dans ce cas, les coefficients indiquent comment le taux de réussite des élèves boursiers est influencé par le nombre total d'élèves admis et le nombre

total d'élèves boursiers admis. Il semblerait que plus le nombre total d'élèves admis augmente plus la prédiction y devient faible mais que pour le nombre total d'élèves admis boursier, plus elle est forte, plus la prédiction y est forte.

## Interprétations personnelles :

Sur la base des résultats obtenus, il est difficile de tirer des conclusions précises, cependant il semblerait que le total admis ait une influence sur le nombre d'élèves boursiers admis puisque plus il y a d'élèves admis, plus il y a de chance qu'un élève boursier soit admis.