TD/TP 2 : régression linéaire multiple

UE Modèle linéaire

moi

> ⚠ Avertissement
>
> Pensez à mettre votre nom dans l'entête du document.

```
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

Exercice 1 : utilisation de la régression linéaire multiple

Reprendre, si ce n'était pas déjà fait, le dernier exercice de la fiche précédente sur le QI d'enfants des États-unis.

1. Dans le bloc ci-dessous, on redonne les commandes pour importer les données. Ensuite, on ajuste un modèle de régression linéaire multiple. De quel modèle s'agit-il ?

```
kidiq = pd.read_csv('../TDTP1/tp1/kidiq.csv', sep=",")
kidiq['mom_hs_num'] = kidiq['mom_hs']
kidiq['mom_hs'] = kidiq['mom_hs'].astype('category')
kidiq['mom_work'] = kidiq['mom_work'].astype('category')

reg3 = smf.ols('kid_score ~ mom_iq * mom_hs', data = kidiq).fit()
print(np.sqrt(reg3.scale))
print(reg3.summary())
```

2. Dans ce modèle, l'ordonnée à l'origine représente la moyenne de deux-sous populations fictives de paires mères-enfants. Donner la formule mathématiques en terme d'espérance conditionnelle et

l'interpréter. Quelle remarque importante doit-on faire ici sur la valeur de `mom_iq` ? Le t-test de nullité de cet intercept a-t-il un intérêt ?

3. Ce modèle permet d'obtenir deux droites de régression, en fonction de la valeur de `mom_hs`. Donner les équations estimées des deux droites.

4. Représenter graphiquement les deux nuages de points et les deux droites estimées, en changeant la couleur suivant `mom_iq`.

```
# Extraire les coefficients
intercept = reg3.params['Intercept']
slope_mom_iq = reg3.params['mom_iq']
intercept_mom_hs = reg3.params['mom_hs[T.1]']
slope_interaction = reg3.params['mom_iq:mom_hs[T.1]']

# Créer le scatter plot
plt.figure(figsize=(10, 6))
scatter = sns.scatterplot(data=kidiq, x='mom_iq', y='kid_score', hue='mom_hs')

# Obtenir les couleurs des points
palette = scatter.get_legend().get_texts()
color_no = scatter.get_lines()[0].get_color()
color_yes = scatter.get_lines()[1].get_color()

# Ajouter les lignes de régression
x_vals = pd.Series(plt.gca().get_xlim())
y_vals1 = slope_mom_iq * x_vals + intercept
y_vals2 = (slope_mom_iq + slope_interaction) * x_vals + (intercept + intercept_mom_hs)

plt.plot(x_vals, y_vals1, color=color_no)
plt.plot(x_vals, y_vals2, color=color_yes)

plt.xlabel('mom_iq')
plt.ylabel('kid_score')
plt.show()
```

On peut également faire ce graphique en une seule ligne avec la commande ci-dessous.

```
sns.lmplot(data=kidiq, x='mom_iq', y='kid_score', hue='mom_hs', markers=["o", "x"])
```

5. Quelles vérifications graphiques doit-on faire pour pouvoir utiliser les intervalles de confiance et de prédiction ? Le faire et conclure.

```
from scipy.stats import norm
residus_student = reg3.get_influence().resid_studentized_internal
predictions = reg3.predict()

plt.figure(figsize=(10, 6))
plt.hist(residus_student, bins=30, density=True, alpha=0.6, color='g', range=(-4, 4))
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, 0, 1)
plt.plot(x, p, 'k', linewidth=2, color='red')
plt.title('Résidus studentisés')
plt.ylim(0, 0.4)
plt.show()

plt.figure(figsize=(10, 6))
sm.qqplot(residus_student, line='45')
plt.title('QQ plot des résidus studentisés')
plt.show()

plt.figure(figsize=(10, 6))
plt.scatter(predictions, residus_student, alpha=0.6, color='b')
plt.axhline(y=-1.96, color='red', linestyle='--')
plt.axhline(y=0, color='black', linestyle='-')
plt.axhline(y=1.96, color='red', linestyle='--')
plt.ylim(-4, 4)
plt.xlabel('Prédictions')
plt.ylabel('Résidus studentisés')
plt.title('Résidus studentisés vs Prédictions')
plt.show()
```

6. On note $\theta = \beta_{\texttt{mom\_iq}} + \beta_{\texttt{mom\_iq:mom\_hs1}}$ la pente de la droite lorsque la mère a obtenu son diplôme.

On rappelle que la matrice de variance-covariance de $\widehat{\boldsymbol{\beta}}$ est

$$\mathbb{V}(\widehat{\boldsymbol{\beta}}) = \sigma^2 \mathbf{V}, \quad \text{où } \mathbf{V} = \mathbf{x}'\mathbf{x}.$$

La matrice $\mathbf{V}$ et l'estimation $\hat{\sigma}$ de l'écart-type de l'erreur s'obtiennent avec les commandes ci-dessous.

```
V = reg3.normalized_cov_params
hat_sigma = np.sqrt(reg3.scale)
print(V)
print(hat_sigma)
```

Écrire θ mathématiquement sous la forme $\mathbf{T}\boldsymbol{\beta}$, puis proposer un intervalle de confiance pour θ.

On rappelle que

$$\mathbb{V}(\mathbf{T}\widehat{\boldsymbol{\beta}}) = \mathbf{T}\mathbb{V}(\widehat{\boldsymbol{\beta}})\mathbf{T}' = \sigma^2\mathbf{T}\mathbf{V}\mathbf{T}'$$

7. Quelle est l'intention du statisticien qui veut tester ces hypothèses ?

$$H_0 : \theta \geq 1 \quad \text{vs.} \quad H_1 : \theta < 1.$$

Construire le test, calculer la p-value et conclure.

Exercice 2 : production d'arbres mesquites

L'arbre mesquite fait partie de la famille des légumineuses, qui produit des gousses de haricots. Ces petits arbres sont endogènes de certaines régions désertiques et montagneuses du sud de l'Amérique du Nord. Son aire naturelle de répartition géographique va du Nord du Mexique jusqu'au Kansas.

On s'intéresse aux variables ci-dessous sur différents plants, collectées en deux lots :

- `weight` : biomasse produite (mesurée après récolte)
- `diam1` : plus long diamètre horizontal de la canopée (partie feuillue de l'arbre), en mètre,
- `diam2` : plus court diamètre horizontal de la canopée
- `canopy_height` : hauteur de la canopée
- `total_height` : hauteur totale
- `density` : densité d'arbres par unité d'exploitation (en nombres d'arbre par unité de surface)
- `group` : lot de l'étude (MCD pour le 1er lot, ALS pour le second).

Le premier lot correspond à des mesures faites sur 26 arbres juste avant la récolte. Le second lot correspond à des mesures faites sur 20 arbres à une autre date.

Voici comment importer les données.

```
mesquite = pd.read_csv("tp2/mesquite.dat", delim_whitespace=True)
mesquite['group'] = mesquite['group'].astype('category')
```

La mesure d'intérêt (la biomasse) est beaucoup plus compliquée à obtenir que les autres variables, puisqu'il faut faire une récolte. On souhaite donc construire des modèles qui permettent de prédire la biomasse à partir des autres variables. Pour cela, on va ajuster, étudier et comparer différents modèles de régression permettant de prédire la biomasse. On essaie ici de répondre à deux questions :

- doit-on transformer les variables à l'échelle logarithmique ?
- doit-on introduire de nouvelles variables fonction des variables mesurées ?

1. Ajuster un premier modèle linéaire où `weight` est prédit linéairement à partir des autres covariables. Que pensez-vous de ce modèle ?

La valeur de R^2 ici, de l'ordre de 85%, est assez grande. Elle mesure la qualité des prédictions sur les observations utilisées pour ajuter le modèle. Or, il est attendu qu'un modèle prédise à-peu-près correctement les observations sur lesquelles il a été ajusté. En effet, l'ajustement du modèle revient à trouver les valeurs de $\boldsymbol{\beta}$ pour lesquelles les prédictions sont les plus proches des valeurs de Y observées au sens des moindres carrés. Lorsqu'il y a plusieurs covariables et peu d'observations (comme dans ce jeu de données), le modèle ajusté peut donc ne pas être représentif de la population, mais uniquement être représentatif des données. Dans ce cas, on parle de **sur-apprentissage** : le modèle ajusté colle trop aux données pour être représentatif du phénomène général.

On souhaite se **prémunir contre le sur-apprentissage**, c'est-à-dire évaluer la capacité de ce modèle à faire de bonnes prédictions sur de nouvelles données. Nous avons besoin de nouvelles données pour évaluer notre modèle, alors que notre table ne comporte que 46 observations. Faute d'avoir accès à de nouvelles données, il faut donc enlever des observations avant l'ajustement du modèle, pour ensuite pouvoir les utiliser comme "nouvelles données".

Parmis les méthodes pour réaliser ce programme, nous allons utiliser la validation croisée en mettant une observation de côté (ou *leave-one-out cross-validation*). Pour observation i de la table initiale, il faut :

1. ajuster le modèle sans cette i-ème observation,
2. utiliser le modèle ajusté pour prédire la valeur de Y pour cette i-ème observation,
3. enregister la valeur prédite.

2. À l'aide d'une boucle, faire ces opérations sur 46 observations de `mesquite` et enregistrer la i-ème prédiction dans `y_pred[i]`.

```
n = mesquite.shape[0]
y_pred = np.zeros(n)

for i in range(n):
    # Compléter ici
```

3. Avec quelles nouvelles quantités, calculée sur ses prédictions, doit-on comparer $\hat{\sigma}$ et R^2 ? Faire les calculs et conclure.

4. On s'intéresse maintenant à un modèle qui prédit la log-biomasse en fonction du logarithme de toutes les variables numériques et de `group`. Ajuster ce nouveau modèle.

Puis utiliser de nouveau une démarche de validation croisée pour étudier les qualités prédictives de ce second modèle. On enregistrera les prédictions dans le vecteur `logy_pred`. Commenter les résultats obtenus.

5. Les estimations de σ par validation croisée ne sont pas comparables directement. En utilisant la fonction exponentielle pour quitter l'échelle logarithmique, calculer sur le second modèle une estimation de l'erreur de prédiction sur `weight`.

Comparer ce nombre avec celui obtenu en question 3 et conclure quant au choix entre les deux modèles.

6. Proposer deux autres modèles qui utilisent les variables ci-dessous, les ajuster et prédire la biomasse sur les 46 observations avec une démarche de validation croisée.

$$\texttt{canopy\_volume} = \texttt{diam1} \times \texttt{diam2} \times \texttt{canopy\_height}$$

$$\texttt{canopy\_area} = \texttt{diam1} \times \texttt{diam2}$$

$$\texttt{canopy\_shape} = \texttt{diam1}/\texttt{diam2}$$

Voici comment ajouter ces variables à la table :

```
mesquite['canopy_volume'] = mesquite['diam1'] * mesquite['diam2'] * mesquite['canopy_height']
mesquite['canopy_area'] = mesquite['diam1'] * mesquite['diam2']
mesquite['canopy_shape'] = mesquite['diam1'] / mesquite['diam2']
```

7. Comparer les 4 modèles ajustés et conclure.

Exercice 3 : le théorème de Cochran

On rappelle deux résultats importants sur les vecteurs gaussiens.

- Toute transformation linéaire d'un vecteur gaussien est un vecteur gaussien.
- Un bloc de coordonnées d'un vecteur gaussien est indépendant d'un autre bloc de coordonnées si et seulement si la matrice de covariance entre ces deux blocs est nulle.

Voici le théorème de Cochran.

> Soit $\mathbf{Z} \sim \mathcal{N}_d(\boldsymbol{\mu}, \sigma^2 I_d)$ un vecteur gaussien de dimension d, dont les coordonnées sont indépendantes et de même variance σ^2. Soit Π une matrice de projection orthogonale sur un sous-espace F de dimension d_1 : $\Pi' = \Pi$, $\Pi^2 = \Pi$, $\mathrm{tr}(\Pi) = d_1$.
>
> Dans ce cas, on rappelle que $(I_d - \Pi)$ est la projection orthogonale sur l'orthogonal de F, noté $F^\perp$, de dimension $d_2 = d - d_1$.
>
> Alors,
>
> - les vecteurs aléatoires $\Pi\mathbf{Z}$ et $(I_d - \Pi)\mathbf{Z}$ sont deux vecteurs gaussiens indépendants, de lois respectives

$$\Pi\mathbf{Z} \sim \mathcal{N}_d\Big(\Pi\boldsymbol{\mu};\ \sigma^2\Pi\Big), \quad (I_d - \Pi)\mathbf{Z} \sim \mathcal{N}_d\Big((I_d - \Pi)\boldsymbol{\mu};\ \sigma^2(I_d - \Pi)\Big)$$

- les variables aléatoires

$$S_1 = \sigma^{-2}\left\|\Pi\Big(\mathbf{Z}-\boldsymbol{\mu}\Big)\right\|_2^2, \quad S_2 = \sigma^{-2}\left\|(I_d-\Pi)\Big(\mathbf{Z}-\boldsymbol{\mu}\Big)\right\|_2^2$$

  sont indépendantes, et de lois respectives $\chi^2(d_1)$ et $\chi^2(d_2)$.

Donc, la variable aléatoire

$$F = \frac{S_1/d_1}{S_2/d_2}$$

suit une loi de Fisher $\mathcal{F}(d_1, d_2)$.

On note $\mathbf{Y}$ le vecteur colonne de dimension $2d$, formé par la concaténation des vecteurs $\Pi\mathbf{Z}$ et $(I_d - \Pi)\mathbf{Z}$.

Partie A. Démonstration du théorème

1. Calculer les espérances de $\Pi\mathbf{Z}$ et $(I_d - \Pi)\mathbf{Z}$.

On suppose maintenant, et jusqu'à la fin du problème, que $\boldsymbol{\mu} = (0, \dots, 0)$. Quitte à remplacer $\mathbf{Z}$ par $\mathbf{Z} - \boldsymbol{\mu}$, cette hypothèse ne nous fait pas perdre de généralité.

2. Montrer que $\mathbf{Y} = \mathbf{A}\mathbf{Z}$, où $\mathbf{A}$ est une matrice $(2d) \times d$ qui s'écrit par bloc

$$\begin{pmatrix} \Pi \\ I_d - \Pi \end{pmatrix}$$

3. Calculer l'espérance de $\mathbf{Y}$ et montrer que la matrice de variance-covariance s'écrit par bloc sous la forme

$$\begin{pmatrix} \sigma^2\Pi & 0_d \\ 0_d & \sigma^2(I_d - \Pi) \end{pmatrix}$$

où 0_d est la matrice nulle de dimension $d \times d$.

4. En déduire le premier item du théorème et l'indépendance de S_1 et S_2.

Soit $\{u_1, \dots, u_d\}$ une base orthonormée de $\mathbb{R}^d$ adaptée à la décomposition $F \oplus F^\perp$. Autrement dit, $\{u_1, \dots, u_{d_1}\}$ est une base de F et $\{u_{d_1+1}, \dots, u_d\}$ est une base de $F^\perp$.

5. On pose $X_i = \langle Z, u_i \rangle = u_i'\mathbf{Z}$. Montrer que le vecteur colonne $\mathbf{X} = (X_1, \dots X_d)$ vérifie $\mathbf{X} = \mathbf{U}'\mathbf{Z}$, où $\mathbf{U}$ est une matrice carré.

6. Quelle est la loi de $\mathbf{X}$?

7. Montrer que $S_1 = X_1^2 + \dots + X_{d_1}^2$ et conclure.

Partie B. Utilisation dans le modèle linéaire gaussien

La modélisation des données dans ce cas est

$$[\mathbf{Y}|\mathbf{X} = \mathbf{x}] \sim \mathcal{N}_n(\mathbf{x}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n).$$

L'estimateurs des moindres carrés de $\boldsymbol{\beta}$ (et du maximum de vraisemblance) est

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$

1. Montrer que

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}.$$

2. En déduire que $[\widehat{\boldsymbol{\beta}}|\mathbf{X} = \mathbf{x}]$ suit la loi gaussienne multivariée caractérisée par

$$\mathbb{E}[\widehat{\boldsymbol{\beta}}\big|\mathbf{X} = \mathbf{x}] = \boldsymbol{\beta}, \quad \text{et} \quad \mathbb{V}[\widehat{\boldsymbol{\beta}}\big|\mathbf{X} = \mathbf{x}] = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}.$$

3. On s'intéresse maintenant à

$$\text{SSE} = \|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|_2^2.$$

Écrire SSE sous la forme $\|\mathbf{M}\boldsymbol{\varepsilon}\|_2^2$, où $\mathbf{M}$ est une matrice.

4. Montrer que $\mathbf{M}$ est une matrice de projection orthogonale en montrant que $\mathbf{M} = \mathbf{I}_n - \mathbf{N}$, où $\mathbf{N}$ est une matrice de projection orthogonale qui vérifie $\mathbf{N}^2 = \mathbf{N}$ et $\mathbf{N}' = \mathbf{N}$.

5. Montrer qu'un vecteur $\mathbf{u} \in \mathbb{R}^n$ est dans le noyau de $\mathbf{N}$ si et seulement si $\mathbf{u}$ est dans l'orthogonal de l'espace engendré par les colonnes de $\mathbf{X}$. En déduire que $\mathbf{M}$ est la matrice de projection sur l'espace engendré par les colonnes de $\mathbf{X}$, et $\mathbf{N}$ sur son orthogonal.

Remarque pour la question 5. On rappelle que si $\mathbf{A}$ est une matrice de taille $k \times k$, symétrique, définie positive (donc de rang k) et $\mathbf{v} \in \mathbb{R}^k$, alors $\mathbf{v}'\mathbf{A}\mathbf{v} = 0$ si et seulement si $\mathbf{v} = 0$.

6. En déduire la loi conditionnellement à $\mathbf{X} = \mathbf{x}$ de l'estimateur non biaisé de σ^2, et que cet estimateur est indépendant de $\widehat{\boldsymbol{\beta}}$, conditionnellement à $\mathbf{X} = \mathbf{x}$.