

TD/TP 1 : régression linéaire simple et un peu de multiple

UE Modèle linéaire

moi

⚠️ Attention

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
```

Exercice 1 : quelques résultats théoriques

On se place dans le cadre de la régression linéaire simple. Autrement dit, pour un individu (X, Y) de la population, on suppose que

$$Y = \beta_0 + \beta_1 X + \varepsilon, \quad \text{avec } \sigma^2 = \text{Var}(\varepsilon),$$

où X et ε sont indépendants, et $\mathbb{E}(\varepsilon) = 0$.

1. Soit $x \in \mathbb{R}$. Calculer $\mathbb{E}(Y|X = x)$, la moyenne de Y sur la sous-population où $X = x$.

On se donne maintenant un échantillon de taille n , modélisé par des paires indépendantes (X_i, Y_i) , $i = 1, \dots, n$. Et on considère les estimateurs définis par

$$\hat{\beta}_1 = \frac{\sum_i (Y_i - \bar{Y})(X_i - \bar{X})}{\sum_i (X_i - \bar{X})^2}, \quad \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$$

2. Que vaut \bar{Y} en fonction de β_0 , β_1 , \bar{X} et $\bar{\varepsilon}$? Et que vaut $\sum_i \bar{\varepsilon}(X_i - \bar{X})$? En déduire que

$$\hat{\beta}_1 = \beta_1 + \sum_i \frac{(X_i - \bar{X})}{S_X^2} \varepsilon_i, \quad \text{où } S_X^2 = \sum_j (X_j - \bar{X})^2.$$

3. Soient $x_1, \dots, x_n \in \mathbb{R}$. En déduire que

$$\text{Var}\left(\hat{\beta}_1 \middle| X_{1:n} = x_{1:n}\right) = \sigma^2 / \sum_i (x_i - \bar{x})^2$$

4. Avec la loi des grands nombres, montrer que, presque sûrement, l'équivalence suivante est vraie

$$S_X^2 \sim_{n \rightarrow \infty} n \text{Var}(X).$$

5. En déduire l'équivalent presque sûr que

$$\text{Var}\left(\hat{\beta}_1 \middle| X_{1:n}\right) = \frac{\sigma^2}{S_X^2} \sim_{n \rightarrow \infty} \frac{1}{n} \frac{\sigma^2}{\text{Var}(X)}$$

Par quel facteur environ faut-il multiplier la taille de l'échantillon pour que l'erreur standard soit divisée par 10 ?

Maintenant, on considère le modèle de régression linéaire simple gaussien, où pour tout $x \in \mathbb{R}$,

$$[Y | X = x] \sim \mathcal{N}(\beta_0 + \beta_1 x; \sigma^2).$$

6. Montrer que, sur l'échantillon introduit précédemment, la log-vraisemblance, conditionnellement à $X_{1:n} = x_{1:n}$ est donnée par

$$\ell(\beta_0, \beta_1, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i)^2 - \frac{n}{2} \log(\sigma^2) + \text{constante.}$$

7. Trouver la solution explicite du maximum de vraisemblance.

Exercice 2 : mise en œuvre de la régression

Aux élections présidentielles américaines, l'économique influence le résultat de l'élection. On dit souvent que le parti au pouvoir le conserve si l'économique se porte bien. On va donc considérer un jeu de données à cinq variables :

- `year` : l'année de l'élection,
- `growth` : la croissance moyenne du revenu des américains (en %),
- `vote` : part du parti sortant dans le vote aux élections présidentielles (en %),
- `inc_party_candidate` : nom du candidat du parti sortant,
- `other_candidate` : nom du candidat d'opposition.

On peut charger les données ainsi :

```
hibbs = pd.read_csv('tp1/hibbs.dat', sep=" ")
```

1. Quelle est la variable à expliquer ici ? Et la covariable ? Et la “population statistique” ? Y a-t-il une chance que le modèle puisse se prolonger pour faire des projections au delà des valeurs observées de *growth* ?

2. L'ajustement se fait à la commande ci-dessous.

```
regression = smf.ols('vote ~ growth', data=hibbs).fit()
resume_regression = regression.summary()
print(resume_regression)
```

Comment s'interprète β_0 et β_1 ici ? Les valeurs estimées sont-elles plausibles ? Et comment s'interprète σ ici.

Quelle est la part de variabilité de Y expliquée par la covariable ?

3. Pour les β_i , on peut obtenir des intervalles de confiance au niveau 95% avec la commande ci-dessous.

```
intervalle_confiance = regression.conf_int(alpha=0.05)
intervalle_confiance.columns = ['2.5%', '97.5%']
intervalle_confiance
```

Comment peut-on interpréter le fait que 50% ne soit pas dans l'intervalle de confiance de β_0 ?

4. Quelles hypothèses doit-on vérifier sur ε pour utiliser ces intervalles de confiance ? Et faire des intervalles de prédiction ?

Faites ces vérifications graphiquement en commandant les graphiques ci-dessous. Quel graphique est inutile ?

```
residus_student = regression.get_influence().resid_studentized_internal
predictions = regression.fittedvalues

from scipy.stats import norm

# Histogram of residuals with normal curve
plt.figure(figsize=(10, 6))
plt.hist(residus_student, bins=5, density=True, alpha=0.6, color='blue',
         edgecolor='black')
x = np.linspace(-4, 4, 100)
plt.plot(x, norm.pdf(x, 0, 1), 'r', label="Loi normale")
plt.xlabel('Résidus studentisés')
plt.ylabel('Densité')
```

```

plt.legend()
plt.show()

# QQ plot
sm.qqplot(residus_student, line ='45')
plt.title('QQ Plot des résidus studentisés')
plt.show()

# Residuals vs fitted values
plt.figure(figsize=(10, 6))
plt.scatter(predictions, residus_student, edgecolor='k', facecolor='blue',
            alpha=0.7)
plt.axhline(0, color='black')
plt.axhline(-1.96, color='red', linestyle='--')
plt.axhline(1.96, color='red', linestyle='--')
plt.xlabel('Valeurs prédictes')
plt.ylabel('Résidus studentisés')
plt.show()

```

5. On admet que les intervalles de confiance et de prédiction sont utilisables. [Vus les résultats sur les graphiques, leurs bornes inférieures sont sans doute un peu trop élevées.] Au moment de l'élection opposant Hillary Clinton et Donald Trump, le parti sortant était celui d'Hillary Clinton (démocrate). Et la variable `growth` était à environ 2%. Quel intervalle doit-on utiliser si l'on s'intéresse au résultat de cette élection ?

```

nouvelle_croissance = pd.DataFrame({'growth': [2.0]})

# Confidence interval prediction
conf_pred = regression.get_prediction(nouvelle_croissance).conf_int(alpha=0.05)

# Prediction interval
pred_interval = regression.get_prediction(nouvelle_croissance).\
    summary_frame(alpha=0.05)[['obs_ci_lower', 'obs_ci_upper']]

print("Intervalle de confiance : \n", conf_pred)
print("Intervalle de prédiction : \n", pred_interval)

```

6. Calculer les carrés des demi-longueurs des intervalles de confiance et de prédiction. Que vaut la différence entre ces deux carrés ? Justifier la réponse.

```

from scipy.stats import t

conf_lower, conf_upper = conf_pred[0]
pred_lower, pred_upper = pred_interval.iloc[0]

```

```

carre1 = ((conf_upper - conf_lower)/2)**2
carre2 = ((pred_upper - pred_lower)/2)**2
print("Carré de la demi-longueur de l'intervalle de confiance : ", carre1)
print("Carré de la demi-longueur de l'intervalle de prédiction : ", carre2)
print("Différence entre les deux carrés : ", carre2 - carre1)
np.sqrt(carre2 - carre1)/t.ppf(0.975, regression.df_resid)

```

Exercice 3 : Des modèles plus complexes

Voici le jeu de données issu d'une étude reliant le QI d'enfants à leur mère aux USA.

```

kidiq = pd.read_csv('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')

```

Il est composé de 5 variables :

- `kid_score` : le QI de l'enfant,
- `mom_hs` : une variable catégorielle qui indique si la mère a obtenu son diplôme à l'issu des études secondaires (*high school*), égale à 1 si c'est le cas, 0 sinon,
- `mon_hs_num` : la même variable, au format numérique,
- `mom_iq` : le QI de la mère,
- `mom_work` : un indicateur catégoriel sur l'emploi de la mère,
- `mom_age` : l'âge de la mère.

On s'intéresse à des modèles où l'on cherche à prédire le QI de l'enfant.

1. On s'intéresse à prédire `kid_score` avec `mom_hs`.

Le code ci-dessous calcule les moyennes et écarts-types de `kid_score` au sein des deux sous-échantillons caractérisées par la valeur de `mom_hs`.

```
kidiq.groupby('mom_hs')['kid_score'].agg(['count', 'mean', 'std'])
```

Voici un premier modèle :

```

reg1 = smf.ols('kid_score ~ mom_hs', data=kidiq).fit()
print(reg1.summary())

```

Quelle différence y a-t-il avec ce modèle ?

```
reg1bis = smf.ols('kid_score ~ mom_hs_num', data=kidiq).fit()
print(reg1bis.summary())
```

Comment s'interprète β_1 ici? Quelle est la part de variance de Y expliquée par `mom_hs` ?

2. On essaie maintenant de faire la même chose avec `mom_work` comme covariable.

Les moyennes et écarts-types au sein des 4 sous-échantillons sont :

```
kidiq.groupby('mom_work')['kid_score'].agg(['count', 'mean', 'std'])
```

```
reg1ter = smf.ols('kid_score ~ mom_work', data=kidiq).fit()
print(reg1ter.summary())
```

Comment s'interprètent les β ici ?

3. On veut maintenant prédire Y à l'aide de `mom_iq`. Faire l'analyse complète avec un modèle de régression linéaire simple et étude des hypothèses.

4. Montrer que le modèle de régression linéaire simple s'écrit aussi

$$Y - \mathbb{E}(Y) = \beta_1 (X - \mathbb{E}(X)) + \varepsilon.$$

Donner un intervalle de confiance pour β_1 au niveau 95%. Interpréter le fait que l'intervalle de confiance est inclus dans $[0; 1[$ comme un phénomène de régression vers la moyenne.

5. On s'intéresse maintenant à prédire Y avec `mom_iq` et `mom_hs` conjointement. On notera que ces deux variables sont liées :

```
kidiq.boxplot(column='mom_iq', by='mom_hs', grid=False)
```

Voici deux modèles de régression linéaire qui utilisent conjointement ces deux variables pour prédire Y :

```
reg2 = smf.ols('kid_score ~ mom_hs + mom_iq', data=kidiq).fit()
print(reg2.summary())
```

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

Quelles sont les équations de ces modèles ? Comment se lisent ses équations dans les deux sous-groupes définis par la variable `mom_hs` ?

Quelles sont les parts de variances expliquées par les deux modélisations ?

Comparer l'interprétation des effets β en facteur de la covariable `mom_iq` dans les deux modèles ci-dessus et dans le modèle de la question 3.

Voit-on encore des phénomènes de régression vers la moyenne pour l'une ou l'autre des modélisation, dans l'un ou l'autre des sous-groupes ?