TP N° 3 : Inférence sur échantillons gaussiens

Objectifs du TP : savoir réaliser des tests simples en `Python` sur des populations suivant des lois gaussiennes.

Le fichier `TP-Gaussiennes_squelette.ipynb` contient une ébauche de réponses que l'on pourra utiliser pour le TP.

1 Inférence sur la moyenne d'un échantillon

On se place ici dans le cadre où le statisticien dispose d'un seul échantillon à analyser. On note X la variable aléatoire du modèle. L'espérance $\mathbb{E}(X)$ est inconnue. Soit μ_0 un réel fixé. On désire tester l'hypothèse suivante :

$$\mathcal{H}_0 \ : \ \mathbb{E}(X) \ = \ \mu_0 \quad \text{contre} \quad \mathcal{H}_1 \ : \ \mathbb{E}(X) \neq \mu_0 \ ,$$

à l'aide du test de Student. Selon le contexte, les tests de Student s'effectuent à l'aide des commandes de `scipy.stats` telles que `ttest_1samp` pour le cas d'un seul échantillon, ou `ttest_ind` (teste si deux échantillons indépendants ont la même espérance), `ttest_ind_from_stats` (teste si deux échantillons indépendants ont la même espérance), etc.

En 1879, le physicien américain Michelson a fait plusieurs expériences pour vérifier la valeur de la vitesse de la lumière c proposée par le physicien français Cornu en 1876. La valeur proposée par Cornu était $c = 299\ 990$ km/s. Michelson a obtenu les 20 mesures suivantes pour la vitesse de la lumière : les valeurs des données sont les valeurs mesurées par Michelson auxquelles on a soustrait 299 000 afin de ne pas avoir à manipuler des chiffres trop grands.

Ces 20 observations peuvent être considérées comme les valeurs observées de 20 variables aléatoires gaussiennes ayant une espérance commune mais inconnue $\mathbb{E}(X)$. Si les conditions expérimentales pour mesurer la vitesse de la lumière sont sans biais, il est alors raisonnable de supposer que $\mathbb{E}(X)$ est la vraie vitesse de la lumière.

1) Télécharger les données[1] du fichier `michelson.txt` comme dans les TPs précédents.
2) Importer les données à l'aide de la commande `read_csv` de `pandas`.
3) Représenter les données sous forme de densité et commenter en particulier l'hypothèse de données gaussiennes.
4) En supposant que la variance théorique est connue et vaut $\sigma = 105$, tester si les mesures de Michelson confirment la valeur de la vitesse de la lumière proposée par Cornu. On pourra regarder la p-value obtenue par $2 - 2\Phi\left(\frac{|\mu - \mu_0|}{\frac{\sigma}{\sqrt{n}}}\right)$.
5) Sans faire l'hypothèse que la variance est connue, utiliser la commande `ttest_1samp` pour tester si les mesures de Michelson confirment la valeur de la vitesse de la lumière proposée par Cornu. Quelle est la conclusion du test ?
6) Donner un intervalle de confiance pour la moyenne, aux niveaux 0.90 et 0.95, avec ou sans l'hypothèse de variance connue.
7) Refaire le test pour confirmer la vitesse calculée par Cornu en créant une variable `df_michelson['true-speed']` contenant la vraie valeur des mesures. Cela change-t-il la conclusion ?

1. Données disponibles ici : http://josephsalmon.eu/enseignement/datasets/michelson.txt

2 Comparaison de deux échantillons indépendants

On se place dans le cadre où le statisticien dispose de deux échantillons indépendants à analyser. On note μ_1 l'espérance de la variable dans le modèle de la première population et μ_2 l'espérance de la variable dans le modèle de la seconde. Bien sûr, ces deux paramètres sont inconnus. On souhaite tester

$$\mathcal{H}_0 \ : \ \mu_1 = \mu_2 \quad \text{contre} \quad \mathcal{H}_1 \ : \ \mu_1 \neq \mu_2$$

avec un test de Student. On pose $\mu_{\text{diff}} = \mu_1 - \mu_2$.

Dans une étude[2] sur les mécanismes de détoxication, on dispose de la concentration du DDT et de ses dérivés, DDD et DDE, (en mg/g) contenus dans des brochets du Nord (*Esox lucius*), capturés dans la rivière Richelieu (province de Québec). Les données en question sont relatives aux brochets de 2 et 3 ans.

1) Ouvrir le fichier de données et regarder comment il est organisé.
2) Importer le jeu de données en `Python` avec la commande `read_csv` de `pandas`.
3) Re-coder la variable âge à l'aide de la fonction `replace` de `pandas`
4) Calculer et commenter les statistiques résumées obtenues avec les commandes ci-dessous :

```
df_brochets.query('age==2').describe()
df_brochets.query('age==3').describe()
```

On pourra aussi garder les échantillons par âge pour la suite avec la commande :

```
X2 = df_brochets.query('age==2')['conc']
X3 = df_brochets.query('age==3')['conc']
```

5) Afficher un graphique en "violon" de la concentration pour chacun des âges.
6) Effectuer un test de Student pour deux échantillons indépendants avec la commande `ttest_ind` et interpréter les résultats.

2.1 Test d'égalité des variances : les brochets

Lorsque l'on a deux échantillons indépendants, pour déterminer le choix correct pour l'option `var.equal` dans l'utilisation de la commande `t.test`, il est préférable de tester au préalable l'égalité de variance par un test de Fisher.

Pour σ_1^2 la vraie variance de la première population et σ_2^2 la vraie variance de la seconde population, la statistique de ce test sous l'hypothèse nulle :

$$\mathcal{H}_0 \ : \ \sigma_1^2 \ = \ \sigma_2^2 \ ,$$

est la suivante :

$$\frac{\hat{\sigma}_1^2}{\hat{\sigma}_2^2} \sim \mathcal{F}(n_1 - 1, n_2 - 1) \ ,$$

où $\mathcal{F}$ est la loi de Fisher[3]. Pour effectuer le test, on utilise donc la fonction `f` de `scipy.stats` qui permet d'obtenir par exemple la fonction de répartition.

Pour appliquer correctement ce test, il faut vérifier que les deux échantillons proviennent de distributions normales.

Par défaut, l'hypothèse alternative est $\mathcal{H}_1 \ : \sigma_1^2 \neq \sigma_2^2$ (cas bilatéral)[4].

On reprend l'exemple des brochets vu précédemment.

7) Lancer le test d'égalité des variances entre les deux classes d'âge différentes et conclure. Pour cela on admettra que la p-value peut être obtenue par :

2. Données disponibles ici : http://josephsalmon.eu/enseignement/datasets/brochet2.dat
3. La loi de Fisher est décrite ici : https://fr.wikipedia.org/wiki/Loi_de_Fisher
4. On peut changer d'hypothèse alternative pour des tests unilatéraux : pour avoir $H_1 \ : \sigma_1^2 > \sigma_2^2$ (resp. $H_1 \ : \sigma_1^2 < \sigma_2^2$)

```
from scipy.stats import f
f_distrib = f(len(X2) - 1, len(X3) - 1)
f_stat = np.var(X2, ddof=1) / np.var(X3, ddof=1)
p_value = 2 * min(f_distrib.cdf(f_stat), 1 - f_distrib.cdf(f_stat))
```

8) Que signifie `ddof` dans le calcul des variances ci-dessus ?

9) Reprendre le test de Student d'égalité des moyennes sur le cas des brochets, cette fois avec l'hypothèse supplémentaire que les variances sont identiques.

2.2 Comparaison de la pollution sur Toulouse et sur Montpellier

Nous allons maintenant utiliser des données de pollution recueillies sur diverses communes de l'Occitanie[5].

10) Charger les données dans `pandas` dans un dataframe `df_pol_occ`.

11) Observer ce que donnent les commandes suivantes :

```
df_pol_occ['polluant'].unique()
df_pol_occ['nom_com'].unique()
```

12) Compléter le script suivant pour visualiser les violons des divers polluants sur Montpellier et Toulouse sur la période étudiée :

```
polluants = ['O3', 'PM10', 'NO', 'NO2', 'PM2.5']
fig, axes = plt.subplots(2, 3, figsize=(10, 10))
for i, polluant in enumerate(polluants):
    querry = "(nom_com=='TOULOUSE' or nom_com=='MONTPELLIER') and " + \
        "polluant=='{}'".format(polluant)
    df_polluant = df_pol_occ.query(querry)[['nom_com', 'valeur_originale']]
    plt.title(polluant)
    ax = axes.reshape(-1)[i]
    sns.violinplot(????, cut=0, ax=ax)
    ax.set_title('polluant')
plt.tight_layout()
```

13) En utilisant une boucle `for`, tester l'égalité des niveaux des pollutions pour les cinq polluants vus précédemment.

3 Comparaison d'échantillons appariés

On se place dans le cas où le statisticien dispose de deux échantillons appariés à analyser. On désire faire le test (de Student) de l'hypothèse nulle

$$\mathcal{H}_0 \ : \ \mu_{\text{diff}} \ = \ 0$$

avec $\mu_{\text{diff}} = \mu_1 - \mu_2$ où μ_1 est l'espérance (inconnue) de la première variable dans la population et μ_2 est l'espérance (inconnue) de la seconde variable dans la population.
Manipulation avec Python. Il faut utiliser la commande `ttest_rel`. Afficher la page d'aide de cette commande en tapant `ttest_rel`. Par défaut, on teste l'égalité des espérances, c'est-à-dire avec $\mu_{\text{diff}} = 0$.

5. Données disponibles ici :
http://josephsalmon.eu/enseignement/datasets/Mesure_journaliere_Region_Occitanie_Polluants_Principaux.csv

3.1 Traitement des eaux usées : différences entre deux filtres

Dans une étude sur le traitement des eaux usées, l'efficacité de deux filtres, l'un en fibre de verre (variable **verre**) et l'autre en papier filtre Whatman numéro 40 (variable **papier**) a été testée. Sur des prélèvements de 200 millilitres d'eau provenant d'usine de pâte à papier, la quantité de solides en suspension retenus par les deux filtres a été mesurée. Les résultats de ces analyses sont contenus dans le fichier **filtre.dat**.

14) Télécharger les données[6] du fichier `filtre.dat` comme dans les TPs précédents.

15) Créer une variable **delta** égale à la différence entre les deux mesures. Pour cela, taper dans la fenêtre de commande :

```
df_filtre['delta'] = df_filtre['verre'] - df_filtre['papier']
```

16) Représenter la distribution de la variable `delta` à l'aide d'une méthode à noyau.

17) Faire un test de Shapiro-Wilk[7] sur cette dernière variable et conclure. On pourra utiliser la fonction `shapiro` du package `scipy.stats`

18) Effectuer le test de Student d'égalité des moyennes pour ces deux échantillons appariés. Commenter.

3.2 Comparaison de mesures de hauteur d'arbre : différence avant et après abatage

L'objectif de l'étude[8] décrite ci-dessous est de valider ou non une nouvelle technique de mesure de la taille des arbres sur pied (non abattus) en prenant un risque de 5%.

Dans une forêt, on choisit des arbres au hasard que l'on mesure sur pied (debout). Ensuite, on les abat puis on les mesure à nouveau. Chaque arbre a donc été mesuré deux fois. On veut tester l'égalité des moyennes de ces deux séries pour comparer les deux méthodes de mesure.

19) Importer ces données avec Python

20) Créer une variable **delta** égale à la différence entre les deux mesures et représenter la densité de cette variable. Commenter.

21) Peut-on dire au risque $\alpha = 5\%$ que la nouvelle technique de mesure est (en moyenne) valide ou biaisée ?

6. données disponibles ici : http://josephsalmon.eu/enseignement/datasets/filtre.dat
7. https://fr.wikipedia.org/wiki/Test_de_Shapiro-Wilk
8. Données disponibles ici : http://josephsalmon.eu/enseignement/datasets/tailles_arbres.csv