Institut National des Télécommunications et des Technologies de l'Information et de la Communication INT-TIC d'Oran Algérie

Analyse de Données

3ème année ingénieur

Fiche de TP N° 3

Statistique Descriptive Univariée et Bivariée

2ème Partie : Tests Statistiques

Un **test statistique** est une procédure permettant de choisir entre deux hypothèses, dites hypothèses statistiques. Cette décision (le choix entre les deux hypothèses) est faite selon un ensemble de données observées. Les deux hypothèses sont formulées suivant le phénomène étudié.

Si la démarche générale est la même pour tous les tests statistiques, les lois utilisés diffèrent en fonction du phénomène étudié et de la formulation du problème. La démarche générale consiste à supposer qu'une des deux hypothèses (appelée l'hypothèse nulle, ou H_0) est vraie. Le test utilisé ensuite permet de **rejeter** ou **pas** cette hypothèse.

Dans le cadre de ce TP, nous allons nous intéresser à un sous ensemble de tests statistiques. Il s'agit des tests permettant de comparer entre deux variables (analyse bivariée). Dans ce cas spécifique, les deux hypothèses correspondront respectivement à l'**indépendance** ou à la **dépendance** entre les deux variables.

Deux tests statistiques sont abordés dans ce TP : le **test d'indépendance de Chi 2** pour étudier le lien entre **deux variables qualitatives**, et le **test d'ANOVA à un facteur** pour étudier le lien entre **une variable qualitative** et **une variable quantitative**.

Généralement, ces tests procèdent au calcul d'une certaine statistique (une mesure) qu'on va appeler **`st`** dans ce TP. Chaque type de test possède une connaissance à priori sur la distribution de cette statistique. Par exemple, dans le cas du test de chi 2, la statistique utilisée suit une loi de chi 2. Dans le cas d'ANOVA à un facteur, la distribution est une loi de Fisher-Snedecor. En fonction de la loi de distribution utilisée et de la statistique calculée à partir des données observées, une valeur (dite **`p`** ou **`p-value`**) comprise entre 0 et 1 est calculée. Si la valeur **`p`** est inférieure à un certain seuil (généralement **`0.05`**), l'hypthèse H_0 est rejetée. Cela veut dire que l'hypothèse H_0 est peu probable dans ce cas.

Le fichier de données utilisé dans ce TP est le même utilisé dans les exemples précédents : `cardio.csv`. Il en est de même pour les modules utilisés : `numpy`, `scipy`, `pandas` et `matplotlib`.

```
>>> import numpy as np
>>> import scipy.stats as stat
>>> import pandas as pd
>>> import matplotlib.pyplot as plt
>>> import os

>>> os.chdir('C:\\...\\...\\DataExamples')
>>> table = pd.read_csv('cardio.csv', sep=';')
```

1. Test d'indépendance de Chi 2 entre deux variables qualitatives

Considérons les deux variables qualitatives `gender` et `cholesterol`. La question est de savoir si ces deux variables sont indépendantes ou pas. Autrement dit, est ce que le taux de cholestérol a un lien avec le sexe de la personne ou pas. L'hypothèse H_0 suppose que les deux variables sont indépendantes.

Dans un premier temps, on calcule le tableau de contingence :

```
>>> gender_chol = pd.crosstab(table.gender, table.cholesterol, margins=True).apply(
lambda x:x/len(table))
>>> gender_chol
cholesterol  Abnormal  Critical  Normal   All
gender
F                0.07      0.09    0.41  0.57
M                0.04      0.05    0.34  0.43
All              0.11      0.14    0.75  1.00
```

Ce tableau de contingence est à comparer avec un autre tableau de contingence, dit théorique. En effet, si les deux variables étaient totalement indépendantes, on aurait eu le tableau théorique.

Prenons l'exemple de la case (1,1) de notre tableau de contingence, c'est à dire la fréquence des femmes qui ont un taux de cholestérol anormal. En effet, d'une part, il existe 11% de personnes avec un taux anormal. D'autre part, il existe 57% de femmes. Par conséquent, s'il n'y avait pas de lien entre les deux variables, idéalement on aurait eu 0,11*0,57 = 0,0627 (ou 6,27%) de personnes qui sont des femmes et qui ont un taux de cholestérol anormal (au lieu du 7% calculé de notre tableau). Le tableau théorique complet se calcule de la même façon pour toutes les cases.

La statistique **`st`** représente une "différence" entre notre tableau de contingence et le tableau de contingence théorique : c'est la somme des carrées des différences entre les fréquences respectives (case par case) des deux tableaux. Le but est de savoir si cette différence est suffisamment significative (en d'autres termes, si les deux tableaux sont suffisamment éloignés) pour dire que l'hypothèse de départ (H_0 : indépendance entre les deux variables) est à rejeter.

La loi de chi 2 est une famille de lois qui diffèrent selon ce que l'on appelle le **degré de liberté**. Il est démontré que la somme des carrées des différences des fréquences des deux tableaux suit une loi de chi 2 à un certain degré de liberté (qu'on va nommer `df`). Le degré de liberté `df` est égal à `(n-1)*(m-1)`, avec `n` et `m` le nombre de modalités de la première variable et de la deuxième variable respectivement. Dans notre cas, la variable `gender` a `2` modalités (`F` et `M`), tandis que la variable `cholestrol` en a `3` (`Abnormal`, `Critical` et `Normal`), ce qui donne un `df = (2-1)*(3-1) = 2`.

Enfin, la valeur `p` (ou la `p_value`) est calculée en fonction de la statistique `st` et de la loi de chi 2 à `df` degré de liberté.

La fonction `chi2_contingency` de `scipy.stats` permet de faire tous ces calculs à la fois : la statistique `st`, la `p_value`, le degré de liberté `df` et le tableau de contingence théorique (qu'on va nommer `expected`) :

```
>>> st, p_value, df, expected = stat.chi2_contingency(gender_chol)

>>> expected                                    # Affichage du tableau de contingence
array([[ 6.27,  7.98, 42.75],                   # théorique
       [ 4.73,  6.02, 32.25]])
                                                # Affichage des autres valeurs
>>> df, st, p_value
(2, 0.6674544006245351, 0.7162491481220656)
```

Si on se fixe un risque d'erreur de 5% (0.05), la décision est prise comme suit :

```
>>> if p_value <= 0.05:
...  print ('Dépendance : rejet de H0')
... else:
...  print ('Indépendance : pas de rejet de H0')
...
Independance : pas de rejet de H0
```

Ce qui veut dire qu'on ne peut pas rejeter H_0, et que les deux variables sont indépendantes. On peut donc dire qu'il n'y pas de rapport entre le sexe et le taux de cholestérol.

Remarques :

La `p_value` se déduit de la statistique `st` de la façon suivante :

```
>>> p_value = 1 - stat.chi2.cdf(st, df)       # cdf :      cumulative distribution
>>> st, p_value                               #            function
(0.6674544006245351, 0.7162491481220656)
```

C'est l'équivalent de l'utilisation d'une table de chi 2. En effet, avec un calcul fait à la main, cela nécessite d'utiliser la table de chi 2 pour rechercher la `p_value` correspondante à la statistique `st` et au degré de liberté `df`.

Inversement, `st` peut être calculée à partir de la `p_value` en utilisant la fonction inverse de la fonction `cdf`, appelée `ppf` :

```
>>> st = stat.chi2.ppf(1 - p_value, df)
```

C'est de la même manière que la statistique critique (ou le seuil) est calculée à partir du risque d'erreur toléré 0.05 :

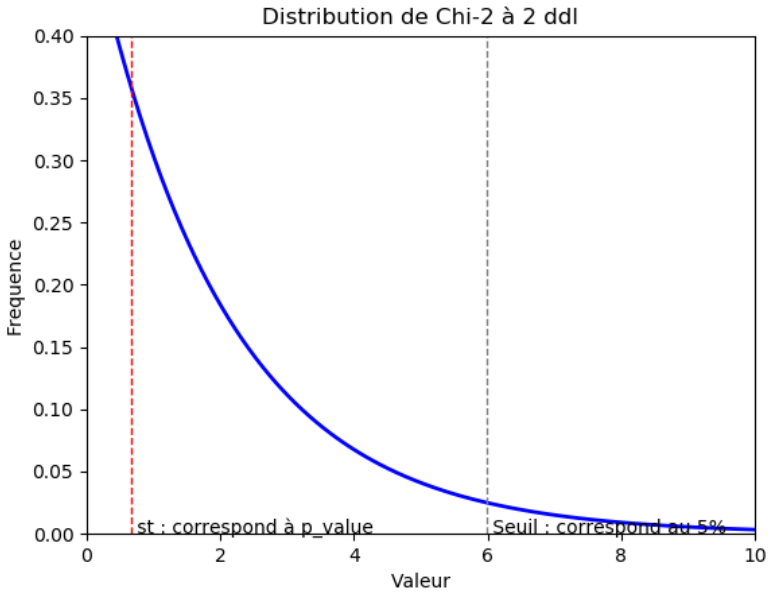
```
>>> critical = stat.chi2.ppf(1 - 0.05, df)
>>> critical
5.991464547107979
```

On voit bien que `st < critical`, synonyme du non rejet de l'hypothèse H_0.

Visualisation

Le code suivant permet de visualiser la distribution de chi 2 à `df` degré de liberté. La statistique `st` et le seuil correspondant au risque d'erreur toléré de 5% sont également visualisés (traits pointillés verticaux).

```
>>> x = np.linspace(0, 10, 100)                # Générer une série de 100 valeurs
                                               # comprises entre 0 et 10

                                               # Calculer la distribution (la pdf) de
                                               # chi 2 à df degré de liberté pour
                                               # toutes les valeurs de x, et
                                               # construire le plot correspondant
>>> plt.plot(x, stat.chi2.pdf(x, df), color='blue', linewidth=2, linestyle='-')

                                               # Labels des abscisses et des ordonnées
                                               # et titre du plot
>>> plt.xlabel('Valeur'), plt.ylabel('Frequence'), plt.title('Distribution de Chi-2 à
2 ddl')
                                               # Délimiter l'affichage du plot entre
>>> plt.xlim(0, 10), plt.ylim(0, 0.4)          # les valeurs indiquées

                                               # Tracer une ligne verticale pour
                                               # indiquer la valeur de st (qui
                                               # correspondant à la p_value)
>>> plt.axvline(x=st, color='red', linewidth=1, linestyle='--')

                                               # Tracer une ligne verticale pour
                                               # indiquer la valeur du seuil (qui
                                               # correspondant au 5%)
>>> plt.axvline(x=critical, color='gray', linewidth=1, linestyle='--')

                                               # Labels aux cotés des lignes
                                               # verticales tracées
>>> plt.text(st, 0, 'st : correspond à p_value'), plt.text(critical, 0, 'Seuil :
correspond au 5 %')

>>> plt.show()                                 # Affichage
```



A FAIRE : faites le même test pour vérifier s'il y a dépendance ou pas entre le taux de cholestérol et le taux de glycémie.

2. ANOVA à un facteur : variable qualitative vs variable quantitative

Considérons la variable quantitative `ap_hi` (pression artérielle systolique) et la variable qualitative `cholesterol`. La question est de savoir si ces deux variables sont indépendantes ou pas. Autrement dit, est ce que la pression artérielle a un lien avec le taux de cholestérol ou pas.

Etant donné que la variable `ap_hi` est quantitative, on peut décomposer cette variable en plusieurs séries, ou chaque série correspond à une modalité donnée de la variable qualitative `cholesterol` : `Normal`, `Abnormal` et `Critical`. Il en résulte donc trois séries : les mesures de la pression artérielle des personnes ayant respectivement un taux de cholestérol normal, anormal ou critique :

```
                                                # Filtrage de "table" en ne gardant que
                                                # les lignes correspondant à
                                                # 'choleterol' égal à 'Normal'
>>> t1 = table[table['cholesterol']=='Normal']
>>> s1 = t1['ap_hi']                            # Extraire la colonne 'ap_hi'

                                                # Filtrage de "table" en ne gardant que
                                                # les lignes correspondant à
                                                # 'choleterol' égal à 'Abnormal'
>>> t2 = table[table['cholesterol']=='Abnormal']
>>> s2 = t2['ap_hi']                            # Extraire la colonne 'ap_hi'

                                                # Filtrage de "table" en ne gardant que
                                                # les lignes correspondant à
                                                # 'choleterol' égal à 'Critical'
>>> t3 = table[table['cholesterol']=='Critical']
>>> s3 = t3['ap_hi']                            # Extraire la colonne 'ap_hi'
```

Dans le cas où les deux variables `ap_hi` et `cholesterol` sont indépendantes, les moyennes respectives des trois séries `s1`, `s2` et `s3` ne présenteront pas une différence significative. Inversement, une dépendance entre les deux variables se traduit par une différence significative entre les moyennes des trois séries.

L'hypothèse H_0 suppose donc que les trois séries ont une même moyenne (ou pas de différence significative entre les trois moyennes). Autrement dit, que les trois séries sont issues d'une même distribution (trois échantillons d'une même population). Il suffit qu'une des trois séries soit différente pour que l'hypothèse H_0 ne soit pas vérifiée.

La méthode d'ANOVA (ANalysis Of VAriance) à un facteur permet de réaliser ce test. Elle repose pour cela sur une analyse des variances. La statistique utilisée estime le rapport qui existe entre la variance inter-classes (la variation des échantillons entre elles) et la variance totale. Il est démontré que cette statistique (sous certaines conditions) suit une loi de Fisher-Snedecor.

La loi de Fisher-Snedecor est une famille de lois qui diffèrent selon **deux degrés de liberté**. Dans le cas notre exemple, la statistique `st` utilisée par ANOVA suit une loi de Fisher-Snedecor aux degrés de liberté suivants :

```
>>> df_inter = 3 - 1                          # Nombre de classes (de modalités) - 1

>>> df_intra = table['ap_hi'].size - 3        # Nombre de tous les individus (100) -
                                              # le nombre de classes
>>> df_inter, df_intra                        # Affichage
(2, 97)
```

La `p_value` est calculée en fonction de la statistique `st` calculée et de la loi de Ficher-Snedecor aux degrés de liberté `df_inter` et `df_intra`.

La fonction `f_oneway` de `scipy.stats` permet de faire ces calculs en un seul appel :

```
>>> st, p_value = stat.f_oneway(s1, s2, s3)   # Transmettre les trois séries
>>> st, p_value                               # Affichage
(10.05619599722633, 0.00010744613980808956)
```

Si on se fixe un risque d'erreur de 5% (0.05), la décision est prise comme suit :

```
>>> if p_value <= 0.05:
...  print ('Rejet de H0 : Moyennes différentes => Dépendance entre les 2 variables')
... else:
...  print ('Pas de rejet de H0 : Indépendance entre les deux variables')
...
Rejet de H0 : Moyennes différentes => Dépendance entre les 2 variables
```

Ce qui veut dire que l'hypothèse H_0 est à rejeter, et que les deux variables sont dépendantes. On peut donc dire qu'il y a un rapport entre la pression artérielle systolique et le taux de cholestérol.

Remarques :

La `p_value` se déduit de la statistique `st` de la façon suivante :

```
>>> p_value = 1 - stat.f.cdf(st, df_inter, df_intra)
                                              # cdf :      cumulative distribution
                                              #            function
>>> st, p_value
(10.05619599722633, 0.00010744613980808956)
```

C'est l'équivalent de l'utilisation d'une table de Fisher-Snedecor avec les deux degrés de liberté.

Inversement, `st` peut être calculée à partir de la `p_value` en utilisant la fonction inverse de la fonction `cdf`, appelée `ppf` :

```
>>> st = stat.f.ppf(1 - p_value, df_inter, df_intra)
```

C'est de la même manière que la statistique critique (ou le seuil) est calculée à partir du risque d'erreur toléré 0.05 :

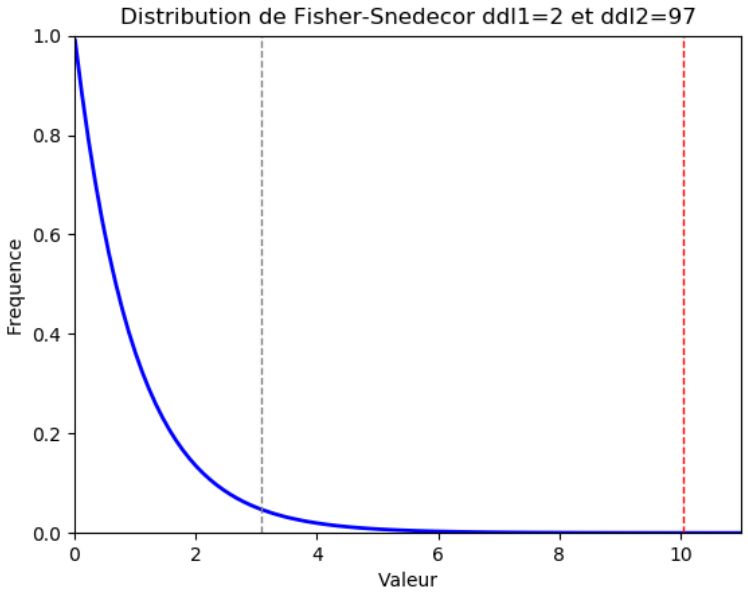
```
>>> critical = stat.f.ppf(1 - 0.05, df_inter, df_intra)
>>> critical
3.090186675154859
```

On voit bien que `st > critical`, synonyme du rejet de l'hypothèse H_0.

Visualisation

Le code suivant permet de visualiser la distribution de Fisher-Snedecor à `2 et 97` degrés de liberté. La statistique `st` et le seuil correspondant au risque d'erreur toléré de 5% sont visualisés également (traits pointillés verticaux).

```
>>> x = np.linspace(0.01, 11, 100)              # Générer une série de 100 valeurs
                                                # comprises entre 0.01 et 11

                                                # Calculer la distribution (la pdf) de
                                                # Fisher Snedecor à df_inter et
                                                # df_intra degrés de liberté pour
                                                # toutes les valeurs de x, et
                                                # construire le plot correspondant
>>> plt.plot(x, stat.f.pdf(x, df_inter, df_intra), color='blue', linewidth=2,
linestyle='-')

                                                # Labels des abscisses et des ordonnées
                                                # et titre du plot
>>> plt.xlabel('Valeur'), plt.ylabel('Frequence'), plt.title('Distribution de Fisher-
Snedecor ddl1=2 et ddl2=97')
                                                # Délimiter l'affichage du plot entre
>>> plt.xlim(0, 11), plt.ylim(0, 1)             # les valeurs indiquées

                                                # Tracer une ligne verticale pour
                                                # indiquer la valeur de st (qui
                                                # correspondant à la p_value)
>>> plt.axvline(x=st, color='red', linewidth=1, linestyle='--')

                                                # Tracer une ligne verticale pour
                                                # indiquer la valeur du seuil (qui
                                                # correspondant au 5%)
>>> plt.axvline(x=critical, color='gray', linewidth=1, linestyle='--')

>>> plt.show()                                  # Affichage
```



Remarques

Rem. 1

Dans cet exemple, nous avons supposé que les séries `s1`, `s2` et `s3` suivent une loi normale. Nous avons également supposé que leurs variances sont proches. En effet, ces deux suppositions sont nécessaires pour pouvoir utiliser le test d'ANOVA.

Pour être plus rigoureux, avant de procéder au test d'ANOVA, il convient d'utiliser d'autres tests statistiques (qui dépassent le cadre de ce cours) afin de vérifier ces suppositions. Il s'agit notamment du test de Shapiro-Wilk pour vérifier la normalité des échantillons (des trois séries), et du test de Levene pour vérifier si leurs variances sont proches.

Rem. 2

Dans le cadre du test d'ANOVA, le rejet de l'hypothèse H_0 veut dire que les moyennes des échantillons (des séries) sont significativement différentes. Pour que ça soit le cas, il suffit qu'un seul échantillon soit différent.

Dans ce cas, on peut pousser nos tests plus loin afin de savoir quels sont les échantillons qui sont différents des autres. Pour cela, il faut comparer les échantillons deux à deux entres eux. On peut utiliser, pour chaque comparaison, un test de Student.

Rem. 3

Le code suivant est synonyme au code précédent (extraction des trois séries `s1`, `s2` et `s3` de la table des données et application du test d'ANOVA), en étant plus générique :

```
>>> modalities = table['cholesterol'].unique()
>>> modalities
array(['Normal', 'Critical', 'Abnormal'], dtype=object)

>>> st, p_value = stat.f_oneway(*[list(table[table['cholesterol'] == m].ap_hi) for m
in modalities])

>>> st, p_value
(10.05619599722633, 0.00010744613980808956)
```

R. HACHEMI