

Test de conformité d'une moyenne.

Après usinage, on prélève **au hasard 100 pièces** dont on mesure le diamètre. Chaque pièce doit avoir un **diamètre de 58mm**. Après mesure de chacune d'entre elle, on s'aperçoit que le **diamètre moyen** est de **58.2mm**. Il semblerait que la qualité d'usinage des pièces ne soit pas au rendez-vous. Toutefois, il se peut aussi que cet écart entre le diamètre souhaité (58mm) et le diamètre observé (58.5mm) soit le fruit du hasard. Cela signifierai que la différence observée est due à la façon dont on a sélectionné les pièces : si d'autres pièces avait été sélectionnées on aurait observé une différence autre ou pas de différence du tout.

La **statistique inferentielle** permet de **lever se doute** entre **fruit du hasard** (échantillonnage) et phénomène observable dans la **population générale** (population parente).

Voici les données recueillies.

```
import pandas as pd
import numpy as np
import random
random.seed(10)

moy_cible = 58.00
n = 100

data = np.random.normal(loc=58.2, scale=0.2, size=n)
serie = pd.Series(data)

moy_obs = serie.mean()
print(moy_obs)

58.2031
```

Pour accéder au notebook de l'étude de cas, c'est par [ici](#).

Test d'hypothèses.

Pour répondre à la problématique du défaut d'usinage, on peut utiliser un **test de conformité à la moyenne**. Il s'agit de **vérifier que la différence observée n'est pas spécifique à l'échantillon sélectionné et donc imputable à la population globale**.

Selon les contextes, on peut vouloir vérifier que cette différence observée est strictement plus grande ou plus petite que le paramètre d'une population cible. Par exemple, s'il s'agit de vérifier la conformité d'une ampoule, il ne faut pas que sa durée de vie soit inférieure mais si elle est supérieure, tant mieux. À l'inverse, si on s'intéresse au poids moyen contenu dans un sachet de bonbon, ni l'entreprise, ni le client ne voudront être lésés : la quantité moyenne ne devra pas différer de celle convenu lors de la mise en production et de celle affichée sur l'emballage. En outre, c'est ce contexte qui va définir la structure du test statistique : soit il sera bilatéral, soit unilatéral (à gauche ou à droite).

Jeu des hypothèses.

Dans notre contexte, il s'agit d'un test bilatéral. Sous l'hypothèse nulle, les deux moyennes sont identiques : moyenne observée = moyenne de référence. L'hypothèse alternative (celle qu'on souhaite démontrer) stipule que les deux moyennes sont différentes. Formellement,

$$H_0 : \text{moy}_{\text{obs}} = \text{moy}_{\text{cible}},$$

$$H_1 : \text{moy}_{\text{obs}} \neq \text{moy}_{\text{cible}}.$$

La **moyenne cible** vaut **58mm** et la moyenne observée **58.2mm**.

Risque d'erreur.

Le **risque d'erreur** est un **risque auquel on accepte de se soumettre lors du rejet de l'hypothèse nulle**.

Précisément, avec un risque d'erreur de 5%, on accepte de rejeter l'hypothèse nulle à tort avec 5% de chances.

On fixe notre risque d'erreur α à 5%.

$$\alpha = 0.05$$

Validité du test.

Avant de fournir la statistique test, sachez que la validité du test repose sur la distribution normale de la moyenne observée. Ainsi, si votre **échantillon est suffisamment grand ($n > 30$)** vous pouvez appliquer le Théorème Central Limite (TCL) : il stipule qu'à mesure que l'échantillon croît (à l'infini), *la somme de variables aléatoires indépendantes et identiquement distribués tend vers une distribution gaussienne*^[1]. Cependant, si votre **échantillon est petit ($n < 30$)** vous devez vérifier que sa distribution est gaussienne pour ne pas compromettre la validité de vos conclusions.

Également, la statistique de test repose sur l'**écart type de la population**. Aussi, si vous ne connaissez pas la variance de la population vous devez fournir une **estimation sans biais** de ce paramètre. C'est-à-dire que vous devez corriger la variance de l'échantillon d'un facteur .

Dans notre cas, l'échantillon comprend 100 individus et la variance de la population est inconnue. Nous pouvons donc appliquer le TCL et utiliser une estimation sans biais de la variance.

Statistique de test.

La statistique de test est la suivante :

$$t = \frac{\text{moy}_{\text{obs}} - \text{moy}_{\text{cible}}}{\frac{s}{\sqrt{n}}},$$

où s est l'estimation sans biais de l'écart-type.

Cette statistique de test suit une **loi de Student à $n-1$ degrés de liberté** (ddl) : $t_{(n-1; \text{ddl}; 0.05)}$.

Calcul de la statistique de test t.

Pour calculer notre statistique de test il nous manque l'estimation sans biais de l'écart-type.

```
import numpy as np

std = np.std(serie , ddof=1)
```

On peut calculer la statistique de test.

```
t = (moy_obs - moy_cible) / (std/np.sqrt(n))
print(t)

9.036274695663673
```

Quantile de la loi de Student.

On va maintenant comparer cette statistique au quantile de la loi de Student pour n-1 ddl et un risque d'erreur de 5%.

```
from scipy import stats
quantile = stats.t.ppf(1-(0.05/2), n-1)
print(quantile)

1.9842169515086827
```

Avec un risque d'erreur de 5%, on peut rejeter l'hypothèse nulle d'égalité des moyennes. En effet, d'après les relevés effectués, nous avons suffisamment d'évidences pour conclure à la défaillance machine.

Pvalue.

Calculons la probabilité d'observer une statistique au moins aussi grande. Pour ça, on va utiliser la fonction de répartition cumulative.

```
pvalue = 2 * (1-stats.f.cdf(1-(0.05/2), 100-1))
print(pvalue)

1.3766765505351941e-14
```

La probabilité d'observer une statistique de test au moins aussi extrême est largement inférieur au seuil de singificativité de 5%. On peut donc conclure à la défaillance machine.

Intervalle de confiance.

Pour vérifier la conformité de notre usinage on peut utiliser une autre méthode inferentielle : l'intervalle de confiance. On fixe le risque d'erreur à 5%. L'intervalle de confiance à 95% est :

$$SI_{95} = moy_{obs} \pm t_{n-1 \text{ ddl} ; \alpha} * \frac{s}{\sqrt{n}}$$

Cet intervalle de confiance nous dit que si on utilisait 100 échantillons différents, 95 intervalles de confiance sur 100 contiendraient la vraie moyenne de la population. Autrement dit, si la moyenne cible n'appartient pas à l'IC_{95}, on a 5 chances sur 100 de conclure à tort à la défaillance de la machine.

On trouve,

```
ic_inf = moy_obs - quantile * (std/np.sqrt(n))  
  
ic_sup = moy_obs + quantile * (std/np.sqrt(n))  
  
print(ic_inf, moy_cible, ic_sup)  
  
58.171463255736086  
58.0  
58.26795148964892
```

Effectivement, la moyenne cible n'appartient pas à l'intervalle de confiance de notre moyenne observée. Il est donc peu probable qu'on conclut à tort à la défaillance machine.

Test de Student pour 1 échantillon.

Finalement, et pour vérifier notre démarche, on peut utiliser la fonction `ttest_1samp` du module `scipy.stats`. Avec cette fonction vous obtiendrez la statistique de test et la pvalue. Également, depuis la *version 1.10.0* de `scipy` vous disposez d'une méthode pour obtenir l'intervalle de confiance (avec un risque d'erreur donné) ainsi que le nombre de degrés de liberté.

```
from scipy.stats import ttest_1samp as test_conformite  
  
results = test_conformite(a=serie, popmean=moy_cible, alternative='two-  
sided', alpha=0.05)  
  
display(results.statistic, results.pvalue)  
  
9.036274695663673  
1.3814898377141245e-14  
  
display(results.df)  
  
99  
  
display(results.confidence_interval(confidence_level=0.95))  
  
ConfidenceInterval(low=58.171463255736086, high=58.26795148964892)
```

Nos observations précédentes étaient bonnes. Avec un risque d'erreur de 5%, on peut rejeter l'hypothèse nulle. On conclut donc à la défaillance significative de la machine.

Voilà, nous sommes arrivés au bout de notre Test de conformité à une moyenne.

À bientôt ! 🤖

[^1]: Par exemple, si on considère les distributions des salaires de 100 entreprises, celles-ci ont peu de chance d'être normales puisque des hauts salaires étirent notre distribution. Cependant, si on prend le salaire moyen de chacune des entreprises, alors il est très probable que sa distribution ressemble à une gaussienne.