Analyse de l'Impact d'une Nouvelle Landing Page sur le Taux de Conversion : A/B Testing et Évaluation Statistique

ABDOULAYE TANGARA

Student in MSc in Quantitative and Computable Economics

Contacts

LinkedIn

Mon GitHub

Mon Portfolio

abdoulayetangara722@gmail.com

"To get something you never had, you've got to do something you never did. Get your mind right"

Contexte

Une entreprise a développé une nouvelle page web afin d'augmenter le nombre d'utilisateurs qui « convertissent », c'est-à-dire qui décident d'acheter le produit de l'entreprise. Votre objectif est d'analyser ce carnet pour aider l'entreprise à déterminer s'il est judicieux de mettre en place cette nouvelle page, de conserver l'ancienne ou de prolonger l'expérience avant de prendre sa décision.

---

1 Objectif : *Augmenter le nombre d'achat grâce à une nouvelle page*

2 Les hypothèses

- ***H_0*** : La nouvelle page **n'a pas** d'effet significatif sur le taux de conversion.
- ***H_1*** : La nouvelle page **a un effet positif** sur le taux de conversion.

3 Element à tester : *taux de couversion*

Il s'agit d'un indicateur clé de performance (KPI) qui mesure le pourcentage d'utilisateurs ayant effectué une action souhaitée par rapport au nombre total de visiteurs ou de participants à une expérience. Ainsi, un taux de conversion élevé signifie une bonne efficacité de la stratégie mise en place, tandis qu'un taux faible indique un besoin d'optimisation (ex. : amélioration du design, changement du message marketing, simplification du processus d'achat).

[5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import random
import plotly.express as px
import scipy.stats as stat
from statsmodels.stats.proportion import proportions_ztest
import warnings
warnings.filterwarnings("ignore")
```

[6]:
```
data = pd.read_csv("ab_data.csv", delimiter=",")
data.head(5)
```

[6]:

	user_id	timestamp	group	landing_page	converted
0	851104	2017-01-21 22:11:48.556739	control	old_page	0
1	804228	2017-01-12 08:01:45.159739	control	old_page	0
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0
4	864975	2017-01-21 01:52:26.210827	control	old_page	1

🔍 Exploration des données

Analyse de la dimensionnalité de base de données

[7]:
```
# Analyse de la dimensionnalité de la base
print(f"""

La base de donnée utilisée dans cette analyse comprend  : {data.shape[0]}␣
 ↪lignes et {data.shape[1]} colonnes, dont les colonnes suivantes \n

| Variable | Description  | Type de données  | Exemple |
|----------|------------|-----------------|--------|
| `user_id` | Identifiant unique de chaque USER.|{data["converted"].dtype}|␣
 ↪10234 | | `timestamp`   | Date et heure à laquelle USER a visité la page.
 ↪|{data["timestamp"].dtype}| 2024-03-10 14:35:22 |
| `group`| Indique si l'USER appartient au groupe test ou contrôle.␣
 ↪|{data["group"].dtype}| {data["group"].unique()} |
| `landing_page`  | Page sur laquelle l'USER est arrivé␣
 ↪|{data["landing_page"].dtype}| {data["landing_page"].unique()} |
| `converted`| Indique si l'USER a effectué l'action␣
 ↪souhaitée|{data["converted"].dtype}| {data["converted"].unique()} |

""")

print("Statistique descriptive des variables\n")
data.describe(include="all")
```

La base de donnée utilisée dans cette analyse comprend : 294478 lignes et 5 colonnes, dont les colonnes suivantes

Variable	Description	Type de données	Exemple
`user_id`	Identifiant	int64	10234
`timestamp`	Date & heure	object	2024-03-10 14:35:22
`group`	Groupe de USER (test ou controle)	object	['control','treatment']
`landing_page`	Page web	object	['old_page','new_page']
`converted`	Action souhaitée de USER	int64	[0 1]

Statistique descriptive des variables

[7]:
```
              user_id                   timestamp      group landing_page  \
count   294478.000000                      294478     294478       294478
unique            NaN                      294478          2            2
top               NaN  2017-01-21 22:11:48.556739  treatment     old_page
freq              NaN                           1     147276       147239
mean    787974.124733                         NaN        NaN          NaN
std      91210.823776                         NaN        NaN          NaN
min     630000.000000                         NaN        NaN          NaN
25%     709032.250000                         NaN        NaN          NaN
50%     787933.500000                         NaN        NaN          NaN
75%     866911.750000                         NaN        NaN          NaN
max     945999.000000                         NaN        NaN          NaN

            converted
count   294478.000000
unique            NaN
top               NaN
freq              NaN
mean         0.119659
std          0.324563
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          1.000000
```

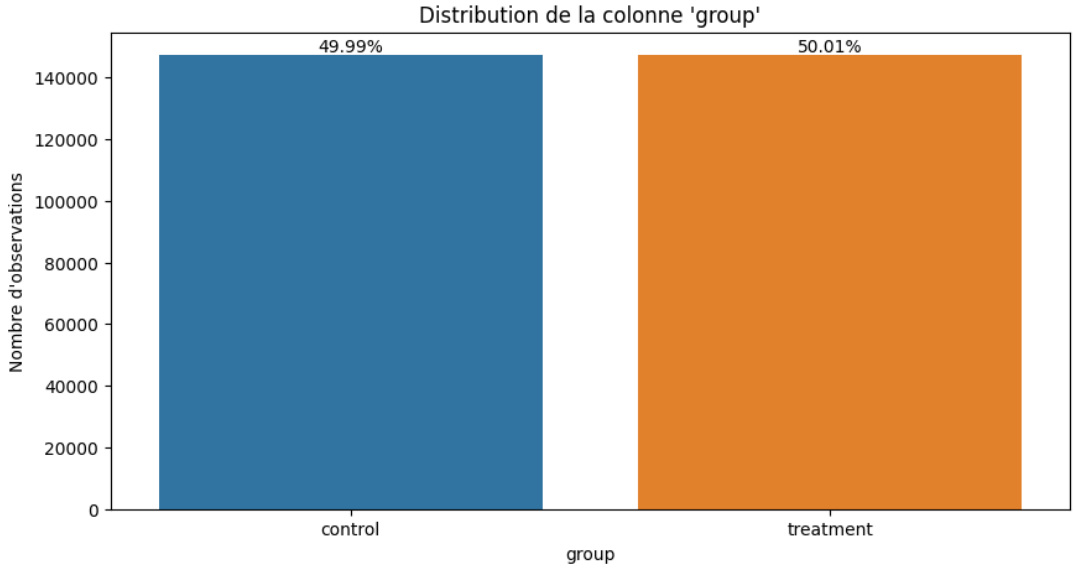
Traitement des variables : Nettoyage et Formatage

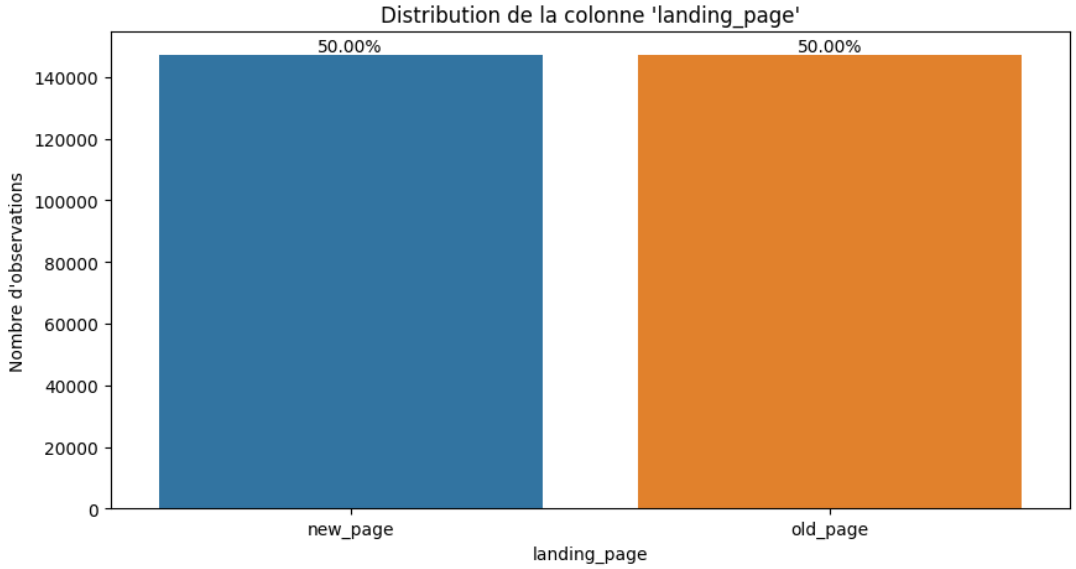
[8]:
```
# Suppression des doublons possible
data = data.drop_duplicates()

# Conversion des variables
data["timestamp"] = pd.to_datetime(data["timestamp"])
for col in ["group","landing_page","converted"]:
    data[col] = data[col].astype("category")


# Création de nouvelle variable
data["day_of_week"] = data["timestamp"].dt.day_name()
data["hour_in_day"] = data["timestamp"].dt.hour
```

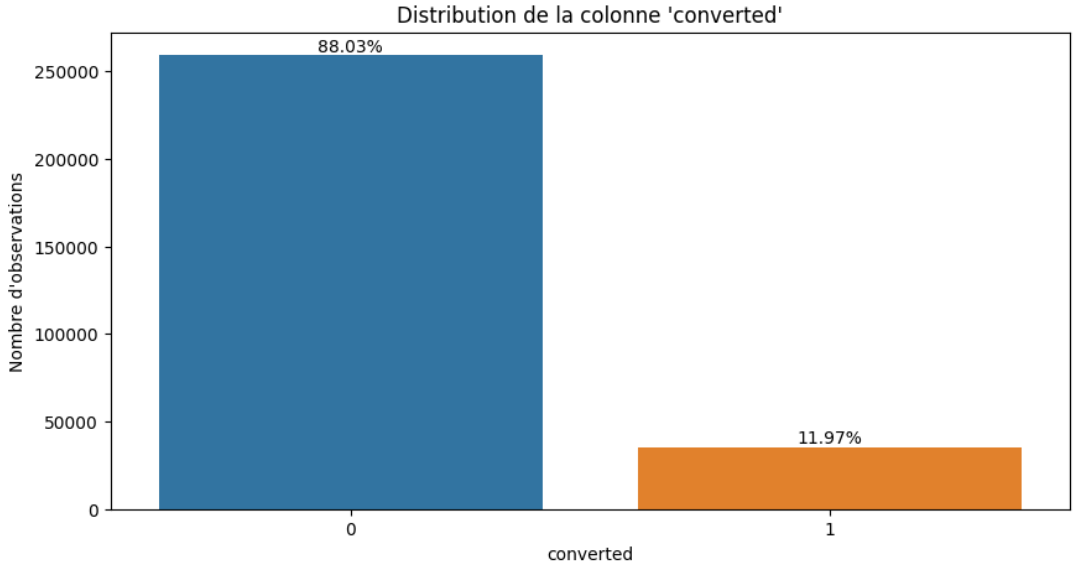
Analyse des KPIs

[9]:
```python
def kpi_1(df):
    for i in data.select_dtypes(include='category').columns:
        plt.figure(figsize=(10, 5))
        ax = sns.countplot(data=data, x=i, palette="tab10")

        total = len(data)

        for p in ax.patches:
            height = p.get_height()
            if height > 0:
                percentage = f"{100 * height / total:.2f}%"
                x = p.get_x() + p.get_width() / 2
                y = p.get_height()
                ax.annotate(percentage, (x, y), ha="center", va="bottom")

        plt.title(f"Distribution de la colonne '{i}'")
        plt.xlabel(i)
        plt.ylabel("Nombre d'observations")
        plt.show()
```

[10]:
```python
kpi_1(data)
```

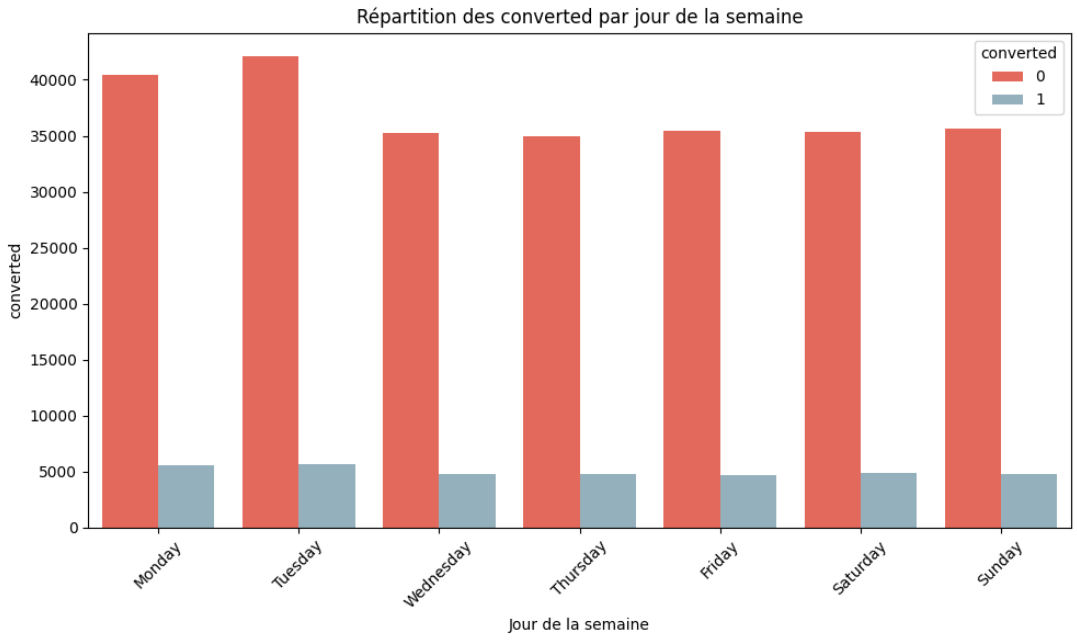


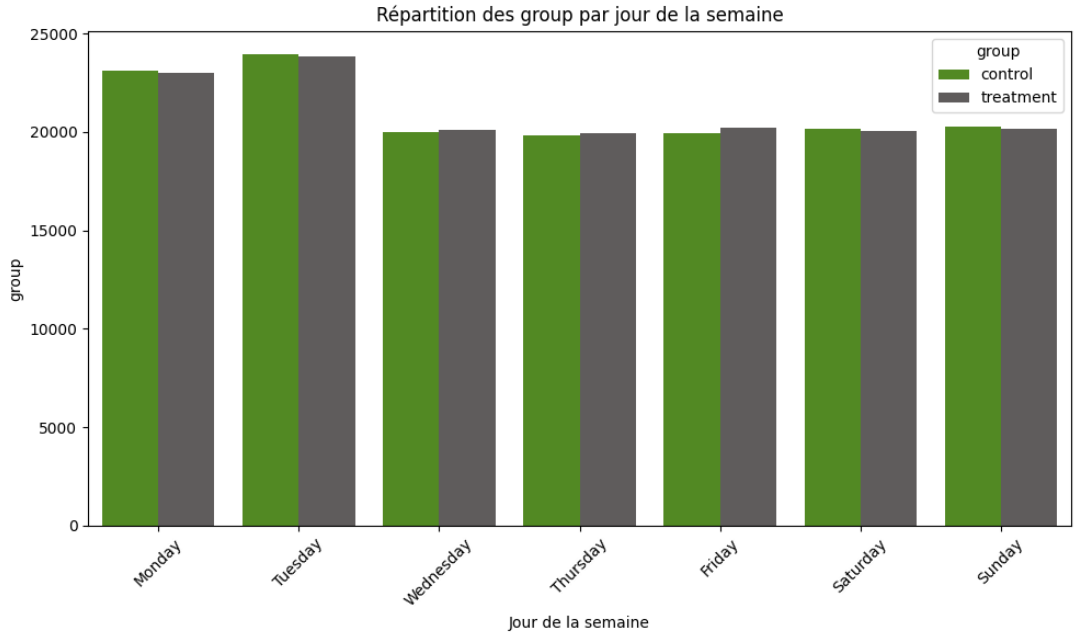

Distribution de la colonne 'landing_page'
50.00%
50.00%
140000
120000
100000
80000
60000
40000
20000
0
Nombre d'observations
new_page
old_page
landing_page

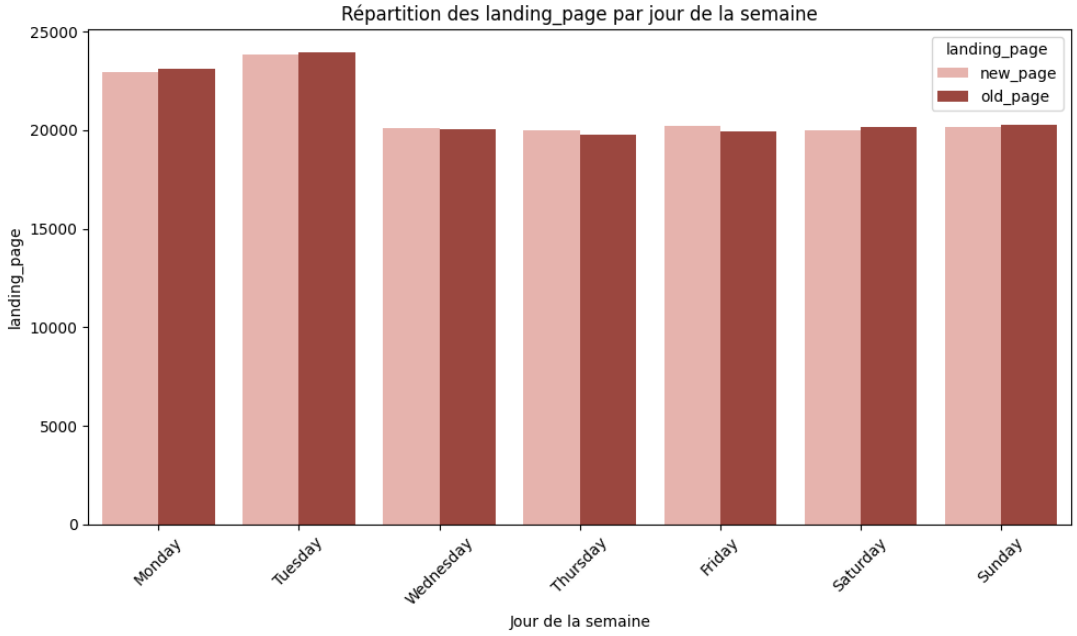

Distribution de la colonne 'converted'
88.03%
11.97%
250000
200000
150000
100000
50000
0
Nombre d'observations
0
1
converted

[11]:
```
def kpi_2(df):
    def couleur_aleatoire():
        return "#{:06x}".format(random.randint(0, 0xFFFFFF))

    for i in ["converted", "group", "landing_page"]:
        plt.figure(figsize=(10, 6))


        categories_hue = data[i].unique()
        palette = {cat: couleur_aleatoire() for cat in categories_hue}

        sns.countplot(x='day_of_week', hue=i, data=data,
                      order=['Monday', 'Tuesday', 'Wednesday', 'Thursday',␣
↪'Friday', 'Saturday', 'Sunday'],
                      palette=palette)
        plt.title(f"Répartition des {i} par jour de la semaine")
        plt.xlabel("Jour de la semaine")
        plt.ylabel(f"{i}")
        plt.legend(title=i)
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()
```

[12]:
```
kpi_2(data)
```




Répartition des group par jour de la semaine
group
control
treatment
25000
20000
15000
10000
5000
0
group
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Jour de la semaine


Répartition des landing_page par jour de la semaine
landing_page
new_page
old_page
25000
20000
15000
10000
5000
0
landing_page
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Jour de la semaine

Comparaison des proportions des deux groupes - Test Statistique

Le test Z permet de comparer deux proportions en évaluant si la proportion d'une caractéristique diffère significativement entre deux échantillons indépendants. Ce test repose sur le théorème central limite, selon lequel les proportions échantillonnées suivent une distribution normale asymptotique.

La statistique z est donnée par :

$$z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

Où :

- $\hat{p}_1$: proportion dans le premier échantillon,
- $\hat{p}_2$: proportion dans le deuxième échantillon,
- n_1, n_2 : tailles des échantillons,
- $\hat{p}$: proportion regroupée, calculée par $\hat{p} = \frac{x_1+x_2}{n_1+n_2}$.

Pour interpréter la p-valeur :

- Si p-valeur $< \alpha$, où α est le seuil de signification (généralement 0,05), on rejette l'hypothèse nulle (H_0) et on conclut qu'il existe une différence significative entre les proportions.
- Si p-valeur $\geq \alpha$, on ne rejette pas H_0, indiquant qu'aucune différence significative n'est détectée.

Justification du choix du Test Z:

L'élément de test étant le "taux de conversion", une variable binaire, la comparaison se fera sur la proportion de chaque groupe. Dans ce contexte, le test Z de proportions est le test statistique le plus approprié pour comparer les taux de conversion entre le groupe contrôle et le groupe test.

En effet, le test Z est utilisé lorsque les échantillons sont grands (dans notre cas : 294478 clients tout groupe confondus) et que la variable étudiée suit une loi binomiale pouvant être approximée par une loi normale sous certaines conditions. Ces conditions sont :

- Indépendance des observations : Chaque utilisateur appartient à un seul groupe (contrôle ou test), assurant l'indépendance des échantillons.
- Taille d'échantillon suffisante (théorème central limite) : Pour que l'approximation normale soit valide, il faut que chaque groupe contienne un nombre suffisamment élevé d'individus ayant converti et n'ayant pas converti, vérifiant ainsi la règle.

[13]:
```
# Repartition des groupes
## Groupe A
group_A = data[data["group"] == "control"]
n_group_A = group_A.shape[0]
cov_group_A = group_A[data["converted"]==1].shape[0]

## Groupe B
group_B = data[data["group"] == "treatment"]
n_group_B = group_B.shape[0]
cov_group_B = group_B[data["converted"]==1].shape[0]

# Test statistique
successes = np.array([cov_group_A, cov_group_B])
samples = np.array([n_group_A, n_group_B ])
stat, p_value = proportions_ztest(successes, samples, alternative="two-sided")
print("Statistique de test (z) :", stat)
print("p-value :", p_value)

# Interprétation
alpha = 0.05
if p_value < alpha:
    print("✔ Résultat significatif : La nouvelle page a un effet sur le taux␣
 ↪de conversion.")
else:
    print("✖ Pas de résultat significatif : La nouvelle page n'améliore pas␣
 ↪significativement le taux de conversion.")
```

```
Statistique de test (z) : 1.237
p-value : 0.216
✖ Pas de résultat significatif : La nouvelle page n'améliore pas␣
 ↪significativement le taux de conversion.
Autrement dit, Avec un échantillon aussi large, une p-value élevée,indique que␣
 ↪la nouvelle page n'a probablement aucun impact business, même si on␣
 ↪prolongeait le test.
```

Intervalle de confiance de la différence des proportions et Erreur de Type II, β

$$(\hat{p}_1 - \hat{p}_2) \pm z_{\alpha/2}\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}$$

Où :

- $z_{\alpha/2}$: est la valeur critique de la distribution normale standard (par exemple, 1,96 pour un niveau de confiance de 95%).

L'EDM pour l'utilisation de la formule du test Z (bilatéral) pour comparer deux proportions, en incorporant des valeurs critiques pour α et $1-\beta$, et les erreurs standard des proportions :

$$\text{MDE} = |p_1 - p_2| = z_{1-\alpha/2}\sqrt{p_0(1-p_0)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)} + z_{1-\beta}\sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}}$$

Où :

- $z_{1-\alpha/2}$: valeur critique pour le niveau de signification.
- $z_{1-\beta}$: quantile pour la puissance désirée.
- $p_0 = p_1 = p_2$: en supposant que la valeur nulle est correcte.

[23] :
```
# Calcul des proportions
p_groupe_A = cov_group_A/ n_group_A
p_groupe_B = cov_group_B/ n_group_B
diff_obs = p_groupe_A - p_groupe_B

print(f"Différence observée: {diff_obs:.4f}")

n_A, cov_A = 147202, 17723  # Groupe contrôle
n_B, cov_B = 147276, 17514  # Groupe test

# Erreur standard (SE) et IC automatique
ci_diff = confint_proportions_2indep(cov_B, n_B, cov_A, n_A, method='wald')
SE = (ci_diff[1] - ci_diff[0]) / (2 * 1.96)  # Calcul dérivé de l'IC

print(f"SE calculée: {SE:.5f}")
print(f"IC différence (Wald): [{ci_diff[0]:.4f}, {ci_diff[1]:.4f}]")
```

```
Différence observée: 0.0015
SE calculée: 0.00120
IC différence (Wald): [-0.0038, 0.0009]
```

1. Différence Observée (0.0015 ou 0.15%) La nouvelle page a un taux de conversion supérieur de 0.15% par rapport à l'ancienne. Mais : Cette différence est extrêmement faible
2. Erreur Standard (SE = 0.00120 soit 0.12%) La marge d'erreur sur la différence estimée est de ±0.12%. Cela nous indique que la mesure est précise (grâce à la grande taille d'échantillon).
3. Intervalle de Confiance à 95% (Wald) : [-0.38%, 0.09%] La vraie différence de taux de conversion a 95% de chances de se situer entre -0.38% et +0.09%.
   Vue que l'IC inclut 0 (et même des valeurs négatives), nous pouvons donc concluire que la nouvelle page n'a aucun effet statistiquement significatif.