CAP 2

Implementar una nueva funcionalidad

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot, font

Description automatically generated

A picture containing text, screenshot, font, document

Description automatically generated A picture containing text, screenshot, font, diagram

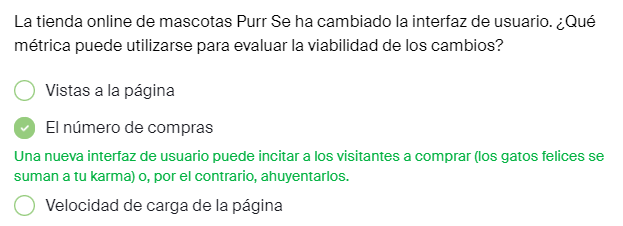
Description automatically generated



A white background with black text

Description automatically generated with low confidence



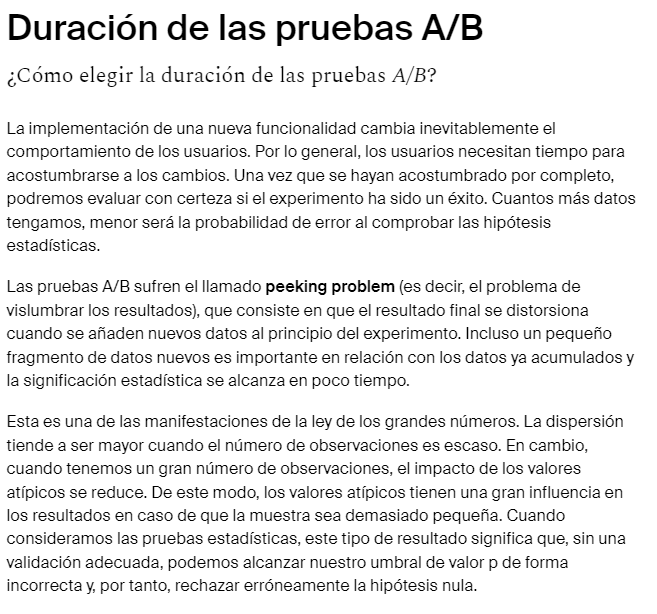


A picture containing text, screenshot, font, document

Description automatically generated

A screenshot of a computer screen

Description automatically generated with low confidence

 A picture containing text, screenshot, diagram, line

Description automatically generated

A picture containing text, screenshot, font, document

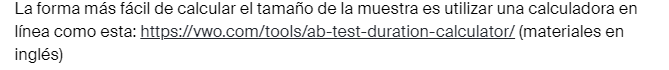
Description automatically generated

A diagram of a flowchart

Description automatically generated with low confidence

A picture containing text, screenshot, diagram, font

Description automatically generated



<https://vwo.com/tools/ab-test-duration-calculator/>

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a survey

Description automatically generated with medium confidence A picture containing text, screenshot, font

Description automatically generated

A picture containing text, screenshot, font, document

Description automatically generated

A screenshot of a survey

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a survey

Description automatically generated with low confidence

A picture containing text, screenshot, font, number

Description automatically generated



A picture containing text, screenshot, font, document

Description automatically generated



A screenshot of a computer

Description automatically generated with medium confidence

sample\_after = pd.Series([

439, 518, 452, 505, 493, 470, 498, 442, 497,

423, 524, 442, 459, 452, 463, 488, 497, 500,

476, 501, 456, 425, 438, 435, 516, 453, 505,

441, 477, 469, 497, 502, 442, 449, 465, 429,

442, 472, 466, 431, 490, 475, 447, 435, 482,

434, 525, 510, 494, 493, 495, 499, 455, 464,

509, 432, 476, 438, 512, 423, 428, 499, 492,

493, 467, 493, 468, 420, 513, 427])

print("La media de antes:", sample\_before.mean())

print("La media de después:", sample\_after.mean())

# nivel crítico de significación

# la hipótesis se rechaza si el valor p es menor que ese

alpha = .05

results = st.ttest\_ind(sample\_before, sample\_after)

pvalue = results.pvalue/2

print('p-value: ', pvalue)

if pvalue < alpha:

print(

"La hipótesis nula se rechaza, a saber, es probable que el importe promedio de las compras aumente"

)

else:

print(

"La hipótesis nula no se rechaza, a saber, es poco probable que el importe medio de las compras aumente"

)

La media de antes: 396.9714285714286

La media de después: 470.5285714285714

p-value: 3.090979110409132e-29

La hipótesis nula se rechaza, a saber, es probable que el importe promedio de las compras aumente

A picture containing text, font, screenshot, document

Description automatically generated



A picture containing text, screenshot, font

Description automatically generated



A picture containing text, screenshot, diagram, line

Description automatically generated

A picture containing text, screenshot, font, algebra

Description automatically generated



A picture containing text, screenshot, font, document

Description automatically generatedA picture containing text, screenshot, font, document

Description automatically generatedA picture containing text, receipt, font, white

Description automatically generated



A screenshot of a computer

Description automatically generated with low confidence

import pandas as pd

from scipy import stats as st

sample = pd.Series([

439, 518, 452, 505, 493, 470, 498, 442, 497,

423, 524, 442, 459, 452, 463, 488, 497, 500,

476, 501, 456, 425, 438, 435, 516, 453, 505,

441, 477, 469, 497, 502, 442, 449, 465, 429,

442, 472, 466, 431, 490, 475, 447, 435, 482,

434, 525, 510, 494, 493, 495, 499, 455, 464,

509, 432, 476, 438, 512, 423, 428, 499, 492,

493, 467, 493, 468, 420, 513, 427])

print('Media:', sample.mean())

confidence\_interval =st.t.interval(

0.95, len(sample)-1,sample.mean(),sample.sem())

print('Intervalo de confianza del 95 %:', confidence\_interval)

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, screenshot, font, document

Description automatically generatedA picture containing text, screenshot, diagram, font

Description automatically generated



A picture containing text, font, screenshot, algebra

Description automatically generated



A picture containing text, screenshot, font

Description automatically generated



A picture containing text, screenshot, font

Description automatically generatedA screenshot of a computer

Description automatically generated with low confidence



A screenshot of a computer code

Description automatically generated with low confidence

A screenshot of a chat

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

example\_data = pd.Series([1, 2, 3, 4, 5])

print('Sin reemplazo')

print(example\_data.sample(frac=1, replace=False, random\_state=state))

print('Con reemplazo')

print(example\_data.sample(frac=1, replace=True, random\_state=state))

A screenshot of a computer

Description automatically generated with medium confidence

import pandas as pd

import numpy as np

data = pd.Series([

10.7 , 9.58, 7.74, 8.3 , 11.82, 9.74, 10.18, 8.43, 8.71,

6.84, 9.26, 11.61, 11.08, 8.94, 8.44, 10.41, 9.36, 10.85,

10.41, 8.37, 8.99, 10.17, 7.78, 10.79, 10.61, 10.87, 7.43,

8.44, 9.44, 8.26, 7.98, 11.27, 11.61, 9.84, 12.47, 7.8 ,

10.54, 8.99, 7.33, 8.55, 8.06, 10.62, 10.41, 9.29, 9.98,

9.46, 9.99, 8.62, 11.34, 11.21, 15.19, 20.85, 19.15, 19.01,

15.24, 16.66, 17.62, 18.22, 17.2 , 15.76, 16.89, 15.22, 18.7 ,

14.84, 14.88, 19.41, 18.54, 17.85, 18.31, 13.68, 18.46, 13.99,

16.38, 16.88, 17.82, 15.17, 15.16, 18.15, 15.08, 15.91, 16.82,

16.85, 18.04, 17.51, 18.44, 15.33, 16.07, 17.22, 15.9 , 18.03,

17.26, 17.6 , 16.77, 17.45, 13.73, 14.95, 15.57, 19.19, 14.39,

15.76])

state = np.random.RandomState(12345)

for i in range(10):

subsample = data.sample(frac=1,replace=True,random\_state=state)

print(subsample.quantile(.99))

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a chat

Description automatically generated with medium confidence

state = np.random.RandomState(12345)

# Guarda los valores del cuantil del 99 % en la variable de valores

values = []

for i in range(1000):

subsample = data.sample(frac=1, replace=True, random\_state=state)

values.append(subsample.quantile(0.99))

values=pd.Series(values)

lower =values.quantile(0.05)

upper =values.quantile(.985)

print(lower)

print(upper)

A screenshot of a message

Description automatically generated with low confidence

A picture containing text, screenshot, font, document

Description automatically generated

A picture containing text, font, screenshot, number

Description automatically generated

A picture containing text, receipt, font, screenshot

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

import pandas as pd

import numpy as np

# datos del grupo de control A

samples\_A = pd.Series([

98.24, 97.77, 95.56, 99.49, 101.4 , 105.35, 95.83, 93.02,

101.37, 95.66, 98.34, 100.75, 104.93, 97. , 95.46, 100.03,

102.34, 98.23, 97.05, 97.76, 98.63, 98.82, 99.51, 99.31,

98.58, 96.84, 93.71, 101.38, 100.6 , 103.68, 104.78, 101.51,

100.89, 102.27, 99.87, 94.83, 95.95, 105.2 , 97. , 95.54,

98.38, 99.81, 103.34, 101.14, 102.19, 94.77, 94.74, 99.56,

102. , 100.95, 102.19, 103.75, 103.65, 95.07, 103.53, 100.42,

98.09, 94.86, 101.47, 103.07, 100.15, 100.32, 100.89, 101.23,

95.95, 103.69, 100.09, 96.28, 96.11, 97.63, 99.45, 100.81,

102.18, 94.92, 98.89, 101.48, 101.29, 94.43, 101.55, 95.85,

100.16, 97.49, 105.17, 104.83, 101.9 , 100.56, 104.91, 94.17,

103.48, 100.55, 102.66, 100.62, 96.93, 102.67, 101.27, 98.56,

102.41, 100.69, 99.67, 100.99])

# datos del grupo experimental B

samples\_B = pd.Series([

101.67, 102.27, 97.01, 103.46, 100.76, 101.19, 99.11, 97.59,

101.01, 101.45, 94.8 , 101.55, 96.38, 99.03, 102.83, 97.32,

98.25, 97.17, 101.1 , 102.57, 104.59, 105.63, 98.93, 103.87,

98.48, 101.14, 102.24, 98.55, 105.61, 100.06, 99. , 102.53,

101.56, 102.68, 103.26, 96.62, 99.48, 107.6 , 99.87, 103.58,

105.05, 105.69, 94.52, 99.51, 99.81, 99.44, 97.35, 102.97,

99.77, 99.59, 102.12, 104.29, 98.31, 98.83, 96.83, 99.2 ,

97.88, 102.34, 102.04, 99.88, 99.69, 103.43, 100.71, 92.71,

99.99, 99.39, 99.19, 99.29, 100.34, 101.08, 100.29, 93.83,

103.63, 98.88, 105.36, 101.82, 100.86, 100.75, 99.4 , 95.37,

107.96, 97.69, 102.17, 99.41, 98.97, 97.96, 98.31, 97.09,

103.92, 100.98, 102.76, 98.24, 97. , 98.99, 103.54, 99.72,

101.62, 100.62, 102.79, 104.19])

# diferencia real entre las medias de los grupos

AB\_difference = samples\_B.mean()-samples\_A.mean()

print("Diferencia entre los importes promedios de compra:", AB\_difference)

alpha = .05

state = np.random.RandomState(12345)

bootstrap\_samples = 1000

count = 0

for i in range(bootstrap\_samples):

# concatena las muestras

united\_samples =pd.concat([samples\_A,samples\_B])

# crea una submuestra

subsample = united\_samples.sample(frac=1, replace=True, random\_state= state)

# divide la submuestra por la mitad

subsample\_A = subsample[:len(samples\_A)]

subsample\_B = subsample[len(samples\_B):]

# encuentra la diferencia entre las medias

bootstrap\_difference = subsample\_B.mean() - subsample\_A.mean()

# si la diferencia no es menor que la diferencia real, añade "1" al contador

if bootstrap\_difference >= AB\_difference:

count += 1

# el valor p es igual al porcentaje de valores excedentes

pvalue = 1. \* count / bootstrap\_samples

print('p-value =', pvalue)

if pvalue < alpha:

print("La hipótesis nula se rechaza, a saber, es probable que el importe promedio de las compras aumente")

else:

print("La hipótesis nula no se rechaza, a saber, es poco probable que el importe medio de las compras aumente")

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, screenshot, font, number

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

import pandas as pd

def revenue(target, probabilities, count):

probs\_sorted = probabilities.sort\_values(ascending=False)

selected = target[probs\_sorted.index][:count]

return 10 \* selected.sum()

target = pd.Series([1, 1, 0, 0, 1, 0])

probab = pd.Series([0.2, 0.9, 0.8, 0.3, 0.5, 0.1])

res = revenue(target, probab, 3)

print(res)

A screenshot of a phone

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

import pandas as pd

import numpy as np

# Abre los archivos

# toma el índice “0” para convertir los datos a pd-Series

target = pd.read\_csv('/datasets/eng\_target.csv')['0']

probabilities = pd.read\_csv('/datasets/eng\_probabilites.csv')['0']

def revenue(target, probabilities, count):

probs\_sorted = probabilities.sort\_values(ascending=False)

selected = target[probs\_sorted.index][:count]

return 10 \* selected.sum()

state = np.random.RandomState(12345)

values = []

for i in range(1000):

target\_subsample= target.sample(n=25, replace=True,random\_state=state)

prob\_subsample= probabilities[target\_subsample.index]

values.append(revenue(target\_subsample,prob\_subsample,10))

values = pd.Series(values)

lower = values.quantile(.01)

mean = values.mean()

print("Ingresos promedio:", mean)

print("Cuantil del 1 %:", lower)

A screenshot of a phone

Description automatically generated with low confidence

A picture containing text, screenshot, font, document

Description automatically generated