Resumen del capítulo: Analizar los resultados del Test A/B

Probar la hipótesis de que las proporciones son iguales

Otra tarea típica en estadística es probar hipótesis sobre la igualdad de proporciones de las poblaciones. Si una parte de una población estadística tiene una determinada característica y otra parte no, podemos sacar conclusiones sobre el tamaño de esa proporción en función de una muestra tomada de la población. Al igual que con la media, las proporciones de la muestra se distribuirán normalmente alrededor de la real. Python no tiene una prueba estándar para esto, así que escribiremos una.

La diferencia entre las proporciones que observamos en las muestras será la **estadística**. Así es como llamamos a una variable cuyos valores solo se pueden encontrar a partir de datos de muestra. Puedes comprobar que se distribuye normalmente:

$$Z \cong \frac{(P_1 - P_2) - (\pi_1 - \pi_2)}{\sqrt{P(1-P)(1/n_1 - 1/n_2)}} \sim N(0,1)$$

`Z` es el valor estándar para un criterio con una distribución normal estándar, donde la media es 0 y la desviación estándar es 1. Todo esto se indica en la parte derecha de la fórmula después del signo '∼', que significa que la expresión se distribuye como **N**(0,1).

`n₁` y `n₂` representan los tamaños de las dos muestras que se comparan (el número de observaciones que contienen). `P₁` y `P₂` son las proporciones observadas en las muestras y `P` es la proporción en la muestra formada por `P₁` y `P₂`. `π₁` y `π₂` son las proporciones mismas en las poblaciones que estamos comparando.

En el caso de las pruebas A/B, generalmente se prueba la hipótesis de que π₁ = π₂. Entonces, si la hipótesis nula es cierta, la expresión `(π₁ - π₂)` en el nominador será igual a 0 y será posible calcular el criterio usando solo los datos de la muestra.

La estadística obtenida será distribuida normalmente, lo que permitirá realizar pruebas bilaterales y unilaterales. Usando la misma hipótesis nula de que las proporciones de dos poblaciones son iguales, podemos probar las hipótesis alternativas de que 1) las proporciones simplemente no son iguales o que 2) una proporción es mayor o menor que la otra.

```
from scipy import stats as st
import numpy as np
import math as mth

alpha = .05 # nivel de significación

successes = np.array([78, 120])
trials = np.array([830, 909])

# proporción de éxito en el primer grupo:
p1 = successes[0]/trials[0]

# proporción de éxito en el segundo grupo:
p2 = successes[1]/trials[1]

# proporción de éxito en el dataset unido:
p_combined = (successes[0] + successes[1]) / (trials[0] + trials[1])

# la diferencia entre las proporciones de los datasets
difference = p1 - p2
```

Calculemos la estadística en términos de desviaciones estándar de la distribución normal estándar:

```
# calcula la estadística en desviaciones estándar de la distribución normal estándar
z_value = difference / mth.sqrt(p_combined * (1 - p_combined) * (1/trials[0] + 1/trials[1]))

# establece la distribución normal estándar (media 0, desviación estándar 1)
distr = st.norm(0, 1)
```

Si las proporciones fueran iguales, la diferencia entre ellas sería 0. Calculemos qué tan lejos de 0 resultó estar nuestra estadística. ¿Cuál es la probabilidad de obtener tal diferencia o una mayor? Dado que la distribución de la estadística es normal, llamaremos al método `cdf()`. Obtendremos el valor absoluto de la estadística utilizando el método `abs()`. Esto asegurará que obtengamos el resultado correcto sin importar que signo tenga la estadística. (Podría ser negativo ya que la prueba es bilateral.) Por la misma razón, duplicamos el resultado:

```
# calcula la estadística en desviaciones estándar de la distribución normal estándar
z_value = difference / mth.sqrt(p_combined * (1 - p_combined) * (1/trials[0] + 1/trials[1]))

# establece la distribución normal estándar (media 0, desviación estándar 1)
distr = st.norm(0, 1)

p_value = (1 - distr.cdf(abs(z_value))) * 2

print('p-value: ', p_value)

if (p_value < alpha):
    print("Rechazar la hipótesis nula: hay una diferencia significativa entre las proporciones")
```

```
else:
    print("No se pudo rechazar la hipótesis nula: no hay razón para pensar que las proporciones so
n diferentes")
```

Pruebas de normalidad. Test de Shapiro–Wilk

En la vida real, muchas variables divergen de la distribución normal: tienen valores atípicos que no se pueden ignorar. Para probar si los datasets están modelados con precisión por la distribución normal, utilizamos **pruebas de normalidad**.

Según el teorema del límite central, las medias muestrales se distribuyen normalmente alrededor de la media verdadera de la población (y las proporciones muestrales alrededor de la proporción verdadera). Esto es cierto incluso para las distribuciones que contienen grandes valores atípicos. Este enfoque es bueno si estás sacando conclusiones basadas en docenas de muestras. Cada muestra individual podría contener valores atípicos que cambiarán los resultados y los afectarán.

Primero, debes aprender a probar la hipótesis de que se tomó una muestra de una población distribuida normalmente.

Un criterio simple es χ^2 (chi-cuadrado). La suma de las diferencias al cuadrado entre los valores observados y esperados se divide entre los valores esperados:

$$\frac{\sum(O_i - E_i)^2}{E_i}$$

O en esta fórmula son valores observados mientras que E son valores esperados. Se supone que la diferencia entre los valores esperados y observados normalmente se distribuye alrededor de 0, de modo que la probabilidad de desviaciones disminuye a medida que te alejas de los valores esperados en cualquier dirección. Como resultado, este criterio se distribuirá como la suma de *n* distribuciones normales estándar al cuadrado, donde n es el número de observaciones en una muestra. Esta es la **distribución chi-cuadrado**.

Chi-cuadrado es un criterio común pero hay uno mejor para las pruebas de normalidad: **el test de Shapiro–Wilk.** Su ventaja es que es más potente que chi-cuadrado si el nivel de significación es fijo: descubre diferencias entre distribuciones con más frecuencia si hay alguna por descubrir. Este criterio es significativamente más complejo y es más fácil probar su alto poder en varios datasets que demostrarlo teóricamente. El cálculo del criterio de Shapiro-Wilk está integrado en el módulo estándar de `scipy.stats`. Vamos a ver cómo funciona en la práctica.

`sample_1` almacena datos sobre el número de sesiones de un usuario por semana en un sitio web durante un año. Usemos el método **st.shapiro(x)** para probar si la variable puede considerarse normalmente distribuida:

```
from scipy import stats as st

alpha = .05 # nivel de significación

results = st.shapiro(sample_1)
p_value = results[1] # el segundo valor en la matriz de resultados (con índice 1) - el valor p

print('p-value: ', p_value)

if (p_value < alpha):
    print("Hipótesis nula rechazada: la distribución no es normal")
else:
    print("No se pudo rechazar la hipótesis nula: la distribución parece ser normal")
```

La prueba no paramétrica de Wilcoxon-Mann-Whitney

Cuando tus datos contienen valores atípicos grandes (en comparación con la distribución normal), las métricas algebraicas no funcionan muy bien. Es cierto que tienen en cuenta todos los valores pero ahí radica su debilidad: un valor atípico puede desequilibrarlo todo.

Los criterios algebraicos para probar hipótesis sobre la normalidad de los datos originales, como chi-cuadrado y la **prueba de Shapiro-Wilk**, son paramétricos, ya que utilizas una muestra para evaluar los parámetros de la distribución esperada (por ejemplo, la media).

También puedes utilizar un **enfoque estructural** o una prueba no paramétrica. El método que usaremos para las pruebas A/B se llama **st.mannwhitneyu()** (la prueba U de Mann-Whitney).

La idea clave detrás de la prueba es clasificar dos muestras en orden ascendente y comparar los rangos de valores que aparecen en ambas muestras (es decir, en qué posición aparecen en las muestras). Si las diferencias entre sus rangos son las mismas de una muestra a otra, esto significa que el cambio es **típico**. Eso significa que simplemente agregaron algunos valores, lo que provocó que el resto cambiase.

Los cambios **no típicos** significan que ocurrió un cambio real. La suma de tales cambios de rango (del n.° 1 al n.° 4 sería 3, etc.) es el valor del criterio. Cuanto mayor sea, mayor será la probabilidad de que las distribuciones de las dos muestras difieran.

Las probabilidades de obtener varios valores de una prueba de Mann-Whitney se han calculado teóricamente, lo que nos permite concluir que existe o no existe una diferencia para cualquier nivel de significación que se haya establecido.

Los métodos no paramétricos son útiles porque no hacen suposiciones sobre cómo se distribuyen los datos por lo que no es necesario estimar los parámetros de distribución. Dichos

métodos a menudo se usan cuando es difícil (o incluso imposible) estimar parámetros debido a una gran cantidad de valores atípicos.

```
alpha = .05 # nivel de significancia

results = st.mannwhitneyu(messages_old, messages_new)

print('p-value: ', results.pvalue)

if (results.pvalue < alpha):
    print("Hipótesis nula rechazada: la diferencia es estadísticamente significativa")
else:
    print("No se pudo rechazar la hipótesis nula: no podemos sacar conclusiones sobre la diferencia")
```

Estabilidad de las métricas acumuladas

Para evitar el peeking problem, los analistas examinan los gráficos.

Analizan gráficos de métricas **acumuladas**. Digamos que una prueba duró dos semanas. Si creas un gráfico con datos **acumulados**, en el punto del primer día tendrás los valores de las métricas para ese día, en el punto del segundo día tendrás la suma de las métricas de los dos primeros días y así adelante. De esa manera, puedes realizar un seguimiento de los cambios en los resultados del experimento cada día de la prueba.

De acuerdo con el teorema del límite central, los valores de las métricas acumuladas a menudo convergen y se establecen alrededor de una media particular. Después, un gráfico de métrica acumulada puede ayudarte a decidir si continuar con la prueba.

Para que las diferencias entre los grupos sean más obvias, los analistas trazan **gráficos de diferencias relativas**. Cada uno de sus puntos se calcula de la siguiente manera: `métrica acumulada del grupo B / métrica acumulada del grupo A - 1`.

Otra función bastante útil para calcular métricas acumuladas es **np.logical_and()**.
Nos permite aplicar operaciones booleanas a objetos Series. Esto es útil cuando necesitas elegir un subconjunto de filas de una tabla que coincida con varias condiciones.

```
np.logical_and(first_condition, second_condition)
np.logical_or(first_condition, second_condition)
np.logical_not(first_condition)
```

Digamos que tenemos dos DataFrames, `orders` y `visitors`, que contienen datos sobre pedidos de una tienda en línea y visitantes recopilados durante una prueba A/B. Vamos a crear un nuevo DataFrame con datos acumulados:

```
# obtén los datos diarios acumulados agregados sobre los pedidos
ordersAggregated = datesGroups.apply(lambda x: orders[np.logical_and(orders['date'] <= x['date'],
 orders['group'] == x['group'])].agg({'date' : 'max', 'group' : 'max', 'orderId' : pd.Series.nuniq
ue, 'userId' : pd.Series.nunique, 'revenue' : 'sum'}), axis=1).sort_values(by=['date','group'])

# obtén los datos diarios acumulados agregados sobre los visitantes
visitorsAggregated = datesGroups.apply(lambda x: visitors[np.logical_and(visitors['date'] <= x['da
te'], visitors['group'] == x['group'])].agg({'date' : 'max', 'group' : 'max', 'visitors' : 'su
m'}), axis=1).sort_values(by=['date','group'])

# fusiona las dos tablas en una y da a sus columnas nombres descriptivos
cumlativeData = ordersAggregated.merge(visitorsAggregated, left_on=['date', 'group'], right_on=['d
ate', 'group'])
cumulativeData.columns = ['date', 'group', 'orders', 'buyers', 'revenue', 'visitors']
```

Creemos gráficos de ingresos acumulados por día y grupo de prueba A/B:

```
import matplotlib.pyplot as plt

# DataFrame con pedidos acumulados e ingresos acumulados por día, grupo A
cumulativeRevenueA = cumulativeData[cumulativeData['group']=='A'][['date','revenue', 'orders']]

# DataFrame con pedidos acumulados e ingresos acumulados por día, grupo B
cumulativeRevenueB = cumulativeData[cumulativeData['group']=='B'][['date','revenue', 'orders']]

# Trazar el gráfico de ingresos del grupo A
plt.plot(cumulativeRevenueA['date'], cumulativeRevenueA['revenue'], label='A')

# Trazar el gráfico de ingresos del grupo B
plt.plot(cumulativeRevenueB['date'], cumulativeRevenueB['revenue'], label='B')

plt.legend()
```

Ahora vamos a trazar el tamaño promedio de compra por grupo. Vamos a dividir los ingresos acumulados entre el número acumulado de pedidos:

```
plt.plot(cumulativeRevenueA['date'], cumulativeRevenueA['revenue']/cumulativeRevenueA['orders'], l
abel='A')
plt.plot(cumulativeRevenueB['date'], cumulativeRevenueB['revenue']/cumulativeRevenueB['orders'], l
abel='B')
plt.legend()
```

Vamos a trazar un gráfico de diferencia relativa para los tamaños promedio de compra. Agregaremos un eje horizontal con el **método axhline()** (es decir, una línea horizontal a lo largo del eje):

```
# reunir los datos en un DataFrame
mergedCumulativeRevenue = cumulativeRevenueA.merge(cumulativeRevenueB, left_on='date', right_on='d
```

```
ate', how='left', suffixes=['A', 'B'])

# trazar un gráfico de diferencia relativa para los tamaños de compra promedio
plt.plot(mergedCumulativeRevenue['date'], (mergedCumulativeRevenue['revenueB']/mergedCumulativeRev
enue['ordersB'])/(mergedCumulativeRevenue['revenueA']/mergedCumulativeRevenue['ordersA'])-1)

# agregar el eje X
plt.axhline(y=0, color='black', linestyle='--')
```

Analizar valores atípicos y aumentos: valores extremos

Un problema que puedes encontrar durante el análisis de la prueba A/B son los valores atípicos/anomalías, que pueden distorsionar los resultados de una prueba A/B. Una anomalía es un valor que rara vez aparece en una población estadística pero que puede causar errores cuando lo hace.

Los histogramas y diagramas de distribución son muy útiles cuando se trata de analizar anomalías.

Si dividimos un conjunto ordenado en 100 partes en lugar de cuatro, obtendremos **percentiles** (del latín per centum, "por cien"). Funcionan del mismo principio que los cuartiles: la percentil número n marca el valor mayor que n por ciento de los valores en la muestra. La probabilidad de que un valor aleatorio sea menor que el percentil n es n por ciento.

Para calcular los percentiles, necesitarás el método **percentil()** de NumPy:

```
# values - el rango de valores
# percentiles - la matriz de percentiles a calcular

import numpy as np
np.percentile(values, percentiles)
```

Para detectar anomalías, deberás analizar los percentiles 95, 97,5 y 99.

Errores comunes en el análisis de pruebas A/B

Dividir el tráfico de prueba incorrectamente

Este problema es común. Por ejemplo, los usuarios se dividen incorrectamente en segmentos.

Por ejemplo, no puedes considerar correcta la división del tráfico si el grupo A utiliza la versión para móviles de un sitio mientras que el grupo B utiliza la versión de escritorio.

Tener grupos de diferentes tamaños también distorsiona los resultados.

Ignorar la significancia estadística

Las decisiones sobre las diferencias en los resultados de las pruebas a menudo se toman basadas exclusivamente en el cambio relativo.

"Un segmento es un 5% mejor que el otro": ¿es realmente mejor o se trata simplemente de una fluctuación estadística? A largo plazo, todas las decisiones equivocadas resultan en pérdidas o disminución de ingresos para el negocio.

El problema de vislumbrar los resultados

Trabaja para evitarlo y trata de proteger a los demás de esto: a menudo se les pide a los analistas que tomen decisiones basadas en resultados intermedios. ¡No te rindas!

La muestra es demasiado pequeña

Vienen y te piden que hagas un test A/B en una muestra de 10 o 20 usuarios. Podrías decir que sí pero los resultados que obtengas no serán fiables. El impacto de cada observación individual será demasiado fuerte por lo que no habrá ni significancia, ni precisión.

La prueba fue demasiado corta

Calculaste el tamaño de muestra requerido utilizando una calculadora. Después inicias la prueba, obtienes tu muestra y tomas una decisión. Como habrás entendido del capítulo, los resultados de las pruebas pueden variar mucho en condiciones de la vida real.

Toma decisiones solo cuando estés seguro de los resultados.

La prueba fue demasiado larga

Tampoco deberías optar por el otro extremo. A veces los resultados aún no se han estabilizado, fluctúan, pero continuar con el experimento no tendrá ningún impacto en tu decisión.

Por ejemplo, no se ha alcanzado la significación estadística y los resultados del grupo B están fluctuando pero está claro que son más bajos que los resultados del grupo A. Por lo tanto, no tiene sentido continuar con la prueba: el grupo B no podrá tomar la delantera y no habrá un aumento en los ingresos.

Fallos al analizar anomalías

Nunca te olvides de las anomalías y tenlas en cuenta al analizar los resultados.

Los datos de la vida real contienen muchas más anomalías que los datos de este capítulo. Con el tiempo y la experiencia, te convertirás en un maestro cazador de anomalías.

No corregir la significación estadística con varias comparaciones

Cuantos más grupos tengas en tu prueba, más a menudo obtendrás un resultado falso positivo para al menos una comparación.