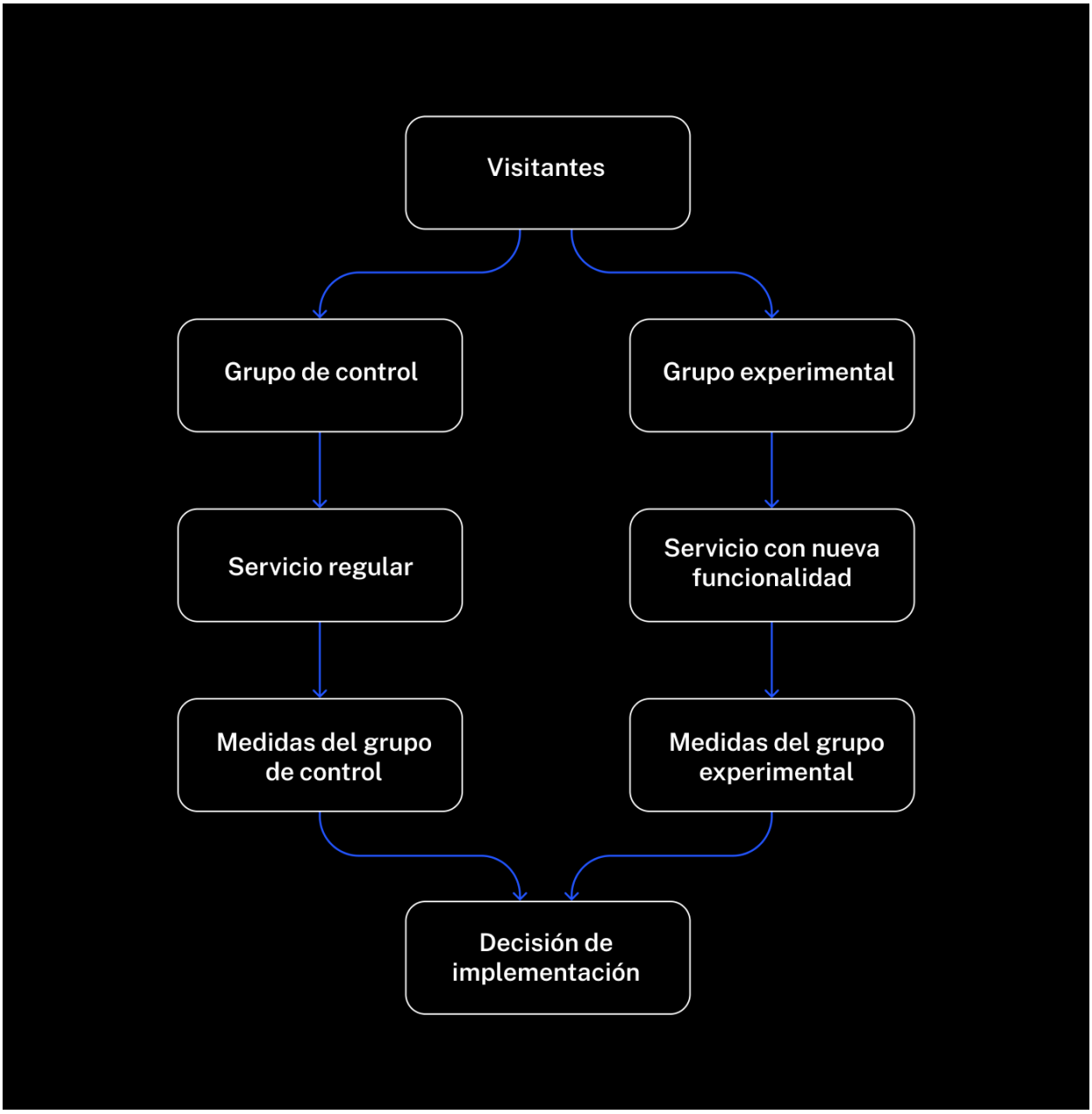
Resumen del capítulo: Implementación de nuevas funciones

Planificación de la implementación

Las **pruebas A/B** o **split testing** son una técnica de comprobación de hipótesis que ayuda a controlar el impacto que provocan los cambios de un servicio o producto sobre los usuarios. Se realiza de la siguiente manera: la población se divide en grupo de control (*A*) y grupo experimental (*B*). El grupo *A* utiliza el servicio habitual sin cambios. El grupo *B* utiliza la nueva versión, que es la que tenemos que probar.

El experimento dura un periodo determinado (por ejemplo, dos semanas). El objetivo es recopilar datos sobre el comportamiento de los visitantes en ambos grupos. Si la métrica clave en el grupo experimental mejora en comparación con el grupo de control, entonces se implementará la nueva funcionalidad.



Antes de las pruebas A/B, a menudo se utiliza la prueba A/A, o comprobación de validez, en la que los visitantes se dividen en dos grupos de control que se exponen a la misma versión del servicio. La métrica clave debe coincidir en ambos grupos, en caso contrario, hay que buscar un error.

Duración de las pruebas A/B

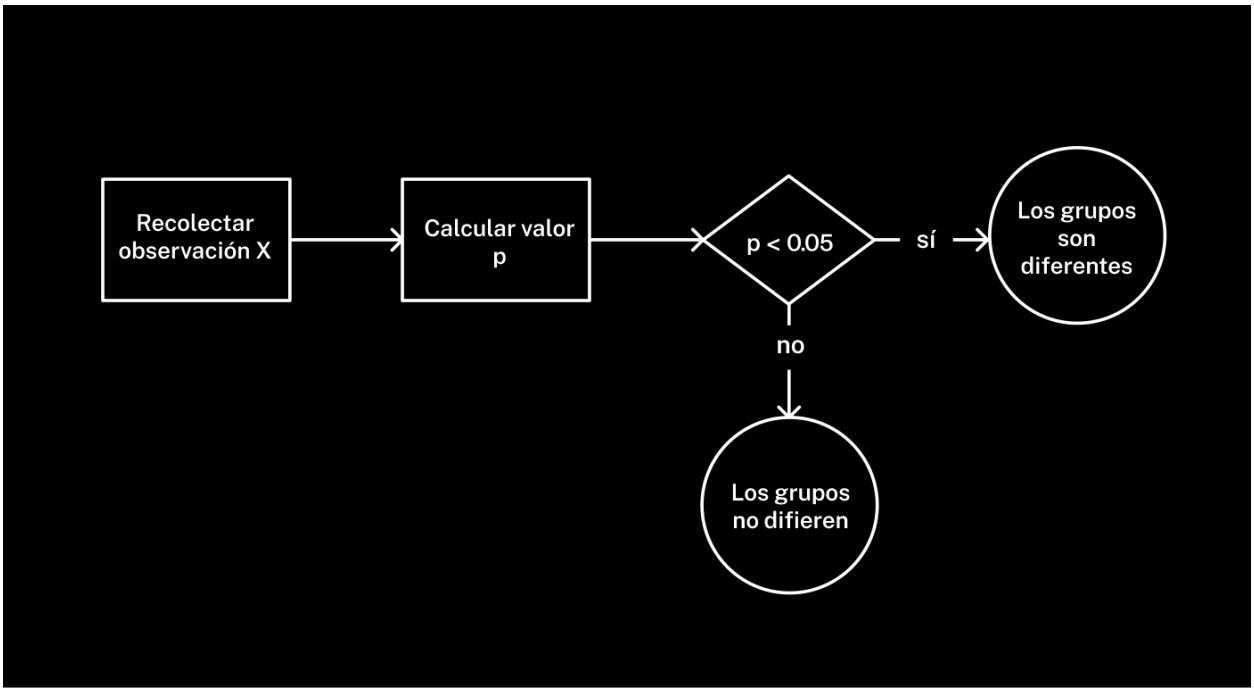
La implementación de una nueva funcionalidad cambia inevitablemente el comportamiento de los usuarios. Por lo general, los usuarios necesitan tiempo para acostumbrarse a los cambios. Una vez que se hayan acostumbrado por completo, podremos evaluar con certeza si el experimento ha sido un éxito. Cuantos más datos tengamos, menor será la probabilidad de error al comprobar las hipótesis estadísticas.

Las pruebas A/B sufren el llamado **peeking problem** (es decir, el problema de vislumbrar los resultados), que consiste en que el resultado final se distorsiona cuando se añaden nuevos datos al principio del experimento. Incluso un pequeño fragmento de datos nuevos es importante en relación con los datos ya acumulados y la significación estadística se alcanza en poco tiempo.

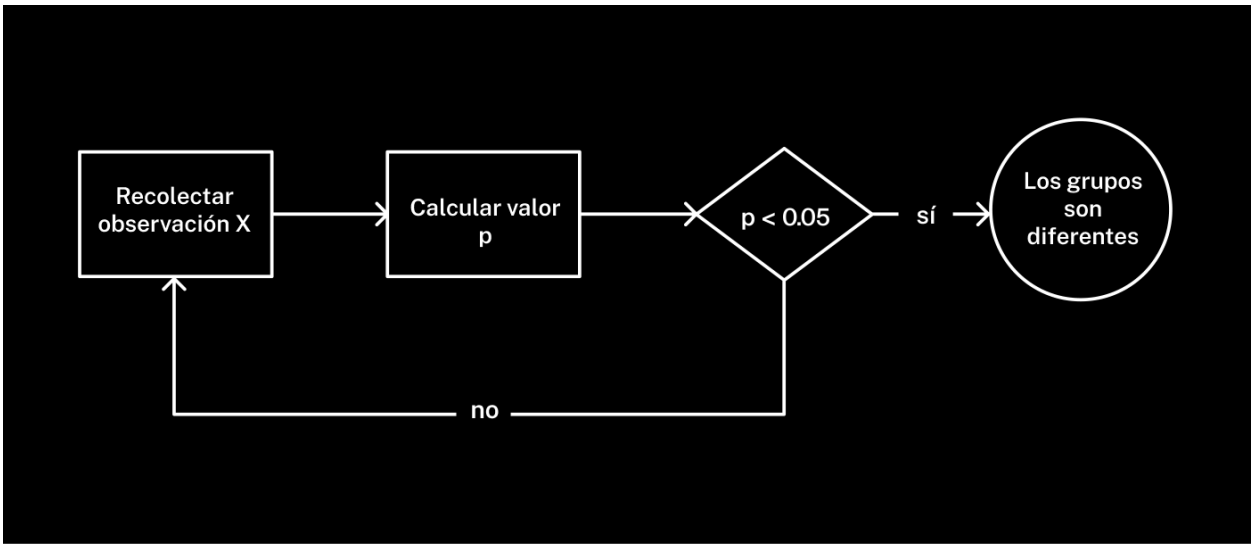
Esta es una de las manifestaciones de la ley de los grandes números. La dispersión tiende a ser mayor cuando el número de observaciones es escaso. En cambio, cuando tenemos un gran número de observaciones, el impacto de los valores atípicos se reduce. Entonces, si la muestra es demasiado pequeña, las diferencias son fáciles de ver. Para una prueba estadística, es una disminución del valor p hasta los valores lo suficientemente pequeños como para rechazar la hipótesis nula de que no hay diferencia.

Para solucionar el peeking problem, el tamaño de la muestra debe establecerse antes del inicio de la prueba.

Este es el procedimiento correcto de las pruebas *A/B*:



Y así es como no debes realizar una prueba *A/B*:



La forma más fácil de calcular el tamaño de la muestra es utilizar una calculadora en línea como esta: https://vwo.com/tools/ab-test-duration-calculator/

Comparación de las medias

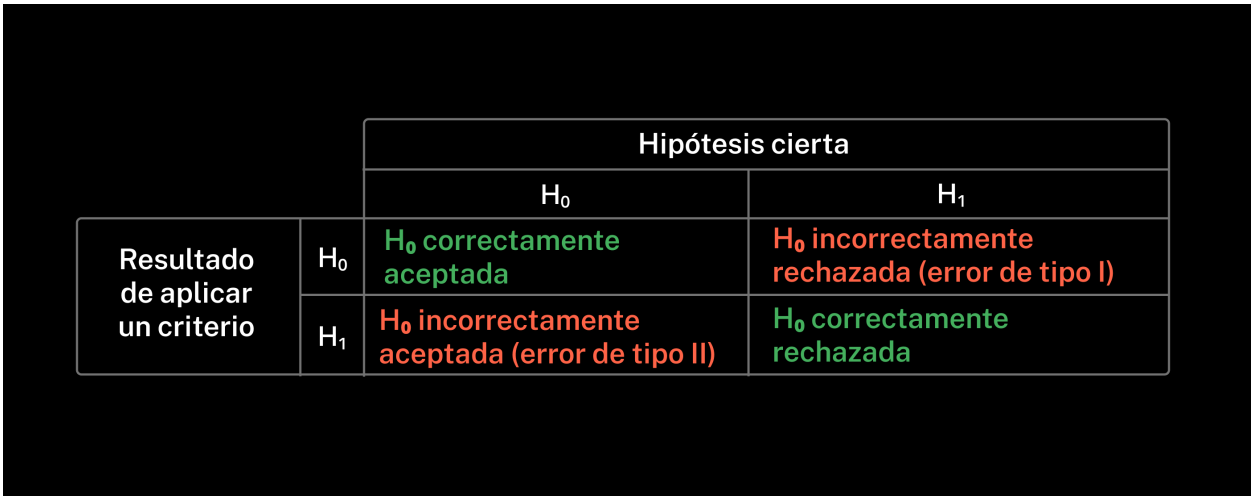
Analicemos los resultados de las pruebas *A/B*: el valor de la métrica describe el comportamiento de todos los usuarios.

Los resultados de las mediciones y los valores medios contienen un elemento aleatorio. Por lo tanto, tienen un componente de error aleatorio. No podemos predecir el valor exacto de cada observación con exactitud absoluta, pero podemos estimarlo utilizando métodos estadísticos.

Supongamos que nuestra hipótesis nula **H_0** dice: la nueva funcionalidad no mejora las métricas. Entonces nuestra hipótesis correspondiente **H_1** será: la nueva funcionalidad mejora las métricas.

En la fase de comprobación de la hipótesis, son posibles dos tipos de errores:

1. **Error de tipo I:** se produce cuando la hipótesis nula es correcta, pero se rechaza (resultado *falso positivo*. En este caso, la nueva funcionalidad se aprueba y, por lo tanto, es *positiva*)
2. **Error de tipo II:** se produce cuando la hipótesis nula es incorrecta, pero se acepta (resultado *falso negativo*)



Para aceptar o rechazar la hipótesis nula, calculemos el nivel de significación, también conocido como **valor p** (valor de probabilidad). Muestra la probabilidad del error de tipo I, pero no revela nada sobre el error de tipo II.

Ten en cuenta que si el valor p es mayor que el **valor de umbral**, la hipótesis nula no debería rechazarse. Si es menor que el umbral, puede que no valga la pena aceptar la hipótesis nula. Los umbrales generalmente aceptados son del 5 % y del 1 %. Pero solo el data scientist toma la decisión final sobre qué umbral podría considerarse suficiente.

Los valores medios se comparan utilizando los métodos de prueba de hipótesis unilateral. La hipótesis unilateral se acepta si el valor que se está comprobando es mucho mayor o mucho menor que el de la hipótesis nula. A nosotros nos interesa la desviación en una sola dirección, que es "mayor que".

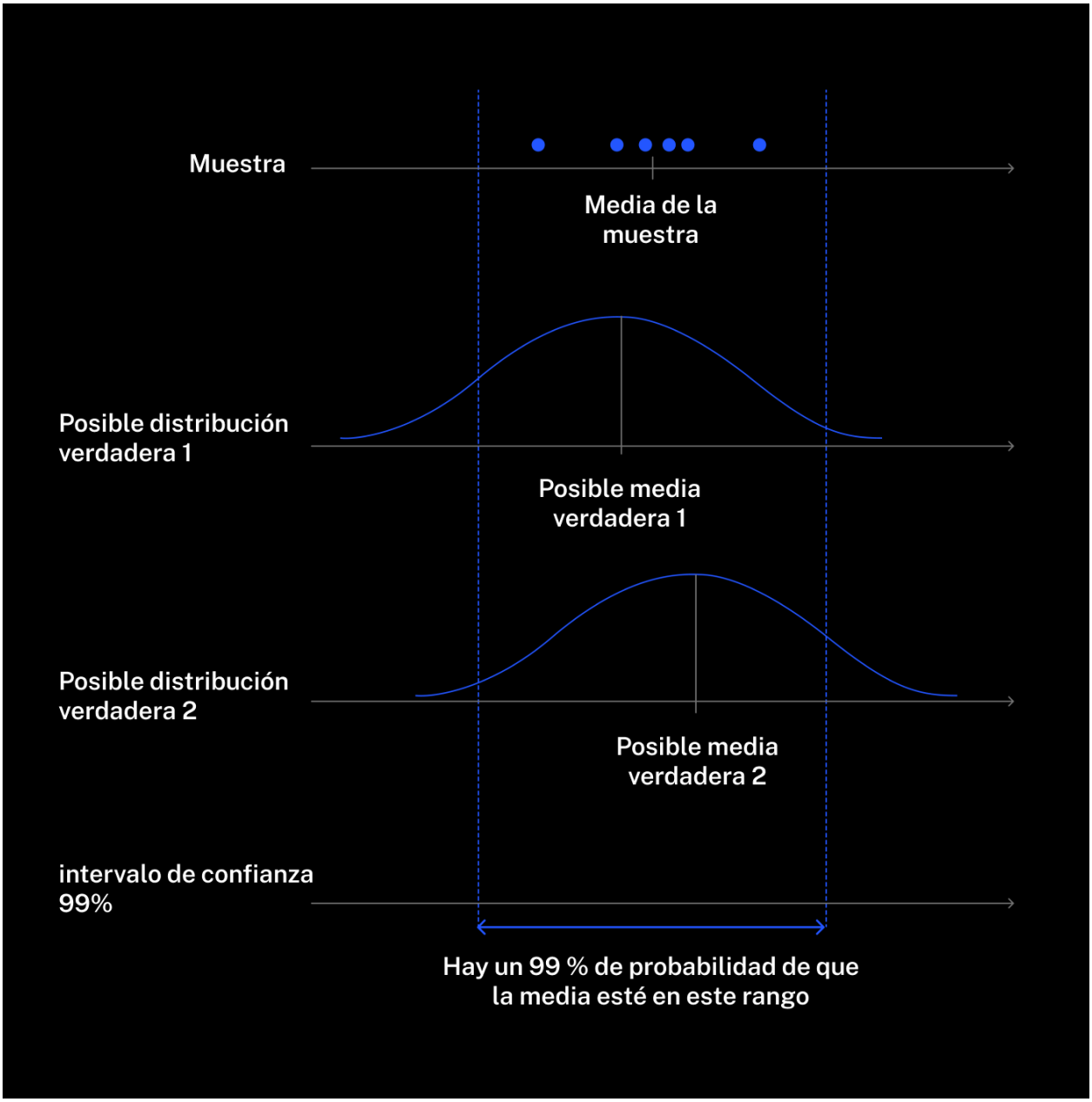
Si la distribución de los datos se aproxima a la normalidad (no hay valores atípicos significativos en los datos), se utiliza la prueba estándar para comparar las medias. Este método supone una distribución normal de las medias de todas las muestras y determina si la diferencia entre los valores comparados es lo suficientemente grande como para rechazar la hipótesis nula.

Intervalo de confianza

Un **intervalo de confianza** representa un segmento del eje numérico dentro del que cae el parámetro poblacional de interés, con una probabilidad predeterminada. El parámetro es desconocido, pero podemos estimarlo a partir de la muestra. Si el valor cae dentro del rango de 300 a 500 con una probabilidad del 99 %, entonces el intervalo de confianza del 99 % para este valor es **(300, 500)**.

Al calcular el intervalo de confianza, normalmente descartamos la misma cantidad de valores de cada uno de sus extremos.

El intervalo de confianza no es solo un rango de valores aleatorios. El valor que evaluamos no es aleatorio debido a su diseño. La causa de la variabilidad radica en el hecho de que el número es desconocido y que se calcula a partir de la muestra. El carácter aleatorio de la muestra introduce aleatoriedad en la estimación. El intervalo de confianza mide la confianza en dicha estimación.



Cálculo del intervalo de confianza

Podemos construir un intervalo de confianza para la media basado en la muestra utilizando el teorema del límite central.

Supongamos que tomamos nuestra muestra a partir de una distribución con los siguientes parámetros:

$$\mu = \text{media poblacional}$$

$$\sigma^2 = \text{varianza poblacional}$$

Denota la media de la muestra:

$$\bar{X} = \text{media de la muestra}$$

El teorema del límite central dice que todas las medias de todas las muestras posibles con un tamaño *n* se distribuyen normalmente alrededor de la verdadera media poblacional. "Alrededor" significa que la media de esta distribución de todas las medias muestrales será igual a la verdadera media poblacional. La varianza será igual a la varianza poblacional dividida entre *n* (el tamaño de la muestra).

$$\bar{X} \sim \mathbf{N}\left(\mu, \frac{\sigma^2}{n}\right)$$

La desviación estándar de esta distribución se denomina **error estándar** (*error estándar de la media*, o *SEM*, abreviado del "standard error of mean"):

$$\text{SEM}(\bar{X}) = \frac{\sigma}{\sqrt{n}}$$

Cuanto mayor sea el tamaño de la muestra, menor será el error estándar, es decir, todas las medias muestrales estarán más cerca de la media real. Cuanto mayor sea la muestra, más precisa será la estimación.

Vamos a estandarizar esta distribución normal:

$$\frac{\bar{X} - \mu}{\text{SEM}(\bar{X})} \sim \mathbf{N}(0, 1^2)$$

A partir de la distribución normal estándar, toma el percentil de 5 % *F*(0.05) y el percentil de 95 % *F*(0.95) para obtener el intervalo de confianza del 90 %:

$$P\left(F(0.05) < \frac{\bar{X} - \mu}{\text{SEM}(\bar{X})} < F(0.95)\right) = 90\%$$

Volvamos a escribirlo todo:

$$P\left(\bar{X} - F(0.05) \cdot \text{SEM}(\bar{X}) < \mu < \bar{X} + F(0.95) \cdot \text{SEM}(\bar{X})\right) = 90\%$$

¡Aquí lo tenemos! El intervalo de confianza del 90 % para la media real.

Solo nos queda un problema. Para calcular el error estándar, utilizamos la varianza poblacional, pero la desconocemos al igual que la media poblacional. La estimamos a partir de la muestra.

Este hecho también afecta a la distribución de las medias muestrales de modo que, si la varianza es desconocida, no podemos utilizar la distribución normal y tenemos que describirla con la distribución de Student. Al poner en la fórmula el percentil de 5 % *t*(0.05) y el percentil de 95 % *t*(0.95), obtenemos:

$$P\left(\bar{X} - t(0.05) \cdot \text{SEM}(\bar{X}) < \mu < \bar{X} + t(0.95) \cdot \text{SEM}(\bar{X})\right) = 90\%$$

Es posible simplificar el cálculo utilizando la distribución de Student *scipy.stats.t*. Tiene una función para el intervalo de confianza, *interval()*, que toma:

- *alpha*: nivel de significación
- *df*: número de grados de libertad (igual a n - 1)
- loc (de *localización*): la distribución media igual a la estimación media. Para la *muestra*, se calcula del modo siguiente: `sample.mean()`.
- *scale*: el error estándar de la distribución igual a la estimación del error estándar. Se calcula de la siguiente manera: `sample.sem()`.

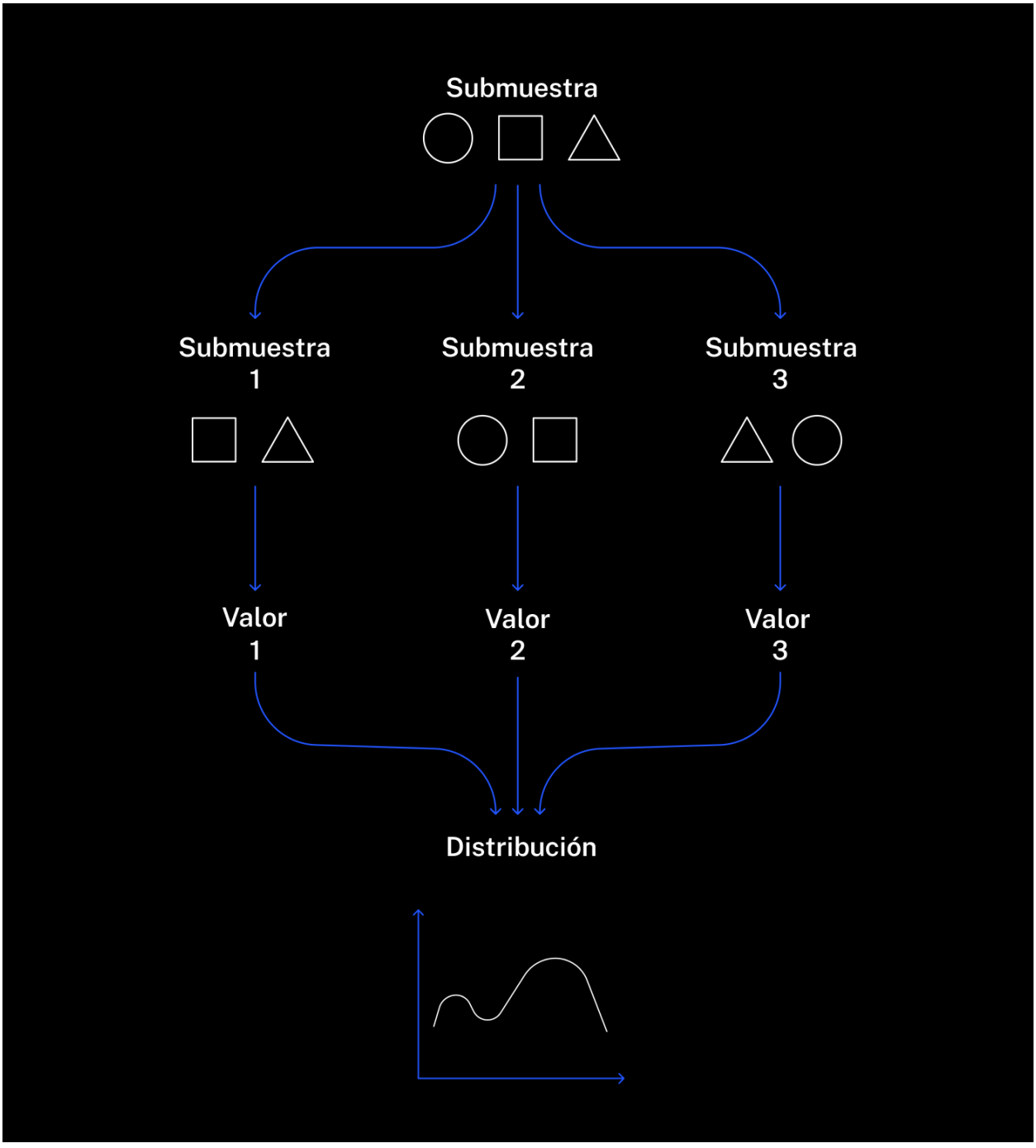
```
import pandas as pd
from scipy import stats as st

confidence_interval = st.t.interval(alpha, len(sample)-1,
                                    loc=sample.mean(), scale=sample.sem())
```

Bootstrap

Los valores complejos se pueden calcular con la ayuda de la técnica del **bootstrapping**.

Para conseguir un valor deseado, por ejemplo, la media, podemos obtener las submuestras (pseudomuestras) del conjunto fuente de datos. A continuación, calcularemos la media de cada una de ellas. En teoría, podemos formar submuestras y calcular el valor deseado a partir de ellas muchas veces. De este modo, podemos obtener varios valores para el parámetro de interés y estimar la distribución.



El bootstrapping es aplicable a cualquier muestra. Es útil cuando:

- Las observaciones no pueden ser descritas mediante una distribución normal;
- No hay pruebas estadísticas para el valor objetivo.

De hecho, no siempre se puede confiar en la distribución normal.

Bootstrap para el intervalo de confianza

Vamos a averiguar cómo formar submuestras para el bootstrapping. Ya conoces la función *sample()*. Para esta tarea necesitamos llamarla en un bucle. Pero aquí nos encontramos con un problema:

```
for i in range(5):
    # extrae un elemento aleatorio de la muestra 1
    # especifica random_state para su reproducción
    print(data.sample(1, random_state=54321))
```

Como especificamos el `random_state`, el elemento aleatorio es siempre el mismo. Para solucionarlo, crea una instancia `RandomState()` del módulo `numpy.random`:

```
from numpy.random import RandomState
state = RandomState(54321)
```

Esta instancia se puede pasar al argumento `random_state` de cualquier función. Es importante que, con cada nueva llamada, su estado cambie a aleatorio. Así obtendremos diferentes submuestras:

```
for i in range(5):
    # extrae un elemento aleatorio de la muestra 1
    print(data.sample(1, random_state=state))
```

Otro detalle importante a la hora de crear submuestras consiste en que deben proporcionar una selección de elementos con reemplazo. Es decir, el mismo elemento puede caer en una submuestra varias veces. Para ello, especifica `replace=True` para la función `sample()`. Compara:

```
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))
```

Bootstrap para análisis de prueba A/B

El bootstrapping también se utiliza para analizar los resultados de las pruebas *A/B*.

Mientras se realizaba la prueba, acumulamos datos sobre el parámetro objetivo en el grupo de control y en el grupo de tratamiento. Calculamos la *diferencia real de los parámetros objetivo* entre los grupos. Luego formulamos y probamos las hipótesis. La hipótesis nula es que no hay diferencia entre los parámetros objetivo de ambos grupos. La hipótesis alternativa es que, en el grupo experimental, el valor del parámetro objetivo es mayor. Encontremos el *valor p*.

Ahora vamos a investigar cuál es la probabilidad de que dicha diferencia se haya obtenido por casualidad (este será nuestro valor p). Concatena las muestras y usa bootstrap para obtener la distribución del monto promedio de compra.

Crea muchas submuestras y divide cada submuestra en dos con el índice *i*:

$$A_i \text{ — primera mitad del ejemplo}$$

$$B_i \text{ — segunda mitad del ejemplo}$$

Encuentra la diferencia del monto promedio de compra entre ellas:

Evaleemos en bootstrap la proporción de diferencias en el monto promedio de compra que resultaron ser no menos que las diferencias en el monto promedio de compra entre las muestras originales:

$$\text{p-value} = P(D_i \geq D)$$

```
import pandas as pd
import numpy as np

# diferencia real entre las medias de los grupos
AB_difference = samples_B.mean() - samples_A.mean()

alpha = 0.05

state = np.random.RandomState(54321)

bootstrap_samples = 1000
count = 0
for i in range(bootstrap_samples):
    # calcula cuántas veces excederá la diferencia entre las medias
    # el valor actual, siempre que la hipótesis nula sea cierta
    united_samples = pd.concat([samples_A, samples_B])
    subsample = united_samples.sample(frac=1, replace=True, random_state=state)

    subsample_A = subsample[:len(samples_A)]
    subsample_B = subsample[len(samples_A):]
    bootstrap_difference = subsample_B.mean() - subsample_A.mean()
```

```
    if bootstrap_difference >= AB_difference:
        count += 1

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")
```

Bootstrap para modelos

Bootstrap se puede utilizar para evaluar los intervalos de confianza en los modelos ML.