**INFERENCIAL**

**Población, muestra y sesgo**

**Población y muestra**

Empecemos por unos conceptos básicos que ya conocíamos:

Texto

Descripción generada automáticamente con confianza baja

Lo que suele ocurrir es que no tenemos acceso a toda la población, bien porque es muy grande o bien porque en realidad queremos considerar no sólo los elementos actuales de la población sino los futuros también para poder hacer predicciones sobre ellos (por ejemplo, si quiero estimar el resultado de los partidos de fútbol de la semana que viene o el impacto de un medicamento en los futuros pacientes).

En definitiva, a partir de ahora trabajaremos casi siempre con datos que consideraremos muestras de una población más amplia.

Para ilustrar estos conceptos y todos los de la unidad vamos a utilizar dos datasets:

**import** matplotlib.pyplot **as** plt

**import** pandas **as** pd

df\_titanic **=** pd.read\_csv("./data/titanic.csv")

df\_life\_satisfaction **=** pd.read\_csv("./data/lifesat\_full.csv")

df\_life\_satisfaction.head()

Tabla

Descripción generada automáticamente con confianza baja

len(df\_titanic), len(df\_life\_satisfaction)

(891, 36)

Realmente los dos son muestras de una población mayor, el dataset de Titanic tiene 891 registros de los dos mil y picos posibles (la población completa de todos los pasajeros que viajaron en él) y el dataset de nivel de satisfacción vital contiene datos de 36 países de los más de 190 países reconocidos en el mundo. En ese sentido ambos datasets son muestras de poblaciones mayores.

**Lo importante de una muesta es que sea "REPRESENTATIVA" de la población de la que ha sido extraída**. Esto quiere decir que lo que podamos estudiar y deducir de los datos de la muestra podamos aplicarlo al resto de la población. De esa manera podremos hacer estimaciones, deducciones e inferencias sobre la población en general sin necesidad de tener todos los datos o estudiar la población completa.

**Muestra aleatoria**

En general, cuando queremos estudiar una población de la que no podemos obtener todos los datos (por los motivos que sean) acudimos a obtener una muestra de la misma. La forma de obtener la muestra impacta en la representatividad de la misma y por lo tanto en la validez de las inferencias que hagamos de lo analizado sobre dicha muestra.

Para mantener en la muestra las propiedades de la población, es decir su representatividad, intentaremos obtener una muestra aleatoria (más adelante intentaremos que además sea estratificada, pero por ahora sólo que te suene). Es decir, escogiendo al azar los elementos que van a formar parte de nuestra muestra. Al proceso de obtención de una o varias muestras aleatorias se le denomina **muestreo aleatorio**.

Diagrama

Descripción generada automáticamente

Como muestra la figura es como hacer un sorteo y escoger a los ganadores para la muestra. Vamos a verlo con nuestros datasets, suponiendo que estos sean TODA la población y no muestras. Obtengamos una muestra de cada uno de ellos de forma aleatoria aprovechando el método sample de pandas:

num\_muestras\_titanic **=** int(len(df\_titanic) **\*** 0.2)

muestra\_titanic **=** df\_titanic.sample(num\_muestras\_titanic, random\_state **=** 42)

print(f"Poblacion titanic: {len(df\_titanic)} pasajeros")

print(f"Muestra titanic: {len(muestra\_titanic)} pasajeros")

num\_muestras\_paises **=** int(len(df\_life\_satisfaction) **\*** 0.2)

muestra\_paises **=** df\_life\_satisfaction.sample(num\_muestras\_paises, random\_state **=** 42)

print(f"Poblacion Indice Satifaccion: {len(df\_life\_satisfaction)} países")

print(f"Muestra Indice Satisfaccion: {len(muestra\_paises)} países")

Poblacion titanic: 891 pasajeros

Muestra titanic: 178 pasajeros

Poblacion Indice Satifaccion: 36 países

Muestra Indice Satisfaccion: 7 países

En este ejemplo hemos obtenido dos muestras aleatorias (escogidas al azar) del 20% del tamaño de la población. ¿Y el tamaño importa? Sí, pero no asegura la representatividad aunque la mejore.

Viendo los números de nuestro muestreo, ¿serán representativas nuestras muestras? Para poder contestar ahondemos en eso de la representatividad y en dos conceptos "nuevos": parámetros y estadísticos.

**Representatividad, parámetros y estadísticos**

Nuestras muestras aleatorias serían representativas si sus características son muy muy parecidas a las de la población a la que "representan". Es decir, si las medidas que obtengamos sobre las muestras son parecidas a las que obtendríamos en la población.

¿Y qué medidas? Pues las que vimos al tratar la estadística descriptiva y el análisis de datos en el módulo anterior. Sí, las medidas univariantes (frecuencias, medias, desviaciones), bivariantes, etc de una muestra deberían parecerse a las de la población para que podamos decir que esta es representativa.

Por ejemplo, comparemos el ratio de supervivencia en la muestra aleatoria y en la "población" total del dataset del titanic:

fig,axs **=** plt.subplots(1,2,figsize **=** (10,8))

freq **=** df\_titanic["alive"].value\_counts(normalize **=** **True**)

axs[0].bar(x **=** freq.index, height **=** freq.values, label **=** "Poblacion")

axs[0].set\_title("superviviencia poblacion")

axs[0].set\_ylim(0,0.7)

freq\_m **=** muestra\_titanic["alive"].value\_counts(normalize **=** **True**)

axs[1].bar(x **=** freq\_m.index, height **=** freq\_m.values, label **=** "Muestra")

axs[1].set\_title("supervivencia muestra")

axs[1].set\_ylim(0,0.7)

(0.0, 0.7)

Gráfico, Gráfico de barras

Descripción generada automáticamente

Se parecen pero hmmm, no del todo. Prueba tú a aumentar o disminuir el número de pasajeros en la muestra a ver qué ocurre.

Ahora comprobemos las medidas de tendencia central, posición y dispersión de la edad y el precio del ticket, por comparar variables numéricas:

df\_titanic\_medidas **=** df\_titanic[["age","fare"]].describe()

df\_muestra\_medidas **=** muestra\_titanic[["age","fare"]].describe()

df\_titanic\_medidas.merge(df\_muestra\_medidas, left\_index **=** **True**, right\_index**=** **True**, suffixes**=** ["\_pob","\_muestra"])

Tabla

Descripción generada automáticamente

Aquí hay más parecido "visual", pero todavía no podemos asegurarnos... Y eso que aquí podemos comparar muestra y población... Pero, lo normal no será tener la población (en cuyo caso no necesitamos la muestra), para esas situaciones para saber si las medidas son representativas acudiremos a los test de hipótesis. Y lo veremos en las siguientes unidades. Antes apliquemos esto mismo al dataset de satisfacción vital y hablemos del sesgo.

df\_lifesat\_medidas **=** df\_life\_satisfaction.describe()

df\_muestra\_medidas **=** muestra\_paises.describe()

df\_lifesat\_medidas.merge(df\_muestra\_medidas, left\_index**=** **True**, right\_index**=** **True**, suffixes **=** ["\_pob","\_muestra"]).loc["mean"].T

GDP per capita (USD)\_pob 42620.982007

Life satisfaction\_pob 6.599421

Life expectancy\_pob 79.455556

Homicide rate\_pob 3.741667

Employment rate\_pob 68.250000

GDP per capita (USD)\_muestra 56824.405352

Life satisfaction\_muestra 7.072510

Life expectancy\_muestra 81.600000

Homicide rate\_muestra 1.557143

Employment rate\_muestra 69.857143

Name: mean, dtype: float64

Este es incluso más difícil, porque hay medias que se parecen y otras no mucho... Prueba de nuevo a aumentar los valores de muestras.

Cuando las medidas hacen referencia a la población (nuetros "\_pob" en los ejemplos anteriores) se les llama **parámetros**. Cuando son las medidas hechas sobre las muestras, se les denomina **estadísticos**. Además para distinguirlos se les designa por letras diferentes:

Tabla

Descripción generada automáticamente

NOTA: La mediana de una muestra se representa por �~ y la proporción en la población por �, en cuyo caso, la proporción en la muestra se representa por �^

**Sesgos**

Se trata de una **distorsión no aleatoria que introducimos en la muestra como consecuencia de la selección de dicha muestra**. Si existe sesgo y no se controla, entonces, en general, haremos inferencias y deducciones erróneas a partir de la muestra.

Por ejemplo, si realizamos una encuesta sobre cómo va el país, no será lo mismo si todos los encuestados son de una clase social determinada, o de un partido político concreto.

Para verlo, supongamos que escogemos nuestra muestra del titanic pero sin darnos cuenta lo hacemos únicamente sobre pasajeros de Cherbourgh, por ejemplo porque sólo teníamos datos de esa ciudad:

muestra\_sesgada **=** df\_titanic.loc[df\_titanic.embark\_town **==** "Cherbourg"]

Si consideramos los estadísticos de nuestra como parámetros, veamos que error habríamos cometido, por ejemplo con el ratio de supervivencia:

fig,axs **=** plt.subplots(1,2,figsize **=** (10,8))

freq **=** df\_titanic["alive"].value\_counts(normalize **=** **True**)

axs[0].bar(x **=** freq.index, height **=** freq.values, label **=** "Poblacion")

axs[0].set\_title("superviviencia poblacion")

axs[0].set\_ylim(0,0.7)

freq\_m **=** muestra\_sesgada["alive"].value\_counts(normalize **=** **True**)

axs[1].bar(x **=** freq\_m.index, height **=** freq\_m.values, label **=** "Muestra")

axs[1].set\_title("supervivencia muestra")

axs[1].set\_ylim(0,0.7)

(0.0, 0.7)

Gráfico, Gráfico de barras

Descripción generada automáticamente

Como ya sabíamos hay una gran diferencia, tanto como para convertir al Titanic de una catastrofe a un accidente grave pero no tanto...

Por eso a la hora de seleccionar nuestra muestra debemos controlar mucho los posibles sesgos. Las implicaciones pueden ser mucho más impactantes incluso...

**Otro ejemplo de sesgo: El sesgo de supervivencia**

**El sesgo de supervivencia** (nada que ver con la supervivencia del Titanic) es un sesgo muy difícil de controlar ya que es el sesgo que se produce por tratar con los datos que tenemos y no con todos los posibles, porque son los datos que nos han llegado (que han sobrevivido). Dos ejemplos característicos son el estudio de los castillos y el de los aviones británicos en la segunda guerra mundial:

Imagen que contiene foto, diferente, computadora, tablero

Descripción generada automáticamente

**Estadística Inferencial**

Para terminar recordar qué es lo que buscamos con la estadística inferencial frente a la estadística descriptiva:

Word

Descripción generada automáticamente con confianza baja

**Media, intervalo de confianza y error estándar**

Antes de empezar, carguemos nuestras librerías y los datasets que nos sirven de ejemplo:

**import** matplotlib.pyplot **as** plt

**import** pandas **as** pd

df\_titanic **=** pd.read\_csv("./data/titanic.csv")

df\_life\_satisfaction **=** pd.read\_csv("./data/lifesat\_full.csv")

La media de las variables de una muestra, en general, no coincidirá con la media de la población [aunque no es imposible, por ejemplo si consideramos el conjunto de coches con cuatro ruedas, la media de ruedas es constante e igual para cualquier muestra, pero salvo casos tan "simples" como este ejemplo en general no será el mismo valor]. Eso no quiere decir que no sea representativa o no nos sirva para estimar la media poblacional. Por ejemplo en el caso de la edad del Titanic y una muestra aleatoria:

num\_muestras\_titanic **=** int(len(df\_titanic) **\*** 0.4)

muestra\_titanic **=** df\_titanic.sample(num\_muestras\_titanic, random\_state **=** 42)

print(f"Media de edad de la muestra: {muestra\_titanic['age'].mean()} años")

print(f"Muestra poblacion: {df\_titanic['age'].mean()} años")

print(rf"Error muestral: {round(abs(muestra\_titanic['age'].mean() **-** df\_titanic['age'].mean()),2)}")

Media de edad de la muestra: 30.786685189215653 años

Muestra poblacion: 30.319458263017587 años

Error muestral: 0.47

Si consideramos que ese "error muestral" es asumible, nuestra muestra será, en términos de edad, representativa de la media de edad poblacional.

Pero claro, no vamos a tener la media poblacional para calcular el error, y en caso de tenerla para que andarnos con muestras... Entonces, ¿qué hacemos?¿qué podemos decir de la edad media del pasaje del Titanic a partir de la edad media de nuestra muestra?

Aquí es donde entra **el intervalo de confianza.**

**Intervalo de Confianza**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Es decir, ese margen asumible de error es el intervalo de confianza, que depende de la probabilidad que queramos tener de que la media poblacional esté en el intervalo indicado. En nuestro caso anterior un intervalo de confianza de un año, supondría que estamos seguros con una cierta probabilidad de que la media poblacional estará en el intervalo 29,31.

De hecho como sabemos la media poblacional estamos al 100% seguros, pero ese no es el caso general. ¿Pero cómo se obtiene el IC y para qué probabilidades, o siempre es para el 95% como en la diapositiva?

Tranquilidad, tenemos una fórmula de cálculo que dependerá de esa probabilidad que nos deje tranquilos. Para eso tenemos que introducir el concepto de **error estándar** y además un poquito de "fe".

**Error Estándar**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Recordemos que la desviación estándar es una medida de dispersión de las variables continuas que se obtiene con la siguiente fórmula (cuando es un estadístico de una muestra):

Esquemático

Descripción generada automáticamente con confianza media

Obtengamos el IC (intervalo de confianza) de nuestra muestra del titanic para una probabilidad del 95%, con matices que veremos luego:

**from** math **import** sqrt

s **=** muestra\_titanic["age"].std()

SE **=** s**/**sqrt(len(muestra\_titanic))

media\_edad **=** round(muestra\_titanic.age.mean(),2)

IC **=** round(1.96 **\*** SE,2)

print("IC = ", IC)

print(f"Intervalo de confianza: [{media\_edad **-**IC},{media\_edad **+**IC}]")

IC = 1.36

Intervalo de confianza: [29.43,32.15]

Es decir, con una confianza del 95% podríamos decir que la media de edad poblacional del titanic está entre esos dos valores (que sabemos que es así, porque tenemos la población)

**El matiz**: Para poder usar este estadístico hemos supuesto que la distribución de la variable que queremos caracterizar sigue una figura específica de la que hemos hablado, la distribución normal o gaussiana.

Hasta aquí hemos visto como obtener el intervalo en el cuál podríamos situar la media poblacional a partir de la media de la muestra, pero la pregunta era si podíamos tener cierta confianza estadística de si la media de la muestra representa la media poblacional.

En general, la media muestral estadísticamente hablando tiene probabilidad 0 de ser la media poblacional, siempre hablaremos de un intervalo de confianza, pero lo que podemos hacer es ajustar ese IC, en función de la probabilidad de estar seguros. En las siguientes sesiones veremos el mecanismo para cambiar el IC en función de la confianza estadística que queramos tener. Y el mismo mecanismo, nos permitirá también analizar si a partir de la media muestral podemos saber si la media poblacional estará por encima o por debajo de un umbral. Para ello utilizaremos contraste de hipótesis y test como los empleados durante el módulo de análisis de datos.

**Contraste de hipótesis para la media (I)**

Antes de empezar, carguemos nuestras librerías y los datasets que nos sirven de ejemplo:

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**import** pandas **as** pd

df\_titanic **=** pd.read\_csv("./data/titanic.csv")

df\_life\_satisfaction **=** pd.read\_csv("./data/lifesat\_full.csv")

*# Obtengamos ya las muestras suponiendo los datos anteriores como las "poblaciones"*

**Contraste de hipótesis**

Dentro de la inferencia estadística, un contraste de hipótesis (también denominado test de hipótesis o prueba de significación) es un procedimiento para juzgar si una propiedad que se supone en una población estadística es compatible con lo observado en una muestra de dicha población.

La mayoría de las veces, vamos a realizar **T-test** o **Z-test**

Diagrama

Descripción generada automáticamente

En general no sabremos cual es la desviación estándar de la variable que estamos estudiando para la población así que lo que haremos será aplicar el test T o T-test o T de Student.

**T-Test: Consideraciones**

En una **prueba t de Student**, o **Test-T**, el estadístico utilizado tiene una distribución t de Student si la hipótesis nula es cierta. *Se aplica cuando la población estudiada sigue una distribución normal, pero el tamaño muestral es pequeño* (ojo, porque nosotros nos vamos a saltar esta comprobación, hay otras técnicas como el boostraping, para aplicar cuando no se cumplen estas condiciones, pero se escapan del ámbito abarcable por el curso)

**T-Test para comprobar la igualdad a un valor**

Si lo que queremos es testar la hipótesis de que no hay diferencia estadística significativa entre la media poblacional y un valor concreto, emplearemos el **T-test bilateral**. Esto lo vamos a emplear para "estrechar" o "ensanchar" el IC:

En este caso, la hipótesis nula (la que rechazaremos si el p-value nos sale por debajo de nuestro 0.05 o el grado de confianza que queramos poner) es:



**Es decir media poblacional es igual a un hipotético valor a**

Y la hipótesis alternativa, la que ganará confianza si el p-value del test nos sale por debajo de nuestro umbral de confianza (hasta ahora siempre hemos usado 0.05), sería:



Si el p-value nos sale por debajo del umbral de confianza, rechazaremos la hipótesis nula y diremos que estamos seguros (al nivel de confianza escogido) de que la media poblacional no es ese valor a (es decir aceptamos la hipótesis alternativa)

Vamos a hacer el test para nuestras muestras respecto a la "edad" en el caso de Titanic y al nivel de satisfacción vital para ese otro dataset, pero poniendo como valor de � las medias muestrales, a ver qué ocurre:

num\_muestras\_titanic **=** int(len(df\_titanic) **\*** 0.4)

muestra\_titanic **=** df\_titanic.sample(num\_muestras\_titanic, random\_state **=** 42)

print(f"Poblacion titanic: {len(df\_titanic)} pasajeros")

print(f"Muestra titanic: {len(muestra\_titanic)} pasajeros")

num\_muestras\_paises **=** int(len(df\_life\_satisfaction) **\*** 0.4)

muestra\_paises **=** df\_life\_satisfaction.sample(num\_muestras\_paises, random\_state **=** 42)

print(f"Poblacion Indice Satifaccion: {len(df\_life\_satisfaction)} países")

print(f"Muestra titanic: {len(muestra\_paises)} países")

Poblacion titanic: 891 pasajeros

Muestra titanic: 356 pasajeros

Poblacion Indice Satifaccion: 36 países

Muestra titanic: 14 países

Utilizamos nuestro viejo amigo scipy:

**import** scipy.stats **as** stats

a **=** muestra\_titanic.age.mean()

stats.ttest\_1samp(muestra\_titanic.age.values,popmean**=** a)

TtestResult(statistic=0.0, pvalue=1.0, df=355)

a **=** muestra\_paises["Life satisfaction"].mean()

stats.ttest\_1samp(muestra\_paises["Life satisfaction"].values, popmean **=** a)

TtestResult(statistic=0.0, pvalue=1.0, df=13)

Como puedes ver en ambos casos sale un p-value = 1, es decir no podemos rechazar que la media muestral no sea la media poblacional, pero eso ya lo suponíamos o de ahí partíamos, que una muestra siempre puede ser la media de la Población si no...

El T-test nos sirve, entre otras cosas para obtener un IC más ancho o estrecho que el otorgado por la fórmula de la sesión anterior.

Para ello, interpretamos a como si fuera el resultado de aplicar el intervalo de confianza IC:



Fijado un p-value podemos ir forzando los IC hasta encontrar el que da justo un p-value en el test ligeramente superior.

Para empezar, comprobemos que para la "muestra" del titanic el valor de IC que nos da un p-value límite de 0.05 es precisamente el calculado con la fórmula de la sesión anterior (IC=1.96\*SE), es decir 1.36 años:

*# Vamos a ir probando ICs desde 0 años,cada vez más grandes hasta llegar a dar un p-value menor que 0.05 el anterior lo consideraremos nuestro IC máximo de garantía*

*# para el caso de p-value = 0.05*

p\_value\_target **=** 0.05

previous\_IC **=** 0

**for** IC **in** np.arange(0,1.5,0.05): *# partimos de IC de 0 años (a = media muestral) y vamos subiendo hasta 1.5, hasta encontrar el IC máximo (que debería ser 1.35), que nos garantice al 95% que la media puede ser ese valor*

a **=** muestra\_titanic.age.mean() **+** IC *# La curva es simétrica nos vale con testar un lado solo*

p\_value **=** stats.ttest\_1samp(a**=**muestra\_titanic.age.values,popmean**=** a).pvalue

print(f"Para un IC: {IC}, comprobamos a = {a} y nos da un p-value = {p\_value}")

**if** p\_value **<=** p\_value\_target:

print(f"para IC = {IC} se rechaza la hipótesis nula de igualda, es decir a = {a}",\

f"ya no es una posible media poblacional al {(1**-**p\_value\_target)**\***100}% de signifcacion")

print(f"Nuestro IC escogido es entonces, IC = {previous\_IC}")

**break**

previous\_IC **=** IC

Para un IC: 0.0, comprobamos a = 30.786685189215653 y nos da un p-value = 1.0

Para un IC: 0.05, comprobamos a = 30.836685189215654 y nos da un p-value = 0.9426585918232326

Para un IC: 0.1, comprobamos a = 30.886685189215655 y nos da un p-value = 0.8856142034354949

Para un IC: 0.15000000000000002, comprobamos a = 30.936685189215652 y nos da un p-value = 0.829159221404719

Para un IC: 0.2, comprobamos a = 30.986685189215653 y nos da un p-value = 0.7735768867669685

Para un IC: 0.25, comprobamos a = 31.036685189215653 y nos da un p-value = 0.7191370201025511

Para un IC: 0.30000000000000004, comprobamos a = 31.086685189215654 y nos da un p-value = 0.6660920918175457

Para un IC: 0.35000000000000003, comprobamos a = 31.136685189215655 y nos da un p-value = 0.6146737329749277

Para un IC: 0.4, comprobamos a = 31.186685189215652 y nos da un p-value = 0.5650897663098774

Para un IC: 0.45, comprobamos a = 31.236685189215653 y nos da un p-value = 0.5175218188417169

Para un IC: 0.5, comprobamos a = 31.286685189215653 y nos da un p-value = 0.4721235575827958

Para un IC: 0.55, comprobamos a = 31.336685189215654 y nos da un p-value = 0.42901956911951244

Para un IC: 0.6000000000000001, comprobamos a = 31.386685189215655 y nos da un p-value = 0.38830488318756884

Para un IC: 0.65, comprobamos a = 31.436685189215652 y nos da un p-value = 0.3500451206280947

Para un IC: 0.7000000000000001, comprobamos a = 31.486685189215653 y nos da un p-value = 0.31427722805914643

Para un IC: 0.75, comprobamos a = 31.536685189215653 y nos da un p-value = 0.28101074587754266

Para un IC: 0.8, comprobamos a = 31.586685189215654 y nos da un p-value = 0.25022954332502667

Para un IC: 0.8500000000000001, comprobamos a = 31.636685189215655 y nos da un p-value = 0.22189394465598777

Para un IC: 0.9, comprobamos a = 31.686685189215652 y nos da un p-value = 0.19594316410831686

Para un IC: 0.9500000000000001, comprobamos a = 31.736685189215653 y nos da un p-value = 0.17229796441614167

Para un IC: 1.0, comprobamos a = 31.786685189215653 y nos da un p-value = 0.15086345387051053

Para un IC: 1.05, comprobamos a = 31.836685189215654 y nos da un p-value = 0.1315319401540534

Para un IC: 1.1, comprobamos a = 31.886685189215655 y nos da un p-value = 0.11418576496137979

Para un IC: 1.1500000000000001, comprobamos a = 31.936685189215652 y nos da un p-value = 0.09870005130063607

Para un IC: 1.2000000000000002, comprobamos a = 31.986685189215653 y nos da un p-value = 0.08494530483616042

Para un IC: 1.25, comprobamos a = 32.03668518921565 y nos da un p-value = 0.0727898211412502

Para un IC: 1.3, comprobamos a = 32.08668518921565 y nos da un p-value = 0.062101861757017346

Para un IC: 1.35, comprobamos a = 32.136685189215655 y nos da un p-value = 0.05275157300630952

Para un IC: 1.4000000000000001, comprobamos a = 32.18668518921565 y nos da un p-value = 0.04461263215372979

para IC = 1.4000000000000001 se rechaza la hipótesis nula de igualda, es decir a = 32.18668518921565 ya no es una posible media poblacional al 95.0% de signifcacion

Nuestro IC escogido es entonces, IC = 1.35

Pero podemos cambiar el target y ver cómo reducir el IC a expensas de la confianza estadística, por ejemplo p-value = 0.1 (el 90% de significación)

*# Vamos a ir probando ICs cada vez más grandes hasta llegar a dar un p-value menor que 0.05 el anterior lo consideraremos nuestro IC máximo de garantía*

*# para el caso de p-value = 0.05*

p\_value\_target **=** 0.1

**import** numpy **as** np

previous\_IC **=** 0.5

**for** IC **in** np.arange(0.5,1.5,0.05): *# partimos de IC de 0.5 años y vamos subiendo hasta 1.5 hasta encontrar el IC máximo (que debería ser 1.35)*

a **=** muestra\_titanic.age.mean() **+** IC *# La curva es simétrica nos vale con testar un lado solo*

p\_value **=** stats.ttest\_1samp(a**=**muestra\_titanic.age.values,popmean**=** a).pvalue

**if** p\_value **<=** p\_value\_target:

print(f"para IC = {IC} se rechaza la hipótesis nula de igualda, es decir a = {a}",\

f"ya no es una posible media poblacional al {(1**-**p\_value\_target)**\***100}% de signifcacion")

print(f"Nuestro IC escogido es entonces, IC = {previous\_IC}")

**break**

previous\_IC **=** IC

para IC = 1.1500000000000006 se rechaza la hipótesis nula de igualda, es decir a = 31.936685189215655 ya no es una posible media poblacional al 90.0% de signifcacion

Nuestro IC escogido es entonces, IC = 1.1000000000000005

O podemos aumentar el IC para estar más seguros todavía de que la media poblacional caerá en ese intervalor, por ejemplo al 99%, p-target = 0.01

*# Vamos a ir probando ICs cada vez más grandes hasta llegar a dar un p-value menor que 0.05 el anterior lo consideraremos nuestro IC máximo de garantía*

*# para el caso de p-value = 0.05*

p\_value\_target **=** 0.01

**import** numpy **as** np

previous\_IC **=** 0.5

**for** IC **in** np.arange(0.5,2,0.05): *# partimos de IC de 0.5 años y vamos subiendo hasta 1.5 hasta encontrar el IC máximo (que debería ser 1.35)*

a **=** muestra\_titanic.age.mean() **+** IC *# La curva es simétrica nos vale con testar un lado solo*

p\_value **=** stats.ttest\_1samp(a**=**muestra\_titanic.age.values,popmean**=** a).pvalue

**if** p\_value **<=** p\_value\_target:

print(f"para IC = {IC} se rechaza la hipótesis nula de igualda, es decir a = {a}",\

f"ya no es una posible media poblacional al {(1**-**p\_value\_target)**\***100}% de signifcacion")

print(f"Nuestro IC escogido es entonces, IC = {previous\_IC}")

**break**

previous\_IC **=** IC

para IC = 1.8000000000000012 se rechaza la hipótesis nula de igualda, es decir a = 32.58668518921566 ya no es una posible media poblacional al 99.0% de signifcacion

Nuestro IC escogido es entonces, IC = 1.750000000000001

Ahora necesitaríamos un IC de 1.75 años.

Para terminar apliquémoslo a la satisfacción vital y obtener el IC para pvalue de 0.05 (el IC según la fórmula) y para pvalue del 0.1:

*# Nos hacemos una idea de donde puede estar el margen*

print(muestra\_paises["Life satisfaction"].mean())

print(df\_life\_satisfaction["Life satisfaction"].mean())

6.503470342300567

6.599421401893583

*# Para p-value = 0.05*

previous\_IC **=** 0.1

p\_target **=** 0.05

**for** IC **in** np.arange(0.1,1,0.1): *# Probamos de 0.1 a 1 puntos de IC en saltos de 0.1*

a **=** muestra\_paises["Life satisfaction"].mean() **+** IC

pvalue **=** stats.ttest\_1samp(a**=**muestra\_paises["Life satisfaction"].values,popmean**=** a).pvalue

**if** pvalue **<=** p\_target:

print("IC escogido:", round(previous\_IC,2))

**break**

previous\_IC **=** IC

IC escogido: 0.4

*# Para p-value = 0.1*

previous\_IC **=** 0.1

p\_target **=** 0.1

**for** IC **in** np.arange(0.1,1,0.1): *# Probamos de 0.1 a 1 puntos de IC en saltos de 0.1*

a **=** muestra\_paises["Life satisfaction"].mean() **+** IC

pvalue **=** stats.ttest\_1samp(a**=**muestra\_paises["Life satisfaction"].values,popmean**=** a).pvalue

**if** pvalue **<=** p\_target:

print("IC escogido:", round(previous\_IC,2))

**break**

previous\_IC **=** IC

IC escogido: 0.3

*# Para p-value = 0.01*

previous\_IC **=** 0.1

p\_target **=** 0.01

**for** IC **in** np.arange(0.1,1,0.1): *# Probamos de 0.1 a 1 puntos de IC en saltos de 0.1*

a **=** muestra\_paises["Life satisfaction"].mean() **+** IC

pvalue **=** stats.ttest\_1samp(a**=**muestra\_paises["Life satisfaction"].values,popmean**=** a).pvalue

**if** pvalue **<=** p\_target:

print("IC escogido:", round(previous\_IC,2))

**break**

previous\_IC **=** IC

IC escogido: 0.6

**Contraste de hipótesis para la media (II)**

Como hasta ahora antes de empezar, carguemos nuestras librerías y los datasets de ejemplo:

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**import** pandas **as** pd

df\_titanic **=** pd.read\_csv("./data/titanic.csv")

df\_life\_satisfaction **=** pd.read\_csv("./data/lifesat\_full.csv")

*# Obtengamos ya las muestras suponiendo los datos anteriores como las "poblaciones"*

num\_muestras\_titanic **=** int(len(df\_titanic) **\*** 0.4)

muestra\_titanic **=** df\_titanic.sample(num\_muestras\_titanic, random\_state **=** 42)

print(f"Poblacion titanic: {len(df\_titanic)} pasajeros")

print(f"Muestra titanic: {len(muestra\_titanic)} pasajeros")

num\_muestras\_paises **=** int(len(df\_life\_satisfaction) **\*** 0.4)

muestra\_paises **=** df\_life\_satisfaction.sample(num\_muestras\_paises, random\_state **=** 42)

print(f"Poblacion Indice Satifaccion: {len(df\_life\_satisfaction)} países")

print(f"Muestra titanic: {len(muestra\_paises)} países")

Poblacion titanic: 891 pasajeros

Muestra titanic: 356 pasajeros

Poblacion Indice Satifaccion: 36 países

Muestra titanic: 14 países

**T-test para umbrales**

En este caso la hipótesis nula puede tomar dos "aspectos":

Texto

Descripción generada automáticamente con confianza baja

Y las hipótesis alternativas equivalentes quedarían:

Texto

Descripción generada automáticamente con confianza media

¿Cómo se aplica? Comprobemos si la media de "Life satisfaction" de la poblacion de países es mayor que 6.6, por ejemplo (que sabemos que es menor) con una confianza estadística del 95%. En este caso nuestra hipótesis alternativa es Ha: mu>6.6 y la hipótesis nula H0:mu≤a.

**import** scipy.stats **as** stats

data **=** muestra\_paises["Life satisfaction"]

stats.ttest\_1samp(a**=**data,popmean**=**6.6,alternative**=**'greater')

TtestResult(statistic=-0.43985840098335277, pvalue=0.6663681557371673, df=13)

Es decir en este caso no podemos rechazar la hipótesis nula (p-value mayor que 0,05), y no podemos afirmar que la media de la satisfacción vital sea para la población mayor que 6.6 con una significación del 95%. Probemos ahora a ver si podemos aceptar que la satisfacción vital media es menor que 4.5, por ejemplo:

Word

Descripción generada automáticamente con confianza media

data **=** muestra\_paises["Life satisfaction"]

stats.ttest\_1samp(a**=**data,popmean**=**4.5,alternative**=**'less')

TtestResult(statistic=9.129248794457022, pvalue=0.9999997438782476, df=13)

Claramente tampoco podemos aceptar la hipótesis alternativa de que la media de la satisfacción vital de todos los países estás por debajo de 4.5, al menos no al 95%.

**T-test para dos muestras**

Podemos comparar las medias de variables de dos muestras de poblaciones diferentes.

Ejemplo: Queremos comprobar el desempeño de dos call centers en términos de longitud media de las llamadas y encontrar si hay diferencias estadísticamente significativas.

**H0** (hipótesis nula): no hay diferencia entre la media del tiempo de llamadas entre ambos call centers  
**Ha** (hipótesis alternativa): existe una diferencia entre la media del tiempo de llamadas de ambos call centers

Seleccionamos aleatoriamente 20 llamadas de cada call center y medimos la longitud media de cada llamada (en segundos):

Call Center A: 115, 138, 109, 141, 139, 131, 123, 128, 144, 167, 134, 142, 174, 149, 115, 152, 131, 161, 131, 121  
Call Center B: 137, 125, 86, 121, 101, 110, 129, 136, 102, 114, 131, 109, 119, 144, 145, 121, 137, 131, 134, 126

*# Call Center A*

A **=** [115, 138, 109, 141, 139, 131, 123, 128, 144, 167, 134, 142, 174, 149, 115, 152, 131, 161, 131, 121]

*# Call Center B*

B **=** [137, 125, 86, 121, 101, 110, 129, 136, 102, 114, 131, 109, 119, 144, 145, 121, 137, 131, 134, 126]

stats.ttest\_ind(a**=**A, b**=**B, alternative**=**'two-sided')

Ttest\_indResult(statistic=2.761988576903894, pvalue=0.008802045351068374)

Con este p-value podemos rechazar la hipótesis nula de que no existe diferencia entre ambas y aceptar que existe una diferencia en la duración media de las llamadas a un call center y otro.

**Frecuencias Muestrales**

Como hasta ahora antes de empezar, carguemos nuestras librerías y los datasets de ejemplo:

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**import** pandas **as** pd

df\_titanic **=** pd.read\_csv("./data/titanic.csv")

*# Obtengamos ya las muestras suponiendo los datos anteriores como las "poblaciones"*

num\_muestras\_titanic **=** int(len(df\_titanic) **\*** 0.4)

muestra\_titanic **=** df\_titanic.sample(num\_muestras\_titanic, random\_state **=** 42)

print(f"Poblacion titanic: {len(df\_titanic)} pasajeros")

print(f"Muestra titanic: {len(muestra\_titanic)} pasajeros")

Poblacion titanic: 891 pasajeros

Muestra titanic: 356 pasajeros

**Z-test para proporciones y frecuencias**

Es hora de trabajar sobre las medidas que tomamos en las variables categóricas, es decir sobre sus frecuencias y poder hacer suposiciones y testar hipótesis sobre las frecuencias de esas mismas variables para la población total de la que hemos obtenido nuestra muestra.

**Test para un valor concreto de frecuencia o proporción**

Usemos un ejemplo para ver el Z-test aplicado a una frecuencia. En concreto, supongamos que queremos saber si dada la tasa de supervivencia de nuestra muestra:

tasa\_supervivencia\_muestral **=** round(muestra\_titanic["alive"].value\_counts(normalize **=** **True**)["yes"] **\*** 100,2)

tasa\_supervivencia\_muestral

39.33

Podemos rechazar una tasa un poco menor, digamos un 37%, con una significación del 95% (probabilidad de no equivocarnos al rechazar la hipótesis nula). En términos de hipótesis:

 (La hipótesis nula es la contraria a lo que estamos queriendo reforzar)



Para aplicar el test recurrimos a otro paquete, esta vez statsmodels:

**from** statsmodels.stats.proportion **import** proportions\_ztest

p\_hat **=** tasa\_supervivencia\_muestral**/**100 *# Proporcion o frecuecia de la variable en la muestra*

n **=** len(muestra\_titanic) *# Tamaño de la muestra*

Pi **=** 0.37 *# Proporción o frecuencia que queremos probar en la población*

*# Calcular el estadístico Z y el p-valor*

z\_stat, p\_value **=** proportions\_ztest(p\_hat **\*** n, n, Pi)

*# Mostrar resultados*

print(f"Estadístico Z: {z\_stat}")

print(f"P-valor: {p\_value}")

Estadístico Z: 0.8999781162822953

P-valor: 0.3681318966771918

El p-valor obtenido no nos permite rechazar la hipótesis nula, por lo que la tasa de supervivencia en la población para una confianza del 95% podría ser 37. Probemos un valor aún menor, 32% (y sí así podríamos encontrar el equivalente al IC para esta tasa)

p\_hat **=** tasa\_supervivencia\_muestral**/**100 *# Proporcion o frecuecia de la variable en la muestra*

n **=** len(muestra\_titanic) *# Tamaño de la muestra*

Pi **=** 0.32 *# Proporción o frecuencia que queremos probar en la población*

*# Calcular el estadístico Z y el p-valor*

z\_stat, p\_value **=** proportions\_ztest(p\_hat **\*** n, n, Pi)

*# Mostrar resultados*

print(f"Estadístico Z: {z\_stat}")

print(f"P-valor: {p\_value}")

Estadístico Z: 2.8312616276176983

P-valor: 0.004636477880043044

Para ese p-value sí que podríamos rechazar la hipótesis nula y decir que la tasa de supervivencia en la "población" del titanic es superior a 32%, con una confianza del 95%

**Test para umbrales de una proporción**

Al igual que ocurría con el caso de testar si la media de una población sería mayor o menor que un umbral, podemos aplicar el test Z para estimar la significación estadística de hipótesis análogas sobre proporciones o frecuencias

Por ejemplo, supongamos que nuestra hipótesis es que la tasa de supervivencia es menor que 40%, cómo serían nuestras hipótesis nula y alternativa:

Word

Descripción generada automáticamenteY nuestro test:

p\_hat **=** tasa\_supervivencia\_muestral**/**100 *# Proporcion o frecuecia de la variable en la muestra*

n **=** len(muestra\_titanic) *# Tamaño de la muestra*

Pi **=** 0.40 *# Proporción o frecuencia que queremos probar en la población*

*# Calcular el estadístico Z y el p-valor*

z\_stat, p\_value **=** proportions\_ztest(p\_hat **\*** n, n, Pi, alternative **=** "smaller")

*# Mostrar resultados*

print(f"Estadístico Z: {z\_stat}")

print(f"P-valor: {p\_value}")

Estadístico Z: -0.2587919905189477

P-valor: 0.3978978690261754

No podemos rechazar la hipótesis nula y por tanto no podemos asumir que la tasa de supervivencia en la población sea menor al 40%, con un 95% de confianza estadística.

Probemos ahora para la hipótesis de que la tasa debe ser mayor que un 30%:

Word

Descripción generada automáticamente con confianza baja

p\_hat **=** tasa\_supervivencia\_muestral**/**100 *# Proporcion o frecuecia de la variable en la muestra*

n **=** len(muestra\_titanic) *# Tamaño de la muestra*

Pi **=** 0.30 *# Proporción o frecuencia que queremos probar en la población*

*# Calcular el estadístico Z y el p-valor*

z\_stat, p\_value **=** proportions\_ztest(p\_hat **\*** n, n, Pi, alternative **=** "larger") *# Fijate que los valores alternative son diferentes a los usados t-test*

*# Mostrar resultados*

print(f"Estadístico Z: {z\_stat}")

print(f"P-valor: {p\_value}")

Estadístico Z: 3.60377503215186

P-valor: 0.0001568142715050189

El p-value obtenido nos lleva a rechazar la hipótesis nula y por tanto podemos decir con una confianza estadística del 95% que la tasa de supervivencia en la población es mayor al 30%.

**Z-Test para comprobar proporciones y ratios en dos poblaciones**

Igual que en la sesión anterior comparamos dos muestras de llamadas de dos callcenter diferentes para poder decir si la media de la duración de llamadas era diferetne, podemos hacer lo mismo respecto a la frecuencia o ratio o proporción de una variable categórica medida en dos muestras distintas.

Por ejemplo:

* Efectividad de medicamentos (proporción de personas que han mejorado tras un tratamiento con dos medicamentos distintos o un medicamento y un placebo)
* Resultados electorales (proporción de personas que votan a cada candidato)
* Comportamiento de consumidores (proporción de consumidores que compran un producto)
* Resultados de encuestas (proporción de personas que responden en dos regiones diferentes)
* **A/B testing**: En estudios de mercado, se implementan dos versiones de un producto (A y B) y se prueban en diferentes grupos de usuarios

Veamos un ejemplo del A/B testing:

Diagrama

Descripción generada automáticamente

**Ejemplo**: Para conseguir registros en nuestra web, creamos dos secciones de registro diferentes (una más colorida, y otra más sobria).

El primer diseño se muestra a 200 personas, de las cuales conseguimos 130 registros (65% de conversiones)  
El segundo diseño se muestra a 100 personas, de las cuales conseguimos 56 registros (56% de conversiones)

¿Es esta diferencia estadísticamente significativa? ¿Qué opción escojo?

1. Definimos las hipótesis, en nuestro caso la alternativa es que la primera tiene un ratio de conversión mayor que la segunda:  
   Texto, Carta

   Descripción generada automáticamente
2. Calculamos el estadístico (fíjate en cómo introducimos las dos proporciones):

exitos **=** np.array([130,56])

muestras **=** np.array([200,100])

(z,pvalor) **=** proportions\_ztest(count**=**exitos, nobs**=**muestras, alternative**=**'larger')

print(z)

print(pvalor)

1.5139419980680997

0.06502027550522334

No podemos rechazar la hipótesis nula con un 95% de confianza y por tanto, no podemos decir que una de las webs haya resultado mejor que la otra con ese nivel de confianza estadística (ojo fíjate que sí podríamos decirlo con un 90% de confianza)