TALLER DE MÉTODOS CUANTITATIVOS

Apunte Semana 06

Apunte Semana 04, taller de métodos cuantitativos.
Universidad Andrés Bello, Santiago.

Contents

1 Estimación

1.1 Error absoluto y tamaño de la muestra

En esta sección abordamos el problema de determinar el tamaño adecuado de una muestra. ¿Cuántos datos son necesarios para garantizar la validez de nuestro estudio? Esta es una pregunta general que no tiene una respuesta única, ya que requiere precisar qué esperamos obtener de nuestros datos. En este contexto, nos enfocamos en responder la siguiente pregunta: ¿Cuántos datos necesitamos recolectar para que, al estimar una media poblacional, el error máximo que cometemos sea inferior a un valor previamente establecido? Por ejemplo, supongamos que queremos determinar la concentración promedio de un elemento en una región. Deseamos estimar este valor, representado por μ, con un error máximo de δ unidades, donde δ es un número positivo que definimos previamente.

Es importante destacar que, en Estadística, no podemos asegurar nada con absoluta certeza. Podemos, sin embargo, trabajar bajo altos niveles de probabilidad o confianza, aunque garantizar que algo suceda con seguridad es imposible. Todas nuestras afirmaciones se basan en la probabilidad de que ciertos eventos ocurran o no, y, por tanto, no podemos asegurar que el error será siempre menor a un determinado nivel δ.

Para resolver esta cuestión, utilizaremos la base de datos *mpg* de la librería seaborn. El objetivo es estimar el rendimiento promedio μ de los vehículos. Ya hemos calculado un intervalo de confianza para μ con un nivel de confianza de $1-\alpha$. Por ejemplo, cuando $\alpha = 0.05$, el intervalo queda definido como:

```
import seaborn as sns
data = sns.load_dataset("mpg")
# Media y desviación muestral
media = np.mean(data["mpg"])
desviacion = np.std(data["mpg"], ddof=1)
n= len(data)
# Nivel de confianza
confianza = 0.95
# intervalo
Intervalo_confianza = ss.t.interval(confianza, df=n-1, media, desviacion/np
    .sqrt(n))
## (22.7443502740561, 24.284795454587115)
```

Con un nivel de confianza del 95%, consideramos que el valor de μ probablemente se encuentra dentro del intervalo obtenido. Suponiendo que todo ha salido según lo esperado, asumimos que μ está efectivamente contenido en el intervalo. Aunque no podemos afirmarlo con certeza absoluta, confiamos en que este nivel de confianza es suficiente para respaldar nuestra suposición. Por lo tanto, ante la pregunta: ¿cuál es tu estimación para μ?, respondemos que es

```
np.mean(data["mpg"])

## [1] 23.51457
```

En otras palabras, respondemos utilizando el valor de la media muestral $\overline{x}$ calculada a partir de las

observaciones obtenidas para construir el intervalo. Esta media muestral corresponde al punto medio del intervalo. Por lo tanto, si μ se encuentra dentro del intervalo, la distancia entre μ y el centro será, como máximo, igual a la mitad de la longitud del intervalo, cuyo valor es

```
## [1] 0.77022
```

¿Cuál es el error absoluto asociado a la muestra que hemos tomado? Para calcularlo, almacenamos el intervalo de confianza en `Intervalo_confianza`.

```
Intervalo_confianza = ss.t.interval(
       0.95,
       df = n-1,
       loc = media,
       scale = desviacion/np.sqrt(n)
)
```

Sabemos que el error absoluto es la mitad de la longitud del intervalo de confianza. Lo calculamos.

```
lim_inf, lim_sup = intervalo_confianza
(lim_sup - lim_inf)/2
## [1] 0.770222590265508
```

La expresión que define el intervalo de confianza para la media de una población normal es la siguiente.

$$\left[\overline{X} - t_{n-1,1-\frac{\alpha}{2}}\frac{S}{\sqrt{n}}, \overline{X} + t_{n-1,1-\frac{\alpha}{2}}\frac{S}{\sqrt{n}}\right]$$

Por lo tanto, el error absoluto se calcula mediante la siguiente expresión.

$$t_{n-1,1-\frac{\alpha}{2}}\frac{S}{\sqrt{n}}$$

En `Python`, esta fórmula se implementa utilizando el siguiente código:

```
alpha = .05
n = len(data)
ss.t.ppf(1-alpha/2, df=n-1)*(desviacion/np.sqrt(n))

## [1] 0.770222590265507
```

Asumimos que la desviación estándar permanecerá prácticamente igual aunque se incrementen las muestras.

```
desv = np.std(data["mpg"])
```

Establecemos un valor para el error máximo que estamos dispuestos a aceptar.

```
delta=0.2
```

Es necesario comprobar que se cumple:

$$t_{n-1,1-\frac{\alpha}{2}} \frac{S}{\sqrt{n}} \leq \delta$$

Si consideramos que s permanece aproximadamente constante.

```
m = n+10
ss.t.ppf(1-alpha/2, df=m-1)*(desv/np.sqrt(m))

#[1] 0.760667882522969
```

Observamos que no es suficiente. ¿Qué sucedería si añadimos 100 observaciones más?

```
m = n+100
ss.t.ppf(1-alpha/2, df=m-1)*(desv/np.sqrt(m))

#[1] 0.6881387253307018
```

¿Y si agregamos 200 observaciones adicionales?

```
m = n+200
ss.t.ppf(1-alpha/2, df=m-1)*(desv/np.sqrt(m))

#[1] 0.6277150397254374
```

Tal vez hemos exagerado. Probemos ahora con 150 observaciones.

```
m = n+150
ss.t.ppf(1-alpha/2, df=m-1)*(desv/np.sqrt(m))

#[1] 0.6558484489575538
```

Una forma de realizarlo sería la siguiente.

```
for m in range(550, 560): # range(289, 291) incluye 289 y 290
    valor = ss.t.ppf(1 - alpha/2, df=m-1) * (desv / np.sqrt(m))
    print(f"m = {m}, resultado = {valor}")

## m = 550, resultado = 0.6538266814229902
## m = 551, resultado = 0.6532304787832416
## m = 552, resultado = 0.6526359041526755
## m = 553, resultado = 0.652042950135572
## m = 554, resultado = 0.6514516093831637
## m = 555, resultado = 0.6508618745932504
## m = 556, resultado = 0.6502737385098226
## m = 557, resultado = 0.6496871939226847
## m = 558, resultado = 0.6491022336670857
## m = 559, resultado = 0.6485188506233497
```

Para observar esto, te recomendamos revisar el siguiente video en donde se explica este ejemplo paso a paso.

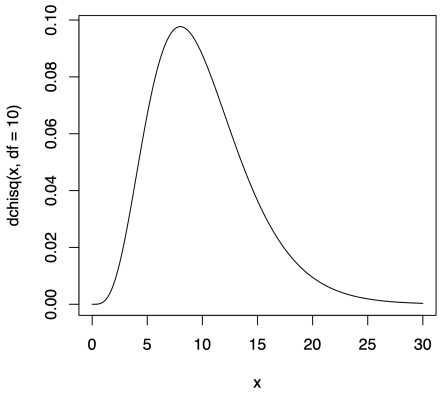
AGREGAR LINK

1.2 Estimación de la varianza en poblaciones normales

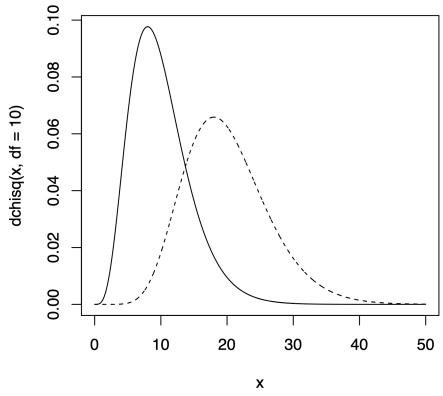
Si $X_1, ..., X_n$ es una muestra alearotia de una normal con media μ y varianza σ^2, entonces:

$$\sum_{i=1}^{n} \frac{(X_i - \overline{X}_n)^2}{\sigma^2} = \frac{(n-1)S^2}{\sigma^2} \sim X^2_{n-1}$$

Se afirma que la variable aleatoria $\frac{(n-1)S^2}{\sigma^2}$ sigue una distribución ji-cuadrado con $n-1$ grados de libertad. En la siguiente figura se ha ilustrado la función de densidad de una ji-cuadrado con 10 grados de libertad.



Para analizar cómo varía la forma de la función de densidad al cambiar el número de grados de libertad, en la figura que a continuacón se presenta, se muestran las densidades correspondientes a una ji-cuadrado con 10 grados de libertad (representada con un trazo continuo) y con 20 grados de libertad (representada con un trazo discontinuo).



El percentil de orden p de una ji-cuadrado con k grados de libertad se denota como $\chi^2_{p,k}$. En otras palabras, si la variable X sigue una distribución ji-cuadrado con k grados de libertad, es decir, $X \sim \chi^2_k$, entonces se cumple que:

$$P(X \leq \chi^2_{p,k}) = p$$

El valor correspondiente se puede calcular utilizando la función `qchisq`.

```
import scipy.stats as ss

p = 0.75
k = 13

# Calcular el cuantil para la distribución chi-cuadrado
qchisq_value = ss.chi2.ppf(p, df=k)
print(qchisq_value)
## 15.983906216312052
```

Por lo que obtenemos:

$$P\left(X^2_{\alpha/2,n-1} \leq \frac{(n-1)S^2}{\sigma^2} \leq X^2_{1-\alpha/2,n-1}\right) = 1-\alpha$$

Entonces:

$$P\left(\frac{(n-1)S^2}{X^2_{1-\alpha/2,n-1}} \leq \sigma^2 \leq \frac{(n-1)S^2}{X^2_{\alpha/2,n-1}}\right) = 1-\alpha$$

Es decir, el intervalo:

$$\left[\frac{(n-1)S^2}{X^2_{1-\alpha/2,n-1}}, \frac{(n-1)S^2}{X^2_{\alpha/2,n-1}}\right]$$

El intervalo de confianza para σ^2 con un nivel de confianza $1-\alpha$ está claramente definido. De manera similar, el intervalo de confianza para la desviación estándar poblacional, con el mismo nivel de confianza, está dado por:

$$\left[\sqrt{\frac{(n-1)S^2}{\chi^2_{1-\alpha/2,n-1}}}, \sqrt{\frac{(n-1)S^2}{\chi^2_{\alpha/2,n-1}}}\right]$$

A continuación, exploraremos cómo realizar este cálculo en `Python`. La función que permite obtener los percentiles de la distribución ji-cuadrado es `chi2.ppf`. Procedemos con la recopilación de los datos.

```
# Consideremos
X = data["mpg"]
n = 100
x = np.random.choice(X,n)
```

Procedemos a calcular el intervalo de confianza.

```
alpha = .05
s2 = np.var(x)
extremoinferior = (n-1)*s2 / ss.chi2.ppf(1-alpha/2,df=n-1)
## [1] 49.11355343119836
extremosuperior = (n-1)*s2 / ss.chi2.ppf(alpha/2,df=n-1)
## [1] 85.97556339893413
```

El intervalo de confianza para σ^2 con un nivel de confianza del 95% está definido como [49.1135 , 85.9755].

Para observar esto, te recomendamos revisar el siguiente video en donde se explica este ejemplo paso a paso.

AGREGAR LINK

1.3 Estimación de una proporción

Supongamos que denotamos por p la proporción que deseamos estimar (por ejemplo, la proporción de aprobados en un examen o la proporción de votantes de un partido político). Antes de recolectar los datos, asumimos que contamos con una muestra aleatoria $X_1, \ldots, X_n$ donde cada variable aleatoria X_i puede tomar los valores 1 o 0 con probabilidades p y $1-p$, respectivamente. El estimador de p se define como:

$$\hat{p} = \frac{\sum_{i=1}^{n} X_i}{n}$$

De acuerdo con el teorema central del límite, sabemos que, aproximadamente (si el tamaño de la muestra n es grande), $\hat{p}$ sigue una distribución normal con media p y varianza $p(1-p)/n$. Es decir:

$$\hat{p} \sim N\left(p, \frac{p(1-p)}{n}\right)$$

Esto se puede reescribir de la siguiente manera:

$$\frac{\hat{p}-p}{\sqrt{\frac{p(1-p)}{n}}} \sim N(0,1)$$

Sin embargo, esta expresión no es práctica para construir un intervalo de confianza. Una alternativa es estimar la varianza utilizando $\hat{p}(1-\hat{p})/n$ y asumir que, aproximadamente (lo que implica que necesitamos una muestra grande), se cumple:

$$\frac{\hat{p}-p}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}} \sim N(0,1)$$

Si denotamos el percentil de orden γ $(0 < \gamma < 1)$ de una distribución normal estándar como Z_γ, se verifica para un α dado:

$$P\left(-Z_{1-\alpha/2} \leq \frac{\hat{p}-p}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}} \leq Z_{1-\alpha/2}\right) = 1-\alpha$$

o, de forma equivalente:

$$P\left(\hat{p} - Z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \leq p \leq \hat{p} + Z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right) = 1-\alpha$$

1.3.1 Intervalo de confianza para una proporción

Si $X_1, \ldots, X_n$ es una muestra aleatoria de variables binomiales con una única prueba y probabilidad de éxito p, entonces el intervalo:

$$\hat{p} \pm Z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

es un intervalo de confianza para la proporción p, donde:

$$\hat{p} = \frac{\sum_{i=1}^{n} X_i}{n}$$

Table 1: Efecto preventivo de la aspirina

Tratamiento	Ataque fatal y no fatal	No ataque
Placebo	189	10845
Aspirina	104	10933

Utilizaremos Python para calcular este intervalo de confianza. Los datos que emplearemos evalúan si una persona ha tomado aspirina o placebo y si ha sufrido un ataque cardíaco. Estos datos se presentan en la tabla. Procederemos a estimar la proporción de personas que experimentan un ataque cardíaco tras consumir un placebo.

```
import statsmodels.api as sm
from statsmodels.stats.proportion import proportion_confint

# Parámetros
x = 189       # número de éxitos
n = 11034     # número de ensayos
alpha = 0.05  # nivel de significancia para un intervalo del 95%

# Calcular el intervalo de confianza asintótico (método normal)
lower, upper = proportion_confint(count=x, nobs=n, alpha=alpha, method='
   normal')

print(f"Intervalo de confianza al 95%: ({lower}, {upper})")

## 0.014707877106365601, 0.01954987167014337
```

De manera similar, es posible calcular la proporción de personas que sufren un ataque cardíaco al tomar aspirina. La estimación puntual y el intervalo de confianza correspondientes se presentan en el siguiente resultado.

```
# Parámetros
x = 104       # número de éxitos
n = 11037     # número de ensayos
alpha = 0.05  # nivel de significancia para un intervalo del 95%

# Calcular el intervalo de confianza asintótico (método normal)
lower, upper = proportion_confint(count=x, nobs=n, alpha=alpha, method='
   normal')

print(f"Intervalo de confianza al 95%: ({lower}, {upper})")
## 0.007620422638288568, 0.011225278186210842
```

Aparentemente, tomar aspirina reduce la probabilidad de sufrir un ataque cardíaco.

Para observar esto, te recomendamos revisar el siguiente video en donde se explica este ejemplo paso a paso.

AGREGAR LINK