Tarea 3. Estimación por intervalos

17 de enero de 2022

2. Genere una muestra aleatoria $X_1, \ldots, X_n$, de tamaño $n = 30$, de una población con distribución $\mathcal{N}(5, 4)$. Genere otra muestra aleatoria $Y_1, \ldots, Y_m$, de tamaño $m = 50$, de una población con distribución $\mathcal{N}(2, 4)$. Obtenga los intervalos de confianza para $\mu_x - \mu_y$ tales que

   a) Intervalo del $80\,\%$ de confianza, suponiendo σ^2 conocida.
   b) Intervalo del $80\,\%$ de confianza, suponiendo σ^2 desconocida común.
   c) Intervalo del $95\,\%$ de confianza, suponiendo σ^2 conocida.
   d) Intervalo del $95\,\%$ de confianza, suponiendo σ^2 desconocida común.

   Repita el proceso generando cada una de las muestras 100 veces. ¿Cómo son los intervalos? Identifique los intervalos con mayor longitud y con menos longitud. Compare y explique los resultados.

   Solución:

   $a)$ Usaremos que un intervalo al $100(1-\alpha)\,\%$ de confianza para $\mu_x - \mu_y$ cuando σ^2 es conocida es (el cual está dado en el libro del profe)

$$\left((\bar{X} - \bar{Y}) - z_{1-\alpha/2}\sqrt{\frac{\sigma^2}{n} + \frac{\sigma^2}{m}}, (\bar{X} - \bar{Y}) + z_{1-\alpha/2}\sqrt{\frac{\sigma^2}{n} + \frac{\sigma^2}{m}} \right) \tag{1}$$

   De tal manera comenzamos generando las muestras aleatorias con las indicaciones dadas

```r
n <- 30
m <- 50

mx <- rnorm(n, mean = 5, sd=2)
my <- rnorm(m, mean = 2, sd=2)
```

   y después hacemos los cálculos pertientes

```r
xbarra <- sum(mx) / n
ybarar <- sum(my) / m

# notemos que alpha = 0.2
```

```r

alfamedios <- 1 - 0.2 / 2
z <- pnorm(alphamedios, mean = 0, sd = 1)

r <- ( (4 / n) + (4 / m) ) ** 0.5
```

Después definimos y cálculamos los límites inferior y superior del intervalo de confianza dado en (1):

```r
lim_inferior <- xbarra - ybarra - z * r
lim_superior <- xbarra - ybarra + z * r
```

Finalmente vemos el intervalo del $80\,\%$ de confianza para $\mu_x - \mu_y$:

```r
print(paste("(", lim_inferior, ",", lim_superior, ")"))

## "( 2.64181502427697 , 3.39554799431473 )"
```

b) Usaremos que un intervalo al $100(1-\alpha)\,\%$ para $\mu_x - \mu_y$, cuando σ^2 es desconocido es (el cual está dado en el libro del profe)

$$\left((\bar{X} - \bar{Y}) - t_{n+m-2}^{1-\alpha/2} \sqrt{\left(\frac{1}{n} + \frac{1}{m}\right) S_p^2}, (\bar{X} - \bar{Y}) + t_{n+m-2}^{1-\alpha/2} \sqrt{\left(\frac{1}{n} + \frac{1}{m}\right) S_p^2} \right) \tag{2}$$

Donde

$$S_p^2 = \frac{(n-1)S_x^2 + (m-1)S_y^2}{n+m-2}$$

De forma análoga al inciso anterior haremos los respectivos cálculos:

```r
# cuantil para la t

alphamedios <- 1 - 0.2 / 2
t <- qt(uno_alphamedios, df = n + m - 2)

# S cuadrada

Sx <- ( 1 / (n-1) ) * sum( (mx - xbarra) ** 2 )
Sy <- ( 1 / (m-1) ) * sum( (my - ybarra) ** 2 )

# Ademas

Sp <- ( (n-1) * Sx + (m-1) * Sy ) / (n + m -2)

r <- ( ((1 / n) + (1 / m)) * Sp ) ** 0.5
```

para posteriormente crear el intervalo

```
# limites

lim_inferior <- xbarra - ybarra - t * r
lim_superior <- xbarra - ybarra + t * r

# intervalo

print(paste("(", lim_inferior, ",", lim_superior, ")"))

## "( 1.98675117512603 , 3.18265949255385 )"
```

Incisos $c)$ y $d)$: resolveremos estos ejercicios a la par. De hecho, bastará con reutilizar el código de los incisos $a)$ y $b)$:

Código para el inciso c)

```
# En este caso alpha = 0.05

alphamedios <- 1 - 0.05 / 2
z <- pnorm(alphamedios, mean = 0, sd = 1)

r <- ( (4 / n) + (4 / m) ) ** 0.5

# limites

lim_inferior <- xbarra - ybarra - z * r
lim_superior <- xbarra - ybarra + z * r

# Resultado
print(paste("(", lim_inferior, ",", lim_superior, ")"))

## "( 2.53351045798609 , 3.30505352484752 )"
```

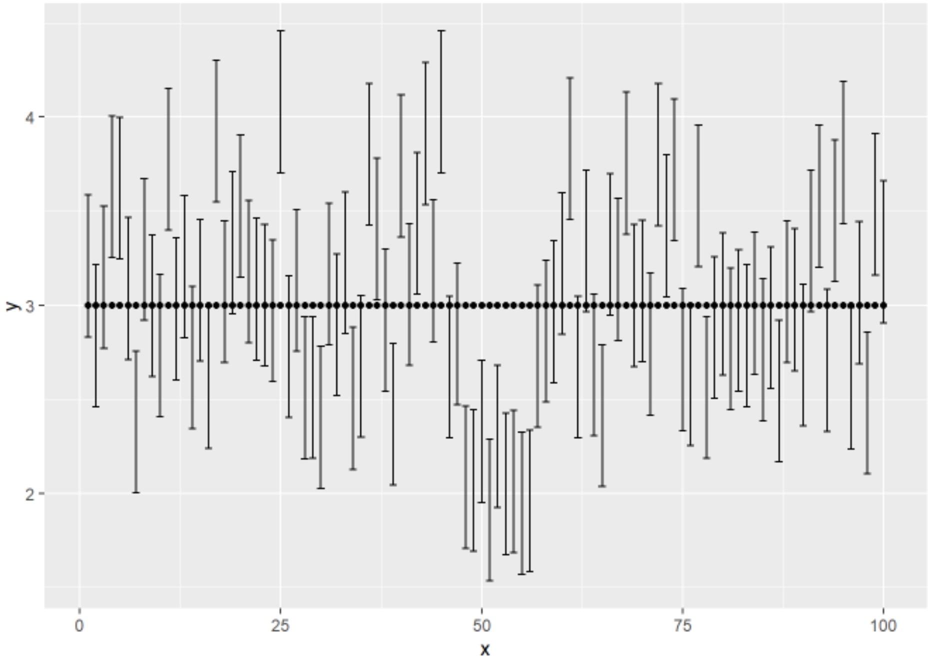
Código para el inciso d)

```
# notemos que alpha = 0.05

alphamedios <- 1 - 0.05 / 2
t <- qt(alphamedios, df = n + m - 2)

# S cuadrada

Sx <- ( 1 / (n-1) ) * sum( (mx - xbarra) ** 2 )
Sy <- ( 1 / (m-1) ) * sum( (my - ybarra) ** 2 )

# Sp

Sp <- ( (n-1) * Sx + (m-1) * Sy ) / (n + m -2)

r <- ( ((1 / n) + (1 / m)) * Sp ) ** 0.5

# limites

lim_inferior <- xbarra - ybarra - t * r
lim_superior <- xbarra - ybarra + t * r

# intervalo

print(paste("(", limite_inferior, ",", limite_superior, ")"))

## "( 1.74441177103715 , 3.52176604804797 )"
```

Finalmente, para la parte final nos auxiliaremos del código que el ayudante compartió.

Haremos la simulación 100 ayudándonos de un *for* y del código que escribimos en los incisos anteriores

```
# Cargamos las librer as
library(ggplot2)
library(dplyr)

# nos auxiliamos de dos vectores
lim_inf <- c()
lim_sup <- c()

# hacemos la simulaci n 100
for( i in 1:100 ){
  mx <- rnorm(n, mean = 5, sd=2)
  my <- rnorm(m, mean = 2, sd=2)
  xbarra <- sum(mx) / n
  ybarra <- sum(my) / m
  alphamedios <- 1 - 0.2 / 2
  z <- pnorm(alphamedios, mean = 0, sd = 1)
  r <- ( (4 / n) + (4 / m) ) ** 0.5
  lim_inferior <- x_bar - y_bar - z * r
  lim_superior <- x_bar - y_bar + z * r
  lim_inf[i] <- lim_inferior
  lim_sup[i] <- lim_superior
}

x <- 100
df <- data.frame(x = x,
                 y = 3,
                 limite_inferior = lim_inf,
                 limite_superior = lim_sup)
df <- df %>% mutate(long_int = limite_superior - limite_inferior)
df <- df %>% arrange(long_int)

df['x'] <- 1:100

ggplot(df, aes(x, y)) + geom_point() +
  geom_errorbar(aes(ymin = limite_inferior, ymax = limite_superior
    ))
```



Luego, para simular 100 veces una muestra cuando la σ^2 es desconocida usamos

```

# nos auxiliamos de dos vectores
lim_inf <- c()
lim_sup <- c()

# hacemos la simulaci n 100
for( i in 1:100 ){
  mx <- rnorm(n, mean = 5, sd=2)
  my <- rnorm(m, mean = 2, sd=2)
  xbarra <- sum(mx) / n
  ybarra <- sum(my) / m
  alphamedios <- 1 - 0.2 / 2
  t <- qt(alphamedios, df = n + m - 2)
  Sx <- ( 1 / (n-1) ) * sum( (mx - xbarra) ** 2 )
  Sy <- ( 1 / (m-1) ) * sum( (my - ybarra) ** 2 )
  Sp <- ( (n-1) * Sx + (m-1) * Sy ) / (n + m -2)
  r <- ( ((1 / n) + (1 / m)) * Sp ) ** 0.5
  lim_inferior <- x_bar - y_bar - t * r
  lim_superior <- x_bar - y_bar + t * r
  lim_inf[i] <- lim_inferior
  lim_sup[i] <- lim_superior
}

x <- 100
df <- data.frame(x = x,
                 y = 3,
```

```
                    limite_inferior = lim_inf,
                    limite_superior = lim_sup)
df <- df %>% mutate(long_int = limite_superior - limite_inferior)
df <- df %>% arrange(long_int)

df['x'] <- 1:100

ggplot(df, aes(x, y)) + geom_point() +
  geom_errorbar(aes(ymin = limite_inferior, ymax = limite_superior
    ))
```

Después debemos repetir el proceso para:

c. Intervalos del $95\,\%$ de confianza cuando la σ^2 es conocida
d. Intervalos del $95\,\%$ de confianza cuando la σ^2 es desconocida

para lo cual el código des las páginas anteriores sirve y tan sólo debemos cambiar el valor de α.

Faltan escribir las conclusiones.

Referencias

[1] Hogg, Robert. et.al. (2018). Introduction to Mathematical Statistics. Pearson.

[2] Ross, Sheldon. (2010). A First Course In Probability. Pearson.