

04_ErroresTipo-I-II

Alicia Gil Matute

2024-09-21

Contents

Contrastes de hipótesis	1
Errores Tipo I	1
Tarea 1	2
Tarea 2	5
Tarea 3	6
Errores Tipo II	8
Tarea 4	9
Tarea 5	11
Tarea 6	13

Contrastes de hipótesis

Se pueden cometer dos errores en un contraste de hipótesis:

- Errores tipo I: α , nivel de significación. Probabilidad de rechazar la H_0 cuando es verdadera.
- Errores tipo II: β . Probabilidad de aceptar la H_0 cuando esta es falsa.

Errores Tipo I

Los errores tipo I son prefijados antes del contraste por el investigador, y los llamamos α teóricos (α_{teo}), sin embargo es posible que ciertas circunstancias, como el incumplimiento de los supuestos de la prueba, lleve a que estemos cometiendo más error del establecido a priori, es lo que vamos a llamar el α empírico (α_{emp}).

En realidad, la comprobación de este α_{emp} requiere más conocimiento de lo que vamos a utilizar en esta práctica, es decir, lo que vamos a hacer no es del todo correcto, solo una simplificación, pero nos valdrá para poner a prueba la idea principal.

Si extraemos k muestras de una población que es normal $N(\mu_0, \sigma)$ y realizamos un contraste para una media en el que ponemos a prueba la $H_0 : \mu = \text{media}$ deberíamos encontrarnos con que rechazaríamos la H_0 el

5% de esas k veces (si hemos establecido $\alpha_{teo} = 0.05$). Es decir, partimos de un $\alpha_{teo} = 0.05$ y queremos comprobar si realmente nuestro α_{emp} es también igual a 0.05.

Si hacemos un único experimento es posible que α_{teo} y α_{emp} coincidan, pero si repitiéramos el experimento más veces (una nueva población, una nueva extracción, etc), podríamos obtener resultados diferentes.

Tarea 1

Realiza simulaciones para comprobar si se mantienen iguales ambos alfas (teórico y empírico) cuando partimos de variables de rango limitado (y por tanto no normales). Crea un gráfico en el que se pueda ver si cambia el alfa empírico desde un rango de valores de: 2,3,4 y 5.

```
set.seed(4)

expo1 <- function(N,n,k){

  poblacion <- sample(1:2,N,replace = T) #distribucion no normal
  mu_pob <- mean(poblacion)
  sd_pob <- sd(poblacion)

  p <- c()

  for (i in 1:k){

    muestra <- poblacion[sample(1:N,n)]
    p[i] <- t.test(muestra,mu=mu_pob)$p.value
  }

  a_teo <- 0.05
  a_emp <- length(p[p<a_teo])/k

  return(a_emp)
}

#500 repeticiones, tamaño poblacional=1000, tamaño muestral=25

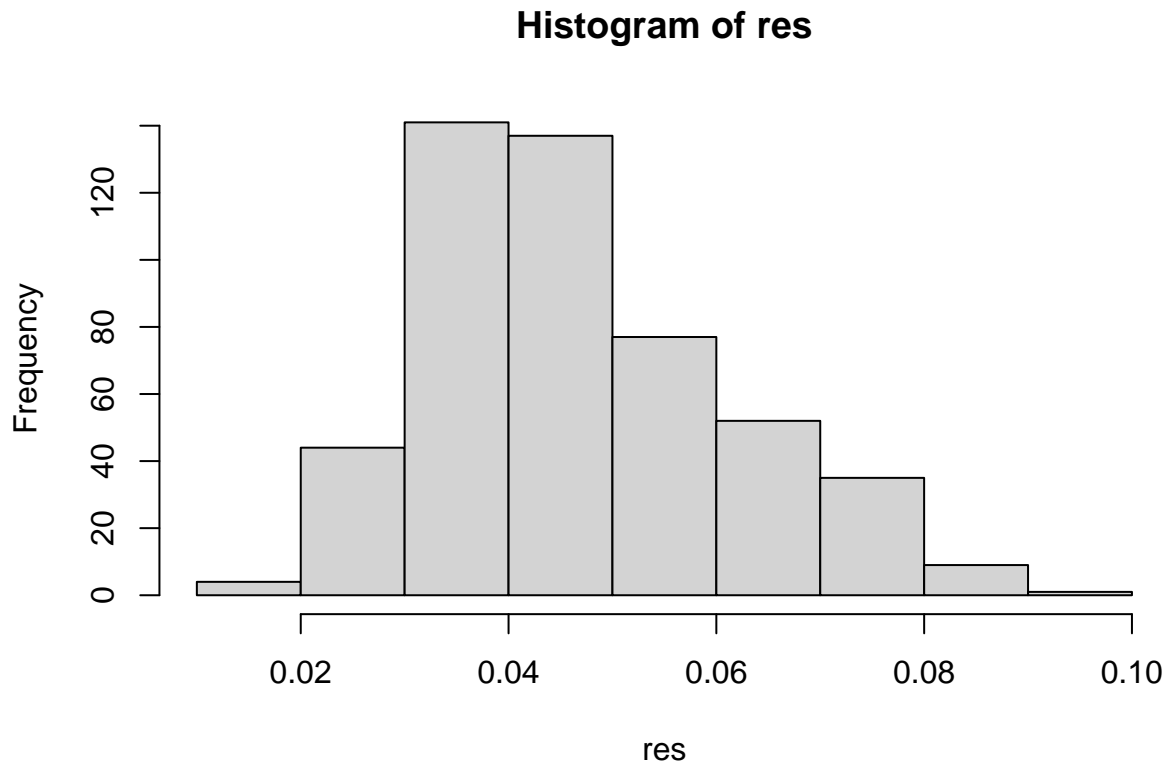
rep <- 500

res <- c()

for (i in 1:rep){

  res[i] <- expo1(1000,25,500)
}

hist(res)
```



```
grafico <- function(longitud) {  
  
  rep <- 100  
  res <- numeric(rep)  
  
  # Inicializar los vectores  
  datos_1 <- numeric(longitud)  
  des_1 <- numeric(longitud)  
  lim_sup <- numeric(longitud)  
  lim_inf <- numeric(longitud)  
  
  # Loop para calcular el alfa empírico y los intervalos de confianza  
  for (i in 1:longitud) {  
    for (j in 1:rep) {  
      res[j] <- expo1(1000, 25, 500) # Usar la función correcta (expo1)  
    }  
  
    # Promedio y desviación estándar de las repeticiones  
    datos_1[i] <- mean(res)  
    des_1[i] <- sd(res)  
  
    # Intervalos de confianza al 95%  
    lim_sup[i] <- datos_1[i] + 1.96 * (des_1[i] / sqrt(rep))  
    lim_inf[i] <- datos_1[i] - 1.96 * (des_1[i] / sqrt(rep))  
  }  
}
```

```

# Crear el rango para el eje x
rangos <- 1:longitud

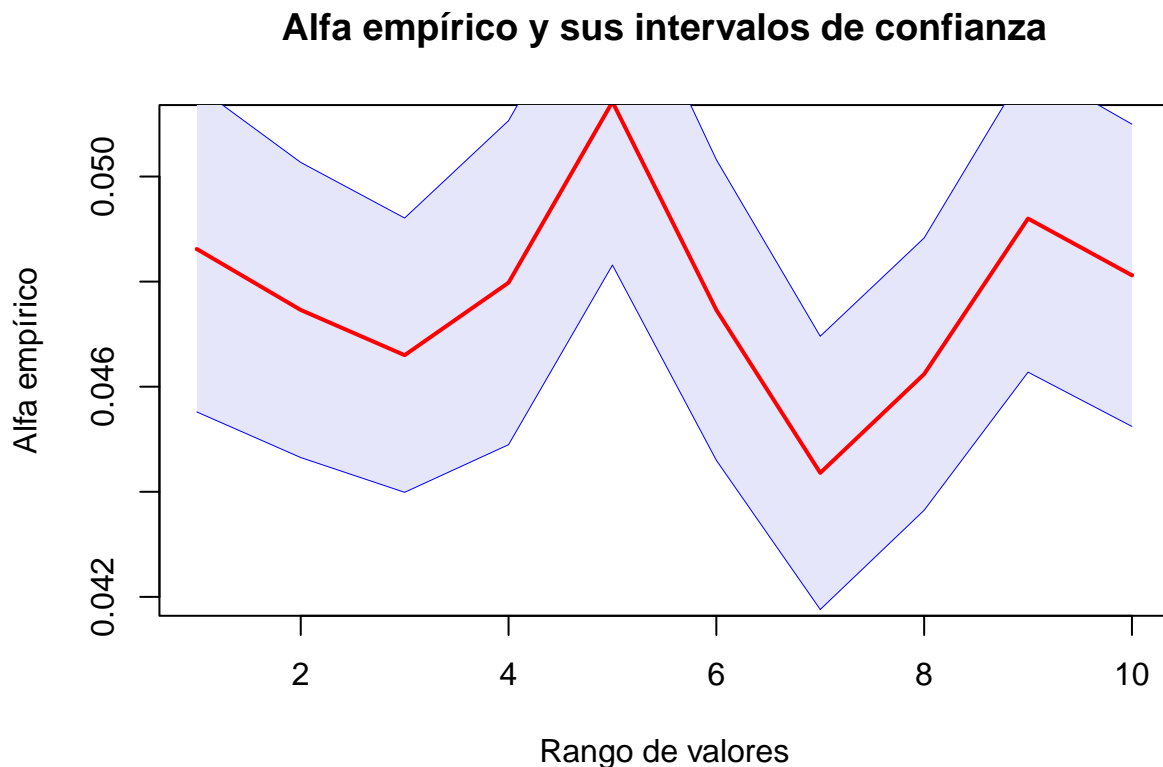
# Gráfico
plot(rangos, datos_1, ylim = c(0.042, 0.051), type = "l", lwd = 2, col = "red",
     xlab = "Rango de valores", ylab = "Alfa empírico",
     main = "Alfa empírico y sus intervalos de confianza")

# Añadir las zonas de los intervalos de confianza
polygon(c(rangos, rev(rangos)), c(lim_sup, rev(lim_inf)),
       col = "lavender", border = NA)

# Añadir las líneas de los valores de alfa empírico y los límites superior e inferior
lines(rangos, datos_1, type = "l", lwd = 2, col = "red")
lines(rangos, lim_sup, type = "l", lwd = 0.5, col = "blue")
lines(rangos, lim_inf, type = "l", lwd = 0.5, col = "blue")
}

# Ejecutar la función con un rango de 5 (incluye 2,3,4,5)
rango <- 10
grafico(rango)

```



Tarea 2

Realiza simulaciones para comprobar cómo es la relación entre los dos alfas al aumentar el tamaño de la muestra ($n=2,10,15,20$). Haz un gráfico.

```
set.seed(4)

a_teo <- 0.05

expo1 <- function(N,n,k){

  poblacion <- rnorm(N,0,1) #distribucion normal
  mu_pob <- mean(poblacion)
  sd_pob <- sd(poblacion)

  p <- c()

  for (i in 1:k){

    muestra <- poblacion[sample(1:N,n)]
    p[i] <- t.test(muestra,mu=mu_pob)$p.value
  }

  a_emp <- length(p[p<a_teo])/k

  return(a_emp)
}

rep <- 500

tamaños <- c(2,10,15,20)

res <- c()
res_sd <- c()
lim_sup <- c()
lim_inf <- c()

for (i in 1:length(tamaños)){

  res_in <- c()

  for (j in 1:rep){

    res_in[j] <- expo1(1000,tamaños[i], 500)
    res[i] <- mean(res_in)
    res_sd <- sd(res_in)
    lim_sup[i] <- res[i]+1.96*(res_sd[i]/sqrt(rep))
    lim_inf[i] <- res[i]-1.96*(res_sd[i]/sqrt(rep))
  }
}

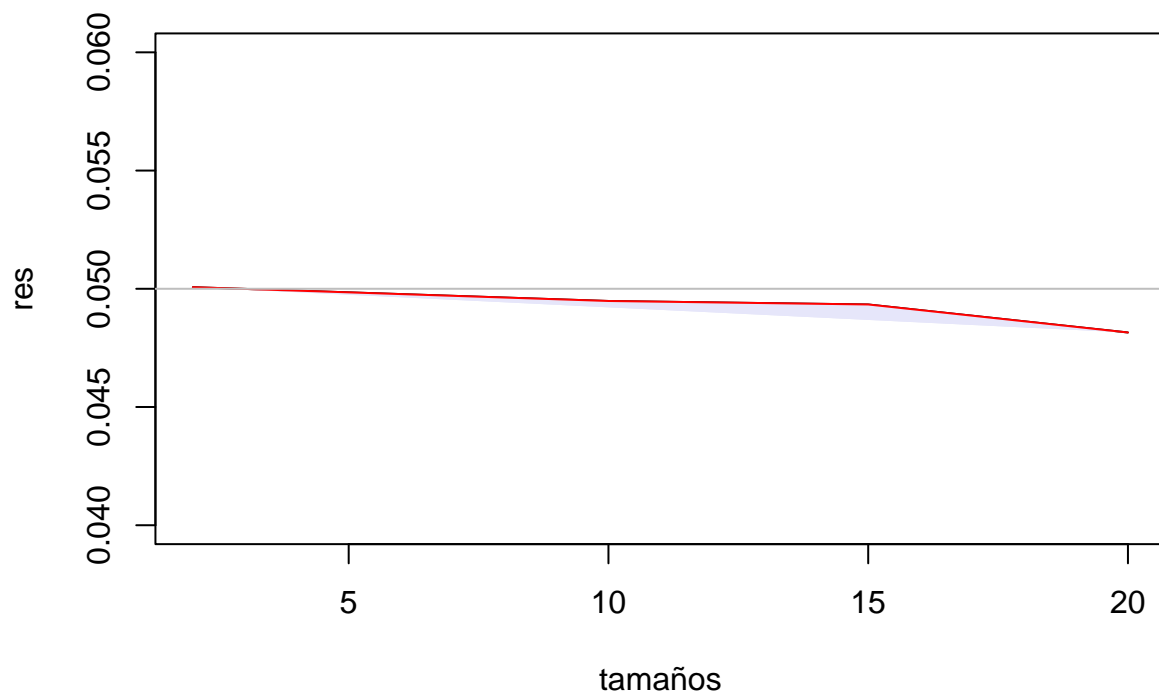
plot(tamaños, res, ylim=c(0.040,0.060), type="l")
```

```

polygon(c(tamaños, rev(tamaños)), c(lim_sup, rev(res)),
       col="lavender", border=NA)

polygon(c(tamaños, rev(tamaños)), c(lim_inf, rev(res)),
       col="lavender", border=NA)
lines(tamaños, res, type="l", lwd=1, col="red")
lines(tamaños, lim_sup, type="l", lwd=0.5, lty=2)
lines(tamaños, lim_inf, type="l", lwd=0.5, lty=2)
abline(a_teo, 0, col="grey")

```



Tarea 3

Utilizando la distribución de Johnson estudia cómo afecta al contraste de una media que la distribución se vaya haciendo más y más asimétrica (utiliza dos o tres niveles de asimetría).

```

set.seed(4)

library(SuppDists)

```

```
## Warning: package 'SuppDists' was built under R version 4.3.3
```

```
expo9 <- function(N,n,k,s){
```

```

parms <- JohnsonFit(c(0,2,s,20), moment="use")
sJohnson(parms)
poblacion <- rJohnson(N,parms)

mu_pob <- mean(poblacion)
sd_pob <- sd(poblacion)

p <- c()

for(i in 1:k){

  muestra <- poblacion[sample(1:N, n)]
  pi[i] <- t.test(muestra,mu=mu_pob)$p.value

}

a_teo <- 0.05
a_emp <- length(p[p<a_teo])/k

return(a_emp)
}

#Diferentes niveles de sesgo

sesgo <- c(0,3,0.5)

errorI <- c()

c=1

res_sd <- c()
lim_sup <-c()
lim_inf <- c()

for(j in sesgo){
  rep <- 500
  res <- c()

  for (i in 1:rep){res[i] <- expo9(1000,10,500,j)}

  errorI[c] <- mean(res)
  res_sd[c] <- sd(res)
  lim_sup[c] <- errorI[c]+1.96*(res_sd[c]/sqrt(rep))
  lim_inf[c] <- errorI[c]-1.96*(res_sd[c]/sqrt(rep))

  c=c+1

  sd(res)
}

plot(sesgo, errorI, type="l")
polygon(c(sesgo, rev(sesgo)), c(lim_sup,rev(errorI)),

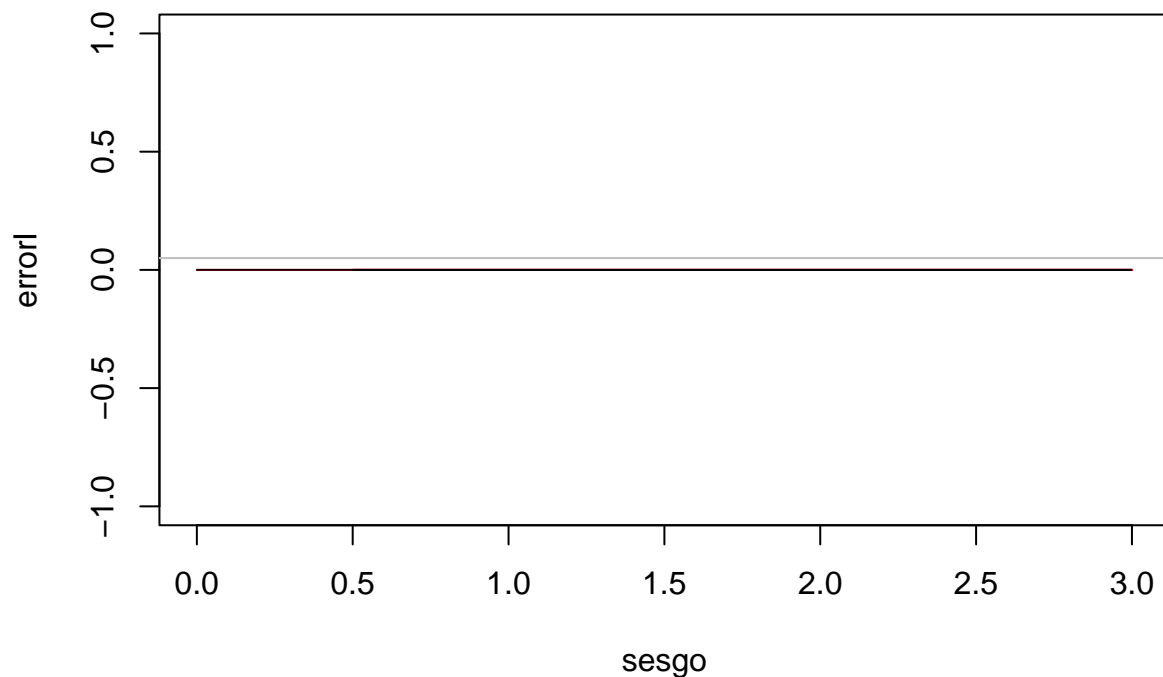
```

```

col="lavender", border=NA)
polygon(c(sesgo, rev(sesgo)), c(lim_inf, rev(errorI)),
col="lavender", border=NA)

lines(sesgo, errorI, type="l", lwd=1, col="red")
lines(sesgo, errorI, type="l", lwd=0.05, lty=2)
lines(sesgo, errorI, type="l", lwd=0.05, lty=2)
abline(0.05,0,col="grey")

```



Errores Tipo II

La evaluación de los errores tipo II (β) es más compleja, ya que dependen de tres elementos:

- El alfa teórico
- El tamaño del efecto
- El error típico de medida

Para la evaluación de β , o lo que es equivalente, la potencia, necesitamos dejar todo lo constantes que podamos esos tres elementos. Veamos cómo podemos estimar (recordamos que es una simplificación) una β empírica (β_{emp}).

En un contraste de una media, el tamaño del efecto se calcula simplemente como:

$$d = \frac{\mu_0 - media}{S_x}$$

Así que podemos fijar a priori un μ_0 , por ejemplo $\mu_0 = 0$ y generar k muestras que provengan de una distribución $N(\mu, \sigma)$ donde la media es diferente a μ_0 , por ejemplo $N(3, 2)$. Posteriormente, con cada una de las k muestras generadas podemos poner a prueba la $H_0 : \mu = \mu_0$ y, encontrar la beta empírica simplemente contando las veces que hemos aceptado la H_0 . Para calcular la beta teórica podemos usar la librería `pwr` y seguir sus instrucciones. Por ejemplo, en nuestros datos para conocer la potencia teórica usaríamos la función `pwr.t.test()`. Como la función nos da el valor de la potencia para saber el valor de beta simplemente: $\beta = 1 - potencia$.

Tarea 4

Realiza simulaciones para comprobar si mantienen iguales ambos betas (teórica dada por `pwr` y empírica) cuando partimos de variables de rango limitado (y por tanto no normales). Crea un gráfico en el que se pueda ver si cambia la diferencia entre las betas para un rango de valores 2,3, 4 y 5.

```
set.seed(4)
```

```
library(pwr)
```

```
## Warning: package 'pwr' was built under R version 4.3.3
```

```
#Se elige un valor cualquiera de mu pero que genere valores razonables de potencia
#Si esta muy proximo a la media poblacional la potencia será 0.
#Si esta muy lejos será 1 y no nos interesan esos caso
#Una estrategia es hacer que mu=mean(poblacion)+0.8*sd(poblacion)
```

```
expo3 <- function(N,n,k,range_limit){

  poblacion <- sample(1:2,N,replace=T)
  mu_pob <- mean(poblacion)
  sd_pob <- sd(poblacion)

  p <- c()

  for (i in 1:k){

    muestra <- poblacion[sample(1:N,n)]

    p[i] <- t.test(muestra,mu=mu_pob+0.8*sd_pob)$p.value
  }

  beta_emp <- length(p[p>a_teo])/k

  d=(10-(mu_pob+0.8*sd_pob))/sd_pob
  beta_teo <- 1-pwr.t.test(n,
                           d=d,
                           sig.level = a_teo,
                           type="one.sample")$power

  return(beta_teo-beta_emp)
```

```

}

grafico <- function(longitud){

  rep <- 500
  res <- c()
  datos <- c()
  desv <- c()
  lim_sup <- c()
  lim_inf <- c()

  for (i in 1:longitud){
    for (j in 1:rep){
      res[j] <- expo3(1000,25,500,i)
    }
    datos[i] <- mean(res)
    desv[i] <- sd(res)
    lim_sup[i] <- datos[i]+1.96*(desv[i]/sqrt(rep))
    lim_inf[i] <- datos[i]-1.96*(desv[i]/sqrt(rep))

  }

  datos <- datos[-1]
  desv <- desv[-1]
  lim_sup <- lim_sup[-1]
  lim_inf <- lim_inf[-1]

  rangos <- seq(2,longitud,1)

  plot(rangos,datos,type="l")
  polygon(c(rangos,rev(rangos)),c(lim_sup,rev(datos)),
    col="lavender", border=NA)
  polygon(c(rangos,rev(rangos)),c(lim_inf,rev(datos)),
    col="lavender",border=NA)

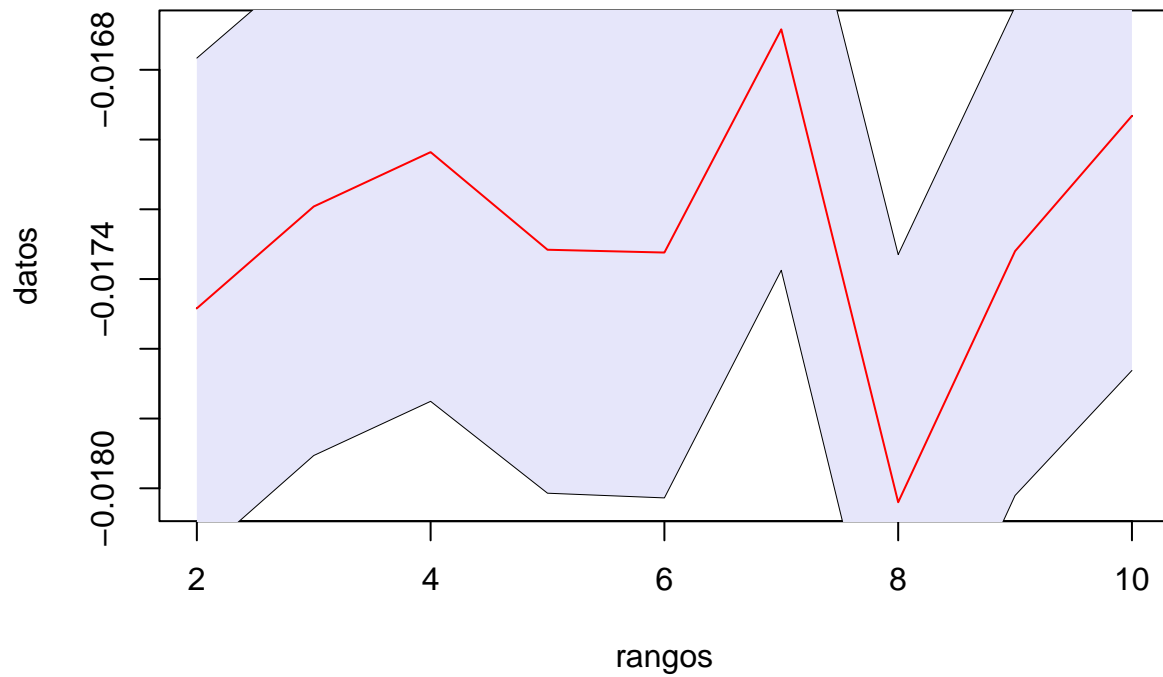
  lines(rangos,datos,type="l", lwd=1,col="red")
  lines(rangos,lim_sup,type="l",lwd=0.5)
  lines(rangos,lim_inf,type="l",lwd=0.5)

}

rango <- 10

grafico(rango)

```



Tarea 5

Realiza simulaciones para comprobar cómo es la diferencia entre las dos betas al aumentar el tamaño de la muestra ($n=2,10,15,20$). Haz un gráfico.

```
set.seed(4)

a_teo <- 0.05

expo4 <- function(N,n,k){

  poblacion <- rnorm(N,10,10)
  mu_pob <- mean(poblacion)
  sd_pob <- sd(poblacion)
  p <- c()

  for (i in 1:k){

    muestra <- poblacion[sample(1:N,n)]
    p[i] <- t.test(muestra,mu=mu_pob+0.8*sd_pob)$p.value

  }

  beta_emp <- length(p[p>a_teo])/k
```

```

d=(10-(mu_pob+0.8*sd_pob))/sd_pob
beta_teo <- 1-pwr.t.test(n,
                        d=d,
                        sig.level = a_teo,
                        type="one.sample")$power

return(beta_teo-beta_emp)
}

rep <- 250
tamaños <- c(2,10,15,20)
res <- c()
res_sd <- c()
lim_sup <- c()
lim_inf <- c()

for (i in 1:length(tamaños)){

  res_in <- c()

  for(j in 1:rep){
    res_in[j] <- expo4(1000,tamaños[i],250)
  }

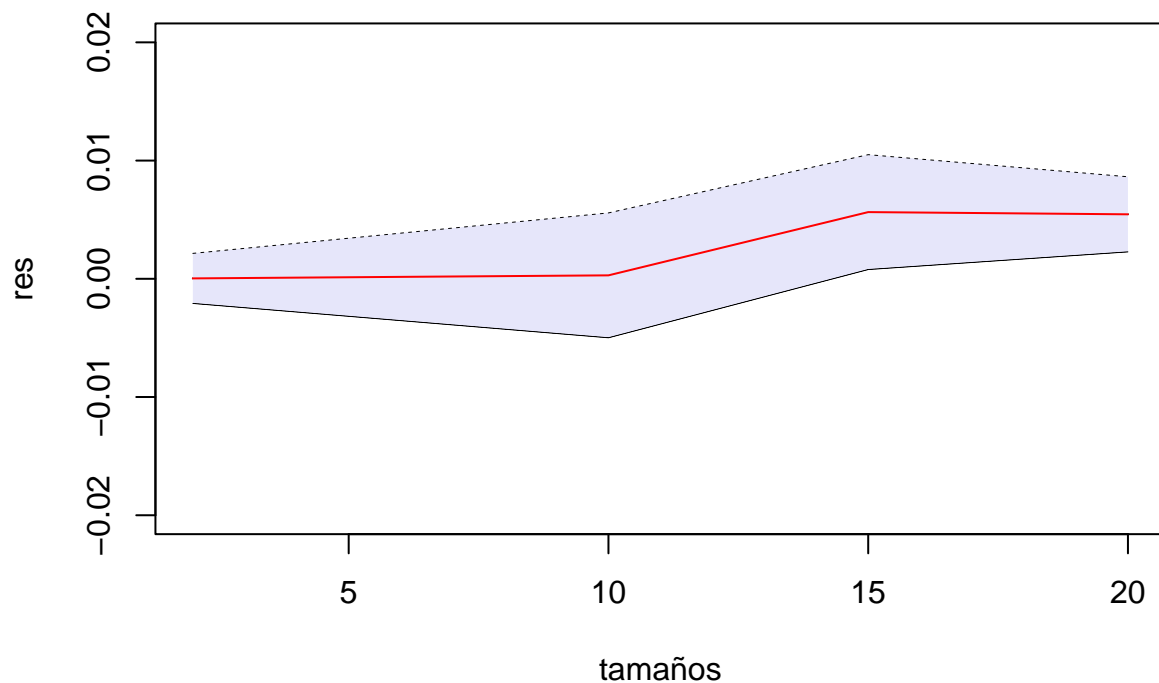
  res[i] <- mean(res_in)
  res_sd[i] <- sd(res_in)
  lim_sup[i] <- res[i]+1.96*(res_sd[i]/sqrt(rep))
  lim_inf [i] <- res[i]-1.96*(res_sd[i]/sqrt(rep))

}

plot(tamaños,res, ylim=c(-0.02,0.02), type="l")
polygon(c(tamaños,rev(tamaños)),c(lim_sup,rev(res)),
       col="lavender",border=NA)
polygon(c(tamaños,rev(tamaños)), c(lim_inf,rev(res)),
       col="lavender",border=NA)

lines(tamaños,res,type="l",lwd=1,col="red")
lines(tamaños,lim_sup,type="l",lwd=0.5,lty=2)
lines(tamaños,lim_inf,type="l",lwd=0.05,lty=2)

```



Tarea 6

Utilizando la distribución de Johnson estudia como afecta al contraste de una media que la distribución se vaya haciendo más y más asimétrica (utiliza dos o tres niveles de asimetría).

```
set.seed(4)

expto12 <-function(N,n,k,s){
  parms<-JohnsonFit(c(0, 2, s, 20), moment="use")
  sJohnson(parms)
  poblacion <- rJohnson(N, parms)
  mu.pob <- mean(poblacion)
  sd.pob <- sd(poblacion)
  p <- vector(length=k)
  for (i in 1:k){
    muestra <- poblacion[sample(1:N, n)]
    p[i] <- t.test(muestra, mu=mu.pob+0.8*sd.pob)$p.value
  }
  a_teo = 0.05
  # calculamos beta empirica
  beta_emp <- length(p[p>a_teo])/k
  # calculamos beta teorica
  d=(10 - mu.pob+0.8*sd.pob) / sd.pob
  beta_teo <- 1 - pwr.t.test(n,
                             d=d,
```

```

                                sig.level=a_teo,
                                type="one.sample")$power
  return (beta_teo-beta_emp)
}

#Exploramos en diferentes niveles de sesgo
Sesgo = seq(0,3,0.5)
ErrorII=vector(length=length(seq(0,3,0.5)))
set.seed(1)
c=1
res_sd <- c()
lim_sup <- c()
lim_inf <- c()
for (j in seq(0,3,0.5)) {
  rep <- 500
  res <- vector(length=rep)
  for (i in 1:rep) {
    res[i] <- expto12(1000, 25, 500,j)
  }
  # hist(res)
  ErrorII[c]=mean (res)
  res_sd[c] = sd(res)
  lim_sup[c] = ErrorII [c] + 1.95 * (res_sd[c]/sqrt(rep))
  lim_inf[c] = ErrorII [c] - 1.95 * (res_sd[c]/sqrt(rep))
  c =c+1
  sd(res)
}

plot(Sesgo, ErrorII, type = "l")
polygon(c(Sesgo, rev(Sesgo)), c(lim_sup, rev(ErrorII)),
        col = "lavender", border = NA)
polygon(c(Sesgo, rev(Sesgo)), c(lim_inf, rev(ErrorII)),
        col = "lavender", border = NA)
lines(Sesgo,ErrorII, type = "l", lwd = 1, col = "red")
lines(Sesgo, lim_sup, type = "l", lwd = 0.5, lty = 2)
lines(Sesgo, lim_inf, type = "l", lwd = 0.5, lty = 2)
abline(0.05, 0, col = "grey")

```

