Parcial-2023 Sofia Gerard y Róman Vélez

Entrega: 10 de octubre antes de las 16:00 horas, por correo electrónico con el título fundamentos-parcial, un solo documento (pdf/html) por equipo.

Instrucciones:

- Tus respuestas deben ser claras y debes explicar los resultados, incluye también tus procedimientos/código de manera ordenada, y el código comentado.
- Se evaluará la presentación de resultados (calidad de las gráficas, tablas, ...), revisa la sección de visualización en las notas.
- Se puede realizar individual o en parejas.
- Si tienes preguntas puedes escribirlas en el anuncio de canvas del examen.

Pruebas de hipótesis

Nos solicitan hacer un análisis con el objetivo de probar un material nuevo para suela de zapatos (el material B) y ver si es comparable con el material que se usa normalmente (el material A).

Nos dan el siguiente conjunto de datos:

```
library(readr)
zapatos <- read_csv("datos/zapatos-1.csv")
glimpse(zapatos)
```

```
## Rows: 20
## Columns: 2
## $ desgaste <dbl> 13.2, 14.0, 8.2, 8.8, 10.9, 11.2, 14.3, 14.2, 10.7, 11.8, 6.6~
## $ material <dbl> 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2
```

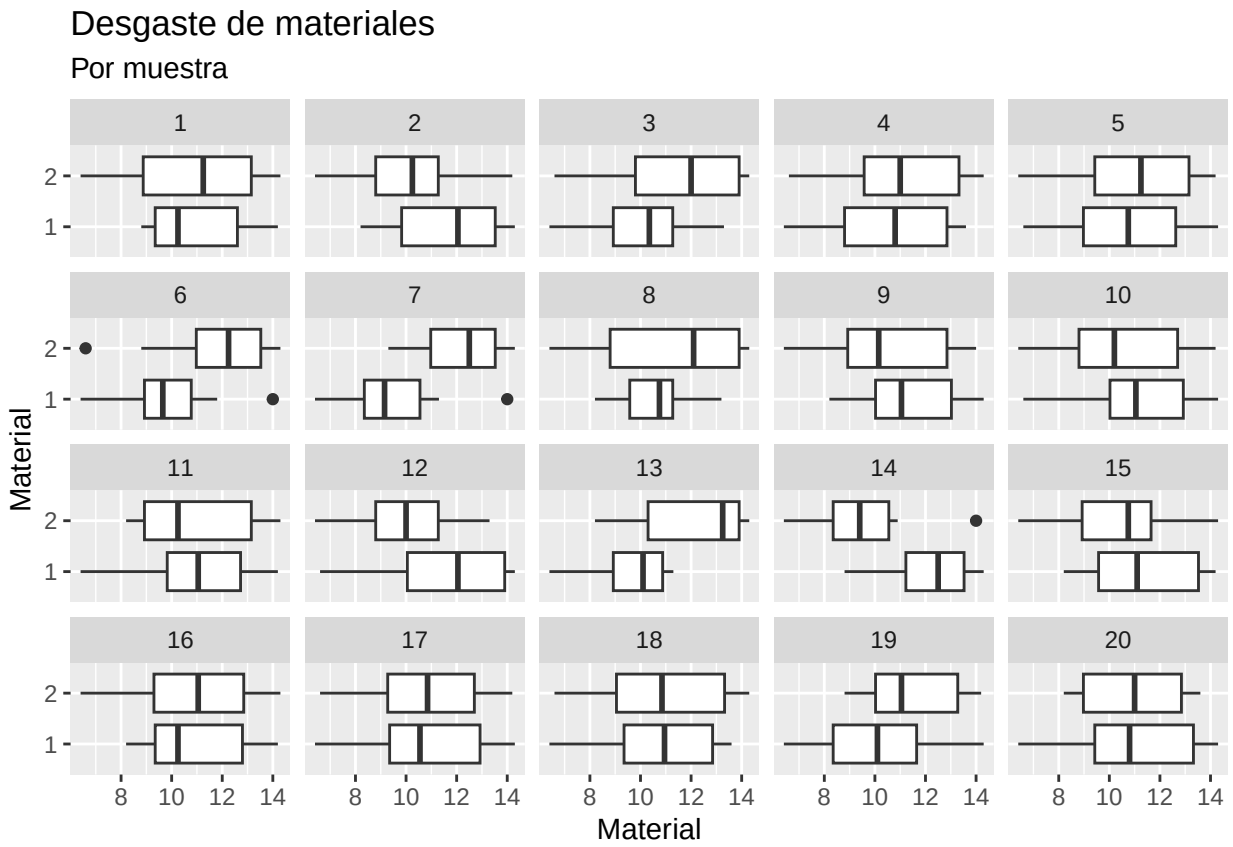
1. Realiza una prueba de hipótesis visual y describe tus conclusiones (cuál es el nivel de significancia de la prueba?).

```
library(nullabor)
perms_materiales<- lineup(null_permute("material"), zapatos, n = 20)
glimpse(perms_materiales)
```

```
## Rows: 400
## Columns: 3
## $ desgaste <dbl> 13.2, 14.0, 8.2, 8.8, 10.9, 11.2, 14.3, 14.2, 10.7, 11.8, 6.6~
## $ material <dbl> 1, 2, 2, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1~
## $ .sample  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2~
```

```
graf_materiales <- ggplot(perms_materiales, aes(x= desgaste, y= factor(material))) +
  geom_boxplot(aes(group = material)) +
  facet_wrap(~.sample) +
  labs(
    x = "Material",
    y = "Material",
    title = "Desgaste de materiales",
    subtitle = "Por muestra" )
```

```
graf_materiales
```




```
decrypt("clZx bKhK oL 3OHohoOL BQ")
```

```
## [1] "True data in position  5"
```

```
# Ho = No hay diferencia en el desgaste entre el material A y B.
# H1 = Existe una diferencia en el desgaste entre el material A y B.
# Dado que visualmente no podemos reconocer la gráfica de los datos originales
#( es la número 5), no hay suficiente evidencia para rechazar Ho,
#con un nivel de significancia alpha = 1/20.
```

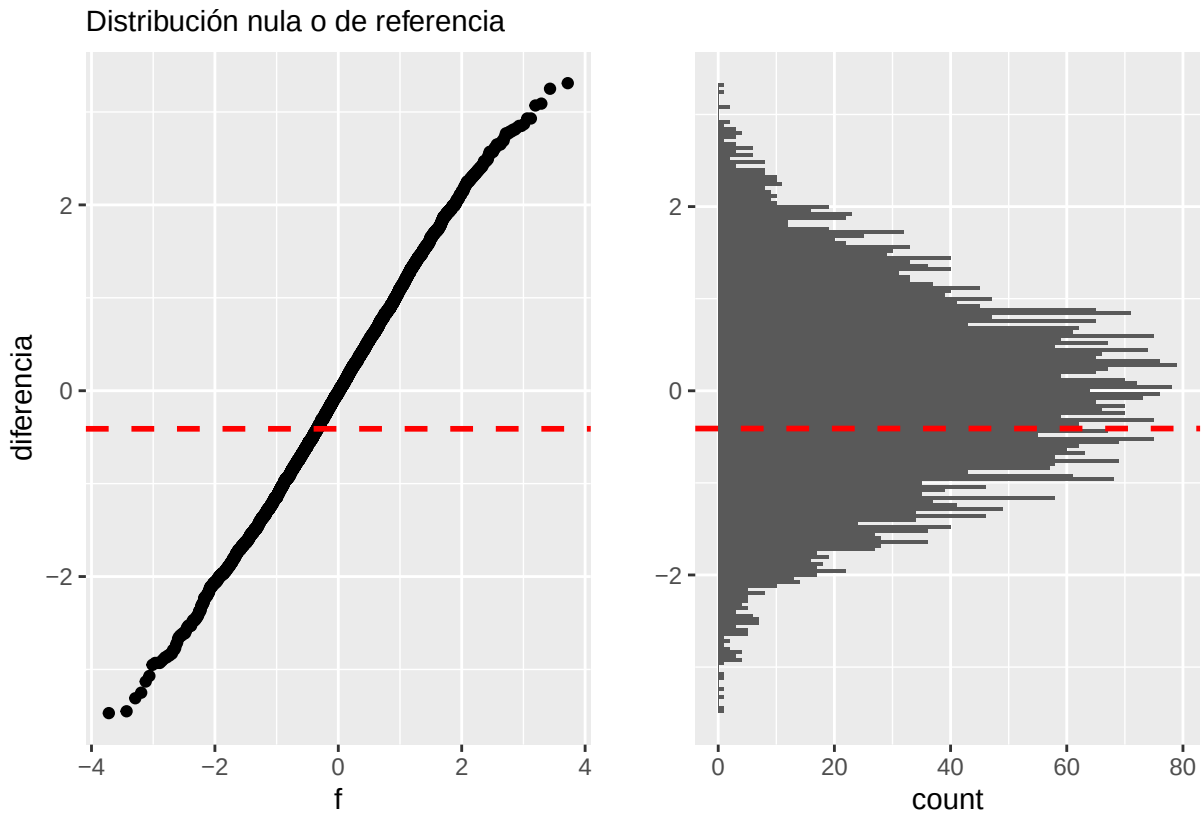
2. Realiza una prueba de permutaciones para la diferencia de las medias, escribe la hipótesis nula, la hipótesis alterna y tus conclusiones.

```
# Ho = No hay diferencia en la diferencia de medias de los materiales A y B.
# Es decir que x_bar_a - x_bar_b = 0

# H1 = Existe una diferencia en la diferencia de medias de los materiales A y B.


# Promedio de desgaste de materiales muestra:

prom_desgaste <- zapatos |>
  group_by(material) %>%
  summarise(prom=mean(desgaste)) |>
  ungroup() %>%
  pivot_wider(names_from = material, values_from = prom) |>
```

```
  rename(A = `1`, B = `2`) |>
  mutate(diferencia = A - B )

prom_desgaste
```

```
## # A tibble: 1 x 3
##       A     B diferencia
##   <dbl> <dbl>      <dbl>
## 1  10.6  11.0      -0.41
```

```
# Repeticiones con permutación

reps_materiales <- lineup(null_permute("material"), zapatos, n = 5000)

valores_ref <- reps_materiales |>
  group_by(.sample, material) |>
  summarise(media = mean(desgaste), .groups = 'drop') |>
  pivot_wider(names_from = material, values_from =media) |>
  rename(A = `1`, B = `2`) |>
  mutate(diferencia = A - B )

g_1 <- ggplot(valores_ref, aes(sample = diferencia)) + geom_qq()  +
  geom_hline(yintercept = -0.41, color = "red", linetype = "dashed", linewidth = 1) +
  xlab("f") + ylab("diferencia") + labs(subtitle = "Distribución nula o de referencia")

g_2 <- ggplot(valores_ref, aes(x = diferencia)) +
  geom_histogram(binwidth = 0.04) +
  geom_vline(xintercept = -0.41, color = "red", linetype = "dashed", linewidth = 1)+
  coord_flip() + xlab("") + labs(subtitle = " ")

g_1 + g_2
```




```
#Nuestra estadística de prueba para la diferencia de medias es T(X) = -0.41, la cual no resulta extrema

dist_ref <- ecdf(valores_ref$diferencia)

valor_p <- 2 * min(dist_ref(prom_desgaste$diferencia),
                   (1 - dist_ref(prom_desgaste$diferencia)))
valor_p
```

```
## [1] 0.7348
```

```
#La probabilidad de obtener un resultado igual o más extremo que el observado (-0.41), asumiendo que la
```

3. Después de discutir con los responsables del proyecto descubrimos que nos faltaba conocer detalles del proceso generador de datos: el experimento se realizó asignando al azar un material a uno de sus zapatos y el otro material al otro zapato de cada niño. ¿Cómo incorporas esta información en tu prueba de hipótesis del inciso 2? ¿Cambian tus conclusiones?

```
zapatos2 <- read_csv("datos/zapatos-2.csv")
glimpse(zapatos2)
```

```
## Rows: 20
## Columns: 3
## $ desgaste <dbl> 13.2, 14.0, 8.2, 8.8, 10.9, 11.2, 14.3, 14.2, 10.7, 11.8, 6.6~
## $ material <dbl> 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2
## $ niño     <dbl> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10
```

```
# Al cambiar el proceso generador de datos cambia la manera de hacer la prueba
#de hipótesis. Se convierte en un caso de pruebas pareadas donde tenemos que
#verificar para cada niño la diferencia de medias.
```

```r
##NO ME SALE ESTA PARTE porque no me agarra el df correcto sos

pareadas <- zapatos2 |>
  group_split(niño) |>
  map_df(~lineup(null_permute("material"), n = 5000)) |>
  group_by(.sample, material) |>
  summarise(mean_desgaste = mean(desgaste), .groups = 'drop') |>
  pivot_wider(names_from = material, values_from =mean_desgaste) |>
  rename(A = `1`, B = `2`) |>
  mutate(diferencia = A - B ) |>
  summarise(diff_means = diff(mean_desgaste))
```

```
## Error in `group_by()`:
## ! Must group by variables found in `.data`.
## x Column `material` is not found.
```

Bootstrap

Antecedentes En México, las elecciones tienen lugar un domingo, los resultados oficiales del proceso se presentan a la población una semana después. A fin de evitar proclamaciones de victoria injustificadas durante ese periodo el INE organiza un conteo rápido. Un conteo rápido es un procedimiento para estimar, a partir de una muestra aleatoria de casillas, el porcentaje de votos a favor de cada opción en la boleta.

En 2021 se realizó un conteo rápido para estimar los resultados de la consulta popular 2021 y en los siguientes incisos estimarán los resultados de la consulta y evaluarán la metodología.

Diseño de la muestra El diseño utilizado en los conteos rápidos es *muestreo estratificado simple*, es decir:

i) se particionan las casillas de la pablación en estratos (cada casilla pertenece a exactamente un estrato), y

ii) dentro de cada estrato se usa *muestreo aleatorio* para seleccionar las casillas que estarán en la muestra.

Estimación Una de las metodolgías de estimación, que se usa en el conteo rápido (tanto de elecciones como en consultas) es *estimador de razón combinado*, con intervalos de 95% de confianza construidos con el método normal y error estándar bootstrap. En este ejercicio debes construir intervalos usando este procedimiento.

Para cada opción en la consulta (sí/no/nulos) usarás la muestra del conteo rápido para estimar los resultados de la consulta.

1. Calcula el estimador de razón combinado, para muestreo estratificado la fórmula es:

$$\hat{p} = \frac{\sum_h \frac{N_h}{n_h} \sum_i Y_{hi}}{\sum_h \frac{N_h}{n_h} \sum_i X_{hi}}$$

donde:

- $\hat{p}$ es la estimación de la proporción de votos que recibió la opción (ej: *sí*).
- Y_{hi} es el número total de votos que recibió *la opción* (ej: *sí*) en la i-ésima casillas, que pertence al h-ésimo estrato.
- X_{hi} es el número total de votos en la i-ésima casilla, que pertence al h-ésimo estrato.
- N_h es el número total de casillas en el h-ésimo estrato.
- n_h es el número de casillas del h-ésimo estrato que se seleccionaron en la muestra.

Datos Necesitarás los siguientes datos:

- Cómputos aquí
- Muestra del conteo rápido usada en la estimación aquí

```
# preprocesamiento de tablas de datos

computos <- read_delim("datos/20210802-2130_INE-CONSULTA-POPULAR-2021/20210802-2130_COMPUTOS-INE-CP2021
    delim = "|", escape_double = FALSE, trim_ws = TRUE, quote = "\'",
    skip = 5)
computos <- computos |>
  rename(ID = CLAVE_MRCP) |>
  mutate(ESTRATO = str_c(str_pad(ID_ENTIDAD, 2, pad = "0"),
                         str_pad(ID_DISTRITO_FEDERAL, 2, pad = "0")),
         LISTA_NOMINAL = LISTA_NOMINAL_MRCP,
         TOTAL = TOTAL_OPINIONES)

muestra <- read_delim("https://ine.mx/wp-content/uploads/2021/08/Conteos-ConsPop21-Lista-MuestraCalculo
muestra_tidy <- muestra |>
  mutate(
    ID_ESTADO = str_pad(ID_ESTADO, 2, pad = "0"),
    SECCION = str_pad(SECCION, 4, pad = "0"),
    ID_CASILLA = str_pad(ID_CASILLA, 2, pad = "0"),
    ID = str_c(ID_ESTADO, SECCION, TIPO_CASILLA, ID_CASILLA)
    ) |>
  group_by(ESTRATO) |>
  mutate(n = n()) |>
  ungroup()
```

```
# Primera parte

colnames(muestra_tidy)
```

```
##  [1] "ID_ESTADO"           "ID_DISTRITO_FEDERAL" "SECCION"
##  [4] "TIPO_CASILLA"        "ID_CASILLA"          "EXT_CONTIGUA"
##  [7] "TIPO_SECCION"        "LISTA_NOMINAL"       "ID_MUNICIPIO"
## [10] "ID_DIST_LOC"         "ID_ESTRATO"          "ESTRATO"
## [13] "NUMERO_ARE"          "SI"                  "NO"
## [16] "NULOS"               "TOTAL"               "ANIO"
## [19] "MES"                 "DIA"                 "HORA"
## [22] "MINUTOS"             "SEGUNDOS"            "ORIGEN_CAPTURA"
## [25] "MODIFICADO"          "ID"                  "n"
```

```
colnames(computos)
```

```
##  [1] "ID"                  "ID_ENTIDAD"          "ENTIDAD"
##  [4] "ID_DISTRITO_FEDERAL" "DISTRITO_FEDERAL"    "SECCION_SEDE"
##  [7] "TIPO_MRCP"           "ID_MRCP"             "OPINION_SI"
## [10] "OPINION_NO"          "NULOS"               "TOTAL_OPINIONES"
## [13] "LISTA_NOMINAL_MRCP"  "OBSERVACIONES"       "RECUENTO_TOTAL"
## [16] "FECHA_HORA"          "ESTRATO"             "LISTA_NOMINAL"
## [19] "TOTAL"
```

```
### Numero total de casillas por estrato N y n casillas seleccionadas para la muestra por estrato:
total_casillas_computos <- computos |>
  group_by(ESTRATO) |>
```

```r
  summarise(numero_casillas = n()) |>
  left_join(muestra_tidy|> select(ESTRATO,n) |> unique())

## Factor de expansión N/n para cada estrato
total_casillas_computos <- total_casillas_computos |>
  mutate(factor_expansion = numero_casillas/n)


votos_muestra <- muestra_tidy |>
  select(ESTRATO, SI, NO, NULOS, TOTAL, ID )

### Dividimos por estratos

grupos <- votos_muestra |> group_split(ESTRATO, .keep = TRUE)

### Calculamos el total de SI NO y NULOS por estrato en la muestra:

resultados_si <- map(grupos, function(grupo) {
  suma_si <- sum(grupo$SI)
  estrato <- unique(grupo$ESTRATO)
  return(data.frame(Estrato = estrato, Suma_SI = suma_si))
})

resultados_no <- map(grupos, function(grupo) {
  suma_no <- sum(grupo$NO)
  estrato <- unique(grupo$ESTRATO)
  return(data.frame(Estrato = estrato, suma_NO = suma_no))
})

resultados_nulos <- map(grupos, function(grupo) {
  suma_nulos <- sum(grupo$NULOS)
  estrato <- unique(grupo$ESTRATO)
  return(data.frame(Estrato = estrato, suma_NULOS = suma_nulos))
})


### Calculamos los votos totales por estrato en la muestra:

resultados_totales <- map(grupos, function(grupo) {
  suma_totales <- sum(grupo$TOTAL)
  estrato <- unique(grupo$ESTRATO)
  return(data.frame(Estrato = estrato, suma_TOTALES = suma_totales))
})


### Hago un df con toda la información que necesitamos, "tabla_joins"

todos_resultados <- bind_rows(resultados_si, .id = "Fuente") %>%
  inner_join(bind_rows(resultados_no, .id = "Fuente"), by = c("Estrato", "Fuente")) %>%
  inner_join(bind_rows(resultados_nulos, .id = "Fuente"), by = c("Estrato", "Fuente")) %>%
  inner_join(bind_rows(resultados_totales, .id = "Fuente"), by = c("Estrato", "Fuente"))

todos_resultados <- todos_resultados |>
```

```
   rename(ESTRATO = Estrato)

tabla_joins <- inner_join(total_casillas_computos, todos_resultados, by = "ESTRATO")


## Vamos a calcular el estimador de razón combinado para muestreo estratificado p gorro para SI, NO y N

tabla_ponderada_si <- tabla_joins |>
  select(ESTRATO, factor_expansion, Suma_SI, suma_TOTALES) |>
  mutate(Suma_si_confactor = Suma_SI*factor_expansion, Suma_total_confactor = suma_TOTALES*factor_expans

p_gorro_si <- sum(tabla_ponderada_si$Suma_si_confactor)/sum(tabla_ponderada_si$Suma_total_confactor)


tabla_ponderada_no <- tabla_joins |>
  select(ESTRATO, factor_expansion, suma_NO, suma_TOTALES) |>
  mutate(Suma_no_confactor = suma_NO*factor_expansion, Suma_total_confactor = suma_TOTALES*factor_expans

p_gorro_no <- sum(tabla_ponderada_no$Suma_no_confactor)/sum(tabla_ponderada_no$Suma_total_confactor)

tabla_ponderada_nulos <- tabla_joins |>
  select(ESTRATO, factor_expansion, suma_NULOS, suma_TOTALES) |>
  mutate(Suma_nulos_confactor = suma_NULOS*factor_expansion, Suma_total_confactor = suma_TOTALES*factor

p_gorro_nulos <- sum(tabla_ponderada_nulos$Suma_nulos_confactor)/sum(tabla_ponderada_nulos$Suma_total_c

cat("El estimador de razón combinado para muestreo estratificado para SI es:", p_gorro_si)
```

```
## El estimador de razón combinado para muestreo estratificado para SI es: 0.9295214
```

```
cat("El estimador de razón combinado para muestreo estratificado para NO es:", p_gorro_no)
```

```
## El estimador de razón combinado para muestreo estratificado para NO es: 0.01475835
```

```
cat("El estimador de razón combinado para muestreo estratificado para NULOS es:", p_gorro_nulos)
```

```
## El estimador de razón combinado para muestreo estratificado para NULOS es: 0.05572026
```

2. Utiliza **bootstrap** para calcular el error estándar, y reporta tu estimación del error.
   - Genera 1000 muestras bootstrap.
   - Recuerda que las muestras bootstrap tienen que tomar en cuenta la metodología que se utilizó en la selección de la muestra original, en este caso implica que para cada remuestra debes tomar muestra aleatoria independiente dentro de cada estrato.

```
## Buscamos el error estándar de p gorro $$\hat{p}=\frac{\sum_h \frac{N_h}{n_h} \sum_i Y_{hi}}{\sum_h \
## 300 estratos = 300 muestras independientes
## Bootstrap de Rao y Wu

replicas= numeric(1000)
```

3. Construye un intervalo del 95% de confianza utilizando el método normal. Revisa si el supuesto de normalidad es razonable.
4. Reporta tus intervalos en una tabla. Compara la longitud de los 3 intervalos y describe que observas.
5. ¿Tus intervalos contienen los valores observados en los cómputos? Explica los resultados observados.

Calibración Selecciona al menos 50 muestras del mismo tamaño y con el mismo diseño que la muestra utilizada en el conteo rápido. Esto es, selecciona el mismo número de casillas, usando muestreo aleatorio simple dentro de cada estrato.

- Para cada muestra calcula un intervalo del 95% de confianza usando bootstrap.
- Grafica los intervalos y calcula la proporción de ellos que contienen el verdadero valor observado. Describe tus observaciones y compara con el intervalo obtenido en el ejercicio anterior.

Análisis Exploratorio Un voto nulo corresponde a una boleta donde el ciudadano acudió a las urnas y anuló su voto.

Antes de contestar los siguiente incisos piensen que rango esperarían ver para la proporción de votos nulos en una casilla.

- Describe la distribución de datos nulos en la muestra, y como se relaciona con el total de votos, realiza gráficas y describe tus observaciones.
- En la distribución de proporción de nulos se observan datos atípicos, ¿cuál crees que sea la razón de estas observaciones extremas? ¿consideras que se deben eliminar de la muestra antes de realizar la estimación?