Tarea 3 - Respuestas

Ejercicio 1

En clase analizamos si la **media** y la **mediana** eran estimadores insesgados del promedio de una población **normal**.

En este ejercicio harás lo mismo, pero para una distribución **beta**. En esta (p. 906) y esta otra[1] ligas puedes aprender más sobre esta distribución.

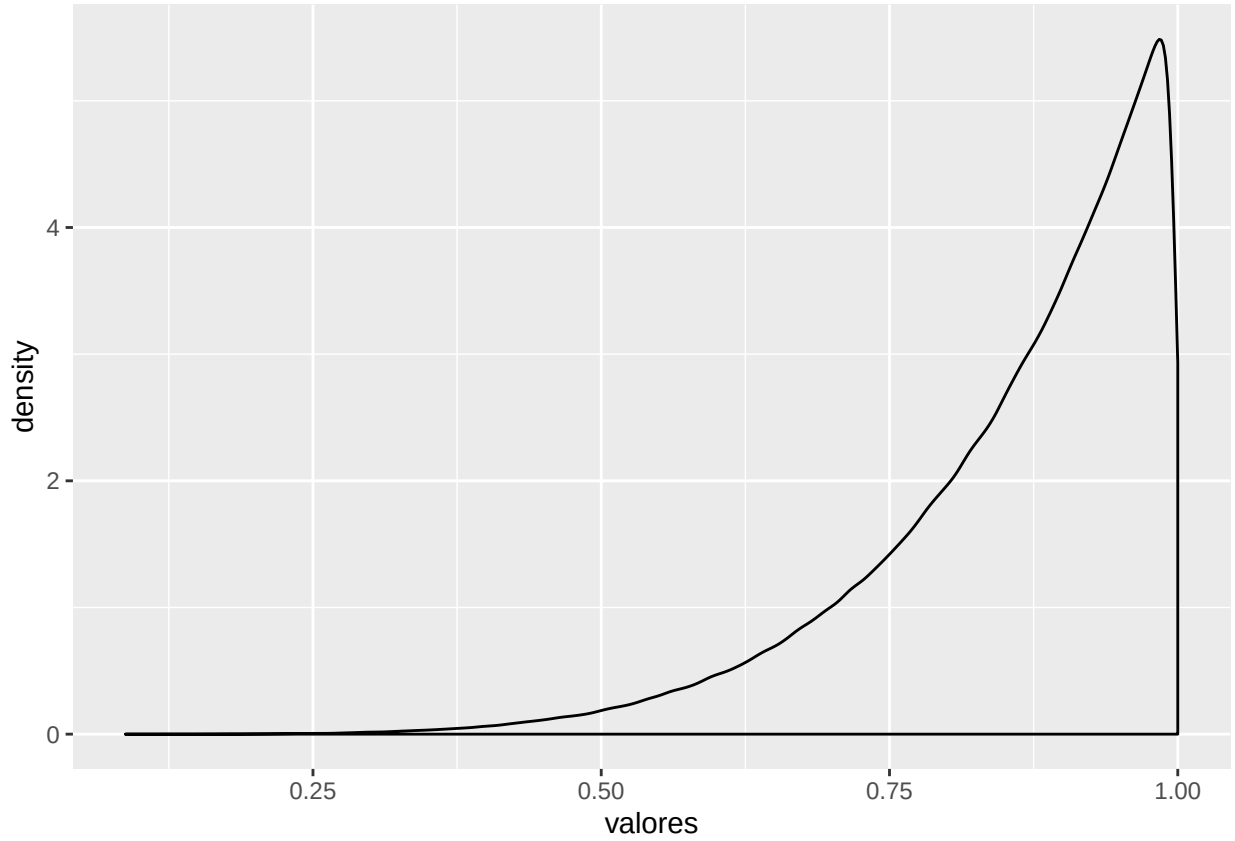
a. Usa el siguiente código para construir la distribución:

```
set.seed(1)
datos <- tibble(valores = rbeta(n = 1e6, shape1 = 6, shape2 = 1))
```

b. Grafica la distribución. ¿Es simétrica? Si la respuesta es no, ¿qué tipo de sesgo tiene?

```
datos %>%
  ggplot(aes(valores)) +
  geom_density()
```

[1] En esta segunda liga basta con que leas los primeros dos párrafos, pero todo el post vale la pena.




```
# Sesgo izquierdo
```

c. Calcula la media y mediana poblacional

```
est_pob <-
  datos %>%
  summarise(media_pob = mean(valores),
            mediana_pob = median(valores))

est_pob
```

```
## # A tibble: 1 x 2
##   media_pob mediana_pob
##       <dbl>       <dbl>
## 1     0.857       0.891
```

Escenario 1

- Planta un `set.seed(1)` y toma **1,000** muestras sin remplazo, cada una de tamaño **5**.
- Calcula la media y mediana de cada muestra. Guarda los cálculos de las medias muestrales en una variable llamada `media_muestral` y los de las medianas en `mediana_muestral`.
- Calcula el promedio de `media_muestral` y `mediana_muestral`. Guarda los resultados en `promedio_medias_muestrales` y `promedio_medianas_muestrales`.

```r
set.seed(1)
datos %>%
  rep_sample_n(size = 5, reps = 1000) %>%
  summarise(media_muestral = mean(valores),
            mediana_muestral = median(valores)) %>%
  summarise(promedio_media_muestral = mean(media_muestral),
            promedio_mediana_muestral = mean(mediana_muestral)) %>%
  gather(key = variable,
         value = valor)
```

```
## # A tibble: 2 x 2
##   variable                  valor
##   <chr>                     <dbl>
## 1 promedio_media_muestral   0.858
## 2 promedio_mediana_muestral 0.882
```

Por favor responde:

i. ¿El promedio de alguno de los estimadores es igual a su respectivo parámetro poblacional?

ii. Si es así, ¿cuál? Si no es así, ¿cuál está más cerca y por cuánto?

OJO: antes de responder estas preguntas, asegúrate de haber corrido primero `set.seed(1)` y después el chunk con tus cálculos. De lo contrario no podremos replicar tus datos :'(.

```r
set.seed(1)
datos %>%
  rep_sample_n(size = 5, reps = 1000) %>%
  summarise(media_muestral = mean(valores),
            mediana_muestral = median(valores)) %>%
  summarise(promedio_media_muestral = mean(media_muestral),
            promedio_mediana_muestral = mean(mediana_muestral)) %>%
  gather(key = variable,
         value = valor) %>%
  mutate(dif_absoluta_media = ifelse(variable == "promedio_media_muestral", abs(est_pob$media_pob - valo
```

```
## # A tibble: 2 x 3
##   variable                  valor dif_absoluta_media
##   <chr>                     <dbl>              <dbl>
## 1 promedio_media_muestral   0.858           0.000476
## 2 promedio_mediana_muestral 0.882           0.00902
```

Escenario 2

Repite los pasos del Escenario 1, pero ahora aumenta el tamaño de cada muestra a **50**.

Por favor responde:

i. ¿El promedio de alguno de los estimadores es igual a su respectivo parámetro poblacional?

ii. Si es así, ¿cuál? Si no es así, ¿cuál está más cerca y por cuánto?

```r
set.seed(1)
datos %>%
  rep_sample_n(size = 50, reps = 1000) %>%
  summarise(media_muestral = mean(valores),
            mediana_muestral = median(valores)) %>%
  summarise(promedio_media_muestral = mean(media_muestral),
            promedio_mediana_muestral = mean(mediana_muestral)) %>%
  gather(key = variable,
         value = valor) %>%
  mutate(dif_absoluta_media = ifelse(variable == "promedio_media_muestral", abs(est_pob$media_pob - val
```

```
## # A tibble: 2 x 3
##   variable                  valor dif_absoluta_media
##   <chr>                     <dbl>              <dbl>
## 1 promedio_media_muestral   0.857           0.000137
## 2 promedio_mediana_muestral 0.890           0.000569
```

Escenario 3

Repite los pasos del Escenario 1, pero ahora aumenta el tamaño de cada muestra a **100**.

Por favor responde:

i. ¿El promedio de alguno de los estimadores es igual a su respectivo parámetro poblacional?

ii. Si es así, ¿cuál? Si no es así, ¿cuál está más cerca y por cuánto?

```r
set.seed(1)
datos %>%
  rep_sample_n(size = 100, reps = 1000) %>%
  summarise(media_muestral = mean(valores),
            mediana_muestral = median(valores)) %>%
  summarise(promedio_media_muestral = mean(media_muestral),
            promedio_mediana_muestral = mean(mediana_muestral)) %>%
  gather(key = variable,
         value = valor) %>%
  mutate(dif_absoluta_media = ifelse(variable == "promedio_media_muestral", abs(est_pob$media_pob - val
```

```
## # A tibble: 2 x 3
##   variable                  valor dif_absoluta_media
##   <chr>                     <dbl>              <dbl>
## 1 promedio_media_muestral   0.857          0.0000185
## 2 promedio_mediana_muestral 0.890          0.000764
```

Escenario 4

Repite los pasos del Escenario 1, pero ahora cambia el **número de muestras** de **1,000** a **10,000**. El tamaño de cada una seguirá siendo de **5**.

Por favor responde:

i. ¿El promedio de alguno de los estimadores es igual a su respectivo parámetro poblacional?

ii. Si es así, ¿cuál? Si no es así, ¿cuál está más cerca y por cuánto?

```
set.seed(1)
datos %>%
  rep_sample_n(size = 5, reps = 10000) %>%
  summarise(media_muestral = mean(valores),
            mediana_muestral = median(valores)) %>%
  summarise(promedio_media_muestral = mean(media_muestral),
            promedio_mediana_muestral = mean(mediana_muestral)) %>%
  gather(key = variable,
         value = valor) %>%
  mutate(dif_absoluta_media = ifelse(variable == "promedio_media_muestral", abs(est_pob$media_pob - val
```

```
## # A tibble: 2 x 3
##   variable                  valor dif_absoluta_media
##   <chr>                     <dbl>              <dbl>
## 1 promedio_media_muestral   0.857           0.000141
## 2 promedio_mediana_muestral 0.880           0.0108
```

Reflexiones generales

Por favor piensa y responde las siguientes preguntas:

- ¿Qué pasa con el estimador de la media y la mediana conforme aumenta el **tamaño de cada muestra**? (comparación de escenarios 2 y 3 vs. 1)
- ¿Qué pasa con el estimador de la media y la mediana conforme aumenta el **número muestras**? (comparación de escenarios 4 vs. 1)

Ejercicio 2

A continuación verás el código que utilicé en clase para ilustrar qué significa el nivel de confianza de un intervalo de confianza. Por favor cópialo y ejecútalo en el script de tu tarea:

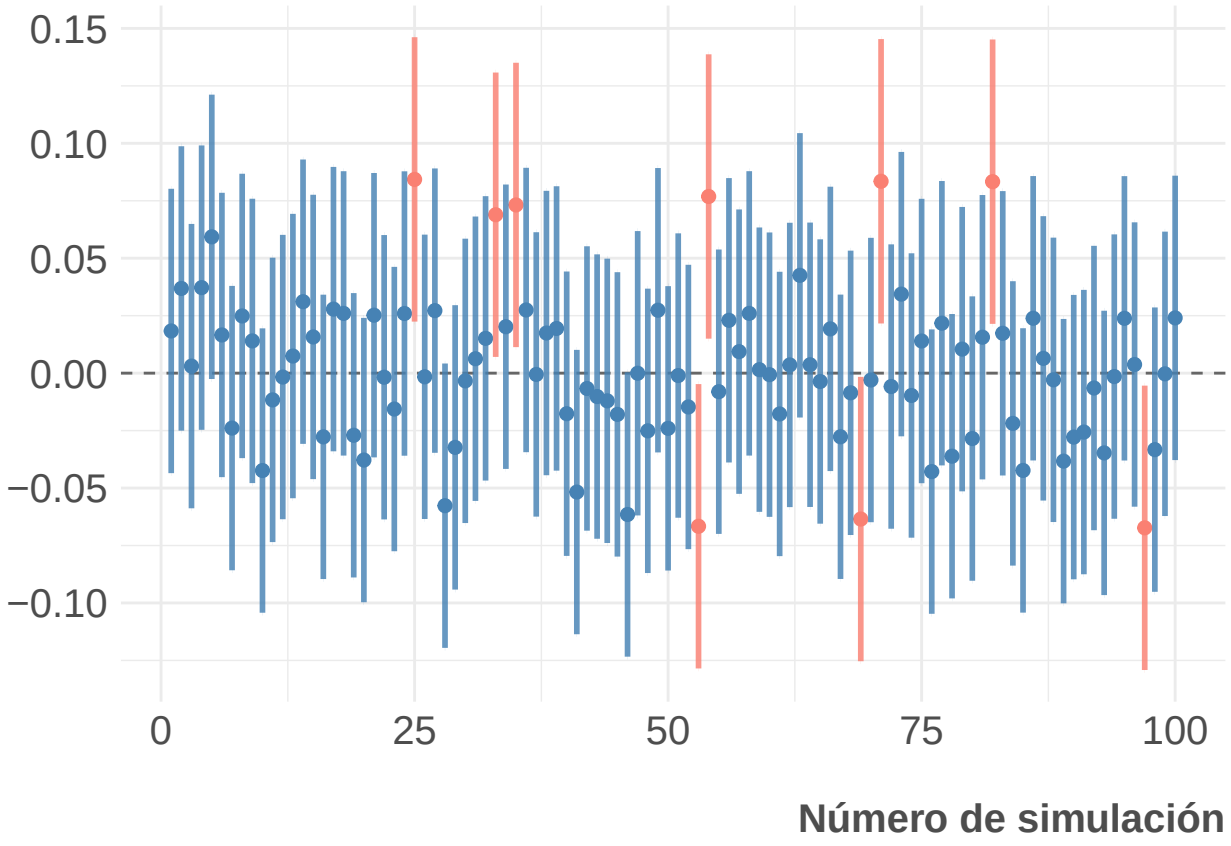
```
# Paso 1 - Generar los datos poblacionales
set.seed(33)
datos <-
  tibble(valores = rnorm(n = 1e6, mean = 0, sd = 1))

# Paso 2 - Definir el tamaño de cada muestra y el número de muestras
tamaño_muestra <- 1000
numero_muestras <- 100

# Paso 3 - Calcular el error estándar de la media muestral, asumiendo
# que el tamaño de la muestra es 1,000
error_est_con_datos_pob <- sd(datos$valores)/sqrt(tamaño_muestra)

# Paso 4 - Calcular la media de cada muestra y construye su respectivo
# intervalo de confianza usando el error estándar que calculamos
# en el paso 3
set.seed(4)
```

```r
datos %>%
  rep_sample_n(size = tamaño_muestra,
               replace = T,
               reps = numero_muestras) %>%
  summarise(media_muestral = mean(valores),
            lim_sup = media_muestral + 1.96*error_est_con_datos_pob,
            lim_inf = media_muestral - 1.96*error_est_con_datos_pob) %>%
  mutate(color_intervalos = ifelse(lim_sup < 0 | lim_inf > 0,
                                   "salmon", "steelblue")) %>%
  ggplot(aes(x = replicate, y = media_muestral, color = color_intervalos)) +
  geom_hline(yintercept = 0, color = "grey40", linetype = 2, size = 0.5) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymin = lim_inf, ymax = lim_sup), width = 0.01,
                alpha = 0.82, size = 1) +
  scale_color_manual(values = c("salmon", "steelblue")) +
  labs(x = "\nNúmero de simulación") +
  theme_minimal() +
  theme(axis.title.x = element_text(hjust = 1, face = "bold",
                                    color = "grey30", size = 15),
        axis.title.y = element_blank(),
        axis.text = element_text(size = 15),
        legend.position = "none")
```



1. Por favor explica qué es lo que estoy haciendo en cada paso. Para el paso 3 basta con que expliques hasta el renglón en donde uso `scale_color_manual()`

2. Asume que el viejo Segasi no sabe nada de estadística. Por favor explícale de forma comprensible qué **SÍ** es y que **NO** es el nivel de confianza de un intervalo de confianza.

3. Por favor explica por qué si en el código anterior usas `set.seed(4)` al comienzo del paso 3, hay más de cinco intervalos de confianza rojos. ¿R se equivocó?

4. Partiendo del código que tienes arriba, ajusta el código para (i) aumentar el número de muestras a **10,000**; (ii) calcular la proporción de intervalos de confianza generados con estas muestras que incluyen al parámetro poblacional. **NO TIENES QUE GRAFICAR LOS 100K INTERVALOS DE CONFIANZA**.

```
# Paso 1 - Generar los datos poblacionales
set.seed(33)
datos <-
  tibble(valores = rnorm(n = 1e6, mean = 0, sd = 1))

# Paso 2 - Definir el tamaño de cada muestra y el número de muestras
tamaño_muestra <- 1000
numero_muestras <- 10000

# Paso 3 - Calcular el error estándar de la media muestral, asumiendo
# que el tamaño de la muestra es 1,000
error_est_con_datos_pob <- sd(datos$valores)/sqrt(tamaño_muestra)

# Paso 4 - Calcular la media de cada muestra y construye su respectivo
# intervalo de confianza usando el error estándar que calculamos
# en el paso 3
set.seed(4)
datos %>%
  rep_sample_n(size = tamaño_muestra,
               replace = T,
               reps = numero_muestras) %>%
  summarise(media_muestral = mean(valores),
            lim_sup = media_muestral + 1.96*error_est_con_datos_pob,
            lim_inf = media_muestral - 1.96*error_est_con_datos_pob) %>%
  summarise(intervalo_incluye = sum(ifelse(lim_sup < 0 | lim_inf > 0, 0, 1))) %>%
  mutate(prop_incluye = intervalo_incluye/numero_muestras)
```

```
## # A tibble: 1 x 2
##   intervalo_incluye prop_incluye
##               <dbl>        <dbl>
## 1              9488        0.949
```

Ejercicio 3

En el paso 3 del ejercicio anterior usé la desviación estándar de la variable poblacional (misma que en el paso 1 definimos como `sd = 1`) para calcular el error estándar de la distribución muestral de la media.

En la vida real, es muy poco probable que conozcamos la desviación estándar de la variable poblacional, así que en este ejercicio asumiremos que no la conocemos.

Por favor calcula el error estándar de la distribución muestral de la media usando la estimación de la desviación estándar de la variable poblacional con los datos de **100** muestras. Guardarás el resultado de este cálculo en una variablle llamada `error_est_datos_muestra`.

OJO: Dado que debes estimar la desviación estándar de la variable poblacional con los datos de cada muestra por separado, al final del ejercicio deberás tener un tibble con 100 renglones (uno por muestra) y tres columnas: `replicate`, `media_muestral` y `error_est_datos_muestra`.

```
# Paso 1 - Generar los datos poblacionales
set.seed(33)
datos <-
  tibble(valores = rnorm(n = 1e6, mean = 0, sd = 1))

# Paso 2 - Definir el tamaño de cada muestra y el número de muestras
tamaño_muestra <- 1000
numero_muestras <- 100

# Paso 3 - Calcular la media de cada muestra y el error estándar de la media muestral, estimando la des
set.seed(4)
datos %>%
  rep_sample_n(size = tamaño_muestra,
               replace = T,
               reps = numero_muestras) %>%
  summarise(media_muestral = mean(valores),
            error_est_datos_muestra = sd(valores)/sqrt(tamaño_muestra))
```

```
## # A tibble: 100 x 3
##    replicate media_muestral error_est_datos_muestra
##        <int>          <dbl>                   <dbl>
##  1         1        0.0184                   0.0314
##  2         2        0.0369                   0.0307
##  3         3        0.00306                  0.0322
##  4         4        0.0372                   0.0317
##  5         5        0.0593                   0.0318
##  6         6        0.0166                   0.0324
##  7         7       -0.0239                   0.0319
##  8         8        0.0249                   0.0306
##  9         9        0.0140                   0.0323
## 10        10       -0.0424                   0.0313
## # ... with 90 more rows
```

1. ¿Los valores en la columna `error_est_datos_muestra` todos iguales? ¿Esto está bien o es señal de que hay un problema?
2. ¿Los valores en la columna `error_est_datos_muestra` son exactamente iguales al valor que obtuvimos en el paso 3 del ejercicio anterio? ¿Esto está bien o es señal de que hay un problema?

Ejercicio 4

Repasemos la diferencia entre error **muestral** y error **estándar**. Para ello usaremos los datos de una distribución **gamma**. En esta (p. 910) y esta liga encontrarás más detalles sobre esta distribución.

a. Usa el siguiente código para construir la distribución:

```
datos <- tibble(valores = rgamma(n = 1e5, shape = 2, scale = 1))
```

b. Toma **100** muestras aleatorias de tamaño **120**, calcula la media de cada muestra y el o los error(es) muestral(es)

```
datos %>%
  rep_sample_n(size = 120, reps = 100) %>%
  summarise(media_muestral = mean(valores)) %>%
  mutate(error_muestral = media_muestral - mean(datos$valores))
```

```
## # A tibble: 100 x 3
##    replicate media_muestral error_muestral
##        <int>          <dbl>          <dbl>
##  1         1           2.00        0.00458
##  2         2           1.91       -0.0776
##  3         3           2.08        0.0862
##  4         4           1.87       -0.125
##  5         5           1.73       -0.258
##  6         6           1.92       -0.0678
##  7         7           2.04        0.0525
##  8         8           2.02        0.0257
##  9         9           2.12        0.133
## 10        10           2.24        0.251
## # ... with 90 more rows
```

c. Toma **100** muestras aleatorias de tamaño **120**, calcula la media de cada muestra y el o los error(es) estándar.

```
datos %>%
  rep_sample_n(size = 120, reps = 100) %>%
  summarise(media_muestral = mean(valores)) %>%
  summarise(error_estandar = sd(media_muestral))
```

```
## # A tibble: 1 x 1
##   error_estandar
##            <dbl>
## 1          0.112
```

Ejercicio 5

Basta de simulaciones. Bueno, casi. En este ejercicio tendrás que seguir el caminito lleno de incertidumbre de toda persona que trata de entender las características de una población a partir de una muestra.

a. Asume que **67%** de la población de México está a favor del paro nacional del 9 de marzo "Un día sin mujeres" (liga) y el **33%** restante está en contra. Construye una población hipotética de **1 millón** de personas con estas proporciones. Puedes representar a los que están a favor del paro con un **1** y a los que están en contra con un **0**.

b. Olvida que sabes las proporciones poblacionales.

c. Planta un `set.seed(1)` y toma **una** sola muestra de tamaño **500**.

d. Usando los datos de tu muestra, estima la proporción de personas que está a favor del paro.

e. Usando los datos de tu muestra, construye un intervalo de confianza de 99%.

f. Reporta los resultados mostrando la/el pro que eres.

```
poblacion <- tibble(preferencia = c(rep(1, 670000), rep(0, 330000)))

tamaño_muestra <-  1000


# set.seed(1)
# poblacion %>%
#   rep_sample_n(size = tamaño_muestra, reps = 1) %>%
#   summarise(proporcion_muestral = mean(preferencia)) %>%
#   mutate(error_estandar = sqrt((proporcion_muestral*(1-proporcion_muestral))/tamaño_muestra),
#          lim_inf = proporcion_muestral - 2.58*error_estandar,
#          lim_sup = proporcion_muestral + 2.58*error_estandar)

# Mi propuesta :3
poblacion %>%
  rep_sample_n(size = tamaño_muestra, reps = 1) %>%
  summarise(proporcion_muestral = mean(preferencia),
            EE = sd(preferencia)/sqrt(tamaño_muestra)
            ) %>%
  mutate(error_estandar = sqrt((proporcion_muestral*(1-proporcion_muestral))/tamaño_muestra),
         lim_inf = proporcion_muestral - 2.58*error_estandar,
         lim_sup = proporcion_muestral + 2.58*error_estandar)
```

```
## # A tibble: 1 x 6
##   replicate proporcion_muestral     EE error_estandar lim_inf lim_sup
##       <int>               <dbl>  <dbl>          <dbl>   <dbl>   <dbl>
## 1         1               0.668 0.0149         0.0149   0.630   0.706
```

g. Repite los pasos c), d) y e) usando una muestra de tamaño **100**, y posteriormente una muestra de tamaño **1,000**. ¿Qué efecto tiene el tamaño de la muestra en el tamaño del intervalo de confianza?