Tarea 05

1. **Proporciones.** Usaremos datos de reincidencia en conducta criminal del estado de Iowa, este estado sigue a los delincuentes por un periodo de 3 años y registra el número de días hasta reincidencia para aquellos que son readmitidos en prisión. El deprtamento de correcciones utiliza los datos de reincidencia para evaluar sus programas de prevención de recaída en conducta criminal.

Los datos Recidivism contienen información de todos los delincuentes condenados por dos tipos de delito durante 2010 (*Recid* indica si recayeron en conducta criminal).

- De éstos 31.6% reincidieron y volvieron a prisión. Utiliza simulación para aproximar la simulación muestral de $\hat{p}$, la proporción de delincuentes que reincidieron para muestras de tamaño 25.
- Calcula el error estándar de $\hat{p}$, y compáralo con el teórico $\sqrt{p(1-p)/n}$.
- Repite para muestras de tamaño 250 y compara.

```
setwd("~/Maestría Ciencia de datos/Fundamentos estadísticos/Tarea_05")
library(tidyverse)

## -- Attaching packages ---------------------------------------------
---- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

recidivism <- read_csv("Recidivism.csv")

## Parsed with column specification:
## cols(
##   Gender = col_character(),
##   Age = col_character(),
##   Age25 = col_character(),
##   Race = col_character(),
##   Offense = col_character(),
##   Recid = col_character(),
##   Type = col_character(),
##   Days = col_double()
## )
```

```r
glimpse(recidivism)
```

```
## Rows: 17,022
## Columns: 8
## $ Gender  <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M",
"M",...
## $ Age     <chr> "Under 25", "55 and Older", "25-34", "55 and Older",
"25-34...
## $ Age25   <chr> "Under 25", "Over 25", "Over 25", "Over 25", "Over
25", "Un...
## $ Race    <chr> "White-NonHispanic", "White-NonHispanic", "White-
NonHispani...
## $ Offense <chr> "Felony", "Felony", "Felony", "Felony", "Felony",
"Felony",...
## $ Recid   <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",
"Yes", "Ye...
## $ Type    <chr> "Tech", "Tech", "Tech", "Tech", "Tech", "Tech", "New",
"Tec...
## $ Days    <dbl> 16, 19, 22, 25, 26, 27, 28, 41, 44, 46, 48, 49, 49,
51, 51,...
```

```r
set.seed(841)
muestra_recid <- sample_n(recidivism, 25) %>%
  select(Gender, Age, Age25, Race, Offense, Recid, Type, Days)

sprintf("Hay %0.0f participantes en total, tomamos muestra de %0.0f",
nrow(recidivism), nrow(muestra_recid))
```

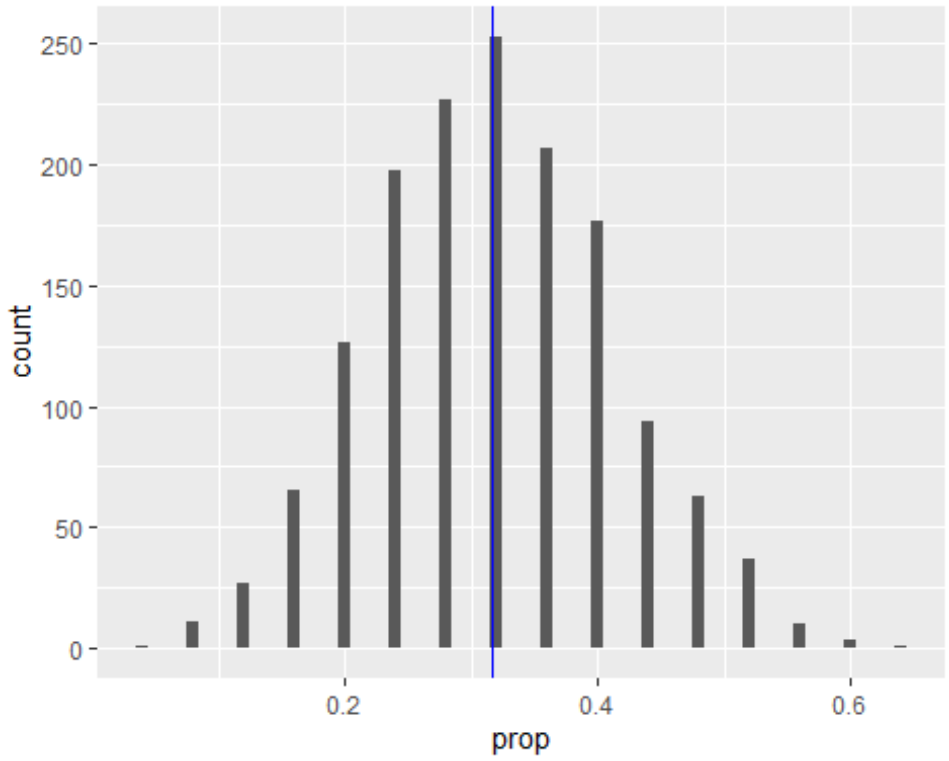
```
## [1] "Hay 17022 participantes en total, tomamos muestra de 25"
```

```r
simulaciones <- 10000
vect_simulacion <- replicate(simulaciones,0)

for (i in 1:simulaciones){
  muestra <- sample(recidivism$Recid,25)
  vect_simulacion [i] <- mean(muestra == 'Yes')
}

mean(vect_simulacion)
```

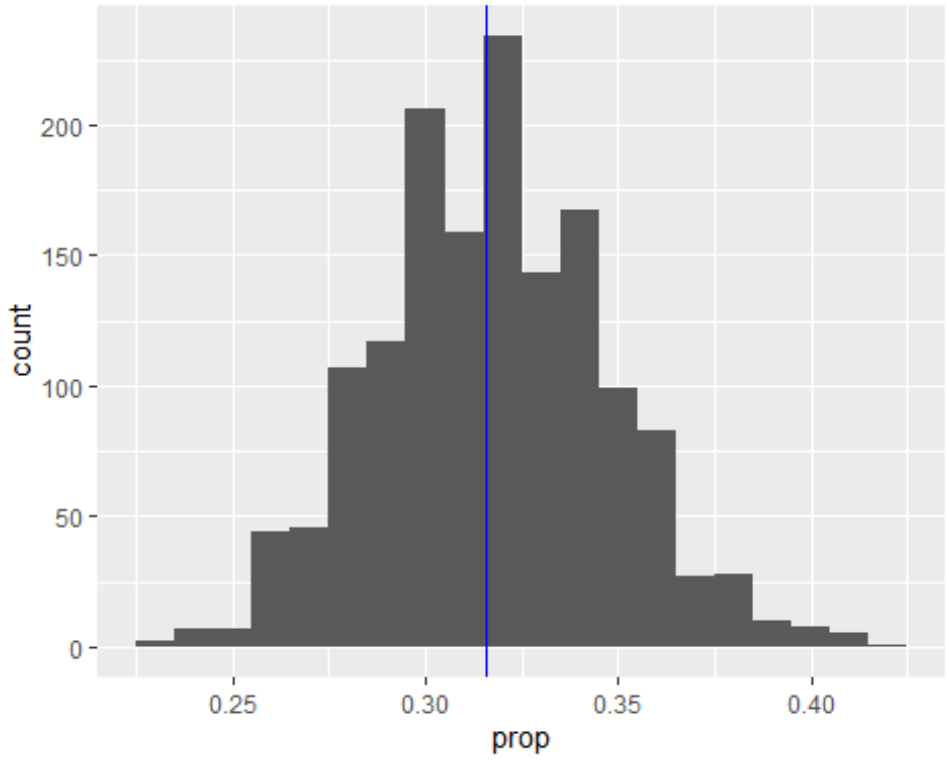
```
## [1] 0.316852
```

```r
n <- nrow(muestra_recid) # tamaño muestra
N <- nrow(recidivism) # tamaño población

estimar_prop <- function(recidivism){
  prop_est <- mean(recidivism$Recid == 'Yes')
  res <- tibble(prop = prop_est, n_muestra = nrow(muestra_recid))
}

est_1 <- estimar_prop(recidivism)
```

```r
replicar_muestreo <- function(recidivism, m, n, estimar_prop){
  # m número de veces que queremos replicar el muestro
  resultados <- map(1:m,
                    function(id){
                      sample_n(recidivism, n) %>%
                       estimar_prop() %>%
                        mutate(id_muestra = id) %>%
                        select(id_muestra, everything())
                    })
  resultados %>% bind_rows
}

replicaciones_1 <- replicar_muestreo(recidivism, m = 1500, n ,
estimar_prop)

set.seed(841)
muestra_recid <- sample_n(recidivism, 250) %>%
  select(Gender, Age, Age25, Race, Offense, Recid, Type, Days)

n <- nrow(muestra_recid) # tamaño muestra
est_2 <- estimar_prop(recidivism)

replicaciones_2 <- replicar_muestreo(recidivism, m = 1500, n ,
estimar_prop)

ggplot(replicaciones_1, aes(x= prop)) +
  geom_histogram(binwidth = 0.01) +
  geom_vline(xintercept = 0.316, colour = 'blue')
```




```r
ggplot(replicaciones_2, aes(x= prop)) +
  geom_histogram(binwidth = 0.01) +
  geom_vline(xintercept = 0.316, colour = 'blue')
```



Se puede ver que mientras más grande es el tamaño de muestra que tomamos, la media muestral está más al rededor de la media poblacional

```
replicas <- bind_rows(replicaciones_1, replicaciones_2)
p_h <- mean(recidivism$Recid == 'Yes')

replicas %>% group_by(n_muestra) %>%
  summarise(ee_muestra = sd(prop),
            ee_teorico = sqrt(p_h*(1 - p_h)/min(n_muestra))
            )

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 3
##   n_muestra ee_muestra ee_teorico
##       <int>      <dbl>      <dbl>
## 1        25     0.0956     0.0930
## 2       250     0.0300     0.0294
```

2. **El error estándar de una media.** Supongamos que x es una variable aleatoria que toma valores en los reales con distribución de probabilidad F. Denotamos por μ y σ^2 la media y varianza de F,

$$\mu = E(x),$$

$$\sigma^2 = var(x) = E[(x-\mu)^2]$$

Ahora, sea $(X_1, \ldots, X_n)$ una muestra aleatoria de F, de tamaño n, la media de la muestra $\overline{X} = \sum_{i=1}^{n} X_i \, /n$ tiene:

- esperanza μ,
- varianza σ^2/n.

En palabras: la esperanza de $\overline{X}$ es la misma que la esperanza de x, pero la varianza de $\overline{X}$ es $1/n$ veces la varianza de x, así que entre mayor es la n tenemos una mejor estimación de μ.

En el caso del estimador de la media $\overline{X}$, el error estándar quedaría

$$ee(\overline{X}) = [var(\overline{X})]^{1/2} = \sigma/\sqrt{n}.$$

Entonces,

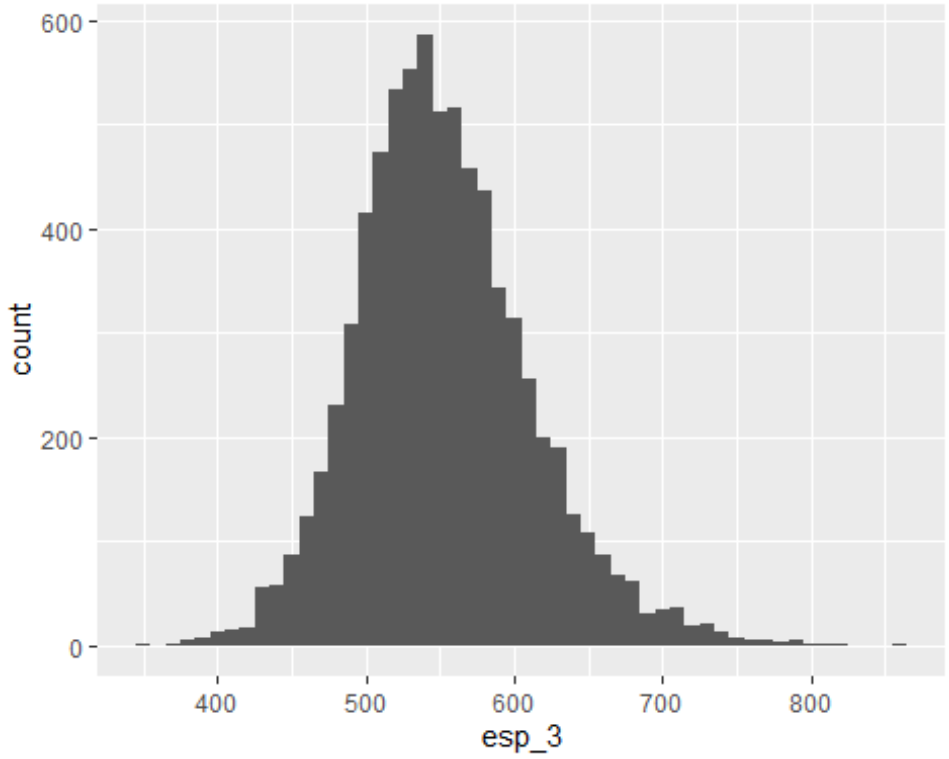
- Consideramos los datos de ENLACE edo. de México (ENLACE era una prueba estandarizada que se aplicaba a todos los alumnos de primaria en México), y la columna de calificaciones de español 3º de primaria (`esp_3`).

```
enlace <- read_csv("enlace_15.csv")

## Parsed with column specification:
## cols(
```

```
##   id = col_double(),
##   cve_ent = col_double(),
##   turno = col_character(),
##   tipo = col_character(),
##   esp_3 = col_double(),
##   esp_6 = col_double(),
##   n_eval_3 = col_double(),
##   n_eval_6 = col_double()
## )
```

- Genera un histograma de las calificaciones de 3º de primaria. Calcula la media y la desviación estándar.

```
ggplot(enlace, aes(x = esp_3)) +
  geom_histogram(binwidth = 10)
```




```
media_enlace <-  enlace %>% summarise(meida = mean(esp_3))
sd_enlace <- enlace %>% summarise(meida = sd(esp_3))

media_enlace
```

```
## # A tibble: 1 x 1
##   meida
##   <dbl>
## 1  553.
```

```
sd_enlace
```

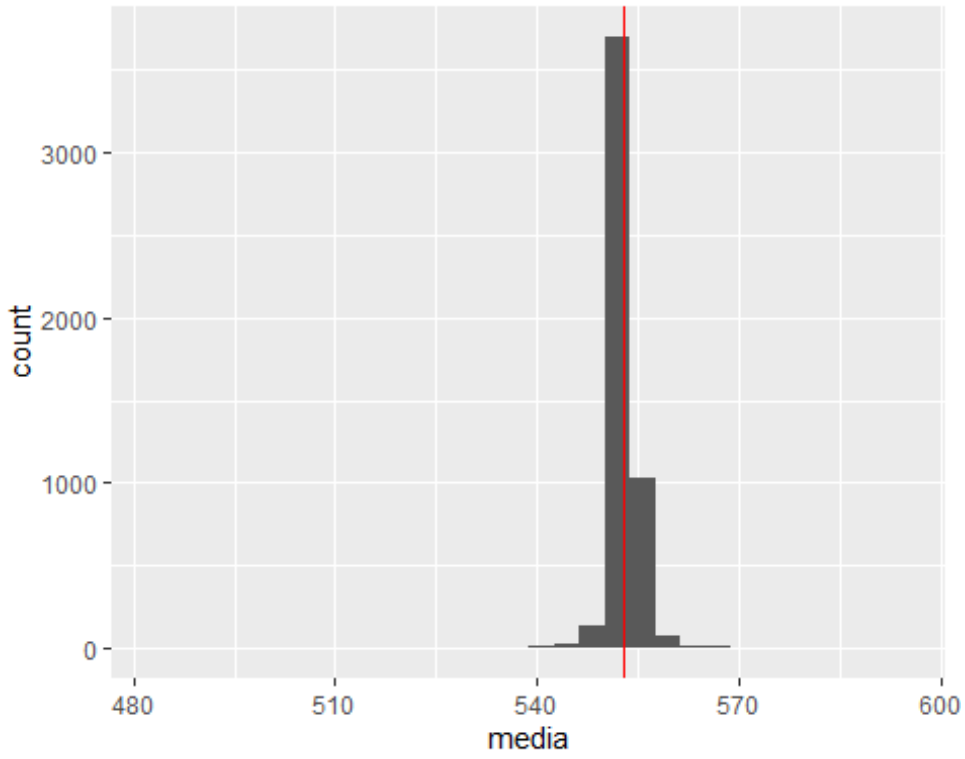
```
## # A tibble: 1 x 1
##   meida
##   <dbl>
## 1  59.3
```

- Para tamaños de muestra $n = 10,100,1000$:
- Aproximareos la distribución muestral:

i) simula 5,000 muestras aleatorias, ii) calcula la media en cada muestra, iii) Realiza un histograma de la distribución muestral de las medias (las medias del paso anterior) iv) aproxima el error estándar calculando la desviación estándar de las medias del paso ii.

- Calcula el error estándar de la media para cada tamaño de muestra usando la fórmula derivada arriba y compara con tus simulaciones.

```
muestra_enlace <- sample_n(enlace, 10) %>%
  select(id, cve_ent, turno, tipo, esp_3, esp_6, n_eval_3, n_eval_6)

estimar_media <- function(n){
  enlace %>%
    sample_n(n) %>%
    summarise(media = mean(esp_3))
}

n <- 10
m <- 5000
# m es el tamaño de la muestra

replicar_muestreo <- function(enlace, m, n, estimar_media){
  # m número de veces que queremos replicar el muestro
  resultados <- map(1:m, estimar_media)
  resultados %>% bind_rows
}

muestra_10 <- replicar_muestreo(enlace, m, n, estimar_media)

ggplot(muestra_10, aes(x = media)) +
  geom_histogram() +
  geom_vline(xintercept = mean(muestra_10$media), col = 'red')

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
sd(muestra_10$media)
```

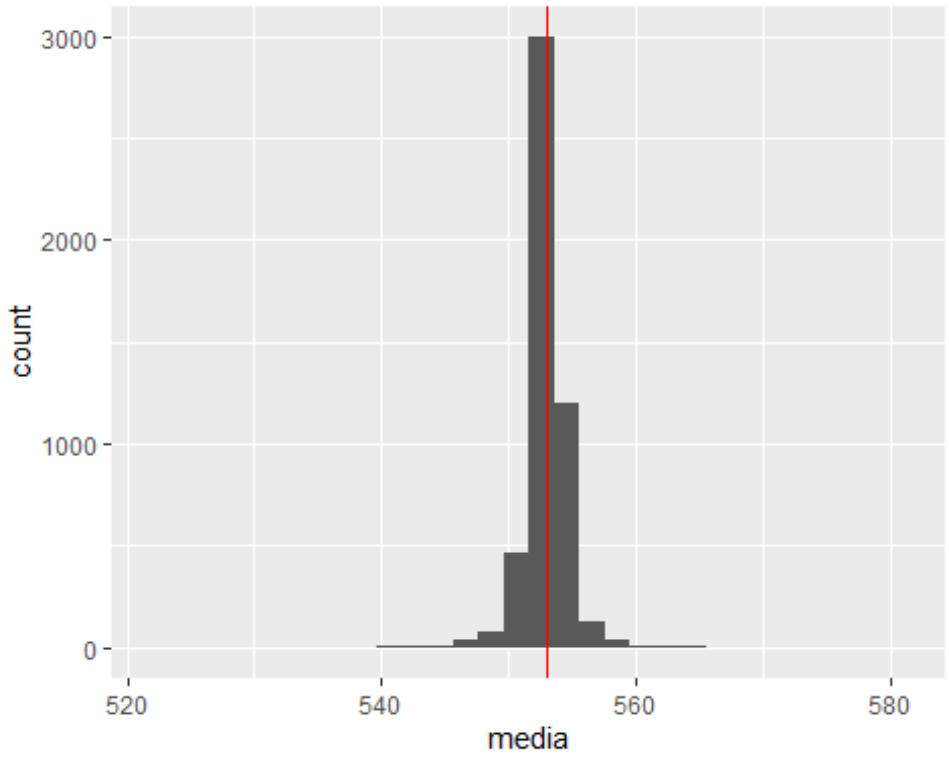
```
## [1] 2.398347
```

```r
sd(enlace$esp_3)/sqrt(10)
```

```
## [1] 18.73902
```

```r
n <- 100
muestra_100 <- replicar_muestreo(enlace, m, n, estimar_media)

ggplot(muestra_100, aes(x = media)) +
  geom_histogram() +
  geom_vline(xintercept = mean(muestra_100$media), col = 'red')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
sd(muestra_100$media)
```

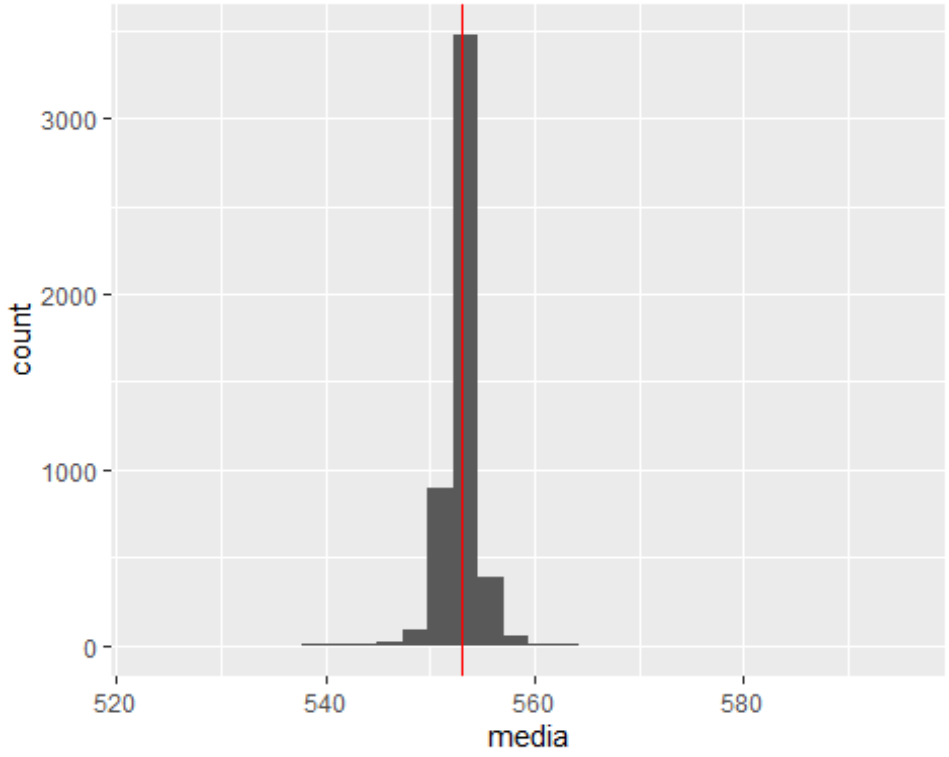
```
## [1] 2.087515
```

```r
sd(enlace$esp_3)/sqrt(100)
```

```
## [1] 5.925797
```

```r
n <-1000

muestra_1000 <- replicar_muestreo(enlace, m, n, estimar_media)

ggplot(muestra_1000, aes(x = media)) +
  geom_histogram() +
  geom_vline(xintercept = mean(muestra_1000$media), col = 'red')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
sd(muestra_1000$media)
```

```
## [1] 2.139605
```

```
sd(enlace$esp_3)/sqrt(1000)
```

```
## [1] 1.873902
```

- ¿Cómo se comparan los errores estándar correspondientes a los distintos tamaños de muestra?

Entre más grande es la muestra nuestro error estándar es más chico.