Estadística Inferencial

Programación para el análisis de datos

Departamento de Ciencias Sociales, UCU - Martín Opertti

Intervalos de confianza

Simular datos de resultados electorales

```
# Fijo semilla para reproducir los resultados
set.seed(77)

# Crear data poblacional
datos_votos <- sample(x = c("Partido A", "Partido B", "Partido C"),
                      size = 3000000,
                      replace = TRUE,
                      prob = c(.25, .35, .4)
)

datos_votos <- tibble(voto = datos_votos)

datos_votos
```

```
## # A tibble: 3,000,000 x 1
##    voto
##    <chr>
##  1 Partido C
##  2 Partido B
##  3 Partido A
##  4 Partido A
##  5 Partido B
##  6 Partido B
##  7 Partido A
##  8 Partido A
##  9 Partido B
## 10 Partido B
## # ... with 2,999,990 more rows
```

Resumen datos

```
# Tabla resumen de los datos simulados
datos_votos %>%
  group_by(voto) %>%
  summarise(n = n()) %>%
  mutate(per = n / sum(n))
```

```
## # A tibble: 3 x 3
##   voto            n   per
##   <chr>       <int> <dbl>
## 1 Partido A  750865 0.250
## 2 Partido B 1050109 0.350
## 3 Partido C 1199026 0.400
```

Resumen con margen de error e intervalo de confianza

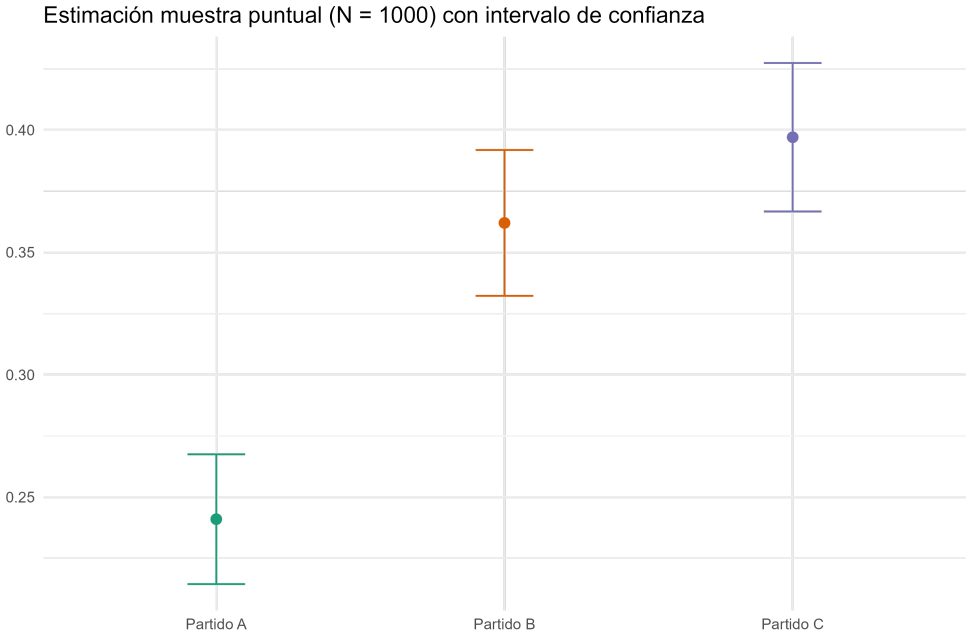
R tiene muchas funciones para realizar estadística inferencial. En el caso de los intervalos de confianza -entre otras opciones- podemos calcularlos manualmente o, por ejemplo, con la función `summarySE()` del paquete Rmisc.

```
# Extraigo una muestra de 1000 casos
m_1000 <- sample_n(datos_votos, 1000)

# Ahora calculamos manualmente el margen de error e intervalo de confianza
# para esta muestra
m_1000_sum <- m_1000 %>%
  group_by(voto) %>%
  summarise(n = n()) %>%
  mutate(
    prop = n/sum(n), # Proporción de cada categoría
    moe = (qnorm(0.975) * sqrt(prop*(1-prop)/1000)), # margen de error al 95% co
    ci_inf =  prop - moe, # Intervalo superior
    ci_sup =  prop + moe # Intervalo superior
  )

m_1000_sum
```

```
## # A tibble: 3 x 6
##   voto           n  prop    moe ci_inf ci_sup
##   <chr>      <int> <dbl>  <dbl>  <dbl>  <dbl>
## 1 Partido A    241 0.241 0.0265  0.214  0.268
## 2 Partido B    362 0.362 0.0298  0.332  0.392
## 3 Partido C    397 0.397 0.0303  0.367  0.427
```

Intervalos de confianza con geom_errorbar()

```
# Graficamos
ggplot(m_1000_sum, aes(x = voto, y = prop, color = voto)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymin = ci_inf, ymax = ci_sup), width = .2, lwd = .75) +
  theme_minimal(base_size = 16) +
  labs(title = "Estimación muestra puntual (N = 1000) con intervalo de confianza",
       x = "",
       y = "") +
  scale_color_brewer(palette = "Dark2") +
  theme(legend.position = "none")
```



Correlación

Correlación

En R la correlación entre dos variables se calcula con la función `cor()`

```
gapminder <- gapminder::gapminder

cor(gapminder$lifeExp, gapminder$gdpPercap)
```

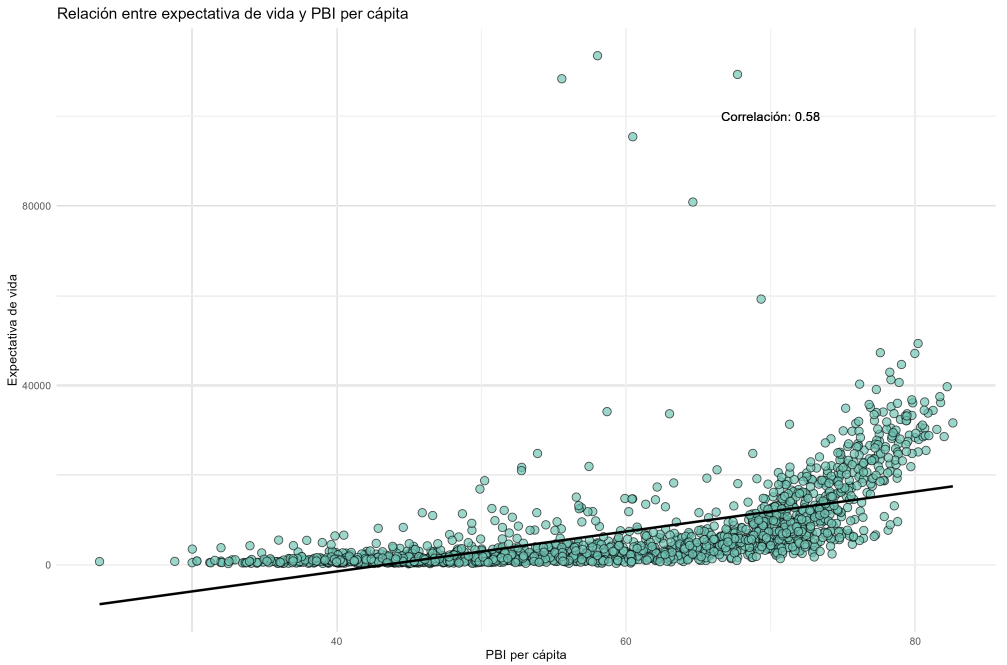
```
## [1] 0.5837062
```

```
cor(gapminder$lifeExp, gapminder$gdpPercap, method = "spearman")
```

```
## [1] 0.8264712
```

Graficar correlación

```
ggplot(data = gapminder, aes(x = lifeExp, y = gdpPercap)) +
  geom_point(shape = 21, fill = '#73C6B6',  size = 3, alpha = .7) +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  geom_text(label = paste("Correlación:", round(cor(gapminder$lifeExp, gapminder
  theme_minimal()
```



Pruebas de hipótesis

Pruebas t

En R podemos realizar pruebas t con La función t.test() del R Base.

En este casos realizaremos una prueba t para muestras independientes (dos grupos)

```
df_gap <- filter(gapminder::gapminder,
                 continent %in% c("Americas", "Europe")) %>%
  mutate(continent = as.factor(as.character(continent)))

prueba_t <- t.test(lifeExp ~ continent, data = df_gap)

prueba_t
```

```
##
##     Welch Two Sample t-test
##
## data:  lifeExp by continent
## t = -11.861, df = 460.72, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.445287 -6.044612
## sample estimates:
## mean in group Americas   mean in group Europe
##                64.65874                71.90369
```

Limpiar resultados con broom

Para analizar mejor los resultados de nuestros modelos podemos utilizar el paquete broom. La función `tidy()`, por ejemplo, nos permite extraer los resultados de la prueba (o de cualquier modelo de regresión) en un dataframe en formato tidy

```
prueba_t <- broom::tidy(prueba_t)

prueba_t
```

```
## # A tibble: 1 x 10
##   estimate estimate1 estimate2 statistic  p.value parameter conf.low conf.high
##      <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>    <dbl>     <dbl>
## 1    -7.24      64.7      71.9     -11.9 1.67e-28      461.    -8.45     -6.04
## # ... with 2 more variables: method <chr>, alternative <chr>
```

Limpiar resultados con broom

Al transformar los resultados en un objeto `tibble()` podemos manipular, visualizar y guardar los resultados de igual manera que cualquier otro dataframe.

```
prueba_t <- prueba_t %>%
  rename(estimador = estimate)

prueba_t
```

```
## # A tibble: 1 x 10
##   estimador estimate1 estimate2 statistic  p.value parameter conf.low conf.high
##       <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>    <dbl>     <dbl>
## 1     -7.24      64.7      71.9     -11.9 1.67e-28      461.    -8.45     -6.04
## # ... with 2 more variables: method <chr>, alternative <chr>
```

```
writexl::write_xlsx(prueba_t, "resultados_t.xlsx")
```

Modelos de regresión

Modelos de regresión lineal

R trae un conjunto de funciones para estimar modelos de regresión. `lm()` sirve para estimar una regresión lineal. El primer argumento es la variable dependiente, luego `~` seguido de las variables independientes separadas por `+`, luego en el argumento data especificamos el dataframe a utilizar:

```
reg <- lm(var_dependiente ~ var_ind_1 + var_ind2, data = mi_data)

summary(reg) # Con summary podemos ver los resultados
```

Modelos de regresión

Volvamos a la data de gapminder y estimemos un modelo de regresión cuya variable dependiente sea expectativa de vida y las variables independientes el PBI per cápita.

```
gapminder <- gapminder::gapminder

reg <- lm(lifeExp ~ gdpPercap + pop + year + continent, data = gapminder)

summary(reg) # Con summary podemos ver los resultados
```

```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap + pop + year + continent, data = gapminder)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.4051  -4.0550   0.2317   4.5073  20.0217
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -5.185e+02  1.989e+01 -26.062   <2e-16 ***
## gdpPercap          2.985e-04  2.002e-05  14.908   <2e-16 ***
## pop                1.791e-09  1.634e-09   1.096    0.273
## year               2.863e-01  1.006e-02  28.469   <2e-16 ***
## continentAmericas  1.429e+01  4.946e-01  28.898   <2e-16 ***
## continentAsia      9.375e+00  4.719e-01  19.869   <2e-16 ***
## continentEurope    1.936e+01  5.182e-01  37.361   <2e-16 ***
## continentOceania   2.056e+01  1.469e+00  13.995   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.883 on 1696 degrees of freedom
## Multiple R-squared:  0.7172,    Adjusted R-squared:  0.716
## F-statistic: 614.5 on 7 and 1696 DF,  p-value: < 2.2e-16
```

Modelos de regresión generalizados

Con `glm()` podemos estimar otros modelos como probit, poisson o logit. `glm()` tiene la misma lógica que `lm()` solo que especificamos el tipo de modelo mediante el argumento family y cuando es necesario una especificación adicional con el argumento link

```
# Logit
reg <- glm(var_dependiente ~ var_ind_1 + var_ind2,
           data = mi_data,
           family = binomial(link = "logit"))

# Probit
reg <- glm(var_dependiente ~ var_ind_1 + var_ind2,
           data = mi_data,
           family = binomial(link = "probit"))

# Poisson
reg <- glm(var_dependiente ~ var_ind_1 + var_ind2,
           data = mi_data,
           family = "poisson")
```

Regresión logística

Ahora creemos una nueva variable de expectativa de vida que sea binaria y estimemos una regresión logística. La variable tendrá valor 1 cuando la expectativa de vida sea mayor a 70 y 0 si no lo es.

```
# Creo nueva variable
gapminder <- mutate(gapminder,
                    lifeExp_rec = case_when(lifeExp > 70 ~ 1,
                                            TRUE ~ 0)
                    )

# Por más que tenga solo dos valores, es numérica
class(gapminder$lifeExp_rec)
```

```
## [1] "numeric"
```

```
# Para esto debo transformarla a factor
gapminder <- mutate(gapminder,
                    lifeExp_rec = as.factor(lifeExp_rec))

class(gapminder$lifeExp_rec)
```

```
## [1] "factor"
```

Regresión logística

```
reg_logit <- glm(lifeExp_rec ~ gdpPercap + pop + year + continent,
                 data = gapminder,
                 family = binomial(link = "logit"))
summary(reg_logit)
```

```
##
## Call:
## glm(formula = lifeExp_rec ~ gdpPercap + pop + year + continent,
##     family = binomial(link = "logit"), data = gapminder)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3649  -0.3727  -0.1272   0.1175   2.6974
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.052e+02  1.486e+01 -13.808  < 2e-16 ***
## gdpPercap          1.221e-04  1.431e-05   8.534  < 2e-16 ***
## pop                1.520e-10  6.416e-10   0.237    0.813
## year               1.011e-01  7.434e-03  13.604  < 2e-16 ***
## continentAmericas  3.010e+00  3.073e-01   9.795  < 2e-16 ***
## continentAsia      1.951e+00  3.088e-01   6.317 2.67e-10 ***
## continentEurope    5.221e+00  3.745e-01  13.941  < 2e-16 ***
## continentOceania   6.911e+00  9.175e-01   7.532 4.98e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2050.05  on 1703  degrees of freedom
## Residual deviance:  869.98  on 1696  degrees of freedom
## AIC: 885.98
##
## Number of Fisher Scoring iterations: 6
```

Regresión logística

Cambiar categoría de referencia de un factor

```
gapminder <- mutate(gapminder,
                    continent = relevel(continent, ref = "Americas"))

reg_logit_2 <- glm(lifeExp_rec ~ gdpPercap + pop + year + continent,
                   family = binomial(link = "logit"),
                   data = gapminder)

summary(reg_logit_2)
```

```
##
## Call:
## glm(formula = lifeExp_rec ~ gdpPercap + pop + year + continent,
##     family = binomial(link = "logit"), data = gapminder)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3649  -0.3727  -0.1272   0.1175   2.6974
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.022e+02  1.477e+01 -13.686  < 2e-16 ***
## gdpPercap         1.221e-04  1.431e-05   8.534  < 2e-16 ***
## pop               1.520e-10  6.416e-10   0.237    0.813
## year              1.011e-01  7.434e-03  13.604  < 2e-16 ***
## continentAfrica  -3.010e+00  3.073e-01  -9.795  < 2e-16 ***
## continentAsia    -1.059e+00  2.374e-01  -4.463 8.10e-06 ***
## continentEurope   2.211e+00  2.667e-01   8.291  < 2e-16 ***
## continentOceania  3.901e+00  8.704e-01   4.482 7.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

Modelos de regresión

Utilizamos nuevamente la función tidy del paquete Broom para limpiar los resultados.

```
library(broom)

coef <- tidy(reg, conf.int = TRUE)

print(coef)
```

```
## # A tibble: 8 x 7
##   term                estimate std.error statistic   p.value conf.low conf.high
##   <chr>                  <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)       -5.18e+2   1.99e+1    -26.1  3.25e-126 -5.57e+2  -4.79e+2
## 2 gdpPercap          2.98e-4   2.00e-5     14.9  2.52e- 47  2.59e-4   3.38e-4
## 3 pop                1.79e-9   1.63e-9      1.10 2.73e-  1 -1.41e-9   5.00e-9
## 4 year               2.86e-1   1.01e-2     28.5  4.80e-146  2.67e-1   3.06e-1
## 5 continentAmericas  1.43e+1   4.95e-1     28.9  1.18e-149  1.33e+1   1.53e+1
## 6 continentAsia      9.38e+0   4.72e-1     19.9  3.80e- 79  8.45e+0   1.03e+1
## 7 continentEurope    1.94e+1   5.18e-1     37.4  2.03e-223  1.83e+1   2.04e+1
## 8 continentOceania   2.06e+1   1.47e+0     14.0  3.39e- 42  1.77e+1   2.34e+1
```

Modelos de regresión

Uso `mutate_if()` para redondear todas las variables numéricas, para utilizar `mutate()` para varias columnas al mismo tiempo ver

```
# También para la regresión logística

coef_log2 <- tidy(reg_logit_2, conf.int = TRUE) %>%
    mutate_if(is.numeric, ~ round(., 4))

print(coef_log2)
```

```
## # A tibble: 8 x 7
##   term              estimate std.error statistic p.value  conf.low conf.high
##   <chr>                <dbl>     <dbl>     <dbl>   <dbl>     <dbl>     <dbl>
## 1 (Intercept)      -202.       14.8      -13.7     0      -232.      -174.
## 2 gdpPercap           0.0001    0           8.53    0         0.0001    0.0002
## 3 pop                 0         0           0.237   0.813     0         0
## 4 year                0.101     0.0074     13.6     0         0.087     0.116
## 5 continentAfrica    -3.01      0.307      -9.79    0        -3.64     -2.43
## 6 continentAsia      -1.06      0.237      -4.46    0        -1.53     -0.598
## 7 continentEurope     2.21      0.267       8.29    0         1.70      2.75
## 8 continentOceania    3.90      0.870       4.48    0         2.35      5.90
```

Modelos de regresión

Si utilizamos fijamos `exponentiate = TRUE` dentro de `tidy()` en una regresión logística obtenemos los odds ratios

```
coef_log3 <- tidy(reg_logit_2,
                  exponentiate = TRUE,
                  conf.int = TRUE) %>%
    mutate_if(is.numeric, ~ round(., 5))

print(coef_log3)
```

```
## # A tibble: 8 x 7
##   term              estimate std.error statistic p.value conf.low conf.high
##   <chr>                <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)        0        14.8       -13.7   0          0         0
## 2 gdpPercap          1.00      0.00001     8.53  0          1.00      1.00
## 3 pop                1         0           0.237 0.813      1         1
## 4 year               1.11      0.00743    13.6   0          1.09      1.12
## 5 continentAfrica    0.0493    0.307      -9.79  0          0.0264    0.0883
## 6 continentAsia      0.347     0.237      -4.46  0.00001    0.216     0.550
## 7 continentEurope    9.12      0.267       8.29  0          5.47     15.6
## 8 continentOceania  49.4       0.870       4.48  0.00001   10.5     366.
```

Modelos de regresión

Con `glance()` también del paquete broom podemos obtener un tibble de una fila con estadísticas de bondad del modelo

```
glance(reg_logit_2)
```

```
## # A tibble: 1 x 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1         2050.    1703  -435.  886.  930.     870.        1696  1704
```

Visualizar coeficientes

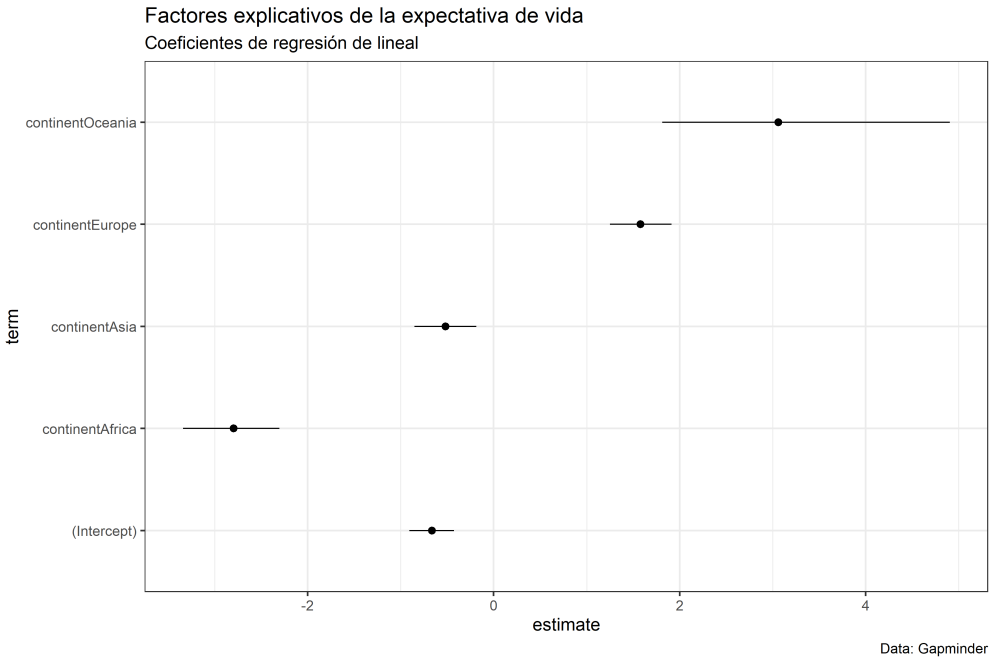
Con broom y ggplot2 podemos graficar los coeficientes de regresión:

```
# Modelo con un solo predictor (continentes)
r_logit_1 <- glm(lifeExp_rec ~ continent,
                 family = binomial(link = "logit"),
                 data = gapminder)

coef_l_1 <- broom::tidy(r_logit_1, conf.int = TRUE)

coef_l_1
```

```
## # A tibble: 5 x 7
##   term              estimate std.error statistic  p.value conf.low conf.high
##   <chr>                <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)        -0.663     0.122     -5.44 5.26e- 8   -0.906    -0.427
## 2 continentAfrica    -2.80      0.263    -10.6  1.96e-26   -3.34     -2.30
## 3 continentAsia      -0.518     0.170     -3.05 2.32e- 3   -0.852    -0.185
## 4 continentEurope     1.58      0.169      9.35 8.55e-21    1.25      1.91
## 5 continentOceania    3.06      0.749      4.09 4.32e- 5    1.81      4.90
```

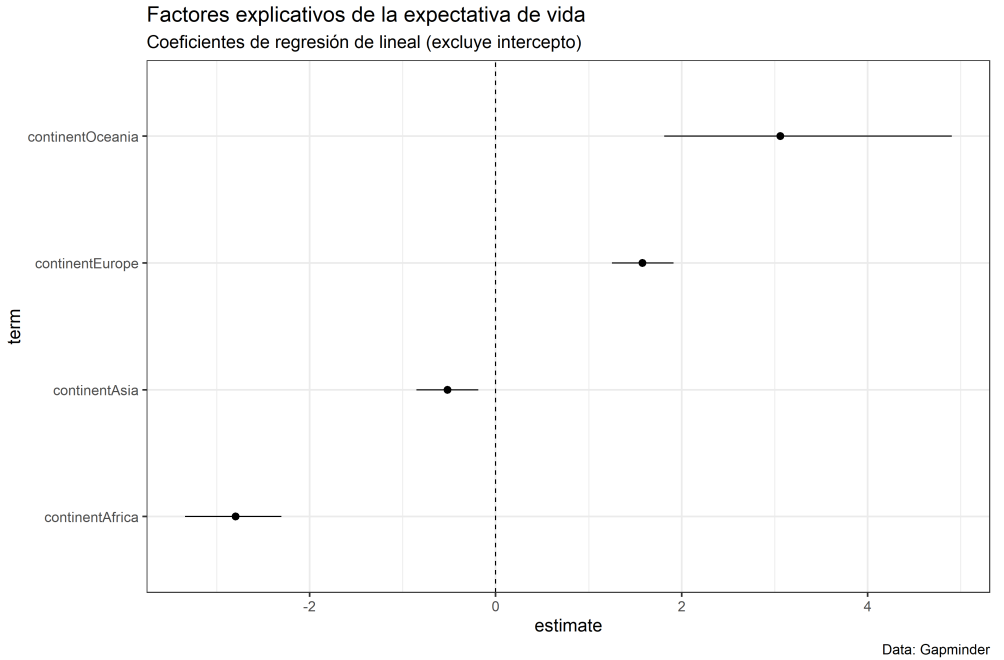
Visualizar coeficientes

```
ggplot(coef_l_1, aes(x = estimate, y = term)) +
  geom_pointrange(aes(xmin = conf.low, xmax = conf.high)) +
  labs(title = "Factores explicativos de la expectativa de vida",
       subtitle = "Coeficientes de regresión de lineal",
       caption = "Data: Gapminder")
```



Visualizar coeficientes

```
# Quitamos el intercepto y agregamos línea vertical en 0
  ggplot(coef_l_1 %>%
  filter(term != "(Intercept)"), aes(x = estimate, y = term)) +
  geom_pointrange(aes(xmin = conf.low, xmax = conf.high)) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  labs(title = "Factores explicativos de la expectativa de vida",
       subtitle = "Coeficientes de regresión de lineal (excluye intercepto)",
       caption = "Data: Gapminder")
```



Comparar coeficientes de dos modelos anidados

Estimamos dos modelos anidados y los unimos los dos dataframes que contienen los resultados obtenidos con la función `tidy()`

```
# Solo continente
r_logit_1 <- glm(lifeExp_rec ~ continent,
                 family = binomial(link = "logit"),
                 data = gapminder)

coef_l_1 <- tidy(r_logit_1, conf.int = TRUE)

# Continente + gdp
r_logit_2 <- glm(lifeExp_rec ~ continent + gdpPercap,
                 family = binomial(link = "logit"),
                 data = gapminder)

coef_l_2 <- tidy(r_logit_2, conf.int = TRUE)

# Primero variable que identifique cada dataframe
coef_l_1 <- mutate(coef_l_1, modelo = "Modelo 1")
coef_l_2 <- mutate(coef_l_2, modelo = "Modelo 2")

# Unimos los resultados de ambos modelos
coef_l_merge <- rbind(coef_l_1, coef_l_2)
```

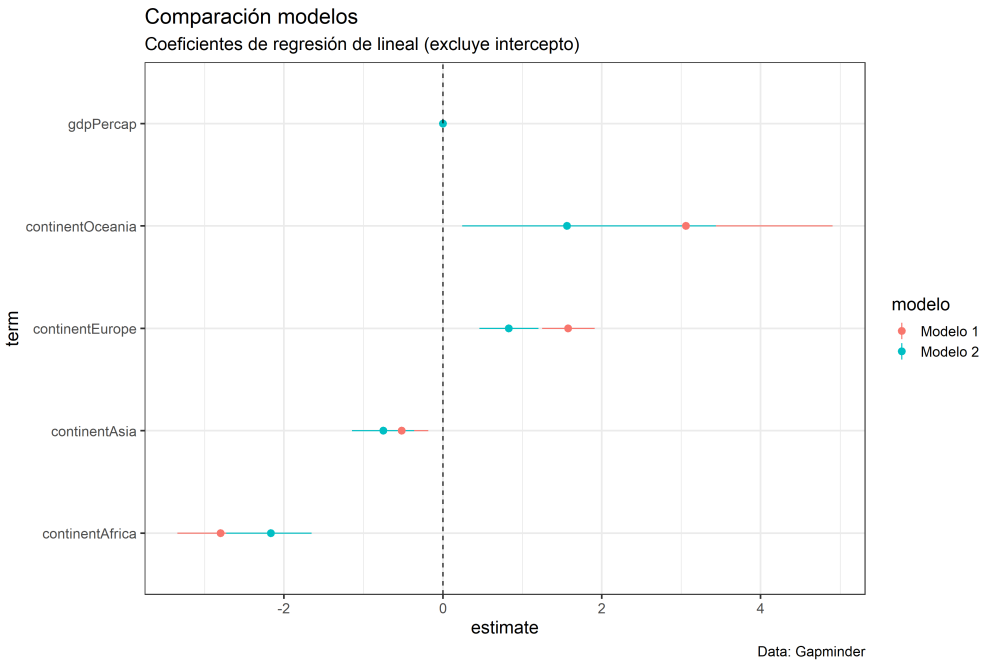
Comparar coeficientes de dos modelos anidados

```
# La data resultante:
coef_l_merge
```

```
## # A tibble: 11 x 8
##    term          estimate std.error statistic  p.value conf.low conf.high modelo
##    <chr>            <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl> <chr>
##  1 (Intercept) -0.663    0.122         -5.44 5.26e- 8 -9.06e-1 -0.427    Modelo~
##  2 continentA~ -2.80     0.263        -10.6  1.96e-26 -3.34e+0 -2.30     Modelo~
##  3 continentA~ -0.518    0.170         -3.05 2.32e- 3 -8.52e-1 -0.185    Modelo~
##  4 continentE~  1.58     0.169          9.35 8.55e-21  1.25e+0  1.91     Modelo~
##  5 continentO~  3.06     0.749          4.09 4.32e- 5  1.81e+0  4.90     Modelo~
##  6 (Intercept) -1.79     0.156        -11.5  1.39e-30 -2.10e+0 -1.49     Modelo~
##  7 continentA~ -2.17     0.273         -7.94 2.06e-15 -2.73e+0 -1.65     Modelo~
##  8 continentA~ -0.750    0.200         -3.76 1.72e- 4 -1.14e+0 -0.361    Modelo~
##  9 continentE~  0.831    0.189          4.39 1.13e- 5  4.62e-1  1.20     Modelo~
## 10 continentO~  1.56     0.774          2.02 4.36e- 2  2.44e-1  3.44     Modelo~
## 11 gdpPercap    0.000159 0.0000125     12.8  3.06e-37  1.35e-4  0.000184 Modelo~
```

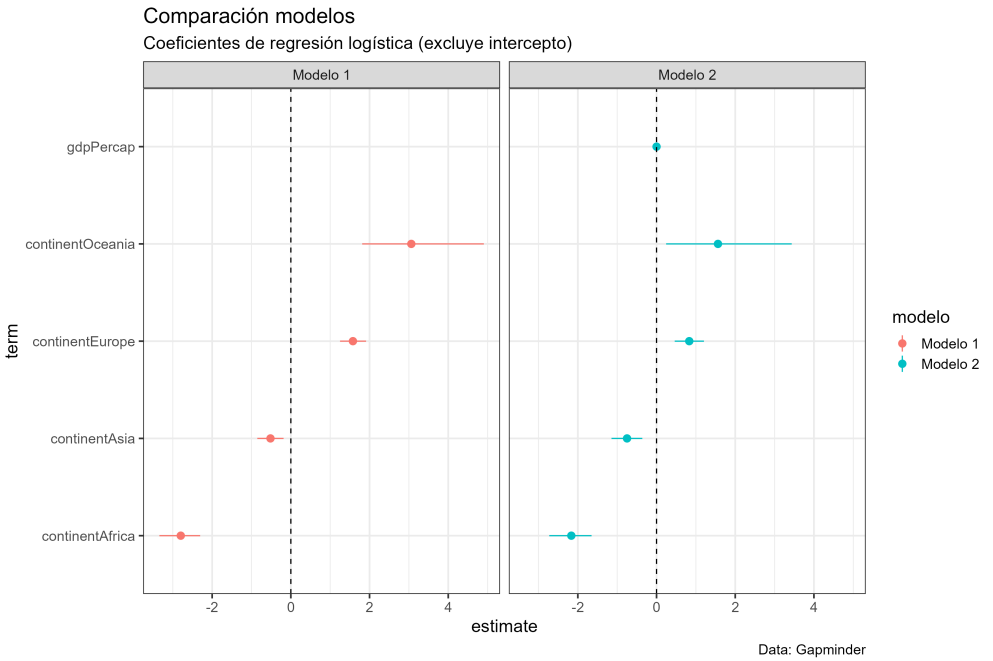
Comparar coeficientes de dos modelos anidados

```
coef_l_merge %>%
  filter(term != "(Intercept)") %>%
  ggplot(aes(x = estimate, y = term, color = modelo)) +
  geom_pointrange(aes(xmin = conf.low, xmax = conf.high)) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  labs(title = "Comparación modelos",
       subtitle = "Coeficientes de regresión logística (excluye intercepto)",
       caption = "Data: Gapminder")
```



Comparar coeficientes de dos modelos anidados

```
coef_l_merge %>%
  filter(term != "(Intercept)") %>%
  ggplot(aes(x = estimate, y = term, color = modelo)) +
  geom_pointrange(aes(xmin = conf.low, xmax = conf.high)) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  labs(title = "Comparación modelos",
       subtitle = "Coeficientes de regresión logística (excluye intercepto)",
       caption = "Data: Gapminder") +
  facet_wrap(~ modelo)
```



Clases y evaluaciones

Clases restantes

- Viernes 11: Clase extra
- Martes 15: Parcial R
- Jueves 17: Reportes con R Markdown y pauta trabajo final
- Martes 22: Técnicas de programación avanzada
- Jueves 24: Taller trabajo final
- Martes 29: Entrega y presentación del trabajo final
- Jueves 1: Cierre del curso

Parcial R

- Importar y exportar datos
- Clases y tipos de objetos en R
- Indexación de vectores y dataframes
- Interpetar mensajes de error
- Explorar argumentos de una función
- Verbos del dplyr y pipeline
- Resumir datos con `group_by()` y `summarise()`
- Recodifiación de variables con `case_when()`
- Unir y cambiar de estructura datos
- Gráficos con ggplot2