# Informe final - Términos de Referencia del llamado AT06-2023

Paquete ARISTAS en lenguaje R

Hanwen Zhang, hanwenzhang1982@gmail.com

2024-07-21

## ARISTAS

El Instituto Nacional de Evaluación Educativa tiene como función principal "aportar información que contribuya a garantizar el derecho de los educandos a recibir una educación de calidad". La misión institucional del INEEd es aportar información que enriquezca la discusión sobre políticas educativas, que sea relevante para la gestión y las prácticas educativas, y que nutra los debates públicos sobre la educación.

Como parte de su plan estratégico para el período 2021-2025, el INEEd se propuso diseñar una línea de estudios sobre trayectorias de adolescentes. En ese marco, como parte del plan de trabajo para 2023-2024 se encuentra desarrollando el proyecto de investigación "Trayectorias educativas de los estudiantes uruguayos: aportando información para su acompañamiento desde los centros educativos".

Dentro de este marco, el INEEd realizó un llamado para la contratación de un consultor para la elaboración de un paquete de R (Términos de Referencia AT06-2023), que incorpore: (i) el cálculo de estimaciones puntual, intervalos de confianza y pruebas de hipótesis de la media (para variables continuas, como los desempeños o índices) o la proporción para variables categóricas, como niveles de desempeño o repetición, (ii) comparar los resultados de las diferentes variables entre distintas ediciones de Aristas, y (iii) ajuste de modelos lineales, multinivel para variable numéricas y modelo de regresión logística para variables binarias. Es importante resaltar que los anteriores análisis deben considerar el diseño de la muestra ARISTAS incluyendo los ponderadores y pesos réplicas, de tal forma que los resultados de análisis sean válido a nivel de las poblaciones.

Como resultado del desarrollo de este proyecto, se ha construido el paquete llamado `ARISTAS` en el software R (R Core Team (2023).) que contienen por un lado funciones que efectúan los anteriores análisis, y por otro lado, las bases de datos de ARISTAS edición 2018 y 2022 de contexto, lecturas, matemáticas, socioemocional.

El paquete puede ser instalado ejecutando desde R o RStudio ejecutando el siguiente comando:

```
devtools::install_github("hanwengutierrez/ARISTAS")
```

Posteriormente, se debe cargar el paquete ejecutando el siguiente comando:

```
library(ARISTAS)
```

## Alcance del paquete

El paquete está diseñado inicialmente para el análisis de las bases de datos de ARISTAS edición 2018 y 2022, ya incluidas en el mismo paquete (que estarán disponbiles para ser usados una vez el usuario cargue el paquete con el comando `library(ARISTAS)`). Sin embargo, al aplicar nuevas pruebas ARISTAS en futuras ediciones, éstas también pueden ser analizadas por el paquete de alguna de las siguientes formas:

(1) por medio de la actualización del paquete ARISTAS, se incluyen las nuevas bases de dato de las nuevas ediciones.

(2) el usuario importa la nueva base de datos desde el disco local al ambiente de `R` para analizar con las funciones del paquete.

En cualquier de estos dos casos, las nuevas bases de datos deben guardar las mismas esctrucas que las ediciones de 2018 y 2022, esto es, (i) en términos de la variable de interés y las variables de cortes, deben tener el mismo nombre en las diferentes ediciones, y los valores que toman cada variable de análisis deben coincidir, y (ii) en términos de los pesos y los pesos réplicas, el vector de pesos a nivel de estudiantes se llama `peso_MEst`, y los pesos réplicas se llaman `EST_W_REP_1`, `EST_W_REP_2`, etc. las futuras bases de datos de ARISTAS deben tener estos mismos nombres para poder ser analizadas directamente con las funciones del paquete.

## Análisis transversal de una edición

### Análisis transversal de una edición para una sola variable de corte

Para llevar a cabo estimaciones de una edición particular de ARISTAS con una sola variable de corte, el paquete cuenta con la siguiente función:

- `transversal`: Esta función calcula las estimaciones para una variable de interés numérica o categórica una edición específica de ARISTAS con **una variable de corte** o **sin variable de corte**.

El uso de esta función es:

```
transversal(data, y, x = NULL, limits = NULL, digit = 3, IC = TRUE, plot = NULL,
            descarga = FALSE, archivo = NULL, ...)
```

A continuación se explican en detalle los argumentos de esta función:

- `data`, se refiere a la base de datos que se quiere analizar, puede ser cualquiera de las bases de ARISTAS incluidas en el paquete o cualquiera otra base ARISTAS que no sea de uso público. Tenga en cuenta que las columnas que contengan los pesos réplica deben tener nombres: `EST_W_REP_1`, `EST_W_REP_2`, etc, también se debe incluir los pesos de estudiantes en la variable `peso_MEst`. Adicionalmente, los resultados de la función son con base a observaciones donde el peso no sea `NA`.

- `y`, vector de caracteres indicando el nombre de la variable de interés, ésta variable puede ser de tipo numérico (por ejemplo el desempeño en las pruebas) o categórico (por ejemplo el nivel de desempeño).

- `x`, este argumento es opcional, y se refiere al nombre de la variable de corte que puede ser numérica o categórica, la cual define los subgrupos para la comparación de `y`, por ejemplo, comparar el desempeño de una prueba de los hombres con el de las mujeres, o comparar el nivel de desempeño de las diferentes regiones del país.

- `limits`, este argumento es opcional y se refiere al vector de caracteres o de valores numéricos que definen cuáles son los subgrupos para la comparación de la variable `y`. En el caso de una variable numérica `x`, es obligatorio que el usuario provea los valores límites para definir los subgrupos. Por ejemplo, si `x` es la variable edad de los estudiantes, y el usuario provee los valores límites de 15 y 20, entonces se crearán tres subgrupos (intervalos): $(-\infty, 15]$, $(15, 20]$ y $(20, \infty)$, y se calculará la estimación e intervalo de confianza para la media de `y` para cada uno de los subgrupos, también se llevará a cabo la prueba de hipótesis de igual de medias para cada pareja de subgrupos, así como la comparación global de que las medias de `y` son iguales en todos los subgrupos. En el caso que `x` sea una variable categórica, los

valores límites son opcionales, por ejemplo, si `x` corresponde a la variable `regiones` (con 5 valores para las 5 regiones de Uruguay), entonces los datos se dividirá automáticamente en las regiones del país, y no es necesario el argumento `limits`. Por otro lado, si `x` es la variable región, y se provee los valores 1 y 3 en el argumento `limits`, entonces la función creará tres subgrupos: `region==1`, `region==3` y la unión de las demás regiones.

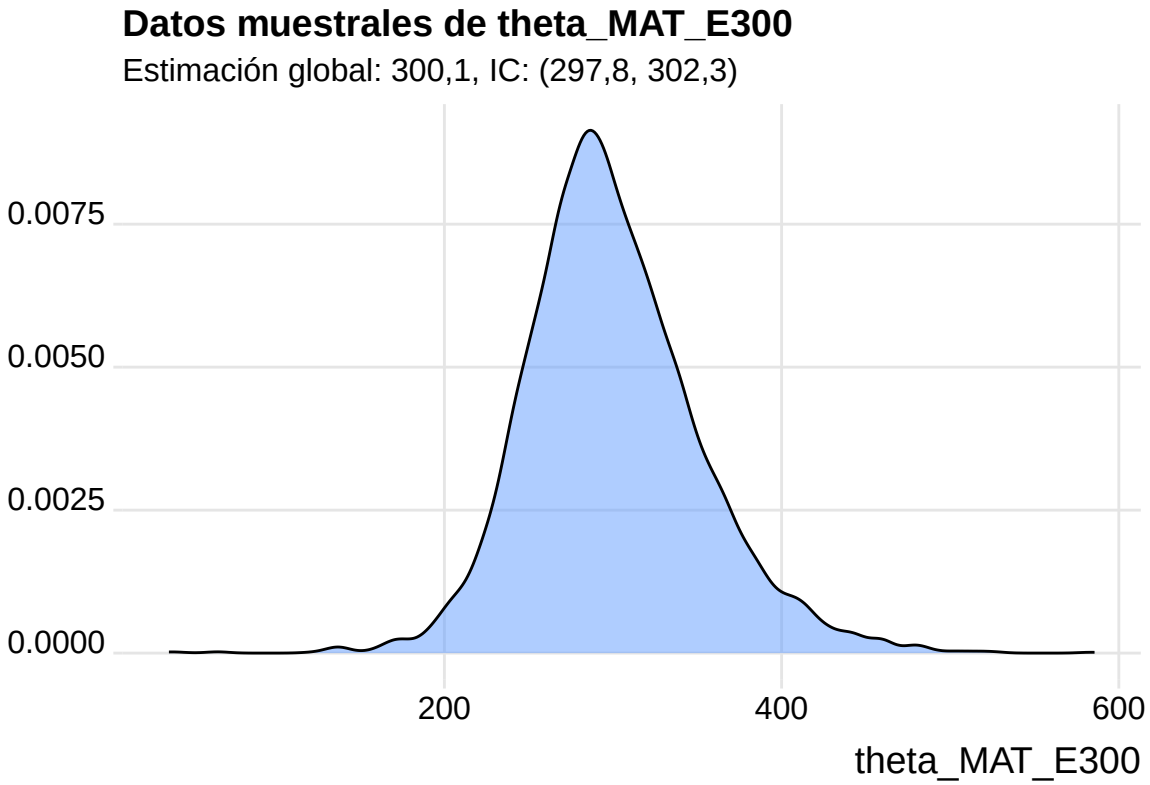
- `digit`, este argumento es opcional, y se refiere al número de dígitos decimales, por defecto son 3 dígitos decimales.
- `IC`, este argumento es opcional, y se refiere si se muestran las estimaciones e intervalos de confianza sobre la gráfica. Por defecto, `IC = TRUE`
- `plot`, este argumento es opcional, y se refiere al tipo de gráfica que se requiera. El usuario puede elegir entre `histogram`, `density`, `violin`, `Lollipop` para una variable de interés numérica, la opción `violin` incluirá también la gráfica de cajas. Para una variable categórica, el usuaio puede usar la opción `bar`. Si el usuario anteriormente no indicó una variable de corte `x`, la gráfica se realizará para todo el conjunto de datos; si el usuario indicó una variable de corte `x`, entonces se elaborará la gráfica para cada subgrupo y también para la conjunto completo de datos ingresados. De las anteriores gráficas, el histograma se elabora con `svyhist` del paquete `survey` teniendo en cuenta los pesos réplica, las gráficas de densidad, violin son resumenes de datos muestrales sin considerar los pesos, la gráfica de Lollipop muestra las estimaciones y los intervalos de confianza de la media de la variable de interés, la gráfica de barras para una variable `y` categórica muestra las estimaciones de los porcentajes para cada valor de `y`.
- `descarga`, este argumento es opcional, y se refiere si desea descargar los resultados de estimación, intervalo, prueba de hipótesis y la gráfica, por defecto, `descarga = FALSE`. Al usar la opción `descarga = TRUE` se descargará un archivo excel con los resultados numéricos y un pdf con la gráfica del mismo nombre que el archivo excel en la misma carpeta.
- `archivo`, si `descarga = TRUE`, el usuario debe indicar el nombre del archivo excel que se descarga. Si `plot = TRUE`, se descargará con el mismo nombre y extensión pdf en la misma ubicación que el archivo excel.
- `ancho` y `alto`, estos dos argumentos son opcionales, y se refieren al ancho y el alto en pulgadas de la gráfica. Si el usuario no especifica estos valores, la gráfica que se descarga tendrá la misma dimensión que la gráfica en el panel de gráficas.

### Ejemplo 1

A continuación se incluyen algunos ejemplos del uso de la función `transversal`, en primer lugar consideramos el análisis de una variable numérica `theta_MAT_E300`, cuando no se ingresa ninguna variable de corte.

```
transversal(data = ARISTAS18.matematicas, y = "theta_MAT_E300", plot = "density")
```

```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
```




En esta gráfica no se tienen en cuenta los pesos muestrales.

```
## $estimaciones
##   y X.Global.   var.y    SE   2.5 %  97.5 %    n n.NA   %.NA
## 1      Global 300,062 1,129 297,849 302,276 8845  956 10,808
```

La función arroja una estimación de 300.062 como el desempeño promedio en matemáticas a nivel nacional, con error estándar de 1.129, el intervalo de confianza de 95% de confianza está dado por (297.849, 302,276). El número de datos después de eliminar los registros con `NA` en los pesos `peso_MEst` es de 8845 estudiantes, de los cuales 956 corresponden a `NA` para la variable `theta_MAT_E300`. Es decir, los resultados de estimación son base en 7889 datos. En la gráfica se muestra la densidad de estos 7889 datos, sin considerar los pesos de estudiantes.

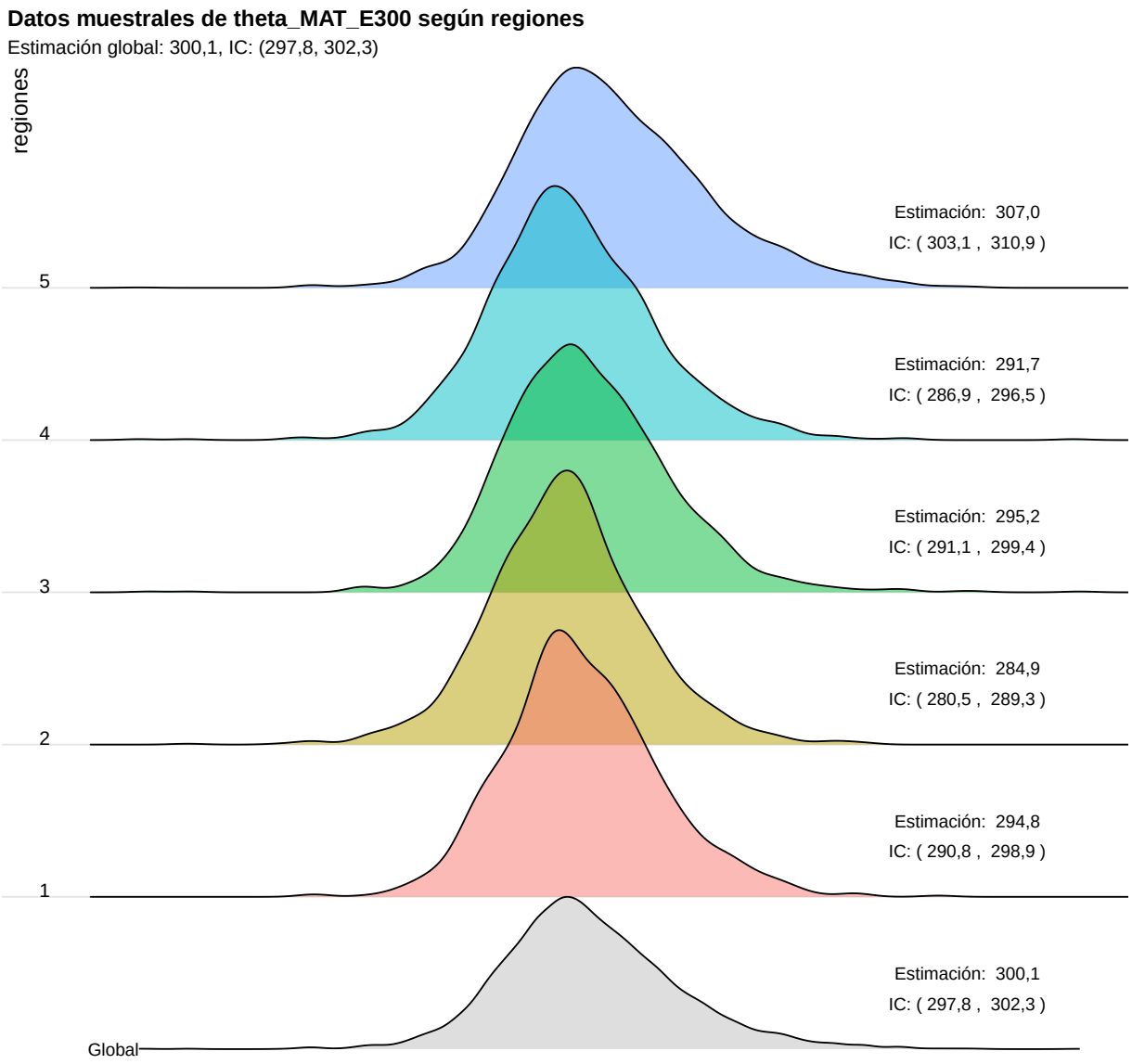
**Ejemplo 2**

Ahora, realizamos el mismo análisis para la variable `theta_MAT_E300` para las regiones del país. En primer lugar, la variable `regiones` toman valores de 0 a 5, por lo que se asigna el valor 0 como `NA`, y posteriormente se ejecuta la función `transversal`:

```
ARISTAS18.matematicas$regiones[ARISTAS18.matematicas$regiones == 0] <- NA
transversal(data=ARISTAS18.matematicas, y = "theta_MAT_E300", x = "regiones", plot = "density")
```

```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
```

```
## Picking joint bandwidth of 9.41
```




En esta gráfica no se tienen en cuenta los pesos muestrales.

```
## $estimaciones
##   regiones theta_MAT_E300    se   2.5 %  97.5 %    n n.NA   %.NA
## 1        1        294,845 2,086 290,756 298,934 1069  132 12,348
## 2        2        284,933 2,246 280,530 289,335 1537  173 11,256
## 3        3        295,242 2,137 291,054 299,431 1573  174 11,062
## 4        4        291,725 2,440 286,944 296,507 1407  182 12,935
## 5        5        306,959 1,993 303,052 310,865 3259  295  9,052
## 6   Global        300,062 1,129 297,849 302,276 8845  956 10,808
##
## $contraste.individual
##    Categoria1 Categoria2 Estadistica ValorP signif
## 1           2          5       6,717  0,000    ***
## 2           2          3       3,163  0,003     **
## 3           2          1      -3,162  0,003     **
```

```
## 4           2          4       2,009  0,051      +
## 5           5          3       4,005  0,000    ***
## 6           5          1       4,119  0,000    ***
## 7           5          4       4,457  0,000    ***
## 8           3          1       0,128  0,899
## 9           3          4      -1,049  0,300
## 10          1          4      -1,016  0,316
##
## $contraste.global
## [1] "Estadística F con 4 y 78 grados de libertad: 11,875, valor p: 1,39e-07"
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

Vemos que en la salida de la función, en su primer resultado (denotados con `$estimaciones`), se muestran las estimaciones por región y a nivel global, evidenciando que la región con mejor desempeño es la región 5, mientras que la de peor desempeño es la región 2. En cuanto a la cantidad de datos y de datos faltantes, vemos que: sin considerar los datos con `NA` en los pesos, la región 5 es la región con mayor cantidad de estudiantes evaluados con 3259 estudiantes y solo 9.052% de no respuesta, mientras que la región 1 es la región con menos estudiantes con 1069 estudiantes donde 12.348% de ellos no registran información en `theta_MAT_E300`. Las estimaciones correspondientes al `Global` coinciden con los resultados mostrados anterioremente, cuando no se ingresó ninguna variable de corte.

Los anteriores resultados fueron obtenidos de esta forma: primero se especifica el diseño muestral con pesos réplica usando la función `svrepdesign` del paquete `survey`, así:

```
dis <- svrepdesign(data=datos, type="Fay", weights=~peso_MEst,
                   repweights=paste0("EST_W_REP_[", "1-", length(pesos), "]"),
                   combined.weights=TRUE, rho = 1.5, mse = F)
```

y posteriormente se calculan las estimaciones del desempeño promedio con la función `svymean` así: `svymean(~var.y, dis, na.rm = T)`.

En el siguiente resultado que muestra la función (denotados con `$contraste.individual`), se muestran los resultados de prueba de hipótesis para la igualdad de desempeño promedio entre cada pareja de regiones, la salida incluye el valor de la estadística de prueba $t$, el valor $p$ y la marca de significancia (*** para valor $p$ entre 0 y 0.001; ** para valor $p$ entre 0.001 y 0.01; * para valor $p$ entre 0.01 y 0.05; + para valor $p$ entre 0.05 y 0.1, y sin marca para valor $p$ mayor al 0.1.). Podemos ver que, por ejemplo, no hay diferencia significativa entre la región 1 y la región 4 en cuanto al desempeño en matemáticas, pero sí hay diferencia significativa entre la región 2 y la región 5. Estas pruebas de hipótesis se realizan por medio de la función `svyttest` así: `svyttest(var.y ~ as.character(var.x1), dis1)` donde `dis1` es similar al diseño muestral `dis` presentado anteriormente, pero limitado a los datos de las regiones involucrados en la prueba de hipótesis.

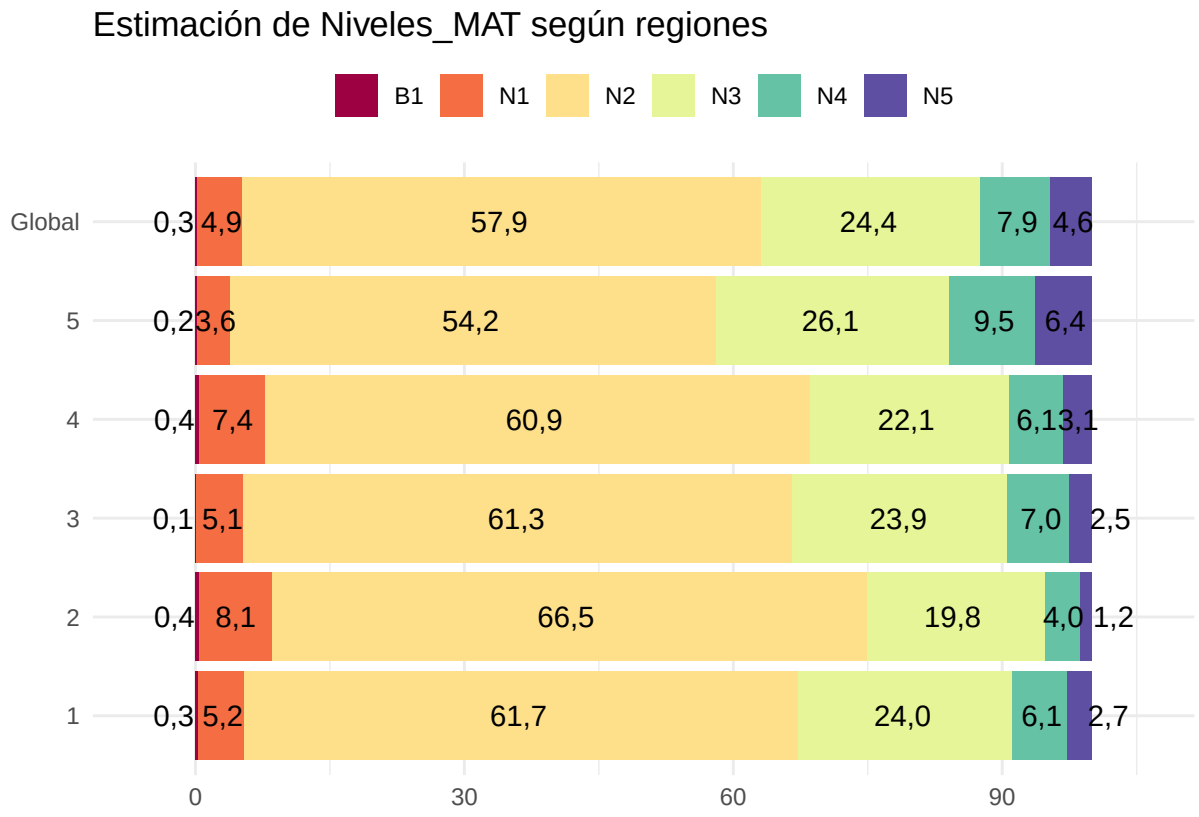
En el siguiente y último resultado que muestra la función (denotados con `$contraste.global`), podemos ver los resultados de la prueba de hipótesis donde la hipótesis nula es: el desempeño promedio es lo mismo en todas las regiones, frente a la hipótesis alterna de que existen al menos dos regiones que difieren en el desempeño promedio. Vemos que la estadística de prueba corresponde a una estadística $F$ con 4 grados de libertad en el numerador y 48 grados en el denominador, cuyo valor es de 11.875, que con un valor $p$ asociado de 1.39e-07 conduce a la conclusión de que existe diferencia en el desempeño promedio en algunas de las 5 regiones. Estos resultados fueron obtenidos así: primero se especifica un modelo de regresión donde la variable respuesta es la variable de interés `theta_MAT_E300` y la variable regresora es la variable de corte `regiones`, esto se logra con la función `svyglm` así: `model <- svyglm(var.y ~ var.x1, design=dis, family=gaussian())`, y posteriormente se lleva a cabo la prueba de hipótesis con la función `regTermTest` del paquete `survey` así: `regTermTest(model, ~ var.x1)`.

**Ejemplo 3**

Como un último ejemplo, se analizará una variable categórica, nivel de desempeño en la prueba de matemáticas `Niveles_MAT` según región. El análisis se puede realizar de la siguiente manera:

```
transversal(data=ARISTAS18.matematicas, y = "Niveles_MAT", x = "regiones", plot  = "bar")
```

```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
```




```
## $estimaciones
##    regiones Niveles_MAT porcentaje    se  2.5 % 97.5 %    n n.NA   %.NA
## 1         1          B1      0,271 0,185 -0,092  0,634 1069  132 12,348
## 2         1          N1      5,190 0,512  4,186  6,194 1069  132 12,348
## 3         1          N2     61,724 2,283 57,248 66,199 1069  132 12,348
## 4         1          N3     23,968 2,021 20,007 27,929 1069  132 12,348
## 5         1          N4      6,144 0,700  4,772  7,516 1069  132 12,348
## 6         1          N5      2,703 0,457  1,808  3,598 1069  132 12,348
## 7         2          B1      0,403 0,138  0,132  0,674 1537  173 11,256
## 8         2          N1      8,108 0,980  6,188 10,028 1537  173 11,256
## 9         2          N2     66,453 1,529 63,457 69,449 1537  173 11,256
## 10        2          N3     19,819 1,532 16,815 22,822 1537  173 11,256
## 11        2          N4      3,980 0,855  2,304  5,656 1537  173 11,256
## 12        2          N5      1,237 0,299  0,651  1,823 1537  173 11,256
## 13        3          B1      0,147 0,108 -0,065  0,359 1573  174 11,062
## 14        3          N1      5,146 0,774  3,629  6,663 1573  174 11,062
```

```
## 15        3          N2     61,315 1,912 57,568 65,062 1573  174 11,062
## 16        3          N3     23,928 1,356 21,270 26,587 1573  174 11,062
## 17        3          N4      6,952 0,605  5,766  8,139 1573  174 11,062
## 18        3          N5      2,512 0,661  1,217  3,807 1573  174 11,062
## 19        4          B1      0,393 0,097  0,204  0,583 1407  182 12,935
## 20        4          N1      7,385 1,143  5,145  9,625 1407  182 12,935
## 21        4          N2     60,856 1,494 57,927 63,785 1407  182 12,935
## 22        4          N3     22,104 1,271 19,614 24,594 1407  182 12,935
## 23        4          N4      6,131 0,800  4,563  7,698 1407  182 12,935
## 24        4          N5      3,131 0,720  1,720  4,542 1407  182 12,935
## 25        5          B1      0,230 0,096  0,041  0,418 3259  295  9,052
## 26        5          N1      3,605 0,395  2,832  4,379 3259  295  9,052
## 27        5          N2     54,190 1,471 51,306 57,074 3259  295  9,052
## 28        5          N3     26,099 0,894 24,346 27,852 3259  295  9,052
## 29        5          N4      9,520 0,626  8,293 10,748 3259  295  9,052
## 30        5          N5      6,356 0,832  4,726  7,986 3259  295  9,052
## 31   Global          B1      0,259 0,065  0,131  0,387 8845  956 10,808
## 32   Global          N1      4,918 0,315  4,302  5,534 8845  956 10,808
## 33   Global          N2     57,949 0,850 56,282 59,616 8845  956 10,808
## 34   Global          N3     24,417 0,555 23,329 25,505 8845  956 10,808
## 35   Global          N4      7,880 0,378  7,140  8,621 8845  956 10,808
## 36   Global          N5      4,576 0,468  3,658  5,495 8845  956 10,808
##
## $contraste.individual
##    Categoria1 Categoria2 Estadistica ValorP signif
## 1           2          5     124,132  0,000    ***
## 2           2          3      39,868  0,001    ***
## 3           2          1      24,540  0,002     **
## 4           2          4      24,958  0,009     **
## 5           5          3      42,332  0,000    ***
## 6           5          1      16,621  0,000    ***
## 7           5          4      53,186  0,000    ***
## 8           3          1       0,953  0,973
## 9           3          4      10,572  0,298
## 10          1          4       4,994  0,497
##
## $contraste.global
## [1] "Estadística Chi-cuadrado de Pearson: 230,507 con df = 20 , valor p: 3,345e-27"
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

En la salida de la función, en su primer resultado (denotados con `$estimaciones`), se muestran las estimaciones, error estándar y los intervalos de confianza para el porcentaje de estudiantes en cada nivel de desempeño según región y también de forma global. Vemos por ejemplo, que la región con mayor porcentaje de estudiantes en el nivel N5 es la región 5 con 6.356% de estudiantes ubicados en este nivel, mientras que la región con más porcentaje de estudiantes en el nivel B1 corresponde a la región 2 con un 0.403% de estudiantes en ese nivel. En cuanto al conteo de casos y de datos faltantes, los resultados son idénticos a los mostrados en el ejemplo 2. Las estimacions se obtuvieron de la misma según descrita en el ejemplo 2, es decir, usando funciones `svymean` combinando con un diseño muestral con réplicas especificado con la función `svrepdesign`.

En el segundo resultado de la salida de la función (denotados con `$contraste.individual`), se encuentran los resultados de la prueba de hipótesis para la comparación del vector de porcentajes para cada pareja de regiones. Por ejemplo, para la primera fila donde comparación la región 2 con la región 5, la hipótesis nula es: los porcentajes de estudiantes en los diferentes niveles de desempeño son iguales en la región 2 y la región 5.

A continuación se muestra el valor de la estadística Chi-cuadrado, el valor $p$ y la marca de significancia. De esta forma, podemos ver que sí existe diferencia significativa entre la región 2 y la región 5, mientras que no existe diferencia significativa entre la región 3 y la región 1. Los anteriores resultados se obtuvieron haciendo uso de la función `svytable` así: `svytable(~var.y+var.x2, dis1)` donde `dis1` es el diseño muestral con pesos réplica restringido a las regiones correspondientes a cada prueba.

En el tercer y último resultado de la salida de la función (denotados con `$contraste.global`), encontramos los resultados de prueba de hipótesis donde compara los porcentajes de estudiantes en todas las regiones, esto es, la hipótesis nula es: los porcentajes de estudiantes en los diferentes niveles de desempeño son iguales en todas las regiones, frente a la hipótesis alterna de que existen al menos dos regiones donde se difieren en cuanto al porcentaje de estudiantes en los niveles de desempeño. El valor de la estadística de prueba Chi-cuadrado y el valor $p$ indican fuertes evidencias en los datos en contra de la hipótesis nula.

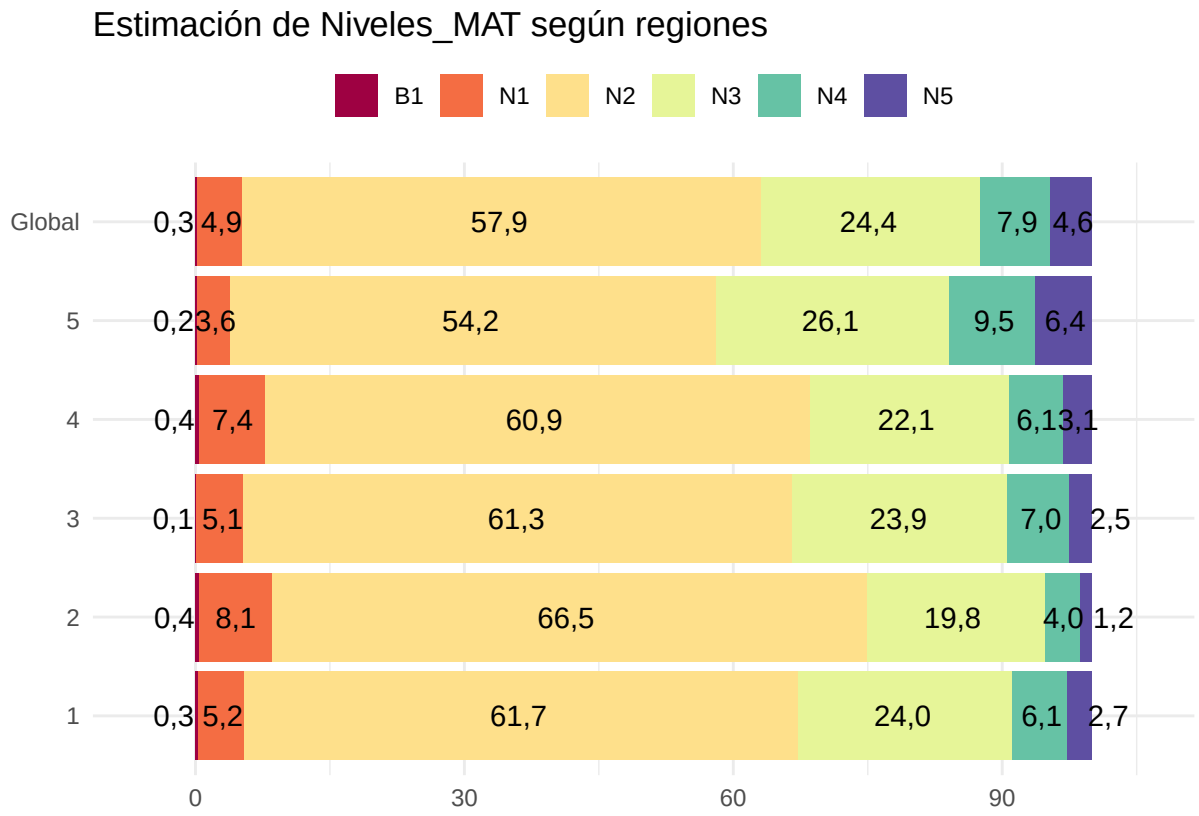
Como una última ilustración, si se ejecuta el siguiente comando:

```
transversal(data=ARISTAS18.matematicas, y = "Niveles_MAT", x = "regiones", plot  = "bar",
            descarga = TRUE, archivo = "Niveles_MAT.xlsx", ancho = 10, alto = 8)
```

```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
```

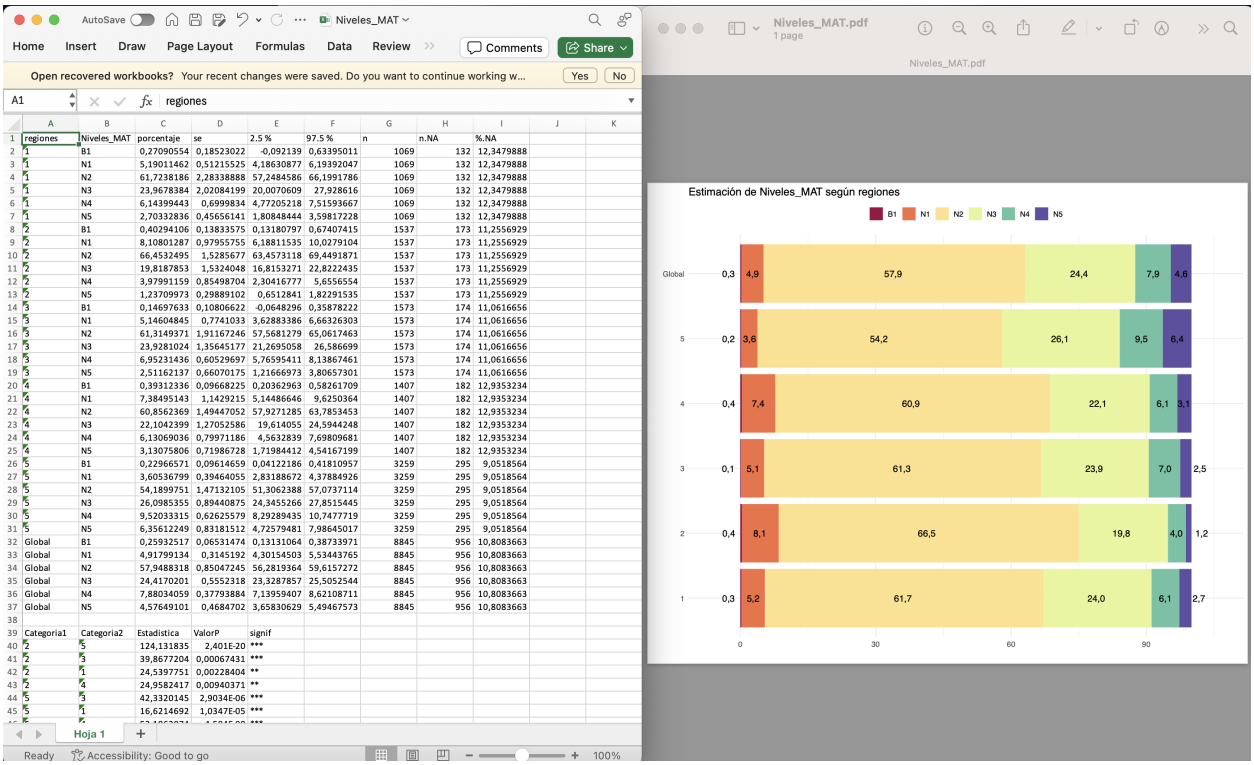
```
## $estimaciones
##    regiones Niveles_MAT porcentaje    se  2.5 % 97.5 %    n n.NA   %.NA
## 1         1          B1      0,271 0,185 -0,092  0,634 1069  132 12,348
## 2         1          N1      5,190 0,512  4,186  6,194 1069  132 12,348
## 3         1          N2     61,724 2,283 57,248 66,199 1069  132 12,348
## 4         1          N3     23,968 2,021 20,007 27,929 1069  132 12,348
## 5         1          N4      6,144 0,700  4,772  7,516 1069  132 12,348
## 6         1          N5      2,703 0,457  1,808  3,598 1069  132 12,348
## 7         2          B1      0,403 0,138  0,132  0,674 1537  173 11,256
## 8         2          N1      8,108 0,980  6,188 10,028 1537  173 11,256
## 9         2          N2     66,453 1,529 63,457 69,449 1537  173 11,256
## 10        2          N3     19,819 1,532 16,815 22,822 1537  173 11,256
## 11        2          N4      3,980 0,855  2,304  5,656 1537  173 11,256
## 12        2          N5      1,237 0,299  0,651  1,823 1537  173 11,256
## 13        3          B1      0,147 0,108 -0,065  0,359 1573  174 11,062
## 14        3          N1      5,146 0,774  3,629  6,663 1573  174 11,062
## 15        3          N2     61,315 1,912 57,568 65,062 1573  174 11,062
## 16        3          N3     23,928 1,356 21,270 26,587 1573  174 11,062
## 17        3          N4      6,952 0,605  5,766  8,139 1573  174 11,062
## 18        3          N5      2,512 0,661  1,217  3,807 1573  174 11,062
## 19        4          B1      0,393 0,097  0,204  0,583 1407  182 12,935
## 20        4          N1      7,385 1,143  5,145  9,625 1407  182 12,935
## 21        4          N2     60,856 1,494 57,927 63,785 1407  182 12,935
## 22        4          N3     22,104 1,271 19,614 24,594 1407  182 12,935
## 23        4          N4      6,131 0,800  4,563  7,698 1407  182 12,935
## 24        4          N5      3,131 0,720  1,720  4,542 1407  182 12,935
## 25        5          B1      0,230 0,096  0,041  0,418 3259  295  9,052
## 26        5          N1      3,605 0,395  2,832  4,379 3259  295  9,052
## 27        5          N2     54,190 1,471 51,306 57,074 3259  295  9,052
## 28        5          N3     26,099 0,894 24,346 27,852 3259  295  9,052
## 29        5          N4      9,520 0,626  8,293 10,748 3259  295  9,052
## 30        5          N5      6,356 0,832  4,726  7,986 3259  295  9,052
```

```
## 31   Global          B1      0,259 0,065  0,131  0,387 8845  956 10,808
## 32   Global          N1      4,918 0,315  4,302  5,534 8845  956 10,808
## 33   Global          N2     57,949 0,850 56,282 59,616 8845  956 10,808
## 34   Global          N3     24,417 0,555 23,329 25,505 8845  956 10,808
## 35   Global          N4      7,880 0,378  7,140  8,621 8845  956 10,808
## 36   Global          N5      4,576 0,468  3,658  5,495 8845  956 10,808
##
## $contraste.individual
##    Categoria1 Categoria2 Estadistica ValorP signif
## 1           2          5     124,132  0,000    ***
## 2           2          3      39,868  0,001    ***
## 3           2          1      24,540  0,002     **
## 4           2          4      24,958  0,009     **
## 5           5          3      42,332  0,000    ***
## 6           5          1      16,621  0,000    ***
## 7           5          4      53,186  0,000    ***
## 8           3          1       0,953  0,973
## 9           3          4      10,572  0,298
## 10          1          4       4,994  0,497
##
## $contraste.global
## [1] "Estadística Chi-cuadrado de Pearson: 230,507 con df = 20 , valor p: 3,345e-27"
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```




```
## Los resultados se han exportado a la ruta suministrada.
```

En la carpeta correspondiente al Working Directory, se creará un archivo Excel de nombre `Niveles_MAT.xlsx` y un archivo pdf de nombre `Niveles_MAT.pdf`, en la siguiente foto se encuentra la imagen de estos dos archivos creados.




**Análisis transversal de una edición para más de una variable de corte**

Para llevar a cabo estimaciones de una edición particular de ARISTAS con más de una variable de corte, el paquete cuenta con la siguiente función:

- `transversal.multiple`: Esta función calcula las estimaciones de una variable de interés numérica o categórica para una edición específica de ARISTAS para **más de una variable de corte**.

El uso de esta función es:

```
transversal.multiple(data, y, x, IC = TRUE, digit = 3, plot = NULL, descarga = FALSE,
                     archivo = NULL, ...)
```

Los argumentos de `data`, `y`, `digit`, `IC`, `plot`, `descarga`, `archivo`, `ancho` y `alto` son idénticos a los de la función `transversal`. El argumento `x` debe ser un vector de caracteres indicando el nombre de las variables de corte, estas variable deben ser de tipo categórico. En caso de que necesite comparar la media de `y` para diferentes subgrupos definidos por una variable numérica `x`, por ejemplo, comparar el desempeño para estudiantes de diferentes grupos de edad, se debe crear la variable categórica definiendo los rangos de edad, y posteriormente utilizar esta función.

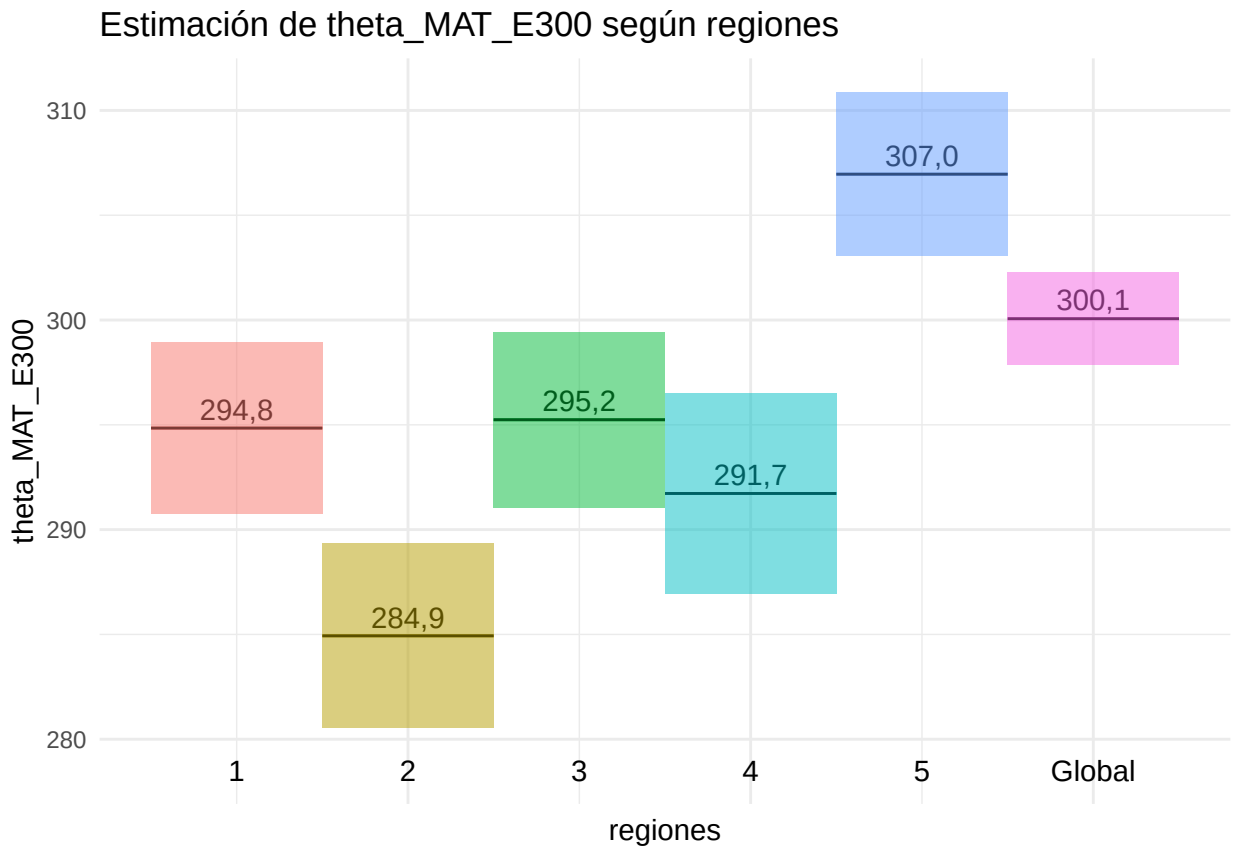
Esta función hace uso de la función `transversal`, por lo que los detalles teóricos de cómo se realizan cálculos son idénticos.

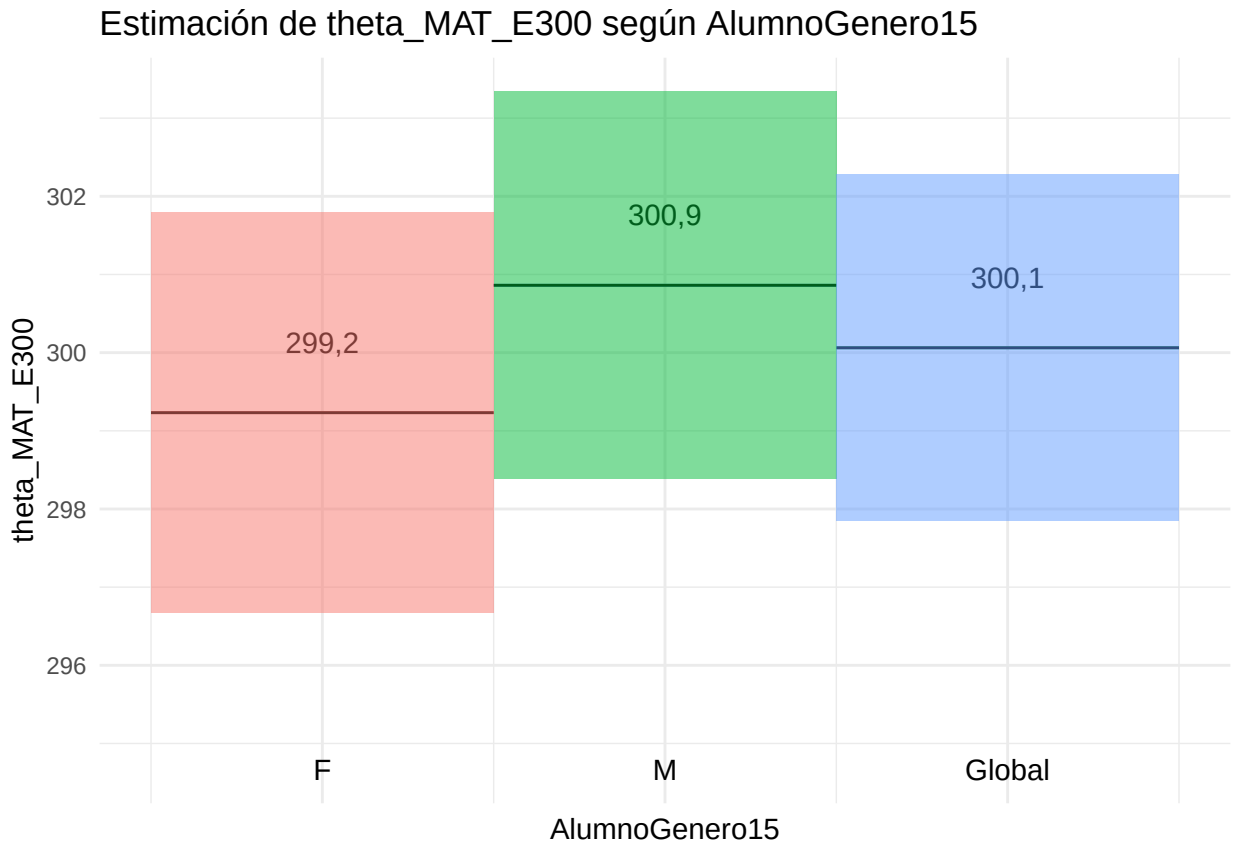
A continuación se muestran algunos ejemplos sobre el uso de esta función:

**Ejemplo 4**

Se presenta el análisis de la variable numérica `theta_MAT_E300` para dos variables de corte `regiones` y `AlumnoGenero15`, donde se pide la gráfica de Lollipop.

```
transversal.multiple(data = ARISTAS18.matematicas, y = "theta_MAT_E300",
                     x = c("regiones","AlumnoGenero15"), plot = "Lollipop")
```






```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
## $estimaciones
## $estimaciones$regiones
##   regiones theta_MAT_E300    se   2.5 %  97.5 %    n n.NA   %.NA
## 1        1        294,845 2,086 290,756 298,934 1069  132 12,348
## 2        2        284,933 2,246 280,530 289,335 1537  173 11,256
## 3        3        295,242 2,137 291,054 299,431 1573  174 11,062
## 4        4        291,725 2,440 286,944 296,507 1407  182 12,935
## 5        5        306,959 1,993 303,052 310,865 3259  295  9,052
## 6   Global        300,062 1,129 297,849 302,276 8845  956 10,808
##
## $estimaciones$AlumnoGenero15
##   AlumnoGenero15 theta_MAT_E300    se   2.5 %  97.5 %    n n.NA   %.NA
## 1              F        299,232 1,308 296,669 301,795 4317  456 10,563
## 2              M        300,862 1,264 298,385 303,339 4528  500 11,042
## 3         Global        300,062 1,129 297,849 302,276 8845  956 10,808
##
##
## $contrastes_individuales
## $contrastes_individuales$regiones
##    Categoria1 Categoria2 Estadistica ValorP signif
## 1           2          5       6,717  0,000    ***
## 2           2          3       3,163  0,003     **
## 3           2          1      -3,162  0,003     **
```

```
## 4           2          4       2,009  0,051      +
## 5           5          3       4,005  0,000    ***
## 6           5          1       4,119  0,000    ***
## 7           5          4       4,457  0,000    ***
## 8           3          1       0,128  0,899
## 9           3          4      -1,049  0,300
## 10          1          4      -1,016  0,316
##
##
## $contraste_global
## $contraste_global$regiones
## [1] "Estadística F con 4 y 78 grados de libertad: 11,875, valor p: 1,39e-07"
##
## $contraste_global$AlumnoGenero15
## [1] "Estadística F con 1 y 81 grados de libertad: 1,748, valor p: 0,19"
```

Esta función muestra en su primer resultado, `$estimaciones`, todos los resultados de `$estimaciones` de la función `transversal` con cada una de las dos variables de corte: `regiones` y `AlumnoGenero15`, estos resultados se identifican con los nombres de `$estimaciones$regiones` y `$estimaciones$AlumnoGenero15`.

Posteriormente, para cada variable de corte, realiza la prueba de hipótesis de igualdad de medias para cada pareja de valores. Estos resultados se identifican con el nombre `$contrastes_individuales`, los resultados de comparación para la variable de corte `regiones` se identifican con el nombre `$contrastes_individuales$regiones`. Nótese que no existe el resultado `$contrastes_individuales$AlumnoGenero15`, ya que la variable de corte `AlumnoGenero15` solo tiene dos valores, por lo que la comparación individual coincide con la comparación global.

Finalmente, la función arroja el resultado `$contraste_global` con los resultados de prueba de hipótesis donde para cada variable de corte se juzga la hipótesis de la igualdad de media de la variable `theta_MAT_E300` para todos los niveles. Los resultados se identifican con los nombres de `$contraste_global$regiones` y `$contraste_global$AlumnoGenero15`.

## Unión de bases para la comparación entre ediciones

Con el fin de alistar las bases de datos para la comparación entre diferentes ediciones, el paquete cuenta con la función `unionbases.longitudinal`. El uso de la función es:

```
unionbases.longitudinal(ediciones, bases, y, x = NULL)
```

donde:

- `ediciones` es el vector que indica para cuáles ediciones se quiere unir las bases de datos.
- `datos` es el vector con el nombre de las bases que se quiere unir, este debe tener el mismo orden del parámetro `ediciones`.
- `y` es el nombre de la variable de interés `y`.
- `x` es un argumento opcional donde se indica el nombre de una o más variables de corte `x`.

Para usar esta función, el usuario debe asegurar que el nombre de las variables `x`, `y`, el vector de pesos de estudiantes y los pesos réplica de las bases de datos de las diferentes ediciones sean idénticos. El número de filas de la base de datos que resulte de aplicar la función `unionbases.longitudinal` será igual a la sumatoria entre el número de filas de las bases de las diferentes ediciones. Es decir, las bases de diferentes ediciones se organizan una debajo de la otra coincidiendo en las variables que tienen en común, estos son, la variable de interés `y`, la variable de corte `x`, los pesos de estudiantes, `peso_MEst` y los pesos réplica empezando

los nombres con `EST_W_REP_1`, `EST_W_REP_2`, etc, adicionalmente, la base de datos consolidad tendrá una columna llamada `edicion` indicando a cuál edición corresponde cada fila de la base. Adicionalmente, las variables de diferentes ediciones deben estar codificados de la misma forma.

A continuación se ilustra la forma de usar esta función para unir la bases `ARISTAS18.matematicas` y `ARISTAS22.matematicas` tomando en primer lugar la variable de interés el desempeño en matemáticas y variable de corte `regiones` y `AlumnoGenero15`.

En primer lugar, teniendo en cuenta que la variable de corte `regioines` no tienen la misma codificación entre la edición 2018 y 2022, se procede a unificar la codificación.

```
# Convertir las variables numéricas a categóricas
ARISTAS18.matematicas$regiones <- as.character(ARISTAS18.matematicas$regiones)
# Unificar las categorías de ARISTAS 2018 según las categorías de 2022
ARISTAS18.matematicas2 <- ARISTAS18.matematicas |>
  dplyr::mutate(regiones = dplyr::case_when(
    regiones == 1 ~ "SUR",
    regiones == 2 ~ "OESTE",
    regiones == 3 ~ "NORTE",
    regiones == 4 ~ "ESTE",
    regiones == 5 ~ "CENTRO",
    TRUE ~ NA_character_  # Para asegurarse de que otros valores no definidos se traten como NA
    )
  )
```

En segundo lugar, el nombre de la variable de interés es diferente en las diferentes ediciones, en el 2018 tiene el nombre de `theta_MAT_E300`, mientras que en la edición 2022 se llama `theta_MAT_300_50`, por lo que el paso a seguir es cambiar el nombre de esta variable de 2018 para que éstea coincida con la base 2022.

```
# Unificar el nombre de la variable theta_MAT_300_50
ARISTAS18.matematicas2 <- ARISTAS18.matematicas2 |>
  dplyr::rename(theta_MAT_300_50 = theta_MAT_E300,
                AlumnoGenero = AlumnoGenero15)
```

Finalmente, se utiliza el siguiente comando para unir la base de 2022 con la de 2018.

```
ARISTAS18.22.matematicas <- unionbases.longitudinal(ediciones = c(2018, 2022),
                     bases = c("ARISTAS18.matematicas2", "ARISTAS22.matematicas"),
                     y = "theta_MAT_300_50", x = "regiones")
dim(ARISTAS18.22.matematicas)
```

```
## [1] 20606   164
```

```
dim(ARISTAS18.matematicas2)
```

```
## [1] 9060  402
```

```
dim(ARISTAS22.matematicas)
```

```
## [1] 11546   415
```

La base que resulte de la operación `ARISTAS18.22.matematicas` tiene en total 20606 filas, igual al número de filas de `ARISTAS18.matematicas2` más el número de filas de `ARISTAS22.matematicas`; y tiene 164 columnas, correspondiendo a: `theta_MAT_300_50`, `regiones`, `ediciones`, `peso_MEst`, y 160 pesos réplica.

Finalmente, esta función `unionbases.longitudinal` también permite crear la unión de bases con más de una variable de corte. El comando cuando tenemos las variables de corte `regiones` y `AlumnoGenero` es como sigue:

```
ARISTAS18.22.matematicas <- unionbases.longitudinal(ediciones = c(2018, 2022),
                     bases = c("ARISTAS18.matematicas2", "ARISTAS22.matematicas"),
                     y = "theta_MAT_300_50", x = c("regiones", "AlumnoGenero"))
dim(ARISTAS18.22.matematicas)
```

```
## [1] 20606   165
```

## Análisis longitudinal de una edición

### Análisis longitudinal para una sola variable de corte

Para llevar a cabo la comparación de una variable de interés para diferentes ediciones de ARISTAS con una sola variable de corte o sin variable de corte, el paquete cuenta con la siguiente función:

- `longitudinal`: Esta función compara la media de una variable de interés numérica o categórica entre las diferentes ediciones de ARISTAS con una variable de corte o sin variable de corte.

El uso de esta función es:

```
longitudinal(datos, ediciones, y, x = NULL, digit = 3, plot = "density",
             descarga = FALSE, archivo = NULL, ...)
```

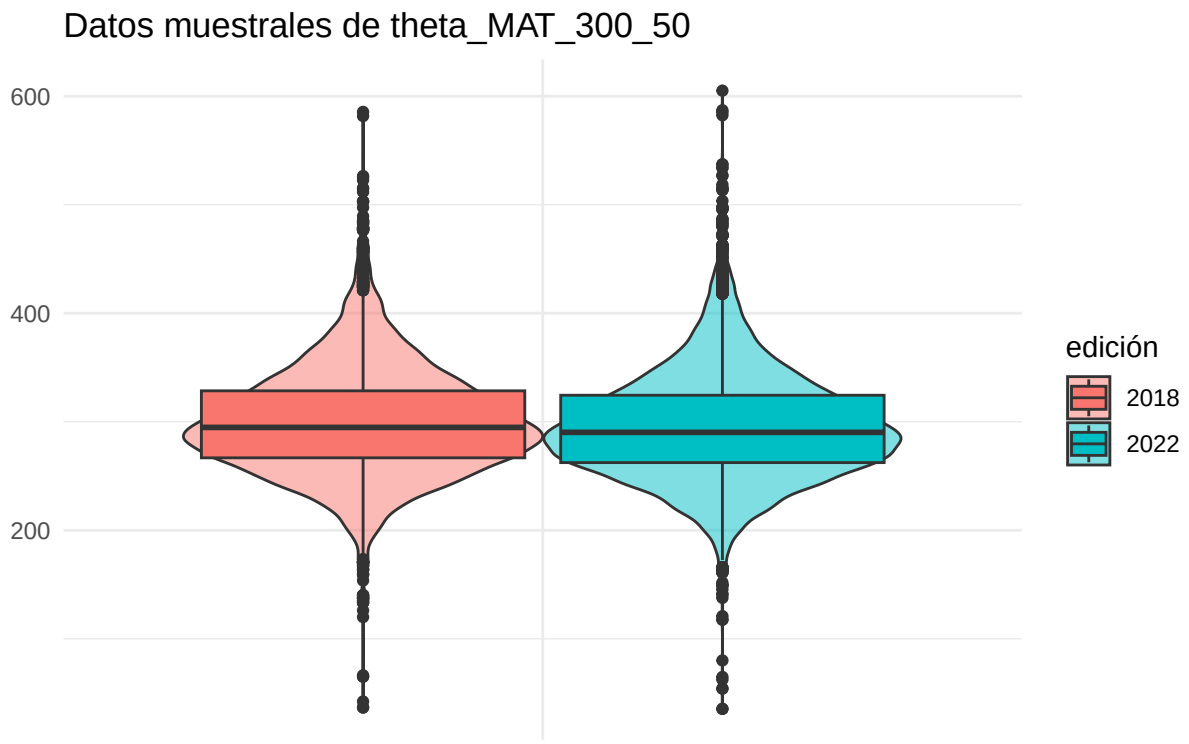
En cuanto a los argumentos de la función, los argumentos `y`, `x`, `digit`, `plot`, `descarga`, `archivo`, `ancho` y `alto` se utilizan de la forma forma que la función `transversal`. El argumento adicional `ediciones` corresponde a un vector de caracteres o números indicando cuáles son las ediciones que desea comparar. En cuanto a la base de datos ingresado en el argumento `datos` La base de datos debe estar organizadas así: las datos de diferentes ediciones deben estar una debajo de otra coincidiendo en las variables que tienen en común, estos son, la variable de interés `y`, la variable de corte `x`, los pesos de estudiantes, `peso_MEst` y los pesos réplica empezando los nombres con `EST_W_REP_1`, `EST_W_REP_2`, etc, adicionalmente, la base de datos debe tener una columna llamada `edicion` indicando a cuál edición corresponde cada fila de la base. La consolidación de la base de datos de 2 o más versiones se puede lograr con la función `unionbases.longitudinal`, explicada anteriormente.

**Ejemplo 5**

A continuación, se presenta el análisis de una variable de interés numérica, `theta_MAT_300_50`, para las ediciones de 2018 y 2022, sin variable de corte. El siguiente comando se aplica para la base de datos `ARISTAS18.22.matematicas` que fue creado anteriormente por medio de la función `unionbases.longitudinal`

```
longitudinal(datos = ARISTAS18.22.matematicas, ediciones = c(2018, 2022),
             y = "theta_MAT_300_50", digit = 3, plot = "violin")
```

```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
```




En esta gráfica no se tienen en cuenta los pesos muestrales.

```
## $estimaciones
##   edicion theta_MAT_300_50    se   2.5 %  97.5 %     n n.NA   %.NA
## 1    2018          300,062 1,129 297,849 302,276  8845  956 10,808
## 2    2022          297,156 1,174 294,856 299,456 11546 2007 17,383
##
## $contraste.global
## [1] "Estadística F para comparación de theta_MAT_300_50 entre las 2 ediciones: 3,76 con 1 y 81 grados
```

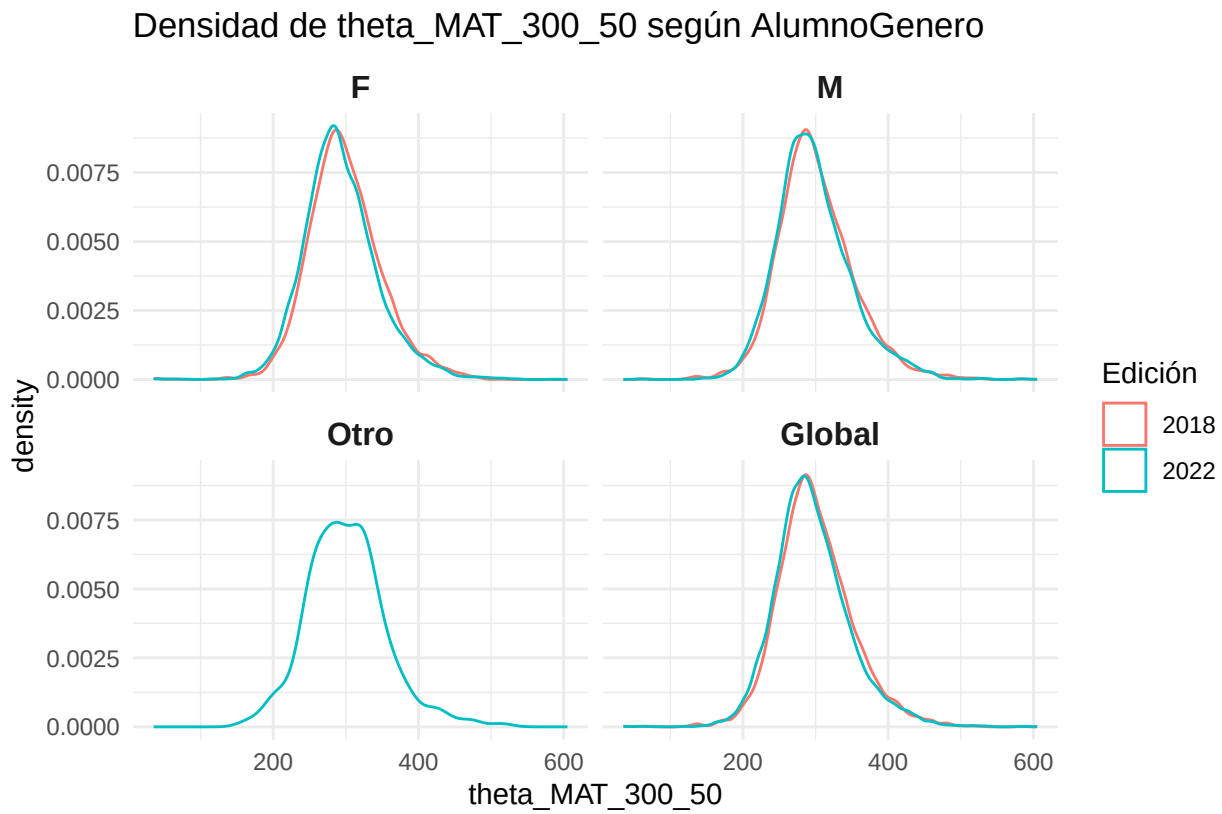
Al no ingresar ninguna variable de corte `x` la función en primer lugar calcula la estimación e intervalo de confianza para la media de la variable de interés `y` para cada edición (resultado en `$estimaciones`), y posteriormente en el resultado `$contraste.global` lleva a cabo la prueba de hipótesis de igualdad de medias para las diferentes ediciones. Las funciones internas para el cálculo son tomadas del paquete `survey`, las cuales fueron explicadas en la función `transversal`. También la función produce la gráfica de violin junto con la gráfica de cajas para cada edición.

**Ejemplo 6**

A continuación, se presenta el análisis de una variable de interés numérica, `theta_MAT_300_50`, para las ediciones de 2018 y 2022, con variable de corte `AlumnoGenero`. El siguiente comando se aplica para la base de datos `ARISTAS18.22.matematicas` que fue creado anteriormente por medio de la función `unionbases.longitudinal`

```
longitudinal(datos = ARISTAS18.22.matematicas, ediciones = c(2018, 2022),
             y = "theta_MAT_300_50", x = "AlumnoGenero", digit = 2,
             plot = "density")
```

```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
```




En esta gráfica no se tienen en cuenta los pesos muestrales.

```
## $estimaciones
##   edicion var.x  var.y   se  2.5 % 97.5 %    n n.NA  %.NA
## 1    2018     F 299,23 1,31 296,67 301,79 4317  456 10,56
## 2    2022     F 294,40 1,44 291,57 297,23 5482  892 16,27
## 3    2018     M 300,86 1,26 298,38 303,34 4528  500 11,04
## 4    2022     M 299,38 1,13 297,17 301,58 5603  978 17,45
## 5    2022  Otro 304,51 4,79 295,11 313,90  263   32 12,17
##
## $estimaciones.globales
##   edicion theta_MAT_300_50   se  2.5 % 97.5 %     n n.NA  %.NA
## 1    2018           300,06 1,13 297,85 302,28  8845  956 10,81
## 2    2022           297,08 1,18 294,78 299,39 11348 1902 16,76
##
## $contraste.global
## [1] "Estadística F para comparación de theta_MAT_300_50 entre las 2 ediciones: 4 con 1 y 81 grados d
##
## $contrastes.invididuales.entre.ediciones
##   x Estadistica_F ValorP signif
## 1 M          0,93   0,34
## 2 F          7,07   0,01     **
##
## Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘+’ 0.1 ‘ ’ 1
```

Al ingresar la variable de corte `AlumnoGenero` la función en primer lugar calcula la estimación e intervalo de confianza para la media de la variable de interés `y` para cada edición y para cada valor de `AlumnoGenero` (resultado en `$estimaciones`). Nótese que esta variable tiene el valor `Otro` en el edición 2022, pero no en edición 2018, y se muestran valores de estimaciones para esta categoría. Posteriormente en el resultado `$estimaciones.globales`, tenemos los resultados de estimaciones para las dos ediciones considerando todos los valores de `AlumnoGenero`.

En cuanto a las pruebas de hipótesis, la función en su resultado `contraste.global` lleva a acabo la comparación general de igualdad de medias para diferentes ediciones. Posteriormente en el resultado `$contraste.individual.entre.ediciones` lleva a cabo la prueba de hipótesis de igualdad de medias para las diferentes ediciones para cada valor de la variable de corte. Nótese que no existe resultado para el valor `Otro` de la variable `AlumnoGenero`, ya que este valor no está presente en el 2018. De los resultados vemos que no hay diferencia significativa en el desempeño de los estudiantes de género masculino entre 2018 y 2022, mientras que sí existe tal diferencia para los alumnos de género femenino.

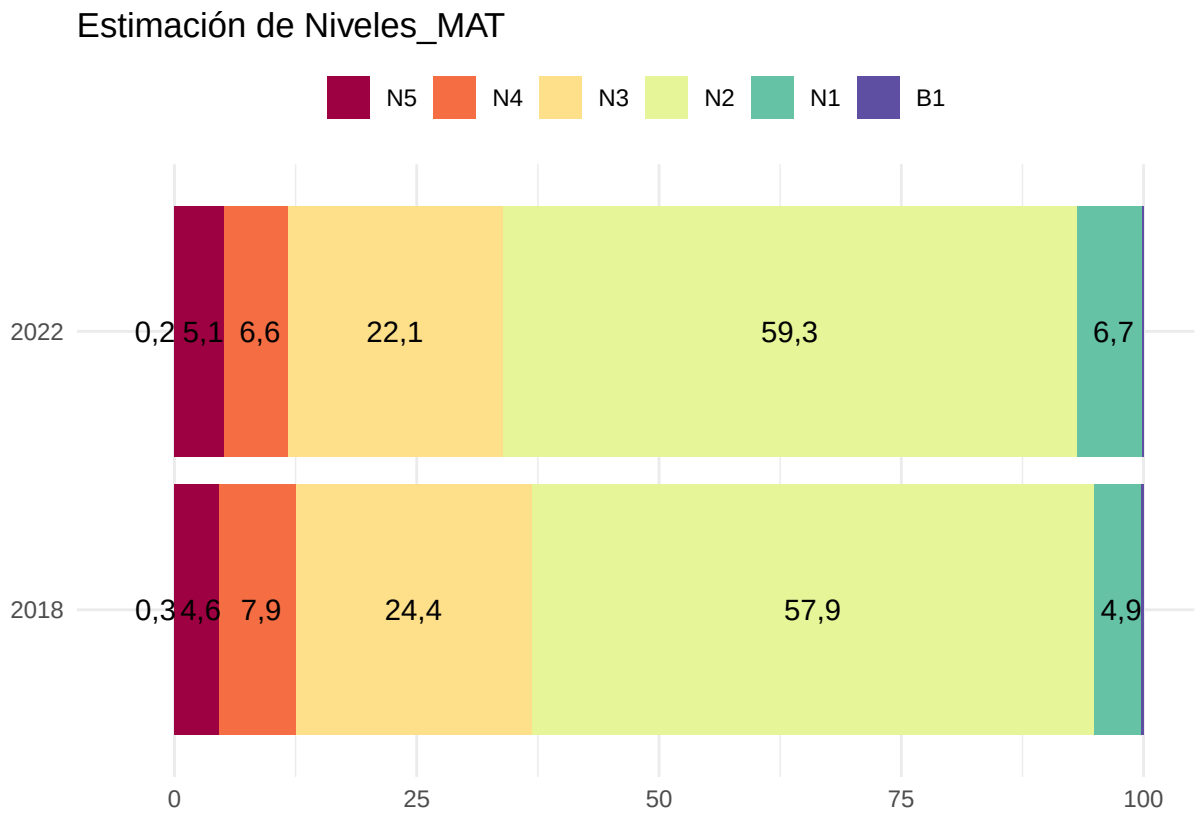
La gráfica solicitada en el comando corresponde a una gráfica de densidad, las cuales se muestran para cada valor de `AlumnoGenero` y también para el nivel global.

**Ejemplo 7**

A continuación, se presenta el análisis de una variable de interés categórica, `Niveles_MAT`, para las ediciones de 2018 y 2022, sin variable de corte `AlumnoGenero`. Para eso primero se debe crear la base de datos combinada usando la función `unionbases.longitudinal` especificando la variable de interés. Por facilidad, incluiremos también las variables de corte `regiones`, `AlumnoGenero`.

```
ARISTAS18.22.matematicas <- unionbases.longitudinal(ediciones = c(2018, 2022),
                     bases = c("ARISTAS18.matematicas2", "ARISTAS22.matematicas"),
                     y = "Niveles_MAT", x = c("regiones", "AlumnoGenero"))
longitudinal(datos = ARISTAS18.22.matematicas, ediciones = c(2018, 2022),
             y = "Niveles_MAT", digit = 2, plot = "bar")
```

```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
```




```
## $estimaciones
##    edicion Niveles_MAT porcentaje   se 2.5 % 97.5 %     n n.NA  %.NA
## 1     2018          B1       0,26 0,07  0,13   0,39  8845  956 10,81
## 2     2018          N1       4,92 0,31  4,30   5,53  8845  956 10,81
## 3     2018          N2      57,95 0,85 56,28  59,62  8845  956 10,81
## 4     2018          N3      24,42 0,56 23,33  25,51  8845  956 10,81
## 5     2018          N4       7,88 0,38  7,14   8,62  8845  956 10,81
## 6     2018          N5       4,58 0,47  3,66   5,49  8845  956 10,81
## 7     2022          B1       0,18 0,05  0,08   0,27 11546 2007 17,38
## 8     2022          N1       6,68 0,34  6,01   7,34 11546 2007 17,38
## 9     2022          N2      59,32 0,98 57,41  61,24 11546 2007 17,38
## 10    2022          N3      22,13 0,60 20,95  23,31 11546 2007 17,38
## 11    2022          N4       6,59 0,43  5,74   7,44 11546 2007 17,38
## 12    2022          N5       5,10 0,37  4,39   5,82 11546 2007 17,38
##
## $contraste.global
## [1] "Estadística Chi-cuadrado de Pearson: 245,34 con 6 grados de libertad, valor p: 4,8e-23"
```

De los resultados arrojados, vemos que en primer lugar en el resultado `$esetimaciones` tenemos las estimaciones y resumen de cantidad de datos y de datos faltantes para cada edición para cada nivel de desempeño, y posteriormente en el resultado `$contraste.global`, tenemos los resultados de prueba de hipótesis Chi-cuadrado de igualdad del vector de porcentajes entre las dos ediciones, los resultados indican que sí hay diferencia de nivel de desempeño (al menos en algún nivel) entre las dos ediciones.

**Ejemplo 8**

A continuación, se presenta el análisis de una variable de interés categórica, `Niveles_MAT`, para las ediciones de 2018 y 2022, con variable de corte **regiones** El siguiente comando se aplica para la base de datos `ARISTAS18.22.matematicas` que fue creado anteriormente por medio de la función `unionbases.longitudinal`

```
longitudinal(datos = ARISTAS18.22.matematicas, ediciones = c(2018, 2022),
             y = "Niveles_MAT", x = "regiones", digit = 3, plot = "bar")
```

```
## Joining with `by = join_by(edicion, var.x)`
```

```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
```


```
## $estimaciones
##    edicion  var.x Niveles_MAT porcentaje    se  2.5 % 97.5 %    n n.NA   %.NA
## 1     2018 CENTRO          B1      0,230 0,096  0,041  0,418 3259  295  9,052
## 2     2018 CENTRO          N1      3,605 0,395  2,832  4,379 3259  295  9,052
## 3     2018 CENTRO          N2     54,190 1,471 51,306 57,074 3259  295  9,052
## 4     2018 CENTRO          N3     26,099 0,894 24,346 27,852 3259  295  9,052
## 5     2018 CENTRO          N4      9,520 0,626  8,293 10,748 3259  295  9,052
## 6     2018 CENTRO          N5      6,356 0,832  4,726  7,986 3259  295  9,052
## 7     2022 CENTRO          B1      0,125 0,091 -0,054  0,304  915  138 15,082
## 8     2022 CENTRO          N1      7,383 1,303  4,830  9,937  915  138 15,082
## 9     2022 CENTRO          N2     59,360 2,732 54,005 64,715  915  138 15,082
```

```
## 10    2022 CENTRO          N3     21,529 2,142 17,331 25,728  915  138 15,082
## 11    2022 CENTRO          N4      7,084 1,057  5,012  9,155  915  138 15,082
## 12    2022 CENTRO          N5      4,519 1,183  2,201  6,837  915  138 15,082
## 13    2018   ESTE          B1      0,393 0,097  0,204  0,583 1407  182 12,935
## 14    2018   ESTE          N1      7,385 1,143  5,145  9,625 1407  182 12,935
## 15    2018   ESTE          N2     60,856 1,494 57,927 63,785 1407  182 12,935
## 16    2018   ESTE          N3     22,104 1,271 19,614 24,594 1407  182 12,935
## 17    2018   ESTE          N4      6,131 0,800  4,563  7,698 1407  182 12,935
## 18    2018   ESTE          N5      3,131 0,720  1,720  4,542 1407  182 12,935
## 19    2022   ESTE          B1      0,225 0,127 -0,024  0,473 1840  325 17,663
## 20    2022   ESTE          N1      6,589 0,714  5,190  7,988 1840  325 17,663
## 21    2022   ESTE          N2     64,362 1,296 61,823 66,902 1840  325 17,663
## 22    2022   ESTE          N3     20,898 1,142 18,660 23,135 1840  325 17,663
## 23    2022   ESTE          N4      5,256 0,647  3,989  6,523 1840  325 17,663
## 24    2022   ESTE          N5      2,670 0,609  1,476  3,864 1840  325 17,663
## 25    2018  NORTE          B1      0,147 0,108 -0,065  0,359 1573  174 11,062
## 26    2018  NORTE          N1      5,146 0,774  3,629  6,663 1573  174 11,062
## 27    2018  NORTE          N2     61,315 1,912 57,568 65,062 1573  174 11,062
## 28    2018  NORTE          N3     23,928 1,356 21,270 26,587 1573  174 11,062
## 29    2018  NORTE          N4      6,952 0,605  5,766  8,139 1573  174 11,062
## 30    2018  NORTE          N5      2,512 0,661  1,217  3,807 1573  174 11,062
## 31    2022  NORTE          B1      0,140 0,084 -0,024  0,305 1919  326 16,988
## 32    2022  NORTE          N1      8,263 0,610  7,068  9,459 1919  326 16,988
## 33    2022  NORTE          N2     67,752 2,005 63,822 71,682 1919  326 16,988
## 34    2022  NORTE          N3     17,832 1,189 15,501 20,162 1919  326 16,988
## 35    2022  NORTE          N4      4,080 0,925  2,267  5,892 1919  326 16,988
## 36    2022  NORTE          N5      1,932 0,430  1,089  2,776 1919  326 16,988
## 37    2018  OESTE          B1      0,403 0,138  0,132  0,674 1537  173 11,256
## 38    2018  OESTE          N1      8,108 0,980  6,188 10,028 1537  173 11,256
## 39    2018  OESTE          N2     66,453 1,529 63,457 69,449 1537  173 11,256
## 40    2018  OESTE          N3     19,819 1,532 16,815 22,822 1537  173 11,256
## 41    2018  OESTE          N4      3,980 0,855  2,304  5,656 1537  173 11,256
## 42    2018  OESTE          N5      1,237 0,299  0,651  1,823 1537  173 11,256
## 43    2022  OESTE          B1      0,000 0,000  0,000  0,000 2073  305 14,713
## 44    2022  OESTE          N1      7,987 0,647  6,718  9,255 2073  305 14,713
## 45    2022  OESTE          N2     63,019 1,673 59,739 66,298 2073  305 14,713
## 46    2022  OESTE          N3     21,082 0,961 19,198 22,966 2073  305 14,713
## 47    2022  OESTE          N4      5,420 0,838  3,777  7,063 2073  305 14,713
## 48    2022  OESTE          N5      2,493 0,472  1,568  3,419 2073  305 14,713
## 49    2018    SUR          B1      0,271 0,185 -0,092  0,634 1069  132 12,348
## 50    2018    SUR          N1      5,190 0,512  4,186  6,194 1069  132 12,348
## 51    2018    SUR          N2     61,724 2,283 57,248 66,199 1069  132 12,348
## 52    2018    SUR          N3     23,968 2,021 20,007 27,929 1069  132 12,348
## 53    2018    SUR          N4      6,144 0,700  4,772  7,516 1069  132 12,348
## 54    2018    SUR          N5      2,703 0,457  1,808  3,598 1069  132 12,348
## 55    2022    SUR          B1      0,236 0,084  0,071  0,401 4799  913 19,025
## 56    2022    SUR          N1      5,893 0,565  4,786  7,000 4799  913 19,025
## 57    2022    SUR          N2     55,348 1,458 52,491 58,206 4799  913 19,025
## 58    2022    SUR          N3     23,682 0,967 21,788 25,577 4799  913 19,025
## 59    2022    SUR          N4      7,717 0,637  6,470  8,965 4799  913 19,025
## 60    2022    SUR          N5      7,123 0,648  5,852  8,394 4799  913 19,025
##
## $estimaciones.globales
##    edicion Niveles_MAT porcentaje    se  2.5 % 97.5 %     n n.NA   %.NA
```

```
## 1     2018          B1      0,259 0,065  0,131  0,387  8845  956 10,808
## 2     2018          N1      4,918 0,315  4,302  5,534  8845  956 10,808
## 3     2018          N2     57,949 0,850 56,282 59,616  8845  956 10,808
## 4     2018          N3     24,417 0,555 23,329 25,505  8845  956 10,808
## 5     2018          N4      7,880 0,378  7,140  8,621  8845  956 10,808
## 6     2018          N5      4,576 0,468  3,658  5,495  8845  956 10,808
## 7     2022          B1      0,179 0,049  0,082  0,275 11546 2007 17,383
## 8     2022          N1      6,676 0,341  6,007  7,344 11546 2007 17,383
## 9     2022          N2     59,322 0,978 57,406 61,238 11546 2007 17,383
## 10    2022          N3     22,131 0,604 20,946 23,315 11546 2007 17,383
## 11    2022          N4      6,589 0,432  5,743  7,436 11546 2007 17,383
## 12    2022          N5      5,104 0,365  4,388  5,820 11546 2007 17,383
##
## $contraste.global
## [1] "Estadística Chi-cuadrado de Pearson: 245,343 con 6 grados de libertad, valor p: 4,78e-23"
##
## $contrastes.invididuales.entre.ediciones
##        x Estadistica_Chi2 ValorP signif
## 1  OESTE           29,077  0,003     **
## 2 CENTRO           30,354      0    ***
## 3  NORTE           64,739      0    ***
## 4    SUR           21,168      0    ***
## 5   ESTE           19,758  0,079      +
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

Al ingresar la variable de corte `regiones` la función en primer lugar calcula la estimación e intervalo de confianza para la media de la variable de interés `y` para cada edición y para cada valor de `regiones` (resultado en `$estimaciones`). Posteriormente en el resultado `$estimaciones.globales`, tenemos los resultados de estimaciones para las dos ediciones considerando todos los valores de `regiones`.

En cuanto a las pruebas de hipótesis, la función en su resultado `contraste.global` lleva a acabo la comparación general de igualdad de porcentajes para diferentes ediciones donde vemos que el nivel de desempeño sí es diferente entre las dos ediciones. Posteriormente en el resultado `$contraste.individual.entre.ediciones` lleva a cabo la prueba de hipótesis de igualdad de porcentajes para las diferentes ediciones para cada región. Vemos que aunque se detectan diferencias significativas en todas las regiones, la que menos diferencia presenta es la región ESTE.

La gráfica solicitada en el comando corresponde a una gráfica de barras, las cuales se muestran para cada valor de 'regiones" y también para el nivel global, el usuario puede elegir descargar los resultados de estimación a un archivo excel y la gráfica en un pdf.

### Análisis longitudinal para más de una variable de corte

Para llevar a cabo comparaciones entre diferentes ediciones con más de una variable de corte, el paquete cuenta con la siguiente función:

- `longitudinal.multiple`: Esta función compara la media de una variable de interés numérica o categórica entre las diferentes ediciones de ARISTAS para más de una variable de corte.

El uso de la función es:

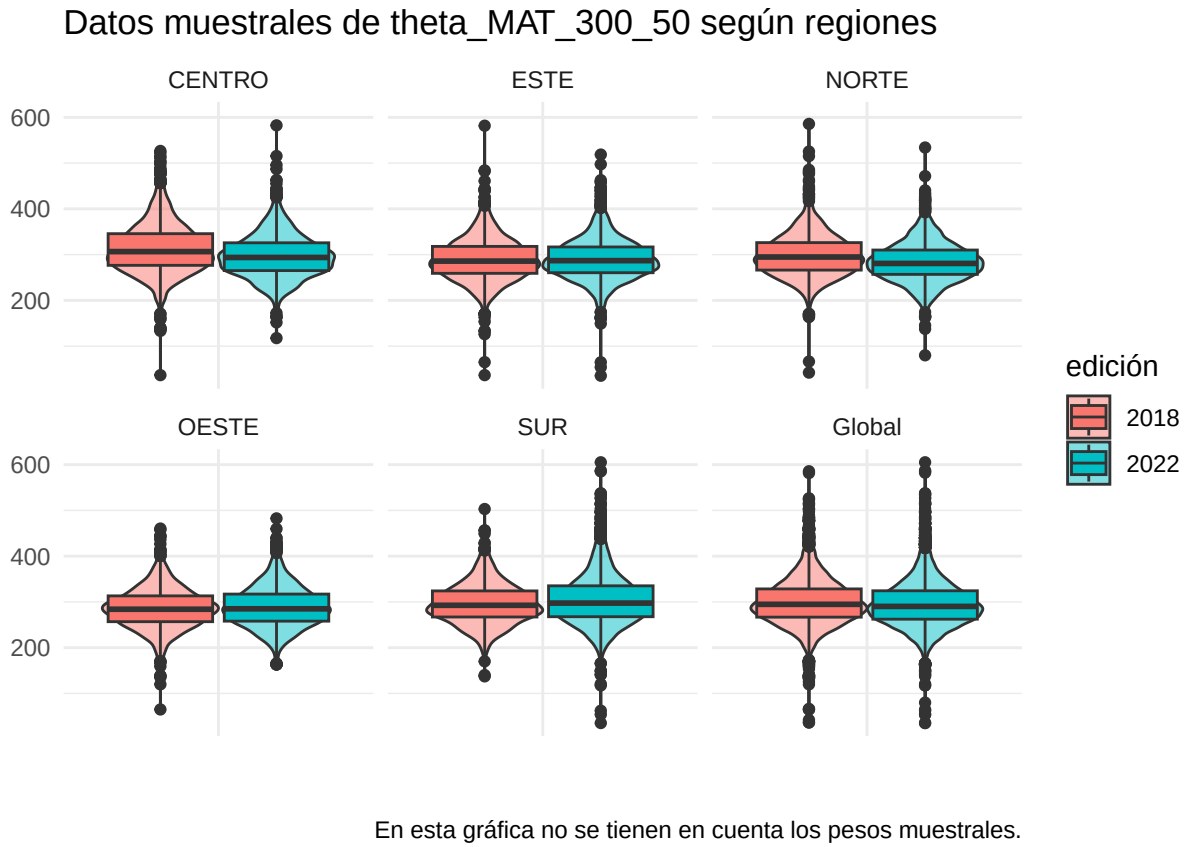
```
longitudinal.multiple(datos, ediciones, y, x, digit = 3, plot = "NULL",
                      descarga = FALSE, archivo = NULL, ...)
```

Los argumentos de `datos`, `ediciones`, `y`, `x`, `digit`, `plot`, `descarga`, `archivo`, son idénticos a los de la función `transversal`. El argumento `x` debe ser un vector de caracteres indicando el nombre de las variables de corte, estas variable deben ser de tipo categórico. En caso de que necesite comparar la media de `y` para diferentes subgrupos definidos por una variable numérica `x`, por ejemplo, comparar el desempeño para estudiantes de diferentes grupos de edad, se debe crear la variable categórica definiendo los rangos de edad, y posteriormente utilizar esta función. Los argumentos `ancho` y `alto` deben ser vector de valores numéricos del mismo tamaño que el número de variables de corte, y corresponden al ancho y alto en pulgadas de las gráficas producidas.

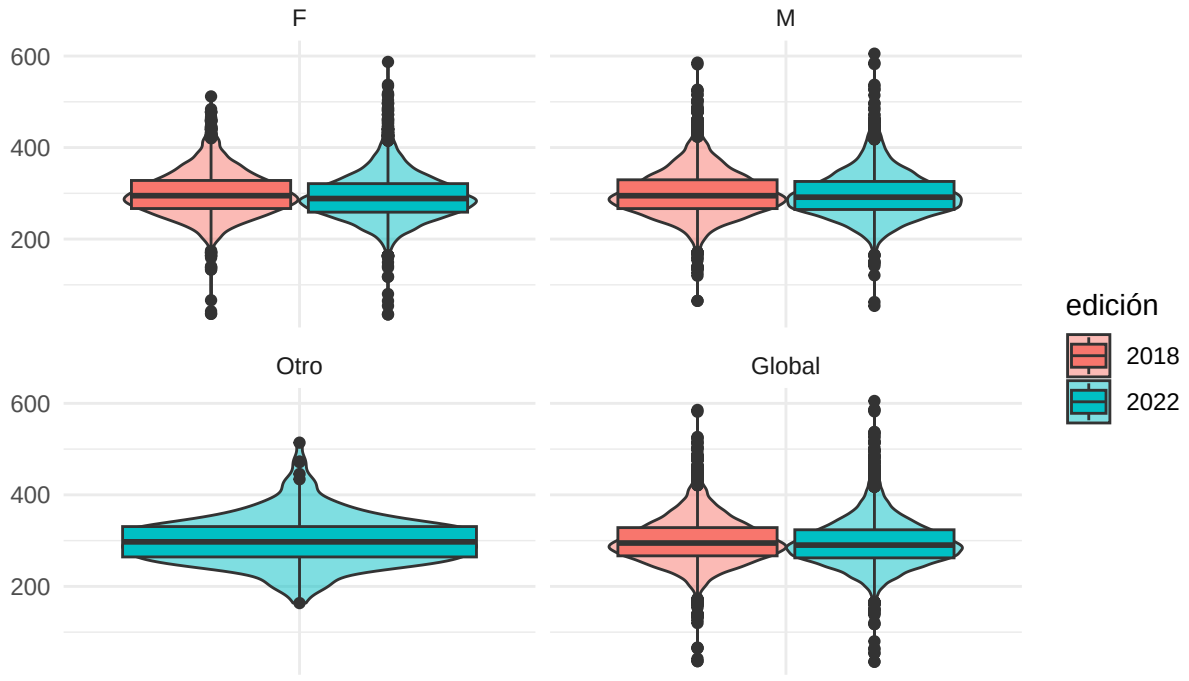
Esta función hace uso de la función `longitudinal`, por lo que los detalles teóricos de cómo se realizan cálculos son idénticos.

A continuación se muestran algunos ejemplos sobre el uso de esta función:

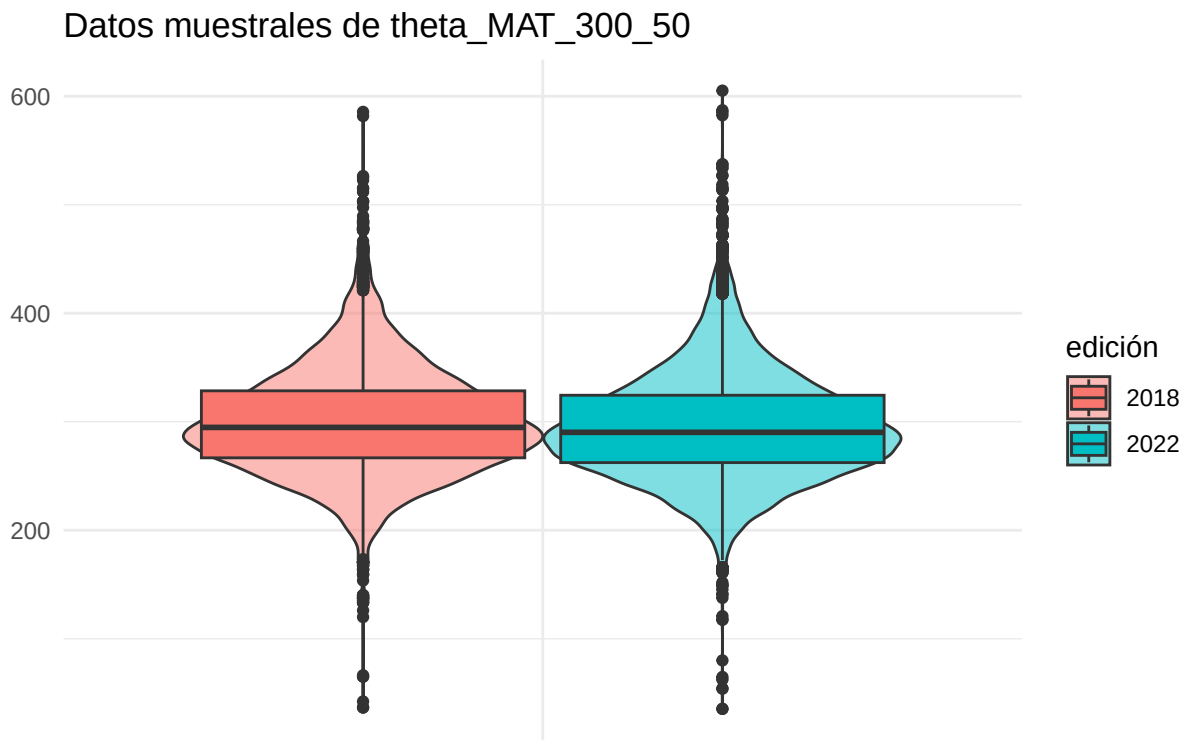
```
ARISTAS18.22.matematicas <- unionbases.longitudinal(ediciones = c(2018, 2022),
                      bases = c("ARISTAS18.matematicas2", "ARISTAS22.matematicas"),
                      y = "theta_MAT_300_50", x = c("regiones", "AlumnoGenero"))
longitudinal.multiple(datos = ARISTAS18.22.matematicas, ediciones = c(2018, 2022),
                      y = "theta_MAT_300_50", x = c("regiones", "AlumnoGenero"),
                      digit = 2, plot = "violin")
```




En esta gráfica no se tienen en cuenta los pesos muestrales.




En esta gráfica no se tienen en cuenta los pesos muestrales.




En esta gráfica no se tienen en cuenta los pesos muestrales.

```
## Los siguientes resultados consideran todos los estudiantes para los que se cuenta con ponderador
##  los resultados de estimación y prueba de hipótesis consideran pesos réplica y pesos de estudiantes
##  los conteos de datos y de NA no son ponderados
## $estimaciones
## $estimaciones$regiones
##    edicion  var.x  var.y   se  2.5 % 97.5 %    n n.NA  %.NA
## 1     2018 CENTRO 306,96 1,99 303,05 310,87 3259  295  9,05
## 2     2022 CENTRO 297,03 4,06 289,08 304,98  915  138 15,08
## 3     2018   ESTE 291,73 2,44 286,94 296,51 1407  182 12,94
## 4     2022   ESTE 290,23 1,83 286,64 293,82 1840  325 17,66
## 5     2018  NORTE 295,24 2,14 291,05 299,43 1573  174 11,06
## 6     2022  NORTE 284,98 1,93 281,20 288,76 1919  326 16,99
## 7     2018  OESTE 284,93 2,25 280,53 289,33 1537  173 11,26
## 8     2022  OESTE 290,16 1,95 286,33 293,99 2073  305 14,71
## 9     2018    SUR 294,85 2,09 290,76 298,93 1069  132 12,35
## 10    2022    SUR 303,33 2,04 299,33 307,33 4799  913 19,02
##
## $estimaciones$AlumnoGenero
##   edicion var.x  var.y   se  2.5 % 97.5 %    n n.NA  %.NA
## 1    2018     F 299,23 1,31 296,67 301,79 4317  456 10,56
## 2    2022     F 294,40 1,44 291,57 297,23 5482  892 16,27
## 3    2018     M 300,86 1,26 298,38 303,34 4528  500 11,04
## 4    2022     M 299,38 1,13 297,17 301,58 5603  978 17,45
## 5    2022  Otro 304,51 4,79 295,11 313,90  263   32 12,17
##
##
```

```
## $contrastes_individuales_entre_ediciones
## $contrastes_individuales_entre_ediciones$regiones
##        x Estadistica_F ValorP signif
## 1  OESTE          3,22   0,08      +
## 2 CENTRO          4,53   0,04      *
## 3  NORTE         12,01      0     **
## 4    SUR          9,68      0     **
## 5   ESTE          0,25   0,62
##
## $contrastes_individuales_entre_ediciones$AlumnoGenero
##   x Estadistica_F ValorP signif
## 1 M          0,93   0,34
## 2 F          7,07   0,01     **
##
##
## $estimaciones_globales
##   edicion theta_MAT_300_50   se  2.5 % 97.5 %     n n.NA  %.NA
## 1    2018           300,06 1,13 297,85 302,28  8845  956 10,81
## 2    2022           297,16 1,17 294,86 299,46 11546 2007 17,38
##
## $contraste_global
## [1] "Estadística F para comparación de theta_MAT_300_50 entre las 2 ediciones: 3,76 con 1 y 81 grados
```

Esta función muestra en su primer resultado, `$estimaciones`, todos los resultados de `$estimaciones` de la función `longitudinal` para cada una de las dos variables de corte: `regiones` y `AlumnoGenero`, estos resultados se identifican con los nombres de `$estimaciones$regiones` y `$estimaciones$AlumnoGenero`.

Posteriormente, para cada valor de cada variable de corte, realiza la prueba de hipótesis de igualdad de medias entre las ediciones. Estos resultados se identifican con el nombre `$contrastes_individuales_entre_ediciones`.

Finalmente, la función arroja el resultado `estimaciones_globales` y `$contraste_global` con los resultados de estimación y de prueba de hipótesis para la totadlidad de los datos ingresados.

## Modelamiento estadístico

Para ajustar modelos estadísticos para explicar el comportamiento de una variable de interés, el paquete cuenta con una única función `modelo`, esta función permite ajustar cinco tipos de modelos. El uso general de la función es:

```
modelo(base, formula, tipo, pesos = NULL, pesos.rep = NULL, pesos.centro = NULL, cluster_var = NULL)
```

A continuación se explican los diferentes argumentos de la función:

- `base` La base de datos que se quiere analizar, puede ser cualquiera de las bases de ARISTAS incluidas en el paquete o cualquiera otra base ARISTAS que no sea de uso público. Tenga en cuenta que las columnas que contengan los pesos replicados deben tener nombres: `EST_W_REP_1`, `EST_W_REP_2`, etc.
- `formula` La fórmula para especificar (i) el modelo para un modelo de regresión o un modelo con errores clusterizados, (ii) el modelo de los efectos fijos en un modelo mixto
- `tipo` Especificar el tipo de modelo que se estimará, `lm` para un modelo de regresión, `lm_error_cluster` para un modelo de regresión con errores clusterizados, `linear_mixed` parea un modelo multinivel con pesos en el nivel de estudiantes, `linear_mixed_weight_2` para un modelo multinivel con pesos en el nivel de estudiantes y a nivel de centros, y `logit` para un modelo de regresión multinivel logística con pesos en el nivel de estudiantes.

- `pesos` Un vector de pesos o ponderadores para el nivel de estudiantes.
- `pesos.rep` Pesos réplica para el nivel de estudiantes.
- `pesos.centro` Un vector de pesos o ponderadores para el nivel de centros.
- `cluster_var` La variable que indica los clusters para todos los modelos que consideren dos niveles

Antes de entrar a explicar cada modelo, crearemos una base de datos combinando variables de desempeño de matemáticas y variables de contexto para la edición de 2022. Las variables de interés que consideraremos son `theta_MAT_300_50` y `Niveles_MAT`, y las covariables serán `EF1m`: género y `EF2d`: ascendencia.

```
# Seleccionar variables relevantes de la base de ARISTAS22.matematicas
 ARISTAS22.matematicas.modelo <- ARISTAS22.matematicas |>
          dplyr::select(AlumnoCodigoDes, CentroCodigoDes, #Seleccionar código de estudiantes y centros,
          peso_CENTRO, peso_MEst, #Seleccionar los ponderadores de estudiantes, de centro,
          theta_MAT_300_50, # Seleccionar la variabla dependiente numérica
          Niveles_MAT, # Seleccionar la variabla dependiente categórica
          starts_with("EST_W_REP_")) # Pesos réplica

 # Procesamiento previo de información de contexto
 # Variable EF1m: género. Convertir el 99 a NA, y convertir la variable a factor
 ARISTAS22.contexto$EF1m[ARISTAS22.contexto$EF1m == 99] <- NA_integer_
 ARISTAS22.contexto$EF1m <- factor(ARISTAS22.contexto$EF1m)
 # Variable EF2d: ascendencia. Convertir el 99 a NA, y convertir la variable a factor
 ARISTAS22.contexto$EF2d[ARISTAS22.contexto$EF2d == 99] <- NA_integer_
 ARISTAS22.contexto$EF2d <- factor(ARISTAS22.contexto$EF2d)
 # Variable EF2d: ascendencia. Unir las categorías de ascendencia
 ARISTAS22.contexto$EF2d <- as.double(ARISTAS22.contexto$EF2d == "1") # 1 es ascendencia blanca

 # Seleccionar las covariables de la base de ARISTAS22.contexto
 ARISTAS22.contexto.modelo <- ARISTAS22.contexto |>
   dplyr::select(AlumnoCodigoDes, EDAD,
          EF1m, EF2d)

 # Unir las informaciones de desempeño de matemáticas y las dos covariables
 datos1 <- ARISTAS22.matematicas.modelo |>
   dplyr::left_join(ARISTAS22.contexto.modelo,
             by = "AlumnoCodigoDes")
```

Los ejemplos que se mostrarán a continuación utilizarán la anterior base de datos `datos1`.

### Modelo de regresión

Cuando el argumento ingresado es `tipo == "lm"`, esta función ajusta tres posibles modelos de regresión según los pesos que el usuario ingrese, no se admiten pesos para el segundo nivel. Estos modelos son:

- Si el usuario no ingresa ni pesos, ni pesos réplica, la función ajustará un modelo de regresión con el método de mínimos cuadrados ordinarios por medio de la función `lm`.
- Si el usuario ingresa los pesos pero no ingresa los pesos réplica, la función ajustará un modelo de regresión con el método de mínimos cuadrados ponderados por medio de la función `lm`, y los pesos ingresados por el usuario ingresarán al argumento `weights` de la función `lm`.

- Si el usuario ingresa ambos pesos y pesos réplica, la función ajustará un modelo de regresión (con la función `svyglm` del paquete `survey`) con base en un diseño muestral con pésos réplica especificado con la función `svrepdesign`.

**Ejemplo 9**

A continuación ajustaremos un modelo lineal para explicar el desempeño `theta_MAT_300_50` en función de `edad`, `género` y `ascendencia`:

```
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "lm")
```

```
##
## Call:
## lm(formula = formula, data = base)
##
## Coefficients:
## (Intercept)         EDAD        EF1m2        EF1m3         EF2d
##     410.896       -8.185       -5.962        2.844       13.837
```

```
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "lm",
                    pesos = "peso_MEst")
```

```
##
## Call:
## lm(formula = formula, data = base, weights = pesos.lm)
##
## Coefficients:
## (Intercept)         EDAD        EF1m2        EF1m3         EF2d
##     420.622       -8.663       -6.366        3.000       13.031
```

```
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "lm",
                   pesos = "peso_MEst", pesos.rep = paste0("EST_W_REP_", 1:160))
```

```
## Call: svrepdesign.default(data = base, type = "Fay", weights = ~pesos.lm,
##     repweights = paste0("EST_W_REP_[", 1, "-", length(pesos.rep),
##         "]"), combined.weights = TRUE, rho = 1.5, mse = F)
## Fay's variance method (rho= 1.5 ) with 83 replicates.
##
## Call:  survey::svyglm(formula = formula, design = dis, family = gaussian())
##
## Coefficients:
## (Intercept)         EDAD        EF1m2        EF1m3         EF2d
##     420.622       -8.663       -6.366        3.000       13.031
##
## Degrees of Freedom: 8779 Total (i.e. Null);  78 Residual
##   (2766 observations deleted due to missingness)
## Null Deviance:       23460000
## Residual Deviance: 22410000  AIC: 94960
```

Podemos que los dos últimos modelos comparten los mismos coeficientes de regresión, ya que ambos modelos utilizan el mismo conjunto de pesos.

**Modelo de regresión con errores clusterizados con o sin pesos a nivel de estudiantes.**

Cuando el argumento ingresado es `tipo == "lm_error_cluster"`, esta función ajusta tres posibles modelos de regresión con errores estándares robustos según los pesos de estudiantes que el usuario ingrese, no se admiten pesos para el segundo nivel. Estos modelos son:

- Si el usuario no ingresa ni pesos, ni pesos réplica, la función ajustará un modelo de regresión con errores clusterizados por medio de la función `lm_robust` del paquete `estimatr` sin ningún peso en su argumento `weights`
- Si el usuario ingresa los pesos pero no ingresa los pesos réplica, la función ajustará un modelo de regresión con errores clusterizados por medio de la función `lm_robust` del paquete `estimatr` y los pesos ingresados por el usuario ingresarán al argumento `weights` de la función `lm_robust`.
- Si el usuario ingresa ambos pesos y pesos réplica, la función ajustará un modelo de regresión con errores clusterizados por medio de la función `lm_robust` del paquete `estimatr` y los pesos ingresados por el usuario ingresarán al argumento `weights` de la función `lm_robust`. Los pesos réplica serán utilizados para calcular el error estándar de las estimaciones así:

  (i) Para el vector de pesos, ajustar el modelo de error clusterizado y se denota el parámetro de interés (coeficiente de regresión) con $\hat{\beta}$,

  (ii) Para cada conjunto de pesos de los pesos réplica, ajustar el modelo de error clusterizado y el parámetro de interés correspondiente se denota por $\hat{\beta}_k$ para $k = 1, \cdots, K$, donde $K$ denota el número de vectores de pesos réplica.

  (iii) El error estándar de la estimación se calcula como $\sqrt{\frac{\sum_{i=1}^{K}(\hat{\beta}_k - \hat{\beta})^2}{K}}$

**Ejemplo 10**

A continuación ajustaremos un modelo de regresión con errores clusterizados para explicar el desempeño `theta_MAT_300_50` en función de `edad`, `género` y `ascendencia`:

```
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "lm_error_cluster",
       cluster_var = "CentroCodigoDes")
```

```
##               Estimate Std. Error    t value     Pr(>|t|)   CI Lower   CI Upper
## (Intercept) 410.896416  16.868961 24.3581337 7.153337e-29 376.989018 444.803814
## EDAD         -8.184919   1.118187 -7.3198107 2.771132e-09 -10.434775  -5.935064
## EF1m2        -5.961725   1.109119 -5.3751888 1.654859e-07  -8.145341  -3.778109
## EF1m3         2.844078   3.618726  0.7859335 4.335756e-01  -4.326417  10.014572
## EF2d         13.836899   1.655720  8.3570303 4.175275e-15  10.576287  17.097511
##                    DF
## (Intercept)  48.54991
## EDAD         46.72240
## EF1m2       270.16056
## EF1m3       111.35501
## EF2d        255.23030
```

```
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "lm_error_cluster",
                   pesos = "peso_MEst", cluster_var = "CentroCodigoDes")
```

```
##                Estimate Std. Error    t value     Pr(>|t|)   CI Lower   CI Upper
## (Intercept) 420.621626  21.862584 19.2393373 2.357067e-16 375.561413 465.681838
## EDAD         -8.662787   1.448806 -5.9792598 3.620176e-06 -11.653349  -5.672224
## EF1m2        -6.365790   1.224240 -5.1997876 9.186666e-07  -8.791698  -3.939881
## EF1m3         3.000273   4.257426  0.7047153 4.841370e-01  -5.543206  11.543752
## EF2d         13.030721   1.837252  7.0925058 1.594645e-10   9.387783  16.673659
##                    DF
## (Intercept)  24.64003
## EDAD         23.94340
## EF1m2       111.02337
## EF1m3        51.91600
## EF2d        104.98303
```

```
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "lm_error_cluster",
                    pesos = "peso_MEst", cluster_var = "CentroCodigoDes",
       pesos.rep = paste0("EST_W_REP_", 1:160))
```

```
##             estimacion       s.e.    valor.t
## (Intercept) 420.621626 10.7287451  39.205110
## EDAD         -8.662787  0.7132790 -12.145019
## EF1m2        -6.365790  0.4970562 -12.806981
## EF1m3         3.000273  1.9888148   1.508573
## EF2d         13.030721  0.8494432  15.340309

## NULL
```

**Modelo lineal mixto con dos niveles.**

Cuando el argumento ingresado es `tipo == "linear_mixed"`, esta función ajusta un modelo multinivel con o sin pesos en el nivel de estudiantes, no se admiten pesos en el segundo nivel. Dependiendo de los pesos que el usuario ingrese, se tienen los siguientes casos:

- Si el usuario no ingresa ni pesos, ni pesos réplica, la función ajustará un modelo de regresión de dos niveles, donde el segundo nivel será indicado por el argumento `cluster_var`. El ajuste del modelo es por medio de la función `lmer` del paquete `lme4` sin ningún peso en su argumento `weights`.
- Si el usuario ingresa los pesos pero no ingresa los pesos réplica, la función ajustará un modelo de regresión de dos niveles, donde el segundo nivel será indicado por el argumento `cluster_var`. El ajuste del modelo es por medio de la función `lmer` del paquete `lme4` y los pesos ingresados por el usuario ingresarán al argumento `weights` de la función `lmer`.
- Si el usuario ingresa ambos pesos y pesos réplica, la función ajustará un modelo de regresión de dos niveles por medio de la función `lmer` del paquete `lme4` y los pesos ingresados por el usuario ingresarán al argumento `weights` de la función `lm_robust`. Los pesos réplica serán utilizados para calcular el error estándar de las estimaciones similar a los procedimientos presentados anteriormente.

**Ejemplo 11**

A continuación ajustaremos un modelo mixto para explicar el desempeño `theta_MAT_300_50` en función de `edad`, `género` y `ascendencia`:

```r
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "linear_mixed",
       cluster_var = "CentroCodigoDes")
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: theta_MAT_300_50 ~ EDAD + EF1m + EF2d + (1 | CentroCodigoDes)
##    Data: base
## REML criterion at convergence: 92056.96
## Random effects:
##  Groups          Name        Std.Dev.
##  CentroCodigoDes (Intercept) 25.94
##  Residual                    43.93
## Number of obs: 8780, groups:  CentroCodigoDes, 339
## Fixed Effects:
## (Intercept)         EDAD        EF1m2        EF1m3         EF2d
##     359.261       -4.469       -6.491        0.462        4.349
```

```r
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "linear_mixed",
                   pesos = "peso_MEst", cluster_var = "CentroCodigoDes")
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: theta_MAT_300_50 ~ EDAD + EF1m + EF2d + (1 | CentroCodigoDes)
##    Data: base
## Weights: pesos.lm
## REML criterion at convergence: 93405.11
## Random effects:
##  Groups          Name        Std.Dev.
##  CentroCodigoDes (Intercept) 25.41
##  Residual                    44.56
## Number of obs: 8780, groups:  CentroCodigoDes, 339
## Fixed Effects:
## (Intercept)         EDAD        EF1m2        EF1m3         EF2d
##     371.095       -5.202       -6.862        1.707        4.219
```

```r
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "linear_mixed",
                   pesos = "peso_MEst", cluster_var = "CentroCodigoDes",
       pesos.rep = paste0("EST_W_REP_", 1:160))
```

```
##             estimacion      s.e.     valor.t
## (Intercept) 371.095308 7.5824517  48.9413350
## EDAD         -5.201745 0.5015031 -10.3723094
## EF1m2        -6.861790 0.4718721 -14.5416314
## EF1m3         1.706750 1.8035443   0.9463309
## EF2d          4.218835 0.6508799   6.4817406
## El R2 condicional para el modelo mixto es  0.2522427
```

```
## NULL
```

### Modelo lineal mixto con dos niveles con pesos en nivel de centros

Cuando el argumento ingresado es `tipo == "linear_mixed_weight_2"`, esta función ajusta un modelo multinivel con pesos en ambos niveles (estudiantes y centros). Para usar esta opción es indispensable que el

usuario ingreso estos dos conjuntos de pesos, no se utilizarán los pesos réplicas. El ajuste del modelo es por medio de la función `mix` del paquete `WeMix`, los pesos de ambos niveles ingresados por el usuario ingresarán al argumento `weights` de la función `mix`.

**Ejemplo 12**

A continuación ajustaremos un modelo mixto con pesos en ambos niveles para explicar el desempeño `theta_MAT_300_50` en función de `edad`, `género` y `ascendencia`:

```
modelo(base = datos1, theta_MAT_300_50 ~ EDAD + EF1m + EF2d, tipo = "linear_mixed_weight_2",
                    pesos = "peso_MEst", pesos.centro = "peso_CENTRO", cluster_var = "CentroCodigoDes")
```

```
## (Intercept)        EDAD       EF1m2       EF1m3        EF2d
##  420.621626   -8.662787   -6.365790    3.000273   13.030721
```

**Modelo mixto logística con dos niveles**

Cuando el argumento ingresado es `tipo == "logit"`, esta función ajusta un modelo de regresión logística multinivel que permite pesos y también permite pesos réplica en el primer nivel (estudiantes). El ajuste del modelo es por medio de la función `glmer` del paquete `lme4`, los pesos de ambos niveles ingresados por el usuario ingresarán al argumento `weights` de la función `mix`. Dependiendo si el usuario ingresa o no los pesos de estudiantes en el argumento `pesos`, la función `glmer` se ejecutará con o sin `weights`. Cuando el usuario ingresan los pesos réplica, el error estándar de las estimaciones se calcularán según la metodología explicada anteriormente.

**Ejemplo 13**

A continuación ajustaremos un modelo de regresión multinivel logístico para la variable dependiente nivel de desempeño igual o superior a dos, vs. nivel de desempeño menor a dos con pesos en ambos niveles. En primer lugar se crea la variable de interés:

```
datos1 <- datos1 |> dplyr::mutate(nivel_dos= Niveles_MAT == "B1" | Niveles_MAT == "N1")
```

```
formula.fija.logit  <- nivel_dos ~ EDAD + EF1m + EF2d
modelo(base = datos1, formula = formula.fija.logit, tipo = "logit", cluster_var = "CentroCodigoDes")
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: nivel_dos ~ EDAD + EF1m + EF2d + (1 | CentroCodigoDes)
##    Data: base
##       AIC       BIC    logLik  deviance  df.resid
##  4564.133  4607.431 -2276.066  4552.133     10055
## Random effects:
##  Groups          Name        Std.Dev.
##  CentroCodigoDes (Intercept) 0.6278
## Number of obs: 10061, groups:  CentroCodigoDes, 340
## Fixed Effects:
## (Intercept)         EDAD        EF1m2        EF1m3         EF2d
##    -5.36242      0.16636      0.28283      0.06186     -0.19398
```

```
modelo(base = datos1, formula = formula.fija.logit, tipo = "logit", pesos = "peso_MEst",
       cluster_var = "CentroCodigoDes")
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: nivel_dos ~ EDAD + EF1m + EF2d + (1 | CentroCodigoDes)
##    Data: base
## Weights: base[["pesos.estudiantes"]]
##       AIC       BIC    logLik  deviance  df.resid
##  3896.051  3939.350 -1942.025  3884.051     10055
## Random effects:
##  Groups          Name        Std.Dev.
##  CentroCodigoDes (Intercept) 0.431
## Number of obs: 10061, groups:  CentroCodigoDes, 340
## Fixed Effects:
## (Intercept)         EDAD        EF1m2        EF1m3         EF2d
##    -5.97549      0.20407      0.27499      0.06097     -0.22357
## optimizer (Nelder_Mead) convergence code: 0 (OK) ; 0 optimizer warnings; 1 lme4 warnings
```

```
modelo(base = datos1, formula = formula.fija.logit, tipo = "logit", pesos = "peso_MEst",
       cluster_var = "CentroCodigoDes", pesos.rep = paste0("EST_W_REP_", 1:160))
```

```
##              estimacion       s.e.    valor.t
## (Intercept) -5.97549208 0.59900574 -9.9756842
## EDAD         0.20407484 0.04000332  5.1014478
## EF1m2        0.27499216 0.06441630  4.2689843
## EF1m3        0.06097038 0.22466594  0.2713824
## EF2d        -0.22357140 0.07543973 -2.9635765

## NULL
```

## Funciones auxiliares

Aparte de las anteriores funciones descritas, el paquete ARISTAS también contiene otras funciones auxiliares que son útiles para reducir la extensión de las funciones esenciales, sin embargo, estas funciones no son exportadas, es decir, los usuarios no tendrán acceso a ellas. Estas funciones son las siguientes:

- `ajustar.decimales`: esta función toma una base de datos, y aproxima el número de decimales de todas las variables numéricas al número de decimales que el usuario indique.
- `ajustarDecimalesFormatoLista`: esta función toma una lista de bases de datos, y aproxima el número de decimales de todas las variables numéricas de todas las bases de datos al número de decimales que el usuario indique.
- `reorder_factor_last`: esta función toma un vector de factores, convierte a los `NA` a un nivel y además asigna al nivel indicado por el usuario el último orden.
- `significancia`: esta función toma un vector de valores $p$, y devuelven las marcas de siginifcancia estadística, siguiendo esta notación: *** para valor $p$ entre 0 y 0.001; ** para valor $p$ entre 0.001 y 0.01; * para valor $p$ entre 0.01 y 0.05; + para valor $p$ entre 0.05 y 0.1, y sin marca para valor $p$ mayor al 0.1.

## Bibliografía


- R Core Team (2023). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.