

# Package: electorAr (via r-universe)

November 4, 2024

**Title** Toolbox for Argentina's Electoral Data

**Version** 0.0.1.3

**Description** The package provides tools for accessing and working with electoral data from Argentina // El paquete facilita herramientas para el acceso a datos y el trabajo con datos electorales de Argentina.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**Imports** assertthat, attempt, curl, dplyr, forcats, glue, httr, purrr, stringr, tibble, tidyr, magrittr, vroom, jsonlite

**Depends** R (>= 2.10)

**Suggests** DT, knitr

**URL** <https://github.com/PoliticaArgentina/electorAr>

**BugReports** <https://github.com/PoliticaArgentina/electorAr/issues>

**Repository** <https://politicaargentina.r-universe.dev>

**RemoteUrl** <https://github.com/PoliticaArgentina/electorAr>

**RemoteRef** HEAD

**RemoteSha** 515a3ac32bc6d2186144145e6338b4bf878bb1cf

## Contents

|                                   |   |
|-----------------------------------|---|
| compute_competitiveness . . . . . | 2 |
| compute_concentration . . . . .   | 3 |
| compute_disproportion . . . . .   | 4 |
| compute_nep . . . . .             | 6 |
| compute_seats . . . . .           | 7 |
| electorAr . . . . .               | 8 |

|                                    |    |
|------------------------------------|----|
| get_election_data . . . . .        | 9  |
| get_election_results . . . . .     | 10 |
| get_multiple_elections . . . . .   | 11 |
| get_names . . . . .                | 12 |
| make_long . . . . .                | 13 |
| show_available_elections . . . . . | 14 |
| tucuman_dip_gral_2017 . . . . .    | 15 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>16</b> |
|--------------|-----------|

---

compute\_competitiveness

*Calcula Competitividad (Computes Competitiveness)*

---

## Description

Funcion que describe el grado de competencia de una eleccion

*(Function that describes the degree of competition of an election)*

## Usage

```
compute_competitiveness(data, level = "provincia")
```

## Arguments

|       |   |
|-------|---|
| data  | un tibble guardado como objeto en el Enviroment luego de consultar <a href="#">get_election_data</a> o <a href="#">get_election_results</a><br><i>(A tibble saved as an object in the Enviroment after querying <a href="#">get_election_data</a> or <a href="#">get_election_results</a>).</i>   |
| level | un character que establece el nivel de agregacion sobre el que se quiere calcular la competitividad: por default es provincia y para datos obtenidos con <a href="#">get_election_data</a> se desagregan las observaciones asignando los valores departamento o circuito al parametro<br><i>(Establishes the level of aggregation on which you want to compute competitiveness: by definition it is provincia and for data obtained with <a href="#">get_election_data</a> the observations are disaggregated by assigning the values departamento or circuito to the parameter).</i> |

## Details

*Competitividad* mide la diferencia porcentual de votos válidos obtenidos por los dos partidos más votados  $a$  y  $b$ . Puede tomar valores entre  $[0, 1]$  donde 1 es lo más comeptitivo ( $a = 50\%$  y  $b = 50\%$  de los votos). La fórmula utilizada es:

$$\text{Competitividad} = 1 - (a - b)$$

## REQUISITO:

Para datos obtenidos con [get\\_election\\_data](#) el formato de data debe ser long para calcular [compute\\_competitiveness](#). Si data es *wide* se puede transformar con [make\\_long](#)

*(For data obtained with [get\\_election\\_data](#) long format of data is required for [compute\\_competitiveness](#). If data is in wide format you can transform it with [make\\_long](#)).*

### Value

devuelve un tibble con class "tbl\_df", "tbl", "data.frame" con el computo de competitividad de una elccion. El resultado puede ser un valor entre 0 y 1, los cuales representan falta de competencia y competitividad maxima, respectivamente

*(returns a tibble with class "tbl\_df", "tbl", "data.frame" with the computation of the level of competitiveness of an elction. The result can be a value between 0 and 1, which represent a lack of competition and maximum competitiveness, respectively).*

### See Also

[compute\\_nep](#), [compute\\_seats](#), [compute\\_disproportion](#)

### Examples

```
tucuman_dip_gral_2017  
  
tucuman_dip_gral_2017 %>%  
  compute_competitiveness()
```

---

compute\_concentration *Calcula Concentración Elecotral (Computes Electoral Concentration)*

---

### Description

Calcula el índice de concentración electoral de una elección en función del porcentaje acumulado de las dos listas más votadas

*(Computes the electoral concentration index of an election based on the accumulated percentage of the two most voted lists)*

### Usage

```
compute_concentration(data)
```

### Arguments

data            la base de datos para hacer el cálculo obtenida con [get\\_election\\_data](#)  
*(tibble downloaded with [get\\_election\\_data](#) needed to compute disproportion).#'*

**Details****REQUISITO:**

El formato de data debe ser long para calcular `compute_concentration`. Si data es *wide* se puede transformar con `make_long`

*(long format of data is required for `compute_concentration`. If data is in wide format you can transform it with `make_long`)*

**NOTA:**

el grado de concentración será sensible al nivel de agregación de los datos determinados por el parámetro `level` de `get_election_data`

*(the degree of concentration will be sensitive to the level of aggregation of the data determined by the parameter `level` of `get_election_data`).*

**Value**

Devuelve un tibble con class "tbl\_df", "tbl", "data.frame" con el cómputo de concentración. Puede tomar valores entre 0 y 1, siendo `concentration = 1` el de mayor grado (un solo partido obtiene todos los votos).

*(Returns a tibble with class "tbl\_df", "tbl", "data.frame" with concentration computation. It can take values between 0, 1, with `concentration = 1` being the highest degree (a single party gets all the votes)).*

**See Also**

[compute\\_competitiveness](#), [compute\\_seats](#), [compute\\_nep](#), [compute\\_disproportion](#)

**Examples**

```
tucuman_dip_gral_2017
tucuman_dip_gral_2017 %>%
  compute_concentration()
```

---

`compute_disproportion` *Calcula Desproporción del Sistema de Partidos (Computes Party System Disproportion)*

---

**Description**

Función que calcula el índice de desproporción del sistema de partidos midiendo la distancia relativa entre proporción de votos y de bancas obtenidos por los partidos políticos (Gallagher por defecto)

*(Function that computes the party system disproportion index)*

**Usage**

```
compute_disproportion(data, formula = "gallagher")
```

## Arguments

|         |  |
|---------|--|
| data    | la base de datos para hacer el cálculo obtenida con <a href="#">get_election_data</a> ( <i>tibble downloaded with <a href="#">get_election_data</a> needed to compute disproportion</i> ). |
| formula | variante de cálculo elegido ('gallagher', 'cox_shugart', 'lijphart' o 'rae'). Por defecto es la formula de Gallagher   |

## Details

Una discusión sobre distintas formulas para el cálculo puede encontrarse en "[Taagepera and Grofman \(2009\)](#)"

(A discussion of different ways to compute it can be found at "[Taagepera and Grofman \(2009\)](#)")

### REQUISITO:

El formato de data debe ser long para calcular [compute\\_disproportion](#). Si data es wide se puede transformar con [make\\_long](#)

(*long format of data is required for [compute\\_disproportion](#). If data is in wide format you can transform it with [make\\_long](#)*)

### NOTA:

para calcular el grado de desproporción el parámetro level de [get\\_election\\_data](#) debe ser el que descarga por defecto: "provincia"

(*To compute the degree of disproportion, the parameter level of [get\\_election\\_data](#) must be "provincia" (default)*).

## Value

Devuelve un tibble con class "tbl\_df", "tbl", "data.frame" con el cómputo

(*Returns a tibble with class "tbl\_df", "tbl", "data.frame" with the computation*).

## See Also

[compute\\_competitiveness](#), [compute\\_seats](#), [compute\\_nep](#)

## Examples

```
tucuman_dip_gral_2017  
  
tucuman_dip_gral_2017 %>%  
  electorAr::compute_disproportion()
```

compute\_nep

Calcula Numero Efectivo de Partidos Politicos - NEP

**Description**

Función que calcula el NEP: indicador que provee un número "ajustado" de partidos políticos en un sistema de partidos

*(Function that computes NEP: indicator that provides a "tight" number of political parties in a party system)*

**Usage**

```
compute_nep(data, index = "All")
```

**Arguments**

**data** la base de datos para hacer el cálculo obtenida con [get\\_election\\_data](#) (tibble downloaded with [get\\_election\\_data](#) needed to compute nep).

**index** un character con la fórmula elegida: "Laakso-Taagepera", "Golosoov" o ambas -opción por defecto- (a character with the chosen formula: 'Laakso-Taagepera', 'Golosoov' or both -default value).

**Details**

*(Computes Effective Number of Political Parties)*

El computo solo se hace a partir de la cantidad de votos de cada lista y no de las bancas

*(The computation is only made from the number of votes for each ballot and not from the corresponding legislativa seats).*

Implementación de las fórmulas "**Laakso-Taagepera**" y "**Golosoov**" donde  $p_1$  es el porcentaje de votos de una lista  $i$  y  $p_{max}$  es el porcentaje de votos que sacó la lista más votada

*(Implementation of the "Laakso-Taagepera" and "Golosoov" formulas, where  $p_1$  is vote percentage for a list  $i$  and  $p_{max}$  the percentage for the most voted party).*

*Laakso – Tagepera:*

$$NEP_{Laakso-Tagepera} = \frac{1}{\sum_i^n p_i^2}$$

*Golosoov:*

$$NEP_{Golosoov} = \frac{p_i}{\sum_i^n p_i + p_{max}^2 - p_i^2}$$

**REQUISITO:**

El formato de data debe ser long para calcular [compute\\_nep](#). Si data es wide se puede transformar con [make\\_long](#)

(long format of data is required for `compute_nep`. If data is in wide format you can transform it with `make_long`)

**NOTA:**

el parámetro `level` de `get_election_data` determina el nivel de agregación sobre el que se computa el NEP: provincia, departamento o circuito

(level at `get_election_data` determines aggregation on which NEP calculation will be made: provincia, departamento or circuito).

**Value**

Devuelve un tibble con class "tbl\_df", "tbl", "data.frame" de dimensiones variables según el nivel de agrupamiento de data con el cómputo del número ajustado de fuerzas en un sistema de partidos

(Returns a tibble with class "tbl\_df", "tbl", "data.frame", of variable dimensions according to the grouping level of data with a "tight" number of political parties in a party system as outcome).

Como mínimo tres variables (At least three variables):

\*`codprov` que es el agrupamiento de más alto nivel (provincial). Se pueden sumar otros niveles: `coddepto` para nivel departamental y `circuito` para el nivel que le sigue

(`codprov` which is the highest level grouping -provincial. Other levels can be added: `coddepto` for departmental level and `circuito` for the level that follows).

- `value`: el valor calculado del NEP (NEP computation)
- `index`: el método utilizado para calcular el NEP (NEP methodology used for computation)

**See Also**

[compute\\_competitiveness](#), [compute\\_seats](#), [compute\\_disproportion](#)

**Examples**

```
tucuman_dip_gral_2017
tucuman_dip_gral_2017 %>%
  electorAr::compute_nep()
```

---

compute\_seats

Calcula número de bancas legislativas a repartir en una elección

---

**Description**

Función que calcula la distribución proporcional de escaños para la categoría Diputado Nacional y de mayoría/minoría para Senador Nacional en función de votos obtenidos

(Function that computes propotional allocation of Diputados and Senadores seats)

## Usage

```
compute_seats(data)
```

## Arguments

**data** un `data.frame` con los resultados de una elección para agregados a nivel provincial  
(a *data.frame with aggregate electoral results at provincial level*).

## Details

(*Computes allocation of legislative seats*)

1. La distribución de escaños esta regida por la formula del sistema *D'Hondt* para Diputados y mayoría/minoría para Senadores.
2. La cantidad de escaños de cada provincia dependen de su población con un mínimo de 5 por provincia. En caso de Senadores se asignan 2 al de mayor votos y 1 al segundo.
3. En el caso de Diputados, La renovación de bancas de cada provincia se realiza por mitades cada dos años. Cuando la cantidad de escaños que corresponden a una provincia es impar las mismas eligen un diputado más en uno de los turnos: o concurrentes con elecciones presidenciales, o en elecciones de mitad de termino presidencial. En el caso de Senadores su mandato es de 6 años y se renuevan parcialmente por grupos de 8 provincias.

## See Also

[compute\\_nep](#), [compute\\_competitiveness](#), [compute\\_disproportion](#)

## Examples

```
electorAr::get_election_data(district = "caba",  
                             category = "dip",  
                             round = "gral",  
                             year = 2007) -> caba_dip_2007
```

```
caba_dip_2007
```

```
compute_seats(data = caba_dip_2007)
```

---

electorAr

electorAr *package*

---

## Description

Caja de Herramientas para el procesamiento de discursos presidenciales de Argentina See the README on [Github](#)



---

get\_election\_data      *Descarga bases de datos electorales* (Download electoral data)

---

### Description

Función que descarga datos electorales de escrutinios provisorios nacionales desde 2003  
(*Function that downloads national electoral data since 2003 from provisional count*)

### Usage

```
get_election_data(  
  district = NULL,  
  category = NULL,  
  round = NULL,  
  year = NULL,  
  level = "provincia",  
  long = TRUE,  
  raw = FALSE  
)
```

### Arguments

|          |   |
|----------|---|
| district | un character con código para Argentina y las 24 provincias<br>( <i>a named character with code for Argentina and the 24 provinces</i> ).  |
| category | un character para la categoría electoral: diputado dip, senador sen o presidente presi<br>( <i>a character with a name for the electoral category: deputy dip, senator sen or president presi</i> ).  |
| round    | un character para tipo de elección: primaria paso o general gral<br>( <i>a character with a name for the election round: primary paso or general gral</i> ).  |
| year     | un integer para el año de elección ( <i>an integer for the year of choice</i> ).  |
| level    | un character para seleccionar level de agregación de los resultados: provincia, departamento o circuito ( <i>a character to select the level of aggregation of the results: province provincia, department -departamento or electoral precincts circuito</i> ). |
| long     | un booleano para estructura de los datos. Por default long = FALSE<br>( <i>a boolean for data structure. By default long = FALSE</i> ).   |
| raw      | un booleano TRUE/FALSE que define si descargar base de datos desagregada a nivel MESA o no<br>( <i>a TRUE/FALSE boolean to define whether to download disaggregated data at BALLOT level or not</i> )   |

**Value**

devuelve un tibble con class "grouped\_df", "tbl\_df", "tbl", "data.frame" con los resultados de una eleccion determinada

*(returns a tibble with electoral results of interest with class "grouped\_df", "tbl\_df", "tbl", "data.frame").*

**See Also**

[get\\_multiple\\_elections](#)

**Examples**

```
get_election_data(district = "caba",
                 category = "dip",
                 round = "paso",
                 year = 2011,
                 level = "provincia",
                 long = TRUE, raw = FALSE)
```

---

get\_election\_results *Descarga resultados electorales* (Download electoral results)

---

**Description**

Función que descarga resultados electorales agregados, nacionales y provinciales desde 2007

*(Function that downloads national electoral data since 2007)*

**Usage**

```
get_election_results(
  district = NULL,
  category = NULL,
  round = NULL,
  year = NULL
)
```

**Arguments**

|          |  |
|----------|--|
| district | un character con código para Argentina y las 24 provincias<br><i>(a named character with code for Argentina and the 24 provinces).</i>   |
| category | un character para la categoría electoral: diputado dip, senador sen o presidente presi<br><i>(a character with a name for the electoral category: deputy dip, senator sen or president presi).</i> |

round un character para tipo de elección: primaria paso o general gral  
(*a character with a name for the election round: primary paso or general gral*).

year un integer para el año de eleccion  
(*an integer for the year of choice*).

**Value**

devuelve un tibble con class "spec\_tbl\_df", "tbl\_df", "tbl", "data.frame" con los resultados de una elección determinada

(*returns a tibble with electoral results of interest with class "spec\_tbl\_df", "tbl\_df", "tbl", "data.frame"*).

**Examples**

```
get_election_data(district = "caba",
                 category = "dip",
                 round = "paso",
                 year = 2011,
                 level = "provincia",
                 long = TRUE, raw = FALSE)
```

---

get\_multiple\_elections

*Descarga resultados de múltiples elecciones* (Download multiple election electoral data)

---

**Description**

Esta función permite descargar resultados de una multiplicidad de elecciones al mismo tiempo.

(*Function that downloads multiple national electoral data with one call*)

**Usage**

```
get_multiple_elections(
  data,
  source = NULL,
  unnest = FALSE,
  level = "provincia"
)
```

**Arguments**

data data.frame con tantas filas como elecciones se quiere descargar y cuatro columnas con las siguientes variables: *district, category, round, year*  
(*data.frame with as many rows as elections you want to download and four columns with the following variables:district, category, round, year*).

|        |  |
|--------|--|
| source | Fuente de los datos. Las opciones son 'data' para datos <a href="#">get_election_data</a> y 'results' para <a href="#">get_election_results</a>  |
| unnest | un booleano que devuelve los datos anidados cuando TRUE agrupando cada elección o un data.frame cuando es FALSE que incluye una variable de id de la elección<br><i>(a boolean that returns nested data when TRUE, grouping each election or a data.frame when FALSE that includes an election id variable).</i> |
| level  | parámetro para definir el nivel de agregación de los datos que se quieren descargar ('provincia', 'departamento', 'circuito'). Por defecto es provincia<br><i>(parameter to define the level of aggregation of the data to be downloaded ('province', 'department', 'circuit'). Default is province).</i>        |

**Value**

devuelve un tibble con class "grouped\_df", "tbl\_df", "tbl", "data.frame" con los resultados de las elecciones seleccionadas, con tantas filas como elecciones se consultaron y dos columnas: *id* de la elección construido como concatenación de los parámetros year\_category\_round\_year; *election* contiene un listado de tibbles con los resultados agregados a nivel provincial para cada elección

*(returns a tibble of class "grouped\_df", "tbl\_df", "tbl", "data.frame" with as many rows as elections requested and two columns: id of the election build as a concatenation of the parameters year\_category\_round\_year; election contains a list of tibbles with electoral results aggregated at the provincial level for each each row).*

**See Also**

[get\\_election\\_data](#)

**Examples**

```

electorAr::show_available_elections(source = 'data') %>%
  dplyr::filter(district == "caba",
               category == "dip",
               round == "paso") -> caba_paso_diputados

caba_paso_diputados

get_multiple_elections(data = caba_paso_diputados, source = 'data')
```

---

get\_names

*Obtiene nombres de listas (Get party names)*

---

**Description**

Función que agrega el nombre de las listas o partidos como columna a un tibble obtenido con [get\\_election\\_data\(long = TRUE\)](#) (Function that adds party labels as a column to a tibble obtained with [get\\_election\\_data\(long = TRUE\)](#)).

**Usage**

```
get_names(data)
```

**Arguments**

`data` un tibble descargado con `get_election_data(long = TRUE)` guardado como objeto en el Environment (A `get_election_data(long = TRUE)` tibble saved as an object in the Environment).

**Details**

El formato de data debe ser long para poder obtener nombres de listas con `get_names`. Si data es wide se puede transformar con `make_long` (long format of data is required to get party labels with `get_names`. If data is in wide format you can transform it with `make_long`).

**Value**

Devuelve el data set original con una columna extra con la identificación de las listas o partidos políticos. Como el objeto de entrada, este es class "tbl\_df", "tbl", "data.frame" (it retruns the original data set with a binded column with political parties names. As the original input the object is of class "tbl\_df", "tbl", "data.frame").

**Examples**

```
tucuman_dip_gral_2017
tucuman_dip_gral_2017 %>%
  get_names()
```

---

|           |  |
|-----------|--|
| make_long | Transforma tibble a formato largo (Transforms a tibble into long format) |
|-----------|--|

---

**Description**

Función auxiliar que transforma el tibble wide obtenido con `get_election_data` a long (Auxiliary function that transforms a wide tibble obtained with `get_election_data` to long format).

**Usage**

```
make_long(data)
```

**Arguments**

`data` es el tibble que devuelve `get_election_data` con `long = FALSE` como parámetro (tibble output from `get_election_data` with `long = FALSE` as a parameter).

**Details**

El formato de data debe ser long tanto para calcular `compute_nep` y `compute_competitiveness`, como para obtener nombre de listas con `get_names` (*long format of data is required for `compute_nep`, `compute_competitiveness` and `get_names`*).

**Value**

transforma data a formato alargado utilizando `pivot_longer` sin cambiar class "tbl\_df", "tbl", "data.frame" de origen pero aumentando el número de filas y reduciendo el de columnas (*It makes data longer with `pivot_longer`. It returns data of class "tbl\_df", "tbl", "data.frame" as the original but increasing the number of rows and decreasing the number of columns*).

---

show\_available\_elections

*Diccionario de elecciones disponibles (Elections collection)*

---

**Description**

Función que devuelve un tibble con los parámetros necesarios para llamar elecciones con `get_election_data` o `get_election_results` (*Function that returns a tibble with the necessary parameters to call elections with `get_election_data` or `get_election_results`*).

**Usage**

```
show_available_elections(source = NULL, viewer = FALSE)
```

**Arguments**

|        |   |
|--------|---|
| source | Fuente de consulta. Las opciones son 'data' para <code>get_election_data</code> y 'results' para <code>get_election_results</code>  |
| viewer | Por defecto es FALSE. Cuando TRUE devuelve una tabla en el <i>Viewer</i> de <i>RStudio</i> ( <i>The default is FALSE. When TRUE it returns a table in RStudio Viewer</i> ). |

**Value**

El objeto de salida es un data set con los id de elecciones necesarios como parámetros para usar con `get_election_data`. Cuando el parámetro es `viewer = FALSE`, devuelve un tibble con class "tbl\_df", "tbl", "data.frame", y cuando es `viewer = TRUE` devuelve un objeto con class "datatables", "htmlwidget" (*The output is a data set with elections id needed as parameters in `get_election_data`. When parameter is set to `viewer = FALSE` it returns a tibble and when it is `viewer = TRUE` it returns an object of class "datatables", "htmlwidget"*).

**Examples**

```
show_available_elections(source= 'data', viewer = FALSE)
```

---

tucuman\_dip\_gral\_2017 *Data set de muestra*

---

**Description**

Un tibble con los resultados agregados de la elección general para Diputado Nacional por la provincia de Tucuman en 2017

**Usage**

```
tucuman_dip_gral_2017
```

**Format**

Un data frame con 6 filas y 8 variables:

**category** Categoría Electoral  
**round** Ronda o Truno Electoral  
**year** Fecha de eleccion  
**codprov**Codigo de Provincia  
**name\_prov** Nombre de Provincia  
**electores** Cantidad de Electores  
**listas** Codigo de lista o partido  
**votos** Cantidad de votos absolutos ...

# Index

## \* datasets

tucuman\_dip\_gral\_2017, 15

compute\_competitiveness, 2, 3–5, 7, 8, 14

compute\_concentration, 3, 4

compute\_disproportion, 3, 4, 4, 5, 7, 8

compute\_nep, 3–6, 6, 7, 8, 14

compute\_seats, 3–5, 7, 7

electorAr, 8

get\_election\_data, 2–7, 9, 12–14

get\_election\_results, 2, 10, 12, 14

get\_multiple\_elections, 10, 11

get\_names, 12, 13, 14

make\_long, 3–7, 13, 13

pivot\_longer, 14

show\_available\_elections, 14

tucuman\_dip\_gral\_2017, 15