

Package ‘sivirep’

December 4, 2024

Title Data Wrangling and Automated Reports from 'SIVIGILA' Source

Type Package

Version 1.0.1

Description Data wrangling, pre-processing, and generating automated reports from Colombia's epidemiological surveillance system, 'SIVIGILA' <<https://portalsivigila.ins.gov.co/>>. It provides a customizable R Markdown template for analysis and automatic generation of epidemiological reports that can be adapted to local, regional, and national contexts. This tool offers a standardized and reproducible workflow that helps to reduce manual labor and potential errors in report generation, improving their efficiency and consistency.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports config, dplyr, epitrix, ggplot2, httr2, kableExtra, readxl, rlang, sf, showtext, stats, stringr, sysfonts, tools, utils, xml2

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

Depends R (>= 4.0.0)

Config/Needs/website r-lib/pkgdown, epiverse-trace/epiversetheme

VignetteBuilder knitr

Language es-ES

URL <https://epiverse-trace.github.io/sivirep/>,
<https://github.com/epiverse-trace/sivirep>

BugReports <https://github.com/epiverse-trace/sivirep/issues>

Config/testthat/edition 3

NeedsCompilation no

Author Geraldine Gómez-Millán [aut, cre, ctb]
 (<<https://orcid.org/0009-0007-8701-0568>>),
 Zulma M. Cucunubá [aut, ctb] (<<https://orcid.org/0000-0002-8165-3198>>),
 Jennifer A. Mendez-Romero [aut, ctb]
 (<<https://orcid.org/0009-0001-6138-0225>>),
 Claudia Huguett-Aragón [aut, ctb]
 (<<https://orcid.org/0000-0002-9814-2386>>),
 Hugo Gruson [ctb] (<<https://orcid.org/0000-0002-4094-1476>>),
 Juanita Romero-Garcés [ctb] (<<https://orcid.org/0000-0001-9796-6626>>),
 Jaime Pavlich-Mariscal [ctb] (<<https://orcid.org/0000-0002-3892-6680>>),
 Laura Gómez Bermeo [ctb] (<<https://orcid.org/0000-0003-4028-2893>>),
 Andrés Moreno [ctb] (<<https://orcid.org/0000-0001-9266-731X>>),
 Miguel Gámez [ctb],
 Johan Calderón [ctb],
 Lady Flórez-Tapiero [ctb],
 Verónica Tangarife-Arredondo [ctb],
 Gerard Alarcon [ctb],
 International Development Research Center (IDRC) [fnd],
 Pontificia Universidad Javeriana [cph]

Maintainer Geraldine Gómez-Millán <geraldine.gomez@javeriana.edu.co>

Repository CRAN

Date/Publication 2024-12-03 23:10:02 UTC

Contents

| | |
|-------------------------------------|----|
| agrupar_area_geo | 4 |
| agrupar_cols_casos | 4 |
| agrupar_dpto | 5 |
| agrupar_edad | 6 |
| agrupar_edad_sex | 7 |
| agrupar_eventos | 8 |
| agrupar_fecha_inisintomas | 9 |
| agrupar_mpio | 9 |
| agrupar_per_etn | 11 |
| agrupar_rango_edad | 11 |
| agrupar_semanaepi | 13 |
| agrupar_sex | 13 |
| agrupar_sex_semanaepi | 14 |
| agrupar_tipo_caso | 15 |
| agrupar_top_area_geo | 16 |
| agrupar_years | 17 |
| calcular_incidencia | 17 |
| calcular_incidencia_geo | 19 |
| calcular_incidencia_sex | 21 |
| convert_edad | 22 |
| dengue2020 | 23 |
| divipoladata | 26 |

| | |
|--------------------------------------|----|
| estandarizar_geo_cods | 26 |
| geo_filtro | 27 |
| import_data_event | 28 |
| import_geo_cods | 29 |
| import_pob_incidencia | 29 |
| import_pob_proyecciones | 30 |
| import_pob_riesgo | 31 |
| limpiar_data_sivigila | 32 |
| limpiar_edad_event | 33 |
| limpiar_encabezado | 34 |
| limpiar_fecha_event | 34 |
| limpiar_val_atipic | 35 |
| list_events | 36 |
| obtener_cond_inciden_event | 36 |
| obtener_dptos | 37 |
| obtener_fila_mas_casos | 37 |
| obtener_info_depts | 38 |
| obtener_meses_mas_casos | 39 |
| obtener_nombre_dpto | 40 |
| obtener_nombre_mpio | 41 |
| obtener_text_sex | 42 |
| obtener_tip_ocurren_geo | 42 |
| obtener_val_config | 43 |
| plot_area_geo | 44 |
| plot_dptos | 44 |
| plot_edad | 45 |
| plot_edad_sex | 46 |
| plot_fecha_inisintomas | 47 |
| plot_map | 48 |
| plot_mpios | 50 |
| plot_per_etn | 51 |
| plot_sex | 52 |
| plot_sex_semanaepi | 53 |
| plot_tabla_incidencia_geo | 54 |
| plot_tabla_incidencia_sex | 55 |
| plot_tabla_tipos_event | 56 |
| plot_tipo_caso | 56 |
| plot_tipo_caso_years | 57 |
| plot_top_area_geo | 58 |
| plot_years | 59 |

agrupar_area_geo *Agrupar por área geográfica*

Description

Función que agrupa los datos de una enfermedad o evento por área geográfica.

Usage

```
agrupar_area_geo(data_event, col_area = "area", porcentaje = FALSE)
```

Arguments

| | |
|------------|--|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| col_area | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene las áreas geográficas en los datos de la enfermedad o evento; su valor por defecto es "area". |
| porcentaje | Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'FALSE'. |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por área geográfica.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_area_geo(
  data_event = data_limpia,
  col_area = "area",
  porcentaje = FALSE
)
```

agrupar_cols_casos *Agrupar por columnas y casos*

Description

Función que agrupa los datos de una enfermedad o evento por nombre de columna(s) y número de casos.

Usage

```
agrupar_cols_casos(data_event, nomb_cols, porcentaje = FALSE, estandar = TRUE)
```

Arguments

| | |
|------------|--|
| data_event | Un 'data.frame' que contiene los datos de una enfermedad o evento. |
| nomb_cols | Un 'character' (cadena de caracteres) o 'array (arreglo) de character' que contiene el nombre de la(s) columna(s) en los datos de la enfermedad o evento. |
| porcentaje | Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'FALSE'. |
| estandar | Un 'logical' (TRUE o FALSE) que indica si se debe utilizar el estándar de agrupación de los datos del evento o enfermedad propuesto por el paquete, es decir, que se incluyan estas columnas o variables como parte del resultado 'c("cod_eve", "nombre_evento", "ano")'; su valor por defecto es 'TRUE', si su valor es 'FALSE' agrupará los datos solamente por las columnas o variables enviadas en el parámetro 'nomb_cols'. |

Value

Un 'data.frame' con los datos de una enfermedad o evento agrupados por el nombre de la(s) columna(s) y el número de casos.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_cols_casos(
  data_event = data_limpia,
  nomb_cols = "sexo",
  porcentaje = TRUE
)
agrupar_cols_casos(
  data_event = data_limpia,
  nomb_cols = c("sexo", "semana")
)
```

agrupar_dpto

Agrupar por departamento y casos

Description

Función que agrupa los datos por códigos de departamento y número de casos.

Usage

```
agrupar_dpto(data_event, col_dpto = "cod_dpto_o", porcentaje = FALSE)
```

Arguments

| | |
|------------|---|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| col_dpto | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene los códigos de los departamentos en los datos de la enfermedad o evento; su valor por defecto es "cod_dpto_o". |
| porcentaje | Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'FALSE'. |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por códigos de departamento y número de casos.

Examples

```
data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_dpto(
  data_event = data_limpiar,
  col_dpto = "cod_dpto_o",
  porcentaje = FALSE
)
```

agrupar_edad

Agrupar por edad y casos

Description

Función que agrupa los datos de una enfermedad o evento por edad y número de casos.

Usage

```
agrupar_edad(
  data_event,
  col_edad = "edad",
  interval_edad = 10,
  porcentaje = FALSE
)
```

Arguments

| | |
|---------------|---|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| col_edad | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene las edades en los datos de la enfermedad o evento; su valor por defecto es "edad". |
| interval_edad | Un 'numeric' (numérico) que contiene el intervalo del rango de edades; su valor por defecto es '10'. |

porcentaje Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'FALSE'.

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por edad y número de casos.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_edad(
  data_event = data_limpia,
  col_edad = "edad",
  porcentaje = FALSE
)
```

agrupar_edad_sex *Agrupar por edades, sexo y casos*

Description

Función que agrupa los datos de una enfermedad o evento por edades, sexo y número de casos.

Usage

```
agrupar_edad_sex(
  data_event,
  col_edad = "edad",
  col_sex = "sexo",
  porcentaje = TRUE,
  interval_edad = 10
)
```

Arguments

data_event Un 'data.frame' que contiene los datos de la enfermedad o evento.

col_edad Un 'character' (cadena de caracteres) con el nombre de la columna que contiene las edades en los datos de la enfermedad o evento; su valor por defecto es "edad".

col_sex Un 'character' (cadena de caracteres) con el nombre de la columna que contiene el sexo en los datos de la enfermedad o evento; su valor por defecto es "sexo".

porcentaje Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'TRUE'.

interval_edad Un 'numeric' (numérico) que contiene el intervalo del rango de edades; su valor por defecto es '10'.

Value

Un 'data.frame' con los datos de enfermedades agrupados por edades, sexo y número de casos.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_edad_sex(
  data_event = data_limpia,
  col_edad = "edad",
  col_sex = "sexo",
  porcentaje = TRUE
)
```

agrupar_eventos

Agrupar por tipo de enfermedad o evento

Description

Función que agrupa los casos por tipo de enfermedad o evento.

Usage

```
agrupar_eventos(data_event, col_event = "cod_eve")
```

Arguments

| | |
|------------|--|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| col_event | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene los códigos de los eventos o de las enfermedades en los datos; su valor por defecto es "cod_eve". |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por tipo.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_eventos(
  data_event = data_limpia,
  col_event = "cod_eve"
)
```

`agrupar_fecha_inisintomas`*Agrupar por fecha de inicio de síntomas y casos*

Description

Función que agrupa los datos de una enfermedad o evento por fecha de inicio de síntomas y número de casos.

Usage

```
agrupar_fecha_inisintomas(data_event, col_fecha = "ini_sin")
```

Arguments

| | |
|-------------------------|--|
| <code>data_event</code> | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| <code>col_fecha</code> | Un 'character' (cadena de caracteres) con el nombre de la columna de los datos de la enfermedad o evento que contiene las fechas de inicio de síntomas; su valor por defecto es "ini_sin". |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por fecha de inicio de síntomas y número de casos.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_fecha_inisintomas(
  data_event = data_limpia,
  col_fecha = "ini_sin"
)
```

`agrupar_mpio`*Agrupar por municipios y casos*

Description

Función que agrupa los datos de una enfermedad o evento por código de municipios y número de casos.

Usage

```
agrupar_mpio(  
  data_event,  
  dpto = NULL,  
  col_mpio = "cod_mun_o",  
  porcentaje = FALSE  
)
```

Arguments

| | |
|------------|---|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| dpto | Un 'character' (cadena de caracteres) o 'numeric' (numérico) que contiene el nombre del departamento; su valor por defecto es 'NULL'. |
| col_mpio | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene los códigos de los municipios en los datos de la enfermedad o evento; su valor por defecto es "cod_mun_o". |
| porcentaje | Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'FALSE'. |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por códigos de municipios y número de casos.

Examples

```
data(dengue2020)  
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)  
agrupar_mpio(  
  data_event = data_limpiar,  
  dpto = "ANTIOQUIA",  
  col_mpio = "cod_mun_o",  
  porcentaje = FALSE  
)  
agrupar_mpio(  
  data_event = data_limpiar,  
  dpto = "05",  
  col_mpio = "cod_mun_o",  
  porcentaje = FALSE  
)  
agrupar_mpio(  
  data_event = data_limpiar,  
  dpto = 05,  
  col_mpio = "cod_mun_o",  
  porcentaje = TRUE  
)
```

| | |
|-----------------|--|
| agrupar_per_etn | <i>Agrupar por la pertenencia étnica</i> |
|-----------------|--|

Description

Función que agrupa los casos por la pertenencia étnica.

Usage

```
agrupar_per_etn(data_event, cols_etn = "per_etn", porcentaje = TRUE)
```

Arguments

| | |
|------------|---|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| cols_etn | Un 'character' (cadena de caracteres) o un 'array' de 'character' con el nombre de la(s) columna(s) que contiene(n) la pertenencia étnica en los datos de la enfermedad o evento; su valor por defecto es "per_etn" |
| porcentaje | Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'TRUE'. |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por la pertenencia étnica.

Examples

```
data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_per_etn(
  data_event = data_limpiar,
  cols_etn = "per_etn"
)
```

| | |
|--------------------|--|
| agrupar_rango_edad | <i>Agrupar por rango de edad y casos</i> |
|--------------------|--|

Description

Función que agrupa los datos de una enfermedad o evento por rango de edad y número de casos.

Usage

```
agrupar_rango_edad(
  data_event,
  col_edad = "edad",
  col_adicional = NULL,
  min_val,
  max_val,
  paso,
  porcentaje = TRUE
)
```

Arguments

| | |
|----------------------------|---|
| <code>data_event</code> | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| <code>col_edad</code> | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene las edades en los datos de la enfermedad o evento. |
| <code>col_adicional</code> | Un 'character' (cadena de caracteres) con el nombre de la columna adicional para agrupar con las edades en los datos de la enfermedad o evento; su valor por defecto es 'NULL'. |
| <code>min_val</code> | Un 'numeric' (numérico) que contiene la edad mínima con la que debe iniciar el rango de edades. |
| <code>max_val</code> | Un 'numeric' (numérico) que contiene la edad máxima con la que debe finalizar el rango de edades. |
| <code>paso</code> | Un 'numeric' (numérico) que contiene el valor del paso para generar el rango de edades. |
| <code>porcentaje</code> | Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'TRUE'. |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por el rango de edad y número de casos.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
data_edad <- agrupar_cols_casos(
  data_event = data_limpia,
  c("edad", "semana"),
  porcentaje = TRUE
)
agrupar_rango_edad(
  data_event = data_edad,
  col_edad = "edad",
  min_val = 0,
  max_val = max(data_edad$edad, na.rm = TRUE),
  paso = 10,
```

```
    porcentaje = TRUE  
  )
```

agrupar_semanaepi *Agrupar por semana epidemiológica y casos*

Description

Función que agrupa los datos de una enfermedad o evento por semana epidemiológica y número de casos.

Usage

```
agrupar_semanaepi(data_event, col_semanaepi = "semana")
```

Arguments

`data_event` Un 'data.frame' que contiene los datos de una enfermedad o evento.
`col_semanaepi` Un 'character' (cadena de caracteres) con el nombre de la columna que contiene las semanas epidemiológicas en los datos de la enfermedad o evento; su valor por defecto es "semana".

Value

Un 'data.frame' con los datos de una enfermedad o evento agrupados por semana epidemiológica y número de casos.

Examples

```
data(dengue2020)  
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)  
agrupar_semanaepi(  
  data_event = data_limpia,  
  col_semanaepi = "semana"  
)
```

agrupar_sex *Agrupar por sexo y casos*

Description

Función que agrupa los datos de una enfermedad o evento por sexo y número de casos.

Usage

```
agrupar_sex(data_event, col_sex = "sexo", porcentaje = TRUE)
```

Arguments

| | |
|------------|--|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| col_sex | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene el sexo en los datos de la enfermedad o evento; su valor por defecto es "sexo". |
| porcentaje | Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'TRUE'. |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por sexo y número de casos.

Examples

```
data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_sex(
  data_event = data_limpiar,
  col_sex = "sexo",
  porcentaje = TRUE
)
```

agrupar_sex_semanaepi *Agrupar por sexo, semana epidemiológica y casos*

Description

Función que agrupa los datos de enfermedades por sexo, semana epidemiológica y número de casos.

Usage

```
agrupar_sex_semanaepi(
  data_event,
  cols_sex = c("sexo", "semana"),
  porcentaje = TRUE
)
```

Arguments

| | |
|------------|--|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| cols_sex | Un 'character' (cadena de caracteres) o 'array' (arreglo) de 'character' con el nombre de la(s) columna(s) que contienen el sexo y las semanas epidemiológicas; su valor por defecto es 'c("sexo", "semana")'. |
| porcentaje | Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'TRUE'. |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por sexo, semana epidemiológica y número de casos.

Examples

```
data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_sex_semanaepi(
  data_event = data_limpiar,
  cols_sex = c("sexo", "semana"),
  porcentaje = TRUE
)
```

| | |
|-------------------|--|
| agrupar_tipo_caso | <i>Agrupar por la clasificación inicial del caso</i> |
|-------------------|--|

Description

Función que agrupa los casos por la clasificación inicial del caso.

Usage

```
agrupar_tipo_caso(data_event, cols_tipo = "tip_cas")
```

Arguments

| | |
|------------|--|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| cols_tipo | Un 'character' (cadena de caracteres) o 'array' (arreglo) de 'character' con el nombre de las columna(s) que contiene la clasificación inicial del caso en los datos de la enfermedad o evento; su valor por defecto es "tip_cas". |

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por la clasificación inicial del caso y/u otras variables como los años.

Examples

```
data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_tipo_caso(
  data_event = data_limpiar,
  cols_tipo = "tip_cas"
)
```

agrupar_top_area_geo *Agrupar por área geográfica a nivel departamental o municipal*

Description

Función que agrupa los datos de una enfermedad o evento por área geográfica a nivel departamental o municipal.

Usage

```
agrupar_top_area_geo(
  data_event,
  dpto = NULL,
  col_area = "area",
  porcentaje = FALSE,
  top = 10
)
```

Arguments

| | |
|------------|--|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| dpto | Un 'character' (cadena de caracteres) que contiene el nombre del departamento; su valor por defecto es 'NULL'. Si se ingresa un valor en este parámetro se procederá agrupar los datos por los municipios del departamento y sus áreas geográficas. Si no se ingresa un valor en este parámetro validará si los datos ya están filtrados por algún departamento; si no lo están generará la agrupación por departamento. |
| col_area | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene las áreas geográficas en los datos de la enfermedad o evento; su valor por defecto es "cod_mun_o". |
| porcentaje | Un 'logical' (TRUE o FALSE) que indica si se debe agregar una columna con el porcentaje de casos; su valor por defecto es 'FALSE'. |
| top | Un 'numeric' (numérico) que indica la cantidad de departamentos o municipios con mayor número de casos que se deben retornar; su valor por defecto es '10'. |

Value

Un 'data.frame' con el top 10 de los datos de la enfermedad o evento agrupados por áreas geográficas y número de casos.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
agrupar_top_area_geo(
  data_event = data_limpia,
```

```
dpto = "Antioquia",  
col_area = "area",  
porcentaje = FALSE,  
top = 10  
)
```

agrupar_years*Agrupar por años de una enfermedad o evento*

Description

Función que agrupa los casos por los años de una enfermedad o evento.

Usage

```
agrupar_years(data_event, col_year = "ano")
```

Arguments

data_event Un 'data.frame' que contiene los datos de la enfermedad o evento.
col_year Un 'character' (cadena de caracteres) con el nombre de la columna que contiene los años en los datos de la enfermedad o evento; su valor por defecto es "ano".

Value

Un 'data.frame' con los datos de la enfermedad o evento agrupados por año.

Examples

```
data(dengue2020)  
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)  
agrupar_years(  
  data_event = data_limpiar,  
  col_year = "ano"  
)
```

calcular_incidencia*Calcular incidencia*

Description

Función que calcula la incidencia de una enfermedad o evento para todo Colombia, un departamento o un municipio.

Usage

```

calcular_incidencia(
  data_incidencia = NULL,
  cache = FALSE,
  ruta_dir = NULL,
  data_agrupada,
  poblacion = NULL,
  year = NULL,
  dpto = NULL,
  mpio = NULL,
  sex = NULL
)

```

Arguments

| | |
|------------------------------|---|
| <code>data_incidencia</code> | Un <code>'data.frame'</code> que contiene la población a riesgo o las proyecciones poblacionales DANE. Si este parámetro está vacío, se importará la población a riesgo o las proyecciones, dependiendo de la disponibilidad de la información; su valor por defecto es <code>'NULL'</code> . |
| <code>cache</code> | Un <code>'logical'</code> (<code>'TRUE'</code> o <code>'FALSE'</code>) que indica si la población a riesgo o las proyecciones poblacionales DANE descargadas deben ser almacenados en caché. Su valor por defecto es <code>'FALSE'</code> . |
| <code>ruta_dir</code> | Un <code>'character'</code> (cadena de caracteres) que especifica la ruta del directorio donde se almacenarán la población a riesgo o las proyecciones poblacionales DANE. Su valor por defecto es <code>'NULL'</code> . |
| <code>data_agrupada</code> | Un <code>'data.frame'</code> que contiene los datos de la enfermedad agrupados por departamento o municipio y número de casos. |
| <code>poblacion</code> | Un <code>'character'</code> (cadena de caracteres) con el tipo de población para calcular la incidencia. Puede ser <code>"riesgo"</code> para la población a riesgo o <code>"proyecciones"</code> para las proyecciones poblacionales DANE; su valor por defecto es <code>'NULL'</code> . |
| <code>year</code> | Un <code>'numeric'</code> (numérico) con el año que se debe tomar en la población a riesgo o en las proyecciones poblacionales DANE; su valor por defecto es <code>'NULL'</code> . |
| <code>dpto</code> | Un <code>'character'</code> (cadena de caracteres) o <code>'numeric'</code> (numérico) que contiene el código o nombre del departamento; su valor por defecto es <code>'NULL'</code> . |
| <code>mpio</code> | Un <code>'character'</code> (cadena de caracteres) o <code>'numeric'</code> (numérico) que contiene el código o nombre del municipio; su valor por defecto es <code>'NULL'</code> . |
| <code>sex</code> | Un <code>'character'</code> (cadena de caracteres) que especifica el sexo: <code>"F"</code> para Femenino y <code>"M"</code> para Masculino; su valor por defecto es <code>'NULL'</code> . |

Value

Un `'numeric'` con el cálculo de la incidencia para todo Colombia, un departamento, municipio o sexo específico.

Examples

```

data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
# Cálculo de la incidencia con proyecciones poblacionales por departamento
data_agrupada_mpios <- agrupar_mpio(data_limpiar, dpto = "Antioquia")
if (interactive()) {
  calcular_incidencia(
    data_agrupada = data_agrupada_mpios,
    poblacion = "proyecciones",
    dpto = "05",
    year = 2020,
    cache = TRUE
  )
}
# Cálculo de la incidencia con proyecciones poblacionales por municipio
calcular_incidencia(
  data_agrupada = data_agrupada_mpios,
  poblacion = "proyecciones",
  dpto = "Antioquia",
  mpio = "05001",
  year = 2020,
  ruta_dir = tempdir()
)
# Cálculo de la incidencia con población a riesgo para Colombia
data_agrupada_dptos <- agrupar_dpto(data_limpiar)
calcular_incidencia(
  poblacion = "riesgo",
  data_agrupada = data_agrupada_dptos,
  year = 2020,
  ruta_dir = tempdir()
)

```

calcular_incidencia_geo

Calcular incidencia según distribución geográfica

Description

Función que calcula la incidencia de una enfermedad o evento para todos los departamentos de Colombia o los municipios de un departamento.

Usage

```

calcular_incidencia_geo(
  data_incidencia = NULL,
  cache = FALSE,
  ruta_dir = NULL,
  data_agrupada,

```

```

    poblacion = NULL,
    year = NULL
  )

```

Arguments

| | |
|-----------------|--|
| data_incidencia | Un 'data.frame' que contiene las proyecciones poblacionales del DANE; su valor por defecto es 'NULL'. |
| cache | Un 'logical' ('TRUE' o 'FALSE') que indica si la población a riesgo o las proyecciones poblacionales DANE descargadas deben ser almacenados en caché. Su valor por defecto es 'FALSE'. |
| ruta_dir | Un 'character' (cadena de caracteres) que especifica la ruta del directorio donde se almacenarán la población a riesgo o las proyecciones poblacionales DANE. Su valor por defecto es 'NULL'. |
| data_agrupada | Un 'data.frame' que contiene los datos de la enfermedad agrupados por departamento o municipio y número de casos. |
| poblacion | Un 'character' (cadena de caracteres) con el tipo de población para calcular la incidencia. Puede ser "riesgo" para la población a riesgo o "proyecciones" para las proyecciones poblacionales DANE; su valor por defecto es 'NULL'. |
| year | Un 'numeric' (numérico) con el año que se debe tomar en la población a riesgo o en las proyecciones poblacionales DANE; su valor por defecto es 'NULL'. |

Value

Un 'data.frame' con el cálculo de la incidencia para todos los departamentos de Colombia o los municipios de un departamento.

Examples

```

data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
data_agrupada_mpios <- agrupar_mpio(data_limpiar, dpto = "Antioquia")
# Cálculo de la incidencia con población a riesgo por departamento
if (interactive()) {
  calcular_incidencia_geo(
    poblacion = "riesgo",
    data_agrupada = data_agrupada_mpios,
    year = 2020,
    cache = TRUE
  )
}
data_agrupada_dptos <- agrupar_dpto(data_limpiar)
# Cálculo de la incidencia con proyecciones poblacionales para Colombia
calcular_incidencia_geo(
  poblacion = "proyecciones",
  data_agrupada = data_agrupada_dptos,
  year = 2020,
  ruta_dir = tempdir()
)

```

```
)
```

```
calcular_incidencia_sex
```

```
    Calcular incidencia por sexo
```

Description

Función que calcula la incidencia de una enfermedad o evento para todos los departamentos de Colombia o los municipios de un departamento por cada sexo.

Usage

```
calcular_incidencia_sex(
  data_incidencia = NULL,
  ruta_dir = NULL,
  cache = FALSE,
  data_agrupada,
  year = NULL,
  dpto = NULL,
  mpio = NULL
)
```

Arguments

| | |
|------------------------------|---|
| <code>data_incidencia</code> | Un 'data.frame' que contiene las proyecciones poblacionales del DANE; su valor por defecto es 'NULL'. |
| <code>ruta_dir</code> | Un 'character' (cadena de caracteres) que especifica la ruta del directorio donde se almacenarán la población a riesgo o las proyecciones poblacionales DANE. Su valor por defecto es 'NULL'. |
| <code>cache</code> | Un 'logical' ('TRUE' o 'FALSE') que indica si la población a riesgo o las proyecciones poblacionales DANE descargadas deben ser almacenados en caché. Su valor por defecto es 'FALSE'. |
| <code>data_agrupada</code> | Un 'data.frame' que contiene los datos de la enfermedad agrupados por departamento o municipio y número de casos. |
| <code>year</code> | Un 'numeric' (numérico) con el año que se debe tomar en la población a riesgo o en las proyecciones poblacionales DANE; su valor por defecto es 'NULL'. |
| <code>dpto</code> | Un 'character' (cadena de caracteres) o 'numeric' (numérico) que contiene el código o nombre del departamento; su valor por defecto es 'NULL'. |
| <code>mpio</code> | Un 'character' (cadena de caracteres) o 'numeric' (numérico) que contiene el código o nombre del municipio; su valor por defecto es 'NULL'. |

Value

Un 'data.frame' con el cálculo de la incidencia para todos los departamentos de Colombia o los municipios de un departamento por sexo.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
# Cálculo de la incidencia con proyecciones poblacionales por sexo y
# departamento
data_filtrada <- geo_filtro(
  data_event = data_limpia,
  dpto = "05"
)
data_agrupada <- agrupar_sex(data_filtrada)
if (interactive()) {
  calcular_incidencia_sex(
    data_agrupada = data_agrupada,
    dpto = "05",
    year = 2020,
    cache = TRUE
  )
}
#' Cálculo de la incidencia con proyecciones poblacionales por sexo y
# municipio
data_filtrada <- geo_filtro(
  data_event = data_limpia,
  dpto = "05",
  mpio = "Medellin"
)
calcular_incidencia_sex(
  data_agrupada = data_agrupada,
  dpto = "05",
  mpio = "Medellin",
  ruta_dir = tempdir()
)
```

convert_edad

Convertir edad a años

Description

Función que convierte las edades a años según las unidades de medida del SIVIGILA.

Usage

```
convert_edad(
  data_event,
```

```

    col_edad = "edad",
    col_uni_med = "uni_med",
    uni_med = 1
  )

```

Arguments

| | |
|--------------------------|---|
| <code>data_event</code> | Un ‘data.frame’ que contiene los datos de una enfermedad o evento. |
| <code>col_edad</code> | Un ‘character’ (cadena de caracteres) con el nombre de la columna que contiene las edades en los datos de la enfermedad o evento; su valor por defecto es “edad”. |
| <code>col_uni_med</code> | Un ‘character’ (cadena de caracteres) con el nombre de la columna que contiene las unidades de medida en los datos de una enfermedad o evento; su valor por defecto es “uni_med”. |
| <code>uni_med</code> | Un ‘numeric’ (numérico) o ‘character’(cadena de caracteres) que contiene la unidad de medida a la que se debe estandarizar la edad; su valor por defecto es ‘1’. |

Value

Un ‘data.frame’ con las edades convertidas en años según las unidades de medida del SIVIGILA.

Examples

```

data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
convert_edad(
  data_event = data_limpiar,
  col_edad = "edad",
  col_uni_med = "uni_med",
  uni_med = 1
)

```

dengue2020

Datos Dengue 2020 del SIVIGILA en sivirep

Description

Datos obtenidos del SIVIGILA (Sistema de Vigilancia en Salud Pública de Colombia).

Usage

```
data(dengue2020)
```

Format

'dengue2020' Un '<data.frame>' con 65535 filas y 74 columnas:

particion Particion
consecutive Consecutivo
cod_eveCodigo del evento
fec_not Fecha de notificacion
semana Semana
ano Anio - Year
cod_pre Codigo del prestador de servicios de salud
cod_sub Codigo del prestador de servicios de salud - subindice
edad Edad
uni_med Unidad de medida de la edad
nacionalidad Nacionalidad
nombre_nacionalidad Nombre de la nacionalidad
sexo Sexo
cod_pais_o Codigo del pais de ocurrencia
cod_dpto_o Codigo del departamento de ocurrencia
cod_mun_o Codigo del municipio de ocurrencia
area Area geografica
ocupacion Ocupacion del paciente
tip_ss Tipo de regimen en salud
cod_ase Codigo de administradora
per_etn Pertenencia etnica
gru_pob Grupo poblacional
nom_grupo Nombre del grupo poblacional
estrato Estrato socioeconomico
gp_discapa Grupo poblacional - discapacitados
gp_desplaz Grupo poblacional - desplazados
gp_migrant Grupo poblacional - migrantes
gp_carcela Grupo poblacional - carcelarios
gp_gestan Grupo poblacional - gestantes
sem_ges semana gestante
gp_indigen Grupo poblacional - indigentes
gp_pobicfb Grupo poblacional - poblacion infantil a cargo del ICBF
gp_mad_com Grupo poblacional - madres comunitarias
gp_desmovi Grupo poblacional - desmovilizados
gp_psiquia Grupo poblacional - poblacion en centros psiquiatricos

gp_vic_vio Grupo poblacional - victima de violencia
gp_otros Grupo poblacional - otros
fuelle Fuente
cod_pais_r Codigo del pais de residencia
cod_dpto_r Codigo del departamento de residencia
cod_mun_r Codigo del municipio de residencia
cod_dpto_n Codigo del departamento de notificacion
cod_mun_n Codigo del municipio de notificacion
fec_con Fecha de constitucion o inicio de actividades asistenciales
ini_sin Fecha de inicio de sintomas
tip_cas Clasificacion inicial del caso
pac_hos Hospitalizado
fec_hos Fecha de hospitalizacion
con_fin Condicion final
fec_def Fecha de defuncion
ajuste seguimiento y clasificacion final del caso
fecha_nton Fecha de nacimiento
cer_def Numero del certificado de defuncion
cbmte Causa basica de defuncion o muerte
fec_arc_xl Fecha de creacion del archivo de Excel
fec_aju Fecha de ajuste
fm_fuerza Fuerza militar
fm_unidad Unidad - codigo de la unidad militar
fm_grado Grado - codigo del grado militar
confirmados Confirmados
consecutive_origen Consecutivo origen
va_sispro Sistema Integral de Informacion para la Proteccion Social
estado_final_de_caso Estado final del caso
nom_est_f_caso nom_est_f_caso
nom_upgd Nombre de la unidad primaria generadora de dato
pais_ocurrencia Pais ocurrencia
nombre_evento Nombre evento
departamento_ocurrencia Departamento ocurrencia
municipio_ocurrencia Municipio ocurrencia
pais_residencia pais residencia
departamento_residencia Departamento residencia
municipio_residencia Municipio residencia
departamento_notificacion Departamento notificacion
municipio_notificacion Municipio notificacion

Examples

```
data(dengue2020)
```

```
divipoladata
```

Códigos e información geográfica del DIVIPOLA en sivirep

Description

Datos obtenidos de la API de Datos Abiertos de Colombia.

Usage

```
data(divipoladata)
```

Format

'divipoladata' Un '<data.frame>' con 1121 filas y 5 columnas:

codigo_departamento Código de los departamentos

codigo_municipio Código de los municipios

nombre_departamento Nombre de los departamentos

nombre_municipio Nombre de los municipios

tipo_municipio_isla_area_no_municipalizada Tipo

Examples

```
data(divipoladata)
```

```
estandarizar_geo_cods
```

Estandarizar códigos geográficos de los datos de una enfermedad o evento

Description

Función que estandariza los códigos geográficos de los datos de una enfermedad o evento según la codificación del DIVIPOLA.

Usage

```
estandarizar_geo_cods(data_event)
```

Arguments

data_event Un 'data.frame' que contiene los datos de una enfermedad o evento con códigos geográficos.

Value

Un 'data.frame' con los códigos geográficos estandarizados de los datos de una enfermedad o evento según la codificación del DIVIPOLA.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
estandarizar_geo_cods(data_event = data_limpia)
```

geo_filtro

Filtrar por departamentos y municipios

Description

Función que filtra los datos de una enfermedad o evento por departamentos y municipios.

Usage

```
geo_filtro(data_event, dpto = NULL, mpio = NULL)
```

Arguments

| | |
|------------|---|
| data_event | Un 'data.frame' con los datos de una enfermedad o evento. |
| dpto | Un 'character' (cadena de caracteres) o 'numeric' (numérico) que contiene el nombre o código del departamento; valor por defecto 'NULL'. |
| mpio | Un 'character' (cadena de caracteres) o 'numeric' (numérico) que contiene el nombre o código del municipio; su valor por defecto es 'NULL'. |

Value

Un 'data.frame' con los datos filtrados con la enfermedad, departamentos y municipios seleccionados.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(data_event = dengue2020)
geo_filtro(data_event = data_limpia, dpto = "ANTIOQUIA")
geo_filtro(data_event = data_limpia, dpto = "ANTIOQUIA", mpio = "MEDELLIN")
geo_filtro(data_event = data_limpia, dpto = "05")
geo_filtro(data_event = data_limpia, dpto = "05", mpio = "05001")
geo_filtro(data_event = data_limpia, dpto = 05, mpio = 05001)
geo_filtro(data_event = data_limpia, dpto = 05, mpio = 001)
geo_filtro(data_event = data_limpia, dpto = "bogota dc", mpio = "bogota dc")
```

| | |
|-------------------|--|
| import_data_event | <i>Importar los datos de una enfermedad o evento por año desde los microdatos del SIVIGILA</i> |
|-------------------|--|

Description

Función que importa los datos de una enfermedad o evento por año desde los microdatos del SIVIGILA.

Usage

```
import_data_event(nombre_event, years, ruta_dir = NULL, cache = FALSE)
```

Arguments

| | |
|--------------|---|
| nombre_event | Un 'character' (cadena de caracteres) con el nombre de la enfermedad o evento. |
| years | Un 'numeric' (numérico) con el año o años deseado(s) para la descarga de los datos. |
| ruta_dir | Un 'character' (cadena de caracteres) que contiene la ruta del directorio donde se almacenarán los datos del evento o enfermedad. Su valor por defecto es 'NULL'. |
| cache | Un 'logical' ('TRUE' o 'FALSE') que indica si los datos descargados deben ser almacenados en caché. Su valor por defecto es 'FALSE'. |

Value

Un 'data.frame' con los datos del año de la enfermedad o evento seleccionado desde los microdatos del SIVIGILA.

Examples

```
if (interactive()) {  
  import_data_event(nombre_event = "DENGUE",  
                    years = 2020,  
                    cache = TRUE)  
  import_data_event(nombre_event = "CHAGAS",  
                    years = c(2019, 2020),  
                    ruta_dir = tempdir())  
  import_data_event(nombre_event = "CHAGAS",  
                    years = seq(2018, 2020),  
                    cache = TRUE)  
}
```

| | |
|-----------------|---|
| import_geo_cods | <i>Importar datos geográficos de Colombia</i> |
|-----------------|---|

Description

Función que importa los nombres y códigos de los departamentos y municipios de Colombia a través de una URL.

Usage

```
import_geo_cods(descargar = FALSE)
```

Arguments

descargar Un 'logical' ('TRUE' o 'FALSE') que indica si los datos deben descargarse desde la API de datos abiertos de Colombia; su valor por defecto es 'FALSE'.

Value

Un 'data.frame' con los nombres y códigos de los departamentos y municipios de Colombia.

Examples

```
import_geo_cods(descargar = FALSE)
```

| | |
|-----------------------|--|
| import_pob_incidencia | <i>Importar la población para efectuar el cálculo de la incidencia</i> |
|-----------------------|--|

Description

Función que importa la población a riesgo de un evento o enfermedad o las proyecciones poblacionales DANE desde el año 2005 hasta el 2035.

Usage

```
import_pob_incidencia(  
  poblacion = c("riesgo", "proyecciones"),  
  event,  
  year,  
  ruta_dir = NULL,  
  cache = FALSE  
)
```

Arguments

| | |
|-----------|---|
| poblacion | Un 'character' (cadena de caracteres) con el tipo de población que se desea importar. Puede ser "riesgo" para la población a riesgo del evento o "proyecciones" para las proyecciones poblacionales DANE; su valor por defecto es "riesgo". |
| event | Un 'character' (cadena de caracteres) o un 'numeric' (numérico) con el nombre o código de la enfermedad o evento. Es obligatorio para importar la población a riesgo. |
| year | Un 'numeric' (numérico) con el año deseado de la población a riesgo. Es obligatorio para importar la población a riesgo. |
| ruta_dir | Un 'character' (cadena de caracteres) que especifica la ruta del directorio donde se almacenarán la población a riesgo o las proyecciones poblacionales DANE. Su valor por defecto es 'NULL'. |
| cache | Un 'logical' ('TRUE' o 'FALSE') que indica si la población a riesgo o las proyecciones poblacionales DANE descargadas deben ser almacenados en caché. Su valor por defecto es 'FALSE'. |

Value

Un 'data.frame' con la población a riesgo o las proyecciones poblacionales DANE.

Examples

```
# Importación proyecciones poblaciones DANE
if (interactive()) {
  import_pob_incidencia(poblacion = "proyecciones", year = 2020,
                        cache = TRUE)
}
# Importación población a riesgo de Dengue del año 2020
import_pob_incidencia(poblacion = "riesgo", event = "dengue", year = 2020,
                      ruta_dir = tempdir())
```

```
import_pob_proyecciones
```

Importar las proyecciones DANE del año 2005 hasta el 2035

Description

Función que importa las proyecciones poblacionales DANE desde el año 2005 hasta el 2035.

Usage

```
import_pob_proyecciones(year, ruta_dir = NULL, cache = FALSE)
```

Arguments

| | |
|----------|---|
| year | Un 'numeric' (numérico) con el año de las proyecciones poblacionales DANE que desea importar. |
| ruta_dir | Un 'character' (cadena de caracteres) que especifica la ruta del directorio donde se almacenarán la población a riesgo o las proyecciones poblacionales DANE. Su valor por defecto es 'NULL'. |
| cache | Un 'logical' ('TRUE' o 'FALSE') que indica si la población a riesgo o las proyecciones poblacionales DANE descargadas deben ser almacenados en caché. Su valor por defecto es 'FALSE'. |

Value

Un 'data.frame' con las proyecciones poblacionales DANE.

Examples

```
import_pob_proyecciones(year = 2020, ruta_dir = tempdir())
if (interactive()) {
  import_pob_proyecciones(year = 2020, cache = TRUE)
}
```

| | |
|-------------------|---|
| import_pob_riesgo | <i>Importar la población a riesgo de un evento o enfermedad</i> |
|-------------------|---|

Description

Función que importa la población a riesgo de un evento o enfermedad para un año específico.

Usage

```
import_pob_riesgo(event, year, ruta_dir = NULL, cache = FALSE)
```

Arguments

| | |
|----------|---|
| event | Un 'character' (cadena de caracteres) o un 'numeric' (numérico) con el nombre o código de la enfermedad o evento. |
| year | Un 'numeric' (numérico) con el año deseado de la población a riesgo. |
| ruta_dir | Un 'character' (cadena de caracteres) que especifica la ruta del directorio donde se almacenarán la población a riesgo o las proyecciones poblacionales DANE. Su valor por defecto es 'NULL'. |
| cache | Un 'logical' ('TRUE' o 'FALSE') que indica si la población a riesgo o las proyecciones poblacionales DANE descargadas deben ser almacenados en caché. Su valor por defecto es 'FALSE'. |

Value

Un 'data.frame' con la población a riesgo de un año específico.

Examples

```
import_pob_riesgo(event = "Dengue", year = 2020, ruta_dir = tempdir())
if (interactive()) {
  import_pob_riesgo(event = "Dengue", year = 2020, cache = TRUE)
}
```

limpiar_data_sivigila *Limpiar datos de SIVIGILA*

Description

Función que limpia los datos seleccionados de una enfermedad o evento provenientes de la fuente SIVIGILA.

Usage

```
limpiar_data_sivigila(data_event, uni_med_edad = 1)
```

Arguments

| | |
|--------------|--|
| data_event | Un 'data.frame' que contiene los datos de una enfermedad o evento. |
| uni_med_edad | Un 'numeric' (numérico) o 'character'(cadena de caracteres) que contiene la unidad de medida a la que se debe estandarizar la edad; su valor por defecto es '1'. |

Value

Un 'data.frame' con los datos limpios de la enfermedad o evento.

Examples

```
data(dengue2020)
limpiar_data_sivigila(data_event = dengue2020)
```

| | |
|--------------------|---|
| limpiar_edad_event | <i>Limpiar las edades de los datos de una enfermedad o evento</i> |
|--------------------|---|

Description

Función que limpia y estandariza las edades de los datos de una enfermedad o evento, convirtiéndolas en años, según la clasificación del Instituto Nacional de Salud:

- No aplica = 0
- Años = 1
- Meses = 2
- Días = 3
- Horas = 4
- Minutos = 5

Usage

```
limpiar_edad_event(data_event, col_edad = "edad")
```

Arguments

| | |
|------------|---|
| data_event | Un 'data.frame' que contiene los datos de una enfermedad o evento. |
| col_edad | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene las edades en los datos de la enfermedad o evento; su valor por defecto es "edad". |

Value

Un 'data.frame' con los datos de una enfermedad o evento con las edades limpias.

Examples

```
data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
limpiar_edad_event(data_event = data_limpiar, col_edad = "edad")
```

limpiar_encabezado *Limpiar las etiquetas del encabezado*

Description

Función que limpia las etiquetas del encabezado de los datos de una enfermedad o evento.

Usage

```
limpiar_encabezado(data_event)
```

Arguments

data_event Un 'data.frame' que contiene los datos de una enfermedad o evento.

Value

Un 'data.frame' con las etiquetas del encabezado formateadas con guiones bajos (_).

Examples

```
data(dengue2020)
limpiar_encabezado(data_event = dengue2020)
```

limpiar_fecha_event *Limpiar fechas de los datos de una enfermedad o evento*

Description

Función que limpia y estandariza las fechas de los datos de una enfermedad o evento.

Usage

```
limpiar_fecha_event(
  data_event,
  year,
  format_fecha = "%Y-%m-%d",
  col_fecha = "ini_sin",
  col_comp = NULL
)
```

Arguments

| | |
|--------------|---|
| data_event | Un 'data.frame' que contiene los datos de una enfermedad o evento. |
| year | Un 'numeric' (numérico) o 'character' (cadena de caracteres) que contiene el año de los datos de una enfermedad o evento. |
| format_fecha | Un 'character' (cadena de caracteres) que contiene el formato deseado de fecha; su valor por defecto es "%AAAA-%MM-%DD". |
| col_fecha | Un 'character' (cadena de caracteres) que contiene el nombre de la columna con la fecha que se desea limpiar en los datos de la enfermedad o evento. |
| col_comp | Un 'character' (cadena de caracteres) que contiene el nombre de la columna con la cual se va a comparar la columna 'col_fecha' para limpiarla, estandarizarla o aplicar las reglas definidas. |

Value

Un 'data.frame' con las fechas limpias.

Examples

```
data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
limpiar_fecha_event(
  data_event = data_limpiar,
  year = 2020,
  format_fecha = "%Y-%m-%d",
  col_fecha = "ini_sin",
  col_comp = "fec_hos"
)
```

limpiar_val_atipic *Limpiar los valores atípicos de los datos*

Description

Función que limpia los valores atípicos de los datos de una enfermedad o evento del SIVIGILA.

Usage

```
limpiar_val_atipic(data_event)
```

Arguments

| | |
|------------|--|
| data_event | Un 'data.frame' que contiene los datos de una enfermedad o evento. |
|------------|--|

Value

Un 'data.frame' con los datos de una enfermedad o evento con los valores atípicos limpios (convertidos a 'NA').

Examples

```
data(dengue2020)
data_limpia <- limpiar_encabezado(data_event = dengue2020)
limpiar_val_atipic(data_limpia)
```

| | |
|-------------|--|
| list_events | <i>Importar enfermedades y años disponibles para su descarga desde los microdatos del SIVIGILA</i> |
|-------------|--|

Description

Función que obtiene las enfermedades y los años disponibles para su descarga desde los microdatos del SIVIGILA.

Usage

```
list_events()
```

Value

Una 'list' con las enfermedades y los años disponibles para su descarga desde los microdatos del SIVIGILA.

Examples

```
if (interactive()) {
  list_events()
}
```

| | |
|----------------------------|---|
| obtener_cond_inciden_event | <i>Obtener las condiciones para calcular la incidencia de una enfermedad o evento</i> |
|----------------------------|---|

Description

Función que obtiene las condiciones del numerador, denominador y coeficiente de multiplicación para calcular la incidencia de un evento.

Usage

```
obtener_cond_inciden_event(cod_eve)
```

Arguments

cod_eve Un 'numeric' (numérico) o 'character' (cadena de caracteres) que contiene el código de una enfermedad o evento.

Value

Un 'data.frame' con las condiciones para calcular la incidencia de una enfermedad o evento.

Examples

```
obtener_cond_inciden_event(cod_eve = 210)
```

| | |
|---------------|--|
| obtener_dptos | <i>Obtener departamentos de Colombia</i> |
|---------------|--|

Description

Función que obtiene los departamentos de Colombia.

Usage

```
obtener_dptos()
```

Value

Un 'data.frame' con los departamentos de Colombia.

Examples

```
obtener_dptos()
```

| | |
|------------------------|--|
| obtener_fila_mas_casos | <i>Obtener la fila con mayor número de casos</i> |
|------------------------|--|

Description

Función que obtiene la fila con el mayor número de casos.

Usage

```
obtener_fila_mas_casos(data_event, nomb_col = "casos", porcentaje = TRUE)
```

Arguments

| | |
|------------|---|
| data_event | Un 'data.frame' que contiene los datos de la enfermedad o evento. |
| nomb_col | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene el número de casos en los datos de la enfermedad o evento. |
| porcentaje | Un 'logical' ('TRUE' o 'FALSE') que indica si se requiere agregar un porcentaje de casos como columna. |

Value

Un 'data.frame' que contiene la fila con mayor número de casos.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
casos_sex <- agrupar_sex(
  data_event = data_limpia,
  porcentaje = TRUE
)
obtener_fila_mas_casos(
  data_event = casos_sex,
  nomb_col = "casos",
  porcentaje = TRUE
)
```

| | |
|--------------------|--|
| obtener_info_depts | <i>Obtener información geográfica de los datos de la enfermedad o evento</i> |
|--------------------|--|

Description

Función que obtiene la información geográfica de los datos de la enfermedad o evento.

Usage

```
obtener_info_depts(dpto = NULL, mpio = NULL)
```

Arguments

| | |
|------|--|
| dpto | Un 'character' (cadena de caracteres) o 'numeric' (numérico) que contiene el nombre o código del departamento; su valor por defecto es 'NULL'. |
| mpio | Un 'character' (cadena de caracteres) o 'numeric' (numérico) que contiene el nombre o código del municipio; su valor por defecto es 'NULL'. |

Value

Un 'data.frame' con la información geográfica de los datos de la enfermedad o evento.

Examples

```

obtener_info_depts(dpto = "ANTIOQUIA")
obtener_info_depts(dpto = "ANTIOQUIA", mpio = "MEDELLIN")
obtener_info_depts(dpto = "05")
obtener_info_depts(dpto = "05", mpio = "05001")
obtener_info_depts(dpto = 05, mpio = 05001)
obtener_info_depts(dpto = 05, mpio = 001)
obtener_info_depts(dpto = "bogota dc", mpio = "bogota dc")

```

```
obtener_meses_mas_casos
```

Obtener los meses con mayor número de casos

Description

Función que obtiene los meses con el mayor número de casos

Usage

```

obtener_meses_mas_casos(
  data_event,
  col_fechas,
  col_casos = "casos",
  top = 1,
  concat_vals = TRUE
)

```

Arguments

| | |
|-------------|--|
| data_event | Un 'data.frame' con los datos de la enfermedad o evento. |
| col_fechas | Un 'array' (arreglo) de 'character' (cadena de caracteres) con los nombres de las columnas que contienen las fechas en los datos de la enfermedad o evento. |
| col_casos | Un 'character' (cadena de caracteres) con el nombre de la columna de los datos de la enfermedad o evento que contiene el número de casos; su valor por defecto es "casos". |
| top | Un 'numeric' (numérico) que contiene la cantidad máxima de meses a retornar; su valor por defecto es '3'. |
| concat_vals | Un 'logical' ('TRUE' o 'FALSE') que indica si se requiere concatenar los meses como una cadena; su valor por defecto es 'TRUE'. |

Value

Un 'data.frame' que contiene los meses con mayor número de casos.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
casos_inisintomas <- agrupar_fecha_inisintomas(data_limpia)
obtener_meses_mas_casos(
  data_event = casos_inisintomas,
  col_fechas = "ini_sin",
  col_casos = "casos",
  top = 3,
  concat_vals = TRUE
)
```

obtener_nombre_dpto *Obtener el nombre de un departamento de Colombia*

Description

Función que obtiene el nombre de un departamento de Colombia a partir de su código geográfico.

Usage

```
obtener_nombre_dpto(data_geo, cod_dpto)
```

Arguments

| | |
|----------|---|
| data_geo | Un ‘data.frame’ que contiene los códigos geográficos (departamentos y municipios de Colombia). |
| cod_dpto | Un ‘numeric’ (numérico) o ‘character’ (cadena de caracteres) que contiene el código del departamento. |

Value

Un ‘character’ (cadena de caracteres) con el nombre del departamento.

Examples

```
data_geo <- import_geo_cods()
obtener_nombre_dpto(data_geo,
  cod_dpto = "05"
)
obtener_nombre_dpto(data_geo,
  cod_dpto = 05
)
obtener_nombre_dpto(data_geo,
  cod_dpto = 5
)
```

obtener_nombre_mpio *Obtener el nombre de un municipio de Colombia*

Description

Función que obtiene el nombre de un municipio de Colombia a partir de su código geográfico.

Usage

```
obtener_nombre_mpio(data_geo, cod_dpto, cod_mpio)
```

Arguments

| | |
|----------|---|
| data_geo | Un 'data.frame' que contiene los códigos geográficos (departamentos y municipios de Colombia). |
| cod_dpto | Un 'numeric' (numérico) o 'character' (cadena de caracteres) que contiene el código del departamento. |
| cod_mpio | Un 'numeric' (numérico) o 'character' (cadena de caracteres) que contiene el código del municipio. |

Value

Un 'character' (cadena de caracteres) con el nombre del municipio.

Examples

```
data_geo <- import_geo_cods()
obtener_nombre_mpio(data_geo,
  cod_dpto = "05",
  cod_mpio = "001"
)
obtener_nombre_mpio(data_geo,
  cod_dpto = 05,
  cod_mpio = 001
)
obtener_nombre_mpio(data_geo,
  cod_dpto = 5,
  cod_mpio = 1
)
```

obtener_text_sex *Obtener el párrafo de la distribución de casos por sexo*

Description

Función que obtiene el párrafo descriptivo de la sección de distribución de casos por sexo de la plantilla del reporte.

Usage

```
obtener_text_sex(data_agrupada, year, figura)
```

Arguments

`data_agrupada` Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados por sexo.

`year` Un 'numeric' (numérico) con el año de los datos agrupados por sexo.

`figura` Un 'numeric' (numérico) con el número de la figura de la distribución de casos por sexo.

Value

Un 'character' (cadena de caracteres) con el párrafo descriptivo de la distribución de casos por sexo.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_sex(
  data_event = data_limpia,
  porcentaje = TRUE
)
obtener_text_sex(data_agrupada, year = 2020, figura = 3)
```

obtener_tip_ocurren_geo *Obtener columnas de ocurrencia geográfica de los datos de la enfermedad o evento*

Description

Función que obtiene las columnas de ocurrencia geográfica de los datos de la enfermedad o evento.

Usage

```
obtener_tip_ocurren_geo(cod_event = NULL, nombre_event = NULL)
```

Arguments

| | |
|--------------|--|
| cod_event | Un 'numeric' (numérico) o 'character' (cadena de caracteres) que contiene el código de la enfermedad o evento. |
| nombre_event | Un 'character' (cadena de caracteres) con el nombre de la enfermedad o evento. |

Value

Un 'data.frame' con las columnas de ocurrencia geográfica de los datos de la enfermedad o evento.

Examples

```
obtener_tip_ocurren_geo(cod_event = 210)
```

obtener_val_config *Obtener valor del archivo de configuración*

Description

Función que obtiene el valor de una llave del archivo de configuración.

Usage

```
obtener_val_config(llave)
```

Arguments

| | |
|-------|--|
| llave | Un 'character' (cadena de caracteres) con el nombre de la llave que se encuentra en el archivo de configuración del paquete. |
|-------|--|

Value

Un 'character' (cadena de caracteres) con el valor de la llave del archivo de configuración del paquete.

Examples

```
obtener_val_config("request_timeout")
```

plot_area_geo *Generar gráfico de distribución de casos por área geográfica*

Description

Función que genera el gráfico de casos por área geográfica.

Usage

```
plot_area_geo(data_agrupada, col_area = "area", fuente_data = NULL)
```

Arguments

`data_agrupada` Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados.

`col_area` Un 'character' (cadena de caracteres) con el nombre de la columna con el área geográfica en los datos agrupados de la enfermedad o evento; su valor por defecto es "area".

`fuente_data` Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es 'NULL'.

Value

Un 'plot' o gráfico de distribución de casos por área geográfica.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_area_geo(data_event = data_limpia)
plot_area_geo(data_agrupada,
  col_area = "area"
)
```

plot_dptos *Generar gráfico de distribución de casos por departamentos*

Description

Función que genera el gráfico de distribución de casos por departamentos.

Usage

```
plot_dptos(data_agrupada, col_dptos = NULL, fuente_data = NULL)
```

Arguments

| | |
|---------------|--|
| data_agrupada | Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados por departamentos. |
| col_dptos | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene los departamentos en los datos agrupados de la enfermedad o evento; su valor por defecto es 'NULL'. |
| fuentes_data | Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es 'NULL'. |

Value

Un 'plot' o gráfico de distribución de casos por departamentos.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_limpia <- estandarizar_geo_cods(data_limpia)
data_agrupada <- agrupar_dpto(data_event = data_limpia)
plot_dptos(data_agrupada,
  col_dptos = "departamento_ocurrencia"
)
```

plot_edad

Generar gráfico de distribución de casos por edad

Description

Función que genera un gráfico de distribución de casos por edad.

Usage

```
plot_edad(data_agrupada, col_edad = "edad", fuentes_data = NULL)
```

Arguments

| | |
|---------------|---|
| data_agrupada | Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados. |
| col_edad | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene las edades en los datos agrupados de la enfermedad o evento; su valor por defecto es "edad". |
| fuentes_data | Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es 'NULL'. |

Value

Un 'plot' o gráfico de distribución de casos por edad.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_edad(data_event = data_limpia)
plot_edad(
  data_agrupada = data_agrupada,
  col_edad = "edad"
)
```

plot_edad_sex

Generar gráfico de distribución de casos por edad y sexo

Description

Función que genera un gráfico de distribución de casos por edad y sexo.

Usage

```
plot_edad_sex(
  data_agrupada,
  col_edad = "edad",
  col_sex = "sexo",
  fuente_data = NULL
)
```

Arguments

`data_agrupada` Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados.

`col_edad` Un 'character' (cadena de caracteres) con el nombre de la columna que contiene las edades en los datos agrupados de la enfermedad o evento; su valor por defecto es "edad".

`col_sex` Un 'character' (cadena de caracteres) con el nombre de la columna que contiene el sexo en los datos agrupados de la enfermedad o evento; su valor por defecto es "sexo".

`fuente_data` Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es 'NULL'.

Value

Un 'plot' o gráfico de distribución de casos por edad y sexo.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_edad_sex(data_event = data_limpia)
plot_edad_sex(
  data_agrupada = data_agrupada,
  col_edad = "edad",
  col_sex = "sexo"
)
```

plot_fecha_inisintomas

Generar gráfico de distribución de casos por fecha de inicio de síntomas

Description

Función que genera un gráfico de distribución de casos por fecha de inicio de síntomas.

Usage

```
plot_fecha_inisintomas(
  data_agrupada,
  col_fecha = "ini_sin",
  uni_marca = "semanaepi",
  tipo = "barras",
  fuente_data = NULL
)
```

Arguments

| | |
|---------------|--|
| data_agrupada | Un ‘data.frame’ que contiene los datos de la enfermedad o evento agrupados. |
| col_fecha | Un ‘character’ (cadena de caracteres) que contiene el nombre de la columna con las fechas de notificación en los datos de la enfermedad o evento agrupados; su valor por defecto es “ini_sin”. |
| uni_marca | Un ‘character’ (cadena de caracteres) que contiene la unidad de las marcas del gráfico (“dia”, “semanaepi” y “mes”); su valor por defecto es “semanaepi”. |
| tipo | Un ‘character’ (cadena de caracteres) que contiene el tipo de gráfico (“barras” o “tendencia”); su valor por defecto es “barras”. |
| fuente_data | Un ‘character’ (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es ‘NULL’. |

Value

Un ‘plot’ o gráfico de la distribución de casos por fecha de inicio de síntomas.

Examples

```

data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_fecha_inisintomas(
  data_event = data_limpia
)
plot_fecha_inisintomas(
  data_agrupada = data_agrupada,
  col_fecha = "ini_sin",
  uni_marca = "semanaepi"
)

```

plot_map

Generar mapa

Description

Función que genera el mapa por departamentos o municipios con el número de casos o la incidencia de una enfermedad o evento.

Usage

```

plot_map(
  data_agrupada,
  col_distribucion = "incidencia",
  col_codigos = NULL,
  fuente_data = NULL,
  dpto = NULL,
  mpio = NULL,
  ruta_dir = NULL,
  cache = FALSE
)

```

Arguments

- `data_agrupada` Un 'data.frame' que contiene los datos de la enfermedad agrupados por departamento y número de casos.
- `col_distribucion` Un 'character' (cadena de caracteres) que contiene el nombre de la columna que tiene los valores de la distribución, ya sea por número de casos o incidencia; su valor por defecto es "incidencia".
- `col_codigos` Un 'character' (cadena de caracteres) que contiene el nombre de la columna con los códigos de los departamentos o municipios, los cuales se utilizan para obtener los polígonos de las áreas geográficas del archivo geoespacial o Shapefile; su valor por defecto 'NULL'.

| | |
|-------------|--|
| fuelle_data | Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos de la enfermedad o evento; su valor por defecto 'NULL'. |
| dpto | Un 'character' (cadena de caracteres) que contiene el nombre del departamento; su valor por defecto 'NULL'. |
| mpio | Un 'character' (cadena de caracteres) que contiene el nombre del municipio; su valor por defecto 'NULL'. |
| ruta_dir | Un 'character' (cadena de caracteres) que contiene la ruta del directorio donde se almacenará el Shapefile del mapa de Colombia. Su valor por defecto es 'NULL'. |
| cache | Un 'logical' ('TRUE' o 'FALSE') que indica si el Shapefile del mapa de Colombia debe ser almacenado en caché. Su valor por defecto es 'FALSE'. |

Value

Un 'plot' o mapa por departamentos o municipios con el número de casos o incidencia de un evento o enfermedad específica.

Examples

```

data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_estandar <- estandarizar_geo_cods(data_limpia)
# Mapa por departamentos
geo_ocurrencia <- obtener_tip_ocurren_geo(nombre_event = "dengue")
data_espacial <- agrupar_dpto(data_event = data_estandar,
                             geo_ocurrencia[1:4])

if (interactive()) {
  plot_map(
    data_agrupada = data_espacial,
    col_distribucion = "casos",
    cache = TRUE
  )
}
# Mapa por municipios de un departamento especifico
data_filtrada_dpto <- geo_filtro(
  data_event = data_estandar,
  dpto = "Cundinamarca"
)
data_espacial_dpto <- agrupar_mpio(data_event = data_filtrada_dpto)
plot_map(
  data_agrupada = data_espacial_dpto,
  col_codigos = "cod_mun_o",
  col_distribucion = "casos",
  ruta_dir = tempdir()
)
# Mapa por municipio especifico
data_filtrada_mpio <- geo_filtro(
  data_event = data_estandar,
  dpto = "Antioquia",
  mpio = "Medellin"
)

```

```

data_espacial_mpio <- agrupar_mpio(data_event = data_filtrada_mpio)
if (interactive()) {
  plot_map(
    data_agrupada = data_espacial_mpio,
    col_codigos = "cod_mun_o",
    col_distribucion = "casos",
    dpto = "Antioquia",
    mpio = "Medellin",
    cache = TRUE
  )
}
# Mapa con la incidencia por municipios de un departamento específico
incidencia_dpto <-
  calcular_incidencia_geo(data_agrupada = data_espacial_dpto,
    ruta_dir = tempdir())
plot_map(
  data_agrupada = incidencia_dpto$data_incidencia,
  col_codigos = "cod_mun_o",
  col_distribucion = "incidencia",
  ruta_dir = tempdir()
)

```

plot_mpios

Generar gráfico de distribución de casos por municipios

Description

Función que genera un gráfico de distribución de casos por municipios.

Usage

```
plot_mpios(data_agrupada, col_mpios = NULL, fuente_data = NULL)
```

Arguments

| | |
|---------------|---|
| data_agrupada | Un ‘data.frame’ que contiene los datos de la enfermedad o evento agrupados por municipios. |
| col_mpios | Un ‘character’ (cadena de caracteres) con el nombre de la columna que contiene los municipios en los datos agrupados de la enfermedad o evento; su valor por defecto es ‘NULL’. |
| fuente_data | Un ‘character’ (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es ‘NULL’. |

Value

Un ‘plot’ o gráfico de distribución de casos por municipios.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_limpia <- estandarizar_geo_cods(data_limpia)
data_agrupada <- agrupar_mpio(
  data_event = data_limpia,
  dpto = "Antioquia"
)
plot_mpios(data_agrupada,
  col_mpios = "municipio_ocurrencia"
)
```

plot_per_etn

Generar gráfico de distribución de casos por pertenencia étnica

Description

Función que genera el gráfico de la distribución de casos por pertenencia étnica.

Usage

```
plot_per_etn(
  data_agrupada,
  col_etn = "per_etn",
  porcentaje = TRUE,
  fuente_data = NULL
)
```

Arguments

| | |
|---------------|---|
| data_agrupada | Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados por pertenencia étnica. |
| col_etn | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene la pertenencia étnica en los datos agrupados de la enfermedad o evento; su valor por defecto es "per_etn". |
| porcentaje | Un 'logical' ('TRUE' o 'FALSE') que indica si los datos tienen porcentajes; su valor por defecto es 'TRUE'. |
| fuente_data | Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es 'NULL'. |

Value

Un 'plot' o gráfico de la distribución de casos por pertenencia étnica.

Examples

```

data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_per_etn(data_event = data_limpia)
plot_per_etn(data_agrupada,
  col_etn = "per_etn"
)

```

plot_sex

Generar gráfico de distribución de casos por sexo

Description

Función que genera un gráfico de distribución de casos por sexo.

Usage

```

plot_sex(
  data_agrupada,
  col_sex = "sexo",
  col_distribucion = "casos",
  porcentaje = TRUE,
  fuente_data = NULL
)

```

Arguments

| | |
|------------------|--|
| data_agrupada | Un ‘data.frame’ que contiene los datos de la enfermedad o evento agrupados. |
| col_sex | Un ‘character’ (cadena de caracteres) con el nombre de la columna que contiene el sexo en los datos agrupados de la enfermedad o evento; su valor por defecto es “sexo”. |
| col_distribucion | Un ‘character’ (cadena de caracteres) que contiene el nombre de la columna que tiene los valores de la distribución, por número de casos o incidencia; su valor por defecto es “incidencia”. |
| porcentaje | Un ‘logical’ (‘TRUE’ o ‘FALSE’) que indica si los datos tienen porcentajes; su valor por defecto es ‘TRUE’. |
| fuente_data | Un ‘character’ (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es ‘NULL’. |

Value

Un ‘plot’ o gráfico de distribución de casos por sexo.

Examples

```

data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_sex(
  data_event = data_limpia,
  porcentaje = TRUE
)
plot_sex(
  data_agrupada = data_agrupada,
  col_sex = "sexo",
  porcentaje = TRUE
)

```

| | |
|--------------------|--|
| plot_sex_semanaepi | <i>Generar gráfico de distribución de casos por sexo y semana epidemiológica</i> |
|--------------------|--|

Description

Función que genera un gráfico de distribución de casos por sexo y semana epidemiológica.

Usage

```

plot_sex_semanaepi(
  data_agrupada,
  col_sex = "sexo",
  col_semanaepi = "semana",
  fuente_data = NULL
)

```

Arguments

| | |
|---------------|--|
| data_agrupada | Un ‘data.frame’ que contiene los datos de la enfermedad o evento agrupados. |
| col_sex | Un ‘character’ (cadena de caracteres) con el nombre de la columna que contiene el sexo en los datos agrupados de la enfermedad o evento; su valor por defecto es “sexo”. |
| col_semanaepi | Un ‘character’ (cadena de caracteres) con el nombre de la columna que contiene las semanas epidemiológicas en los datos agrupados de la enfermedad o evento; su valor por defecto es “semana”. |
| fuente_data | Un ‘character’ (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es ‘NULL’. |

Value

Un ‘plot’ o gráfico de distribución de casos por sexo y semana epidemiológica.

Examples

```

data(dengue2020)
data_limpiar <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_sex_semanaepi(data_event = data_limpiar)
plot_sex_semanaepi(
  data_agrupada = data_agrupada,
  col_sex = "sexo",
  col_semanaepi = "semana"
)

```

plot_tabla_incidencia_geo

Generar tabla con la incidencia

Description

Función que genera la tabla con la incidencia según la distribución geográfica.

Usage

```
plot_tabla_incidencia_geo(data_agrupada, col_geo = NULL)
```

Arguments

| | |
|---------------|--|
| data_agrupada | Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados por departamento o municipio. |
| col_geo | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene los nombres de los departamentos o municipios en los datos agrupados de la enfermedad o evento; su valor por defecto es 'NULL'. |

Value

Una 'kable' (tabla gráfica) con la incidencia según distribución geográfica.

Examples

```

data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
data_agrupada <- agrupar_mpio(data_limpiar, dpto = "Antioquia")
incidencia_mpios <- calcular_incidencia_geo(
  data_agrupada =
    data_agrupada,
  ruta_dir = tempdir()
)
plot_tabla_incidencia_geo(
  data_agrupada = incidencia_mpios$data_incidencia,
  col_geo = "municipio_ocurrencia"
)

```

`plot_tabla_incidencia_sex`*Generar tabla con la incidencia por sexo*

Description

Función que genera la tabla con la incidencia por sexo.

Usage

```
plot_tabla_incidencia_sex(data_agrupada, col_sex = "sexo")
```

Arguments

`data_agrupada` Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados por departamento o municipio.

`col_sex` Un 'character' (cadena de caracteres) con el nombre de la columna que contiene el sexo en los datos agrupados de la enfermedad o evento; su valor por defecto es "sexo".

Value

Una 'kable' (tabla gráfica) con la incidencia por sexo.

Examples

```
data(dengue2020)
data_limpiar <- limpiar_data_sivigila(data_event = dengue2020)
data_agrupada_sex <- agrupar_sex(data_limpiar)
incidencia_mpios <-
  calcular_incidencia_sex(
    data_agrupada = data_agrupada_sex,
    dpto = "Antioquia",
    ruta_dir = tempdir()
  )
plot_tabla_incidencia_sex(
  data_agrupada = incidencia_mpios$data_incidencia,
  col_sex = "sexo"
)
```

plot_tabla_tipos_event

Generar tabla con la distribución de casos por tipo de enfermedad o evento

Description

Función que genera la tabla con la distribución de casos por tipo de enfermedad o evento.

Usage

```
plot_tabla_tipos_event(data_agrupada, col_event = "nombre_evento")
```

Arguments

`data_agrupada` Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados por tipo.

`col_event` Un 'character' (cadena de caracteres) con el nombre de la columna que contiene el tipo de evento en los datos agrupados de la enfermedad o evento; su valor por defecto es "nombre_evento".

Value

Una 'kable' (tabla gráfica) con la distribución de casos por tipo de enfermedad o evento.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_eventos(
  data_event = data_limpia,
  col_event = "cod_eve"
)
plot_tabla_tipos_event(data_agrupada,
  col_event = "nombre_evento"
)
```

plot_tipo_caso

Generar gráfico de distribución de casos por la clasificación inicial del caso

Description

Función que genera un gráfico de distribución de casos según su clasificación inicial.

Usage

```
plot_tipo_caso(data_agrupada, col_tipo = "tip_cas", fuente_data = NULL)
```

Arguments

`data_agrupada` Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados según la clasificación inicial de los casos.

`col_tipo` Un 'character' (cadena de caracteres) con el nombre de la columna que contiene la clasificación inicial de los casos en los datos agrupados de la enfermedad o evento; su valor por defecto es "tip_cas".

`fuentes_data` Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es 'NULL'.

Value

Un 'plot' o gráfico de distribución de casos según su clasificación inicial.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_tipo_caso(data_event = data_limpia)
plot_tipo_caso(data_agrupada,
  col_tipo = "tip_cas"
)
```

`plot_tipo_caso_years` *Generar gráfico de distribución de casos por la clasificación inicial del caso y los años seleccionados*

Description

Función que genera un gráfico de distribución de casos según su clasificación inicial y los años seleccionados.

Usage

```
plot_tipo_caso_years(
  data_agrupada,
  col_tipo = "tip_cas",
  col_year = "ano",
  fuentes_data = NULL
)
```

Arguments

| | |
|---------------|---|
| data_agrupada | Un 'data.frame' que contiene los datos de la enfermedad o evento, agrupados por la clasificación inicial y los años seleccionados. |
| col_tipo | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene la clasificación inicial del caso en los datos agrupados de la enfermedad o evento; su valor por defecto es "tip_cas". |
| col_year | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene el año en los datos agrupados de la enfermedad o evento; su valor por defecto es "ano". |
| fuentes_data | Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es 'NULL'. |

Value

Un 'plot' o gráfico de distribución de casos según su clasificación inicial y los años seleccionados.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_tipo_caso(
  data_event = data_limpia,
  cols_tipo = c(
    "tip_cas",
    "ano"
  )
)
plot_tipo_caso_years(data_agrupada,
  col_tipo = "tip_cas",
  col_year = "ano"
)
```

| | |
|-------------------|---|
| plot_top_area_geo | <i>Generar gráfico de distribución de casos por área geográfica a nivel departamental o municipal</i> |
|-------------------|---|

Description

Función que genera el gráfico de casos por área geográfica a nivel departamental o municipal.

Usage

```
plot_top_area_geo(data_agrupada, col_area = "area", fuentes_data = NULL)
```

Arguments

| | |
|---------------|---|
| data_agrupada | Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados. |
| col_area | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene el área geográfica en los datos agrupados de la enfermedad o evento; su valor por defecto es "area". |
| fuentes_data | Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es 'NULL'. |

Value

Un 'plot' o gráfico de distribución de casos por área geográfica.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_top_area_geo(
  data_event = data_limpia,
  dpto = "Antioquia"
)
plot_top_area_geo(data_agrupada,
  col_area = "area"
)
```

plot_years

Generar gráfico de distribución de casos por año

Description

Función que genera un gráfico de distribución de casos por año.

Usage

```
plot_years(data_agrupada, col_year = "ano", fuentes_data = NULL)
```

Arguments

| | |
|---------------|--|
| data_agrupada | Un 'data.frame' que contiene los datos de la enfermedad o evento agrupados por año. |
| col_year | Un 'character' (cadena de caracteres) con el nombre de la columna que contiene los años en los datos agrupados de la enfermedad o evento; su valor por defecto es "ano". |
| fuentes_data | Un 'character' (cadena de caracteres) que contiene la leyenda o fuente de información de los datos; su valor por defecto es 'NULL'. |

Value

Un 'plot' o gráfico de distribución de casos por año.

Examples

```
data(dengue2020)
data_limpia <- limpiar_data_sivigila(dengue2020)
data_agrupada <- agrupar_years(data_event = data_limpia)
plot_years(data_agrupada,
  col_year = "ano"
)
if (interactive()) {
  data_years <- import_data_event(
    nombre_event = "CHAGAS",
    years = c(2019, 2020)
  )
  data_limpia <- limpiar_data_sivigila(data_years)
  data_agrupada <- agrupar_years(data_event = data_limpia)
  plot_years(data_agrupada, col_year = "ano")
}
```

Index

* datasets

dengue2020, [23](#)
divipoladata, [26](#)

agrupar_area_geo, [4](#)
agrupar_cols_casos, [4](#)
agrupar_dpto, [5](#)
agrupar_edad, [6](#)
agrupar_edad_sex, [7](#)
agrupar_eventos, [8](#)
agrupar_fecha_inisintomas, [9](#)
agrupar_mpio, [9](#)
agrupar_per_etn, [11](#)
agrupar_rango_edad, [11](#)
agrupar_semanaepi, [13](#)
agrupar_sex, [13](#)
agrupar_sex_semanaepi, [14](#)
agrupar_tipo_caso, [15](#)
agrupar_top_area_geo, [16](#)
agrupar_years, [17](#)

calcular_incidencia, [17](#)
calcular_incidencia_geo, [19](#)
calcular_incidencia_sex, [21](#)
convert_edad, [22](#)

dengue2020, [23](#)
divipoladata, [26](#)

estandarizar_geo_cods, [26](#)

geo_filtro, [27](#)

import_data_event, [28](#)
import_geo_cods, [29](#)
import_pob_incidencia, [29](#)
import_pob_proyecciones, [30](#)
import_pob_riesgo, [31](#)

limpiar_data_sivigila, [32](#)
limpiar_edad_event, [33](#)

limpiar_encabezado, [34](#)
limpiar_fecha_event, [34](#)
limpiar_val_atipic, [35](#)
list_events, [36](#)

obtener_cond_inciden_event, [36](#)
obtener_dptos, [37](#)
obtener_fila_mas_casos, [37](#)
obtener_info_depts, [38](#)
obtener_meses_mas_casos, [39](#)
obtener_nombre_dpto, [40](#)
obtener_nombre_mpio, [41](#)
obtener_text_sex, [42](#)
obtener_tip_ocurren_geo, [42](#)
obtener_val_config, [43](#)

plot_area_geo, [44](#)
plot_dptos, [44](#)
plot_edad, [45](#)
plot_edad_sex, [46](#)
plot_fecha_inisintomas, [47](#)
plot_map, [48](#)
plot_mpios, [50](#)
plot_per_etn, [51](#)
plot_sex, [52](#)
plot_sex_semanaepi, [53](#)
plot_tabla_incidencia_geo, [54](#)
plot_tabla_incidencia_sex, [55](#)
plot_tabla_tipos_event, [56](#)
plot_tipo_caso, [56](#)
plot_tipo_caso_years, [57](#)
plot_top_area_geo, [58](#)
plot_years, [59](#)