

# Package: istat (via r-universe)

January 9, 2025

**Title** Download and Manipulate Data from Istat

**Version** 1.0

**Maintainer** Elena Gradi <elenaagradi@gmail.com>

**Description** Download data from Istat (Italian Institute of Statistics) database, both old and new provider (respectively, <<http://dati.istat.it/>> and <<https://esploradati.istat.it/databrowser/>>). Additional functions for manipulating data are provided. Moreover, a 'shiny' application called 'shinyIstat' can be used to search, download and filter datasets in an easier way.

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** datamods, DT, dplyr, ggplot2, htmltools, httr, magrittr, openxlsx, reactable, readxl, rsdmx, shiny, shinyBS, shinydashboard, shinyhelper, shinyjs, shinyWidgets, writexl

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Elena Gradi [aut, cre], Alissa Lelli [aut], Daniela Ichim [aut]

**Repository** CRAN

**Date/Publication** 2024-09-10 10:00:06 UTC

**Config/pak/sysreqs** make libicu-dev libxml2-dev libssl-dev libx11-dev  
zlib1g-dev

## Contents

filter_istat . . . . .	2
filter_istat_interactive . . . . .	3
get_istatdata . . . . .	3
get_i_stat . . . . .	4

list_istatdata . . . . .	5
list_i_stat . . . . .	6
plot_interactive . . . . .	7
search_istatdata . . . . .	7
search_i_stat . . . . .	8
shinyIstat . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

filter_istat	<i>Filter data sets</i>
--------------	-------------------------

---

### Description

Filter data set by column(s). Takes as input a data.frame (not only ISTAT ones) and allows you to select for which column(s) value(s) to filter the dataset. Alternatively, use filter\_istat\_interactive for an interactive version.

### Usage

```
filter_istat(dataset, columns, datatype, lang = "ita")
```

### Arguments

dataset	as data.frame
columns	data set column(s) for which you want to filter the data. If you want to filter for more than one column, write c("column1", "column2", ...) as argument.
datatype	column(s) value(s) for which you want to filter the data. Write as many datatype as many columns that you selected in "columns" (as in examples).
lang	language parameter for labels ("ita" for Italian, "eng" for English)

### Value

It returns the filtered data set.

### Examples

```
filter_istat(iris, columns = "Species", datatype = "setosa") #Here,
#the function filters the data set 'iris' for the value 'setosa' of the column 'Species'.
filter_istat(iris, columns = c("Species", "Petal.Width"),
datatype = list(c("virginica", "setosa"), c("0.1", "1.9")))
# Here, the function filters the data set 'iris' for the values 'virginica'
#and 'setosa' of the column 'Species' and for the values '0.1' and '1.9' of
#the column 'Petal.Width'.
```

---

`filter_istat_interactive`*Filter data set interactively*

---

**Description**

An interactive and more intuitive version of `filter_istat` function. It filters data set by column(s). Takes as input a `data.frame` (not only ISTAT ones) and allows you to select for which column(s) value(s) to filter the data set interactively.

**Usage**

```
filter_istat_interactive(dataset, lang = "ita")
```

**Arguments**

`dataset` as `data.frame`

`lang` language parameter for labels ("ita" for Italian, "eng" for English)

**Value**

It returns the filtered data set.

**Note**

In this first version, language parameter works only with data sets downloaded with `get_i_stat` (provider I.Stat).

**Examples**

```
filter_istat_interactive(iris)
```

---

`get_istatdata`*Download data set by agencyId, id and version (source: IstatData)*

---

**Description**

Download data sets from `IstatData` (new ISTAT provider). Alternatively, use `get_i_stat` to download data sets from `I.Stat` (old ISTAT provider). Note that in this first version of the package, only `get_i_stat` allows you to download variable labels, while `get_istatdata` has not this functionality yet. The parameters "agencyId", "dataset\_id" and "version" to download the data sets can be found using `list_istatdata` function or `search_istatdata` function.

**Usage**

```
get_istatdata(agencyId,
              dataset_id,
              version,
              start=NULL,
              end=NULL,
              recent=FALSE,
              csv=FALSE,
              xlsx=FALSE)
```

**Arguments**

agencyId	data set agencyId
dataset_id	data set id
version	data set version
start	Time value for the start (NULL by default)
end	Time value for the end (NULL bu default)
recent	False by default, if TRUE, the function retrieves data from last 10 years
csv	False by default, if TRUE, the function saves the data set to directory as .csv
xlsx	False by default, if TRUE, the function saves the data set to directory as .xlsx

**Value**

It returns data set as data.frame. It can be saved to environment or as .csv/.xlsx.

**Note**

Downloading may take some time. Future versions will speed up the process.

**Examples**

```
get_istatdata("IT1", "12_60_DF_DCCV_CONSACQUA_2" , "1.0", recent = TRUE)
get_istatdata("IT1", "12_60_DF_DCCV_CONSACQUA_2" , "1.0", start = 2015, end = 2018)
```

---

get\_i\_stat

*Download data sets by id (source: I.Stat)*

---

**Description**

Download data sets from I.Stat (old ISTAT provider). Alternatively, use get\_istatdata to download data sets from IstatData (new ISTAT provider). Note that in this first version of the package, only get\_i\_stat allows you to download variable labels, while get\_istatdata has not this functionality yet. The parameters "id\_dataset" to download data sets can be found using list\_i\_stat function or search\_i\_stat function.

**Usage**

```
get_i_stat(id_dataset,  
           start_period=NULL,  
           end_period=NULL,  
           recent=FALSE,  
           csv=FALSE,  
           xlsx=FALSE,  
           lang="both")
```

**Arguments**

id_dataset	data set id
start_period	Time value for the start (NULL by default)
end_period	Time value for the end (NULL by default)
recent	False by default, if TRUE, the function retrieves data from last 10 years
csv	False by default, if TRUE, the function saves the data set to directory as .csv
xlsx	False by default, if TRUE, the function saves the data set to directory as .xlsx
lang	Language parameter for labels ("ita" for Italian, "eng" for English)

**Value**

It returns the data set as data.frame. It can be saved to environment or as .csv/.xlsx.

**Note**

Downloading may take some time. Future versions will speed up the process.

**Examples**

```
get_i_stat("12_60")  
get_i_stat("12_60", start_period=2015, end_period=2018)  
get_i_stat("12_60", start_period=2015, end_period=2015)  
get_i_stat("12_60", recent=TRUE, lang="eng")
```

---

list_istatdata	<i>Obtain the complete list of available data sets (source: IstatData)</i>
----------------	--

---

**Description**

Obtain the complete list of available data sets (source: IstatData)

**Usage**

```
list_istatdata(lang = "ita")
```

**Arguments**

lang                    language parameter for labels ("ita" for Italian, "eng" for English)

**Value**

It returns the complete list of available data sets from IstatData with their agencyId, id, version and name.

**Examples**

```
list_istatdata()
```

---

list\_i\_stat                    *Obtain the complete list of available data sets (source: I.Stat)*

---

**Description**

Obtain the complete list of available data sets (source: I.Stat)

**Usage**

```
list_i_stat(lang = "ita")
```

**Arguments**

lang                    language parameter for labels ("ita" for Italian, "eng" for English)

**Value**

It returns the complete list of available data sets from I.Stat with their id and name.

**Examples**

```
list_i_stat()
```

---

plot_interactive	<i>Plot dataset interactively</i>
------------------	-----------------------------------

---

**Description**

Build different types of exploratory graphs (scatter plot, bar plot, pie chart). You can interactively choose the plot that you want create. Once you have chosen the plot, you can interactively choose the variables from the data set for which you want to build the plot.

**Usage**

```
plot_interactive(df)
```

**Arguments**

df	data set as data.frame
----	------------------------

**Value**

It returns the chosen plot.

**Note**

plot\_interactive allows you to have an idea about the general trend of your data, and it's intended to be used with exploratory purpose.

**Examples**

```
if(interactive()) {  
  plot_interactive(iris)  
}
```

---

search_istatdata	<i>Search data sets by keywords (source: IstatData).</i>
------------------	--

---

**Description**

Search IstatData data sets by keywords. To download data sets use "get\_istatdata" function and insert agencyId, id, version.

**Usage**

```
search_istatdata(keywords,  
                 lang = "ita")
```

**Arguments**

keywords      keyword(s) to search data sets  
 lang          language parameter for labels ("ita" for Italian, "eng" for English)

**Value**

It returns a list of data sets containing the keyword(s) with their agencyId, id, version and name.

**Note**

Searching may take some time. Future versions will speed up the process.

**Examples**

```
search_istatdata(c( "incidenti", "stradali"))
search_istatdata("population", lang="eng")
```

---

search\_i\_stat      *Search data sets by keywords (source: I.Stat)*

---

**Description**

Search I.Stat data sets by keywords. To download data sets, use "get\_i\_stat" function and insert data set id.

**Usage**

```
search_i_stat(keywords,
              lang = "ita")
```

**Arguments**

keywords      keyword(s) to search data sets  
 lang          language parameter for labels ("ita" for Italian, "eng" for English)

**Value**

It returns a list of data sets containing the keyword(s) with their id and name.

**Note**

Searching may take some time. Future versions will speed up the process.

**Examples**

```
search_i_stat(c( "incidenti", "stradali"))
search_i_stat("population", lang="eng")
```



---

`shinyIstat`*shinyIstat*

---

**Description**

A graphic interface that makes searching, downloading and filtering data sets from Istat easier. Call `shinyIstat()` to get started. This shinyApp was built using the same functions of `istat` package but they have been adapted for the shiny. You will find additional information and help inside the app.

**Usage**

```
shinyIstat()
```

**Value**

It opens the app.

**Note**

Calling a ShinyApp equals to calling a R function. For this reason, once called `shinyIstat()`, R will be busy processing it until the app will be closed. As a consequence, all other R functionalities can be processed only when the app is closed.

**Examples**

```
if(interactive()) {  
  shinyIstat()  
}
```

# Index

`filter_istat`, [2](#)  
`filter_istat_interactive`, [3](#)  
  
`get_i_stat`, [4](#)  
`get_istatdata`, [3](#)  
  
`list_i_stat`, [6](#)  
`list_istatdata`, [5](#)  
  
`plot_interactive`, [7](#)  
  
`search_i_stat`, [8](#)  
`search_istatdata`, [7](#)  
`shinyIstat`, [9](#)