Package 'tidytuesdayR'

September 11, 2026

Type Package

Title Access the Weekly 'TidyTuesday' Project Dataset

Version 1.3.3

Description 'TidyTuesday' is a project by the 'Data Science Learning Community' in which they post a weekly dataset in a public data repository (<https://github.com/rfordatascience/tidytuesday>) for people to analyze and visualize. This package provides the tools to easily download this data and the description of the source.

License MIT + file LICENSE

URL https://dslc-io.github.io/tidytuesdayR/, https://github.com/dslc-io/tidytuesdayR

BugReports https://github.com/dslc-io/tidytuesdayR/issues

Depends R (>= 4.1.0)

Imports cli, gh, glue, jsonlite, lubridate (>= 1.7.0), magrittr, markdown, purrr (>= 1.0.0), readr (>= 1.0.0), rlang, rvest (>= 0.3.2), stbl (>= 0.4.0), tidyr, tools (>= 3.1.0), utils, xml2 (>= 1.2.0)

Suggests base64enc, covr, dplyr, fs, knitr, magick, openssl, readxl (>= 1.0.0), rmarkdown, rstudioapi (>= 0.2), stringr, testthat (>= 3.0.0), tibble, usethis, vctrs, withr, yaml

VignetteBuilder knitr

Config/Needs/website pkgdown

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

Language en-US

NeedsCompilation no

Author Jon Harmon [aut, cre] (ORCID: <https://orcid.org/0000-0003-4781-4346>), Ellis Hughes [aut], Thomas Mock [ctb], Data Science Learning Community [dtc]

Maintainer Jon Harmon <jonthegeek@gmail.com>

Repository CRAN

Date/Publication 2026-09-11 15:20:02 UTC

Contents

last_tuesday 2
print.tt_data 3
readme 4
tt_available 4
tt_clean 5
tt_curate_data 6
tt_download 7
tt_download_file 7
tt_intro 8
tt_load 9
tt_load_gh 10
tt_meta 11
tt_print 12
tt_save_dataset 13
tt_submit 14
use_tidytemplate 15

Index **16**

`last_tuesday` *Find the most recent tuesday*

Description

Identify the most recent 'TidyTuesday' date relative to a specified date.

Usage

```
last_tuesday(date = today(tzone = "America/New_York"))
```

Arguments

`date`	A date as a date object or character string in `YYYY-MM-DD` format. Defaults to today's date.

Value

The TidyTuesday date in the same week as the specified date, using Monday as the start of the week.

Examples

```
last_tuesday() # get last Tuesday relative to today's date
last_tuesday("2020-01-01") # get last Tuesday relative to a specified date
```

`print.tt_data` *print methods of the tt objects*

Description

In tidytuesdayR there are nice print methods for the objects that were used to download and store the data from the TidyTuesday repo. They will always print the available datasets/files. If there is a readme available, it will try to display the TidyTuesday readme.

Usage

```
## S3 method for class 'tt_data'
print(x, ...)

## S3 method for class 'tt'
print(x, ...)
```

Arguments

`x`	a tt_data or tt object
`...`	further arguments passed to or from other methods.

Value

`x`, invisibly.

Examples

```
tt <- tt_load_gh("2019-01-15")
print(tt)

tt_data <- tt_download(tt, files = "All")
print(tt_data)
```

readme *Readme HTML maker and Viewer*

Description

Readme HTML maker and Viewer

Usage

```
readme(tt)
```

Arguments

tt	tt_data object for printing

Value

Null, invisibly. Used to show readme of the downloaded TidyTuesday dataset in the Viewer.

Examples

```
if (rate_limit_check(quiet = TRUE) > 30) {
  tt_output <- tt_load_gh("2019-01-15")
  readme(tt_output)
}
```

tt_available *Listing all available TidyTuesdays*

Description

The TidyTuesday project is a constantly growing repository of data sets. Knowing what type of data is available for each week requires going to the source. However, one of the hallmarks of 'tidytuesdayR' is that you never have to leave your R console. These functions were created to help maintain this philosophy.

Usage

```
tt_available(auth = gh::gh_token())

tt_datasets(year, auth = gh::gh_token())
```

Arguments

auth	A GitHub token. See gh::gh_token() for more details.
year	What year of TidyTuesday to use

Details

To find out the available datasets for a specific year, the user can use the function `tt_datasets()`. This function will either populate the Viewer or print to console all the available data sets and the week/date they are associated with.

To get the whole list of all the data sets ever released by TidyTuesday, the function `tt_available()` was created. This function will either populate the Viewer or print to console all the available data sets ever made for TidyTuesday.

Value

`tt_available()` returns a `tt_dataset_table_list`, which is a list of `tt_dataset_table`. This class has special printing methods to show the available data sets.

`tt_datasets()` returns a `tt_dataset_table` object. This class has special printing methods to show the available datasets for the year.

Examples

```
# check to make sure there are requests still available
if (rate_limit_check(quiet = TRUE) > 30) {
  ## show data available from 2018
  tt_datasets(2018)

  ## show all data available ever
  tt_available()
}
```

`tt_clean` *Create and open cleaning.R*

Description

The first step of curating a TidyTuesday dataset is cleaning the data. This function creates a simple `cleaning.R` file in the specified path (creating that path if it does not already exist), and (if possible) opens it for editing.

Usage

```
tt_clean(
  path = "tt_submission",
  open = rlang::is_interactive(),
  ignore = FALSE
)
```

Arguments

path	The relative path to the directory to hold your submission files (tt_submission by default). If this directory does not exist, it will be created.
open	Open the newly created file for editing? Happens in RStudio, if applicable, or via utils::file.edit() otherwise.
ignore	Should the newly created file be added to .Rbuildignore?

Value

A logical vector indicating whether the file was created or modified, invisibly.

Examples

```
tt_clean()
```

---

tt_curate_data *Guidance for TidyTuesday dataset curation*

Description

Open an R script to guide you through the process of curating and submitting a TidyTuesday dataset. See vignette("curating", package = "tidytuesdayR") for more information.

Usage

```
tt_curate_data(open = rlang::is_interactive())
```

Arguments

open	Whether to open the file for interactive editing.

Value

The path to the tt_curation.R script, invisibly.

Examples

```
tt_curate_data()
```

tt_download *Download TidyTuesday data*

Description

Download all or specific files identified in a TidyTuesday dataset.

Usage

```
tt_download(tt, files = "All", ..., auth = gh::gh_token())
```

Arguments

tt	A tt object, output from tt_load_gh().
files	Which file names to download. Default "All" downloads all files for the specified week.
...	Additional parameters to pass to the parsing functions. Note: These arguments will be passed for all filetypes.
auth	A GitHub token. See gh::gh_token() for more details.

Value

A list of tibbles from the downloaded files.

Examples

```
# Get the list of files for a week.
tt_output <- tt_load_gh("2019-01-15")

# Download a specific file.
agencies <- tt_download(tt_output, files = "agencies.csv")
```

tt_download_file *Download a TidyTuesday dataset file*

Description

Download an actual data file from the TidyTuesday github repository.

Usage

```
tt_download_file(tt, x, ..., auth = gh::gh_token())
```

Arguments

tt	A tt object, output from tt_load_gh().
x	Index or name of file to download.
...	Additional parameters to pass to the parsing functions. Note: These arguments will be passed for all filetypes.
auth	A GitHub token. See gh::gh_token() for more details.

Value

tibble containing the contents of the file downloaded from git

Examples

```
tt_gh <- tt_load_gh("2019-01-15")

agencies <- tt_download_file(tt_gh, 1)
launches <- tt_download_file(tt_gh, "launches.csv")
```

---

tt_intro *Create and open intro.md*

Description

When curating a TidyTuesday dataset, you need to introduce the dataset. This function creates a simple intro.md file in the specified path (creating that path if it does not already exist), and (if possible) opens it for editing.

Usage

```
tt_intro(
  path = "tt_submission",
  open = rlang::is_interactive(),
  ignore = FALSE
)
```

Arguments

path	The relative path to the directory to hold your submission files (tt_submission by default). If this directory does not exist, it will be created.
open	Open the newly created file for editing? Happens in RStudio, if applicable, or via utils::file.edit() otherwise.
ignore	Should the newly created file be added to .Rbuildignore?

Value

A logical vector indicating whether the file was created or modified, invisibly.

Examples

```
tt_intro()
```

`tt_load` *Load TidyTuesday data from Github*

Description

Load TidyTuesday data from Github

Usage

```
tt_load(x, week = NULL, files = "All", ..., auth = gh::gh_token())
```

Arguments

`x`	The date of data to pull (in "YYYY-MM-dd" format), or the four-digit year as a number.
`week`	Which week number to use within a given year. Only used when `x` is a valid year.
`files`	Which file names to download. Default "All" downloads all files for the specified week.
`...`	Additional parameters to pass to the parsing functions. Note: These arguments will be passed for all filetypes.
`auth`	A GitHub token. See `gh::gh_token()` for more details.

Value

`tt_data` object, which contains data that can be accessed via `$`, and the readme for the week's TidyTuesday, which can be viewed by printing the object or calling `readme()`.

Examples

```
tt_output <- tt_load("2019-01-15")
tt_output
agencies <- tt_output$agencies
```

`tt_load_gh` *Load TidyTuesday data from Github*

Description

Pulls the readme and URLs of the data from the TidyTuesday github folder based on the date provided

Usage

```
tt_load_gh(x, week = NULL, auth = gh::gh_token())
```

Arguments

`x`	The date of data to pull (in "YYYY-MM-dd" format), or the four-digit year as a number.
`week`	Which week number to use within a given year. Only used when `x` is a valid year.
`auth`	A GitHub token. See `gh::gh_token()` for more details.

Value

A `tt` object. This contains the files available for the week, readme html, and the date of the TidyTuesday.

Examples

```
# check to make sure there are requests still available
if (rate_limit_check(quiet = TRUE) > 30) {
  tt_gh <- tt_load_gh("2019-01-15")
  ## readme attempts to open the readme for the weekly dataset
  readme(tt_gh)

  agencies <- tt_download(
    tt_gh,
    files = "agencies.csv"
  )
}
```

`tt_meta` *Create and open meta.yaml*

Description

We need a set of metadata information about each TidyTuesday dataset. Use this function to set up the `meta.yaml` file for your submission (and create the submission directory if it does not already exist). If you do not provide values for the parameters, you will be prompted to enter them in an interactive session.

Usage

```
tt_meta(
  path = "tt_submission",
  title,
  article_title,
  article_url,
  source_title,
  source_url,
  image_filename,
  image_alt,
  attribution = getOption("tidytuesdayR.attribution"),
  github = gh::gh_whoami()$login,
  bluesky = getOption("tidytuesdayR.bluesky"),
  linkedin = getOption("tidytuesdayR.linkedin"),
  mastodon = getOption("tidytuesdayR.mastodon"),
  open = rlang::is_interactive(),
  ignore = FALSE
)
```

Arguments

`path`	The relative path to the directory to hold your submission files (`tt_submission` by default). If this directory does not exist, it will be created.
`title`	A short title for your submission. It should fit into the sentence "This week we're exploring `title`!" For example, for "This week we're exploring The 50 US States!", the `title` would be `"The 50 US States"`.
`article_title`	The title of an article or other website that has something to do with the data. This should usually be an article that uses or describes the dataset, but any related website is acceptable.
`article_url`	The URL of the article whose title is `article_title`.
`source_title`	The title of the source of the dataset. This is usually a website, but might be an R package or a journal article, for example.
`source_url`	A URL associated with the source. Ideally this should be a URL where users can download the data, but, if that isn't possible, provide a URL that is somehow related to the source of the data.

image_filename	A character vector with at least one file name for an image to accompany the post. This might be a plot of the data, or some othe image somehow connected to the data.
image_alt	Text that can take the place of the image for a visually impaired user or anybody else who cannot see the image. Don't just say "A plot of the data", but rather describe what information you can glean from the plot, such as "A map of the continental United States, with each state colored in shades of blue by population as of 1975. California and New York are the lightest, indicating the highest population. Maine, New Hampshire, Vermont, and the Plains States are all quite dark, indicating low population."
attribution	Your name as you would like it to appear when we credit you in the post for this dataset. You can include a title and/or affiliation if you like, such as "Jon Harmon, Executive Director, Data Science Learning Community". Defaults to `getOption("tidytuesdayR.attribution")`. We recommend setting this option in your `.Rprofile` with `usethis::edit_r_profile()` so you don't have to provide it every time you submit a dataset.
github	Your GitHub username, or a link to your profile on GitHub.
bluesky	Your Bluesky username, or a link to your profile on Bluesky. Leave as `NULL` if you do not wish to be credited on Bluesky. Defaults to `getOption("tidytuesdayR.bluesky")`.
linkedin	Your LinkedIn username, or a link to your profile on LinkedIn. Leave as `NULL` if you do not wish to be credited on LinkedIn. Defaults to `getOption("tidytuesdayR.linkedin")`.
mastodon	Your mastodon server and username, or a link to your profile on a mastodon server. Leave as `NULL` if you do not wish to be credited on Mastodon. Defaults to `getOption("tidytuesdayR.mastodon")`.
open	Open the newly created file for editing? Happens in RStudio, if applicable, or via `utils::file.edit()` otherwise.
ignore	Should the newly created file be added to `.Rbuildignore`?

Value

A logical vector indicating whether the file was created or modified, invisibly.

Examples

```
tt_meta()
```

`tt_print` *Printing Utilities for Listing Available Datasets*

Description

printing utilities for showing the available datasets for a specific year or all time

Usage

```
## S3 method for class 'tt_dataset_table'
print(x, ..., is_interactive = interactive())

## S3 method for class 'tt_dataset_table_list'
print(x, ..., is_interactive = interactive())
```

Arguments

x	an object used to select a method.
...	further arguments passed to or from other methods.
is_interactive	Whether the function is being used interactively.

Value

x, invisibly

Examples

```
# check to make sure there are requests still available
if (rate_limit_check(quiet = TRUE) > 30) {
  available_datasets_2018 <- tt_datasets(2018)
  print(available_datasets_2018)

  all_available_datasets <- tt_available()
  print(all_available_datasets)
}
```

---

tt_save_dataset *Save datasets for submission*

Description

Datasets for TidyTuesday submissions should be saved in a specific format, with an accompanying data dictionary `dataset_name.md` file. This function saves the dataset as a CSV file in your submission directory (creating the submission directory if it does not already exist), and creates a data dictionary file for you to fill out. If you're in an interactive session, the dictionary file is opened for editing.

Usage

```
tt_save_dataset(
  dataset,
  path = "tt_submission",
  dataset_name = rlang::caller_arg(dataset),
  open = rlang::is_interactive(),
  ignore = FALSE
)
```

Arguments

dataset	The clean dataset to save. The dataset must be a data.frame.
path	The relative path to the directory to hold your submission files (tt_submission by default). If this directory does not exist, it will be created.
dataset_name	The name to save the dataset as. By default, the name of the dataset variable is used.
open	Open the newly created file for editing? Happens in RStudio, if applicable, or via utils::file.edit() otherwise.
ignore	Should the newly created file be added to .Rbuildignore?

Value

A logical vector indicating whether the file was created or modified, invisibly.

Examples

```
tt_save_dataset(mtcars)
```

---

tt_submit *Submit a TidyTuesday dataset*

Description

Submit a curated dataset for review by uploading it to GitHub and creating a pull request. The dataset should be prepared using tt_clean(), tt_save_dataset(), tt_intro(), and tt_meta(). You can also use this function to submit changes to your local copies of the files.

Usage

```
tt_submit(
  path = "tt_submission",
  auth = gh::gh_token(),
  open = rlang::is_interactive()
)
```

Arguments

path	The relative path to the directory to hold your submission files (tt_submission by default). If this directory does not exist, it will be created.
auth	A GitHub token. See gh::gh_token() for more details.
open	Whether to open the pull request in a browser. Defaults to TRUE in an interactive session.

Value

The URL of the pull request, invisibly.

Examples

```
# First set up a dataset in the "tt_submission" folder.
tt_submit()
```

`use_tidytemplate` *Create and open the tidytemplate*

Description

Use the tidytemplate Rmd for starting your analysis with a leg up for processing

Usage

```
use_tidytemplate(
  name = NULL,
  open = rlang::is_interactive(),
  refdate = today(),
  ignore = FALSE
)
```

Arguments

`name`	A name for your generated TidyTuesday analysis Rmd, such as "My_TidyTuesday.Rmd".
`open`	Open the newly created file for editing? Happens in RStudio, if applicable, or via `utils::file.edit()` otherwise.
`refdate`	Date to use as reference to determine which TidyTuesday to use for the template. Either date object or character string in YYYY-MM-DD format.
`ignore`	Should the newly created file be added to `.Rbuildignore`?

Value

A logical vector indicating whether the file was created or modified, invisibly.

Examples

```
use_tidytemplate(name = "My_Awesome_TidyTuesday.Rmd")
```

Index

∗ **tt_download_file**
    tt_download_file, 7

gh::gh_token(), *4*, *7–10*, *14*

last_tuesday, 2

print.tt *(*print.tt_data*)*, 3
print.tt_data, 3
print.tt_dataset_table *(*tt_print*)*, 12
print.tt_dataset_table_list *(*tt_print*)*, 12
printing *(*print.tt_data*)*, 3

readme, 4
readme(), *9*

tt_available, 4
tt_clean, 5
tt_clean(), *14*
tt_curate_data, 6
tt_datasets *(*tt_available*)*, 4
tt_download, 7
tt_download_file, 7
tt_intro, 8
tt_intro(), *14*
tt_load, 9
tt_load_gh, 10
tt_load_gh(), *7*, *8*
tt_meta, 11
tt_meta(), *14*
tt_print, 12
tt_save_dataset, 13
tt_save_dataset(), *14*
tt_submit, 14

use_tidytemplate, 15
utils::file.edit(), *6*, *8*, *12*, *14*, *15*