

# Package ‘tidytuesdayR’

December 13, 2023

**Type** Package

**Title** Access the Weekly 'TidyTuesday' Project Dataset

**Version** 1.0.3

**Description** 'TidyTuesday' is a project by the 'R4DS Online Learning Community' in which they post a weekly dataset onto 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://github.com/thebioengineer/tidytuesdayR>

**BugReports** <https://github.com/thebioengineer/tidytuesdayR/issues>

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Depends** R (>= 3.4.0)

**Suggests** testthat (>= 2.1.0), covr, pkgdown, tibble, withr

**Imports** readxl (>= 1.0.0), rvest (>= 0.3.2), tools (>= 3.1.0),  
lubridate (>= 1.7.0), purrr (>= 0.2.5), readr (>= 1.0.0),  
rstudioapi (>= 0.2), xml2 (>= 1.2.0), httr, jsonlite, magrittr,  
usethis

**NeedsCompilation** no

**Author** Ellis Hughes [aut, cre],  
Jon Harmon [ctb],  
Thomas Mock [ctb],  
R4DS Online Learning Community [dct]

**Maintainer** Ellis Hughes <ellishughes@live.com>

**Repository** CRAN

**Date/Publication** 2023-12-13 10:30:02 UTC

## R topics documented:

available . . . . .	2
Available_Printing . . . . .	3
github_pat . . . . .	4
last_tuesday . . . . .	5
printing . . . . .	5
rate_limit_check . . . . .	6
readme . . . . .	7
tt_date . . . . .	7
tt_download . . . . .	8
tt_download_file . . . . .	9
tt_load . . . . .	10
tt_load_gh . . . . .	11
use_tidytemplate . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

available	<i>Listing all available TidyTuesdays</i>
-----------	---

---

### 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 = github_pat())

tt_datasets(year, auth = github_pat())
```

### Arguments

auth	github Personal Access Token. See PAT section for more information
year	numeric entry representing the year of TidyTuesday you want the list of datasets for. Leave empty for most recent year.

### 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.

**PAT**

A Github PAT is a Personal Access Token. This allows for signed queries to the github api, and increases the limit on the number of requests allowed from 60 to 5000. Follow instructions at <<https://happygitwithr.com/github-pat.html>> to set your PAT.

**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()
}
```

---

Available\_Printing      *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    is the console interactive?

**Value**

used for side effects to show the available datasets for the year or for all time.

## 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)

}
```

---

github\_pat

*Return the local user's GitHub Personal Access Token*

---

## Description

Extract the GitHub Personal Access Token (PAT) from the system environment for authenticated requests.

## Usage

```
github_pat(quiet = TRUE)
```

## Arguments

quiet            Should this be loud? default TRUE.

## Value

a character vector that is your Personal Access Token, or NULL

## PAT

A Github 'PAT' is a Personal Access Token. This allows for signed queries to the github api, and increases the limit on the number of requests allowed from 60 to 5000. Follow instructions from <<https://happygitwithr.com/github-pat.html>> to set the PAT.

## Examples

```
## if you have a personal access token saved, this will return that value
github_pat()
```

---

last_tuesday	<i>Find the most recent tuesday</i>
--------------	-------------------------------------

---

**Description**

Utility function to assist users in identifying the most recent 'TidyTuesday' date

**Usage**

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

**Arguments**

date                    todays date as a date object or character string in YYYY-MM-DD format.

**Examples**

```
last_tuesday() # get last tuesday from todays date  
last_tuesday("2020-01-01") # get last tuesday from specified date
```

---

printing	<i>print methods of the tt objects</i>
----------	--

---

**Description**

In tidyuesdayR 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**

used to show readme and list names of available datasets

used to show available datasets for the TidyTuesday

**Examples**

```
if(interactive()){  
  tt <- tt_load_gh("2019-01-15")  
  print(tt)  
  
  tt_data <- tt_download(tt, files = "All")  
  print(tt_data)  
}
```

---

rate\_limit\_check

*Get Rate limit left for GitHub Calls*

---

**Description**

The GitHub API limits the number of requests that can be sent within an hour. This function returns the stored rate limits that are remaining.

**Usage**

```
rate_limit_check(n = 10, quiet = FALSE)
```

**Arguments**

n	number of requests that triggers a warning indicating the user is close to the limit
quiet	should messages be returned when the rate limit is zero or less than n?

**Value**

return the number of calls are remaining as a numeric values

**Examples**

```
rate_limit_check()
```

---

readme	<i>Readme HTML maker and Viewer</i>
--------	-------------------------------------

---

**Description**

Readme HTML maker and Viewer

**Usage**

```
readme(tt)
```

**Arguments**

tt                    tt\_data object for printing

**Value**

Does not return anything. Used to show readme of the downloaded TidyTuesday dataset in the Viewer.

**Examples**

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

---

tt_date	<i>Get date of TidyTuesday, given the year and week</i>
---------	---

---

**Description**

Sometimes we don't know the date we want, but we do know the week. this function provides the ability to pass the year and week we are interested in to get the correct date

**Usage**

```
tt_date(year, week)
```

**Arguments**

year                    what year of TidyTuesday to use  
week                    what week of TidyTuesday to use

**Examples**

```
if(interactive()){
  tt_date(2019, week = 42)
}
```

---

tt_download	<i>download tt data</i>
-------------	-------------------------

---

**Description**

Download all or specific files identified in the tt dataset

**Usage**

```
tt_download(tt, files = c("All"), ..., branch = "master", auth = github_pat())
```

**Arguments**

tt	a 'tt' object, output from <a href="#">tt_load_gh</a>
files	List the file names to download. Default to asking.
...	pass methods to the parsing functions. These will be passed to ALL files, so be careful.
branch	which branch to be downloading data from. Default and always should be "master".
auth	github Personal Access Token. See PAT section for more information

**Value**

list of tibbles of the files downloaded.

**PAT**

A Github PAT is a personal Access Token. This allows for signed queries to the github api, and increases the limit on the number of requests allowed from 60 to 5000. Follow instructions at <https://happygitwithr.com/github-pat.html> to set the PAT.

**Examples**

```
if(interactive()){
  tt_output <- tt_load_gh("2019-01-15")
  agencies <- tt_download(tt_output, files = "agencies.csv")
}
```



---

tt_download_file	<i>Reads in TidyTuesday datasets from Github repo</i>
------------------	---

---

## Description

Reads in the actual data from the TidyTuesday github

## Usage

```
tt_download_file(tt, x, ..., auth = github_pat())
```

## Arguments

tt	tt_gh object from tt_load_gh function
x	index or name of data object to read in. string or int
...	pass methods to the parsing functions. These will be passed to ALL files, so be careful.
auth	github Personal Access Token. See PAT section for more information

## Value

tibble containing the contents of the file downloaded from git

## PAT

A Github PAT is a personal Access Token. This allows for signed queries to the github api, and increases the limit on the number of requests allowed from 60 to 5000. Follow instructions at <<https://happygitwithr.com/github-pat.html>> to set the PAT.

## Examples

```
if(interactive()){  
  tt_gh <- tt_load_gh("2019-01-15")  
  
  agencies <- tt_download_file(tt_gh, 1)  
  launches <- tt_download_file(tt_gh, "launches.csv")  
}
```

---

tt\_load *Load TidyTuesday data from Github*

---

### Description

Load TidyTuesday data from Github

### Usage

```
tt_load(x, week, download_files = "All", ..., auth = github_pat())
```

### Arguments

x	string representation of the date of data to pull, in YYYY-MM-dd format, or just numeric entry for year
week	left empty unless x is a numeric year entry, in which case the week of interest should be entered
download_files	which files to download from repo. defaults and assumes "All" for the week.
...	pass methods to the parsing functions. These will be passed to ALL files, so be careful.
auth	github Personal Access Token. See PAT section for more information

### Value

tt\_data object, which contains data that can be accessed via '\$', and the readme for the weeks TidyTuesday through printing the object or calling 'readme()'

### PAT

A Github PAT is a personal Access Token. This allows for signed queries to the github api, and increases the limit on the number of requests allowed from 60 to 5000. Follow instructions from <<https://happygitwithr.com/github-pat.html>> to set the PAT.

### Examples

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

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

}
```

---

tt_load_gh	<i>Load TidyTuesday data from Github</i>
------------	--

---

**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, auth = github_pat())
```

**Arguments**

x	string representation of the date of data to pull, in YYYY-MM-dd format, or just numeric entry for year
week	left empty unless x is a numeric year entry, in which case the week of interest should be entered
auth	github Personal Access Token. See PAT section for more information

**Value**

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

**PAT**

A Github PAT is a personal Access Token. This allows for signed queries to the github api, and increases the limit on the number of requests allowed from 60 to 5000. Follow instructions from <<https://happygitwithr.com/github-pat.html>> to set the PAT.

**Examples**

```
# check to make sure there are requests still available
if(rate_limit_check(quiet = TRUE) > 30 & interactive()){
  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"
  )
}
```

use\_tidytemplate      *Call 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 = interactive(), ..., refdate = today())
```

**Arguments**

name	name of your TidyTuesday analysis file
open	should the file be opened after being created
...	arguments to be passed to <a href="#">use_template</a>
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.

**Examples**

```
if(interactive()){  
  use_tidytemplate(name = "My_Awesome_TidyTuesday.Rmd")  
}
```

# Index

- \* **tt\_download\_file**
  - tt\_download\_file, 9
- available, 2
- Available\_Printing, 3
- github\_pat, 4
- last\_tuesday, 5
- print.tt (printing), 5
- print.tt\_data (printing), 5
- print.tt\_dataset\_table
  - (Available\_Printing), 3
- print.tt\_dataset\_table\_list
  - (Available\_Printing), 3
- printing, 5
- rate\_limit\_check, 6
- readme, 7
- tt\_available (available), 2
- tt\_datasets (available), 2
- tt\_date, 7
- tt\_download, 8
- tt\_download\_file, 9
- tt\_load, 10
- tt\_load\_gh, 8, 11
- use\_template, 12
- use\_tidytemplate, 12