Package ‘anyflights’

January 10, 2025

Title Query 'nycflights13'-Like Air Travel Data for Given Years and Airports

Version 0.3.5

Description Supplies a set of functions to query air travel data for user-specified years and airports. Datasets include on-time flights, airlines, airports, planes, and weather.

License CC0

Depends R (>= 3.5.0)

Imports httr, dplyr, readr, utils, lubridate, vroom, glue, purrr, stringr, curl, usethis, roxygen2, progress, tidyr

URL https://github.com/simonpcouch/anyflights, https://simonpcouch.github.io/anyflights/

BugReports https://github.com/simonpcouch/anyflights/issues

RoxygenNote 7.3.2

Encoding UTF-8

Suggests testthat, nycflights13, covr

NeedsCompilation no

Author Simon P. Couch [aut, cre],
Hadley Wickham [ctb],
Jay Lee [ctb],
Dennis Irorere [ctb]

Maintainer Simon P. Couch <simonpatrickcouch@gmail.com>

Repository CRAN

Date/Publication 2025-01-10 20:30:02 UTC

Contents

anyflights 2
anyflights_description 3
as_flights_package 4

get_airlines 4
get_airports 5
get_flights 6
get_planes 8
get_weather 9

Index **11**

anyflights *Query nycflights13-Like Air Travel Data*

Description

This function generates a list of dataframes similar to those found in the `nycflights13` data package for any US airports and time frames. Please note that, even with a strong internet connection, this function may take several minutes to download relevant data.

Usage

```
anyflights(station, year, month = 1:12, dir = NULL)
```

Arguments

`station`	A character vector giving the origin US airports of interest (as the FAA LID airport code).
`year`	A numeric giving the year of interest. This argument is currently not vectorized, as dataset sizes for single years are significantly large. Information for the most recent year is usually available by February or March in the following year.
`month`	A numeric giving the month(s) of interest.
`dir`	An optional character string giving the directory to save datasets in. By default, datasets will not be saved to file.

Details

The `anyflights()` function is a wrapper around the following functions:

- `get_airlines`: Grab data to translate between two letter carrier codes and names
- `get_airports`: Grab data on airport names and locations
- `get_flights`: Grab data on all flights that departed given US airports in a given year and month
- `get_planes`: Grab construction information about each plane
- `get_weather`: Grab hourly meterological data for a given airport in a given year and month

The recommended approach to download data for many stations (airports) is to supply a vector of stations to the `station` argument rather than iterating over many calls to `anyflights()`. The `faa` column in dataframes outputted by `get_airports()` provides the FAA LID codes for all supported airports. See ?`get_flights` for more details on implementation.

Value

A list of dataframes (and, optionally, a directory of datasets) similar to those found in the `nycflights13` data package.

See Also

`get_flights` for flight data, `get_weather` for weather data, `get_airlines` for airlines data, `get_airports` for airports data, or `get_planes` for planes data.

Use the `as_flights_package` function to convert the output of this function to a data-only package.

Examples

```
# grab data on all flights departing from
# Portland International Airport in June 2019 and
# other useful metadata without saving to file
## Not run: anyflights("PDX", 2018, 6)

# ...or, grab that same data and opt to save the
# file as well! (tempdir() can usually be specified
# as a character string giving the path to a folder)
## Not run: anyflights("PDX", 2018, 6, tempdir())
```

---

anyflights_description

anyflights: 'nycflights13'-Like Data for Specified Years and Airports

---

Description

The anyflights package supplies a set of functions to generate `nycflights13`-like datasets and data packages for specified years and airports.

Author(s)

Maintainer: Simon P. Couch <simonpatrickcouch@gmail.com>

Other contributors:

- Hadley Wickham <hadley@rstudio.com> [contributor]
- Jay Lee <jaylee@reed.edu> [contributor]
- Dennis Irorere <denironyx@gmail.com> [contributor]

See Also

Useful links:

- https://github.com/simonpcouch/anyflights
- https://simonpcouch.github.io/anyflights/
- Report bugs at https://github.com/simonpcouch/anyflights/issues

as_flights_package *Generate a Data Package from 'anyflights' Data*

Description

Generate a data-only package, including documentation, from data outputted by the 'anyflights()' function. Please do not submit the outputted package to CRAN or similar repositories as original packages.

Usage

```
as_flights_package(data, name = make.names(deparse(substitute(data))))
```

Arguments

`data`	A named list of dataframes outputted by `anyflights`.
`name`	The desired name of the resulting package as a character string. The package will check that the supplied package name is valid using the regular expression `.standard_regexps()$valid_package_name`, and save the output in a directory by the same name. Defaults to `make.names(deparse(substitute(data)))`.

Value

A directory containing a data-only package built around the supplied data.

get_airlines *Query nycflights13-Like Airlines Data*

Description

This function generates a dataframe similar to the `airlines` dataset from `nycflights13` for any US airports and time frame. Please note that, even with a strong internet connection, this function may take several minutes to download relevant data.

Usage

```
get_airlines(dir = NULL, flights_data = NULL)
```

Arguments

`dir`	An optional character string giving the directory to save datasets in. By default, datasets will not be saved to file.
`flights_data`	Optional—either a filepath as a character string or a dataframe outputted by `get_flights` that will be used to subset the output to only include relevant carriers/planes. If not supplied, all carriers/planes will be returned.

Value

A data frame with <2k rows and 2 variables:

carrier Two or three length letter or number abbreviation. In cases whgere the the Unique Carrier Code has been use more than once, a suffix is added. ex. ML, ML (1). This list matches the 'Reporting_Airline' field in the BTS documentation for the flights data set

name Full name

Source

https://www.bts.gov/

See Also

get_flights for flight data, get_weather for weather data, get_airports for airports data, get_planes for planes data, or anyflights for a wrapper function.

Use the as_flights_package function to convert this dataset to a data-only package.

Examples

```
# run with defaults
## Not run: get_airlines()

# if you'd like to only return the airline
# abbreviations only for airlines that appear in
# \code{flights}, query your flights dataset first,
# and then supply it as a flights_data argument
## Not run: get_airlines(flights_data = get_flights("PDX", 2018, 6))
```

---

get_airports *Query nycflights13-Like Airports Data*

Description

This function generates a dataframe similar to the airports dataset from nycflights13 for any US airports and time frame. Please note that, even with a strong internet connection, this function may take several minutes to download relevant data.

Usage

```
get_airports(dir = NULL)
```

Arguments

dir	An optional character string giving the directory to save datasets in. By default, datasets will not be saved to file.

Value

A data frame with ~1350 rows and 8 variables:

faa FAA airport code

name Usual name of the airport

lat, lon Location of airport

alt Altitude, in feet

tz Timezone offset from GMT/UTC

dst Daylight savings time zone. A = Standard US DST: starts on the second Sunday of March, ends on the first Sunday of November. U = unknown. N = no dst.

tzone IANA time zone, as determined by GeoNames webservice

Source

‘https://openflights.org/data.html‘

See Also

`get_flights` for flight data, `get_weather` for weather data, `get_airlines` for airlines data, `get_planes` for planes data, or `anyflights` for a wrapper function.

Use the `as_flights_package` function to convert this dataset to a data-only package.

Examples

```
# grab airports data
## Not run: get_airports()
```

`get_flights` *Query nycflights13-Like Flights Data*

Description

This function generates a dataframe similar to the `flights` dataset from `nycflights13` for any US airport and time frame. Please note that, even with a strong internet connection, this function may take several minutes to download relevant data.

Usage

```
get_flights(station, year, month = 1:12, dir = NULL, ...)
```

Arguments

`station` A character vector giving the origin US airports of interest (as the FAA LID airport code).

`year` A numeric giving the year of interest. This argument is currently not vectorized, as dataset sizes for single years are significantly large. Information for the most recent year is usually available by February or March in the following year.

`month` A numeric giving the month(s) of interest.

`dir` An optional character string giving the directory to save datasets in. By default, datasets will not be saved to file.

`...` Currently only used internally.

Details

This function currently downloads data for *all* stations for each month supplied, and *then* filters out data for relevant stations. Thus, the recommended approach to download data for many airports is to supply a vector of airport codes to the `station` argument rather than iterating over many calls to `get_flights()`.

Value

A data frame with ~1k-500k rows and 19 variables:

`year, month, day` Date of departure

`dep_time, arr_time` Actual departure and arrival times, UTC.

`sched_dep_time, sched_arr_time` Scheduled departure and arrival times, UTC.

`dep_delay, arr_delay` Departure and arrival delays, in minutes. Negative times represent early departures/arrivals.

`hour, minute` Time of scheduled departure broken into hour and minutes.

`carrier` Two letter carrier abbreviation. See `get_airlines` to get full name

`tailnum` Plane tail number

`flight` Flight number

`origin, dest` Origin and destination. See `get_airports` for additional metadata.

`air_time` Amount of time spent in the air, in minutes

`distance` Distance between airports, in miles

`time_hour` Scheduled date and hour of the flight as a `POSIXct` date. Along with `origin`, can be used to join flights data to weather data.

Note

If you are repeatedly getting a timeout error when downloading flights, this could be because your download is taking longer than the default timeout R option. You can change the timeout value for your R session by running the code `options(timeout = timeout_value_in_seconds)` in your console.

Source

RITA, Bureau of transportation statistics, https://www.bts.gov

See Also

get_weather for weather data, get_airlines for airlines data, get_airports for airports data, get_planes for planes data, or anyflights for a wrapper function.

Use the as_flights_package function to convert this dataset to a data-only package.

Examples

```
# flights out of Portland International in June 2018
## Not run: get_flights("PDX", 2018, 6)

# ...or the original nycflights13 flights dataset
## Not run: get_flights(c("JFK", "LGA", "EWR"), 2013)

# use the dir argument to indicate the folder to
# save the data in \code{dir} as "flights.rda"
## Not run: get_flights("PDX", 2018, 6, dir = tempdir())
```

---

get_planes *Query nycflights13-Like Planes Data*

Description

This function generates a dataframe similar to the planes dataset from nycflights13 for any US airports and time frame. Please note that, even with a strong internet connection, this function may take several minutes to download relevant data.

Usage

```
get_planes(year, dir = NULL, flights_data = NULL)
```

Arguments

year A numeric giving the year of interest. This argument is currently not vectorized, as dataset sizes for single years are significantly large. Information for the most recent year is usually available by February or March in the following year.

dir An optional character string giving the directory to save datasets in. By default, datasets will not be saved to file.

flights_data Optional—either a filepath as a character string or a dataframe outputted by get_flights that will be used to subset the output to only include relevant carriers/planes. If not supplied, all carriers/planes will be returned.

Value

A data frame with ~3500 rows and 9 variables:

tailnum Tail number

year Year manufactured

type Type of plane

manufacturer, model Manufacturer and model

engines, seats Number of engines and seats

speed Average cruising speed in mph

engine Type of engine

Source

FAA Aircraft registry, https://www.faa.gov/licenses_certificates/aircraft_certification/aircraft_registry/releasable_aircraft_download

See Also

get_flights for flight data, get_weather for weather data, get_airlines for airlines data, get_airports for airports data, or anyflights for a wrapper function.

Use the as_flights_package function to convert this dataset to a data-only package.

Examples

```
# grab airplanes data for 2018
## Not run: get_planes(2018)

# if you'd like to only return the planes that appear
# in \code{flights}, query your flights dataset first,
# and then supply it as a \code{flights_data} argument
## Not run: get_planes(2018,
                 flights_data = get_flights("PDX", 2018, 6))
## End(Not run)
```

---

get_weather *Query nycflights13-Like Weather Data*

Description

This function generates a dataframe similar to the weather dataset from nycflights13 for any US airports and time frame. Please note that, even with a strong internet connection, this function may take several minutes to download relevant data.

Usage

```
get_weather(station, year, month = 1:12, dir = NULL)
```

Arguments

`station` A character vector giving the origin US airports of interest (as the FAA LID airport code).

`year` A numeric giving the year of interest. This argument is currently not vectorized, as dataset sizes for single years are significantly large. Information for the most recent year is usually available by February or March in the following year.

`month` A numeric giving the month(s) of interest.

`dir` An optional character string giving the directory to save datasets in. By default, datasets will not be saved to file.

Value

A data frame with ~1k-25k rows and 15 variables:

`origin` Weather station. Named `origin` to facilitate merging with flights data

`year`, `month`, `day`, `hour` Time of recording, UTC

`temp`, `dewp` Temperature and dewpoint in F

`humid` Relative humidity

`wind_dir`, `wind_speed`, `wind_gust` Wind direction (in degrees), speed and gust speed (in mph)

`precip` Precipitation, in inches

`pressure` Sea level pressure in millibars

`visib` Visibility in miles

`time_hour` Date and hour of the recording as a `POSIXct` date, UTC

Source

ASOS download from Iowa Environmental Mesonet, https://mesonet.agron.iastate.edu/request/download.phtml

See Also

`get_flights` for flight data, `get_airlines` for airlines data, `get_airports` for airports data, `get_planes` for planes data, or `anyflights` for a wrapper function.

Use the `as_flights_package` function to convert this dataset to a data-only package.

Examples

```
# query weather at Portland International in June 2018
## Not run: get_weather("PDX", 2018, 6)

# ...or the original nycflights13 weather dataset
## Not run: get_weather(c("JFK", "LGA", "EWR"), 2013)

# use the dir argument to indicate the folder to
# save the data in as "weather.rda"
## Not run: get_weather("PDX", 2018, 6, dir = tempdir())
```

Index

airlines, *4*

airports, *5*

anyflights, 2, *4–6*, *8–10*

anyflights-package (anyflights_description), 3

anyflights_description, 3

anyflights_package (anyflights_description), 3

as_flights_package, *3*, 4, *5*, *6*, *8–10*

flights, *6*

get_airlines, *2*, *3*, 4, *6–10*

get_airports, *2*, *3*, *5*, 5, *7–10*

get_flights, *2–6*, 6, *8–10*

get_planes, *2*, *3*, *5*, *6*, *8*, 8, *10*

get_weather, *2*, *3*, *5*, *6*, *8*, *9*, 9

planes, *8*

weather, *9*