

Package: RKaggle (via r-universe)

May 28, 2026

Title 'Kaggle' Dataset Downloader 'API'

Version 0.1.1

Description Easily download datasets from Kaggle
<<https://www.kaggle.com/>> directly into your R environment
using 'RKaggle'. Streamline your data analysis workflows by
importing datasets effortlessly and focusing on insights rather
than manual data handling. Perfect for data enthusiasts and
professionals looking to integrate Kaggle datasets into their R
projects with minimal hassle.

License MIT + file LICENSE

URL <https://github.com/benyamindsmith/RKaggle>

BugReports <https://github.com/benyamindsmith/RKaggle/issues>

Encoding UTF-8

LazyData true

Imports httr, readr, withr, jsonlite, arrow, readODS, readxl, sf

Depends R (>= 4.1.0)

Suggests tibble, conflicted

RoxygenNote 7.3.2

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libssl-dev libproj-
dev libsqlite3-dev libudunits2-dev libx11-dev

Repository <https://benyamindsmith.r-universe.dev>

Date/Publication 2025-07-25 01:27:40 UTC

RemoteUrl <https://github.com/benyamindsmith/rkaggle>

RemoteRef HEAD

RemoteSha 242c005a363d9376006930406fa33f4fff72b75f

Contents

get_dataset 2

Index 4

`get_dataset`*Download and Read a Dataset from Kaggle*

Description

This function retrieves a dataset from Kaggle by downloading its metadata and associated ZIP file and then reads all supported files contained in its archive. Each supported file is loaded into appropriate function (see details for more information about this). The function returns a single data frame if there is only one file detected and an unnamed list of data frames otherwise. This function is only capable of pulling data from Kaggle Datasets and not competitions.

Usage

```
get_dataset(dataset)
```

Arguments

`dataset` A character string specifying the dataset identifier on Kaggle. It should follow the format "username/dataset-name".

Details

The function constructs the metadata URL based on the provided dataset identifier, then sends a GET request using the `httr` package. If the request is successful, the returned JSON metadata is parsed. The function searches the metadata for a file with an encoding format of "application/zip", then downloads that ZIP file using a temporary file (managed by the `withr` package). After unzipping the file into a temporary directory, the function locates all files with extensions corresponding to popular dataset formats (csv, tsv, xlsx, json, rds, parquet, ods, shp, geojson and feather). Each file is then read using the appropriate function:

- `readr::read_csv` for CSV files.
- `readr::read_tsv` for TSV files.
- `readxl::read_excel` for xlsx files.
- `jsonlite::fromJSON` for JSON files.
- `readRDS` for RDS files.
- `arrow::read_parquet` for Parquet files.
- `readODS::read_ods` for ODS files
- `sf::read_sf` for SHP and GEOJSON files.
- `arrow::read_feather` for Feather files.

The function stops with an error if any of the following occur:

- The HTTP request fails.
- No ZIP file URL is found in the metadata.
- No supported data files are found in the unzipped contents.

Value

An unnamed list of dataframes corresponding to the files that were able to be read by `get_data()`. If only one file is able to be read, a individual dataframe is returned.

Examples

```
# Download and read the "canadian-prime-ministers" dataset from Kaggle
canadian_prime_ministers <- get_dataset("benjaminsmith/canadian-prime-ministers")
canadian_prime_ministers

# csv
canadian_prime_ministers <- get_dataset("benjaminsmith/canadian-prime-ministers")
# tsv
arabic_twitter <- get_dataset("mksaad/arabic-sentiment-twitter-corpus")
# xlsx
hr_data <- get_dataset("kmlDas/hr-employee-data-descriptive-analytics")
# json
iris_json <- get_dataset("rtatman/iris-dataset-json-version")
# rds
br_pop_2019 <- get_dataset("ianfukushima/br-pop-2019")
# parquet
iris_datasets <- get_dataset("gpreda/iris-dataset")
# ods
new_houses <- get_dataset("nm8883/new-houses-built-each-year-in-england")
# shp
india_states <- get_dataset("dhruvanurag20/final-shp")
# geojson
montreal <- get_dataset("rinichristy/montreal-geojson")
# feather
ncaa <- get_dataset("corochann/ncaa-march-madness-2020-womens")
```

Index

`get_dataset`, [2](#)