

Package ‘m5’

October 13, 2022

Type Package

Title 'M5 Forecasting' Challenges Data

Version 0.1.1

Description Contains functions, which facilitate downloading, loading and preparing data from 'M5 Forecasting' challenges (by 'University of Nicosia', hosted on 'Kaggle'). The data itself is set of time series of different product sales in 'Walmart'. The package also includes a ready-to-use built-in M5 subset named 'tiny_m5'. For detailed information about the challenges, see: Makridakis, S. & Spiliotis, E. & Assimakopoulos, V. (2020). <[doi:10.1016/j.ijforecast.2021.10.009](https://doi.org/10.1016/j.ijforecast.2021.10.009)>.

Encoding UTF-8

LazyData true

BugReports <https://github.com/krzjoa/m5/issues>

URL <https://github.com/krzjoa/m5>, <https://krzjoa.github.io/m5/>

RoxygenNote 7.1.2

License MIT + file LICENSE

Depends R (>= 3.5.0)

Suggests ggplot2, knitr, testthat (>= 3.0.0), rmarkdown, zeallot

VignetteBuilder knitr

Imports data.table, stringi, lubridate

Config/testthat/edition 3

NeedsCompilation no

Author Krzysztof Joachimiak [aut, cre]
(<<https://orcid.org/0000-0003-4780-7947>>)

Maintainer Krzysztof Joachimiak <joachimiak.krzysztof@gmail.com>

Repository CRAN

Date/Publication 2022-09-12 08:32:55 UTC

R topics documented:

m5_demand_type	2
m5_download	3
m5_get_raw	4
m5_prepare	5
tiny_m5	6

Index	8
--------------	----------

m5_demand_type	<i>Classify time series of the particular items</i>
----------------	---

Description

Each time series in the dataset can be assigned one of the following classes:

Usage

```
m5_demand_type(data)
```

Arguments

data The result of the m5_prepare function; tiny_m5 can be passed as well.

Details

- Smooth ($ADI < 1.32$ and $CV^2 < 0.49$).
- Intermittent ($ADI \geq 1.32$ and $CV^2 < 0.49$)
- Erratic ($ADI < 1.32$ and $CV^2 \geq 0.49$)
- Lumpy ($ADI \geq 1.32$ and $CV^2 \geq 0.49$)

Value

A data.table containing item ids (item_id and store_id), ADI and CV2 scores (adi and cv2 respectively) as well as the final class chosen based on the aforementioned scores (demand_type).

References

Syntetos A. A. and Boylan J. E., 2005, The accuracy of intermittent demand estimates. *International Journal of Forecasting* 21: 303–314 *Forecast Error Measures: Intermittent Demand*

Examples

```
head(m5_demand_type(tiny_m5))
```

m5_download	<i>Download and unzip the raw data to the specified directory</i>
-------------	---

Description

Download and unzip the raw data to the specified directory

Usage

```
m5_download(path, unzip = TRUE)
```

Arguments

path	A directory name to save the zip file
unzip	Automatically unzip the file when the downloading is finished. Default: TRUE. The exdir argument in the unzip function is the directory name the file was downloaded into.

Value

Returns nothing; the result of the function call is the m5.zip file downloading and extraction.

Note

If you struggle with timeout problems, please increase the timeout value using `options(timeout=<new_timeout_value>)`

References

- [M5 Forecasting - Accuracy](#)
- [M5 Forecasting - Uncertainty](#)

Examples

```
m5_download('data')
```

`m5_get_raw`*Load raw CSV files using `data.table::fread()` function*

Description

Load raw CSV files using `data.table::fread()` function

Usage

```
m5_get_raw_evaluation(path)
```

```
m5_get_raw_validation(path)
```

Arguments

`path` The directory with the unzipped M5 data files

Value

The function returns a list of five `data.tables`:

- `sales_train` (evaluation/validation)
- `sales_test` (evaluation/validation)
- `sell_prices`
- `calendar`
- `weights` (evaluation/validation)

References

[m5-forecasts repo by Nixtla](#)

Examples

```
## Not run:
library(m5)
library(zeallot)

m5_download('data')
c(sales_train,
  sales_test,
  sell_prices,
  calendar,
  ) %<-% m5_get_raw_evaluation('data')

## End(Not run)
```

m5_prepare	<i>Prepare the ready-to-use M5 data in one data.frame</i>
------------	---

Description

It's a memory-efficient function, which uses `data.table` under the hood. However, it still not recommended to use this function on PCs with < 16GB RAM. In such case, consider to use a custom solution based on `[arrow]`(<https://arrow.apache.org/docs/r/>) or `[disk.frame]`(<https://diskframe.com/index.htm>).

Usage

```
m5_prepare(sales_train, sales_test, calendar, sell_prices)
```

Arguments

<code>sales_train</code>	A <code>data.frame</code> with M5 train data
<code>sales_test</code>	A <code>data.frame</code> with M5 test data
<code>calendar</code>	A <code>data.frame</code> with M5 calendar
<code>sell_prices</code>	A <code>data.frame</code> with M5 <code>sell_prices</code>

Value

A `data.table` composed from input objects, which contains the following columns:

- `item_id`
- `dept_id`
- `cat_id`
- `store_id`
- `state_id`
- `d` - day ordinal number
- `value` - number of sold items
- `wm_yr_wk` - week identifier
- `weekday` - weekday name (character)
- `wday` - weekday as an integer
- `month`
- `year`
- `event_name_1` - special event name, like holidays etc.
- `event_type_1` - special event type
- `event_name_2` - as above
- `event_type_2` - as above
- `snap` - promotion flag
- `sell_price`

Examples

```
library(m5)
library(zeallot)

m5_download('data')
c(sales_train,
  sales_test,
  sell_prices,
  calendar,
  weights) %<-% m5_get_raw_evaluation('data')

m5_data <-
  m5_prepare(sales_train, sales_test, calendar, sell_prices)
```

 tiny_m5

A subset from M5 Walmart Challenge Dataset in one data frame

Description

A piece of data cut from the training dataset used in the M5 challenges on Kaggle. M5 is a challenge from a series organized by Spyros Makridakis.

Usage

```
tiny_m5
```

Format

item_id The id of the product
dept_id The id of the department the product belongs to
cat_id The id of the category the product belongs to
store_id The id of the store where the product is sold
state_id The State where the store is located
value The number of sold units
date The date in a “y-m-d” format
wm_yr_wk The id of the week the date belongs to
weekday The type of the day (Saturday, Sunday, ..., Friday)
wday The id of the weekday, starting from Saturday
month The month of the date
year The year of the date
event_name_1 If the date includes an event, the name of this event

event_type_1 If the date includes an event, the type of this event

event_name_2 If the date includes a second event, the name of this event

event_type_2 If the date includes a second event, the type of this event

snap A binary variable (0 or 1) indicating whether the stores of CA, TX or WI allow SNAP purchases on the examined date. 1 indicates that SNAP purchases are allowed

sell_price The price of the product for the given week/store. The price is provided per week (average across seven days). If not available, this means that the product was not sold during the examined week. Note that although prices are constant at weekly basis, they may change through time (both training and test set)

See Also

[M5 Forecasting - Accuracy](#)

[M5 Forecasting - Uncertainty](#)

[The M5 competition: Background, organization, and implementation](#)

[Other Walmart datasets in timetk](#)

Examples

```
library(m5)
# Head of tiny_m5
head(tiny_m5)
```

Index

* datasets

tiny_m5, 6

data.table::fread(), 4

m5_demand_type, 2

m5_download, 3

m5_get_raw, 4

m5_get_raw_evaluation (m5_get_raw), 4

m5_get_raw_validation (m5_get_raw), 4

m5_prepare, 5

tiny_m5, 6