Package 'rpredictit'

October 14, 2022

Title Interface to the 'PredictIt' API

Version 0.1.0

Description Wrapper to retrieve market data, explore available markets, and plot historical price data from the 'PredictIt' public API (<https://www.predictit.org/api/marketdata/all/>). The package comes with a demo 'shiny' application for illustrating example use cases. License to use data made available via the API is for non-commercial use and 'PredictIt' is the sole source of such data.

License MIT + file LICENSE

Encoding UTF-8

Imports httr, jsonlite, dplyr, DT, dygraphs, magrittr, quantmod, xts, shiny

RoxygenNote 7.2.0

URL https://github.com/danielkovtun/rpredictit

BugReports https://github.com/danielkovtun/rpredictit/issues

Suggests testthat (>= 2.1.0), knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Daniel Kovtun [cre, aut]

Maintainer Daniel Kovtun <quantumfusetrader@gmail.com>

Repository CRAN

Date/Publication 2022-07-19 12:00:02 UTC

R topics documented:

`all_markets` *Get bids and asks for all 'PredictIt' markets*

Description

Wrapper function to get all available 'PredictIt' markets and contract prices.

Usage

```
all_markets()
```

Value

A `tibble` containing bid and ask data for all 'PredictIt' markets.

Examples

```
markets <- all_markets()
markets
```

`format_market_data` *Format bid and ask market data with HTML*

Description

Wrapper function to apply HTML formatting to 'PredictIt' market data. Can be displayed in a `shiny` app, or standalone in an `htmlwidget` (e.g. `datatable`).

Usage

```
format_market_data(data)
```

Arguments

`data`	'PredictIt' market data, of class `data.frame` or `tibble`, as returned by `all_markets()` or `single_market`.

Value

A `tibble` containing bid and ask data formatted with HTML tags and user-friendly column names.

Examples

```
## Only run this example in interactive R sessions
if (interactive()) {
    data <- all_markets()
    format_market_data(data)
}
```

`historical_plot` *Plot historical contract data obtained from the 'PredictIt' website*

Description

Function to make an interactive `dygraphs::dygraph` plot of historical contract data.

Usage

```
historical_plot(contract_data)
```

Arguments

`contract_data`	Named list containing contract name and data of class `xts`, as returned by `parse_historical_csv()`.

Value

Interactive `dygraphs::dygraph` plot containing time series data for contract 'close' prices.

Examples

```
filename <- "What_will_be_the_balance_of_power_in_Congress_after_the_2020_election.csv"
csv_path <- system.file("extdata", filename, package = "rpredictit")
contract_data <- parse_historical_csv(csv_path)
historical_plot(contract_data)
```

`markets_table` *Get JavaScript datatable containing bids and asks for all 'PredictIt' markets*

Description

Wrapper function to return a `datatable` containing 'PredictIt' market data. Can be displayed in a `shinyApp`, RMarkdown document, or exported via `saveWidget()`.

Usage

```
markets_table(data)
```

Arguments

data	'PredictIt' market data, of class data.frame or tibble, as returned by all_markets() or single_market.

Value

An interactive datatable object containing formatted bid and ask data for the provided market data.

Examples

```
data <- all_markets()
markets_table(data)
```

---

parse_historical_csv *Parse csv file containing historical OHLCV data*

Description

Helper function to parse a 'csv' file obtained from the 'PredictIt' website, containing historical 'OHLCV' (Open, High, Low, Close, Volume) data, into an object of class xts.

Usage

```
parse_historical_csv(csv_path, filename = NA)
```

Arguments

csv_path	Path to a 'csv' file containing historical 'OHLCV' data for a specific contract. Expected format is the same schema as the 'csv' file downloaded from the 'PredictIt' website.
filename	Optional name to give the 'csv' file when the filepath is derived from a temporary directory.

Value

A named list containing the following elements:

data An S3 object of class xts. Time series containing the 'close' price data for the contract provided.

contract A character representing the contract name, derived from the input file name. If a filename argument is provided, the contract name will be assigned to that value.

Examples

```
filename <- "What_will_be_the_balance_of_power_in_Congress_after_the_2020_election.csv"
csv_path <- system.file("extdata", filename, package = "rpredictit")
parse_historical_csv(csv_path)
```

`runExample` *Run rpredictit examples*

Description

Launch a `rpredictit` example Shiny app that shows how to easily use `rpredictit` in an app.

Run without any arguments to see a list of available example apps.

Usage

```
runExample(example)
```

Arguments

`example`	The app to launch

Value

None. Runs a demo Shiny application. This function normally does not return; interrupt R to stop the application.

Examples

```
## Only run this example in interactive R sessions
if (interactive()) {
  # List all available example apps
  runExample()

  runExample("demo")
}
```

single_market *Get bids and asks for a specific 'PredictIt' market*

Description

Wrapper function to get data for a specific market.

Usage

```
single_market(id)
```

Arguments

id	Numerical code pertaining to the market. You can find a market's numerical code by consulting its URL or by first calling the all markets API.

Value

A tibble containing bid and ask data for a specific 'PredictIt' market.

Examples

```
## Only run this example in interactive R sessions
if (interactive()) {
   markets <- all_markets()
   id <- markets$id[1]
   single_market(id)
}
```

tweet_markets *Get bids and asks for all 'PredictIt' tweet markets*

Description

Wrapper function to get all available 'PredictIt' tweet count markets and contract prices.

Usage

```
tweet_markets()
```

Value

A tibble containing bid and ask data for all tweet count markets.

Examples

```
## Only run this example in interactive R sessions
if (interactive()) {
    tweet_markets()
}
```

Index