Package 'crypto'

September 20, 2018

Type Package

Title Cryptocurrency Market Data

Description Retrieves crypto currency current and historical information as well as information on the exchanges they are listed on. For current and historical it will retrieve the daily open, high, low and close values for all crypto currencies. This retrieves the historical market data by web scraping tables provided by 'Cryptocurrency Market Capitalizations' <https://coinmarketcap.com>.

Version 1.0.3

Date 2018-09-20

Maintainer Jesse Vent <cryptopackage@icloud.com>

URL https://github.com/JesseVent/crypto,
https://CRAN.R-project.org/package=crypto

BugReports https://github.com/JesseVent/crypto/issues

Depends R (>= 3.4.0), foreach, rvest, xml2

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports magrittr, tibble, jsonlite, dplyr, lubridate, xts, curl, utils, parallel, stats, doSNOW, tidyr, yaml

RoxygenNote 6.1.0

NeedsCompilation no

Author Jesse Vent [aut, cre]

Repository CRAN

Date/Publication 2018-09-20 17:10:18 UTC

R topics documented:

`crypto2xts` *crypto2xts*

Description

Converts the `getCoins()` dataframe into an xts object. Provide frequency to summarise into specific time periods.

Usage

```
crypto2xts(df, frequency = NULL)

crypto_xts(df, frequency = NULL)
```

Arguments

`df`	data.frame from `getCoins()`
`frequency`	string `?round_date` for help

Value

xts

Note

Each value in `frequency <- c('second', 'minute','hour', 'day', 'week', 'month', 'year')` can have an integer in front of it to retrieve the expressed time period. i.e. 3month

Examples

```
 ## Not run:
You can lookup additional frequencies at \code{?round_date}
from the lubridate package.
   crypto2xts(df, '.5s')
   crypto2xts(df, 'sec')
   crypto2xts(df, 'second')
   crypto2xts(df, 'minute')
```

```
    crypto2xts(df, '5 mins')
    crypto2xts(df, 'hour')
    crypto2xts(df, '2 hours')
    crypto2xts(df, 'day')
    crypto2xts(df, 'week')
    crypto2xts(df, 'month')
    crypto2xts(df, 'bimonth')
    crypto2xts(df, '3 months')
    crypto2xts(df, 'halfyear')
    crypto2xts(df, 'year')

## End(Not run)
```

`daily_market` *Daily cryptocurrency market data*

Description

Retrieve timeseries of market_cap, price_btc, price_usd and volume of specified coin - perfect for charting or timeseries analysis.

Usage

```
daily_market(coin = NULL)
```

Arguments

`coin`	Name, symbol or slug of crypto currency

Details

Most tokens are refreshed every 6 hours.. results may vary.

Value

Daily timeseries of token data in a dataframe:

`timestamp`	Timestamp (POSIXct)
`market_cap`	Market Cap in USD
`price_btc`	Price in BTC
`price_usd`	Price in USD
`volume`	Volume traded in USD
`slug`	Coin URL slug (unique)

Examples

```
## Not run:
coin       <- "kin"
kin_charts <- daily_market(coin)

## End(Not run)
```

`getCoins` *Get historic crypto currency market data*

Description

Scrape the crypto currency historic market tables from CoinMarketCap <https://coinmarketcap.com> and display the results in a date frame. This can be used to conduct analysis on the crypto financial markets or to attempt to predict future market movements or trends.

Usage

```
getCoins(coin = NULL, limit = NULL, cpu_cores = NULL,
  start_date = NULL, end_date = NULL)

crypto_history(coin = NULL, limit = NULL, cpu_cores = NULL,
  start_date = NULL, end_date = NULL)
```

Arguments

`coin`	string Name, symbol or slug of crypto currency, default is all tokens
`limit`	integer Return the top n records, default is all tokens
`cpu_cores`	integer Uses n cores for processing, default uses all cores
`start_date`	string Start date to retrieve data from, format 'yyyymmdd'
`end_date`	string End date to retrieve data from, format 'yyyymmdd'
`...`	No arguments, return all coins

Value

Crypto currency historic OHLC market data in a dataframe:

`slug`	Coin url slug
`symbol`	Coin symbol
`name`	Coin name
`date`	Market date
`ranknow`	Current Rank
`open`	Market open
`high`	Market high

low	Market low
close	Market close
volume	Volume 24 hours
market	USD Market cap
close_ratio	Close rate, min-maxed with the high and low values that day
spread	Volatility premium, high minus low for that day

This is the main function of the crypto package. If you want to retrieve ALL coins then do not pass a argument to getCoins(), or pass the coin name.

Please note that the doSNOW package is required to load the progress bar on both linux and macOS systems as the doParallel package does not support it.

Note

If experiencing issues, explicitly set cpu_cores=1 to turn off parallel processing.

Examples

```
# retrieving market history for specific crypto currency

coin <- "kin"
kin_coins <- listCoins(coin)

## Not run:

# retrieving market history for ALL crypto currencies

all_coins <- getCoins()

# retrieving this years market history for ALL crypto currencies

all_coins <- getCoins(start_date = '20180101')

## End(Not run)
```

---

getExchanges *Get current crypto market exchanges*

Description

Scrape the crypto currency exchange tables from CoinMarketCap <https://coinmarketcap.com> and display the results in a date frame. This can be used to conduct analysis on the exchanges or to attempt to predict exchange arbiture.

Usage

```
getExchanges(coin = NULL, limit = NULL, cpu_cores = NULL,
  start_date = NULL, end_date = NULL)

crypto_exchanges(coin = NULL, limit = NULL, cpu_cores = NULL,
  start_date = NULL, end_date = NULL)
```

Arguments

coin	string Name, symbol or slug of crypto currency, default is all tokens
limit	integer Return the top n records, default is all tokens
cpu_cores	integer Uses n cores for processing, default uses all cores
start_date	string Start date to retrieve data from, format 'yyyymmdd'
end_date	string End date to retrieve data from, format 'yyyymmdd'
...	No arguments, return all coins

Value

Crypto currency historic OHLC market data in a dataframe:

slug	Coin url slug
symbol	Coin symbol
name	Coin name
trading_pair	Coin trading pair
exchange_name	Name of exchange
last_updated	Exchange refresh
exchange_volume	Exchange $USD volume
exchange_price	Exchange $USD price
exchange_share	Percent exchange traded
coin_rank	Rank of current coin
exchange_rank	Exchange ranking for coin

If you want to retrieve ALL coins and their exchanges, then do not pass a argument to getExchanges(),

Please note that the doSNOW package is required to load the progress bar on both linux and macOS systems as the doParallel package does not support it.

Examples

```
## Not run:
# Retrieving exchange data for specific crypto currency

coin <- "kin"
kin_exchanges <- getExchanges(coin)
```

```
# retrieving market history for ALL crypto currencies

all_exchanges <- getExchanges()


## End(Not run)
```

`getPrices` *Get current crypto currency prices*

Description

This will retrieve the current market prices from CoinMarketCap. Data gets refreshed every 5 minutes.

Usage

```
getPrices(coin = NULL, limit = 0, currency = NULL)

crypto_prices(coin = NULL, limit = 0, currency = NULL)
```

Arguments

`coin`	Token name, default is all, Default: NULL
`limit`	Return top n coins, default is all, Default: 0
`currency`	Convert into local currency. Must be one of "AUD", "BRL", "CAD", "CHF", "CLP", "CNY", "CZK", "DKK", "EUR", "GBP", "HKD", "HUF", "IDR", "ILS", "INR", "JPY", "KRW", "MXN", "MYR", "NOK", "NZD", "PHP", "PKR", "PLN", "RUB", "SEK", "SGD", "THB", "TRY", "TWD", "ZAR", Default: NULL

Details

Updated every 5 minutes

Value

Will provide data frame of current prices

Examples

```
{
kin_price <- getPrices("kin")
}
```

global_market *Global cryptocurrency market data*

Description

Retrieve daily snapshot of market_cap and the volume traded for either total cryptocurrency market or the altcoin market only. Selecting 'total' will include bitcoin and all altcoins.

Usage

```
global_market(market = NULL)
```

Arguments

market	Either 'total' or 'altcoin'

Value

Daily timeseries of token data in a dataframe:

timestamp	Timestamp (POSIXct)
market_cap	Market Cap in USD
volume	Volume traded in USD

Examples

```
market         <- "total"
global_markets <- global_market(market)
```

listCoins *Retrieves name, symbol, slug and rank for all tokens*

Description

List all of the crypto currencies that have existed on CoinMarketCap and use this to populate the URL base for scraping historical market data. It retrieves name, slug, symbol and rank of crypto currencies from CoinMarketCap and creates URLS for `scraper()` to use.

Usage

```
listCoins(coin = NULL, start_date = NULL, end_date = NULL)

crypto_list(coin = NULL, start_date = NULL, end_date = NULL)
```

Arguments

coin	Name, symbol or slug of crypto currency
start_date	Start date to retrieve data from, format yyyymmdd
end_date	Start date to retrieve data from, format yyyymmdd
...	No arguments, return all coins

Value

Crypto currency historic OHLC market data in a dataframe:

symbol	Coin symbol (not-unique)
name	Coin name
slug	Coin URL slug (unique)
rank	Current rank by market cap
exchange_url	Exchange market tables urls for scraping
history_url	Historical market tables urls for scraping

Required dependency that is used in function call `getCoins()`.

Examples

```
# return specific coin

coin <- "kin"
coins <- listCoins(coin)

## Not run:

# return all coins
coin_list <- listCoins()

## End(Not run)
```

---

repair_dependencies *Installs all dependant packages*

Description

Helper function to install dependencies

Usage

```
repair_dependencies()
```

Value

dependant packages

Examples

```
## Not run:
# Fix dependency issues by reinstalling all
fix_crypto <- repair_dependencies()

## End(Not run)
```

`replace_encoding` *Check locale encoding*

Description

Helper function to ensure encoding is UTF-8

Usage

```
replace_encoding(sys_locale)
```

Arguments

`sys_locale`	string system locale

Value

Sets system variable to UTF-8

Examples

```
sys_locale <- Sys.getlocale(category = "LC_TIME")
replace_encoding(sys_locale)
```

reset_encoding *Reset locale encoding*

Description

Helper function to reset encoding from UTF-8 back to R sessions original locale value.

Usage

```
reset_encoding(sys_locale)
```

Arguments

sys_locale	string Original system locale

Value

Sets locale back to original value

Examples

```
sys_locale <- Sys.getlocale(category = "LC_TIME")
reset_encoding(sys_locale)
```

scraper *Historical table scraper*

Description

This web scrapes the historic price tables from CoinMarketCap and provides back a dataframe for the coin provided as an input. This function is a dependency of getCoins and is used as part of a loop to retrieve all crypto currencies.

Usage

```
scraper(attributes, slug)
```

Arguments

attributes	URL generated from `listCoins()`
slug	Unique identifier required for merging

Value

Raw OHLC market data in a dataframe:

`slug`	Coin url slug
`symbol`	Coin symbol
`name`	Coin name
`date`	Market date
`open`	Market open
`high`	Market high
`low`	Market low
`close`	Market close
`volume`	Volume 24 hours
`market`	USD Market cap

This function is not to be called individually by a user but is to be consumed as part of the getCoins.

Examples

```
## Not run:
# Only to be executed by getCoins
scraper(attributes)

## End(Not run)
```

Index