

Package ‘crypto2’

July 18, 2022

Title Download Crypto Currency Data from 'CoinMarketCap' without 'API'

Version 1.4.4

Date 2022-07-18

Maintainer Sebastian Stoeckl <sebastian.stoeckl@uni.li>

URL <https://github.com/sstoeckl/crypto2>

Description Retrieves crypto currency information and historical prices as well as information on the exchanges they are listed on. Historical data contains daily open, high, low and close values for all crypto currencies. All data is scraped from <<https://coinmarketcap.com>> via their 'web-api'.

License MIT + file LICENSE

Depends R (>= 4.0.0)

Imports dplyr, tibble, tidyr, purrr, progress, stats, lubridate,
jsonlite, cli, plyr

Encoding UTF-8

RoxygenNote 7.1.2

Suggests spelling

Language en-US

NeedsCompilation no

Author Sebastian Stoeckl [aut, cre] (<<https://orcid.org/0000-0002-4196-6093>>,
Package commissioner and maintainer.),
Jesse Vent [ctb] (Creator of the crypto package that provided the idea
and basis for this package.)

Repository CRAN

Date/Publication 2022-07-18 13:20:05 UTC

R topics documented:

crypto_history	2
crypto_info	4
crypto_list	5

crypto_listings	7
exchange_info	9
exchange_list	11

Index	12
--------------	-----------

crypto_history	<i>Get historic crypto currency market data</i>
----------------	---

Description

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

Usage

```
crypto_history(
  coin_list = NULL,
  convert = "USD",
  limit = NULL,
  start_date = NULL,
  end_date = NULL,
  interval = NULL,
  sleep = 60,
  finalWait = TRUE
)
```

Arguments

coin_list	string if NULL retrieve all currently existing coins (crypto_list()), or provide list of crypto currencies in the crypto_list() format (e.g. current and/or dead coins since 2015)
convert	(default: USD) to one or more of available fiat or precious metals prices (fiat_list()). If more than one are selected please separate by comma (e.g. "USD,BTC")
limit	integer Return the top n records, default is all tokens
start_date	string Start date to retrieve data from, format 'yyyymmdd'
end_date	string End date to retrieve data from, format 'yyyymmdd', if not provided, today will be assumed
interval	string Interval with which to sample data according to what seq() needs
sleep	integer (default 60) Seconds to sleep between API requests
finalWait	to avoid calling the web-api again with another command before 60s are over (TRUE=default)

Value

Crypto currency historic OHLC market data in a dataframe and additional information via attribute "info":

timestamp	Timestamp of entry in database
slug	Coin url slug
id	Coin market cap unique id
name	Coin name
symbol	Coin symbol
ref_cur	Reference currency
open	Market open
high	Market high
low	Market low
close	Market close
volume	Volume 24 hours
market_cap	Market cap - close x circulating supply
time_open	Timestamp of open
time_close	Timestamp of close
time_high	Timestamp of high
time_low	Timestamp of low

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

Examples

```
## Not run:

# Retrieving market history for ALL crypto currencies
all_coins <- crypto_history(limit = 2)
one_coin <- crypto_history(limit = 1)

# Retrieving market history since 2020 for ALL crypto currencies
all_coins <- crypto_history(start_date = '20200101', limit=10)

# Retrieve 2015 history for all 2015 crypto currencies
coin_list_2015 <- crypto_list(only_active=TRUE) %>%
  dplyr::filter(first_historical_data<="2015-12-31",
               last_historical_data>="2015-01-01")
coins_2015 <- crypto_history(coin_list = coin_list_2015,
start_date = "20150101", end_date="20151231", limit=20, interval="90d")

## End(Not run)
```

crypto_info	<i>Retrieves info (urls, logo, description, tags, platform, date_added, notice, status) on CMC for given id or slug</i>
-------------	---

Description

This code uses the web api. It retrieves data for all active, delisted and untracked coins! It does not require an 'API' key.

Usage

```
crypto_info(
  coin_list = NULL,
  limit = NULL,
  requestLimit = 300,
  sleep = 60,
  finalWait = TRUE
)
```

Arguments

coin_list	string if NULL retrieve all currently active coins (crypto_list()), or provide list of cryptocurrencies in the crypto_list() format (e.g. current and/or dead coins since 2015)
limit	integer Return the top n records, default is all tokens
requestLimit	limiting the length of request URLs when bundling the api calls
sleep	integer (default 60) Seconds to sleep between API requests
finalWait	to avoid calling the web-api again with another command before 60s are over (TRUE=default)

Value

List of (active and historically existing) cryptocurrencies in a tibble:

id	CMC id (unique identifier)
name	Coin name
symbol	Coin symbol (not-unique)
slug	Coin URL slug (unique)
category	Coin category: "token" or "coin"
description	Coin description according to CMC
logo	CMC url of CC logo
subreddit	Name of subreddit community
notice	Markdown formatted notices from CMC

date_added	Date CC was added to the CMC database
twitter_username	Username of CCs twitter account
is_hidden	TBD
date_launched	Date CC was launched
self_reported_circulating_supply	Self reported circulating supply
self reported tags	Self_reported_tags
status	timestamp and other status messages
tags	Tibble of tags and tag categories
url	Tibble of various resource urls. Gives website, technical_doc (whitepaper), source_code, message_board, chat, announcement, reddit, twitter, (block) explorer urls
Platform	Metadata about the parent coin if available. Gives id, name, symbol, slug, and token address according to CMC

Examples

```
## Not run:
# return info for bitcoin
coin_info <- crypto_info(limit=3)

## End(Not run)
```

crypto_list	<i>Retrieves name, CMC id, symbol, slug, rank, an activity flag as well as activity dates on CMC for all coins</i>
-------------	--

Description

This code uses the web api. It retrieves data for all historic and all active coins and does not require an 'API' key.

This code uses the web api. It retrieves data for all available fiat currencies and does not require an API key.

Usage

```
crypto_list(only_active = TRUE, add_untracked = FALSE)
```

```
fiat_list(include_metals = FALSE)
```

Arguments

only_active Shall the code only retrieve active coins (TRUE=default) or include inactive coins (FALSE)

add_untracked Shall the code additionally retrieve untracked coins (FALSE=default)

include_metals (default FALSE) Shall the code also retrieve available precious metals?

Value

List of (active and historically existing) cryptocurrencies in a tibble:

id	CMC id (unique identifier)
name	Coin name
symbol	Coin symbol (not-unique)
slug	Coin URL slug (unique)
rank	Current rank on CMC (if still active)
is_active	Flag showing whether coin is active (1), inactive(0) or untracked (-1)
first_historical_data	First time listed on CMC
last_historical_data	Last time listed on CMC, <i>today's date</i> if still listed

List of (active and historically existing) cryptocurrencies in a tibble:

id	CMC id (unique identifier)
symbol	Coin symbol (not-unique)
name	Coin name
sign	Fiat currency sign
code	Precious metals code

Examples

```
## Not run:
# return all coins
active_list <- crypto_list(only_active=TRUE)
all_but_untracked_list <- crypto_list(only_active=FALSE)
full_list <- crypto_list(only_active=FALSE,add_untracked=TRUE)

# return all coins active in 2015
coin_list_2015 <- active_list %>%
  dplyr::filter(first_historical_data<="2015-12-31",
                last_historical_data>="2015-01-01")

## End(Not run)

## Not run:
# return all fiat currencies available on CMC as well as precious metals
```

```
fiat_list <- fiat_list(include_metals=TRUE)

## End(Not run)
```

crypto_listings	<i>Retrieves name, CMC id, symbol, slug, rank, an activity flag as well as activity dates on CMC for all coins</i>
-----------------	--

Description

This code uses the web api. It retrieves listing data (latest/new/historic) and does not require an 'API' key.

Usage

```
crypto_listings(
  which = "latest",
  convert = "USD",
  limit = 5000,
  start_date = NULL,
  end_date = NULL,
  interval = "day",
  quote = FALSE,
  sleep = 60,
  finalWait = TRUE
)
```

Arguments

which	string Shall the code retrieve the latest listing, the new listings or a historic listing?
convert	string (default: USD) to one or more of available fiat or precious metals prices (fiat_list()). If more than one are selected please separate by comma (e.g. "USD,BTC"), only necessary if 'quote=TRUE'
limit	integer Return the top n records, default is 5000 (max allowed by CMC)
start_date	string Start date to retrieve data from, format 'yyyymmdd'
end_date	string End date to retrieve data from, format 'yyyymmdd', if not provided, today will be assumed
interval	string Interval with which to sample data according to what seq() needs
quote	logical set to TRUE if you want to include price data (FALSE=default)
sleep	integer (default 60) Seconds to sleep between API requests
finalWait	to avoid calling the web-api again with another command before 60s are over (TRUE=default)

Value

List of latest/new/historic listings of cryptocurrencies in a tibble (depending on the "which"-switch and whether "quote" is requested, the result may only contain some of the following variables):

id	CMC id (unique identifier)
name	Coin name
symbol	Coin symbol (not-unique)
slug	Coin URL slug (unique)
date_added	Date when the coin was added to the dataset
last_updated	Last update of the data in the database
rank	Current rank on CMC (if still active)
market_cap	market cap - close x circulating supply
market_cap_by_total_supply	market cap - close x total supply
market_cap_dominance	market cap dominance
fully_diluted_market_cap	fully diluted market cap
self_reported_market_cap	is the source of the market cap self-reported
self_reported_circulating_supply	is the source of the circulating supply self-reported
tv1_ratio	percentage of total value locked
price	latest average price
circulating_supply	approx. number of coins in circulation
total_supply	approx. total amount of coins in existence right now (minus any coins that have been verifiably burned)
max_supply	CMC approx. of max amount of coins that will ever exist in the lifetime of the currency
num_market_pairs	number of market pairs across all exchanges this coin
tv1	total value locked
volume_24h	Volume 24 hours
volume_change_24h	Volume change in 24 hours
percent_change_1h	1 hour return
percent_change_24h	24 hour return
percent_change_7d	7 day return

Examples

```
## Not run:
# return new listings from the last 30 days
new_listings <- crypto_listings(which="new", quote=FALSE)
new_listings2 <- crypto_listings(which="new", quote=TRUE, convert="BTC,USD")
# return latest listing (last available data of all CC including quotes)
latest_listings <- crypto_listings(which="latest", quote=TRUE)

# return all listings in the first week of January 2014
listings_2014w1 <- crypto_listings(which="historical", quote=TRUE,
start_date = "20140101", end_date="20140107", interval="day")

# report in two different currencies
listings_2014w1_USDBTC <- crypto_listings(which="historical", quote=TRUE,
start_date = "20140101", end_date="20140107", interval="day", convert="USD,BTC")

## End(Not run)
```

exchange_info	<i>Retrieves info (urls,logo,description,tags,platform,date_added,notice,status) on CMC for given exchange slug</i>
---------------	---

Description

This code uses the web api. It retrieves data for all active, delisted and untracked exchanges! It does not require an 'API' key.

Usage

```
exchange_info(
  exchange_list = NULL,
  limit = NULL,
  requestLimit = 300,
  sleep = 60,
  finalWait = TRUE
)
```

Arguments

exchange_list	string if NULL retrieve all currently active exchanges (exchange_list()), or provide list of exchanges in the exchange_list() format (e.g. current and/or delisted)
limit	integer Return the top n records, default is all exchanges
requestLimit	limiting the length of request URLs when bundling the api calls
sleep	integer (default 60) Seconds to sleep between API requests
finalWait	to avoid calling the web-api again with another command before 60s are over (TRUE=default)

Value

List of (active and historically existing) exchanges in a tibble:

id	CMC exchange id (unique identifier)
name	Exchange name
slug	Exchange URL slug (unique)
description	Exchange description according to CMC
notice	Exchange notice (markdown formatted) according to CMC
logo	CMC url of CC logo
type	Type of exchange
date_launched	Launch date of this exchange
is_hidden	TBD
is_redistributable	TBD
maker_fee	Exchanges maker fee
taker_fee	Exchanges maker fee
spot_volume_usd	Current volume in USD according to CMC
spot_volume_last_updated	Latest update of spot volume
status	timestamp and other status messages
tags	Tibble of tags and tag categories
url	Tibble of various resource urls. Gives website, blog, fee, twitter.
countries	Tibble of countries the exchange is active in
fiat	Tibble of fiat currencies the exchange trades in

Examples

```
## Not run:
# return info for the first three exchanges
exchange_info <- exchange_info(limit=3)

## End(Not run)
```

exchange_list	<i>Retrieves name, CMC id, symbol, slug, rank, an activity flag as well as activity dates on CMC for all coins</i>
---------------	--

Description

This code uses the web api. It retrieves data for all historic and all active coins and does not require an 'API' key.

Usage

```
exchange_list(only_active = TRUE, add_untracked = FALSE)
```

Arguments

`only_active` Shall the code only retrieve active coins (TRUE=default) or include inactive coins (FALSE)

`add_untracked` Shall the code additionally retrieve untracked coins (FALSE=default)

Value

List of (active and historically existing) cryptocurrencies in a tibble:

<code>id</code>	CMC exchange id (unique identifier)
<code>name</code>	Exchange name
<code>slug</code>	Exchange URL slug (unique)
<code>is_active</code>	Flag showing whether exchange is active (1), inactive(0) or untracked (-1)
<code>first_historical_data</code>	First time listed on CMC
<code>last_historical_data</code>	Last time listed on CMC, <i>today's date</i> if still listed

Examples

```
## Not run:
# return all coins
ex_active_list <- exchange_list(only_active=TRUE)
ex_all_but_untracked_list <- exchange_list(only_active=FALSE)
ex_full_list <- exchange_list(only_active=FALSE,add_untracked=TRUE)

## End(Not run)
```

Index

crypto_history, 2
crypto_info, 4
crypto_list, 5
crypto_listings, 7

exchange_info, 9
exchange_list, 11

fiat_list (crypto_list), 5