Package ‘crypto2’

October 19, 2022

Title Download Crypto Currency Data from ‘CoinMarketCap’ without ‘API’
Version 1.4.5
Date 2022-10-12
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 7.2.1
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 2022-10-19 20:55:05 UTC

R topics documented:

- crypto_global_quotes ........................................... 2
- crypto_history .................................................... 4
- crypto_info ....................................................... 6
crypto_global_quotes

Retrieves historical quotes for the global aggregate market

Description

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

Usage

crypto_global_quotes(
    which = "latest",
    convert = "USD",
    start_date = NULL,
    end_date = NULL,
    interval = "daily",
    quote = FALSE,
    finalWait = FALSE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>which</td>
<td>string Shall the code retrieve the latest listing or a historic listing?</td>
</tr>
<tr>
<td>convert</td>
<td>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. &quot;USD,BTC&quot;), only necessary if 'quote=TRUE'</td>
</tr>
<tr>
<td>start_date</td>
<td>string Start date to retrieve data from, format 'yyyymmdd'</td>
</tr>
<tr>
<td>end_date</td>
<td>string End date to retrieve data from, format 'yyyymmdd', if not provided, today will be assumed</td>
</tr>
<tr>
<td>interval</td>
<td>string Interval with which to sample data, default 'daily'. Must be one of &quot;hourly&quot; &quot;daily&quot; &quot;weekly&quot; &quot;monthly&quot; &quot;weekly&quot; &quot;monthly&quot; &quot;yearly&quot;</td>
</tr>
<tr>
<td>quote</td>
<td>logical set to TRUE if you want to include price data (FALSE=default)</td>
</tr>
<tr>
<td>finalWait</td>
<td>to avoid calling the web-api again with another command before 60s are over (TRUE=default)</td>
</tr>
</tbody>
</table>
Value

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

- **btc_dominance**
  number Bitcoin’s market dominance percentage by market cap.

- **eth_dominance**
  number Ethereum’s market dominance percentage by market cap.

- **active_cryptocurrencies**
  number Count of active crypto currencies tracked by CMC This includes all crypto currencies with a listing_status of "active" or "listed".

- **total_cryptocurrencies**
  number Count of all crypto currencies tracked by CMC This includes "inactive" listing_status crypto currencies.

- **active_market_pairs**
  number Count of active market pairs tracked by CoinMarketCap across all exchanges.

- **active_exchanges**
  number Count of active exchanges tracked by CMC This includes all exchanges with a listing_status of "active" or "listed".

- **total_exchanges**
  number Count of all exchanges tracked by CMC This includes "inactive" listing_status exchanges.

- **last_updated**
  Timestamp of when this record was last updated.

- **total_market_cap**
  number The sum of all individual cryptocurrency market capitalizations in the requested currency.

- **total_volume_24h**
  number The sum of rolling 24 hour adjusted volume (as outlined in our methodology) for all crypto currencies in the requested currency.

- **total_volume_24h_reported**
  number The sum of rolling 24 hour reported volume for all crypto currencies in the requested currency.

- **altcoin_volume_24h**
  number The sum of rolling 24 hour adjusted volume (as outlined in our methodology) for all crypto currencies excluding Bitcoin in the requested currency.

- **altcoin_volume_24h_reported**
  number The sum of rolling 24 hour reported volume for all crypto currencies excluding Bitcoin in the requested currency.

- **altcoin_market_cap**
  number The sum of all individual cryptocurrency market capitalizations excluding Bitcoin in the requested currency.
Examples

```r
## Not run:
# return new listings from the last 30 days
new_quotes <- crypto_global_quotes(which="latest", quote=FALSE)
new_quotes2 <- crypto_global_quotes(which="latest", quote=TRUE, convert="BTC,USD")
# return all global quotes in the first week of January 2014
quotes_2014w1 <- crypto_global_quotes(which="historical", quote=TRUE,
start_date = "20140101", end_date="20140107", interval="daily")

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

## End(Not run)
```

---

### crypto_history

**Get historic crypto currency market data**

**Description**

Scrape the crypto currency historic market tables from `CoinMarketCap` [https://coinmarketcap.com](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**

```r
crypto_history(
    coin_list = NULL,
    convert = "USD",
    limit = NULL,
    start_date = NULL,
    end_date = NULL,
    interval = NULL,
    sleep = 0,
    finalWait = FALSE
)
```

**Arguments**

- **coin_list** string if NULL retrieve all currently existing coins (crypto_list()), or provide list of crypto currencies in the crypto_list() or cryptoi_listings() 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 `yyyymmd`
crypto_history

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

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
```r
crypto_info(
  coin_list = NULL,
  limit = NULL,
  requestLimit = 300,
  sleep = 0,
  finalWait = FALSE
)
```

Arguments

- `coin_list` string if NULL retrieve all currently active coins (crypto_list()), or provide list of cryptocurrencies in the crypto_list() or cryptoi_listings() 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)
slugs: 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
selfreported_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

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

## End(Not run)
```

### crypto_list

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieves name, CMC id, symbol, slug, rank, an activity flag as well as activity dates on CMC for all coins</td>
</tr>
</tbody>
</table>

**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.
crypto_list

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, today's date 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")
crypto_listings

Retrieves name, CMC id, symbol, slug, rank, an activity flag as well as activity dates on CMC for all coins

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 = 0,
    finalWait = FALSE
)

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
- **tvl_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
- **tvl**: 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
exchange_info

Examples

```r
# 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**

Retrieves info (urls,logo,description,tags,platform,date_added,notice,status) on CMC for given exchange slug

**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**

```r
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:

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

### Examples

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

```r
## End(Not run)
```
Description

This code uses the web api. It retrieves data for all historic and all active exchanges 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 exchanges (TRUE=default) or include inactive coins (FALSE)
add_untracked Shall the code additionally retrieve untracked exchanges (FALSE=default)

Value

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

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

Examples

## Not run:
# return all exchanges
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_global_quotes, 2
crypto_history, 4
crypto_info, 6
crypto_list, 7
crypto_listings, 9

exchange_info, 11
exchange_list, 13

fiat_list (crypto_list), 7