

Package: alphavantagepf (via r-universe)

June 8, 2026

Title Comprehensive R Wrapper and Shiny Interface for 'Alphavantage Financial Data' API

Version 0.7.41

Description Download, manage, and visualize 'Alphavantage financial data' <<https://www.alphavantage.co/documentation/>> to reduced 'data.table' objects. API interface Includes support functions to extract and simplify returned complex data. A Shiny application is also provided to download, manage, and graph asset prices and characteristics.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.1.0)

Imports dplyr (>= 1.0.0), tidyr (>= 0.6.3), data.table, glue (>= 1.1.1), httr (>= 1.2.1), jsonlite (>= 1.5), purrr (>= 1.0), readr (>= 1.1.1), stringr (>= 1.2.0), timeDate (>= 4000.0), lubridate (>= 1.0.0), utils (>= 4.0.0), fst (>= 0.9.8), shiny (>= 1.0.0), shinyFeedback (>= 0.4.0), dygraphs (>= 1.1.0), gt (>= 1.3.0), gtExtras (>= 0.6.0), TTR (>= 0.24.0), usethis, FinanceGraphs (>= 0.8.0)

Suggests bslib, testthat, knitr, rmarkdown

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

LazyData true

VignetteBuilder knitr

URL <https://github.com/derekholmes0/alphavantagepf>

BugReports <https://github.com/derekholmes0/alphavantagepf/issues>

Config/pak/sysreqs

libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev git make libgit2-dev libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev libssl-dev python3 libnode-dev libx11-dev zlib1g-dev

Repository <https://derekholmes0.r-universe.dev>

Date/Publication 2026-06-08 19:02:26 UTC

RemoteUrl <https://github.com/derekholmes0/alphavantagepf>

RemoteRef HEAD

RemoteSha a7495818bc406de3d20aab49203a2498f680c63d

Contents

alphavantagepf	2
av_add_data	3
av_api_key	4
av_extract_df	5
av_funhelp	6
av_get_pf	7
av_grep_opts	9
av_make_funcmap	11
av_opt_helper_cols	11
av_reset_defaults	13
av_runShiny	13
dump_the	15
gt.basetheme	16
message_if	17
Index	18

alphavantagepf	<i>alphavantagepf: A lightweight R interface to the Alpha Vantage API</i>
----------------	---

Description

An R interface to the Alpha Vantage API which emphasizes normalized data and data.table conventions

Details

The alphavantagepf package provides a lightweight interface to the Alpha Vantage API. Alpha Vantage is a free source for financial data that in many cases is more accurate than Yahoo Finance and Google Finance. Get a free API KEY at <https://www.alphavantage.co>. Then use the R interface functions `av_api_key("YOUR_KEY")` to set the API key and the `av_get_pf()` function to get financial data.

av_add_data	<i>App database functions</i>
-------------	-------------------------------

Description

Adds price data to [av_runShiny\(\)](#) internal data.

Adds asset lists to [av_runShiny\(\)](#) internal data.

Usage

```
av_add_data(indta, assettypes = NULL, delay = 0)
```

```
av_add_assetlist(indta)
```

Arguments

indta	A data.frame with two columns c("listnm", "ticker") with one or more lines for each "listnm"
assettypes	(default NULL) An optional data.frame with minimal columns c(symbol, type, currency, name) with descriptive data for the assets given in indta. If not specified, a call to av_get_pf(. , "SYMBOL_SEARCH") is necessary to determine the asset type (one of c("Equity", "ETF", "FX", "Index")) for subsequent calls to av_get_pf()
delay	(default 0) Seconds to delay calls to determine asset type for future AV downloads. This is unused if assettypes is given.

Details

Entire set of columns from [av_get_pf\(\)](#) can be added. First date column renamed to timestamp

Lists are specified in normalized form. Duplicate list names with those currently in use are replaced.

Value

Nothing

Nothing

See Also

[av_runShiny\(\)](#)

Examples

```
## Not run:  
av_add_data(av_get_pf("IBM", "TIME_SERIES_DAILY_ADJUSTED"))
```

```
## End(Not run)
```

```
## Not run:
```

```
newtickers <- c("QQQ", "QQQE", "NDX")
av_add_assetlist(data.table(listnm=rep("nasdaq", length(newtickers)), ticker=newtickers))
# To remove an asset list, just use an empty string for the ticker
av_add_assetlist(data.table(listnm=c("new"), ticker=c("")))

## End(Not run)
```

av_api_key	<i>Set the Alpha Vantage API Key</i>
------------	--------------------------------------

Description

av_api_key() sets Alphavantage API key and entitlement code

Usage

```
av_api_key(api_key, entitlement = NULL)
```

Arguments

api_key	A character string with your Alpha Vantage API Key.
entitlement	A character string with your Alpha Vantage entitlement status. If not "delayed" or "realtime" entitlement not added to API string.

Details

The Alpha Vantage API key must be set prior to using `av_get_pf()`. You can obtain an API key at the [Alpha Vantage Website](#).

Value

Invisibly returns two item list with API key and entitlement string once set). Use print method to view.

See Also

[av_get_pf\(\)](#)

Examples

```
## Not run:
av_api_key("YOUR_API_KEY", entitlement="delayed")
av_get_pf("IBM", "TIME_SERIES_INTRADAY")

## End(Not run)
```

av_extract_df	<i>Extract data from Alpha Vantage returned data</i>
---------------	--

Description

av_extract_df() pulls out nested data.frames from mixed data returned by av_get_pf() av_extract_fx() returns a simplified FX quote in data.table form from av_get_pf() calls. av_extract_analytics() returns melted data.table from calls to av_get_pf("ANALYTICS_FIXED_WINDOW") or av_get_pf("ANALYTICS_SLIDING_WINDOW") av_extract_divs_or_splits() returns melted data.table from calls to av_get_pf("DIVIDENDS") or av_get_pf("SPLITS")

Usage

```
av_extract_df(indta, grepstring = "", melt = FALSE)
```

```
av_extract_fx(indta, outputform = "common", cols = "")
```

```
av_extract_analytics(indta, separate_vars = FALSE)
```

```
av_extract_divs_or_splits(indta)
```

Arguments

indta	A data.table as returned by av_get()
grepstring	select which variable (data item) to unnest in data.table returned from av_get_pf
melt	Return data in melted/normalized form
outputform	(default : common, av_extract_fx() only): Use common names from REALTIME_BULK_QUOTES
cols	(default : all columns: av_extract_fx() only): String or List of columns to return ‘
separate_vars	(default : FALSE) separate out multiple levels of variable names into new keys

Details

av_get_pf() frequently returns a nested data.table, or a structure with nested data.frames. These are utilities functions to extract, filter and summarize returned values.

Value

Extracted data.tables for nested data returned from av_get_pf(), If grepstring is not specified, first nested table is returned. av_extract_fx() returns a shortened data.table with FX quotes.

See Also

[av_get_pf\(\)](#), [av_grep_opts\(\)](#)

Examples

```
## Not run:
av_get_pf("", "MARKET_STATUS") |> av_extract_df()
av_get_pf("", "TOP_GAINERS_LOSERS") |> av_extract_df("top_losers")
av_get_pf("USD/BRL", "CURRENCY_EXCHANGE_RATE") |> av_extract_fx()
av_get_pf(c("ORCL", "IBM"), "ANALYTICS_FIXED_WINDOW") |> av_extract_analytics(separate_vars=TRUE)

## End(Not run)
```

av_funhelp

Extract data from Alpha Vantage returned data

Description

Extract data from Alpha Vantage returned data

Usage

```
av_funhelp(av_fun_grep = "", verbose = TRUE)
```

Arguments

av_fun_grep	A Alpha Vantage function name or portions of one. For a list of parameters, visit the Alpha Vantage API documentation .
verbose	(Default: TRUE) Prints the help string

Details

Returns defaults and parameter lists for Alphavantage functions

Value

Help Text and default parameters.

See Also

[av_get_pf\(\)](#)

Examples

```
av_funhelp("GLOBAL_QUOTE")
```

av_get_pf

*Get financial data from the Alpha Vantage API***Description**

Interface to alphavantage API.

Usage

```
av_get_pf(
  symbol,
  av_fun,
  symbolvarnm = "symbol",
  dfonerror = TRUE,
  melted = "default",
  delay = 0,
  verbose = FALSE,
  ...
)
```

Arguments

symbol	A character string of an appropriate stock, fund, or currency See parameter "symbol" in Alpha Vantage API documentation .
av_fun	A character string matching an appropriate Alpha Vantage "function". See parameter "function" in Alpha Vantage API documentation .
symbolvarnm	(default: symbol) Variable name which has the symbol requested. Set to a blank string if not wanted.
dfonerror	(default: TRUE) Return an empty data.table when any error occurs
melted	(default: "default") String specifying when to melt, "default" is chosen by the package, "TRUE always" always melt, "FALSE never" never melts
delay	(default: 0) Delay in seconds after API call, used to embed within large single-symbol calls.
verbose	(default: FALSE) Print debug information helpful for errors. Also copies full url to clipboard.
...	Additional parameters or overrides passed to the Alpha Vantage API. For a list of parameters, visit the Alpha Vantage API documentation .

Details

The av_fun argument replaces the API parameter “function” because function is a reserved name in R. All other arguments match the Alpha Vantage API parameters.

There is no need to specify the apikey, datatype, outputsize or entitlement parameters as arguments to av_get_pf(). Before using, you must set the API key using av_api_key("YOUR_API_KEY"). outputsize defaults to "full" unless overridden with "compact in ..."

Required parameters other than symbol must be passed as named arguments via

Optional parameters have defaults which can be obtained by calling `av_funhelp()` and overridden via

There is no need to specify the datatype parameter as an argument to `av_get_pf()`. The function will return a `data.table`.

ForEx "FROM/TO" symbol details. FOREX symbols in the `av_get_pf()` function are supplied in "FROM/TO" format, which are then parsed in the Alpha Vantage API into `from_currency` and `to_currency` API parameters. Usage example: `av_get_pf("USD/BRL", "FX_DAILY")`

Value

Returns a `data.table` with results dependent on the function called. Mixed data is returned as a melted `data.table`, possibly with nested `data.frames`. Time series are returned as `data.tables`.

See Also

`av_api_key()`, `av_extract_df()`, `av_extract_fx()`, `av_grep_opts()`, `av_funhelp()`

Examples

```
## Not run:
av_api_key("YOUR_API_KEY")
av_api_key("YOUR_API_KEY", "delayed") # if you have such access

# example code

# ---- 1.0 SINGLE NAME EQUITY SUMMARY INFORMATION AND SEARCH ----

av_get_pf("IBM", "OVERVIEW") |> str()
av_get_pf("EWZ", "ETF_PROFILE")
av_get_pf("EWZ", "ETF_PROFILE") |> av_extract_df("holdings")
av_get_pf("", "SYMBOL_SEARCH", keywords="COMMERCE")

# ---- 2.0 MARKET QUOTES ----

av_get_pf("IBM", "GLOBAL_QUOTE")
av_get_pf("USD/BRL", "CURRENCY_EXCHANGE_RATE") |> av_extract_fx()
av_get_pf(c("ORCL", "IBM", "EWZ", "ARGT"), "REALTIME_BULK_QUOTES", melt=FALSE)
# Note you need advanced permissioning for REALTIME_BULK_QUOTES

# ---- 3.0 SINGLE NAME HISTORICAL DATA ----

av_get_pf("IBM", "TIME_SERIES_DAILY")
av_get_pf("IBM", "TIME_SERIES_INTRADAY")

# ---- 4.0 MARKET PRICING DATA ----

av_get_pf("", "MARKET_STATUS") |> av_extract_df()
av_get_pf("", "TOP_GAINERS_LOSERS") |> av_extract_df("top_losers")
av_get_pf("", "TREASURY_YIELD", maturity='7year')
```

```

# ---- 4.0 SINGLE NAME NON-PRICING DATA ----

av_get_pf("IBM", "DIVIDENDS")
av_get_pf("IBM", "EARNINGS") |> av_extract_df("quarter", melt=TRUE)
av_get_pf("IBM", "NEWS_SENTIMENT") |> av_extract_df("feed")

av_get_pf("IBM", "EARNINGS_CALL_TRANSCRIPT", quarter="2024Q3") |> av_extract_df("transcript")
# Note that quarter is a required parameter, not specifying will throw an error

# ---- 5.0 SINGLE NAME OPTION PRICING DATA ----

av_get_pf("IBM", "HISTORICAL_OPTIONS") |> av_grep_opts("F,M,put", mindays=2)

# ---- 6.0 TECHNICAL INDICATORS ----

av_funhelp("SMA") # Shows parameters and defaults chosen by this package.
av_get_pf("IBM", "SMA", time_period=20)

# ---- 7.0 WINDOW ANALYTICS ----

av_get_pf(c("ORCL", "IBM", "EWZ", "ARGT"), "ANALYTICS_FIXED_WINDOW", verbose=TRUE) |>
  av_extract_analytics(separate_vars=TRUE)

## End(Not run)

```

av_grep_opts

Extract data from Alpha Vantage returned data

Description

Extract data from Alpha Vantage returned data

Usage

```

av_grep_opts(
  indta,
  grepstring = "F,M,C,otm",
  spot = NULL,
  mindays = 3,
  startdt = Sys.Date(),
  mindelta = 0.05,
  dropsymbol = FALSE
)

```

Arguments

indta A data.table as returned by [av_get_pf\(\)](#)

grepstring (default F,M,call,otm,act) Three to five item string to select specific maturities, option types, strike ranges, and open interest categories from option sets returned by `av_get_pf(. , "HISTORICAL_OPTIONS")`. Each item in the list is an abbreviated code for what to select. The items are not case sensitive and in order are

Item	Values	description
Expiration Limit	'[F B	
Maturity Type	'[M Q	
Option Type	'[C P	
Moneyness	'[otm itm	
Activity	'[act A]'	

spot (default NULL) Spot to be used to determine itm/otm, If null then it is inferred from most out of the money call or put Note: This parameter only applies if there is one symbol in `indta`. If there is more than one ticker in `indta` a column `spot` must be in `indta` to get correct results.

mindays (default 3). Minimum number of days to expiration to be passed through from `startdt`

startdt (default `Sys.Date()`). Date from which expirations will be considered.

mindelta (default 0.05) delta limit on both moneyness sides, i.e. pass only options with deltas in range `c(mindelta,1-mindelta)`

dropsymbol (default FALSE). Drop symbol from returned data table.

Details

`av_get_pf()` returns a large list of options. This function helps to narrow down the list by maturity and moneyness.

Value

A reduced set of options obtained from `av_get_pf()` using Alphavantage HISTORICAL_OPTIONS function.

See Also

`av_get_pf()`

Examples

```
## Not run:
av_get_pf("IBM", "HISTORICAL_OPTIONS") |> av_grep_opts("F,M,put", mindays=2)

## End(Not run)
```

av_make_funcmap *Extract data from Alpha Vantage returned data*

Description

Extract data from Alpha Vantage returned data

Usage

```
av_make_funcmap()
```

Details

Creates a data.table with API signatures and default values for Alphavantage API calls

Value

function map dat.table, also written to data directory

See Also

[av_get_pf\(\)](#)

Examples

```
## Not run:  
av_make_funcmap()  
  
## End(Not run)
```

av_opt_helper_cols *Add additional data to returned option sets*

Description

Add additional data to returned option sets

Usage

```
av_opt_helper_cols(indta, scaling = NULL, spot = NULL)
```

Arguments

indta	An option data.table as returned by av_get_pf()
scaling	(default NULL) Scaling factor for marks, last values, and greeks. Options are <ul style="list-style-type: none"> • NULL or "none" : No Scaling • "10contracts" : 10 contracts converted into market value • "10kMV" : 10,000 USD converted into equivalent market value • number : Any numeric value in thousands of USD into equivalent market value
spot	(default NULL) Spot to be used to determine itm/otm, If null then it is inferred from most out of the money call or put Note: This parameter only applies if there is one symbol in indta. If there is more than one ticker in indta a column spot must be in indta to get correct results.

Details

Adds columns including

Column	Definition
daysExp	Days to Expiration
bo_pct	Bid offer in percent of option mid
bid_size_poi	Bid size percent of open interest
ncak	Notional number of contracts

Value

An option data.table with extra columns helpful for further analysis

See Also

[av_grep_opts\(\)](#)

Examples

```
## Not run:
av_get_pf("IBM", "HISTORICAL_OPTIONS") |> av_grep_opts("F,M,put", mindays=2) |> av_opt_helper_cols()

## End(Not run)
```

av_reset_defaults	<i>Extract data from Alpha Vantage returned data</i>
-------------------	--

Description

Extract data from Alpha Vantage returned data

Usage

```
av_reset_defaults()
```

Details

Resets [av_runShiny\(\)](#) defaults to original (newly installed) state

Value

No return

See Also

[av_runShiny\(\)](#)

Examples

```
## Not run:  
av_reset_defaults()  
  
## End(Not run)
```

av_runShiny	<i>RShiny App</i>
-------------	-------------------

Description

`av_runShiny()` runs an interactive RShiny app to download, manage, and visualize data from the [Alpha Vantage](#) data service. The app treats equities, ETFs, indices, and FX equally, so users do not need to know specific API calls to integrate data. Downloaded data is cached to reduce API calls, all data can be saved for external use, and external data can be added to the app. Sets of securities can be easily added and managed.

Usage

```
av_runShiny()
```

Details

Invocation starts Shiny application. See vignette for full details. **On first use**, click on AVOPTS tab, fill in the following fields, and hit the "Set Opts" button.

- ++AV API Key++: API key obtained from [Alpha Vantage](#)
- ++Entitlement++ Entitlement status (either *delayed* or *realtime*) Other options that can be set are given below.

To run analysis, Enter a semicolon delimited set of securities (e.g. SPY;QQQ;DIA), select a runtime option from the adjacent dropdown, and press the 'RUN' button. Options include:

Option	Description
Inventory	Table of what asset lists and price data is available
LivePx	Live prices of all tickers
NameSearch	Search for tickers
PriceTS	Time Series of prices or Total Return indices
ActiveTS	Active returns of tickers in top row relative to first ticker in bottom row
HistVolTS	Historic vol and rolling correlations of tickers in top row.
DES,News	Descriptions and News for tickers in top row.
DivEarn	Dividends, earnings, and earnings estimates for tickers in top row.
OptSearch	Search for options for tickers in top row. (OPTS tab)

** Options that can be set**. click on AVOPTS tab, fill in the following fields, and hit the "Set Opts" button.

- **fgts colorset**: A named list of colors which can be set using the `FinanceGraphs::fg_update_aes()` function.
- **Regr Significance**: p-level to highlight significant regressions. (Used in the HistVolTS function above.)
- **Cache Directory**: A directory where internal App data can be stored (and added to). If not filled in, a temporary directory is created and used.

The app allows all downloaded data to be saved. If used, data is saved in one file, and is stored as a named (by API call function) list of `data.tables`. Options to control this behavior are

- **AV dump Directory**: A directory where individual API requests can be cached. Data is saved in one file as a named (by API call function) list of `data.tables`.
- **Capture AV Data**: Save Price data, non-price data, all or none.
- **Update or Cumulative**: If update is selected, data is keyed relevantly, so for example there will only be one set of dividend data stored. If cumulative then all API data (including a timestamp) is saved sequentially. The latter will lead to much larger files.
- **Other options**:
 - `CleanOnStart` : cleans out the cache file every time the app is started.
 - `SaveEveryAVCall` : Saves cache file after every call.
 - `SaveNow` : Save cache file when this is selected.

Value

Nothing

Examples

```
## Not run:  
av_runShiny()  
  
## End(Not run)
```

dump_the	<i>Extract internal state</i>
----------	-------------------------------

Description

Prints internal data state of `av_runShiny()` `dump_the(typegrep="*")` `dump_inv()` `dump_assetlist()` `dump_captured()`

Usage

```
dump_the(typegrep = "*")  
  
dump_inv()  
  
dump_assetlist(returngt = TRUE)  
  
dump_captured(todo = "byfunction")
```

Arguments

typegrep : Grep string for internal state parameters
returngt : Return GT table
todo : One of c("byfunction", "pxhist", any av function name)

Value

data.table with desired data.

See Also

[av_runShiny\(\)](#)

Examples

```
## Not run:
`dump_the()`
`dump_inv()`
`dump_assetlist(returngt=TRUE)`
`dump_captured(todo="byfunction")`

## End(Not run)
```

gt.basetheme

*App formatting functions***Description**

Adds default elements to `gt()` table

Usage

```
gt.basetheme(
  x,
  gtopts = "all",
  sizepct = 100,
  style = 4,
  digits = 2,
  seps = FALSE,
  na_format = "-",
  size = "",
  interactive = FALSE
)
```

Arguments

<code>x</code>	Input data or <code>gt</code> table
<code>gtopts</code>	Which elements to add
<code>sizepct</code>	(default: 70) How big to make tables
<code>style</code>	(default: 4) Style number (see <code>gt()</code>)
<code>digits</code>	(default: 2) Number formatting
<code>seps</code>	(default: FALSE) Number formatting thousands separators
<code>na_format</code>	(default: "-") What to show for NAs
<code>size</code>	(default: "") Size of text, either "" or "small"
<code>interactive</code>	(default: FALSE) Add interactive elements

message_if

Date Utilities

Description

Prints message if condition met

Usage

message_if(reallydothis, ...)

Arguments

reallydothis Condition to be met

... Additional items to be passed to message

Details

Other utilities internal to this package

Index

alphavantagepf, [2](#)
av_add_assetlist (av_add_data), [3](#)
av_add_data, [3](#)
av_api_key, [4](#)
av_api_key(), [8](#)
av_extract_analytics (av_extract_df), [5](#)
av_extract_df, [5](#)
av_extract_df(), [8](#)
av_extract_divs_or_splits
 (av_extract_df), [5](#)
av_extract_fx (av_extract_df), [5](#)
av_extract_fx(), [5](#), [8](#)
av_funhelp, [6](#)
av_funhelp(), [8](#)
av_get_pf, [7](#)
av_get_pf(), [3–6](#), [8–12](#)
av_grep_opts, [9](#)
av_grep_opts(), [5](#), [8](#), [12](#)
av_make_funcmap, [11](#)
av_opt_helper_cols, [11](#)
av_reset_defaults, [13](#)
av_runShiny, [13](#)
av_runShiny(), [3](#), [13](#), [15](#)

dump_assetlist (dump_the), [15](#)
dump_captured (dump_the), [15](#)
dump_inv (dump_the), [15](#)
dump_the, [15](#)

FinanceGraphs::fg_update_aes(), [14](#)

gt(), [16](#)
gt.basetheme, [16](#)

message_if, [17](#)