
quantmod Documentation

Release 0.1.3

Jack Luo

May 26, 2020

Contents

1	Quantmod Reference	3
1.1	Submodules	3
1.2	quantmod.auth module	3
1.3	quantmod.chart module	3
1.4	quantmod.core module	10
1.5	quantmod.datetools module	11
1.6	quantmod.factory module	12
1.7	quantmod.ta module	14
1.8	quantmod.tanolib module	17
1.9	quantmod.tools module	17
1.10	Module contents	18
1.10.1	Author	18
1.10.2	Credits	18
2	Indices and tables	19
	Python Module Index	21
	Index	23

A powerful financial charting library based on R's Quantmod.

Quantmod makes creating interactive financial charts easy and intuitive. Furthermore, Quantmod has over 50 technical indicators built-in, in addition to a variety of technical and quantitative financial tools.

Read the full documentation below:

1.1 Submodules

1.2 quantmod.auth module

Functions that manage configuration writing

Refactored from Plotly's 'auth.py'.

`quantmod.auth.check_file_permissions()`
Return True if write permissions, else return False.

`quantmod.auth.get_path()`
Get path of AUTH_DIR.

1.3 quantmod.chart module

Main Chart fonctionnality

Chart is a wrapper on top of DataFrame that adds fonctionnality and allows for easy plotting. Features include time series adjustment, volume adjustment, and plotting of OHLCV data with over 100 technical indicators.

class `quantmod.chart.Chart(df, src=None, ticker=None, start=None, end=None)`
Bases: `object`

Quantmod Chart based on Pandas DataFrame.

Chart is a wrapper on top of DataFrame that adds fonctionnality and allows for easy plotting. Features include time series adjustment, volume adjustment, and plotting of OHLCV data with over 100 technical indicators.

add_ADX (*timeperiod=14, type='line', color='secondary', **kwargs*)
Average Directional Movement Index.

add_ADXR (*timeperiod=14, type='line', color='secondary', **kwargs*)
Average Directional Movement Index Rating.

add_APO (*fastperiod=12, slowperiod=26, matype=0, type='line', color='secondary', **kwargs*)
Absolute Price Oscillator.

add_AROON (*timeperiod=14, types=['line', 'line'], colors=['increasing', 'decreasing'], **kwargs*)
Aroon indicators.

Note that the first argument of types and colors refers to Aroon up while the second argument refers to Aroon down.

add_AROONOSC (*timeperiod=14, type='area', color='secondary', **kwargs*)
Aroon Oscillator.

add_BBANDS (*timeperiod=20, nbdevup=2, nbdevdn=2, matype=0, types=['line_dashed_thin', 'line_dashed_thin'], colors=['tertiary', 'grey_strong'], **kwargs*)
Bollinger Bands.

Note that the first argument of types and colors refers to upper and lower bands while second argument refers to middle band. (Upper and lower are symmetrical arguments, hence only 2 needed.)

add_BOP (*type='histogram', color='tertiary', **kwargs*)
Balance of Power.

add_CCI (*timeperiod=14, type='line', color='secondary', **kwargs*)
Channel Commodity Index.

add_CMO (*timeperiod=14, type='line', color='secondary', **kwargs*)
Chande Momentum Indicator.

add_DEMA (*timeperiod=26, type='line', color='secondary', **kwargs*)
Double Exponential Moving Average.

add_DX (*timeperiod=14, type='line', color='secondary', **kwargs*)
Directional Movement Index.

add_EMA (*timeperiod=26, type='line', color='secondary', **kwargs*)
Exponential Moving Average.

add_HT_TRENDLINE (*type='line', color='secondary', **kwargs*)
Hilert Transform Instantaneous Trendline.

add_KAMA (*timeperiod=20, type='line', color='secondary', **kwargs*)
Kaufmann Adaptive Moving Average.

add_MA (*timeperiod=20, matype=0, type='line', color='secondary', **kwargs*)
Moving Average (customizable).

add_MACD (*fastperiod=12, slowperiod=26, signalperiod=9, types=['line', 'line', 'histogram'], colors=['primary', 'tertiary', 'fill'], **kwargs*)
Moving Average Convergence Divergence.

Note that the first argument of types and colors refers to MACD, the second argument refers to MACD signal line and the third argument refers to MACD histogram.

add_MACDEXT (*fastperiod=12, fastmatype=0, slowperiod=26, slowmatype=0, signalperiod=9, signalmatype=0, types=['line', 'line', 'histogram'], colors=['primary', 'tertiary', 'fill'], **kwargs*)
Moving Average Convergence Divergence with Controllable MA Type.

Note that the first argument of types and colors refers to MACD, the second argument refers to MACD signal line and the third argument refers to MACD histogram.

add_MAMA (*fastlimit=0.5, slowlimit=0.05, types=['line', 'line'], colors=['secondary', 'tertiary'], **kwargs*)
 MESA Adaptive Moving Average.

Note that the first argument of *types* and *colors* refers to MAMA while the second argument refers to FAMA.

add_MAVP (*periods, minperiod=2, maxperiod=30, matype=0, type='line', color='secondary', **kwargs*)
 Moving Average with Variable Period.

Parameters *periods* (*Series or array*) – Moving Average period over timeframe to analyze, as a 1-dimensional shape of same length as chart.

add_MFI (*timeperiod=14, type='line', color='secondary', **kwargs*)
 Money Flow Index.

add_MIDPOINT (*timeperiod=14, type='line', color='secondary', **kwargs*)
 Midpoint Price over Period.

add_MINUS_DI (*timeperiod=14, type='line', color='decreasing', **kwargs*)
 Minus Directional Indicator.

add_MINUS_DM (*timeperiod=14, type='line', color='decreasing', **kwargs*)
 Minus Directional Movement.

add_MOM (*timeperiod=10, type='line', color='secondary', **kwargs*)
 Momentum Indicator.

add_PLUS_DI (*timeperiod=14, type='line', color='increasing', **kwargs*)
 Plus Directional Indicator.

add_PLUS_DM (*timeperiod=14, type='line', color='increasing', **kwargs*)
 Plus Directional Movement.

add_PPO (*fastperiod=12, slowperiod=26, matype=0, type='line', color='secondary', **kwargs*)
 Percent Price Oscillator.

add_ROC (*timeperiod=10, type='line', color='tertiary', **kwargs*)
 Rate of Change.

add_ROCP (*timeperiod=10, type='line', color='tertiary', **kwargs*)
 Rate of Change (Percentage).

add_ROCR (*timeperiod=10, type='line', color='tertiary', **kwargs*)
 Rate of Change (Ratio).

add_ROCR100 (*timeperiod=10, type='line', color='tertiary', **kwargs*)
 Rate of Change (Ratio * 100).

add_RSI (*timeperiod=14, type='line', color='secondary', **kwargs*)
 Relative Strength Index.

add_SAR (*acceleration=0.02, maximum=0.2, type='scatter', color='tertiary', **kwargs*)
 Parabolic SAR.

add_SAREXT (*startvalue=0, offsetonreverse=0, accelerationinitlong=0.02, accelerationlong=0.02, accelerationmaxlong=0.2, accelerationinitshort=0.02, accelerationshort=0.02, accelerationmaxshort=0.2, type='scatter', color='tertiary', **kwargs*)
 Parabolic SAR Extended.

add_SMA (*timeperiod=20, type='line', color='secondary', **kwargs*)
 Simple Moving Average.

add_STOCH (*fastk_period=5, slowk_period=3, slowk_matype=0, slowd_period=3, slowd_matype=0, types=['line', 'line'], colors=['primary', 'tertiary'], **kwargs*)
Slow Stochastic Oscillator.

Note that the first argument of types and colors refers to Slow Stoch %K, while second argument refers to Slow Stoch %D (signal line of %K obtained by MA).

add_STOCHF (*fastk_period=5, fastd_period=3, fastd_matype=0, types=['line', 'line'], colors=['primary', 'tertiary'], **kwargs*)
Fast Stochastic Oscillator.

Note that the first argument of types and colors refers to Fast Stoch %K, while second argument refers to Fast Stoch %D (signal line of %K obtained by MA).

add_STOCHRSI (*timeperiod=14, fastk_period=5, fastd_period=3, fastd_matype=0, types=['line', 'line'], colors=['primary', 'tertiary'], **kwargs*)
Stochastic Relative Strength Index.

Note that the first argument of types and colors refers to StochRSI %K while second argument refers to StochRSI %D (signal line of %K obtained by MA).

add_T3 (*timeperiod=20, vfactor=0.7, type='line', color='secondary', **kwargs*)
T3 Exponential Moving Average.

add_TEMA (*timeperiod=26, type='line', color='secondary', **kwargs*)
Triple Moving Exponential Average.

add_TRIMA (*timeperiod=20, type='line', color='secondary', **kwargs*)
Triangular Moving Average.

add_TRIX (*timeperiod=15, type='area', color='secondary', **kwargs*)
1-day Rate of Change of Triple Smooth EMA.

add_ULTOSC (*timeperiod=14, timeperiod2=14, timeperiod3=28, type='line', color='secondary', **kwargs*)
Ultimate Oscillator.

add_WILLR (*timeperiod=14, type='line', color='secondary', **kwargs*)
Williams %R.

add_WMA (*timeperiod=20, type='line', color='secondary', **kwargs*)
Weighted Moving Average.

adjust (*inplace=False*)
Adjust OHLC data for splits, dividends, etc.

Requires an adjusted close column to adjust the rest of the OHLC bars.

Parameters **inplace** (*bool*) – Modifies Chart inplace (returns None) if True, else returns modified Chart by default.

adjust_volume (*inplace=False*)
Adjust volume data for splits, dividends, etc.

Requires a close and an adjusted close column to adjust volume.

Parameters **inplace** (*bool, default False*) – Modifies Chart inplace (returns None) if True, else returns modified Chart by default.

has_OHLC
Return True if Chart DataFrame has OHLC, False otherwise.

has_OHLCV
Return True if Chart DataFrame has OHLCV, False otherwise.

has_adjusted_close

Return True if Chart DataFrame has adjusted close, False otherwise.

has_adjusted_high

Return True if Chart DataFrame has adjusted high, False otherwise.

has_adjusted_low

Return True if Chart DataFrame has adjusted low, False otherwise.

has_adjusted_open

Return True if Chart DataFrame has adjusted open, False otherwise.

has_close

Return True if Chart DataFrame has close, False otherwise.

has_dividend

Return True if Chart DataFrame has dividend, False otherwise.

has_high

Return True if Chart DataFrame has high, False otherwise.

has_low

Return True if Chart DataFrame has low, False otherwise.

has_open

Return True if Chart DataFrame has open, False otherwise.

has_volume

Return True if Chart DataFrame has volume, False otherwise.

head (*n=252*)

Return first *n* elements of chart.

Parameters *n* (*int*, *default 252*) – Number of elements to keep, starting from first element.

iplot (*type=None*, *volume=None*, *log=None*, *theme=None*, *layout=None*, *title=None*, *subtitle=None*, *hovermode=None*, *legend=None*, *annotations=None*, *shapes=None*, *dimensions=None*, *width=None*, *height=None*, *margin=None*, *filename=None*, *online=None*, ***kwargs*)

Generate a Plotly chart from Chart specifications.

The iplot function returns an embedded chart suitable for Jupyter notebooks, while the plot function simply opens it in the browser.

Parameters

- **type** (*{'ohlc', 'candlestick'}*) – *'line', 'line_thin', 'line_thick', 'line_dashed', 'line_dashed_thin', 'line_dashed_thick', 'area', 'area_dashed', 'area_dashed_thin', 'area_dashed_thick', 'area_threshold', 'scatter'*

Determine the chart type of the main trace. For candlestick and OHLC bars Chart needs to have OHLC enabled.

- **volume** (*bool*) – Toggle the display of a volume subplot in chart. Default True.
- **log** (*bool*) – Toggle logarithmic y-axis. Default False.
- **theme** (*string*) – Quantmod theme.
- **layout** (*dict or Layout*) – Plotly layout dict or `graph_objs.Layout` object. Will override all other arguments if conflicting as user-inputted layout is updated last.
- **title** (*string*) – Chart title.
- **subtitle** (*bool*, *default True*) – Toggle the display of last price and/or volume in chart.

- **log** (*bool*) – Toggle logarithmic y-axis. Default False.
- **hovermode** (*{'x', 'y', 'closest', False}*) – Toggle how a tooltip appears on cursor hover.
- **legend** (*dict, Legend or bool, default True*) – True/False or Plotly legend dict / `graph_objs.Legend` object. If legend is bool, Quantmod will only toggle legend visibility.
- **annotations** (*list or Annotations*) – Plotly annotations list / `graph_objs.Annotations` object.
- **shapes** (*list or Shapes*) – Plotly shapes list or `graph_objs.Shapes` object.
- **dimensions** (*tuple*) – Dimensions 2-tuple in order (width, height). Disables `autosize=True`.
- **width** (*int*) – Width of chart. Default 1080 pixels. If used with height, disables `autosize=True` (Equivalent to using dimensions).
- **height** (*int*) – Height of chart. Default 720 pixels. If used with width, disables `autosize=True` (Equivalent to using dimensions).
- **margin** (*dict or tuple*) – Plotly margin dict or 4-tuple in order (l, r, b, t) or 5-tuple in order (l, r, b, t, margin). Tuple input added for Cufflinks compatibility.
- **filename** (*string, default datetime.now()*) – Filename of chart that will appear on plot.ly. By default, filename is set according to current system time.
- **online** (*bool, default False*) – If True, forces chart to be drawn online even if `qm.go_offline()` has been called.

Examples

```
ch = qm.Chart(df) ch.iplot(type='ohlc', dimensions=(2560,1440))
```

```
ch = qm.Chart(df) ch.add_BBANDS() ch.add_RSI(14) ch.iplot(type='candlestick', title='EQUITY')
```

```
plot (type=None, volume=None, theme=None, layout=None, title=None, subtitle=None, log=None,
      hovermode=None, legend=None, annotations=None, shapes=None, dimensions=None,
      width=None, height=None, margin=None, filename=None, online=None, **kwargs)
Generate a Plotly chart from Chart specifications.
```

Parameters

- **type** (*{'ohlc', 'candlestick'}*) – **'line', 'line_thin', 'line_thick', 'line_dashed', 'line_dashed_thin', 'line_dashed_thick', 'area', 'area_dashed', 'area_dashed_thin', 'area_dashed_thick', 'area_threshold', 'scatter'**
- Determine the chart type of the main trace. For candlestick and OHLC bars Chart needs to have OHLC enabled.
- **volume** (*bool*) – Toggle the display of a volume subplot in chart. Default True.
 - **log** (*bool*) – Toggle logarithmic y-axis. Default False.
 - **theme** (*string*) – Quantmod theme.
 - **layout** (*dict or Layout*) – Plotly layout dict or `graph_objs.Layout` object. Will override all other arguments if conflicting as user-inputted layout is updated last.
 - **title** (*string*) – Chart title.
 - **subtitle** (*bool, default True*) – Toggle the display of last price and/or volume in chart.
 - **log** (*bool*) – Toggle logarithmic y-axis. Default False.
 - **hovermode** (*{'x', 'y', 'closest', False}*) – Toggle how a tooltip appears on cursor hover.

- **legend** (*dict, Legend or bool, default True*) – True/False or Plotly legend dict / graph_objs.Legend object. If legend is bool, Quantmod will only toggle legend visibility.
- **annotations** (*list or Annotations*) – Plotly annotations list / graph_objs.Annotations object.
- **shapes** (*list or Shapes*) – Plotly shapes list or graph_objs.Shapes object.
- **dimensions** (*tuple*) – Dimensions 2-tuple in order (width, height). Disables autosize=True.
- **width** (*int*) – Width of chart. Default 1080 pixels. If used with height, disables autosize=True (Equivalent to using dimensions).
- **height** (*int*) – Height of chart. Default 720 pixels. If used with width, disables autosize=True (Equivalent to using dimensions).
- **margin** (*dict or tuple*) – Plotly margin dict or 4-tuple in order (l, r, b, t) or 5-tuple in order (l, r, b, t, margin). Tuple input added for Cufflinks compatibility.
- **filename** (*string, default datetime.now()*) – Filename of chart that will appear on plot.ly. By default, filename is set according to current system time.
- **online** (*bool, default False*) – If True, forces chart to be drawn online even if qm.go_offline() has been called.

Examples

```
ch = qm.Chart(df) ch.plot(type='ohlc', dimensions=(2560,1440))
```

```
ch = qm.Chart(df) ch.add_BBANDS() ch.add_RSI(14) ch.plot(type='candlestick', title='EQUITY')
```

shape

Return shape of Chart object.

tail (n=252)

Return last n elements of chart.

Parameters **n** (*int, default 252*) – Number of elements to keep, starting from last element.

to_figure (*type=None, volume=None, theme=None, layout=None, title=None, subtitle=None, log=None, hovermode=None, legend=None, annotations=None, shapes=None, dimensions=None, width=None, height=None, margin=None, **kwargs*)

Return Plotly figure (dict) that is used to generate the stock chart.

Parameters

- **type** (*{'ohlc', 'candlestick'}*) – **'line', 'line_thin', 'line_thick', 'line_dashed', 'line_dashed_thin', 'line_dashed_thick', 'area', 'area_dashed', 'area_dashed_thin', 'area_dashed_thick', 'area_threshold', 'scatter'**

Determine the chart type of the main trace. For candlestick and OHLC bars Chart needs to have OHLC enabled.

- **volume** (*bool*) – Toggle the display of a volume subplot in chart. Default True.
- **theme** (*string*) – Quantmod theme.
- **layout** (*dict or Layout*) – Plotly layout dict or graph_objs.Layout object. Will override all other arguments if conflicting as user-inputted layout is updated last.
- **title** (*string*) – Chart title.
- **subtitle** (*bool, default True*) – Toggle the display of last price and/or volume in chart.

- **log** (*bool*) – Toggle logarithmic y-axis. Default False.
- **hovermode** (*{'x', 'y', 'closest', False}*) – Toggle how a tooltip appears on cursor hover.
- **legend** (*dict, Legend or bool, default True*) – True/False or Plotly legend dict / graph_objs.Legend object. If legend is bool, Quantmod will only toggle legend visibility.
- **annotations** (*list or Annotations*) – Plotly annotations list / graph_objs.Annotations object.
- **shapes** (*list or Shapes*) – Plotly shapes list or graph_objs.Shapes object.
- **dimensions** (*tuple*) – Dimensions 2-tuple in order (width, height). Disables autosize=True in plotly.
- **width** (*int*) – Width of chart. Default 1080 pixels. If used with height, disables autosize=True (Equivalent to using dimensions).
- **height** (*int*) – Height of chart. Default 720 pixels. If used with width, disables autosize=True (Equivalent to using dimensions).
- **margin** (*dict or tuple*) – Plotly margin dict or 4-tuple in order (l, r, b, t) or 5-tuple in order (l, r, b, t, margin). Tuple input added for Cufflinks compatibility.

Examples

```
ch = qm.Chart(df) ch.to_figure(type='ohlc', dimensions=(2560,1440))
ch = qm.Chart(df) ch.add_BBANDS() ch.add_RSI(14) ch.to_figure(type='candlestick', title='EQUITY')
to_frame()
```

Return DataFrame representation of Chart, including all added technical indicators.

1.4 quantmod.core module

Core Quantmod functions

Contains some wrappers over 'chart.py' for those used to R's Quantmod style, along with other tools for financial data acquisition.

`quantmod.core.chart_series` (*chart, iplot=False, **kwargs*)
Wrapper over Chart.plot() and Chart.iplot().

Parameters

- **chart** (*Quantmod Chart*) – Quantmod Chart to plot with.
- **iplot** (*bool, default False*) – If True, plots chart for interactive display inside Jupyter notebooks instead of opening a browser HTML.

`quantmod.core.get_symbol` (*ticker, src='yahoo', start='01/01/2010', end=datetime.datetime(2020, 5, 26, 6, 37, 8, 357416), to_frame=False*)

Get symbols

Currently just a wrapper over pandas_datareader.data.DataReader.

Parameters

- **ticker** (*string or list*) – Stock ticker or list of stock tickers to fetch from.
- **src** (*string, default 'yahoo'*) – String specifying the data source to fetch data from.

- **start** (*string or datetime*) – Left boundary for date range, specified either as string or as a datetime object. Defaults to 1/1/2010.
- **end** (*string or datetime*) – Right boundary for date range, specified either as string or as a datetime object. Defaults to `datetime.today()`.
- **to_frame** (*bool, default False*) – If True, returns obtained symbols as a DataFrame instead of a Quantmod Chart.

Examples

```
# Get Apple stock df = get_symbols('AAPL')
# Get Apple stock from Google and return DataFrame df = get_symbols('AAPL', 'google', to_frame=True)
# Get the VIX from Fred df = get_symbols('VIX', 'fred')
```

Returns symbol

Return type DataFrame

1.5 quantmod.datetools module

Date and time functions

Refactored from Cufflinks' `'date_tools.py'` module. Credits to @jorgesantos.

`quantmod.datetools.date_to_int` (*date, strfmt='%Y%m%d'*)
Converts a datetime date into int.

Parameters

- **date** (*datetime*) – date in datetime format
- **strfmt** (*string*) – format in which the int date will be generated

Example

```
date_to_int(dt.date(2015,10,23),'%Y')
```

`quantmod.datetools.get_date_from_today` (*delta, strfmt='%Y%m%d'*)
Returns a string that represents a date n numbers of days from today.

Parameters

- **delta** (*int*) – number of days
- **strfmt** (*string*) – format in which the date will be represented

`quantmod.datetools.int_to_date` (*int_date*)
Converts an int format date into datetime.

Parameters `int_date` (*int*) – date in int format

Example

```
int_date(20151023)
```

```
quantmod.datetools.string_to_date(string_date, strfmt='%Y%m%d')
```

Converts a string format date into datetime.

Parameters

- **string_date** (*string*) – date in string format
- **strfmt** (*string*) – format in which the input date is represented

1.6 quantmod.factory module

High-level functions meant for user access

This includes various plotting and theming helpers. Module also contains all argument validity checks.

```
quantmod.factory.get_base_layout(figures)
```

Generate a layout with the union of multiple figures' layouts.

Parameters **figures** (*list*) – List of Plotly figures to get base layout from.

```
quantmod.factory.get_skeleton()
```

Return the base Quantmod skeleton.

```
quantmod.factory.get_source(source)
```

Return a Quantmod source (as a dict).

Parameters **source** (*string*) – Quantmod source.

```
quantmod.factory.get_sources()
```

Return the list of available sources, or none if there is a problem.

```
quantmod.factory.get_template(theme=None, layout=None, title=None, hovermode=None, legend=None, annotations=None, shapes=None, dimensions=None, width=None, height=None, margin=None, **kwargs)
```

Generate color, traces, additions and layout dicts.

Parameters

- **theme** (*string*) – Quantmod theme.
- **layout** (*dict or Layout*) – Plotly layout dict or graph_objs.Layout object. Will override all other arguments if conflicting as user-inputted layout is updated last.
- **title** (*string*) – Chart title.
- **hovermode** (*{'x', 'y', 'closest', False}*) – Toggle how a tooltip appears on cursor hover.
- **legend** (*dict, Legend or bool*) – True/False or Plotly legend dict / graph_objs.Legend object. If legend is bool, Quantmod will only toggle legend visibility.
- **annotations** (*list or Annotations*) – Plotly annotations list / graph_objs.Annotations object.
- **shapes** (*list or Shapes*) – Plotly shapes list or graph_objs.Shapes object.
- **dimensions** (*tuple*) – Dimensions 2-tuple in order (width, height). Disables autosize=True.
- **width** (*int*) – Width of chart. Default 1080 pixels. If used with height, disables autosize=True (Equivalent to using dimensions).

- **height** (*int*) – Height of chart. Default 720 pixels. If used with width, disables autosize=True (Equivalent to using dimensions).
- **margin** (*dict or tuple*) – Plotly margin dict or 4-tuple in order (l, r, b, t) or 5-tuple in order (l, r, b, t, margin). Tuple input added for Cufflinks compatibility.

`quantmod.factory.get_theme(theme)`

Return a Quantmod theme (as a dict).

Parameters *theme* (*string*) – Quantmod theme.

`quantmod.factory.get_themes()`

Return the list of available themes, or none if there is a problem.

`quantmod.factory.make_additions(base_additions, additions)`

Make trace configuration from theme/skeleton and theme/additions.

Recursively update base_theme with theme using custom tool in utils.

Parameters

- **base_additions** (*dict*) – Additions file containing primitives from ‘skeleton.py’.
- **additions** (*dict*) – Additions configuration from specified theme.

`quantmod.factory.make_colors(base_colors, colors)`

Make trace configuration from theme/skeleton and theme/colors.

Recursively update base_theme with theme using custom tool in utils.

Parameters

- **base_colors** (*dict*) – Additions file containing primitives from ‘skeleton.py’.
- **colors** (*dict*) – Additions configuration from specified theme.

`quantmod.factory.make_layout(base_layout, layout, custom_layout, title, hovermode, legend, annotations, shapes, dimensions, width, height, margin, **kwargs)`

Make layout configuration from theme/skeleton and theme/traces.

Recursively update base_theme with theme using custom tool in utils.

Parameters

- **base_traces** (*dict*) – Layout file containing primitives from ‘skeleton.py’.
- **layout** (*dict*) – Layout configuration from specified theme.
- **custom_layout** (*dict*) – Plotly layout dict or `graph_objs.Layout` object. Will override all other arguments if conflicting as user-inputted layout is updated last.
- **title** (*string*) – Chart title.
- **hovermode** (*{‘x’, ‘y’, ‘closest’, False}*) – Toggle how a tooltip appears on cursor hover.
- **legend** (*dict, Legend or bool*) – True/False or Plotly legend dict / `graph_objs.Legend` object. If legend is bool, Quantmod will only toggle legend visibility.
- **annotations** (*list*) – Plotly annotations list.
- **shapes** (*list or*) – Plotly shapes list.
- **dimensions** (*tuple*) – Dimensions 2-tuple in order (width, height). Disables autosize=True.
- **width** (*int*) – Width of chart. Default 1080 pixels. If used with height, disables autosize=True (Equivalent to using dimensions).

- **height** (*int*) – Height of chart. Default 720 pixels. If used with width, disables autosize=True (Equivalent to using dimensions).
- **margin** (*dict or tuple*) – Plotly margin dict or 4-tuple in order (l, r, b, t) or 5-tuple in order (l, r, b, t, margin). Tuple input added for Cufflinks compatibility.

`quantmod.factory.make_traces` (*base_traces, traces*)

Make trace configuration from theme/skeleton and theme/traces.

Recursively update base_theme with theme using custom tool in utils.

Parameters

- **base_traces** (*dict*) – Trace file containing primitives from ‘skeleton.py’.
- **traces** (*dict*) – Trace configuration from specified theme.

`quantmod.factory.strip_figure` (*figure*)

Strip a Plotly figure into multiple figures with a trace on each of them.

Parameters **figure** (*dict or Figure*) – Plotly figure to strip into multiple figures.

1.7 quantmod.ta module

Wrappers around Ta-Lib technical indicators

Python native indicators in ‘tanolib.py’ file.

`quantmod.ta.add_ADX` (*self, timeperiod=14, type='line', color='secondary', **kwargs*)

Average Directional Movement Index.

`quantmod.ta.add_ADXR` (*self, timeperiod=14, type='line', color='secondary', **kwargs*)

Average Directional Movement Index Rating.

`quantmod.ta.add_APO` (*self, fastperiod=12, slowperiod=26, matype=0, type='line', color='secondary', **kwargs*)

Absolute Price Oscillator.

`quantmod.ta.add_AROON` (*self, timeperiod=14, types=['line', 'line'], colors=['increasing', 'decreasing'], **kwargs*)

Aroon indicators.

Note that the first argument of types and colors refers to Aroon up while the second argument refers to Aroon down.

`quantmod.ta.add_AROONOSC` (*self, timeperiod=14, type='area', color='secondary', **kwargs*)

Aroon Oscillator.

`quantmod.ta.add_BBANDS` (*self, timeperiod=20, nbdevup=2, nbdevdn=2, matype=0, types=['line_dashed_thin', 'line_dashed_thin'], colors=['tertiary', 'grey_strong'], **kwargs*)

Bollinger Bands.

Note that the first argument of types and colors refers to upper and lower bands while second argument refers to middle band. (Upper and lower are symmetrical arguments, hence only 2 needed.)

`quantmod.ta.add_BOP` (*self, type='histogram', color='tertiary', **kwargs*)

Balance of Power.

`quantmod.ta.add_CCI` (*self, timeperiod=14, type='line', color='secondary', **kwargs*)

Channel Commodity Index.

`quantmod.ta.add_CMO` (*self*, *timeperiod*=14, *type*='line', *color*='secondary', ***kwargs*)
Chande Momentum Indicator.

`quantmod.ta.add_DEMA` (*self*, *timeperiod*=26, *type*='line', *color*='secondary', ***kwargs*)
Double Exponential Moving Average.

`quantmod.ta.add_DX` (*self*, *timeperiod*=14, *type*='line', *color*='secondary', ***kwargs*)
Directional Movement Index.

`quantmod.ta.add_EMA` (*self*, *timeperiod*=26, *type*='line', *color*='secondary', ***kwargs*)
Exponential Moving Average.

`quantmod.ta.add_HT_TRENDLINE` (*self*, *type*='line', *color*='secondary', ***kwargs*)
Hilert Transform Instantaneous Trendline.

`quantmod.ta.add_KAMA` (*self*, *timeperiod*=20, *type*='line', *color*='secondary', ***kwargs*)
Kaufmann Adaptive Moving Average.

`quantmod.ta.add_MA` (*self*, *timeperiod*=20, *matype*=0, *type*='line', *color*='secondary', ***kwargs*)
Moving Average (customizable).

`quantmod.ta.add_MACD` (*self*, *fastperiod*=12, *slowperiod*=26, *signalperiod*=9, *types*=['line', 'line', 'histogram'], *colors*=['primary', 'tertiary', 'fill'], ***kwargs*)
Moving Average Convergence Divergence.

Note that the first argument of *types* and *colors* refers to MACD, the second argument refers to MACD signal line and the third argument refers to MACD histogram.

`quantmod.ta.add_MACDEXT` (*self*, *fastperiod*=12, *fastmatype*=0, *slowperiod*=26, *slowmatype*=0, *signalperiod*=9, *signalmatype*=0, *types*=['line', 'line', 'histogram'], *colors*=['primary', 'tertiary', 'fill'], ***kwargs*)
Moving Average Convergence Divergence with Controllable MA Type.

Note that the first argument of *types* and *colors* refers to MACD, the second argument refers to MACD signal line and the third argument refers to MACD histogram.

`quantmod.ta.add_MAMA` (*self*, *fastlimit*=0.5, *slowlimit*=0.05, *types*=['line', 'line'], *colors*=['secondary', 'tertiary'], ***kwargs*)
MESA Adaptive Moving Average.

Note that the first argument of *types* and *colors* refers to MAMA while the second argument refers to FAMA.

`quantmod.ta.add MAVP` (*self*, *periods*, *minperiod*=2, *maxperiod*=30, *matype*=0, *type*='line', *color*='secondary', ***kwargs*)
Moving Average with Variable Period.

Parameters *periods* (*Series* or *array*) – Moving Average period over timeframe to analyze, as a 1-dimensional shape of same length as chart.

`quantmod.ta.add_MFI` (*self*, *timeperiod*=14, *type*='line', *color*='secondary', ***kwargs*)
Money Flow Index.

`quantmod.ta.add_MIDPOINT` (*self*, *timeperiod*=14, *type*='line', *color*='secondary', ***kwargs*)
Midpoint Price over Period.

`quantmod.ta.add_MINUS_DI` (*self*, *timeperiod*=14, *type*='line', *color*='decreasing', ***kwargs*)
Minus Directional Indicator.

`quantmod.ta.add_MINUS_DM` (*self*, *timeperiod*=14, *type*='line', *color*='decreasing', ***kwargs*)
Minus Directional Movement.

`quantmod.ta.add_MOM` (*self*, *timeperiod*=10, *type*='line', *color*='secondary', ***kwargs*)
Momentum Indicator.

`quantmod.ta.add_PLUS_DI` (*self*, *timeperiod*=14, *type*='line', *color*='increasing', ***kwargs*)
 Plus Directional Indicator.

`quantmod.ta.add_PLUS_DM` (*self*, *timeperiod*=14, *type*='line', *color*='increasing', ***kwargs*)
 Plus Directional Movement.

`quantmod.ta.add_PPO` (*self*, *fastperiod*=12, *slowperiod*=26, *matype*=0, *type*='line', *color*='secondary', ***kwargs*)
 Percent Price Oscillator.

`quantmod.ta.add_ROC` (*self*, *timeperiod*=10, *type*='line', *color*='tertiary', ***kwargs*)
 Rate of Change.

`quantmod.ta.add_ROCP` (*self*, *timeperiod*=10, *type*='line', *color*='tertiary', ***kwargs*)
 Rate of Change (Percentage).

`quantmod.ta.add_ROCR` (*self*, *timeperiod*=10, *type*='line', *color*='tertiary', ***kwargs*)
 Rate of Change (Ratio).

`quantmod.ta.add_ROCR100` (*self*, *timeperiod*=10, *type*='line', *color*='tertiary', ***kwargs*)
 Rate of Change (Ratio * 100).

`quantmod.ta.add_RSI` (*self*, *timeperiod*=14, *type*='line', *color*='secondary', ***kwargs*)
 Relative Strength Index.

`quantmod.ta.add_SAR` (*self*, *acceleration*=0.02, *maximum*=0.2, *type*='scatter', *color*='tertiary', ***kwargs*)
 Parabolic SAR.

`quantmod.ta.add_SAREXT` (*self*, *startvalue*=0, *offsetonreverse*=0, *accelerationinitlong*=0.02, *accelerationlong*=0.02, *accelerationmaxlong*=0.2, *accelerationinitshort*=0.02, *accelerationshort*=0.02, *accelerationmaxshort*=0.2, *type*='scatter', *color*='tertiary', ***kwargs*)
 Parabolic SAR Extended.

`quantmod.ta.add_SMA` (*self*, *timeperiod*=20, *type*='line', *color*='secondary', ***kwargs*)
 Simple Moving Average.

`quantmod.ta.add_STOCH` (*self*, *fastk_period*=5, *slowk_period*=3, *slowk_matype*=0, *slowd_period*=3, *slowd_matype*=0, *types*=['line', 'line'], *colors*=['primary', 'tertiary'], ***kwargs*)
 Slow Stochastic Oscillator.

Note that the first argument of types and colors refers to Slow Stoch %K, while second argument refers to Slow Stoch %D (signal line of %K obtained by MA).

`quantmod.ta.add_STOCHF` (*self*, *fastk_period*=5, *fastd_period*=3, *fastd_matype*=0, *types*=['line', 'line'], *colors*=['primary', 'tertiary'], ***kwargs*)
 Fast Stochastic Oscillator.

Note that the first argument of types and colors refers to Fast Stoch %K, while second argument refers to Fast Stoch %D (signal line of %K obtained by MA).

`quantmod.ta.add_STOCHRSI` (*self*, *timeperiod*=14, *fastk_period*=5, *fastd_period*=3, *fastd_matype*=0, *types*=['line', 'line'], *colors*=['primary', 'tertiary'], ***kwargs*)
 Stochastic Relative Strength Index.

Note that the first argument of types and colors refers to StochRSI %K while second argument refers to StochRSI %D (signal line of %K obtained by MA).

`quantmod.ta.add_T3` (*self*, *timeperiod*=20, *vfactor*=0.7, *type*='line', *color*='secondary', ***kwargs*)
 T3 Exponential Moving Average.

```
quantmod.ta.add_TEMA(self, timeperiod=26, type='line', color='secondary', **kwargs)
    Triple Moving Exponential Average.

quantmod.ta.add_TRIMA(self, timeperiod=20, type='line', color='secondary', **kwargs)
    Triangular Moving Average.

quantmod.ta.add_TRIX(self, timeperiod=15, type='area', color='secondary', **kwargs)
    1-day Rate of Change of Triple Smooth EMA.

quantmod.ta.add_ULTOSC(self, timeperiod=14, timeperiod2=14, timeperiod3=28, type='line',
                        color='secondary', **kwargs)
    Ultimate Oscillator.

quantmod.ta.add_WILLR(self, timeperiod=14, type='line', color='secondary', **kwargs)
    Williams %R.

quantmod.ta.add_WMA(self, timeperiod=20, type='line', color='secondary', **kwargs)
    Weighted Moving Average.
```

1.8 quantmod.tanolib module

Python native technical indicators

Ta-Lib indicators in 'ta.py' file.

1.9 quantmod.tools module

Functions meant for user access

All non-Chart related functions are in this module. For Chart-related functions go in 'factory.py'.

```
quantmod.tools.check_url(url=None)
    Check URL integrity.
```

Parameters `url` (*string*) – URL to be checked.

```
quantmod.tools.ensure_local_files()
    Ensure that filesystem is setup/filled out in a valid way.
```

```
quantmod.tools.get_config_file(*args)
    Return specified args from ~/config. as dict. Return all if no arguments are specified.
```

Example

```
get_config_file('sharing')

quantmod.tools.go_offline(connected=False)
    Take plotting offline.

__PLOTLY_OFFLINE_INITIALIZED is a secret variable in plotly/offline/offline.py.

Parameters connected (bool) – Determines if init_notebook_mode should be set to 'connected'.
    99% of time will not need to touch this.

quantmod.tools.go_online()
    Take plotting offline.
```

```
quantmod.tools.is_offline()
```

Check online/offline status.

```
quantmod.tools.reset_config_file()
```

Reset config file to package defaults.

```
quantmod.tools.set_config_file(sharing=None, theme=None, dimensions=None, of-  
fline=None, offline_url=None, offline_show_link=None,  
offline_link_text=None)
```

Set the keyword-value pairs in `~/config`.

Parameters

- **sharing** (*string or bool*) –
Sets the sharing level permission. True / ‘public’ - anyone can see this chart False / ‘private’ - only you can see this chart ‘secret’ - only people with the link can see the chart
- **theme** (*string*) – Sets the default theme. See `factory.get_themes()` for available themes.
- **dimensions** (*tuple*) – Sets the default (width, height) of the chart.
- **offline** (*bool*) – If true then the charts are rendered locally.
- **offline_show_link** (*bool*) – If true then the chart will show a link to plot.ly at the bottom right of the chart.
- **offline_link_text** (*string*) – Text to display as link at the bottom right of the chart.

1.10 Module contents

Quantmod

A powerful financial charting library based on R’s Quantmod.

With a Plotly backend and Cufflinks simplicity, Quantmod provides beautiful charts and a variety of quantitative and technical finance tools.

1.10.1 Author

@jackwluo

1.10.2 Credits

plotly.py : @chriddyp, @theengineear, et al. cufflinks : @jorgesantos

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

q

- `quantmod`, [18](#)
- `quantmod.auth`, [3](#)
- `quantmod.chart`, [3](#)
- `quantmod.core`, [10](#)
- `quantmod.datetools`, [11](#)
- `quantmod.factory`, [12](#)
- `quantmod.ta`, [14](#)
- `quantmod.tanolib`, [17](#)
- `quantmod.tools`, [17](#)

A

- add_ADX () (in module quantmod.ta), 14
- add_ADX () (quantmod.chart.Chart method), 3
- add_ADXR () (in module quantmod.ta), 14
- add_ADXR () (quantmod.chart.Chart method), 3
- add_APO () (in module quantmod.ta), 14
- add_APO () (quantmod.chart.Chart method), 4
- add_AROON () (in module quantmod.ta), 14
- add_AROON () (quantmod.chart.Chart method), 4
- add_AROONOSC () (in module quantmod.ta), 14
- add_AROONOSC () (quantmod.chart.Chart method), 4
- add_BBANDS () (in module quantmod.ta), 14
- add_BBANDS () (quantmod.chart.Chart method), 4
- add_BOP () (in module quantmod.ta), 14
- add_BOP () (quantmod.chart.Chart method), 4
- add_CCI () (in module quantmod.ta), 14
- add_CCI () (quantmod.chart.Chart method), 4
- add_CMO () (in module quantmod.ta), 14
- add_CMO () (quantmod.chart.Chart method), 4
- add_DEMA () (in module quantmod.ta), 15
- add_DEMA () (quantmod.chart.Chart method), 4
- add_DX () (in module quantmod.ta), 15
- add_DX () (quantmod.chart.Chart method), 4
- add_EMA () (in module quantmod.ta), 15
- add_EMA () (quantmod.chart.Chart method), 4
- add_HT_TRENDLINE () (in module quantmod.ta), 15
- add_HT_TRENDLINE () (quantmod.chart.Chart method), 4
- add_KAMA () (in module quantmod.ta), 15
- add_KAMA () (quantmod.chart.Chart method), 4
- add_MA () (in module quantmod.ta), 15
- add_MA () (quantmod.chart.Chart method), 4
- add_MACD () (in module quantmod.ta), 15
- add_MACD () (quantmod.chart.Chart method), 4
- add_MACDEXT () (in module quantmod.ta), 15
- add_MACDEXT () (quantmod.chart.Chart method), 4
- add_MAMA () (in module quantmod.ta), 15
- add_MAMA () (quantmod.chart.Chart method), 4
- add_MAVP () (in module quantmod.ta), 15
- add_MAVP () (quantmod.chart.Chart method), 5
- add_MFI () (in module quantmod.ta), 15
- add_MFI () (quantmod.chart.Chart method), 5
- add_MIDPOINT () (in module quantmod.ta), 15
- add_MIDPOINT () (quantmod.chart.Chart method), 5
- add_MINUS_DI () (in module quantmod.ta), 15
- add_MINUS_DI () (quantmod.chart.Chart method), 5
- add_MINUS_DM () (in module quantmod.ta), 15
- add_MINUS_DM () (quantmod.chart.Chart method), 5
- add_MOM () (in module quantmod.ta), 15
- add_MOM () (quantmod.chart.Chart method), 5
- add_PLUS_DI () (in module quantmod.ta), 15
- add_PLUS_DI () (quantmod.chart.Chart method), 5
- add_PLUS_DM () (in module quantmod.ta), 16
- add_PLUS_DM () (quantmod.chart.Chart method), 5
- add_PPO () (in module quantmod.ta), 16
- add_PPO () (quantmod.chart.Chart method), 5
- add_ROC () (in module quantmod.ta), 16
- add_ROC () (quantmod.chart.Chart method), 5
- add_ROCP () (in module quantmod.ta), 16
- add_ROCP () (quantmod.chart.Chart method), 5
- add_ROCR () (in module quantmod.ta), 16
- add_ROCR () (quantmod.chart.Chart method), 5
- add_ROCR100 () (in module quantmod.ta), 16
- add_ROCR100 () (quantmod.chart.Chart method), 5
- add_RSI () (in module quantmod.ta), 16
- add_RSI () (quantmod.chart.Chart method), 5
- add_SAR () (in module quantmod.ta), 16
- add_SAR () (quantmod.chart.Chart method), 5
- add_SAREXT () (in module quantmod.ta), 16
- add_SAREXT () (quantmod.chart.Chart method), 5
- add_SMA () (in module quantmod.ta), 16
- add_SMA () (quantmod.chart.Chart method), 5
- add_STOCH () (in module quantmod.ta), 16
- add_STOCH () (quantmod.chart.Chart method), 5
- add_STOCHF () (in module quantmod.ta), 16
- add_STOCHF () (quantmod.chart.Chart method), 6
- add_STOCHRSI () (in module quantmod.ta), 16
- add_STOCHRSI () (quantmod.chart.Chart method), 6
- add_T3 () (in module quantmod.ta), 16

`add_T3()` (*quantmod.chart.Chart method*), 6
`add_TEMA()` (*in module quantmod.ta*), 16
`add_TEMA()` (*quantmod.chart.Chart method*), 6
`add_TRIMA()` (*in module quantmod.ta*), 17
`add_TRIMA()` (*quantmod.chart.Chart method*), 6
`add_TRIX()` (*in module quantmod.ta*), 17
`add_TRIX()` (*quantmod.chart.Chart method*), 6
`add_ULTOSC()` (*in module quantmod.ta*), 17
`add_ULTOSC()` (*quantmod.chart.Chart method*), 6
`add_WILLR()` (*in module quantmod.ta*), 17
`add_WILLR()` (*quantmod.chart.Chart method*), 6
`add_WMA()` (*in module quantmod.ta*), 17
`add_WMA()` (*quantmod.chart.Chart method*), 6
`adjust()` (*quantmod.chart.Chart method*), 6
`adjust_volume()` (*quantmod.chart.Chart method*), 6

C

`Chart` (*class in quantmod.chart*), 3
`chart_series()` (*in module quantmod.core*), 10
`check_file_permissions()` (*in module quantmod.auth*), 3
`check_url()` (*in module quantmod.tools*), 17

D

`date_to_int()` (*in module quantmod.datetools*), 11

E

`ensure_local_files()` (*in module quantmod.tools*), 17

G

`get_base_layout()` (*in module quantmod.factory*), 12
`get_config_file()` (*in module quantmod.tools*), 17
`get_date_from_today()` (*in module quantmod.datetools*), 11
`get_path()` (*in module quantmod.auth*), 3
`get_skeleton()` (*in module quantmod.factory*), 12
`get_source()` (*in module quantmod.factory*), 12
`get_sources()` (*in module quantmod.factory*), 12
`get_symbol()` (*in module quantmod.core*), 10
`get_template()` (*in module quantmod.factory*), 12
`get_theme()` (*in module quantmod.factory*), 13
`get_themes()` (*in module quantmod.factory*), 13
`go_offline()` (*in module quantmod.tools*), 17
`go_online()` (*in module quantmod.tools*), 17

H

`has_adjusted_close` (*quantmod.chart.Chart attribute*), 6
`has_adjusted_high` (*quantmod.chart.Chart attribute*), 7
`has_adjusted_low` (*quantmod.chart.Chart attribute*), 7

`has_adjusted_open` (*quantmod.chart.Chart attribute*), 7
`has_close` (*quantmod.chart.Chart attribute*), 7
`has_dividend` (*quantmod.chart.Chart attribute*), 7
`has_high` (*quantmod.chart.Chart attribute*), 7
`has_low` (*quantmod.chart.Chart attribute*), 7
`has_OHLC` (*quantmod.chart.Chart attribute*), 6
`has_OHLCV` (*quantmod.chart.Chart attribute*), 6
`has_open` (*quantmod.chart.Chart attribute*), 7
`has_volume` (*quantmod.chart.Chart attribute*), 7
`head()` (*quantmod.chart.Chart method*), 7

I

`int_to_date()` (*in module quantmod.datetools*), 11
`ipplot()` (*quantmod.chart.Chart method*), 7
`is_offline()` (*in module quantmod.tools*), 17

M

`make_additions()` (*in module quantmod.factory*), 13
`make_colors()` (*in module quantmod.factory*), 13
`make_layout()` (*in module quantmod.factory*), 13
`make_traces()` (*in module quantmod.factory*), 14

P

`plot()` (*quantmod.chart.Chart method*), 8

Q

`quantmod` (*module*), 18
`quantmod.auth` (*module*), 3
`quantmod.chart` (*module*), 3
`quantmod.core` (*module*), 10
`quantmod.datetools` (*module*), 11
`quantmod.factory` (*module*), 12
`quantmod.ta` (*module*), 14
`quantmod.tanolib` (*module*), 17
`quantmod.tools` (*module*), 17

R

`reset_config_file()` (*in module quantmod.tools*), 18

S

`set_config_file()` (*in module quantmod.tools*), 18
`shape` (*quantmod.chart.Chart attribute*), 9
`string_to_date()` (*in module quantmod.datetools*), 12
`strip_figure()` (*in module quantmod.factory*), 14

T

`tail()` (*quantmod.chart.Chart method*), 9
`to_figure()` (*quantmod.chart.Chart method*), 9
`to_frame()` (*quantmod.chart.Chart method*), 10