# Package: seasonalityPlot (via r-universe)

August 17, 2024

**Type** Package

**Title** Seasonality Variation Plots of Stock Prices and Cryptocurrencies

**Version** 1.2.1

**Description** The price action at any given time is determined by investor sentiment and market conditions. Although there is no established principle, over a long period of time, things often move with a certain periodicity. This is sometimes referred to as anomaly. The seasonPlot() function in this package calculates and visualizes the average value of price movements over a year for any given period. In addition, the monthly increase or decrease in price movement is represented with a colored background. This seasonPlot() function can use the same symbols as the 'quantmod' package (e.g. ^IXIC, ^DJI, SPY, BTC-USD, and ETH-USD etc).

**Depends** R (>= 4.0.0)

**Imports** magrittr, quantmod, dygraphs, plotrix, htmltools, grDevices, graphics, zoo, lubridate, crypto2, TTR, assertthat

**Suggests** testthat

**License** Artistic-2.0

**Encoding** UTF-8

**LazyData** true

**URL** https://github.com/kumeS/seasonalityPlot, https://kumes.github.io/seasonalityPlot/

**BugReports** https://github.com/kumeS/seasonalityPlot/issues

**RoxygenNote** 7.3.1

**Repository** https://kumes.r-universe.dev

**RemoteUrl** https://github.com/kumes/seasonalityplot

**RemoteRef** HEAD

**RemoteSha** f0651942004d844450c2f8d52958696d47319049

# Contents



---

`CryptoRSIheatmap` *CryptoRSI Heatmap Function*

---

## Description

This function provides a heatmap visualization of RSI values for a specified number of cryptocurrencies. Selected randomly based on their market cap ranking, it aims to offer insights into the current market sentiment.

## Usage

```
CryptoRSIheatmap(
  coin_num = 200,
  useRank = 1000,
  n = 21,
  useRankPlot = TRUE,
  OutputData = FALSE
)
```

## Arguments

| | |
|---|---|
| `coin_num` | An integer specifying the number of coins to display in the heatmap. |
| `useRank` | An integer defining the range within which coins are randomly selected based on their market cap ranking. |
| `n` | An integer indicating the number of periods for calculating moving averages in the RSI computation. |
| `useRankPlot` | A boolean that determines if the x-axis should plot ranks instead of sequential numbers. |
| `OutputData` | A boolean that decides if the function should return the final plot data table. |

## Details

CryptoRSI Heatmap

Generates a heatmap of the Relative Strength Index (RSI) for a randomly selected subset of cryptocurrencies. This function leverages the ‘crypto2‘ and ‘TTR‘ packages to fetch cryptocurrency data and calculate RSI values, respectively. The heatmap visualizes RSI values to identify potential overbought or oversold conditions in the crypto market.

## Value

If 'OutputData' is TRUE, returns a data frame with symbols, ranks (or sequential numbers), RSI values, and colors for plotting. Otherwise, displays a heatmap plot.

## Author(s)

Satoshi Kume

## Examples

```
## Not run:
CryptoRSIheatmap(coin_num = 200, useRank = 1000, n = 21,
        useRankPlot = TRUE, OutputData = FALSE)

## End(Not run)
```

---

`seasonPlot` *seasonPlot: create seasonality variation plots for stock prices or cryptocurrencies*

---

## Description

This function is to easily create seasonality variation plots of annual averages of stock prices or cryptocurrencies with some color options. This can use the same symbols as the 'quantmod' package. For the average calculation, the trading days for each month are aligned and then the percentages of change with the beginning of the year being zero are calculated. This function can set any given time period for averaging. In addition, years with many missing data are automatically excluded before the average calculation. The positive and negative monthly changes are shown in green and red background color, respectively.

## Usage

```
seasonPlot(
  Symbols,
  StartYear = lubridate::year(Sys.Date()) - 11,
  EndYear = lubridate::year(Sys.Date()) - 1,
  useAdjusted = FALSE,
  LineColor = 1,
  xlab = "Month",
  BackgroundMode = TRUE,
  alpha = 0.05,
  OutputData = FALSE,
  Save = FALSE,
  output_width = 1000,
  output_height = 700,
  family = "Helvetica",
  PlotAll = FALSE
)
```

## Arguments

| | |
|---|---|
| Symbols | a character vector specifying the names of each symbol to be loaded. e.g. ^IXIC (NASDAQ Composite), ^DJI (Dow Jones Industrial Average), SPY (SPDR S&P500 ETF), BTC-USD (Bitcoin), ETH-USD (Ethereum), and XRP-USD (Ripple). |
| StartYear | a numeric of start year (Common Er). The default is 11 years from now. |
| EndYear | a numeric of end year (Common Er). The default is the last year. |
| useAdjusted | Choose whether to use the closing price adjusted for dividends. If FALSE, normal close price is used. In the case of cryptocurrencies, the useAdjusted option is expected to return the same result. |
| LineColor | a numeric between 1 and 4; The value 1 is to select red1, the value 2 is to select blue1, the value 3 is to select green1, and the value 4 is to select black. When BackgroundMode is TRUE, this argument is disabled. |
| xlab | a character of X-axis label. |
| BackgroundMode | a logical; draw a background color by react. |
| alpha | a numeric; The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque). |
| OutputData | a logical; output as a data.frame type or not. |
| Save | a logical; save as an image (PNG) or not |
| output_width | a output size of width (pixel). Initial value recommended. |
| output_height | a output size of height (pixel). Initial value recommended. |
| family | a character of font. |
| PlotAll | a logical; display all period by dygraph function or not. |

## Value

plot results

## Author(s)

Satoshi Kume

## Examples

```
## Plot seasonality of NASDAQ Composite Index (^IXIC)
seasonPlot(Symbols = "^IXIC", useAdjusted = TRUE)

## Plot seasonality of Bitcoin (BTC-USD)
seasonPlot(Symbols = "BTC-USD", StartYear=2015, EndYear=2020)
```

# Index