

# R 中的量化投资包之 candlesticks 包简介

MatrixSpk

## 目录

<b>1</b>	<b>引言</b>	<b>1</b>
<b>2</b>	<b>candlesticks 包的安装</b>	<b>1</b>
<b>3</b>	<b>candlesticks 包核心函数分类与作用</b>	<b>2</b>
3.1	形态识别类函数 . . . . .	2
3.2	趋势及动量函数 . . . . .	14
3.3	辅助计算类函数 . . . . .	18
<b>4</b>	<b>投资信号生成策略示例</b>	<b>25</b>
4.1	反转信号组合 . . . . .	25
4.2	趋势延续策略 . . . . .	25

## 1 引言

## 2 candlesticks 包的安装

旧版本的 candlesticks 包可以从R-Forge上安装，代码如下：

```
install.packages("candlesticks", repos="http://R-Forge.R-project.org")
```

最新版本的 candlesticks 包可以从 github 上下载并安装，代码如下：

```
devtools::install_github(repo="https://github.com/dengyishuo/candlesticks")
```

### 3 candlesticks 包核心函数分类与作用

candlesticks 包里提供了众多识别蜡烛图形态的函数，按作用可以将其分为几类：

- 形态识别类函数
- 趋势识别类函数
- 辅助计算类函数

顾名思义，形态识别类函数用以识别“早晨之星”、“锤子线”、“乌云盖顶”等常见的蜡烛图形态；趋势识别类函数则主要用以识别当前金融标的的价格趋势；而辅助计算函数则是帮助进行一些杂项的运算。

#### 3.1 形态识别类函数

形态识别类函数是 candlesticks 包中为数最众的函数，它们又可以区分为翻转形态函数、持续形态函数以及特殊形态函数。

##### 3.1.1 反转形态函数

**3.1.1.1 CSPDarkCloudCover** 该函数的作用是检测“乌云盖顶”形态，属看跌反转信号，通常出现在上升趋势末端。

参数示例：

```
CSP <- CSPDarkCloudCover(AAPL)
head(CSP)
```

```
##           DarkCloudCover
## 2020-01-02             NA
## 2020-01-03            FALSE
## 2020-01-06            FALSE
## 2020-01-07            FALSE
## 2020-01-08            FALSE
## 2020-01-09            FALSE
```

```
CSP <- CSPDarkCloudCover(AAPL, n=20, minbodysizeMedian=1)
head(CSP)
```

```
##           DarkCloudCover
## 2020-01-02             NA
## 2020-01-03            FALSE
## 2020-01-06            FALSE
## 2020-01-07            FALSE
## 2020-01-08            FALSE
## 2020-01-09            FALSE
```

```
# filter dark cloud covers that occur in uptrends.
# the lag of 2 periods of the time series for trend detection
# ensures that the uptrend is active *before* the
# dark cloud cover occurs.
```

```
CSP <- CSPDarkCloudCover(AAPL) & TrendDetectionChannel(lag(AAPL,k=2))[, "UpTrend"]
head(CSP)
```

```
##           DarkCloudCover
## 2020-01-02             NA
## 2020-01-03            FALSE
```

```
## 2020-01-06      FALSE
## 2020-01-07      FALSE
## 2020-01-08      FALSE
## 2020-01-09      FALSE
```

扩展:

在上升趋势中, 若检测到乌云盖顶且 \$ RSI 70\$, 可视为卖出信号。

**3.1.1.2 CSPPiercingPattern** 乌云盖顶: 上涨后出现高开低走阴线, 预示顶部反转。刺透形态: 下跌后出现低开高走阳线, 暗示底部反转。

```
CSP<- CSPPiercingPattern(AAPL)

# filter piercing patterns that occur in downtrends.
# the lag of 2 periods of the time series for trend detection
# ensures that the uptrend is active *before* the
# dark cloud cover occurs.

CSP <- CSPPiercingPattern(AAPL) & TrendDetectionChannel(lag(AAPL,k=2))[, "DownTrend"]
```

**3.1.1.3 CSPHammer** 作用: 识别锤子线(看涨)或上吊线(看跌), 需长下影线。

参数示例:

```
# filter for hammer patterns
CSP <- CSPHammer(AAPL)

# filter for hammer patterns that occur in downtrends
CSP <- CSPHammer(AAPL) & TrendDetectionChannel(AAPL)[, "DownTrend"]
```

**3.1.1.4 CSPInvertedHammer** 作用: 用来识别倒锤子线, 倒锤子线出现在下跌趋势末端(上影线长、实体小), 预示趋势可能翻转

```
# filter for hanging man patterns

CSP <- CSPInvertedHammer(AAPL)

# filter for hanging man patterns that occur in downtrends

CSP <- CSPInvertedHammer(AAPL) & TrendDetectionChannel(AAPL)[,"DownTrend"]
```

```
CSP <- CSPStar(AAPL)

# allow only a small second candle body
CSP <- CSPStar(AAPL, maxbodysizeMedian=.5)

# filter for morning stars that occur in downtrends
CSP <- MorningStar <- CSPStar(AAPL)[,"MorningStar"]&TrendDetectionChannel(lag(AAPL,k=3)
```

#### 3.1.1.5 CSPStar

**3.1.1.6 CSPDoji** 作用：识别十字星 (Doji)，反映市场犹豫，常预示趋势反转。

参数示例：

```
CSP <- CSPDoji(AAPL)

# filter for doji patterns that have exactly equal
# open and close prices

CSP <- CSPDoji(AAPL, maxbodyCL=1)

# filter for gravestone doji patterns that occur in uptrends
```

```
CSP <- CSPDoji(AAPL)[,"GravestoneDoji"] & TrendDetectionChannel(AAPL)[,"UpTrend"]
```

应用示例:

若十字星出现在长期下跌后,且成交量放大,可能暗示底部反转。

**3.1.1.7 CSPEngulfing** 作用:检测吞没形态(看涨/看跌),第二根实体完全覆盖前一根实体。

参数示例:

```
CSP <- CSPEngulfing(AAPL)
```

```
CSP <- CSPEngulfing(AAPL)[,"Bear.Engulfing"] & TrendDetectionChannel(AAPL)[,"UpTrend"]
```

应用示例:

看涨吞没形态配合 MACD 底背离,可作为买入信号。

**3.1.1.8 CSPHarami** 作用:检测孕线形态,小实体完全包含在前一根大实体内。

参数示例:

```
CSP <- CSPHarami(AAPL)
```

```
# filter for bullish harami that occur in downtrends
```

```
BullHarami <- CSPHarami(AAPL)[,"Bull.Harami"] & TrendDetectionChannel(lag(AAPL,k=2))[,"D
```

**3.1.1.9 CSPKicking** 作用:CSPKicking 用于检测金融市场的“跳空启动形态”(Kicking Pattern),该形态由两根跳空且颜色相反的蜡烛组成,通常出现在趋势反转阶段。

看涨启动形态:第一根为长阴线,第二根跳空高开形成长阳线,预示空头趋势反转。

看跌启动形态：第一根为长阳线，第二根跳空低开形成长阴线，提示多头趋势结束。

```
# look for Kicking Pattern right out of the textbook
# they occur only once in a blue moon

CSP <- CSPKicking(AAPL, ignoreShadows=FALSE, maxshadowCL=0)
# use less strict filter rules

CSP <- CSPKicking(AAPL)
```

应用示例:

看涨启动形态策略

```
# 检测看涨 Kicking 形态
bullish_kick <- CSPKicking(AAPL)[,"Bull.Kicking"]&TrendDetectionChannel(lag(AAPL,k=2))

# 结合成交量验证：第二根阳线需放量
volume_cond <- quantmod::Vo(AAPL) > SMA(quantmod::Vo(AAPL), 10)

# 生成买入信号
buy_signal <- bullish_kick & volume_cond
```

逻辑：跳空高开阳线突破前阴线高点，且成交量放大，确认反转可信度。

看跌启动形态策略:

```
# 检测看跌 Kicking 形态
bearish_kick <- CSPKicking(AAPL)[,"Bear.Kicking"]&&TrendDetectionChannel(lag(AAPL,k=2))

## Warning in CSPKicking(AAPL)[, "Bear.Kicking"] &&
## TrendDetectionChannel(lag(AAPL, : 'length(x) = 1315 > 1' in coercion to
## 'logical(1)'
```

```
# 结合 RSI 超买: RSI >70 时增强信号
rsi_cond <- RSI(quantmod::C1(AAPL)) > 70

# 生成卖出信号
sell_signal <- bearish_kick & rsi_cond
```

逻辑：跳空低开阴线伴随超买指标，预示多头力量衰竭。

```
CSP <- CSPStomach(AAPL)

CSP <- CSPStomach(AAPL)[,"AboveTheStomach"] & TrendDetectionChannel(lag(AAPL,k=2))[,"Do
```

#### 3.1.1.10 CSPStomach

### 3.1.2 持续形态函数

**3.1.2.1 CSPGap** 用于检测缺口形态（如向上跳空或向下跳空），缺口形态常被视为短期趋势延续或反转的标志。

```
CSP <- CSPGap(AAPL) # examine whole candle lenght

CSP <- CSPGap(AAPL, ignoreShadows=TRUE) # examine only candle bodies
```

**3.1.2.2 CSPTasukiGap** 作用：识别跳空并列阴阳线（Tasuki Gap），预示趋势延续。

参数示例：

```
CSP <- CSPTasukiGap(AAPL)

# filter upside tasuki gaps in uptrends
```



```
CSP <- CSPTasukiGap(AAPL)[,"UpsideTasukiGap"] & TrendDetectionChannel(lag(AAPL,k=3))[, "U
head(CSP)
```

```
##                UpsideTasukiGap
## 2020-01-02             NA
## 2020-01-03            FALSE
## 2020-01-06            FALSE
## 2020-01-07            FALSE
## 2020-01-08            FALSE
## 2020-01-09            FALSE
```

```
CSP_TRUE <- CSP[, "UpsideTasukiGap"][which(CSP[, "UpsideTasukiGap"] == TRUE)]
CSP_TRUE
```

```
##                UpsideTasukiGap
## 2020-06-24             TRUE
```

应用示例：

上升趋势中跳空后出现小实体阴线但未回补缺口，可加仓。

**3.1.2.3 CSPMarubozu** 作用：识别光头光脚蜡烛（无影线），显示极端单边行情。

参数示例：

```
CSP <- CSPMarubozu(AAPL)

# include not-so-long-marubozus
CSP <- CSPMarubozu(AAPL, ATRFactor=.8)

# filter for white closing marubozus (Cl(TS)=Hi(TS))
CSP <- CSPMarubozu(AAPL, maxuppershadowCL=0)[,"WhiteMarubozu"]
```

**3.1.2.4 CSPThreeMethods** 作用：检测“三法形态”（上升/下降三法），属趋势中继信号。

参数示例：

```
# filter rising three methods in uptrends

CSP <- CSPThreeMethods(AAPL)[,"RisingThreeMethods"] & TrendDetectionChannel(lag(AAPL,k=
```

**3.1.2.5 CSPInsideDay** 识别“内部日”形态，即当日价格波动范围完全包含于前一日价格区间内，暗示市场犹豫或潜在反转可能。

```
CSP <- CSPInsideDay(AAPL)
```

```
CSP <- CSPOutsideDay(AAPL)
```

**3.1.2.6 CSPOutsideDay**

```
# filter three inside up in downtrends

CSP <- CSPThreeInside(AAPL)[,"ThreeInsideUp"]&TrendDetectionChannel(lag(AAPL,k=3))[,"Do

CSP <- CSPThreeInside(AAPL)[,"ThreeInsideDown"]&TrendDetectionChannel(lag(AAPL,k=3))[,"
```

**3.1.2.7 CSPThreeInside**

```
CSP <- CSPThreeOutside(AAPL)
```

```
# filter three outside up in downtrends
CSP <- CSPThreeOutside(AAPL)[,"ThreeOutsideUp"] & TrendDetectionChannel(lag(AAPL,k=3))[,
```

### 3.1.2.8 CSPThreeOutside

**3.1.2.9 CSPShortCandle/CSPLoneCandle** 作用：检测短实体蜡烛（实体占长度 30% 以下），反映市场犹豫。参数示例：

```
CSP <- CSPLongCandle(AAPL)

CSP <- CSPShortCandle(AAPL, threshold=.5) # filter for very small candles
```

短实体常出现在震荡或趋势反转前。

```
CSP <- CSPNLongBlackCandles(AAPL, N=3)
```

### 3.1.2.10 CSPNLongWhiteCandles/CSPNLongBlackCandles

**3.1.2.11 CSPLongCandleBody/CSPShortCandleBody** 作用：筛选长实体蜡烛（实体占蜡烛长度 70% 以上）。参数示例：

```
CSP <- CSPLongCandleBody(AAPL)

CSP <- CSPShortCandleBody(AAPL)
```

长实体反映单边强势，常见于趋势加速阶段。

```
CSP <- CSPNLongWhiteCandleBodies(AAPL)

CSP <- CSPNLongBlackCandleBodies(AAPL, N=4, n=50, threshold=1.2)
```

### 3.1.2.12 CSPNLongWhiteCandleBodies/CSPNLongBlackCandleBodies

```
CSP <- CSPNHigherClose(AAPL, N=3) # filter for 3 consecutive higher close
```

### 3.1.2.13 CSPNHigherClose

```
CSP <- CSPNLowerClose(AAPL, N=4) # filter for 4 consecutive lower close
```

### 3.1.2.14 CSPNLowerClose

## 3.1.3 特殊形态函数

**3.1.3.1 CSPThreeWhiteSoldiers** 作用：检测“三白兵”形态，属强势看涨信号。

参数示例：

```
CSP <- CSPThreeWhiteSoldiers(AAPL)
```

每根阳线实体需占蜡烛长度的 70% 以上 67。

**3.1.3.2 CSPThreeBlackCrows** 作用：识别“三只乌鸦”形态，预示下跌延续。参数示例：

```
CSP <- CSPThreeBlackCrows(AAPL)
```

与三白兵逻辑相反，用于看跌趋势确认。

```
TLS <- CSPThreeLineStrike(AAPL)

# how often does that occur?
colSums(TLS, na.rm=TRUE)
```

### 3.1.3.3 CSPThreeLineStrike

```
## Bull.ThreeLineStrike Bear.ThreeLineStrike
##                                0                                0
```

```
# when did that occur?
TLS[TLS[,1]>0 | TLS[,2]>0,]
```

```
##      Bull.ThreeLineStrike Bear.ThreeLineStrike
```

```
# filter for bearish three line strikes that occur in downtrends
TLS1 <- CSPThreeLineStrike(AAPL)[,"Bear.ThreeLineStrike"] &
  TrendDetectionChannel(lag(AAPL,k=4))[, "DownTrend"]
TLS1[TLS1[,1]>0,]
```

```
##      Bear.ThreeLineStrike
```

```
# show in a chart
chartSeries(AAPL["2020-09::2020-10"])
```



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

## 3.2 趋势及动量函数

### 3.2.1 TrendDetectionSMA/TrendDetectionChannel

SMA 趋势检测：通过移动平均线方向判断当前趋势 7。

通道检测：结合价格通道（如 Donchian 通道）判断支撑/压力位。

```
# create chart of AAPL
chartSeries(AAPL, subset="last 1 year", TA=NULL)
```



```
# visualize the result of trend detection in a indicator box  
addTA(TrendDetectionSMA(AAPL)[,4])
```



```
# filter AAPL for Hammer Candlestick Patterns that occur in downtrends
Hammer <- CSPHammer(AAPL) & TrendDetectionSMA(AAPL)[,"DownTrend"]

# how frequent are these hammers?
colSums(Hammer, na.rm=TRUE)
```

```
## Hammer
##      15
```

```
# create chart of Apple
chartSeries(AAPL, subset="last 1 year", TA=NULL)
```



```
# visualize the result of trend detection in a indicator box
addTA(TrendDetectionChannel(AAPL)[,4])
```





```
# filter YHOO for Hammer Candlestick Patterns that occur in downtrends
Hammer <- CSPHammer(AAPL) & TrendDetectionChannel(AAPL)[, "DownTrend"]

# how frequent are these hammers?
colSums(Hammer, na.rm=TRUE)
```

```
## Hammer
##      9
```

### 3.2.2 DonchianChannel2

生成唐奇安通道，用于突破策略的信号生成（如价格突破通道上轨为买入信号）

```
dc <- DonchianChannel2(AAPL)

dc
```

```
##           high      mid      low
## 2020-01-02      NA      NA      NA
## 2020-01-03      NA      NA      NA
## 2020-01-06      NA      NA      NA
## 2020-01-07      NA      NA      NA
## 2020-01-08      NA      NA      NA
## 2020-01-09      NA      NA      NA
## 2020-01-10      NA      NA      NA
## 2020-01-13      NA      NA      NA
## 2020-01-14      NA      NA      NA
## 2020-01-15      NA      NA      NA
##      ...
## 2025-03-13 244.03 229.470 214.91
## 2025-03-14 244.03 226.225 208.42
## 2025-03-17 244.03 226.225 208.42
## 2025-03-18 241.37 224.895 208.42
## 2025-03-19 241.37 224.895 208.42
## 2025-03-20 241.37 224.895 208.42
## 2025-03-21 241.37 224.895 208.42
## 2025-03-24 236.16 222.290 208.42
## 2025-03-25 225.84 217.130 208.42
## 2025-03-26 224.10 216.260 208.42
```

### 3.3 辅助计算类函数

#### 3.3.1 is.HL

```
is.HL(AAPL)
```

```
## [1] TRUE
```

## 3.3.2 is.OC

```
is.OC(AAPL)
```

```
## [1] TRUE
```

## 3.3.3 LagOC

```
LagOC(AAPL)           # Lag OC series by one period
```

```
##           Open.Lag.1 Close.Lag.1
## 2020-01-02           NA           NA
## 2020-01-03       74.0600       75.0875
## 2020-01-06       74.2875       74.3575
## 2020-01-07       73.4475       74.9500
## 2020-01-08       74.9600       74.5975
## 2020-01-09       74.2900       75.7975
## 2020-01-10       76.8100       77.4075
## 2020-01-13       77.6500       77.5825
## 2020-01-14       77.9100       79.2400
## 2020-01-15       79.1750       78.1700
##           ...
## 2025-03-13      220.1400      216.9800
## 2025-03-14      215.9500      209.6800
## 2025-03-17      211.2500      213.4900
## 2025-03-18      213.3100      214.0000
## 2025-03-19      214.1600      212.6900
## 2025-03-20      214.2200      215.2400
## 2025-03-21      213.9900      214.1000
## 2025-03-24      211.5600      218.2700
## 2025-03-25      221.0000      220.7300
## 2025-03-26      220.7700      223.7500
```

```
LagOC(AAPL, k=1:3) # Lag OC series by one, two and three periods
```

```
##           Open.Lag.1 Open.Lag.2 Open.Lag.3 Close.Lag.1 Close.Lag.2 Close.Lag.3
## 2020-01-02           NA           NA           NA           NA           NA           NA
## 2020-01-03      74.0600           NA           NA      75.0875           NA           NA
## 2020-01-06      74.2875      74.0600           NA      74.3575      75.0875           NA
## 2020-01-07      73.4475      74.2875      74.0600      74.9500      74.3575      75.0875
## 2020-01-08      74.9600      73.4475      74.2875      74.5975      74.9500      74.3575
## 2020-01-09      74.2900      74.9600      73.4475      75.7975      74.5975      74.9500
## 2020-01-10      76.8100      74.2900      74.9600      77.4075      75.7975      74.5975
## 2020-01-13      77.6500      76.8100      74.2900      77.5825      77.4075      75.7975
## 2020-01-14      77.9100      77.6500      76.8100      79.2400      77.5825      77.4075
## 2020-01-15      79.1750      77.9100      77.6500      78.1700      79.2400      77.5825
##           ...
## 2025-03-13      220.1400      223.8100      235.5400      216.9800      220.8400      227.4800
## 2025-03-14      215.9500      220.1400      223.8100      209.6800      216.9800      220.8400
## 2025-03-17      211.2500      215.9500      220.1400      213.4900      209.6800      216.9800
## 2025-03-18      213.3100      211.2500      215.9500      214.0000      213.4900      209.6800
## 2025-03-19      214.1600      213.3100      211.2500      212.6900      214.0000      213.4900
## 2025-03-20      214.2200      214.1600      213.3100      215.2400      212.6900      214.0000
## 2025-03-21      213.9900      214.2200      214.1600      214.1000      215.2400      212.6900
## 2025-03-24      211.5600      213.9900      214.2200      218.2700      214.1000      215.2400
## 2025-03-25      221.0000      211.5600      213.9900      220.7300      218.2700      214.1000
## 2025-03-26      220.7700      221.0000      211.5600      223.7500      220.7300      218.2700
```

```
# this will return a 6 columns xts object
```

### 3.3.4 LagOHLC

```
LagOHLC(AAPL) # Lag OHLC series by one period
```

```
##           Open.Lag.1 High.Lag.1 Low.Lag.1 Close.Lag.1
```

```
## 2020-01-02      NA      NA      NA      NA
## 2020-01-03    74.0600    75.1500    73.7975    75.0875
## 2020-01-06    74.2875    75.1450    74.1250    74.3575
## 2020-01-07    73.4475    74.9900    73.1875    74.9500
## 2020-01-08    74.9600    75.2250    74.3700    74.5975
## 2020-01-09    74.2900    76.1100    74.2900    75.7975
## 2020-01-10    76.8100    77.6075    76.5500    77.4075
## 2020-01-13    77.6500    78.1675    77.0625    77.5825
## 2020-01-14    77.9100    79.2675    77.7875    79.2400
## 2020-01-15    79.1750    79.3925    78.0425    78.1700
##      ...
## 2025-03-13    220.1400    221.7500    214.9100    216.9800
## 2025-03-14    215.9500    216.8400    208.4200    209.6800
## 2025-03-17    211.2500    213.9500    209.5800    213.4900
## 2025-03-18    213.3100    215.2200    209.9700    214.0000
## 2025-03-19    214.1600    215.1500    211.4900    212.6900
## 2025-03-20    214.2200    218.7600    213.7500    215.2400
## 2025-03-21    213.9900    217.4900    212.2200    214.1000
## 2025-03-24    211.5600    218.8400    211.2800    218.2700
## 2025-03-25    221.0000    221.4800    218.5800    220.7300
## 2025-03-26    220.7700    224.1000    220.0800    223.7500
```

```
LagOHLC(AAPL, k=1:3) # Lag OHLC series by one, two and three periods this will return a
```

```
##      Open.Lag.1 Open.Lag.2 Open.Lag.3 High.Lag.1 High.Lag.2 High.Lag.3
## 2020-01-02      NA      NA      NA      NA      NA      NA
## 2020-01-03    74.0600      NA      NA    75.1500      NA      NA
## 2020-01-06    74.2875    74.0600      NA    75.1450    75.1500      NA
## 2020-01-07    73.4475    74.2875    74.0600    74.9900    75.1450    75.1500
## 2020-01-08    74.9600    73.4475    74.2875    75.2250    74.9900    75.1450
## 2020-01-09    74.2900    74.9600    73.4475    76.1100    75.2250    74.9900
## 2020-01-10    76.8100    74.2900    74.9600    77.6075    76.1100    75.2250
## 2020-01-13    77.6500    76.8100    74.2900    78.1675    77.6075    76.1100
## 2020-01-14    77.9100    77.6500    76.8100    79.2675    78.1675    77.6075
```

```
## 2020-01-15    79.1750    77.9100    77.6500    79.3925    79.2675    78.1675
##      ...
## 2025-03-13    220.1400    223.8100    235.5400    221.7500    225.8400    236.1600
## 2025-03-14    215.9500    220.1400    223.8100    216.8400    221.7500    225.8400
## 2025-03-17    211.2500    215.9500    220.1400    213.9500    216.8400    221.7500
## 2025-03-18    213.3100    211.2500    215.9500    215.2200    213.9500    216.8400
## 2025-03-19    214.1600    213.3100    211.2500    215.1500    215.2200    213.9500
## 2025-03-20    214.2200    214.1600    213.3100    218.7600    215.1500    215.2200
## 2025-03-21    213.9900    214.2200    214.1600    217.4900    218.7600    215.1500
## 2025-03-24    211.5600    213.9900    214.2200    218.8400    217.4900    218.7600
## 2025-03-25    221.0000    211.5600    213.9900    221.4800    218.8400    217.4900
## 2025-03-26    220.7700    221.0000    211.5600    224.1000    221.4800    218.8400
##      Low.Lag.1 Low.Lag.2 Low.Lag.3 Close.Lag.1 Close.Lag.2 Close.Lag.3
## 2020-01-02         NA         NA         NA         NA         NA         NA
## 2020-01-03    73.7975         NA         NA    75.0875         NA         NA
## 2020-01-06    74.1250    73.7975         NA    74.3575    75.0875         NA
## 2020-01-07    73.1875    74.1250    73.7975    74.9500    74.3575    75.0875
## 2020-01-08    74.3700    73.1875    74.1250    74.5975    74.9500    74.3575
## 2020-01-09    74.2900    74.3700    73.1875    75.7975    74.5975    74.9500
## 2020-01-10    76.5500    74.2900    74.3700    77.4075    75.7975    74.5975
## 2020-01-13    77.0625    76.5500    74.2900    77.5825    77.4075    75.7975
## 2020-01-14    77.7875    77.0625    76.5500    79.2400    77.5825    77.4075
## 2020-01-15    78.0425    77.7875    77.0625    78.1700    79.2400    77.5825
##      ...
## 2025-03-13    214.9100    217.4500    224.2200    216.9800    220.8400    227.4800
## 2025-03-14    208.4200    214.9100    217.4500    209.6800    216.9800    220.8400
## 2025-03-17    209.5800    208.4200    214.9100    213.4900    209.6800    216.9800
## 2025-03-18    209.9700    209.5800    208.4200    214.0000    213.4900    209.6800
## 2025-03-19    211.4900    209.9700    209.5800    212.6900    214.0000    213.4900
## 2025-03-20    213.7500    211.4900    209.9700    215.2400    212.6900    214.0000
## 2025-03-21    212.2200    213.7500    211.4900    214.1000    215.2400    212.6900
## 2025-03-24    211.2800    212.2200    213.7500    218.2700    214.1000    215.2400
## 2025-03-25    218.5800    211.2800    212.2200    220.7300    218.2700    214.1000
```

```
## 2025-03-26 220.0800 218.5800 211.2800 223.7500 220.7300 218.2700
```

### 3.3.5 nextCandlePosition

```
CSPInvertedHammer(AAPL) & TrendDetectionChannel(AAPL)[,"UpTrend"] & nextCandlePosition(A
```

```
##           InvertedHammer
## 2020-01-02          FALSE
## 2020-01-03          FALSE
## 2020-01-06          FALSE
## 2020-01-07          FALSE
## 2020-01-08          FALSE
## 2020-01-09          FALSE
## 2020-01-10          FALSE
## 2020-01-13          FALSE
## 2020-01-14          FALSE
## 2020-01-15          FALSE
##           ...
## 2025-03-13          FALSE
## 2025-03-14          FALSE
## 2025-03-17          FALSE
## 2025-03-18          FALSE
## 2025-03-19          FALSE
## 2025-03-20          FALSE
## 2025-03-21          FALSE
## 2025-03-24          FALSE
## 2025-03-25          FALSE
## 2025-03-26          FALSE
```

### 3.3.6 addPriceInfo

在生成的信号对象上添加价格信息，就是把 OHLC 添加上去。

```
# return detected Engulfing Pattern
CSP <- CSPEngulfing(AAPL)

# returns detected Engulfing Pattern including formation's OHLC
CSP_addPri <- addPriceInfo(AAPL, CSPEngulfing(AAPL))
```

### 3.3.7 CSPNBlended

```
CSP <- CSPNBlended(AAPL, N=3) # combine 3 candles into one
```

### 3.3.8 OHLC\_Average/HLC\_Average/HL\_Average

计算开盘-最高-最低-收盘价（OHLC）或最高-最低-收盘价（HLC）的平均值，辅助确定价格中枢或支撑阻力位 68。

```
res <- cbind(OHLC_Average(AAPL),
             HLC_Average(AAPL),
             HL_Average(AAPL))

head(res)
```

```
##           OHLC_Average HLC_Average HL_Average
## 2020-01-01      74.52375      74.67833      74.47375
## 2020-01-02      74.47875      74.54250      74.63500
## 2020-01-05      74.14375      74.37583      74.08875
## 2020-01-06      74.78813      74.73083      74.79750
## 2020-01-07      75.12188      75.39917      75.20000
## 2020-01-08      77.09375      77.18833      77.07875
```

### 3.3.9 CandleBodyLength/CandleLength

量化蜡烛实体或整体长度，用于形态筛选（如区分十字星或长实体）。



```
Len <- CandleLength(AAPL)
Bo_Len <- CandleBodyLength(AAPL)
```

## 4 投资信号生成策略示例

### 4.1 反转信号组合

倒锤子线 + 成交量放大：若 CSPInvertedHammer 检测到形态且成交量增加，可视为买入信号。

吞没形态 + 趋势线突破：CSPEngulfing 结合价格突破下降趋势线，确认反转做多。

### 4.2 趋势延续策略

三白兵 + 均线多头排列：CSPThreeWhiteSoldiers 与 TrendDetectionSMA 显示均线向上发散时，加仓持有。风控辅助

长阴线 + 超买指标：CSPNLongBlackCandles 连续出现且 RSI>80 时，提示减仓。

```
# 看涨吞没 + 放量 + 突破20日均线
engulfing <- CSPEngulfing(ohlc_data)
volume_cond <- Volume(ohlc_data) > SMA(Volume(ohlc_data), 10)
ma_break <- Close(ohlc_data) > SMA(Close(ohlc_data), 20)
buy_signal <- engulfing & volume_cond & ma_break
```