

R 语言量化投资系列之 TTR 包简介

邓一硕

目录

1 引言	2
2 TTR 优势简介	3
2.1 功能众多的函数	3
2.2 灵活变更计算方法	5
2.3 支持多种时序格式	5
2.4 高性能	6
2.5 生态兼容性强	7
2.6 可靠性验证	7
3 TTR 包功能介绍	7
3.1 TTR 包与技术指标类函数	7
3.2 数据处理类函数	157
3.3 滚动窗口类函数	166
4 与 quantmod 包及 quantstrat 包的协作	173
4.1 与 quantmod 包协作进行金融数据可视化	173
4.2 与 quantstrat 包协作开发量化投资策略	185

1 引言

TTR包 (Technical Trading Rules) 是 R 语言中专注于金融技术分析与量化交易的经典工具包，由Joshua Ulrich 开发并维护。该包集成了百余种技术指标的计算方法，为股票、期货、加密货币等金融资产的数据分析提供了标准化、高效率的解决方案，广泛应用于投资策略开发、市场规律研究及交易信号生成等领域。

TTR 的核心功能在于其丰富的技术指标库。它覆盖了从基础趋势指标（如简单移动平均线 SMA、布林带 BBands）到复杂震荡指标（如相对强弱指数 RSI、随机振荡器 stochastic）的全系列工具，同时还支持成交量分析（如能量潮 OBV）和波动率测算（如平均真实波幅 ATR）。这些函数均经过性能优化，能够快速处理大规模历史数据，满足高频交易和批量回测的需求。

TTR 的设计注重灵活性与兼容性。其函数支持 xts、data.frame 等多种数据结构，并可无缝衔接 quantmod 包的数据获取功能，形成“数据下载—指标计算—可视化—策略回测”的一体化分析流程。例如，用户可通过 getSymbols 获取实时数据后，直接调用 RSI() 计算指标，再通过 chartSeries 进行图表叠加，快速验证交易逻辑。

此外，TTR 还为高级用户提供了底层计算工具（如滚动窗口函数 runMax、runSum），支持自定义指标开发。无论是传统技术分析还是机器学习特征工程，均可借助其模块化函数构建复杂模型。

作为开源工具，TTR 凭借其可靠性已被全球金融机构和学术研究广泛采用。无论是初学者的策略探索，还是专业团队的量化系统搭建，TTR 均为金融数据分析提供了高效、可复现的解决方案，是 R 语言在金融领域应用的重要基石。

安装 TTR 包的代码如下：

```
devtools::install_github("https://github.com/joshuaulrich/TTR")
```

下载本手册需要的数据：

```
getSymbols("TSLA")
```

2 TTR 优势简介

2.1 功能众多的函数

TTR 包集成百余种经典技术指标（如 RSI、MACD、布林带等），覆盖趋势跟踪、震荡分析、波动率测算、成交量验证等场景，满足从基础研究到高频交易的多样化需求。

2.1.1 TTR 包函数简介

TTR 包中的函数大致可以分为技术指标类函数、我们分类来看一看 TTR 包中的函数。

2.1.1.1 技术指标类 技术指标类函数主要用来计算常见的技术指标，具体函数如下：

类别	函数名	描述
移动平均	SMA()	简单移动平均
	EMA()	指数移动平均
	DEMA()	双指数移动平均
	ZLEMA()	零滞后指数移动平均线
	WMA()	加权移动平均
	EVWMA()	指数加权移动平均线
	VWAP()	成交量加权平均价
	HMA()	Hull 移动平均线
	ALMA()	Arnaud Legoux 高级移动平均线
趋势强度	T3()	Tim Tillson 高级移动平均线
	ADX()	平均趋向指数（趋势强度）
	CCI()	商品通道指数
震荡指标	SAR()	抛物线止损指标
	RSI()	相对强弱指数
	MACD()	指数平滑异同平均线

类别	函数名	描述
波动率	<code>stochastic()</code>	随机震荡指标
	<code>WPR()</code>	威廉百分比指标
	<code>CMO()</code>	钱德动量摆动指标
	<code>BBands()</code>	布林带 (波动范围)
	<code>ATR()</code>	平均真实波幅
	<code>chaikinVolatility()</code>	波动率测算
成交量分析	<code>OBV()</code>	能量潮指标
	<code>AD()</code>	累积/派发线
	<code>CLV()</code>	收盘价位置指标

2.1.1.2 数据处理类

类别	函数名	描述
数据验证	<code>is.OHLC()</code>	验证 OHLC 数据结构
	<code>has.OHLC()</code>	检查是否包含 OHLC 字段
	<code>has.Ad()</code>	检查调整后价格字段
数据转换	<code>adjustOHLC()</code>	调整 OHLC 数据 (如拆分/分红处理)
	<code>Delt()</code>	计算价格差值或变动率
极值识别	<code>findPeaks()</code>	识别价格峰值
	<code>valleys()</code>	识别价格低谷

2.1.1.3 可视化辅助类

函数名	描述
<code>addBBands()</code>	在图表中添加布林带
<code>addRSI()</code>	叠加相对强弱指数图层
<code>addMACD()</code>	添加 MACD 指标图层

2.1.1.4 扩展计算工具类

函数名	描述
runSum()	滚动窗口求和
runMin()/runMax()	滚动窗口最小/最大值
TR()	计算真实波幅 (ATR 的基础)
ROC()	计算变动率 (Rate of Change)

2.2 灵活变更计算方法

```

#### 默认 RSI 指标
rsi <- RSI(Cl(TSLA), n=14)
#### 经加权成交量调整后的 RSI 指标
rsi <- RSI(Cl(TSLA), n=14, maType="WMA", wts=Vo(TSLA))
#### 经不同的向上/向下移动平均调整后的 RSI 指标
rsi <- RSI(Cl(TSLA),
            maType=list(
                maUp=list(
                    EMA,
                    n=14),
                maDown=list(
                    WMA,
                    n=16,
                    wts=1:16)
            )
        )
    )

```

2.3 支持多种时序格式

函数设计简洁直观，支持 xts、data.frame 等通用数据结构，即便非编程背景的金融从业者亦可快速上手，同时满足专业开发者的深度定制需求。TTR

可以处理以下数据对象：

- zoo / xts
- timeSeries
- ts
- its
- irts
- fts
- data.frame
- matrix

```
class(RSI(Cl(TSLA),2))

## [1] "xts" "zoo"

class(RSI(as.timeSeries(Cl(TSLA)),2))

## [1] "timeSeries"
## attr(,"package")
## [1] "timeSeries"

class(RSI(as.zoo(Cl(TSLA)),2))

## [1] "zoo"
```

2.4 高性能

底层采用 C/C++ 优化算法，支持高频数据快速处理；提供 runSum、runMax 等滚动计算函数，便于用户灵活开发自定义指标，适应复杂策略的迭代需求。

```
# 看看 TTR 包中最慢的编译函数
# 抛物线止损翻转指标
# C
system.time({S <- SAR(TSLA[,c('High','Low')])})

##      user    system elapsed
```

```
##    0.002   0.000   0.002
```

看看处理大数据时候的性能

```
# P 抛物线止损翻转指标, 600 万个观测值
x <- .xts(cumprod(1+rnorm(6e6)/1e5), 1:6e6)
x <- merge(Low=x, High=x*(1+runif(6e6)/100))
# C
system.time({S <- SAR(x[, c('High', 'Low')])})
```

```
##      user    system elapsed
##  11.035   0.422  12.863
```

2.5 生态兼容性强

无缝衔接 quantmod（数据获取）、PerformanceAnalytics（策略回测）等金融分析包，形成“数据-计算-可视化-验证”的全链路工具链，极大提升量化研究效率。

2.6 可靠性验证

作为开源社区经十余年维护的成熟工具，其计算结果与主流交易平台（如 TradingView）高度一致，被全球金融机构及学术论文广泛引用，确保分析结果的可信度。

3 TTR 包功能介绍

3.1 TTR 包与技术指标类函数

3.1.1 移动平均指标

移动平均指标通过计算价格数据的阶段性均值，有效平滑短期波动并提取长期趋势信号，是技术分析中判断趋势方向、支撑阻力位的核心工具。不同

类型均线通过差异化权重分配，可适应不同交易场景的灵敏度需求。

3.1.1.1 简单移动平均（SMA） 指标含义

简单移动平均指标是将过去若干个周期的数据进行算术平均，得到一个平滑的数值序列，用来代表数据在该时间段内的平均水平，帮助投资者或分析师观察数据的长期趋势。

它可用作如下判断：

- 趋势判断：通过观察简单移动平均线的走势，可以判断价格的长期趋势。当价格在移动平均线上方运行时，通常表示市场处于上升趋势；当价格在移动平均线下方运行时，一般意味着市场处于下降趋势。而且，短期移动平均线向上穿过长期移动平均线，被视为买入信号，称为“黄金交叉”；反之，短期移动平均线向下穿过长期移动平均线，是卖出信号，称为“死亡交叉”。
- 支撑与阻力位：简单移动平均线也可以作为支撑位或阻力位。在上升趋势中，移动平均线可以作为价格回调时的支撑线，当价格回落到移动平均线附近时，可能会受到支撑而继续上涨；在下降趋势中，移动平均线则成为价格反弹的阻力线，价格上涨到移动平均线附近时，可能会遇到阻力而再次下跌。

数学公式

简单移动平均指标的计算公式如下：

$$SMA_t = (P_t + P_{t-1} + \dots + P_{t-n+1})/n$$

其平等对待窗口期内所有价格，反映均衡趋势水平，适合识别长期趋势方向。

代码实现及可视化

TTR 提供了 SMA() 函数来计算简单移动平均指数：

```
sma_20 <- SMA(Cl(TSLA), n = 20) # 计算 20 日 SMA
chartSeries(TSLA, TA = "addSMA(20)", subset="2025::", theme = "white")
```



AI 引导词

“介绍简单移动平均线的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.1.2 指数移动平均线（EMA） 指标含义

指数移动平均线（Exponential Moving Average, EMA）是一种技术分析中常用的趋势跟踪指标，与简单移动平均线（SMA）相比，EMA 给予近期价格更高的权重，对价格变化更加敏感。EMA 广泛应用于：

- 识别市场趋势方向
- 作为动态支撑/阻力位
- 生成交易信号（如金叉/死叉）
- 作为其他指标的基础（如 MACD）

数学公式

EMA 计算公式采用递归形式:

$$EMA_t = \begin{cases} P_0 & \text{当 } t = 0 \\ \alpha \cdot P_t + (1 - \alpha) \cdot EMA_{t-1} & \text{当 } t > 0 \end{cases}$$

$$\alpha = \frac{2}{n + 1}$$

其中:

- P_t 为时间 t 的价格
- n 为平滑周期 (常用 12 日或 26 日)
- α 为平滑因子

代码实现及可视化

```
# 计算 12 日和 26 日 EMA
# 计算 12 日和 26 日 EMA
EMA_12 <- EMA(Cl(TSLA), n = 12)
EMA_26 <- EMA(Cl(TSLA), n = 26)

# 查看最后 5 日数据
tail(TSLA, 5)

##           Open   High    Low  Close       Volume Adjusted
## 2025-04-25 261.69 286.85 259.63 284.95 8.278599e-316 284.95
## 2025-04-28 288.98 294.86 272.42 285.88 7.496547e-316 285.88
## 2025-04-29 285.50 293.32 279.47 292.03 5.380701e-316 292.03
## 2025-04-30 279.90 284.45 270.78 282.16 6.371525e-316 282.16
## 2025-05-01 280.01 290.87 279.81 280.52 4.923809e-316 280.52

chartSeries(TSLA, TA = "addEMA(12)", subset="2025::", theme = "white")
```



常见 EMA 交易策略：

- 趋势判断：价格在 EMA 之上为上升趋势，反之为下降趋势
- 交叉信号：
 - 当短期 EMA 上穿长期 EMA（金叉）→ 买入信号
 - 当短期 EMA 下穿长期 EMA（死叉）→ 卖出信号
- EMA 斜率：向上倾斜表示趋势增强，向下倾斜表示趋势减弱

注意：EMA 作为滞后指标，在震荡市中可能产生假信号，建议结合其他指标使用。

AI 引导词

“介绍指数移动平均线的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.1.3 双指数移动平均 (DEMA) 指标含义

DEMA (Double Exponential Moving Average) 由 Patrick Mulloy 于 1994 年提出，相比传统 EMA：

- 减少滞后：通过双重平滑处理降低延迟效应
- 噪声过滤：平衡趋势跟踪与噪声消除
- 动态支撑阻力：快速响应价格变化，捕捉短期趋势
- 信号灵敏度：交叉点比 SMA/EMA 产生更早的交易信号

数学公式

DEMA 计算分为三个步骤：

1. 计算 n 日 EMA

$$EMA_t(n) = P_t \times \alpha + EMA_{t-1} \times (1 - \alpha)$$

其中 $\alpha = 2/(n + 1)$ 。

2. 计算 EMA 的 EMA

$$EMA_{EMA}(n) = EMA(EMA(n))$$

3. 最终 DEMA 计算

$$DEMA(n) = 2 \times EMA(n) - EMA_{EMA}(n)$$

其中：

- $\alpha = 2/(n + 1)$
- n 为周期参数（默认 20）
- P_t 为 t 期价格

代码实现及可视化

```
dema.20 <- DEMA(Ad(TSLA), 20)
tail(dema.20, 5)
```

```
##          DEMA
## 2025-04-25 248.0333
## 2025-04-28 254.3010
## 2025-04-29 260.8338
```

```
## 2025-04-30 264.6740
## 2025-05-01 267.6786
```

AI 引导词

“介绍双指数移动平均线的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包) 的 R 语言实例, 结果请返回 RMarkdown 代码。”

3.1.1.4 零滞后指数移动平均线 (ZLEMA) 指标含义

零滞后指数移动平均线 (Zero Lag EMA, ZLEMA) 由 John Ehlers 和 Ric Way 开发, 通过数据平移校正解决传统移动平均的滞后问题。核心特点:

- 保留 EMA 对新数据加权的特性
- 通过向前调整数据消除相位滞后
- 比 DEMA 更直接处理滞后问题
- 常用参数周期为 20/50/100 日

数学公式

修正数据滞后分两步实现:

1. 计算价格数据的中移

$$P'_t = P_t + (P_t - P_{t-k})$$

其中 $k = \frac{n-1}{2}$ 为滞后期数

2. 对调整后数据计算 EMA

$$ZLEMA(n) = EMA(P', n)$$

代码实现及可视化

```
zlema.20 <- ZLEMA(Ad(TSLA), 20)
```

使用建议:

- 趋势市中使用 ZLEMA 可更快捕捉突破
- 震荡行情中建议结合其他滤波器使用
- 周期参数需根据资产波动率调整
- 与成交量指标结合可验证信号强度

AI 引导词

“介绍零滞后指数移动平均线的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包) 的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.1.5 加权移动平均线 (WMA) 指标含义

加权移动平均线 (Weighted Moving Average, WMA) 通过为不同时期数据分配差异化权重，相比简单移动平均线 (SMA)：

- 赋予近期价格更高权重
- 对趋势变化更敏感
- 保留价格序列的波动特征
- 常用参数周期为 10/20/50 日

数学公式

对于 n 日窗口期的 WMA 计算：

$$WMA_t = \frac{\sum_{i=0}^{n-1} (n-i) \cdot P_{t-i}}{\sum_{i=0}^{n-1} (n-i)}$$

其中：

- P_{t-i} 表示第 $t-i$ 期的价格
- 分子为各期价格与其权重的乘积和
- 分母为权重总和 $\frac{n(n+1)}{2}$

代码实现及可视化

```
wma.20 <- WMA(Ad(TSLA), n = 20, wts = 1:20)
```

应用建议：

- 短期交易中 WMA 突破策略更有效
- 结合波动率指标 (如 ATR) 过滤虚假信号
- 多时间框架组合使用 (如 5 分钟 WMA+ 日线 WMA)
- 参数优化范围建议 8-30 周期

AI 引导词

“介绍加权移动平均线的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包) 的 R 语言实例, 结果请返回 RMarkdown 代码。”

3.1.1.6 指数加权移动平均线 (EVWMA) 指标含义

EVWMA (Exponentially Weighted Moving Average with Volume Adjustment) 是一种结合了交易量因素的指数加权移动平均线。与传统 EWMA 相比, EVWMA 在计算时考虑了交易量的影响, 使价格变动在交易量较大时具有更高的权重, 从而更敏感地反映市场的实际供需情况。这种指标特别适用于金融市场分析, 可以帮助识别趋势、判断价格支撑和阻力位, 以及生成交易信号。

数学公式

EVWMA 的计算公式如下:

$$\text{EVWMA}_t = \frac{\sum_{i=0}^{n-1} w_i \cdot P_{t-i} \cdot V_{t-i}}{\sum_{i=0}^{n-1} w_i \cdot V_{t-i}}$$

其中:

- $w_i = \alpha(1 - \alpha)^i$ 是指数衰减权重
- $\alpha = \frac{2}{n+1}$ 是平滑因子
- n 是计算周期
- P_{t-i} 是 $t - i$ 时刻的价格
- V_{t-i} 是 $t - i$ 时刻的交易量

代码实现及可视化

```
evwma.20 <- EVWMA(Ad(TSLA), Vo(TSLA), 20)
tail(evwma.20, 5)
```

```
##                  EVWMA
## 2025-04-25 265.3686
## 2025-04-28 266.4590
## 2025-04-29 267.4434
## 2025-04-30 268.1185
## 2025-05-01 268.5765
```

AI 引导词

“介绍指数加权移动平均线的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包) 的 R 语言实例, 结果请返回 RMarkdown 代码。”

3.1.1.7 交易量加权的价格平均 (VWAP) 指标含义

VWAP (Volume Weighted Average Price) 是一种基于交易量加权的价格指标, 它计算了某段时间内证券价格的加权平均值, 其中每个价格的权重由该价格对应的交易量决定。VWAP 常用于机构投资者和日内交易者, 主要有以下用途:

- 衡量交易执行质量: 机构投资者常将交易成本与 VWAP 比较, 评估交易执行的优劣。
- 识别价格趋势: 若当前价格高于 VWAP, 表明市场处于上升趋势; 反之则为下降趋势。
- 确定支撑和阻力位: VWAP 常被视为动态支撑或阻力位, 尤其在日内交易中。
- 生成交易信号: 当价格突破 VWAP 或 VWAP 方向改变时, 可能产生交易信号。

数学公式

VWAP 的计算公式如下:

$$\text{VWAP} = \frac{\sum_{i=1}^n (P_i \times V_i)}{\sum_{i=1}^n V_i}$$

其中：

- P_i 是第 i 个时间点的价格（通常使用收盘价或中间价）
- V_i 是第 i 个时间点的交易量
- n 是计算周期内的总交易次数

代码实现及可视化

```
vwap.20 <- VWAP(Ad(TSLA), Vo(TSLA), n = 20)
tail(vwap.20)
```

```
##          VWAP
## 2025-04-24 253.8062
## 2025-04-25 254.5411
## 2025-04-28 255.8163
## 2025-04-29 257.0520
## 2025-04-30 257.6093
## 2025-05-01 256.4721
```

AI 引导词

“介绍交易量加权移动平均线的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包) 的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.1.8 HULL 移动平均线 (HMV) 指标含义

HMA (Hull Moving Average) 是一种改进型的移动平均线，由 Alan Hull 在 2005 年开发。它通过加权计算和双重平滑处理，显著减少了传统移动平均线的滞后问题，同时保留了良好的抗噪声特性。HMA 特别适合用于识别趋势方向、判断支撑阻力位以及生成交易信号。与其他移动平均线相比，HMA 能够更快地适应价格变化，更准确地反映当前市场趋势。

数学公式

HMA 的计算公式如下：

- 首先计算加权移动平均 (WMA)：

$$\text{WMA}(n) = \frac{\sum_{i=0}^{n-1} (n-i) \cdot P_{t-i}}{\sum_{i=1}^n i}$$

- 计算两个不同周期的 WMA：

- 短期 WMA: $\text{WMA}_1 = \text{WMA}(n/2)$
- 长期 WMA: $\text{WMA}_2 = \text{WMA}(n)$

- 计算差值：

$$\text{Diff} = 2 \cdot \text{WMA}_1 - \text{WMA}_2$$

- 对差值进行再次 WMA 计算，得到最终的 HMA：

$$\text{HMA}(n) = \text{WMA}(\text{Diff}, \sqrt{n})$$

其中：

- P_{t-i} 是 $t-i$ 时刻的价格
- n 是计算周期

代码实现及可视化

```
hma.20 <- HMA(Ad(TSLA), n = 20)
tail(hma.20)
```

```
##          HMA
## 2025-04-24 241.0354
## 2025-04-25 246.7743
## 2025-04-28 254.8626
## 2025-04-29 264.4605
## 2025-04-30 273.1926
## 2025-05-01 280.2056
```

AI 引导词

“介绍 HULL 移动平均线的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包) 的 R 语言实例, 结果请返回 RMarkdown 代码。”

3.1.1.9 Arnaud Legoux 高级移动平均线 (ALMA) 指标含义

ALMA (Arnaud Legoux Moving Average) 是一种由金融工程师 Arnaud Legoux 开发的高级移动平均线。它结合了高斯分布权重和傅里叶变换的特性, 旨在提供比传统移动平均线更平滑、更少滞后的价格趋势追踪效果。ALMA 特别适合在噪声较大的市场环境中使用, 能够有效过滤短期价格波动, 同时快速响应真实的趋势变化。该指标常用于识别市场趋势方向、生成交易信号以及确定支撑和阻力位。

数学公式

ALMA 的计算公式如下:

$$\text{ALMA}_t = \sum_{i=0}^{n-1} w_i \cdot P_{t-i}$$

其中权重 w_i 的计算方式为:

$$w_i = \frac{\exp\left(-\frac{(i-o)^2}{2\sigma^2}\right)}{\sum_{j=0}^{n-1} \exp\left(-\frac{(j-o)^2}{2\sigma^2}\right)}$$

其中:

- P_{t-i} 是 $t - i$ 时刻的价格
- n 是计算周期
- o 是偏移量参数, 通常取值为 0.85 , 控制权重中心位置
- σ 是平滑参数, 计算公式为 $\sigma = \frac{6}{s}$, 其中 s 是缩放因子 (通常为 2)

代码实现及可视化

```
alma.9 <- ALMA(Ad(TSLA), n = 9, offset = 0.85, sigma = 6)
tail(alma.9)

##                      ALMA
## 2025-04-24 241.6174
## 2025-04-25 250.6815
## 2025-04-28 262.4090
## 2025-04-29 274.1026
## 2025-04-30 281.8276
## 2025-05-01 284.7129
```

AI 引导词

“介绍 Arnaud Legoux 高级移动平均线 (ALMA) 的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包)的 R 语言实例,结果请返回 RMarkdown 代码。”

3.1.1.10 Tim Tillson 高级移动平均线 (T3) 指标含义

Tim Tillson's T3 指标是一种由 Tim Tillson 开发的高级移动平均线，它通过对价格数据进行多重指数平滑处理，旨在提供比传统移动平均线更平滑、更少滞后的趋势追踪效果。T3 指标特别适合在波动较大的市场环境中使用，能够有效过滤噪声，同时快速响应真实的趋势变化。该指标常用于识别市场趋势方向、生成交易信号以及确定支撑和阻力位。

数学公式

T3 指标的计算公式如下：

1. 首先进行三重 EMA 计算：

$$\begin{aligned}
 EMA1_t &= \alpha \cdot Price_t + (1 - \alpha) \cdot EMA1_{t-1} \\
 EMA2_t &= \alpha \cdot EMA1_t + (1 - \alpha) \cdot EMA2_{t-1} \\
 EMA3_t &= \alpha \cdot EMA2_t + (1 - \alpha) \cdot EMA3_{t-1} \\
 EMA4_t &= \alpha \cdot EMA3_t + (1 - \alpha) \cdot EMA4_{t-1} \\
 EMA5_t &= \alpha \cdot EMA4_t + (1 - \alpha) \cdot EMA5_{t-1} \\
 EMA6_t &= \alpha \cdot EMA5_t + (1 - \alpha) \cdot EMA6_{t-1}
 \end{aligned}$$

2. 然后通过加权组合这些 EMA 值得到最终的 T3 指标:

$$T_3 = c_1 \cdot EMA6_t + c_2 \cdot EMA5_t + c_3 \cdot EMA4_t + c_4 \cdot EMA3_t$$

其中:

- $\alpha = \frac{2}{p+1}$ 是平滑因子
- p 是计算周期
- $c_1 = -b^3$, $c_2 = 3b^2 + 3b^3$, $c_3 = -6b^2 - 3b - 3b^3$, $c_4 = 1 + 3b + b^3 + 3b^2$
- b 是 volume factor, 通常取值为 0.7

代码实现及可视化

```
T3 <- function(x, n=10, v=1) DEMA(DEMA(DEMA(x,n,v),n,v),n,v)
t3.10 <- T3(Cl(TSLA), n = 10, v = 1)
tail(t3.10)
```

```
##          DEMA
## 2025-04-24 242.2645
## 2025-04-25 243.1396
## 2025-04-28 246.1517
## 2025-04-29 251.1084
## 2025-04-30 257.1084
## 2025-05-01 263.3850
```

AI 引导词

“介绍 Tim Tillson 高级移动平均线（T3）的意义、对应公式的 Latex 代码

和基于数据 TSLA 及 TTR 包) 的 R 语言实例, 结果请返回 RMarkdown 代码。”

3.1.2 趋势强度类指标

3.1.2.1 万无一失指标 (KST) 指标含义

KST 指标 (Know Sure Thing) 由 Martin Pring 开发, 是一种多周期动量指标, 用于识别中长期趋势和潜在的反转点。它通过组合多个不同周期的 ROC (Rate of Change) 指标, 形成一个综合性的动量指标, 能够有效减少短期波动的干扰。

指标的核心特点如下:

1. 多周期分析: 同时考虑不同时间周期的动量, 提供更全面的市场视角
2. 趋势确认: 与价格走势配合使用, 可有效确认趋势的强度和持续性
3. 反转信号: 指标与价格的背离可提前预警潜在的趋势反转
4. 交叉策略: KST 线与信号线的交叉可生成买卖信号

数学公式

KST 指标的计算分为四个主要步骤:

1. 计算不同周期的 ROC 值

$$ROC(n) = \left(\frac{Price_t}{Price_{t-n}} - 1 \right) \times 100$$

其中:

- n 为周期长度, 通常使用 10、15、20、30 四个周期
- $Price_t$ 为当前价格
- $Price_{t-n}$ 为 n 周期前的价格

2. 对各 ROC 值进行移动平均

$$\text{Smoothed } ROC(n) = SMA(ROC(n), m)$$

其中：

- m 为移动平均的周期，通常分别为 10、10、10、15
- 对平滑后的 ROC 值进行加权求和

$$KST = \sum_{i=1}^4 (Smoothed\ ROC(n_i) \times w_i)$$

其中：

- $n_1 = 10, n_2 = 15, n_3 = 20, n_4 = 30$
- $w_1 = 1, w_2 = 2, w_3 = 3, w_4 = 4$ (权重系数)

- 计算信号线

$$Signal = SMA(KST, s)$$

其中：

- s 为信号线的周期，通常为 9

```
# 使用 TTR 包计算 KST 指标
tsla_kst <- KST(Cl(TSLA)) # 默认参数: ROC1=10, ROC2=15, ROC3=20, ROC4=30,
# SMA1=10, SMA2=10, SMA3=10, SMA4=15,
# SCAL1=1, SCAL2=2, SCAL3=3, SCAL4=4,
# signal=9
#
# 将结果合并到数据框中
tsla_data <- data.frame(
  Date = index(TSLA),
  Close = as.numeric(Cl(TSLA)),
  KST = as.numeric(tsla_kst[,1]), # KST 线
  Signal = as.numeric(tsla_kst[,2]) # 信号线
)
```

```
# 移除 NA 值
tsla_data <- na.omit(tsla_data)

# 查看计算结果
head(tsla_data)

##           Date   Close      KST     Signal
## 53 2020-03-18 24.0813 -216.2053  12.71534
## 54 2020-03-19 28.5093 -269.9447 -38.98069
## 55 2020-03-20 28.5020 -320.8956 -91.09628
## 56 2020-03-23 28.9527 -364.0928 -143.71859
## 57 2020-03-24 33.6667 -396.3222 -195.66436
## 58 2020-03-25 35.9500 -418.1697 -245.48380

# 创建价格图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "特斯拉 (TSLA) 收盘价",
       x = "日期",
       y = "收盘价 ($)") +
  theme_minimal(base_family = "SimHei")

# 创建 KST 指标图表
p2 <- ggplot(tsla_data, aes(x = Date)) +
  geom_line(aes(y = KST), color = "red", size = 1) +
  geom_line(aes(y = Signal), color = "green", size = 1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray", alpha = 0.7) +
  labs(title = "KST 指标",
       x = "日期",
       y = "指标值") +
  theme_minimal(base_family = "SimHei")

# 使用 patchwork 组合图表
```



AI 引导词

“介绍 KST 指标的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.2 平均方向指数 (ADX) 指标含义

平均方向指数 (Average Directional Index, ADX) 由 J. Welles Wilder 在 1978 年提出，是一种衡量市场趋势强度的技术指标。ADX 不判断趋势方向，只评估趋势的强弱程度，其值越高表示趋势越强。ADX 常与方向移动指标 (+DI 和 - DI) 结合使用，形成完整的 DMI (Directional Movement Index) 系统：

- +DI (正方向指标): 衡量上升趋势的强度
- -DI (负方向指标): 衡量下降趋势的强度
- ADX (平均方向指数): 衡量整体趋势的强度

ADX 的主要应用场景：

- 判断市场是否存在明显趋势 (ADX>25 表示强趋势)
- 区分趋势市场和盘整市场
- 避免在无趋势市场中使用趋势跟踪策略
- 确认趋势反转信号

数学公式

ADX 的计算步骤

1. 计算方向移动值 (DM):

$$+DM = \begin{cases} H_t - H_{t-1}, & \text{如果 } H_t - H_{t-1} > L_{t-1} - L_t \text{ 且 } H_t - H_{t-1} > 0 \\ 0, & \text{其他情况} \end{cases}$$

$$-DM = \begin{cases} L_{t-1} - L_t, & \text{如果 } L_{t-1} - L_t > H_t - H_{t-1} \text{ 且 } L_{t-1} - L_t > 0 \\ 0, & \text{其他情况} \end{cases}$$

2. 计算真实范围 (TR):

$$TR = \max(H_t - L_t, |H_t - C_{t-1}|, |L_t - C_{t-1}|)$$

3. 计算方向指标 (DI):

$$+DI = \frac{EMA(+DM, n)}{EMA(TR, n)} \times 100$$

$$-DI = \frac{EMA(-DM, n)}{EMA(TR, n)} \times 100$$

4. 计算方向移动指数 (DX):

$$DX = \frac{|+DI - -DI|}{+DI + -DI} \times 100$$

5. 计算平均方向指数 (ADX):

$$ADX = EMA(DX, n)$$

其中：

- EMA 表示指数移动平均
- n 是计算周期 (默认 14)

代码实现及可视化

```
adx.14 <- ADX(TSLA[, c("High", "Low", "Close")], n = 14)
tail(adx.14)
```

	DI _p	DIn	DX	ADX
## 2025-04-24	21.57461	23.47878	4.226464	18.99854
## 2025-04-25	28.87952	21.28879	15.130534	18.72226
## 2025-04-28	29.39865	19.66728	19.833261	18.80161
## 2025-04-29	27.98202	18.71957	19.833261	18.87530
## 2025-04-30	25.91838	20.35491	12.023058	18.38586
## 2025-05-01	27.19375	19.54691	16.360145	18.24116

ADX 指标的常见解读：

1. $ADX > 25$: 市场存在明显趋势，可考虑使用趋势跟踪策略
2. $ADX < 20$: 市场处于盘整状态，建议使用区间交易策略
3. $+DI > -DI$: 上升趋势占主导，看涨信号
4. $-DI > +DI$: 下降趋势占主导，看跌信号
5. ADX 达到峰值后开始下降：可能预示当前趋势即将结束

需要注意的是， ADX 只能衡量趋势强度，不能预测趋势方向，因此建议与其他指标结合使用，如 MACD、RSI 等，以提高交易决策的准确性。

AI 引导词

“介绍平均方向指数的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.3 黎明曙光指标 (Aroon) 指标含义

Aroon 指标由 Tushar Chande 在 1995 年开发，用于衡量价格趋势的强度和方向，特别擅长识别趋势的早期阶段和潜在反转点。“Aroon”一词源于梵语，意为“黎明曙光”，象征该指标能够提前预警趋势变化。

Aroon 系统包含两个核心指标：

- Aroon-Up：衡量上涨趋势的强度，计算自周期内最高价以来的天数占比
- Aroon-Down：衡量下跌趋势的强度，计算自周期内最低价以来的天数占比
- Aroon Oscillator：Aroon-Up 与 Aroon-Down 的差值，直观反映多空力量对比

Aroon 指标的主要特点：

- 数值范围从 0 到 100，数值越高表示趋势越强
- 对新趋势的识别速度快于传统指标（如 ADX）
- 能够有效区分趋势市场和盘整市场
- 交叉信号可用于生成交易策略

数学公式

Aroon 指标的计算公式

1. Aroon-Up

$$\text{Aroon-Up} = \left(\frac{n - \text{Days Since H}}{n} \right) \times 100$$

2. Aroon-Down

$$\text{Aroon-Down} = \left(\frac{n - \text{Days Since L}}{n} \right) \times 100$$

3. Aroon Oscillator

$$\text{Aroon Oscillator} = \text{Aroon-Up} - \text{Aroon-Down}$$

其中：

- n 是计算周期（默认 20 天）
- Days Since High 是自周期内最高价出现以来的天数
- Days Since Low 是自周期内最低价出现以来的天数

代码实现及可视化

```

aroon.20 <- aroon(TSLA[,c("High","Low")],n=20)
tail(aroon.20, 5)

##          aroonUp aroonDn oscillator
## 2025-04-25      0     35     -35
## 2025-04-28    100     30      70
## 2025-04-29     95     25      70
## 2025-04-30     90     20      70
## 2025-05-01     85     15      70

temp <- na.omit(data.frame(coredata(merge(x=TSLA,y=aroon.20)),Date = as.Date(index(TSLA)))
tsla_data <- temp[c("Date","Close","aroonUp","aroonDn","oscillator")]

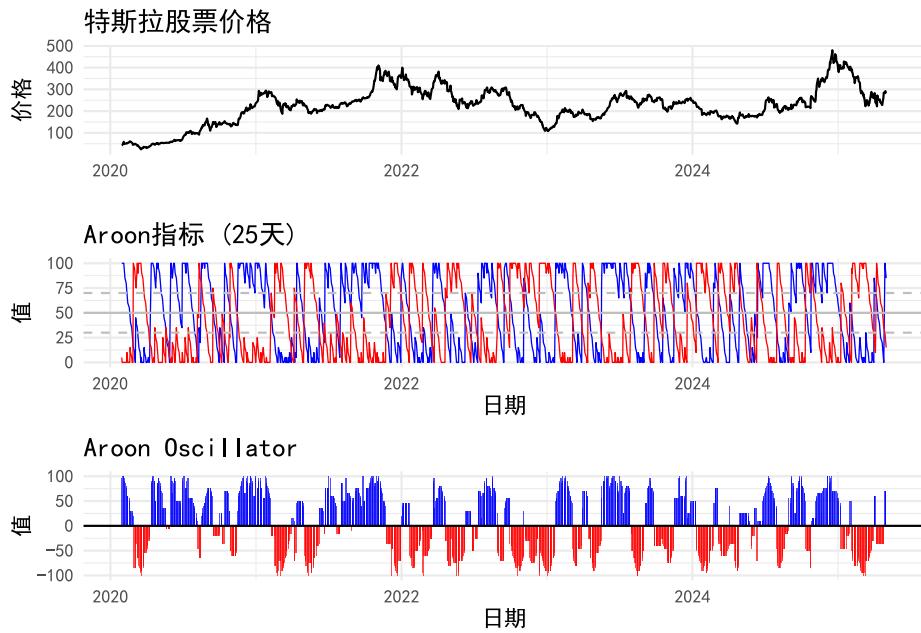
# 可视化分析
# 创建价格图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "black") +
  labs(title = "特斯拉股票价格", y = "价格", x = "") +
  theme_minimal(base_family = "SimHei")

# 创建 Aroon 指标图表
p2 <- ggplot(tsla_data, aes(x = Date)) +
  geom_line(aes(y = aroonUp), color = "blue", linewidth = .3) +
  geom_line(aes(y = aroonDn), color = "red", linewidth = .3) +
  geom_hline(yintercept = 70, linetype = "dashed", color = "gray") +
  geom_hline(yintercept = 50, linetype = "solid", color = "gray") +
  geom_hline(yintercept = 30, linetype = "dashed", color = "gray") +
  labs(title = "Aroon 指标 (25 天)", y = "值", x = "日期") +
  theme_minimal(base_family = "SimHei")

```

```
# 创建 Aroon Oscillator 图表
p3 <- ggplot(tsla_data, aes(x = Date, y = oscillator)) +
  geom_bar(stat = "identity", fill = ifelse(tsla_data$oscillator >= 0, "blue", "red"),
  geom_hline(yintercept = 0, linetype = "solid", color = "black") +
  labs(title = "Aroon Oscillator", y = "值", x = "日期") +
  theme_minimal(base_family = "SimHei")

# 组合图表
p1 / p2 / p3
```



Aroon 指标的常见交易信号：

1. Aroon-Up > 70 且 Aroon-Down < 30: 强上升趋势信号
2. Aroon-Down > 70 且 Aroon-Up < 30: 强下降趋势信号
3. Aroon-Up 从下向上穿越 Aroon-Down: 看涨信号
4. Aroon-Down 从下向上穿越 Aroon-Up: 看跌信号
5. Aroon Oscillator > 0: 多头力量占优
6. Aroon Oscillator < 0: 空头力量占优

7. Aroon-Up 和 Aroon-Down 均低于 50: 市场处于盘整状态

Aroon 指标在趋势识别方面具有独特优势，尤其适合捕捉中长期趋势的启动阶段。但在使用时应注意，短期市场波动可能导致 Aroon 信号频繁变化，建议结合其他指标（如 MACD、RSI）进行综合分析。

AI 引导词

“介绍黎明曙光的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.4 顾比复合移动平均线（GMMA） 指标含义

顾比复合移动平均线 (Guppy Multiple Moving Average, GMMA) 由澳大利亚金融分析师 Daryl Guppy 在 2009 年提出，是一种多周期移动平均线系统，用于判断市场趋势、识别交易机会和确认趋势反转。与传统单一或双移动平均线不同，GMMA 同时使用两组不同周期的移动平均线，分别反映短期和长期投资者的行为：

- 短期组：由 3、5、8、10、12、15 日 EMA 组成，反映短期交易者的行为和情绪
- 长期组：由 30、35、40、45、50、60 日 EMA 组成，反映长期投资者的行为和情绪

GMMA 的核心思想：

- 当短期组均线向上穿越长期组均线时，形成看涨信号
- 当短期组均线向下穿越长期组均线时，形成看跌信号
- 均线组的发散程度反映趋势强度，收敛则可能预示趋势减弱或反转
- 长期组均线的方向决定市场主趋势

数学公式

GMMA 的计算公式

短期组移动平均线

$$\text{短期组} = \{EMA_3, EMA_5, EMA_8, EMA_{10}, EMA_{12}, EMA_{15}\}$$

长期组移动平均线

$$\text{长期组} = \{EMA_{30}, EMA_{35}, EMA_{40}, EMA_{45}, EMA_{50}, EMA_{60}\}$$

其中：

$$EMA_t = \alpha \times P_t + (1 - \alpha) \times EMA_{t-1}$$

$$\alpha = \frac{2}{n+1}$$

- P_t 是第 t 期的价格
- n 是移动平均的周期
- α 是平滑系数

代码实现及可视化

```
gmma.tsla <- GMMA(Cl(TSLA))
tail(gmma.tsla, 5)

##           short lag 3 short lag 5 short lag 8 short lag 10 short lag 12
## 2025-04-25    268.3145    260.7404    255.9960    254.6719    254.0386
## 2025-04-28    277.0972    269.1202    262.6369    260.3461    258.9372
## 2025-04-29    284.5636    276.7568    269.1687    266.1068    264.0284
## 2025-04-30    283.3618    278.5579    272.0556    269.0256    266.8179
## 2025-05-01    281.9409    279.2119    273.9366    271.1155    268.9259
##           short lag 15 long lag 30 long lag 35 long lag 40 long lag 45
## 2025-04-25    253.8643    260.6670    264.4441    268.3833    272.2497
## 2025-04-28    257.8663    262.2937    265.6350    269.2368    272.8423
## 2025-04-29    262.1367    264.2121    267.1014    270.3487    273.6766
## 2025-04-30    264.6396    265.3701    267.9380    270.9248    274.0454
## 2025-05-01    266.6247    266.3475    268.6370    271.3929    274.3269
##           long lag 50 long lag 60
## 2025-04-25    275.8912    282.1927
## 2025-04-28    276.2829    282.3136
```

```
## 2025-04-29      276.9004    282.6321  
## 2025-04-30      277.1067    282.6167  
## 2025-05-01      277.2405    282.5479
```

GMMA 的常见交易信号：

1. 看涨信号：

- 短期组均线向上穿越长期组均线
- 短期组均线发散且位于长期组均线上方
- 长期组均线开始向上发散

2. 看跌信号：

- 短期组均线向下穿越长期组均线
- 短期组均线发散且位于长期组均线下方
- 长期组均线开始向下发散

3. 趋势确认：

- 短期组和长期组均线均向上发散：强上升趋势
- 短期组和长期组均线均向下发散：强下降趋势
- 两组均线收敛并交织：盘整或趋势反转信号

GMMA 的优势在于能够同时反映不同时间周期投资者的行为，帮助交易者区分短期波动和长期趋势。在实际应用中，建议结合价格形态和其他指标（如成交量、MACD）进行综合分析，以提高交易决策的准确性。

AI 引导词

“介绍顾比复合移动平均线的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包) 的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.5 垂直水平过滤器 (VHF) 指标含义

(Vertical Horizontal Filter, VHF) 由 Adam White 在 1991 年提出，是一种用于判断市场是处于趋势状态还是盘整状态的技术指标。VHF 通过比较价格在垂直方向 (趋势) 和水平方向 (震荡) 的移动幅度，评估市场的方向性强度。

VHF 的主要作用：

- 识别趋势市场：高 VHF 值表示市场存在明确趋势
- 识别盘整市场：低 VHF 值表示市场处于横向震荡
- 避免趋势跟踪策略在盘整市场中失效
- 确认趋势强度变化：VHF 值的上升或下降反映趋势强度的变化
- 与其他指标结合使用：在高 VHF 环境中使用趋势策略，在低 VHF 环境中使用震荡策略。

VHF 指标的常见解读：

1. $VHF > 0.5$: 市场存在明确趋势，适合使用趋势跟踪策略
2. $VHF < 0.3$: 市场处于盘整状态，适合使用区间交易策略
3. VHF 在 0.3-0.5 之间：市场方向性不明确，建议谨慎操作
4. VHF 持续上升：趋势强度增强
5. VHF 达到峰值后开始下降：当前趋势可能即将结束
6. VHF 与价格形成背离：可能预示趋势反转

需要注意的是，VHF 指标的阈值可以根据不同市场特性和交易品种进行调整。在实际应用中，建议结合其他指标（如 ADX、Aroon）和价格形态进行综合分析，以提高交易决策的准确性。

数学公式

VHF 的计算公式如下：

1. 价格变动绝对值的累积和

$$\text{Sum of Absolute Changes} = \sum_{i=1}^n |P_i - P_{i-1}|$$

其中：

- P_i 是第 i 期的价格
- n 是计算周期

2. 价格波动范围

$$\text{Price Range} = \max(P_1, P_2, \dots, P_n) - \min(P_1, P_2, \dots, P_n)$$

3. VHF 值计算

$$VHF = \frac{\text{Price Range}}{\text{Sum of Absolute Changes}}$$

解释：

- VHF 值接近 1：表示市场处于强趋势状态（累积变动接近最大波动范围）
- VHF 值接近 0：表示市场处于震荡状态（累积变动远小于最大波动范围）

代码实现及可视化

```

tsla_vhf <- VHF(Cl(TSLA), n = 28)
tsla_vhf_b <- VHF(merge(Cl(TSLA),Hi(TSLA),Lo(TSLA)), n = 28)

# 将结果合并到数据框中
tsla_data <- data.frame(
  Date = as.Date(index(TSLA)),
  Close = as.numeric(Cl(TSLA)),
  VHF = as.numeric(tsla_vhf)
)

# 移除 NA 值
tsla_data <- na.omit(tsla_data)

# 查看计算结果
head(tsla_data)

##           Date   Close      VHF
## 29 2020-02-12 51.1527 0.5775206
## 30 2020-02-13 53.6000 0.5493089
## 31 2020-02-14 53.3353 0.5302611

```

```
## 32 2020-02-18 57.2267 0.4931724
## 33 2020-02-19 61.1613 0.5077785
## 34 2020-02-20 59.9607 0.5035820

# 创建价格图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "特斯拉 (TSLA) 收盘价",
       x = "日期",
       y = "收盘价 ($)") +
  theme_minimal(base_family="SimHei")

# 创建 VHF 指标图表
p2 <- ggplot(tsla_data, aes(x = Date, y = VHF)) +
  geom_line(color = "red", size = 1) +
  geom_hline(yintercept = 0.5, linetype = "dashed", color = "gray", alpha = 0.7) +
  labs(title = "垂直水平指标 (VHF) - 28 天周期",
       x = "日期",
       y = "VHF 值") +
  theme_minimal(base_family = "SimHei")

# 使用 patchwork 组合图表

p1 / p2
```



信号应用

基于 VHF 值，我们可以将市场状态分为趋势市场和震荡市场：

```
# 定义市场状态
tsla_data$Market_State <- ifelse(
  tsla_data$VHF > 0.5, " 趋势市场", " 震荡市场"
)

# 统计不同市场状态的天数
market_state_counts <- table(tsla_data$Market_State)
print(market_state_counts)

##
## 趋势市场 震荡市场
##      163      1149

# 计算不同市场状态下的平均收益率
tsla_data$Return <- (tsla_data$Close - lag(tsla_data$Close)) / lag(tsla_data$Close) * 100
market_state_returns <- tsla_data %>%
```

```
group_by(Market_State) %>%
  summarise(
    Avg_Return = mean(Return, na.rm = TRUE),
    Std_Return = sd(Return, na.rm = TRUE),
    Positive_Days = sum(Return > 0, na.rm = TRUE),
    Negative_Days = sum(Return < 0, na.rm = TRUE)
  )

print(market_state_returns)

## # A tibble: 2 x 5
##   Market_State Avg_Return Std_Return Positive_Days Negative_Days
##   <chr>          <dbl>      <dbl>        <int>        <int>
## 1 趋势市场       0.963      5.24         104          58
## 2 震荡市场       0.114      4.07         580         568

# 可视化市场状态
ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  geom_rect(
    data = subset(tsla_data, Market_State == "震荡市场"),
    aes(xmin = Date, xmax = lead(Date), ymin = -Inf, ymax = Inf),
    fill = "gray", alpha = 0.2, inherit.aes = FALSE
  ) +
  labs(title = "特斯拉 (TSLA) 市场状态",
       x = "日期",
       y = "收盘价 ($)") +
  theme_minimal(base_family = "SimHei")
```



```
# 计算价格趋势 (使用 20 天移动平均线)
tsla_data$MA20 <- SMA(tsla_data$Close, n = 20)
tsla_data$Trend <- ifelse(tsla_data$Close > tsla_data$MA20, "上升", "下降")

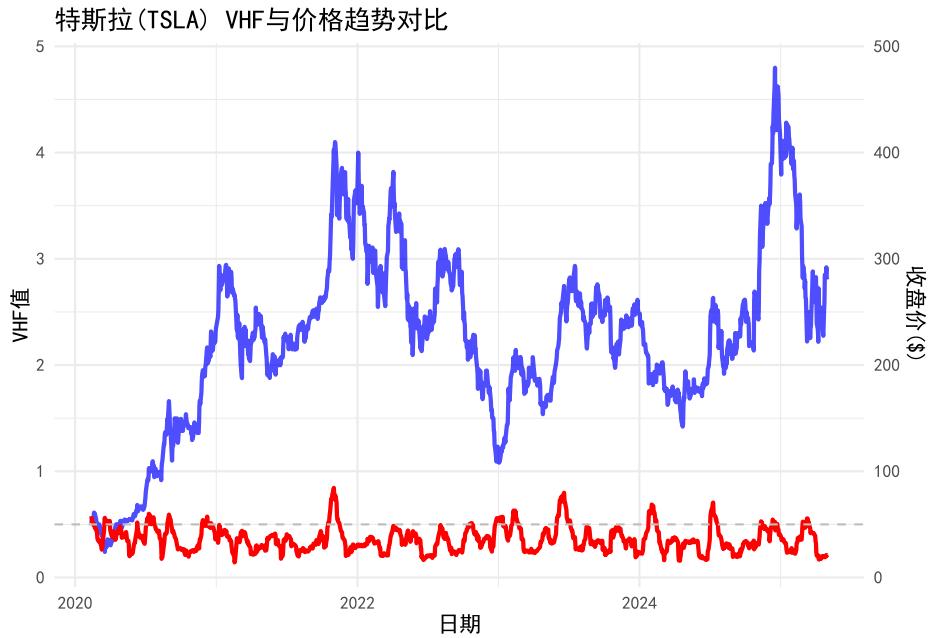
# 分析 VHF 与价格趋势的关系
trend_vhf_analysis <- tsla_data %>%
  group_by(Trend) %>%
  summarise(
    Avg_VHF = mean(VHF, na.rm = TRUE),
    High_VHF_Days = sum(VHF > 0.5, na.rm = TRUE),
    Low_VHF_Days = sum(VHF <= 0.5, na.rm = TRUE)
  )

print(trend_vhf_analysis)

## # A tibble: 3 x 4
##   Trend Avg_VHF High_VHF_Days Low_VHF_Days
##   <chr>   <dbl>        <int>        <int>
#>   上升     0.5          100          100
#>   下降     0.5           0            0
```

```
## 1 上升    0.375      109      599
## 2 下降    0.346      49       536
## 3 <NA>    0.421      5        14

# 可视化 VHF 与价格趋势
ggplot(tsla_data, aes(x = Date)) +
  geom_line(aes(y = Close/100), color = "blue", size = 1, alpha = 0.7) + # 缩放价格以适
  geom_line(aes(y = VHF), color = "red", size = 1) +
  geom_hline(yintercept = 0.5, linetype = "dashed", color = "gray") +
  labs(
    title = "特斯拉 (TSLA) VHF 与价格趋势对比",
    x = "日期",
    y = "指标值"
  ) +
  scale_y_continuous(
    name = "VHF 值",
    sec.axis = sec_axis(~ . * 100, name = "收盘价 ($)")
  ) +
  theme_minimal(base_family = "SimHei")
```



AI 引导词

“介绍垂直水平过滤器的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.6 埃勒相关性趋势指标（CTI） 指标含义

埃勒相关性趋势指标，即 Ehler's Correlation Trend Indicator (CTI) 是由 John Ehlers 开发的一种技术分析工具，用于识别市场趋势并过滤价格噪音。该指标基于价格与移动平均线之间的相关性计算，能够有效区分趋势市场和震荡市场。

CTI 的核心特点包括：

1. 趋势识别：通过计算价格与移动平均线的相关性，量化趋势强度
2. 噪音过滤：比传统移动平均线更能抵抗价格波动的干扰
3. 多周期应用：可用于不同时间周期，适应短期交易和长期投资
4. 信号明确：指标值接近 1 表示强上升趋势，接近-1 表示强下降趋势，接近 0 表示无明显趋势

数学公式

CTI 的计算公式基于价格序列与自身移动平均线之间的相关性：

1. 计算价格序列与移动平均线的相关性

$$CTI = \frac{\sum_{i=1}^n (P_i - \bar{P})(MA_i - \bar{MA})}{\sqrt{\sum_{i=1}^n (P_i - \bar{P})^2 \sum_{i=1}^n (MA_i - \bar{MA})^2}}$$

其中：

- P_i 是第 i 期的价格
- MA_i 是第 i 期的移动平均线值
- \bar{P} 是价格序列的均值
- \bar{MA} 是移动平均线序列的均值
- n 是计算周期

2. 使用改进的移动平均线

Ehlers 推荐使用 **SuperSmoothen Filter** 作为移动平均线，以减少延迟并提高对价格变化的响应速度：

$$MA_i = \alpha \cdot P_i + (1 - \alpha) \cdot MA_{i-1} + \beta \cdot (MA_{i-1} - MA_{i-2})$$

其中：

- $\alpha = \sin\left(\frac{2\pi}{n}\right)$
- $\beta = \exp\left(-\frac{1.414\pi}{n}\right)$
- n 是计算周期

代码实现及可视化

```
tsla_cti <- CTI(TSLA[, "Close"], n = 20) # 使用 20 天周期

tail(tsla_cti)

##                  cti
## 2025-04-24 -0.509774436
## 2025-04-25 -0.230075188
## 2025-04-28  0.001503759
## 2025-04-29  0.186466165
## 2025-04-30  0.342857143
## 2025-05-01  0.536842105

# 将结果合并到数据框中
tsla_data <- data.frame(
  Date = index(TSLA),
  Close = as.numeric(Cl(TSLA)),
  CTI = as.numeric(tsla_cti)
)

# 移除 NA 值
tsla_data <- na.omit(tsla_data)
```

```
# 查看计算结果
head tsla_data

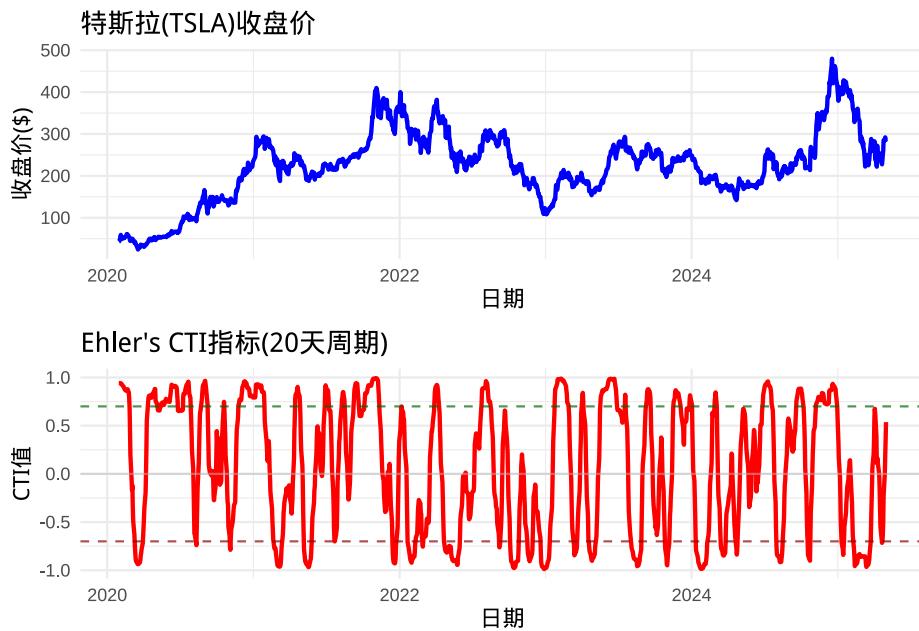
##          Date   Close      CTI
## 20 2020-01-30 42.7207 0.9413534
## 21 2020-01-31 43.3713 0.9413534
## 22 2020-02-03 52.0000 0.9413534
## 23 2020-02-04 59.1373 0.9413534
## 24 2020-02-05 48.9800 0.9368421
## 25 2020-02-06 49.9307 0.9338346

# 创建价格图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "特斯拉 (TSLA) 收盘价",
       x = "日期",
       y = "收盘价 ($)") +
  theme_minimal()

# 创建 CTI 指标图表
p2 <- ggplot(tsla_data, aes(x = Date, y = CTI)) +
  geom_line(color = "red", size = 1) +
  geom_hline(yintercept = 0.7, linetype = "dashed", color = "darkgreen", alpha = 0.7) +
  geom_hline(yintercept = -0.7, linetype = "dashed", color = "darkred", alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "solid", color = "gray", alpha = 0.5) +
  labs(title = "Ehler's CTI 指标 (20 天周期)",
       x = "日期",
       y = "CTI 值") +
  theme_minimal()

# 使用 patchwork 组合图表

p1 / p2
```



AI 引导词

“介绍的 Ehler’s Correlation Trend Indicator 意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.7 商品通道指数 (CCI) 指标含义

商品通道指数 (Commodity Channel Index, CCI) 由 Donald Lambert 在 1980 年提出，是一种衡量价格偏离其统计平均水平的技术指标。CCI 最初用于大宗商品市场，但现在广泛应用于各类金融市场。该指标通过比较当前价格与一定周期内的平均价格，识别超买超卖状态和潜在趋势反转点。

CCI 的主要特点：

- 波动范围没有固定限制，但通常在 ± 100 之间震荡
- 正值表示价格高于平均水平，负值表示价格低于平均水平
- 超买超卖阈值通常设为 $+100$ 和 -100
- 可用于识别趋势强度和潜在转折点
- 与价格形成背离时，往往预示趋势即将反转

数学公式

CCI 的计算公式

$$CCI = \frac{TP - SMA(TP, n)}{0.015 \times MAD(TP, n)}$$

其中：

- TP (Typical Price) 是典型价格: $TP = \frac{H+L+C}{3}$
- $SMA(TP, n)$ 是 n 周期内典型价格的简单移动平均
- $MAD(TP, n)$ 是 n 周期内典型价格的平均绝对偏差: $MAD = \frac{\sum_{i=1}^n |TP_i - SMA(TP, n)|}{n}$
- 0.015 是缩放因子，用于将 CCI 值标准化到常用范围内
- n 是计算周期 (默认 20)

代码实现及可视化

```
cci <- CCI(TSLA[, c("High", "Low", "Close")],  
            n = 20,  
            c = 0.015  
            )  
tail(cci, 5)
```

```
##                  cci  
## 2025-04-25 154.5936  
## 2025-04-28 179.0336  
## 2025-04-29 164.7422  
## 2025-04-30 112.2681  
## 2025-05-01 125.5566
```

CCI 指标的常见交易信号：

1. $CCI > +100$: 市场处于超买状态，可能出现回调
2. $CCI < -100$: 市场处于超卖状态，可能出现反弹
3. CCI 从下向上穿越 $+100$: 看涨信号，趋势可能转为上升
4. CCI 从上向下穿越 -100 : 看跌信号，趋势可能转为下降

5. 价格创新高而 CCI 未能同步创新高：看跌背离，可能预示顶部
6. 价格创新低而 CCI 未能同步创新低：看涨背离，可能预示底部

需要注意的是，CCI 在强势趋势市场中可能会持续处于超买或超卖区域，因此应结合其他指标和价格形态进行综合分析，避免在单边趋势中产生误判。此外，交易者也可以根据市场特性调整 CCI 的计算周期和阈值设置。

AI 引导词

“介绍商品通道指数的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.8 方向移动指标 (DMI) 指标含义

方向移动指标 (Directional Movement Index, DMI) 由 J. Welles Wilder 在 1978 年提出，是一种用于识别市场趋势方向和强度的技术分析工具。DMI 系统包含三个核心指标：

- +DI(Positive Directional Indicator): 衡量上升趋势的强度
- -DI(Negative Directional Indicator): 衡量下降趋势的强度
- ADX(Average Directional Index): 衡量整体趋势的强度（不考虑方向）

DMI 的主要作用：

- 判断市场是否存在明确趋势（通过 ADX 值）
- 确定趋势的方向（通过 + DI 和 - DI 的交叉）
- 识别潜在的趋势反转点
- 过滤无趋势市场中的虚假信号

DMI 特别适用于趋势跟踪策略，能够有效避免在盘整市场中频繁交易。

数学公式

DMI 的计算步骤

1. 计算方向移动值 (DM)

$$+DM = \begin{cases} H_t - H_{t-1}, & \text{如果 } H_t - H_{t-1} > L_{t-1} - L_t \text{ 且 } H_t - H_{t-1} > 0 \\ 0, & \text{其他情况} \end{cases}$$

$$-DM = \begin{cases} L_{t-1} - L_t, & \text{如果 } L_{t-1} - L_t > H_t - H_{t-1} \text{ 且 } L_{t-1} - L_t > 0 \\ 0, & \text{其他情况} \end{cases}$$

2. 计算真实范围 (TR)

$$TR = \max(H_t - L_t, |H_t - Cl_{t-1}|, |L_t - Cl_{t-1}|)$$

3. 计算方向指标 (DI):

$$+DI = \frac{EMA(+DM, n)}{EMA(TR, n)} \times 100$$

$$-DI = \frac{EMA(-DM, n)}{EMA(TR, n)} \times 100$$

4. 计算方向移动指数 (DX):

$$DX = \frac{|+DI - -DI|}{+DI + -DI} \times 100$$

5. 计算平均方向指数 (ADX):

$$ADX = EMA(DX, n)$$

其中:

- EMA 表示指数移动平均
- n 是计算周期 (默认 14)

代码实现及可视化

```
adx.14 <- ADX(TSLA[,c("High","Low","Close")],  
                 n = 14)  
  
tail(adx.14)  
  
##                DIp      DIn       DX      ADX  
## 2025-04-24 21.57461 23.47878 4.226464 18.99854  
## 2025-04-25 28.87952 21.28879 15.130534 18.72226  
## 2025-04-28 29.39865 19.66728 19.833261 18.80161  
## 2025-04-29 27.98202 18.71957 19.833261 18.87530  
## 2025-04-30 25.91838 20.35491 12.023058 18.38586  
## 2025-05-01 27.19375 19.54691 16.360145 18.24116
```

AI 引导词

“介绍方向移动指标的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.9 趋势检测指数 (TDI) 指标含义

趋势检测指数 (Trend Detection Index, TDI) 是一种技术分析工具，用于识别市场趋势的强度和方向。TDI 结合了价格动量和波动率的概念，通过多周期分析来减少假信号，特别适用于判断市场是否处于明确趋势中，或处于盘整阶段。

TDI 的核心优势：

- 趋势强度评估：量化趋势的强弱程度，帮助避免在弱趋势环境下使用趋势跟踪策略
- 方向确认：明确指示当前主导趋势方向（上升或下降）
- 早期反转信号：通过动量变化提前预警潜在的趋势反转
- 多时间框架分析：可同时应用于不同周期，提供更全面的市场视角

数学公式

TDI 的计算公式

1. 计算价格变化率 (ROC)

$$ROC(n) = \frac{C_t - C_{t-n}}{C_{t-n}} \times 100$$

2. 计算 ROC 的指数移动平均 (EMA)

$$ROC(EMA) = EMA(ROC(n), m)$$

3. 计算 ROC 的绝对 EMA

$$|ROC|(EMA) = EMA(|ROC(n)|, m)$$

4. 计算 TDI

$$TDI = \frac{ROC\ EMA}{|ROC|\ EMA} \times 100$$

其中：

- n 是计算 ROC 的周期（默认 12）
- m 是计算 EMA 的周期（默认 9）
- EMA 表示指数移动平均

代码实现及可视化

```
tdi.20 <- TDI(Cl(TSLA), n = 20)
```

AI 引导词

“介绍趋势检测指数(TDI)的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包) 的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.10 移动便利性指标 (EMV) 指标含义

EMV 指标 (Ease of Movement) 由 Richard W. Arms Jr. 开发，用于衡量“价格移动的”难易程度”。它结合了价格变动和成交量两个维度，能够有效识别市场的强弱状态和潜在的趋势反转点。

EMV 指标的核心思路是：

- 高 EMV 值：表示价格上涨轻松，成交量相对较小，市场可能处于强势状态
- 低 EMV 值：表示价格下跌轻松，成交量相对较小，市场可能处于弱势状态
- 接近零的 EMV 值：表示价格变动困难，需要较大成交量推动，市场可能处于盘整状态

主要用途：

1. 识别价格趋势的强度和持续性
2. 发现潜在的趋势反转信号
3. 过滤虚假突破信号
4. 辅助判断市场是否处于超买或超卖状态

数学公式

EMV 指标的计算分为几个主要步骤：

1. 计算价格中心变动值 (DM)

$$DM = \frac{(H + L)}{2} - \frac{(H_{prev} + L_{prev})}{2}$$

其中：

- H 和 L 是当前周期的最高价和最低价
- H_{prev} 和 L_{prev} 是前一周期的最高价和最低价

2. 计算标准化成交量比率 (BR)

$$BR = \frac{V}{(H - L)}$$

其中：

- $Volume$ 是当前周期的成交量

3. 计算 EMV 值

$$EMV = \frac{DM}{BR} \times 10000$$

4. 计算 EMV 的移动平均线

$$EMV\ MA = SMA(EMV, n)$$

其中：

- n 通常为 14，代表移动平均的周期

```
tsla_emv <- EMV(TSLA[, c("High", "Low")], TSLA[, "Volume"])
tsla_emv[is.infinite(tsla_emv)] <- NA

# 将结果合并到数据框中
tsla_data <- data.frame(
  Date = as.Date(index(TSLA)),
  Close = as.numeric(Cl(TSLA)),
  EMV = as.numeric(tsla_emv[, 1]), # EMV 线
  EMV_MA = as.numeric(tsla_emv[, 2]) # EMV 的移动平均线
)

# 移除 NA 值
tsla_data <- na.omit(tsla_data)

# 查看计算结果
head(tsla_data)

## [1] Date   Close   EMV     EMV_MA
## <0 rows> (or 0-length row.names)

# 创建价格图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "特斯拉 (TSLA) 收盘价",
       x = "日期",
```

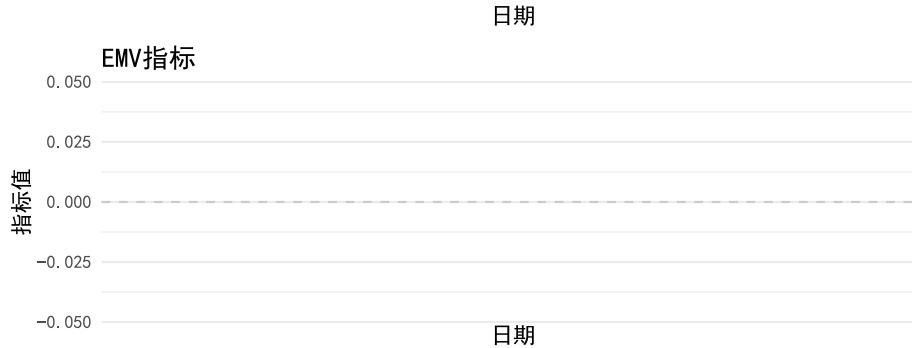
```
y = " 收盘价 ($)" +
theme_minimal(base_family = "SimHei")

# 创建 EMV 指标图表
p2 <- ggplot(tsla_data, aes(x = Date)) +
  geom_line(aes(y = EMV), color = "red", size = 0.8, alpha = 0.5) +
  geom_line(aes(y = EMV_MA), color = "green", size = 1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray", alpha = 0.7) +
  labs(title = "EMV 指标",
       x = " 日期",
       y = " 指标值") +
  theme_minimal(base_family = "SimHei")

# 使用 patchwork 组合图表
library(patchwork)
p1 / p2
```

特斯拉(TSLA)收盘价

收盘价(\$)



AI 引导词

“介绍 EMV 指标的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.2.11 趋势强度指数 (TSI) 指标含义

趋势强度指标 (Trend Strength Index, TSI) 由 William Blau 在 1991 年提出，是一种结合价格动量和趋势的技术指标。TSI 通过双重平滑处理价格变动，既能快速响应趋势变化，又能有效过滤市场噪声，特别适合识别市场转折点和确认趋势强度。

TSI 的主要特点：

- 在 -100 至 +100 之间波动，0 为中性线
- 正值表示上升趋势，负值表示下降趋势
- 绝对值越大，趋势越强
- 穿越零轴可视为趋势转变信号
- 与价格形成背离时，往往预示趋势即将反转
- 相比其他震荡指标（如 RSI），对趋势变化的响应更灵敏

数学公式

TSI 的计算公式

1. 计算价格变动 (PC)：

$$PC_t = P_t - P_{t-1}$$

2. 计算 PC 的双重 EMA：

$$EMA_r = EMA(PC, r)$$

$$EMA_s = EMA(EMA1, s)$$

3. 计算 PC 绝对值的双重 EMA：

$$|EMA_r| = EMA(|PC|, r)$$

$$|EMA_s| = EMA(|EMA_r|, s)$$

4. 计算 TSI:

$$TSI = \frac{EMA_s}{|EMA_s|} \times 100$$

其中：

- r 是快速周期 (默认 13)
- s 是慢速周期 (默认 25)
- EMA 表示指数移动平均

代码实现及可视化

```
source("TSI.r")
tsl.ex <- TSI(Ad(TSLA), r = 13, s = 25, signal_period = 9)
tail(tsl.ex, 5)
```

```
##           tsi      signal
## [1337,] -4.765870 -9.724461
## [1338,] -1.524952 -8.084559
## [1339,]  1.659386 -6.135770
## [1340,]  3.302931 -4.248030
## [1341,]  4.481355 -2.502153
```

AI 引导词

介绍趋势强度指数 (TSI) 的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包) 的 R 语言实例, 结果请返回 RMarkdown 代码。

3.1.2.12 抛物线停损指标 (SAR) 指标含义

抛物线停损指标的意义抛物线停损指标 (Parabolic Stop-and-Reverse) 是由 Welles Wilder 发明的一种技术分析工具。其主要意义如下：

- 趋势判断：通过在价格图表上绘制 SAR 点，直观地显示市场趋势。当 SAR 点位于价格下方时，表明市场处于上升趋势；当 SAR 点位于价格上方时，表明市场处于下降趋势。
- 止损设置：提供动态的止损参考。随着趋势的发展，SAR 点会根据价格的变化而调整，交易者可以将 SAR 点作为止损的依据，帮助控制风险。
- 趋势反转预警：当 SAR 点从价格下方穿越到价格上方（或反之）时，通常意味着市场趋势可能发生反转，为交易者提供趋势反转的信号，帮助捕捉潜在的交易机会。

数学公式

抛物线停损指标（Parabolic SAR）的计算公式如下：

1. 基本计算公式

上升趋势：

$$SAR_t = SAR_{t-1} + AF \times (EP_{t-1} - SAR_{t-1})$$

下降趋势：

$$SAR_t = SAR_{t-1} - AF \times (SAR_{t-1} - EP_{t-1})$$

其中：

- SAR_t 是当前周期的 SAR 值
- SAR_{t-1} 是上一周期的 SAR 值
- AF 是加速因子（Acceleration Factor）
- EP 是极值点（Extreme Point）

2. 加速因子（AF）调整规则

上升趋势中：

当价格创新高时：

$$AF_t = \min(AF_{t-1} + 0.02, AF_{\max})$$

下降趋势中：

当价格创新低时：

$$AF_t = \min(AF_{t-1} + 0.02, AF_{\max})$$

其中：

- AF_{\max} 是加速因子的最大值，通常为 0.2

3. 极值点 (EP) 更新规则

上升趋势中：

$$EP_t = \max(EP_{t-1}, H_t)$$

下降趋势中：

$$EP_t = \min(EP_{t-1}, L_t)$$

其中：

- H_t 是当前周期的最高价
- L_t 是当前周期的最低价

4. 趋势反转条件

上升趋势转为下降趋势：

当 $L_t < SAR_t$ 时，趋势反转，新的 SAR 和 EP 设置为：

$$SAR_t = \max(EP_{t-1}, H_t)$$

下降趋势转为上升趋势:

当 $H_t > SAR_t$ 时, 趋势反转, 新的 SAR 和 EP 设置为:

$$SAR_t = \min(EP_{t-1}, L_t)$$

代码实现及可视化

```
sar_tsla <- SAR(TSLA[, c("High", "Low")], accel = c(0.02, 0.2))
tail(sar_tsla)
```

```
##             sar
## 2025-04-24 261.7963
## 2025-04-25 214.2500
## 2025-04-28 214.2500
## 2025-04-29 217.4744
## 2025-04-30 220.5698
## 2025-05-01 223.5414
```

AI 引导词

介绍抛物线停损指标的意义, 对应 Latex 公式的 Rmd 代码, 3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例, 返回为 Rmd 代码。

3.1.2.13 线性回归斜率 (LRS) 指标含义

线性回归斜率指标 (Linear Regression Slope) 是一种技术分析工具, 用于量化价格趋势的强度和方向。其核心意义在于:

1. 趋势强度评估:

- 斜率绝对值越大, 表明趋势越强
- 斜率接近零, 表明市场处于盘整状态

2. 趋势方向判断:

- 正斜率表示上升趋势
- 负斜率表示下降趋势

3. 提前预警潜在反转:

- 斜率由正转负可能预示顶部形成
- 斜率由负转正可能预示底部形成

4. 与价格形成背离:

- 价格创新高但斜率下降, 可能形成看跌背离
- 价格创新低但斜率上升, 可能形成看涨背离

5. 交易信号生成:

- 斜率上穿零轴可视为买入信号
- 斜率下穿零轴可视为卖出信号

数学公式

线性回归斜率指标基于最小二乘法拟合价格数据, 计算公式如下:

线性回归方程:

给定 n 个数据点 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, 线性回归方程为:

$$y = \beta_0 + \beta_1 x + \epsilon$$

其中:

- β_0 是截距
- β_1 是斜率
- ϵ 是误差项

斜率计算公式

$$\beta_1 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$

截距计算公式

$$\beta_0 = \frac{\sum_{i=1}^n y_i - \beta_1 \sum_{i=1}^n x_i}{n}$$

在时间序列分析中，通常令 $x_i = i$ ，即 $x_1 = 1, x_2 = 2, \dots, x_n = n$ ，此时斜率公式可简化为：

$$\beta_1 = \frac{12 \sum_{i=1}^n i \cdot y_i - 6(n+1) \sum_{i=1}^n y_i}{n(n^2 - 1)}$$

代码实现及可视化

```
source("runRegression.r")
rR.14 <- runRegression(Cl(TSLA), n = 14, output = "all")
rR.14.df <- convertRegressionToDataframe(rR.14, prefix = "reg14")
# 查看结果结构
str(rR.14.df)

## 'data.frame':    1327 obs. of  7 variables:
##   $ date           : Date, format: "2020-01-22" "2020-01-23" ...
##   $ reg14_intercept : num  28.6 29.3 30.2 31.2 31.9 ...
##   $ reg14_slope_14d : num  0.608 0.601 0.568 0.504 0.464 ...
##   $ reg14_r_squared : num  0.86 0.859 0.844 0.8 0.769 ...
##   $ reg14_adj_r_squared: num  0.848 0.847 0.83 0.783 0.749 ...
##   $ reg14_f_statistic : num  73.4 73 64.7 48 39.9 ...
##   $ reg14_p_value    : num  1.85e-06 1.91e-06 3.56e-06 1.58e-05 3.86e-05 ...

# 计算不同周期的线性回归斜率
reg_14 <- runRegression(Cl(TSLA), n = 14)
reg_28 <- runRegression(Cl(TSLA), n = 28)

# 将回归结果转换为数据框
reg_14_df <- convertRegressionToDataframe(reg_14, prefix = "reg14")
reg_28_df <- convertRegressionToDataframe(reg_28, prefix = "reg28")

# 合并所有数据
```

```
combined_df <- merge(
  data.frame(date=as.Date(index(TSLA)),coredata(TSLA)),
  merge(reg_14_df, reg_28_df, by = "date", all = TRUE),
  by = "date",
  all = TRUE
) %>%
  na.omit()

# 生成交易信号
combined_df <- combined_df %>%
  mutate(
    # 基于斜率方向的信号
    direction_signal = ifelse(reg14_slope_14d > 0, "买入", "卖出"),

    # 基于斜率交叉的信号
    cross_signal = ifelse(
      reg14_slope_14d > reg28_slope_28d &
        lag(reg14_slope_14d) <= lag(reg28_slope_28d), "金叉买入",
      ifelse(
        reg14_slope_14d < reg28_slope_28d &
          lag(reg14_slope_14d) >= lag(reg28_slope_28d), "死叉卖出", NA
      )
    ),
    # 结合 R2 过滤的信号 (只在 R2>0.7 时生成信号)
    filtered_signal = ifelse(
      reg14_slope_14d > 0 & reg14_r_squared > 0.7, "强买入",
      ifelse(reg14_slope_14d < 0 & reg14_r_squared > 0.7, "强卖出", NA)
    )
  )

# 定义可视化函数
```

```
create_price_chart <- function(data) {  
  ggplot(data, aes(x = date, y = Close)) +  
    geom_line(color = "black", linewidth = 1) +  
    labs(title = " 特斯拉股票价格", y = " 价格", x = "") +  
    theme_minimal()  
}  
  
create_slope_chart <- function(data) {  
  ggplot(data, aes(x = date)) +  
    geom_line(aes(y = reg14_slope_14d, color = "14 天斜率"), linewidth = .3) +  
    geom_line(aes(y = reg28_slope_28d, color = "28 天斜率"), linewidth = .3) +  
    geom_hline(yintercept = 0, color = "gray", linetype = "dashed") +  
  
    # 添加交叉信号标记  
    geom_point(  
      data = data[!is.na(data$cross_signal), ],  
      aes(y = ifelse(cross_signal == " 金叉买入", reg14_slope_14d, reg14_slope_14d),  
           color = cross_signal),  
      size = 1,  
      shape = 16  
    ) +  
  
    scale_color_manual(  
      values = c("14 天斜率" = "blue", "28 天斜率" = "red",  
                " 金叉买入" = "green", " 死叉卖出" = "red"),  
      breaks = c("14 天斜率", "28 天斜率", " 金叉买入", " 死叉卖出")  
    ) +  
  
    labs(title = " 线性回归斜率指标", y = " 斜率值", x = " 日期", color = "") +  
    theme_minimal() +  
    theme(legend.position = "top")  
}
```

```
create_rsquared_chart <- function(data) {  
  ggplot(data, aes(x = date)) +  
    geom_line(aes(y = reg14_r_squared, color = "14 天 R2"), size = 1) +  
    geom_line(aes(y = reg28_r_squared, color = "28 天 R2"), size = 1) +  
    geom_hline(yintercept = 0.7, color = "gray", linetype = "dashed") +  
  
    scale_color_manual(  
      values = c("14 天 R2" = "purple", "28 天 R2" = "orange")  
    ) +  
  
    labs(title = " 回归拟合优度 (R2)", y = "R2 值", x = " 日期", color = "") +  
    theme_minimal() +  
    theme(legend.position = "top")  
}  
  
# 创建图表  
price_chart <- create_price_chart(combined_df)  
slope_chart <- create_slope_chart(combined_df)  
rsquared_chart <- create_rsquared_chart(combined_df)  
  
# 组合并显示图表  
grid.arrange(price_chart, slope_chart, rsquared_chart, ncol = 1, heights = c(2, 1.5, 1.5))
```



```
# 查看最近 10 个交易日的信号
tail(combined_df[, c("date", "Close", "reg14_slope_14d", "reg28_slope_28d",
                     "reg14_r_squared", "direction_signal", "cross_signal", "filtered_sig
                     ...]
```

	date	Close	reg14_slope_14d	reg28_slope_28d	reg14_r_squared
## 1331	2025-04-17	241.37	-1.5770769	0.25874111	0.15959183
## 1332	2025-04-21	227.50	-2.0873187	-0.09846743	0.24350653
## 1333	2025-04-22	237.97	-1.8526593	-0.23765736	0.20345594
## 1334	2025-04-23	250.74	-0.6476923	-0.33575807	0.03693363
## 1335	2025-04-24	259.51	0.4382637	-0.30075260	0.01891816
## 1336	2025-04-25	284.95	1.3670989	-0.17798303	0.11541753
## 1337	2025-04-28	285.88	1.9200000	-0.17482759	0.18117751
## 1338	2025-04-29	292.03	2.0406374	-0.07565134	0.19317881
## 1339	2025-04-30	282.16	3.2303956	-0.07697592	0.44885505
## 1340	2025-05-01	280.52	3.6767692	-0.01673235	0.53983400
			direction_signal	cross_signal	filtered_signal
## 1331		卖出	<NA>	<NA>	<NA>
## 1332		卖出	<NA>	<NA>	<NA>

```
## 1333      卖出      <NA>      <NA>
## 1334      卖出      <NA>      <NA>
## 1335      买入      金叉买入    <NA>
## 1336      买入      <NA>      <NA>
## 1337      买入      <NA>      <NA>
## 1338      买入      <NA>      <NA>
## 1339      买入      <NA>      <NA>
## 1340      买入      <NA>      <NA>
```

AI 引导词

1. 介绍线性回归斜率指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.2.14 平滑异同移动平均线 (MACD) 指标含义

MACD (Moving Average Convergence Divergence) 是技术分析中常用的动量指标，由 Gerald Appel 于 1979 年提出。它通过计算两条不同周期移动平均线之间的差异，来识别市场趋势的强度、方向、动量和持续时间。

MACD 由三部分组成：

- MACD 线：通常是 12 日 EMA 与 26 日 EMA 的差值，反映短期与长期趋势的差异
- 信号线：MACD 线的 9 日 EMA，用作触发买卖信号的参考
- 柱状图 (Histogram)：MACD 线与信号线的差值，直观展示两者距离和方向变化

MACD 的核心思想是：当短期均线向上穿越长期均线时 (MACD 线从下向上穿越信号线)，产生买入信号；反之则产生卖出信号。柱状图的收缩或扩张可以预示趋势的强弱变化，柱状图由正转负或由负转正时，通常是趋势反转的信号。

数学公式

MACD 指标的计算基于以下三个主要公式：

1. MACD 线的计算公式:

$$MACD = EMA_{short} - EMA_{long}$$

其中, EMA_{short} 通常为 12 日指数移动平均线, EMA_{long} 通常为 26 日指数移动平均线。

2. 信号线的计算公式:

$$Signal = EMA(MACD, n)$$

其中, n 通常为 9, 表示对 MACD 值再计算 9 日指数移动平均线。

3. 柱状图的计算公式:

$$Histogram = MACD - Signal$$

在 R 语言中, 我们可以使用 TTR 包中的 `MACD()` 函数来计算这些值, 该函数的默认参数与上述标准设置一致。

代码实现及可视化

```
macd <- MACD( C1(TSLA), 12, 26, 9, maType="EMA" )
macd2 <- MACD( C1(TSLA), 12, 26, 9,
                 maType=list(list(SMA), list(EMA, wilder=TRUE), list(SMA)) )
```

AI 引导词

1. 介绍「MACD」指标的意义; 2. 对应 Latex 公式的 Rmd 代码;
3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.3 震荡类指标

震荡指标 (Oscillators) 是技术分析中用于判断市场超买或超卖状态的工具, 通常在 0-100 或 -100-100 范围内波动。以下介绍常见的震荡指标及其在 R

语言 TTR 包中的计算方法。

3.1.3.1 随机振荡器 (SO) 指标含义

随机振荡器 (Stochastic Oscillator) 是由 George Lane 在 20 世纪 50 年代开发的动量指标，用于比较某一收盘价与一定时间周期内价格区间的相对位置，从而判断市场超买或超卖状态。其核心思想是：在上涨趋势中，收盘价往往接近当日最高价；而在下跌趋势中，收盘价倾向于接近当日最低价。该指标有两条线： $\%K$ （快速线）和 $\%D$ （慢速线），通过两者的交叉和与超买超卖阈值的关系来生成交易信号。

随机指标的主要特点：

- 适用于震荡市场，帮助识别潜在反转点
- 常与其他指标结合使用，增强信号可靠性
- 有慢速和快速版本，慢速版本更平滑，减少虚假信号
- 可通过背离现象预测趋势变化

数学公式

随机指标的计算公式

- 快速随机指标

$$\%K = \frac{Close - Lowest\ Low}{Highest\ High - Lowest\ Low} \times 100$$

其中：

- $Close$ 是当前收盘价
- $Lowest\ Low$ 是过去 n 天的最低价格
- $Highest\ High$ 是过去 n 天的最高价格
- 慢速随机指标

$$\%D = SMA(\%K, m)$$

其中：

- SMA 是简单移动平均
- m 是平滑周期（通常为 3）

常见变种”Slow %K”是对快速%K 的平滑：

$$Slow\%K = SMA(\%K, m)$$

代码实现及可视化

```
stochOSC <- stoch(TSLA[, c("High", "Low", "Close")],
                     nFastK = 14,
                     nSlowK = 3,
                     nSlowD = 3)

tail(stochOSC, n = 5)

##           fastK      fastD      slowD
## 2025-04-25 0.9738292 0.7702484 0.5659629
## 2025-04-28 0.8834674 0.8687128 0.7321236
## 2025-04-29 0.9607326 0.9393431 0.8594348
## 2025-04-30 0.8237824 0.8893275 0.8991278
## 2025-05-01 0.8010268 0.8618473 0.8968393
```

AI 引导词

“介绍的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.3.2 随机动量指数 (SMI) 指标含义

SMI (Stochastic Momentum Index) 是一种经过改进的随机指标，由 William Blau 在 1993 年提出。它通过衡量收盘价与价格区间的相对位置，来识别市场超买超卖状态和潜在趋势反转点。与传统随机指标相比，SMI 对价格变化更为敏感，能够减少虚假信号，更准确地捕捉市场动量变化。

SMI 的核心原理基于以下几点：

- 动量分析：SMI 通过计算收盘价与一定时期内价格区间中点的偏离程度，衡量价格动量的强弱
- 超买超卖识别：指标值在 ± 100 之间波动，通常将 ± 40 作为超买超卖阈值
- 趋势反转信号：当 SMI 从超买区域向下穿越中心线（0）或从超卖区域向上穿越中心线时，可能预示趋势反转
- 背离检测：价格创新高 / 新低而 SMI 未能同步，可能形成顶 / 底背离，暗示趋势即将结束

SMI 特别适用于震荡市场，能够有效过滤短期价格波动干扰，提供更可靠的交易信号。

数学公式

Stochastic Momentum Index 的计算基于以下步骤和公式：

1. 计算价格离差值 (Difference)：

$$Diff = C - \frac{(H + L)}{2}$$

该值衡量收盘价与当期价格区间中点的偏离程度。

2. 计算离差的 n 周期指数移动平均 (EMA1)：

$$EMA1 = EMA(Difference, n_1)$$

通常取 $n_1 = 3$ ，用于平滑短期价格波动。

3. 计算离差绝对值的 n 周期指数移动平均 (EMA2)：

$$EMA2 = EMA(|Diff|, n_2)$$

通常取 $n_2 = 3$ ，用于衡量价格波动的绝对幅度。

4. 计算 SMI 值：

$$SMI = 100 \times \frac{EMA1}{0.5 \times EMA2}$$

其中 0.5 为缩放因子，使 SMI 值在 ± 100 范围内波动。

在 R 语言中，我们可以使用 TTR 包中的 `SMI()` 函数来计算这些值，默认参数为 $n = 14, n_1 = 3, n_2 = 3$ 。

代码实现及可视化

```
SMI3MA <- SMI(TSLA[,c("High","Low","Close")],  
                 maType=list(list(SMA), list(EMA, wilder=TRUE), list(SMA)))
```

AI 引导词

1. 介绍「SMI」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.3.3 方向性波动指标 (DVI) 指标含义

DVI 指标 (Directional Volatility Index) 由 David Bostian 开发，是一种结合了价格方向性和波动性的技术分析工具。该指标通过计算价格变动的方向性和波动性比率，帮助交易者识别市场趋势的强度和潜在的反转点。

核心思想：

1. 方向性分析：衡量价格变动的方向性强度，区分随机波动和真实趋势
2. 波动性分析：评估价格波动的幅度，识别市场活跃度变化
3. 比率计算：通过方向性与波动性的比率，过滤弱趋势信号，聚焦强趋势机会

主要用途：

1. 确认市场趋势的强度和持续性
2. 识别潜在的趋势反转点
3. 过滤虚假突破信号
4. 优化交易入场和出场时机

数学公式

DVI 指标的计算分为几个主要步骤：

1. 计算方向性变动 (Dir)

$$Dir = \frac{C - C_{prev}}{|C - C_{prev}|}$$

其中：

- Dir 的值为 1 (上涨)、-1 (下跌) 或 0 (无变动)
- $Close$ 和 $Close_{prev}$ 分别是当前和前一周期的收盘价

2. 计算方向性强度 (DS)

$$DS = \frac{C - C_n}{ATR(n)}$$

其中：

- $Close_n$ 是 n 周期前的收盘价
- $ATR(n)$ 是 n 周期的平均真实波幅

3. 计算波动性比率 (VR)

$$VR = \frac{ATR(n)}{ATR(m)}$$

其中：

- $ATR(n)$ 是短期平均真实波幅 (通常 $n=14$)
- $ATR(m)$ 是长期平均真实波幅 (通常 $m=28$)

4. 计算 DVI 值

$$DVI = DS \times VR$$

```
# 计算 TSLA 的 DVI 指标
tsla_dvi <- DVI(Cl(TSLA))

# 将结果合并到数据框中
tsla_data <- data.frame(
  Date = index(TSLA),
  Close = as.numeric(Cl(TSLA)),
  DVI = as.numeric(tsla_dvi)
)

# 移除 NA 值
tsla_data <- na.omit(tsla_data)

# 查看计算结果
head(tsla_data)

##           Date   Close      DVI
## 357 2021-06-02 201.7067 0.6587302
## 358 2021-06-03 190.9467 0.5396825
## 359 2021-06-04 199.6833 0.3968254
## 360 2021-06-07 201.7100 0.2817460
## 361 2021-06-08 201.1967 0.2222222
## 362 2021-06-09 199.5933 0.1904762

# 创建价格图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "特斯拉 (TSLA) 收盘价",
       x = "日期",
       y = "收盘价 ($)") +
  theme_minimal(base_family = "SimHei")

# 创建 DVI 指标图表
p2 <- ggplot(tsla_data, aes(x = Date, y = DVI)) +
```

```

geom_line(color = "red", size = 1) +
geom_hline(yintercept = 0, linetype = "dashed", color = "gray", alpha = 0.7) +
geom_hline(yintercept = 2, linetype = "dotted", color = "darkgreen", alpha = 0.7) +
geom_hline(yintercept = -2, linetype = "dotted", color = "darkred", alpha = 0.7) +
labs(title = "DVI 指标",
x = "日期",
y = "DVI 值") +
theme_minimal(base_family = "SimHei")

# 使用 patchwork 组合图表
library(patchwork)
p1 / p2

```



交易信号分析

基于 DVI 指标的交易信号通常包括:

```

# 生成 DVI 信号
tsla_data$DVI_Signal <- ifelse(
  tsla_data$DVI > 2, "买入", "# 强上升趋势"
)

```

```
ifelse(
  tsla_data$DVI < -2, "卖出", # 强下降趋势
  "持有" # 弱趋势或无趋势
)
)

# 统计信号数量
table(tsla_data$DVI_Signal)

## 
## 持有
## 2956

# 可视化 DVI 信号
ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  geom_point(data = subset(tsla_data, DVI_Signal == "买入"),
             aes(color = "买入"), size = 3, shape = 16) +
  geom_point(data = subset(tsla_data, DVI_Signal == "卖出"),
             aes(color = "卖出"), size = 3, shape = 17) +
  scale_color_manual(values = c("买入" = "green", "卖出" = "red")) +
  labs(title = "DVI 指标交易信号",
       x = "日期",
       y = "收盘价 ($)",
       color = "交易信号") +
  theme_minimal(base_family = "SimHei")
```



DVI 与价格趋势的关系

```
# 计算价格趋势 (使用 20 天移动平均线)
tsla_data$MA20 <- SMA(tsla_data$Close, n = 20)
tsla_data$Price_Trend <- ifelse(tsla_data$Close > tsla_data$MA20, "上升", "下降")

# 分析 DVI 与价格趋势的关系
dvi_trend_analysis <- tsla_data %>%
  group_by(Price_Trend) %>%
  summarise(
    Avg_DVI = mean(DVI, na.rm = TRUE),
    Positive_DVI_Days = sum(DVI > 0, na.rm = TRUE),
    Negative_DVI_Days = sum(DVI < 0, na.rm = TRUE)
  )

print(dvi_trend_analysis)

## # A tibble: 3 x 4
##   Price_Trend Avg_DVI Positive_DVI_Days Negative_DVI_Days
```

```
## <chr>      <dbl>      <int>      <int>
## 1 上升      0.674     1484        0
## 2 下降      0.295     1453        0
## 3 <NA>      0.464      19         0

# 可视化 DVI 与价格趋势
ggplot(tsla_data, aes(x = Date)) +
  geom_line(aes(y = Close/100), color = "blue", size = 1, alpha = 0.7) + # 缩放价格以适
  geom_line(aes(y = DVI), color = "red", size = 1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray") +
  geom_hline(yintercept = 2, linetype = "dotted", color = "darkgreen") +
  geom_hline(yintercept = -2, linetype = "dotted", color = "darkred") +
  labs(
    title = "特斯拉 (TSLA) DVI 与价格趋势对比",
    x = "日期",
    y = "指标值"
  ) +
  scale_y_continuous(
    name = "DVI 值",
    sec.axis = sec_axis(~ . * 100, name = "收盘价 ($)")
  ) +
  theme_minimal(base_family = "SimHei")
```



AI 引导词

1. 介绍 DVI 指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.3.4 相对强弱指数 (RSI) 指标含义

RSI (Relative Strength Index) 是由技术分析家 J. Welles Wilder 于 1978 年提出的动量指标，用于衡量价格变动的速度和幅度，判断市场是否处于超买或超卖状态。该指标通过比较一段时期内股价上涨幅度的平均值与下跌幅度的平均值，来评估多空双方力量的强弱。

RSI 的核心原理和应用包括：

- 取值范围：RSI 值在 0-100 之间波动，通常将 70 视为超买阈值，30 视为超卖阈值
- 超买超卖信号：当 RSI 超过 70 时，表明市场处于超买状态，可能面临回调；当 RSI 低于 30 时，表明市场处于超卖状态，可能出现反弹
- 背离分析：当价格创新高而 RSI 未能同步创新高，形成顶背离，是潜在的看跌信号；反之，价格创新低而 RSI 未能同步创新低，形成底背

离，是潜在的看涨信号

- 趋势确认：在强势趋势中，RSI 可能长时间处于超买 / 超卖区域而不反转，此时需结合价格趋势综合判断

RSI 是技术分析中最常用的指标之一，适用于各种时间周期，尤其在震荡市场中表现出色，但在强趋势市场中需谨慎使用，避免过早抄底或逃顶。

数学公式

Relative Strength Index 的计算基于以下步骤和公式：

1. 计算价格变动：

$$\Delta P = C_t - C_{t-1}$$

其中， C_t 为当前收盘价， C_{t-1} 为前一日收盘价。

2. 分离上涨和下跌变动：

$$U = \begin{cases} \Delta Price, & \text{如果 } \Delta Price > 0 \\ 0, & \text{如果 } \Delta Price \leq 0 \end{cases}$$

$$D = \begin{cases} -\Delta Price, & \text{如果 } \Delta Price < 0 \\ 0, & \text{如果 } \Delta Price \geq 0 \end{cases}$$

3. 计算平均上涨和下跌幅度（使用 Wilder 的平滑方法）：

$$\bar{U} = \frac{\sum_{i=1}^n U_i}{n}$$

$$\bar{D} = \frac{\sum_{i=1}^n D_i}{n}$$

其中， n 为计算周期，通常取 14。

4. 计算相对强度 (RS)：

$$RS = \frac{\bar{U}}{\bar{D}}$$

5. 计算 RSI 值：

$$RSI = 100 - \frac{100}{1 + RS}$$

在 R 语言中，我们可以使用 TTR 包中的 `RSI()` 函数来计算这些值，默认参数为 $n = 14$ 。

代码实现及可视化

```
# Default case
rsi <- RSI(Ad(TSLA))

# Case of one 'maType' for both MAs
rsiMA1 <- RSI(Ad(TSLA), n=14, maType="WMA", wts=Vo(TSLA))

# Case of two different 'maType's for both MAs
rsiMA2 <- RSI(Ad(TSLA), n=14, maType=list(maUp=list(EMA), maDown=list(WMA)))
```

AI 引导词

1. 介绍「RSI」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.3.5 平滑异同移动平均线 (MACD)

3.1.3.5.1 见章节3.1.2.14

3.1.3.6 威廉指标 (%R) 指标含义

Williams % R（威廉指标）是由 Larry Williams 在 1973 年开发的超买超卖型技术指标，用于衡量收盘价与特定周期内最高价之间的相对位置。与其他震荡指标类似，Williams % R 在 -100 至 0 的区间内波动，值越接近 -100 表示市场超买，越接近 0 表示市场超卖。该指标常与价格走势结合使用，以识别潜在的趋势反转点。

Williams % R 的主要特点包括：

- 对短期价格变动敏感，适合短期交易决策
- 可用于确认价格趋势的强度和可持续性
- 与其他指标（如 RSI、Stochastic Oscillator）结合使用时效果更佳
- 能够提前预警价格反转，尤其是在出现背离时

数学公式

Williams %R 的计算公式如下：

$$\%R = \frac{\text{Highest High} - \text{Close}}{\text{Highest High} - \text{Lowest Low}} \times (-100)$$

其中：

- *Highest High* 是过去 n 天内的最高价
- *Lowest Low* 是过去 n 天内的最低价
- *Close* 是当前收盘价
- n 是计算周期（通常为 14 天）

代码实现及可视化

```
wpr.14 <- WPR(TSLA[, c("High", "Low", "Close")],  
  n = 14,  
  scale = FALSE)  
  
tail(wpr.14)  
  
##           Close  
## 2025-04-24 0.25115817  
## 2025-04-25 0.02617080  
## 2025-04-28 0.11653257  
## 2025-04-29 0.03926738  
## 2025-04-30 0.17621757  
## 2025-05-01 0.19897322
```

AI 引导词

“介绍的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包）的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.3.7 终极震荡指标 (UO) 指标含义

终极振荡指标（Ultimate Oscillator, UO）是由 Larry Williams 在 1976 年提出的一种技术分析工具。它通过综合考量三个不同周期（一般是 7 天、14 天和 28 天）的价格动量，来降低市场短期波动所产生的虚假信号，进而更精准地识别超买和超卖状态。

该指标的核心原理在于：短期周期能够敏锐捕捉到近期价格的变动情况；中期周期可以反映价格的中期趋势；长期周期则能体现价格的长期支撑与压力水平。UO 将这三个周期的权重进行合理分配，最终得出一个综合的振荡指标，其数值范围在 0 到 100 之间。

UO 指标具有以下几方面的应用价值：

- 当 UO 数值超过 70 时，表明市场处于超买状态，价格可能即将下跌；
- 当 UO 数值低于 30 时，意味着市场处于超卖状态，价格可能即将上涨；
- 若 UO 指标与价格走势出现背离的情况，通常是市场趋势即将反转的重要信号；
- 当 UO 向上突破 50 时，显示市场上涨动能正在增强；当 UO 向下突破 50 时，则表明市场下跌动能正在加大。

数学公式

终极振荡指标的计算公式如下：

$$UO = 100 \times \frac{4 \times BP_1/SR_1 + 2 \times BP_2/SR_2 + BP_3/SR_3}{4 + 2 + 1}$$

其中：

- \$ BP \$ 代表的是买方力量 (Bull Power)，其计算公式为：\$ BP = 收盘价 - 最低价 \$

- $\$ SR \$$ 表示的是标准化比率 (Standardized Ratio)，计算公式为： $\$ SR = \text{最高价} - \text{最低价} \$$
- 下标 1、2、3 分别对应短期 (7 天)、中期 (14 天) 和长期 (28 天) 这三个不同的周期。

该公式通过加权平均的方式，将三个不同周期的买方力量与价格波动范围的比值进行整合，从而得到一个能够全面反映市场动量的综合指标。

代码实现及可视化

```
# 计算终极振荡指标 (UO)
tsla_uo <- ultimateOscillator(TSLA[, c("High", "Low", "Close")])

# 准备绘图数据
tsla_data <- data.frame(
  date = index(TSLA),
  close = as.numeric(Cl(TSLA)),
  uo = as.numeric(tsla_uo)
) %>%
  na.omit() %>%
  tail(120) # 只取最近 120 个交易日数据

# 创建收盘价图表
price_plot <- ggplot(tsla_data, aes(x = date, y = close)) +
  geom_line(color = "#3182bd", size = 1) +
  labs(title = "特斯拉 (TSLA) 收盘价",
       y = " 价格 (美元)",
       x = "") +
  theme_minimal(base_family = "SimHei") +
  theme(plot.title = element_text(face = "bold", hjust = 0.5),
        axis.text.x = element_blank()) +
  scale_x_date(date_breaks = "1 month", labels = date_format("%b %d"))

# 创建 UO 指标图表
uo_plot <- ggplot(tsla_data, aes(x = date, y = uo)) +
```

```
geom_line(color = "#e6550d", size = 1) +
  geom_hline(yintercept = 70, linetype = "dashed", color = "#de2d26") +
  geom_hline(yintercept = 50, linetype = "dashed", color = "#969696") +
  geom_hline(yintercept = 30, linetype = "dashed", color = "#31a354") +
  labs(title = " 终极振荡指标 (UO) ",
       y = "UO 值",
       x = " 日期") +
  theme_minimal(base_family = "SimHei") +
  theme(plot.title = element_text(face = "bold", hjust = 0.5)) +
  scale_y_continuous(limits = c(0, 100)) +
  scale_x_date(date_breaks = "1 month", labels = date_format("%b %d"))

# 使用 patchwork 组合图表
combined_plot <- price_plot / uo_plot +
  plot_layout(heights = c(2, 1)) +
  plot_annotation(
    title = " 特斯拉 (TSLA) 股票终极振荡指标分析",
    subtitle = " 基于 2024 年以来的交易数据",
    caption = " 数据来源: Yahoo Finance | 分析工具: R 语言",
  )

# 显示组合图表
print(combined_plot)
```

特斯拉(TSLA)股票终极振荡指标分析
基于2024年以来的交易数据



```
# 生成交易信号
tsla_data$signal <- ifelse(tsla_data$uo < 30, "买入",
                             ifelse(tsla_data$uo > 70, "卖出", "持有"))

# 计算最近的 UO 值和信号
latest_uo <- tail(tsla_data$uo, 1)
latest_signal <- tail(tsla_data$signal, 1)

cat(" 最新 UO 值:", latest_uo, "\n")

## 最新UO值: 63.6414
cat(" 当前信号:", latest_signal, "\n")

## 当前信号: 持有
```

AI 引导词

1. 介绍「UO」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.3.8 钱德动量摆动指标 (CMO) 指标含义

CMO 指标全称为钱德动量摆动指标 (Chande Momentum Oscillator)，由图莎尔·钱德 (Tushar Chande) 于 1994 年提出。该指标是一种技术分析工具，主要用于衡量价格变动的动量，其值在 -100 到 +100 之间波动。

CMO 指标的核心原理是对比一段时间内价格上涨幅度总和与下跌幅度总和的差异，进而判断市场买卖双方的力量对比。与 RSI (相对强弱指标) 和 MACD (指数平滑异同移动平均线) 等其他动量指标相比，CMO 对价格变化更为敏感，能够更及时地反映市场动量的转变。

在实际应用中，CMO 指标的主要作用如下：

- 当 CMO 值高于 +50 时，表明市场处于强势上涨状态；
- 当 CMO 值低于 -50 时，意味着市场处于强势下跌状态；
- CMO 指标与价格之间的背离现象，可作为潜在趋势反转的预警信号；
- 通过设定超买（如 +50）和超卖（如 -50）阈值，为交易决策提供参考。

数学公式

CMO 指标的计算公式如下：

$$CMO(n) = \frac{\sum_{i=1}^n U_i - \sum_{i=1}^n D_i}{\sum_{i=1}^n U_i + \sum_{i=1}^n D_i} \times 100$$

其中：

- U_i 代表第 i 期的上涨幅度，当本期价格高于上期价格时， U_i 等于本期价格与上期价格的差值，否则为 0；
- D_i 表示第 i 期的下跌幅度，当本期价格低于上期价格时， D_i 等于上期价格与本期价格差值的绝对值，否则为 0；
- n 为计算周期，一般默认取值为 14。

代码实现及可视化

```
cmo <- CMO(Cl(TSLA), 14)
tail(cmo)
```

```
## cmo
```

```
## 2025-04-24 -4.413018
## 2025-04-25 26.212139
## 2025-04-28 31.219947
## 2025-04-29 43.004229
## 2025-04-30 8.117359
## 2025-05-01 26.898795
```

AI 引导词

1. 介绍「CMO」指标的意义；2. 对应 Latex 公式的 Rmd 代码；
3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.3.9 动量指标 (Momentum) 指标含义

动量指标 (Momentum Indicator) 是金融市场技术分析中最基础的动能指标之一，用于衡量资产价格变化的速度。其核心思想是：价格上涨或下跌的速度加快，通常预示着当前趋势将持续；而价格上涨或下跌速度放缓，则可能是趋势即将反转的信号。

动量指标通过比较当前价格与过去某个时间点的价格，计算两者之间的差值或比率，从而量化价格变动的速度。与其他技术指标（如 RSI、MACD 等）相比，动量指标更加直观地反映了价格变动的加速度，因此被广泛用于识别市场超买超卖状态和潜在的趋势反转点。

动量指标的主要应用场景包括：

- 判断市场趋势强度：动量值越大，表明当前趋势越强
- 识别超买超卖状态：过高或过低的动量值可能预示价格即将回调
- 发现价格与指标的背离：这往往是趋势反转的早期信号
- 生成交易信号：动量方向变化可作为买卖决策的依据

数学公式

动量指标有多种计算方式，最常见的两种形式是：

1. 价格差值动量：直接计算当前价格与 n 期前价格的差值

$$M(n) = P_t - P_{t-n}$$

2. 价格比率动量：计算当前价格与 n 期前价格的比率

$$M(n) = \frac{P_t}{P_{t-n}} \times 100$$

其中：

- P_t 表示当前价格
- P_{t-n} 表示 n 期前的价格
- n 为计算周期，常见取值为 10、12 或 20

在技术分析软件中，通常默认使用价格差值动量。当动量值大于 0 时，表示价格呈上升趋势；当动量值小于 0 时，表示价格呈下降趋势。动量值的绝对值越大，表明价格变动的速度越快。

代码实现及可视化

```
mom <- momentum(Cl(TSLA), n = 1)
```

```
mom <- momentum(Cl(TSLA), n = 10)
```

AI 引导词

1. 介绍「Momentum」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.3.10 价格震荡指标 (PO) 指标含义

价格振荡指标 (Price Oscillator, PO) 是一种基于价格的技术分析工具，用于衡量两个不同周期移动平均线之间的差异。它通过计算短期移动平均线与长期移动平均线的差值或百分比差异，来判断市场趋势的强度和方向变化。

PO 指标的核心思想来源于移动平均线交叉策略。当短期移动平均线向上穿越长期移动平均线时，表明短期价格上涨速度快于长期趋势，市场处于上升趋势；反之则表明市场可能进入下降趋势。PO 指标将这种交叉关系量化，使交易者能够更直观地观察趋势强度和潜在转折点。

主要作用：

1. 趋势确认：PO 值为正时，表明市场处于上升趋势；为负时则处于下降趋势
2. 超买/超卖识别：PO 值远离零轴时，表明市场处于超买/超卖状态，可能面临回调
3. 背离信号：价格创新高/新低但 PO 未能同步，预示趋势可能即将反转
4. 交易信号：PO 值穿越零轴或与价格形成背离时，可作为买卖决策依据

与 MACD 指标相比，PO 使用简单移动平均线 (SMA) 而非指数移动平均线 (EMA)，计算方式更为直接，能更清晰地反映价格趋势变化。

数学公式

价格振荡指标有两种常见的计算方式：

1. 差值形式

$$PO(t) = SMA_{short}(t) - SMA_{long}(t)$$

2. 百分比形式

$$PO(t) = \left(\frac{SMA_{short}(t) - SMA_{long}(t)}{SMA_{long}(t)} \right) \times 100\%$$

其中：

- $SMA_{short}(t)$ 表示 t 时刻的短期移动平均线值
- $SMA_{long}(t)$ 表示 t 时刻的长期移动平均线值
- 常用的短期周期为 12，长期周期为 26（与 MACD 保持一致以便对比）

当 PO 值大于 0 时，短期均线在长期均线之上，市场处于上升趋势；当 PO 值小于 0 时，短期均线在长期均线之下，市场处于下降趋势。PO 的绝对值

越大，表明趋势越强。

代码实现及可视化

```
source("PO.R")
po <- PO(Cl(TSLA))
tail(po)

##          PO
## 2025-04-24 -7.2550641
## 2025-04-25 -3.8856410
## 2025-04-28 -4.6541026
## 2025-04-29 -3.0177564
## 2025-04-30 -0.6752564
## 2025-05-01  1.9653205
```

AI 引导词

1. 介绍「PO」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.3.11 去趋势价格震荡指标（DPO） 指标含义

去趋势价格震荡指标，De-Trended Price Oscillator 是一种技术分析工具，用于消除价格趋势的影响，帮助识别潜在的市场转折点和周期性波动。它通过将价格与移动平均线进行比较，测量价格与其趋势之间的偏离程度，从而发现超买或超卖情况。

DPO 的主要特点：

- 消除长期趋势的影响，专注于短期价格波动
- 帮助识别价格周期的顶部和底部
- 可用于判断市场是否处于超买或超卖状态

数学公式

DPO 的计算公式如下：

$$DPO(n) = Price(t) - MA\left(t - \left(\frac{n}{2} + 1\right)\right)$$

其中：

- $DPO(n)$ 是 n 周期的去趋势价格震荡指标
- $Price(t)$ 是当前时间 t 的价格
- MA 是移动平均线（通常使用简单移动平均线 SMA）
- n 是计算周期数
- $t - (\frac{n}{2} + 1)$ 是移动平均线的时间偏移

代码实现及可视化

下面使用 R 语言和 TTR 包对特斯拉 (TSLA) 股票进行 DPO 分析。

```
priceDPO <- DPO(TSLA[, "Close"])
volumeDPO <- DPO(TSLA[, "Volume"])

# 计算 DPO (默认周期 n=20)
tsla_dpo <- DPO(Cl(TSLA)) # 使用收盘价计算 DPO

# 将结果合并到原始数据中
tsla_data <- data.frame(
  Date = index(TSLA),
  Close = as.numeric(Cl(TSLA)),
  DPO = as.numeric(tsla_dpo)
)

# 查看计算结果
head(tsla_data)

##           Date   Close      DPO
## 1 2020-01-02 28.6840     NA
## 2 2020-01-03 29.5340     NA
## 3 2020-01-06 30.1027     NA
## 4 2020-01-07 31.2707 -0.90784
## 5 2020-01-08 32.8093  0.07589
```

```
## 6 2020-01-09 32.0893 -1.09404

# 绘制收盘价图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue") +
  labs(title = "特斯拉 (TSLA) 收盘价",
       x = "日期",
       y = "收盘价 ($)") +
  theme_minimal()

# 绘制 DPO 图表
p2 <- ggplot(tsla_data, aes(x = Date, y = DPO)) +
  geom_line(color = "red") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray") +
  labs(title = "特斯拉 (TSLA) DPO (20 天周期)",
       x = "日期",
       y = "DPO 值") +
  theme_minimal()

# 显示图表
p1 / p2
```



AI 引导词

“介绍 DPO 的意义、对应公式的 Latex 代码和基于数据 TSLA 及 TTR 包的 R 语言实例，结果请返回 RMarkdown 代码。”

3.1.3.12 三种指数平滑震荡指标 (TRIX) 指标含义

TRIX 指标 (Triple Exponential Average) 是一种经过三次指数平滑处理的动量指标，由 Jack Hutson 开发。它通过多重平滑有效过滤短期价格波动，专注于识别中长期趋势，特别适合用于判断市场的超买超卖状态和潜在反转点。

核心特点：

1. 减少噪音：三重指数平滑显著降低短期波动的干扰
2. 趋势识别：对中长期趋势变化反应敏感
3. 信号明确：指标线与信号线的交叉，以及与零轴的穿越，可生成清晰的交易信号
4. 提前预警：与价格的背离现象可提前预示趋势反转

数学公式

TRIX 指标的计算分为以下几个步骤：

1. 计算价格的第一次 EMA

$$EMA_1 = EMA(Price, n)$$

其中：

- n 为计算周期，通常为 15 或 30
- $Price$ 为收盘价

2. 计算第一次 EMA 的第二次 EMA

$$EMA_2 = EMA(EMA_1, n)$$

3. 计算第二次 EMA 的第三次 EMA

$$EMA_3 = EMA(EMA_2, n)$$

4. 计算 TRIX 值

$$TRIX = \frac{EMA_3 - EMA_3(-1)}{EMA_3(-1)} \times 100$$

其中：

- $EMA_3(-1)$ 为前一周期的 EMA_3 值

5. 计算 TRIX 信号线

$$Signal = EMA(TRIX, m)$$

其中：

- m 为信号线周期，通常为 9

```
tsla_trix <- TRIX(TSLA[, "Close"])
tsla_trix4 <- TRIX(TSLA[, "Close"],
                     maType=list(list(SMA),
                                  list(EMA, wilder=TRUE),
                                  list(SMA),
                                  list(DEMA)))

# 将结果合并到数据框中
tsla_data <- data.frame(
  Date = index(TSLA),
  Close = as.numeric(C1(TSLA)),
  TRIX = as.numeric(tsla_trix[,1]), # TRIX 线
  Signal = as.numeric(tsla_trix[,2]) # 信号线
)

# 移除 NA 值
tsla_data <- na.omit(tsla_data)

# 查看计算结果
head(tsla_data)

##           Date   Close      TRIX    Signal
## 67 2020-04-07 36.3633 -0.9373789 -0.8685113
## 68 2020-04-08 36.5893 -0.9272980 -0.8802687
## 69 2020-04-09 38.2000 -0.9057845 -0.8853718
## 70 2020-04-13 43.3967 -0.8645276 -0.8812030
## 71 2020-04-14 47.3260 -0.8001479 -0.8649920
## 72 2020-04-15 48.6553 -0.7160598 -0.8352055

# 创建价格图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = " 特斯拉 (TSLA) 收盘价",
```

```
x = "日期",
y = "收盘价 ($)" +
theme_minimal(base_family = "SimHei")

# 创建 TRIX 指标图表
p2 <- ggplot(tsla_data, aes(x = Date)) +
  geom_line(aes(y = TRIX), color = "red", size = 1) +
  geom_line(aes(y = Signal), color = "green", size = 1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray", alpha = 0.7) +
  labs(title = "TRIX 指标",
       x = "日期",
       y = "指标值") +
  theme_minimal(base_family = "SimHei")

# 使用 patchwork 组合图表
library(patchwork)
p1 / p2
```



交易信号分析

基于 TRIX 指标的交易信号通常包括：

```
# 生成交叉信号
tsla_data$Cross_Signal <- ifelse(
  tsla_data$TRIX > tsla_data$Signal & lag(tsla_data$TRIX) <= lag(tsla_data$Signal), "买入",
  ifelse(
    tsla_data$TRIX < tsla_data$Signal & lag(tsla_data$TRIX) >= lag(tsla_data$Signal), "卖出",
    )
)
)

# 统计信号数量
table(tsla_data$Cross_Signal)

## 
## 买入 卖出 持有
## 24 23 1226

# 可视化交叉信号
ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  geom_point(data = subset(tsla_data, Cross_Signal == "买入"),
             aes(color = "买入"), size = 3, shape = 16) +
  geom_point(data = subset(tsla_data, Cross_Signal == "卖出"),
             aes(color = "卖出"), size = 3, shape = 17) +
  scale_color_manual(values = c("买入" = "green", "卖出" = "red")) +
  labs(title = "TRIX 指标交叉信号",
       x = "日期",
       y = "收盘价 ($)",
       color = "交易信号") +
  theme_minimal(base_family = "SimHei")
```



零轴穿越信号分析

```
# 生成零轴穿越信号
tsla_data$Zero_Cross_Signal <- ifelse(
  tsla_data$TRIX > 0 & lag(tsla_data$TRIX) <= 0, "买入",
  ifelse(
    tsla_data$TRIX < 0 & lag(tsla_data$TRIX) >= 0, "卖出", "持有"
  )
)

# 统计信号数量
table(tsla_data$Zero_Cross_Signal)

## 
## 买入 卖出 持有
##   11   11 1251

# 可视化零轴穿越信号
ggplot(tsla_data, aes(x = Date, y = Close)) +
```

```
geom_line(color = "blue", size = 1) +
  geom_point(data = subset(tsla_data, Zero_Cross_Signal == "买入"),
             aes(color = "买入"), size = 3, shape = 16) +
  geom_point(data = subset(tsla_data, Zero_Cross_Signal == "卖出"),
             aes(color = "卖出"), size = 3, shape = 17) +
  scale_color_manual(values = c("买入" = "green", "卖出" = "red")) +
  labs(title = "TRIX 指标零轴穿越信号",
       x = "日期",
       y = "收盘价 ($)",
       color = "交易信号") +
  theme_minimal(base_family = "SimHei")
```



AI 引导词

“介绍 TRIX 指标指标含义、对应数学公式的 Latex 代码和基于数据 TSLA 及 TTR 包的 R 语言实例（包括计算指标部分、可视化部分），结果请返回 RMarkdown 代码。”

3.1.4 波动率指标

3.1.4.1 平均真实范围 (ATR) 指标含义

平均真实范围 (Average True Range, ATR) 是由技术分析大师威尔斯·威尔德 (J. Welles Wilder) 在 1978 年提出的一种技术指标，用于衡量市场价格的波动强度。与其他反映价格方向的指标不同，ATR 专注于量化价格波动的幅度，是衡量市场波动性的重要工具。

核心概念

真实范围 (TR): 是以下三个值中的最大值：

1. 当前最高价与最低价的差值
2. 当前最高价与前一收盘价的绝对差值
3. 当前最低价与前一收盘价的绝对差值

真实范围的设计旨在捕捉价格的跳空缺口和异常波动，提供更准确的波动度量。

平均真实范围 (ATR):

是真实范围的移动平均值，通常使用 14 天周期计算。ATR 值越高，表示价格波动越剧烈；反之则表示市场趋于平静。

主要应用场景

1. 波动率分析：判断市场当前的波动水平，高 ATR 值可能预示趋势即将加速或反转
2. 止损位设置：根据 ATR 值动态调整止损距离，在高波动市场中设置更宽的止损
3. 交易系统设计：结合 ATR 构建波动率自适应的交易策略
4. 市场状态判断：区分高波动市场与低波动市场，选择合适的交易策略

数学公式

平均真实范围 (ATR) 的数学公式

1. 真实范围 (TR) 的计算公式

$$TR = \max \left(\begin{array}{l} H_t - L_t \\ |H_t - C_{t-1}| \\ |L_t - C_{t-1}| \end{array} \right)$$

其中：

- H_t 表示当前周期的最高价
- L_t 表示当前周期的最低价
- C_{t-1} 表示前一周期的收盘价

2. 平均真实范围 (ATR) 的计算公式

威尔德推荐的原始计算方法是使用指数移动平均 (EMA)：

$$ATR(n) = \frac{(n-1) \times ATR_{t-1} + TR_t}{n}$$

其中：

- $ATR(n)$ 表示 n 周期的平均真实范围
- ATR_{t-1} 表示前一周期的 ATR 值
- TR_t 表示当前周期的真实范围
- n 为计算周期，通常取 14

现代计算中，也常用简单移动平均 (SMA) 来计算 ATR：

$$ATR(n) = \frac{\sum_{i=1}^n TR_i}{n}$$

代码实现及可视化

```
atr <- ATR(TSLA[, c("High", "Low", "Close")], n=14)
```

AI 引导词

1. 介绍「ATR」指标的意义；2. 对应 Latex 公式的 Rmd 代码；
3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.4.2 真实范围 (TR) 指标含义

真实范围 (True Range, TR) 是技术分析中衡量价格波动幅度的基本指标，由威尔斯·威尔德 (J. Welles Wilder) 在 1978 年提出。它是平均真实范围 (ATR) 的基础，用于量化市场价格的实际波动强度，特别适合捕捉价格跳空和异常波动。

核心概念

- 传统范围：简单地计算当日最高价与最低价的差值 (H-L)，但无法反映因隔夜跳空导致的价格缺口。
- 真实范围：通过考虑前一收盘价，弥补了传统范围的不足，能更准确地反映价格的实际波动幅度。

主要应用场景

1. 波动率分析：TR 值越高，市场波动性越强，反之则越弱。
2. 止损位设置：基于 TR 设置动态止损，避免因市场正常波动而被过早止损。
3. ATR 计算基础：TR 是计算 ATR 的核心输入，ATR 是 TR 的移动平均。
4. 突破确认：异常高的 TR 值可能预示价格突破或趋势加速。

数学公式

真实范围是以下三个值中的最大值：

$$TR = \max \left(\begin{array}{l} H_t - L_t \\ |H_t - C_{t-1}| \\ |L_t - C_{t-1}| \end{array} \right)$$

其中：

- H_t 表示当前周期的最高价
- L_t 表示当前周期的最低价
- C_{t-1} 表示前一周期的收盘价

公式解释

1. $H_t - L_t$: 当日最高价与最低价的差值，即传统范围。
2. $|H_t - C_{t-1}|$: 当日最高价与前一收盘价的绝对差值，捕捉向上跳空缺口。
3. $|L_t - C_{t-1}|$: 当日最低价与前一收盘价的绝对差值，捕捉向下跳空缺口。

通过取三者中的最大值，TR 确保了即使存在价格跳空，也能准确反映价格波动的真实幅度。

代码实现及可视化

```
tr <- TR(TSLA[,c("High","Low","Close")])
```

AI 引导词

1. 介绍「TR」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.4.3 布林带 (Bollinger Bands) 指标含义

布林带 (Bollinger Bands) 是由技术分析师约翰·布林格 (John Bollinger) 在 20 世纪 80 年代提出的一种技术分析工具，用于衡量价格波动和识别潜在的超买超卖区域。布林带由三条线组成：中轨 (通常是简单移动平均线)、上轨 (中轨加上一定倍数的标准差) 和下轨 (中轨减去一定倍数的标准差)。

核心概念

- 中轨 (Middle Band): 通常使用 20 日简单移动平均线 (SMA)，代表价格的趋势方向。
- 上轨 (Upper Band): 中轨加上一定倍数 (通常为 2) 的标准差，代表价格的压力线。
- 下轨 (Lower Band): 中轨减去一定倍数 (通常为 2) 的标准差，代表价格的支撑线。
- 带宽 (Band Width): 上轨与下轨之间的距离，反映市场波动性的变化。

主要应用场景

1. 趋势判断：价格位于上轨上方可能处于超买状态，位于下轨下方可能处于超卖状态。
2. 波动率分析：带宽收窄（布林带收紧）通常预示价格即将大幅波动，形成突破。
3. 支撑阻力位：上轨和下轨可视为动态的支撑和阻力位。
4. 买卖信号：价格突破上轨可能是买入信号，跌破下轨可能是卖出信号；价格与布林带的背离也可能预示趋势反转。

数学公式

布林带 (BBands) 的数学公式

1. 中轨计算

$$MB = SMA(n)$$

其中：

- MB 表示中轨 (Middle Band)
- $SMA(n)$ 表示 n 周期的简单移动平均线，通常 $n = 20$

2. 标准差计算

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (P_i - SMA(n))^2}{n}}$$

其中：

- P_i 表示第 i 期的价格
- $SMA(n)$ 表示 n 周期的简单移动平均线

3. 上轨和下轨计算

$$UB = MB + k \times \sigma$$

$$LB = MB - k \times \sigma$$

其中：

- UB 表示上轨 (Upper Band)
- LB 表示下轨 (Lower Band)
- k 是倍数因子，通常取 2，表示 2 个标准差

4. 带宽计算

$$BW = \frac{UB - LB}{MB} \times 100\%$$

其中：

- BW 表示带宽 (Band Width)，以百分比形式表示

代码实现及可视化

```
bbands.HLC <- BBands( TSLA[,c("High","Low","Close")] )
bbands.close <- BBands( TSLA[,"Close"] )
```

AI 引导词

1. 介绍「BBands」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.4.4 价格通道指标 (PBands) 指标含义

PBands 指标 (Price Bands)，又称价格通道指标，是基于移动平均线和标准差的技术分析工具，用于衡量价格波动的范围和强度。它由三条线组成：中轨（通常是简单或指数移动平均线）、上轨（中轨加上一定倍数的标准差）和下轨（中轨减去一定倍数的标准差）。

核心思想：

1. 波动区间识别：通过上下轨界定价格的正常波动范围
2. 超买超卖判断：价格触及或突破上轨时可能超买，触及或跌破下轨时可能超卖
3. 趋势强度评估：通道宽度变化反映市场波动性的增减

4. 突破信号捕捉：价格突破通道可能预示新趋势的开始

主要用途：

1. 识别价格波动的极值区域
2. 发现潜在的反转或突破信号
3. 评估市场波动性变化
4. 辅助设置止损和止盈点位

数学公式

PBands 指标的计算分为以下几个步骤：

1. 计算中轨线 (MA)

$$MA = SMA(C, n) \quad MA = EMA(C, n)$$

其中：

- C 为收盘价
- n 为计算周期，通常为 20
- SMA 为简单移动平均， EMA 为指数移动平均

2. 计算标准差 (SD)

$$SD = \sqrt{\frac{\sum_{i=1}^n (C_i - MA_i)^2}{n}}$$

其中：

- C_i 为第 i 日收盘价
- MA_i 为第 i 日移动平均值

3. 计算上轨线 (Upper Band)

$$Upper Band = MA + k \times SD$$

4. 计算下轨线 (Lower Band)

$$\text{Lower Band} = MA - k \times SD$$

其中：

- k 为倍数系数，通常为 2，表示约 95% 的价格波动会落在上下轨之间

代码实现及可视化

```
# 使用 TTR 包计算 PBands 指标
tsla_pbands <- BBands(Cl(TSLA), n = 20, sd = 2) # 默认参数: n=20, sd=2

# 将结果合并到数据框中
tsla_data <- data.frame(
  Date = index(TSLA),
  Close = as.numeric(Cl(TSLA)),
  Upper = as.numeric(tsla_pbands[,1]), # 上轨
  Middle = as.numeric(tsla_pbands[,2]), # 中轨
  Lower = as.numeric(tsla_pbands[,3]), # 下轨
  PercentB = as.numeric(tsla_pbands[,4]) # %B 指标: (Close-Lower)/(Upper-Lower)
)

# 移除 NA 值
tsla_data <- na.omit(tsla_data)

# 查看计算结果
head(tsla_data)
```

	Date	Close	Upper	Middle	Lower	PercentB
## 20	2020-01-30	42.7207	27.81238	34.83758	41.86277	1.0610607
## 21	2020-01-31	43.3713	28.21070	35.57194	42.93318	1.0297587
## 22	2020-02-03	52.0000	26.90619	36.69524	46.48429	1.2817286
## 23	2020-02-04	59.1373	24.75174	38.14697	51.54220	1.2834999
## 24	2020-02-05	48.9800	25.23713	39.03244	52.82774	0.8605417
## 25	2020-02-06	49.9307	25.62709	39.88851	54.14992	0.8520757

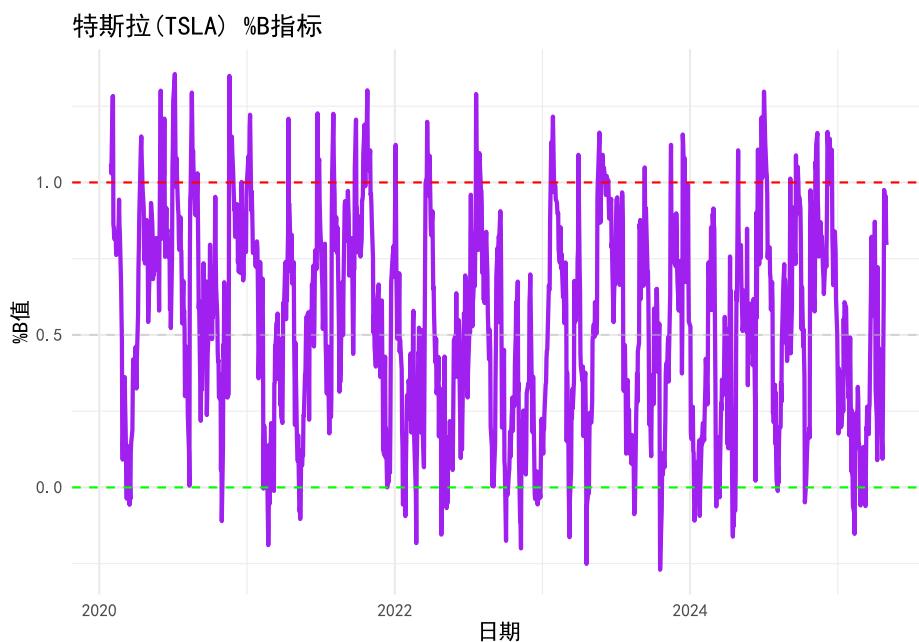
```
# 创建 PBands 图表
ggplot(tsla_data, aes(x = Date)) +
  geom_line(aes(y = Close), color = "blue", size = 1) +
  geom_line(aes(y = Upper), color = "red", linetype = "dashed") +
  geom_line(aes(y = Middle), color = "gray50", linetype = "solid") +
  geom_line(aes(y = Lower), color = "red", linetype = "dashed") +
  labs(
    title = "特斯拉 (TSLA) 价格通道指标 (PBands)",
    x = "日期",
    y = "价格"
  ) +
  theme_minimal(base_family = "SimHei")
```



```
# 创建%B 指标图表
p2 <- ggplot(tsla_data, aes(x = Date, y = PercentB)) +
  geom_line(color = "purple", size = 1) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "green") +
```

```
geom_hline(yintercept = 0.5, linetype = "dashed", color = "gray", alpha = 0.5) +
  labs(
    title = "特斯拉 (TSLA) %B 指标",
    x = "日期",
    y = "%B 值"
  ) +
  theme_minimal(base_family = "SimHei")
```

p2



交易信号分析

基于 PBands 指标的交易信号通常包括：

```
# 生成 PBands 信号
tsla_data$PBands_Signal <- ifelse(
  tsla_data$Close > tsla_data$Upper, "超买",
  ifelse(
    tsla_data$Close < tsla_data$Lower, "超卖",
```

```
ifelse(
  tsla_data$Close > tsla_data$Middle, "看涨", "看跌"
)
)
)

# 统计信号数量
table(tsla_data$PBands_Signal)

## 
## 超买 超卖
## 1252 69

# 可视化 PBands 信号
ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  geom_line(aes(y = Upper), color = "red", linetype = "dashed") +
  geom_line(aes(y = Middle), color = "gray50", linetype = "solid") +
  geom_line(aes(y = Lower), color = "red", linetype = "dashed") +
  geom_point(data = subset(tsla_data, PBands_Signal == "超买"),
             aes(color = "超买"), size = 1, shape = 16) +
  geom_point(data = subset(tsla_data, PBands_Signal == "超卖"),
             aes(color = "超卖"), size = 1, shape = 17) +
  scale_color_manual(values = c("超买" = "red", "超卖" = "green", "看涨" = "gray", "看跌" = "black")) +
  labs(
    title = "特斯拉 (TSLA) PBands 指标交易信号",
    x = "日期",
    y = "收盘价 ($)",
    color = "信号类型"
) +
  theme_minimal(base_family = "SimHei")
```

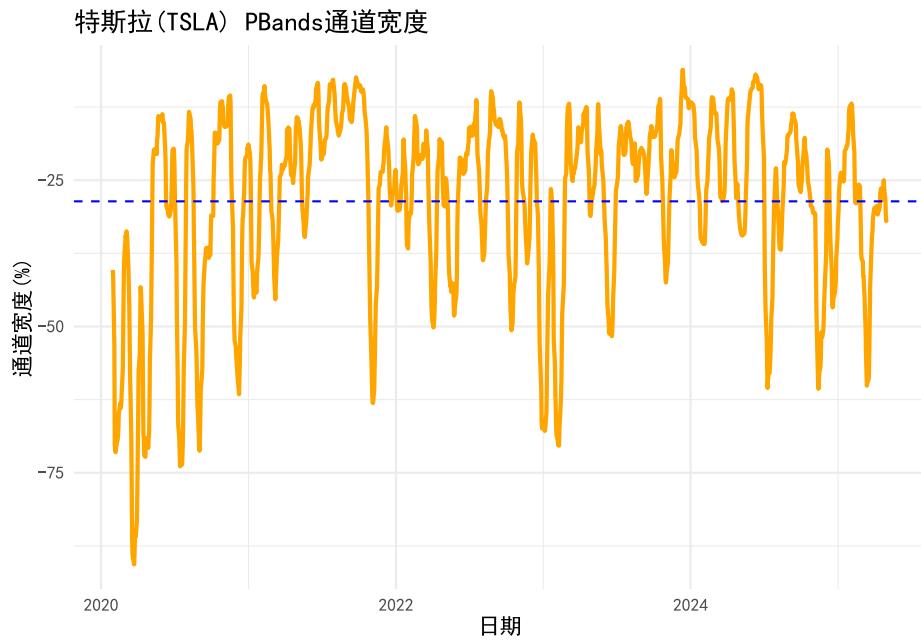


通道宽度分析

通道宽度是衡量市场波动性的重要指标：

```
# 计算通道宽度
tsla_data$BandWidth <- (tsla_data$Upper - tsla_data$Lower) / tsla_data$Middle * 100

# 可视化通道宽度
ggplot(tsla_data, aes(x = Date, y = BandWidth)) +
  geom_line(color = "orange", size = 1) +
  geom_hline(yintercept = mean(tsla_data$BandWidth), linetype = "dashed", color = "blue")
  labs(
    title = "特斯拉 (TSLA) PBands 通道宽度",
    x = "日期",
    y = "通道宽度 (%)"
  ) +
  theme_minimal(base_family = "SimHei")
```



```
# 识别窄幅震荡区域（可能预示突破）
tsla_data$NarrowRange <- ifelse(
  tsla_data$BandWidth < quantile(tsla_data$BandWidth, 0.2), "窄幅", "正常"
)

# 统计窄幅震荡天数
table(tsla_data$NarrowRange)

## 
## 正常 窄幅
## 1057 264
```

基于 PBands 指标的交易策略通常包括：

- 超买超卖策略：
 - 当价格突破上轨时，视为超买，可能出现回调，考虑卖出
 - 当价格跌破下轨时，视为超卖，可能出现反弹，考虑买入
- 突破策略：
 - 当价格从下方突破上轨时，视为强势上涨信号，考虑买入

- 当价格从上方跌破下轨时，视为强势下跌信号，考虑卖出
- 通道宽度策略：
 - 当通道宽度收缩至历史低位时，可能预示即将出现大幅波动
 - 通道宽度开始扩张时，结合价格方向确认新趋势的形成
- %B 指标策略：
 - $\%B > 1$ 时为超买状态
 - $\%B < 0$ 时为超卖状态
 - $\%B$ 从下方穿越 0.5 时为看涨信号
 - $\%B$ 从上方穿越 0.5 时为看跌信号

需要注意的是：

PBands 指标在震荡市场中表现较好，但在单边趋势中可能产生频繁的虚假信号参数 (n 和 sd) 可根据交易风格和标的特性进行调整建议结合其他指标 (如 RSI、MACD) 共同使用，提高信号可靠性 PBands 指标是一种简单而有效 的技术分析工具，通过界定价格波动区间，能够帮助交易者识别潜在的交易机会和市场状态变化。

AI 引导词

1. 介绍 PBands 指标的意义；2. 对应 Latex 公式的 Rmd 代码；
3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.4.5 肯特纳通道通道 (Keltner Channels) 指标含义

Keltner 通道 (Keltner Channels) 是一种技术分析工具，用于识别证券价格的波动范围和趋势方向。与布林带 (Bollinger Bands) 类似，Keltner 通道也由三条线组成：中轨、上轨和下轨。但两者的计算方法不同：布林带使用标准差衡量波动性，而 Keltner 通道使用平均真实范围 (ATR)。

核心概念

- 中轨 (Middle Line)：通常使用 20 日指数移动平均线 (EMA)，代表价格的趋势方向。
- 上轨 (Upper Channel Line)：中轨加上一定倍数的 ATR，代表价格的压力线。

- 下轨 (Lower Channel Line): 中轨减去一定倍数的 ATR，代表价格的支撑线。
- 通道宽度 (Channel Width): 上轨与下轨之间的距离，反映市场波动性的变化。

主要应用场景

1. 趋势判断: 价格位于上轨上方可能处于强势上涨趋势, 位于下轨下方可能处于强势下跌趋势。
2. 波动率分析: 通道收窄表示市场波动性降低, 可能预示价格即将突破; 通道扩张表示波动性增加。
3. 支撑阻力位: 上轨和下轨可视为动态的支撑和阻力位。
4. 买卖信号: 价格突破上轨可能是买入信号, 跌破下轨可能是卖出信号; 价格与通道的背离也可能预示趋势反转。

数学公式

Keltner 通道的数学公式

1. 中轨计算

$$ML = EMA(n)$$

其中:

- ML 表示中轨 (Middle Line)
- $EMA(n)$ 表示 n 周期的指数移动平均线, 通常 $n = 20$

2. 平均真实范围 (ATR) 计算

$$ATR(n) = \frac{(n-1) \times ATR_{t-1} + TR_t}{n}$$

其中:

- $ATR(n)$ 表示 n 周期的平均真实范围
- ATR_{t-1} 表示前一周期的 ATR 值
- TR_t 表示当前周期的真实范围

- n 为计算周期，通常取 14

3. 上轨和下轨计算

$$UCL = ML + k \times ATR(m)$$

$$LCL = ML - k \times ATR(m)$$

其中：

- UCL 表示上轨 (Upper Channel Line)
- LCL 表示下轨 (Lower Channel Line)
- k 是倍数因子，通常取 2 或 2.5，表示 ATR 的倍数
- m 是 ATR 的计算周期，通常取 10 或 20

4. 通道宽度计算

$$CW = \frac{UCL - LCL}{ML} \times 100\%$$

其中：

- CW 表示通道宽度 (Channel Width)，以百分比形式表示

代码实现及可视化

```
kc <- keltnerChannels(TSLA[, c("High", "Low", "Close")])
```

AI 引导词

1. 介绍「Keltner 通道」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.4.6 唐奇安通道 (Donchian Channel) 指标含义

唐奇安通道是由技术分析师理查德·唐奇安 (Richard Donchian) 提出的趋势跟踪指标，主要用于识别价格的波动区间和潜在的趋势突破点。该指标通

过计算一段时间内（通常为 N 天）的最高价、最低价和中间值（或简单移动平均线）来构建三条轨道：

- 上轨 (Upper Channel): N 期内的最高价，代表短期价格的阻力位；
- 下轨 (Lower Channel): N 期内的最低价，代表短期价格的支撑位；
- 中轨 (Middle Line): 通常为上轨和下轨的中间值，或 N 期收盘价的简单移动平均线（可选）。

当价格突破上轨时，可能预示着看涨趋势的开始；突破下轨时，则可能预示着看跌趋势的开始。唐奇安通道常用于趋势交易策略（如海龟交易法则），帮助交易者识别入场和离场信号。

数学公式

唐奇安通道的上轨和下轨公式分别为：

$$\text{Upper}(t, N) = \max_{i=0}^{N-1} \text{High}(t - i)$$

$$\text{Lower}(t, N) = \min_{i=0}^{N-1} \text{Low}(t - i)$$

代码实现及可视化

```
dc <- DonchianChannel(TSLA[, c("High", "Low")])
tail(dc)

##           high     mid    low
## 2025-04-24 262.49 242.640 222.79
## 2025-04-25 286.85 254.820 222.79
## 2025-04-28 294.86 258.825 222.79
## 2025-04-29 294.86 258.825 222.79
## 2025-04-30 294.86 258.825 222.79
## 2025-05-01 294.86 258.825 222.79
```

AI 引导词

1. 介绍「唐奇安通道」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.4.7 相对波动率指数 (RVI) 指标含义

相对波动率指数 (Relative Volatility Index, RVI) 是由 Donald Dorsey 在 1993 年提出的一种技术分析工具，用于衡量价格变动的波动性。与传统的相对强弱指数 (RSI) 关注价格变动方向不同，RVI 专注于价格变动的幅度，通过比较上涨和下跌期间的波动性来识别市场状态。

相对波动率指数有如下作用：

- 波动性测量：RVI 通过计算价格上涨和下跌期间的标准差来衡量波动性。
- 超买超卖信号：与 RSI 类似，RVI 值高于 70 通常表示超买状态，低于 30 表示超卖状态。
- 趋势确认：RVI 高于 50 表示上涨波动性占优，可能预示上升趋势；低于 50 表示下跌波动性占优，可能预示下降趋势。
- 背离信号：当价格创新高而 RVI 未能同步创新高，或价格创新低而 RVI 未能同步创新低时，可能出现趋势反转信号。

主要应用场景：

1. 波动强度分析：识别市场波动性的变化，判断市场是处于平静还是剧烈波动状态。
2. 超买超卖识别：结合价格走势，识别潜在的超买超卖区域。
3. 趋势确认与反转预警：通过 RVI 与价格的背离，提前预警趋势可能的反转。
4. 交易信号生成：当 RVI 穿越 50 时，可能产生买入或卖出信号。

数学公式

相对波动率指数 (RVI) 的数学公式

1. 计算价格变动

$$\text{Change}_t = \text{Close}_t - \text{Close}_{t-1}$$

其中：

- Change_t 表示第 t 期的价格变动
- Close_t 表示第 t 期的收盘价
- Close_{t-1} 表示前一期的收盘价

2. 分离上涨和下跌变动

$$\text{Up}_t = \begin{cases} \text{Change}_t, & \text{如果 } \text{Change}_t > 0 \\ 0, & \text{否则} \end{cases}$$

$$\text{Down}_t = \begin{cases} |\text{Change}_t|, & \text{如果 } \text{Change}_t < 0 \\ 0, & \text{否则} \end{cases}$$

3. 计算上涨和下跌变动的指数移动平均 (EMA)

$$\text{EMA_Up}_t = \frac{\text{Up}_t + 2 \times \text{EMA_Up}_{t-1}}{3}$$

$$\text{EMA_Down}_t = \frac{\text{Down}_t + 2 \times \text{EMA_Down}_{t-1}}{3}$$

其中：

- EMA_Up_t 表示第 t 期上涨变动的指数移动平均
- EMA_Down_t 表示第 t 期下跌变动的指数移动平均

4. 计算 RVI

$$\text{RVI}_t = 100 \times \frac{\text{EMA_Up}_t}{\text{EMA_Up}_t + \text{EMA_Down}_t}$$

代码实现及可视化

```
source("RVI.r")
rvi <- RVI(Cl(TSLA), n=14, ema.n=3, keepNA = TRUE)
```

1. 介绍「RVI」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.4.8 历史波动率 (HV) 指标含义

历史波动率 (Historical Volatility, HV) 是衡量资产价格在过去一段时间内波动程度的指标。它反映了价格变动的剧烈程度，通常以年化标准差表示。HV 是期权定价、风险评估和交易策略设计的重要输入参数，高波动率意味着价格波动剧烈，低波动率则表示价格相对稳定。

数学公式

历史波动率的计算步骤如下：

1. 计算对数收益率：

$$r_t = \ln \left(\frac{P_t}{P_{t-1}} \right)$$

其中， P_t 是第 t 期的价格。

2. 计算平均收益率：

$$\bar{r} = \frac{1}{n} \sum_{t=1}^n r_t$$

3. 计算样本标准差：

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{t=1}^n (r_t - \bar{r})^2}$$

4. 年化波动率：

$$HV = \sigma \times \sqrt{T}$$

其中, T 是一年中的交易周期数 (如日交易取 252, 周交易取 52)。

代码实现及可视化

```

ohlc <- TSLA[,c("Open","High","Low","Close")]
vClose <- volatility(ohlc, calc="close")
vClose0 <- volatility(ohlc, calc="close", mean0=TRUE)
vGK <- volatility(ohlc, calc="garman")
vParkinson <- volatility(ohlc, calc="parkinson")
vRS <- volatility(ohlc, calc="rogers")

# 计算 TSLA 的历史波动率 (20 日和 50 日)
ohlc <- TSLA[,c("Open","High","Low","Close")]
tsla_hv20 <- volatility(ohlc, calc="close",n = 20)
tsla_hv50 <- volatility(ohlc, calc="close",n = 50)

# 准备数据框
tsla_df <- data.frame(
  Date = index(tsla_hv20),
  Close = as.numeric(Cl(TSLA)),
  HV_20 = as.numeric(tsla_hv20),
  HV_50 = as.numeric(tsla_hv50)
) %>% na.omit()

# 创建价格图表
p1 <- ggplot(tsla_df, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "TSLA 收盘价", y = "价格") +
  theme_minimal(base_family = "SimHei")

# 创建历史波动率图表
p2 <- ggplot(tsla_df, aes(x = Date)) +

```

```
geom_line(aes(y = HV_20), color = "red", size = 0.8) +
geom_line(aes(y = HV_50), color = "green", size = 0.8) +
labs(title = "TSLA 历史波动率", y = " 波动率") +
theme_minimal(base_family = "SimHei")
```

组合图表

p1 / p2

TSLA收盘价



TSLA历史波动率

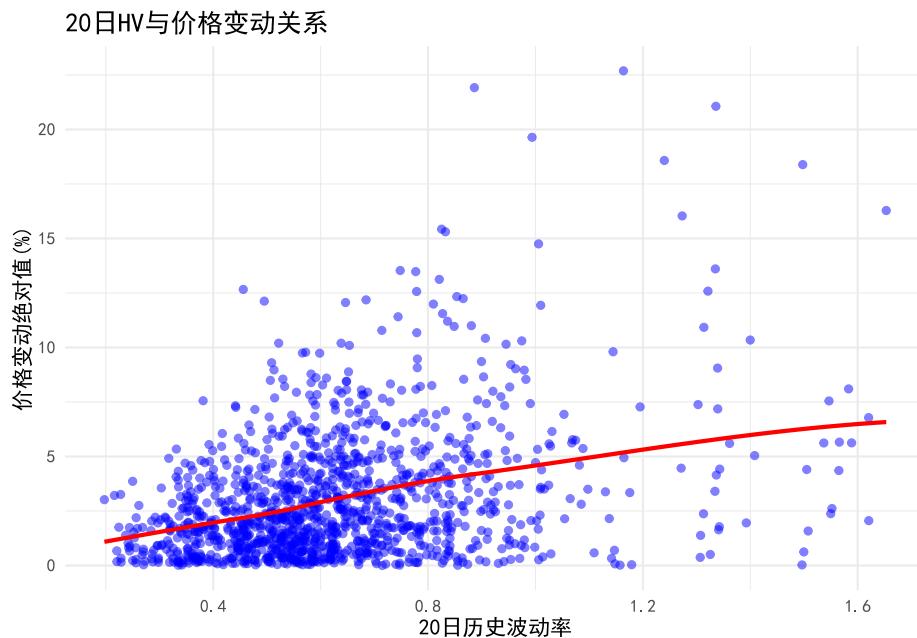


计算价格变动百分比

```
tsla_df$PriceChange <- abs((tsla_df$Close - lag(tsla_df$Close)) / lag(tsla_df$Close) *
```

绘制波动率与价格变动关系图

```
ggplot(tsla_df, aes(x = HV_20, y = PriceChange)) +
  geom_point(color = "blue", alpha = 0.5) +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  labs(title = "20 日 HV 与价格变动关系",
       x = "20 日历史波动率",
       y = " 价格变动绝对值 (%)") +
  theme_minimal(base_family = "SimHei")
```



AI 引导词

1. 介绍「HV」指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.4.9 蔡金波动率 (Chaikin Volatility) 指标含义

Chaikin 波动率指标由 Marc Chaikin 提出，通过衡量价格区间（最高价与最低价之差）的变化率来评估市场波动性。与传统波动率指标不同，Chaikin 波动率关注价格区间的扩张和收缩，能够提前预警市场趋势的变化。当 Chaikin 波动率上升时，表示市场波动性增加，趋势可能加速；当其下降时，表示市场趋于平静，可能预示趋势即将结束。

数学公式

Chaikin 波动率的计算步骤如下：

1. 计算价格区间：

$$R_t = H_t - L_t$$

其中, H_t 和 L_t 分别是第 t 期的最高价和最低价。

2. 对价格区间进行移动平均:

$$MA(R_t) = SMA(R_t, n)$$

其中, SMA 表示简单移动平均, n 是移动平均的周期。

3. 计算移动平均的百分比变化:

$$\text{Chaikin Volatility}_t = \frac{\text{MA Range}_t - \text{MA Range}_{t-p}}{\text{MA Range}_{t-p}} \times 100\%$$

其中, p 是计算变化率的周期。

更常见的实现方式是对移动平均取差分:

$$\text{Chaikin Volatility}_t = EMA(\text{MA Range}, m)$$

其中, EMA 表示指数移动平均, m 是平滑周期。

```
tsla_cv <- chaikinVolatility(TSLA[, c("High", "Low")])
```

代码实现及可视化

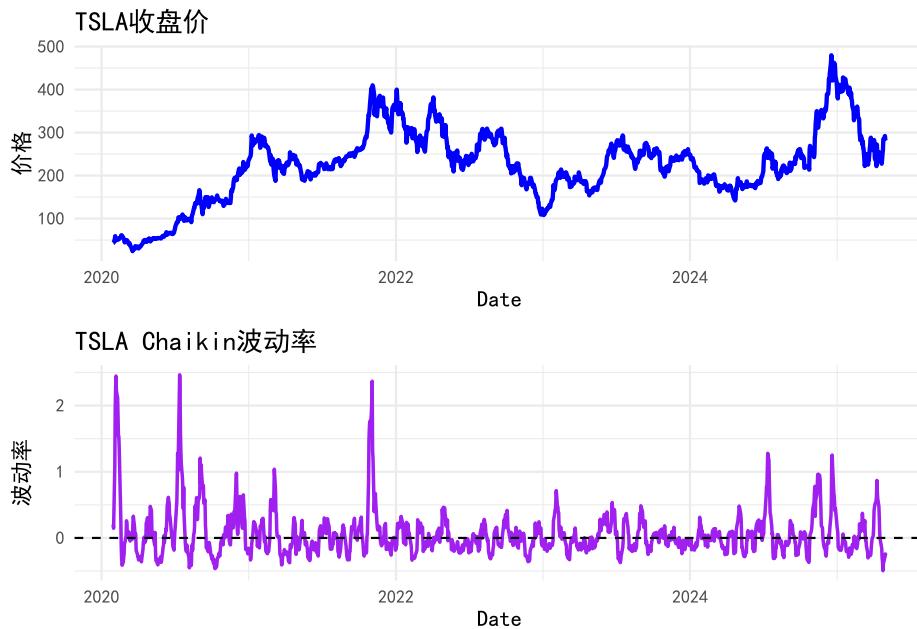
```
# 准备数据框
tsla_df <- data.frame(
  Date = index(tsla_cv),
  Close = as.numeric(Cl(TSLA)),
  ChaikinVol = as.numeric(tsla_cv)
) %>% na.omit()

# 创建价格图表
```

```
p1 <- ggplot(tsla_df, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "TSLA 收盘价", y = "价格") +
  theme_minimal(base_family = "SimHei")

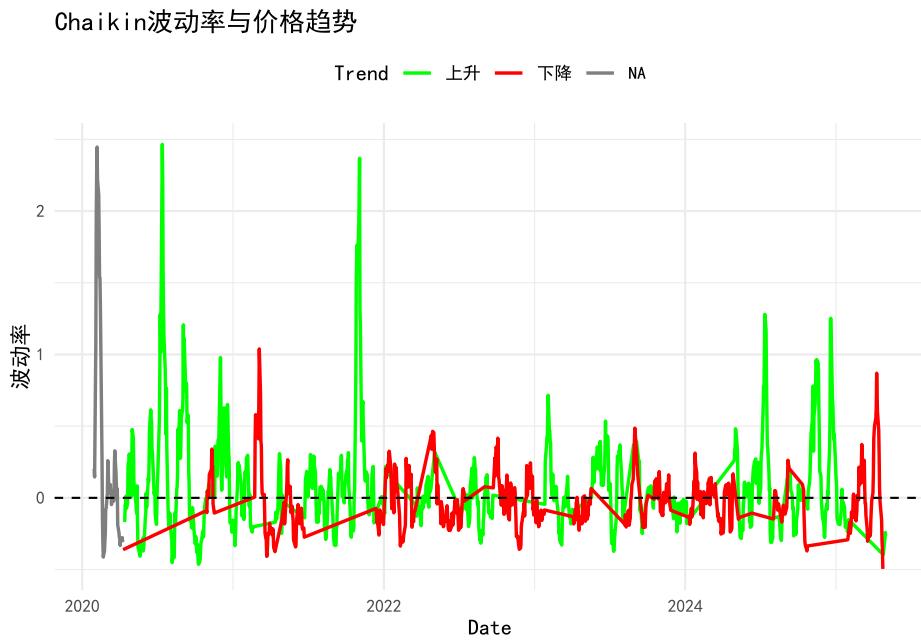
# 创建 Chaikin 波动率图表
p2 <- ggplot(tsla_df, aes(x = Date, y = ChaikinVol)) +
  geom_line(color = "purple", size = 0.8) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  labs(title = "TSLA Chaikin 波动率", y = "波动率") +
  theme_minimal(base_family = "SimHei")

# 组合图表
p1 / p2
```



```
# 添加趋势判断（简单移动平均）
tsla_df$SMA50 <- SMA(tsla_df$Close, n = 50)
tsla_df$Trend <- ifelse(tsla_df$Close > tsla_df$SMA50, "上升", "下降")
```

```
# 绘制 Chaikin 波动率与趋势关系
ggplot(tsla_df, aes(x = Date, y = ChaikinVol, color = Trend)) +
  geom_line(size = 0.8) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  scale_color_manual(values = c("上升" = "green", "下降" = "red")) +
  labs(title = "Chaikin 波动率与价格趋势",
       y = "波动率") +
  theme_minimal(base_family = "SimHei") +
  theme(legend.position = "top")
```



AI 引导词

1. 介绍 Chaikin 波动率 (Chaikin Volatility) 指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.4.10 威廉累积/派发线 (williamsAD) 指标含义

Williams 累积/派发线 (Accumulation/Distribution Line) 是由 Larry

Williams 开发的一种技术分析工具，用于衡量资金流向和市场买卖压力。其核心思想是通过比较收盘价与当日价格区间的关系，判断市场主力资金的动向：

- 累积 (Accumulation): 当收盘价接近当日最高价时，表明买方力量较强，产生正的资金流入
- 派发 (Distribution): 当收盘价接近当日最低价时，表明卖方力量较强，产生负的资金流入

AD 线通过对每日资金流向的累积计算，形成一条趋势线，可用于：

1. 确认价格趋势的强度和持续性
2. 发现价格与指标之间的背离信号（潜在反转信号）
3. 衡量市场买卖压力的变化

数学公式

Williams AD 指标的计算公式如下：

1. 方向性运动 (Directional Movement) 计算

$$CMF = \frac{Close - Low}{High - Low} \quad (Close > Low)$$

$$CMF = \frac{Close - High}{High - Low} \quad (Close < High)$$

其中：

- *High*: 当日最高价
- *Low*: 当日最低价
- *Close*: 当日收盘价
- *CMF* (Close Location Value): 收盘价位置值，范围为 [-1, 1]

2. 累积/派发值计算

$$AD = \sum_{i=1}^n (CMF_i \times Volume_i)$$

其中：

- $Volume_i$: 第 i 日的成交量
- AD : Williams 累积/派发线, 初始值通常设为 0

```
tsla_ad <- williamsAD(TSLA[,c("High","Low","Close")])
```

```
# 将结果合并到数据框中
tsla_data <- data.frame(
  Date = index(TSLA),
  Close = as.numeric(Cl(TSLA)),
  Volume = as.numeric(Vo(TSLA)),
  AD = as.numeric(tsla_ad)
)

# 查看计算结果
head(tsla_data)
```

```
##           Date   Close      Volume       AD
## 1 2020-01-02 28.6840 7.064225e-316     NA
## 2 2020-01-03 29.5340 1.317562e-315 0.8500
## 3 2020-01-06 30.1027 7.509551e-316 1.6194
## 4 2020-01-07 31.2707 1.325240e-315 2.7874
## 5 2020-01-08 32.8093 2.308099e-315 4.3814
## 6 2020-01-09 32.0893 2.107714e-315 3.2174
```

```
# 创建价格图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "特斯拉 (TSLA) 收盘价",
       x = "日期",
       y = "收盘价 ($)") +
  theme_minimal(base_family = "SimHei")
```

```
# 创建 Williams AD 指标图表
p2 <- ggplot(tsla_data, aes(x = Date, y = AD)) +
  geom_line(color = "red", size = 1) +
```

```
geom_hline(yintercept = 0, linetype = "dashed", color = "gray") +
  labs(title = "Williams 累积/派发线 (AD) 指标",
       x = "日期",
       y = "AD 值") +
  theme_minimal(base_family = "SimHei")

# 创建成交量图表
p3 <- ggplot(tsla_data, aes(x = Date, y = Volume)) +
  geom_col(fill = "darkgreen", alpha = 0.7) +
  labs(title = "特斯拉 (TSLA) 成交量",
       x = "日期",
       y = "成交量") +
  theme_minimal(base_family = "SimHei")

p1 / p2 /p3
```



```
# 确认趋势和背离
# 计算价格和 AD 指标的移动平均线 (用于趋势确认)
tsla_data$Close_MA20 <- SMA(tsla_data$Close, n = 20)
```

```
tsla_data$AD_MA20 <- SMA(tsla_data$AD, n = 20)

# 判断趋势方向
tsla_data$Price_Trend <- ifelse(tsla_data$Close > tsla_data$Close_MA20, "上升", "下降")
tsla_data$AD_Trend <- ifelse(tsla_data$AD > tsla_data$AD_MA20, "上升", "下降")

# 识别背离信号
tsla_data$Divergence <- ifelse(
  (tsla_data$Price_Trend == "上升" & tsla_data$AD_Trend == "下降"), "看跌背离",
  ifelse(
    (tsla_data$Price_Trend == "下降" & tsla_data$AD_Trend == "上升"), "看涨背离",
    )
)
table(tsla_data$Divergence)

## 
## 无背离 看涨背离 看跌背离
##      1196       57       67

# 可可视化背离信号
ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  geom_line(aes(y = Close_MA20), color = "purple", linetype = "dashed") +
  geom_point(data = subset(tsla_data, Divergence == "看涨背离"),
             aes(color = "看涨背离"), size = 3, shape = 16) +
  geom_point(data = subset(tsla_data, Divergence == "看跌背离"),
             aes(color = "看跌背离"), size = 3, shape = 17) +
  scale_color_manual(values = c("看涨背离" = "green", "看跌背离" = "red")) +
  labs(title = "特斯拉 (TSLA) 价格与 AD 指标背离信号",
       x = "日期",
       y = "收盘价 ($)",
       color = "背离类型") +
```



AI 引导词

1. 介绍 williamsAD 指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包的 R 语言实例的 Rmd 代码。

3.1.4.11 威廉%R 指标 (WPR) 指标含义

Williams %R 指标 (Williams Percent Range) 由 Larry Williams 开发，是一种衡量市场超买超卖状态的动量指标。与其他振荡指标（如 RSI、KDJ）类似，Williams %R 通过比较当前收盘价与一定周期内最高价和最低价的关系，判断市场短期是否处于极端状态。

该指标的核心特点包括：

1. 超买超卖判断：指标值在-20 至 0 之间表示超买，在-80 至-100 之间表示超卖

2. 动量反转信号：当指标从超买/超卖区域反转时，可能预示价格趋势即将改变
3. 背离分析：价格与 Williams %R 的背离可作为潜在趋势反转的预警信号

数学公式

Williams %R 的计算公式如下：

$$\%R = \frac{\text{Highest High} - \text{Close}}{\text{Highest High} - \text{Lowest Low}} \times (-100)$$

其中：

- *Highest High*: n 周期内的最高价
- *Lowest Low*: n 周期内的最低价
- *Close*: 当前收盘价
- *n*: 计算周期，通常为 14 天

与 KDJ 指标的关系：

Williams %R 与 KDJ 指标中的%K 线计算逻辑相似，但有两点主要区别：

1. Williams %R 的取值范围是 [-100, 0]，而 KDJ 指标的取值范围是 [0, 100]
2. Williams %R 的计算公式中没有平滑处理，而 KDJ 指标通常会对%K 线进行平滑得到%D 线

```
hlc <- TSLA[, c("High", "Low", "Close")]
stoch0sc <- stoch(hlc)

tsla_williams_r <- WPR(hlc)

# WPR is a transformation of stochastics' fastK
all.equal(na.omit(tsla_williams_r), na.omit(1-stoch0sc[, 'fastK'])) # TRUE

## [1] "Names: 1 string mismatch"
## [2] "Attributes: < Component \\"dimnames\\": Component 2: 1 string mismatch >"
```

```
# WPR converted to the usual scaling between 0 and -100
scaledWPR <- WPR(hlc, scale=TRUE)

tsla_data <- data.frame(
  Date = index(TSLA),
  Close = as.numeric(Cl(TSLA)),
  Williams_R = as.numeric(tsla_williams_r)
)

# 创建价格图表
p1 <- ggplot(tsla_data, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "特斯拉 (TSLA) 收盘价",
       x = "日期",
       y = "收盘价 ($)") +
  theme_minimal(base_family = "SimHei")

# 创建 Williams %R 指标图表
p2 <- ggplot(tsla_data, aes(x = Date, y = -Williams_R*100)) +
  geom_line(color = "red", size = 1) +
  geom_hline(yintercept = -20, linetype = "dashed", color = "darkred", alpha = 0.7) +
  geom_hline(yintercept = -80, linetype = "dashed", color = "darkgreen", alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "solid", color = "gray", alpha = 0.5) +
  geom_hline(yintercept = -100, linetype = "solid", color = "gray", alpha = 0.5) +
  labs(title = "Williams %R 指标 (默认 14 周期)",
       x = "日期",
       y = "Williams %R 值") +
  theme_minimal(base_family = "SimHei")

p1 / p2
```



AI 引导词

1. 介绍 williams %R 指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包的 R 语言实例的 Rmd 代码。

3.1.5 成交量分析

3.1.5.1 能量潮指标 (OBV) 指标含义

能量潮 (On-Balance Volume, OBV) 由 Joe Granville 在 1963 年提出，是一种通过成交量变化来预测价格趋势的技术指标。OBV 将成交量进行累积，当价格上涨时加入成交量，价格下跌时减去成交量。其核心思想是“成交量是价格的先行指标”，通过观察 OBV 的变化可以提前发现资金流向，识别潜在的价格突破或反转信号。

数学公式

OBV 的计算公式如下：

1. 如果今日收盘价高于昨日收盘价:

$$OBV_t = OBV_{t-1} + \text{Volume}_t$$

2. 如果今日收盘价低于昨日收盘价:

$$OBV_t = OBV_{t-1} - \text{Volume}_t$$

3. 如果今日收盘价等于昨日收盘价:

$$OBV_t = OBV_{t-1}$$

其中, OBV_t 是第 t 期的 OBV 值, Volume_t 是第 t 期的成交量。

代码实现及可视化

```
# 计算 TSLA 的 OBV
tsla_obv <- OBV(Cl(TSLA), Vo(TSLA))

# 计算 OBV 的移动平均 (用于信号生成)
tsla_obv_ma <- SMA(tsla_obv, n = 20)

# 准备数据框
tsla_df <- data.frame(
  Date = index(tsla_obv),
  Close = as.numeric(Cl(TSLA)),
  OBV = as.numeric(tsla_obv),
  OBV_MA = as.numeric(tsla_obv_ma)
) %>% na.omit()

# 创建价格图表
p1 <- ggplot(tsla_df, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "TSLA 收盘价", y = "价格") +
```

```
theme_minimal(base_family = "SimHei")

# 创建 OBV 指标图表
p2 <- ggplot(tsla_df, aes(x = Date)) +
  geom_line(aes(y = OBV), color = "purple", size = 0.8) +
  geom_line(aes(y = OBV_MA), color = "red", size = 0.8) +
  labs(title = "TSLA OBV 指标", y = "OBV 值") +
  theme_minimal(base_family = "SimHei")

# 组合图表
p1 / p2
```



```
# 识别 OBV 与价格的背离
# 计算价格和 OBV 的趋势（使用 SMA）
tsla_df$Price_MA = SMA(tsla_df$Close, n = 50)
tsla_df$OBV_MA = SMA(tsla_df$OBV, n = 50)

# 识别价格创新高但 OBV 未创新高的看跌背离
tsla_df$Bearish_Divergence <- tsla_df$Close > tsla_df$Price_MA &
```

```
tsla_df$OBV < tsla_df$OBV_MA &
lag(tsla_df$Close) > lag(tsla_df$Price_MA) &
lag(tsla_df$OBV) < lag(tsla_df$OBV_MA)

# 识别价格创新低但 OBV 未创新低的看涨背离
tsla_df$Bullish_Divergence <- tsla_df$Close < tsla_df$Price_MA &
tsla_df$OBV > tsla_df$OBV_MA &
lag(tsla_df$Close) < lag(tsla_df$Price_MA) &
lag(tsla_df$OBV) > lag(tsla_df$OBV_MA)

# 绘制带背离标记的价格图表
ggplot(tsla_df, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  geom_line(aes(y = Price_MA), color = "gray", size = 0.8, linetype = "dashed") +
  geom_point(data = subset(tsla_df, Bearish_Divergence == TRUE),
             aes(x = Date, y = Close), color = "red", size = 3) +
  geom_point(data = subset(tsla_df, Bullish_Divergence == TRUE),
             aes(x = Date, y = Close), color = "green", size = 3) +
  labs(title = "TSLA 价格与 OBV 背离分析",
       y = "价格") +
  theme_minimal(base_family = "SimHei") +
  annotate("text", x = subset(tsla_df, Bearish_Divergence == TRUE)$Date,
           y = subset(tsla_df, Bearish_Divergence == TRUE)$Close + 50,
           label = "看跌背离", color = "red", family = "SimHei") +
  annotate("text", x = subset(tsla_df, Bullish_Divergence == TRUE)$Date,
           y = subset(tsla_df, Bullish_Divergence == TRUE)$Close - 50,
           label = "看涨背离", color = "green", family = "SimHei")
```



AI 引导词

1. 介绍 OBV 指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.5.2 收盘价定位指标 (CLV) 指标含义

收盘价定位指标 Close Location Value (CLV) 是一种衡量收盘价在当日价格区间中相对位置的技术指标，其值在 -1 到 +1 之间。CLV 由 Marc Chaikin 提出，作为计算 Chaikin 累积 / 派发线 (Chaikin Accumulation/Distribution Line) 的基础组件。该指标的核心价值在于量化收盘价在当日价格波动中的位置，从而反映市场买卖力量的强弱对比：

- CLV = +1：收盘价等于最高价，表明多方完全主导市场，买盘强劲；
- CLV = -1：收盘价等于最低价，表明空方完全主导市场，卖盘强劲；
- CLV = 0：收盘价位于最高价和最低价的中点，表明买卖力量均衡；
- CLV > 0：收盘价接近当日最高价，多方力量占优；
- CLV < 0：收盘价接近当日最低价，空方力量占优。

CLV 常用于构建更复杂的资金流向指标，如 Chaikin AD 线，通过结合价

格区间位置和成交量来评估资金流入和流出的强度。

数学公式

Close Location Value (CLV) 的计算公式如下：

$$\text{CLV}_t = \frac{(\text{Close}_t - \text{Low}_t) - (\text{High}_t - \text{Close}_t)}{\text{High}_t - \text{Low}_t}$$

展开后可简化为：

$$\text{CLV}_t = \frac{2 \times \text{Close}_t - \text{High}_t - \text{Low}_t}{\text{High}_t - \text{Low}_t}$$

其中：

- High_t 是第 t 期的最高价；
- Low_t 是第 t 期的最低价；
- Close_t 是第 t 期的收盘价。

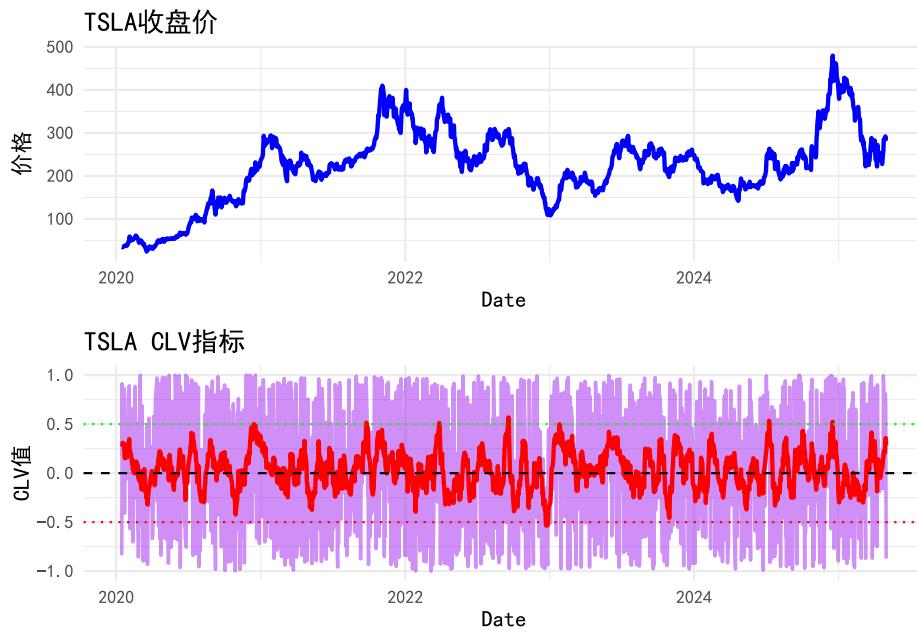
代码实现及可视化

```
# 计算 TSLA 的 CLV
tsla_clv <- CLV(TSLA[, c("High", "Low", "Close")])
# 计算 CLV 的移动平均（用于平滑）
tsla_clv_ma <- SMA(tsla_clv, n = 10)

# 准备数据框
tsla_df <- data.frame(
  Date = index(tsla_clv),
  Close = as.numeric(C1(TSLA)),
  CLV = as.numeric(tsla_clv),
  CLV_MA = as.numeric(tsla_clv_ma)
) %>% na.omit()

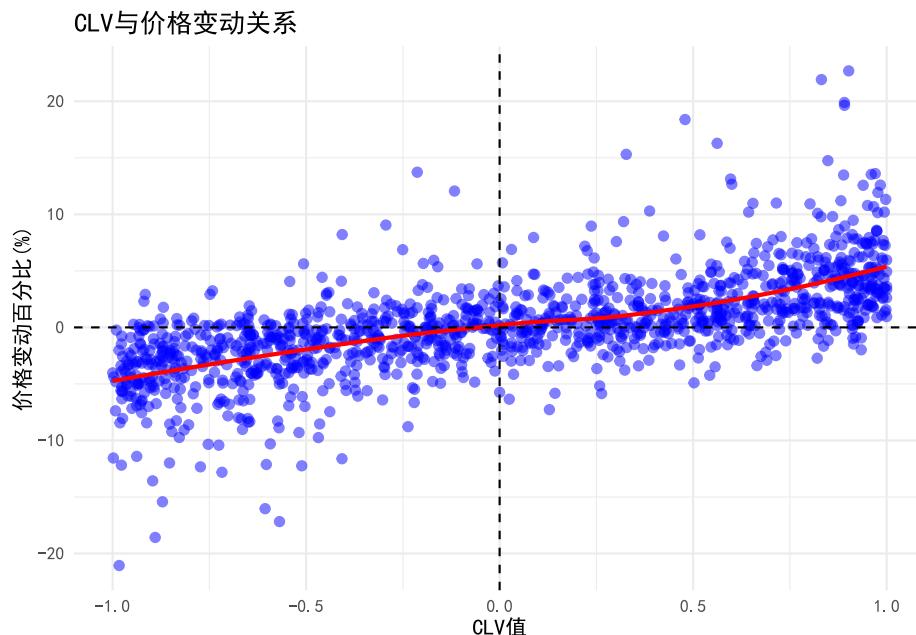
# 创建价格图表
p1 <- ggplot(tsla_df, aes(x = Date, y = Close)) +
```

```
geom_line(color = "blue", size = 1) +  
  labs(title = "TSLA 收盘价", y = "价格") +  
  theme_minimal(base_family = "SimHei")  
  
# 创建 CLV 指标图表  
p2 <- ggplot(tsla_df, aes(x = Date)) +  
  geom_line(aes(y = CLV), color = "purple", size = 0.8, alpha = 0.5) +  
  geom_line(aes(y = CLV_MA), color = "red", size = 1) +  
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +  
  geom_hline(yintercept = 0.5, linetype = "dotted", color = "green") +  
  geom_hline(yintercept = -0.5, linetype = "dotted", color = "red") +  
  labs(title = "TSLA CLV 指标", y = "CLV 值") +  
  theme_minimal(base_family = "SimHei")  
  
# 组合图表  
p1 / p2
```



```
# 计算价格变动百分比
tsla_df$PriceChange <- (tsla_df$Close - lag(tsla_df$Close)) / lag(tsla_df$Close) * 100

# 绘制 CLV 与价格变动散点图
ggplot(tsla_df, aes(x = CLV, y = PriceChange)) +
  geom_point(color = "blue", alpha = 0.5, size = 2) +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  geom_vline(xintercept = 0, linetype = "dashed", color = "black") +
  labs(title = "CLV 与价格变动关系",
       x = "CLV 值",
       y = " 价格变动百分比 (%)") +
  theme_minimal(base_family = "SimHei")
```



```
# 找出 CLV 极端值的日期（接近 +1 或 -1）
extreme_clv_dates <- tsla_df %>%
  filter(CLV > 0.9 | CLV < -0.9) %>%
  arrange(desc(abs(CLV)))
```

```
# 显示极端值日期和 CLV 值
knitr::kable(extreme_clv_dates[, c("Date", "Close", "CLV")],
             caption = "TSLA 的极端 CLV 值日期")
```

表 5: TSLA 的极端 CLV 值日期

Date	Close	CLV
2020-05-29	55.6667	1.0000000
2020-06-25	65.7320	1.0000000
2020-11-25	191.3333	1.0000000
2020-12-04	199.6800	1.0000000
2020-12-18	231.6667	1.0000000
2024-09-23	250.0000	1.0000000
2022-03-07	268.1933	-0.9996784
2020-05-14	53.5553	0.9984756
2022-01-27	276.3667	-0.9981164
2020-03-02	49.5747	0.9975798
2020-10-28	135.3400	-0.9968095
2020-06-08	63.3280	0.9961067
2023-05-31	203.9300	0.9954700
2025-04-24	259.5100	0.9941973
2024-07-02	231.2600	0.9939577
2024-12-13	436.2300	0.9932006
2025-03-25	288.1400	0.9929078
2020-10-02	138.3633	-0.9925405
2021-03-24	210.0900	-0.9915642
2022-11-30	194.7000	0.9915074
2022-05-27	253.2100	0.9913369
2022-07-05	233.0667	0.9905771
2024-06-24	182.5800	-0.9904000
2024-12-11	424.7700	0.9902222
2021-08-30	243.6367	0.9901478
2024-12-16	463.0200	0.9874260

Date	Close	CLV
2024-09-25	257.0200	0.9874214
2020-09-14	139.8733	0.9837217
2020-11-09	140.4200	-0.9834857
2020-09-08	110.0700	-0.9830159
2022-08-12	300.0300	0.9828118
2021-10-15	281.0100	0.9827420
2024-09-06	210.7300	-0.9809441
2020-06-29	67.2900	0.9788708
2024-04-15	161.4800	-0.9785177
2022-01-19	331.8833	-0.9782202
2023-12-20	247.1400	-0.9781931
2020-12-10	209.0233	0.9778505
2025-03-24	278.3900	0.9775885
2023-09-20	262.5900	-0.9773322
2022-04-26	292.1400	-0.9772816
2020-04-27	53.2500	0.9770660
2022-04-29	290.2533	-0.9764007
2023-02-17	208.3100	0.9762340
2023-02-21	197.3700	-0.9759808
2023-01-30	166.6600	-0.9758855
2022-09-07	283.7000	0.9757995
2022-11-02	214.9800	-0.9754789
2022-01-10	352.7067	0.9752262
2023-09-25	246.9900	0.9749716
2023-02-16	202.0400	-0.9746996
2020-05-04	50.7460	0.9746877
2021-11-01	402.8633	0.9745285
2022-09-21	300.8000	-0.9741838
2022-12-20	137.8000	-0.9740981
2021-04-13	254.1067	0.9740236
2021-09-24	258.1300	0.9728770
2024-10-07	240.8300	-0.9715225

Date	Close	CLV
2024-12-06	389.2200	0.9711075
2023-01-20	133.4200	0.9707792
2020-04-13	43.3967	0.9706172
2023-08-29	257.1800	0.9695586
2021-04-21	248.0400	0.9692570
2024-11-25	338.5900	-0.9671302
2022-02-24	266.9233	0.9666276
2022-01-31	312.2400	0.9665551
2021-03-01	239.4767	0.9664213
2021-10-22	303.2267	0.9664072
2022-12-16	150.2300	-0.9652968
2020-06-11	64.8560	-0.9642253
2024-12-04	357.9300	0.9642105
2025-02-10	350.7300	-0.9639048
2024-12-24	462.2800	0.9638205
2022-06-28	232.6633	-0.9636914
2021-12-21	312.8433	0.9636494
2021-05-27	210.2833	0.9624402
2023-06-27	250.2100	0.9622642
2022-02-15	307.4767	0.9615128
2022-09-09	299.6800	0.9604651
2023-06-26	241.0500	-0.9603848
2022-01-03	399.9267	0.9603261
2020-06-01	59.8733	0.9599105
2025-01-13	403.3100	0.9595278
2022-10-21	214.4400	0.9594843
2023-04-12	180.5400	-0.9591837
2023-03-22	191.1500	-0.9588054
2021-07-14	217.7933	-0.9580908
2024-10-25	269.1900	0.9576570
2023-11-28	246.7200	0.9568899
2023-06-20	274.4500	0.9559795

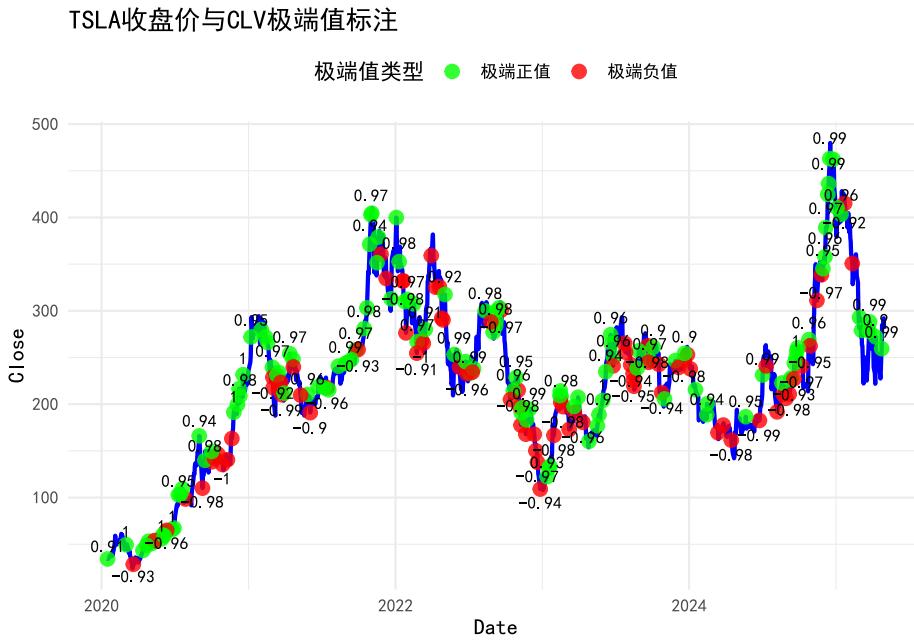
Date	Close	CLV
2021-01-07	272.0133	0.9545298
2020-07-28	98.4327	-0.9541422
2024-05-21	186.6000	0.9539852
2021-06-23	218.8567	0.9536086
2023-06-22	264.6100	0.9534328
2024-08-07	191.7600	-0.9533722
2022-06-09	239.7067	-0.9531443
2024-08-22	210.6600	-0.9530387
2021-08-12	240.7500	0.9530000
2024-10-28	262.5100	-0.9522124
2021-12-09	334.6000	-0.9521037
2021-11-03	404.6200	0.9512505
2022-09-08	289.2600	0.9507187
2020-07-10	102.9767	0.9506216
2022-11-09	177.5900	-0.9499201
2022-11-10	190.7200	0.9489517
2023-08-16	225.6000	-0.9487776
2024-09-19	243.9200	0.9471511
2024-11-29	345.1600	0.9462963
2023-12-06	239.3700	-0.9459459
2022-10-28	228.5200	0.9456435
2023-10-25	212.4200	-0.9443038
2023-02-14	209.2500	0.9440628
2021-10-29	371.3333	0.9423714
2024-09-11	228.1300	0.9417309
2024-08-19	222.7200	0.9415073
2024-09-10	226.1700	0.9407216
2023-07-27	255.7100	-0.9407086
2023-09-22	244.8800	-0.9398948
2023-06-08	234.8600	0.9394435
2020-08-31	166.1067	0.9393703
2021-03-03	217.7333	-0.9391794

Date	Close	CLV
2022-12-12	167.8200	-0.9390863
2021-03-18	217.7200	-0.9376793
2023-03-31	207.4600	0.9376771
2022-12-27	109.1000	-0.9376719
2023-01-17	131.4900	0.9371257
2023-08-09	242.1900	-0.9369565
2022-03-11	265.1167	-0.9368340
2023-03-09	172.9200	-0.9352802
2024-02-14	188.7100	0.9350181
2023-01-13	122.4000	0.9345661
2024-01-17	215.5500	0.9344262
2021-11-26	360.6400	-0.9337581
2022-04-20	325.7333	-0.9336172
2021-09-30	258.4933	-0.9320594
2020-03-20	28.5020	-0.9320445
2022-11-07	197.0800	-0.9313725
2024-08-29	206.2800	-0.9304933
2023-07-25	265.2800	-0.9291139
2022-11-21	167.8700	-0.9284940
2023-04-27	160.1900	0.9284834
2024-02-15	200.4500	0.9284526
2021-11-19	379.0200	0.9278618
2022-06-24	245.7067	0.9278557
2022-03-31	359.2000	-0.9275469
2024-09-18	227.2000	-0.9272727
2021-03-17	233.9367	0.9271626
2024-09-27	260.4600	0.9270517
2022-07-14	238.3133	0.9270386
2022-09-01	277.1600	0.9265092
2022-07-11	234.3433	-0.9262628
2023-05-18	176.8900	0.9262473
2022-10-14	204.9900	-0.9248869

Date	Close	CLV
2022-11-23	183.2000	0.9244604
2021-02-12	272.0400	0.9243808
2021-07-19	215.4067	0.9243684
2023-12-28	253.1800	-0.9243156
2025-03-05	279.1000	0.9239865
2022-04-11	325.3100	-0.9237366
2021-05-10	209.6800	-0.9236058
2025-01-22	415.1100	-0.9224459
2022-11-11	195.9700	0.9210337
2021-03-02	228.8133	-0.9202439
2020-05-12	53.9607	-0.9200884
2021-03-12	231.2433	0.9199582
2024-01-04	237.9300	-0.9195171
2021-03-22	223.3333	-0.9190282
2023-11-01	205.6600	0.9189189
2020-11-20	163.2033	-0.9181696
2022-05-04	317.5400	0.9179722
2024-11-14	311.1800	-0.9173891
2020-04-22	48.8073	0.9165369
2023-03-21	197.5800	0.9156627
2024-03-26	177.6700	-0.9155750
2020-06-05	59.0440	0.9153994
2022-08-26	288.0900	-0.9146593
2020-12-08	216.6267	0.9145937
2022-01-24	310.0000	0.9144321
2020-07-15	103.0673	0.9141935
2020-07-20	109.5333	0.9135741
2020-01-16	34.2327	0.9130552
2021-03-30	211.8733	0.9125402
2021-09-03	244.5233	0.9122050
2024-07-11	241.0300	-0.9119617
2023-08-15	232.9600	-0.9112801

Date	Close	CLV
2025-02-28	292.9800	0.9112426
2022-09-16	303.3500	0.9112207
2025-01-03	410.4400	0.9111933
2022-03-16	280.0767	0.9109212
2021-02-17	266.0500	0.9106582
2021-11-16	351.5767	0.9102181
2020-10-08	141.9733	-0.9095169
2021-04-22	239.8967	-0.9076406
2022-02-23	254.6800	-0.9068770
2021-02-01	279.9367	0.9056848
2023-08-25	238.5900	0.9052023
2022-01-20	332.0900	-0.9047379
2021-07-08	217.6033	0.9046214
2023-10-18	242.6800	-0.9043825
2024-03-13	169.4800	-0.9043478
2020-10-16	146.5567	-0.9040702
2023-12-21	254.5000	0.9040000
2021-06-03	190.9467	-0.9027903
2023-05-22	188.8700	0.9022801
2025-04-09	272.2000	0.9019878
2023-10-04	261.1600	0.9018233
2023-02-15	214.2400	0.9017544
2023-08-17	219.2200	-0.9013906
2021-09-13	247.6667	0.9009243
2020-10-01	149.3867	0.9004149

```
data = extreme_clv_dates,
aes(color = ifelse(CLV > 0, " 极端正值", " 极端负值")),
size = 3, alpha = 0.8
) +
geom_text(
  data = extreme_clv_dates,
  aes(
    label = round(CLV, 2),
    y = Close * ifelse(CLV > 0, 1.02, 0.98) # 垂直偏移
  ),
  vjust = ifelse(extreme_clv_dates$CLV > 0, -0.5, 1.5), # 方向调整
  size = 3,
  family = "SimHei",
  check_overlap = TRUE # 自动过滤重叠标签
) +
scale_color_manual(
  name = " 极端值类型",
  values = c(" 极端正值" = "green", " 极端负值" = "red"))
) +
labs(title = "TSLA 收盘价与 CLV 极端值标注") +
theme_minimal(base_family = "SimHei") +
theme(legend.position = "top")
```



1. 计算累积/派发线 (Accumulation/Distribution Line):

$$AD_t = AD_{t-1} + CMF_t \times Volume_t$$

其中, CMF_t 是资金流向乘数 (Money Flow Multiplier):

$$CMF_t = \frac{(Close_t - Low_t) - (High_t - Close_t)}{High_t - Low_t}$$

展开后:

$$CMF_t = \frac{2 \times Close_t - High_t - Low_t}{High_t - Low_t}$$

2. 简化后的计算公式:

$$AD_t = AD_{t-1} + \left(\frac{Close_t - Low_t - (High_t - Close_t)}{High_t - Low_t} \right) \times Volume_t$$

当收盘价接近最高价时, CMF 为正, AD 线上升; 当收盘价接近最低价时, CMF 为负, AD 线下降。

代码实现及可视化

```
# 计算 TSLA 的 Chaikin 累积/派发指标
tsla_ad <- chaikinAD(TSLA[, c("High", "Low", "Close")], TSLA[, "Volume"])

# 计算 AD 线的移动平均 (用于信号生成)
tsla_ad_ma <- SMA(tsla_ad, n = 20)

# 准备数据框
tsla_df <- data.frame(
  Date = index(tsla_ad),
  Close = as.numeric(Cl(TSLA)),
  AD = as.numeric(tsla_ad),
  AD_MA = as.numeric(tsla_ad_ma)
```

```
) %>% na.omit()

# 创建价格图表
p1 <- ggplot(tsla_df, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "TSLA 收盘价", y = "价格") +
  theme_minimal(base_family = "SimHei")

# 创建 Chaikin 累积/派发指标图表
p2 <- ggplot(tsla_df, aes(x = Date)) +
  geom_line(aes(y = AD), color = "purple", size = 0.8) +
  geom_line(aes(y = AD_MA), color = "red", size = 0.8) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  labs(title = "TSLA Chaikin 累积/派发指标", y = "AD 值") +
  theme_minimal(base_family = "SimHei")

# 识别 AD 与价格的背离
# 计算价格和 AD 的趋势（使用 SMA）
tsla_df$Price_MA = SMA(tsla_df$Close, n = 50)
tsla_df$AD_MA = SMA(tsla_df$AD, n = 50)

# 识别价格创新高但 AD 未创新高的看跌背离
tsla_df$Bearish_Divergence <- tsla_df$Close > tsla_df$Price_MA &
  tsla_df$AD < tsla_df$AD_MA &
  lag(tsla_df$Close) > lag(tsla_df$Price_MA) &
  lag(tsla_df$AD) < lag(tsla_df$AD_MA)

# 识别价格创新低但 AD 未创新低的看涨背离
tsla_df$Bullish_Divergence <- tsla_df$Close < tsla_df$Price_MA &
  tsla_df$AD > tsla_df$AD_MA &
  lag(tsla_df$Close) < lag(tsla_df$Price_MA) &
  lag(tsla_df$AD) > lag(tsla_df$AD_MA)
```

```
# 绘制带背离标记的价格图表
p3 <- ggplot(tsla_df, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  geom_line(aes(y = Price_MA), color = "gray", size = 0.8, linetype = "dashed") +
  geom_point(data = subset(tsla_df, Bearish_Divergence == TRUE),
             aes(x = Date, y = Close), color = "green", size = 1) +
  geom_point(data = subset(tsla_df, Bullish_Divergence == TRUE),
             aes(x = Date, y = Close), color = "red", size = 1) +
  labs(title = "TSLA 价格与 AD 指标背离分析",
       y = " 价格") +
  theme_minimal(base_family = "SimHei") +
  annotate("text", x = subset(tsla_df, Bearish_Divergence == TRUE)$Date,
           y = subset(tsla_df, Bearish_Divergence == TRUE)$Close + 50,
           label = " 看跌背离", color = "green",family="SimHei",size=2) +
  annotate("text", x = subset(tsla_df, Bullish_Divergence == TRUE)$Date,
           y = subset(tsla_df, Bullish_Divergence == TRUE)$Close - 50,
           label = " 看涨背离", color = "red",family="SimHei",size=2)

# 组合图表
p1 / p2 / p3
```



1. 介绍 Chaikin Accumulation / Distribution 指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包) 的 R 语言实例的 Rmd 代码。

3.1.5.4 蔡金资金流量 (CMF) 指标含义

Chaikin 资金流向 (Chaikin Money Flow, CMF) 由 Marc Chaikin 提出，是一种结合价格和成交量来衡量资金流入和流出强度的技术指标。CMF 通过计算一段时间内的累积资金流向百分比，评估多方和空方的力量对比。当 CMF 为正时，表示资金流入，多方力量较强；当 CMF 为负时，表示资金流出，空方力量较强。CMF 值接近 0 表示买卖双方力量均衡。

数学公式

Chaikin 资金流向 (CMF) 的计算公式如下：

1. 计算资金流向乘数 (Money Flow Multiplier)：

$$MFM_t = \frac{(Close_t - Low_t) - (High_t - Close_t)}{High_t - Low_t}$$

2. 计算资金流量 (Money Flow Volume):

$$\text{MFV}_t = \text{MFM}_t \times \text{Volume}_t$$

3. 计算 n 周期的 CMF:

$$\text{CMF}_n = \frac{\sum_{i=0}^{n-1} \text{MFV}_{t-i}}{\sum_{i=0}^{n-1} \text{Volume}_{t-i}}$$

其中, Close_t 、 High_t 、 Low_t 分别是第 t 期的收盘价、最高价和最低价, Volume_t 是第 t 期的成交量。

代码实现及可视化

```
tsla_cmf <- CMF(TSLA[, c("High", "Low", "Close")], TSLA[, "Volume"])

## [1]
## 2025-04-24 0.05110992
## 2025-04-25 0.15203000
## 2025-04-28 0.18788751
## 2025-04-29 0.18122168
## 2025-04-30 0.21222269
## 2025-05-01 0.12050120

# 准备数据框
tsla_df <- data.frame(
  Date = index(tsla_cmf),
  Close = as.numeric(Cl(TSLA)),
  CMF = as.numeric(tsla_cmf)
) %>% na.omit()

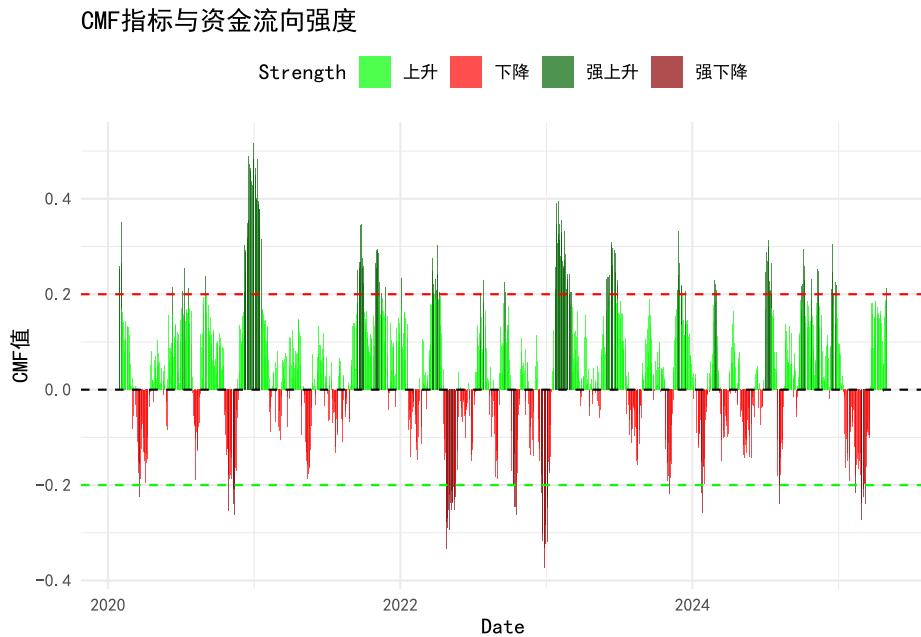
# 创建价格图表
p1 <- ggplot(tsla_df, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
```

```
labs(title = "TSLA 收盘价", y = "价格") +  
  theme_minimal(base_family = "SimHei")  
  
# 创建 CMF 指标图表  
p2 <- ggplot(tsla_df, aes(x = Date, y = CMF)) +  
  geom_line(color = "purple", size = 0.8) +  
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +  
  geom_hline(yintercept = 0.2, linetype = "dashed", color = "red") +  
  geom_hline(yintercept = -0.2, linetype = "dashed", color = "green") +  
  labs(title = "TSLA CMF 指标", y = "CMF 值") +  
  theme_minimal(base_family = "SimHei")  
  
# 组合图表  
p1 / p2
```



```
# 添加趋势判断  
tsla_df$Trend <- ifelse(tsla_df$CMF > 0, "上升", "下降")  
tsla_df$Strength <- ifelse(tsla_df$CMF > 0.2, "强上升",  
                           ifelse(tsla_df$CMF < -0.2, "强下降", tsla_df$Trend))
```

```
# 绘制 CMF 与趋势关系
ggplot(tsla_df, aes(x = Date, y = CMF, fill = Strength)) +
  geom_bar(stat = "identity", alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  geom_hline(yintercept = 0.2, linetype = "dashed", color = "red") +
  geom_hline(yintercept = -0.2, linetype = "dashed", color = "green") +
  scale_fill_manual(values = c("强上升" = "darkgreen", "上升" = "green",
                               "下降" = "red", "强下降" = "darkred")) +
  labs(title = "CMF 指标与资金流向强度",
       y = "CMF 值") +
  theme_minimal(base_family = "SimHei") +
  theme(legend.position = "top")
```



1. 介绍 CMF 指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.1.5.5 资金流量指标（MFI） 指标含义

资金流量指标 (Money Flow Index, MFI) 是一种结合价格和成交量来衡量资金流入和流出的技术指标，由 Gene Quong 和 Avrum Soudack 在 1991 年提出。MFI 类似于 RSI，但考虑了成交量因素，因此被称为“成交量加权 RSI”。MFI 通过计算一段时间内的正资金流量和负资金流量之比，判断市场的超买超卖状态。当 MFI 高于 80 时，表示市场处于超买状态；当 MFI 低于 20 时，表示市场处于超卖状态。

数学公式

资金流量指标 (MFI) 的计算公式如下：

1. 计算典型价格 (Typical Price):

$$TP_t = \frac{\text{High}_t + \text{Low}_t + \text{Close}_t}{3}$$

2. 计算资金流量 (Money Flow):

$$MF_t = TP_t \times \text{Volume}_t$$

3. 区分正资金流量和负资金流量：

- 如果 $TP_t > TP_{t-1}$, 则 Positive $MF_t = MF_t$, 否则为 0;
- 如果 $TP_t < TP_{t-1}$, 则 Negative $MF_t = MF_t$, 否则为 0.

4. 计算 n 周期的资金流量比率 (Money Flow Ratio):

$$MFR_n = \frac{\sum_{i=0}^{n-1} \text{Positive } MF_{t-i}}{\sum_{i=0}^{n-1} \text{Negative } MF_{t-i}}$$

5. 计算 MFI:

$$MFI_n = 100 - \frac{100}{1 + MFR_n}$$

其中, High_t 、 Low_t 、 Close_t 分别是第 t 期的最高价、最低价和收盘价, Volume_t 是第 t 期的成交量。

代码实现及可视化

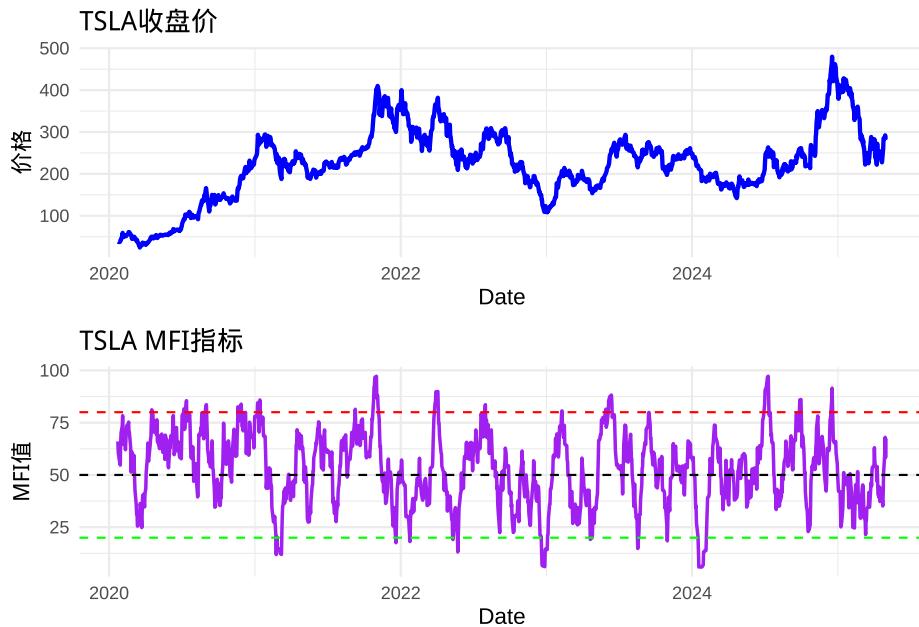
```
tsla_mfi <- MFI(TSLA[, c("High", "Low", "Close")], TSLA[,"Volume"])

# 准备数据框
tsla_df <- data.frame(
  Date = index(tsla_mfi),
  Close = as.numeric(Cl(TSLA)),
  MFI = as.numeric(tsla_mfi)
) %>% na.omit()

# 创建价格图表
p1 <- ggplot(tsla_df, aes(x = Date, y = Close)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "TSLA 收盘价", y = "价格") +
  theme_minimal()

# 创建 MFI 指标图表
p2 <- ggplot(tsla_df, aes(x = Date, y = MFI)) +
  geom_line(color = "purple", size = 0.8) +
  geom_hline(yintercept = 80, linetype = "dashed", color = "red") +
  geom_hline(yintercept = 20, linetype = "dashed", color = "green") +
  geom_hline(yintercept = 50, linetype = "dashed", color = "black") +
  labs(title = "TSLA MFI 指标", y = "MFI 值") +
  theme_minimal()

# 组合图表
p1 / p2
```



1. 介绍 MFI 指标的意义；2. 对应 Latex 公式的 Rmd 代码；3. 基于数据 TSLA 及 TTR 包）的 R 语言实例的 Rmd 代码。

3.2 数据处理类函数

3.2.1 其它实用函数

```
splits <- getSplits("TSLA")
dividends <- getDividends("TSLA")
adjRatios(splits, dividends, C1(TSLA))
```

3.2.1.1 adjRatios 函数

3.2.1.2 ROC 函数 函数的作用是计算一个价格序列或者交易量序列在 n 个周期内的（变化）率。运行下面代码：

```
tsla_roc_price <- ROC(Cl(TSLA), n = 1, type = c("continuous", "discrete"), na.pad = TRUE)

tsla_roc_volume <- ROC(Vo(TSLA), n = 1, type = c("continuous", "discrete"), na.pad = TRUE)
```

3.2.1.3 momentum 函数 函数的作用是计算一个价格序列或者交易量序列在 n 个周期内的（变化）率。运行下面代码

```
tsla_mom <- momentum(TSLA, n = 1, na.pad = TRUE)
tail(tsla_mom)
```

	Open	High	Low	Close	Volume	Adjusted
## 2025-04-24	-4.36	0.09	4.77	8.77	-2.762701e-316	8.77
## 2025-04-25	11.19	27.31	10.43	25.44	3.611447e-316	25.44
## 2025-04-28	27.29	8.01	12.79	0.93	-7.820516e-317	0.93
## 2025-04-29	-3.48	-1.54	7.05	6.15	-2.115846e-316	6.15
## 2025-04-30	-5.60	-8.87	-8.69	-9.87	9.908239e-317	-9.87
## 2025-05-01	0.11	6.42	9.03	-1.64	-1.447716e-316	-1.64

3.2.1.4 lags 函数 函数的作用是生成参数序列的滞后序列。代码如下：

```
tsla_lags <- lags(TSLA, n = 1)
```

```
# 计算双均线（快线 5 日，慢线 20 日）
price <- Ad(TSLA) # 使用调整后价格
volume <- Vo(TSLA)

# 计算双均线（带成交量过滤）
fast_sma <- SMA(price, 5)
slow_sma <- SMA(price, 20)
vol_sma <- SMA(volume, 10) # 10 日平均成交量阈值

# 生成增强型三态信号
```

```
signals <- ifelse(
  fast_sma > slow_sma & volume > vol_sma, 1, # 放量金叉做多
  ifelse(
    fast_sma < slow_sma & volume > vol_sma, -1, # 放量死叉做空
    0 # 无量或均线粘合时空仓
  )
)
signals <- lag(signals, 1) # 避免未来偏差
signals[is.na(signals)] <- 0 # 处理初始 NA

# 计算策略收益 (修正 growth 函数实现)
returns <- ROC(price) * signals # 使用价格变动率计算
returns[is.na(returns)] <- 0 # 处理初始 NA
cum_returns <- cumprod(1 + returns) - 1 # 累计收益

# 性能分析
print(table(signals)) # 查看信号分布
```

3.2.1.5 growth 函数

```
## signals
## -1 0 1
## 248 779 313

charts.PerformanceSummary(merge.xts(
  ROC(price),
  cum_returns
), main="TSLA 增强型双均线策略")
```

TSLA增强型双均线策略



```
# 打印关键指标
cat(" 年化收益率:", round(Return.annualized(returns)*100,2), "%\n")

## 年化收益率: -6.79 %

cat(" 最大回撤:", round(maxDrawdown(returns)*100,2), "%")
## 最大回撤: 55.83 %
```

3.2.1.6 naCheck 函数 该函数的作用是识别序列中是否存在 NA 值。

```
tsla_nc <- naCheck(TSLA, n = 0)
tsla_nc

## $NAs
## [1] 0
##
## $nonNA
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14
## [15] 15 16 17 18 19 20 21 22 23 24 25 26 27 28
## [29] 29 30 31 32 33 34 35 36 37 38 39 40 41 42
```

```
## [43] 43 44 45 46 47 48 49 50 51 52 53 54 55 56
## [57] 57 58 59 60 61 62 63 64 65 66 67 68 69 70
## [71] 71 72 73 74 75 76 77 78 79 80 81 82 83 84
## [85] 85 86 87 88 89 90 91 92 93 94 95 96 97 98
## [99] 99 100 101 102 103 104 105 106 107 108 109 110 111 112
## [113] 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## [127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140
## [141] 141 142 143 144 145 146 147 148 149 150 151 152 153 154
## [155] 155 156 157 158 159 160 161 162 163 164 165 166 167 168
## [169] 169 170 171 172 173 174 175 176 177 178 179 180 181 182
## [183] 183 184 185 186 187 188 189 190 191 192 193 194 195 196
## [197] 197 198 199 200 201 202 203 204 205 206 207 208 209 210
## [211] 211 212 213 214 215 216 217 218 219 220 221 222 223 224
## [225] 225 226 227 228 229 230 231 232 233 234 235 236 237 238
## [239] 239 240 241 242 243 244 245 246 247 248 249 250 251 252
## [253] 253 254 255 256 257 258 259 260 261 262 263 264 265 266
## [267] 267 268 269 270 271 272 273 274 275 276 277 278 279 280
## [281] 281 282 283 284 285 286 287 288 289 290 291 292 293 294
## [295] 295 296 297 298 299 300 301 302 303 304 305 306 307 308
## [309] 309 310 311 312 313 314 315 316 317 318 319 320 321 322
## [323] 323 324 325 326 327 328 329 330 331 332 333 334 335 336
## [337] 337 338 339 340 341 342 343 344 345 346 347 348 349 350
## [351] 351 352 353 354 355 356 357 358 359 360 361 362 363 364
## [365] 365 366 367 368 369 370 371 372 373 374 375 376 377 378
## [379] 379 380 381 382 383 384 385 386 387 388 389 390 391 392
## [393] 393 394 395 396 397 398 399 400 401 402 403 404 405 406
## [407] 407 408 409 410 411 412 413 414 415 416 417 418 419 420
## [421] 421 422 423 424 425 426 427 428 429 430 431 432 433 434
## [435] 435 436 437 438 439 440 441 442 443 444 445 446 447 448
## [449] 449 450 451 452 453 454 455 456 457 458 459 460 461 462
## [463] 463 464 465 466 467 468 469 470 471 472 473 474 475 476
## [477] 477 478 479 480 481 482 483 484 485 486 487 488 489 490
## [491] 491 492 493 494 495 496 497 498 499 500 501 502 503 504
```

```
## [505] 505 506 507 508 509 510 511 512 513 514 515 516 517 518
## [519] 519 520 521 522 523 524 525 526 527 528 529 530 531 532
## [533] 533 534 535 536 537 538 539 540 541 542 543 544 545 546
## [547] 547 548 549 550 551 552 553 554 555 556 557 558 559 560
## [561] 561 562 563 564 565 566 567 568 569 570 571 572 573 574
## [575] 575 576 577 578 579 580 581 582 583 584 585 586 587 588
## [589] 589 590 591 592 593 594 595 596 597 598 599 600 601 602
## [603] 603 604 605 606 607 608 609 610 611 612 613 614 615 616
## [617] 617 618 619 620 621 622 623 624 625 626 627 628 629 630
## [631] 631 632 633 634 635 636 637 638 639 640 641 642 643 644
## [645] 645 646 647 648 649 650 651 652 653 654 655 656 657 658
## [659] 659 660 661 662 663 664 665 666 667 668 669 670 671 672
## [673] 673 674 675 676 677 678 679 680 681 682 683 684 685 686
## [687] 687 688 689 690 691 692 693 694 695 696 697 698 699 700
## [701] 701 702 703 704 705 706 707 708 709 710 711 712 713 714
## [715] 715 716 717 718 719 720 721 722 723 724 725 726 727 728
## [729] 729 730 731 732 733 734 735 736 737 738 739 740 741 742
## [743] 743 744 745 746 747 748 749 750 751 752 753 754 755 756
## [757] 757 758 759 760 761 762 763 764 765 766 767 768 769 770
## [771] 771 772 773 774 775 776 777 778 779 780 781 782 783 784
## [785] 785 786 787 788 789 790 791 792 793 794 795 796 797 798
## [799] 799 800 801 802 803 804 805 806 807 808 809 810 811 812
## [813] 813 814 815 816 817 818 819 820 821 822 823 824 825 826
## [827] 827 828 829 830 831 832 833 834 835 836 837 838 839 840
## [841] 841 842 843 844 845 846 847 848 849 850 851 852 853 854
## [855] 855 856 857 858 859 860 861 862 863 864 865 866 867 868
## [869] 869 870 871 872 873 874 875 876 877 878 879 880 881 882
## [883] 883 884 885 886 887 888 889 890 891 892 893 894 895 896
## [897] 897 898 899 900 901 902 903 904 905 906 907 908 909 910
## [911] 911 912 913 914 915 916 917 918 919 920 921 922 923 924
## [925] 925 926 927 928 929 930 931 932 933 934 935 936 937 938
## [939] 939 940 941 942 943 944 945 946 947 948 949 950 951 952
## [953] 953 954 955 956 957 958 959 960 961 962 963 964 965 966
```

```

## [967] 967 968 969 970 971 972 973 974 975 976 977 978 979 980
## [981] 981 982 983 984 985 986 987 988 989 990 991 992 993 994
## [995] 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008
## [1009] 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022
## [1023] 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036
## [1037] 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050
## [1051] 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064
## [1065] 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078
## [1079] 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092
## [1093] 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106
## [1107] 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120
## [1121] 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134
## [1135] 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148
## [1149] 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162
## [1163] 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176
## [1177] 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190
## [1191] 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204
## [1205] 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218
## [1219] 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232
## [1233] 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246
## [1247] 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260
## [1261] 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274
## [1275] 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288
## [1289] 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302
## [1303] 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316
## [1317] 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330
## [1331] 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340

##
## $beg
## [1] 0

```

3.2.1.7 ZigZag

- 功能：过滤微小价格波动，连接极端点凸显趋势。

- 核心参数:

- HL: 高低价或收盘价序列。
- change: 最小价格变动 (绝对值或百分比)。
- percent: 是否百分比变动。

应用: 可可视化历史趋势转折点, 非预测性工具。注意: 最后值不稳定, 不适合实时交易。

```
zz <- ZigZag(xts(data.frame(high = Hi(TSLA),
                           low = Lo(TSLA)
                           ),
                           order.by = index(TSLA),
                           ),
                           change = 10,
                           percent = TRUE,
                           retrace = FALSE,
                           lastExtreme = TRUE
)
tail(zz,5)

##          [,1]
## 2025-04-25 280.446
## 2025-04-28 294.860
## 2025-04-29      NA
## 2025-04-30      NA
## 2025-05-01      NA

# 提取收盘价
close_price <- C1(TSLA)

# 计算 ZigZag 指标 (使用 12% 的回撤百分比)
zigzag <- TTR::ZigZag(close_price, percent = 12)

# 转换为数据框以便 ggplot2 使用
tsla_df <- data.frame(
```

```
Date = index(TSLA),  
Close = as.numeric(close_price),  
ZigZag = as.numeric(zigzag)  
)  
  
# 创建 ZigZag 转折点的数据框  
zigzag_points <- tsla_df[!is.na(tsla_df$ZigZag), ]  
  
# 使用 ggplot2 绘制图表  
ggplot(tsla_df, aes(x = Date)) +  
    # 绘制收盘价线  
    geom_line(aes(y = Close), color = "black", linewidth = 0.3) +  
  
    # 绘制 ZigZag 线  
    geom_line(data = zigzag_points, aes(y = ZigZag), color = "blue", linewidth = 0.3) +  
  
    # 标记 ZigZag 转折点  
    geom_point(data = zigzag_points, aes(y = ZigZag), color = "red", size = 0.1) +  
  
    # 添加标题和标签  
    labs(  
        title = "特斯拉 (TSLA) 股票价格与 ZigZag 指标",  
        subtitle = paste("从", min tsla_df$Date, "到", max tsla_df$Date)),  
        y = "价格 (美元)",  
        x = "日期",  
        caption = "ZigZag 指标基于 12% 的回撤百分比计算"  
    ) +  
  
    # 设置主题  
    theme_minimal() +  
  
    # 自定义主题  
    theme(
```

```
plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
plot.subtitle = element_text(size = 12, hjust = 0.5, color = "gray40"),
axis.title = element_text(face = "bold"),
panel.grid.minor = element_blank(),
legend.position = "bottom"
)
```



3.3 滚动窗口类函数

- runSum 函数

功能

计算 n 周期移动窗口的和

应用

常用于计算滚动累计收益或交易量聚合

```
tsla_runsum <- runSum(Vo(TSLA), n=10)
tail(tsla_runsum)
```

```
## [1]
## 2025-04-24 5.680057e-315
## 2025-04-25 5.610088e-315
## 2025-04-28 5.722654e-315
## 2025-04-29 5.765991e-315
## 2025-04-30 6.009895e-315
## 2025-05-01 5.947051e-315
```

- runMin 函数

功能

计算 n 周期移动窗口的最小值

应用

识别价格支撑阻力位或波动区间

```
tsla_runmin<- runMin(Cl(TSLA))
tail(tsla_runmin)
```

```
## [1]
## 2025-04-24 227.5
## 2025-04-25 227.5
## 2025-04-28 227.5
## 2025-04-29 227.5
## 2025-04-30 227.5
## 2025-05-01 227.5
```

- runMax 函数

功能

计算 n 周期移动窗口的最大值

应用

识别价格支撑阻力位或波动区间

```
tsla_runmax<- runMax(Cl(TSLA))  
tail(tsla_runmax)
```

```
## [1]  
## 2025-04-24 259.51  
## 2025-04-25 284.95  
## 2025-04-28 285.88  
## 2025-04-29 292.03  
## 2025-04-30 292.03  
## 2025-05-01 292.03
```

- runMean 函数

功能

计算 n 周期移动窗口的均值

应用

生成移动平均线（MA）以平滑价格趋势

```
tsla_runmean <- runMean(Cl(TSLA),10)  
tail(tsla_runmean)
```

```
## [1]  
## 2025-04-24 246.981  
## 2025-04-25 250.236  
## 2025-04-28 253.593  
## 2025-04-29 257.561  
## 2025-04-30 260.366  
## 2025-05-01 264.263
```

- runMedian 函数

功能

计算 n 周期移动窗口的中位数

应用

减少异常值影响，用于稳健趋势分析

```
tsla_runmedian <- runMedian(Cl(TSLA), 10)  
tail(tsla_runmedian)
```

```
## [,1]  
## 2025-04-24 251.525  
## 2025-04-25 251.525  
## 2025-04-28 251.545  
## 2025-04-29 252.425  
## 2025-04-30 255.125  
## 2025-05-01 270.015
```

- runCov 函数

功能

计算 n 周期滚动协方差/相关系数

应用

资产相关性分析或配对交易策略

- runCor 函数

功能

计算 n 周期滚动协方差/相关系数

应用

资产相关性分析或配对交易策略

```
tsla_runcor <- runCor(Cl(TSLA), Cl(AAPL))  
tail(tsla_runcor)
```

```
## [,1]  
## 2025-04-24 0.5690956  
## 2025-04-25 0.8488571
```

```
## 2025-04-28 0.8966050
## 2025-04-29 0.9143418
## 2025-04-30 0.9216754
## 2025-05-01 0.9296932
```

- runVar 函数

功能

计算 n 周期滚动方差/标准差

应用

量化风险指标（如波动率计算）

```
tsla_runvar <- runVar(Cl(TSLA), Cl(AAPL))
tail(tsla_runvar)
```

```
## [,1]
## 2025-04-24 30.31446
## 2025-04-25 71.90199
## 2025-04-28 104.57962
## 2025-04-29 139.62491
## 2025-04-30 162.06618
## 2025-05-01 154.23768
```

- runSD 函数

功能

计算 n 周期滚动方差/标准差

应用

量化风险指标（如波动率计算）

```
tsla_runsd <- runSD(Cl(TSLA), n=10)
tail(tsla_runsd)
```

```
## [,1]
```

```
## 2025-04-24 9.604427  
## 2025-04-25 15.407553  
## 2025-04-28 19.119603  
## 2025-04-29 22.628499  
## 2025-04-30 23.858290  
## 2025-05-01 23.624934
```

- runMAD 函数

功能

计算 n 周期滚动中位数/均值绝对偏差

应用

评估数据离散程度的稳健方法

```
tsla_runmad <- runMAD(Cl(TSLA),n=10)  
tail(tsla_runmad)
```

```
## [,1]  
## 2025-04-24 7.835541  
## 2025-04-25 13.313748  
## 2025-04-28 14.952021  
## 2025-04-29 18.910563  
## 2025-04-30 32.758047  
## 2025-05-01 26.049282
```

- wilderSum 函数

功能

按 Welles Wilder 加权法计算滚动窗口加权和

应用

经典技术指标（如 RSI）的核心计算逻辑

```
tsla_wildersum <- wilderSum(Cl(TSLA),n=10)  
tail(tsla_wildersum)
```

```
## [1]
## 2025-04-24 2512.647
## 2025-04-25 2546.332
## 2025-04-28 2577.579
## 2025-04-29 2611.851
## 2025-04-30 2632.826
## 2025-05-01 2650.064
```

- runPercentRank 函数

功能

在指定时间窗口内（如 20 日、60 日等）动态计算当前值在历史数据中的百分比排名。

应用

- 量化交易信号：识别资产价格在近期历史中的相对强弱位置，生成超买/超卖信号 57
- 风险监控：监测指标（如波动率）在滚动周期内的极端百分位状态 3
- 绩效归因：评估投资组合收益在同类产品中的滚动排名变化

```
tsla_runpercentrank <- runPercentRank(Cl(TSLA))
tail(tsla_runpercentrank)
```

```
## [1]
## 2025-04-24 0.6173077
## 2025-04-25 0.7019231
## 2025-04-28 0.7019231
## 2025-04-29 0.7134615
## 2025-04-30 0.6826923
## 2025-05-01 0.6750000
```

- rollSFM 函数

功能

- 滚动窗口计算在指定时间窗口内（如 20 日、60 日等）动态计算统计

特征，支持均值、标准差、偏度、峰度等指标

- 多维度分析可同时输出窗口期内数据的分布特征和形态指标，形成结构化统计矩阵
- 高效处理采用向量化计算优化性能，支持大数据量下的滚动窗口快速迭代

应用

量化因子构建生成资产价格的滚动波动特征作为因子输入，如 20 日滚动波动率排名 13 异常检测通过滚动统计识别数据分布突变点（如波动率骤升）7 模型特征工程为机器学习模型提供时间序列的动态统计特征

```
tsla_rollsrm <- rollSFM(Cl(TSLA), Cl(AAPL))
```

```
tail(tsla_rollsrm)
```

```
##           alpha      beta r.squared
## 2025-04-24 -152.42661 1.991257 0.4659584
## 2025-04-25 -137.39875 1.919991 0.4565924
## 2025-04-28 -120.80631 1.840173 0.4524928
## 2025-04-29 -103.47017 1.756432 0.4529799
## 2025-04-30  -95.97511 1.716753 0.4683615
## 2025-05-01  -83.56101 1.654331 0.4802689
```

4 与 quantmod 包及 quantstrat 包的协作

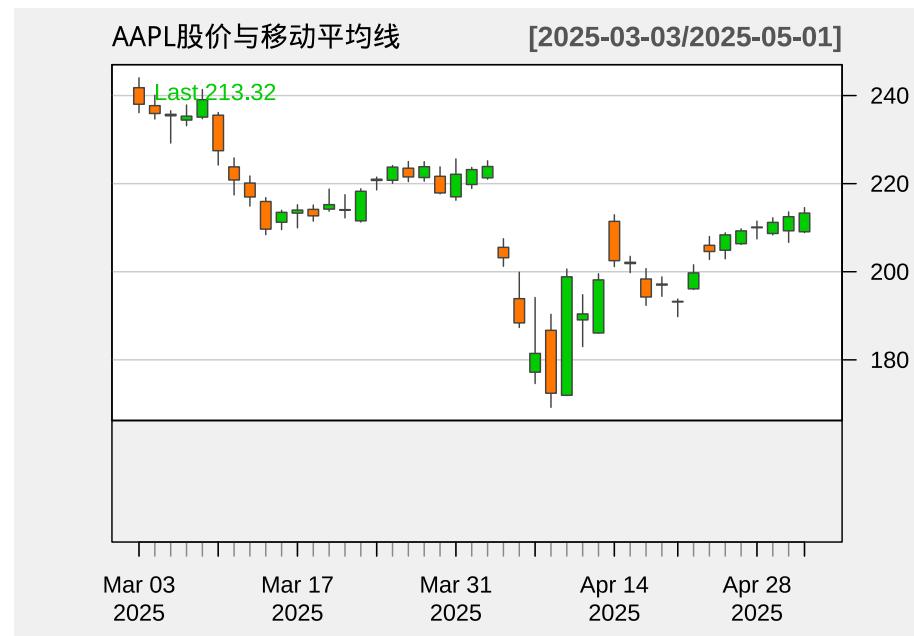
4.1 与 quantmod 包协作进行金融数据可视化

使用 TTR 包计算一些常用的技术指标，并结合 quantmod 包进行可视化：

```
### 移动平均线
# 计算 5 日和 20 日简单移动平均线 (SMA) 和指数移动平均线 (EMA):
# 计算移动平均线
AAPL$SMA5 <- SMA(Cl(AAPL), n = 5)
AAPL$SMA20 <- SMA(Cl(AAPL), n = 20)
AAPL$EMA5 <- EMA(Cl(AAPL), n = 5)
```

```
AAPL$EMA20 <- EMA(Cl(AAPL), n = 20)
```

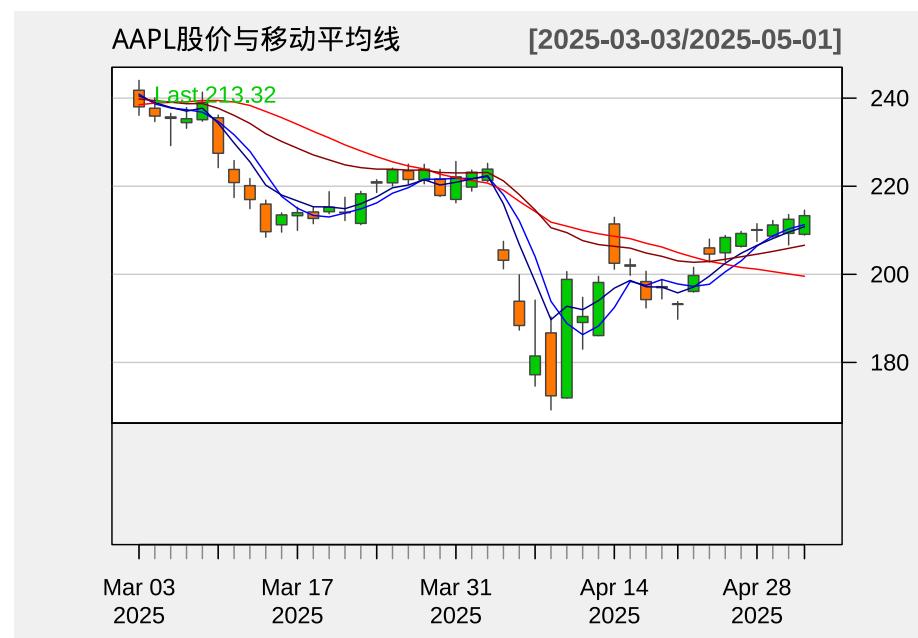
```
# 绘制价格和移动平均线  
chartSeries(AAPL, subset = "last 3 months",  
            theme = chartTheme("white"),  
            name = "AAPL 股价与移动平均线")
```



```
addSMA(n = c(5, 20), on = 1, col = c("blue", "red"))
```

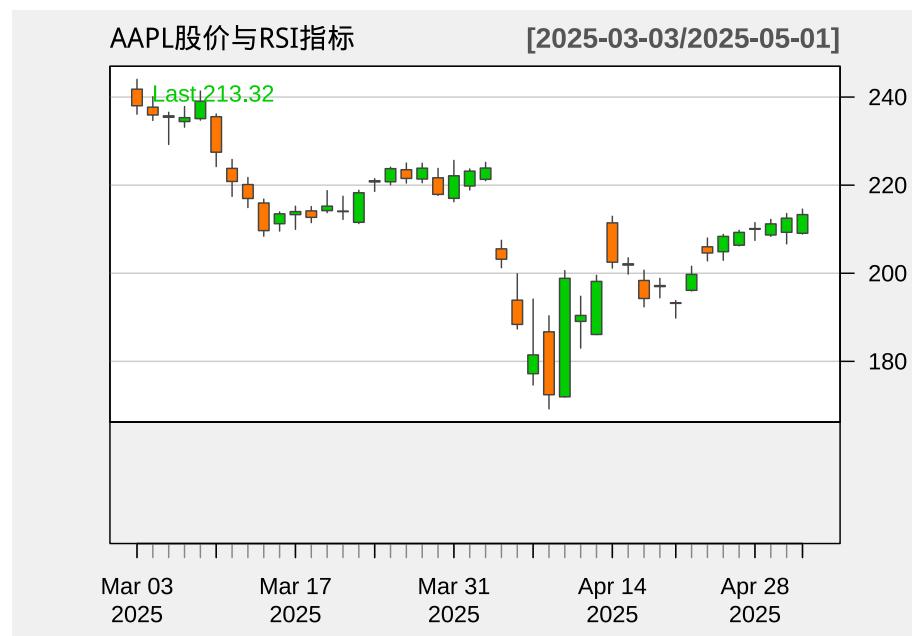


```
addEMA(n = c(5, 20), on = 1, col = c("darkblue", "darkred"))
```

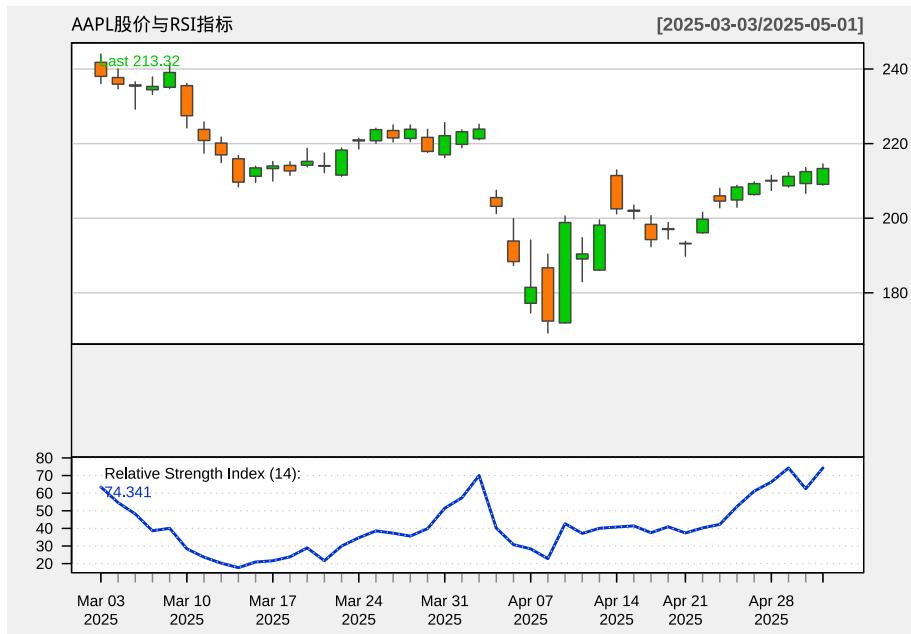


```
### RSI 指标
# 计算 14 日相对强弱指数 (RSI):
# 计算 RSI
AAPL$RSI14 <- RSI(Cl(AAPL), n = 14)

# 绘制价格和 RSI
chartSeries(AAPL, subset = "last 3 months",
            theme = chartTheme("white"),
            name = "AAPL 股价与 RSI 指标")
```

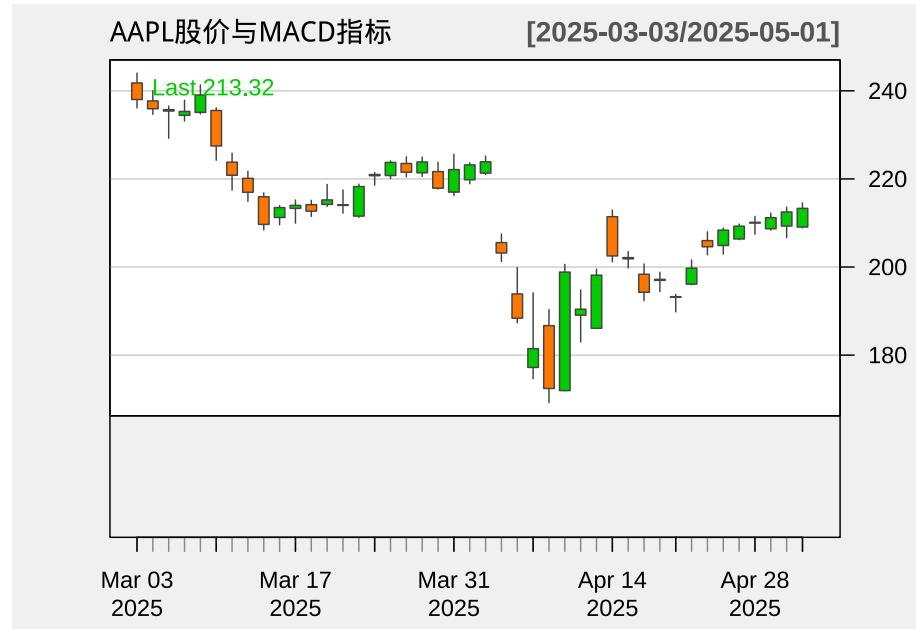


```
addRSI(n = 14, maType = "SMA")
```

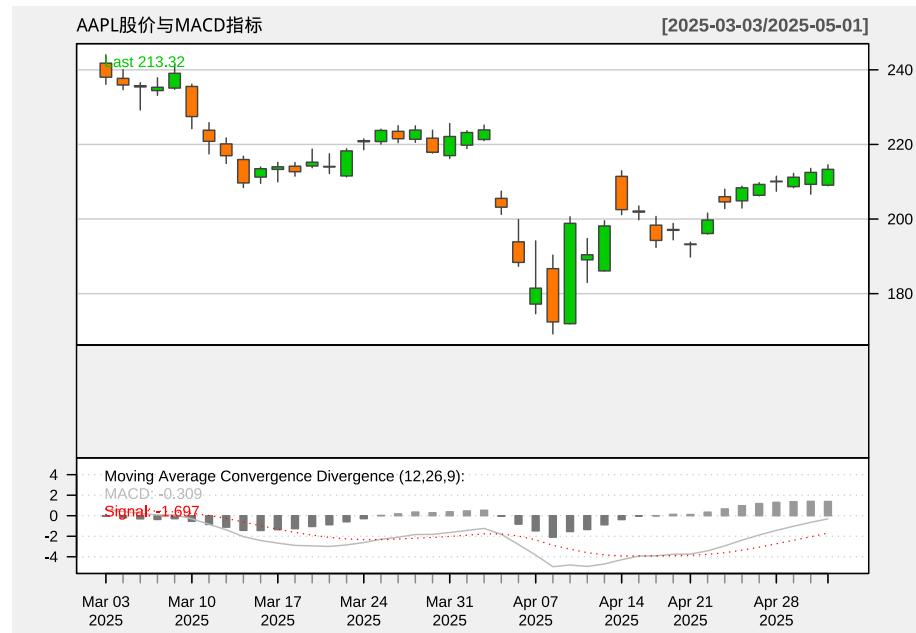


```
### MACD 指标
# 计算 MACD 指标:
# 计算 MACD
AAPL$MACD <- MACD(Cl(AAPL), fast = 12, slow = 26, signal = 9)

# 绘制价格和 MACD
chartSeries(AAPL, subset = "last 3 months",
            theme = chartTheme("white"),
            name = "AAPL 股价与 MACD 指标")
```

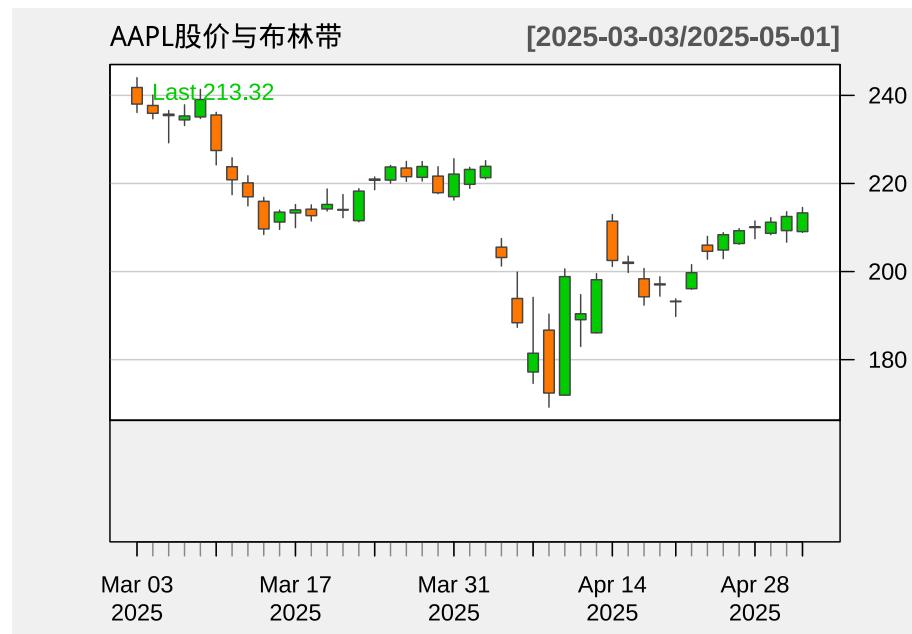


```
addMACD(fast = 12, slow = 26, signal = 9)
```

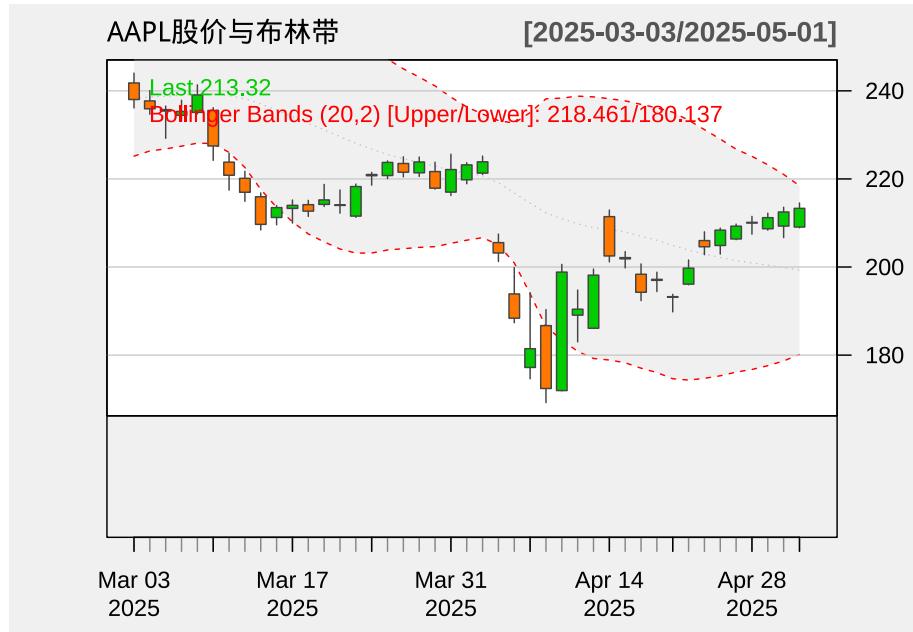


```
## 布林带指标
## 计算布林带并绘制：
# 计算布林带
AAPL$BBands <- BBands(Cl(AAPL), n = 20, sd = 2)

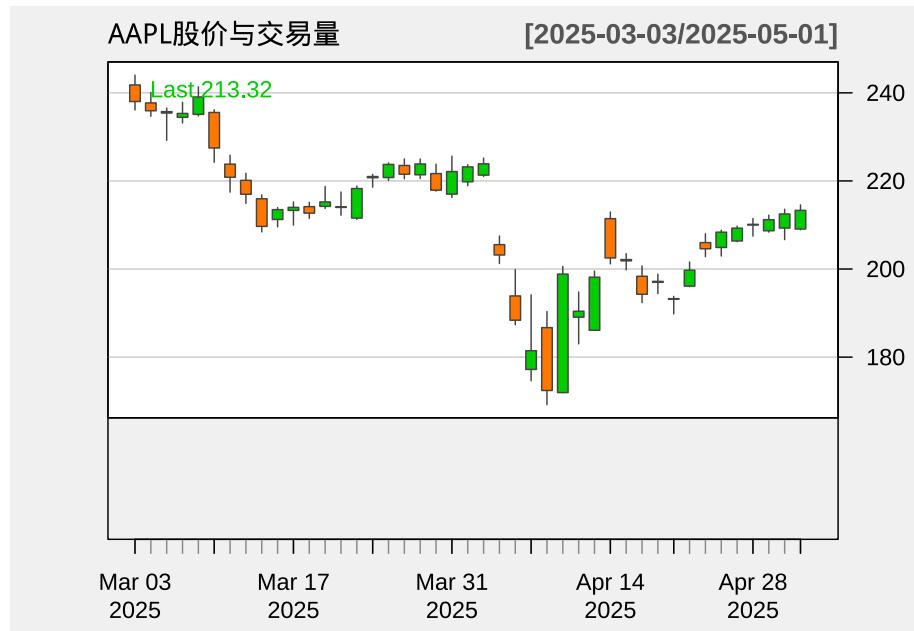
# 绘制价格和布林带
chartSeries(AAPL, subset = "last 3 months",
            theme = chartTheme("white"),
            name = "AAPL 股价与布林带")
```



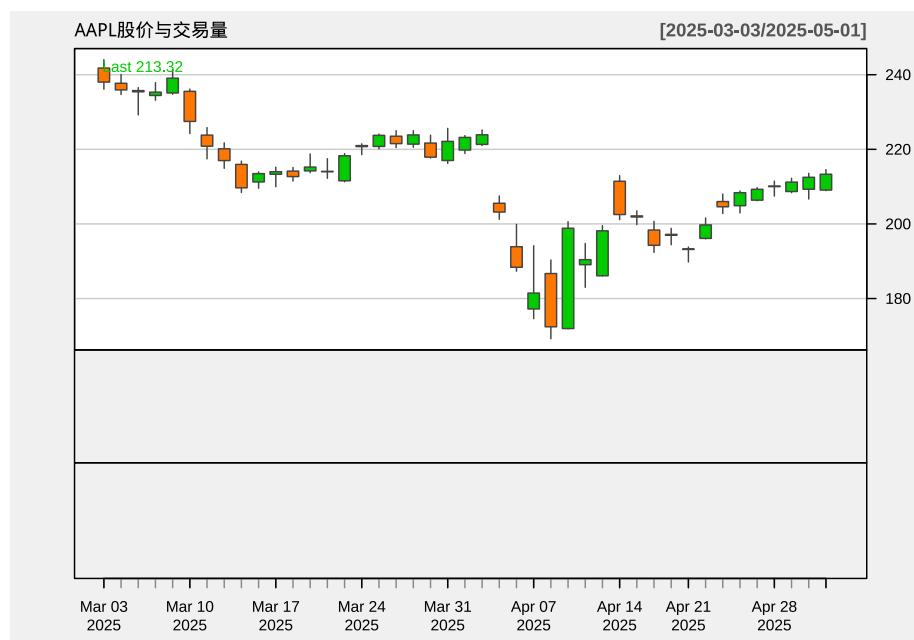
```
addBBands(n = 20, sd = 2)
```



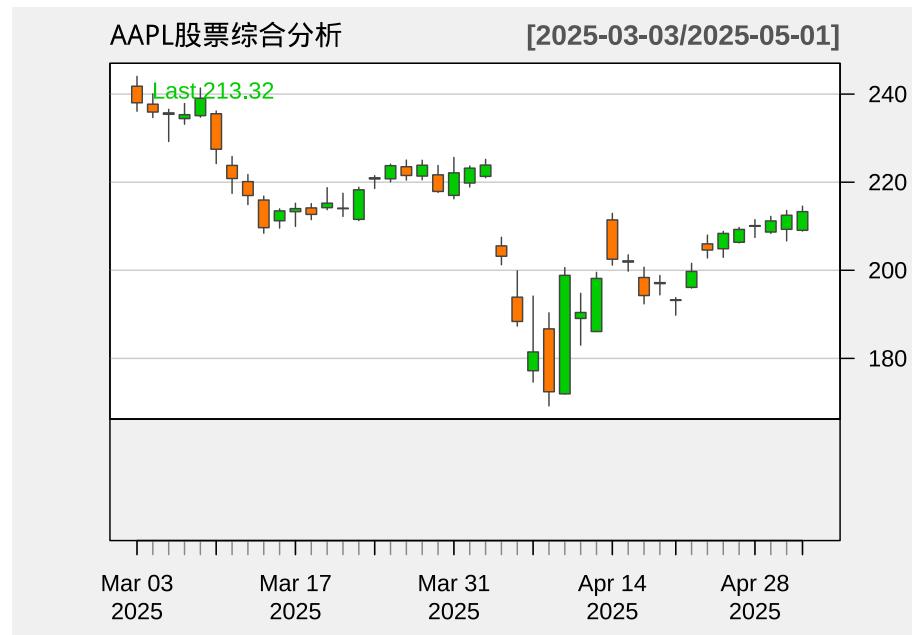
```
## 交易量分析  
## 结合价格和交易量进行分析:  
# 绘制价格和交易量  
chartSeries(AAPL, subset = "last 3 months",  
            theme = chartTheme("white"),  
            name = "AAPL 股价与交易量")
```



addVo()



```
## 综合指标分析  
## 将多个指标组合在一起进行综合分析:  
# 绘制综合指标图  
chartSeries(AAPL, subset = "last 3 months",  
            theme = chartTheme("white"),  
            name = "AAPL 股票综合分析")
```



```
addSMA(n = c(5, 20, 50), on = 1)
```



```
addBands(n = 20, sd = 2)
```



```
addRSI(n = 14)
```



```
addMACD(fast = 12, slow = 26, signal = 9)
```



`addVo()`



4.2 与 quantstrat 包协作开发量化投资策略

设置初始参数和环境：

```
# 加载 quantstrat 包，用于量化交易策略回测
require(quantstrat)

# 清除之前运行的策略相关数据，避免干扰当前回测
suppressWarnings(rm("order_book.bbands", pos = .strategy))
suppressWarnings(rm("account.bbands", "portfolio.bbands", pos = .blotter))
suppressWarnings(rm("account.st", "portfolio.st", "stock.str", "stratBBands", "startDate"))

# 设置回测参数
stock.str = 'TSLA' # 交易标的为 IBM 股票
SD = 2 # 布林带的标准偏差倍数，传统取值为 2
N = 20 # 布林带移动平均的周期数，传统取值为 20
```

```
# 初始化金融工具
currency('USD') # 设置基础货币为美元

## [1] "USD"

stock(stock.str, currency = 'USD', multiplier = 1) # 定义股票交易品种

## [1] "TSLA"

# 设置回测时间范围和初始资金
startDate = '2006-12-31' # 回测开始日期
initEq = 1000000 # 初始资金为 100 万美元

# 设置投资组合和账户名称
portfolio.st = 'bbands' # 投资组合名称
account.st = 'bbands' # 账户名称

# 初始化投资组合、账户和订单簿
initPortf(portfolio.st, symbols = stock.str) # 初始化投资组合

## [1] "bbands"

initAcct(account.st, portfolios = 'bbands') # 初始化账户

## [1] "bbands"

initOrders(portfolio = portfolio.st) # 初始化订单簿

# 设置持仓限制（最大持仓量为 200 股，最小为 2 股）
addPosLimit(portfolio.st, stock.str, startDate, 200, 2)

# 设置策略参数
maType = 'SMA' # 移动平均类型为简单移动平均
n = 20 # 移动平均周期
sdp = 2 # 标准偏差倍数
```

```
# 定义策略名称并初始化
strat.st <- portfolio.st
strategy(strat.st, store = TRUE) # 创建策略对象并启用存储功能

# 添加布林带指标
add.indicator(strategy = strat.st,
               name = "BBands",
               arguments = list(HLC = quote(HLC(mktdata)), # 使用最高价、最低价、收盘价计算
                                n = n, # 移动平均周期
                                maType = maType, # 移动平均类型
                                sdp = sdp # 标准偏差倍数
               ),
               label = 'BBands') # 指标标签

## [1] "bbands"

# 添加交易信号: 当收盘价突破上轨时产生卖出信号
add.signal(strategy = strat.st,
           name = "sigCrossover",
           arguments = list(columns = c("Close", "up"), # 比较收盘价和上轨
                            relationship = "gt"), # 大于关系
           label = "Cl.gt.UpperBand") # 信号标签

## [1] "bbands"

# 添加交易信号: 当收盘价跌破下轨时产生买入信号
add.signal(strategy = strat.st,
           name = "sigCrossover",
           arguments = list(columns = c("Close", "dn"), # 比较收盘价和下轨
                            relationship = "lt"), # 小于关系
           label = "Cl.lt.LowerBand") # 信号标签

## [1] "bbands"
```

```
# 添加交易信号：当最高价或最低价穿过中轨时产生平仓信号
add.signal(strategy = strat.st, name = "sigCrossover",
           arguments = list(columns = c("High", "Low", "mavg"),
                             relationship = "op"),
           label = "Cross.Mid") # 信号标签

## [1] "bbands"

# 添加交易规则：当收盘价突破上轨时卖出 100 股
add.rule(strategy = strat.st, name = 'ruleSignal',
          arguments = list(sigcol = "Cl.gt.UpperBand", # 信号列
                           sigval = TRUE, # 信号值为真时触发
                           orderqty = -100, # 卖出 100 股
                           ordertype = 'market', # 市价单
                           orderside = NULL, # 自动确定买卖方向
                           threshold = NULL, # 无阈值限制
                           osFUN = osMaxPos), # 最大持仓量函数
          type = 'enter') # 入场规则

## [1] "bbands"

# 添加交易规则：当收盘价跌破下轨时买入 100 股
add.rule(strategy = strat.st, name = 'ruleSignal',
          arguments = list(sigcol = "Cl.lt.LowerBand", # 信号列
                           sigval = TRUE, # 信号值为真时触发
                           orderqty = 100, # 买入 100 股
                           ordertype = 'market', # 市价单
                           orderside = NULL, # 自动确定买卖方向
                           threshold = NULL, # 无阈值限制
                           osFUN = osMaxPos), # 最大持仓量函数
          type = 'enter') # 入场规则

## [1] "bbands"
```

```
# 添加交易规则：当价格穿过中轨时平仓
add.rule(strategy = strat.st, name = 'ruleSignal',
         arguments = list(sigcol = "Cross.Mid",           # 信号列
                           sigval = TRUE,                 # 信号值为真时触发
                           orderqty = 'all',              # 全部平仓
                           ordertype = 'market',          # 市价单
                           orderside = NULL,              # 自动确定买卖方向
                           threshold = NULL,              # 无阈值限制
                           osFUN = osMaxPos),             # 最大持仓量函数
         label = 'exitMid',                  # 规则标签
         type = 'exit')  # 出场规则

## [1] "bbands"

# 获取股票数据
getSymbols(stock.str, from = startDate, index.class = c('POSIXt', 'POSIXct'), src = 'ya

## [1] "TSLA"

# 应用策略并计时
start_t <- Sys.time()
out <- try(applyStrategy(strategy = 'bbands', portfolios = 'bbands', parameters = list()

## [1] "2010-08-11 00:00:00 TSLA 100 @ 1.19333302974701"
## [1] "2010-08-24 00:00:00 TSLA -100 @ 1.27999997138977"
## [1] "2010-09-03 00:00:00 TSLA -100 @ 1.40333294868469"
## [1] "2010-09-16 00:00:00 TSLA -100 @ 1.39600002765656"
## [1] "2010-09-20 00:00:00 TSLA 200 @ 1.40400004386902"
## [1] "2010-09-30 00:00:00 TSLA -100 @ 1.36066699028015"
## [1] "2010-10-01 00:00:00 TSLA 100 @ 1.37333297729492"
## [1] "2010-11-01 00:00:00 TSLA -100 @ 1.42733299732208"
## [1] "2010-11-05 00:00:00 TSLA -100 @ 1.62933301925659"
## [1] "2010-12-07 00:00:00 TSLA 200 @ 2.10400009155273"
## [1] "2010-12-28 00:00:00 TSLA 100 @ 1.76066696643829"
## [1] "2011-01-12 00:00:00 TSLA -100 @ 1.79733300209045"
```

```
## [1] "2011-01-21 00:00:00 TSLA 100 @ 1.53600001335144"
## [1] "2011-02-09 00:00:00 TSLA -100 @ 1.54733300209045"
## [1] "2011-02-23 00:00:00 TSLA 100 @ 1.45533299446106"
## [1] "2011-02-28 00:00:00 TSLA -100 @ 1.59266698360443"
## [1] "2011-03-07 00:00:00 TSLA -100 @ 1.66266703605652"
## [1] "2011-03-14 00:00:00 TSLA 100 @ 1.54999995231628"
## [1] "2011-04-01 00:00:00 TSLA -100 @ 1.77733302116394"
## [1] "2011-04-13 00:00:00 TSLA 100 @ 1.66199994087219"
## [1] "2011-05-26 00:00:00 TSLA -100 @ 1.96533298492432"
## [1] "2011-06-09 00:00:00 TSLA 100 @ 1.84133303165436"
## [1] "2011-08-05 00:00:00 TSLA 100 @ 1.61600005626678"
## [1] "2011-08-11 00:00:00 TSLA 100 @ 1.68666696548462"
## [1] "2011-08-15 00:00:00 TSLA -200 @ 1.74866700172424"
## [1] "2011-08-22 00:00:00 TSLA 100 @ 1.46333301067352"
## [1] "2011-08-30 00:00:00 TSLA -100 @ 1.64199995994568"
## [1] "2011-09-19 00:00:00 TSLA -100 @ 1.71800005435944"
## [1] "2011-09-29 00:00:00 TSLA 100 @ 1.60800004005432"
## [1] "2011-10-11 00:00:00 TSLA -100 @ 1.84066700935364"
## [1] "2011-11-04 00:00:00 TSLA -100 @ 2.15400004386902"
## [1] "2011-11-22 00:00:00 TSLA 200 @ 2.13800001144409"
## [1] "2011-12-07 00:00:00 TSLA -100 @ 2.27933311462402"
## [1] "2011-12-09 00:00:00 TSLA 100 @ 2.06933307647705"
## [1] "2011-12-14 00:00:00 TSLA 100 @ 1.90199995040894"
## [1] "2011-12-19 00:00:00 TSLA 100 @ 1.85000002384186"
## [1] "2012-01-12 00:00:00 TSLA -200 @ 1.88333296775818"
## [1] "2012-01-17 00:00:00 TSLA 100 @ 1.77333295345306"
## [1] "2012-01-20 00:00:00 TSLA -100 @ 1.77333295345306"
## [1] "2012-02-06 00:00:00 TSLA -100 @ 2.11999988555908"
## [1] "2012-02-21 00:00:00 TSLA -100 @ 2.29999995231628"
## [1] "2012-03-07 00:00:00 TSLA 200 @ 2.20799994468689"
## [1] "2012-03-13 00:00:00 TSLA -100 @ 2.40599989891052"
## [1] "2012-03-23 00:00:00 TSLA 100 @ 2.2720000743866"
## [1] "2012-03-27 00:00:00 TSLA -100 @ 2.52933311462402"
```

```
## [1] "2012-04-05 00:00:00 TSLA 100 @ 2.29866695404053"
## [1] "2012-04-11 00:00:00 TSLA 100 @ 2.20600008964539"
## [1] "2012-05-02 00:00:00 TSLA -100 @ 2.26266694068909"
## [1] "2012-05-09 00:00:00 TSLA 100 @ 2.00399994850159"
## [1] "2012-05-11 00:00:00 TSLA -100 @ 2.15000009536743"
## [1] "2012-05-15 00:00:00 TSLA 100 @ 1.96200001239777"
## [1] "2012-05-25 00:00:00 TSLA -100 @ 1.98733305931091"
## [1] "2012-06-19 00:00:00 TSLA -100 @ 2.13933300971985"
## [1] "2012-06-25 00:00:00 TSLA -100 @ 2.20733308792114"
## [1] "2012-07-03 00:00:00 TSLA 200 @ 2.04399991035461"
## [1] "2012-07-16 00:00:00 TSLA -100 @ 2.39733290672302"
## [1] "2012-07-19 00:00:00 TSLA 100 @ 2.15133309364319"
## [1] "2012-07-26 00:00:00 TSLA 100 @ 1.87533295154572"
## [1] "2012-07-31 00:00:00 TSLA 100 @ 1.82799994945526"
## [1] "2012-08-08 00:00:00 TSLA -200 @ 1.93933296203613"
## [1] "2012-09-11 00:00:00 TSLA 100 @ 1.85333299636841"
## [1] "2012-09-14 00:00:00 TSLA -100 @ 2.02600002288818"
## [1] "2012-09-18 00:00:00 TSLA -100 @ 2.08933305740356"
## [1] "2012-09-26 00:00:00 TSLA 100 @ 1.83599996566772"
## [1] "2012-11-06 00:00:00 TSLA -100 @ 2.07666707038879"
## [1] "2012-12-04 00:00:00 TSLA -100 @ 2.25999999046326"
## [1] "2012-12-14 00:00:00 TSLA 200 @ 2.25399994850159"
## [1] "2013-01-03 00:00:00 TSLA -100 @ 2.31800007820129"
## [1] "2013-01-07 00:00:00 TSLA 100 @ 2.28933310508728"
## [1] "2013-01-14 00:00:00 TSLA 100 @ 2.2173330783844"
## [1] "2013-01-16 00:00:00 TSLA -100 @ 2.27333307266235"
## [1] "2013-01-23 00:00:00 TSLA -100 @ 2.40000009536743"
## [1] "2013-02-19 00:00:00 TSLA 100 @ 2.61866688728333"
## [1] "2013-02-22 00:00:00 TSLA 100 @ 2.40733289718628"
## [1] "2013-03-07 00:00:00 TSLA -100 @ 2.54866695404053"
## [1] "2013-04-02 00:00:00 TSLA -100 @ 2.95600008964539"
## [1] "2013-04-23 00:00:00 TSLA -100 @ 3.40066695213318"
## [1] "2013-06-06 00:00:00 TSLA 200 @ 6.48999977111816"
```

```
## [1] "2013-06-28 00:00:00 TSLA -100 @ 7.15733289718628"
## [1] "2013-07-02 00:00:00 TSLA -100 @ 7.85466718673706"
## [1] "2013-07-17 00:00:00 TSLA 200 @ 8.01666736602783"
## [1] "2013-07-30 00:00:00 TSLA -100 @ 8.78266716003418"
## [1] "2013-08-05 00:00:00 TSLA -100 @ 9.6453332901001"
## [1] "2013-08-16 00:00:00 TSLA 200 @ 9.46666717529297"
## [1] "2013-08-23 00:00:00 TSLA -100 @ 10.7893333435059"
## [1] "2013-09-18 00:00:00 TSLA 100 @ 11.0813331604004"
## [1] "2013-09-20 00:00:00 TSLA -100 @ 12.2259998321533"
## [1] "2013-10-04 00:00:00 TSLA 100 @ 12.065333366394"
## [1] "2013-10-24 00:00:00 TSLA 100 @ 11.5433330535889"
## [1] "2013-10-31 00:00:00 TSLA 100 @ 10.6626672744751"
## [1] "2013-11-05 00:00:00 TSLA -200 @ 11.78733253479"
## [1] "2013-11-07 00:00:00 TSLA 100 @ 9.31799983978271"
## [1] "2013-11-19 00:00:00 TSLA 100 @ 8.4060001373291"
## [1] "2013-12-04 00:00:00 TSLA -200 @ 9.26333332061768"
## [1] "2014-01-14 00:00:00 TSLA 100 @ 10.7513332366943"
## [1] "2014-01-15 00:00:00 TSLA -100 @ 10.9420003890991"
## [1] "2014-02-11 00:00:00 TSLA -100 @ 13.1079998016357"
## [1] "2014-02-14 00:00:00 TSLA -100 @ 13.2153329849243"
## [1] "2014-03-17 00:00:00 TSLA 200 @ 15.5986671447754"
## [1] "2014-03-25 00:00:00 TSLA 100 @ 14.6960000991821"
## [1] "2014-03-27 00:00:00 TSLA 100 @ 13.8213329315186"
## [1] "2014-04-03 00:00:00 TSLA -200 @ 15.0266666412354"
## [1] "2014-05-09 00:00:00 TSLA 100 @ 12.1506671905518"
## [1] "2014-05-21 00:00:00 TSLA -100 @ 13.296667098999"
## [1] "2014-06-17 00:00:00 TSLA -100 @ 15.4446668624878"
## [1] "2014-06-24 00:00:00 TSLA -100 @ 15.5"
## [1] "2014-07-07 00:00:00 TSLA 200 @ 14.8439998626709"
## [1] "2014-08-04 00:00:00 TSLA -100 @ 15.9013328552246"
## [1] "2014-08-12 00:00:00 TSLA -100 @ 17.3306674957275"
## [1] "2014-09-16 00:00:00 TSLA 200 @ 17.3826675415039"
## [1] "2014-10-13 00:00:00 TSLA 100 @ 14.9726667404175"
```

```
## [1] "2014-10-29 00:00:00 TSLA -100 @ 15.8733329772949"
## [1] "2014-11-12 00:00:00 TSLA -100 @ 16.6066665649414"
## [1] "2014-11-14 00:00:00 TSLA -100 @ 17.2453327178955"
## [1] "2014-11-24 00:00:00 TSLA 200 @ 16.4479999542236"
## [1] "2014-12-02 00:00:00 TSLA 100 @ 15.4286670684814"
## [1] "2014-12-08 00:00:00 TSLA 100 @ 14.2906665802002"
## [1] "2014-12-23 00:00:00 TSLA -200 @ 14.7313327789307"
## [1] "2015-01-15 00:00:00 TSLA 100 @ 12.7913331985474"
## [1] "2015-01-27 00:00:00 TSLA -100 @ 13.7320003509521"
## [1] "2015-02-04 00:00:00 TSLA -100 @ 14.5699996948242"
## [1] "2015-02-13 00:00:00 TSLA 100 @ 13.5846672058105"
## [1] "2015-03-10 00:00:00 TSLA 100 @ 12.6879997253418"
## [1] "2015-03-19 00:00:00 TSLA -100 @ 13.0433330535889"
## [1] "2015-03-30 00:00:00 TSLA 100 @ 12.7046670913696"
## [1] "2015-04-07 00:00:00 TSLA -100 @ 13.5500001907349"
## [1] "2015-04-09 00:00:00 TSLA -100 @ 14.0059995651245"
## [1] "2015-04-28 00:00:00 TSLA -100 @ 15.3653326034546"
## [1] "2015-06-08 00:00:00 TSLA 200 @ 17.0860004425049"
## [1] "2015-06-09 00:00:00 TSLA -100 @ 17.0666675567627"
## [1] "2015-06-16 00:00:00 TSLA 100 @ 16.8746662139893"
## [1] "2015-06-18 00:00:00 TSLA -100 @ 17.4593334197998"
## [1] "2015-06-24 00:00:00 TSLA -100 @ 17.67799949646"
## [1] "2015-07-08 00:00:00 TSLA 200 @ 16.9973335266113"
## [1] "2015-07-21 00:00:00 TSLA -100 @ 17.7846660614014"
## [1] "2015-07-22 00:00:00 TSLA 100 @ 17.8579998016357"
## [1] "2015-07-28 00:00:00 TSLA 100 @ 17.6546669006348"
## [1] "2015-07-30 00:00:00 TSLA -100 @ 17.7859992980957"
## [1] "2015-08-07 00:00:00 TSLA 100 @ 16.1673336029053"
## [1] "2015-08-18 00:00:00 TSLA -100 @ 17.3813323974609"
## [1] "2015-08-25 00:00:00 TSLA 100 @ 14.6686668395996"
## [1] "2015-08-31 00:00:00 TSLA -100 @ 16.6040000915527"
## [1] "2015-10-08 00:00:00 TSLA 100 @ 15.1146669387817"
## [1] "2015-11-05 00:00:00 TSLA -100 @ 15.4513330459595"
```

```
## [1] "2016-01-11 00:00:00 TSLA 100 @ 13.8566665649414"
## [1] "2016-01-14 00:00:00 TSLA 100 @ 13.7453327178955"
## [1] "2016-02-23 00:00:00 TSLA -200 @ 11.8140001296997"
## [1] "2016-03-21 00:00:00 TSLA -100 @ 15.8879995346069"
## [1] "2016-04-06 00:00:00 TSLA -100 @ 17.6946678161621"
## [1] "2016-04-20 00:00:00 TSLA 200 @ 16.6646671295166"
## [1] "2016-05-02 00:00:00 TSLA 100 @ 16.1200008392334"
## [1] "2016-05-04 00:00:00 TSLA 100 @ 14.8373327255249"
## [1] "2016-05-24 00:00:00 TSLA -200 @ 14.5273332595825"
## [1] "2016-06-08 00:00:00 TSLA -100 @ 15.7013330459595"
## [1] "2016-06-13 00:00:00 TSLA 100 @ 14.5246667861938"
## [1] "2016-06-23 00:00:00 TSLA 100 @ 13.0933332443237"
## [1] "2016-07-05 00:00:00 TSLA -100 @ 14.2653331756592"
## [1] "2016-08-01 00:00:00 TSLA -100 @ 15.3339996337891"
## [1] "2016-08-03 00:00:00 TSLA 100 @ 15.0526666641235"
## [1] "2016-08-29 00:00:00 TSLA 100 @ 14.3466672897339"
## [1] "2016-09-20 00:00:00 TSLA -100 @ 13.6426668167114"
## [1] "2016-10-04 00:00:00 TSLA -100 @ 14.0939998626709"
## [1] "2016-10-07 00:00:00 TSLA 100 @ 13.1073331832886"
## [1] "2016-10-18 00:00:00 TSLA 100 @ 13.2733325958252"
## [1] "2016-10-20 00:00:00 TSLA -100 @ 13.2733325958252"
## [1] "2016-11-02 00:00:00 TSLA 100 @ 12.5346670150757"
## [1] "2016-11-15 00:00:00 TSLA 100 @ 12.2513332366943"
## [1] "2016-11-21 00:00:00 TSLA -200 @ 12.3013334274292"
## [1] "2016-12-19 00:00:00 TSLA -100 @ 13.5153331756592"
## [1] "2016-12-21 00:00:00 TSLA -100 @ 13.8466672897339"
## [1] "2017-02-24 00:00:00 TSLA 200 @ 17.1333332061768"
## [1] "2017-03-28 00:00:00 TSLA -100 @ 18.4966678619385"
## [1] "2017-04-04 00:00:00 TSLA -100 @ 20.2466678619385"
## [1] "2017-05-05 00:00:00 TSLA 200 @ 20.5566673278809"
## [1] "2017-05-11 00:00:00 TSLA -100 @ 21.5400009155273"
## [1] "2017-05-18 00:00:00 TSLA 100 @ 20.8706665039062"
## [1] "2017-05-31 00:00:00 TSLA -100 @ 22.7339992523193"
```

```
## [1] "2017-06-06 00:00:00 TSLA -100 @ 23.5233325958252"
## [1] "2017-06-28 00:00:00 TSLA 200 @ 24.7493324279785"
## [1] "2017-07-06 00:00:00 TSLA 100 @ 20.5886669158936"
## [1] "2017-07-25 00:00:00 TSLA -100 @ 22.6399993896484"
## [1] "2017-08-04 00:00:00 TSLA -100 @ 23.7940006256104"
## [1] "2017-08-21 00:00:00 TSLA 100 @ 22.5240001678467"
## [1] "2017-09-14 00:00:00 TSLA -100 @ 25.1760005950928"
## [1] "2017-09-25 00:00:00 TSLA 100 @ 22.9993324279785"
## [1] "2017-10-24 00:00:00 TSLA 100 @ 22.4893321990967"
## [1] "2017-10-26 00:00:00 TSLA 100 @ 21.7446670532227"
## [1] "2017-11-17 00:00:00 TSLA -200 @ 21.0033321380615"
## [1] "2017-12-12 00:00:00 TSLA -100 @ 22.7353324890137"
## [1] "2017-12-27 00:00:00 TSLA 100 @ 20.7759990692139"
## [1] "2018-02-09 00:00:00 TSLA 100 @ 20.6946678161621"
## [1] "2018-02-20 00:00:00 TSLA -100 @ 22.318000793457"
## [1] "2018-03-20 00:00:00 TSLA 100 @ 20.703332901001"
## [1] "2018-03-26 00:00:00 TSLA 100 @ 20.2786674499512"
## [1] "2018-04-06 00:00:00 TSLA -200 @ 19.953332901001"
## [1] "2018-06-07 00:00:00 TSLA -100 @ 21.0726661682129"
## [1] "2018-06-19 00:00:00 TSLA -100 @ 23.5033321380615"
## [1] "2018-06-27 00:00:00 TSLA 200 @ 22.966667175293"
## [1] "2018-07-05 00:00:00 TSLA 100 @ 20.6106662750244"
## [1] "2018-07-31 00:00:00 TSLA 100 @ 19.8759994506836"
## [1] "2018-08-03 00:00:00 TSLA -100 @ 23.211332321167"
## [1] "2018-08-08 00:00:00 TSLA -100 @ 24.6893329620361"
## [1] "2018-09-10 00:00:00 TSLA 100 @ 19.033332824707"
## [1] "2018-09-18 00:00:00 TSLA -100 @ 18.9973335266113"
## [1] "2018-10-01 00:00:00 TSLA 100 @ 20.7133331298828"
## [1] "2018-10-02 00:00:00 TSLA -100 @ 20.068000793457"
## [1] "2018-10-08 00:00:00 TSLA 100 @ 16.7040004730225"
## [1] "2018-10-18 00:00:00 TSLA -100 @ 17.5939998626709"
## [1] "2018-10-26 00:00:00 TSLA -100 @ 22.0599994659424"
## [1] "2018-11-21 00:00:00 TSLA 100 @ 22.5459995269775"
```

```
## [1] "2018-11-26 00:00:00 TSLA 100 @ 23.0666675567627"
## [1] "2018-11-27 00:00:00 TSLA -100 @ 22.92799949646"
## [1] "2018-12-04 00:00:00 TSLA -100 @ 23.9799995422363"
## [1] "2018-12-07 00:00:00 TSLA -100 @ 23.8646678924561"
## [1] "2018-12-18 00:00:00 TSLA 200 @ 22.4686660766602"
## [1] "2018-12-21 00:00:00 TSLA 100 @ 21.318000793457"
## [1] "2018-12-26 00:00:00 TSLA 100 @ 21.7393321990967"
## [1] "2019-01-08 00:00:00 TSLA -200 @ 22.3566665649414"
## [1] "2019-01-24 00:00:00 TSLA 100 @ 19.4340000152588"
## [1] "2019-02-06 00:00:00 TSLA -100 @ 21.1480007171631"
## [1] "2019-03-05 00:00:00 TSLA 100 @ 18.4360008239746"
## [1] "2019-03-15 00:00:00 TSLA -100 @ 18.3619995117188"
## [1] "2019-04-26 00:00:00 TSLA 100 @ 15.6759996414185"
## [1] "2019-05-20 00:00:00 TSLA 100 @ 13.6906671524048"
## [1] "2019-06-07 00:00:00 TSLA -200 @ 13.6333332061768"
## [1] "2019-06-18 00:00:00 TSLA -100 @ 14.9826669692993"
## [1] "2019-07-05 00:00:00 TSLA -100 @ 15.539999961853"
## [1] "2019-07-26 00:00:00 TSLA 200 @ 15.2026672363281"
## [1] "2019-08-26 00:00:00 TSLA 100 @ 14.333330154419"
## [1] "2019-09-03 00:00:00 TSLA -100 @ 15.0006666183472"
## [1] "2019-09-11 00:00:00 TSLA -100 @ 16.4733333587646"
## [1] "2019-09-25 00:00:00 TSLA 100 @ 15.2466669082642"
## [1] "2019-10-15 00:00:00 TSLA -100 @ 17.1926670074463"
## [1] "2019-10-25 00:00:00 TSLA -100 @ 21.8753337860107"
## [1] "2019-11-25 00:00:00 TSLA 200 @ 22.4226665496826"
## [1] "2019-12-17 00:00:00 TSLA -100 @ 25.2660007476807"
## [1] "2020-01-09 00:00:00 TSLA -100 @ 32.0893325805664"
## [1] "2020-02-27 00:00:00 TSLA 200 @ 45.2666664123535"
## [1] "2020-03-10 00:00:00 TSLA 100 @ 43.0219993591309"
## [1] "2020-03-17 00:00:00 TSLA 100 @ 28.6800003051758"
## [1] "2020-04-01 00:00:00 TSLA -200 @ 32.1040000915527"
## [1] "2020-04-14 00:00:00 TSLA -100 @ 47.326000213623"
## [1] "2020-05-04 00:00:00 TSLA 100 @ 50.7459983825684"
```

```
## [1] "2020-06-02 00:00:00 TSLA -100 @ 58.7706680297852"
## [1] "2020-06-09 00:00:00 TSLA -100 @ 62.7113342285156"
## [1] "2020-06-26 00:00:00 TSLA 200 @ 63.982666015625"
## [1] "2020-07-01 00:00:00 TSLA -100 @ 74.6419982910156"
## [1] "2020-07-27 00:00:00 TSLA 100 @ 102.639999389648"
## [1] "2020-08-12 00:00:00 TSLA 100 @ 103.650665283203"
## [1] "2020-08-13 00:00:00 TSLA -100 @ 108.066665649414"
## [1] "2020-08-17 00:00:00 TSLA -100 @ 122.375999450684"
## [1] "2020-09-01 00:00:00 TSLA -100 @ 158.350006103516"
## [1] "2020-09-08 00:00:00 TSLA 200 @ 110.069999694824"
## [1] "2020-10-29 00:00:00 TSLA 100 @ 136.943328857422"
## [1] "2020-11-02 00:00:00 TSLA 100 @ 133.503326416016"
## [1] "2020-11-04 00:00:00 TSLA -200 @ 140.32666015625"
## [1] "2020-11-19 00:00:00 TSLA -100 @ 166.42333984375"
## [1] "2020-12-21 00:00:00 TSLA -100 @ 216.619995117188"
## [1] "2021-01-29 00:00:00 TSLA 200 @ 264.510009765625"
## [1] "2021-02-11 00:00:00 TSLA 100 @ 270.553344726562"
## [1] "2021-02-23 00:00:00 TSLA 100 @ 232.946670532227"
## [1] "2021-03-11 00:00:00 TSLA -200 @ 233.199996948242"
## [1] "2021-04-14 00:00:00 TSLA -100 @ 244.07666015625"
## [1] "2021-04-28 00:00:00 TSLA 100 @ 231.466659545898"
## [1] "2021-05-11 00:00:00 TSLA 100 @ 205.733337402344"
## [1] "2021-05-27 00:00:00 TSLA -100 @ 210.283340454102"
## [1] "2021-06-24 00:00:00 TSLA -100 @ 226.606674194336"
## [1] "2021-07-08 00:00:00 TSLA 100 @ 217.603332519531"
## [1] "2021-08-02 00:00:00 TSLA -100 @ 236.556671142578"
## [1] "2021-08-17 00:00:00 TSLA 100 @ 221.903335571289"
## [1] "2021-09-27 00:00:00 TSLA -100 @ 263.786682128906"
## [1] "2021-10-18 00:00:00 TSLA -100 @ 290.036682128906"
## [1] "2021-11-10 00:00:00 TSLA 200 @ 355.983337402344"
## [1] "2021-12-14 00:00:00 TSLA 100 @ 319.503326416016"
## [1] "2021-12-27 00:00:00 TSLA -100 @ 364.646667480469"
## [1] "2022-01-04 00:00:00 TSLA -100 @ 383.196655273438"
```

```
## [1] "2022-01-07 00:00:00 TSLA 100 @ 342.320007324219"
## [1] "2022-01-24 00:00:00 TSLA 100 @ 310"
## [1] "2022-01-28 00:00:00 TSLA 100 @ 282.116668701172"
## [1] "2022-02-11 00:00:00 TSLA -200 @ 286.666656494141"
## [1] "2022-02-23 00:00:00 TSLA 100 @ 254.679992675781"
## [1] "2022-03-02 00:00:00 TSLA -100 @ 293.296661376953"
## [1] "2022-03-21 00:00:00 TSLA -100 @ 307.053344726562"
## [1] "2022-03-29 00:00:00 TSLA -100 @ 366.523345947266"
## [1] "2022-04-12 00:00:00 TSLA 200 @ 328.983337402344"
## [1] "2022-04-27 00:00:00 TSLA 100 @ 293.836669921875"
## [1] "2022-05-10 00:00:00 TSLA 100 @ 266.679992675781"
## [1] "2022-06-01 00:00:00 TSLA -200 @ 246.789993286133"
## [1] "2022-07-22 00:00:00 TSLA -100 @ 272.243347167969"
## [1] "2022-07-29 00:00:00 TSLA -100 @ 297.149993896484"
## [1] "2022-08-22 00:00:00 TSLA 200 @ 289.913330078125"
## [1] "2022-08-31 00:00:00 TSLA 100 @ 275.609985351562"
## [1] "2022-09-06 00:00:00 TSLA 100 @ 274.420013427734"
## [1] "2022-09-12 00:00:00 TSLA -200 @ 304.420013427734"
## [1] "2022-10-04 00:00:00 TSLA 100 @ 249.440002441406"
## [1] "2022-10-10 00:00:00 TSLA 100 @ 222.960006713867"
## [1] "2022-10-27 00:00:00 TSLA -200 @ 225.089996337891"
## [1] "2022-11-08 00:00:00 TSLA 100 @ 191.300003051758"
## [1] "2022-12-01 00:00:00 TSLA -100 @ 194.699996948242"
## [1] "2022-12-14 00:00:00 TSLA 100 @ 156.800003051758"
## [1] "2022-12-19 00:00:00 TSLA 100 @ 149.869995117188"
## [1] "2023-01-18 00:00:00 TSLA -200 @ 128.779998779297"
## [1] "2023-01-24 00:00:00 TSLA -100 @ 143.889999389648"
## [1] "2023-02-02 00:00:00 TSLA -100 @ 188.270004272461"
## [1] "2023-02-27 00:00:00 TSLA 200 @ 207.630004882812"
## [1] "2023-03-08 00:00:00 TSLA 100 @ 182"
## [1] "2023-03-22 00:00:00 TSLA -100 @ 191.149993896484"
## [1] "2023-04-03 00:00:00 TSLA -100 @ 194.770004272461"
## [1] "2023-04-06 00:00:00 TSLA 100 @ 185.059997558594"
```

```
## [1] "2023-04-21 00:00:00 TSLA 100 @ 165.080001831055"
## [1] "2023-04-27 00:00:00 TSLA 100 @ 160.190002441406"
## [1] "2023-05-09 00:00:00 TSLA -200 @ 169.149993896484"
## [1] "2023-05-19 00:00:00 TSLA -100 @ 180.139999389648"
## [1] "2023-05-30 00:00:00 TSLA -100 @ 201.160003662109"
## [1] "2023-07-21 00:00:00 TSLA 200 @ 260.019989013672"
## [1] "2023-08-17 00:00:00 TSLA 100 @ 219.220001220703"
## [1] "2023-08-29 00:00:00 TSLA -100 @ 257.179992675781"
## [1] "2023-09-12 00:00:00 TSLA -100 @ 267.480010986328"
## [1] "2023-09-22 00:00:00 TSLA 100 @ 244.880004882812"
## [1] "2023-10-20 00:00:00 TSLA 100 @ 211.990005493164"
## [1] "2023-11-09 00:00:00 TSLA -100 @ 209.979995727539"
## [1] "2023-11-16 00:00:00 TSLA -100 @ 233.589996337891"
## [1] "2023-12-12 00:00:00 TSLA 100 @ 237.009994506836"
## [1] "2023-12-15 00:00:00 TSLA -100 @ 253.5"
## [1] "2024-01-03 00:00:00 TSLA 100 @ 238.449996948242"
## [1] "2024-01-12 00:00:00 TSLA 100 @ 218.889999389648"
## [1] "2024-01-26 00:00:00 TSLA 100 @ 183.25"
## [1] "2024-02-16 00:00:00 TSLA -200 @ 199.949996948242"
## [1] "2024-03-07 00:00:00 TSLA 100 @ 178.649993896484"
## [1] "2024-03-11 00:00:00 TSLA 100 @ 177.770004272461"
## [1] "2024-03-27 00:00:00 TSLA -200 @ 179.830001831055"
## [1] "2024-04-16 00:00:00 TSLA 100 @ 157.110000610352"
## [1] "2024-04-25 00:00:00 TSLA -100 @ 170.179992675781"
## [1] "2024-04-30 00:00:00 TSLA -100 @ 183.279998779297"
## [1] "2024-05-13 00:00:00 TSLA 100 @ 171.889999389648"
## [1] "2024-06-12 00:00:00 TSLA 100 @ 177.289993286133"
## [1] "2024-06-13 00:00:00 TSLA -100 @ 182.470001220703"
## [1] "2024-06-18 00:00:00 TSLA -100 @ 184.860000610352"
## [1] "2024-06-27 00:00:00 TSLA -100 @ 197.419998168945"
## [1] "2024-07-25 00:00:00 TSLA 200 @ 220.25"
## [1] "2024-08-05 00:00:00 TSLA 100 @ 198.880004882812"
## [1] "2024-08-19 00:00:00 TSLA -100 @ 222.720001220703"
```

```
## [1] "2024-09-06 00:00:00 TSLA -100 @ 210.729995727539"
## [1] "2024-09-09 00:00:00 TSLA 100 @ 216.270004272461"
## [1] "2024-09-13 00:00:00 TSLA -100 @ 230.289993286133"
## [1] "2024-09-20 00:00:00 TSLA -100 @ 238.25"
## [1] "2024-10-04 00:00:00 TSLA 200 @ 250.080001831055"
## [1] "2024-10-14 00:00:00 TSLA 100 @ 219.160003662109"
## [1] "2024-10-25 00:00:00 TSLA -100 @ 269.190002441406"
## [1] "2024-10-28 00:00:00 TSLA -100 @ 262.510009765625"
## [1] "2024-11-07 00:00:00 TSLA -100 @ 296.910003662109"
## [1] "2025-01-02 00:00:00 TSLA 200 @ 379.279998779297"
## [1] "2025-02-10 00:00:00 TSLA 100 @ 350.730010986328"
## [1] "2025-02-26 00:00:00 TSLA 100 @ 290.799987792969"
## [1] "2025-03-25 00:00:00 TSLA -200 @ 288.140014648438"
## [1] "2025-04-28 00:00:00 TSLA -100 @ 285.880004882812"
## [1] "2025-05-13 00:00:00 TSLA -100 @ 334.070007324219"

end_t <- Sys.time()
print(" 策略执行时间:")

## [1] "策略执行时间:"

print(end_t - start_t)

## Time difference of 5.891073 secs

# 更新投资组合价值并计时
start_t <- Sys.time()
updatePortf(Portfolio = 'bbands', Dates = paste(':::', as.Date(Sys.time()), sep = ''))

## [1] "bbands"

end_t <- Sys.time()
print(" 更新投资组合价值时间:")

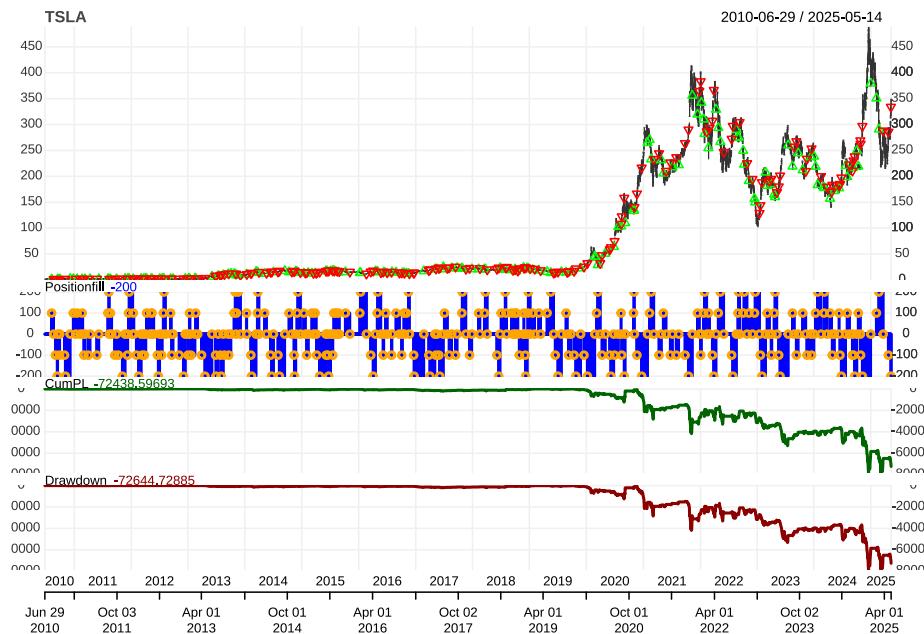
## [1] "更新投资组合价值时间:"
```

```
print(end_t - start_t)
```

```
## Time difference of 0.03077912 secs
```

```
# 绘制持仓和价格图表
```

```
chart.Posn(Portfolio = 'bbands', Symbol = stock.str)
```



```
# 绘制布林带 (注释中的代码修正了语法错误)
```

```
# plot(add_BBands(on = 1, sd = SD, n = N))
```

以上代码演示了 quanstrat 和 TTR 包的配合使用，通过结合 TTR 包的技术指标和 quanstrat 的交易系统框架，实现了一个完整的股票交易策略回测。你可以根据需要调整参数和指标，开发更复杂的交易策略。