

国泰安量化投资研究平台

数据指标 API 说明文档

(MATLAB 版)



国泰安信息技术有限公司

GTA Information Technology Co., Ltd.

目 录

1. 基础数据访问 API	11
1.1. getLowFrequencyData-低频数据提取 API	11
1.2. getFinancialData-财务数据提取 API	14
1.3. getDailyTradingData-交易数据统一提取 API	16
1.4. getQuantitativeFactor-量化因子提取 API	19
1.5. getOrgIDByStkcd-公司代码提取 API	22
1.6. getStkcdByOrgID-交易代码提取 API	25
1.7. getFieldNamedByChineseName-物理字段查询 API	27
1.8. getChineseNameByFieldName-中文名称查询 API	28
1.9. getHighFrequencySnapShotData-分时高频数据提取 API	29
1.10. getHighFrequencyTickData-分笔高频数据提取 API	32
2. 技术指标 API	38
2.1. 底层函数	38
2.1.1. AVEDEV-平均绝对离差	38
2.1.2. DMA-动态移动平均值	39
2.1.3. EMA-指数平滑移动平均值	40
2.1.4. HHV-序列最大值	42
2.1.5. LLV-序列最小值	43
2.1.6. MA -简单移动平均值	43
2.1.7. SMA -指数移动平均值	45

2.2.	趋向指标.....	46
2.2.1.	ACD-升降线指标.....	46
2.2.2.	ADX-动向平均数指标.....	48
2.2.3.	BBI -多空指数	49
2.2.4.	CHO-佳庆指标.....	51
2.2.5.	CYE-市场趋势指标.....	52
2.2.6.	DDI-方向标准离差指数.....	54
2.2.7.	DFMA -平均线差.....	56
2.2.8.	DMI-趋向指标.....	57
2.2.9.	EMV-简单波动指标	61
2.2.10.	GDX-鬼道线指标.....	63
2.2.11.	JLHB-绝路航标.....	65
2.2.12.	JS-加速线指标	67
2.2.13.	MACD -平滑异同平均指标.....	68
2.2.14.	MDI-下降动向指标.....	70
2.2.15.	MTM-动力指标.....	72
2.2.16.	OSC -变动速率线.....	73
2.2.17.	PBX-瀑布线指标.....	74
2.2.18.	PDI-上升动向指标	75
2.2.19.	QACD-快速异同平均指标	77
2.2.20.	QR-强弱值指标	78
2.2.21.	TRIX-三重指数平滑平均线.....	80

2.2.22.	UOS-终极指标.....	81
2.2.23.	VMA-变异平均线指标.....	83
2.3.	反趋向指标.....	85
2.3.1.	ACCER-幅度涨速指标	85
2.3.2.	ADTM-动态买卖气指标	86
2.3.3.	BB-布林极限指标.....	89
2.3.4.	BIAS-乖离率指标	90
2.3.5.	CCI-顺势指标.....	92
2.3.6.	CYF-市场能量指标.....	94
2.3.7.	DBCD-异同离差乖离率指标.....	95
2.3.8.	DKX-多空线指标.....	97
2.3.9.	DPO-区间震荡线指标	99
2.3.10.	FSL-分水岭指标.....	100
2.3.11.	KDJ-随机指标.....	102
2.3.12.	LWR-L 威廉指标	105
2.3.13.	ROC-变动速率指标.....	107
2.3.14.	RSI-相对强弱指标.....	108
2.3.15.	SKDJ-慢速随机指标.....	109
2.3.16.	SI-摆动指标.....	112
2.3.17.	SRDM-动向速度比率指标	114
2.3.18.	UDL-引力线指标.....	116
2.3.19.	WR-威廉指标	118

2.3.20.	WIDTH-布林极限宽度指标.....	119
2.4.	量价指标.....	121
2.4.1.	AMV-成本线均价指标.....	121
2.4.2.	ARBR-人气意愿指标.....	123
2.4.3.	CR-能量指标.....	125
2.4.4.	CYR-市场强弱指标	127
2.4.5.	MASS-梅斯线指标.....	129
2.4.6.	MFI-资金流量指标.....	130
2.4.7.	OBV-能量潮指标.....	133
2.4.8.	PCNT-幅度比指标	135
2.4.9.	PSY-心理线指标.....	136
2.4.10.	PVT-价量趋向指标.....	138
2.4.11.	TAPI-加权指数成交值指标.....	139
2.4.12.	VR-容量比率指标.....	141
2.4.13.	WAD-威廉多空力度线指标	143
2.4.14.	WVAD-威廉变异离散量指标.....	145
2.5.	压力支撑指标.....	148
2.5.1.	BBIBOLL-多空布林线.....	148
2.5.2.	BOLL-布林线指标	149
2.5.3.	ENE-轨道线指标	151
2.5.4.	ENV-Envelops 指标.....	153
2.5.5.	MIKE-麦克指标	155

2.5.6.	XS-薛斯通道指标	158
2.6.	大盘指标	161
2.6.1.	ABI-绝对广量指标	161
2.6.2.	ADL-腾落指数	162
2.6.3.	ADR-涨跌比率指标	164
2.6.4.	ARMS-阿姆氏指标	165
2.6.5.	BTI-Breadth Trust 指标	167
2.6.6.	CHAIKIN-佳庆线指标	169
2.6.7.	MCO-麦克连指标	170
2.6.8.	OBOS-超买超卖指标	172
2.6.9.	STIX-指数平滑广量指标	174
2.7.	摆动指标	176
2.7.1.	AD-累积分布指标	176
2.7.2.	ATR-真实波幅指标	177
2.7.3.	MI-动量指标	179
2.7.4.	MICD-异同离差动力指数	181
2.7.5.	RC-变化率指标	182
2.7.6.	RCCD-异同离差变化率指数	184
2.7.7.	SRMI-动量修正指标	185
2.8.	统计指标	187
2.8.1.	DDBS-大单笔数指标	187
2.8.2.	DDTQ-大单提取	188

2.8.3.	DPTB-大盘同步指标.....	190
2.8.4.	DSBC-大手笔差.....	192
2.8.5.	JDAMOUNT-阶段内总成交额.....	193
2.8.6.	JDHSL-阶段内换手率.....	195
2.8.7.	JDVOL-阶段内总成交量.....	196
2.8.8.	JDZF-阶段内涨幅.....	198
2.8.9.	NWPB-内外盘比.....	199
2.8.10.	NWPC-内外盘差.....	201
3.	选股模型 API.....	202
3.1.	指标条件选股模型.....	202
3.1.1.	BIASBUY-乖离率买入.....	202
3.1.2.	BIASSELL-乖离率卖出.....	204
3.1.3.	BOLLBUY-布林带买入.....	206
3.1.4.	BOLLSSELL-布林带卖出.....	208
3.1.5.	KDBUY –KD 随机指标买入.....	210
3.1.6.	KDJBUTY –KDJ 随机指标买入.....	212
3.1.7.	KDJSELL-KDJ 随机指标卖出.....	215
3.1.8.	KDSELL-KD 随机指标卖出.....	217
3.1.9.	MABUY-均线指标买入.....	220
3.1.10.	MACDBUY-MACD 指标买入.....	222
3.1.11.	MACDSELL-MACD 指标卖出.....	224
3.1.12.	MASELL-均线指标卖出.....	226

3.1.13.	MTMBUY-动力指标买入	228
3.1.14.	MTMSELL-动力指标卖出	229
3.1.15.	RSIBUY-相对强弱指标买入	231
3.1.16.	RSISELL-相对强弱指标卖出	233
3.1.17.	WRBUY-威廉指标买入.....	235
3.1.18.	WRSELL-威廉指标卖出.....	238
3.2.	走势特征选股模型.....	240
3.2.1.	BREAKPLAT-突破长期盘整	240
3.2.2.	MDFALL-M 周期下跌	242
3.2.3.	MDFALLMORE-M 周期下跌多于上涨.....	243
3.2.4.	MDGMORE-M 周期阴线多于阳线	245
3.2.5.	MDGREEN-M 周期收阴线.....	247
3.2.6.	MDHIGHPRICE-M 周期创新高.....	249
3.2.7.	MDLOW-M 周期创新低.....	251
3.2.8.	MDRED-M 周期收阳线	253
3.2.9.	MDRISE-M 周期上涨.....	255
3.2.10.	MDRISEMORE-M 周期上涨多于下跌	257
3.2.11.	MDRMORE-M 周期阳线多于阴线.....	259
3.2.12.	PLATFORM-平台整理.....	261
3.2.13.	VOLUMEUP-突然放量.....	262
3.3.	K 线形态选股模型	264
3.3.1.	BALDGREEN-光脚阴线	264

3.3.2.	BALDRED-光头阳线.....	266
3.3.3.	COUNTERATTACK-好友反攻.....	268
3.3.4.	CROSSHEADFOOT-穿头破脚.....	270
3.3.5.	CROSSSTAR-十字星.....	273
3.3.6.	DARKCLOUD-乌云盖顶.....	275
3.3.7.	DEADCROSS-垂死十字星.....	277
3.3.8.	DOUBLECROWS-双飞乌鸦.....	279
3.3.9.	DOWNSHADOW-长下影.....	282
3.3.10.	DOWNSTEPS-下跌三部曲.....	284
3.3.11.	EVENINGCROSS-黄昏十字星.....	287
3.3.12.	EVENINGSTAR-黄昏之星.....	289
3.3.13.	LONGCROSS-长十字.....	292
3.3.14.	FLATBOTTOM-平底.....	294
3.3.15.	FLATCEILING-平顶.....	296
3.3.16.	GREENCANDLE-大阴烛.....	297
3.3.17.	HAMMER-锤头.....	300
3.3.18.	MORNINGCROSS-早晨十字星.....	302
3.3.19.	MORNINGDAWN-曙光初现.....	304
3.3.20.	MORNINGSTAR-早晨之星.....	307
3.3.21.	PREGCROSS-十字胎.....	309
3.3.22.	PREGLINE-孕育线.....	311
3.3.23.	PREGNANT-身怀六甲.....	314

3.3.24.	REDCANDLE-太阳烛	316
3.3.25.	SEPARATE-分离	319
3.3.26.	SHOTSTAR-射击之星	321
3.3.27.	SPRUNGROOF-吊顶	323
3.3.28.	THREECROWS-三只乌鸦	326
3.3.29.	THREEKNIGHTS-三个白武士	328
3.3.30.	TURNHAMMER-倒转锤头	330
3.3.31.	UPSHADOW-长上影	333
3.3.32.	UPSTEPS-上升三部曲	335
3.3.33.	VACANT-跳空缺口	338

1. 基础数据访问 API

1.1. getLowFrequencyData-低频数据提取 API

函数解释：

1. **用途：**根据输入的查询表名、字段、代码范围、日期范围等查询出具体的数据矩阵返回。

本 API 适用的范围是量化投资平台基本面数据库的所有数据表,包括宏观经济、股票、上市公司、基金、债券、股指期货、商品期货等数据,将返回 k 个 n*m 的矩阵,其中 k 为查询的指标字段个数,n 为代码个数, m 为日期天数。

本 API 中条件字段输入中的各字段条件之间只能是'And'的关系,不支持'or'的关系,例如,想查出 2008、2010 年的交易数据,在条件字段设置两次:

'2008-01-01'<= trdd && trdd <= '2008-12-31','2010-01-01'<= trdd && trdd <= '2010-12-31'将返回空值。

2. 函数详细结构:

[results, rowIndex, columnIndex] = getLowFrequencyData

(tableName,codeField,dateField,codeList, fieldList,conditionFields, lowerBound, upperBound)

3. 输入参数说明:

tableName 数据表名,字符串格式,例如:'Stk_dlyr';

codeField	代码字段名称，结果矩阵的行索引字段，根据这个字段的数值进行行的划分。字符串格式，例如：'Stkcd'；
codeList	代码列表 可以输入交易代码、公司代码列表, n*1 或者 1*n 的 cell 数组，格式例如{'000001';'2';'4356'}，当 codeField 输入 stkcd 时，在 codeField 第一个单元格输入'all' \ 'alla' \ 'allb' \ ' sha' \ ' sza' \ ' shb' \ ' szb' \ ' hs300' ，系统将根据这些字母自动生成返回这些板块的交易代码对应的信息，all 表示沪深所有 AB 股股票，sha 表示所有上海 A 股，shb 表示所有上海 b 股，sza 表示所有深圳 A 股，szb 表示所有深圳 b 股，alla 表示所有 A 股，allb 表示所有 b 股，hs300 表示最新的沪深 300 指数成分股，需要更多的指数成分股查询，需要同时结合其他的 API 使用；
fieldList	查询字段列表，n*1 或者 1*n 的 cell 数组，例如： {'clsprc';'dretwd'}； 详细说明请参阅《国泰安量化投资研究平台数据库说明书》；
dateField	日期字段名称,结果矩阵的列索引字段，根据这个字段的数据进行列的分割，字符串格式，例如：'Trddt'；
conditionFields	条件字段，n*1 或者 1*n 的 cell 数组，例如：{'Trddt';'dretwd'}，当选 10-30 时，下限日期为交易所开始交易最早日期 '1990-12-16'；
lowerBound	条件下限值，n*1 或者 1*n 的 cell 数组，例如{'2012-01-01'; []}, {'2012-01-01'; 12.45}，维数必须与 conditionFields cell 一致；
upperBound	条件上限值，n*1 或者 1*n 的 cell 数组，例如{'2012-01-01'; []},

{'2012-01-01'; 12.45} , 维数必须与上限 cell 一致。

4. 公式:

无

5. 中文含义:

本 API 通过输入表名、代码、时间、查询字段, 以及其他扩展条件查询出数据并以矩阵形式返回。

本 API 适用的范围是量化投资平台基本面数据库的所有数据表。

6. 输出参数说明:

results 一个结构体 cell, 包含 k 个矩阵, k = 查询的指标字段个数, 当字段为数值型时, 返回 double 的矩阵, 否则返回 cell 矩阵;

columnIndex 列索引, cell 数组, 包含数值结果矩阵的列索引;

rowIndex 行索引, cell 数组, 包含数值结果矩阵的行索引。

7. 示例:

```
>>[results,rowIndex,columnIndex]=getLowFrequencyData('stk_dlyr','stkcd','Trddt',{'szb'}, {'clsprc';'Dretwd';'Capchgdt'},{'Trddt'}, {'2008-01-01'}, {'2012-01-01'});
```

```
>>[results,rowIndex,columnIndex]=getLowFrequencyData('STK_Mnthlyr','stkcd','Trdmnt',{'000001';'600000'},{'Mclsprc';'Mretwd'},{'Trdmnt'}, {'2008-01-01'}, {'2012-01-01'});
```

```
>>[results,rowIndex,columnIndex]=getLowFrequencyData('CO_Combas','Orgid','Accper',{'1314';'2012'},{'Totast';'Totl';'Totse'},{'Accper';'Totast'},{'2008-01-01';'2012-01-01'});
```

-01';10000000)},{'':[]});

1.2. getFinancialData-财务数据提取 API

函数解释：

1. **用途：**根据输入的查询表名、字段、代码范围、日期范围、报告期条件查询出具体的财务数据矩阵返回。

2. **函数详细结构：**

[results, rowIndex, columnIndex] = getFinancialData

(tableName,codeType,codeList,fieldList,reportPeriod,reportType,

beginDate, endDate)

3. **输入参数说明：**

tableName 数据表名,字符串格式，例如：'Stk_dlyr'；

codeType 代码字段名称，结果矩阵的行索引字段，根据这个字段的数值进行行的划分。字符串格式，例如：'Stkcd'；

codeList 代码列表 可以输入交易代码、公司代码列表,n*1 或者 1*n 的 cell 数组，格式例如{'000001';'2';'4356'}，当 codeType 为交易代码 'Stkcd' 时可以在 codeList 第一个单元格输入'all' \ ' sha' \ ' sza' \ ' shb' \ ' szb' \ ' hs300'，系统将根据这些字母自动生成对应的代码列表返回，all 表示沪深所有 AB 股股票，sha 表示所有上海 A 股 shb 表示所有上海 b 股 sza 表示所有深圳 A 股，szb 表示所有深圳 b 股 hs300 表示最新的沪深 300 指数成分股，

需要 更多的指数成分股查询，需要同时结合其他的 API 使用。

fieldList 查询字段列表，n*1 或者 1*n 的 cell 数组，例如：

{'Totast';'Totl';'Totse'}；

详细说明请参阅《国泰安量化投资研究平台数据库说明书》；

reportPeriod 0：表示所有报告期数据；

1：表示第一季报数据；

2：表示中报数据；

3：表示第三季报数据；

4：表示年报数据；

reportType 报表类别：

0：表示无报表类别，

1：表示合并报表，

2：表示母公司报表；

beginDate 开始日期,字符串格式，例如：'2012-01-01'；

endDate 结束日期 字符串格式，例如：'2012-01-01'。

4. 公式:

无

5. 中文含义:

财务数据的查询一般通过代码和会计期间、报告类型的标识组合查询，本 API 通过输入表名、代码、时间、报表类型这几个常用的参数查询出数据并以矩阵形式返回。

本 API 适用的范围是具有统一报告周期的财务数据，适用范围是三大财

务报表及财务指标表数据。

6. 输出参数说明:

Results 一个结构体 cell，包含 k 个矩阵，k = 查询的指标字段个数，

当字段为数值型时，返回 double 的矩阵，否则返回 cell 矩阵；

columnIndex 列索引, cell 数组, 包含数值结果矩阵的列索引;

rowIndex 行索引, cell 数组, 包含数值结果矩阵的行索引;

7. 示例:

1,单独查某个公司财务数据：

```
> [results,rowIndex,columnIndex]=getFinancialData('CO_Combas','Orgid',
{'1314';'2012'}, {'Totast';'Totl';'Totse'},1,1, '2008-01-01', '2012-01-01');
```

2, 根据交易代码查公司财务数据:

```
>>orgIDList = getOrgIDByStkcd(1,{ 'sza' }, 1);

>>[results,rowIndex,columnIndex]=getFinancialData('CO_Combas','Orgid',
    orgIDList(:,1), { 'Totast';'Totl';'Totse'},1,1, '2008-01-01', '2012-01-01') ;
```

1.3. getDailyTradingData-交易数据统一提取 API

函数解释：

1. **用途:** 根据输入的代码类别,代码列表、字段名称、开始日期、结束日期统一提取股票、股指期货、商品期货的日频交易数据,主要用于跨市场策略的回验。

2. 函数详细结构:

```
[results, rowIndex, columnIndex] = getDailyTradingData (codeType,codeList,  
fieldList,beginDate, endDate)
```


3. 输入参数说明:

codeType - 代码类型，整数，用以标识第二个参数 codeList 的类型：

0 表示传入的是公司代码，当公司代码中同时对应 AB 股时，会同时返回 AB 股数据，

1 表示传入的是交易代码；

11 表示传入的是公司 ID，只返回公司 ID 中包含的 A 股记录，

12 表示传入的是公司 ID，只返回公司 ID 中包含的 A 股记录，

13 表示传入的是公司 ID，当公司 ID 同时对应 AB 股的时候，只返回公司 A 股记录；

14 表示传入的是公司 ID，当公司 ID 同时对应 AB 股的时候，只返回公司 B 股记录；

codeList - 代码列表 可以输入交易代码、公司代码列表,1*n 或 n*1 的 cell 矩阵，格式例如{'000001';'2';'4356'}；

fieldList - 查询字段列表 ,cell 数组，例如：{'Totast';'Totl';'Totse'}

OrgID	:	公司编码
TradingCode	:	交易代码
Trddt	:	交易日期
Opnprc	:	日开盘价
Hiprc	:	日最高价
Loprc	:	日最低价
Clsprc	:	日收盘价
tradingVolume	:	日成交量

TradingValue	:	日成交金额
Opint	:	日持仓量
Dretwd	:	日回报率
Ystprc	:	昨结算价
Ceilprc	:	日涨停价
Floorprc	:	日跌停价

beginDate - 开始日期,字符串格式，例如：'2012-01-01'；

endDate - 结束日期 字符串格式，例如：'2012-01-01'；

4. 公式:

无

5. 中文含义:

在日频跨市场策略回验中，需要同时提取股票、股指期货、商品期货的交易数据，并以矩阵形式返回。

本 API 针对跨市场策略回验的需求，提供了 14 个字段的通用提取接口，支持所有 A、B 股股票、股指期货、商品期货的日频交易数据综合提取，可以通过公司代码和交易代码两种方式查询。

6. 输出参数说明:

Results 一个结构体 cell，包含 k 个矩阵，k = 查询的指标字段个数，返回 double 的矩阵；

columnIndex 列索引，cell 数组，包含数值结果矩阵的列索引；

rowIndex 行索引，cell 数组，包含数值结果矩阵的行索引；

7. 示例:

```

>>[results,          rowIndex,          columnIndex]          =

      getDailyTradingData(1,{ '000001';'IF1207';'AL1207'},

      {'clsprc';'dretwd'},'2012-06-01', '2012-06-30') ;

>>orgIDList = getOrgIDByStkcd(1,{'sza'}, 1);

>>[results,  rowIndex,  columnIndex]  =  getDailyTradingData(0,orgIDList(:,1),

      {'clsprc';'dretwd'}, '2012-06-01', '2012-06-30');

>>[results, rowIndex, columnIndex] =getDailyTradingData(0, {'193'; '2'; '3000027';

      '3000026';   '15000151';   '17000151'}   ,   {'clsprc';'dretwd'},'2012-06-01',

      '2012-06-30');

>>[results,          rowIndex,          columnIndex]          =

      getHighFrequencySnapShotData(5,{ 'A1201';'A1203'},

      {'cp';'op'},'2011-12-01', '2011-12-01','080000','130501',15);

```

1.4. getQuantitativeFactor-量化因子提取 API

函数解释：

1. **用途：**根据输入的因子名、因子数值类型、代码范围、日期范围等查询出具体的量化因子数据矩阵返回。

2. 函数详细结构:

```
[results, rowIndex, columnIndex] =
```

getQuantitativeFactor(factorName, factorType, codeType, codeList, beginDate, endDate)

3. 输入参数说明:

factorName- 因子名称,字符串格式, 例如: 'QF_MACD';

factorType- 因子类型,整数, 0 表示原始数值, 1 表示正态化数值;

codeType - 代码类型, 整数, 0 表示公司代码, 1 表示交易代码;

codeList 代码列表 可以输入交易代码、公司代码列表,n*1 或 1*n 的 cell 矩阵,

格式例如{'000001';'2';'4356'}, 当 codeType 为 1 时, 可以在

codeList 第一个单元格输入'all'\ ' sha' \ ' sza' \ '

\ ' hs300' , 系统将根据这些字母自动生成对应的代码列表返回,

all 表示沪深所有 AB 股股票, sha 表示所有上海 A 股, sza 表示

所有深圳 A 股 hs300 表示最新的沪深 300 指数成分股, 需要 更

多的指数成分股查询, 需要同时结合其他的 API 使用;

beginDate 开始日期,字符串格式, 例如: '2012-01-01';

endDate 结束日期 字符串格式, 例如: '2012-01-01'.

4. 公式:

无

5. 中文含义:

本 API 根据输入的因子名、因子数值类型、代码范围、日期范围等查询出具体的数据矩阵返回, 适用的范围是国泰安量化因子仓库的所有因子(目前仅针对 A 股), 将返回 n*m 的矩阵, 其中 n 为代码个数, m 为日期天数。

6. 输出参数说明:

Results 一个结构体 cell , 包含 k 个矩阵 , k = 查询的指标字段个数 ,
当字段为数值型时 , 返回 double 的矩阵 , 否则返回 cell 矩阵 ;

columnIndex 列索引 , cell 数组 , 包含数值结果矩阵的列索引 ;

rowIndex 行索引 , cell 数组 , 包含数值结果矩阵的行索引。

7. 示例:

1 , 输入公司代码获取因子示例 :

```
>> [results, columnIndex, rowIndex] =  
getQuantitativeFactor('QF_PS',0,0, {'1314';'2012'}, '2008-01-01',  
'2012-01-01');
```

2 , 输入交易代码获取因子示例 :

```
>>[results, columnIndex, rowIndex] = getQuantitativeFactor  
('QF_Momentum1M', 1, 1, {'000001';'600000'}, '2008-01-01',  
'2012-01-01');
```

3 , 批量输入交易代码获取因子示例 :

```
>>[results, columnIndex, rowIndex] = getQuantitativeFactor  
('QF_Momentum1M', 1, 1, {'sza'}, '2008-01-01', '2012-01-01');
```

1.5. getOrgIDByStkcd-公司代码提取 API

函数解释：

1. **用途：**根据输入的交易代码、指数代码、期货代码等列表查找对应的公司及合约编码 ID 信息矩阵。

2. **函数详细结构：**

orgIDList = getOrgIDByStkcd(codeType,stkcdList, querySign)

3. **输入参数说明：**

codeType 品种类型：

- 0 表示公司 ID,
- 1 表示股票，
- 2 表示指数，
- 3 表示基金，
- 4 表示股指期货，
- 5 表示商品期货；

stkcdList 交易代码列表,n*1 或者 1*n 的 cell 数组 ,每个单元格是交易代码的字符串，例如 {'000001';'600000'},当 codeType = 1 时，在 stkcdList 第一个单元格输入'all' \ 'allif' \ 'allmf' \ 'alla' \ 'allb' \ ' sha' \ ' sza' \ ' shb' \ ' szb' \ ' hs300' ,系统将根据这些字母自动生成返回这些板块的交易代码对应的信息：

- all 表示中国 6 大交易所所有代码，
- allt 表示 6 大交易所所有可交易的合约，
- allif 表示所有股指期货合约，

allmf 表示所有商品期货合约，

allab 表示所有 AB 股股票，

alla 表示所有 A 股，

allb 表示所有 B 股，

sha 表示所有上海 A 股，

shb 表示所有上海 B 股，

sza 表示所有深圳 A 股，

szb 表示所有深圳 B 股，

hs300 表示最新的沪深 300 指数成分股，

需要更多的指数成分股查询，需要同时结合其他的 API 使用；

querySign 查询标识，整数，

0 标识该交易代码对应的所有公司 ID，

1 标识该交易代码对应的最新公司 ID。

2 表示该公司 ID 对应的最新仍可交易代码

4. 公式:

无

5. 中文含义:

通过输入交易代码列表，以及查询的类别，查找到相应的公司 ID 信息（公司 ID，交易代码,开始日期，结束日期），可以扩展为根据输入的指数代码、基金代码、股指期货代码查询相应的 OrgID。

6. 输出参数说明:

orgIDList 一个 n*6 的 cell 矩阵，第一列是公司 ID，第二列是交易代码，

第三列是品种类型，第四列是交易所，第五列是开始交易日期（格式'2012-01-01'），第六列是结束交易日期（格式'2012-01-01'）。

7. 备注说明:

国泰安公司量化投资数据库（3.0 版）公司代码编码分配规则如下：

- 0~1000000 代码段属于 AB 股上市公司的 ID
- 1000001~2000000 属于指数品种的 ID
- 2000001~3000000 属于基金品种的 ID
- 3000001~4000000 属于股指期货品种的 ID
- 4000001~ 属于商品期货品种的 ID

8. 示例:

提取深圳 A 股公司代码：

```
>> orgIDList = getOrgIDByStkcd(1,{'sza'}, 1)
```

orgIDList =

'6'	'000004'	'A 股'	'深交所主板'	'2005-08-18'	'null'
'11'	'000008'	'A 股'	'深交所主板'	'2005-05-17'	'null'
'18'	'000014'	'A 股'	'深交所主板'	'2002-07-12'	'null'
'28'	'000024'	'A 股'	'深交所主板'	'1993-06-07'	'null'
.....					

1.6. getStkcdByOrgID-交易代码提取 API

函数解释：

1. **用途：**根据输入的公司 ID 列表查找对应的交易代码信息矩阵。

2. **函数详细结构：**

stkcdList = getStkcdByOrgID(orgIDList, querySign)

3. **输入参数说明：**

orgIDList 公司代码列表, n*1 或者 1*n 的 cell 数组，每个单元格是公司 ID 的字符串，例如{'1';'2';'4356'}；

querySign 查询标识，整数，

0 标识所有该公司ID对应的所有交易代码，

1 标识所有该公司ID对应的所有交易代码，

2 标识该公司ID对应的所有A股交易代码，

3 标识该公司ID对应的最新A股代码，

4 标识该公司ID对应的所有B股代码，

5 标识该公司ID对应的最新B股代码。

4. **公式：**

无

5. **中文含义：**

通过输入公司 ID cell 列表，以及查询的类别，输出对应的公司 ID、交易代码、开始日期、结束日期

6. **输出参数说明：**

stkcdList 一个 n*6 的 cell 字符串矩阵，第一列是输入的公司 ID，第二列是交易代码，第三列是证券类型，第四列是交易场所，第五列是开始交易日期（格式'2012-01-01'），第六列是结束交易日期（格式'2012-01-01'），当输入的公司 ID 没有查询到相应的交易代码信息时，返回的结果表中，保留这些公司 ID，但其余列为空。

7. 备注说明:

国泰安公司量化投资数据库（3.0 版）公司代码编码分配规则如下：

- 0~1000000 代码段属于 AB 股上市公司的 ID
- 1000001~2000000 属于指数品种的 ID
- 2000001~3000000 属于基金品种的 ID
- 3000001~4000000 属于股指期货品种的 ID
- 4000001~ 属于商品期货品种的 ID

8. 示例:

```
>>stkcdList = getStkcdByOrgID({'1';'2'},0)

stkcdList =

    '1'    '000001'    'A 股'    '深交所主板'    '1991-04-03'    'null'
    '2'    '000002'    'A 股'    '深交所主板'    '1991-01-29'    'null'
    '2'    '200002'    'B 股'    '深交所主板'    '1993-05-28'    'null'
    .....
```

1.7. getFieldNamedByChineseName- 物理字段查询 API

函数解释：

1. **用途：**根据输入的中文名称查找对应的物理表名、字段名。

2. **函数详细结构：**

```
fieldNameList = getFieldNamedByChineseName(chineseNameList)
```

3. **输入参数说明：**

chineseNameList - 物理字段列表,n*1 或者 1*n 的 cell 字符串矩阵，每个单元格是字符串，例如{'指数回报率';'市盈率'}；

4. **公式：**

无

5. **中文含义：**

通过输入中文字段名称列表，查询输出对应的中文字段名称、物理字段名称、中文表名、物理表名。

6. **输出参数说明：**

fieldNameList - 一个 n*4 的 cell 字符串矩阵 第一列是输入的中文字段名称，第二列是物理字段名称，第三列是中文表名，第四列是物理表名。

7. **示例：**

```
>>fieldNameList = getFieldNamedByChineseName({'指数回报率'})
```

```
fieldNameList =
```

```
'年指数回报率'    'Yretindex'    '债券指数年文件表'    'BND_Tbyindexsh'
```

'日指数回报率'	'Dretindex'	'债券指数日文件表'	'BND_Tbdindexsh'
'月指数回报率'	'mretindex'	'债券指数月文件表'	'BND_Tbmindexsh'
'指数回报率'	'idxrtn'	'指数日行情'	'IDX_trd'

1.8. getChineseNameByFieldName-中文名称查询 API

函数解释：

1. **用途：**根据输入的中文名称查找对应的物理表名、字段名。

2. **函数详细结构：**

```
chineseNameList = getChineseNameByFieldName(fieldNameList)
```

3. **输入参数说明：**

fieldNameList – 物理字段名称列表,n*1 或者 1*n 的 cell 字符串矩阵，每个单元格是物理字段名称，例如{'dretwd';'clsprc'}；

4. **公式：**

无

5. **中文含义：**

通过输入字段物理名称列表查找对应的中文信息，输出对应的中文字段名称、物理字段名称、中文表名、物理表名。

6. **输出参数说明：**

fieldNameList - 一个 n*4 的 cell 字符串矩阵 第一列是输入的中文字段名称，第二列是物理字段名称，第三列是中文表名，第四列是物理表名。

7. **示例：**

```
>>chineseNameList = getChineseNameByFieldName({'stockcode'})
```

```
chineseNameList =
```

```
'证券代码'    'StockCode'    '业绩预告表(季)'    'FIN_F_ForecFin'
```

```
'证券代码'    'StockCode'    '业绩快报表(季)'    'FIN_F_QuiTraFin'
```

```
'证券代码'    'StockCode'    '业绩快报简表'      'FIN_F_QuiTraSimFin'
```

```
'证券代码'    'StockCode'    '业绩预告简表'      'FIN_F_SimForecFin'
```

1.9. getHighFrequencySnapShotData- 分时高频数据提取 API

函数解释：

1. **用途:** 根据输入的代码列表、字段、日期、时间范围等查询出具体的分时高频数据矩阵返回。

2. **函数详细结构:**

```
[results, rowIndex, columnIndex] =
```

```
getHighFrequencySnapShotData(marketType, codeList, fieldList, beginDate,  
endDate, beginTime, endTime, frenquenceType)
```

3. **输入参数说明:**

marketType 市场类型，整数，标识 codeList 参数的类型：

0 表示的代码为公司 ID，

1 表示深圳市场交易代码；

2 表示上海市场交易代码；

- 4 表示中金所交易代码；
- 5 表示大商所交易代码；
- 6 表示郑商所交易代码；
- 7 表示上期所交易代码；
- 11 表示传入的是公司 ID 要求在公司 ID 对应的所有交易代码中，
只返回公司 A 股记录；
- 12 表示传入的是公司 ID 要求在公司 ID 对应的所有交易代码中，
只返回公司 B 股记录；
- 13 表示传入的是公司 ID 要求在公司 ID 同时对应 AB 股的时候，
只返回公司 A 股记录；
- 14 表示传入的是公司 ID 要求在公司 ID 同时对应 AB 股的时候，
只返回公司 B 股记录；

CodeList 代码列表 可以输入交易代码、公司代码列表,n*1 或者 1*n 的 cell 数组，格式例如{'000001';'2';'4356'}，当输入公司 ID 时，marketType 需为 0 ；

fieldList 查询字段列表 ,cell 数组，例如：{'op';'cp'}，可供填写的字段如

下:

- OP： 分时内开盘价
- CP： 分时内收盘价
- HIP： 分时内最高价
- LOP： 分时内最低价
- TM： 分时内成交金额

TQ： 分时分内成交量

详细说明请参阅《国泰安量化投资研究平台数据库说明书》；

beginDate 开始日期,字符串格式,例如:'2012-01-01'；

endDate 结束日期 字符串格式,例如:'2012-01-01'；

beginTime 开始时间,字符串格式,例如:'093000'；

endTime 结束时间 字符串格式,例如:'150000'；

frenquenceType 分时高频类型,共 1 5 10 15 30 60 种类型,分笔表示 1 分钟、5 分钟、10 分钟、15 分钟、30 分钟、60 分钟频率的高频数据。

4. 公式:

无

5. 中文含义:

本 API 根据客户输入的代码、字段、日期时间范围,查询出分时高频数据,并以行为代码,列为时间的矩阵返回。

本 API 适用的范围是量化投资平台高频数据库的所有年份数据,三大特点是:

1)支持公司 ID、指数代码、股指期货代码等品种构成的投资组合批量查询,将自动过滤不同公司代码对应相同股票在同时间区间的重复数据;

2)支持查一段时期一段固定时间区间的数据,比如 2012-07-08 至 2012-08-09 期间每天 093000 至 100000 之间的分时数据;

3)支持 1,5,10,15,30,60 共 6 种频率分时数据的查询。

本 API 将为策略历史回验的数据准备提供极大的方便。

6. 输出参数说明:

Results 一个结构体 cell,包含 k 个矩阵,k = 查询的指标字段个数,

当字段为数值型时 ,返回 double 的矩阵 ,否则返回 cell 矩阵 ;

rowIndex 行索引 , cell 数组 , 包含数值结果矩阵的行索引 , 当

marketType 输入为 0 时 是 n*6 的矩阵 ,否则是 n*1 的向量 ;

columnIndex 列索引 , cell 数组 , 包含数值结果矩阵的列索引。

7. 示例:

```
>>[results,rowIndex,columnIndex]=getHighFrequencySnapShotData(
    1,{'000001';'000002'},    {'op';'cp'}, '2012-06-04', '2012-06-08',
    '093500', '100500', 1);
```

```
>>[results,rowIndex,columnIndex]=getHighFrequencySnapShotData(
    0,{'2290';'431'},{'cp';'op'},'2011-07-01','2011-07-01','093459','10050
    1',5);
```

```
>>[results,      rowIndex,      columnIndex]      =
    getHighFrequencySnapShotData(5, {'A1201';'A1203'} , {'cp';'op'},
    '2011-12-01', '2011-12-01', '080000', '130501', 1);
```

1.10. getHighFrequencyTickData- 分笔高频数据提取 API

函数解释 :

1. **用途:** 根据输入的代码列表、字段、日期、时间范围等查询出具体的分笔高频数据矩阵返回。

2. 函数详细结构:

[results, rowIndex, columnIndex] = getHighFrequencyTickData

(marketType,codeList, fieldList,Date,beginTime,endTime)

3. 输入参数说明:

marketType 市场类型，整数，标识 codeList 参数的类型：

0 表示的代码为公司 ID；

1 表示深圳市场交易代码；

2 表示上海市场交易代码；

4 表示中金所交易代码；

5 表示大商所交易代码；

6 表示郑商所交易代码；

7 表示上期所交易代码；

11 表示传入的是公司 ID 要求在公司 ID 对应的所有交易代码中，

只返回公司 A 股记录；

12 表示传入的是公司 ID 要求在公司 ID 对应的所有交易代码中，

只返回公司 B 股记录；

13 表示传入的是公司 ID 要求在公司 ID 同时对应 AB 股的时候，

只返回公司 A 股记录；

14 表示传入的是公司 ID 要求在公司 ID 同时对应 AB 股的时候，

只返回公司 B 股记录；

codeList 代码列表 可以输入交易代码、公司代码列表,n*1 或者 1*n 的 cell

数组，格式例如{'000001';'2';'4356'}，当输入公司 ID 时，参数

marketType 需为 0；

fieldList 查询字段列表，n*1 或者 1*n 的 cell 数组，例如：{'op';'cp'}；字

段明细如下：

LASTCLOSE： 昨收盘价

OP： 今开盘价

CP： 最新成交价

TQ： 累计成交量

TM： 累计成交金额

TT： 累计成交笔数

TABQ： 累计主动买盘量

TABM： 累计主动买盘金额

TASQ： 累计主动卖盘量

TASM： 累计主动卖盘金额

CQ： 分笔期间成交量

CM： 分笔期间成交金额

CT： 分笔期间成交笔数

HIP： 截至当前最高价

LOP： 截至当前最低价

SYL1： 市盈率 1

SYL2： 市盈率 2

RF1： 价格涨跌 1

RF2： 价格升跌 2

S5： 卖价五

S4： 卖价四

S3： 卖价三

S2： 卖价二

S1： 卖价一

B1： 买价一

B2： 买价二

B3： 买价三

B4： 买价四

B5： 买价五

SV5： 卖量五

SV4： 卖量四

SV3： 卖量三

SV2： 卖量二

SV1： 卖量一

BV1： 买量一

BV2： 买量二

BV3： 买量三

BV4： 买量四

BV5： 买量五

ULP： 涨停价

LLP： 跌停价

POI： 昨持仓量

OI： 持仓量

SP： 今结算

PSP： 昨日结算价

CLOSEP： 今收盘

PREDELTA： 昨虚实度

DELTA： 今虚实度

SETTLEID： 结算编号

详细说明请参阅《国泰安量化投资研究平台数据库说明书(3.0版)》；

Date 查询数据日期,字符串格式,例如:'2012-01-01'；

beginTime 开始时间,字符串格式,例如:'093000'；

endTime 结束时间 字符串格式,例如:'150000'。

公式:

无

4. 中文含义:

本 API 根据客户输入的代码、字段、日期时间范围,查询出分时高频数据,并以行为代码,列为时间的矩阵返回。

本 API 适用的范围是量化投资平台高频数据库的所有年份数据,支持公司 ID、指数代码、股指期货代码等品种构成的投资组合批量查询,将自动过滤不同公司代码对应相同股票在同时间区间的重复数据；

本 API 将为策略历史回验的数据准备提供极大的方便。

5. 输出参数说明:

Results 一个结构体 cell , 包含 k 个矩阵 , k = 查询的指标字段个数 ,
 当字段为数值型时 ,返回 double 的矩阵 ,否则返回 cell 矩阵 ;

rowIndex 行索引 , cell 数组 , 包含数值结果矩阵的行索引 , 当
 marketType 输入为 0 时 ,是 n*6 的矩阵 ,否则是 n*1 的向量 ;

columnIndex 列索引 , cell 数组 , 包含数值结果矩阵的列索引,格式 :
 yyyy-mm-dd hh:mm:ss:mmm。

6. 示例:

```
>>[results, rowIndex,columnIndex] = getHighFrequencyTickData(1,  
{'000001';'000002'}, {'op';'cp'}, '2011-07-01', '093500', '100500');
```

```
>>[results, rowIndex,columnIndex] = getHighFrequencyTickData(0,  
{'2290';'431'}, {'cp';'op'}, '2011-07-01', '093459', '100501');
```

```
>>[results,rowIndex,columnIndex] = getHighFrequencyTickData(4,  
{'IF1212'}, {'op';'cp'}, '2012-06-05', '093500', '105500');
```

```
>>[results,rowIndex,columnIndex] = getHighFrequencyTickData(5,  
{'CU1210'}, {'op';'cp'}, '2012-06-05', '112500', '135500');
```

2. 技术指标 API

2.1. 底层函数

2.1.1. AVEDEV-平均绝对离差

函数解释：

1. **用途:** 计算平均绝对离差 AVEDEV 指标值。

2. **函数详细结构:**

[avedev] = AVEDEV(price,period)

3. **输入参数说明:**

price-为输入的数据矩阵。

period-为计算的周期数。

4. **公式:**

$$\text{avedev} = \text{SUM}(\text{price} - \text{MEAN}(\text{price}, \text{period})) / \text{period}$$

5. **中文含义:**

计算各周期值与 period 周期内平均值的距离总和，然后取其平均数。

6. **输出参数说明:**

avedev-是计算结果矩阵，与输入的 price 为同型矩阵，表示平均绝对离差值矩阵。

7. **示例:**

```
>>closePrice = [ 20.4000  19.5000  19.4800  19.1500  19.2000  18.8600
```

```

18.7000    19.1000    19.7100    19.2500    18.2300    17.4100    17.0800

16.1100    14.5000;  5.4200    5.3900    5.4000    5.2900    5.2600

5.2500     5.1700     5.1600     5.2900     5.2700     5.1500     5.0000

4.9100     4.7300     4.2600]

>> avedev= AVEDEV (closePrice,5)

avedev=

NaN        NaN        NaN        NaN        0.3416    0.2016    0.2384

0.1776     0.2728     0.2848     0.4264     0.7360    0.9152    0.8992

1.0888

NaN        NaN        NaN        NaN        0.0616    0.0616    0.0568

0.0488     0.0488     0.0504     0.0576     0.0848    0.1352    0.1584

0.2520

```

2.1.2. DMA-动态移动平均值

函数解释：

1. **用途:** 计算序列的动态移动平均值。当前序列值的权重为 `weight` , 前一周期 DMA 值的权重为 `1-weight`。

2. **函数详细结构:**

```
[ dma ]= DMA( price, weight )
```

3. **输入参数说明:**

`price`-为输入的数据矩阵。

weight-为权重参数，表示当前序列值的权重 $0 < \text{weight} \leq 1$ ，小数。

4. 公式:

$$\text{DMA}_t(\text{price}, \text{weight}) = \text{weight} * \text{price}_0 + (1 - \text{weight}) * \text{DMA}_{t-1}(\text{price}, \text{weight})$$

5. 中文含义:

$$\text{DMA}_t(\text{price}, \text{weight}) = \text{weight} * \text{当前周期价格} + (1 - \text{weight}) * \text{上周期DMA值}$$

6. 输出参数说明:

dma-是计算结果矩阵，与输入的 price 为同型矩阵，表示动态移动平均值矩阵。

7. 示例

```
>>a = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20
24.00];

>>[dma]=DMA(a,5)

dma =

9.9400    9.9650   10.0425   10.1913   10.2106   10.1753   23.9000

24.1100   24.3550   24.3425   24.2713   24.1356
```

2.1.3. EMA-指数平滑移动平均值

函数解释：

1. **用途:** 计算序列的指数平滑移动平均值。当前序列值的权重为 $2/(\text{period}+1)$ ，

前一周期 EMA 值的权重为 $(\text{period}-1)/(\text{period}+1)$ 。

2. 函数详细结构:

[ema]= EMA (price, period)

3. 输入参数说明:

price-为输入的价格矩阵。

period 为周期参数，Period>=1。

4. 公式:

$$\begin{aligned} & \text{EMA}_t(\text{price}, \text{period}) \\ &= \frac{2 * \text{prarray}_0}{\text{period} + 1} + \text{EMA}_{t-1}(\text{price}, \text{weight}) * \frac{\text{period}-1}{\text{period} + 1} \end{aligned}$$

5. 中文含义:

$$\begin{aligned} & \text{EMA}_t(\text{price}, \text{period}) \\ &= \frac{2 * \text{当前周期价格}}{\text{period} + 1} + \text{上周期的EMA值} * \frac{\text{period}-1}{\text{period} + 1} \end{aligned}$$

6. 输出参数说明:

ema-是计算结果矩阵，与输入的 price 为同型矩阵，表示指数平滑移动平均值矩阵

7. 示例:

```
>>closePrice = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20
24.00];

>>[ ema]= EMA (closePrice,5)

ema =

9.9400    9.9567    10.0111    10.1207    10.1572    10.1514    23.9000
```

24.0400 24.2267 24.2611 24.2407 24.1605

2.1.4. HHV-序列最大值

函数解释：

1. **用途:** 计算序列 period 周期内的最大值。

2. **函数详细结构:**

[hhv]= HHV(price, period)

3. **输入参数说明:**

price 为输入的数据矩阵。

period 为计算的周期数。

4. **输出参数说明:**

hhv-是计算结果矩阵，与输入的 price 为同型矩阵，表示阶段最高值矩阵。

5. **示例:**

```
>> a = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20
```

```
24.00];
```

```
>> [hhv]= HHV(a,5)
```

```
hhv =
```

NaN	NaN	NaN	NaN	10.3400	10.3400
NaN	NaN	NaN	NaN	24.6000	24.6000

2.1.5. LLV-序列最小值

函数解释：

1. **用途:** 计算序列 period 周期内的最小值。

2. **函数详细结构:**

[llv]= LLV(price ,period)

3. **输入参数说明:**

price -为输入的数据矩阵。

period-为计算的周期数。

4. **输出参数说明:**

llv-是计算结果矩阵，与输入的 price 为同型矩阵，表示阶段最小值矩阵

5. **示例:**

```
>>a = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20  
24.00];
```

```
>> [llv]= LLV(a, 5)
```

```
llv =
```

NaN	NaN	NaN	NaN	9.9400	9.9900
NaN	NaN	NaN	NaN	23.9000	24.0000

2.1.6. MA -简单移动平均值

函数解释：

1. **用途:** 计算序列的简单移动平均值，是 period 周期内序列值的算术平均值。

2. 函数详细结构:

[ma]= MA(price , period)

3. 输入参数说明:

price-为输入的 m*n 维价格矩阵 , 可以同时输入 m 个股票 n 日的价格数据。

period-为计算的周期数 , 1 <= period <= n。

4. 公式:

$$MA(price, period) = \frac{\sum_{i=0}^{period-1} price_i}{period}$$

5. 中文含义:

$$MA(price, period) = \frac{\text{最后的}period\text{个周期的序列值之和}}{\text{周期数}period}$$

6. 输出参数说明:

ma-是计算结果矩阵 与输入的Array为同型矩阵 表示简单移动平均值矩阵。

7. 示例:

```
>>price = [9.94 9.99 10.12 10.34 10.23 10.14;
```

```
23.90 24.32 24.60 24.33 24.20 24.00];
```

```
>>[ma]=MA(price,5)
```

ma =

NaN	NaN	10.0167	10.1500	10.2300	10.2367
-----	-----	---------	---------	---------	---------

NaN	NaN	24.2733	24.4167	24.3767	24.1767
-----	-----	---------	---------	---------	---------

2.1.7. SMA -指数移动平均值

函数解释：

1. **用途:** 计算序列的指数移动平均值。当前序列值的权重为 $\text{weight}/\text{period}$, 前一周期 SMA 值的权重为 $(\text{period}-\text{weight})/\text{period}$ 。

2. **函数详细结构:**

[sma] = SMA(price, period, weight)

3. **输入参数说明:**

price-为输入的价格矩阵。

period-为周期参数， $\text{period} \geq 1$, 整数。

weight-为权重参数，表示权重的分子， $1 \leq \text{weight} \leq \text{period}$, 整数。

4. **公式:**

$$\begin{aligned} \text{SMA}_t(\text{price}, \text{period}, \text{weight}) = & \frac{\text{weight}}{\text{period}} * \text{price}_0 \\ & + \frac{\text{period} - \text{weight}}{\text{period}} * \text{SMA}_{t-1}(\text{price}, \text{period}, \text{weight}) \end{aligned}$$

5. **中文含义:**

$$\begin{aligned} \text{SMA}_t(\text{price}, \text{period}, \text{weight}) = & \frac{\text{weight}}{\text{period}} * \text{当前周期价格} \\ & + \frac{\text{period} - \text{weight}}{\text{period}} * \text{上周期的SMA值} \end{aligned}$$

6. **输出参数说明:**

sma 是计算结果矩阵，与输入的 price 为同型矩阵，表示指数加权移动平均值矩阵。

7. **示例:**

```
>>closePrice = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33
24.20 24.00];

>>[sma]=SMA(a,5,2);

sma =

    9.9400    9.9700   10.0600   10.2280   10.2292   10.1757   23.9000

    24.1520   24.4208   24.3663   24.2665   24.1066
```

2.2. 趋向指标

2.2.1. ACD-升降线指标

函数解释：

1. **用途：** 计算升降线指标 ACD 指标值。ACD 线下降，而股价上升时，为卖出信号；ACD 线上升，而股价下降时，为买进信号。

2. 函数详细结构:

```
[ acd ]= ACD(highPrice, lowPrice, closePrice )
```

3. 输入参数说明:

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

closePrice 为输入的成交量序列，为 M*N 数值矩阵。

4. 公式:

```

LC=REF(closePrice,1);
DIF=closePrice-IF(closePrice>LC,
MIN(lowPrice,LC),MAX(highPrice,LC));
ACD = SUM(IF(closePrice==LC,0,DIF),0);

```

5. 中文含义:

LC=昨收

DIF=收盘价-如果收盘价>LC,返回最低价和 LC 的较小值,否则返回最高价和 LC 的较大值

ACD=如果收盘价等于昨收则返回 0,否则返回 DIF 的历史累和

6. 输出参数说明:

acd-是计算结果矩阵 ,与输入的价格序列为同型矩阵 ,表示升降线指标值矩阵。

7. 示例:

```

>> a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];

>> b = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20 24.00];

>> c = [10987654 13643303 13245654 20954432 18094333 15237873;4559934
8959874 7645532 6572344 3455221 6984451];

>> [acd] = ACD (a,b,c) ;

acd =

1.0e+007 *

      NaN      NaN      NaN      3.4200      1.7697      1.5238

      NaN      NaN      NaN      0.6572     -0.5505      0.2794

```

2.2.2. ADX-动向平均数指标

函数解释：

1. **用途:** 计算动向平均数指标 ADX 指标值。ADX 常与 MDI 和 PDI 配合使用，当股价走势向上发展时，同时 PDI 从下向上突破 MDI，表明市场上做多力量在加强，为买入信号。如果伴随 ADX 上升，则预示股价的涨势可能更强劲；当股价走势向下发展时，同时 MDI 从上向下突破 PDI，表明市场上做空力量在加强，为卖出信号。如果伴随 ADX 下降，则预示股价的跌势将加剧。

2. **函数详细结构:**

[adx]= ADX(highPrice, lowPrice, closePrice, period1, period2)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period1 为周期参数 1，period1 >=1，整数。

period2 为周期参数 2，period2 >=1，整数。period1+period2<N

4. **公式:**

$$DX = (PDI(highPrice, lowPrice, closePrice, Period1) - MDI(highPrice, lowPrice, closePrice, Period1)) / (PDI(highPrice, lowPrice, closePrice, Period1) + MDI(highPrice, lowPrice, closePrice, Period1)) * 100;$$
$$ADX = MA(DX, Period2);$$

5. **中文含义:**

DX=(PDI(最高价,最低价,收盘价,period1)-MID(最高价,最低价,收盘

价,period1))/(PDI(最高价,最低价,收盘价,period1)+MID(最高价,最低价,收盘价,period1))*100

ADX=DX 的 period2 日简单移动平均

6. 输出参数说明:

adx-是计算结果矩阵,与输入的价格序列为同型矩阵,表示趋向指标值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];  
>>b = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20 24.00];  
>>c = [10987654 13643303 13245654 20954432 18094333 15237873;4559934  
8959874 7645532 6572344 3455221 6984451];  
>> [adx] = ADX(a,b,c,2,3);  
  
adx =  
  
      NaN      NaN      NaN      NaN -24.7795      6.1728  
  
      NaN      NaN      NaN      NaN  -0.2222     -4.3210
```

2.2.3. BBI –多空指数

函数解释：

1. **用途:** 计算多空指数 BBI 指标值。股价位于 BBI 上方,视为多头市场。股价位于 BBI 下方,视为空头市场。

2. 函数详细结构:

[bbi]= BBI(closePrice, period1, period2, period3, period4)

3. 输入参数说明:

closePrice 为输入的收盘价序列 , 为 M*N 数值矩阵。

period1 为周期参数 1 , $1 \leq \text{period1} \leq N$, 整数。

period2 为周期参数 2 , $1 \leq \text{period2} \leq N$, 整数。

period3 为周期参数 3 , $1 \leq \text{period3} \leq N$, 整数。

period4 为周期参数 4 , $1 \leq \text{period4} \leq N$, 整数。

4. 公式:

$$\text{BBI} = (\text{MA}(\text{closePrice}, \text{period1}) + \text{MA}(\text{closePrice}, \text{period2}) + \text{MA}(\text{closePrice}, \text{period3}) + \text{MA}(\text{closePrice}, \text{period4})) / 4$$

5. 中文含义:

BBI = (收盘价的 period1 日简单移动平均+收盘价的 period2 日简单移动平均+收盘价的 period3 日简单移动平均+收盘价的 period4 日简单移动平均)/4

6. 输出参数说明:

BBI-是计算结果矩阵 , 与 closePrice 输入序列为同型矩阵 , 表示计算得到的

BBI

值矩阵。

7. 示例:

```
>>a = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20 24.00];
```

```
>>[bbi]= BBI(a,1,2,3,4 )
```

bbi =

NaN	NaN	NaN	10.2044	10.2288	10.1923
NaN	NaN	NaN	24.3748	24.3010	24.1398

2.2.4. CHO-佳庆指标

函数解释：

1. **用途:** 计算佳庆指标 CHO 指标值。股价>90 天平均线，CHO 由负转正时，为买进信号；股价<90 天平均线，CHO 由正转负时，为卖出信号。

2. **函数详细结构:**

[cho]= CHO(highPrice, lowPrice, closePrice, TradingVolume, period1, period2)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

TradingVolume 为输入的成交量序列，为 M*N 数值矩阵。

period1 为周期参数 1，1 = < period1 = < N，整数。

period2 为周期参数 2，1 = < period2 = < N，整数。

4. **公式:**

$$MID = \frac{\sum (VOL * (2 * closePrice - highPrice - LOW))}{(highPrice + lowPrice), 0};$$

$$CHO = MA(MID, Period1) - MA(MID, Period2);$$

5. **中文含义:**

MID=成交量(手)*(2*收盘价-最高价-最低价)/(最高价+最低价)的历史累和

CHO=MID 的 period1 日简单移动平均-MID 的 period2 日简单移动平均

6. 输出参数说明:

cho-是计算结果矩阵,与输入的价格序列为同型矩阵,表示佳庆指标值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];  
>>b = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20 24.00];  
>>c = [9.71 9.93 9.80 9.99 10.24 10.26; 23.50 23.88 24.55 24.51 24.19 24.05];  
>>d = [10987654 13643303 13245654 20954432 18094333 15237873;4559934  
8959874 7645532 6572344 3455221 6984451];  
>>[cho] = CHO(a,b,c,d,2,3);
```

cho =

1.0e+005 *

NaN	NaN	-0.5649	-1.4760	-1.4084	-0.1041
NaN	NaN	-0.3386	-0.0016	0.0463	-0.0351

2.2.5. CYE-市场趋势指标

函数解释：

1. **用途:** 计算市场趋势指标 CYE 指标值。CYE 指标值大于 0 则为上升趋势,CYE 指标值小于 0 则为下降趋势。

2. 函数详细结构:

[cyel]= CYE(closePrice)

[cyel, cyes]= CYE(closePrice)

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 数值矩阵，N>=25。

4. 公式:

MAL=MA(closePrice,5);

MAS=MA(MA(closePrice,20),5);

CYEL = (MAL-REF(MAL,1))/REF(MAL,1)*100;

CYES = (MAS-REF(MAS,1))/REF(MAS,1)*100;

5. 中文含义:

MAL=收盘价的 5 日简单移动平均

MAS=收盘价的 20 日简单移动平均的 5 日简单移动平均

CYEL=(MAL-昨日 MAL)/昨日 MAL*100

CYES=(MAS-昨日 MAS)/昨日 MAS*100

6. 输出参数说明:

cyel 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 CYEL

值矩阵。

cyes 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的

CYES 值矩阵。

7. 示例:

```
>>a=[20.25 ,20.4,19.5 ,19.48 ,19.15 ,19.2 ,18.86 ,18.7 ,19.1 ,19.71 ,19.25 ,18.23 ,1
7.41 ,17.08,16.11,14.50,15.30,15.11 ,16.62 ,15.78 ,15.5 ,15.2 ,15.85 ,16.7 ,18.37 ,17
.81 ,17.81 ,17.81 ,17.81 ,17.81 ,17.81 ,17.5 ,18.2 ,18.2 ,18.5 ,18.44 ,18,17.01 ,17.68
,16.8 ,16.5 ,17.5 ,17.25 ,17.9 ,17.5 ,17,16.8 ,17,16.68 ,15.78 ]
```

```
>>[cye1, cye5 ] = CYE(a)
```

```
cye1 =
```

```
NaN    NaN    NaN    NaN    NaN    -1.06    1.58    -0.83          -0.40
0.59      0.05    -0.66    -1.36          -2.16          -3.93          -5.39
-3.52    -2.86    -0.59          -0.43          1.29          -0.13
0.95     0.10     3.28          2.83          3.11          2.26
1.25     -0.62
```

```
cye5 =
```

```
NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
NaN    NaN    NaN    NaN    -0.98    -0.79    -0.56    -0.40    -0.32    -0.38
```

2.2.6. DDI-方向标准离差指数

函数解释：

1. **用途：** 计算方向标准离差指数 DDI 指标值。DDI 由正变负 ,为多头卖出信号；
由负变正，为多头买入信号。

2. **函数详细结构：**

[ddi]= DDI(highPrice, lowPrice, period)

3. 输入参数说明:

highPrice 为输入的最高价序列，为 M*N 列数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 列数值矩阵。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. 公式:

$$\text{DMZ} = \text{IF}((\text{highPrice} + \text{lowPrice}) \leq (\text{REF}(\text{highPrice}, 1) + \text{REF}(\text{lowPrice}, 1)), 0, \text{MAX}(\text{ABS}(\text{highPrice} - \text{REF}(\text{highPrice}, 1)), \text{ABS}(\text{lowPrice} - \text{REF}(\text{lowPrice}, 1))));$$
$$\text{DMF} = \text{IF}((\text{highPrice} + \text{lowPrice}) \geq (\text{REF}(\text{highPrice}, 1) + \text{REF}(\text{lowPrice}, 1)), 0, \text{MAX}(\text{ABS}(\text{highPrice} - \text{REF}(\text{highPrice}, 1)), \text{ABS}(\text{lowPrice} - \text{REF}(\text{lowPrice}, 1))));$$
$$\text{DIZ} = \text{SUM}(\text{DMZ}, M) / (\text{SUM}(\text{DMZ}, M) + \text{SUM}(\text{DMF}, M));$$
$$\text{DIF} = \text{SUM}(\text{DMF}, M) / (\text{SUM}(\text{DMF}, M) + \text{SUM}(\text{DMZ}, M));$$
$$\text{DDI} = \text{DIZ} - \text{DIF};$$

5. 中文含义:

DMZ=如果(最高价+最低价)<=(昨日最高价+昨日最低价),返回 0,否则返回最高价-昨日最高价的绝对值和最低价-昨日最低价的绝对值的较大值

DMF=如果(最高价+最低价)>=(昨日最高价+昨日最低价),返回 0,否则返回最高价-昨日最高价的绝对值和最低价-昨日最低价的绝对值的较大值

DIZ=DMZ 的 M 日累和/(DMZ 的 M 日累和+DMF 的 M 日累和)

DIF=DMF 的 M 日累和/(DMF 的 M 日累和+DMZ 的 M 日累和)

DDI=DIZ-DIF

6. 输出参数说明:

DDI-是计算结果矩阵，与输入的价格序列为同型矩阵，表示方向标准离差指

数值矩阵。

7. 示例:

```
>> a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];  
>> b = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20 24.00];  
>> [ddi]=DDI(a, b, 5)
```

ddi =

NaN	NaN	1.0000	1.0000	1.0000	0.5610
NaN	NaN	1.0000	0.3647	-1.0000	-1.0000

2.2.7. DFMA -平均线差

函数解释：

1. **用途:** 计算平均线差 DFMA 指标值。DFMA 向上交叉其平均线时，为买进信号。DFMA 向下交叉其平均线时，为卖出信号。

2. **函数详细结构:**

```
[ dfma ]= DFMA(closePrice, period1, period2 )
```

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period1 为短周期参数， $1 \leq \text{period1} \leq N$ ，整数。

period2 为长周期参数， $1 \leq \text{period2} \leq N$ ，整数。

4. 公式:

$$DFMA = MA(closePrice, period1) - MA(closePrice, period2);$$

5. 中文含义:

DFMA= 收盘价的 period1 日简单移动平均-收盘价的 period2 日简单移动平均

6. 输出参数说明:

DFMA-是计算结果矩阵，与输入的价格序列为同型矩阵，表示平均线差值矩阵。

7. 示例:

```
>> a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>> [dfma]= DFMA(a, 3, 5 )
```

dfma =

NaN	NaN	0.0367	-0.0017	0.1033	0.0967
NaN	NaN	0.2233	0.1950	-0.0433	-0.1167

2.2.8. DMI-趋向指标

函数解释：

1. **用途:** 计算趋向指标 DMI 指标值。ADX 在高于 50 的地方向下反折时，应小心股价随时有反转的可能。

2. 函数详细结构:

$[pdi] = DMI(highPrice, lowPrice, closePrice, period1, period2)$

[pdi, mdi]= DMI(highPrice, lowPrice, closePrice, period1, period2)

[pdi, mdi, adx]= DMI(highPrice, lowPrice, closePrice, period1, period2)

[pdi, mdi, adx, adxr]= DMI(highPrice, lowPrice, closePrice, period1,
period2)

3. 输入参数说明:

highPrice 为输入的最高价序列，为 M*N 数值矩阵，
N>=period1+period2+1。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵，N>=period1+period2+1。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵，
N>=period1+period2+1。

period1 为周期参数 1，period1>=1，整数。

period2 为周期参数 2，period2>=1，整数。

4. 公式:

TR = SUM(MAX(MAX(highPrice-lowPrice,ABS(highPrice-REF(closePrice,1))),
ABS(lowPrice-REF(closePrice,1))),period1);
HD = highPrice-REF(highPrice,1);
LD = REF(lowPrice,1)-lowPrice;
DMP = SUM(IF(HD > 0 AND HD > LD,HD,0),period1);
DMM = SUM(IF(LD > 0 AND LD > HD,LD,0),period1);
PDI = DMP*100/TR;
MDI = DMM*100/TR;
ADX = MA(ABS(MDI-PDI)/(MDI + PDI)*100,period2);
ADXR = (ADX + REF(ADX,period2))/2

5. 中文含义:

TR=最高价-最低价和最高价-昨收的绝对值的较大值和昨收-最低价的绝对值

的较大值的 period1 日之和

HD=最高价-昨日最高价

LD=昨日最低价-最低价

DMP=如果 HD>0 并且 HD>LD,返回 HD,否则返回 0 的 period1 日之和

DMM=如果 LD>0 并且 LD>HD,返回 LD,否则返回 0 的 period1 日之和

PDI= DMP*100/TR

MDI= DMM*100/TR

ADX= MDI-PDI 的绝对值/(MDI+PDI)*100 的 period2 日简单移动平均

ADXR= (ADX+period2 日前的 ADX 值) /2

6. 输出参数说明:

pdi-是计算结果矩阵 ,与输入的价格为同型矩阵 ,表示计算得到的 PDI 值矩阵。

mdi-是计算结果矩阵 ,与输入的价格为同型矩阵 ,表示计算得到的 MDI 值矩阵。

adx-是计算结果矩阵 ,与输入的价格为同型矩阵 ,表示计算得到的 ADX 值矩阵。

adxr-是计算结果矩阵 ,与输入的价格为同型矩阵 ,表示计算得到的 ADXR 值矩阵。

7. 示例:

```
highPrice = [ 20.6800  20.3000  19.7700  19.9800  19.5000  19.1000
19.3100  19.3000  19.7300  19.7000  18.9500  18.2300  17.6900
17.0000  15.2900
```

```

5.5100    5.4200    5.4800    5.4000    5.4000    5.2600    5.2800

5.2600    5.2900    5.3000    5.2500    5.2000    5.0000    4.9000

4.6100]

lowPrice = [ 20.1100  19.1000  19.1000  19.0000  19.0600  18.6500

18.5500  18.7600  19.1000  19.2500  18.0700  17.3100  16.5100

16.1000  14.5000

5.3900    5.2900    5.3000    5.2000    5.2200    5.1300    5.1600    5.1200

5.0300    5.2000    5.0800    4.8400    4.8600    4.7100    4.2600]

closePrice = [ 20.4000  19.5000  19.4800  19.1500  19.2000  18.8600

18.7000  19.1000  19.7100  19.2500  18.2300  17.4100  17.0800

16.1100  14.5000

5.4200    5.3900    5.4000    5.2900    5.2600    5.2500    5.1700    5.1600

5.2900    5.2700    5.1500    5.0000    4.9100    4.7300    4.2600]

[pdi mdi adx adxr] = DMI(highPrice,lowPrice,closePrice,3,5) ;

pdi =

NaN      NaN      0  12.7273  14.7887      0  16.0305  15.4412

34.9593  39.4495

NaN      NaN  19.3548  15.7895      0      0  8.0000  7.6923

0  2.7778

mdi =

NaN      NaN  51.2690      0      0  41.4141  31.2977      0

0      0

```

NaN	NaN	32.2581	26.3158	26.3158	29.0323	36.0000	15.3846
32.5000	25.0000						
adx =							
NaN	NaN	NaN	NaN	100.0000	100.0000	77.4194	
77.4194	77.4194	100.0000					
NaN	NaN	NaN	NaN	50.0000	75.0000	87.8788	
65.6566	65.6566	71.1111					
adxr =							
NaN	NaN	NaN	NaN	NaN	NaN	NaN	
88.7097	88.7097	88.7097					
NaN	NaN	NaN	NaN	NaN	NaN	NaN	
57.8283	70.3283	79.4949					

2.2.9. EMV-简单波动指标

函数解释：

1. **用途:** 计算简单波动指标 EMV 指标值。EMV 由下往上穿越 0 轴时，视为中期买进信号。EMV 由上往下穿越 0 轴时，视为中期卖出信号。

2. **函数详细结构:**

[emv]= EMV(highPrice, lowPrice, TradingVolume, period)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

TradingVolume 为输入的成交量序列，为 M*N 数值矩阵。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. 公式:

$$\text{VOLUME} = \text{MA}(\text{tradingVolumn}, \text{period}) / \text{tradingVolumn};$$

$$\text{MID} = 100 * (\text{highPrice} + \text{lowPrice} - \text{REF}(\text{highPrice} + \text{lowPrice}, 1)) / (\text{highPrice} + \text{lowPrice});$$

$$\text{EMV} = \text{MA}(\text{MID} * \text{VOLUME} * (\text{highPrice} - \text{lowPrice}) /$$

$$\text{MA}(\text{highPrice} - \text{lowPrice}, \text{period}), \text{period});$$

5. 中文含义:

$$\text{TRADINGVOLUMEUME} = \text{成交量(手)} \text{的 period 日简单移动平均} / \text{成交量(手)}$$

$$\text{MID} = 100 * (\text{最高价} + \text{最低价} - \text{昨日最高价} + \text{最低价}) / (\text{最高价} + \text{最低价})$$

$$\text{EMV} = \text{MID} * \text{TRADINGVOLUMEUME} * (\text{最高价} - \text{最低价}) \text{的 M 日简单移动平均} /$$

$$\text{最高价} - \text{最低价} \text{的 period 日简单移动平均, period})$$

6. 输出参数说明:

emv 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 EMV

值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>>b = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20 24.00];
```

```
>>c = [10987654 13643303 13245654 20954432 18094333 15237873;4559934
```

```
8959874 7645532 6572344 3455221 6984451];
```

```
>> [emv] = EMV(a,b,c,5);
```

```
emv =
```

NaN	NaN	0.4201	1.0514	0.2555	-1.0269
NaN	NaN	0.8638	-0.4902	-1.0247	-0.6644

2.2.10.GDX-鬼道线指标

函数解释：

1. **用途:** 计算鬼道线 GDX 指标值。通道理论公式，是一种用技术手段和经验判断来决定买卖股票的方法。该公式对趋势线做了平滑和修正处理，更精确的反映了股价运行规律。

2. **函数详细结构:**

```
[ gdx ]= GDX(highPrice, lowPrice, closePrice, period, weight )
```

```
[ gdx, yl ]= GDX(highPrice, lowPrice, closePrice, period, weight )
```

```
[ gdx, yl, zc ]= GDX(highPrice, lowPrice, closePrice, period, weight )
```

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period 为周期参数， $1 \leq \text{period} \leq N$ ，整数。

weight 为权重参数， $1 \leq \text{weight} \leq 100$ ，整数。

4. **公式:**

```

AA=ABS((2*closePrice+highPrice+lowPrice)/4-
MA(closePrice,period))/MA(closePrice,period);
GDX = DMA(closePrice,AA);
YL = (1+weight/100)*GDX;
ZC = (1-weight/100)*GDX;

```

5. 中文含义:

AA=(2*收盘价+最高价+最低价)/4-收盘价的 period 日简单移动平均的绝对

值/收盘价的 period 日简单移动平均

GDX=以 AA 为权重收盘价的动态移动平均

YL=(1+weight/100)*鬼道线

ZC=(1-weight/100)*鬼道线

6. 输出参数说明:

gdx-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 GDX

值矩阵。

yl-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 YL 值

矩阵。

zc-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 ZC 值

矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>>b =[9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20
```



```
24.00];
```

```
>> c =[9.71 9.93 9.80 9.99 10.24 10.26;23.50 23.88 24.55 24.51 24.19
```

```
24.05];
```

```
>> [ gdx, yl, zc ]=GDX(a,b,c,3,60)
```

```
gdx =
```

```
NaN NaN 9.7126 9.7205 9.7376 9.7217
```

```
NaN NaN 23.5349 23.5166 23.5252 23.5280
```

```
yl =
```

```
NaN NaN 15.5402 15.5527 15.5801 15.5547
```

```
NaN NaN 37.6558 37.6266 37.6404 37.6448
```

```
zc =
```

```
NaN NaN 3.8851 3.8882 3.8950 3.8887
```

```
NaN NaN 9.4139 9.4067 9.4101 9.4112
```

2.2.11.JLHB-绝路航标

函数解释：

1. **用途：**计算绝路航标 JLHB 指标值。综合了动量观念、强弱指标与移动平均线的优点，在计算过程中主要研究高低价位与收市价的关系，反映价格走势的强弱和超买超卖现象。在市场短期超买超卖的预测方面又较敏感。

2. **函数详细结构：**

```
[ jlhb ]= JLHB(highPrice, lowPrice, closePrice, period1, period2 )
```

3. 输入参数说明:

highPrice 为输入的最高价序列，为 M*N 数值矩阵，N>=61。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵，N>=61。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵，N>=61。

period1 为权重参数 1， period1 >=1，整数。

period2 为权重参数 2， period2 >=1，整数。

4. 公式:

```
VAR1=(closePrice-LLV(lowPrice,60))/  
(HHV(highPrice,60)-LLV(lowPrice,60))*80;  
B = SMA(VAR1,period1,1);  
VAR2 = SMA(B,period2,1);  
JLHB = IF(CROSS(B,VAR2) AND B<40,50,0);
```

5. 中文含义:

VAR1=(收盘价-60 日内最低价的最低值)/(60 日内最高价的最高值-60 日内最低价的最低值)*80

B=VAR1 的 period1 日[1 日权重]移动平均

VAR2=B 的 period2 日[1 日权重]移动平均

JLHB=如果 B 上穿 VAR2 且 B<40,返回 50,否则返回 0

6. 输出参数说明:

jlhb 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 JLHB 值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>>b = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20 24.00];

>>c = [9.71 9.93 9.80 9.99 10.24 10.26;23.50 23.88 24.55 24.51 24.19 24.05];

>>[jlhb]=JLHB(a, b, c, 5, 10)
```

2.2.12.JS-加速线指标

函数解释：

1. **用途：**计算加速线指标 JS 指标值。加速线指标是衡量股价涨速的工具，加速线指标上升表明股价上升动力增加，加速线指标下降表明股价下降压力增加。

2. **函数详细结构：**

[js]= JS(closePrice, period)

3. **输入参数说明：**

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. **公式：**

$$JS = 100 * (\text{closePrice} - \text{REF}(\text{closePrice}, \text{Period})) / (M * \text{REF}(\text{closePrice}, \text{Period}));$$

5. **中文含义：**

$$JS = 100 * (\text{收盘价} - \text{period 日前的收盘价}) / (\text{period} * \text{period 日前的收盘价})$$

6. **输出参数说明：**

JS-是计算结果矩阵，与输入的价格序列为同型矩阵，表示加速线指标值矩阵。

7. 示例:

```
>> a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>> [js]=JS(a,3)
```

```
js =
```

NaN	NaN	NaN	0.7483	1.3065	1.3847
NaN	NaN	NaN	1.3701	0.4309	-0.5565

2.2.13.MACD -平滑异同平均指标

函数解释：

1. **用途:** 计算平滑异同平均指标 MACD 指标值。DIF 向上交叉 MACD，买进；DIF 向下交叉 MACD，卖出。DIF 连续两次向下交叉 MACD，将造成较大的跌幅。DIF 连续两次向上交叉 MACD，将造成较大的涨幅。DIF 与股价形成背离时产生的信号，可信度较高。

2. 函数详细结构:

```
[ dif ]= MACD(closePrice, period1, period2, period3 )
```

```
[ dif, dea ]= MACD(closePrice, period1, period2, period3 )
```

```
[ dif, dea, macd ]= MACD(closePrice, period1, period2, period3 )
```

3. 输入参数说明:

closePrice 为输入的收盘价矩阵，为 M*N 数值矩阵。

period1 为长周期参数，period1 >=1，整数。

period2 为短周期参数，period2 >=1，整数。

period3 为计算 dea 的周期参数，period3 >=1，整数。

4. 公式:

```
EMA_cur_long = EMA(closePrice,period1);  
EMA_cur_short=EMA(closePrice,period2);  
DIF=EMA_cur_short - EMA_cur_long;  
DEA=EMA(DIF,period3);  
MACD=(DIF-DEA)*2;
```

5. 中文含义:

EMA_cur_period1 = 当前收盘价 period1 周期 EMA 值；

EMA_cur_period2=当前收盘价 period2 周期 EMA 值；

DIF = EMA_cur_period2 – EMA_cur_period1;

DEA = DIF 的 period3 周期 EMA 值

MACD = (DIF-DEA)*2;

6. 输出参数说明:

dif 是计算结果矩阵，与输入序列为同型矩阵，表示计算得到的 dif 值矩阵。

dea 是计算结果矩阵，与输入序列为同型矩阵，表示计算得到的 dea 值矩阵。

macd 是计算结果矩阵，与输入序列为同型矩阵，表示计算得到的 MACD 值矩阵。

7. 示例:

```
>>a = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20 24.00];
```

```
>>[dif,dea,macd] = MACD(a,6,3,5)
```

dif =

0	0.0107	0.0409	0.0929	0.0747	0.0382
0	0.0900	0.1693	0.1156	0.0520	-0.0210

dea =

0	0.0036	0.0160	0.0416	0.0527	0.0478
---	--------	--------	--------	--------	--------

0	0.0300	0.0764	0.0895	0.0770	0.0443
---	--------	--------	--------	--------	--------

macd =

0	0.0143	0.0497	0.1026	0.0441	-0.0193
---	--------	--------	--------	--------	---------

0	0.1200	0.1857	0.0522	-0.0500	-0.1306
---	--------	--------	--------	---------	---------

2.2.14.MDI-下降动向指标

函数解释：

1. **用途:** 计算下降动向指标 MDI 指标值。MDI 常与 PDI 和 ADX 配合使用，当股价走势向上发展时，同时 PDI 从下向上突破 MDI，表明市场上做多力量在加强，为买入信号。如果伴随 ADX 上升，则预示股价的涨势可能更强劲；当股价走势向下发展时，同时 MDI 从上向下突破 PDI，表明市场上做空力量在加强，为卖出信号。如果伴随 ADX 下降，则预示股价的跌势将加剧。

2. **函数详细结构:**

[mdi]= MDI(highPrice, lowPrice, closePrice, period)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. 公式:

$$DDM = IF((highPrice - REF(highPrice, 1)) \leq (lowPrice - REF(lowPrice, 1)), \\ lowPrice - REF(lowPrice, 1), 0);$$
$$TR = MAX(MAX(highPrice - lowPrice, highPrice - REF(closePrice, 1)), \\ lowPrice - REF(closePrice, 1));$$
$$MDI = SUM(DDM, period) / SUM(TR, period) * 100;$$

5. 中文含义:

DDM=如果(最高价-昨日最高价)<=(最低价-昨日最低价),返回最低价-昨日最低价,否则返回 0

TR=最高价-最低价和最高价-昨收的较大值和最低价-昨收的较大值

MDI=DDM 的 period 日累和/TR 的 period 日累和*100

6. 输出参数说明:

mdi-是计算结果矩阵,与输入的价格序列为同型矩阵,表示下降动向指标值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];  
>>b = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20 24.00];  
>>c = [10987654 13643303 13245654 20954432 18094333 15237873;4559934  
8959874 7645532 6572344 3455221 6984451];  
>>[mdi] = MDI(a,b,c,3) ;  
  
mdi =  
  
NaN NaN NaN -57.3770 -76.0870 -314.2857
```

NaN NaN NaN -300.0000 -44.8276 -27.0833

2.2.15.MTM-动力指标

函数解释：

1. **用途:** 计算动力指标 MTM 指标值。MTM 从下向上突破其均线，多头买入信号；MTM 从上向下跌破其均线，多头卖出信号。

2. **函数详细结构:**

[mtm]= MTM(closePrice, period)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. **公式:**

$$\text{MTM} = \text{closePrice} - \text{REF}(\text{closePrice}, \text{period})$$

5. **中文含义:**

$$\text{MTM} = \text{收盘价} - \text{period 日前收盘价}$$

6. **输出参数说明:**

MTM-是计算结果矩阵，与输入的价格序列为同型矩阵，表示动力指标值矩阵。

7. **示例:**

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15]
```

```
>> [mtm ]= MTM(a,5)
```

```
mtm =
```


NaN	NaN	NaN	0.2200	0.3900	0.4100
NaN	NaN	NaN	0.9700	0.3100	-0.4100

2.2.16.OSC -变动速率线

函数解释：

1. **用途:** 计算变动速率线 OSC 指标值。OSC 穿过零线向上，市场走强，可视为买入信号。反之，OSC 跌破零线继续下行，市场走弱，视为卖出信号。

2. **函数详细结构:**

[osc]= OSC(closePrice, period1, period2)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period1 为短周期参数，1 <= period1 <= N，整数。

period2 为长周期参数，1 <= period2 <= N，整数。

4. **公式:**

$$OSC = (MA(closePrice, period1) - MA(closePrice, period2)) / MA(closePrice, period1) * 100$$

5. **中文含义:**

OSC=(收盘价的 period1 日简单移动平均-收盘价的 period2 日简单移动平均)/收盘价的 period1 日简单移动平均*100

6. **输出参数说明:**

OSC-是计算结果矩阵，与输入的价格序列为同型矩阵，表示变动速率线值矩阵。

7. **示例:**

```
>> a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>> [osc] = OSC(a,2,3)
```

```
osc =
```

NaN	NaN	0.3700	-0.0168	1.0151	0.9376
NaN	NaN	0.9202	0.7938	-0.1774	-0.4817

2.2.17.PBX-瀑布线指标

函数解释：

1. **用途:** 计算瀑布线指标 PBX 指标值。股价上升穿越轨道线上限时，回档机率大；股价下跌穿越轨道线下限时，反弹机率大。

2. **函数详细结构:**

```
[ pbx ]= PBX(closePrice, period )
```

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 数值矩阵，N>=4。

period 为周期参数，1 <= period <= N/4，整数。

4. **公式:**

$$\text{EMA_cur} = \text{EMA}(\text{closePrice}, \text{period});$$
$$\text{PBX} = (\text{EMA_cur} + \text{MA}(\text{closePrice}, \text{period} * 2) + \text{MA}(\text{closePrice}, \text{period} * 4)) / 3;$$

5. **中文含义:**

当前收盘价 EMA=EMA (收盘价，period)

PBX=(当前收盘价的 EMA+收盘价的 period*2 日简单移动平均+收盘价的

period*4 日简单移动平均)/3 ;

6. 输出参数说明:

PBX-是计算结果矩阵，与输入的价格序列为同型矩阵，表示瀑布线值矩阵。

7. 示例:

```
>> a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>> [pbx]=PBX(a,3)
```

pbx =

NaN	NaN	NaN	9.96	10.19	10.24
NaN	NaN	NaN	24.44	24.36	24.25

2.2.18.PDI-上升动向指标

函数解释：

- 用途:** 计算上升动向指标 PDI 指标值。PDI 常与 MDI 和 ADX 配合使用，当股价走势向上发展时，同时 PDI 从下向上突破 MDI，表明市场上做多力量在加强，为买入信号。如果伴随 ADX 上升，则预示股价的涨势可能更强劲；当股价走势向下发展时，同时 MDI 从上向下突破 PDI，表明市场上做空力量在加强，为卖出信号。如果伴随 ADX 下降，则预示股价的跌势将加剧。

2. 函数详细结构:

[pdi]= PDI(highPrice, lowPrice, closePrice, period)

3. 输入参数说明:

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. 公式:

$$\begin{aligned} UDM &= IF((\text{highPrice} - REF(\text{highPrice}, 1)) \geq (\text{lowPrice} - REF(\text{lowPrice}, 1)), \\ &\quad \text{highPrice} - REF(\text{highPrice}, 1), 0); \\ TR &= MAX(MAX(\text{highPrice} - \text{lowPrice}, \text{highPrice} - REF(\text{closePrice}, 1)), \\ &\quad \text{lowPrice} - REF(\text{closePrice}, 1)); \\ PDI &= SUM(UDM, \text{period}) / SUM(TR, \text{period}) * 100; \end{aligned}$$

5. 中文含义:

UDM=如果(最高价-昨日最高价)>=(最低价-昨日最低价),返回最高价-昨日最高价,否则返回 0

TR=最高价-最低价和最高价-昨收的较大值和最低价-昨收的较大值

PDI=UDM 的 period 日累和/TR 的 period 日累和*100

6. 输出参数说明:

pdi-是计算结果矩阵，与输入的价格序列为同型矩阵，表示 PDI 值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];  
>>b = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20 24.00];  
>>c = [10987654 13643303 13245654 20954432 18094333 15237873; 4559934  
8959874 7645532 6572344 3455221 6984451];
```

```
>>[pdi] = PDI(a,b,c,3);
```

```
pdi =
```

```
      NaN      NaN      NaN  -24.5902  -69.5652 -371.4286
      NaN      NaN      NaN -421.4286  203.4483  -27.083
```

2.2.19.QACD-快速异同平均指标

函数解释：

1. **用途：**计算快速异同平均指标 QACD 指标值。QACD 是 MACD 的一种进化指标，相对于 MACD 而言，QACD 更适合中短线操作。

2. **函数详细结构：**

```
[ qacd ]= QACD(closePrice, period1, period2, period3 )
```

3. **输入参数说明：**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period1 为周期参数 1，period1 >=1，整数。

period2 为周期参数 2，period2 >=1，整数。

period3 为周期参数 3，period3 >=1，整数。

4. **公式：**

$$\text{EMA_cur_M1} = \text{EMA}(\text{closePrice}, \text{period1});$$
$$\text{EMA_cur_M2} = \text{EMA}(\text{closePrice}, \text{period2});$$
$$\text{DIF} = \text{EMA_cur_M1} - \text{EMA_cur_M2};$$
$$\text{DEA} = \text{EMA}(\text{DIF}, \text{period3});$$
$$\text{QACD} = \text{DIF} - \text{DEA};$$

5. **中文含义：**

当前 period1 周期 EMA 值=EMA (收盘价 , period1);

当前 period2 周期 EMA 值=EMA (收盘价 , period2);

DIF=当前 period1 周期 EMA 值 – 当前 period2 周期 EMA 值 ;

DIF=EMA(DIF,period3);

QACD=DIF-DEA;

6. 输出参数说明:

qacd-是计算结果矩阵，与输入的价格序列为同型矩阵，表示快速异动平均值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>>[qacd]=QACD(a,3,5,8)
```

qacd =

0	0.0194	0.0080	0.0212	0.0584	0.0401
0	0.0493	0.1217	0.1016	0.0218	-0.0340

2.2.20.QR-强弱值指标

函数解释：

1. **用途:** 计算强弱值指标 QR 指标值。指标攀升表明个股走势渐强于大盘，后市看好；指标滑落表明个股走势弱于大盘，可择机换股。同时要结合大盘走势研判，应选择大盘转暖或走牛时出击。

2. 函数详细结构:

[qr]= QR(closePrice, indexClosePrice, period)

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

indexClosePrice 为输入的指数收盘价序列，为 M*N 数值矩阵。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. 公式:

$GG = (\text{closePrice} - \text{REF}(\text{closePrice}, \text{period})) / \text{REF}(\text{closePrice}, \text{period}) * 100;$

$DP = (\text{indexcloseprice} - \text{REF}(\text{indexcloseprice}, \text{period})) / \text{REF}(\text{indexcloseprice}, \text{period}) * 100;$

$QR = \text{EMA}(GG - DP, 2)$

5. 中文含义:

$GG = (\text{收盘价} - \text{period 日前的收盘价}) / \text{period 日前的收盘价} * 100$

$DP = (\text{大盘收盘价} - \text{period 日前的开盘收盘价}) / \text{period 日前的开盘收盘价} * 100$

QR=GG-DP 的 2 日指数移动平均

6. 输出参数说明:

qr-是计算结果矩阵，与输入的价格序列为同型矩阵，表示强弱指标值矩阵。

7. 示例:

```
>>closePrice = [20.25  20.4  19.5  19.48  19.15  19.2  18.86  18.7 19.1
19.71 19.25  18.23  17.41  17.08];

>>indexClosePrice = [2003.10  2015.77  2020.55  2027.86  2028.63
2042.89 2060.31 2061.06 2069.34 2084.85 2120.62 2123.84 2127.39
2129.69]
```

```
>>[qr]=QR(closePrice,indexClosePrice,5)
```

```
qr =
```

```
      NaN      NaN      NaN      NaN      NaN      -7.17      -8.89      -7.03      -5.01  
-1.57 -2.89      -5.24      -8.49     -11.83
```

2.2.21.TRIX-三重指数平滑平均线

函数解释：

1. **用途：** 计算三重指数平滑平均线 TRIX 指标值。TRIX 由下往上交叉平均线时，为长期买进信号。TRIX 由上往下交叉平均线时，为长期卖出信号。

2. **函数详细结构：**

```
[ trix ]= TRIX(closePrice, period )
```

3. **输入参数说明：**

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period 为权重参数，period>=1，整数。

4. **公式：**

$$\text{EMA1_cur} = \text{EMA}(\text{closePrice}, \text{period});$$
$$\text{EMA2_cur} = \text{EMA}(\text{EMA}(\text{closePrice}, \text{period}), \text{period});$$
$$\text{EMA3_cur} = \text{EMA}(\text{EMA}(\text{EMA}(\text{closePrice}, \text{period}), \text{period}), \text{period});$$
$$\text{TRIX} = (\text{EMA3_cur} - \text{REF}(\text{EMA3_cur}, 1)) / \text{REF}(\text{EMA3_cur}, 1) * 100;$$

5. **中文含义：**

当前收盘价 1 重 EMA 值=EMA(收盘价,period);

当前收盘价 2 重 EMA 值=EMA(EMA(收盘价,period),period);

当前收盘价 3 重 EMA 值=EMA(EMA(EMA(收盘价,period),period),period);

$$\text{TRIX} = (\text{当前收盘价 3 重 EMA 值} - \text{上周期收盘价 3 重 EMA 值}) / \text{上周期收盘价 3 重 EMA 值} * 100$$

6. 输出参数说明:

trix 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 TRIX 值矩阵。

7. 示例:

```
>> a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>> [trix] = TRIX(a, 3)
```

trix =

NaN	0.1913	0.1846	0.3240	0.7758	0.8617
NaN	0.2013	0.6079	0.7618	0.5631	0.2781

2.2.22.UOS-终极指标

函数解释：

1. **用途:** 计算终极指标 UOS 指标值。UOS 上升至 50~70 的间，而后向下跌破其N字曲线低点时，为短线卖点；UOS 上升超过 70 以上，而后向下跌破 70 时，为中线卖点；UOS 下跌至 45 以下，而后向上突破其N字曲线高点时，为短线买点；UOS 下跌至 35 以下，产生一底比一底高的背离现象时，为底部特征。

2. 函数详细结构:

[uos]= UOS(highPrice, lowPrice, closePrice, period1, period2, period3)

3. 输入参数说明:

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

closePrice 为输入的成交量序列，为 M*N 数值矩阵。

period1 为周期参数 1， $1 \leq \text{period1} < N$ ，整数。

period2 为周期参数 2， $1 \leq \text{period2} < N$ ，整数。

period3 为周期参数 3， $1 \leq \text{period3} < N$ ，整数。

4. 公式:

$TH = \text{MAX}(\text{highPrice}, \text{REF}(\text{closePrice}, 1));$

$TL = \text{MIN}(\text{lowPrice}, \text{REF}(\text{closePrice}, 1));$

$ACC1 = \text{SUM}(\text{closePrice} - TL, M1) / \text{SUM}(TH - TL, \text{period1});$

$ACC2 = \text{SUM}(\text{closePrice} - TL, M2) / \text{SUM}(TH - TL, \text{period2});$

$ACC3 = \text{SUM}(\text{closePrice} - TL, M3) / \text{SUM}(TH - TL, \text{period3});$

$$UOS = (\text{ACC1} * \text{period2} * \text{period3} + \text{ACC2} * \text{period1} * \text{period3} + \text{ACC3} * \text{period1} * \text{period2}) * 100 / (\text{period1} * \text{period2} + \text{period1} * \text{period3} + \text{period2} * \text{period3});$$

5. 中文含义:

TH=最高价和昨收的较大值

TL=最低价

ACC1=收盘价-TL 的 period1 日累和/TH-TL 的 period1 日累和

ACC2=收盘价-TL 的 period2 日累和/TH-TL 的 period2 日累和

ACC3=收盘价-TL 的 period3 日累和/TH-TL 的 period3 日累和

$$UOS = (\text{ACC1} * \text{period2} * \text{period3} + \text{ACC2} * \text{period1} * \text{period3} + \text{ACC3} * \text{period1} * \text{period2}) * 100 / (\text{period1} * \text{period2} + \text{period1} * \text{period3} + \text{period2} * \text{period3})$$

6. 输出参数说明:

uos-是计算结果矩阵,与输入的价格序列为同型矩阵,表示终极波动(UOS)

指标值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28; 23.6 23.98 24.56 24.57 24.29 24.15];  
>>b = [9.94 9.99 10.12 10.34 10.23 10.14; 23.90 24.32 24.60 24.33 24.20 24.00];  
>>c = [10987654 13643303 13245654 20954432 18094333 15237873; 4559934  
8959874 7645532 6572344 3455221 6984451];  
>>[uos] = UOS (a,b,c,5);
```

uos=

NaN	NaN	NaN	143.9445	98.1131	88.0892
NaN	NaN	NaN	90.1510	61.7741	156.1549

2.2.23.VMA-变异平均线指标

函数解释：

1. **用途:** 计算变异平均线指标 VMA 指标值。股价高于平均线,视为强势;股价低于平均线,视为弱势。

2. **函数详细结构:**

[vma]= VMA(openPrice, highPrice, lowPrice, closePrice, period)

3. **输入参数说明:**

openPrice 为输入的开盘价序列,为 M*N 数值矩阵。

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period 为周期参数， $1 \leq \text{period} \leq N$ ，整数。

4. 公式:

$$\text{VMA} = \text{MA}((\text{highPrice} + \text{openPrice} + \text{lowPrice} + \text{closePrice})/4, \text{period});$$

5. 中文含义:

VMA=(最高价+开盘价+最低价+收盘价)/4 的 period 日简单移动平均

6. 输出参数说明:

vma-是计算结果矩阵 ,与输入的价格序列为同型矩阵 ,表示变异移动平均值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];  
  
>>b = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20  
24.00];  
  
>>c = [9.71 9.93 9.80 9.99 10.24 10.26;23.50 23.88 24.55 24.51 24.19  
24.05];  
  
>>d = [9.73 9.88 9.80 9.95 10.31 10.27; 23.52 23.94 24.54 24.57 24.29  
24.07];  
  
>>[vma] = VMA(a,b,c,d,5)  
  
vma =  
  
NaN NaN NaN NaN 9.9970 10.0855
```

NaN NaN NaN NaN 24.1920 24.2795

2.3. 反趋向指标

2.3.1. ACCER-幅度涨速指标

函数解释：

1. **用途:** 计算幅度涨速指标 ACCER 指标值。求出价格走势的斜率，并进行归一化处理。

2. **函数详细结构:**

[accer]= ACCER(closePrice, period)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵， $N \geq 2$ 。

period 为周期参数， $2 \leq \text{period} \leq N$ ，整数。

4. **公式:**

$$\text{ACCER} = \text{SLOPE}(\text{closePrice}, \text{period}) / \text{closePrice};$$

5. **中文含义:**

$$\text{ACCER} = \text{收盘价的 period 日线性回归斜率} / \text{收盘价}$$

6. **输出参数说明:**

accer-是计算结果矩阵，与输入的价格序列为同型矩阵，表示幅度涨速值矩阵。

7. **示例:**

```
>> closePrice = [ 20.4000  19.5000  19.4800  19.1500  19.2000
18.8600  18.7000  19.1000  19.7100  19.2500  18.2300  17.4100
17.0800  16.1100  14.5000; 5.4200  5.3900  5.4000  5.2900  5.2600
5.2500  5.1700  5.1600  5.2900  5.2700  5.1500  5.0000  4.9100
4.7300  4.2600]

>> [ accer ]= ACCER(closePrice, 3)

accr =

NaN      NaN   -0.0236  -0.0091  -0.0073  -0.0077  -0.0134   0.0063
0.0256   0.0039  -0.0406  -0.0528  -0.0337  -0.0403  -0.0890
NaN      NaN   -0.0019  -0.0095  -0.0133  -0.0038  -0.0087  -0.0087
0.0113   0.0104  -0.0136  -0.0270  -0.0244  -0.0285  -0.0763
```

2.3.2. ADTM-动态买卖气指标

函数解释：

1. **用途:** 计算动态买卖气指标 ADTM 指标值。该指标在+1 到-1 之间波动，低于-0.5 时为很好的买入点，高于+0.5 时需注意风险。

2. **函数详细结构:**

```
[ adtm ]= ADTM(openPrice, highPrice, lowPrice, closePrice, period)
```

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 M*N 列数值矩阵。

highPrice 为输入的最高价序列，为 M*N 列数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 列数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period 为周期参数， $1 \leq \text{period} \leq N$ ，整数。

4. 公式:

$$\text{DTM} = \text{IF}(\text{openPrice} \leq \text{REF}(\text{openPrice}, 1), 0, \\ \text{MAX}((\text{highPrice} - \text{openPrice}), (\text{openPrice} - \text{REF}(\text{openPrice}, 1))));$$

$$\text{DBM} = \text{IF}(\text{openPrice} \geq \text{REF}(\text{openPrice}, 1), 0, \\ \text{MAX}((\text{openPrice} - \text{lowPrice}), (\text{openPrice} - \text{REF}(\text{openPrice}, 1))));$$

$$\text{STM} = \text{SUM}(\text{DTM}, \text{period});$$
$$\text{SBM} = \text{SUM}(\text{DBM}, \text{period});$$

$$\text{ADTM} = \text{IF}(\text{STM} > \text{SBM}, (\text{STM} - \text{SBM}) / \text{STM}, \\ \text{IF}(\text{STM} = \text{SBM}, 0, (\text{STM} - \text{SBM}) / \text{SBM}));$$

5. 中文含义:

DTM=如果开盘价 \leq 昨日开盘价,返回 0,否则返回(最高价-开盘价)和(开盘价-昨日开盘价)的较大值

DBM=如果开盘价 \geq 昨日开盘价,返回 0,否则返回(开盘价-最低价)和(开盘价-昨日开盘价)的较大值

STM=DTM 的 period 日累和

SBM=DBM 的 period 日累和

ADTM=如果 $\text{STM} > \text{SBM}$,返回 $(\text{STM} - \text{SBM}) / \text{STM}$,否则返回如果 $\text{STM} = \text{SBM}$,返回 0,否则返回 $(\text{STM} - \text{SBM}) / \text{SBM}$

6. 输出参数说明:

adtm-是计算结果矩阵，与输入的价格序列为同型矩阵，表示动态买卖气指标

值矩阵。

7. 示例:

```
>>openPrice = [ 20.2700 20.2000 19.1000 19.2000 19.0600
19.0800 19.2000 18.7600 19.1000 19.5800 18.9000 18.0000
17.2200 16.9100 14.9000; 5.5100 5.4100 5.4500 5.3800 5.3000
5.2000 5.2400 5.1300 5.1900 5.3000 5.2000 5.2000 5.0000
4.8200 4.5800]
```

```
>>highPrice = [ 20.6800 20.3000 19.7700 19.9800 19.5000 19.1000
19.3100 19.3000 19.7300 19.7000 18.9500 18.2300 17.6900
17.0000 15.2900; 5.5100 5.4200 5.4800 5.4000 5.4000 5.2600
5.2800 5.2600 5.2900 5.3000 5.2500 5.2000 5.0000 4.9000
4.6100]
```

```
>>lowPrice = [ 20.1100 19.1000 19.1000 19.0000 19.0600
18.6500 18.5500 18.7600 19.1000 19.2500 18.0700 17.3100
16.5100 16.1000 14.5000; 5.3900 5.2900 5.3000 5.2000 5.2200
5.1300 5.1600 5.1200 5.0300 5.2000 5.0800 4.8400 4.8600
4.7100 4.2600]
```

```
>>[ adtm ]= ADTM( openPrice, highPrice, lowPrice, 5 )
```

adtm =

```
NaN NaN NaN NaN 0.7436 0.2125 -0.2812
-0.2812 -0.4577 -0.3421 -0.1633 0.2613 0.2613 0 0
NaN NaN NaN NaN -0.7333 -0.7333 -0.6522
```


-0.5000 -0.4167 -0.2857 -0.2857 -0.2222 -0.2222 0 0

2.3.3. BB-布林极限指标

函数解释：

1. **用途：**计算布林极限指标 BB 指标值。布林极限是由布林线演化出来的一个指标。布林线上下两线间的距离反映了近期股价的平均震荡程度，震荡越剧烈，布林线上下两线的距离越宽，震荡越小，距离越窄。而平均震荡幅度的大小提示了下一步股价走势的性质。

2. **函数详细结构：**

[bb]= BB(closePrice, period, weight)

3. **输入参数说明：**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period 为周期参数， $1 < \text{period} \leq N$ ，整数。

weight 为权重参数， $\text{weight} \geq 1$ ，整数。

4. **公式：**

$\text{MID} = \text{MA}(\text{closePrice}, \text{period})$;

$\text{UPPER} = \text{MID} + \text{weight} * \text{STD}(\text{closePrice}, \text{period})$;

$\text{LOWER} = \text{MID} - \text{weight} * \text{STD}(\text{closePrice}, \text{period})$;

$\text{BB} = (\text{closePrice} - \text{LOWER}) / (\text{UPPER} - \text{LOWER})$;

5. **中文含义：**

MID=收盘价的 period 日简单移动平均

UPPER=MID+weight*收盘价的 period 日估算标准差

LOWPRICEER=MID-weight*收盘价的 period 日估算标准差

$$BB = (\text{收盘价} - \text{LOWPRICEER}) / (\text{UPPER} - \text{LOWPRICEER})$$

6. 输出参数说明:

bb-是计算结果矩阵，与输入的价格序列为同型矩阵，表示布林极限值矩阵。

7. 示例:

```
>>closePrice = [    20.4000    19.5000    19.4800    19.1500    19.2000
18.8600    18.7000    19.1000    19.7100    19.2500    18.2300    17.4100
17.0800    16.1100    14.5000; 5.4200    5.3900    5.4000    5.2900    5.2600
5.2500    5.1700    5.1600    5.2900    5.2700    5.1500    5.0000    4.9100
4.7300    4.2600]

>>[ bb ]= BB(closePrice, 5,2 )

bb =

NaN        NaN        NaN        NaN    0.3281    0.1424    0.1902
0.6148    0.8851    0.5807    0.1578    0.1372    0.2238    0.1831    0.1211
NaN        NaN        NaN        NaN    0.1801    0.2636    0.1877
0.2141    0.7773    0.6770    0.2819    0.1246    0.1784    0.1636    0.0991
```

2.3.4. BIAS-乖离率指标

函数解释：

1. **用途:** 计算乖离率指标 BIAS 指标值。当股价的正乖离扩大到一定极限时，股价会产生向下拉回的作用力；当股价的负乖离扩大到一定极限时，股价会产生

向上拉升的作用力。

2. 函数详细结构:

[bias]= BIAS(closePrice, period)

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period 为周期参数， $1 \leq \text{period} \leq N$ ，整数。

4. 公式:

$$\text{BIAS} = (\text{closePrice} - \text{MA}(\text{closePrice}, \text{period})) / \text{MA}(\text{closePrice}, \text{period}) * 100;$$

5. 中文含义:

$$\text{BIAS} = (\text{收盘价} - \text{收盘价的 period 日简单移动平均}) / \text{收盘价的 period 日简单移动平均} * 100$$

6. 输出参数说明:

bias-是计算结果矩阵，与输入的价格序列为同型矩阵，表示乖离率值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>>[bias]=BIAS(a,3 )
```

bias =

	NaN	NaN	-0.03	0.74	2.61
0.65					
	NaN	NaN	2.13	0.82	-0.75

2.3.5. CCI-顺势指标

函数解释：

1. **用途:** 计算顺势指标 CCI 指标值。CCI 为正值时，视为多头市场；CCI 为负值时，视为空头市场。常态行情时，CCI 波动于 0 到 100 之间；强势行情，CCI 会超出 100。CCI>100 时，买进；直到 CCI<100 时，卖出。CCI<100 时，放空；直到 CCI>-100 时，回补。

2. **函数详细结构:**

[cci]= CCI(highPrice, lowPrice, closePrice, period)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 列数值矩阵，N>=2。

lowPrice 为输入的最低价序列，为 M*N 列数值矩阵，N>=2。

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=2。

period 为周期参数，2 =< period < N，整数。

4. **公式:**

$$TYP = (highPrice + lowPrice + CLOSE) / 3;$$
$$CCI = (TYP - MA(TYP, period)) / (0.015 * AVEDEV(TYP, period));$$

5. **中文含义:**

$$TYP = (\text{最高价} + \text{最低价} + \text{收盘价}) / 3$$
$$CCI = (TYP - TYP \text{ 的 } period \text{ 日简单移动平均}) / (0.015 * TYP \text{ 的 } period \text{ 日平均绝对偏差})$$

6. 输出参数说明:

cci-是计算结果矩阵，与输入的价格序列为同型矩阵，表示顺势指标值矩阵。

7. 示例:

```
>>highPrice = [ 20.6800  20.3000  19.7700  19.9800  19.5000  19.1000
19.3100  19.3000  19.7300  19.7000  18.9500  18.2300  17.6900
17.0000  15.2900 5.5100;  5.4200  5.4800  5.4000  5.4000  5.2600
5.2800  5.2600  5.2900  5.3000  5.2500  5.2000  5.0000  4.9000
4.6100]

>> lowPrice  =  [ 20.1100  19.1000  19.1000  19.0000  19.0600
18.6500  18.5500  18.7600  19.1000  19.2500  18.0700  17.3100
16.5100  16.1000  14.5000; 5.3900  5.2900  5.3000  5.2000  5.2200
5.1300  5.1600  5.1200  5.0300  5.2000  5.0800  4.8400  4.8600
4.7100  4.2600]

>>closePrice = [ 20.4000  19.5000  19.4800  19.1500  19.2000
18.8600  18.7000  19.1000  19.7100  19.2500  18.2300  17.4100
17.0800  16.1100  14.5000; 5.4200  5.3900  5.4000  5.2900  5.2600
5.2500  5.1700  5.1600  5.2900  5.2700  5.1500  5.0000  4.9100
4.7300  4.2600]

>>[ cci ]= CCI( highPrice, lowPrice, closePrice, 5 )

cci =

NaN      NaN      NaN      NaN -78.1736 -145.9695 -85.6559 -9.9857
122.7751  68.5146 -127.4589 -124.6408 -105.5715 -103.7745 -136.6355
```

NaN	NaN	NaN	NaN	-85.5379	-122.9373	-89.1473
-82.8516	-34.2262	159.6244	-110.5072	-163.7427	-109.5571	-113.1498
-145.0428						

2.3.6. CYF-市场能量指标

函数解释：

1. **用途：**计算市场能量指标 CYF 指标值。CYF 反映了市场公众的状态和追涨热情。使用 CYF 判断股票的活跃程度, CYF 小于 10 的股票是冷门股 , CYF 在 20 到 40 之间是活跃股 , CYF 大于 50 是热门股。

2. **函数详细结构:**

[cyf]= CYF(Hsl, period)

3. **输入参数说明:**

Hsl 为输入的换手率序列 , 为 M*N 列数值矩阵。

period 为周期参数 , period >= 1 , 整数。

4. **公式:**

$$CYF = 100 - 100 / (1 + EMA(HSL, period));$$

5. **中文含义:**

$CYF = 100 - 100 / (1 + \text{换手线的 period 日指数移动平均})$

6. **输出参数说明:**

cyf-是计算结果矩阵 , 与输入的价格序列为同型矩阵 , 表示市场能量指标值

矩阵。

7. 示例:

```
>>hsl = [ 0.7060    0.2769    0.0971    0.6948    0.9502    0.4387  
0.7655    0.1869    0.4456    0.7094    0.2760    0.6551    0.1190  
0.9597    0.5853; 0.0318    0.0462    0.8235    0.3171    0.0344    0.3816  
0.7952    0.4898    0.6463    0.7547    0.6797    0.1626    0.4984  
0.3404    0.2238]  
  
>>[ cyf ]= CYF( hsl, 5)  
  
cyf =  
  
41.3849    36.0207    28.9628    33.4849    39.4803    36.7549    39.1211  
32.9172    32.2336    35.6317    31.5560    34.4577    28.0655    36.7095  
36.7796  
  
3.0851     3.5319    23.0114    23.3695    17.6813    21.2834    30.8112  
31.5132    34.3054    37.4880    38.5131    32.0550    32.4617    30.2596  
26.6789
```

2.3.7. DBCD-异同离差乖离率指标

函数解释：

1. **用途:** 计算异同离差乖离率指标 DBCD 指标值。计算不同日的乖离率之间的离差，并对离差进行指数移动平滑处理。

2. **函数详细结构:**

[dbcd]= DBCD(closePrice, period1, period2, period3)

3. 输入参数说明:

closePrice 为输入的收盘价序列 , 为 M*N 数值矩阵。

period1 为周期参数 1 , period1 >=1 , 整数。

period2 为周期参数 2 , period2 >=1 , 整数。 period1+period2 = < N

period3 为周期参数 , period3 >= 1 , 整数。

4. 公式:

$$BIAS = (closePrice - MA(closePrice, period1)) / MA(closePrice, period1);$$

$$DIF = BIAS - REF(BIAS, period2);$$

$$DBCD = SMA(DIF, period3, 1);$$

5. 中文含义:

BIAS=(收盘价-收盘价的 period1 日简单移动平均)/收盘价的 period1 日简单移动平均

DIF=乖离率-日前的(乖离率,period2)

DBCD=DIF 的 period3 日[1 日权重]移动平均

6. 输出参数说明:

dbcd -是计算结果矩阵 , 与输入的价格序列为同型矩阵 , 表示异同离差乖离率值矩阵。

7. 示例:

```
>> a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>> [dbcd]=DBCD(a,2,2,3)
```

```
dbcd =
```


NaN	NaN	NaN	NaN	0.0198	0.0097
NaN	NaN	NaN	NaN	-0.0177	-0.0128

2.3.8. DKX-多空线指标

函数解释：

1. **用途:** 计算多空线指标 DKX 指标值。当多空线上穿其均线时为买入信号；当多空线下穿其均线时为卖出信号。

2. **函数详细结构:**

[dkx]= DKX(openPrice, highPrice, lowPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 M*N 列数值矩阵，N>=20。

highPrice 为输入的最高价序列，为 M*N 列数值矩阵，N>=20。

lowPrice 为输入的最低价序列，为 M*N 列数值矩阵，N>=20。

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=20。

4. **公式:**

$$\begin{aligned} \text{MID} &= (3 * \text{closePrice} + \text{lowPrice} + \text{openPrice} + \text{highPrice}) / 6; \\ \text{DKX} &= (20 * \text{MID} + 19 * \text{REF}(\text{MID}, 1) + 18 * \text{REF}(\text{MID}, 2) + \\ &17 * \text{REF}(\text{MID}, 3) + 16 * \text{REF}(\text{MID}, 4) + 15 * \text{REF}(\text{MID}, 5) + \\ &14 * \text{REF}(\text{MID}, 6) + 13 * \text{REF}(\text{MID}, 7) + 12 * \text{REF}(\text{MID}, 8) + \\ &11 * \text{REF}(\text{MID}, 9) + 10 * \text{REF}(\text{MID}, 10) + 9 * \text{REF}(\text{MID}, 11) + \\ &8 * \text{REF}(\text{MID}, 12) + 7 * \text{REF}(\text{MID}, 13) + 6 * \text{REF}(\text{MID}, 14) + \\ &5 * \text{REF}(\text{MID}, 15) + 4 * \text{REF}(\text{MID}, 16) + 3 * \text{REF}(\text{MID}, 17) + \\ &2 * \text{REF}(\text{MID}, 18) + \text{REF}(\text{MID}, 19)) / 210; \end{aligned}$$

5. **中文含义:**

$\text{MID} = (3 * \text{收盘价} + \text{最低价} + \text{开盘价} + \text{最高价}) / 6$

$$DKX = (20 * MID + 19 * \text{昨日 MID} + 18 * 2 \text{ 日前的 MID} + 17 * 3 \text{ 日前的 MID} + 16 * 4 \text{ 日前的 MID} + 15 * 5 \text{ 日前的 MID} + 14 * 6 \text{ 日前的 MID} + 13 * 7 \text{ 日前的 MID} + 12 * 8 \text{ 日前的 MID} + 11 * 9 \text{ 日前的 MID} + 10 * 10 \text{ 日前的 MID} + 9 * 11 \text{ 日前的 MID} + 8 * 12 \text{ 日前的 MID} + 7 * 13 \text{ 日前的 MID} + 6 * 14 \text{ 日前的 MID} + 5 * 15 \text{ 日前的 MID} + 4 * 16 \text{ 日前的 MID} + 3 * 17 \text{ 日前的 MID} + 2 * 18 \text{ 日前的 MID} + 19 \text{ 日前的 MID}) / 210$$

6. 输出参数说明:

dkx-是计算结果矩阵，与输入的价格序列为同型矩阵，表示多空线值矩阵。

7. 示例:

```
>> openPrice = [ 20.2700    20.2000    19.1000    19.2000    19.0600
19.0800    19.2000    18.7600    19.1000    19.5800    18.9000    18.0000
17.2200    16.9100    14.9000; 5.5100    5.4100    5.4500    5.3800    5.3000
5.2000    5.2400    5.1300    5.1900    5.3000    5.2000    5.2000    5.0000
4.8200    4.5800]

>> highPrice = [ 20.6800    20.3000    19.7700    19.9800    19.5000    19.1000
19.3100    19.3000    19.7300    19.7000    18.9500    18.2300    17.6900
17.0000    15.2900; 5.5100    5.4200    5.4800    5.4000    5.4000    5.2600
5.2800    5.2600    5.2900    5.3000    5.2500    5.2000    5.0000    4.9000
4.6100]

>> lowPrice = [ 20.1100    19.1000    19.1000    19.0000    19.0600
18.6500    18.5500    18.7600    19.1000    19.2500    18.0700    17.3100
16.5100    16.1000    14.5000; 5.3900    5.2900    5.3000    5.2000    5.2200
```

```

5.1300    5.1600    5.1200    5.0300    5.2000    5.0800    4.8400    4.8600

4.7100    4.2600]

>>  closePrice = [    20.4000    19.5000    19.4800    19.1500    19.2000

18.8600    18.7000    19.1000    19.7100    19.2500    18.2300    17.4100

17.0800    16.1100    14.5000; 5.4200    5.3900    5.4000    5.2900    5.2600

5.2500    5.1700    5.1600    5.2900    5.2700    5.1500    5.0000    4.9100

4.7300    4.2600]

>> [ dkx ]= DKX( openPrice, highPrice, lowPrice, closePrice )

```

2.3.9. DPO-区间震荡线指标

函数解释：

1. **用途:** 计算区间震荡线 DPO 指标值。DPO > 0 ,表明目前处于多头市场 ;DPO < 0 , 表明目前处于空头市场。

2. **函数详细结构:**

```
[ dpo ]= DPO(closePrice, period )
```

3. **输入参数说明:**

closePrice 为输入的收盘价序列 , 为 M*N 数值矩阵。

period 为周期参数 , $1 \leq \text{period} \leq 2/3 * (N-1)$, 整数。

4. **公式:**

$$DPO = \text{closePrice} - \text{REF}(\text{MA}(\text{closePrice}, \text{period}), \text{period}/2 + 1);$$

5. **中文含义:**

$DPO = \text{收盘价} - \text{period}/2 + 1 \text{ 日前的收盘价的 period 日简单移动平均}$

6. 输出参数说明:

dpo-是计算结果矩阵 ,与输入的价格序列为同型矩阵 ,表示区间震荡线值矩阵。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>>[dpo]=DPO(a,3)
```

dpo =

NaN	NaN	NaN	NaN	0.4667	0.3333
NaN	NaN	NaN	NaN	0.2433	-0.2200

2.3.10.FSL-分水岭指标

函数解释：

1. **用途:** 计算分水岭指标 FSL 指标值。股价在分水岭之上为强势,反之为弱势。

2. 函数详细结构:

```
[ swl ]= FSL( closePrice, TradingVolume, Capital )
```

```
[ swl, sws ]= FSL( closePrice, TradingVolume, Capital )
```

3. 输入参数说明:

closePrice 为输入的收盘价序列 , 为 $M \times N$ 列数值矩阵 , $N \geq 10$ 。

TradingVolume 为输入的成交量序列 , 为 $M \times N$ 列数值矩阵 , $N \geq 10$ 。

Capital 为输入的流通股本序列 , 为 $M \times N$ 列数值矩阵 , $N \geq 10$ 。

4. 公式:

$$\begin{aligned} \text{SWL} &= (\text{EMA}(\text{closePrice}, 5) * 7 + \text{EMA}(\text{closePrice}, 10) * 3) / 10; \\ \text{SWS} &= \text{DMA}(\text{EMA}(\text{closePrice}, 12), \\ &\quad \text{MIN}(1, (\text{SUM}(\text{tradingVolumn}, 5) / (3 * \text{CAPITAL}))))); \end{aligned}$$

5. 中文含义:

SWL=(收盘价的 5 日指数移动平均*7+收盘价的 10 日指数移动平均*3)/10

SWS=以 MIX(1,100*(成交量(手)的 5 日累和/(3*当前流通股本(手)))为权重收盘价的 12 日指数移动平均的动态移动平均

6. 输出参数说明:

swl 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 SWL 值矩阵。

sws 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 SWS 值矩阵。

7. 示例:

```
>> closePrice = [ 20.40  19.50  19.48  19.15  19.20  18.86  18.70  19.10
19.71  19.25;  5.42  5.39  5.40  5.29  5.26  5.25  5.17  5.16
5.29  5.27  ];

>> TradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70  4314.14
1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10  8686.95
3997.83  8000.68  9106.48  2638.03  1360.69  5797.05  1449.55];

>> capital = [ 6885622.00  6885622.00  6885622.00  6885622.00
```

```

6885622.00    6885622.00    6885622.00    6885622.00    6885622.00

6885622.00; 97464642.00    97464642.00    97464642.00    97464642.00

97464642.00    97464642.00    97464642.00    97464642.00    97464642.00

97464642.00]

>> [ swl, sws ]=FSL( closePrice, TradingVolume, capital )

swl =

20.4000    20.1409    19.9550    19.7298    19.5862    19.3866    19.1998

19.1827    19.3440    19.3217

5.4200    5.4114    5.4082    5.3744    5.3421    5.3167    5.2758    5.2441

5.2592    5.2635

sws =

20.4000    20.2615    20.1413    19.9888    20.3992    20.3984    20.3975

20.3980    20.3957    20.3941

5.4200    5.4154    5.4130    5.3941    5.4200    5.4200    5.4200    5.4200

5.4200    5.4200

```

2.3.11.KDJ-随机指标

函数解释：

1. **用途:** 计算随机指标 KDJ 指标值。J>80 时，回档机率大；J<20 时，反弹机率大。K 在 20 左右向上交叉 D 时，视为买进信号。K 在 80 左右向下交叉 D 时，视为卖出信号。J>100 时，股价易反转下跌；J<0 时，股价易反转上涨。

2. 函数详细结构:

[k]= KDJ(highPrice, lowPrice, closePrice, period1, period2, period3)

[k, d]= KDJ(highPrice, lowPrice, closePrice, period1, period2, period3)

[k, d, j]= KDJ(highPrice, lowPrice, closePrice, period1, period2, period3)

3. 输入参数说明:

highPrice 为输入的最高价序列，为 M*N 数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵。

period1 为周期参数 1， $1 \leq \text{period1} \leq N$ ，整数。

period2 为周期参数 2， $\text{period2} \geq 1$ ，整数。

period3 为周期参数 3， $\text{period3} \geq 1$ ，整数。

4. 公式:

$$\begin{aligned} \text{RSV} &= (\text{closePrice} - \text{LLV}(\text{lowPrice}, \text{period1})) / \\ &(\text{HHV}(\text{highPrice}, \text{period1}) - \text{LLV}(\text{lowPrice}, \text{period1})) * 100; \\ \text{K} &= \text{SMA}(\text{RSV}, \text{period2}, 1); \\ \text{D} &= \text{SMA}(\text{K}, \text{period3}, 1); \\ \text{J} &= 3 * \text{K} - 2 * \text{D} \end{aligned}$$

5. 中文含义:

$\text{RSV} = (\text{收盘价} - \text{period1 日内最低价的最小值}) / (\text{period1 日内最高价的最大值} - \text{period1 日内最低价的最小值}) * 100$

$\text{K} = \text{RSV 的 period2 日}[1 \text{ 日权重}] \text{移动平均}$

$\text{D} = \text{K 的 period3 日}[1 \text{ 日权重}] \text{移动平均}$

$\text{J} = 3 * \text{K} - 2 * \text{D}$

6. 输出参数说明:

k 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 K 值矩阵。

d 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 D 值矩阵。

j 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 J 值矩阵。

7. 示例:

```
>> a = [20.2500    20.6800    20.3000    19.7700    19.9800    19.5000  
19.1000    19.3100    19.3000    19.7300];
```

```
>> b = [20.2500    20.1100    19.1000    19.1000    19.0000    19.0600  
18.6500    18.5500    18.7600    19.1000];
```

```
>> c = [20.2500    20.4000    19.5000    19.4800    19.1500    19.2000  
18.8600    18.7000    19.1000    19.7100];
```

```
>> [k d j]=KDJ(a, b, c, 3, 5, 9)
```

k =

```
      NaN      NaN    25.3165    25.0633    22.3583    21.9683    20.7325  
19.7439    30.2688    43.8761
```

d =

```
      NaN      NaN    25.3165    25.2883    24.9628    24.6301    24.1970  
23.7022    24.4318    26.5923
```

j =

```
      NaN      NaN    25.3165    24.6132    17.1494    16.6448    13.8036  
11.8273    41.9428    78.4436
```


2.3.12.LWR-L 威廉指标

函数解释：

1. **用途:** 计算 L 威廉指标 LWR 指标值。LWR<30，超买；LWR>70，超卖。

2. **函数详细结构:**

[lwr]= LWR(highPrice, lowPrice, closePrice, period1, period2)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 列数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 列数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period1 为周期参数 1，2 ≤ period1 ≤ N，整数。

period2 为周期参数 2，period2 ≥ 1，整数。

4. **公式:**

$$\begin{aligned} RSV &= (HHV(highPrice, M1) - closePrice) / \\ & (HHV(highPrice, M1) - LLV(lowPrice, period1)) * 100; \\ LWR &= SMA(RSV, period2, 1); \end{aligned}$$

5. **中文含义:**

$RSV = (N \text{ 日内最高价的最高值} - \text{收盘价}) / (N \text{ 日内最高价的最高值} - N \text{ 日内最低价的最低值}) * 100$

$LWR = RSV \text{ 的 } period2 \text{ 日}[1 \text{ 日权重}] \text{移动平均}$

6. **输出参数说明:**

lwr-是计算结果矩阵，与输入的价格序列为同型矩阵，表示威廉指标值矩阵。

7. **示例:**

```
>>highPrice = [ 20.6800 20.3000 19.7700 19.9800 19.5000 19.1000
```

```
19.3100    19.3000    19.7300    19.7000    18.9500    18.2300    17.6900
17.0000    15.2900; 5.5100    5.4200    5.4800    5.4000    5.4000    5.2600
5.2800    5.2600    5.2900    5.3000    5.2500    5.2000    5.0000    4.9000
4.6100]
```

```
>>lowPrice  = [ 20.1100    19.1000    19.1000    19.0000    19.0600
18.6500    18.5500    18.7600    19.1000    19.2500    18.0700    17.3100
16.5100    16.1000    14.5000; 5.3900    5.2900    5.3000    5.2000    5.2200
5.1300    5.1600    5.1200    5.0300    5.2000    5.0800    4.8400    4.8600
4.7100    4.2600]
```

```
>>closePrice = [ 20.4000    19.5000    19.4800    19.1500    19.2000
18.8600    18.7000    19.1000    19.7100    19.2500    18.2300    17.4100
17.0800    16.1100    14.5000; 5.4200    5.3900    5.4000    5.2900    5.2600
5.2500    5.1700    5.1600    5.2900    5.2700    5.1500    5.0000    4.9100
4.7300    4.2600]
```

```
>> [ lwr ]= LWR( highPrice, lowPrice, closePrice, 3, 5)
```

```
lwr =
```

```
NaN        NaN    75.9494    78.4518    78.6798    79.7860    80.6709    70.0630
56.3894    55.0084    62.0790    68.8264    70.3890    76.2173    80.9738
NaN        NaN    50.0000    53.5714    58.5714    57.9683    63.4116    65.7293
52.5834    44.2890    46.5423    50.2773    56.8072    64.6294    71.7036
```

2.3.13.ROC-变动速率指标

函数解释：

1. **用途:** 计算变动速率指标 ROC 指标值。本指标的超买超卖界限值随个股不同而不同，使用者应自行调整，一般介于±6.5 之间。

2. **函数详细结构:**

[roc]= ROC(closePrice, period)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. **公式:**

$$\text{ROC} = (\text{closePrice} - \text{REF}(\text{closePrice}, \text{period})) / \text{REF}(\text{closePrice}, \text{period}) * 100;$$

5. **中文含义:**

$$\text{ROC} = (\text{收盘价} - \text{period 日前的收盘价}) / \text{period 日前的收盘价} * 100$$

6. **输出参数说明:**

roc-是计算结果矩阵，与输入的价格序列为同型矩阵，表示变动率值矩阵。

7. **示例:**

```
>>a = [20.2500  20.6800  20.3000  19.7700  19.9800  19.5000  19.1000  
19.3100  19.3000  19.7300];
```

```
>>[roc]=ROC(a,3)
```

```
roc =
```

NaN	NaN	NaN	-2.3704	-3.3849	-3.9409	-3.3890	-3.3534
-1.0256	3.2984						

2.3.14.RSI-相对强弱指标

函数解释：

1. **用途:** 计算相对强弱指标 RSI 指标值。RSI>80 为超买；RSI<20 为超卖。

RSI 以 50 为中界线，大于 50 视为多头行情，小于 50 为空头行情。

2. **函数详细结构:**

[rsi]= RSI(closePrice, period)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 数值矩阵，N>=2。

period 为周期参数， period >= 1，整数。

4. **公式:**

LC = REF(closePrice,1);

RSI_part1=SMA(MAX(closePrice-LC,0),period,1);

RSI_part2=SMA(ABS(closePrice-LC),period,1);

RSI=RSI_part1/RSI_part2*100;

5. **中文含义:**

LC=昨收

RSI_part1=SperiodA(收盘价-昨收与 0 的较大值， period， 1)

RSI_part2=SperiodA(收盘价-昨收的绝对值， period， 1)

RSI=RSI_part1/RSI_part2*100;

6. **输出参数说明:**

rsi 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 RSI 值矩阵。

7. 示例:

```
>>a = [20.2500  20.6800  20.3000  19.7700  19.9800  19.5000  19.1000
19.3100  19.3000  19.7300];

>> [rsi]=RSI(a,3)

rsi =

NaN  100.0000  69.3548  42.2604  53.1406  32.2835  21.6579  37.7839
37.2364  67.5573
```

2.3.15.SKDJ-慢速随机指标

函数解释：

1. **用途:** 计算慢速随机指标 SKDJ 指标值。指标>80 时，回档机率大；指标<20 时，反弹机率大；K 在 20 左右向上交叉 D 时，视为买进信号；K 在 80 左右向下交叉 D 时，视为卖出信号。

2. 函数详细结构:

[k]= SKDJ(highPrice, lowPrice, closePrice, period1, period2)

[k, d]= SKDJ(highPrice, lowPrice, closePrice, period1, period2)

3. 输入参数说明:

highPrice 为输入的最高价序列，为 M*N 列数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 列数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period1 为周期参数 1，period1>=2，整数。

period2 为周期参数 2，period2>=1，整数。period1+period2=<N+1

4. 公式:

LOWV=LLV(lowPrice,M1);
HIGHV=HHV(highPrice,M1);
RSV=EMA((closePrice-LOWV)/(HIGHV-LOWV)*100,M2);
K = EMA(RSV,M2);
D = MA(K,M2);

5. 中文含义:

LOWPRICEV=period1 日内最低价的最低值

HIGHPRICEV=period1 日内最高价的最高值

RSV=(收盘价-LOWPRICEV)/(HIGHPRICEV-LOWPRICEV)*100 的 period2

日指数移动平均

K=RSV 的 period2 日指数移动平均

D=K 的 period2 日简单移动平均

6. 输出参数说明:

k 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 K 值矩阵。

d 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的 D 值矩阵。

7. 示例:

```
>>highPrice = [ 20.6800  20.3000  19.7700  19.9800  19.5000  19.1000
19.3100  19.3000  19.7300  19.7000  18.9500  18.2300  17.6900
17.0000  15.2900; 5.5100  5.4200  5.4800  5.4000  5.4000  5.2600
5.2800  5.2600  5.2900  5.3000  5.2500  5.2000  5.0000  4.9000
```

4.6100]

```
>> lowPrice = [ 20.1100  19.1000  19.1000  19.0000  19.0600
18.6500  18.5500  18.7600  19.1000  19.2500  18.0700  17.3100
16.5100  16.1000  14.5000; 5.3900  5.2900  5.3000  5.2000  5.2200
5.1300  5.1600  5.1200  5.0300  5.2000  5.0800  4.8400  4.8600
4.7100  4.2600]
```

```
>> closePrice = [ 20.4000  19.5000  19.4800  19.1500  19.2000
18.8600  18.7000  19.1000  19.7100  19.2500  18.2300  17.4100
17.0800  16.1100  14.5000; 5.4200  5.3900  5.4000  5.2900  5.2600
5.2500  5.1700  5.1600  5.2900  5.2700  5.1500  5.0000  4.9100
4.7300  4.2600]
```

```
>> [ k d ]= SKDJ( highPrice, lowPrice, closePrice, 3, 5)
```

k =

```
NaN      NaN  24.0506  22.6604  21.7923  20.7394  19.7215  25.1188
35.6494  41.9814  41.2020  36.7425  33.2736  28.0870  22.6611
NaN      NaN  50.0000  48.0159  44.1799  42.5044  38.6831  35.4644
41.2045  49.0539  52.0304  51.4368  47.3549  40.7325  33.2634
```

d =

```
NaN      NaN      NaN      NaN      NaN      NaN  21.7929
22.0065  24.6043  28.6421  32.7346  36.1388  37.7698  36.2573
32.3932
NaN      NaN      NaN      NaN      NaN      NaN  44.6767
```

41.7695 40.4073 41.3821 43.2873 45.8380 48.2161 48.1217
44.9636

2.3.16.SI-摆动指标

函数解释：

1. **用途:** 计算摆动指标 SI 指标值。当摆动指针上破移动平均线时，可视为买入信号；当摆动指针下破移动平均线时，为卖出信号。

2. **函数详细结构:**

[si]= SI(openPrice, highPrice, lowPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 M*N 数值矩阵，N>=2。

highPrice 为输入的最高价序列，为 M*N 数值矩阵，N>=2。

lowPrice 为输入的最低价序列，为 M*N 数值矩阵，N>=2。

closePrice 为输入的收盘价序列，为 M*N 数值矩阵，N>=2。

4. **公式:**

```
LC = REF(closePrice ,1);  
AA = ABS(highPrice-LC);  
BB = ABS(lowPrice-LC);  
CC = ABS(highPrice-REF(lowPrice,1));  
DD = ABS(LC-REF(openPrice,1));  
R = IF(AA > BB AND AA > CC,AA + BB/2 + DD/4,  
IF(BB > CC AND BB > AA,BB + AA/2 + DD/4,CC + DD/4));  
X = (closePrice-LC + (closePrice-OPEN)/2 + LC-REF(openPrice,1));  
SI = 16*X/R*MAX(AA,BB)
```

5. **中文含义:**

LC=昨收

AA=最高价-LC 的绝对值

BB=最低价-LC 的绝对值

CC=最高价-昨日最低价的绝对值

DD=LC-昨日开盘价的绝对值

R=如果 AA>BB AND AA>CC,返回 AA+BB/2+DD/4,否则返回如果 BB>CC

AND BB>AA,返回 BB+AA/2+DD/4,否则返回 CC+DD/4

X=(收盘价-LC=(收盘价-开盘价)/2+LC-昨日开盘价)

SI=16*X/R*AA 和 BB 的较大值

6. 输出参数说明:

si-是计算结果矩阵，与输入的价格序列为同型矩阵，表示累计震荡量指标值矩阵。

7. 示例:

```
>> a = [20.2500  20.6800  20.3000  19.7700  19.9800  19.5000  19.1000
19.3100  19.3000  19.7300];

>> b = [20.2500  20.1100  19.1000  19.1000  19.0000  19.0600  18.6500
18.5500  18.7600  19.1000];

>> c = [20.2500  20.4000  19.5000  19.4800  19.1500  19.2000  18.8600
18.7000  19.1000  19.7100];

>> d = [20.2500  20.2700  20.2000  19.1000  19.2000  19.0600
19.0800  19.2000  18.7600  19.1000];
```

```
>> [si]=SI(d, a, b, c)
```

```
si =
```

```
NaN      3.4400  -16.8506  -4.0142    0.2051    0.7649   -4.2961   -6.3441
```

```
0.7680    11.9909
```

2.3.17.SRDM-动向速度比率指标

函数解释：

1. **用途：**计算动向速度比率 SRDM 指标值。SRDM 指标>80 并下破 80 时，回档机率大；指标<20 并转身时，反弹机率大。

2. **函数详细结构：**

```
[ srdm ]= SRDM( highPrice, lowPrice )
```

3. **输入参数说明：**

highPrice 为输入的最高价序列，为 M*N 数值矩阵，N>=11。

lowPrice 为输入的最低价序列，为 M*N 列数值矩阵，N>=11。

4. **公式：**

$$DMZ = IF((highPrice + lowPrice) \leq (REF(highPrice,1) + REF(lowPrice,1)),0, \\ MAX(ABS(highPrice-REF(highPrice,1)),ABS(lowPrice-REF(lowPrice,1))));$$
$$DMF = IF((highPrice + lowPrice) \geq (REF(highPrice,1) + REF(lowPrice,1)),0, \\ MAX(ABS(highPrice-REF(highPrice,1)),ABS(lowPrice-REF(lowPrice,1))));$$
$$ADMZ = MA(DMZ,10);$$
$$ADMF = MA(DMF,10);$$
$$SRDM = IF(ADMZ > ADMF,(ADMZ-ADMF)/ADMZ, \\ IF(ADMZ = ADMF,0,(ADMZ-ADMF)/ADMF));$$

5. 中文含义:

DMZ=如果(最高价+最低价)<=(昨日最高价+昨日最低价),返回 0,否则返回最高价-昨日最高价的绝对值和最低价-昨日最低价的绝对值的较大值

DMF=如果(最高价+最低价)>=(昨日最高价+昨日最低价),返回 0,否则返回最高价-昨日最高价的绝对值和最低价-昨日最低价的绝对值的较大值

ADMZ=DMZ 的 10 日简单移动平均

ADMF=DMF 的 10 日简单移动平均

SRDM= 如果 ADMZ>ADMF,返回 (ADMZ-ADMF)/ADMZ,否则返回如果 ADMZ=ADMF,返回 0,否则返回(ADMZ-ADMF)/ADMF

6. 输出参数说明:

srdm -是计算结果矩阵，与输入的价格序列为同型矩阵，表示 SRDM 值矩阵。

7. 示例:

```
>>highPrice = [ 20.6800  20.3000  19.7700  19.9800  19.5000  19.1000
19.3100  19.3000  19.7300  19.7000  18.9500  18.2300  17.6900
17.0000  15.2900; 5.5100  5.4200  5.4800  5.4000  5.4000  5.2600
5.2800  5.2600  5.2900  5.3000  5.2500  5.2000  5.0000  4.9000
4.6100]
```

```
>>lowPrice = [ 20.1100  19.1000  19.1000  19.0000  19.0600
18.6500  18.5500  18.7600  19.1000  19.2500  18.0700  17.3100
```

```

16.5100    16.1000    14.5000;      5.3900    5.2900    5.3000    5.2000
5.2200    5.1300    5.1600    5.1200    5.0300    5.2000    5.0800    4.8400
4.8600    4.7100    4.2600]

>>[ srdm ]= SRDM( highPrice, lowPrice )

srdm =

NaN        NaN        NaN        NaN        NaN        NaN        NaN
NaN        NaN        NaN    -0.6648   -0.6399   -0.6667   -0.7685   -0.8198
NaN        NaN        NaN        NaN        NaN        NaN        NaN
NaN        NaN        NaN    -0.5254   -0.6164   -0.7634   -0.7755   -0.8601

```

2.3.18.UDL-引力线指标

函数解释：

1. **用途：**计算引力线指标 UDL 指标值。本指标的超买超卖界限值随个股不同而不同，使用者应自行调整；通常 UDL 高于某个极限时，短期股价会下跌；UDL 低于某个极限时，短期股价会上涨。

2. 函数详细结构：

```
[ udl ]= UDL(closePrice, period1, period2, period3, period4 )
```

3. 输入参数说明：

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period1 为周期参数 1， $1 \leq \text{period1} \leq N$ ，整数。

period2 为周期参数 2， $1 \leq \text{period2} \leq N$ ，整数。

period3 为周期参数 3 , 1 <= period3 = < N , 整数。

period4 为周期参数 4 , 1 <= period4 = < N , 整数。

4. 公式:

$$\text{UDL} = (\text{MA}(\text{closePrice}, \text{period1}) + \text{MA}(\text{closePrice}, \text{period2}) + \text{MA}(\text{closePrice}, \text{period3}) + \text{MA}(\text{closePrice}, \text{period4})) / 4;$$

5. 中文含义:

UDL=(收盘价的 period1 日简单移动平均+收盘价的 period2 日简单移动平均+收盘价的period3日简单移动平均+收盘价的period4日简单移动平均)/4

6. 输出参数说明:

udl-是计算结果矩阵，与输入的价格序列为同型矩阵，表示引力线指标值矩阵。

详细例子：

```
>>closePrice = [      20.4000      19.5000      19.4800      19.1500      19.2000
18.8600      18.7000      19.1000      19.7100      19.2500      18.2300      17.4100
17.0800      16.1100      14.5000; 5.4200      5.3900      5.4000      5.2900      5.2600
5.2500      5.1700      5.1600      5.2900      5.2700      5.1500      5.0000      4.9100
4.7300      4.2600]
```

```
>> [ udl ]= UDL(closePrice, 3,5,7,9 )
```

udl =

```
NaN      NaN      12.0284      11.9243      11.8993      11.8476      11.8101      11.8018
11.8726      11.9184      11.8459      11.6543      11.4734      11.2968      11.0543
NaN      NaN      8.4309      8.4201      8.4093      8.3968      8.3868      8.3784
8.3818      8.3901      8.3893      8.3651      8.3351      8.3001      8.2384
```

2.3.19.WR-威廉指标

函数解释：

1. **用途:** 计算威廉指标 WR 指标值。WR 波动于 0-100 , 0 置于顶部 , 100 置于底部。本指标以 50 为中轴线 , 高于 50 视为股价转强 ; 低于 50 视为股价转弱。本指标高于 20 后再度向下跌破 20 , 卖出 ; 低于 80 后再度向上突破 80 , 买进。

2. **函数详细结构:**

[wr]= WR(highPrice, lowPrice, closePrice, period)

3. **输入参数说明:**

highPrice 为输入的最高价序列 , 为 M*N 数值矩阵。

lowPrice 为输入的最低价序列 , 为 M*N 数值矩阵。

closePrice 为输入的收盘价序列 , 为 M*N 数值矩阵。

period 为周期参数 , $1 \leq \text{period} \leq N$, 整数。

4. **公式:**

$$WR = \frac{HHV(\text{highPrice}, \text{period}) - \text{closePrice}}{(HHV(\text{highPrice}, \text{period}) - LLV(\text{lowPrice}, \text{period}))} * 100;$$

5. **中文含义:**

$$WR = \frac{(\text{period 日内最高价的最高值} - \text{收盘价})}{(\text{period 日内最高价的最高值} - \text{period 日内最低价的最低值})} * 100$$

6. **输出参数说明:**

wr-是计算结果矩阵 , 与输入的价格序列为同型矩阵 , 表示威廉指标值矩阵。

7. 示例:

```
>>a = [20.2500  20.6800  20.3000  19.7700  19.9800  19.5000  19.1000
19.3100  19.3000  19.7300];

>> b = [20.2500  20.1100  19.1000  19.1000  19.0000  19.0600  18.6500
18.5500  18.7600  19.1000];

>> c = [20.2500  20.4000  19.5000  19.4800  19.1500  19.2000  18.8600
18.7000  19.1000  19.7100];

>> [wr]=WR(a, b, c, 3)

wr =

NaN      NaN    74.6835    75.9494    88.4615    79.5918    84.2105    84.2105
27.6316    1.6949
```

2.3.20.WIDTH-布林极限宽度指标

函数解释：

1. **用途:** 计算布林极限宽度指标 WIDTH 指标值。布林极限宽度是由布林线演化出来的一个指标。布林线上下两线间的距离反映了近期股价的平均震荡程度，震荡越剧烈，布林线上下两线的距离越宽，震荡越小，距离越窄。而平均震荡幅度的大小提示了下一步股价走势的性质。

2. 函数详细结构:

[width]= WIDTH(closePrice, period, weight)

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period 为周期参数， $1 < \text{period} \leq N$ ，整数。

weight 为权重参数， $\text{weight} \geq 1$ ，整数。

4. 公式:

$$\begin{aligned} \text{MID} &= \text{MA}(\text{closePrice}, \text{period}); \\ \text{UPPER} &= \text{MID} + \text{weight} * \text{STD}(\text{closePrice}, \text{period}); \\ \text{LOWER} &= \text{MID} - \text{weight} * \text{STD}(\text{closePrice}, \text{period}); \\ \text{WIDTH} &= (\text{UPPER} - \text{LOWER}) / \text{MID}; \end{aligned}$$

5. 中文含义:

MID=收盘价的 period 日简单移动平均

UPPER=MID+weight*收盘价的 period 日估算标准差

LOWPRICEER=MID-weight*收盘价的 period 日估算标准差

WIDTH=(UPPER-LOWPRICEER)/MID

6. 输出参数说明:

width-是计算结果矩阵，与输入的价格序列为同型矩阵，表示布林线开口大

小值矩阵。

7. 示例:

```
>>closePrice = [ 20.4000  19.5000  19.4800  19.1500  19.2000
18.8600  18.7000  19.1000  19.7100  19.2500  18.2300  17.4100
17.0800  16.1100  14.5000; 5.4200  5.3900  5.4000  5.2900  5.2600
5.2500  5.1700  5.1600  5.2900  5.2700  5.1500  5.0000  4.9100
4.7300  4.2600]
```



```
>>[width ]= WIDTH(closePrice, 5 ,2)
```

```
width =
```

```
NaN      NaN      NaN      NaN      0.1030    0.0549    0.0639
0.0449    0.0810    0.0816    0.1181    0.1956    0.2480    0.2697    0.3430
NaN      NaN      NaN      NaN      0.0537    0.0541    0.0631
0.0442    0.0442    0.0454    0.0511    0.0896    0.1299    0.1673    0.2852
```

2.4. 量价指标

2.4.1. AMV-成本线均价指标

函数解释：

1. **用途：** 计算成本线均价指标 AMV 指标值。AMV 为行情支撑线，跌破支撑线时为卖出信号。

2. **函数详细结构：**

```
[ amov , amv ] = AMV(openPrice, closePrice, TradingVolume,  
M)
```

3. **输入参数说明：**

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

TradingVolume 为输入的成交量序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

M 为周期参数， $1 \leq M \leq N$ ，整数。

4. 公式:

$$AMOV = \text{tradingVolumn} * (\text{openPrice} + \text{closePrice}) / 2;$$
$$AMV = \text{SUM}(AMOV, M) / \text{SUM}(\text{tradingVolumn}, M);$$

5. 中文含义:

AMOV=成交量(手)*(开盘价+收盘价)/2

AMV=AMOV 的 M 日累和/成交量(手)的 M 日累和

6. 输出参数说明:

AMV-是计算结果矩阵，与输入的价格序列为同型矩阵，表示方向标准离差指

数值矩阵。

AMOV=成交量(手)*(开盘价+收盘价)/2。

7. 示例:

```
>>a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>>b = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20 24.00];
```

```
>>c = [10987654 13643303 13245654 20954432 18094333 15237873;4559934  
8959874 7645532 6572344 3455221 6984451];
```

```
>>[amv] = AMV(a,b,c,5) ;
```

AMV =

NaN	NaN	NaN	NaN	10.09
10.14				
NaN	NaN	NaN	NaN	24.27

2.4.2. ARBR-人气意愿指标

函数解释：

1. **用途:** 计算人气意愿指标 ARBR 指标值。AR>180，能量耗尽，为卖出信号；

AR<40，能量已累积爆发力，为买进信号。BR>400，暗示行情过热，为反向

卖出信号；BR<40，行情将起死回生，为买进信号。

2. **函数详细结构:**

`[ar,br] = ARBR(openPrice,highPrice,lowPrice,ClosePrice,period)`

3. **输入参数说明:**

`openPrice` - 股票开盘价，m*n 的 double 矩阵，m 个股票 n 日的开盘价，
各列之间按照时间顺序排列；

`highPrice` - 股票最高价，m*n 的 double 矩阵，m 个股票 n 日的最高价，
各列之间按照时间顺序排列，矩阵维数需与 `openPrice` 一致；

`lowPrice` - 股票最低价，m*n 的 double 矩阵，m 个股票 n 日的最低价，
各列之间按照时间顺序排列，矩阵维数需与 `openPrice` 一致；

`ClosePrice` - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，
各列之间按照时间顺序排列，矩阵维数需与 `openPrice` 一致；

`period` - 计算 ARBR 的数值取样周期。

4. 公式:

$$AR = \text{SUM}(\text{highPrice} - \text{openPrice}, \text{period}) / \text{SUM}(\text{openPrice} - \text{lowPrice}, \text{period}) * 100;$$
$$BR = \text{SUM}(\text{MAX}(0, \text{highPrice} - \text{REF}(\text{closePrice}, 1)), \text{period}) /$$
$$\text{SUM}(\text{MAX}(0, \text{REF}(\text{closePrice}, 1) - \text{lowPrice}), \text{period}) * 100$$

5. 中文含义:

AR=最高价-开盘价的 period 日累和/开盘价-最低价的 period 日累和*100

BR=0 和最高价-昨收的较大值的 period 日累和/0 和昨收-最低价的较大值的
period 日累和*100

6. 输出参数说明:

ar - 多空双方力量发展对比指标，m*n 的矩阵，前 period-1 列为空值；

br - 多空双方力量争斗指标，m*n 的矩阵，前 period 列为空值。

7. 示例:

```
>>o = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20  
24.00];
```

```
>>h = [10.11 10.23 10.12 10.52 10.40 10.28;23.98 24.58 24.92 24.60 24.45  
24.15];
```

```
>>l = [9.67 9.75 9.87 9.97 10.12 10.04;23.6 23.9 24.52 24.21 24.20 23.80];
```

```
>>[ar,br] = ARBR(o,h,l,c,3);
```

```

ar =
      NaN      NaN      254.55      412.00      300.00
109.80
      NaN      NaN      1116.67      206.25      112.24
23.75
br =
      NaN      NaN      NaN      178.26      120.41
91.07
      NaN      NaN      NaN      328.21      138.46
13.04

```

2.4.3. CR-能量指标

函数解释：

1. **用途:** 计算能量指标 CR 指标值。CR>400 时，为卖出信号；CR<40，为买进信号。

2. **函数详细结构:**

```
[cr] = CR(highPrice,lowPrice,ClosePrice,period)
```

3. **输入参数说明:**

highPrice - 股票最高价，m*n 的 double 矩阵，m 个股票 n 日的最高价，

各列之间按照时间顺序排列；

lowPrice - 股票最低价，m*n 的 double 矩阵，m 个股票 n 日的最低价，

各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，

各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

period - 计算 CR 的数值取样周期。

4. 公式:

$$\begin{aligned} \text{MID} &= (\text{highPrice} + \text{lowPrice} + \text{closePrice})/3; \\ \text{CR} &= \text{SUM}(\text{MAX}(0, \text{highPrice} - \text{REF}(\text{MID}, 1)), \text{period}) / \\ &\quad \text{SUM}(\text{MAX}(0, \text{REF}(\text{MID}, 1) - \text{lowPrice}), \text{period}) * 100; \end{aligned}$$

5. 中文含义:

$$\text{MID} = (\text{最高价} + \text{最低价} + \text{收盘价}) / 3$$
$$\text{CR} = 0 \text{ 和最高价} - \text{昨日 MID 的较大值的 period 日累和} / 0 \text{ 和昨日 MID} - \text{最低价} \\ \text{的较大值的 period 日累和} * 100$$

6. 输出参数说明:

cr - 能量指标，m*n 的矩阵，前 period 列为空值。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20 24.00];
```

```
>>h = [10.11 10.23 10.12 10.52 10.40 10.28;23.98 24.58 24.92 24.60 24.45 24.15];
```

```
>>l = [9.67 9.75 9.87 9.97 10.12 10.04;23.6 23.9 24.52 24.21 24.20 23.80];
```

```
>> [cr] = CR(h,l,c,3);
```

```
cr =
```

NaN	NaN	NaN	272.82	214.56
146.92				
NaN	NaN	NaN	299.29	111.28
6.18				

2.4.4. CYR-市场强弱指标

函数解释：

1. **用途：**计算市场强弱指标 CYR 指标值。CYR 是成本均线派生出的指标,是 13 日成本均线的升降幅度。使用 CYR 可以对股票的强弱进行排序,找出其中的强势和弱势股票。

2. **函数详细结构:**

```
[cyr] = CYR(tradingTradingVolumeume,tradingValue,period)
```

3. **输入参数说明:**

tradingTradingVolumeume - 某市场交易量 ,m*n 的 double 矩阵 ,可以同时输入 m 个市场或者板块的交易量 ,各列之间按照时间顺序排列 ;

tradingValue - 某市场交易金额 ,m*n 的 double 矩阵 ,可以同时输入 m 个市场或者板块的交易金额 ,矩阵维数需与 tradingTradingVolumeume 一致 ;

period - 计算强弱指标的数值取样周期，当样本区间不足时，不予计算。

4. 公式:

$$\text{DIVE} = 0.01 * \text{EMA}(\text{tradingValue}, \text{period}) / \text{EMA}(\text{tradingVolume}, \text{period});$$
$$\text{CYR} = (\text{DIVE} / \text{REF}(\text{DIVE}, 1) - 1) * 100;$$

5. 中文含义:

$$\text{DIVE} = 0.01 * \text{成交额(元)的 M 日指数移动平均} / \text{成交量(手)的 period 日指数移动平均}$$

$$\text{CYR} = (\text{DIVE} / \text{昨日 DIVE} - 1) * 100 ;$$

6. 输出参数说明:

cyr - 市场强弱指标 ,m*n 的 double 矩阵 ,返回 m 个市场的强弱指标值 ,第一列为空。

7. 示例:

```
>>a=[112 245 856 345 554 54;34 56 87 34 23 6];
```

```
>>b=[896 1343 543 1345 210 1900;56 23 12 45 60 89];
```

```
>>[cyr] = CYR(a,b,3) ;
```

```
cyr =
```

NaN	-21.6036	-74.3761	57.0523	-47.7905	253.9285
NaN	-46.7063	-55.5524	81.3398	84.6653	146.1661

2.4.5. MASS-梅斯线指标

函数解释：

1. **用途:** 计算梅斯线指标 MASS 指标值。当 MASS 指标向上穿越 27，随后又掉头跌落 26.5 以下时,如 K 线的 9 天移动平均线正处于上升状态，代表多头行情将反转下跌 当 MASS 指标向上穿越 27 随后又掉头跌落 26.5 以下时，如 K 线的 9 天移动平均线正处于下跌状态，代表空头行情将反转上涨。

2. **函数详细结构:**

[mass] = MASS(highPrice,lowPrice,period1,period2)

3. **输入参数说明:**

highPrice - 股票最高价，m*n 的 double 矩阵，m 个股票 n 日的最高价，

各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

lowPrice - 股票最低价，m*n 的 double 矩阵，m 个股票 n 日的最低价，

各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

period1 - 计算 MASS 的数值取样周期 1；

period2 - 计算 MASS 的数值取样周期 2。

4. **公式:**

$$\text{MASS} = \text{SUM}(\text{MA}(\text{highPrice} - \text{lowPrice}, \text{period1}) / \text{MA}(\text{MA}(\text{highPrice} - \text{lowPrice}, \text{period1}), \text{period1}), \text{period2});$$

5. 中文含义:

MASS=(最高价-最低价)的 period 1 日简单移动平均/(最高价-最低价)的 period 1 日简单移动平均的 period 1 日简单移动平均的 period 2 日累和

6. 输出参数说明:

MASS - 梅斯线指标，m*n 的矩阵。

7. 示例:

```
>>h = [10.11 10.23 10.12 10.52 10.40 10.28;23.98 24.58 24.92 24.60 24.45  
24.15];
```

```
>>l = [9.67 9.75 9.87 9.97 10.12 10.04;23.6 23.9 24.52 24.21 24.20 23.80];
```

```
>>[mass] = MASS(h,l,2,2) ;
```

```
mass =
```

NaN	NaN	NaN	1.9306	2.0642	1.7888
NaN	NaN	NaN	1.8543	1.7400	1.8628

2.4.6. MFI-资金流量指标

函数解释：

1. **用途:** 计算资金流量指标 MFI 指标值。MFI>80 为超买，当其回头向下跌破 80 时，为短线卖出时机；MFI<20 为超卖，当其回头向上突破 20 时，为短线买进时机；MFI>80，而产生背离现象时，视为卖出信号； MFI<20，而产生

背离现象时，视为买进信号。

2. 函数详细结构:

[mfi] =

MFI(highPrice,lowPrice,ClosePrice,tradingTradingVolumeume,period)

3. 输入参数说明:

highPrice - 股票最高价，m*n 的 double 矩阵，m 个股票 n 日的最高价，
各列之间按照时间顺序排列；

lowPrice - 股票最低价，m*n 的 double 矩阵，m 个股票 n 日的最低价，各
列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，
各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

tradingTradingVolumeume - 股票交易量，m*n 的 double 矩阵，m 个股
票 n 日的交易量，各列之间按照时间顺序排列，矩阵维数需与 highPrice 一
致；

period - 计算 MFI 的数值取样周期。

4. 公式:

$TYP = (highPrice + lowPrice + closePrice) / 3;$

$MFI = \frac{SUM(IF(TYP > REF(TYP,1), TYP * tradingVolume, 0), period)}{SUM(IF(TYP < REF(TYP,1), TYP * tradingVolume, 0), period)};$

5. 中文含义:

$TYP = (\text{最高价} + \text{最低价} + \text{收盘价}) / 3$

MFI=如果 $TYP >$ 昨日 TYP , 返回 $TYP * \text{成交量}$, 否则返回 0 的 period 日累和/如果 $TYP <$ 昨日 TYP , 返回 $TYP * \text{成交量}$, 否则返回 0 的 period 日累和

6. 输出参数说明:

mfi - 资金流量指标, $m * n$ 的矩阵, 前 period 列为空值。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20
24.00];
```

```
>>h = [10.11 10.23 10.12 10.52 10.40 10.28;23.98 24.58 24.92 24.60 24.45
24.15];
```

```
>>l = [9.67 9.75 9.87 9.97 10.12 10.04;23.6 23.9 24.52 24.21 24.20 23.80];
```

```
>>v = [10987654 13643303 13245654 20954432 18094333
15237873;4559934 8959874 7645532 6572344 3455221 6984451];
```

```
>>[mfi] = MFI(h,l,c,v,3);
```

mfi =

NaN	NaN	NaN	NaN	1.2503	0.3317
-----	-----	-----	-----	--------	--------

NaN NaN NaN 1.7281 0.6231 0

2.4.7. OBV-能量潮指标

函数解释：

1. **用途:** 计算能量潮指标 OBV 指标值。股价一顶比一顶高，而 OBV 一顶比一顶低，暗示头部即将形成；股价一底比一底低，而 OBV 一底比一底高，暗示底部即将形成。

2. **函数详细结构:**

[obv] = OBV(ClosePrice,tradingTradingVolumeume)

3. **输入参数说明:**

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，各列之间按照时间顺序排列

tradingTradingVolumeume - 股票交易量，m*n 的 double 矩阵，m 个股票 n 日的交易量，各列之间按照时间顺序排列，矩阵维数需与 ClosePrice 一致；

4. **公式:**

$$OBV = \text{SUM}(\text{IF}(\text{closePrice} > \text{REF}(\text{closePrice},1), \text{tradingVolume}, \text{IF}(\text{closePrice} < \text{REF}(\text{closePrice},1), -\text{tradingVolume}, 0)), 0)$$

5. 中文含义:

OBV=如果收盘价>昨收,返回成交量(手),否则返回如果收盘价<昨收,返回-成交量(手),否则返回 0 的历史累和

6. 输出参数说明:

obv - 能量潮指标, m*1 的矩阵。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20
```

```
24.00];
```

```
>>v = [10987654 13643303 13245654 20954432 18094333
```

```
15237873;4559934 8959874 7645532 6572344 3455221 6984451];
```

```
>>[obv] = OBV(c,v) ;
```

```
obv =
```

```
14511183
```

```
-406610
```

2.4.8. PCNT-幅度比指标

函数解释：

1. **用途:** 计算幅度比指标 PCNT 指标值。PCNT 重视价格的涨跌幅度，排除观察涨跌跳动值；较高的 PCNT 值，表示该股波动幅度大；较低的 PCNT 值，表示该股波动幅度小。

2. **函数详细结构:**

[pcnt] = PCNT(ClosePrice)

3. **输入参数说明:**

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，各列之间按照时间顺序排列。

4. **公式:**

$$PCNT = (closePrice - REF(closePrice, 1)) / closePrice * 100;$$

5. **中文含义:**

$$PCNT = (收盘价 - 昨收) / 收盘价 * 100$$

6. **输出参数说明:**

pcnt - 幅度比指标，m*n 的矩阵，第一列为空值。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20
```

```
24.00];
```

```
>>[pcnt] = PCNT(c);
```

```
>>pcnt =
```

```
NaN    0.5030    1.3013    2.1739   -1.0638   -0.8798
```

```
NaN    1.7573    1.1513   -1.0976   -0.5343   -0.8264
```

2.4.9. PSY-心理线指标

函数解释：

1. **用途:** 计算心理线指标 PSY 指标值。PSY>75 ,股价容易遭遇压力。PSY<25 ,
股价容易获得支撑。

2. 函数详细结构:

```
[psy] = PSY(ClosePrice,period)
```

3. 输入参数说明:

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，各列之间按照时间顺序排列；

period - 计算 PSY 的数值取样周期。

4. 公式:

$$PSY = \text{COUNT}(\text{closePrice} > \text{REF}(\text{closePrice}, 1), \text{period}) / \text{period} * 100$$

5. 中文含义:

PSY=统计 period 日中满足收盘价>昨收的天数/period*100

6. 输出参数说明:

psy - 心理线指标，m*n 的矩阵，前 period-1 列为空。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20
```

```
24.00];
```

```
>>[psy] = PSY(c,3) ;
```

```
psy =
```

```
NaN    NaN     3     2     1     0
```

```
NaN    NaN     2     1     0     0
```

2.4.10.PVT-价量趋向指标

函数解释：

1. **用途:** 计算价量趋向指标 PVT 指标值。PVT 线下降而股价上升时，为卖出信号；PVT 线上升而股价下降时，为买进信号。

2. **函数详细结构:**

[pvt] = PVT(ClosePrice,tradingTradingVolumeume)

3. **输入参数说明:**

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，各列之间按照时间顺序排列；

tradingTradingVolumeume - 股票交易量，m*n 的 double 矩阵，m 个股票 n 日的交易量，各列之间按照时间顺序排列，矩阵维数需与 ClosePrice 一致。

4. **公式:**

$$PVT = \text{SUM}((\text{closePrice} - \text{REF}(\text{closePrice}, 1)) / \text{REF}(\text{closePrice}, 1) * \text{tradingVolume}, 0)$$

5. **中文含义:**

$PVT = (\text{收盘价} - \text{昨收}) / \text{昨收} * \text{成交量的历史累和}$

6. **输出参数说明:**

pvt - 价量趋势, m*1 的矩阵。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20  
24.00];
```

```
>>v = [10987654 13643303 13245654 20954432 18094333  
15237873;4559934 8959874 7645532 6572344 3455221 6984451];
```

```
>>[pvt] = PVT(c,v) ;
```

pvt =

1.0e+007 *

3.6997

0.9716

2.4.11.TAPI-加权指数成交值指标

函数解释：

1. **用途:** 计算加权指数成交值指标 TAPI 指标值。指数上涨, TAPI 值下降, 此为卖出信号, 可逢高出脱或于次日获利了结。指数下降, TAPI 值上扬, 此为买进信号, 可逢低买进。

2. 函数详细结构:

[tapi] = TAPI(tradingValue,indexClosePrice)

3. 输入参数说明:

indexClosePrice - 指数收盘价, $m \times n$ 的 double 矩阵, m 个指数 n 日的收盘价, 各列之间按照时间顺序排列

tradingValue - 股票交易量额, $m \times n$ 的 double 矩阵, m 个股票 n 日的交易量, 各列之间按照时间顺序排列, 矩阵维数需与 indexClosePrice 一致。

4. 公式:

$$TAPI = \text{tradingValue} / \text{indexClosePrice};$$

5. 中文含义:

TAPI=成交额/指数收盘价

6. 输出参数说明:

tapi - 加权指数成交值指标, $m \times n$ 的矩阵。

7. 示例:

```
>>ic = [2994.18 2999.23 3012.56 3034.87 3023.03 3014.33;2390.12  
2432.94 2460.22 2433.34 2420.45 2400.87];
```

```
>>v = [10987654 13643303 13245654 20954432 18094333
```

```
15237873;4559934 8959874 7645532 6572344 3455221 6984451];
```

```
>>[tapi] = TAPI(v,ic);
```

```
tapi =
```

```
1.0e+003 *
```

```
3.6697    4.5489    4.3968    6.9046    5.9855    5.0551
```

```
1.9078    3.6827    3.1077    2.7010    1.4275    2.9091
```

2.4.12.VR-容量比率指标

函数解释：

1. **用途：**计算容量比率指标 VR 指标值。VR>450 ,市场成交过热 ,应反向卖出。

VR<40 , 市场成交低迷 , 人心看淡之际 , 应反向买进。

2. **函数详细结构:**

```
[vr] = VR(ClosePrice,tradingTradingVolume,period)
```

3. **输入参数说明:**

ClosePrice - 股票收盘价 , m*n 的 double 矩阵 , m 个股票 n 日的收盘价 , 各列之间按照时间顺序排列 ;

tradingTradingVolume - 股票成交量 , m*n 的 double 矩阵 , m 个股

票 n 日的成交量，各列之间按照时间顺序排列，矩阵维数需与 ClosePrice 一致；

period - 计算容量比率指标的数值取样周期。

4. 公式:

$$\begin{aligned} LC &= \text{REF}(\text{closePrice}, 1); \\ VR &= \text{SUM}(\text{IF}(\text{closePrice} > LC, \text{tradingVolume}, 0), \text{period}) / \\ &\quad \text{SUM}(\text{IF}(\text{closePrice} \leq LC, \text{tradingVolume}, 0), \text{period}) * 100 \end{aligned}$$

5. 中文含义:

LC=昨收

VR=如果收盘价>LC,返回成交量(手),否则返回0的period日累和/如果收盘价
<=LC,返回成交量(手),否则返回0的period日累和*100

6. 输出参数说明:

vr - 容量比率指标，m*1 的 double 矩阵，返回 m 个股票的容量比率指标。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20  
24.00];
```

```
>>v = [10987654 13643303 13245654 20954432 18094333  
15237873;4559934 8959874 7645532 6572344 3455221 6984451];
```

```
>>[vr] = VR(c,v,5) ;
```

```
vr =
```

```
1.4354
```

```
0.9761
```

2.4.13.WAD-威廉多空力度线指标

函数解释：

1. **用途:** 计算威廉多空力度线指标 WAD 指标值。股价一波比一波高，WAD 一波比一波低，两者产生牛背离时，为卖出信号。股价一波比一波低，WAD 一波比一波高，两者产生熊背离时，为买进信号。

2. 函数详细结构:

```
[wad] = WAD(highPrice,lowPrice,ClosePrice)
```

3. 输入参数说明:

highPrice - 股票最高价，m*n 的 double 矩阵，m 个股票 n 日的最高价，各列之间按照时间顺序排列；

lowPrice - 股票最低价，m*n 的 double 矩阵，m 个股票 n 日的最低价，各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致。

4. 公式:

```
MIDA=closePrice-MIN(REF(closePrice,1),lowPrice);  
MIDB=IF(closePrice<REF(closePrice,1),closePrice-  
MAX(REF(closePrice,1),highPrice),0);  
WAD = SUM(IF(closePrice>REF(closePrice,1),MIDA,MIDB),0);
```

5. 中文含义:

MIDA=收盘价-昨收和最低价的较小值

MIDB=如果收盘价<昨收,返回收盘价-昨收和最高价的较大值,否则返回 0

WAD=如果收盘价>昨收,返回 MIDA,否则返回 MIDB 的历史累和

6. 输出参数说明:

wad - 威廉多空力度线指标，m*1 的矩阵。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20  
24.00];
```

```
>>h = [10.11 10.23 10.12 10.52 10.40 10.28;23.98 24.58 24.92 24.60  
24.45 24.15];
```



```
>>l = [9.67 9.75 9.87 9.97 10.12 10.04;23.6 23.9 24.52 24.21 24.20
```

```
23.80];
```

```
>>[wad] = WAD(h,l,c) ;
```

```
wad =
```

```
0.5500
```

```
-0.0200
```

2.4.14.WVAD-威廉变异离散量指标

函数解释：

1. **用途:** 计算威廉变异离散量指标 WVAD 指标值。WVAD 由下往上穿越 0 轴时，视为长期买进信号。WVAD 由上往下穿越 0 轴时，视为长期卖出信号。

2. **函数详细结构:**

```
[wvad] =
```

```
WVAD(openPrice,highPrice,lowPrice,ClosePrice,tradingTradingVolumeum  
e, period)
```

3. **输入参数说明:**

openPrice - 股票开盘价，m*n 的 double 矩阵，m 个股票 n 日的开盘价，

各列之间按照时间顺序排列；

highPrice - 股票最高价，m*n 的 double 矩阵，m 个股票 n 日的最高价，

各列之间按照时间顺序排列，矩阵维数需与 openPrice 一致；

lowPrice - 股票最低价，m*n 的 double 矩阵，m 个股票 n 日的最低价，

各列之间按照时间顺序排列，矩阵维数需与 openPrice 一致；

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，

各列之间按照时间顺序排列，矩阵维数需与 openPrice 一致；

tradingTradingVolumeume - 股票成交量，m*n 的 double 矩阵，m 个股

票 n 日的成交量，各列之间按照时间顺序排列，矩阵维数需与 openPrice 一致；

period - 计算 WVAD 的数值取样周期。

4. 公式:

$$\text{WVAD} = \text{SUM}((\text{closePrice} - \text{openPrice}) / (\text{highPrice} - \text{lowPrice}) * \text{tradingVolume}, \text{period});$$

5. 中文含义:

WVAD=(收盘价-开盘价)/(最高价-最低价)*成交量的 period 日累和

6. 输出参数说明:

wvad - 威廉变异离散量指标，m*n 的矩阵，前 period - 1 列为空值。

7. 示例:

```
>>o = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29  
24.15];
```

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20  
24.00];
```

```
>>h = [10.11 10.23 10.12 10.52 10.40 10.28;23.98 24.58 24.92 24.60  
24.45 24.15];
```

```
>>l = [9.67 9.75 9.87 9.97 10.12 10.04;23.6 23.9 24.52 24.21 24.20 23.80];
```

```
>>v = [10987654 13643303 13245654 20954432 18094333  
15237873;4559934 8959874 7645532 6572344 3455221 6984451];
```

```
>>[wvad] = WVAD(o,h,l,c,v,3);
```

```
wvad =
```

```
1.0e+007 *
```

```
NaN NaN 1.7879 2.6574 1.8329 -0.3806
```

```
NaN NaN 0.8844 0.1200 -0.4524 -0.8282
```

2.5. 压力支撑指标

2.5.1. BBIBOLL-多空布林线

函数解释：

1. **用途:** 计算多空布林线 BBIBOLL 指标值。股价上升穿越布林线上限时，回档机率大。股价下跌穿越布林线下限时，反弹机率大。

2. **函数详细结构:**

`[bbi,upper,lowPriceer] = BBIBOLL(ClosePrice,period,weight)`

3. **输入参数说明:**

`ClosePrice` - 为输入的 `period*n` 维价格数据矩阵，可以同时输入 `period` 个股票 `n` 日的价格数据。

`period` - 为计算多空布林线的的周期数。

`weight` - 计算多空布林线的权重参数。

4. **公式:**

$$\begin{aligned} \text{BBI} &= (\text{MA}(\text{closePrice}, 3) + \text{MA}(\text{closePrice}, 6) + \\ &\quad \text{MA}(\text{closePrice}, 12) + \text{MA}(\text{closePrice}, 24)) / 4; \\ \text{UPPER} &= \text{BBI} + \text{weight} * \text{STD}(\text{BBI}, \text{period}); \\ \text{LOWER} &= \text{BBI} - \text{weight} * \text{STD}(\text{BBI}, \text{period}); \end{aligned}$$

5. **中文含义:**

$\text{BBI} = (\text{收盘价的 } 3 \text{ 日简单移动平均} + \text{收盘价的 } 6 \text{ 日简单移动平均} + \text{收盘价的 } 12 \text{ 日简单移动平均} + \text{收盘价的 } 24 \text{ 日简单移动平均}) / 4$

UPPER=多空指标+ weight *多空指标的 period 日估算标准差

LOWPRICEER=多空指标- weight *多空指标的 period 日估算标准差

6. 输出参数说明:

bbi - period 周期移动平均指标, period*n 的矩阵, 前 period 列为空值;

upper - 多空布林线上限, period*n 的矩阵, 前 period 列为空值;

lowPriceer - 多空布林线下限, period*n 的矩阵, 前 period 列为空值。

7. 示例:

```
[rst, c, r] = getlowPriceFrequencyData('stk_dlyr','stkcd','Trddt',{'sza'},
{'clsprc'},{'Trddt'}, {'2012-01-01'}, {'2012-04-01'});

[bbi,upper,lowPriceer] = BBIBOLL(rst.clsprc,3,12);
```

2.5.2. BOLL-布林线指标

函数解释：

1. **用途:** 计算布林线 BOLL 指标值。股价上升穿越布林线上限时, 回档机率大。

股价下跌穿越布林线下限时, 反弹机率大。

2. **函数详细结构:**

```
[mid,upper,lowPriceer] = BOLL(ClosePrice,period,weight)
```

3. **输入参数说明:**

ClosePrice - 为输入的 $m \times n$ 维价格数据矩阵,可以同时输入 m 个股票 n 日的价格数据。

period - 为计算的周期数, $1 \leq \text{period} \leq \text{weight}$ 。

weight - 计算布林线的权重参数。

4. 公式:

$\text{MID} = \text{MA}(\text{closePrice}, \text{period});$

$\text{UPPER} = \text{MID} + \text{weight} * \text{STD}(\text{closePrice}, \text{period};$

$\text{LOWER} = \text{MID} - \text{weight} * \text{STD}(\text{closePrice}, \text{period});$

5. 中文含义:

MID=收盘价的 period 日简单移动平均

UPPER=MID+ weight *收盘价的 period 日估算标准差

LOWPRICEER=MID- weight *收盘价的 period 日估算标准差

6. 输出参数说明:

mid - period 周期移动平均指标, $m \times n$ 的矩阵, 前 period 列为空值;

upper - 布林线上限, $m \times n$ 的矩阵, 前 period 列为空值;

lowPriceer - 布林线下限, $m \times n$ 的矩阵, 前 period 列为空值。

7. 示例:

```
>>a = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20 24.00];
```

```
>>[mid,upper,lowPriceer] = BOLL(a,3,3)
```

```
mid =
```

	NaN	NaN	10.02	10.15	10.23
10.24					
	NaN	NaN	24.27	24.42	24.38
24.18					
upper =					
	NaN	NaN	10.30	10.68	10.56
10.54					
	NaN	NaN	25.33	24.89	24.99
24.68					
lowPriceer =					
	NaN	NaN	9.74	9.62	9.90
9.94					
	NaN	NaN	23.22	23.94	23.76
23.68					

2.5.3. ENE-轨道线指标

函数解释：

- 用途:** 计算轨道线指标 ENE 指标值。股价上升穿越轨道线上限时，回档机率大；股价下跌穿越轨道线下限时，反弹机率大。
- 函数详细结构:**

$[upper, lowPriceer, ene] = ENE(ClosePrice, period, weight1, weight2)$

3. 输入参数说明:

ClosePrice - 股票收盘价, $m \times n$ 的 double 矩阵, m 个股票 n 日的收盘价, 各列之间按照时间顺序排列;

period - 计算 ENE 的数值取样周期;

weight1 - 计算 ENE 的权重参数 1;

weight2 - 计算 ENE 的权重参数 2。

4. 公式:

$UPPER = (1 + weight1/100) * MA(closePrice, period);$

$LOWER = (1 - weight2/100) * MA(closePrice, period);$

$ENE = (UPPER + LOWER) / 2;$

5. 中文含义:

$UPPER = (1 + weight1/100) * \text{收盘价的 period 日简单移动平均}$

$LOWPRICEER = (1 - weight2/100) * \text{收盘价的 period 日简单移动平均}$

$ENE = (UPPER + LOWPRICEER) / 2$

6. 输出参数说明:

upper - 上轨线指标, $m \times n$ 的矩阵, 前 period 列为空值

lowPriceer - 下轨线指标, $m \times n$ 的矩阵, 前 period 列为空值

ene - 中轨线指标, $m \times n$ 的矩阵, 前 period 列为空值。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20 24.00];
```

```
>>[upper,lowPriceer,ene] = ENE(c,3,3,3) ;
```

upper =

	NaN	NaN	10.32	10.45	10.54
10.54					
	NaN	NaN	25.00	25.15	25.11
24.90					

lowPriceer =

	NaN	NaN	9.72	9.85	9.92
9.93					
	NaN	NaN	23.55	23.68	23.65
23.45					

ene =

	NaN	NaN	10.02	10.15	10.23
10.24					
	NaN	NaN	24.27	24.42	24.38
24.18					

2.5.4. ENV-Envelops 指标

函数解释：

1. **用途:** 计算 Envelops 指标 ENV 指标值。股价突破上轨后再下穿上轨，卖出信号，突破下轨后再上穿下轨，买入信号。

2. **函数详细结构:**

[upper,lowPriceer] = ENV(ClosePrice,period)

3. **输入参数说明:**

ClosePrice - 股票收盘价 ,m*n 的 double 矩阵 ,m 个股票 n 日的收盘价 ,

各列之间按照时间顺序排列 ;

period - 计算 ENV 的数值取样周期 。

4. **公式:**

UPPER = MA(closePrice,period)*1.06;

LOWER = MA(closePrice,period)*0.94

5. **中文含义:**

UPPER =收盘价的 period 日简单移动平均*1.06

LOWPRICEER=收盘价的 period 日简单移动平均*0.94

6. **输出参数说明:**

upper - 上轨线指标，m*n 的矩阵，前 period 列为空值

lowPriceer - 下轨线指标，m*n 的矩阵，前 period 列为空值。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20 24.00];
```

```
>>[upper,lowPriceer] = ENV(c,3)
```

upper =

NaN	NaN	10.62	10.76	10.84
10.85				

NaN	NaN	25.73	25.88	25.84
25.63				

lowPriceer =

NaN	NaN	9.42	9.54	9.62
9.62				

NaN	NaN	22.82	22.95	22.91
22.73				

2.5.5. MIKE-麦克指标

函数解释：

1. **用途:** 计算麦克指标 MIKE 指标值。MIKE 指标共有六条曲线，上方三条压力线，下方三条支撑线。

2. 函数详细结构:

```
[wr,mr,sr,ws,ms,ss] = MIKE(highPrice,lowPrice,ClosePrice,period)
```

3. 输入参数说明:

highPrice - 股票最高价，m*n 的 double 矩阵，m 个股票 n 日的最高价，各列之间按照时间顺序排列；

lowPrice - 股票最低价，m*n 的 double 矩阵，m 个股票 n 日的最低价，各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

period - 计算 MIKE 的数值取样周期。

4. 公式:

$$TYP = (highPrice + lowPrice + closePrice) / 3;$$

$$LL = LLV(lowPrice, period);$$

$$HH = HHV(highPrice, period);$$

$$WR = TYP + (TYP - LL);$$

$$MR = TYP + (HH - LL);$$

$$SR = 2 * HH - LL;$$

$$WS = TYP - (HH - TYP);$$

$$MS = TYP - (HH - LL);$$

$$SS = 2 * LL - HH$$

5. 中文含义:

$$TYP = (\text{最高价} + \text{最低价} + \text{收盘价}) / 3$$

$$LL = M \text{ 日内最低价的最低值}$$

$$HH = M \text{ 日内最高价的最高值}$$

$$WR = TYP + (TYP - LL)$$

$$MR = TYP + (HH - LL)$$

$$SR = 2 * HH - LL$$

$$WS = TYP - (HH - TYP)$$

$$MS = TYP - (HH - LL)$$

$$SS = 2 * LL - HH$$

6. 输出参数说明:

wr - 初级压力, m*n 的矩阵, 前 period 列为空值

mr - 中级压力, m*n 的矩阵, 前 period 列为空值

wr - 强力压力, m*n 的矩阵, 前 period 列为空值

ws - 初级支撑, m*n 的矩阵, 前 period 列为空值

ms - 中级支撑, m*n 的矩阵, 前 period 列为空值

ws - 强力支撑, m*n 的矩阵, 前 period 列为空值。

7. 示例:

```
>>c = [9.94 9.99 10.12 10.34 10.23 10.14;23.90 24.32 24.60 24.33 24.20
24.00];
```

```
>>h = [10.11 10.23 10.12 10.52 10.40 10.28;23.98 24.58 24.92 24.60 24.45
24.15];
```

```
>>l = [9.67 9.75 9.87 9.97 10.12 10.04;23.6 23.9 24.52 24.21 24.20 23.80];
```

```
>>[wr,mr,sr,ws,ms,ss] = MIKE(h,l,c,3) ;
```

```
wr =
```

NaN	NaN	10.4033	10.8033	10.6300	10.3367
NaN	NaN	25.7600	24.8600	24.3667	24.1667

```
mr =
```

NaN	NaN	10.5967	11.0467	10.9000	10.7033
NaN	NaN	26.0000	25.4000	25.0033	24.7833

```
sr =
```

NaN	NaN	10.7900	11.2900	11.1700	11.0700
NaN	NaN	26.2400	25.9400	25.6400	25.4000

```
ws =
```

NaN	NaN	9.8433	10.0333	9.9800	9.7867
NaN	NaN	24.4400	23.8400	23.6467	23.3667

```
ms =
```

NaN	NaN	9.4767	9.5067	9.6000	9.6033
NaN	NaN	23.3600	23.3600	23.5633	23.1833

```
ss =
```

NaN	NaN	9.1100	8.9800	9.2200	9.4200
NaN	NaN	22.2800	22.8800	23.4800	23.0000

2.5.6. XS-薛斯通道指标

函数解释：

1. **用途:** 计算薛斯通道指标 XS 指标值。当股价运行到短周期通道的下轨时是短线买入机会,当股价运行到短周期通道的上轨时是短线的卖出时机。当股价运行到长周期的下轨时是中长线买入时机,而当股价运行到长周期的上轨时是中长线的卖出时机。

2. 函数详细结构:

$[TD1,TD2,TD3,TD4] = XS(highPrice,lowPrice,ClosePrice,weight1,weight2)$

3. 输入参数说明:

highPrice - 股票最高价，m*n 的 double 矩阵，m 个股票 n 日的最高价，各列之间按照时间顺序排列；

lowPrice - 股票最低价，m*n 的 double 矩阵，m 个股票 n 日的最低价，各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

ClosePrice - 股票收盘价，m*n 的 double 矩阵，m 个股票 n 日的收盘价，各列之间按照时间顺序排列，矩阵维数需与 highPrice 一致；

weight1 - 计算薛斯通道指标的权重参数 1

weight2 - 计算薛斯通道指标的权重参数 2 。

4. 公式:

```

AA=MA((2*closePrice+highPrice+lowPrice)/4,5);
CC=ABS((2*closePrice+highPrice+lowPrice)/4-
MA(closePrice,20))/MA(closePrice,20);
DD=DMA(closePrice,CC);
TD1 = AA*weight1/100;
TD2 = AA*(200-weight1)/100;
TD3 = (1+weight2/100)*DD;
TD4 = (1-weight2/100)*DD;

```

5. 中文含义:

AA=(2*收盘价+最高价+最低价)/4 的 5 日简单移动平均

CC=(2*收盘价+最高价+最低价)/4-收盘价的 20 日简单移动平均的绝对值/收盘价的 20 日简单移动平均

DD=以 CC 为权重收盘价的动态移动平均

TD1=AA* weight 1/100

TD2=AA*(200- weight 1)/100

TD3=(1+ weight 2/100)*DD

TD4=(1- weight 2/100)*DD

6. 输出参数说明:

TD1 是计算结果，表示计算得到的 TD1 值。

TD2 是计算结果，表示计算得到的 TD2 值。

TD3 是计算结果，表示计算得到的 TD3 值。

TD4 是计算结果，表示计算得到的 TD4 值。

7. 示例:

```
>>
```

```
[results,rowIndex,columnIndex]=getlowPriceFrequencyData('stk_dlyr','
```



```
stkcd','Trddt',{'szb'}, {'clsprc';'hiprc';'loprc'},{'Trddt'}, {'2012-01-01'},  
{'2012-02-28'});
```

```
>> [TD1,TD2,TD3,TD4] =  
XS(results.hiprc,results.loprc,results.clsprc,5,5);
```

2.6. 大盘指标

2.6.1. ABI-绝对广量指标

函数解释：

1. **用途：**计算绝对广量指标 ABI 指标值。ABI 绝对广量主要用于扫描瞬间极端的多头或空头力道。

2. **函数详细结构：**

[abi] = ABI(advanceNumber,declineNumber)

3. **输入参数说明：**

advanceNumber - 某市场上涨股票数，m*n 的 double 矩阵，可以同时输入 m 个市场或者板块的上涨股票数。

declineNumber - 某市场下跌股票数，m*n 的 double 矩阵，可以同时输入 m 个市场或者板块的下跌股票数，矩阵维数需与 advanceNumber 一致。

4. 公式:

$$ABI = \text{ABS}(\text{advanceNumber} - \text{declineNumber}) / (\text{advanceNumber} + \text{declineNumber});$$

5. 中文含义:

$$ABI = \text{上涨家数} - \text{下跌家数的绝对值} / (\text{上涨家数} + \text{下跌家数})$$

6. 输出参数说明:

abi - 市场绝对广义量指标, m*n 的 double 矩阵, 返回 m 个市场 n 天的绝对广义量指标值

7. 示例:

```
>>a=[112 245 856 345 554 54;34 56 87 34 23 6];
```

```
>>b=[896 1343 543 1345 210 1900;56 23 12 45 60 89]
```

```
>>[abi] = ABI(a,b) ;
```

```
abi =
```

0.78	0.69	0.22	0.59	0.45	0.94
0.24	0.42	0.76	0.14	0.45	0.87

2.6.2. ADL-腾落指数

函数解释：

1. **用途:** 计算腾落指数 ADL 指标值。ADL 与指数顶背离时, 指数向下反转机会大; ADL 与指数底背离时, 指数向上反转机会大。

2. 函数详细结构:

`[adl] = ADL(advanceNumber,declineNumber,period)`

3. 输入参数说明:

`advanceNumber` - 市场上涨股票数, $m*n$ 的 double 矩阵, 可以同时输入 m 个市场或者板块的上涨股票数, 各列之间按照时间顺序排列;

`declineNumber` - 市场下跌股票数, $m*n$ 的 double 矩阵, 可以同时输入 m 个市场或者板块的下跌股票数, 矩阵维数需与 `advanceNumber` 一致;

`period` - 计算腾落指标的数值取样周期。

4. 公式:

$$ADL = \text{SUM}(\text{advanceNumber} - \text{declineNumber}, \text{period});$$

5. 中文含义:

ADL=上涨家数-下跌家数的历史累和

6. 输出参数说明:

`adl` - 市场腾落指标, $m*n$ 的 double 矩阵, 返回 m 个市场的腾落指标值。

7. 示例:

```
>>a=[112 245 856 345 554 54;34 56 87 34 23 6];
```

```
>>b=[896 1343 543 1345 210 1900;56 23 12 45 60 89];
```

```
>>[adl] = ADL(a,b,5);
```

```

adl =
NaN      NaN      NaN      NaN      -2225.00
-3287.00
NaN      NaN      NaN      NaN      38.00
-23.00

```

2.6.3. ADR-涨跌比率指标

函数解释：

1. **用途：**计算涨跌比率 ADR 指标值。ADR 一般常态分于 0.5 ~ 1.5 的间；
ADR>1.5 ,大盘回档机会大 ;ADR<0.65 ,大盘反弹机会大 ;ADR<0.3 或 0.5 ,
容易形成底部。

2. 函数详细结构:

[adr] = ADR(advanceNumber,declineNumber,period)

3. 输入参数说明:

advanceNumber - 某市场上涨股票数 ,m*n 的 double 矩阵 ,可以同时输入

m 个市场或者板块的上涨股票数 ,各列之间按照时间顺序排列 ;

declineNumber - 某市场下跌股票数 , m*n 的 double 矩阵 ,可以同时输入

m 个市场或者板块的下跌股票数 , 矩阵维数需与 advanceNumber 一致 ;

period - 计算涨跌比率的数值取样周期 ,当样本区间不足时 ,不予计

算。

4. 公式:

$$ADR = \text{SUM}(\text{advanceNumber}, \text{period}) / \text{SUM}(\text{declineNumber}, \text{period});$$

5. 中文含义:

ADR=上涨家数的 period 日累和/下跌家数的 period 日累和

6. 输出参数说明:

adr - 市场涨跌比率，m*n 的 double 矩阵，返回 m 个市场的涨跌比率值。

7. 示例:

```
>>a=[112 245 856 345 554 54;34 56 87 34 23 6];
```

```
>>b=[896 1343 543 1345 210 1900;56 23 12 45 60 89];
```

```
>>[adr] = ADR(a,b,5);
```

adr =

NaN	NaN	NaN	NaN	0.49
0.38				
NaN	NaN	NaN	NaN	1.19
0.90				

2.6.4. ARMS-阿姆氏指标

函数解释：

1. **用途:** 计算阿姆氏指标 ARMS 指标值。当该指标为 1 时，说明市场做多和做空力量处于势均力敌状态；当指标高于 1 时，表明做空力量占主导，该指标越大，说明做空力量越强；当该指标低于 1 时，反映做多力量走强，越接近 0 多头势力越大。

2. **函数详细结构:**

`[arms] = ARMS(advanceNumber,declineNumber,period)`

3. **输入参数说明:**

`advanceNumber` - 某市场上涨股票数， $m \times n$ 的 double 矩阵，可以同时输入 m 个市场或者板块的上涨股票数，各列之间按照时间顺序排列；

`declineNumber` - 某市场下跌股票数， $m \times n$ 的 double 矩阵，可以同时输入 m 个市场或者板块的下跌股票数，矩阵维数需与 `advanceNumber` 一致；

`period` - 计算阿姆氏指标的数值取样周期，当样本区间不足时，不予计算。

4. **公式:**

$$ARMS = EMA(advanceNumber/declineNumber,period);$$

5. **中文含义:**

$ARMS = \text{上涨家数} / \text{下跌家数}$ 的 `period` 日指数移动平均

6. **输出参数说明:**

`arms` - 市场阿姆氏指标， $m \times n$ 的 double 矩阵，返回 m 个市场的阿姆氏指

标。

7. 示例:

```
>>a=[112 245 856 345 554 54;34 56 87 34 23 6];
```

```
>>b=[896 1343 543 1345 210 1900;56 23 12 45 60 89];
```

```
>>[arms] = ARMS(a,b,3);
```

```
arms =
```

NaN	NaN	1.88	2.02	4.47
2.92				
NaN	NaN	10.29	10.44	8.39
1.21				

2.6.5. BTI-Breadth Trust 指标

函数解释：

1. **用途:** 计算 Breadth Trust 指标值。上涨家数除以上涨家数与下跌家数的差，求这个比值的 N 日移动平均。

2. 函数详细结构:

```
[bti] = BTI(advanceNumber,declineNumber,period)
```

3. 输入参数说明:

advanceNumber - 某市场上涨股票数，m*n 的 double 矩阵，可以同时输入

m 个市场或者板块的上涨股票数，各列之间按照时间顺序排列；

declineNumber - 某市场下跌股票数，m*n 的 double 矩阵，可以同时输入

m 个市场或者板块的下跌股票数，矩阵维数需与 advanceNumber 一致；

period - 计算广量冲力指标的数值取样周期，当样本区间不足时，不予计算。

4. 公式:

$$BTI = MA(advanceNumber/(advanceNumber-declineNumber),period);$$

5. 中文含义:

BTI=上涨家数/(上涨家数-下跌家数)的 period 日简单移动平均

6. 输出参数说明:

bti - 市场广量冲力指标，m*n 的 double 矩阵，返回 m 个市场的广量冲力指标。

7. 示例:

```
>>a=[112 245 856 345 554 54;34 56 87 34 23 6];

>>b=[896 1343 543 1345 210 1900;56 23 12 45 60 89];

>>[bti] = BTI(a,b,3);

bti =

NaN          NaN          0.79          0.72          1.33
```


	NaN	NaN	0.44	-0.08	-0.85
-1.26					

2.6.6. CHAIKIN-佳庆线指标

函数解释：

1. **用途：**计算佳庆线指标 CHAIKIN 指标值。CHAIKIN 曲线产生急促的凸起时，代表行情即将向上或向下反转。股价>90 天平均线，CHAIKIN 由负转正时，买进。股价<90 天平均线，CHAIKIN 由正转负时，卖出。

2. 函数详细结构:

[chaikin]=CHAIKIN(advanceNumber,declineNumber,period1,period2)

3. 输入参数说明:

advanceNumber - 某市场上涨股票数，m*n 的 double 矩阵，可以同时输入 m 个市场或者板块的上涨股票数，各列之间按照时间顺序排列；

declineNumber - 某市场下跌股票数，m*n 的 double 矩阵，可以同时输入 m 个市场或者板块的下跌股票数，矩阵维数需与 advanceNumber 一致；

period1 - 计算佳庆线指标的数值取样周期 1，当样本区间不足时，不予计算；

period2 - 计算佳庆线指标的数值取样周期 2，当样本区间不足时，不予计算。

4. 公式:

$$ADL = \text{SUM}(\text{advanceNumber} - \text{declineNumber}, 0);$$
$$\text{CHAIKIN} = \text{MA}(ADL, \text{period1}) - \text{MA}(ADL, \text{period2});$$

5. 中文含义:

ADL=上涨家数-下跌家数的历史累和

CHAIKIN=ADL 的 period 1 日简单移动平均-ADL 的 period 2 日简单移动平均

6. 输出参数说明:

chaikin - 市场佳庆线指标, m*n 的 double 矩阵, 返回 m 个市场的佳庆线指标。

7. 示例:

```
>>a=[112 245 856 345 554 54;34 56 87 34 23 6];
```

```
>>b=[896 1343 543 1345 210 1900;56 23 12 45 60 89];
```

```
>>[chaikin] = CHAIKIN(a,b,3,2);
```

chaikin =

NaN	NaN	-130.5000	-251.5000	213.6667	-83.0000
NaN	NaN	-25.3333	0.3333	33.0000	16.3333

2.6.7. MCO-麦克连指标

函数解释：

1. **用途:** 计算麦克连指标 MCO (McClellan Oscillator) 指标值。MCO 到达 5 以上属超买区，当由上向下再跌破 5 时，则短线卖出；当 MCO 跌至-5 以下时属超卖区，当由下向上再突破-5 时，则短线买入。

2. **函数详细结构:**

[mco] = MCO(advanceNumber,declineNumber,period1,period2)

3. **输入参数说明:**

advanceNumber 为输入的大盘上涨家数序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

declineNumber 为输入的大盘下跌家数序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

period1 为周期参数， $\text{period1} \geq 1$ ，整数。

period2 为周期参数， $\text{period2} \geq 1$ ，整数。

4. **公式:**

$$\text{MCO} = \text{EMA}(\text{advanceNumber} - \text{declineNumber}, \text{period1}) / 10 - \text{EMA}(\text{advanceNumber} - \text{declineNumber}, \text{period2}) / 20;$$

5. **中文含义:**

MCO=上涨家数-下跌家数的 period 1 日指数移动平均/10-上涨家数-下跌家数的 period 2 日指数移动平均/20

6. **输出参数说明:**

mco - 市场麦克连指标 ,m*n的double矩阵 ,返回m个市场的麦克连指标值。

7. 示例:

```
>>a=[112 245 856 345 554 54;34 56 87 34 23 6];
```

```
>>b=[896 1343 543 1345 210 1900;56 23 12 45 60 89];
```

```
>>[mco] = MCO(a,b,5,5) ;
```

```
mco =
```

```
-39.2000 -44.4333 -24.4056 -32.9370 -16.2247 -41.5831
```

```
-1.1000 -0.1833 1.1278 0.5685 -0.2377 -1.5418
```

2.6.8. OBOS-超买超卖指标

函数解释：

1. **用途:** 计算大盘超买超卖指标 OBOS 指标值。指标上升至+80 时为超买，下降至-80 时为超卖 指标若超越+100 或-100 时 应等待其产生背离才可确认。

2. 函数详细结构:

```
[obos] = OBOS(advanceNumber,declineNumber,period)
```

3. 输入参数说明:

advanceNumber - 某市场上涨股票数, $m \times n$ 的 double 矩阵, 可以同时输入

m 个市场或者板块的上涨股票数, 各列之间按照时间顺序排列;

declineNumber - 某市场下跌股票数, $m \times n$ 的 double 矩阵, 可以同时输入

m 个市场或者板块的下跌股票数, 矩阵维数需与 advanceNumber 一致;

period - 计算超买超卖指标的数值取样周期。

4. 公式:

$$OBOS = MA(\text{advanceNumber} - \text{declineNumber}, \text{period});$$

5. 中文含义:

OBOS = 上涨家数 - 下跌家数的 period 日简单移动平均

6. 输出参数说明:

obos - 市场超买超卖指标, $m \times n$ 的 double 矩阵, 返回 m 个市场的超买超卖指标值, 前 period-1 列为空。

7. 示例:

```
>>a = [112 245 856 345 554 54;34 56 87 34 23 6];
```

```
>>b = [896 1343 543 1345 210 1900;56 23 12 45 60 89];
```

```
>>[obos] = OBOS(a,b,3);
```

obos =

NaN NaN -523.0000 -595.0000 -114.3333 -834.0000

NaN NaN 28.6667 32.3333 9.0000 -43.6667

2.6.9. STIX-指数平滑广量指标

函数解释：

1. **用途:** 计算指数平滑广量指标 STIX 指标值。常态行情时，STIX 一般波动于 45~56 的间，强势行情波动于 42~58 之间；指标上升至 56~58 间时，短线应卖出；指标下降至 42~45 间时，短线应买进。

2. 函数详细结构:

[stix] = STIX(advanceNumber,declineNumber,period)

3. 输入参数说明:

advanceNumber - 某市场上涨股票数，m*n 的 double 矩阵，可以同时输入

m 个市场或者板块的上涨股票数，各列之间按照时间顺序排列；

declineNumber - 某市场下跌股票数，m*n 的 double 矩阵，可以同时输入

m 个市场或者板块的下跌股票数，矩阵维数需与 advanceNumber 一致；

period - 计算涨跌比率的数值取样周期，当样本区间不足时，不予计算。

4. 公式:

$$STIX = EMA(\text{advanceNumber}/(\text{advanceNumber} + \text{declineNumber}) * 100, \text{period});$$

5. 中文含义:

STIX = 上涨家数/(上涨家数+下跌家数)*100 的 period 日指数移动平均

6. 输出参数说明:

stix - 市场涨跌比率 ,m*n 的 double 矩阵 ,返回 m 个市场的指数平滑广量。

7. 示例:

```
>>a=[112 245 856 345 554 54;34 56 87 34 23 6];
```

```
>>b=[896 1343 543 1345 210 1900;56 23 12 45 60 89];
```

```
>>[stix] = STIX(a,b,5) ;
```

stix =

11.1111 12.5501 28.7623 25.9796 41.4908 28.5817

37.7778 48.8139 61.8355 55.5697 46.2834 32.9609 ;

2.7. 摆动指标

2.7.1. AD-累积分布指标

函数解释：

1. **用途:** 计算累积分布指标 AD 指标值。ACD 线下降，而股价会上升时，为卖出信号。ACD 线上升，而股价下降时，为买进信号。

2. **函数详细结构:**

[ad]= AD(highPrice, lowPrice, closePrice, TradingVolume)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 列数值矩阵。

lowPrice 为输入的最低价序列，为 M*N 列数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

TradingVolume 为输入的成交量序列，为 M*N 列数值矩阵。

4. **公式:**

$$AD = \text{SUM}(((\text{closePrice} - \text{lowPrice}) - (\text{highPrice} - \text{closePrice})) / (\text{highPrice} - \text{lowPrice}) * \text{tradingVolumn}, 0);$$

5. **中文含义:**

$AD = ((\text{收盘价} - \text{最低价}) - (\text{最高价} - \text{收盘价})) / (\text{最高价} - \text{最低价}) * \text{成交量的历史累和}$

6. **输出参数说明:**

ad-是计算结果矩阵，与输入的价格序列为同型矩阵，表示集散量值矩阵。

7. **示例**

详细例子：


```

>>highPrice = [ 20.68  20.30  19.77  19.98  19.50  19.10  19.31  19.30
19.73  19.70;  5.51  5.42  5.48  5.40  5.40  5.26  5.28  5.26
5.29  5.30 ];

>>lowPrice  = [ 20.11  19.10  19.10  19.00  19.06  18.65  18.55  18.76
19.10  19.25;  5.39  5.29  5.30  5.20  5.22  5.13  5.16  5.12
5.03  5.20 ];

>>closePrice = [ 20.40  19.50  19.48  19.15  19.20  18.86  18.70  19.10
19.71  19.25;  5.42  5.39  5.40  5.29  5.26  5.25  5.17  5.16
5.29  5.27 ];

>>TradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70  4314.14
1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10  8686.95
3997.83  8000.68  9106.48  2638.03  1360.69  5797.05  1449.55];

>> [ad]=AD(highPrice, lowPrice, closePrice, TradingVolume)

ad =

18.71          3.26      1101.13      515.25      -429.73      -717.34
-1817.99     -1440.67      6700.32      1201.72
-4809.49     -636.90      328.32      -71.46      -4516.29      3189.19
990.84       407.68      6204.73      6784.55

```

2.7.2. ATR-真实波幅指标

函数解释：

1. **用途:** 计算真实波幅指标 ATR 指标值。ATR 过高, 并且股价上涨过快时, 则卖出。ATR 过低, 表明股价已经进入窄幅整理行情中, 随时将面临突破。

2. **函数详细结构:**

[atr]= ATR(highPrice, lowPrice, closePrice, period)

3. **输入参数说明:**

highPrice 为输入的最高价序列, 为 M*N 列数值矩阵。

lowPrice 为输入的最低价序列, 为 M*N 列数值矩阵。

closePrice 为输入的收盘价序列, 为 M*N 列数值矩阵。

period 为周期参数, $1 \leq \text{period} < N$, 整数。

4. **公式:**

$$TR = \text{MAX}(\text{MAX}((\text{highPrice} - \text{lowPrice}), \text{ABS}(\text{REF}(\text{closePrice}, 1) - \text{highPrice})), \text{ABS}(\text{REF}(\text{closePrice}, 1) - \text{lowPrice}));$$
$$ATR = \text{MA}(TR, \text{period})$$

5. **中文含义:**

TR=(最高价-最低价)和昨收-最高价的绝对值的较大值和昨收-最低价的绝对值的较大值

ATR=真实波幅的 period 日简单移动平均

6. **输出参数说明:**

atr-是计算结果矩阵, 与输入的价格序列为同型矩阵, 表示真实波幅指标值矩阵。

7. **示例**

```
>>highPrice = [ 20.68  20.30  19.77  19.98   19.50   19.10  19.31   19.30
19.73   19.70;  5.51   5.42   5.48   5.40   5.40   5.26   5.28   5.26
5.29   5.30  ];

>>lowPrice  = [ 20.11  19.10  19.10  19.00   19.06   18.65  18.55   18.76
19.10  19.25;  5.39   5.29   5.30   5.20   5.22   5.13   5.16   5.12
5.03   5.20  ];

>>closePrice = [ 20.40  19.50  19.48  19.15   19.20   18.86  18.70   19.10
19.71  19.25;  5.42   5.39   5.40   5.29   5.26   5.25   5.17   5.16
5.29   5.27  ];

>> [atr]=ATR(highPrice, lowPrice, closePrice, 5)

atr =

    NaN         NaN         NaN         NaN         0.68
    0.79         0.68         0.67         0.60         0.60
    NaN         NaN         NaN         NaN         0.14
    0.16         0.16         0.15         0.17         0.15
```

2.7.3. MI-动量指标

函数解释：

1. **用途:** 计算动量指标 MI 指标值。动量指标表示的是股票价格的涨跌速度，如果股票价格不断上升则动量指标继续向上发展，就说明股票上升的速度在加快。反之，如果股票价格始终在下降，则动量指标始终保持在 0 线的下方。如果动

量指标继续向下发展，就说明股票价格下降的速度在加快。

2. 函数详细结构:

[mi]= MI(closePrice, period)

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=2。

period 为周期参数，1 <= period < N，整数。

4. 公式:

A = closePrice-REF(closePrice,period);

MI = SMA(A,period,1);

5. 中文含义:

A=收盘价-period 日前的(常量和变涨跌幅)

MI=A 的 period 日[1 日权重]移动平均

6. 输出参数说明:

mi-是计算结果矩阵，与输入的价格序列为同型矩阵，表示动量指标值矩阵。

7. 示例

```
>> closePrice = [ 20.40  19.50  19.48  19.15  19.20  18.86  18.70  19.10  
19.71  19.25;  5.42  5.39  5.40  5.29  5.26  5.25  5.17  5.16  
5.29  5.27  ];
```

```
>> [mi] = MI(closePrice, 3)
```

mi =

NaN	NaN	NaN	-1.25	-0.93	-0.83
-0.70	-0.50	-0.05	0.15		

NaN	NaN	NaN	-0.13	-0.13	-0.14
-0.13	-0.12	-0.07	-0.01		

2.7.4. MICD-异同离差动力指数

函数解释：

1. **用途:** 计算异同离差动力指数 MICD 指标值。如果股票价格向上攀升则该指数不断向上发展。反之如果股票价格一直向下，则该指数始终保持在 0 线的下方。

2. **函数详细结构:**

[micd]= MICD(closePrice, period1, period2, period3)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=3。

period1 为周期参数 1， period1 >=1，整数。

period2 为周期参数 2， 1 =< period2 =< N-2，整数。

period3 为周期参数 3， 1 =< period3 =< N-2，整数。

4. **公式:**

MI = closePrice-REF(closePrice,1);

AMI = SMA(MI,period1,1);

DIF = MA(REF(AMI,1),period2)-MA(REF(AMI,1),period3);

MICD = SMA(DIF,10,1)

5. **中文含义:**

MI=收盘价-昨收

AMI=MI 的 period1 日[1 日权重]移动平均

DIF=昨日 AMI 的 period2 日简单移动平均-昨日 AMI 的 period3 日简单移动平均

MICD=DIF 的 10 日[1 日权重]移动平均

6. 输出参数说明:

micd-是计算结果矩阵，与输入的价格序列为同型矩阵，表示异同离差动力指数数值矩阵。

7. 示例

```
>> closePrice = [ 20.40  19.50  19.48  19.15  19.20  18.86  18.70  19.10
19.71  19.25;  5.42  5.39  5.40  5.29  5.26  5.25  5.17  5.16
5.29  5.27  ];

>> [micd]=MICD(closePrice, 1, 2, 3)

micd =

NaN          NaN          NaN          NaN          0.24          0.21
0.20          0.17          0.17          0.17
NaN          NaN          NaN          NaN          -0.01          -0.01
-0.00         -0.00         -0.01         -0.00
```

2.7.5. RC-变化率指标

函数解释：

1. **用途:** 计算变化率指标 RC 指标值。如果 RC 指数在 100 以上且不断上升，说

明价格上升的速度在加快。反之如果 RC 指数在 100 以下且不断下降，说明价格下降的速度在加快。

2. 函数详细结构:

`[rc]= RC(closePrice, period)`

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵， $N \geq 2$ 。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. 公式:

$$RC = \text{closePrice} / \text{REF}(\text{closePrice}, \text{period});$$

5. 中文含义:

RC=收盘价/period 日前的收盘价

6. 输出参数说明:

rc-是计算结果矩阵，与输入的价格序列为同型矩阵，表示变化率指数值矩阵。

7. 示例

```
>> closePrice = [ 20.40  19.50  19.48  19.15  19.20  18.86  18.70  19.10  
19.71  19.25;  5.42  5.39  5.40  5.29  5.26  5.25  5.17  5.16  
5.29  5.27 ];
```

```
>> [rc]=RC(closePrice,3)
```

rc =

NaN	NaN	NaN	0.94	0.98	0.97
0.98	0.99	1.05	1.03		

NaN	NaN	NaN	0.98	0.98	0.97
0.98	0.98	1.01	1.02		

2.7.6. RCCD-异同离差变化率指数

函数解释：

1. **用途：**计算异同离差变化率指数 RCCD 指标值。如果 RCCD 指数不断上升，说明价格上升的速度在加快。反之如果 RCCD 指数不断下降，说明价格下降的速度在加快。

2. **函数详细结构：**

[rccd]= RCCD(closePrice, period1, period2, period3)

3. **输入参数说明：**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=4。

period1 为周期参数 1， period1 >=1，整数。

period2 为周期参数 2， period2 >=1，整数。period1+period2 =< N-2

period3 为周期参数 3， period3>=1，整数。period1+period3 =< N-2

4. **公式：**

$RC = \text{closePrice} / \text{REF}(\text{closePrice}, \text{period1})$;

$ARC = \text{SMA}(\text{REF}(RC, 1), \text{period1}, 1)$;

$DIF = \text{MA}(\text{REF}(ARC, 1), \text{period2}) - \text{MA}(\text{REF}(ARC, 1), \text{period3})$;

$RCCD = \text{SMA}(DIF, M1, 1)$

5. **中文含义：**

RC=收盘价/period1 日前的收盘价

ARC=昨日 RC 的 period1 日[1 日权重]移动平均

DIF=昨日 ARC 的 period2 日简单移动平均-昨日 ARC 的 period3 日简单移动

平均

RCCD=DIF 的 period1 日[1 日权重]移动平均

6. 输出参数说明:

rccd-是计算结果矩阵，与输入的价格序列为同型矩阵，表示异同离差变化

率指数值矩阵。

详细例子：

```
>> closePrice = [ 20.40  19.50  19.48  19.15  19.20  18.86  18.70  19.10  
19.71  19.25;  5.42  5.39  5.40  5.29  5.26  5.25  5.17  5.16  
5.29  5.27  ];
```

```
>> [rccd]=RCCD(closePrice, 1, 2, 3)
```

rccd =

NaN	NaN	NaN	NaN	NaN
NaN	0.01	0.01	0.00	0.01
NaN	NaN	NaN	NaN	NaN
NaN	-0.01	-0.00	-0.00	-0.00

2.7.7. SRMI-动量修正指标

函数解释：

1. **用途:** 计算动量修正指标 SRMI 指标值。SRMI 由上向下跌破中心线时为卖出

时机，相反，SRMI 由下向上突破中心线时为买进时机。

2. 函数详细结构:

[srmi]= SRMI(closePrice, period)

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=2。

period 为周期参数，1 ≤ period < N，整数。

4. 公式:

$$SRMI = IF(closePrice < REF(closePrice, period), (closePrice - REF(closePrice, period)) / REF(closePrice, period), IF(closePrice = REF(closePrice, period), 0, (closePrice - REF(closePrice, period)) / closePrice))$$

5. 中文含义:

SRMI=如果收盘价<period 日前的收盘价返回(收盘价-period 日前的收盘价
/period 日前的收盘价否则返回如果收盘价=period 日前的收盘价返回 0,否则
返回(收盘价-period 日前的收盘价/收盘价

6. 输出参数说明:

srmi-是计算结果矩阵，与输入的价格序列为同型矩阵，表示动量修正指标
值矩阵。

详细例子：

```
>> closePrice = [ 20.40  19.50  19.48  19.15  19.20  18.86  18.70  19.10  
19.71  19.25;  5.42  5.39  5.40  5.29  5.26  5.25  5.17  5.16  
5.29  5.27  ];
```

```
>> [srmi]=SRMI(closePrice,3)

srmi =

NaN          NaN          NaN          -0.06          -0.02          -0.03
-0.02        -0.01         0.04         0.03
NaN          NaN          NaN          -0.02          -0.02          -0.03
-0.02        -0.02         0.01         0.02
```

2.8. 统计指标

2.8.1. DDBS-大单笔数指标

函数解释：

1. **用途:** 计算大单笔数指标 DDBS 指标值。统计大于 M 手的买卖单的笔数，说明成交活跃的程度。

2. **函数详细结构:**

[ddbbs]= DDBS(TradingVolume, threshold)

3. **输入参数说明:**

TradingVolume 为输入的成交量序列，为 M*N 列数值矩阵。

threshold 为阈值参数，表示成交量的最低阈值，threshold >=1，整数。

4. **公式:**

$$DDBS = COUNT(tradingVolume > threshold, 0)$$

5. **中文含义:**

DDBS=统计满足成交量>threshold 的周期数

6. 输出参数说明:

ddbbs-是计算结果矩阵，与输入的成交量序列为同型矩阵，表示今日大单笔数值矩阵。

7. 示例

```
>> TradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70  4314.14
1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10  8686.95
3997.83  8000.68  9106.48  2638.03  1360.69  5797.05  1449.55];

>> [ddbbs]=DDBS(TradingVolume,4000)

ddbbs =

0          0          1.00          1.00          1.00          2.00
2.00        2.00          3.00          4.00
1.00        2.00          3.00          3.00          4.00          5.00
5.00        5.00          6.00          6.00
```

2.8.2. DDTQ-大单提取

函数解释：

1. **用途:** 计算大单提取 DDTQ 指标值。提取成交量序列中的大于 M 的大手笔成

交，并对其求和。

2. 函数详细结构:

[ddtq]= DDTQ(TradingVolume, threshold)

3. 输入参数说明:

TradingVolume 为输入的成交量序列，为 M*N 列数值矩阵。

threshold 为阈值参数， threshold >=1，整数，表示成交量的最低阈值。

4. 公式:

$$DDTQ = \text{SUM}(\text{IF}(\text{tradingVolume} \geq \text{threshold}, \text{tradingVolumn}, 0), 0);$$

5. 中文含义:

DDTQ=如果成交量>=threshold,返回成交量,否则返回 0 的历史累和

6. 输出参数说明:

ddtq-是计算结果矩阵，与输入的成交量序列为同型矩阵，表示大单提取值矩阵。

7. 示例:

```
>> TradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70  4314.14
1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10  8686.95
3997.83  8000.68  9106.48  2638.03  1360.69  5797.05  1449.55];

>> [ddtq]=DDTQ(TradingVolume,3)

ddtq =

0          0      8173.03      8173.03      10771.73      15085.87
15085.87    15085.87    23778.79    29277.39
9618.98    17368.08    26055.03    30052.86    38053.54
```

47160.02 49798.05 49798.05 55595.10 55595.10

2.8.3. DPTB-大盘同步指标

函数解释：

1. **用途：**计算大盘同步指标 DPTB 指标值。统计比例越接近 1，说明该股和大盘走势越一致。

2. **函数详细结构：**

[dptb]= DPTB(openPrice, closePrice, indexopenPrice, indexClosePrice, period)

3. **输入参数说明：**

openPrice 为输入的开盘价序列，为 M*N 列数值矩阵。

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

indexopenPrice 为输入的大盘开盘价序列，为 M*N 列数值矩阵。

indexClosePrice 为输入的大盘收盘价序列，为 M*N 列数值矩阵。

period 为周期参数， $1 \leq \text{period} \leq N$ ，整数。

4. **公式：**

$$\text{DPTB} = \text{COUNT}((\text{closePrice} > \text{OPEN AND INDEXC} > \text{INDEXO}) \text{ OR } (\text{closePrice} < \text{OPEN AND INDEXC} < \text{INDEXO}), 1, 1, \text{period}) / \text{period};$$

5. **中文含义：**

DPTB=统计 period 日中满足(收盘价>开盘价 并且 大盘收盘价>大盘开盘价)

或者(收盘价<开盘价 并且 大盘收盘价<大盘开盘价)的天数/period

6. **输出参数说明：**

dptb-是计算结果矩阵，与输入的价格序列为同型矩阵，表示大盘同步指标值矩阵。

7. 示例:

```
>> openPrice = [ 20.27 20.20 19.10 19.20 19.06 19.08 19.20 18.76 19.10 19.58;
5.51 5.41 5.45 5.38 5.30 5.20 5.24 5.13 5.19 5.30]

>> closePrice = [ 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71 19.25 ;
5.42 5.39 5.40 5.29 5.26 5.25 5.17 5.16 5.29 5.27]

>> indexopenPrice = [2044.83 2060.61 2038.87 2040.67 2062.99 2128.07
2127.52 2129.69 2123.84 2127.39 ; 2044.83 2060.61 2038.87 2040.67 2062.99
2128.07 2127.52 2129.69 2123.84 2127.39];

>> indexClosePrice = [2059.15 2043.65 2037.68 2051.92 2127.76 2134.89
2120.55 2126.55 2110.38 2123.85; 2059.15 2043.65 2037.68 2051.92
2127.76 2134.89 2120.55 2126.55 2110.38 2123.85];

>> [dptb] = DPTB(openPrice,closePrice,indexopenPrice,indexClosePrice,3)

dptb =

NaN NaN 0.67 0.33 0.33 0.33
0.67 0.33 0.33 0.33
NaN NaN 0.67 0.67 0.33 0.33
0.67 0.67 0.33 0.33
```

2.8.4. DSBC-大手笔差

函数解释：

1. **用途:** 计算大手笔差提取 DSBC 指标值。求当日大手笔的买单与大手笔的卖单之差，该差值大于 0，则说明多方占据优势，形势有利于多方，可看涨；该值越大多方优势越明显。

2. **函数详细结构:**

[dsbc]= DSBC(BuyTradingVolume, SellTradingVolume, threshold)

3. **输入参数说明:**

BuyTradingVolume 为输入的主动性买单成交量序列，为 M*N 列数值矩阵，

SellTradingVolume 为输入的主动性卖单成交量序列，为 M*N 列数值矩阵。

threshold 为阈值参数， threshold >=1，整数，表示成交量的最低阈值。

4. **公式:**

$$DSBC = \text{SUM}(\text{IF}(\text{BUYVOL} \geq \text{threshold}, \text{BUYVOL}, 0), 0) - \text{SUM}(\text{IF}(\text{SELLVOL} \geq \text{threshold}, \text{SELLVOL}, 0), 0);$$

5. **中文含义:**

DSBC=如果主动性买单成交量 >=threshold,返回主动性买单成交量,否则返回 0 的历史累和-如果主动性卖单成交量 >=threshold,返回主动性卖单成交量,

否则返回 0 的历史累和

6. **输出参数说明:**

dsbc-是计算结果矩阵，与输入的主动性买单成交量序列为同型矩阵，表示

主动性买卖大单的差值矩阵。

详细例子：

```
>> buyTradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70
4314.14  1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10
8686.95  3997.83  8000.68  9106.48  2638.03  1360.69  5797.05
1449.55];

>> sellTradingVolume = [ 4172.67  9027.16  4908.64  3377.19  692.47
7802.52  2416.91  964.55  9420.51  5752.09; 496.54  9447.87  4892.53
9000.54  1112.03  3897.39  4039.12  1319.73  9561.35  597.80];

>> [dsbc]=DSBC(buyTradingVolume,sellTradingVolume,3000)

dsbc =

-4172.67    -13199.83    -9935.44    -13312.64    -17005.10
-20493.49    -20493.49    -20493.49    -21221.07    -21474.56
9618.98      7920.21     11714.63      6711.92     14712.60
19921.69     15882.57     15882.57     12118.28     12118.28
```

2.8.5. JDAMOUNT-阶段内总成交额

函数解释：

1. **用途：**计算阶段内总成交额 JDAMOUNT 指标值。统计一段时间内的总成交额。

2. **函数详细结构：**

[jdamount]= JDAMOUNT(Amount, period)

3. 输入参数说明:

Amount 为输入的成交额序列，为 M*N 列数值矩阵。

period 为周期参数， $1 \leq \text{period} \leq N$ ，整数。

4. 公式:

$$JDAMOUNT = \text{SUM}(AMOUNT, \text{period});$$

5. 中文含义:

JDAMOUNT=成交额的 period 周期累和

6. 输出参数说明:

jdamount-是计算结果矩阵，与输入的成交额序列为同型矩阵，表示阶段

内总成交额矩阵。

7. 示例

```
>> TradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70  4314.14
1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10  8686.95
3997.83  8000.68  9106.48  2638.03  1360.69  5797.05  1449.55];

>> [jdamount]=JDAMOUNT(TradingVolume,3)

jdamount =

NaN          NaN      9285.90      9063.73      11616.09      7757.20
8731.31      7588.00      11966.78      15646.91
NaN          NaN      26055.03      20433.88      20685.46      21104.99
19745.19      13105.20      9795.77      8607.29
```

2.8.6. JDHSL-阶段内换手率

函数解释：

1. **用途:** 计算阶段内换手率 JDHSL 指标值。统计一段时间内的总换手率。

2. **函数详细结构:**

[jdhsl]= JDHSL(TradingVolume, Capital, period)

3. **输入参数说明:**

TradingVolume 为输入的成交量序列，为 M*N 列数值矩阵。

Capital 为输入的流通股本序列，为 M*N 列数值矩阵。

period 为周期参数， $1 \leq \text{period} \leq N$ ，整数。

4. **公式:**

$$JDHSL = \text{SUM}(\text{tradingVolume}, \text{period}) / \text{CAPITAL};$$

5. **中文含义:**

JDHSL=成交量的 period 周期累和/当前流通股本

6. **输出参数说明:**

jdhsl-是计算结果矩阵，与输入的成交量序列为同型矩阵，表示阶段内换手

率矩阵。

7. **示例**

```
>> TradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70  4314.14
1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10  8686.95
```

```

3997.83    8000.68    9106.48    2638.03    1360.69    5797.05    1449.55];

>> capital = [ 6885622.00    6885622.00    6885622.00    6885622.00

6885622.00    6885622.00    6885622.00    6885622.00    6885622.00

6885622.00; 97464642.00    97464642.00    97464642.00    97464642.00

97464642.00    97464642.00    97464642.00    97464642.00    97464642.00

97464642.00]

>> [jdhs1]=JDHSL(TradingVolume,capital,3)

jdhs1 =

NaN        NaN    0.0013    0.0013    0.0017    0.0011    0.0013    0.0011

0.0017    0.0023

NaN        NaN    0.0003    0.0002    0.0002    0.0002    0.0002    0.0001

0.0001    0.0001

```

2.8.7. JDVOL-阶段内总成交量

函数解释：

1. **用途:** 计算阶段内总成交量 JDVOL 指标值。统计一段时间内的总成交量。
2. **函数详细结构:**

```
[jdTradingVolume]=JDVOL(TradingVolume, period)
```

3. **输入参数说明:**

TradingVolume 为输入的成交量序列，为 M*N 列数值矩阵。

period 为周期参数， $1 \leq \text{period} \leq N$ ，整数。

4. 公式:

$JDVOL = \text{SUM}(\text{tradingVolume}, \text{period});$

5. 中文含义:

JDVOL=成交量的 period 周期累和

6. 输出参数说明:

jdTradingVolume-是计算结果矩阵，与输入的价格序列为同型矩阵，表示阶段内总成交

量值矩阵。

详细例子：

```
>> TradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70  4314.14  
1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10  8686.95  
3997.83  8000.68  9106.48  2638.03  1360.69  5797.05  1449.55];
```

```
>> [jdTradingVolume]=JDVOL(TradingVolume,3)
```

jdTradingVolume =

NaN	NaN	9285.90	9063.73	11616.09	7757.20
8731.31	7588.00	11966.78	15646.91		
NaN	NaN	26055.03	20433.88	20685.46	21104.99
19745.19	13105.20	9795.77	8607.29		

2.8.8. JDZF-阶段内涨幅

函数解释：

1. **用途:** 计算阶段内涨幅 JDZF 指标值。统计一段时间内的涨幅。

2. **函数详细结构:**

`[jdzf]= JDZF(closePrice, period)`

3. **输入参数说明:**

closePrice 为输入的成交额序列，为 M*N 列数值矩阵。

period 为周期参数， $1 \leq \text{period} < N$ ，整数。

4. **公式:**

$$JDZF = (\text{closePrice} - \text{REF}(\text{closePrice}, \text{period})) / \text{REF}(\text{closePrice}, \text{period});$$

5. **中文含义:**

$$JDZF = (\text{收盘价} - \text{period 日前的收盘价}) / \text{period 日前的收盘价}$$

6. **输出参数说明:**

jdzf-是计算结果矩阵，与输入的成交额序列为同型矩阵，表示阶段内涨幅值矩阵。

7. **示例:**

```
>> closePrice = [ 20.40  19.50  19.48  19.15  19.20  18.86  18.70  19.10
19.71  19.25;  5.42  5.39  5.40  5.29  5.26  5.25  5.17  5.16
5.29  5.27  ];

>> [jdzf]=JDZF(highPrice, lowPrice, closePrice, 5)

jdzf =

NaN      NaN      NaN   -0.0613  -0.0154  -0.0318  -0.0235  -0.0052
```

0.0451	0.0294						
NaN	NaN	NaN	-0.0240	-0.0241	-0.0278	-0.0227	-0.0190
0.0076	0.0193						

2.8.9. NWPB-内外盘比

函数解释：

1. **用途:** 计算内外盘比 NWPB 指标值。该比值大于 1 则说明多方占据一定优势，比值越大多头优势越明显，可谨慎看涨。

2. **函数详细结构:**

[nwpb]= NWPB(BuyTradingVolume, SellTradingVolume)

3. **输入参数说明:**

BuyTradingVolume 为输入的主动性买单成交量序列，为 M*N 列数值矩阵，

SellTradingVolume 为输入的主动性卖单成交量序列，为 M*N 列数值矩阵。

4. **公式:**

$$NWPB = \text{SUM}(\text{BUYVOL}, 0) / \text{SUM}(\text{SELLVOL}, 0);$$

5. **中文含义:**

NWPB=主动性买单成交量的历史累和/主动性卖单成交量的历史累和

6. 输出参数说明:

nwpb-是计算结果矩阵，与输入的主动性买单成交量序列为同型矩阵，表示

内外盘比率值矩阵。

7. 示例:

```
>> buyTradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70
4314.14  1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10
8686.95  3997.83  8000.68  9106.48  2638.03  1360.69  5797.05
1449.55];

>> sellTradingVolume = [ 4172.67  9027.16  4908.64  3377.19  692.47
7802.52  2416.91  964.55  9420.51  5752.09; 496.54  9447.87  4892.53
9000.54  1112.03  3897.39  4039.12  1319.73  9561.35  597.80];

>> [nwpb]=NWPB(buyTradingVolume,sellTradingVolume)

nwpb =

0.26      0.08      0.51      0.47      0.51      0.52
0.53      0.56      0.63      0.67
19.37      1.75      1.76      1.26      1.53      1.63
1.51      1.50      1.30      1.32
```


2.8.10.NWPC-内外盘差

函数解释：

1. **用途:** 计算内外盘差 NWPC 指标值。该差值大于 0，则多方占据一定优势，差值越大优势越明显，可谨慎看涨。

2. **函数详细结构:**

[nwpc]= NWPC(BuyTradingVolume, SellTradingVolume)

3. **输入参数说明:**

BuyTradingVolume 为输入的主动性买单成交量序列，为 M*N 列数值矩阵，

SellTradingVolume 为输入的主动性卖单成交量序列，为 M*N 列数值矩阵。

4. **公式:**

$$NWPC = \text{SUM}(\text{BUYVOL}, 0) - \text{SUM}(\text{SELLVOL}, 0);$$

5. **中文含义:**

NWPC=主动性买单成交量的历史累和-主动性卖单成交量的历史累和

6. **输出参数说明:**

nwpc-是计算结果矩阵，与输入的主动性买单成交量序列为同型矩阵，表示

内外盘差值矩阵。

7. **示例:**

```
>> buyTradingVolume = [ 1066.53  46.34  8173.03  844.36  2598.70
4314.14  1818.47  1455.39  8692.92  5498.60; 9618.98  7749.10
8686.95  3997.83  8000.68  9106.48  2638.03  1360.69  5797.05
```

```

1449.55];

>> sellTradingVolume = [ 4172.67  9027.16  4908.64  3377.19  692.47
7802.52  2416.91  964.55  9420.51  5752.09; 496.54  9447.87  4892.53
9000.54  1112.03  3897.39  4039.12  1319.73  9561.35  597.80];

>> [nwpc]=NWPC(buyTradingVolume,sellTradingVolume)

nwpc =

-3106.14      -12086.96      -8822.57      -11355.41      -12449.17
-15937.56     -16536.00     -16045.15     -16772.74     -17026.23
 9122.44       7423.66      11218.09       6215.38      13104.03
18313.12      16912.03      16952.99      13188.69      14040.45

```

3. 选股模型 API

3.1. 指标条件选股模型

3.1.1. BIASBUY-乖离率买入

函数解释：

1. **用途：** 计算乖离率买入 BIASBUY 选股模型。负的乖离率越小，空头回补的可能性越大，因此，负的乖离率向下穿买入线，为买入时机。
2. **函数详细结构：**

```
[ biasbuy, diff ]= BIASBUY(closePrice, period, threshold )
```

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period 为周期参数， $1 \leq \text{period} \leq N$ ，整数。

threshold 为阈值参数，为 BIAS 买入需小于的临界值。

4. 公式:

$$\text{BIAS} = (\text{closePrice} - \text{MA}(\text{closePrice}, \text{period})) / \text{MA}(\text{closePrice}, \text{period}) * 100;$$
$$\text{BIASBUY} = \text{BIAS} < \text{threshold};$$
$$\text{DIFF} = \text{BIAS} - \text{threshold};$$

5. 中文含义:

乖离率 = (收盘价 - 收盘价的 period 日简单移动平均) / 收盘价的 period 日简单移动平均 * 100

$$\text{BIASBUY} = \text{乖离率} < \text{threshold}$$
$$\text{DIFF} = \text{乖离率} - \text{threshold}$$

6. 输出参数说明:

biasbuy 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的买入信号值矩阵。值=1 表示满足买入条件，值=0 表示不满足买入条件。

diff 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 BIAS 与临界值 threshold 的差值矩阵。

详细例子：

```
>> closePrice = [ 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71 19.25 ;  
5.42 5.39 5.40 5.29 5.26 5.25 5.17 5.16 5.29 5.27]  
  
>> [biasbuy, diff] = BIASBUY(closePrice, 3, -1)  
  
biasbuy =
```

0	0	1	1	0	1	1	0	0	0
0	0	0	1	1	0	1	0	0	0

diff =

NaN	NaN	-0.58	-0.17	0.60	-0.10
-0.16	2.13	3.82	0.47		
NaN	NaN	0.94	-0.31	-0.07	0.68
-0.08	0.36	2.60	1.57		

3.1.2. BIASSELL-乖离率卖出

函数解释：

1. **用途:** 计算乖离率卖出 BIASSELL 选股模型。乖离率越大，多头清仓的可能性越大，因此，正的乖离率向上穿卖出线，为卖出时机。

2. **函数详细结构:**

[biassell, diff]= BIASSELL(closePrice, period, threshold)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵。

period 为周期参数，1 = < period = < N，整数。

threshold 为阈值参数，为 BIAS 卖出需大于的临界值。

4. **公式:**

```
BIAS = (closePrice-MA(closePrice,period))/MA(closePrice,period)*100;
BIASSELL = BIAS > threshold;
DIFF= BIAS - threshold;
```

5. 中文含义:

乖离率=(收盘价-收盘价的 period 日简单移动平均)/收盘价的 period 日简单移动平均*100

BIASSELL=乖离率>threshold

DIFF=乖离率-threshold

6. 输出参数说明:

biassell-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的卖出信号值矩阵。值=1 表示满足卖出条件，值=0 表示不满足卖出条件。

diff-是计算结果矩阵，与输入的价格序列为同型矩阵，表示 BIAS 与临界值 threshold 的差值矩阵。

详细例子：

```
>> closePrice = [ 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71 19.25 ;
5.42 5.39 5.40 5.29 5.26 5.25 5.17 5.16 5.29 5.27]

>> [biassell, diff] = BIASSELL(closePrice, 3, -3)
```

biassell =

0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	1	0

diff =

NaN	NaN	-2.58	-2.17	-1.40	-2.10
-2.16	0.13	1.82	-1.53		

NaN	NaN	-1.06	-2.31	-2.07	-1.32
-2.08	-1.64	0.60	-0.43		

3.1.3. BOLLBUY-布林带买入

函数解释：

1. **用途：**计算布林带买入 BOLLBUY 选股模型。股票收盘价向上突破下限

LOWPRICEER，为买入时机。

2. **函数详细结构：**

[bollbuy]= BOLLBUY(closePrice, period, weight)

[bollbuy, diff1]= BOLLBUY(closePrice, period, weight)

[bollbuy, diff1, diff2]= BOLLBUY(closePrice, period, weight)

3. **输入参数说明：**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=2。

period 为周期参数，1 =< period <N，整数。

weight 为阈值参数，weight >=1，整数。

4. **公式：**

MID = MA(closePrice,period);

UPPER= MID+weight*STD(closePrice,period);

LOWER= MID-weight*STD(closePrice,period);

BOLLBUY = CROSS(closePrice,LOWER);

diff1 = REF(closePrice,1) – REF(LOWER,1)

diff2 = closePrice - LOWER;

5. **中文含义：**

MID=收盘价的 period 日简单移动平均

UPPER=MID+weight*收盘价的 period 日估算标准差

$LOWPRICEER = MID - weight * \text{收盘价的 period 日估算标准差}$

$BOLLBUY = \text{收盘价上穿 } LOWPRICEER$

$diff1 = \text{昨收} - \text{昨日 } LOWPRICEER$

$diff2 = \text{收盘价} - LOWPRICEER$

6. 输出参数说明:

BOLLBUY-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的
买入信号值矩阵。值=1 表示满足买入条件，值=0 表示不满足买入条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期 closePrice
与布林带下界的差值矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期 closePrice
与布林带下界的差值矩阵。

7. 示例：

>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]

>> [bollbuy, diff1, diff2] = BOLLBUY(a, 2, 2)

bollbuy =

NaN	0	0	0	0
NaN	0	0	0	0

diff1 =

NaN	NaN	0.1531	0.1723	0.0640
NaN	NaN	0.0383	0	0.2680

diff2 =

NaN	NaN	0.1723	0.0640	0.2194
NaN	NaN	0	0.2680	0.0640

3.1.4. BOLLSELL-布林带卖出

函数解释：

1. **用途:** 计算布林带卖出 BOLLSELL 选股模型。股价收盘价向下突破上限，为卖出时机。

2. **函数详细结构:**

[bollsell]= BOLLSELL(closePrice, period, weight)

[bollsell, diff1]= BOLLSELL(closePrice, period, weight)

[bollsell, diff1, diff2]= BOLLSELL(closePrice, period, weight)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=2。

period 为周期参数，1 =< period <N，整数。

weight 为阈值参数，weight >=1，整数。

4. **公式:**

MID = MA(closePrice,period);

UPPER= MID+weight*STD(closePrice,period);

LOWER= MID-weight*STD(closePrice,period);

BOLLSELL = CROSS(UPPER,closePrice);

diff1 = REF(closePrice,1) - REF(UPPER,1)

diff2 = CLOSE - UPPER;

5. **中文含义:**

MID=收盘价的 period 日简单移动平均

$UPPER = MID + weight * \text{收盘价的 period 日估算标准差}$

$LOWPRICEER = MID - weight * \text{收盘价的 period 日估算标准差}$

$BOLLSSELL = UPPER$ 上穿收盘价

$diff1 = \text{昨收} - \text{昨日 UPPER}$

$diff2 = \text{收盘价} - UPPER$

6. 输出参数说明:

BOLLSSELL 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的卖出信号值矩阵。值=1 表示满足卖出条件，值=0 表示不满足卖出条件。

diff1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期 closePrice 与

布林带上界的差值矩阵。

diff2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期 closePrice

与布林带上界的差值矩阵。

详细例子：

```
>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [bollsell, diff1, diff2] = BOLLBUY(a, 2, 2)
```

例子运行结果展示：

bollsell =

NaN	0	0	0	0
-----	---	---	---	---

NaN	0	0	0	0
-----	---	---	---	---

diff1 =

NaN	NaN	-0.0731	-0.0823	-0.1340
NaN	NaN	-0.0183	0	-0.1280

diff2 =

NaN	NaN	-0.0823	-0.1340	-0.4594
NaN	NaN	0	-0.1280	-0.1340

3.1.5. KDBUY –KD 随机指标买入

函数解释：

1. **用途:** 计算 KD 买入 KDBUY 选股模型。KD 指标线 K 向上冲破线 D，且 K 值在 20 以下，是买进信号。

2. **函数详细结构:**

```
[ kdbuy ]= KDBUY(highPrice, lowPrice, closePrice, period1, period2,
period3)
```

```
[ kdbuy, diff1 ]= KDBUY(highPrice, lowPrice, closePrice, period1, period2,
period3)
```

```
[ kdbuy, diff1, diff2 ]= KDBUY(highPrice, lowPrice, closePrice, period1,
period2, period3)
```

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 列数值矩阵，N>=2。

lowPrice 为输入的最低价序列，与 highPrice 同维度。

closePrice 为输入的收盘价序列，与 highPrice 同维度。

period1 为周期参数 1 , $1 \leq \text{period1} < N$, 整数。

period2 为周期参数 2 , $\text{period2} \geq 1$, 整数。

period3 为阈值参数 3 , $\text{period3} \geq 1$, 整数。

4. 公式:

$$\text{RSV} = (\text{closePrice} - \text{LLV}(\text{lowPrice}, \text{period1})) /$$
$$(\text{HHV}(\text{HIGH}, \text{M1}) - \text{LLV}(\text{lowPrice}, \text{period1})) * 100;$$
$$\text{K} = \text{SMA}(\text{RSV}, \text{period2}, 1);$$
$$\text{D} = \text{SMA}(\text{K}, \text{period3}, 1);$$
$$\text{KDBUY} = \text{CROSS}(\text{K}, \text{D}) \text{ AND } \text{K} < 20;$$
$$\text{diff1} = \text{REF}(\text{K}, 1) - \text{REF}(\text{D}, 1);$$
$$\text{diff2} = \text{K} - \text{D}$$

5. 中文含义:

$$\text{RSV} = (\text{收盘价} - \text{period1 日内最低价的最小值}) / (\text{period1 日内最高价的最大值}$$
$$- \text{period1 日内最低价的最小值}) * 100$$

$$\text{K} = \text{RSV 的 period2 日}[1 \text{ 日权重}] \text{移动平均}$$

$$\text{D} = \text{K 的 period3 日}[1 \text{ 日权重}] \text{移动平均}$$

$$\text{KDBUY} = \text{K 上穿 D 并且 } \text{K} < 20$$

$$\text{diff1} = \text{昨日 K} - \text{昨日 D}$$

$$\text{diff2} = \text{K} - \text{D}$$

6. 输出参数说明:

kdbuy-是计算结果矩阵 , 与输入的价格序列为同型矩阵 , 表示计算得到的

买入信号值矩阵。值=1 表示满足买入条件 , 值=0 表示不满足买入条件。

diff1-是计算结果矩阵 , 与输入的价格序列为同型矩阵 , 表示上周期的 K 值
与 D 值的差值矩阵。

diff2-是计算结果矩阵 , 与输入的价格序列为同型矩阵 , 表示最新周期的 K

值与 D 值的差值矩阵。

7. 示例:

```
>> a=[10.78,11,10.98,11.08,10.88;3.23,3.29,3.25,3.41,3.44;]  
>> b=[10.78,10.63,10.67,10.7,10.55;3.23,3.18,3.25,3.25,3.28;]  
>> c=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]  
>> [kdbuy, diff1, diff2] = KDBUY(a, b, c, 2, 2, 3)
```

kdbuy =

NaN	0	0	0	0
-----	---	---	---	---

NaN	0	0	0	0
-----	---	---	---	---

diff1 =

NaN	NaN	0	8.1081	-2.2962
-----	-----	---	--------	---------

NaN	NaN	0	0	7.9545
-----	-----	---	---	--------

diff2 =

NaN	NaN	0	-11.7557	-25.1277
-----	-----	---	----------	----------

NaN	NaN	0	7.9545	-7.6057
-----	-----	---	--------	---------

3.1.6. KDJBUY –KDJ 随机指标买入

函数解释：

1. **用途:** 计算 KDJ 买入 KDJBUY 选股模型。KDJ 指标中,J 线向上突破零线，买入信号。

2. **函数详细结构:**

[kdjbuy]= KDJBUY(highPrice, lowPrice, closePrice, period1, period2,

period3)

[kdjbuy, diff1]= KDJBUI(highPrice, lowPrice, closePrice, period1, period2,
period3)

[kdjbuy, diff1, diff2]= KDJBUI(highPrice, lowPrice, closePrice, period1,
period2, period3)

3. 输入参数说明:

highPrice 为输入的最高价序列，为 M*N 列数值矩阵，N>=2。

lowPrice 为输入的最低价序列，与 highPrice 同维度。

closePrice 为输入的收盘价序列，与 highPrice 同维度。

period1 为周期参数 1，1 = < period1 < N，整数。

period2 为周期参数 2，period2 >=1，整数。

period3 为阈值参数 3，period3 >=1，整数。

4. 公式:

$$RSV = (\text{closePrice} - \text{LLV}(\text{lowPrice}, \text{period1})) /$$
$$(\text{HHV}(\text{highPrice}, \text{M1}) - \text{LLV}(\text{lowPrice}, \text{period1})) * 100;$$
$$K = \text{SMA}(RSV, \text{period2}, 1);$$
$$D = \text{SMA}(K, \text{period3}, 1);$$
$$J = 3 * K - 2 * D;$$
$$\text{KDJBUI} = \text{CROSS}(J, 0);$$
$$\text{diff1} = \text{REF}(J, 1);$$
$$\text{diff2} = J;$$

5. 中文含义:

$$RSV = (\text{收盘价} - \text{period1 日内最低价的最低值}) / (\text{period1 日内最高价的最高值}$$
$$- \text{period1 日内最低价的最低值}) * 100$$

K = RSV 的 period2 日[1 日权重]移动平均

D = K 的 period3 日[1 日权重]移动平均

$J=3*K-2*D$

KDJBUIY=J 上穿 0

diff1=昨日 J

diff2=J

6. 输出参数说明:

kdjbuy-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的
买入信号值矩阵。值=1 表示满足买入条件，值=0 表示不满足买入条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 J 值
矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的 J
值矩阵。

7. 示例:

```
>> a=[10.78,11,10.98,11.08,10.88;3.23,3.29,3.25,3.41,3.44;]
```

```
>> b=[10.78,10.63,10.67,10.7,10.55;3.23,3.18,3.25,3.25,3.28;]
```

```
>> c=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [kdjbuy, diff1, diff2] = KDJBUIY(a, b, c, 2, 2, 3)
```

kdjbuy =

NaN	0	0	0	0
-----	---	---	---	---

NaN	0	0	0	0
-----	---	---	---	---

diff1 =

NaN	NaN	62.1622	90.5405	58.1795
-----	-----	---------	---------	---------

NaN	NaN	63.6364	63.6364	91.4773
-----	-----	---------	---------	---------

diff2 =

NaN NaN 86.4865 45.3417 -7.3384

NaN NaN 63.6364 91.4773 40.9938

3.1.7. KDJSELL-KDJ 随机指标卖出

函数解释：

1. **用途：**计算 KDJ 卖出 KDJSELL 选股模型。KDJ 指标中,J 线向下突破一百线 , 卖出信号。

2. **函数详细结构:**

[kdjsell]= KDJSELL(highPrice, lowPrice, closePrice, period1, period2, period3)

[kdjsell, diff1]= KDJSELL(highPrice, lowPrice, closePrice, period1, period2, period3)

[kdjsell, diff1,diff2]= KDJSELL(highPrice, lowPrice, closePrice, period1, period2, period3)

3. **输入参数说明:**

highPrice 为输入的 M*N 最高价矩阵 , N>=2。

lowPrice 为输入的最低价矩阵 , 与 highPrice 同维度。

closePrice 为输入的收盘价矩阵 , 与 highPrice 同维度。

period1 为周期参数 1 , 1 =< period1 <N , 整数。

period2 为周期参数 2 , period2 >=1 , 整数。

period3 为阈值参数 3 , period3 >=1 , 整数。

4. 公式:

```
RSV=(closePrice-LLV(lowPrice,period1))/  
(HHV(highPrice,M1)-LLV(lowPrice,period1))*100;  
K=SMA(RSV,period2,1);  
D=SMA(K,period3,1);  
J=3*K-2*D;  
KDJSSELL = CROSS(100,J);  
diff1 = REF(J, 1) -100;  
diff2 = J - 100;
```

5. 中文含义:

$RSV = (\text{收盘价} - \text{period1 日内最低价的最小值}) / (\text{period1 日内最高价的最大值} - \text{period1 日内最低价的最小值}) * 100$

$K = \text{RSV 的 period2 日[1 日权重]移动平均}$

$D = K \text{ 的 period3 日[1 日权重]移动平均}$

$J = 3 * K - 2 * D$

$KDJSSELL = 100 \text{ 上穿 } J$

$diff1 = \text{昨日 } J - 100$

$diff2 = J - 100$

6. 输出参数说明:

$KDJSSELL$ -是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的卖出信号值矩阵。值=1 表示满足卖出条件，值=0 表示不满足卖出条件。

$diff1$ -是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 J 值与 100 的差值矩阵。

$diff2$ -是计算结果，与输入的价格序列为同型矩阵，表示最新周期的 J 值与 100 的差值矩阵。

7. 示例:


```
>> a=[10.78,11,10.98,11.08,10.88;3.23,3.29,3.25,3.41,3.44;]

>> b=[10.78,10.63,10.67,10.7,10.55;3.23,3.18,3.25,3.25,3.28;]

>> c=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]

>> [kdjsell, diff1, diff2] = KDJSELL(a, b, c, 2, 2, 3)

kdjsell =

    NaN     0     0     0     0

    NaN     0     0     0     0

diff1 =

    NaN     NaN  -37.8378   -9.4595  -41.8205

    NaN     NaN  -36.3636  -36.3636   -8.5227

diff2 =

    NaN     NaN  -13.5135  -54.6583 -107.3384

    NaN     NaN  -36.3636   -8.5227  -59.0062
```

3.1.8. KDSELL-KD 随机指标卖出

函数解释：

1. **用途:** 计算 KD 卖出 KDSELL 选股模型。KD 指标线 K 向下穿破线 D , 且 K 值在 80 以上 , 卖出信号。

2. **函数详细结构:**

```
[ kdsell ]= KDSELL(highPrice, lowPrice, closePrice, period1, period2,
period3)
```

```
[ kdsell, diff1 ]= KDSELL(highPrice, lowPrice, closePrice, period1, period2,
```

period3)

[kdsell, diff1, diff2]= KDSELL(highPrice, lowPrice, closePrice, period1,
period2, period3)

3. 输入参数说明:

highPrice 为输入的最高价矩阵，为 M*N 列数值矩阵，N >= 2。

lowPrice 为输入的最低价矩阵，与 highPrice 同维度。

closePrice 为输入的收盘价矩阵，与 highPrice 同维度。

period1 为周期参数 1，1 <= period1 < N，整数。

period2 为周期参数 2，period2 >= 1，整数。

period3 为阈值参数 3，period3 >= 1，整数。

4. 公式:

$$RSV = (closePrice - LLV(lowPrice, period1)) / (HHV(highPrice, M1) - LLV(lowPrice, period1)) * 100;$$
$$K = SMA(RSV, period2, 1);$$
$$D = SMA(K, period3, 1);$$
$$J = 3 * K - 2 * D;$$
$$KDSELL = CROSS(D, K) \text{ AND } D > 80;$$
$$diff1 = REF(K, 1) - REF(D, 1);$$
$$diff2 = K - D$$

5. 中文含义:

$$RSV = (收盘价 - period1 \text{ 日内最低价的最小值}) / (period1 \text{ 日内最高价的最大值} - period1 \text{ 日内最低价的最小值}) * 100$$

K = RSV 的 period2 日[1 日权重]移动平均

D = K 的 period3 日[1 日权重]移动平均

KDSELL = D 上穿 K 并且 D > 80

diff1 = 昨日 K - 昨日 D

diff2=K-D

6. 输出参数说明:

kdsell-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的
卖出信号值矩阵。值=1 表示满足卖出条件，值=0 表示不满足卖出条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 K 值与
D 值的差值矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的 K
值与 D 值的差值矩阵

7. 示例：

```
>> a=[10.78,11,10.98,11.08,10.88;3.23,3.29,3.25,3.41,3.44;]
```

```
>> b=[10.78,10.63,10.67,10.7,10.55;3.23,3.18,3.25,3.25,3.28;]
```

```
>> c=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [kdsell, diff1, diff2] = KDSELL(a, b, c, 2, 2, 3)
```

kdsell =

NaN	0	0	0	0
NaN	0	0	0	0

diff1 =

NaN	NaN	0	8.1081	-2.2962
NaN	NaN	0	0	7.9545

diff2 =

NaN	NaN	0	-11.7557	-25.1277
NaN	NaN	0	7.9545	-7.6057

3.1.9. MABUY-均线指标买入

函数解释：

1. **用途:** 计算简单移动平均线买入 MABUY 选股模型。短期均线从下向上与长期均线交叉，为买入信号。

2. **函数详细结构:**

[mabuy]= MABUY(closePrice, period1, period2)

[mabuy,diff1]= MABUY(closePrice, period1, period2)

[mabuy,diff1, diff2]= MABUY(closePrice, period1, period2)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=2。

period1 为短周期参数 1，1 = < period1 < N，整数。

period2 为长周期参数 1，1 = < period2 < N，整数。

4. **公式:**

$MABUY = CROSS(MA(closePrice, period1), MA(closePrice, period2));$

$diff1 = REF(MA(closePrice, period1), 1) - REF(MA(closePrice, period2), 1);$

$diff2 = MA(closePrice, period1) - MA(closePrice, period2);$

5. **中文含义:**

MABUY=收盘价的 period1 日简单移动平均上穿收盘价的 period2 日简单移动平均

diff1=昨日收盘价的 period1 日简单移动平均-昨日收盘价的 period2 日简单移动平均

diff2=收盘价的 period1 日简单移动平均-收盘价的 period2 日简单移动平均

6. 输出参数说明:

mabuy-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的买入信号值矩阵。值=1 表示满足买入条件，值=0 表示不满足买入条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 period1 周
期平均值与 period2 周期平均值的差值矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的 period1
周期平均值与 period2 周期平均值的差值矩阵。

7. 示例:

```
>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [mabuy, diff1, diff2] = MABUY(a, 2, 3)
```

mabuy =

NaN	0	0	0	1
NaN	0	0	0	0

diff1 =

NaN	NaN	NaN	0.0100	-0.0000
NaN	NaN	NaN	0.0133	0.0583

diff2 =

NaN	NaN	NaN	0.0700	0.0100
NaN	NaN	NaN	0.0133	0.0133

3.1.10.MACDBUY-MACD 指标买入

函数解释：

1. **用途:** 计算 MACD 买入 MACDBUY 选股模型。diff 与 DEA 形成金叉时为多头买入信号。

2. **函数详细结构:**

[macdbuy]= MACDBUY(closePrice, period1, period2, period3)

[macdbuy,diff1]= MACDBUY(closePrice, period1, period2, period3)

[macdbuy,diff1, diff2]= MACDBUY(closePrice, period1, period2, period3)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=2。

period1 为周期参数 1，period1 >=1，整数。

period2 为周期参数 1，period2 >=1，整数。

period3 为周期参数 1，period3 >=1，整数。

4. **公式:**

DIF = EMA(closePrice,period1) - EMA(closePrice,period2);

DEA= EMA(DIF,period3);

MACDBUY = CROSS(DIF,DEA);

diff1 = REF(DIF,1) - REF(DEA, 1);

diff2 = DIF - DEA

5. **中文含义:**

DIF=收盘价的 period1 日指数移动平均-收盘价的 period2 日指数移动平均

DEA=DIF 的 period3 日指数移动平均

MACDBUY=DIF 上穿 DEA

diff1=昨日 DIF-昨日 DEA

diff2=DIF-DEA

6. 输出参数说明:

macdbuy-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的买入信号值矩阵。值=1 表示满足买入条件，值=0 表示不满足买入条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 DIF 值与 DEA 值的差值矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的 DIF 值与 DEA 值的差值矩阵。

7. 示例:

```
>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [macdbuy, diff1, diff2] = MACDBUY(a, 3, 1, 2)
```

macdbuy =

NaN	0	0	0	0
NaN	0	0	1	0

diff1 =

NaN	0	0.0133	0.0128	-0.0182
NaN	0	0.0033	-0.0006	0.0223

diff2 =

NaN	0	0.0150	-0.0142	-0.0426
NaN	0	0	0.0233	-0.0156

3.1.11.MACDSELL-MACD 指标卖出

函数解释：

1. **用途：**计算 MACD 卖出 MACDSELL 选股模型。DEA 与 diff 形成死叉时为卖出信号。

2. **函数详细结构：**

[macdsell]= MACDSELL(closePrice, period1, period2, period3)

[macdsell, diff1]= MACDSELL(closePrice, period1, period2, period3)

[macdsell, diff1, diff2]= MACDSELL(closePrice, period1, period2, period3)

3. **输入参数说明：**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=2。

period1 为周期参数 1，period1 >=1，整数。

period2 为周期参数 1，period2 >=1，整数。

period3 为周期参数 1，period3 >=1，整数。

4. **公式：**

$DIFF = EMA(closePrice, period1) - EMA(closePrice, period2);$

$DEA = EMA(DIFF, period3);$

$MACDSELL = CROSS(DEA, DIFF);$

$diff1 = REF(DEA, 1) - REF(DIF, 1);$

$diff2 = DEA - DIF$

5. **中文含义：**

diff=收盘价的 period1 日指数移动平均-收盘价的 period2 日指数移动平均

DEA=diff 的 period3 日指数移动平均

MACDSELL=DEA 上穿 DIF

diff1=昨日 DEA-昨日 DIF

diff2=DEA-DIF

6. 输出参数说明:

macdsell-是计算结果矩阵，与输入的价格序列为同型矩阵，表示卖出信号值矩阵。值=1 表示满足卖出条件，值=0 表示不满足卖出条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 DEA 值与 DIF 值的差值矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的 DEA 值与 DIF 值的差值矩阵。

7. 示例:

```
>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [macdsell, diff1, diff2] = MACDSELL(a, 3, 1, 2)
```

macdsell =

NaN	0	0	1	0
NaN	0	0	0	1

diff1 =

NaN	0	0.0133	0.0128	-0.0182
NaN	0	0.0033	-0.0006	0.0223

diff2 =

NaN	0	0.0150	-0.0142	-0.0426
NaN	0	0	0.0233	-0.0156

3.1.12.MASELL-均线指标卖出

函数解释：

1. **用途:** 计算简单移动平均线卖出 MASELL 选股模型。短期均线从上往下穿越长期均线，为卖出信号。

2. **函数详细结构:**

[masell]= MASELL(closePrice, period1, period2)

[masell, diff1]= MASELL(closePrice, period1, period2)

[masell, diff1, diff2]= MASELL(closePrice, period1, period2)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=2。

period1 为短周期参数 1，1 =< period1 <N，整数。

period2 为长周期参数 1，1 =< period2 <N，整数。

4. **公式:**

$MASELL = CROSS(MA(closePrice, period1), MA(closePrice, period2));$

$diff1 = REF(MA(closePrice, period1), 1) - REF(MA(closePrice, period2), 1);$

$diff2 = MA(closePrice, period1) - MA(closePrice, period2);$

5. **中文含义:**

masell=收盘价的 period2 日简单移动平均上穿收盘价的 period1 日简单移动平均

diff1=昨日收盘价的 period1 日简单移动平均-昨日收盘价的 period2 日简单移动平均

diff2=收盘价的 period1 日简单移动平均-收盘价的 period2 日简单移动平均

6. 输出参数说明:

masell-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的
卖出信号值矩阵。值=1 表示满足卖出条件，值=0 表示不满足卖出条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周的 period1
周
期平均值与 period2 周期平均值的差值矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的
period1
周期平均值与 period2 周期平均值的差值矩阵。

7. 示例:

```
>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [masell, diff1, diff2] = MASELL(a, 2, 3)
```

masell =

NaN	0	0	0	0
NaN	0	0	0	0

diff1 =

NaN	NaN	NaN	0.0183	-0.0633
NaN	NaN	NaN	0.0233	0.0350

diff2 =

NaN	NaN	NaN	0.0417	0.0183
NaN	NaN	NaN	0.0067	0.0233

3.1.13.MTMBUY-动力指标买入

函数解释：

1. **用途:** 计算动力指标买入 MTMBUY 选股模型。MTM 向上突破零轴，为买入信号。

2. **函数详细结构:**

[mtmbuy]= MTMBUY(closePrice, period)

[mtmbuy, diff1]= MTMBUY(closePrice, period)

[mtmbuy, diff1, diff2]= MTMBUY(closePrice, period)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵， $N \geq 3$ 。

period 为周期参数 1， $1 \leq \text{period} \leq N-2$ ，整数。

4. **公式:**

$MTM = \text{closePrice} - \text{REF}(\text{closePrice}, \text{period})$;

$MTMBUY = \text{CROSS}(MTM, 0)$;

$\text{diff1} = \text{REF}(MTM, 1)$;

$\text{diff2} = MTM$;

5. **中文含义:**

动量线=收盘价- period 日前的(常量和变源张碳?

MTMBUY=动量线上穿 0

diff1=昨日动量线

diff2=动量线

6. **输出参数说明:**

mtmbuy-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的
买入信号值矩阵。值=1 表示满足买入条件，值=0 表示不满足买入条件。

diff1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 MTM
值矩阵。

diff2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的 MTM
值矩阵。

7. 示例:

```
>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [mtmbuy, diff1, diff2] = MTMBUY(a, 2)
```

mtmbuy =

NaN	0	0	0	0
-----	---	---	---	---

NaN	0	0	0	0
-----	---	---	---	---

diff1 =

NaN	NaN	NaN	0.1700	0.0200
-----	-----	-----	--------	--------

NaN	NaN	NaN	0.0200	0.1400
-----	-----	-----	--------	--------

diff2 =

NaN	NaN	NaN	0.0200	-0.3100
-----	-----	-----	--------	---------

NaN	NaN	NaN	0.1400	0.0700
-----	-----	-----	--------	--------

3.1.14.MTMSELL-动力指标卖出

函数解释：

1. **用途:** 计算动力指标卖出 MTMSELL 选股模型。MTM 向下突破零轴，为卖出

信号。

2. 函数详细结构:

[mtmsell]= MTMSELL(closePrice, period)

[mtmsell, diff1]= MTMSELL(closePrice, period)

[mtmsell, diff1, diff2]= MTMSELL(closePrice, period)

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵， $N \geq 3$ 。

period 为周期参数 1， $1 \leq \text{period} \leq N-2$ ，整数。

4. 公式:

$\text{MTM} = \text{closePrice} - \text{REF}(\text{closePrice}, \text{period})$;

$\text{MTMSELL} = \text{CROSS}(0, \text{MTM})$;

$\text{diff1} = \text{REF}(\text{MTM}, 1)$;

$\text{diff2} = \text{MTM}$;

5. 中文含义:

动量线=收盘价-M 日前的(常量和变潦张碳?

MTMSELL=0 上穿动量线

diff1=昨日动量线

diff2=动量线

6. 输出参数说明:

mtmsell 是计算结果矩阵，与输入矩阵的价格序列为同型矩阵，表示计算得到的卖出信号值矩阵。值=1 表示满足卖出条件，值=0 表示不满足卖出条件。

diff1 是计算结果矩阵，与输入矩阵的价格序列为同型矩阵，表示上周期的 MTM 值矩阵。

diff2 是计算结果矩阵，与输入矩阵的价格序列为同型矩阵，表示最新周期的

MTM 值矩阵。

7. 示例:

```
>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [mtmsell, diff1, diff2] = MTMSELL(a, 3)
```

mtmsell =

NaN	0	0	0	1
-----	---	---	---	---

NaN	0	0	0	0
-----	---	---	---	---

diff1 =

NaN	NaN	NaN	NaN	0.1000
-----	-----	-----	-----	--------

NaN	NaN	NaN	NaN	0.1600
-----	-----	-----	-----	--------

diff2 =

NaN	NaN	NaN	NaN	-0.2200
-----	-----	-----	-----	---------

NaN	NaN	NaN	NaN	0.0700
-----	-----	-----	-----	--------

3.1.15.RSIBUY-相对强弱指标买入

函数解释：

1. **用途:** 计算相对强弱指标买入 RSIBUY 选股模型。RSI 向上突破下限，为多头买入信号。

2. 函数详细结构:

```
[ rsibuy ]= RSIBUY(closePrice, period, threshold)
```

```
[ rsibuy, diff1 ]= RSIBUY(closePrice, period, threshold)
```

```
[ rsibuy, diff1, diff2 ]= RSIBUY(closePrice, period, threshold)
```

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=3。

period 为周期参数，period >=1，整数。

threshold 为阈值参数，threshold >=1，整数。

4. 公式:

```
LC= REF(closePrice,1);  
RSI_part1 = SMA(MAX(closePrice-LC,0),period,1);  
RSI_part2 = SMA(ABS(closePrice-LC),period,1);  
RSI=RSI_part1/RSI_part2*100  
RSIBUY = CROSS(RSI, threshold);  
diff1 = REF(RSI, 1) - threshold;  
diff2 = RSI - threshold
```

5. 中文含义:

LC=昨收

RSI_part1=SMA(收盘价-昨收与 0 的较大值,period,1) ;

RSI_part2=SMA(收盘价-昨收的绝对值,period,1) ;

RSI=RSI_part1/RSI_part2*100 ;

RSIBUY=RSI 上穿 threshold ;

diff1=昨日 RSI-threshold ;

diff2=RSI-threshold ;

6. 输出参数说明:

rsibuy-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的买入信号值矩阵。值=1 表示满足买入条件，值=0 表示不满足买入条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 RSI 值与阈值 threshold 的差值矩阵。

diff2-是计算结果，与输入的价格序列为同型矩阵，表示最新周期的 RSI 值与
阈值 threshold 的差值矩阵。

7. 输出参数说明:

```
>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [rsibuy, diff1, diff2] = RSIBUY(a, 6, 4)
```

rsibuy =

NaN	0	0	0	0
-----	---	---	---	---

NaN	0	0	0	0
-----	---	---	---	---

diff1 =

NaN	NaN	96.0000	96.0000	81.3659
-----	-----	---------	---------	---------

NaN	NaN	96.0000	96.0000	96.0000
-----	-----	---------	---------	---------

diff2 =

NaN	NaN	96.0000	82.5385	51.6931
-----	-----	---------	---------	---------

NaN	NaN	NaN	96.0000	58.5000
-----	-----	-----	---------	---------

3.1.16. RSISELL-相对强弱指标卖出

函数解释：

1. **用途:** 计算相对强弱指标卖出 RSISELL 选股模型。RSI 向下突破上限，为卖出信号。

2. 函数详细结构:

```
[ rsisell ]= RSISELL(closePrice, period, threshold)
```

```
[ rsisell, diff1 ]= RSISELL(closePrice, period, threshold)
```

[rsisell, diff1, diff2]= RSISELL(closePrice, period, threshold)

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 M*N 列数值矩阵，N>=3。

period 为周期参数，period >=1，整数。

threshold 为阈值参数，threshold >=1，整数。

4. 公式:

LC= REF(closePrice,1);
RSI_part1 = SMA(MAX(closePrice-LC,0),period,1);
RSI_part2 = SMA(ABS(closePrice-LC),period,1);
RSI=RSI_part1/RSI_part2*100
RSIBUY = CROSS(RSI, threshold);
diff1 = REF(RSI, 1) - threshold;
diff2 = RSI - threshold

5. 中文含义:

LC=昨收

RSI_part1=SMA(收盘价-昨收与 0 的较大值,period,1) ;

RSI_part2=SMA(收盘价-昨收的绝对值,period,1) ;

RSI=RSI_part1/RSI_part2*100 ;

RSIBUY=RSI 上穿 threshold ;

diff1=昨日 RSI-threshold ;

diff2=RSI-threshold ;

6. 输出参数说明:

rsisell-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的

卖出信号值矩阵。值=1 表示满足卖出条件，值=0 表示不满足卖出条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 RSI 值

与阈值 threshold 的差值矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的 RSI

值与阈值 threshold 的差值矩阵。

7. 示例:

```
>> a=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [rsisell, diff1, diff2] = RSISELL(a, 4, 6)
```

rsisell =

NaN	0	0	0	0
-----	---	---	---	---

NaN	0	0	0	0
-----	---	---	---	---

diff1 =

NaN	NaN	94.0000	94.0000	71.9528
-----	-----	---------	---------	---------

NaN	NaN	94.0000	94.0000	94.0000
-----	-----	---------	---------	---------

diff2 =

NaN	NaN	94.0000	73.4118	34.9091
-----	-----	---------	---------	---------

NaN	NaN	NaN	94.0000	54.0000
-----	-----	-----	---------	---------

3.1.17.WRBUY-威廉指标买入

函数解释：

1. **用途:** 计算威廉指标买入 WRBUY 选股模型。WR 上穿指标区间，出现超卖，即将见底，为买进信号。

2. **函数详细结构:**

[wrbuy]= WRBUY(highPrice, lowPrice, closePrice, period, threshold)

[wrbuy,diff1]= WRBUY(highPrice, lowPrice, closePrice, period, threshold)

[wrbuy,diff1, diff2]= WRBUY(highPrice, lowPrice, closePrice, period, threshold)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 列数值矩阵，N>=2。

lowPrice 为输入的最低价矩阵，与 highPrice 同维度。

closePrice 为输入的收盘价矩阵，与 highPrice 同维度。

period 为周期参数，1 = < period < N，整数。

threshold 为阈值参数，threshold >=1，整数。

4. **公式:**

$$\begin{aligned} WR &= (HHV(highPrice, M) - closePrice) / \\ & (HHV(highPrice, period) - LLV(lowPrice, period)) * 100; \\ WRBUY &= CROSS(threshold, WR); \\ VAR1 &= REF(WR, 1) - threshold; \\ VAR2 &= WR - threshold; \end{aligned}$$

5. **中文含义:**

WR=(period 日内最高价的最高值-收盘价)/(period 日内最高价的最高值
-period 日内最低价的最低值)*100

WRBUY=threshold 上穿 WR

diff1=昨日 WR-threshold

diff2=WR-threshold

6. 输出参数说明:

wrbuy-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的买入信号值矩阵。值=1 表示满足买入条件，值=0 表示不满足买入条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 WR 值与阈值 threshold 的差值矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的 WR 值与阈值 threshold 的差值矩阵。

详细例子：

```
>> a=[10.78,11,10.98,11.08,10.88;3.23,3.29,3.25,3.41,3.44;]  
>> b=[10.78,10.63,10.67,10.7,10.55;3.23,3.18,3.25,3.25,3.28;]  
>> c=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]  
>> [wrbuy, diff1, diff2] = WRBUY(a, b, c, 2, 2)
```

例子运行结果展示：

wrbuy =

NaN	0	0	0	0
-----	---	---	---	---

NaN	0	0	0	0
-----	---	---	---	---

diff1 =

NaN	NaN	35.8378	11.5135	46.7805
-----	-----	---------	---------	---------

NaN	NaN	34.3636	34.3636	10.5000
-----	-----	---------	---------	---------

diff2 =

NaN	NaN	11.5135	46.7805	81.0189
-----	-----	---------	---------	---------

NaN NaN 34.3636 10.5000 61.1579

3.1.18.WRSELL-威廉指标卖出

函数解释：

1. **用途:** 计算威廉指标卖出 WRSELL 选股模型。威廉指标向下突破某一区域为卖出信号。

2. **函数详细结构:**

[wrsell]= WRSELL(highPrice, lowPrice, closePrice, period, threshold)

[wrsell, diff1]= WRSELL(highPrice, lowPrice, closePrice, period, threshold)

[wrsell, diff1, diff2]= WRSELL(highPrice, lowPrice, closePrice, period, threshold)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 列数值矩阵，N>=2。

lowPrice 为输入的最低价矩阵，与 highPrice 同维度。

closePrice 为输入的收盘价矩阵，与 highPrice 同维度。

period 为周期参数，1 = < period < N，整数。

threshold 为阈值参数，threshold >=1，整数。

4. **公式:**

$$\begin{aligned} WR &= (HHV(highPrice, period) - closePrice) / \\ & (HHV(highPrice, period) - LLV(lowPrice, period)) * 100; \\ WRSELL &= CROSS(WR, threshold); \\ VAR1 &= REF(WR, 1) - threshold; \\ VAR2 &= WR - threshold; \end{aligned}$$

5. **中文含义:**

$$WR = (\text{period 日内最高价的最高值} - \text{收盘价}) / (\text{period 日内最高价的最高值} - \text{period 日内最低价的最低值}) * 100$$

$$WRSELL = WR \text{ 上穿 threshold}$$

$$\text{diff1} = \text{昨日 WR} - \text{threshold}$$

$$\text{diff2} = WR - \text{threshold}$$

6. 输出参数说明:

wrsell-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的卖出信号值矩阵。值=1 表示满足卖出条件，值=0 表示不满足卖出条件。

diff1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上周期的 WR 值与 threshold 的差值矩阵。

diff2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的 WR 值与 threshold 的差值矩阵。

7. 示例:

```
>> a=[10.78,11,10.98,11.08,10.88;3.23,3.29,3.25,3.41,3.44;]
>> b=[10.78,10.63,10.67,10.7,10.55;3.23,3.18,3.25,3.25,3.28;]
>> c=[10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
>> [wrsell, diff1, diff2] = WRSELL(a, b, c, 2, 2)
```

wrsell =

NaN	0	0	0	0
NaN	0	0	0	0

diff1 =

NaN	NaN	35.8378	11.5135	46.7805
-----	-----	---------	---------	---------

	NaN	NaN	34.3636	34.3636	10.5000
--	-----	-----	---------	---------	---------

diff2 =

	NaN	NaN	11.5135	46.7805	81.0189
	NaN	NaN	34.3636	10.5000	61.1579

3.2. 走势特征选股模型

3.2.1. BREAKPLAT-突破长期盘整

函数解释：

1. **用途:** 计算突破长期盘整 BREAKPLAT 选股模型。突破长期盘整，股价有望开始走强。

2. **函数详细结构:**

[breakplat]= BREAKPLAT(highPrice, lowPrice, closePrice, M1, M2)

[breakplat, var]= BREAKPLAT(highPrice, lowPrice, closePrice, M1, M2)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 M*N 列数值矩阵，N>=2。

lowPrice 为输入的最低价矩阵，与 highPrice 同维度。

closePrice 为输入的收盘价矩阵，与 highPrice 同维度。

M1 为周期参数，1 ≤ M1 < N，整数。

M2 为阈值参数，M2 ≥ 1，整数，M2/100 表示振幅需小于的百分比临界值。

4. **公式:**


```
breakplat = REF(((HHV(highPrice,M1)-LLV(lowPrice,M1))/
                LLV(LOW,M1)),1) <= (M2/100) AND
                closePrice >= REF(HHV(highPrice,M1),1) ;
var = closePrice / REF(HHV(highPrice,M1),1) ;
```

5. 中文含义:

breakplat=昨日((M1 日内最高价的最高值-M1 日内最低价的最低值)/M1 日内最低价的最低值)<=(M2/100) 并且 收盘价>=昨日 M1 日内最高价的最高值

var=收盘价/昨日 M1 日内最高价的最高值

6. 输出参数说明:

breakplat 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到

的条件值矩阵。值=1 表示满足突破长期盘整条件，值=0 表示不满足突破长期盘整条件。

var 是计算结果矩阵，与与输入的价格序列为同型矩阵，表示最新周期收盘

价与前 M1 周期内的最高价的比值矩阵。

7. 示例:

```
>> a = [20.25 20.68 20.30 19.77 19.98 19.50 19.10 19.31 19.30 19.73];
```

```
>> b = [20.25 20.11 19.10 19.10 19.00 19.06 18.65 18.55 18.76 19.10];
```

```
>> c = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [breakplat,var] = BREAKPLAT(a,b,c,1,2)
```

```
breakplat =
```

NaN	1	1	0	0	1	0	1	0	0
-----	---	---	---	---	---	---	---	---	---

var =

NaN	0.9853	0.9853	0.9853	0.9853	0.9853
0.9853	0.9853	0.9853	0.9853	0.9853	

3.2.2. MDFALL-M 周期下跌

函数解释：

1. **用途:** 计算连续 M 周期下跌 MDFALL 选股模型。连续 M 周期下跌，股价可能继续走弱。

2. **函数详细结构:**

[mdfall]= MDFALL(closePrice, M)

[mdfall, var]= MDFALL(closePrice, M)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，N>=3。

M 为周期参数，1 =< M =< N-2，整数。

4. **公式:**

$mdfall = \text{COUNT}(\text{closePrice} < \text{REF}(\text{closePrice}, 1), M) = M;$

$\text{var} = (\text{closePrice} - \text{REF}(\text{closePrice}, M+1)) / \text{REF}(\text{closePrice}, M+1);$

5. **中文含义:**

mdfall=统计 M 日中满足收盘价<昨收的天数=M

var=(收盘价-M+1 日前的收盘价/M+1 日前的收盘价

6. 输出参数说明:

mdfall-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足连续 M 周期下跌条件，值=0 表示不满足连续 M 周期下跌条件。

var-是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内的涨跌幅值矩阵。

7. 示例:

```
>> c = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [mdfall,var] = MDFALL(c,1)
```

```
mdfall =
```

```
0    0    1    1    1    0    1    1    0    0
```

```
var =
```

```
0    0    0    0    0    0    0    0    0    0
```

3.2.3. MDFALLMORE-M 周期下跌多于上涨

函数解释：

1. **用途:** 计算 M 周期下跌多于上涨 MDFALLMORE 选股模型。M 周期内下跌周期数量多于上涨周期数量，股价可能维持弱势。

2. 函数详细结构:

[mdfallmore]= MDFALLMORE(closePrice, M)

[mdfallmore, var1]= MDFALLMORE(closePrice, M)

[mdfallmore, var1, var2]= MDFALLMORE(closePrice, M)

3. 输入参数说明:

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 2$ 。

M 为周期参数， $1 \leq M < N$ ，整数。

4. 公式:

mdfallmore = COUNT(closePrice > REF(closePrice,1),M) <
COUNT(closePrice < REF(closePrice,1),M);
var1 = COUNT(closePrice < REF(closePrice,1),M);
var2 = COUNT(closePrice > REF(closePrice,1),M);

5. 中文含义:

mdfallmore 价>昨收的天数) < (统计 M 日中满足收盘价<昨收的天数)

var1=统计 M 日中满足收盘价<昨收的天数

var2=统计 M 日中满足收盘价>昨收的天数

6. 输出参数说明:

mdfallmore 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得

到的条件值矩阵。值=1 表示满足 M 周期内下跌多于上涨条件，值=0 表示不

满

足 M 周期内下跌多于上涨条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内下跌周

期的数量值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内上涨周

期的数量值矩阵。

8. 示例:

```
>> c = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [mdfallmore,var1,var2] = MDFALLMORE(c,1)
```

mdfallmore =

```
0    0    1    1    1    0    1    1    0    0
```

var1 =

```
NaN    0    1    1    1    0    1    1    0    0
```

var2 =

```
NaN    1    0    0    0    1    0    0    1    1
```

3.2.4. MDGMORE-M 周期阴线多于阳线

函数解释：

1. **用途:** 计算 M 周期阴线多于阳线 MDGMORE 选股模型。M 周期内阴线数量多于阳线数量，股价可能维持弱势。

2. 函数详细结构:

```
[ mdgmore ]= MDGMORE(openPrice, closePrice, M )
```

```
[ mdgmore, var1 ]= MDGMORE(openPrice, closePrice, M )
```

[mdgmore, var1, var2]= MDGMORE(openPrice, closePrice, M)

3. 输入参数说明:

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

M 为周期参数，1 =< M =< N，整数。

4. 公式:

MDGMORE = COUNT(openPrice > closePrice,M) >
COUNT(openPrice < closePrice,M);
VAR1 = COUNT(openPrice < closePrice,M);
VAR2 = COUNT(openPrice > closePrice,M);

5. 中文含义:

MDGMORE=统计 M 日中满足收阴线天数>统计 M 日中满足收阳线天数

diff1=统计 M 日中满足收阳线的天数

diff2=统计 M 日中满足收阴线的天数

6. 输出参数说明:

mdgmore 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到

的条件值矩阵。值=1 表示满足 M 周期内阴线多于阳线条件，值=0 表示不满

足 M 周期内阴线多于阳线条件。

diff1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内阳线

的数量值矩阵。

diff2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内阴线

的数量值矩阵。

7. 示例:

```
>> a = [10.78,10.75,10.74,10.92,10.83;3.23,3.25,3.25,3.29,3.36]
```

```
>> b = [10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [mdgmore,var1,var2] = MDGMORE(a,b,3)
```

mdgmore =

0	0	0	0	1
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

var1 =

NaN	NaN	2	2	1
-----	-----	---	---	---

NaN	NaN	0	1	1
-----	-----	---	---	---

var2 =

NaN	NaN	0	1	2
-----	-----	---	---	---

NaN	NaN	0	0	1
-----	-----	---	---	---

3.2.5. MDGREEN-M 周期收阴线

函数解释：

1. **用途:** 计算连续 M 周期收阴线 MDGREEN 选股模型。连续 M 周期收阴线，

股价可能继续走弱。

2. 函数详细结构:

[mdgreen]= MDGREEN(openPrice, closePrice, M)

[mdgreen, var]= MDGREEN(openPrice, closePrice, M)

3. 输入参数说明:

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=3。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=3。

M 为周期参数，1 =< M =< N-2，整数。

4. 公式:

$$\text{mdgreen} = \text{COUNT}(\text{closePrice} < \text{openPrice}, M) = M;$$
$$\text{var} = (\text{closePrice} - \text{REF}(\text{closePrice}, M+1)) / \text{REF}(\text{closePrice}, M+1);$$

5. 中文含义:

mdgreen=统计 M 日中满足收阴线天数=M

var=(收盘价-M+1 日前的收盘价/M+1 日前的收盘价

6. 输出参数说明:

mdgreen 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到

的条件值矩阵。值=1 表示满足连续 M 周期收阴线条件，值=0 表示不满足连续 M 周期收阴线条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内的涨跌

幅值矩阵。

7. 示例:

```
>> a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];  
  
>> b = [9.94 9.99 10.12 10.34 10.23 10.14;23.9 24.32 24.60 24.33 24.2  
24.00];  
  
>> [mdgreen,var] = MDGREEN(a,b,1)  
  
mdgreen =  
  
      0      0      0      0      1      1  
  
      0      0      0      1      1      1  
  
var =  
  
      NaN      0.0050      0.0054      0.0220     -0.0045     -0.0089  
  
      NaN      0.0423      0.0117     -0.0270     -0.0053     -0.0198
```

3.2.6. MDHIGH-M 周期创新高

函数解释：

1. **用途:** 计算 M 周期内创新高 MDHIGH 选股模型。M 周期内创出历史新高，股价有望继续走强。

2. 函数详细结构:

```
[ mdhighPrice, var1, var2 ]= MDHIGH(highPrice, M )
```

3. 输入参数说明:

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度，

$N \geq 1$ 。

M 为周期参数， $1 \leq M \leq N$ ，整数。

4. 公式:

$MDHIGH = HHV(highPrice, M) == HHV(highPrice, 0) \text{ AND}$
 $BARSCOUNT(highPrice) \geq M;$
 $VAR1 = REF(HHV(highPrice, 0), 1);$
 $VAR2 = HHV(highPrice, M);$

5. 中文含义:

MDHIGHPRICE=M 日内最高价的最高值=0 日内最高价的最高值并且最高价
的有效数据周期数 $\geq M$

diff1=昨日 0 日内最高价的最高值

diff2=M 日内最高价的最高值

6. 输出参数说明:

MDHIGHPRICE 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的

条件值矩阵。值=1 表示满足 M 周期内创新高条件，值=0 表示不满足 M 周期

内创新高条件。

diff1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示截止上周期的历史最高价矩阵。

diff2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新的 M 周期内的最高价矩阵。

7. 示例:

>> a=[10.78,11,10.98,11.08,10.88;3.23,3.29,3.25,3.41,3.44;]

```
>> [mdhighPrice,var1,var2] = MDHIGH (a,4)

mdhighPrice =

    0    0    0    1    1
    0    0    0    1    1

var1 =

    10.7800    11.0000    11.0000    11.0800    11.0800
     3.2300     3.2900     3.2900     3.4100     3.4400

var2 =

     NaN     NaN     NaN    11.0800    11.0800
     NaN     NaN     NaN     3.4100     3.4400
```

3.2.7. MDLOW-M 周期创新低

函数解释：

1. **用途:** 计算 M 周期内创新低 MDLOW 选股模型。M 周期内创出历史新低，股价可能维持弱势。

2. **函数详细结构:**

```
[ mdlowPrice ]= MDLOW (lowPrice, M )
```

```
[ mdlowPrice, var1 ]= MDLOW (lowPrice, M )
```

```
[ mdlowPrice, var1, var2 ]= MDLOW (lowPrice, M )
```

3. **输入参数说明:**

lowPrice 为输入的最低价序列 ,为 N 列数值矩阵。N 为提取的时间长度 , $N \geq 1$ 。

M 为周期参数 , $1 \leq M \leq N$, 整数。

4. 公式:

$$\text{MDLOW} = \text{LLV}(\text{lowPrice}, M) == \text{LLV}(\text{lowPrice}, 0) \text{ AND}$$
$$\text{BARSCOUNT}(\text{lowPrice}) \geq M;$$
$$\text{VAR1} = \text{REF}(\text{LLV}(\text{lowPrice}, 0), 1);$$
$$\text{VAR2} = \text{LLV}(\text{lowPrice}, M);$$

5. 中文含义:

MDLOW=M 日内最低价的最低值=0 日内最低价的最低值并且最低价的有效
数据周期数>=M

diff1=昨日 0 日内最低价的最低值

diff2=M 日内最低价的最低值

6. 输出参数说明:

mdlowPrice-是计算结果矩阵,与输入的价格序列为同型矩阵,表示计算得到的
条

件值矩阵。值=1 表示满足 M 周期内创新低条件,值=0 表示不满足 M 周期内创
新

低条件。

diff1-是计算结果矩阵,与输入的价格序列为同型矩阵,表示截止上周期的历史
最低价矩阵。

diff2-是计算结果矩阵,与输入的价格序列为同型矩阵,表示最新的 M 周期内的
最低价矩阵。

8. 示例:

```
>> a=[10.78,10.63,10.67,10.7,10.55;3.23,3.18,3.25,3.25,3.28;]
```

```
>> [mdlowPrice,var1,var2] = MDLOW(lowPrice,period)
```

mdlowPrice =

0	0	1	1	1
---	---	---	---	---

0	0	1	1	0
---	---	---	---	---

var1 =

10.7800	10.6300	10.6300	10.6300	10.5500
---------	---------	---------	---------	---------

3.2300	3.1800	3.1800	3.1800	3.1800
--------	--------	--------	--------	--------

var2 =

NaN	NaN	10.6300	10.6300	10.5500
-----	-----	---------	---------	---------

NaN	NaN	3.1800	3.1800	3.2500
-----	-----	--------	--------	--------

3.2.8. MDRED-M 周期收阳线

函数解释：

1. **用途:** 计算连续 M 周期收阳线 MDRED 选股模型。连续 M 周期收阳线，股价有望继续走强。

2. 函数详细结构:

```
[ mdred ]= MDRED(openPrice, closePrice, M )
```

```
[ mdred, var ]= MDRED(openPrice, closePrice, M )
```

3. 输入参数说明:

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=3。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=3。

M 为周期参数， $1 \leq M \leq N-2$ ，整数。

4. 公式:

$$\text{mdred} = \text{COUNT}(\text{closePrice} > \text{openPrice}, M) == M;$$
$$\text{var} = (\text{closePrice} - \text{REF}(\text{closePrice}, M+1)) / \text{REF}(\text{closePrice}, M+1);$$

5. 中文含义:

mdred=统计 M 日中满足收阳线的天数=M

var=(收盘价-M+1 日前的收盘价/M+1 日前的收盘价

6. 输出参数说明:

mdred 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足连续 M 周期收阳线条件，值=0 表示不满足连续 M 周期收阳线条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内的涨跌幅值矩阵。

7. 示例:

```
>> a = [9.8 9.95 9.87 10.02 10.34 10.28;23.6 23.98 24.56 24.57 24.29 24.15];
```

```
>> b = [9.94 9.99 10.12 10.34 10.23 10.14;23.9 24.32 24.60 24.33 24.2
```

24.00];

```
>> [mdred,var] = MDRED(a,b,1)
```

mdred =

1	1	1	1	0	0
1	1	1	0	0	0

var =

NaN	0.0050	0.0054	0.0220	-0.0045	-0.0089
NaN	0.0423	0.0117	-0.0270	-0.0053	-0.0198

3.2.9. MDRISE-M 周期上涨

函数解释：

1. **用途：**计算连续 M 周期上涨 MDRISE 选股模型。连续 M 周期上涨，股价有望继续走强。

2. **函数详细结构：**

```
[mdrise]=MDRISE(closePrice,M)
```

```
[mdrise,var]=MDRISE(closePrice,M)
```

3. **输入参数说明：**

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=3。

M 为周期参数，1<=M<=N-2，整数。

4. 公式:

$$\begin{aligned} \text{mdrise} &= \text{COUNT}(\text{closePrice} > \text{REF}(\text{closePrice}, 1), M) = M; \\ \text{var} &= (\text{closePrice} - \text{REF}(\text{closePrice}, M+1)) / \text{REF}(\text{closePrice}, M+1); \end{aligned}$$

5. 中文含义:

mdrise=统计 M 日中满足收盘价>昨收的天数=M

$$\text{var} = (\text{收盘价} - M + 1 \text{ 日前的收盘价}) / M + 1 \text{ 日前的收盘价}$$

6. 输出参数说明:

mdrise 是计算结果矩阵，与输入的价格序列为同型矩阵，表计算得到的条

件值矩阵。值=1 表示满足连续 M 周期上涨条件, 值=0 表示不满足连续 M 周

期上涨条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内的涨跌

幅值矩阵。

7. 示例:

```
>> c = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [mdrise,var] = MDRISE(c,1)
```

mdrise =

0 1 0 0 0 1 0 0 1 1

```
var =
```

0 0 0 0 0 0 0 0 0 0

3.2.10.MDRISEMORE-M 周期上涨多于下跌

函数解释：

1. **用途:** 计算 M 周期上涨多于下跌 MDRISEMORE 选股模型。M 周期内上涨周期数量多于下跌周期数量，股价有望维持强势。

2. **函数详细结构:**

[mdrisemore]= MDRISEMORE(closePrice, M)

[mdrisemore, var1]= MDRISEMORE(closePrice, M)

[mdrisemore, var1, var2]= MDRISEMORE(closePrice, M)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 2$ 。

M 为周期参数， $1 \leq M < N$ ，整数。

4. **公式:**

mdrisemore = COUNT(closePrice > REF(closePrice,1),N) >
COUNT(closePrice < REF(closePrice,1),N);

var1 = COUNT(closePrice < REF(closePrice,1),N);

var2 = COUNT(closePrice > REF(closePrice,1),N);

5. **中文含义:**

mdrisemore=统计 N 日中满足收盘价>昨收的天数>统计 N 日中满足收盘价
<昨收的天数

var1=统计 N 日中满足收盘价<昨收的天数

var2=统计 N 日中满足收盘价>昨收的天数

6. **输出参数说明:**

mdrisemore 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足 M 周期内上涨多于下跌条件，值=0 表示不满足 M 周期内上涨多于下跌条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内下跌周期的数量值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内上涨周期的数量值矩阵。

7. 示例:

```
>> c = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];  
>> [mdrisemore,var1, var2] = MDRISEMORE(c,1)  
  
mdrisemore =  
  
      0      1      0      0      0      1      0      0      1      1  
  
var1 =  
  
    NaN      0      1      1      1      0      1      1      0      0  
  
var2 =  
  
    NaN      1      0      0      0      1      0      0      1      1
```

3.2.11.MDRMORE-M 周期阳线多于阴线

函数解释：

1. **用途:** 计算 M 周期阳线多于阴线 MDRMORE 选股模型。M 周期内阳线数量多于阴线数量，股价有望维持强势。

2. **函数详细结构:**

[mdrmore]= MDRMORE(openPrice, closePrice, M)

[mdrmore, var1]= MDRMORE(openPrice, closePrice, M)

[mdrmore, var1, var2]= MDRMORE(openPrice, closePrice, M)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

M 为周期参数，1 =< M =< N，整数。

4. **公式:**

$MDRMORE = COUNT(openPrice > closePrice, M) <$

$COUNT(openPrice < closePrice, M);$

$VAR1 = COUNT(openPrice < closePrice, M);$

$VAR2 = COUNT(openPrice > closePrice, M);$

5. **中文含义:**

MDRMORE=统计 M 日中满足收阴线的天数<统计 M 日中满足收阳线的天数

diff1=统计 M 日中满足收阳线的天数

diff2=统计 M 日中满足收阴线的天数

6. 输出参数说明:

mdrmore 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足 M 周期内阳线多于阴线条件，值=0 表示不满足 M 周期内阳线多于阴线条件。

diff1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内阳线的数量值矩阵。

diff2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M 周期内阴线的数量值矩阵。

7. 示例:

```
>> a = [10.78,10.75,10.74,10.92,10.83;3.23,3.25,3.25,3.29,3.36]
```

```
>> b = [10.78,10.86,10.95,10.88,10.64;3.23,3.25,3.25,3.39,3.32]
```

```
>> [mdrmore,var1,var2] = MDRMORE(a,b,3)
```

mdrmore =

0	0	1	1	0
0	0	0	1	0

var1 =

NaN	NaN	2	2	1
NaN	NaN	0	1	1

var2 =

NaN	NaN	0	1	2
NaN	NaN	0	0	1

3.2.12.PLATFORM-平台整理

函数解释：

1. **用途:** 计算平台整理 PLATFORM 选股模型。成交量分布比较均匀，偶尔有所放大，但马上又缩小。如果发生在底部区域，且盘整时间较长，则很有可能是庄家的吸货行为。

2. **函数详细结构:**

[platform]= PLATFORM(closePrice, M1, M2)

[platform, var]= PLATFORM(closePrice, M1, M2)

3. **输入参数说明:**

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

M1 为周期参数，1 ≤ M1 ≤ N，整数。

M2 为阈值参数，M2 ≥ 1，整数，M2/100 表示振幅需小于的百分比临界值。

4. **公式:**

$$\begin{aligned} \text{platform} &= (\text{HHV}(\text{closePrice}, \text{M1}) - \text{LLV}(\text{closePrice}, \text{M1})) / \\ &\quad \text{LLV}(\text{closePrice}, \text{M1}) \leq (\text{M2}/100); \\ \text{var} &= \text{HHV}(\text{closePrice}, \text{M1}) - \text{LLV}(\text{closePrice}, \text{M1}) / \text{LLV}(\text{closePrice}, \text{M1}); \end{aligned}$$

5. **中文含义:**

platform=(M1 日内收盘价的最高值-M1 日内收盘价的最低值)/M1 日内收盘价的最低值≤(M2/100)

var=(M1 日内收盘价的最高值-M1 日内收盘价的最低值)/M1 日内收盘价的最低值

6. 输出参数说明:

platform 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足平台整理条件，值=0 表示不满足平台整理条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示 M1 周期内的振幅值矩阵。

7. 示例:

```
>> c = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [platform,var] =PLATFORM(c,1,1)
```

```
platform =
```

```
1    1    1    1    1    1    1    1    1    1
```

```
var =
```

```
0    0    0    0    0    0    0    0    0    0
```

3.2.13.VOLUMEUP-突然放量

函数解释：

1. **用途:** 计算突然放量 VOLUMEUP 选股模型。成交量突然放大，股票开始受到主力关注。

2. **函数详细结构:**

[VolumeUp]= VOLUMEUP(TradingVolume, M1, M2)

[VolumeUp, var]= VOLUMEUP(TradingVolume, M1, M2)

3. 输入参数说明:

TradingVolume 为输入的成交量序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 2$ 。

M1 为周期参数， $1 \leq M1 < N$ ，整数。

M2 为阈值参数， $M2 \geq 1$ ，整数，表示成交量放大倍数。

4. 公式:

$\text{volumeup} = \text{tradingVolume} > \text{REF}(\text{HHV}(\text{tradingVolume}, M1), 1) * M2;$

$\text{var} = \text{tradingVolume} / \text{REF}(\text{HHV}(\text{tradingVolume}, M1), 1);$

5. 中文含义:

VolumeUp=成交量(手)>昨日 M1 日内成交量(手)的最高值*M2

var=成交量/昨日 M1 日内成交量的最高值

6. 输出参数说明:

VolumeUp 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到

的条件值矩阵。值=1 表示满足突然放量条件，值=0 表示不满足突然放量条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期成交量

与前 M1 周期内成交量最大值的比值矩阵。

7. 示例:

```
>> c = [10987654 13643303 13245654 20954432 18094333
15237873;4559934 8959874 7645532 6572344 3455221
6984451];

>> [VolumeUp,var] = VOLUMEUP(c,2,1)

VolumeUp =

    NaN    0    0    1    0    0
    NaN    0    0    0    0    1

var =

    NaN    NaN    0.9709    1.5359    0.8635    0.7272
    NaN    NaN    0.8533    0.7335    0.4519    1.062
```

3.3. K 线形态选股模型

3.3.1. BALDGREEN-光脚阴线

函数解释：

1. **用途:** 计算光脚阴线 BALDGREEN 选股模型。一般来说，如果在低价位区域出现光脚阴线，表明买方开始聚积上攻的能量，但卖方仍占有优势。如果在高价位区域出现光脚阴线，表明买方上攻的能量已经衰竭，卖方的做空能量不断增强，且占据主动地位，行情有可能在此发生逆转。

2. 函数详细结构:

```
[ baldgreen, var ]= BALDGREEN(highPrice, lowPrice, closePrice )
```

3. 输入参数说明:

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

lowPrice 为输入的最低价序列，为 N 列数值矩阵。N 为提取的时间长度，N>=1。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

4. 公式:

$$\begin{aligned} \text{baldgreen} &= \text{ABS}(\text{lowPrice} - \text{closePrice}) / \text{closePrice} < 0.005 \\ &\quad \text{AND } \text{highPrice} > \text{lowPrice}; \\ \text{var} &= \text{ABS}(\text{lowPrice} - \text{closePrice}) / \text{closePrice}; \end{aligned}$$

5. 中文含义:

baldgreen=最低价-收盘价的绝对值/收盘价<0.005 并且 最高价>最低价

var=最低价-收盘价的绝对值/收盘价

6. 输出参数说明:

BALDGREEN 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足光脚阴线条件，值=0 表示不满足光脚阴线条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的最低价减去收盘价的绝对值比上收盘价的值矩阵。

7. 示例:

>>closePrice = [2.3 3.6 3.3 2.5];highPrice = [3 3.9 3.7 3.1];lowPrice = [2 2.1 2 2];

```
>> [baldgreen,var] = BALDGREEN(highPrice,lowPrice,closePrice)
```

```
baldgreen =
```

```
0      0      0      0
```

```
var =
```

```
0.1304    0.4167    0.3939    0.2000
```

3.3.2. BALDRED-光头阳线

函数解释：

1. **用途:** 计算光头阳线 BALDRED 选股模型。光头阳线是指没有上影线的 K 线。

光头阳线若出现在低价位区域，预示为一轮上升行情的开始。如果出现在上升行情途中，表明后市继续看好。

2. **函数详细结构:**

```
[ baldred, var ]= BALDRED(highPrice, lowPrice, closePrice )
```

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

lowPrice 为输入的最低价序列 ,为 N 列数值矩阵。N 为提取的时间长度 , $N \geq 1$ 。

closePrice 为输入的收盘价序列 , 为 N 列数值矩阵。N 为提取的时间长度 ,
 $N \geq 1$ 。

4. 公式:

```
baldred = ABS(highPrice- closePrice)/closePrice<0.005  
          AND highPrice > lowPrice;  
var= ABS(highPrice - closePrice)/closePrice;
```

5. 中文含义:

baldred=最高价-收盘价的绝对值/收盘价<0.005 并且 最高价>最低价

var=最高价-收盘价的绝对值/收盘价

6. 输出参数说明:

BALDRED 是计算结果矩阵 , 与输入的价格序列为同型矩阵 , 表示计算得到的
条件值矩阵。值=1 表示满足光头阳线条件 , 值=0 表示不满足光头阳线条件。

var 是计算结果 , 表示最新周期的最高价减去收盘价的绝对值比上收盘价。

7. 示例:

```
>>closePrice = [2.3 3.6 3.3 2.5];highPrice = [3 3.9 3.7 3.1];lowPrice = [2 2.1 2 2];
```

```
>>[baldred,var] = BALDRED(highPrice,lowPrice,closePrice)
```

baldred =

```
0    0    0    0
```

var =

0.3043 0.0833 0.1212 0.2400

3.3.3. COUNTERATTACK-好友反攻

函数解释：

1. **用途:** 计算好友反攻 COUNTERATTACK 选股模型。好友反攻出现在下跌趋势中，由一阴一阳两根 k 线组成，先是一根大阴线，接着跳空低开，收出一根大阳线或中阳线，阳线收盘价与前一根阴线收盘价相同或接近。好友反攻是见底信号，后市看涨。

2. **函数详细结构:**

[counterattack, var1, var2]= COUNTERATTACK(openPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=2。

closePrice 为输入的收盘价序列，为 N 列数值矩阵，N>=2。

4. **公式:**

counterattack = (REF(closePrice,1) < REF(openPrice,1) AND
closePrice > openPrice AND ABS(closePrice-REF(closePrice,1))/closePrice < 0.0
OR (REF(closePrice,1) > REF(openPrice,1) AND closePrice < openPrice AND
ABS(closePrice-REF(closePrice,1))/closePrice < 0.002;
var1 = REF(closePrice,1)/REF(openPrice,1);
var2 = ABS(closePrice - REF(closePrice,1))/closePrice;

5. 中文含义:

$\text{counterattack} = \text{昨日收盘价} < \text{昨日开盘价}$ 并且

$\text{收盘价} > \text{开盘价}$ 并且 $\text{收盘价} - \text{昨日收盘价的绝对值} / \text{收盘价} < 0.002$

或者

$\text{昨日收盘价} > \text{昨日开盘价}$ 并且 $\text{收盘价} < \text{开盘价}$ 并且

$\text{收盘价} - \text{昨日收盘价的绝对值} / \text{收盘价} < 0.002$

$\text{var1} = \text{昨日收盘价} / \text{昨日开盘价}$

$\text{var2} = \text{收盘价} - \text{昨日收盘价的绝对值} / \text{收盘价}$

6. 输出参数说明:

COUNTERATTACK 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计

算得到的条件值矩阵。值=1 表示满足好友反攻条件，值=0 表示不满足好友

反

攻条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示前一周期的收盘

价与开盘价的比值矩阵。

var2 是计算结果，与输入的价格序列为同型矩阵，表示最新周期的收盘价

减去前一周期收盘价的绝对值比上最新周期收盘价的值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7];closePrice = [2.3 3.6 3.3 2.5];
```

```
>>[counterattack,var1,var2] = COUNTERATTACK(openPrice,closePrice)
```

```
counterattack =
```

```
NaN    0    0    0
```

```
var1 =
```

```
NaN    0.9583    1.4400    1.5714
```

```
var2 =
```

```
NaN    0.3611    0.0909    0.3200
```

3.3.4. CROSSHEADFOOT-穿头破脚

函数解释：

1. **用途:** 计算穿头破脚 CROSSHEADFOOT 选股模型。底部穿头破脚是股价回

升的信号。顶部穿头破脚是股价见顶回落的信号。

2. 函数详细结构:

[crossheadfoot]= CROSSHEADFOOT(openPrice, closePrice)

[crossheadfoot, var1]= CROSSHEADFOOT(openPrice, closePrice)

[crossheadfoot, var1, var2]= CROSSHEADFOOT(openPrice, closePrice)

3. 输入参数说明:

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=2。

closePrice 为输入的收盘价序列，为 N 列数值矩阵，N>=2。

4. 公式:

crossheadfoot = (REF(closePrice,1)/REF(openPrice,1) > 1.03 AND
closePrice/openPrice < 0.96 AND closePrice < REF(openPrice,1)
AND openPrice > REF(closePrice,1)OR
(REF(closePrice,1)/REF(openPrice,1) < 0.97 AND) closePrice/openPrice >
AND closePrice > REF(openPrice,1) AND openPrice < REF(closePrice,1)
var1 = REF(closePrice,1)/REF(openPrice,1);
var2 = closePrice/openPrice;

5. 中文含义:

crossheadfoot=(昨收/昨日开盘价>1.03 并且 收盘价/开盘价<0.96 并且
收盘价<昨日开盘价 并且 开盘价>昨收) 或者 (昨收/昨日开盘价<0.97 并
且 收盘价/开盘价>1.04 并且 收盘价>昨日开盘价 并且 开盘价<昨收)
var1=昨收/昨日开盘价
var2=收盘价/开盘价

6. 输出参数说明:

crossheadfoot 是计算结果矩阵，与输入的价格序列为同型矩阵，表示

头

计算得到的条件值矩阵。值=1 表示满足穿头破脚条件，值=0 表示不满足穿

破脚条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上一周期的收盘

价与开盘价的比值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价与开盘价的比值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [crossheadfoot,var1,var2] = CROSSHEADFOOT(a,d)
```

crossheadfoot =

NaN	0	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---	---

var1 =

NaN	1.0000	0.9995	0.9995	1.0015	1.0026
0.9984	1.0005	0.9963	0.9974		

var2 =

NaN	0.9995	0.9995	1.0015	1.0026	0.9984
-----	--------	--------	--------	--------	--------

1.0005 0.9963 0.9974 1.0015

3.3.5. CROSSSTAR-十字星

函数解释：

1. **用途:** 计算十字星 CROSSSTAR 选股模型。十字星通常是将要变盘的信号。

2. **函数详细结构:**

[crosstar]= CROSSSTAR(openPrice, highPrice, lowPrice, closePrice)

[crosstar, var]= CROSSSTAR(openPrice, highPrice, lowPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，

N>=1。

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度，

N>=1。

lowPrice 为输入的最低价序列，为 N 列数值矩阵。N 为提取的时间长度，N>=1。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，

N>=1。

4. **公式:**

crosstar = ABS(closePrice-openPrice)/openPrice< 0.01 AND

highPrice>closePrice AND lowPrice<closePrice;

var = ABS(closePrice-openPrice)/openPrice ;

5. **中文含义:**

crosstar=收盘价-开盘价的绝对值/开盘价<0.01 并且 最高价>收盘价 并

且 最低价<收盘价

var=收盘价-开盘价的绝对值/开盘价

6. 输出参数说明:

crossstar 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到

的条件值矩阵。值=1 表示满足十字星条件，值=0 表示不满足十字星条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价

减开盘价的绝对值与开盘价的比值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> b = [20.25 20.68 20.30 19.77 19.98 19.50 19.10 19.31 19.30 19.73];
```

```
>> c = [20.25 20.11 19.10 19.10 19.00 19.06 18.65 18.55 18.76 19.10];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [crossstar,var] = CROSSSTAR(a,b,c,d)
```

crossstar =

0	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---

var =

0	0.0005	0.0005	0.0015	0.0026	0.0016
0.0005	0.0037	0.0026	0.0015		

3.3.6. DARKCLOUD-乌云盖顶

函数解释：

1. **用途:** 计算乌云盖顶 DARKCLOUD 选股模型。乌云盖顶经常发生在一段上升行情的顶部，由一阴一阳两根 K 线组成，是一个看跌反转信号。

2. **函数详细结构:**

[darkcloud, var1, var2]= DARKCLOUD(openPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 2$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 2$ 。

4. **公式:**

darkcloud = REF(closePrice,1)/REF(openPrice,1) > 1.03 AND
closePrice/openPrice < 0.97 AND
openPrice > REF(closePrice,1) AND
closePrice < REF(closePrice,1);
var1 = REF(closePrice, 1)/REF(openPrice 1);
var2 = closePrice/openPrice;

5. **中文含义:**

darkcloud=昨收/昨日开盘价>1.03 并且 收盘价/开盘价<0.97 并且 开盘价>昨收 并且 收盘价<昨日开盘价

var1=昨收/昨日开盘价

var2=收盘价/开盘价

6. **输出参数说明:**

DARKCLOUD 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得

到的条件值矩阵。值=1 表示满足乌云盖顶条件，值=0 表示不满足乌云盖顶条件。

件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上一周期的收盘价与开盘价的比值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价与开盘价的比值矩阵。

7. 示例:

```
>> openPrice = [2.4 2.5 2.1];closePrice = [2.3 3.6 3.3];
```

```
>> [darkcloud,var1,var2] = DARKCLOUD(openPrice,closePrice)
```

darkcloud =

NaN	0	0
-----	---	---

var1 =

NaN	0.9583	1.4400
-----	--------	--------

var2 =

NaN 1.4400 1.5714

3.3.7. DEADCROSS-垂死十字星

函数解释：

1. **用途:** 计算垂死十字星 DEADCROSS 选股模型。垂死十字星显示上方抛售压力很大，是股价走弱的信号。

2. **函数详细结构:**

```
[ deadcross ]= DEADCROSS(openPrice, highPrice, lowPrice, closePrice )
```

```
[ deadcross, var1 ]= DEADCROSS(openPrice, highPrice, lowPrice,  
closePrice )
```

```
[ deadcross, var1, var2 ]= DEADCROSS(openPrice, highPrice, lowPrice,  
closePrice )
```

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

lowPrice 为输入的最低价序列 ,为 N 列数值矩阵。N 为提取的时间长度 , $N \geq 1$ 。

closePrice 为输入的收盘价序列 , 为 N 列数值矩阵。N 为提取的时间长度 ,
 $N \geq 1$ 。

4. 公式:

$$\text{deadcross} = \text{ABS}(\text{closePrice} - \text{openPrice}) / \text{openPrice} < 0.01 \text{ AND}$$
$$\text{ABS}(\text{closePrice} - \text{lowPrice}) / \text{lowPrice} < 0.01 \text{ AND closePrice} < \text{highPrice};$$
$$\text{var1} = \text{ABS}(\text{closePrice} - \text{openPrice}) / \text{openPrice};$$
$$\text{var2} = \text{ABS}(\text{closePrice} - \text{lowPrice}) / \text{lowPrice};$$

5. 中文含义:

deadcross=收盘价-开盘价的绝对值/开盘价<0.01 并且 收盘价-最低价的

绝对值/最低价<0.01 并且 收盘价<最高价

var1=收盘价-开盘价的绝对值/开盘价

var2=收盘价-最低价的绝对值/最低价

6. 输出参数说明:

deadcross 是计算结果矩阵 , 与输入的价格序列为同型矩阵 , 表示计算得

到的条件值矩阵。值=1 表示满足垂死十字星条件 , 值=0 表示不满足垂死十

字

星条件。

var1 是计算结果矩阵 , 与输入的价格序列为同型矩阵 , 表示最新周期的收盘

价减开盘价的绝对值与开盘价的比值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价减最低价的绝对值与最低价的比值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];  
>> b = [20.25 20.68 20.30 19.77 19.98 19.50 19.10 19.31 19.30 19.73];  
>> c = [20.25 20.11 19.10 19.10 19.00 19.06 18.65 18.55 18.76 19.10];  
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];  
>> [ deadcross, var1, var2 ]= DEADCROSS(a,b,c,d)
```

deadcross =

0	0	0	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---

var1 =

0	0.0005	0.0005	0.0015	0.0026	0.0016
0.0005	0.0037	0.0026	0.0015		

var2 =

0	0.0144	0.0209	0.0199	0.0079	0.0073
0.0113	0.0081	0.0181	0.0319		

3.3.8. DOUBLECROWS-双飞乌鸦

函数解释：

1. **用途:** 计算双飞乌鸦 DOUBLECROWS 选股模型。双飞乌鸦是在大阳线后连续两根阴线，表明后市看淡。

2. **函数详细结构:**

[doublecrows, var1, var2]= DOUBLECROWS(openPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=2。

closePrice 为输入的收盘价序列，为 N 列数值矩阵，N>=2。

4. **公式:**

doublecrows = REF(closePrice,1) < REF(openPrice,1) AND
closePrice < openPrice AND closePrice/openPrice < 0.98;
var1 = REF(closePrice,1)/REF(openPrice,1);
var2 = closePrice/openPrice;

5. **中文含义:**

doublecrows=昨收<昨日开盘价 并且 收阴线 并且 收盘价/开盘价<0.98

var1=昨收/昨日开盘价

var2=收盘价/开盘价

6. **输出参数说明:**

DOUBLECROWS-是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足双飞乌鸦条件，值=0 表示不满足双飞乌鸦条件。

var1-是计算结果矩阵，与输入的价格序列为同型矩阵，表示上一周期的收盘

价与开盘价的比值矩阵。

var2-是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价与开盘价的比值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7];closePrice = [2.3 3.6 3.3 2.5];
```

```
>>[doublecrows,var1,var2] = DOUBLECROWS(openPrice,closePrice)
```

doublecrows =

NaN	0	0	0
-----	---	---	---

var1 =

NaN	0.9583	1.4400	1.5714
-----	--------	--------	--------

var2 =

NaN	1.4400	1.5714	0.9259
-----	--------	--------	--------

3.3.9. DOWNSHADOW-长下影

函数解释：

1. **用途:** 计算长下影 DOWNSHADOW 选股模型。长下影线表明低位承接力强，通常是止跌的信号。

2. **函数详细结构:**

[downshadow, var]= DOWNSHADOW(openPrice, highPrice, lowPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

4. **公式:**

$$\text{downshadow} = (\text{MIN}(\text{closePrice}, \text{openPrice}) - \text{lowPrice}) / (\text{HIGH} - \text{lowPrice}) > 0.667;$$
$$\text{var} = (\text{MIN}(\text{closePrice}, \text{openPrice}) - \text{lowPrice}) / (\text{highPrice} - \text{lowPrice});$$

5. **中文含义:**

$$\text{downshadow} = (\text{收盘价和开盘价的较小值} - \text{最低价}) / (\text{最高价} - \text{最低价}) > 0.667$$
$$\text{var} = (\text{收盘价和开盘价的较小值} - \text{最低价}) / (\text{最高价} - \text{最低价})$$

6. **输出参数说明:**

DOWNSHADOW 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算

得到的条件值矩阵。值=1 表示满足长下影条件，值=0 表示不满足长下影条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价和开盘价的较小者与最低价的差比上最高价与最低价的差值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7];closePrice = [2.3 3.6 3.3 2.5];highPrice = [3 3.9  
3.7 3.1];lowPrice = [2 2.1 2 2];
```

```
>>[downshadow,var] =
```

```
DOWNSHADOW(openPrice,highPrice,lowPrice,closePrice)
```

```
downshadow =
```

```
0      0      0      0
```

```
var =
```

```
0.3000    0.2222    0.0588    0.4545
```

3.3.10.DOWNSTEPS-下跌三部曲

函数解释：

1. **用途:** 计算下跌三部曲 DOWNSTEPS 选股模型。一般情况下，在下跌三部曲出现之后，股价都会形成一轮较大的跌势。

2. **函数详细结构:**

[downsteps, var1, var2]= DOWNSTEPS(openPrice, highPrice, lowPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

highPrice 为输入的最高价序列，为 N 列数值矩阵， $N \geq 5$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵， $N \geq 5$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

4. **公式:**

```

downsteps = REF(closePrice,4)/REF(openPrice,4) < 0.97 AND
REF(closePrice,3) > REF(openPrice,3) AND
REF(closePrice,2) > REF(openPrice,2) AND
REF(closePrice,1) > REF(openPrice,1) AND
REF(lowPrice ,4) < REF(lowPrice ,3) AND
REF(lowPrice ,4) < REF(lowPrice ,2) AND
REF(lowPrice ,4) < REF(lowPrice ,1) AND
REF(highPrice ,4) > REF(highPrice ,3) AND
REF(highPrice ,4) > REF(highPrice ,2) AND
REF(highPrice ,4) > REF(highPrice ,1) AND
closePrice/openPrice < 0.97 AND
closePrice < REF(closePrice,4);
var1=closePrice/REF(closePrice,4);
var2=closePrice/openPrice;

```

5. 中文含义:

downsteps=4 日前的收盘价/4 日前的开盘价<0.97 并且

3 日前的收盘价>3 日前的开盘价 并且

2 日前的收盘价>2 日前的开盘价 并且

昨日收盘价>昨日开盘价 并且

4 日前的最低价<3 日前的最低价 并且

4 日前的最低价<2 日前的最低价 并且

4 日前的最低价<昨日最低价 并且

4 日前的最高价>3 日前的最高价 并且

4 日前的最高价>2 日前的最高价 并且

4 日前的最高价>昨日最高价 并且

收盘价/开盘价<0.97 并且

收盘价<4 日前的收盘价

var1=收盘价/4 日前的收盘价

var2=收盘价/开盘价

6. 输出参数说明:

DOWNSTEPS 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足下跌三部曲条件，值=0 表示不满足下跌三部曲条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价比上 4 个周期前的收盘价的值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价与开盘价的比值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7 2.6];closePrice = [2.3 3.6 3.3 2.5 3];highPrice = [3  
3.9 3.7 3.1 3.2];lowPrice = [2 2.1 2 2 2.4];
```

```
>>[downsteps,var1,var2] =
```

```
DOWNSTEPS(openPrice,highPrice,lowPrice,closePrice)
```

```
downsteps =
```

```
NaN    NaN    NaN    NaN    0
```

var1 =

NaN	NaN	NaN	NaN	1.3043
-----	-----	-----	-----	--------

var2 =

NaN	NaN	NaN	NaN	1.1538
-----	-----	-----	-----	--------

3.3.11.EVENINGCROSS-黄昏十字星

函数解释：

1. **用途:** 计算黄昏十字星 EVENINGCROSS 选股模型。黄昏十字星通常是将要见顶回落的信号。

2. **函数详细结构:**

[eveningcross]= EVENINGCROSS(openPrice, closePrice)

[eveningcross, var1]= EVENINGCROSS(openPrice, closePrice)

[eveningcross, var1, var2]= EVENINGCROSS(openPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=3。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 3$ 。

4. 公式:

```
eveningcross = REF(closePrice,2)/REF(openPrice,2) > 1.05 AND  
              REF(openPrice,1) > REF(closePrice,2) AND  
              ABS(REF(openPrice,1) - REF(closePrice,1)) < 0.01 AND  
              closePrice/openPrice < 0.95 AND closePrice < REF(closePrice,2);  
var1 = ABS(REF(openPrice,1) - REF(closePrice,1)) ;  
var2 = closePrice / openPrice;
```

5. 中文含义:

eveningcross=2 日前的收盘价/2 日前的开盘价>1.05 并且 昨日开盘价>2
日前的收盘价 并且 昨日开盘价-昨收的绝对值<0.01 并且 收盘价/开盘价
<0.95 并且 收盘价<2 日前的收盘价

var1=昨日开盘价-昨收的绝对值

var2=收盘价/开盘价

6. 输出参数说明:

eveningcross 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算
得到的条件值矩阵。值=1 表示满足黄昏十字星条件，值=0 表示不满足黄昏十
字星条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示前一周期的收盘
价减开盘价的绝对值矩阵。

var2 是计算结果，与输入的价格序列为同型矩阵，表示最新周期的收盘价与

开盘价的比值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [eveningcross,var1,var2] = EVENINGCROSS(a,d)
```

eveningcross =

	NaN	NaN	0	0	0	0	0	0	0
0									

var1 =

NaN	NaN	0.0100	0.0100	0.0300	0.0500
0.0300	0.0100	0.0700	0.0500		

var2 =

NaN	NaN	0.9995	1.0015	1.0026	0.9984
1.0005	0.9963	0.9974	1.0015		

3.3.12.EVENINGSTAR-黄昏之星

函数解释：

1. **用途:** 计算黄昏之星 EVENINGSTAR 选股模型。黄昏之星通常是将要见顶回落的信号。

2. **函数详细结构:**

[eveningstar]= EVENINGSTAR(openPrice, closePrice)

[eveningstar, var1]= EVENINGSTAR(openPrice, closePrice)

[eveningstar, var1, var2]= EVENINGSTAR(openPrice, closePrice)

3. 输入参数说明:

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 3$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 3$ 。

4. 公式:

eveningstar = REF(closePrice,2)/REF(openPrice,2) > 1.05 AND
REF(openPrice,1) > REF(closePrice,2) AND
ABS(REF(openPrice,1)-REF(closePrice,1))/REF(closePrice,1) < 0.01 AND
closePrice/openPrice < 0.95 AND closePrice < REF(closePrice,2);
var1 = ABS(REF(openPrice,1)-REF(closePrice,1))/REF(closePrice,1);
var2 = closePrice / openPrice;

5. 中文含义:

eveningstar=(2 日前的收盘价/2 日前的开盘价>1.05)并且(昨日开盘价>2 日前的收盘价)并且(昨日开盘价-昨收的绝对值/昨收<0.01) 并且 收盘价/开盘价<0.95 并且 收盘价<2 日前的收盘价

var1=昨日开盘价-昨收的绝对值/昨收

var2=收盘价/开盘价

6. 输出参数说明:

eveningstar 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算

得到的条件值矩阵。值=1 表示满足黄昏之星条件，值=0 表示不满足黄昏之

条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示前一周期的收

盘价减开盘价的绝对值与收盘价的比值矩阵。

var2 是计算结果，与输入的价格序列为同型矩阵，表示最新周期的收盘价与

开盘价的比值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [eveningstar,var1,var2] = EVENINGSTAR(a,d)
```

eveningstar =

	NaN	NaN	0	0	0	0	0	0	0
0									

var1 =

NaN	NaN	0.0005	0.0005	0.0015	0.0026
0.0016	0.0005	0.0037	0.0026		

var2 =

NaN	NaN	0.9995	1.0015	1.0026	0.9984
1.0005	0.9963	0.9974	1.0015		

3.3.13.LONGCROSS-长十字

函数解释：

1. **用途:** 计算长十字 LONGCROSS 选股模型。在涨势中出现长十字线，尤其是股价有了一段较大涨幅之后出现，预示股价见顶回落的可能性极大。在跌势中出现长十字线，尤其是股价有了一段较大跌幅之后，出现见底回升的可能性极大。

2. **函数详细结构:**

[longcross]= LONGCROSS(openPrice, highPrice, lowPrice, closePrice)

[longcross, var1]= LONGCROSS(openPrice, highPrice, lowPrice,
closePrice)

[longcross, var1, var2]= LONGCROSS(openPrice, highPrice, lowPrice,
closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
 $N \geq 1$ 。

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度，
 $N \geq 1$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
 $N \geq 1$ 。

4. **公式:**

```
longcross = ABS(closePrice-openPrice)/openPrice < 0.01 AND  
            highPrice/lowPrice>1.03;  
var1 = ABS(closePrice-openPrice)/openPrice ;  
var2 = highPrice/lowPrice;
```

5. 中文含义:

longcross= 收盘价-开盘价的绝对值/开盘价 <0.01 并且 最高价/最低价 >1.03

var1=收盘价-开盘价的绝对值/开盘价

var2=最高价/最低价

6. 输出参数说明:

longcross 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得

到的条件值矩阵。值=1 表示满足长十字条件，值=0 表示不满足长十字条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价减开盘价的绝对值与开盘价的比值矩阵。

var2 是计算结果，与输入的价格序列为同型矩阵，表示最新周期的最高价与最低价的比值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> b = [20.25 20.68 20.30 19.77 19.98 19.50 19.10 19.31 19.30 19.73];
```

```
>> c = [20.25 20.11 19.10 19.10 19.00 19.06 18.65 18.55 18.76 19.10];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];

>> [longcross,var1,var2] = LONGCROSS(a,b,c,d)

longcross =

         0         0         1         1         1         0         0         1         0         1

var1 =

         0        0.0005        0.0005        0.0015        0.0026        0.0016

        0.0005        0.0037        0.0026        0.0015

var2 =

        1.0000        1.0283        1.0628        1.0351        1.0516        1.0231

        1.0241        1.0410        1.0288        1.0330
```

3.3.14.FLATBOTTOM-平底

函数解释：

1. **用途:** 计算平底 FLATBOTTOM 选股模型。股价下降到低位后，出现了两条最低价相同的 K 线，通常是见底信号。

2. **函数详细结构:**

```
[ flatbottom, var ]= FLATBOTTOM(lowPrice )
```

3. **输入参数说明:**

lowPrice 为输入的最低价序列，为 N 列数值矩阵，N>=2。

4. **公式:**

```
flatbottom = ABS(lowPrice-REF(lowPrice,1))/lowPrice < 0.001 ;  
var = ABS(lowPrice-REF(lowPrice,1))/lowPrice;
```

5. 中文含义:

flatbottom=最低价-昨日最低价的绝对值/最低价<0.001

var = 最低价-昨日最低价的绝对值/最低价

6. 输出参数说明:

FLATBOTTOM 是计算结果矩阵，与输入价格序列为同型矩阵，表示计算得到

的条件值矩阵。值=1 表示满足平底条件，值=0 表示不满足平底条件。

var 是计算结果矩阵，与输入价格序列为同型矩阵，表示最新周期的最低价

减去上一周期最低价的绝对值比上最新周期的最低价的值矩阵。

7. 示例:

```
>>lowPrice = [2.4 3.5 3.3];
```

```
>>[flatbottom,var] = FLATBOTTOM(lowPrice)
```

flatbottom =

NaN 0 0

var =

NaN 0.3143 0.0606

3.3.15.FLATCEILING-平顶

函数解释：

1. **用途：**计算平顶 FLATCEILING 选股模型。股价上升到高位后，出现了两条最高价相同的 K 线，通常是见顶信号。

2. **函数详细结构：**

[flatceiling, var]= FLATCEILING(highPrice)

3. **输入参数说明：**

highPrice 为输入的最高价序列，为 N 列数值矩阵，N>=2。

4. **公式：**

flatceiling = ABS(highPrice-REF(highPrice,1))/highPrice < 0.001;
var = ABS(highPrice-REF(highPrice,1))/highPrice;

5. **中文含义：**

flatceiling=最高价-昨日最高价的绝对值/最高价<0.001

var=最高价-昨日最高价的绝对值/最高价

6. **输出参数说明：**

FLATCEILING 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得

到的条件值矩阵。值=1 表示满足平顶条件，值=0 表示不满足平顶条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的最高

价减去上一周期最高价的绝对值比上最新周期的最高价的值矩阵。

7. 示例:

```
>>highPrice = [2.4 3.3 3.5]
```

```
flatceiling =
```

```
NaN    0    0
```

```
var =
```

```
NaN    0.2727    0.0571
```

3.3.16.GREENCANDLE-大阴烛

函数解释：

1. **用途:** 计算大阴烛 GREENCANDLE 选股模型。烛身较长，头部短少甚至没有头部出现，价位以接近全日高价开出后一直下跌，收市价接近或其企于全日最低位，反映典型的高开低收，显示市场看淡。

2. 函数详细结构:

[greencandle, var1, var2]= GREENCANDLE(openPrice, highPrice,
lowPrice, closePrice)

3. 输入参数说明:

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
 $N \geq 1$ 。

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度，
 $N \geq 1$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
 $N \geq 1$ 。

4. 公式:

greencandle = openPrice/closePrice > 1.05 AND
highPrice/lowPrice < openPrice/closePrice + 0.018;
var1 = openPrice/closePrice;
var2 = highPrice/lowPrice;

5. 中文含义:

greencandle=开盘价/收盘价>1.05 并且

最高价/最低价<开盘价/收盘价+0.018

var1=开盘价/收盘价

var2=最高价/最低价

6. 输出参数说明:

GREENCANDLE 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算

得到的条件值矩阵。值=1 表示满足大阴烛条件，值=0 表示不满足大阴烛条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价与开盘价的比值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的最高

价与最低价的比值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7 2.6];closePrice = [2.3 3.6 3.3 2.5 3];highPrice = [3
```

```
3.9 3.7 3.1 3.2];lowPrice = [2 2.1 2 2 2.4];
```

```
>>[greencandle,var1,var2] =
```

```
GREENCANDLE(openPrice,highPrice,lowPrice,closePrice)
```

```
greencandle =
```

```
0      0      0      0      0
```

```
var1 =
```

```
1.0435    0.6944    0.6364    1.0800    0.8667
```

var2 =

1.5000 1.8571 1.8500 1.5500 1.3333

3.3.17.HAMMER-锤头

函数解释：

1. **用途：**计算锤头 HAMMER 选股模型。锤头通常是短期见底信号，后市看涨。

2. **函数详细结构：**

[hammer]= HAMMER(openPrice, highPrice, lowPrice, closePrice)

[hammer, var1]= HAMMER(openPrice, highPrice, lowPrice, closePrice)

[hammer, var1, var2]= HAMMER(openPrice, highPrice, lowPrice,
closePrice)

3. **输入参数说明：**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=5。

highPrice 为输入的最高价序列，为 N 列数值矩阵，N>=5。

lowPrice 为输入的最低价序列，为 N 列数值矩阵，N>=5。

closePrice 为输入的收盘价序列，为 N 列数值矩阵，N>=5。

4. **公式：**

$$\begin{aligned} hammer = & \text{highPrice} == \text{MAX}(\text{openPrice}, \text{closePrice}) \text{ AND} \\ & \text{highPrice} - \text{lowPrice} > 3 * (\text{highPrice} - \text{MIN}(\text{openPrice}, \text{closePrice})) \text{ AND} \\ & \text{closePrice} < \text{MA}(\text{closePrice}, 5); \\ \text{var1} = & (\text{highPrice} - \text{lowPrice}) / (\text{highPrice} - \text{MIN}(\text{openPrice}, \text{closePrice})); \\ \text{var2} = & \text{closePrice} / \text{MA}(\text{closePrice}, 5); \end{aligned}$$

5. 中文含义:

hammer=最高价=开盘价和收盘价的较大值 并且 (最高价-最低价)>3*(最高价-开盘价和收盘价的较小值) 并且 收盘价<收盘价的 5 日简单移动平均

var1=(最高价-最低价)/(最高价-开盘价和收盘价的较小值)

var2=收盘价/收盘价的 5 日简单移动平均

6. 输出参数说明:

hammer 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足锤头条件，值=0 表示不满足锤头条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的最高价减最低价的差比上最高价与收盘价，开盘价的较小者的差矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价比上收盘价的 5 日简单移动平均值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> b = [20.25 20.68 20.30 19.77 19.98 19.50 19.10 19.31 19.30 19.73];
```

```
>> c = [20.25 20.11 19.10 19.10 19.00 19.06 18.65 18.55 18.76 19.10];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [hammer,var1,var2] = HAMMER(a,b,c,d)
```

```

hammer =
      NaN    NaN    NaN    NaN    0    0    0    0    0
0

var1 =
      NaN    NaN    NaN    NaN    1.1136    1.4667    1.8000
      1.2459    2.7000    12.6000

var2 =
      NaN    NaN    NaN    NaN    0.9693    0.9823    0.9804
      0.9802    1.0052    1.0312

```

3.3.18.MORNINGCROSS-早晨十字星

函数解释：

1. **用途:** 计算早晨十字星 MORNINGCROSS 选股模型。早晨十字星通常是将要见底反弹的信号。

2. **函数详细结构:**

```
[ morningcross ]= MORNINGCROSS(openPrice, closePrice )
```

```
[ morningcross, var1 ]= MORNINGCROSS(openPrice, closePrice )
```

```
[ morningcross, var1, var2 ]= MORNINGCROSS(openPrice, closePrice )
```

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=3。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 3$ 。

4. 公式:

```
morningcross = REF(closePrice,2)/REF(openPrice,2) < 0.95 AND  
              REF(openPrice,1) < REF(closePrice,2) AND  
              ABS(REF(openPrice,1) - REF(closePrice,1)) < 0.01 AND  
              closePrice/openPrice > 1.05 AND closePrice > REF(closePrice,2);  
var1 = ABS(REF(openPrice,1) - REF(closePrice,1)) ;  
var2 = closePrice / openPrice;
```

5. 中文含义:

morningcross=2 日前的收盘价/2 日前的开盘价<0.95 并且 昨日开盘价<2
日前的收盘价 并且 昨日开盘价-昨收的绝对值<0.01 并且 收盘价/开盘
价>1.05 并且 收盘价>2 日前的收盘价

var1=昨日开盘价-昨收的绝对值

var2=收盘价/开盘价

6. 输出参数说明:

morningcross 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计
算得到的条件值矩阵。值=1 表示满足早晨十字星条件，值=0 表示不满足早

晨

十字星条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示前一周期的收盘
价减开盘价的绝对值矩阵。

var2 是计算结果，与输入的价格序列为同型矩阵，表示最新周期的收盘价与开盘价的比值矩阵。

详细例子：

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [morningcross,var1,var2] = MORNINGCROSS(a,d)
```

morningcross =

	NaN	NaN	0	0	0	0	0	0	0
0									

var1 =

NaN	NaN	0.0100	0.0100	0.0300	0.0500
0.0300	0.0100	0.0700	0.0500		

var2 =

NaN	NaN	0.9995	1.0015	1.0026	0.9984
1.0005	0.9963	0.9974	1.0015		

3.3.19.MORNINGDAWN-曙光初现

函数解释：

1. **用途:** 计算曙光初现 MORNINGDAWN 选股模型。曙光初现是由两支不同颜色的阴阳烛组成，意味着市况由淡转好，通常在一个下跌市况后出现。

2. 函数详细结构:

[morningdawn, var1, var2]= MORNINGDAWN(openPrice, closePrice)

3. 输入参数说明:

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 2$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 2$ 。

4. 公式:

morningdraw = REF(closePrice,1)/REF(openPrice,1) < 0.97 AND
closePrice/openPrice > 1.03 AND
openPrice < REF(closePrice,1) AND
closePrice > REF(closePrice,1);
var1 = REF(closePrice, 1)/REF(openPrice,1);
var2 = closePrice/openPrice;

5. 中文含义:

morningdawn=昨收/昨日开盘价<0.97 并且 收盘价/开盘价>1.03 并且

开盘价<昨收 并且 收盘价>昨收

var1=昨收/昨日开盘价

var2=收盘价/开盘价

6. 输出参数说明:

morningdawn 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计

算得到的条件值矩阵。值=1 表示满足曙光初现条件，值=0 表示不满足曙光

初现条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上一周期的收

盘价与开盘价的比值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收

盘价与开盘价的比值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7];closePrice = [2.3 3.6 3.3 2.5];
```

```
>>[morningdawn,var1,var2] = MORNINGDAWN(openPrice,closePrice)
```

morningdawn =

NaN	0	0	0
-----	---	---	---

var1 =

NaN	0.9583	1.4400	1.5714
-----	--------	--------	--------

var2 =

NaN	1.4400	1.5714	0.9259
-----	--------	--------	--------

3.3.20.MORNINGSTAR-早晨之星

函数解释：

1. **用途：** 计算早晨之星 MORNINGSTAR 选股模型。早晨之星通常是将要见底反弹的信号。

2. **函数详细结构：**

[morningstar]= MORNINGSTAR(openPrice, closePrice)

[morningstar, var1]= MORNINGSTAR(openPrice, closePrice)

[morningstar, var1, var2]= MORNINGSTAR(openPrice, closePrice)

3. **输入参数说明：**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=3。

closePrice 为输入的收盘价序列，为 N 列数值矩阵，N>=3。

4. **公式：**

morningstar = REF(closePrice,2)/REF(openPrice,2) < 0.95 AND
REF(openPrice,1) < REF(closePrice,2) AND
ABS(REF(openPrice,1)-REF(closePrice,1))/REF(closePrice,1) < 0.01 AND
closePrice/openPrice > 1.05 AND closePrice > REF(closePrice,2);
var1 = ABS(REF(openPrice,1)-REF(closePrice,1))/REF(closePrice,1);
var2 = closePrice/openPrice;

5. **中文含义：**

morningstar=(2 日前的收盘价/2 日前的开盘价<0.95)并且(昨日开盘价 < 2
日前的收盘价)并且(昨日开盘价-昨收的绝对值/昨收<0.01) 并且 收盘价/开
盘价>1.05 并且 收盘价>2 日前的收盘价
var1=昨日开盘价-昨收的绝对值的绝对值/昨收

var2=收盘价/开盘价

6. 输出参数说明:

morningstar 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足早晨之星条件，值=0 表示不满足早晨之星条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示前一周期的收盘价减开盘价的绝对值与收盘价的比值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价与开盘价的比值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [morningstar,var1,var2] = MORNINGSTAR(a,d)
```

morningstar =

	NaN	NaN	0	0	0	0	0	0	0	0
--	-----	-----	---	---	---	---	---	---	---	---

var1 =

	NaN	NaN	0.0005	0.0005	0.0015	0.0026
	0.0016	0.0005	0.0037	0.0026		

var2 =

	NaN	NaN	0.9998	0.9998	0.9998	0.9998
	0.9998	0.9998	0.9998	0.9998		

3.3.21.PREGCROSS-十字胎

函数解释：

1. **用途：**计算十字胎 PREGCROSS 选股模型。十字胎是身怀六甲的一种特殊形态，十字胎代表市场原来的趋势难以维持，但并不是说市场即刻会发生反转。十字胎也可能是市场多空力量暂时的平衡点，若市场原有的力量仍占主导，则其演变成盘整状态的可能较大。

2. **函数详细结构：**

[pregcross, var1, var2]= PREGCROSS(openPrice, closePrice)

3. **输入参数说明：**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=2。

closePrice 为输入的收盘价序列，为 N 列数值矩阵，N>=2。

4. **公式：**

```
pregcross = ABS(REF(closePrice,1)-REF(openPrice,1))/REF(closePrice,1)
> 0.04 AND ABS(closePrice-openPrice/closePrice <0.005 AND
closePrice < MAX(REF(closePrice,1),REF(openPrice,1)) AND
closePrice > MIN(REF(closePrice,1),REF(openPrice,1));
var1=ABS(REF(closePrice,1)-REF(openPrice,1))/REF(closePrice,1);
var2 = ABS(closePrice-openPrice)/closePrice;
```

5. **中文含义：**

pregcross=昨收-昨日开盘价的绝对值/昨收>0.04 并且 收盘价-开盘价的绝对值/收盘价<0.005 并且 收盘价<昨收和昨日开盘价的较大值 并且 收盘价>昨收和昨日开盘价的较小值

$\text{var1} = \text{昨收} - \text{昨日开盘价的绝对值} / \text{昨收}$

$\text{var2} = \text{收盘价} - \text{开盘价的绝对值} / \text{收盘价}$

6. 输出参数说明:

PREGCROSS 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得

到的条件值矩阵。值=1 表示满足十字胎条件，值=0 表示不满足十字胎条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上一周期的收盘

价减去开盘价的绝对值比上收盘价的值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价减去开盘价的绝对值比上收盘价的值矩阵。

7. 示例:

```
>> openPrice = [2.4 2.5 2.1]; closePrice = [2.3 3.6 3.3];
```

```
>> [pregcross,var1,var2] = PREGCROSS(openPrice,closePrice)
```

```
pregcross =
```

```
NaN    0    0
```

```
var1 =
```

NaN 0.0435 0.3056

var2 =

NaN 0.3056 0.3636

3.3.22.PREGLINE-孕育线

函数解释：

1. **用途:** 计算孕育线 PREGLINE 选股模型。经过连日飙升后，当日的开收盘价完全孕育在前一日的大阳线之中，并出现一根阴线，这也代表上涨力道不足，是下跌的前兆。若隔天再拉出一条上影阴线，更可判断为行情暴跌的征兆。

2. **函数详细结构:**

[preglime, var1, var2]= PREGLINE(openPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 3$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 3$ 。

4. **公式:**

```

pregline = REF(closePrice,2) > REF(openPrice,2) AND
(REF(closePrice,2)-REF(openPrice,2))/REF(openPrice,2) >= 0.02 AND
REF(openPrice,1) >= REF(closePrice,1) AND
(REF(openPrice,1)-REF(closePrice,1))/REF(closePrice,1) <= 0.02 AND
REF(openPrice,1) <= REF(closePrice,2) AND REF(closePrice,1) >=
REF(openPrice,2) AND C >= openPrice AND
(closePrice-openPrice)/openPrice >= 0.02
AND closePrice >= REF(openPrice,1) AND openPrice <= REF(closePrice,1)
AND (MAX(closePrice,REF(C,2))-MIN(closePrice,REF(closePrice,2)))/
MIN(closePrice,REF(closePrice,2)) <= 0.01;
var1 = (REF(openPrice,1)-REF(closePrice,1))/REF(closePrice,1);
var2 = (closePrice - openPrice) / openPrice;

```

5. 中文含义:

pregline=2 日前的收盘价>2 日前的开盘价 并且 (2 日前的收盘价-2 日前的
 开盘价)/2 日前的开盘价=0.02 并且 昨日开盘价>=昨收 并且 (昨日开盘价-
 昨收)/昨收<=0.02 并且 昨日开盘价<=2 日前的收盘价 并且 昨收>=2 日前
 的开盘价 并且 收盘价>=开盘价 并且 (收盘价-开盘价)/开盘价>=0.02 并
 且 收盘价>=昨日开盘价 并且 开盘价<=昨收 并且 (收盘价和 2 日前的收盘
 价的较大值-收盘价和 2 日前的收盘价的较小值/收盘价和 2 日前的收盘价的较
 小值<=0.01

var1=(昨日开盘价-昨收)/昨收

var2=(收盘价-开盘价)/开盘价

6. 输出参数说明:

PREGLINE 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到
 的条件值矩阵。值=1 表示满足孕育线条件，值=0 表示不满足孕育线条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上一周期的开盘

价减去收盘价的差比上收盘价的值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收

盘价减去开盘价的差比上开盘价的值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1];closePrice = [2.3 3.6 3.3];
```

```
>>[pregline,var1,var2] = PREGLINE(openPrice,closePrice)
```

```
pregline =
```

```
NaN    NaN    0
```

```
var1 =
```

```
NaN      NaN  -0.3056
```

```
var2 =
```

NaN NaN 0.5714

3.3.23.PREGNANT-身怀六甲

函数解释：

1. **用途：**计算身怀六甲 PREGNANT 选股模型。身怀六甲形态的出现，一般预示着股价上升或下跌的力量已趋衰竭，随之而来的很可能就是股价的转势。

2. **函数详细结构：**

[pregnant, var1, var2]= PREGNANT(openPrice, closePrice)

3. **输入参数说明：**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=2。

closePrice 为输入的收盘价序列，为 N 列数值矩阵，N>=2。

4. **公式：**

pregnant = ABS(REF(closePrice,1)-REF(openPrice,1))/REF(closePrice,1)
> 0.04 AND ABS(closePrice-openPrice)/closePrice < 0.005 AND
MAX(closePrice,openPrice) < MAX(REF(closePrice,1),REF(openPrice,1)) AND
MIN(closePrice,openPrice) > MIN(REF(closePrice,1),REF(openPrice,1));
var1 = ABS(REF(closePrice,1)-REF(openPrice,1))/REF(closePrice,1) ;
var2 = ABS(closePrice-openPrice)/closePrice;

5. **中文含义：**

pregnant=昨收-昨日开盘价的绝对值/昨收>0.04 并且 收盘价-开盘价的绝对值/收盘价<0.005 并且 收盘价和开盘价的较大值<昨收和昨日开盘价的较

大值 并且 收盘价和开盘价的较小值>昨收和昨日开盘价的较小值

$\text{var1} = \text{昨收} - \text{昨日开盘价的绝对值} / \text{昨收}$

$\text{var2} = \text{收盘价} - \text{开盘价的绝对值} / \text{收盘价}$

6. 输出参数说明:

PREGNANT 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到

的条件值矩阵。值=1 表示满足身怀六甲条件，值=0 表示不满足身怀六甲条

件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上一周期的收盘

价减去开盘价的绝对值比上收盘价矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价减去开盘价的绝对值比上收盘价矩阵。

7. 示例:

```
>> openPrice = [2.4 2.5 2.1]; closePrice = [2.3 3.6 3.3];
```

```
>> [pregnant,var1,var2] = PREGNANT(openPrice,closePrice)
```

`pregnant =`

NaN	0	0
-----	---	---

var1 =

NaN 0.0435 0.3056

var2 =

NaN 0.3056 0.3636

3.3.24.REDCANDLE-太阳烛

函数解释：

1. **用途:** 计算太阳烛 REDCANDLE 选股模型。烛身较长，脚部短少甚至没有脚部出现，价位以接近全日低价开出后一直上升，收市价接近或其企于全日最高位，反映典型的低开高收，显示市场向好信号。

2. **函数详细结构:**

```
[ redcandle, var1, var2 ]= REDCANDLE(openPrice, highPrice, lowPrice,  
closePrice )
```

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
N>=1。

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度，
 $N \geq 1$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度，
 $N \geq 1$ 。

4. 公式:

```
redcandle = closePrice/openPrice > 1.05 AND  
            highPrice/lowPrice < closePrice/openPrice + 0.018;  
var1 = closePrice/openPrice;  
var2 = highPrice/lowPrice;
```

5. 中文含义:

redcandle=收盘价/开盘价>1.05 并且

最高价/最低价<收盘价/开盘价+0.018

var1=收盘价/开盘价

var2=最高价/最低价

6. 输出参数说明:

REDCANDLE 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到

的条件值矩阵。值=1 表示满足大阳烛条件，值=0 表示不满足大阳烛条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价与开盘价的比值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的最高

价与最低价的比值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7 2.6];closePrice = [2.3 3.6 3.3 2.5 3];highPrice = [3
```

```
3.9 3.7 3.1 3.2];lowPrice = [2 2.1 2 2 2.4];
```

```
>>[redcandle,var1,var2] =
```

```
REDCANDLE(openPrice,highPrice,lowPrice,closePrice)
```

redcandle =

0	0	0	0	0
---	---	---	---	---

var1 =

1.0435	0.6944	0.6364	1.0800	0.8667
--------	--------	--------	--------	--------

var2 =

1.5000	1.8571	1.8500	1.5500	1.3333
--------	--------	--------	--------	--------

3.3.25.SEPARATE-分离

函数解释：

1. **用途:** 计算分离 SEPARATE 选股模型。一阴一阳两根实体长度大致相同的 K 线，通常的行情反转的信号。

2. **函数详细结构:**

[separate, var1, var2]= SEPARATE(openPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 2$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 2$ 。

4. **公式:**

separate = $\text{ABS}(\text{openPrice} - \text{REF}(\text{openPrice}, 1)) < 0.005 \text{ AND}$
 $(\text{closePrice} - \text{openPrice}) * (\text{REF}(\text{closePrice}, 1) - \text{REF}(\text{openPrice}, 1)) < 0;$
var1 = $\text{ABS}(\text{openPrice} - \text{REF}(\text{openPrice}, 1));$
var2 = $(\text{closePrice} - \text{openPrice}) * (\text{REF}(\text{closePrice}, 1) - \text{REF}(\text{openPrice}, 1));$

5. **中文含义:**

separate=开盘价-昨日开盘价的绝对值<0.005 并且 (收盘价-开盘价)*(昨收-昨日开盘价)<0

var1=开盘价-昨日开盘价的绝对值

var2=(收盘价-开盘价)*(昨收-昨日开盘价)

6. **输出参数说明:**

SEPARATE 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的

条件值矩阵。值=1 表示满足分离条件，值=0 表示不满足分离条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示当前周期开盘价

减去上一周期开盘价的绝对值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示当前周期收盘价

与开盘价的差乘以上一周收盘价与开盘价的差的值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7];closePrice = [2.3 3.6 3.3 2.5];
```

```
>>[separate,var1,var2] = SEPARATE(openPrice,closePrice)
```

separate =

NaN	0	0	0
-----	---	---	---

var1 =

NaN	0.1000	0.4000	0.6000
-----	--------	--------	--------

var2 =

NaN -0.1100 1.3200 -0.2400

3.3.26.SHOTSTAR-射击之星

函数解释：

1. **用途:** 计算射击之星 SHOTSTAR 选股模型。射击之星表示股价已经失去了上升的持久力，多方已抵抗不住空方打击，股价随时可能见顶回落，因此在此在股价大幅上扬后，见到射击之星应退出观望为宜。

2. **函数详细结构:**

[shotstar]= SHOTSTAR(openPrice, highPrice, lowPrice, closePrice)

[shotstar, var1]= SHOTSTAR(openPrice, highPrice, lowPrice, closePrice)

[shotstar, var1, var2]= SHOTSTAR(openPrice, highPrice, lowPrice,
closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

highPrice 为输入的最高价序列，为 N 列数值矩阵， $N \geq 5$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵， $N \geq 5$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

4. **公式:**

```

shotstar = MIN(openPrice,closePrice) == lowPrice AND
            highPrice -LOW > 3*(MAX(openPrice,closePrice)-lowPrice ) AND
            closePrice > MA(closePrice,5);
var1 = (highPrice-lowPrice)/(MAX(openPrice, closePrice) - lowPrice );
var2 = closePrice/MA(closePrice, 5);

```

5. 中文含义:

shotstar=开盘价和收盘价的较小值=最低价 并且 最高价-最低价>3*(开盘价和收盘价的较大值-最低价) 并且 收盘价>收盘价的 5 日简单移动平均

var1=(最高价-最低价)/(开盘价和收盘价的较大值-最低价)

var2=收盘价/收盘价的 5 日简单移动平均

6. 输出参数说明:

shotstar 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到

的条件值矩阵。值=1 表示满足射击之星条件，值=0 表示不满足射击之星条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的最高价减最低价的差比上收盘价，开盘价的较大者与最低价的差值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘价比上收盘价的 5 日简单移动平均值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> b = [20.25 20.68 20.30 19.77 19.98 19.50 19.10 19.31 19.30 19.73];

>> c = [20.25 20.11 19.10 19.10 19.00 19.06 18.65 18.55 18.76 19.10];

>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];

>> [shotstar,var1,var2] = SHOTSTAR(a,b,c,d)
```

```
shotstar =
```

```
NaN NaN NaN NaN 0 0 0 0 0
```

```
0
```

```
var1 =
```

```
NaN NaN NaN NaN 6.5333 2.5882 2.1429
```

```
3.4545 1.3846 1.0328
```

```
var2 =
```

```
NaN NaN NaN NaN 0.9693 0.9823 0.9804
```

```
0.9802 1.0052 1.0312
```

3.3.27.SPRUNGROOF-吊顶

函数解释：

1. **用途：** 计算吊顶 SPRUNGROOF 选股模型。吊顶通常是见顶信号 ,后市看跌。

2. **函数详细结构:**

```
[ sprungroof ]= SPRUNGROOF(openPrice, highPrice, lowPrice, closePrice )
```

```
[ sprungroof, var1 ]= SPRUNGROOF(openPrice, highPrice, lowPrice,
```

closePrice)

[sprunproof, var1, var2]= SPRUNGROOF(openPrice, highPrice, lowPrice,
closePrice)

3. 输入参数说明:

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

highPrice 为输入的最高价序列，为 N 列数值矩阵， $N \geq 5$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵， $N \geq 5$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

4. 公式:

$$\begin{aligned} \text{sprunproof} &= \text{highPrice} == \text{MAX}(\text{openPrice}, \text{closePrice}) \text{ AND} \\ &\quad \text{highPrice} - \text{lowPrice} > 3 * (\text{highPrice} - \text{MIN}(\text{openPrice}, \text{closePrice})) \text{ AND} \\ &\quad \text{closePrice} > \text{MA}(\text{closePrice}, 5); \\ \text{var1} &= (\text{highPrice} - \text{lowPrice}) / (\text{highPrice} - \text{MIN}(\text{openPrice}, \text{closePrice})); \\ \text{var2} &= \text{closePrice} / \text{MA}(\text{closePrice}, 5); \end{aligned}$$

5. 中文含义:

sprunproof= 最高价=开盘价和收盘价的较大值 并且 （最高价-最低价）>3*(最高价-开盘价和收盘价的较小值) 并且 收盘价>收盘价的 5 日简单移动平均

var1=(最高价-最低价)/(最高价-开盘价和收盘价的较小值)

var2=收盘价/收盘价的 5 日简单移动平均

6. 输出参数说明:

sprunproof 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算

得到的条件值矩阵。值=1 表示满足吊顶条件，值=0 表示不满足吊顶条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的最

高价减最低价的差比上收盘价，开盘价的较大者与最低价的差矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收

盘价比上收盘价的 5 日简单移动平均值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> b = [20.25 20.68 20.30 19.77 19.98 19.50 19.10 19.31 19.30 19.73];
```

```
>> c = [20.25 20.11 19.10 19.10 19.00 19.06 18.65 18.55 18.76 19.10];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [sprungroof,var1,var2] = SPRUNGROOF(a,b,c,d)
```

sprungroof =

NaN	NaN	NaN	NaN	0	0	0	0	0
-----	-----	-----	-----	---	---	---	---	---

0

var1 =

NaN	NaN	NaN	NaN	1.1136	1.4667	1.8000
1.2459	2.7000	12.6000				

var2 =

NaN	NaN	NaN	NaN	0.9693	0.9823	0.9804
-----	-----	-----	-----	--------	--------	--------

0.9802 1.0052 1.0312

3.3.28.THREECROWS-三只乌鸦

函数解释：

1. **用途：**计算三只乌鸦 THREECROWS 选股模型。三只乌鸦是由三根短小的连续下降的阴 K 线组成，K 线收盘价一日比一日低，后市将继续下跌。

2. **函数详细结构：**

[threecrows, var1, var2]= THREECROWS(openPrice, closePrice)

3. **输入参数说明：**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=3。

closePrice 为输入的收盘价序列，为 N 列数值矩阵，N>=3。

4. **公式：**

threecrows = REF(closePrice,2) < REF(openPrice,2) AND
REF(closePrice,1) < REF(openPrice,1) AND
closePrice < openPrice AND
REF(closePrice,1) < REF(closePrice,2) AND
closePrice < REF(closePrice,1);

var1 = REF(closePrice,1)/REF(openPrice,1);

var 2 = closePrice / openPrice;

5. **中文含义：**

threecrows=2 日前的收盘价<2 日前的开盘价 并且 昨收<昨日开盘价 并且

收阴线 并且 昨收<2 日前的收盘价 并且 收盘价<昨收

var1=昨收/昨日开盘价

var2=收盘价/开盘价

6. 输出参数说明:

THREECROWS 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算

得到的条件值矩阵。值=1 表示满足三只乌鸦条件，值=0 表示不满足三只乌

鸦

条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上一周期的收盘

价与开盘价的比值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价与开盘价的比值矩阵。

7. 示例:

```
>> openPrice = [2.4 2.5 2.1 2.7]; closePrice = [2.3 3.6 3.3 2.5];
```

```
>> [threecrows,var1,var2] = THREECROWS(openPrice,closePrice)
```

```
threecrows =
```

```
NaN    NaN    0    0
```

```
var1 =
```

NaN	NaN	1.4400	1.5714
-----	-----	--------	--------

var2 =

NaN	NaN	1.5714	0.9259
-----	-----	--------	--------

3.3.29.THREEKNIGHTS-三个白武士

函数解释：

1. **用途:** 计算三个白武士 THREEKNIGHTS 选股模型。三个白武士是由三根短小的连续上升的阳 K 线组成，K 线收盘价一日比一日高，表示武士勇敢前进，基础扎实，后势涨幅将加大。

2. **函数详细结构:**

[threeknights, var1, var2]= THREEKNIGHTS(openPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵，N>=2。

closePrice 为输入的收盘价序列，为 N 列数值矩阵，N>=2。

4. **公式:**

threeknights = REF(closePrice,2) > REF(openPrice,2) AND
 REF(closePrice,1) > REF(openPrice,1) AND
 closePrice > openPrice AND
 REF(closePrice,1) > REF(closePrice,2) AND
 closePrice > REF(closePrice,1);
 var1 = REF(closePrice,1)/REF(openPrice,1) ;
 var 2 = closePrice / openPrice;

5. 中文含义:

threeknights=2 日前的收盘价>2 日前的开盘价 并且 昨收>昨日开盘价 并

且 收阳线 并且 昨收>2 日前的收盘价 并且 收盘价>昨收

var1=昨收/昨日开盘价

var2=收盘价/开盘价

6. 输出参数说明:

THREEKNIGHTS 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算

得到的条件值矩阵。值=1 表示满足三个白武士条件，值=0 表示不满足三个白

武士条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示上一周期的收盘

价与开盘价的比值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价与开盘价的比值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7];closePrice = [2.3 3.6 3.3 2.5];
```

```
>>[threeknights,var1,var2] = THREEKNIGHTS(openPrice,closePrice)
```

```
threeknights =
```

NaN NaN 0 0

var1 =

NaN NaN 1.4400 1.5714

var2 =

NaN NaN 1.5714 0.9259

3.3.30.TURNHAMMER-倒转锤头

函数解释：

1. **用途:** 计算倒转锤头 TURNHAMMER 选股模型。倒转锤头通常是短期见顶的信号。

2. **函数详细结构:**

[turnhammer]= TURNHAMMER(openPrice, highPrice, lowPrice,
closePrice)

[turnhammer, var1]= TURNHAMMER(openPrice, highPrice, lowPrice,
closePrice)

[turnhammer, var1, var2]= TURNHAMMER(openPrice, highPrice, lowPrice,

closePrice)

3. 输入参数说明:

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

highPrice 为输入的最高价序列，为 N 列数值矩阵， $N \geq 5$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵， $N \geq 5$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

4. 公式:

$$\begin{aligned} turnhammer = & \text{MIN}(\text{openPrice}, \text{closePrice}) == \text{lowPrice} \text{ AND} \\ & \text{highPrice} - \text{lowPrice} > 3 * (\text{MAX}(\text{openPrice}, \text{closePrice}) - \text{lowPrice}) \text{ AND} \\ & \text{closePrice} < \text{MA}(\text{closePrice}, 5); \\ \text{var1} = & (\text{highPrice} - \text{lowPrice}) / (\text{MAX}(\text{openPrice}, \text{closePrice}) - \text{lowPrice}); \\ \text{var2} = & \text{closePrice} / \text{MA}(\text{closePrice}, 5); \end{aligned}$$

5. 中文含义:

turnhammer= 开盘价和收盘价的较小值 = 最低价 并且 最高价 - 最低价 > 3 * (开盘价和收盘价的较大值 - 最低价) 并且 收盘价 < 收盘价的 5 日简单移动平均

var1=(最高价-最低价)/(开盘价和收盘价的较大值-最低价)

var2=收盘价/收盘价的 5 日简单移动平均

6. 输出参数说明:

turnhammer 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足倒转锤头条件，值=0 表示不满足倒转锤头条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的最高

价减最低价的差比上收盘价，开盘价的较大者与最低价的差值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价比上收盘价的 5 日简单移动平均值矩阵。

7. 示例:

```
>> a = [20.25 20.41 19.51 19.45 19.10 19.23 18.85 18.77 19.15 19.68];
```

```
>> b = [20.25 20.68 20.30 19.77 19.98 19.50 19.10 19.31 19.30 19.73];
```

```
>> c = [20.25 20.11 19.10 19.10 19.00 19.06 18.65 18.55 18.76 19.10];
```

```
>> d = [20.25 20.40 19.50 19.48 19.15 19.20 18.86 18.70 19.10 19.71];
```

```
>> [turnhammer,var1,var2] = TURNHAMMER(a,b,c,d)
```

```
turnhammer =
```

```
NaN NaN NaN NaN 0 0 0 0 0
```

```
0
```

```
var1 =
```

```
NaN NaN NaN NaN 6.5333 2.5882 2.1429
```

```
3.4545 1.3846 1.0328
```

```
var2 =
```

```
NaN NaN NaN NaN 0.9693 0.9823 0.9804
```

```
0.9802 1.0052 1.0312
```

3.3.31.UPSHADOW-长上影

函数解释：

1. **用途:** 计算长上影 UPSHADOW 选股模型。长上影线是上方抛压重，股价上行压力大的反映，通常是见顶信号。

2. **函数详细结构:**

[upshadow, var]= UPSHADOW(openPrice, highPrice, lowPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

highPrice 为输入的最高价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵。N 为提取的时间长度， $N \geq 1$ 。

4. **公式:**

$$\text{upshadow} = (\text{highPrice} - \text{MAX}(\text{closePrice}, \text{openPrice})) / (\text{highPrice} - \text{lowPrice}) > 0.667$$
$$\text{var} = (\text{highPrice} - \text{MAX}(\text{closePrice}, \text{openPrice})) / (\text{highPrice} - \text{lowPrice});$$

5. **中文含义:**

$$\text{upshadow} = (\text{最高价} - \text{收盘价和开盘价的较大值}) / (\text{最高价} - \text{最低价}) > 0.667$$

$var = (\text{最高价} - \text{收盘价和开盘价的较大值}) / (\text{最高价} - \text{最低价})$

6. 输出参数说明:

UPSHADOW 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的条件值矩阵。值=1 表示满足长上影条件，值=0 表示不满足长上影条件。

var 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的最高价减去收盘价与开盘价的较大者的差比上最高价与最低价的差值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7];closePrice = [2.3 3.6 3.3 2.5];highPrice = [3 3.9  
3.7 3.1];lowPrice = [2 2.1 2 2];  
>>[upshadow,var] = UPSHADOW(openPrice,highPrice,lowPrice,closePrice)
```

upshadow =

0 0 0 0

var =

0.6000 0.1667 0.2353 0.3636

3.3.32.UPSTEPS-上升三部曲

函数解释：

1. **用途:** 计算上升三部曲 UPSTEPS 选股模型。一般情况下，在上升三部曲出现之后，股价都会形成一轮较大的升势。

2. **函数详细结构:**

[upsteps, var1, var2]= UPSTEPS(openPrice, highPrice, lowPrice, closePrice)

3. **输入参数说明:**

openPrice 为输入的开盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

highPrice 为输入的最高价序列，为 N 列数值矩阵， $N \geq 5$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵， $N \geq 5$ 。

closePrice 为输入的收盘价序列，为 N 列数值矩阵， $N \geq 5$ 。

4. **公式:**

upsteps = REF(closePrice,4)/REF(openPrice,4) > 1.03 AND
REF(closePrice,3) < REF(openPrice,3) AND
REF(closePrice,2) < REF(openPrice,2) AND
REF(closePrice,1) < REF(openPrice,1) AND
REF(lowPrice,4) < REF(lowPrice,3) AND
REF(lowPrice,4) < REF(lowPrice,2) AND
REF(lowPrice,4) < REF(lowPrice,1) AND
REF(highPrice,4) > REF(highPrice,3) AND
REF(highPrice,4) > REF(highPrice,2) AND
REF(highPrice,4) > REF(highPrice,1) AND
closePrice/openPrice > 1.03 AND
closePrice > REF(closePrice,4);
var1 = closePrice/REF(closePrice, 4);
var2 = closePrice/openPrice;

5. **中文含义:**

upsteps=4 日前的收盘价/4 日前的开盘价>1.03 并且

3 日前的收盘价<3 日前的开盘价 并且

2 日前的收盘价<2 日前的开盘价 并且

昨日收盘价<昨日开盘价 并且

4 日前的最低价<3 日前的最低价 并且

4 日前的最低价<2 日前的最低价 并且

4 日前的最低价<昨日最低价 并且

4 日前的最高价>3 日前的最高价 并且

4 日前的最高价>2 日前的最高价 并且

4 日前的最高价>昨日最高价 并且

收盘价/开盘价>1.03 并且

收盘价>4 日前的收盘价

var1=收盘价/4 日前的收盘价

var2=收盘价/开盘价

6. 输出参数说明:

UPSTEPS 是计算结果矩阵，与输入的价格序列为同型矩阵，表示计算得到的

条件值矩阵。值=1 表示满足上升三部曲条件，值=0 表示不满足上升三部曲条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价比上 4 个周期前的收盘价的值矩阵。

var2 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期的收盘

价与开盘价的比值矩阵。

7. 示例:

```
>>openPrice = [2.4 2.5 2.1 2.7 2.6];closePrice = [2.3 3.6 3.3 2.5 3];highPrice = [3
```

```
3.9 3.7 3.1 3.2];lowPrice = [2 2.1 2 2 2.4];
```

```
>>[upsteps,var1,var2] = UPSTEPS(openPrice,highPrice,lowPrice,closePrice)
```

upsteps =

NaN	NaN	NaN	NaN	0
-----	-----	-----	-----	---

var1 =

NaN	NaN	NaN	NaN	1.3043
-----	-----	-----	-----	--------

var2 =

NaN	NaN	NaN	NaN	1.1538
-----	-----	-----	-----	--------

3.3.33.VACANT-跳空缺口

函数解释：

1. **用途:** 计算跳空缺口 VACANT 选股模型。跳空高开或跳空多开，将形成日后的支撑位或阻力位。

2. **函数详细结构:**

[vacant, var1, var2]= VACANT(highPrice, lowPrice)

3. **输入参数说明:**

highPrice 为输入的最高价序列，为 N 列数值矩阵， $N \geq 2$ 。

lowPrice 为输入的最低价序列，为 N 列数值矩阵， $N \geq 2$ 。

4. **公式:**

$$\text{vacant} = \text{highPrice} < \text{REF}(\text{lowPrice}, 1) \text{ OR } \text{lowPrice} > \text{REF}(\text{highPrice}, 1);$$

$$\text{var1} = \text{highPrice} / \text{REF}(\text{lowPrice}, 1);$$

$$\text{var2} = \text{lowPrice} / \text{REF}(\text{highPrice}, 1);$$

5. **中文含义:**

$$\text{vacant} = \text{最高价} < \text{昨日最低价} \text{ 或者 } \text{最低价} > \text{昨日最高价}$$

$$\text{var1} = \text{最高价} / \text{昨日最低价}$$

$$\text{var2} = \text{最低价} / \text{昨日最高价}$$

6. **输出参数说明:**

VACANT 是计算结果矩阵，与输入的价格序列为同型矩阵，表计算得到的条

件值矩阵。值=1 表示满足跳空缺口条件，值=0 表示不满足跳空缺口条件。

var1 是计算结果矩阵，与输入的价格序列为同型矩阵，表示最新周期最高价

与前一周期最低价的比值矩阵。

var2 是计算结果，与输入的价格序列为同型矩阵，表示最新周期最低价与前

一周期最高价的比值矩阵。

7. 示例:

```
>>lowPrice = [2.4 3.5 3.3];highPrice = [2.7 3.8 3.9];
```

```
>>[vacant,var1,var2] = VACANT(highPrice,lowPrice)
```

vacant =

NaN	1	0
-----	---	---

var1 =

NaN	1.5833	1.1143
-----	--------	--------

var2 =

NaN	1.2963	0.8684
-----	--------	--------