

1. 因子定义

1.1 因子定义

因子定义有两种方式：

- 定义因子函数，使用金融算子和公共因子：

```
def MA(close):  
    return Mean(close, 5) / close;
```

- 完全自定义，入口函数名定义为factor_calc：

```
def get_kline_fields(task_info):  
    pass  
  
### 系统入口函数  
def factor_calc(task_info):  
    get_kline_fields(task_info)
```

1.1.2 金融算子介绍

可以使用银河提供的金融算子，快捷进行因子构建

```
def swMean(close):  
    return TS_MEAN(close, 5) / close;
```

上述代码中TS_MEAN为金融算子，表示滚动n日均值。

银河提供的金融算子科分为简单算子、截面算子、时序算子、时序切割算子和技术指标类算子。

1.1.2.1 元素级运算算子

- ADD_CONST(x, c)
 - 参数：
x: 标量/向量/矩阵，c: 标量。对x内的元素逐个求和并返回。
- MUL_CONST(x, c)
 - 参数：
x: 标量/向量/矩阵，c: 标量。对x内的元素逐个乘以c并返回。
- ADD(x, y)
 - 参数：
x: 标量/向量/矩阵，y: 标量或者和x长度相同的向量或矩阵。对x和y内的元素求和并返回。
- SUB(x, y)
 - 参数：
x: 标量/向量/矩阵，y: 标量或者和x长度相同的向量或矩阵。逐元素地返回x和y的差。
- MUL(x, y)
 - 参数：
x: 标量/向量/矩阵，y: 标量或者和x长度相同的向量或矩阵。逐元素地返回x和y的积。

- DIV(x, y)
 - 参数:
 - x: 标量/向量/矩阵, y: 标量或者和x长度相同的向量或矩阵。返回x和y中元素依次相除的结果。
- LOG(x)
 - 参数:
 - x: 标量/向量/数据对/矩阵。求x的自然对数（以常数e为底）。
- EXP(x)
 - 参数:
 - x: 标量/向量/数据对/矩阵。返回e的x次方。e 是一个常数, 为2.71828。
- SQRT(x)
 - 参数:
 - x: 标量/向量/矩阵。计算x中每个元素的平方根。
- SQUARE(x)
 - 参数:
 - x: 标量/向量/数据对/矩阵。返回x的平方。
- SIN(x)
 - 参数:
 - x: 标量/向量/矩阵。返回x的正弦。
- COS(x)
 - 参数:
 - x: 标量/向量/矩阵。返回x的余弦。
- NEG(x)
 - 参数:
 - x: 标量/向量/数据对/矩阵。返回x的相反数。
- RECIPROCAL(x)
 - 参数:
 - x: 标量/向量/矩阵。返回x的倒数。
- SIGN(x)
 - 参数:
 - x: 标量/向量/矩阵。返回x的正负号, 正数返回1, 0返回0, 负数返回-1。
- ABS(x)
 - 参数:
 - x: 标量/向量/数据对/矩阵。返回x的绝对值。
- SIGMOID(x)
 - 参数:
 - x: 标量/向量/矩阵。对应sigmoid激活函数, 返回 $1/(1+\exp(-1*x))$ 。
- HARDSIGMOID(x)
 - 参数:
 - x: 标量/向量/矩阵。对应hardsigmoid激活函数, 返回 $\min(\max((x+3)/6, 0), 1)$ 。
- LEAKYRELU(x, a)
 - 参数:
 - x: 标量/向量/矩阵, a:标量/向量/矩阵, 对应leakyrelu激活函数, 返回 $\max(x, 0) + a*\min(x, 0)$ 。

- GELU(x)
 - 参数:
 - x: 标量/向量。对应gelu激活函数, 返回 $x \cdot \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \right)$ (均值为0, 标准差为1的正态分布的累计密度函数的值)。

1.1.2.2 截面运算算子

- XS_CUTQUARTILE(x, a)
 - 参数:
 - x: 向量, a: 四分位距倍数: 1.5, 2.0, 2.5, 3.0。采用四分位数法拉回截面上的离群值。
- XS_CUTZSCORE(x, a):
 - 参数:
 - x: 向量, a: 四分位距倍数: 1.5, 2.0, 2.5, 3.0。采用zscore法拉回截面上的离群值。
- RANK_PCT(x, ascending=True):
 - 参数:
 - x: 标量/向量。截面上分位数。
- XS_REGRES(x, y, intercept=True)
 - 参数:
 - x: 矩阵/元组, y: 向量。截面上y对x开展一元线性回归的残差。
- xs_sortreverse(x, n, mode)
 - 参数:
 - x: 向量, n: 标量, mode: 0-前端, 1-后端, 2-两端。x截面前/后n名对应的x乘以-1。
- xs_zscorereverse(x, n, mode):
 - 参数:
 - x: 向量, n: 标量, mode: 0-前端, 1-后端, 2-两端。x截面z值大于/小于阈值对应的x乘以-1。
- xs_grouping_sortreverse(x, y, n, mode)
 - 参数:
 - x: 向量, y: 向量, n: 标量, mode: 0-前端, 1-后端, 2-两端。y截面前/后n名对应的x乘以-1。
- xs_grouping_zscorereverse(x, y, a, mode)
 - 参数:
 - x: 向量, y: 向量, a: 1.5, 2.0, 2.5, 3.0, mode: 0-前端, 1-后端, 2-两端。y截面z值大于/小于阈值对应的x乘以-1。

1.1.2.3 时序运算算子

- TS_CUTQUARTILE(x, iqr_scale)
 - 参数:
 - x: 向量, iqr_scale: 1.5, 2.0, 2.5, 3.0。采用四分位数法拉回时序扩展窗口上的离群值。
- TS_CUTZSCORE(x, std_scale)
 - 参数:
 - x: 向量, std_scale: 1.5, 2.0, 2.5, 3.0。采用zscore法拉回时序扩展窗口上的离群值。
- TS_DELAY(x, n)
 - 参数:
 - x: 向量/矩阵/表, n: 可选参数, 一个整数, 用于减数相较于被减数的索引偏移。默认是1。时间序列延后n期。

- 对于 X 中的每一个元素，计算 $X_i - X_{i-n}$ ，NULL 值不参与计算。
 若 X 是向量，返回一个包含 X 中两个元素之差的向量。
 若 X 是矩阵，在每列内进行上述计算，返回一个与 X 维度相同的矩阵。
 若 X 是表，在每列内进行上述计算，返回一个与 X 行数与列数都相同的表。
- TS_DELTAS(x, n)
 - 参数：
 - x: 向量/矩阵/表, n: 可选参数，一个整数，用于减数相较于被减数的索引偏移。默认是1。时间序列的n期差分。
 - 对于 X 中的每一个元素，计算 $X_i - X_{i-n}$ ，NULL 值不参与计算。
 若 X 是向量，返回一个包含 X 中两个元素之差的向量。
 若 X 是矩阵，在每列内进行上述计算，返回一个与 X 维度相同的矩阵。
 若 X 是表，在每列内进行上述计算，返回一个与 X 行数与列数都相同的表。
- TS_PCTCHANGE(x, n)
 - 参数：
 - x: 向量/矩阵/表, n: 可选参数，整型，用于指定计算 X 中两个元素值变化百分比时的元素间隔数，默认值为 1。
 计算两个元素之间的值变化比例。即对于 X 中的每一个元素，计算 $(X_i / X_{i-n}) - 1$ 。
- TS_ROLLRANK(x, d, ascending=True)
 - 参数：
 - x: 向量/矩阵/表, d: 表示滑动窗口的长度，大于等于 2 的正整型，ascending: 布尔值，表示是否按升序排序。默认值是 true。
 滚动d日的分位数。
- TS_ROLLZSCORE(x, d)
 - 参数：
 - x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
 滚动d日的Z值。
- TS_ROLLCORR(x, y, window)
 - 参数：
 - x: 向量/矩阵/表, y: 向量/矩阵/表, window: 表示滑动窗口的长度。大于等于 2 的正整型。
 x和y 滚动 d 日的相关系数。
- TS_RANKCORR(x, y, window)
 - 参数：
 - x: 向量/矩阵/表, y: 向量/矩阵/表, window: 表示滑动窗口的长度。大于等于 2 的正整型。
 x和y滚动d日的秩(Spearman)相关系数
- TS_COVARIANCE(x, y, window)
 - 参数：
 - x: 向量/矩阵/表, y: 向量/矩阵/表, window: 表示滑动窗口的长度。大于等于 2 的正整型。
 x和y滚动d日的协方差。
- TS_MAX(x, d)
 - 参数：
 - x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
 滚动d日最大值。
- TS_MIN(x, d)

- 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日最小值。
- TS_SUM(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日之和。
- TS_MEAN(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日均值
- TS_MEDIAN(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日中位数。
- TS_VAR(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日方差。
- TS_STDDEV(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日标准差。
- TS_AVGDEV(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日平均偏差。
- TS_KURT(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日峰度。
- TS_SKEW(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日偏度。
- TS_ARGMAX(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日最大值的索引除以d。
- TS_ARGMIN(x, d)
 - 参数:
x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
滚动d日最大值的索引除以d。
- TS_DECAY_LINEAR_MEAN(x, d)

- 参数:
 - x: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
 - 滚动d日按时间索引作为权重求均值。
- TS_ROOLWEIGHTED_MEAN(x, y, d)
 - 参数:
 - x: 向量/矩阵/表, y: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
 - 滚动d日中, 以归一化的y为权重, 求x均值。
- TS_REGBETA(x, y, d)
 - 参数:
 - x: 向量/矩阵/表, y: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
 - 滚动d日中, y对x开展一元线性回归的斜率。
- TS_REGALPHA(x, y, d)
 - 参数:
 - x: 向量/矩阵/表, y: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
 - 滚动d日中, y对x开展一元线性回归的截距。
- TS_REGRES(x, y, d)
 - 参数:
 - x: 向量/矩阵/表, y: 向量/矩阵/表, d: 表示滑动窗口的长度。大于等于 2 的正整型。
 - 滚动d日中, y对x开展一元线性回归的最近一期残差。

1.1.2.4 时序切割算子

- TS_ASCSORTCUT(seqa, window, n, mode)
 - 参数:
 - seqa: 向量/矩阵/表, window: 表示滑动窗口的长度。10, 15...,60, n: 1,2,...,10,
 - mode: 1-切割部分求seqa之和, 2-全序列seqa之和减去切割部分的seqa之和, 3-切割部分seqa乘以-1后再求全序列seqa之和。
 - 滚动window日中, 将seqa最小的n天切割出来。
- TS_DECSORTCUT(seqa, window, n, mode)
 - 参数:
 - seqa: 向量/矩阵/表, window: 表示滑动窗口的长度。10, 15...,60, n: 1,2,...,10,
 - mode: 1-切割部分求seqa之和, 2-全序列seqa之和减去切割部分的seqa之和, 3-切割部分seqa乘以-1后再求全序列seqa之和。
 - 滚动window日中, 将seqa最大的n天切割出来。
- TS_ASCZSORTCUT(seqa, window, a, mode)
 - 参数:
 - seqa: 向量/矩阵/表, window: 表示滑动窗口的长度。10, 15...,60, a: 1.5,2.0,2.5,3.0,
 - mode: 1-切割部分求seqa之和, 2-全序列seqa之和减去切割部分的seqa之和, 3-切割部分seqa乘以-1后再求全序列seqa之和。
 - 滚动window日中, 将seqa的z值小于-a的时间切割出来; 若不存在, 则切出seqa最小的1天。
- TS_DECZSORTCUT(seqa, window, a, mode)
 - 参数:
 - seqa: 向量/矩阵/表, window: 表示滑动窗口的长度。10, 15...,60, a: 1.5,2.0,2.5,3.0,
 - mode: 1-切割部分求seqa之和, 2-全序列seqa之和减去切割部分的seqa之和, 3-切割部分seqa乘以-1后再求全序列seqa之和。
 - 滚动window日中, 将seqa的z值大于a的时间切割出来; 若不存在, 则切出seqa最大的1天。

- TS_GROUPING_ASCSORTCUT(seqa, seqb, window, n, mode)
 - 参数:

seqa: 向量/矩阵/表, seqb: 向量/矩阵/表, window: 表示滑动窗口的长度。10, 15...,60, n: 1,2,...,10,

mode: 1-切割部分求seqa之和, 2-全序列seqa之和减去切割部分的seqa之和, 3-切割部分seqa乘以-1后再求全序列seqa之和。

滚动window日中, 将seqb最小的n天切割出来。
- TS_GROUPING_DECSORTCUT(seqa, seqb, window, n, mode)
 - 参数:

seqa: 向量/矩阵/表, seqb: 向量/矩阵/表, window: 表示滑动窗口的长度。10, 15...,60, n: 1,2,...,10,

mode: 1-切割部分求seqa之和, 2-全序列seqa之和减去切割部分的seqa之和, 3-切割部分seqa乘以-1后再求全序列seqa之和。

滚动window日中, 将seqb最大的n天切割出来。
- TS_GROUPING_ASCZSORTCUT(seqa, seqb, window, a, mode)
 - 参数:

seqa: 向量/矩阵/表, seqb: 向量/矩阵/表, window: 表示滑动窗口的长度。10, 15...,60, a: 1.5,2.0,2.5,3.0,

mode: 1-切割部分求seqa之和, 2-全序列seqa之和减去切割部分的seqa之和, 3-切割部分seqa乘以-1后再求全序列seqa之和。

滚动window日中, 将seqb的z值小于-a的时间切割出来; 若不存在, 则切割出seqb最小的1天。
- TS_GROUPING_DECZSORTCUT(seqa, seqb, window, a, mode)
 - 参数:

seqa: 向量/矩阵/表, seqb: 向量/矩阵/表, window: 表示滑动窗口的长度。10, 15...,60, a: 1.5,2.0,2.5,3.0,

mode: 1-切割部分求seqa之和, 2-全序列seqa之和减去切割部分的seqa之和, 3-切割部分seqa乘以-1后再求全序列seqa之和。

滚动window日中, 将seqb的z值大于a的时间切割出来; 若不存在, 则切割出seqb最大的1天

1.1.2.5 技术指标类算子

- MA(x, timeperiod=30, matype=0)
 - 参数:

x: 向量/矩阵/表, timeperiod: 长度, 默认30, matype: 0-sma,1-ema,2-wma,3-dema,4-tema,5-trima,6-kama,7-(mama),8-t3, 默认为0。

移动平均。
- EMA(x, timeperiod=30, warmup=False)
 - 参数:

x: 向量/矩阵/表, timeperiod: 长度, 默认30, warmup: 默认为false, 即计算结果的前timeperiod-1个元素为空值。若为true, 则结果的前timeperiod-1元素将由计算得出。

指数移动平均。
- WMA(x, timeperiod=30):
 - 参数:

x: 向量/矩阵/表, timeperiod: 长度, 默认30。

加权移动平均。
- SMA(x, timeperiod=30)

- 参数:
 - x: 向量/矩阵/表, timeperiod: 长度, 默认30。
 - 简单移动平均。
- DEMA(x, timeperiod=30)
 - 参数:
 - x: 向量/矩阵/表, timeperiod: 长度, 默认30。
 - 双重指数移动平均。
- KAMA(x, timeperiod=30)
 - 参数:
 - x: 向量/矩阵/表, timeperiod: 长度, 默认30。
 - 考夫曼自适应移动平均。
- MMAX(x, timeperiod=30)
 - 参数:
 - x: 向量/矩阵/表, timeperiod: 长度, 默认30。
 - 在给定长度（以元素个数或时间长度衡量）的滑动窗口内计算 x 元素的最大值。
- MIDPOINT(x, timeperiod=14)
 - 参数:
 - x: 向量/矩阵/表, timeperiod: 长度, 默认14。
 - 在给定长度（以元素个数或时间长度衡量）的滑动窗口内计算 x 元素的最大值和最小值的平均值。
- MIDPRICE(high, low, timeperiod=14)
 - 参数:
 - high: 最高价, 标量/向量/矩阵, low: 最低价, 标量/向量/矩阵, timeperiod: 长度, 默认14。
 - 在给定长度（以元素个数或时间长度衡量）的滑动窗口内计算high序列和low序列最小值的平均值。
- AROONOSC(high, low, timeperiod=14)
 - 参数:
 - high: 最高价, 标量/向量/矩阵, low: 最低价, 标量/向量/矩阵, timeperiod: 长度, 默认14。
 - 在给定长度（以元素个数或时间长度衡量）的滑动窗口内阿隆震荡指标。

def WILLR(high, low, close, timeperiod=14)

- 参数:

high: 最高价, 标量/向量/矩阵, low: 最低价, 标量/向量/矩阵, timeperiod: 长度, 默认14。
在给定长度（以元素个数或时间长度衡量）的滑动窗口内威廉指标, 表示市场属于超买或者超卖。

- CCI(high, low, close, timeperiod=14)
 - 参数:
 - high: 最高价, 标量/向量/矩阵, low: 最低价, 标量/向量/矩阵, close: 收盘价, 标量/向量/矩阵, timeperiod: 长度, 默认14。
 - 在给定长度（以元素个数或时间长度衡量）的滑动窗口内顺势指标, 测量股价是否已超出正常分布范围。
- ADX(high, low, close, timeperiod=14)
 - 参数:
 - high: 最高价, 标量/向量/矩阵, low: 最低价, 标量/向量/矩阵, close: 收盘价, 标量/向量/矩阵, timeperiod: 长度, 默认14。
 - 在给定长度（以元素个数或时间长度衡量）的滑动窗口内平均趋势指标, 指标判断盘整、震荡

和单变趋势。

- MFI(high, low, close, volume, timeperiod=14)
 - 参数:
high: 最高价, 标量/向量/矩阵, low: 最低价, 标量/向量/矩阵, close: 收盘价, 标量/向量/矩阵, volume: 成交量, 标量/向量/矩阵, timeperiod: 长度, 默认14。
在给定长度 (以元素个数或时间长度衡量) 的滑动窗口内资金流量指标, 反映市场的运行趋势。
- NATR(high, low, close, timeperiod=14)
 - 参数:
high: 最高价, 标量/向量/矩阵, low: 最低价, 标量/向量/矩阵, close: 收盘价, 标量/向量/矩阵, timeperiod: 长度, 默认14。
在给定长度 (以元素个数或时间长度衡量) 的滑动窗口内归一化波动幅度均值。
- BETA(x, y, timeperiod=5)
 - 参数:
x: 标量/向量/矩阵, y: 标量/向量/矩阵, timeperiod: 长度, 默认5。
贝塔系数, 给定长度的滑动窗口内x相对y的波动情况。
- LINEARREG_ANGLE(x, timeperiod=14)
 - 参数:
x: 标量/向量/矩阵, timeperiod: 长度, 默认14。
给定长度的滑动窗口内x值为因变量, 序列1,...,timeperiod为自变量的线性回归角度。
- LINEARREG_INTERCEPT(x, timeperiod=14)
 - 参数:
x: 标量/向量/矩阵, timeperiod: 长度, 默认14。
给定长度的滑动窗口内x值为因变量, 序列1,...,timeperiod为自变量的线性回归截距。
- LINEARREG_SLOPE(x, timeperiod=14)
 - 参数:
x: 标量/向量/矩阵, timeperiod: 长度, 默认14。
给定长度的滑动窗口内x值为因变量, 序列1,...,timeperiod为自变量的线性回归斜率。
- EWMA(x, com=None, span=None, half_life=None, alpha=None, min_periods=0, adjust=True, ignore_na=False)
 - 参数:
x: 标量/向量/矩阵; com 是一个大于等于 0 的数值型标量, 表示质心; span 是一个大于等于 1 的数值型标量, 表示跨度;
halfLife 是一个大于 0 的数值型标量, 表示半衰期; alpha 是一个 (0,1] 之间的浮点数, 表示平滑系数;
minPeriods 是一个整数, 表示窗口中的最小观察数, 默认值为 0; adjust 是一个布尔值, 表示是否除以开始阶段的衰减调整因子。默认值为 true;
ignoreNA 是一个布尔值, 表示计算权重时是否忽略 NULL 值。默认值为 false, 即不忽略 NULL 值。
返回 x 的指数加权移动平均值。该函数必须指定 com, span, halfLife, alpha 四个参数中的一个。
- ATR(high, low, close, d)

- 参数:
high: 最高价, 标量/向量/矩阵, low: 最低价, 标量/向量/矩阵, close: 收盘价, 标量/向量/矩阵, d: 长度。
在给定长度 (以元素个数或时间长度衡量) 的滑动窗口内在给定长度 (以元素个数衡量) 的滑动窗口内, 计算真实波幅(|最高价-最低价|和|最高价-昨收|和|昨收-最低价|的最大值)的指数移动平均。
- MACD(x, fastperiod=12, slowperiod=26, signalperiod=9)
 - 参数:
x: 标量/向量/矩阵, fastperiod: 默认12, slowperiod: 默认26, signalperiod: 默认9。
异同移动平均。

2. 因子计算

可以通过calc_factor函数来计算单因子分析中定义的因子值

```
#载入函数库
from onequant.factor.factor_cal import calc_factor

calc_factor(market_code, factor_name, frequency, start_time, end_time)
```

参数:

- market_code: 股票代码列表
- factor_name: 因子代码
- frequency: 周期
- start_time: 开始日期 (时间)
- end_time: 结束日期 (时间)

3. 因子分析

3.1 参数

```
#载入函数库
from onequant.factor.factor_analyzer import FactorAnalyzer

#载入函数库
FactorAnalyzer(factor, prices, groupby=None, weights=1.0,
                quantiles=None, bins=None, periods=(1, 5, 10),
                binning_by_group=False, max_loss=0.25, zero_aware=False)
```

- factor :
因子值, 可输入两种类型的值
 - DataFrame, index 为日期, columns 为资产, values 为因子的值
 - Series, index 为日期和资产的 MultiIndex, values 为因子的值
- prices :
用于计算因子远期收益的价格数据, 可输入两种类型的值
 - DataFrame, index 为日期, columns 为资产。价格数据必须覆盖因子分析时间段以及额外远期收益计算中的最大预期期数。

- function, 输入参数为 securities, start_date, end_date, count, 返回值为价格数据的 DataFrame
- groupby :
分组数据, 默认为 None, 可输入三种类型的值
 - pd.DataFrame, index 为日期, columns 为资产, 为每个资产每天的分组.
 - dict, 资产-分组映射的字典. 如果传递了 dict, 则假定分组映射在整个时间段内保持不变.
 - function, 输入参数为 securities, start_date, end_date, 返回值为分组数据的 DataFrame 或 dict
- weights :
权重数据, 默认为 1, 可输入三种类型的值
 - pd.DataFrame, index 为日期, columns 为资产, 为每个资产每天的权重.
 - dict, 资产-权重映射的字典. 如果传递了 dict, 则假定权重映射在整个时间段内保持不变.
 - function, 输入参数为 securities, start_date, end_date, 返回值为权重数据的 DataFrame 或 dict
- binning_by_group :
bool 类型, 如果为 True, 则对每个组分别计算分位数; 默认为 False。
适用于因子值范围在各个组上变化很大的情况。如果要分析分组(行业)中性的组合, 您最好设置为 True。
- quantiles :
类型为 int or sequence[float], 默认为 None
在因子分组中按照因子值大小平均分组的组数或分位数序列, 允许不均匀分组。
例如 [0, .10, .5, .90, 1.] 或 [.05, .5, .95]
'quantiles' 和 'bins' 有且只能有一个不为 None
- bins :
类型为 int or sequence[float], 默认为 None
在因子分组中使用的等宽 (按照因子值) 区间的数量或边界值序列, 允许不均匀的区间宽度。
例如 [-4, -2, -0.5, 0, 10]
'quantiles' 和 'bins' 有且只能有一个不为 None
- periods :
类型为 int or sequence[int]。
远期收益的期数, 默认为 (1, 5, 10)
- max_loss :
类型为 float, 默认为 0.25。允许的丢弃因子数据的最大百分比 (0.00 到 1.00),
计算比较输入因子索引中的项目数和输出 DataFrame 索引中的项目数。
因子数据本身存在缺陷 (例如 NaN), 没有提供足够的价格数据来计算所有因子值的远期收益,
或者因为分组失败, 因此可以部分地丢弃因子数据。
设置 max_loss = 0 以停止异常捕获。
- zero_aware :
类型为 bool, 默认为 False。如果为 True, 则分别为正负因子值计算分位数。
适用于您的信号聚集并且零是正值和负值的分界线的情况。

3.2 因子分析API

3.2.1 因子值特征分析

```
create_summary_tear_sheet(demeaned=False, group_adjust=False)
```

参数:

demeaned:

- True: 对每日因子收益去均值求得因子收益表
- False: 因子收益表

group_adjust:

- True: 按行业对因子收益去均值后求得因子收益表
- False: 因子收益表

3.2.2 因子收益分析

```
create_returns_tear_sheet(demeaned=False, group_adjust=False, by_group=False)
```

参数:

demeaned:

- True: 使用超额收益计算累积收益 (基准收益被认为是每日所有股票收益按照weight列中权重加权的均值)
- False: 不使用超额收益

group_adjust:

- True: 使用行业中性化后的收益计算累积收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重加权的均值)
- False: 不使用行业中性化后的收益

by_group:

- True: 画各行业的各分位数平均收益图
- False: 不画各行业的各分位数平均收益图

3.2.3 因子 IC 分析

```
create_information_tear_sheet(group_adjust=False, by_group=False)
```

参数:

group_adjust:

- True: 使用行业中性收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)
- False: 不使用行业中性收益

by_group:

- True: 画按行业分组信息比率(IC)图
- False: 画月度信息比率(IC)图

3.2.4 因子换手率分析

```
create_turnover_tear_sheet(turnover_periods=None)
```

参数:

turnover_periods: 调仓周期

3.2.5 因子预测能力分析

```
create_event_returns_tear_sheet(avgretpoint=(5, 15), demeaned=False,  
group_adjust=False, std_bar=False)
```

参数:

avgretpoint: tuple 因子预测的天数

-(计算过去的天数, 计算未来的天数)

demeaned:

- True: 使用超额收益计算累积收益 (基准收益被认为是每日所有股票收益按照weight列中权重加权的均值)

- False: 不使用超额收益

group_adjust:

- True: 使用行业中性化后的收益计算累积收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重加权的均值)

- False: 不使用行业中性化后的收益

std_bar:

- True: 显示标准差

- False: 不显示标准差

3.2.5 全部分析

```
create_full_tear_sheet(demeaned=False, group_adjust=False, by_group=False,  
turnover_periods=None, avgretpoint=(5, 15), std_bar=False)
```

参数:

demeaned:

- True: 使用超额收益计算累积收益 (基准收益被认为是每日所有股票收益按照weight列中权重加权的均值)

- False: 不使用超额收益

group_adjust:

- True: 使用行业中性化后的收益计算累积收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重加权的均值)

- False: 不使用行业中性化后的收益

by_group:

- True: 画各行业的各分位数平均收益图

- False: 不画各行业的各分位数平均收益图

turnover_periods: 调仓周期

avgretpoint: tuple 因子预测的天数

-(计算过去的天数, 计算未来的天数)

std_bar:

- True: 显示标准差
- False: 不显示标准差

3.3 绘制图表

3.3.1 因子收益表

```
plot_returns_table(demeaned=False, group_adjust=False)
```

参数:

demeaned:

- True: 使用超额收益计算 (基准收益被认为是每日所有股票收益按照weight列中权重的加权的均值)
- False: 不使用超额收益

group_adjust:

- True: 使用行业中性收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)
- False: 不使用行业中性收益

3.3.2 换手率表

```
plot_turnover_table()
```

3.3.3 信息比率(IC)相关表

```
plot_information_table(group_adjust=False, method=None)
```

参数:

group_adjust:

- True: 使用行业中性收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)
- False: 不使用行业中性收益

method:

- 'rank': 用秩相关系数计算IC值
- 'normal': 用相关系数计算IC值

3.3.4 各分位数统计表

```
plot_quantile_statistics_table()
```

3.3.5 信息比率(IC)时间序列图

```
plot_ic_ts(group_adjust=False, method=None)
```

参数:

group_adjust:

- True: 使用行业中性收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)
- False: 不使用行业中性收益

method:

- 'rank': 用秩相关系数计算IC值
- 'normal': 用相关系数计算IC值

3.3.6 信息比率(IC)分布直方图

```
plot_ic_hist(group_adjust=False, method=None)
```

参数:

group_adjust:

- True: 使用行业中性收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)

- False: 不使用行业中性收益

method:

- 'rank': 用秩相关系数计算IC值

- 'normal': 用相关系数计算IC值

3.3.7 各分位数平均收益图

```
plot_quantile_returns_bar(by_group=False, demeaned=False, group_adjust=False)
```

参数:

by_group:

- True: 各行业的各分位数平均收益图

- False: 各分位数平均收益图

demeaned:

- True: 使用超额收益计算累积收益 (基准收益被认为是每日所有股票收益按照weight列中权重加权的均值)

- False: 不使用超额收益

group_adjust:

- True: 使用行业中性化后的收益计算累积收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重加权的均值)

- False: 不使用行业中性化后的收益

3.3.8 最高分位减最低分位收益图

```
plot_mean_quantile_returns_spread_time_series(demeaned=False, group_adjust=False, bandwidth=1)
```

参数:

demeaned:

- True: 使用超额收益计算累积收益 (基准收益被认为是每日所有股票收益按照weight列中权重加权的均值)

- False: 不使用超额收益

group_adjust:

- True: 使用行业中性化后的收益计算累积收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重加权的均值)

- False: 不使用行业中性化后的收益

bandwidth: n, 加减 n 倍当日标准差

3.3.9 按行业分组信息比率(IC)图

```
plot_ic_by_group(group_adjust=False, method=None)
```

参数:

group_adjust:

- True: 使用行业中性收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)

- False: 不使用行业中性收益

method:

- 'rank': 用秩相关系数计算IC值

- 'normal': 用相关系数计算IC值

3.3.10 因子自相关图

```
plot_factor_auto_correlation(periods=None, rank=True)
```

参数:

periods: 滞后周期

rank:

- True: 用秩相关系数

- False: 用相关系数

3.3.11 最高最低分位换手率图

```
plot_top_bottom_quantile_turnover(periods=None)
```

参数:

periods: 调仓周期

3.3.12 月度信息比率(IC)图

```
plot_monthly_ic_heatmap(group_adjust=False)
```

参数:

group_adjust:

- True: 使用行业中性收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)

- False: 不使用行业中性收益

3.3.13 按因子值加权组合每日累积收益图

```
plot_monthly_ic_heatmap(period=None, demeaned=False, group_adjust=False)
```

参数:

periods: 调仓周期

demeaned:

- True: 对权重去均值 (每日权重 = 每日权重 - 每日权重的均值), 使组合转换为 cash-neutral 多空组合

- False: 不对权重去均值

group_adjust:

- True: 对权重分行业去均值 (每日权重 = 每日权重 - 每日各行业权重的均值), 使组合转换为 industry-neutral 多空组合

- False: 不对权重分行业去均值

3.3.14 做多最大分位数做空最小分位数组合每日累积收益图

```
plot_top_down_cumulative_returns(period=None, demeaned=False, group_adjust=False)
```

参数:

period: 指定调仓周期

demeaned:

- True: 使用超额收益计算各分位数收益, 超额收益=收益-基准收益 (基准收益被认为是每日所有股票收益按照weight列中权重的加权的均值)

- False: 不使用超额收益

group_adjust:

- True: 使用行业中性收益计算各分位数收益, 行业中性收益=收益-行业收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)

- False: 不使用行业中性收益

3.3.15 各分位数每日累积收益图

```
plot_cumulative_returns_by_quantile(period=None, demeaned=False, group_adjust=False)
```

参数:

period: 指定调仓周期

demeaned:

- True: 使用超额收益计算各分位数收益, 超额收益=收益-基准收益 (基准收益被认为是每日所有股票收益按照weight列中权重的加权的均值)

- False: 不使用超额收益

group_adjust:

- True: 使用行业中性收益计算各分位数收益, 行业中性收益=收益-行业收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)

- False: 不使用行业中性收益

3.3.16 因子预测能力平均累计收益图

```
plot_quantile_average_cumulative_return(periods_before=5, periods_after=10, by_quantile=False, std_bar=False, demeaned=False, group_adjust=False)
```

参数:

periods_before: 计算过去的天数

periods_after: 计算未来的天数

by_quantile: 是否各分位数分别显示因子预测能力平均累计收益图

std_bar:

- True: 显示标准差

- False: 不显示标准差

demeaned:

- True: 使用超额收益计算各分位数收益, 超额收益=收益-基准收益 (基准收益被认为是每日所有股票收益按照weight列中权重的加权的均值)

- False: 不使用超额收益

group_adjust:

- True: 使用行业中性收益计算各分位数收益, 行业中性收益=收益-行业收益 (行业收益被认为是每日各个行业股票收益按照weight列中权重的加权的均值)

- False: 不使用行业中性收益

3.3.17 有效因子数量统计图

```
plot_events_distribution(num_days=5)
```

参数:

num_days: 统计间隔天数

3.3 属性列表

用于访问因子分析的结果

3.3.1 查看因子值

```
factor_data
```

- 类型: pandas.Series
- index: 为日期和股票代码的MultiIndex

3.3.2 去除 nan/inf, 整理后的因子值、forward_return 和分位数

```
clean_factor_data
```

- 类型: pandas.DataFrame
- index: 为日期和股票代码的MultiIndex
- columns: 根据period选择后的forward_return。(如果调仓周期为1天, 那么 forward_return 为 [第二天的收盘价-今天的收盘价]/今天的收盘价), 因子值、行业分组、分位数数组、权重

3.3.3 按分位数分组加权平均因子收益

```
mean_return_by_quantile
```

- 类型: pandas.DataFrame
- index: 分位数分组
- columns: 调仓周期

3.3.4 按分位数分组加权因子收益标准差

```
mean_return_std_by_quantile
```

- 类型: pandas.DataFrame
- index: 分位数分组
- columns: 调仓周期

3.3.5 按分位数及日期分组加权平均因子收益

```
mean_return_by_date
```

- 类型: pandas.DataFrame
- index: 为日期和分位数的MultiIndex
- columns: 调仓周期

3.3.6 按分位数及日期分组加权因子收益标准差

```
mean_return_std_by_date
```

- 类型: pandas.DataFrame
- index: 为日期和分位数的MultiIndex
- columns: 调仓周期

3.3.7 按分位数及行业分组加权平均因子收益

```
mean_return_by_group
```

- 类型: pandas.DataFrame
- index: 为行业 and 分位数的MultiIndex
- columns: 调仓周期

3.3.8 按分位数及行业分组加权因子收益标准差

```
mean_return_std_by_group
```

- 类型: pandas.DataFrame
- index: 为行业 and 分位数的MultiIndex
- columns: 调仓周期

3.3.9 最高分位数因子收益减最低分位数因子收益每日均值

```
mean_return_spread_by_quantile
```

- 类型: pandas.DataFrame
- index: 日期
- columns: 调仓周期

3.3.10 最高分位数因子收益减最低分位数因子收益每日标准差

mean_return_spread_std_by_quantile

- 类型: pandas.DataFrame
- index: 日期
- columns: 调仓周期

3.3.11 各分位数每日累积收益

cumulative_return_by_quantile

- 类型: pandas.DataFrame
- index: 日期
- columns: 调仓周期和分位数

3.3.12 按因子值加权多空组合每日累积收益

cumulative_returns

- 类型: pandas.DataFrame
- index: 日期
- columns: 调仓周期

3.3.13 做多最高分位做空最低分位多空组合每日累计收益

top_down_cumulative_returns

- 类型: pandas.DataFrame
- index: 日期
- columns: 调仓周期

3.3.14 信息比率

ic

- 类型: pandas.DataFrame
- index: 日期
- columns: 调仓周期

3.3.15 分行业信息比率

ic_by_group

- 类型: pandas.DataFrame
- index: 行业
- columns: 调仓周期

3.3.16 月度信息比率

ic_monthly

- 类型: pandas.DataFrame
- index: 月度
- columns: 调仓周期表

3.3.17 换手率

quantile_turnover

- 类型: dict
- 键: 调仓周期
 - index: 日期
 - columns: 分位数分组