

# WorldQuant Operators

- [add](#)
- [log](#)
- [subtract](#)
- [signed\\_power](#)
- [sign](#)
- [reverse](#)
- [power](#)
- [multiply](#)
- [min](#)
- [max](#)
- [inverse](#)
- [densify](#)
- [abs](#)
- [divide](#)
- [equal](#)
- [less](#)
- [and](#)
- [or](#)
- [not\\_equal](#)
- [not](#)
- [greater](#)
- [greater\\_than\\_equal](#)
- [less\\_than\\_equal](#)
- [is\\_nan](#)
- [if\\_else](#)
- [ts\\_mean](#)
- [ts\\_sum](#)
- [ts\\_std\\_dev](#)
- [ts\\_scale](#)
- [ts\\_rank](#)
- [ts\\_quantile](#)
- [ts\\_zscore](#)
- [ts\\_arg\\_min](#)
- [kth\\_element](#)
- [ts\\_corr](#)
- [ts\\_count\\_nans](#)
- [ts\\_covariance](#)
- [ts\\_decay\\_linear](#)
- [ts\\_product](#)
- [ts\\_regression](#)

- [ts\\_step](#)
- [ts\\_delay](#)
- [ts\\_backfill](#)
- [ts\\_av\\_diff](#)
- [hump](#)
- [ts\\_arg\\_max](#)
- [last\\_diff\\_value](#)
- [ts\\_delta](#)
- [days\\_from\\_last\\_change](#)
- [zscore](#)
- [normalize](#)
- [quantile](#)
- [rank](#)
- [scale](#)
- [winsorize](#)
- [vec\\_sum](#)
- [vec\\_avg](#)
- [bucket](#)
- [trade\\_when](#)
- [group\\_zscore](#)
- [group\\_backfill](#)
- [group\\_mean](#)
- [group\\_rank](#)
- [group\\_neutralize](#)

## add

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `add(x, y, filter = false)`,  $x + y$
- **Description:** 将所有输入相加（至少需要2个输入）。如果`filter = true`，则在相加前将所有输入中的NaN替换为0。

## log

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `log(x)`
- **Description:** 自然对数。例如：`Log(high/low)` 使用高/低比率的自然对数作为股票权重。

## subtract

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `subtract(x, y, filter=false)`,  $x - y$
- **Description:**  $x - y$ 。如果`filter = true`，则在相减前将所有输入中的NaN替换为0。

## signed\_power

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `signed_power(x, y)`
- **Description:** 将x的y次幂的结果保留x的符号。例如，当 $y=2$ 时， $x^y$ 是一个抛物线，但`signed_power(x, y)`是一个奇函数，且每个x值对应一个唯一的`signed_power(x, y)`值，与抛物线不同。
- **Documentation:** [https://api.worldquantbrain.com/operators/signed\\_power](https://api.worldquantbrain.com/operators/signed_power)
- **Documentation Content:**

**sign(x) \* (abs(x) ^ y)**

将x的y次幂的结果保留x的符号。例如，当 $y=2$ 时， $x^y$ 是一个抛物线，但`signed_power(x, y)`是一个奇函数，且每个x值对应一个唯一的`signed_power(x, y)`值，与抛物线不同。

**Example:**

如果 $x = 3$ ,  $y = 2 \Rightarrow \text{abs}(x) = 3 \Rightarrow \text{abs}(x) ^ y = 9$  且  $\text{sign}(x) = +1 \Rightarrow \text{sign}(x) * (\text{abs}(x) ^ y) = \text{signed\_power}(x, y) = 9$

如果 $x = -9$ ,  $y = 0.5 \Rightarrow \text{abs}(x) = 9 \Rightarrow \text{abs}(x) ^ y = 3$  且  $\text{sign}(x) = -1 \Rightarrow \text{sign}(x) * (\text{abs}(x) ^ y) = \text{signed\_power}(x, y)$

## sign

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `sign(x)`
- **Description:** 如果输入值为NaN，则返回NaN。

## reverse

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `reverse(x)`
- **Description:** 返回-x的值。

## power

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `power(x, y)`
- **Description:** 计算x的y次幂。
- **Documentation:** <https://api.worldquantbrain.com/operators/power>

## multiply

- **Category:** Arithmetic
- **Scope:** REGULAR

- **Definition:** `multiply(x, y, ..., filter=false), x * y`
- **Description:** 将所有输入相乘。至少需要2个输入。如果`filter = true`，则将所有输入中的NaN替换为1。
- **Documentation:** <https://api.worldquantbrain.com/operators/multiply>

## min

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `min(x, y, ...)`
- **Description:** 返回所有输入中的最小值。至少需要2个输入。
- **Documentation:** <https://api.worldquantbrain.com/operators/min>

## max

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `max(x, y, ...)`
- **Description:** 返回所有输入中的最大值。至少需要2个输入。
- **Documentation:** <https://api.worldquantbrain.com/operators/max>

## inverse

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `inverse(x)`
- **Description:** 计算x的倒数，即 $1/x$ 。

## densify

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:** `densify(x)`
- **Description:** 将包含多个分桶的分组字段转换为较少数量的可用分桶，以提高处理分组字段的计算效率。
- **Documentation:** <https://api.worldquantbrain.com/operators/densify>
- **Documentation Content:**

此操作符将包含多个分桶的分组字段转换为较少数量的可用分桶，以提高处理分组字段的计算效率。以下示例将澄清实现方式。

示例：

假设分组字段以整数形式提供（例如，行业：科技 -> 0，航空 -> 1，...），并且在某一天，我们有一些仪器的分组字段值为{0, 1, 2, 99}。与其创建100个分桶并保留96个空桶，不如只创建4个分桶，其值为{0, 1, 2, 3}。因此，如果x中的唯一值数量为n，则densify将这些值映射到0和(n-1)之间。值的大小顺序不一定保留。

## abs

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:**  $\text{abs}(x)$
- **Description:** 返回 $x$ 的绝对值。

## divide

- **Category:** Arithmetic
- **Scope:** REGULAR
- **Definition:**  $\text{divide}(x, y)$
- **Description:** 计算 $x$ 除以 $y$ 的结果。

## equal

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:**  $\text{input1} == \text{input2}$
- **Description:** 如果两个输入值相同，则返回`true`；否则返回`false`。

## less

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:**  $\text{input1} < \text{input2}$
- **Description:** 如果输入1小于输入2，则返回`true`；否则返回`false`。

## and

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:**  $\text{and}(\text{input1}, \text{input2})$
- **Description:** 逻辑与操作符，只有当两个操作数都为`true`时返回`true`，否则返回`false`。

## or

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:**  $\text{or}(\text{input1}, \text{input2})$
- **Description:** 逻辑或操作符，当两个输入值中至少有一个为`true`时返回`true`，否则返回`false`。

## not\_equal

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:**  $\text{input1} != \text{input2}$
- **Description:** 如果两个输入值不相同，则返回`true`；否则返回`false`。

## not

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:** not(x)
- **Description:** 返回x的逻辑取反。如果x为true（1），则返回false（0）；如果x为false（0），则返回true（1）。

## greater

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:** input1 > input2
- **Description:** 如果输入1大于输入2，则返回true；否则返回false。

## greater\_than\_equal

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:** input1 >= input2
- **Description:** 如果输入1大于或等于输入2，则返回true；否则返回false。

## less\_than\_equal

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:** input1 <= input2
- **Description:** 如果输入1小于或等于输入2，则返回true；否则返回false。

## is\_nan

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:** is\_nan(input)
- **Description:** 如果输入值为NaN，则返回1；否则返回0。

## if\_else

- **Category:** Logical
- **Scope:** REGULAR
- **Definition:** if\_else(input1, input2, input3)
- **Description:** 如果输入1为true，则返回输入2；否则返回输入3。
- **Documentation:** [https://api.worldquantbrain.com/operators/if\\_else](https://api.worldquantbrain.com/operators/if_else)
- **Documentation Content:**

```
if_else(event_condition, Alpha_expression_1, Alpha_expression_2)
```

如果提供的事件条件为true，则返回Alpha\_expression\_1；如果提供的事件条件为false，则返回Alpha\_expression\_2。

示例：

我们感兴趣的是测试这样一个假设：如果一家公司的股票价格在过去两天内有所上涨，那么它在未来可能会下跌。此外，如果今天买卖股票的数量高于月平均水平，那么反转效应可能会更加明显。

我们将通过根据今天收盘价与3天前的差值来实施这一假设，使用`ts\_delta`操作符。当当前成交量高于平均日成交量时，我们将通过乘以2来获取更大的头寸，即`alpha\_1`。

## ts\_mean

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_mean(x, d)`
- **Description:** 返回过去d天内x的平均值。

## ts\_sum

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_sum(x, d)`
- **Description:** 计算过去d天内x的总和。

## ts\_std\_dev

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_std_dev(x, d)`
- **Description:** 返回过去d天内x的标准差。

## ts\_scale

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_scale(x, d, constant = 0)`
- **Description:** 返回过去d天内x的标准化值，计算公式为： $(x - ts\_min(x, d)) / (ts\_max(x, d) - ts\_min(x, d)) + constant$ 。此操作符类似于缩放操作符，但作用于时间序列空间。
- **Documentation:** [https://api.worldquantbrain.com/operators/ts\\_scale](https://api.worldquantbrain.com/operators/ts_scale)
- **Documentation Content:**

此操作符返回过去d天内x的标准化值，计算公式为： $(x - ts\_min(x, d)) / (ts\_max(x, d) - ts\_min(x, d)) + constant$ 。

#### Example:

如果  $d = 6$ ，过去6天的值为  $[6, 2, 8, 5, 9, 4]$ ，其中第一个元素为今天的值，则  $ts\_min(x, d) = 2$ ， $ts\_max(x, d) = 9$ 。

$$ts\_scale(x, d, constant = 1) = 1 + (6 - 2) / (9 - 2) = 1.57$$

## ts\_rank

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:**  $ts\_rank(x, d, constant = 0)$
- **Description:** 对过去  $d$  天内  $x$  的值进行排名，然后返回当前值的排名 +  $constant$ 。如果未指定， $constant$  默认为 0。

## ts\_quantile

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:**  $ts\_quantile(x, d, driver="gaussian")$
- **Description:** 计算过去  $d$  天内  $x$  的排名，并将其值应用于驱动分布的逆累积密度函数。 $driver$ （可选）的可能值为 "gaussian"、"uniform"、"cauchy"，默认为 "gaussian"。

## ts\_zscore

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:**  $ts\_zscore(x, d)$
- **Description:** Z分数是描述一组值中某个值与均值关系的数值测量方法。Z分数以标准差的形式衡量与均值的距离： $(x - ts\_mean(x, d)) / ts\_stddev(x, d)$ 。此操作符有助于减少异常值和回撤。

## ts\_arg\_min

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:**  $ts\_arg\_min(x, d)$
- **Description:** 返回过去  $d$  天内时间序列中最小值的相对索引；如果当前天是过去  $d$  天内的最小值，则返回 0；如果前一天是过去  $d$  天内的最小值，则返回 1。
- **Documentation:** [https://api.worldquantbrain.com/operators/ts\\_arg\\_min](https://api.worldquantbrain.com/operators/ts_arg_min)
- **Documentation Content:**

```
ts_arg_min(x, d)
```



返回过去d天内时间序列中最小值的相对索引。如果当前天是过去d天内的最小值，则返回0；如果前一天是过去d天内的最小值，则返回1。

**Example:**

如果d = 6，过去6天的值为[6, 2, 8, 5, 9, 4]，其中第一个元素为今天的值，则最小值为2，且它出现在4天前。因此，`ts_arg_min(x, d) = 4`

## kth\_element

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `kth_element(x, d, k)`
- **Description:** 返回过去d天内输入的第k个值。此操作符也可用于填补缺失数据，当k=1时。
- **Documentation:** [https://api.worldquantbrain.com/operators/kth\\_element](https://api.worldquantbrain.com/operators/kth_element)
- **Documentation Content:**

返回过去d天内输入的第k个值，同时忽略ignore列表中指定的标量。

**ignore**参数用于提供要从计数中忽略的分隔标量列表。

**Example of backfill:**

## ts\_corr

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_corr(x, y, d)`
- **Description:** 返回过去d天内x和y的相关性。
- **Documentation:** [https://api.worldquantbrain.com/operators/ts\\_corr](https://api.worldquantbrain.com/operators/ts_corr)
- **Documentation Content:**

`ts_corr(x, y, d)`

皮尔逊相关性衡量两个变量之间的线性关系。当变量呈正态分布且关系为线性时，效果最佳。

```
Correlation(x,y) = \frac{\sum_{i=t-d+1}^t (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=t-d+1}^t (x_i - \bar{x})^2 (y_i - \bar{y})^2}}
```

**Example:**

## ts\_count\_nans

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** ts\_count\_nans(x, d)
- **Description:** 返回过去d天内x中的NaN值数量。

## ts\_covariance

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** ts\_covariance(y, x, d)
- **Description:** 返回过去d天内y和x的协方差。

## ts\_decay\_linear

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** ts\_decay\_linear(x, d, dense = false)
- **Description:** 返回过去d天内x的线性衰减值。当dense参数为false时，操作符以稀疏模式工作，我们将NaN视为0。在密集模式下则不这样处理。
- **Documentation:** [https://api.worldquantbrain.com/operators/ts\\_decay\\_linear](https://api.worldquantbrain.com/operators/ts_decay_linear)
- **Documentation Content:**

```
ts_decay_linear(x, d, dense = false)
```

返回过去d天内x的线性衰减值。当dense参数为false时，操作符以稀疏模式工作，我们将NaN视为0。在密集模式下则不这样处理。数据平滑技术如线性衰减通过为旧观测值应用衰减因子来减少时间序列数据中的噪声，从而有助于稳定数据集。

此操作符有助于改善换手率和回撤。

**Example:**

- 假设某股票过去5天的价格如下：

- 第0天：30（异常值）

- 第-1天：5

- 第-2天：4

- 第-3天：5

- 第-4天：6

- 计算过程如下：

- 分子 =  $(30 \cdot 5) + (5 \cdot 4) + (4 \cdot 3) + (5 \cdot 2) + (6 \cdot 1) = 150 + 20 + 12 + 10 + 6 = 198$

- 分母 =  $5 + 4 + 3 + 2 + 1 = 15$

- 加权平均值 =  $198 / 15 = 13.2$

- 使用加权平均值13.2代替异常值20来进行权重分配。

## ts\_product

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_product(x, d)`
- **Description:** 返回过去d天内x的乘积。

## ts\_regression

- **Category:** Time Series
- **Scope:** REGULAR

- **Definition:** `ts_regression(y, x, d, lag = 0, rettype = 0)`
- **Description:** 返回与回归函数相关的各种参数。
- **Documentation:** [https://api.worldquantbrain.com/operators/ts\\_regression](https://api.worldquantbrain.com/operators/ts_regression)
- **Documentation Content:**

```
ts_regression(y, x, d, lag = 0, rettype = 0)
```

给定过去d天内两个变量的值（X：自变量，Y：因变量），可以定义一个近似线性函数，使得该集合上的平方误差之和最小：

Beta和Alpha在第二行是OLS线性回归系数。

`ts_regression`操作符返回与所述回归相关的各种参数。这由“`rettype`”关键字参数控制，其默认值为0。其他“`rettype`”参数值对应于：

这里，“`di`”是当前天的索引，“`n`”（可能与d不同）是用于计算的有效（`x`，`y`）元组的数量。所有求和都是在天索引上进行的，仅使用有效元组。

“`lag`”关键字参数可以可选地指定（默认值为零），以计算滞后回归参数：

#### **Example:**

。

- `ts_regression(est_netprofit, est_netdebt, 252, lag = 0, rettype = 2)`
- 使用过去252个交易日（1年）的数据，返回使用`est_netdebt`估计`est_netprofit`时的 $\beta$ 系数。

## **ts\_step**

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_step(1), step(1)`

2025 WorldQuant -  
Kusu

- **Description:** 返回天数计数器。

## ts\_delay

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_delay(x, d)`
- **Description:** 返回d天前的x值。

## ts\_backfill

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_backfill(x, lookback = d, k=1, ignore="NaN")`
- **Description:** 如果某个数据字段x的输入值为NaN，则用过去d天内最近的非NaN值替换。如果设置了k参数，则返回过去d天内第k个最近的非NaN值。
- **Documentation:** [https://api.worldquantbrain.com/operators/ts\\_backfill](https://api.worldquantbrain.com/operators/ts_backfill)
- **Documentation Content:**

```
ts_backfill(x, lookback = d, k=1, ignore="NaN")
```

如果某个数据字段x的输入值为NaN，则从过去d天内相同数据字段的可用输入值中，输出最近的非NaN值。如果设置了k参数，则输出过去d天内第k个最近的非NaN值。

此操作符有助于提高权重覆盖率，并可能降低回撤风险。

### Example:

- 如果数据字段x的输入值为非NaN，则输出 = x
- 如果数据字段x的输入值为NaN，则输出 = 过去252天内最近的非NaN值

## ts\_av\_diff

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:** `ts_av_diff(x, d)`
- **Description:** 返回 $x - ts\_mean(x, d)$ ，但会小心处理NaN值。即在计算均值时忽略NaN。

- **Documentation:** [https://api.worldquantbrain.com/operators/ts\\_av\\_diff](https://api.worldquantbrain.com/operators/ts_av_diff)
- **Documentation Content:**

此操作符返回  $x - ts\_mean(x, d)$ ，但会小心处理NaN值。

**Example:**

如果  $d = 6$ ，过去6天的值为  $[6, 2, 8, 5, 9, NaN]$ ，则  $ts\_mean(x, d) = 6$ （因为NaN在计算均值时被忽略）。因此， $ts\_av\_diff(x, d) = 6 - 6 = 0$

## hump

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:**  $hump(x, hump = 0.01)$
- **Description:** 限制输入值的变化量和变化幅度（从而减少换手率）。
- **Documentation:** <https://api.worldquantbrain.com/operators/hump>
- **Documentation Content:**

$hump(x, hump = 0.01)$

此操作符限制Alpha的变化频率和幅度（从而减少换手率）。如果今天的值与昨天的值之间的变化不超过阈值，则hump操作符的输出与昨天相同。如果变化超过限制，则输出为昨天的值加上变化方向上的限制。

此操作符有助于减少换手率和回撤。

## ts\_arg\_max

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:**  $ts\_arg\_max(x, d)$
- **Description:** 返回过去d天内时间序列中最大值的相对索引。如果当前天是过去d天内的最大值，则返回0；如果前一天是过去d天内的最大值，则返回1。
- **Documentation:** [https://api.worldquantbrain.com/operators/ts\\_arg\\_max](https://api.worldquantbrain.com/operators/ts_arg_max)
- **Documentation Content:**

返回过去d天内时间序列中最大值的相对索引。如果当前天是过去d天内的最大值，则返回0；如果前一天是过去d天内的最大值，则返

回1。

**Example:**

如果  $d = 6$ ，过去6天的值为  $[6, 2, 8, 5, 9, 4]$ ，其中第一个元素为今天的值，则最大值为9，且它出现在4天前。因此， $ts\_arg\_max(x, d) = 4$

## last\_diff\_value

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:**  $last\_diff\_value(x, d)$
- **Description:** 返回过去d天内最后一个不等于当前x值的x值。

## ts\_delta

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:**  $ts\_delta(x, d)$
- **Description:** 返回x与d天前的x值之差，即  $x - ts\_delay(x, d)$ 。

## days\_from\_last\_change

- **Category:** Time Series
- **Scope:** REGULAR
- **Definition:**  $days\_from\_last\_change(x)$
- **Description:** 返回自x最后一次变化以来的天数。

## zscore

- **Category:** Cross Sectional
- **Scope:** REGULAR
- **Definition:**  $zscore(x)$
- **Description:** Z分数是描述一组值中某个值与均值关系的数值测量方法。Z分数以标准差的形式衡量与均值的距离。
- **Documentation:** <https://api.worldquantbrain.com/operators/zscore>
- **Documentation Content:**

**zscore(x)**

Z分数是一种统计工具，用于指示一个数据点相对于一组值的均值的偏离程度。它衡量一个数据点相对于均值的异常程度，是理解和比较数据偏差的有用工具。

z分数的计算公式为：

$$Z\text{-score} = \frac{x - \text{mean}(x)}{\text{std}(x)}$$

其中：

- $x$  是单个数据点
- $\text{mean}(x)$  是数据集的均值
- $\text{std}(x)$  是数据集的标准差

根据这一定义，z分数分布的均值始终为0，标准差始终为1。

如果z分数为正，则数据点高于均值；如果为负，则低于均值。

z分数特别适用于标准化和比较不同股票或不同数据字段的值。它们允许研究人员计算分数在标准正态分布内的概率，并比较来自不同样本（可能具有不同均值和标准差）的两个分数。

此操作符有助于减少异常值。

## normalize

- **Category:** Cross Sectional
- **Scope:** REGULAR
- **Definition:** `normalize(x, useStd = false, limit = 0.0)`
- **Description:** 计算某一天所有有效Alpha值的均值，然后从每个元素中减去该均值。
- **Documentation:** <https://api.worldquantbrain.com/operators/normalize>
- **Documentation Content:**

```
normalize(x, useStd = false, limit = 0.0)
```



此操作符计算某一天所有有效Alpha值的均值，然后从每个元素中减去该均值。如果useStd = true，则操作符计算结果值的标准差，并将每个标准化元素除以它。如果limit不等于0.0，则操作符将结果Alpha值限制在- limit到+ limit之间。

#### Example:

如果某一天，某个输入x的工具值为[3, 5, 6, 2]，则均值 = 4，标准差 = 1.82

```
normalize(x, useStd = false, limit = 0.0) = [3-4, 5-4, 6-4, 2-4] = [-1, 1, 2, -2]
```

```
normalize(x, useStd = true, limit = 0.0) = [-1/1.82, 1/1.82, 2/1.82, -2/1.82] = [-0.55, 0.55, 1.1, -1.1]
```

## quantile

- **Category:** Cross Sectional
- **Scope:** REGULAR
- **Definition:** quantile(x, driver = gaussian, sigma = 1.0)
- **Description:** 对原始向量进行排名，移动排名后的Alpha向量，并应用驱动分布（高斯、柯西、均匀）。如果驱动器为均匀分布，则只需从Alpha向量的所有Alpha值中减去均值。
- **Documentation:** <https://api.worldquantbrain.com/operators/quantile>
- **Documentation Content:**

```
quantile(x, driver = gaussian, sigma = 1.0)
```

对输入的原始Alpha向量进行排名

排名后的Alpha值将在[0, 1]范围内

#### 1. 移动排名后的Alpha向量

对于排名后的Alpha向量中的每个Alpha值，按照以下公式进行移动： $\text{Alpha\_value} = 1/N + \text{Alpha\_value} * (1 - 2/N)$ ，其中假设Alpha向量中有N个工具。移动后的Alpha值将在 $[1/N, 1-1/N]$ 范围内。

#### 2. 使用指定的驱动器为排名后的Alpha向量中的每个Alpha值应用分布。驱动器可以是"gaussian"、"uniform"、

"cauchy".

注意：Sigma仅影响最终值的尺度。

此操作符有助于减少异常值。

#### Example

## rank

- **Category:** Cross Sectional
- **Scope:** REGULAR
- **Definition:** rank(x, rate=2)
- **Description:** 对输入数据x在所有工具中的值进行排名，并返回一个在0.0到1.0之间均匀分布的浮点数。如果将rate设置为0，则进行精确排序。
- **Documentation:** <https://api.worldquantbrain.com/operators/rank>
- **Documentation Content:**

```
rank(x, rate=2):
```

Rank操作符对给定股票的输入数据x在所有工具中的值进行排名，并返回一个在0.0到1.0之间均匀分布的浮点数。当rate设置为0时，进行精确排序。rate的默认值为2。

此操作符有助于减少异常值和回撤，同时提高夏普比率。

#### Example:

```
Rank(close); Rank(close, rate=0) # 精确排序
```

```
X = (4, 3, 6, 10, 2) => Rank(x) = (0.5, 0.25, 0.75, 1, 0)
```

## scale

- **Category:** Cross Sectional
- **Scope:** REGULAR
- **Definition:** `scale(x, scale=1, longscale=1, shortscale=1)`
- **Description:** 将输入按书籍规模进行缩放。我们还可以通过为操作符指定附加参数，将多头和空头头寸分别缩放到不同的规模。
- **Documentation:** <https://api.worldquantbrain.com/operators/scale>
- **Documentation Content:**

```
scale(x, scale=1, longscale=1, shortscale=1)
```

该操作符将输入缩放到书籍规模。我们可以通过指定附加参数 'scale=booksize\_value' 来调整书籍规模。我们还可以通过指定附加参数: `longscale=long_booksize` 和 `shortscale=short_booksize`, 将多头和空头头寸分别缩放到不同的规模。每个比例的默认值为0, 表示不进行缩放, 除非另有说明。将alpha缩放, 使得所有工具的`abs(x)`之和等于1。若要缩放到不同的书籍规模, 使用`Scale(x) * booksize`。

此操作符有助于减少异常值。

#### Examples

```
scale(returns, scale=4); scale(returns, scale=1) + scale(close, scale=20); scale(returns, longscale=4, shortscale=3)
```

## winsorize

- **Category:** Cross Sectional
- **Scope:** REGULAR
- **Definition:** `winsorize(x, std=4)`
- **Description:** 对x进行截断, 确保x中的所有值都在上下限之间, 上下限由标准差的倍数指定。详情可参考维基百科。

## vec\_sum

- **Category:** Vector
- **Scope:** REGULAR
- **Definition:** `vec_sum(x)`
- **Description:** 计算向量字段x的总和。

## vec\_avg

- **Category:** Vector
- **Scope:** REGULAR
- **Definition:** `vec_avg(x)`

- **Description:** 计算向量字段x的平均值。

## bucket

- **Category:** Transformational
- **Scope:** REGULAR
- **Definition:** `bucket(rank(x), range="0, 1, 0.1" or buckets = "2,5,6,7,10")`
- **Description:** 将浮点值转换为用户指定的分桶索引。分桶可用于创建组值，这些组值可以作为GROUP操作符的输入。
- **Documentation:** <https://api.worldquantbrain.com/operators/bucket>
- **Documentation Content:**

### Bucket

将浮点值转换为用户指定的分桶索引。分桶可用于创建组值，这些组值可以作为GROUP操作符的输入。

如果**buckets**指定为"`num_1, num_2, ..., num_N`", 则将其转换为由 `[(num_1, num_2, idx_1), (num_2, num_3, idx_2), ..., (num_N-1, num_N, idx_N-1)]` 组成的括号。

因此，当**buckets**="`2, 5, 6, 7, 10`"时，向量"`-1, 3, 6, 8, 12`"将变为"`0, 1, 2, 4, 5`"。

如果**range**指定为"`start, end, step`", 则将其转换为由 `[(start, start+step, idx_1), (start+step, start+2*step, idx_2), ..., (start+N*step, end, idx_N)]` 组成的括号。

因此，当**range**="`0.1, 1, 0.1`"时，向量"`0.05, 0.5, 0.9`"将变为"`0, 4, 8`"。

注意，默认情况下会添加两个隐藏的分桶，分别对应于 `(-inf, start]` 和 `[end, +inf)`。使用**skipBegin**和**skipEnd**参数可以移除这些分桶。使用**skipBoth**可以同时将**skipEnd**和**skipBegin**设置为true。

如果你想将所有NaN值分配到一个单独的组中，请使用**NANGroup**。索引值将位于最后一个分桶之后。

### Examples:

```
my_group = bucket(rank(volume), range="0.1,1,0.1");
```

```
group_neutralize(sales/assets, my_group)
```

```
my_group = bucket(rank(volume), buckets = "0.2,0.5,0.7", skipBoth=True, NANGroup=True);

group_neutralize(sales/assets, my_group)
```

## trade\_when

- **Category:** Transformational
- **Scope:** REGULAR
- **Definition:** trade\_when(x, y, z)
- **Description:** 仅在指定条件下更改Alpha值，并在其他情况下保留Alpha值。它还可以在指定条件下关闭Alpha头寸（将Alpha值设置为NaN）。
- **Documentation:** [https://api.worldquantbrain.com/operators/trade\\_when](https://api.worldquantbrain.com/operators/trade_when)
- **Documentation Content:**

此操作符可以在指定条件下更改Alpha值，并在其他情况下保留Alpha值。它还可以在指定条件下关闭Alpha头寸（将Alpha值设置为NaN）。

```
Trade_When (x=triggerTradeExp, y=AlphaExp, z=triggerExitExp)
```

如果triggerExitExp > 0, 则Alpha = NaN。

如果triggerTradeExp > 0, 则Alpha = AlphaExp;

否则, Alpha = previousAlpha

此操作符有助于减少相关性和换手率。

### Examples:

```
Trade_When (volume >= ts_sum(volume,5)/5, rank(-returns), -1)
```

如果 (volume >= ts\_sum(volume,5)/5), 则Alpha = rank(-returns);

否则，交易前一个Alpha值；

退出条件始终为假。

```
Trade_When (volume >= ts_sum(volume,5)/5, rank(-returns), abs(returns) > 0.1)
```

如果`abs(returns) > 0.1`，则`Alpha = nan`；

否则，如果`volume >= ts_sum(volume,5)/5`，则`Alpha = rank(-returns)`；

否则，交易前一个Alpha值。

## group\_zscore

- **Category:** Group
- **Scope:** REGULAR
- **Definition:** `group_zscore(x, group)`
- **Description:** 计算组Z分数——描述一个值与一组值的均值关系的数值测量方法。Z分数以标准差的形式衡量与均值的距离。 $zscore = (\text{数据} - \text{均值}) / x \text{的标准差}$ ，针对每个工具在其所属组内。

## group\_backfill

- **Category:** Group
- **Scope:** REGULAR
- **Definition:** `group_backfill(x, group, d, std = 4.0)`
- **Description:** 如果某个日期和工具的某个值为NaN，则从同一组工具中计算过去d天内所有非NaN值的截断均值。
- **Documentation:** [https://api.worldquantbrain.com/operators/group\\_backfill](https://api.worldquantbrain.com/operators/group_backfill)
- **Documentation Content:**

```
group_backfill(x, group, d, std = 4.0)
```

如果某个日期和工具的某个值为NaN，则从同一组工具中计算过去d天内所有非NaN值的截断均值。截断均值意味着输入值被截断为  $std * stddev$ ，其中`stddev`是输入值的标准差。

**Example:**

如果  $d = 4$ ，且某个组中有3个工具 ( $i1, i2, i3$ )，它们在过去4天内的值分别为  $x[i1] = [4, 2, 5, 5]$ ， $x[i2] = [7, \text{NaN}, 2, 9]$ ， $x[i3] = [\text{NaN}, -4, 2, \text{NaN}]$ ，其中第一个元素为最近的值，则如果我们要对  $x$  进行填补，我们只需要填补  $x[i3]$  的第一个元素，因为其他工具的第一个元素均为非NaN值。

其他组的非NaN值为  $[4, 2, 5, 5, 7, 2, 9, -4, 2]$ 。均值 = 3.56，标准差为3.71，且没有任何值超出  $3.56 - 4 * 3.71$  和  $3.56 + 4 * 3.71$  的范围。因此，我们不需要将元素截断到这些限制范围内。因此，截断均值 = 填补值 = 3.56。

对于三个工具，`group_backfill(x, group, d, std = 4.0) = [4, 7, 3.56]`

## group\_mean

- **Category:** Group
- **Scope:** REGULAR
- **Definition:** `group_mean(x, weight, group)`
- **Description:** 组内所有元素的值等于该组的均值。

## group\_rank

- **Category:** Group
- **Scope:** REGULAR
- **Definition:** `group_rank(x, group)`
- **Description:** 组内每个元素被分配其在该组中的相应排名。
- **Documentation:** [https://api.worldquantbrain.com/operators/group\\_rank](https://api.worldquantbrain.com/operators/group_rank)
- **Documentation Content:**

**group\_rank(x, group)**

组操作符是一种跨截面操作符，它在更细的层面上比较股票，其中跨截面操作应用于每个组内，而不是整个市场。group\_rank操作符将股票分配到你指定的组中，然后在每个组内根据数据字段x的输入值对股票进行排名，并返回一个在0.0到1.0之间均匀分布的数字。

此操作符有助于减少异常值和回撤，同时降低相关性。

**Example:** `group_rank(x, subindustry)`

- 股票首先被分配到你各自的子行业中。

- 在每个子行业内，根据数据字段x的输入值对子行业内的股票进行排名，并分配一个在0.0到1.0之间均匀分布的数字。

## group\_neutralize

- **Category:** Group
- **Scope:** REGULAR
- **Definition:** group\_neutralize(x, group)
- **Description:** 对组进行Alpha中性化。这些组可以是子行业、行业、板块、国家或常量。
- **Documentation:** [https://api.worldquantbrain.com/operators/group\\_neutralize](https://api.worldquantbrain.com/operators/group_neutralize)
- **Documentation Content:**

**group\_neutralize(x, group)**

对组进行Alpha中性化。normalize和group\_neutralize之间的区别在于，在normalize中，每个元素都减去当天所有工具的所有值的均值；而在group\_neutralize中，元素减去其所属组的工具当天所有值的均值。

此操作符有助于减少相关性，具体取决于使用的中性化方法。

### **Example:**

如果在某一天，某个字段x的值为10个工具[3, 2, 6, 5, 8, 9, 1, 4, 8, 0]，其中前5个工具属于一个组，后5个工具属于另一个组，那么第一个组的均值为  $(3+2+6+5+8)/5 = 4.8$ ，第二个组的均值为  $(9+1+4+8+0)/5 = 4.4$ 。从各自组的工具中减去均值后得到 [-1.8, -2.8, 1.2, 0.2, 3.2, 4.6, -3.4, -0.4, 3.6, -4.4]