WebSim v1.0

Code: `Call_i( (6*(n+1)*sum(vwap, n) - 12*Sum_i(i*Delay(vwap,i), i, 1, n, 1) ) / (n^3 - n), n, 5)`

Max Chains (futures only, 0 means ALL): 0

Max stock fraction (between 0 and 1, 0 means no truncate): 0

Decay (# of days, 0 means no decay): 20

Delay: 0

Region / Universe: USA: TOP3000

Neutralization: subindustry **Futures can only be neutralized on market & chain**

Category: price_reversion

Submit

In the simulation, you enter an expression that consists of operators, data, and constants. The expression is evaluated for each stock to construct a portfolio. Then we make investments in each stock for a one-day period in proportion to the values of the expression. The process repeats each day.

Note that the "raw" values (i.e., the values of the expression) are not used directly. A "neutralization" operation is implicitly applied to calculate the final values. With the final values, negative numbers would result in short positions, and positive numbers long positions.2f9NUp40MRY4B

You can write your expression using the following data:

Data Name	Description	Equities	Futures 0
open	open price	yes	yes
close	close price	yes	yes
high	high price	yes	yes
low	high price	yes	yes
volume	daily volume	yes	yes
vwap	vwap price	yes	no
returns	daily returns	yes	yes
adv20	20-day average daily volume	yes	yes
sharesout	shares outstanding	yes	no
sector	sector	yes	no
industry	industry	yes	no
subindustry	subindustry	yes	no
openint	open interest	no	yes
openint_calls	open interest on calls	no	yes
openint_puts	open interest on puts	no	yes

and the following operators:

Operator	Description
+	
-	
*	
/	
^	Power
<	
<=	
>	
>=	
==	
!=	
||	Logical OR
&&	Logical AND
!	Logical negation
cond ? expr1 : expr2	If cond is true, expr1; else expr2. For example, close < open ? close : open
Rank(x)	Rank the values in vector x and the return values are between 0.0 and 1.0
Min(x, y)	Parallel minimum of vectors x and y (similar to the pmin function in R)
Max(x, y)	Parallel maximum of vectors x and y (similar to the pmax function in R)
StdDev(x, n)	Standard deviation of the values in vector x for the past n days. Note that n must be less than 256
Correlation(x, y, n)	Correlation of the values in vectors x and y for the past n days. Note that n must be less than 256
Sum(x, n)	Sum of the values in vector x for the past n days. Note that n must be less than 256
Covariance(x, y, n)	Covariance of the values in vectors x and y for the past n days. Note that n must be less than 256
CountNans(x, n)	Number of NaN values in vector x for the past n days. Note that n must be less than 256
Abs(x)	Absolute value
Delay(x, n)	Value of x at n days ago. Note that n must be less than 256
Step(x)	For all stocks, current day is x, yesterday is x-1, and so on
Delta(x, n)	x[date] - x[date - n]. Note that n must be less than 256
Decay_linear(x, n)	Linear decay over the last n days
Decay_exp(x, f, n)	Exponential decay. f is the smoothing factor, and the process starts from n days ago. For example, Decay_exp(close, 0.1, 20)
Product(x, n)	Product of the values in vector x for the past n days

Tail(x, lower, upper, newval)	Set the values of x to newval if they are between lower and upper
Ts_Min(x, n)	Minimum value of x over the last n days. Note that this is different than Min
Ts_Max(x, n)	Maximum value of x over the last n days. Note that this is different than Max
Sum_i(expr, var, start, stop, step)	Loop over var (from start to stop with step) and calculate expr at every iteration (presumably expr would contain var), then sum over all the values. For example, Sum_i(Delay(close, i)*i, i, 2, 4, 1) would be equivalent to Delay(close, 2)*2 + Delay(close, 3)*3 + Delay(close, 4)*4
Call_i(expr, var, subexpr)	Substitute subexpr for var in expr, and then evaluate expr. For example, Call_i(x + 4, x, 2 + 3) would be equivalent to (2 + 3) + 4
Sign(x)	1 if x > 0, -1 if x < 0, 0 if x == 0
SignedPower(x, e)	Sign(x) * (Abs(x)^e)
Pasteurize(x)	Pasteurize the signals. Set to NaN if it is INF or if the underlying instrument is not in the universe
Log(x)	Natural logarithm
Ts_Rank(x, n)	Rank the values of x over the past n days, then return the rank of the current value. For example, if the current value is the max of the past n days, Ts_Rank is 1. If it's a min, Ts_Rank is 0. With all other values in between
Ts_Skewness(x, n)	Compute the skewness of x on the last n days
Ts_Kurtosis(x, n)	Compute the kurtosis of x on the last n days
Ts_Moment(x, k, n)	Compute the kth central moment of x on the last n days
IndNeutralize(x, y)	Neutralize alpha x against groupings specified by y. For example, IndNeutralize(x, industry). To neutralize against market, use IndNeutralize(x, 1)
Scale(x)	Scale alpha x so that its booksize is 1. To scale to a different book size, say 1000, use Scale(x) * 1000

Sample expressions (copy & paste into the above):

```
-(1 / close)
```

-or-

```
Rank(volume) / (((Rank(volume) * open) / close)^2)
```