# Activity1

## 1.1

userparams

SHORT\_TERM\_PERIOD: "The number of minutes to look back": 15;

BENPERIOD: "The number of trading days for benchmarking": 30;

ST\_DEV: "The number of standard deviations away from distribution mean": 2;

end userparams

declare threshold[security]: percent

declare change\_price\_dist[security]: distribution

on trade

declare let one\_minute = 1 minutes

declare let tap = SHORT\_TERM\_PERIOD \* one\_minute

declare let time\_tap = time - tap

if controlstatus(security) = "O" then

declare let comp\_price = trueprice(time\_tap)

if undefined(comp\_price) then

comp\_price = lastprice

end if

if defined(comp\_price) then

declare let price\_change = abs(change(price, comp\_price))

declare let dir\_str = format(price\_change > 0%,"increased","decreased")

if price\_change > threshold[security] then

alert 101, "STPM", "SHORT TERM PRICE MOVEMENT: [security] has

[dir\_str] in price from [comp\_price] to [price] in the last [SHORT\_TERM\_PERIOD]

minutes. This is an [dir\_str] of [price\_change], which

is greater than the threshold of [threshold[security]].",

intensity = (price\_change - threshold[security])/price\_change \* 100,

reissue = "101S+10"

end if

end if

end if

end on

Benchmarks\_below: BENPERIOD trdays

on trade

declare let one\_minute = 1 minutes

declare let tap = SHORT\_TERM\_PERIOD \* one\_minute

declare let time\_tap = time - tap

declare let comp\_price = trueprice(time\_tap)

if undefined(comp\_price) then

comp\_price = lastprice

end if

declare let price\_change = abs(change(price, comp\_price))

if defined(price\_change) then

change\_price\_dist[security] <- price\_change

end if

end on

at end

per security

if defined(change\_price\_dist[security]) then

declare let mean = distaverage(change\_price\_dist[security])

declare let std = diststdev(change\_price\_dist[security])

threshold[security] = (mean + ST\_DEV\*std) \* 100%

end if

end per

end at

## 1.2

userparams

SHORT\_TERM\_PERIOD: "The number of minutes to look back": 15;

BENPERIOD: "The number of trading days for benchmarking": 30;

ST\_DEV: "The number of standard deviations away from distribution mean": 2;

HURDLE: "Threshold for observations less than 50": 3%;

end userparams

declare threshold[security]: percent

declare change\_price\_dist[security]: distribution

on trade

declare let one\_minute = 1 minutes

declare let tap = SHORT\_TERM\_PERIOD \* one\_minute

declare let time\_tap = time - tap

if controlstatus(security) = "O" then

declare let comp\_price = trueprice(time\_tap)

if undefined(comp\_price) then

comp\_price = lastprice

end if

if defined(comp\_price) then

declare let price\_change = abs(change(price, comp\_price))

declare let dir\_str = format(price\_change > 0%,"increased","decreased")

if price\_change > threshold[security] then

alert 101, "STPM", "SHORT TERM PRICE MOVEMENT: [security] has

[dir\_str] in price from [comp\_price] to [price] in the last [SHORT\_TERM\_PERIOD]

minutes. This is an [dir\_str] of [price\_change], which

is greater than the threshold of [threshold[security]].",

intensity = (price\_change - threshold[security])/price\_change \* 100,

reissue = "101S+10"

end if

end if

end if

end on

Benchmarks\_below: BENPERIOD trdays

on trade

declare let one\_minute = 1 minutes

declare let tap = SHORT\_TERM\_PERIOD \* one\_minute

declare let time\_tap = time - tap

declare let comp\_price = trueprice(time\_tap)

if undefined(comp\_price) then

comp\_price = lastprice

end if

declare let price\_change = abs(change(price, comp\_price))

if defined(price\_change) then

change\_price\_dist[security] <- price\_change

end if

end on

at end

per security

if defined(change\_price\_dist[security]) then

declare let num\_objs = distcount(change\_price\_dist[security])

if num\_objs > 200 then

declare let mean = distaverage(change\_price\_dist[security])

declare let std = diststdev(change\_price\_dist[security])

threshold[security] = (mean + ST\_DEV\*std) \* 100%

elsif num\_objs > 50 then

threshold[security] = distcutoff(change\_price\_dist[security], 95%) \* 100%

else

threshold[security] = HURDLE

end if

end if

end per

end at

## 1.3

userparams

SHORT\_TERM\_PERIOD: "The number of minutes to look back": 15;

BENPERIOD: "The number of trading days for benchmarking": 30;

ST\_DEV: "The number of standard deviations away from distribution mean": 2;

HURDLE: "Threshold for observations less than 50": 3%;

end userparams

declare threshold[security]: percent

declare change\_price\_dist[security]: distribution

on trade

if flag(+ON) then

declare let one\_minute = 1 minutes

declare let tap = SHORT\_TERM\_PERIOD \* one\_minute

declare let time\_tap = time - tap

if controlstatus(security) = "O" then

declare let comp\_price = trueprice(time\_tap)

if undefined(comp\_price) then

comp\_price = lastprice

end if

if defined(comp\_price) then

declare let price\_change = abs(change(price, comp\_price))

declare let dir\_str = format(price\_change > 0%,"increased","decreased")

if price\_change > threshold[security] then

alert 101, "STPM", "SHORT TERM PRICE MOVEMENT: [security] has

[dir\_str] in price from [comp\_price] to [price] in the last [SHORT\_TERM\_PERIOD]

minutes. This is an [dir\_str] of [price\_change], which

is greater than the threshold of [threshold[security]].",

intensity = (price\_change - threshold[security])/price\_change \* 100,

reissue = "101S+10"

end if

end if

end if

end if

end on

Benchmarks\_below: BENPERIOD trdays

on trade

if flag(+ON) then

declare let one\_minute = 1 minutes

declare let tap = SHORT\_TERM\_PERIOD \* one\_minute

declare let time\_tap = time - tap

declare let comp\_price = trueprice(time\_tap)

if undefined(comp\_price) then

comp\_price = lastprice

end if

declare let price\_change = abs(change(price, comp\_price))

if defined(price\_change) then

change\_price\_dist[security] <- price\_change

end if

end if

end on

at end

per security

if defined(change\_price\_dist[security]) then

declare let num\_objs = distcount(change\_price\_dist[security])

if num\_objs > 200 then

declare let mean = distaverage(change\_price\_dist[security])

declare let std = diststdev(change\_price\_dist[security])

threshold[security] = (mean + ST\_DEV\*std) \* 100%

elsif num\_objs > 50 then

threshold[security] = distcutoff(change\_price\_dist[security], 95%) \* 100%

else

threshold[security] = HURDLE

end if

end if

end per

end at

# 2

at end

declare let old\_date = trday(date, -30)

per security

declare let avg\_count = tcount(date, old\_date) / 30

print "[security]'s past 30 days average trade-count is: [avg\_count]"

end per

end at