Capital Markets CRC Limited

MARKET DATA ANALYSIS USING ALICE

Creating Alerts

Capital Markets CRC Limited

Creating Alerts

Presented by
Tim Cooper
Director, SmartsGroup International
Co-founder of SmartsGroup,
Creator of Alice, and
Head of R&D for SmartsGroup

Lecture prepared by
Shan Ji & Will Renner

Capital Markets CRC Limited

Creating Alerts

In this lecture
- Creation of Alerts
- Three different approaches are covered

Time
- 35 Minutes

Requirements
- Session 6

The *alert* command

Syntax:

alert <code>, “<shorttext>”, “<fulltext>”

- <code> a number representing the alert code
- <shorttext> a brief description of the alert
- <fulltext> a description of the alert

```
declare Security_Traded[security] : boolean
on trade
        if undefined(Security_Traded[security]) then
                Security_Traded[security] = true
                alert 999, "TEST ALERT", "TEST ALERT: Security [security] has just had its first trade of the day"
        end if
end on
```

Example of an Alert

Exercise 1 – Run an Alert

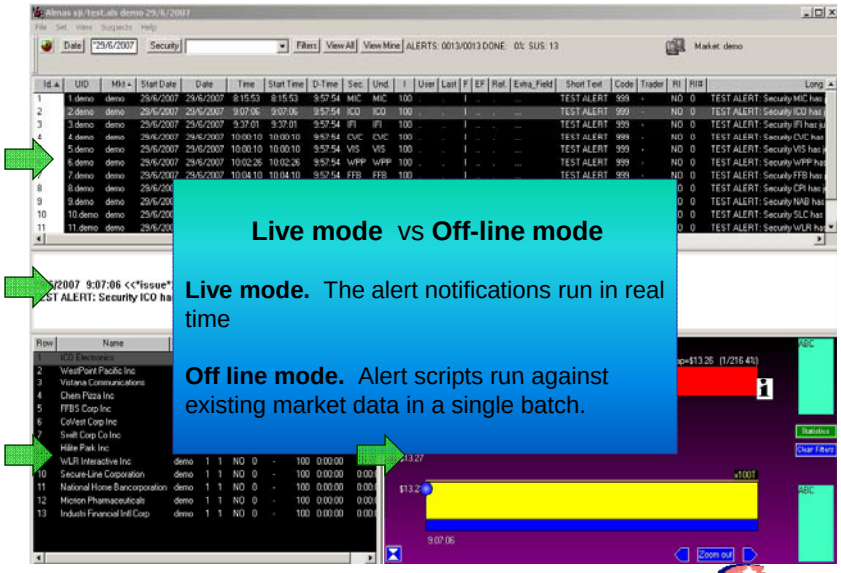
Calibration Run

```
declare Security_Traded[security] : boolean
on trade
        if undefined(Security_Traded[security]) then
                Security_Traded[security] = true
                alert 999, "TEST ALERT", "TEST ALERT: Security [security] has just had its first trade of the day"
        end if
end on
```

Run Options:

- Select the run options indicated
- Set one day for your Run Dates

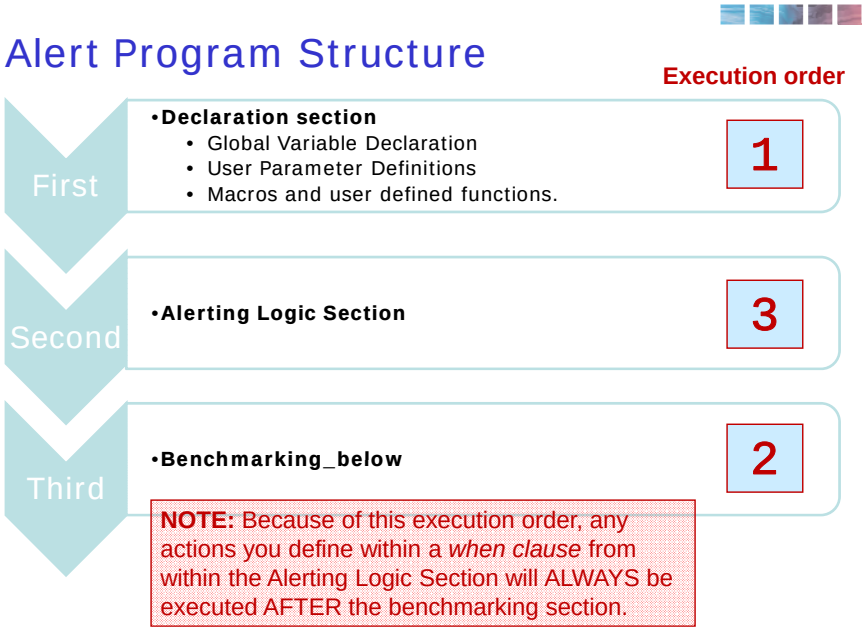
Output to ALMAS



Surveillance Alerts

- Detect Abnormal Market Behavior
- Consist of
  - Suspicious Market Behavior
  - Thresholds
  - When suspicious market behavior breaches thresholds, alert triggers
- First example
  - Trade to Trade Price Change Alert

Alert Program Structure



Alert Program Structure

```
userparams
        PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
        BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
        NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
end userparams

declare Trade2Trade_PChg_Distribution[security] : distribution
declare Trade2Trade_PChg_Threshold[security] : percent
```

Declaration Section

```
on trade
        declare let pchg = change(price, lastprice)
        declare let direction = format(pchg > 0%, "increase", "decrease")
        if abs(pchg) > Trade2Trade_PChg_Threshold[security] then
                alert 101, "T2T PCHG", "Trade to Trade Price Change Alert: Security [security] has just been
                                        traded at [price] while the previous trade price was [lastprice].
                                        This is [abs(pchg)] [direction] which exceeds the
                                        [Trade2Trade_PChg_Threshold[security]] threshold."
                intensity = (abs(pchg) - Trade2Trade_PChg_Threshold[security]) / abs(pchg) * 100,
                reissue = "101S+10"
        end if
end on
```

Alerting Logic Section

```
Benchmarks_below: BENCHMARK_PERIOD trdays

on trade
        declare let pchg = abs(change(price, lastprice))
        if defined(pchg) then
                Trade2Trade_PChg_Distribution[security] <- pchg
        end if
end on

at end
        per security
                if defined(Trade2Trade_PChg_Distribution[security]) then
                        declare let mean = distaverage(Trade2Trade_PChg_Distribution[security])
                        declare let stdev = diststdev(Trade2Trade_PChg_Distribution[security])
                        Trade2Trade_PChg_Threshold[security] = (mean + NUMBEROFSTDEV*stdev)*100%
                end if
        end per
end at
```

Benchmarking Section

Capital Markets CRC Limited

Trade to trade (T2T) price change alert

- 'T2T' Purpose
  - Trigger an alert when the current trade price is more than X% of the previous trade price
  - Trade 1: Price $56.87 Volume: x10000
  - Trade 2: Price $57.33 Volume: x5000
  - Trade 3: Price $57.65 Volume: x20000
  - Trade 4: Price $50.12 Volume: x50000

Capital Markets CRC Limited

Trade to trade (T2T) price change alert

- 'T2T' Purpose
  - Trigger an alert when the current trade price is more than X% of the previous trade price
  - Trade 1: Price $56.87 Volume: x10000
  - Trade 2: Price $57.33 Volume: x5000
  - Trade 3: Price $57.65 Volume: x20000
  - Trade 4: Price $50.12 Volume: x50000

Trade to trade (T2T) price change alert

- Purpose
  - Trigger an alert when the current trade price is more than X% of the previous trade price
- We need to think about:
  - How do we define X% (the *threshold*)?
  - Which 'when clause' will trigger the alert?
  - What will be the alert code?
  - What alert text will be displayed?
  - How often will the alert be triggered (the *intensity* of the alert)?

Trade to trade (T2T) price change alert

Two types of thresholds

- Static approach (eg User Defined Parameters)
- Historical benchmarking approaches

```
userparams
        PRICECHGTHRESHOLD: "The Price Change Threshold for Trade To Trade Alert" : 5%;
end userparams
```

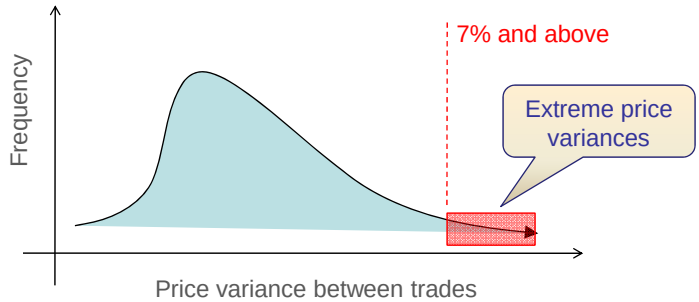
SYNTAX for USERPARAMS

[userparam name] : "[userparam description]" : userparam value

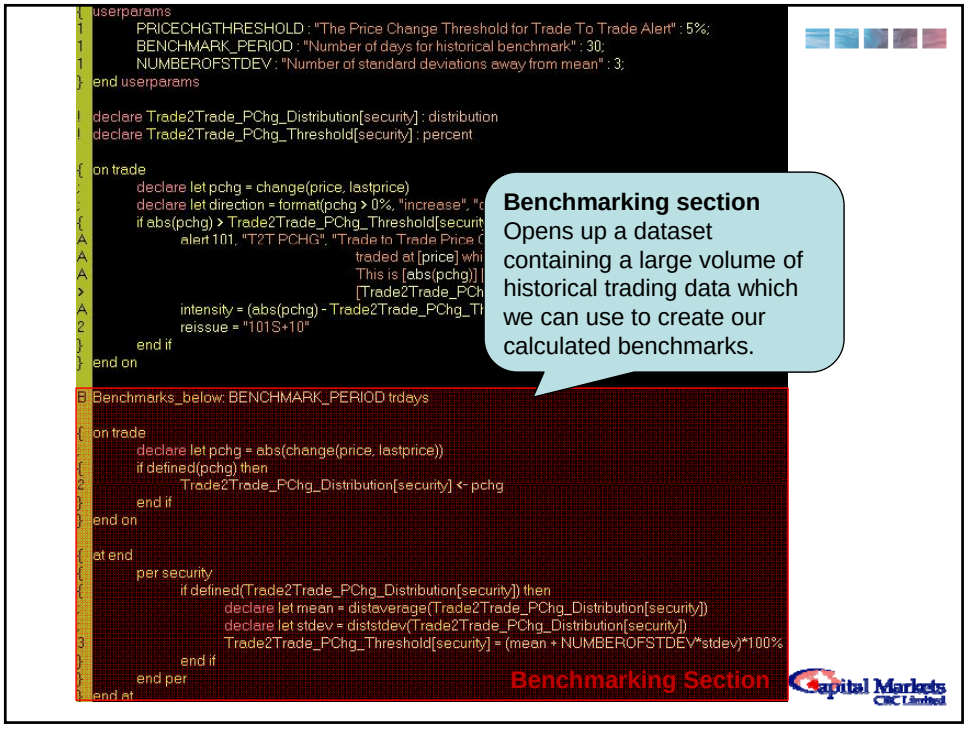
Trade to trade (T2T) price change alert

Two types of thresholds

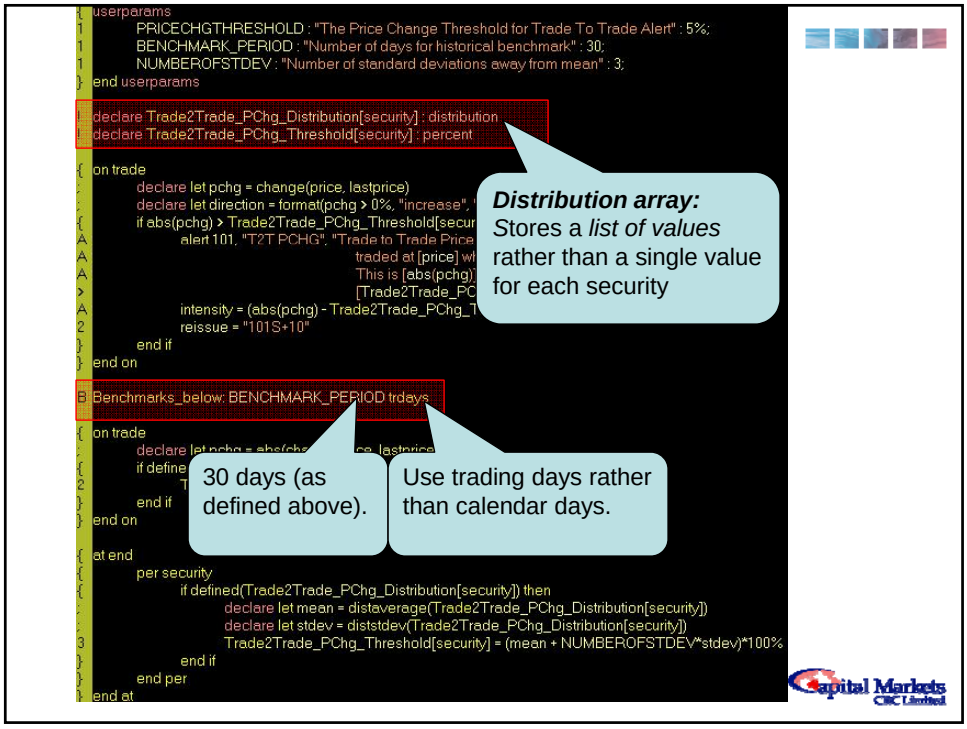
- Static approach (eg User Defined Parameters)
- Historical benchmarking approaches




userparams
PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
end userparams
declare Trade2Trade_PChg_Distribution[security] : distribution
declare Trade2Trade_PChg_Threshold[security] : percent
Declaration Section
on trade
declare let pchg = change(price, lastprice)
declare let direction = format(pchg > 0%, "increase", "decrease")
if abs(pchg) > Trade2Trade_PChg_Threshold[security] then
alert 101, "T2T PCHG", "Trade to Trade Price Change Alert: Security [security] has just been
traded at [price] while the previous trade price was [lastprice].
This is [abs(pchg)] [direction] which exceeds the
[Trade2Trade_PChg_Threshold[security]] threshold.",
intensity = (abs(pchg) - Trade2Trade_PChg_Threshold[security]) / abs(pchg) * 100,
reissue = "101S+10"
end if
end on
Benchmarks_below: BENCHMARK_PERIOD trdays
on trade
declare let pchg = abs(change(price, lastprice))
if defined(pchg) then
Trade2Trade_PChg_Distribution[security] <- pchg
end if
end on
at end
per security
if defined(Trade2Trade_PChg_Distribution[security]) then
declare let mean = distaverage(Trade2Trade_PChg_Distribution[security])
declare let stdev = diststdev(Trade2Trade_PChg_Distribution[security])
Trade2Trade_PChg_Threshold[security] = (mean + NUMBEROFSTDEV*stdev)*100%
end if
end per
end at
Benchmarking Section
Capital Markets CRC Limited


userparams
PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
end userparams
declare Trade2Trade_PChg_Distribution[security] : distribution
declare Trade2Trade_PChg_Threshold[security] : percent
on trade
declare let pchg = change(price, lastprice)
declare let direction = format(pchg > 0%, "increase", "
if abs(pchg) > Trade2Trade_PChg_Threshold[securit
alert 101, "T2T PCHG", "Trade to Trade Price
traded at [price] whi
This is [abs(pchg)]
[Trade2Trade_PCh
intensity = (abs(pchg) - Trade2Trade_PChg_Th
reissue = "101S+10"
end if
end on
Benchmarks_below: BENCHMARK_PERIOD trdays
on trade
declare let pchg = abs(change(price, lastprice))
if defined(pchg) then
Trade2Trade_PChg_Distribution[security] <- pchg
end if
end on
at end
per security
if defined(Trade2Trade_PChg_Distribution[security]) then
declare let mean = distaverage(Trade2Trade_PChg_Distribution[security])
declare let stdev = diststdev(Trade2Trade_PChg_Distribution[security])
Trade2Trade_PChg_Threshold[security] = (mean + NUMBEROFSTDEV*stdev)*100%
end if
end per
end at
Benchmarking Section
Benchmarking section
Opens up a dataset containing a large volume of historical trading data which we can use to create our calculated benchmarks.
Capital Markets CRC Limited

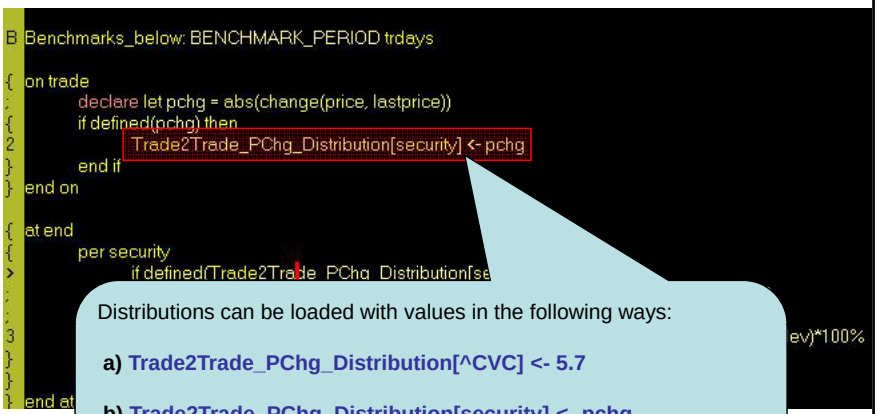

userparams
PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
end userparams
declare Trade2Trade_PChg_Distribution[security] : distribution
declare Trade2Trade_PChg_Threshold[security] : percent
on trade
declare let pchg = change(price, lastprice)
declare let direction = format(pchg > 0%, "increase", "decrease")
if abs(pchg) > Trade2Trade_PChg_Threshold[security] then
alert 101, "T2T PCHG", "Trade to Trade Price Change Alert: Security [security] has just been
traded at [price] while the previous trade price was [lastprice].
This is [abs(pchg)] [direction] which exceeds the
[Trade2Trade_PChg_Threshold[security]] threshold.",
intensity = (abs(pchg) - Trade2Trade_PChg_Threshold[security]) / abs(pchg) * 100,
reissue = "101S+10"
end if
end on
Benchmarks_below: BENCHMARK_PERIOD trdays
on trade
declare let pchg = abs(change(price, lastprice))
if defined(pchg) then
Trade2Trade_PChg_Distribution[security] <- pchg
end if
end on
at end
per security
if defined(Trade2Trade_PChg_Distribution[security]) then
declare let mean = distaverage(Trade2Trade_PChg_Distribution[security])
declare let stdev = diststdev(Trade2Trade_PChg_Distribution[security])
Trade2Trade_PChg_Threshold[security] = (mean + NUMBEROFSTDEV*stdev)*100%
end if
end per
end at
SLIDE 17
Capital Markets CRC Limited


userparams
PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
end userparams
declare Trade2Trade_PChg_Distribution[security] : distribution
declare Trade2Trade_PChg_Threshold[security] : percent
Distribution array:
Stores a list of values rather than a single value for each security
Benchmarks_below: BENCHMARK_PERIOD trdays
30 days (as defined above).
Use trading days rather than calendar days.
Capital Markets CRC Limited

```
{ userparams
1       PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
1       BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
1       NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
} end userparams

! declare Trade2Trade_PChg_Distribution[security] : distribution
! declare Trade2Trade_PChg_Threshold[security] : percent

{ on trade
:       declare let pchg = change(price, lastprice)
:       declare let direction = format(pchg > 0%, "increase", "decrease")
{       if abs(pchg) > Trade2Trade_PChg_Threshold[security] then
A           alert 101, "T2T PCHG", "Trade to Trade Price Change Alert: Security [security] has just been
A                                   traded at [price] while the previous trade price was [lastprice].
A                                   This is [abs(pchg)] [direction] which exceeds the
>                                   [Trade2Trade_PChg_Threshold[security]] threshold.",
A           intensity = (abs(pchg) - Trade2Trade_PChg_Threshold[security]) / abs(pchg) * 100,
2           reissue = "101S+10"
}       end if
} end on

B Benchmarks_below: BENCHMARK_PERIOD trdays

{ on trade
:       declare let pchg = abs(change(price, lastprice))
{       if defined(pchg) then
2           Trade2Trade_PChg_Distribution[security] <- pchg
}       end if
} end on

{ at end
{       per security
{           if defined(Trade2Trade_PChg_Distribution[security]) then
:               declare let mean = distaverage(Trade2Trade_PChg_Distribution[security])
:               declare let stdev = diststdev(Trade2Trade_PChg_Distribution[security])
3               Trade2Trade_PChg_Threshold[security] = (mean + NUMBEROFSTDEV*stdev)*100%
}           end if
}       end per
} end at
```



```
B Benchmarks_below: BENCHMARK_PERIOD trdays

{ on trade
:       declare let pchg = abs(change(price, lastprice))
{       if defined(pchg) then
2           Trade2Trade_PChg_Distribution[security] <- pchg
}       end if
} end on

{ at end
{       per security
>           if defined(Trade2Trade_PChg_Distribution[security]) then
:               declare let mean = distaverage(Trade2Trade_PChg_Distribution[security])
:               declare let stdev = diststdev(Trade2Trade_PChg_Distribution[security])
3               Trade2Trade_PChg_Threshold[security] = (mean + NUMBEROFSTDEV*stdev)*100%
}           end if
}       end per
} end at
```

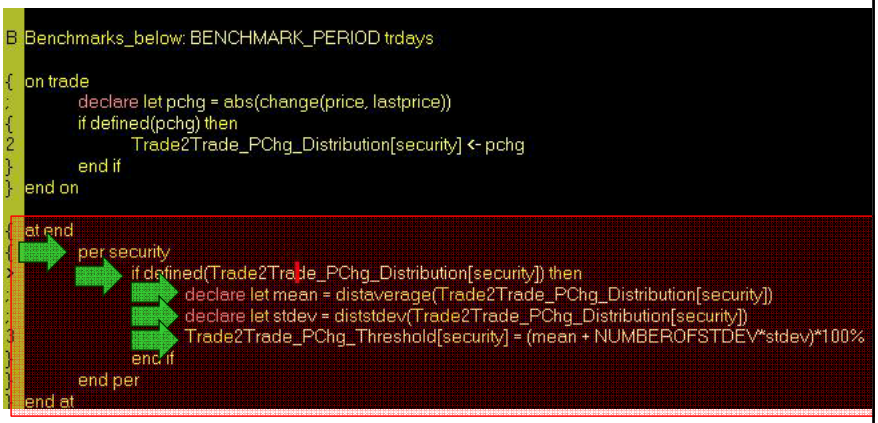


Distributions can be loaded with values in the following ways:

a) **Trade2Trade_PChg_Distribution[^CVC] <- 5.7**

b) **Trade2Trade_PChg_Distribution[security] <- pchg**

After loading up values, you end up with a distribution, such as: **3.25, 5.34, 6.72, 1.18, 2.88 …**



Trade to trade (T2T) price change alert

- Historical Benchmark
  - Common approach
    - Benchmarks_below: [number of days/trading days]
    - Distributions
  - Useful Functions
    - lastprice, change, abs
    - distaverage(distribution), diststdev(distribution)
  - Why convert all observations to positive values?
    - To capture extreme price changes regardless of whether they are positive or negative
  - Things to Note:
    - Any actions defined under the Benchmarks_below statement will only be executed for the benchmark period (i.e., the number of days/trading days defined for benchmarking)

Trade to trade (T2T) price change alert

Three approaches to T2T alerts:
One static approach, and
Two historical benchmarking approaches:

- **Standard deviation approach** (already discussed)
- **Distribution cutoff approach** (next)

New ALICE function used for the distribution cutoff approach:

- Distcutoff([distribution], [percentage cutoff])
- eg
  Distcutoff([Trade2Trade_PChg_Distribution[security]], 99%)

Exercise 2

- Please implement the Distribution Cutoff Benchmark Approach for the Trade To Trade Price Change Alert

```
userparams
        PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
        BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
        NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
end userparams

declare Trade2Trade_PChg_Distribution[security] : distribution
declare Trade2Trade_PChg_Threshold[security] : percent

Benchmarks_below: BENCHMARK_PERIOD trdays

on trade
        declare let pchg = abs(change(price, lastprice))
        if defined(pchg) then
                Trade2Trade_PChg_Distribution[security] <- pchg
        end if
end on

at end
        per security
                if defined(Trade2Trade_PChg_Distribution[security]) then
                        Trade2Trade_PChg_Threshold[security] =
                end if
        end per
end at
```

PAUSE THE SLIDESHOW HERE

Exercise 2

- Sample Solution

```
userparams
        PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
        BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
        NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
end userparams

declare Trade2Trade_PChg_Distribution[security] : distribution
declare Trade2Trade_PChg_Threshold[security] : percent

Benchmarks_below: BENCHMARK_PERIOD trdays

on trade
        declare let pchg = abs(change(price, lastprice))
        if defined(pchg) then
                Trade2Trade_PChg_Distribution[security] <- pchg
        end if
end on

at end
        per security
                if defined(Trade2Trade_PChg_Distribution[security]) then
                        Trade2Trade_PChg_Threshold[security] = distcutoff(Trade2Trade_PChg_Distribution[security], 99%)*100%
                end if
        end per
end at
```

Trade to trade (T2T) price change alert

Exercise 3

Within which type of *When Clause* will the alert need to be triggered?

PAUSE THE SLIDESHOW HERE

Trade to trade (T2T) price change alert

- Solution for exercise 3
  - When clause to be used: **On trade**
- Other details:
  - Alert code: assume 101
  - Short Alert Text: T2T PCHG
  - Long Alert Text:

    Trade to Trade Price Change Alert: Security [security] has just been traded at [price] while the previous trade price was [lastprice]. This is [pchg%] [increase/decrease] which exceeds the [threshold%] threshold.

Exercise 4

- Giving the following details
  - Alert code: assume 101
  - Short Alert Text: T2T PCHG
  - Long Alert Text:
    Trade to Trade Price Change Alert: Security [security] has just been traded at [price] while the previous trade price was [lastprice]. This is [pchg%] [increase/decrease] which exceeds the [threshold%] threshold.
- Please complete the Trade to Trade Price Change Alert by using the Static User Parameter PRICECHGTHRESHOLD as the threshold.

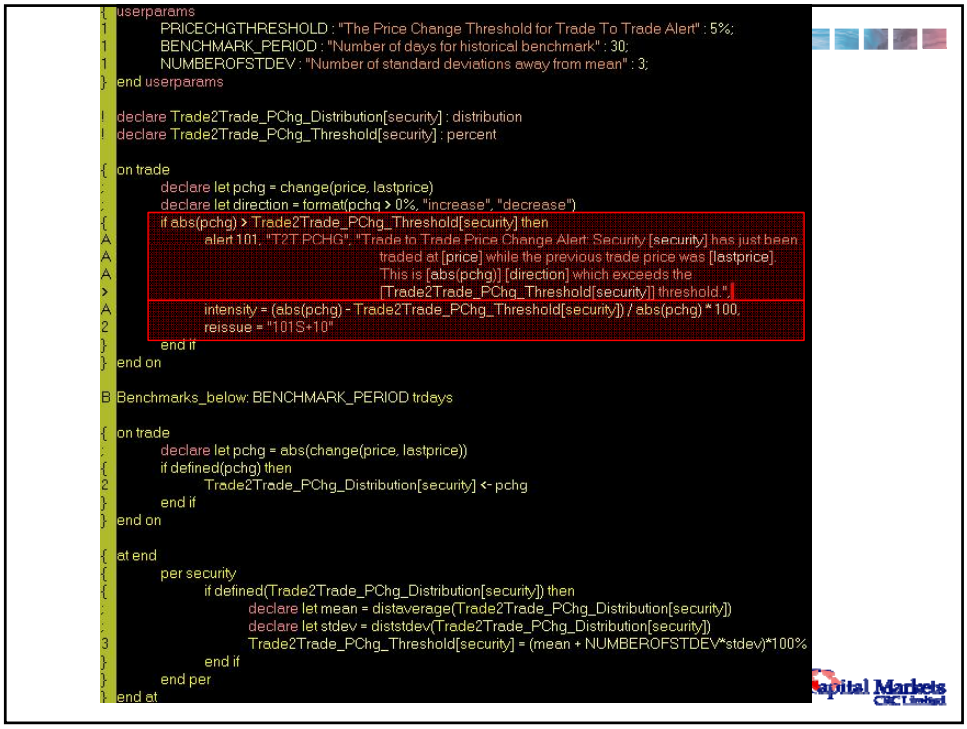
```
{ userparams
1       PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
```

PAUSE THE SLIDESHOW HERE

Exercise 4 - Sample solution for Static Approach

```
{ userparams
1       PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
1       BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
1       NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
} end userparams

{ on trade
;       declare let pchg = change(price, lastprice)
;       declare let direction = format(pchg > 0%, "increase", "decrease")
{       if abs(pchg) > PRICECHGTHRESHOLD then
A               alert 101, "T2T PCHG", "Trade to Trade Price Change Alert: Security [security] has
A                                       just been traded at [price] while the previous trade price
A                                       was [lastprice]. This is [abs(pchg)] [direction] which
2                                       exceeds the [PRICECHGTHRESHOLD] threshold."
}       end if
} end on
```

- Useful function
  format(condition, [str1], [str2])



```
userparams
        PRICECHGTHRESHOLD : "The Price Change Threshold for Trade To Trade Alert" : 5%;
        BENCHMARK_PERIOD : "Number of days for historical benchmark" : 30;
        NUMBEROFSTDEV : "Number of standard deviations away from mean" : 3;
end userparams

declare Trade2Trade_PChg_Distribution[security] : distribution
declare Trade2Trade_PChg_Threshold[security] : percent

on trade
        declare let pchg = change(price, lastprice)
        declare let direction = format(pchg > 0%, "increase", "decrease")
        if abs(pchg) > Trade2Trade_PChg_Threshold[security] then
                alert 101, "T2T PCHG", "Trade to Trade Price Change Alert: Security [security] has just been
                                        traded at [price] while the previous trade price was [lastprice].
                                        This is [abs(pchg)] [direction] which exceeds the
                                        [Trade2Trade_PChg_Threshold[security]] threshold."
                intensity = (abs(pchg) - Trade2Trade_PChg_Threshold[security]) / abs(pchg) * 100,
                reissue = "101S+10"
        end if
end on

Benchmarks_below: BENCHMARK_PERIOD trdays

on trade
        declare let pchg = abs(change(price, lastprice))
        if defined(pchg) then
                Trade2Trade_PChg_Distribution[security] <- pchg
        end if
end on

at end
        per security
                if defined(Trade2Trade_PChg_Distribution[security]) then
                        declare let mean = distaverage(Trade2Trade_PChg_Distribution[security])
                        declare let stdev = diststdev(Trade2Trade_PChg_Distribution[security])
                        Trade2Trade_PChg_Threshold[security] = (mean + NUMBEROFSTDEV*stdev)*100%
                end if
        end per
end at
```

Intensity & reissue

How often should an alert be triggered?

For example:

- An alert was triggered for security CVC at 9:00 am this morning when the security had a 10% T2T price change.
- When the security has another T2T price change of 10%, should another alert be triggered?
- When should the alert be retriggered, and how many times?

Intensity

"The intensity of an alert is a number, ranging from 0 to 100, which estimates the *significance* of an alert"

Syntax

intensity = <a number from 0 to 100>

Things to remember

- The intensity (normally a formula) results in a value between 0 and 100
- Intensity is calculated on each potential alert incidence
- A *high value* rates the Alert as one of high significance

Intensity

```
on trade
    declare let pchg = change(price, lastprice)
    declare let direction = format(pchg > 0%, "increase", "decrease")
    if abs(pchg) > Trade2Trade_PChg_Threshold[security] then
        alert 101, "T2T PCHG", "Trade to Trade Price Change Alert: Security [security] has just been
                                traded at [price] while the previous trade price was [lastprice].
                                This is [abs(pchg)] [direction] which exceeds the
                                [Trade2Trade_PChg_Threshold[security]] threshold."
        intensity = (abs(pchg) - Trade2Trade_PChg_Threshold[security]) / abs(pchg) * 100
        reissue = "101S+10"
    end if
end on
```

Reissue

- Suppresses redundant alerts
- Uses *intensity* to decide if subsequent alters need reissuing after the first instance of an alert
- By default: the reissue mechanism looks at any previous alert of the same alert code on the same security, and only reissues an alert if the intensity has increased by 10 or more.
- The example reissue rule(below) of “101S+10” does exactly this, but can be left out.

```
{AAA>A2}
    if abs(pchg) > Trade2Trade_PChg_Threshold[security] then
        alert 101, "T2T PCHG", "Trade to Trade Price Change Alert: Security [security] has just
                                traded at [price] while the previous trade price was [lastpric
                                This is [abs(pchg)] [direction] which exceeds the
                                [Trade2Trade_PChg_Threshold[security]] threshold.",
        intensity = (abs(pchg) - Trade2Trade_PChg_Threshold[security]) / abs(pchg) * 100,
        reissue = "101S+10"
    end if
```

Key terms and concepts

- Alerts
- Userparams
- Historical Benchmarks
- Standard Deviation Approach
- Distribution Cutoff Approach
- Reissue
- Intensity

Help is available

- Review this lecture
- Consult wikipedia, Alice Reference manual
- Post a question to the class forum