9 Feb 2018

TOPIC 9 COURSE PROJECT

INTRODUCTION

The Course Project consists of two parts.

Part One requires you to create an Alice alert based on the specifications provided in Part1. ALERTS below. Please choose **ONLY ONE** Alert.

Part Two is to create an orderbook database for academic research using Alice. Students should use both the *InsiderTrading* and the *DemoMarket* databases to test their Alice scripts.

PREPARATION

1. Create a new Microsoft Word document (use WordPad if you do not have MS Word).

2. On page one, create a cover page with the following information:

Your name:	eg John Smith
Tasks attempted:	the activities you completed/attempted

3. Save your document to your desktop with the filename *YourName.doc* (use your own name). The output of the activities below will need to be pasted into this document, as described below.

INSTRUCTIONS FOR SUBMISSION AND MARK ALLOCATION

The following activities require you to copy and paste Alice source code and spreadsheet data into the Word document, which you created above.

Please label each section of output clearly in your submission.

Part One (7 marks)

Please submit source code only. Please copy and paste the *text* from your completed program source code into your word document.

Part Two (7 marks)

Please submit both the Alice source code and the input/output csv files. To do this, please copy and paste the *text* from your completed program source code into your word document. The *text* of your csv files should be copied and pasted into your word document as formatted text and *not* as an embedded excel file.

PART 1: ALERTS

1.1. Closing Price Change Alert

1.1.1. Purpose

This alert identifies unusual price changes at the end of the trading session

1.1.2. Design

Event types	TRADES
Alert Code	101
Short Text	CPC
Long Text	CLOSING PRICE [increase/decrease]: [security] has [increased/decreased] [price change] percent from [previous price] to [current price] in the last [X] minutes of trading.
Parameters	BENCHMARK_PERIOD 30 trading days CLOSE_PRICE_LOOKBACK_PERIOD 15 minutes DISTRIBUTION_CUTOFF 95%
Algorithm	Event 1: At 16:01 Rule 1: If the security has closed for trading (e.g., controlstatus = "C") Rule 2: if the change in price between the last trade price and the true price CLOSE_PRICE_LOOKBACK_PERIOD minutes prior to the close is greater than the threshold then Issue Alert 101
Benchmark	(1) For each security, create historical distribution of the change between close price and price [CLOSE_PRICE_LOOKBACK_PERIOD] minute before closing for the past [BENCHMARK_PERIOD] trading days (2) For each security, take the [DISTRIBUTION_CUTOFF] distribution cutoff from that security's distribution created in (1). (3) For each security, use the distribution cutoff calculated in (2) as the threshold
Intensity	(Price Change – Benchmark) / Price Change *100
Reissue	101S+100
Definitions	*price change* – (Price 2 – Price 1) / Price 1

1.2. Abnormal Volume Alert

1.2.1. Purpose

This alert identifies securities that have had a successive run of high volume during a trading day which may indicate the need for information release from the issuer of the security.

1.2.2 Design

Event types	Trades
Alert Code	102
Short Text	LTV
Long Text	ABNORMAL VOLUME: [security] has traded [volume] so far today. This is more than the threshold of [Threshold]
Parameters	**User parameters** BENCHMARK_PERIOD 30 trading days
Algorithm	Event 1: On every trade Rule 1: if the volume traded so far on that day is greater than the threshold then Issue Alert 102
Benchmark	**Benchmark Collection** (1) For each security, calculate the average daily volume traded over the past BENCHMARK_PERIOD trading days (2) For each security, use the average daily volume calculated in (1) as the threshold
Intensity	(Volume traded so far today – Threshold) / Volume traded so far today *100
Reissue	102S+10
Definitions	

1.3. Abnormally Large Trade Alert

1.3.1. Purpose

Identify unusually large trades.

1.3.2. Design

Event types	Trades
Alert Code	103
Short Text	ALT
Long Text	ABNORMAL LARGE TRADE: [security] just had an abnormally large trade. The volume of the trade is [volume] that is more than the threshold of [Threshold]
Parameters	**User parameters** BENCHMARK_PERIOD 30 trading days DISTRIBUTION_CUTOFF 95%
Algorithm	Event 1: On every trade Rule 1: if the volume of that trade is greater than the threshold, Issue alert 103.
Benchmark	**Benchmark Collection** (1) For each security, create historical distribution of volume of each trade over the past BENCHMARK_PERIOD trading days. (2) For each security, take the [DISTRIBUTION_CUTOFF] distribution cutoff from that security's distribution created in (1). (3) For each security, use the distribution cutoff calculated in (2) as the threshold
Intensity	(trade's volume – Threshold) / trade's volume *100
Reissue	103S+10

1.4. Abnormal Order Rate

1.4.1. Purpose

Identify larger than usual order flow, which may indicate upcoming supply/demand shifts in a security or an erroneous order entry.

1.4.2. Design

Event types	Orders
Alert Code	104
Short Text	AOR
Long Text	ABNORMAL ORDER RATE: [security] has received [number] of orders so far today. This is greater than the threshold for this security of [threshold].
Parameters	**User Parameters** BENCHMARK_PERIOD 30 trading days DISTRIBUTION_CUTOFF 95%
Algorithm	Event 1: **F_ENTER** • Increase the number of orders entered for that security by one Rule 1: If the total number of orders entered for that security is greater than the threshold Issue Alert 104
Benchmark	(1) Track the number of orders entered for each security during each of the trading day from the past BENCHMARK_PERIOD trading days. (2) At the end of the benchmarking, calculate the average number of orders entered per day over the past BENCHMARK_PERIOD trading days for each security. (3) Set the average number of orders entered for each security calculated in (2) as the threshold
Intensity	(Number of Orders-threshold)/Number of Orders *100
Reissue	104S+20
Definitions	N/A

PART 2: Market Data Analysis using ALICE for Research

2.1 Introduction

Please read the following research paper:

> Michael Aitken & Audris Siow (2004) *Ranking world equity markets on the basis of market efficiency and integrity*

In this paper Aitken & Siow (2004) discuss a methodology to measure market integrity and market efficiency. Here we are going to focus on the market efficiency part.

2.2 Description of Task

Aitken & Siow (2004) used Time Weighted Relative Spreads to proxy for the efficiency of a stock market (i.e., costs of dealing in each market).

To calculate the relative spread for each security, the following formula is used:
Relative spread = ((best ask - best bid) / ((best ask + best bid)/2)) at each change in spread.

The time weight was calculated by taking the time that each spread existed during a trading day. A summation of the changes in spreads multiplied by the time it was available is created for each security for each trading day using the following:
Σ (time weights) x (relative spread)
where :
time weight = the amount of time the spread was in existence / total time during the day

Please complete ***BOTH*** of the following tasks:

Task 1: Please code an ALICE program, which can calculate the Daily Time Weighted Relative Spreads for each security over a range of dates and export the results to csv files. The required columns are Security, Date, and Time Weighted Relative Spreads. The ALICE script for Part Two should be able to be run over multiple trading days. Please use the *asx_fob* market to test your program.

Task 2: Please code an ALICE program that can read in the csv files created in Task 1 and calculate the arithmetic average Time Weighted Relative Spreads across all securities for each trading day. The results will also need to be exported to a csv file. Required columns are Market Code (e.g., Demo), Date, Average Time Weighted Relative Spreads across all securities.
Hint: Search for *marketcode* in the *Search Alice Function* tool within ALDIT tools menu. The *marketcode* function returns the *shortname* of the market you are working on.

MARKING GUIDE

Part One: Alert coding: 15 Marks total

(1) Declarations: 3 marks
(2) Benchmarking: 5 marks
(3) Algorithm: 5 marks
(4) Intensity: 1 mark
(5) Reissue: 1 mark

Part Two: Report coding: 15 Marks total

(1) Task 1: 8 marks total
   - Time Weighted Relative Spread: 4 marks
   - Feasibility for running across a period: 2 marks
   - Creation of csv files: 1 marks
   - Submission of csv files: 1 mark

(2) Task 2: 7 marks total
   - Reading csv: 1 mark
   - Securities count: 1 mark
   - Average Time Weighted Relative Spread: 3 marks
   - Creation of csv files: 1 mark
   - Submission of csv files: 1 mark