Описание советника

Советник работает в торговые дни (кроме выходных). В эти дни каждые 5 минут производит запись в файл(ы), полученных/расчетных данных, а также проводит нормирование, т.е. следит за корректностью самих данных до записи, при необходимости корректирует их, и после производит запись.

Особенностью советника является возможность нормирования N количества строк с конца. То есть при инициализации/работе он должен пробежаться по всем/N строкам в файле и скорректировать данные, или добавить пропущенные строки.

Условия корректности:

1. Следующая дата + 5 минут от предыдущей (исключения выходной)
2. Данные больше X, и меньше Y.
3. Данные не должны отличаться больше чем в Z раз, где Z коэф из параметров

В итоге мы должны получить файл/файлы с записями данных соответствующими каждой пятиминутке (М15…) работы советника.

Проблемы. Может возникнуть ситуация, что данные будут получаться долго, т.е более 5 минут, это не должно сказаться на работе советника. При длительном (>1 часа) пропуске и или не получении данных выслать письмо.

Параметры

1. Н/Нр - На какое количество строк проверять файл при инициализации/работе.
2. Х и У диапазон корректных данных.(возможно придется сделать для каждого столбца данных свою пару Х/У)
3. Z коэффициент который определяет допустимое изменение новых данных от предыдущих. Например изменение больше чем в 2 раза будет вылет. 0-не использовать.
4. MaxMin параметр относительно которого брать мин или мах для записи в файл др ТФ. .(возможно придется сделать для каждого столбца данных свой)
5. Путь сохранения файла.

Алгоритм

Инициализация: Если задано проверить Н строк в файле, то советник должен проверить Н строк с конца на корректность, в случае несоответствия дописать не достающие строки либо изменить данные.

Основной цикл: раз в 5 минут советник запускает процедуру получения данных (завязывать на появлении свечи не корректно, тк бывает пропуск котировок),далее советник рассчитывает данные (это мой код). в итоге получается следующая строка:

Дата (год.месяц....);Время;Данные1;Данные2;…

Пример: 2016.11.17;22:45:06;27;44;94;16;

После получения данных проводится проверка корректности.

1. Текущие данные это + 5 минут от предыдущих (исключения выходной)
2. Данные больше X, и меньше Y, где Х/У из параметров.
3. Данные не должны отличаться больше чем в Z раз, где Z коэф из параметров

Если текущие данные не являются следующими, то значит, у нас в файле есть пропуск дат или по какой-то причине не были записаны данные и его надо восполнить, а после записать текущие данные, при необходимости скорректированные.

Недостающие/пропущенные строки вставляются с данными последней корректной строки. То есть мы копируем последнюю строку с данными до текущей даты, меняя только дату (+5 минут). Получается у нас как бы данные не меняются.

Если не выполняется условие 2, то мы также для записи в текущей дате используем данные из предыдущей строки.

Мы должны обеспечить не прерывный поток дат с данными.

**Пример**

Имеем сейчас такую ситуацию (очень близкую к фактической, но малость для примера правленую)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 21.11.2016 | 10:00:42 | 77 | 26 | 27 | 10 |
| 21.11.2016 | 10:15:42 | 78 | 26 | 27 | 10 |
| 21.11.2016 | 10:20:41 | 1 | 1 | 3 | 3 |
| 21.11.2016 | 10:25:42 | 78 | 27 | 27 | 10 |
| 21.11.2016 | 10:35:43 | 78 | 27 | 28 | 11 |
| 21.11.2016 | 10:40:42 | 79 | 27 | 30 | 11 |
| 21.11.2016 | 10:45:42 | 41 | 12 | 8 | 11 |
| 21.11.2016 | 10:50:43 | 12 | 8 | 4 | 7 |
| 21.11.2016 | 11:00:42 | 13 | 8 | 4 | 7 |
| 21.11.2016 | 11:05:43 | 78 | 24 | 28 | 11 |
| 21.11.2016 | 11:10:43 | 78 | 28 | 28 | 11 |
|  |  |  |  |  |  |

Здесь мы видим пропуски дат и «вылеты»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 21.11.2016 | 10:00:42 | 77 | 26 | 27 | 10 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 21.11.2016 | 10:15:42 | 78 | 26 | 27 | 10 |
| 21.11.2016 | 10:20:41 | 1 | 1 | 3 | 3 |
| 21.11.2016 | 10:25:42 | 78 | 27 | 27 | 10 |
|  |  |  |  |  |  |
| 21.11.2016 | 10:35:43 | 78 | 27 | 28 | 11 |
| 21.11.2016 | 10:40:42 | 79 | 27 | 30 | 11 |
| 21.11.2016 | 10:45:42 | 41 | 12 | 8 | 11 |
| 21.11.2016 | 10:50:43 | 12 | 8 | 4 | 7 |
|  |  |  |  |  |  |
| 21.11.2016 | 11:00:42 | 13 | 8 | 4 | 7 |
| 21.11.2016 | 11:05:43 | 78 | 24 | 28 | 11 |
| 21.11.2016 | 11:10:43 | 78 | 28 | 28 | 11 |

Вот от этих пропусков и вылетов надо избавиться в текущих файлах, и не допустить появление в будущем. После корректировки по условиям должны получить следующее.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 21.11.2016 | 10:00:42 | 77 | 26 | 27 | 10 |
| 21.11.2016 | 10:05:42 | 77 | 26 | 27 | 10 |
| 21.11.2016 | 10:10:42 | 77 | 26 | 27 | 10 |
| 21.11.2016 | 10:15:42 | 78 | 26 | 27 | 10 |
| 21.11.2016 | 10:20:41 | 78 | 26 | 27 | 10 |
| 21.11.2016 | 10:25:42 | 78 | 27 | 27 | 10 |
| 21.11.2016 | 10:30:41 | 78 | 27 | 28 | 10 |
| 21.11.2016 | 10:35:43 | 78 | 27 | 28 | 11 |
| 21.11.2016 | 10:40:42 | 79 | 27 | 30 | 11 |
| 21.11.2016 | 10:45:42 | 79 | 27 | 30 | 11 |
| 21.11.2016 | 10:50:43 | 79 | 27 | 30 | 11 |
| 21.11.2016 | 10:55:42 | 79 | 27 | 30 | 11 |
| 21.11.2016 | 11:00:42 | 79 | 27 | 30 | 11 |
| 21.11.2016 | 11:05:43 | 78 | 24 | 28 | 11 |
| 21.11.2016 | 11:10:43 | 78 | 28 | 28 | 11 |

А вот так должны выглядеть файлы м15 м30 Н1 Н4

Для других ТФ создаем свои файлы. Данные необходимо записывать так, максимальное значение если больше MaxMin и минимальное если данные меньше MaxMin. Запись новой строки каждые 15/30 и тд мин. А обновление каждые 5 минут если требуется, то есть если MaxMin = 50, данные уже записаны = 77, а новые = 76, тогда мы ничего не переписываем, а если новые 80, то следует переписать 77 на 80. Как бы получаются динамичные файлы ТФ.

Пример. Пусть MaxMin = 50. Тогда у нас выходят следующие файлы.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 21.11.2016 | 10:00:42 | 77 | 26 | 27 | 10 |
| 21.11.2016 | 10:15:42 | 78 | 26 | 27 | 10 |
| 21.11.2016 | 10:30:41 | 78 | 27 | 28 | 10 |
| 21.11.2016 | 10:45:42 | 79 | 27 | 30 | 11 |
| 21.11.2016 | 11:00:42 | 79 | 24 | 28 | 11 |
|  |  |  |  |  |  |
| 21.11.2016 | 10:00:42 | 78 | 26 | 27 | 10 |
| 21.11.2016 | 10:30:41 | 79 | 27 | 28 | 10 |
| 21.11.2016 | 11:00:42 | 79 | 24 | 28 | 11 |
|  |  |  |  |  |  |
| 21.11.2016 | 10:00:42 | 79 | 26 | 27 | 10 |
| 21.11.2016 | 11:00:42 | 79 | 24 | 28 | 11 |

Пропуски появляются когда слабый инет и сильно загружен ВПС и советник не успевал все обработать за 5 мин….то есть он еще не записал предыдущее, а пора новое обрабатывать, и получается своеобразный на хлёст и «новое» вылетает. А вылеты в 90% появляются при сбое во время записи в файл. А вылеты в 90% появляются при сбое во время записи в файл. Но так как данные не особо динамичны, то простое заполнение пропусков и вылетов не сильно влияет на данные. Еще есть проблема это запуск расчета данных уплывает…если сначала он был каждые 5 мин в 0-5 сек, то через несколько дней уходит на минуту и более, а бывает и не уходит далеко. Но это все касается моего кода.