时间序列数据的预处理

该部分主要对时间序列数据进行预处理，其中主要包括两部分：

对原始数据进行断点查找（breakpoint searching）,即捕捉原始序列的关键走势点，起到对高频序列降频的目的。主要采用基于滑动窗口的**交替趋势平滑法**（alternating trends smoothing, ATS）进行提取。

对数据进行标准化。其标准化的公式为：

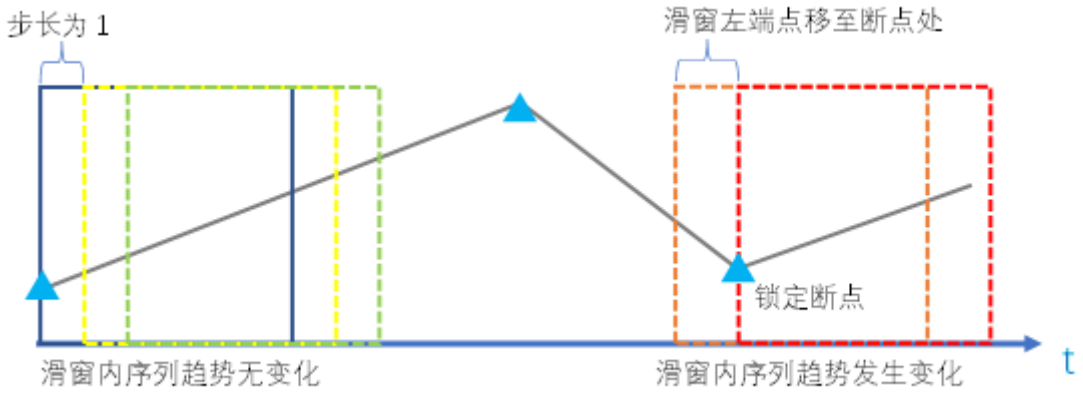
$$z_i = \frac{x_i - \mu}{\sigma},\ i = 1, \ldots n$$

其中μ表示时间序列的均值，σ表示时间序列的方差。以下将主要介绍ATS方法的原理。

一、交替趋势平滑法

断点序列本文采用基于滑动窗口的**交替趋势平滑法**（alternating trends smoothing, ATS）进行提取。该方法的原理为给定一个固定窗宽为h的滑窗，在时序时间轴上以步长为1进行滑动，遍历原始时间序列。

每经过一个步长，对新的步长内的趋势进行检验，如增长或下降，与上一个滑窗内的趋势是否相同，若相同则考察下一个步长；若不同，则在当前步长中寻找趋势发生变化的点，定位变点的位置。



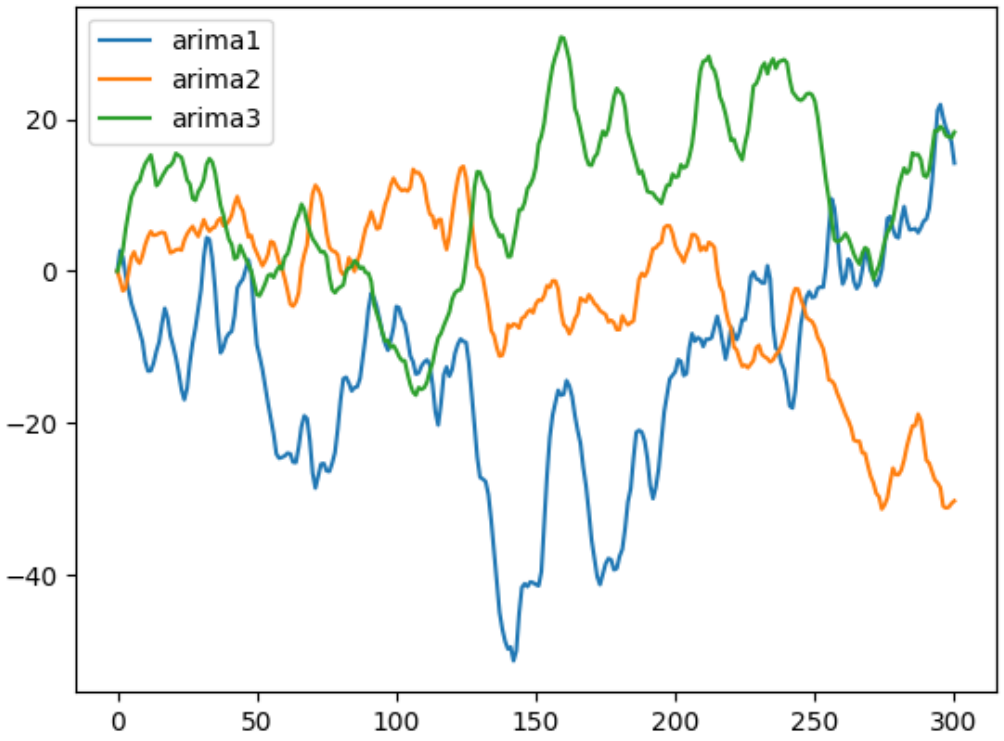
该算法的伪代码如下所示：

> 输入：时间序列$q = \{q_1, q_2, \cdots, q_m\}$，窗宽为h
>
> 输出：变点序列CP
>
> 1. 取变点下标d=1；
> 2. 取第一个变点的横坐标为$b_d = 1$，纵坐标为$v_d = q_1$；
> 3. CP=$\{(b_d, v_d)\}$；
> 4. 取j=0，r=0；

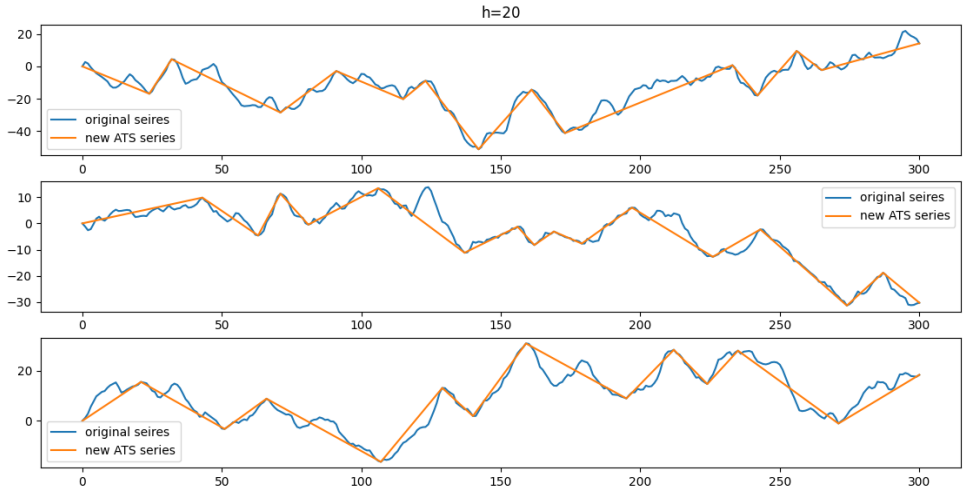
5. 当j+h≤m并且r=0时，执行
6. 取子列$\{q_1,q_2,\cdots,q_{j+h}\}$，计算首尾两点间的斜率s，并取r=sign(s)；
7. 取j=j+1；
8. 结束
9. 取d=d+1；
10. 当j+h≤m时，执行
11. 取子列$\{q_1,q_2,\cdots,q_{j+h}\}$，计算首尾两点间的斜率，更新s；
12. 如果sign(s)=r则
13. 取j=j+1；
14. 否则
15. 取$\{rq_{j+1},rq_{j+2},\cdots,rq_{j+h}\}$中的最大值，计其下标为$j^+$，并令$b_d=j^+$，$v_d=q_{j^+}$；
16. 将(b_d,v_d)加入变点序列CP；
17. $j=j^+$；
18. d=d+1；
19. r=sign(s)；
20. 结束
21. 结束

二、断点查找模拟检验

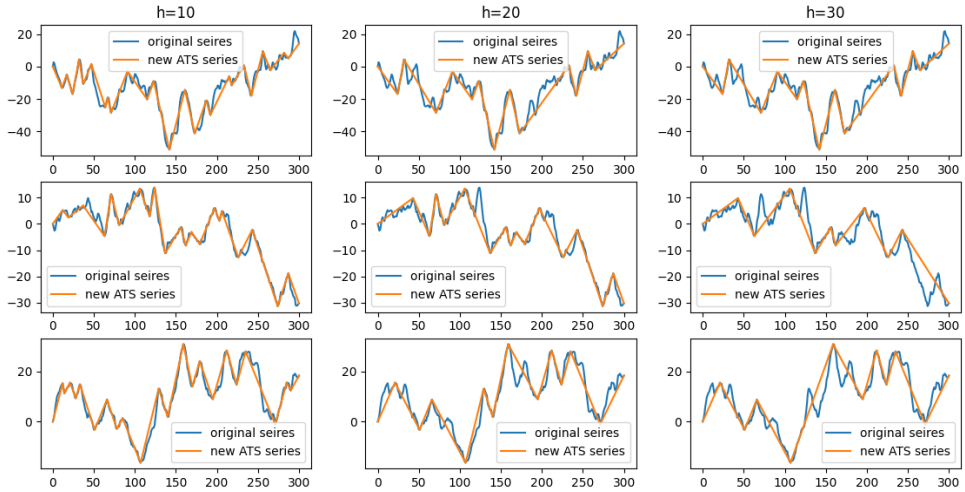
首先通过R语言生成三条长度为300的随机arima序列，序列图如下所示：



选择窗宽长度h为20，分别对三条序列进行断点查找，得到的结果如下图所示：



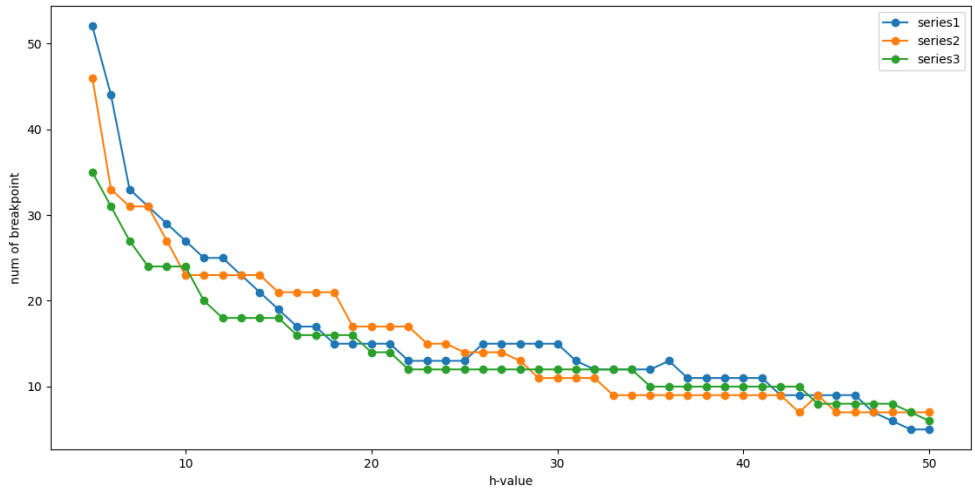
此外，还分别选择窗宽h为10，20，30，分别对三条序列进行断点查找，得到的结果如下图所示：



从图中可以看出，基于滑动窗口的ATS算法能够较好的拟合时间序列的主要趋势特征。

三、观察断点数随窗宽变化的变化情况

遍历窗宽大小h为[5,50]间的全部自然数，观察变点数量的变化趋势：



从图中可以看出，当窗宽大于20时，断点个数逐渐趋于平稳。因此后续分析可选择20作为窗宽。或者后续我们可以考虑改变时间序列的长度，探究窗宽与时序长度比例的变化。