

发布时间: 01-16 14:51

由于系统编辑器限制，代码行用无序列表表示。

滑动窗口示意图

整合使用Nacos的配置管理功

	则数值表示计算统计量的观测值的数量即向前几个数据。如果是offset类型，表示时间窗的大小
min_periods	每个窗口最少包含的观测值数量，小于这个值的窗口结果为NA。值可以是int，默认None。offset情况下，默认为1。
center	把窗口的标签设置为居中。布尔型，默认False，居右
win_type	窗口的类型。截取窗的各种函数。字符串类型，默认为None。
on	可选参数。对于dataframe而言，指定要计算滚动窗口的列。值为列名
axis	int、字符串，默认为0，即对列进行计算
closed	定义区间的开闭，支持int类型的window。对于offset类型默认是左开右闭的即默认为right。可以根据情况指定为left、both等。

rolling参数详解

用法代码演示

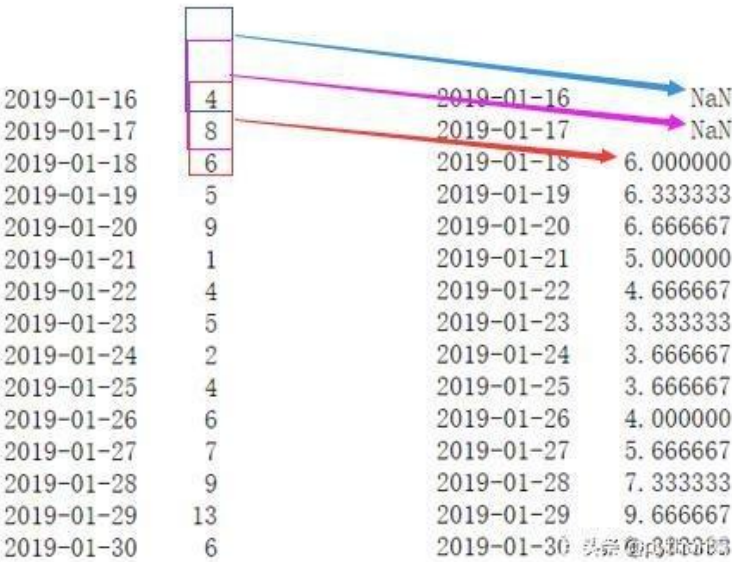
上面我们介绍了滑动窗口的概念及实现函数的参数，下面我们通过代码演示，依次展示各参数的作用。

- import matplotlib.pyplot as plt
- import numpy as np
- import pandas as pd
- index=pd.date_range('20190116','20190130')
- data=[4,8,6,5,9,1,4,5,2,4,6,7,9,13,6]
- ser_data=pd.Series(data,index=index)

ser_data	
2019-01-16	5
2019-01-17	7
2019-01-18	6
2019-01-19	5
2019-01-20	9
2019-01-21	1
2019-01-22	4
2019-01-23	5
2019-01-24	4
2019-01-25	4
2019-01-26	6
2019-01-27	7
2019-01-28	9
2019-01-29	13
2019-01-30	6

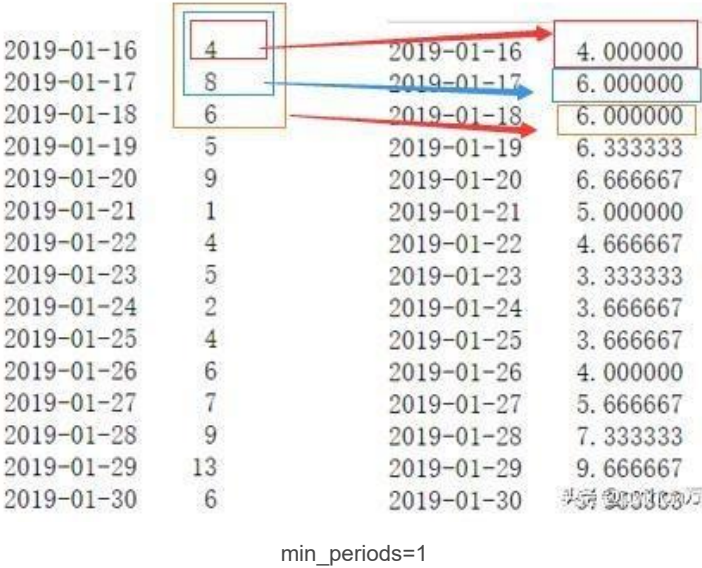
数据预览

- ser_data.rolling(3).mean()



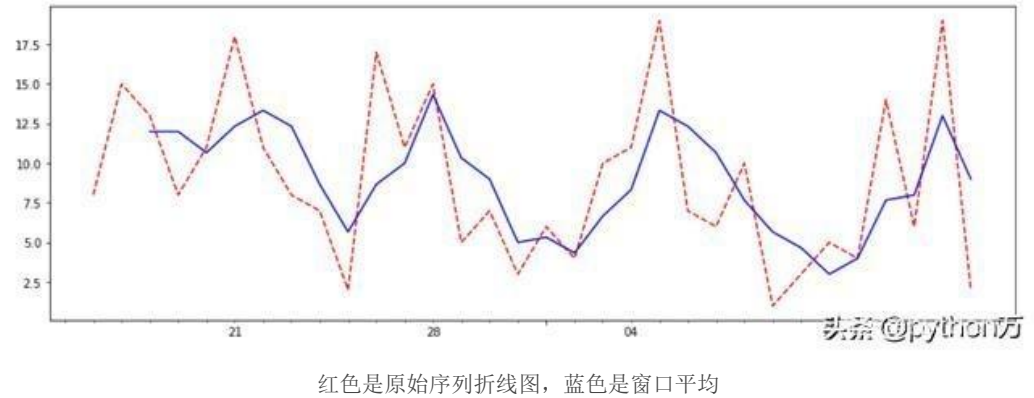
如上图所示，当窗口开始滑动时，第一个时间点和第二个时间点的时间为空，这是因为这里窗口长度为3，他们前面的数都不够3，所以到2019-01-18时，他的数据就是2019-01-16到2019-01-18三天的均值。那么有人就会这样想，在计算2019-01-16序列的窗口数据时，虽然不够窗口长度3，但是至少有当天的数据，那么能否就用当天的数据代表窗口数据呢？答案是肯定的，这里我们可以通过min_periods参数控制，表示窗口最少包含的观测值，小于这个值的窗口长度显示为空，等于和大于时有值，如下所示：

- #表示窗口最少包含的观测值为1
- ser_data.rolling(3,min_periods=1).mean()



上面讲解了两个常用的参数，下面我们通过绘图来观察原始序列与滑动窗口产生的序列值的关系图，如下：

- index=pd.date_range('20190116','20190216')
- np.random.seed(2)
- data=np.random.randint(20,size=len(index))
- ser_data=pd.Series(data,index=index)
- plt.figure(figsize=(15, 5))
- ser_data.plot(style='r--')
- ser_data.rolling(3).mean().plot(style='b')



可以从上面的图中看到通过窗口取的值比原始序列数据更加平稳，这也是为何使用滑动窗口的原因。下节讲解时间序列分析中的平稳性和差分法，偏理论，喜欢的小伙伴关注收藏！