

# python时间序列分析之\_用pandas中的rolling函数计算时间窗口数据



### 简介

由于系统编辑器限制,代码行用无序列表表示。

上篇文章中,我们讲解了如何对时间数据进行重采样及重采样中降采样和升采样的概览和使用方法,通过重采样我们可以得到任何想要频率的数据,但是这些数据也是一个时点的数据,那么就存在这样一个问题:时点的数据波动较大,某一点的数据不能很好的表现它本身的特性,于是我们就想,能否用一个区间的的数据去表现呢,这样数据的准确性是不是更好一些呢?因此,引出滑动窗口(移动窗口)的概念,简单点说,为了提升数据的可靠性,将某个点的取值扩大到包含这个点的一段区间,用区间来进行判断,这个区间就是窗口。如下面的示意图所示,其中时间序列数据代表的是15日每日的温度,现在我们以3天为一个窗口,将这个窗口从左至右依次滑动,统计出3天的平均值作为这个点的值,比如3号的温度就是1号、2号、3号的平均温度,通过示意图我们已经理解了滑动窗口,下面我们就看看pandas中使用的函数及各参数的含义。



滑动窗口示意图

## pandas.DataFrame.rolling参数详解

pandas中提供了pandas.DataFrame.rolling这个函数来实现滑动窗口值计算,下面是这个函数的原型:

DataFrame.rolling(window, min\_periods=None, center=False, win\_type=None, on=None, axis=0, closed=None),参数含义如下图:

# 作者最新文章

时间序列的平稳性及使用差分法 处理非平稳时间序列

python时间序列分析之\_用 pandas中的rolling函数计算时间 窗口数据

Python时间序列分析之\_时间重 采样(降采样和升采样)

### 相关文章

对照检查! 高效程序员几乎都有 这七项技能



Java 服务端乱象大盘点



还不会if函数的嵌套判断,试试 这种方法,点点鼠标即可



为什么说Python是学习人工智能的第一语言?





	则数值表示计异统计重的观测值的数重即向前儿个数据。如果定 offset类型,表示时间窗的大小		
min_periods	ods 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N 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offeet类型默认是 在开右闭的即默认为right。可以根据情况指定为reft 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.pylab 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	6,490,000

数据预览

ser\_data.rolling(3).mean()

2019-01-16	4	2019-01-16 Na.
2019-01-17	8	2019-01-17 Nal
2019-01-18	6	2019-01-18 6. 000000
2019-01-19	5	2019-01-19 6. 333333
2019-01-20	9	2019-01-20 6. 66666
2019-01-21	1	2019-01-21 5. 000000
2019-01-22	4	2019-01-22 4. 66666
2019-01-23	5	2019-01-23 3, 333333
2019-01-24	2	2019-01-24 3. 66666
2019-01-25	4	2019-01-25 3. 66666
2019-01-26	6	2019-01-26 4, 000000
2019-01-27	7	2019-01-27 5. 66666
2019-01-28	9	2019-01-28 7. 333333
2019-01-29	13	2019-01-29 9. 66666
2019-01-30	6	2019-01-30 共產黨的銀額的



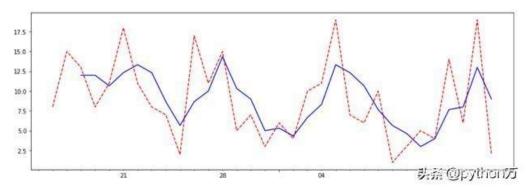
- #表示窗口最少包含的观测值为1
- ser data.rolling(3,min periods=1).mean()

2019-01-16	4	2019-01-16	4. 000000
2019-01-17	8 -	2019-01-17	6. 000000
2019-01-18	6	2019-01-18	6. 000000
2019-01-19	5	2019-01-19	6. 333333
2019-01-20	9	2019-01-20	6.666667
2019-01-21	1	2019-01-21	5.000000
2019-01-22	4	2019-01-22	4.666667
2019-01-23	5	2019-01-23	3. 333333
2019-01-24	2	2019-01-24	3.666667
2019-01-25	4	2019-01-25	3.666667
2019-01-26	6	2019-01-26	4.000000
2019-01-27	7	2019-01-27	5.666667
2019-01-28	9	2019-01-28	7. 333333
2019-01-29	13	2019-01-29	9.666667
2019-01-30	6	2019-01-30	學房 到1000000万

min\_periods=1

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')



红色是原始序列折线图,蓝色是窗口平均

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!