result\_02\_09\_3:

1. 使用了BCD三周;
2. 多项式,degree = 3
3. SumABCD,open\_ratio,ratio\_wk
4. 线下结果：0.0803401271018
5. 线上结果：0.08626608

Result\_02\_10\_1:

1. 使用了CD两周；
2. 2、3都与上面相同
3. 线下结果：0.0803401271018
4. 线上结果：0.08683246

结论:使用BCD三周作为特征的效果更好.

Result\_02\_10\_2：

1. 使用了BCD三周；
2. 多项式degree=3；
3. sumABCD、open\_ratio、ratio\_wk、meanABCD、OHE\_city\_name、OHE\_cate\_2；
4. 线下结果：0.082923905297
5. 线上结果：0.08588201

Result\_02\_11\_1: 特殊原因，提交于2-10 night

1. 使用了BCD三周；
2. 多项式 degree=3
3. sumABCD、open\_ratio、ratio\_wk、meanABCD、OHE\_city\_name、OHE\_cate\_1、OHE\_cate\_2、OHE\_cate\_3、OHE\_shop\_level
4. 线下结果：0.0824784103763
5. 线上结果：0.08589597

Result\_02\_11\_1\_degree2：特殊原因，提交于 2-11 00:00

1. 多项式 degree=2;
2. 其余与Result\_02\_11\_1相同；
3. 线下结果：0.0824784103763
4. 线上结果：0.08518808

结论:降低多项式degree会取得更好效果（模型为RandomForestRegressor）

Result\_02\_11\_2:

1. 未使用多项式；
2. 其余与Result\_02\_11\_1\_degree2相同；
3. 线下结果：0.0829401099118
4. 线上结果：0.0541657

结论:若使用多项式,则degree为2最合适。

Result\_02\_11\_3:

1. 使用了BCD三周;
2. 多项式 degree=2;
3. sumABCD、open\_ratio、ratio\_wk、meanABCD、OHE\_city\_name、OHE\_cate\_1、OHE\_cate\_2、OHE\_cate\_3、OHE\_shop\_level、count\_user\_view(未使用多项式)
4. 线下结果：0.0825589501658
5. 线上结果：0.08508653

Result\_02\_12\_1:

1. 与 \_02\_11\_3 相同，count\_user\_view也是用了多项式；
2. 线下结果：未测
3. 线上结果：0.085087

结论：count\_user\_pay/view 不适合同时使用多项式。

ResultA:

1. 直接使用 get\_result(weekA.values) 得到结果；
2. 线上成绩：0.12960395

ResultB:

1. ~ weekB
2. 线上成绩：0.10467468

ResultC：

1. ~weekC
2. 线上成绩：0.011034316

ResultD：

1. ~weekD
2. 线上成绩：0.10270536

Result\_02\_12\_3：

1. 使用了BCD三周；
2. 多项式 degree=2；
3. 模型参数：RandomForestRegressor(n\_estimators=1200,random\_state=1,n\_jobs=-1,min\_samples\_split=2,min\_samples\_leaf=2,max\_depth=15)
4. 线下结果：0.0825338016242
5. 线上结果：0.08516256

Result\_02\_13\_2:

1. 模型参数：RandomForestRegressor(n\_estimators=1200,random\_state=1,n\_jobs=-1,min\_samples\_split=2,min\_samples\_leaf=1,max\_depth=25)
2. 线下结果：未测
3. 线上结果：0.08503127

Result\_02\_13\_3：

1. 模型参数：RandomForestRegressor(n\_estimators=1200,random\_state=1,n\_jobs=-1,min\_samples\_split=2,min\_samples\_leaf=2,max\_depth=25)
2. sumABCD、open\_ratio、ratio\_wk、meanABCD、OHE\_city\_name、OHE\_cate\_1、OHE\_cate\_2、OHE\_cate\_3、OHE\_shop\_level、count\_user\_view(未使用多项式)、std、max、min、median、mad、var
3. 线下结果：0.0825495321341
4. 线上结果：0.08502533

Result\_02\_14\_1：

1. 模型参数：ExtraTreesRegressor(n\_estimators=1200,random\_state=1,n\_jobs=-1,min\_samples\_split=2,min\_samples\_leaf=2,max\_depth=25)
2. 其他与 02\_13\_3相同
3. 线下结果：0.0819716712401
4. 线上结果：0.08480985

Result\_02\_14\_3:

1. 多项式只有后两周使用 degree=2;
2. 其他与 02\_14\_1相同；
3. 线下结果：未测
4. 线上结果：0.08513521

结论：在Rxtra tree 和 RF的模型中，使用BCD三周，degree =2 是目前最好的方法。

Result\_02\_15\_2:

1. Max\_features=270 (总共：297)
2. Degree =2
3. 采纳的特征与上述相同。
4. 线下结果：0.0819070038684
5. 线上结果：0.08466170

Result\_02\_16\_3:

1. 先利用 02\_15\_2得出第一周；
2. 在根据第一周预测第二周；
3. 线上结果：0.085

结论:利用迭代的预测,效果不好.

Result\_02\_17\_2: 推翻重做

1. Train\_x：2015-10-12 ~ 2015-10-31

Train\_y：2015-11-3 ~ 2015-11-16

Test\_x： 2016-10-10 ~ 2016-10-29

1. 模型参数：

ET = ExtraTreesRegressor(n\_estimators=1200,random\_state=1,n\_jobs=-1,min\_samples\_split=2,min\_samples\_leaf=2,max\_depth=25,max\_features='auto')

1. 特别说明：
2. train的sample 仅为2015-07-17~ 07-23开业的商铺（并集），共331家；
3. train\_y 中，11-10，11-11 被 11-3，11-4替换，因为“11-11”对客流有促进作用，但前后两年星期与日期对不上。后面考虑加入“11-11”效应。

4． 线上结果：0.13562398

Result\_02\_17\_3\_x：

1. Train\_x：2015-10-13 ~ 2015-11-02

Train\_y：2015-11-3 ~ 2015-11-16

Test\_x： 2016-10-11 ~ 2016-10-31

1. train\_y 中，11-10，11-11 被 11-3，11-4替换；在11-12，11-13中加入了对应时间的差值/2。（由于星期的不匹配，“11-11”应当被推迟两天）
2. 利用2015年的数据做交叉验证，得到优化后的参数：

ET = ExtraTreesRegressor(n\_estimators=2600,random\_state=1,n\_jobs=-1,min\_samples\_split=2,min\_samples\_leaf=1,max\_depth=12,max\_features='auto')

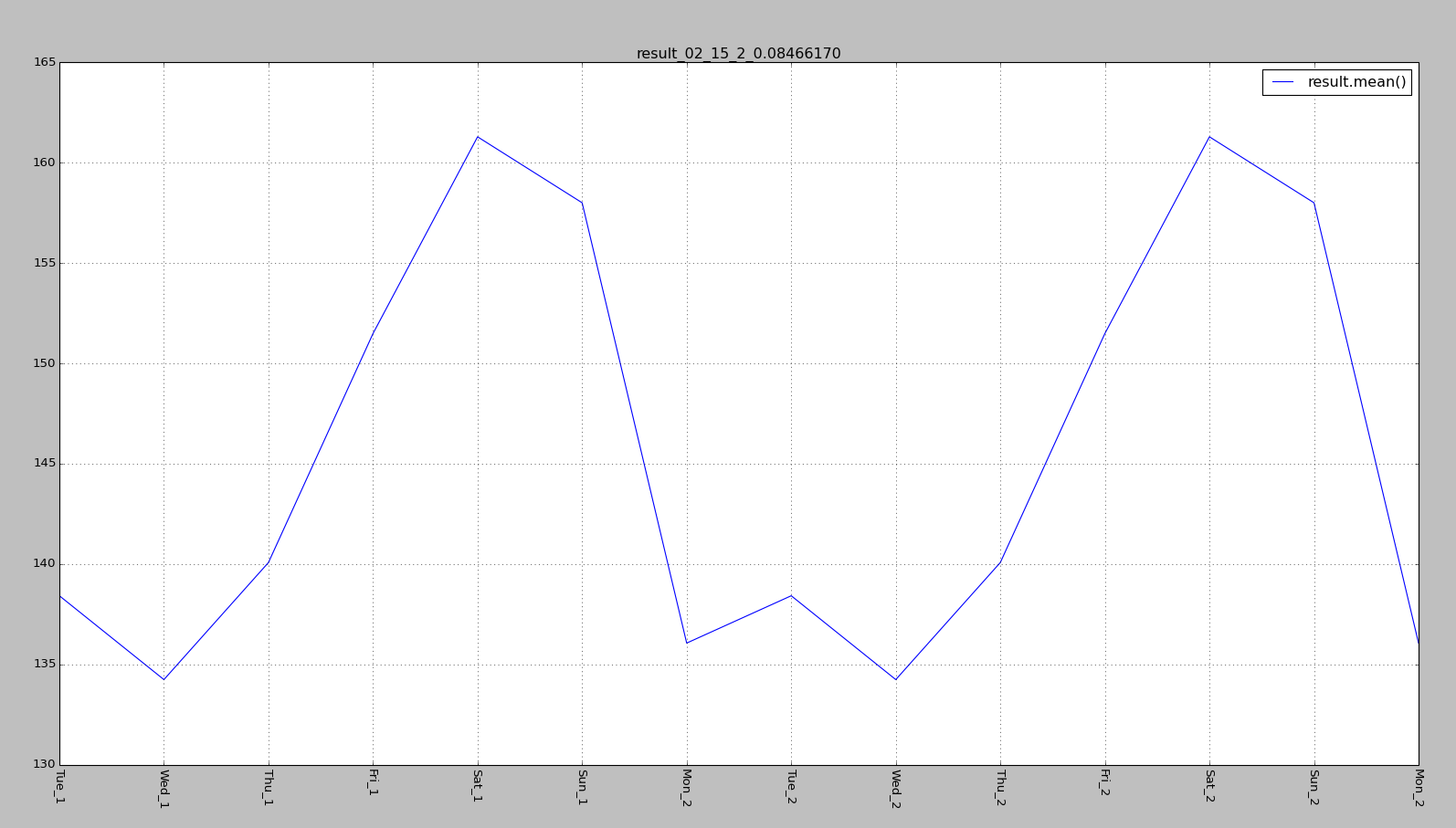
1. 加入了新的特征；
2. Val = (4\*res\_02\_17\_3+res\_02\_18\_1)/5；
3. 线上结果：0.08565399

result\_02\_18\_1：

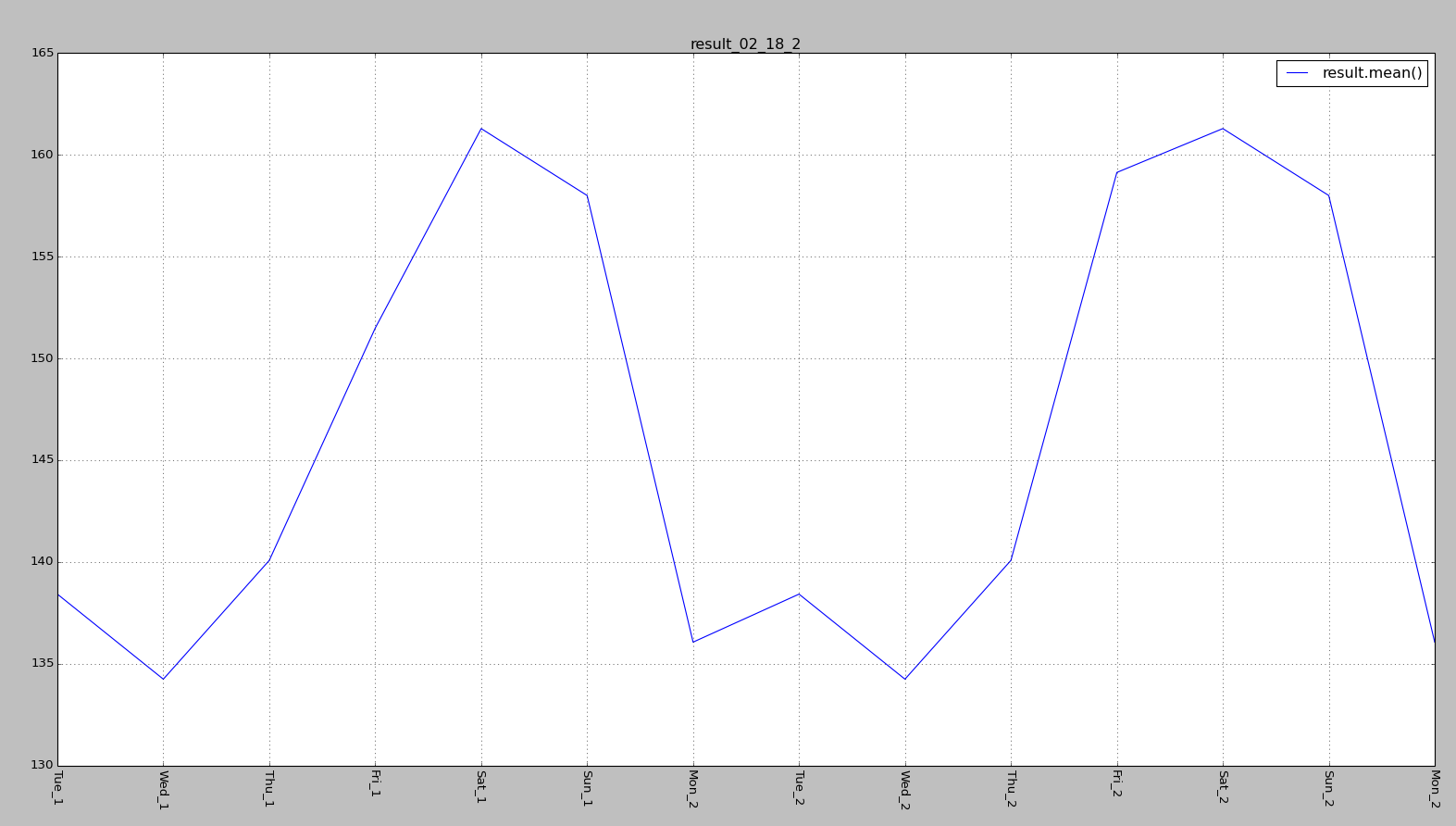
1. 使用 features\_make\_0 制作的特征；
2. 筛选ABCD四周中“开业率>0.85”的商铺，剩余1942个商铺；
3. 线上成绩：0.08501186

Result\_02\_18\_2：

1. Result\_02\_15\_2.Fri\_2 = (4\*result\_02\_15\_2.Fri\_2+result\_02\_17\_3.Fri\_2)/5 #Fri\_2 是双十一对应的日期；
2. 注意：
   1. Result\_02\_15\_2是之前有ET算法的最佳成绩；
   2. Result\_02\_17\_3是利用2015年双十一的情况做的预期；
   3. 这里只对Result\_02\_15\_2 的 Fri\_2做了修改，其余保持其原本数值；



Fri\_2 修改前



Fri\_2 修改后

1. 线上结果：0.08426518

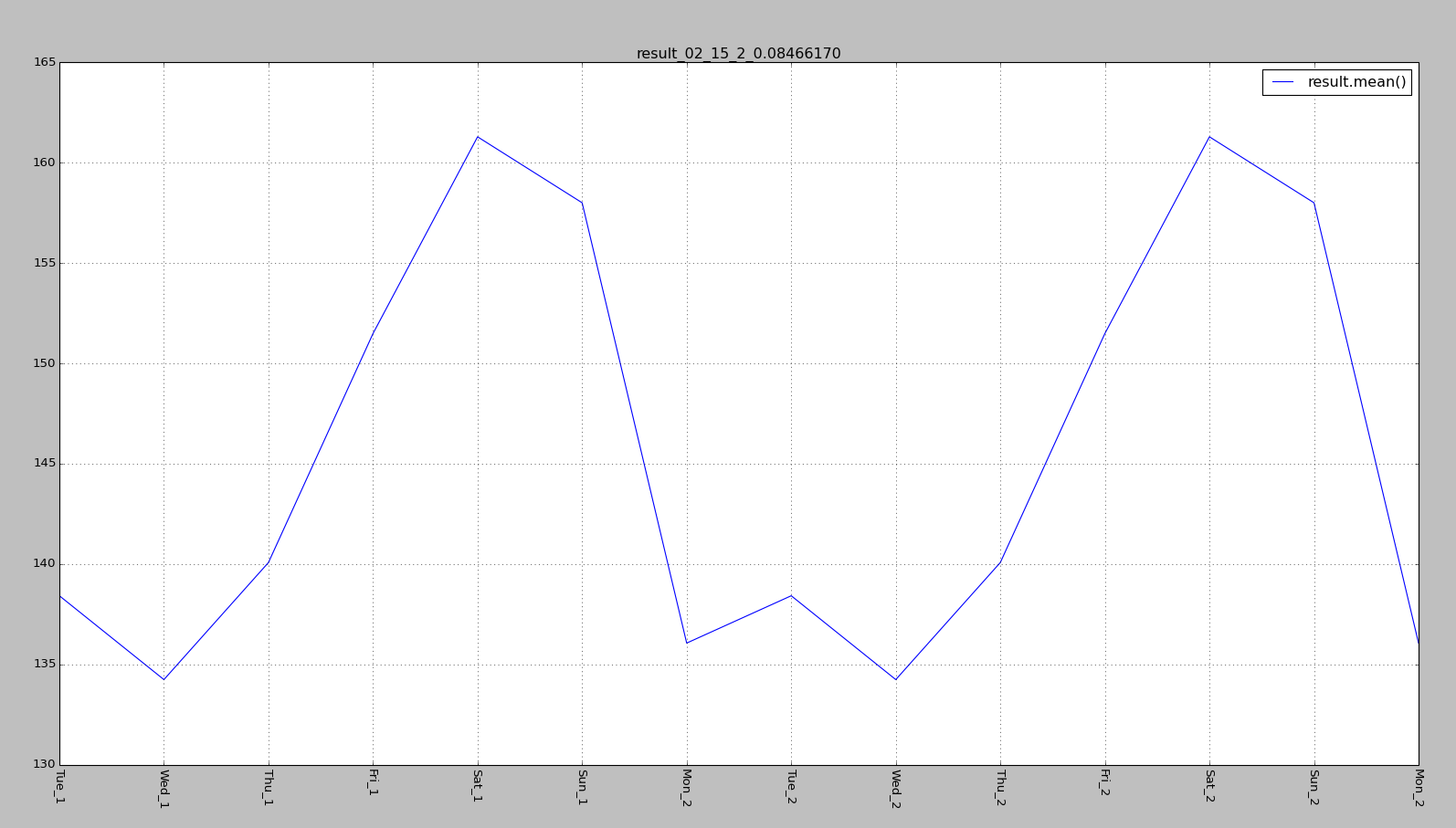
Result\_02\_18\_3:

1. Result\_02\_15\_2 & Result\_02\_17\_3
2. Result\_02\_15\_2.Fri\_2 = (4\*result\_02\_15\_2.Fri\_2+1\*result\_02\_17\_3.Fri\_2)/5

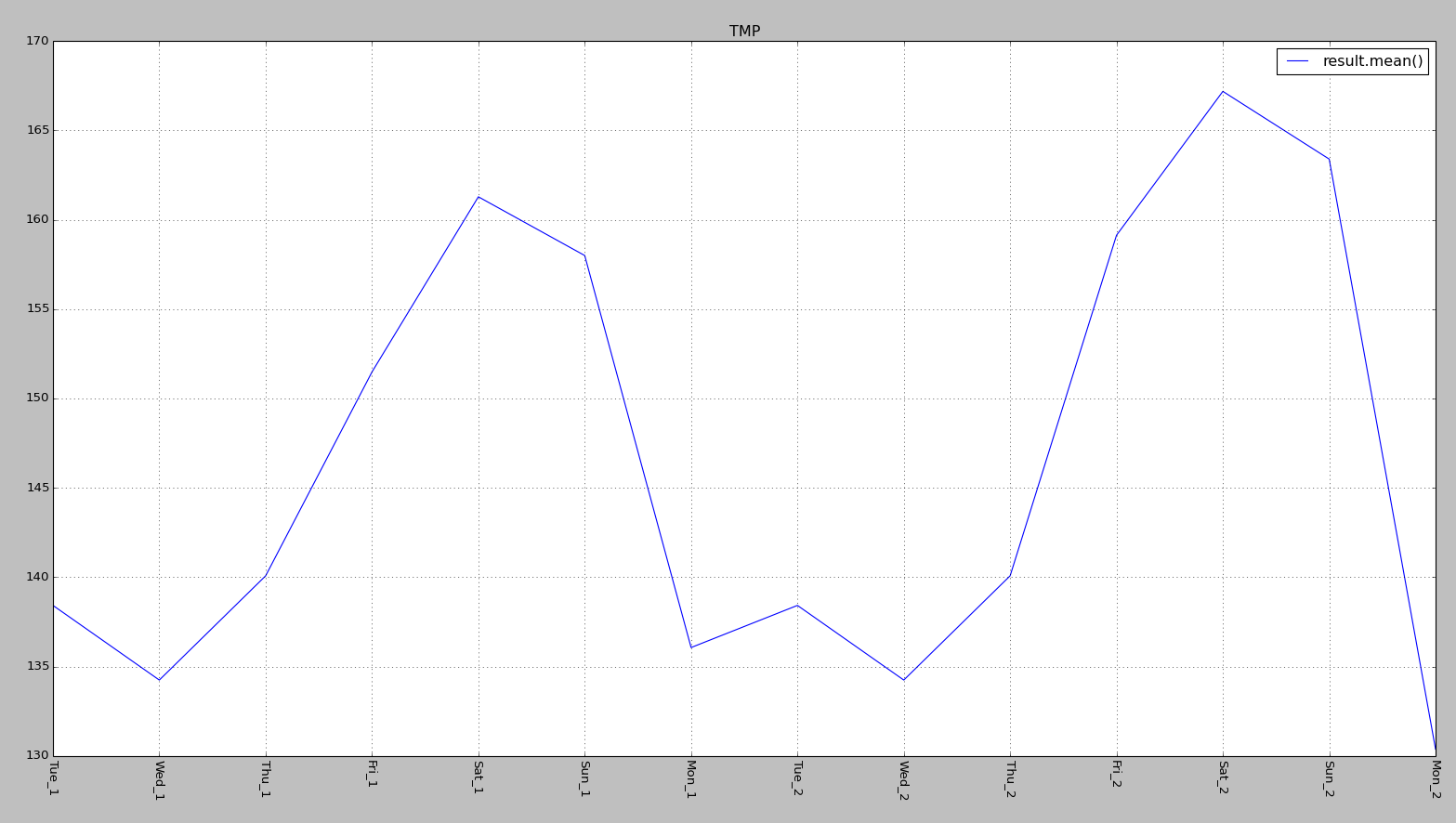
Result\_02\_15\_2.Sat\_2 = (4\*result\_02\_15\_2.Sat\_2+1\*result\_02\_17\_3.Sat\_2)/5

Result\_02\_15\_2.Sun\_2 = (4\*result\_02\_15\_2. Sun\_2+1\*result\_02\_17\_3. Sun\_2)/5

Result\_02\_15\_2.Mon\_2 = (4\*result\_02\_15\_2. Sun\_2+1\*result\_02\_17\_3. Sun\_2)/5



修改前



修改后

1. 线上结果：0.08485787

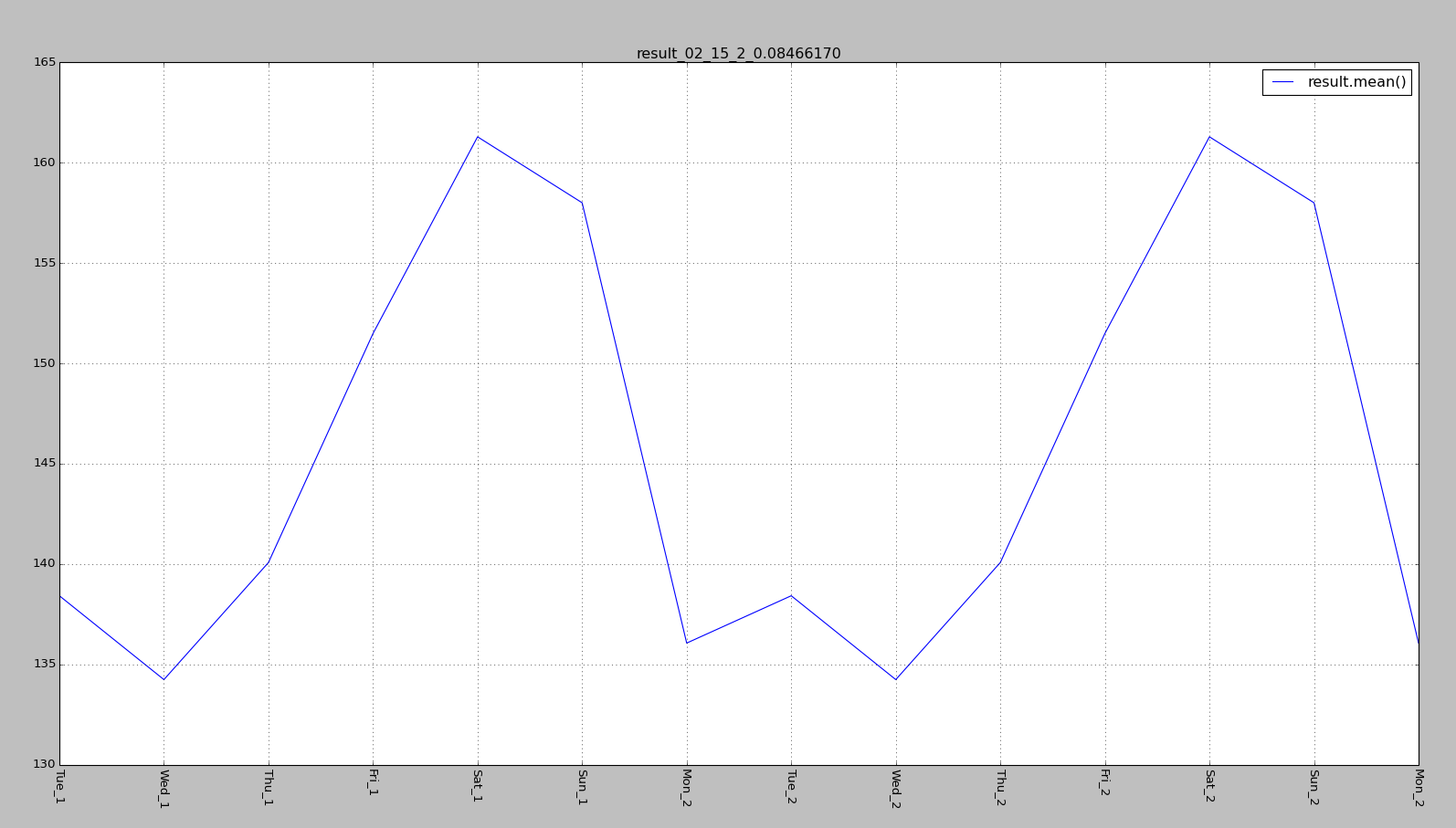
Result\_02\_19\_1:

1. Result\_02\_15\_2 & Result\_02\_17\_3
2. Result\_02\_15\_2.Fri\_2 = (4\*result\_02\_15\_2.Fri\_2+1\*result\_02\_17\_3.Fri\_2)/5

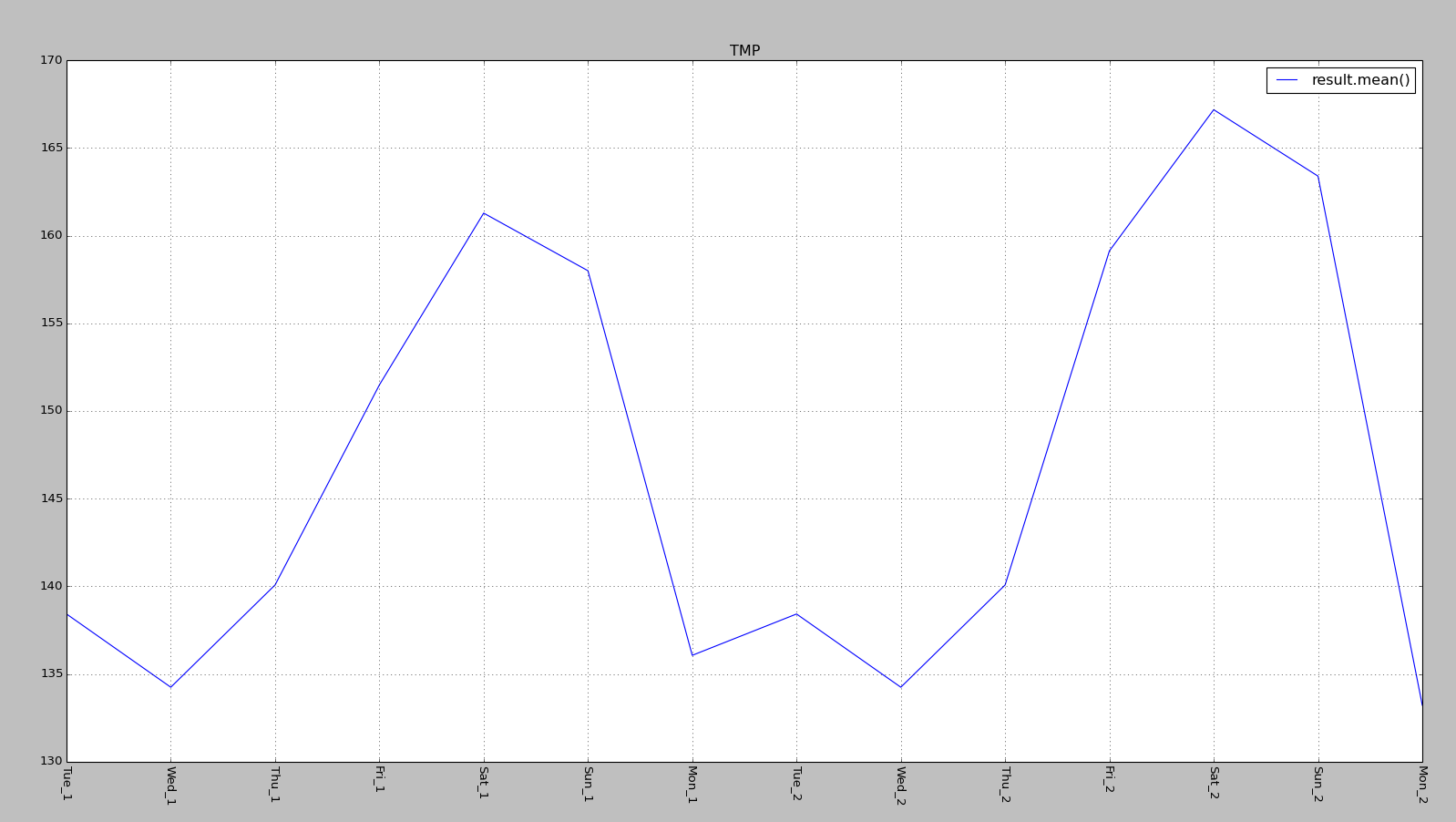
Result\_02\_15\_2.Sat\_2 = (4\*result\_02\_15\_2.Sat\_2+1\*result\_02\_17\_3.Sat\_2)/5

Result\_02\_15\_2.Sun\_2 = (4\*result\_02\_15\_2. Sun\_2+1\*result\_02\_17\_3. Sun\_2)/5

Result\_02\_15\_2.Mon\_2 = (9\*result\_02\_15\_2. Sun\_2+1\*result\_02\_17\_3. Sun\_2)/10



修改前



修改后

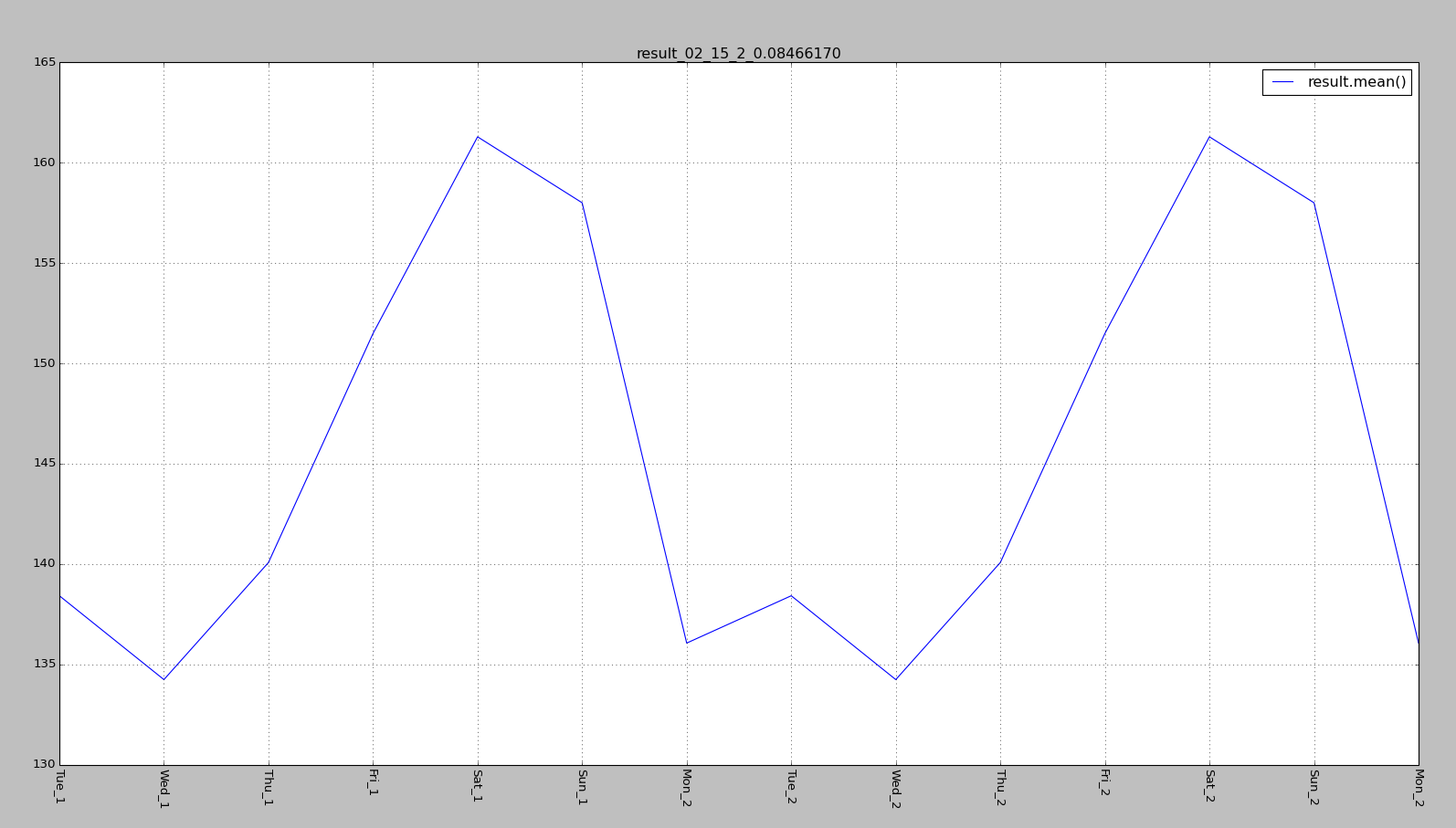
1. 线上结果：0.08482675

Result\_02\_19\_2:

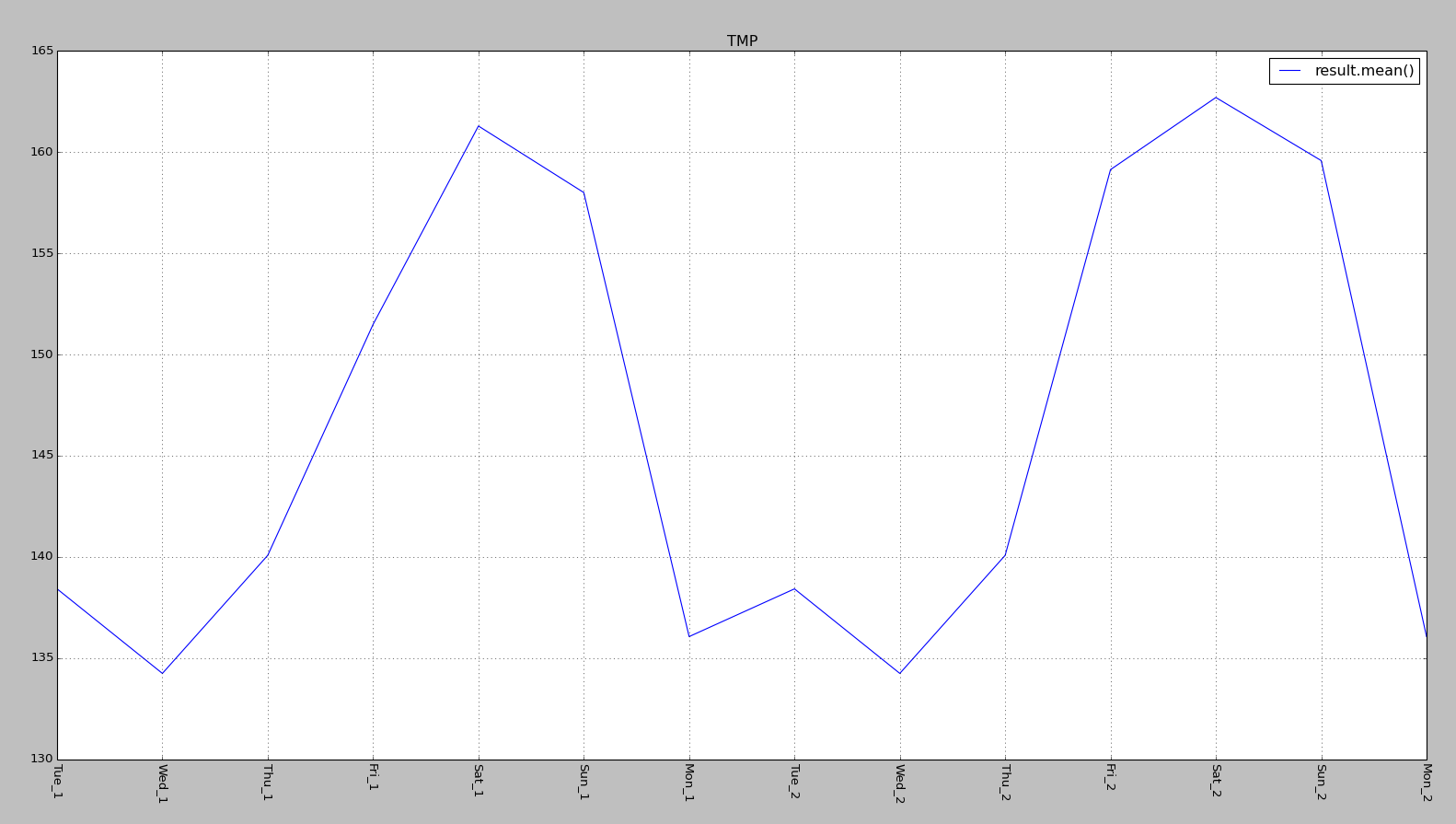
1. Result\_02\_15\_2 & Result\_02\_17\_3
2. Result\_02\_15\_2.Fri\_2 = (4\*result\_02\_15\_2.Fri\_2+1\*result\_02\_17\_3.Fri\_2)/5

Result\_02\_15\_2.Sat\_2 = (19\*result\_02\_15\_2.Sat\_2+1\*result\_02\_17\_3.Sat\_2)/20

Result\_02\_15\_2.Sun\_2 = (19\*result\_02\_15\_2. Sun\_2+1\*result\_02\_17\_3. Sun\_2)/20



修改前

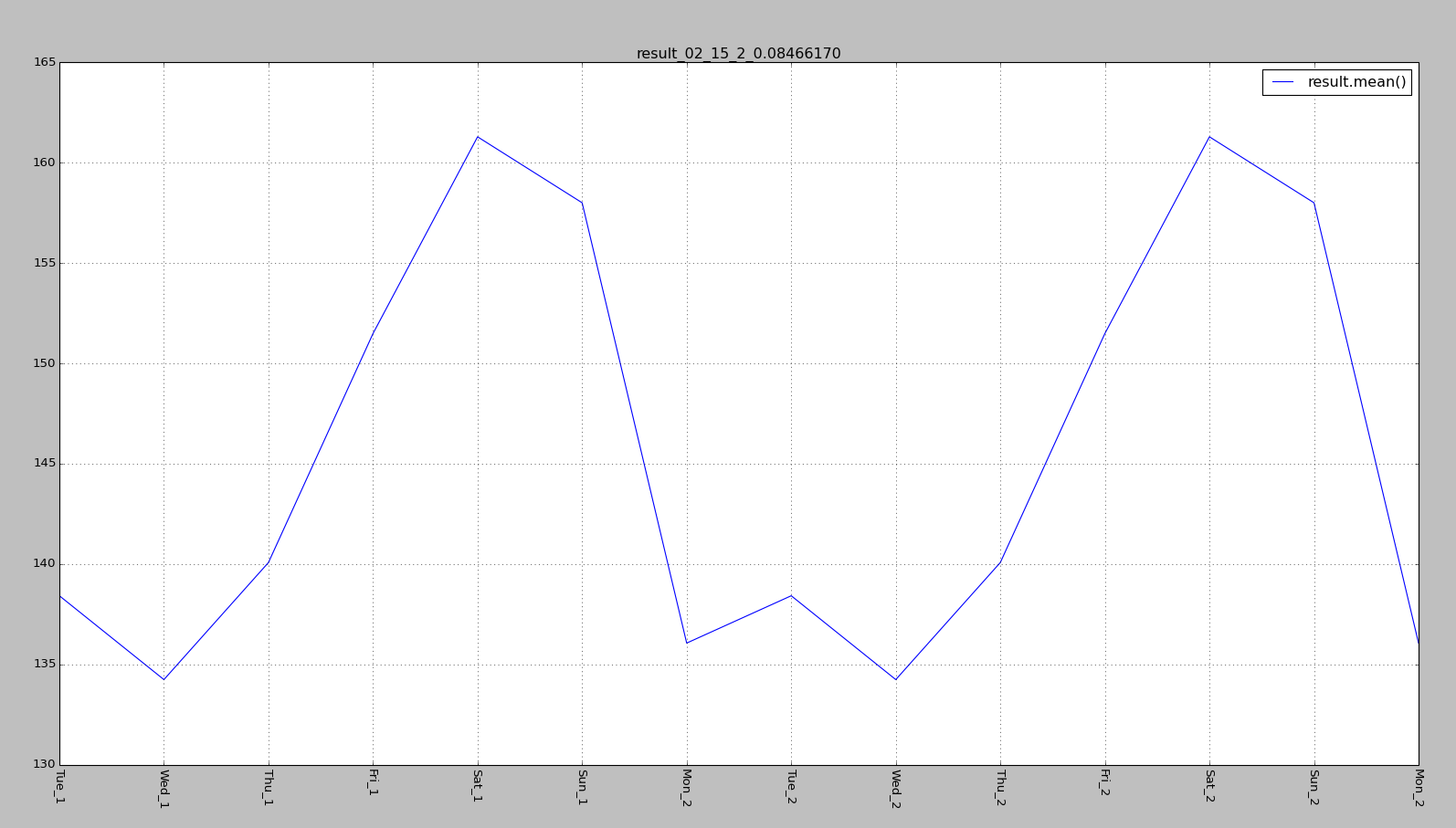


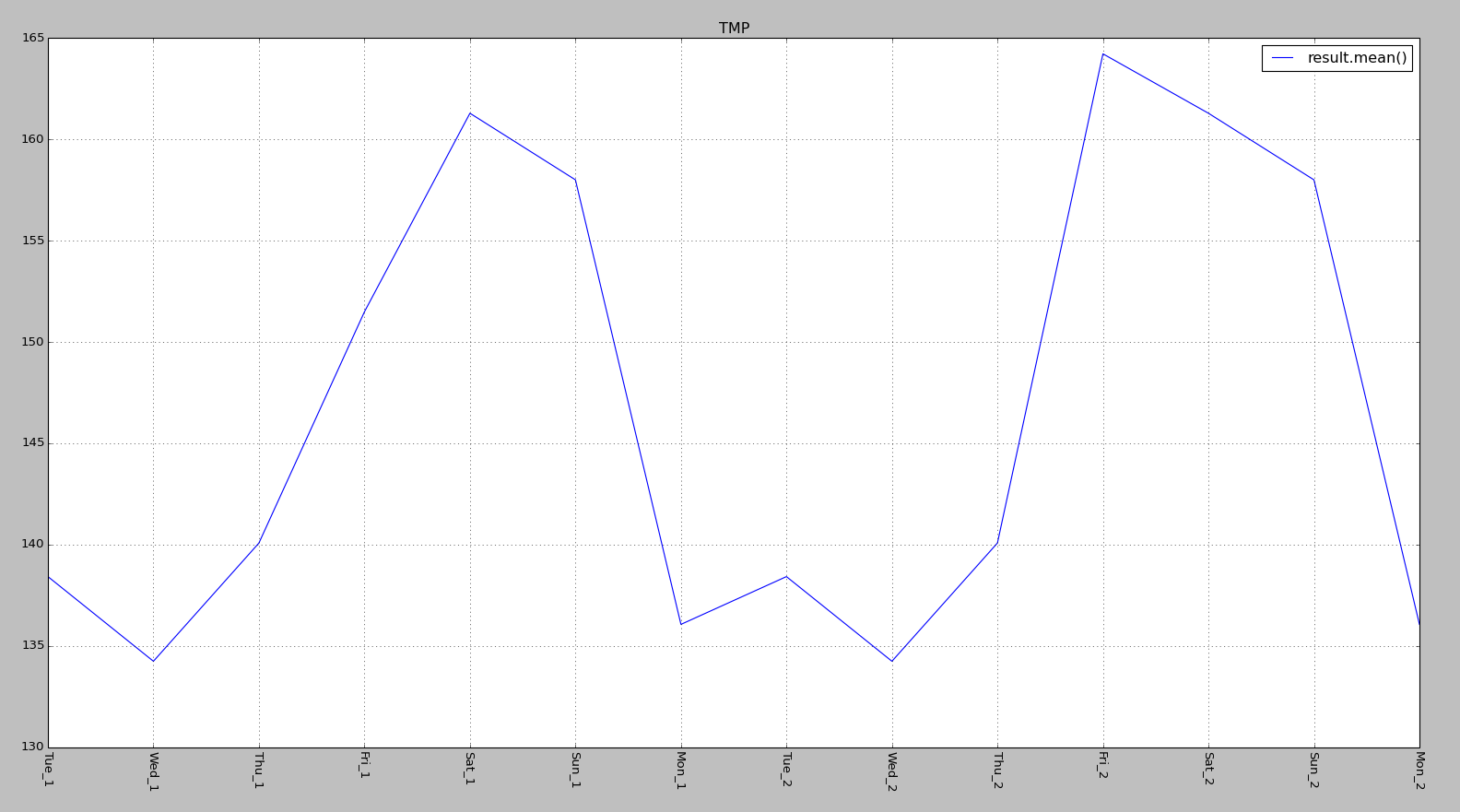
修改后

1. 线上结果：0.08447191

Result\_02\_19\_3：

1. Result\_02\_15\_2 & Result\_02\_17\_3
2. Result\_02\_15\_2.Fri\_2 = (2\*result\_02\_15\_2.Fri\_2+1\*result\_02\_17\_3.Fri\_2)/3

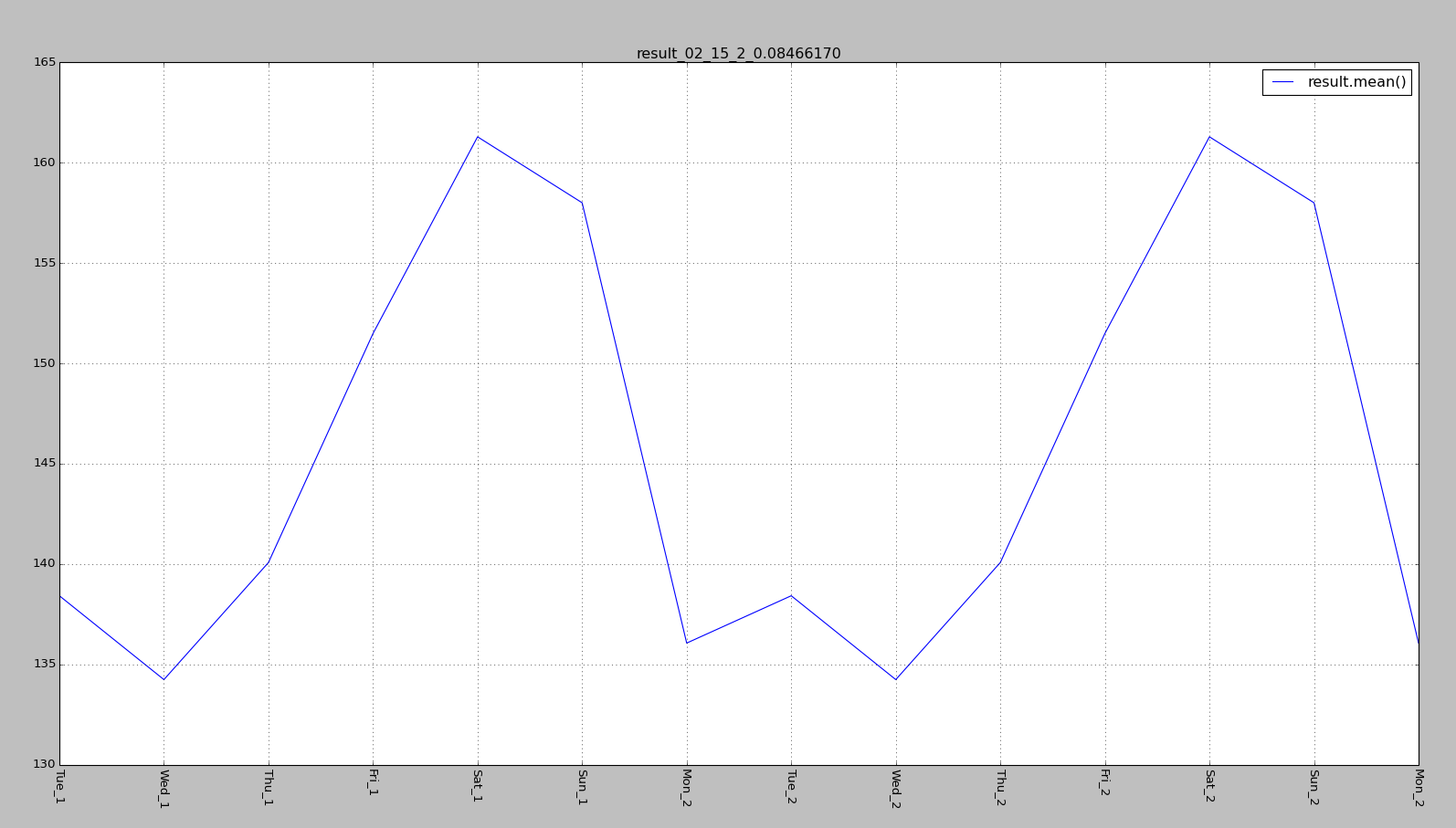


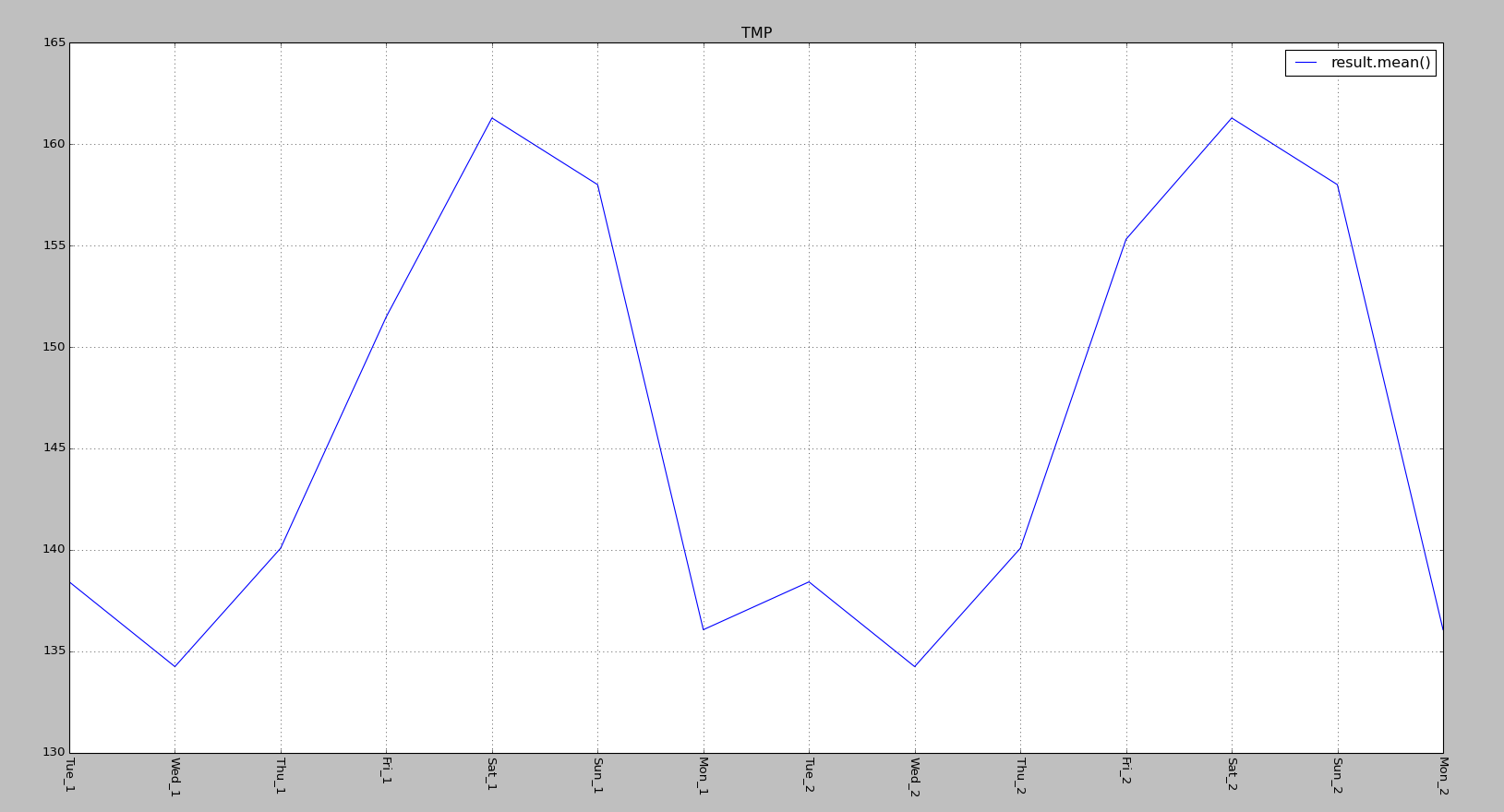


1. 线上结果：0.08432109

Result\_02\_20\_1:

1. Result\_02\_15\_2 & Result\_02\_17\_3
2. Result\_02\_15\_2.Fri\_2 = (9\*result\_02\_15\_2.Fri\_2+1\*result\_02\_17\_3.Fri\_2)/10

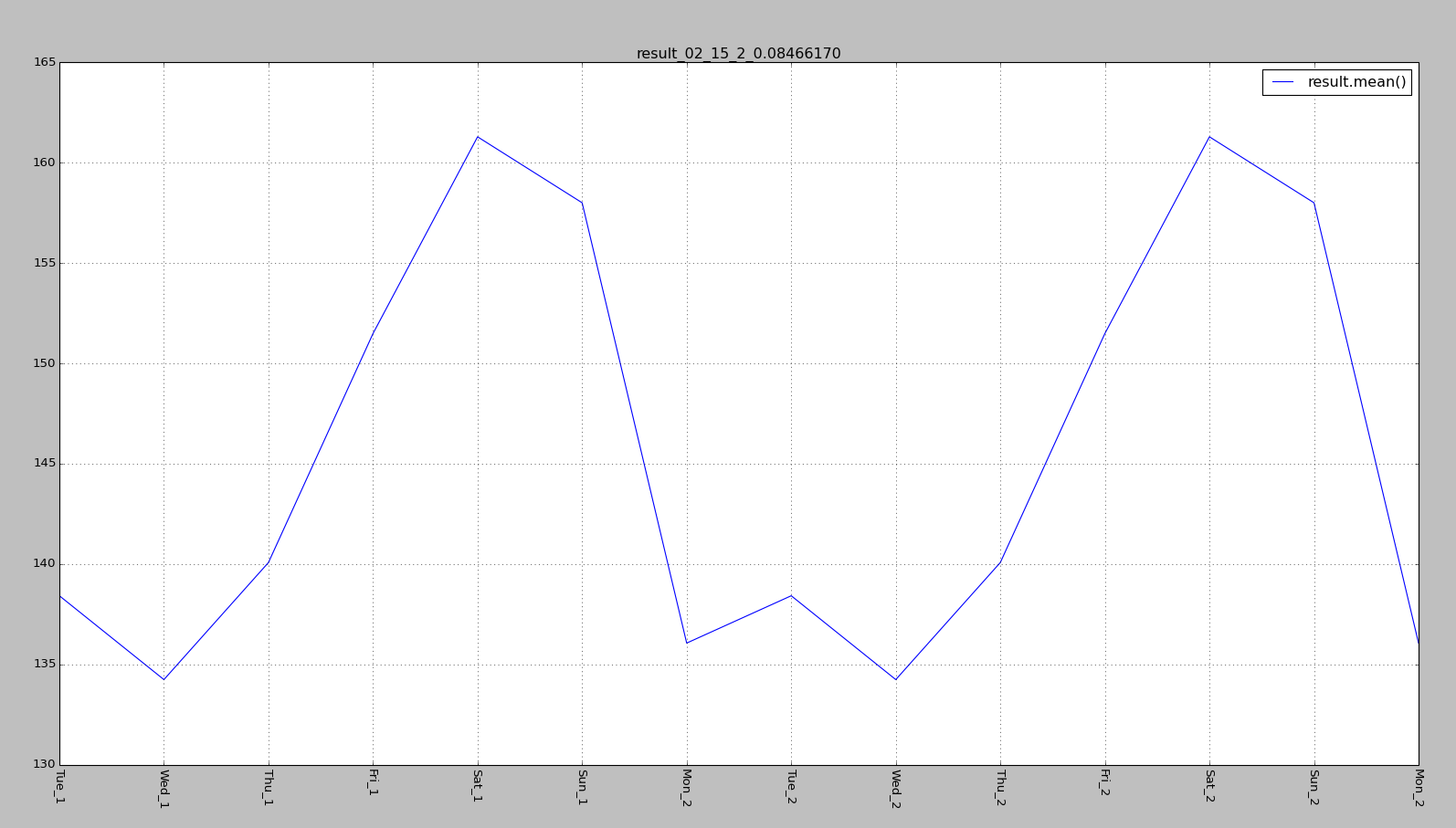


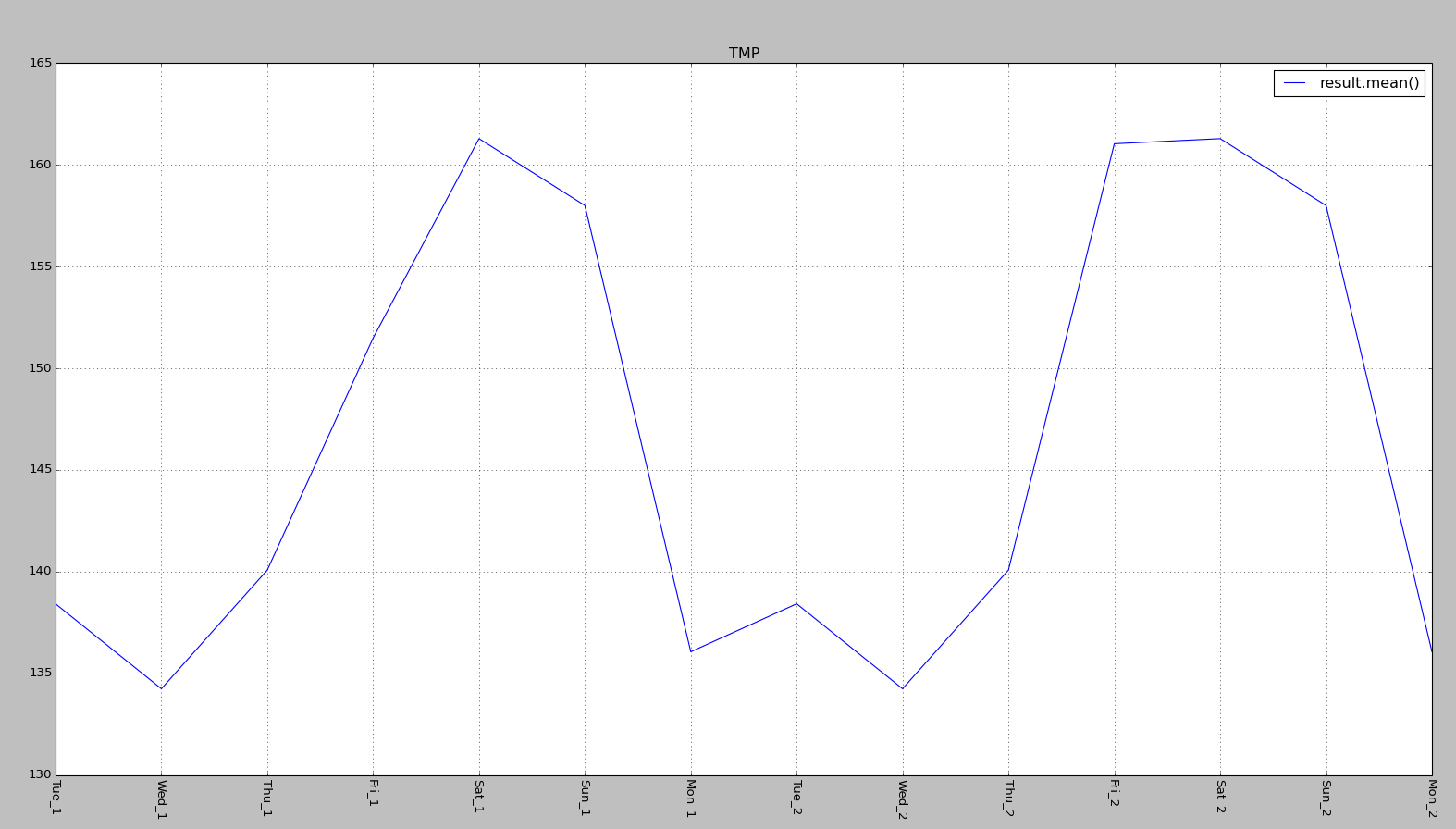


1. 线上结果：0.08437528

Result\_02\_20\_1:

1. Result\_02\_15\_2 & Result\_02\_17\_3
2. Result\_02\_15\_2.Fri\_2 = (3\*result\_02\_15\_2.Fri\_2+1\*result\_02\_17\_3.Fri\_2)/4





1. 线上结果：？

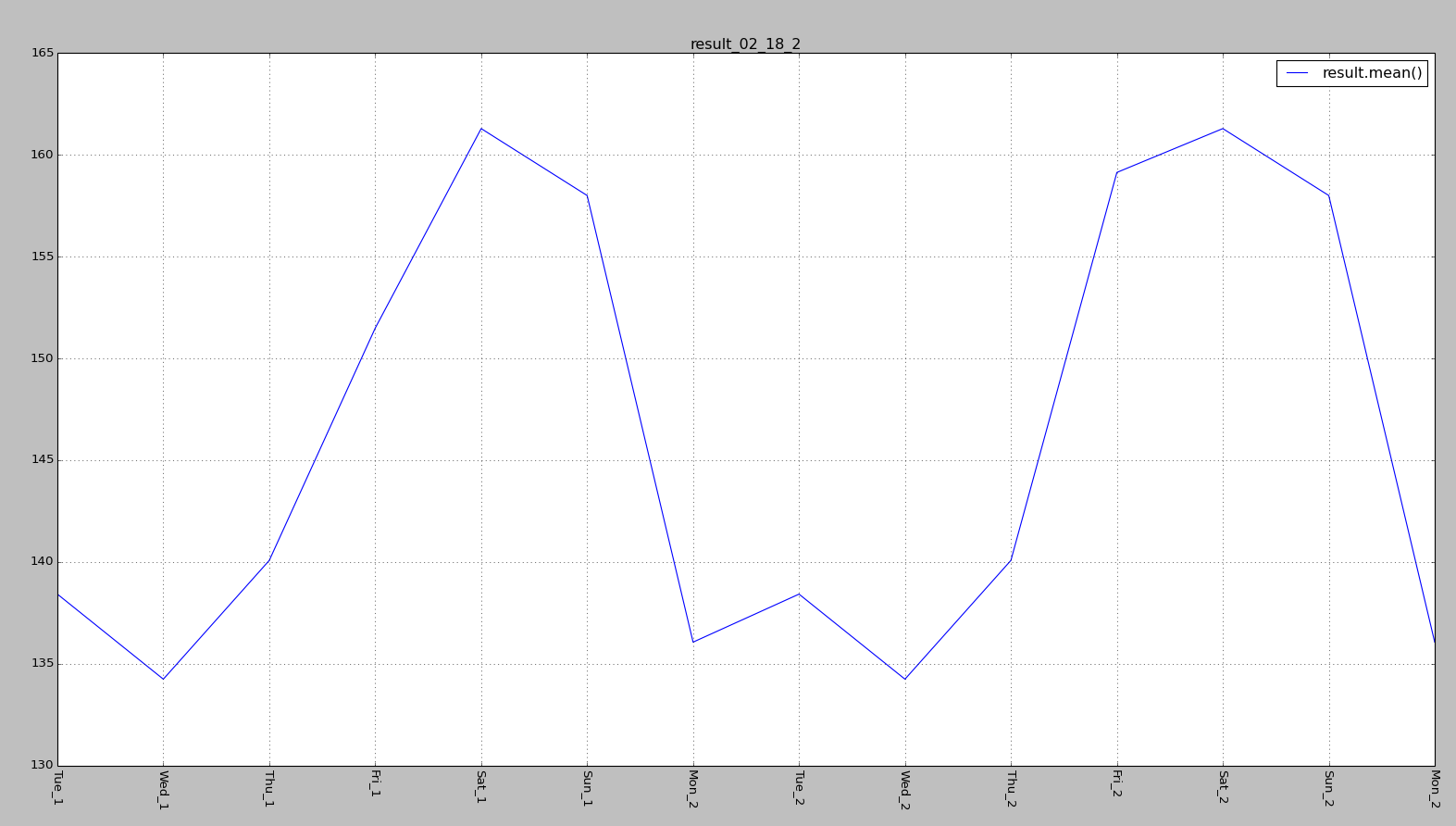
Result\_02\_20\_3\_x:

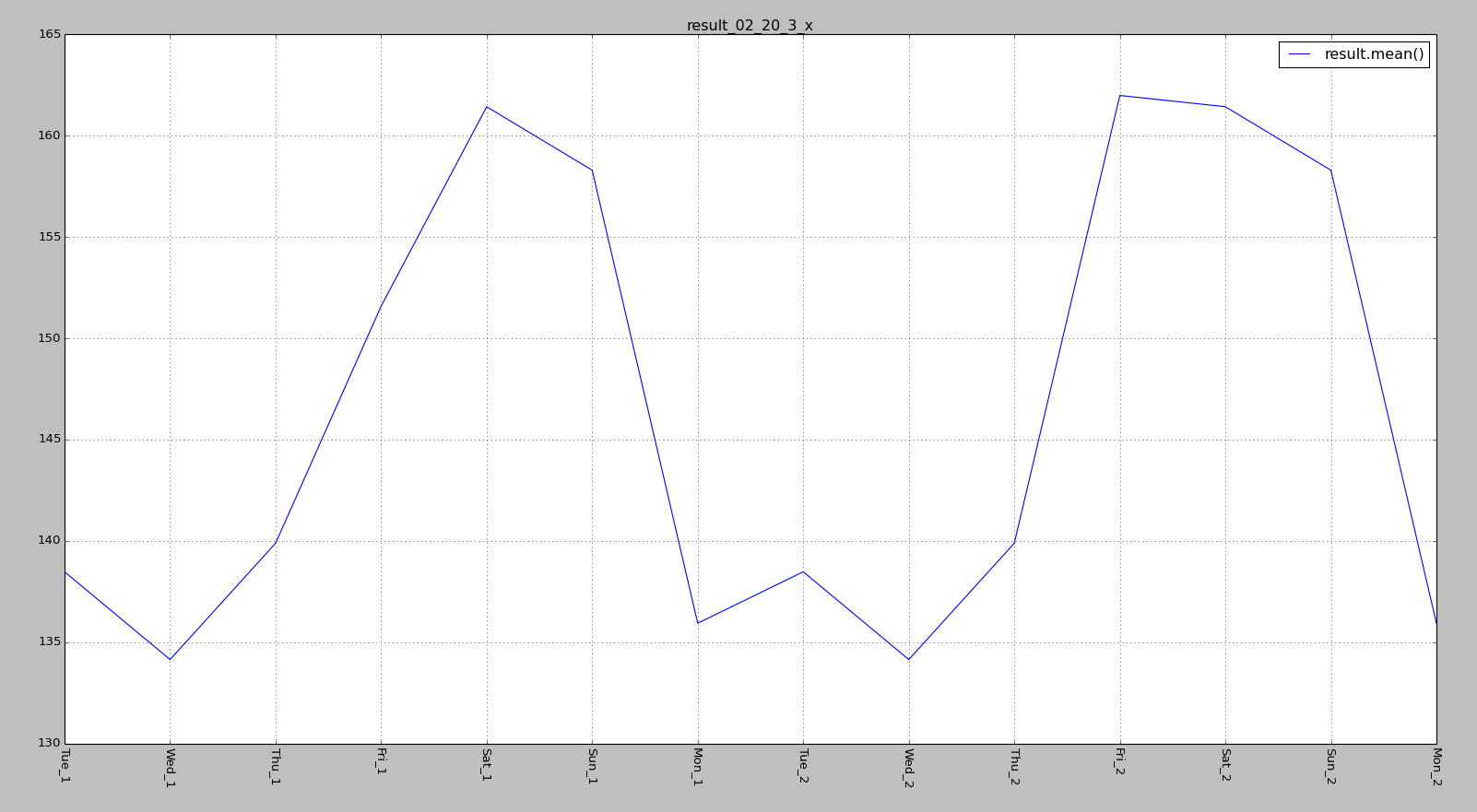
1. result\_02\_20\_3 & result\_02\_21\_1
2. result\_02\_20\_3 将 max\_features = ‘sqrt’
3. result\_02\_21\_1 在 make\_features\_2 中加入了新的特征，并且 max\_features = ‘sqrt’，

（2、3的目的都是防止过拟合）

4. 合并方式详见 tools-> combine\_two\_results()

5. 只修改了Fri\_2

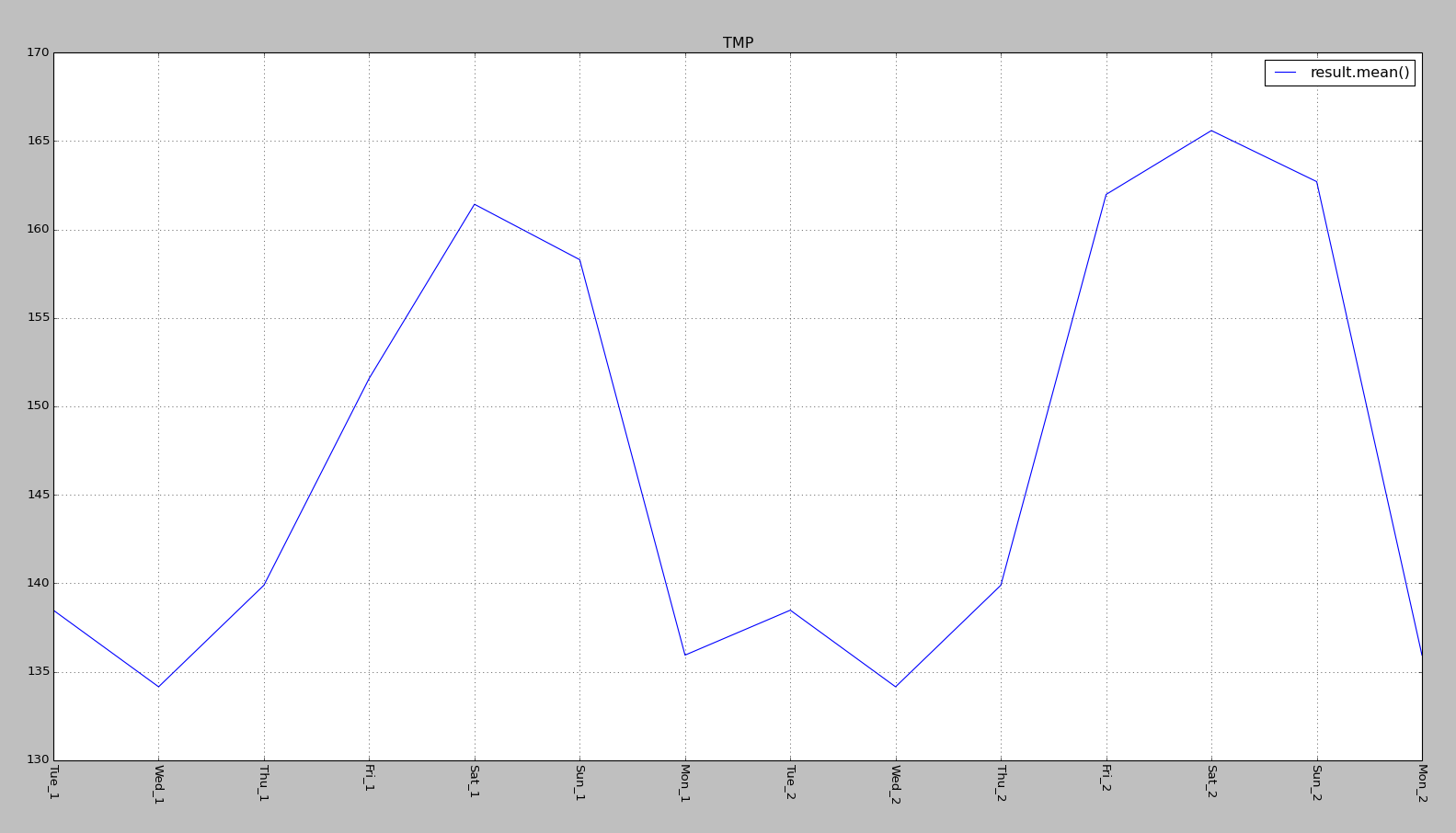
既 Fri\_2 = (res2.Fri\_2 + res1.Fri\_2)/2.0



修改后

1. 线上结果：0.08427561

Result\_02\_21\_1\_x:

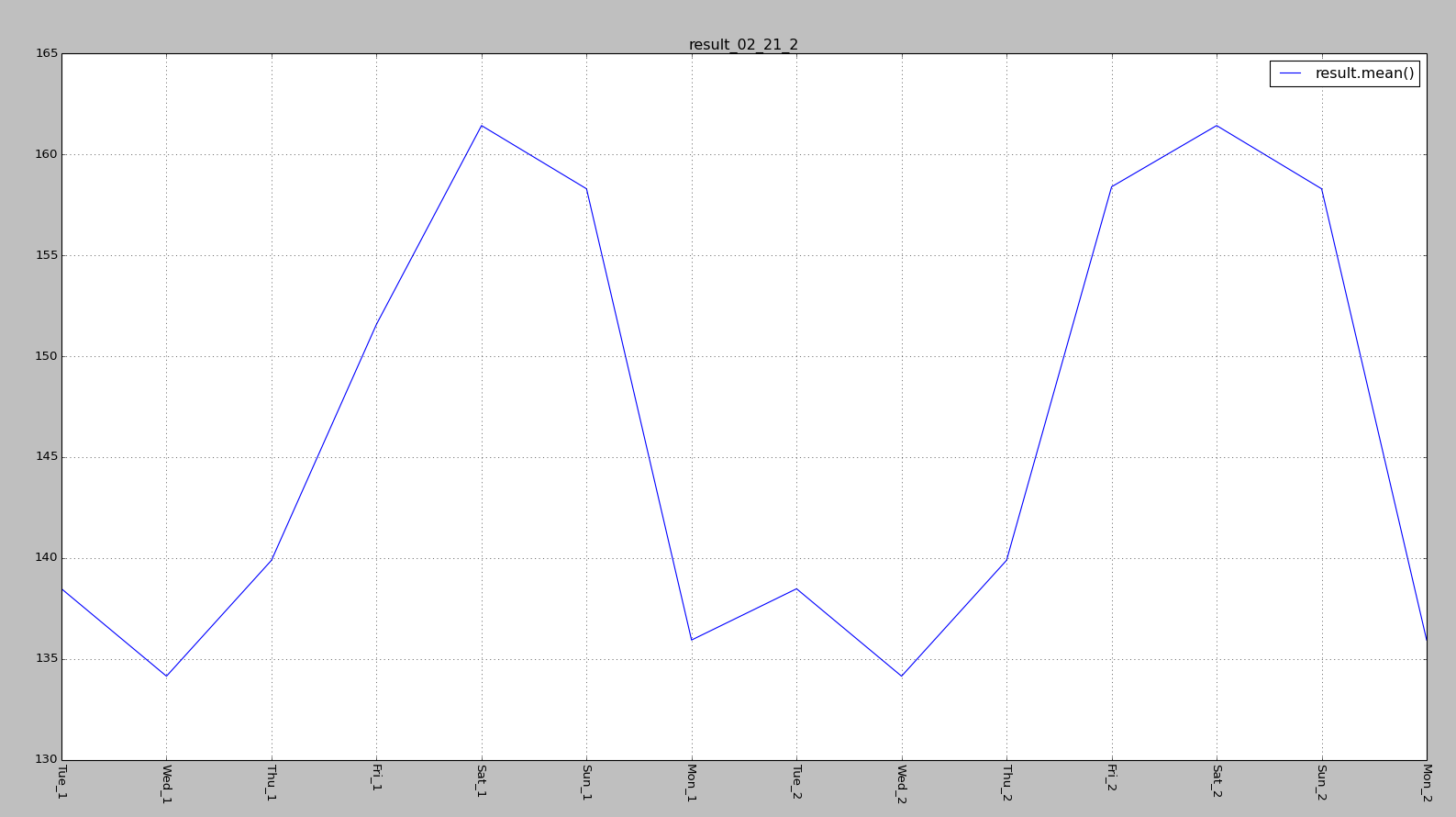
1. 修改了Fri\_2、Sat\_2、Sun\_2 详见combine\_two\_results;
2. 如图
3. 
4. 线上结果：0.08509682

result\_02\_21\_2:

1. result\_name1 = 'result\_02\_20\_3'

result\_name2 = 'result\_02\_21\_1'

1. res1\_Fri\_2[i] = round((res2\_Fri\_2[i]+2\*res1\_Fri\_2[i])/3.0)



1. 线上结果：0.08412928

Result\_02\_21\_3:

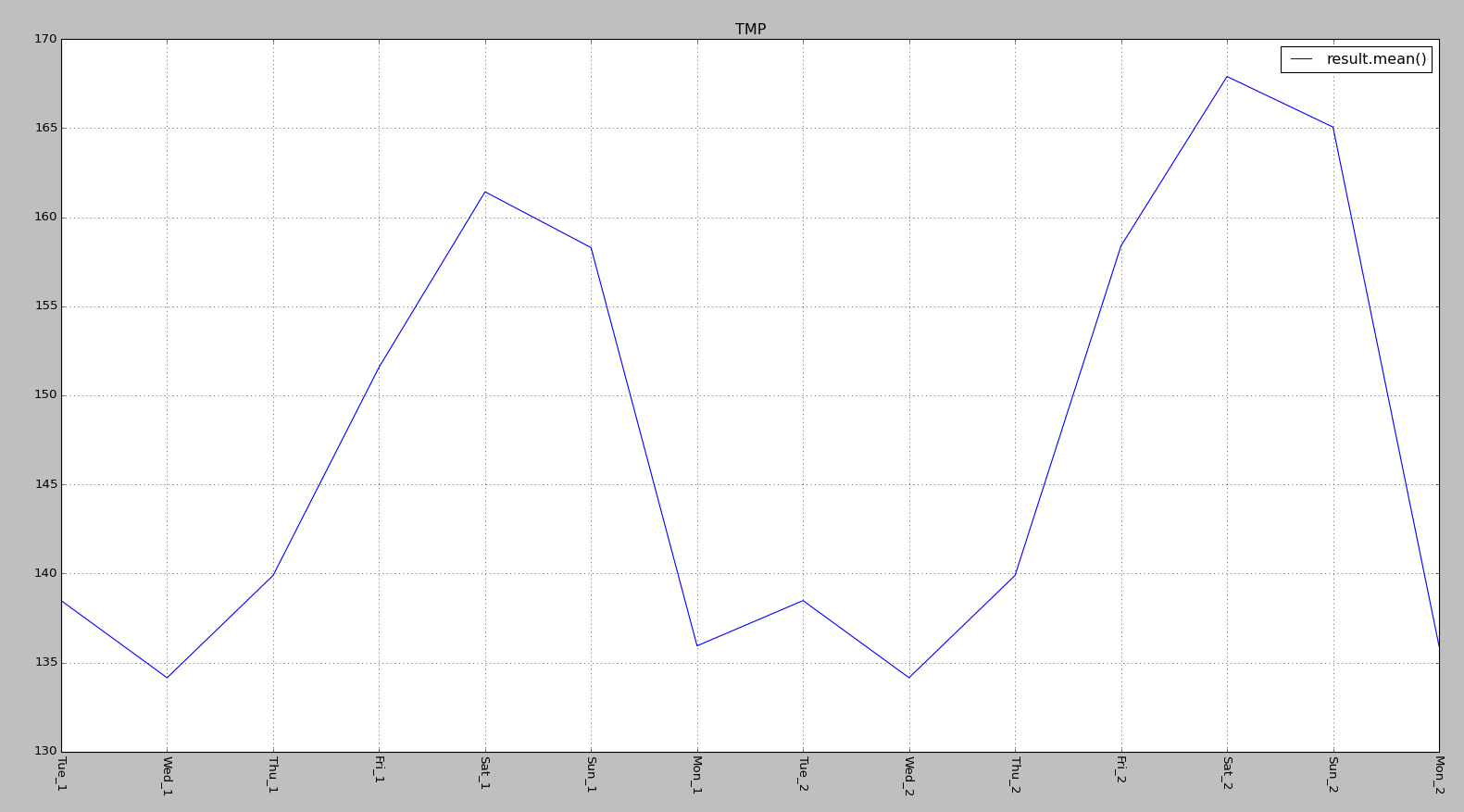
1. result\_name1 = 'result\_02\_20\_3'

result\_name2 = 'result\_02\_21\_1'

1. res1\_Fri\_2[i] = round((res2\_Fri\_2[i]+2\*res1\_Fri\_2[i])/3.0) 阈值<200

res1\_Sat\_2[i] = round((res2\_Sat\_2[i]+res1\_Sat\_2[i])/2.0) 阈值<200 & >100

res1\_Sun\_2[i] = round((res2\_Sun\_2[i]+res1\_Sun\_2[i])/2.0) 阈值<200 & >100

1. 
2. 线上结果: 0.08555694

Result\_02\_22\_1：

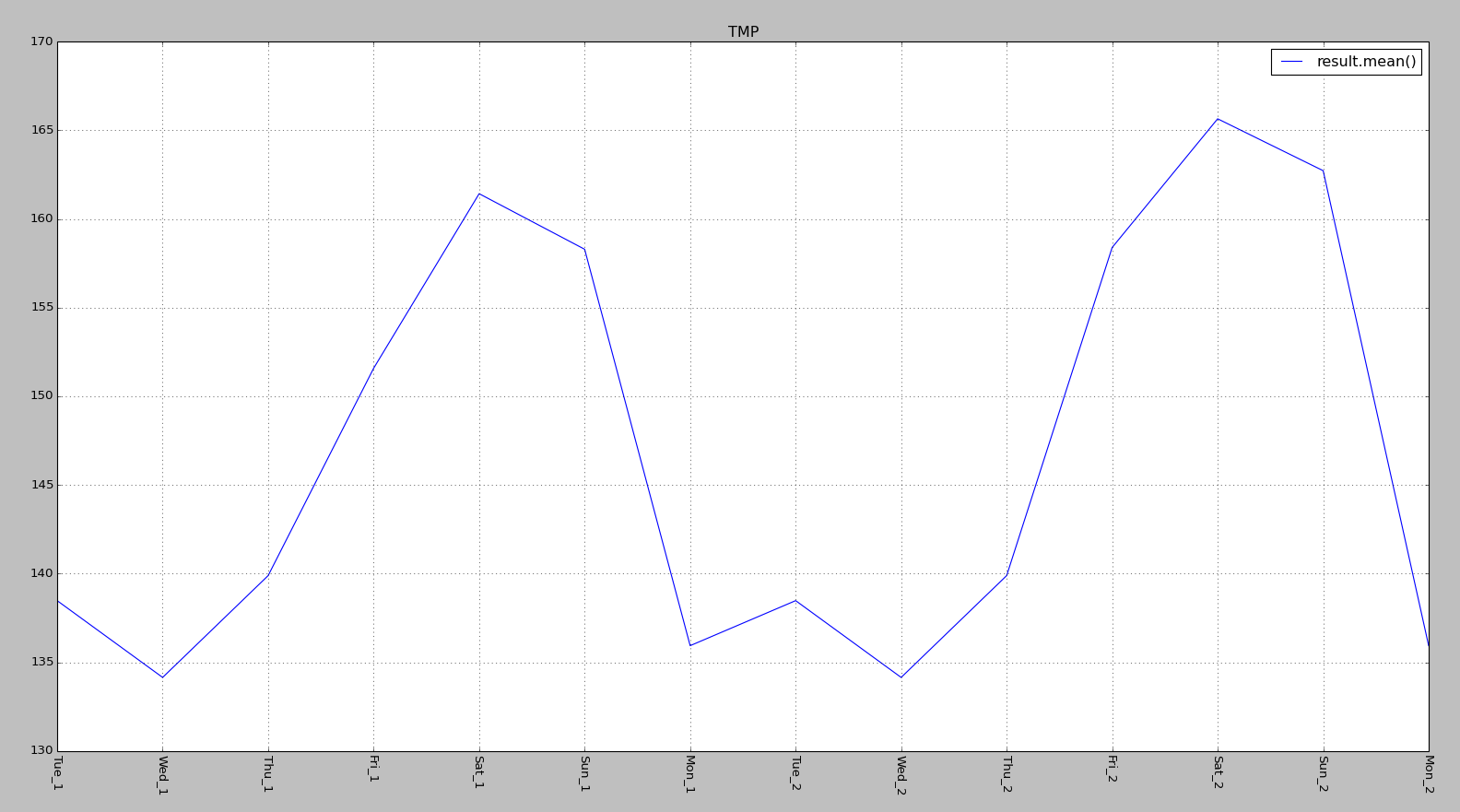
1. result\_name1 = 'result\_02\_20\_3'

result\_name2 = 'result\_02\_21\_1'

1. res1\_Fri\_2[i] = round((res2\_Fri\_2[i]+2\*res1\_Fri\_2[i])/3.0) 阈值<200

res1\_Sat\_2[i] = round((res2\_Sat\_2[i]+2\*res1\_Sat\_2[i])/3.0) 100<res2<200

res1\_Sun\_2[i] = round((res2\_Sun\_2[i]+2\*res1\_Sun\_2[i])/3.0) 100<res2<200

1. 
2. 线上结果: 0.08491331

Result\_02\_22\_2：

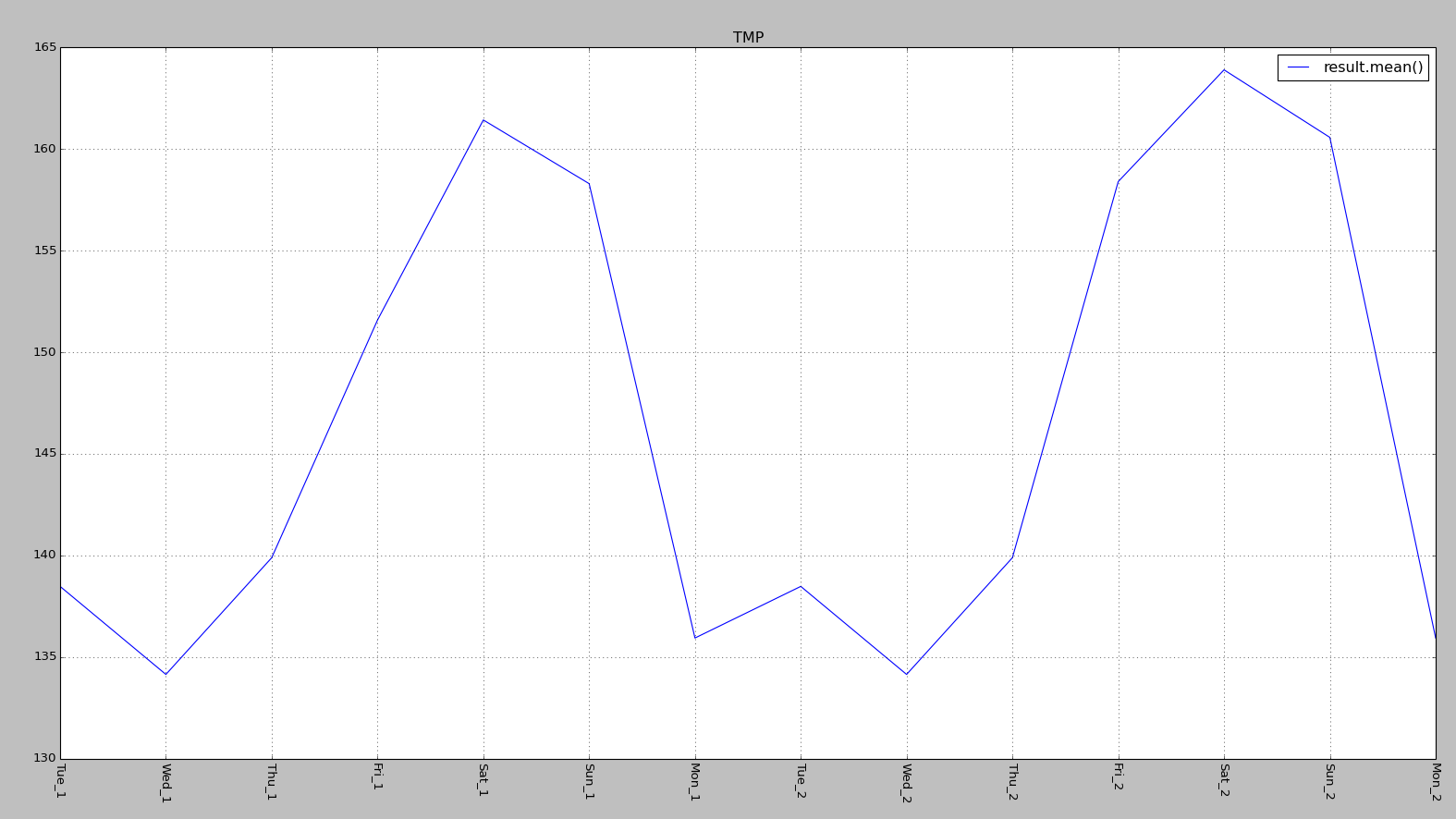
1. result\_name1 = 'result\_02\_20\_3'

result\_name2 = 'result\_02\_21\_1'

1. res1\_Fri\_2[i] = round((res2\_Fri\_2[i]+2\*res1\_Fri\_2[i])/3.0) res2<200

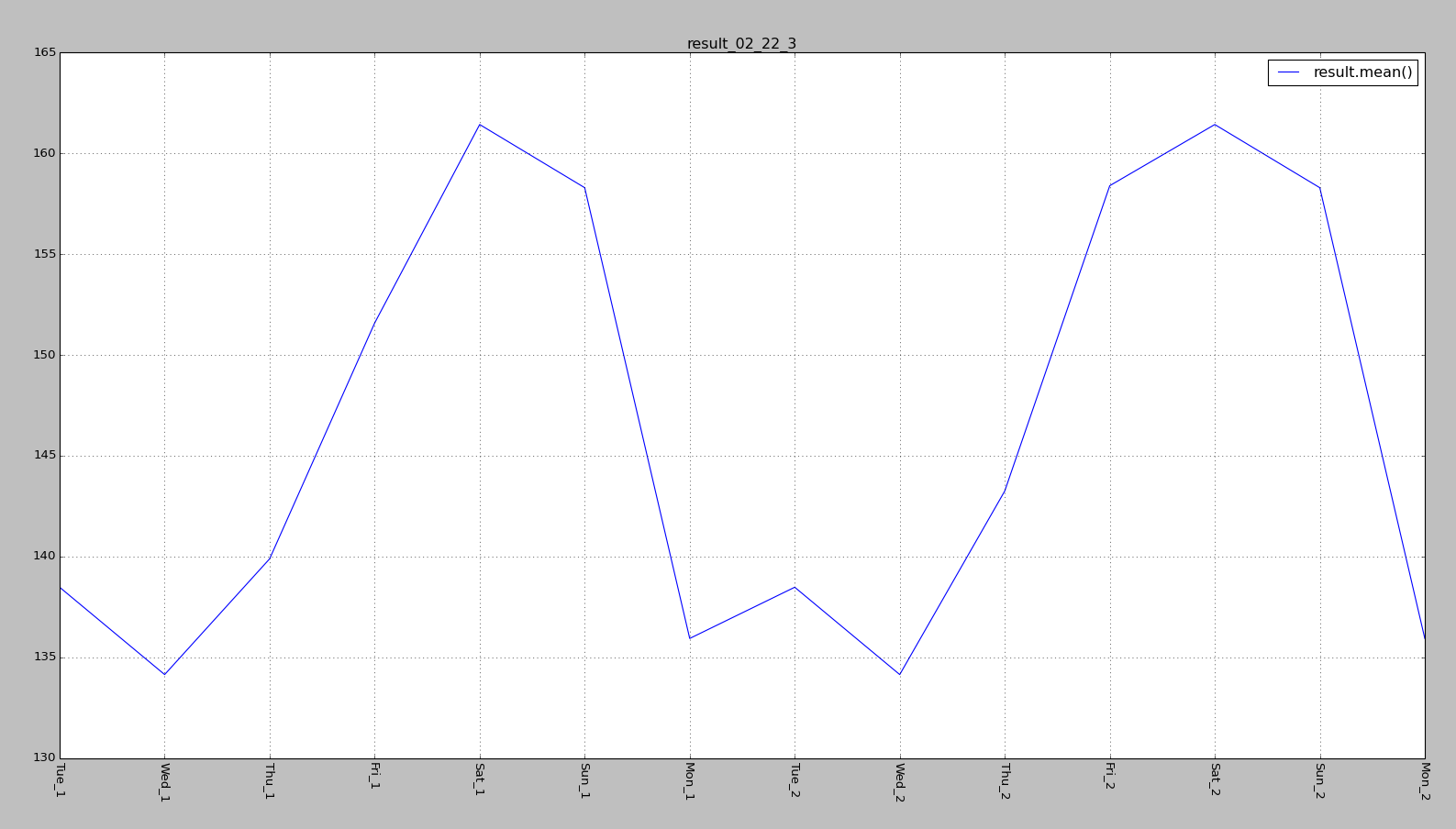
res1\_Sat\_2[i] = round((res2\_Sat\_2[i]+2\*res1\_Sat\_2[i])/3.0) 100<res2<200; res1>100

res1\_Sun\_2[i] = round((res2\_Sun\_2[i]+2\*res1\_Sun\_2[i])/3.0) 100<res2<200; res1>100

1. 
2. 线上结果：？

Result\_02\_22\_3:

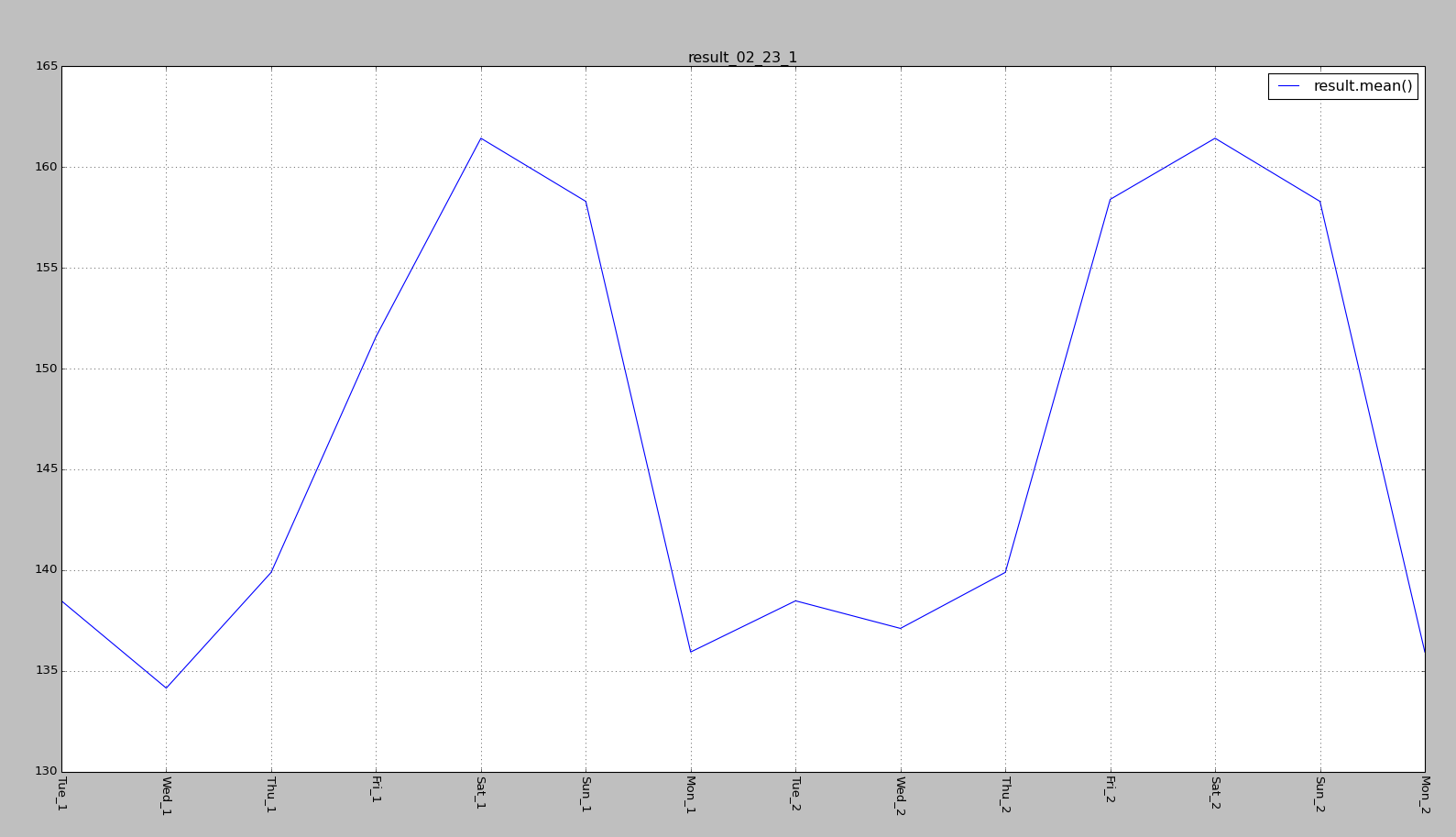
1. res1\_Fri\_2[i] = round((res2\_Fri\_2[i]+2\*res1\_Fri\_2[i])/3.0)
2. res1\_Thu\_2[i] = round((res2\_Thu\_2[i]+2\*res1\_Thu\_2[i])/3.0)



1. 线上结果：0.08422245

Result\_02\_23\_1:

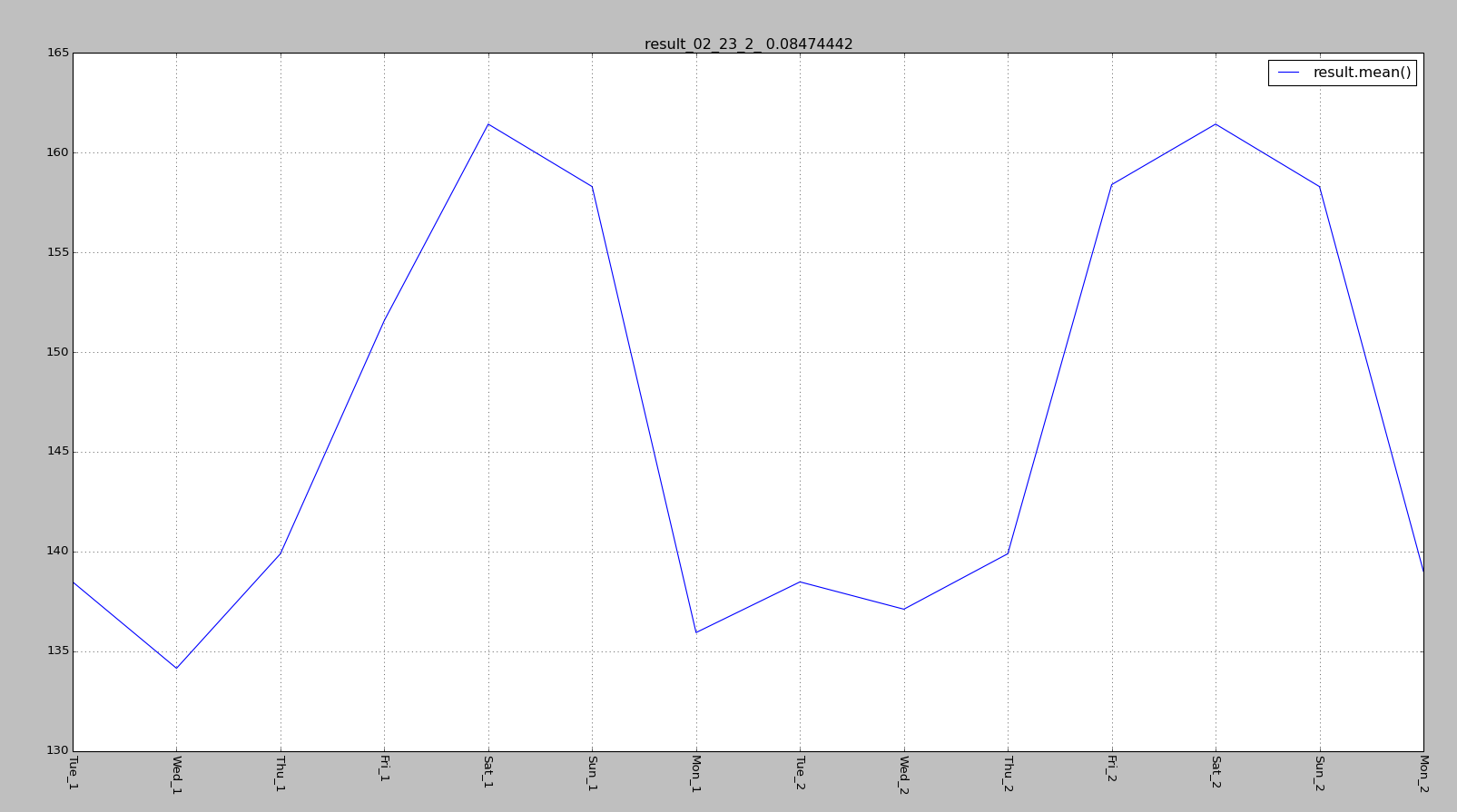
1. res1\_Fri\_2[i] = round((res2\_Fri\_2[i]+2\*res1\_Fri\_2[i])/3.0)
2. res1\_Wed\_2[i] = round((res2\_Wed\_2[i]+2\*res1\_Wed\_2[i])/3.0)



1. 线上结果：0.08405323

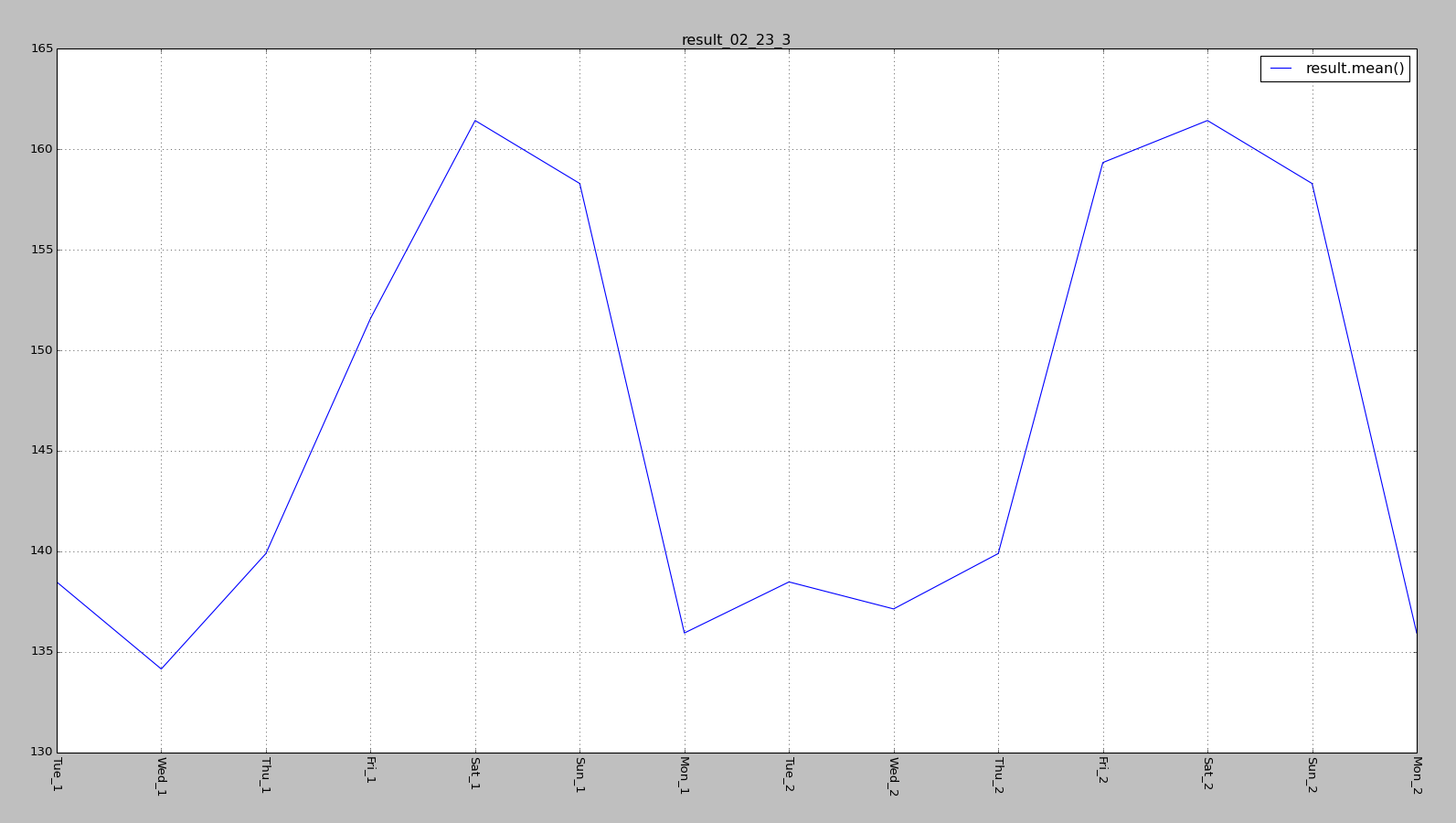
Result\_02\_23\_2：

1. res1\_Fri\_2[i] = round((res2\_Fri\_2[i]+2\*res1\_Fri\_2[i])/3.0)
2. res1\_Wed\_2[i] = round((res2\_Wed\_2[i]+2\*res1\_Wed\_2[i])/3.0)
3. res1\_Mon\_2[i] = round((res2\_Mon\_1[i]+2\*res1\_Mon\_2[i])/3.0) #特殊修改

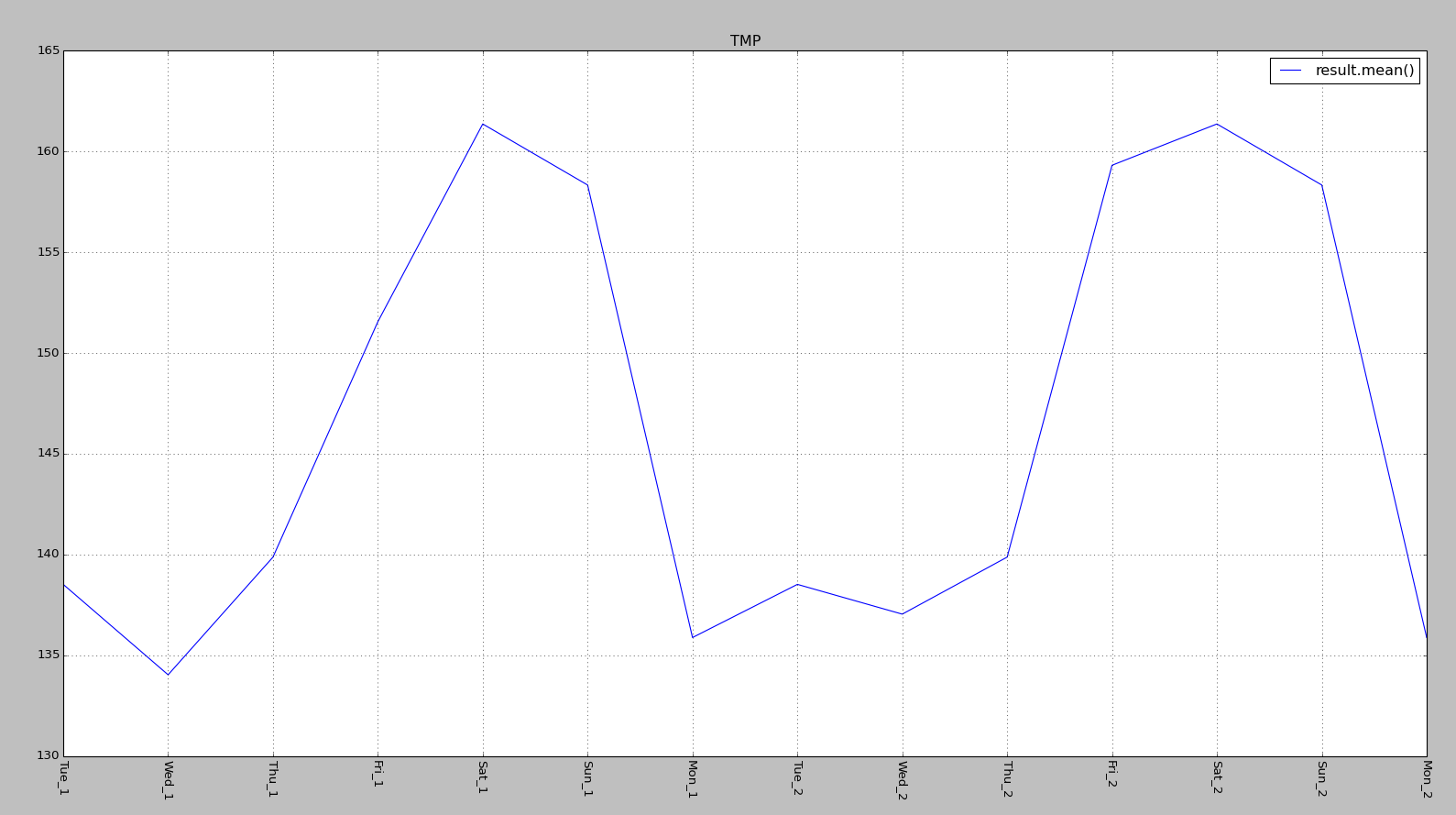


线上结果: 0.08474442

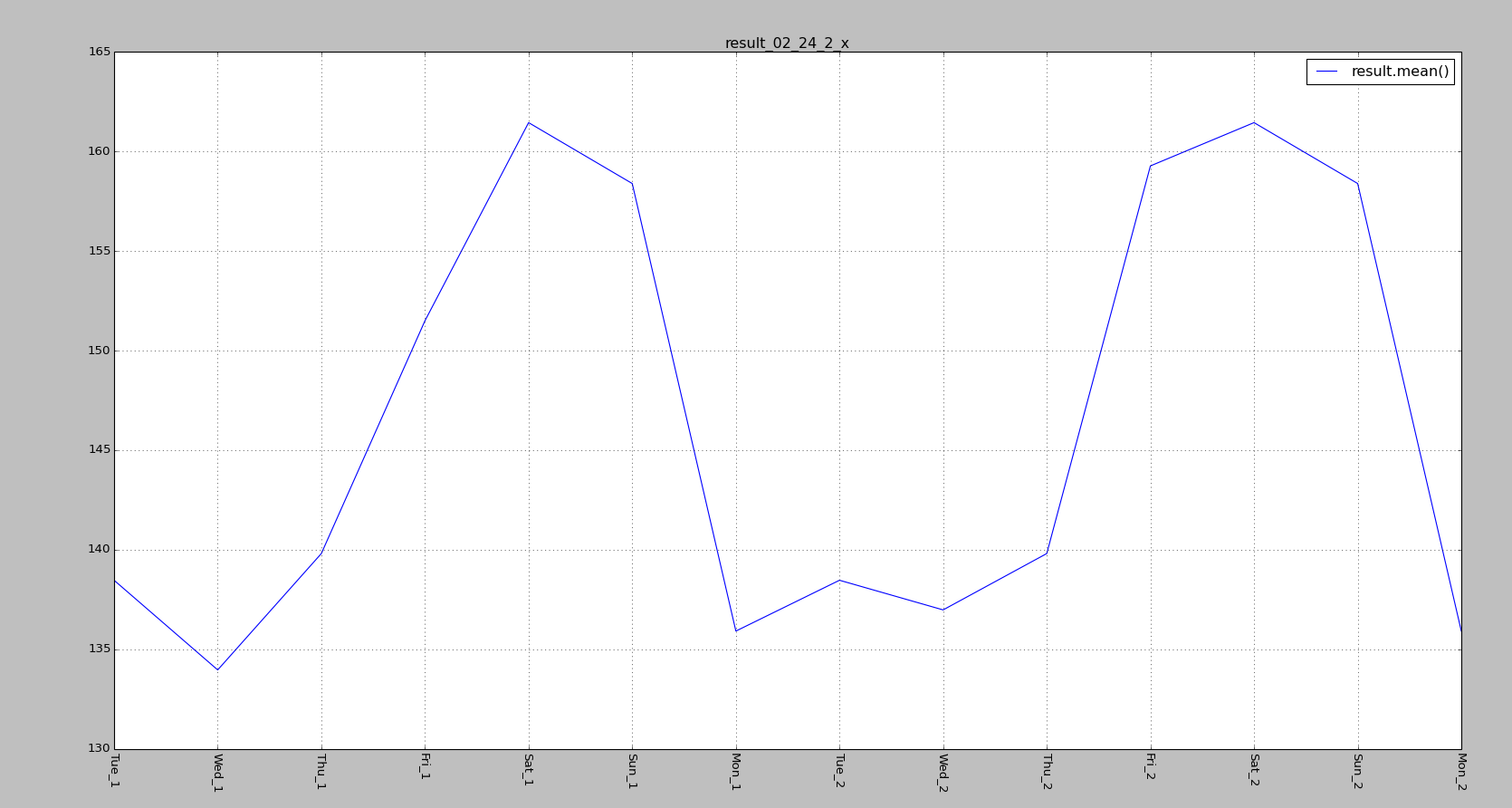
Result\_02\_23\_3：

1. res = combine\_two\_results(labels==1) #labels==1 是周末下降的商家
2. result\_name1 = 'result\_02\_20\_3' #保守算法
3. result\_name2 = 'result\_02\_21\_1' #激进算法
4. 
5. 线上结果：？

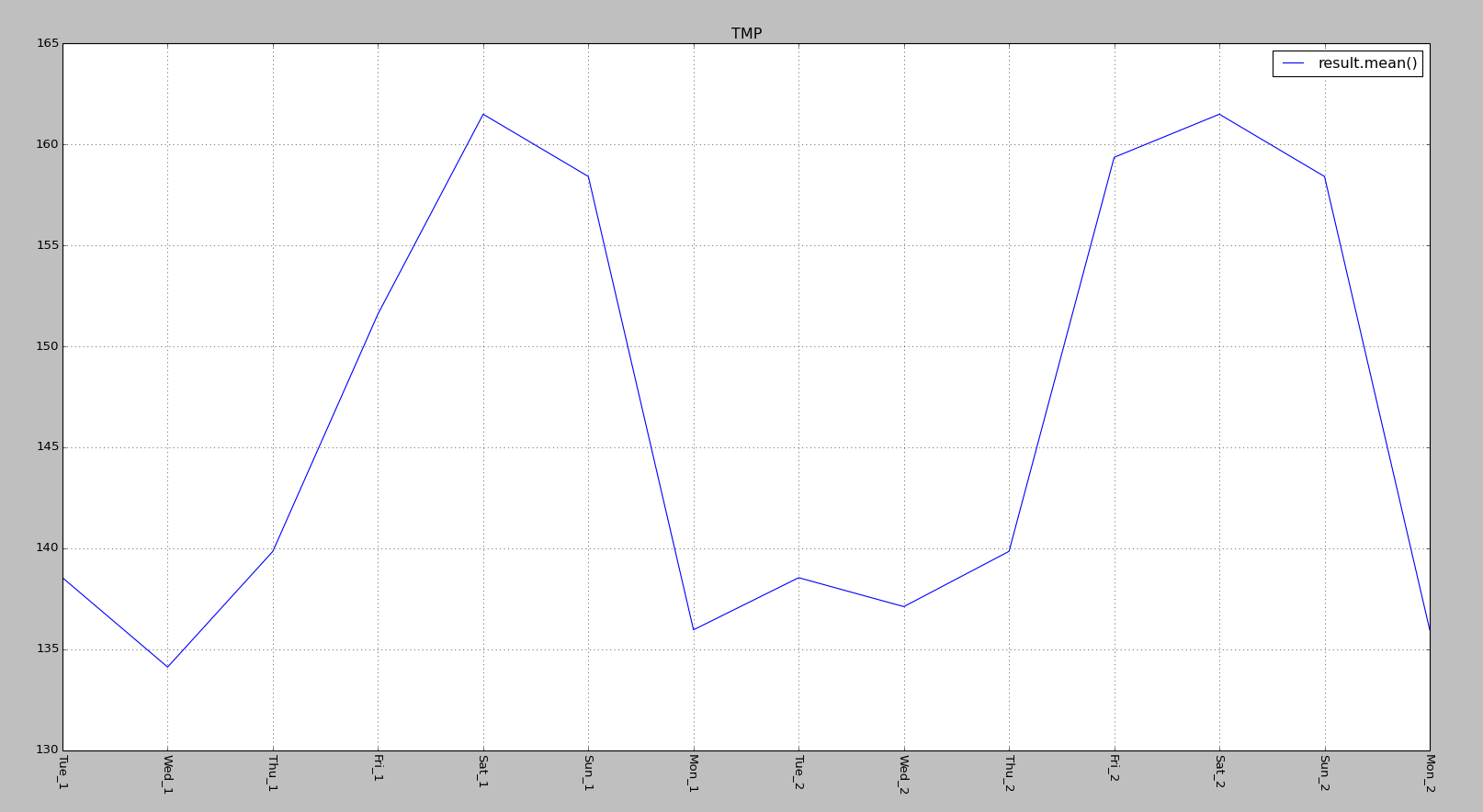
Result\_02\_24\_1\_x：

1. res = combine\_two\_results(labels==1) #labels==1 是周末下降的商家
2. result\_name1 = 'result\_02\_24\_1' #保守算法：添加了7、8两个月的count\_user\_pay
3. result\_name2 = 'result\_02\_21\_1' #激进算法
4. 
5. 线上结果：未提交

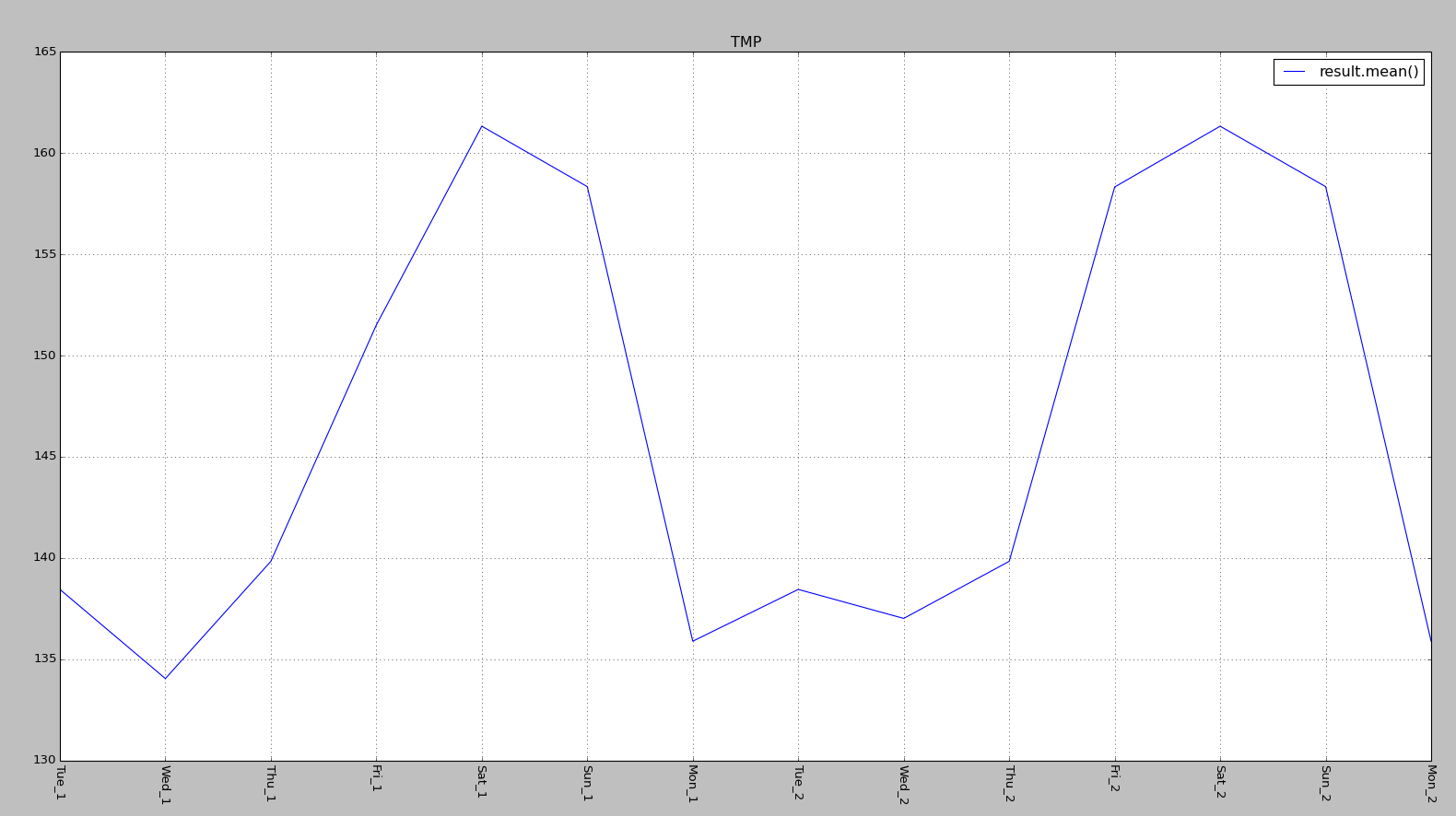
Result\_02\_24\_2\_x：

1. result\_name1 = 'result\_02\_24\_2' #保守算法：添加了7、8两个月add\_July\_Aug；2016年全年按星期的统计，analysis\_all\_week；
2. result\_name2 = 'result\_02\_21\_1' #激进算法
3. res = combine\_two\_results(labels==1) #labels==1 是周末下降的商家
4. 
5. 线上结果：0.08381585

Result\_02\_26\_2\_x:

1. result\_name1 = 'result\_02\_26\_2\_p' #保守算法 此处使用了天气因素
2. result\_name2 = 'result\_02\_21\_1' #激进算法
3. 其余与02\_24\_2\_x相同；
4. 
5. 线上结果：0.08381245

Result\_02\_27\_2\_x：

1. result\_name1 = 'result\_02\_27\_2\_p' #保守算法 此处使用了天气因素,以及改良后的analysis\_all\_week.csv
2. 其余与 02\_26\_2\_x相同。
3. 
4. 线上结果：

Result\_03\_01\_1:

1. 直接用周杰跑的结果
2. 线上结果：0.08304989

Result\_03\_01\_2：

1. 周杰的结果合并激进算法的结果；
2. 线上结果：0.08249906

Result\_03\_01\_3\_zj：

1. 黄李童在周杰的基础上改了特征1；
2. 线上结果：？

Result\_03\_02\_1\_zj：

1. 黄李童在周杰的基础上改了特征2；
2. 线上结果：？

Result\_03\_02\_2\_x:

1. result\_name1 = 'result\_03\_01\_1' #（zj）保守算法
2. result\_name2 = 'result\_03\_02\_2\_pre' #激进算法 (analysis\_2015 产生)
3. 合并了 Wed\_2 和 Fri\_2
4. 线上结果：0.08236296

Rsesult\_03\_02\_3\_x:

1. 合并了 Wed\_2、Fri\_2、Sat\_2、Sun\_2
2. 其余与上述相同；
3. 线上结果：？

Result\_02\_15\_2 & Result\_02\_17\_3混合调参记录

A: Result\_02\_15\_2

B: Result\_02\_17\_3

Fri\_2: (4\*A+1\*B)/5 ~ (3\*A+1\*B)/4

Sat\_2: (1\*A+0\*B)/1 ~ (4\*A+1\*B)/5

Sun\_2: (1\*A+0\*B)/1 ~ (4\*A+1\*B)/5

Mon\_2: (1\*A+0\*B)/1 ~ (9\*A+1\*B)/10