

ABU 量化系统 简介（版本 0.1）

- author = 'BBFamily'
- email = 'bbfamily@126.com'
- weixin = 'aaaabbbuu'

第四部分 理论基础

类似物理有基础理论试验物理，基于基础物理上的应用物理
量化系统中构成因子等信号触发器的基础理论技术分析叫做
基础技术

```
python import ZEnv import ZLog import ZCommonUtil %matplotlib inline
```

```
python import SymbolPd kl_pd = SymbolPd.make_kfold_pd('usNOAH')
```

*vwap n日成交量于n日成交均价除总量，计算出n日均线，以std作为波动信号（类似boll）vwap_mean = (kl_pd.volume * uq_close).sum() / kl_pd.volume.sum()

```
python import TLineVwap
```

```
python TLineVwap.calc_vwap(kl_pd)
```



```
vwap(below=18.653499238200503, vwap=24.158091511147536, above=29.66268378409457)
```

跳空

跳空阈值不是取的固定值根据每个月的波动率以时间加权计算跳空阈值
详情看calc_jump具体实现
计算每个缺口的强度，记录当前调空阈值，方向等信息
jump jump_diff jump_power jump_threshold

```
python import TLineJump reload(TLineJump)
```

```
<module 'TLineJump' from '/Users/Bailey/Desktop/my_work/stock_fact/TLine/TLineJump.py'>
```

```
python jumps = TLineJump.calc_jump(kl_pd, True)
```



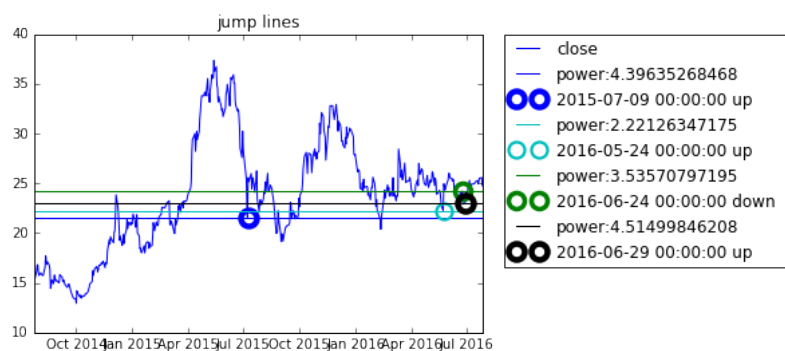
```
python jumps.filter(['jump', 'jump_diff', 'jump_power', 'jump_threshold', 'netChangeRatio'])
```

| | jump | jump_diff | jump_power | jump_threshold | netChangeRatio |
|------------|------|-----------|------------|----------------|----------------|
| 2015-07-06 | -1.0 | 0.917585 | 1.187900 | 3.300667 | -8.27 |
| 2015-07-07 | -1.0 | 0.841670 | 1.770290 | 3.300667 | -9.88 |
| 2015-07-08 | -1.0 | 0.758493 | 1.001987 | 3.300667 | -6.14 |
| 2015-07-09 | 1.0 | 0.711954 | 4.396353 | 3.300667 | 18.45 |
| 2015-07-27 | -1.0 | 0.731796 | 1.434826 | 3.127333 | -5.30 |
| 2015-08-24 | -1.0 | 0.828710 | 1.315297 | 3.717857 | -10.45 |
| 2016-01-04 | -1.0 | 0.745172 | 1.919019 | 2.668000 | -6.41 |
| 2016-03-14 | -1.0 | 0.590735 | 1.134181 | 2.077857 | -5.59 |
| 2016-04-13 | 1.0 | 0.415305 | 1.035384 | 1.678000 | 8.97 |
| 2016-05-24 | 1.0 | 0.324140 | 2.221263 | 1.456154 | 12.08 |
| 2016-06-24 | -1.0 | 0.186667 | 3.535708 | 0.770714 | -5.45 |
| 2016-06-29 | 1.0 | 0.177187 | 4.514998 | 0.770714 | 4.05 |
| 2016-07-06 | -1.0 | 0.194991 | 1.282112 | 0.770714 | -3.64 |

应用举例，fiter出合适做为信号触发器的跳空点

跳口power > 2.0的组成namedtuple jump array 接口不考虑跳空时间权重

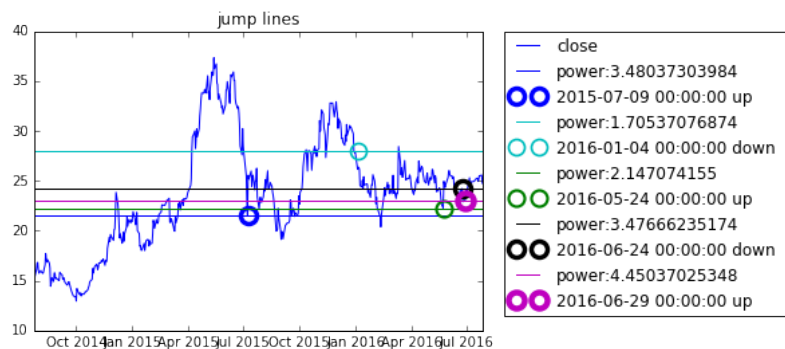
```
python TLineJump.calc_jump_line(kl_pd)
```



```
[jump(date=Timestamp('2015-07-09 00:00:00'), direction=1.0, power=4.3963526846826282, price=21.57),
jump(date=Timestamp('2016-05-24 00:00:00'), direction=1.0, power=2.2212634717493693, price=22.260000000000002),
jump(date=Timestamp('2016-06-24 00:00:00'), direction=-1.0, power=3.535707971950051, price=24.219999999999999),
jump(date=Timestamp('2016-06-29 00:00:00'), direction=1.0, power=4.514998462078653, price=22.989999999999998)]
```

组成namedtuple jump array 接口考虑跳空时间权重:使用线性时间权重分配sw[1]给予的权重

```
python TLineJump.calc_jump_line_weight(kl_pd, sw=(0.6, 0.4))
```



```
[jump(date=Timestamp('2015-07-09 00:00:00'), direction=1.0, power=3.4803730398421919, price=21.57),
jump(date=Timestamp('2016-01-04 00:00:00'), direction=-1.0, power=1.7053707687351682, price=27.93),
jump(date=Timestamp('2016-05-24 00:00:00'), direction=1.0, power=2.1470741549990922, price=22.260000000000002),
jump(date=Timestamp('2016-06-24 00:00:00'), direction=-1.0, power=3.4766623517425348, price=24.219999999999999),
jump(date=Timestamp('2016-06-29 00:00:00'), direction=1.0, power=4.4503702534763345, price=22.989999999999998)]
```

跳空应用2

在某个因子中首先做全量股票n年数据回归, 记录diff_days 上次跳空距离买入的时间, 跳空强度jump_power 做成大量有结果的数据, 之后使用适当的机器学习技术分析拟合数据, 学习成果融于因子中

更多请参看 机器学习部分

```
python last_test_pd = ZCommonUtil.load_hdf5('./data/cache/golden_n6_ma5cross_use_last_test',
'golden_n6_ma5cross_use_last_test')
```

```
python last_test_pd
```

| | buy Date | buy Price | buy Cnt | buyFactor | Sell Date | Sell Price | Sell Type | Symbol | MaxLoss | |
|------------|----------|-----------|---------|-------------------------|-----------|------------|-----------|--------|---------|---|
| 2016-02-08 | 20160208 | 10.6550 | 18770 | BuyGoldenFactorClass:42 | None | NaN | keep | usIRC | 9.88 | {'wave_score':0.283408117, 'wave_sco... |
| 2016-02-10 | 20160210 | 13.4750 | 5106 | BuyGoldenFactorClass:42 | None | NaN | keep | usMW | 7.55 | {'wave_score':0.783895745, 'wave_sco... |
| 2016-05-02 | 20160502 | 26.6685 | 7499 | BuyGoldenFactorClass:42 | None | NaN | keep | usEMQ | 25.95 | {'wave_score':1.039993616, 'wave_sco... |
| 2016-05-18 | 20160518 | 11.2950 | 15326 | BuyGoldenFactorClass:42 | None | NaN | keep | usCRY | 9.79 | {'wave_score':0.696774357, 'wave_sco... |

| | | | | | | | | | | |
|------------|----------|----------|-------|-------------------------|------|-----|------|--------|--------|--|
| 2016-06-09 | 20160609 | 10.2250 | 11506 | BuyGoldenFactorClass:42 | None | NaN | keep | usTWIN | 6.76 | {'wave_score 1.041240815 'wave_scor... |
| 2016-06-20 | 20160620 | 20.9800 | 9532 | BuyGoldenFactorClass:42 | None | NaN | keep | usKKD | 18.61 | {'wave_score 1.072991383 'wave_scor... |
| 2016-07-05 | 20160705 | 2.3300 | 26654 | BuyGoldenFactorClass:42 | None | NaN | keep | usBDSI | -0.44 | {'wave_score 0.446743383 'wave_sco... |
| 2016-07-06 | 20160706 | 37.0800 | 4407 | BuyGoldenFactorClass:42 | None | NaN | keep | usKSS | 30.62 | {'wave_score 0.106391411 'wave_scor... |
| 2016-07-12 | 20160712 | 9.2400 | 9909 | BuyGoldenFactorClass:42 | None | NaN | keep | usSKY | 8.53 | {'wave_score -0.31063416 'wave_sc... |
| 2016-07-13 | 20160713 | 36.3750 | 5498 | BuyGoldenFactorClass:42 | None | NaN | keep | usHRL | 33.96 | {'wave_score 0.688010308 'wave_sco... |
| 2016-07-13 | 20160713 | 27.0800 | 7385 | BuyGoldenFactorClass:42 | None | NaN | keep | usFISI | 24.52 | {'wave_score 1.877227743 'wave_scor... |
| 2016-07-15 | 20160715 | 23.1950 | 8622 | BuyGoldenFactorClass:42 | None | NaN | keep | usMFLX | 19.61 | {'wave_score -0.29543052 'wave_sco... |
| 2016-07-18 | 20160718 | 23.2990 | 7390 | BuyGoldenFactorClass:42 | None | NaN | keep | usHOFT | 20.63 | {'wave_score 0.236341260 'wave_sco... |
| 2016-07-19 | 20160719 | 9.7900 | 20429 | BuyGoldenFactorClass:42 | None | NaN | keep | usUBCP | 9.08 | {'wave_score -0.20766132 'wave_sc... |
| 2016-07-19 | 20160719 | 42.0050 | 3077 | BuyGoldenFactorClass:42 | None | NaN | keep | usCAVM | 31.98 | {'wave_score 1.342210640 'wave_scor... |
| 2016-07-19 | 20160719 | 96.7600 | 2066 | BuyGoldenFactorClass:42 | None | NaN | keep | usERIE | 90.15 | {'wave_score -0.15309654 'wave_sc... |
| 2016-07-19 | 20160719 | 29.4500 | 5999 | BuyGoldenFactorClass:42 | None | NaN | keep | usGBLI | 25.99 | {'wave_score 1.620842317 'wave_scor... |
| 2016-07-20 | 20160720 | 26.1400 | 7651 | BuyGoldenFactorClass:42 | None | NaN | keep | usUBSH | 23.66 | {'wave_score 0.473917409 'wave_scor... |
| 2016-07-20 | 20160720 | 116.1450 | 1721 | BuyGoldenFactorClass:42 | None | NaN | keep | usAET | 105.72 | {'wave_score 0.051839587 'wave_sc... |
| 2016-07-20 | 20160720 | 25.3900 | 7877 | BuyGoldenFactorClass:42 | None | NaN | keep | usSTN | 21.10 | {'wave_score 0.014155204 'wave_sc... |
| 2016-07-21 | 20160721 | 10.3150 | 10605 | BuyGoldenFactorClass:42 | None | NaN | keep | usMSB | 7.58 | {'wave_score -0.34165875 'wave_sc... |
| 2016-07-21 | 20160721 | 35.8500 | 4416 | BuyGoldenFactorClass:42 | None | NaN | keep | usPAG | 30.33 | {'wave_score 1.313956058 'wave_scor... |
| 2016-07-21 | 20160721 | 28.3600 | 5749 | BuyGoldenFactorClass:42 | None | NaN | keep | usCBG | 25.42 | {'wave_score 1.253984026 'wave_scor... |
| 2016-07-21 | 20160721 | 107.2650 | 1455 | BuyGoldenFactorClass:42 | None | NaN | keep | usJLL | 94.02 | {'wave_score 0.930067238 'wave_sco... |
| 2016-07-22 | 20160722 | 62.3600 | 3207 | BuyGoldenFactorClass:42 | None | NaN | keep | usBKH | 57.16 | {'wave_score -0.38295824 'wave_sc... |
| 2016- | | | | | | | | | | {'wave_score 0.241559468 |

[illegible]

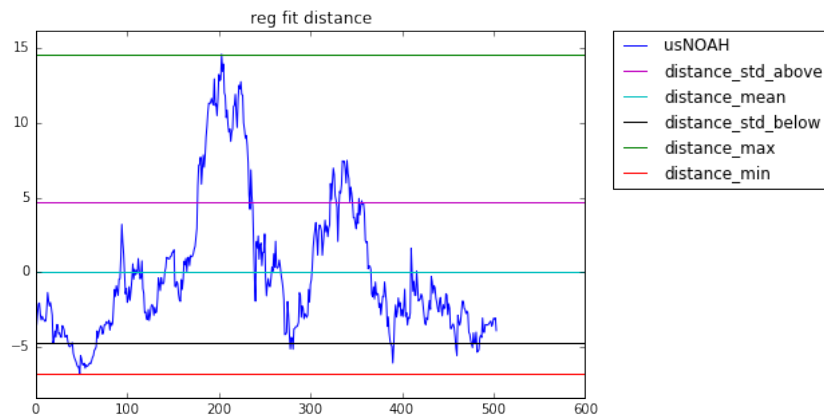
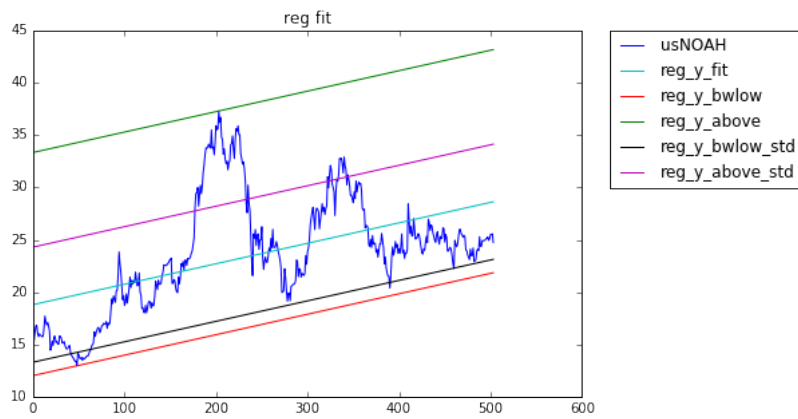
| | | | | | | | | | | |
|------------|----------|---------|-------|-------------------------|----------|---------|------|--------|-------|--|
| 2012-06-25 | 20120625 | 14.5375 | 11924 | BuyGoldenFactorClass:42 | 20120719 | 13.4850 | loss | usPBI | 12.39 | 1.232290120: 'wave_scor... |
| 2012-06-26 | 20120626 | 26.0950 | 7664 | BuyGoldenFactorClass:42 | 20120711 | 26.4600 | win | usEMQ | 25.20 | {'wave_score -0.06529186('wave_s... |
| 2012-06-26 | 20120626 | 8.2750 | 24169 | BuyGoldenFactorClass:42 | 20120706 | 8.6850 | win | usIRC | 7.32 | {'wave_score -0.09937319' 'wave_s... |
| 2012-06-26 | 20120626 | 14.4200 | 13869 | BuyGoldenFactorClass:42 | 20120717 | 14.3100 | loss | usIBKR | 12.83 | {'wave_score 0.187959764: 'wave_sco... |
| 2012-06-26 | 20120626 | 17.6980 | 11300 | BuyGoldenFactorClass:42 | 20120709 | 19.5735 | win | usJAH | 16.08 | {'wave_score -0.139548694 'wave_sc... |
| 2012-06-26 | 20120626 | 46.8200 | 4271 | BuyGoldenFactorClass:42 | 20120709 | 49.7450 | win | usKRC | 43.13 | {'wave_score -0.187959764: 'wave_s... |
| 2012-06-27 | 20120627 | 11.8750 | 16235 | BuyGoldenFactorClass:42 | 20120720 | 11.3145 | loss | usPETS | 9.92 | {'wave_score 0.893804352' 'wave_sco... |
| 2012-06-28 | 20120628 | 5.7225 | 17083 | BuyGoldenFactorClass:42 | 20120713 | 5.5950 | loss | usIMMR | 4.30 | {'wave_score 0.315096904: 'wave_sco... |
| 2012-06-29 | 20120629 | 10.2900 | 8170 | BuyGoldenFactorClass:42 | 20120718 | 8.7750 | loss | usOMER | 9.00 | {'wave_score 0.985505766 'wave_scor... |
| 2012-07-02 | 20120702 | 34.6400 | 4400 | BuyGoldenFactorClass:42 | 20120712 | 30.8500 | loss | usADSK | 29.52 | {'wave_score 0.601655440: 'wave_sco... |
| 2012-07-02 | 20120702 | 15.4300 | 12961 | BuyGoldenFactorClass:42 | 20120712 | 15.9900 | win | usPGC | 13.72 | {'wave_score 0.718892897: 'wave_sco... |
| 2012-07-03 | 20120703 | 12.0100 | 15339 | BuyGoldenFactorClass:42 | 20120720 | 11.4750 | loss | usFPO | 10.22 | {'wave_score 0.065608822 'wave_sc... |
| 2012-07-09 | 20120709 | 22.4500 | 8908 | BuyGoldenFactorClass:42 | 20120718 | 23.5350 | win | usPFX | 20.56 | {'wave_score -0.135719008 'wave_sc... |
| 2012-07-10 | 20120710 | 4.2150 | 27213 | BuyGoldenFactorClass:42 | 20120718 | 4.0450 | loss | usTNK | 2.72 | {'wave_score -0.50024715: 'wave_sco... |

3348 rows × 31 columns

趋势

拟合趋势线寻找支撑阻力

```
python import TLineTrend
python TLineTrend.calc_reg_fit(kl_pd)
```

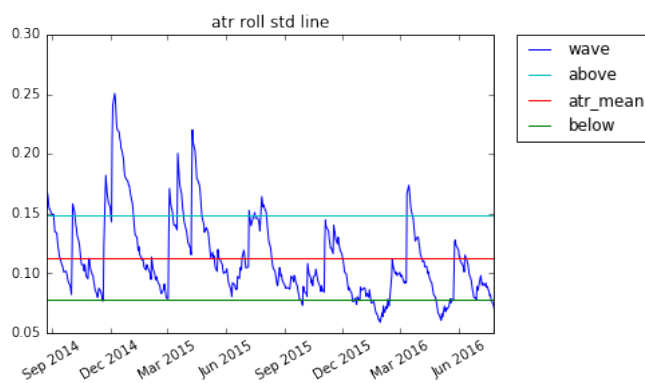


```
reg_fit(now=-3.8868615432971794, mean=3.157967714489334e-15, above=4.7051428025675852, below=-4.7051428025675781,
distance_max=14.552347530407687, distance_min=-6.7852277815114697)
```

atr

```
python import TLineAtr import TLineUtil
```

```
python atr_t = TLineAtr.calc_atr_std(kl_pd) score = TLineUtil.calc_score(atr_t) ZLog.info('art score =
{}'.format(score)) atr_t
```



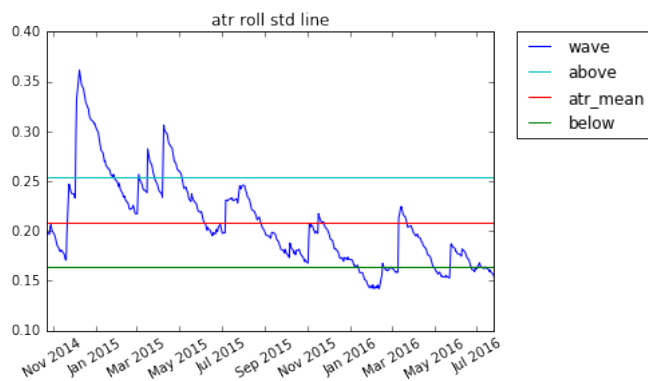
```
art score = -0.109856700631
```

```
atr_roll_std(atr_mean=0.11334196683255461, above=0.14850406592684917, below=0.07817986773826005, now=0.070454283350754349)
```

atr_t score使用TLineUtil计算出的代表一种标准化状态，对atr来说也就是周期内波动程度状态的量化

参数： 1. 加权周期 2. 加权方式（默认指数加权）

```
python atr_t = TLineAtr.calc_atr_std(kl_pd, xd=3*21) score = TLineUtil.calc_score(atr_t) ZLog.info('art score = {}'.format(score)) atr_t
```



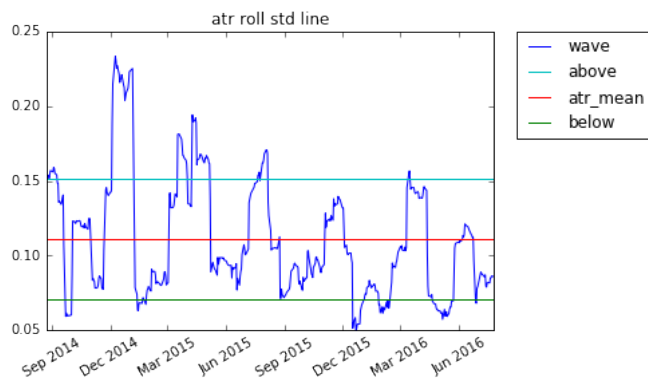
```
art score = -0.109446075174
```

```
atr_roll_std(atr_mean=0.20851545560667786, above=0.2537072958356352, below=0.1633236153777205, now=0.15343147629185575)
```

python

ewm=False 使用线性加权

```
atr_t = TLineAtr.calc_atr_std(kl_pd, ewm=False) score = TLineUtil.calc_score(atr_t) ZLog.info('art score = {}'.format(score)) atr_t
```



```
art score = 0.183535756333
```

```
atr_roll_std(atr_mean=0.11114064745762825, above=0.15113888986835614, below=0.07114240504690036, now=0.085824620392554618)
```

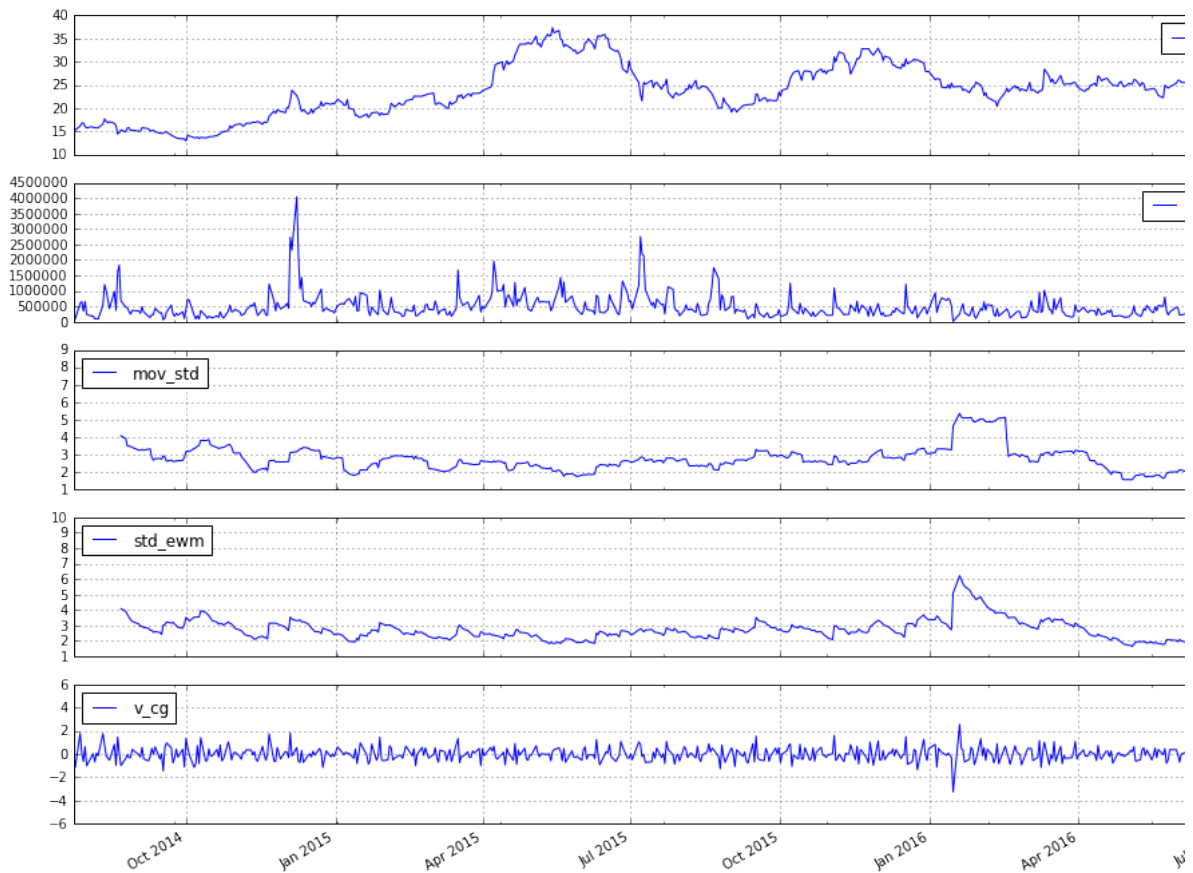
volume

对量的量化由于数据源的问题，导致数据不可信，类似obv，vwap等都不可以实际使用，重点需要解决的问题

```
python import TLineVolume
```

```
python TLineVolume.show_volume_cg(kl_pd, data_proxy=False)
```


min: 1661
mean: 481841.071429

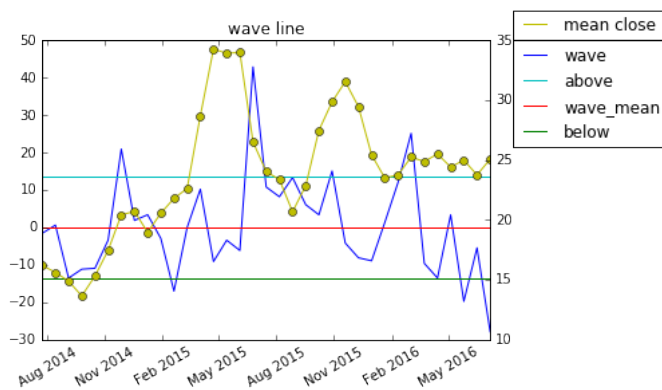


波动程度

```
python import TLineWave
```

abs 不考虑正负, 只考虑波动 与股票的涨跌没有关系, 只是自定义一个指标 量化指定周期内波动程度, 不考虑之前的波动 的时间效应, 🤖 不考虑波动的时间轴上的传播

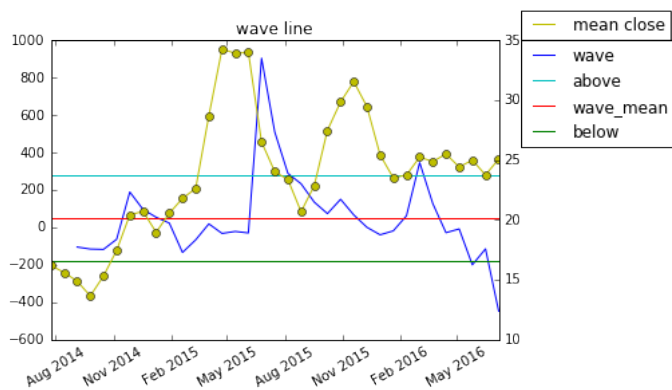
```
python TLineWave.calc_wave_abs(kl_pd)
```



```
wave(wave_mean=-2.0301221021717148e-16, above=13.65844868585747, below=-13.65844868585747, now=-27.970285714285716)
```

abs 不考虑正负, 只考虑波动 考虑波动的传播, 按照时间指数移动加权, 波动的影响

```
python TLineWave.calc_wave_weight_abs(kl_pd)
```



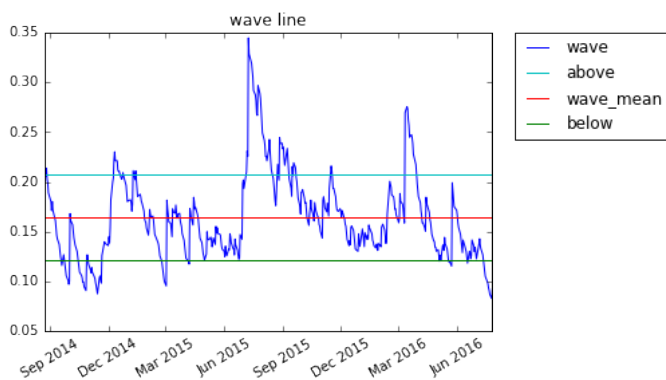
```
wave(wave_mean=51.724993410214715, above=281.5533354798916, below=-178.10334865946217, now=-449.05885687225992)
```

考虑正负，只是按照rolling std计算

```
kl_pd['return'] = np.log(kl_pd['close'] / kl_pd['close'].shift(1))
if ewm:
    kl_pd['roll_std'] = kl_pd['return'].ewm(span=xd, min_periods=xd, adjust=True).std() * np.sqrt(xd)
else:
    kl_pd['roll_std'] = kl_pd['return'].rolling(window=xd, center=False).std() * np.sqrt(xd)

wave_mean = kl_pd['roll_std'].mean()
std = kl_pd['roll_std'].std()
above = wave_mean + std
below = wave_mean - std
```

python TLineWave.calc_wave_std(kl_pd)



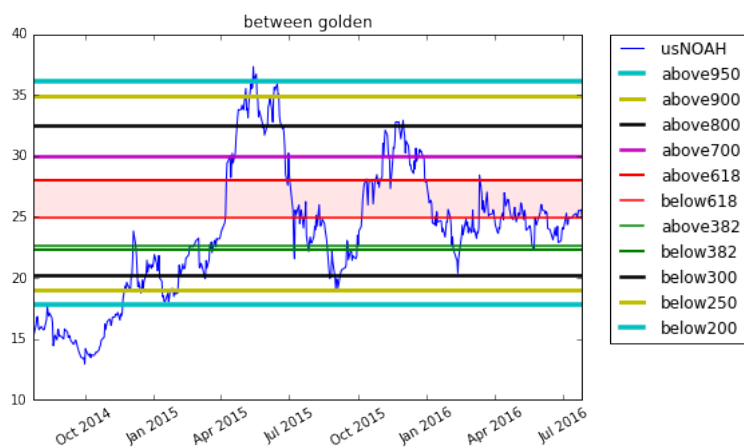
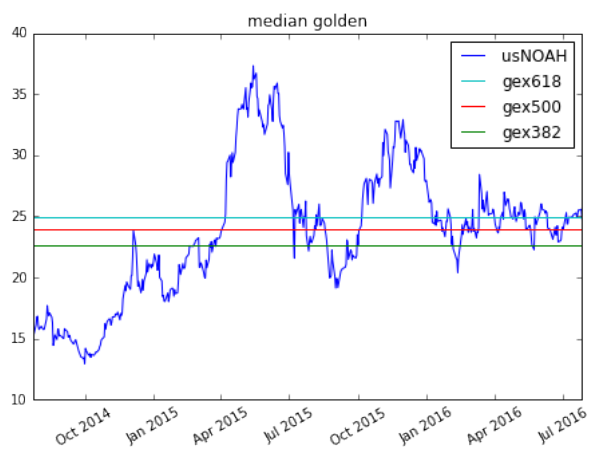
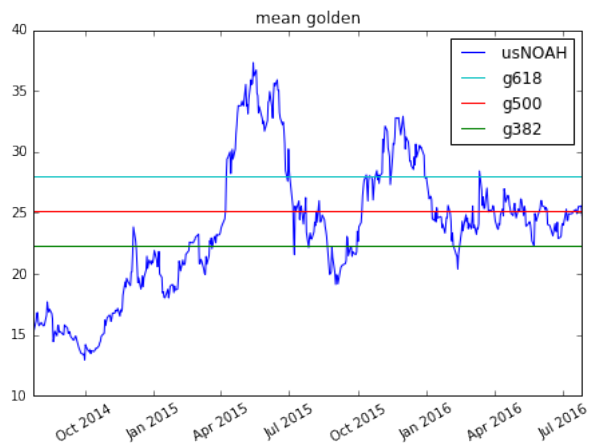
```
wave(wave_mean=0.1640141385837507, above=0.2072127223998287, below=0.12081555476767267, now=0.09241816110003423)
```

黄金分割

综合两种黄金切割方式，根据应用使用
maximum, minimum进行选择

python import TLineGolden

python TLineGolden.calc_golden(kl_pd)

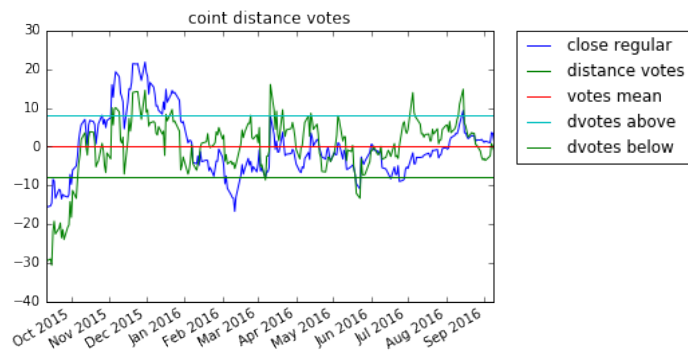
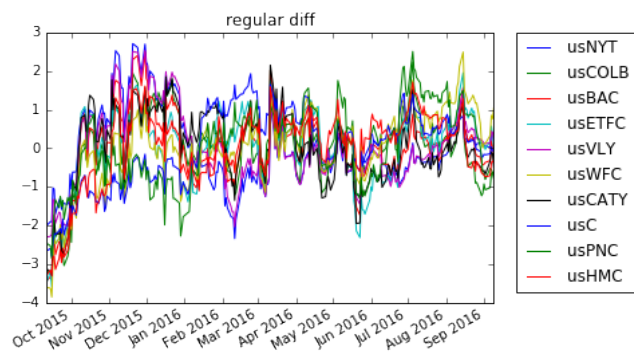
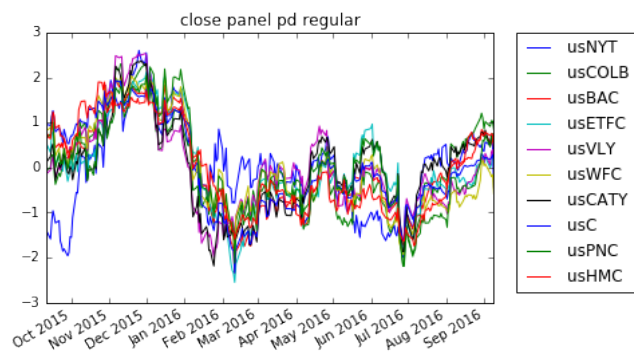


golden(g382=22.259340000000002, gex382=22.664380000000001, g500=25.134999999999998, gex500=23.939999999999998, g618=28.010660000000001, gex618=24.948540000000001, above618=28.010660000000001, below618=24.948540000000001, above382=22.664380000000001, below382=22.259340000000002, above950=36.101500000000001, above900=34.882999999999996, above800=32.445999999999998, above700=30.009, below300=20.260999999999999, below250=19.0425, below200=17.823999999999998)

相关协整

```
python import TLineSimilar
```

```
python TLineSimilar.coint_similar('usNOAH')
```



```
python
'''
```