

# A Simple Application of Pairs Trading Strategy

Qingmin Shi

May 10, 2016

## 1. Introduction

The basic idea of pairs trading is to find two similar financial assets which trade at some spread. If there exists equilibrium between two assets and an anomaly is observed in the relationship, one can seek to profit from the comparative mispricing by selling the relative overvalued asset and simultaneously buying the undervalued asset.

The main theoretical foundation of my project is cointegration. Specifically, suppose we have asset  $X$  and  $Y$ , we need to find weights  $a$  and  $b$ , such that the linear combination of  $X$  and  $Y$  is a stationary noise process. That is to say, we need to make  $aX + bY = \epsilon$ , where  $\epsilon$  is a stationary noise process.

This paper explored several different approaches for cointegration. It is natural to think about linear regression, i.e.,  $Y = aX + \epsilon$ . However, the diagnostic plot indicate that the residuals are not stationary. We also tried AR models and got better performance. Finally, we decide on the Johansen procedure, which uses VAR (vector autoregressive) models instead of regression. Using this method, we can obtain a stationary noise process.

After getting the stationary noise process, we can begin to trade the spread. Firstly, we can normalize the process  $aX + bY$ . Then, we observe the process. When its value is below 1.5 standard error, we build a long position. Once its value is within 0.3 standard error, we cover the long position we built. When its value is above 1.5 standard error, we build a short position. Once its value is within 0.3 standard error, we cover the short position we built.

The above is the main framework of the project. We can see all the details in the following contents.

## 2. Data Description

Our choice of the stock pair is AT&T and Verizon, starting from May 1st in 2008, to April 1st in 2011. We use the daily adjusted closing price, so we have totally 737 prices for each stock. Denote log price of AT&T as  $X$ , log price of Verizon as  $Y$ , we can see that  $X$  is a  $737 \times 1$  vector,  $Y$  is also a  $737 \times 1$  vector. We can see from the plots that these two stocks do have similar trend of prices. The correlation coefficient is 0.82.

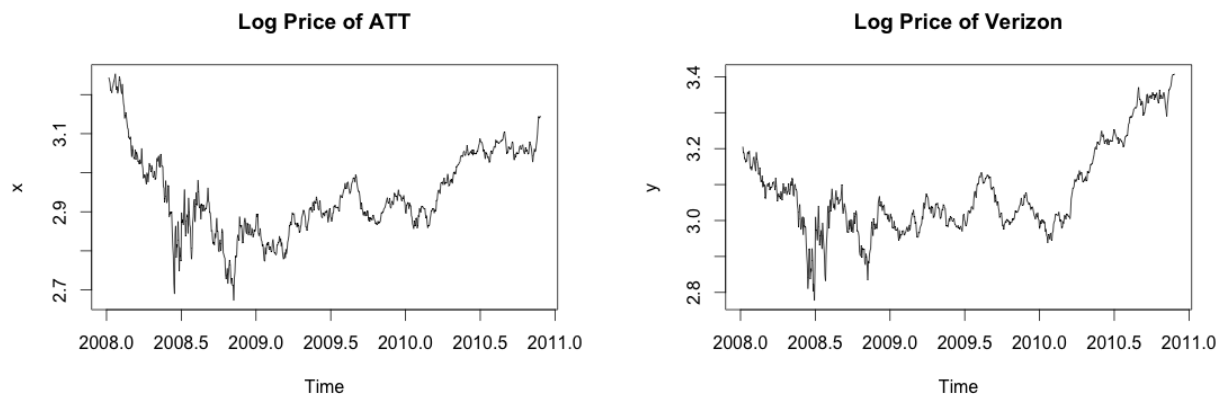


Figure 1: Price trend of AT&T and Verizon

### 3. Cointegration

#### 3.1 The linear regression approach and AR(3) approach

First, we consider the simple linear regression model. The residual plot shows the residual series has certain patterns and it is not stationary. To further confirm it, we plot the ACF, which shows that it decays very slowly. So we fit the AR(3) model to the residual series. The coefficients are not significant. Otherwise, although the ADF test shows that the residuals of the AR(3) model is stationary, the ACF and PACF plot are not clean. After we plot the ACF and PACF of the squared of residual, it is obvious to tell that there is ARCH effect. Thus, the simple linear regression model with the AR(3) model is not satisfying. So we decide to use the Johansen procedure to fit the series.

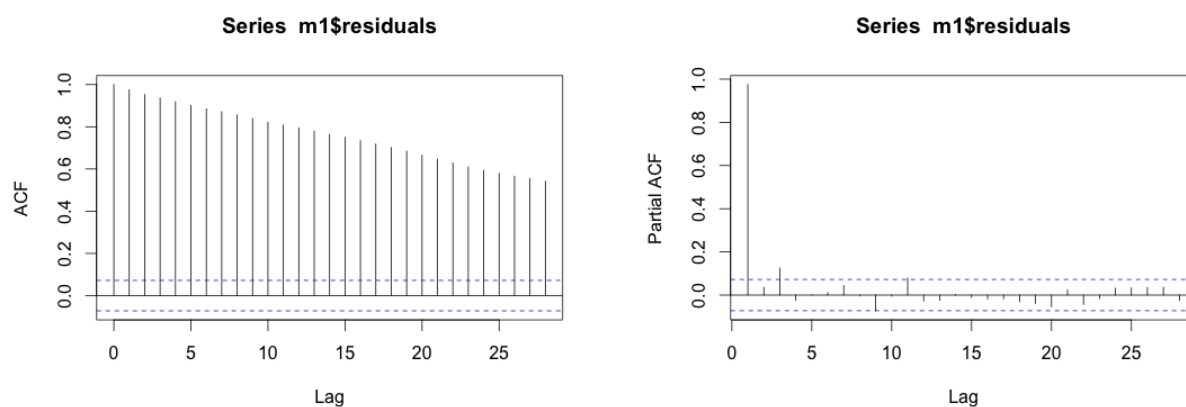


Figure 2: ACF and PACF of linear residuals indicating it is not a good model

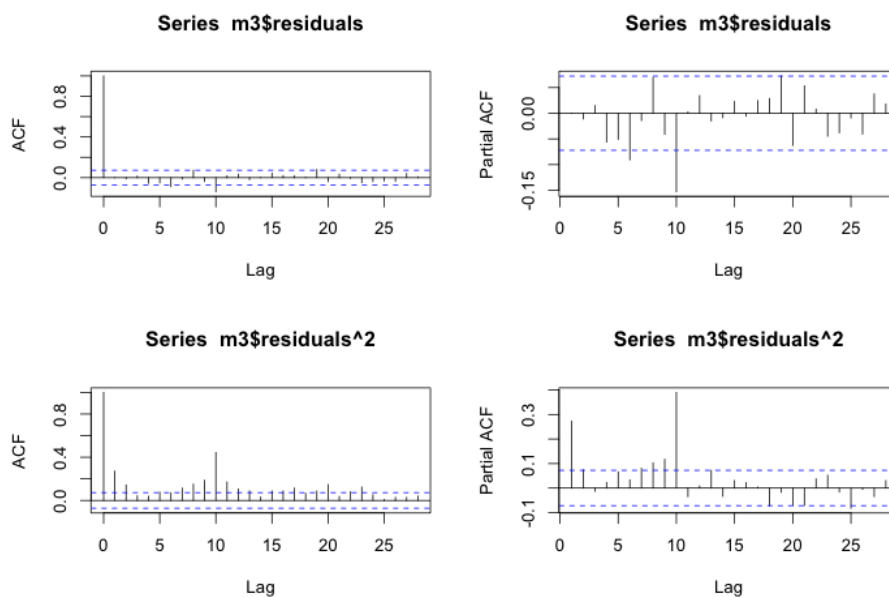


Figure 3: ACF and PACF of AR(3) squared residuals indicating it is not a good model

### 3.2 The Johansen procedure

```
#####
# Johansen-Procedure #
#####

Test type: trace statistic , with linear trend

Eigenvalues (lambda):
[1] 0.4902120 0.1794881

Values of teststatistic and critical values of test:

      test 10pct 5pct 1pct
r <= 1 | 5.54 6.50 8.18 11.65
r = 0  | 24.40 15.66 17.95 23.52

Eigenvectors, normalised to first column:
(These are the cointegration relations)

      ATTad.l3 Verad.l3
ATTad.l3 1.0000000 1.00000
Verad.l3 -0.3394999 12.10922

Weights W:
(This is the loading matrix)

      ATTad.l3 Verad.l3
ATTad.d -0.9852927 -0.01696224
Verad.d 0.4193060 -0.02548269
```

Figure 4: Results of the Johansen procedure

We can see from results of Johansen procedure that the estimated cointegration coefficient  $\gamma$  is -0.34.

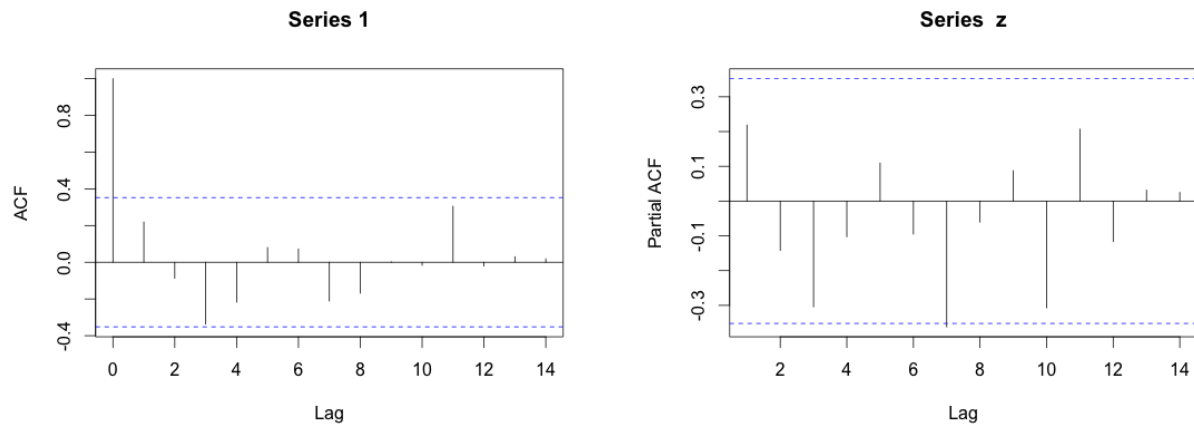


Figure 5: ACF and PACF of the residuals generated by Johansen procedure

We can see that this model is satisfactory. We can treat the residuals as a stationary noise process.

#### 4. Trading Strategy and Application

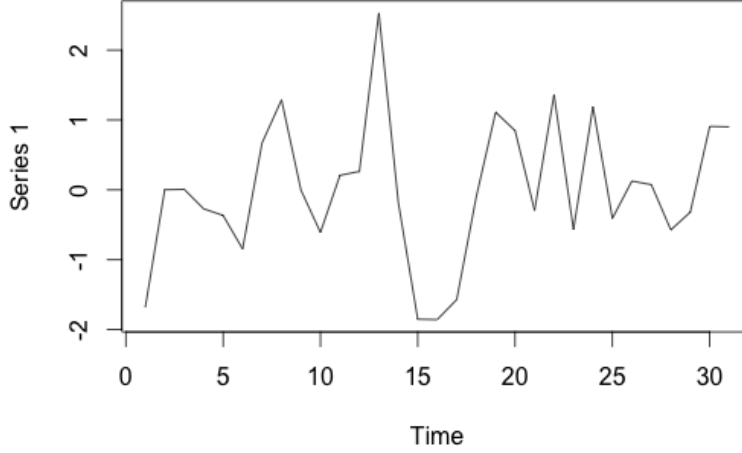


Figure 6: A glance of the normalized residuals

##### 4.1 The fixed gamma approach

Using fixed gamma approach, our trading strategy can be expressed as follows. Firstly, we run Johansen procedure from day 1 to day 30 and obtain gamma. Then, in the next 30 trading days, i.e., day 31 to day 60, we use this fixed gamma to build the stationary noise process.

##### 4.2 The floating gamma approach

Using floating gamma approach, we use gamma computed from the most recent 30 days and update our gamma every 10 days or 5 days. For example, if we update gamma every 10 days, the gamma used in days 31-40 is computed from days 1-30, the gamma used in days 41-50 is computed from days 11 to 40, and the gamma used in days 51-60 is computed from days 21-50.

##### 4.3 Trading Strategy and Performance Comparation

Our trading process is as follows: Go over the residual process day by day. When the value of the residual is below 1.5 standard error, we build a long position. Once its value is within 0.3 standard error, we cover the long position we built. When its value is above 1.5 standard error, we build a short position. Once its value is within 0.3 standard error, we cover the short position we built.

Backtesting performances of our trading strategies are as follows:

**Apply the Strategy to the next 30 Days  
Using Fixed Gamma**

	<b>31-60 Days</b>
<b>Gamma</b>	<b>-0.3394999</b>
<b>Gain</b>	<b>0.06663566</b>

**Apply the Strategy to the next 30 Days  
Using Floating Gamma  
(Changing Every 10 Days)**

	<b>31-40 days</b>	<b>41-50 days</b>	<b>51-60 days</b>
<b>Gamma</b>	-0.3394999	-1.420094	-1.558607
<b>Gain</b>	0.3372988	-0.3185177	0.401776
<b>Total Gain</b>	0.4205571		

**Apply the Strategy to the next 30 Days  
Using Floating Gamma  
(Changing Every 5 Days)**

	<b>31-35 days</b>	<b>36-40 days</b>	<b>41-45 days</b>	<b>46-50 days</b>	<b>51-55 days</b>	<b>56-60 days</b>
<b>Gamma</b>	0.3395	0.6975	1.4201	1.3287	1.5586	1.8208
<b>Gain</b>	0.3728	0.0367	0.0445	0.0579	0.4987	0.5359
<b>Total Gain</b>	1.268230477					

We can see clearly that the floating gamma approach has a better performance than fixed gamma approach. We can conclude that updating gamma every 5 trading days, and each gamma generating from the previous 30 days of price data is a good choice.

## Code

```

1 library(tseries)
2 library(forecast)
3 library(urca)
4 ###Data Selection
5 ATT=ATT <- read.csv("~/Desktop/ts project/ATT.csv", sep=";")
6 Ver=Verizon <- read.csv("~/Desktop/ts project/Verizon.csv", sep="")
7 par(mfrow=c(1,1))
8 x=ts(log(ATT[85:821,]),start=c(2008,5,1),freq=255)
9 plot(x,main='Log Price of ATT')
10 y=ts(log(Ver[85:821,]),start=c(2008,5,1),freq=255)
11 plot(y,main='Log Price of Verizon')
12 cor(log(ATT[85:821,]),log(Ver[85:821,]))
13
14 ###Simple Linear Regression with AR(3) Model
15 m1=lm(log(ATT[85:821,]) ~ log(Ver[85:821,]))
16 summary(m1)
17 wt=m1$residuals
18 plot(ts(wt),start=c(2008,5,1),freq=255)
19 acf(wt)
20 pacf(wt)
21 m3=arima(wt,order=c(3,0,0),include.mean = F)
22 m3
23 summary(m3)
24 adf.test(m3$residuals)
25 par(mfrow=c(2,2))
26 acf(m3$residuals)
27 acf(m3$residuals^2)
28 pacf(m3$residuals)
29 pacf(m3$residuals^2)
30 par(mfrow=c(1,1))
31
32
33 ###Cointegration
34 ATTad=log(ATT[684:714,])
35 Verad=log(Ver[684:714,])
36 xx=ts(ATTad)
37 plot(xx,main='Selected Log Price of ATT')
38 yy=ts(Verad)
39 plot(yy,main='Selected Log Price of Verizon')
40 xxy=cbind(ATTad,Verad)
41 mx=ar(xxy)
42 mx$order
43 cot=ca.jo(xxy, ecdet='none', type='trace', K=3, spec='longrun')
44 summary(cot)
45 cot@V[2,1] #gamma
46
47 ###Strategy Application
48 zz=cbind(xx,yy)
49 weight=c(1,cot@V[2,1])
50 z=ts(zz*weight)
51 plot(z)
52 acf(z)

```

```
53 pacf(z)
54 sd(z)
55 mean(z)
56 abs((z-mean(z))/sd(z))
57 plot((z-mean(z))/sd(z))
58
59 ###Fixed Gamma
60 spread = (z-mean(z))/sd(z)
61 longpos = rep(0,31)
62 i = 1
63 while(i != 31)
64 {
65   if(spread[i]<-1.5)
66   {
67     longpos[i] = 1
68     for(j in (i+1):31)
69     {
70       if((spread[j]>-0.3)|| j == 31)
71         break
72     }
73     longpos[j] = -1
74     i = j + 1
75   }
76   else
77   {
78     i = i + 1
79   }
80 }
81 shortpos = rep(0,31)
82 i = 1
83 while(i != 31)
84 {
85   if(spread[i] > 1.5)
86   {
87     shortpos[i] = 1
88     for(j in (i+1):31)
89     {
90       if((spread[j]<0.3)|| j == 31)
91         break
92     }
93     shortpos[j] = -1
94     i = j + 1
95   }
96   else
97   {
98     i = i + 1
99   }
100 }
101
102 gain1 = 0
103 buy=c(0,0)
104 sell=c(0,0)
105 for(i in 1:31)
106 {
```



```

107   if(longpos[i]==1)
108   {
109       buy = c(ATT[714+i,],Ver[714+i,])
110   }
111   if(longpos[i] == -1)
112   {
113       sell = c(ATT[714+i,],Ver[714+i,])
114       gain1=gain1+c(1,-Ver[714,]^((cot@V[2,1]-1))%*(sell-buy)
115   }
116 }
117
118 gain2 = 0
119 buy=c(0,0)
120 sell=c(0,0)
121 for(i in 1:31)
122 {
123     if(shortpos[i]==1)
124     {
125         sell = c(ATT[714+i,],Ver[714+i,])
126     }
127     if(shortpos[i] == -1)
128     {
129         buy = c(ATT[714+i,],Ver[714+i,])
130         gain2=gain2+c(1,-Ver[714,]^((cot@V[2,1]-1))%*(sell-buy)
131     }
132 }
133
134 gain = gain1 + gain2
135 gain
136
137 #####Floating Gamma(Changing Every 5 Days)
138 #####For simplification, I only post the calculation of the first two gains.
139 #####The method of using floating gamma changing every 10 days to calculate the gain is the same.
140 #####First Gamma
141 ATTad=log(ATT[684:714,])
142 Verad=log(Ver[684:714,])
143 xx=ts(ATTad)
144 plot(xx,main='Selected Log Price of ATT')
145 yy=ts(Verad)
146 plot(yy,main='Selected Log Price of Verizon')
147 xxy=cbind(ATTad,Verad)
148 mx=ar(xxy)
149 mx$order
150 cot=ca.jo(xxy, ecdet='none', type='trace', K=3, spec='longrun')
151 summary(cot)
152 cot@V[2,1] #gamma
153 zz=cbind(xx,yy)
154 weight=c(1,cot@V[2,1])
155 z=ts(zz%*%weight)
156 plot(z)
157 acf(z)
158 pacf(z)
159 sd(z)
160 mean(z)

```

```

161 abs((z-mean(z))/sd(z))
162 plot((z-mean(z))/sd(z))
163
164 spread = (z-mean(z))/sd(z)
165 longpos = rep(0,6)
166 i = 1
167 while(i != 6)
168 {
169   if(spread[i]<-1.5)
170   {
171     longpos[i] = 1
172     for(j in (i+1):6)
173     {
174       if((spread[j]>-0.3)|| j == 6)
175         break
176     }
177     longpos[j] = -1
178     i = j + 1
179   }
180   else
181   {
182     i = i + 1
183   }
184 }
185 shortpos = rep(0,6)
186 i = 1
187 while(i != 6)
188 {
189   if(spread[i] > 1.5)
190   {
191     shortpos[i] = 1
192     for(j in (i+1):6)
193     {
194       if((spread[j]<0.3)|| j == 6)
195         break
196     }
197     shortpos[j] = -1
198     i = j + 1
199   }
200   else
201   {
202     i = i + 1
203   }
204 }
205
206 gain1 = 0
207 buy=c(0,0)
208 sell=c(0,0)
209 for(i in 1:6)
210 {
211   if(longpos[i]==1)
212   {
213     buy = c(ATT[714+i,],Ver[714+i,])
214   }

```

```

215   if(longpos[i] == -1)
216   {
217       sell = c(ATT[714+i,],Ver[714+i,])
218       gain1=gain1+c(1,-Ver[714,]^(cot@V[2,1]-1))%*(sell-buy)
219   }
220 }
221
222 gain2 = 0
223 buy=c(0,0)
224 sell=c(0,0)
225 for(i in 1:6)
226 {
227     if(shortpos[i]==1)
228     {
229         sell = c(ATT[714+i,],Ver[714+i,])
230     }
231     if(shortpos[i] == -1)
232     {
233         buy = c(ATT[714+i,],Ver[714+i,])
234         gain2=gain2+c(1,-Ver[714,]^(cot@V[2,1]-1))%*(sell-buy)
235     }
236 }
237
238 gain = gain1 + gain2
239 gain
240
241 ####Second Gamma
242 ATTad=log(ATT[689:719,])
243 Verad=log(Ver[689:719,])
244 xx=ts(ATTad)
245 plot(xx,main='Selected Log Price of ATT')
246 yy=ts(Verad)
247 plot(yy,main='Selected Log Price of Verizon')
248 xxy=cbind(ATTad,Verad)
249 mx=ar(xxy)
250 mx$order
251 cot=ca.jo(xxy, ecdet='none', type='trace', K=3, spec='longrun')
252 summary(cot)
253 cot@V[2,1] #gamma
254
255 spread = (z-mean(z))/sd(z)
256 longpos = rep(0,6)
257 i = 1
258 while(i != 6)
259 {
260     if(spread[i]<-1.5)
261     {
262         longpos[i] = 1
263         for(j in (i+1):6)
264         {
265             if((spread[j]>-0.3)|| j == 6)
266                 break
267         }
268         longpos[j] = -1

```

```

269     i = j + 1
270 }
271 else
272 {
273     i = i + 1
274 }
275 }
276 shortpos = rep(0,6)
277 i = 1
278 while(i != 6)
279 {
280     if(spread[i] > 1.5)
281     {
282         shortpos[i] = 1
283         for(j in (i+1):6)
284         {
285             if((spread[j]<0.3)|| j == 6)
286                 break
287         }
288         shortpos[j] = -1
289         i = j + 1
290     }
291     else
292     {
293         i = i + 1
294     }
295 }
296
297 gain1 = 0
298 buy=c(0,0)
299 sell=c(0,0)
300 for(i in 1:6)
301 {
302     if(longpos[i]==1)
303     {
304         buy = c(ATT[719+i,],Ver[719+i,])
305     }
306     if(longpos[i] == -1)
307     {
308         sell = c(ATT[719+i,],Ver[719+i,])
309         gain1=gain1+c(1,-Ver[719,]^((cot@V[2,1]-1))%*(sell-buy)
310     }
311 }
312
313 gain2 = 0
314 buy=c(0,0)
315 sell=c(0,0)
316 for(i in 1:6)
317 {
318     if(shortpos[i]==1)
319     {
320         sell = c(ATT[719+i,],Ver[719+i,])
321     }
322     if(shortpos[i] == -1)

```

```
323 {
324     buy = c(ATT[719+i,],Ver[719+i,])
325     gain2=gain2+c(1,-Ver[719,]^(cot@V[2,1]-1))*%(sell-buy)
326 }
327 }
328
329 gain = gain1 + gain2
330 gain
```