

PairsTrading.m

El objetivo de esta clase-proyecto es implementar la estrategia Pairs Trading con datos de Pepsi y Coca Cola.

Contents

- Rendimientos históricos.
- Loop Principal
- Estimar alpha y beta
- Proceso OU
- Reversion time
- Implementación de la estrategia
- Calcular P&L .

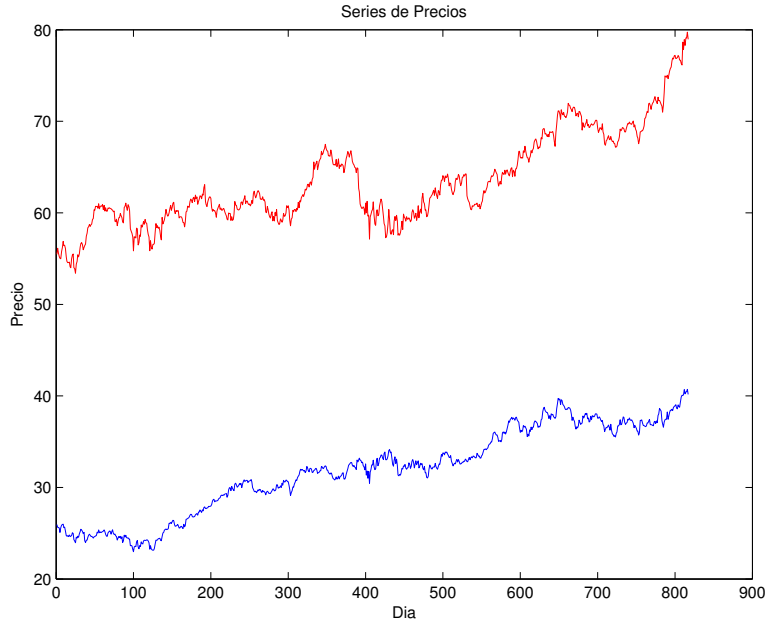
Rendimientos históricos.

Primero limpiamos todo en Matlab e importamos los precios de los activos. El precio del activo l en el día i se denota S_i^l . Nótese que $i = 1, 2, \dots, 817$ y $l = 1, 2$. Vamos a denotar por N el número de precios que tenemos por cada acción.

```
clear all
close all
clc
format long

S = load('Pairs_HistPrices.csv');
[N,m] = size(S);

figure
plot([1:N],S(:,1))
hold on
plot([1:N],S(:,2),'red')
title('Series de Precios')
xlabel('Dia')
ylabel('Precio')
```



A continuación calculamos los rendimientos diarios de los activos y los guardamos en la matriz R (una activo en cada columna, un día en cada fila) utilizando la fórmula

$$R_i^l = \frac{S_{i+1}^l - S_i^l}{S_i^l} \quad i = 1, \dots, 816; \quad l = 1, 2.$$

El número de días en los que se tiene registro de los rendimientos es $N - 1$.

```
R = (S(2:end,:) - S(1:end-1,:)) ./ S(1:end-1,:);
```

La estrategia tiene una ventana de días en los que se se estiman diversos parámetros. El número de días en la ventana se denota con la variable `windowSize`. El intervalo de tiempo entre observaciones consecutivas de precios se denota por dt (en años).

```
windowSize = 63 ;  
dt = 1/252 ;
```

Un porcentaje del portafolio se va invertir libre de riesgo a una tasa anual r .

```
r = 0.01;
```

La estrategia consiste en abrir una posición cuando el *score* es igual a `openStd` y cerrar la posición cuando es igual a `closeStd`.

```
openStd = 2.00 ;  
closeStd = 0.5 ;
```

En cada ventana debemos decidir si el tiempo que esperamos tener abierta la posición es suficientemente corto. Para esto introducimos un tiempo de cierre `diasCierre` (en días).

```
diasCierre = 30 ;
```

Decidiremos si el modelo que tenemos es confiable de acuerdo a una prueba de hipótesis, para lo cual guardamos el valor de a tal que $P(-a < Z < a) = 0,95$ en la variable `conf95`.

```
conf95 = norminv(1.95/2,0,1) ;
```

El portafolio que vamos a considerar va tener una posición larga/corta en una acción de S_1 . Asumiremos que al principio de la simulación no tenemos ninguna acción de S_1 en el portafolio.

```
LongS1(1)=0 ;  
ShortS1(1)=0 ;
```

El portafolio se va a financiar con una Money Market Account, la cual asumiremos vacía al empezar la simulación.

```
MMA(1) = 0 ;
```

Guardaremos el número de acciones de cada activo en la matriz `position`. Una columna por activo y una fila por ventana. Dado que al iniciar la simulación no tenemos ningún activo en el portafolio, el valor inicial del portafolio es cero.

```
position = zeros(1,2) ;  
portfolioValue(1) = 0 ;
```

Loop Principal

El siguiente loop es sobre las ventanas en las que se estiman diversos parámetros

```
for w = 1:N-windowSize
```

Estimar alpha y beta

En cada ventana vamos a estimar los parámetros α y β de el modelo lineal

$$R_i^1 = \alpha dt + \beta R_i^2$$

```
x = R(w:w+windowSize-1,2);
X = [ones(size(x)) x];
y = R(w:w+windowSize-1,1);
[Coefficients,bint] = regress(y,X);
f=@(z) [ones(size(z)) z]*Coefficients;
alpha(w) = Coefficients(1)/dt ;
beta(w) = Coefficients(2);
```

Ahora calculamos los residuos de la regresión lineal que acabamos de calcular.

$$\varepsilon_i = R_i^1 - (\alpha dt + \beta R_i^2)$$

```
residual = y - f(x);
```

Proceso OU

La suma cumulativa de los residuos es el proceso

$$X_i = \varepsilon_1 + \varepsilon_2 + \cdots + \varepsilon_i .$$

La idea principal del presente método es que este es un proceso estacionario que se regresa a su media . En el código vamos a utilizar la variable `OU` para denotar este proceso estacionario.

```
OU = cumsum(residual);
```

El proceso X_i se va a modelar como un proceso Ornstein-Uhlenbeck, es decir

$$X_{i+1} - X_i = k(m - X_i)dt + \sigma\sqrt{dt}Z_i .$$

En el modelo anterior las variables Z_i tienen una distribución normal estándar y son independientes. Para poder estimar los parámetros k, m y σ escribimos el modelo anterior como una proceso Auto Regresivo con lag 1, es decir un proceso AR(1).

$$X_{i+1} = a + bX_i + \xi_i .$$

Las constantes a, b se calculan fácilmente con una regresión lineal.

```

x = OU(1:end-1);
X = [ones(size(x)) x];
y = OU(2:end);
[Coefficients,bint] = regress(y,X);
a(w) = Coefficients(1);
b(w) = Coefficients(2);
f=@(z) [ones(size(z)) z]*Coefficients;
xi = y - f(x);

```

Con esta información podemos calcular

$$k = \frac{1-b}{dt}$$

$$m = \frac{a}{1-b}$$

$$\sigma^2 = Var(\xi_i)/dt$$

```

k(w) = (1 - b(w))*(1/dt);
m(w) = a(w)/(1-b(w));
sigma(w) = sqrt(var(xi)/dt);

```

Podemos verificar que la correlación entre errores sea pequeña. Para esto utilizaremos que la correlación muestral tiene una distribución aproximadamente normal con media zero y varianza $1/L$ donde L es el número de muestras utilizadas.

```

Correlation(w) = corr( xi(1:end-1) , xi(2:end)) ;
upconf = conf95/sqrt(length(xi(1:end-1)));

```

Reversion time

Ahora que tenemos una idea de que tan buena es la regresión lineal que acabamos de aplicar, debemos cuantificar que tan *mean reverting* es el proceso X_i . Una medida de esta propiedad es el *reversion time* dado por la ecuación

$$\tau = \frac{1}{k dt} : .$$

```

reversionTime(w) = (1/k(w))/dt;

```

El proceso X_i es estacionario y por lo tanto podemos calcular su desviación estándar en equilibrio. El resultado es

$$\sigma_{eq}^2 = Var(\xi_i)/(1-b^2) : .$$

```
sigma_eq(w) = sqrt(var(xi)/(1-b(w)*b(w)));
```

El score es la normalización de la distancia del proceso X_i a su media m

```
Score(w) = ( OU(end) - m(w) )/sigma_eq(w);
```

Decidir si el proceso es "Mean Reverting" los errores ξ_i son ruido. Guardamos el resultado en la variable `confianzaModelo`

```
confianzaModelo(w) = (reversionTime(w) < diasCierre).*( abs(Correlation(w)) < upconf );
```

Implementación de la estrategia

A continuación implementamos la estrategia "pairs trading"

1. Verificar si alguna posición esta abierta y necesita cerrarse.

```
if ( LongS1(w) > 0.5 )
    if ( or( Score(w) > -closeStd , ~confianzaModelo(w) ) )
        LongS1(w+1) = 0 ;
    else
        LongS1(w+1) = 1 ;
    end
end

if ( ShortS1(w) > 0.5 )
    if ( or ( Score(w) < closeStd , ~confianzaModelo(w) ) )
        ShortS1(w+1) = 0 ;
    else
        ShortS1(w+1) = 1 ;
    end
end
```

2. Verificar si alguna posición esta cerrada y necesita abrirse.

```
if ( LongS1(w) < 0.5 )
    if ( (Score(w) < -openStd) && confianzaModelo(w) )
        LongS1(w+1) = 1 ;
    else
        LongS1(w+1) = 0 ;
    end
end

if ( ShortS1(w) < 0.5 )
    if ( (Score(w) > openStd) && confianzaModelo(w) )
```

```

        ShortS1(w+1) = 1 ;
    else
        ShortS1(w+1) = 0 ;
    end
end
end

```

Calcular P&L .

1. Calcular el rendimiento del portafolio durante el día anterior

```

PairsReturnDollars(w)=(position(w,:).*S(w + windowSize - 1 ,:))*(R(w + windowSize - 1,:))

```

2. Calcular el nuevo valor del portafolio.

```

portfolioValue(w+1) = portfolioValue(w) + PairsReturnDollars(w) + MMA(w)*r*dt;

```

3. Calcular la nueva posición.

```

position(w+1,1) = LongS1(w+1)-ShortS1(w+1) ;
position(w+1,2) = -beta(w)*position(w+1,1) ;

```

4. Ajustar el dinero en la MMA de acuerdo al cambio de posición.

```

MMA(w+1) = portfolioValue(w+1)-position(w+1,:)*(S( w + windowSize,:))';

```

```

end

```

```

figure
plot(m)
title('Media del proceso OU')
xlabel('ventana')
ylabel('m')
figure
plot(Score)
title('Score para entrada/salida')
xlabel('ventana')
ylabel('s')
hold on
plot(1:length(Score),openStd*ones(length(Score),1),'green')
hold on
plot(1:length(Score),-openStd*ones(length(Score),1),'green')
hold on
plot(1:length(Score),closeStd*ones(length(Score),1),'red')
hold on
plot(1:length(Score),-closeStd*ones(length(Score),1),'red')

```

```

figure
plot(position(:,1))
title('Posicion en la primera accion')

%-----
figure
subplot(5,1,1)
plot(position(:,1))
grid on
title('Posicion en la primera accion')

subplot(5,1,2)
plot(portfolioValue)
grid on
title('P&L del portafolio Pairs Trading')

subplot(5,1,3)
plot(Score)
grid on
title('Score para entrada/salida')
hold on
plot(1:length(Score),openStd*ones(length(Score),1),'green')
hold on
plot(1:length(Score),-openStd*ones(length(Score),1),'green')
hold on
plot(1:length(Score),closeStd*ones(length(Score),1),'red')
hold on
plot(1:length(Score),-closeStd*ones(length(Score),1),'red')

subplot(5,1,4)
plot(reversionTime)
grid on
title('Tiempo de regreso a la media')
xlabel('ventana')
ylabel('días')

subplot(5,1,5)
plot(Correlation)
xlabel('ventana')
grid on
title('One-step residual correlation for the AR(1) fit')
hold on
plot(1:length(Correlation),upconf,'red')
hold on
plot(1:length(Correlation),-upconf,'red')

```



```
figure
plot(alpha)
xlabel('ventana')
title('alpha time series')
```

```
figure
plot(beta)
ylabel('ventana')
title('beta time series')
```

%-----

