

Vivek_FE570_HW4.R

viveksathyanarayana

Wed May 8 18:37:12 2019

```
#Vivek Sathyanarayana
#FE 570 Spring 2019
#HW4 Problem 1
setwd("~/Desktop/Stevens SEM 2/FE 570- Market Microstructure and Trading Strategies/HW 4")

library("stats")
library("tseries")
library("dplyr")

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

#Read file from text file
dataCVX <- read.csv("CVX.csv", head=T)
dataXOM <- read.csv("XOM.csv", head=T)

#Create the Log return vector for CVX
y<-vector(mode="numeric",length=(nrow(dataCVX)-1))
for(i in 1:(nrow(dataCVX)-1))
{
  y[i]<-log(dataCVX$Close[i+1])-log(dataCVX$Close[i])
}

#Create the Log return vector for XOM
x<-vector(mode="numeric",length=(nrow(dataXOM)-1))
for(i in 1:(nrow(dataXOM)-1))
{
  x[i]<-log(dataXOM$Close[i+1])-log(dataXOM$Close[i])
}

#Part (1)
#Linear Regression
```

```

fit <- lm(y~x)
summary(fit)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.045283 -0.004507  0.000057  0.004919  0.059951
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0004814   0.0003980    1.21   0.227
## x           0.8882734   0.0374877   23.70 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008918 on 500 degrees of freedom
## Multiple R-squared:  0.5289, Adjusted R-squared:  0.528
## F-statistic: 561.5 on 1 and 500 DF,  p-value: < 2.2e-16

#Extract coefficients from regression
e <- as.numeric(fit$residuals)
c <- fit$coefficients[[1]]
a <- fit$coefficients[[2]]

#Part (2)
test <- adf.test(e)

## Warning in adf.test(e): p-value smaller than printed p-value

#Print adf test results
test

##
## Augmented Dickey-Fuller Test
##
## data:  e
## Dickey-Fuller = -8.5793, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary

#Result shows that residuals are stationary

#Part (3)
z <- y - (a*x) + c
delta <- 2 * sd(z)

#Declare time and order type vectors
t = vector(mode="integer",length = 0)
port.order <- vector(mode="character",length = 0)

```

```

for (i in 1:length(x)) {
#round() is used on the values in the conditions to obtain good trade signals

#Condition for Short portfolio
if(round(y[i]-(a*x[i]),digits = 2)==round(c+delta,digits = 2)) {
  if(length(t)==0) {
    t=rbind(t,i)
    port.order <- rbind(port.order,"SHORT")

  }
  #Additional condition to ensure alternating buy/sell strategy
  if((length(t)>0)&(port.order[length(port.order)]=="LONG")) {
    t=rbind(t,i)
    port.order <- rbind(port.order,"SHORT")
  }
}

#Condition for Long Portfolio
if((round(y[i]-(a*x[i]),digits = 2)==round(c-delta,digits = 2))) {
  if(length(t)==0) {
    t=rbind(t,i)
    port.order <- rbind(port.order,"LONG")
  }
  #Additional condition to ensure alternating buy/sell strategy
  if((length(t)>0)&(port.order[length(port.order)]=="SHORT")) {
    t=rbind(t,i)
    port.order <- rbind(port.order,"LONG")
  }
}
}

#Compile data frame with trade data
trade.info <- data.frame(t,port.order)
colnames(trade.info) <- c("Time","Portfolio Order Type")

#Create vector to number rows
vec1 <- seq(1,length(trade.info[,1]),1)
rownames(trade.info) <- c(vec1)

#Print trades
trade.info

##      Time Portfolio Order Type
## 1      6              SHORT
## 2      9              LONG
## 3     47              SHORT
## 4     48              LONG

```

##	5	62	SHORT
##	6	77	LONG
##	7	167	SHORT
##	8	292	LONG
##	9	300	SHORT
##	10	332	LONG
##	11	363	SHORT
##	12	368	LONG
##	13	484	SHORT