novikovPrac5.R

Новиков Денис Владиславович ИНБО-05-22

2024-04-04

# Загрузка данных ---------------------------------------------------------  
  
if ("quantmod" %in% rownames(installed.packages()) == FALSE) {  
 install.packages("quantmod") }  
library(quantmod)

## Загрузка требуемого пакета: xts

## Загрузка требуемого пакета: zoo

##   
## Присоединяю пакет: 'zoo'

## Следующие объекты скрыты от 'package:base':  
##   
## as.Date, as.Date.numeric

## Загрузка требуемого пакета: TTR

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

if ("stringr" %in% rownames(installed.packages()) == FALSE) {  
 install.packages("stringr") }  
library(stringr)  
  
# Мы хотим загрузить акции с данными наименованиями в yahoo  
# "ATVI" не загружался, поэтоу загружаем только "^IXIC"  
downloadable\_stocks <- c("^IXIC")  
  
# Функция получения фреймов с данными  
quantmod::getSymbols(Symbols = downloadable\_stocks,  
 src = "yahoo",  
 from = as.Date.character("1900-01-01"))

## [1] "IXIC"

# Функция get() позволяет получить содержимое объекта по его названию-строке  
# Мы можем и не знать названия акций в скрипте, но всё равно работать с ними  
# при пользовательском вводе названий  
df <- data.frame(get('IXIC'))  
  
# Применяем регулярное выражение для поиска и удаления ненужных символов  
downloadable\_stocks <- stringr::str\_remove(downloadable\_stocks,  
 "[:punct:\\^]")  
# Удалим полученные объекты  
rm(list = downloadable\_stocks)  
  
# Часть 2 -----------------------------------------------------------------  
  
# Функция Альтера-Джонса --------------------------------------------------  
  
# Задание 1 ---------------------------------------------------------------  
  
out\_of\_trend = function(x, dt, method = "Arifm") {  
 if (!is.numeric(x) || !is.numeric(dt)) {  
 stop("Input vectors must be numeric")  
 }  
 if (length(x) < 3) {  
 stop("Input vector length must be at least 3")  
 }  
 if (any(dt >= ceiling(length(x) / 2) - 1)) {  
 stop("dt incorrect")  
 }  
   
 min\_value = min(x) + 1  
 x\_shifted = x + min\_value  
 y = numeric(length(x))  
   
 for (i in 1:length(x)) {  
 if (method == "Arifm") {  
 if (i - dt >= 1 && i + dt <= length(x)) {  
 y[i] = log((x\_shifted[i - dt] + x\_shifted[i + dt]) / (2 \* x\_shifted[i]))  
 }  
 } else if (method == "Geom") {  
 if (i - dt >= 1 && i + dt <= length(x)) {  
 y[i] = log((x\_shifted[i - dt] \* x\_shifted[i + dt]) / (x\_shifted[i]\*\*2))  
 }  
 } else if (method == "Garm") {  
 if (i - dt >= 1 && i + dt <= length(x)) {  
 y[i] = log(2 \* (x\_shifted[i - dt] \* x\_shifted[i + dt]) / (x\_shifted[i] \* (x\_shifted[i - dt] + x\_shifted[i + dt])))  
 }  
 } else {  
 stop("Invalid method specified")  
 }  
 }  
   
 return(y[y != 0])  
}  
  
first\_local\_min = function(x, dx) {  
 for (i in (1 + dx):(length(x) - dx)) {  
 if (x[i - dx] > x[i] & x[i] < x[i + dx]) {  
 return(i)  
 }  
 }  
}  
  
t = seq(0, 10, 0.1)  
x = 2 \* t + 3 + sin(2 \* t)  
  
mean(x)

## [1] 13.03372

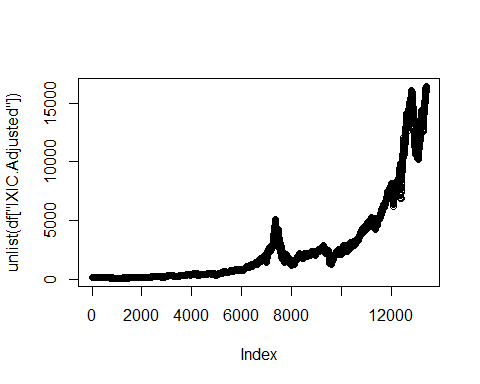
xn = out\_of\_trend(x, 10)  
  
mean(xn)

## [1] 0.005483608

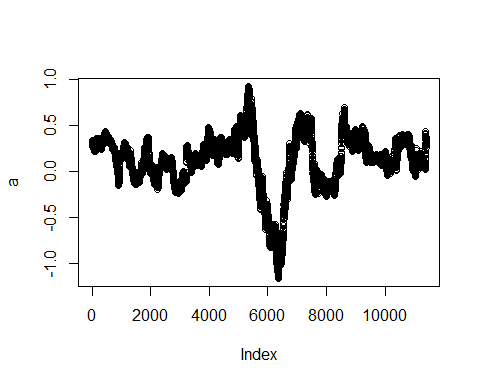
alter\_johns = function(y) {  
   
 a = numeric(length(y))  
   
 for (tau in 1:(length(y) - 1)) {  
   
 val = 0.0  
   
 for (i in 1:(length(y) - tau)) {  
 val = val + abs(y[i + tau] - y[i])  
 }  
   
 val = val \* (1 / (length(y) - tau))  
 a[tau] = val  
 }  
   
 return(a[a != 0])  
}  
  
xn\_a = alter\_johns(xn)  
  
first\_local\_min(xn\_a, 5)

## [1] 29

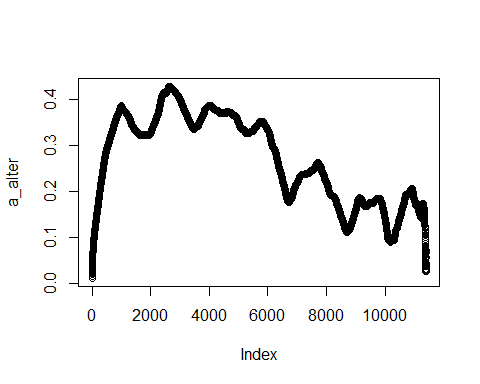
a = out\_of\_trend(unlist(df['IXIC.Adjusted']), 1000)  
  
a\_alter = alter\_johns(a)  
  
plot(unlist(df['IXIC.Adjusted']))



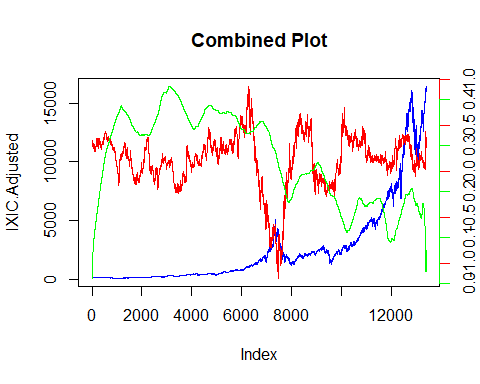
plot(a)



plot(a\_alter)



# Создаем пустой график с нужным нам диапазоном значений по оси Y  
plot(unlist(df['IXIC.Adjusted']), type='n', ylim=c(min(unlist(df['IXIC.Adjusted'])), max(unlist(df['IXIC.Adjusted']), a, a\_alter)), xlab='Index', ylab='IXIC.Adjusted', main='Combined Plot')  
  
# Добавляем первый график  
lines(unlist(df['IXIC.Adjusted']), col='blue')  
  
# Добавляем второй график с отдельной осью Y  
par(new=TRUE)  
plot(1:length(a), a, type='l', col='red', axes=FALSE, xlab='', ylab='')  
axis(side=4, col='red', ylim=c(min(a), max(a)))  
mtext("a", side=4, line=3, col='red')  
  
# Добавляем третий график с отдельной осью Y  
par(new=TRUE)  
plot(1:length(a\_alter), a\_alter, type='l', col='green', axes=FALSE, xlab='', ylab='')  
axis(side=4, col='green', ylim=c(min(a\_alter), max(a\_alter)))  
mtext("a\_alter", side=4, line=2, col='green')



first\_local\_min(a\_alter, 3)

## [1] 1508

# Часть 3 -----------------------------------------------------------------  
  
SIM = function(A, u0, f, n\_iter = 1e5, eps = 1e-2) {  
 if (length(u0) != ncol(A) || length(f) != nrow(A)) {  
 stop("Incorrect input")  
 }  
   
 maximum = max(A)  
 A = A / maximum  
 f = f / maximum  
   
 B = diag(1, ncol(A), ncol(A)) - A  
   
 u = u0  
 iter = 0  
   
 while (iter < n\_iter) {  
 u\_new = B %\*% u + f  
   
 if (max(abs(u\_new - u)) < eps) {  
 print(iter+1)  
 return(u\_new)  
 }  
   
 u = u\_new  
 iter = iter + 1  
 }  
   
 return(u)  
}  
  
A = A = diag(x = c(4, 9), nrow = 2, ncol = 2)  
u0 <- c(0, 0)  
f <- c(4, 2)  
SIM(A, u0, f)

## [1] 8

## [,1]  
## [1,] 0.9909256  
## [2,] 0.2222222