Задание 1. Прогнозирование месячных сезонных рядов по SAR(I)MA

Для выполнения задания были взяты данные с предыдущего задание с именем “GBR\_Dairy.tsv”.

Данные были просмотрены на корректность и разделены на два массива для обучения и тестирования наших данных.

setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

df <- read.delim("GBR\_Dairy.tsv")

View(df)

h <- 24

train\_len <- length(df$dairy) - h

data <- c(df$dairy[1:train\_len])

test <- c(df$dairy[(train\_len + 1): length(df$dairy)])

plot.ts(data, type = "o", pch = 18, col = "#0000ff")

grid()

Graphical user interface

Description automatically generated

Было построено две модели. Одна модель согласно подобранным параметрам, вторая модель построена с помощью auto.arima(). Обе модели были просмотрены на статистики и для каждой из них был построен график автогрегрессии остатков расчётных значений

Первая модель:

model1 <- auto.arima(train\_ts, seasonal = TRUE)

summary(model1)

print(model1)

acf(resid(model1))

Chart, histogram

Description automatically generated

Text

Description automatically generated

Вторая модель:

model2 <- arima(train\_ts, c(2, 1, 1), list(order = c(2, 1, 1)), period = 12)

summary(model2)

print(model2)

acf(resid(model2))

Text

Description automatically generated

Text

Description automatically generated

Chart, histogram

Description automatically generated

Посмотрев на лаги ещё раз можно сделать предположение, что автоарима лучше подобрала параметры, хотя и параметры второй модели подходят. Смущает лишь, что у второй модели на графике лага приближение некоторых значений ближе к границе по сравнению с первой моделью.

Исходя из вышесказанного для дальнейшей работы была взята первая модель и построен прогноз

pr <- predict(model1, n.ahead = h)

yp <- pr$pred

str(yp)

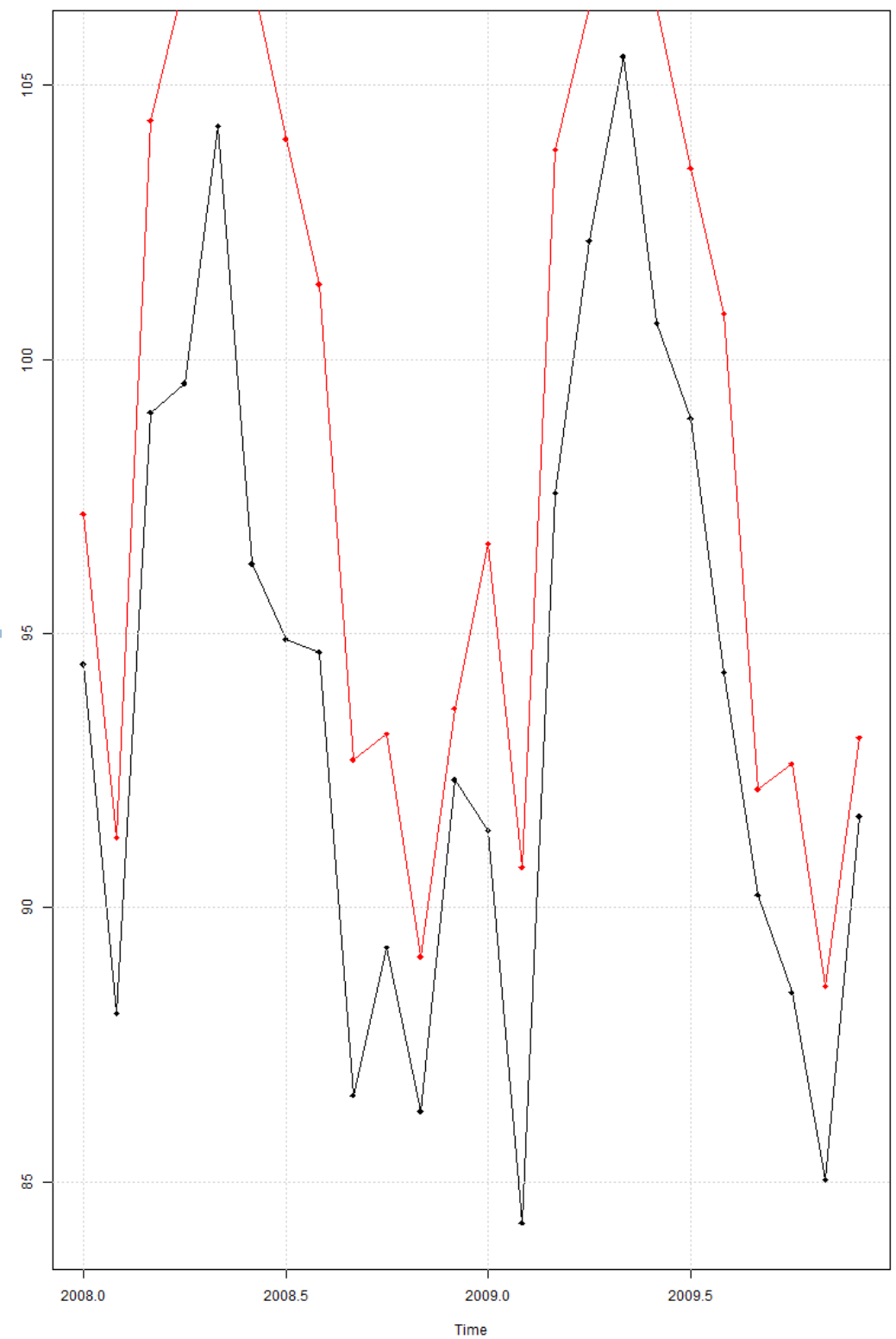
Далее был построен график фактического ряда и точечного прогноза

test\_ts <- ts(data = test, start = c(2008, 1), frequency = 12)

plot(test\_ts, type = "o", pch = 18)

grid()

lines(yp, col = "#ff0000", type = "o", pch = 18)



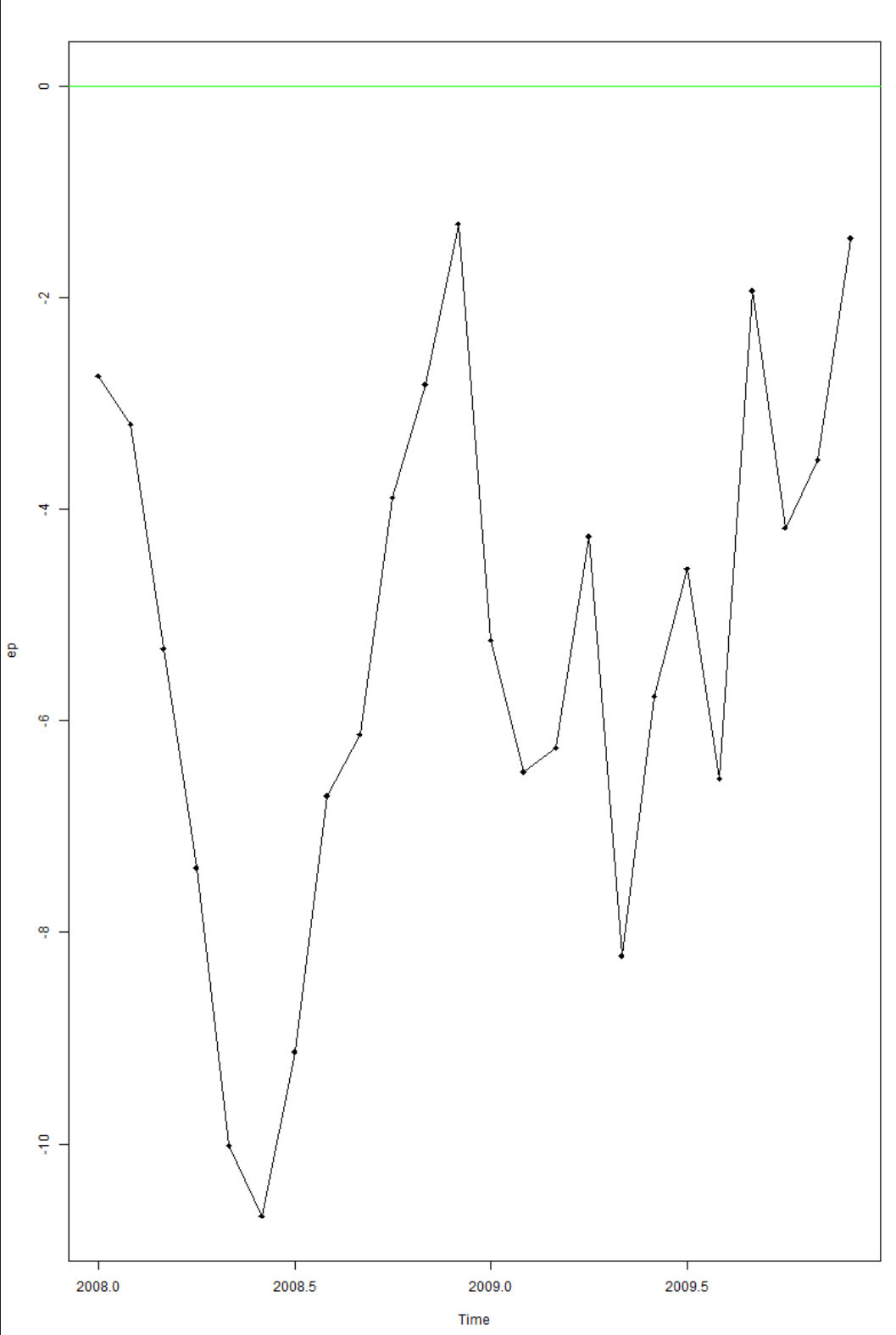
Далее был построен график ошибок прогноза

ep <- test\_ts - yp

plot(ep, type = "o", pch = 18, ylim = range(ep, 0))

abline(h = 0, col = "#00FF00")

где abline – линия нуля(нулевой ошибки)



Теперь к графику фактического ряда и точечного ряда добавляем границы 90%го интервального прогноза.

Построим верхнюю и нижнюю границу интервального прогноза

bottom\_boundary <- pr$pred - pr$se \* qnorm(1 - 0.1)

top\_boundary <- pr$pred + pr$se \* qnorm(1 - 0.1)

Построим необходимый график

full\_ts <- ts(data= df$dairy, start = c(1990, 1), frequency = 12)

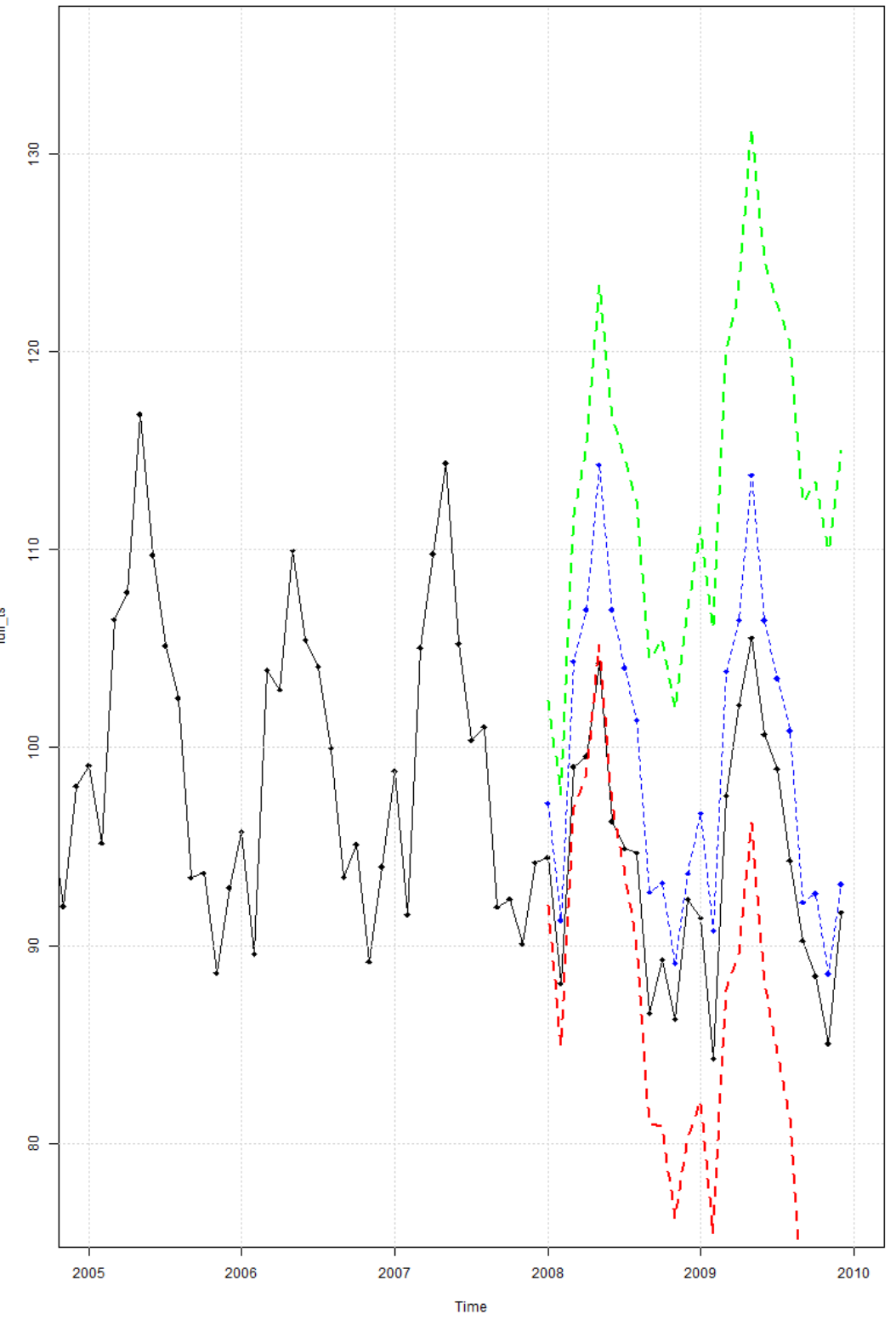
plot(full\_ts, type = "o", pch = 18, xlim = c(2005, 2010))

grid()

lines(yp, col = "blue", type = "o", pch = 18, lty = 2)

lines(bottom\_boundary, col = "#FF0000", lty = 2, lwd = 2)

lines(top\_boundary, col = "#00FF00", lty = 2, lwd = 2)



Задание 2. Волатильность в фондовых индексах

Для выполнения работы был взят фондовый индекс S&P500. Данные были считаны и развёрнуты из-за их некорректной развёртки.

df <- read.delim("US\_sp500.csv", sep = ",")

View(df)

df$Date <- rev(df$Date)

df$Date <- as.Date(df$Date, "%Y-%m-%d")

df$Open <- rev(df$Open)

df$Close <- rev(df$Close)

df$High <- rev(df$High)

df$Low <- rev(df$Low)

df$Volume <- rev(df$Volume)

df[Adj.Close] <- rev(df[Adj.Close])

Рассчитаем логарифмические доходности и построем их график с линией в нуле

df$r <- diff(log(df$Close)) \* 100

plot.ts(df$r)

grid()

abline(h = 0, col = "#FF0000")

Chart, histogram

Description automatically generated

Теперь построим график АКФ для квадратов доходностей

acf(r^2)

Chart, histogram

Description automatically generated

И график модулей доходностей

acf(abs(r))

Chart, histogram

Description automatically generated

Теперь рассчитаем волатильность по методу Riskmetrics при λ = 0.95

data\_len <- length(df$Close)

lambda <- 0.95

s2 <- c(1)

for (t in 2:data\_len){

  s2[t] <- lambda \* s2[t - 1] + (1 - lambda) \* r[t - 1]^2

}

vol.rm <- sqrt(s2)

Построим график значений с линией в нуле

plot.ts(vol.rm, ylim = range(vol.rm, 0))

abline(h = 0, col = "#FF0000", lwd = 2)

Chart

Description automatically generated

И построим график логарифмической доходности с положительными и отрицательными значениями волатильности по методу Riskmetrics

plot.ts(r)

lines(vol.rm, col = "#0000FF", lwd = 3)

lines(-vol.rm, col = "#ff0000", lwd = 3)

abline(h = 0, col = "#00ff00", lwd = 2)

Chart

Description automatically generated

Теперь импортируем библиотеку fGarch и построим модель GARCH(1, 1) для логарифма доходностей

library(fGarch)

ga <- fGarch::garchFit(~ garch(1, 1), r,

                       include.mean = FALSE,

                       cond.dist = "std",

                       trace = FALSE)

print(ga)

vol.ga <- volatility(ga)

Text

Description automatically generated

Построим график волатильности

plot.ts(vol.ga, ylim = range(vol.ga, 0))

abline(h = 0, col = "#FF0000", lwd = 2)

Chart, histogram

Description automatically generated

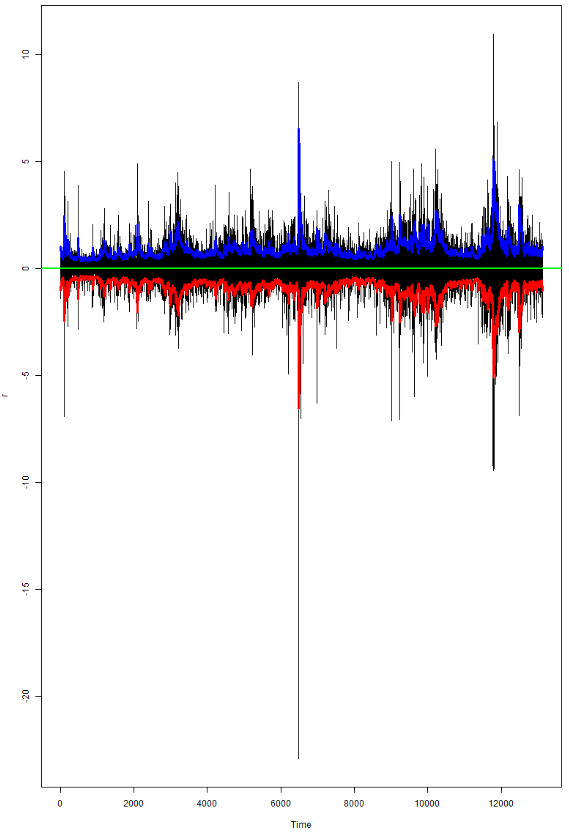
Теперь построим график логарифмических доходностей с волатильностью от гарча

plot.ts(r)

lines(vol.ga, col = "#0000FF", lwd = 3)

lines(-vol.ga, col = "#ff0000", lwd = 3)

abline(h = 0, col = "#00ff00", lwd = 2)



Построим прогноз и построим график стандартного отклонения

pred <- predict(ga, n.ahead = 100)

plot(pred$standardDeviation)

Chart

Description automatically generated

Теперь построим график прогноза волатильности

volp <- c(vol.ga, pred$standardDeviation)

plot.ts(volp, xlim = c(12500, data\_len + 100),

col = "#000000", ylim = c(0, 3))

abline(v = TT, lty = 3)

lines(vol.ga, col = "#ff0000")

abline(h = 0, col = "#00ff00", lwd = 2)

Chart, line chart

Description automatically generated