Задача 1

Задаём начальные условия

sigma\_eps\_sq <- 6400

N <- 100

sd <- sqrt(sigma\_eps\_sq)

#1а

y[1] <- 0

y[2] <- 1

for(t in 3:N){

    eps\_1 <- rnorm(1, 0, sd)

    y[t] <- 1.7 \* y[t-1] - 0.8 \* y[t - 2] + eps\_1

}

#1б

corr\_vector <- c()

    first <- c(1, 0, 2)

    second <- c(1, 0, 2)

for (i in 1:N){

  for (t in 3:N){

    first[t] <- 1.7 \* first[t - 1] - 0.8 \* first[t - 2] + rnorm(1, 0, sd)

    second[t] <- 1.7 \* second[t - 1] - 0.8 \* second[t - 2] + rnorm(1, 0, sd)

  }

  corr\_vector[i] <- cor(first, second)

}

hist(corr\_vector)

summary(corr\_vector)

#1в

intervals <- t.test(corr\_vector)

print(intervals$conf.int[1])

print(intervals$conf.int[2])

Задача 2

Был взят ряд j\_and\_j. Было решено взять логарифм ряда и работать уже с ним.

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

df <- read.delim(file = "J\_and\_J.csv", sep = ",",

                 stringsAsFactors = TRUE, skip = 3) # читаем данные

df$log <- log(df$jj)

plot(df$jj)

plot(df$log)

2а) Были построены автокореляционные функции самого ряда и его первых разностей.

facf <- acf(df$log)

first\_diff <- diff(df$log)

fdiff\_acf <- acf(first\_diff)

plot(facf)

plot(fdiff\_acf)

В результате у нас уменьшилось количество вылетающих лагов и лаги стали чередоваться по знаку

2б) Построили неообходимые графики ошибки и фактических данных

data\_len <- length(df$jj)

train\_data <- df$log[1:(data\_len - 13)]

train\_time <- df$Date[1:(data\_len - 13)]

test\_data <- df$log[(data\_len - 12):(data\_len)]

test\_time <- df$Date[(data\_len - 12):(data\_len)]

model <- lm(train\_data ~ train\_time)

library(forecast)

predict <- forecast(model, newdata = test\_time, h = 12)

# Смотрим на предсказанные данные и наши данные

plot(predict, col = "#00FF00")

points(df$log, col = "#FF0000")

# Смотрим на ошибку

plot(predict$pred - test)

abline(h = 0, col = "#0000FF")

2в) Построили сезонную ариму по тем же данным и визуализировали

model <- arima(train\_data, order = c(1, 1, 1),

seasonal = c(1, 1, 1), period = 4)

pred\_2 <- forecast(model, h = 12)

plot(pred\_2)

points(df$log, col = "#FF0000")

2г) Для сравнения двух рядов берём предыдущие полученные результаты и сравниваем по показателю средней обратной ошибки

Задача 3

Считали данные VEM

3а) Рассчитали волатильность и методом анализа графика нашли, что в январе была самая большая волатильность

data\_len <- length(my\_data$VEN)

r <- diff(log(my\_data$VEN)) \* 100

lambda <- 0.95

s2 <- c(1)

s2\_max <- -10000000

index = 2

for (t in 2:data\_len){

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

  if (s2[t] > s2\_max) {

    s2\_max = s2[t]

    index = t

  }

}

print(my\_data$Date[index]) # самая большая волатильность дневная

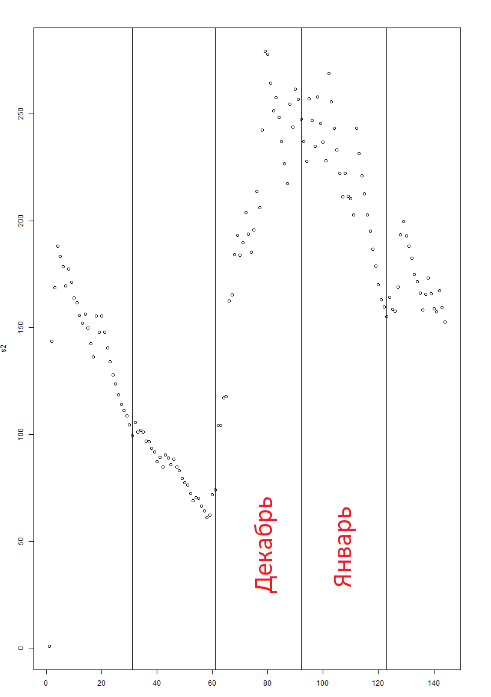
plot(s2)

abline(v = 31)

abline(v = 61)

abline(v = 92)

abline(v = 123)



3б) Теперь создадим фиктивную переменную положительной доходности, далее построим по 60 наблюдениям модель логит регрессии знака от волатильности и Оценим зависимость статистической значимости

sign <- c(1)

plus <- as.numeric(diff(my\_data$VEN))

s2\_len <- length(s2)

vol.rm <- sqrt(s2)

sign <- ifelse(plus > 0, 1, 0)

reg <- glm(sign[1:60] ~ vol.rm[1:60], family = "binomial")

summary(reg)

#согласно значению p-value и самой статистики можно сказать, что не является статистически значимой

Deviance Residuals:

Min 1Q Median 3Q Max

-1.3026 -1.1412 -0.9869 1.1696 1.4057

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.3868 1.3984 0.992 0.321

vol.rm[1:60] -0.1392 0.1313 -1.060 0.289

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 83.111 on 59 degrees of freedom

Residual deviance: 81.913 on 58 degrees of freedom

AIC: 85.913

3в) По оставшимся наблюдениям построим прогноз вероятности и нарисуем график ROC и найдём показатель AUC

AUC = 0.5193

Пришлось создать датафрейм потому что иначе предикт не хотел данные принимать и вылезали ошибки

#3б

sign <- c(1)

plus <- as.numeric(diff(my\_data$VEN))

s2\_len <- length(s2)

vol.rm <- sqrt(s2)

sign <- ifelse(plus > 0, 1, 0)

neededFrame <- data.frame( vol.rm = vol.rm, sign = c(0, sign))

train\_subset <- 1:60

reg <- glm(sign ~ vol.rm, data = neededFrame,

subset = train\_subset, family = "binomial")

summary(reg)

#согласно значению p-value и самой статистики можно сказать что не является статистически значимой

#3в

predict\_data <- predict(reg,

newdata = neededFrame[c(61:144), ], type = 'response')

print(predict\_data)

install.packages("pROC")

print(length(fitted(lgt)))

test\_subset <- c(61:length(neededFrame$sign))

pROC::plot.roc(neededFrame$sign[test\_subset], predict\_data)

pROC::roc(sign[test\_subset], predict\_data)

3г) Получили значение 0.2494047

spec\_diff <- sign[test\_subset] - predict\_data[test\_subset]

brier\_test <- mean((spec\_diff^2))

print(brier\_test)

3д) получили что не значимый

support\_vector <- rep((1.0 / 2.0), length(predicted\_data))

binded <- cbind(predict\_data, support\_vector)

epep <- sign[test\_subset] - binded

epep\_1 <- epep[,1]^2

epep\_2 <- epep[,2]^2

reg.DM <- lm(epep\_2- epep\_1 ~ 1)

summary(reg.DM)

library(lmtest)

library(sandwich)

coeftest(reg.DM)

coeftest(reg.DM, vcov=vcovHAC)

coeftest(lm(epep\_2 - epep\_1 ~ 1), vcov=vcovHAC)