Задание 1. Одношаговое прогнозирование валютного курса с помощью авторегрессии

Для выполнения задания были скачаны дневные данные по курсу одной из валют к доллару США с 1994. Была выбрана валюта мексиканское песо. Данные были обработаны и сохранены в файл “datatask11.csv”. Файл будет прилагаться к работе.

Загрузим данные в программу, добавим логарифм данных, обработаем данные и поменяем имена столбцов

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

df <- read.delim(file = "datatask11.csv", sep = ",",

                 stringsAsFactors = TRUE)

names(df) <- c("Date", "MXN")

df$MXN <- as.numeric(df$MXN)

df$Date <- as.Date(df$Date)

df$log <- log(df$MXN)

head(df)

Сделаем разбивку данных построим первый лаг и отобразим график лага к логарифму данных

data\_len <- length(df$log)

tr <- 1:data\_length

train\_data\_len <- 365 \* 2

train\_data <- tr <= data\_len - train\_data\_len

test\_data <- !train\_data

x <- df$log

x1 <- c(x[1], x[-length(x)])

plot(x ~ x1)

Chart, scatter chart

Description automatically generated

Построим второй лаг и также отобразим график

x2 <- c(x1[1], x1[-length(x1)])

plot(x ~ x2)

Chart, scatter chart

Description automatically generated

Далее маленько вернёмся по процессу и начнём реализовывать своё решение, поскольку первоначальное решение было продемонстрировано на доске.

Создадим 2 вспомогательных переменных длины данных и добавим лаги в наш датафрейм

data\_length <- length(df$MXN)

train\_length <- 365 \* 2

x1 <- df$MXN[c(2:((data\_length) - 1))]

x2 <- df$MXN[c(1:((data\_length) - 2))]

df <- df[c(3:data\_length), ]

df$x1 <- x1

df$x2 <- x2

Теперь создадим объекты для тестовых, тренировочных и прогнозируемых данных и сразу добавим в прогнозируемые данные RW(случайное блуждение)

train\_data <- df[c(1:(data\_length - train\_length)), ]

test\_data <- df[c(data\_length - train\_length:data\_length), ]

predicted\_data <- data.frame(RW = test\_data$x1)

Построим одношаговые прогнозы AR1, AR2, AR02 и M2

ar\_1 <- lm(data <- train\_data, MXN ~ x1)

summary(ar\_1)

predicted\_data$ar\_1 <-  predict(ar\_1, newdata = test\_data$x1)

ar\_2 <- lm(data = train\_data, MXN ~ x1 + x2)

summary(ar\_2)

predicted\_data$ar\_2 <-  predict(ar\_2,

newdata = data.frame(x1 = test\_data$x1, x2 = test\_data$x2))

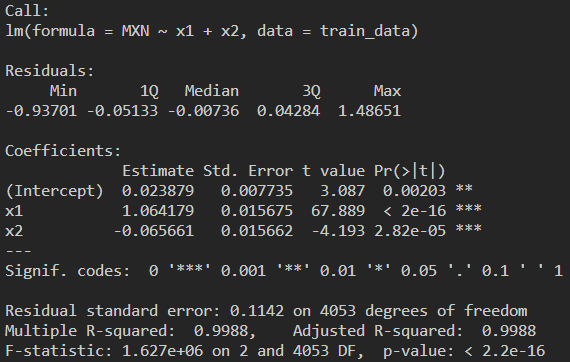
ar\_02 <- lm(data = train\_data, MXN ~ x2)

summary(ar\_02)

predicted\_data$ar\_02 <-  predict(ar\_02,

newdata = data.frame(x1 = test\_data$x1, x2 = test\_data$x2))

predicted\_data$m2 <- (test\_data$x1 + test\_data$x2) / 2

Text

Description automatically generatedText

Description automatically generated

Для визуализации прогнозов была создана специальная функция

draw\_plot <- function(test\_data, predicted\_data,

data\_name, data\_begin, data\_end) {

    plot(data = test\_data[c(data\_begin:data\_end), ],

MXN ~ Date, type = "l", xlab = data\_name)

lines(predicted\_data[c(data\_begin:data\_end)] ~

test\_data$Date[c(data\_begin:data\_end)], col = "#0000ff")

}

С помощью данной функции построим графики прогноза для:

1. RW  
   Chart, line chart

   Description automatically generated
2. AR1  
   Chart, line chart

   Description automatically generated
3. AR2  
   Chart, line chart

   Description automatically generated
4. AR02  
   Chart, line chart

   Description automatically generated
5. M2  
   Chart, line chart

   Description automatically generated

Вычислим RMSE

rmse <- {}

rmse$rw <- sqrt(mean((test\_data$MXN - predicted\_data$RW)^2)) \* 100

rmse$ar\_1 <- sqrt(mean((test\_data$MXN - predicted\_data$ar\_1)^2)) \* 100

rmse$ar\_2 <- sqrt(mean((test\_data$MXN - predicted\_data$ar\_2)^2)) \* 100

rmse$ar\_02 <- sqrt(mean((test\_data$MXN - predicted\_data$ar\_02)^2)) \* 100

rmse$m2 <- sqrt(mean((test\_data$MXN - predicted\_data$m2)^2)) \* 100

print(rmse)

$rw

[1] 11.46213

$ar\_1

[1] 11.44483

$ar\_2

[1] 11.4201

$ar\_02

[1] 16.69509

$m2

[1] 13.15429

Теперь сравним пару прогнозов RW попарно с другими прогнозами

rw\_compare <- function(rw\_data, compare\_data, test\_data) {

    compare <- lm((test\_data - rw\_data)^2 -

(test\_data - compare\_data)^2 ~ 1)

    summary(compare)

}

rw\_compare(predicted\_data$RW, predicted\_data$ar\_1, test\_data$MXN)

rw\_compare(predicted\_data$RW, predicted\_data$ar\_2, test\_data$MXN)

rw\_compare(predicted\_data$RW, predicted\_data$ar\_02, test\_data$MXN)

rw\_compare(predicted\_data$RW, predicted\_data$m2, test\_data$MXN)

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated

Задание 2. Ложная корреляция при сезонности

Взяли два совершенно разных помесячных временных ряда GBR\_Dairy и GRC\_Confect, считали их, приравняли к одному размеру и добавили к ним логарифм данных

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

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

df2 <- read.delim("GRC\_Confect.tsv")

head(df)

df <-data.frame(dairy = df$dairy[1:length(df2$confect)],

month = df$month[1:length(df2$confect)])

df$log <- log(df$dairy)

df2$log <- log(df2$confect)

Построили график данных для первого датасета и посчитали корреляцию

plot.ts(df$dairy)

grid()

cor(df$dairy, df2$confect)

A picture containing bar chart

Description automatically generated

> cor(df$dairy, df2$confect)

[1] -0.399636

Также можно посмотреть

cor.test\_data(df$dairy, df2$confect)

Задание 3. Прогнозирование с помощью тренда и сезонных фиктивных переменных

Берём один из рядов из предыдущего задания

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

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

predict\_length <- 240 - 24

dairy <- df$dairy

month <- rep(1:12, length(dairy) / 12)

times <- c(1:length(dairy))

lms <- lm(dairy ~ times + factor(month), subset = c(1:predict\_length))

Строим прогноз на последние 24 месяца по модели с линейным трендом и месячными фиктивными переменными

predicted\_data <- predict(lms,

newdata = data.frame("times" = c(1:(predict\_length + 24)), "month" = month))

Рисуем график прогноза и ошибок

plot(dairy, type = "l", lwd = 2, col = "#2b00ff")

lines(predicted\_data, col = "#04ff00")

abline(v = predict\_length, col = "#ff0000")

Chart

Description automatically generated