matrix-computations

Ata COLAK

2023-07-28

Computations on Matrices

Introduction

This task shows a few examples of matrix computations using R. The task shows examples of graph plots, selecting custom rows and columns in a matrix, creating custom functions, exporting certain dates of the year to analyse, and creating custom tables.

The dataset used is EUR currency rates until 2023-02-02, and can be accessed via the link below:

https://raw.githubusercontent.com/gagolews/teaching-data/“,”master/marek/eurxxx-20230202-numeric.csv

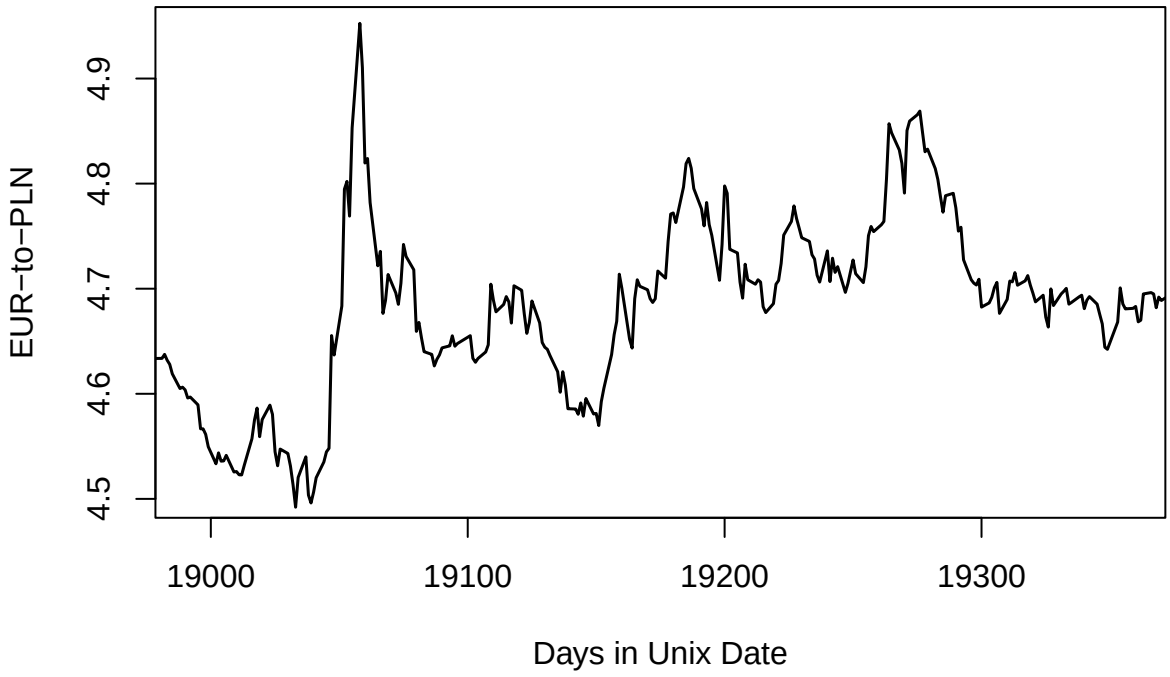
First Part - EUR-PLN Plot

Code is explained via the comments.

```
X <- as.matrix(read.csv(
paste0("https://raw.githubusercontent.com/gagolews/teaching-data/",
"master/marek/eurxxx-20230202-numeric.csv"), comment="#"))
head(X, 3)
```

```
##       Date    USD    JPY    BGN CYP    CZK    DKK EEK     GBP    HUF LTL LVL
## [1,] 19390 1.0988 141.12 1.9558  NA 23.809 7.4393  NA 0.89289 387.20  NA  NA
## [2,] 19389 1.0894 141.37 1.9558  NA 23.775 7.4396  NA 0.88413 390.20  NA  NA
## [3,] 19388 1.0833 141.27 1.9558  NA 23.792 7.4388  NA 0.88073 390.91  NA  NA
##      MTL    PLN ROL    RON     SEK SIT SKK    CHF   ISK     NOK HRK RUB TRL
## [1,]  NA 4.7015  NA 4.9025 11.3587  NA  NA 0.9992 153.5 10.9535  NA  NA  NA
## [2,]  NA 4.7075  NA 4.9117 11.3455  NA  NA 0.9980 153.5 10.8569  NA  NA  NA
## [3,]  NA 4.7090  NA 4.9210 11.3480  NA  NA 1.0032 153.1 10.9083  NA  NA  NA
##          TRY    AUD    BRL    CAD    CNY    HKD      IDR    ILS     INR     KRW
## [1,] 20.6766 1.5407 5.4859 1.4602 7.3878 8.6183 16360.69 3.7577 90.3015 1345.90
## [2,] 20.4978 1.5392 5.5174 1.4506 7.3452 8.5444 16299.47 3.7690 89.1060 1339.22
## [3,] 20.3787 1.5476 5.5373 1.4570 7.3198 8.4898 16282.57 3.7675 88.6360 1338.90
##          MXN    MYR    NZD    PHP    SGD    THB     ZAR
## [1,] 20.4050 4.6655 1.6855 59.170 1.4352 36.030 18.7046
## [2,] 20.4919 4.6471 1.6903 59.318 1.4303 35.841 18.8328
## [3,] 20.3961 4.6211 1.6858 59.192 1.4268 35.787 18.9223
```

```r
# Plot of EUR-PLN for the year 2022,
# Values of "xlim" are columns where "Date" corresponds to year 2022.
# "ylim" has been increased to show the plot close-up.
plot(X[,1],X[,14] , xlim = c(18993, 19357), ylim = c(4.5, 4.95), type = "l", lwd = 1.5, ylab = "EUR-to-
```



Second Part - Min, Median, Mean, Max Table

Code is explained via the comments.

```r
# Custom function
min_median_mean_max <- function(x)
{
    c(min=min(x), median=median(x), mean=mean(x), max=max(x))
}


# Transpose to show the matrix the way it is in the task sheet,
# Only includes rows with "Date" less than 19357 and more than or equal to 18993,
# Using custom function min_median_mean_max,
# Columns 2 to 42 to include every currency, but not the date column.

t(apply(t(X[X[,"Date"]<19357 & X[,"Date"]>=18993,2:42]), 1, min_median_mean_max))
```

```
##             min      median         mean         max
```

```
## USD     0.95650     1.05240 1.053049e+00     1.14640
## JPY   125.55000   138.26000 1.380274e+02   147.59000
## BGN     1.95580     1.95580 1.955800e+00     1.95580
## CYP          NA          NA           NA          NA
## CZK    24.11600    24.53500 2.456593e+01    25.86600
## DKK     7.43640     7.43910 7.439564e+00     7.44570
## EEK          NA          NA           NA          NA
## GBP     0.82388     0.85115 8.527609e-01     0.90268
## HUF   352.92000   397.35000 3.912865e+02   430.65000
## LTL          NA          NA           NA          NA
## LVL          NA          NA           NA          NA
## MTL          NA          NA           NA          NA
## PLN     4.49210     4.69250 4.686107e+00     4.95250
## ROL          NA          NA           NA          NA
## RON     4.81980     4.94270 4.931313e+00     4.95010
## SEK    10.23000    10.60200 1.062958e+01    11.15800
## SIT          NA          NA           NA          NA
## SKK          NA          NA           NA          NA
## CHF     0.94370     0.99600 1.004709e+00     1.05710
## ISK   136.30000   141.10000 1.422432e+02   153.30000
## NOK     9.49230    10.12250 1.010261e+01    10.58380
## HRK     7.50280     7.53010 7.534873e+00     7.58150
## RUB          NA          NA           NA          NA
## TRL          NA          NA           NA          NA
## TRY    14.93100    17.87020 1.740879e+01    19.96490
## AUD     1.43060     1.50990 1.516692e+00     1.61650
## BRL     4.96820     5.35470 5.439904e+00     6.44200
## CAD     1.28810     1.36130 1.369491e+00     1.45830
## CNY     6.75180     7.04670 7.078800e+00     7.53260
## HKD     7.50840     8.23600 8.245100e+00     8.92890
## IDR 14604.52000 15594.75000 1.562525e+04 16765.93000
## ILS     3.25370     3.54790 3.534544e+00     3.75750
## INR    78.26550    82.60380 8.268639e+01    88.22950
## KRW  1311.40000  1352.69000 1.358073e+03  1428.57000
## MXN    19.23630    20.96750 2.118686e+01    23.58560
## MYR     4.42810     4.62850 4.627867e+00     4.79140
## NZD     1.56570     1.66010 1.658247e+00     1.74190
## PHP    54.44900    57.30400 5.731381e+01    59.67800
## SGD     1.38380     1.44550 1.451165e+00     1.54270
## THB    35.73200    36.81600 3.685616e+01    38.08000
## ZAR    15.68200    17.15240 1.720863e+01    18.68550
```

Third Task - AUD-PLN Plot

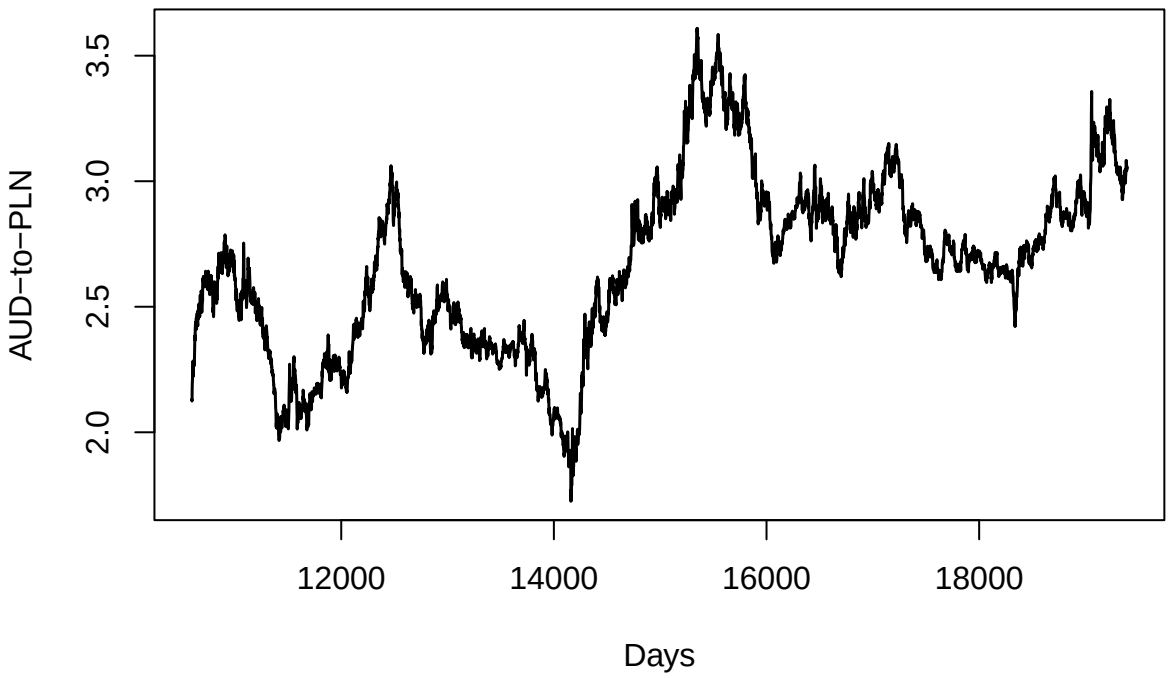
Code is explained via the comments.

```r
# EUR-PLN divided by EUR-AUD gives AUD-PLN rates.

aud2pln <-  X[,14] / X[,27]

# Plot of equation:

plot(X[,1], aud2pln, xlab = "Days", ylab = "AUD-to-PLN", type = "l", lwd = 1.5)
```



Fourth Task - Average AUD-PLN Rates for 2020, 2021, 2022

Code is explained via the comments.

```
# Matrix seperated in dates 2020, 2021 and 2022.

date_2020 <- X[X[,"Date"]<=18627 & X[,"Date"]>=18262,]
date_2021 <- X[X[,"Date"]<= 18992 & X[,"Date"] >=18628,]
date_2022 <- X[X[,"Date"]<=19357 & X[,"Date"]>=18993,]

# AUD-PLN rates in each year.

aud2pln2020 <- date_2020[,14] / date_2020[,27]
aud2pln2021 <- date_2021[,14] / date_2021[,27]
aud2pln2022 <- date_2022[,14] / date_2022[,27]


# Custom table with values above.
exchange_table <- matrix(c(2020, mean(aud2pln2020), 2021, mean(aud2pln2021), 2022, mean(aud2pln2022)), r
colnames(exchange_table) <- c("Year", "AverageRate")

exchange_table
```

```
##      Year AverageRate
```

```
## [1,] 2020    2.686661
## [2,] 2021    2.899268
## [3,] 2022    3.093018
```

Conclusion

In conclusion, this task demonstrated the use of matrices in R for analysing financial data, and various methods to access certain information in matrices.

While doing this task, I have learned how to select certain rows and columns from matrices and analyse the data I have received by putting it through certain conditions.

I have observed that during 2020-2021-2022, the AUD-PLN rates have increased in value every year.

What was interesting to me was the peak in EUR-PLN values around Unix date 19050. The value has increased over 5 and immediately went down again.