Homework Assignment #1

Hornella Fokem Fosso & Jan Lennarz

1/23/2021

Exercise 1

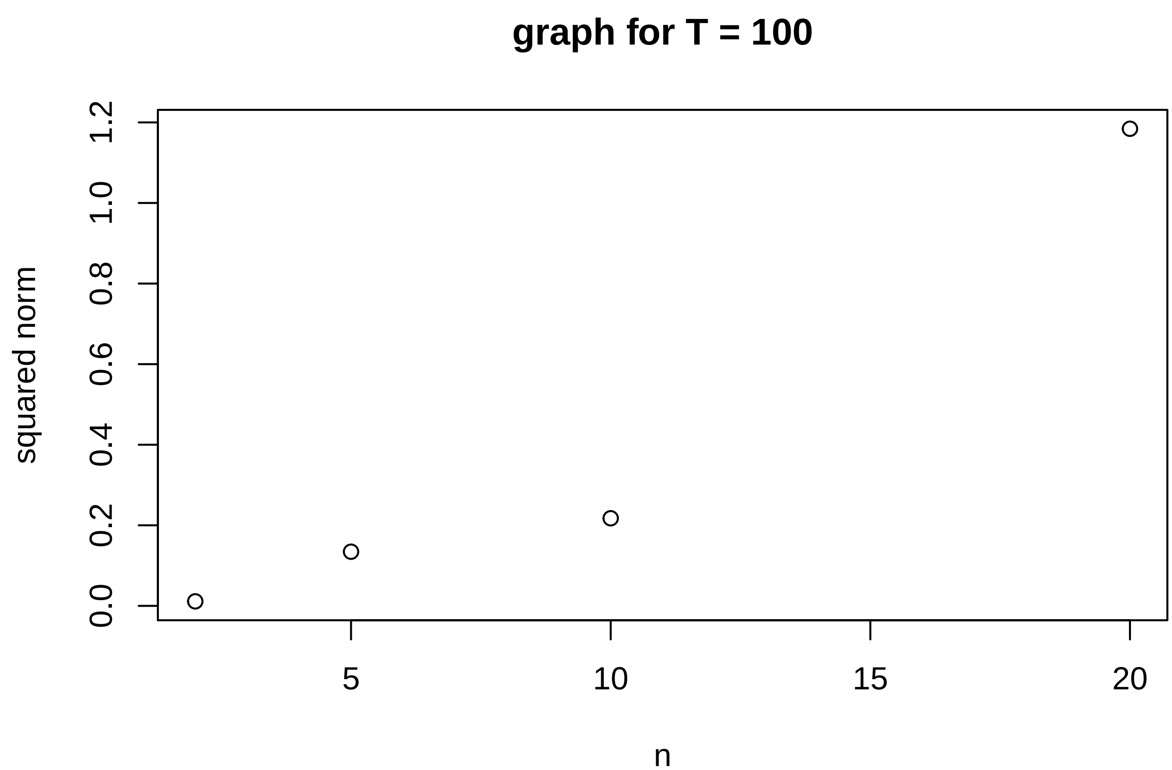
In this exercise, it will be simulated for different values of n, a path of length T from the VAR(1) model

$$\mathbf{X}_t = A_n \mathbf{X}_{t-1} + \epsilon_t, \quad t \in \mathbb{Z},$$

where $(\epsilon_t) \subset \mathbb{R}$ is s a multivariate standard Gaussian white noise.

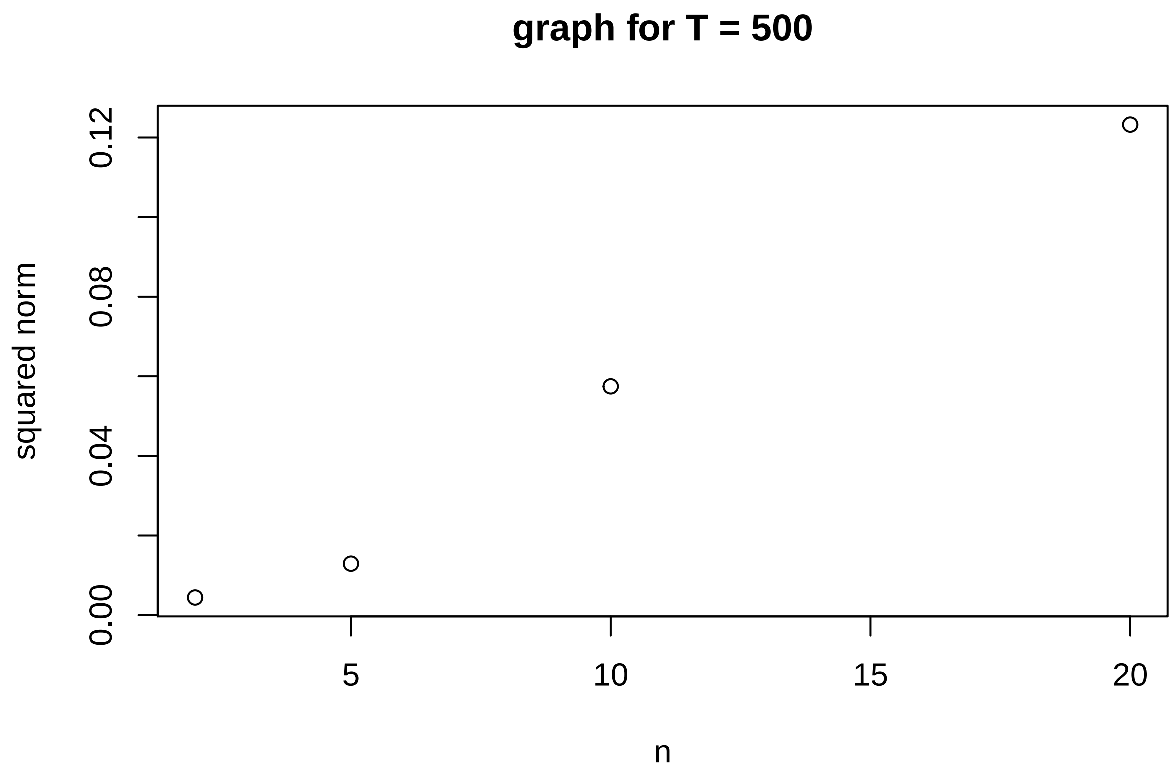
we perform the simulation for $T = 100$ and plot the values of the squared norm corresponding to each n.

```r
ns <- c(2,5,10,20) # different values of n
lT <- 100          # length T
sNorm <- numeric(length(ns))  # for storing the squared norms
for (s in 1:length(ns)) {
  n <- ns[s]
  X=matrix(0,nrow=n ,ncol=lT)
  An <- 0.5*diag(n)
  for (i in 1:(n-1)) {
    An[i,i+1] <- 1/5
  }
  for (j in 2:lT) {
    X[,j] <- An %*% X[,(j-1)] + t(t(rnorm(n)))
  }
  Estim <- VAR(t(X),1, output = F)
  A_hat <- Estim$Phi
  B <- A_hat - An
  B <- crossprod(B)
  sNorm[s] <- max(abs(eigen(B)$values))
}

plot(ns, sNorm, xlab = "n", ylab = "squared norm", main = "graph for T = 100")
```

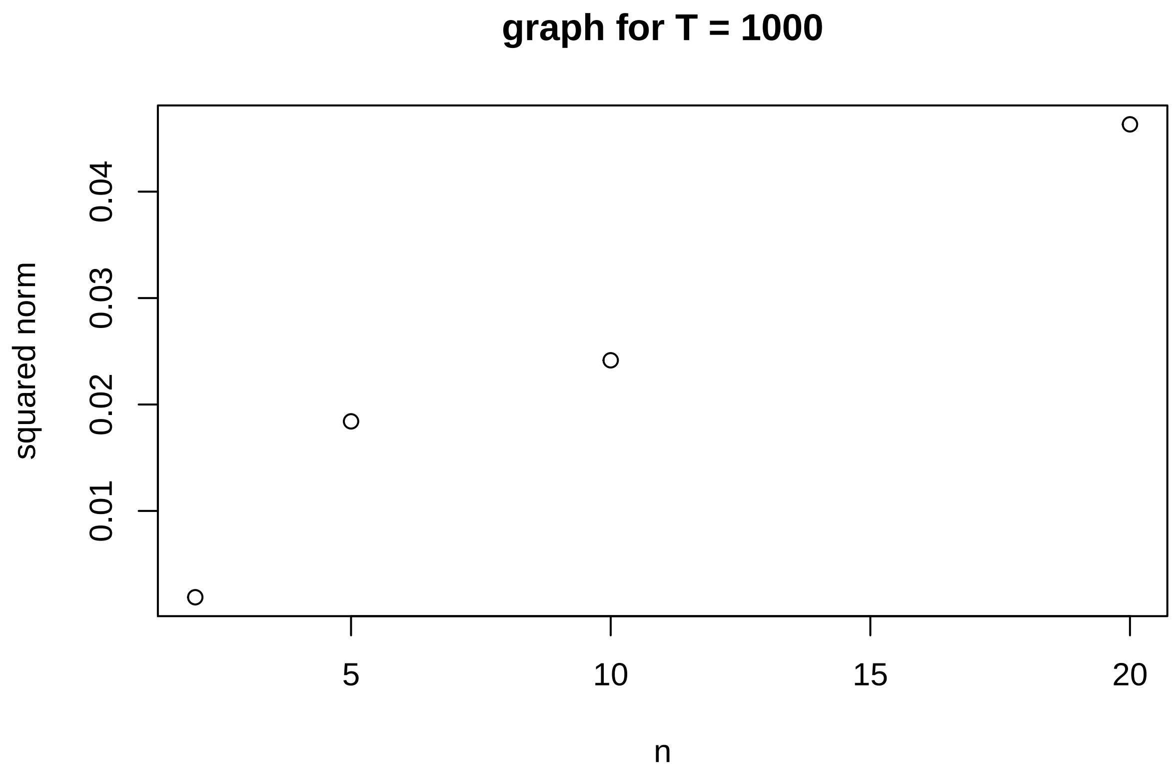


The plot shows that the squared norm increases when n becomes larger. In other words, the estimation error of the seems to be larger when the dimension of the vector is increased. Let's repeat the experiment for $T = 500$ and for $T = 1000$

```
ns <- c(2,5,10,20)
lT <- 500
sNorm <- numeric(length(ns))
for (s in 1:length(ns)) {
  n <- ns[s]
  X=matrix(0,nrow=n ,ncol=lT)
  An <- 0.5*diag(n)
  for (i in 1:(n-1)) {
    An[i,i+1] <- 1/5
  }
  for (j in 2:lT) {
    X[,j] <- An %*% X[,(j-1)] + t(t(rnorm(n)))
  }
  Estim <- VAR(t(X),1, output = F)
  A_hat <- Estim$Phi
  B <- A_hat - An
  B <- crossprod(B)
  sNorm[s] <- max(abs(eigen(B)$values))
}

plot(ns, sNorm, xlab = "n", ylab = "squared norm", main = "graph for T = 500")
```




```
ns <- c(2,5,10,20)
lT <- 1000
sNorm <- numeric(length(ns))
for (s in 1:length(ns)) {
  n <- ns[s]
  X=matrix(0,nrow=n ,ncol=lT)
  An <- 0.5*diag(n)
  for (i in 1:(n-1)) {
    An[i,i+1] <- 1/5
  }
  for (j in 2:lT) {
    X[,j] <- An %*% X[,(j-1)] + t(t(rnorm(n)))
  }
  Estim <- VAR(t(X),1, output = F)
  A_hat <- Estim$Phi
  B <- A_hat - An
  B <- crossprod(B)
  sNorm[s] <- max(abs(eigen(B)$values))
}

plot(ns, sNorm, xlab = "n", ylab = "squared norm", main = "graph for T = 1000")
```



Exercise 2

Data: The series are of various lengths but all end in 1988. The data set contains the following series: consumer price index, industrial production, nominal GNP, velocity, employment, interest rate, nominal wages, GNP deflator, money stock, real GNP, stock prices (S&P500), GNP per capita, realwages, unemployment.

We look only at the GNP per capita, nominal GNP and the real GNP.

Source: C. R. Nelson and C. I. Plosser (1982), Trends and Random Walks in Macroeconomic Time Series.Journal of Monetary Economics,10, 139–162. doi: 10.1016/03043932(82)900125.Formerly in the Journal of Business and Economic Statistics data archive, currently athttp://korora.econ.yale.edu/phillips/data/np&enp.dat.

1.

Stationarity

First we read the data and do some preprocessing.

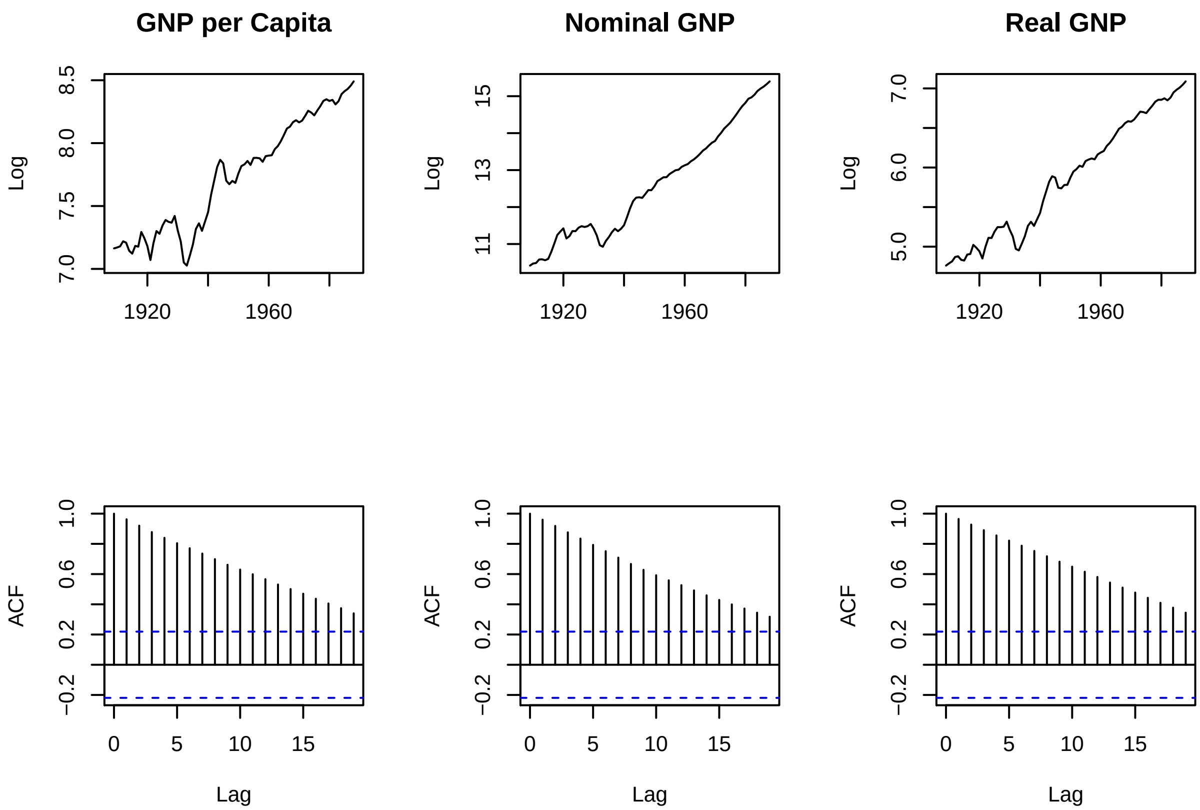
```
data(NelPlo)
gnp <- cbind(1,2,gnp.capita, gnp.nom, gnp.real)
n <- dim(gnp)[1]
```

We will look at 3 different versions of the data: Original, log-transformation, series of differences of the log-transformation.

```
Y_orig <- gnp[,3:5]
Y_log=log(gnp[,3:5])
Y_rate <- Y_log[2:n,] - Y_log[1:(n-1),]
Y_rate <- 100*Y_rate
```

Original:

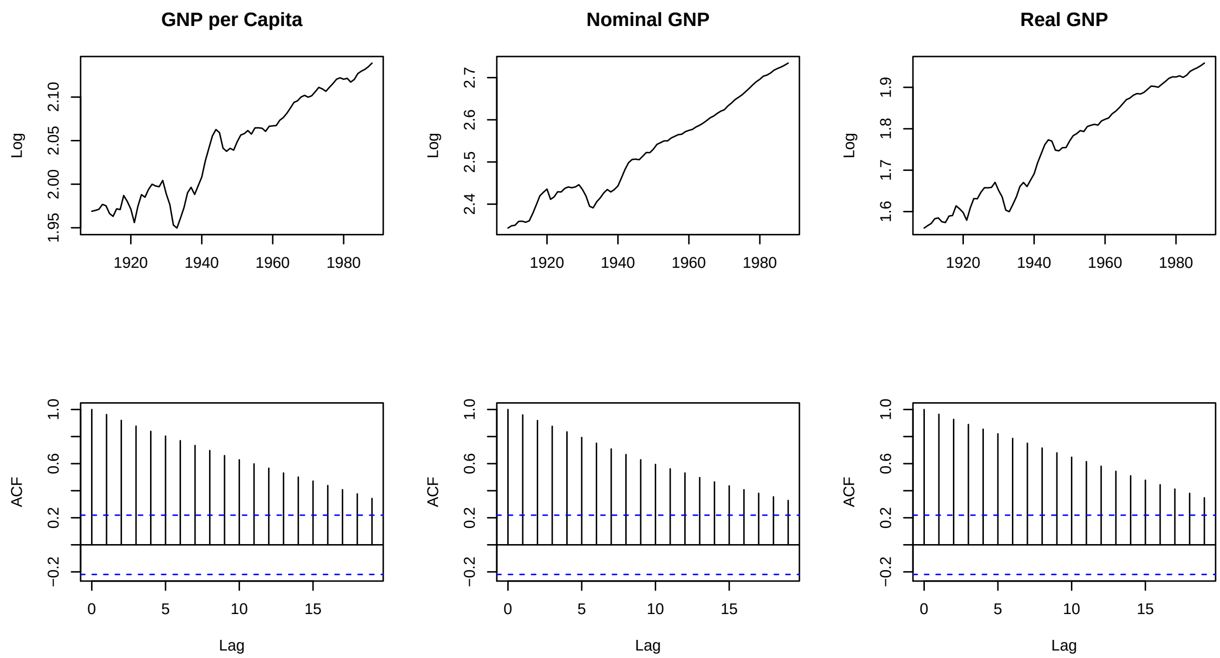
```
par(mfrow=c(2,3))
plot(Y_orig[,1],type="l",xlab="",ylab="Log",main="GNP per Capita")
plot(Y_orig[,2],type="l",xlab="",ylab="Log",main="Nominal GNP")
plot(Y_orig[,3],type="l",xlab="",ylab="Log",main="Real GNP")
acf(Y_orig[,1],main="")
acf(Y_orig[,2],main="")
acf(Y_orig[,3],main="")
```



We see that the original data is not stationary.

Log-Transformation:

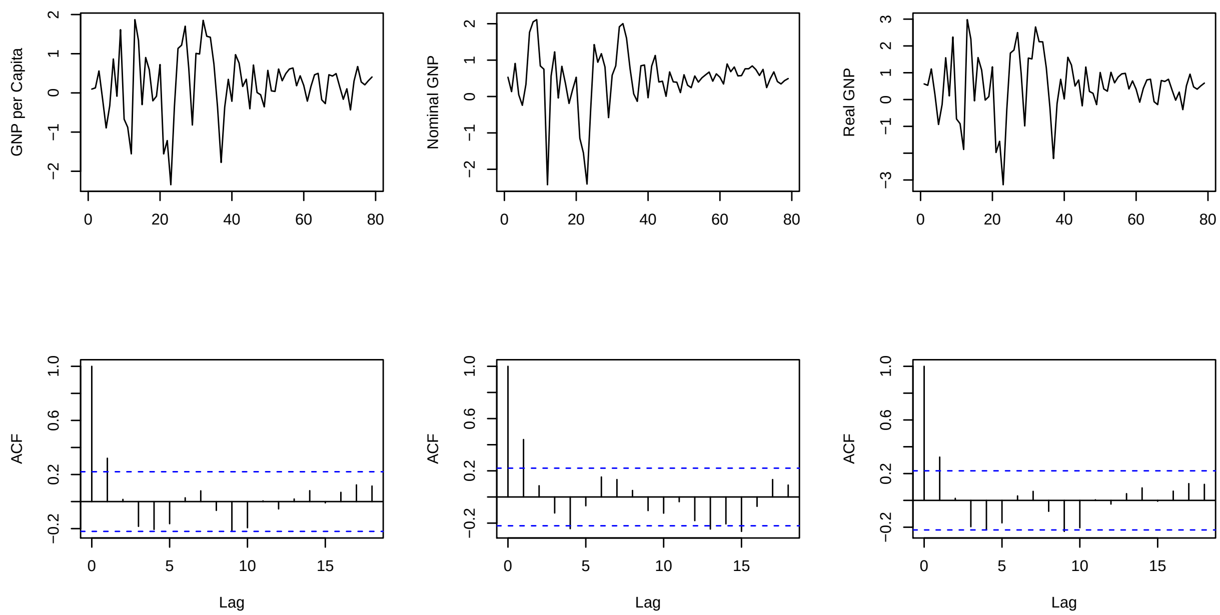
```
par(mfrow=c(2,3))
plot(Y_log[,1],type="l",xlab="",ylab="Log",main="GNP per Capita")
plot(Y_log[,2],type="l",xlab="",ylab="Log",main="Nominal GNP")
plot(Y_log[,3],type="l",xlab="",ylab="Log",main="Real GNP")
acf(Y_log[,1],main="")
acf(Y_log[,2],main="")
acf(Y_log[,3],main="")
```



Same goes for the log-transformation.

Log-Transformation rates:

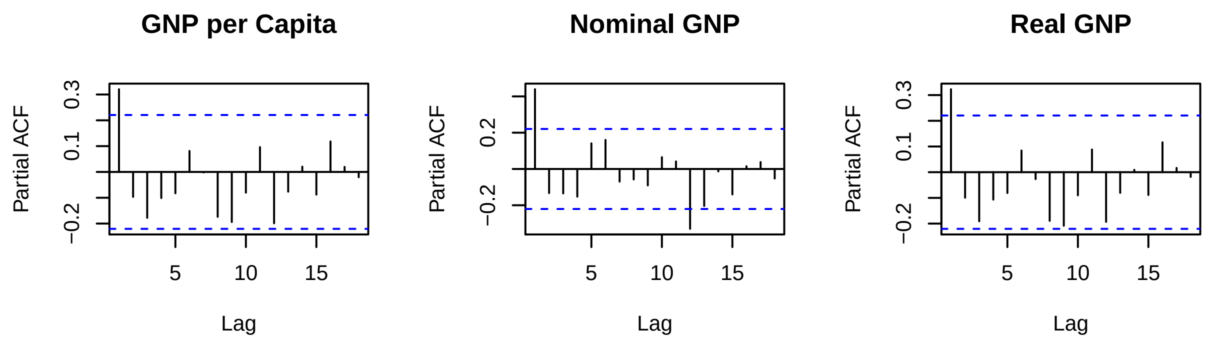
```
par(mfrow=c(2,3))
plot(Y_rate[,1],type="l",xlab="",ylab="GNP per Capita")
plot(Y_rate[,2],type="l",xlab="",ylab="Nominal GNP")
plot(Y_rate[,3],type="l",xlab="",ylab="Real GNP")
acf(Y_rate[,1],main="")
acf(Y_rate[,2],main="")
acf(Y_rate[,3],main="")
```



For the rates we can find stationary for all three GNP series. For all three the autocorrelation vanhishes with a lag of 3 which results in $q = 2$ for the MA.

Looking at the partial autocorrelation we find the following:

```r
par(mfrow=c(1,3))
pacf(Y_rate[,1], main="GNP per Capita")
pacf(Y_rate[,2], main="Nominal GNP")
pacf(Y_rate[,3], main="Real GNP")
```



The GNP partial autocorrelation vanishes after a lag of 2, which results in $p = 1$ for the AR part.

ARMA

We can create an ARMA model for each series individually.

GNP per Capita:

```r
arma.1 <- arma(Y_rate[,1], order = c(1, 2))
summary(arma.1)
```

```
##
## Call:
## arma(x = Y_rate[, 1], order = c(1, 2))
##
## Model:
## ARMA(1,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.08533 -0.40616  0.07057  0.40935  2.05279
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1         -0.5169      0.8175   -0.632    0.527
## ma1          0.8642      0.7989    1.082    0.279
## ma2          0.2704      0.2269    1.192    0.233
## intercept    0.3384      0.2444    1.384    0.166
##
## Fit:
## sigma^2 estimated as 0.5651,  Conditional Sum-of-Squares = 42.95,  AIC = 187.11
```

Nominal GNP:

```r
arma.2 <- arma(Y_rate[,2], order = c(1, 2))
summary(arma.2)
```

```
##
## Call:
## arma(x = Y_rate[, 2], order = c(1, 2))
##
## Model:
## ARMA(1,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.01390 -0.22814  0.06308  0.26984  1.48959
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1         0.08394     0.45566    0.184   0.8538
## ma1         0.40260     0.44142    0.912   0.3617
## ma2         0.10387     0.18602    0.558   0.5766
## intercept   0.46341     0.26048    1.779   0.0752 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 0.4739,  Conditional Sum-of-Squares = 36.02,  AIC = 173.21
```

Real GNP:

```r
arma.3 <- arma(Y_rate[,3], order = c(1, 2))
summary(arma.3)
```

```
##
## Call:
## arma(x = Y_rate[, 3], order = c(1, 2))
##
## Model:
## ARMA(1,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.61164 -0.46023 -0.05682  0.27586  2.66355
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1         0.80636          NA       NA       NA
## ma1        -0.62229          NA       NA       NA
## ma2        -0.58908          NA       NA       NA
## intercept   0.06485          NA       NA       NA
##
## Fit:
## sigma^2 estimated as 0.8679,  Conditional Sum-of-Squares = 67.64,  AIC = 221
```

We see that for each ARMA model the fit is not perfect. Especially the model for the Real GNP shows flaws.

2.

VAR(1) model

```r
mod=VAR(Y_rate,1)
```

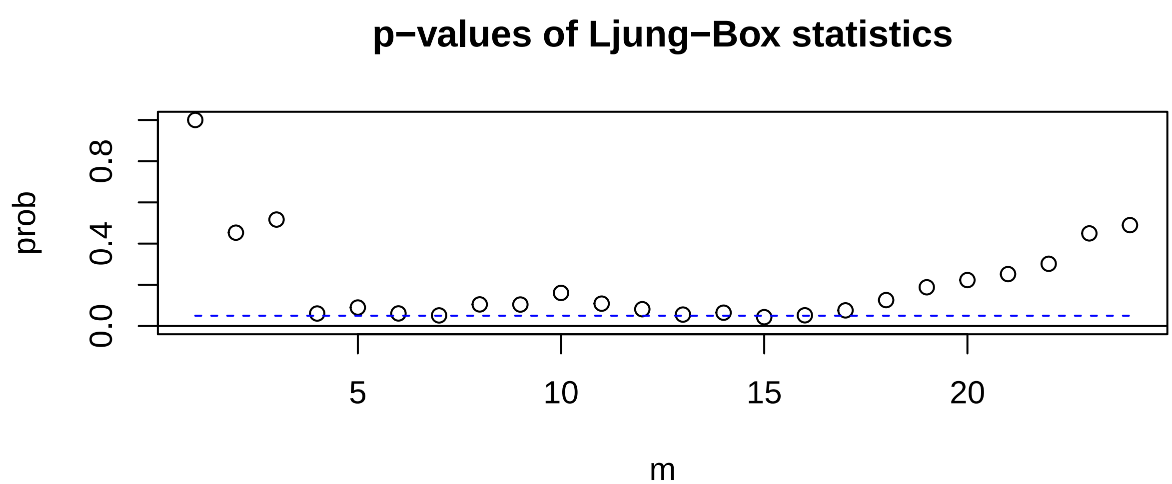
```
## Constant term:
## Estimates:  0.1857775 0.3354287 0.3012857
## Std.Error:  0.2122837 0.1959048 0.2912996
## AR coefficient matrix
## AR( 1 )-matrix
##        [,1]   [,2]   [,3]
## [1,]  0.303 -0.187  0.117
## [2,]  0.362  0.365 -0.197
## [3,] -0.651 -0.273  0.945
## standard error
##      [,1]  [,2]  [,3]
## [1,] 1.22 0.184 0.887
## [2,] 1.13 0.170 0.819
## [3,] 1.67 0.252 1.218
##
## Residuals cov-mtx:
##           [,1]      [,2]      [,3]
```

```
## [1,] 0.5515589 0.4168790 0.7556839
## [2,] 0.4168790 0.4697305 0.5773254
## [3,] 0.7556839 0.5773254 1.0385766
##
## det(SSE) =  0.0002539885
## AIC =  -8.050373
## BIC =  -7.780436
## HQ  =  -7.942228
```

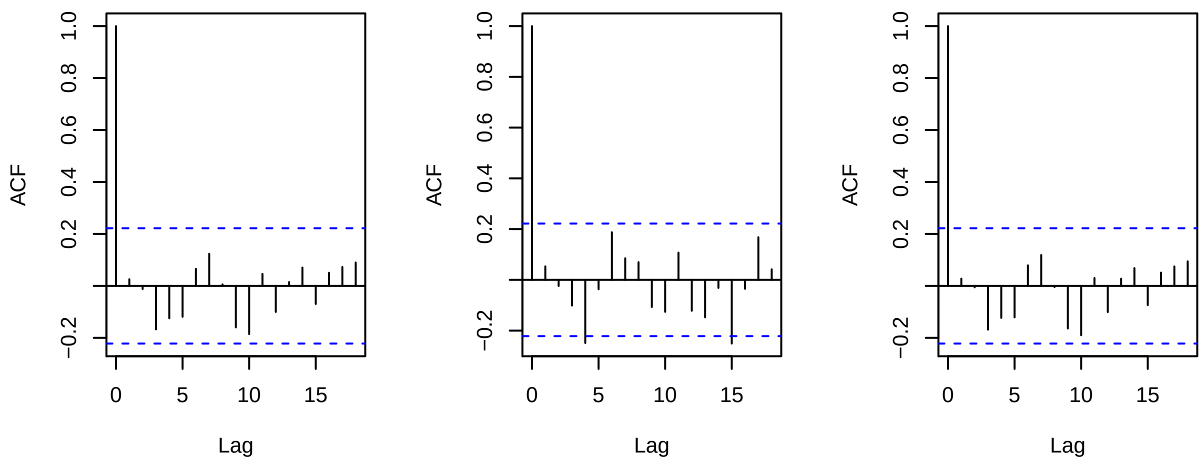
```
res=mod$residuals
```

Checking the WN assumption:

```
mq(res,adj=1*3^2)
```




```
par(mfrow=c(1,3))
acf(res[,1], main="")
acf(res[,2], main="")
acf(res[,3], main="")
```




```
VARorder(Y_rate) # Selected order is 1
mod2=refVAR(mod,thres=1.96)       #remove non significant coefficients using t stats
mod$aic
mod2$aic
```

Considering the AIC and BIC the reduced model performs better.

```
pred1 <- VARpred(mod,1)
pred2 <- VARpred(mod2,1)
rmse <- rbind(mod1=pred1$rmse, mod2=pred2$rmse)
rownames(rmse) <- c("model1", "model2")
```

```
##             [,1]      [,2]     [,3]
## model1 0.7612397 0.7025057 1.044587
## model2 0.7804935 0.7043776 1.054106
```

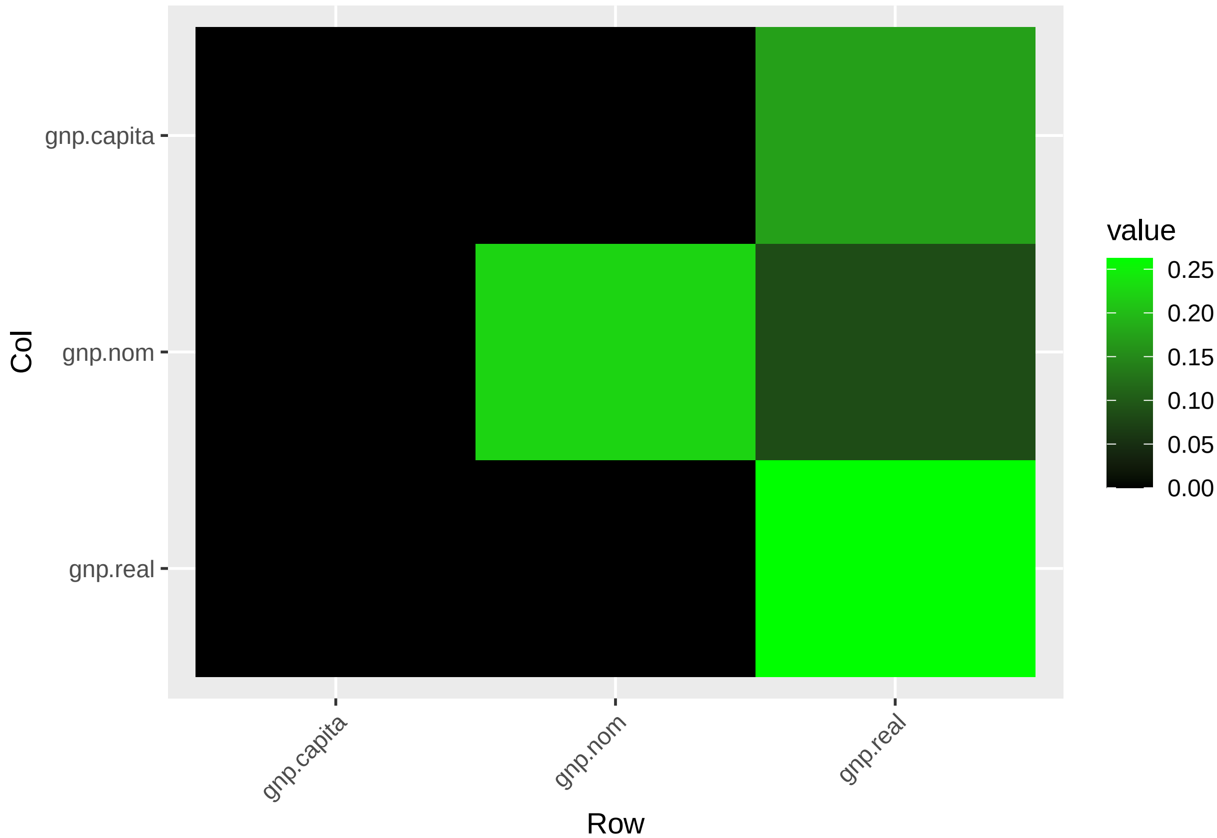
We can see that the prediction is better for the full model (mod1). But the difference is rather small. It might make sense to consider the simpler model (mod2) then.

3. VAR with LASSO

```
mod_lasso=fitVAR(Y_rate,p=1,penalty="ENET",method="cv")
```

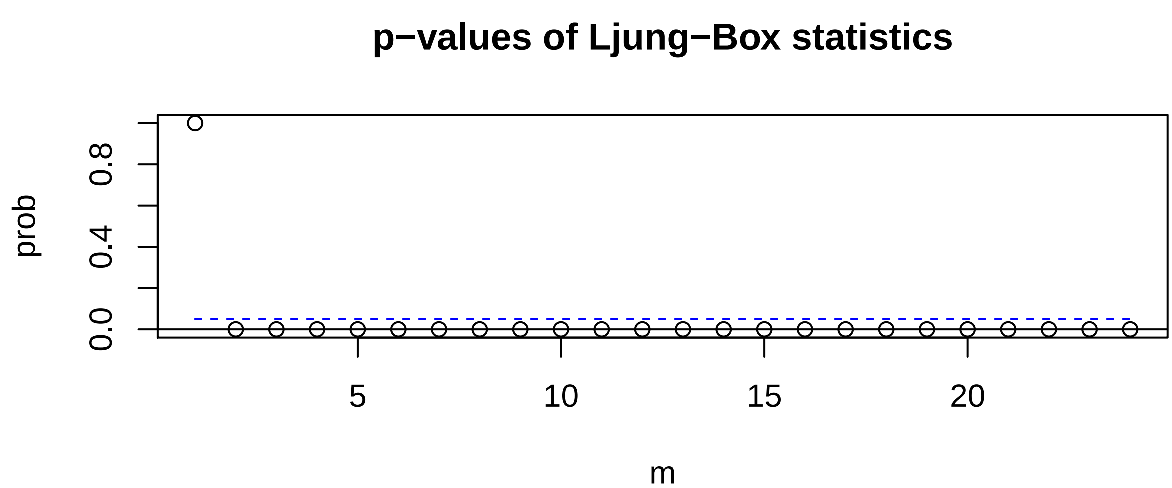
When we look at the coefficients we see that only the coefficients for the real GNP are of a considerable amplitude.

```
coef=mod_lasso$A;A1lasso=coef[[1]]
plotMatrix(A1lasso)
```

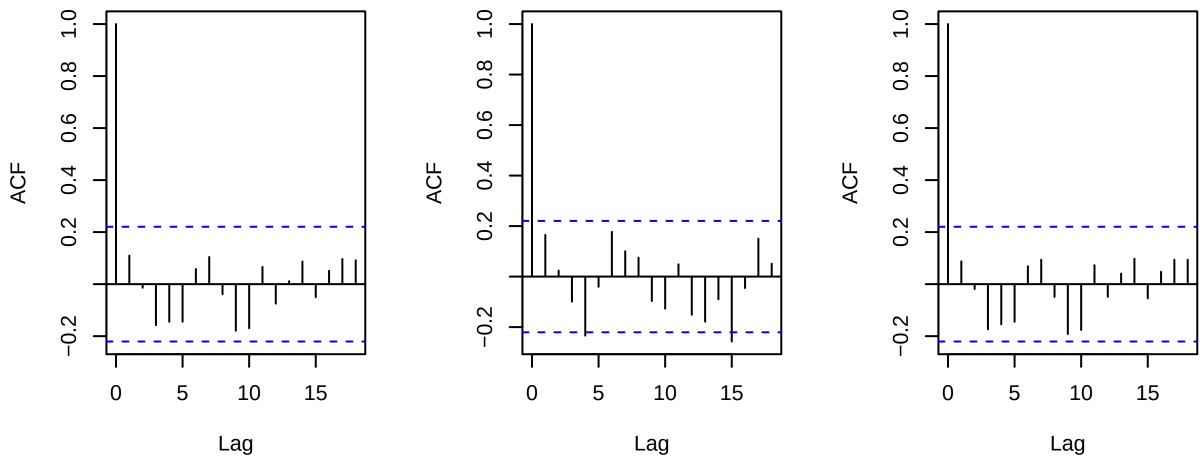


Checking the WN assumption

```
res_lasso=mod_lasso$residuals

mq(res_lasso,adj=1*3^2)
```



```r
par(mfrow=c(1,3))
acf(res_lasso[,1], main="")
acf(res_lasso[,2], main="")
acf(res_lasso[,3], main="")
```



We see that the White Noise assumption does hold for all three series.

Comparison with the simple VAR

```r
mod_lasso$A
```

```
## [[1]]
##             gnp.capita   gnp.nom   gnp.real
## gnp.capita           0 0.0000000 0.17198365
## gnp.nom              0 0.2231456 0.08412493
## gnp.real             0 0.0000000 0.26247845
```

I don't know :(