PS 4

Xiang (Cyllia) Li

2024-02-20

```r
source("create_monthly_price.R")
source("model_selection_function.R") # function for model selection
source("ols_function.R")
source("t_test_function.R")
source("generate_data_functions.R")
source("expanding_window_forecast_function.R") # function for forecasti
ng using expanding windows
source("rolling_window_forecast_function.R") # function for forecasting
 using rolling windows
library("quantmod")
```

Fetch Data

```r
getSymbols(Symbols ="GNP",src = "FRED",warnings = FALSE)
```

```
## [1] "GNP"
```

```r
Y <- as.matrix(GNP[,1])
DY <- diff(Y)/Y[1:(dim(Y)[1]-1),]
head(DY)
```

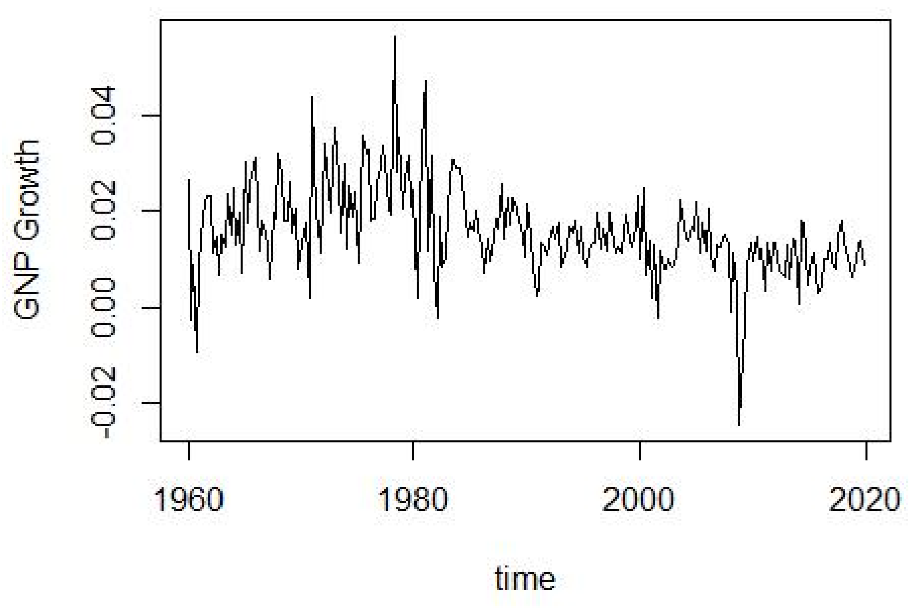
```
##                   GNP
## 1947-04-01 0.01196435
## 1947-07-01 0.01478570
## 1947-10-01 0.04094274
## 1948-01-01 0.02357260
## 1948-04-01 0.02587849
## 1948-07-01 0.02420397
```

```r
tail(DY)
```

```
##                    GNP
## 2022-04-01 0.022580326
## 2022-07-01 0.017074777
## 2022-10-01 0.015584918
## 2023-01-01 0.014233663
## 2023-04-01 0.009771869
## 2023-07-01 0.019756463
```

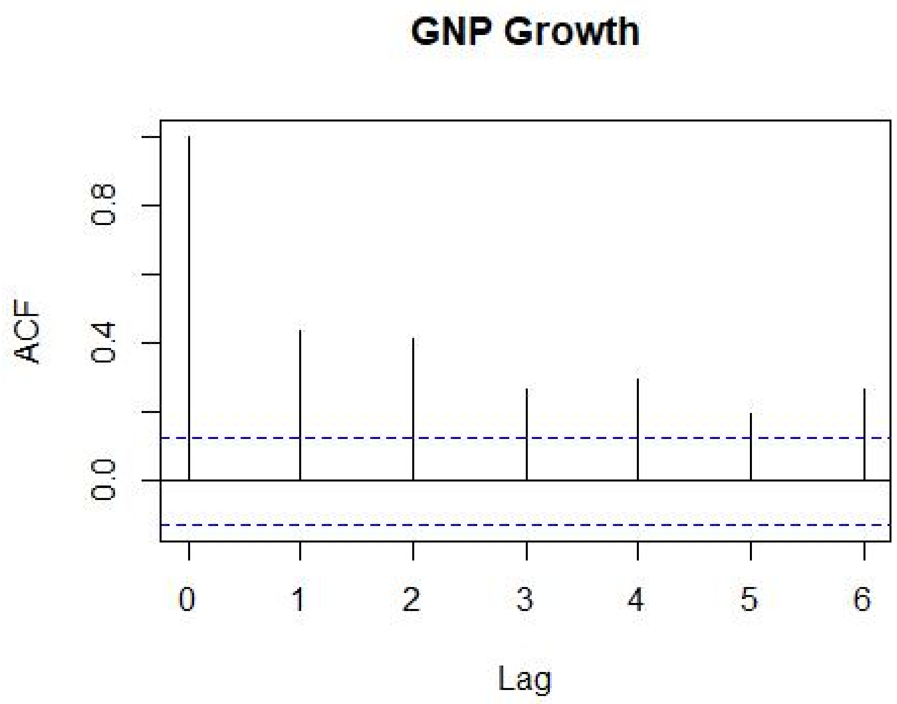
```r
keep_data <- seq(from = as.Date("1960-01-01"), to = as.Date("2019-10-1
"), by = "quarter")
Y_new = as.matrix(Y[as.Date(rownames(Y)) %in% keep_data,]) # precovid d
ata
```

```
DY_new = as.matrix(DY[as.Date(rownames(DY)) %in% keep_data,])
colnames(DY_new) = "GNP Growth"
n_obs = dim(DY_new)[1]
DY_new_date = as.Date(row.names(DY_new))

plot(x = DY_new_date, y = DY_new,xlab='time',ylab='GNP Growth',type='l',
col="black")
```



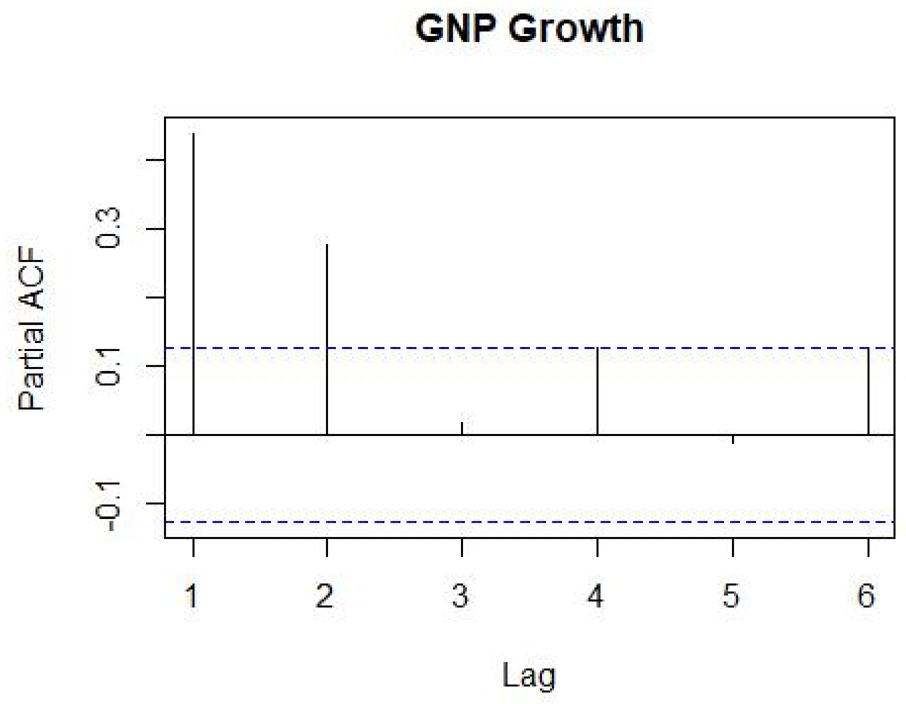
```
basicStats(DY_new)

##                GNP.Growth
## nobs           240.000000
## NAs              0.000000
## Minimum         -0.024656
## Maximum          0.056421
## 1. Quartile      0.010223
## 3. Quartile      0.020117
## Mean             0.015709
## Median           0.014341
## Sum              3.770248
## SE Mean          0.000617
## LCL Mean         0.014494
## UCL Mean         0.016925
## Variance         0.000091
## Stdev            0.009560
## Skewness         0.368624
## Kurtosis         2.576997
```

```
acf(DY_new,lag=round(n_obs^(1/3))) # command to obtain sample ACF of th
e data
```



```
Box.test(DY_new, lag = round(n_obs^(1/3)), type = "Ljung-Box") # applyi
ng Ljung and Box (1978) joint test of auto correlations

##
##  Box-Ljung test
##
## data:  DY_new
## X-squared = 154.78, df = 6, p-value < 2.2e-16
```

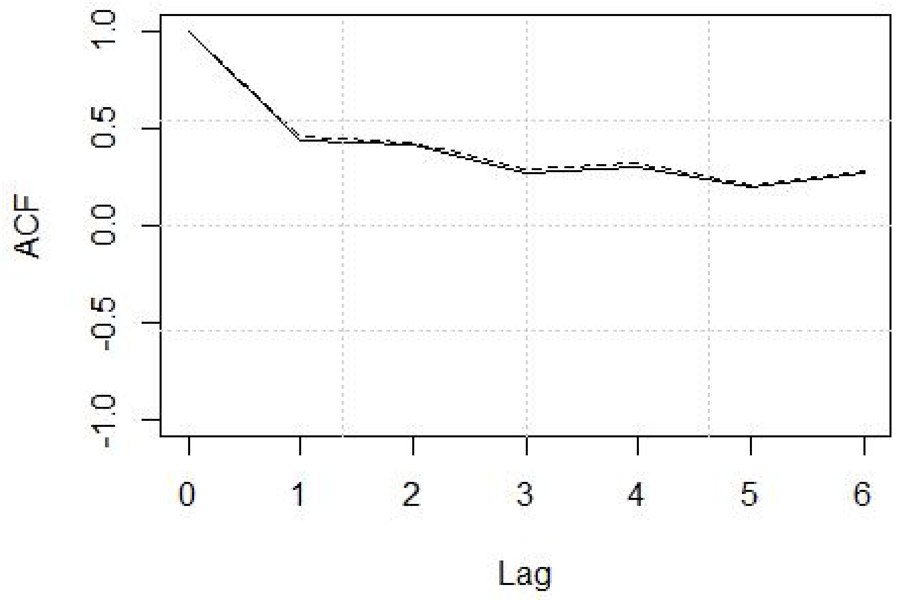
```
pacf(DY_new,lag=round(n_obs^(1/3)),main="GNP Growth") # command to obta
in sample PACF of the data
```



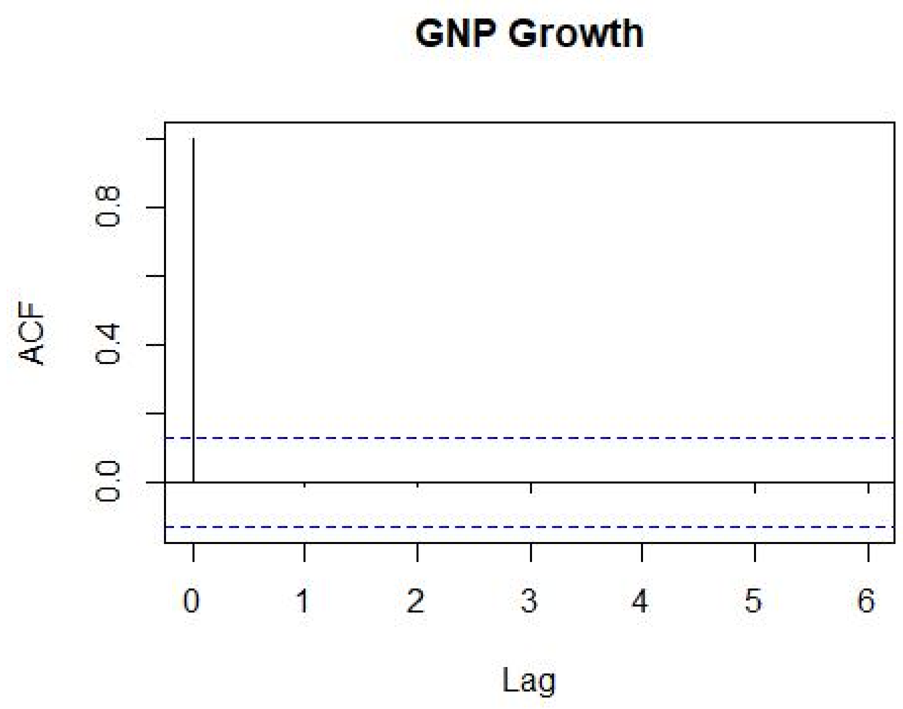
select the number of lags and model checking

```
results <- model_selection(round(n_obs^(1/3)),DY_new)
aic_values = results$AIC
bic_values = results$BIC
num_lags_aic = results$op_lag_AIC
num_lags_bic = results$op_lag_BIC
num_lags_aic

## [1] 6

num_lags_bic

## [1] 2

num_lags = num_lags_aic
lags_DY_new = matrix(NA,nrow = n_obs, ncol = num_lags)
for (i in 1:num_lags) {
  lags_DY_new[(i+1):n_obs,i] = as.matrix(DY_new[1:(n_obs-i),1])
}
intercept = matrix(1,n_obs)
X = cbind(intercept,lags_DY_new)
y = DY_new
reg_result = ols(X[(num_lags+1):n_obs,],as.matrix(y[(num_lags+1):n_obs,
1]))
beta_hat = reg_result$beta_hat
```

```
ar_coeff <- as.numeric(beta_hat[2:(num_lags+1)])
ma_coeff <- 0
ACF = acf(DY_new,lag=round(n_obs^(1/3)),plot = FALSE) # command to obta
in sample ACF of the data
TACF <- ARMAacf(ar_coeff, ma_coeff, lag.max = round(n_obs^(1/3))) # com
mand to obtain theorical ACF
plot(c(0:round(n_obs^(1/3))),ACF$acf,type='l',xlab='Lag',ylab='ACF',yli
m=c(-1,1))
lines(0:round(n_obs^(1/3)),TACF,lty=2)
grid(nx = 4, ny = 4)
```



```
residuals = reg_result$u_hat # get the AR model residuals
acf(residuals,lag=round(n_obs^(1/3)),main = "GNP Growth") # command to
obtain sample ACF of the data
```

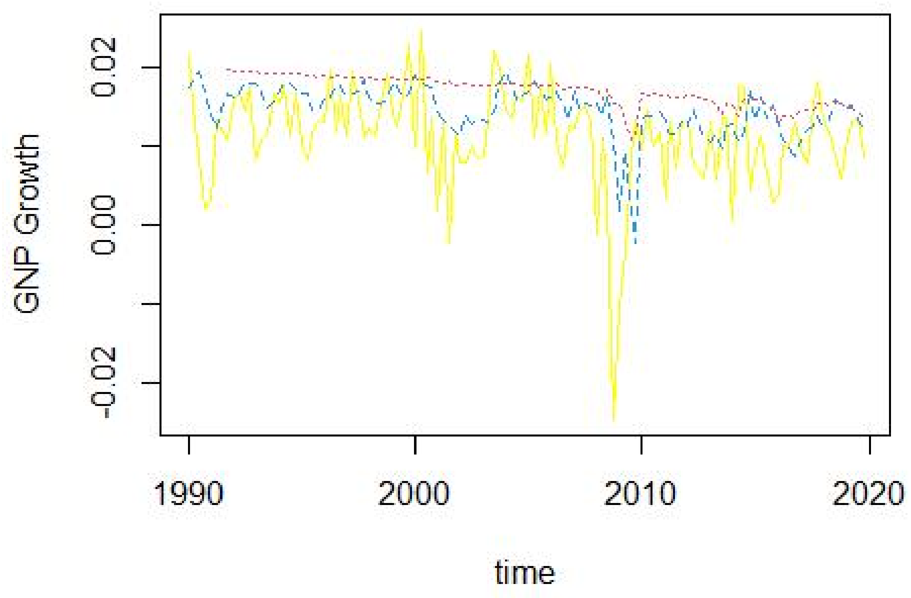


```
Box.test(residuals, lag = round(n_obs^(1/3)), type = "Ljung-Box") # app
lying Ljung and Box (1978) joint test of auto correlations

##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 0.6574, df = 6, p-value = 0.9954
```

#1. Expanding windows

```
## Forecasting GNP growth using expanding windows
## Using AIC
lag_choice = NA
init_win_len = 120 # the first 30 years
num_step_ahead = 8 # 1 to 8 steps ahead forecastes
prediction_results = expanding_window(y = DY_new, init_win_len = init_w
in_len, pre_sel_num_lags = lag_choice, num_step_ahead = num_step_ahead,
 sel_method = 'aic')
yhat_f_aic <- prediction_results$forecast

y_f_aic <- prediction_results$actual_value

## Plot
plot(x = DY_new_date[121:n_obs], y = y_f_aic, xlab='time',ylab='GNP Gro
wth',type='l',col="yellow")
```

```
lines(x = DY_new_date[121:n_obs],y = yhat_f_aic[,1],lty=2, col = 4)
lines(x = DY_new_date[121:n_obs],y = yhat_f_aic[,8],lty=3, col = 2)
```

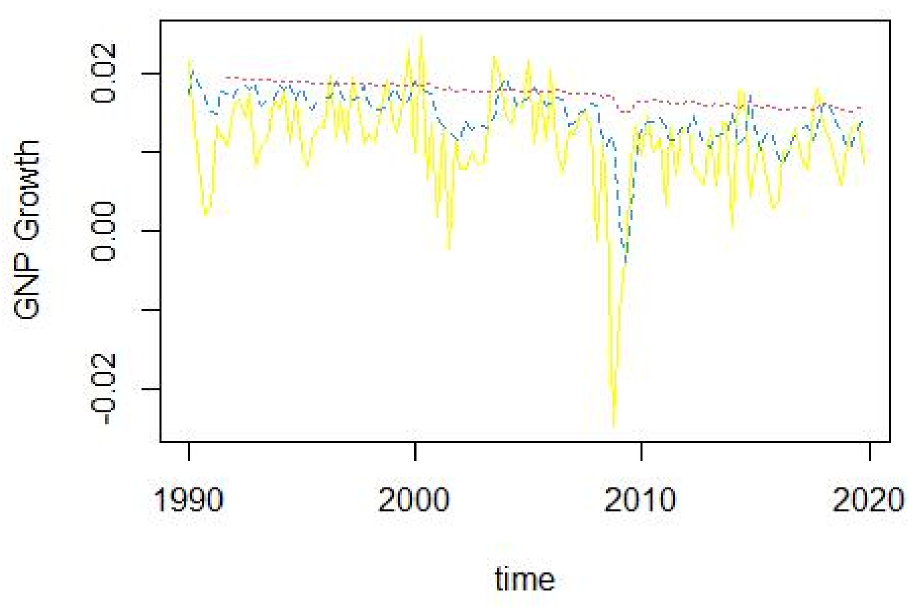


```
## Root mean square forecast errors
forecast_error =  kronecker(matrix(1,ncol = num_step_ahead),y_f_aic) -
yhat_f_aic
rmsfe_ar_aic = sqrt(colMeans(forecast_error^2, na.rm = TRUE, dims = 1))
rmsfe_ar_aic

## [1] 0.007000978 0.007180999 0.007735637 0.007892120 0.008108322 0.00
8156490
## [7] 0.008342269 0.008432837
```

##2. Using BIC

```
## Forecasting GNP growth using expanding windows
## Using BIC
lag_choice = NA
init_win_len = 120 # the first 8 years
num_step_ahead = 8 # 1 to 8 steps ahead forecastes
prediction_results = expanding_window(y = DY_new, init_win_len = init_w
in_len, pre_sel_num_lags = lag_choice, num_step_ahead = num_step_ahead,
 sel_method = 'bic')
yhat_f_bic <- prediction_results$forecast

y_f_bic <- prediction_results$actual_value
```

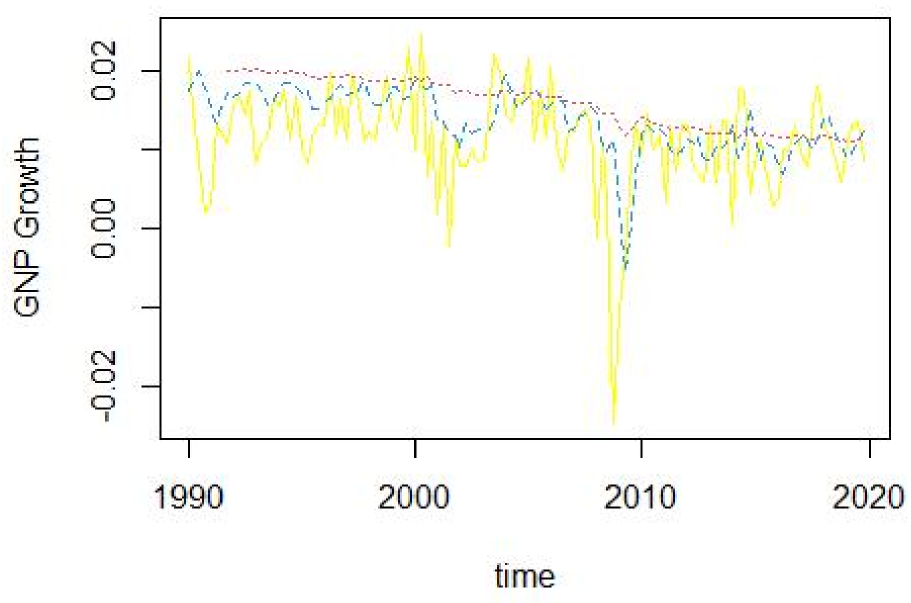
```
## Plot
plot(x = DY_new_date[121:n_obs], y = y_f_bic, xlab='time',ylab='GNP Gro
wth',type='l',col="yellow")
lines(x = DY_new_date[121:n_obs],y = yhat_f_bic[,1],lty=2, col = 4)
lines(x = DY_new_date[121:n_obs],y = yhat_f_bic[,8],lty=3, col = 2)
```



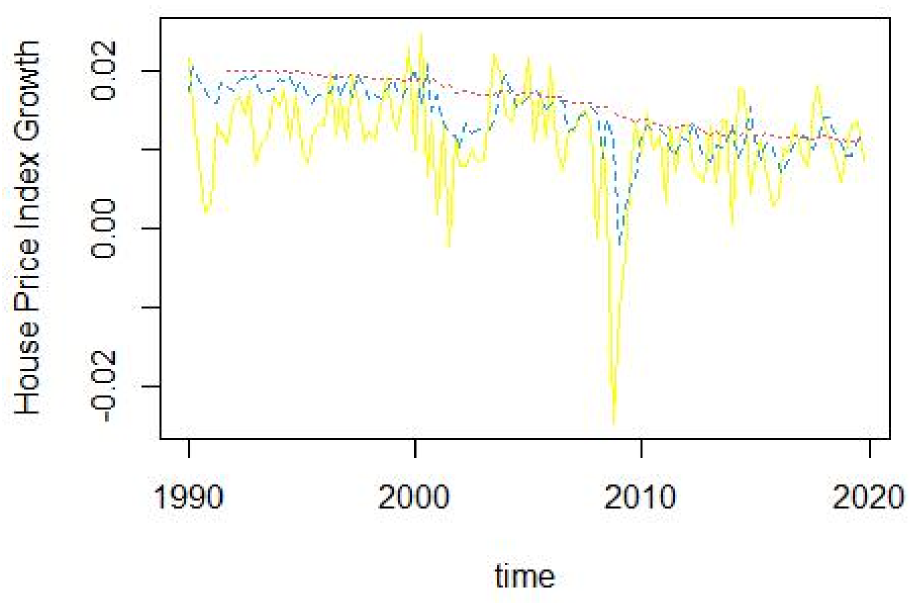
```
## Root mean square forecast errors
forecast_error =  kronecker(matrix(1,ncol = num_step_ahead),y_f_bic) -
yhat_f_bic
rmsfe_ar_bic = sqrt(colMeans(forecast_error^2, na.rm = TRUE, dims = 1))
rmsfe_ar_bic

## [1] 0.006768129 0.007327876 0.007891828 0.008170631 0.008322748 0.00
8401066
## [7] 0.008560302 0.008668237

## Aerage
yhat_f_ave = (yhat_f_aic + yhat_f_bic)/2
forecast_error =  kronecker(matrix(1,ncol = num_step_ahead),y_f_bic) -
yhat_f_ave
rmsfe_ave = sqrt(colMeans(forecast_error^2, na.rm = TRUE, dims = 1))
rmsfe_ave

## [1] 0.006800907 0.007213739 0.007799146 0.008014225 0.008197299 0.00
8265888
## [7] 0.008442060 0.008540591
```

#3. Rolling windows

```
## AIC
lag_choice = NA
init_win_len = 120 # the first 30 years
num_step_ahead = 8 # 1 to 8 steps ahead forecastes
prediction_results = rolling_window(y = DY_new, init_win_len = init_win
_len, pre_sel_num_lags = lag_choice, num_step_ahead = num_step_ahead, s
el_method = 'aic')
yhat_f_aic <- prediction_results$forecast

y_f_aic <- prediction_results$actual_value

## Plot
plot(x = DY_new_date[121:n_obs], y = y_f_aic,xlab='time',ylab='GNP Grow
th',type='l',col="yellow")
lines(x = DY_new_date[121:n_obs],y = yhat_f_aic[,1],lty=2, col = 4)
lines(x = DY_new_date[121:n_obs],y = yhat_f_aic[,8],lty=3, col = 2)
```



```
## Root mean square forecast errors
forecast_error =  kronecker(matrix(1,ncol = num_step_ahead),y_f_aic) -
yhat_f_aic
rmsfe_ar_aic = sqrt(colMeans(forecast_error^2, na.rm = TRUE, dims = 1))
rmsfe_ar_aic

## [1] 0.006532133 0.006869404 0.007297877 0.007496990 0.007556114 0.00
7561955
## [7] 0.007672865 0.007741224
## BIC
lag_choice = NA
init_win_len = 120 # the first 30 years
num_step_ahead = 8 # 1 to 24 steps ahead forecastes
prediction_results = rolling_window(y = DY_new, init_win_len = init_win
_len, pre_sel_num_lags = lag_choice, num_step_ahead = num_step_ahead, s
el_method = 'bic')
yhat_f_bic <- prediction_results$forecast

y_f_bic <- prediction_results$actual_value

## Plot
plot(x = DY_new_date[121:n_obs], y = y_f_bic,xlab='time',ylab='House Pr
ice Index Growth',type='l',col="yellow")
lines(x = DY_new_date[121:n_obs],y = yhat_f_bic[,1],lty=2, col = 4)
lines(x = DY_new_date[121:n_obs],y = yhat_f_bic[,8],lty=3, col = 2)
```

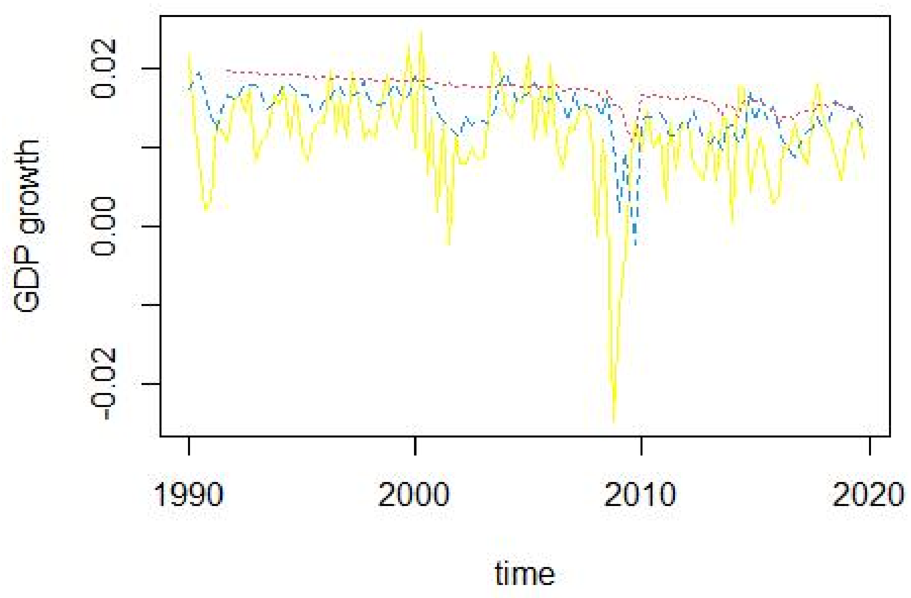


```
forecast_error =  kronecker(matrix(1,ncol = num_step_ahead),y_f_bic) -
yhat_f_bic
rmsfe_ar_bic = sqrt(colMeans(forecast_error^2, na.rm = TRUE, dims = 1))
rmsfe_ar_bic

## [1] 0.006767772 0.007373192 0.007748851 0.007883883 0.007847288 0.00
7784938
## [7] 0.007838395 0.007868308
```

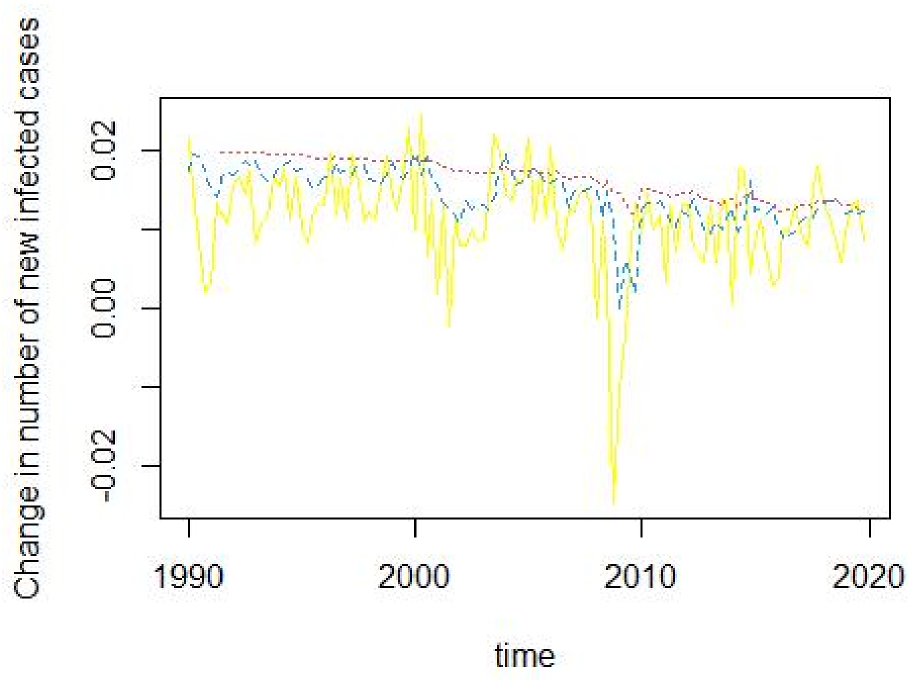
```
yhat_f_ave = (yhat_f_aic + yhat_f_bic)/2
forecast_error =  kronecker(matrix(1,ncol = num_step_ahead),y_f_bic) -
yhat_f_ave
rmsfe_ave = sqrt(colMeans(forecast_error^2, na.rm = TRUE, dims = 1))
rmsfe_ave

## [1] 0.006612314 0.007095963 0.007506544 0.007679328 0.007695086 0.00
7669349
## [7] 0.007753135 0.007803207

rmsfe_all_rolling = rbind(rmsfe_ar_aic,rmsfe_ar_bic,rmsfe_ave)
rmsfe_all_rolling

##                     [,1]        [,2]        [,3]        [,4]
[,5]
## rmsfe_ar_aic 0.006532133 0.006869404 0.007297877 0.007496990 0.00755
6114
## rmsfe_ar_bic 0.006767772 0.007373192 0.007748851 0.007883883 0.00784
7288
## rmsfe_ave    0.006612314 0.007095963 0.007506544 0.007679328 0.00769
5086
##                     [,6]        [,7]        [,8]
## rmsfe_ar_aic 0.007561955 0.007672865 0.007741224
## rmsfe_ar_bic 0.007784938 0.007838395 0.007868308
## rmsfe_ave    0.007669349 0.007753135 0.007803207
```

#4.

```
## Forecasting using AR Model with aic selected number of lags
lag_choice = NA
init_win_len = 120 # the first 30 years
num_step_ahead = 8 # 1 to 8 steps ahead forecastes
prediction_results = expanding_window(y = DY_new, init_win_len = init_w
in_len, pre_sel_num_lags = lag_choice, num_step_ahead = num_step_ahead,
 sel_method = 'aic')
y_f_aic <- prediction_results$actual_value
yhat_f_aic <- prediction_results$forecast
selected_num_lags <- prediction_results$sel_num_lags

plot(x = DY_new_date[121:n_obs], y = y_f_aic,xlab='time',ylab='GDP grow
th',type='l',col="yellow")
lines(x = DY_new_date[121:n_obs],y = yhat_f_aic[,1],lty=2, col = 4)
lines(x = DY_new_date[121:n_obs],y = yhat_f_aic[,8],lty=3, col = 2)
```



```
forecast_error =  kronecker(matrix(1,ncol = num_step_ahead),y_f_aic) -
yhat_f_aic
rmsfe_ar_aic = sqrt(colMeans(forecast_error^2, na.rm = TRUE, dims = 1))

# Forecasting by Model averaging and RMSFE comparison
yhat_f_ave = (yhat_f_aic + yhat_f_bic)/2
forecast_error =  kronecker(matrix(1,ncol = num_step_ahead),y_f_bic) -
yhat_f_ave
rmsfe_ave = sqrt(colMeans(forecast_error^2, na.rm = TRUE, dims = 1))

plot(x = DY_new_date[121:n_obs], y = y_f_bic,xlab='time',ylab='Change i
n number of new infected cases',type='l',col="yellow")
lines(x = DY_new_date[121:n_obs],y = yhat_f_ave[,1],lty=2, col = 4)
lines(x = DY_new_date[121:n_obs],y = yhat_f_ave[,7],lty=3, col = 2)
```



```
rmsfe_all = rbind(rmsfe_ar_aic,rmsfe_ar_bic,rmsfe_ave)
rmsfe_all

##                     [,1]        [,2]        [,3]        [,4]
[,5]
## rmsfe_ar_aic 0.007000978 0.007180999 0.007735637 0.007892120 0.00810
8322
## rmsfe_ar_bic 0.006767772 0.007373192 0.007748851 0.007883883 0.00784
7288
## rmsfe_ave    0.006786340 0.007192845 0.007680859 0.007826290 0.00791
4883
##                     [,6]        [,7]        [,8]
## rmsfe_ar_aic 0.008156490 0.008342269 0.008432837
## rmsfe_ar_bic 0.007784938 0.007838395 0.007868308
## rmsfe_ave    0.007910045 0.008028727 0.008088655
```