

Time Series - Homework 2

Kristóf Reizinger

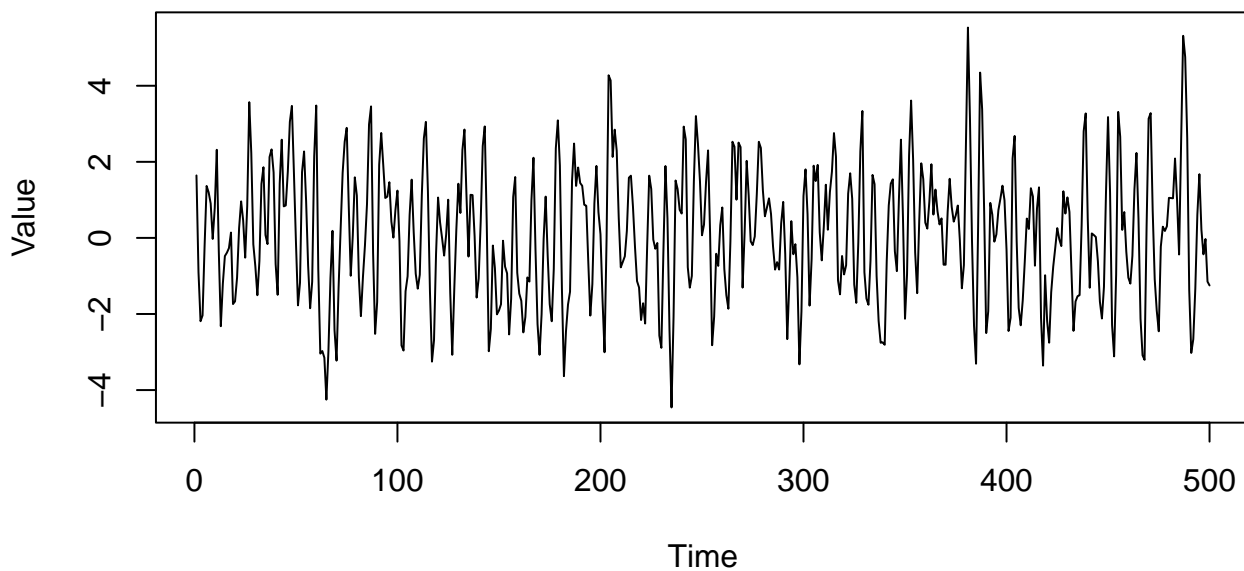
17-05-2022

Exercise 4.,

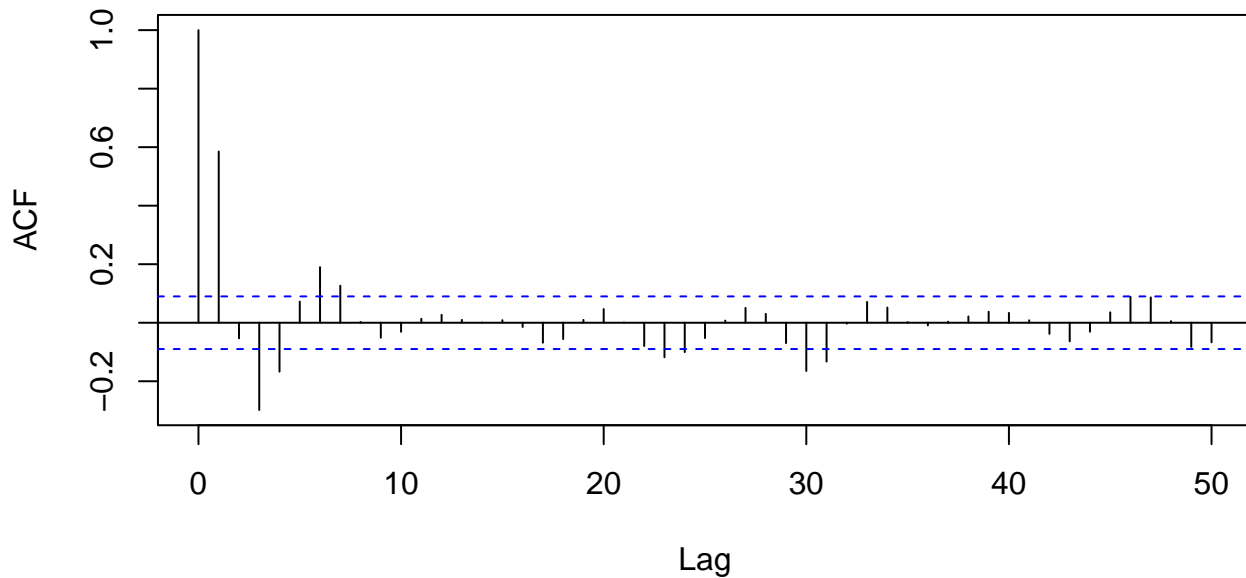
Part a.,

- a) Graph the sample autocorrelation function along with the approximate 95% confidence bands. Graph the sample partial autocorrelation function along with the approximate 95% confidence bands. Make an educated guess about the values of p and q based on these graphs.

Time series plot



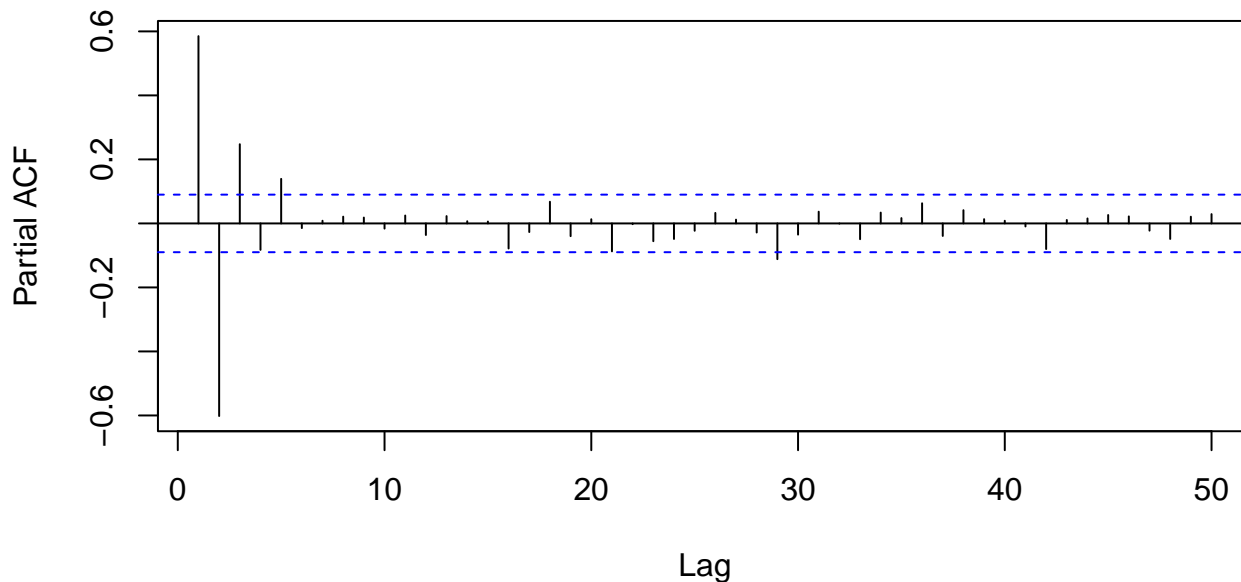
Autocorrelation function



Answer:

It is hard to guess the parameters; thus, almost all of the first seven autocorrelation values are significant, but there are significant higher-order autocorrelation values. (For instance, around 30 lags.) But it cannot be a long-memory process (like a simple AR process with coefficients close to one) , because the autocorrelation function has an exponential decay. The exact p-order of the AR process is complicated to guess due to the outlier ACF values. Considering the roughly exponential decay, I can hypothesize that the MA part of the ARMA model is less than 7. (To guess the order of the AR process, I check the partial autocorrelation function.)

Partial autocorrelation function



Answer:

The autocorrelation function has an exponential decay too. The first four PACF values are almost all significant (the third one is very close to the lower bound of the 95% confidence interval but still non-significant), which suggests that the order of the AR(p) process is less than four.

Considering the autocorrelation and the partial autocorrelation functions, the “biggest model” I would try to fit is an ARMA(4,7). However, the orders seem to be very high, and if I consider only the most significant autocorrelation and partial autocorrelation values, then I would guess that $p = 2$ (thus, the first two lags of the PACF are highly significant, and the significant lags in the PACF indicate the order of the AR process) and $q=3$ (thus the first and the third lags of the ACF are highly significant, but the second one not, so I may not include any lags after the second, insignificant lag). (The significant lags of the ACF indicate the MA process’s order). All in all, I could imagine an ARMA(2,3) or an ARMA(2,1) process which fits the data.

Part b.,

- Use the Bayesian information criterion (BIC) to find the values of p and q that fit the data best. You can summarize the candidate models in a table. Did your initial guess hold up? (Eviews calls BIC the “Schwartz criterion”).

Answer:

I am calculating all possible ARMA models below and I select the best fitting one based on the lowest BIC value. My initial guess was near to the best model, but not perfect. I suggested ARMA(2,3) or ARMA(2,1) models, and my second guess met the data.

In case of ARMA models, it is hard to guess the parameters, because the AR(p) has an exponential decay in PACF and a long memory in ACF, while the ACF of the MA(q) has an exponential decay and the PACF is decreasing gradually.

```
# generate all cases
#p=0:4;d=0 (process is not integrated);q=0:4
cases <- expand.grid(rep(list(0:4), 2))
```

```

cases_d<-cbind(cases$Var1,rep(0,dim(cases)[1]),cases$Var2)
# define BIC vector
BIC_vec<-rep(NA,dim(cases)[1])
# iterate on all models
for(i in 1:dim(cases)[1]){
fitted_model<-arima(arma$V1[1:475],order=cases_d[i,]) # fit the model
BIC_vec[i]<-BIC(fitted_model) # save BIC
}
#store the index of the model with lowest BIC
ind_best_model<-which.min(BIC_vec)
# index of the best model
ind_best_model

## [1] 8

# parameters of the best model ARIMA(p,d,q), d is set to 0
cases_d[ind_best_model,]

## [1] 2 0 1

# AR I(d) MA
#->ARMA(2,1)

# generate table
results<-cbind(1:length(BIC_vec),cases_d,BIC_vec)
results<-results[,-3]
colnames(results)<-c("Id","AR","MA", "BIC")

options("digits" = 3)
results %>% as_tibble %>% knitr::kable(caption= "ARMA model selection based on BIC", digits=5)

```

Table 1: ARMA model selection based on BIC

Id	AR	MA	BIC
1	0	0	1875
2	1	0	1681
3	2	0	1469
4	3	0	1442
5	4	0	1444
6	0	1	1522
7	1	1	1481
8	2	1	1431
9	3	1	1437
10	4	1	1443
11	0	2	1452
12	1	2	1458
13	2	2	1437
14	3	2	1443
15	4	2	1447
16	0	3	1458
17	1	3	1456
18	2	3	1442
19	3	3	1446
20	4	3	1452
21	0	4	1449
22	1	4	1452
23	2	4	1446

Id	AR	MA	BIC
24	3	4	1452
25	4	4	1458

The best-fitting model is ARMA(2,1), with the lowest BIC value (1431). My second guess was correct to the solution. The BIC value of the ARMA(2,3) model is 1442, which is close to the lowest value but not enough. So, I think it was a reasonable guess.

```
# fit the best model
ARMA_2_1_model<-arima(arma$V1[1:475],order=cases_d[ind_best_model,])
# calculate roots
# I check whether there are multiple roots or not
# AR
ar_roots<-polyroot(c(1,-ARMA_2_1_model$coef[1:2])) # selecting AR coefficients from the model
ar_roots

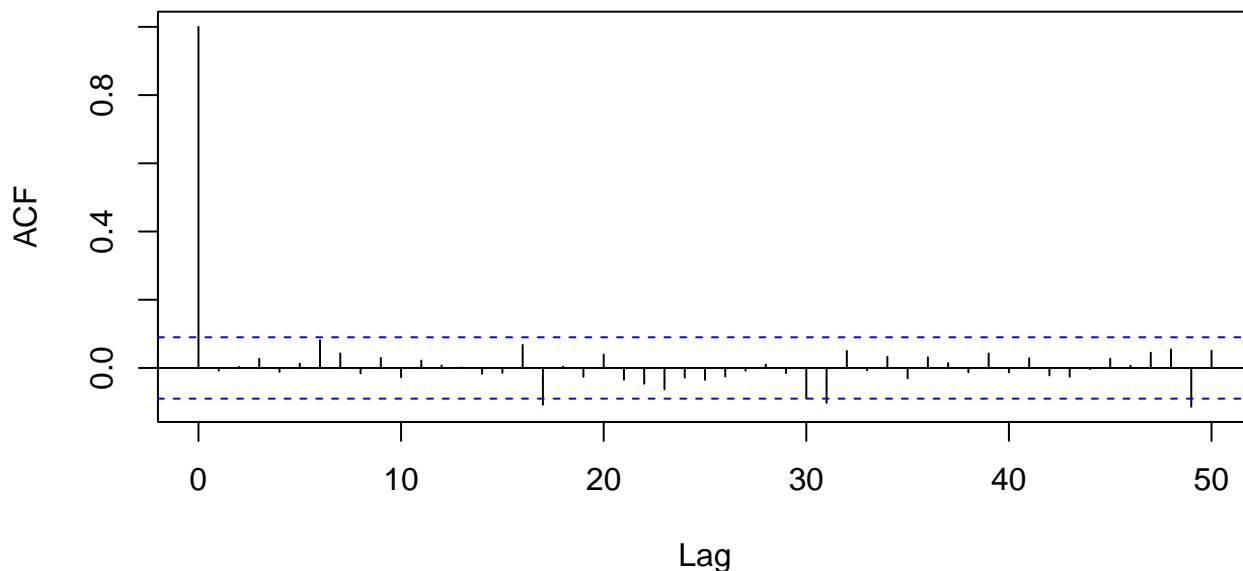
## [1] 0.75+1.31i 0.75-1.31i
abs(ar_roots)

## [1] 1.51 1.51
# MA
ma_roots<-polyroot(c(1,ARMA_2_1_model$coef[3])) # selecting MA coefficient from the model
ma_roots

## [1] -2.01+0i
abs(ma_roots)

## [1] 2.01
# ACF
acf(ARMA_2_1_model$residuals,50,main="Autocorrelation function of ARMA(2,1) residuals")
```

Autocorrelation function of ARMA(2,1) residuals



```
# ACF suggest a white noise process-> appropriate model
```

I have fitted an ARMA(2,1) process on the data. I have checked the AR $(0.75 + 1.31i, 0.75 - 1.31i)$ and MA roots (-2.01) . All of them are higher than one in absolute value ($|AR_{root1}| = |AR_{root2}| = 1.5 > 1$ and $|MA_{root1}| = 2 > 1$ -MA is by definition stationary), so the process is stationary. The model seems appropriate: no roots are close to one, and the MA root is not close to any of the AR roots. The ACF of the residuals suggest a white noise process, which means an appropriate model selection.

Part c.,

Now consider the last 25 observations as well: c) Using the estimated model, forecast the process 25 periods into the future. Produce a graph with the point forecast along with the ± 2 standard error bounds as well as the realizations. Comment on the accuracy and qualitative properties of the forecast.

Answer:

I have calculated the forecast and the confidence intervals. The forecast (darkblue line) reverts to the mean, which is around zero. The 475th data point, the last observation, was much lower than the mean, so it is not surprising that the model predicts an increase in the subsequent few periods, but after some time, the prediction converges to the mean. This is not surprising, considering Exercise 3. I have calculated (approximated) the limit of an ARMA and an AR model, and the calculations showed the reversion to the mean (~ 0) in the long-run.

The forecast is seemingly inaccurate; some observations/realizations are outside the ± 2 standard error bounds. The realizations should be inside the approximately 95% confidence interval in the majority of the cases. (So, from 25 values, about one can be outside the confidence interval, but two observations are outside, which means the forecast is less reliable.)

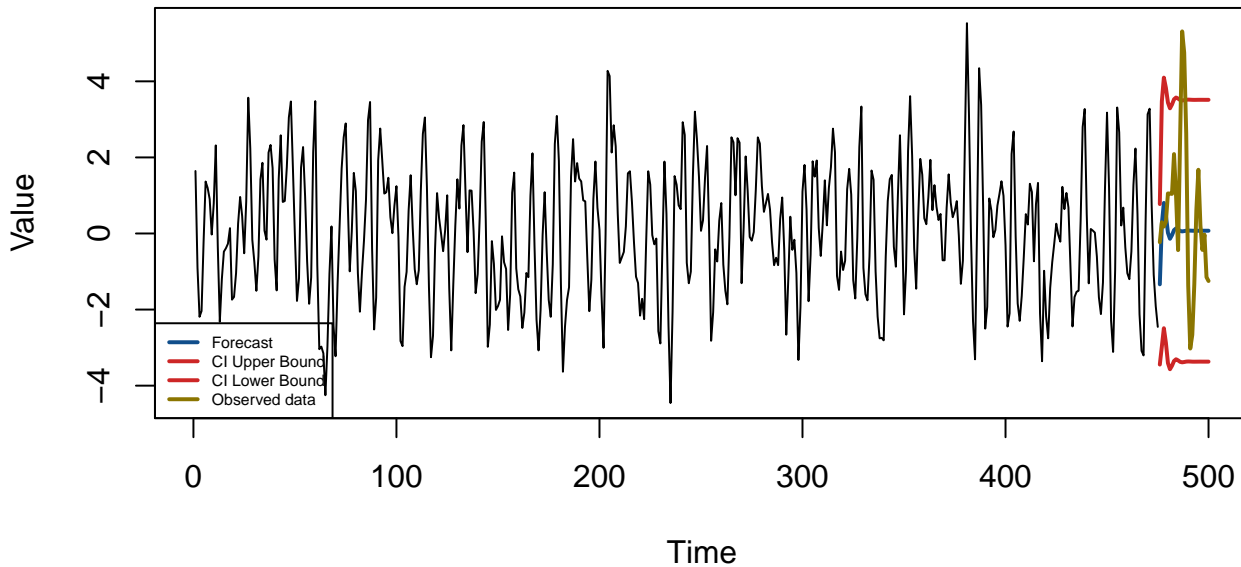
The confidence intervals (upper and lower bounds) are “converging” in the long-run because the ARMA(2,1) model is stationary; the variance is not growing over time but bounded.[Lüthkepol, p.71.]

```
# prediction
pred<-predict(ARMA_2_1_model,25)

# previous values
plot.ts(c(arma$V1[1:475],rep(NA,25)), main="ARMA(2,1) forecast", ylab="Value", pch=1)
lines(c(rep(NA,475),pred$pred), col="dodgerblue4", lwd=2)
# last 25 data points (observations)
last25<-c(rep(NA,475), arma$V1[476:500])

# calculate limits
upper_bound=c(rep(NA,475),2*pred$se+pred$pred)
lower_bound=c(rep(NA,475),-2*pred$se+pred$pred)
# +-2 s.e. CI
# upper bound
lines(upper_bound, col="firebrick3", lwd=2)
# lower bound
lines(lower_bound, col="firebrick3", lwd=2)
# add last 25 observed data points
lines(last25, col="gold4", lwd=2)
# add legend
legend("bottomleft",legend=c("Forecast", "CI Upper Bound", "CI Lower Bound", "Observed data"),
      col=c("dodgerblue4", "firebrick3", "firebrick3", "gold4"), lty = rep(1,4), lwd=rep(2,4),cex=0.5)
```

ARMA(2,1) forecast



```
# which values are above the upper limit
outlier_ind<-which(last25>upper_bound)
outlier_ind
```

```
## [1] 487 488
```

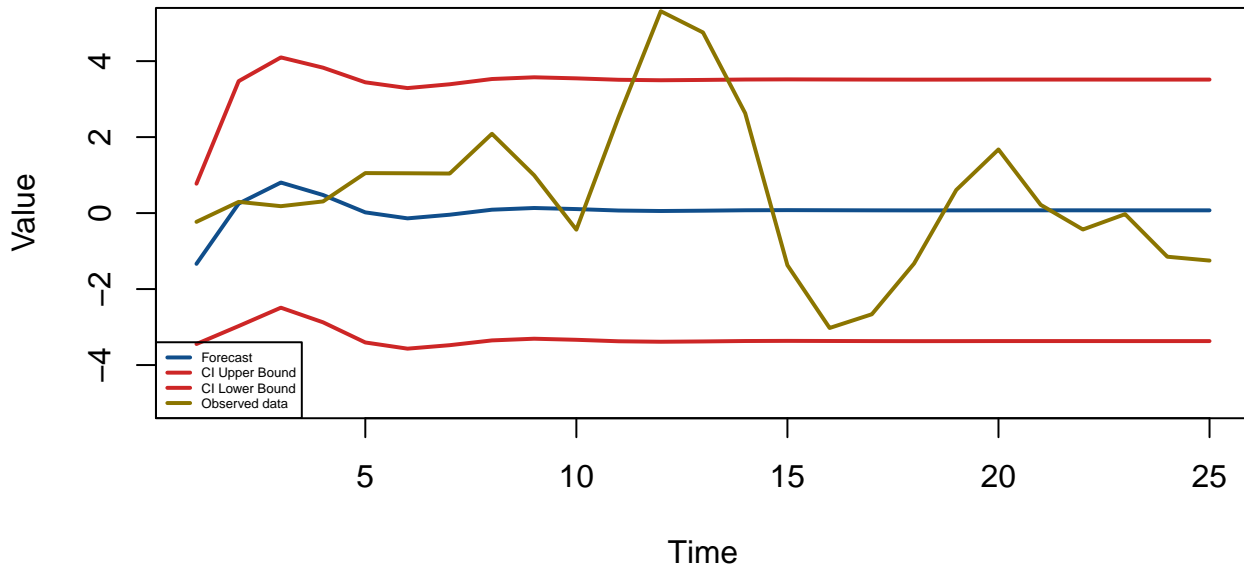
```
# outlying values
last25[outlier_ind]
```

```
## [1] 5.31 4.75
```

Below, you can see only the forecasted values with ± 2 standard deviation confidence intervals, and the realized values of the process.

```
# Zoom in to the forecast
plot.ts(as.vector(pred$pred), main="ARMA(2,1) forecast", ylab="Value", col="dodgerblue4", lwd=2, ylim=c(-5,5))
# +-2 s.e. CI
# upper bound
lines(upper_bound[476:500], col="firebrick3", lwd=2)
# lower bound
lines(lower_bound[476:500], col="firebrick3", lwd=2)
# add last 25 observed data points
lines(last25[476:500], col="gold4", lwd=2)
# add legend
legend("bottomleft", legend=c("Forecast", "CI Upper Bound", "CI Lower Bound", "Observed data"),
      col=c("dodgerblue4", "firebrick3", "firebrick3", "gold4"), lty = rep(1,4), lwd=rep(2,4), cex=0.4)
```

ARMA(2,1) forecast

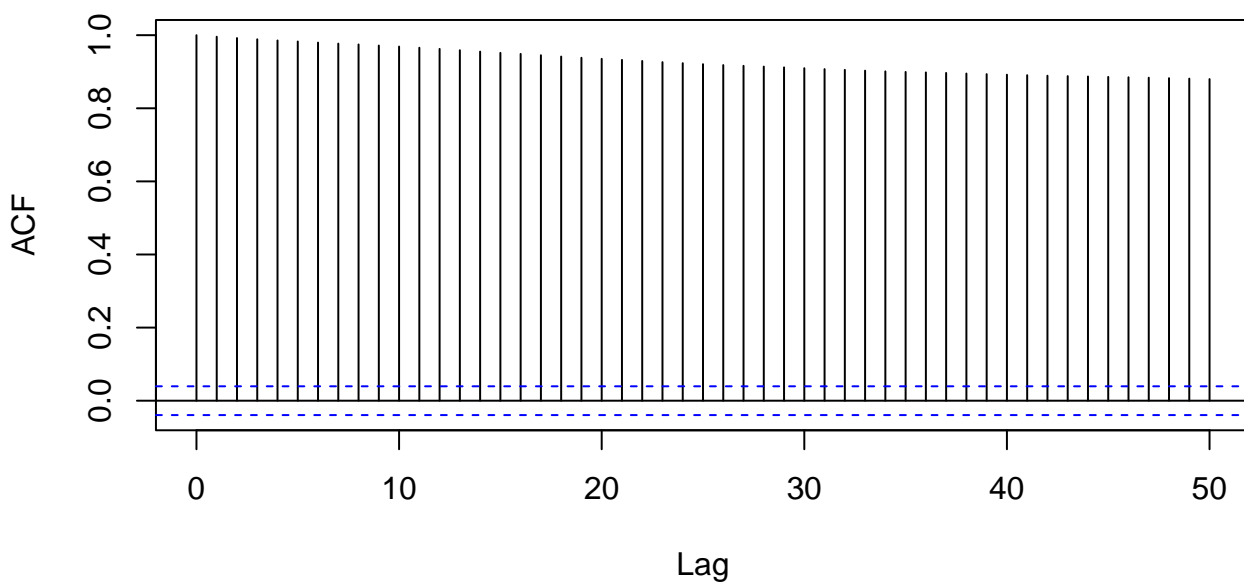


Exercise 5.,

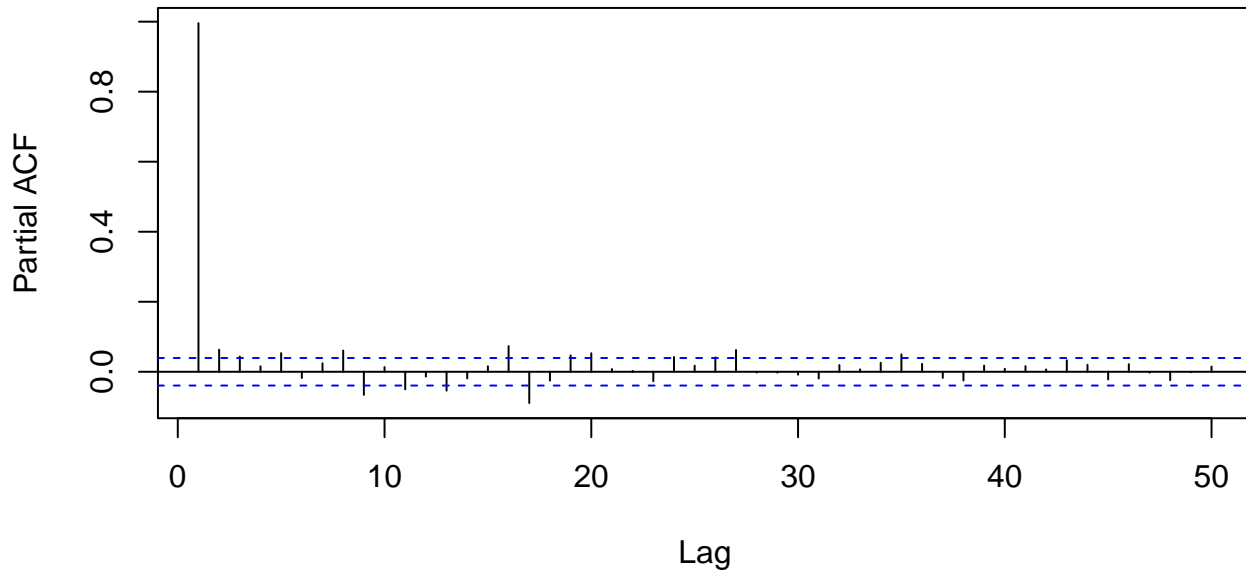
Part a.)

- a) Plot $\log(P_t)$ and its sample ACF. Test for the presence of a unit root using an appropriate case of the (A)DF test.

Autocorrelation function



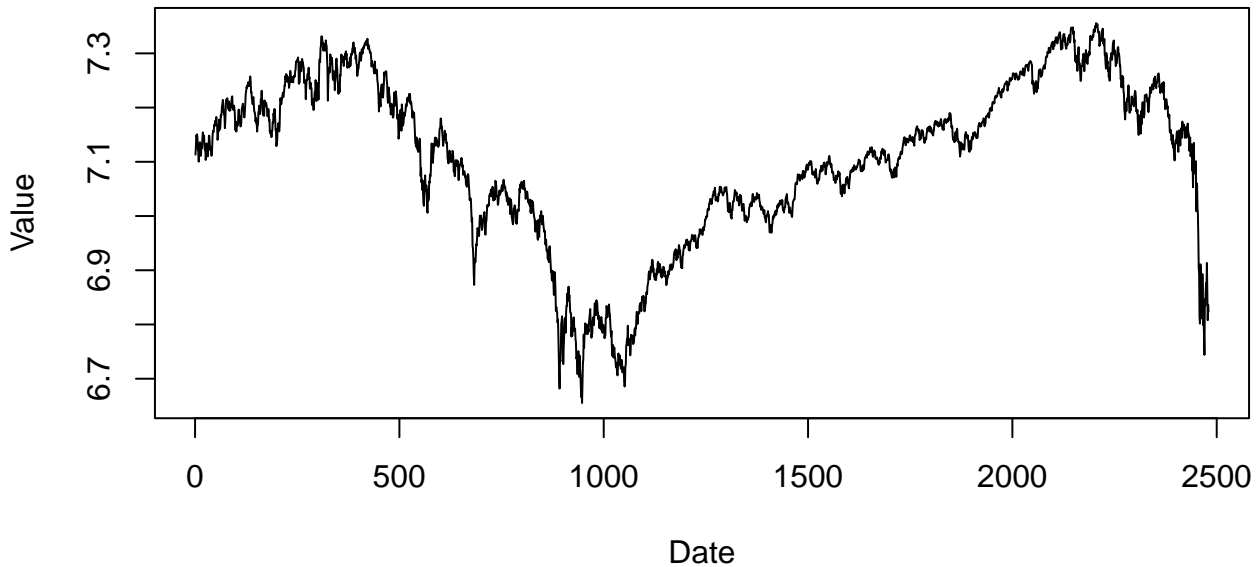
Partial Autocorrelation function



Answer:

The ACF and PACF plots suggest a long-memory (AR) process can be suitable for the data. The autocorrelation function converges very slowly, all the first 50 values are significant. Nevertheless, the values of the PACF are close to the confidence interval, but some of them are significant, which can suggest a stochastic process with long-memory, which is typical for financial data, or it can be a unit-root process. (So, I can suggest an AR(1), thus first PACF is significant, and others are close to the confidence interval, but may be the first lag is insignificant and the observed process is a unit-root process. I test the existence of a unit root later.)

Log Price of the SP500 index



The plot suggests no deterministic, linear trend in the data. (Maybe a “cycle” can be observed or a cubic tendency.) As discussed in class, the two mainly applied versions of the Dickey-Fuller test are (2) and (4) in our class notation. Case (2) means we compare a random walk without drift (H_0), with a stationary process with drift (H_1). While in case of (4), H_0 is a random walk with drift, while the alternative is a stationary process with drift and deterministic trend. I formerly generated the plot of log prices to ensure the existence of a deterministic trend. I think there is no deterministic trend in the data, so I tested the second version of the Dickey-Fuller test. (Remark: R calculates the Augmented version of the Dickey-Fuller test, where there are three types of null hypothesis, and the second one is what I will look. This is an analogous version we did in class, but “augmented” (= differences are included in the tested models).)

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]  0 -0.492  0.502
## [2,]  1 -0.542  0.485
## [3,]  2 -0.615  0.459
## [4,]  3 -0.610  0.461
## [5,]  4 -0.629  0.454
## [6,]  5 -0.614  0.459
## [7,]  6 -0.595  0.466
## [8,]  7 -0.626  0.455
## [9,]  8 -0.567  0.476
## [10,] 9 -0.615  0.459
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]  0 -1.68  0.454
## [2,]  1 -1.52  0.518
## [3,]  2 -1.35  0.579
## [4,]  3 -1.37  0.570
## [5,]  4 -1.32  0.588
```

```
## [6,] 5 -1.32 0.588
## [7,] 6 -1.26 0.611
## [8,] 7 -1.08 0.674
## [9,] 8 -1.20 0.630
## [10,] 9 -1.17 0.643
## Type 3: with drift and trend
##      lag      ADF p.value
## [1,] 0 -1.608 0.744
## [2,] 1 -1.439 0.815
## [3,] 2 -1.264 0.890
## [4,] 3 -1.292 0.878
## [5,] 4 -1.240 0.900
## [6,] 5 -1.236 0.901
## [7,] 6 -1.160 0.913
## [8,] 7 -0.971 0.943
## [9,] 8 -1.095 0.923
## [10,] 9 -1.065 0.928
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

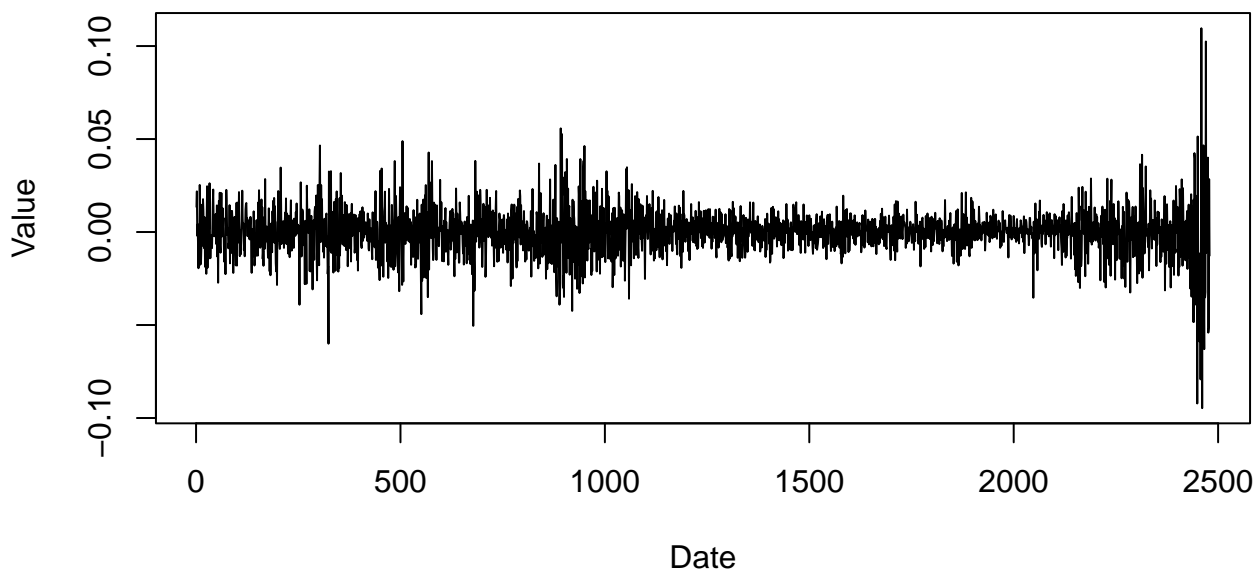
The test automatically presents all the three test results, the second one is important. (The second one is the same as we considered as second case in class.)

Conclusion: all the p-values are above any general significance levels (1%,5%, and 10%), so I cannot reject the null hypothesis that the process is a random walk without drift (second table). Considering the time series plot, there is no deterministic trend in the data. (The test is presented with 10 lags to avoid large tables, but for 30 lags the conclusion is the same, the process contains unit root(s).)

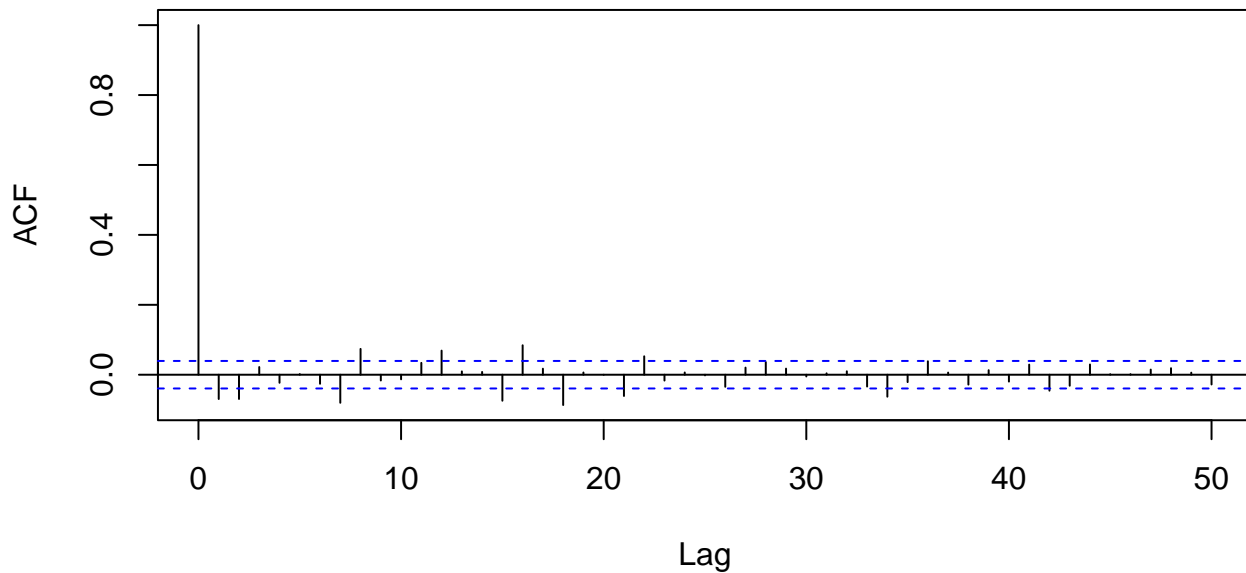
Part b.)

- b) Compute the corresponding return series. Plot `rt` and its sample ACF. Test for the presence of unit roots in `rt` using an appropriate case of the (A)DF test.

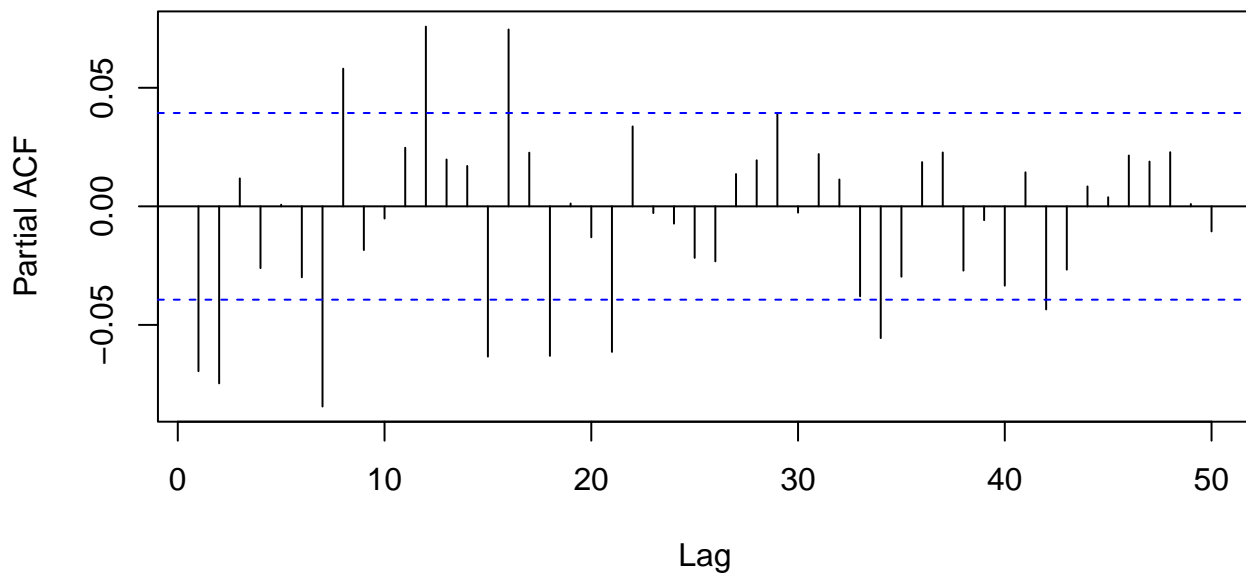
Log Returns of the SP500 index



Autocorrelation function



Autocorrelation function



Answer:

The Autocorrelation Function values are close to the confidence interval bounds, so this process does not have a long-memory, seemingly not an AR-type, but rather an MA one.

I do not see any deterministic trend in the data by plotting the log returns, and maybe no drift. Seemingly, the process has zero mean, but it is hard to decide whether the drift is necessary or not, so for the sake of safety in parallel with practice, I will use the second version of the ADF test (H_0 : RW without drift and H_1 : stationary

process with drift). This will be tested in the second table as before.

Augmented Dickey-Fuller Test

alternative: stationary

##

Type 1: no drift no trend

	lag	ADF	p.value
## [1,]	0	-53.4	0.01
## [2,]	1	-39.3	0.01
## [3,]	2	-30.4	0.01
## [4,]	3	-26.6	0.01
## [5,]	4	-23.4	0.01
## [6,]	5	-21.7	0.01
## [7,]	6	-21.7	0.01
## [8,]	7	-18.7	0.01
## [9,]	8	-17.8	0.01
## [10,]	9	-16.9	0.01

Type 2: with drift no trend

	lag	ADF	p.value
## [1,]	0	-53.4	0.01
## [2,]	1	-39.2	0.01
## [3,]	2	-30.4	0.01
## [4,]	3	-26.6	0.01
## [5,]	4	-23.4	0.01
## [6,]	5	-21.7	0.01
## [7,]	6	-21.7	0.01
## [8,]	7	-18.7	0.01
## [9,]	8	-17.8	0.01
## [10,]	9	-16.9	0.01

Type 3: with drift and trend

	lag	ADF	p.value
## [1,]	0	-53.4	0.01
## [2,]	1	-39.3	0.01
## [3,]	2	-30.4	0.01
## [4,]	3	-26.6	0.01
## [5,]	4	-23.4	0.01
## [6,]	5	-21.8	0.01
## [7,]	6	-21.7	0.01
## [8,]	7	-18.7	0.01
## [9,]	8	-17.9	0.01
## [10,]	9	-16.9	0.01

Note: in fact, p.value = 0.01 means p.value <= 0.01

The second table is relevant now. There is no unit root in the process at a 5% significance level. Remark: the program reports a 0.01 p-value if the p-value is equal to or smaller than 0.01. So, I can reject the H_0 at any general significance level. So, log returns are stationary without drift. No unit root is in the (differentiated) process.