

ENV 797 - Time Series Analysis for Energy and Environment Applications | Spring 2026

Assignment 6 - Due date 02/27/26

Trudy

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github.

Once you have the file open on your local machine the first thing you will do is rename the file such that it includes your first and last name (e.g., “LuanaLima_TSA_A06_Sp26.Rmd”). Then change “Student Name” on line 4 with your name.

Then you will start working through the assignment by **creating code and output** that answer each question. Be sure to use this assignment document. Your report should contain the answer to each question and any plots/tables you obtained (when applicable).

When you have completed the assignment, **Knit** the text and code into a single PDF file. Submit this pdf using Sakai.

R packages needed for this assignment: “ggplot2”, “forecast”, “tseries” and “sarima”. Install these packages, if you haven’t done yet. Do not forget to load them before running your script, since they are NOT default packages.

```
#Load/install required package here  
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':  
##   method           from  
##   as.zoo.data.frame zoo
```

```
library(tseries)  
library(ggplot2)  
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(cowplot)
```

```
install.packages("sarima")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.4'
```

```
## (as 'lib' is unspecified)
```

```
library(sarima)
```

```
## Loading required package: stats4
```

```
##
```

```
## Attaching package: 'sarima'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##      spectrum
```

This assignment has general questions about ARIMA Models.

Q1

Describe the important characteristics of the sample autocorrelation function (ACF) plot and the partial sample autocorrelation function (PACF) plot for the following models:

- AR(2)

Answer: The ACF declines gradually rather than showing a sharp cutoff. The PACF displays significant spikes at lags 1 and 2 and becomes insignificant for higher lags.

- MA(1)

Answer: The ACF shows a significant spike at lag 1 and then drops to zero afterward. The PACF decreases gradually instead of cutting off at a specific lag.

Q2

Recall that the non-seasonal ARIMA is described by three parameters $ARIMA(p, d, q)$ where p is the order of the autoregressive component, d is the number of times the series need to be differenced to obtain stationarity and q is the order of the moving average component. If we don't need to difference the series, we don't need to specify the "I" part and we can use the short version, i.e., the $ARMA(p, q)$.

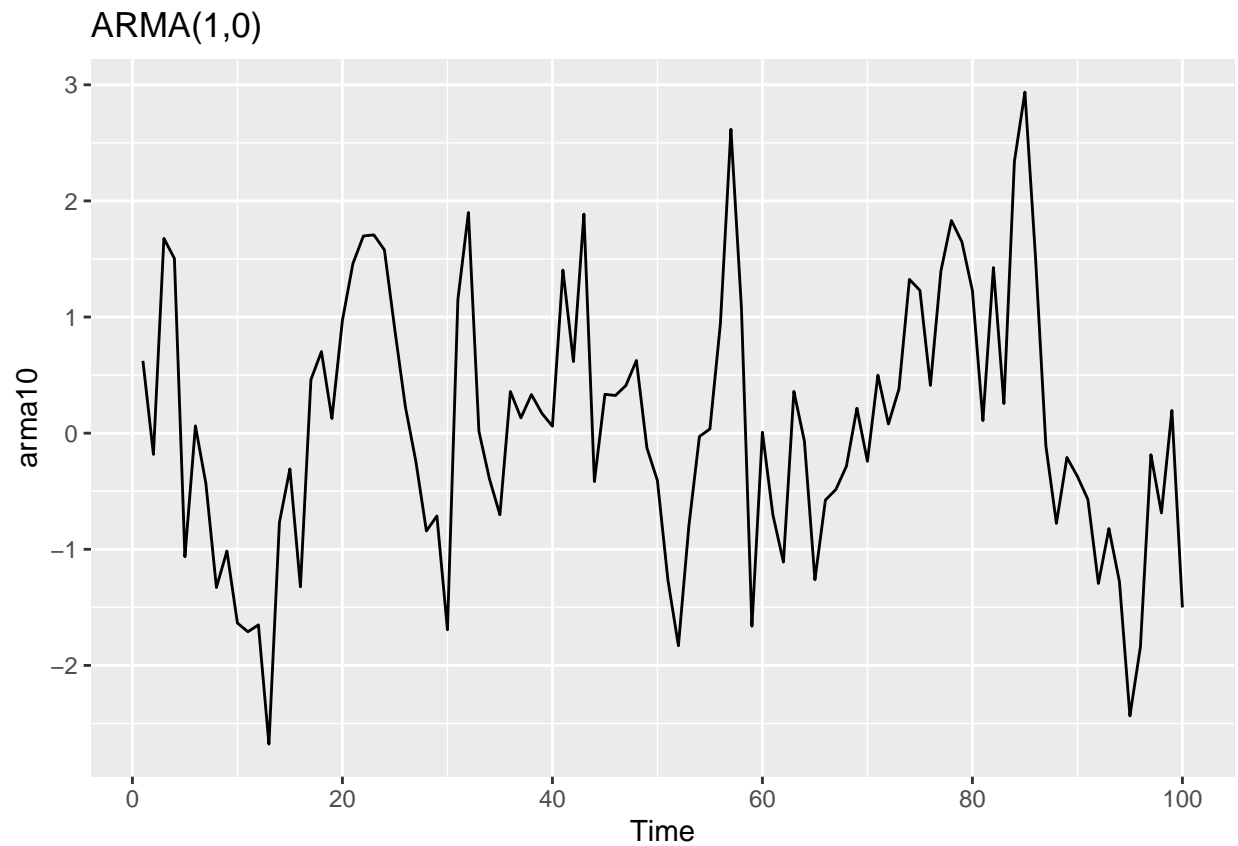
- (a) Consider three models: $ARMA(1,0)$, $ARMA(0,1)$ and $ARMA(1,1)$ with parameters $\phi = 0.6$ and $\theta = 0.9$. The ϕ refers to the AR coefficient and the θ refers to the MA coefficient. Use the `arima.sim()` function in R to generate $n = 100$ observations from each of these three models. Then, using `autoplot()` plot the generated series in three separate graphs.

```

set.seed(123)
n <- 100
phi <- 0.6
theta <- 0.9

# ARMA(1,0)
arma10 <- arima.sim(model = list(ar = phi), n = n)
autoplot(arma10) + ggtitle("ARMA(1,0)")

```

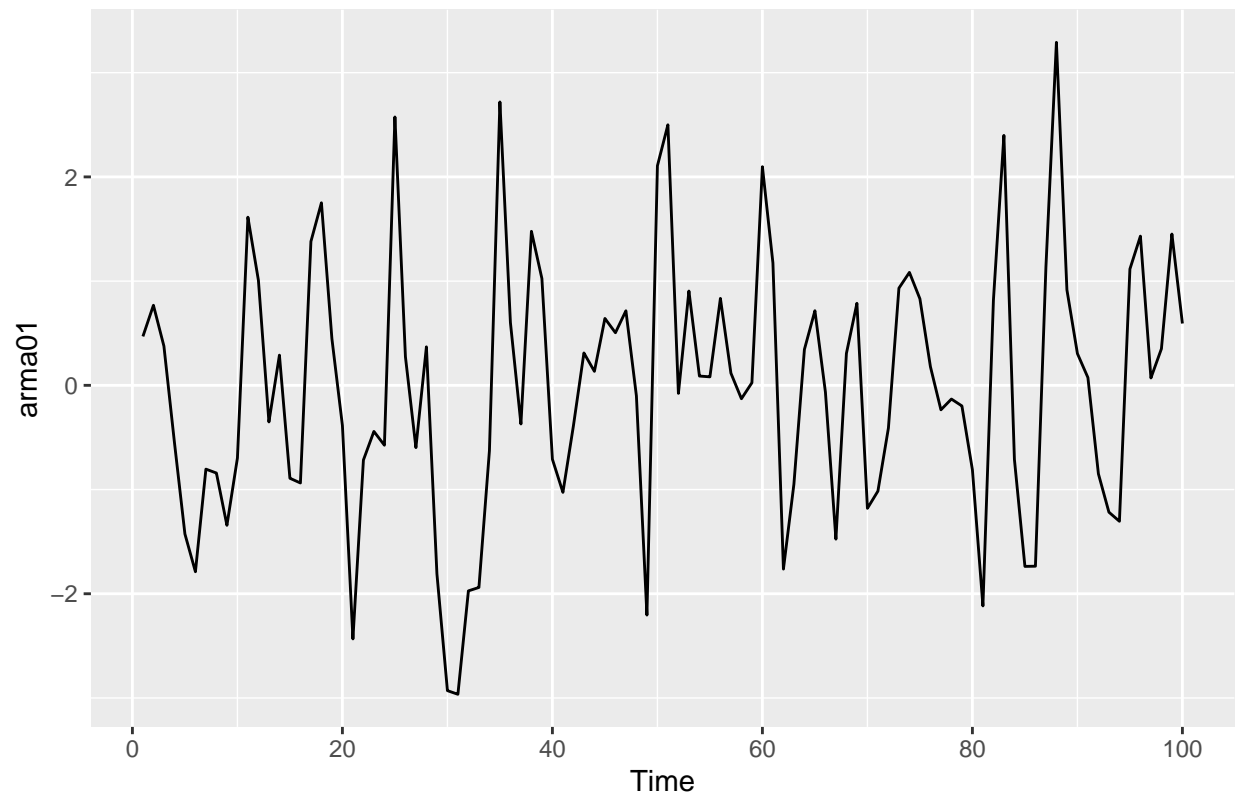


```

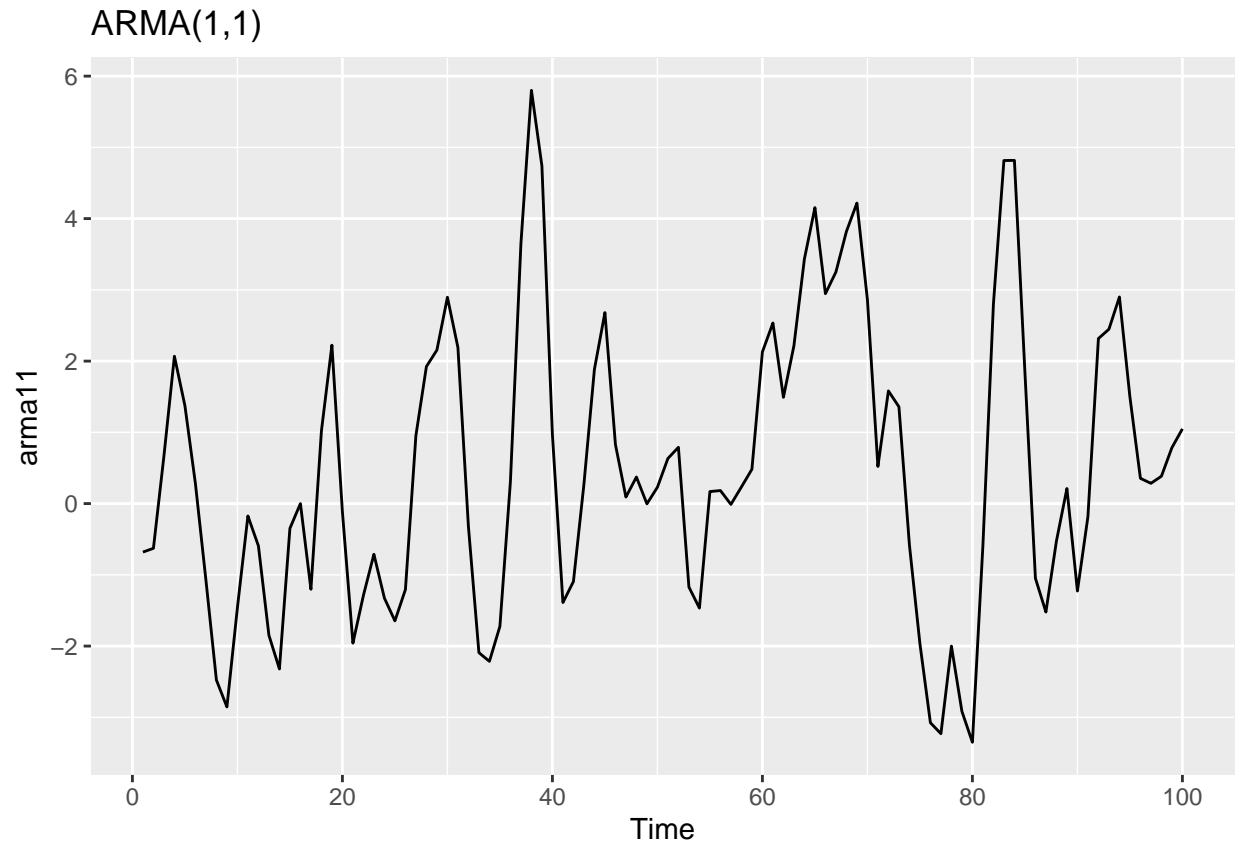
# ARMA(0,1)
arma01 <- arima.sim(model = list(ma = theta), n = n)
autoplot(arma01) + ggtitle("ARMA(0,1)")

```

ARMA(0,1)



```
# ARMA(1,1)
arma11 <- arima.sim(model = list(ar = phi, ma = theta), n = n)
autoplot(arma11) + ggtitle("ARMA(1,1)")
```



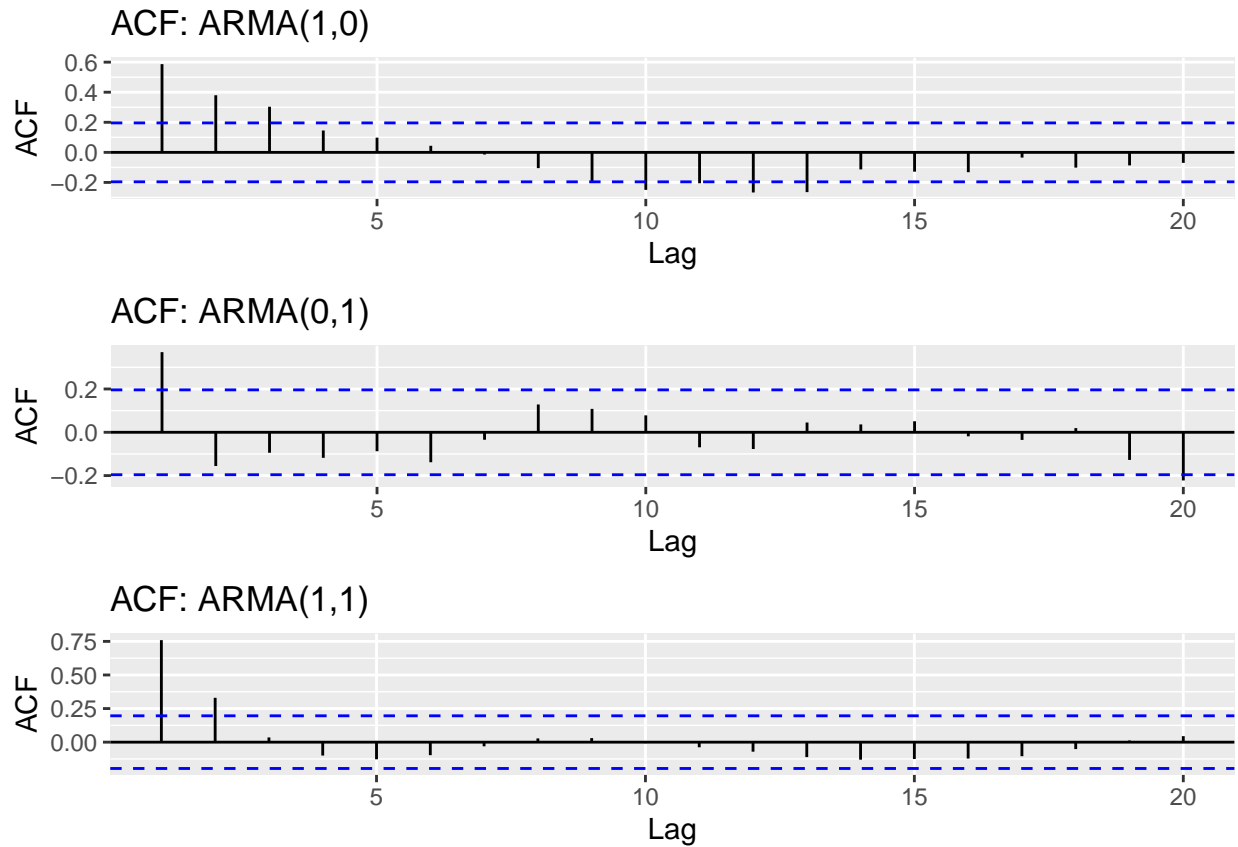
(b) Plot the sample ACF for each of these models in one window to facilitate comparison (Hint: use `cowplot::plot_grid()`).

```
acf1 <- autoplot(Acf(arma10, plot = FALSE)) +
  ggtitle("ACF: ARMA(1,0)")

acf2 <- autoplot(Acf(arma01, plot = FALSE)) +
  ggtitle("ACF: ARMA(0,1)")

acf3 <- autoplot(Acf(arma11, plot = FALSE)) +
  ggtitle("ACF: ARMA(1,1)")

cowplot::plot_grid(acf1, acf2, acf3, ncol = 1)
```



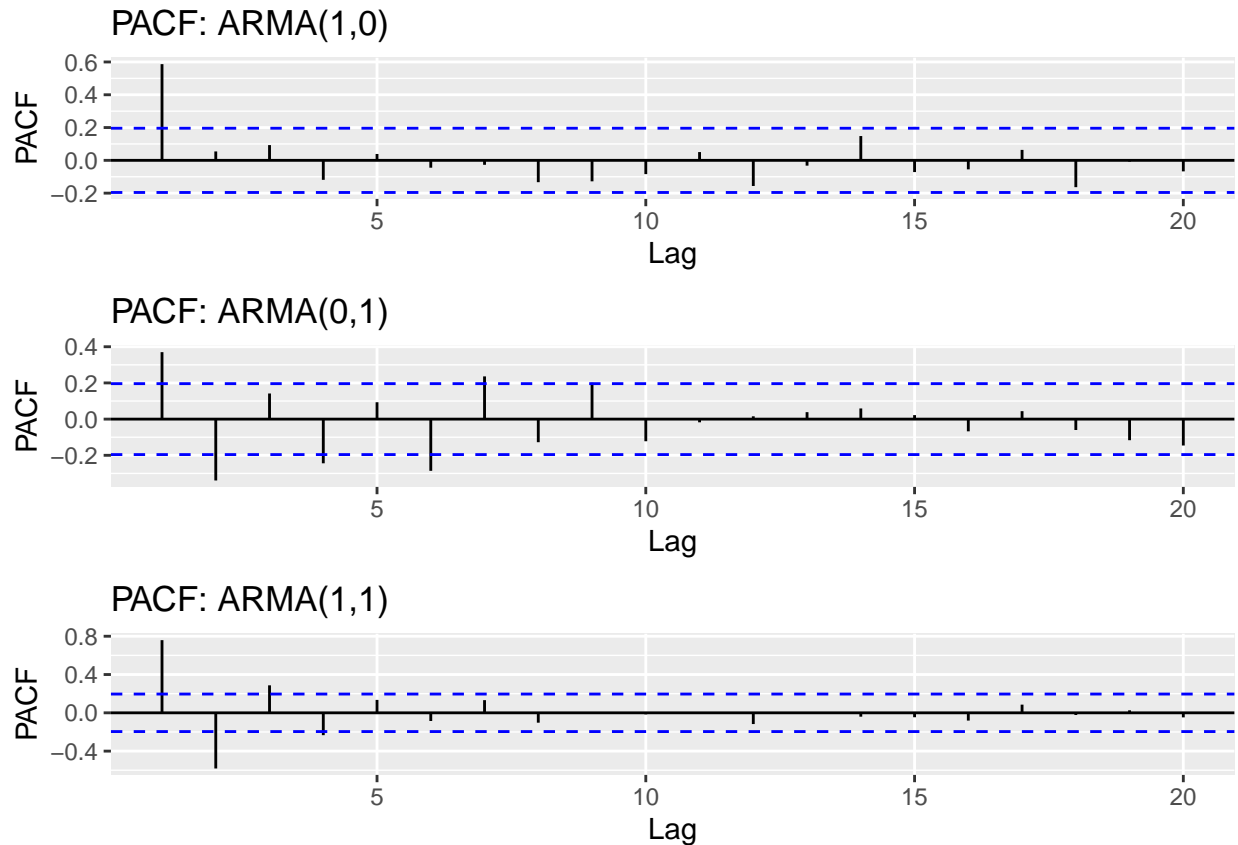
(c) Plot the sample PACF for each of these models in one window to facilitate comparison.

```
pacf1 <- autoplot(Pacf(arma10, plot = FALSE)) +
  ggtitle("PACF: ARMA(1,0)")

pacf2 <- autoplot(Pacf(arma01, plot = FALSE)) +
  ggtitle("PACF: ARMA(0,1)")

pacf3 <- autoplot(Pacf(arma11, plot = FALSE)) +
  ggtitle("PACF: ARMA(1,1)")

cowplot::plot_grid(pacf1, pacf2, pacf3, ncol = 1)
```



- (d) Look at the ACFs and PACFs. Imagine you had these plots for a data set and you were asked to identify the model, i.e., is it AR, MA or ARMA and the order of each component. Would you be able to identify them correctly? Explain your answer.

Answer: In theory, AR, MA, and ARMA models can be identified from the ACF and PACF patterns. AR models show PACF cutoff, MA models show ACF cutoff, and ARMA models show gradual decay in both. However, with finite samples, the plots may not exhibit clear cutoff behavior, so identification may not always be exact.

- (e) Compare the PACF values R computed with the values you provided for the lag 1 correlation coefficient, i.e., does $\phi = 0.6$ match what you see on PACF for ARMA(1,0), and ARMA(1,1)? Should they match?

Answer: For ARMA(1,0), the theoretical PACF at lag 1 equals the AR coefficient ϕ . Since $\phi = 0.6$, we expect the lag 1 PACF to be close to 0.6. However, in a finite sample, the estimated PACF may not be exactly 0.6 due to sampling variability. For ARMA(1,1), the lag 1 PACF is generally not equal to ϕ because the MA component also affects the dependence structure. Therefore, we should not expect the lag 1 PACF to match 0.6 exactly for ARMA(1,1).

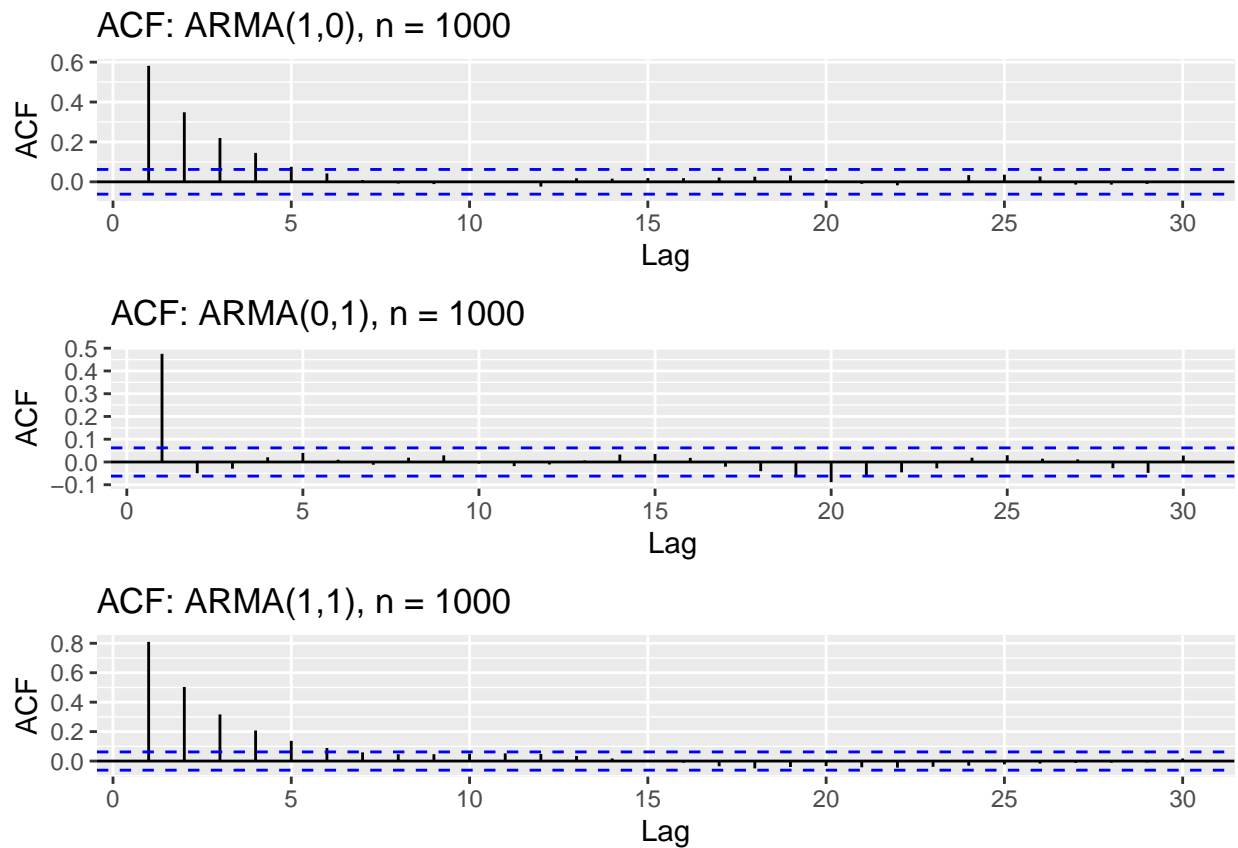
- (f) Increase number of observations to $n = 1000$ and repeat parts (b)-(e).

```
n1 <- 1000
phi <- 0.6
theta <- 0.9
```

```
arma10_1 <- arima.sim(model = list(ar = phi), n = n1)
arma01_1 <- arima.sim(model = list(ma = theta), n = n1)
arma11_1 <- arima.sim(model = list(ar = phi, ma = theta), n = n1)

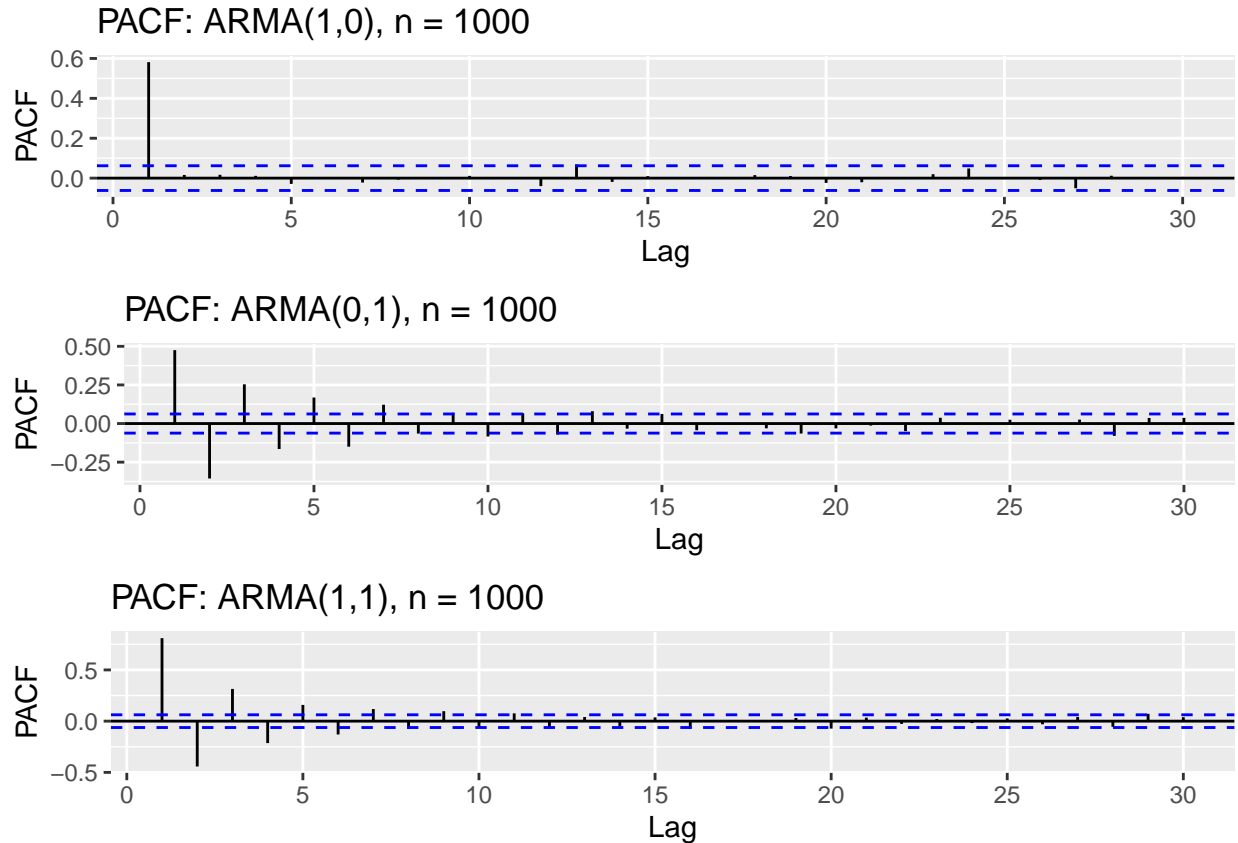
# ACF
acf1_1 <- autoplot(Acf(arma10_1, plot = FALSE)) + ggtitle("ACF: ARMA(1,0), n = 1000")
acf2_1 <- autoplot(Acf(arma01_1, plot = FALSE)) + ggtitle("ACF: ARMA(0,1), n = 1000")
acf3_1 <- autoplot(Acf(arma11_1, plot = FALSE)) + ggtitle("ACF: ARMA(1,1), n = 1000")

cowplot::plot_grid(acf1_1, acf2_1, acf3_1, ncol = 1)
```



```
# PACF
pacf1_1 <- autoplot(Pacf(arma10_1, plot = FALSE)) + ggtitle("PACF: ARMA(1,0), n = 1000")
pacf2_1 <- autoplot(Pacf(arma01_1, plot = FALSE)) + ggtitle("PACF: ARMA(0,1), n = 1000")
pacf3_1 <- autoplot(Pacf(arma11_1, plot = FALSE)) + ggtitle("PACF: ARMA(1,1), n = 1000")

cowplot::plot_grid(pacf1_1, pacf2_1, pacf3_1, ncol = 1)
```

> Answer: When the sample size increases to 1000, the ACF and PACF plots become more stable and closer to the theoretical patterns. The confidence bands become narrower, and random noise is reduced. For AR(1), the PACF clearly cuts off after lag 1 and is close to 0.6. For MA(1), the ACF clearly cuts off after lag 1. For ARMA(1,1), both ACF and PACF show gradual decay. Overall, larger sample size makes the model identification easier and more accurate.

Q3

Consider the ARIMA model $y_t = 0.7 * y_{t-1} - 0.25 * y_{t-12} + a_t - 0.1 * a_{t-1}$

- Identify the model using the notation $ARIMA(p, d, q)(P, D, Q)_s$, i.e., identify the integers p, d, q, P, D, Q, s (if possible) from the equation. > Answer: From the equation, there is an AR term at lag 1, an AR term at lag 12 (seasonal), and an MA term at lag 1. There is no differencing. Therefore the model is: $ARIMA(1,0,1)(1,0,0)[12]$. So $p = 1, d = 0, q = 1, P = 1, D = 0, Q = 0$, and $s = 12$.
- Also from the equation what are the values of the parameters, i.e., model coefficients. > Answer: The coefficients from the equation are: AR(1) coefficient (lag 1): 0.7, Seasonal AR(1) coefficient (lag 12): -0.25, MA(1) coefficient (lag 1): -0.1, Innovation term: a_t has coefficient 1.

Q4

Simulate a seasonal $ARIMA(0, 1) \times (1, 0)_{12}$ model with $\phi = 0.8$ and $\theta = 0.5$ using the `sim_sarima()` function from package `sarima`. The 12 after the bracket tells you that $s = 12$, i.e., the seasonal lag is 12, suggesting monthly data whose behavior is repeated every 12 months. You can generate as many observations as you

like. Note the Integrated part was omitted. It means the series do not need differencing, therefore $d = D = 0$. Plot the generated series using `autoplot()`. Does it look seasonal?

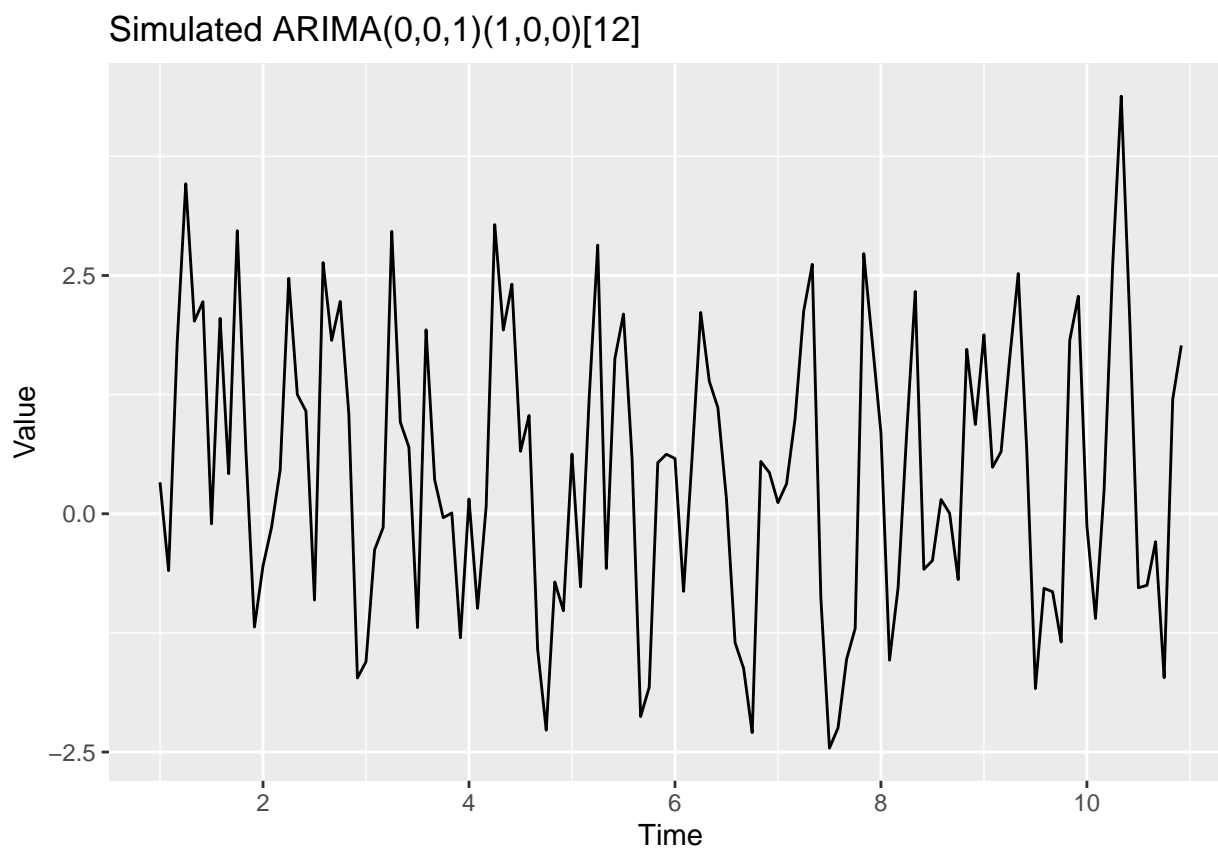
```
set.seed(23)

n <- 120
phi <- 0.8
theta <- 0.5

x <- sim_sarima(
  n = n,
  model = list(ma = theta, sar = phi, nseasons = 12)
)

x_ts <- ts(x, frequency = 12)

autoplot(x_ts) +
  ggtitle("Simulated ARIMA(0,0,1)(1,0,0)[12]") +
  xlab("Time") + ylab("Value")
```

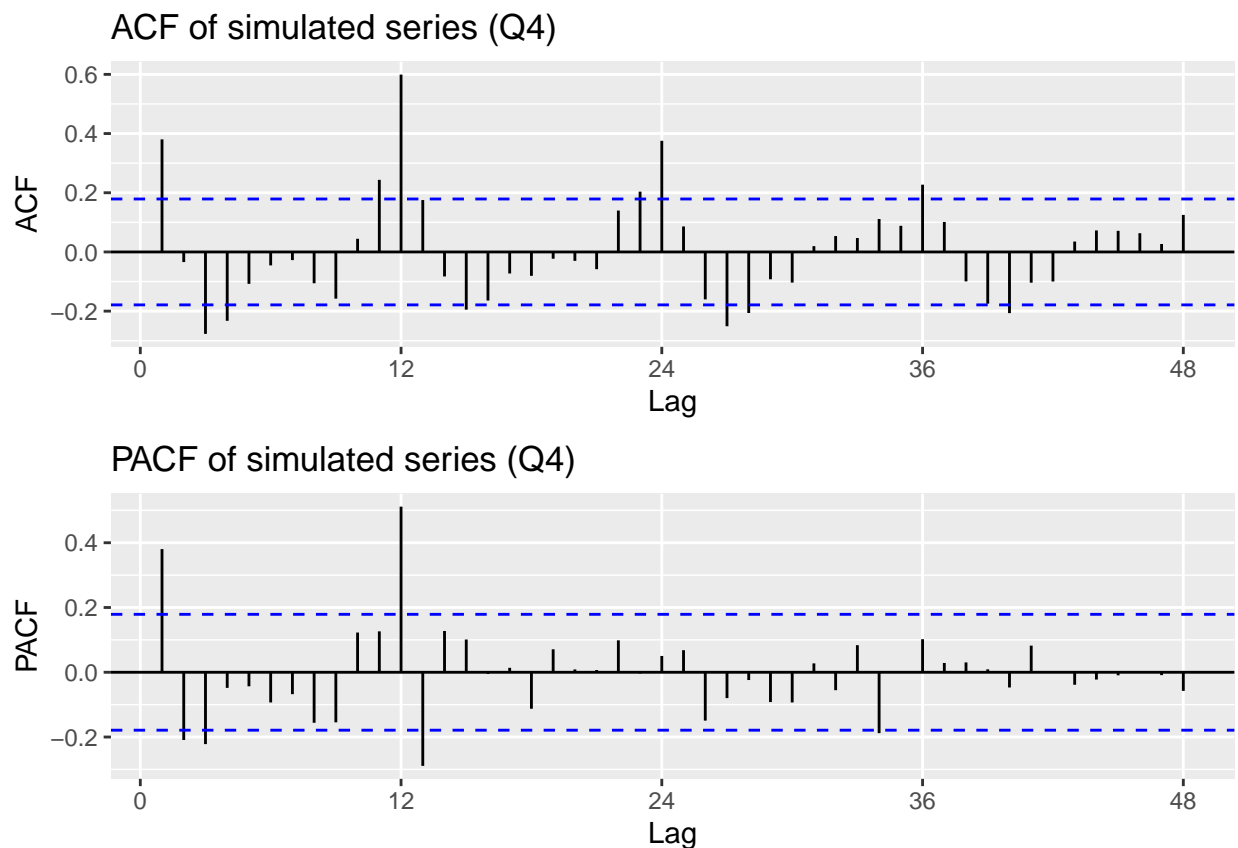


> Answer: Yes. The series shows repeating patterns approximately every 12 periods, indicating clear seasonal behavior.

Q5

Plot ACF and PACF of the simulated series in Q4. Comment if the plots are well representing the model you simulated, i.e., would you be able to identify the order of both non-seasonal and seasonal components from the plots? Explain.

```
acf_plot <- autoplot(Acf(x_ts, plot = FALSE, lag.max = 48)) +  
  ggtitle("ACF of simulated series (Q4)")  
  
pacf_plot <- autoplot(Pacf(x_ts, plot = FALSE, lag.max = 48)) +  
  ggtitle("PACF of simulated series (Q4)")  
  
cowplot::plot_grid(acf_plot, pacf_plot, ncol = 1)
```



> Answer: The ACF shows clear spikes at lags 12, 24, and 36, indicating a seasonal period of 12. The PACF also has a significant spike at lag 12, consistent with a seasonal AR(1) component. There is also evidence of a low-order non-seasonal component. However, the cutoffs are not perfectly sharp, so identifying the exact orders from one realization is not completely certain.