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

Assignment 6 - Due date 02/27/26

Jeffrey Chu

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

```
## Warning: package 'forecast' was built under R version 4.4.3
```

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

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.4.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```r
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.4.3
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```r
library(tidyverse)  #load this package so yon clean the data frame using pipes
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2

## Warning: package 'tidyr' was built under R version 4.4.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.4     v stringr 1.5.1
## v forcats 1.0.0     v tibble  3.2.1
## v purrr   1.0.2     v tidyr   1.3.1
## v readr   2.1.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(readxl)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

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: An AR(2) model exhibits a cut-off in the PACF (p) at lag 2, while the ACF decays gradually (possibly in an exponential or damped oscillatory pattern).

- MA(1)

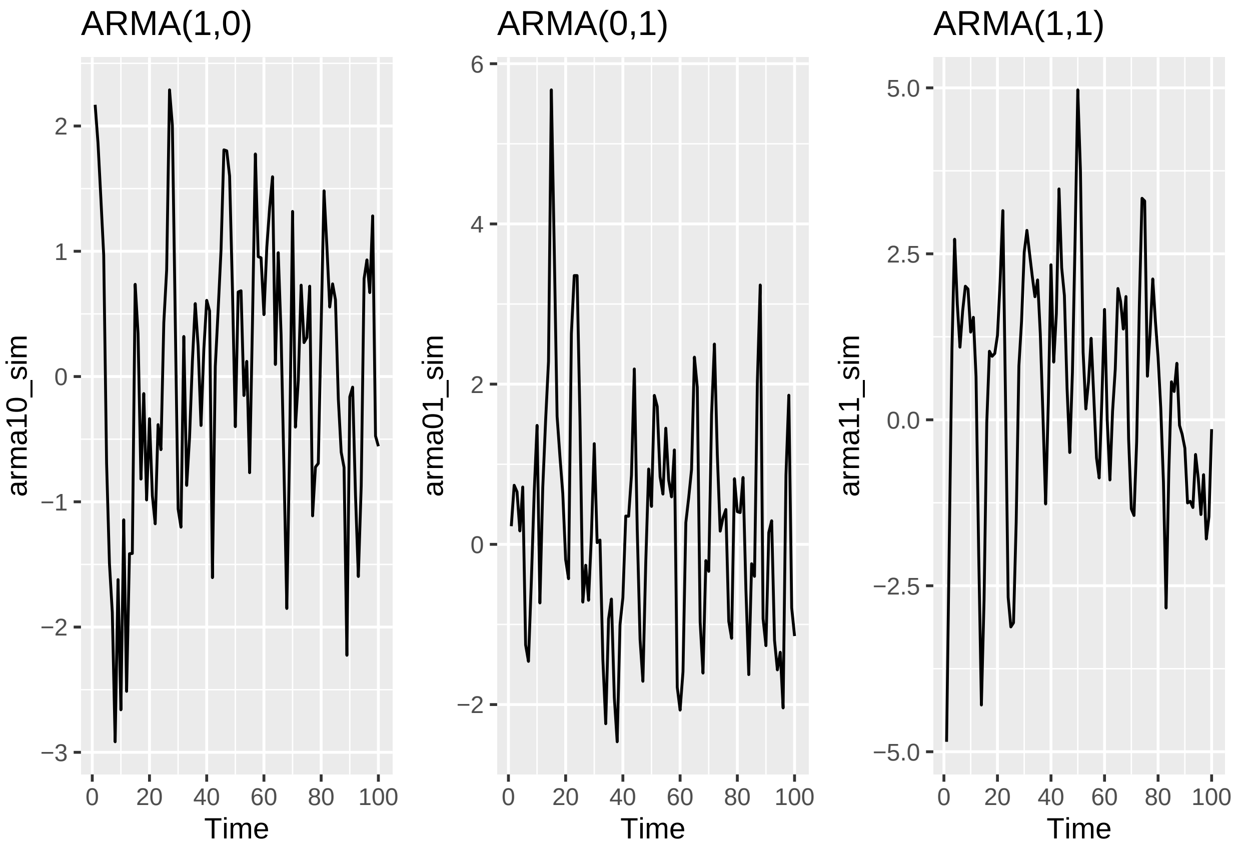
  Answer: An MA(2) model will show a cut-of in the ACF plot at lag(q) = 1, while the PACF decays gradually.

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).

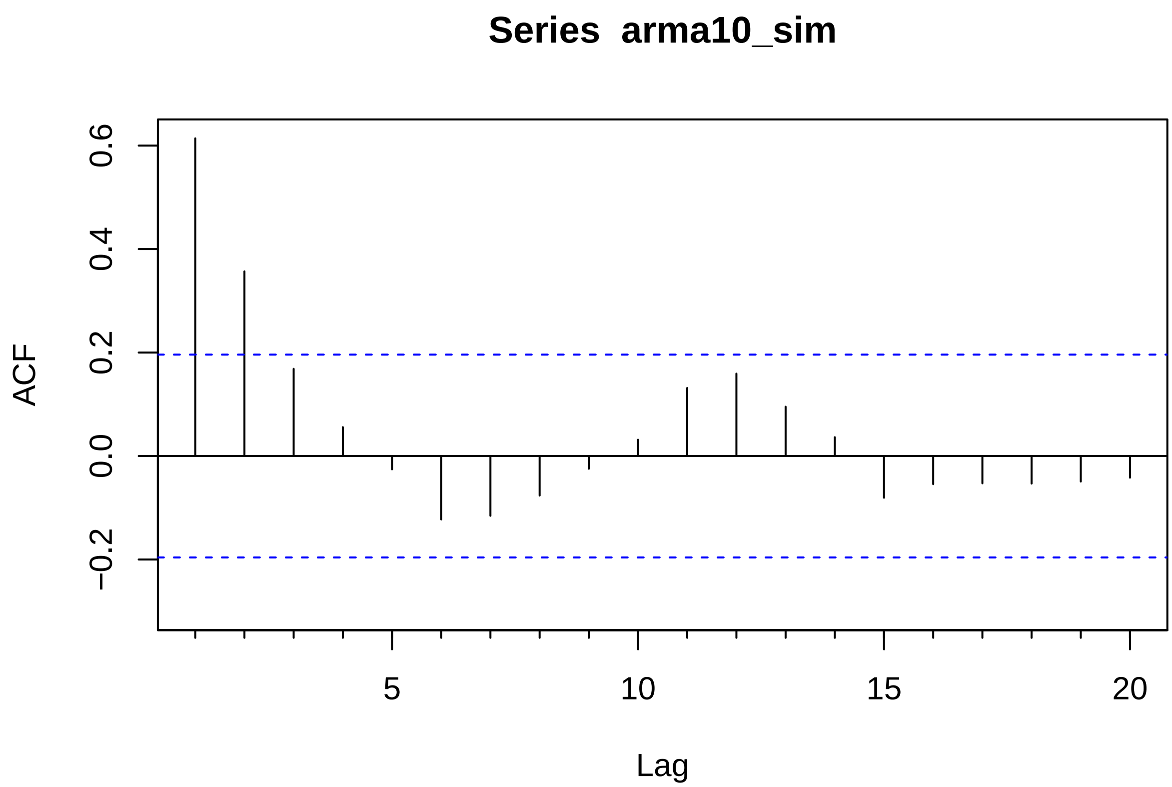
(a) Consider three models: ARMA(1,0), ARMA(0,1) and ARMA(1,1) with parameters $\phi = 0.6$ and $\theta = 0.9$. The ϕ refers to the AR coefficient and the θ refers to the MA coefficient. Use the `arima.sim()` function in R to generate $n = 100$ observations from each of these three models. Then, using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

```r
n<-100
phi<-0.6
theta<-0.9
arma10_sim<- arima.sim(model=list(ar=phi),n=n)
arma01_sim<- arima.sim(model=list(ma=theta),n=n)
arma11_sim <- arima.sim(model = list(ar = phi, ma = theta), n = n)

plot_grid(
  autoplot(arma10_sim) + ggtitle("ARMA(1,0)"),
  autoplot(arma01_sim) + ggtitle("ARMA(0,1)"),
  autoplot(arma11_sim) + ggtitle("ARMA(1,1)"),
  nrow = 1
)
```

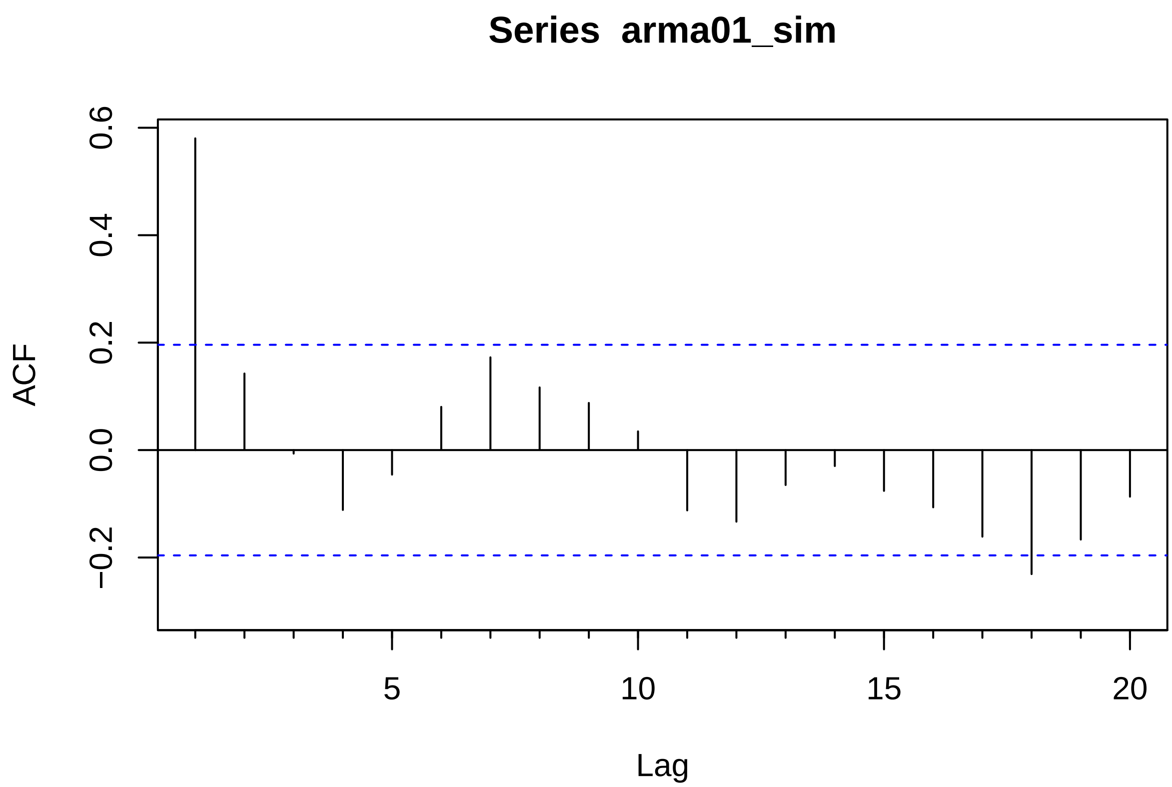


(b) Plot the sample ACF for each of these models in one window to facilitate comparison (Hint: use `cowplot::plot_grid()`).

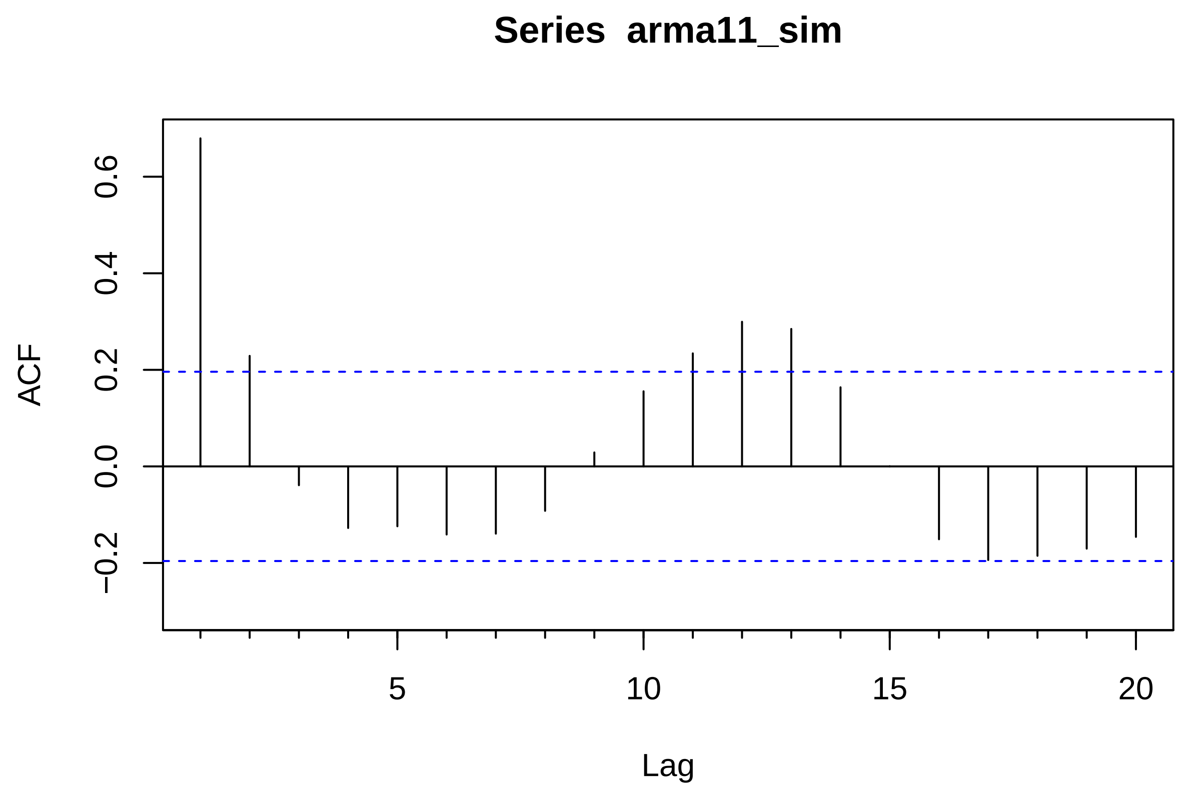
```
Acf_arma10<-Acf(arma10_sim)
```



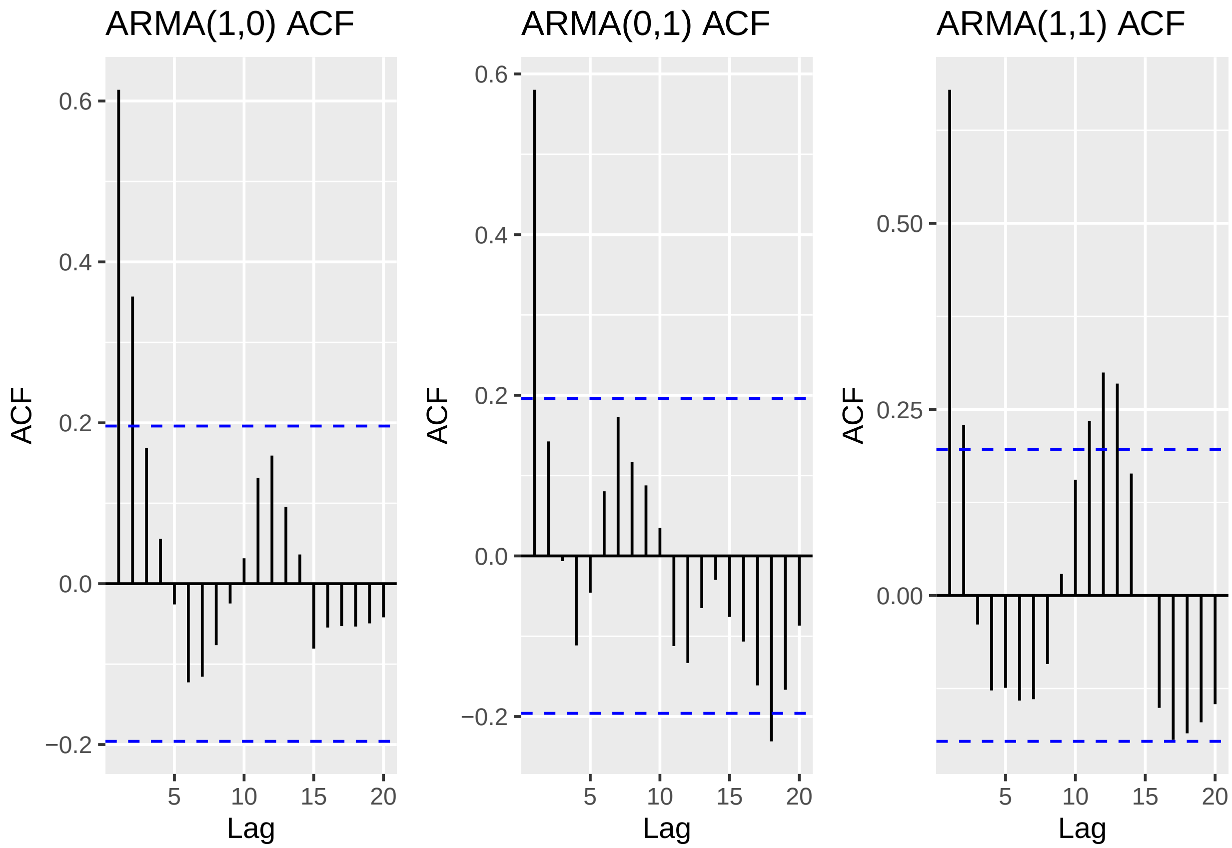

```
Acf_arma01<-Acf(arma01_sim)
```




```
Acf_arma11<-Acf(arma11_sim)
```

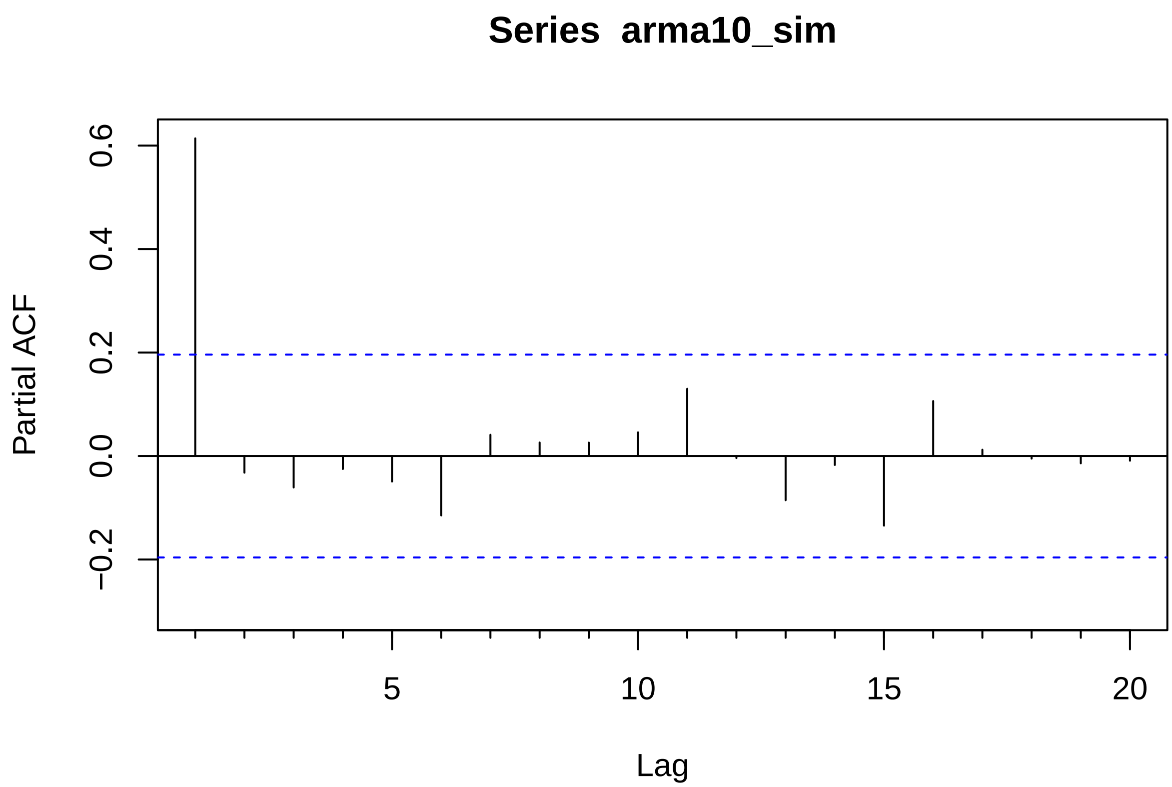



```r
plot_grid(
  autoplot(Acf_arma10) + ggtitle("ARMA(1,0) ACF"),
  autoplot(Acf_arma01) + ggtitle("ARMA(0,1) ACF"),
  autoplot(Acf_arma11) + ggtitle("ARMA(1,1) ACF"),
  nrow = 1
)
```

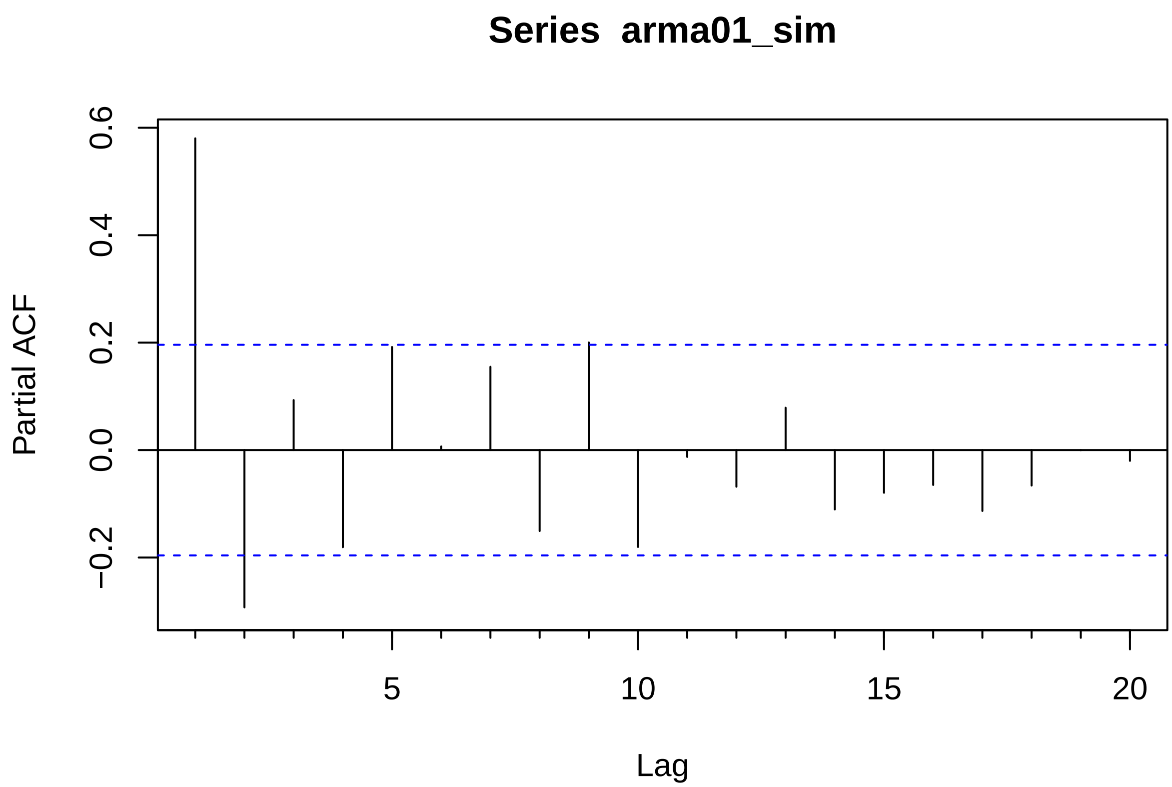


(c) Plot the sample PACF for each of these models in one window to facilitate comparison.

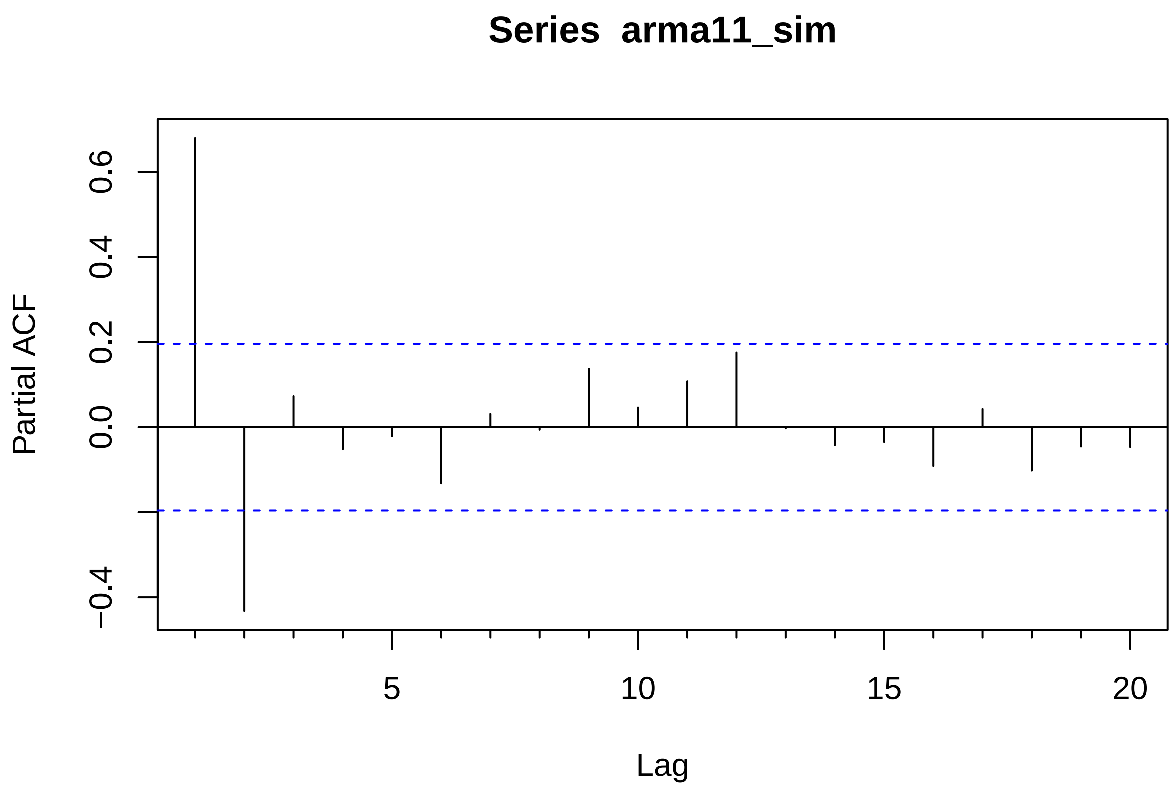
```
Pacf_arma10<-Pacf(arma10_sim)
```



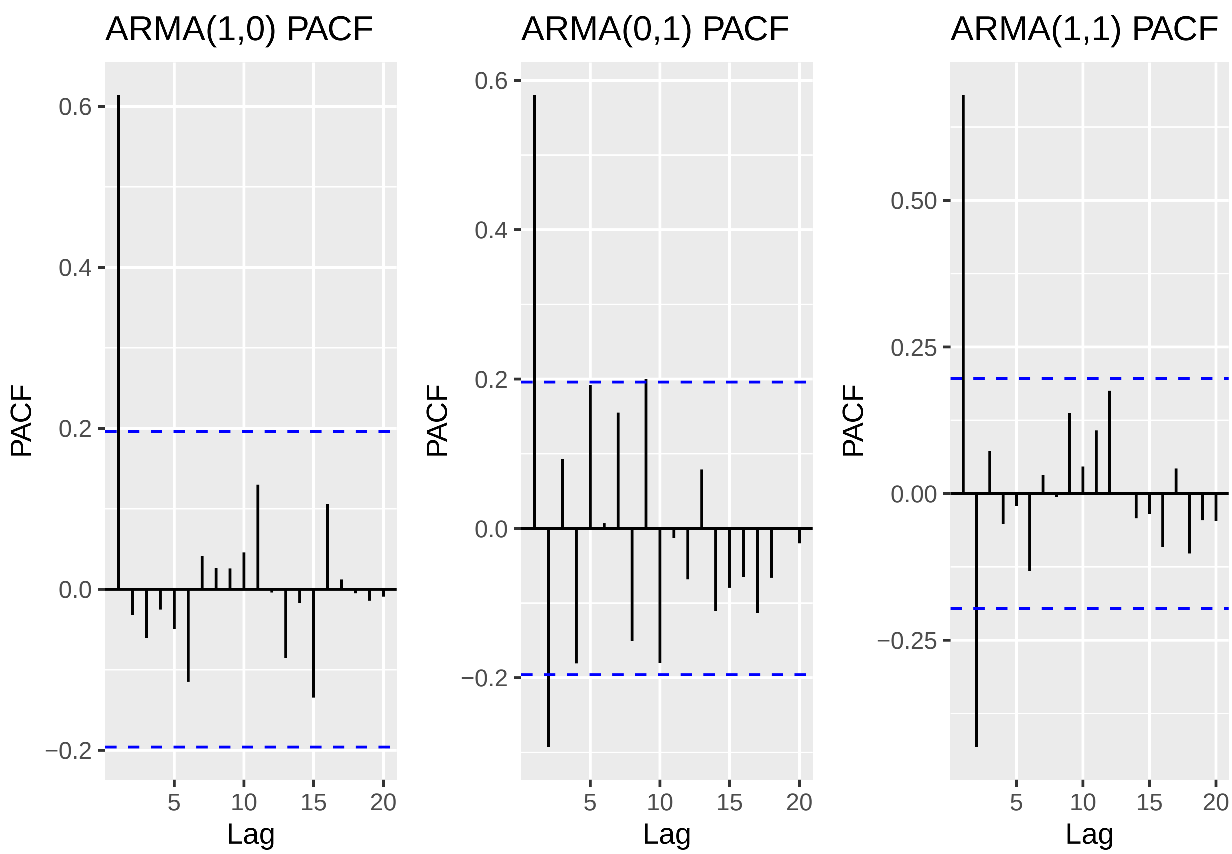

```
Pacf_arma01<-Pacf(arma01_sim)
```




```r
Pacf_arma11<-Pacf(arma11_sim)
```




```
plot_grid(
  autoplot(Pacf_arma10) + ggtitle("ARMA(1,0) PACF"),
  autoplot(Pacf_arma01) + ggtitle("ARMA(0,1) PACF"),
  autoplot(Pacf_arma11) + ggtitle("ARMA(1,1) PACF"),
  nrow = 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 identify them correctly? Explain your answer.

Answer: We would be able to correctly identify the pure AR(1) and MA(1) models because they exhibit clear cut-off point in the PACF and ACF. However, for the ARMA(1,1) model, both ACF and PACF decay gradually without a sharp cut-off, making it more difficult to determine the exact order. We could conclude it is an ARMA model, but identifying the exact (1,1) order would be less straightforward.

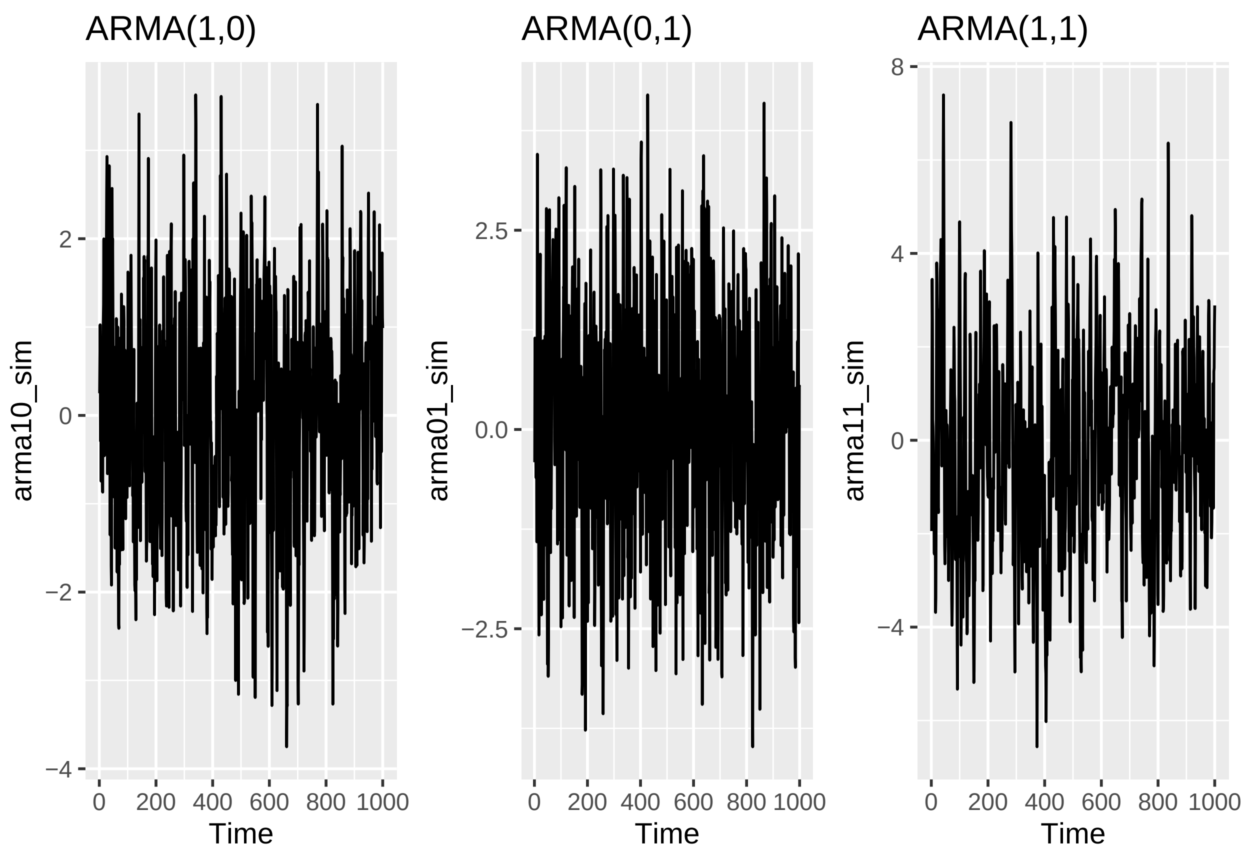
(e) Compare the PACF values R computed with the values you provided for the lag 1 correlation coefficient, i.e., does $\phi = 0.6$ match what you see on PACF for ARMA(1,0), and ARMA(1,1)? Should they match?

Answer: For ARMA(1,0), which is an AR(1) model, the PACF at lag 1 should equal phi = 0.6, and this matches what we observe in the plot. However, for ARMA(1,1), the PACF at lag 1 does not equal phi. The MA component influences the correlation structure, so the first partial autocorrelation is not equal to the AR coefficient alone. Therefore, they should not necessarily match.

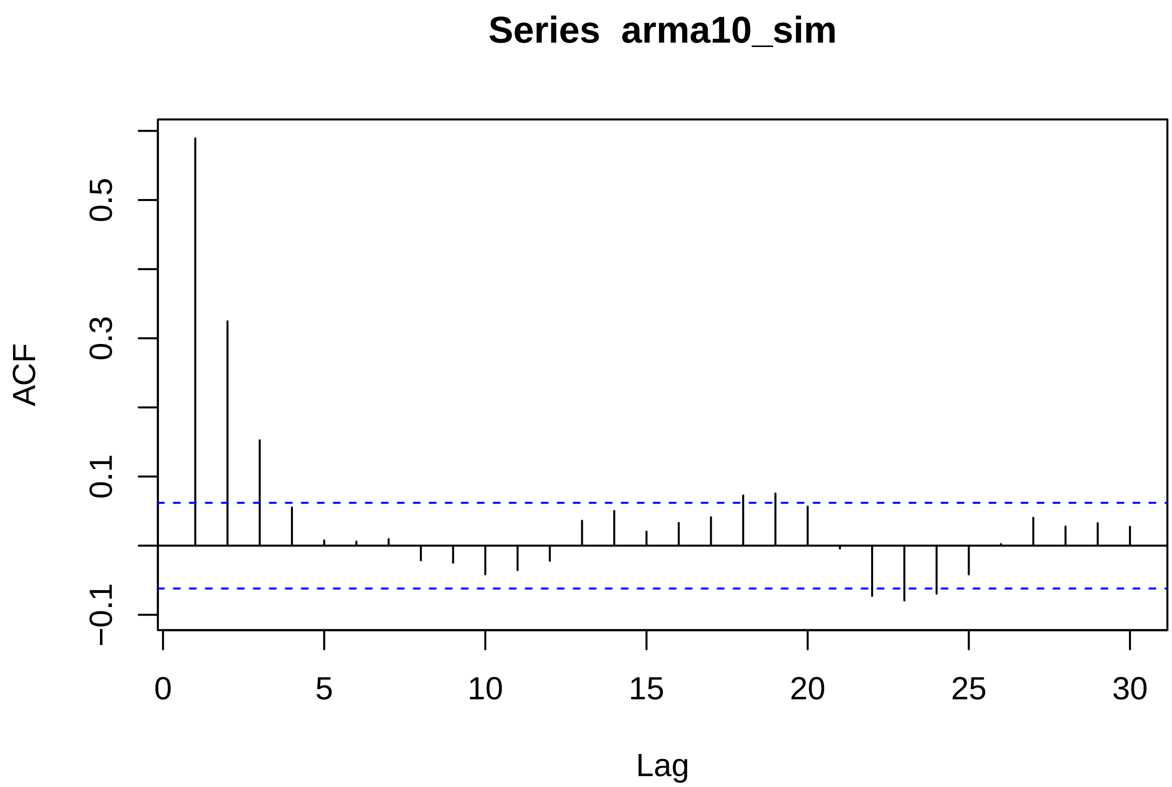
(f) Increase number of observations to $n = 1000$ and repeat parts (b)-(e).

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

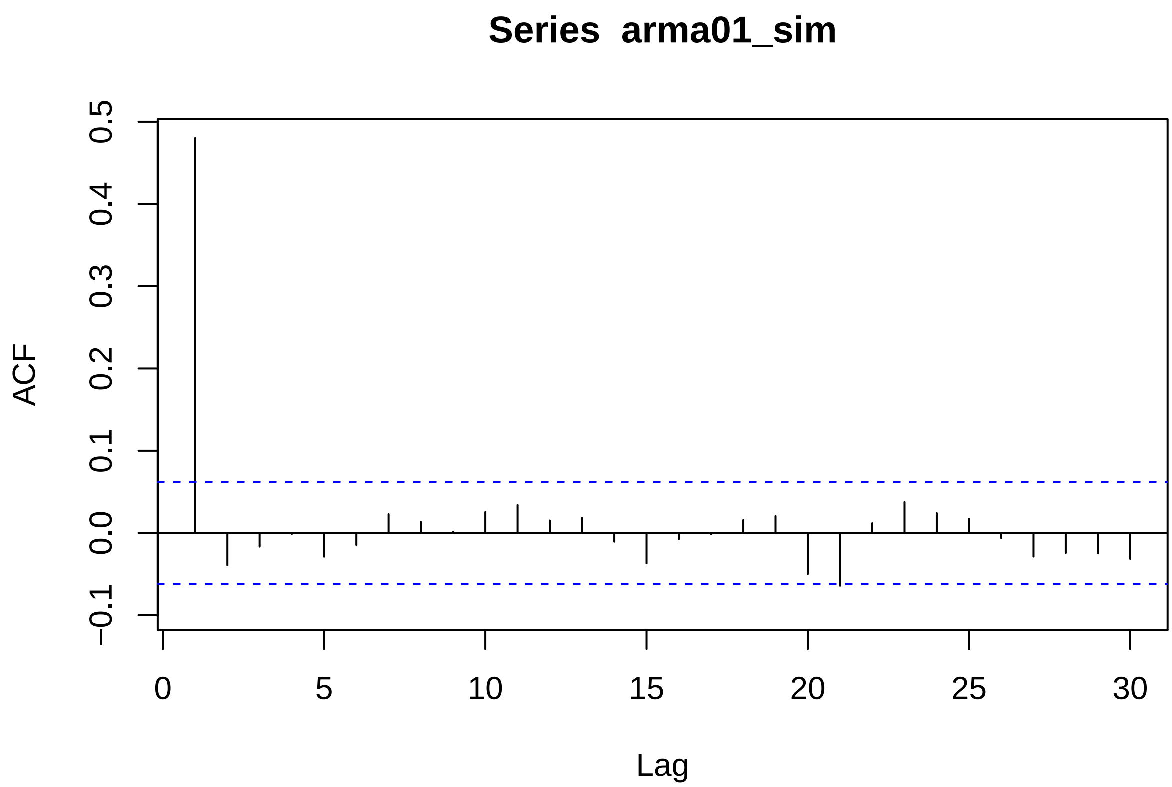
```r
arma10_sim<- arima.sim(model=list(ar=phi),n=n)
arma01_sim<- arima.sim(model=list(ma=theta),n=n)
arma11_sim <- arima.sim(model = list(ar = phi, ma = theta), n = n)

plot_grid(
  autoplot(arma10_sim) + ggtitle("ARMA(1,0)"),
  autoplot(arma01_sim) + ggtitle("ARMA(0,1)"),
  autoplot(arma11_sim) + ggtitle("ARMA(1,1)"),
  nrow = 1
)
```



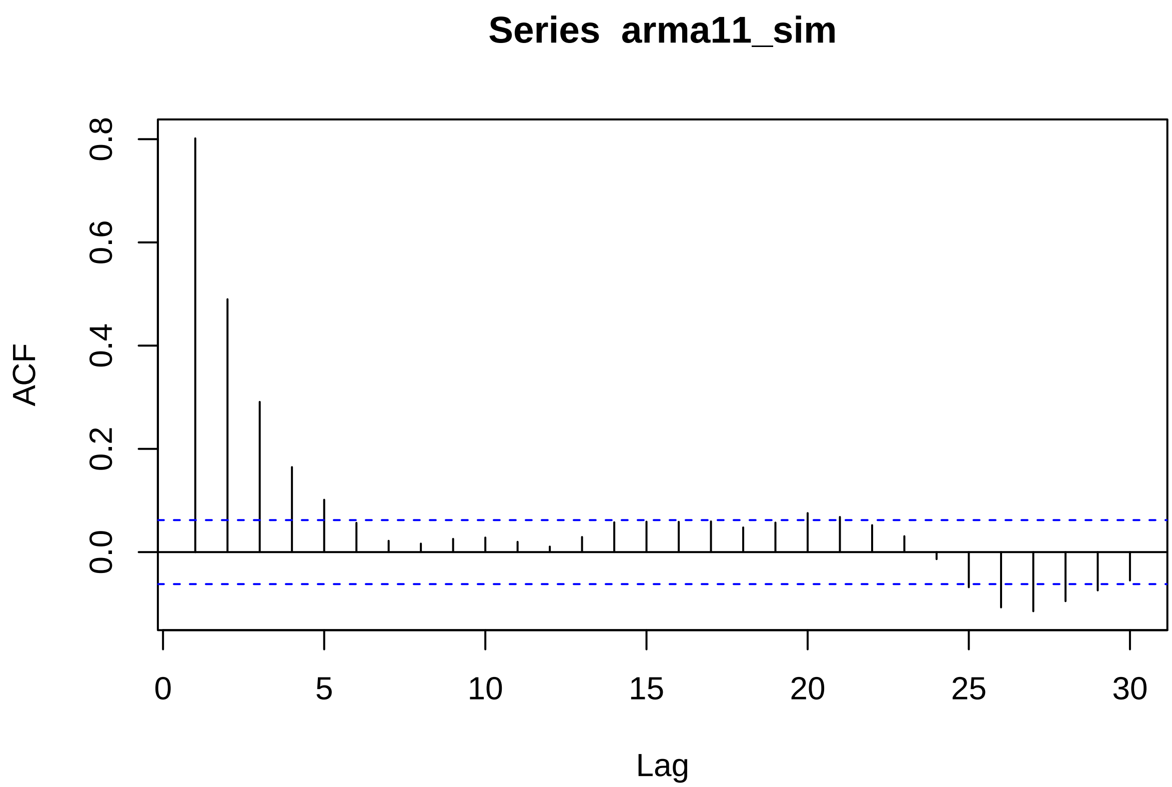

```r
Acf_arma10<-Acf(arma10_sim)
```



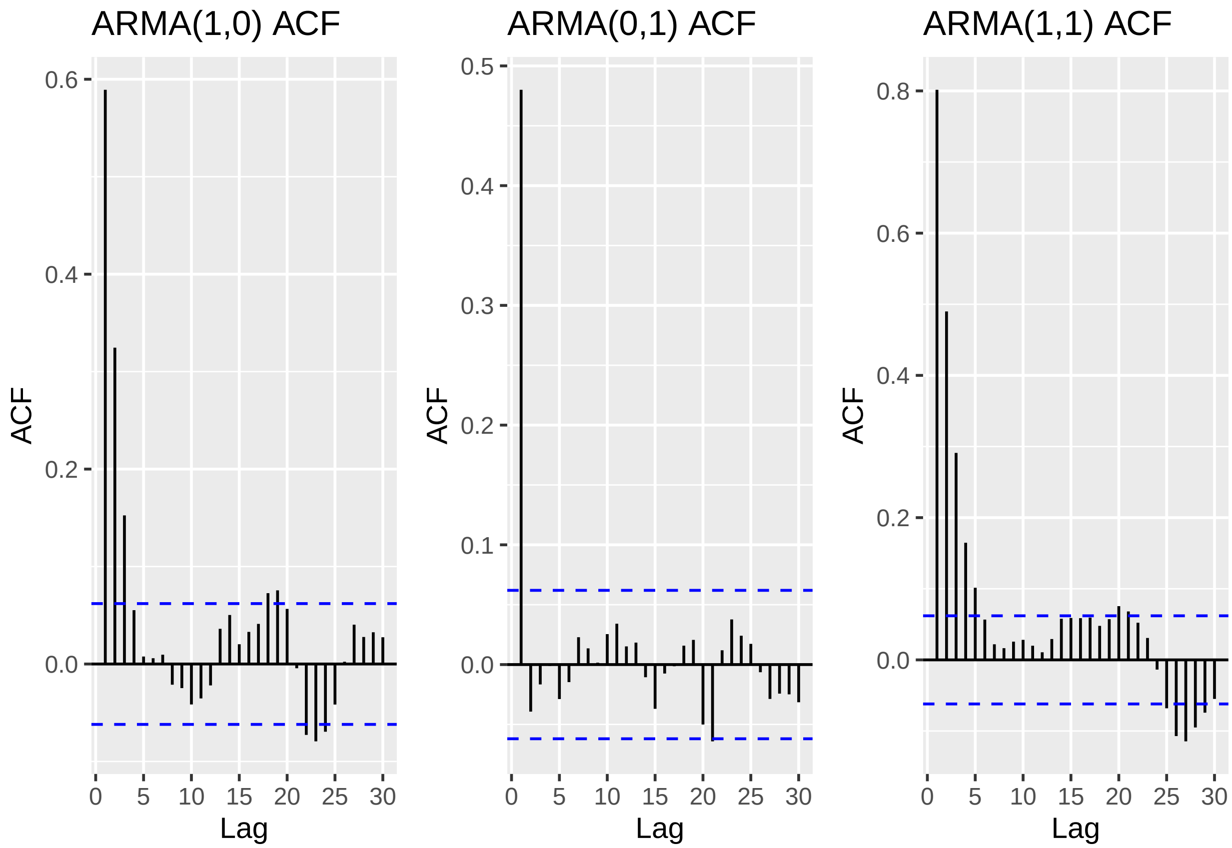

```
Acf_arma01<-Acf(arma01_sim)
```



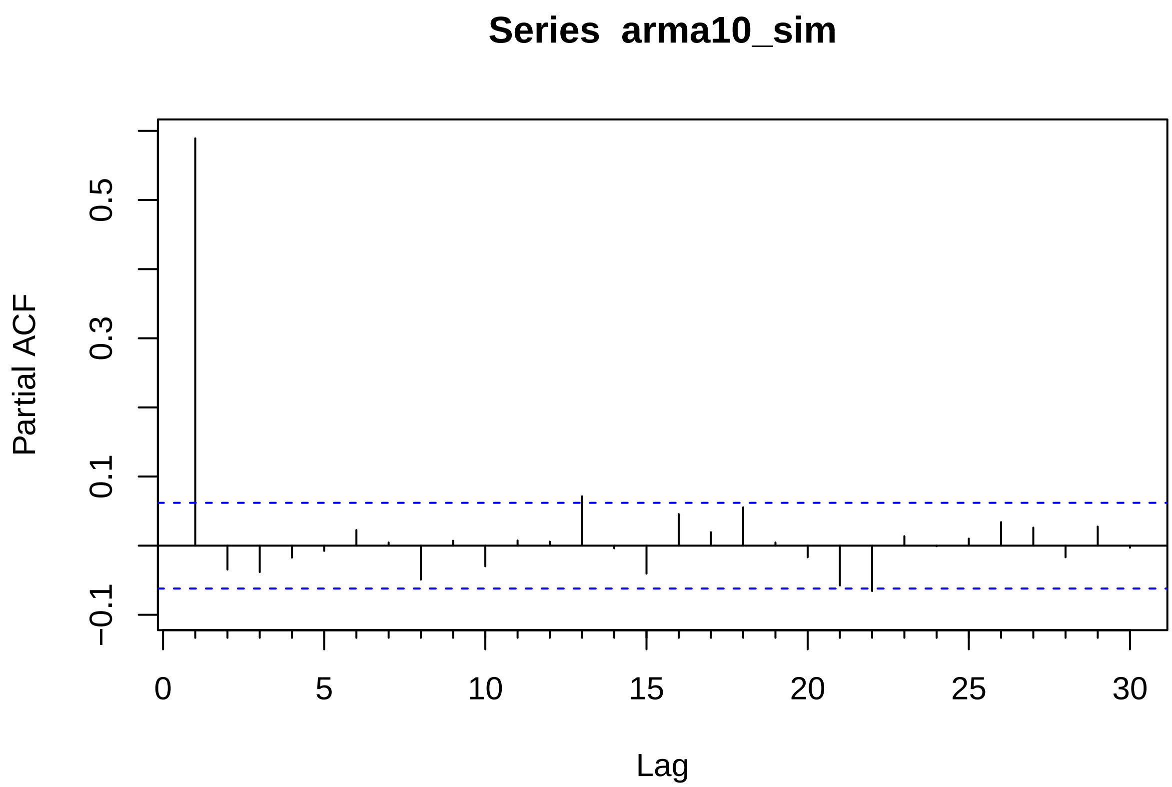

```
Acf_arma11<-Acf(arma11_sim)
```



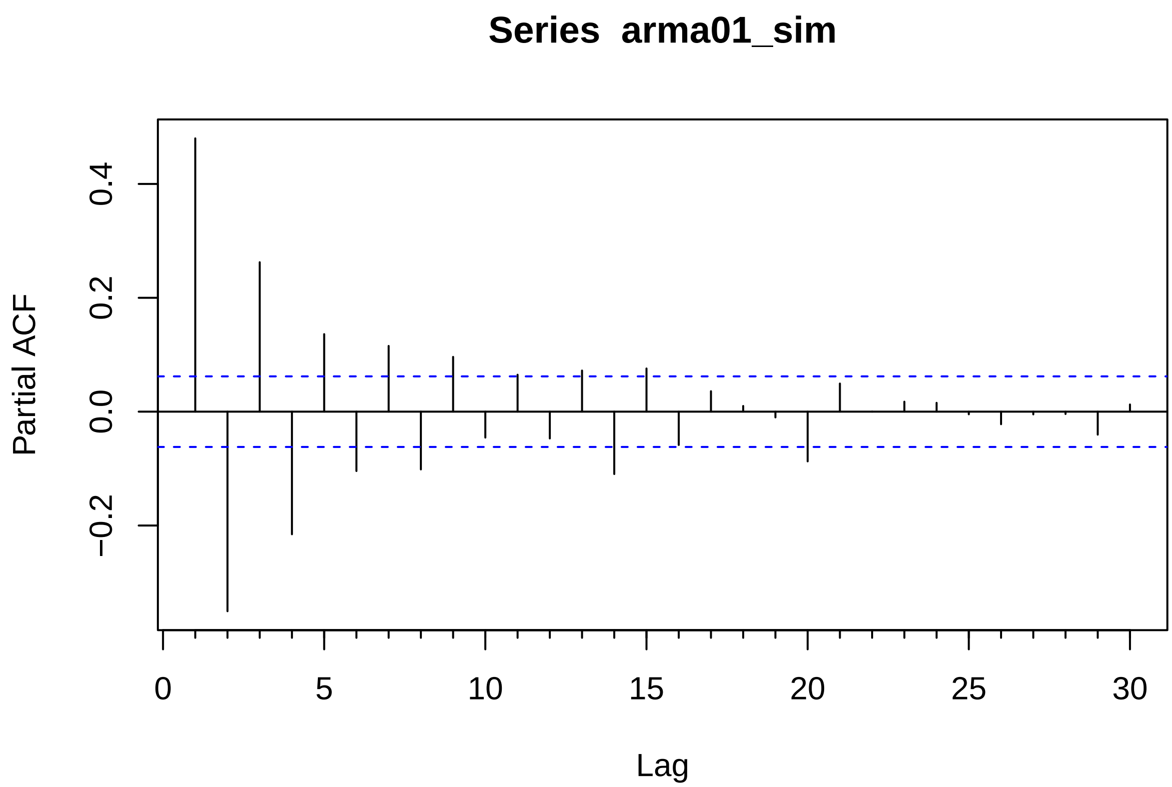

```r
plot_grid(
  autoplot(Acf_arma10) + ggtitle("ARMA(1,0) ACF"),
  autoplot(Acf_arma01) + ggtitle("ARMA(0,1) ACF"),
  autoplot(Acf_arma11) + ggtitle("ARMA(1,1) ACF"),
  nrow = 1
)
```



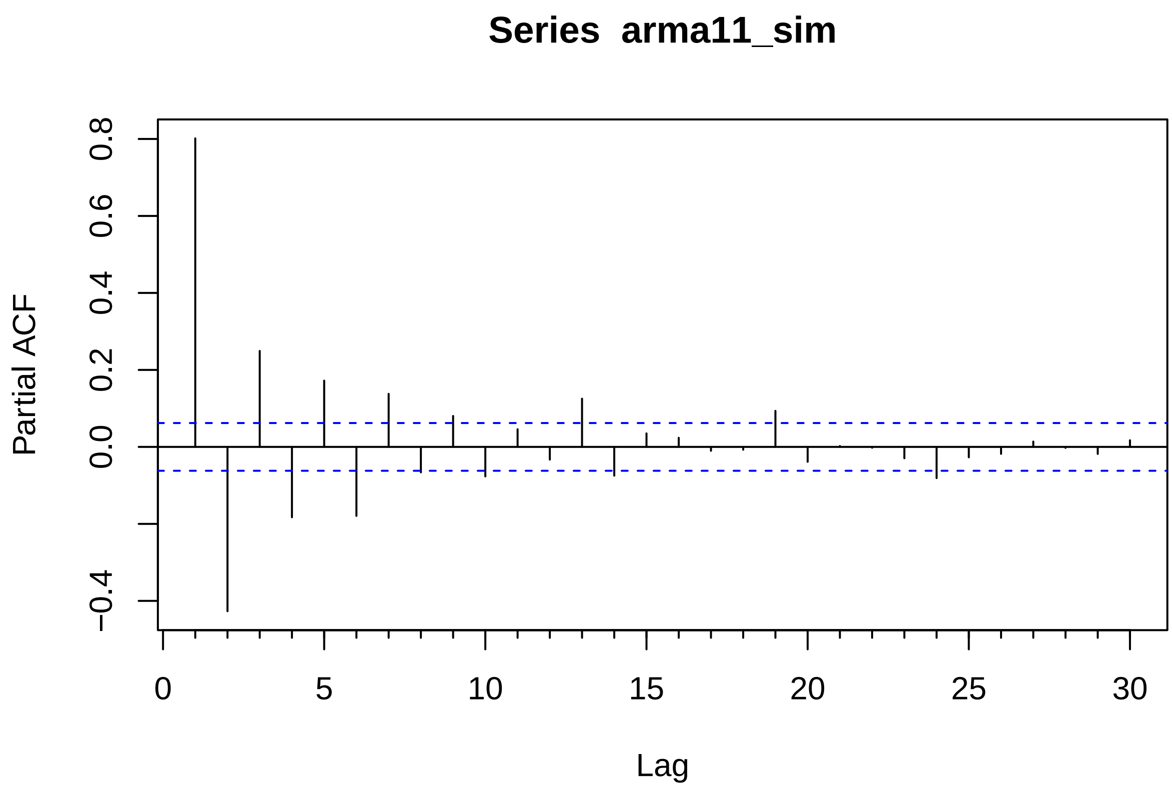

```
Pacf_arma10<-Pacf(arma10_sim)
```



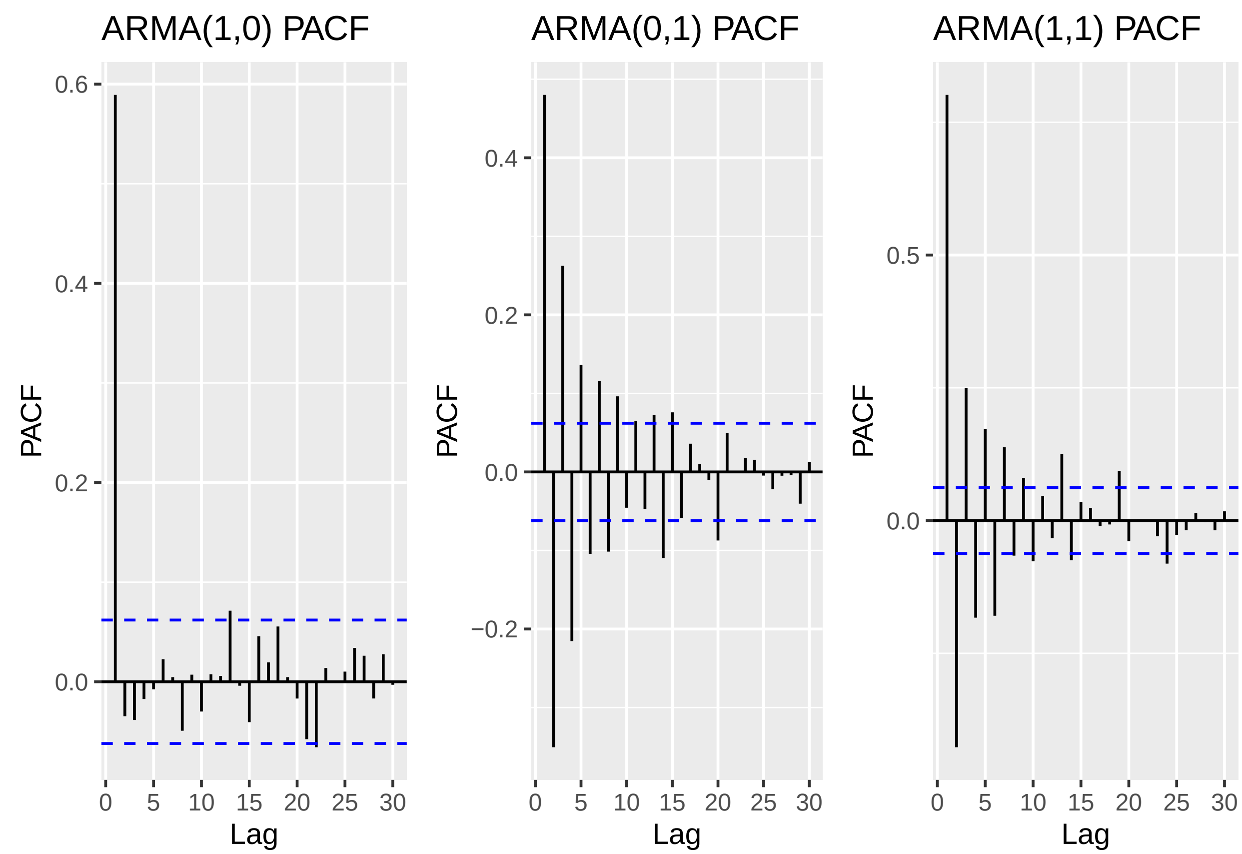

```
Pacf_arma01<-Pacf(arma01_sim)
```




```
Pacf_arma11<-Pacf(arma11_sim)
```




```r
plot_grid(
  autoplot(Pacf_arma10) + ggtitle("ARMA(1,0) PACF"),
  autoplot(Pacf_arma01) + ggtitle("ARMA(0,1) PACF"),
  autoplot(Pacf_arma11) + ggtitle("ARMA(1,1) PACF"),
  nrow = 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 identify them correctly? Explain your answer.

Answer: With n = 1000, the ACF and PACF patterns become much clearer. Cut=off point can now be identified for the AR(1) and MA(1) models. The ARMA(1,1) model is also easier to distinguish from AR or MA, but its exact order still cannot be determined perfectly from ACF/PACF alone.

(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 the AR(1) model, the PACF at lag 1 should equal phi = 0.6, and with n = 1000 the estimate is very close to 0.6. However, for the ARMA(1,1) model, the PACF at lag 1 does not equal phi because the MA component influences the partial autocorrelation structure. Therefore, they should not necessarily match.

Q3

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

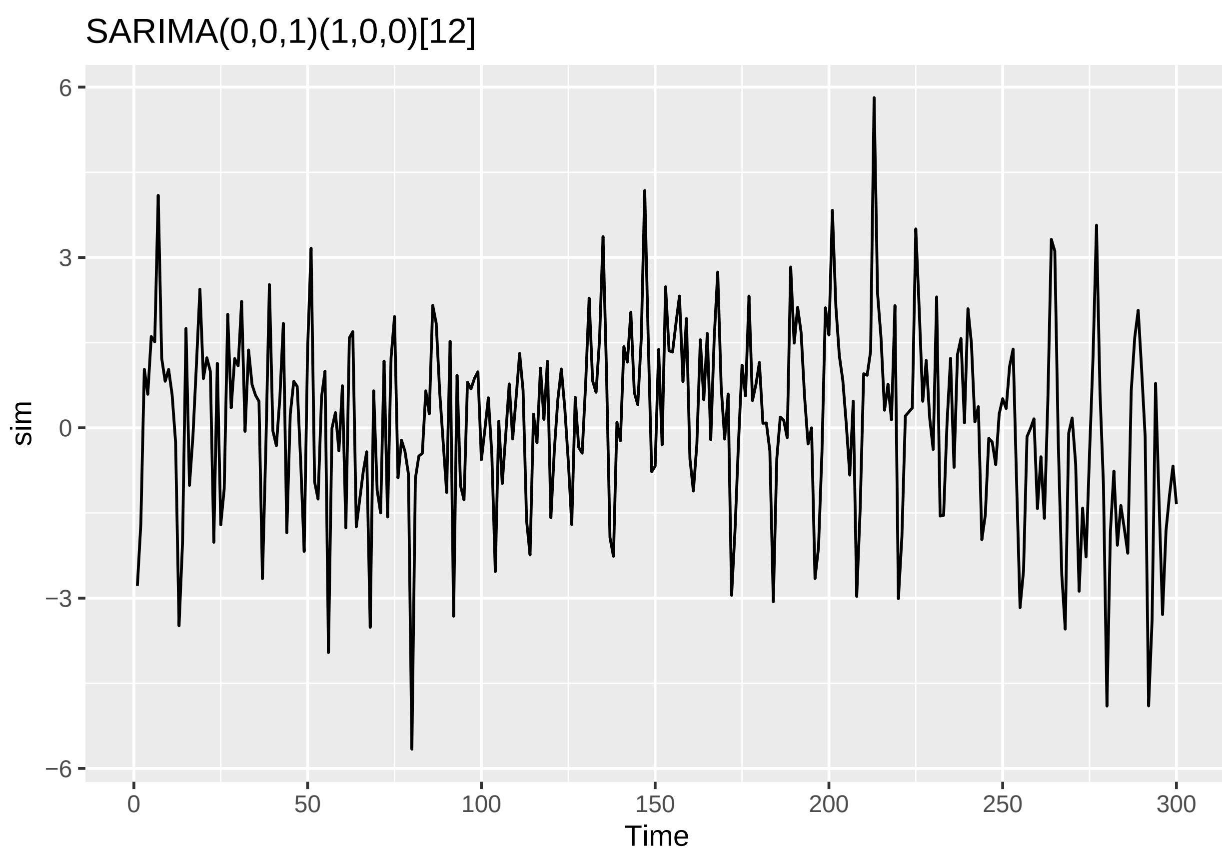
(a) Identify the model using the notation ARIMA$(p, d, q)(P, D, Q)_s$, i.e., identify the integers p, d, q, P, D, Q, s (if possible) from the equation. > Answer: ARIMA(1,0,1)(1,0,0)_12

(b) Also from the equation what are the values of the parameters, i.e., model coefficients. > Answer: The model coefficients are: Non-seasonal AR coefficient: phi1 = 0.7 Seasonal AR coefficient: phi1 = -0.25 Non-seasonal MA coefficient: phi1 = -0.1

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?

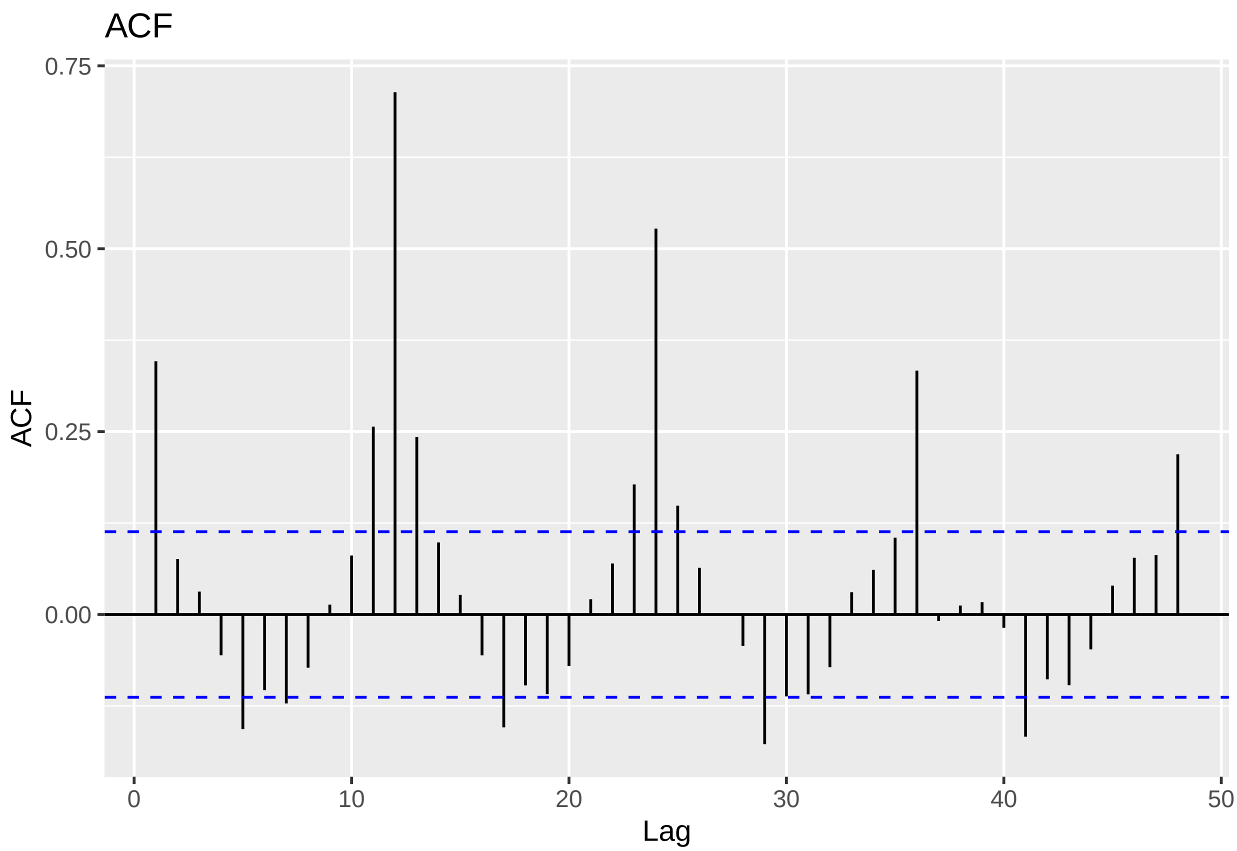
```r
set.seed(123)

n <- 300

ar_coefs <- rep(0, 12)
ar_coefs[12] <- 0.8   # seasonal AR

sim <- arima.sim(
  model = list(
    ar = ar_coefs,
    ma = 0.5
  ),
  n = n
)

autoplot(sim) +
  ggtitle("SARIMA(0,0,1)(1,0,0)[12]")
```



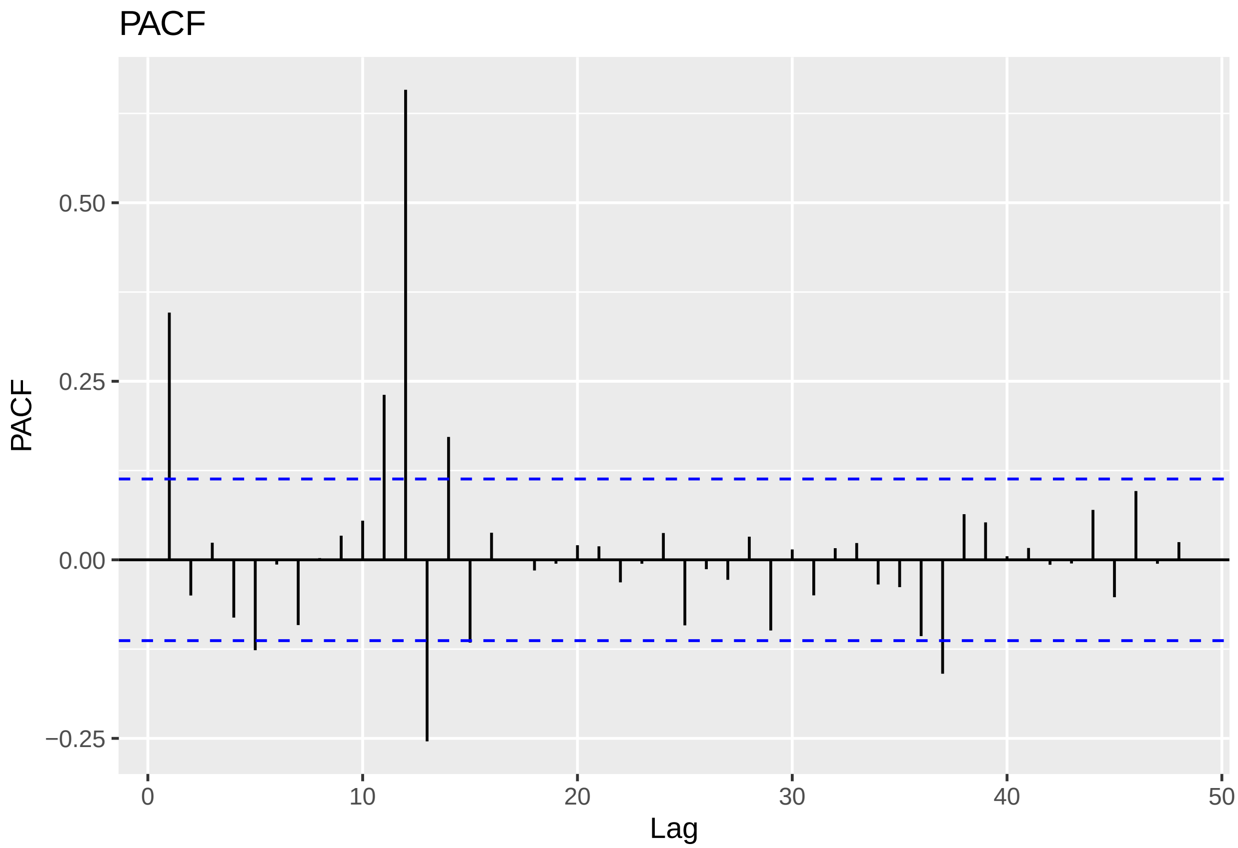
Q5

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.

```
ggAcf(sim, lag.max = 48) +
  ggtitle("ACF")
```




```
ggPacf(sim, lag.max = 48) +
  ggtitle("PACF")
```



> Answer: The ACF shows a significant spike at lag 1, consistent with a non-seasonal MA(1) component. Additionally, there are strong spikes at lags 12, 24, and 36 that gradually decay, which is characteristic of a seasonal AR(1) structure. The PACF shows a big spike at lag 12 and no repeating seasonal spikes afterward, which supports the presence of a seasonal AR(1) component. Therefore, from the ACF and PACF plots, we would be able to identify both the non-seasonal MA(1) and the seasonal AR(1) components correctly.