ENV 797 - Time Series Analysis for Energy and Environment Applications | Spring 2024

Assignment 6 - Due date 02/28/24

Sai Powar

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_Sp24.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(lubridate)
```

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

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

```
library(ggplot2)
library(forecast)
```

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

```
library(Kendall)
library(tseries)
library(outliers)
library(tidyverse)
```

```
## -- 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(cowplot)
```

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

```r
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 plot shows an exponential decay and the PACF model gives the order of the model, such that the first lag is significant/cuts off and then the following lags are within the blue lines.

- MA(1)

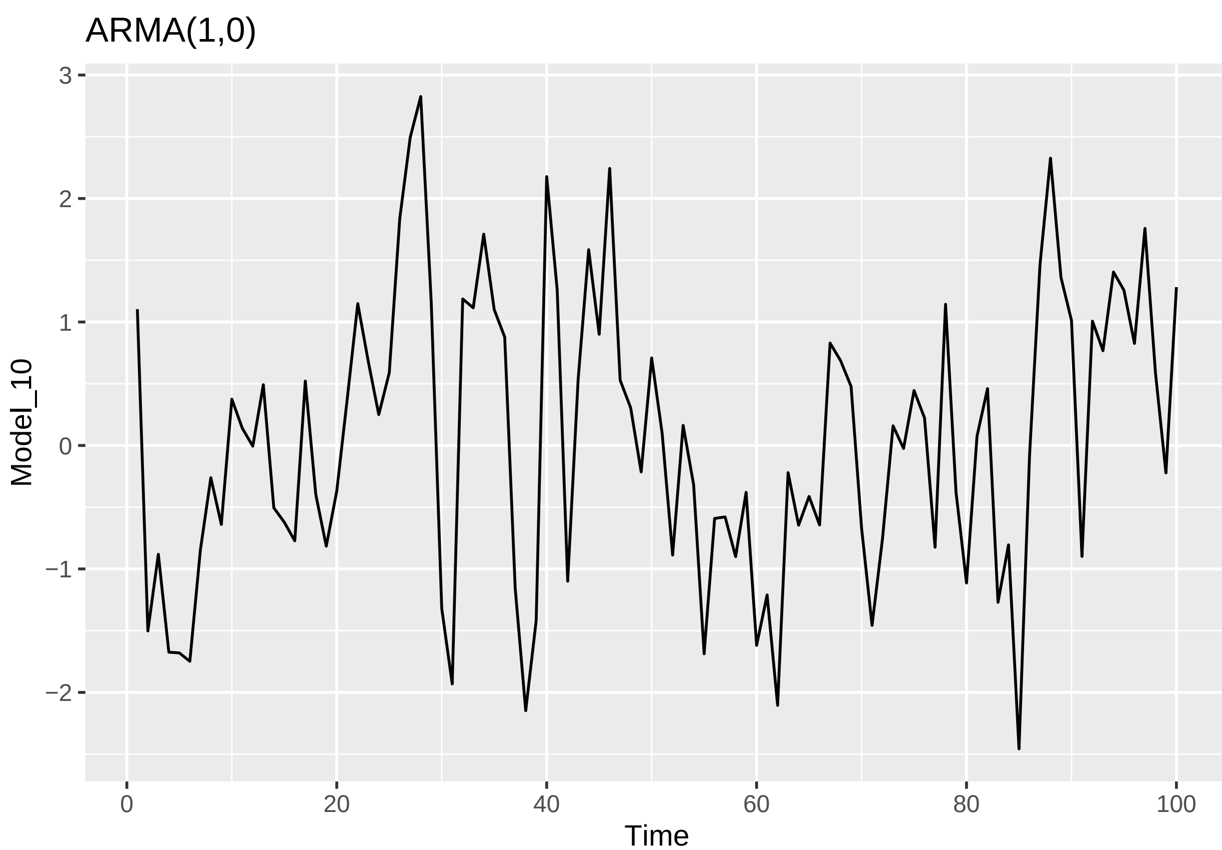
  Answer: The ACF plot gives the order of the model, such that the first lag is significantly negative/cuts off and the following lags are within the limits. The PACF plots shows an exponential decay.

Q2

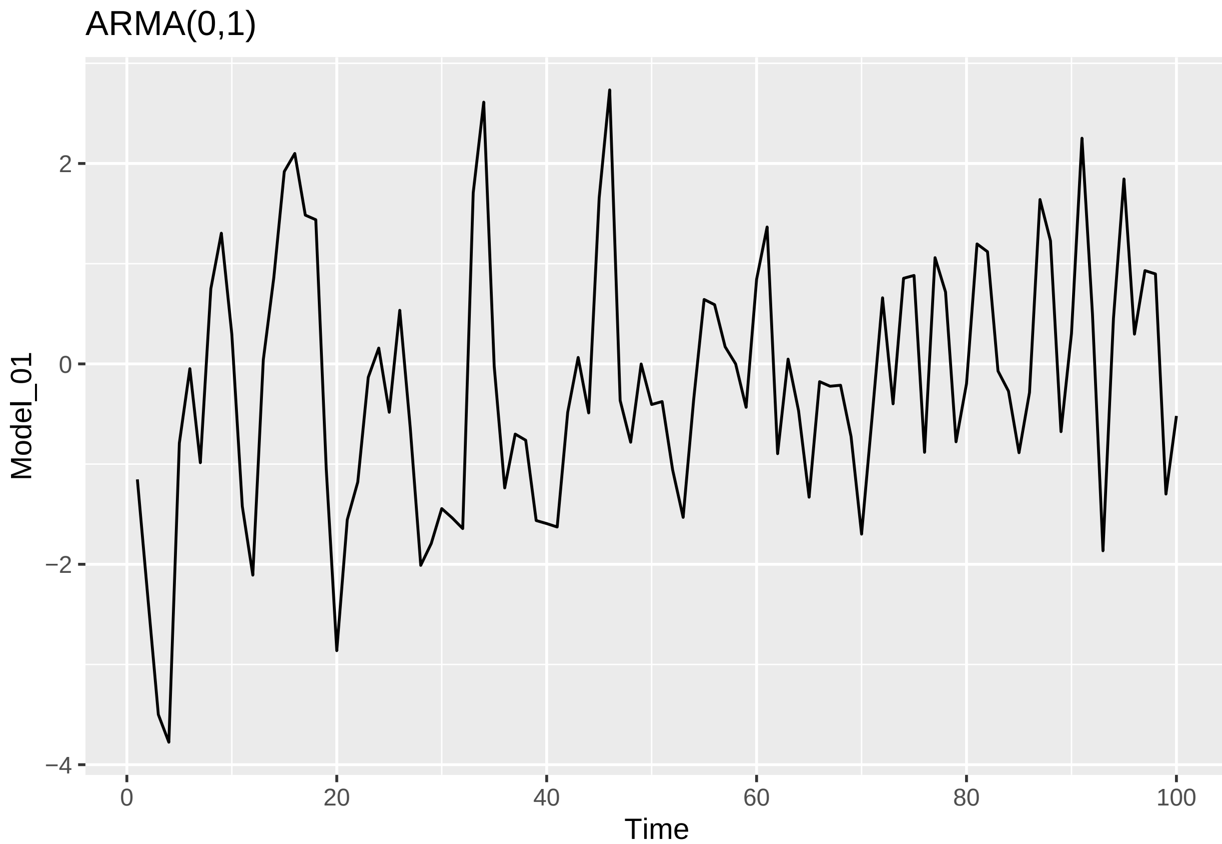
Recall that the non-seasonal ARIMA is described by three parameters ARIMA(p, d, q) where p is the order of the autoregressive component, d is the number of times the series need to be differenced to obtain stationarity and q is the order of the moving average component. If we don't need to difference 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.

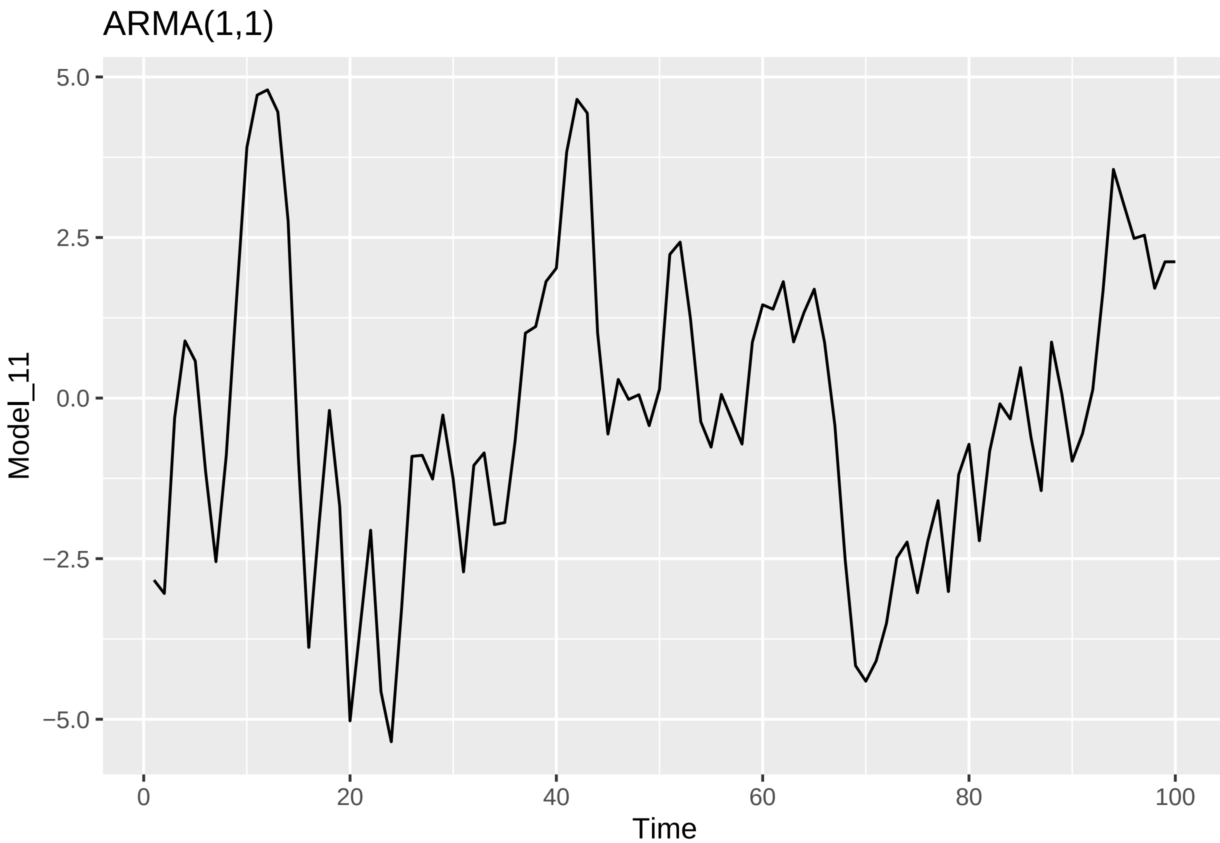
```
Model_10 <- arima.sim(model=list(ar = 0.6, order=c(1,0,0)), n=100)
Model_01 <- arima.sim(model=list(ma = 0.9, order=c(0,0,1)), n=100)
Model_11 <- arima.sim(model=list(ar = 0.6, ma = 0.9, order=c(1,0,1)), n=100)


autoplot(Model_10, main = "ARMA(1,0)")
```




```
autoplot(Model_01, main = "ARMA(0,1)")
```



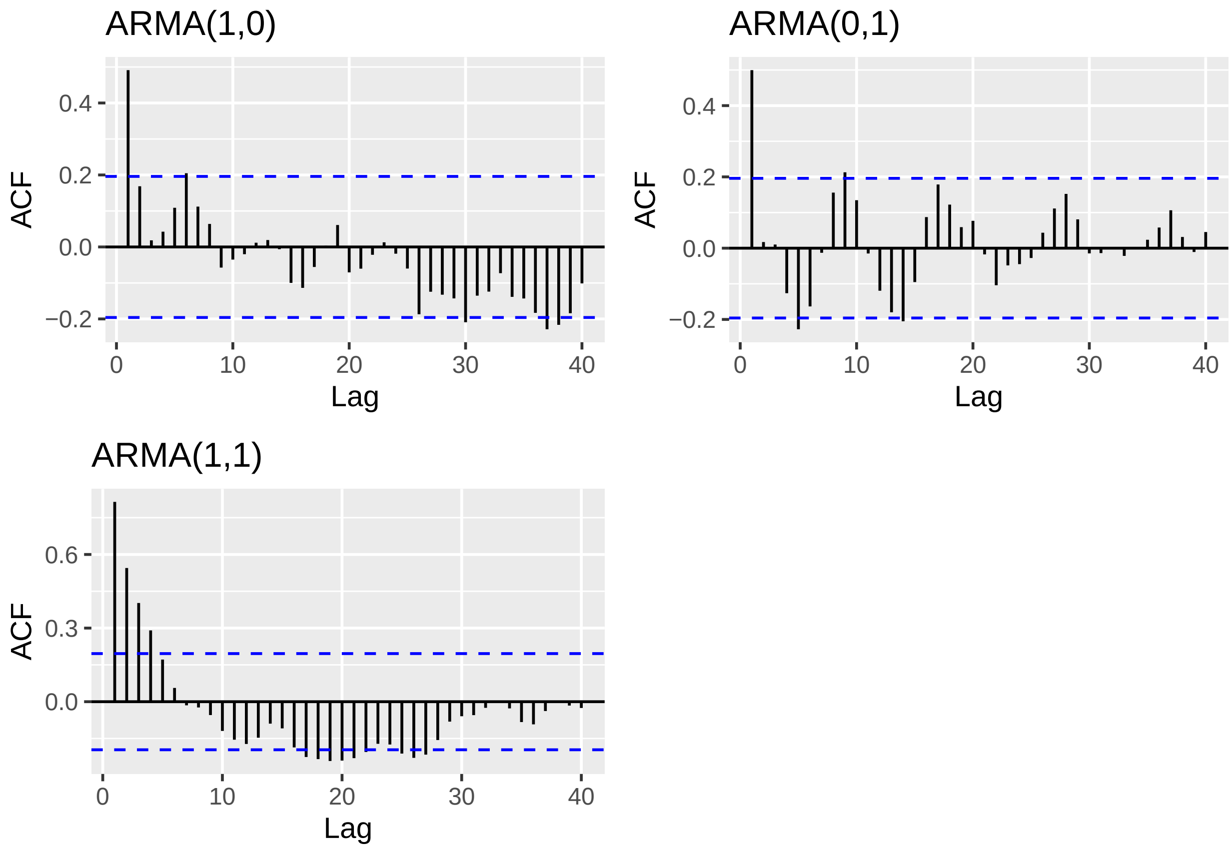

```r
autoplot(Model_11, main = "ARMA(1,1)")
```



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

```
plot_grid(
  autoplot(Acf(Model_10, lag = 40, plot=FALSE), main = "ARMA(1,0)"),
  autoplot(Acf(Model_01, lag = 40, plot=FALSE), main = "ARMA(0,1)"),
  autoplot(Acf(Model_11, lag = 40, plot=FALSE), main = "ARMA(1,1)"))
```

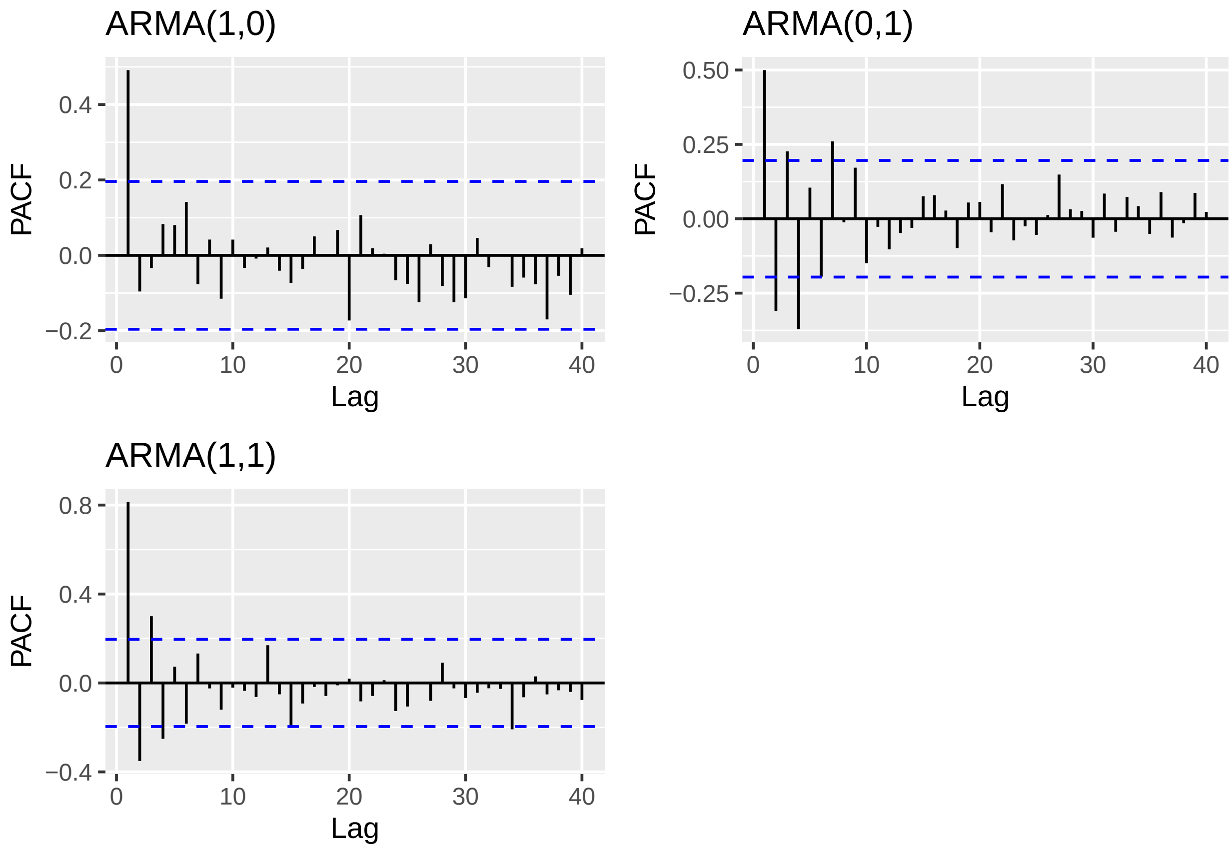
```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
```



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

```r
plot_grid(
  autoplot(Pacf(Model_10, lag = 40, plot=FALSE), main = "ARMA(1,0)"),
  autoplot(Pacf(Model_01, lag = 40, plot=FALSE), main = "ARMA(0,1)"),
  autoplot(Pacf(Model_11, lag = 40, plot=FALSE), main = "ARMA(1,1)"))
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
```



(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: By looking at the ACF plot of the AR model, the exponential decay pattern is clear so I would be able to identify that correctly. But, the PACF plot does not have the characteristic cut off that we discussed in class so identifying the order would be challenging. For the MA model, the PACF model kind of shows an exponential decay but it is not clear so that would also be something that I would look for additional information to confirm. For the ARMA model, the ACF and PACF plots look easier to identify for ARMA models. The ACF plot has the decay we would see for the AR part of a model and same with the PACF.

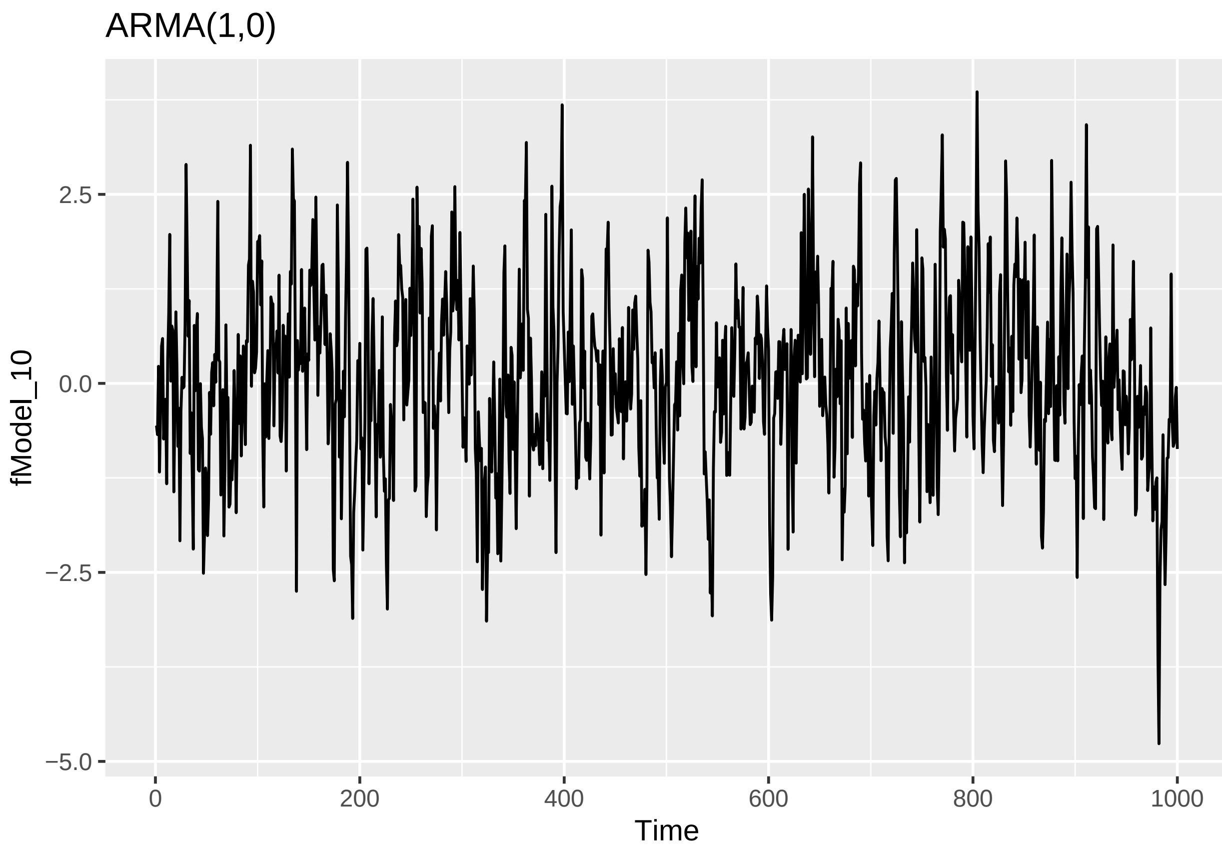
(e) Compare the PACF values R computed with the values you provided for the lag 1 correlation coefficient, i.e., does $\phi = 0.6$ match what you see on PACF for ARMA(1,0), and ARMA(1,1)? Should they match?

Answer: The PACF values for ARMA(1,0) is slightly less than 0.5 and that for ARMA(1,1) is more than 0.6. This makes sense because the PACF value is higher than 0.6 when the model has both AR and MA components.

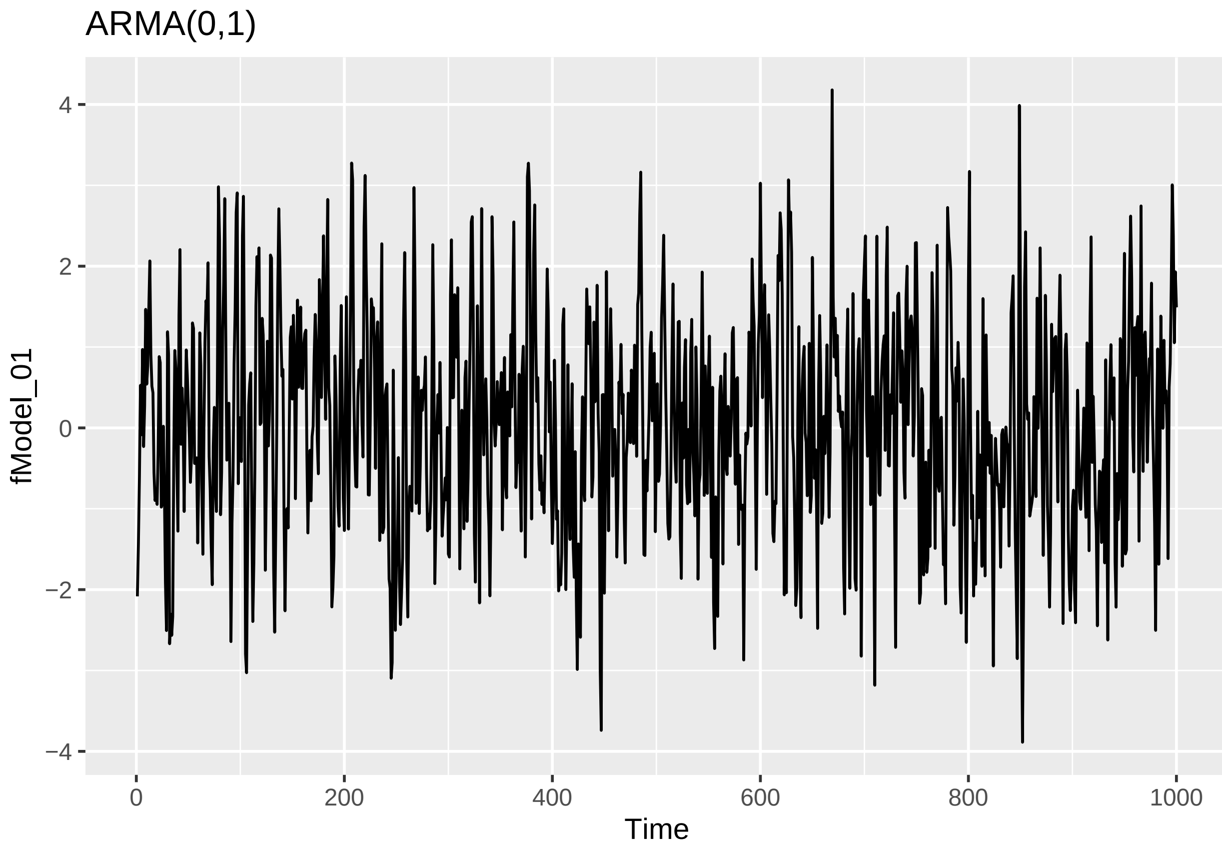
(f) Increase number of observations to $n = 1000$ and repeat parts (b)-(e).

```
fModel_10 <- arima.sim(model=list(ar = 0.6, order=c(1,0,0)), n=1000)
fModel_01 <- arima.sim(model=list(ma = 0.9, order=c(0,0,1)), n=1000)
fModel_11 <- arima.sim(model=list(ar = 0.6, ma = 0.9, order=c(1,0,1)), n=1000)
```

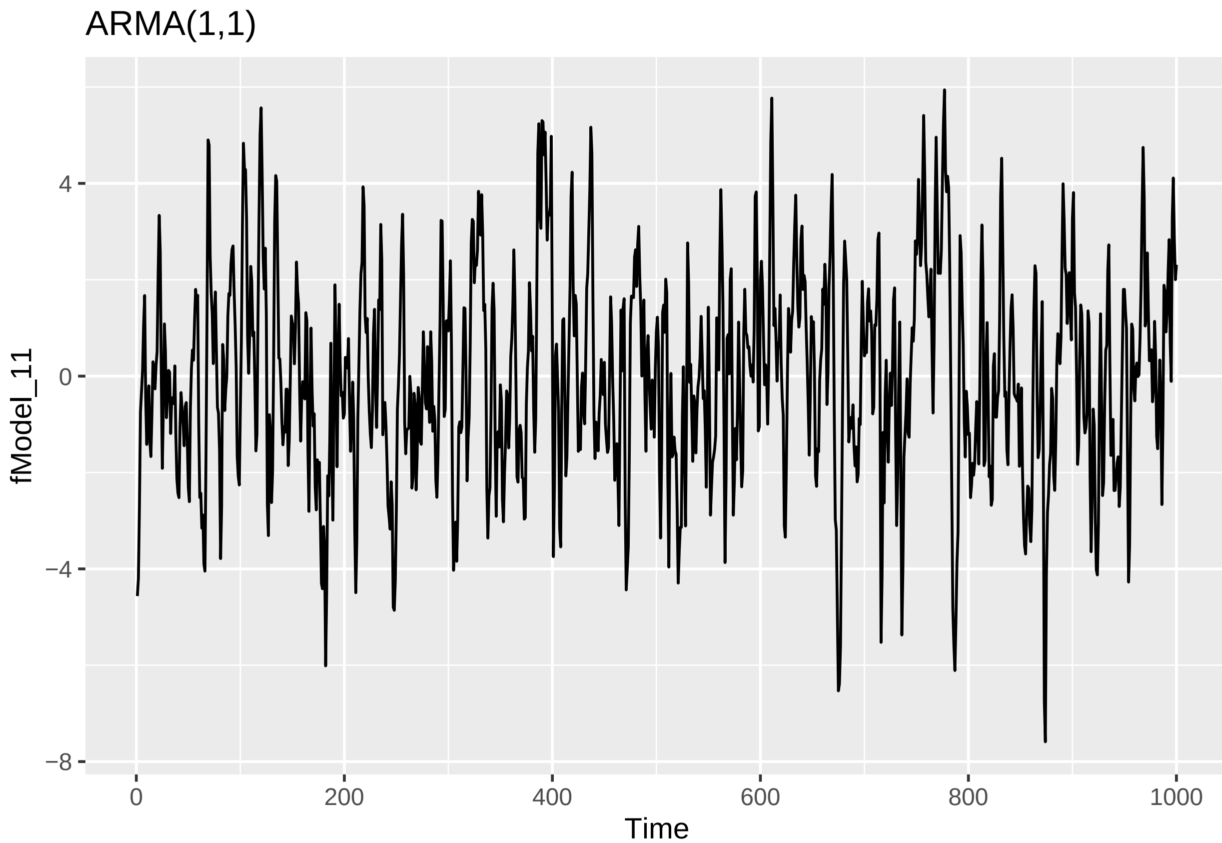
```r
autoplot(fModel_10, main = "ARMA(1,0)")
```




```r
autoplot(fModel_01, main = "ARMA(0,1)")
```

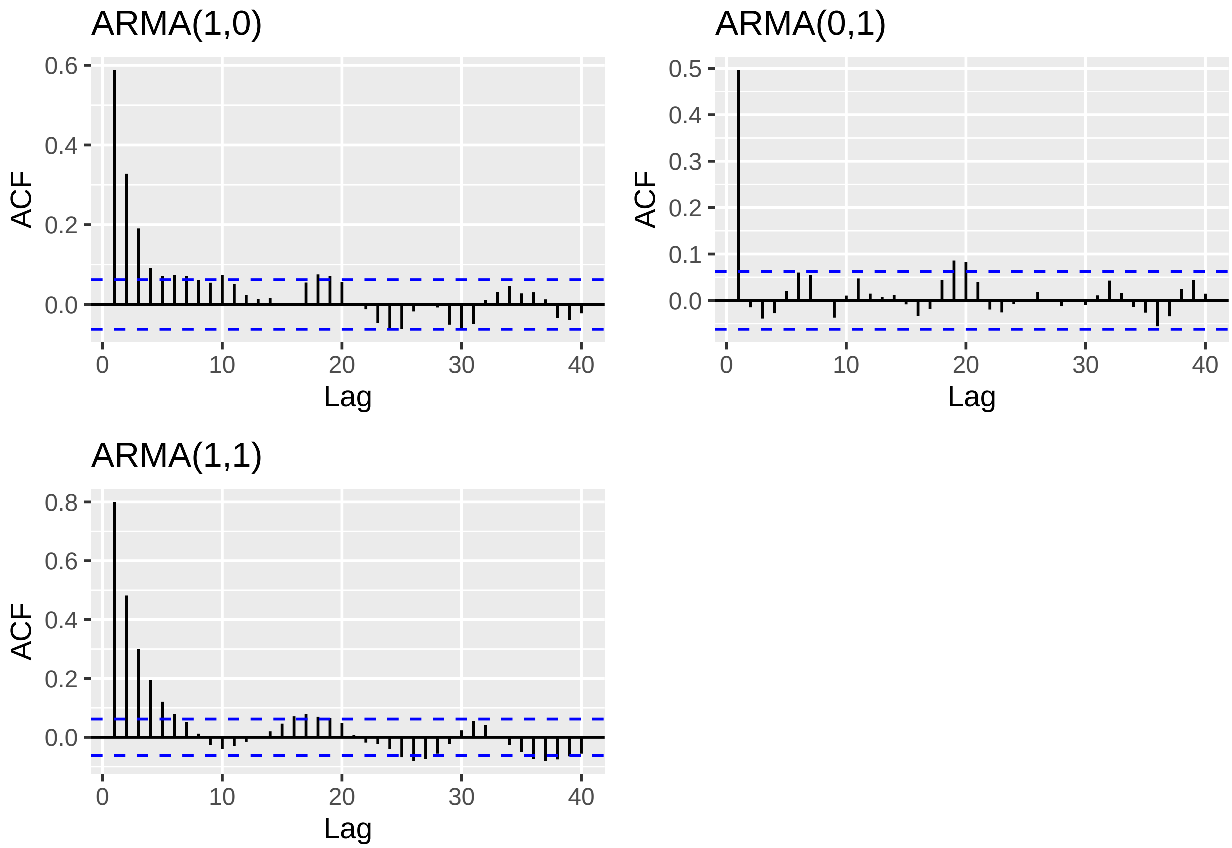



```
autoplot(fModel_11, main = "ARMA(1,1)")
```



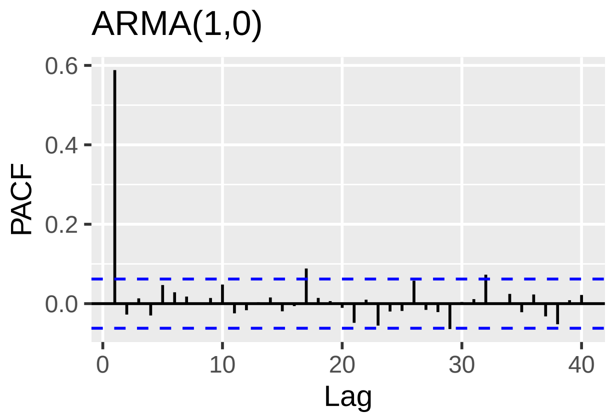

```r
plot_grid(
  autoplot(Acf(fModel_10, lag = 40, plot=FALSE), main = "ARMA(1,0)"),
  autoplot(Acf(fModel_01, lag = 40, plot=FALSE), main = "ARMA(0,1)"),
  autoplot(Acf(fModel_11, lag = 40, plot=FALSE), main = "ARMA(1,1)"))
```

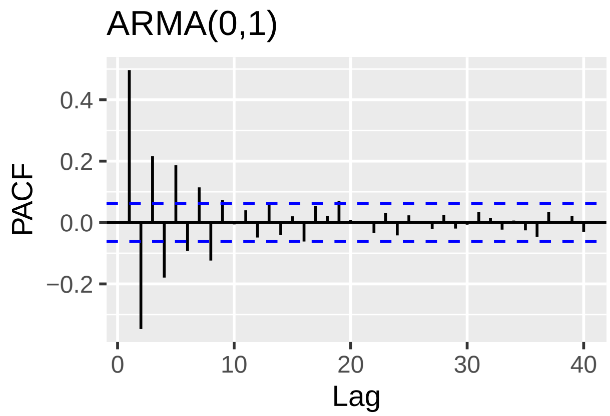
```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
```

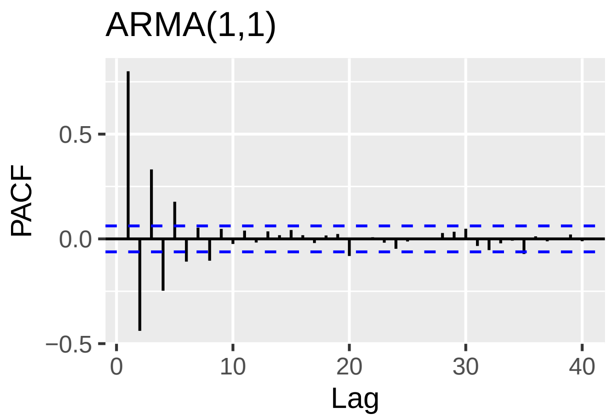



```r
plot_grid(
  autoplot(Pacf(fModel_10, lag = 40, plot=FALSE), main = "ARMA(1,0)"),
  autoplot(Pacf(fModel_01, lag = 40, plot=FALSE), main = "ARMA(0,1)"),
  autoplot(Pacf(fModel_11, lag = 40, plot=FALSE), main = "ARMA(1,1)"))
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
```







Answer: With the increase in observations, it is easier to identify the exponential decay patterns and the cutoff points in the ACF and PACF plots for all three models.

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$(1, 0, 1)(1, 0, 0)_1 2$

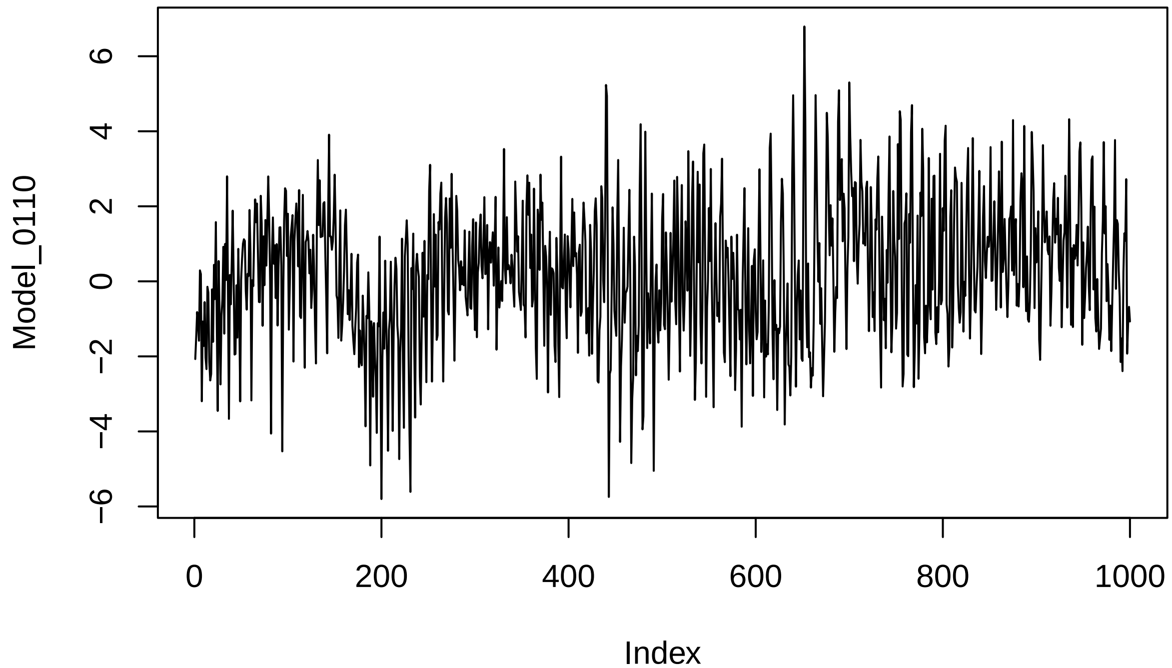
(b) Also from the equation what are the values of the parameters, i.e., model coefficients.

Answer: AR coefficient $= 0.7$, SAR coefficient $= 0.25$, MA coefficient $= 0.1$

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?

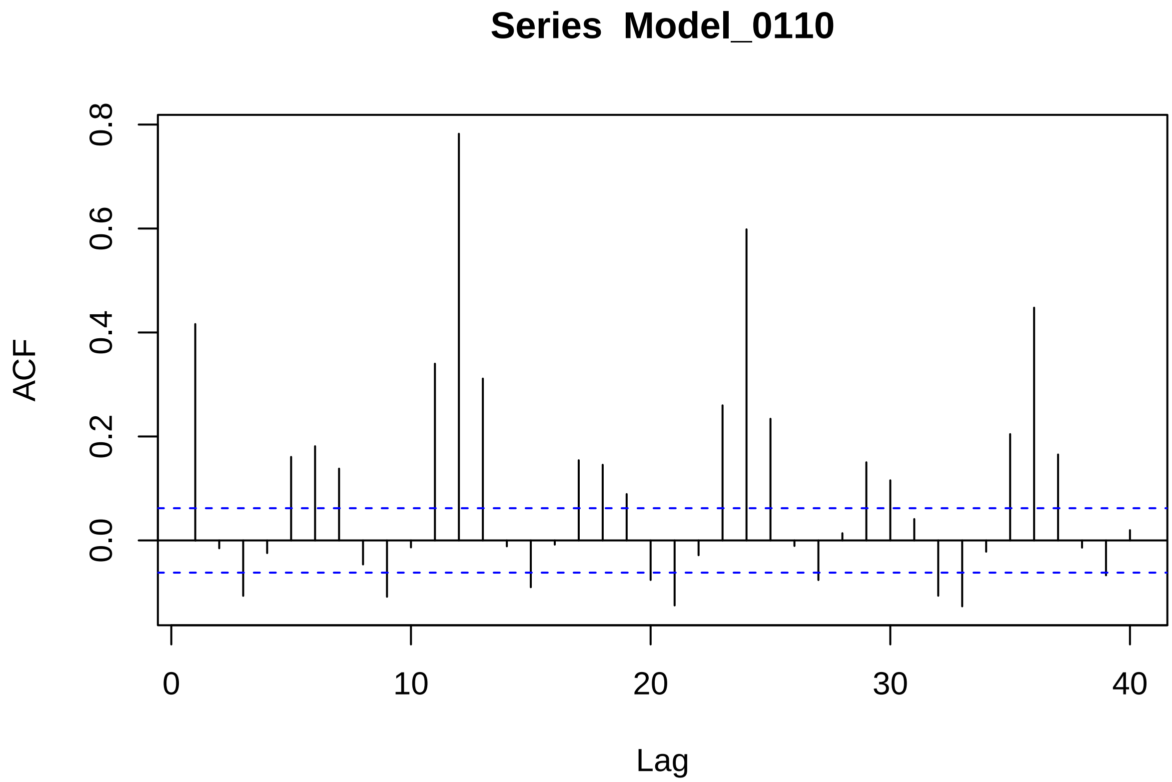
```r
Model_0110 <- sim_sarima(model=list(ma=0.5,sar=0.8,nseasons=12), n = 1000)
plot(Model_0110,type="l")
```

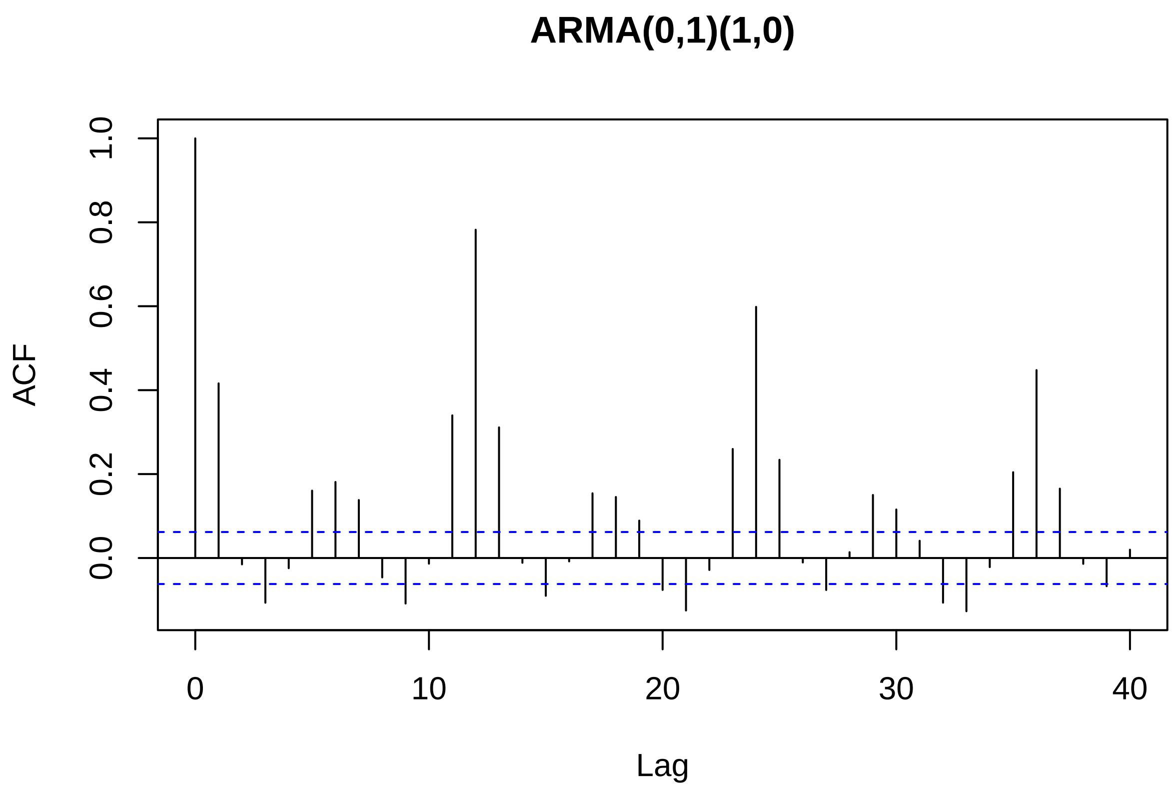


Q5

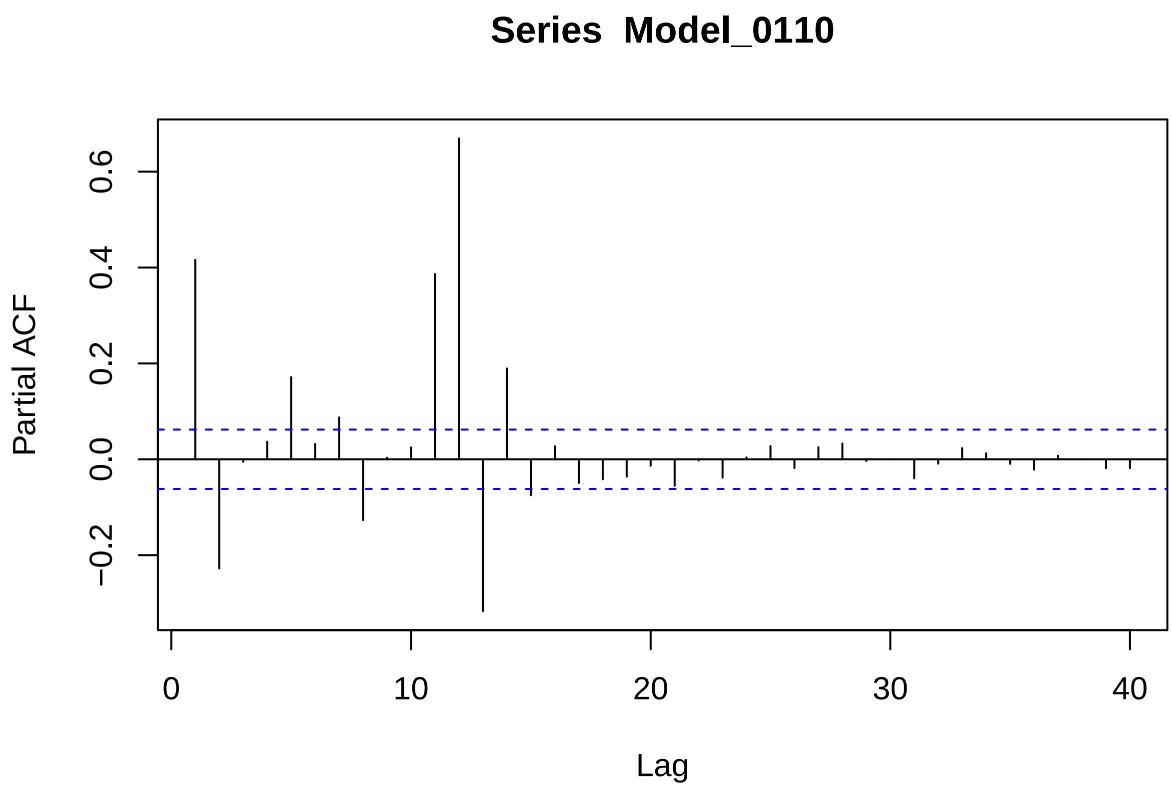
Plot ACF and PACF of the simulated series in Q4. Comment if the plots are well representing the model you simulated, i.e., would you be able to identify the order of both non-seasonal and seasonal components from the plots? Explain.

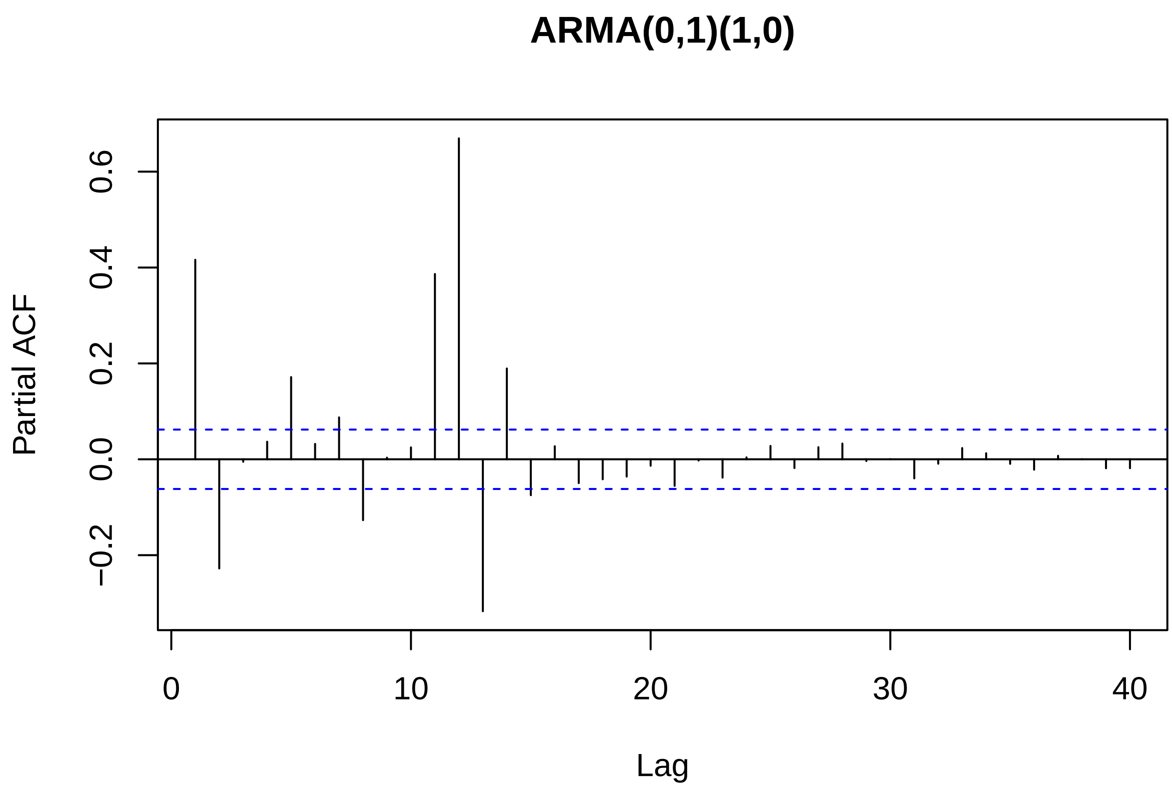
```r
plot(Acf(Model_0110, lag = 40), main = "ARMA(0,1)(1,0)")
```


Series Model_0110
ACF
0.0
0.2
0.4
0.6
0.8
0
10
20
30
40
Lag




```r
plot(Pacf(Model_0110, lag = 40), main = "ARMA(0,1)(1,0)")
```


Series Model_0110
Partial ACF
0.6
0.4
0.2
0.0
−0.2
0
10
20
30
40
Lag



Answer: There are multiplle spikes at the seasonal lag in the ACF plot and a single spike at the seasonal lag in the PACF. So, that clearly indicates a SAR process.