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

Assignment 6 - Due date 02/28/24

Jaimie Wargo

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.

```r
#Load/install required package here
library(ggplot2)
library(forecast)
```

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

```r
library(tseries)
library(sarima)
```

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

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

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

```r
library(here)
```

```
## here() starts at C:/Users/jaimi/OneDrive/Documents/Duke/Spring_2024/TSA_Sp24
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
```

```
## -- 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
```

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: AR(2) represents a second order autoregressive model. The ACF will show an exponential decay, and the PACF will have a significant peak at lags 1 and 2 to indicate that this is a second order model.

- MA(1)

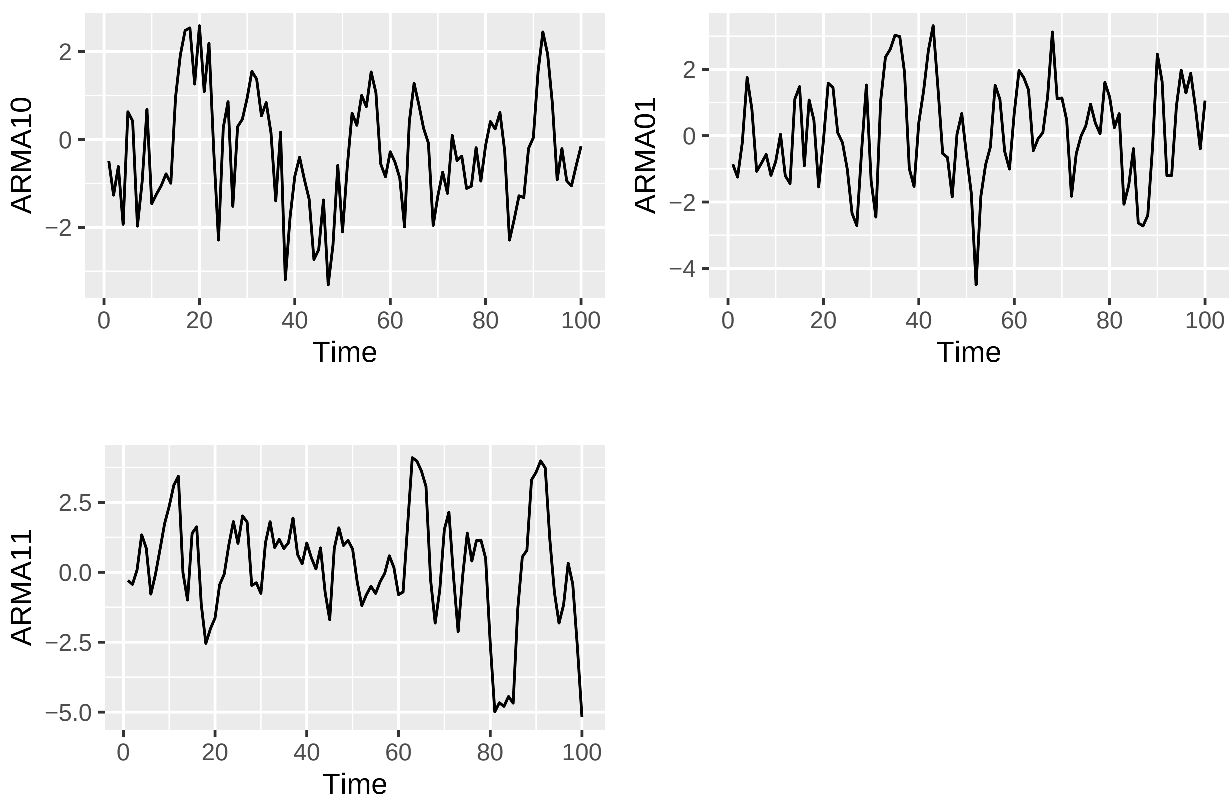
  Answer: MA(1) signifies a moving average plot with an order of 1. The ACF for this model would have a significant peak at lag 1, as this identifies the order of the model. The PACF plot would show a slow exponential decay as lags increase.

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.

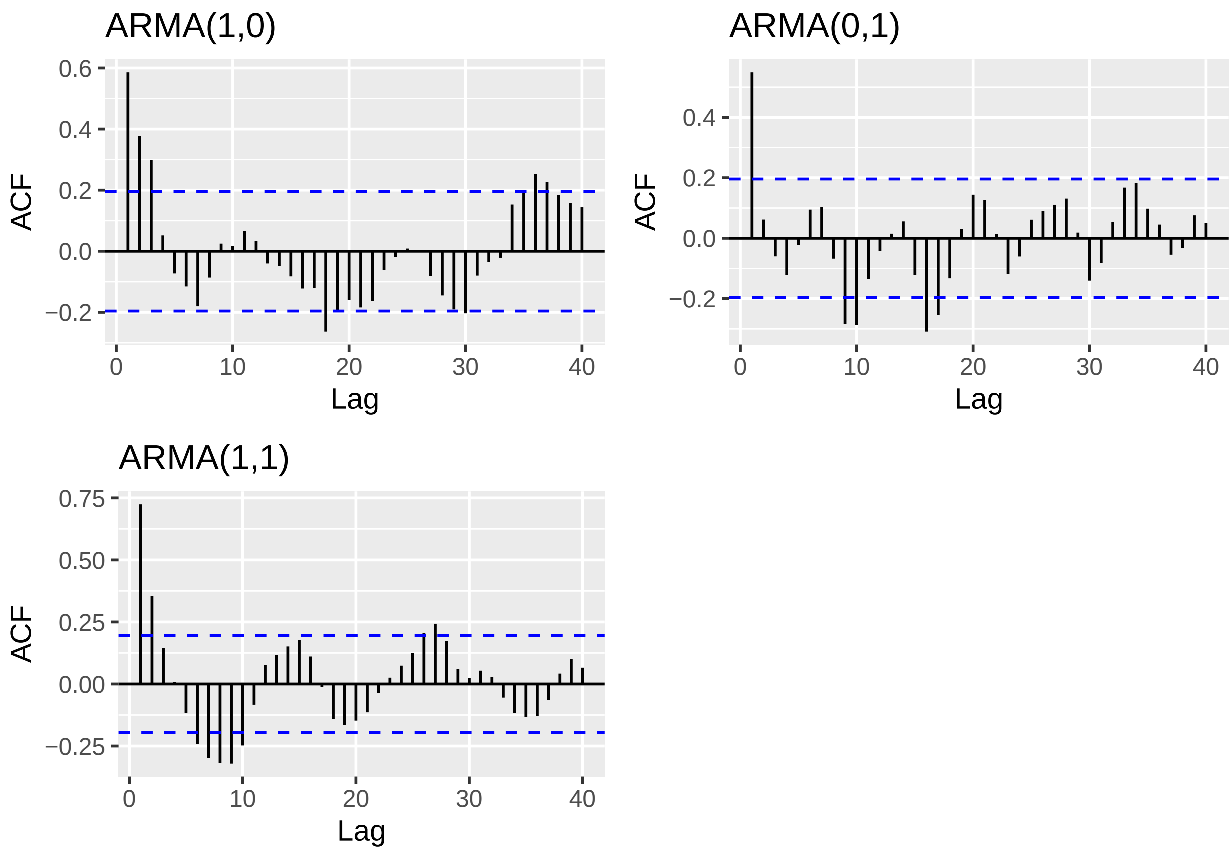
```
ARMA10 <- arima.sim(list(order=c(1,0,0), ar=0.6), n=100)
ARMA01 <- arima.sim(list(order=c(0,0,1), ma=0.9), n=100)
ARMA11 <- arima.sim(list(order=c(1,0,1), ar=0.6, ma=0.9), n=100)

plot_grid(autoplot(ARMA10), autoplot(ARMA01),autoplot(ARMA11))
```



(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(ARMA10, lag = 40, plot=FALSE),main = "ARMA(1,0)"),
autoplot(Acf(ARMA01, lag = 40, plot=FALSE),main = "ARMA(0,1)"),
autoplot(Acf(ARMA11, lag = 40, plot=FALSE),main = "ARMA(1,1)"))
```

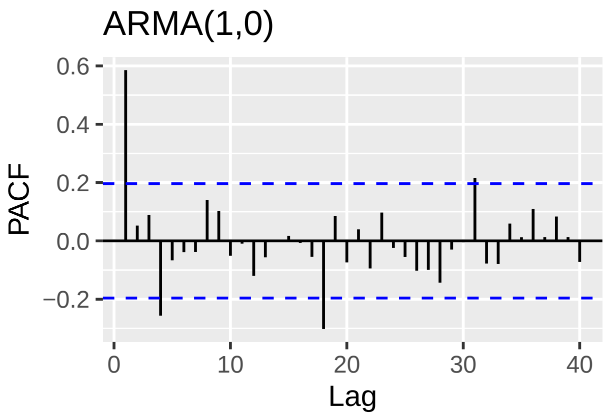
```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: `main`
## Ignoring unknown parameters: `main`
## Ignoring unknown parameters: `main`
```

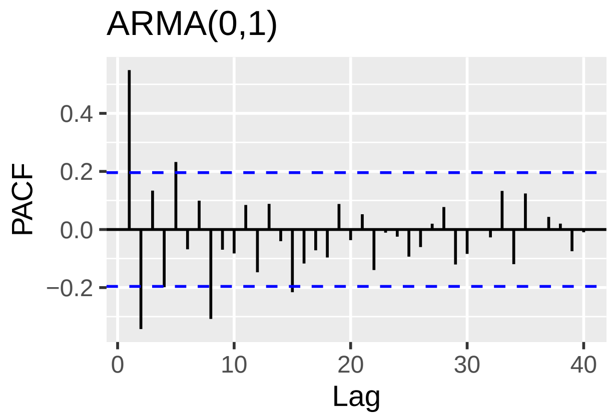


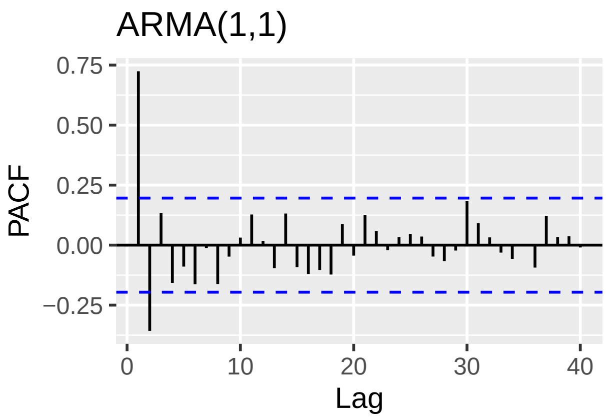
(c) Plot the sample PACF for each of these models in one window to facilitate comparison.

```r
plot_grid(autoplot(Pacf(ARMA10, lag = 40, plot=FALSE),main = "ARMA(1,0)"),
autoplot(Pacf(ARMA01, lag = 40, plot=FALSE),main = "ARMA(0,1)"),
autoplot(Pacf(ARMA11, 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.

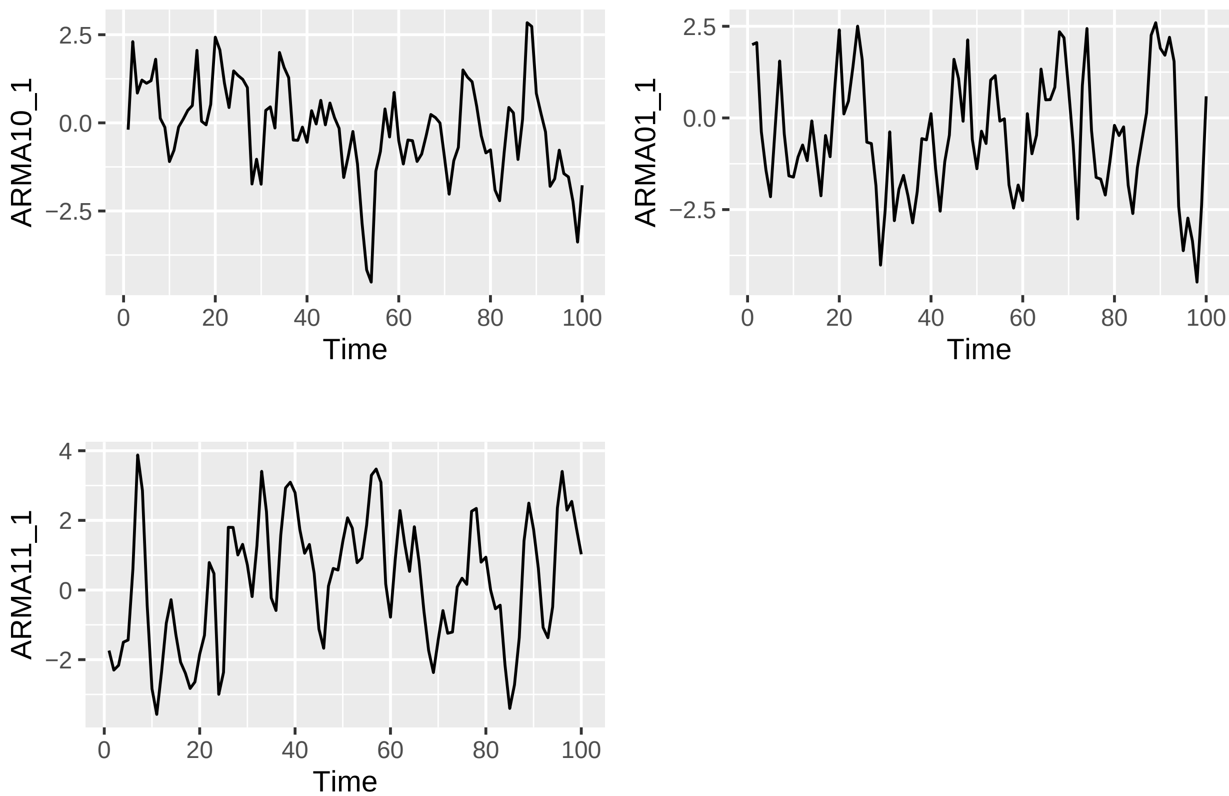
Answer: I believe so. We can see that for ARMA(0,1) there is a strong ACF peak at lag 1 with exponential decay of the PACF, and vice versa for the ARMA(1,0) model. It would be difficult for me to identify the ARMA(1,1) model as it contains both components.

(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: It matches for ARMA(1,0), but not ARMA(1,1). It should match because these plots give us information about the autoregressive component of a series.

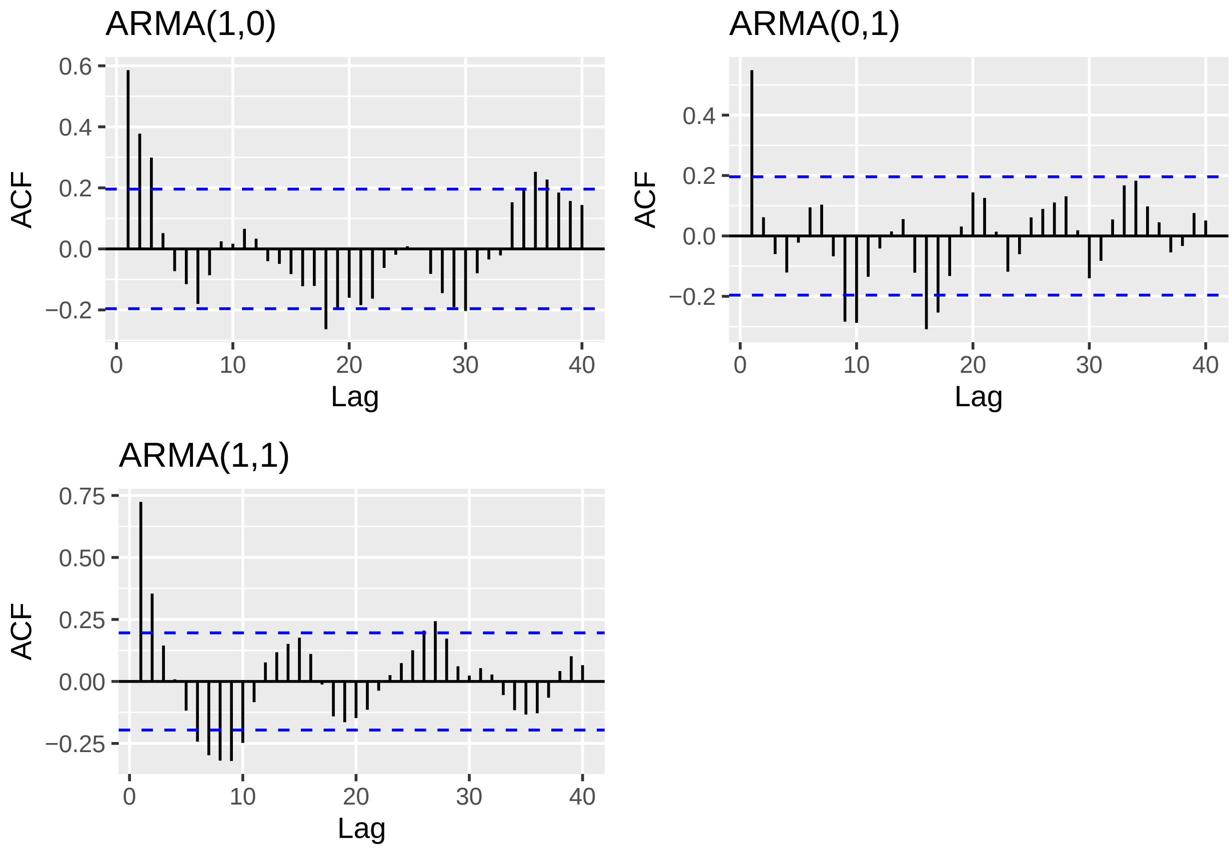
(f) Increase number of observations to $n = 1000$ and repeat parts (b)-(e).

```
ARMA10_1 <- arima.sim(list(order=c(1,0,0), ar=0.6), n=100)
ARMA01_1 <- arima.sim(list(order=c(0,0,1), ma=0.9), n=100)
ARMA11_1 <- arima.sim(list(order=c(1,0,1), ar=0.6, ma=0.9), n=100)

plot_grid(autoplot(ARMA10_1), autoplot(ARMA01_1),autoplot(ARMA11_1))
```



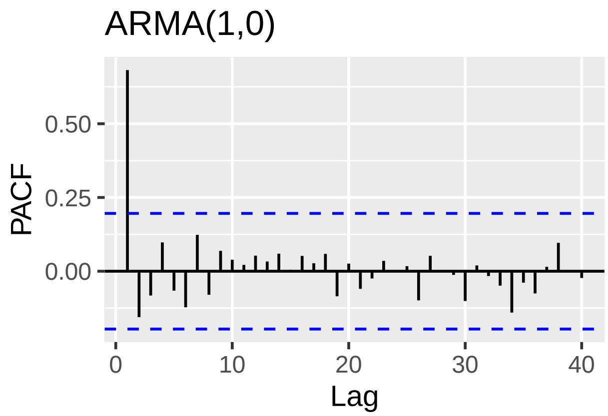

```r
plot_grid(autoplot(Acf(ARMA10, lag = 40, plot=FALSE),main = "ARMA(1,0)"),
autoplot(Acf(ARMA01, lag = 40, plot=FALSE),main = "ARMA(0,1)"),
autoplot(Acf(ARMA11, lag = 40, plot=FALSE),main = "ARMA(1,1)"))
```

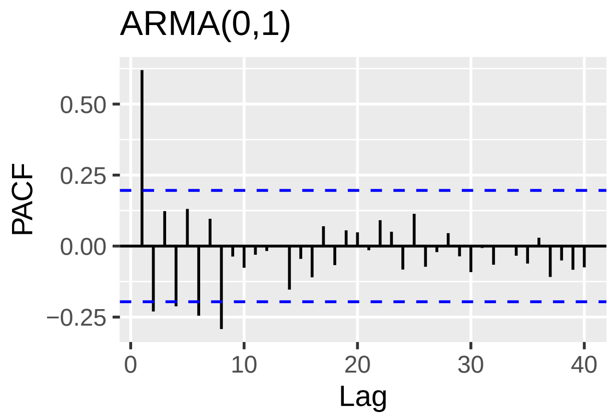
```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
```

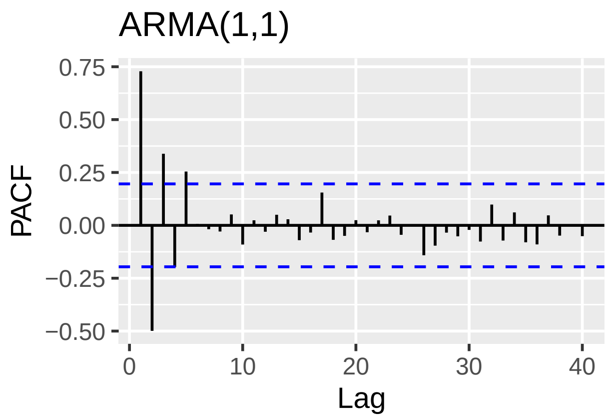



```
plot_grid(autoplot(Pacf(ARMA10_1, lag = 40, plot=FALSE),main = "ARMA(1,0)"),
autoplot(Pacf(ARMA01_1, lag = 40, plot=FALSE),main = "ARMA(0,1)"),
autoplot(Pacf(ARMA11_1, lag = 40, plot=FALSE),main = "ARMA(1,1)"))
```

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







I do still believe I would be able to tell the (0,1) and (1,0) models, but not the (1,1). Here, the (1,1) model best shows the autoregressive component, maybe meaning that the moving average component is more prominent here.

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.

ARIMA(1,1,1)(1,0,0)[12]

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

Parameters:

$\phi(t-1) = 0.7$

$\theta(t-1) = 0.1$

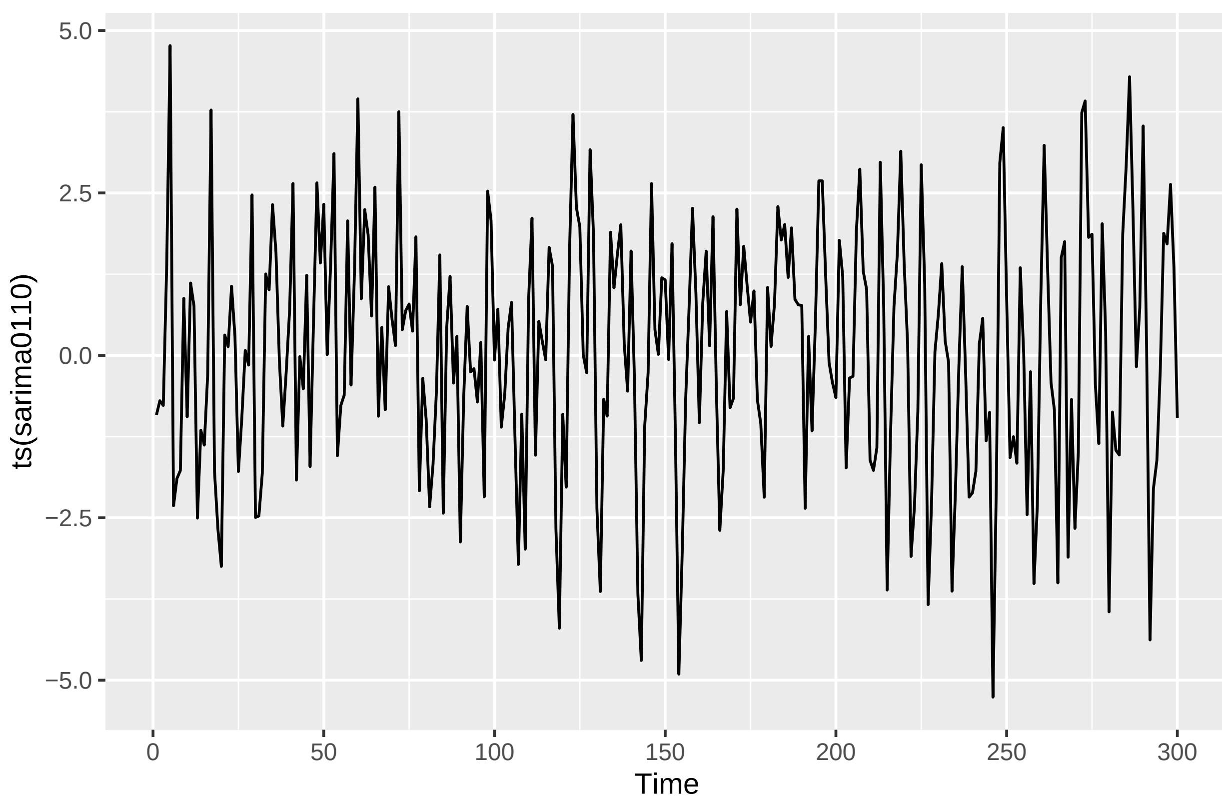
$\phi(t-12) = 0.25$

$\mu = 0$

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?

```
sarima0110 <- sim_sarima(n=300, model = list(ma=0.5, sar=0.8,
                                             iorder=0, siorder=0,
                                             nseasons=12))

autoplot(ts(sarima0110))
```



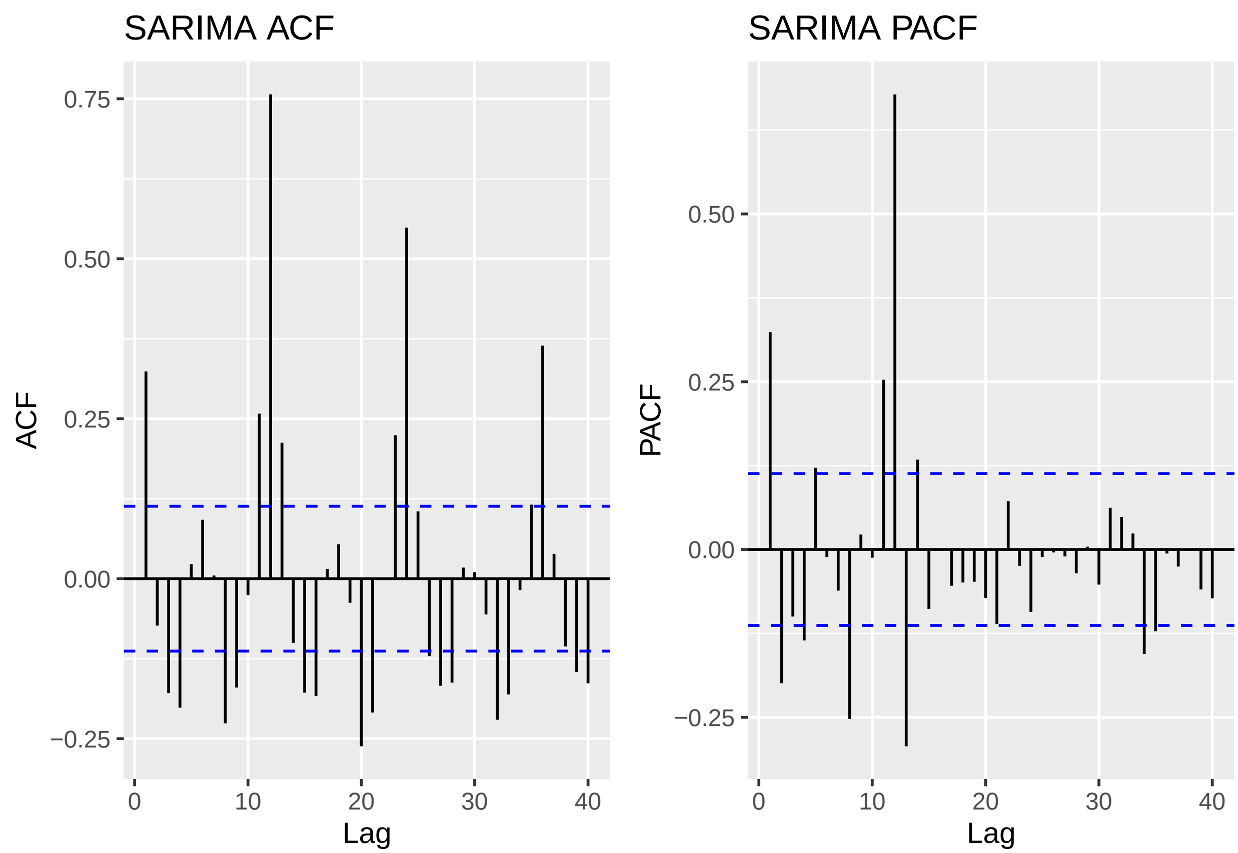
There appears to be seasonality in this series, as shown by the fluctuations.

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.

```
plot_grid(autoplot(Acf(ts(sarima0110), lag = 40, plot=FALSE),main = "SARIMA ACF"),
autoplot(Pacf(ts(sarima0110), lag = 40, plot=FALSE),main = "SARIMA PACF"))
```

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



You can see particular seasonality in these plots, however I would not be confident pulling out the order from either the ACF or PACF for the ar, sar, ma, or sma parameters. My explanation encapsulates why fairly well– there are simply too many parameters all mushed together into these plots that I would not be able to parse out.