ENV 790.30 - Time Series Analysis for Energy Data | Spring 2021

Assignment 6 - Due date 03/16/22

Abbhijith Hari Gopal

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github. And to do so you will need to fork our repository and link it to your RStudio.

Once you have the project open the first thing you will do is change "Student Name" on line 3 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. Rename the pdf file such that it includes your first and last name (e.g., "LuanaLima_TSA_A06_Sp22.Rmd"). Submit this pdf using Sakai.

Questions

This assignment has general questions about ARIMA Models.

Packages needed for this assignment: "forecast","tseries". Do not forget to load them before running your script, since they are NOT default packages.\

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

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

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

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(sarima) #for SARIMA needed in Q4
```

```
## Loading required package: stats4

##
## Attaching package: 'sarima'

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

> NOTE: For all the questions, since we are simulating models (being dynamic) and not working on a fixed data set, the explanations pertain to a fixed case. When I saved to continue my work later, I saw the TS plots, Acf and Pacf plots changed. Therefore, the explanations written might not reflect the values/plots you are seeing.

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) means that the model is an autoregressive model with an order = 2. Basically, what this means is that the present value would depend on its previous state (t-1) and its previous state (t-2). The order for an AR model is deduced from the PACF plot. There will be a slow decay in ACF and the PACF cuts off at lag 2.

MA(1)

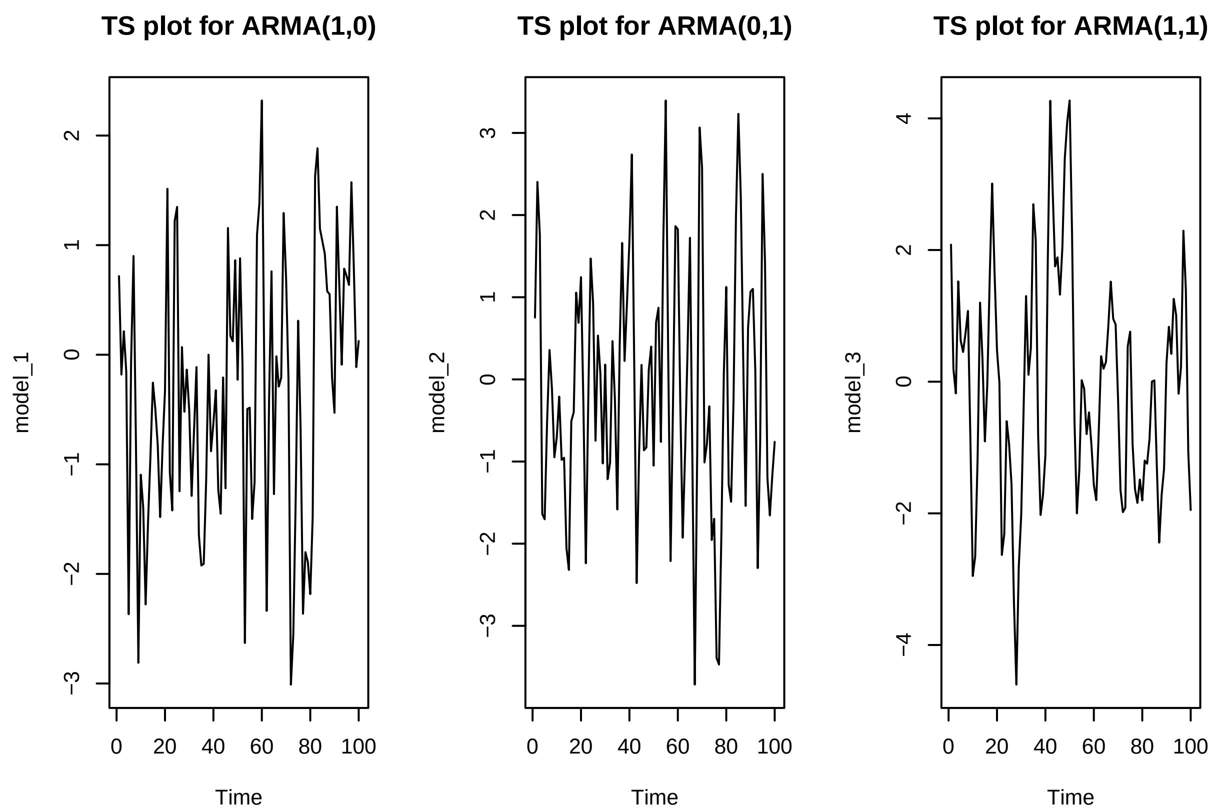
> Answer: A moving average term in a time series model is a past error (multiplied by a coefficient). The order for an MA model is deduced from the ACF plot. There will be a slow decay in PACF and the ACF cuts off at lag 1, for MA(1) model.

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). 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 R to generate $n = 100$ observations from each of these three models

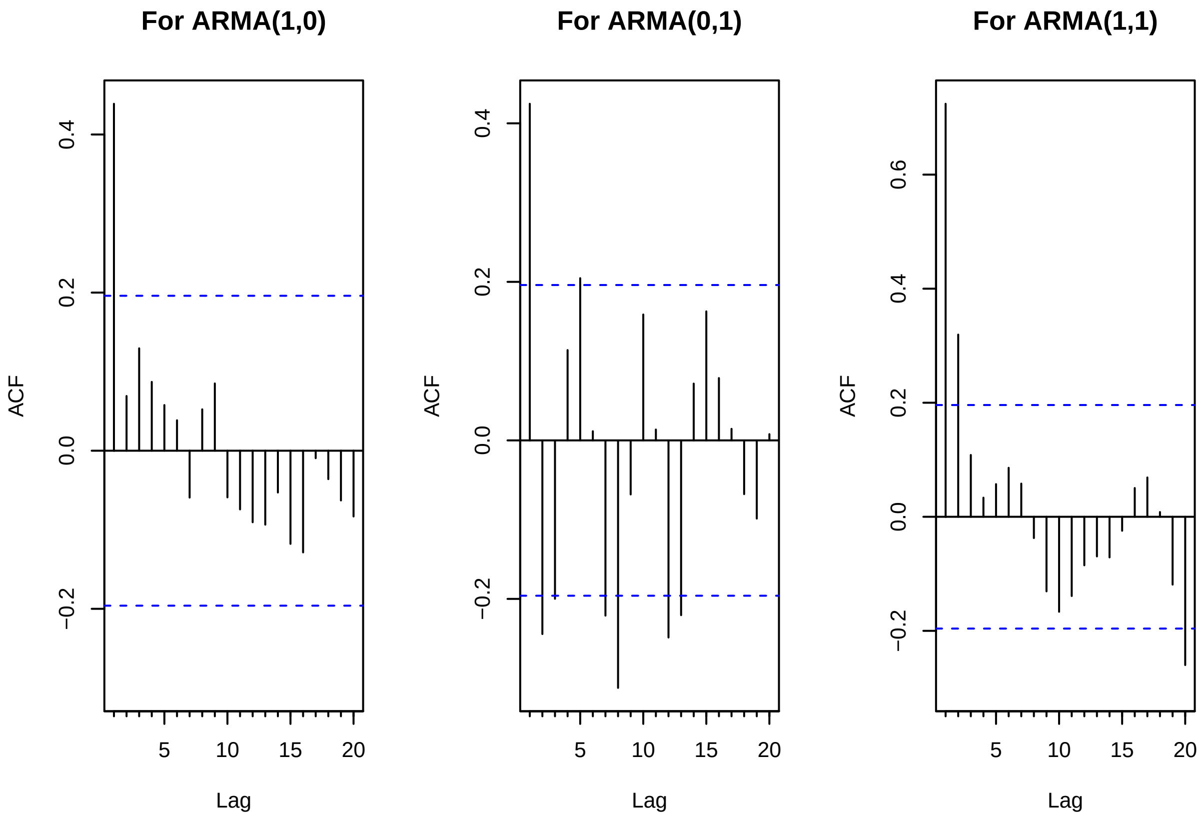
```
#ARMA(1,0), AR coefficient = 0.6
model_1 = arima.sim(model=list(ar=0.6), n=100)
#ARMA(0,1), MA coefficient = 0.9
model_2 = arima.sim(model=list(ma=0.9), n=100)
#ARMA(1,1), AR coefficient = 0.6 and the MA coefficient = 0.9
model_3 = arima.sim(model=list(ar=0.6, ma=0.9), n=100)
```

```r
#plots as a check
par(mfrow=c(1,3))
plot(model_1, main="TS plot for ARMA(1,0)")
plot(model_2, main="TS plot for ARMA(0,1)")
plot(model_3, main="TS plot for ARMA(1,1)")
```



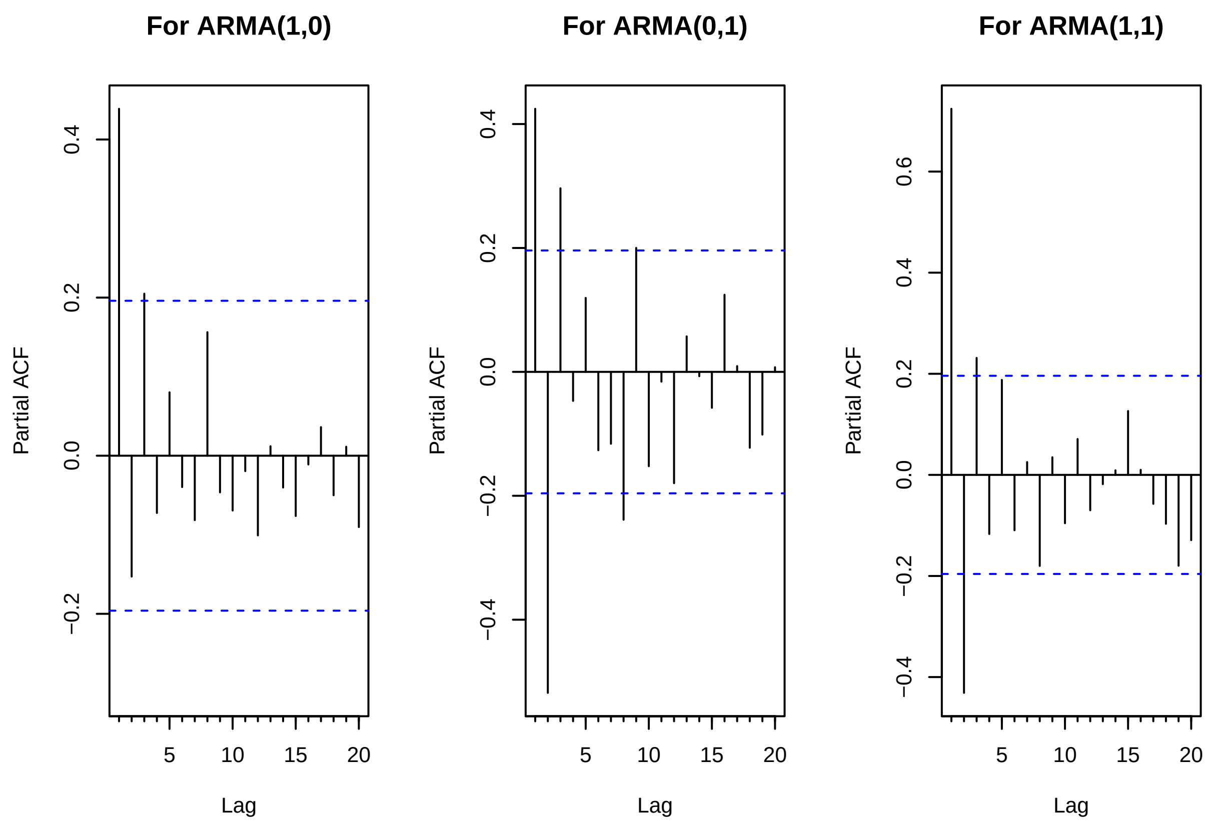
Plot the sample ACF for each of these models in one window to facilitate comparison (Hint: use command $par(mfrow = c(1, 3))$ that divides the plotting window in three columns).

```r
par(mfrow=c(1,3)) #divides the plotting window in three columns
acf_of_model_1 = Acf(model_1, type="correlation", plot=TRUE, main="For ARMA(1,0)")
acf_of_model_2 = Acf(model_2, type="correlation", plot=TRUE, main="For ARMA(0,1)")
acf_of_model_3 = Acf(model_3, type="correlation", plot=TRUE, main="For ARMA(1,1)")
```



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

```
par(mfrow=c(1,3)) #divides the plotting window in three columns
pacf_of_model_1 = Acf(model_1, type="partial", plot=TRUE, main="For ARMA(1,0)")
pacf_of_model_2 = Acf(model_2, type="partial", plot=TRUE, main="For ARMA(0,1)")
pacf_of_model_3 = Acf(model_3, type="partial", plot=TRUE, main="For ARMA(1,1)")
```



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 identify them correctly? Explain your answer.

> Answer: Yes for model 1 and 2. For model 1, we can clearly see that AR model is appropriate because there slow decay in the ACF plot and a clear cut off at lag 1 in the PACF plot i.e, p = 1. For model 2, we can clearly see that MA model is appropriate because the PACF plot has a slow decay and the ACF plot has a cut off at lag 1 i.e, q = 1. For model 3, it is not quite clear in first glance as to which model is appropriate because the Acf and Pacf plots have AR and MA properties.

Compare the ACF and PACF values R computed with the theoretical values you provided for the coefficients. Do they match? Explain your answer.

```
#For model 1
print(acf_of_model_1)
```

```
##
## Autocorrelations of series 'model_1', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.439  0.069  0.130  0.087  0.058  0.039 -0.059  0.052  0.085 -0.059
##     11     12     13     14     15     16     17     18     19     20
## -0.074 -0.090 -0.094 -0.053 -0.118 -0.129 -0.009 -0.036 -0.063 -0.083
```

```r
print(pacf_of_model_1)
```

```
##
## Partial autocorrelations of series 'model_1', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.439 -0.153  0.205 -0.072  0.080 -0.040 -0.081  0.156 -0.046 -0.069 -0.020
##     12     13     14     15     16     17     18     19     20
## -0.101  0.012 -0.040 -0.076 -0.011  0.036 -0.050  0.011 -0.090
```

```r
#For model 2
print(acf_of_model_2)
```

```
##
## Autocorrelations of series 'model_2', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.425 -0.244 -0.200  0.114  0.205  0.012 -0.221 -0.312 -0.068  0.159
##     11     12     13     14     15     16     17     18     19     20
##  0.014 -0.249 -0.221  0.072  0.163  0.079  0.015 -0.068 -0.099  0.008
```

```r
print(pacf_of_model_2)
```

```
##
## Partial autocorrelations of series 'model_2', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.425 -0.518  0.296 -0.047  0.119 -0.126 -0.116 -0.239  0.200 -0.152 -0.016
##     12     13     14     15     16     17     18     19     20
## -0.180  0.057 -0.007 -0.058  0.125  0.009 -0.122 -0.101  0.008
```

```r
#For model 3
print(acf_of_model_3)
```

```
##
## Autocorrelations of series 'model_3', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.724  0.319  0.108  0.034  0.057  0.086  0.058 -0.037 -0.131 -0.167
##     11     12     13     14     15     16     17     18     19     20
## -0.139 -0.085 -0.069 -0.071 -0.024  0.050  0.069  0.008 -0.119 -0.260
```

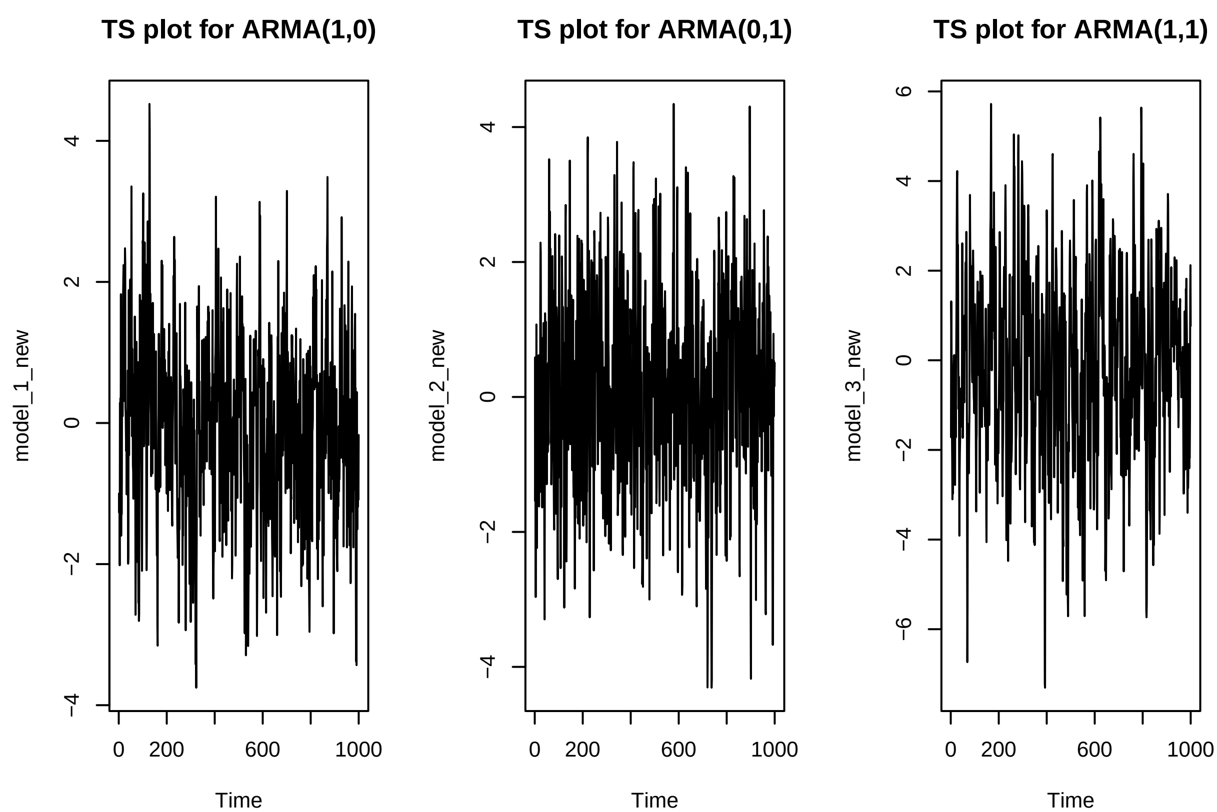
```r
print(pacf_of_model_3)
```

```
##
## Partial autocorrelations of series 'model_3', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.724 -0.431  0.231 -0.117  0.188 -0.110  0.025 -0.180  0.035 -0.096  0.071
##     12     13     14     15     16     17     18     19     20
## -0.070 -0.018  0.009  0.126  0.010 -0.057 -0.097 -0.180 -0.129
```

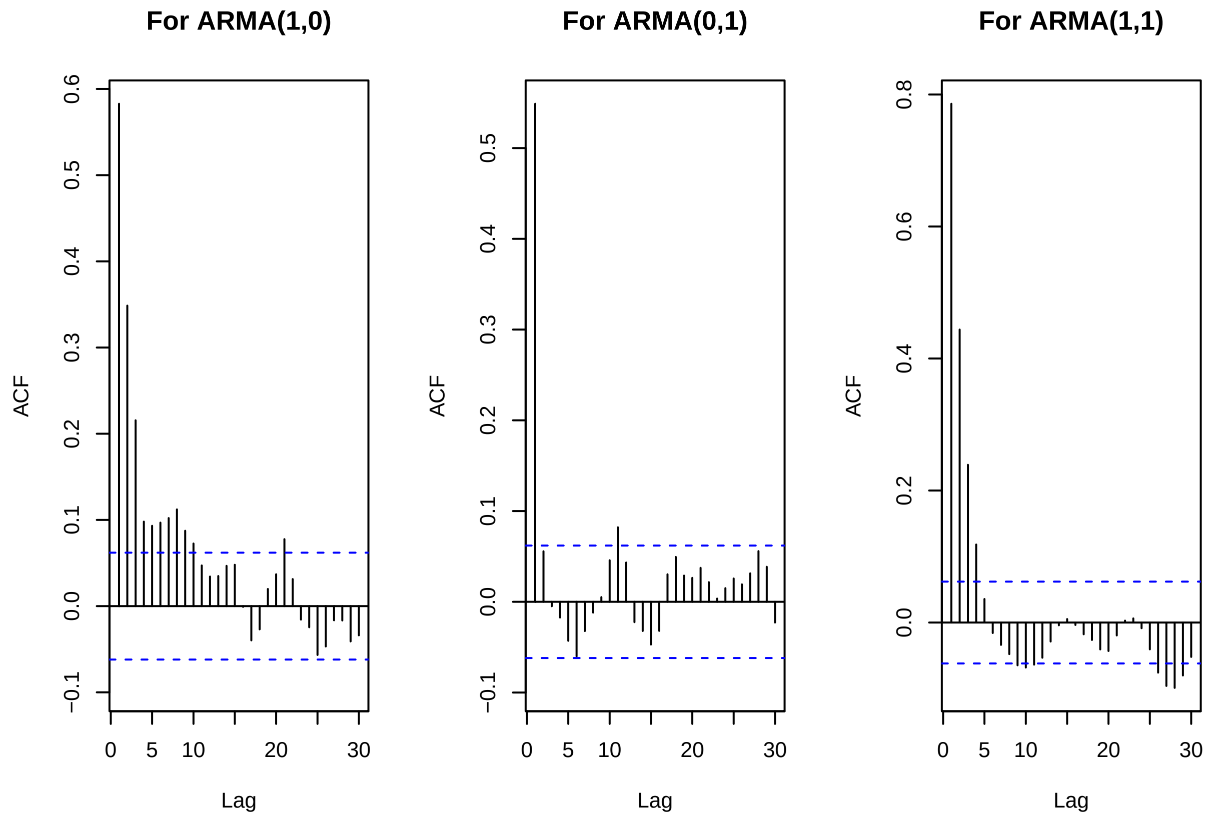
Answer: We can see the Pacf values for model 1 and 3, at lag 1, are 0.583 and 0.785 respectively. This is close to the value we specify (0.6). Model 2 is an MA model. Therefore, we can't deduce the coefficient values from Acf or Pacf plot.

Increase number of observations to $n = 1000$ and repeat parts (a)-(d).

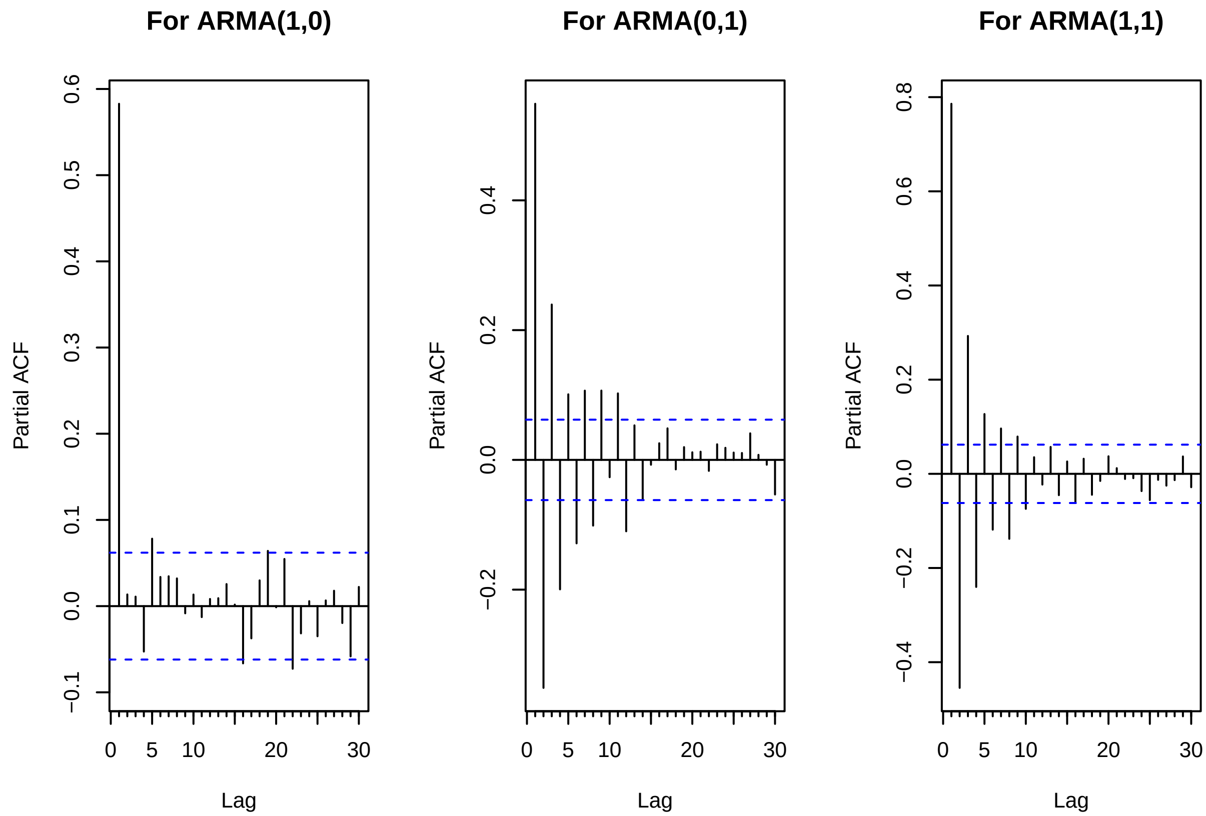
```
#ARMA(1,0), AR coefficient = 0.6
model_1_new = arima.sim(model=list(ar=0.6), n=1000)
#ARMA(0,1), MA coefficient = 0.9
model_2_new = arima.sim(model=list(ma=0.9), n=1000)
#ARMA(1,1), AR coefficient = 0.6 and the MA coefficient = 0.9
model_3_new = arima.sim(model=list(ar=0.6, ma=0.9), n=1000)

#plots as a check
par(mfrow=c(1,3))
plot(model_1_new, main="TS plot for ARMA(1,0)")
plot(model_2_new, main="TS plot for ARMA(0,1)")
plot(model_3_new, main="TS plot for ARMA(1,1)")
```




```
#ACF and PACF Plots
par(mfrow=c(1,3)) #divides the plotting window in three columns
acf_of_model_1_new = Acf(model_1_new, type="correlation", plot=TRUE, main="For ARMA(1,0)")
acf_of_model_2_new = Acf(model_2_new, type="correlation", plot=TRUE, main="For ARMA(0,1)")
acf_of_model_3_new = Acf(model_3_new, type="correlation", plot=TRUE, main="For ARMA(1,1)")
```




```r
pacf_of_model_1_new = Acf(model_1_new, type="partial", plot=TRUE, main="For ARMA(1,0)")
pacf_of_model_2_new = Acf(model_2_new, type="partial", plot=TRUE, main="For ARMA(0,1)")
pacf_of_model_3_new = Acf(model_3_new, type="partial", plot=TRUE, main="For ARMA(1,1)")
```



> Discussion: On increasing the number of observations (n=1000), the following conclusions can be drawn. For model 1, we can clearly see that AR model is appropriate because there slow decay in the ACF plot and a clear cut off at lag 1 in the PACF plot i.e, p = 1. For model 2, we can clearly see that MA model is appropriate because the PACF plot has a slow decay and the ACF plot has a cut off at lag 1 i.e, q = 1. In this scenario, in contrast to the previous case, for model 3, we can clearly see it is an ARMA(1,1) model. This is because, both ACF and PACF plots demonstrate a gradual decreasing pattern.

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 ARIMA model, we can clearly see that y(t) depends on y(t-1) and y(t-12). Therefore, p = 1 since 12 represents the seasonal lag. P, the seasonal AR component is also equal to 1. Also, y(t) depends on a(t) and a(t-1) - the MA terms. Therefore, q = 1 but Q = 0 because we don't have an a(t-12) term. We don't have enough info to deduce d or D. The notion is ARIMA(1,d,1)(1,D,0)_12

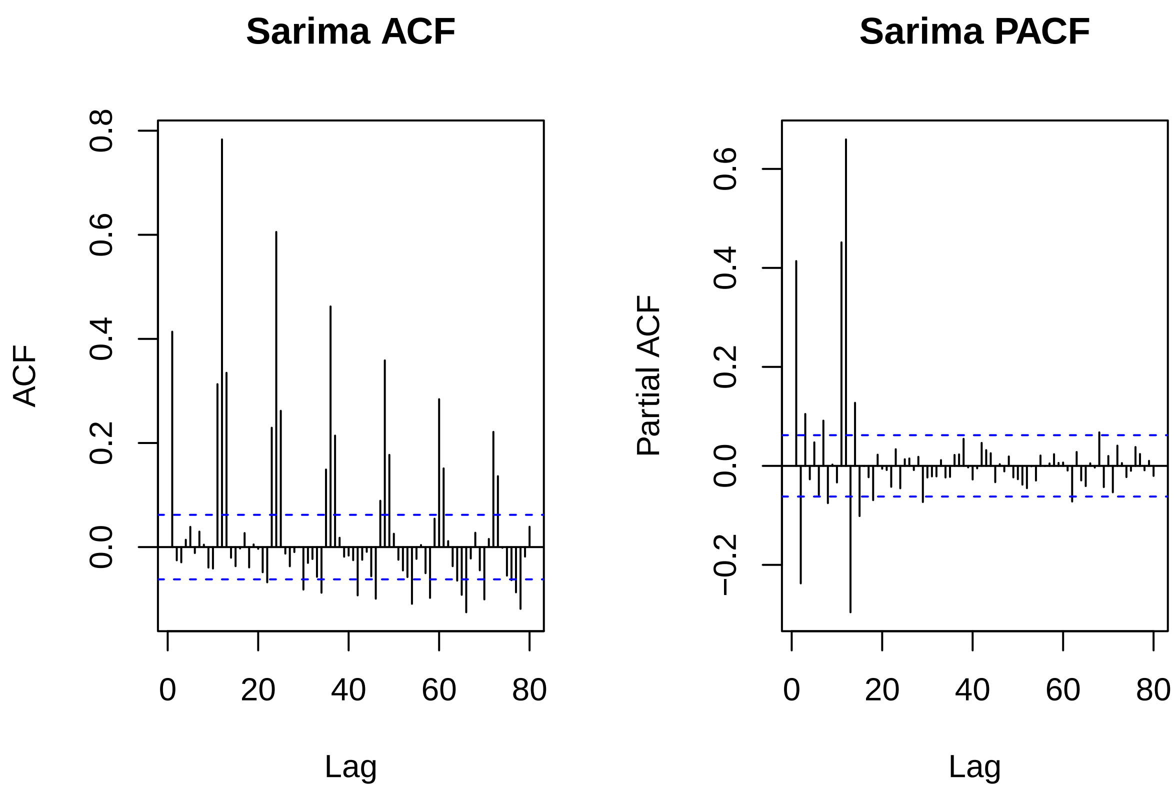
Also from the equation what are the values of the parameters, i.e., model coefficients.

> Answer: Comparing the given equation to the standard equation, we can deduce that the value of phi_1 = 0.7, phi_12 (seasonal part) = -0.25, and theta = -0.1.

Q4

Plot the ACF and PACF of a seasonal ARIMA$(0,1) \times (1,0)_{12}$ model with $\phi = 0.8$ and $\theta = 0.5$ using R. 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 ACF and PACF for the simulated data. 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.

```
#SARIMA model
sarima_model = sim_sarima(model=list(ma=0.5,sar=0.8, nseasons=12), n=1000)

#ACF and PACF plots
par(mfrow=c(1,2))
sarima_acf = Acf(sarima_model, type="correlation", plot=TRUE, main = "Sarima ACF", lag.max=80)
sarima_pacf = Acf(sarima_model, type="partial", plot=TRUE,  main = "Sarima PACF", lag.max=80)
```



>Acf and Pacf plots discussion: For the seasonal part, we must focus on lags that are multiples of 12 i.e, lag 12, lag 24, lag 36, etc. We can clearly observe that Acf has a clear seasonal pattern - has multiple peaks whereas the Pacf plot has only 1 peak. Therefore, we have a seasonal AR model in this scenario. We look at Pacf to get value of P, which is equal to 1 in this case. For the non-seasonal part, we must observe the Acf and Pacf plots for lags < 12 since it represents the non-seasonal component.The Acf has a cut off at lag 1 and the Pacf has a slow decay. Therefore, we are looking at an MA model with q = 1. Discussion of the coeffecients of the model are written below.

```r
print(Arima(sarima_model, order = c(0,0,1), seasonal = list(order=c(1,0,0), period = 12)))
```

```
## Series: sarima_model
## ARIMA(0,0,1)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##          ma1    sar1     mean
##       0.5022  0.7794  -0.1334
## s.e.  0.0267  0.0193   0.2070
##
## sigma^2 = 1.005:  log likelihood = -1425.63
## AIC=2859.26   AICc=2859.3   BIC=2878.9
```

coefficient discussion: We had 0.8 as the i/p value for seasonal AR component. From the Pacf plot, we can find the value to be close to 0.7. Also, from the coefficients we just obtained, we got a SAR coefficient as 0.8116 (NOTE: This is subjected to change on every simulation and is not static). Therefore, we can conclude that the values in simulation and and what we gave as i/p are pretty close. We had 0.5 as MA model coefficient. From the Acf plot, we can find the value to be close to 0.4. Also, from the coefficients we just obtained, we got a MA coefficient as 0.5573 (NOTE: This is subjected to change on every simulation and is not static).Therefore, we can conclude that the values in simulation and and what we gave as i/p are pretty close.