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

Assignment 6 - Due date 03/16/22

Aasha Reddy

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 here
library(lubridate)
```

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

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

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

```
## Warning: package 'forecast' was built under R version 4.1.2

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

```
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.1.2
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.1.2
```

```r
library(outliers)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```r
library(smooth)
```

```
## Warning: package 'smooth' was built under R version 4.1.2
## Loading required package: greybox
## Warning: package 'greybox' was built under R version 4.1.2
## Package "greybox", v1.0.4 loaded.
##
## Attaching package: 'greybox'
## The following object is masked from 'package:tidyr':
##
##     spread
## The following object is masked from 'package:forecast':
##
##     forecast
## The following object is masked from 'package:lubridate':
##
##     hm
## This is package "smooth", v3.1.5
```

```r
library(sarima)
```

```
## Warning: package 'sarima' was built under R version 4.1.3
## Loading required package: stats4
##
## Attaching package: 'sarima'
## The following object is masked from 'package:stats':
##
##     spectrum
```

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):

This is an AR model with order 2. ACF will decay exponentially over time, while PACF will help us to identify the order of the model as 2, I would thus expect a cutoff at lag 2 in the PACF for this model. AR processes have a relatively long memory. Often if the stationary series has positive autocorrelation at lag 1, AR terms work best.

MA(1):

This is an MA model with order 1. ACF will help us identify the order of the model while the PACF will decay exponentially. Thus, I would expect a cutoff at lag 1 in the ACF for this series. MA processes have shorter memory than AR models. If there is negative autocorrelation at lag 1, often MA terms work best.

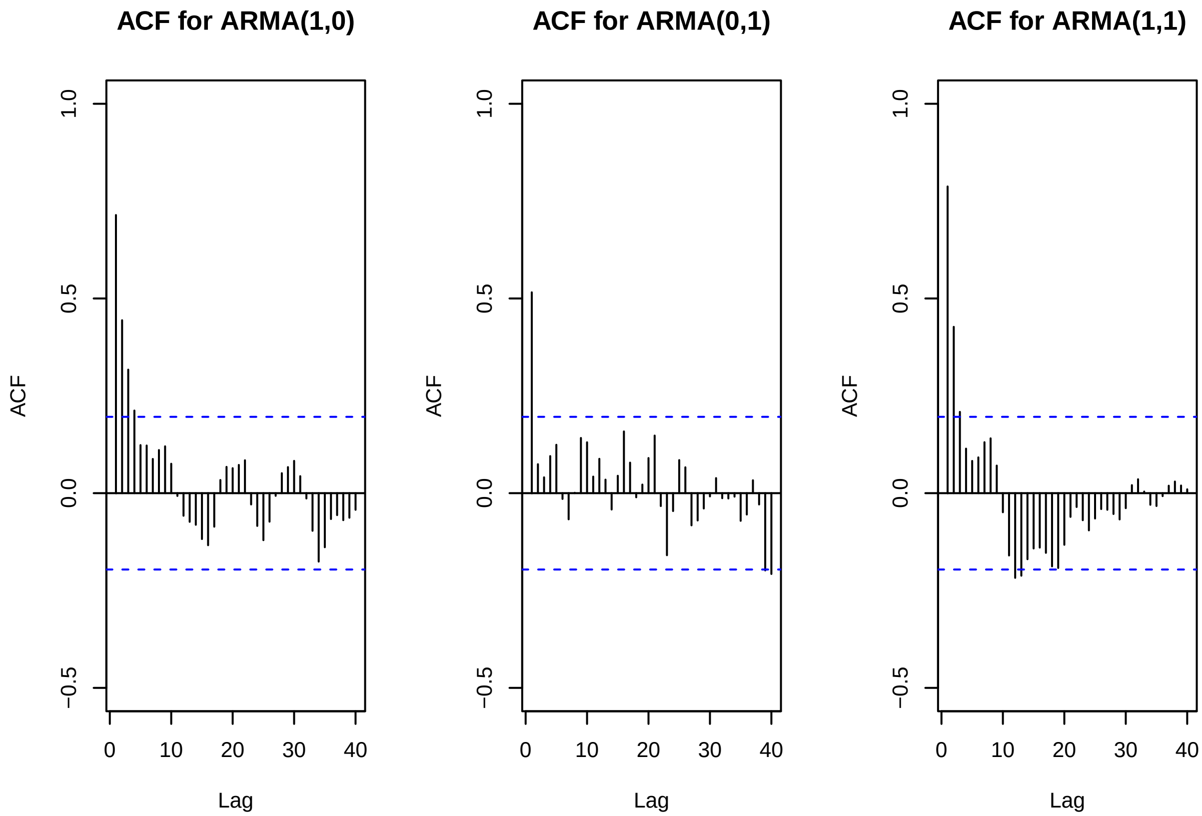
Q2

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

```r
set.seed(99)
# ARMA(1,0)
arma_10 <- arima.sim(n = 100, list(ar = 0.6, order = c(1,0,0)))

# ARMA(0,1)
arma_01 <- arima.sim(n = 100, list(ma = 0.9, order = c(0,0,1)))

# ARMA(1,1
arma_11 <- arima.sim(n = 100, list(ar = 0.6, ma = 0.9, order = c(1,0,1)))
```

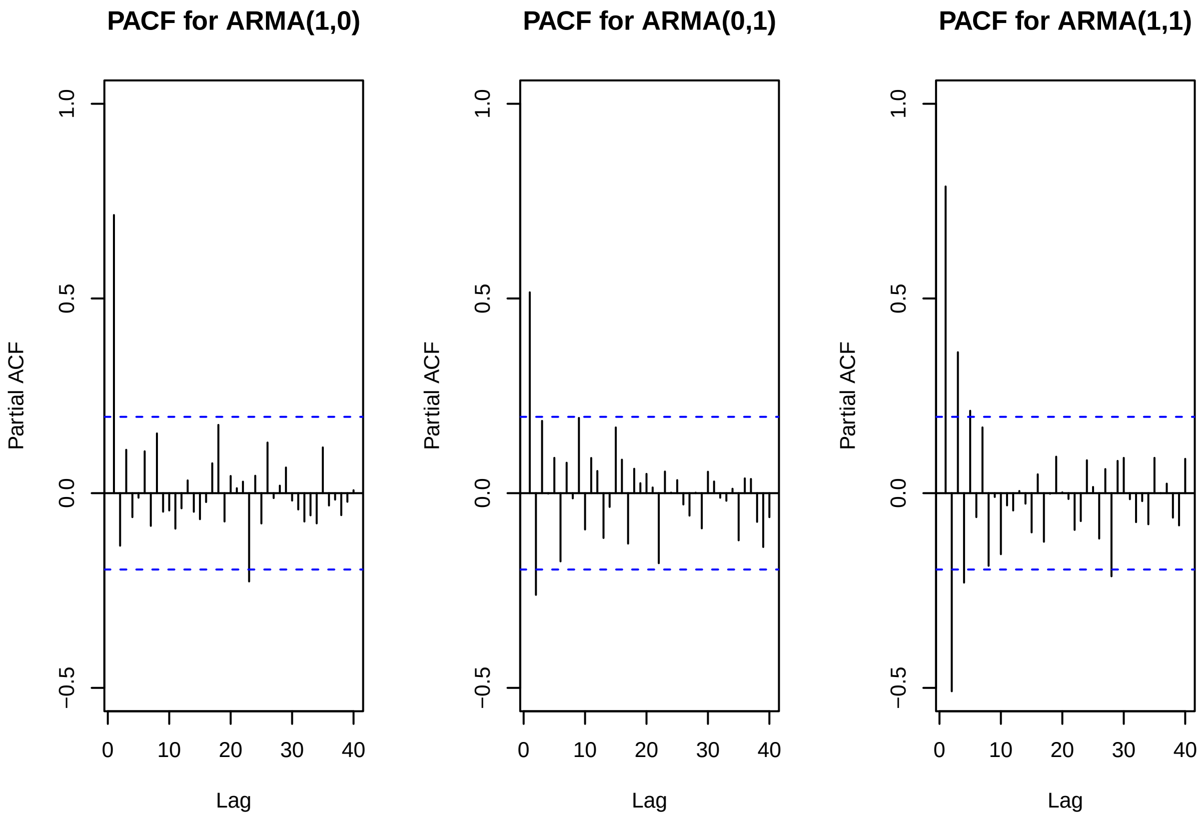
Plot the sample ACF for each of these models in one window to facilitate comparison (Hint: use command $par(mfrow = c(1, 3))$ that divides the plotting window in three columns).

```r
par(mfrow=c(1,3))

Acf(arma_10,lag.max=40,
    main="ACF for ARMA(1,0)",
    ylim = c(-0.5, 1))

Acf(arma_01,lag.max=40,
    main="ACF for ARMA(0,1)",
    ylim = c(-0.5, 1))

Acf(arma_11,lag.max=40,
    main="ACF for ARMA(1,1)",
    ylim = c(-0.5, 1))
```



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

```
par(mfrow=c(1,3))

Pacf(arma_10,lag.max=40,
    main="PACF for ARMA(1,0)",
    ylim = c(-0.5, 1))

Pacf(arma_01,lag.max=40,
    main="PACF for ARMA(0,1)",
    ylim = c(-0.5, 1))

Pacf(arma_11,lag.max=40,
    main="PACF for ARMA(1,1)",
    ylim = c(-0.5, 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.

ACF: For ARMA(1,0), this refers to an AR model of order 1. For an AR model, we should see that the ACF decays exponentially over time which we can see happening slightly. We also see that initial slow decay in the ACF plot for ARMA(1,0) which I would expect.

For ARMA(0,1) this refers to an MA model with an order of 1. For the ACF plot, this will help us identify which order. This ACF plot would help me identify correctly because it looks like there is actually a cutoff at lag 1. It does not have negative correlation at lag 1.

For ARMA(1,1), this is an ARMA model with p = 1 and q = 1. We can see the slow exponential decay in the ACF plot, it does not cut off however but is gradually dying out as I would expect. The shape is similar to the ARMA(1,0) process also as I would expect and superimposed with the ARMA(0,1) ACF. The initial coefficients depend on the MA order and later decay dictated by the AR part.

PACF: For ARMA(1,0), this refers to an AR(1) model the PACF plot should help us identify the order, and we do see a cutoff at lag 1, which would help me identify the order 1 for this model.

For ARMA(0,1), this refers to an MA(1) model and the PACF plot should decay exponentially. Here, I do see some of that slow decay as I would expect so would be able to identify this as an ARMA(0,1) model from this PACF plot.

For ARMA(1,1) the PACF plot does look like a superposition of the AR and MA properties. The PACF initial values are dependent on the AR followed by a decay due to the MA part.

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

```r
Acf(arma_10, plot = F)
```

```
##
## Autocorrelations of series 'arma_10', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.714  0.444  0.317  0.212  0.123  0.122  0.088  0.111  0.120  0.076
##     11     12     13     14     15     16     17     18     19     20
## -0.007 -0.058 -0.074 -0.081 -0.118 -0.134 -0.086  0.034  0.068  0.064
```

```r
Pacf(arma_10, plot = F)
```

```
##
## Partial autocorrelations of series 'arma_10', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.714 -0.135  0.111 -0.062 -0.012  0.108 -0.084  0.154 -0.048 -0.044 -0.091
##     12     13     14     15     16     17     18     19     20
## -0.039  0.033 -0.048 -0.067 -0.023  0.077  0.175 -0.073  0.044
```

For the ARMA(1,0) model, this is AR(1) with the coefficient being $\phi = 0.6$, this should also be close to the lag 1 autocorrelation. In practice the lag 1 autocorrelation is 0.714 as we see above, which does not really match the 0.6 that I would expect. In practice, the sample ACF will rarely fit a perfect theoretical pattern.

```r
Acf(arma_01, plot = F)
```

```
##
## Autocorrelations of series 'arma_01', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.516  0.074  0.041  0.095  0.124 -0.015 -0.067  0.001  0.142  0.131
##     11     12     13     14     15     16     17     18     19     20
##  0.043  0.088  0.035 -0.042  0.045  0.159  0.078 -0.011  0.022  0.090
```

```r
Pacf(arma_01, plot = F)
```

```
##
## Partial autocorrelations of series 'arma_01', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.516 -0.261  0.186 -0.001  0.091 -0.175  0.078 -0.013  0.193 -0.093  0.090
##     12     13     14     15     16     17     18     19     20
##  0.057 -0.115 -0.035  0.169  0.086 -0.129  0.063  0.026  0.050
```

```r
0.9/(1 + 0.9)
```

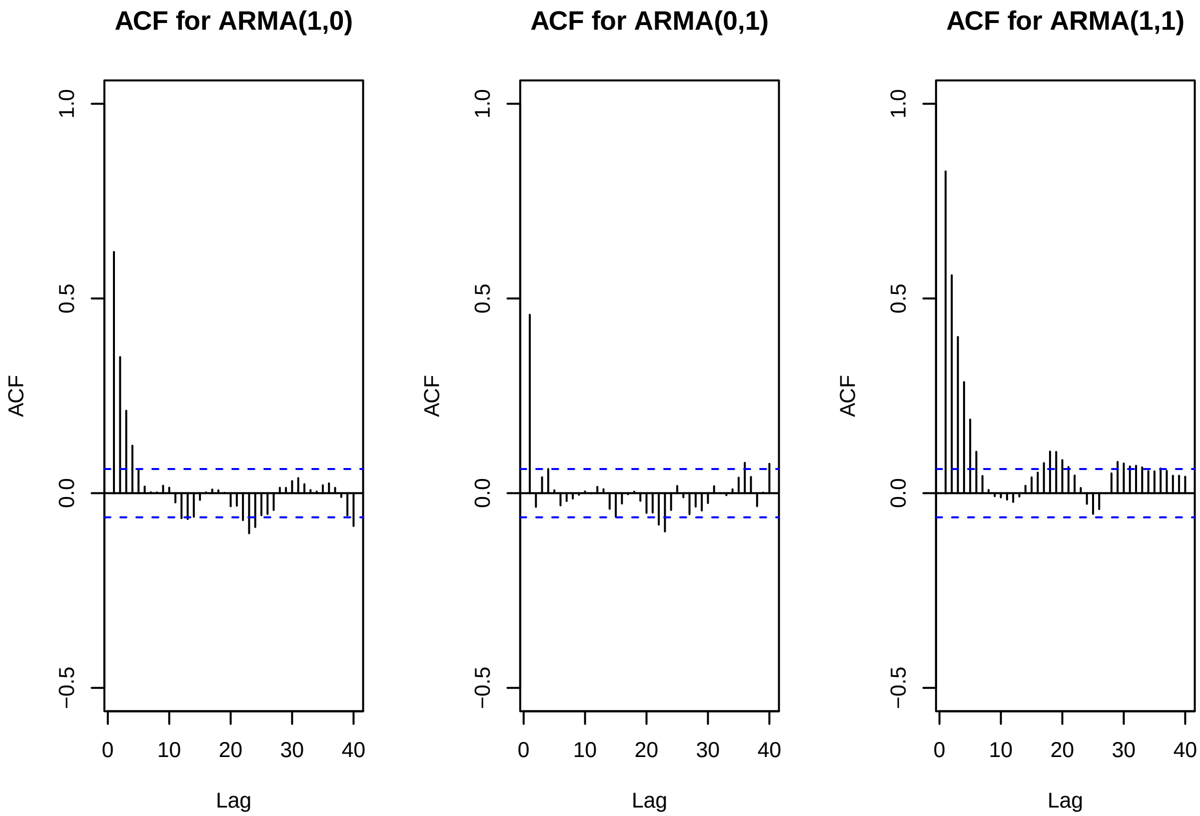
```
## [1] 0.4736842
```

For ARMA(0,1) this is the same as an MA(1) model with the coefficient being $\theta = 0.9$. Based on this coefficient of 0.9, I would expect the lag 1 autocorrelation to be $0.9/(1 + 0.9) = 0.4737$. From the below, we see the lag 1 autocorrelation from the simulated sample is actually 0.516 which is relatively close.

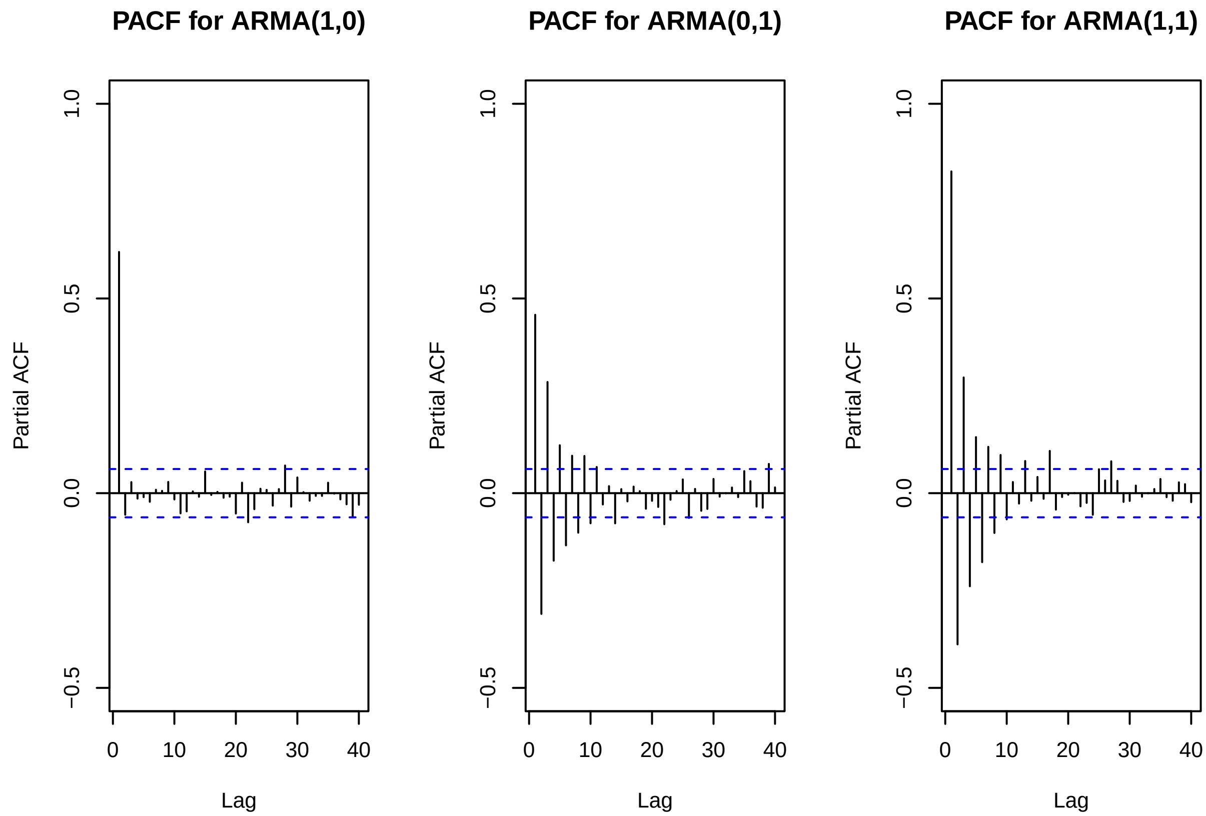
Increase number of observations to $n = 1000$ and repeat parts (a)-(d).

```
set.seed(99)
# Create series for n = 1000
# ARMA(1,0)
arma_10 <- arima.sim(n = 1000, list(ar = 0.6, order = c(1,0,0)))

# ARMA(0,1)
arma_01 <- arima.sim(n = 1000, list(ma = 0.9, order = c(0,0,1)))

# ARMA(1,1
arma_11 <- arima.sim(n = 1000, list(ar = 0.6, ma = 0.9, order = c(1,0,1)))
```

```
# plot ACFs
par(mfrow=c(1,3))

Acf(arma_10,lag.max=40,
    main="ACF for ARMA(1,0)",
    ylim = c(-0.5, 1))

Acf(arma_01,lag.max=40,
    main="ACF for ARMA(0,1)",
    ylim = c(-0.5, 1))

Acf(arma_11,lag.max=40,
    main="ACF for ARMA(1,1)",
    ylim = c(-0.5, 1))
```




```
# plot PACFS
par(mfrow=c(1,3))

Pacf(arma_10,lag.max=40,
    main="PACF for ARMA(1,0)",
    ylim = c(-0.5, 1))

Pacf(arma_01,lag.max=40,
    main="PACF for ARMA(0,1)",
    ylim = c(-0.5, 1))

Pacf(arma_11,lag.max=40,
    main="PACF for ARMA(1,1)",
    ylim = c(-0.5, 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.

ACF: For ARMA(1,0), this refers to an AR model of order 1. For an AR model, we should see that the ACF decays exponentially over time which we can see. We also see the same initial slow decay in the ACF plot for ARMA(1,0) which I would expect.

For ARMA(0,1) this refers to an MA model with an order of 1. For the ACF plot, this will help us identify which order. This ACF plot would help me identify correctly because it looks like there is actually a cutoff at lag 1.

For ARMA(1,1), this is an ARMA model with p = 1 and q = 1. We can see the slow exponential decay in the ACF plot, it does not cut off however but is gradually dying out as I would expect. The shape is similar to the ARMA(1,0) process also as I would expect and superimposed with the ARMA(0,1) ACF. The initial coefficients depend on the MA order and later decay dictated by the AR part.

PACF: For ARMA(1,0), this refers to an AR(1) model the PACF plot should help us identify the order, and we do see a cutoff at lag 1, which would help me identify the order 1 for this model.

For ARMA(0,1), this refers to an MA(1) model and the PACF plot should decay exponentially. Here, I do see some the exponential decay so I would be able to identify this as an ARMA(0,1) from the PACF plot.

For ARMA(1,1) the PACF plot does look like a superposition of the AR and MA properties. The PACF initial values are dependent on the AR followed by a decay due to the MA part.

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

```r
Acf(arma_10, plot = F)
```

```
##
## Autocorrelations of series 'arma_10', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.619  0.350  0.212  0.122  0.062  0.017  0.002  0.002  0.019  0.014
##     11     12     13     14     15     16     17     18     19     20     21
## -0.024 -0.064 -0.066 -0.061 -0.017  0.002  0.009  0.007  0.001 -0.034 -0.032
##     22     23     24     25     26     27     28     29     30
## -0.069 -0.103 -0.087 -0.057 -0.053 -0.043  0.014  0.014  0.031
```

```r
Pacf(arma_10, plot = F)
```

```
##
## Partial autocorrelations of series 'arma_10', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.619 -0.055  0.028 -0.014 -0.010 -0.022  0.009  0.006  0.029 -0.016 -0.052
##     12     13     14     15     16     17     18     19     20     21     22
## -0.047  0.005 -0.009  0.056 -0.005  0.003 -0.012 -0.009 -0.052  0.027 -0.075
##     23     24     25     26     27     28     29     30
## -0.041  0.011  0.008 -0.032  0.011  0.071 -0.035  0.040
```

For the ARMA(1,0) model, this is AR(1) with the coefficient being $\phi = 0.6$, this should also be close to the lag 1 autocorrelation. In practice the lag 1 autocorrelation is 0.619 as we see above. This is a lot closer to the 0.6 number I would expect than when we only had n = 100. This makes sense because now we have n = 1000 so the lag 1 acf should more accurately reflect the coefficient.

```r
Acf(arma_01, plot = F)
```

```
##
## Autocorrelations of series 'arma_01', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.458 -0.035  0.041  0.062  0.007 -0.031 -0.020 -0.014 -0.004  0.005
##     11     12     13     14     15     16     17     18     19     20     21
## -0.001  0.016  0.010 -0.040 -0.061 -0.027 -0.003  0.004 -0.020 -0.051 -0.050
##     22     23     24     25     26     27     28     29     30
## -0.081 -0.098 -0.043  0.018 -0.011 -0.054 -0.034 -0.044 -0.025
```

```r
Pacf(arma_01, plot = F)
```

```
##
## Partial autocorrelations of series 'arma_01', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.458 -0.310  0.286 -0.173  0.123 -0.134  0.096 -0.102  0.096 -0.077  0.067
##     12     13     14     15     16     17     18     19     20     21     22
## -0.029  0.018 -0.077  0.010 -0.021  0.017  0.005 -0.040 -0.020 -0.035 -0.080
##     23     24     25     26     27     28     29     30
## -0.017  0.006  0.035 -0.064  0.011 -0.045 -0.040  0.037
```

```r
0.9/(1 + 0.9)
```

```
## [1] 0.4736842
```

For ARMA(0,1) this is the same as an MA(1) model with the coefficient being $\theta = 0.9$. Based on this coefficient of 0.9, I would expect the lag 1 autocorrelation to be 0.9/(1 + 0.9) = 0.4737. From the below, we see the lag 1 autocorrelation from the simulated sample is actually 0.458 which is also relatively close. This is similar in closeness to the expected value as when we had n = 100, though I would have expected it to be closer for n = 1000.

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.

SARIMA(p = 0.7, d = 0, q = 0.1)(P = 1, D = 1, Q = 0)_[s = 12]

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

$\phi_1 = 0.7$

$\phi_{12} = 0.25$

$\theta = 0.1$

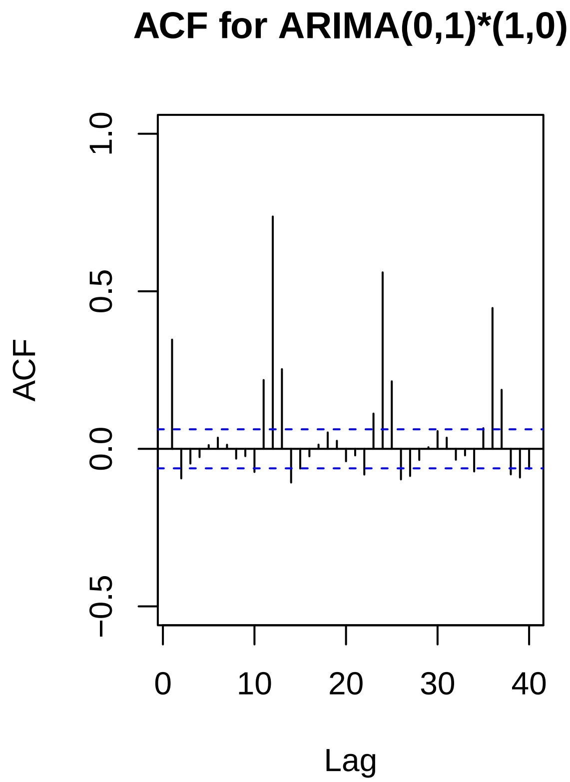
Q4

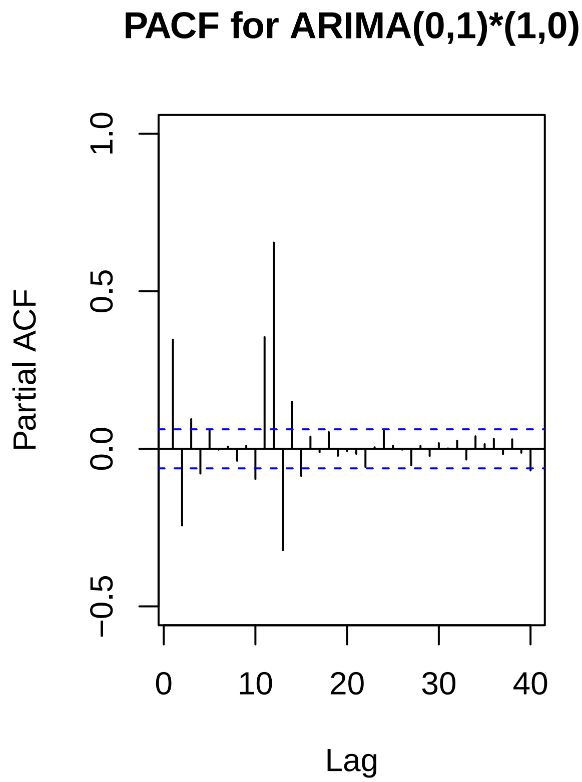
Plot the ACF and PACF of a seasonal ARIMA$(0, 1) \times (1, 0)_{12}$ model with $\phi = 0.8$ and $\theta = 0.5$ using R. The 12 after the bracket tells you that $s = 12$, i.e., the seasonal lag is 12, suggesting monthly data whose behavior is repeated every 12 months. You can generate as many observations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

This is the same as a SARIMA(0,0,1)*(1,0,0)_[s = 12]

```r
set.seed(99)
arima_01_10_12 <- sim_sarima(n = 1000, model = list(sar = 0.8, ma = 0.5, nseasons = 12))
```

```r
par(mfrow=c(1,2))

# plot ACFs
Acf(arima_01_10_12,lag.max=40,
    main="ACF for ARIMA(0,1)*(1,0)",
    ylim = c(-0.5, 1))

# plot PACFS
Pacf(arima_01_10_12,lag.max=40,
    main="PACF for ARIMA(0,1)*(1,0)",
    ylim = c(-0.5, 1))
```





Yes, the plots are well-representing of the model simulated. This model has an order 1 MA non-seasonal component, so I would expect ACF to help us identify the order of the model while the PACF will decay exponentially. Thus, I would expect a cutoff at lag 1 in the ACF for this series, which I do see in the ACF plot above. Thus, I would be able to identify the order of the non-seasonal component from the plot.

This model also has a seasonal AR term of order 1. For an AR term, the PACF will help us to identify the order of the model, and for a seasonal AR term of order 1, I would expect a cutoff at lag 12, which I do see in the PACF. Thus, I would be able to identify the order of the seasonal component from the plot.