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

Assignment 6 - Due date 02/28/24

Emma Kaufman

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(forecast)
```

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

```
library(tseries)
library(ggplot2)
library(Kendall)
library(lubridate)
```

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

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

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.3     v stringr 1.5.0
## v forcats 1.0.0     v tibble  3.2.1
## v purrr   1.0.2     v tidyr   1.3.0
## v readr   2.1.4

## -- 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(ggthemes)
#install.packages('sarima')
library(sarima)
```

```
## Loading required package: stats4
##
## Attaching package: 'sarima'
##
## The following object is masked from 'package:stats':
##
##     spectrum
```

```r
library(dplyr)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:ggthemes':
##
##     theme_map
##
## 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: ACF: Whether or not the data are autoregressive (significant values at early lags and decay exponentially with time). PACF: The order of the model (number of terms) is determined by the number of significant lags in PACF. An AR(2) model will have two signficnat lags in the PACF.

- MA(1)

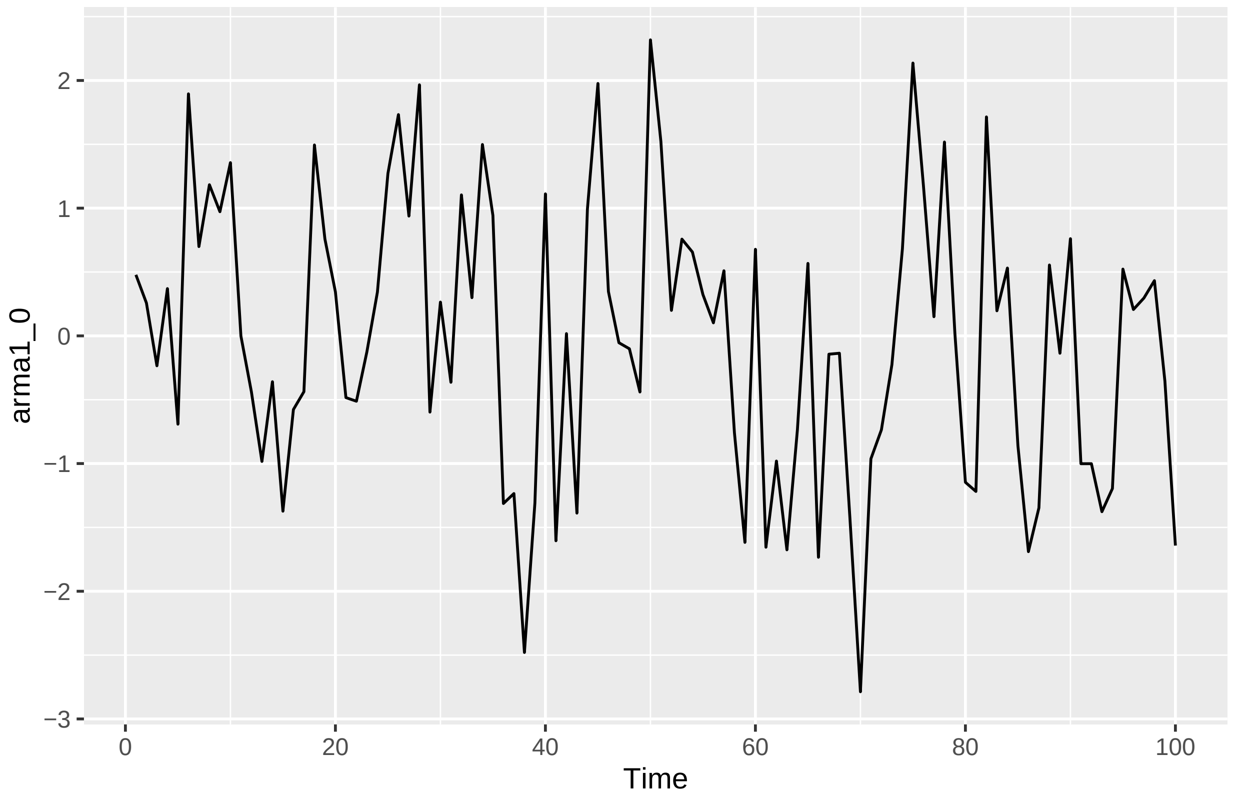
  Answer: ACF: Gives the order of the model. An MA(1) will show one significant lag PACF: Exponential decay in lags, "short memory" process that is less dependent upon the past.

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.

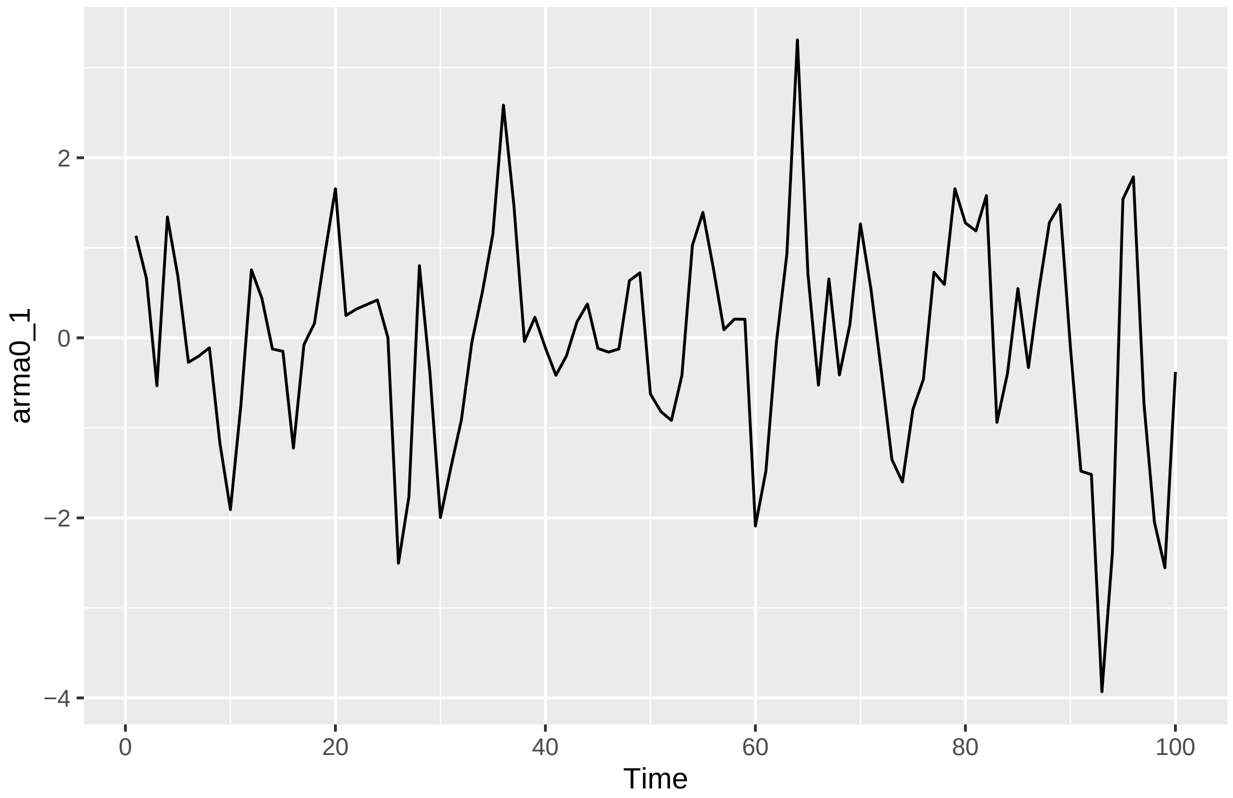
```
set.seed(100)
n = 100
phi <- 0.6 #AR coefficient
theta <- 0.9 #MA coefficient
p <- 1 #order of AR component
q <- 0 #order of MA component

arma1_0 <- arima.sim(model = list(ar = (phi*p), ma = (theta*q)), n)
autoplot(arma1_0)
```



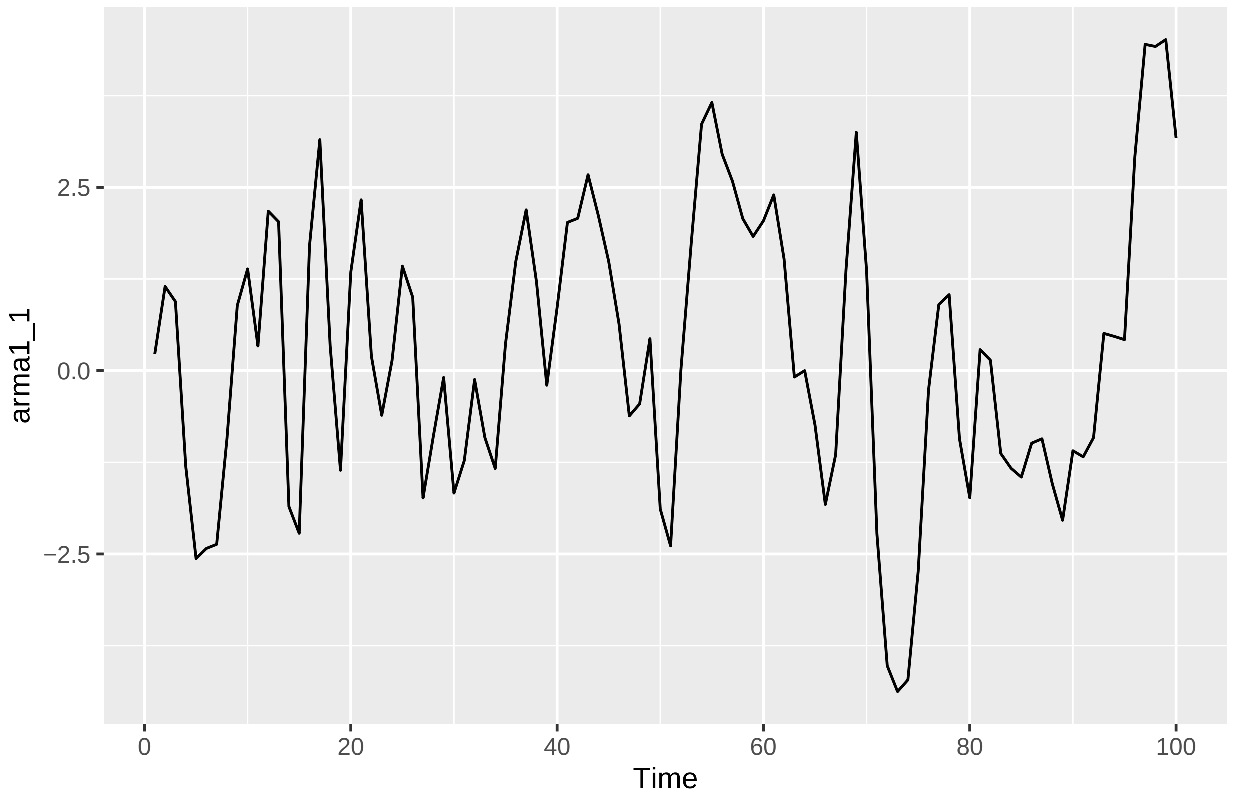
```
p <- 0 #order of AR component
q <- 1 #order of MA component

arma0_1 <- arima.sim(model = list(ar = (phi*p), ma = (theta*q)), n)
```

```
## Warning in min(Mod(polyroot(c(1, -model$ar)))): no non-missing arguments to
## min; returning Inf
```

```
autoplot(arma0_1)
```



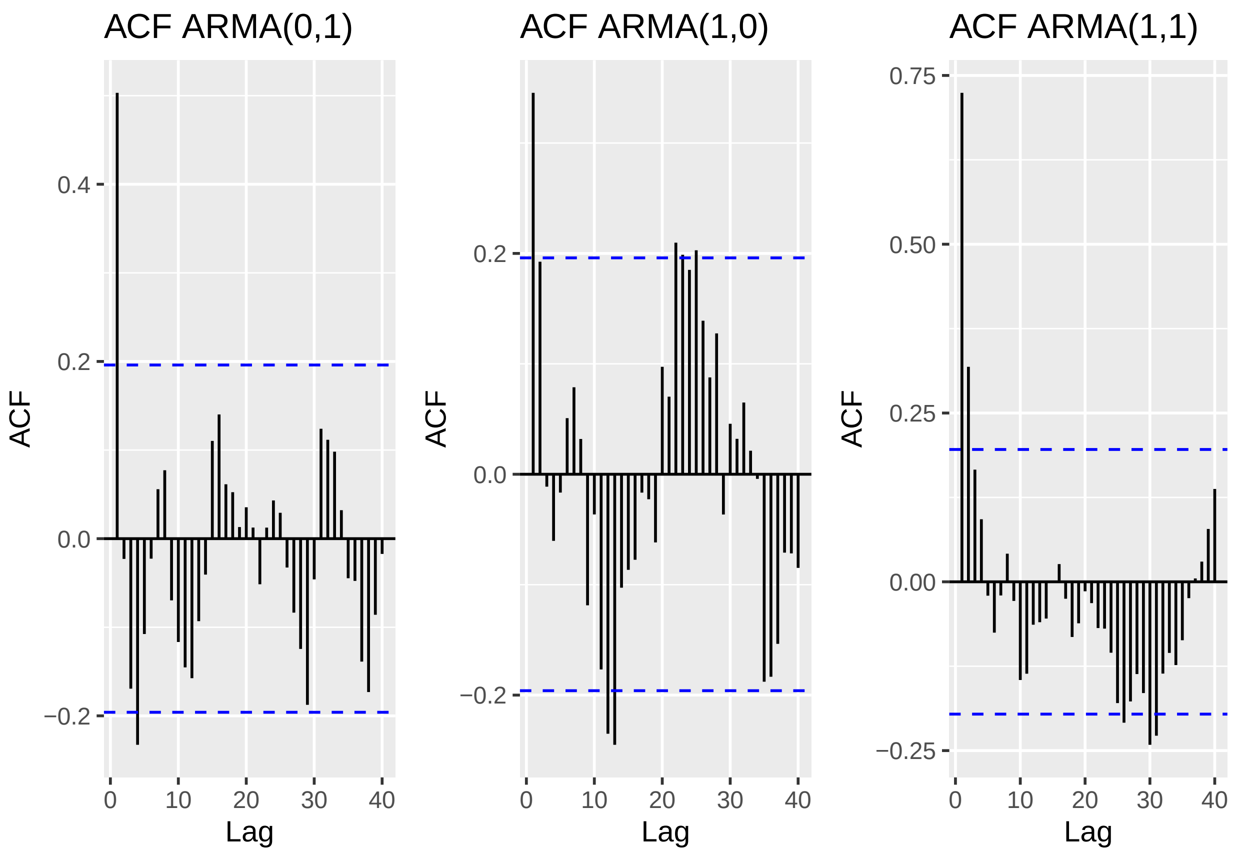
```
p <- 1 #order of AR component
q <- 1 #order of MA component

arma1_1 <- arima.sim(model = list(ar = (phi*p), ma = (theta*q)), n)
autoplot(arma1_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(arma0_1, lag = 40, plot=FALSE),
                main = "ACF ARMA(0,1)"),
   autoplot(Acf(arma1_0, lag = 40, plot=FALSE),
                main = "ACF ARMA(1,0)"),
   autoplot(Acf(arma1_1, lag = 40, plot=FALSE),
                main = "ACF ARMA(1,1)"),
  ncol=3
  )
```

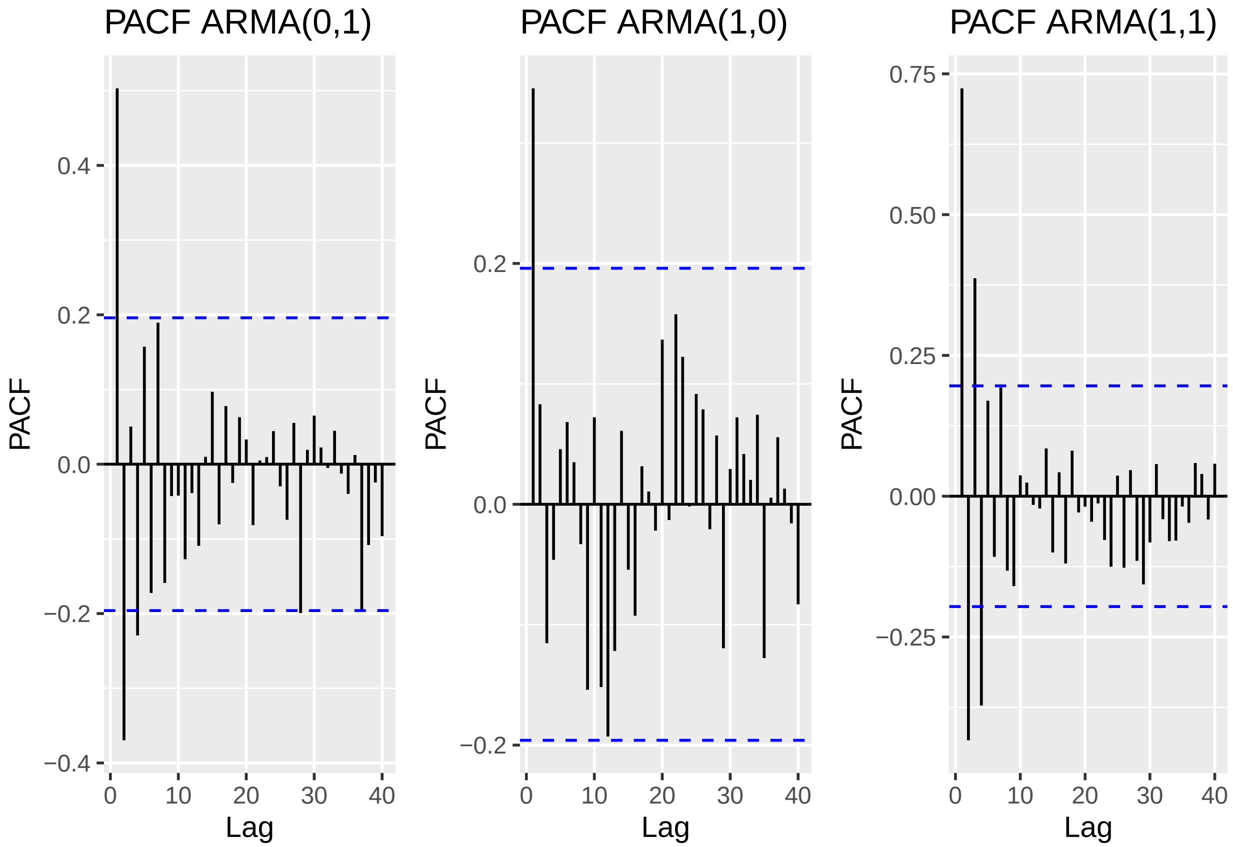
```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
## Ignoring unknown parameters: ‘main‘
```



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

```
plot_grid(
  autoplot(Pacf(arma0_1, lag = 40, plot=FALSE),
                main = "PACF ARMA(0,1)"),
   autoplot(Pacf(arma1_0, lag = 40, plot=FALSE),
                main = "PACF ARMA(1,0)"),
   autoplot(Pacf(arma1_1, lag = 40, plot=FALSE),
                main = "PACF ARMA(1,1)"),
  ncol=3
  )
```

```
## 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.

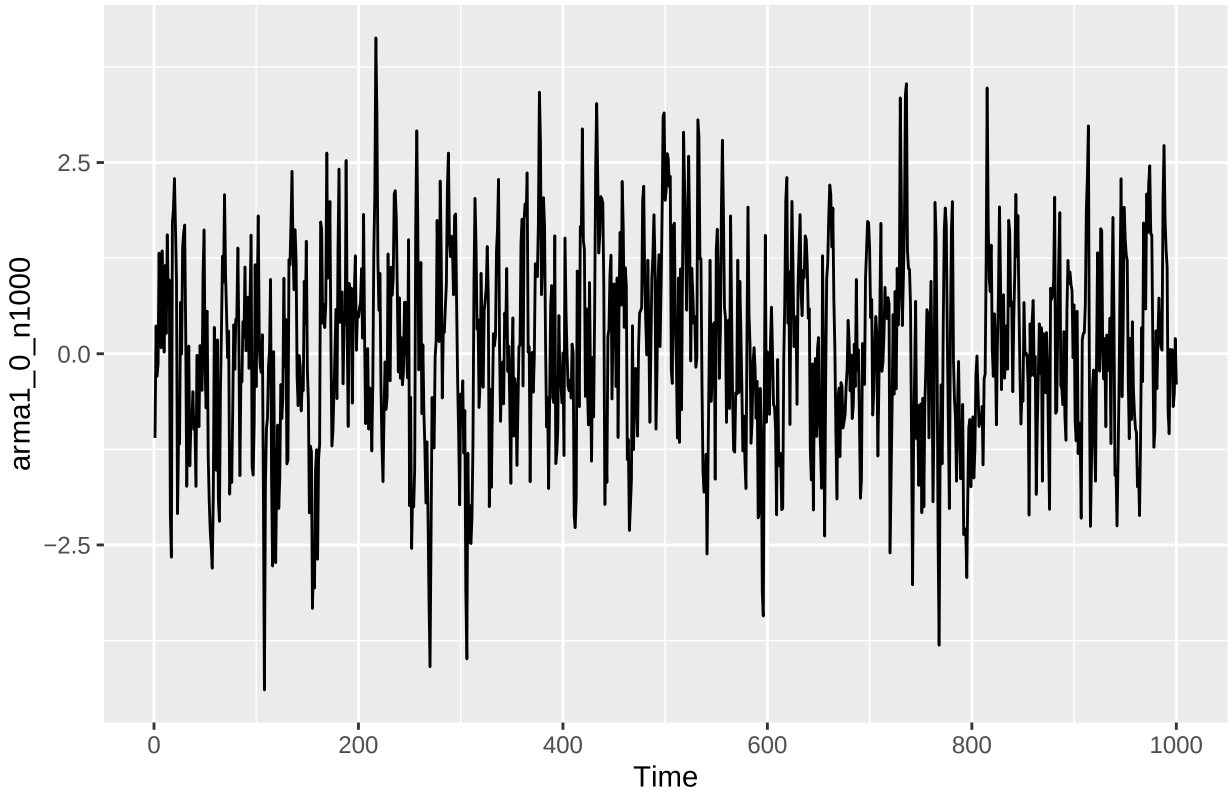
Answer: ARMA(0,1): For the ACF lag one is the most significant, and the following lags are mostly within the blue lines (not signficant). For the PACF if we only look at magnitude, then there is some exponential decay. So we could potentially identify this correctly, but these trends are not as clear as they would be if there were more samples. ARMA(1,0): For the ACF there is decay with the first two lags, and the PACF lag one is the only signifiant lag. These observations are indicative of an ARMA(1,0), but would be clearer with more samples. ARMA(1,1): We cannot really tell much from these graphs about how many MA and AR components are being modeled. It looks like both which makes it difficult to discern.

(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 ARMA(1,0) we would expect to see a PACF first component with a value 0.6, but the value is 0.35. For the ARMA(1,1) the PACF first lag has a value of around 0.75. Neither of these match the values provided for the lag 1 correlation coefficient of phi=0.6. This is because we don't have enough samples to get the true value- it is hard to replicate the theoretical value with fewer samples. We expect the ARMA(1,0) to have a value of 0.6, but because the ARMA(1,1) has both MA and AR components we wouldn't necessarily expect a value of 0.6 for the first lag.

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

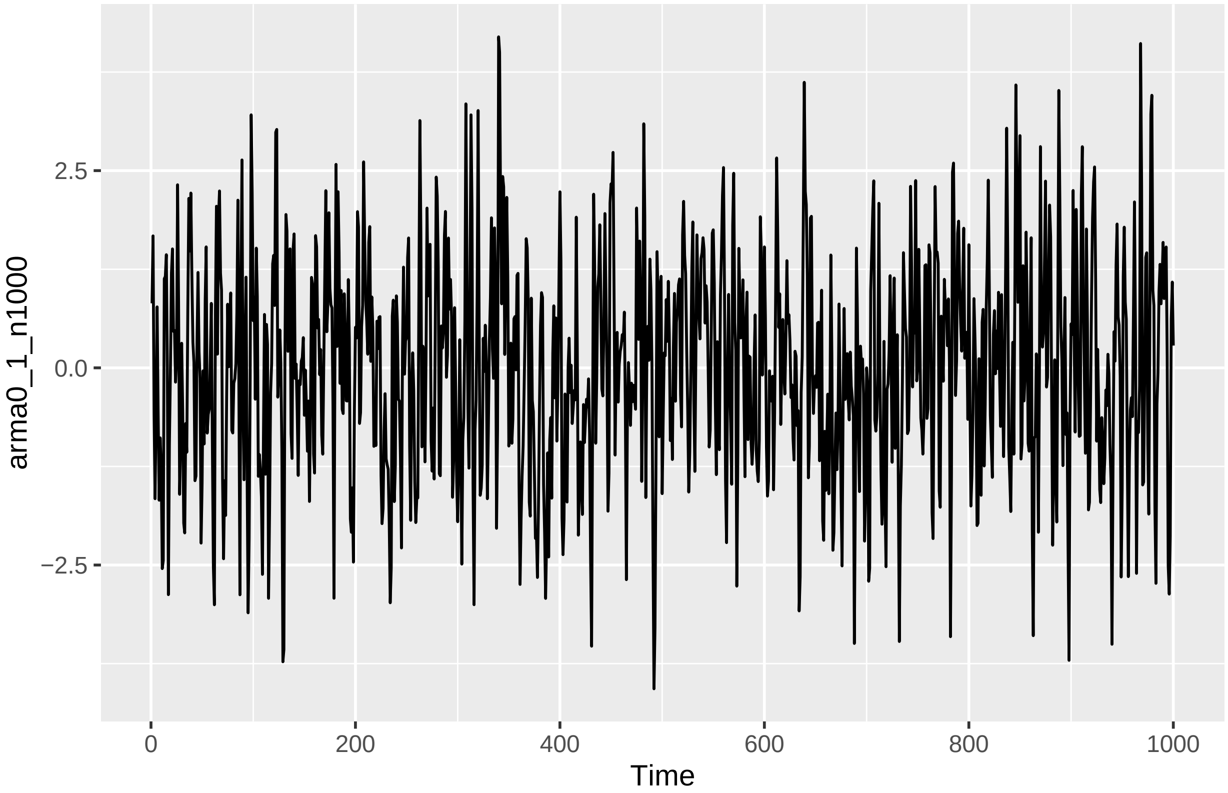
```
n= 1000
p <- 1 #order of AR component
q <- 0 #order of MA component
arma1_0_n1000 <- arima.sim(model = list(ar = (phi*p), ma = (theta*q)), n)
autoplot(arma1_0_n1000)
```



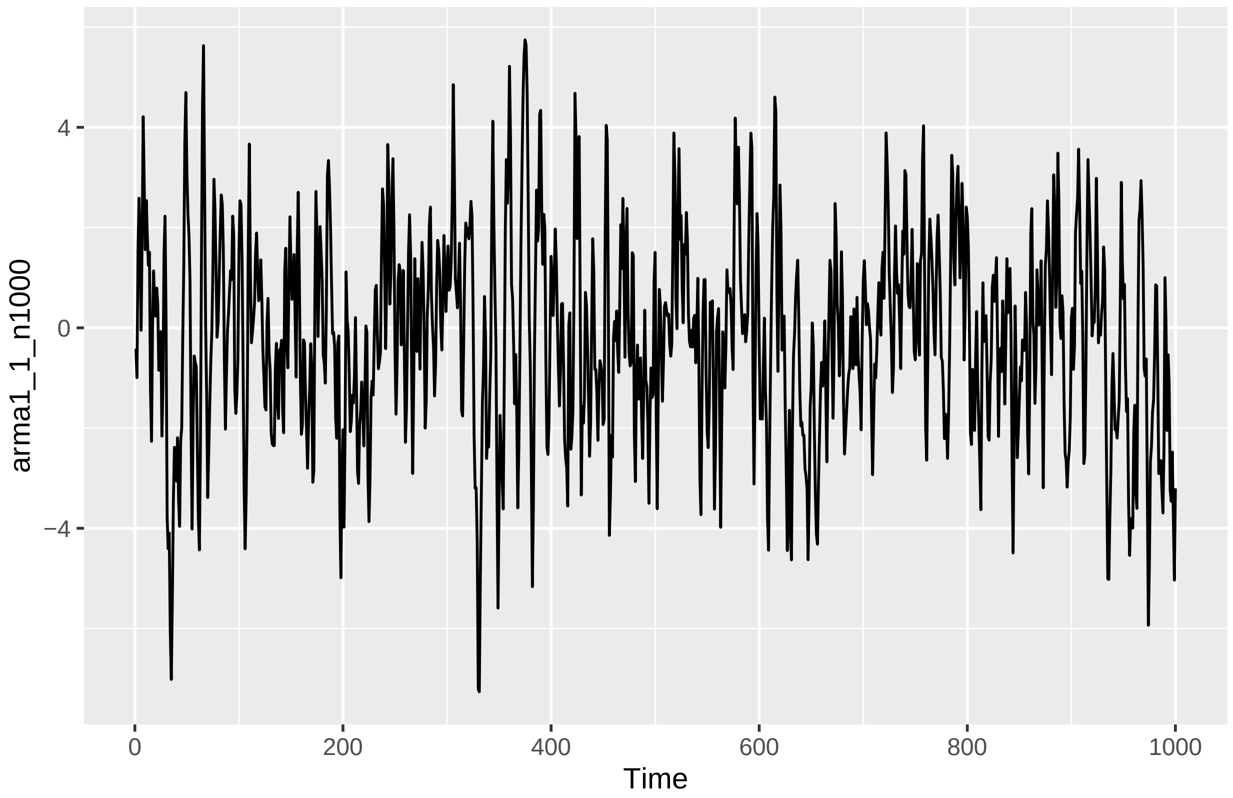
```
p <- 0 #order of AR component
q <- 1 #order of MA component
arma0_1_n1000 <- arima.sim(model = list(ar = (phi*p), ma = (theta*q)), n)
```

```
## Warning in min(Mod(polyroot(c(1, -model$ar)))): no non-missing arguments to
## min; returning Inf
```

```
autoplot(arma0_1_n1000)
```

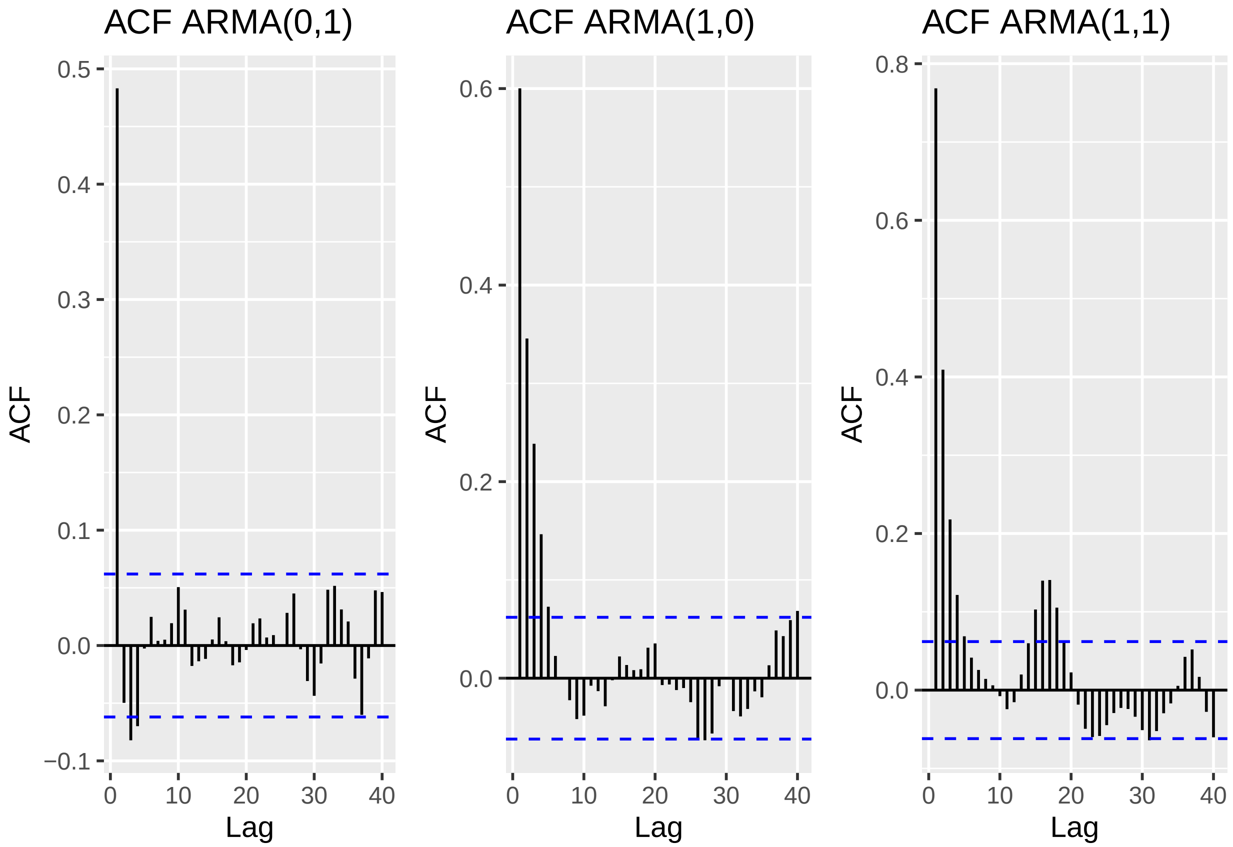


```
p <- 1 #order of AR component
q <- 1 #order of MA component
arma1_1_n1000 <- arima.sim(model = list(ar = (phi*p), ma = (theta*q)), n)
autoplot(arma1_1_n1000)
```



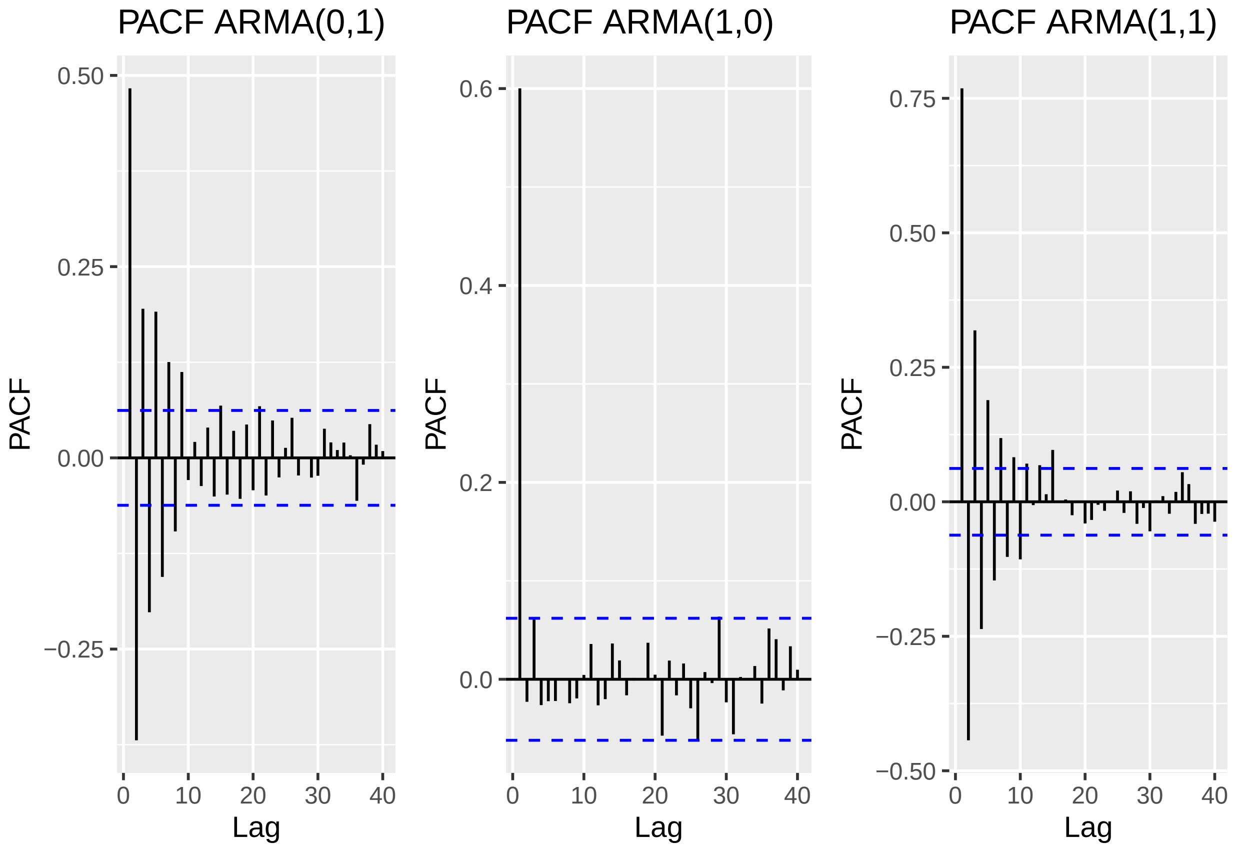
```
#acf for simulations with n=1000
plot_grid(
  autoplot(Acf(arma0_1_n1000, lag = 40, plot=FALSE),
                main = "ACF ARMA(0,1)"),
   autoplot(Acf(arma1_0_n1000, lag = 40, plot=FALSE),
                main = "ACF ARMA(1,0)"),
   autoplot(Acf(arma1_1_n1000, lag = 40, plot=FALSE),
                main = "ACF ARMA(1,1)"),
  ncol=3
  )
```

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



```
#pacf for simulations with n=1000
plot_grid(
  autoplot(Pacf(arma0_1_n1000, lag = 40, plot=FALSE),
                main = "PACF ARMA(0,1)"),
   autoplot(Pacf(arma1_0_n1000, lag = 40, plot=FALSE),
                main = "PACF ARMA(1,0)"),
   autoplot(Pacf(arma1_1_n1000, lag = 40, plot=FALSE),
                main = "PACF ARMA(1,1)"),
  ncol=3
  )
```

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



(d.2) 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: ARMA(1,0) yes! We see a clear exponential decay in the ACF, and one signficant lag for the PACF. ARMA(0,1) this is also more obvious with an increase in the number of samples. When just looking at magnitude in the PACF there is clear slow decay. For the ACF the first lag has the most significance and the remaining lags are mostly insignficant. These are indicative of an ARMA model with q=1. ARMA(1,1): It is still difficult to discern the order of the model for MA and AR becuase there are patterns for both in the ACF and PACF.

(e.2) 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: Yes, for the ARMA(1,0) the PACF value of the first lag is 0.6, which matches our coefficient phi= 0.6 For the ARMA(1,1), we don't see a PACF value of the first lag as 0.6 because there are also MA components being dispalyed in this graph. So it makes sense that the phi doesn't match with the first lag PACF value.

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: SARIMA(1,0,1)(1,0,0)_12

p (# of AR terms) = 1

P (# of SAR terms) = 1

q (# of MA terms) = 1

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

Answer:

phi_1 = 0.7

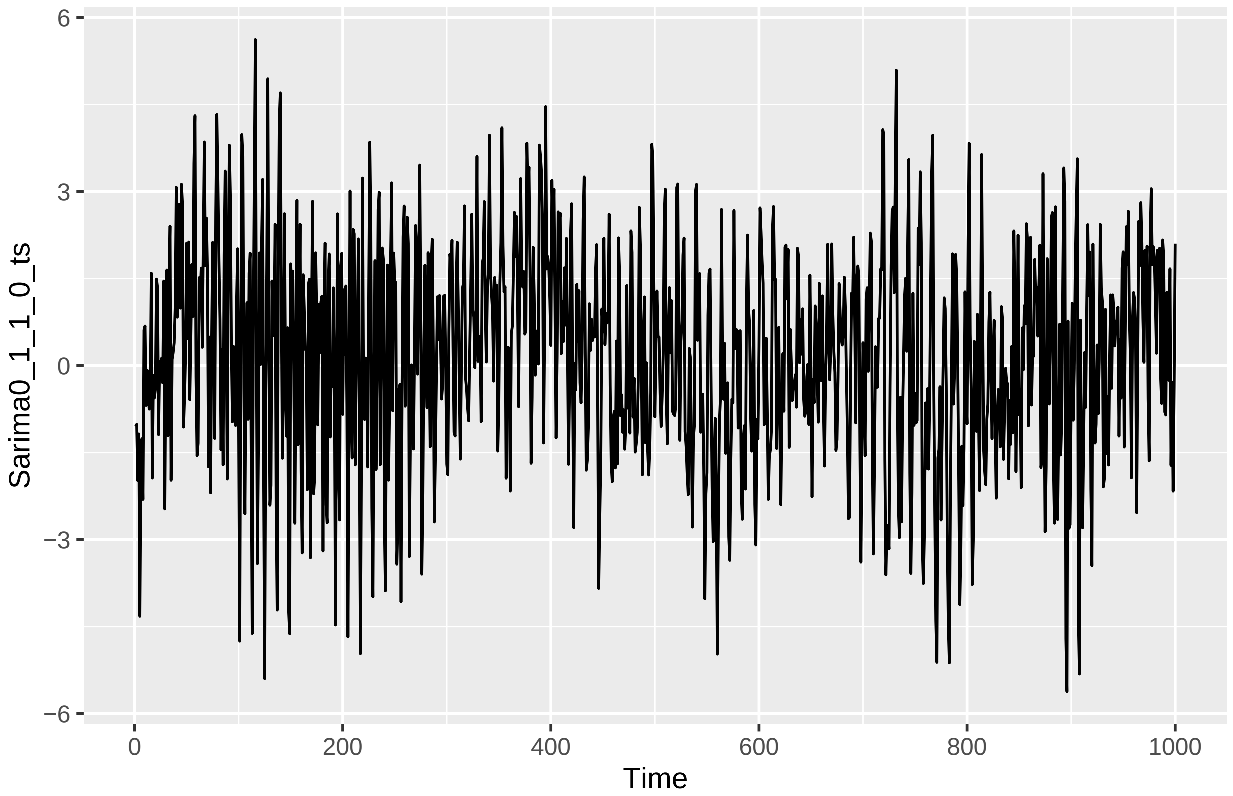
phi_12 = -0.25

theta_1 =-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
phi <- 0.8
theta <- 0.5
Sarima0_1_1_0 <- sim_sarima(model= list(sar= (phi), ma= (theta), nseasons=12), n=1000)
Sarima0_1_1_0_ts <- ts(Sarima0_1_1_0)
autoplot(Sarima0_1_1_0_ts)
```



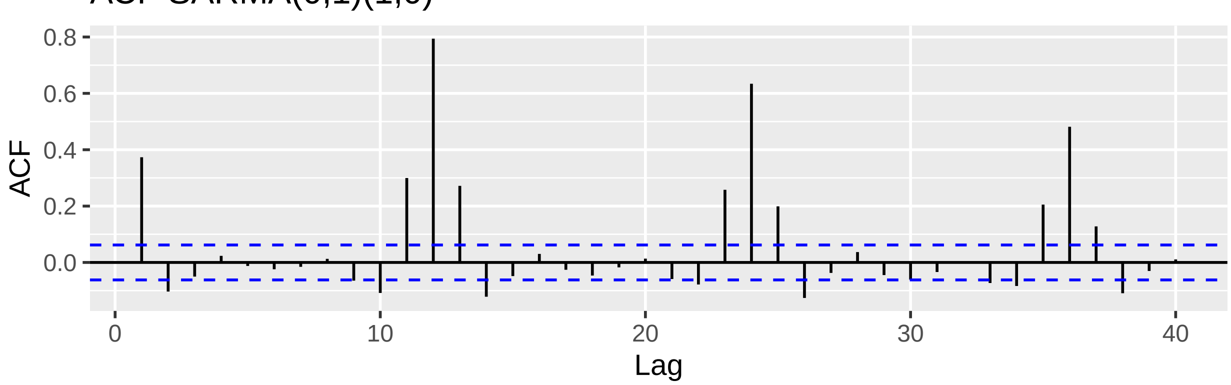
> Answer: It is centered about the mean and has repetitive peaks/troughs which is indicitive of seasonality, but there is variability in the mangitude of the max/min so it is hard to tell whether or not it is seasonal. I would need to look at the ACF/PACF to confirm is there was seasonality.

Q5

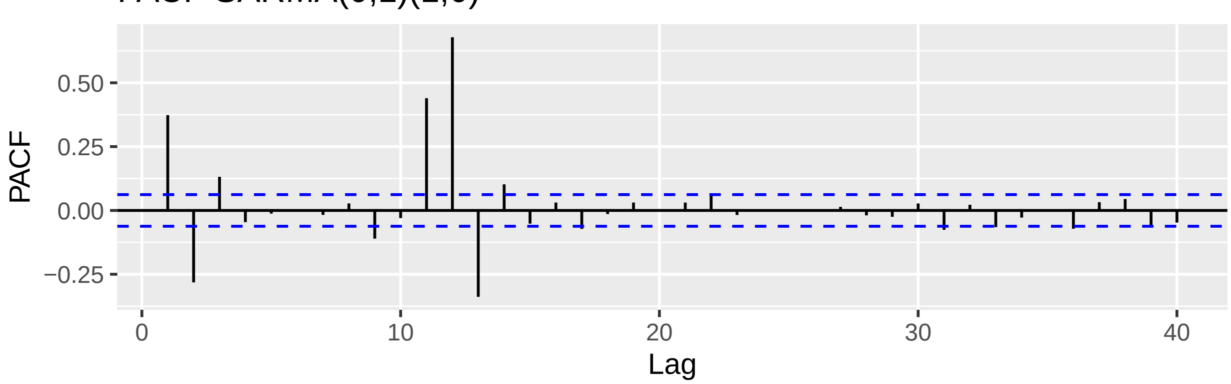
Plot ACF and PACF of the simulated series in Q4. Comment if the plots are well representing the model you simulated, i.e., would you be able to identify the order of both non-seasonal and seasonal components from the plots? Explain.

```
plot_grid(
  autoplot(Acf(Sarima0_1_1_0_ts, lag = 40, plot=FALSE),
                main = "ACF SARMA(0,1)(1,0)"),
   autoplot(Pacf(Sarima0_1_1_0_ts, lag = 40, plot=FALSE),
                main = "PACF SARMA(0,1)(1,0)"),
  nrow=2
  )
```

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



ACF SARMA(0,1)(1,0)



PACF SARMA(0,1)(1,0)

> Answer: There is clear seasonality in the ACF plot (decaying signficant lags at 12, 24, and 36). The PACF also has a single positive and significant spike at lag 12, both of which indicate a seasonal AR component (which is what we want and what we modeled!) Because we have one MA component with a coefficient of theta=0.5 we would expect the first ACF lag to have a magnitude of 0.5, and that the PACF graph would have a slow decay. We do see that the first lag in the ACF is significant, though the magnitude is not what we would expect. Additionally, the magnitude of the PACF does have a slow decay. So it is kind of clear, but would likely be more obvious if seasonality were removed.