ENV 797 - Time Series Analysis for Energy and Environment Applications | Spring 2025

Assignment 6 - Due date 02/27/25

Rosie Wu

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_Sp25.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(lubridate)
```

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

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

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

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

```r
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 sample autocorrelation function (ACF) plot for an AR process typically shows a gradual decay rather than an abrupt cutoff. The decay pattern can be fluctuational or exponential, depending on the nature of the AR coefficients. The partial autocorrelation function (PACF) plot for an AR process shows significant spikes at lags 1 and 2, with all subsequent lags being insignificant (close to zero). This reflects the fact that the AR process has direct dependencies up to lag 2 but not beyond.

- MA(1)

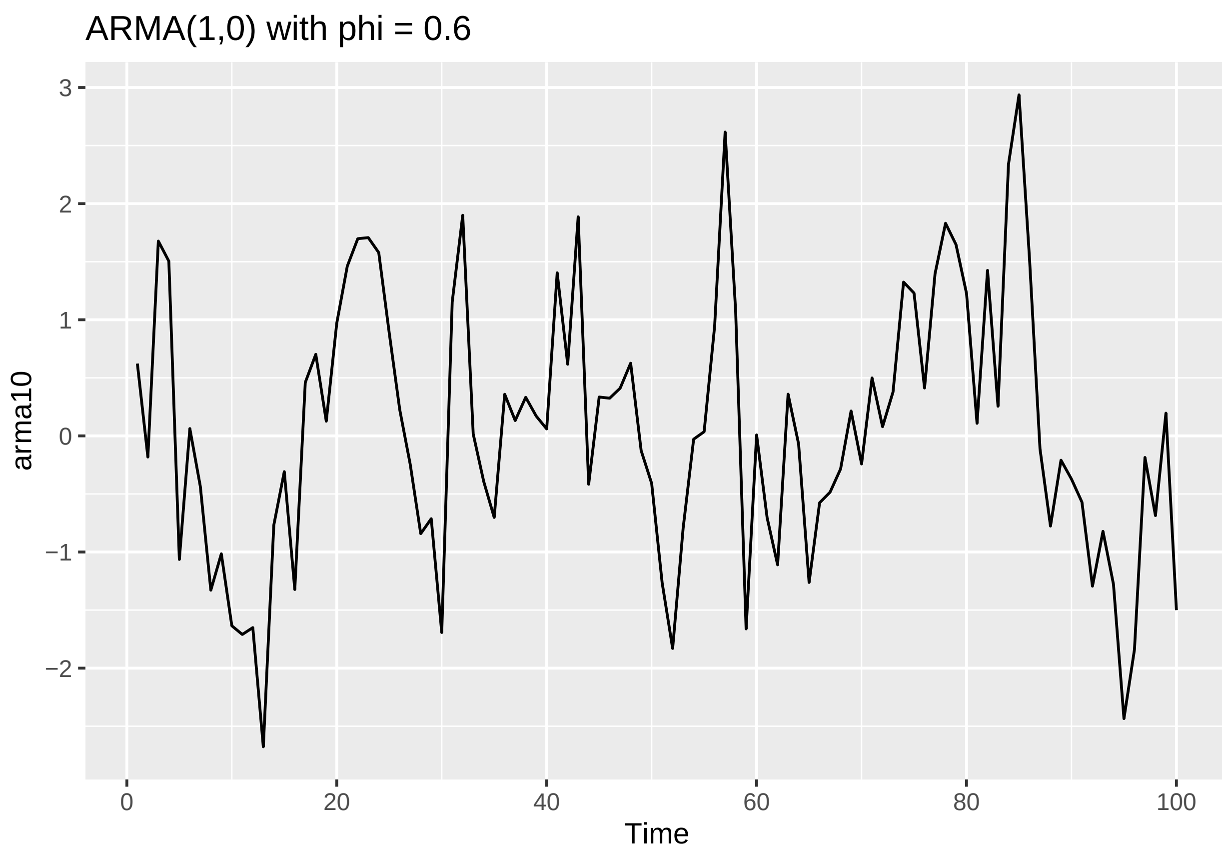
  Answer: The sample autocorrelation function (ACF) plot for an MA(1) process shows a significant spike at lag 1, followed by a sudden cutoff to near zero for higher lags. This reflects the short memory of the MA(1) process. The partial autocorrelation function (PACF) plot exhibits a gradual decay rather than an abrupt cutoff, as the PACF captures the indirect dependencies of the moving average process across multiple lags.

Q2

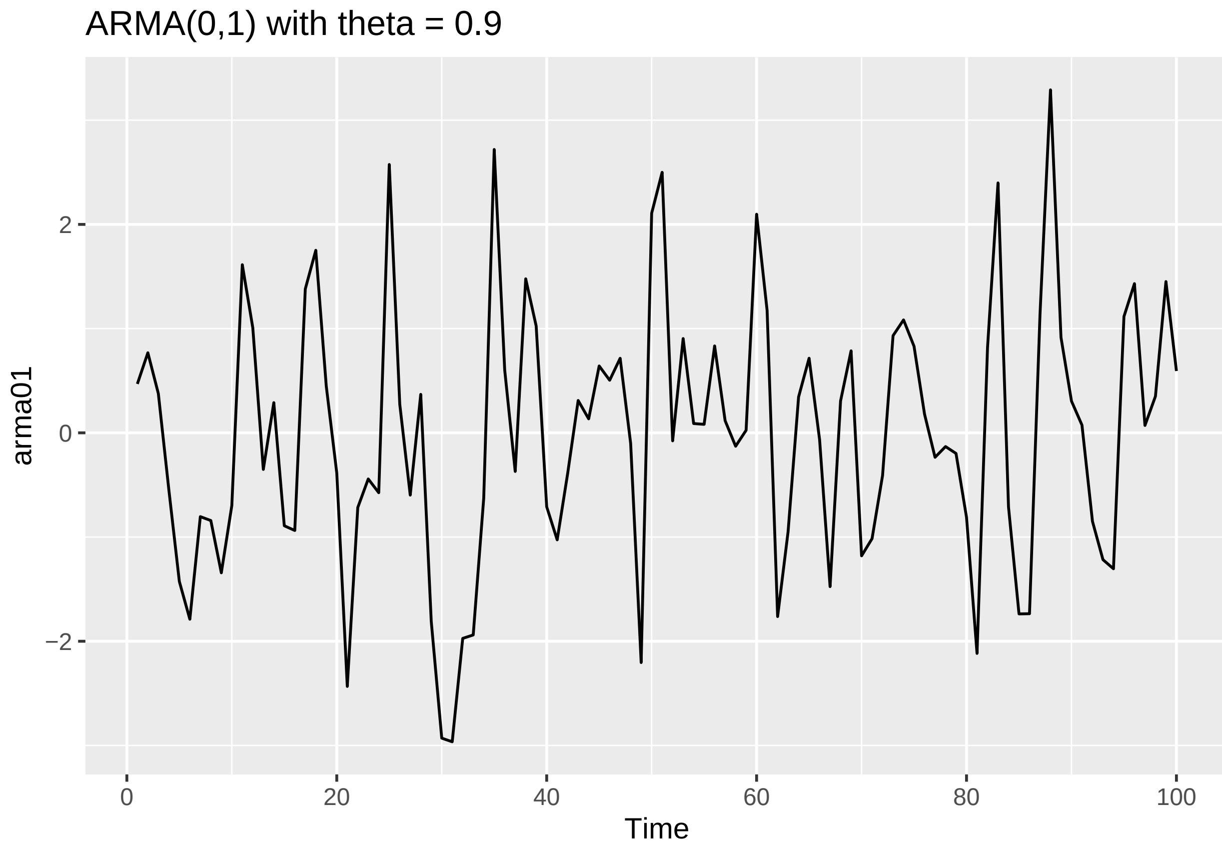
Recall that the non-seasonal ARIMA is described by three parameters ARIMA(p, d, q) where p is the order of the autoregressive component, d is the number of times the series need to be differenced to obtain stationarity and q is the order of the moving average component. If we don't need to difference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

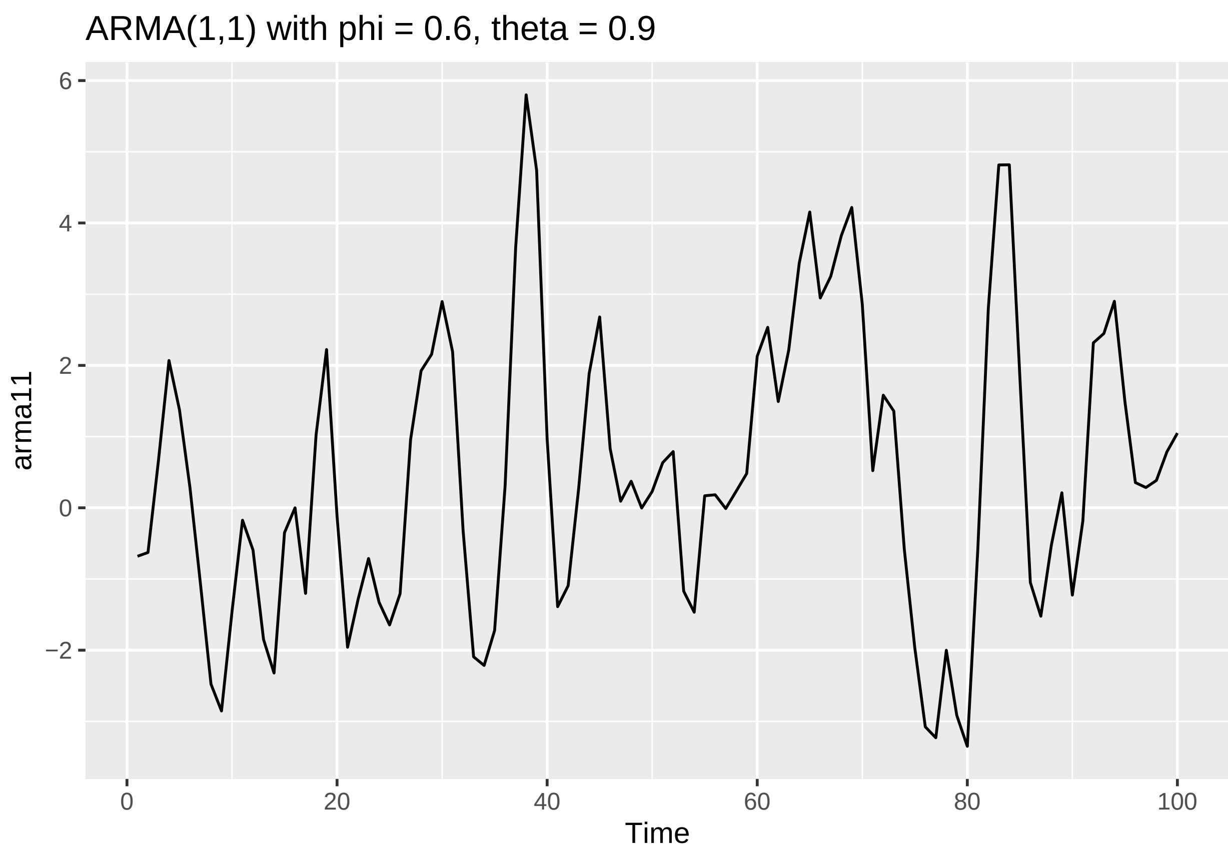
```r
# Set seed for reproducibility
set.seed(123)

# Define parameters
n <- 100
phi <- 0.6
theta <- 0.9

# Generate ARMA(1,0) = AR(1) model
arma10 <- arima.sim(model = list(ar = phi), n = n)

# Generate ARMA(0,1) = MA(1) model
arma01 <- arima.sim(model = list(ma = theta), n = n)

# Generate ARMA(1,1) model
arma11 <- arima.sim(model = list(ar = phi, ma = theta), n = n)

# Plot the generated series
autoplot(arma10) + ggtitle("ARMA(1,0) with phi = 0.6")
```




```r
autoplot(arma01) + ggtitle("ARMA(0,1) with theta = 0.9")
```

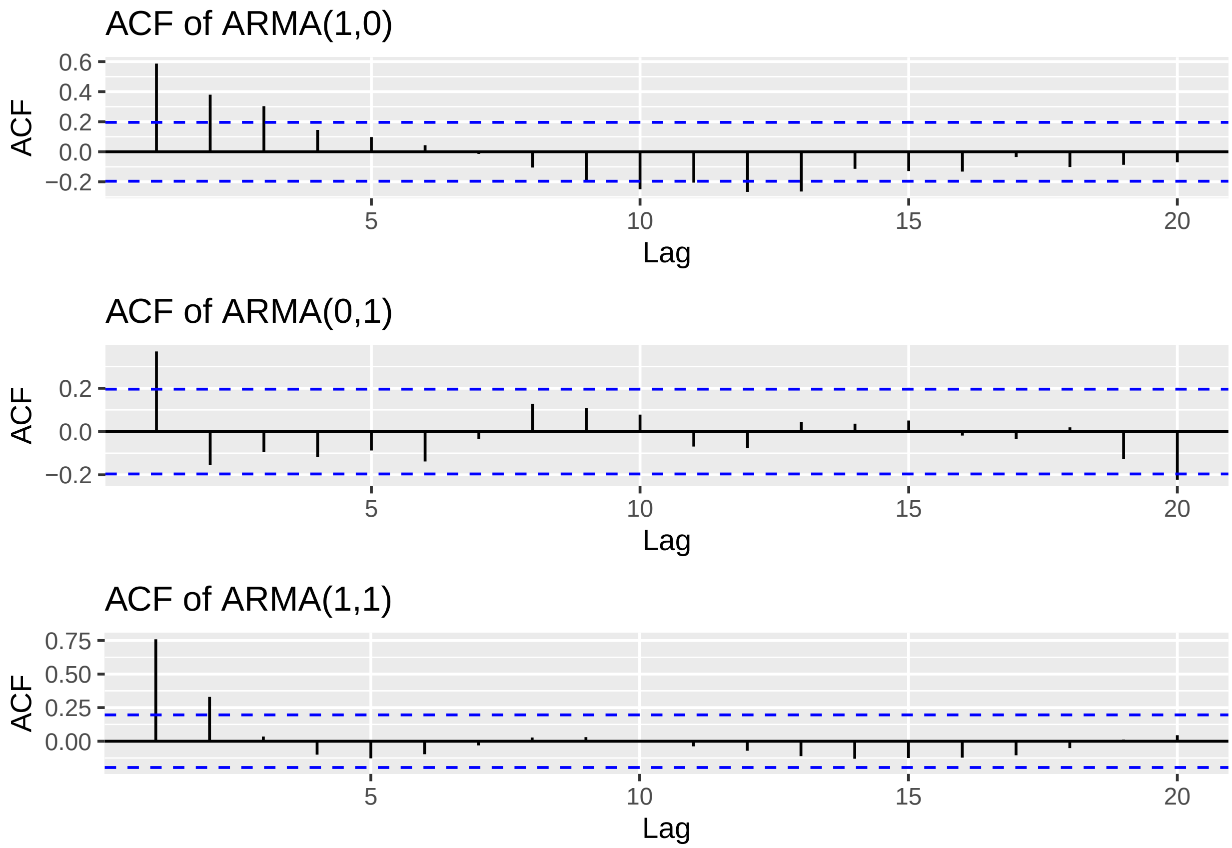



```r
autoplot(arma11) + ggtitle("ARMA(1,1) with phi = 0.6, theta = 0.9")
```



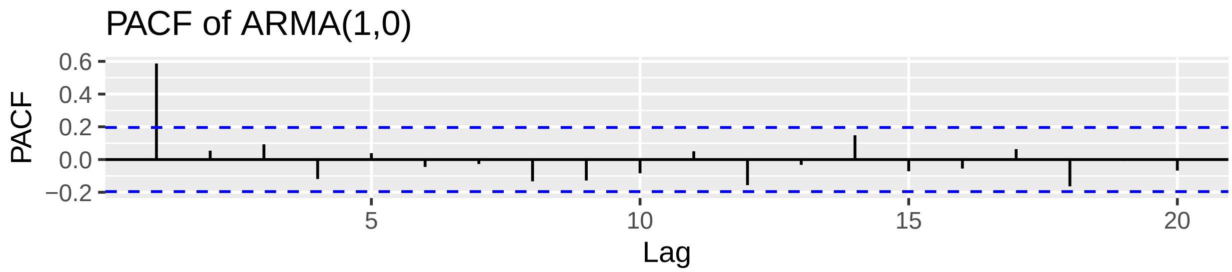
(b) Plot the sample ACF for each of these models in one window to facilitate comparison (Hint: use `cowplot::plot_grid()`).

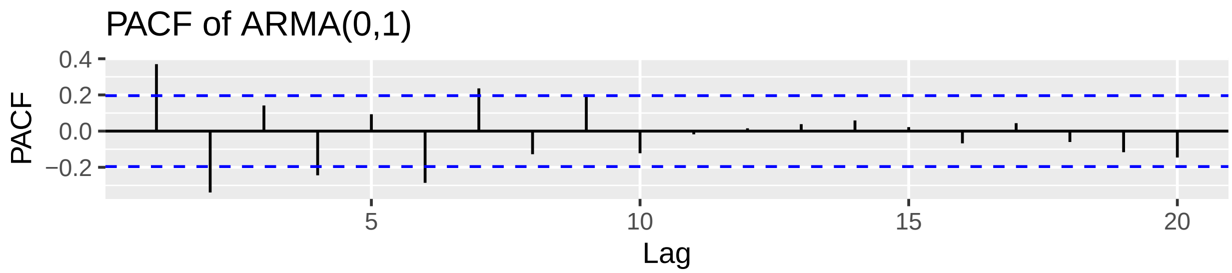
```
# Compute ACF plots
acf_arma10 <- ggAcf(arma10) + ggtitle("ACF of ARMA(1,0)")
acf_arma01 <- ggAcf(arma01) + ggtitle("ACF of ARMA(0,1)")
acf_arma11 <- ggAcf(arma11) + ggtitle("ACF of ARMA(1,1)")

# Arrange plots in one window
plot_grid(acf_arma10, acf_arma01, acf_arma11, ncol = 1)
```

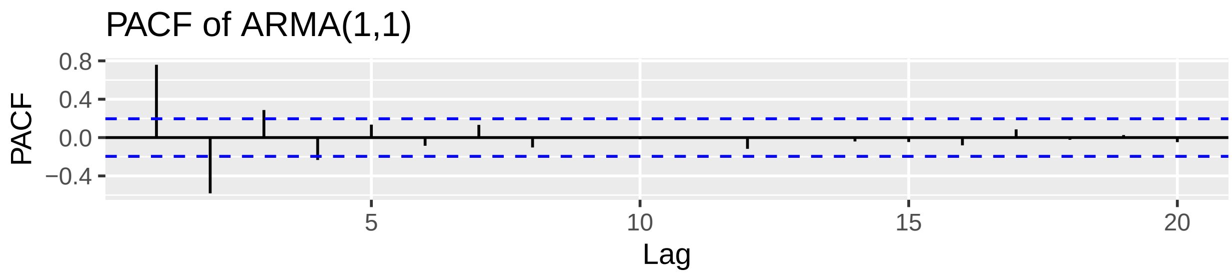


(c) Plot the sample PACF for each of these models in one window to facilitate comparison.

```
pacf_arma10 <- ggPacf(arma10) + ggtitle("PACF of ARMA(1,0)")
pacf_arma01 <- ggPacf(arma01) + ggtitle("PACF of ARMA(0,1)")
pacf_arma11 <- ggPacf(arma11) + ggtitle("PACF of ARMA(1,1)")

# Arrange plots in one window
plot_grid(pacf_arma10, pacf_arma01, pacf_arma11, ncol = 1)
```







(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.

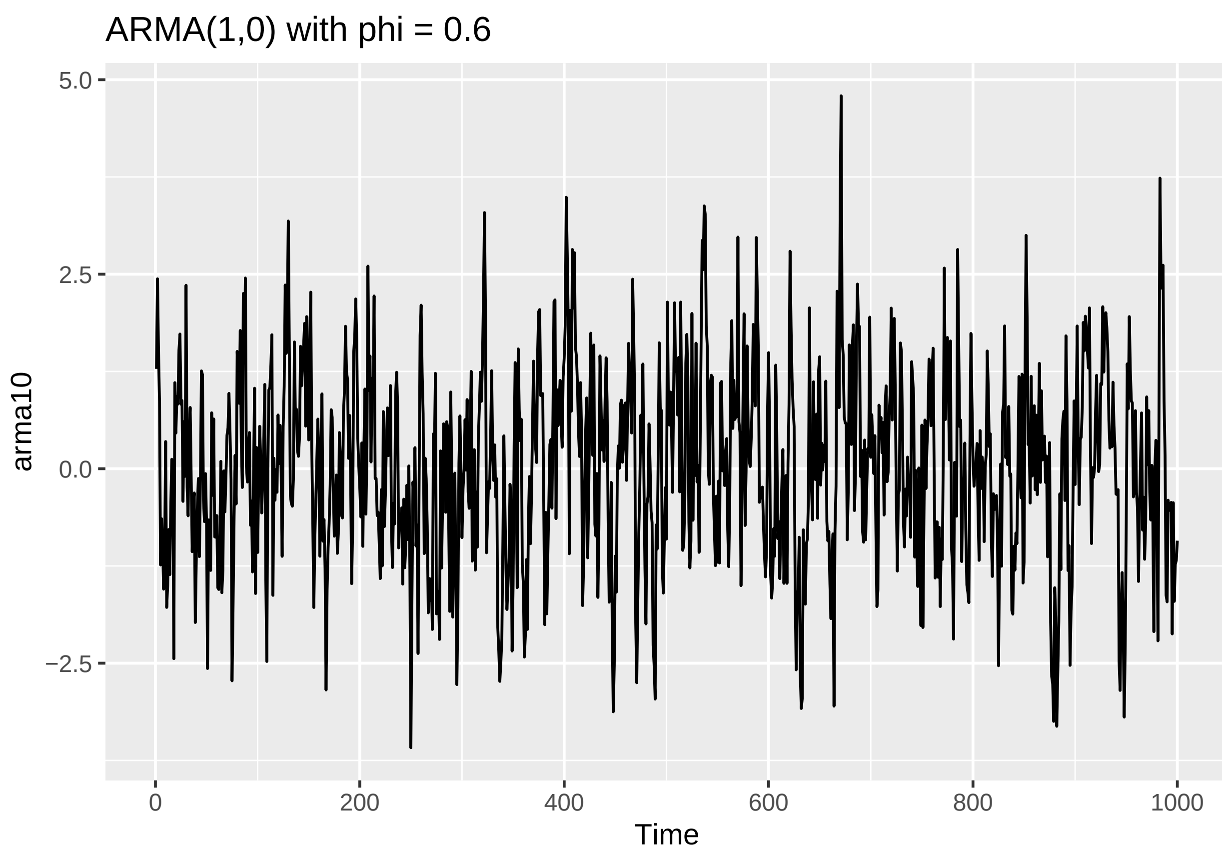
Answer: For ARIMA(1,0), there is a more gradual decay rather than sharp cutoff in the ACF plot; and there is a significant spike at lag 1 in PACF. If we observe these characteristics in a dataset, we would classify it as an AR(p) model with order p = 1. For ARIMA(0,1): ACF: A significant spike at lag 1, then a sharp cutoff. PACF: A gradual decay, rather than an immediate cutoff. This suggests an MA(q) model with order q = 1. For ARMA(1,1): In ACF: There is a decay pattern but not as clear as a pure AR model. In PACF: A significant spike at lag 1, but not a sharp cutoff. When neither the ACF nor the PACF exhibits a strict cutoff and both show some decay, the model is likely ARMA(p, q) with p = 1, q = 1.

(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?

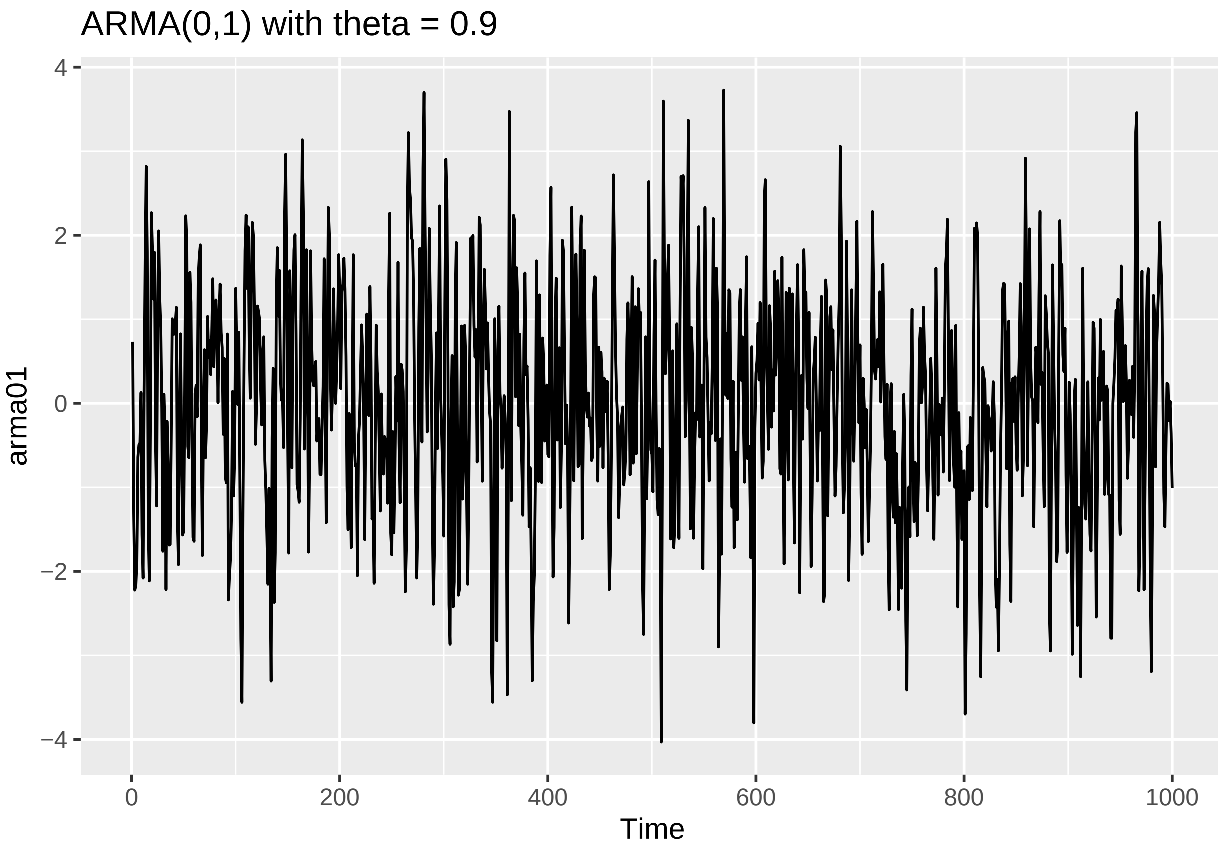
Answer: Lag 1 value for ARMA(1,0) PACF is about 0.6, which is matches the $\phi = 0.6$, and the Lag 1 value for ARMA(1,1) PACF is about 0.8, which is more than 0.6. PACF for ARMA(1,0) correctly reflects $\phi = 0.6$ at lag 1, since it's an AR model and it should match the phi. PACF for ARMA(1,1) does not exactly match 0.6, which is expected due to the moving average influence.

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

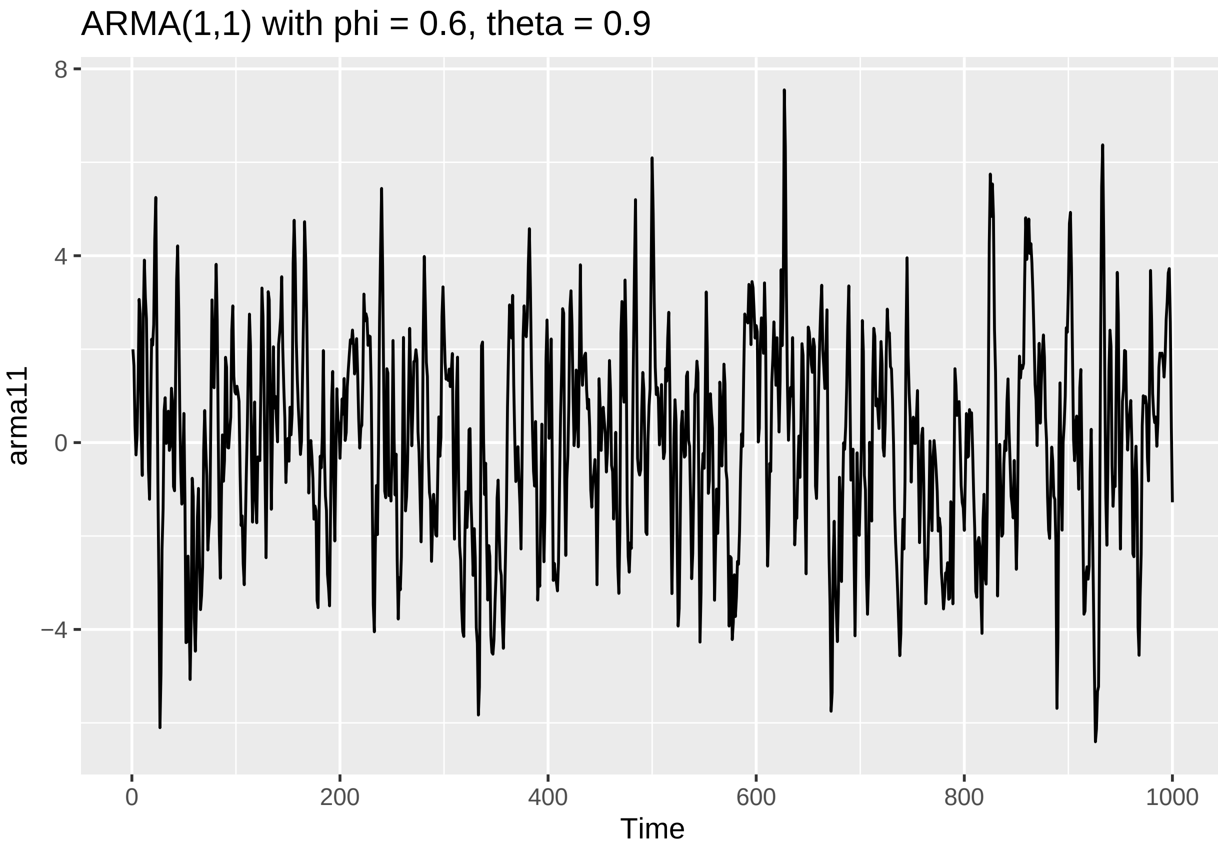
```r
# Define parameters
n2 <- 1000
phi <- 0.6
theta <- 0.9

# Generate ARMA(1,0) = AR(1) model
arma10 <- arima.sim(model = list(ar = phi), n = n2)

# Generate ARMA(0,1) = MA(1) model
arma01 <- arima.sim(model = list(ma = theta), n = n2)

# Generate ARMA(1,1) model
arma11 <- arima.sim(model = list(ar = phi, ma = theta), n = n2)

# Plot the generated series
autoplot(arma10) + ggtitle("ARMA(1,0) with phi = 0.6")
```



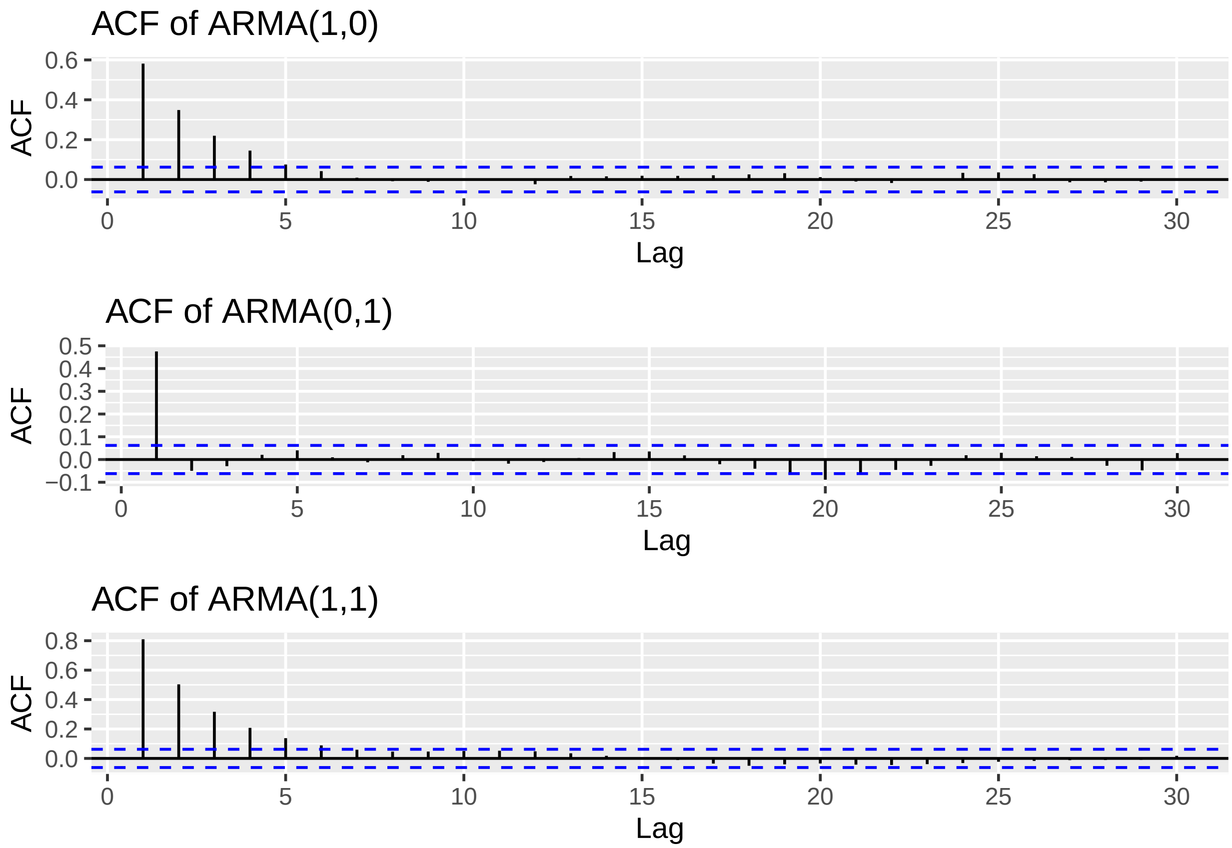

```r
autoplot(arma01) + ggtitle("ARMA(0,1) with theta = 0.9")
```



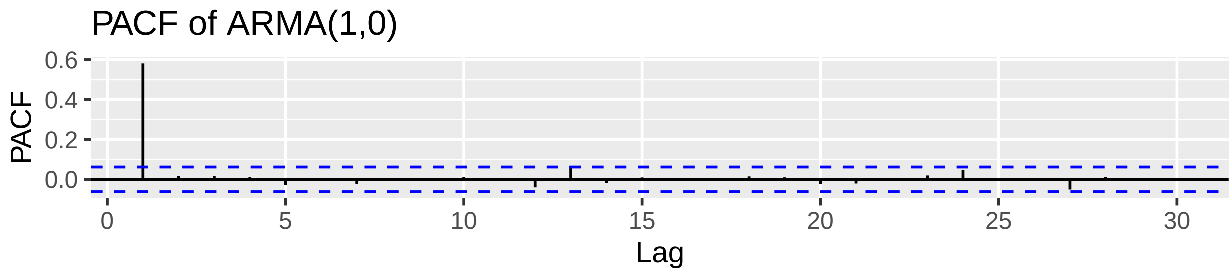

```
autoplot(arma11) + ggtitle("ARMA(1,1) with phi = 0.6, theta = 0.9")
```

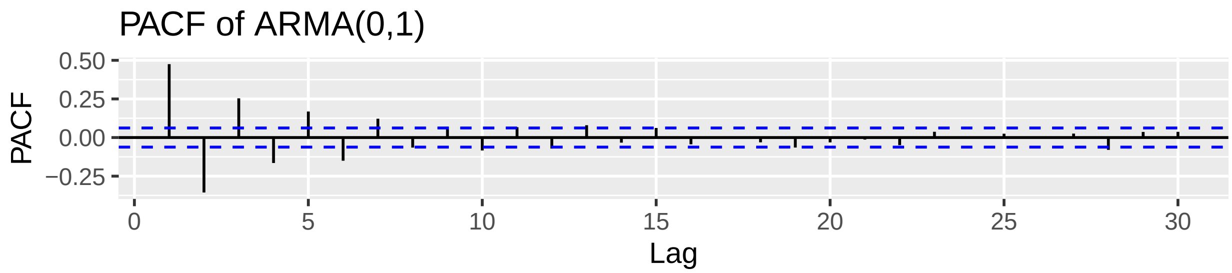


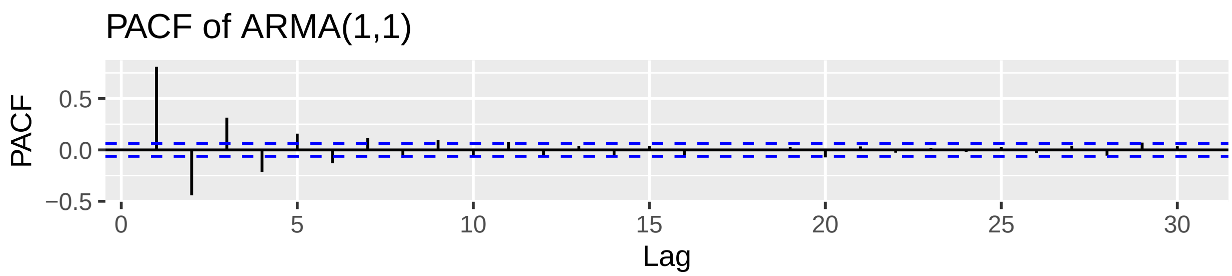

```r
# Compute ACF plots
acf_arma10_f <- ggAcf(arma10) + ggtitle("ACF of ARMA(1,0)")
acf_arma01_f <- ggAcf(arma01) + ggtitle("ACF of ARMA(0,1)")
acf_arma11_f <- ggAcf(arma11) + ggtitle("ACF of ARMA(1,1)")

# Arrange plots in one window
plot_grid(acf_arma10_f, acf_arma01_f, acf_arma11_f, ncol = 1)
```




```r
pacf_arma10_f <- ggPacf(arma10) + ggtitle("PACF of ARMA(1,0)")
pacf_arma01_f <- ggPacf(arma01) + ggtitle("PACF of ARMA(0,1)")
pacf_arma11_f <- ggPacf(arma11) + ggtitle("PACF of ARMA(1,1)")

# Arrange plots in one window
plot_grid(pacf_arma10_f, pacf_arma01_f, pacf_arma11_f, ncol = 1)
```







> After increasing the sample size, I got the same conclusions as what models they fit. > For ARIMA(1,0), there is a more gradual decay rather than sharp cutoff in the ACF plot; and there is a significant spike at lag 1 in PACF. If we observe these characteristics in a dataset, we would classify it as an AR(p) model with order p = 1. For ARIMA(0,1): ACF: A significant spike at lag 1, then a sharp cutoff. PACF: A gradual decay, rather than an immediate cutoff. This suggests an MA(q) model with order q = 1. For ARMA(1,1): In ACF: There is a decay pattern but not as clear as a pure AR model. In PACF: A significant spike at lag 1, but not a sharp cutoff. When neither the ACF nor the PACF exhibits a strict cutoff and both show some decay, the model is likely ARMA(p, q) with p = 1, q = 1. > With the updated models, the values of lag 1 of PACFs of ARMA(1,0) and ARMA(1,1) also remains roughly the same. The ARMA(1,0) lag 1 value still roughly equals to 0.6 (phi), and ARMA(1,1)'s value is a little higher, which still aligns with the characteristics of AR and ARMA 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 $\text{ARIMA}(p, d, q)(P, D, Q)_s$, i.e., identify the integers p, d, q, P, D, Q, s (if possible) from the equation.

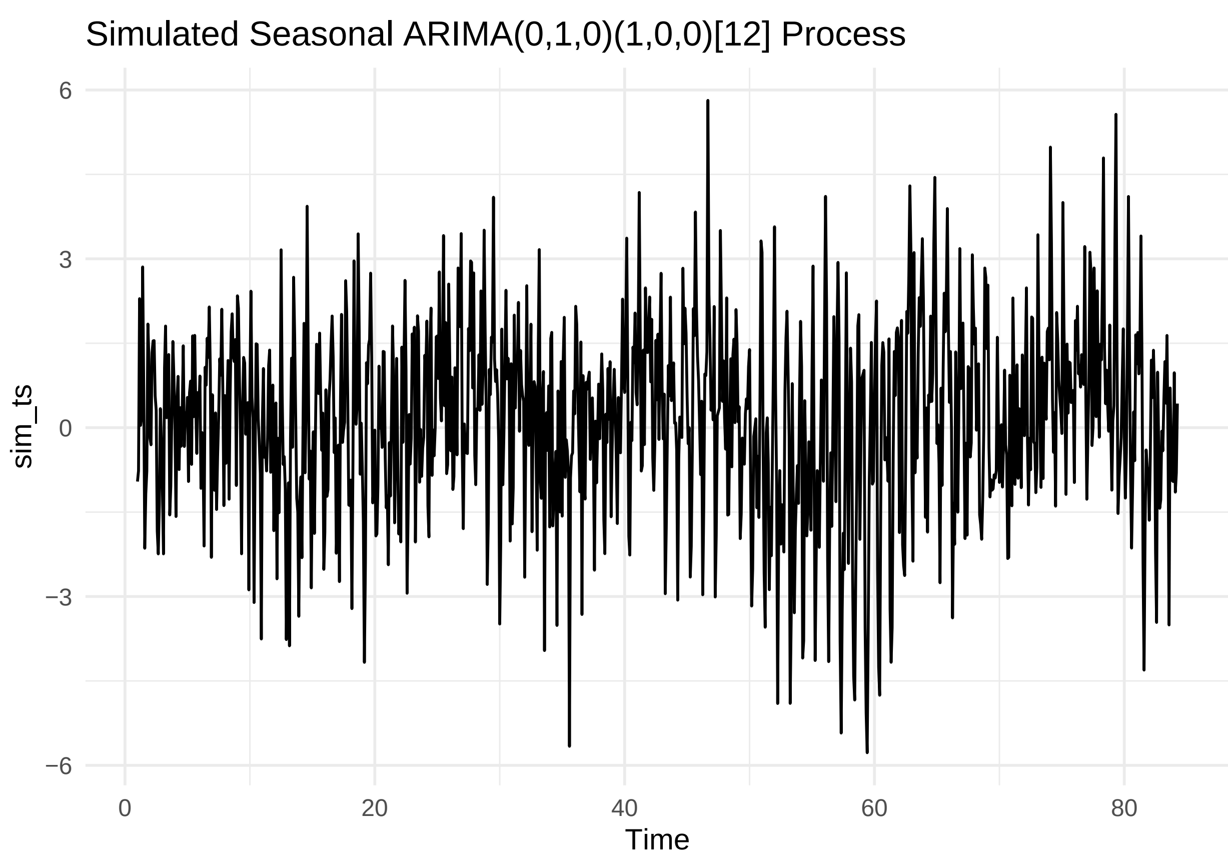
Answer: Autoregressive (AR) terms: $0.7 * y_{t-1}$ indicates an AR(1) component (p=1). $-0.25 * y_{t-12}$ indicates seasonal AR(1) component at lag 12 (p= 1, seasonal period s=12).
No differencing is explicitly stated, meaning d=0 (non-seasonal differencing) and D=0 (seasonal differencing).
$-0.1 * a_{t-1}$ represents an MA(1) component (q=1). No seasonal MA terms present, so Q=0. So, $ARIMA(1, 0, 1)(1, 0, 0)_12$.

(b) Also from the equation what are the values of the parameters, i.e., model coefficients. > Answers: AR(1) coefficient phi_1 = 0.7 > Seasonal AR(1) coefficient = Phi_1 = -0.25
> theta_1 MA(1) 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?

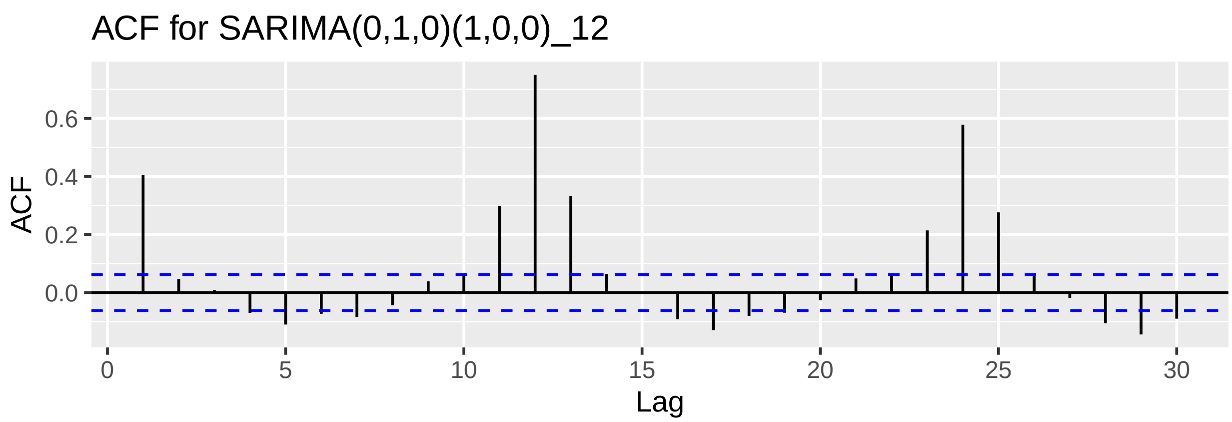
```r
# Simulate the SARIMA(0,1,0)(1,0,0)_12 process
set.seed(123)

sim_data4 <- sim_sarima(n = 1000, model = list(ma = 0.5, sar = 0.8, nseasons=12))

# Convert to time series object, monthly
sim_ts <- ts(sim_data4, frequency = 12)

# plot
autoplot(sim_ts) +
  ggtitle("Simulated Seasonal ARIMA(0,1,0)(1,0,0)[12] Process") +
  theme_minimal()
```

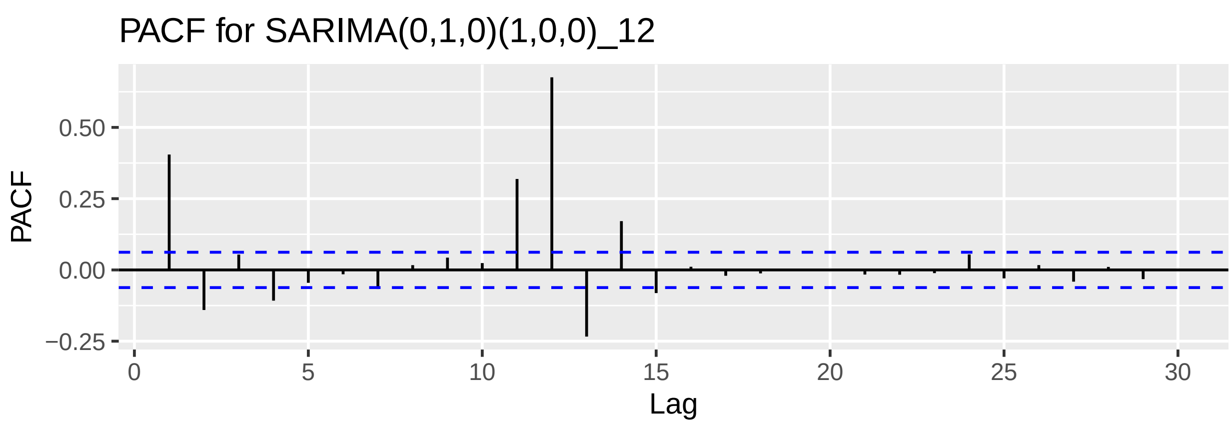


Q5

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
# plots for ACF and PACF
acf_sarima <- ggAcf(sim_data4) + ggtitle("ACF for SARIMA(0,1,0)(1,0,0)_12")
pacf_sarima <- ggPacf(sim_data4) + ggtitle("PACF for SARIMA(0,1,0)(1,0,0)_12")
# Arrange plots in one window
plot_grid(acf_sarima, pacf_sarima, ncol = 1)
```





> The ACF plot shows a slow decay at non-seasonal lags. This is consistent with the differencing (d=1) in the non-seasonal part of the model (ARIMA(0,1,0)). There are also no more significant spikes at non-seasonal lags (lag 1-11) and (lag 13-23). There are seasonal spikes, in contrasts, at lag 12 and 24 in the ACF plots, which means seasonal AR, and the spike decays slowly after lag 12 in later seasonal lags (24), which is expected for seasonal AR(1) models. In PACF plot, the non-seasonal lags have no significant spikes, and the spikes cut off/ dropped after lag 12 (which is the only 1 significant spike), which is expected for AR(1) models.