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

Assignment 6 - Due date 02/27/25

Loo Si Min

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.

```
#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)
```

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

```
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.4.3
```

```
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)
```

```
## Warning: package 'sarima' was built under R version 4.4.3
```

```
## 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: For an AR(2) process, the autocorrelation function (ACF) does not cut off after a specific lag but instead gradually declines toward zero. This decline may follow an exponential pattern or display a wave-like behavior depending on the values of the model parameters. In contrast, the partial autocorrelation function (PACF) for an AR(2) model shows a sharp cutoff after lag 2. Specifically, the PACF will exhibit statistically significant spikes at lags 1 and 2, while autocorrelations at higher lags will be close to zero and fall within the confidence bounds.

- MA(1)

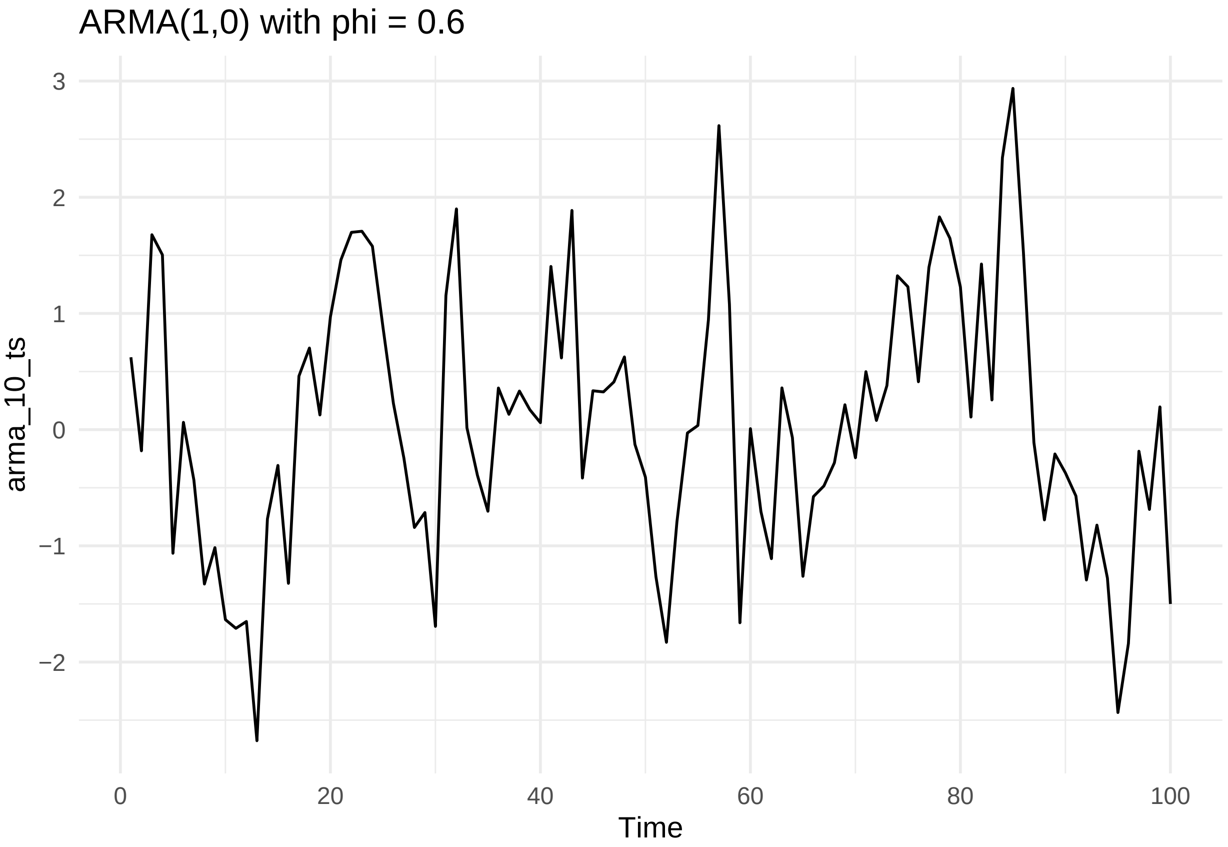
  Answer: For an MA(1) process, the pattern is reversed. The ACF cuts off sharply after lag 1, meaning there is a significant spike at lag 1 and autocorrelations at higher lags are approximately zero. This abrupt cutoff occurs because the process depends only on the current shock and the immediately preceding shock. Meanwhile, the PACF for an MA(1) model does not exhibit a sharp cutoff but instead declines gradually over increasing lags, often in an exponential or oscillatory pattern. Thus, for moving average models, the ACF cuts off at lag q, while the PACF tails off

Q2

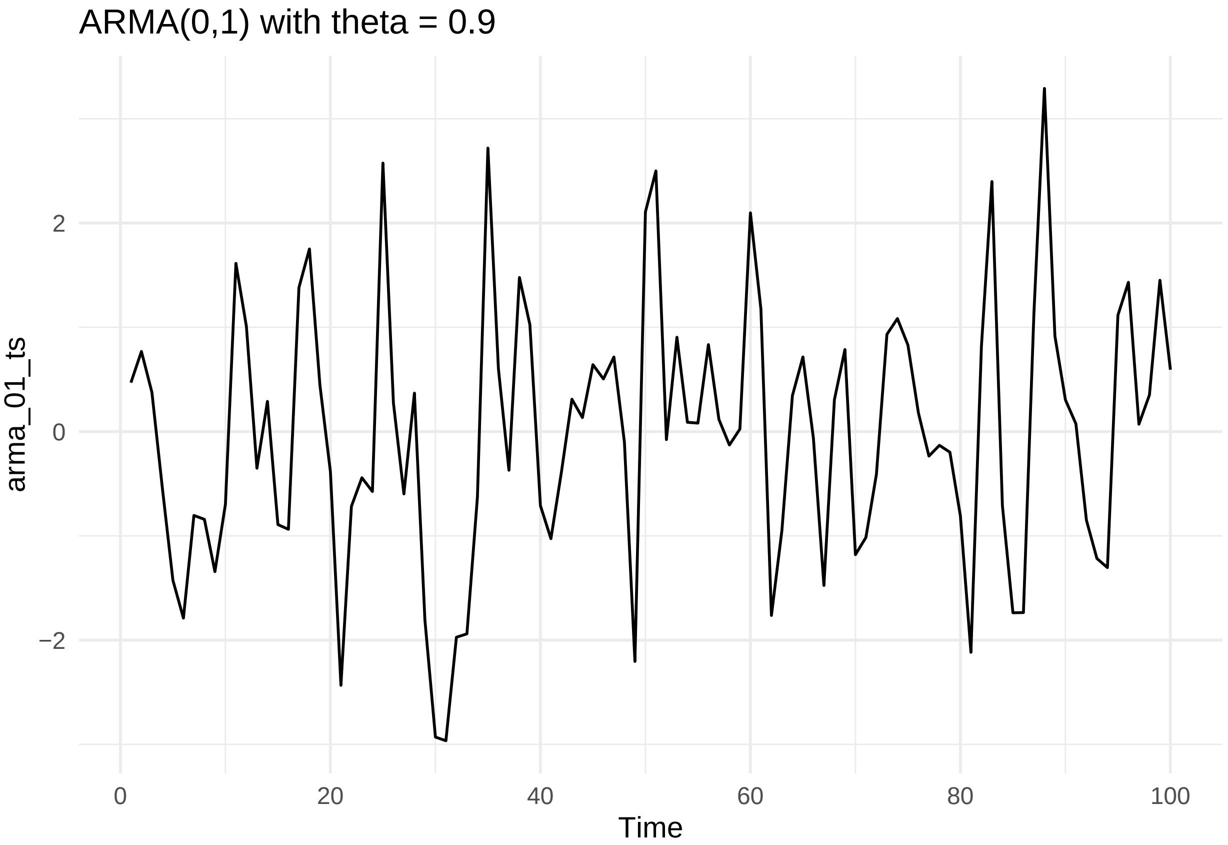
Recall that the non-seasonal ARIMA is described by three parameters ARIMA(p, d, q) where p is the order of the autoregressive component, d is the number of times the series need to be differenced to obtain stationarity and q is the order of the moving average component. If we don't need to difference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

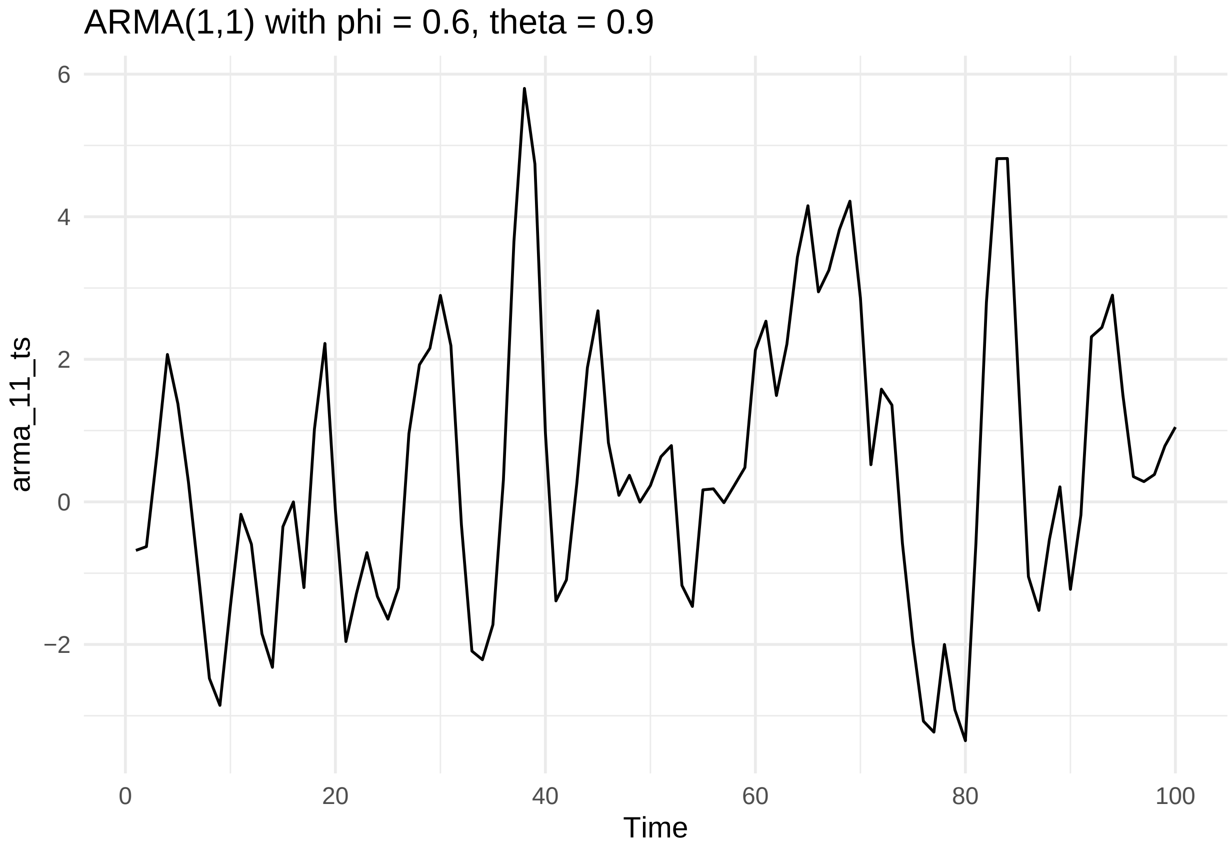
```r
library(ggplot2)
library(forecast)
library(cowplot)

set.seed(123)
n <- 100

# Simulate series
arma_10 <- arima.sim(model = list(ar = 0.6), n = n)
arma_01 <- arima.sim(model = list(ma = 0.9), n = n)
arma_11 <- arima.sim(model = list(ar = 0.6, ma = 0.9), n = n)

arma_10_ts <- ts(arma_10)
arma_01_ts <- ts(arma_01)
arma_11_ts <- ts(arma_11)

p1 <- forecast::autoplot(arma_10_ts) +
  ggtitle("ARMA(1,0) with phi = 0.6") +
  theme_minimal()

p2 <- forecast::autoplot(arma_01_ts) +
  ggtitle("ARMA(0,1) with theta = 0.9") +
  theme_minimal()

p3 <- forecast::autoplot(arma_11_ts) +
  ggtitle("ARMA(1,1) with phi = 0.6, theta = 0.9") +
  theme_minimal()

p1
```



```
p2
```

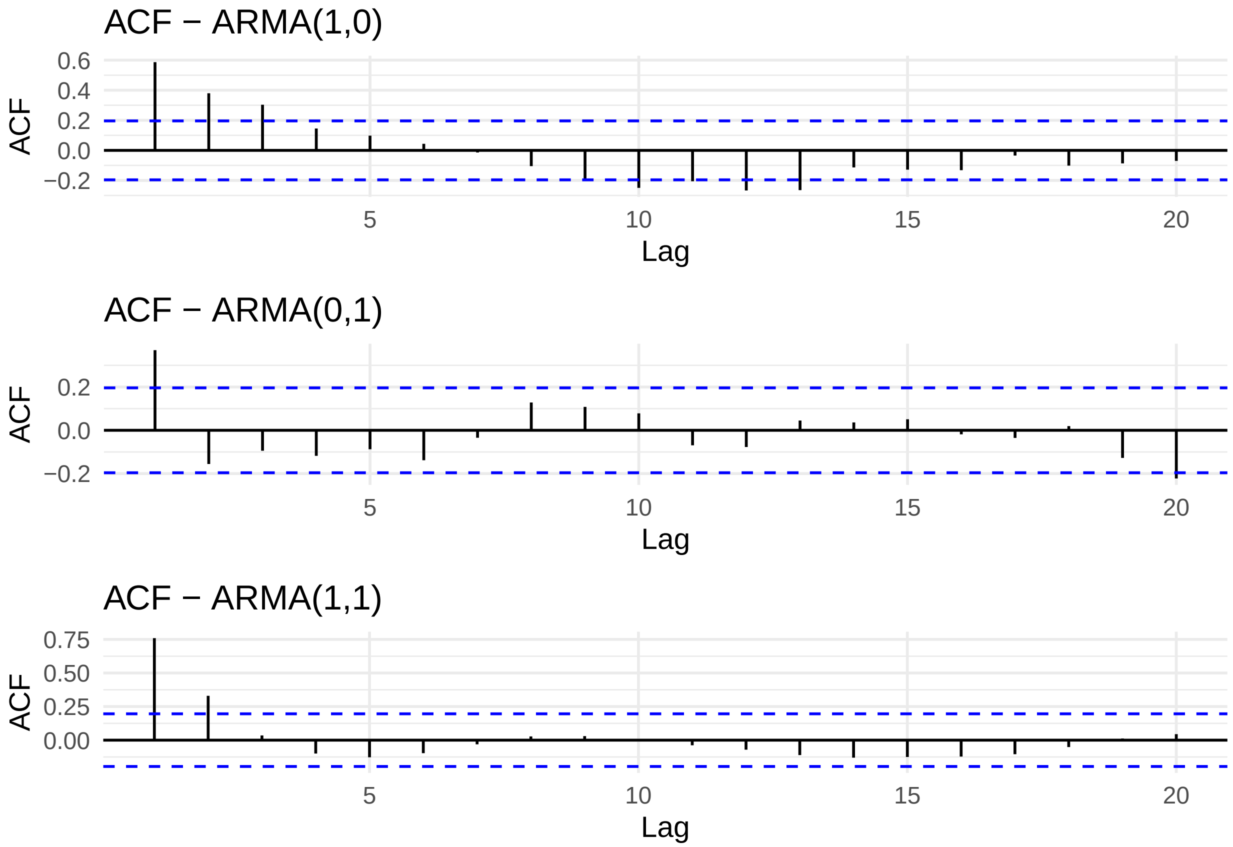


```
p3
```



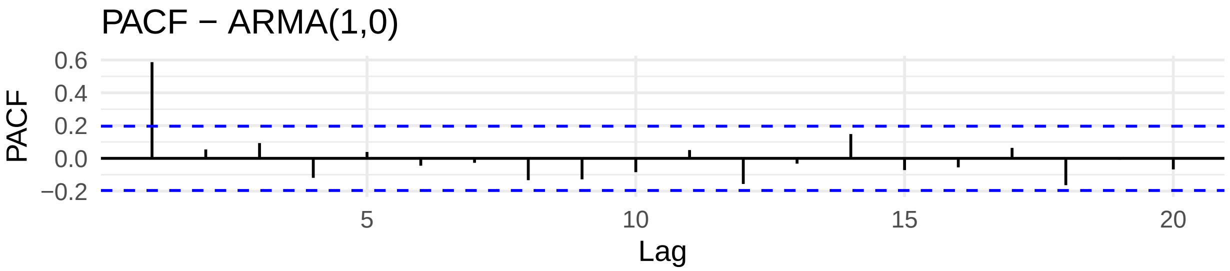
(b) Plot the sample ACF for each of these models in one window to facilitate comparison (Hint: use `cowplot::plot_grid()`).

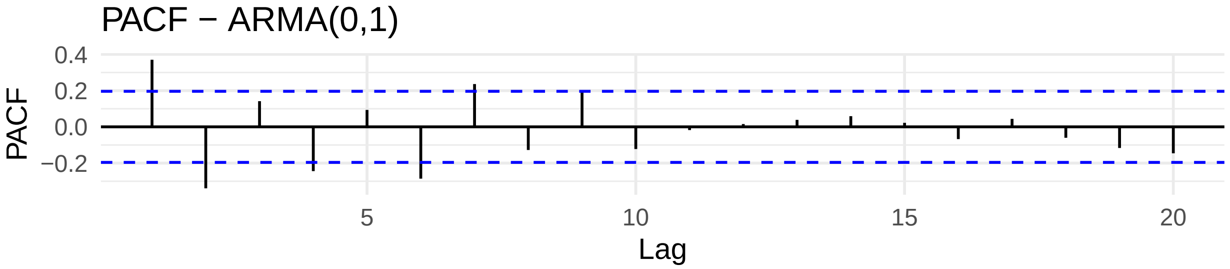
```
acf_10 <- ggAcf(arma_10_ts) +
  ggtitle("ACF - ARMA(1,0)") +
  theme_minimal()

acf_01 <- ggAcf(arma_01_ts) +
  ggtitle("ACF - ARMA(0,1)") +
  theme_minimal()

acf_11 <- ggAcf(arma_11_ts) +
  ggtitle("ACF - ARMA(1,1)") +
  theme_minimal()

cowplot::plot_grid(acf_10, acf_01, acf_11, ncol = 1)
```

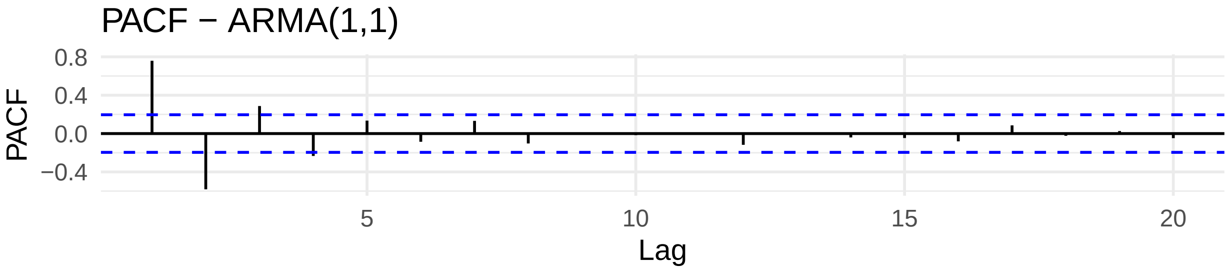


(c) Plot the sample PACF for each of these models in one window to facilitate comparison.

```r
pacf_10 <- ggPacf(arma_10_ts) +
  ggtitle("PACF - ARMA(1,0)") +
  theme_minimal()

pacf_01 <- ggPacf(arma_01_ts) +
  ggtitle("PACF - ARMA(0,1)") +
  theme_minimal()

pacf_11 <- ggPacf(arma_11_ts) +
  ggtitle("PACF - ARMA(1,1)") +
  theme_minimal()

cowplot::plot_grid(pacf_10, pacf_01, pacf_11, 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.

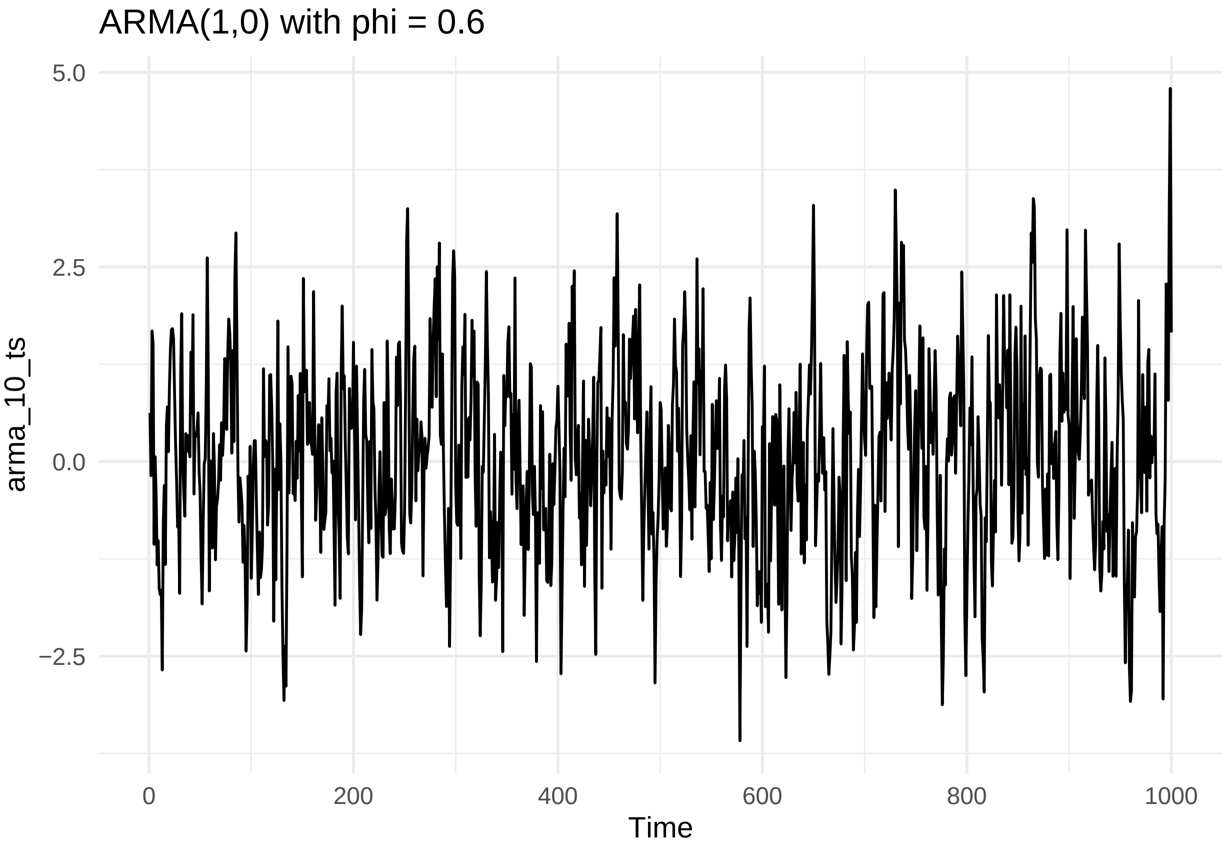
Answer: Yes, based on the ACF and PACF plots, it would generally be possible to correctly identify the three models, although in practice there may be some ambiguity due to sampling variability from only 100 observations. For the ARMA(1,0) model, which is an AR(1) process, the ACF shows a gradual exponential decay rather than a sharp cutoff, while the PACF displays a clear significant spike at lag 1 followed by values close to zero. From these plots, the correct identification would likely be AR(1). For the ARMA(0,1) model, which is an MA(1) process, the ACF exhibits a significant spike at lag 1 and then cuts off, with subsequent lags within the confidence bounds. Meanwhile, the PACF decays gradually instead of cutting off sharply. Therefore, this model could likely be identified correctly as MA(1). For the ARMA(1,1) model, both the ACF and PACF display gradual decay rather than a sharp cutoff at a specific lag. Neither function clearly truncates after lag 1. This joint tailing-off behavior is typical of mixed ARMA models, where both components are present. While the exact orders can be harder to determine visually compared to pure AR or MA processes, the absence of a sharp cutoff in either plot would suggest an ARMA model rather than a pure AR or MA process.

(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?

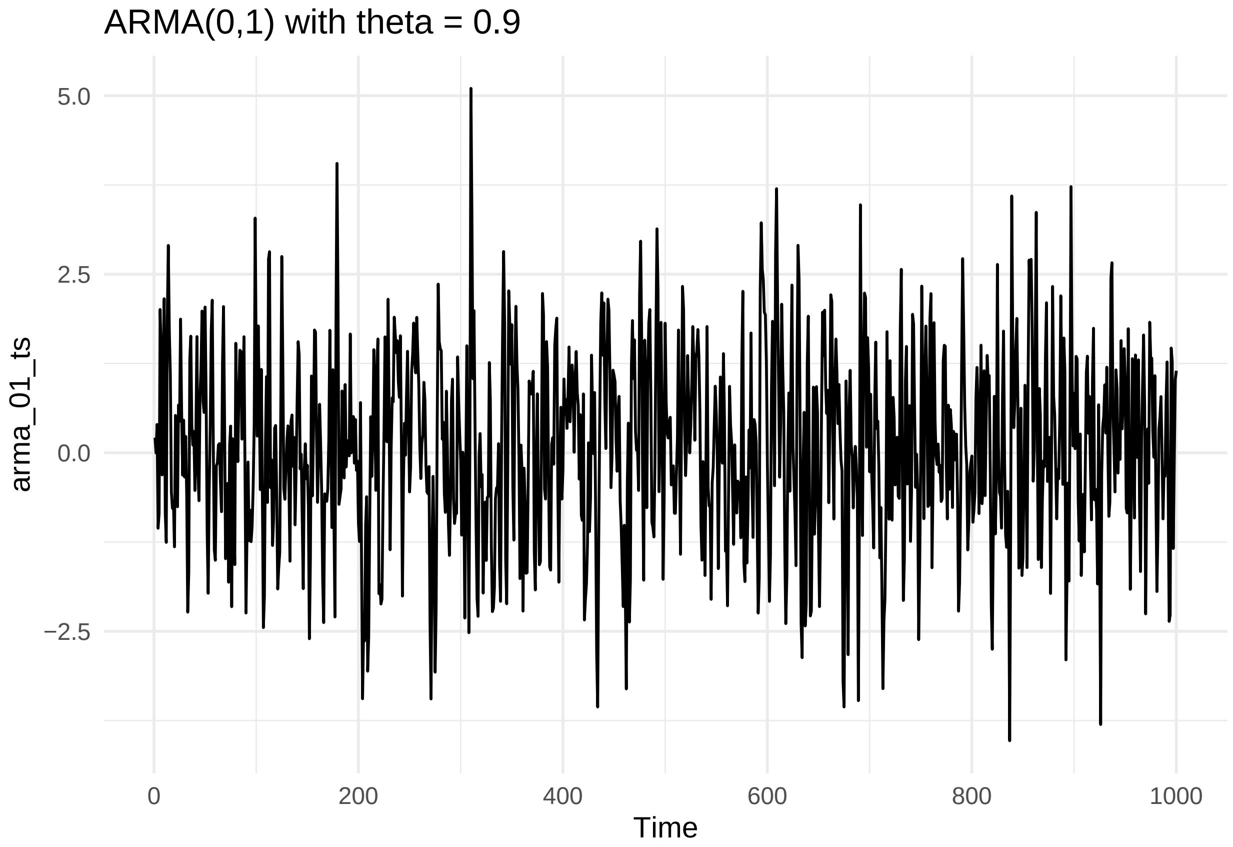
Answer: For ARMA(1,0) (i.e., a pure AR(1)), the PACF at lag 1 is estimating the first autoregressive coefficient, so it should be close to the true value. In the sample PACF plot, the lag 1 spike is indeed around that magnitude (though not exactly 0.6), which is expected because PACF values are sample estimates and will vary due to randomness, especially with a small sample size. For ARMA(1,1), the situation is different. The PACF at lag 1 is not equal to phi, because the presence of the MA(1) component changes the partial correlations. In an ARMA model, neither the ACF nor the PACF has a clean cutoff, and the PACF values are influenced by both. In short, they should approximately match for AR(1), but they should not be expected to match for ARMA(1,1).

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

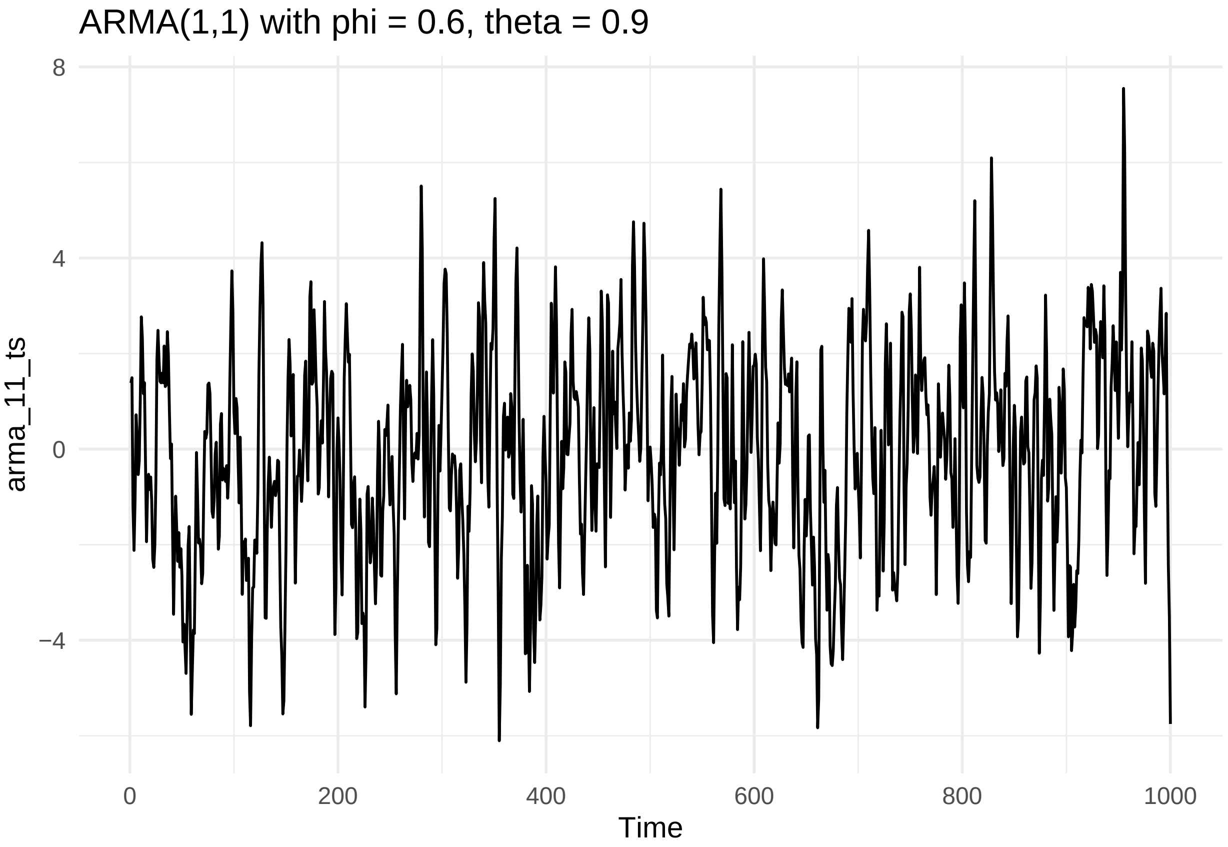
```r
set.seed(123)
n <- 1000

# Simulate series
arma_10 <- arima.sim(model = list(ar = 0.6), n = n)
arma_01 <- arima.sim(model = list(ma = 0.9), n = n)
arma_11 <- arima.sim(model = list(ar = 0.6, ma = 0.9), n = n)

arma_10_ts <- ts(arma_10)
arma_01_ts <- ts(arma_01)
arma_11_ts <- ts(arma_11)

p4 <- forecast::autoplot(arma_10_ts) +
  ggtitle("ARMA(1,0) with phi = 0.6") +
  theme_minimal()

p5 <- forecast::autoplot(arma_01_ts) +
  ggtitle("ARMA(0,1) with theta = 0.9") +
  theme_minimal()

p6 <- forecast::autoplot(arma_11_ts) +
  ggtitle("ARMA(1,1) with phi = 0.6, theta = 0.9") +
  theme_minimal()

p4
```


ARMA(1,0) with phi = 0.6
arma_10_ts
5.0
2.5
0.0
−2.5
0
200
400
600
800
1000
Time

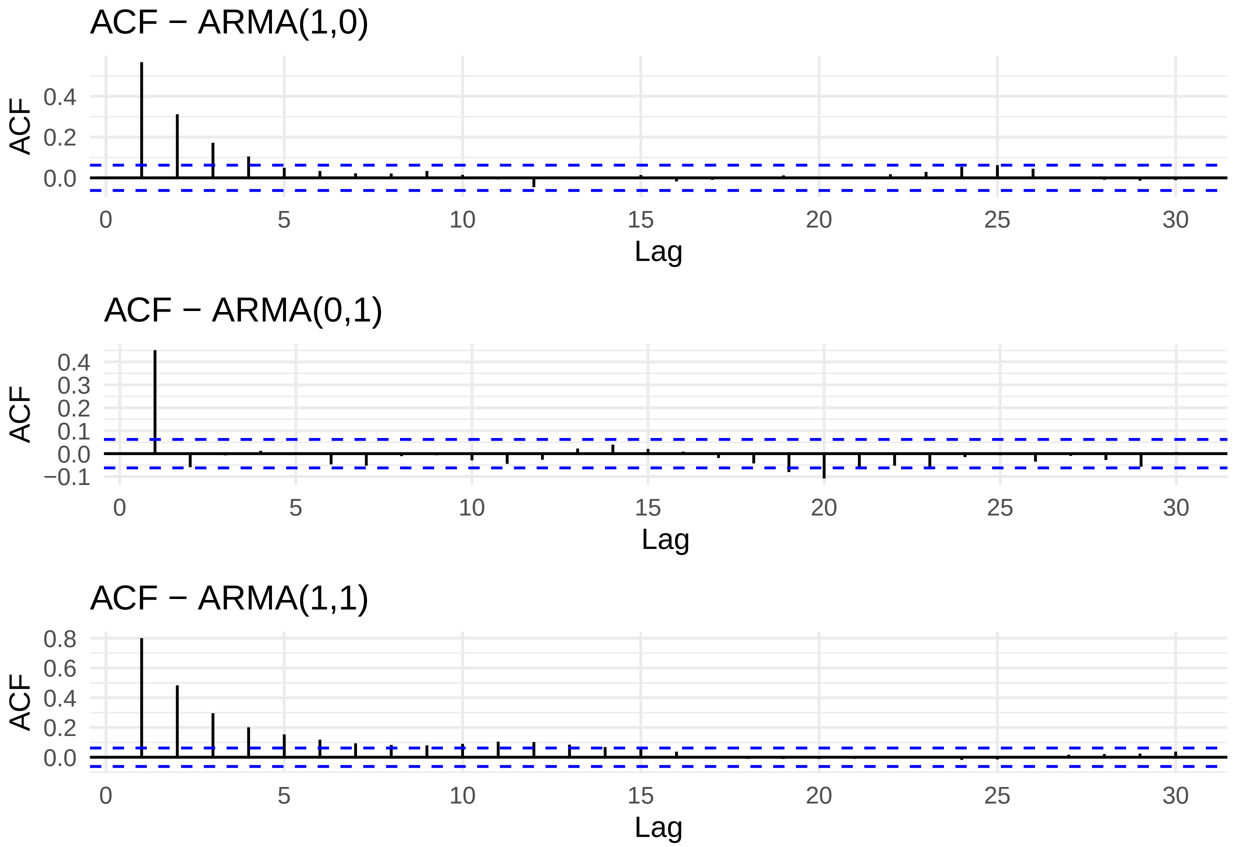
```
p5
```


ARMA(0,1) with theta = 0.9
arma_01_ts
5.0
2.5
0.0
−2.5
0
200
400
600
800
1000
Time

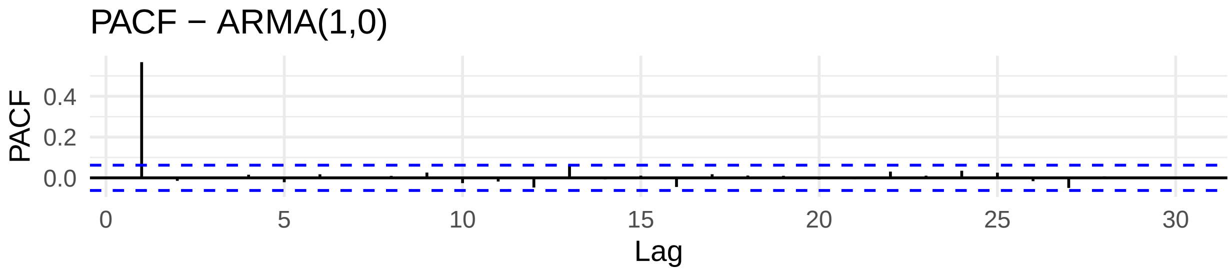
```
p6
```

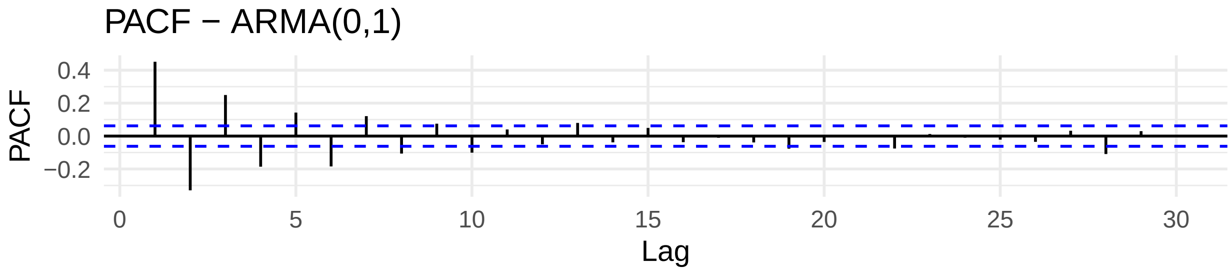


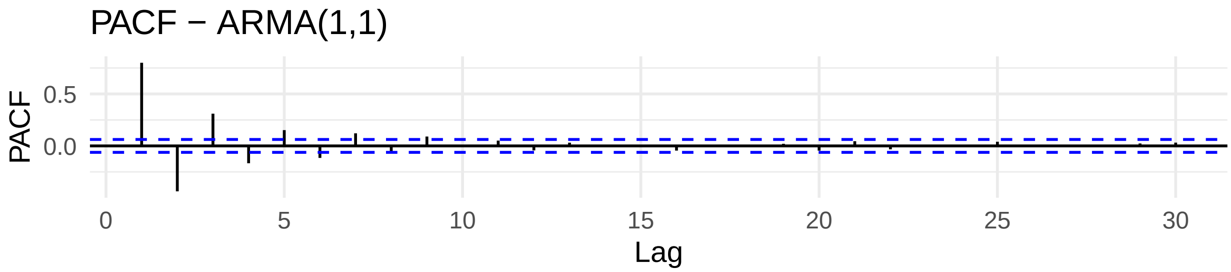
```
acf_10 <- ggAcf(arma_10_ts) + ggtitle("ACF - ARMA(1,0)") + theme_minimal()
acf_01 <- ggAcf(arma_01_ts) + ggtitle("ACF - ARMA(0,1)") + theme_minimal()
acf_11 <- ggAcf(arma_11_ts) + ggtitle("ACF - ARMA(1,1)") + theme_minimal()

cowplot::plot_grid(acf_10, acf_01, acf_11, ncol = 1)
```



```r
pacf_10 <- ggPacf(arma_10_ts) + ggtitle("PACF - ARMA(1,0)") + theme_minimal()
pacf_01 <- ggPacf(arma_01_ts) + ggtitle("PACF - ARMA(0,1)") + theme_minimal()
pacf_11 <- ggPacf(arma_11_ts) + ggtitle("PACF - ARMA(1,1)") + theme_minimal()

cowplot::plot_grid(pacf_10, pacf_01, pacf_11, ncol = 1)
```







```
pacf10_lag1 <- pacf(arma_10_ts, plot = FALSE)$acf[1]
pacf11_lag1 <- pacf(arma_11_ts, plot = FALSE)$acf[1]

cat("Lag-1 PACF for ARMA(1,0):", round(pacf10_lag1, 3), "\n")
```

```
## Lag-1 PACF for ARMA(1,0): 0.567
```

```
cat("Lag-1 PACF for ARMA(1,1):", round(pacf11_lag1, 3), "\n")
```

```
## Lag-1 PACF for ARMA(1,1): 0.8
```

```
cat("True phi =", 0.6, "\n")
```

```
## True phi = 0.6
```

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.

This equation has no differencing terms so d=0 and D=0. It includes a non seasonal AR term yt-1 and a seasonal term yt-12. It also includes a nonseasonal MA term at-1.

Thus, ARIMA(1,0,1)(1,0,0)_12

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

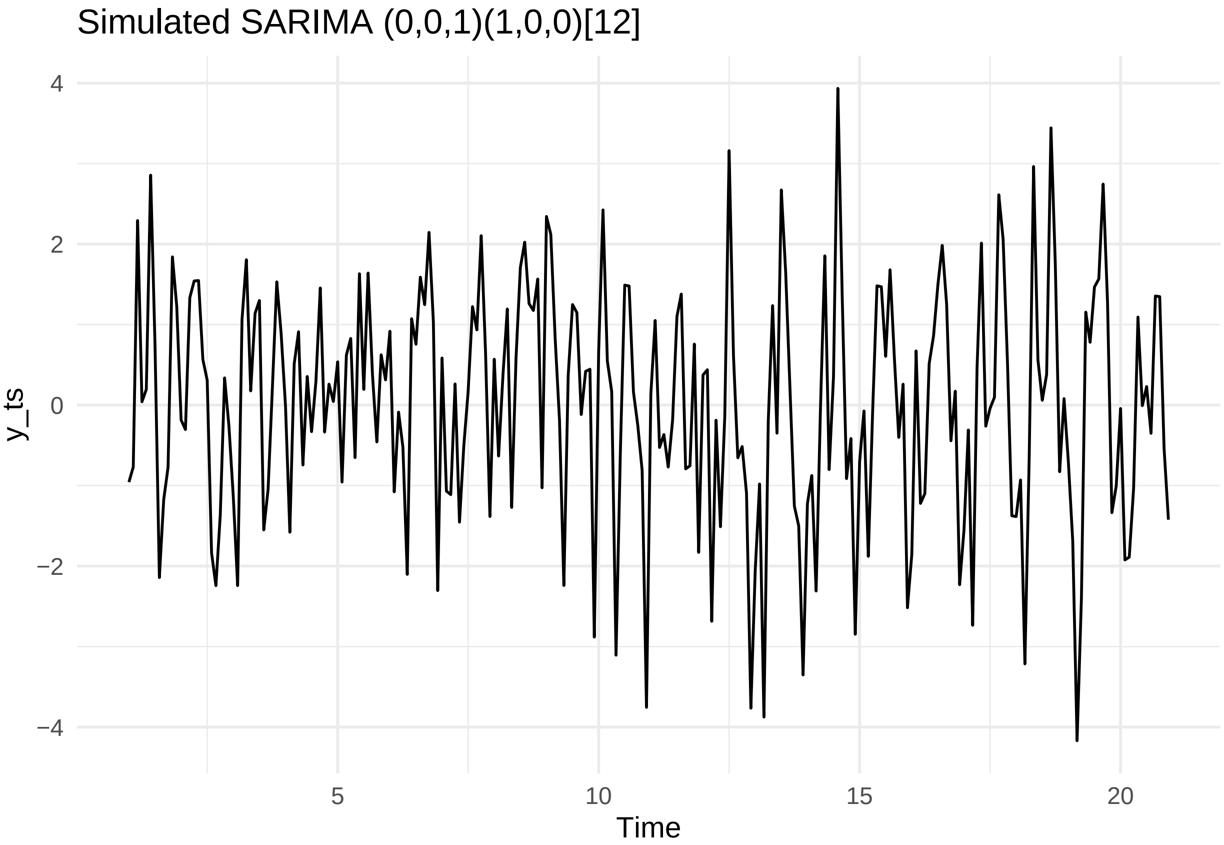
From the equation, the coefficients are:

Non-seasonal AR(1): = 0.7 Seasonal AR(1) at lag 12 = -0.25 Non-seasonal MA(1): =−0.1 Coefficient on at is 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
library(sarima)
library(forecast)
library(ggplot2)

set.seed(123)
n <- 240

y <- sim_sarima(
  n = n,
  model = list(
    ma = 0.5,
    sar = 0.8,
    nseasons = 12,
    sigma2 = 1
  )
)

y_ts <- ts(y, frequency = 12)

forecast::autoplot(y_ts) +
  ggtitle("Simulated SARIMA (0,0,1)(1,0,0)[12]") +
  theme_minimal()
```

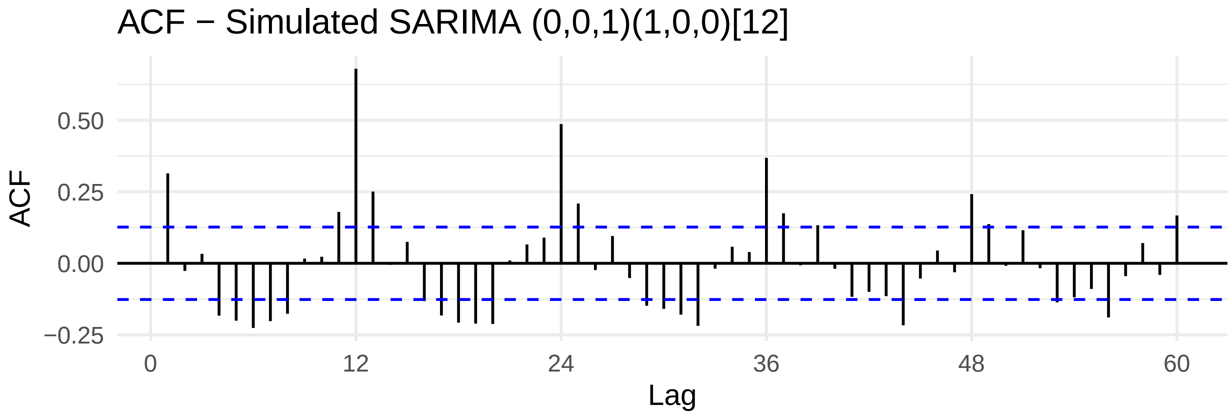


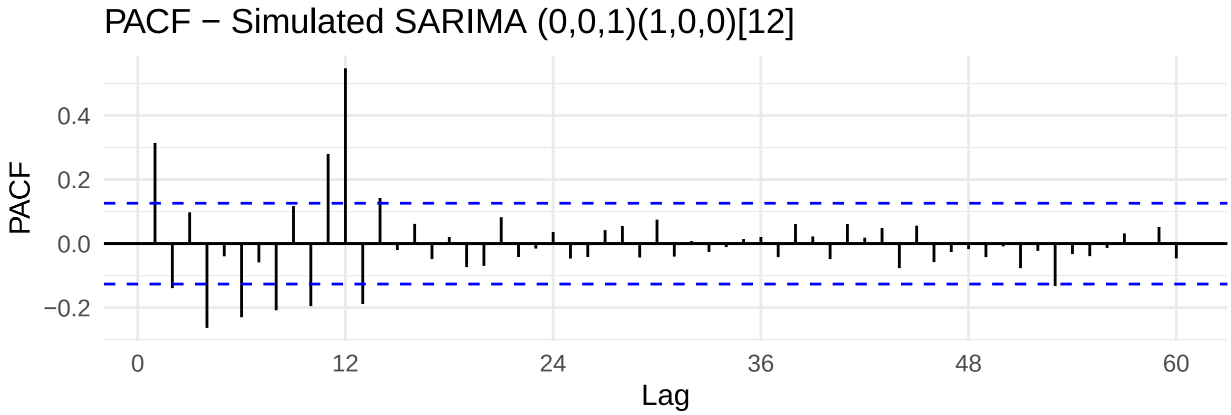
From the plotted series, the seasonal pattern is not visually very strong in the time-domain plot. Although the model contains a seasonal AR(1) term at lag 12, the randomness from the MA(1) component and white noise can obscure clear repeating cycles.

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
library(forecast)
library(cowplot)
library(ggplot2)

acf_plot <- ggAcf(y_ts, lag.max = 60) +
  ggtitle("ACF - Simulated SARIMA (0,0,1)(1,0,0)[12]") +
  theme_minimal()

pacf_plot <- ggPacf(y_ts, lag.max = 60) +
  ggtitle("PACF - Simulated SARIMA (0,0,1)(1,0,0)[12]") +
  theme_minimal()

cowplot::plot_grid(acf_plot, pacf_plot, ncol = 1)
```





The ACF and PACF plots represent the simulated SARIMA model relatively well.

At the seasonal level, the ACF shows strong and significant spikes at lags 12, 24, 36, 48, and 60, which clearly indicates seasonal dependence with period 12. The PACF shows a very strong spike at lag 12 and then much smaller values afterward.

At the non-seasonal level, there is noticeable short-lag behavior around lag 1 in the ACF, consistent with the presence of an MA(1) component. The ACF does not display a long autoregressive decay at low lags, which supports the interpretation of a non-seasonal MA(1) rather than AR(1). The PACF at low lags does not show a sharp cutoff at lag 1, which is also consistent with an MA process.