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

Assignment 6 - Due date 02/28/24

David Robinson

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(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 4.3.2

## 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     1.1.3     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(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```r
#install.packages("sarima")
library(sarima)
```

```
## Warning: package 'sarima' was built under R version 4.3.2
```

```
## 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: AR stands for Auto Regressive and the p in parentheses is the order of the AR process – so this is a second order AR process, which means that the current value depends on its own 2 previous values. ACF – the ACF will decay exponentially with time. PACF – the PACF will identify the order of the AR model.

- MA(1)

  Answer: MA stands for Moving Average and the p in parentheses is the order of the MA process – so this is a first order AR process, which means that the current value depends on its own 1 previous values. ACF – the ACF will identify the order of the MA model. PACF – the PACF will decay exponentially.

Q2

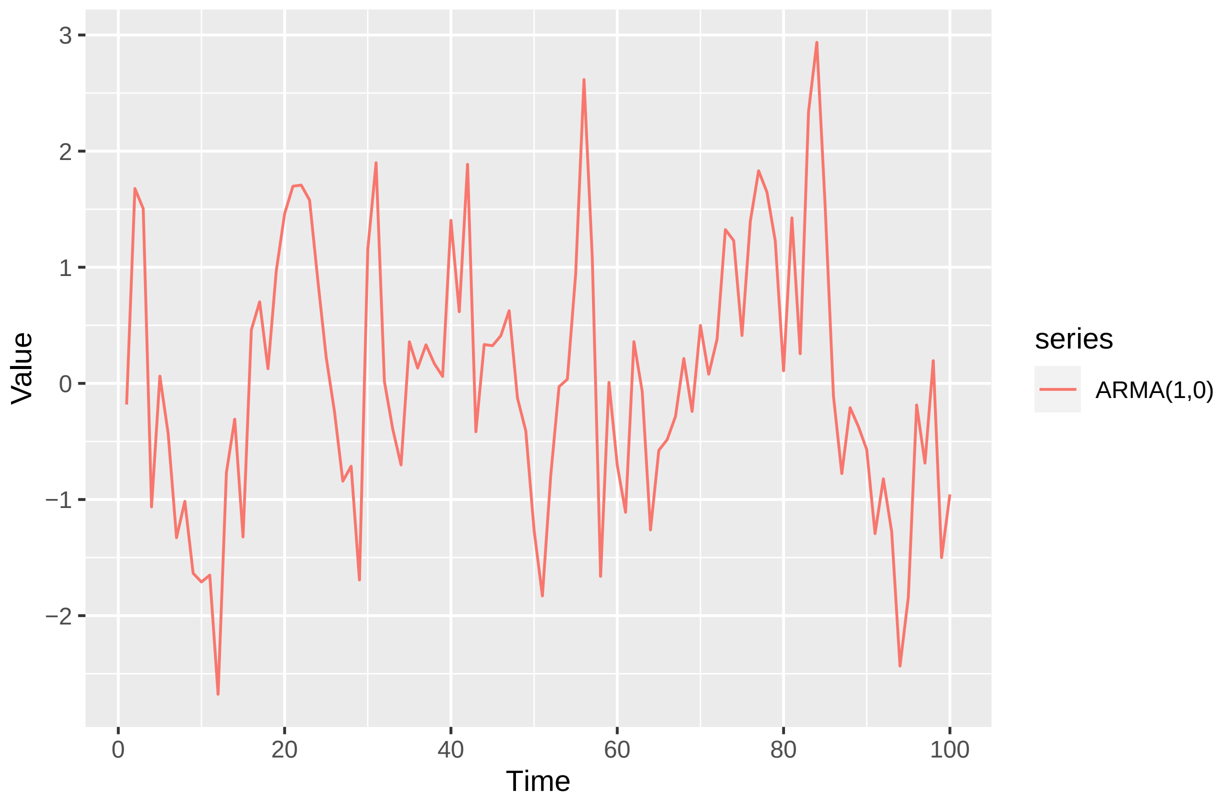
Recall that the non-seasonal ARIMA is described by three parameters ARIMA(p, d, q) where p is the order of the autoregressive component, d is the number of times the series need to be differenced to obtain stationarity and q is the order of the moving average component. If we don't need to difference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

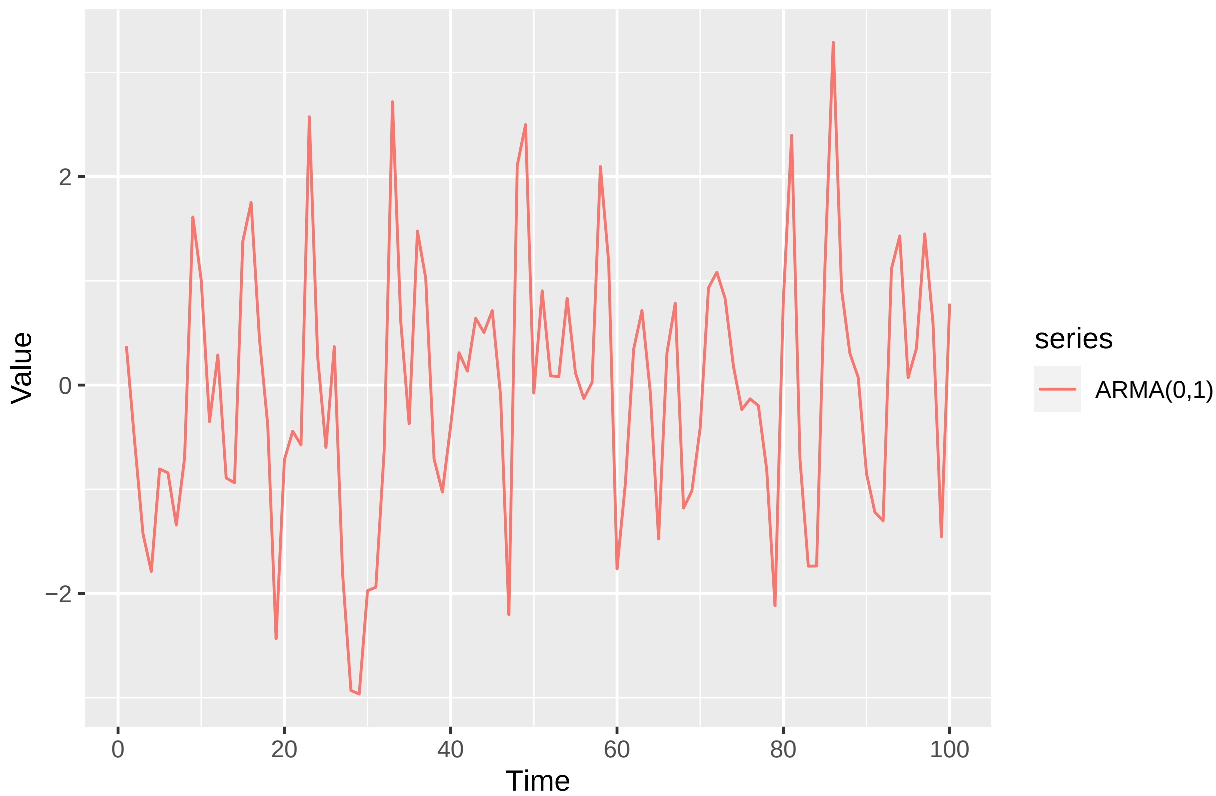
```
# Set seed for reproducibility
set.seed(123)

# Define parameters
n <- 100  # Number of observations
phi <- 0.6  # AR coefficient
theta <- 0.9  # MA coefficient

# Generate data for ARMA(1,0)
arma_10 <- arima.sim(model = list(ar = phi, ma = 0), n = n)

# Generate data for ARMA(0,1)
arma_01 <- arima.sim(model = list(ar = 0, ma = theta), n = n)
```

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

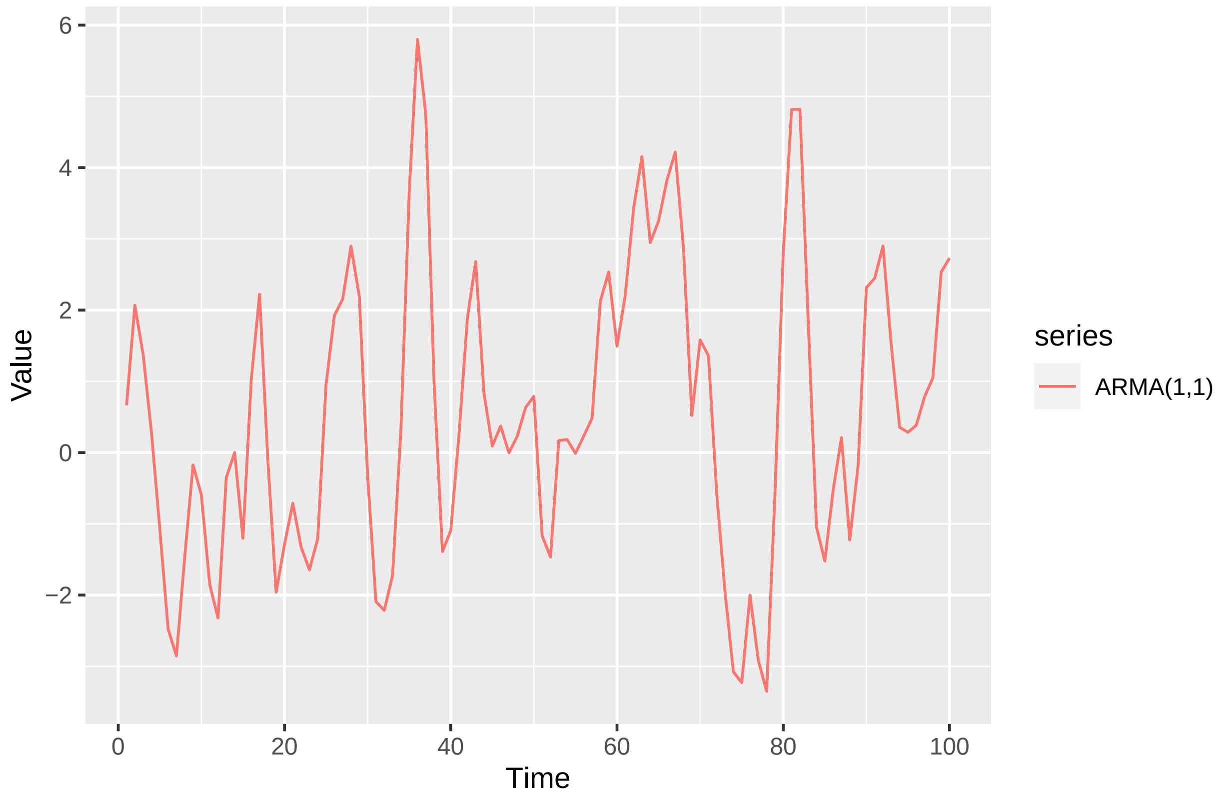
```
# Generate data for ARMA(1,1)
arma_11 <- arima.sim(model = list(ar = phi, ma = theta), n = n)

# Plot the generated series
autoplot(arma_10, series = "ARMA(1,0)", xlab = "Time", ylab = "Value")
```




```
autoplot(arma_01, series = "ARMA(0,1)", xlab = "Time", ylab = "Value")
```



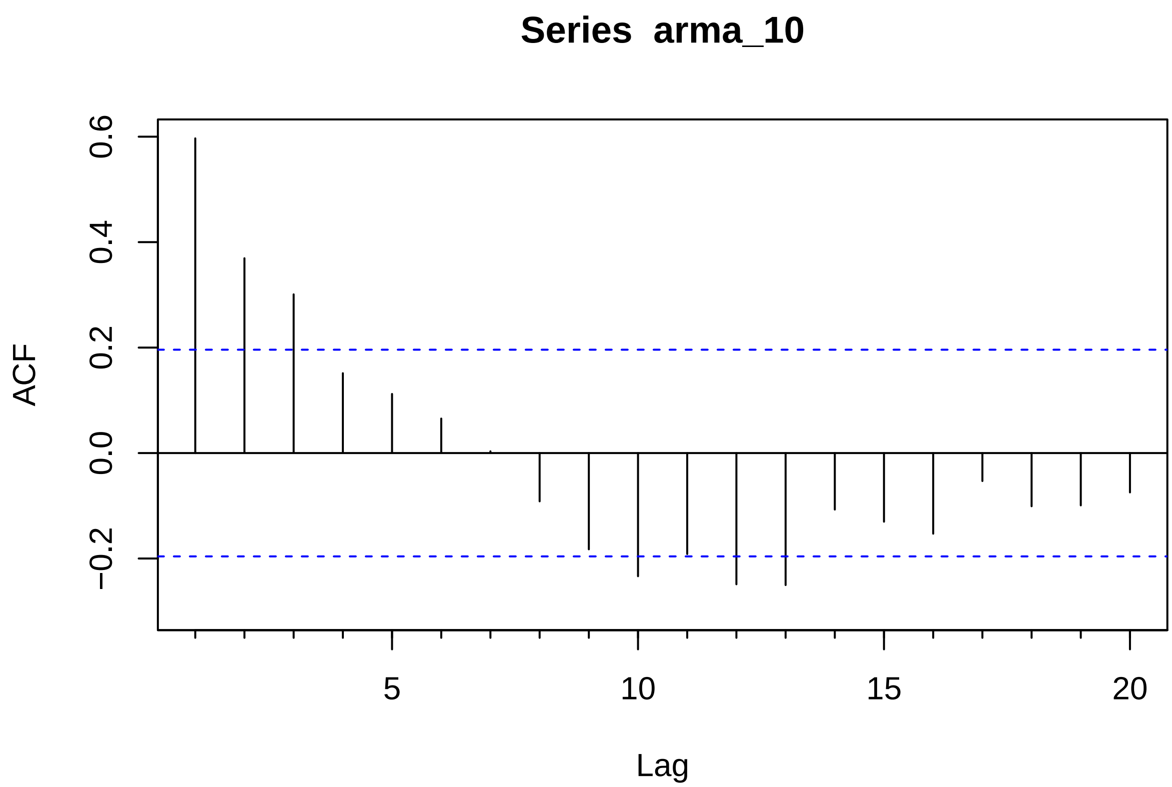

```r
autoplot(arma_11, series = "ARMA(1,1)", xlab = "Time", ylab = "Value")
```



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

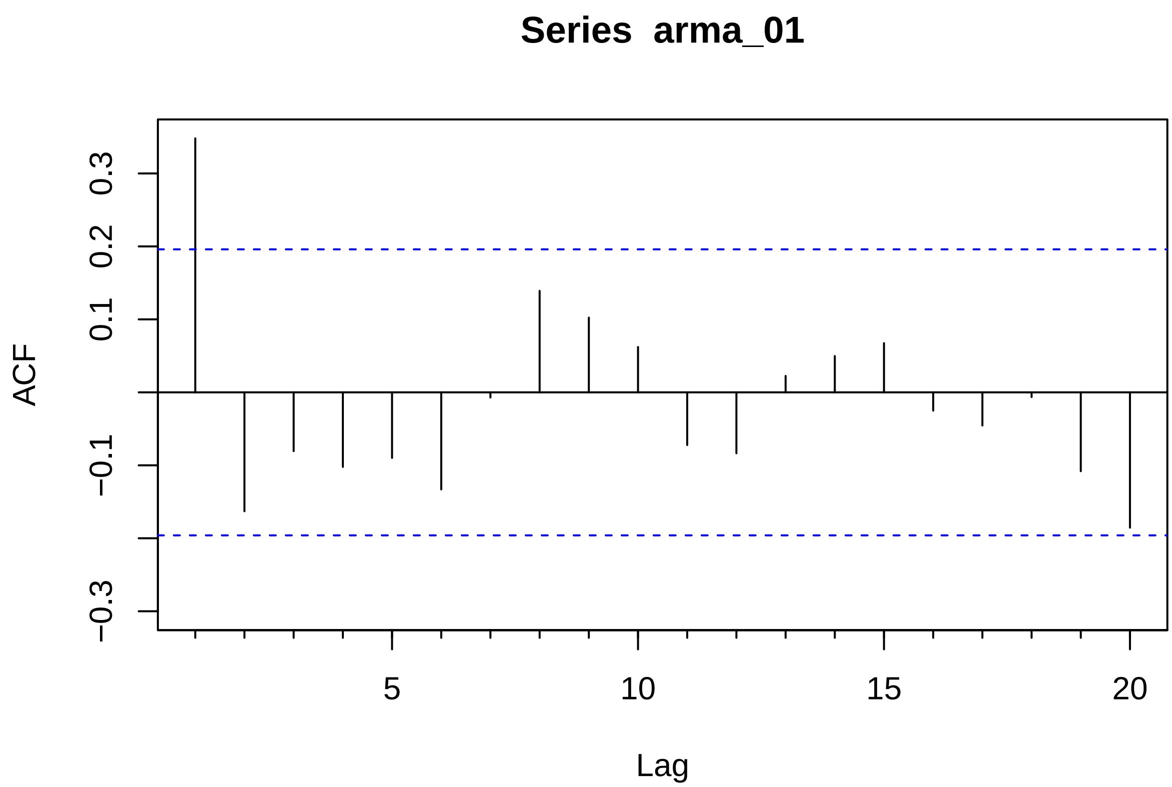
```r
# Compute ACF for each series
acf_arma_10 <- autoplot(Acf(arma_10), main = "ACF for ARMA(1,0)")
```

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



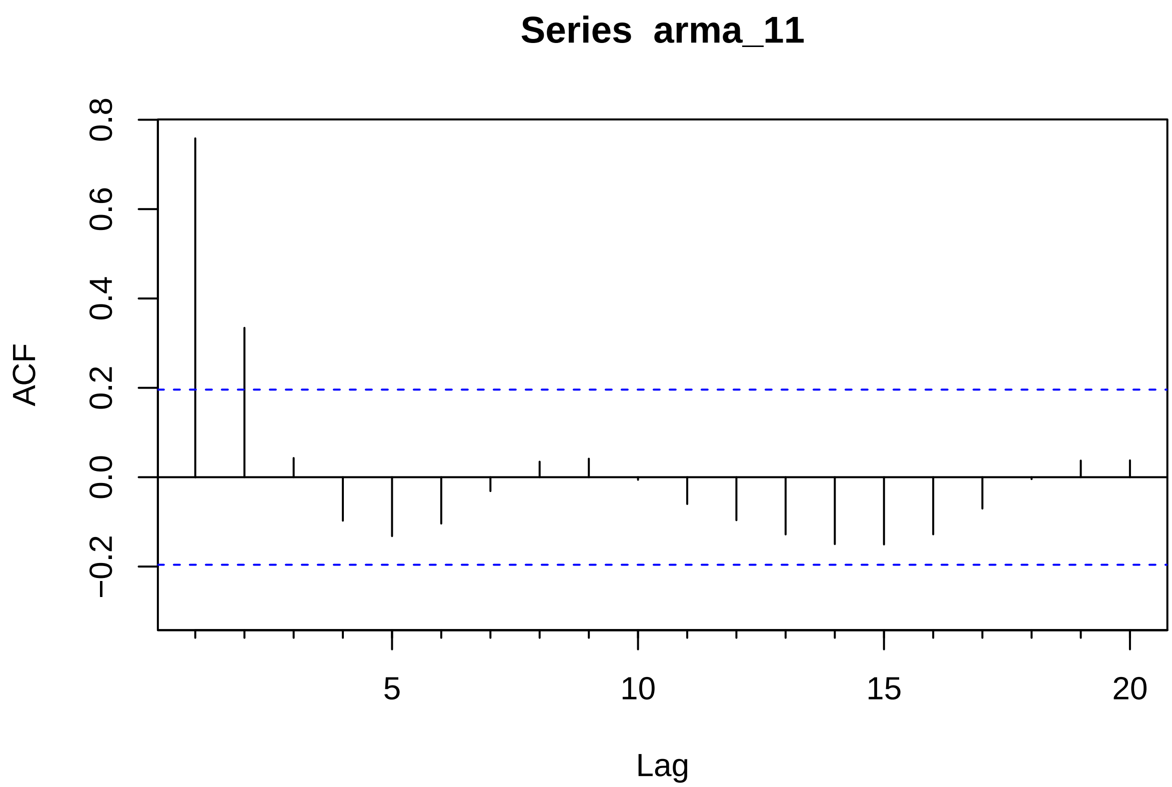

```r
acf_arma_01 <- autoplot(Acf(arma_01), main = "ACF for ARMA(0,1)")
```

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

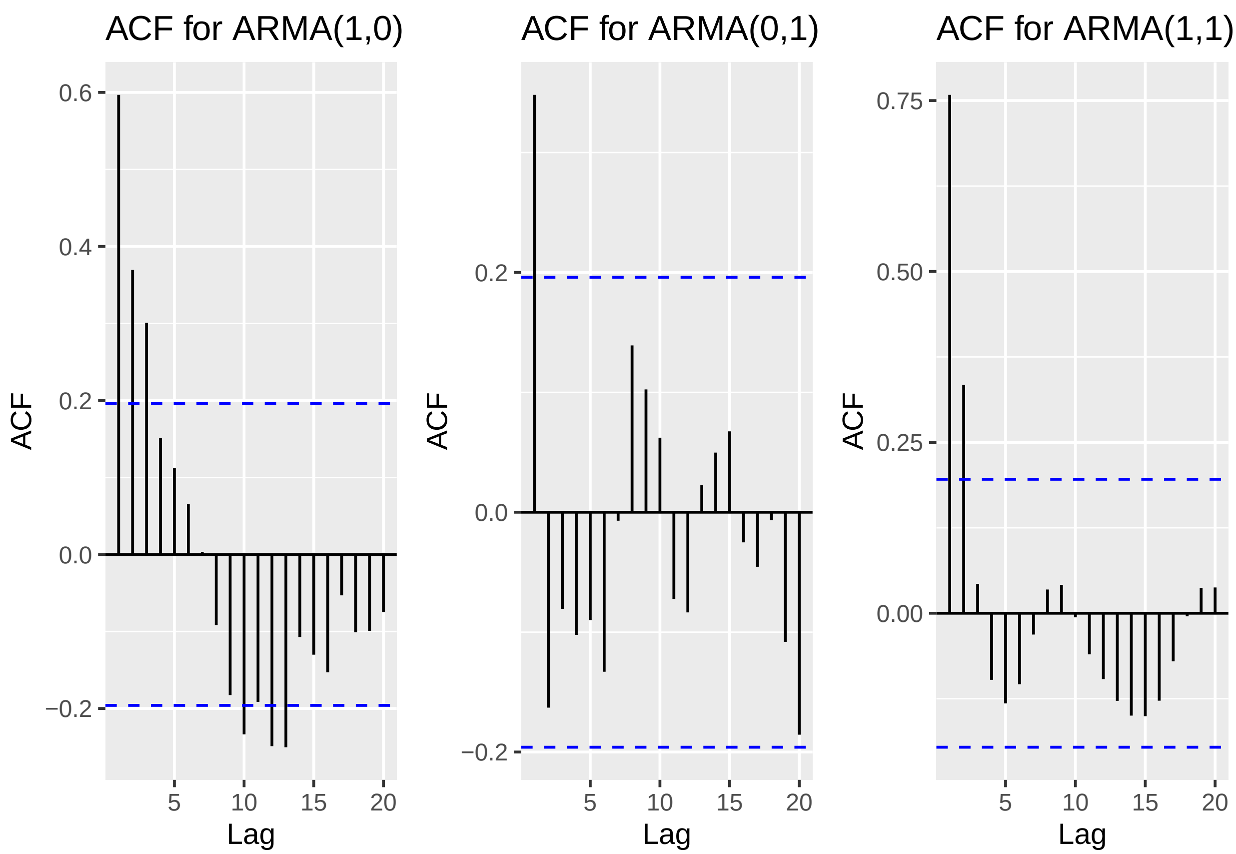



```r
acf_arma_11 <- autoplot(Acf(arma_11), main = "ACF for ARMA(1,1)")
```

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



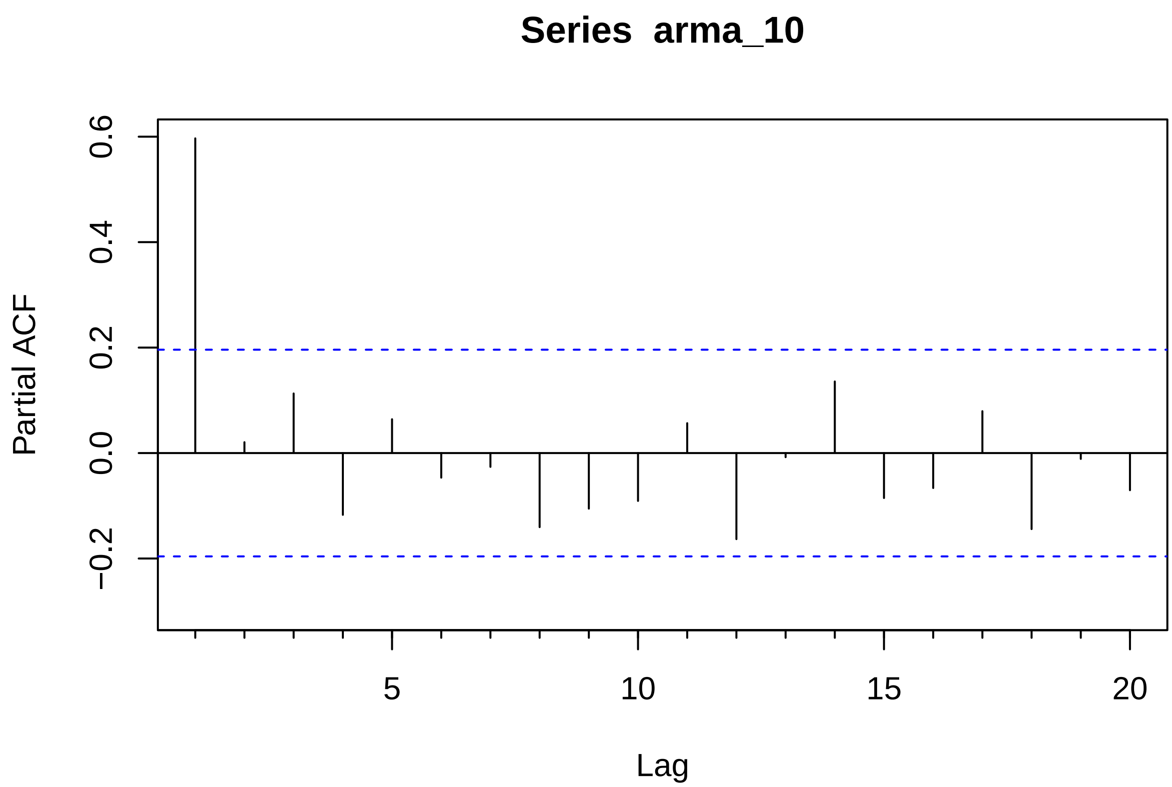

```r
# Plot ACF for each series in one window
plot_grid(acf_arma_10, acf_arma_01, acf_arma_11, ncol = 3)
```



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

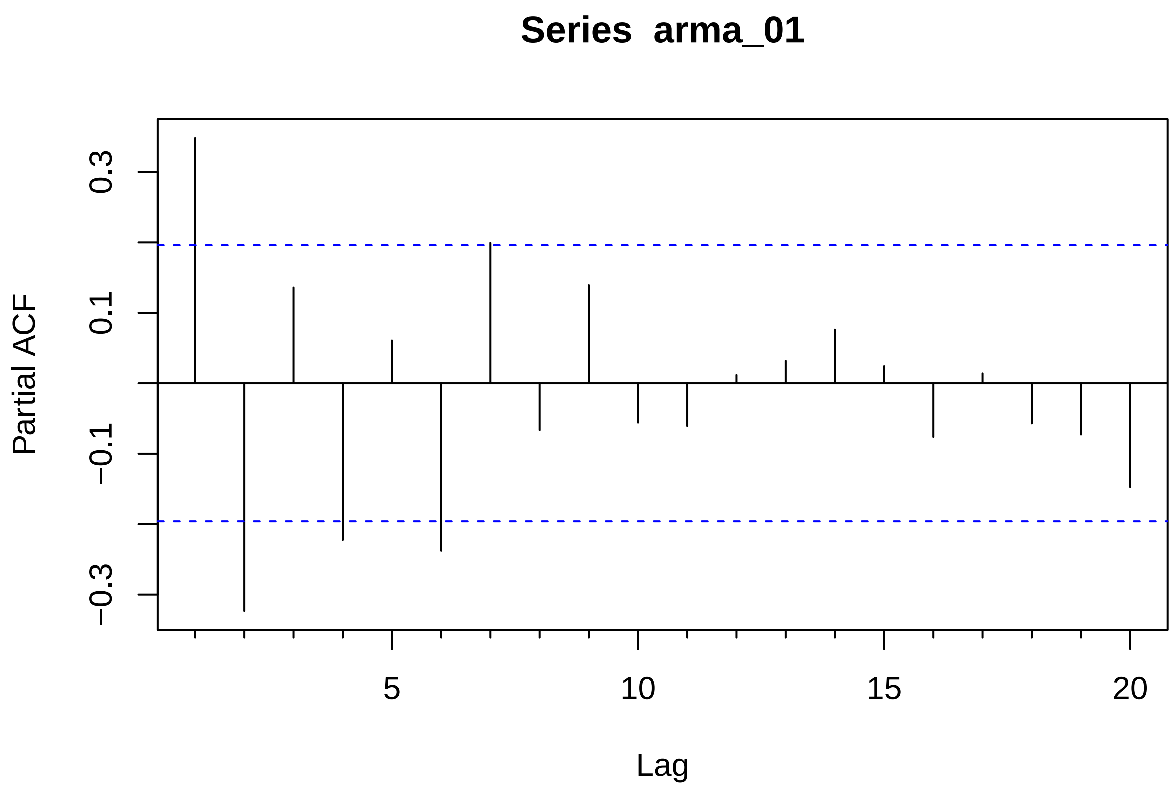
```
# Compute PACF for each series
pacf_arma_10 <- autoplot(Pacf(arma_10), main = "PACF for ARMA(1,0)")
```

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



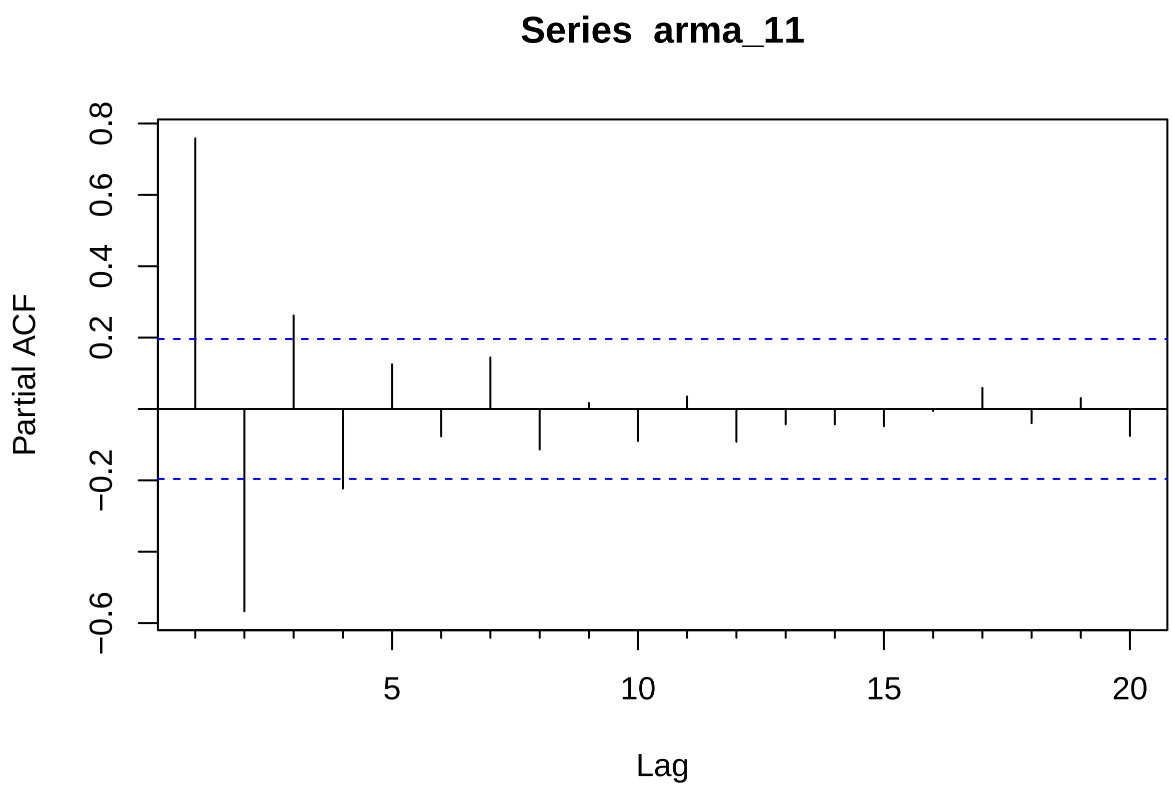

```
pacf_arma_01 <- autoplot(Pacf(arma_01), main = "PACF for ARMA(0,1)")
```

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

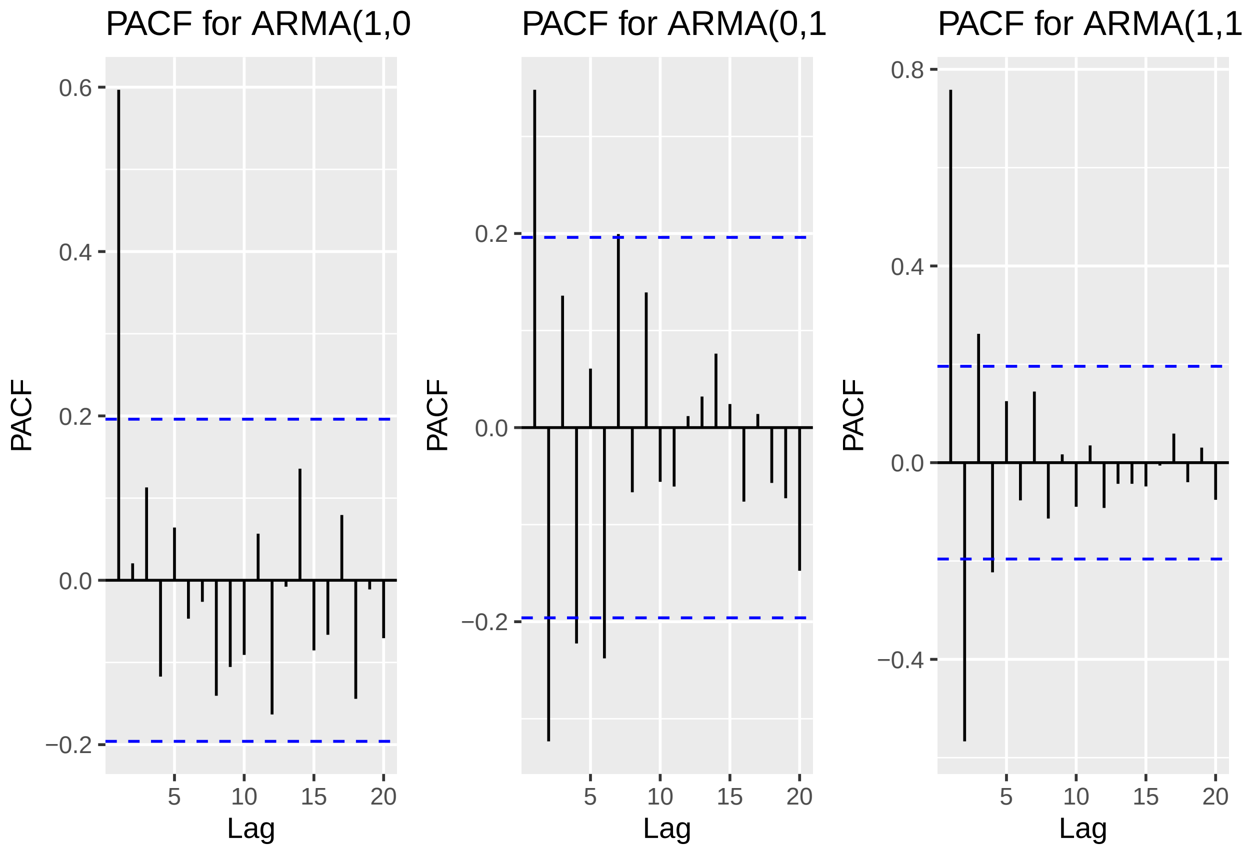



```
pacf_arma_11 <- autoplot(Pacf(arma_11), main = "PACF for ARMA(1,1)")
```

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

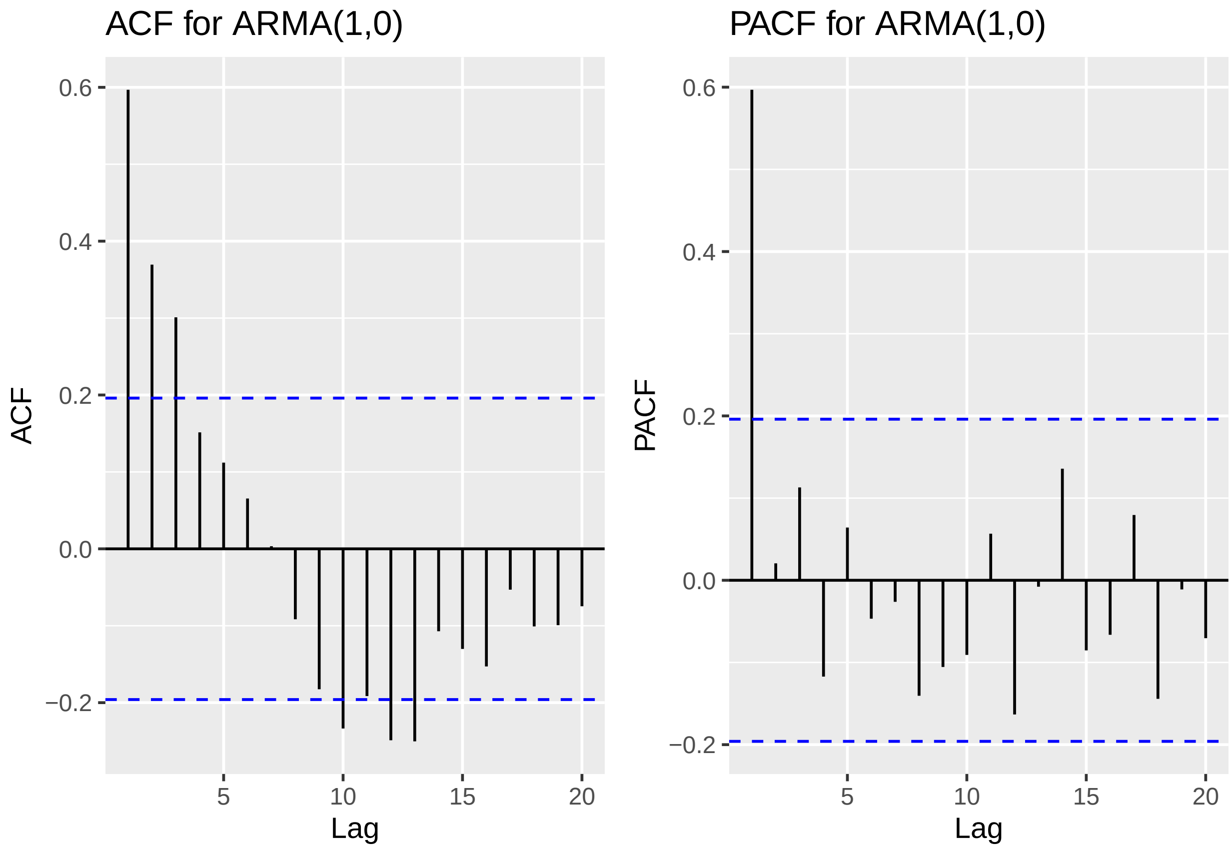



```
# Plot PACF for each series in one window
plot_grid(pacf_arma_10, pacf_arma_01, pacf_arma_11, ncol = 3)
```

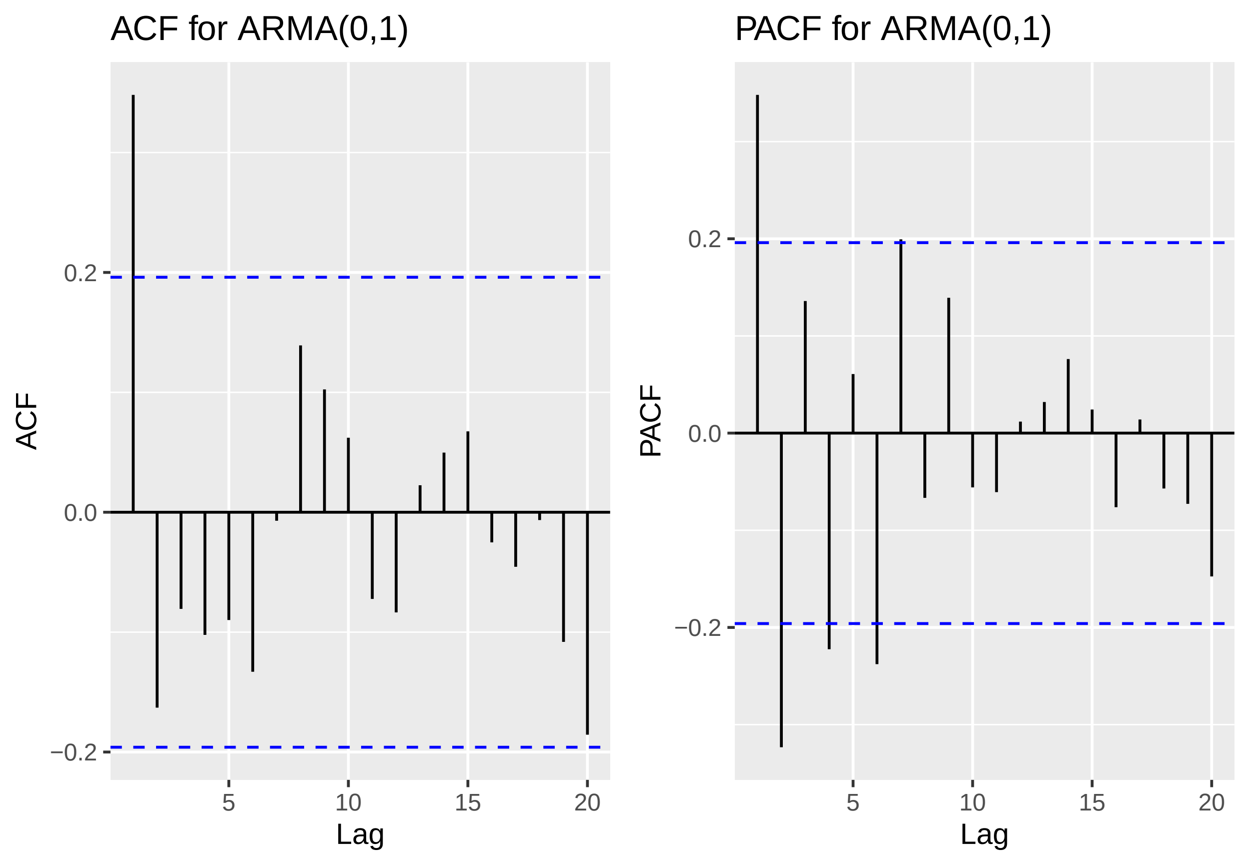


(d) Look at the ACFs and PACFs. Imagine you had these plots for a data set and you were asked to identify the model, i.e., is it AR, MA or ARMA and the order of each component. Would you be able identify them correctly? Explain your answer.

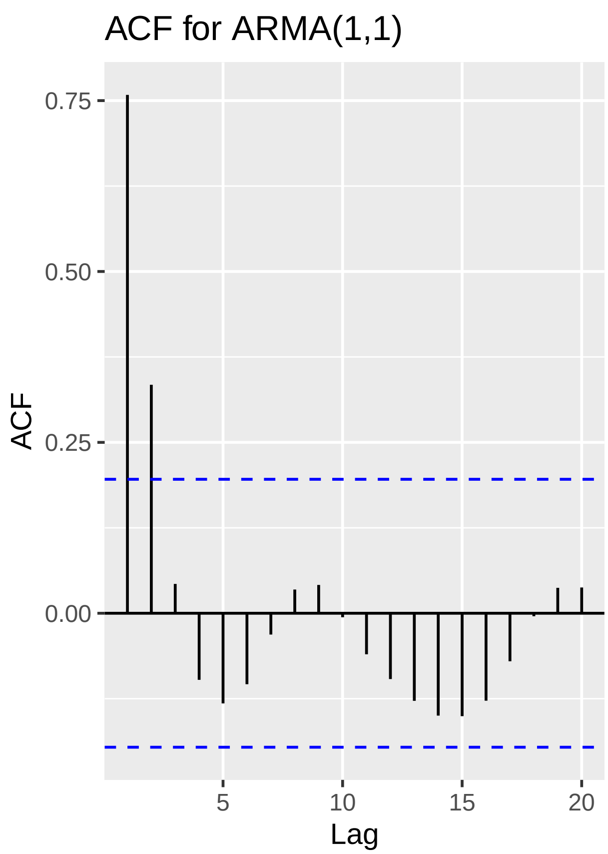
```
# Arrange ACF and PACF pairs for each series
plot_grid(acf_arma_10, pacf_arma_10, ncol = 2)
```

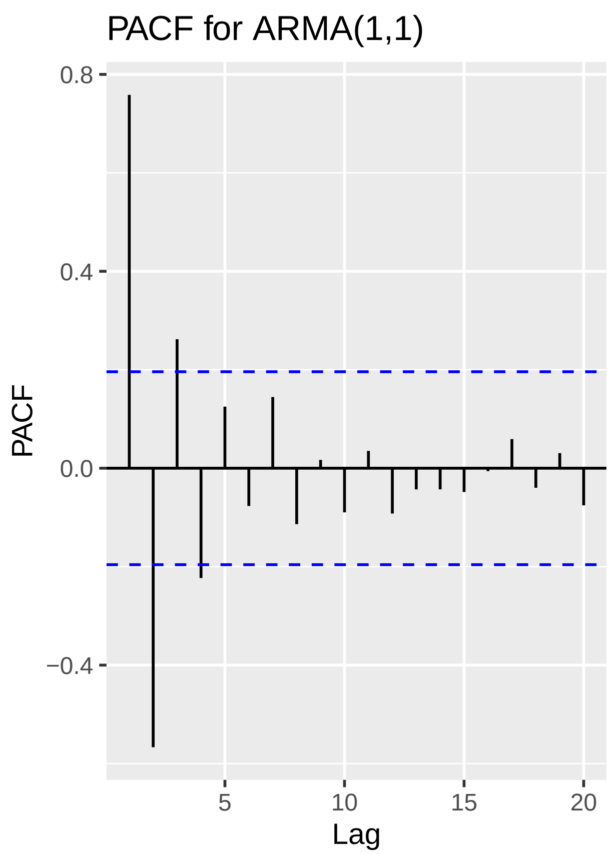



```
plot_grid(acf_arma_01, pacf_arma_01, ncol = 2)
```




```r
plot_grid(acf_arma_11, pacf_arma_11, ncol = 2)
```





Answer: For ARMA(1,0), consistent with Q1 for AR models, the ACF for this model decays exponentially with time and the PACF tells us the order of the process. For ARMA(0,1), consistent with Q1 for MA models, the PACF for this model decays exponentially with time and the ACF tells us the order of the process. For ARMA(1,1), consistent with Q1, both significant spikes at the initial lags and exponential decay afterward suggest an ARMA model. It's a bit tough since we know how these plots are generated instead of just looking at them in isolation and identifying the model types... we'd be able to create a reasonable guess, but likely would not be able to say definitively just based on the ACF and PACF in isolation.

(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), the PACF value R computed at lag 1 does match. For ARMA(1,1), the PACF value computed at lag 1 exceeds 0.6 and is closer to 0.8. We know that both the ACF and the PACF are the result of superimposing the AR and MA properties. In the PACF, initial values are dependent on the AR followed by the decay due to the MA part. Therefore, the ARMA(1,0) lag 1 correlation coefficient should match the value of phi. The ARMA (1,1) lag 1 correlation coefficient should also roughly match, but is subject to the exponential decay of the MA component, so it should be a bit higher.

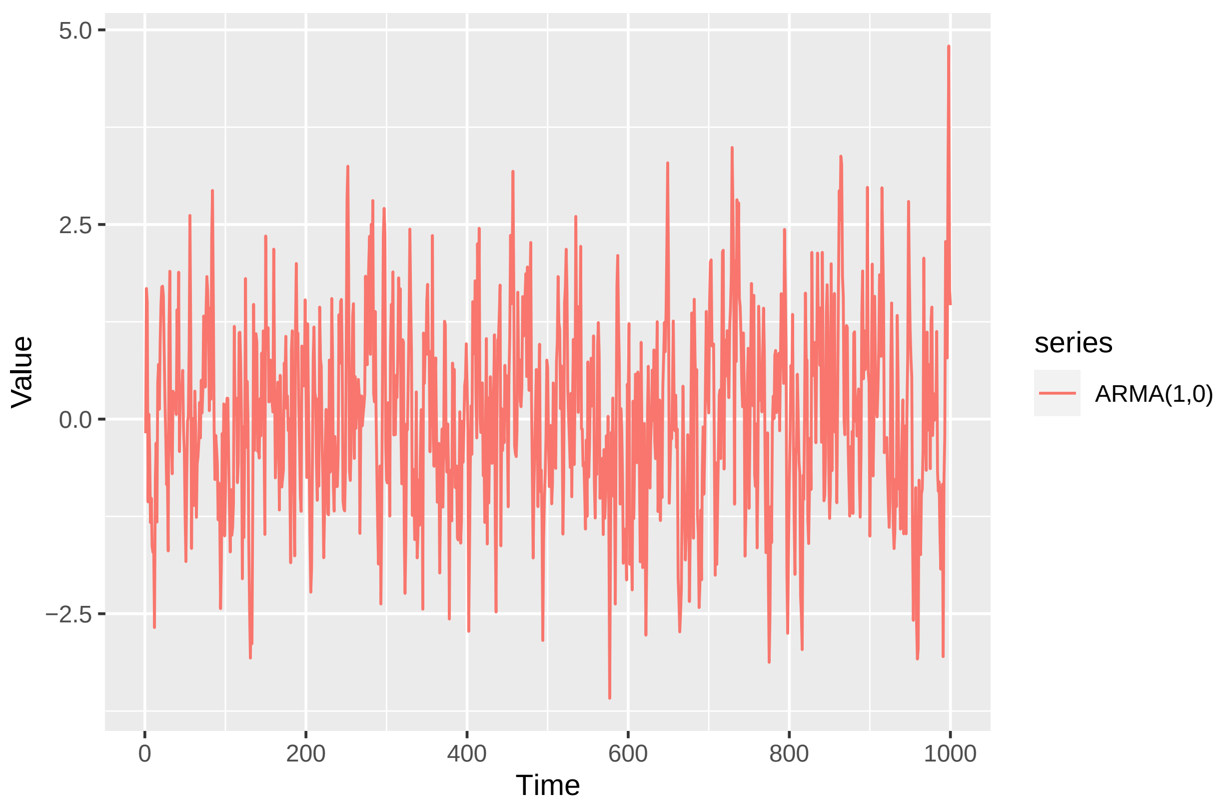
(f) Increase number of observations to $n = 1000$ and repeat parts (b)-(e).

```
#(a)
# Set seed for reproducibility
```

```r
set.seed(123)

# Define parameters
n <- 1000   # Number of observations
phi <- 0.6  # AR coefficient
theta <- 0.9  # MA coefficient

# Generate data for ARMA(1,0)
arma_10 <- arima.sim(model = list(ar = phi, ma = 0), n = n)

# Generate data for ARMA(0,1)
arma_01 <- arima.sim(model = list(ar = 0, ma = theta), n = n)
```

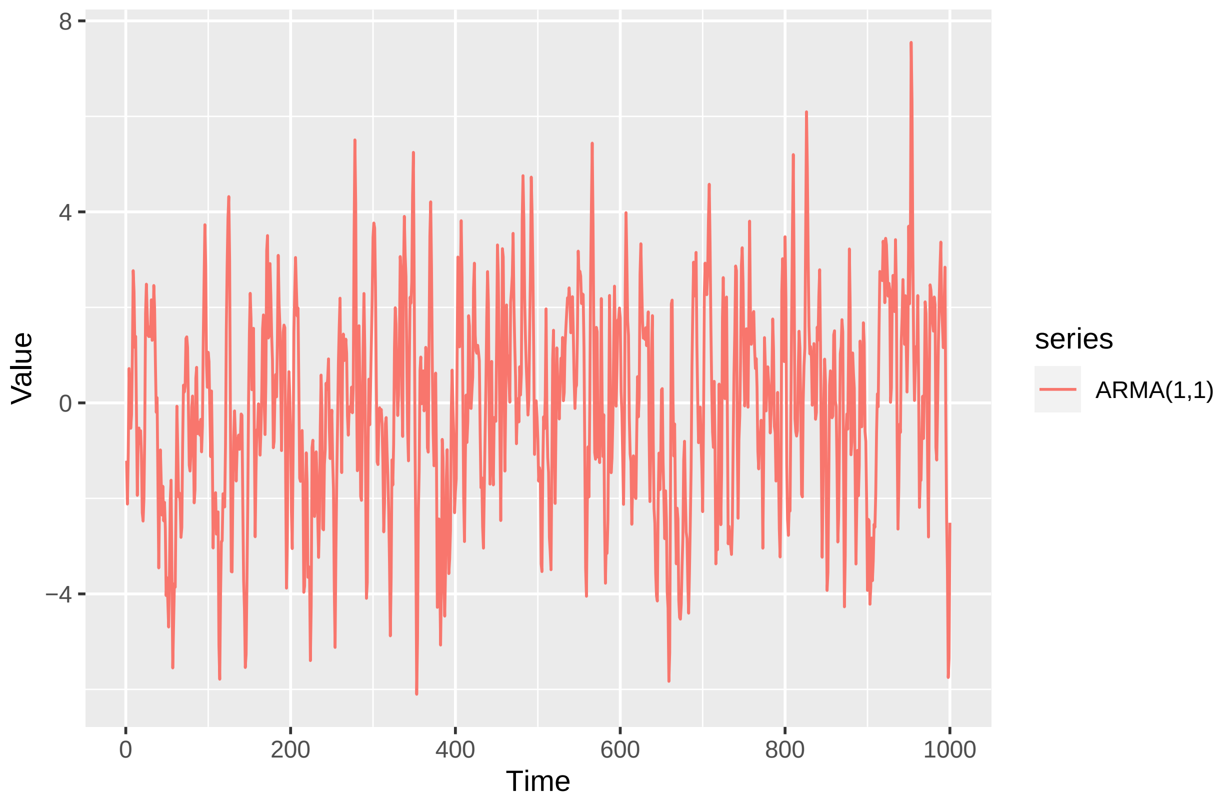
```
## Warning in min(Mod(polyroot(c(1, -model$ar)))): no non-missing arguments to
## min; returning Inf
```

```r
# Generate data for ARMA(1,1)
arma_11 <- arima.sim(model = list(ar = phi, ma = theta), n = n)

# Plot the generated series
autoplot(arma_10, series = "ARMA(1,0)", xlab = "Time", ylab = "Value")
```



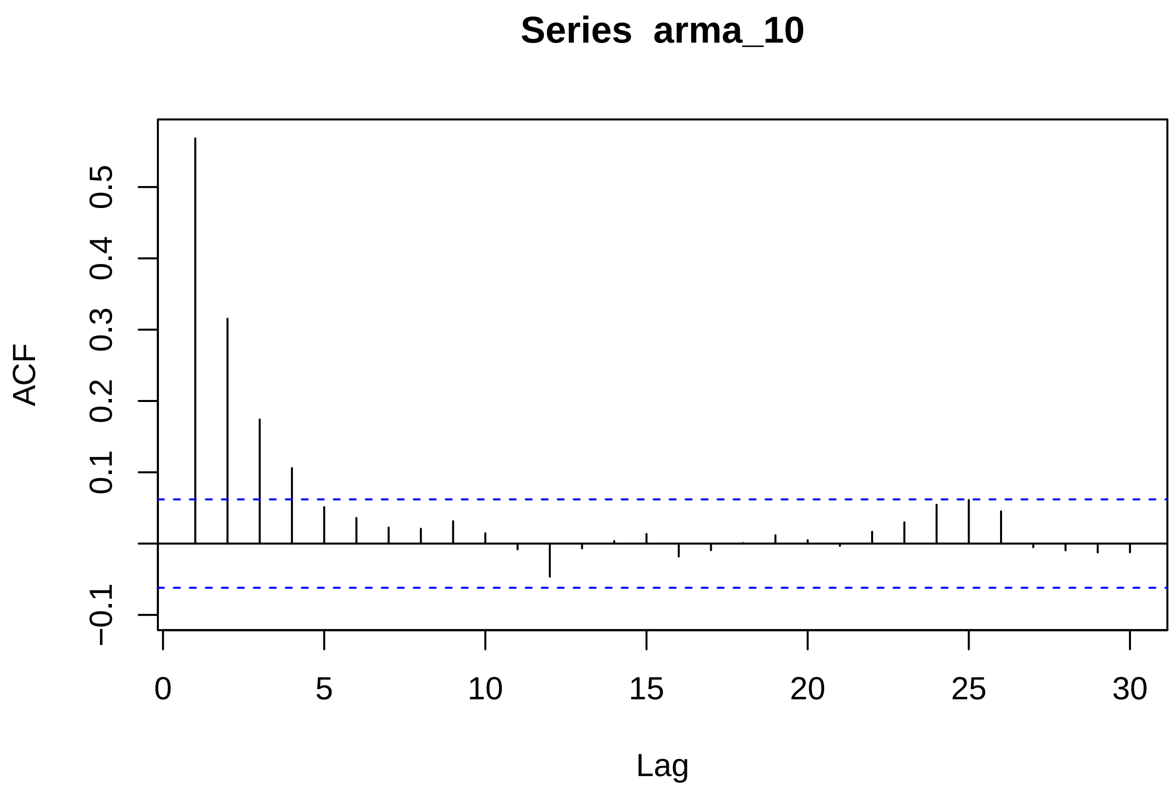

```r
autoplot(arma_01, series = "ARMA(0,1)", xlab = "Time", ylab = "Value")
```


```r
autoplot(arma_11, series = "ARMA(1,1)", xlab = "Time", ylab = "Value")
```



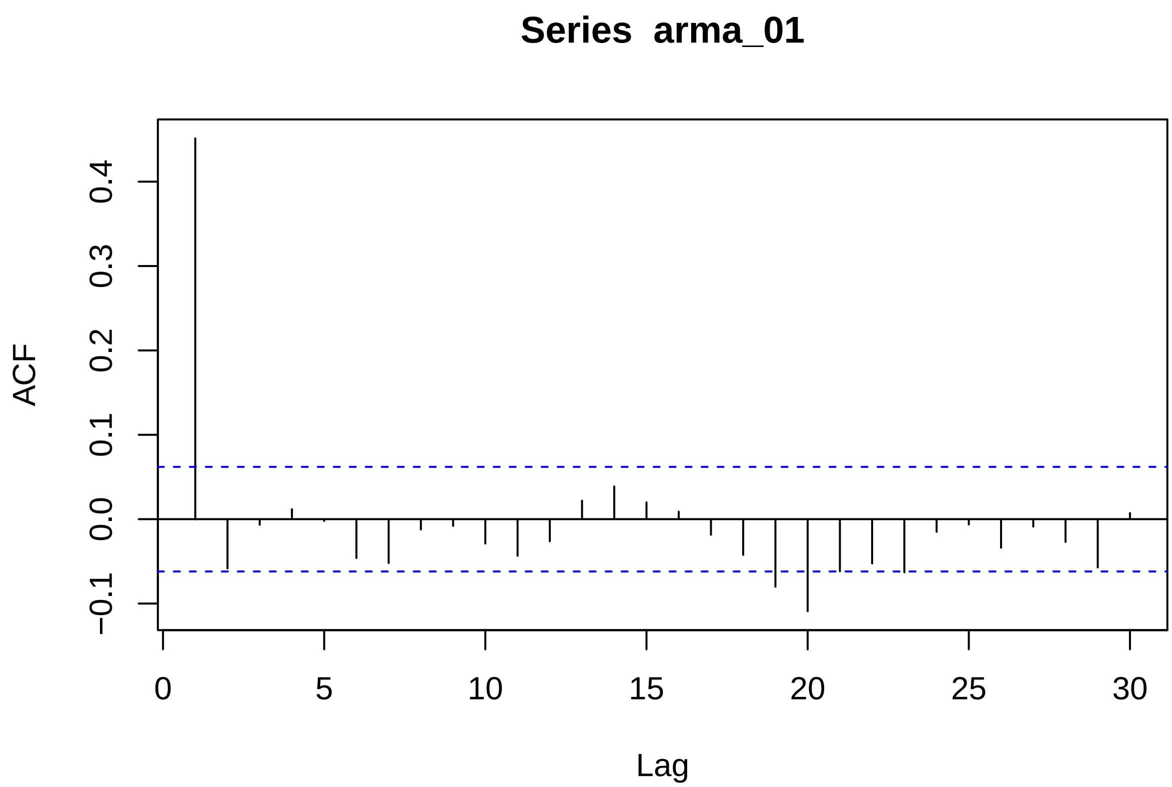

```
#(b)
# Compute ACF for each series
acf_arma_10 <- autoplot(Acf(arma_10), main = "ACF for ARMA(1,0)")
```

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



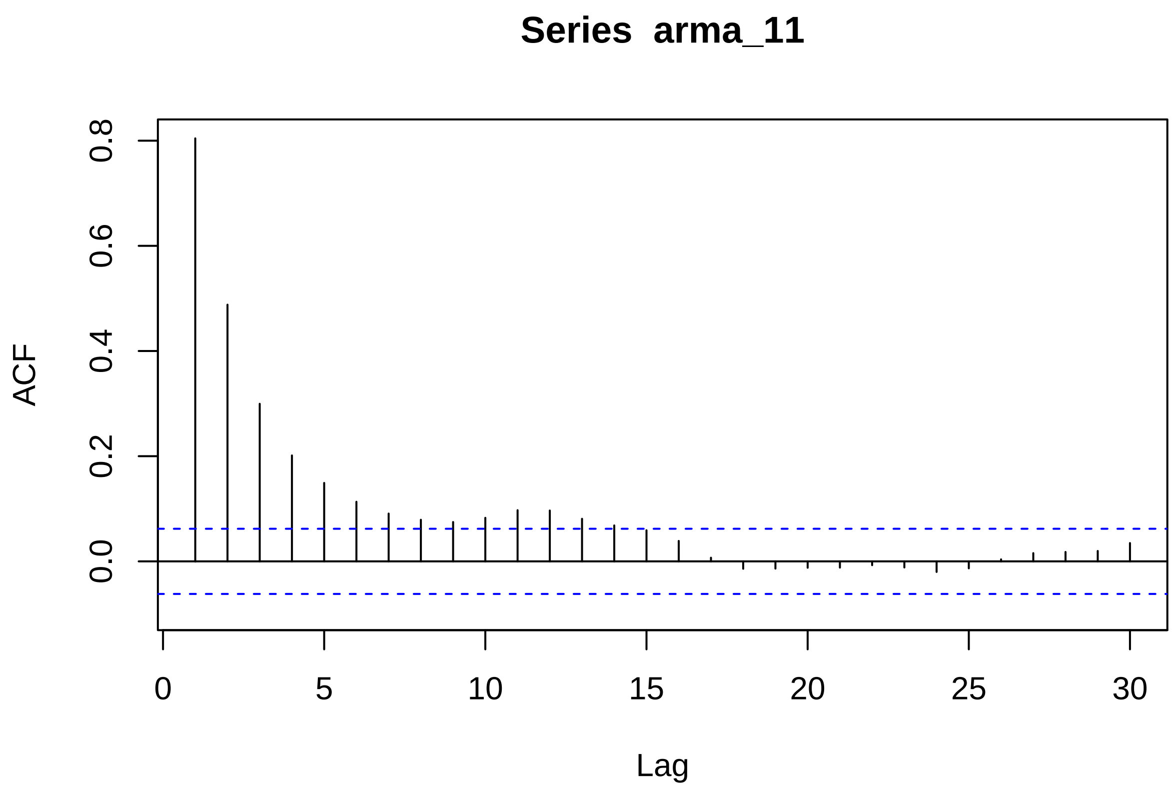

```r
acf_arma_01 <- autoplot(Acf(arma_01), main = "ACF for ARMA(0,1)")
```

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

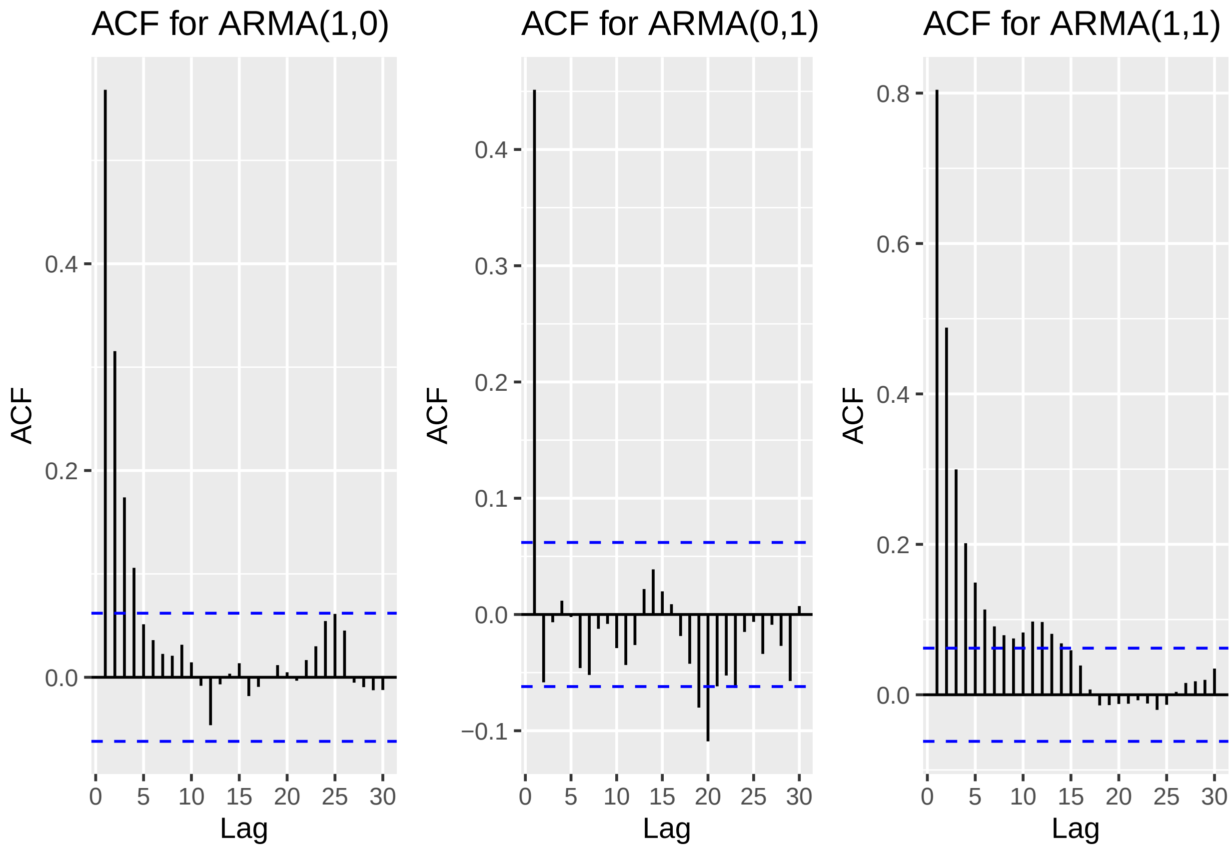



```
acf_arma_11 <- autoplot(Acf(arma_11), main = "ACF for ARMA(1,1)")
```

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

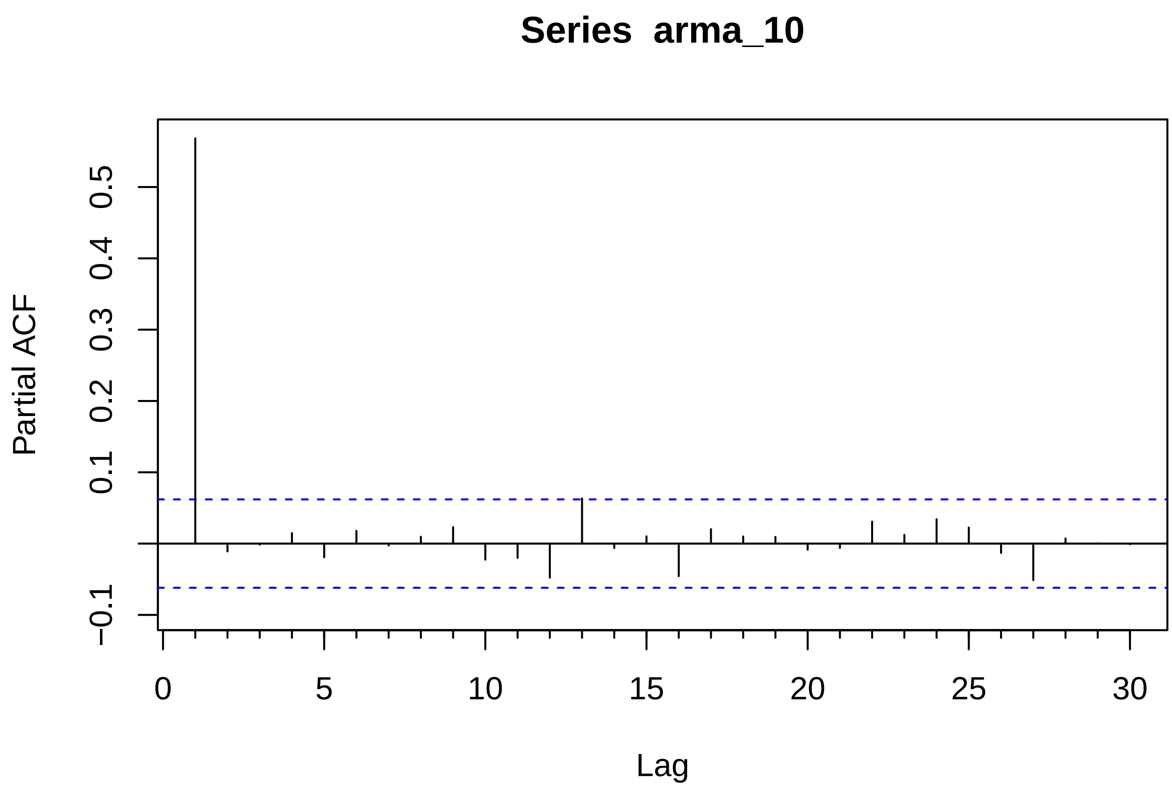



```r
# Plot ACF for each series in one window
plot_grid(acf_arma_10, acf_arma_01, acf_arma_11, ncol = 3)
```



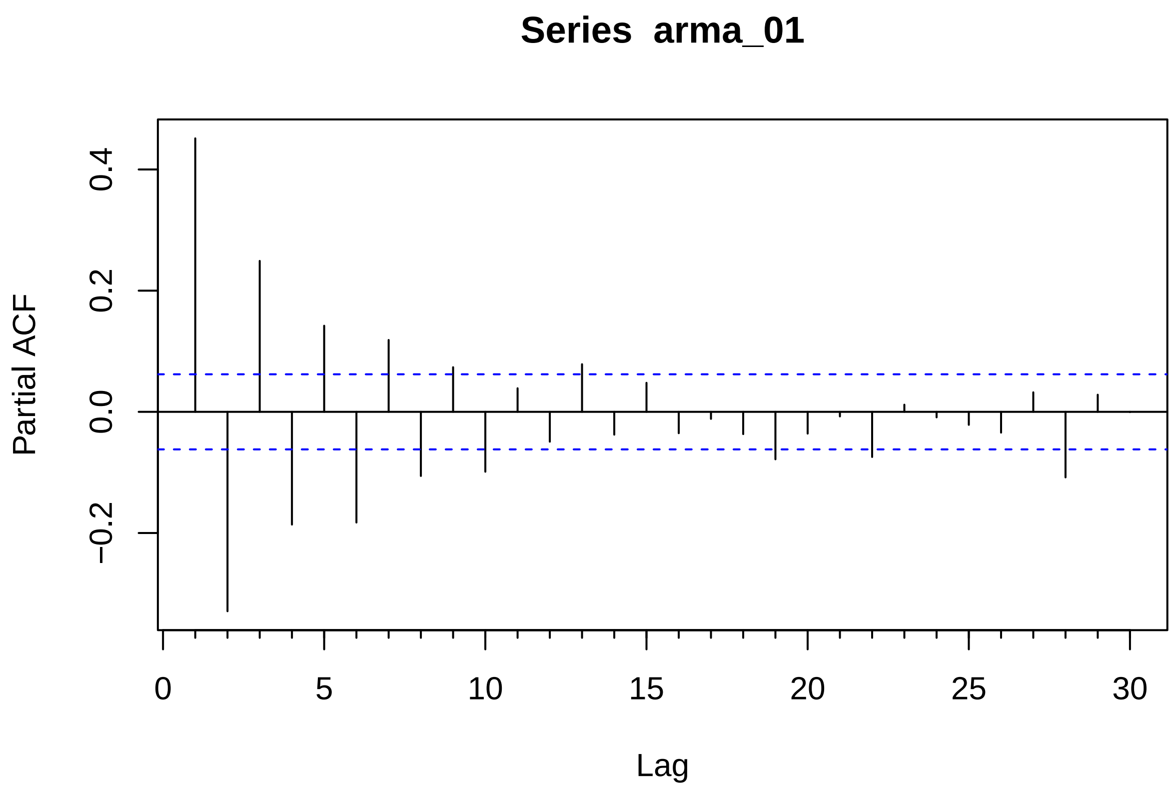

```r
#(c)
# Compute PACF for each series
pacf_arma_10 <- autoplot(Pacf(arma_10), main = "PACF for ARMA(1,0)")
```

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



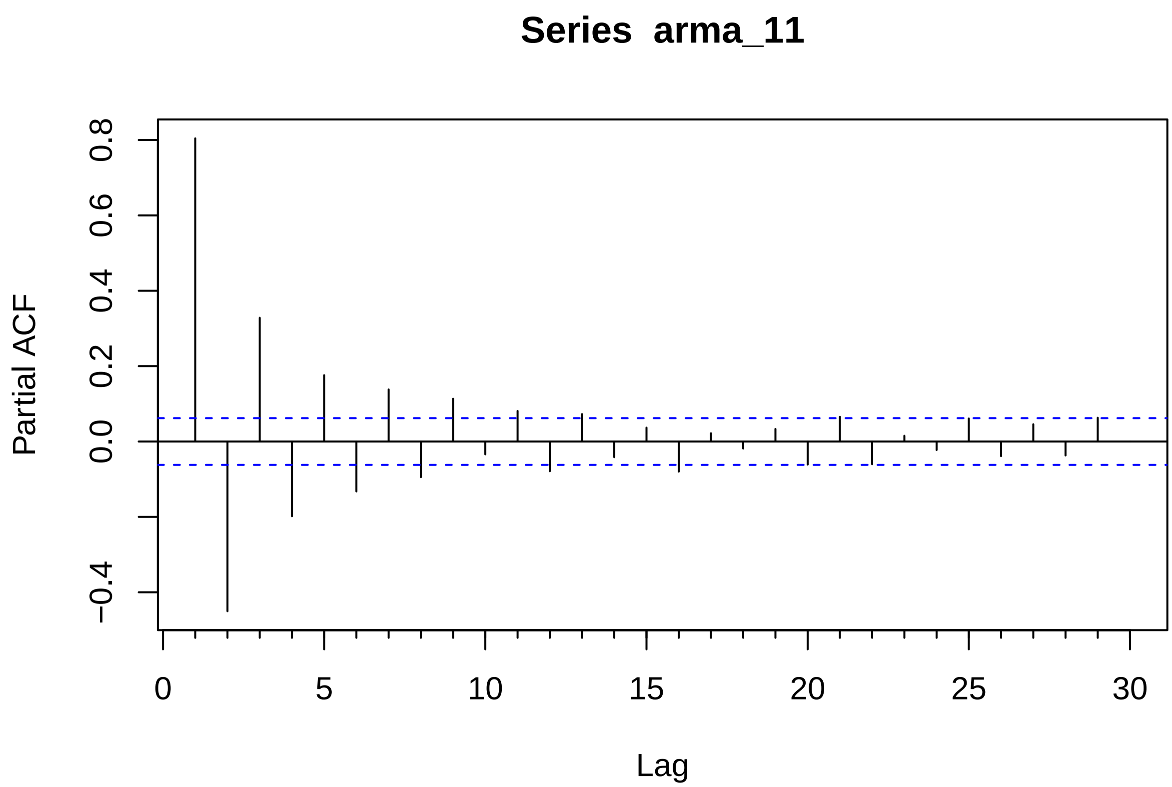

```
pacf_arma_01 <- autoplot(Pacf(arma_01), main = "PACF for ARMA(0,1)")
```

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

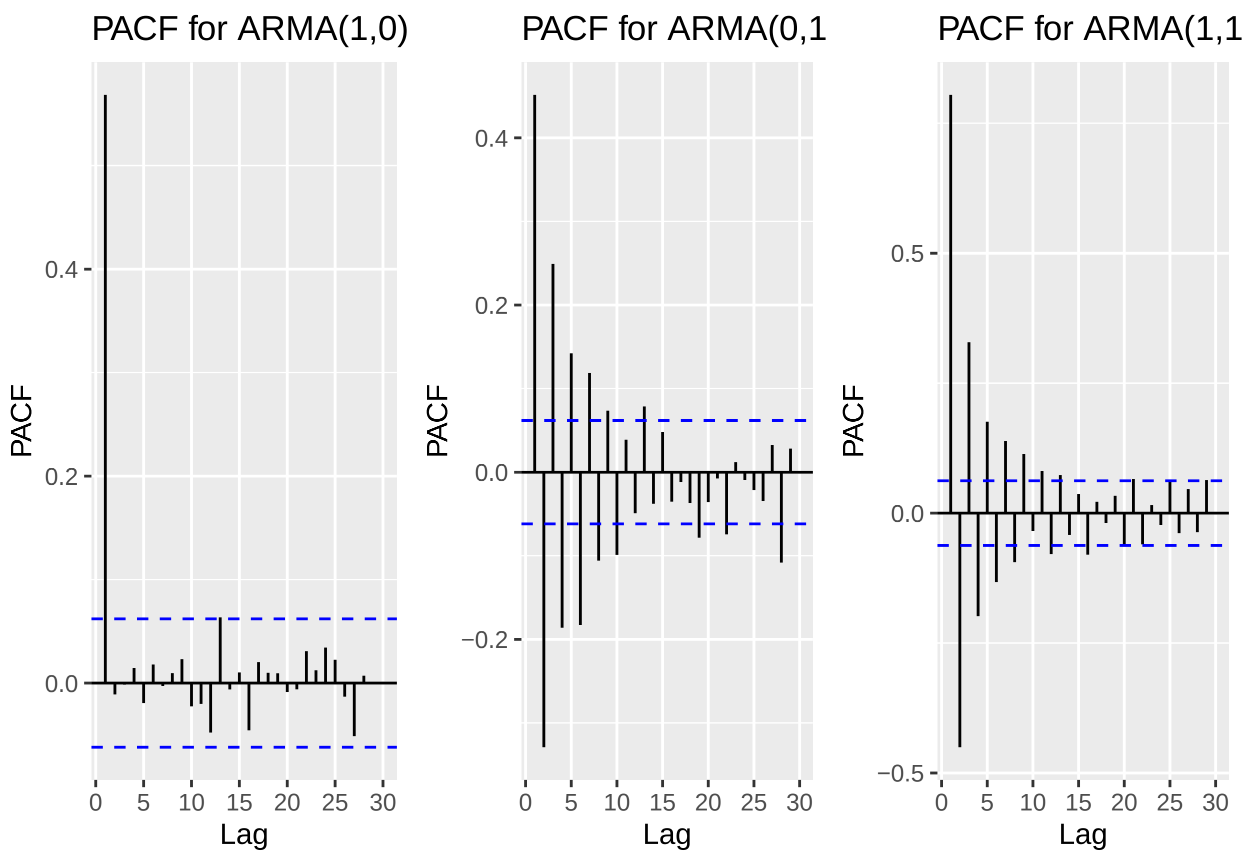



```
pacf_arma_11 <- autoplot(Pacf(arma_11), main = "PACF for ARMA(1,1)")
```

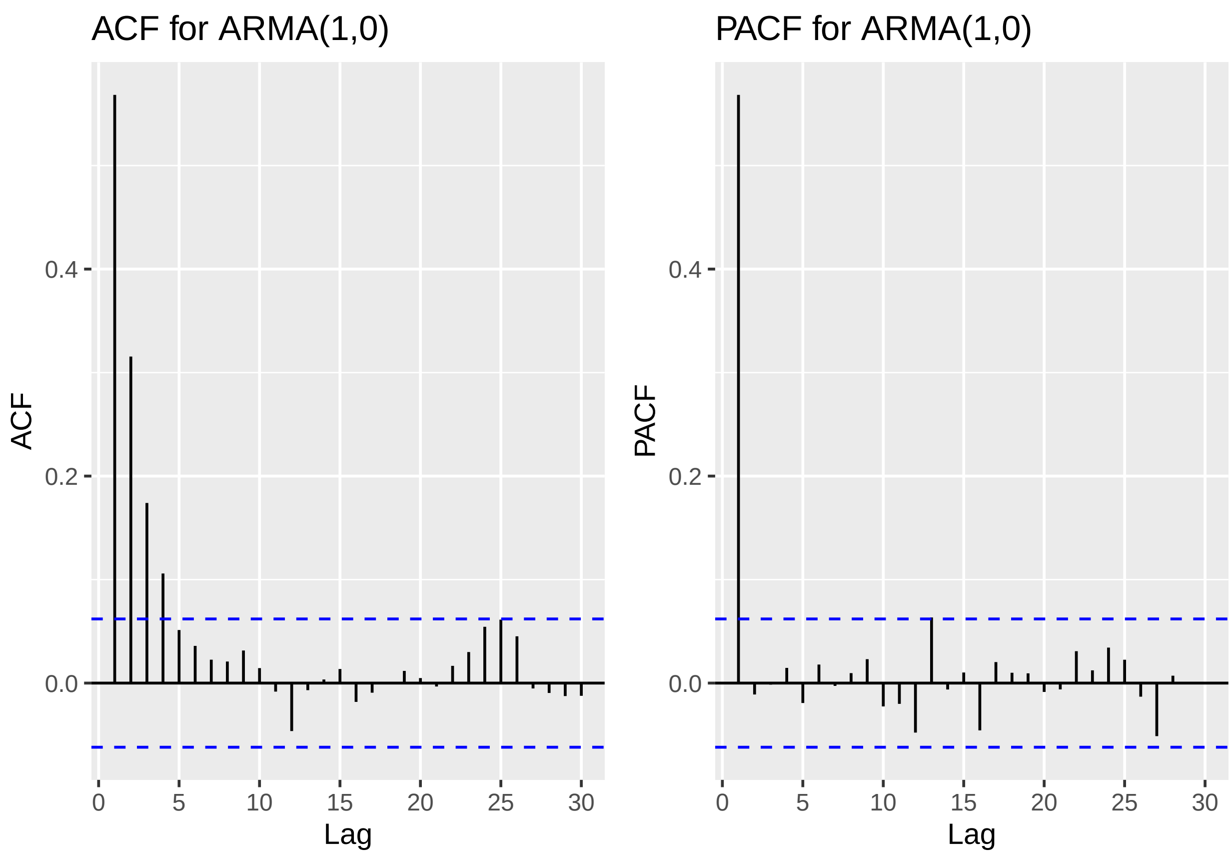
```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: ‘main‘
```



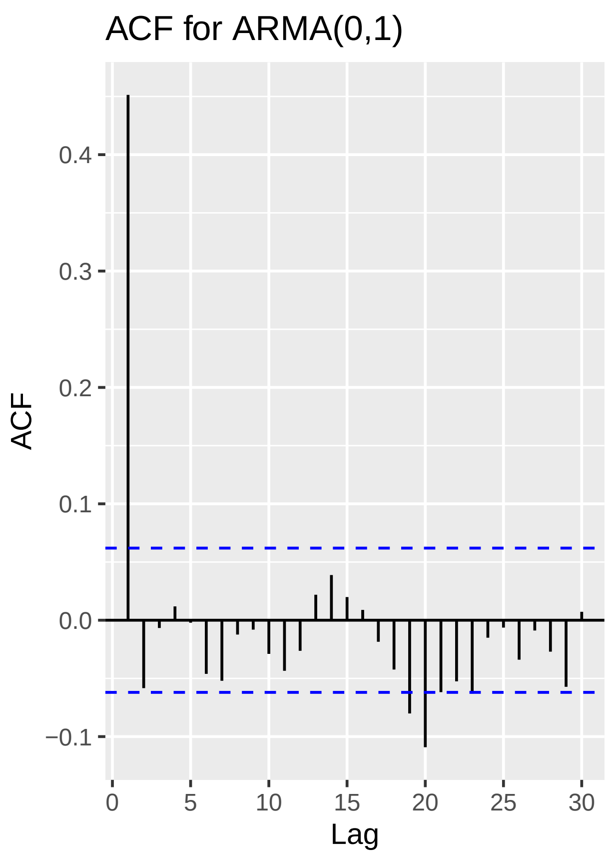

```r
# Plot PACF for each series in one window
plot_grid(pacf_arma_10, pacf_arma_01, pacf_arma_11, ncol = 3)
```

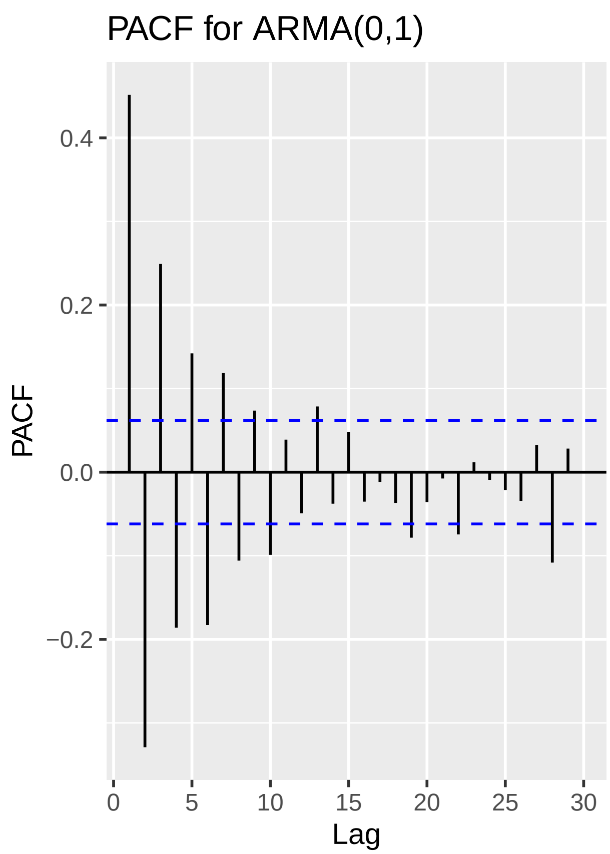



```r
#(d)
# Arrange ACF and PACF pairs for each series
plot_grid(acf_arma_10, pacf_arma_10, ncol = 2)
```

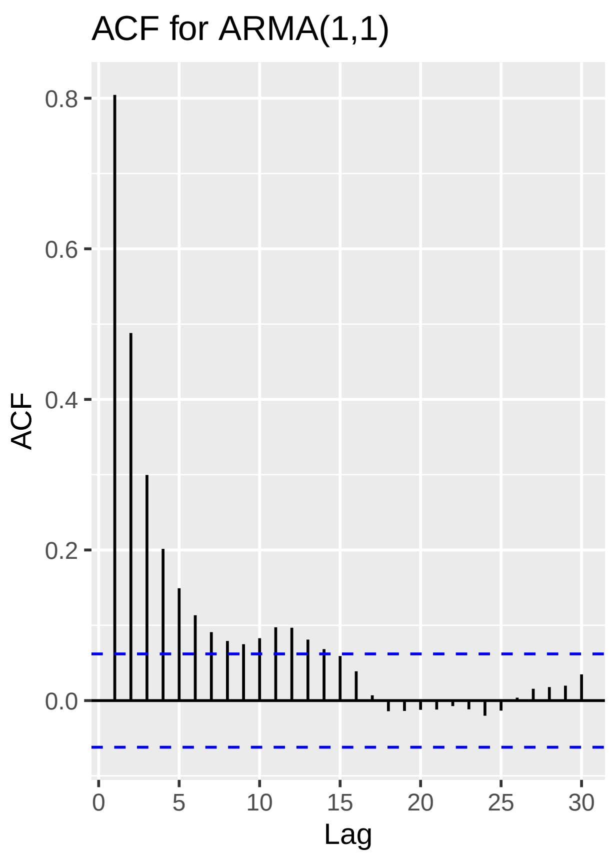


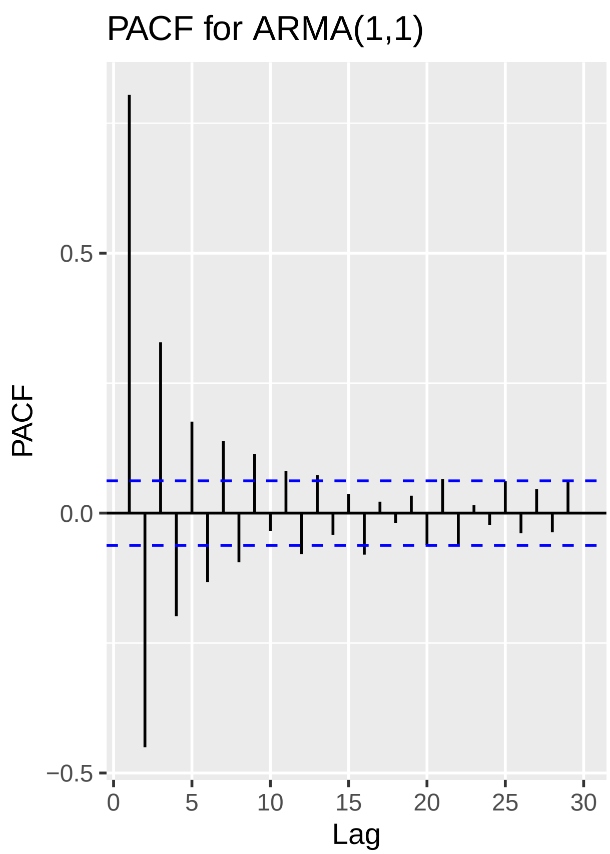

```
plot_grid(acf_arma_01, pacf_arma_01, ncol = 2)
```






```r
plot_grid(acf_arma_11, pacf_arma_11, ncol = 2)
```






```
#For ARMA(1,0), consistent with Q1 for AR models, the ACF for this model decays
#exponentially with time and the PACF tells us the order of the process.
#For ARMA(0,1), consistent with Q1 for MA models, the PACF for this model decays
#exponentially with time and the ACF tells us the order of the process.
#For ARMA(1,1), consistent with Q1, both significant spikes at the initial lags
#and exponential decay afterward suggest an ARMA model.
#It's a bit tough since we know how these plots are generated instead of just
#looking at them in isolation and identifying the model types... we'd be able to
#create a reasonable guess, but likely would not be able to say definitively
#just based on the ACF and PACF in isolation.


#(e)
#For ARMA(1,0), the PACF value R computed at lag 1 is a bit lower than 0.6,
#perhaps at 0.57 or so.
#For ARMA(1,1), the PACF value computed at lag 1 exceeds 0.6 by a bit
#and is closer to 0.62. We know that both the ACF and the PACF are the result
#of superimposing the AR and MA properties. In the PACF, initial values are
#dependent on the AR followed by the decay due to the MA part. Therefore, the
#ARMA(1,0) lag 1 correlation coefficient should match the value of phi and it's
#quite close here. The ARMA (1,1) lag 1 correlation coefficient should also
#roughly match, but is subject to the exponential decay of the MA component,
#so it should be a bit higher, which it is.
```

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: ARIMA$(1, 0, 0)(0, 0, 1)_1 2$

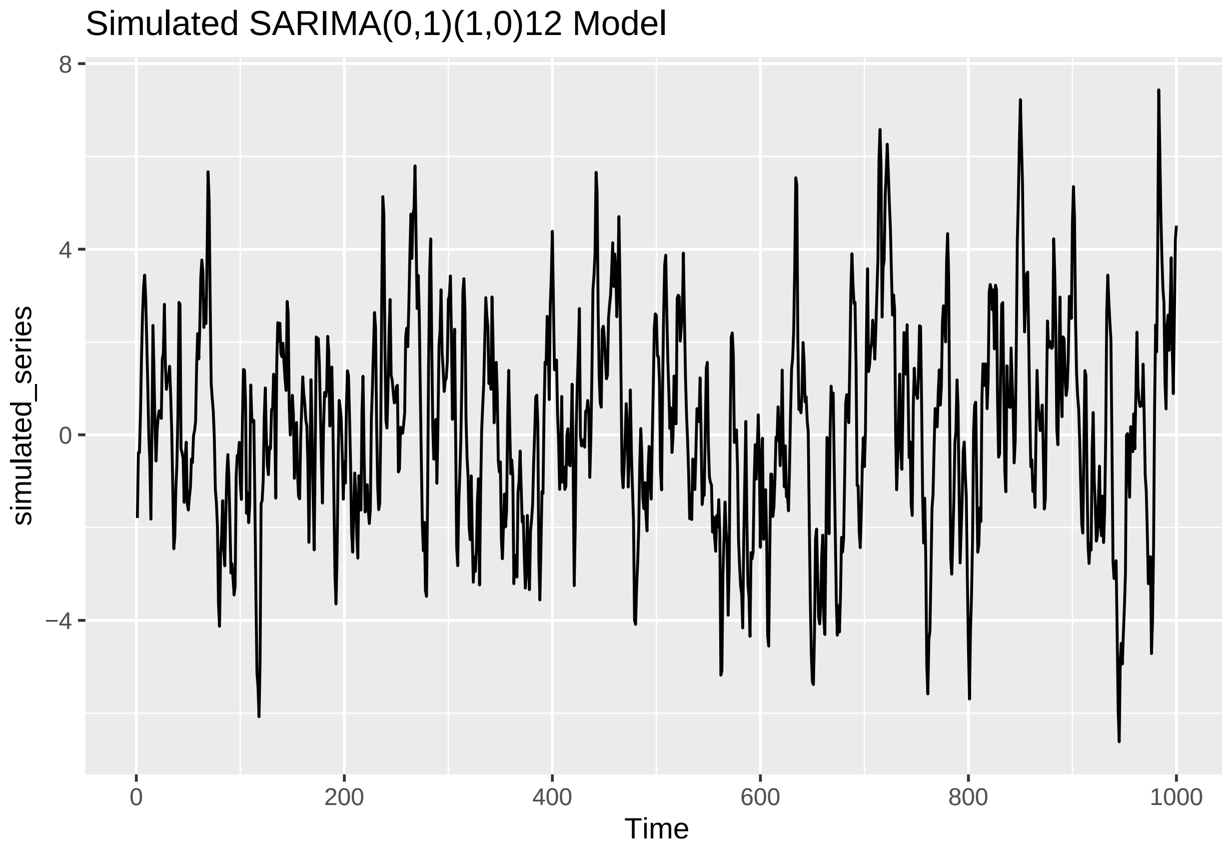
(b) Also from the equation what are the values of the parameters, i.e., model coefficients.

Answer: phi for non-seasonal = 0.7, Phi for seasonal = -0.25, theta = -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
# Set parameters
phi <- 0.8
theta <- 0.5
n <- 1000
seasonal_period <- 12  # Monthly data with a seasonal period of 12 months

# Define SARIMA model list
SARIMA_model_list <- list(order = c(1, 0, 1),
                          seasonal = list(order = c(1, 0, 0), period = 12),
                          ar = phi, ma = theta)

# Simulation of the SARIMA model
set.seed(123)  # Setting seed for reproducibility
simulated_series <- arima.sim(n = n, model = SARIMA_model_list)

# Plot the generated series
autoplot(simulated_series, main = "Simulated SARIMA(0,1)(1,0)12 Model")
```



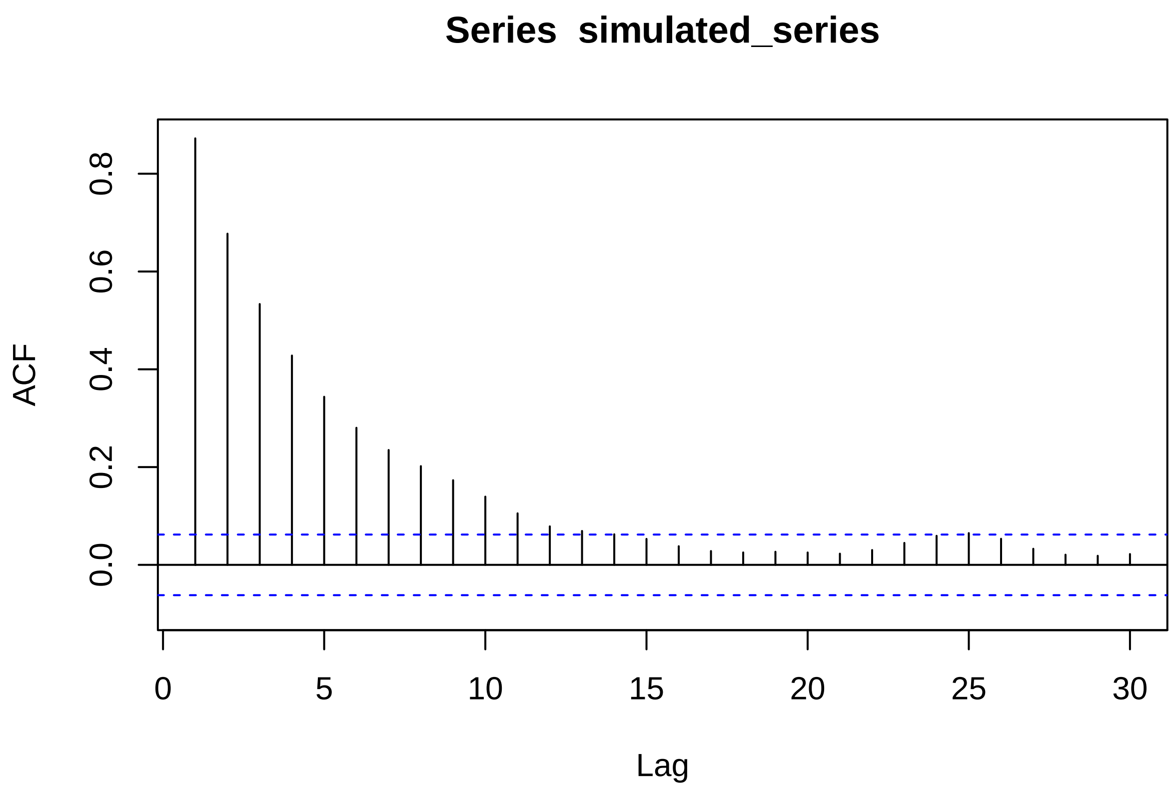

```
#The plot looks somewhat seasonal, but there appears to be a lot of other noise
#in the data that makes it hard to tell.
```

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.

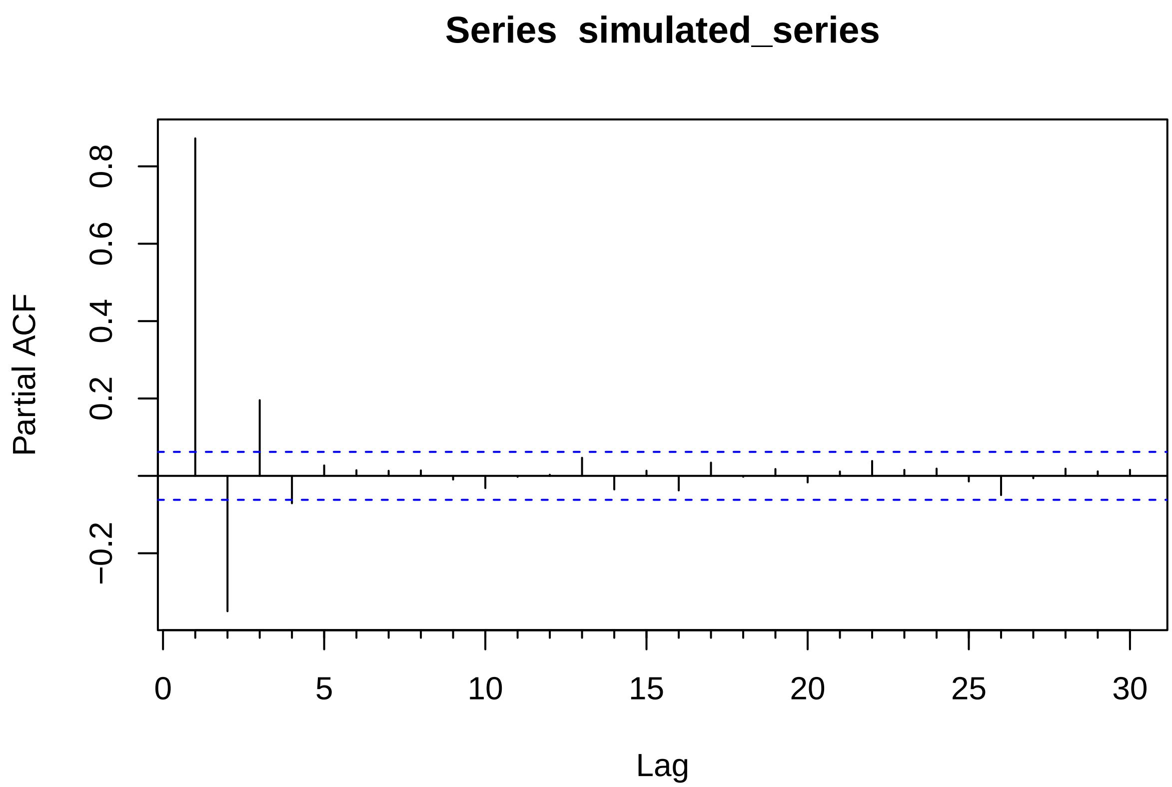
```
# Generate ACF plot
acf_plot <- autoplot(Acf(simulated_series), main = "ACF of Simulated Series")
```

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

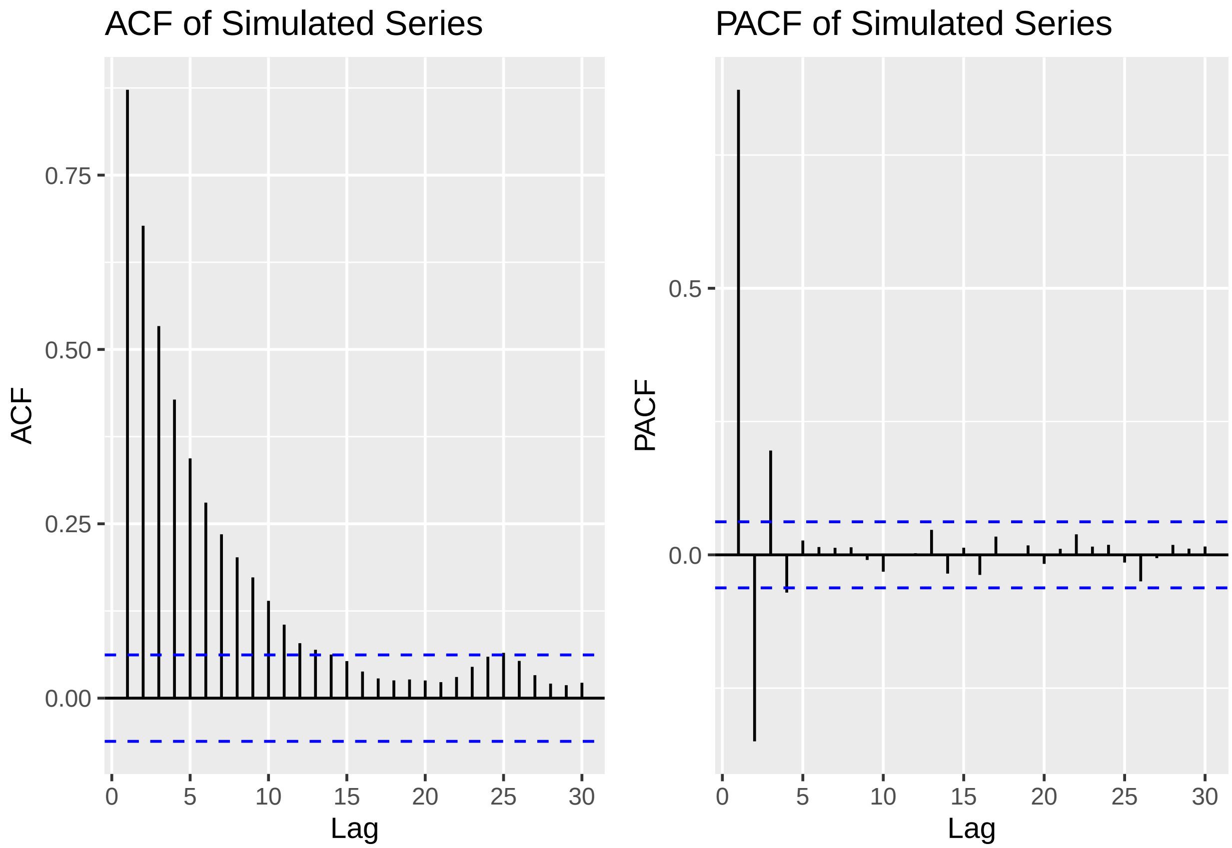



```
# Generate PACF plot
pacf_plot <- autoplot(Pacf(simulated_series), main = "PACF of Simulated Series")
```

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




```r
# Arrange plots side by side using plot_grid
plot_grid(acf_plot, pacf_plot, ncol = 2)
```




```
#In neither the ACF nor the PACF are there spikes at lags 12, 24, 36, etc.
#Therefore, it would appear that the plots are not well-representing the
#simulated model. You would not be able to identify the order of the
#non-seasonal and seasonal components from the plots.
```