# Lab 3

Derek Situ (62222807)

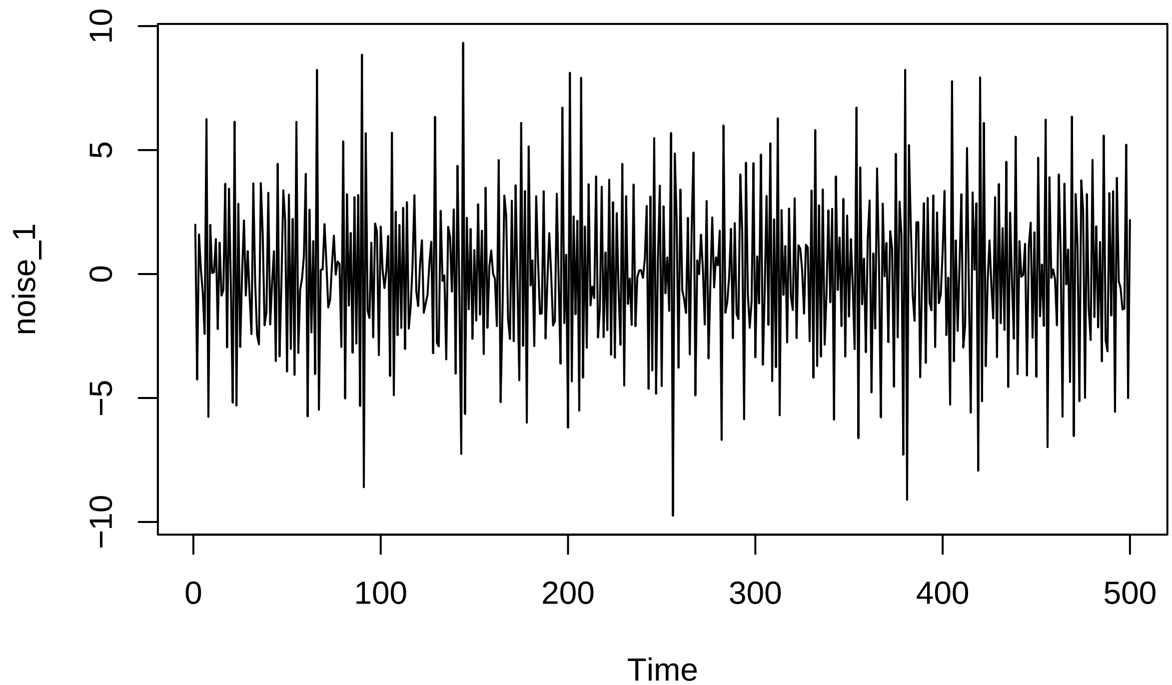
1/31/2022

```r
# For Q1 and Q2: Function to run repeated simulations
simulation <- function(coeffs, n, sd, repeats) {
  for (i in 1:repeats) {
    cat("Simulation", i)
    noise <- arima.sim(list(ma = coeffs), n = n, sd = sd)
    acf(noise)
  }
}
```

## Question 1

### (1a)

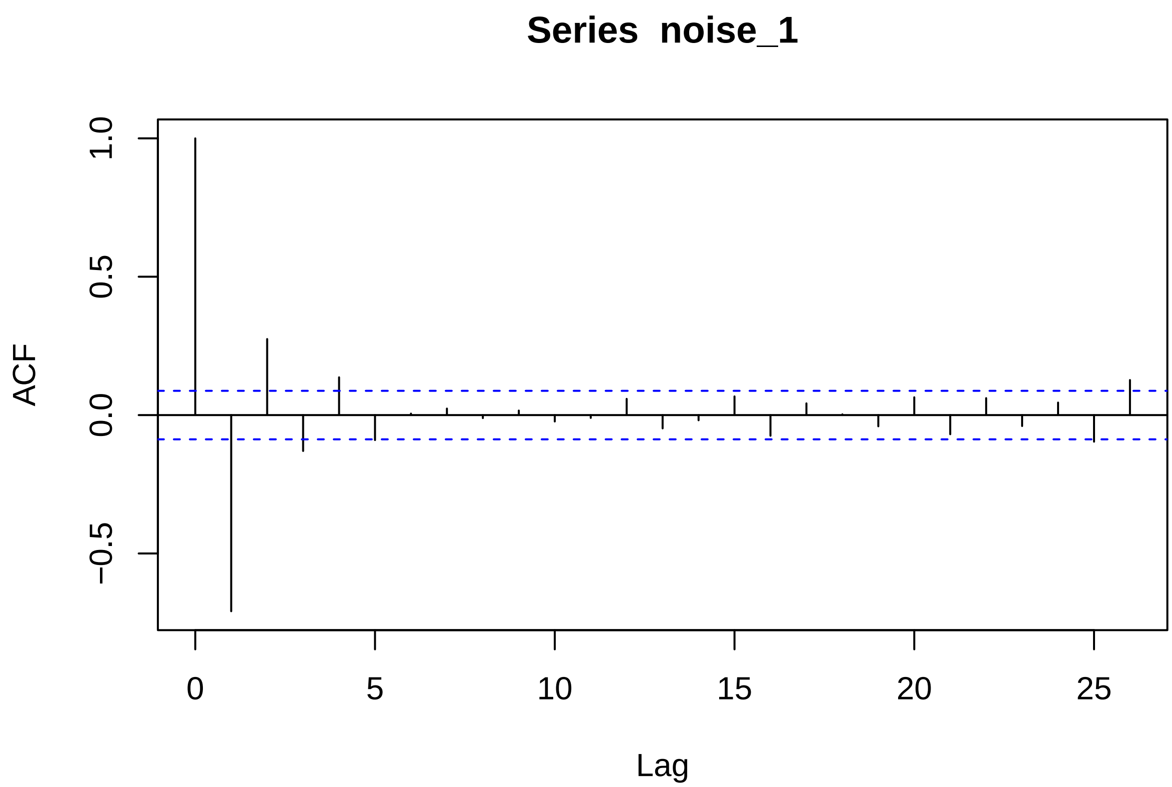
```r
# simulate and plot the series
noise_1 <- arima.sim(list(ma = c(-4.25, 5.75, -1.8)), n = 500, sd = sqrt(0.2))
plot(noise_1)
```




Given the definition of the process, the acf should reflect that observations 1 lag apart will generally be negatively correlated, observations 2 lags apart will generally be positively correlated, observations 3 lags apart might be weakly negatively correlated, and observations more than 3 lags apart will theoretically have no correlation.

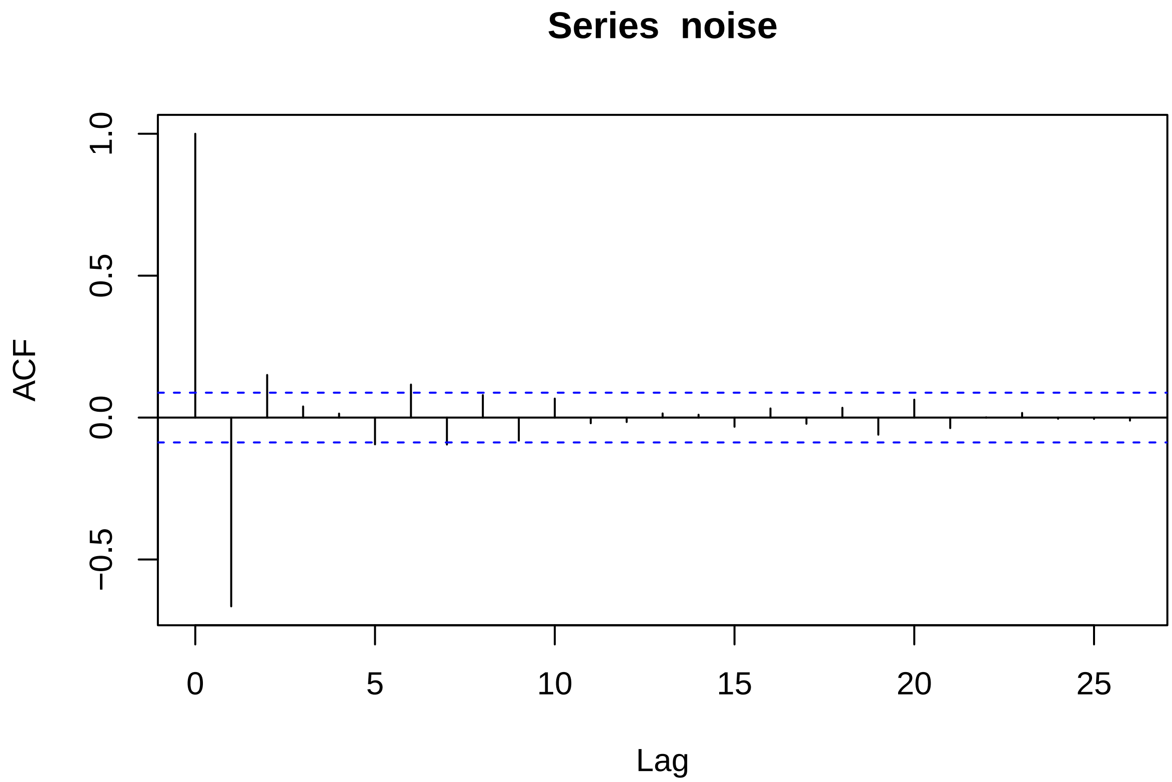
**(1b)**

```r
# generate correlogram for sample
acf(noise_1)
```

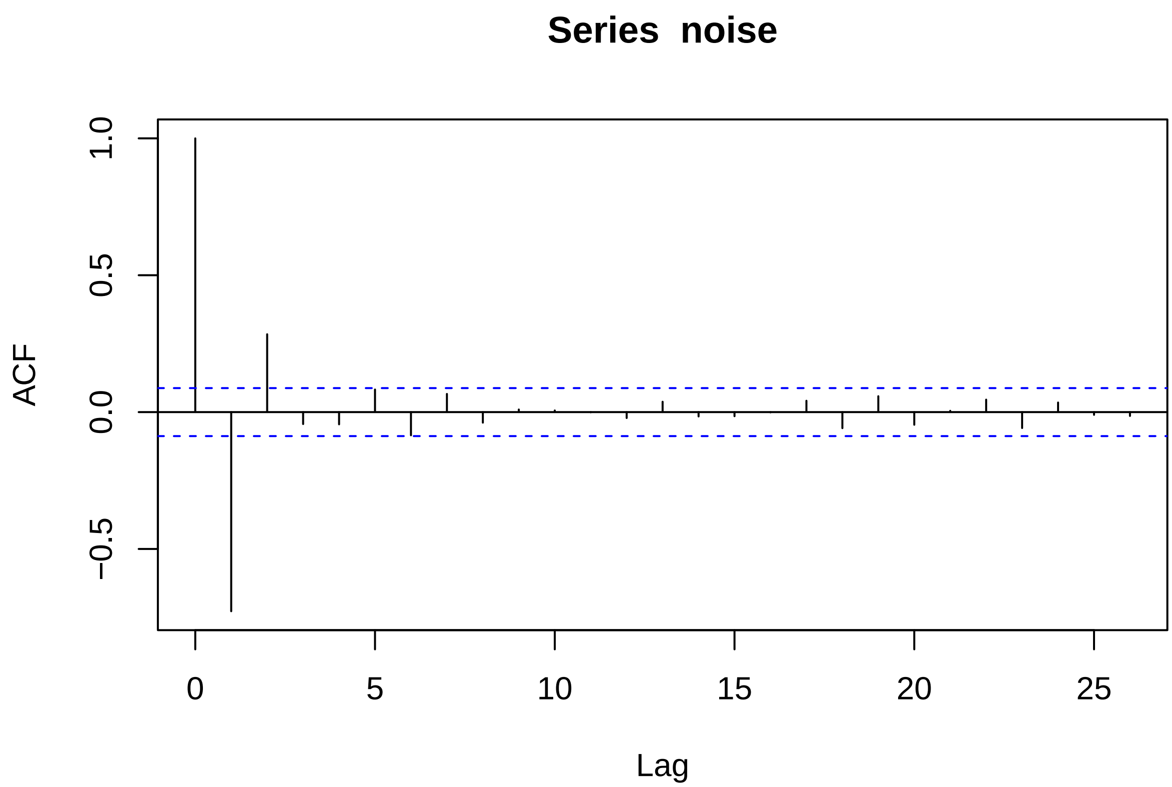



```
# repeat the simulation a couple more times and generate correlogram
simulation(c(-4.25,5.75,-1.8), 500, sqrt(0.2), 2)
```

```
## Simulation 1
```




```
## Simulation 2
```




We see that the correlograms generally look as predicted in (1a).

**(1c)**

```
# Use ARMAacf to compute the acf of the model
ARMAacf(ma = c(-4.25, 5.75, -1.8))
```

```
##           0           1           2           3           4
##  1.00000000 -0.70509347  0.24203016 -0.03251151  0.00000000
```

**(1d)**

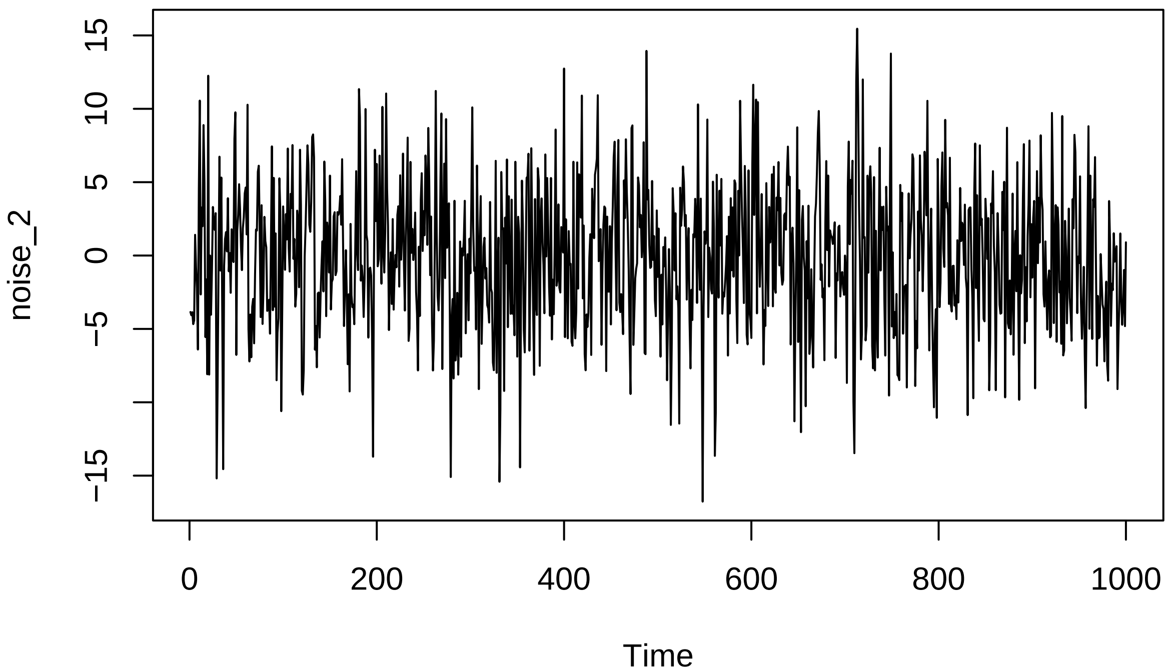
The ARMAacf function returns an acf for the model that is consistent with my prediction in (1a). At 1 lag apart, there is a negative autocorrelation -0.705, at 2 lags apart the autocorrelation is positive at 0.242, and at 3 lags apart there is a weak negative correlation at -0.0325, and at lags greater than 3, the acf of the model is 0.

The sample acf's are about the same as the acf of the model. This is expected because the samples were simulated from the model.

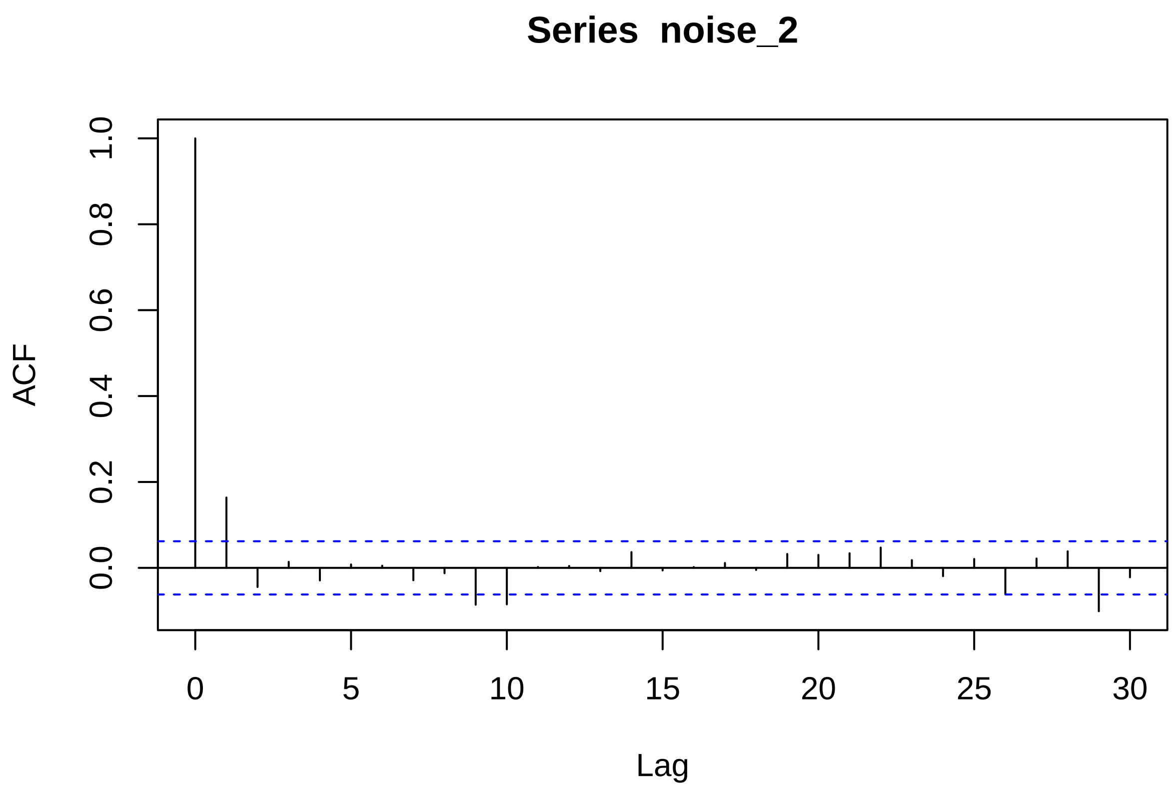
### Question 2

**(2a)**

```
noise_2 <- arima.sim(list(ma = c(5)), n = 1000, sd = sqrt(0.9))
plot(noise_2)
```



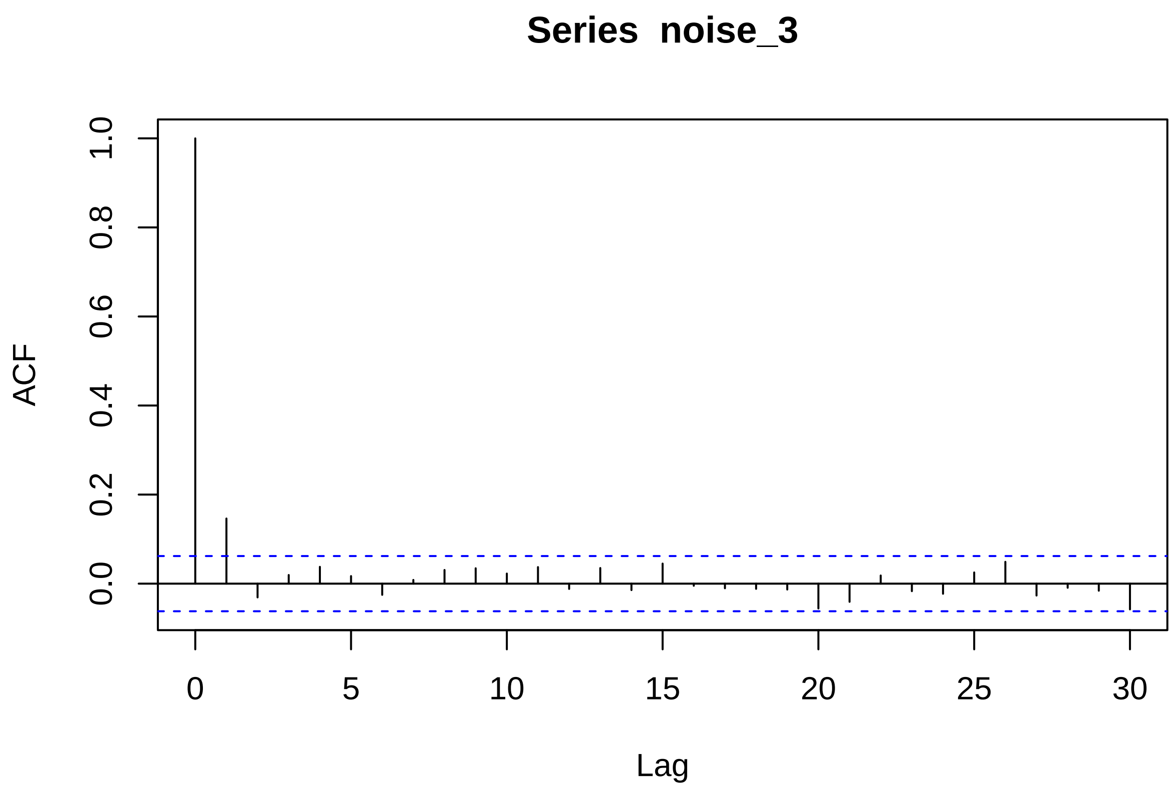

```
acf(noise_2)
```




The sample acf of this series shows a slight positive autocorrelation at lag = 1, and is close to 0 for lags greater than 1.

**(2b)**

```
noise_3 <- arima.sim(list(ma = c(0.2)), n = 1000, sd = sqrt(0.9))
acf(noise_3)
```

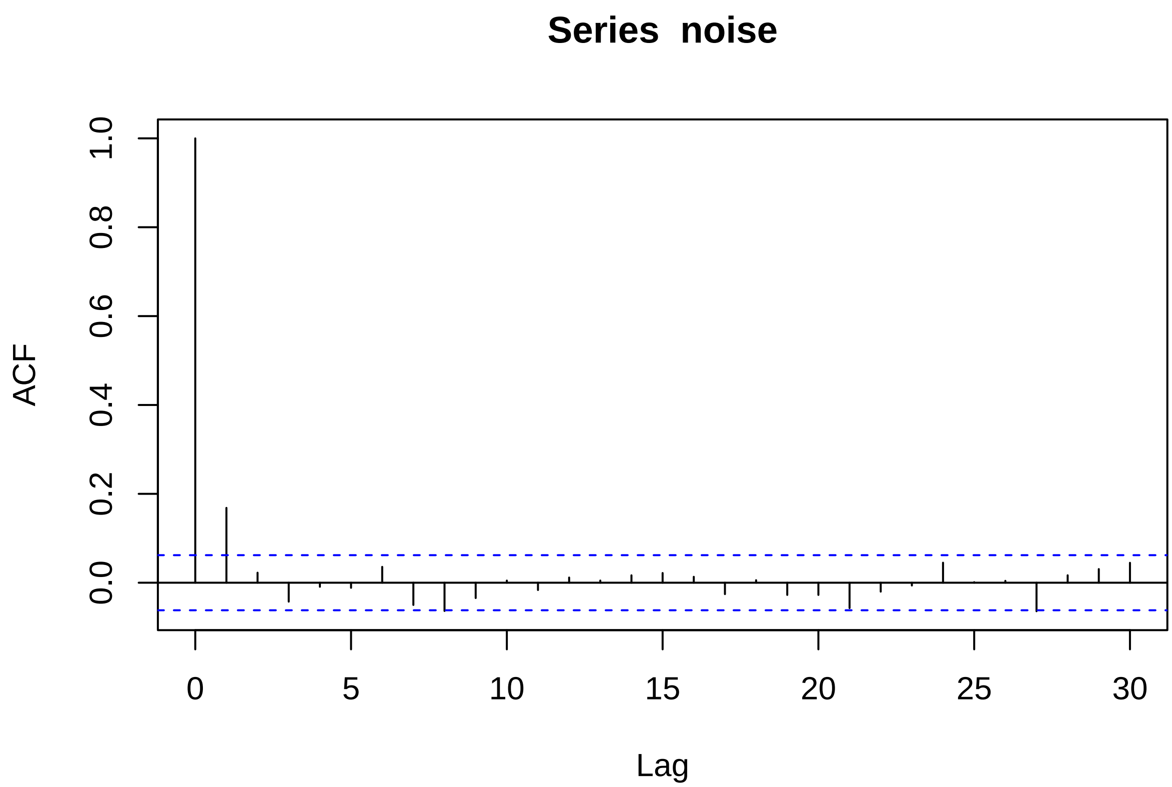



Again the sample acf of this series shows a slight positive autocorrelation at lag $= 1$, and is close to 0 for lags greater than 1.

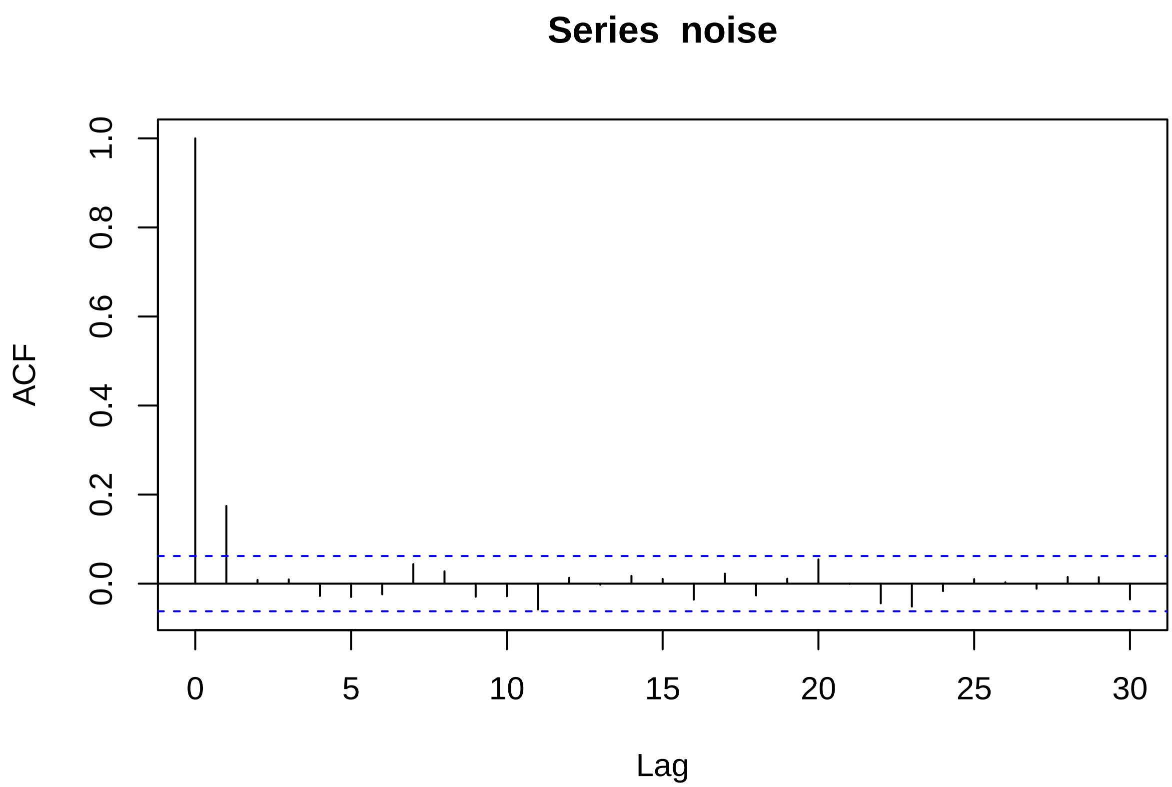
**(2c)**

```
# Simulating the process from (2a) and plotting the sample acf 3 times
simulation(c(5), 1000, sqrt(0.9), 3)
```

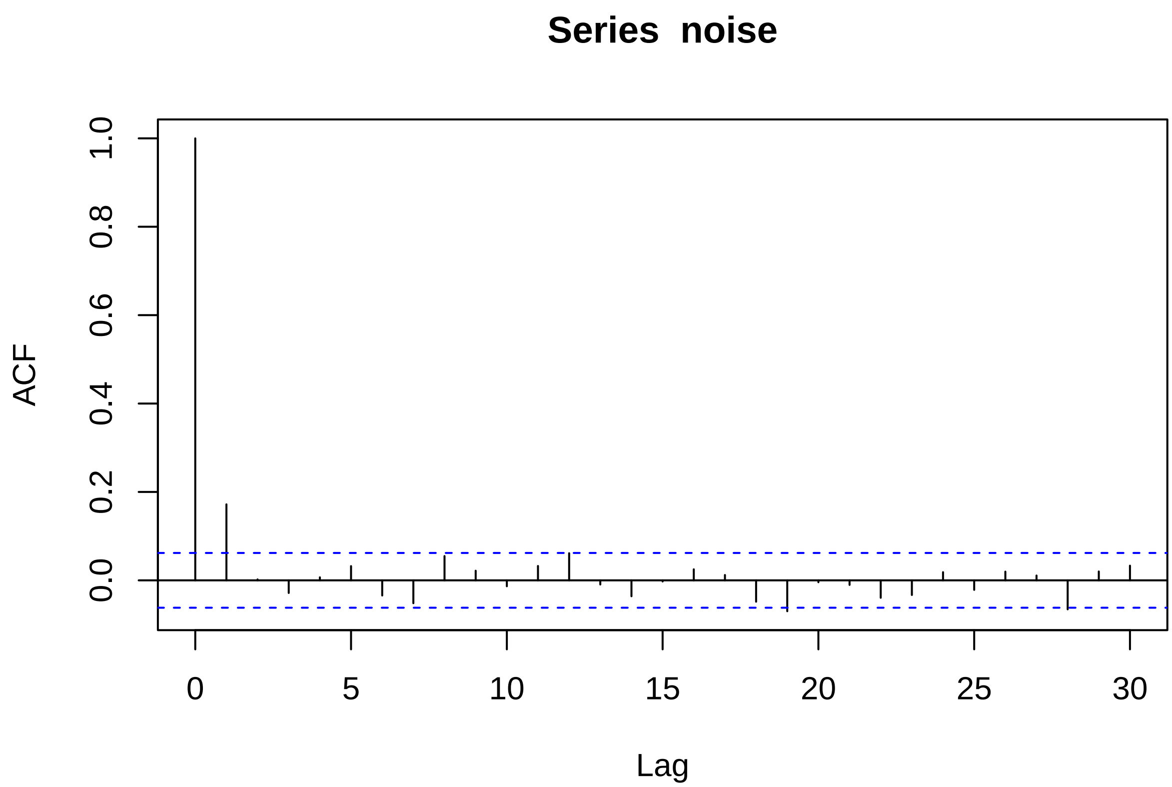
```
## Simulation 1
```




```
## Simulation 2
```

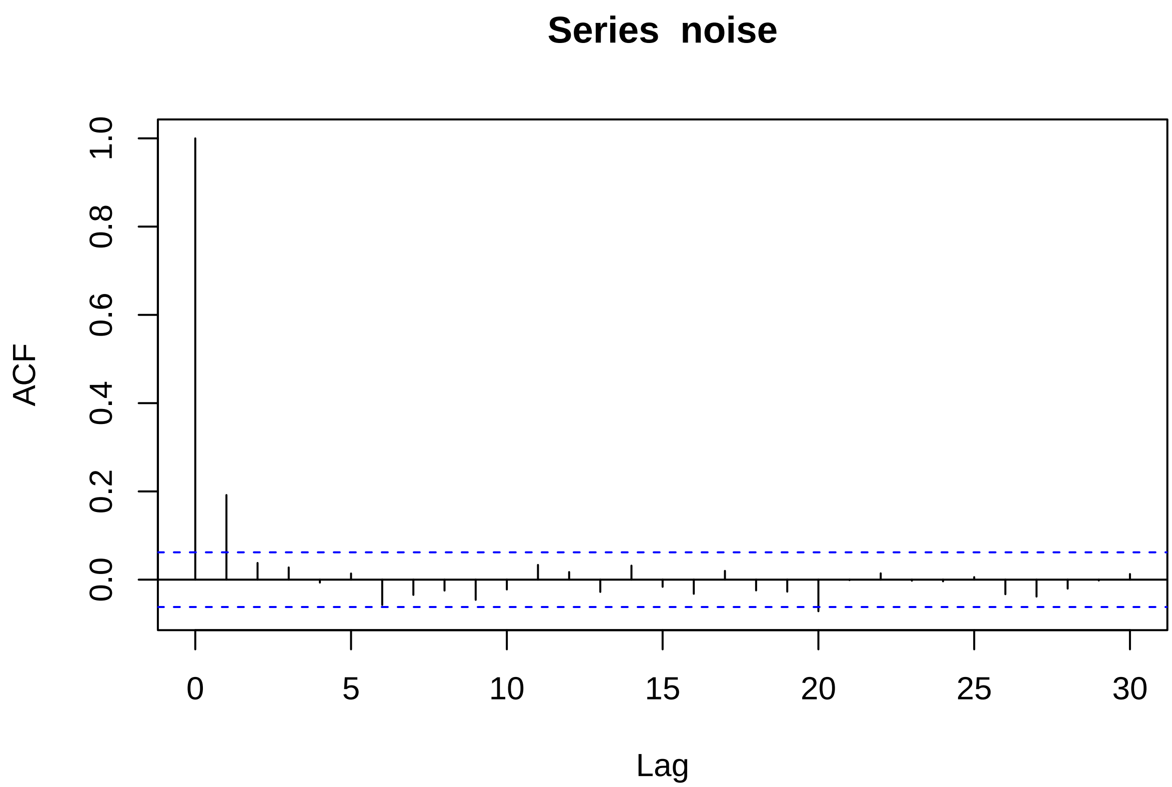



```
## Simulation 3
```

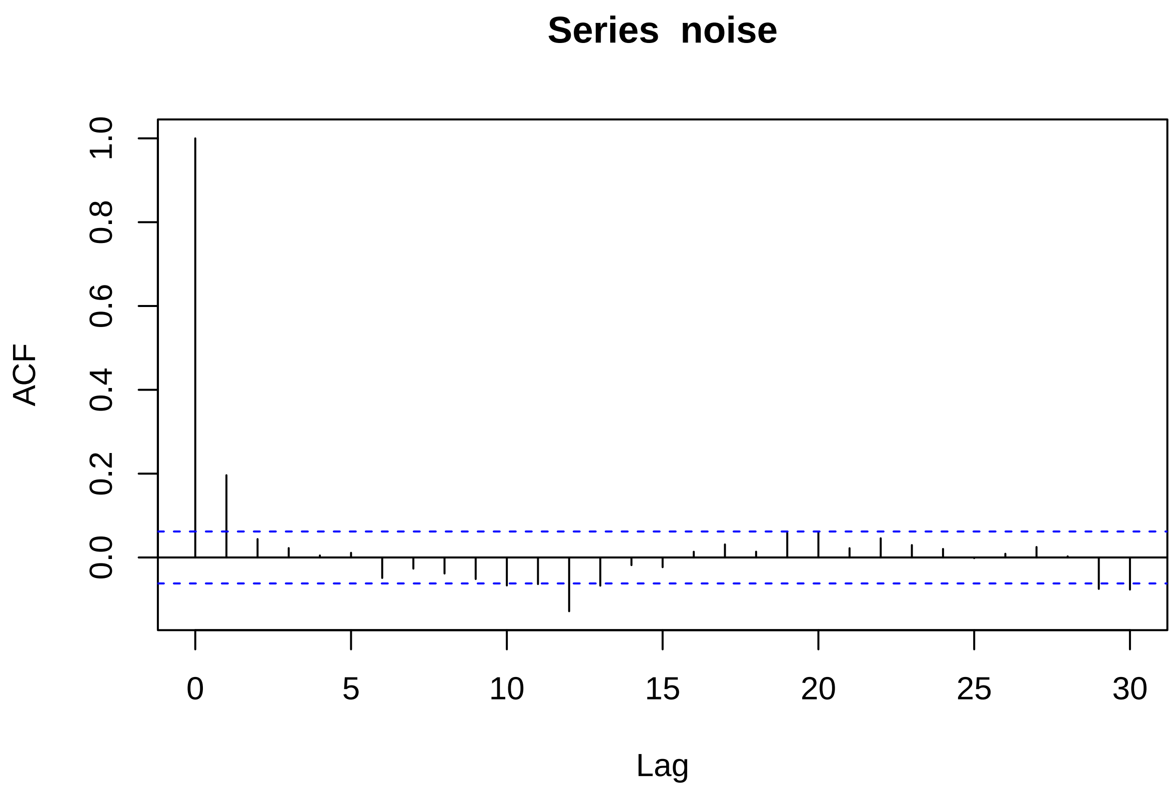



```
# Simulating the process from (2b) and plotting the sample acf 3 times
simulation(c(0.2), 1000, sqrt(0.9), 3)
```

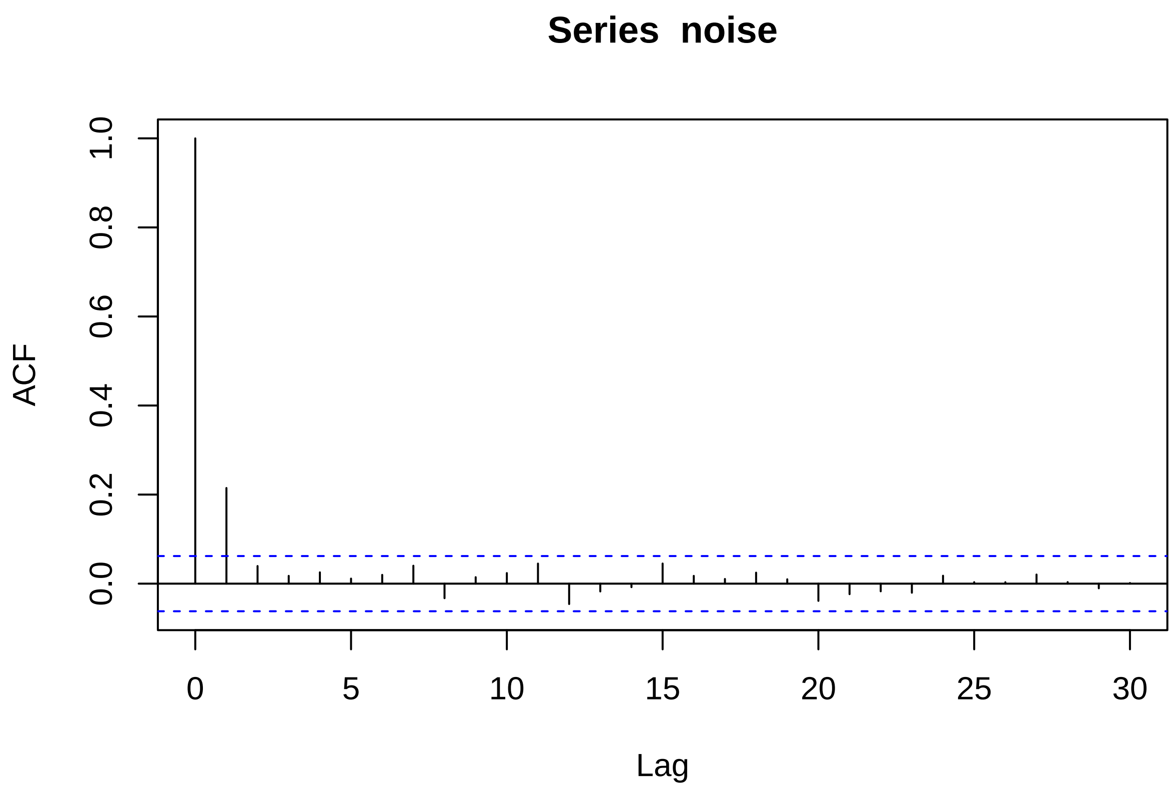
```
## Simulation 1
```




```
## Simulation 2
```




```
## Simulation 3
```



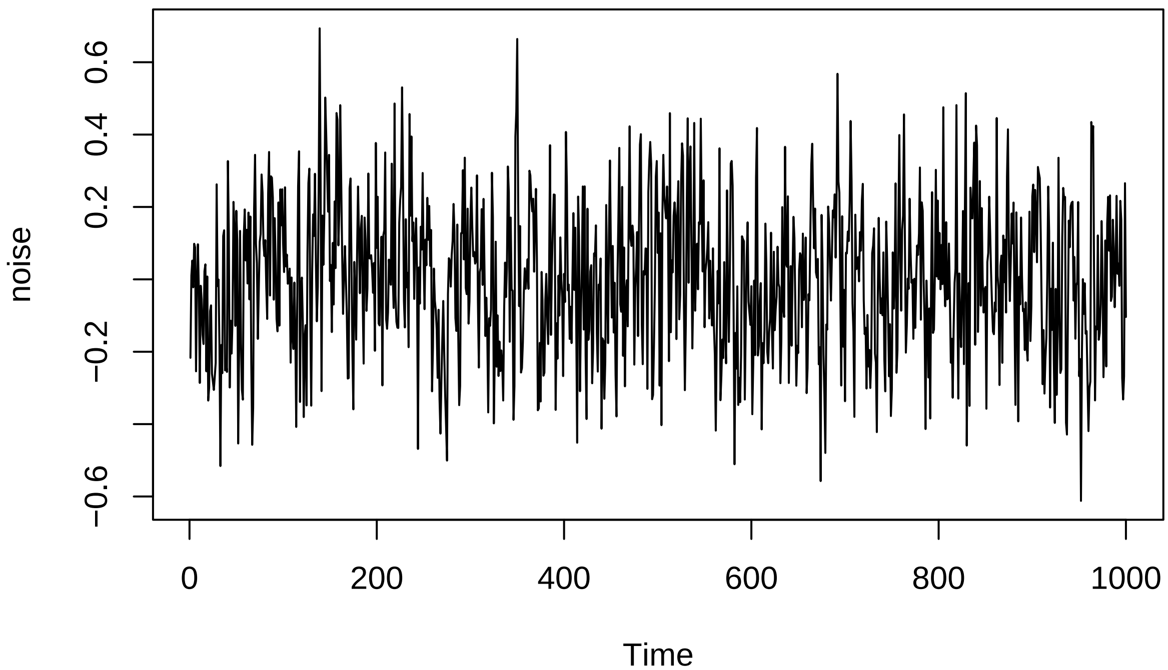

The sample acf's of the two processes are extremely similar. In fact, the acf's of the models are given by ARMAacf(ma = c(5)) and ARMAacf(ma = c(0.2)), which are the same. In general, two MA(1) processes with beta coefficients $\beta$ and $\frac{1}{\beta}$ respectively will share the same acf.

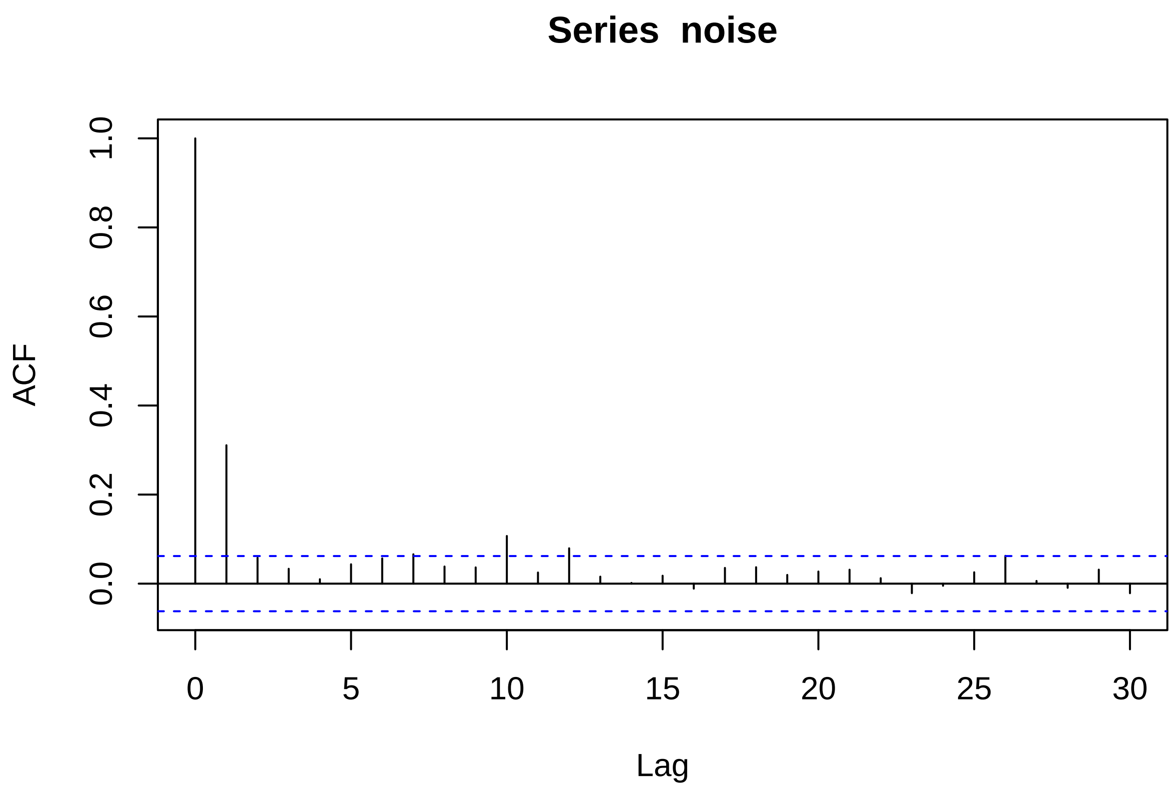
## Question 3

**(3a)**

```
# function to plot and generate correlogram of Xt = alpha Xt-1 + Zt
simulation_2 <- function(alpha) {
  noise <- arima.sim(list(ar = c(alpha)), n = 1000, sd = 0.2)
  plot(noise)
  acf(noise)
}

# run function with alpha = 0.3
simulation_2(0.3)
```


noise
0.6
0.4
0.2
−0.2
−0.6
0
200
400
600
800
1000
Time

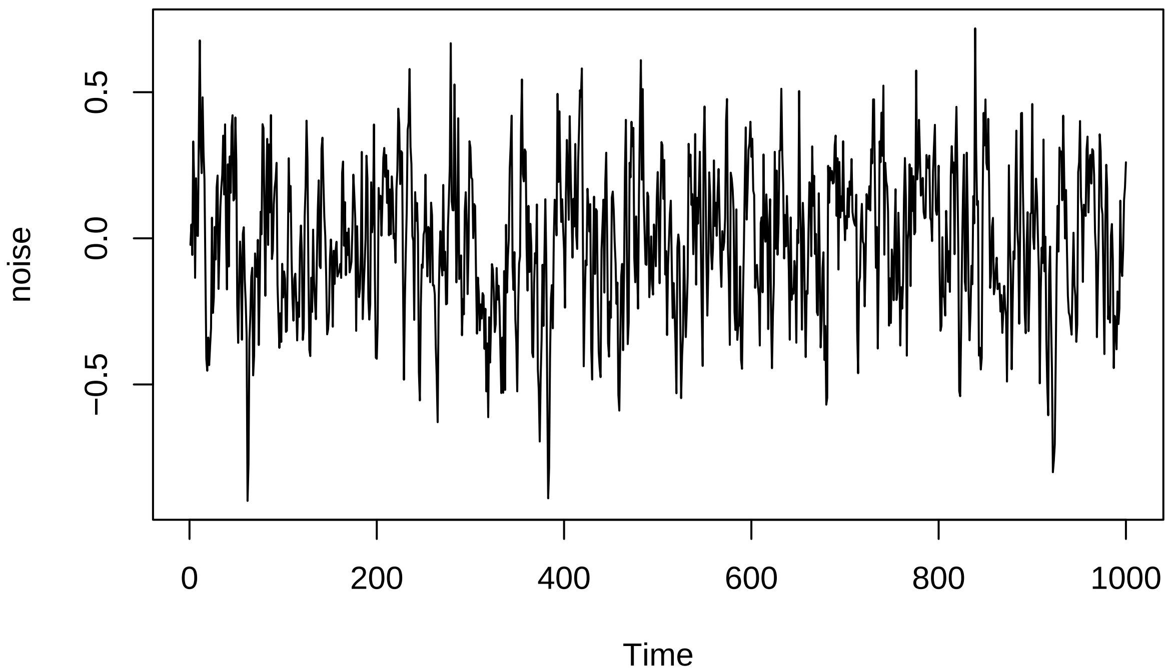



The sample acf at lag $= 1$ is about 0.3, the value of $\alpha$. For lags greater than 1, the sample acf is about 0.

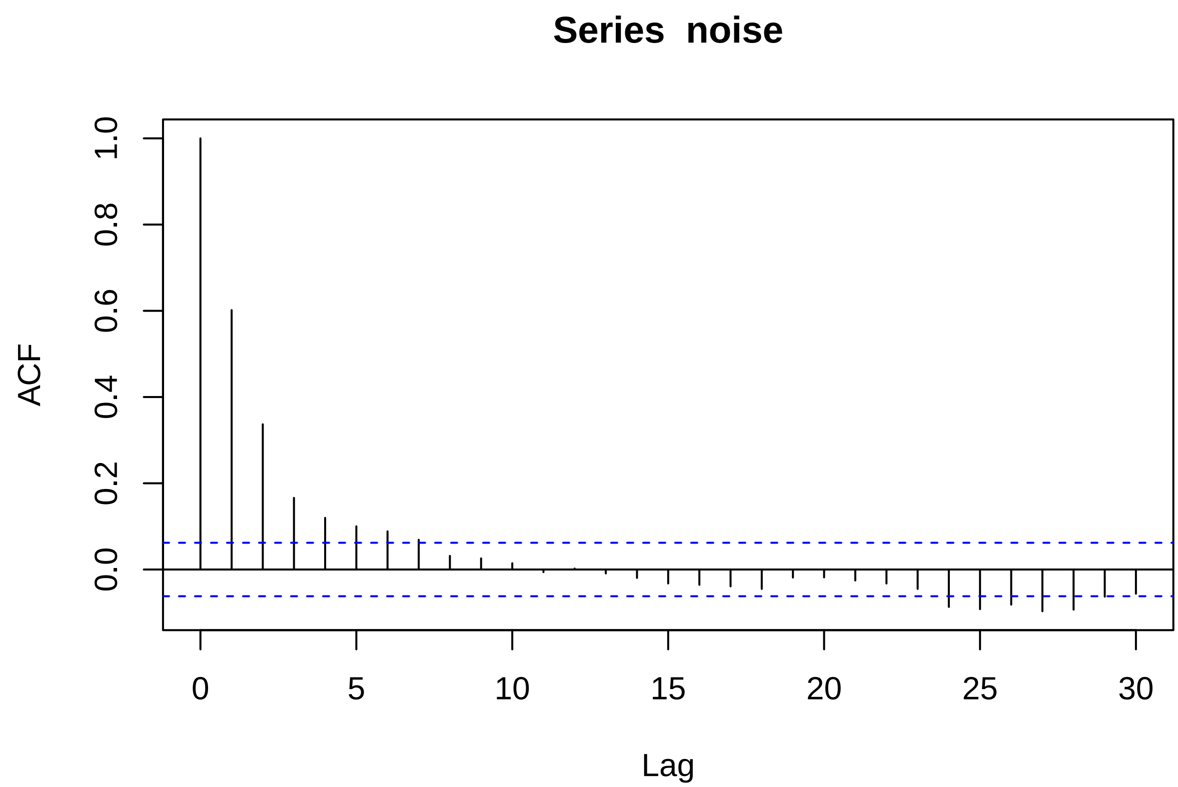
**(3b)**

```
# run simulation 3 more times with increasing alpha and plot acf's
for (alpha in c(0.6, 0.9, 0.99)) {
  cat("Alpha =", alpha, "simulation")
  simulation_2(alpha)
}
```

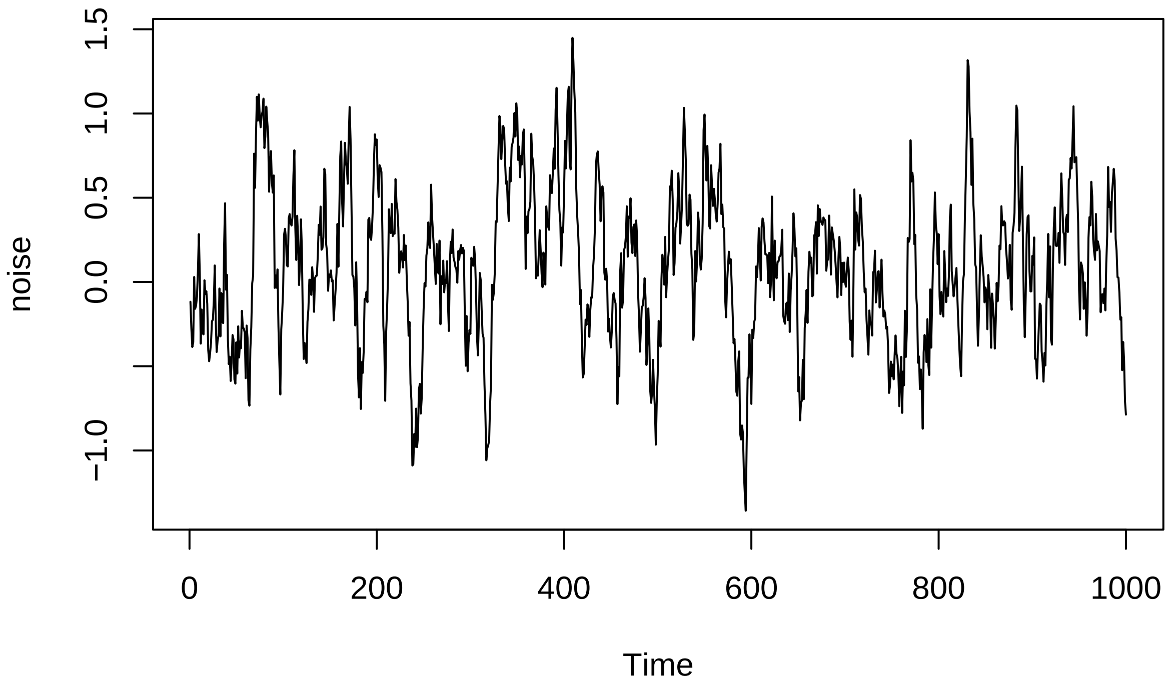
```
## Alpha = 0.6 simulation
```



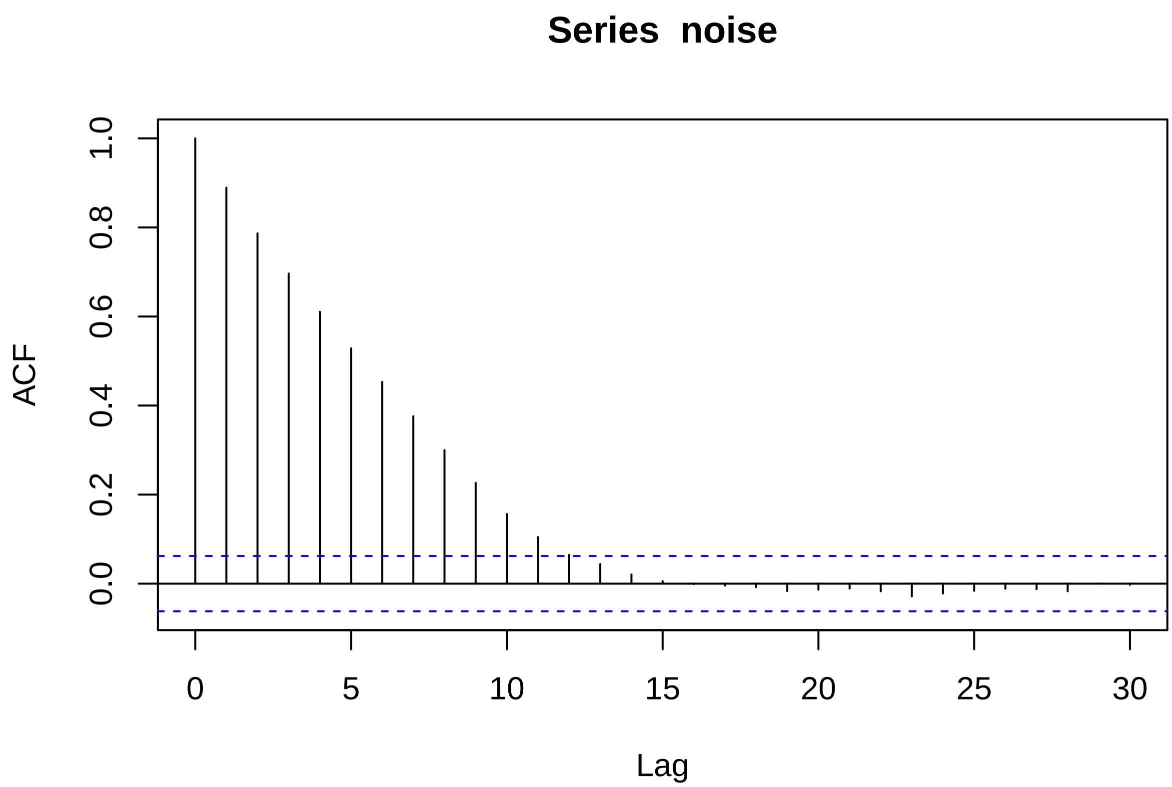
noise
0.5
0.0
−0.5
0
200
400
600
800
1000
Time



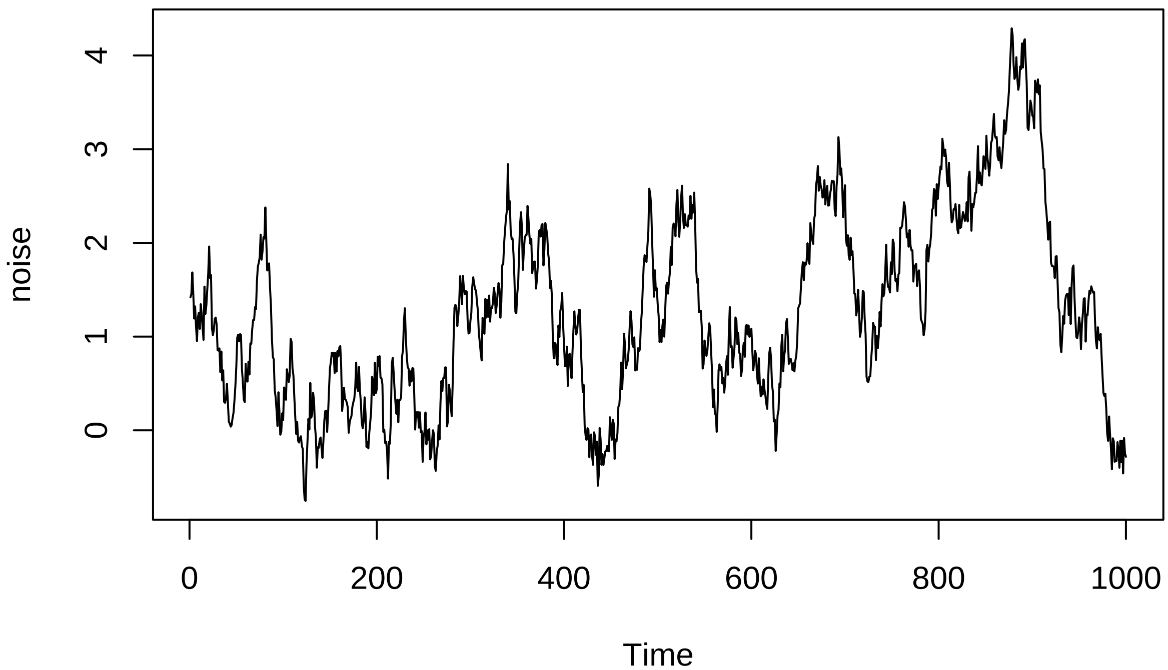

```
## Alpha = 0.9 simulation
```



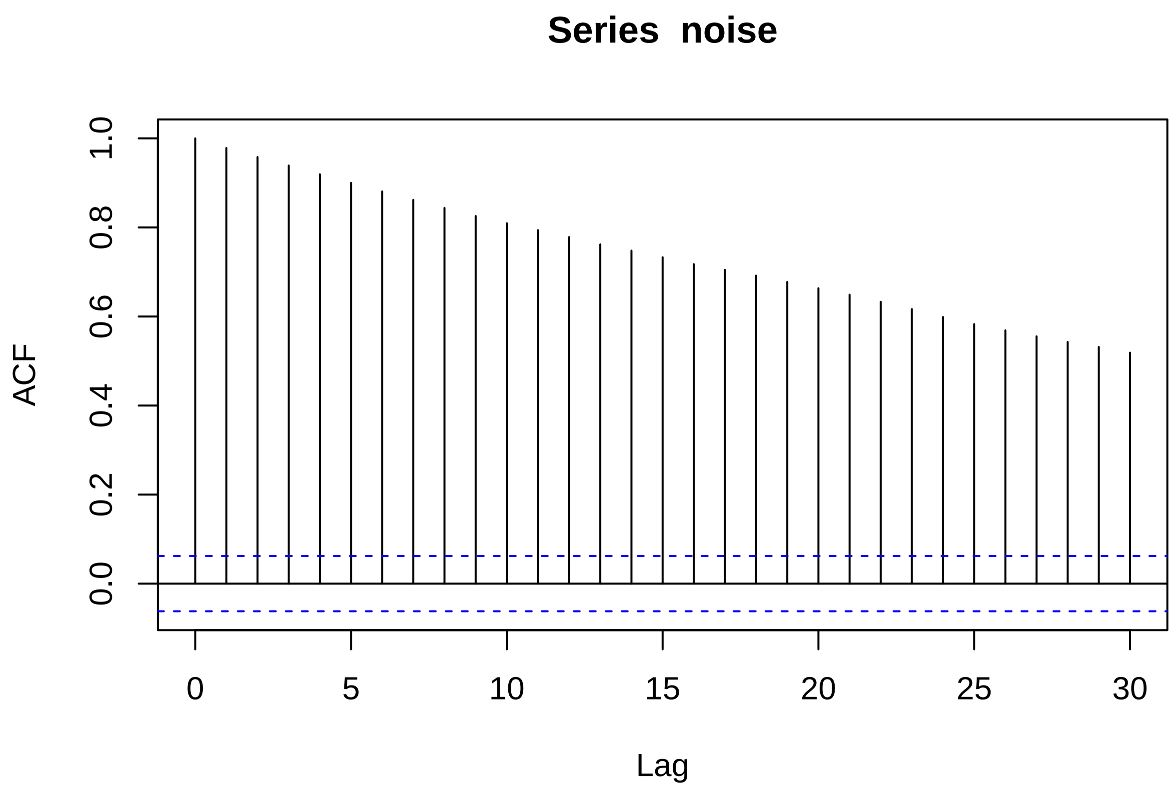
noise
1.5
1.0
0.5
0.0
−1.0
0
200
400
600
800
1000
Time




```
## Alpha = 0.99 simulation
```



noise
4
3
2
1
0
0
200
400
600
800
1000
Time

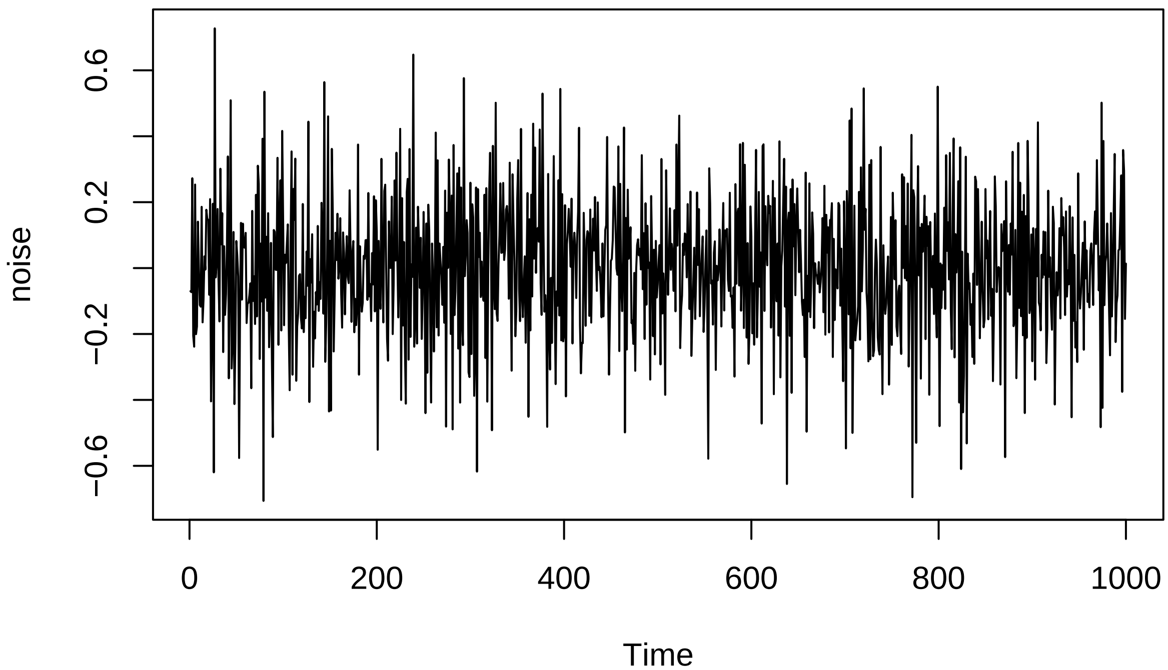



As shown in the repeated simulations, as alpha increases to 1, the acf has a slower decay. This is due to the higher correlation between any $X_t$ and $X_{t-1}$.

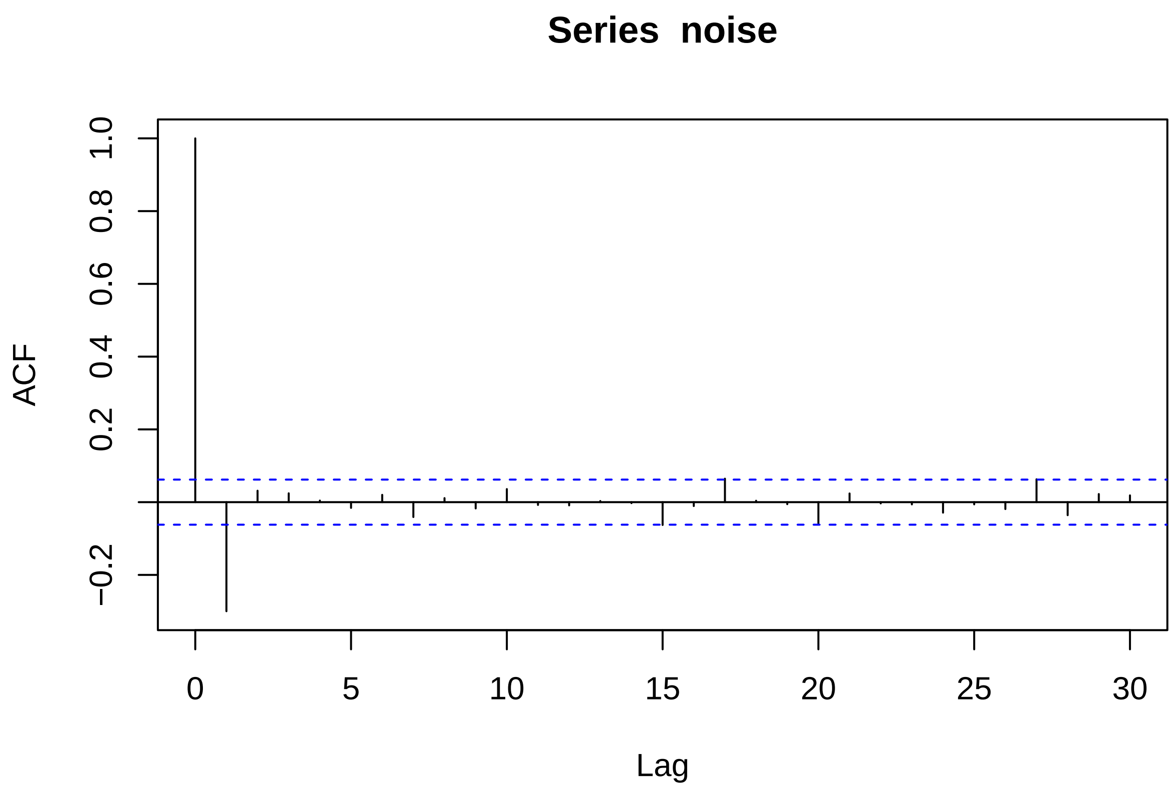
**(3c)**

```
# run simulation 3 more times with alpha approaching -1 and plot acf's
for (alpha in c(-0.3, -0.6, -0.9, -0.99)) {
  cat("Alpha =", alpha, "simulation")
  simulation_2(alpha)
}
```

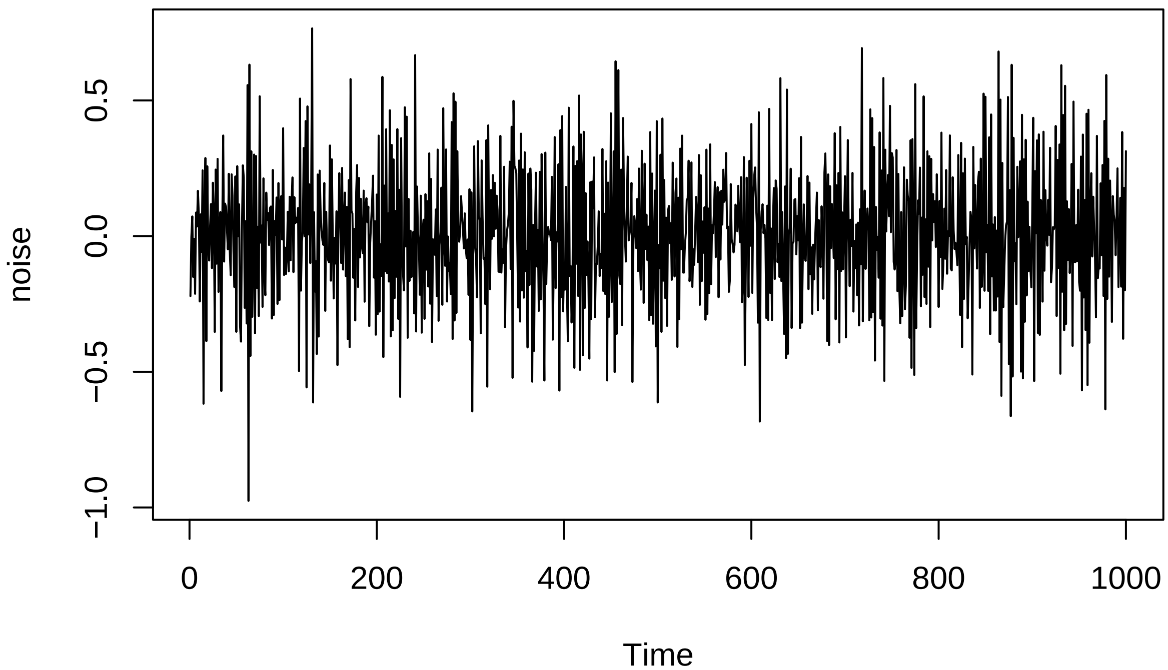
```
## Alpha = -0.3 simulation
```

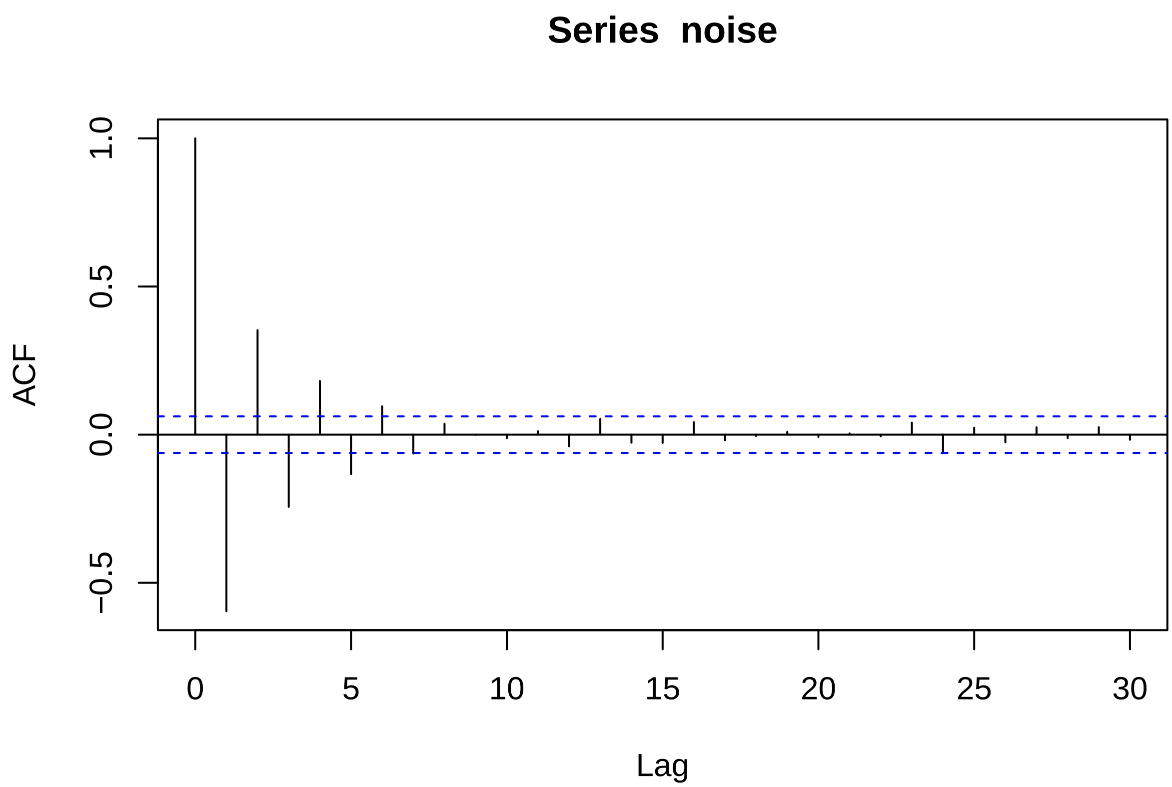


noise
0.6
0.2
−0.2
−0.6
0
200
400
600
800
1000
Time

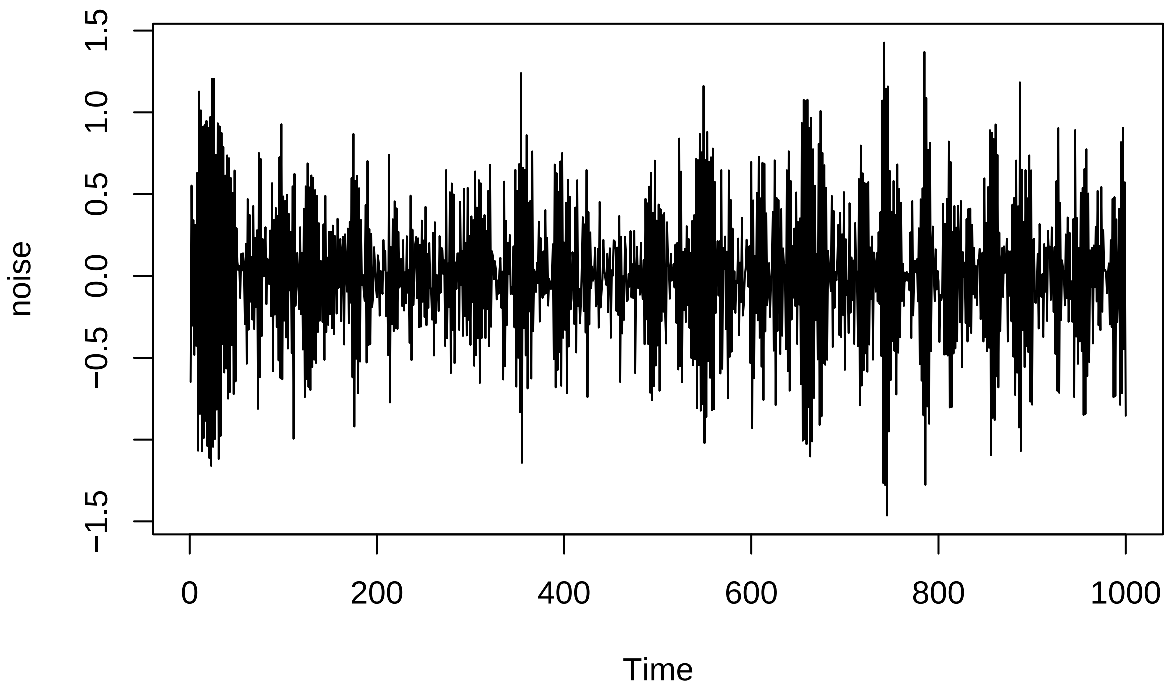


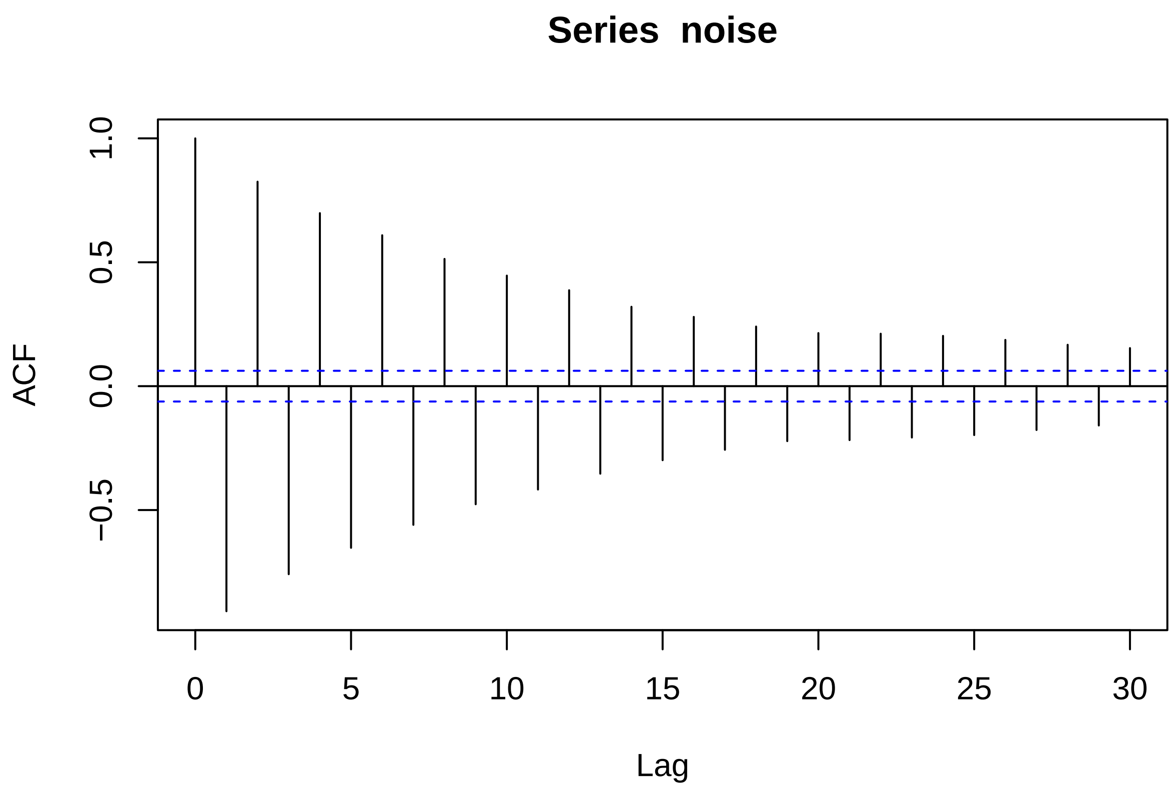

```
## Alpha = -0.6 simulation
```


0.5
0.0
−0.5
−1.0
noise
0
200
400
600
800
1000
Time

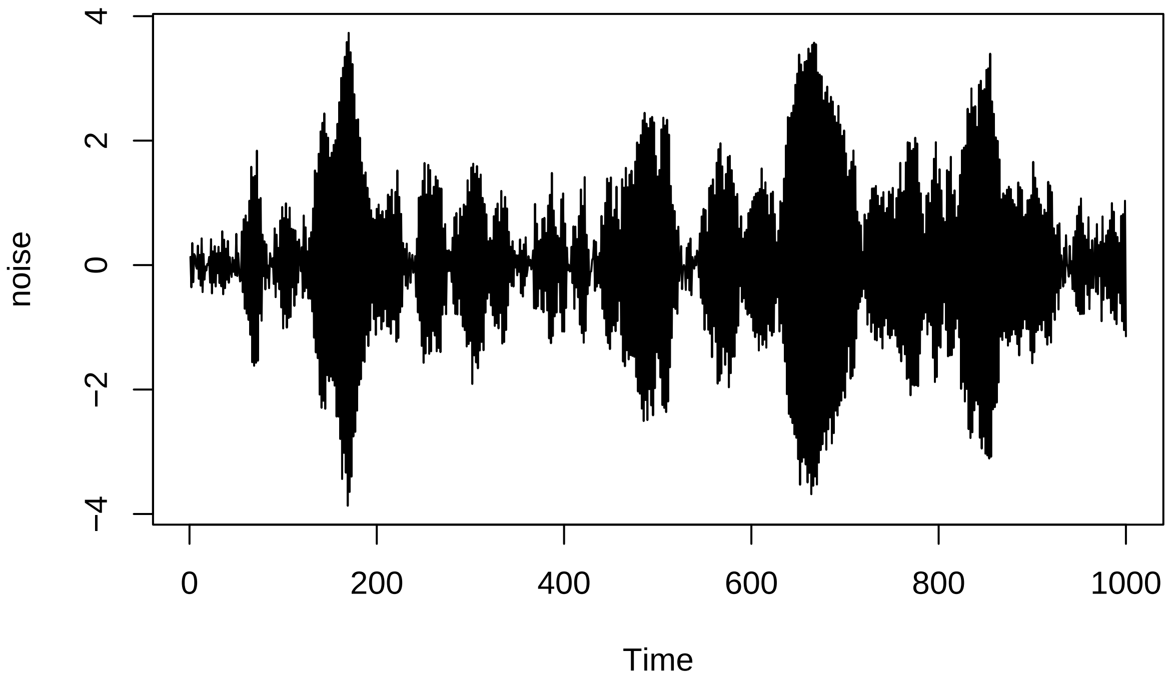



```
## Alpha = -0.9 simulation
```

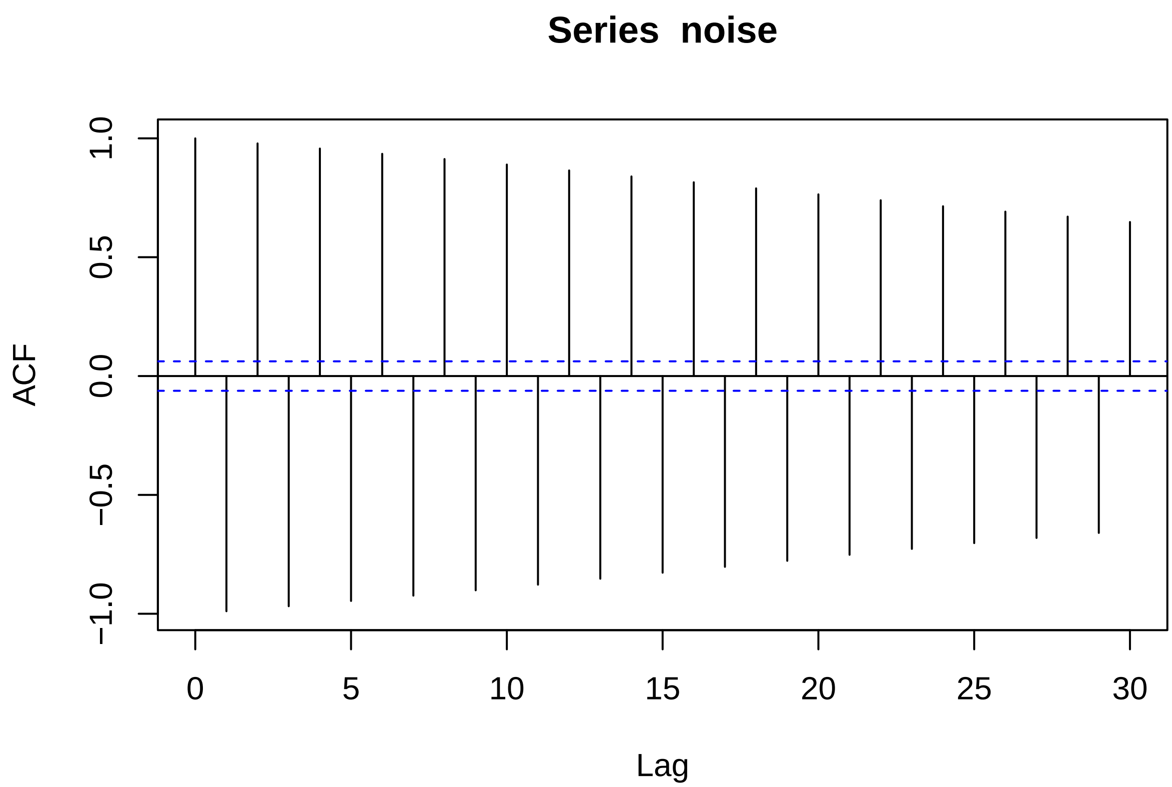

noise
1.5
1.0
0.5
0.0
−0.5
−1.5
0
200
400
600
800
1000
Time




```
## Alpha = -0.99 simulation
```



noise
4
2
0
−2
−4
0
200
400
600
800
1000
Time




With negative values of $\alpha$, the magnitudes of the sample acf's match those with positive values of $\alpha$. The difference with negative values of $\alpha$ is that the acf oscillates around 0 at each lag, since any $X_t$ is negatively correlated with $X_{t-1}$.