ST565: Time Series HW4

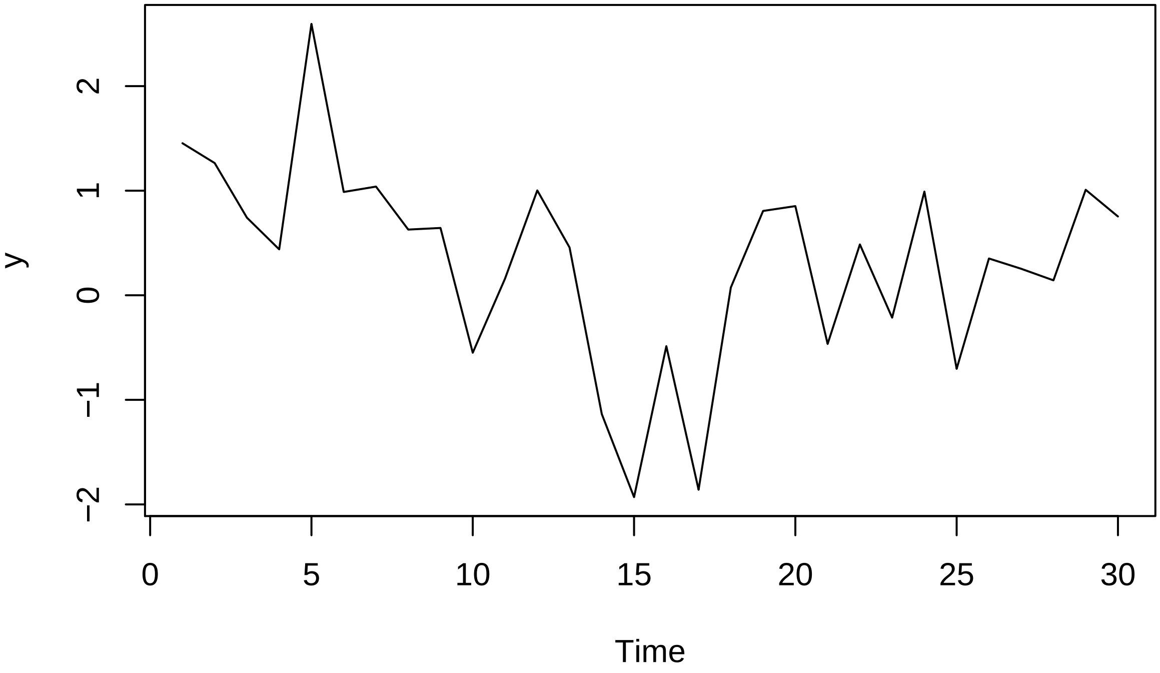
Amirhosein "Emerson" Azarbakht azarbaka@oregonstate.edu

Question 1

Simulate an AR(1) process (you pick the parameter value) of length 30. To this simulated series, fit AR models of order 0 through 6. Which model has the smallest AIC?

I chose the parameter value $\alpha = 0.2$ and the best model was an AR(1). However, depending on the data, if you do this several times, you may not get the same AR(1) model as the best model.

```
# library(dplyr)
# ?arima.sim

# simulate an AR(p)
arP = 1
arLength = 30
arParameterAlpha = 0.2

y = arima.sim(n = arLength, model = list(ar=c(arParameterAlpha)))

# plot the simulated AR(p) TS data
plot(y)
```



```
best.order <- c(0, 0, 0)
best.aic <- Inf

for (i in 0:6){
  fit.aic <- AIC(arima(y, c(i,0,0)))
  if (fit.aic < best.aic) {
    best.order <- i
    best.aic <- fit.aic
  }
}

best.aic
```

```
## [1] 82.55872
```

```
best.order
```

```
## [1] 1
```

Repeat the simulation 500 times, how often is the model with the lowest AIC the true model?

For length = 30, the model with the lowest AIC is the true model in 63.8% of the time.

```
# simulate an AR(p)
startTime <- proc.time()

arP = 1
arLength = 75
arParameterAlpha = -0.8
countTrueModel = 0
numberOfSimulations = 500


for(j in 1:numberOfSimulations){
  y = arima.sim(n = arLength, model = list(order = c(1,0,0), ar= arParameterAlpha))

  best.order <- c(0, 0, 0)
  best.aic <- Inf
  for (i in 0:6){
    fit.aic <- AIC(arima(y, c(i,0,0), method = "ML"))
    # failwith(NA, arima(y, c(i,0,0))
    # fit.aic <- AIC(failwith(NA, arima(y, c(i,0,0)))
    if (fit.aic < best.aic) {
      (best.order <- i)
      best.aic <- fit.aic
    }
  }
  best.order
  if(best.order == arP){
    countTrueModel= countTrueModel+1
  }
}

(trueModelPercentage <- (100*countTrueModel)/numberOfSimulations)
```

```
## [1] 71.2
```

```
(elapsedTime <- proc.time() - startTime)
```

```
##    user  system elapsed
##  63.141   0.074  65.955
```

Repeat again with a longer time series, do things change?

With arParameterAlpha = 0.6

- For length = 30, the model with the lowest AIC is the true model in 63.8% of the time.
- For length = 50, the model with the lowest AIC is the true model in 69.8% of the time.
- For length = 75, the model with the lowest AIC is the true model in 71.4% of the time.
- For length = 100, the model with the lowest AIC is the true model in 73.2% of the time.
- For length = 200, the model with the lowest AIC is the true model in 72.0% of the time.
- For length = 500, the model with the lowest AIC is the true model in 72.0% of the time.

So, the accuracy increases with the longer time series, which suggests that the more data we have about the time series, the more likely it is to guess the correct p for our AR(p) model.

Repeat with a different parameter value, do things change?

For length = 75

- For AR parameter value $\alpha = 0.9$, the model with the lowest AIC is the true model in 71.0% of the time.
- For AR parameter value $\alpha = 0.6$, the model with the lowest AIC is the true model in 70.6% of the time.
- For AR parameter value $\alpha = 0.3$, the model with the lowest AIC is the true model in 58.6% of the time.
- For AR parameter value $\alpha = 0.1$, the model with the lowest AIC is the true model in 18.6% of the time.
- For AR parameter value $\alpha = -0.1$, the model with the lowest AIC is the true model in 24.8% of the time.
- For AR parameter value $\alpha = -0.5$, the model with the lowest AIC is the true model in 73% of the time.
- For AR parameter value $\alpha = -0.8$, the model with the lowest AIC is the true model in 72% of the time.

So, the accuracy decreases with lower absolute values of α, which is intuitive, because the reliance of x_t on x_{t-1} decreases, hence the AR(1) model gets weaker in explaining this weak dependence.

Question 2

Find and fit an ARIMA model to the deere2 dataset:

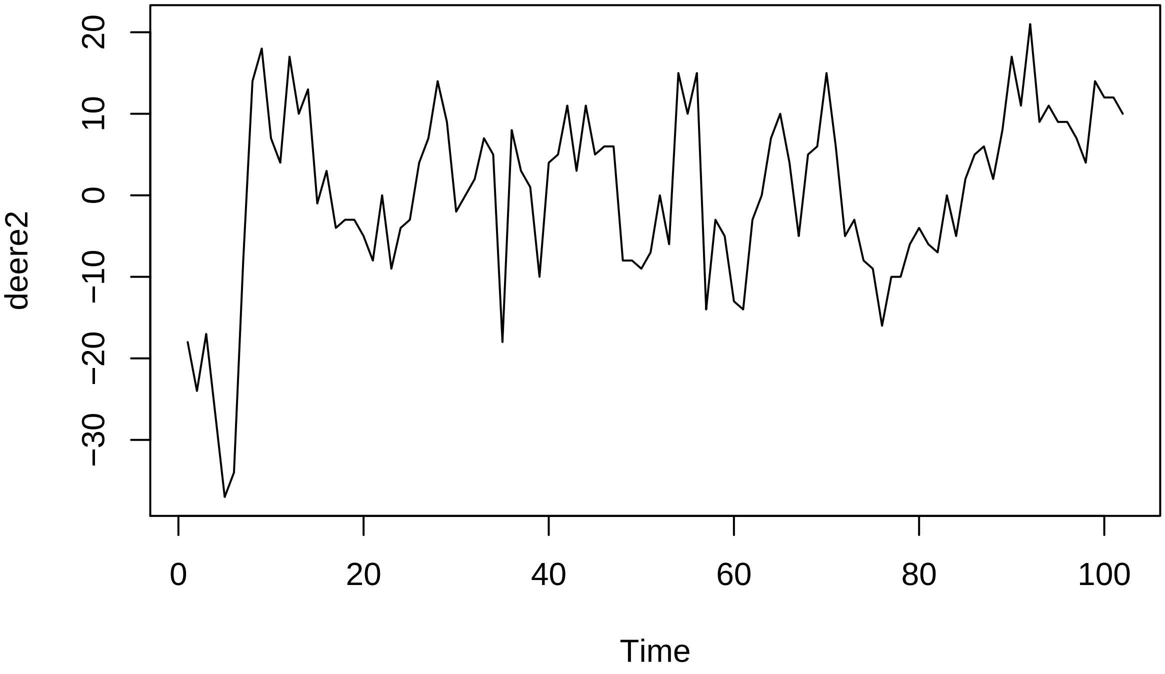
```
library(TSA)
```

```
## Loading required package: leaps
```

```
## Loading required package: locfit
```

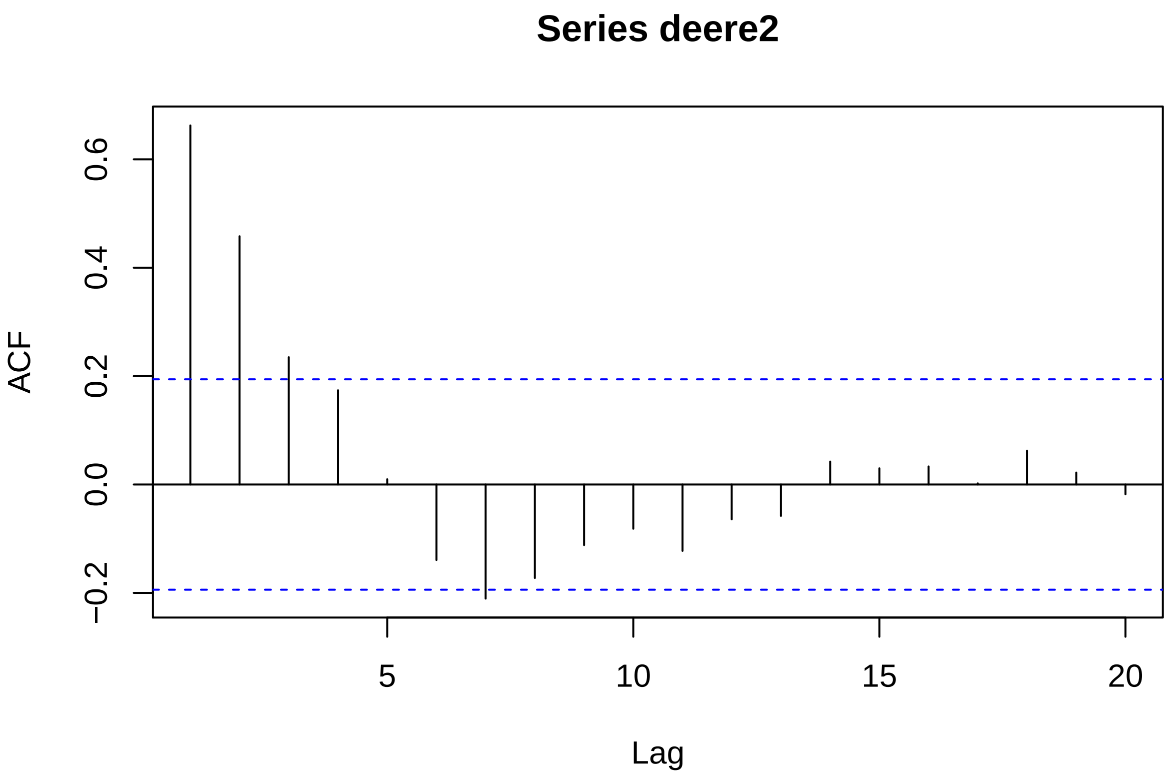
```
## locfit 1.5-9.1    2013-03-22
## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-7. For overview type 'help("mgcv-package")'.

## Loading required package: tseries

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```
# Find and fit an ARIMA model to the robot dataset:
data(deere2)
?deere2
plot(deere2)
```

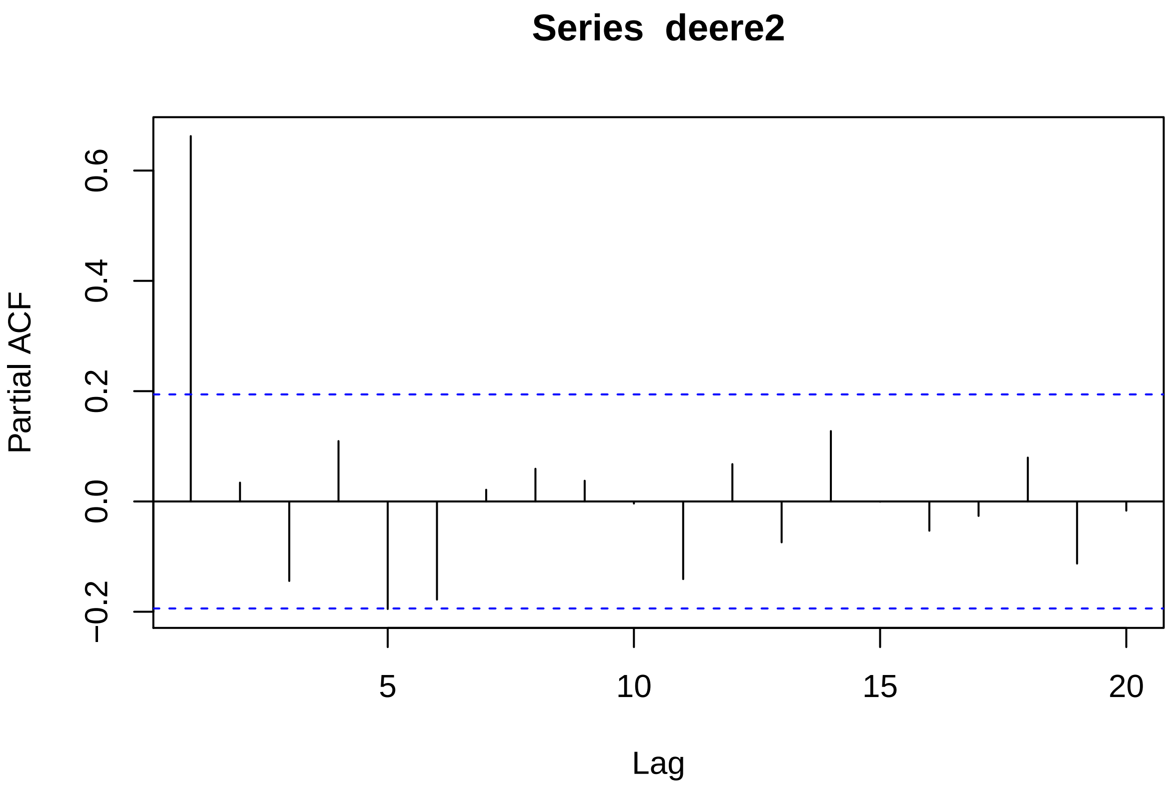


The plot of the deere2 data shows that the times series is non-stationary. So, we plot the ACF function, and the partial ACF.

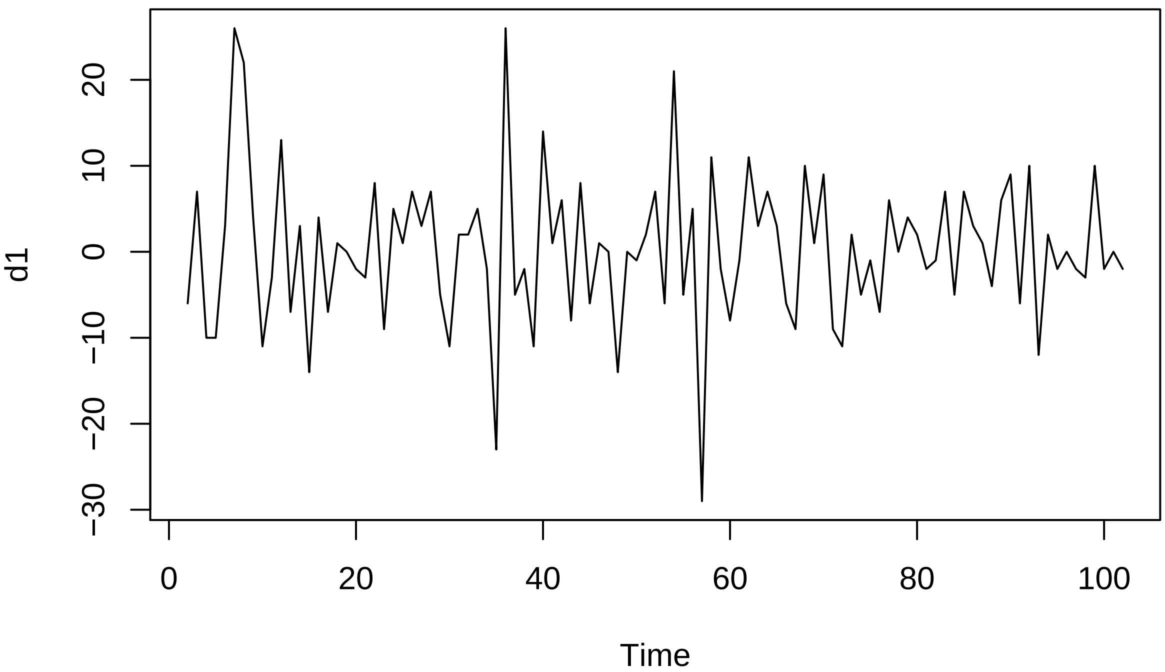
```
acf(deere2)
```


Series deere2
ACF
0.6
0.4
0.2
0.0
−0.2
5
10
15
20
Lag

```
pacf(deere2)
```


Series deere2
Partial ACF
0.6
0.4
0.2
0.0
−0.2
5
10
15
20
Lag

```r
# differencing to find a proper d
d1 <- diff(x = deere2, lag = 1, differences = 1)
plot(d1)
```



With The ACF plot shows that an ARIMA model would be good fit.

```r
best.order <- c(0, 0, 0)
best.aic <- Inf
for (p in 0:5) for (q in 0:5) for (d in 1:3) {
  fit.aic <- AIC(arima(deere2, order = c(p, d, q), method = "ML"))
  if (fit.aic < best.aic) {
    best.order <- c(p, d, q)
    best.arma <- arima(deere2, order = best.order, method = "ML")
    best.aic <- fit.aic
  }
}
```

```
## Warning in stats:::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

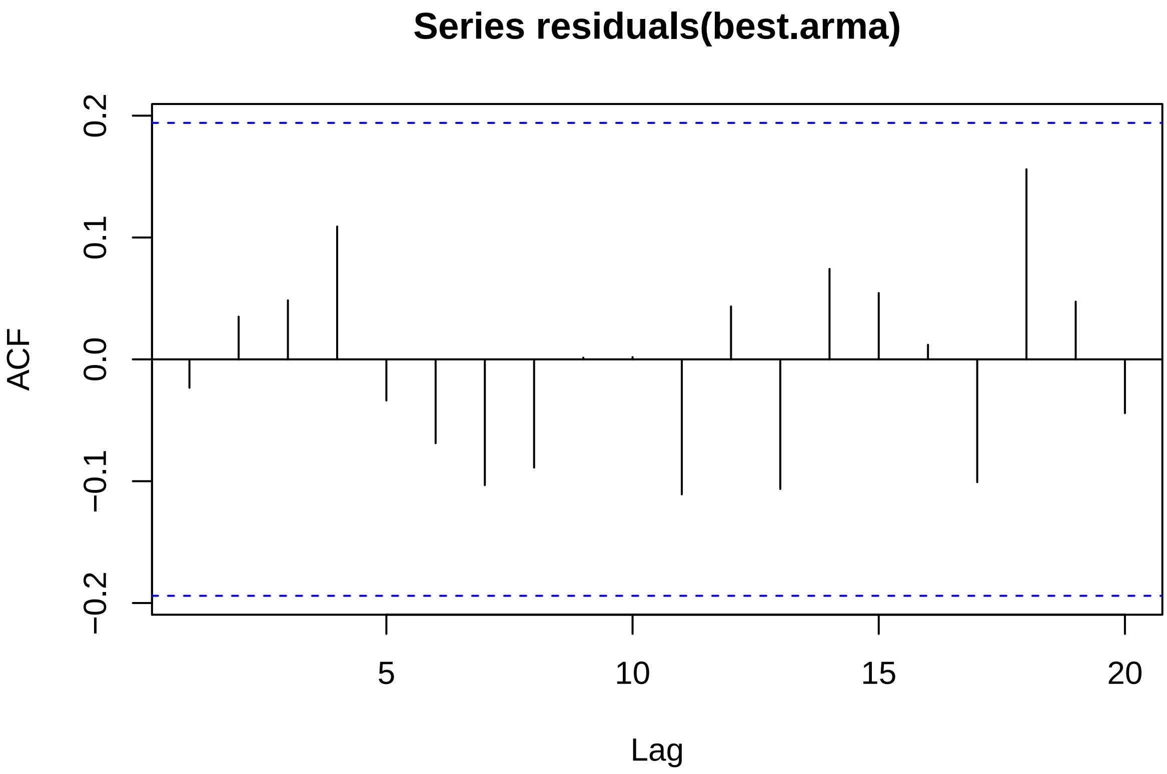
```r
best.arma
```

```
##
## Call:
## arima(x = deere2, order = best.order, method = "ML")
##
## Coefficients:
##           ma1      ma2      ma3
##       -0.2574  -0.1098  -0.4005
## s.e.   0.0986   0.1041   0.1052
##
## sigma^2 estimated as 63.19:  log likelihood = -353.15,  aic = 712.31
```

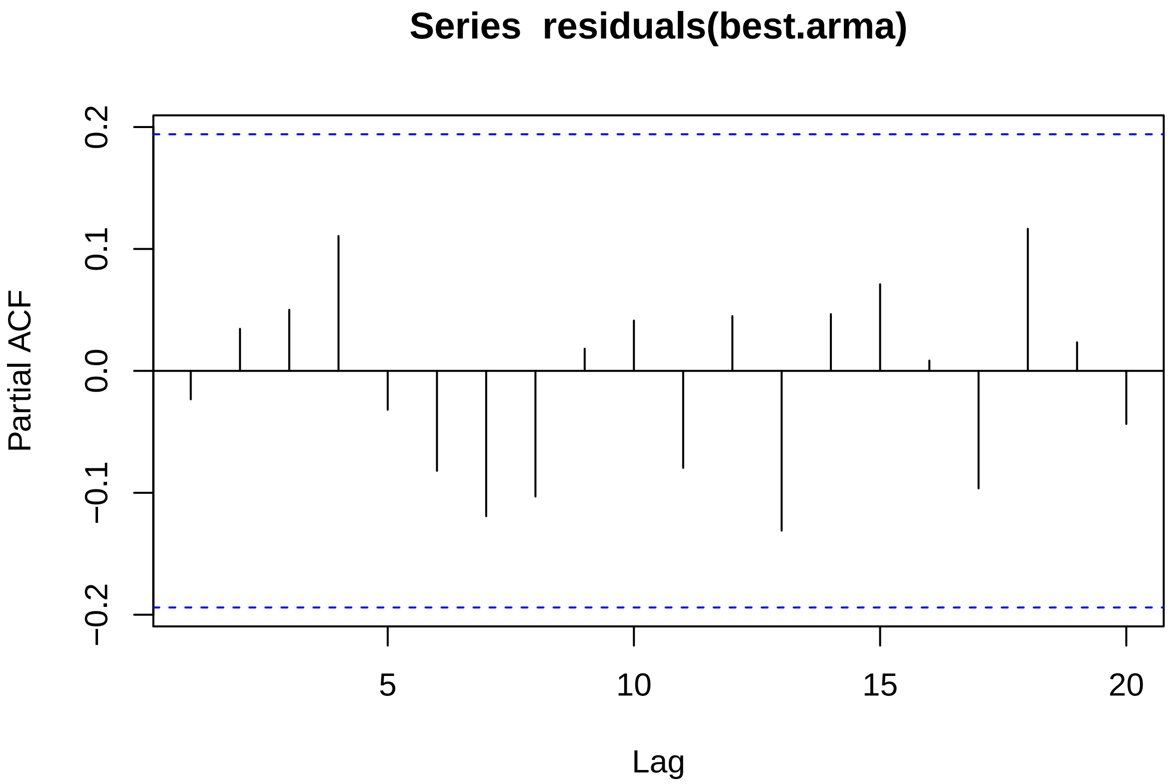
```
best.order
```

```
## [1] 0 1 3
```

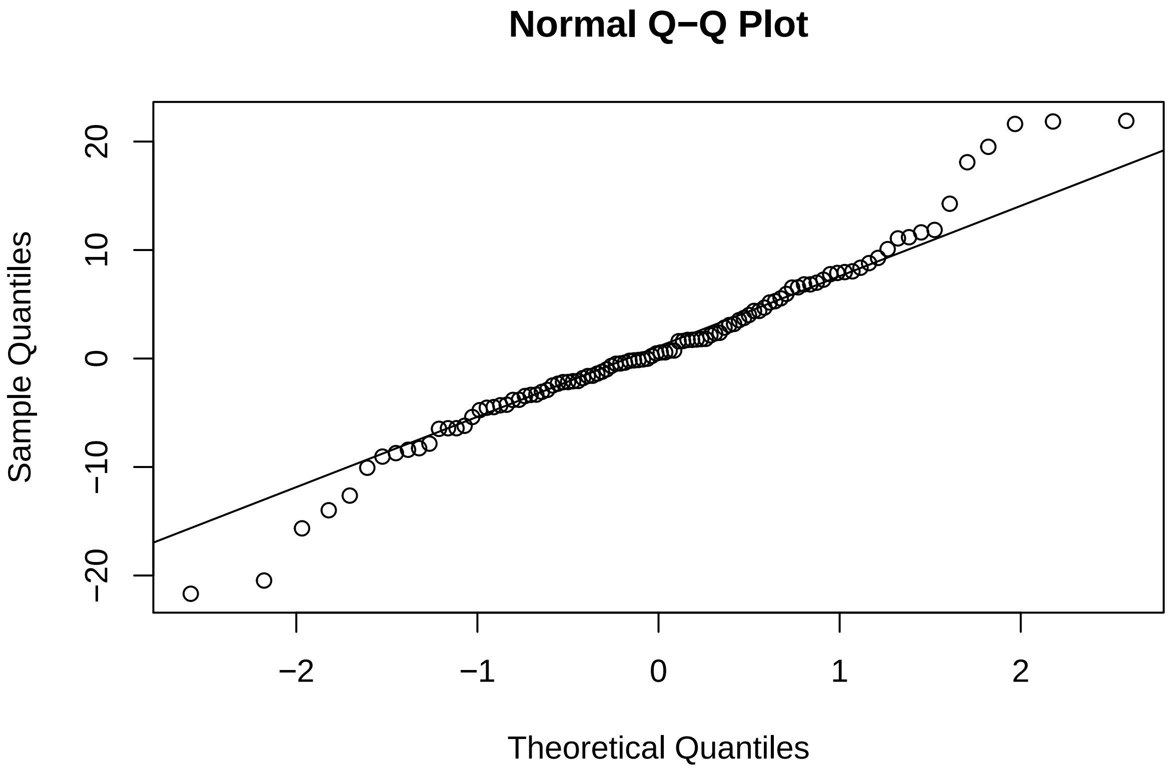
```
# diagnostics
# is there any correlation left in the residuals
acf(residuals(best.arma))
```



```
pacf(residuals(best.arma))
```



```
# check normality
qqnorm(residuals(best.arma))
qqline(residuals(best.arma))
```



I tried experimenting with different p and q values, and looked for the lowest AIC, to find the best-fitting ARIMA model. I checked the ACF and the PACF of the residuals of the best-fitting model, and they looked good. The QQ-plot of the best-fitting model looks normal, as well. So, an ARIMA(0,1,3) is a good model for this data.

Find and fit an ARIMA model to the robot dataset:

```
# Find and fit an ARIMA model to the robot dataset:
data(robot)
?robot

best.order <- c(0, 0, 0)
best.aic <- Inf
for (p in 0:4) for (q in 0:4) for (d in 1:3) {
  fit.aic <- AIC(arima(robot, order = c(p, d, q), method = "ML"))
  if (fit.aic < best.aic) {
    best.order <- c(p, d, q)
    best.arma <- arima(robot, order = best.order, method = "ML")
    best.aic <- fit.aic
  }
}
best.arma
```

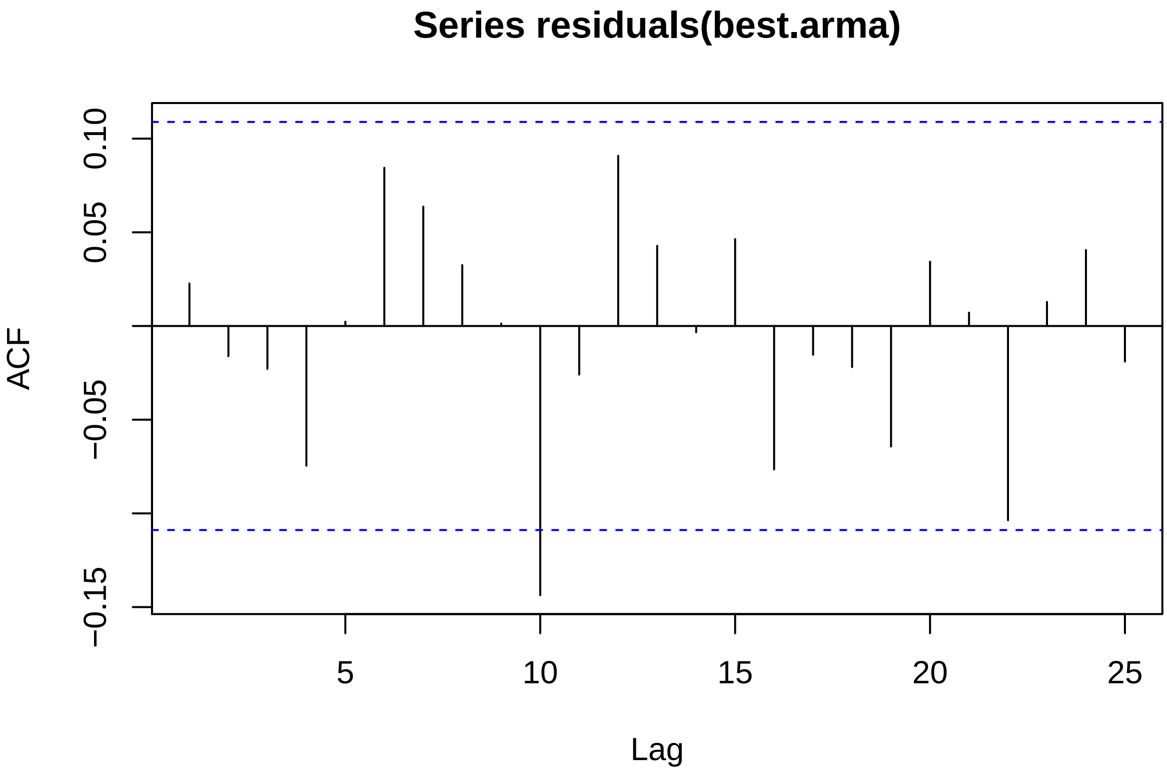
```
##
## Call:
## arima(x = robot, order = best.order, method = "ML")
##
## Coefficients:
```

```
##          ar1      ma1     ma2
##       0.8333  -1.6751  0.6831
## s.e.  0.1146   0.1392  0.1335
##
## sigma^2 estimated as 5.956e-06:  log likelihood = 1483.69,  aic = -2961.38
```

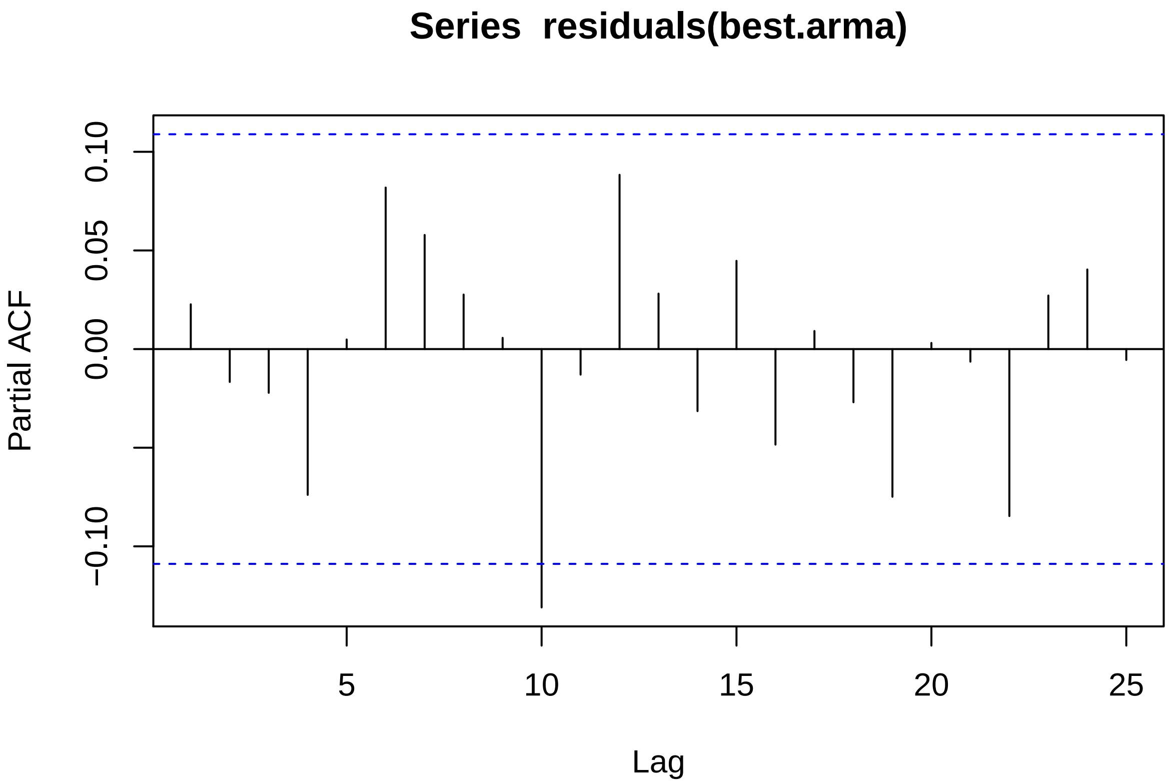
```
best.order
```

```
## [1] 1 1 2
```

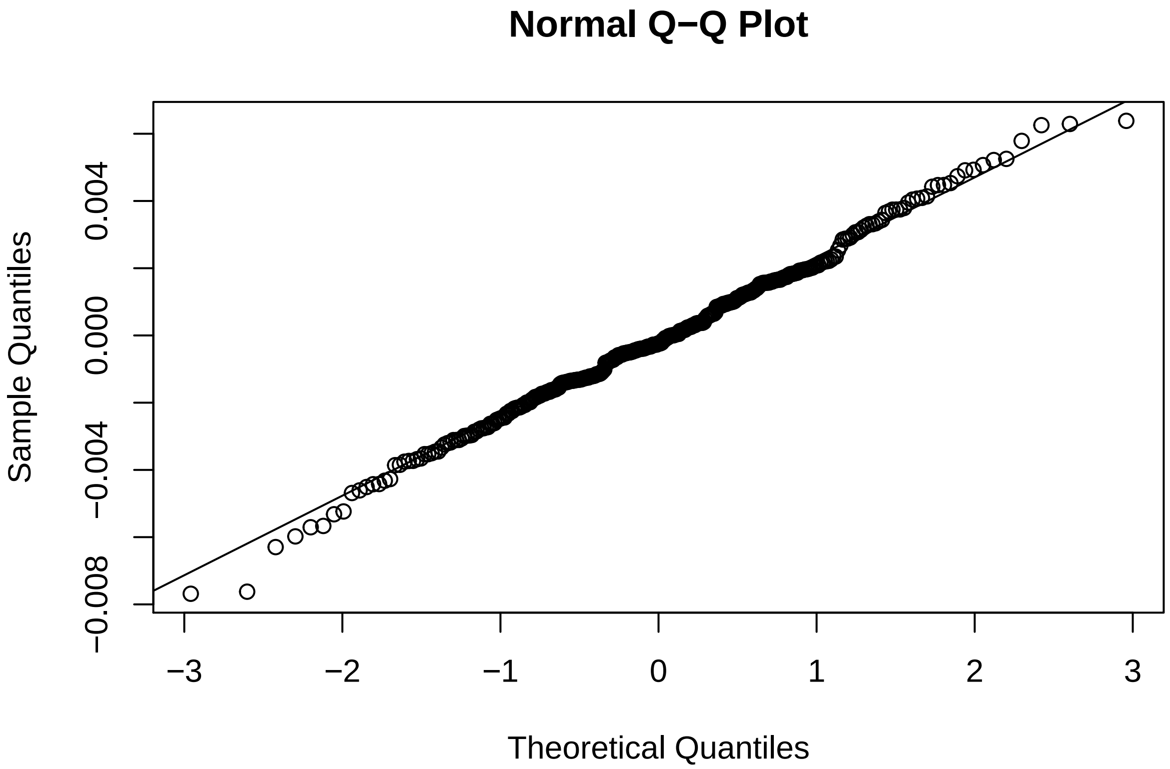
```
# diagnostics
# is there any correlation left in the residuals
acf(residuals(best.arma))
```



```
pacf(residuals(best.arma))
```



```
# check normality
qqnorm(residuals(best.arma))
qqline(residuals(best.arma))
```



I tried experimenting with different p and q values, and looked for the lowest AIC, to find the best-fitting ARIMA model. I checked the ACF and the PACF of the residuals of the best-fitting model, and they looked good. The QQ-plot of the best-fitting model looks normal, as well. So, an ARIMA(1,1,2) is a good model for this data.