

ARMA Model

Kushal Kharel

9/19/2021

The Auto Regressive Moving Average (ARMA) Model. Suppose you are a computer manufacturer and your challenge every month is to figure out how many computers you should make to meet your demand. That value is l_t . Now suppose we decide to model our number of computers using an ARMA(1,1) model. The (1,1) corresponds to the order of AR & MA.

Let $l_t = \beta_0 + \beta_1 l_{t-1} + \phi_1 \epsilon_{t-1} + \epsilon_t \dots$ (i)

The number of computers that you are going to make this month is equals to some coefficient β_0 plus some coefficient β_1 times the number of computers you needed to make last month l_{t-1} which is the autoregressive part plus some coefficient ϕ_1 times ϵ_{t-1} which is our error from the previous time period (last month) plus ϵ_t which is the error in prediction from this month.

Now, If I want to figure out what would I make for my prediction this month then I take \hat{l}_t . In Statistics or Time Series Analysis, \hat{l}_t is our predicted value whose real process is given by equation (i) but of course we don't know the real process because if we could predict the error from this month, we would have all the information we need to make a perfect prediction which is obviously not true. So, my predicted value for computers created this month is going to be equal to,

$\hat{l}_t = \beta_0 + \beta_1 l_{t-1} + \phi_1 \epsilon_{t-1} + \epsilon_t \dots$ (ii),

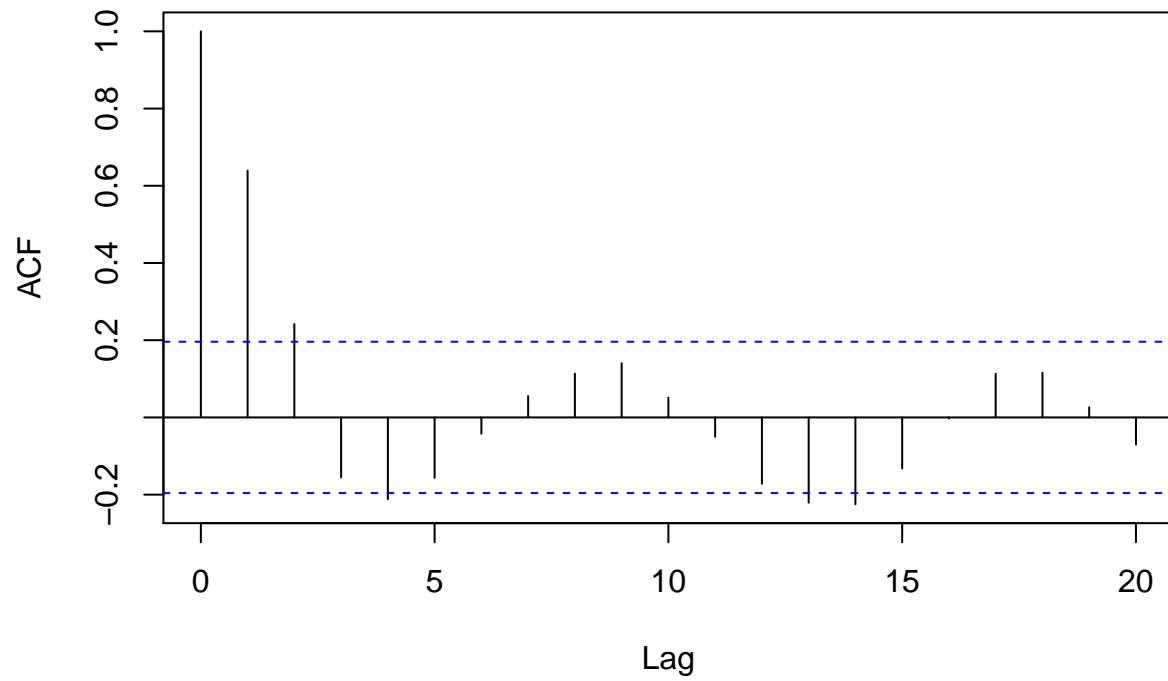
which is equal to coefficient β_0 plus the coefficient β_1 times l_{t-1} , meaning I do have access to the exact number of computers I needed last month because last month is in the past and I do have access to past knowledge plus, $\phi_1 \epsilon_{t-1}$, I also have access to my error from last month because it is past knowledge and I stop here because I do not have access to error from this month because it has not happened yet.

Now, suppose that you see a time series data somewhere online then how do you figure out what order you should set for the AR and the MA bit. It is related to the concept of auto correlation and partial auto correlation.

ACF helps us identify the MA order and PACF helps us identify the AR order. PACF measures the direct effect which is what the autoregressive bit. Let us do some simulation to get an idea of what ACF and PACF plot looks like. The first simulation is MA(2) process, and the second simulation is AR(2) process.

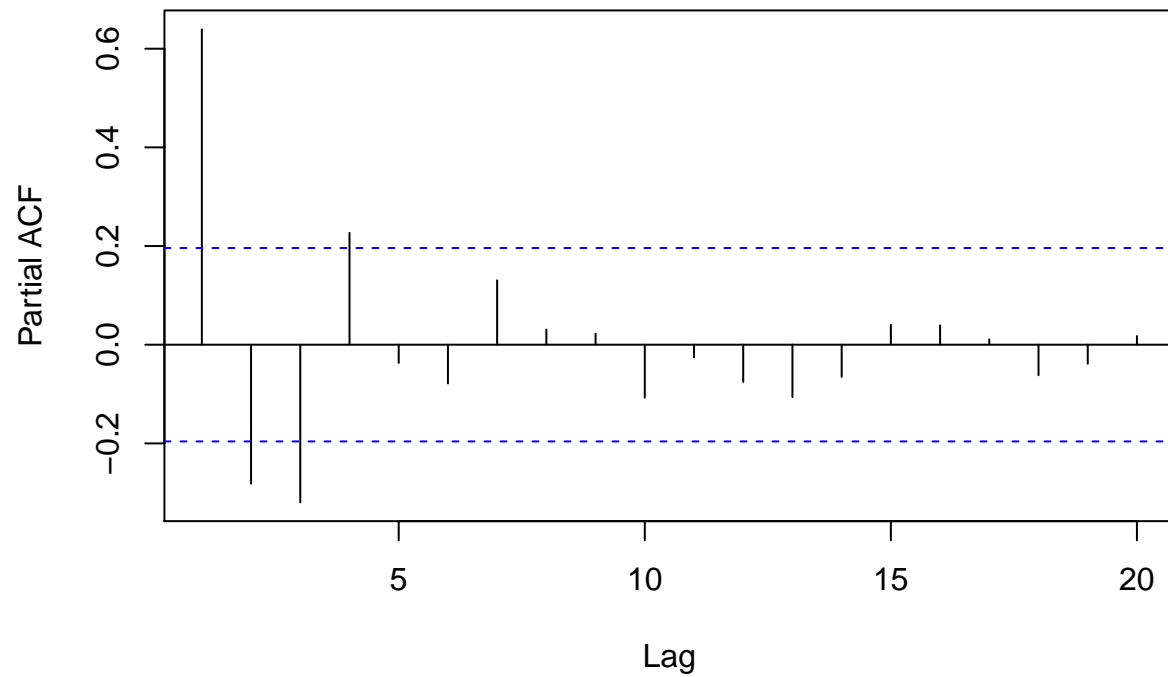
```
set.seed(101)
MA2 <- arima.sim(n = 100, list(ma = c(0.8, 0.5)), sd = 1)
acf(MA2, main = 'MA(2) theta1=0.8, theta2=0.5 ACF')
```

MA(2) $\theta_1=0.8$, $\theta_2=0.5$ ACF



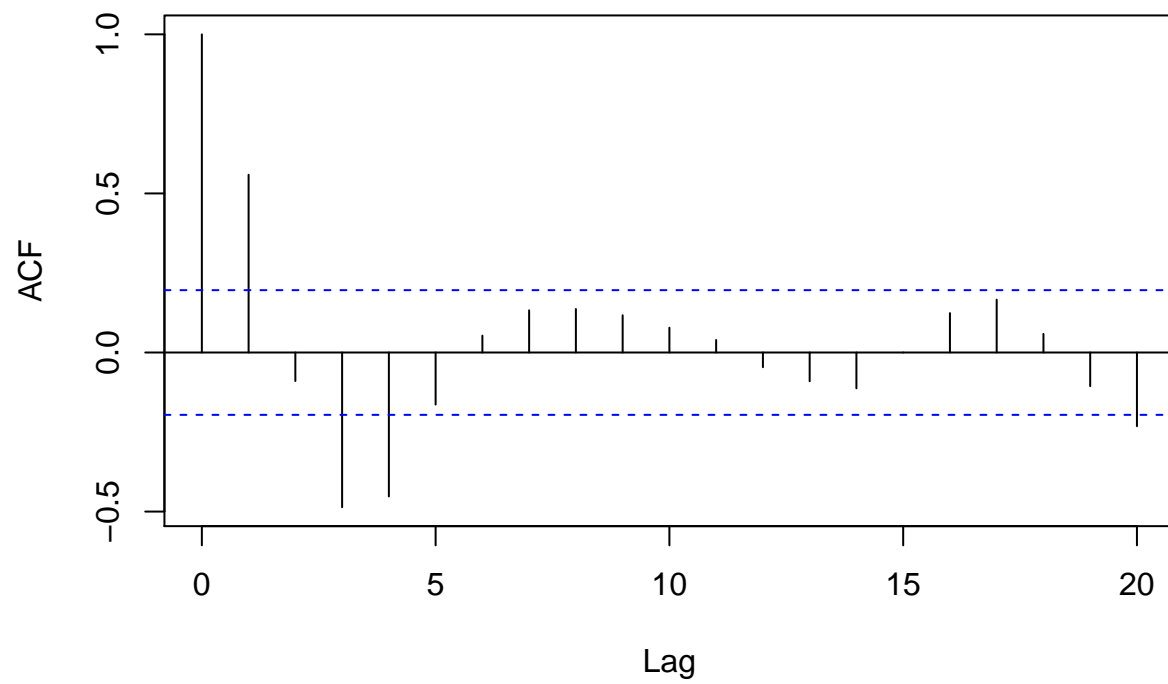
```
pacf(MA2, main = 'MA(2)  $\theta_1=0.8$ ,  $\theta_2=0.5$  PACF')
```

MA(2) $\theta_1=0.8$, $\theta_2=0.5$ PACF



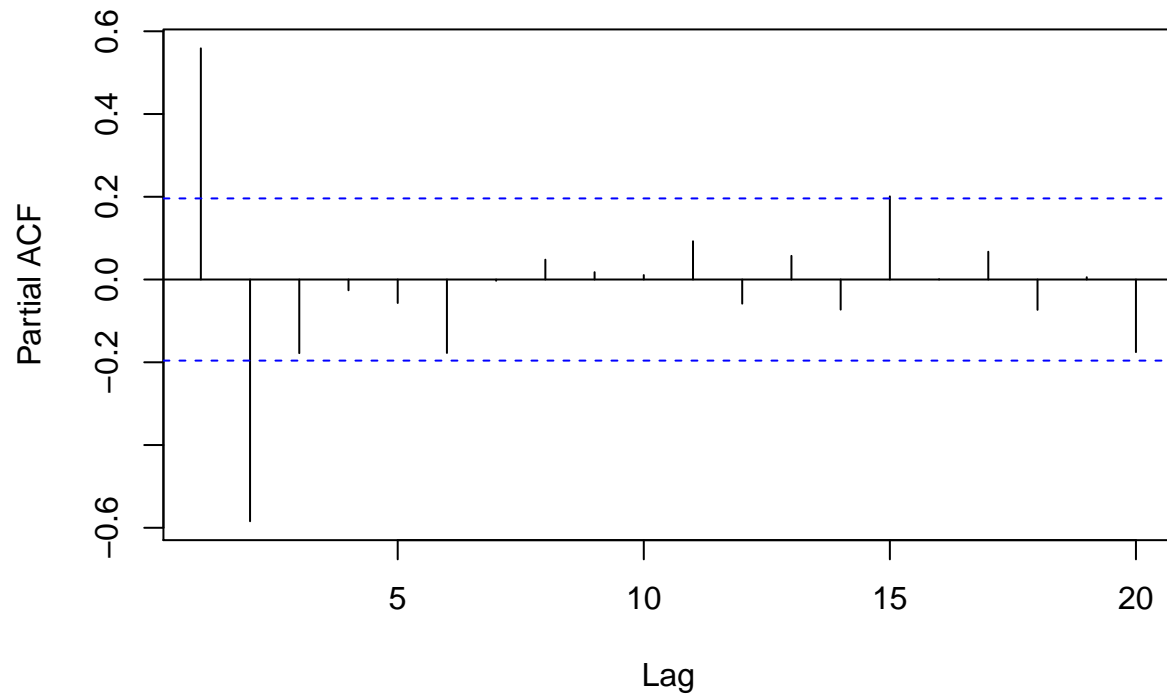
```
set.seed(101)
AR2 <- arima.sim(n = 100, model = list(ar = c(0.8, -0.5)), sd = 1)
acf(AR2, main = 'AR(2)  $\phi_1=0.8$ ,  $\phi_2=-0.5$  ACF')
```

AR(2) $\phi_1=0.8$, $\phi_2=-0.5$ ACF



```
pacf(AR2, main = 'AR(2)  $\phi_1=0.8$ ,  $\phi_2=-0.5$  PACF')
```

AR(2) $\phi_1=0.8$, $\phi_2=-0.5$ PACF



The dotted lines (error bands) says that anything within those bands are not statistically different from zero, no evidence to say it is different from zero. In ACF, anything outside those bands would be telling us the order of the moving average bit. Similarly, in PACF, anything outside those bands would tell us autoregressive bit.

We can see from the above plots that the cutoff happens after lag 2 for each of the corresponding plots. We also can see a slow “tailing-off” which indicates that there is an underlying ARMA(p,q) model.