**SARIMA Models – Introduction and Simulated Example**

Time series data often has a pattern which repeats every s time periods. For example, monthly data may have s = 12.

Seasonal ARIMA (SARIMA) models can be found using the same techniques as discussed previously. A model that only has seasonal components is written as

Φ(Bs)(1-Bs)Dxt = Θ(Bs)wt where wt ~ ind. (0,)

and Φ(Bs)=1-Φ1Bs-Φ2B2s-…-ΦPBPs

Θ(Bs)=1+Θ1Bs+Θ2B2s+…+ΘQBQs

Notice how s is used above! If s = 1, we have the “usual” non-seasonal model. Also, the “Q” here is not for the Ljung-Box-Pierce test!

The seasonal differencing operator (1-Bs)D means that for D = 1, .

In a purely seasonal ARIMA model (no non-seasonal components), the ACF and PACF have similar forms as for the previous non-seasonal ARIMA models. The difference is now only lags h = s, 2s, 3s, … are examined. The remainder of the lags have autocorrelations and partial autocorrelations equal to 0. Below is a table summarizing the results.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **AR(P)s** | **MA(Q)s** | **ARMA(P,Q)s** |
| ACF | Tails off to 0 at lags ks for k=1,2,… | Cuts off to 0 after lag Qs | Tails off to 0 after lag Qs |
| PACF | Cuts off to 0 after lag Ps | Tails off to 0 at lags ks for k=1,2,… | Tails off to 0 after lag Ps |

For example,

|  |  |  |
| --- | --- | --- |
| Seasonal AR(1): (1-Φ1Bs)xt = wt | ρ(sh)= |  |
| ϕsh,sh= |  |

Nonseasonal patterns can be present with seasonal patterns! The ARIMA(p,d,q)×(P,D,Q)s model is

Φ(Bs)ϕ(B)(1-Bs)D(1-B)dxt = Θ(Bs)θ(B)wt

where wt ~ ind. (0,)

The ACF and PACF plots are combinations of the ACF and PACFs from both the seasonal and non-seasonal parts of the model. This can make examining ACF and PACF plots (for determining p, q, P, and Q) quite complicated! Typically, this is the order one proceeds in when trying to determine a SARIMA model:

1. Investigate any nonstationarity problems and try to solve them. For example, nonstationarity in the mean would appear as (s = 4):



1. Determine P and Q; one may need to examine residual ACF and PACF plots in an iterative manner to choose good values.
2. Investigate residual ACF and PACF plots to determine p and q; again, plots for multiple models may need to be examined in an iterative manner.

Seasonal models can become even more complicated.



where

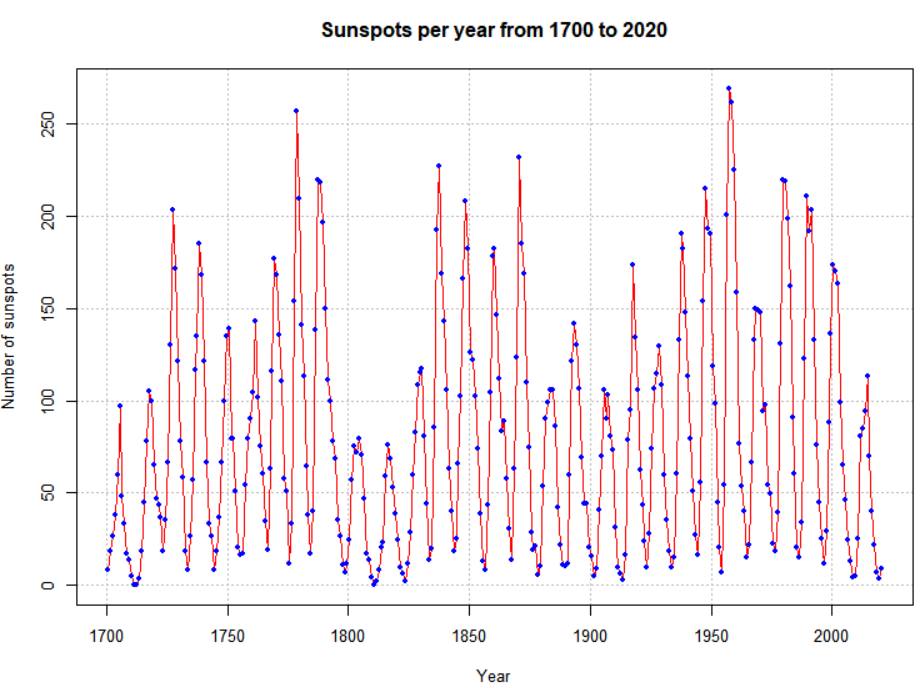
 

Thus, the above representation can incorporate seasonality in more than one way. The Φ and Θ notation is not used so that a more general representation of the model can be given.

Below are a few examples where SARIMA models would be appropriate.

Example: Sunspots (sunspots.R, SN\_y\_tot\_V2.0.csv)

There are approximately a 10-12 year cycle between peaks (highs) and valleys (lows).



s ≈ 10-12

Example: OSU enrollment data (osu\_enroll.R, osu\_enroll.csv)





s = 3

Notes:

* Large estimated autocorrelations are often present surrounding the seasonal lags.
* Strong seasonal variation often causes large estimated autocorrelations at fractional multiples of the seasonal lags.

Example: ARIMA(0,1,1)×(0,1,1)12 (sarima011.011.12.csv, sarima\_011\_011\_12\_sim.R)

Suppose I want to simulate n = 200 observations from the above process with  = 25, μ = 0, θ1 = -0.4, and   
Θ1 = -0.6.

The process can be written so that only xt is on the left side.



Using ARMAacf(), we can see what the ACF and PACF is for model with d = 0, D = 0, q = 1, and Q = 1. Because the function does not have a direct way to input the seasonal portion, we need to multiply out the MA operators, (1 - 0.4B)(1 - 0.6B12) = 1 - 0.4B - 0.6B12 + 0.24B13 and use this form in the ma argument of ARMAacf().

> par(mfrow = c(1,2))

> plot(y = ARMAacf(ma = c(-0.4, rep(x = 0, times = 10),

-0.6, 0.24), lag.max = 50, pacf = FALSE), x = 0:50,

type = "h", ylim = c(-1,1), xlab = "h", ylab =

expression(rho(h)), main = "True ACF for

ARIMA(1,0,1)x(0,0,1)\_12")

> abline(h = 0)

> plot(x = ARMAacf(ma = c(-0.4, rep(x = 0, times = 10),

-0.6, 0.24), lag.max = 50, pacf = TRUE), type =

"h", ylim = c(-1,1), xlab = "h", ylab =

expression(phi1[hh]),main = "True PACF for

ARIMA(1,0,1)x(0,0,1)\_12")

> abline(h = 0)



To simulate data from the model, arima.sim()can be used; however, not in as easy a way as one would expect. Below are a few notes:

* Multiply out the AR and MA operators and incorporate these into onelist() statement in arima.sim().
* Program the model into R and use the for() function to loop over its components in a similar manner to what we did for the AR(1) example.
* Use a user-written package. The astsa package includes the sarima.sim() function for this purpose. An example is given at the end of the program.

I simulated observations from the model in the past and put them into a comma delimited file. I am going to use this data for the rest of the example. An additional 24 additional observations were simulated at the end of the t = 1, …, 200 series so that forecasted values can be compared to the true future values. Below is the code needed to read and plot the data.

> sarima.data <- read.csv(file = "sarima011.011.12.csv")

> head(sarima.data)

x time

1 -167.6940 1

2 -171.0745 2

3 -168.8319 3

4 -172.4671 4

5 -172.4534 5

6 -191.5866 6

> x <- sarima.data$x[1:200]

> #Plot of the data

> plot(x = x, ylab = expression(x[t]), xlab = "t", type

= "l", col = "red", main = "Data simulated from

ARIMA(0,1,1)x(0,1,1)\_12", panel.first = grid(col =

"gray", lty = "dotted"))

> points(x = x, pch = 20, col = "blue")



I also constructed another plot of the data using the gridlines to determine the seasonal behavior (looks like s = 12).

> # The xaxt = "n" option prevents the x-axis tick marks

> # from being drawn

> plot(x = x, ylab = expression(x[t]), xlab = "t", type =

"l", col = "red", xaxt = "n", main =

"Data simulated from ARIMA(0,1,1)x(0,1,1)\_12")

> points(x = x, pch = 20, col = "blue")

> #x-axis

> # Major and minor tick marks - see intro. to R examples

> axis(side = 1, at = seq(from = 0, to = 192, by = 12))

> axis(side = 1, at = seq(from = 0, to = 200, by = 2),

tck = 0.01, labels = FALSE)

> abline(v = seq(from = 0, to = 192, by = 12), lty =

"dotted", col = "lightgray")

> #y-axis

> abline(h = seq(from = -350, to = -150, by = 50), lty =

"dotted", col = "lightgray")



Next are the plots of the estimated ACF and PACF. Notice the lag.max command is used to increase the number of lags shown on the plot. This is helpful to see a seasonal pattern. Obviously, the data is nonstationary in the mean.

> #ACF and PACF of x\_t

> par(mfcol = c(1,2))

> acf(x = x, type = "correlation", lag.max = 50, ylim =

c(-1,1), main = expression(paste("Estimated ACF plot

for ", x[t])))

> pacf(x = x, lag.max = 50, ylim = c(-1,1), xlab = "h",

main = expression(paste("Estimated PACF plot for ",

x[t])))





There are two ways to take care of the nonstationarity:

1. Find the first differences: (1-B)
2. Determine if seasonal differences are needed: (1-Bs)

The ACF plot is showing a “slightly higher” value at every 12th autocorrelation. This possibly shows that (1-B12) is needed. In addition, the ACF at all lags is going slowly to 0 indicating that (1-B) is needed. I decided to look at this type of differences first.

> #ACF and PACF of (1-B)\*x\_t

> par(mfrow = c(1,2))

> acf(x = diff(x = x, lag = 1, differences = 1), type =

"correlation", lag.max = 50, xlim = c(1,50), ylim =

c(-1,1), main = expression(paste("Est. ACF for ", (1-

B)\*x[t], " data")))

> pacf(x = diff(x = x, lag = 1, differences = 1), lag.max

= 50, xlim = c(1,50), ylim = c(-1,1), xlab = "h",

main = expression(paste("Est. PACF for ", (1-B)\*x[t],

" data")))



The ACF plot definitely shows a 12 lag pattern of very high autocorrelations. This suggests that (1-B12) should also be examined. Below is the code used to examine the results of (1-B12)(1-B)xt. Notice how the diff()function is used.

> #ACF and PACF of (1-B)(1-B^12)\*x\_t

> par(mfrow = c(1,2))

> x2 <- diff(x = diff(x = x, lag = 12, differences = 1),

lag = 1, differences = 1)

> acf(x = x2, type = "correlation", lag.max = 50, xlim =

c(1,50), ylim = c(-1,1), main = expression(paste(

"Est. ACF for ", (1-B)\*(1-B^12)\*x[t], " data")),

panel.first = grid(col = "gray", lty = "dotted"))

> pacf(x = x2, lag.max = 50, xlim = c(1,50), ylim =

c(-1,1), xlab = "h", main = expression(paste("Est.

PACF for ", (1-B)\*(1-B^12)\*x[t], " data")),

panel.first = grid(col = "gray", lty = "dotted"))



> #Plot of the differenced data

> plot(x = x2, ylab = expression((1-B)\*(1-B^12)\*x[t]),

xlab = "t", type = "l", col = "red", xaxt = "n",

main = expression(paste("Plot of ", (1-B)\*(1-

B^12)\*x[t])))

> points(x = x2, pch = 20, col = "blue")

> axis(side = 1, at = seq(from = 0, to = 192, by = 12))

> axis(side = 1, at = seq(from = 0, to = 200, by = 2),

tck = 0.01, labels = FALSE)

> abline(v = seq(from = 0, to = 192, by = 12), lty =

"dotted", col = "lightgray")

> abline(h = seq(from = -20, to = 10, by = 10), lty =

"dotted", col = "lightgray")



There is no longer a nonstationary behavior in the resulting time series data. The PACF plot shows definite significant values at lag 1, 12, and 13. A few other partial autocorrelations may be significant also. The ACF has significant autocorrelations at lag 1 and 12. This indicates that an ARIMA(0,1,1)×(0,1,1)12 should be examined. Also, this assumes the PACF behavior is tailing off to 0 for lag 12, 24, … and lag 1, 2, … . Note that there may be justification for other models with P = 1 or 2 and p = 1. I will examine an ARIMA(0,1,0)×(0,1,1)12 model and determine from the residuals if any changes need to be made. The q = 1 is not used because I would like to get the seasonal part of the model figured out first. Below is the code and output. Notice the syntax used with the arima()function to fit the SARIMA model.

> #Fit model to data

> mod.fit.010.011 <- arima(x = x, order = c(0, 1, 0),

seasonal = list(order = c(0, 1, 1), period = 12))

> mod.fit.010.011

Call:

arima(x = x, order = c(0, 1, 0), seasonal = list(order = c(0, 1, 1), period = 12))

Coefficients:

sma1

-0.6447

s.e. 0.0605

sigma^2 estimated as 32.46: log likelihood = -593.95, aic = 1191.89

> #Need to run function in examine.mod.R file first

> examine.mod(mod.fit.obj = mod.fit.010.011,

mod.name = "ARIMA(0,1,1)x(0,1,1)\_12", max.lag = 50)

$z

sma1

-10.65188

$p.value

sma1

0





The residual ACF and PACF show no more signs of a seasonal pattern! It is too bad that R does not allow one to automatically use a larger number of lags on the ACF plot (see the R help for how the number of lags displayed is determined). The Θ1 is significantly different from 0. The ACF has a significant autocorrelation at lag 1 and the PACF appears to be tailing off to 0. Therefore, the model ARIMA(0,1,1)×(0,1,1)12 will be tried. Below is the code and output.

> #ARIMA(0,1,1)x(0,1,1)\_12

> mod.fit.011.011 <- arima(x = x, order = c(0, 1, 1),

seasonal = list(order = c(0, 1, 1), period = 12))

> mod.fit.011.011

Call:

arima(x = x, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))

Coefficients:

ma1 sma1

-0.4113 -0.6347

s.e. 0.0680 0.0617

sigma^2 estimated as 28.19: log likelihood = -580.72, aic = 1167.44

> examine.mod(mod.fit.obj = mod.fit.011.011, mod.name =

"ARIMA(0,1,1)x(0,1,1)\_12", max.lag = 50)

$z

ma1 sma1

-6.051258 -10.285052

$p.value

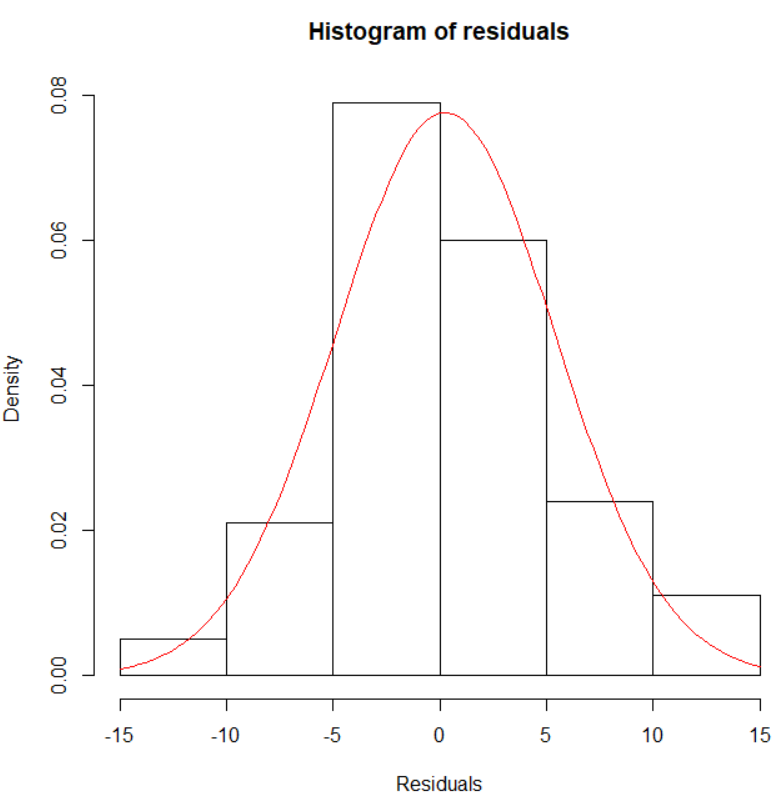
ma1 sma1

1.437193e-09 0.000000e+00





The residual ACF and PACF appear to be similar to the same plots resulting from a white noise process. The Θ1 and θ1 are significantly different from 0. There are no extreme standardized residuals. The residual histogram and Q-Q plot maybe show some non-normality on the tails of the distribution, but it is not too bad. Remember the data was simulated using a normal distribution so this is a good example of something to not be concerned about! The histogram may also include too many classes for the x-axis. Below is a plot of the same residuals now with less classes (see program for code).



The Ljung-Box-Pierce test indicates there is not sufficient evidence of non-zero autocorrelations.

Therefore, my final estimated model is

(1-B12)(1-B)xt = (1-0.6347B12)(1-0.4113B)wt

where  = 28.19

Compare how close these parameter estimates are to what was used to generate the data!

If I did not know what the model should have been at the beginning, it would have been very reasonable to investigate other models indicated previously. If more than one model passed all of the diagnostic tests, I would have used the AIC to help pick the best model.

Forecasts are found for 24 time periods into the future. Notice this corresponds to just 2 “seasons” into the future.

> #Forecasts 24 time periods into the future

> fore.mod <- predict(object = mod.fit.011.011, n.ahead =

24, se.fit = TRUE)

> fore.mod

$pred

Time Series:

Start = 201

End = 224

Frequency = 1

[1] -317.5528 -330.8742 -352.8390 -344.9029 -332.8707

[6] -348.5246 -348.1134 -352.1134 -351.7119 -350.6458

[11] -352.5155 -346.0474 -328.9166 -342.2380 -364.2027

[16] -356.2667 -344.2345 -359.8884 -359.4772 -363.4771

[21] -363.0757 -362.0096 -363.8793 -357.4111

$se

Time Series:

Start = 201

End = 224

Frequency = 1

[1] 5.309460 6.161149 6.908628 7.582779 8.201703

[6] 8.777090 9.317011 9.827313 10.312394 10.775661

[11] 11.219815 11.647044 12.700809 13.398572 14.061753

[16] 14.695035 15.302131 15.886044 16.449241 16.993784

[21] 17.521412 18.033608 18.531654 19.016660

> pred.mod <- x - mod.fit.011.011$residuals

> #Calculate 95% C.I.s

> low <- fore.mod$pred - qnorm(p = 0.975, mean = 0, sd = 1)

\* fore.mod$se

> up <- fore.mod$pred + qnorm(p = 0.975, mean = 0, sd = 1)

\* fore.mod$se

> data.frame(low, up)

low up

1 -327.9592 -307.1465

2 -342.9499 -318.7986

3 -366.3796 -339.2983

EDITED

23 -400.2007 -327.5580

24 -394.6831 -320.1392

> #When I originally simulated the data, 224 observations

> # were actually simulated. I did this so that I could   
> # examine how well the confidence intervals captured

> # the true future values.

> x.extra <- sarima.data$x[201:224]

> plot(x = x, ylab = expression(x[t]), xlab = "t", type =

"o", col = "red", xaxt = "n", ylim = c(-400, -150),

pch = 20, xlim = c(0, 224), main = "Forecast plot")

> axis(side = 1, at = seq(from = 0, to = 224, by = 12),

las = 2) #las makes the labels vertical

> axis(side = 1, at = seq(from = 0, to = 224, by = 2),

tck = 0.01, labels = FALSE)

> abline(v = seq(from = 0, to = 224, by = 12), lty =

"dotted", col = "lightgray")

> abline(h = seq(from = -400, to = -150, by = 50), lty =

"dotted", col = "lightgray")

> #Predicted values for t = 1, ..., 200 and forecasts for

> # t = 201, ..., 224

> lines(x = c(pred.mod, fore.mod$pred), lwd = 1, col =

"black", type = "o", pch = 17)

> #Forecasts and C.I.s

> lines(y = low, x = 201:224, lwd = 1, col = "darkgreen",

lty = "dashed")

> lines(y = up, x = 201:224, lwd = 1, col = "darkgreen",

lty = "dashed")

> #Actual future values

> lines(y = x.extra, x = 201:224, lwd = 1, col = "blue",

lty = "solid", type = "o", pch = 20)

> #Legend

> legend(locator(1), legend = c("Observed", "Forecast",

"95% C.I.", "Actual future values"), lty = c("solid",

"solid", "dashed", "solid"), col = c("red", "black",

"darkgreen", "blue"), pch = c(20, 17, NA, 20), bty =

"n")



This is hard to see so I created another plot that zoomed in on the right side. Please see the program for the code.



Notice that all of the actual values are within the 95% C.I.s!

What if δ was estimated? The term ends up being highly non-significant as shown in the output below.

> mod.fit.011.011.delta <- arima(x = x, order = c(0, 1, 1),

seasonal = list(order = c(0, 1, 1), period = 12),

xreg = 1:length(x))

> mod.fit.011.011.delta

Call:

arima(x = x, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),

xreg = 1:length(x))

Coefficients:

ma1 sma1 1:length(x)

-0.4113 -0.6347 -2.1554

s.e. 0.0680 0.0617 5365.1135

sigma^2 estimated as 28.19: log likelihood = -580.72, aic = 1169.44

> examine.mod(mod.fit.obj = mod.fit.011.011.delta,

mod.name = "ARIMA(0,1,1)x(0,1,1)\_12 with delta",

max.lag = 50)

$z

ma1 sma1 1:length(x)

-6.051177052 -10.283394781 -0.000401746

$p.value

ma1 sma1 1:length(x)

1.437913e-09 0.000000e+00 9.996795e-01

The diagnostic plots are excluded from the notes.