MATH545\_Assignment3\_260677676\_submitted

Dan Yunheum Seol

2018-11-20

# 3.1

Among the time series process of the form

where

and

Identify which are causal ot invertible ##(a)

is solved by (using quadratic formula)

Thus it is causal.

which is a constant polynomial, so it does not have a root. So our condition for invertibility holds trivially. So it is invertible.

## (b)

is solved by (using quadratic formula)

Because , it is not causal.

has roots (by quadratic formula: I will omit the steps from now)

where for both case, so it is invertible.

## (c)

has a solution , thus it is causal.

has a solution at , thus it is not invertible

## d)

has unique solution of multiplicity 2, Thus it is causal.

so it is invertible as well. ##e)

has a solution -Hence it is not causal.

so it has a solution , which makes it invertible.

# 3.9

#Importing possibly necessary packages  
library(itsmr)  
library(tidyverse)

## ── Attaching packages ───────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 2.2.1 ✔ purrr 0.2.4  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.4  
## ✔ tidyr 0.8.0 ✔ stringr 1.3.0  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ──────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(forecast)

##   
## Attaching package: 'forecast'

## The following object is masked from 'package:itsmr':  
##   
## forecast

head(deaths)

## [1] 9007 8106 8928 9137 10017 10826

class(deaths)

## [1] "numeric"

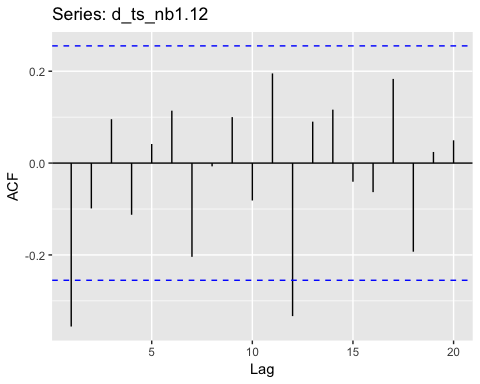
Now we will convert the data into a time series object:

d\_ts <- ts(deaths, frequency = 12, start = c(1973, 1))  
#head(d\_ts)  
d\_ts\_nabla12 <- diff(d\_ts, lag=12)  
d\_ts\_nb1.12 <- d\_ts\_nabla12 %>% diff(., lag=1)  
  
head(d\_ts\_nb1.12)

## Feb Mar Apr May Jun Jul  
## 1974 132 235 175 -588 -11 117

Now we will be finding sample mean, acvf, acf

##sample mean  
d\_bar <- mean(d\_ts\_nb1.12)  
  
##sample acvf  
acf <- ggAcf(d\_ts\_nb1.12, lag.max = 20)  
acf



acf\_values <-acf[["data"]]  
head(acf\_values)

## Lag ACF  
## 2 1 -0.35584371  
## 3 2 -0.09872089  
## 4 3 0.09553025  
## 5 4 -0.11251546  
## 6 5 0.04152922  
## 7 6 0.11410853

print("Correlation values are below:")

## [1] "Correlation values are below:"

acf\_values

## Lag ACF  
## 2 1 -0.355843707  
## 3 2 -0.098720893  
## 4 3 0.095530249  
## 5 4 -0.112515462  
## 6 5 0.041529223  
## 7 6 0.114108535  
## 8 7 -0.204130053  
## 9 8 -0.007123311  
## 10 9 0.100066891  
## 11 10 -0.081448224  
## 12 11 0.195205609  
## 13 12 -0.333182813  
## 14 13 0.090181278  
## 15 14 0.116314298  
## 16 15 -0.040607730  
## 17 16 -0.063250388  
## 18 17 0.183280640  
## 19 18 -0.192938261  
## 20 19 0.024188047  
## 21 20 0.049580453

n <- sum(d\_ts\_nb1.12) / d\_bar #to get the n; dim(ts) did not work  
  
#biased method-of-moments estimator of gamma(0)  
  
gamma.zeros <- c((n-1)/n \* var(d\_ts\_nb1.12), var(d\_ts\_nb1.12))  
  
print("ACVF values (with different adjustments) are below:")

## [1] "ACVF values (with different adjustments) are below:"

acvf\_values1 <- gamma.zeros[1]\*acf\_values #clearly biased  
acvf\_values2 <-gamma.zeros[2]\*acf\_values #less biased  
acvf\_values1

## Lag ACF  
## 2 152669.6 -54326.528  
## 3 305339.3 -15071.682  
## 4 458008.9 14584.568  
## 5 610678.5 -17177.694  
## 6 763348.2 6340.251  
## 7 916017.8 17420.908  
## 8 1068687.4 -31164.460  
## 9 1221357.1 -1087.513  
## 10 1374026.7 15277.175  
## 11 1526696.3 -12434.670  
## 12 1679366.0 29801.969  
## 13 1832035.6 -50866.898  
## 14 1984705.2 13767.943  
## 15 2137374.9 17757.661  
## 16 2290044.5 -6199.567  
## 17 2442714.1 -9656.413  
## 18 2595383.7 27981.388  
## 19 2748053.4 -29455.813  
## 20 2900723.0 3692.780  
## 21 3053392.6 7569.430

acvf\_values2 ## this values will be used in our function

## Lag ACF  
## 2 155301.9 -55263.192  
## 3 310603.7 -15331.539  
## 4 465905.6 14836.026  
## 5 621207.5 -17473.861  
## 6 776509.3 6449.566  
## 7 931811.2 17721.269  
## 8 1087113.1 -31701.778  
## 9 1242414.9 -1106.263  
## 10 1397716.8 15540.575  
## 11 1553018.7 -12649.061  
## 12 1708320.5 30315.796  
## 13 1863622.4 -51743.913  
## 14 2018924.3 14005.321  
## 15 2174226.1 18063.828  
## 16 2329528.0 -6306.456  
## 17 2484829.9 -9822.903  
## 18 2640131.7 28463.826  
## 19 2795433.6 -29963.672  
## 20 2950735.5 3756.449  
## 21 3106037.3 7699.937

auto.arima(d\_ts\_nb1.12)

## Series: d\_ts\_nb1.12   
## ARIMA(0,0,1)(0,0,1)[12] with non-zero mean   
##   
## Coefficients:  
## ma1 sma1 mean  
## -0.4963 -0.6146 21.0220  
## s.e. 0.1351 0.1932 11.9921  
##   
## sigma^2 estimated as 98283: log likelihood=-424.27  
## AIC=856.53 AICc=857.27 BIC=864.84

So we can get the expression for the time series

which is

where

# 5.3

# 5.4

R2 <- c(1, 0.427, 0.427, 1)  
vec <- c(0.427, 0.475)  
R2. <- matrix(R2, nrow=2, ncol=2)  
#R2.  
cor. <- matrix(vec, nrow=2, ncol=1)  
R2.inv <- solve(R2.)  
phi.hat <- R2.inv %\*% cor.  
phi.hat

## [,1]  
## [1,] 0.2741628  
## [2,] 0.3579325

############### calculating the sigma2hat#############  
gamma0 <- 1.15  
  
gamma<- c(0.427\*1.15, 0.475\*1.15)  
gamma. <- matrix(gamma, nrow=2, ncol=1)  
phiTgamma. <- t(phi.hat) %\*% gamma.  
phiTgamma.

## [,1]  
## [1,] 0.3301483

phiTgamma.[1]

## [1] 0.3301483

phiTgamma <- phiTgamma.[1]  
sigma2hat <- gamma0-phiTgamma  
sigma2hat

## [1] 0.8198517

##############CI(mean)#####################  
v. <- 1.15\*(1+2\*(0.427+0.475+0.169))  
  
CI95 <- function(center, v, n){  
 print(c(center-1.96\*(sqrt(v)/sqrt(n)), center+1.96\*(sqrt(v)/sqrt(n))))  
}  
CI95(3.82, v., 200 )

## [1] 3.556553 4.083447

################CI(phi)######################  
Gamma.hat <- c(1.15, 1.15\*0.427, 1.15\*0.427, 1.15)  
Gamma.hat. <- matrix(Gamma.hat, nrow=2, ncol=2)  
Gamma.hat

## [1] 1.15000 0.49105 0.49105 1.15000

V.hat <- sigma2hat\*solve(Gamma.hat.)  
V.hat

## [,1] [,2]  
## [1,] 0.8718843 -0.3722946  
## [2,] -0.3722946 0.8718843

v1 <- V.hat[1,1]  
v2 <- V.hat[2,2]  
CI95(phi.hat[1][1], v1, 200)

## [1] 0.1447520 0.4035736

CI95(phi.hat[2,1], v2, 200)

## [1] 0.2285217 0.4873433

c(0.274-0.139)

## [1] 0.135

##################PACF######################