Homework 4
UMN STAT 5511
Charles R. Doss
Solution

The usual formatting rules:

- Your homework (HW) should be formatted to be easily readable by the grader.
- You may use knitr or Sweave in general to produce the code portions of the HW. However, the output from knitr/Sweave that you include should *be only what is necessary to answer the question*, rather than just any automatic output that R produces. (You may thus need to avoid using default R functions if they output too much unnecessary material, and/or should make use of `invisible()` or `capture.output()`.)
  - For example: for output from regression, the main things we would want to see are the estimates for each coefficient (with appropriate labels of course) together with the computed OLS/linear regression standard errors and p-values. If other output is not needed to answer the question, it should be suppressed!
- Code snippets that directly answer the questions can be included in your main homework document; ideally these should be preceded by comments or text at least explaining what question they are answering. Extra code can be placed in an appendix.
- All plots produced in R should have appropriate labels on the axes as well as titles. Any plot should have explanation of what is being plotted given clearly in the accompanying text.
- Plots and figures should be *appropriately sized*, meaning they should not be too large, so that the page length is not too long. (The arguments fig.height and fig.width to knitr chunks can achieve this.)
- **Directions for "by-hand" problems:** In general, credit is given for (correct) shown work, not for final answers; so show **all** work for each problem and explain your answer fully.

For some questions on this homework you will need to do parameter estimation from a series, which we have not yet discussed at the time of the assignment of this homework (but we will discuss this shortly). You may use the arima() function for estimation. You do not need to worry about the 'method' argument to arima() (any choice will be OK).

1. Find (on Canvas) the file "hw4dat.rsav" (which can be loaded into R using load("hw4dat.rsav")). It contains a time series ("xx"). The series is a "demeaned" monthly revenue stream (in millions of dollars) for a company. There are $n = 96$ observations.

   The series has been "demeaned"; usually that would mean we subtract off $\bar{X}$ from every data point, but pretend for now we know the mean μ exactly so we have subtracted off μ from every data point, so the new series is exactly (theoretically) mean 0. (But thus its sample mean is not precisely 0.)

   We will consider possible ARMA models for the series X_t. We assume that the corresponding white noise is Gaussian (so X_t is Gaussian).

   We will consider first an AR(2) model. We assume we know the true model exactly: it is

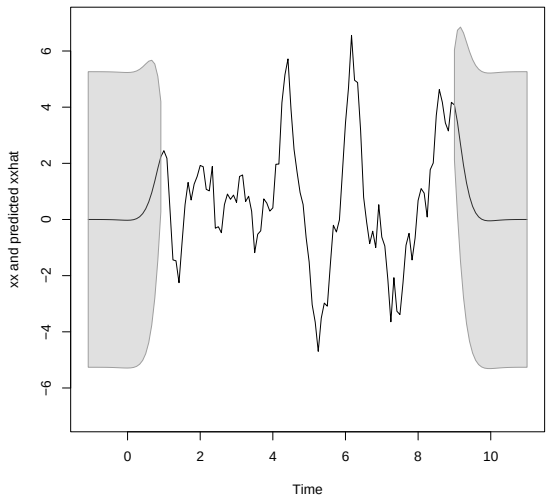
   $$\text{Model 1:} \qquad X_t = 1.34X_{t-1} - .48X_{t-2} + W_t, \quad W_t \overset{\text{iid}}{\sim} N(0,1).$$

   (a) Compute forecasts and backcasts using Model 1, up to 25 time steps in the future and into the past. Write code to do the prediction by hand (i.e., I just/only mean that you should not use the predict() function). Plot the data, forecast, backcast, and 95% prediction intervals based on assuming gaussianity (all on one plot). (Note: you do not need to do a multiplicity correction for the prediction intervals.)

   Solution:

```
load("HW4dat.rsav")
revxx=rev(xx)
phi1=1.34
phi2=-0.48
nn=length(xx)
pred.forward=rep(0,25)
pred.backward=rep(0,25)
```

```
pred.ar2=function(xt1,xt2,coef1,coef2){
  return(coef1*xt1+coef2*xt2)
}
pred.forward[1]=pred.ar2(xx[96],xx[95],phi1,phi2)
pred.forward[2]=pred.ar2(pred.forward[1],xx[96],phi1,phi2)
pred.backward[1]=pred.ar2(revxx[96],revxx[95],phi1,phi2)
pred.backward[2]=pred.ar2(pred.backward[1],revxx[96],phi1,phi2)
for(i in 3:25){
  pred.forward[i]=pred.ar2(pred.forward[i-1],pred.forward[i-2],phi1,phi2)
  pred.backward[i]=pred.ar2(pred.backward[i-1],pred.backward[i-2],phi1,phi2)
}
##calculate the psi_j's (j=0,1,2,...,24)
psi=c(1,phi1,rep(0,23))
for(i in 3:25){
  psi[i]=pred.ar2(psi[i-1],psi[i-2],phi1,phi2)
}
pred.se=sqrt(cumsum(psi^2))
##create the a new time series including the predicted values
xx.new=ts(c(rev(pred.backward),xx,pred.forward),
          frequency=12,start=c(-2,12),end=c(11,1))
ts.plot(xx.new,ylab="xx and predicted xxhat",ylim=c(-7,7))
UU.forward <- pred.forward + 1.96*pred.se
LL.forward <- pred.forward - 1.96*pred.se
time.forward <- time(xx.new)[(length(xx.new)-24):length(xx.new)]
UU.backward <- pred.backward + 1.96*pred.se
LL.backward <- pred.backward - 1.96*pred.se
time.backward <- time(xx.new)[1:25]
AA.forward <- c(time.forward, rev(time.forward))
BB.forward <- c(LL.forward,rev(UU.forward))
polygon(AA.forward,BB.forward, border=8, col=gray(.6, .3))
AA.backward <- c(rev(time.backward), time.backward)
BB.backward <- c(LL.backward,rev(UU.backward))
polygon(AA.backward,BB.backward, border=8, col=gray(.6, .3))
```



(b) Give a constant (nonrandom) number that the 100-step-ahead forecast, X_{196}^{100}, will be approxi-

mately equal to.

Solution:
Zero; the mean is 0 and for forecasts far in the future we will just be guessing the mean.

(c) If you were to do the one-step-ahead prediction but based on no data, what would your prediction be? (Based on no data, it is the same for predicting at time 97 or at any other time.) What would the mean-squared prediction error (call it E) be? Compare P_{97}^{96} to E.

Solution:
The prediction would be the mean, 0. $E = \gamma(0) = \frac{1-\phi_2}{1+\phi_2}\frac{\sigma_w^2}{(1-\phi_2)^2-\phi_1^2} = 7.209103$ and $P_{97}^{96} = \sigma^2\psi_0^2 = 1 \times 1 = 1$ (you can check the equation (3.86) in the textbook on page 108). We see that E is much larger than P_{97}^{96}, which makes sense, since E is the mean-squared prediction error based on no data.

(d) Now say that we know the true mean of the company's revenue series is .3 (million dollars). Provide

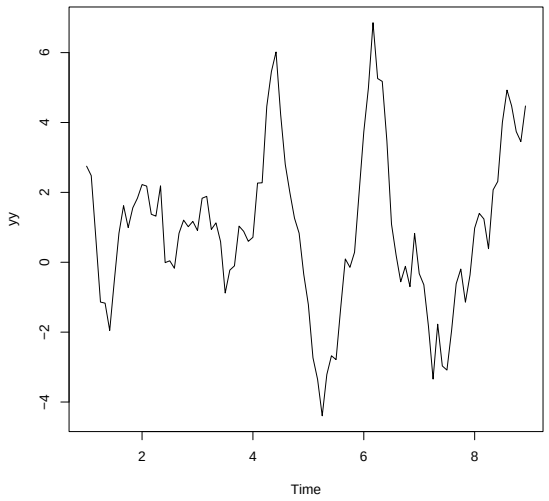
i. a plot of the company's (not-demeaned) revenue series (let's call it Y_t),

ii. and the prediction equation for Y_{n+1}^n (I do not mean for you to write anything having to do with setting any expectations to 0 here; I simply mean for you to write the formula for predicting Y at time $n+1$ based on the model (including the mean), and past data).

Solution:

i. $Y_t = X_t + .3$.

```
mu=0.3
yy=xx+mu
plot(yy)
```



ii. Intercept for Y_t is given by $\mu(1-\phi_1-\phi_2) = 0.042$. Thus $Y_{n+1}^n = 0.042 + 1.34Y_n - 0.48Y_{n-1}$.

(e) The series Y_t is a monthly revenue stream for a company. The company needs to decide, before the current month is up (i.e., before seeing the one-month-ahead revenue, Y_{97}), whether to make an important investment in equipment which will cost 1.1 million dollars. If their revenue next month cannot cover the cost (is less than the cost of the investment) they will go bankrupt (they have exactly 0 cash on hand and cannot take out loans). Explain why they should or should not make the investment.

Solution:

For Y_t, we have $Y_{97}^{96} = 0.042 + 1.34Y_{96} - 0.48Y_{95}$ and $P_{97}^{96} = \sigma^2\psi_0^2 = 1 \times 1 = 1$. We calculate the probability that Y_{97} is at least 1.1

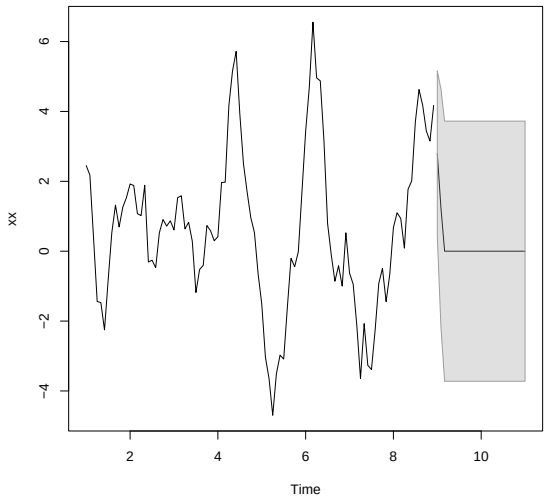
```
1-pnorm(1.1,0.042+1.34*yy[96]-0.48*yy[95],1)

## [1] 0.999478
```

Therefore, there is a 99.95 percent chance that the company has enough money, based on P_{97}^{96} and Gaussianity. So they are safe to make the investment.

(f) The second model we will consider (Model 2) is an MA(2) model. Estimate an MA(2) model on the X_t (demeaned) series using the arima() function (there is an include.mean variable; set it to false). Then plot (on one plot): the data, forecasts up to 25 months ahead, and 95% prediction intervals [assuming gaussianity]. You may use the predict() function.

Solution:

```
fit_ma <- arima(x=xx, order=c(0,0,2), include.mean=F)
preds <- predict(fit_ma, n.ahead=25)
ts.plot(xx, preds$pred,ylab="xx")
UU <- preds$pred + 1.96*preds$se
LL <- preds$pred - 1.96*preds$se
AA <- c(time(UU), rev(time(UU)))
BB <- c(LL,rev(UU))
polygon(AA,BB, border=8, col=gray(.6, .3))
```



(g) Compare the forecasts for Model 1 and Model 2: specifically, discuss how quickly the two forecasts revert to the long run average of the series. Provide an explanation of this. [Note: I am not intending for you to discuss the fact that in one case the coefficients were estimated and in the other they were not estimated.]

Solution:

The forecast for an MA(2) reverts to the mean after just 2 time periods (i.e. on the 3rd month). This is because there is no correlation between X_{96} and X_{99} (or $X_{100}, \ldots$). On the other hand, the AR(2) forecast technically never is exactly equal to the long term average (0) but in this case after around 15 months or so it is pretty close. The reason the forecast is never exactly

0 is because all future time points technically are correlated with the observed data. But that correlation diminishes over time and so eventually the forecast gets close to 0.

(h) Now we consider changing the frequency of observation. Imagine someone outside the company observes the company revenue but only on a quarterly basis, by which I mean every 3 months (thus they observe March's revenue, then June's revenue, ...). Let this series of de-meaned observations be Z_t. If the true model for X_t is AR(1), $X_t = \phi_1 X_{t-1} + W_t$ where $W_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$, then what is the (true) model for Z_t (including the distribution of the white noise series)?

Solution:

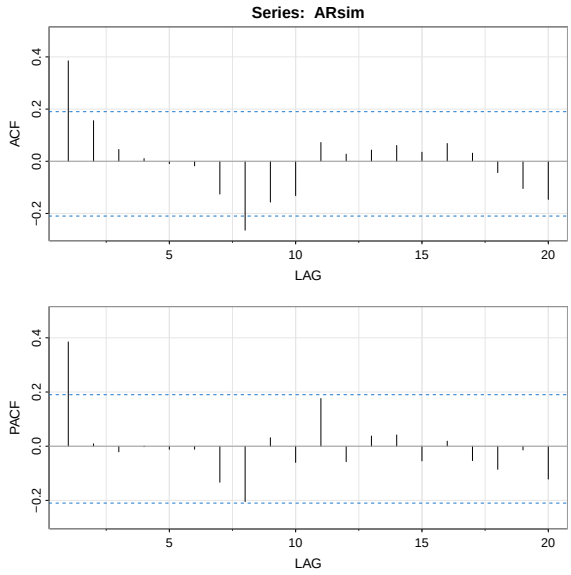
$$\begin{aligned} X_t &= \phi X_{t-1} + W_t \\ &= \phi(\phi X_{t-2} + W_{t-1}) + W_t \\ &= \phi^2(\phi X_{t-3} + W_{t-2}) + \phi W_{t-1} + W_t \\ &= \phi^3 X_{t-3} + \phi^2 W_{t-2} + \phi W_{t-1} + W_t \end{aligned} \tag{1}$$

Therefore, $Z_t = \phi^3 Z_{t-1} + R_t$ where $R_t \overset{\text{iid}}{\sim} N(0, 1 + \phi^2 + \phi^4)\sigma^2)$.

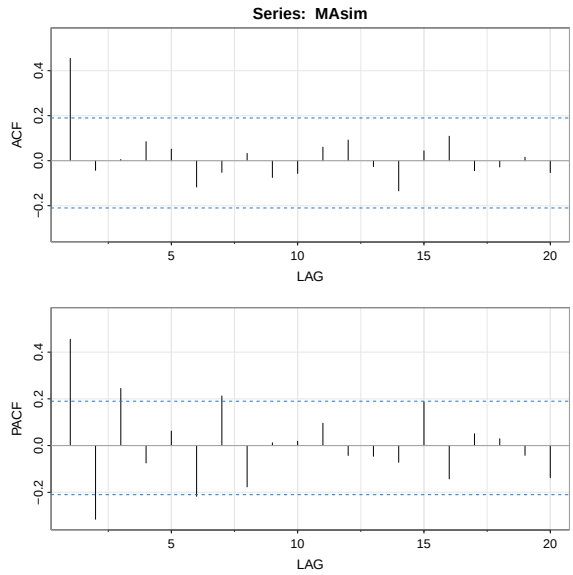
2. Shumway and Stoffer (4th ed.), question 3.9

Solution:

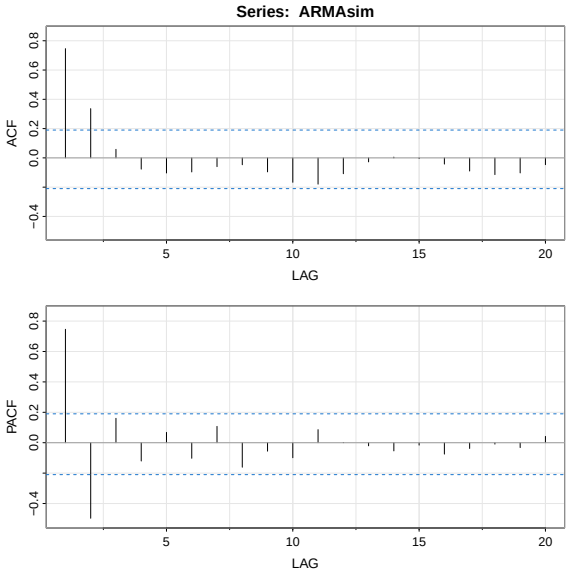
```
library(astsa)
nn <- 100
phi <- .6
theta <- .9
ARsim <- arima.sim(model=list(order=c(1,0,0), ar=phi), n=nn)
MAsim <- arima.sim(model=list(order=c(0,0,1), ma=theta), n=nn)
ARMAsim <- arima.sim(model=list(order=c(1,0,1), ar=phi, ma=theta), n=nn)
invisible(acf2(ARsim))
```




```
invisible(acf2(MAsim))
```




```
invisible(acf2(ARMAsim))
```



The behaviour of sample ACF/PACF are similiar to therotical ACF/PACF.

For AR(1), we can see that the PACF drops to 0 after lag 1 and the ACF slowly decays. So it correctly identify AR(1) here.

The reverse happens for the MA(1): the ACF drops after lag 1 and the PACF slowly decay.

For ARMA(1,1), both ACF and PACF tail off and neither of them identify the orders of p and q.

3. Shumway and Stoffer (4th ed.), question 3.15

Solution:

3.15

For an AR model the (1 to m)-step ahead estimator is equal to the infinite past predictor and so

equals

$$\begin{aligned} x^t_{t+1} &= E(x_{t+1}|x_t, \ldots, x_1, \ldots) = \phi x_t \\ x^t_{t+2} &= E(x_{t+2}|x_t, \ldots, x_1, \ldots) = \phi^2 x_t \\ &\vdots \\ x^t_{t+m} &= E(x_{t+m}|x_t, \ldots, x_1, \ldots) = \phi^m x_t. \end{aligned}$$

The prediction error for each step estimator are

$$\begin{aligned} e(1) &= x_{t+1} - x^t_{t+1} = x_{t+1} - \phi x_t = w_t \\ e(2) &= x_{t+2} - x^t_{t+2} = x_{t+2} - \phi^2 x_t = (\phi x_{t+1} + w_{t+2}) - \phi^2 x_t = \phi(x_{t+1} - \phi x_t) + w_{t+2} = \phi w_{t+1} + w_{t+2} \\ &\vdots \\ e(m) &= \phi^{m-1} w_{t+1} + \phi^{m-2} w_{t+2} + \cdots + w_{t+m} \end{aligned}$$

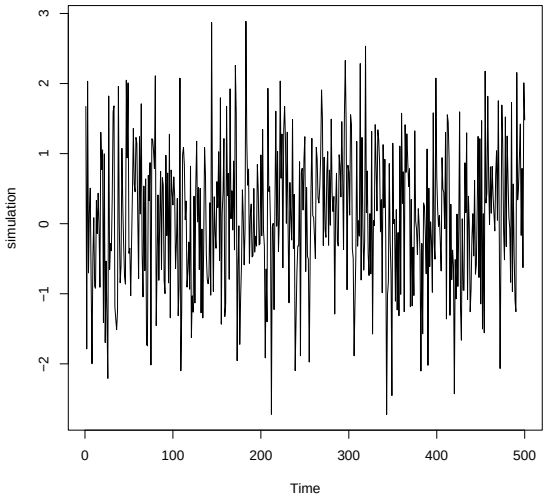
Hence

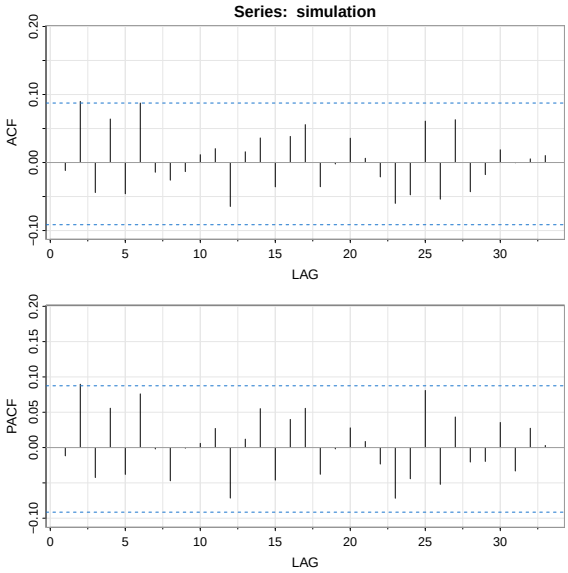
$$E(x_{t+m} - x^t_{t+m})^2 = Var(e(m)) = \sigma_w^2(1 + \phi^2 + \cdots + \phi^{2m-4} + \phi^{2m-2}) = \sigma_w^2 \frac{1 - \phi^{2n}}{1 - \phi^2}$$

4. Shumway and Stoffer (4th ed.), question 3.20

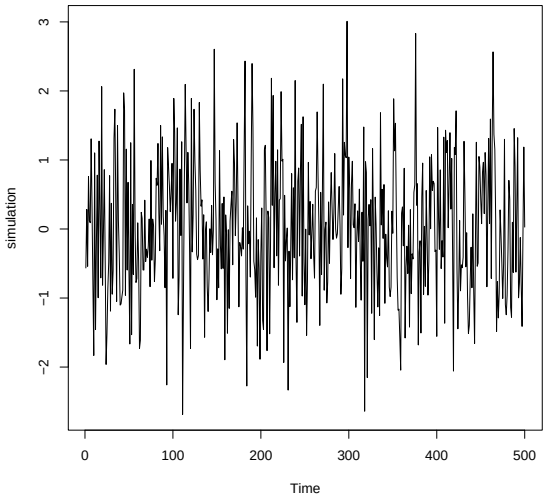
Solution:

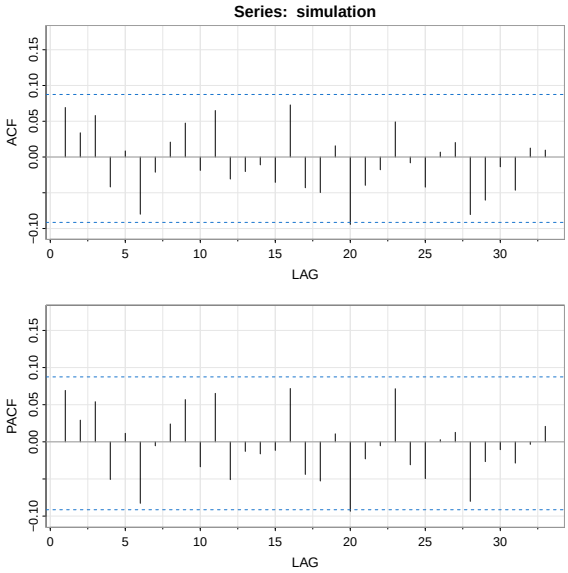
```
library(astsa)
set.seed(2)
nn <- 500
for (ii in 1:3){
    simulation <- arima.sim(model=list(ar=.9, ma=-.9), n=nn)
    plot(simulation)
    acf2(simulation)
    ## Question didn't specify whether to include the mean or not;
    ## The output is similar regardless.
    xxfit <- arima(simulation, order=c(1,0,1), include.mean=F)
    print(paste("Iteration", ii,
                "output (estimate in first row and standard error in second row)"))
    print(xxfit$coef)
    print(sqrt(diag(xxfit$var.coef))) ## SE's
}
```



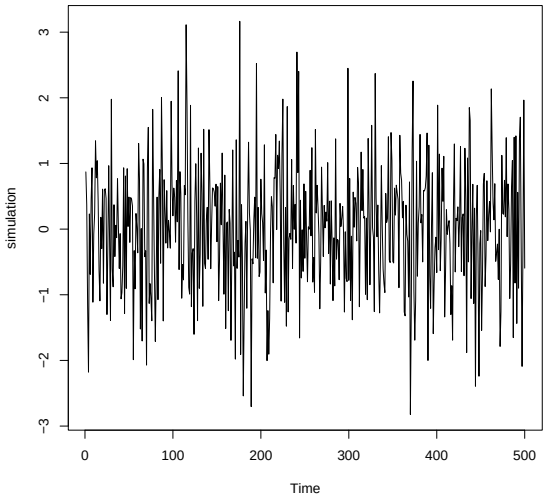


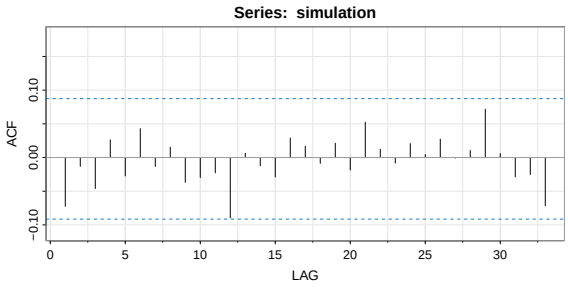

```
## [1] "Iteration 1 output (estimate in first row and standard error in second row)"
##        ar1        ma1
## -0.8530084  0.8007494
##        ar1        ma1
## 0.09473743 0.10619018
```

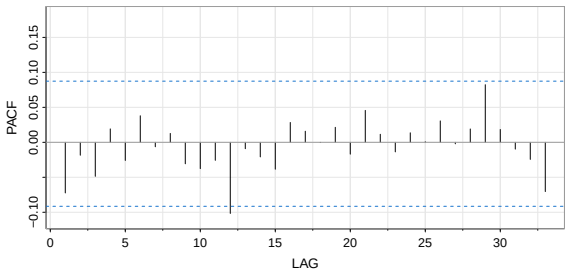





```
## [1] "Iteration 2 output (estimate in first row and standard error in second row)"
##        ar1        ma1
##  0.4739056 -0.4042076
##       ar1       ma1
## 0.3247298 0.3362740
```








```
## [1] "Iteration 3 output (estimate in first row and standard error in second row)"
##        ar1        ma1
##  0.3941494 -0.4669200
##       ar1       ma1
## 0.4733833 0.4557384
```

We have parameter redundancy here: $\phi(z)$ and $\theta(z)$ share a root of $1/.9$. Thus the stationary series that the model $X_t = .9X_{t-1} + W_t - .9W_{t-1}$ specifies is white noise. The parameter estimates yield an approximately redundant parameterization of an ARMA. The estimate for three simulations are totally different. The standard errors are large (but perhaps not large enough to directly conclude that the parameter values are 0, since the parameter values do not have to be 0, because of the redundant model specification). Also, the sample ACF and PACF do not support the ARMA(1,1) model, however, it shows that the model looks like white noise.

5. A data analyst is analyzing a time series with 1000 observations. She fits an ARMA(2,1) model (mean 0 i.e., with no intercept), yielding the following estimate output:

```
Coefficients:
```

```
         ar1      ar2      ma1
      -0.062    0.817    0.971
```

To check the robustness of the fit, the analyst removes the last 100 observations (leaving 900) and fits the ARMA(2,1) model again. The analyst is surprised to see the following very different estimates output:

```
Coefficients:
         ar1      ar2      ma1
       0.752    0.112    0.100
```

(a) Provide an explanation for why these different results were output.

Solution:

The model is not truly ARMA(2,1), rather, there is parameter redundancy. The actual model is ARMA(1,0), i.e. AR(1). We will verify below that the parameters do appear somewhat redundant.

(b) Provide a diagnostic tool or mechanism or method to assess the answer you gave in the previous part, and explain how you would use it or what you would look for.

Solution:

Check the PACF. If it has just one spike at lag 1, then this is good evidence that the AR(1) model is reasonable.