

Homework 6

Benjamin Noland

1. The following code generates 100 observations from the MA(1) model $Y_t = e_t + 0.7e_{t-1}$:

```
set.seed(1000)
y <- arima.sim(model = list(ma = c(0.7)), n = 100)
```

Let Y_1, \dots, Y_{100} denote the observed time series, and let Σ denote the covariance matrix of $Y = (Y_1, \dots, Y_{100})$.

- a. The covariance matrix Σ is given by $\Sigma_{ij} = (\gamma_{|i-j|})$. Explicitly,

$$\Sigma_{ij} = \begin{cases} (1 + \theta_1^2 + \theta_2^2)\sigma_e^2 & \text{if } i = j \\ (-\theta_1 + \theta_1\theta_2)\sigma_e^2 & \text{if } |i - j| = 1 \\ -\theta_2\sigma_e^2 & \text{if } |i - j| = 2 \\ 0 & \text{otherwise} \end{cases}.$$

- b. Assume the elements of the white noise process $\{e_t\}$ are drawn independently from a $N(0, \sigma_e^2)$ distribution (this is the case for the simulated data generated above). Then the sample $Y = (Y_1, \dots, Y_{100})$ has a mean zero multivariate normal distribution with covariance matrix Σ . Let $y = (y_1, \dots, y_{100})$ denote a realization of the sample Y . Then the likelihood function is given by

$$L(\theta_1, \theta_2 | y) = \frac{1}{\sqrt{(2\pi)^{100} |\Sigma|}} \exp\left(-\frac{1}{2} y^T \Sigma^{-1} y\right).$$

Maximizing $L(\theta_1, \theta_2 | y)$ (or equivalently, the log-likelihood $\log L(\theta_1, \theta_2 | y)$) with respect to (θ_1, θ_2) yields a maximum likelihood estimate $(\hat{\theta}_1, \hat{\theta}_2)$ of (θ_1, θ_2) . Note that, as computed in part (a), the covariance matrix Σ depends on θ_1 and θ_2 .

- c. The following code fits an MA(2) model to the data using maximum likelihood:

```
arima(y, order = c(0, 0, 2), method = "ML")

##
## Call:
## arima(x = y, order = c(0, 0, 2), method = "ML")
##
## Coefficients:
##          ma1      ma2  intercept
##          0.525  -0.184          0.011
## s.e.    0.107    0.110          0.134
##
## sigma^2 estimated as 0.995:  log likelihood = -142,  aic = 290
```

From the output, we see that the maximum likelihood estimates for θ_1 and θ_2 are $\hat{\theta}_1 \approx -0.525$ and $\hat{\theta}_2 \approx 0.184$, respectively (taking into account the differences in convention between R and the book with regard to the signs of the parameters).

- d. From the output in part (c), we see that $\hat{\theta}_1$ and $\hat{\theta}_2$ have approximate standard errors of 0.107 and 0.110, respectively.
2. (Cryer & Chan, Exercise 7.1) We can compute method of moments estimates $\hat{\phi}_1$ and $\hat{\phi}_2$ of ϕ_1 and ϕ_2 , respectively, by solving for $\hat{\phi}_1$ and $\hat{\phi}_2$ in the sample Yule-Walker equations:

$$\begin{aligned} r_1 &= \hat{\phi}_1 + r_1 \hat{\phi}_2 \\ r_2 &= r_1 \hat{\phi}_1 + \hat{\phi}_2 \end{aligned}$$

We get

$$\begin{aligned} \hat{\phi}_1 &= \frac{r_1(1 - r_2)}{1 - r_1^2} \approx 1.11 \\ \hat{\phi}_2 &= \frac{r_2 - r_1^2}{1 - r_1^2} \approx -0.39 \end{aligned}$$

Using these estimates, we can get estimates $\hat{\theta}_0$ and $\hat{\sigma}_e^2$ of θ_0 and σ_e^2 , respectively:

$$\begin{aligned} \hat{\theta}_0 &= \bar{Y}(1 - \hat{\phi}_1 - \hat{\phi}_2) \approx 0.56 \\ \hat{\sigma}_e^2 &= (1 - \hat{\phi}_1 r_1 - \hat{\phi}_2 r_2) s^2 \approx 1.54 \end{aligned}$$

3. (Cryer & Chan, Exercise 7.11) The following code simulates the MA(1) process in question:

```
set.seed(1000)
n <- 48
theta <- -0.6
y <- arima.sim(model = list(ma = c(-theta)), n = n)
```

- a. The following code fits an MA(1) model to the data simulated above using maximum likelihood:

```
arima(y, order = c(0, 0, 1), method = "ML")

##
## Call:
## arima(x = y, order = c(0, 0, 1), method = "ML")
##
## Coefficients:
##          ma1  intercept
##          0.542    -0.228
## s.e.    0.124      0.206
##
## sigma^2 estimated as 0.868:  log likelihood = -64.9,  aic = 134
```

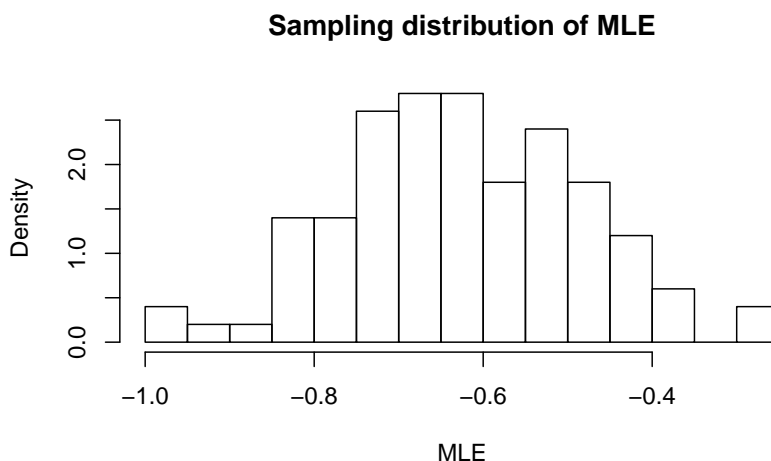
From the output we see that the maximum likelihood estimate of θ is $\hat{\theta} \approx -0.542$.

- b. The following code repeatedly simulates the same series and collects the maximum likelihood estimate of θ on each trial:

```
N <- 100
mle <- numeric(N)
for (i in 1:N) {
  y <- arima.sim(model = list(ma = c(-theta)), n = n)
  fit <- arima(y, order = c(0, 0, 1), method = "ML")
  mle[[i]] <- -fit$coef[["ma1"]]
}
```

- c. The following displays the approximate sampling distribution of the MLE $\hat{\theta}$ based on the simulation in part (b):

```
hist(mle, freq = FALSE, breaks = 20,
     main = "Sampling distribution of MLE", xlab = "MLE")
```



- d. The true parameter value is $\theta = -0.6$. The approximate mean of the MLE sampling distribution based on the simulation in part (b) is $\hat{\theta} \approx -0.627$, with approximate variance $\widehat{\text{Var}}(\hat{\theta}) \approx 0.021$. Since the mean is close to the true value, and the variance is small, the estimates appear to be approximately unbiased (i.e., approximately centered around the true value θ).
- e. The approximate variance of the sampling distribution is $\widehat{\text{Var}}(\hat{\theta}) \approx 0.021$. Large sample theory predicts that for large n , $\text{Var}(\hat{\theta}) = (1 - \theta^2)/n \approx 0.013$. These two values are relatively close.
4. (Cryer & Chan, Exercise 7.31) The following code simulates the time series in question:

```
set.seed(100)
n <- 48
phi <- 0.7
y <- arima.sim(model = list(ar = c(phi)), n = n)
```

Next we fit an AR(1) model to this simulated data using maximum likelihood:

```
fit <- arima(y, order = c(1, 0, 0), include.mean = FALSE, method = "ML")
fit

##
## Call:
## arima(x = y, order = c(1, 0, 0), include.mean = FALSE, method = "ML")
##
## Coefficients:
##          ar1
##          0.442
## s.e.    0.128
##
## sigma^2 estimated as 1.1:  log likelihood = -70.6,  aic = 143
```

Large sample theory predicts that for large n , the MLE $\hat{\phi}$ of ϕ is approximately unbiased and normally distributed with variance $\text{Var}(\hat{\theta}) \approx (1 - \phi^2)/n \approx 0.011$.

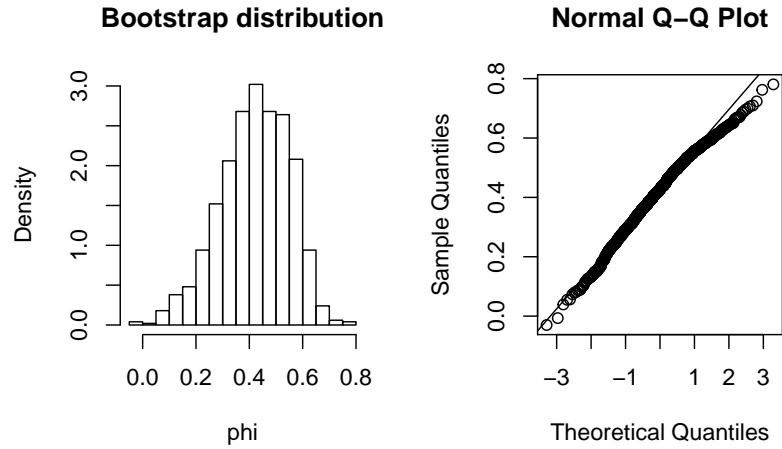
The following function produces a histogram showing the estimated distribution of ϕ based on given bootstrap estimates, as well as a normal Q-Q plot showing adherence to normality (of lack thereof):

```
phi_dist_plots <- function(boot) {
  phi <- boot[,1]
  old_par <- par(mfrow = c(1, 2))
  hist(phi, prob = TRUE, breaks = 20,
       main = "Bootstrap distribution", xlab = "phi")
  qqnorm(phi)
  qqline(phi)
  par(old_par)
}
```

We now compute the estimated distribution of ϕ using four different bootstrapping techniques.

- **Method I:** Bootstrap conditional on first $p + d$ initial values, and errors are assumed normally distributed.

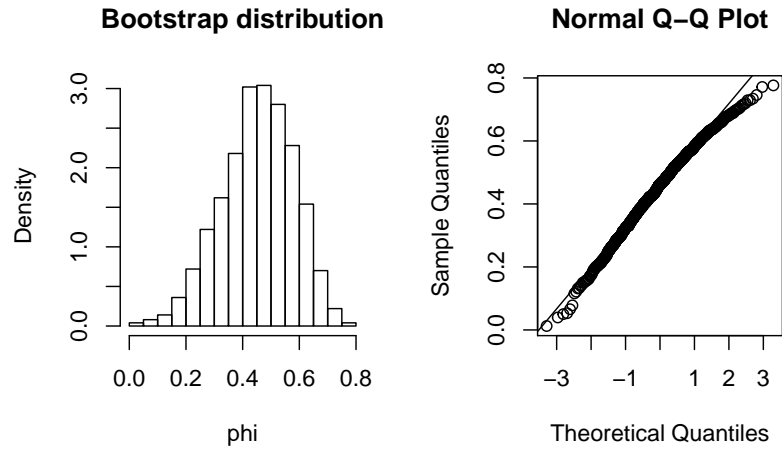
```
boot <- arima.boot(fit, cond.boot = TRUE, is.normal = TRUE,
                  B = 1000, init = y)
phi_dist_plots(boot)
```



This distribution has approximate mean $\hat{\theta} \approx 0.421$, and approximate variance $\widehat{\text{Var}}(\hat{\theta}) \approx 0.017$.

- **Method II:** Bootstrap conditional on first $p + d$ initial values, and errors are sampled with replacement from the residuals of the fitted model.

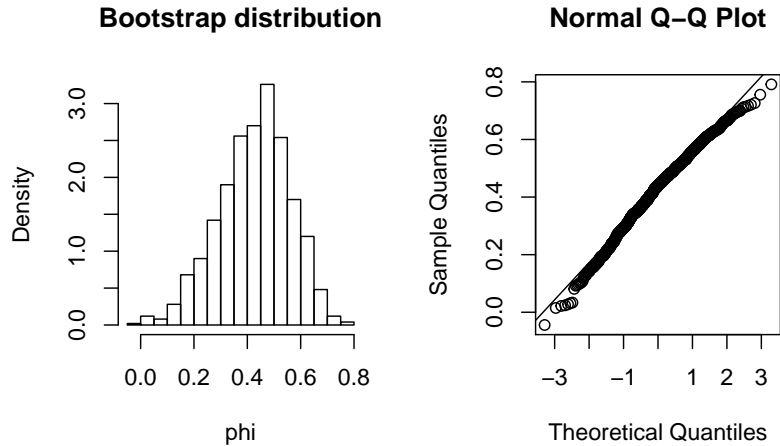
```
boot <- arima.boot(fit, cond.boot = TRUE, is.normal = FALSE,
                  B = 1000, init = y)
phi_dist_plots(boot)
```



This distribution has approximate mean $\hat{\theta} \approx 0.453$, and approximate variance $\widehat{\text{Var}}(\hat{\theta}) \approx 0.017$.

- **Method III:** Bootstrap simulates stationary realizations from the fitted model, and errors are assumed normally distributed.

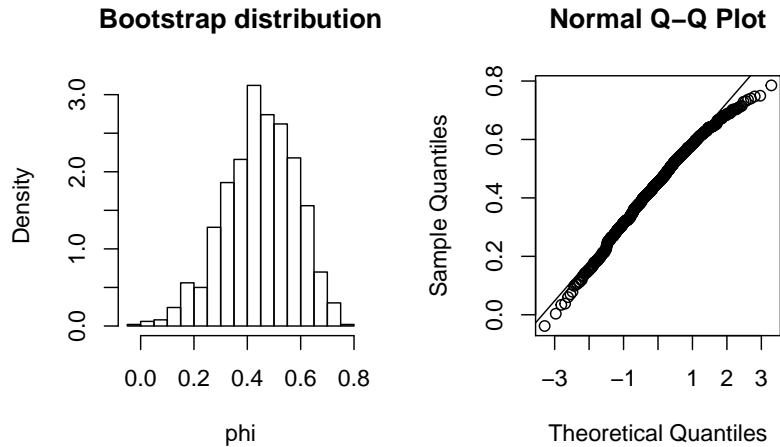
```
boot <- arima.boot(fit, cond.boot = FALSE, is.normal = TRUE,
                  B = 1000, ntrans = 100, init = y)
phi_dist_plots(boot)
```



This distribution has approximate mean $\hat{\theta} \approx 0.427$, and approximate variance $\widehat{\text{Var}}(\hat{\theta}) \approx 0.018$.

- **Method IV:** Bootstrap simulates stationary realizations from the fitted model, and errors are sampled with replacement from the residuals of the fitted model.

```
boot <- arima.boot(fit, cond.boot = FALSE, is.normal = FALSE,
                  B = 1000, ntrans = 100, init = y)
phi_dist_plots(boot)
```



This distribution has approximate mean $\hat{\theta} \approx 0.447$, and approximate variance $\widehat{\text{Var}}(\hat{\theta}) \approx 0.018$.

Each of the four bootstrap methods yield similar results. In each case, the mean bootstrap estimate of ϕ is very close to the maximum likelihood estimate, and the bootstrap estimate of the variance is very close to the asymptotic variance of the MLE. However, each of the bootstrap distributions deviates somewhat from normality in the tails. Moreover, none of the estimates – including the MLE – are close to the true value of $\phi = 0.7$. This is likely due to the small number of observations ($n = 48$).

5. (Cryer & Chan, Exercise 8.9) The following fits both an AR(1) model and an IMA(1, 1) model to the data:

```
data(robot)
ar1_fit <- arima(robot, c(1, 0, 0))
ima11_fit <- arima(robot, c(0, 1, 1))
```

The results of the AR(1) fit are as follows:

```
ar1_fit
```

```
##
## Call:
## arima(x = robot, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##          0.307      0.001
## s.e.    0.053      0.000
##
## sigma^2 estimated as 6.48e-06:  log likelihood = 1476,  aic = -2947
```

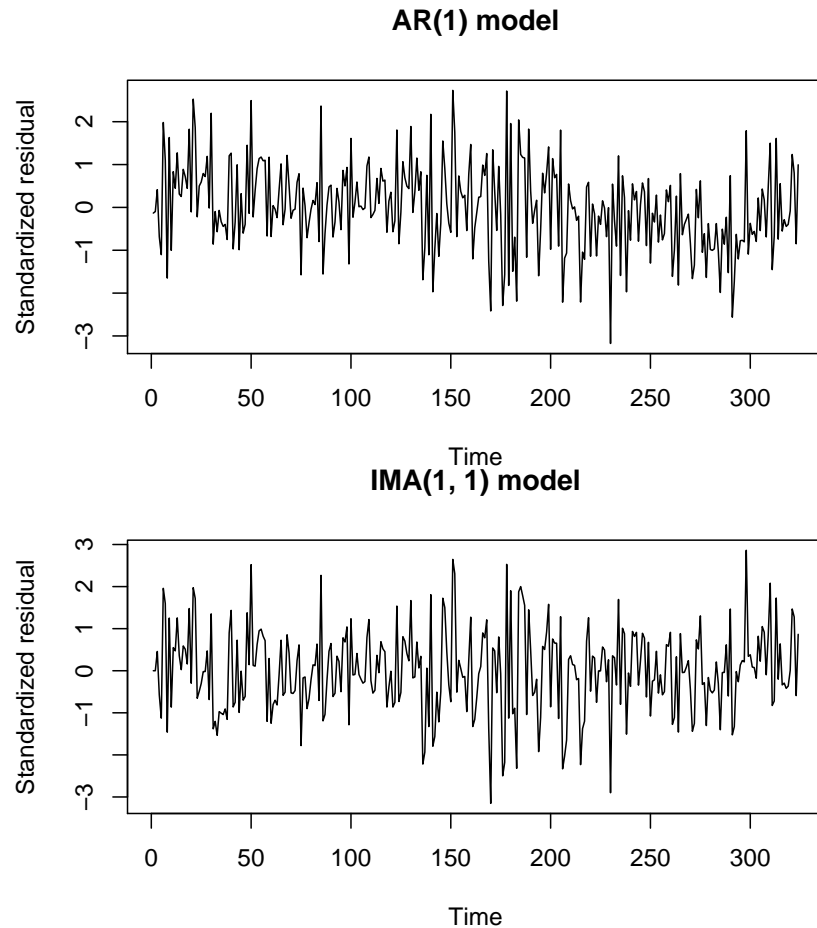
And for the IMA(1, 1) fit:

```
ima11_fit
```

```
##
## Call:
## arima(x = robot, order = c(0, 1, 1))
##
## Coefficients:
##          ma1
##         -0.871
## s.e.    0.039
##
## sigma^2 estimated as 6.07e-06:  log likelihood = 1481,  aic = -2960
```

The two models have comparable AICs. Residual plots for the two models are as follows:

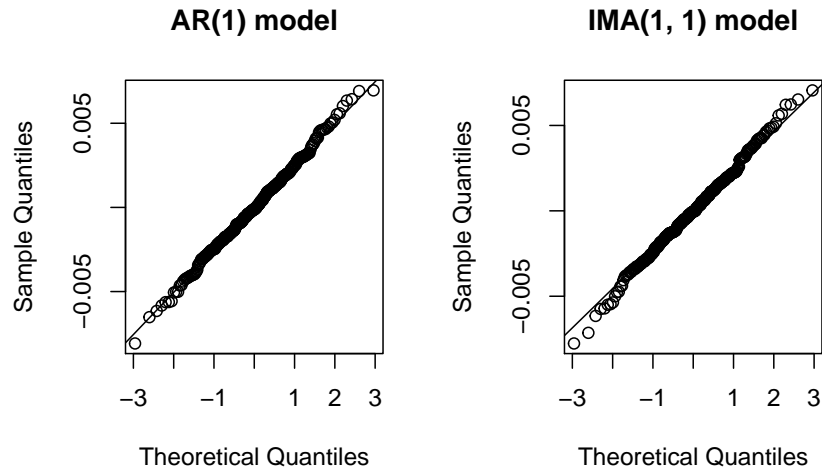
```
plot(rstandard(ar1_fit), main = "AR(1) model",
     ylab = "Standardized residual", type = "l")
plot(rstandard(ima11_fit), main = "IMA(1, 1) model",
     ylab = "Standardized residual", type = "l")
```



The plot for the AR(1) model displays a subtle dip in the value of the residuals later in time, while the plot for the IMA(1, 1) model remains relatively level throughout. Both plots could reasonably be displaying white noise.

Normal Q-Q plots of the residuals for both models are as follows:

```
old_par <- par(mfrow = c(1, 2))
qqnorm(residuals(ar1_fit), main = "AR(1) model")
qqline(residuals(ar1_fit))
qqnorm(residuals(ima11_fit), main = "IMA(1, 1) model")
qqline(residuals(ima11_fit))
par(old_par)
```

The residuals for both models appear to be approximately normal, although the residuals for the IMA(1, 1) model display a bit more deviation from normality in the tails. In addition, the Shapiro-Wilk test yields the following for the AR(1) model:

```
shapiro.test(residuals(ar1_fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ar1_fit)
## W = 1, p-value = 0.9
```

and the following for the IMA(1, 1) model:

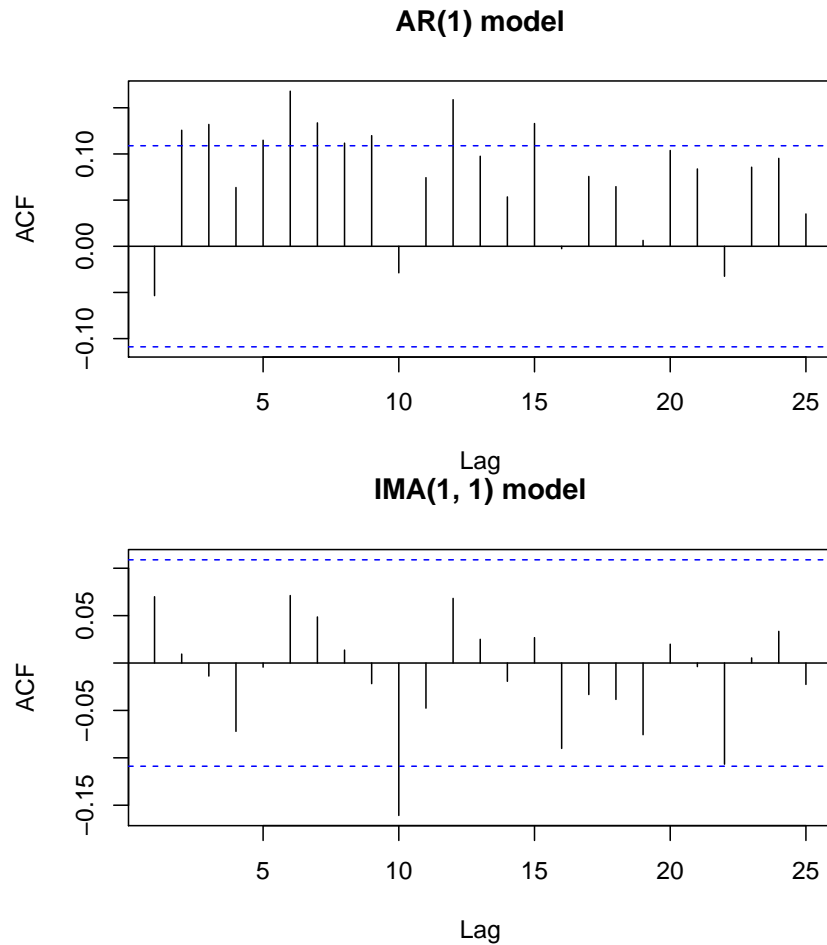
```
shapiro.test(residuals(ima11_fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ima11_fit)
## W = 1, p-value = 0.8
```

Therefore, for both models, we fail to reject the null hypothesis of normality of the residuals on the basis of the Shapiro-Wilk test.

Now for the autocorrelation plots:

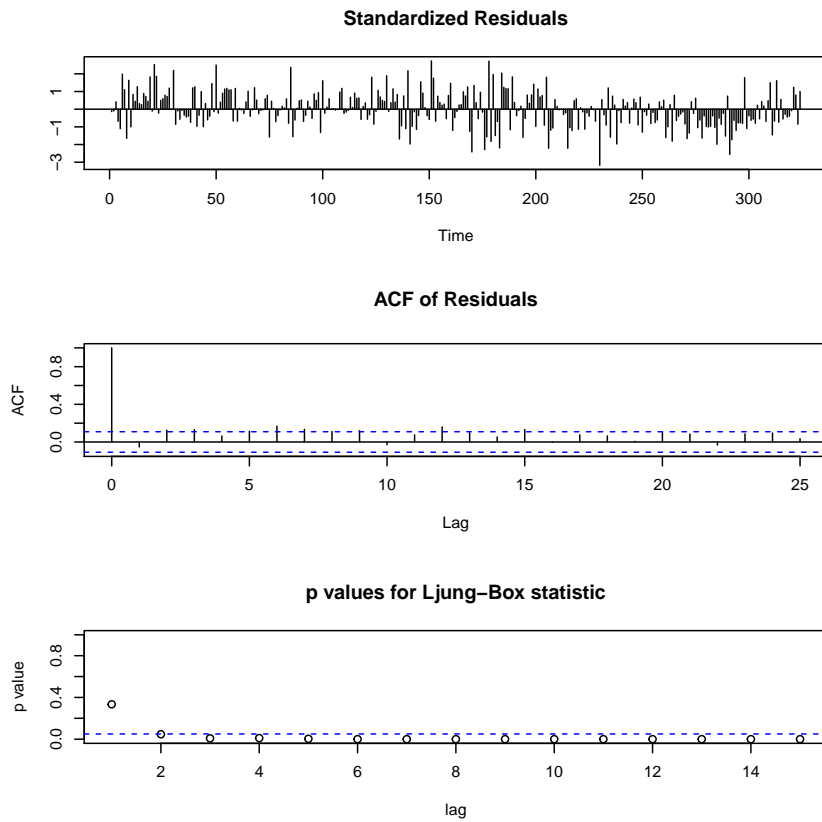
```
acf(residuals(ar1_fit), main = "AR(1) model")
acf(residuals(ima11_fit), main = "IMA(1, 1) model")
```



The sample ACF for the AR(1) model is consistently significantly different from zero for a great deal of the lags shown in the plot. This suggests that the residuals from this model are *not* white noise. On the other hand, the sample ACF of the residuals for the IMA(1, 1) is characteristic of white noise (with the exception of the unusually negative value for lag 10).

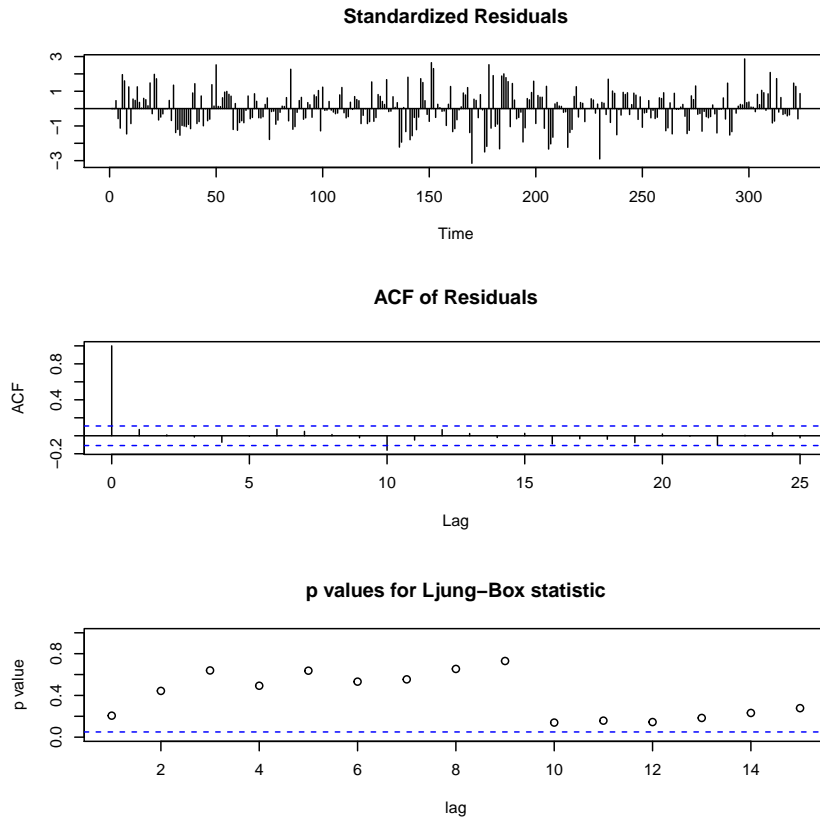
The following produces various diagnostic plots for the AR(1) model, including a plot of p -values from the Ljung-Box test against the number of lags K to consider when computing the associated test statistic:

```
tsdiag(ar1_fit, gof.lag = 15, omit.initial = FALSE)
```



The following produces the same plots for the IMA(1, 1) model:

```
tsdiag(ima11_fit, gof.lag = 15, omit.initial = FALSE)
```



The plot of Ljung-Box test p -values for the AR(1) model indicates that the null hypothesis that the residuals are uncorrelated should likely be rejected, and thus we conclude that the residuals are correlated. On the other hand, the p -values for the IMA(1, 1) model indicate that we should likely *not* reject the null, and thus we conclude that the residuals are uncorrelated.

We also conduct the Ljung-Box test for the specific value of $K = 15$:

```
Box.test(residuals(ar1_fit), lag = 15, type = "Ljung-Box", fitdf = 1)
```

```
##
## Box-Ljung test
##
## data: residuals(ar1_fit)
## X-squared = 60, df = 10, p-value = 4e-08
```

```
Box.test(residuals(ima11_fit), lag = 15, type = "Ljung-Box", fitdf = 1)
```

```
##
## Box-Ljung test
##
## data: residuals(ima11_fit)
## X-squared = 20, df = 10, p-value = 0.2
```

Thus we reject the null for AR(1) model, and fail to reject it for the IMA(1, 1) model

(as the plots above suggested).

On the basis of the above diagnostics, it appears as though the IMA(1, 1) model is a better fit to the data.