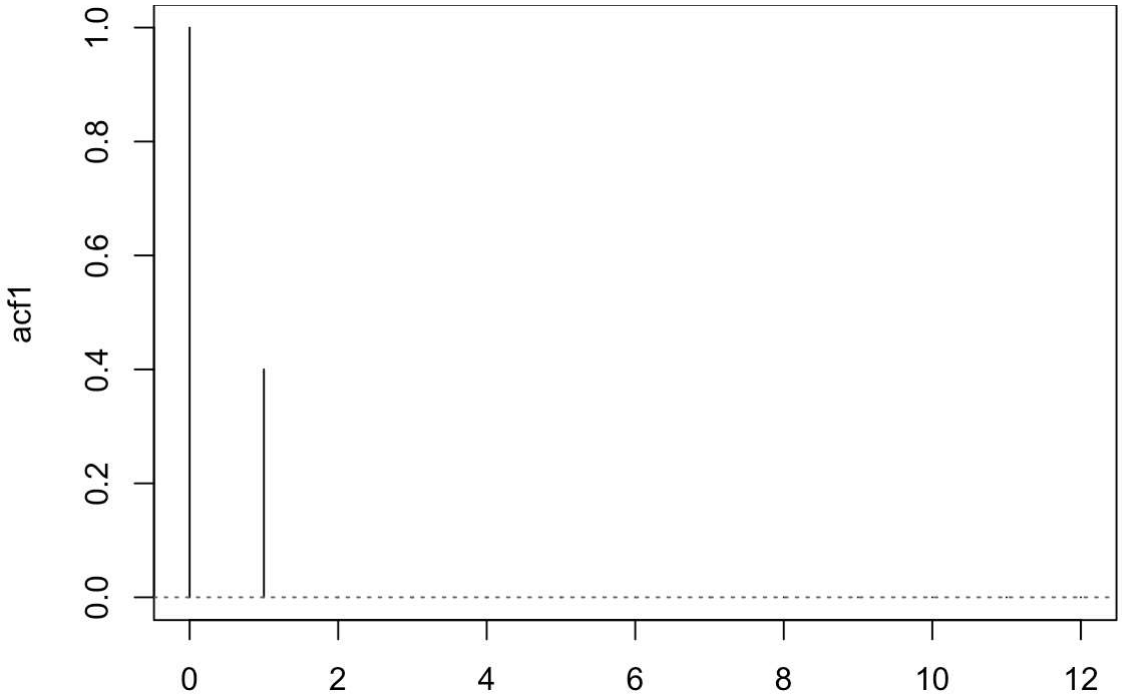
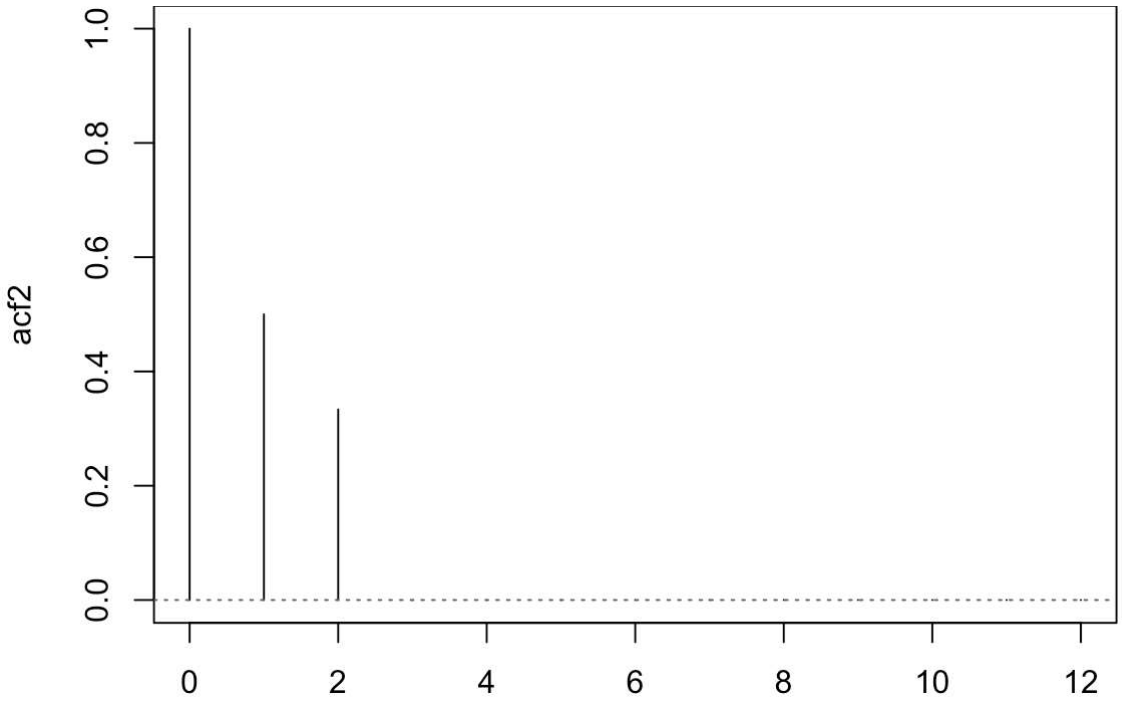
Tutorial 5 Answers

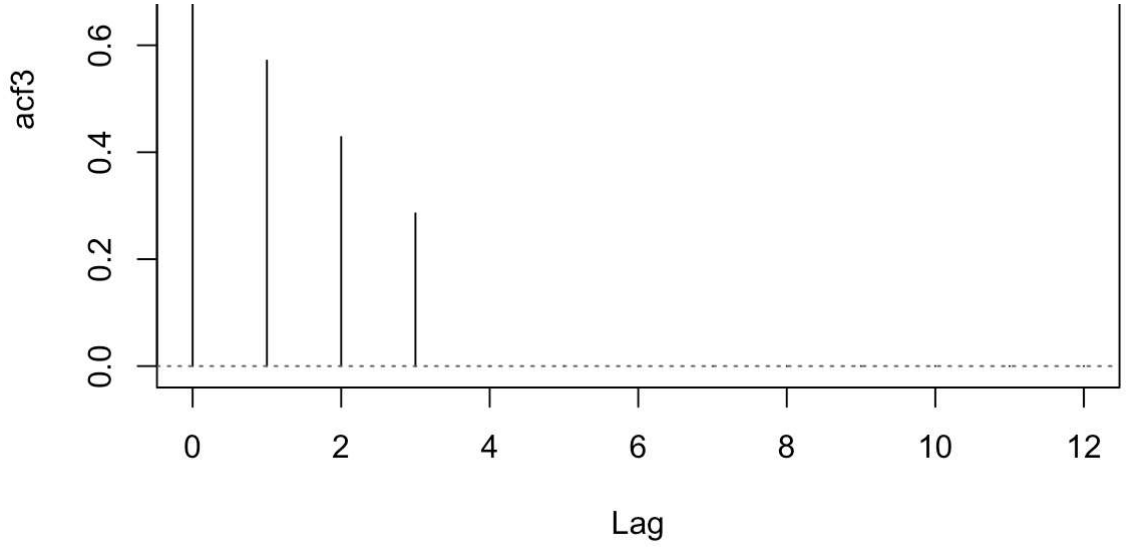
1. First the acf's:

```
theta1 <- 0.5
theta2 <- 0.5
theta3 <- 0.5

acf1 <- ARMAacf(ma=c(theta1), lag.max=12)
acf2 <- ARMAacf(ma=c(theta1,theta2), lag.max=12)
acf3 <- ARMAacf(ma=c(theta1,theta2,theta3), lag.max=12)
```



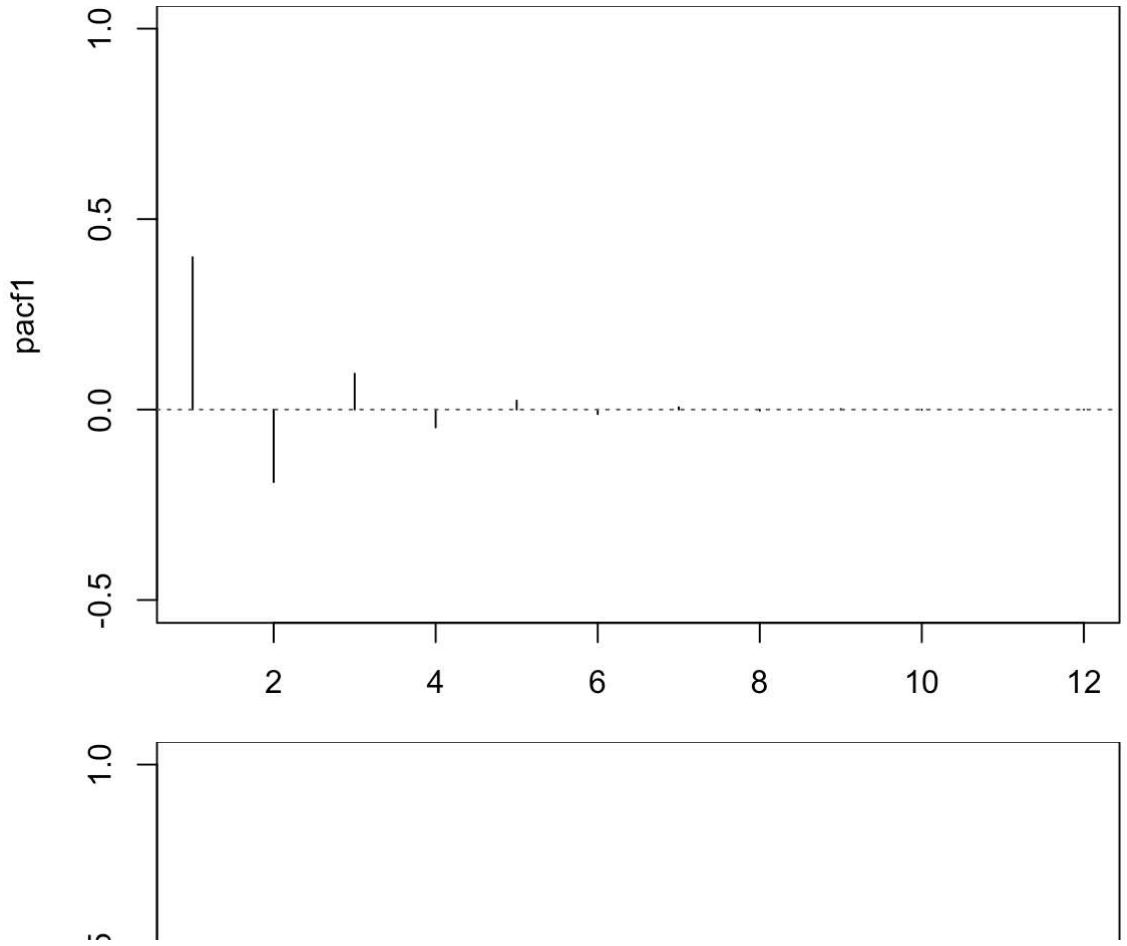


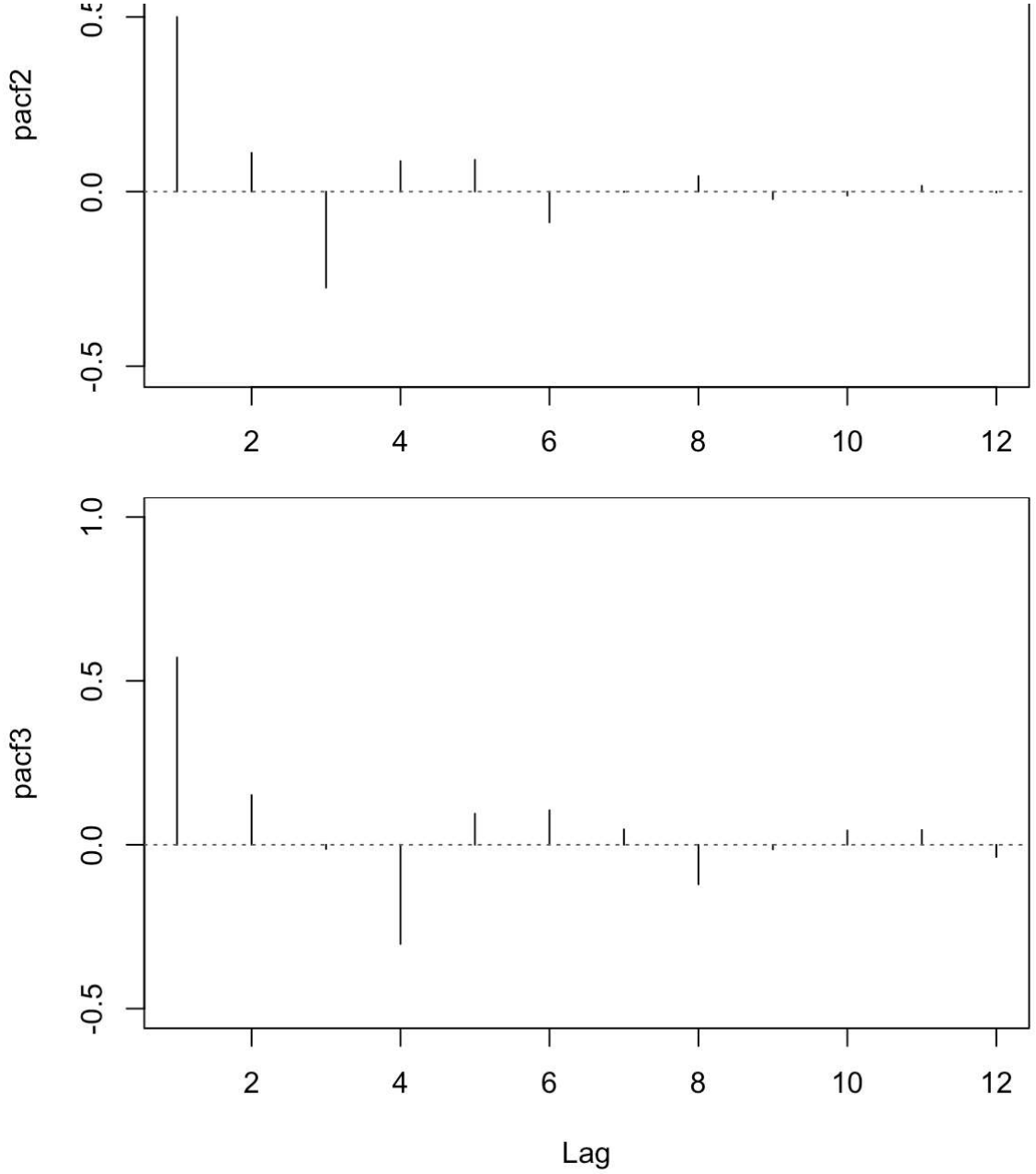


This provides clear illustration of the "cutting off" of the acf after q lags for an MA(q) model. This is the main conclusion. Notice that the values of the autocorrelations for the MA(3) model happen to get smaller in this example, even though the three MA coefficients are all equal $(\theta_1 = \theta_2 = \theta_3 = 0.5)$. We will skip the theoretical derivations but the relationship between the MA coefficients and autocorrelations becomes a bit complicated.

The pacf's are next:

```
pacf1 <- ARMAacf(ma=c(theta1), pacf=TRUE, lag.max=12)
pacf2 <- ARMAacf(ma=c(theta1,theta2), pacf=TRUE, lag.max=12)
pacf3 <- ARMAacf(ma=c(theta1,theta2,theta3), pacf=TRUE, lag.max=12)
```





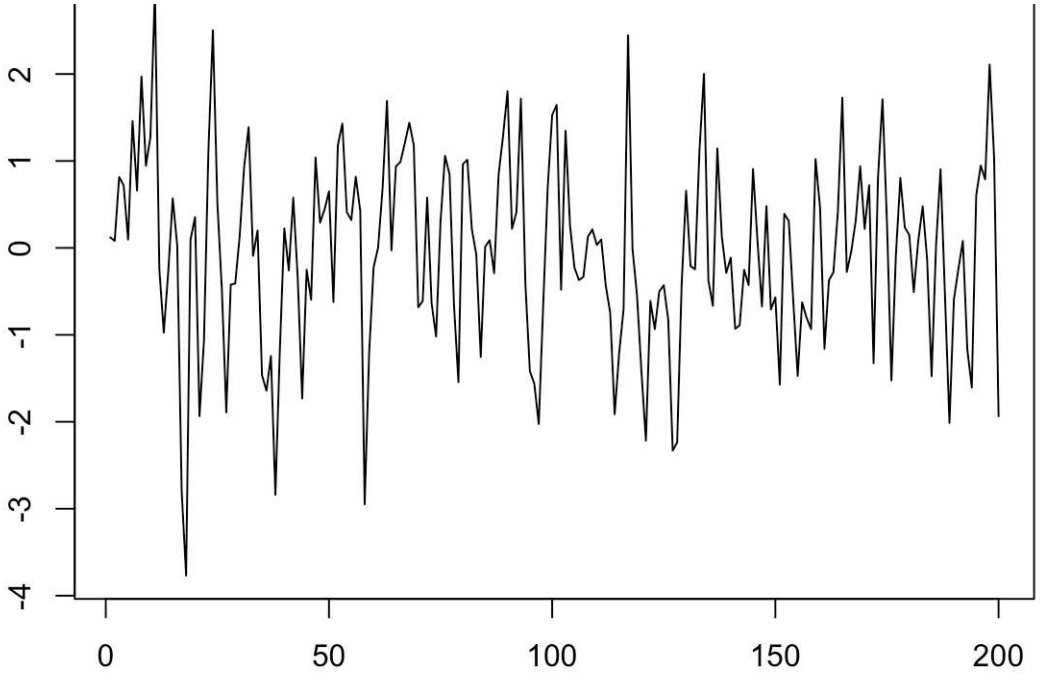
These pacf's show the "declining" feature associated with MA models. Note that negative partial autocorrelations are possible even when all the MA coefficients are positive. Again the relationships are complicated, but the conclusion is that specific pattern of values of the pacf's is difficult to interpret.

2. **a.** The simulation of the MA(1) model will give something that looks somewhat (but not exactly, depending on the randomness of the simulation) like this:

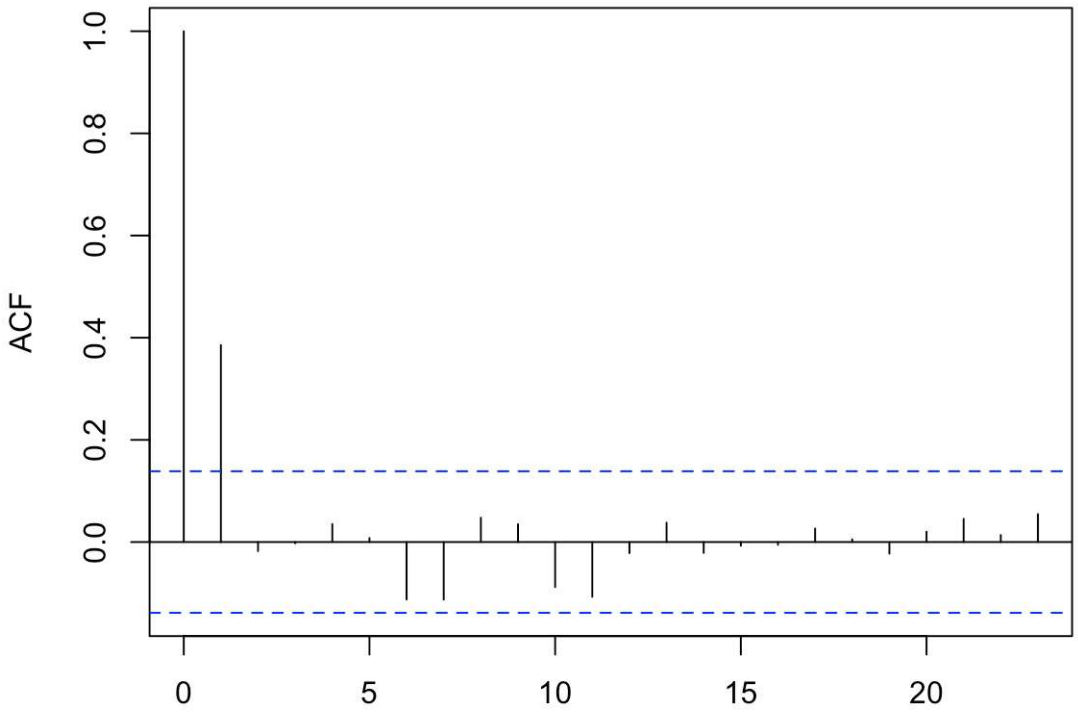
```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```
Y <- arima.sim(n=200, list(ma=0.5))
```

```
plot(Y)
```

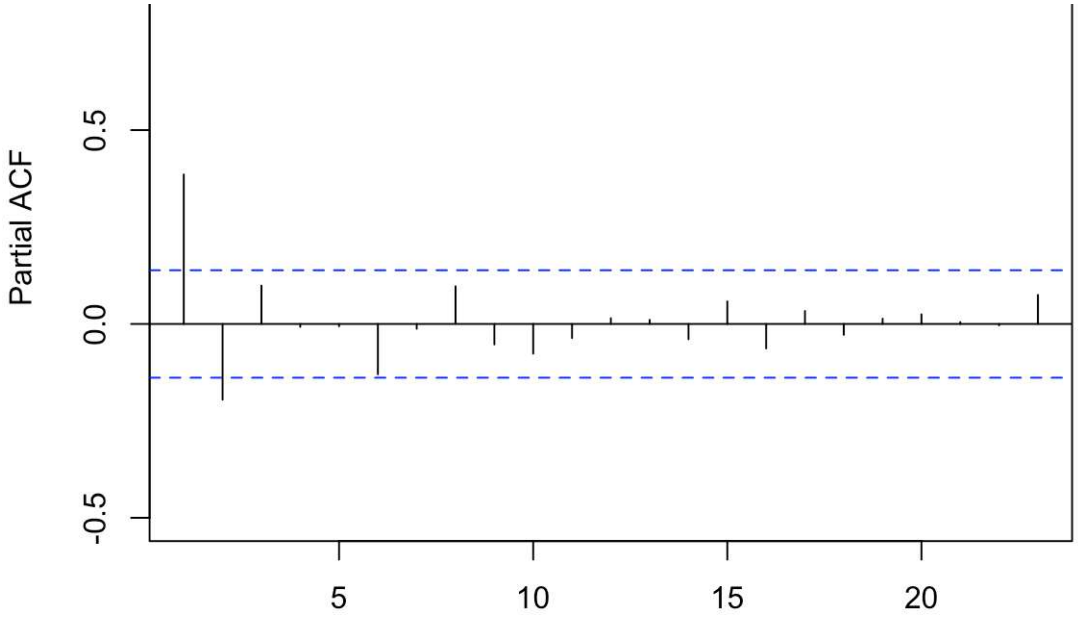



```
acf(Y)
```



This acf illustrates the MA(1) pattern of "cutting off" after the first lag, in the sense that the autocorrelations are small and insignificant (not exactly zero since we only have a sample here) at lags higher than one. The first order autocorrelation shown here is a little below 0.4, the population value from question 1. (Note your particular realisation may differ from this.)

```
pacf(Y)
```



This pacf can reasonably be seen as similar to the theoretical MA(1) pacf above.

b. Here are the estimation and Ljung-Box test commands:

```
ARMA00 <- Arima(Y, order=c(0,0,0))
checkresiduals(ARMA00, plot=FALSE)
```

```
    Ljung-Box test

data:  Residuals from ARIMA(0,0,0) with non-zero mean
Q* = 38.191, df = 10, p-value = 3.515e-05

Model df: 0.   Total lags used: 10
```

```
ARMA01 <- Arima(Y, order=c(0,0,1))
checkresiduals(ARMA01, plot=FALSE)
```

```
    Ljung-Box test

data:  Residuals from ARIMA(0,0,1) with non-zero mean
Q* = 6.8525, df = 9, p-value = 0.6525

Model df: 1.   Total lags used: 10
```

```
ARMA10 <- Arima(Y, order=c(1,0,0))
checkresiduals(ARMA10, plot=FALSE)
```

```
    Ljung-Box test

data:  Residuals from ARIMA(1,0,0) with non-zero mean
Q* = 20.63, df = 9, p-value = 0.0144

Model df: 1.   Total lags used: 10
```

```
ARMA11 <- Arima(Y, order=c(1,0,1))
checkresiduals(ARMA11, plot=FALSE)
```

```
    Ljung-Box test

data:  Residuals from ARIMA(1,0,1) with non-zero mean
Q* = 6.8095, df = 8, p-value = 0.5573

Model df: 2.   Total lags used: 10
```

The ARMA(0,0) and ARMA(1,0) models do not pass the residual autocorrelation test ($p < 0.05$). This reflects that both are misspecified, i.e. in this simulation setting we know the true model is MA(1) (unlike we ever do in practice) and neither ARMA(0,0) nor ARMA(1,0) adequately capture this.

The ARMA(0,1) and ARMA(1,1) models both pass the Ljung-Box tests ($p > 0.05$). Both include the correct MA(1) term. The ARMA(1,1) model also includes a redundant AR(1) term (i.e. coefficient is zero) but this redundancy does not imply misspecification.

Here are the AICc code and statistics:

```
AICc <- rbind(ARMA00$aicc, ARMA01$aicc, ARMA10$aicc, ARMA11$aicc)
rownames(AICc) <- c("ARMA00", "ARMA01", "ARMA10", "ARMA11")
colnames(AICc) <- "AICc"
print(AICc)
```

```
           AICc
ARMA00 605.1673
ARMA01 564.3371
ARMA10 574.6649
ARMA11 566.4172
```

The smallest AICc occurs for the ARMA(0,1) model, which matches the specification uses to generate `Y` in the simulation.

(Note that your numerical results won't exactly match these, and even the conclusions of the Ljung-Box and AICc evaluations may differ.)

c. Compare the results when repeatedly running:

```
Y <- arima.sim(n=20, list(ma=0.5))
print(head(Y))
```

versus

```
set.seed(30004)
Y <- arima.sim(n=20, list(ma=0.5))
print(Y)
```

```
Time Series:
Start = 1
End = 20
Frequency = 1
 [1]  1.084595397  0.456216045 -0.754157840 -2.573838877 -2.607296231
 [6] -2.609465491 -2.715487031 -0.892884219  0.253013586 -1.621653382
[11]  0.770424433  0.470640830 -0.978700760  1.027769031  0.004224224
[16] -0.080036726  0.248864169  1.083275830 -0.437097162 -0.551595077
```

3. Here is the code:

```r
set.seed(5)

# Number of repetitions
reps <- 1000

# Names of the models
Models <- c("ARMA(0,0)","ARMA(0,1)","ARMA(1,0)","ARMA(1,1)")

# AICc storage:
#   each column is a model
#   each row is a repetition
AICc <- matrix(nrow=reps, ncol=4)
colnames(AICc) <- Models

# Ljung-Box test rejection storage:
LB <- AICc

# Loop to repeat 1000 times:
for (r in 1:reps){

  Y <- arima.sim(n=200, list(ma=0.5))

  # Using loops to estimate the four models:
  for (p in 0:1){
    for (q in 0:1){

      # Estimate ARMA(p,q)
      eq <- Arima(Y, order=c(p,0,q))

      # Store AICc
      AICc[r,1+2*p+q] <- eq$aicc

      # Store 1 if LB p-value <0.05, 0 otherwise
      capture.output(LB[r,1+2*p+q] <- 1*(checkresiduals(eq,
                                         plot=FALSE)$p.value<0.05))
    }
  }

}
```

For each model (column of `LB`) we calculate the proportion of the replications where the test rejects, implying residual autocorrelation and hence evidence of an invalid model:

```
print(apply(LB,2,mean))
```

```
ARMA(0,0) ARMA(0,1) ARMA(1,0) ARMA(1,1)
    0.998     0.059     0.501     0.053
```

Notes on these:

- The ARMA(0,0) is misspecified because it does not model autocorrelation at all. The MA(1) autocorrelation is the time series is therefore left in the "residuals" and the LB test will generally reject the null of no autocorrelation. It may not reject in every single replication because the test is not perfect and may commit a Type II error.
- The ARMA(0,1) model is the correct model. Nevertheless the Ljung Box test rejects the null hypothesis in approximately 5% of the replications, corresponding the significance level of 0.05 we are using for the test. These outcomes are Type I errors for the test.
- The ARMA(1,0) model is misspecified for the true ARMA(0,1) model. In an informal sense the ARMA(1,0) is "less misspecified" than the ARMA(0,0) because at least the ARMA(1,0) is fitted something for the first order autocorrelation, even though its particular functional form is not correct. As a result, the Ljung-Box test is rejecting the null roughly half the time. In those other half of the replications where the null is not rejected (Type II errors), the AICc would be used to choose between multiple models that pass the LB test.
- The ARMA(1,1) model is not misspecified since it contains the MA(1) component, but it also contains a redundant AR(1) component. This redundant component (i.e. including an AR term where the coefficient is zero) would not be expected to induce residual autocorrelation, and indeed the Ljung-Box rejections look very similar for the ARMA(0,1) and ARMA(1,1) models.

Next we tabulate which model is chosen by the AICc at each replication.

```
ChosenModels <- Models[apply(AICc,1,which.min)]
print(table(ChosenModels)/reps)
```

```
ChosenModels
ARMA(0,1) ARMA(1,0) ARMA(1,1)
    0.842     0.017     0.141
```

The AICc nearly always selects a valid model (ARMA(0,1) or ARMA(1,1)) with the "correct" MA(1) model being chosen most often. It is a theoretical characteristic of the AIC that it retains some tendency to select a model that is larger than the "correct" model, in the sense of having a non-zero probability of including redundant terms. Including redundant terms does not invalidate the model as a representation of the conditional mean. It would remain a question for further analysis how this influence the forecasting performance of the model.