Tutorial 7 Answers

Here is the code for a particular data generating process (`phi1=0.9`). This can be varied one run at a time, or a loop around these values added.

```
library(forecast)
library(urca)
set.seed(42)

# Sample size
n <- 100

# True value of phi1
phi1 <- 0.9

# Lag length of AR model
p <- 2
t <- (p+1):n

# Simulation set up
reps <- 1000

# Matrix to indicate for each replication
#   1: unit root / differencing chosen
#   2: no differencing chosen
# by ADF test (column 1), AIC (column 2)
ur <- matrix(nrow=reps, ncol=2)
colnames(ur) <- c("ADF", "AIC")

# Matrix of forecast errors for each replication
#   column 1: ARIMA(1,1,0) model
#   column 2: ARIMA(2,0,0) model
#   column 3: model chosen by ADF test
#   column 4: model chosen by AIC
ForecastErrors <- matrix(nrow=reps, ncol=4)
colnames(ForecastErrors) <- c("ARIMA110", "ARIMA200", "ADF", "AIC")

for (r in 1:reps){

  # Simulate time series of length n+1
  #   ARIMA(0,1,0) if phi1=1, ARIMA(1,0,0) otherwise
  if (phi1==1){
    Y <- arima.sim(n=n+1, model=list(order=c(0,1,0)))[-1]
  } else {
    Y <- arima.sim(n=n+1, model=list(ar=phi1))
  }

  # Keep aside Y[n+1] for forecast evaluation
  Yf <- Y[n+1]
  # Use Y1,...,Yn as the estimation sample
  Y <- Y[1:n]

  # Estimation of AR(2) model
  ARIMA200 <- lm(Y[t]~Y[t-1]+Y[t-2])
```

```
  # Estimation of ARIMA(1,1,0) model
  DY <- Y[t]-Y[t-1]
  DY1 <- Y[t-1]-Y[t-2]
  ARIMA110 <- lm(DY[t]~DY[t-1])


  # Compute one step ahead forecast errors
  b1 <- ARIMA110$coefficients
  ForecastErrors[r,1] <- Yf-(Y[n]+b1[1]+b1[2]*DY[length(DY)])

  b2 <- ARIMA200$coefficients
  ForecastErrors[r,2] <- Yf-(b2[1]+b2[2]*Y[n]+b2[3]*Y[n-1])


  # ADF test with drift, lags=1 (for p=2 model)
  # p>0.05 => H0:unit root not rejected, i.e. differencing indicated
  ur[r,1] <- 1*(punitroot(ur.df(Y, type="drift", lags=1)@teststat[1],
                    N=n, trend="c", statistic="t")>0.05)

  # If ADF test selects unit root use ARIMA110 forecast error:
  if (ur[r,1]==1){
    ForecastErrors[r,3] <- ForecastErrors[r,1]
  } else {
    # otherwise use ARIMA200 forecast error:
    ForecastErrors[r,3] <- ForecastErrors[r,2]
  }

  # Use AIC to select between AR(2) and ARIMA(1,1,0)
  ur[r,2] <- 1*(AIC(ARIMA110)<AIC(ARIMA200))

  # If AIC selects unit root use ARIMA110 forecast error:
  if (ur[r,2]==1){
    ForecastErrors[r,4] <- ForecastErrors[r,1]
  } else {
    # otherwise use ARIMA200 forecast error:
    ForecastErrors[r,4] <- ForecastErrors[r,2]
  }

}
```

```
# Selection proportions:
UnitRootSelected <- apply(ur,2,mean)*100
```

```
                 ADF   AIC
UnitRootSelected 70.6% 63.5%
```

```
# RMSEs:
RMSE <- sqrt(apply(ForecastErrors^2,2,mean))
```

```
     ARIMA(1,1,0) ARIMA(2,0,0)     ADF     AIC
RMSE      1.05121      1.02576 1.04707 1.04603
```

Looping over all the data generating processes:

Looping over all the data generating processes:

```
Percent Unit Root Selected:

            ADF  AIC
phi1=1     95.8 93.3
phi1=0.95  89.4 84.6
phi1=0.9   70.6 63.5
phi1=0.85  44.3 37.8
phi1=0.8   20.7 19.4
phi1=0.7    2.3  3.7
phi1=0.6    0.2  0.9
phi1=0.5    0.0  0.1

RMSE:

          ARIMA110 ARIMA200    ADF    AIC
phi1=1      0.9877   1.0039 0.9912 0.9904
phi1=0.95   1.0300   1.0141 1.0328 1.0304
phi1=0.9    1.0512   1.0258 1.0471 1.0460
phi1=0.85   1.0887   1.0474 1.0744 1.0721
phi1=0.8    1.0889   1.0275 1.0389 1.0458
phi1=0.7    1.1163   1.0521 1.0599 1.0575
phi1=0.6    1.0941   1.0042 1.0076 1.0108
phi1=0.5    1.1554   1.0161 1.0161 1.0164
```

Some conclusions about unit root detection:

- In this experiment the AIC demonstrates some tendency to find a unit root in a time series a little less than the ADF test, regardless of whether or not there is a unit root.
- When $\phi_1 = 1$ (unit root) the ADF test finds a unit root 95.8% of the time, consistent (approximately) with the 5% significance level of the test. The AIC finds a unit root 93.3% of the time, implying a "signficance level" (although it's not constructed as a formal test) of approximately 0.067. This does not appear to be a huge difference on the surface.
- When $phi_1 < 1$ the difference between the ADF and AIC selection proportions becomes as high as about 7%. A noticeable but not huge difference. The implications of this difference are explored using forecast performance.
- Overall the AIC provides a reasonable way to distinguish between unit root or not in this experiment.

Some conclusions about forecasting:

- Many of the RMSEs are approximately the same. The biggest differences occur when ϕ_1 is away from 1, i.e. $\phi_1 = 0.5, 0.6$. In this case the ARIMA(1,1,0) model is clearly misspecified, and its forecasting performance suffers (15% worse RMSE than the correctly specified ARIMA(2,0,0)). However both ADF and AIC do a good job in selecting the correct model here.
- When $phi_1 = 1$ the RMSEs are all roughly the same. This is interesting - if there is a unit root then it doesn't matter what we do, at least for one-step-ahead forecasting.

- Overall the RMSEs for the ADF and AIC methods are practically the same. There is no benefit in making this decision in this experiment in using the specialised ADF, we could just as easily have used the AIC.
- Maybe interestingly, there is a “grey area’ around $\phi_1 = 0.95$ to 0.85 where the selected models do worse than the ARIMA(2,0,0), although the RMSE differences are small.

Final note

Relatively there is lots of code in this tutorial, which hopefully you can master, but the most important thing is to understand the simulation approach. We are using simulation to replicate the frequentist statistical concept of a sampling distribution and hence to obtain experimental evidence on the performance of some tests and forecasts. Also note a limitation of this experiment - we have only tried a limited range of data generating processes and models and so the curreent results are only specific to that setting.