Homework 8

Ron Cordell, Lei Yang, Subhashini Raghunathan

Apr 7, 2016

Build an univariate linear time series model (i.e AR, MA, and ARMA models) using the series in hw08_series.csv.

Use all the techniques that have been taught so far to build the model, including date examination, data visualization, etc.

```
library(astsa)    # Time series package by Shummway and Stoffer
```

```
## Warning: package 'astsa' was built under R version 3.2.3
```

```
library(zoo)      # time series package
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.2.4
```

```
## Loading required package: timeDate
```

```
## Warning: package 'timeDate' was built under R version 3.2.3
```

```
## This is forecast 6.2
##
##
## Attaching package: 'forecast'
##
## The following object is masked from 'package:astsa':
##
##     gas
```

```
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 3.2.4
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 3.2.4

## Loading required package: TTR

## Warning: package 'TTR' was built under R version 3.2.4

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
setwd("C:/Subha/WS271-Regression/Labs/Data")

data = read.csv("hw08_series.csv")
data.ts = ts(data=data$x)


# 1. Examining the Data

str(data)
```

```
## 'data.frame':    372 obs. of  2 variables:
##  $ X: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ x: num  40.6 41.1 40.5 40.1 40.4 41.2 39.3 41.6 42.3 43.2 ...
```

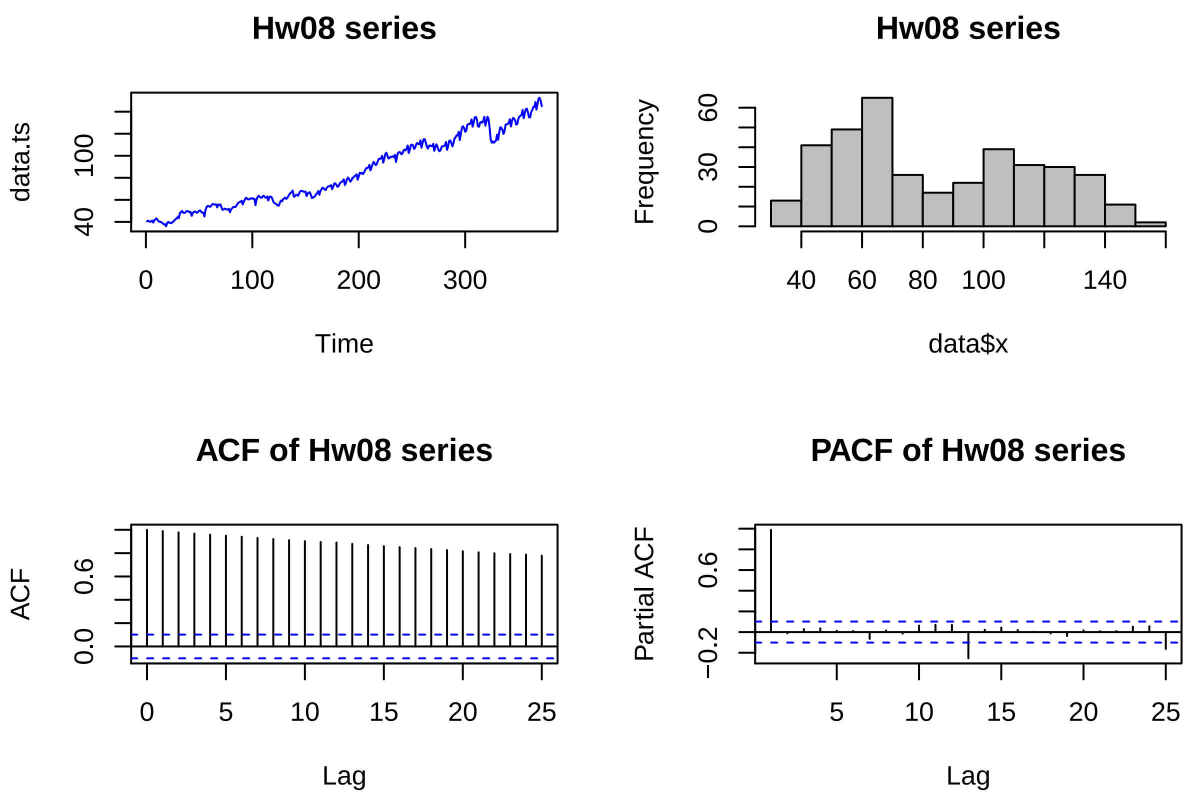
```
summary(data)
```

```
##        X                x
##  Min.   :  1.00   Min.   : 36.00
##  1st Qu.: 93.75   1st Qu.: 57.38
##  Median :186.50   Median : 76.45
##  Mean   :186.50   Mean   : 84.83
##  3rd Qu.:279.25   3rd Qu.:111.53
##  Max.   :372.00   Max.   :152.60
```

```
head(data, 10)
```

```
##     X    x
## 1   1 40.6
## 2   2 41.1
## 3   3 40.5
## 4   4 40.1
## 5   5 40.4
## 6   6 41.2
## 7   7 39.3
## 8   8 41.6
## 9   9 42.3
## 10 10 43.2
```

```
# 2. Data Visualization
par(mfrow=c(2,2))
plot.ts(data.ts, main="Hw08 series",
        col="blue")
hist(data$x, col="gray", main="Hw08 series")
acf(data.ts, main="ACF of Hw08 series")
pacf(data.ts, main="PACF of Hw08 series")
```



All the steps to support your final model need to be shown clearly. Show that the assumptions underlying the model are valid. ** Which model seems most reasonable in terms of satisfying the model's underling assumption?**

From the series plot it is clear that this is not a stationary series. It strongly resembles random walk with drift. It has a strong upward trend and does not come down. The ACF shows high correlation even after 25 lags, and PACF immediately drops to 0.

Given this, it is clear that AR and MA and ARMA models are insufficient to model this series. Still, let's try it.

```
#first, let's try several MA models

best_aic_ma = 10000
best_order_ma = 999
all_aics_ma = vector("list", 30)

for(i in 1:30) {
  data.fit <- arima(data.ts, order=c(0,0,i))
  if(data.fit$aic < best_aic_ma ){
    best_aic_ma = data.fit$aic
    best_order_ma = length(data.fit$coef) -1
    best_model_ma = data.fit
  }
  all_aics_ma[i] = data.fit$aic
}
```

```
## Warning in arima(data.ts, order = c(0, 0, i)): possible convergence
```

```
## problem: optim gave code = 1

## Warning in arima(data.ts, order = c(0, 0, i)): possible convergence
## problem: optim gave code = 1

## Warning in arima(data.ts, order = c(0, 0, i)): possible convergence
## problem: optim gave code = 1

## Warning in arima(data.ts, order = c(0, 0, i)): possible convergence
## problem: optim gave code = 1

## Warning in arima(data.ts, order = c(0, 0, i)): possible convergence
## problem: optim gave code = 1

## Warning in arima(data.ts, order = c(0, 0, i)): possible convergence
## problem: optim gave code = 1

## Warning in arima(data.ts, order = c(0, 0, i)): possible convergence
## problem: optim gave code = 1

## Warning in arima(data.ts, order = c(0, 0, i)): possible convergence
## problem: optim gave code = 1
```

```
#the AIC keep reducing; order 30 is the best model
all_aics_ma
```

```
## [[1]]
## [1] 3183.908
##
## [[2]]
## [1] 2769.159
##
## [[3]]
## [1] 2578.796
##
## [[4]]
## [1] 2466.179
##
## [[5]]
## [1] 2204.439
##
## [[6]]
## [1] 2118.251
##
## [[7]]
## [1] 2085.401
##
## [[8]]
## [1] 2020.487
##
## [[9]]
```

```
## [1] 1981.605
##
## [[10]]
## [1] 1838.152
##
## [[11]]
## [1] 1838.041
##
## [[12]]
## [1] 1823.238
##
## [[13]]
## [1] 1791.933
##
## [[14]]
## [1] 1768.5
##
## [[15]]
## [1] 1739.158
##
## [[16]]
## [1] 1729.897
##
## [[17]]
## [1] 1706.23
##
## [[18]]
## [1] 1709.276
##
## [[19]]
## [1] 1677.502
##
## [[20]]
## [1] 1642.782
##
## [[21]]
## [1] 1650.693
##
## [[22]]
## [1] 1631.677
##
## [[23]]
## [1] 1633.588
##
## [[24]]
## [1] 1609.405
##
## [[25]]
## [1] 1595.974
##
## [[26]]
## [1] 1582.868
##
## [[27]]
```

```
## [1] 1590.013
##
## [[28]]
## [1] 1570.262
##
## [[29]]
## [1] 1571.009
##
## [[30]]
## [1] 1553.305
```

```r
best_order_ma
```

```
## [1] 30
```

```r
best_model_ma
```

```
##
## Call:
## arima(x = data.ts, order = c(0, 0, i))
##
## Coefficients:
##          ma1     ma2     ma3     ma4     ma5     ma6     ma7     ma8
##       1.2002  1.3557  1.4998  1.7458  2.0470  2.2008  2.2892  2.4452
## s.e.  0.0533  0.0878  0.1156  0.1391  0.1599  0.1801  0.1967  0.2071
##          ma9    ma10    ma11    ma12    ma13    ma14    ma15    ma16
##       2.4625  2.5999  2.6202  3.3582  3.3944  3.0427  2.9640  2.8514
## s.e.  0.2112  0.2109  0.2131  0.2142  0.2296  0.2365  0.2446  0.2487
##         ma17    ma18    ma19   ma20    ma21    ma22    ma23    ma24
##       3.1427  3.1124  2.5840  2.475  2.2495  2.1887  1.9011  1.8863
## s.e.  0.2496  0.2502  0.2414  0.239  0.2315  0.2202  0.2080  0.1892
##         ma25    ma26    ma27    ma28    ma29   ma30  intercept
##       1.6513  1.2144  0.9178  0.5830  0.5494  0.551    85.2568
## s.e.  0.1692  0.1463  0.1229  0.1015  0.0770  0.059     5.4287
##
## sigma^2 estimated as 2.87:  log likelihood = -744.65,  aic = 1553.3
```

```r
#now let's try AR models. Here we find that after order 3, the series cannot be estimated because of no
best_aic_ar = 10000
best_order_ar = 999
all_aics_ar = vector("list", 3)


for(i in 1:3) {
  data.fit <- arima(data.ts, order=c(i,0,0))
  if(data.fit$aic < best_aic_ar ){
    best_aic_ar = data.fit$aic
    best_order_ar = length(data.fit$coef) -1
    best_model_ar = data.fit
  }
  all_aics_ar[i] = data.fit$aic
}
```

```r
#AICS are quite similar for all 3
all_aics_ar
```

```
## [[1]]
## [1] 1798.826
##
## [[2]]
## [1] 1798.671
##
## [[3]]
## [1] 1786.825
```

```r
best_order_ar
```

```
## [1] 3
```

```r
best_model_ar
```

```
##
## Call:
## arima(x = data.ts, order = c(i, 0, 0))
##
## Coefficients:
##          ar1      ar2     ar3  intercept
##       0.9061  -0.0994  0.1922    91.5517
## s.e.  0.0511   0.0692  0.0511    45.2799
##
## sigma^2 estimated as 6.839:  log likelihood = -888.41,  aic = 1786.82
```

```r
#for arma, (1,0,1) is the only model that R will estimate because of non-stationarity in higher models.
best_aic_arma = 10000
best_order_arma = 999
all_aics_arma = vector("list", 30)


for(i in 1:1) {
  data.fit <- arima(data.ts, order=c(i,0,i))
  if(data.fit$aic < best_aic_arma ){
    best_aic_arma = data.fit$aic
    best_order_arma = length(data.fit$coef) -1
    best_model_arma = data.fit
  }
  all_aics_arma[i] = data.fit$aic
}
```

```
## Warning in arima(data.ts, order = c(i, 0, i)): possible convergence
## problem: optim gave code = 1
```

```r
all_aics_arma
```

```
## [[1]]
## [1] 1806.361
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
##
## [[10]]
## NULL
##
## [[11]]
## NULL
##
## [[12]]
## NULL
##
## [[13]]
## NULL
##
## [[14]]
## NULL
##
## [[15]]
## NULL
##
## [[16]]
## NULL
##
## [[17]]
## NULL
##
## [[18]]
## NULL
##
```

```
## [[19]]
## NULL
##
## [[20]]
## NULL
##
## [[21]]
## NULL
##
## [[22]]
## NULL
##
## [[23]]
## NULL
##
## [[24]]
## NULL
##
## [[25]]
## NULL
##
## [[26]]
## NULL
##
## [[27]]
## NULL
##
## [[28]]
## NULL
##
## [[29]]
## NULL
##
## [[30]]
## NULL
```

```
best_order_arma
```

```
## [1] 2
```

```
best_model_arma
```

```
##
## Call:
## arima(x = data.ts, order = c(i, 0, i))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.9982  0.0745    97.1995
## s.e.  0.0025  0.0622    43.7234
##
## sigma^2 estimated as 7.249:  log likelihood = -899.18,  aic = 1806.36
```

- Based on the above, it would seem that MA(30) is the best model to pick. It's AIC is 1553. In contrast, the best AR model AR(3) has AIC of 1787.
- An MA(q) process is stationary, so the underlying assumptions are satisfied.
- The assumptions for the AR(p) model are that the series being modeled is stationary. As we have see, R does not allow us to estimate AR models for this series where $p > 3$ because it detects that these models are non-stationary.
- The roots of the characteristic polyomial for AR(3) ae calculated as follows:

```
#we know that the best AR model has order 3.
polyroot(c(best_model_ar$coef[1], best_model_ar$coef[1], best_model_ar$coef[1]))
```

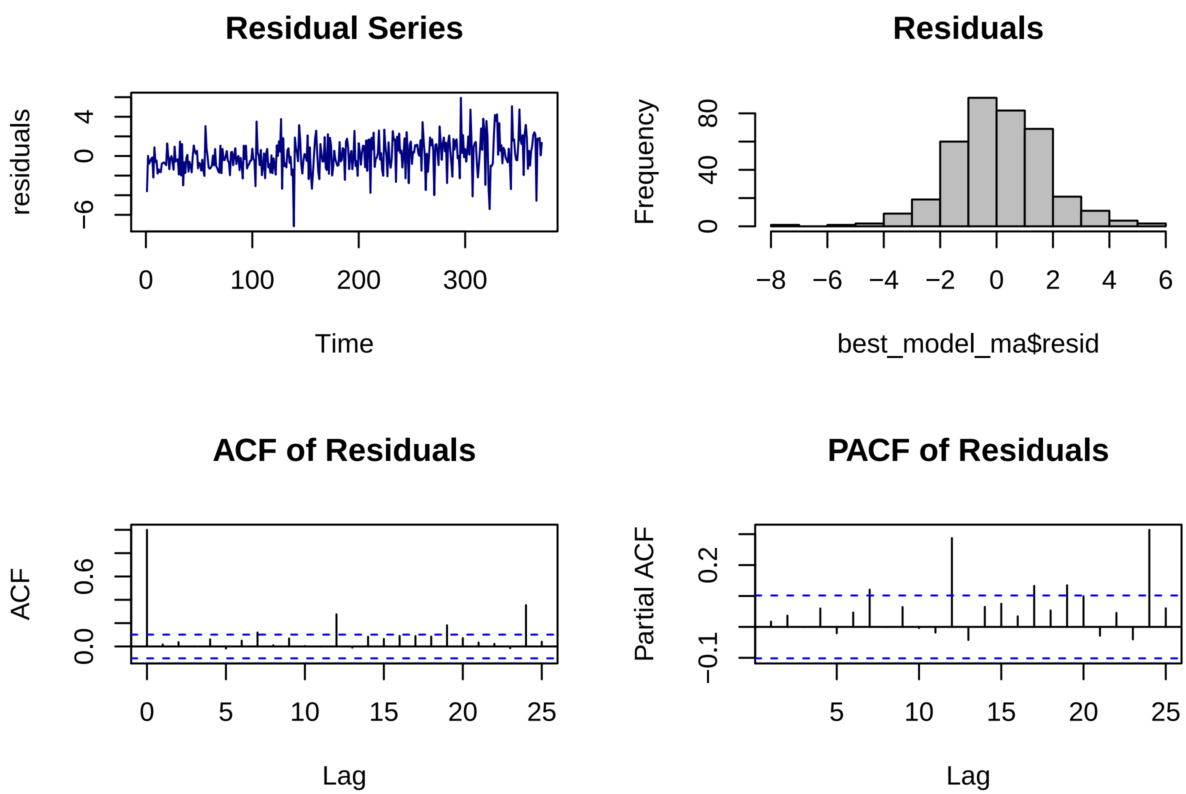
```
## [1] -0.5+0.8660254i -0.5-0.8660254i
```

- The roots of this equation are outside the unit circle in absolute value, hence the estimated process is stationary.

Evaluate the model performance (both in- and out-of-sample)

We evaluate the MA(30) model here.

```
#model diagnostics - residuals
par(mfrow=c(2,2))
plot.ts(best_model_ma$resid, main="Residual Series",
        ylab="residuals", col="navy")
hist(best_model_ma$resid, col="gray", main="Residuals")
acf(best_model_ma$resid, main="ACF of Residuals")
pacf(best_model_ma$resid, main="PACF of Residuals")
```




```r
#analysis of residuals
head(cbind(data.ts, fitted(best_model_ma), best_model_ma$resid),10)
```

```
##        data.ts fitted(best_model_ma) best_model_ma$resid
##  [1,]     40.6              44.20008         -3.60008195
##  [2,]     41.1              41.08549          0.01450618
##  [3,]     40.5              41.28816         -0.78815654
##  [4,]     40.1              40.67934         -0.57934354
##  [5,]     40.4              40.67074         -0.27073626
##  [6,]     41.2              41.34691         -0.14691176
##  [7,]     39.3              41.50122         -2.20122276
##  [8,]     41.6              40.71582          0.88418451
##  [9,]     42.3              42.92817         -0.62817446
## [10,]     43.2              43.66446         -0.46446268
```

```r
df<-data.frame(cbind(data.ts, fitted(best_model_ma), best_model_ma$resid))
library(stargazer)
```

```
## Warning: package 'stargazer' was built under R version 3.2.3
```

```
##
## Please cite as:
##
##  Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

```r
stargazer(df, type="text")
```

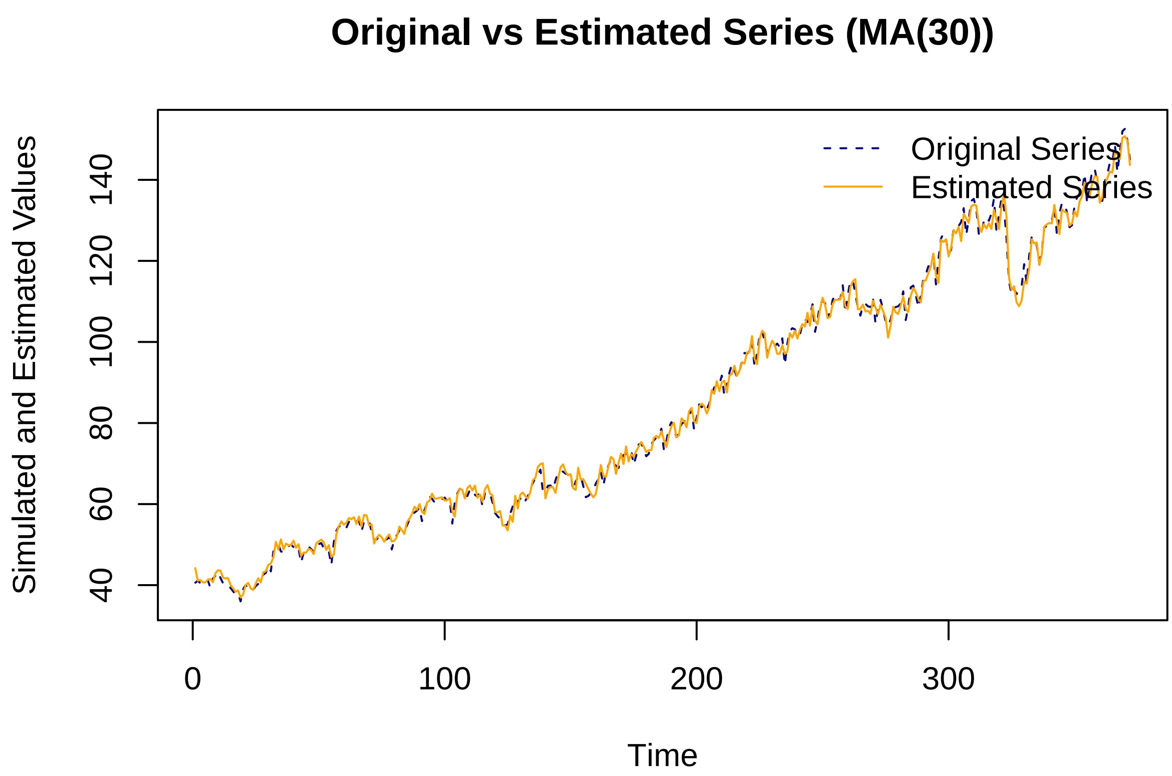
```
##
## ==========================================================
## Statistic              N   Mean  St. Dev.  Min     Max
## ----------------------------------------------------------
## data.ts               372 84.826  31.950  36.000 152.600
## fitted.best_model_ma. 372 84.781  31.455  36.941 150.755
## best_model_ma.resid   372 0.045   1.696   -7.171  5.933
## ----------------------------------------------------------
```

```r
summary(best_model_ma$resid)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -7.17100 -0.98510  0.00719  0.04514  1.11800  5.93300
```

- The residuals resemble white noise and have no significant ACFs; some PACFs are significant but this could be due to sampling error.

```r
# Model Performance Evaluation Using In-Sample Fit
par(mfrow=c(1,1))
  plot(data.ts, col="navy",
       main="Original vs Estimated Series (MA(30))",
       ylab="Simulated and Estimated Values", lty=2)
  lines(fitted(best_model_ma),col="orange")
  leg.txt <- c("Original Series", "Estimated Series")
  legend("topright", legend=leg.txt, lty=c(2,1),
         col=c("navy","orange"), bty='n', cex=1)
```



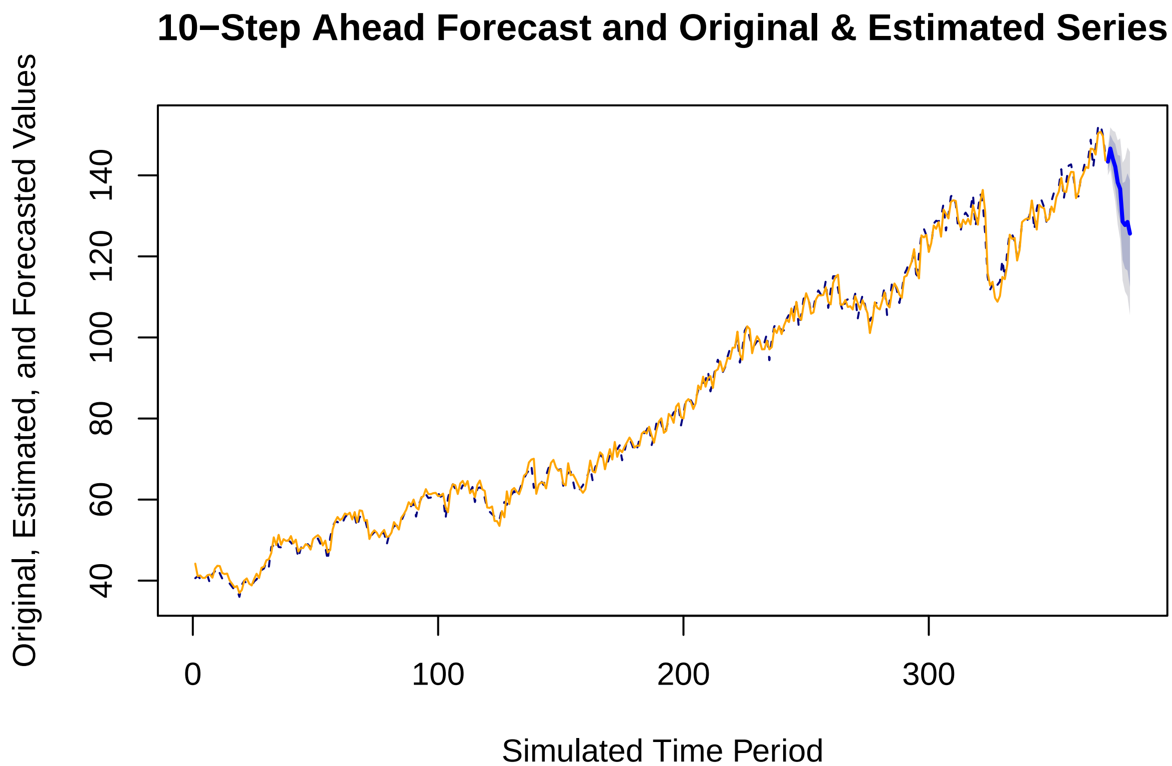
- As we can see, the fitted model follows the original pretty closely.

```
# Forecast - out-of-sample fit
best_model_ma.fcast <- forecast.Arima(best_model_ma, 10)
summary(best_model_ma.fcast)
```

```
##
## Forecast method: ARIMA(0,0,30) with non-zero mean
##
## Model Information:
##
## Call:
## arima(x = data.ts, order = c(0, 0, i))
##
## Coefficients:
##          ma1     ma2     ma3     ma4     ma5     ma6     ma7     ma8
##       1.2002  1.3557  1.4998  1.7458  2.0470  2.2008  2.2892  2.4452
## s.e.  0.0533  0.0878  0.1156  0.1391  0.1599  0.1801  0.1967  0.2071
##          ma9    ma10    ma11    ma12    ma13    ma14    ma15    ma16
##       2.4625  2.5999  2.6202  3.3582  3.3944  3.0427  2.9640  2.8514
## s.e.  0.2112  0.2109  0.2131  0.2142  0.2296  0.2365  0.2446  0.2487
##         ma17    ma18    ma19   ma20    ma21    ma22    ma23    ma24
##       3.1427  3.1124  2.5840  2.475  2.2495  2.1887  1.9011  1.8863
## s.e.  0.2496  0.2502  0.2414  0.239  0.2315  0.2202  0.2080  0.1892
##         ma25    ma26    ma27    ma28    ma29   ma30  intercept
```

```
##       1.6513  1.2144  0.9178  0.5830  0.5494  0.551    85.2568
## s.e.  0.1692  0.1463  0.1229  0.1015  0.0770  0.059     5.4287
##
## sigma^2 estimated as 2.87:  log likelihood = -744.65,  aic = 1553.3
##
## Error measures:
##                      ME    RMSE      MAE        MPE    MAPE      MASE
## Training set 0.04514392 1.69423 1.308539 -0.2110314 1.66694 0.6652069
##                    ACF1
## Training set 0.01738701
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 373       143.3714 141.1781 145.5647 140.0171 146.7258
## 374       146.6039 143.1794 150.0284 141.3665 151.8412
## 375       144.0798 139.5477 148.6119 137.1486 151.0110
## 376       142.1349 136.5281 147.7418 133.5600 150.7098
## 377       138.2422 131.4504 145.0341 127.8550 148.6295
## 378       136.6520 128.4950 144.8089 124.1770 149.1269
## 379       128.5916 119.0975 138.0857 114.0716 143.1116
## 380       127.7335 116.9917 138.4752 111.3054 144.1615
## 381       128.5073 116.4954 140.5192 110.1366 146.8779
## 382       125.6273 112.4550 138.7997 105.4819 145.7727
```

```r
plot(best_model_ma.fcast, main="10-Step Ahead Forecast and Original & Estimated Series",
    xlab="Simulated Time Period", ylab="Original, Estimated, and Forecasted Values",
    xlim=c(), lty=2, col="navy")
lines(fitted(best_model_ma),col="orange")
```



- From the plot above we can see that the model predits a downward trend with a pretty narrow confidence interval, indicating a strong confidence in the continued downward trend in the future.