Regression with ARMA Errors vs. ARIMAX

MX

June 2025

Introduction

This document demonstrates the simulation and estimation of regression models with ARMA errors. We compare different R packages and estimation methods.

The general model structure is:

$$y_t = \beta_0 + \beta_1 x_t + z_t$$

where z_t follows an ARMA process

Example 1: Regression with AR(1) Errors

Data Generation

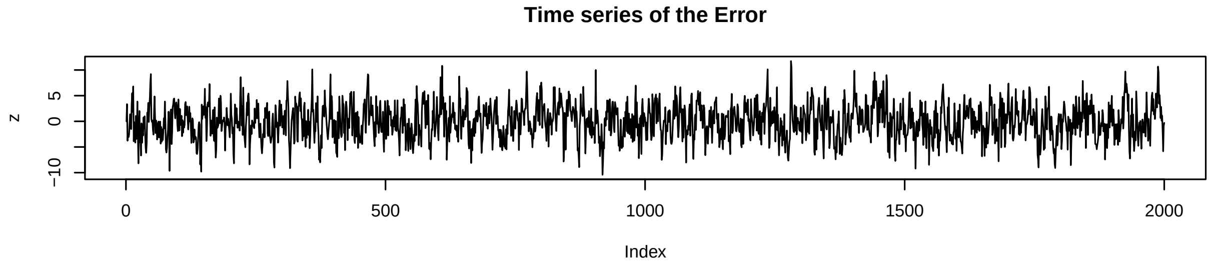
We start by simulating a regression model where the error term follows an AR(1) process:

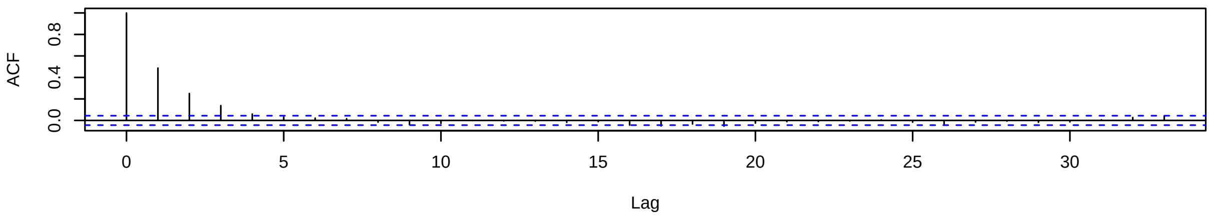
$$z_t = 0.5 z_{t-1} + e_t, \quad e_t \sim WN(0, 9)$$

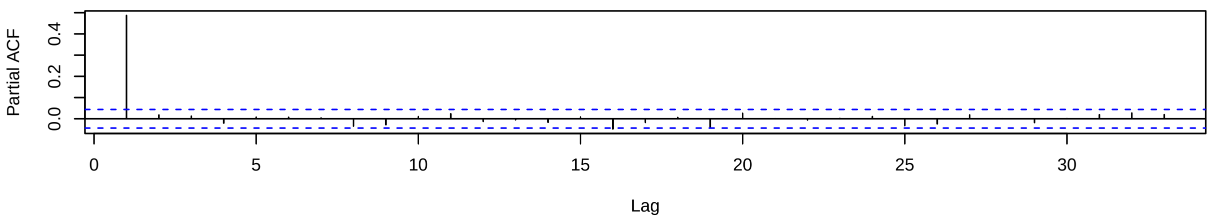
```r
set.seed(1989)
n = 2000

phi = 0.5
z = rep(0, n)
e = rnorm(n, mean=0, sd=3)
for (t in 2:n){
  z[t] = phi * z[t-1] + e[t]
}
```

Let's examine the properties of the AR(1) error process:

```r
par(mfcol=c(3,1))
plot(z, type = "l", main = "Time series of the Error")
acf(z, main = "")
pacf(z, main = "")  # spike at 1, looks correct
```





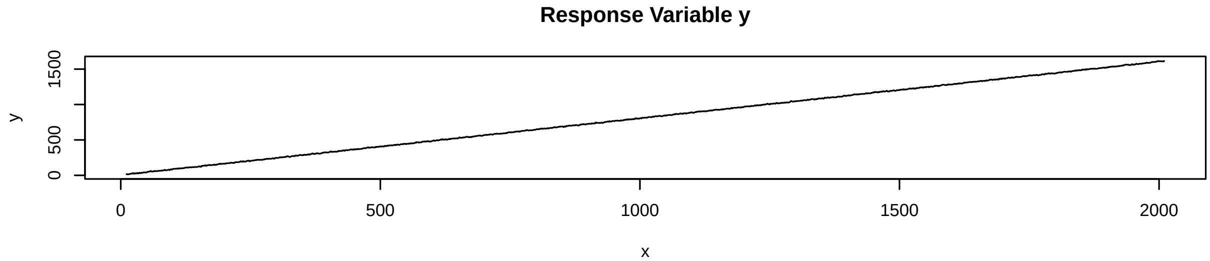


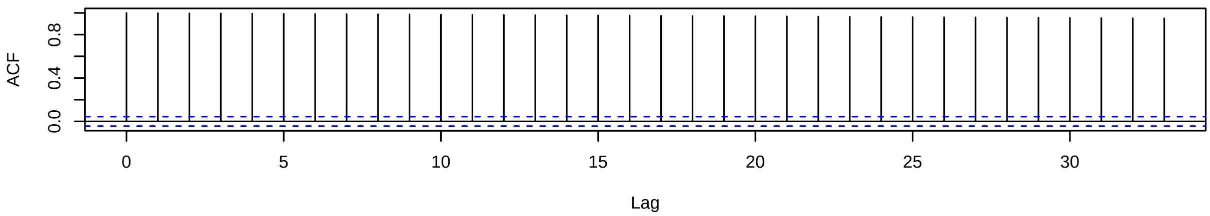
Now we generate the dependent variable:

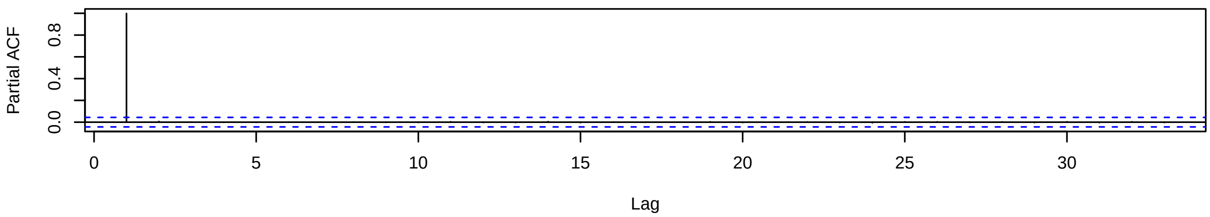
$$y_t = 6 + 0.8x_t + z_t$$

```
x = c(11:(10+n))
beta_0 = 6
beta_1 = 0.8
y = beta_0 + beta_1 * x + z
```

```
par(mfcol=c(3,1))
plot(y~x, type = "l", xlab = "x", main = "Response Variable y")
acf(y, main = "")
pacf(y, main = "")  # spike at 1
```







Model Estimation Comparison

Method 1: Simultaneous Estimation with `arima()` The preferred approach estimates all parameters simultaneously:

```
# arima() in base R
arima(y, order = c(1,0,0), xreg=x, include.mean = T)
```

```
##
## Call:
## arima(x = y, order = c(1, 0, 0), xreg = x, include.mean = T)
##
## Coefficients:
##          ar1  intercept       x
##       0.4866     5.8686  0.8001
## s.e.  0.0195     0.2608  0.0002
##
## sigma^2 estimated as 8.841:  log likelihood = -5017.45,  aic = 10042.89
```

True values: Intercept or $beta_0 = 6$. xreg or $beta_1 == 0.8$. ar1 or $\phi = 0.5$. $\sigma^2 = 9$

Estimates are close to the true values.

Note, include.mean controls whether the 'intercept' in the output or $beta0$ is estimated.

Method 2: Sequential Estimation (Not Recommended) Use lm() to fit y first, then use arima() to fit the residuals. We expect this sequential estimation to be inferior than the above, as the standard errors for the regression coefficients from lm() are incorrect because they are assumed to be independent when the errors are actually autocorrelated.

```
#
fit11 = lm(y ~ x)
summary(fit11)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3751  -2.4181   0.0469   2.2914  11.7473
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.8686088  0.1535164   38.23   <2e-16 ***
## x           0.8000957  0.0001319 6065.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.406 on 1998 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 3.679e+07 on 1 and 1998 DF,  p-value: < 2.2e-16
```

```
arima(resid(fit11), order=c(1,0,0), include.mean=T)
```

```
##
## Call:
## arima(x = resid(fit11), order = c(1, 0, 0), include.mean = T)
##
## Coefficients:
##          ar1  intercept
##       0.4866     0.0000
## s.e.  0.0195     0.1295
##
## sigma^2 estimated as 8.841:  log likelihood = -5017.45,  aic = 10040.89
```

The parameter estimates are similar, but the standard errors from the regression step are understated. Note, include.mean controls whether the 'intercept' in the output or *beta*0 is estimated.

Alternative R Packages

```


library(forecast)
Arima(y, order = c(1,0,0), xreg=x, include.mean=T)
```

Using the `forecast` Package

```
## Series: y
## Regression with ARIMA(1,0,0) errors
##
## Coefficients:
##          ar1  intercept    xreg
##       0.4866     5.8686  0.8001
## s.e.  0.0195     0.2608  0.0002
##
## sigma^2 = 8.855:  log likelihood = -5017.45
## AIC=10042.89   AICc=10042.91   BIC=10065.3
```

The `Arima()` function gives nearly identical results to base R's `arima()`. Note, include.mean controls whether the 'intercept' in the output or *beta*0 is estimated.

Let's see what `auto.arima()` selects:

```
fit21 = auto.arima(y, xreg=x,
                   seasonal = TRUE, stepwise = F, trace = TRUE,
                   allowdrift = T, allowmean = T,
                   lambda = NULL, approximation = F)
```

```
##
##  Regression with ARIMA(0,0,0) errors : 11677.65
##  Regression with ARIMA(0,0,0) errors : 10581.77
##  Regression with ARIMA(0,0,1) errors : 10832.96
##  Regression with ARIMA(0,0,1) errors : 10147.07
##  Regression with ARIMA(0,0,2) errors : 10585.35
##  Regression with ARIMA(0,0,2) errors : 10076.1
##  Regression with ARIMA(0,0,3) errors : 10454.38
##  Regression with ARIMA(0,0,3) errors : 10049.96
##  Regression with ARIMA(0,0,4) errors : 10395.8
##  Regression with ARIMA(0,0,4) errors : 10047.91
##  Regression with ARIMA(0,0,5) errors : 10367.98
##  Regression with ARIMA(0,0,5) errors : 10049.47
##  Regression with ARIMA(1,0,0) errors : 10314.44
##  Regression with ARIMA(1,0,0) errors : 10042.91
##  Regression with ARIMA(1,0,1) errors : 10253.38
##  Regression with ARIMA(1,0,1) errors : 10044.27
##  Regression with ARIMA(1,0,2) errors : Inf
##  Regression with ARIMA(1,0,2) errors : 10046.09
##  Regression with ARIMA(1,0,3) errors : Inf
##  Regression with ARIMA(1,0,3) errors : 10047.13
##  Regression with ARIMA(1,0,4) errors : Inf
##  Regression with ARIMA(1,0,4) errors : 10049
##  Regression with ARIMA(2,0,0) errors : 10275.58
##  Regression with ARIMA(2,0,0) errors : 10044.3
##  Regression with ARIMA(2,0,1) errors : Inf
##  Regression with ARIMA(2,0,1) errors : Inf
##  Regression with ARIMA(2,0,2) errors : Inf
##  Regression with ARIMA(2,0,2) errors : 10047.21
##  Regression with ARIMA(2,0,3) errors : Inf
##  Regression with ARIMA(2,0,3) errors : Inf
##  Regression with ARIMA(3,0,0) errors : 10248.5
##  Regression with ARIMA(3,0,0) errors : 10046.01
```

```
##  Regression with ARIMA(3,0,1) errors : Inf
##  Regression with ARIMA(3,0,1) errors : 10047.22
##  Regression with ARIMA(3,0,2) errors : Inf
##  Regression with ARIMA(3,0,2) errors : Inf
##  Regression with ARIMA(4,0,0) errors : 10238.44
##  Regression with ARIMA(4,0,0) errors : 10047.17
##  Regression with ARIMA(4,0,1) errors : Inf
##  Regression with ARIMA(4,0,1) errors : 10049.11
##  Regression with ARIMA(5,0,0) errors : 10221.17
##  Regression with ARIMA(5,0,0) errors : 10049.07
##
##
##
##  Best model: Regression with ARIMA(1,0,0) errors
```

```
summary(fit21)
```

```
## Series: y
## Regression with ARIMA(1,0,0) errors
##
## Coefficients:
##          ar1  intercept    xreg
##       0.4866     5.8686  0.8001
## s.e.  0.0195     0.2608  0.0002
##
## sigma^2 = 8.855:  log likelihood = -5017.45
## AIC=10042.89   AICc=10042.91   BIC=10065.3
##
## Training set error measures:
##                         ME     RMSE      MAE        MPE      MAPE      MASE
## Training set -8.833442e-05 2.973434 2.382752 -0.0702387 0.7295079 0.8419134
##                      ACF1
## Training set -0.008366473
```

`auto.arima()` correctly selects ARIMA(1,0,0) based on AICc, but it could be wrong.

Note, set stepwise = F for exhaustive search, trace=TRUE for detailed reporting, allowmean=T for including the 'intercept' in the output or β_0, lambda=NULL for no transformation, approximation=F for no approximation.

When the model is not differenced, `allowdrift` parameter is ignored. Suppose the error z_t has an ARIMA(1,1,1) structure:

$$y_t = 6 + 0.8x_t + z_t$$

$$(1-\phi B)(1-B)z_t = (1+\theta B)e_t$$

To estimate this regression model with an ARIMA error, the model needs to be differenced:

$$y_t = 6 + 0.8x_t + z_t, \text{ and } y_{t-1} = 6 + 0.8x_{t-1} + z_{t-1}$$

So

$$\Delta y_t = 0.8\Delta x_t + \Delta z_t$$

where Δz_t follows ARMA(1,1):

$$\Delta z_t = \phi\Delta z_{t-1} + e_t + \theta e_{t-1}$$

So the correct model to be estimated is

$$\Delta y_t = 0.8\Delta x_t + \frac{1+\theta B}{1-\phi B} e_t$$

In a differenced model, `allowdrift=T` means:

$$\Delta z_t = \delta + \phi \Delta z_{t-1} + e_t + \theta e_{t-1}$$

(check later)

```
library(TSA)
arimax(y, order=c(1,0,0), xreg = x)
```

Using the `TSA` Package

```
##
## Call:
## arimax(x = y, order = c(1, 0, 0), xreg = x)
##
## Coefficients:
##          ar1  intercept    xreg
##       0.4866     5.8686  0.8001
## s.e.  0.0195     0.2608  0.0002
##
## sigma^2 estimated as 8.841:  log likelihood = -5017.45,  aic = 10040.89
```

The `arimax()` function produces results identical to base R's `arima()`.

Note: `arimax()` with `xtransf` and `transfer` arguments is used for transfer function models with dynamic regressors, not simple regression-style covariates.

Example 2: Regression with ARMA(1,1) Errors

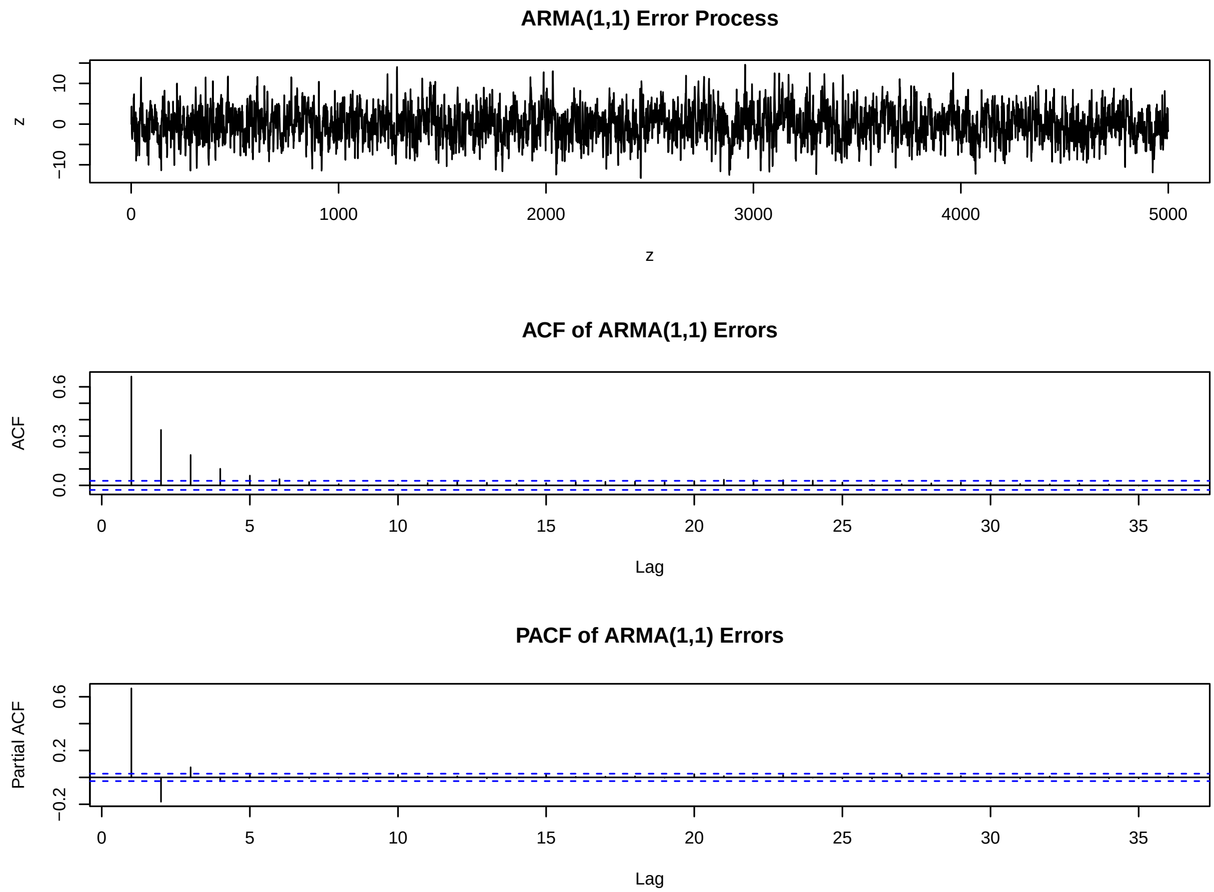
Data Generation

Now let's consider a more complex error structure with both AR and MA components:

$$z_t = 0.5 z_{t-1} + e_t + 0.3 e_{t-1}, \quad e_t \sim N(0, 9)$$

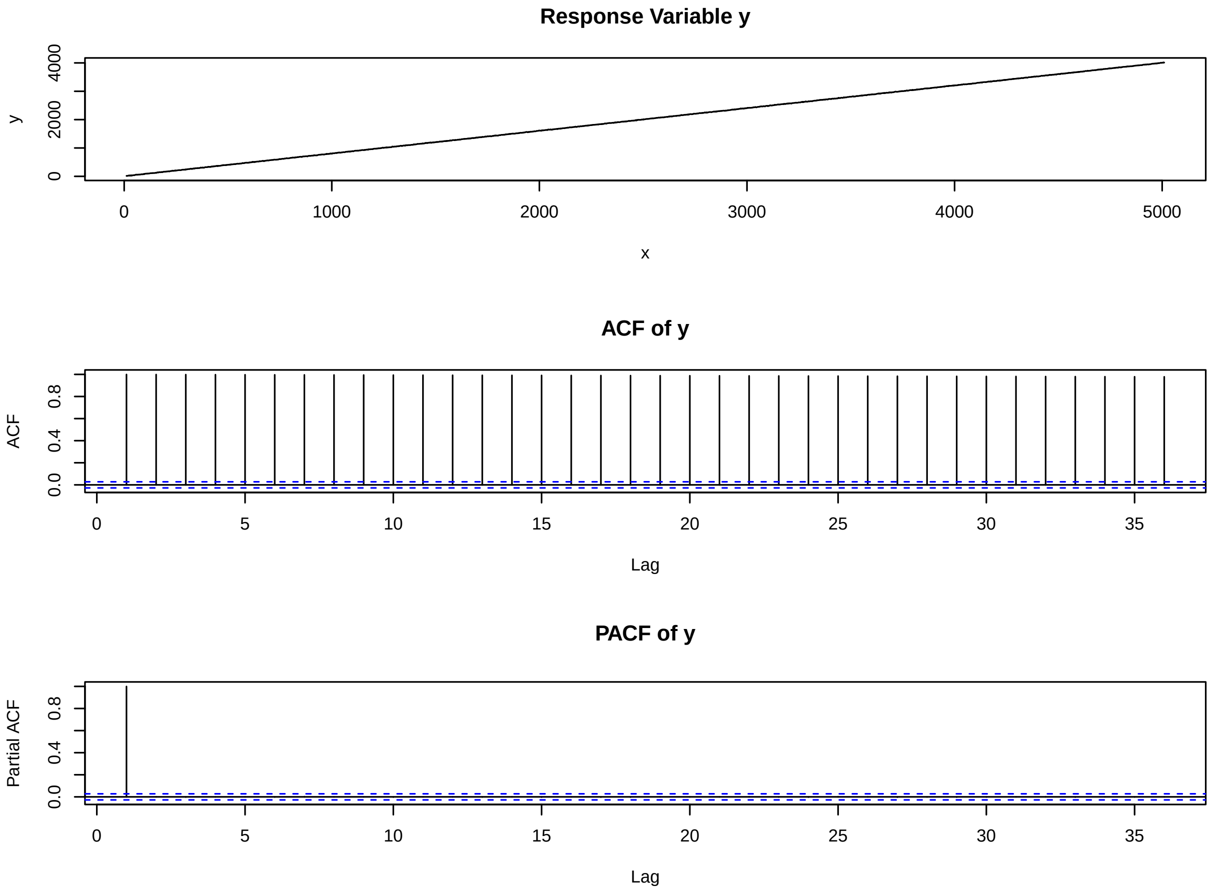
```
rm(list = ls())
set.seed(1989)
n = 5000

phi = 0.5
theta = 0.3
z = rep(0, n)
e = rnorm(n, mean=0, sd=3)
for (t in 2:n){
  z[t] = phi * z[t-1] + e[t] + theta * e[t-1]
}
```

```r
par(mfcol=c(3,1))
plot(z, type = "l", xlab = "z", main = "ARMA(1,1) Error Process")
acf(z, main = "ACF of ARMA(1,1) Errors")
pacf(z, main = "PACF of ARMA(1,1) Errors")
```




```r
x = c(11:(10+n))
beta_0 = 6
beta_1 = 0.8
y = beta_0 + beta_1 * x + z
```

```r
par(mfcol=c(3,1))
plot(y~x, type = "l", xlab = "x", main = "Response Variable y")
acf(y, main = "ACF of y")
pacf(y, main = "PACF of y")
```



Model Estimation for ARMA(1,1) Errors

```r
arima(y, order = c(1,0,1), xreg=x)
```

Base R `arima()`

```
##
## Call:
## arima(x = y, order = c(1, 0, 1), xreg = x)
##
## Coefficients:
##          ar1     ma1  intercept   xreg
##       0.4957  0.3070     6.0258  8e-01
## s.e.  0.0180  0.0201     0.2186  1e-04
##
## sigma^2 estimated as 8.843:  log likelihood = -12544.11,  aic = 25096.22
```

The estimates are reasonably close to true values: - Intercept: (true: 6) - x coefficient:(true: 0.8) - AR(1) coefficient: (true: 0.5) - MA(1) coefficient: (true: 0.3)

```r
library(forecast)
Arima(y, order = c(1,0,1), xreg=x, include.mean=T)
```

forecast Package

```
## Series: y
## Regression with ARIMA(1,0,1) errors
##
## Coefficients:
##          ar1     ma1  intercept   xreg
##       0.4957  0.3070     6.0258  8e-01
## s.e.  0.0180  0.0201     0.2186  1e-04
##
## sigma^2 = 8.85:  log likelihood = -12544.11
## AIC=25098.22   AICc=25098.23   BIC=25130.8
```

```r
fit21 = auto.arima(y, xreg=x,
                   seasonal = TRUE, stepwise = F, trace = TRUE,
                   allowdrift = T, allowmean = T,
                   lambda = NULL, approximation = F)
```

```
##
##  Regression with ARIMA(0,0,0) errors : 30371.37
##  Regression with ARIMA(0,0,0) errors : 28182.3
##  Regression with ARIMA(0,0,1) errors : 27069.78
##  Regression with ARIMA(0,0,1) errors : 25659.55
##  Regression with ARIMA(0,0,2) errors : 26215.83
##  Regression with ARIMA(0,0,2) errors : 25241.37
##  Regression with ARIMA(0,0,3) errors : 25896.78
##  Regression with ARIMA(0,0,3) errors : 25135.86
##  Regression with ARIMA(0,0,4) errors : 25749.8
##  Regression with ARIMA(0,0,4) errors : 25108.75
##  Regression with ARIMA(0,0,5) errors : 25685.54
##  Regression with ARIMA(0,0,5) errors : 25105.94
##  Regression with ARIMA(1,0,0) errors : 25640.63
##  Regression with ARIMA(1,0,0) errors : 25293.76
##  Regression with ARIMA(1,0,1) errors : 25561.9
##  Regression with ARIMA(1,0,1) errors : 25098.23
##  Regression with ARIMA(1,0,2) errors : 25448.93
##  Regression with ARIMA(1,0,2) errors : 25097.04
##  Regression with ARIMA(1,0,3) errors : Inf
##  Regression with ARIMA(1,0,3) errors : 25099.02
##  Regression with ARIMA(1,0,4) errors : Inf
##  Regression with ARIMA(1,0,4) errors : 25100.97
##  Regression with ARIMA(2,0,0) errors : 25590.27
##  Regression with ARIMA(2,0,0) errors : 25128.83
##  Regression with ARIMA(2,0,1) errors : 25523.66
##  Regression with ARIMA(2,0,1) errors : 25097.02
##  Regression with ARIMA(2,0,2) errors : Inf
##  Regression with ARIMA(2,0,2) errors : 25099.02
##  Regression with ARIMA(2,0,3) errors : Inf
```

```
##  Regression with ARIMA(2,0,3) errors : 25101.03
##  Regression with ARIMA(3,0,0) errors : 25467.44
##  Regression with ARIMA(3,0,0) errors : 25102.36
##  Regression with ARIMA(3,0,1) errors : Inf
##  Regression with ARIMA(3,0,1) errors : 25099.01
##  Regression with ARIMA(3,0,2) errors : Inf
##  Regression with ARIMA(3,0,2) errors : Inf
##  Regression with ARIMA(4,0,0) errors : 25459.55
##  Regression with ARIMA(4,0,0) errors : 25100.41
##  Regression with ARIMA(4,0,1) errors : Inf
##  Regression with ARIMA(4,0,1) errors : 25101.07
##  Regression with ARIMA(5,0,0) errors : 25423.56
##  Regression with ARIMA(5,0,0) errors : 25100.64
##
##
##
##  Best model: Regression with ARIMA(2,0,1) errors
```

```r
summary(fit21)
```

```
## Series: y
## Regression with ARIMA(2,0,1) errors
##
## Coefficients:
##          ar1     ar2     ma1  intercept   xreg
##       0.3866  0.0810  0.4125     6.0235  8e-01
## s.e.  0.0610  0.0444  0.0583     0.2236  1e-04
##
## sigma^2 = 8.846:  log likelihood = -12542.5
## AIC=25097   AICc=25097.02   BIC=25136.1
##
## Training set error measures:
##                        ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.0001560642 2.972772 2.373215 -0.03130488 0.346894 0.8713108
##                      ACF1
## Training set 1.306728e-05
```

This time `auto.arima()` fails to identify the ARIMA(1,0,1) structure.

```r
library(TSA)
arimax(y, order=c(1,0,1), xreg = x)
```

`TSA` Package

```
##
## Call:
## arimax(x = y, order = c(1, 0, 1), xreg = x)
##
## Coefficients:
##          ar1     ma1  intercept   xreg
```

```
##       0.4957  0.3070     6.0258  8e-01
## s.e.  0.0180  0.0201     0.2186  1e-04
##
## sigma^2 estimated as 8.843:  log likelihood = -12544.11,  aic = 25096.22
```

Again, `arimax()` produces identical results to base R.

Key Takeaways

1. **Simultaneous estimation** (using `arima()`, `Arima()`, or `arimax()`) is statistically superior to sequential estimation because it properly accounts for the correlation structure in the data.
2. **Sequential estimation** (fitting regression first, then AR model to residuals) produces biased standard errors and is less efficient.
3. **Different R packages** (`stats`, `forecast`, `TSA`) generally produce very similar results for the same model specification.
4. **Automatic model selection** with `auto.arima()` is helpful but obviously cannot guarantee finding the true pattern.