10. Dynamic regression

OTexts.com/fpp/9/1/
1. Regression with ARIMA errors
2. Stochastic and deterministic trends
3. Periodic seasonality
4. Dynamic regression models
Regression models

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + e_t, \]

- \( y_t \) modeled as function of \( k \) explanatory variables \( x_{1,t}, \ldots, x_{k,t} \).
- Previously, we assumed that \( e_t \) was WN.
- Now we want to allow \( e_t \) to be autocorrelated.

Example: ARIMA(1,1,1) errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]

\[ (1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t, \]

where \( e_t \) is white noise.
Regression models

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + e_t, \]

- \( y_t \) modeled as function of \( k \) explanatory variables \( x_{1,t}, \ldots, x_{k,t} \).
- Previously, we assumed that \( e_t \) was WN.
- Now we want to allow \( e_t \) to be autocorrelated.

Example: ARIMA(1,1,1) errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]
\[ (1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t, \]

where \( e_t \) is white noise.
Regression models

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + e_t, \]

- \( y_t \) modeled as function of \( k \) explanatory variables \( x_{1,t}, \ldots, x_{k,t} \).
- Previously, we assumed that \( e_t \) was WN.
- Now we want to allow \( e_t \) to be autocorrelated.

Example: ARIMA(1,1,1) errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]

\[ (1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t, \]

where \( e_t \) is white noise.
Regression models

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + e_t, \]

- \( y_t \) modeled as function of \( k \) explanatory variables \( x_{1,t}, \ldots, x_{k,t} \).
- Previously, we assumed that \( e_t \) was WN.
- Now we want to allow \( e_t \) to be autocorrelated.

Example: ARIMA(1,1,1) errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]

\[ (1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t, \]

where \( e_t \) is white noise.
Regression models

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + e_t, \]

- \( y_t \) modeled as function of \( k \) explanatory variables \( x_{1,t}, \ldots, x_{k,t} \).
- Previously, we assumed that \( e_t \) was WN.
- Now we want to allow \( e_t \) to be autocorrelated.

**Example: ARIMA(1,1,1) errors**

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]
\[ (1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t, \]

where \( e_t \) is white noise.
Example: \( N_t = \text{ARIMA}(1,1,1) \)

\[
y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \\
(1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t,
\]

- Be careful in distinguishing \( n_t \) from \( e_t \).
- Only the errors \( n_t \) are assumed to be white noise.
- In ordinary regression, \( n_t \) is assumed to be white noise and so \( n_t = e_t \).
Residuals and errors

Example: \( N_t = \text{ARIMA}(1,1,1) \)

\[
y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t,
\]

\[
(1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t,
\]

- Be careful in distinguishing \( n_t \) from \( e_t \).
- Only the errors \( n_t \) are assumed to be white noise.
- In ordinary regression, \( n_t \) is assumed to be white noise and so \( n_t = e_t \).
Residuals and errors

Example: \( N_t = ARIMA(1,1,1) \)

\[
y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t,
\]
\[
(1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t,
\]

- Be careful in distinguishing \( n_t \) from \( e_t \).
- Only the errors \( n_t \) are assumed to be white noise.
- In ordinary regression, \( n_t \) is assumed to be white noise and so \( n_t = e_t \).
Residuals and errors

Example: $N_t = \text{ARIMA}(1,1,1)$

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]
\[ (1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t, \]

- Be careful in distinguishing $n_t$ from $e_t$.
- Only the errors $n_t$ are assumed to be white noise.
- In ordinary regression, $n_t$ is assumed to be white noise and so $n_t = e_t$. 

Forecasting: Principles and Practice

Regression with ARIMA errors 4
Estimation

If we minimize $\sum n_t^2$ (by using ordinary regression):

1. Estimated coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k$ are no longer optimal as some information ignored;

2. Statistical tests associated with the model (e.g., t-tests on the coefficients) are incorrect.

3. $p$-values for coefficients usually too small (“spurious regression”).

4. AIC of fitted models misleading.
Estimation

If we minimize $\sum n_t^2$ (by using ordinary regression):

1. Estimated coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k$ are no longer optimal as some information ignored;

2. Statistical tests associated with the model (e.g., t-tests on the coefficients) are incorrect.

3. $p$-values for coefficients usually too small ("spurious regression").

4. AIC of fitted models misleading.
Estimation

If we minimize $\sum n_t^2$ (by using ordinary regression):

1. Estimated coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k$ are no longer optimal as some information ignored;

2. Statistical tests associated with the model (e.g., t-tests on the coefficients) are incorrect.

3. $p$-values for coefficients usually too small (“spurious regression”).

4. AIC of fitted models misleading.

Minimizing $\sum e_t^2$ avoids these problems.
Estimation

If we minimize $\sum n_t^2$ (by using ordinary regression):

1. Estimated coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k$ are no longer optimal as some information ignored;
2. Statistical tests associated with the model (e.g., t-tests on the coefficients) are incorrect.
3. $p$-values for coefficients usually too small (“spurious regression”).
4. AIC of fitted models misleading.

Minimizing $\sum e_t^2$ avoids these problems.
Maximizing likelihood is similar to minimizing $\sum e_t^2$. 
Estimation

If we minimize $\sum n_t^2$ (by using ordinary regression):

1. Estimated coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k$ are no longer optimal as some information ignored;
2. Statistical tests associated with the model (e.g., t-tests on the coefficients) are incorrect.
3. $p$-values for coefficients usually too small (“spurious regression”).
4. AIC of fitted models misleading.

- Minimizing $\sum e_t^2$ avoids these problems.
- Maximizing likelihood is similar to minimizing $\sum e_t^2$. 

Forecasting: Principles and Practice

Regression with ARIMA errors
If we minimize $\sum n_t^2$ (by using ordinary regression):

1. Estimated coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k$ are no longer optimal as some information ignored;
2. Statistical tests associated with the model (e.g., t-tests on the coefficients) are incorrect.
3. $p$-values for coefficients usually too small (“spurious regression”).
4. AIC of fitted models misleading.

- Minimizing $\sum e_t^2$ avoids these problems.
- Maximizing likelihood is similar to minimizing $\sum e_t^2$. 

Forecasting: Principles and Practice
Estimation

If we minimize $\sum n_t^2$ (by using ordinary regression):

1. Estimated coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_k$ are no longer optimal as some information ignored;
2. Statistical tests associated with the model (e.g., t-tests on the coefficients) are incorrect.
3. $p$-values for coefficients usually too small (“spurious regression”).
4. AIC of fitted models misleading.

Minimizing $\sum e_t^2$ avoids these problems.
Maximizing likelihood is similar to minimizing $\sum e_t^2$. 
Stationarity

Regression with ARMA errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]

where \( n_t \) is an ARMA process.

- All variables in the model must be stationary.
- If we estimate the model while any of these are non-stationary, the estimated coefficients can be incorrect.
- Difference variables until all stationary.
- If necessary, apply same differencing to all variables.
Stationarity

Regression with ARMA errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]

where \( n_t \) is an ARMA process.

- All variables in the model must be stationary.
- If we estimate the model while any of these are non-stationary, the estimated coefficients can be incorrect.
- Difference variables until all stationary.
- If necessary, apply same differencing to all variables.
Regression with ARMA errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]

where \( n_t \) is an ARMA process.

- All variables in the model must be stationary.
- If we estimate the model while any of these are non-stationary, the estimated coefficients can be incorrect.
- **Difference variables until all stationary.**
- If necessary, apply same differencing to all variables.
Stationarity

Regression with ARMA errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]

where \( n_t \) is an ARMA process.

- All variables in the model must be stationary.
- If we estimate the model while any of these are non-stationary, the estimated coefficients can be incorrect.
- Difference variables until all stationary.
- If necessary, apply same differencing to all variables.
Stationarity

Model with ARIMA(1,1,1) errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]
\[ (1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t, \]

Equivalent to model with ARIMA(1,0,1) errors

\[ y'_t = \beta_1 x'_{1,t} + \cdots + \beta_k x'_{k,t} + n'_t, \]
\[ (1 - \phi_1 B)n'_t = (1 + \theta_1 B)e_t, \]

where \( y'_t = y_t - y_{t-1} \), \( x'_{t,i} = x_{t,i} - x_{t-1,i} \) and \( n'_t = n_t - n_{t-1} \).
Stationarity

Model with ARIMA(1,1,1) errors

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t, \]
\[ (1 - \phi_1 B)(1 - B)n_t = (1 + \theta_1 B)e_t, \]

Equivalent to model with ARIMA(1,0,1) errors

\[ y'_t = \beta_1 x'_{1,t} + \cdots + \beta_k x'_{k,t} + n'_t, \]
\[ (1 - \phi_1 B)n'_t = (1 + \theta_1 B)e_t, \]

where \( y'_t = y_t - y_{t-1} \), \( x'_{t,i} = x_{t,i} - x_{t-1,i} \) and \( n'_t = n_t - n_{t-1} \).
Regression with ARIMA errors

Any regression with an ARIMA error can be rewritten as a regression with an ARMA error by differencing all variables with the same differencing operator as in the ARIMA model.

### Original data

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t \]

where \( \phi(B)(1 - B)^d N_t = \theta(B)e_t \)

### After differencing all variables

\[ y'_t = \beta_1 x'_{1,t} + \cdots + \beta_k x'_{k,t} + n'_t. \]

where \( \phi(B)N_t = \theta(B)e_t \)

and \( y'_t = (1 - B)^d y_t \)
Regression with ARIMA errors

Any regression with an ARIMA error can be rewritten as a regression with an ARMA error by differencing all variables with the same differencing operator as in the ARIMA model.

**Original data**

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t \]

where \( \phi(B)(1 - B)^d N_t = \theta(B)e_t \)

**After differencing all variables**

\[ y'_t = \beta_1 x'_{1,t} + \cdots + \beta_k x'_{k,t} + n'_t \]

where \( \phi(B)N_t = \theta(B)e_t \)

and \( y'_t = (1 - B)^d y_t \)
Regression with ARIMA errors

Any regression with an ARIMA error can be rewritten as a regression with an ARMA error by differencing all variables with the same differencing operator as in the ARIMA model.

**Original data**

\[ y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + n_t \]

where \( \phi(B)(1 - B)^d N_t = \theta(B)e_t \)

**After differencing all variables**

\[ y'_t = \beta_1 x'_{1,t} + \cdots + \beta_k x'_{k,t} + n'_t. \]

where \( \phi(B)N_t = \theta(B)e_t \)

and \( y'_t = (1 - B)^d y_t \)
To determine ARIMA error structure, first need to calculate $n_t$.

We can’t get $n_t$ without knowing $\beta_0, \ldots, \beta_k$.

To estimate these, we need to specify ARIMA error structure.

**Solution:** Begin with a proxy model for the ARIMA errors.

Estimate model, determine better error structure, and re-estimate.
To determine ARIMA error structure, first need to calculate $n_t$.

We can’t get $n_t$ without knowing $\beta_0, \ldots, \beta_k$.

To estimate these, we need to specify ARIMA error structure.

Solution: Begin with a proxy model for the ARIMA errors.

Assume AR(2) model for non-seasonal data; Assume ARIMA(2,0,0)(1,0,0) model for seasonal data.

Estimate model, determine better error structure, and re-estimate.
To determine ARIMA error structure, first need to calculate $n_t$.

We can’t get $n_t$ without knowing $\beta_0, \ldots, \beta_k$.

To estimate these, we need to specify ARIMA error structure.

Solution: Begin with a proxy model for the ARIMA errors.

- Assume AR(2) model for non-seasonal data;
- Assume ARIMA$(2,0,0)(1,0,0)_m$ model for seasonal data.

Estimate model, determine better error structure, and re-estimate.
Model selection

To determine ARIMA error structure, first need to calculate $n_t$.

We can’t get $n_t$ without knowing $\beta_0, \ldots, \beta_k$.

To estimate these, we need to specify ARIMA error structure.

Solution: Begin with a proxy model for the ARIMA errors.

- Assume AR(2) model for non-seasonal data;
- Assume ARIMA(2,0,0)(1,0,0)$_m$ model for seasonal data.

Estimate model, determine better error structure, and re-estimate.
Model selection

To determine ARIMA error structure, first need to calculate $n_t$.

We can’t get $n_t$ without knowing $\beta_0, \ldots, \beta_k$.

To estimate these, we need to specify ARIMA error structure.

**Solution:** Begin with a proxy model for the ARIMA errors.

- Assume AR(2) model for non-seasonal data;
- Assume ARIMA(2,0,0)(1,0,0)$_m$ model for seasonal data.

Estimate model, determine better error structure, and re-estimate.
Model selection

To determine ARIMA error structure, first need to calculate $n_t$. 

We can’t get $n_t$ without knowing $\beta_0, \ldots, \beta_k$.

To estimate these, we need to specify ARIMA error structure.

Solution: Begin with a proxy model for the ARIMA errors.

- Assume AR(2) model for non-seasonal data;
- Assume ARIMA(2,0,0)(1,0,0)$_m$ model for seasonal data.

Estimate model, determine better error structure, and re-estimate.
Model selection

To determine ARIMA error structure, first need to calculate $n_t$.

We can’t get $n_t$ without knowing $\beta_0, \ldots, \beta_k$.

To estimate these, we need to specify ARIMA error structure.

Solution: Begin with a proxy model for the ARIMA errors.

- Assume AR(2) model for non-seasonal data;
- Assume ARIMA(2,0,0)(1,0,0)$_m$ model for seasonal data.

Estimate model, determine better error structure, and re-estimate.
Model selection

To determine ARIMA error structure, first need to calculate \( n_t \).

We can’t get \( n_t \) without knowing \( \beta_0, \ldots, \beta_k \).

To estimate these, we need to specify ARIMA error structure.

**Solution:** Begin with a proxy model for the ARIMA errors.

- Assume AR(2) model for non-seasonal data;
- Assume ARIMA(2,0,0)(1,0,0)\(_m\) model for seasonal data.

Estimate model, determine better error structure, and re-estimate.
**Model selection**

1. Check that all variables are stationary. If not, apply differencing. Where appropriate, use the same differencing for all variables to preserve interpretability.

2. Fit regression model with AR(2) errors for non-seasonal data or ARIMA(2,0,0)(1,0,0)_m errors for seasonal data.

3. Calculate errors (n_t) from fitted regression model and identify ARMA model for them.

4. Re-fit entire model using new ARMA model for errors.

5. Check that e_t series looks like white noise.
Model selection

1. Check that all variables are stationary. If not, apply differencing. Where appropriate, use the same differencing for all variables to preserve interpretability.

2. Fit regression model with AR(2) errors for non-seasonal data or ARIMA(2,0,0)(1,0,0)$_m$ errors for seasonal data.

3. Calculate errors ($n_t$) from fitted regression model and identify ARMA model for them.

4. Re-fit entire model using new ARMA model for errors.

5. Check that $e_t$ series looks like white noise.
Model selection

1. Check that all variables are stationary. If not, apply differencing. Where appropriate, use the same differencing for all variables to preserve interpretability.

2. Fit regression model with AR(2) errors for non-seasonal data or ARIMA(2,0,0)(1,0,0)$^m$ errors for seasonal data.

3. Calculate errors ($n_t$) from fitted regression model and identify ARMA model for them.

4. Re-fit entire model using new ARMA model for errors.

5. Check that $e_t$ series looks like white noise.
Model selection

1. Check that all variables are stationary. If not, apply differencing. Where appropriate, use the same differencing for all variables to preserve interpretability.

2. Fit regression model with AR(2) errors for non-seasonal data or ARIMA(2,0,0)(1,0,0)_m errors for seasonal data.

3. Calculate errors (n_t) from fitted regression model and identify ARMA model for them.

4. Re-fit entire model using new ARMA model for errors.

5. Check that e_t series looks like white noise.
1 Check that all variables are stationary. If not, apply differencing. Where appropriate, use the same differencing for all variables to preserve interpretability.

2 Fit regression model with AR(2) errors for non-seasonal data or ARIMA(2,0,0)(1,0,0)_m errors for seasonal data.

3 Calculate errors (n_t) from fitted regression model and identify ARMA model for them.

4 Re-fit entire model using new ARMA model for errors.

5 Check that e_t series looks like white noise.
Model selection

1. Check that all variables are stationary. If not, apply differencing. Where appropriate, use the same differencing for all variables to preserve interpretability.

2. Fit regression model with AR(2) errors for non-seasonal data or ARIMA(2,0,0)(1,0,0) errors for seasonal data.

3. Calculate errors ($e_t$) from fitted regression model and identify ARMA model for them.

4. Re-fit entire model using new ARMA model for errors.

5. Check that $e_t$ series looks like white noise.

Selecting predictors

- AIC can be calculated for final model.
- Repeat procedure for all subsets of predictors to be considered, and select model with lowest AIC value.
Model selection

1. Check that all variables are stationary. If not, apply differencing. Where appropriate, use the same differencing for all variables to preserve interpretability.

2. Fit regression model with AR(2) errors for non-seasonal data or ARIMA(2,0,0)(1,0,0) errors for seasonal data.

3. Calculate errors ($n_t$) from fitted regression model and identify ARMA model for them.

4. Re-fit entire model using new ARMA model for errors.

5. Check that $e_t$ series looks like white noise.

Selecting predictors

- AIC can be calculated for final model.

- Repeat procedure for all subsets of predictors to be considered, and select model with lowest AIC value.
Model selection

1. Check that all variables are stationary. If not, apply differencing. Where appropriate, use the same differencing for all variables to preserve interpretability.

2. Fit regression model with AR(2) errors for non-seasonal data or ARIMA(2,0,0)(1,0,0) errors for seasonal data.

3. Calculate errors ($e_t$) from fitted regression model and identify ARMA model for them.

4. Re-fit entire model using new ARMA model for errors.

5. Check that $e_t$ series looks like white noise.

---

Selecting predictors

- AIC can be calculated for final model.
- Repeat procedure for all subsets of predictors to be considered, and select model with lowest AIC value.
Quarterly changes in US consumption and personal income

Year


consumption

income

Forecasting: Principles and Practice
Regression with ARIMA errors
No need for transformations or further differencing.

Increase in income does not necessarily translate into instant increase in consumption (e.g., after the loss of a job, it may take a few months for expenses to be reduced to allow for the new circumstances). We will ignore this for now.

Try a simple regression with AR(2) proxy model for errors.
No need for transformations or further differencing.

Increase in income does not necessarily translate into instant increase in consumption (e.g., after the loss of a job, it may take a few months for expenses to be reduced to allow for the new circumstances). We will ignore this for now.

Try a simple regression with AR(2) proxy model for errors.
No need for transformations or further differencing.

Increase in income does not necessarily translate into instant increase in consumption (e.g., after the loss of a job, it may take a few months for expenses to be reduced to allow for the new circumstances). We will ignore this for now.

Try a simple regression with AR(2) proxy model for errors.
```
fit <- Arima(usconsumption[,1],
            xreg=usconsumption[,2],
            order=c(2,0,0))
tsddisplay(arima.errors(fit),
            main="ARIMA errors")
```
US Personal Consumption and Income

arima.errors(fit)

![Plot of ARIMA errors](image)

```
arima.errors(fit)
−2 −1 0 ... ●
●
●
●
●
●
●
●
●
●
● ●
●
●
5 10 15 20
−0.2 0.0 0.2
Lag
ACF
5 10 15 20
−0.2 0.0 0.2
Lag
PACF
```
Candidate ARIMA models include MA(3) and AR(2).

ARIMA(1,0,2) has lowest $AIC_c$ value.

Refit model with ARIMA(1,0,2) errors.

```r
> (fit2 <- Arima(usconsumption[,1], xreg=usconsumption[,2], order=c(1,0,2)))

Coefficients:

  ar1  ma1  ma2 intercept usconsumption[,2]
  0.6516 -0.5440  0.2187   0.5750   0.2420

s.e.  0.1468  0.1576  0.0790   0.0951   0.0513

sigma^2 estimated as 0.3396:  log likelihood=-144.27
AIC=300.54  AICc=301.08  BIC=319.14
```
Candidate ARIMA models include MA(3) and AR(2).

ARIMA(1,0,2) has lowest AIC\textsubscript{c} value.

Refit model with ARIMA(1,0,2) errors.

```r
> (fit2 <- Arima(usconsumption[,1],
                   xreg=usconsumption[,2],
                   order=c(1,0,2)))
```

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>ma1</th>
<th>ma2</th>
<th>intercept</th>
<th>usconsumption[,2]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6516</td>
<td>-0.5440</td>
<td>0.2187</td>
<td>0.5750</td>
<td>0.2420</td>
</tr>
</tbody>
</table>

s.e.    | 0.1468  | 0.1576  | 0.0790  | 0.0951    | 0.0513            |

sigma^2 estimated as 0.3396: log likelihood=-144.27
AIC=300.54  AICc=301.08  BIC=319.14
Candidate ARIMA models include MA(3) and AR(2).

ARIMA(1,0,2) has lowest AIC\textsubscript{c} value.

Refit model with ARIMA(1,0,2) errors.

```r
> (fit2 <- Arima(usconsumption[,1],
                  xreg=usconsumption[,2], order=c(1,0,2)))

Coefficients:
            ar1    ma1    ma2 intercept usconsumption[,2]
          0.6516 -0.5440  0.2187    0.5750       0.2420
s.e. 0.1468  0.1576  0.0790    0.0951       0.0513

sigma^2 estimated as 0.3396:  log likelihood=-144.27
AIC=300.54  AICc=301.08  BIC=319.14
```
Candidate ARIMA models include MA(3) and AR(2).

ARIMA(1,0,2) has lowest $AIC_c$ value.

Refit model with ARIMA(1,0,2) errors.

```r
> (fit2 <- Arima(usconsumption[,1],
                   xreg=usconsumption[,2], order=c(1,0,2)))
```

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>ma1</th>
<th>ma2</th>
<th>intercept</th>
<th>usconsumption[,2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar1</td>
<td>0.6516</td>
<td>-0.5440</td>
<td>0.2187</td>
<td>0.5750</td>
<td>0.2420</td>
</tr>
<tr>
<td>s.e.</td>
<td>0.1468</td>
<td>0.1576</td>
<td>0.0790</td>
<td>0.0951</td>
<td>0.0513</td>
</tr>
</tbody>
</table>

$sigma^2$ estimated as 0.3396: log likelihood=-144.27

AIC=300.54   AICc=301.08   BIC=319.14
Candidate ARIMA models include MA(3) and AR(2).

ARIMA(1,0,2) has lowest AIC$_c$ value.

Refit model with ARIMA(1,0,2) errors.

```r
> (fit2 <- Arima(usconsumption[,1], xreg=usconsumption[,2], order=c(1,0,2)))
```

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>ma1</th>
<th>ma2</th>
<th>intercept</th>
<th>usconsumption[1,2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coeff.</td>
<td>0.6516</td>
<td>-0.5440</td>
<td>0.2187</td>
<td>0.5750</td>
<td>0.2420</td>
</tr>
<tr>
<td>s.e.</td>
<td>0.1468</td>
<td>0.1576</td>
<td>0.0790</td>
<td>0.0951</td>
<td>0.0513</td>
</tr>
</tbody>
</table>

sigma^2 estimated as 0.3396: log likelihood=-144.27
AIC=300.54   AICc=301.08   BIC=319.14
The whole process can be automated:

> auto.arima(usconsumption[,1], xreg=usconsumption[,2])

Series: usconsumption[, 1]
ARIMA(1,0,2) with non-zero mean

Coefficients:

<table>
<thead>
<tr>
<th>ar1</th>
<th>ma1</th>
<th>ma2</th>
<th>intercept</th>
<th>usconsumption[,2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6516</td>
<td>-0.5440</td>
<td>0.2187</td>
<td>0.5750</td>
<td>0.2420</td>
</tr>
</tbody>
</table>

s.e. 0.1468 0.1576 0.0790 0.0951 0.0513

sigma^2 estimated as 0.3396: log likelihood=-144.27
AIC=300.54 AICc=301.08 BIC=319.14
> Box.test(residuals(fit2), fitdf=5, lag=10, type="Ljung")

*Box-Ljung test*

data:  residuals(fit2)
X-squared = 4.5948, df = 5, p-value = 0.4673
fcast <- forecast(fit2,  
    xreg=rep(mean(usconsumption[,2]),8), h=8)

plot(fcast,  
    main="Forecasts from regression with ARIMA(1,0,2) errors")
To forecast a regression model with ARIMA errors, we need to forecast the regression part of the model and the ARIMA part of the model and combine the results.

Forecasts of macroeconomic variables may be obtained from the ABS, for example.

Separate forecasting models may be needed for other explanatory variables.

Some explanatory variable are known into the future (e.g., time, dummies).
To forecast a regression model with ARIMA errors, we need to forecast the regression part of the model and the ARIMA part of the model and combine the results.

Forecasts of macroeconomic variables may be obtained from the ABS, for example.

Separate forecasting models may be needed for other explanatory variables.

Some explanatory variable are known into the future (e.g., time, dummies).
To forecast a regression model with ARIMA errors, we need to forecast the regression part of the model and the ARIMA part of the model and combine the results.

Forecasts of macroeconomic variables may be obtained from the ABS, for example.

Separate forecasting models may be needed for other explanatory variables.

Some explanatory variable are known into the future (e.g., time, dummies).
To forecast a regression model with ARIMA errors, we need to forecast the regression part of the model and the ARIMA part of the model and combine the results.

Forecasts of macroeconomic variables may be obtained from the ABS, for example.

Separate forecasting models may be needed for other explanatory variables.

Some explanatory variable are known into the future (e.g., time, dummies).
1 Regression with ARIMA errors

2 Stochastic and deterministic trends

3 Periodic seasonality

4 Dynamic regression models
**Stochastic & deterministic trends**

**Deterministic trend**

\[ y_t = \beta_0 + \beta_1 t + n_t \]

where \( n_t \) is ARMA process.

**Stochastic trend**

\[ y_t = \beta_0 + \beta_1 t + n_t \]

where \( n_t \) is ARIMA process with \( d \geq 1 \).

Difference both sides until \( n_t \) is stationary:

\[ y'_t = \beta_1 + n'_t \]

where \( n'_t \) is ARMA process.
Deterministic trend

\[ y_t = \beta_0 + \beta_1 t + n_t \]

where \( n_t \) is ARMA process.

Stochastic trend

\[ y_t = \beta_0 + \beta_1 t + n_t \]

where \( n_t \) is ARIMA process with \( d \geq 1 \).

Difference both sides until \( n_t \) is stationary:

\[ y'_t = \beta_1 + n'_t \]

where \( n'_t \) is ARMA process.
Stochastic & deterministic trends

Deterministic trend

\[ y_t = \beta_0 + \beta_1 t + n_t \]

where \( n_t \) is ARMA process.

Stochastic trend

\[ y_t = \beta_0 + \beta_1 t + n_t \]

where \( n_t \) is ARIMA process with \( d \geq 1 \).

Difference both sides until \( n_t \) is stationary:

\[ y'_t = \beta_1 + n'_t \]

where \( n'_t \) is ARMA process.
International visitors

**Deterministic trend**

```r
> auto.arima(austa,d=0,xreg=1:length(austa))
ARIMA(2,0,0) with non-zero mean
```

Coefficients:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ar1</td>
<td>ar2</td>
<td>intercept</td>
<td>1:length(austa)</td>
<td></td>
</tr>
<tr>
<td>1.0371</td>
<td>-0.3379</td>
<td>0.4173</td>
<td>0.1715</td>
<td></td>
</tr>
<tr>
<td>s.e.</td>
<td>0.1675</td>
<td>0.1797</td>
<td>0.1866</td>
<td>0.0102</td>
</tr>
</tbody>
</table>

\( \sigma^2 \) estimated as 0.02486: log likelihood=12.7
AIC=-15.4   AICc=-13   BIC=-8.23

\[
y_t = 0.4173 + 0.1715t + n_t
\]
\[
n_t = 1.0371n_{t-1} - 0.3379n_{t-2} + e_t
\]
\[
e_t \sim \text{NID}(0, 0.02486).
\]
International visitors

Deterministic trend

\[
\text{ > auto.arima(austa,d=0,xreg=1:length(austa))}
\]

ARIMA(2,0,0) with non-zero mean

Coefficients:

<table>
<thead>
<tr>
<th>ar1</th>
<th>ar2</th>
<th>intercept</th>
<th>1:length(austa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0371</td>
<td>-0.3379</td>
<td>0.4173</td>
<td>0.1715</td>
</tr>
</tbody>
</table>

s.e. 0.1675 0.1797 0.1866 0.0102

\[
\sigma^2 \text{ estimated as 0.02486: log likelihood=12.7}
\]

AIC=-15.4 AICc=-13 BIC=-8.23

\[
y_t = 0.4173 + 0.1715t + n_t
\]

\[
n_t = 1.0371n_{t-1} - 0.3379n_{t-2} + e_t
\]

\[
e_t \sim \text{NID}(0,0.02486).
\]
International visitors

**Stochastic trend**

> auto.arima(austa,d=1)

ARIMA(0,1,0) with drift

Coefficients:
  
  drift
  0.1538
  
  s.e. 0.0323

sigma^2 estimated as 0.03132: log likelihood=9.38

AIC=-14.76  AICc=-14.32  BIC=-11.96

\[ y_t - y_{t-1} = 0.1538 + e_t \]

\[ y_t = y_0 + 0.1538t + n_t \]

\[ n_t = n_{t-1} + e_t \]

\[ e_t \sim \text{NID}(0, 0.03132). \]
International visitors

Stochastic trend

> auto.arima(austa,d=1)
ARIMA(0,1,0) with drift

Coefficients:
  drift
    0.1538
s.e. 0.0323

sigma^2 estimated as 0.03132: log likelihood=9.38
AIC=-14.76 AICc=-14.32 BIC=-11.96

\[ y_t - y_{t-1} = 0.1538 + e_t \]
\[ y_t = y_0 + 0.1538t + n_t \]
\[ n_t = n_{t-1} + e_t \]
\[ e_t \sim \text{NID}(0, 0.03132). \]
International visitors

Forecasts from linear trend + AR(2) error

Forecasts from ARIMA(0,1,0) with drift
Point forecasts are almost identical, but prediction intervals differ.

Stochastic trends have much wider prediction intervals because the errors are non-stationary.

Be careful of forecasting with deterministic trends too far ahead.
Forecasting with trend

- Point forecasts are almost identical, but prediction intervals differ.
- Stochastic trends have much wider prediction intervals because the errors are non-stationary.
- Be careful of forecasting with deterministic trends too far ahead.
Point forecasts are almost identical, but prediction intervals differ.

Stochastic trends have much wider prediction intervals because the errors are non-stationary.

Be careful of forecasting with deterministic trends too far ahead.
1. Regression with ARIMA errors
2. Stochastic and deterministic trends
3. Periodic seasonality
4. Dynamic regression models
Fourier terms for seasonality

Periodic seasonality can be handled using pairs of Fourier terms:

\[ s_k(t) = \sin \left( \frac{2\pi kt}{m} \right) \quad c_k(t) = \cos \left( \frac{2\pi kt}{m} \right) \]

\[ y_t = \sum_{k=1}^{K} [\alpha_k s_k(t) + \beta_k c_k(t)] + n_t \]

- \( n_t \) is non-seasonal ARIMA process.
- Every periodic function can be approximated by sums of sin and cos terms for large enough \( K \).
- Choose \( K \) by minimizing AICc.
Fourier terms for seasonality

Periodic seasonality can be handled using pairs of Fourier terms:

\[ s_k(t) = \sin \left( \frac{2\pi kt}{m} \right) \quad c_k(t) = \cos \left( \frac{2\pi kt}{m} \right) \]

\[ y_t = \sum_{k=1}^{K} [\alpha_k s_k(t) + \beta_k c_k(t)] + n_t \]

- \( n_t \) is non-seasonal ARIMA process.
- Every periodic function can be approximated by sums of sin and cos terms for large enough \( K \).
- Choose \( K \) by minimizing AICc.
Fourier terms for seasonality

Periodic seasonality can be handled using pairs of Fourier terms:

\[ s_k(t) = \sin \left( \frac{2\pi kt}{m} \right) \quad c_k(t) = \cos \left( \frac{2\pi kt}{m} \right) \]

\[ y_t = \sum_{k=1}^{K} \left[ \alpha_k s_k(t) + \beta_k c_k(t) \right] + n_t \]

- \( n_t \) is non-seasonal ARIMA process.
- Every periodic function can be approximated by sums of sin and cos terms for large enough \( K \).
- Choose \( K \) by minimizing AICc.
fit <- auto.arima(USAccDeaths,
    xreg=fourier(USAccDeaths, 5),
    seasonal=FALSE)

fc <- forecast(fit,
    xreg=fourierf(USAccDeaths, 5, 24))

plot(fc)
US Accidental Deaths

Forecasts from ARIMA(0,1,1)
1 Regression with ARIMA errors
2 Stochastic and deterministic trends
3 Periodic seasonality
4 Dynamic regression models
Dynamic regression models

Sometimes a change in $x_t$ does not affect $y_t$ instantaneously

1. $y_t = \text{sales}, \ x_t = \text{advertising}$.
2. $y_t = \text{stream flow}, \ x_t = \text{rainfall}$.
3. $y_t = \text{size of herd}, \ x_t = \text{breeding stock}$.
Dynamic regression models

Sometimes a change in $x_t$ does not affect $y_t$ instantaneously

1. $y_t = \text{sales, } x_t = \text{advertising}$.
2. $y_t = \text{stream flow, } x_t = \text{rainfall}$.
3. $y_t = \text{size of herd, } x_t = \text{breeding stock}$.
Dynamic regression models

Sometimes a change in $x_t$ does not affect $y_t$ instantaneously.

1. $y_t = \text{sales}, \; x_t = \text{advertising}$.
2. $y_t = \text{stream flow}, \; x_t = \text{rainfall}$.
3. $y_t = \text{size of herd}, \; x_t = \text{breeding stock}$.

These are dynamic systems with input ($x_t$) and output ($y_t$).
Dynamic regression models

Sometimes a change in $x_t$ does not affect $y_t$ instantaneously.

1. $y_t = \text{sales}$, $x_t = \text{advertising}$.
2. $y_t = \text{stream flow}$, $x_t = \text{rainfall}$.
3. $y_t = \text{size of herd}$, $x_t = \text{breeding stock}$.

These are dynamic systems with input ($x_t$) and output ($y_t$).

$x_t$ is often a leading indicator.

There can be multiple predictors.
Sometimes a change in $x_t$ does not affect $y_t$ instantaneously

1. $y_t = \text{sales, } x_t = \text{advertising.}$
2. $y_t = \text{stream flow, } x_t = \text{rainfall.}$
3. $y_t = \text{size of herd, } x_t = \text{breeding stock.}$

These are dynamic systems with input ($x_t$) and output ($y_t$).

$x_t$ is often a leading indicator.

There can be multiple predictors.
Dynamic regression models

Sometimes a change in $x_t$ does not affect $y_t$ instantaneously

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y_t = \text{sales}, \ x_t = \text{advertising}$</td>
</tr>
<tr>
<td>2</td>
<td>$y_t = \text{stream flow}, \ x_t = \text{rainfall}$</td>
</tr>
<tr>
<td>3</td>
<td>$y_t = \text{size of herd}, \ x_t = \text{breeding stock}$</td>
</tr>
</tbody>
</table>

- These are dynamic systems with input ($x_t$) and output ($y_t$).
- $x_t$ is often a leading indicator.
- There can be multiple predictors.
Dynamic regression models

Sometimes a change in $x_t$ does not affect $y_t$ instantaneously

1. $y_t = \text{sales}, \ x_t = \text{advertising}.$
2. $y_t = \text{stream flow}, \ x_t = \text{rainfall}.$
3. $y_t = \text{size of herd}, \ x_t = \text{breeding stock}.$

- These are dynamic systems with input ($x_t$) and output ($y_t$).
- $x_t$ is often a leading indicator.
- There can be multiple predictors.
Sometimes a change in $x_t$ does not affect $y_t$ instantaneously

1. $y_t = \text{sales}, \ x_t = \text{advertising}.$
2. $y_t = \text{stream flow}, \ x_t = \text{rainfall}.$
3. $y_t = \text{size of herd}, \ x_t = \text{breeding stock}.$

- These are dynamic systems with input ($x_t$) and output ($y_t$).
- $x_t$ is often a leading indicator.
- There can be multiple predictors.
Lagged explanatory variables

The model include present and past values of predictor: $x_t, x_{t-1}, x_{t-2}, \ldots$

$$y_t = a + \nu_0 x_t + \nu_1 x_{t-1} + \cdots + \nu_k x_{t-k} + n_t$$

where $n_t$ is an ARIMA process.

Rewrite model as

$$y_t = a + (\nu_0 + \nu_1 B + \nu_2 B^2 + \cdots + \nu_k B^k)x_t + n_t$$

$$= a + \nu(B)x_t + n_t.$$

- $\nu(B)$ is called a transfer function since it describes how change in $x_t$ is transferred to $y_t$.
- $x$ can influence $y$, but $y$ is not allowed to influence $x$. 

Forecasting: Principles and Practice

Dynamic regression models 35
Lagged explanatory variables

The model include present and past values of predictor: $x_t, x_{t-1}, x_{t-2}, \ldots$

$$y_t = a + \nu_0 x_t + \nu_1 x_{t-1} + \cdots + \nu_k x_{t-k} + n_t$$

where $n_t$ is an ARIMA process.

**Rewrite model as**

$$y_t = a + (\nu_0 + \nu_1 B + \nu_2 B^2 + \cdots + \nu_k B^k)x_t + n_t$$

$$= a + \nu(B)x_t + n_t.$$

- $\nu(B)$ is called a *transfer function* since it describes how change in $x_t$ is transferred to $y_t$.
- $x$ can influence $y$, but $y$ is not allowed to influence $x$. 

*Forecasting: Principles and Practice*
Lagged explanatory variables

The model include present and past values of predictor: \( x_t, x_{t-1}, x_{t-2}, \ldots \)

\[
y_t = a + \nu_0 x_t + \nu_1 x_{t-1} + \cdots + \nu_k x_{t-k} + n_t
\]

where \( n_t \) is an ARIMA process.

**Rewrite model as**

\[
y_t = a + (\nu_0 + \nu_1 B + \nu_2 B^2 + \cdots + \nu_k B^k) x_t + n_t
\]

\[
= a + \nu(B) x_t + n_t.
\]

\( \nu(B) \) is called a *transfer function* since it describes how change in \( x_t \) is transferred to \( y_t \).

- \( x \) can influence \( y \), but \( y \) is not allowed to influence \( x \).
Lagged explanatory variables

The model includes present and past values of predictor: $x_t, x_{t-1}, x_{t-2}, \ldots$

$$y_t = a + \nu_0 x_t + \nu_1 x_{t-1} + \cdots + \nu_k x_{t-k} + n_t$$

where $n_t$ is an ARIMA process.

**Rewrite model as**

$$y_t = a + (\nu_0 + \nu_1 B + \nu_2 B^2 + \cdots + \nu_k B^k)x_t + n_t$$

$$= a + \nu(B)x_t + n_t.$$  

- $\nu(B)$ is called a *transfer function* since it describes how change in $x_t$ is transferred to $y_t$.
- $x$ can influence $y$, but $y$ is not allowed to influence $x$. 

*Forecasting: Principles and Practice*
Example: Insurance quotes and TV adverts

Insurance advertising and quotations

Quotes

TV Adverts

2002 2003 2004 2005
Example: Insurance quotes and TV adverts

```r
> Advert <- cbind(insurance[,2],
                  c(NA,insurance[1:39,2]))
> colnames(Advert) <- c("AdLag0","AdLag1")
> fit <- auto.arima(insurance[,1], xreg=Advert, d=0)
ARIMA(3,0,0) with non-zero mean

Coefficients:
                 ar1    ar2    ar3 intercept AdLag0 AdLag1
 1.4117  -0.9317  0.3591  2.0393    1.2564   0.1625
 s.e.  0.1698  0.2545  0.1592  0.9931    0.0667   0.0591

sigma^2 estimated as 0.1887: log likelihood=-23.89
AIC=61.78   AICc=65.28   BIC=73.6

y_t = 2.04 + 1.26x_t + 0.16x_{t-1} + n_t
n_t = 1.41n_{t-1} - 0.93n_{t-2} + 0.36n_{t-3}
```
Example: Insurance quotes and TV adverts

> Advert <- cbind(insurance[,2], 
> c(NA,insurance[1:39,2]))
> colnames(Advert) <- c("AdLag0","AdLag1")
> fit <- auto.arima(insurance[,1], xreg=Advert, d=0)
ARIMA(3,0,0) with non-zero mean

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>ar2</th>
<th>ar3</th>
<th>intercept</th>
<th>AdLag0</th>
<th>AdLag1</th>
</tr>
</thead>
<tbody>
<tr>
<td>estimate</td>
<td>1.4117</td>
<td>-0.9317</td>
<td>0.3591</td>
<td>2.0393</td>
<td>1.2564</td>
<td>0.1625</td>
</tr>
<tr>
<td>s.e.</td>
<td>0.1698</td>
<td>0.2545</td>
<td>0.1592</td>
<td>0.9931</td>
<td>0.0667</td>
<td>0.0591</td>
</tr>
</tbody>
</table>

sigma^2 estimated as 0.1887: log likelihood=-23.89
AIC=61.78    AICc=65.28    BIC=73.6

\[ y_t = 2.04 + 1.26x_t + 0.16x_{t-1} + n_t \]
\[ n_t = 1.41n_{t-1} - 0.93n_{t-2} + 0.36n_{t-3} \]
Example: Insurance quotes and TV adverts

Forecast quotes with advertising set to 10

Quotes

2002 2003 2004 2005 2006 2007
8 10 12 14 16 18
Example: Insurance quotes and TV adverts

Forecast quotes with advertising set to 8
Forecast quotes with advertising set to 6

Quotes
2002 2003 2004 2005 2006 2007
8 10 12 14 16 18
fc <- forecast(fit, h=20, 
xreg=cbind(c(Advert[40,1],rep(6,19)), rep(6,20)))
plot(fc)
Dynamic regression models

\[ y_t = a + \nu(B)x_t + n_t \]

where \( n_t \) is an ARMA process. So

\[ \phi(B)n_t = \theta(B)e_t \quad \text{or} \quad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t. \]

ARMA models are rational approximations to general transfer functions of \( e_t \).

We can also replace \( \nu(B) \) by a rational approximation.

There is no R package for forecasting using a general transfer function approach.
Dynamic regression models

\[ y_t = a + \nu(B)x_t + n_t \]

where \( n_t \) is an ARMA process. So

\[ \phi(B)n_t = \theta(B)e_t \quad \text{or} \quad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t. \]

\[ y_t = a + \nu(B)x_t + \psi(B)e_t \]

- ARMA models are rational approximations to general transfer functions of \( e_t \).
- We can also replace \( \nu(B) \) by a rational approximation.
- There is no R package for forecasting using a general transfer function approach.
Dynamic regression models

\[ y_t = a + \nu(B)x_t + n_t \]

where \( n_t \) is an ARMA process. So

\[ \phi(B)n_t = \theta(B)e_t \quad \text{or} \quad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t. \]

\[ y_t = a + \nu(B)x_t + \psi(B)e_t \]

- ARMA models are rational approximations to general transfer functions of \( e_t \).
- We can also replace \( \nu(B) \) by a rational approximation.
- There is no R package for forecasting using a general transfer function approach.
\[ y_t = a + \nu(B)x_t + n_t \]

where \( n_t \) is an ARMA process. So

\[ \phi(B)n_t = \theta(B)e_t \quad \text{or} \quad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t. \]

\[ y_t = a + \nu(B)x_t + \psi(B)e_t \]

- ARMA models are rational approximations to general transfer functions of \( e_t \).
- We can also replace \( \nu(B) \) by a rational approximation.
- There is no R package for forecasting using a general transfer function approach.
Dynamic regression models

\[ y_t = a + \nu(B)x_t + n_t \]

where \( n_t \) is an ARMA process. So

\[ \phi(B)n_t = \theta(B)e_t \quad \text{or} \quad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t. \]

\[ y_t = a + \nu(B)x_t + \psi(B)e_t \]

- ARMA models are rational approximations to general transfer functions of \( e_t \).
- We can also replace \( \nu(B) \) by a rational approximation.
- There is no R package for forecasting using a general transfer function approach.