MONASH University

Rob J Hyndman

Forecasting:

Principles and Practice

10. Dynamic regression

OTexts.com/fpp/9/1/

Outline

Regression with ARIMA errors

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 $x_{1,t}, \dots, x_{k,t}$.
- 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 with ARIMA errors

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 $x_{1,t}, \dots, x_{k,t}$.
- 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 with ARIMA errors

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 $x_{1,t}, \dots, x_{k,t}$.
- 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 with ARIMA errors

Regression models

$$y_t = \beta_0 + \beta_1 x_{1,t} + \dots + \beta_k x_{k,t} + e_t,$$

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

Example: ARIMA(1,1,1) errors

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

where e_t is white noise .

Regression with ARIMA errors

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 $x_{1,t}, \dots, x_{k,t}$.
- 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 .

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 = 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 = 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 $n_t = e_t$.

Estimation

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

1. Estimated coefficients $\hat{\beta}_0, \dots, \hat{\beta}_k$ 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 $\hat{\beta}_0, \dots, \hat{\beta}_k$ 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 $\hat{\beta}_0, \dots, \hat{\beta}_k$ 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 $\hat{\beta}_0, \dots, \hat{\beta}_k$ 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 $\hat{\beta}_0, \dots, \hat{\beta}_k$ 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 $\hat{\beta}_0, \dots, \hat{\beta}_k$ 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.

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.

Stationarity

Model with ARIMA(1,1,1) errors

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

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

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

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

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

$$\text{where} \quad \phi(B)(1-B)^d N_t = \theta(B)e_t$$

After differencing all variables

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

$$\text{where} \quad \phi(B)N_t = \theta(B)e_t$$

$$\text{and} \quad y'_t = (1-B)^d y_t$$

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

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

$$\text{where} \quad \phi(B)(1-B)^d N_t = \theta(B)e_t$$

After differencing all variables

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

$$\text{where} \quad \phi(B)N_t = \theta(B)e_t$$

$$\text{and} \quad y'_t = (1-B)^d y_t$$

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

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

$$\text{where} \quad \phi(B)(1-B)^d N_t = \theta(B) e_t$$

After differencing all variables

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

$$\text{where} \quad \phi(B) N_t = \theta(B) e_t$$

$$\text{and} \quad y'_t = (1-B)^d y_t$$

Model selection

- To determine ARIMA error structure, first need to calculate n_t.
- We can't get n_t without knowing $\beta_0, \dots, \beta_k$.
- 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.

Model selection

- To determine ARIMA error structure, first need to calculate n_t.
- We can't get n_t without knowing $\beta_0, \dots, \beta_k$.
- 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 for non-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 $\beta_0, \dots, \beta_k$.
- 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 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 $\beta_0, \dots, \beta_k$.
- 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 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 $\beta_0, \dots, \beta_k$.
- 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 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 $\beta_0, \dots, \beta_k$.
- 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 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 $\beta_0, \dots, \beta_k$.
- 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 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 $\beta_0, \dots, \beta_k$.
- 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 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.

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 sa… …rve in…
2. Fi… n… e…
3. Ca… 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 sa… …rve in…
2. Fi… no… er…
3. Ca… 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.

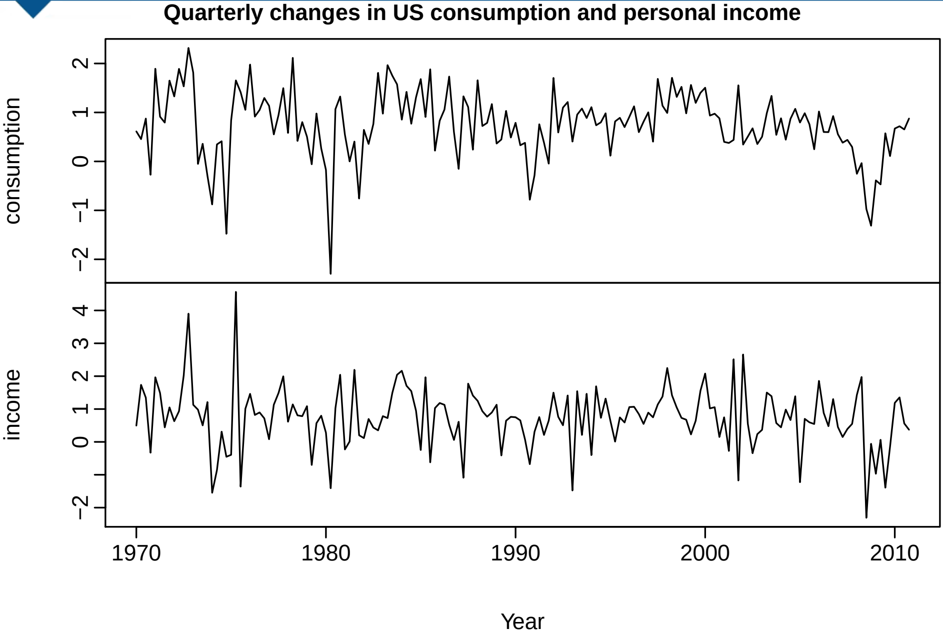
Model selection

1. Check that all variables are stationary. If not, apply differencing. Where appropriate, use the sa… …rve in…
2. Fi… nc… er…
3. Ca… 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.

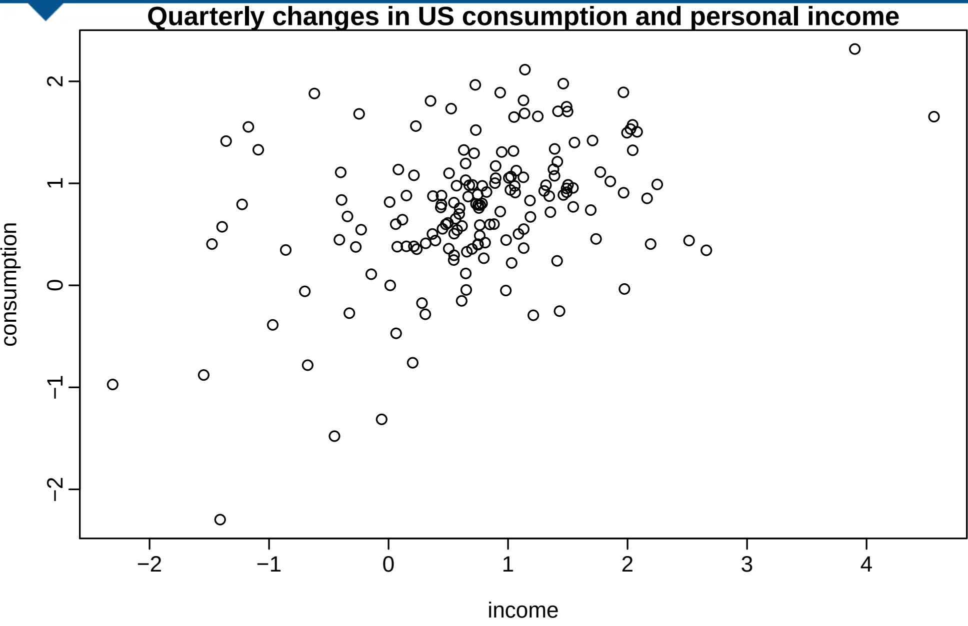
Selecting predictors

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

US personal consumption & income



US personal consumption & income



US Personal Consumption and Income

- 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.

US Personal Consumption and Income

- 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.

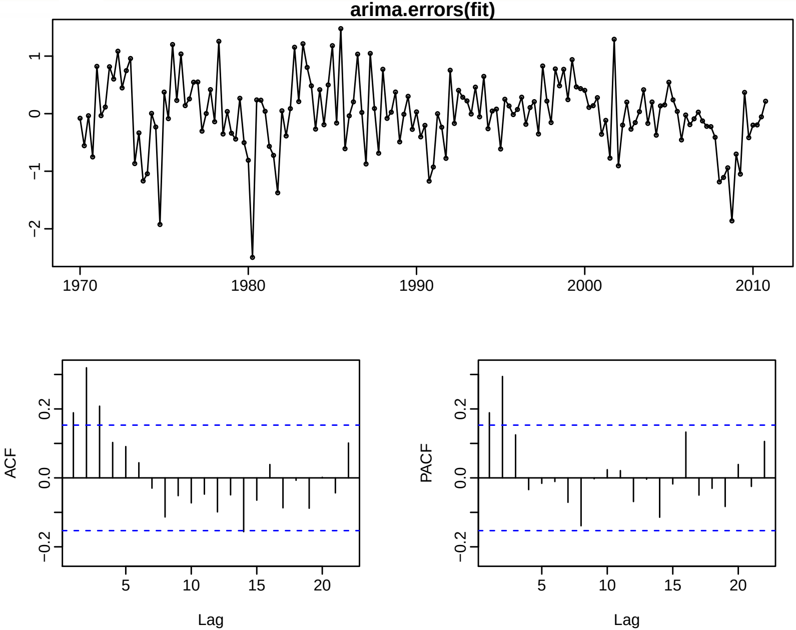
US Personal Consumption and Income

- 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.

US Personal Consumption and Income

```
fit <- Arima(usconsumption[,1],
             xreg=usconsumption[,2],
             order=c(2,0,0))
tsdisplay(arima.errors(fit),
             main="ARIMA errors")
```

US Personal Consumption and Income



US Personal Consumption and Income

- 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.

```
> (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
```

US Personal Consumption and Income

- 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.

```
> (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
```

US Personal Consumption and Income

- 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.

```
> (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
```

US Personal Consumption and Income

- 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.

```
> (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
```

US Personal Consumption and Income

- 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.

```
> (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
```

US Personal Consumption and Income

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:
        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
```

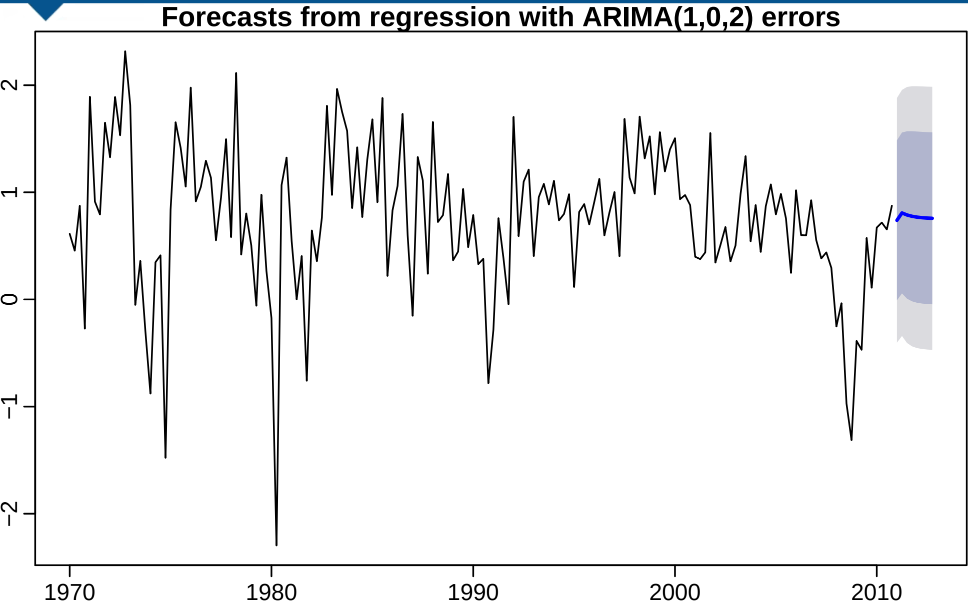
US Personal Consumption and Income

```
> 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
```

US Personal Consumption and Income

```
fcast <- forecast(fit2,
  xreg=rep(mean(usconsumption[,2]),8), h=8)

plot(fcast,
  main="Forecasts from regression with
    ARIMA(1,0,2) errors")
```

US Personal Consumption and Income



Forecasting

- 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).

Forecasting

- 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).

Forecasting

- 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).

Forecasting

- 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).

Outline

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.

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.

Stochastic & deterministic trends

Deterministic trend

$$y_t = \beta_0 + \beta_1 t + n_t$$

where n_t is ARMA process.

Stochastic trend

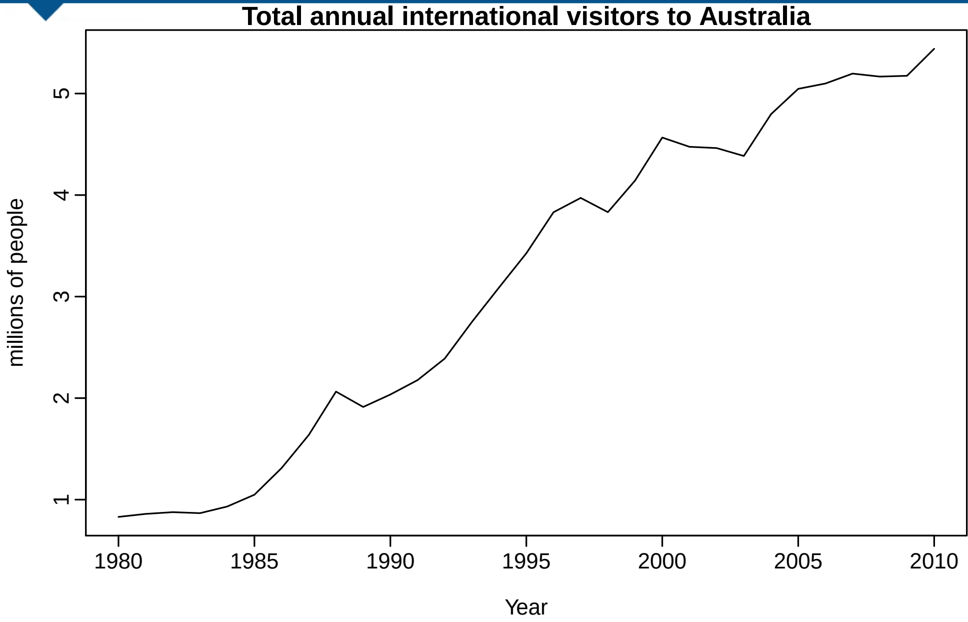
$$y_t = \beta_0 + \beta_1 t + n_t$$

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



International visitors

Deterministic trend

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

Coefficients:
         ar1      ar2  intercept  1:length(austa)
      1.0371  -0.3379     0.4173           0.1715
s.e.  0.1675   0.1797     0.1866           0.0102

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

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

Coefficients:
         ar1      ar2  intercept  1:length(austa)
      1.0371  -0.3379     0.4173           0.1715
s.e.  0.1675   0.1797     0.1866           0.0102

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

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
```

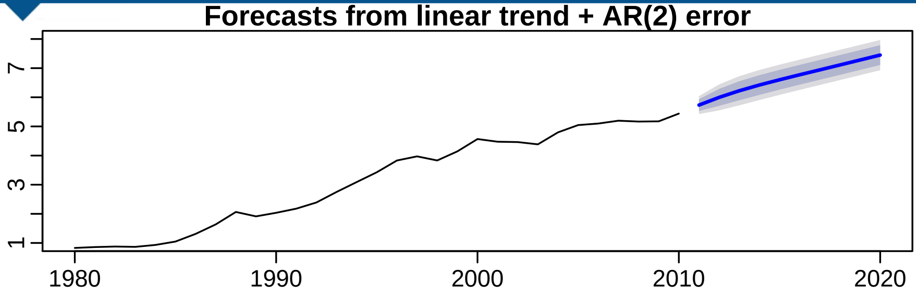
$$y_t - y_{t-1} = 0.1538 + e_t$$

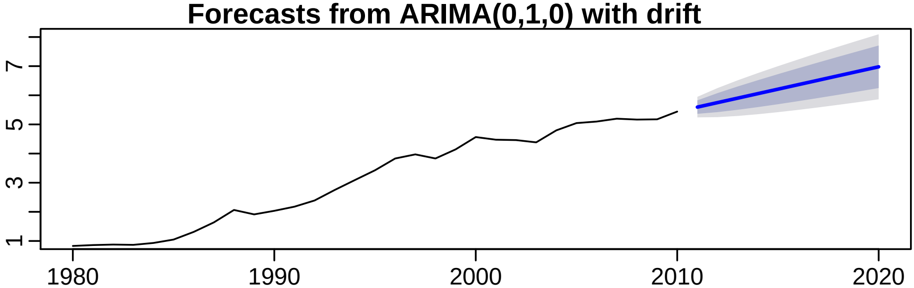
$$y_t = y_0 + 0.1538t + n_t$$

$$n_t = n_{t-1} + e_t$$

$$e_t \sim \text{NID}(0, 0.03132).$$

International visitors





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.

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.

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.

Outline

Fourier terms for seasonality

Periodic seasonality can be handled using pairs of Fourier terms:

$$s_k(t) = \sin\left(\frac{2\pi kt}{m}\right) \qquad c_k(t) = \cos\left(\frac{2\pi kt}{m}\right)$$

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

- 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:

$$s_k(t) = \sin\left(\frac{2\pi kt}{m}\right) \qquad c_k(t) = \cos\left(\frac{2\pi kt}{m}\right)$$

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

- 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:

$$s_k(t) = \sin\left(\frac{2\pi kt}{m}\right) \qquad c_k(t) = \cos\left(\frac{2\pi kt}{m}\right)$$

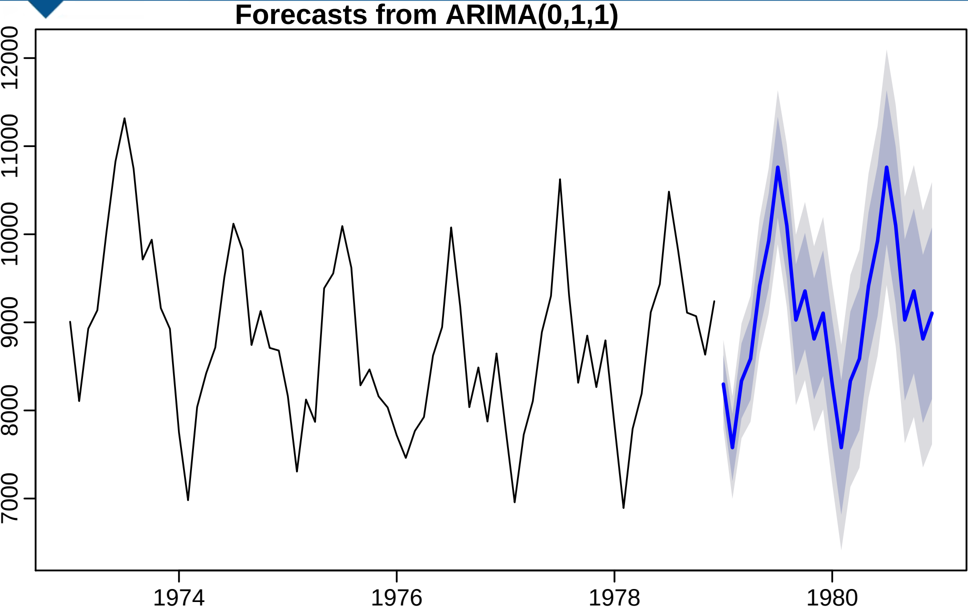
$$y_t = \sum_{k=1}^{K} [\alpha_k s_k(t) + \beta_k c_k(t)] + n_t$$

- 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.

US Accidental Deaths

```
fit <- auto.arima(USAccDeaths,
         xreg=fourier(USAccDeaths, 5),
         seasonal=FALSE)

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

plot(fc)
```

US Accidental Deaths



Outline

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 = sales, x_t = advertising.
2. y_t = stream flow, x_t = rainfall.
3. y_t = size of herd, x_t = breeding stock.

Dynamic regression models

Sometimes a change in x_t does not affect y_t instantaneously

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

Dynamic regression models

Sometimes a change in x_t does not affect y_t instantaneously

1. y_t = sales, x_t = advertising.
2. y_t = stream flow, x_t = rainfall.
3. y_t = size of herd, x_t = 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. $y_t =$ sales, $x_t =$ advertising.
2. $y_t =$ stream flow, $x_t =$ rainfall.
3. $y_t =$ size of herd, $x_t =$ 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

1. y_t = sales, x_t = advertising.
2. y_t = stream flow, x_t = rainfall.
3. y_t = size of herd, x_t = 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

1. $y_t =$ sales, $x_t =$ advertising.
2. $y_t =$ stream flow, $x_t =$ rainfall.
3. $y_t =$ size of herd, $x_t =$ 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

1. y_t = sales, x_t = advertising.
2. y_t = stream flow, x_t = rainfall.
3. y_t = size of herd, x_t = 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

1. $y_t =$ sales, $x_t =$ advertising.
2. $y_t =$ stream flow, $x_t =$ rainfall.
3. $y_t =$ size of herd, $x_t =$ 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

$$\begin{aligned} y_t &= a + (\nu_0 + \nu_1 B + \nu_2 B^2 + \cdots + \nu_k B^k) x_t + n_t \\ &= a + \nu(B) x_t + n_t. \end{aligned}$$

- $\nu(B)$ 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 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

$$\begin{aligned} y_t &= a + (\nu_0 + \nu_1 B + \nu_2 B^2 + \cdots + \nu_k B^k) x_t + n_t \\ &= a + \nu(B) x_t + n_t. \end{aligned}$$

- $\nu(B)$ 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 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

$$\begin{aligned} y_t &= a + (\nu_0 + \nu_1 B + \nu_2 B^2 + \cdots + \nu_k B^k) x_t + n_t \\ &= a + \nu(B) x_t + n_t. \end{aligned}$$

- $\nu(B)$ 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 include present and past values of predictor: $x_t, x_{t-1}, x_{t-2}, \ldots.$

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

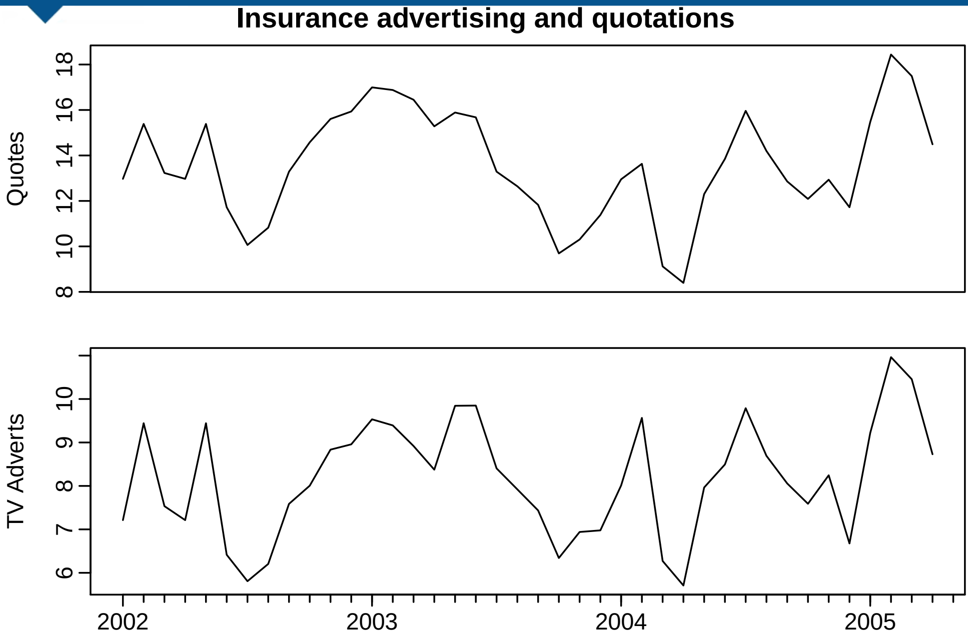
where n_t is an ARIMA process.

Rewrite model as

$$\begin{aligned} y_t &= a + (\nu_0 + \nu_1 B + \nu_2 B^2 + \cdots + \nu_k B^k) x_t + n_t \\ &= a + \nu(B) x_t + n_t. \end{aligned}$$

- $\nu(B)$ 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.

Example: Insurance quotes and TV adverts



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:
        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
```

$$y_t = 2.04 + 1.26x_t + 0.16x_{t-1} + n_t$$

$$n_t = 1.41n_{t-1} - 0.93n_{t-2} + 0.36n_{t-3}$$

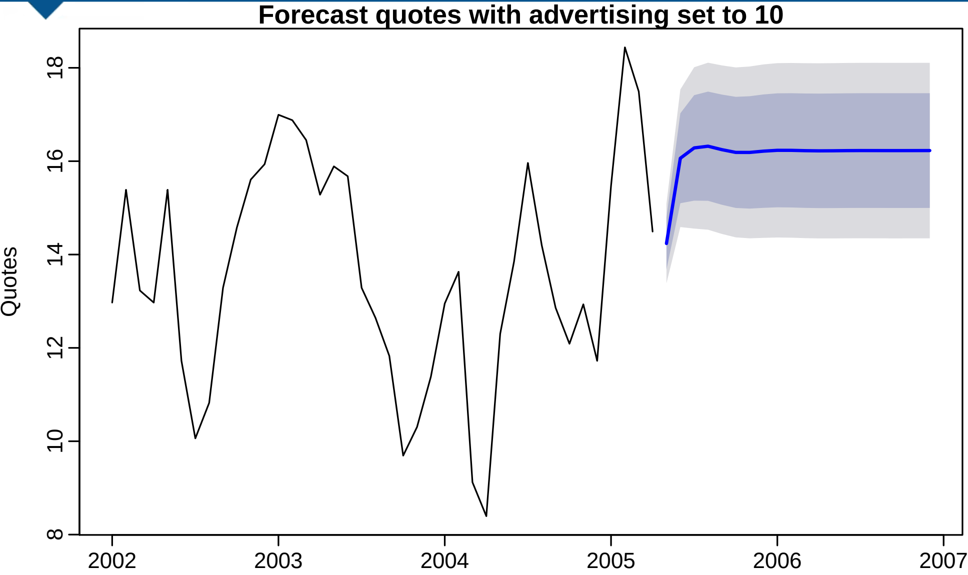
Example: Insurance quotes and TV adverts

```
> 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:
        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
```

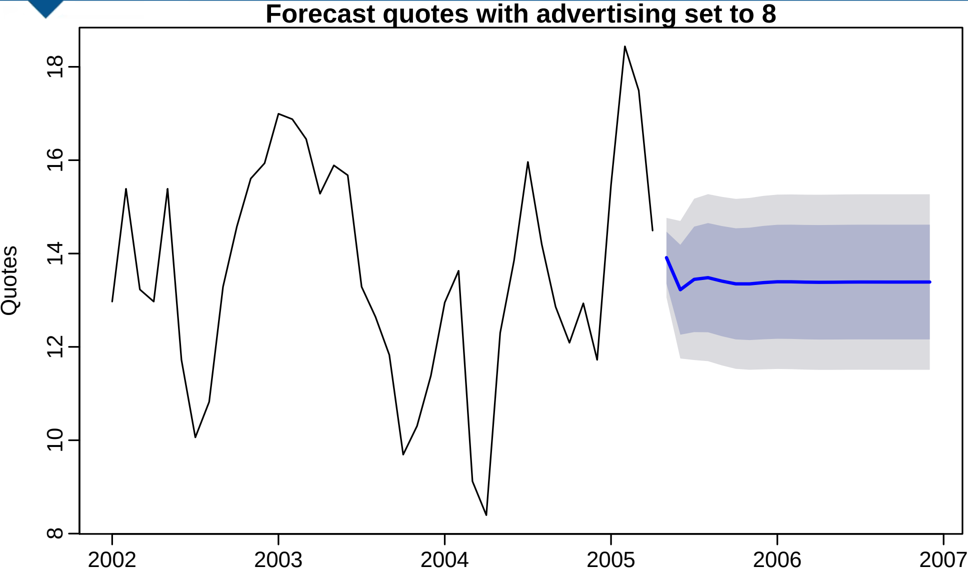
$$y_t = 2.04 + 1.26x_t + 0.16x_{t-1} + n_t$$

$$n_t = 1.41n_{t-1} - 0.93n_{t-2} + 0.36n_{t-3}$$

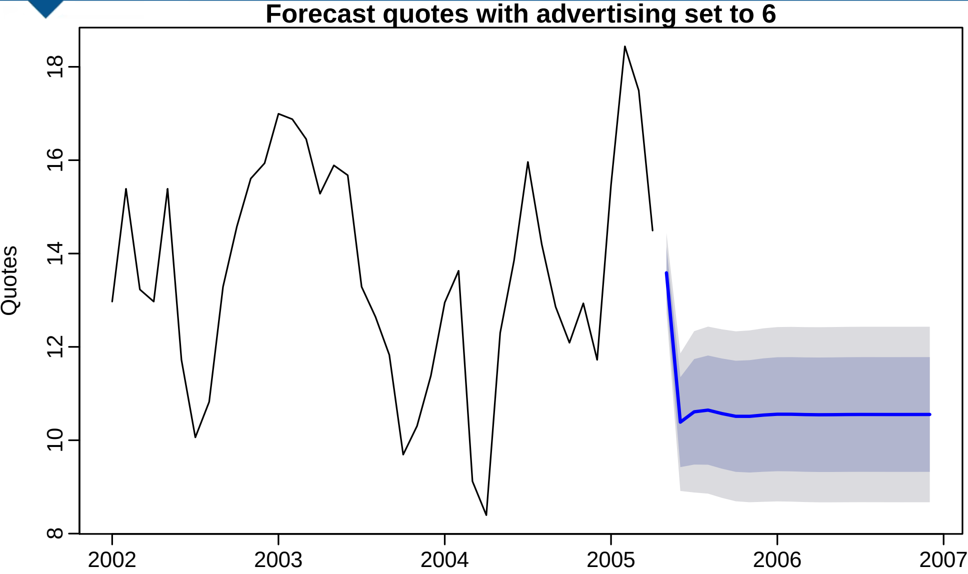
Example: Insurance quotes and TV adverts



Example: Insurance quotes and TV adverts



Example: Insurance quotes and TV adverts



Example: Insurance quotes and TV adverts

```
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

$$\phi(B)n_t = \theta(B)e_t \qquad \text{or} \qquad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t.$$

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

- 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

$$\phi(B)n_t = \theta(B)e_t \qquad \text{or} \qquad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t.$$

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

- 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

$$\phi(B)n_t = \theta(B)e_t \qquad \text{or} \qquad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t.$$

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

- 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

$$\phi(B)n_t = \theta(B)e_t \qquad \text{or} \qquad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t.$$

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

- 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

$$\phi(B)n_t = \theta(B)e_t \qquad \text{or} \qquad n_t = \frac{\theta(B)}{\phi(B)}e_t = \psi(B)e_t.$$

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

- 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.