MONASH University

MONASH BUSINESS SCHOOL

ETC3550: Applied forecasting for business and economics

Ch9. Dynamic regression models

OTexts.org/fpp2/

Outline

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 $x_{1,t}, \dots, x_{k,t}$.
- In regression, we assume that e_t was WN.
- Now we want to allow e_t to be autocorrelated.

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 $x_{1,t}, \dots, x_{k,t}$.
- In regression, we assume 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.

Residuals and errors

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

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

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

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 $\hat{\beta}_0, \dots, \hat{\beta}_k$ 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

Model with ARIMA(1,1,1) errors

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

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

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 $n'_t = n_t - n_{t-1}$.

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.

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

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

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

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

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

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

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

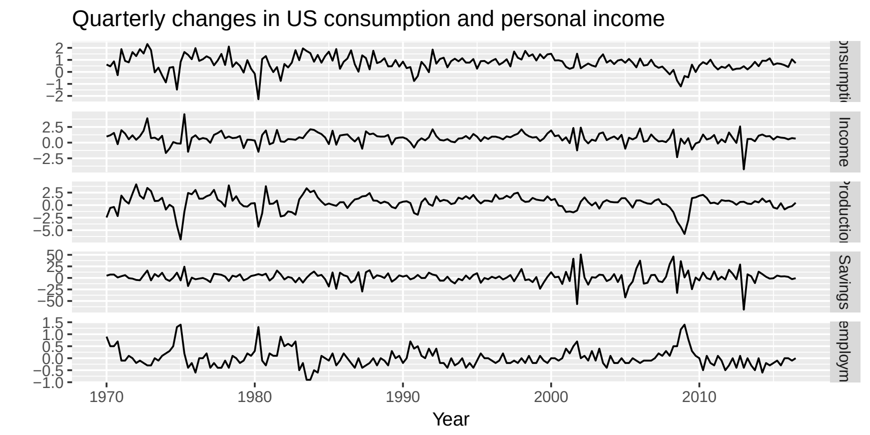
Model selection

- Check that all variables are stationary. If not, apply differencing. Where appropriate, use the same differencing for all variables to preserve interpretability.
- Fit regression model with automatically selected ARIMA errors.
- Check that e_t series looks like white noise.

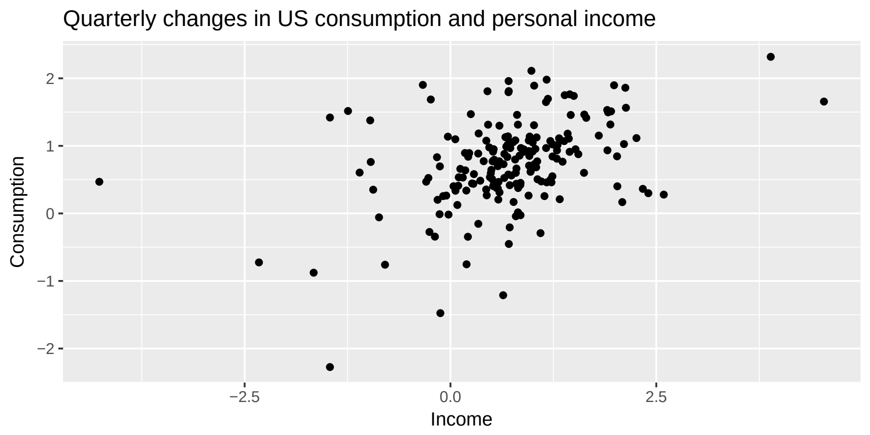
Selecting predictors

- AICc 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 and income



US personal consumption and 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.

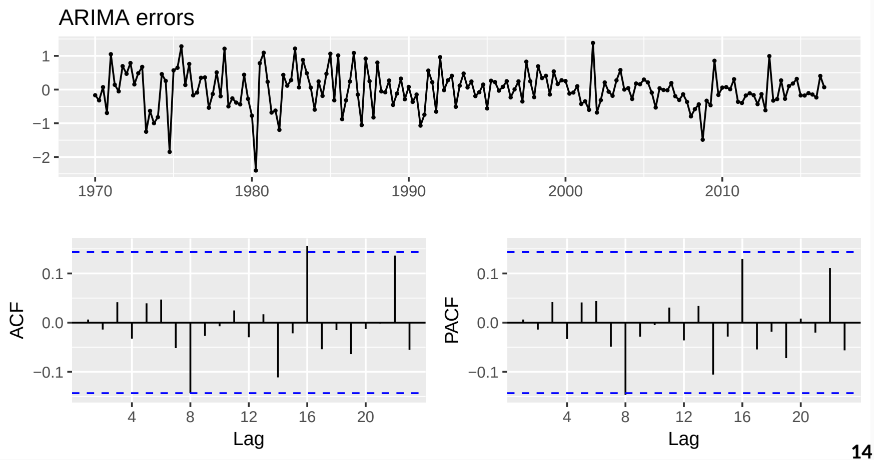
US personal consumption and income

```
(fit <- auto.arima(uschange[,1],
    xreg=uschange[,2]))
```

```
## Series: uschange[, 1]
## Regression with ARIMA(1,0,2) errors
##
## Coefficients:
##          ar1      ma1     ma2  intercept    xreg
##       0.6922  -0.5758  0.1984     0.5990  0.2028
## s.e.  0.1159   0.1301  0.0756     0.0884  0.0461
##
## sigma^2 estimated as 0.3219:  log likelihood=-156.95
## AIC=325.91   AICc=326.37   BIC=345.29
```

US personal consumption and income

```
ggtsdisplay(residuals(fit, type='response'),
  main="ARIMA errors")
```



US personal consumption and income

```r
checkresiduals(fit, test=FALSE)
```

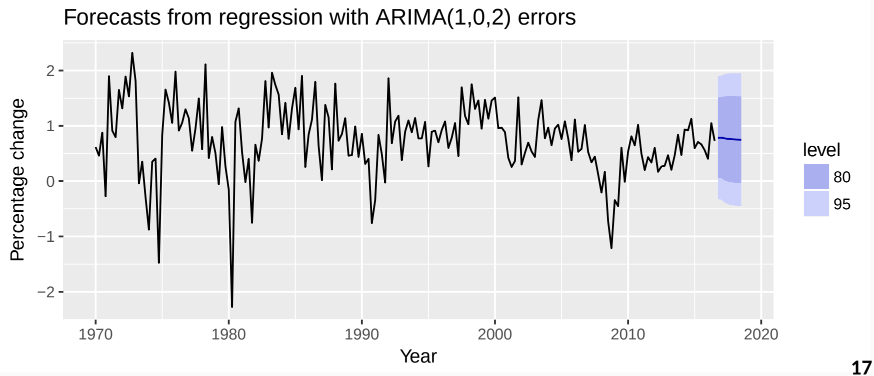
US personal consumption and income

```r
checkresiduals(fit, plot=FALSE)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,2) errors
## Q* = 5.8916, df = 3, p-value = 0.117
##
## Model df: 5.   Total lags used: 8
```

US personal consumption and income

```
fcast <- forecast(fit,
  xreg=rep(mean(uschange[,2]),8), h=8)
autoplot(fcast) + xlab("Year") +
  ylab("Percentage change") +
  ggtitle("Forecasts from regression with ARIMA(1,0,2) errors")
```



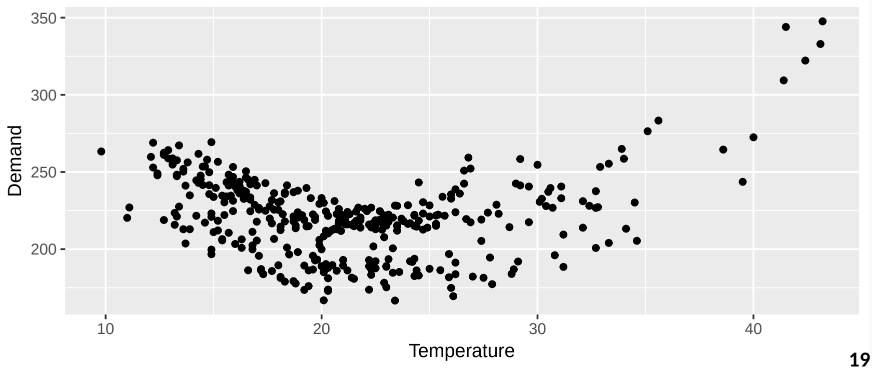
Forecasting

- 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.
  - Some explanatory variable are known into the future (e.g., time, dummies).
  - Separate forecasting models may be needed for other explanatory variables.

Daily electricity demand

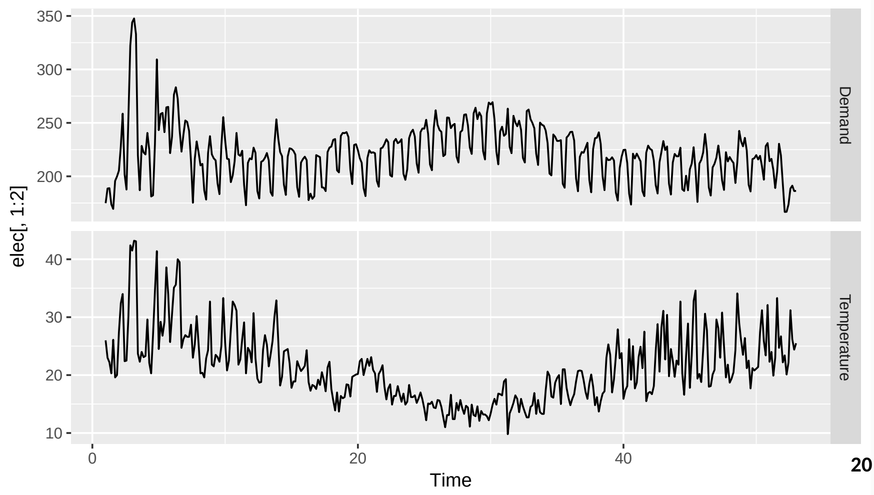
Model daily electricity demand as a function of temperature using quadratic regression with ARMA errors.

```
demand <- colSums(matrix(elecdemand[,"Demand"], nrow=48))
temp <- apply(matrix(elecdemand[,"Temperature"], nrow=48),2,max)
wday <- colMeans(matrix(elecdemand[,"WorkDay"], nrow=48))
qplot(temp, demand) + xlab("Temperature") + ylab("Demand")
```



Daily electricity demand

```
elec <- ts(cbind(Demand=demand, Temperature=temp, Workd
autoplot(elec[,1:2], facets = TRUE)
```



Daily electricity demand

```
# Matrix of regressors
xreg <- cbind(MaxTemp = elec[,"Temperature"],
              MaxTempSq = elec[,"Temperature"]^2,
              Workday = elec[,"Workday"])
# Fit model
(fit <- auto.arima(elec[,"Demand"], xreg=xreg))
```

```
## Series: elec[, "Demand"]
## Regression with ARIMA(2,1,2)(2,0,0)[7] errors
##
## Coefficients:
##              ar1     ar2      ma1      ma2    sar1    sar2    drift
##          -0.0622  0.6731  -0.0234  -0.9301  0.2012  0.4021  -0.0191
## s.e.      0.0714  0.0667   0.0413   0.0390  0.0533  0.0567   0.1091
##          xreg.MaxTemp  xreg.MaxTempSq  xreg.Workday
##               -7.4996          0.1789       30.5695
## s.e.           0.4409          0.0084        1.2891
##
```

Daily electricity demand

```r
# Forecast one day ahead
forecast(fit, xreg = cbind(20, 20^2, 1))
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 53.14286       185.4008 176.9271 193.8745 172.4414 198.3602
```

Outline

Stochastic & deterministic trends

Deterministic trend

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

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 $d \geq 1$.

Stochastic & deterministic trends

Deterministic trend

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

where n_t is ARMA process.

Stochastic trend

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

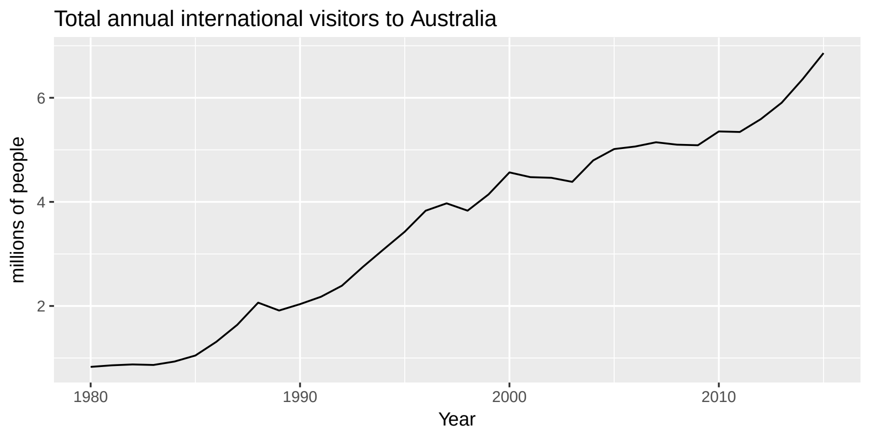
where n_t is ARIMA process with $d \geq 1$.

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

```
trend <- seq_along(austa)
(fit1 <- auto.arima(austa, d=0, xreg=trend))
```

```
## Series: austa
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1      ar2  intercept    xreg
##       1.1127  -0.3805     0.4156  0.1710
## s.e.  0.1600   0.1585     0.1897  0.0088
##
## sigma^2 estimated as 0.02979:  log likelihood=13.6
## AIC=-17.2   AICc=-15.2   BIC=-9.28
```

International visitors

Deterministic trend

```r
trend <- seq_along(austa)
(fit1 <- auto.arima(austa, d=0, xreg=trend))
```

```
## Series: austa
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1      ar2  intercept    xreg
##       1.1127  -0.3805     0.4156  0.1710
## s.e.  0.1600   0.1585     0.1897  0.0088
##
## sigma^2 estimated as 0.02979:  log likelihood=13.6
## AIC=-17.2   AICc=-15.2   BIC=-9.28
```

$$
\begin{aligned}
y_t &= 0.42 + 0.17t + n_t \\
n_t &= 1.11n_{t-1} - 0.38n_{t-2} + e_t \\
e_t &\sim \text{NID}(0, 0.0298).
\end{aligned}
$$

International visitors

Stochastic trend

```
(fit2 <- auto.arima(austa,d=1))
```

```
## Series: austa
## ARIMA(0,1,1) with drift
##
## Coefficients:
##          ma1   drift
##       0.3006  0.1735
## s.e.  0.1647  0.0390
##
## sigma^2 estimated as 0.03376:  log likelihood=10.62
## AIC=-15.24   AICc=-14.46   BIC=-10.57
```

International visitors

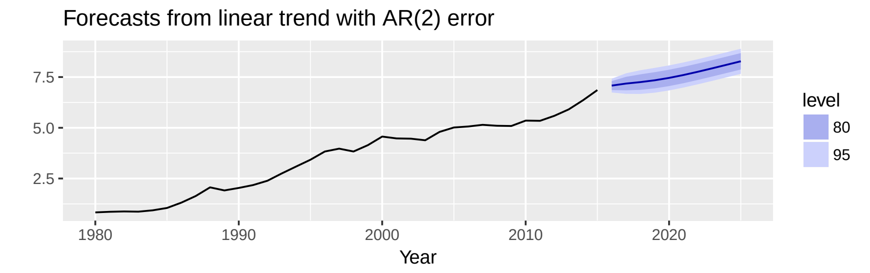
Stochastic trend

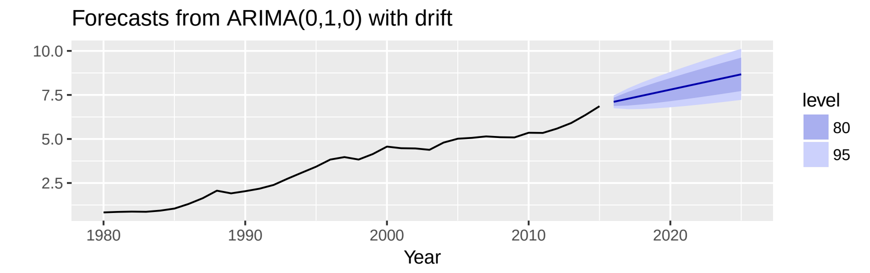
```r
(fit2 <- auto.arima(austa,d=1))
```

```
## Series: austa
## ARIMA(0,1,1) with drift
##
## Coefficients:
##          ma1   drift
##       0.3006  0.1735
## s.e.  0.1647  0.0390
##
## sigma^2 estimated as 0.03376:  log likelihood=10.62
## AIC=-15.24   AICc=-14.46   BIC=-10.57
```

$$
\begin{aligned}
y_t - y_{t-1} &= 0.17 + e_t \\
y_t &= y_0 + 0.17t + n_t \\
n_t &= n_{t-1} + 0.30e_{t-1} + e_t \\
e_t &\sim \text{NID}(0, 0.0338).
\end{aligned}
$$

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.

Outline

Dynamic harmonic regression

Combine Fourier terms with ARIMA errors

Advantages

- it allows any length seasonality;
- for data with more than one seasonal period, you can include Fourier terms of different frequencies;
- the seasonal pattern is smooth for small values of K (but more wiggly seasonality can be handled by increasing K);
- the short-term dynamics are easily handled with a simple ARMA error.

Disadvantages

- seasonality is assumed to be fixed

Example: weekly gasoline products

```r
harmonics <- fourier(gasoline, K = 13)
(fit <- auto.arima(gasoline, xreg = harmonics, seasonal = FALSE))
```

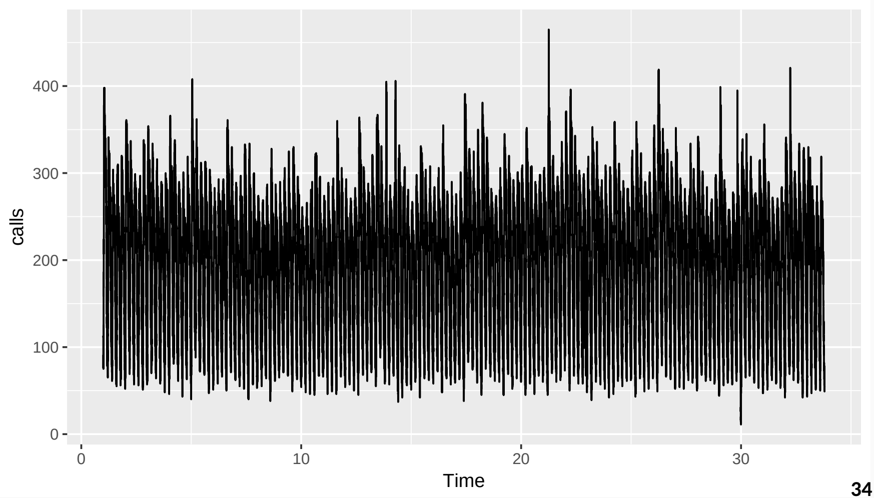
```
## Series: gasoline
## Regression with ARIMA(0,1,2) errors
##
## Coefficients:
##           ma1     ma2   drift   S1-52    C1-52    S2-52    C2-52   S3-52
##       -0.9612  0.0935  0.0014  0.0315  -0.2555  -0.0522  -0.0176  0.0242
## s.e.   0.0275  0.0286  0.0008  0.0124   0.0124   0.0090   0.0089  0.0082
##         C3-52   S4-52    C4-52    S5-52    C5-52   S6-52    C6-52   S7-52
##       -0.0989  0.0321  -0.0257  -0.0011  -0.0472  0.0580  -0.0320  0.0283
## s.e.   0.0082  0.0079   0.0079   0.0078   0.0078  0.0078   0.0078  0.0079
##        C7-52   S8-52   C8-52    S9-52   C9-52   S10-52  C10-52  S11-52
##       0.0369  0.0238  0.0139  -0.0172  0.0119  -0.0236  0.0230  0.0001
## s.e.  0.0079  0.0079  0.0079   0.0080  0.0080   0.0081  0.0081  0.0082
##        C11-52   S12-52   C12-52  S13-52   C13-52
##       -0.0191  -0.0288  -0.0177  0.0012  -0.0175
## s.e.   0.0082   0.0083   0.0083  0.0084   0.0084
##
## sigma^2 estimated as 0.05603:  log likelihood=43.66
## AIC=-27.33   AICc=-25.92   BIC=129
```

Example: weekly gasoline products

```r
newharmonics <- fourier(gasoline, K = 13, h = 156)
fc <- forecast(fit, xreg = newharmonics)
autoplot(fc)
```

5-minute call centre volume

```r
autoplot(calls)
```



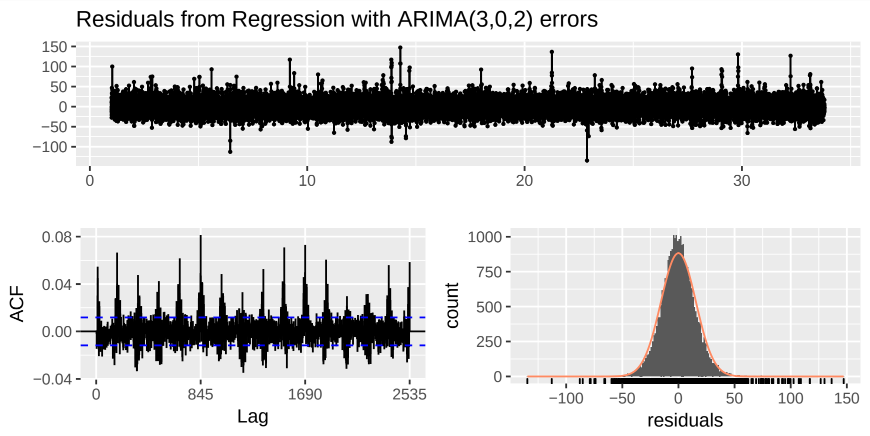
5-minute call centre volume

```r
xreg <- fourier(calls, K = c(10,0))
(fit <- auto.arima(calls, xreg=xreg, seasonal=FALSE, stationary=TRUE))
```

```
## Series: calls
## Regression with ARIMA(3,0,2) errors
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2  intercept   S1-169
##       0.8406  0.1919  -0.0442  -0.5896  -0.1891   192.0697  55.2447
## s.e.  0.1692  0.1782   0.0129   0.1693   0.1369     1.7638   0.7013
##         C1-169   S2-169    C2-169    S3-169   C3-169   S4-169   C4-169
##       -79.0871  13.6738  -32.3747  -13.6934  -9.3270  -9.5318  -2.7972
## s.e.    0.7007   0.3788    0.3787    0.2727   0.2726   0.2230   0.2230
##        S5-169  C5-169  S6-169  C6-169  S7-169  C7-169  S8-169   C8-169
##       -2.2393  2.8934  0.1730  3.3052  0.8552  0.2935  0.8575  -1.3913
## s.e.   0.1956  0.1956  0.1788  0.1788  0.1678  0.1678  0.1602   0.1601
##        S9-169   C9-169  S10-169  C10-169
##       -0.9864  -0.3448  -1.1964   0.8010
## s.e.   0.1546   0.1546   0.1504   0.1504
##
## sigma^2 estimated as 242.5:  log likelihood=-115411.5
## AIC=230877   AICc=230877.1   BIC=231099.3
```

5-minute call centre volume

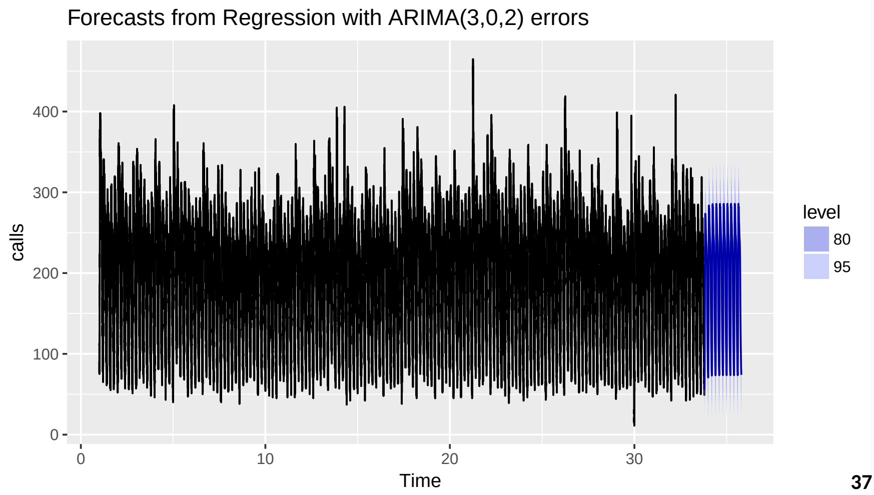
```r
checkresiduals(fit)
```




```
##
## Ljung-Box test
```

5-minute call centre volume

```
fc <- forecast(fit, xreg = fourier(calls, c(10,0), 1690))
autoplot(fc)
```



Outline

Lagged predictors

Sometimes a change in x_t does not affect y_t instantaneously

Lagged predictors

Sometimes a change in x_t does not affect y_t instantaneously

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

Lagged predictors

Sometimes a change in x_t does not affect y_t instantaneously

- y_t = sales, x_t = advertising.
- y_t = stream flow, x_t = rainfall.
- 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 predictors

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.

Lagged predictors

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

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

Lagged predictors

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

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

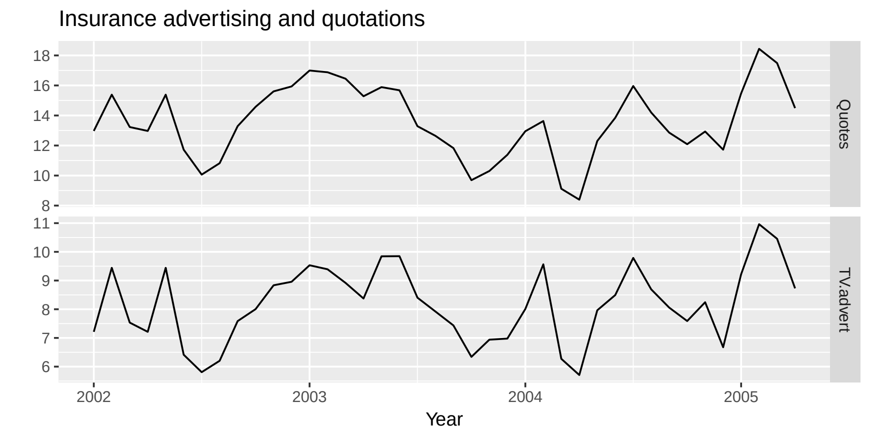
where n_t is an ARIMA process.

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

```r
Advert <- cbind(insurance[,2], c(NA,insurance[1:39,2]))
colnames(Advert) <- paste("AdLag",0:1,sep="")
(fit <- auto.arima(insurance[,1], xreg=Advert, d=0))
```

```
## Series: insurance[, 1]
## Regression with ARIMA(3,0,0) errors
##
## 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 0.2165:  log likelihood=-23.89
## AIC=61.78   AICc=65.28   BIC=73.6
```

Example: Insurance quotes and TV adverts

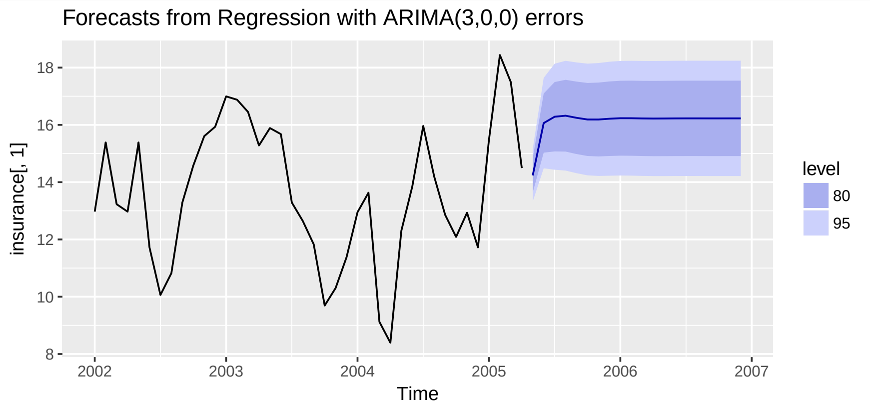
```r
Advert <- cbind(insurance[,2], c(NA,insurance[1:39,2]))
colnames(Advert) <- paste("AdLag",0:1,sep="")
(fit <- auto.arima(insurance[,1], xreg=Advert, d=0))
```

```
## Series: insurance[, 1]
## Regression with ARIMA(3,0,0) errors
##
## 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 0.2165:  log likelihood=-23.89
## AIC=61.78   AICc=65.28   BIC=73.6
```

$$y_t = 2.05 + 1.26x_t + 0.16x_{t-1} + n_t$$
$$n_t = 1.41n_{t-1} - 093n_{t-2} + 0.36n_{t-3} + e_t$$

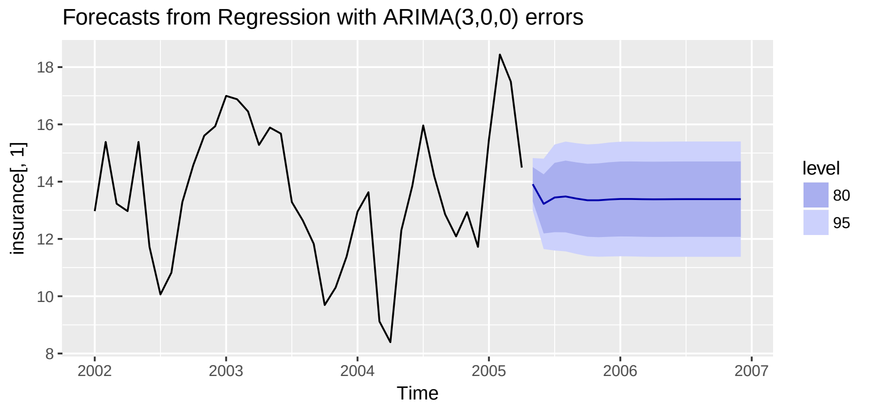
Example: Insurance quotes and TV adverts

```r
fc <- forecast(fit, h=20,
  xreg=cbind(c(Advert[40,1],rep(10,19)), rep(10,20)))
autoplot(fc)
```



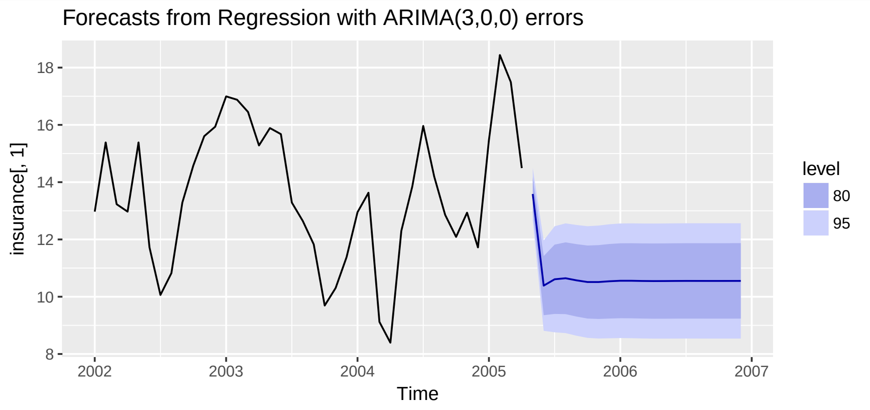
Example: Insurance quotes and TV adverts

```r
fc <- forecast(fit, h=20,
  xreg=cbind(c(Advert[40,1],rep(8,19)), rep(8,20)))
autoplot(fc)
```



Example: Insurance quotes and TV adverts

```r
fc <- forecast(fit, h=20,
  xreg=cbind(c(Advert[40,1],rep(6,19)), rep(6,20)))
autoplot(fc)
```



Transfer function models

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

where n_t is an ARMA process. So

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

Transfer function models

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

where n_t is an ARMA process. So

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

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

Transfer function 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.