

# ETC3550/ETC5550

## Applied forecasting

Ch10. Dynamic regression models

[OTexts.org/fpp3/](https://OTexts.org/fpp3/)

# Outline

- 1 Regression with ARIMA errors
- 2 Stochastic and deterministic trends
- 3 Dynamic harmonic regression
- 4 Lagged predictors

# Outline

- 1 Regression with ARIMA errors
- 2 Stochastic and deterministic trends
- 3 Dynamic harmonic regression
- 4 Lagged predictors

# Regression with ARIMA errors

## Regression models

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

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

# Regression with ARIMA errors

## Regression models

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

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

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

$$y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \eta_t,$$
$$(1 - \phi_1 B)(1 - B)\eta_t = (1 + \theta_1 B)\varepsilon_t,$$

where  $\varepsilon_t$  is white noise.

# Residuals and errors

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

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

$$(1 - \phi_1 B)(1 - B)\eta_t = (1 + \theta_1 B)\varepsilon_t,$$

# Residuals and errors

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

$$y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \eta_t,$$
$$(1 - \phi_1 B)(1 - B)\eta_t = (1 + \theta_1 B)\varepsilon_t,$$

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

# Estimation

If we minimize  $\sum \eta_t^2$  (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  $\sum \eta_t^2$  (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  $\sum \varepsilon_t^2$  avoids these problems.
  - Maximizing likelihood similar to minimizing  $\sum \varepsilon_t^2$ .

# 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

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

$$\text{where } \phi(B)(1-B)^d \eta_t = \theta(B)\varepsilon_t$$

# 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

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

$$\text{where } \phi(B)(1-B)^d \eta_t = \theta(B) \varepsilon_t$$

## After differencing all variables

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

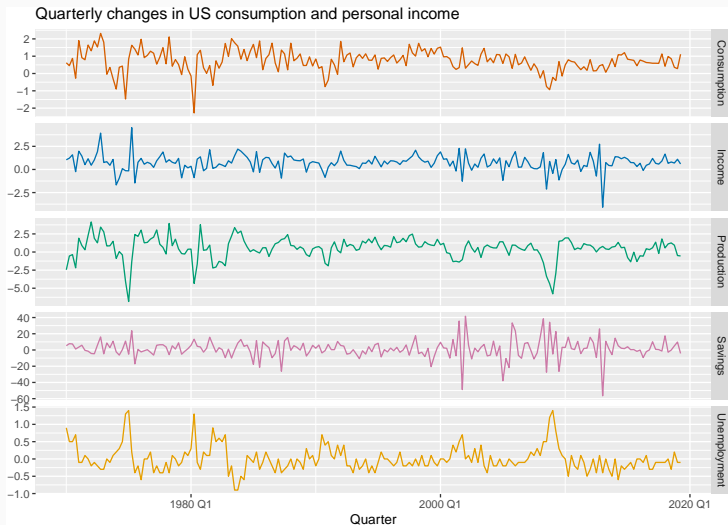
$$\text{where } \phi(B) \eta_t = \theta(B) \varepsilon_t$$

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

# Estimation and stationarity

- All variables in the model must be stationary before estimation.
- 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.
- In R, we can specify an ARIMA( $p, d, q$ ) for the errors, and  $d$  levels of differencing will be applied to all variables ( $y, x_{1,t}, \dots, x_{k,t}$ ).
- Check that  $\varepsilon_t$  series looks like white noise.
- AICc can be calculated for final model.
- Repeat procedure for all subsets of predictors to be considered, and select model with lowest AICc value.

# US personal consumption and income



# US personal consumption and income

```
fit <- us_change %>% model(ARIMA(Consumption ~ Income))  
report(fit)
```

```
## Series: Consumption  
## Model: LM w/ ARIMA(1,0,2) errors  
##  
## Coefficients:  
##          ar1      ma1      ma2  Income  intercept  
##          0.707  -0.617   0.2066  0.1976      0.595  
## s.e.    0.107    0.122   0.0741  0.0462      0.085  
##  
## sigma^2 estimated as 0.3113:  log likelihood=-163  
## AIC=338    AICc=339    BIC=358
```

# US personal consumption and income

```
fit <- us_change %>% model(ARIMA(Consumption ~ Income))  
report(fit)
```

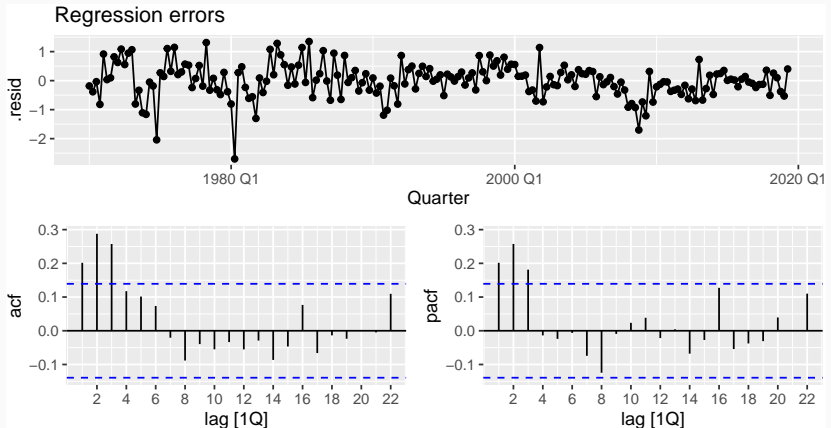
```
## Series: Consumption  
## Model: LM w/ ARIMA(1,0,2) errors  
##  
## Coefficients:  
##          ar1      ma1      ma2  Income  intercept  
##          0.707  -0.617   0.2066  0.1976         0.595  
## s.e.    0.107    0.122   0.0741  0.0462         0.085  
##  
## sigma^2 estimated as 0.3113:  log likelihood=-163  
## AIC=338   AICc=339   BIC=358
```

Write down the equations for the fitted model.



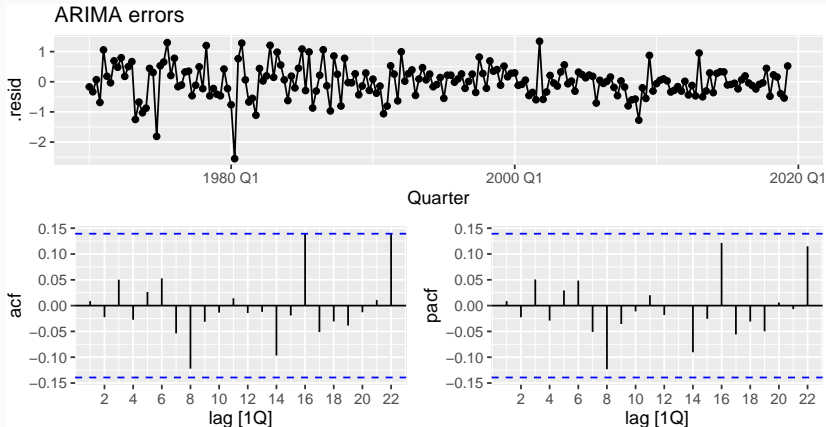
# US personal consumption and income

```
residuals(fit, type='regression') %>%  
  gg_tsdisplay(.resid, plot_type = 'partial') +  
  labs(title = "Regression errors")
```



# US personal consumption and income

```
residuals(fit, type='response') %>%  
  gg_tsdisplay(.resid, plot_type = 'partial') +  
  labs(title = "ARIMA errors")
```



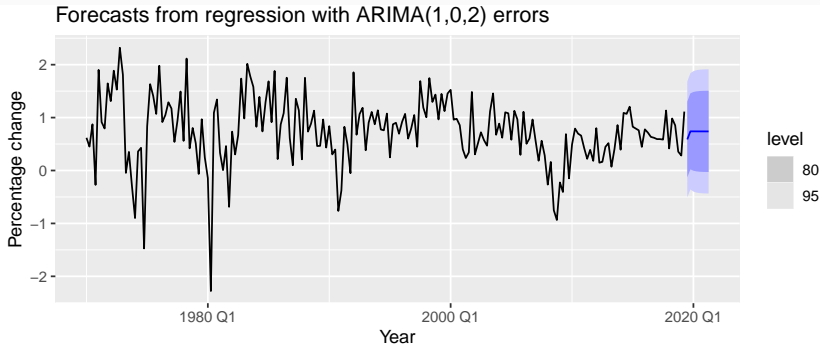
# US personal consumption and income

```
augment(fit) %>%  
  features(.resid, ljung_box, dof = 5, lag = 12)
```

```
## # A tibble: 1 x 3  
##   .model                lb_stat lb_pvalue  
##   <chr>                <dbl>    <dbl>  
## 1 ARIMA(Consumption ~ Income)  5.54    0.595
```

# US personal consumption and income

```
us_change_future <- new_data(us_change, 8) %>%  
  mutate(Income = mean(us_change$Income))  
forecast(fit, new_data = us_change_future) %>%  
  autoplot(us_change) +  
  labs(x = "Year", y = "Percentage change",  
       title = "Forecasts from regression with ARIMA(1,0,2) errors")
```



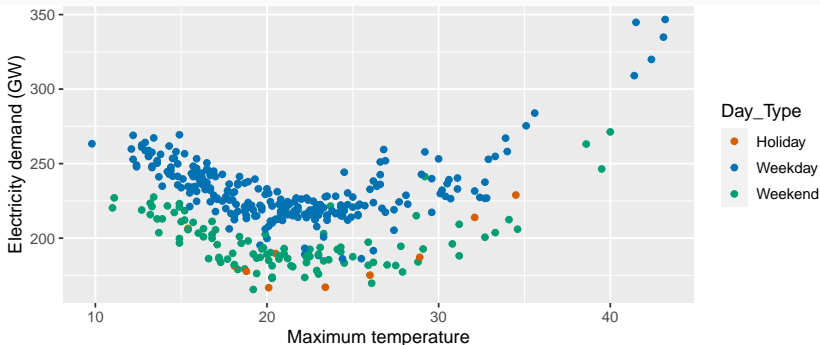
# Forecasting

- 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 predictors are known into the future (e.g., time, dummies).
- Separate forecasting models may be needed for other predictors.
- Forecast intervals ignore the uncertainty in forecasting the predictors.

# Daily electricity demand

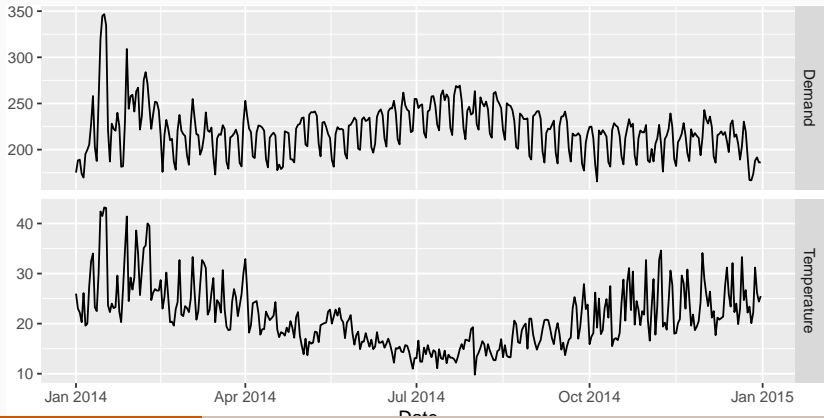
Model daily electricity demand as a function of temperature using quadratic regression with ARMA errors.

```
vic_elec_daily %>%  
  ggplot(aes(x = Temperature, y = Demand, colour = Day_Type)) +  
  geom_point() +  
  labs(x = "Maximum temperature", y = "Electricity demand (GW)")
```



# Daily electricity demand

```
vic_elec_daily %>%  
  pivot_longer(c(Demand, Temperature)) %>%  
  ggplot(aes(x = Date, y = value)) + geom_line() +  
  facet_grid(name ~ ., scales = "free_y") + ylab("")
```



# Daily electricity demand

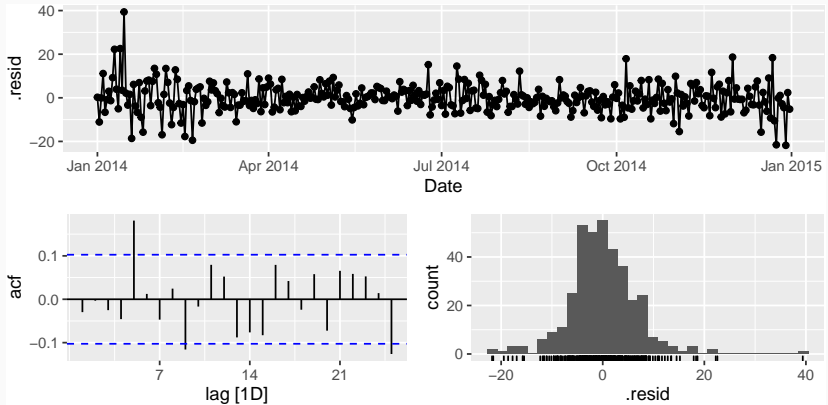
```
fit <- vic_elec_daily %>%  
  model(ARIMA(Demand ~ Temperature + I(Temperature^2) +  
             (Day_Type=="Weekday")))  
report(fit)
```

```
## Series: Demand  
## Model: LM w/ ARIMA(2,1,2)(2,0,0)[7] errors  
##  
## Coefficients:  
##          ar1      ar2      ma1      ma2      sar1      sar2  
##      -0.1093  0.7226 -0.0182 -0.9381  0.1958  0.417  
## s.e.   0.0779  0.0739  0.0494  0.0493  0.0525  0.057  
##      Temperature  I(Temperature^2)  
##          -7.614           0.1810  
## s.e.         0.448           0.0085  
##      Day_Type == "Weekday"TRUE  
##                      30.40  
## s.e.                      1.33  
##
```



# Daily electricity demand

```
gg_tsresiduals(fit)
```



# Daily electricity demand

```
augment(fit) %>%  
  features(.resid, ljung_box, dof = 9, lag = 14)
```

```
## # A tibble: 1 x 3  
##   .model                                lb_stat lb_pvalue  
##   <chr>                                <dbl>     <dbl>  
## 1 "ARIMA(Demand ~ Temperature + I(Tempera~ 28.4 0.0000304
```

# Daily electricity demand

```
# Forecast one day ahead
```

```
vic_next_day <- new_data(vic_elec_daily, 1) %>%  
  mutate(Temperature = 26, Day_Type = "Holiday")  
forecast(fit, vic_next_day)
```

```
## # A tibble: 1 x 6 [1D]
```

```
## # Key:       .model [1]
```

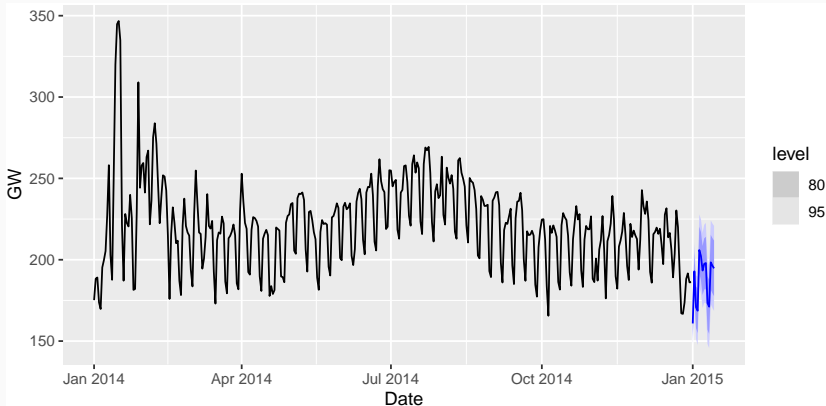
##	.model	Date	Demand	.mean	Temperature	Day_Type
##	<chr>	<date>	<dist>	<dbl>	<dbl>	<chr>
## 1	"ARIMA(D~	2015-01-01	N(161, 45)	161.	26	Holiday

# Daily electricity demand

```
vic_elec_future <- new_data(vic_elec_daily, 14) %>%  
  mutate(  
    Temperature = 26,  
    Holiday = c(TRUE, rep(FALSE, 13)),  
    Day_Type = case_when(  
      Holiday ~ "Holiday",  
      wday(Date) %in% 2:6 ~ "Weekday",  
      TRUE ~ "Weekend"  
    )  
  )
```

# Daily electricity demand

```
forecast(fit, new_data = vic_elec_future) %>%  
  autoplot(vic_elec_daily) + labs(y="GW")
```



# Outline

- 1 Regression with ARIMA errors
- 2 Stochastic and deterministic trends
- 3 Dynamic harmonic regression
- 4 Lagged predictors

# Stochastic & deterministic trends

## Deterministic trend

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

where  $\eta_t$  is ARMA process.

# Stochastic & deterministic trends

## Deterministic trend

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

where  $\eta_t$  is ARMA process.

## Stochastic trend

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

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



# Stochastic & deterministic trends

## Deterministic trend

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

where  $\eta_t$  is ARMA process.

## Stochastic trend

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

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

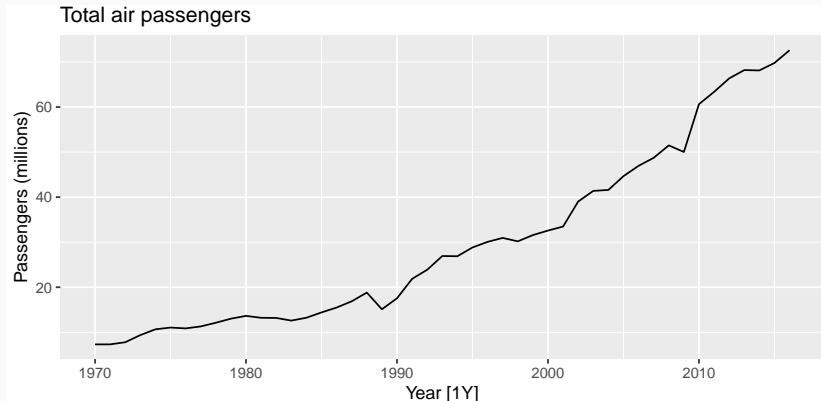
Difference both sides until  $\eta_t$  is stationary:

$$y'_t = \beta_1 + \eta'_t$$

where  $\eta'_t$  is ARMA process.

# Air transport passengers Australia

```
aus_airpassengers %>%  
  autoplot(Passengers) +  
  labs(y = "Passengers (millions)",  
       title = "Total air passengers")
```



# Air transport passengers Australia

## Deterministic trend

```
fit_deterministic <- aus_airpassengers %>%  
  model(ARIMA(Passengers ~ 1 + trend() + pdq(d = 0)))  
report(fit_deterministic)
```

```
## Series: Passengers  
## Model: LM w/ ARIMA(1,0,0) errors  
##  
## Coefficients:  
##          ar1  trend()  intercept  
##      0.9564    1.415    0.901  
## s.e. 0.0362    0.197    7.075  
##  
## sigma^2 estimated as 4.343:  log likelihood=-101  
## AIC=210    AICc=211    BIC=217
```

# Air transport passengers Australia

## Deterministic trend

```
fit_deterministic <- aus_airpassengers %>%  
  model(ARIMA(Passengers ~ 1 + trend() + pdq(d = 0)))  
report(fit_deterministic)
```

```
## Series: Passengers  
## Model: LM w/ ARIMA(1,0,0) errors  
##  
## Coefficients:  
##           ar1  trend()  intercept  
##      0.9564    1.415    0.901  
## s.e. 0.0362    0.197    7.075  
##  
## sigma^2 estimated as 4.343:  log likelihood=-101  
## AIC=210   AICc=211   BIC=217
```

$$y_t = 0.901 + 1.415t + \eta_t$$

$$\eta_t = 0.956\eta_{t-1} + \varepsilon_t$$

$$\varepsilon_t \sim \text{NID}(0, 4.343).$$

# Air transport passengers Australia

## Stochastic trend

```
fit_stochastic <- aus_airpassengers %>%  
  model(ARIMA(Passengers ~ pdq(d = 1)))  
report(fit_stochastic)
```

```
## Series: Passengers  
## Model: ARIMA(0,1,0) w/ drift  
##  
## Coefficients:  
##      constant  
##      1.419  
## s.e.      0.301  
##  
## sigma^2 estimated as 4.271:  log likelihood=-98.2  
## AIC=200    AICc=201    BIC=204
```

# Air transport passengers Australia

## Stochastic trend

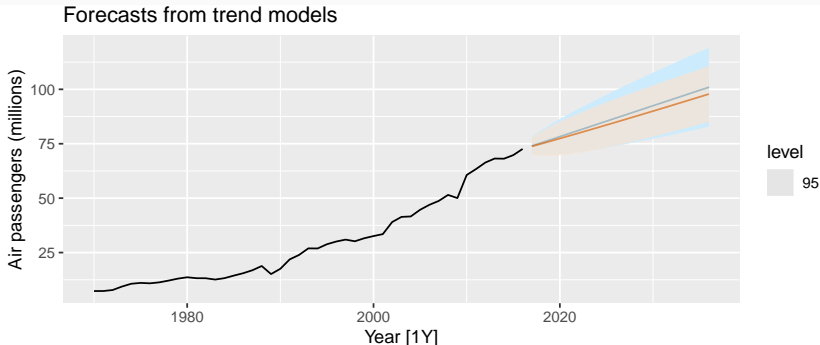
```
fit_stochastic <- aus_airpassengers %>%  
  model(ARIMA(Passengers ~ pdq(d = 1)))  
report(fit_stochastic)
```

```
## Series: Passengers  
## Model: ARIMA(0,1,0) w/ drift  
##  
## Coefficients:  
##      constant  
##      1.419  
## s.e.      0.301  
##  
## sigma^2 estimated as 4.271:  log likelihood=-98.2  
## AIC=200   AICc=201   BIC=204
```

$$y_t - y_{t-1} = 1.419 + \varepsilon_t,$$
$$y_t = y_0 + 1.419t + \eta_t$$
$$\eta_t = \eta_{t-1} + \varepsilon_t$$

# Air transport passengers Australia

```
aus_airpassengers %>%  
  autoplot(Passengers) +  
  autolayer(fit_stochastic %>% forecast(h = 20),  
    colour = "#0072B2", level = 95) +  
  autolayer(fit_deterministic %>% forecast(h = 20),  
    colour = "#D55E00", alpha = 0.65, level = 95) +  
  labs(y = "Air passengers (millions)",  
    title = "Forecasts from trend models")
```



# 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

- 1 Regression with ARIMA errors
- 2 Stochastic and deterministic trends
- 3 Dynamic harmonic regression
- 4 Lagged predictors

# Dynamic harmonic regression

## Combine Fourier terms with ARIMA errors

### Advantages

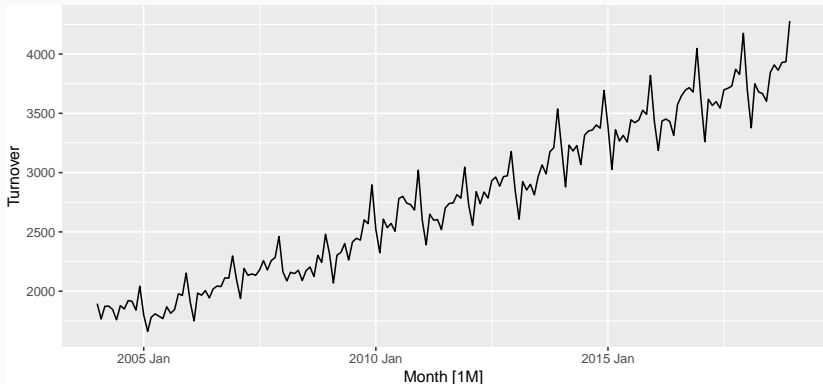
- it allows any length seasonality;
- for data with more than one seasonal period, you can include Fourier terms of different frequencies;
- the seasonal pattern is smooth for small values of  $K$  (but more wiggly seasonality can be handled by increasing  $K$ );
- the short-term dynamics are easily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

### Disadvantages

- seasonality is assumed to be fixed

# Eating-out expenditure

```
aus_cafe <- aus_retail %>% filter(  
  Industry == "Cafes, restaurants and takeaway food services",  
  year(Month) %in% 2004:2018  
) %>% summarise(Turnover = sum(Turnover))  
aus_cafe %>% autoplot(Turnover)
```

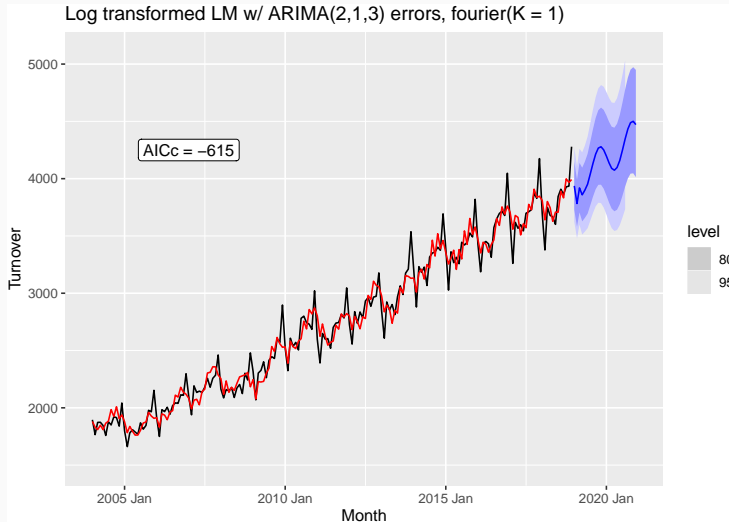


# Eating-out expenditure

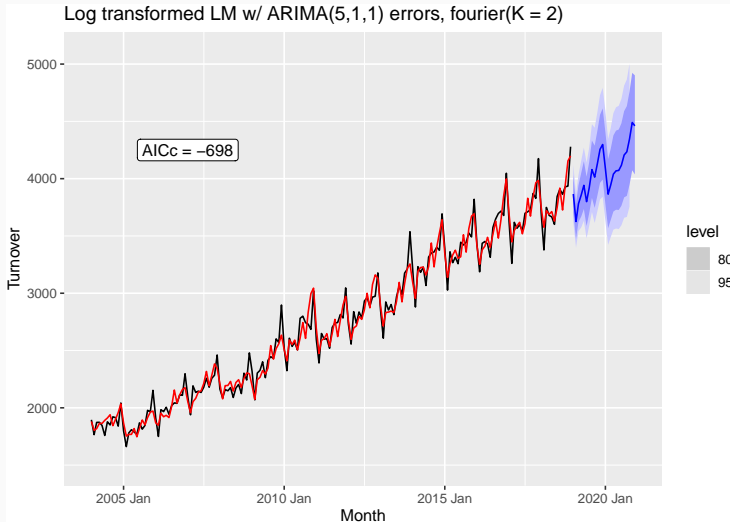
```
fit <- aus_cafe %>% model(  
  `K = 1` = ARIMA(log(Turnover) ~ fourier(K = 1) + PDQ(0,0,0)),  
  `K = 2` = ARIMA(log(Turnover) ~ fourier(K = 2) + PDQ(0,0,0)),  
  `K = 3` = ARIMA(log(Turnover) ~ fourier(K = 3) + PDQ(0,0,0)),  
  `K = 4` = ARIMA(log(Turnover) ~ fourier(K = 4) + PDQ(0,0,0)),  
  `K = 5` = ARIMA(log(Turnover) ~ fourier(K = 5) + PDQ(0,0,0)),  
  `K = 6` = ARIMA(log(Turnover) ~ fourier(K = 6) + PDQ(0,0,0)))  
glance(fit)
```

.model	sigma2	log_lik	AIC	AICc	BIC
K = 1	0.002	317	-616	-615	-588
K = 2	0.001	362	-700	-698	-661
K = 3	0.001	394	-763	-761	-725
K = 4	0.001	427	-822	-818	-771
K = 5	0.000	474	-919	-917	-875
K = 6	0.000	474	-920	-918	-875

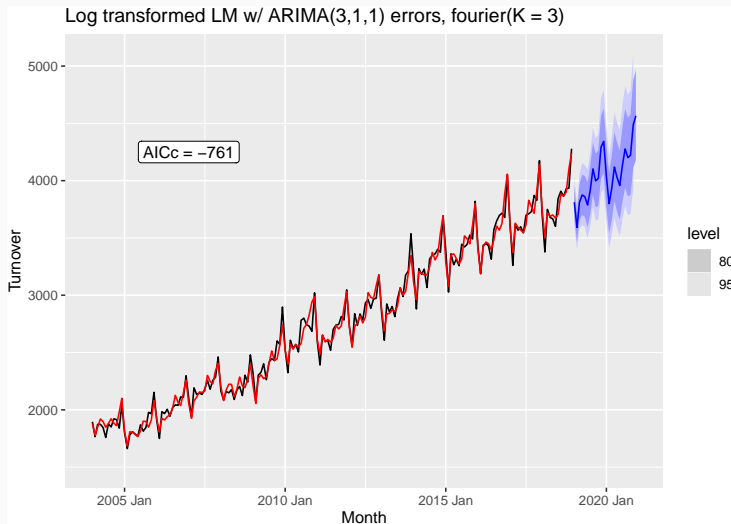
# Eating-out expenditure



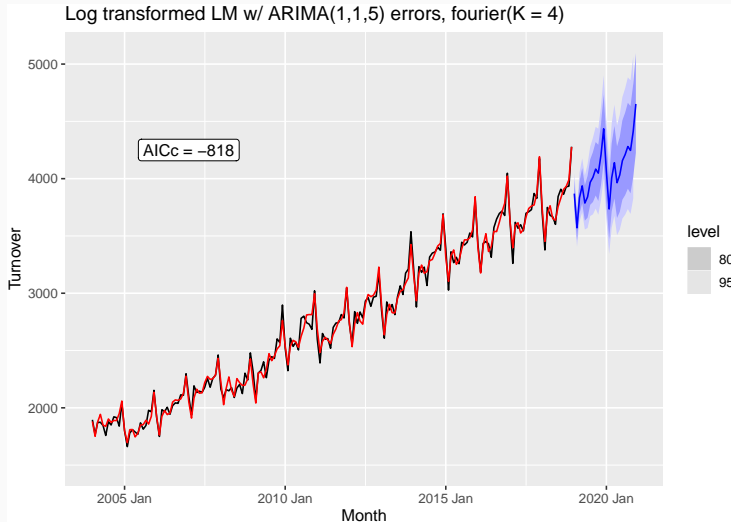
# Eating-out expenditure



# Eating-out expenditure

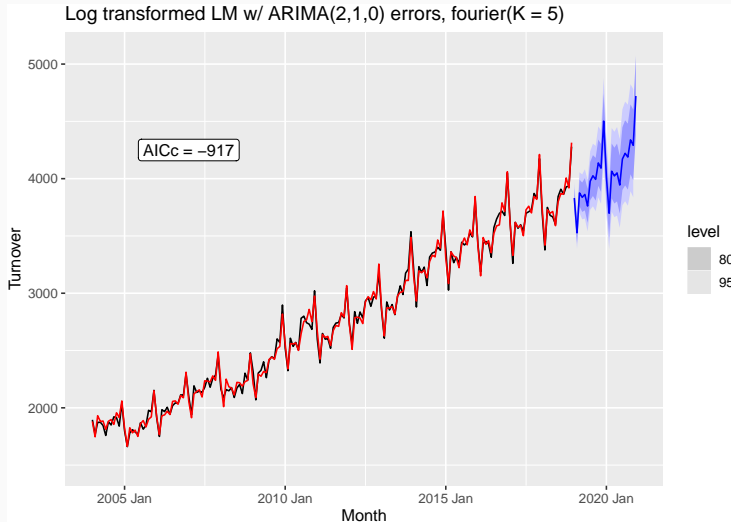


# Eating-out expenditure

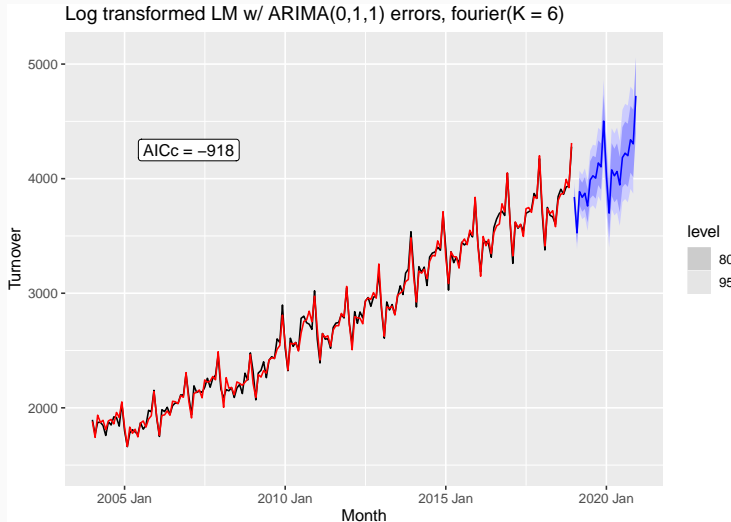




# Eating-out expenditure



# Eating-out expenditure



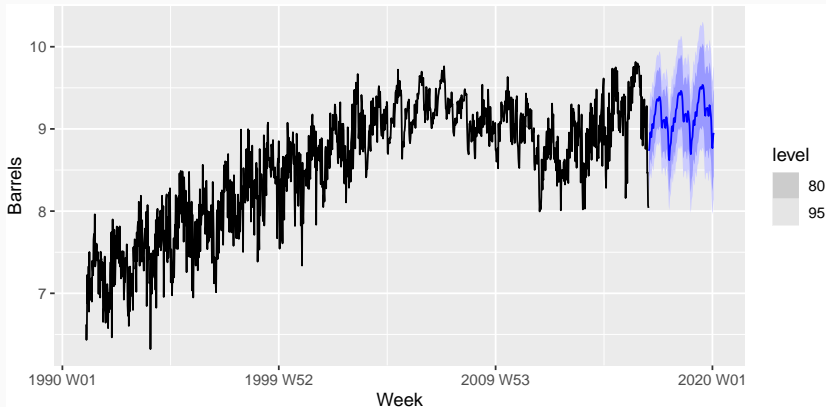
# Example: weekly gasoline products

```
fit <- us_gasoline %>%  
  model(ARIMA(Barrels ~ fourier(K = 13) + PDQ(0,0,0)))  
report(fit)
```

```
## Series: Barrels  
## Model: LM w/ ARIMA(0,1,1) errors  
##  
## Coefficients:  
##          ma1  fourier(K = 13)C1_52  fourier(K = 13)S1_52  
##          -0.8934                -0.1121                -0.2300  
## s.e.       0.0132                0.0123                0.0122  
##          fourier(K = 13)C2_52  fourier(K = 13)S2_52  
##                      0.0420                0.0317  
## s.e.              0.0099                0.0099  
##          fourier(K = 13)C3_52  fourier(K = 13)S3_52  
##                      0.0832                0.0346  
## s.e.              0.0094                0.0094  
##          fourier(K = 13)C4_52  fourier(K = 13)S4_52  
##                      0.0185                0.0398  
## s.e.              0.0092                0.0092  
##          fourier(K = 13)C5_52  fourier(K = 13)S5_52  
##                      -0.0315                0.0009  
## s.e.              0.0091                0.0091  
##          fourier(K = 13)C6_52  fourier(K = 13)S6_52
```

# Example: weekly gasoline products

```
forecast(fit, h = "3 years") %>%  
  autoplot(us_gasoline)
```



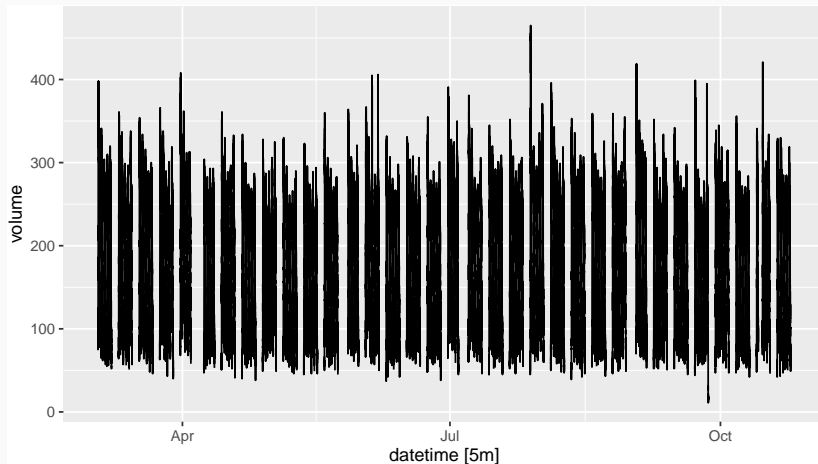
# 5-minute call centre volume

```
(calls <- readr::read_tsv("http://robjhyndman.com/data/callcenter.txt")) %>%  
  rename(time = X1) %>%  
  pivot_longer(-time, names_to = "date", values_to = "volume") %>%  
  mutate(  
    date = as.Date(date, format = "%d/%m/%Y"),  
    datetime = as_datetime(date) + time  
  ) %>%  
  as_tsibble(index = datetime))
```

```
## # A tsibble: 27,716 x 4 [5m] <UTC>  
##   time   date      volume datetime  
##   <time> <date>      <dbl> <dtm>  
## 1 07:00 2003-03-03      111 2003-03-03 07:00:00  
## 2 07:05 2003-03-03      113 2003-03-03 07:05:00  
## 3 07:10 2003-03-03       76 2003-03-03 07:10:00  
## 4 07:15 2003-03-03       82 2003-03-03 07:15:00  
## 5 07:20 2003-03-03       91 2003-03-03 07:20:00  
## 6 07:25 2003-03-03       87 2003-03-03 07:25:00  
## 7 07:30 2003-03-03       75 2003-03-03 07:30:00
```

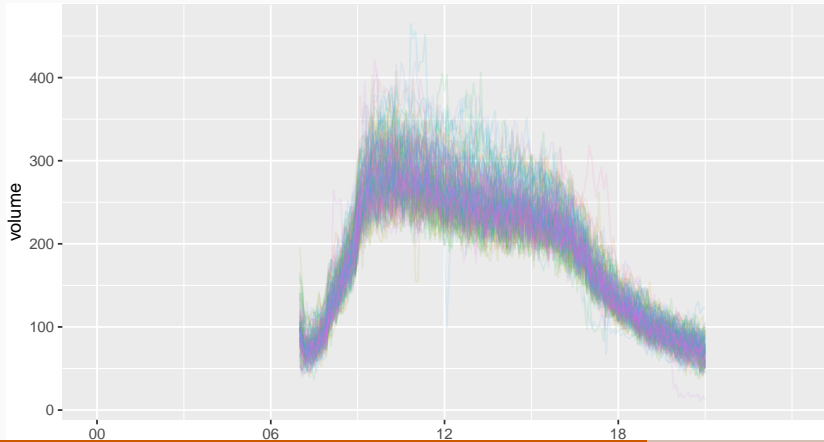
# 5-minute call centre volume

```
calls %>% fill_gaps() %>% autoplot(volume)
```



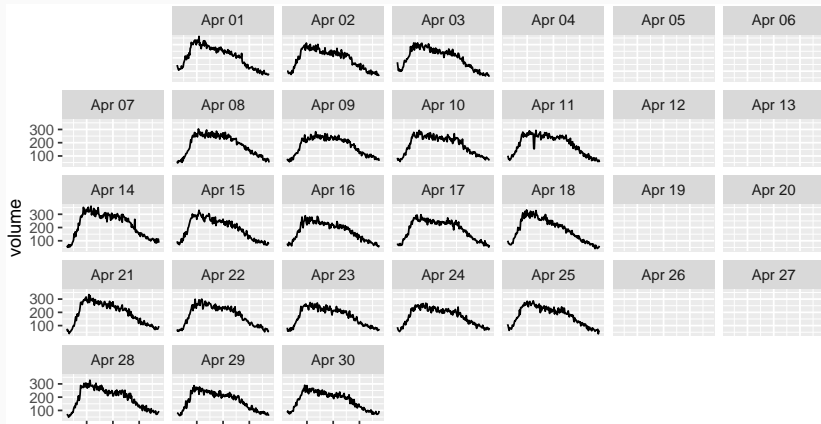
# 5-minute call centre volume

```
calls %>% fill_gaps() %>%  
  gg_season(volume, period = "day", alpha = 0.1) +  
  guides(colour = FALSE)
```



# 5-minute call centre volume

```
library(sugrrants)
calls %>% filter(month(date, label = TRUE) == "Apr") %>%
  ggplot(aes(x = time, y = volume)) +
  geom_line() + facet_calendar(date)
```





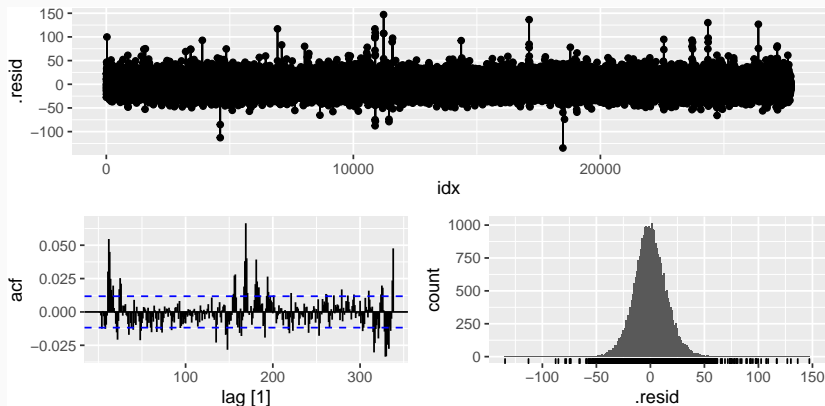
# 5-minute call centre volume

```
calls_md1 <- calls %>%  
  mutate(idx = row_number()) %>%  
  update_tsibble(index = idx)  
fit <- calls_md1 %>%  
  model(ARIMA(volume ~ fourier(169, K = 10) + pdq(d=0) + PDQ(0,0,0)))  
report(fit)
```

```
## Series: volume  
## Model: LM w/ ARIMA(1,0,3) errors  
##  
## Coefficients:  
##          ar1          ma1          ma2          ma3  fourier(169, K = 10)C1_169  
##          0.989   -0.7383   -0.0333   -0.0282                                -79.1  
## s.e.    0.001    0.0061    0.0075    0.0060                                0.7  
##          fourier(169, K = 10)S1_169  fourier(169, K = 10)C2_169  
##                                     55.298                                -32.361  
## s.e.                                0.701                                0.378  
##          fourier(169, K = 10)S2_169  fourier(169, K = 10)C3_169  
##                                     13.742                                -9.318  
## s.e.                                0.379                                0.273  
##          fourier(169, K = 10)S3_169  fourier(169, K = 10)C4_169  
##                                     -13.645                                -2.791  
## s.e.                                0.273                                0.223  
##          fourier(169, K = 10)S4_169  fourier(169, K = 10)C5_169
```

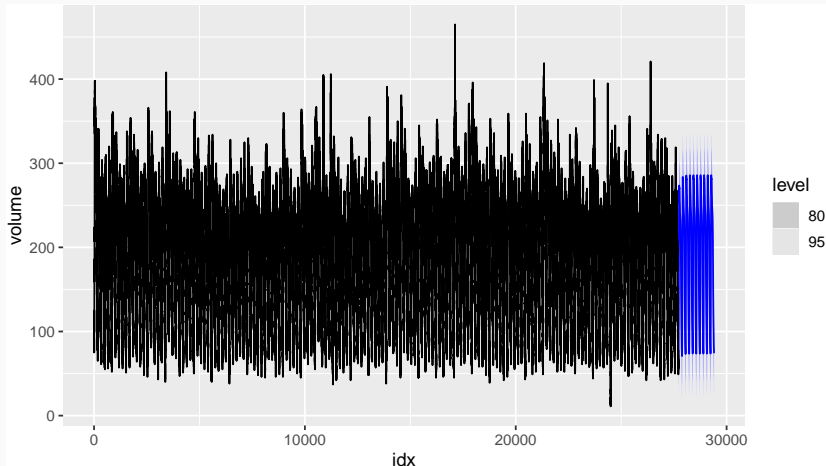
# 5-minute call centre volume

```
gg_tsresiduals(fit, lag = 338)
```



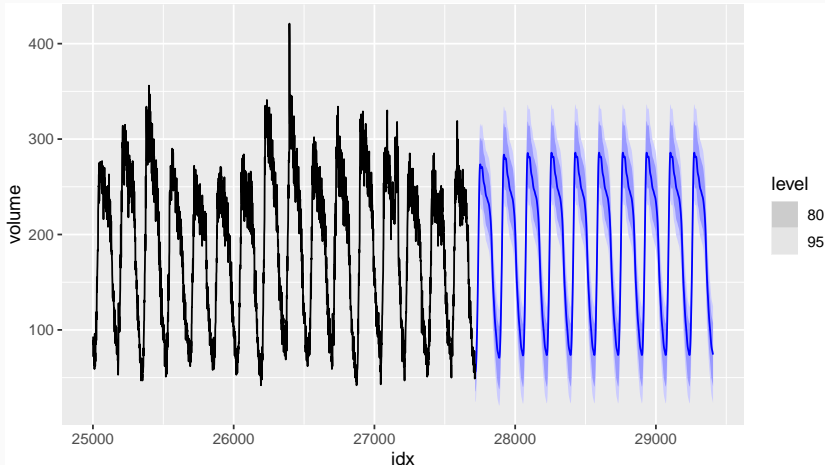
# 5-minute call centre volume

```
fit %>% forecast(h = 1690) %>%  
  autoplot(calls_mdl)
```



# 5-minute call centre volume

```
fit %>% forecast(h = 1690) %>%  
  autoplot(filter(calls_mdl, idx > 25000))
```



# Outline

- 1 Regression with ARIMA errors
- 2 Stochastic and deterministic trends
- 3 Dynamic harmonic regression
- 4 Lagged predictors

# 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:  $x_t, x_{t-1}, x_{t-2}, \dots$

$$y_t = a + \gamma_0 x_t + \gamma_1 x_{t-1} + \dots + \gamma_k x_{t-k} + \eta_t$$

where  $\eta_t$  is an ARIMA process.

# Lagged predictors

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

$$y_t = a + \gamma_0 x_t + \gamma_1 x_{t-1} + \dots + \gamma_k x_{t-k} + \eta_t$$

where  $\eta_t$  is an ARIMA process.

**Rewrite model as**

$$\begin{aligned} y_t &= a + (\gamma_0 + \gamma_1 B + \gamma_2 B^2 + \dots + \gamma_k B^k) x_t + \eta_t \\ &= a + \gamma(B) x_t + \eta_t. \end{aligned}$$

# Lagged predictors

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

$$y_t = a + \gamma_0 x_t + \gamma_1 x_{t-1} + \dots + \gamma_k x_{t-k} + \eta_t$$

where  $\eta_t$  is an ARIMA process.

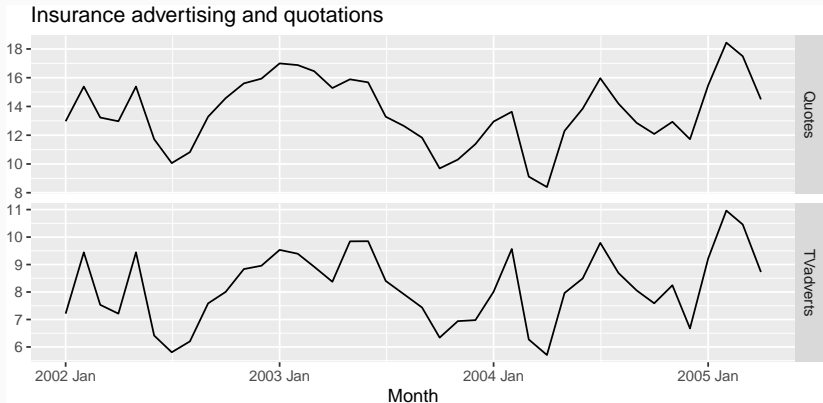
**Rewrite model as**

$$\begin{aligned} y_t &= a + (\gamma_0 + \gamma_1 B + \gamma_2 B^2 + \dots + \gamma_k B^k) x_t + \eta_t \\ &= a + \gamma(B) x_t + \eta_t. \end{aligned}$$

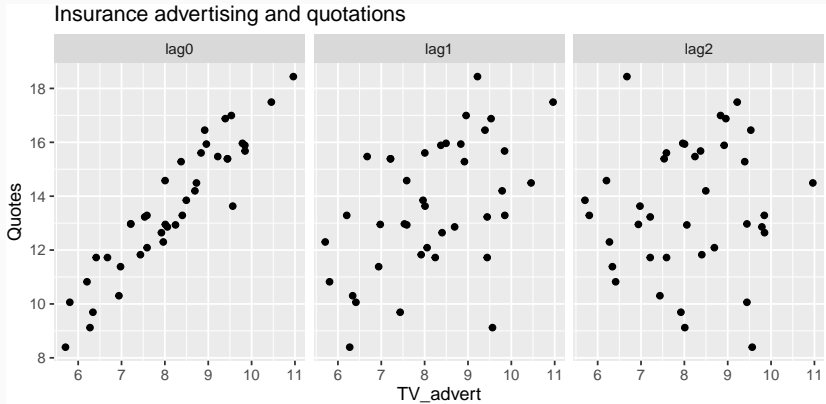
- $\gamma(B)$  is called a *transfer function* since it describes how change in  $x_t$  is transferred to  $y_t$ .
- $x$  can influence  $y$ , but  $y$  is not allowed to influence  $x$ .

# Example: Insurance quotes and TV adverts

```
insurance %>%  
  pivot_longer(Quotes:TVadverts) %>%  
  ggplot(aes(x = Month, y = value)) + geom_line() +  
  facet_grid(vars(name), scales = "free_y") +  
  labs(y = NULL, title = "Insurance advertising and quotations")
```



# Example: Insurance quotes and TV adverts



# Example: Insurance quotes and TV adverts

```
fit <- insurance %>%  
  # Restrict data so models use same fitting period  
  mutate(Quotes = c(NA,NA,NA,Quotes[4:40])) %>%  
  # Estimate models  
  model(  
    ARIMA(Quotes ~ pdq(d = 0) + TVadvert),  
    ARIMA(Quotes ~ pdq(d = 0) + TVadvert + lag(TVadverts)),  
    ARIMA(Quotes ~ pdq(d = 0) + TVadvert + lag(TVadverts) +  
      lag(TVadvert, 2)),  
    ARIMA(Quotes ~ pdq(d = 0) + TVadverts + lag(TVadverts) +  
      lag(TVadvert, 2) + lag(TVadverts, 3))  
  )
```

# Example: Insurance quotes and TV adverts

```
glance(fit)
```

Lag order	sigma2	log_lik	AIC	AICc	BIC
0	0.265	-28.3	66.6	68.3	75.0
1	0.209	-24.0	58.1	59.9	66.5
2	0.215	-24.0	60.0	62.6	70.2
3	0.206	-22.2	60.3	65.0	73.8

# Example: Insurance quotes and TV adverts

```
fit_best <- insurance %>%  
  model(ARIMA(Quotes ~ pdq(d=0) + TVadverts + lag(TVadverts)))  
report(fit_best)
```

```
## Series: Quotes  
## Model: LM w/ ARIMA(1,0,2) errors  
##  
## Coefficients:  
##          ar1      ma1      ma2 TVadverts lag(TVadverts)  
##          0.512  0.917  0.459    1.2527      0.1464  
## s.e.    0.185  0.205  0.190    0.0588      0.0531  
##          intercept  
##              2.16  
## s.e.          0.86  
##  
## sigma^2 estimated as 0.2166: log likelihood=-23.9  
## AIC=61.9   AICc=65.4   BIC=73.7
```



# Example: Insurance quotes and TV adverts

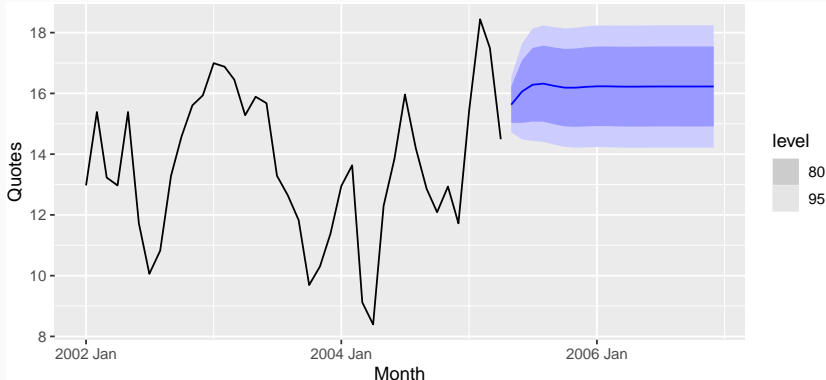
```
fit_best <- insurance %>%  
  model(ARIMA(Quotes ~ pdq(d=0) + TVadverts + lag(TVadverts)))  
report(fit_best)
```

```
## Series: Quotes  
## Model: LM w/ ARIMA(1,0,2) errors  
##  
## Coefficients:  
##          ar1      ma1      ma2 TVadverts lag(TVadverts)  
##          0.512  0.917  0.459    1.2527      0.1464  
## s.e.    0.185  0.205  0.190    0.0588      0.0531  
##          intercept  
##              2.16  
## s.e.          0.86  
##  
## sigma^2 estimated as 0.2166: log likelihood=-23.9  
## AIC=61.9   AICc=65.4   BIC=73.7
```

$$y_t = 2.155 + 1.253x_t + 0.146x_{t-1} + \eta_t,$$
$$\eta_t = 0.512\eta_{t-1} + \varepsilon_t + 0.917\varepsilon_{t-1} + 0.459\varepsilon_{t-2},$$

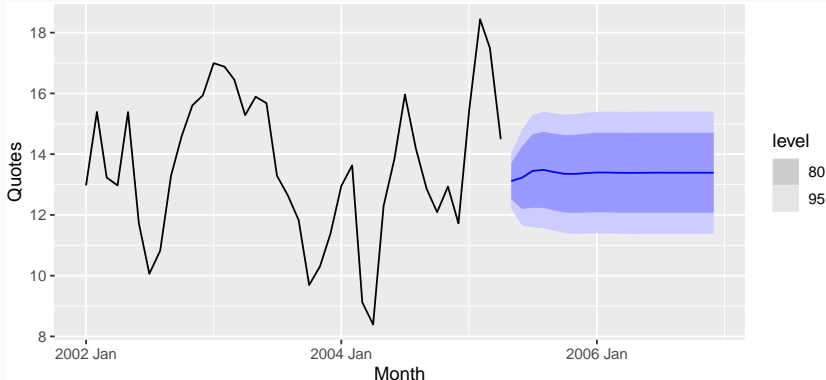
# Example: Insurance quotes and TV adverts

```
advert_a <- new_data(insurance, 20) %>%  
  mutate(TVadverts = 10)  
forecast(fit, advert_a) %>% autoplot(insurance)
```



# Example: Insurance quotes and TV adverts

```
advert_b <- new_data(insurance, 20) %>%  
  mutate(TVadverts = 8)  
forecast(fit, advert_b) %>% autoplot(insurance)
```



# Example: Insurance quotes and TV adverts

```
advert_c <- new_data(insurance, 20) %>%  
  mutate(TVadverts = 6)  
forecast(fit, advert_c) %>% autoplot(insurance)
```

