

Tidy Time Series & Forecasting in R

9. Dynamic regression

bit.ly/fable2020



Outline

- 1 Regression with ARIMA errors
- 2 Lab Session 18
- 3 Dynamic harmonic regression
- 4 Lab Session 19
- 5 Lagged predictors

Outline

1 Regression with ARIMA errors

2 Lab Session 18

3 Dynamic harmonic regression

4 Lab Session 19

5 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
- In regression, we assume that ε_t is white noise.

Regression with ARIMA errors

Regression models

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

- y_t modeled as function of k explanatory variables
- In regression, we assume that ε_t is white noise.

RegARIMA model

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

$$\eta_t \sim \text{ARIMA}$$

- Residuals are from ARIMA model.
- Estimate model in one step using MLE
- Select model with lowest AICc value.

US personal consumption and income

```
us_change
```

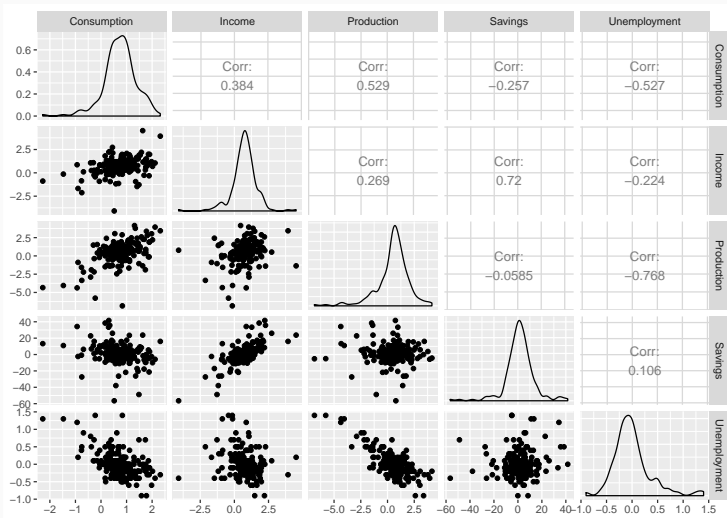
```
## # A tsibble: 198 x 6 [1Q]
##   Quarter Consumption Income Production Savings Unemployment
##   <qtr>      <dbl>  <dbl>      <dbl>    <dbl>      <dbl>
## 1 1970 Q1      0.619  1.04      -2.45     5.30       0.9
## 2 1970 Q2      0.452  1.23      -0.551    7.79       0.5
## 3 1970 Q3      0.873  1.59      -0.359    7.40       0.5
## 4 1970 Q4     -0.272 -0.240     -2.19     1.17      0.700
## 5 1971 Q1      1.90   1.98       1.91     3.54     -0.100
## 6 1971 Q2      0.915  1.45       0.902    5.87     -0.100
## 7 1971 Q3      0.794  0.521     0.308   -0.406     0.100
## 8 1971 Q4      1.65   1.16       2.29    -1.49       0
## 9 1972 Q1      1.31   0.457     4.15    -4.29     -0.2
## 10 1972 Q2     1.89   1.03       1.89    -4.69     -0.100
## # ... with 188 more rows
```

US personal consumption and income



US personal consumption and income

```
us_change %>% as_tibble() %>% select(-Quarter) %>% GGally::ggpairs()
```



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 <- us_change %>%  
  model(regarima = ARIMA(Consumption ~ Income + Production + Savings + Unemployment  
report(fit)
```

```
## Series: Consumption  
## Model: LM w/ ARIMA(0,1,2) errors  
##  
## Coefficients:  
##          ma1      ma2  Income  Production  Savings  Unemployment  
##      -1.0882  0.1118  0.7472      0.0370  -0.0531      -0.2096  
## s.e.   0.0692  0.0676  0.0403      0.0229   0.0029      0.0986  
##  
## sigma^2 estimated as 0.09588: log likelihood=-47.13  
## AIC=108.27  AICc=108.86  BIC=131.25
```

US personal consumption and income

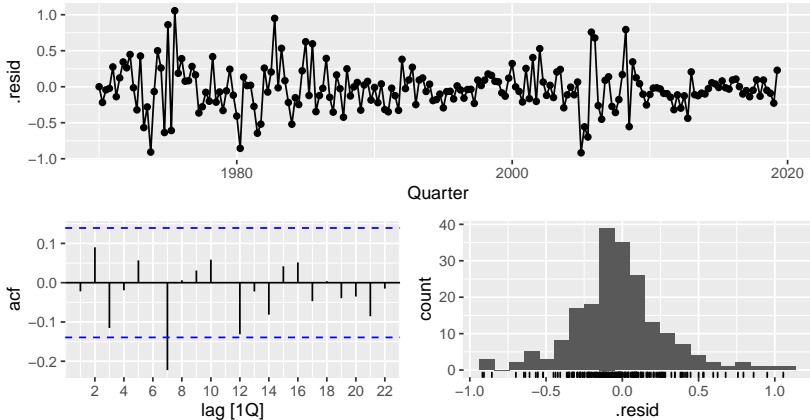
```
fit <- us_change %>%  
  model(regarima = ARIMA(Consumption ~ Income + Production + Savings + Unemployment  
report(fit)
```

```
## Series: Consumption  
## Model: LM w/ ARIMA(0,1,2) errors  
##  
## Coefficients:  
##           ma1      ma2  Income  Production  Savings  Unemployment  
##          -1.0882  0.1118  0.7472      0.0370  -0.0531      -0.2096  
## s.e.      0.0692  0.0676  0.0403      0.0229   0.0029      0.0986  
##  
## sigma^2 estimated as 0.09588: log likelihood=-47.13  
## AIC=108.27  AICc=108.86  BIC=131.25
```

Write down the equations for the fitted model.

US personal consumption and income

```
gg_tsresiduals(fit)
```



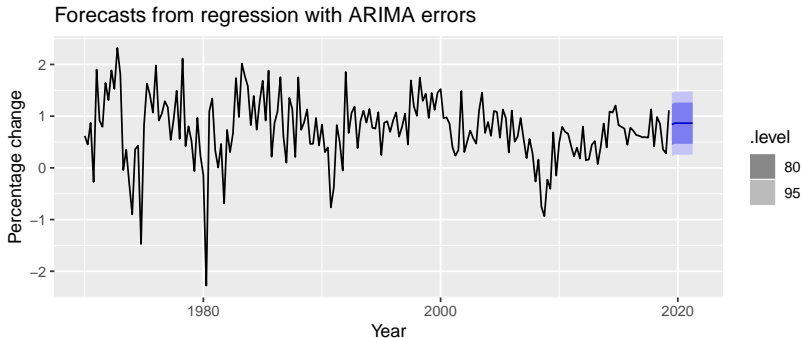
US personal consumption and income

```
augment(fit) %>%  
  features(.resid, ljung_box, dof = 6, lag = 12)
```

```
## # A tibble: 1 x 3  
##   .model    lb_stat lb_pvalue  
##   <chr>      <dbl>    <dbl>  
## 1 regarima    20.0    0.00274
```

US personal consumption and income

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



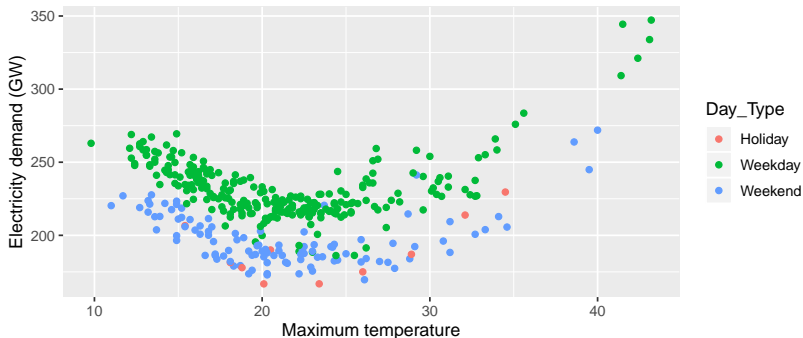
Forecasting

- 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

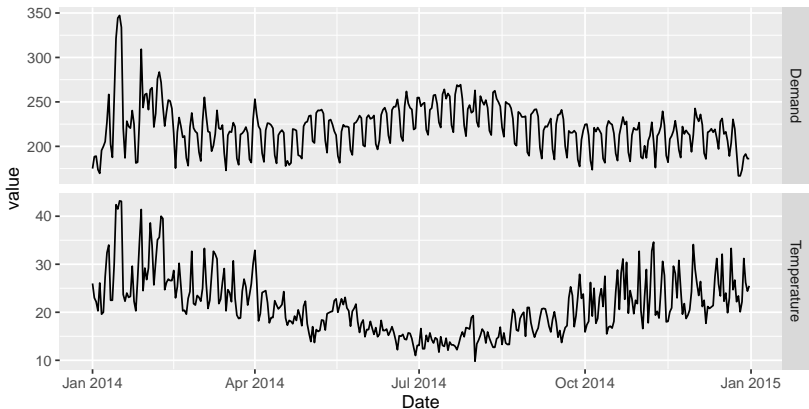
Model daily electricity demand as a function of temperature using quadratic regression with ARMA errors.

```
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(vars(name), scales = "free_y")
```



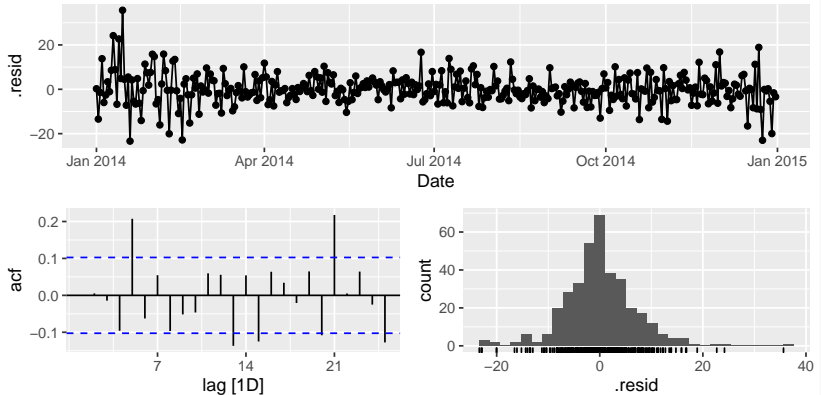
Daily electricity demand

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

```
## Series: Demand  
## Model: LM w/ ARIMA(2,1,2)(0,0,2)[7] errors  
##  
## Coefficients:  
##          ar1      ar2      ma1      ma2      sma1      sma2  Temperature  
##          1.1521 -0.2750 -1.3851  0.4071  0.1589  0.3103      -7.9467  
## s.e.      0.6265  0.4812  0.6082  0.5805  0.0591  0.0538      0.4920  
##          I(Temperature^2)  Day_Type == "Weekday"TRUE  
##                          0.1865                          31.8245  
## s.e.                      0.0097                          1.0189  
##  
## sigma^2 estimated as 48.82: log likelihood=-1220.48  
## AIC=2460.96   AICc=2461.58   BIC=2499.93
```

Daily electricity demand

```
augment(fit) %>%  
  gg_tsdisplay(.resid, plot_type = "histogram")
```



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 fit      38.1 0.000000354
```

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]
```

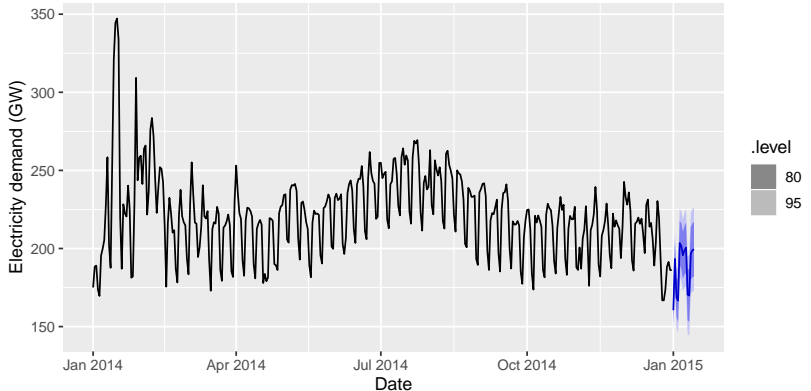
```
##   .model Date      Demand .distribution Temperature Day_Type  
##   <chr>  <date>      <dbl> <dist>          <dbl> <chr>  
## 1 fit    2015-01-01    161. N(161, 49)          26 Holiday
```

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, vic_elec_future) %>%  
  autoplot(vic_elec_daily) + ylab("Electricity demand (GW)")
```



Outline

- 1 Regression with ARIMA errors
- 2 Lab Session 18
- 3 Dynamic harmonic regression
- 4 Lab Session 19
- 5 Lagged predictors

Lab Session 18

Repeat the daily electricity example, but instead of using a quadratic function of temperature, use a piecewise linear function with the “knot” around 20 degrees Celsius (use predictors Temperature & Temp2). How can you optimize the choice of knot?

The data can be created as follows.

```
vic_elec_daily <- vic_elec %>%  
  filter(year(Time) == 2014) %>%  
  index_by(Date = date(Time)) %>%  
  summarise(  
    Demand = sum(Demand) / 1e3,  
    Temperature = max(Temperature),  
    Holiday = any(Holiday)  
  ) %>%  
  mutate(  
    Temp2 = I(pmax(Temperature - 20, 0)),  
    Day_Type = case_when(  
      Holiday ~ "Holiday",  
      wday(Date) %in% 2:6 ~ "Weekday",  
      TRUE ~ "Weekend"))
```

Outline

- 1 Regression with ARIMA errors
- 2 Lab Session 18
- 3 Dynamic harmonic regression
- 4 Lab Session 19
- 5 Lagged predictors

Dynamic harmonic regression

Combine Fourier terms with ARIMA errors

Advantages

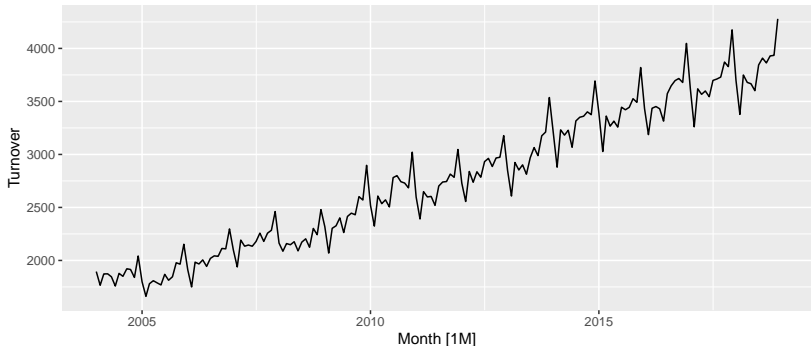
- it allows any length seasonality;
- for data with more than one seasonal period, you can include Fourier terms of different frequencies;
- the seasonal pattern is smooth for small values of K (but more wiggly seasonality can be handled by increasing K);
- the short-term dynamics are easily handled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

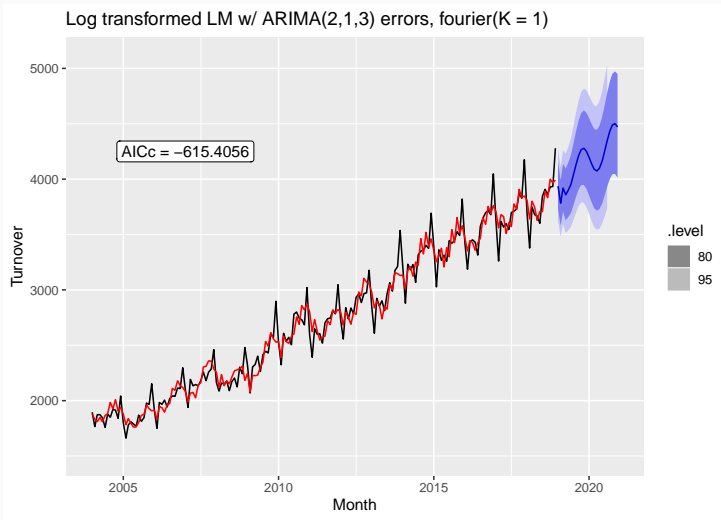


Eating-out expenditure

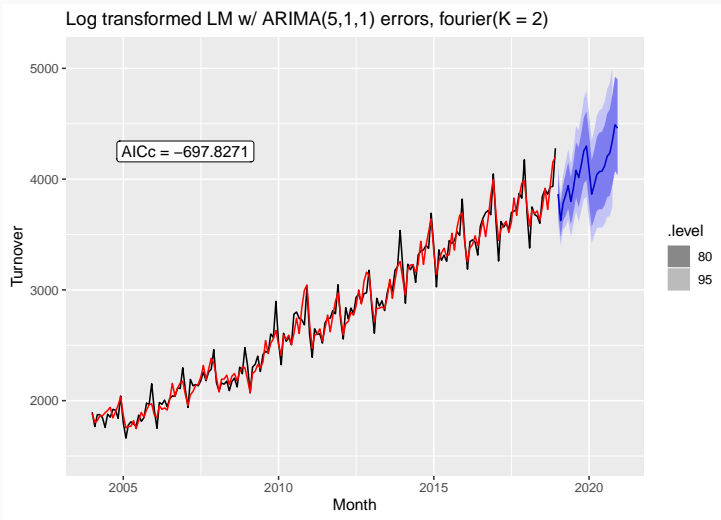
```
fit <- aus_cafe %>% model(  
  K = 1 = ARIMA(log(Turnover) ~ fourier(K = 1) + PDQ(0, 0, 0)),  
  K = 2 = ARIMA(log(Turnover) ~ fourier(K = 2) + PDQ(0, 0, 0)),  
  K = 3 = ARIMA(log(Turnover) ~ fourier(K = 3) + PDQ(0, 0, 0)),  
  K = 4 = ARIMA(log(Turnover) ~ fourier(K = 4) + PDQ(0, 0, 0)),  
  K = 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	0.0017471	317.2353	-616.4707	-615.4056	-587.7842
K = 2	0.0010732	361.8533	-699.7066	-697.8271	-661.4579
K = 3	0.0007609	393.6062	-763.2125	-761.3329	-724.9638
K = 4	0.0005386	426.7839	-821.5678	-818.2098	-770.5697
K = 5	0.0003173	473.7344	-919.4688	-916.9078	-874.8454
K = 6	0.0003163	474.0307	-920.0614	-917.5004	-875.4380

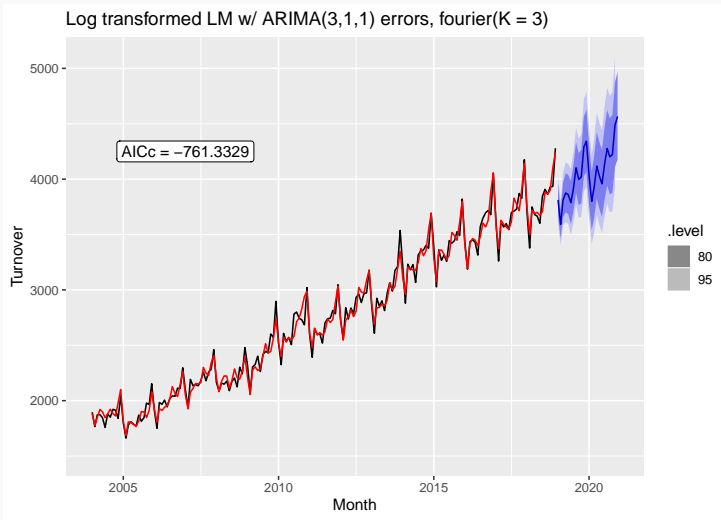
Eating-out expenditure



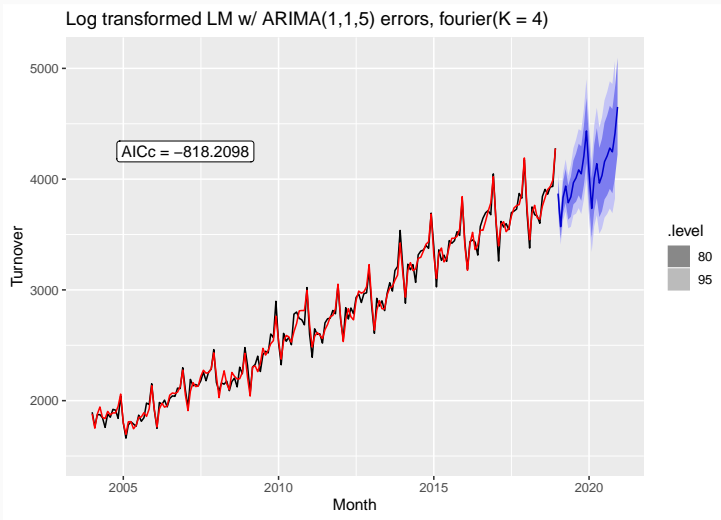
Eating-out expenditure



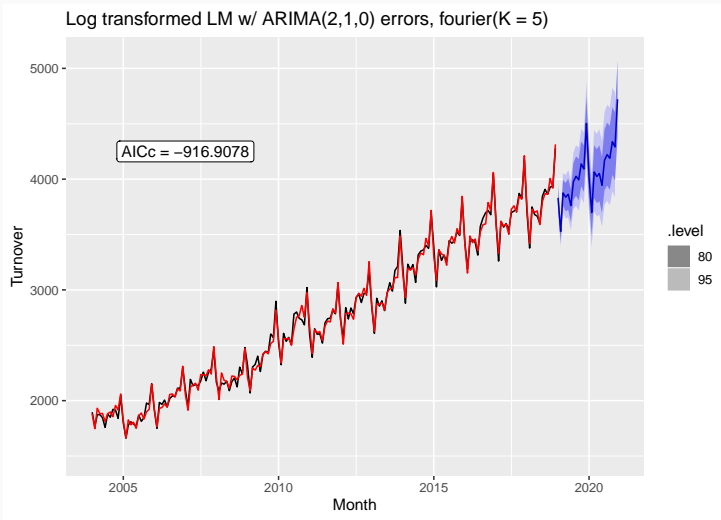
Eating-out expenditure



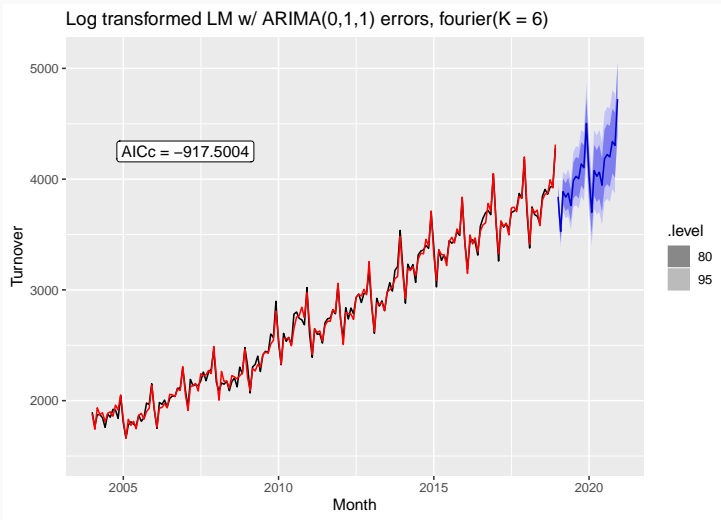
Eating-out expenditure



Eating-out expenditure



Eating-out expenditure



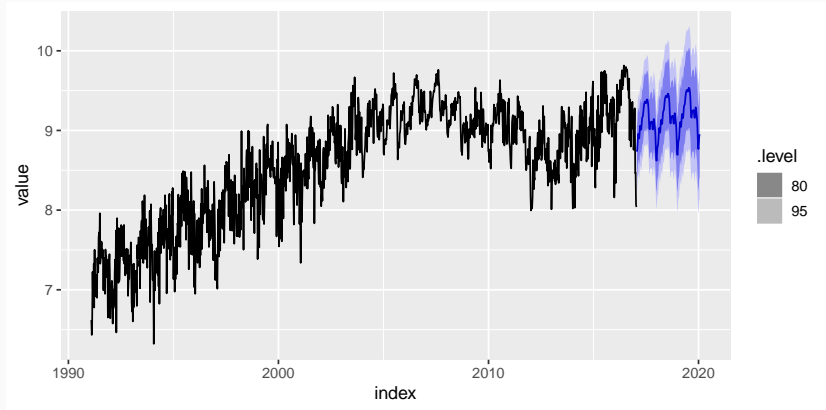
Example: weekly gasoline products

```
gasoline <- as_tsibble(fpp2::gasoline)
fit <- gasoline %>% model(ARIMA(value ~ fourier(K = 13) + PDQ(0, 0, 0)))
report(fit)
```

```
## Series: value
## 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(gasoline)
```



Outline

- 1 Regression with ARIMA errors
- 2 Lab Session 18
- 3 Dynamic harmonic regression
- 4 Lab Session 19
- 5 Lagged predictors

Lab Session 19

Repeat Lab Session 18 but using all available data, and handling the annual seasonality using Fourier terms.

Outline

- 1 Regression with ARIMA errors
- 2 Lab Session 18
- 3 Dynamic harmonic regression
- 4 Lab Session 19
- 5 Lagged predictors

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

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

where η_t is an ARIMA process.

Lagged predictors

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

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

where η_t is an ARIMA process.

- x can influence y , but y is not allowed to influence x .

Example: Insurance quotes and TV adverts

```
## # A tibble: 40 x 3 [1M]
```

```
##           Month Quotes TV.advert
```

```
##           <mth>   <dbl>       <dbl>
```

```
## 1 2002 Jan      13.0         7.21
```

```
## 2 2002 Feb      15.4         9.44
```

```
## 3 2002 Mar      13.2         7.53
```

```
## 4 2002 Apr      13.0         7.21
```

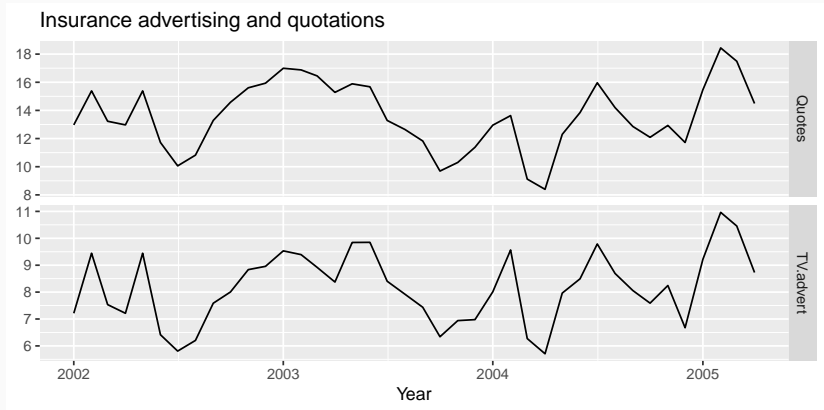
```
## 5 2002 May      15.4         9.44
```

```
## 6 2002 Jun      11.7         6.42
```

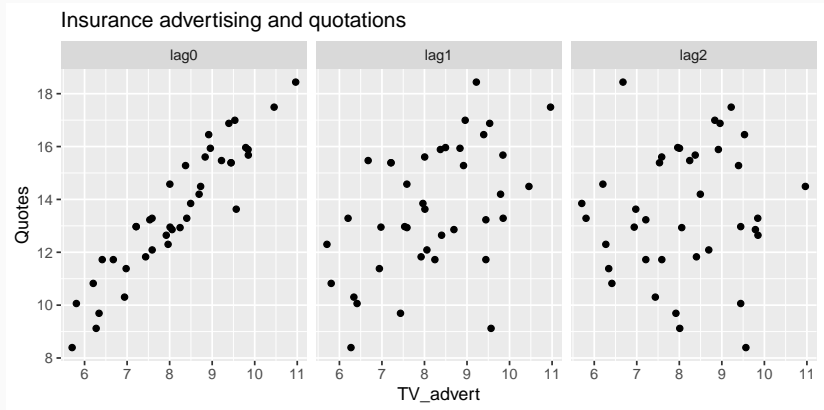
```
## 7 2002 Jul      10.1         5.81
```

```
## 8 2002 Aug      10.8         6.20
```

Example: Insurance quotes and TV adverts



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])) %>%  
  model(  
    ARIMA(Quotes ~ pdq(d = 0) + TV.advert),  
    ARIMA(Quotes ~ pdq(d = 0) + TV.advert +  
          lag(TV.advert)),  
    ARIMA(Quotes ~ pdq(d = 0) + TV.advert +  
          lag(TV.advert) +  
          lag(TV.advert, 2)),  
    ARIMA(Quotes ~ pdq(d = 0) + TV.advert +  
          lag(TV.advert) +  
          lag(TV.advert, 2) +  
          lag(TV.advert, 3))  
  )
```


Example: Insurance quotes and TV adverts

```
glance(fit)
```

Lag order	sigma2	log_lik	AIC	AICc	BIC
0	0.2649757	-28.28210	66.56420	68.32890	75.00859
1	0.2094368	-24.04404	58.08809	59.85279	66.53249
2	0.2150429	-24.01627	60.03254	62.57799	70.16581
3	0.2056454	-22.15731	60.31461	64.95977	73.82565

Example: Insurance quotes and TV adverts

```
# Re-fit to all data
fit <- insurance %>%
  model(ARIMA(Quotes ~ TV.advert + lag(TV.advert) + pdq(d = 0)))
report(fit)

## Series: Quotes
## Model: LM w/ ARIMA(1,0,2) errors
##
## Coefficients:
##          ar1          ma1          ma2  TV.advert  lag(TV.advert)  intercept
##          0.5123  0.9169  0.4591      1.2527           0.1464      2.1554
## s.e.  0.1849  0.2051  0.1895      0.0588           0.0531      0.8595
##
## sigma^2 estimated as 0.2166: log likelihood=-23.94
## AIC=61.88   AICc=65.38   BIC=73.7
```

Example: Insurance quotes and TV adverts

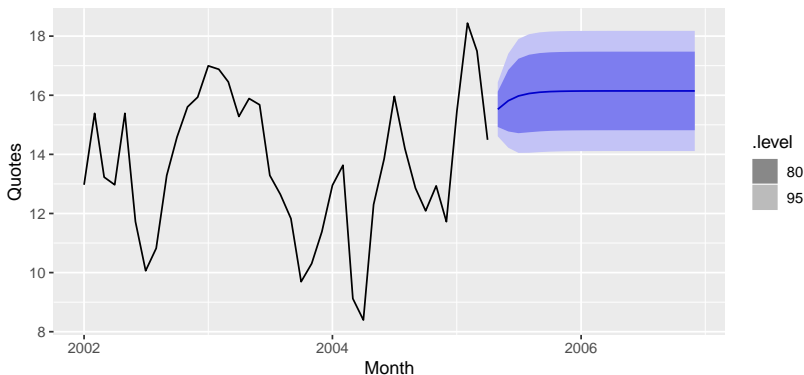
```
# Re-fit to all data
fit <- insurance %>%
  model(ARIMA(Quotes ~ TV.advert + lag(TV.advert) + pdq(d = 0)))
report(fit)
```

```
## Series: Quotes
## Model: LM w/ ARIMA(1,0,2) errors
##
## Coefficients:
##          ar1          ma1          ma2  TV.advert  lag(TV.advert)  intercept
##          0.5123  0.9169  0.4591      1.2527           0.1464      2.1554
## s.e.  0.1849  0.2051  0.1895      0.0588           0.0531      0.8595
##
## sigma^2 estimated as 0.2166:  log likelihood=-23.94
## AIC=61.88   AICc=65.38   BIC=73.7
```

$$y_t = 2.16 + 1.25x_t + 0.15x_{t-1} + \eta_t,$$
$$\eta_t = 0.512\eta_{t-1} + \varepsilon_t + 0.92\varepsilon_{t-1} + 0.46\varepsilon_{t-2}.$$

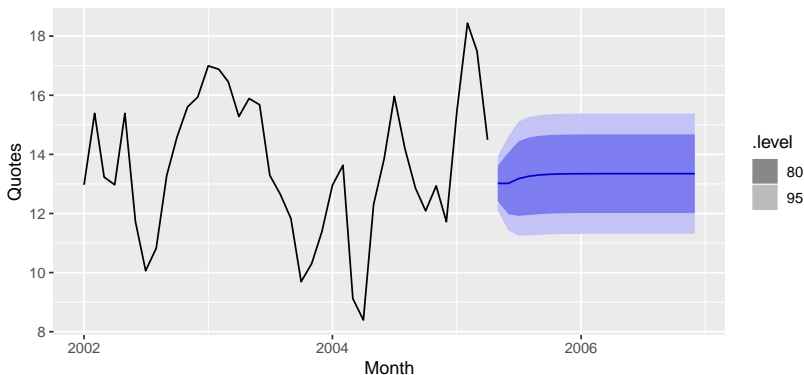
Example: Insurance quotes and TV adverts

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



Example: Insurance quotes and TV adverts

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



Example: Insurance quotes and TV adverts

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

