bit.ly/fable2022

Time Series Analysis & Forecasting Using R

9. Dynamic regression

tsibble feasts fable

Outline

Outline

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.

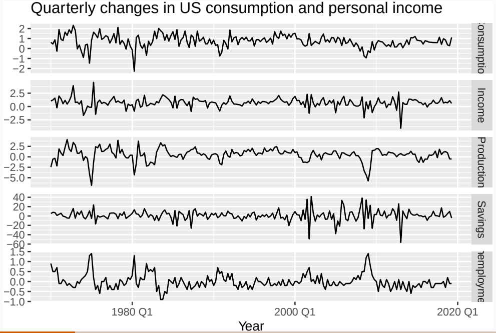
US personal consumption and income

```
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        -0.200
## 10 1972 Q2       1.89   1.03       1.89   -4.69        -0.100
## # ... with 188 more rows
```

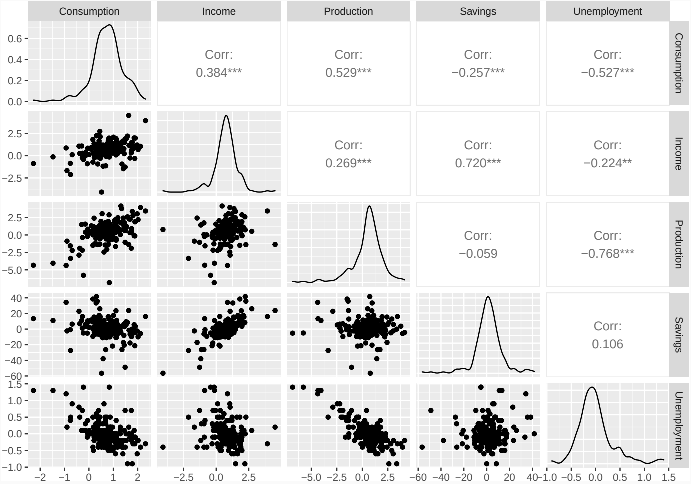
US personal consumption and income

```
us_change ▷
  pivot_longer(-Quarter, names_to = "variable", values_to = "value") ▷
  ggplot(aes(y = value, x = Quarter, group = variable)) +
  geom_line() + facet_grid(variable ~ ., scales = "free_y") +
  labs(x = "Year", y = "",
       title = "Quarterly changes in US consumption and personal 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

```r
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=-47.1
## AIC=108   AICc=109   BIC=131
```

US personal consumption and income

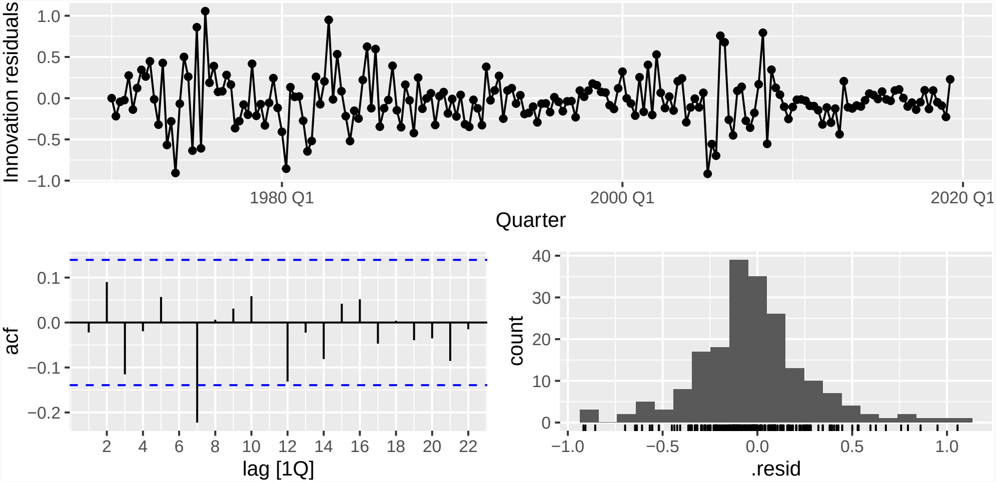
```r
fit <- us_change |>
  model(regarima = ARIMA(Consumption ~ Income + Production + Savings + Unemployment
report(fit)
```

```
## 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=-47.1
## AIC=108   AICc=109   BIC=131
```

Write down the equations for the fitted model.

US personal consumption and income

```
gg_tsresiduals(fit)
```



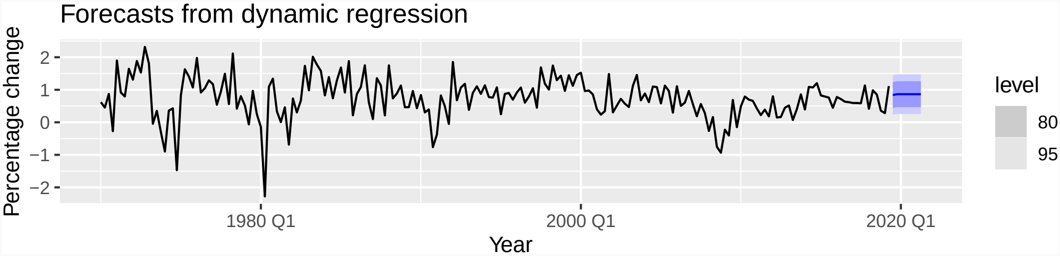
US personal consumption and income

```
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 dynamic regress
```



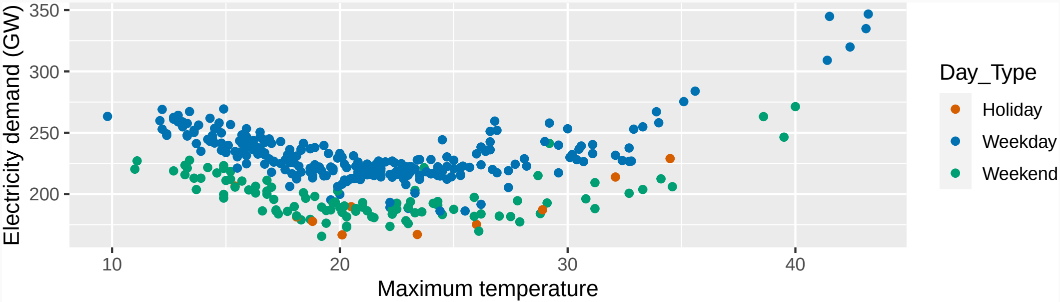
Forecasting

- 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

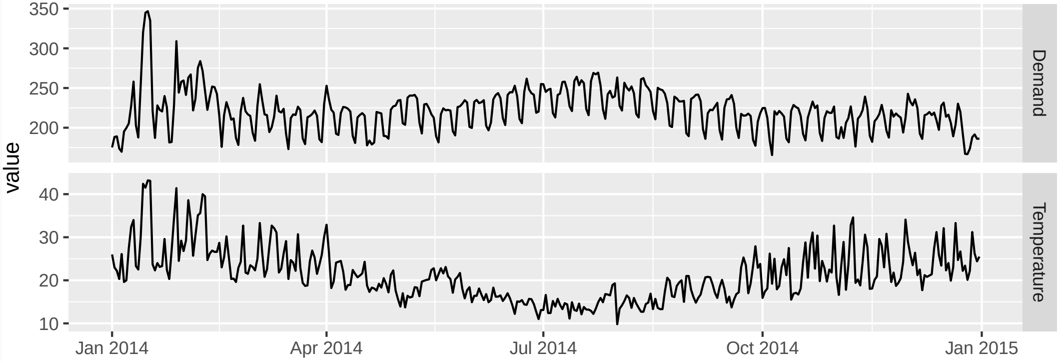
Model daily electricity demand as a function of temperature using quadratic regression with ARMA errors.

```
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")
```



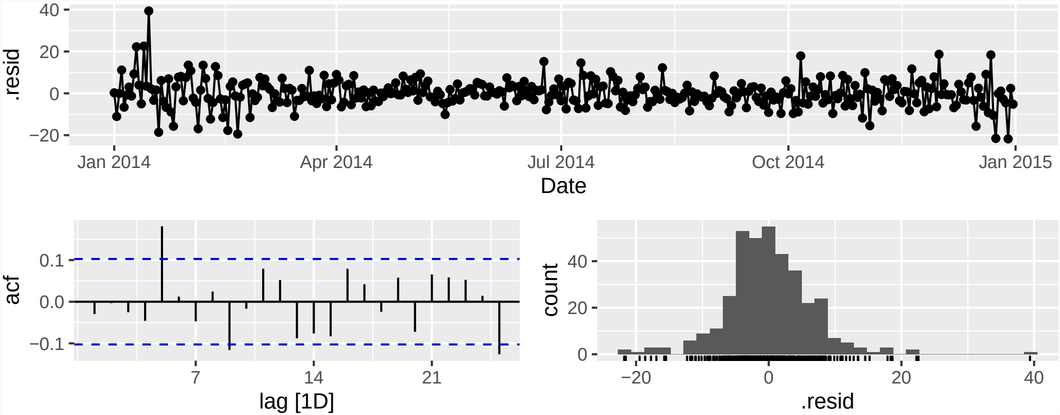
Daily electricity demand

```r
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)(2,0,0)[7] errors
##
## Coefficients:
##           ar1     ar2      ma1      ma2    sar1   sar2  Temperature
##       -0.1093  0.7226  -0.0182  -0.9381  0.1958  0.417       -7.614
## s.e.   0.0779  0.0739   0.0494   0.0493  0.0525  0.057        0.448
##       I(Temperature^2)  Day_Type == "Weekday"TRUE
##                 0.1810                      30.40
## s.e.            0.0085                       1.33
##
## sigma^2 estimated as 44.91:  log likelihood=-1206
## AIC=2432   AICc=2433   BIC=2471
```

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       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)
```

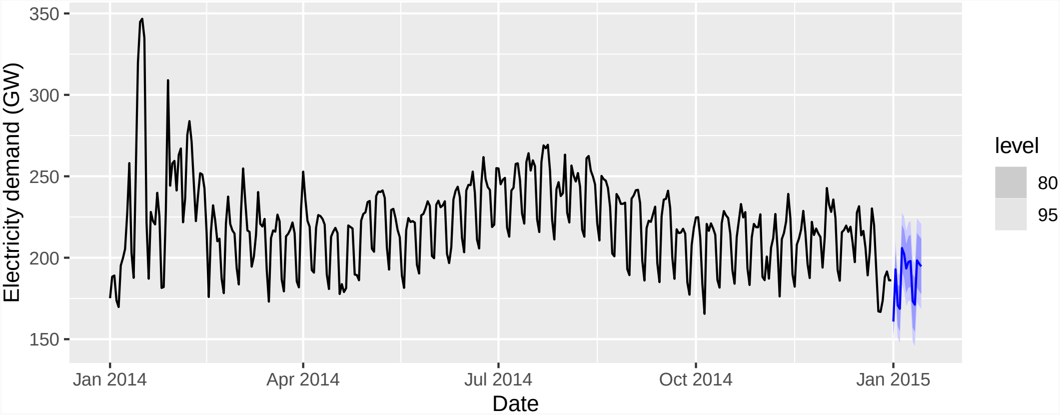
```
## # A fable: 1 x 6 [1D]
## # Key:     .model [1]
##   .model Date           Demand .mean Temperature Day_Type
##   <chr>  <date>         <dist> <dbl>       <dbl> <chr>
## 1 fit    2015-01-01 N(161, 45)  161.          26 Holiday
```

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, vic_elec_future) |>
  autoplot(vic_elec_daily) + labs(y = "Electricity demand (GW)")
```



Outline

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?

```
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

Dynamic harmonic regression

Combine Fourier terms with ARIMA errors

Advantages

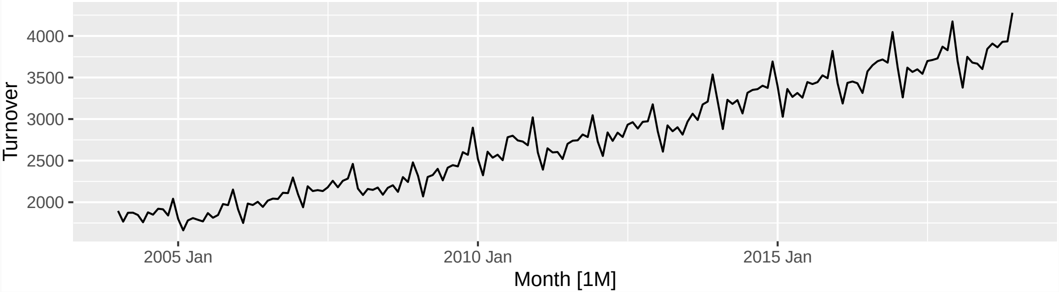
- it allows any length seasonality;
- for data with more than one seasonal period, you can include Fourier terms of different frequencies;
- the seasonal pattern is smooth for small values of K (but more wiggly seasonality can be handled by increasing K);
- the short-term dynamics are easily handled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

Disadvantages

- seasonality is assumed to be fixed

Eating-out expenditure

```r
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)
```

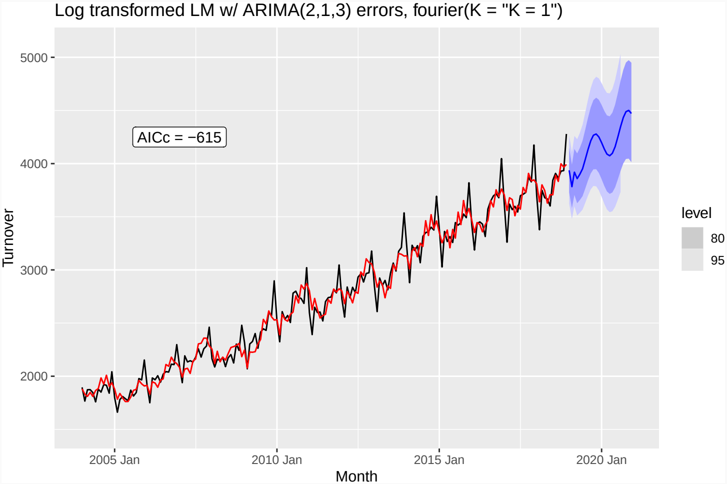


Eating-out expenditure

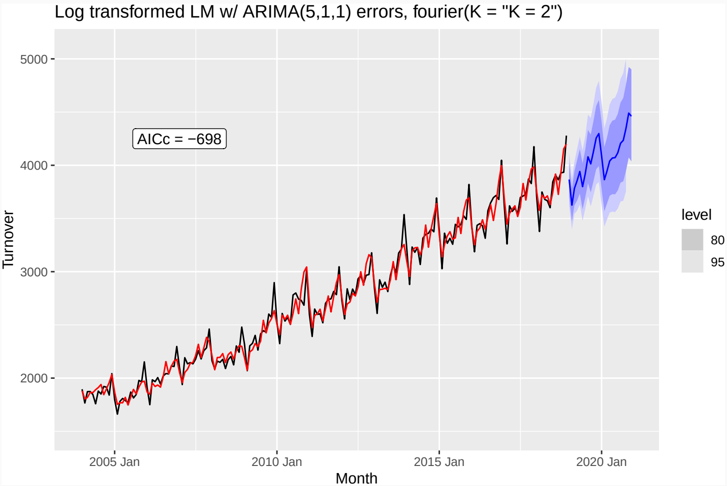
```r
fit <- aus_cafe |> model(
  `K = 1` = ARIMA(log(Turnover) ~ fourier(K = 1) + PDQ(0, 0, 0)),
  `K = 2` = ARIMA(log(Turnover) ~ fourier(K = 2) + PDQ(0, 0, 0)),
  `K = 3` = ARIMA(log(Turnover) ~ fourier(K = 3) + PDQ(0, 0, 0)),
  `K = 4` = ARIMA(log(Turnover) ~ fourier(K = 4) + PDQ(0, 0, 0)),
  `K =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

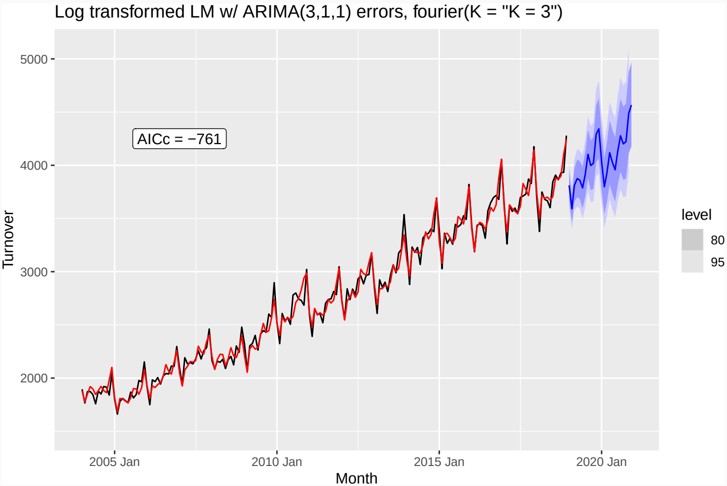
Eating-out expenditure



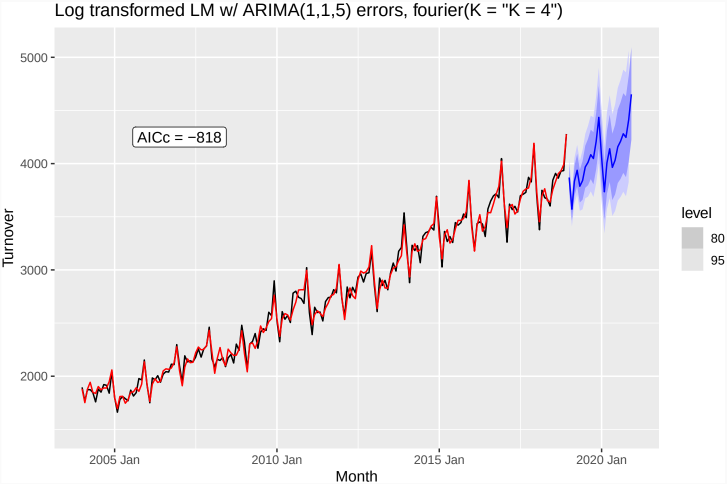
Eating-out expenditure



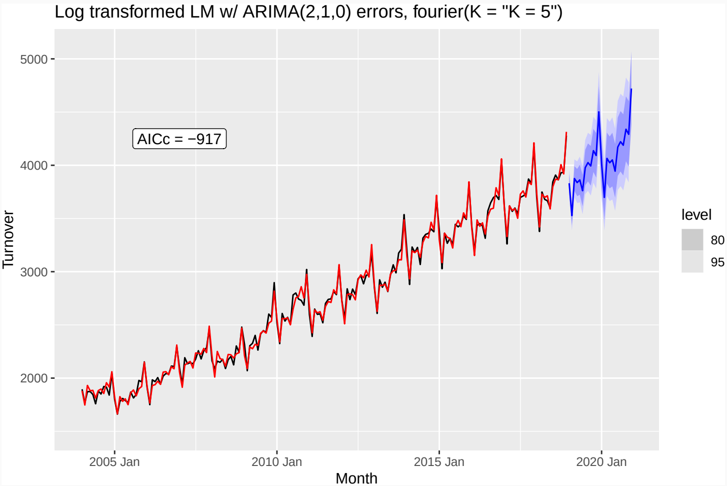
Eating-out expenditure



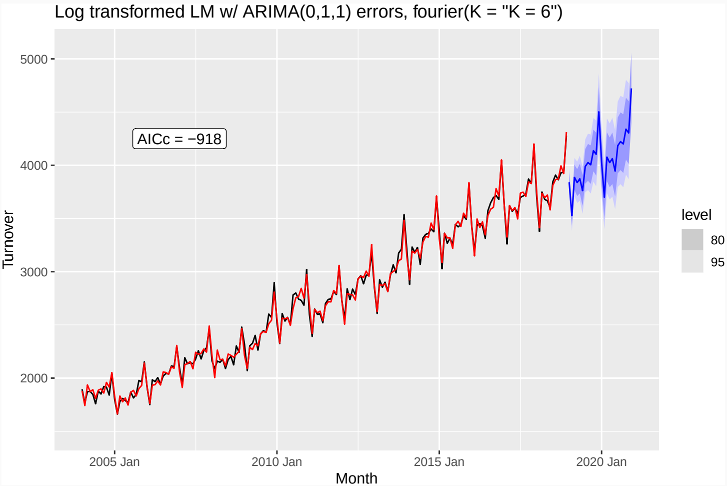
Eating-out expenditure



Eating-out expenditure



Eating-out expenditure



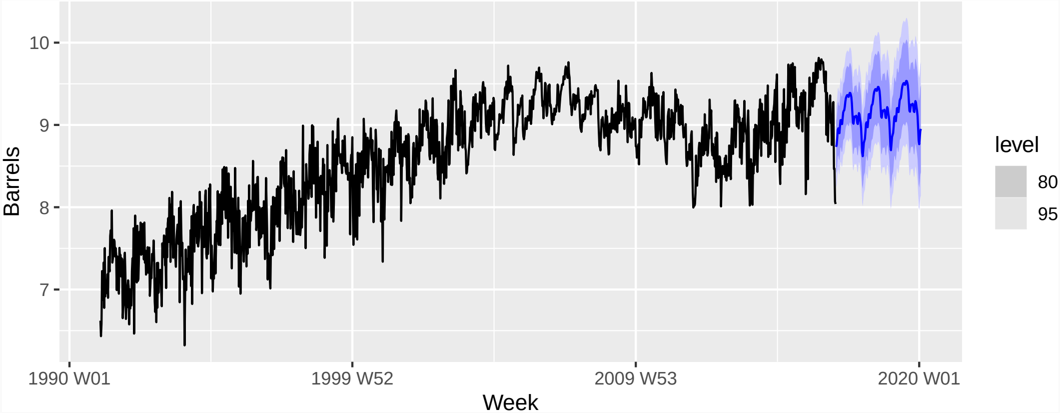
Example: weekly gasoline products

```
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  fourier(K = 13)C3_52
##                     0.0420                0.0317                0.0832
## s.e.                0.0099                0.0099                0.0094
##       fourier(K = 13)S3_52  fourier(K = 13)C4_52  fourier(K = 13)S4_52
##                     0.0346                0.0185                0.0398
## s.e.                0.0094                0.0092                0.0092
##       fourier(K = 13)C5_52  fourier(K = 13)S5_52  fourier(K = 13)C6_52
##                    -0.0315                0.0009               -0.0522
## s.e.                0.0091                0.0091                0.0090
##       fourier(K = 13)S6_52  fourier(K = 13)C7_52  fourier(K = 13)S7_52
##                      0.000               -0.0173                0.0053
```

Example: weekly gasoline products

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



Outline

Lab Session 19

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

Outline

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

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

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.

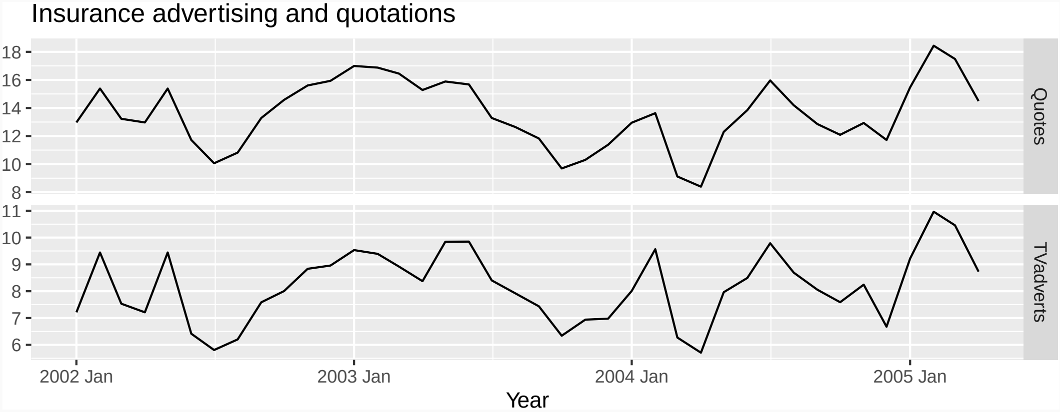
- x can influence y, but y is not allowed to influence x.

Example: Insurance quotes and TV adverts

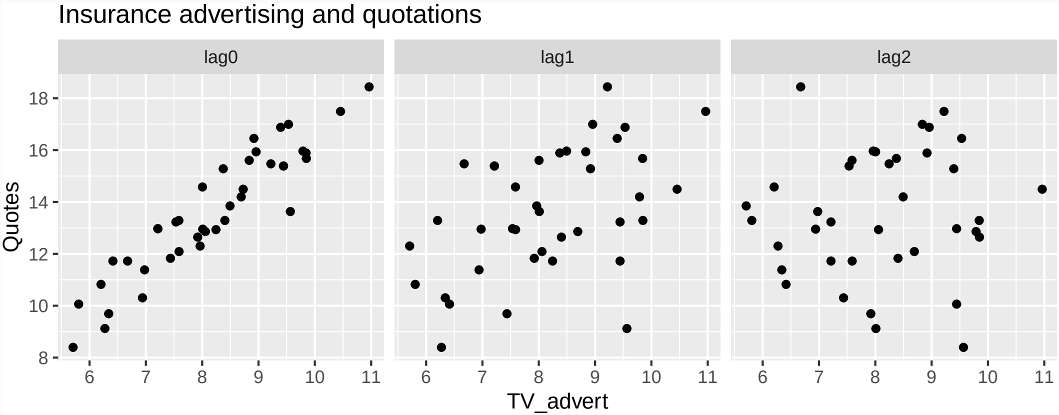
```
insurance
```

```
## # A tsibble: 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
##  9 2002 Sep   13.3      7.59
## 10 2002 Oct   14.6      8.00
```

Example: Insurance quotes and TV adverts



Example: Insurance quotes and TV adverts



Example: Insurance quotes and TV adverts

```r
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) + TVadverts),
    ARIMA(Quotes ~ pdq(d = 0) + TVadverts +
      lag(TVadverts)),
    ARIMA(Quotes ~ pdq(d = 0) + TVadverts +
      lag(TVadverts) +
      lag(TVadverts, 2)),
    ARIMA(Quotes ~ pdq(d = 0) + TVadverts +
      lag(TVadverts) +
      lag(TVadverts, 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

```r
# Re-fit to all data
fit <- insurance |>
  model(ARIMA(Quotes ~ TVadverts + lag(TVadverts) + pdq(d = 0)))
report(fit)
```

```
## Series: Quotes
## Model: LM w/ ARIMA(1,0,2) errors
##
## Coefficients:
##         ar1    ma1    ma2  TVadverts  lag(TVadverts)  intercept
##       0.512  0.917  0.459     1.2527          0.1464       2.16
## s.e.  0.185  0.205  0.190     0.0588          0.0531       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

```r
# Re-fit to all data
fit <- insurance |>
  model(ARIMA(Quotes ~ TVadverts + lag(TVadverts) + pdq(d = 0)))
report(fit)
```

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

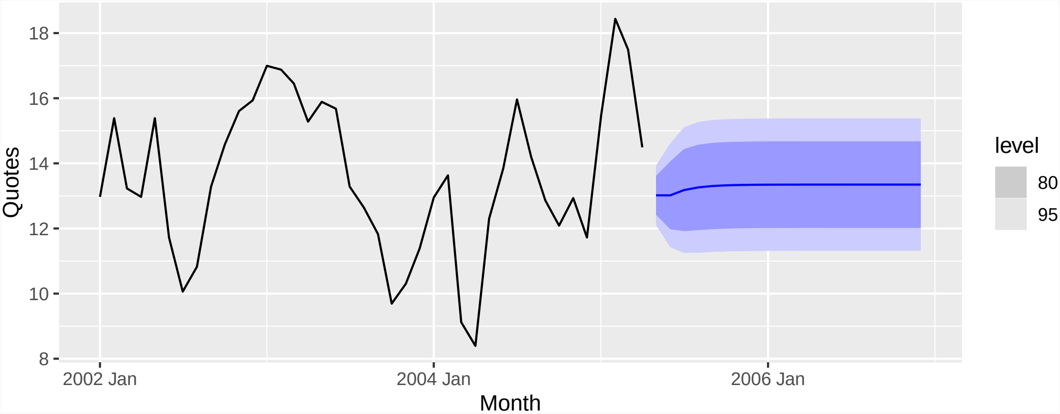
$$y_t = 2.16 + 1.25x_t + 0.15x_{t-1} + \eta_t,$$
$$\eta_t = 0.512\eta_{t-1} + \varepsilon_t + 0.92\varepsilon_{t-1} + 0.46\varepsilon_{t-2}.$$

Example: Insurance quotes and TV adverts

```
advert_a <- new_data(insurance, 20) ▷
  mutate(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)
```