MONASH University

MONASH BUSINESS SCHOOL

ETC3550
Applied forecasting for business and economics

Ch10. Dynamic regression models

OTexts.org/fpp3/

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 $x_{1,t}, \ldots, x_{k,t}$.
- In regression, we assume that ε_t was WN.
- Now we want to allow ε_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 $x_{1,t}, \ldots, x_{k,t}$.
- In regression, we assume that ε_t was WN.
- Now we want to allow ε_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 ε_t is white noise.

Residuals and errors

Example: η_t = 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: η_t = 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 η_t from ε_t.
- Only the errors η_t are assumed to be white noise.
- In ordinary regression, η_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 is similar to minimizing $\sum \varepsilon_t^2$

Stationarity

Regression with ARMA errors

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

where η_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

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

Stationarity

Model with ARIMA(1,1,1) errors

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

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

$$y'_t = \beta_1 x'_{1,t} + \cdots + \beta_k x'_{k,t} + \eta'_t,$$
$$(1 - \phi_1 B)\eta'_t = (1 + \theta_1 B)\varepsilon_t,$$

where $y'_t = y_t - y_{t-1}$, $x'_{t,i} = x_{t,i} - x_{t-1,i}$ and $\eta'_t = \eta_t - \eta_{t-1}$.

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

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

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

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

$$y_t = \beta_0 + \beta_1 x_{1,t} + \dots + \beta_k x_{k,t} + \eta_t$$
$$\text{where} \quad \phi(B)(1-B)^d \eta_t = \theta(B)\varepsilon_t$$

After differencing all variables

$$y'_t = \beta_1 x'_{1,t} + \dots + \beta_k x'_{k,t} + \eta'_t.$$
$$\text{where} \quad \phi(B)\eta_t = \theta(B)\varepsilon_t$$
$$\text{and} \quad y'_t = (1-B)^d y_t$$

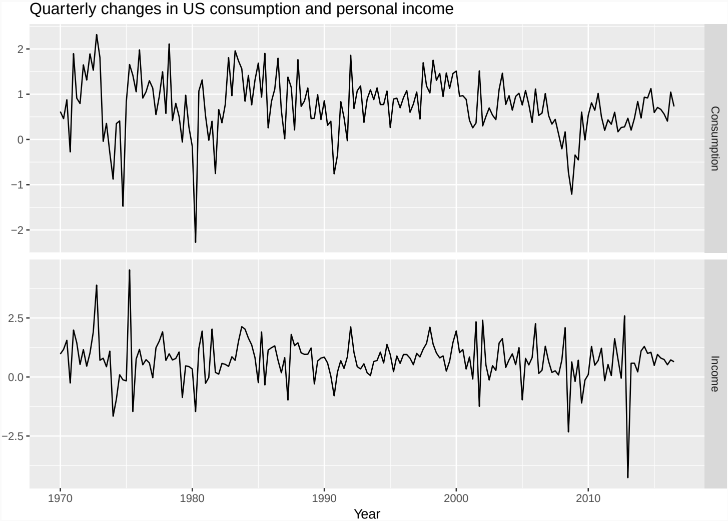
Model selection

- 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 ε_t series looks like white noise.

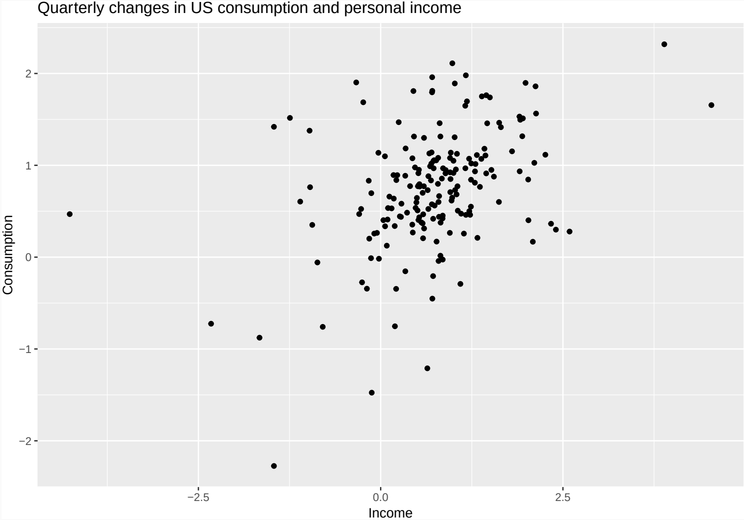
Selecting predictors

- 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

```r
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
##       0.6922  -0.5758  0.1984  0.2028     0.5990
## s.e.  0.1159   0.1301  0.0756  0.0461     0.0884
##
## sigma^2 estimated as 0.3219:  log likelihood=-156.9
## AIC=325.9   AICc=326.4   BIC=345.3
```

US personal consumption and income

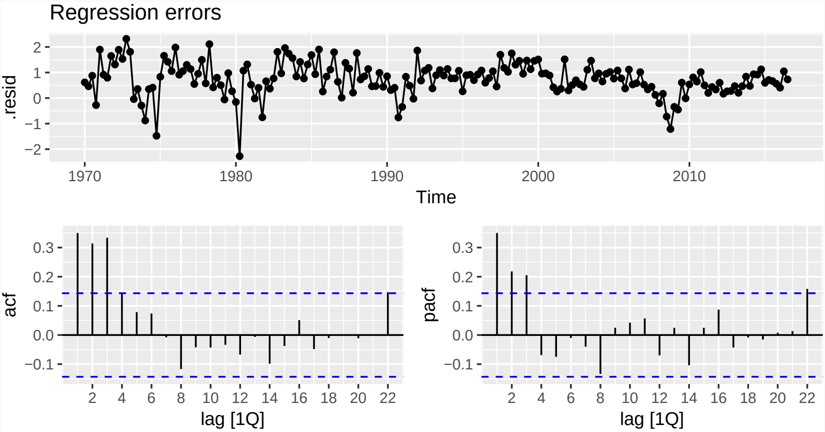
```
fit <- us_change %>% model(ARIMA(Consumption ~ Income))
report(fit)
```

```
## Series: Consumption
## Model: LM w/ ARIMA(1,0,2) errors
##
## Coefficients:
##          ar1      ma1     ma2  Income  intercept
##       0.6922  -0.5758  0.1984  0.2028     0.5990
## s.e.  0.1159   0.1301  0.0756  0.0461     0.0884
##
## sigma^2 estimated as 0.3219:  log likelihood=-156.9
## AIC=325.9   AICc=326.4   BIC=345.3
```

Write down the equations for the fitted model.

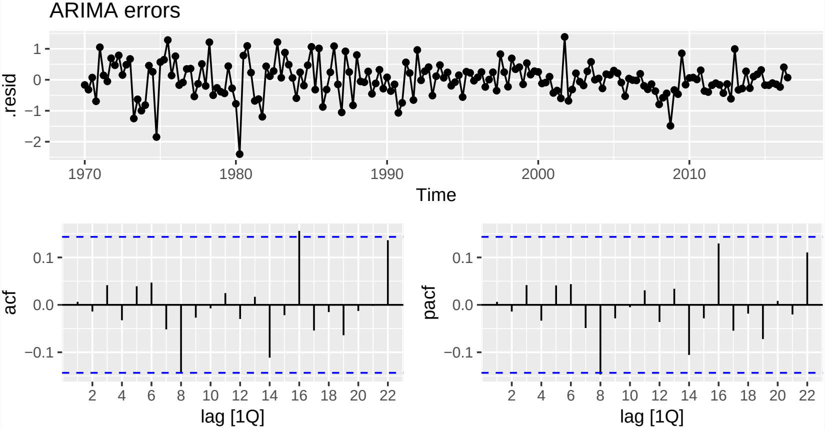
US personal consumption and income

```r
residuals(fit, type='regression') %>%
  gg_tsdisplay(.resid) + ggtitle("Regression errors")
```



US personal consumption and income

```r
residuals(fit, type='response') %>%
  gg_tsdisplay(.resid) + ggtitle("ARIMA errors")
```



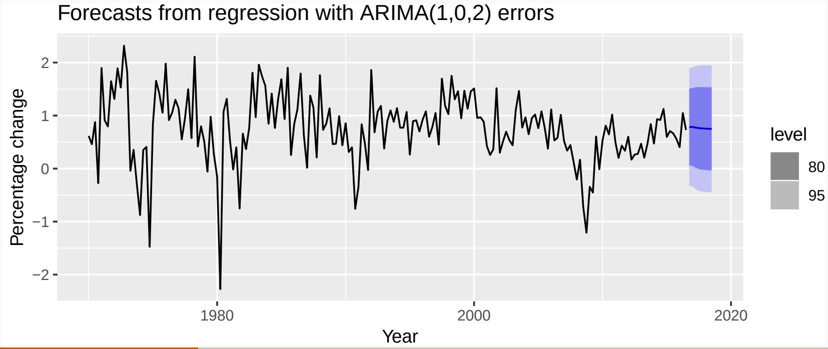
US personal consumption and income

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

```
## # A tibble: 1 x 3
##   .model                           .resid_lb_stat .resid_lb_pval
##   <chr>                                     <dbl>          <dbl>
## 1 ARIMA(Consumption ~ Income)                6.35          0.500
```

US personal consumption and income

```r
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")
```



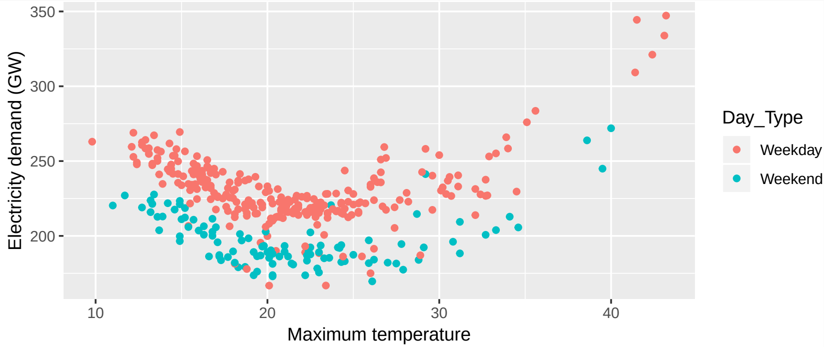
Forecasting

- 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

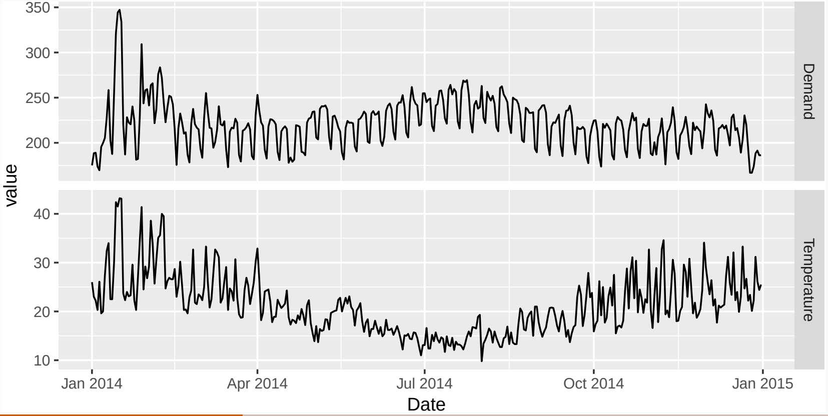
Model daily electricity demand as a function of temperature using quadratic regression with ARMA errors.

```r
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 %>%
  gather("var", "value", Demand, Temperature) %>%
  ggplot(aes(x = Date, y = value)) + geom_line() +
  facet_grid(vars(var), scales = "free_y")
```



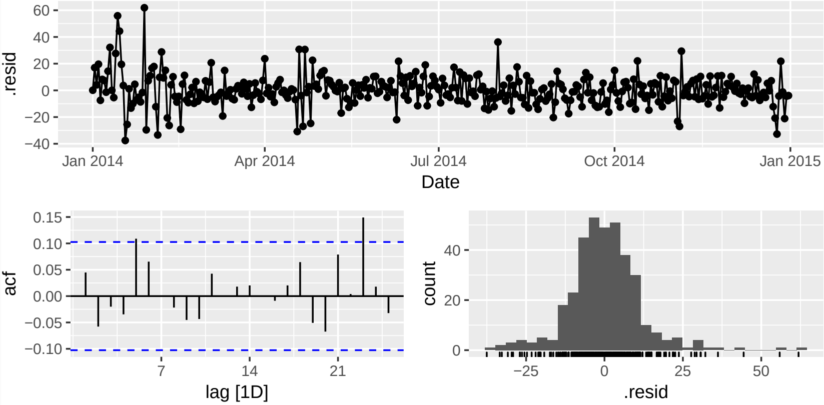
Daily electricity demand

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

```
## Series: Demand
## Model: LM w/ ARIMA(1,1,1)(2,0,1)[7] errors
##
## Coefficients:
##          ar1      ma1     sar1    sar2    sma1
##       0.7170  -0.9362  -0.6999  0.1911  0.8405
## s.e.  0.0594   0.0330   0.1118  0.0587  0.0999
##       xreg.Temperature  xreg.Day_Type == "Weekday"
##                 1.5639                      30.203
## s.e.            0.1562                       1.552
##
## sigma^2 estimated as 142.2:  log likelihood=-1416
## AIC=2847   AICc=2848   BIC=2879
```

Daily electricity demand

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



Daily electricity demand

```r
augment(fit) %>%
  features(.resid, ljung_box, dof = 8, lag = 14)
```

```
## # A tibble: 1 x 3
##   .model                                 .resid_lb_stat .resid_lb_pval
##   <chr>                                           <dbl>          <dbl>
## 1 "ARIMA(Demand ~ Temperature~                     11.2         0.0826
```

Daily electricity demand

```r
# Forecast one day ahead
vic_next_day <- new_data(vic_elec_daily, 1) %>%
  mutate(Temperature = 26, Day_Type = "Holiday")
forecast(fit, vic_next_day)
```

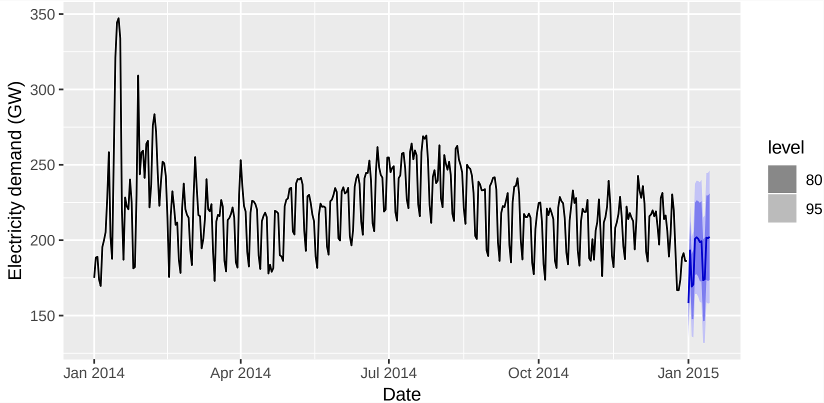
```
## # A fable: 1 x 6 [?]
## # Key:     .model [1]
##   .model Date       Demand .distribution Temperature
##   <chr>  <date>      <dbl> <dist>              <dbl>
## 1 "ARIM~ 2015-01-01   158. N(158, 142)            26
## # ... with 1 more variable: Day_Type <chr>
```

Daily electricity demand

```r
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

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



Outline

Stochastic & deterministic trends

Deterministic trend

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

where η_t is ARMA process.

Stochastic & deterministic trends

Deterministic trend

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

where η_t is ARMA process.

Stochastic trend

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

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

Stochastic & deterministic trends

Deterministic trend

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

where η_t is ARMA process.

Stochastic trend

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

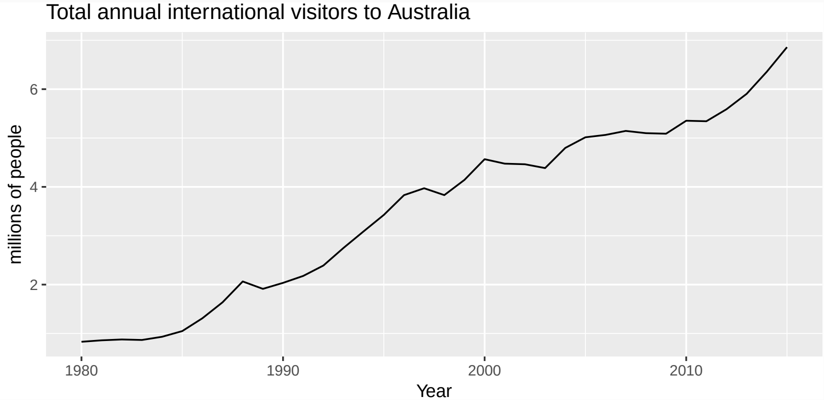
where η_t is ARIMA process with $d \geq 1$.

Difference both sides until η_t is stationary:

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

where η'_t is ARMA process.

International visitors



International visitors

Deterministic trend

```r
fit_deterministic <- aus_visitors %>%
  model(Deterministic = ARIMA(value ~ trend() + pdq(d = 0)))
report(fit_deterministic)
```

```
## Series: value
## Model: LM w/ ARIMA(2,0,0) errors
##
## Coefficients:
##         ar1      ar2   trend  intercept
##       1.113  -0.3805  0.1710     0.4156
## s.e.  0.160   0.1585  0.0088     0.1897
##
## sigma^2 estimated as 0.02979:  log likelihood=13.6
## AIC=-17.2   AICc=-15.2   BIC=-9.28
```

International visitors

Deterministic trend

```
fit_deterministic <- aus_visitors %>%
  model(Deterministic = ARIMA(value ~ trend() + pdq(d = 0)))
report(fit_deterministic)
```

```
## Series: value
## Model: LM w/ ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1      ar2   trend  intercept
##        1.113  -0.3805  0.1710     0.4156
## s.e.   0.160   0.1585  0.0088     0.1897
##
## sigma^2 estimated as 0.02979:  log likelihood=13.6
## AIC=-17.2   AICc=-15.2   BIC=-9.28
```

$$y_t = 0.42 + 0.17t + \eta_t$$
$$\eta_t = 1.11\eta_{t-1} - 0.38\eta_{t-2} + \varepsilon_t$$

International visitors

Stochastic trend

```r
fit_stochastic <- aus_visitors %>%
  model(Stochastic = ARIMA(value ~ pdq(d=1)))
report(fit_stochastic)
```

```
## Series: value
## Model: ARIMA(0,1,1) w/ drift
##
## Coefficients:
##          ma1  constant
##       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

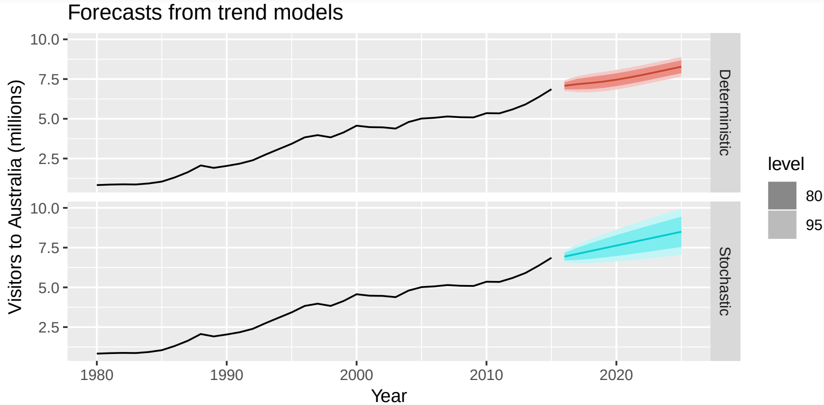
```r
fit_stochastic <- aus_visitors %>%
  model(Stochastic = ARIMA(value ~ pdq(d=1)))
report(fit_stochastic)
```

```
## Series: value
## Model: ARIMA(0,1,1) w/ drift
##
## Coefficients:
##          ma1  constant
##       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
```

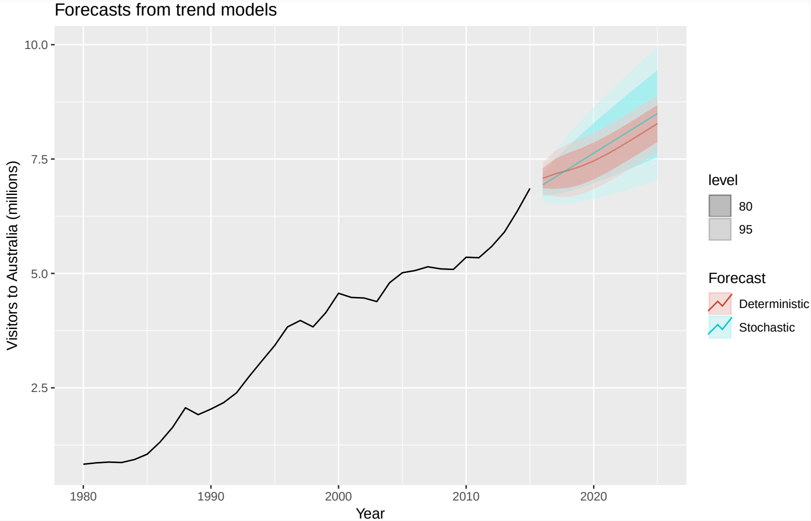
$$y_t - y_{t-1} = 0.17 + \varepsilon_t$$

$$y_t = y_0 + 0.17t + \eta_t$$

International visitors



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

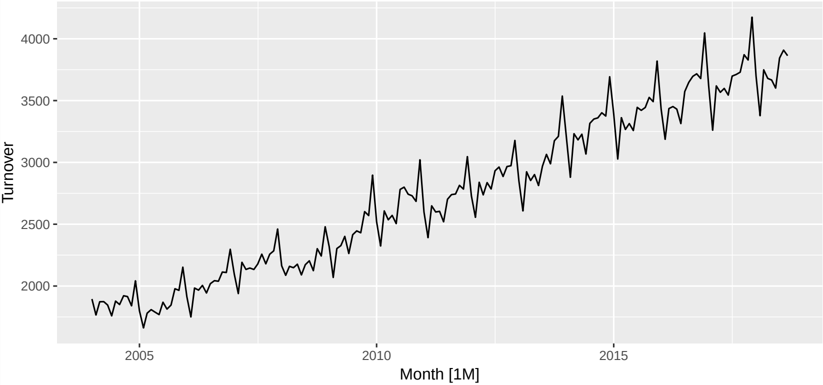
- it allows any length seasonality;
- for data with more than one seasonal period, you can include Fourier terms of different frequencies;
- the seasonal pattern is smooth for small values of K (but more wiggly seasonality can be handled by increasing K);
- the short-term dynamics are easily handled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

Disadvantages

- seasonality is assumed to be fixed

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)
```

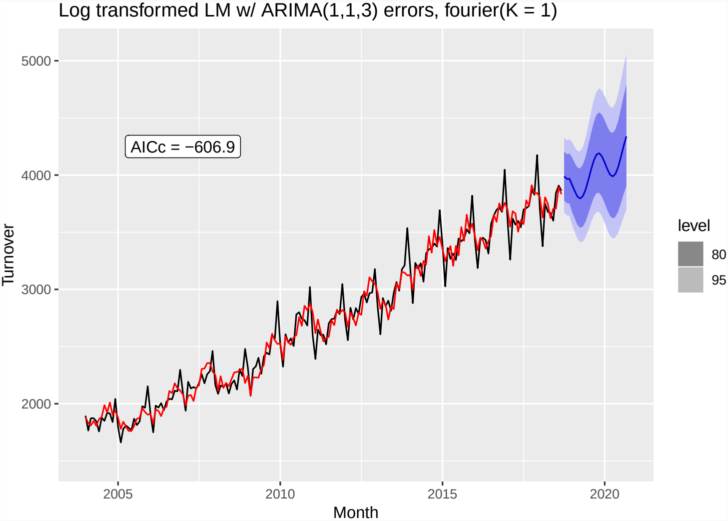


Eating-out expenditure

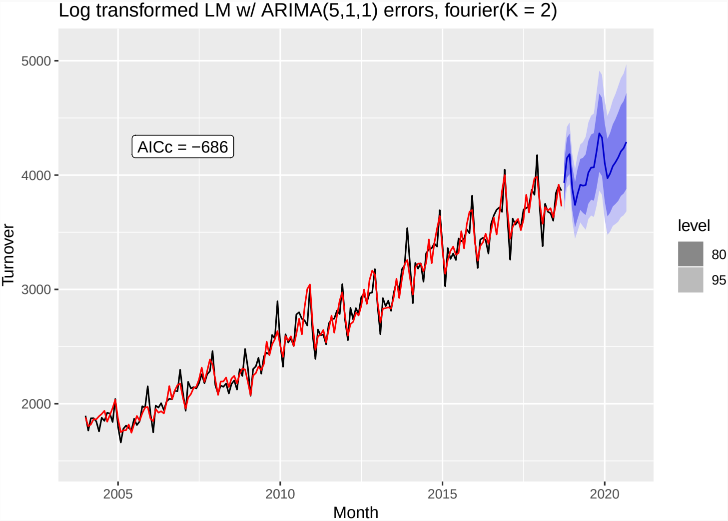
```
fit <- aus_cafe %>% model(
    K = 1 = ARIMA(log(Turnover) ~ fourier(K = 1) + PDQ(0,0,0)),
    K = 2 = ARIMA(log(Turnover) ~ fourier(K = 2) + PDQ(0,0,0)),
    K = 3 = ARIMA(log(Turnover) ~ fourier(K = 3) + PDQ(0,0,0)),
    K = 4 = ARIMA(log(Turnover) ~ fourier(K = 4) + PDQ(0,0,0)),
    K = 5 = ARIMA(log(Turnover)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

.model	sigma	logLik	AIC	AICc	BIC
K = 1	0.0417	311.9	-607.7	-606.9	-582.4
K = 2	0.0327	356.0	-687.9	-686.0	-649.9
K = 3	0.0276	385.9	-751.8	-750.4	-720.1
K = 4	0.0234	418.3	-804.7	-801.3	-754.0
K = 5	0.0179	464.2	-902.5	-900.2	-861.3
K = 6	0.0179	465.2	-902.4	-899.8	-858.0

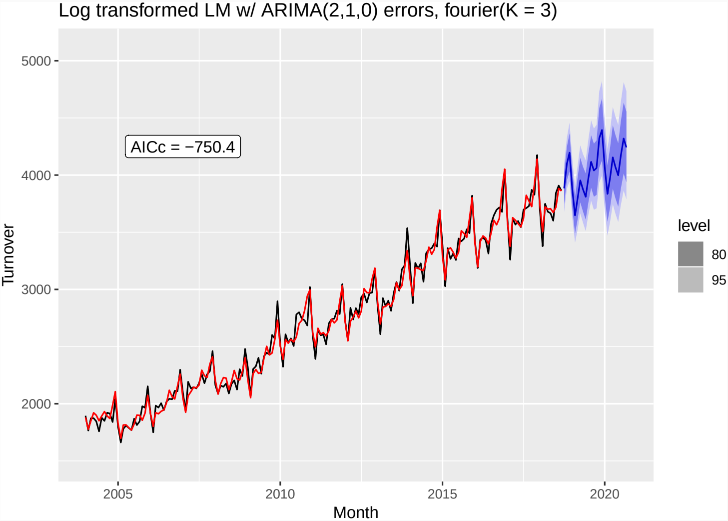
Eating-out expenditure



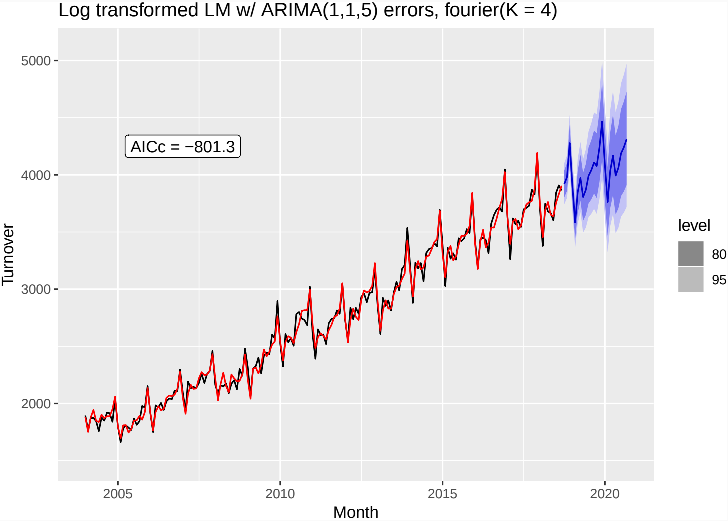
Eating-out expenditure



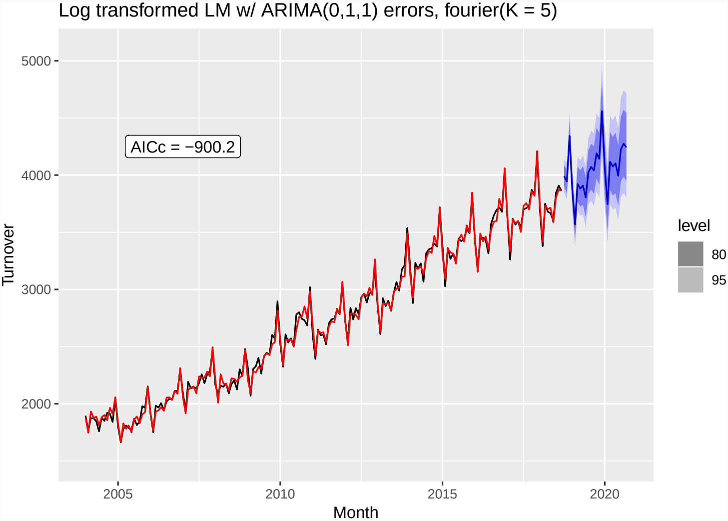
Eating-out expenditure



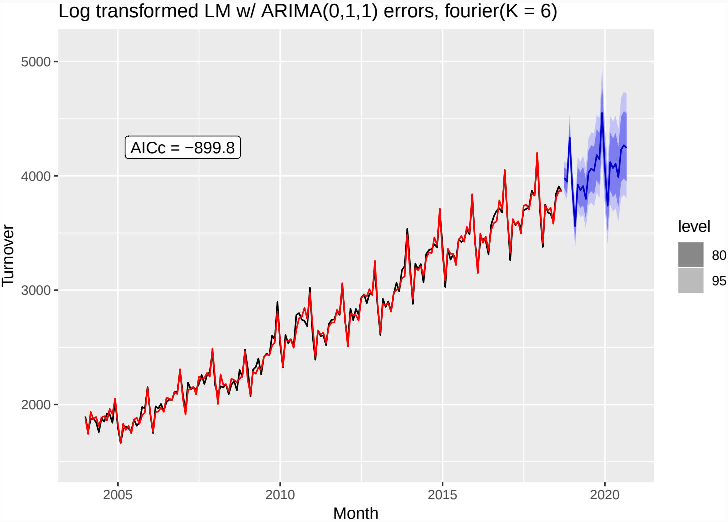
Eating-out expenditure



Eating-out expenditure



Eating-out expenditure



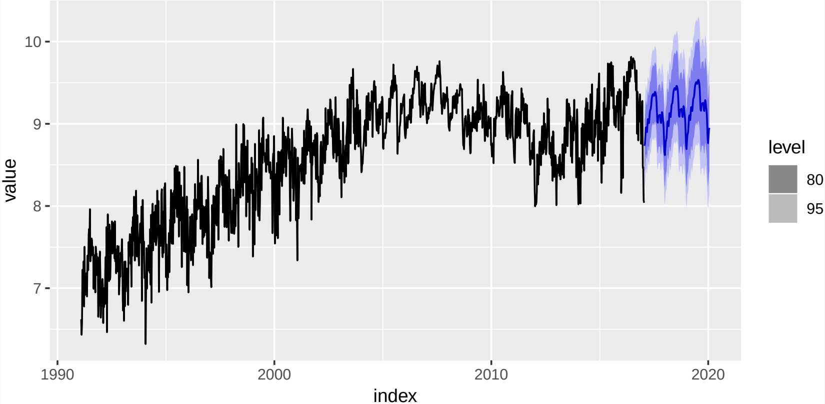
Example: weekly gasoline products

```r
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    C1_52    S1_52   C2_52   S2_52   C3_52
##        -0.8934  -0.1121  -0.2300  0.0420  0.0317  0.0832
## s.e.    0.0132   0.0123   0.0122  0.0099  0.0099  0.0094
##          S3_52   C4_52   S4_52    C5_52   S5_52    C6_52
##         0.0346  0.0185  0.0398  -0.0315  0.0009  -0.0522
## s.e.    0.0094  0.0092  0.0092   0.0091  0.0091   0.0090
##         S6_52    C7_52   S7_52   C8_52   S8_52    C9_52
##         0.000  -0.0173  0.0053  0.0075  0.0048  -0.0024
## s.e.    0.009   0.0090  0.0090  0.0090  0.0090   0.0090
##           S9_52   C10_52   S10_52   C11_52  S11_52   C12_52
##         -0.0035   0.0151  -0.0037  -0.0144  0.0191  -0.0227
## s.e.     0.0090   0.0090   0.0090   0.0090  0.0090   0.0090
##          S12_52   C13_52  S13_52  intercept
##         -0.0052  -0.0035  0.0038     0.0014
## s.e.     0.0090   0.0090  0.0090     0.0007
##
```

Example: weekly gasoline products

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



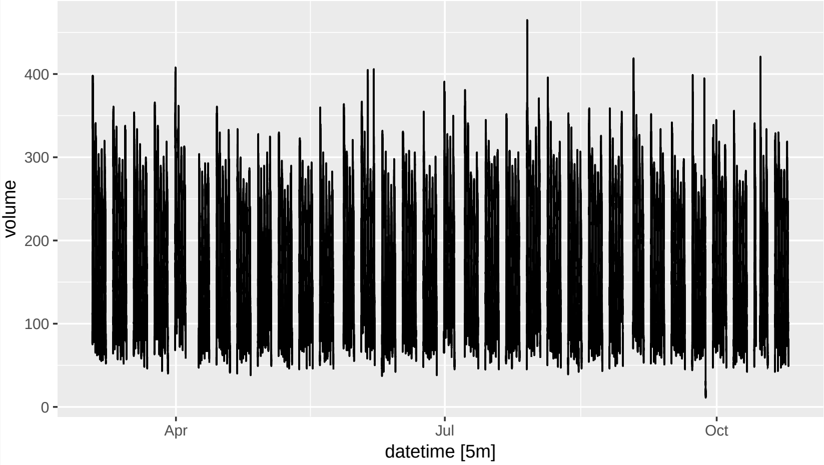
5-minute call centre volume

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

```
## # A tsibble: 27,716 x 4 [5m] <UTC>
##     time   date       datetime            volume
##     <time> <date>     <dttm>               <dbl>
##  1 07:00   2003-03-03 2003-03-03 07:00:00    111
##  2 07:05   2003-03-03 2003-03-03 07:05:00    113
##  3 07:10   2003-03-03 2003-03-03 07:10:00     76
##  4 07:15   2003-03-03 2003-03-03 07:15:00     82
##  5 07:20   2003-03-03 2003-03-03 07:20:00     91
##  6 07:25   2003-03-03 2003-03-03 07:25:00     87
##  7 07:30   2003-03-03 2003-03-03 07:30:00     75
##  8 07:35   2003-03-03 2003-03-03 07:35:00     89
##  9 07:40   2003-03-03 2003-03-03 07:40:00     99
## 10 07:45   2003-03-03 2003-03-03 07:45:00    125
```

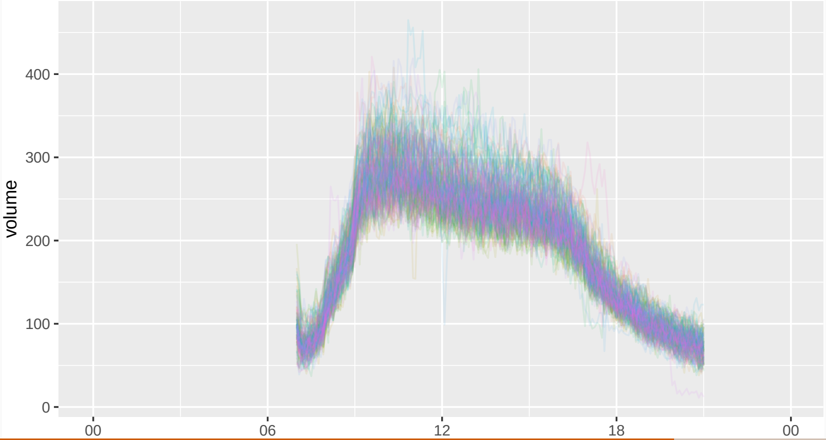
5-minute call centre volume

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



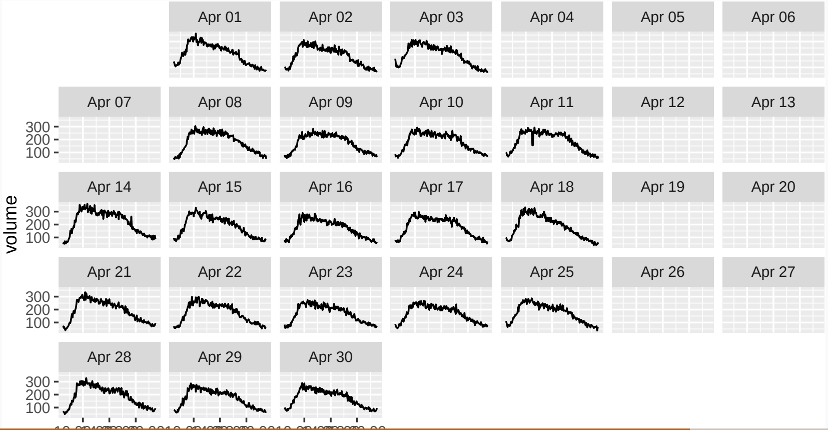
5-minute call centre volume

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



5-minute call centre volume

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



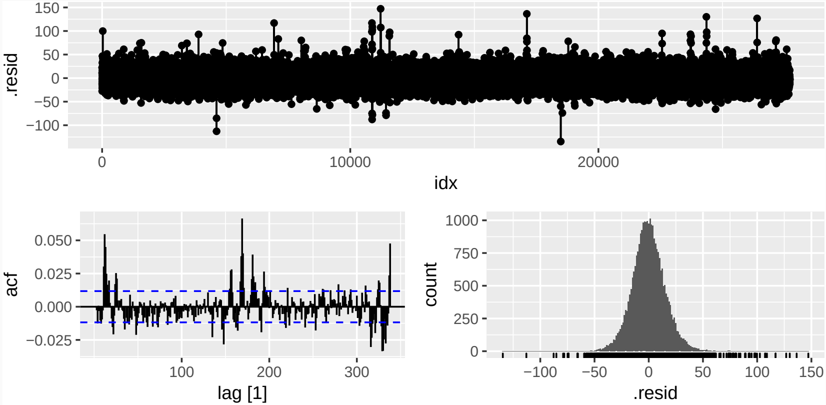
5-minute call centre volume

```
calls_mdl <- calls %>% mutate(idx = row_number()) %>% update_tsibble(index = idx)
fit <- calls_mdl %>%
  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     C1_169   S1_169
##       0.9894  -0.7383  -0.0333  -0.0282  -79.0702  55.2985
## s.e.  0.0010   0.0061   0.0075   0.0060    0.7001   0.7007
##         C2_169   S2_169   C3_169    S3_169  C4_169  S4_169
##       -32.3615  13.7417  -9.3180  -13.6446  -2.791  -9.508
## s.e.    0.3784   0.3786   0.2725    0.2726   0.223   0.223
##       C5_169   S5_169  C6_169  S6_169  C7_169  S7_169
##       2.8975  -2.2323   3.308   0.174  0.2968   0.857
## s.e.  0.1957   0.1957   0.179   0.179  0.1680   0.168
##        C8_169  S8_169   C9_169   S9_169  C10_169  S10_169
##       -1.3878  0.8633  -0.3410  -0.9754   0.8050  -1.1803
## s.e.   0.1604  0.1604   0.1548   0.1548   0.1507   0.1507
##       intercept
##         192.079
## s.e.      1.769
```

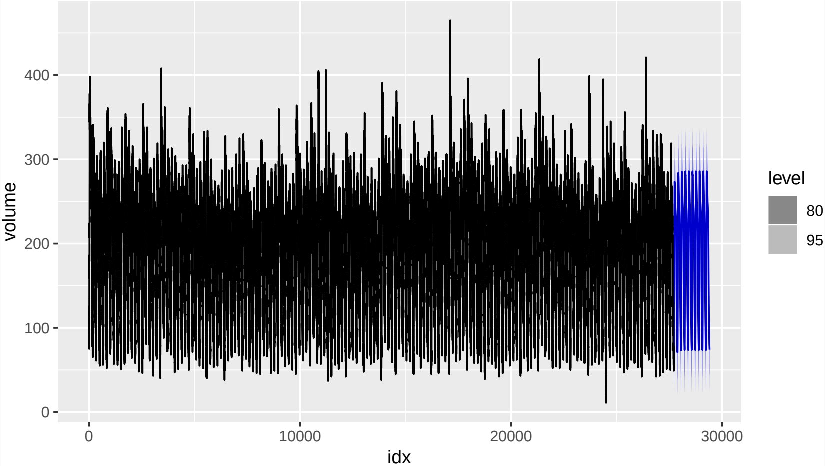
5-minute call centre volume

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



5-minute call centre volume

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



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

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

where η_t is an ARIMA process.

Lagged predictors

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

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

where η_t is an ARIMA process.

Rewrite model as

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

Lagged predictors

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

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

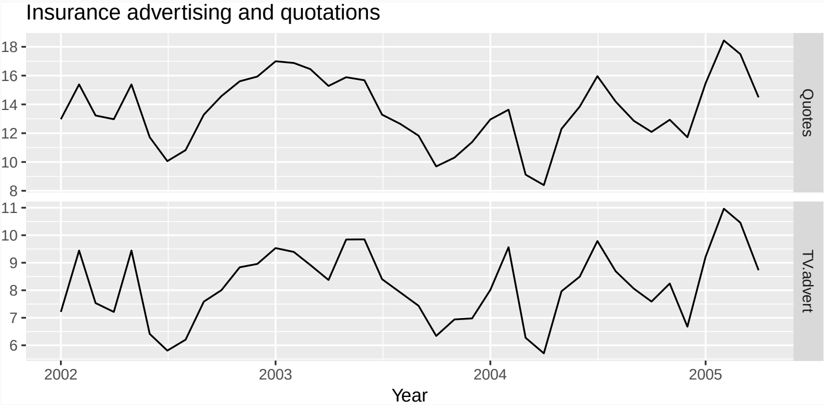
where η_t is an ARIMA process.

Rewrite model as

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

- $\nu(B)$ 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
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) + 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

```r
glance(fit)
```

Lag order	sigma	logLik	AIC	AICc	BIC
0	0.5148	-28.28	66.56	68.33	75.01
1	0.4576	-24.04	58.09	59.85	66.53
2	0.4637	-24.02	60.03	62.58	70.17
3	0.4535	-22.16	60.31	64.96	73.83

Example: Insurance quotes and TV adverts

```r
fit <- insurance %>%
  model(ARIMA(Quotes ~ pdq(3, 0, 0) + TV.advert + lag(TV.advert)))
report(fit)
```

```
## Series: Quotes
## Model: LM w/ ARIMA(3,0,0) errors
##
## Coefficients:
##          ar1      ar2     ar3  xreg.TV.advert
##       1.4117  -0.9317  0.3591          1.2564
## s.e.  0.1698   0.2545  0.1592          0.0667
##       xreg.lag(TV.advert)  intercept
##                    0.1625     2.0393
## s.e.               0.0591     0.9931
##
## 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
fit <- insurance %>%
  model(ARIMA(Quotes ~ pdq(3, 0, 0) + TV.advert + lag(TV.advert)))
report(fit)
```

```
## Series: Quotes
## Model: LM w/ ARIMA(3,0,0) errors
##
## Coefficients:
##          ar1      ar2     ar3  xreg.TV.advert
##       1.4117  -0.9317  0.3591          1.2564
## s.e.  0.1698   0.2545  0.1592          0.0667
##       xreg.lag(TV.advert)  intercept
##                    0.1625     2.0393
## s.e.               0.0591     0.9931
##
## sigma^2 estimated as 0.2165:  log likelihood=-23.89
## AIC=61.78   AICc=65.28   BIC=73.6
```

$$
\begin{aligned}
y_t &= 2.04 + NAx_t + NAx_{t-1} + \eta_t,\\
\eta_t &= 1.41\eta_{t-1} - 0.93\eta_{t-2} + 0.36\eta_{t-3} + \varepsilon_t,
\end{aligned}
$$

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)
```

Transfer function models

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

where η_t is an ARMA process. So

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

Transfer function models

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

where η_t is an ARMA process. So

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

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

Transfer function models

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

where η_t is an ARMA process. So

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

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

- ARMA models are rational approximations to general transfer functions of ε_t.
- We can also replace $\nu(B)$ by a rational approximation.
- There is no R package for forecasting using a general transfer function approach.