

Rob J Hyndman
George Athanasopoulos

FORECASTING

PRINCIPLES AND PRACTICE

A comprehensive introduction to the latest forecasting methods using R. Learn to improve your forecast accuracy using dozens of real data examples.



3RD EDITION

 **OTexts**
OPEN TEXTS FOR PRACTICE

10. Dynamic regression models

10.3 Forecasting

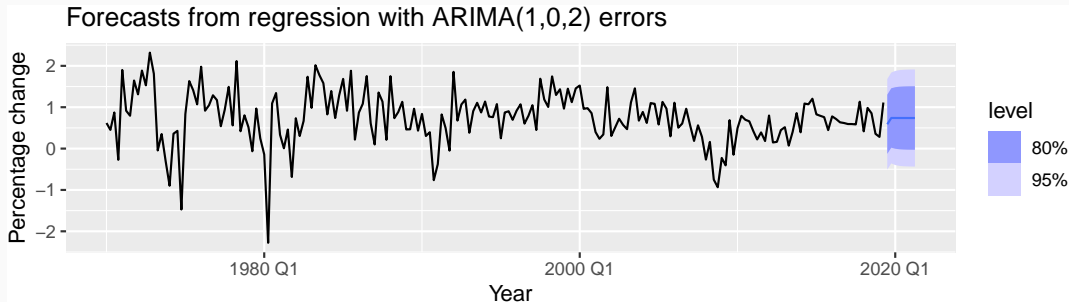
OTexts.org/fpp3/

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.

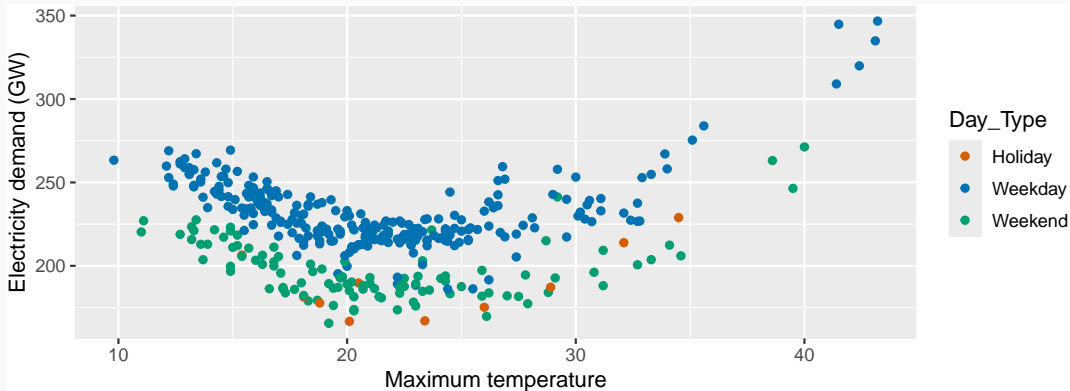
US personal consumption and income

```
fit <- us_change |> model(ARIMA(Consumption ~ 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")
```



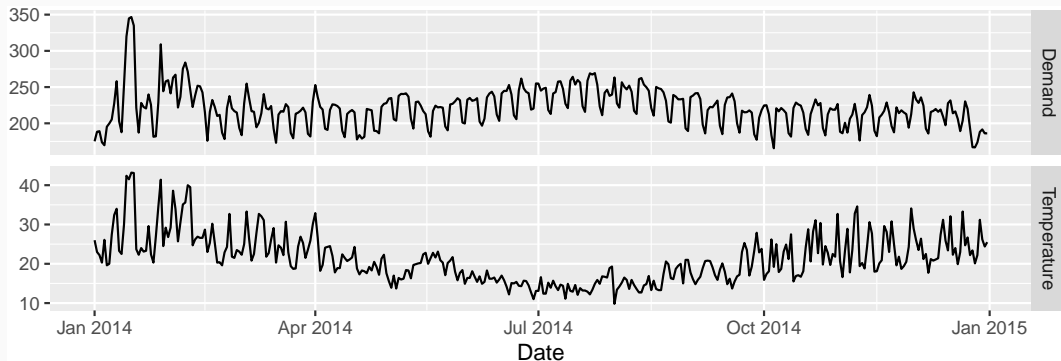
Daily electricity demand

```
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") +  
  labs(y = "")
```



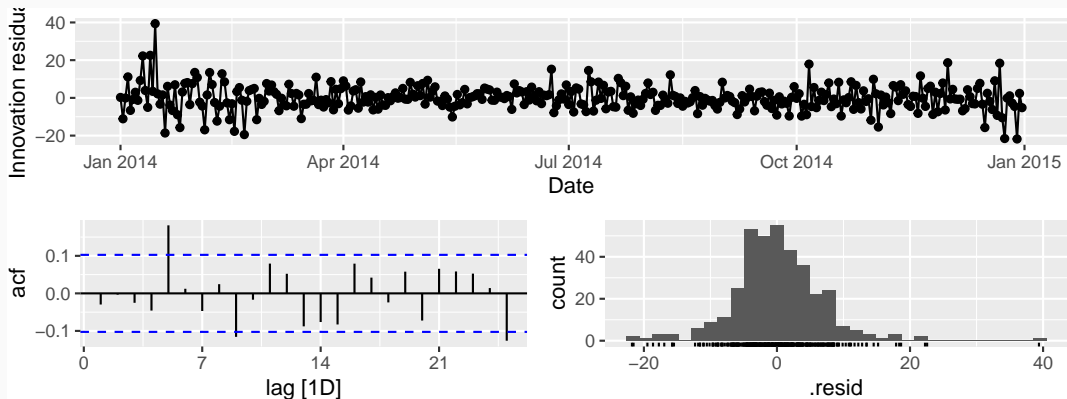
Daily electricity demand

```
fit <- vic_elec_daily |>
  model(arima = 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

```
gg_tsresiduals(fit)
```



Daily electricity demand

```
augment(fit) |>  
  features(.resid, ljung_box, dof = 6, lag = 14)
```

```
## # A tibble: 1 x 3  
##   .model lb_stat lb_pvalue  
##   <chr>    <dbl>    <dbl>  
## 1 arima      28.4    0.000404
```

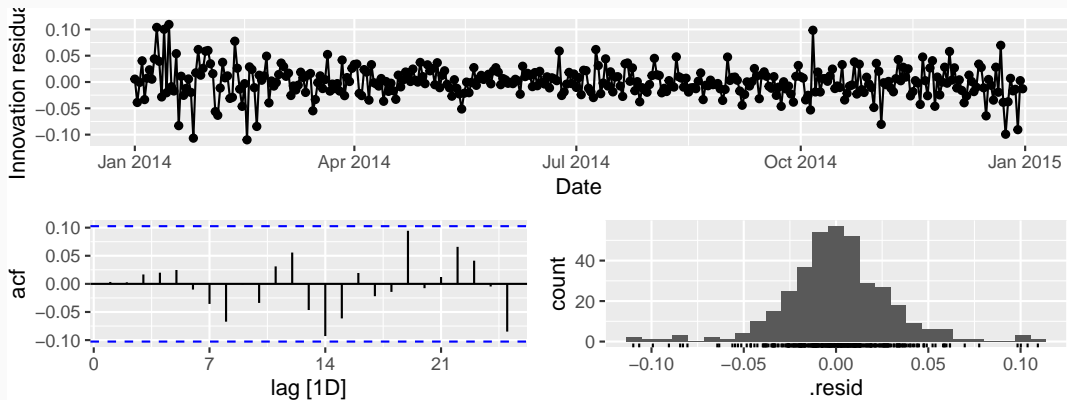

Daily electricity demand - revised

```
fit <- vic_elec_daily |>
  model(arima = ARIMA(log(Demand) ~ Temperature + I(Temperature^2) +
    (Day_Type == "Weekday"), stepwise = FALSE,
    order_constraint = (p+q <=8 & P+Q <=5)))
report(fit)
```

```
## Series: Demand
## Model: LM w/ ARIMA(5,1,3)(2,0,0)[7] errors
## Transformation: log(Demand)
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ar5      ma1      ma2      ma3      sar1
##          0.0262  -0.029   0.539  -0.087   0.185  -0.212  -0.141  -0.622   0.3217
## s.e.      0.2311   0.148   0.117   0.061   0.071   0.230   0.179   0.128   0.0594
##          sar2  Temperature  I(Temperature^2)  Day_Type == "Weekday"TRUE
##          0.3866      -0.0311                8e-04                0.1396
## s.e.      0.0593          0.0004                0e+00                0.0061
##
```

Daily electricity demand - revised

```
gg_tsresiduals(fit)
```



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 2015-01-01 t(N(5.1, 0.00087)) 163.         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")
```

