MONASH University

MONASH BUSINESS SCHOOL

# ETC3550
# Applied forecasting for business and economics

## Ch11. Advanced forecasting methods

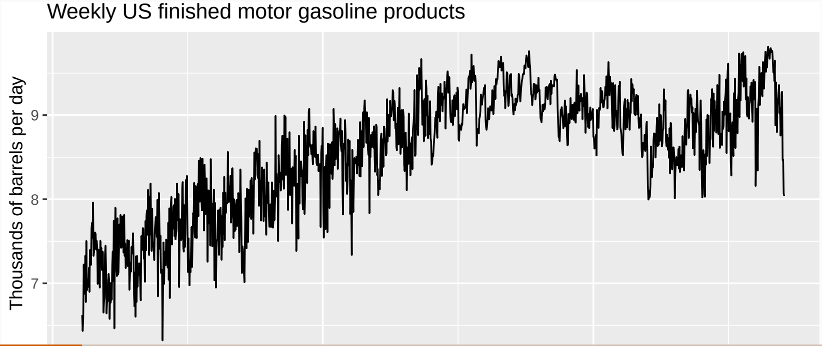
OTexts.org/fpp3/

# Outline

# Outline

# Examples

```
gasoline <- as_tsibble(fpp2::gasoline)
gasoline %>% autoplot(value) +
  labs(x = "Year", y = "Thousands of barrels p
       title = "Weekly US finished motor gasol
```



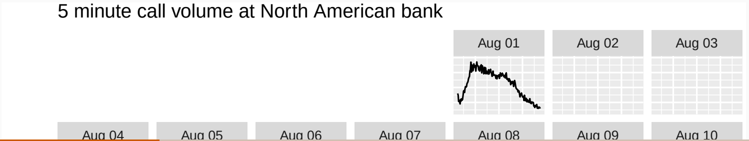
# Examples

```
calls <- read_tsv("http://robjhyndman.com/data
  gather("date", "volume", -X1) %>% transmute(
    time = X1, date = as.Date(date, format = "
    datetime = as_datetime(date) + time, volum
  as_tsibble(index = datetime)
calls %>%
  fill_gaps() %>%
  autoplot(volume) +
  labs(x = "Weeks", y = "Call volume",
       title = "5 minute call volume at North
```

5 minute call volume at North American bank

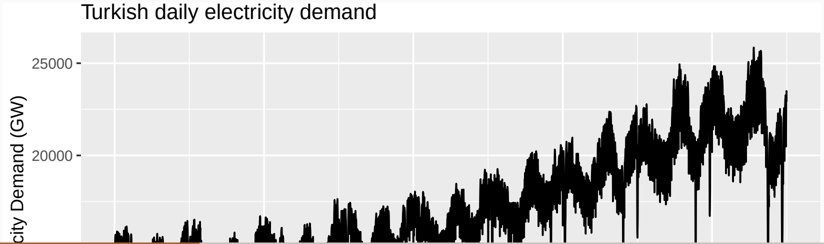
# Examples

```
library(sugrrants)
calls %>%
  filter(yearmonth(date) == yearmonth("2003 Au
  ggplot(aes(x = time, y = volume)) +
  geom_line() +
  facet_calendar(date) +
  labs(x = "Weeks", y = "Call volume",
       title = "5 minute call volume at North
```



# Examples

```
turkey_elec <- read_csv("data/turkey_elec.csv"
  mutate(Date = seq(ymd("2000-01-01"), ymd("20
  as_tsibble(index = Date)
turkey_elec %>% autoplot(Demand) +
  labs(title = "Turkish daily electricity dema
       x = "Year", y = "Electricity Demand (GW
```



# TBATS model

## TBATS

**T**rigonometric terms for seasonality
**B**ox-Cox transformations for heterogeneity
**A**RMA errors for short-term dynamics
**T**rend (possibly damped)
**S**easonal (including multiple and non-integer periods)

# TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t$$

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= \phantom{-}s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

# TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

# TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

$M$ seasonal periods

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

# TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

$M$ seasonal periods

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

global and local trend

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

# TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

$M$ seasonal periods

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t$$

global and local trend

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

ARMA error

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= \phantom{-}s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

# TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

*M* seasonal periods

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t$$

global and local trend

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

ARMA error

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \,\mathrm{c}$$

$$s_{j,t}^{(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t$$

Fourier-like seasonal terms

# TBATS model

$y_t$ = observation at time $t$

$$y_t^{(\omega)} = \begin{cases} (y_t^{\omega} - 1)/\omega & \text{if } \omega \neq 0; \end{cases}$$

Box-Cox transformation

**TBATS**

**Trigonometric**

**Box-Cox**

**ARMA**

**Trend**

**Seasonal**

$$y_t^{(\omega)} = \ell_t \quad \ldots \quad d_t$$

$M$ seasonal periods

$$\ell_t = \ell_t \ldots$$

$$b_t = (1 \ldots$$

global and local trend

$$d_t = \sum_{i=1}^{p} \ldots$$

ARMA error

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \ldots$$

$$s_{j,t}^{(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t$$

Fourier-like seasonal terms

# Complex seasonality

```
gasoline %>% tbats() %>% forecast() %>% autoplot()
```

# Complex seasonality

```
calls %>% tbats() %>% forecast() %>% autoplot()
```

# Complex seasonality

```
telec %>% tbats() %>% forecast() %>% autoplot()
```

# TBATS model

## TBATS

**T**rigonometric terms for seasonality

**B**ox-Cox transformations for heterogeneity

**A**RMA errors for short-term dynamics

**T**rend (possibly damped)

**S**easonal (including multiple and non-integer periods)

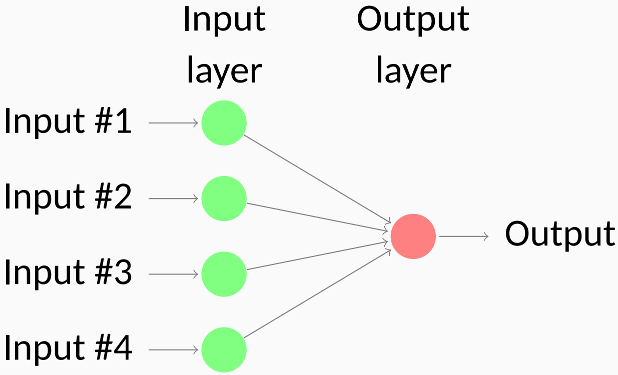
- Handles non-integer seasonality, multiple seasonal periods.
- Entirely automated
- Prediction intervals often too wide
- Very slow on long series

# Outline

# Outline

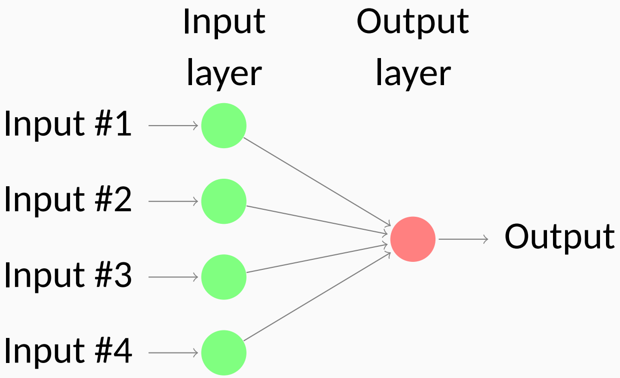
# Neural network models

## Simplest version: linear regression



# Neural network models

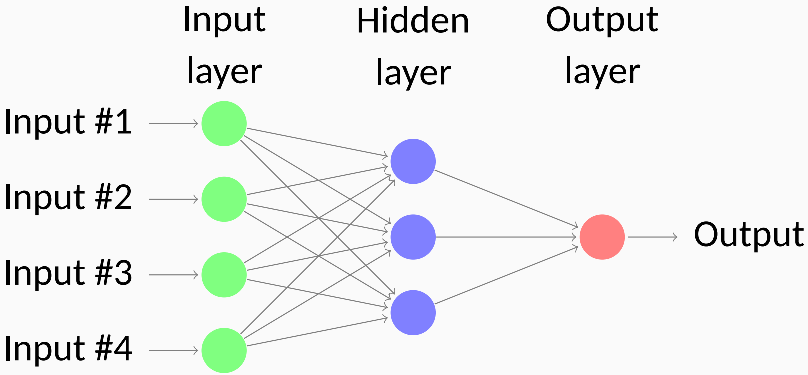
## Simplest version: linear regression




- Coefficients attached to predictors are called "weights".
- Forecasts are obtained by a linear combination of inputs.
- Weights selected using a "learning algorithm" that

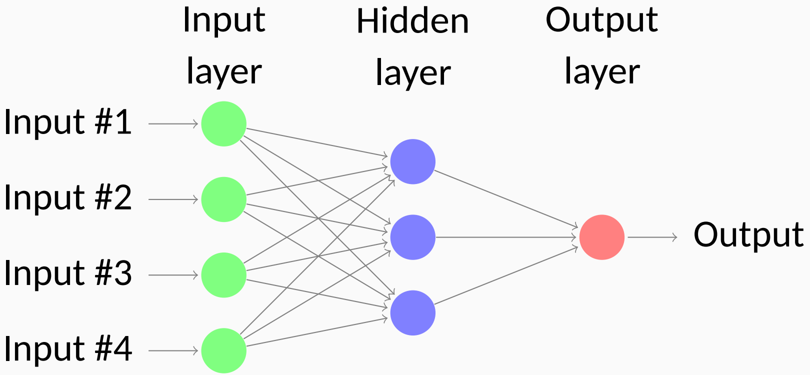
# Neural network models

## Nonlinear model with one hidden layer



# Neural network models

## Nonlinear model with one hidden layer




- A **multilayer feed-forward network** where each layer of nodes receives inputs from the previous layers.

# Neural network models

Inputs to hidden neuron $j$ linearly combined:

$$z_j = b_j + \sum_{i=1}^{4} w_{i,j} x_i.$$

Modified using nonlinear function such as a sigmoid:

$$s(z) = \frac{1}{1 + e^{-z}},$$

This tends to reduce the effect of extreme input values, thus making the network somewhat robust to outliers.

# Neural network models

- Weights take random values to begin with, which are then updated using the observed data.
- There is an element of randomness in the predictions. So the network is usually trained several times using different random starting points, and the results are averaged.
- Number of hidden layers, and the number of nodes in each hidden layer, must be specified in advance.

# NNAR models

- Lagged values of the time series can be used as inputs to a neural network.
- NNAR($p, k$): $p$ lagged inputs and $k$ nodes in the single hidden layer.
- NNAR($p, 0$) model is equivalent to an ARIMA($p, 0, 0$) model but without stationarity restrictions.
- Seasonal NNAR($p, P, k$): inputs $(y_{t-1}, y_{t-2}, \dots, y_{t-p}, y_{t-m}, y_{t-2m}, y_{t-Pm})$ and $k$ neurons in the hidden layer.
- NNAR$(p, P, 0)_m$ model is equivalent to an ARIMA$(p, 0, 0)(P,0,0)_m$ model but without stationarity restrictions.

# NNAR models in R

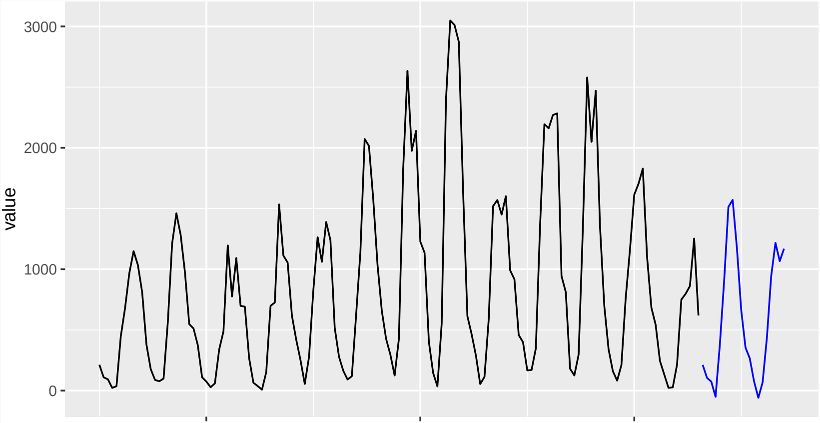
- The `nnetar()` function fits an NNAR$(p, P, k)_m$ model.
- If $p$ and $P$ are not specified, they are automatically selected.
- For non-seasonal time series, default $p =$ optimal number of lags (according to the AIC) for a linear AR($p$) model.
- For seasonal time series, defaults are $P = 1$ and $p$ is chosen from the optimal linear model fitted to the seasonally adjusted data.
- Default $k = (p + P + 1)/2$ (rounded to the nearest integer).

# Sunspots

- Surface of the sun contains magnetic regions that appear as dark spots.
- These affect the propagation of radio waves and so telecommunication companies like to predict sunspot activity in order to plan for any future difficulties.
- Sunspots follow a cycle of length between 9 and 14 years.

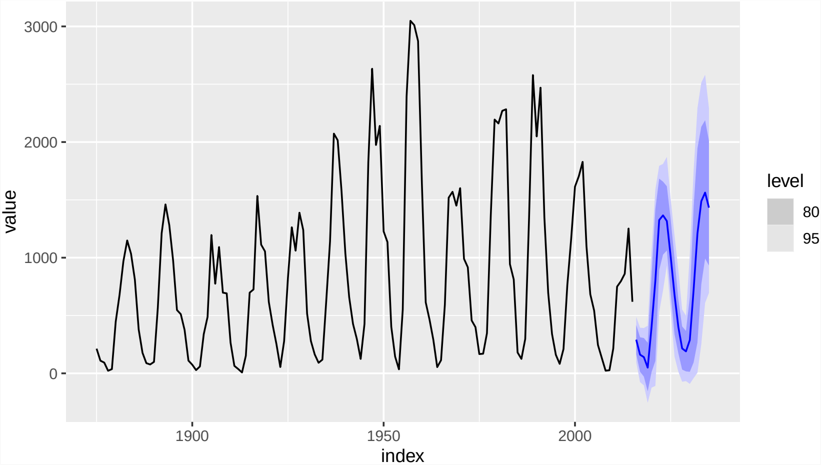
# NNAR(9,5) model for sunspots

```
sunspots <- as_tsibble(fpp2::sunspotarea)
fit <- sunspots %>% model(NNETAR(value))
fit %>% forecast(h=20, times = 1) %>%
  autoplot(sunspots, level = NULL)
```



# Prediction intervals by simulation

```
fit %>% forecast(h=20) %>%
  autoplot(sunspots)
```



# Outline