MONASH University

MONASH BUSINESS SCHOOL

Tidy Forecasting in R

Rob J Hyndman

useR2018

Outline

forecast package

Pre 2003	Private functions used for consulting projects
July/August 2003	**ets** and **thetaf** added
August 2006	v1.0 available on CRAN
May 2007	**auto.arima** added
May 2010	**arfima** added
Feb/March 2011	**tslm**, **stlf**, **naive**, **snaive** added
August 2011	v3.0. Box Cox transformations added
December 2011	**tbats** added
April 2012	Package moved to github
November 2012	v4.0. **nnetar** added
June 2013	Major speed-up of **ets**
February 2016	v7.0. Added ggplot2 graphics
February 2017	v8.0. Added **checkresiduals**, **tsCV** and **%>%**
April 2018	v8.3. Added **mstl**
June 2018	$\approx$ 100,000 package downloads per month

fable package

A replacement for the forecast package.

Why change?

- Integrating with tidyverse packages
- Designed for forecasting many related time series
- Consistency of interface using formulas
- Distribution forecasting rather than point + interval
- Flexible transformations
- Sub-daily data and multiple seasonal data handled more easily
- Simpler interface for forecast reconciliation

Outline

Formula model specification

All modelling functions use a formula similar to `lm()` with automated modelling if RHS not specified.

```
t(y) ~ {model specification}
```

LHS: Response

- Defines the response variable from the data
- Specification of transformations (which are automatically back-transformed)

RHS: Specials

- Model specific special functions
- Exogenous regressors (if supported by model)

Outline

Example: Australian eating-out expenditure

```r
library(tsibble)
(cafe <- as_tsibble(fpp2::auscafe))
```

```
## # A tsibble: 426 x 2 [1MONTH]
##        index value
##        <mth> <dbl>
##  1  1982 Apr 0.342
##  2  1982 May 0.342
##  3  1982 Jun 0.329
##  4  1982 Jul 0.338
##  5  1982 Aug 0.332
##  6  1982 Sep 0.342
##  7  1982 Oct 0.358
##  8  1982 Nov 0.375
##  9  1982 Dec 0.433
## 10  1983 Jan 0.369
## # ... with 416 more rows
```

Example: Australian eating-out expenditure

```r
library(fable)
cafe %>% ARIMA(log(value))
```

```
## # A mable: 1 model [1MONTH]
##   data                    model
##   <list>                  <model>
## 1 <tsibble [426 x 2]> ARIMA(2,1,1)(2,1,2)[12]
```

Example: Australian eating-out expenditure

```r
cafe %>% ARIMA(log(value)) %>% summary()
```

```
## Series: log(value)
## ARIMA(2,1,1)(2,1,2)[12]
##
## Coefficients:
##          ar1     ar2    ma1   sar1    sar2   sma1   sma2
##       -0.925  -0.318  0.588  0.724  -0.213  -1.44  0.557
## s.e.   0.182   0.060  0.189  0.174   0.074   0.17  0.149
##
## sigma^2 estimated as 0.000554:  log likelihood=959
## AIC=-1903   AICc=-1903   BIC=-1871
##
## Training set error measures:
##                    ME   RMSE    MAE     MPE MAPE  MASE
## Training set -0.00106 0.0371 0.0267 -0.0506 1.78 0.256
##                 ACF1
## Training set -0.0184
```

Example: Australian eating-out expenditure

```r
cafe %>% ARIMA(log(value)) %>% forecast()
```

```
## # A fable: 1 forecast [1MONTH]
##   data                 model                    forecast
##   <list>               <model>                  <fc>
## 1 <tsibble [426 x 2]> ARIMA(2,1,1)(2,1,2)[12] ~t(N) [h=24]
```

Example: Australian eating-out expenditure

```
cafe %>% ARIMA(log(value)) %>% forecast() %>% summary()
```

```
## # A tsibble: 24 x 4 [1MONTH]
##       index  mean            80%           95%
##       <mth> <dbl>          <hilo>         <hilo>
##  1 2017 Oct  3.81 [3.70, 3.93]80 [3.64, 3.99]95
##  2 2017 Nov  3.79 [3.65, 3.93]80 [3.58, 4.00]95
##  3 2017 Dec  4.17 [3.99, 4.34]80 [3.91, 4.43]95
##  4 2018 Jan  3.73 [3.55, 3.90]80 [3.46, 4.00]95
##  5 2018 Feb  3.40 [3.22, 3.57]80 [3.14, 3.67]95
##  6 2018 Mar  3.77 [3.56, 3.99]80 [3.46, 4.10]95
##  7 2018 Apr  3.70 [3.48, 3.93]80 [3.37, 4.05]95
##  8 2018 May  3.76 [3.52, 4.00]80 [3.40, 4.13]95
##  9 2018 Jun  3.66 [3.41, 3.90]80 [3.29, 4.04]95
## 10 2018 Jul  3.88 [3.61, 4.15]80 [3.48, 4.31]95
## # ... with 14 more rows
```

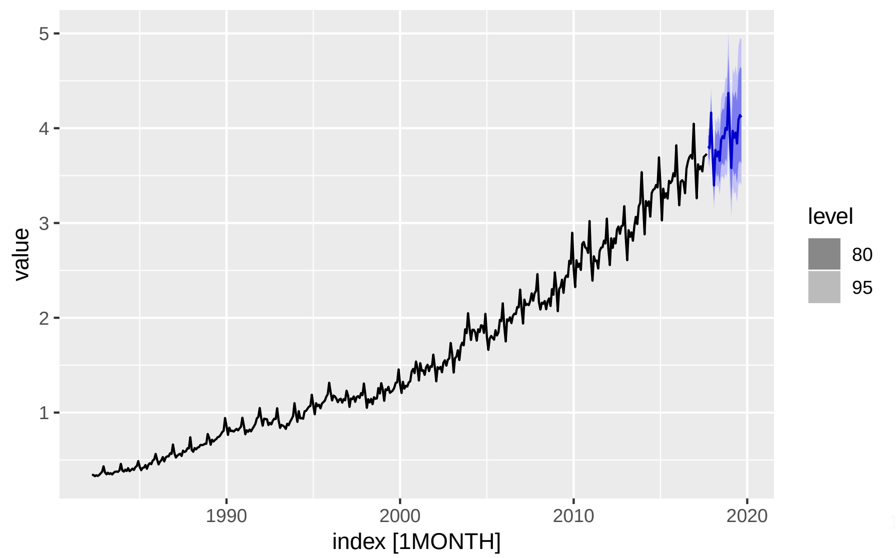
Example: Australian eating-out expenditure

```
cafe %>% ARIMA(log(value)) %>% forecast() %>% summary(level=90)
```

```
## # A tsibble: 24 x 3 [1MONTH]
##        index  mean          90%
##        <mth> <dbl>         <hilo>
##  1 2017 Oct  3.81 [3.66, 3.96]90
##  2 2017 Nov  3.79 [3.62, 3.97]90
##  3 2017 Dec  4.17 [3.95, 4.39]90
##  4 2018 Jan  3.73 [3.50, 3.96]90
##  5 2018 Feb  3.40 [3.18, 3.62]90
##  6 2018 Mar  3.77 [3.51, 4.05]90
##  7 2018 Apr  3.70 [3.42, 3.99]90
##  8 2018 May  3.76 [3.46, 4.07]90
##  9 2018 Jun  3.66 [3.35, 3.98]90
## 10 2018 Jul  3.88 [3.54, 4.24]90
## # ... with 14 more rows
```

Example: Australian eating-out expenditure

```r
cafe %>% ARIMA(log(value)) %>% forecast() %>% autoplot()
```



Outline

Example: Half-hourly electricity demand

```
elecdemand
```

```
## # A tsibble: 17,520 x 4 [30MINUTE]
##    index                    Demand Temperature WorkDay
##    <dttm>                    <dbl>       <dbl>   <dbl>
##  1 2014-01-01 00:00:00   3.91        18.2       0
##  2 2014-01-01 00:30:00   3.67        17.9       0
##  3 2014-01-01 01:00:00   3.50        17.6       0
##  4 2014-01-01 01:30:00   3.34        16.8       0
##  5 2014-01-01 02:00:00   3.20        16.3       0
##  6 2014-01-01 02:30:00   3.10        16.6       0
##  7 2014-01-01 03:00:00   3.04        16.6       0
##  8 2014-01-01 03:30:00   3.01        16.7       0
##  9 2014-01-01 04:00:00   3.02        16.2       0
## 10 2014-01-01 04:30:00   3.03        16.6       0
## # ... with 17,510 more rows
```

Example: Half-hourly electricity demand

```r
fit2 <- ARIMA(elecdemand,
  Demand ~ Temperature + I(Temperature^2) + WorkDay)
summary(fit2)
```

```
## Series: Demand
## Regression with ARIMA(1,1,0)(2,0,2)[2] errors
##
## Coefficients:
##          ar1     sar1    sar2     sma1     sma2  Temperature
##        0.853   -0.181   0.523   -0.066   -0.792       -0.009
## s.e.   0.005    0.015   0.012    0.012    0.011        0.002
##        I(Temperature^2)  WorkDay
##                       0    0.016
## s.e.                  0    0.006
##
## sigma^2 estimated as 0.00846:  log likelihood=16949
## AIC=-33881   AICc=-33881   BIC=-33811
##
## Training set error measures:
##                    ME  RMSE    MAE     MPE MAPE  MASE  ACF1
## Training set 6.51e-06 0.092 0.0634 0.00633 1.39 0.292 0.103
```

```r
forecast(fit2, newdata=elecdemandfuture) %>% autoplot()
```

Outline

Example: Australian prison population

```r
fpp2::prisonLF
```

```
## # A tibble: 1,536 x 5
##    state gender legal     t          count
##    <fct> <fct>  <fct>     <date>     <dbl>
##  1 ACT   Female Remanded 2005-03-01     2
##  2 ACT   Female Remanded 2005-06-01     4
##  3 ACT   Female Remanded 2005-09-01     1
##  4 ACT   Female Remanded 2005-12-01     4
##  5 ACT   Female Remanded 2006-03-01     4
##  6 ACT   Female Remanded 2006-06-01     6
##  7 ACT   Female Remanded 2006-09-01     9
##  8 ACT   Female Remanded 2006-12-01     6
##  9 ACT   Female Remanded 2007-03-01     4
## 10 ACT   Female Remanded 2007-06-01     4
## # ... with 1,526 more rows
```

Example: Australian prison population

```
prison <- fpp2::prisonLF %>%
  mutate(qtr=yearquarter(t)) %>%
  select(-t) %>%
  as_tsibble(index=qtr, key=id(state,gender,legal))
prison
```

```
## # A tsibble: 1,536 x 5 [1QUARTER]
## # Key:       state, gender, legal [32]
##    state gender legal    count     qtr
##    <fct> <fct>  <fct>    <dbl>   <qtr>
##  1 ACT   Female Remanded     2 2005 Q1
##  2 ACT   Female Remanded     4 2005 Q2
##  3 ACT   Female Remanded     1 2005 Q3
##  4 ACT   Female Remanded     4 2005 Q4
##  5 ACT   Female Remanded     4 2006 Q1
##  6 ACT   Female Remanded     6 2006 Q2
##  7 ACT   Female Remanded     9 2006 Q3
##  8 ACT   Female Remanded     6 2006 Q4
##  9 ACT   Female Remanded     4 2007 Q1
## 10 ACT   Female Remanded     4 2007 Q2
## # ... with 1,526 more rows
```

Example: Australian prison population

```r
prison %>% ETS(count)
```

```
## # A mable: 32 models [1QUARTER]
## # Key:     state, gender, legal [32]
##    state gender legal     data                  model
##    <fct> <fct>  <fct>     <list>                <model>
##  1 ACT   Female Remanded  <tsibble [48 x 2]> ETS(M,A,N)
##  2 ACT   Female Sentenced <tsibble [48 x 2]> ETS(A,A,N)
##  3 ACT   Male   Remanded  <tsibble [48 x 2]> ETS(M,N,N)
##  4 ACT   Male   Sentenced <tsibble [48 x 2]> ETS(A,N,N)
##  5 NSW   Female Remanded  <tsibble [48 x 2]> ETS(M,N,M)
##  6 NSW   Female Sentenced <tsibble [48 x 2]> ETS(M,N,M)
##  7 NSW   Male   Remanded  <tsibble [48 x 2]> ETS(M,A,A)
##  8 NSW   Male   Sentenced <tsibble [48 x 2]> ETS(M,A,A)
##  9 NT    Female Remanded  <tsibble [48 x 2]> ETS(M,N,N)
## 10 NT    Female Sentenced <tsibble [48 x 2]> ETS(M,A,A)
## # ... with 22 more rows
```

Example: Australian prison population

```
prison %>% ETS(count) %>% forecast()
```

```
## # A fable: 32 forecasts [1QUARTER]
## # Key:     state, gender, legal [32]
##    state gender legal     data          model     forecast
##    <fct> <fct>  <fct>     <list>        <model>   <fc>
##  1 ACT   Female Remanded  <tsibble [48~ ETS(M,A,N) ~N [h=8]
##  2 ACT   Female Sentenced <tsibble [48~ ETS(A,A,N) ~N [h=8]
##  3 ACT   Male   Remanded  <tsibble [48~ ETS(M,N,N) ~N [h=8]
##  4 ACT   Male   Sentenced <tsibble [48~ ETS(A,N,N) ~N [h=8]
##  5 NSW   Female Remanded  <tsibble [48~ ETS(M,N,M) ~N [h=8]
##  6 NSW   Female Sentenced <tsibble [48~ ETS(M,N,M) ~N [h=8]
##  7 NSW   Male   Remanded  <tsibble [48~ ETS(M,A,A) ~N [h=8]
##  8 NSW   Male   Sentenced <tsibble [48~ ETS(M,A,A) ~N [h=8]
##  9 NT    Female Remanded  <tsibble [48~ ETS(M,N,N) ~N [h=8]
## 10 NT    Female Sentenced <tsibble [48~ ETS(M,A,A) ~N [h=8]
## # ... with 22 more rows
```

Aggregation and reconciliation not yet implemented.

Outline

Equivalent methods: forecast ⟶ fable

```
auto.arima ⟶ ARIMA
ets        ⟶ ETS
tslm/lm    ⟶ LM
tbats      ⟶ TBATS
nnetar     ⟶ NNAR
stlm       ⟶ STL %>%
               modelcomponents(
                 ETS(seasadj),SNAIVE(season))
```

- All functions have a formula interface with automatic modelling if no formula provided.
- All functions produce `mable` class objects.
- Some of these functions not yet implemented

Equivalent methods: forecast $\longrightarrow$ fable

```
naive    ⟶ NAIVE %>% forecast
snaive   ⟶ SNAIVE %>% forecast
thetaf   ⟶ THETA %>% forecast
stlf     ⟶ STL %>%
              modelcomponents(
                ETS(seasadj),SNAIVE(season)) %>%
              forecast
hw       ⟶ HW %>% forecast
holt     ⟶ HOLT %>% forecast
ses      ⟶ SES %>% forecast
splinef  ⟶ SPLINE %>% forecast
croston  ⟶ CROSTON %>% forecast
```

- `forecast` produces `fable` class objects.

Outline

Extending fable

fable simplifies the times-series model development process

Tools to easily create new fable models

- Easily create specials for model formulae
- Focus on model estimation and forecasts

Automatically supported fable functionality

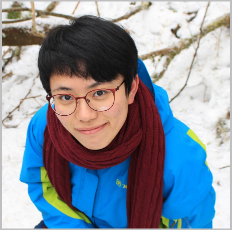
- Transformations and back-transformations (with bias adjustments)
- Plotting tools
- Accuracy measures and evaluation
- Model combinations (hierarchies & ensembles)

Outline

More information

```r
devtools::install_github("tidyverts/tsibble")
devtools::install_github("tidyverts/fable")
```

Di Cook



Earo Wang

Mitchell O'Hara-Wild

Follow our progress

- `tidyverts.org`
- `robjhyndman.com/hyndsight`