

# Tidy Forecasting in R



Rob J Hyndman

ISF 2018

# Outline

- 1 Why change?
- 2 Example: Australian eating-out expenditure
- 3 Example: Australian prison population
- 4 Example: Half-hourly electricity demand
- 5 Equivalent methods
- 6 More information

# forecast package

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

# fable package

A replacement for the forecast package.

## Why change?

- Interacting with tidyverse packages
- Sub-daily data and multiple seasonal data handled more easily
- Consistency of interface
- Distribution forecasting rather than point+interval
- Flexible transformations
- Extensibility
- Simpler interface for forecast reconciliation
- Boosting, combining and ensemble forecasts
- Designed for forecasting many related time series
- Changes will break too much existing code
- Opportunity to re-think forecasting practice

# Outline

- 1 Why change?
- 2 Example: Australian eating-out expenditure
- 3 Example: Australian prison population
- 4 Example: Half-hourly electricity demand
- 5 Equivalent methods
- 6 More information

# Example: Australian eating-out expenditure

fpp2: auscafe

##		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Se
##	1982				0.342	0.342	0.329	0.339	0.332	0.34
##	1983	0.369	0.348	0.366	0.351	0.360	0.347	0.364	0.376	0.37
##	1984	0.389	0.377	0.398	0.383	0.414	0.382	0.393	0.409	0.39
##	1985	0.426	0.392	0.416	0.420	0.446	0.407	0.449	0.466	0.45
##	1986	0.504	0.453	0.480	0.497	0.531	0.485	0.526	0.538	0.53
##	1987	0.572	0.525	0.544	0.558	0.565	0.542	0.599	0.584	0.59
##	1988	0.605	0.586	0.625	0.612	0.630	0.635	0.659	0.656	0.66
##	1989	0.733	0.661	0.713	0.694	0.710	0.722	0.741	0.746	0.76
##	1990	0.858	0.764	0.840	0.805	0.809	0.799	0.815	0.828	0.81
##	1991	0.862	0.771	0.813	0.797	0.821	0.801	0.829	0.854	0.88
##	1992	0.938	0.862	0.936	0.932	0.929	0.869	0.891	0.875	0.91
##	1993	0.918	0.838	0.870	0.862	0.852	0.828	0.882	0.867	0.90
##	1994	0.985	0.902	1.015	0.939	0.941	0.935	1.013	1.018	1.04
##	1995	1.076	0.982	1.099	1.068	1.083	1.045	1.094	1.110	1.12
##	1996	1.213	1.128	1.180	1.169	1.146	1.109	1.138	1.146	1.10
##	1997	1.180	1.060	1.148	1.141	1.170	1.113	1.165	1.173	1.15
##	1998	1.186	1.050	1.141	1.107	1.144	1.088	1.162	1.145	1.14

# Example: Australian eating-out expenditure

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

```
## # 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

```
library(fable)
cafe %>% ETS(value)
```

```
## # A tibble: 1 x 2
##   data          model
##   <list>        <list>
## 1 <tsibble [426 x 2]> <ETS(M,A,M)>
```



# Example: Australian eating-out expenditure

```
cafe %>% ETS(value) %>% summary()
```

```
## ETS(M,A,M)
##
## Call:
## ETS(data = ., formula = value)
##
## Smoothing parameters:
##   alpha = 0.6263
##   beta  = 0.0065
##   gamma = 0.0755
##
## Initial states:
##   l = 0.3477
##   b = 0.0038
##   s = 0.996 0.936 1.01 1.15 1.01 1.01
##         0.983 0.991 0.992 0.951 0.997 0.971
##
## sigma: 0.0249
##
## AIC AICc BIC
## -319 -318 -250
```

## Example: Australian eating-out expenditure

```
cafe %>% ETS(value) %>% forecast()
```

```
## # A tibble: 1 x 3  
##   data                model          forecast  
##   <list>              <list>        <list>  
## 1 <tsibble [426 x 2]> <ETS(M,A,M)> <tsibble [24 x 3]>
```

# Example: Australian eating-out expenditure

```
cafe %>% ETS(value) %>% forecast() %>%  
summary()
```

```
## # A tsibble: 24 x 4 [1MONTH]  
##       index mean      80%      95%  
##       <mt> <dbl>    <hilo>    <hilo>  
## 1 2017 Oct  3.83 [3.71, 3.96]80 [3.65, 4.02]95  
## 2 2017 Nov  3.81 [3.67, 3.96]80 [3.59, 4.03]95  
## 3 2017 Dec  4.22 [4.04, 4.40]80 [3.94, 4.49]95  
## 4 2018 Jan  3.83 [3.64, 4.01]80 [3.55, 4.10]95  
## 5 2018 Feb  3.51 [3.33, 3.70]80 [3.24, 3.79]95  
## 6 2018 Mar  3.87 [3.65, 4.09]80 [3.54, 4.20]95  
## 7 2018 Apr  3.78 [3.56, 4.01]80 [3.44, 4.13]95  
## 8 2018 May  3.81 [3.57, 4.05]80 [3.44, 4.18]95  
## 9 2018 Jun  3.68 [3.43, 3.92]80 [3.30, 4.05]95  
## 10 2018 Jul  3.88 [3.60, 4.15]80 [3.46, 4.29]95  
## # ... with 14 more rows
```

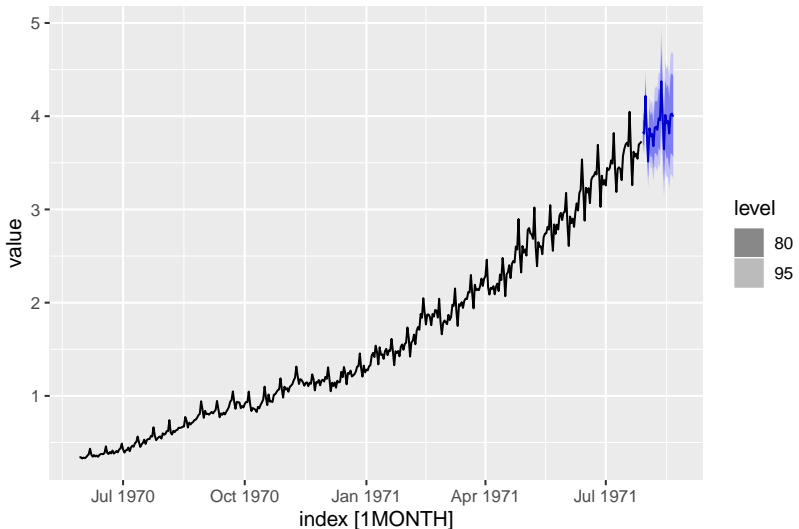
# Example: Australian eating-out expenditure

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

```
## # A tsibble: 24 x 3 [1MONTH]  
##       index mean          90%  
##       <mtm> <dbl>         <hilo>  
## 1 2017 Oct 3.83 [3.68, 3.99] 90  
## 2 2017 Nov 3.81 [3.63, 4.00] 90  
## 3 2017 Dec 4.22 [3.99, 4.45] 90  
## 4 2018 Jan 3.83 [3.59, 4.06] 90  
## 5 2018 Feb 3.51 [3.28, 3.75] 90  
## 6 2018 Mar 3.87 [3.59, 4.15] 90  
## 7 2018 Apr 3.78 [3.49, 4.07] 90  
## 8 2018 May 3.81 [3.50, 4.12] 90  
## 9 2018 Jun 3.68 [3.36, 3.99] 90  
## 10 2018 Jul 3.88 [3.53, 4.22] 90  
## # ... with 14 more rows
```

# Example: Australian eating-out expenditure

```
cafe %>% ETS(value) %>% forecast() %>% autoplot()
```



# Example: Australian eating-out expenditure

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

```
## # A tibble: 24 x 4 [1MONTH]  
##       index mean      80%      95%  
##       <mth> <dbl>    <hilo>    <hilo>  
## 1 2017 Oct  1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## 2 2017 Nov  1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## 3 2017 Dec  1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## 4 2018 Jan  1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## 5 2018 Feb  1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## 6 2018 Mar  1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## 7 2018 Apr  1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## 8 2018 May  1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## 9 2018 Jun  1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## 10 2018 Jul 1.31 [0.557, 3.09]80 [0.354, 4.86]95  
## # ... with 14 more rows
```

# Outline

- 1 Why change?
- 2 Example: Australian eating-out expenditure
- 3 Example: Australian prison population
- 4 Example: Half-hourly electricity demand
- 5 Equivalent methods
- 6 More information

# Example: Australian prison population

```
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]  
## # Keys:      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
```

# Example: Australian prison population

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

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

Aggregation and reconciliation not yet implemented.

# Outline

- 1 Why change?
- 2 Example: Australian eating-out expenditure
- 3 Example: Australian prison population
- 4 Example: Half-hourly electricity demand
- 5 Equivalent methods
- 6 More information

# Example: Half-hourly electricity demand

```
elecddemand
```

```
## # A tsibble: 17,520 x 4 [30MINUTE]
##   index          Demand Temperature WorkDay
##   <dtm>          <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

```
# fit1 <- TBATS(elecdemand, Demand)
fit2 <- ARIMA(elecdemand,
  Demand ~ Temperature + I(Temperature^2) + WorkDay)
summary(fit2)
```

```
## Series: Demand
## Regression with ARIMA(0,0,0) errors
##
## Coefficients:
##      intercept  Temperature  I(Temperature^2)  WorkDay
##           5.415        -0.195           0.006     0.733
## s.e.         0.038         0.004           0.000     0.011
##
## sigma^2 estimated as 0.484:  log likelihood=-18497
## AIC=37004   AICc=37004   BIC=37043
##
## Training set error measures:
##              ME  RMSE   MAE   MPE  MAPE  MASE   ACF1
## Training set 1.32e-14 0.695 0.562 -2.36 12.8  2.59 0.971
```

```
#forecast(fit2) %>% autoplot()
# How to specify future regressors?
```

# Outline

- 1 Why change?
- 2 Example: Australian eating-out expenditure
- 3 Example: Australian prison population
- 4 Example: Half-hourly electricity demand
- 5 Equivalent methods
- 6 More information

## Equivalent methods: forecast → fable

`auto.arima` → ARIMA

`ets` → ETS

`tslm/lm` → LM

`tbats` → TBATS

`nnetar` → NNETAR

`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 → 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

- 1 Why change?
- 2 Example: Australian eating-out expenditure
- 3 Example: Australian prison population
- 4 Example: Half-hourly electricity demand
- 5 Equivalent methods
- 6 More information

# More information



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



Di Cook



Earo Wang



Mitchell O'Hara-Wild

## Follow our progress

- [tidyverts.org](https://tidyverts.org)
- [robjhyndman.com/hyndsight](https://robjhyndman.com/hyndsight)