MONASH University

MONASH BUSINESS SCHOOL

Tidy forecasting in R

tsibble feasts fable

Rob J Hyndman

26 September 2019

Outline

Outline

Tidyverts packages

tidyverts.org

tsibble

tsibbledata

feasts

fable

Time series data

- Four-yearly Olympic winning times
- Annual Google profits
- Quarterly Australian snowy production
- Monthly rainfall
- Weekly retail sales
- Daily IBM stock prices
- Hourly electricity demand
- 5-minute freeway traffic counts
- Time-stamped stock transaction data

tsibble objects

```
global_economy
```

```
## # A tsibble: 15,150 x 6 [1Y]
## # Key:       Country [263]
##     Year Country             GDP Imports Exports Population
##    <dbl> <fct>             <dbl>   <dbl>   <dbl>      <dbl>
##  1  1960 Afghanistan  537777811.    7.02    4.13    8996351
##  2  1961 Afghanistan  548888896.    8.10    4.45    9166764
##  3  1962 Afghanistan  546666678.    9.35    4.88    9345868
##  4  1963 Afghanistan  751111191.   16.9     9.17    9533954
##  5  1964 Afghanistan  800000044.   18.1     8.89    9731361
##  6  1965 Afghanistan 1006666638.   21.4    11.3     9938414
##  7  1966 Afghanistan 1399999967.   18.6     8.57   10152331
##  8  1967 Afghanistan 1673333418.   14.2     6.77   10372630
##  9  1968 Afghanistan 1373333367.   15.2     8.90   10604346
## 10  1969 Afghanistan 1408888922.   15.0    10.1    10854428
## # ... with 15,140 more rows
```

tsibble objects

```
global_economy
```

```
## # A tsibble: 15,150 x 6 [1Y]
## # Key:       Country [263]
##     Year Country             GDP Imports Exports Population
##    Index <fct>             <dbl>   <dbl>   <dbl>      <dbl>
##  1  1960 Afghanistan  537777811.    7.02    4.13    8996351
##  2  1961 Afghanistan  548888896.    8.10    4.45    9166764
##  3  1962 Afghanistan  546666678.    9.35    4.88    9345868
##  4  1963 Afghanistan  751111191.   16.9     9.17    9533954
##  5  1964 Afghanistan  800000044.   18.1     8.89    9731361
##  6  1965 Afghanistan 1006666638.   21.4    11.3     9938414
##  7  1966 Afghanistan 1399999967.   18.6     8.57   10152331
##  8  1967 Afghanistan 1673333418.   14.2     6.77   10372630
##  9  1968 Afghanistan 1373333367.   15.2     8.90   10604346
## 10  1969 Afghanistan 1408888922.   15.0    10.1    10854428
## # ... with 15,140 more rows
```

tsibble objects

```
global_economy
```

```
## # A tsibble: 15,150 x 6 [1Y]
## # Key:       Country [263]
##     Year Country              GDP Imports Exports Population
##     Index Key                <dbl>   <dbl>   <dbl>      <dbl>
##  1  1960 Afghanistan  537777811.    7.02    4.13    8996351
##  2  1961 Afghanistan  548888896.    8.10    4.45    9166764
##  3  1962 Afghanistan  546666678.    9.35    4.88    9345868
##  4  1963 Afghanistan  751111191.   16.9     9.17    9533954
##  5  1964 Afghanistan  800000044.   18.1     8.89    9731361
##  6  1965 Afghanistan 1006666638.   21.4    11.3     9938414
##  7  1966 Afghanistan 1399999967.   18.6     8.57   10152331
##  8  1967 Afghanistan 1673333418.   14.2     6.77   10372630
##  9  1968 Afghanistan 1373333367.   15.2     8.90   10604346
## 10  1969 Afghanistan 1408888922.   15.0    10.1    10854428
## # ... with 15,140 more rows
```

tsibble objects

```r
global_economy
```

```
## # A tsibble: 15,150 x 6 [1Y]
## # Key:       Country [263]
##     Year Country           GDP Imports Exports Population
##    Index Key               Measured variables
##  1  1960 Afghanistan  537777811.    7.02    4.13    8996351
##  2  1961 Afghanistan  548888896.    8.10    4.45    9166764
##  3  1962 Afghanistan  546666678.    9.35    4.88    9345868
##  4  1963 Afghanistan  751111191.   16.9     9.17    9533954
##  5  1964 Afghanistan  800000044.   18.1     8.89    9731361
##  6  1965 Afghanistan 1006666638.   21.4    11.3     9938414
##  7  1966 Afghanistan 1399999967.   18.6     8.57   10152331
##  8  1967 Afghanistan 1673333418.   14.2     6.77   10372630
##  9  1968 Afghanistan 1373333367.   15.2     8.90   10604346
## 10  1969 Afghanistan 1408888922.   15.0    10.1    10854428
## # ... with 15,140 more rows
```

tsibble objects

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##      <qtr> <chr>    <chr> <chr>    <dbl>
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
## # ... with 24,310 more rows
```

tsibble objects

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##     Index  <chr>    <chr> <chr>    <dbl>
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
## # ... with 24,310 more rows
```

tsibble objects

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##    Index   Keys                    <dbl>
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
## # ... with 24,310 more rows
```

tsibble objects

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##    Index   Keys                    Measure
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
## # ... with 24,310 more rows
```

tsibble objects

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
## # ... with 24,310 more rows
```

Index

Keys

Measure

Domestic visitor nights in thousands by state/region and purpose.

`tsibble` objects

- A `tsibble` allows storage and manipulation of multiple time series in R.
- It contains:
  - An index: time information about the observation
  - Measured variable(s): numbers of interest
  - Key variable(s): optional unique identifiers for each series
- It works with tidyverse functions.

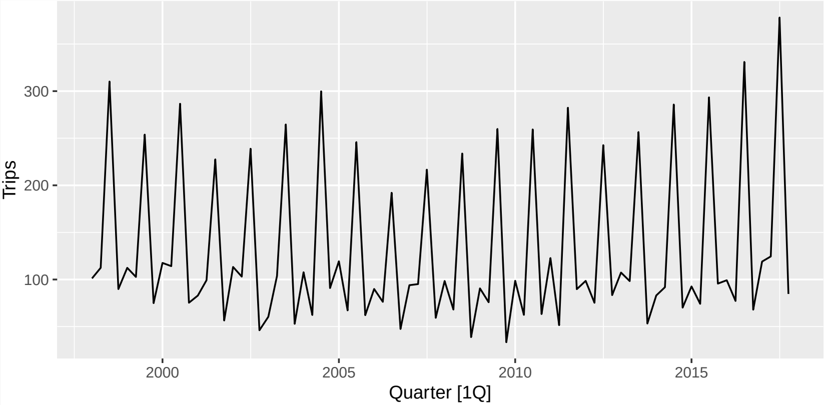
Extracting single time series

```
snowy <- tourism %>%
  filter(
    Region=="Snowy Mountains",
    Purpose=="Holiday"
  )
snowy
```

```
## # A tsibble: 80 x 5 [1Q]
## # Key:       Region, State, Purpose [1]
##    Quarter Region          State Purpose Trips
##      <qtr> <chr>           <chr> <chr>   <dbl>
##  1 1998 Q1 Snowy Mountains NSW   Holiday 101.
##  2 1998 Q2 Snowy Mountains NSW   Holiday 112.
##  3 1998 Q3 Snowy Mountains NSW   Holiday 310.
##  4 1998 Q4 Snowy Mountains NSW   Holiday  89.8
##  5 1999 Q1 Snowy Mountains NSW   Holiday 112.
```

Extracting single time series

```
snowy %>% autoplot(Trips)
```



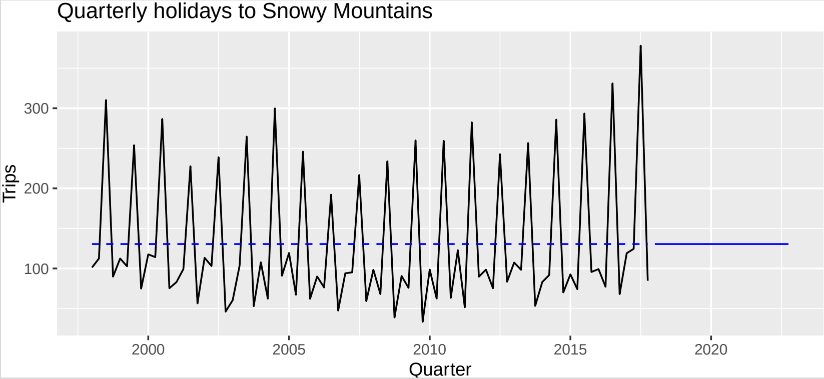
How would you forecast these series?

Outline

Benchmark forecasting methods

Mean method

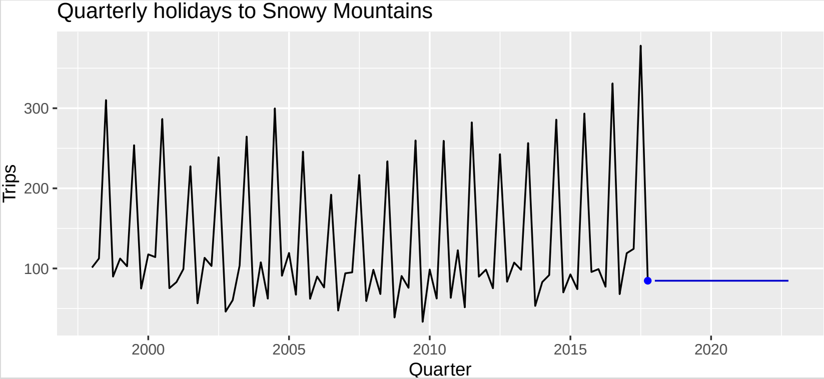
- Forecast of all future values is equal to mean of historical data $\{y_1, \ldots, y_T\}$.
- Forecasts: $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T$



Benchmark forecasting methods

Naïve method

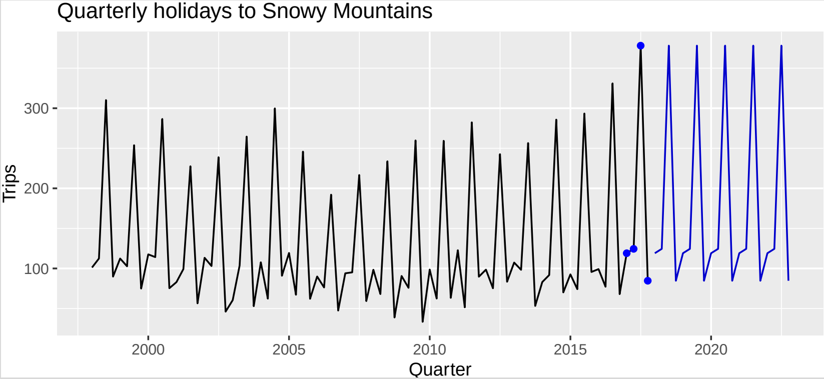
- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Consequence of efficient market hypothesis.



Benchmark forecasting methods

Seasonal naïve method

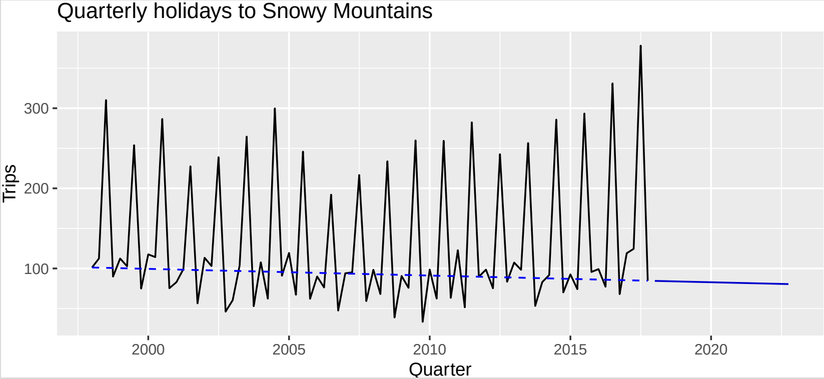
- Forecasts equal to last value from same season.
- Forecasts: $\hat{y}_{T+h|T} = y_{T+h-m(k+1)}$, where m = seasonal period and k is the integer part of $(h-1)/m$.



Benchmark forecasting methods

Drift method

- Forecasts equal to last value plus average change.
- Forecasts: $\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}(y_T - y_1)$.
- Equivalent to line between first and last observations.



Model estimation

The `model()` function trains models to data.

```
# Fit the models
fit <- snowy %>%
  model(
    Mean = MEAN(Trips),
    Naïve = NAIVE(Trips),
    SeasonalNaïve = SNAIVE(Trips),
    Drift = RW(Trips ~ drift())
  )
```

```
fit
```

```
## # A mable: 1 x 7
## # Key:     Region, State, Purpose [1]
##   Region    State Purpose Mean   Naïve SeasonalNaïve Drift
##   <chr>     <chr> <chr>   <mode> <mod> <model>       <model>
## 1 Snowy Mo~ NSW   Holiday <MEAN> <NAI~ <SNAIVE>      <RW w/~
```

A `mable` is a model table, each cell corresponds to a fitted model.

Producing forecasts

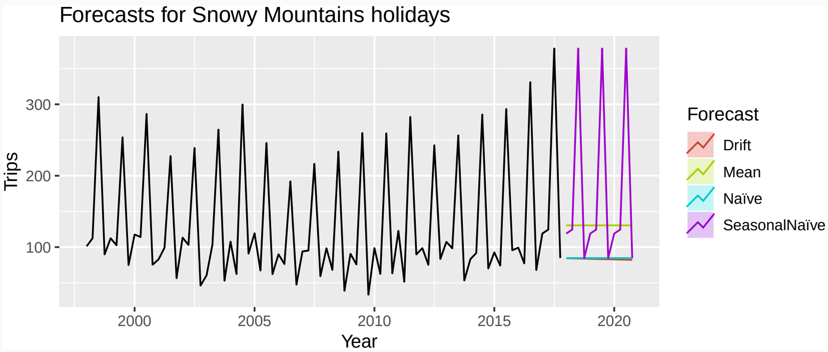
```r
fc <- fit %>%
  forecast(h = 12)
```

```
## # A fable: 48 x 7 [1Q]
## # Key:     Region, State, Purpose, .model [4]
##   Region State Purpose .model    Quarter Trips .distribution
##   <chr>  <chr> <chr>   <chr>       <qtr> <dbl> <dist>
## 1 Snowy~ NSW   Holiday Mean      2015 Q1  126. N(126, 6408)
## 2 Snowy~ NSW   Holiday Mean      2015 Q2  126. N(126, 6408)
## 3 Snowy~ NSW   Holiday Mean      2015 Q3  126. N(126, 6408)
## 4 Snowy~ NSW   Holiday Mean      2015 Q4  126. N(126, 6408)
## # ... with 44 more rows
```

A `fable` is a forecast table with point forecasts and distributions.

Visualising forecasts

```
fc %>%
  autoplot(snowy, level = NULL) +
  ggtitle("Forecasts for Snowy Mountains holidays") +
  xlab("Year") +
  guides(colour=guide_legend(title="Forecast"))
```



Forecasting many series

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##      <qtr> <chr>    <chr> <chr>    <dbl>
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
```

Forecasting many series

```r
tourism %>%
  model(
    mean = MEAN(Trips),
    snaive = SNAIVE(Trips)
  )
```

```
## # A mable: 304 x 5
## # Key:     Region, State, Purpose [304]
##    Region          State Purpose  mean    snaive
##    <chr>           <chr> <chr>    <model> <model>
##  1 Adelaide        SA    Business <MEAN>  <SNAIVE>
##  2 Adelaide        SA    Holiday  <MEAN>  <SNAIVE>
##  3 Adelaide        SA    Other    <MEAN>  <SNAIVE>
##  4 Adelaide        SA    Visiting <MEAN>  <SNAIVE>
##  5 Adelaide Hills  SA    Business <MEAN>  <SNAIVE>
##  6 Adelaide Hills  SA    Holiday  <MEAN>  <SNAIVE>
```

Forecasting many series

```r
tourism %>%
  model(
    mean = MEAN(Trips),
    snaive = SNAIVE(Trips)
  ) %>%
  forecast(h= "3 years")
```

```
## # A fable: 7,296 x 7 [1Q]
## # Key:     Region, State, Purpose, .model [608]
##    Region State Purpose .model    Quarter Trips
##    <chr>  <chr> <chr>   <chr>       <qtr> <dbl>
##  1 Adela~ SA    Busine~ mean      2018 Q1  156.
##  2 Adela~ SA    Busine~ mean      2018 Q2  156.
##  3 Adela~ SA    Busine~ mean      2018 Q3  156.
##  4 Adela~ SA    Busine~ mean      2018 Q4  156.
##  5 Adela~ SA    Busine~ mean      2019 Q1  156.
```

Outline

Historical perspective

- Developed in the 1950s and 1960s as methods (algorithms) to produce point forecasts.
- Combine “level”, “trend” (slope) and “seasonal” states to describe a time series.
- The rate of change of the components are controlled by “smoothing parameters”.
- Need to choose best values for the smoothing parameters and initial states.
- Equivalent ETS state space models developed in the 1990s and 2000s.

ETS state space models

General notation E T S : **E**xponen**T**ial **S**moothing

Error **T**rend **S**eason

Error: Additive (`"A"`) or multiplicative (`"M"`)

ETS state space models

General notation E T S : **E**xponen**T**ial **S**moothing

$\nearrow$ $\uparrow$ $\nwarrow$

Error **T**rend **S**eason

Error: Additive (`"A"`) or multiplicative (`"M"`)

Trend: None (`"N"`), additive (`"A"`), multiplicative (`"M"`), or damped (`"Ad"` or `"Md"`).

ETS state space models

General notation E T S : **E**xponen**T**ial **S**moothing

$\nearrow \uparrow \nwarrow$

Error **T**rend **S**eason

Error: Additive (`"A"`) or multiplicative (`"M"`)

Trend: None (`"N"`), additive (`"A"`), multiplicative (`"M"`), or damped (`"Ad"` or `"Md"`).

Seasonality: None (`"N"`), additive (`"A"`) or multiplicative (`"M"`)

ETS state space models

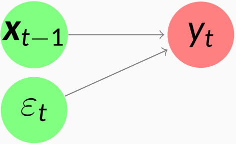
Additive Error

Trend Component		Seasonal Component: N (None)	A (Additive)	M (Multiplicative)
N	(None)	A,N,N	A,N,A	~~A,N,M~~
A	(Additive)	A,A,N	A,A,A	~~A,A,M~~
A_d	(Additive damped)	A,A_d,N	A,A_d,A	~~A,A_d,M~~

Multiplicative Error

Trend Component		Seasonal Component: N (None)	A (Additive)	M (Multiplicative)
N	(None)	M,N,N	M,N,A	M,N,M
A	(Additive)	M,A,N	M,A,A	M,A,M
A_d	(Additive damped)	M,A_d,N	M,A_d,A	M,A_d,M

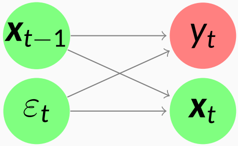
ETS state space models



State space model

$\boldsymbol{x}_t$ = (level, slope, seasonal)

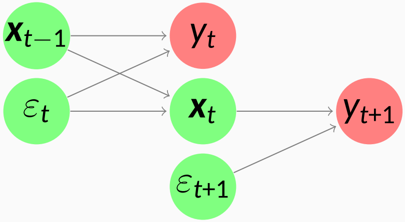
ETS state space models



State space model

$\boldsymbol{x}_t$ = (level, slope, seasonal)

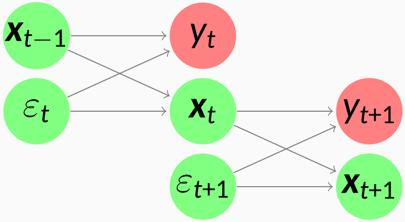
ETS state space models



State space model

$\boldsymbol{x}_t$ = (level, slope, seasonal)

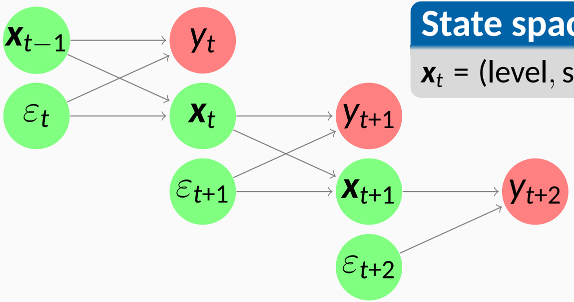
ETS state space models



State space model

$\boldsymbol{x}_t$ = (level, slope, seasonal)

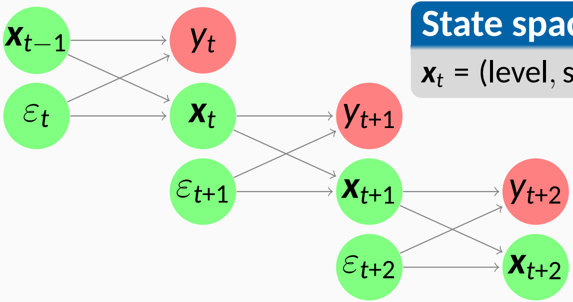
ETS state space models



State space model

$\mathbf{x}_t$ = (level, slope, seasonal)

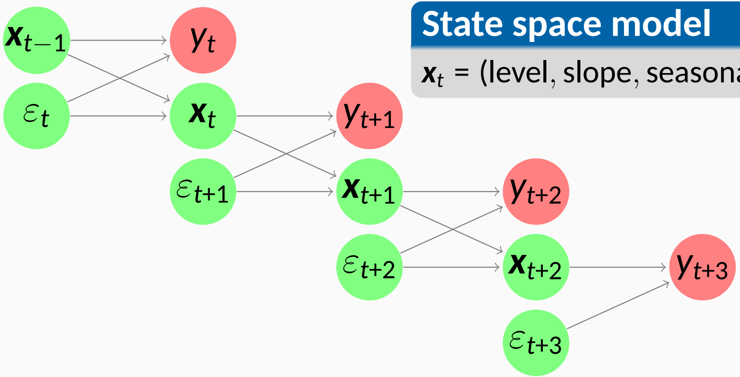
ETS state space models



State space model

$\boldsymbol{x}_t$ = (level, slope, seasonal)

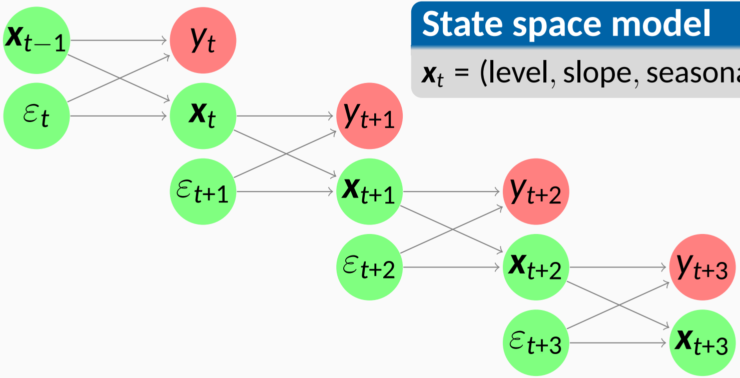
ETS state space models



State space model

$\mathbf{x}_t$ = (level, slope, seasonal)

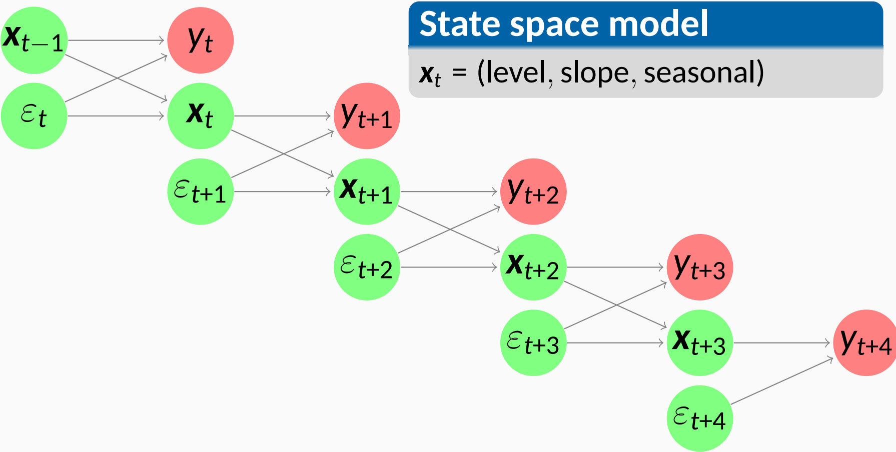
ETS state space models



State space model

$\boldsymbol{x}_t$ = (level, slope, seasonal)

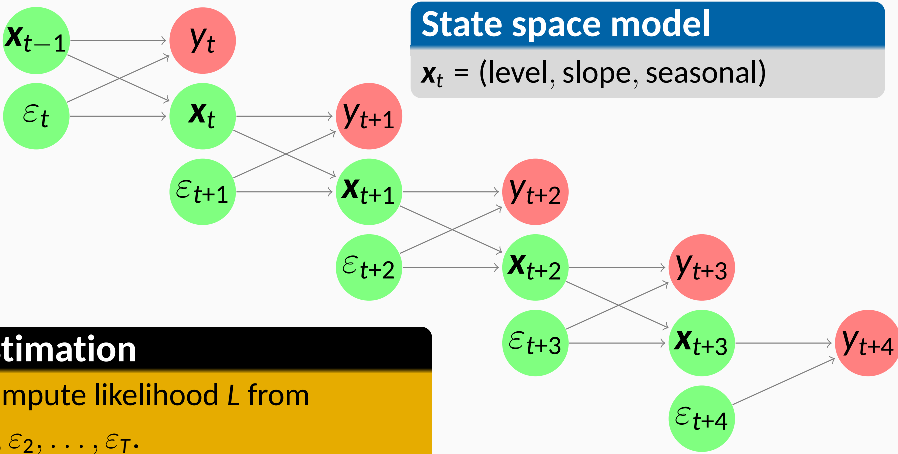
ETS state space models



State space model

$\boldsymbol{x}_t$ = (level, slope, seasonal)

ETS state space models



State space model

$\boldsymbol{x}_t$ = (level, slope, seasonal)

Estimation

Compute likelihood L from $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T$.

Optimize L wrt model parameters.

Automatic forecasting

From Hyndman et al. (IJF, 2002):

1. Apply each model that is appropriate to the data. Optimize parameters and initial values using MLE.
2. Select best method using AICc.
3. Produce forecasts using best method.
4. Obtain forecast intervals using underlying state space model.

- Method performed very well in M3 competition.
- Used as a benchmark in the M4 competition.

Example: Australian tourism

```r
fit <- tourism %>% model(ets = ETS(Trips))
fit
```

```
## # A mable: 304 x 4
## # Key:     Region, State, Purpose [304]
##    Region         State Purpose  ets
##    <chr>          <chr> <chr>    <model>
##  1 Adelaide       SA    Business <ETS(M,N,M)>
##  2 Adelaide       SA    Holiday  <ETS(A,N,A)>
##  3 Adelaide       SA    Other    <ETS(M,A,N)>
##  4 Adelaide       SA    Visiting <ETS(A,N,A)>
##  5 Adelaide Hills SA    Business <ETS(A,N,N)>
##  6 Adelaide Hills SA    Holiday  <ETS(A,A,N)>
##  7 Adelaide Hills SA    Other    <ETS(A,N,N)>
##  8 Adelaide Hills SA    Visiting <ETS(M,A,M)>
##  9 Alice Springs  NT    Business <ETS(M,N,M)>
## 10 Alice Springs  NT    Holiday  <ETS(M,N,A)>
## # ... with 294 more rows
```

Example: Australian tourism

```r
fit %>% filter(Region=="Snowy Mountains", Purpose=="Holiday") %>%
  report()
```

```
## Series: Trips
## Model: ETS(M,N,A)
##   Smoothing parameters:
##     alpha = 0.157
##     gamma = 1e-04
##
##   Initial states:
##    l   s1  s2    s3    s4
##   142 -61 131 -42.2 -27.7
##
##   sigma^2:  0.0388
##
##   AIC  AICc  BIC
##   852  854   869
```

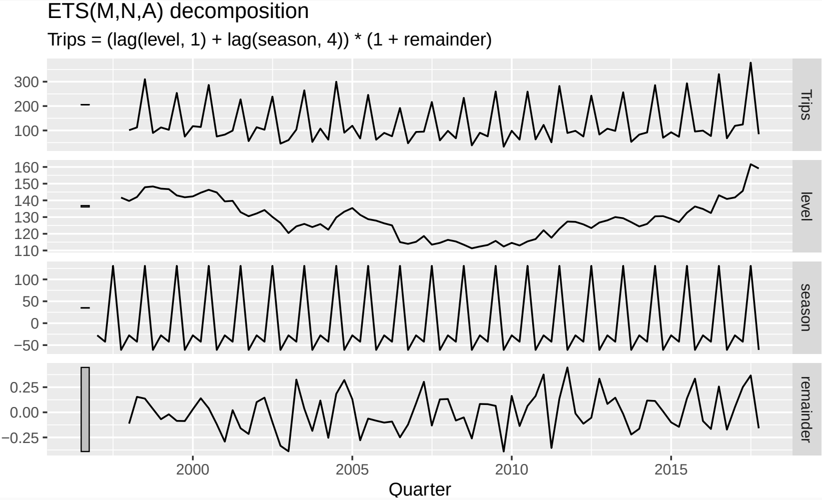
Example: Australian tourism

```r
fit %>% filter(Region=="Snowy Mountains", Purpose=="Holiday") %>%
  components(fit)
```

```
## # A dable:                  84 x 9 [1Q]
## # Key:                      Region, State, Purpose, .model
## #   [1]
## # ETS(M,N,A) Decomposition: Trips = (lag(level, 1) +
## #   lag(season, 4)) * (1 + remainder)
##    Region State Purpose .model    Quarter Trips level season
##    <chr>  <chr> <chr>   <chr>       <qtr> <dbl> <dbl>  <dbl>
##  1 Snowy~ NSW   Holiday ets       1997 Q1  NA      NA  -27.7
##  2 Snowy~ NSW   Holiday ets       1997 Q2  NA      NA  -42.2
##  3 Snowy~ NSW   Holiday ets       1997 Q3  NA      NA  131.
##  4 Snowy~ NSW   Holiday ets       1997 Q4  NA     142. -61.0
##  5 Snowy~ NSW   Holiday ets       1998 Q1 101.    140. -27.7
##  6 Snowy~ NSW   Holiday ets       1998 Q2 112.    142. -42.2
##  7 Snowy~ NSW   Holiday ets       1998 Q3 310.    148. 131.
##  8 Snowy~ NSW   Holiday ets       1998 Q4  89.8   148. -61.0
##  9 Snowy~ NSW   Holiday ets       1999 Q1 112.    147. -27.7
```

Example: Australian tourism

```
fit %>% filter(Region=="Snowy Mountains", Purpose=="Holiday") %>%
  components(fit) %>% autoplot()
```



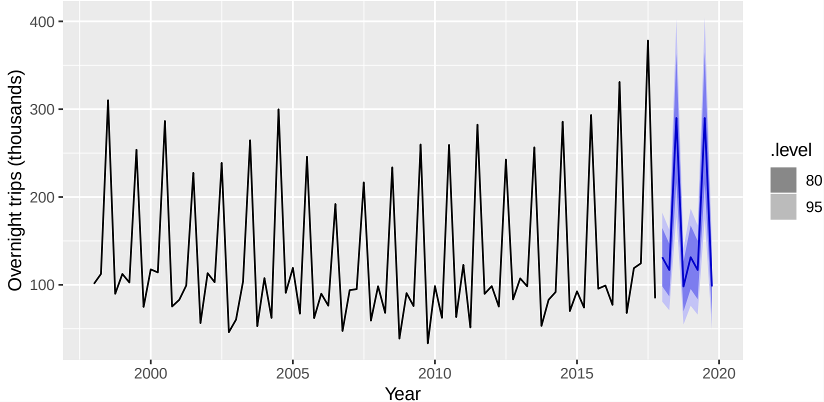
Example: Australian tourism

```r
fit %>% forecast()
```

```
## # A fable: 2,432 x 7 [1Q]
## # Key:     Region, State, Purpose, .model [304]
##    Region State Purpose .model    Quarter Trips
##    <chr>  <chr> <chr>   <chr>       <qtr> <dbl>
##  1 Adela~ SA    Busine~ ets       2018 Q1  149.
##  2 Adela~ SA    Busine~ ets       2018 Q2  173.
##  3 Adela~ SA    Busine~ ets       2018 Q3  184.
##  4 Adela~ SA    Busine~ ets       2018 Q4  171.
##  5 Adela~ SA    Busine~ ets       2019 Q1  149.
##  6 Adela~ SA    Busine~ ets       2019 Q2  173.
##  7 Adela~ SA    Busine~ ets       2019 Q3  184.
##  8 Adela~ SA    Busine~ ets       2019 Q4  171.
##  9 Adela~ SA    Holiday ets       2018 Q1  210.
## 10 Adela~ SA    Holiday ets       2018 Q2  173.
## # ... with 2,422 more rows, and 1 more variable:
## #   .distribution <dist>
```

Example: Australian tourism

```
fit %>% forecast() %>%
  filter(Region=="Snowy Mountains", Purpose=="Holiday") %>%
  autoplot(tourism) +
    xlab("Year") + ylab("Overnight trips (thousands)")
```



Outline

ARIMA models

AR:	autoregressive (lagged observations as inputs)
I:	integrated (differencing to make series stationary)
MA:	moving average (lagged errors as inputs)

ARIMA models

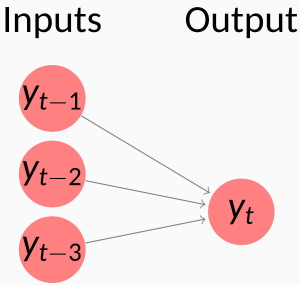
AR: autoregressive (lagged observations as inputs)

I: integrated (differencing to make series stationary)

MA: moving average (lagged errors as inputs)

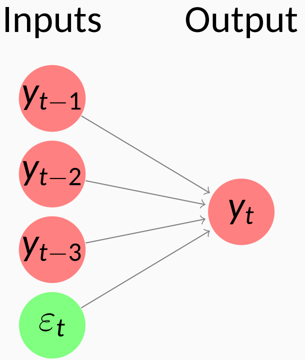
An ARIMA model is rarely interpretable in terms of visible data structures like trend and seasonality. But it can capture a huge range of time series patterns.

ARIMA models



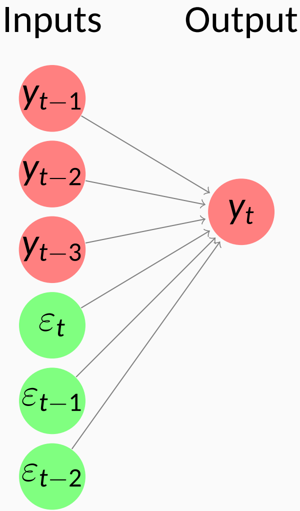
ARIMA models

Autoregression (AR) model

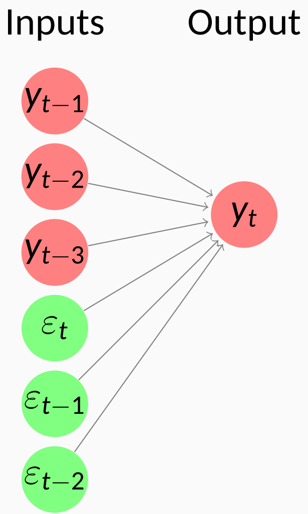


ARIMA models

Autoregression moving average (ARMA) model



ARIMA models



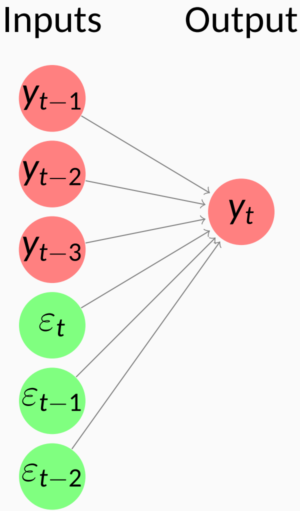
Autoregression moving average (ARMA) model

Estimation

Compute likelihood L from $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$.

Use optimization algorithm to maximize L.

ARIMA models



Autoregression moving average (ARMA) model

ARIMA model

Autoregression moving average (ARMA) model applied to differences.

Estimation

Compute likelihood L from $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T$.
Use optimization algorithm to maximize L.

Seasonal ARIMA models

$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\substack{\uparrow \\ \text{Non-seasonal part} \\ \text{of the model}}} \quad \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \text{Seasonal part of} \\ \text{of the model}}}$$

- m = number of observations per year.
- d first differences, D seasonal differences
- p AR lags, q MA lags
- P seasonal AR lags, Q seasonal MA lags

Seasonal ARIMA models

$$\text{ARIMA} \quad \underbrace{(p, d, q)}_{\substack{\uparrow \\ \text{Non-seasonal part} \\ \text{of the model}}} \quad \underbrace{(P, D, Q)_m}_{\substack{\uparrow \\ \text{Seasonal part of} \\ \text{of the model}}}$$

- m = number of observations per year.
- d first differences, D seasonal differences
- p AR lags, q MA lags
- P seasonal AR lags, Q seasonal MA lags

Hyndman and Khandakar (JSS, 2008) algorithm:

- Select no. differences d and D via tests.
- Select model orders p, q, P, Q by minimising AICc.
- Use stepwise search to traverse model space.

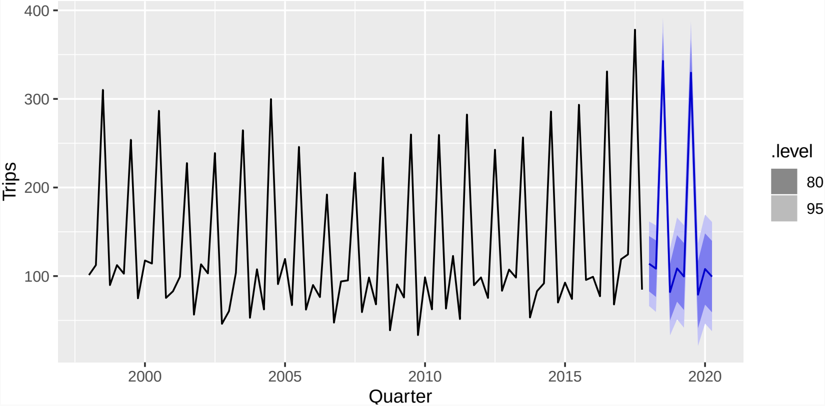
Example: Australian tourism

```
fit <- tourism %>%
  model(arima = ARIMA(Trips))
fit
```

```
## # A mable: 304 x 4
## # Key:     Region, State, Purpose [304]
##    Region         State Purpose  arima
##    <chr>          <chr> <chr>    <model>
##  1 Adelaide       SA    Business <ARIMA(0,0,0)(1,0,1)[4] w/ ~
##  2 Adelaide       SA    Holiday  <ARIMA(0,0,0)(1,0,1)[4] w/ ~
##  3 Adelaide       SA    Other    <ARIMA(0,1,1) w/ drift>
##  4 Adelaide       SA    Visiting <ARIMA(0,0,0)(1,0,1)[4] w/ ~
##  5 Adelaide Hil~ SA    Business <ARIMA(0,0,0) w/ mean>
##  6 Adelaide Hil~ SA    Holiday  <ARIMA(0,1,1)>
##  7 Adelaide Hil~ SA    Other    <ARIMA(0,1,2)(0,0,2)[4]>
##  8 Adelaide Hil~ SA    Visiting <ARIMA(0,1,1)>
##  9 Alice Springs NT    Business <ARIMA(0,1,1)(0,0,1)[4]>
## 10 Alice Springs NT    Holiday  <ARIMA(0,0,0)(0,1,2)[4]>
```

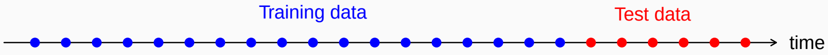
Example: Australian tourism

```
fit %>% forecast(h=10) %>%
  filter(Region=="Snowy Mountains", Purpose=="Holiday") %>%
  autoplot(tourism)
```



Outline

Training and test sets



- A model which fits the training data well will not necessarily forecast well.
- Forecast accuracy is based only on the test set.

Forecast errors

Forecast "error": the difference between an observed value and its forecast.

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

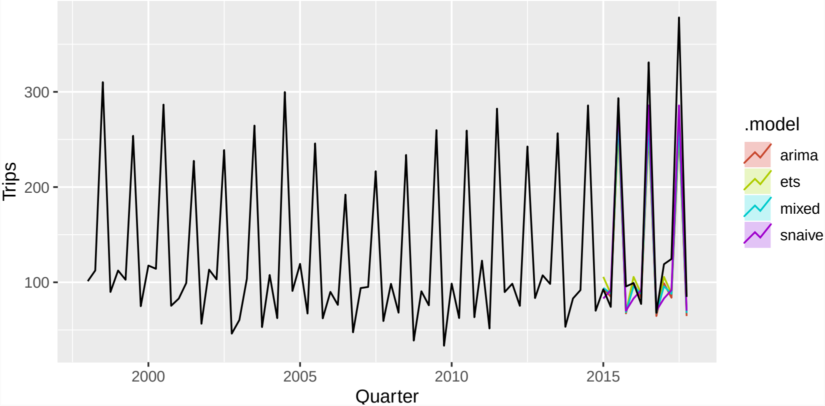
where the training data is given by $\{y_1, \dots, y_T\}$

Forecast errors

```
train <- tourism %>%
  filter(year(Quarter) <= 2014)
fit <- train %>%
  model(
    ets = ETS(Trips),
    arima = ARIMA(Trips),
    snaive = SNAIVE(Trips)
  ) %>%
  mutate(mixed = (ets+arima+snaive)/3)
fc <- fit %>% forecast(h="3 years")
```

Forecast errors

```
fc %>%
  filter(Region=="Snowy Mountains", Purpose=="Holiday") %>%
  autoplot(level=NULL) +
  autolayer(snowy, Trips)
```



Measures of forecast accuracy

$$
\begin{aligned}
y_{T+h} &= \text{$(T+h)$th observation, } h = 1, \dots, H \\
\hat{y}_{T+h|T} &= \text{its forecast based on data up to time } T. \\
e_{T+h} &= y_{T+h} - \hat{y}_{T+h|T}
\end{aligned}
$$

$$
\begin{aligned}
\text{MAE} &= \text{mean}(|e_{T+h}|) \\
\text{MSE} &= \text{mean}(e_{T+h}^2) \qquad\qquad \text{RMSE} = \sqrt{\text{mean}(e_{T+h}^2)} \\
\text{MAPE} &= 100\text{mean}(|e_{T+h}|/|y_{T+h}|)
\end{aligned}
$$

Measures of forecast accuracy

$$
\begin{aligned}
y_{T+h} &= (T+h)\text{th observation}, h = 1, \dots, H \\
\hat{y}_{T+h|T} &= \text{its forecast based on data up to time } T. \\
e_{T+h} &= y_{T+h} - \hat{y}_{T+h|T}
\end{aligned}
$$

$$
\begin{aligned}
\text{MAE} &= \text{mean}(|e_{T+h}|) \\
\text{MSE} &= \text{mean}(e_{T+h}^2) \qquad \text{RMSE} = \sqrt{\text{mean}(e_{T+h}^2)} \\
\text{MAPE} &= 100\text{mean}(|e_{T+h}|/|y_{T+h}|)
\end{aligned}
$$

- MAE, MSE, RMSE are all scale dependent.
- MAPE is scale independent but is only sensible if $y_t \gg 0$ for all t, and y has a natural zero.

Measures of forecast accuracy

Mean Absolute Scaled Error

$$\text{MASE} = \text{mean}(|e_{T+h}|/Q)$$

where Q is a stable measure of the scale of the time series $\{y_t\}$.

Proposed by Hyndman and Koehler (IJF, 2006).

For non-seasonal time series,

$$Q = (T-1)^{-1}\sum_{t=2}^{T}|y_t - y_{t-1}|$$

works well. Then MASE is equivalent to MAE relative to a naïve method.

Measures of forecast accuracy

Mean Absolute Scaled Error

$$\text{MASE} = \text{mean}(|e_{T+h}|/Q)$$

where Q is a stable measure of the scale of the time series $\{y_t\}$.

Proposed by Hyndman and Koehler (IJF, 2006).

For seasonal time series,

$$Q = (T-m)^{-1}\sum_{t=m+1}^{T}|y_t - y_{t-m}|$$

works well. Then MASE is equivalent to MAE relative to a seasonal naïve method.

Measures of forecast accuracy

```r
accuracy(fc, tourism)
```

```
## # A tibble: 1,216 x 12
##    .model Region State Purpose .type    ME  RMSE   MAE
##    <chr>  <chr>  <chr> <chr>   <chr> <dbl> <dbl> <dbl>
##  1 arima  Adela~ SA    Busine~ Test  22.5  28.5  25.3 
##  2 arima  Adela~ SA    Holiday Test  21.9  34.8  28.0 
##  3 arima  Adela~ SA    Other   Test   4.71 17.5  14.6 
##  4 arima  Adela~ SA    Visiti~ Test  32.8  37.1  32.8 
##  5 arima  Adela~ SA    Busine~ Test   1.31  5.58  3.57
##  6 arima  Adela~ SA    Holiday Test   6.46  7.43  6.46
##  7 arima  Adela~ SA    Other   Test   1.35  2.79  1.93
##  8 arima  Adela~ SA    Visiti~ Test   8.37 12.6  10.4 
##  9 arima  Alice~ NT    Busine~ Test   9.85 12.2  10.7 
## 10 arima  Alice~ NT    Holiday Test   4.80 11.3   9.30
## # ... with 1,206 more rows, and 4 more variables:
## #   MPE <dbl>, MAPE <dbl>, MASE <dbl>, ACF1 <dbl>
```

Measures of forecast accuracy

```r
accuracy(fc, tourism) %>%
  group_by(.model) %>%
  summarise(
    RMSE = mean(RMSE),
    MAE = mean(MAE),
    MASE = mean(MASE)
  ) %>%
  arrange(RMSE)
```

```
## # A tibble: 4 x 4
##   .model  RMSE   MAE  MASE
##   <chr>  <dbl> <dbl> <dbl>
## 1 mixed   19.8  16.0 0.997
## 2 ets     20.2  16.4 1.00
## 3 snaive  21.5  17.3 1.17
## 4 arima   21.9  17.8 1.07
```

Acknowledgements

Mitchell O’Hara-Wild

Earo Wang

feasts.tidyverts.org

robjhyndman.com