Feasts & fables

Modern tools for time series analysis

Rob J Hyndman

robjhyndman.com/cornish2021

MONASH University

Outline

E A Cornish (1909–1973)

- Foundation Fellow of the Australian Academy of Science (1954)
- Chief of the CSIRO Mathematical Statistics Division (1954–1973)
- Helped establish CSIRO Division of Computing Research (1963)

E A Cornish (1909–1973)

Rainfall papers

1. Cornish, EA & Coote, GG (1958) *The correlation of monthly rainfall with position and altitude of observing stations in South Australia.* CSIRO Div Math Stats Tech Paper 4.
2. Cornish, EA and Stenhouse, NS (1958) *Inter-station correlations of monthly rainfall in South Australia.* CSIRO Div Math Stats Tech Paper 5.
3. Cornish, EA, Hill, GW, & Evans, MJ (1961) *Inter-station correlations of rainfall in southern Australia.* CSIRO Div Math Stats Tech Paper 10.

- Modelled monthly rainfall at 97 South Australian weather stations based on altitude, longitude and latitude.
- Pairwise correlations of 6-day rainfall totals between weather stations: 90,585 correlation coefficients.

Tidyverts packages

tidyverts.org

tsibble

tsibbledata

feasts

fable

Outline

1. What does modern time series data look like?
2. Feature-based time series analysis
3. Probabilistic forecasting for large time series
4. Evaluating probabilistic forecasts
5. Forecast reconciliation

Annual economic data

```
## # 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
```

Annual economic data

```
## # 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
```

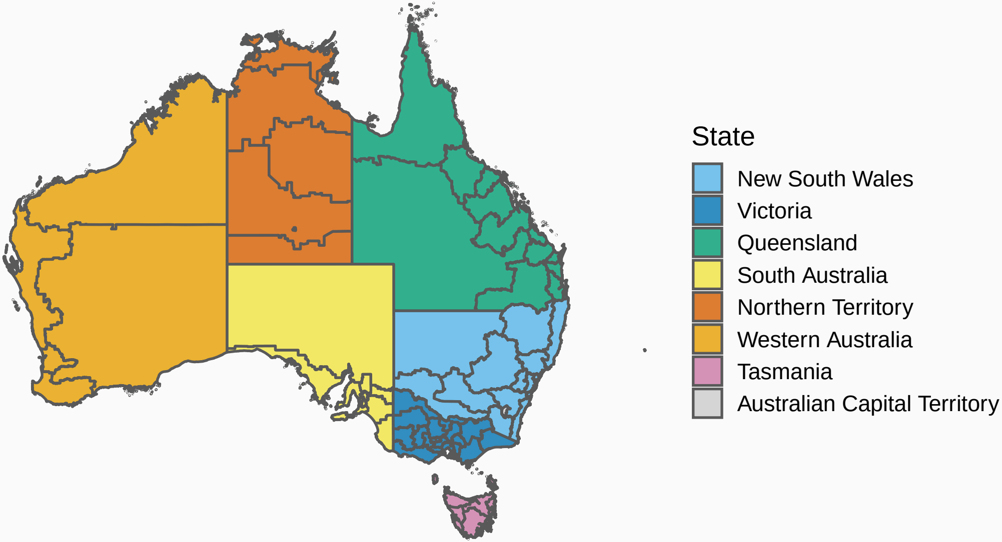
Annual economic data

```
## # 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
```

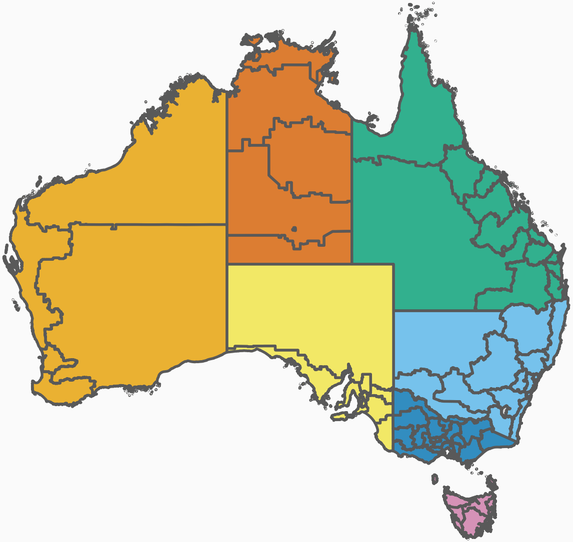
Annual economic data

```
## # 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
```

Australian tourism regions



Australian tourism regions



- Quarterly data on visitor nights: 1998 – 2017
- From *National Visitor Survey*, interviews of 120,000 Australians aged 15+.
- Geographical hierarchy split by
  - 8 states and territories
  - 76 regions
- Purpose:
  - Holidays
  - Business
  - Visiting friends & relatives
  - Other

Quarterly tourism data

```
## # 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
```

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

Quarterly tourism data

```
## # 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
```

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

Quarterly tourism data

```
## # 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
```

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

Quarterly tourism data

```
## # 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
```

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

Australian electricity demand

```
## # A tsibble: 420,864 x 6 [30m] <Australia/Melbourne>
## # Key:       State [8]
##    Time                State Date       Holiday Temperature Demand
##    <dttm>              <fct> <date>     <lgl>         <dbl>  <dbl>
##  1 2012-01-01 00:00:00 VIC   2012-01-01 TRUE           21.4  4383.
##  2 2012-01-01 00:30:00 VIC   2012-01-01 TRUE           21.0  4263.
##  3 2012-01-01 01:00:00 VIC   2012-01-01 TRUE           20.7  4049.
##  4 2012-01-01 01:30:00 VIC   2012-01-01 TRUE           20.6  3878.
##  5 2012-01-01 02:00:00 VIC   2012-01-01 TRUE           20.4  4036.
##  6 2012-01-01 02:30:00 VIC   2012-01-01 TRUE           20.2  3866.
##  7 2012-01-01 03:00:00 VIC   2012-01-01 TRUE           20.1  3694.
##  8 2012-01-01 03:30:00 VIC   2012-01-01 TRUE           19.6  3562.
##  9 2012-01-01 04:00:00 VIC   2012-01-01 TRUE           19.1  3433.
## 10 2012-01-01 04:30:00 VIC   2012-01-01 TRUE           19.0  3359.
## # ... with 420,854 more rows
```

Australian electricity demand

```
## # A tsibble: 420,864 x 6 [30m] <Australia/Melbourne>
## # Key:       State [8]
##    Time                State Date       Holiday Temperature Demand
##                        <fct> <date>     <lgl>         <dbl>  <dbl>
##  1 2012-01-01 00:00:00 VIC   2012-01-01 TRUE           21.4  4383.
##  2 2012-01-01 00:30:00 VIC   2012-01-01 TRUE           21.0  4263.
##  3 2012-01-01 01:00:00 VIC   2012-01-01 TRUE           20.7  4049.
##  4 2012-01-01 01:30:00 VIC   2012-01-01 TRUE           20.6  3878.
##  5 2012-01-01 02:00:00 VIC   2012-01-01 TRUE           20.4  4036.
##  6 2012-01-01 02:30:00 VIC   2012-01-01 TRUE           20.2  3866.
##  7 2012-01-01 03:00:00 VIC   2012-01-01 TRUE           20.1  3694.
##  8 2012-01-01 03:30:00 VIC   2012-01-01 TRUE           19.6  3562.
##  9 2012-01-01 04:00:00 VIC   2012-01-01 TRUE           19.1  3433.
## 10 2012-01-01 04:30:00 VIC   2012-01-01 TRUE           19.0  3359.
## # ... with 420,854 more rows
```

Index

Australian electricity demand

```
## # A tsibble: 420,864 x 6 [30m] <Australia/Melbourne>
## # Key:       State [8]
##    Time                State Date       Holiday Temperature Demand
##    Index               Key   <date>     <lgl>         <dbl>  <dbl>
##  1 2012-01-01 00:00:00 VIC   2012-01-01 TRUE           21.4  4383.
##  2 2012-01-01 00:30:00 VIC   2012-01-01 TRUE           21.0  4263.
##  3 2012-01-01 01:00:00 VIC   2012-01-01 TRUE           20.7  4049.
##  4 2012-01-01 01:30:00 VIC   2012-01-01 TRUE           20.6  3878.
##  5 2012-01-01 02:00:00 VIC   2012-01-01 TRUE           20.4  4036.
##  6 2012-01-01 02:30:00 VIC   2012-01-01 TRUE           20.2  3866.
##  7 2012-01-01 03:00:00 VIC   2012-01-01 TRUE           20.1  3694.
##  8 2012-01-01 03:30:00 VIC   2012-01-01 TRUE           19.6  3562.
##  9 2012-01-01 04:00:00 VIC   2012-01-01 TRUE           19.1  3433.
## 10 2012-01-01 04:30:00 VIC   2012-01-01 TRUE           19.0  3359.
## # ... with 420,854 more rows
```

Australian electricity demand

```
## # A tsibble: 420,864 x 6 [30m] <Australia/Melbourne>
## # Key:       State [8]
##    Time                State Date       Holiday Temperature Demand
##    Index               Key   Measures
##  1 2012-01-01 00:00:00 VIC   2012-01-01 TRUE           21.4  4383.
##  2 2012-01-01 00:30:00 VIC   2012-01-01 TRUE           21.0  4263.
##  3 2012-01-01 01:00:00 VIC   2012-01-01 TRUE           20.7  4049.
##  4 2012-01-01 01:30:00 VIC   2012-01-01 TRUE           20.6  3878.
##  5 2012-01-01 02:00:00 VIC   2012-01-01 TRUE           20.4  4036.
##  6 2012-01-01 02:30:00 VIC   2012-01-01 TRUE           20.2  3866.
##  7 2012-01-01 03:00:00 VIC   2012-01-01 TRUE           20.1  3694.
##  8 2012-01-01 03:30:00 VIC   2012-01-01 TRUE           19.6  3562.
##  9 2012-01-01 04:00:00 VIC   2012-01-01 TRUE           19.1  3433.
## 10 2012-01-01 04:30:00 VIC   2012-01-01 TRUE           19.0  3359.
## # ... with 420,854 more rows
```

Characteristics of modern time series

- Often observed at sub-daily frequency over a long time.
- Multiple keys which may be nested.
- Multiple seasonal patterns.
- Multiple measures for each combination of index and keys.

Characteristics of modern time series

- Often observed at sub-daily frequency over a long time.
- Multiple keys which may be nested.
- Multiple seasonal patterns.
- Multiple measures for each combination of index and keys.

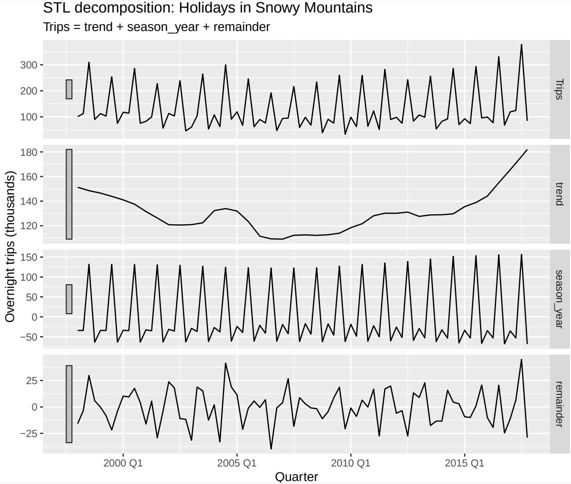
`tsibble` objects

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

Outline

1. What does modern time series data look like?
2. Feature-based time series analysis
3. Probabilistic forecasting for large time series
4. Evaluating probabilistic forecasts
5. Forecast reconciliation

STL decomposition



y_t

T_t

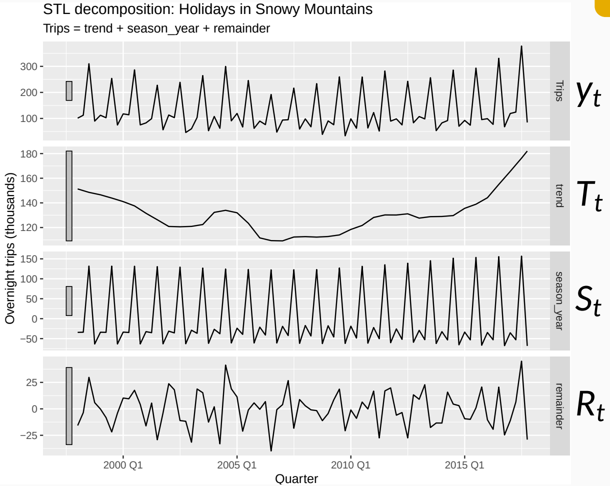
S_t

R_t

STL decomposition

$$y_t = T_t + S_t + R_t$$

STL decomposition



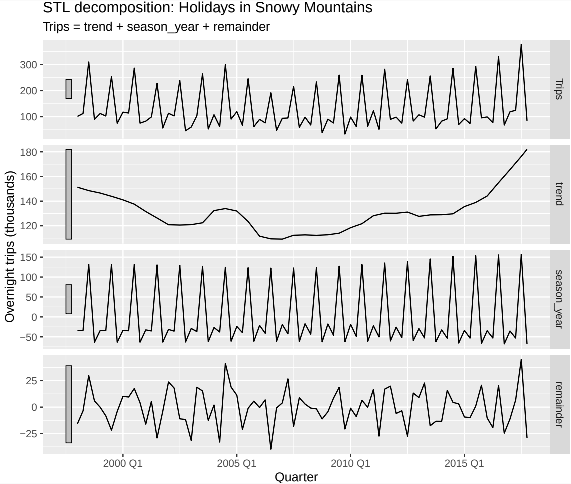
STL decomposition

$$y_t = T_t + S_t + R_t$$

Trend strength

$$\max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(T_t + R_t)}\right)$$

STL decomposition



y_t

T_t

S_t

R_t

STL decomposition

$$y_t = T_t + S_t + R_t$$

Trend strength

$$\max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(T_t + R_t)}\right)$$

Seasonal strength

$$\max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(S_t + R_t)}\right)$$

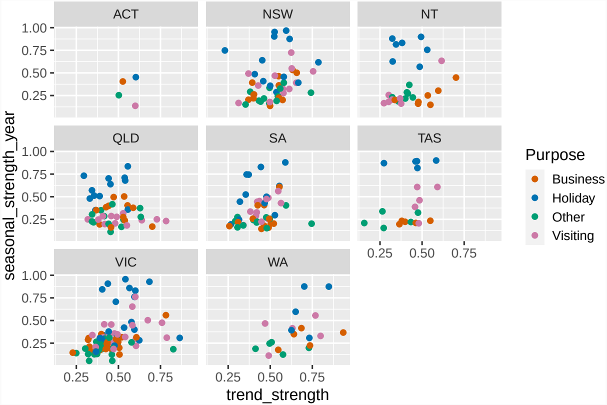
STL-based features

```
tourism %>%
  features(Trips, feat_stl)
```

```
## # A tibble: 304 x 12
##    Region  State Purpose trend_strength seasonal_streng~ seasonal_peak_y~ seasonal_trough~
##    <chr>   <chr> <chr>            <dbl>            <dbl>            <dbl>            <dbl>
##  1 Adelai~ SA    Busine~          0.464            0.407                3                1
##  2 Adelai~ SA    Holiday          0.554            0.619                1                2
##  3 Adelai~ SA    Other            0.746            0.202                2                1
##  4 Adelai~ SA    Visiti~          0.435            0.452                1                3
##  5 Adelai~ SA    Busine~          0.464            0.179                3                0
##  6 Adelai~ SA    Holiday          0.528            0.296                2                1
##  7 Adelai~ SA    Other            0.593            0.404                2                2
##  8 Adelai~ SA    Visiti~          0.488            0.254                0                3
##  9 Alice ~ NT    Busine~          0.534            0.251                0                1
## 10 Alice ~ NT    Holiday          0.381            0.832                3                1
## # ... with 294 more rows, and 5 more variables: spikiness <dbl>, linearity <dbl>,
## #   curvature <dbl>, stl_e_acf1 <dbl>, stl_e_acf10 <dbl>
```

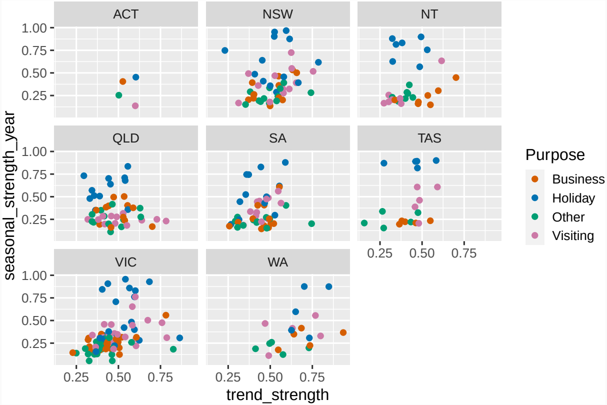
STL-based features

```
tourism %>%
  features(Trips, feat_stl) %>%
  ggplot(aes(x = trend_strength, y = seasonal_strength_year, col = Purpose)) +
  geom_point() + facet_wrap(vars(State))
```



STL-based features

```
tourism %>%
  features(Trips, feat_stl) %>%
  ggplot(aes(x = trend_strength, y = seasonal_strength_year, col = Purpose)) +
  geom_point() + facet_wrap(vars(State))
```

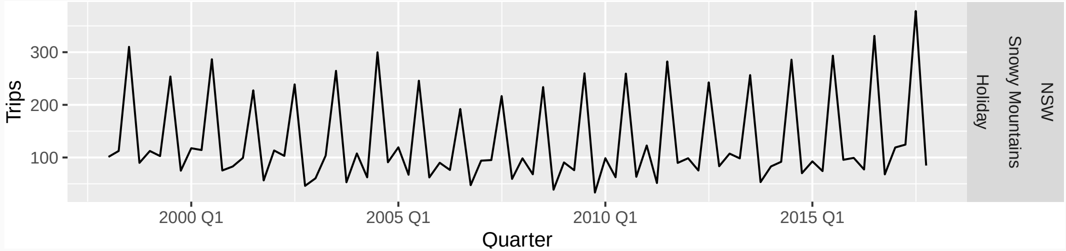


- Holidays more seasonal than other travel.
- WA has strongest trends.

STL-based features

Find the most seasonal time series:

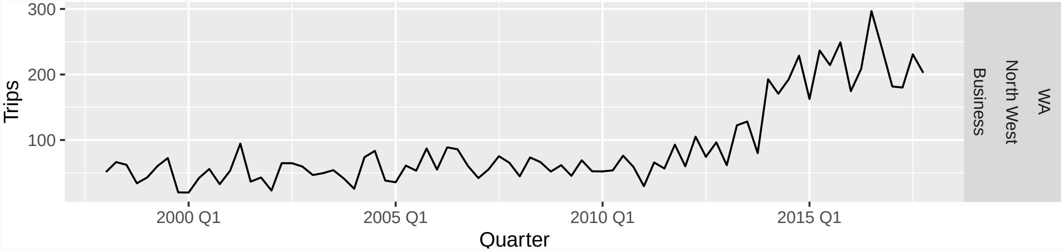
```r
tourism %>%
  features(Trips, feat_stl) %>%
  filter(seasonal_strength_year == max(seasonal_strength_year)) %>%
  left_join(tourism, by = c("State", "Region", "Purpose")) %>%
  ggplot(aes(x = Quarter, y = Trips)) +
  geom_line() +
  facet_grid(vars(State, Region, Purpose))
```



STL-based features

Find the most trended time series:

```
tourism %>%
  features(Trips, feat_stl) %>%
  filter(trend_strength == max(trend_strength)) %>%
  left_join(tourism, by = c("State", "Region", "Purpose")) %>%
  ggplot(aes(x = Quarter, y = Trips)) +
  geom_line() +
  facet_grid(vars(State, Region, Purpose))
```



Time series features

```
tourism_features <- tourism %>%
  features(Trips, feature_set(pkgs = "feasts"))
```

All features from the feasts package

```
## # A tibble: 304 x 50
##    Region  State Purpose trend_strength seasonal_streng~ seasonal_peak_y~ seasonal_trough~
##    <chr>   <chr> <chr>            <dbl>            <dbl>            <dbl>            <dbl>
##  1 Adelai~ SA    Busine~          0.464            0.407                3                1
##  2 Adelai~ SA    Holiday          0.554            0.619                1                2
##  3 Adelai~ SA    Other            0.746            0.202                2                1
##  4 Adelai~ SA    Visiti~          0.435            0.452                1                3
##  5 Adelai~ SA    Busine~          0.464            0.179                3                0
##  6 Adelai~ SA    Holiday          0.528            0.296                2                1
##  7 Adelai~ SA    Other            0.593            0.404                2                2
##  8 Adelai~ SA    Visiti~          0.488            0.254                0                3
##  9 Alice ~ NT    Busine~          0.534            0.251                0                1
## 10 Alice ~ NT    Holiday          0.381            0.832                3                1
## # ... with 294 more rows, and 43 more variables: spikiness <dbl>, linearity <dbl>,
## #   curvature <dbl>, stl_e_acf1 <dbl>, stl_e_acf10 <dbl>, acf1 <dbl>, acf10 <dbl>,
## #   diff1_acf1 <dbl>, diff1_acf10 <dbl>, diff2_acf1 <dbl>, diff2_acf10 <dbl>,
## #   season_acf1 <dbl>, pacf5 <dbl>, diff1_pacf5 <dbl>, diff2_pacf5 <dbl>,
## #   season_pacf <dbl>, zero_run_mean <dbl>, nonzero_squared_cv <dbl>,
## #   zero_start_prop <dbl>, zero_end_prop <dbl>, lambda_guerrero <dbl>, kpss_stat <dbl>,
## #   kpss_pvalue <dbl>, pp_stat <dbl>, pp_pvalue <dbl>, ndiffs <int>, nsdiffs <int>, ...
```

Reduced feature space

```r
pcs <- tourism_features %>%
  select(-State, -Region, -Purpose) %>%
  prcomp(scale = TRUE) %>%
  augment(tourism_features)
```

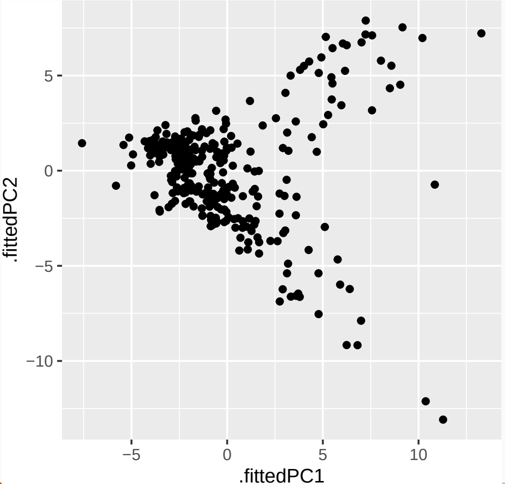
Principal components based on all features from the feasts package

```
## # A tibble: 304 x 98
##    .rownames Region         State Purpose trend_strength seasonal_streng~ seasonal_peak_y~
##    <chr>     <chr>          <chr> <chr>            <dbl>            <dbl>            <dbl>
##  1 1         Adelaide       SA    Busine~          0.464            0.407                3
##  2 2         Adelaide       SA    Holiday          0.554            0.619                1
##  3 3         Adelaide       SA    Other            0.746            0.202                2
##  4 4         Adelaide       SA    Visiti~          0.435            0.452                1
##  5 5         Adelaide Hills SA    Busine~          0.464            0.179                3
##  6 6         Adelaide Hills SA    Holiday          0.528            0.296                2
##  7 7         Adelaide Hills SA    Other            0.593            0.404                2
##  8 8         Adelaide Hills SA    Visiti~          0.488            0.254                0
##  9 9         Alice Springs  NT    Busine~          0.534            0.251                0
## 10 10        Alice Springs  NT    Holiday          0.381            0.832                3
## # ... with 294 more rows, and 91 more variables: seasonal_trough_year <dbl>,
## #   spikiness <dbl>, linearity <dbl>, curvature <dbl>, stl_e_acf1 <dbl>,
## #   stl_e_acf10 <dbl>, acf1 <dbl>, acf10 <dbl>, diff1_acf1 <dbl>, diff1_acf10 <dbl>,
## #   diff2_acf1 <dbl>, diff2_acf10 <dbl>, season_acf1 <dbl>, pacf5 <dbl>,
## #   diff1_pacf5 <dbl>, diff2_pacf5 <dbl>, season_pacf <dbl>, zero_run_mean <dbl>,
```

Reduced feature space

```r
pcs %>% ggplot(aes(x=.fittedPC1, y=.fittedPC2)) +
  geom_point() + theme(aspect.ratio=1)
```

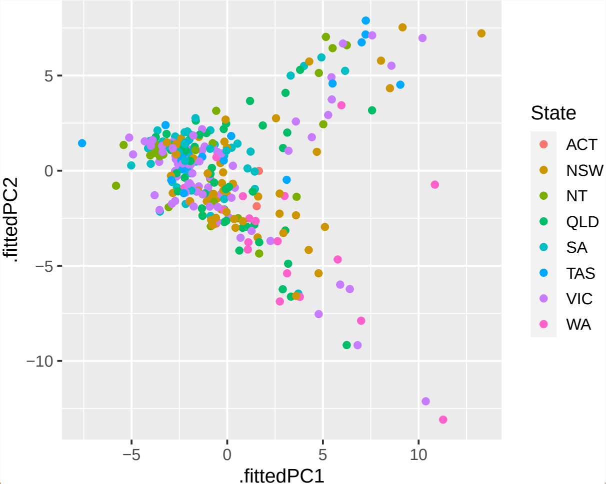
Principal components based on all features from the feasts package



Reduced feature space

```
pcs %>% ggplot(aes(x=.fittedPC1, y=.fittedPC2, col=State)) +
  geom_point() + theme(aspect.ratio=1)
```

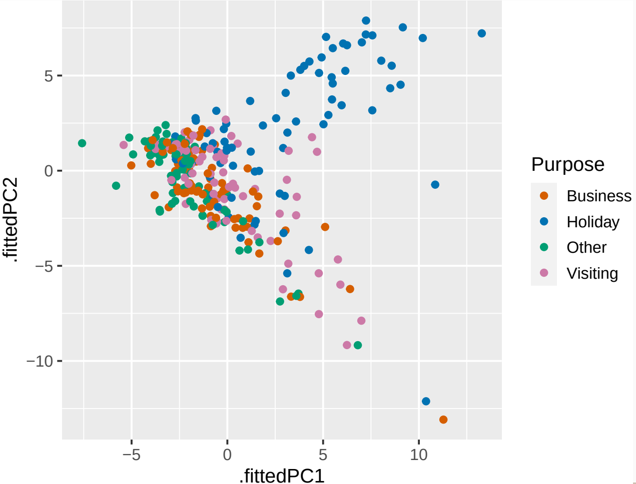
Principal components based on all features from the feasts package



Reduced feature space

```r
pcs %>% ggplot(aes(x=.fittedPC1, y=.fittedPC2, col=Purpose)) +
  geom_point() + theme(aspect.ratio=1)
```

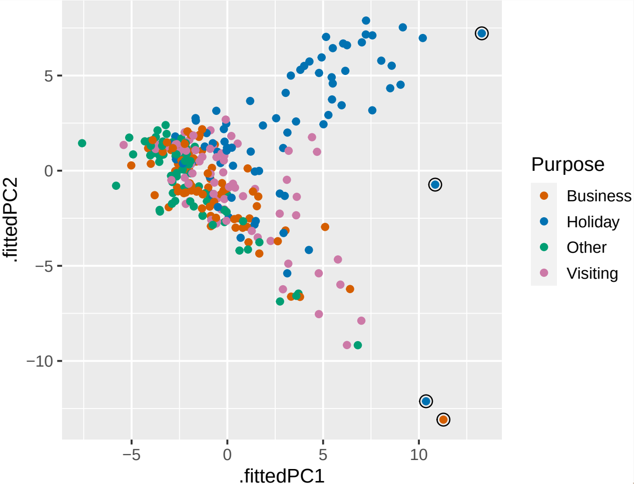
Principal components based on all features from the feasts package



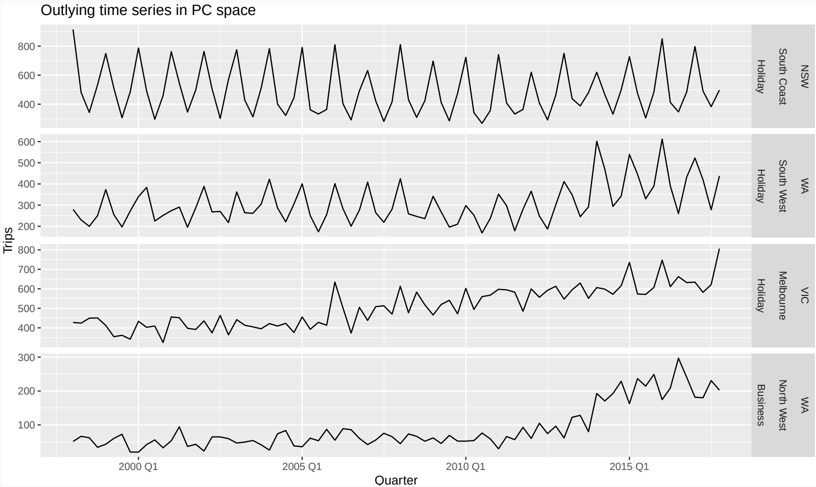
Anomaly detection using time series features

```
pcs %>% ggplot(aes(x=.fittedPC1, y=.fittedPC2, col=Purpose)) +
  geom_point() + theme(aspect.ratio=1)
```

Principal components based on all features from the feasts package



Anomaly detection using time series features



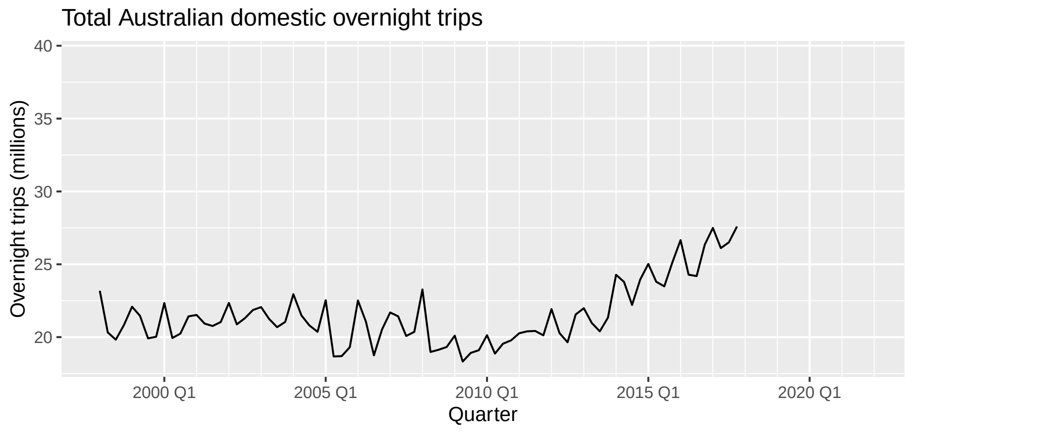
Outline

Random futures

A forecast is an estimate of the probability distribution of a variable to be observed in the future.

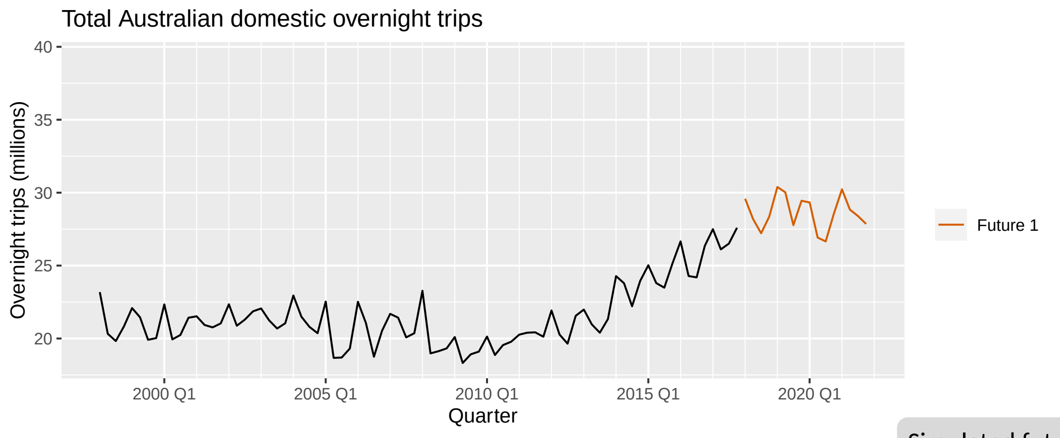
Random futures

A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Random futures

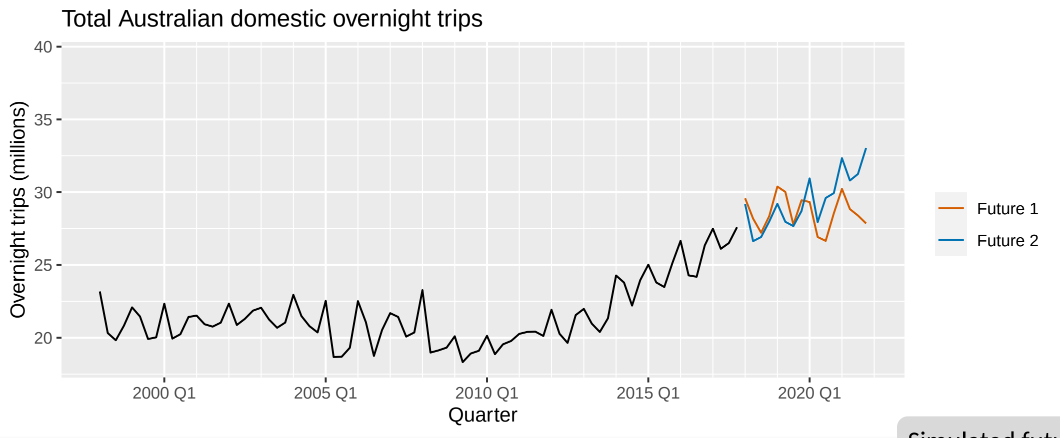
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

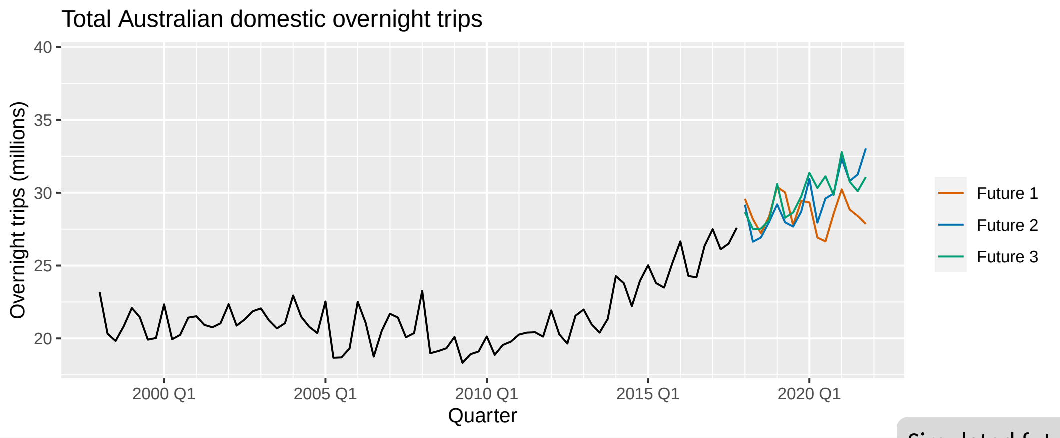
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

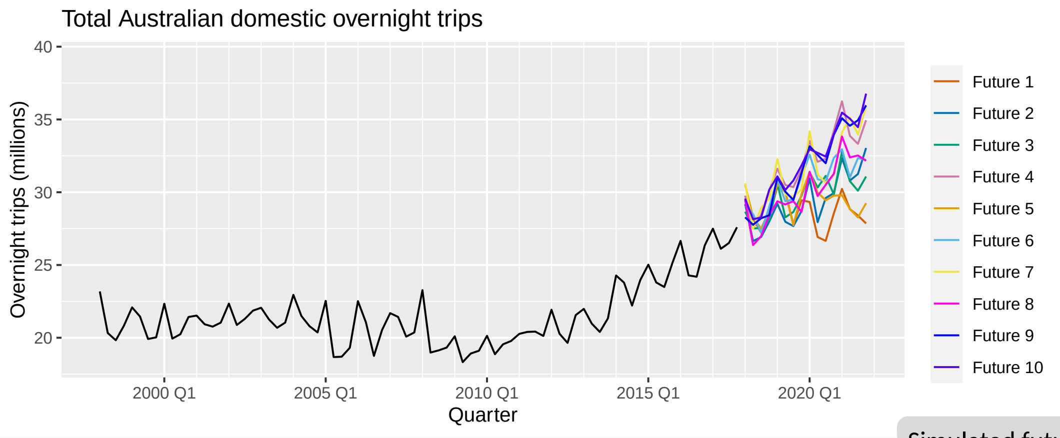
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

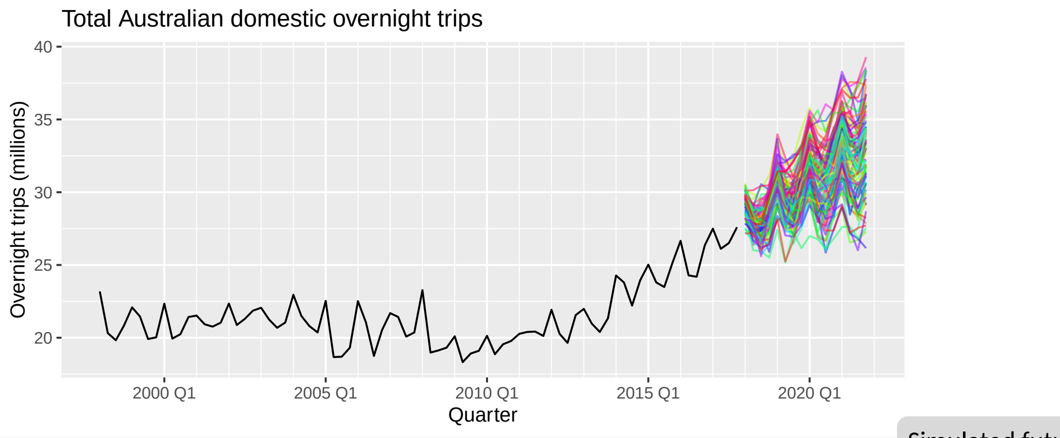
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

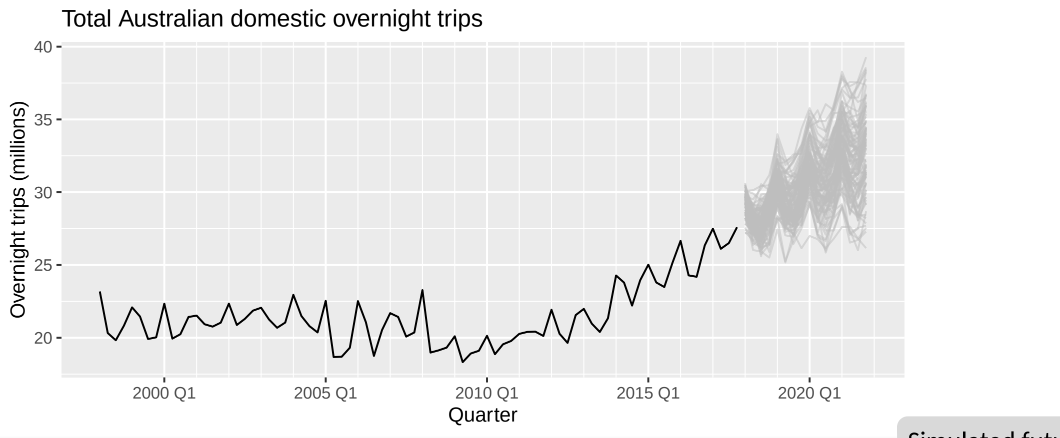
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

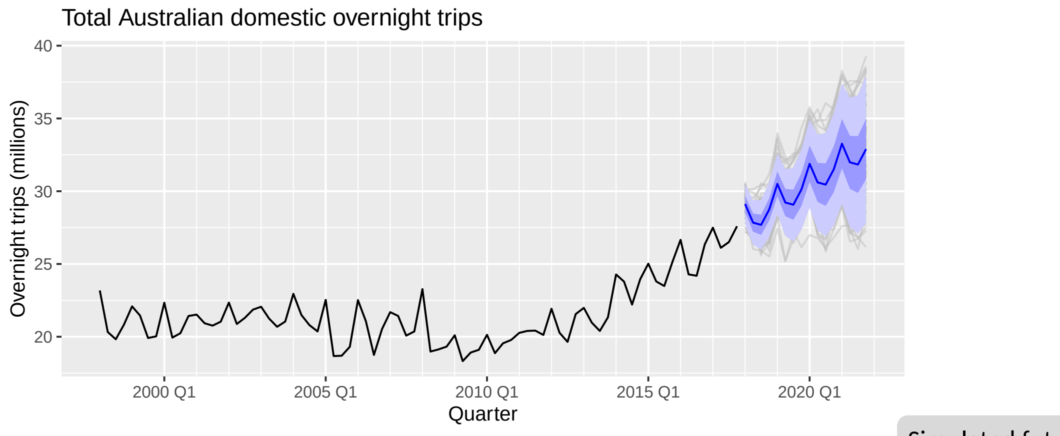
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

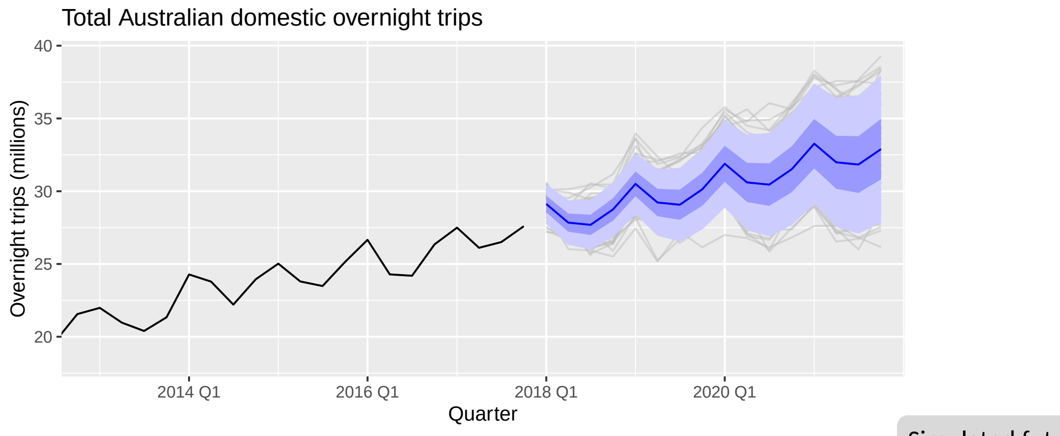
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Model fitting

```r
tourism_fit <- tourism %>%
  filter(year(Quarter) <= 2015) %>%
  model(ets = ETS(Trips), arima = ARIMA(Trips)) %>%
  mutate(ensemble = (ets + arima)/2)
```

```
## # A mable: 304 x 6
## # Key:     Region, State, Purpose [304]
##    Region         State Purpose          ets                             arima      ensemble
##    <chr>          <chr> <chr>        <model>                           <model>       <model>
##  1 Adelaide       SA    Busine~ <ETS(M,N,A)> <ARIMA(0,0,0)(1,0,1)[4] w/ mean> <COMBINATION>
##  2 Adelaide       SA    Holiday <ETS(M,N,A)> <ARIMA(0,0,0)(2,0,0)[4] w/ mean> <COMBINATION>
##  3 Adelaide       SA    Other   <ETS(M,A,N)>          <ARIMA(0,1,1) w/ drift> <COMBINATION>
##  4 Adelaide       SA    Visiti~ <ETS(A,N,A)> <ARIMA(0,0,0)(1,0,1)[4] w/ mean> <COMBINATION>
##  5 Adelaide Hil~ SA    Busine~ <ETS(A,N,N)>           <ARIMA(0,0,0) w/ mean> <COMBINATION>
##  6 Adelaide Hil~ SA    Holiday <ETS(A,A,N)>           <ARIMA(0,0,0) w/ mean> <COMBINATION>
##  7 Adelaide Hil~ SA    Other   <ETS(A,N,N)>         <ARIMA(2,1,1)(2,0,0)[4]> <COMBINATION>
##  8 Adelaide Hil~ SA    Visiti~ <ETS(M,A,A)>                  <ARIMA(0,1,1)> <COMBINATION>
##  9 Alice Springs NT    Busine~ <ETS(A,N,N)> <ARIMA(0,0,0)(0,0,1)[4] w/ mean> <COMBINATION>
## 10 Alice Springs NT    Holiday <ETS(M,N,A)>         <ARIMA(0,0,0)(0,1,2)[4]> <COMBINATION>
## # ... with 294 more rows
```

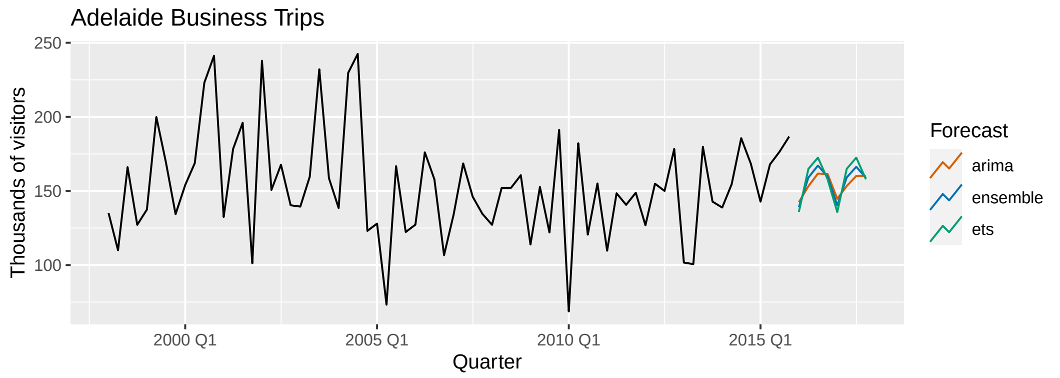
Producing forecasts

```r
tourism_fc <- tourism_fit %>%
  forecast(h = "2 years")
```

```
## # A fable: 7,296 x 7 [1Q]
## # Key:     Region, State, Purpose, .model [912]
##    Region   State Purpose  .model Quarter        Trips .mean
##    <chr>    <chr> <chr>    <chr>    <qtr>       <dist> <dbl>
##  1 Adelaide SA    Business ets    2016 Q1  N(136, 902)  136.
##  2 Adelaide SA    Business ets    2016 Q2 N(165, 1344)  165.
##  3 Adelaide SA    Business ets    2016 Q3 N(173, 1490)  173.
##  4 Adelaide SA    Business ets    2016 Q4 N(158, 1277)  158.
##  5 Adelaide SA    Business ets    2017 Q1  N(136, 979)  136.
##  6 Adelaide SA    Business ets    2017 Q2 N(165, 1422)  165.
##  7 Adelaide SA    Business ets    2017 Q3 N(173, 1569)  173.
##  8 Adelaide SA    Business ets    2017 Q4 N(158, 1356)  158.
##  9 Adelaide SA    Business arima  2016 Q1 N(142, 1232)  142.
## 10 Adelaide SA    Business arima  2016 Q2 N(153, 1232)  153.
## # ... with 7,286 more rows
```

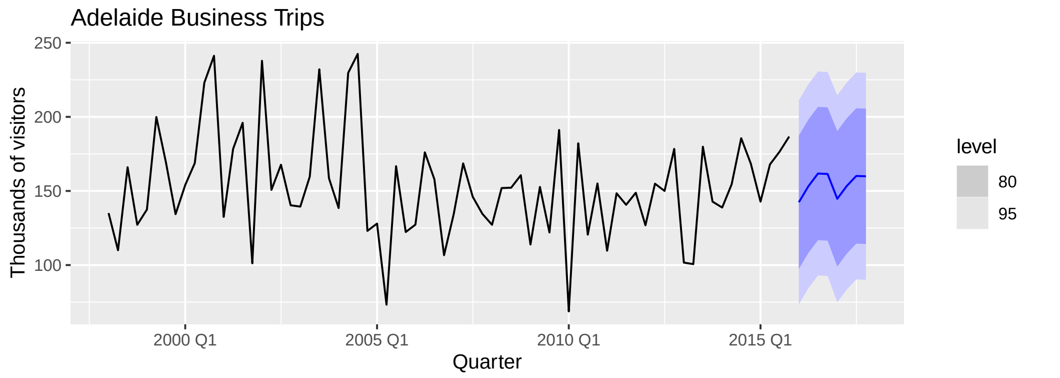
Visualising forecasts

```
tourism_fc %>%
  filter(Region == "Adelaide", Purpose=="Business") %>%
  autoplot(tourism, level = NULL) +
  labs(title = "Adelaide Business Trips", y = "Thousands of visitors") +
  guides(color = guide_legend(title = "Forecast"))
```



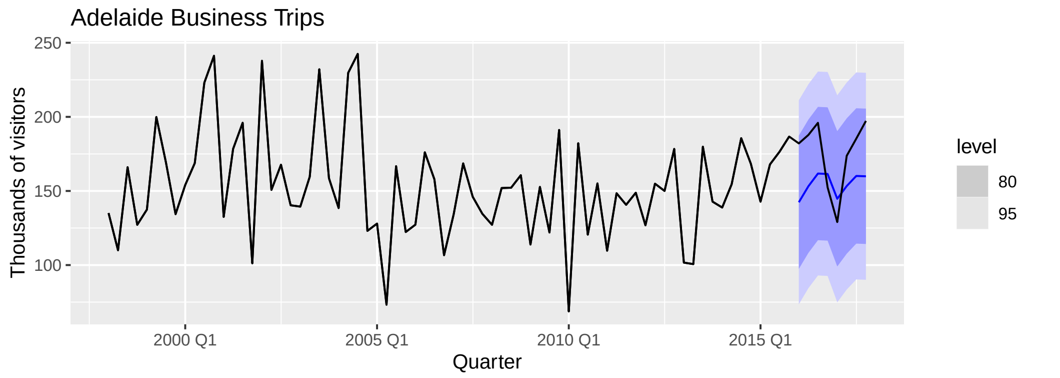
Visualising forecasts

```
tourism_fc %>%
  filter(Region == "Adelaide", Purpose=="Business", .model == "arima") %>%
  autoplot(tourism) +
  labs(title = "Adelaide Business Trips", y = "Thousands of visitors") +
  guides(color = guide_legend(title = "Forecast"))
```



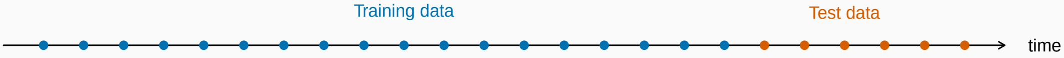
Visualising forecasts

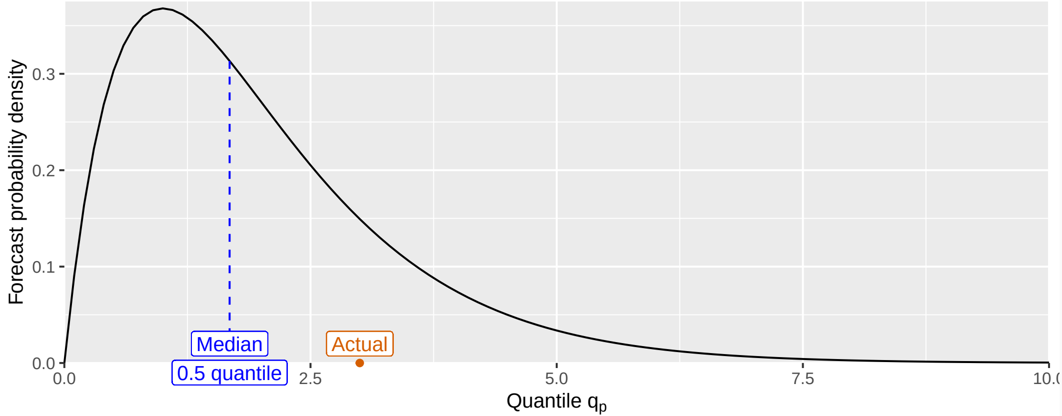
```
tourism_fc %>%
  filter(Region == "Adelaide", Purpose=="Business", .model == "arima") %>%
  autoplot(tourism) +
  labs(title = "Adelaide Business Trips", y = "Thousands of visitors") +
  guides(color = guide_legend(title = "Forecast"))
```



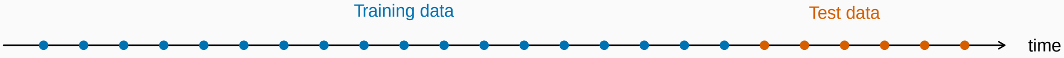
Outline

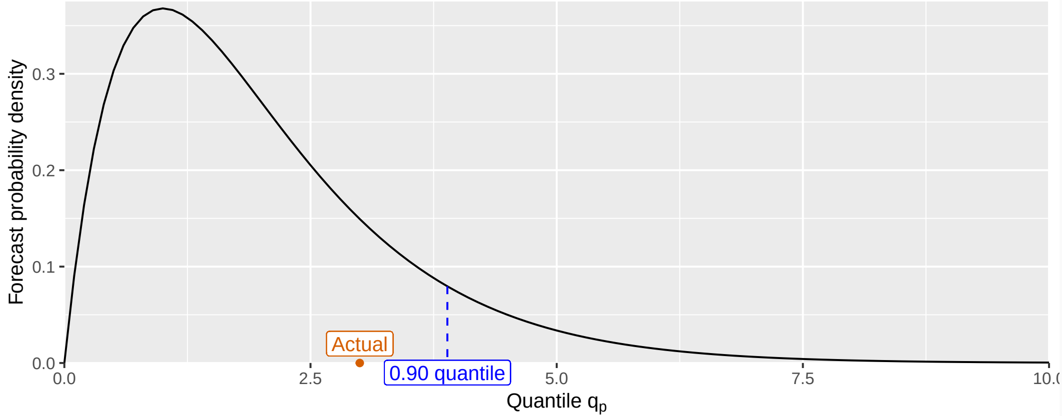
Evaluating probabilistic forecasts



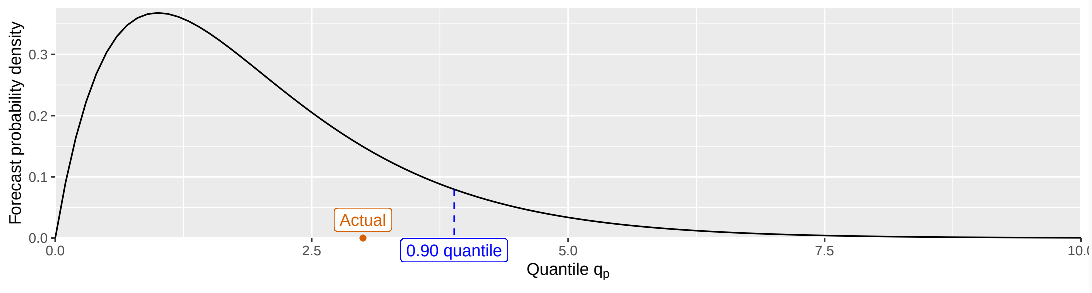


Evaluating probabilistic forecasts





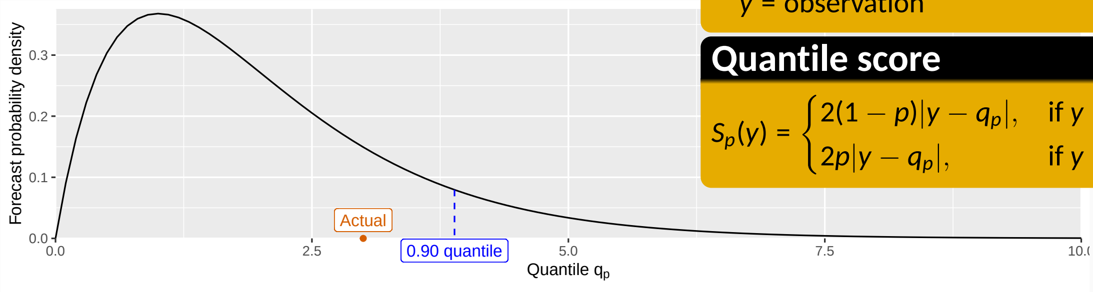
Evaluating probabilistic forecasts



Evaluating probabilistic forecasts

q_p = quantile forecast with prob. p

y = observation



Quantile score

$$S_p(y) = \begin{cases} 2(1-p)|y - q_p|, & \text{if } y < q_p \\ 2p|y - q_p|, & \text{if } y \geq q_p \end{cases}$$

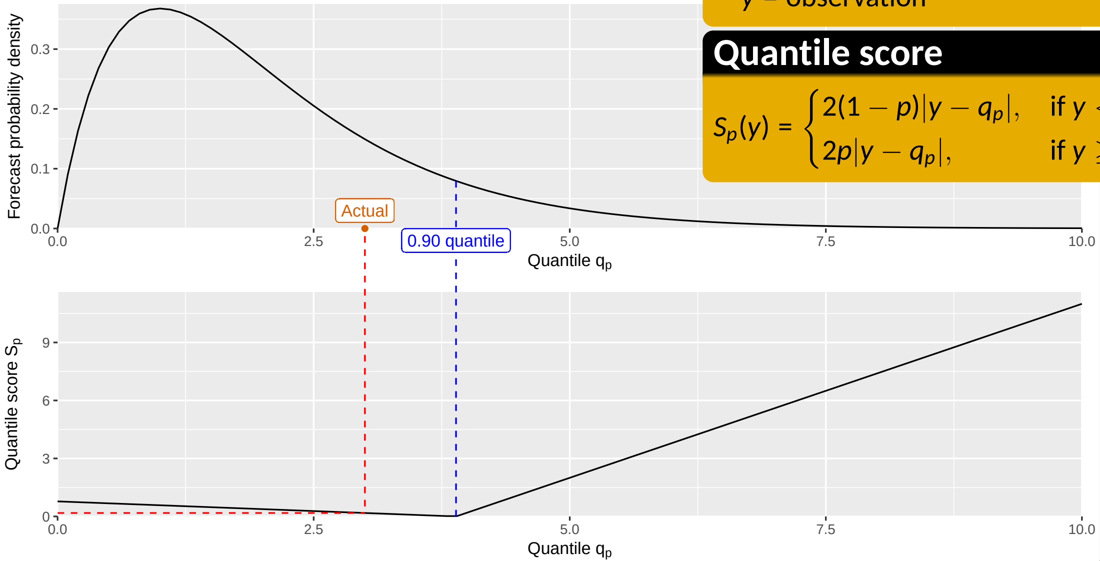
Evaluating probabilistic forecasts

q_p = quantile forecast with prob. p

y = observation

Quantile score

$$S_p(y) = \begin{cases} 2(1-p)|y - q_p|, & \text{if } y < q_p \\ 2p|y - q_p|, & \text{if } y \geq q_p \end{cases}$$



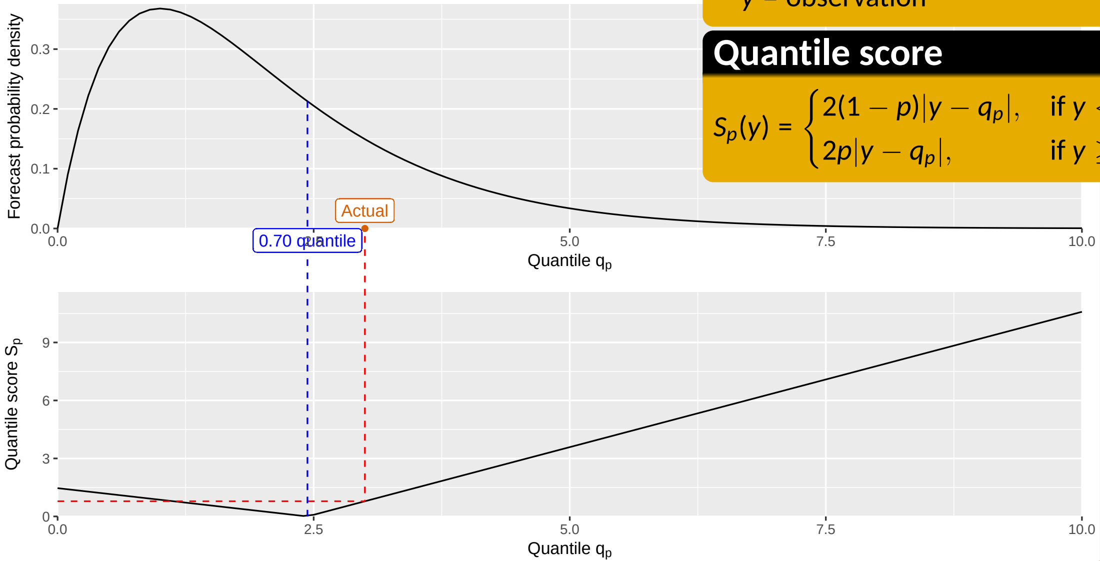
Evaluating probabilistic forecasts

q_p = quantile forecast with prob. p

y = observation

Quantile score

$$S_p(y) = \begin{cases} 2(1-p)|y-q_p|, & \text{if } y < q_p \\ 2p|y-q_p|, & \text{if } y \geq q_p \end{cases}$$



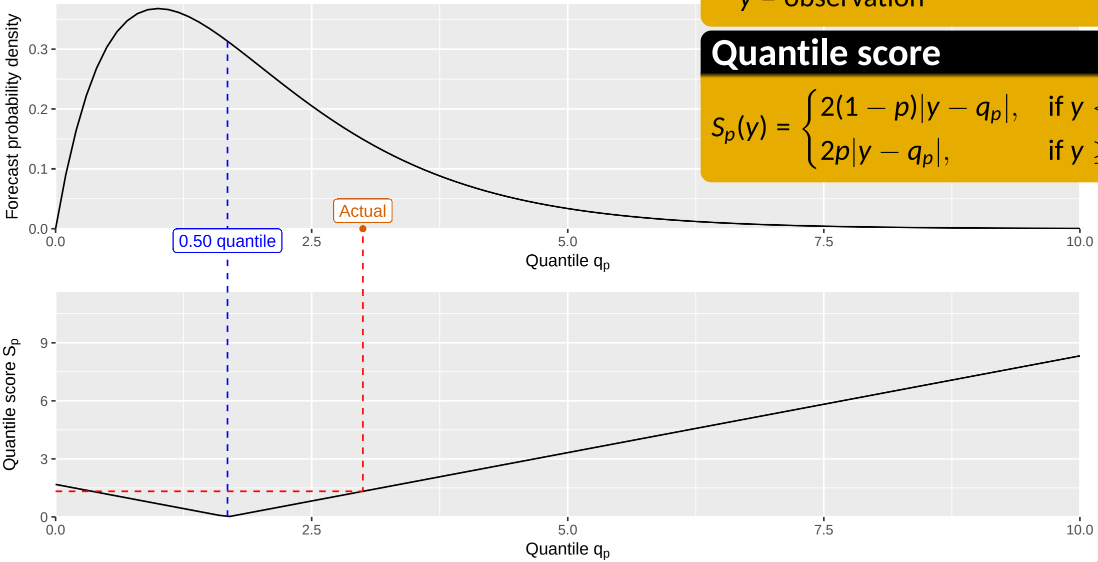
Evaluating probabilistic forecasts

q_p = quantile forecast with prob. p

y = observation

Quantile score

$$S_p(y) = \begin{cases} 2(1-p)|y-q_p|, & \text{if } y < q_p \\ 2p|y-q_p|, & \text{if } y \geq q_p \end{cases}$$



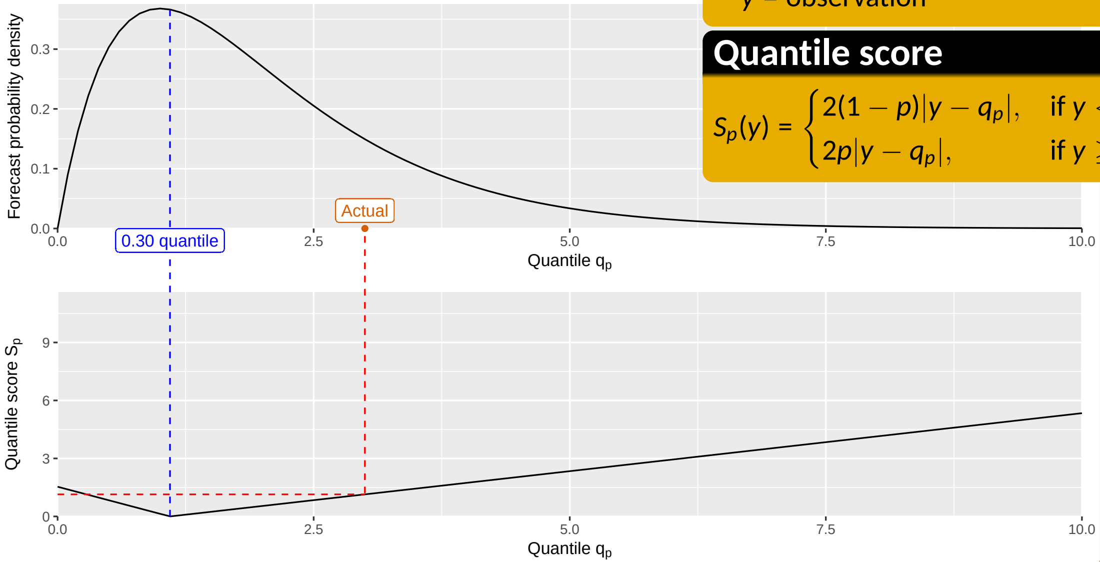
Evaluating probabilistic forecasts

q_p = quantile forecast with prob. p

y = observation

Quantile score

$$S_p(y) = \begin{cases} 2(1-p)|y-q_p|, & \text{if } y < q_p \\ 2p|y-q_p|, & \text{if } y \geq q_p \end{cases}$$



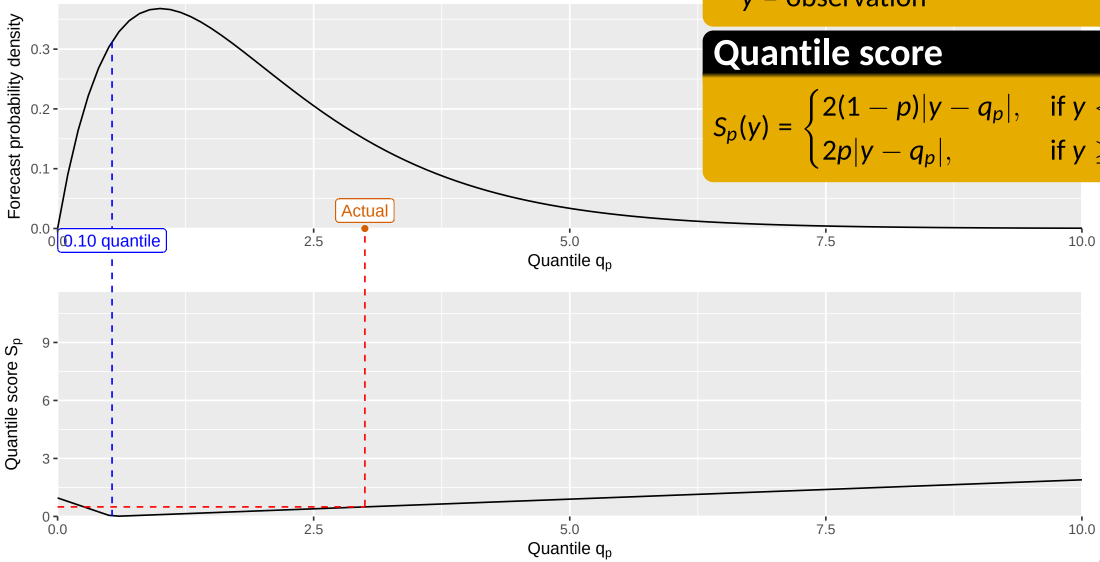
Evaluating probabilistic forecasts

q_p = quantile forecast with prob. p

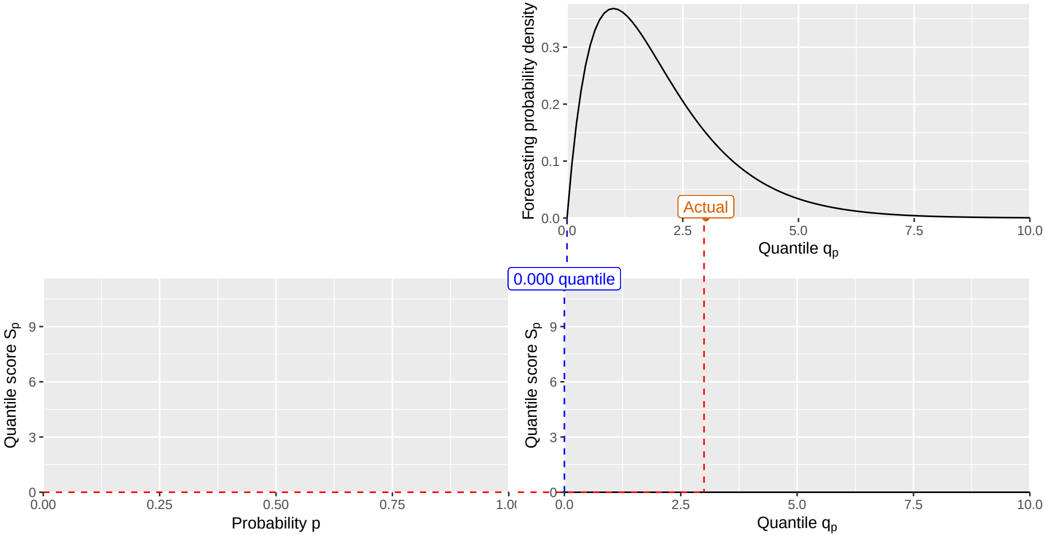
y = observation

Quantile score

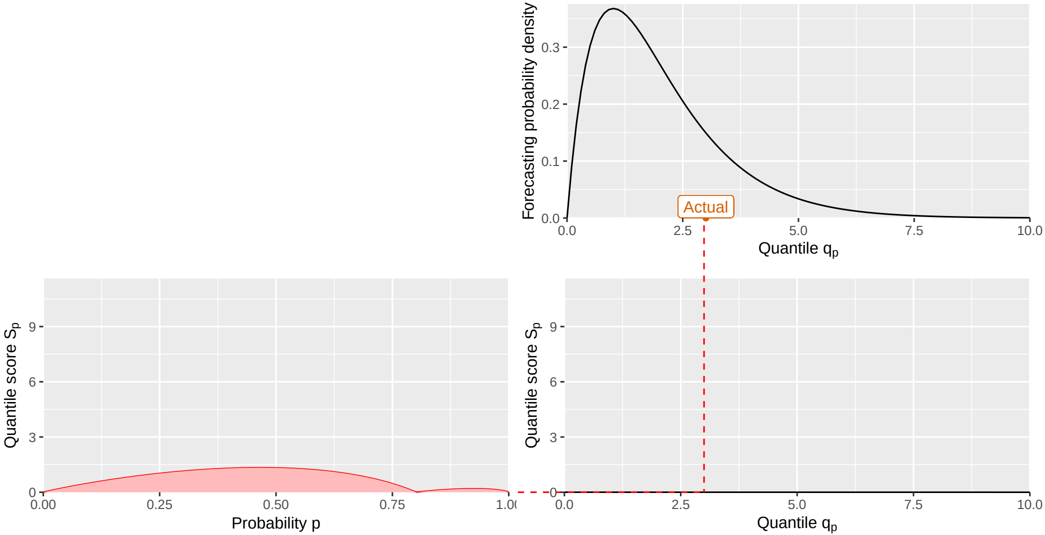
$$S_p(y) = \begin{cases} 2(1-p)|y-q_p|, & \text{if } y < q_p \\ 2p|y-q_p|, & \text{if } y \geq q_p \end{cases}$$



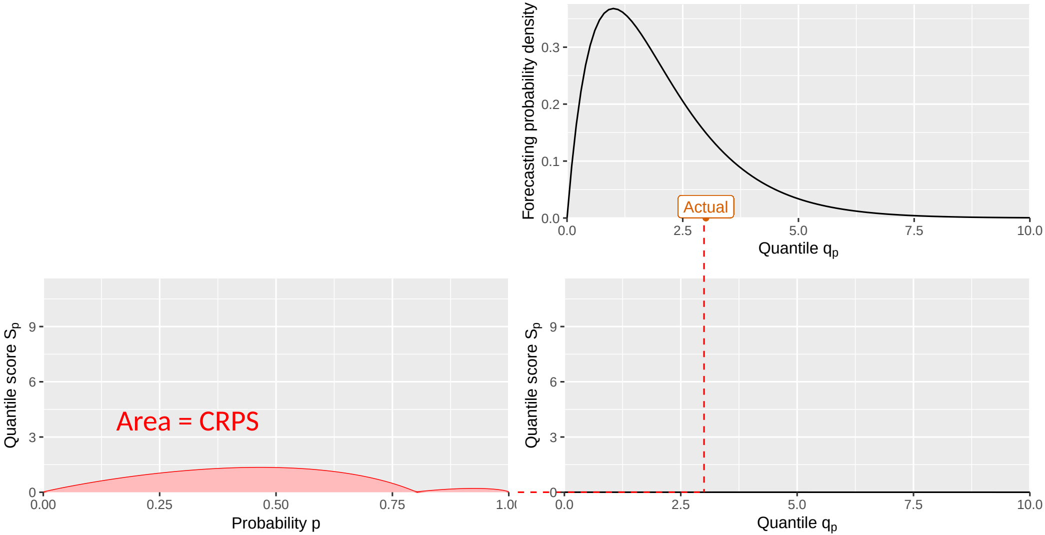
Evaluating probabilistic forecasts



Evaluating probabilistic forecasts



Evaluating probabilistic forecasts



Evaluating probabilistic forecasts

```r
tourism_fc %>%
  accuracy(tourism, measures = list(MSE=MSE, CRPS=CRPS))
```

```
## # A tibble: 912 x 7
##    .model Region         State Purpose  .type    MSE   CRPS
##    <chr>  <chr>          <chr> <chr>    <chr>  <dbl>  <dbl>
##  1 arima  Adelaide       SA    Business Test   840.   17.1
##  2 arima  Adelaide       SA    Holiday  Test   968.   18.1
##  3 arima  Adelaide       SA    Other    Test   188.    7.95
##  4 arima  Adelaide       SA    Visiting Test  1302.   21.4
##  5 arima  Adelaide Hills SA    Business Test    21.7   2.39
##  6 arima  Adelaide Hills SA    Holiday  Test    52.4   4.18
##  7 arima  Adelaide Hills SA    Other    Test     2.31  0.893
##  8 arima  Adelaide Hills SA    Visiting Test   144.    7.08
##  9 arima  Alice Springs  NT    Business Test   150.    7.78
## 10 arima  Alice Springs  NT    Holiday  Test    93.2   5.59
## # ... with 902 more rows
```

Evaluating probabilistic forecasts

```r
tourism_fc %>%
  accuracy(tourism, measures = list(SS_MSE=skill_score(MSE), SS_CRPS=skill_score(CRPS)))
```

```
## # A tibble: 912 x 7
##    .model Region         State Purpose  .type SS_MSE   SS_CRPS
##    <chr>  <chr>          <chr> <chr>    <chr>  <dbl>     <dbl>
##  1 arima  Adelaide       SA    Business Test  -0.719 -0.00789
##  2 arima  Adelaide       SA    Holiday  Test   0.394  0.212
##  3 arima  Adelaide       SA    Other    Test   0.787  0.578
##  4 arima  Adelaide       SA    Visiting Test  -1.01  -0.354
##  5 arima  Adelaide Hills SA    Business Test   0.693  0.508
##  6 arima  Adelaide Hills SA    Holiday  Test  -0.568 -0.136
##  7 arima  Adelaide Hills SA    Other    Test   0.834  0.565
##  8 arima  Adelaide Hills SA    Visiting Test   0.120  0.0788
##  9 arima  Alice Springs  NT    Business Test  -1.43  -0.696
## 10 arima  Alice Springs  NT    Holiday  Test   0.468  0.186
## # ... with 902 more rows
```

Evaluating probabilistic forecasts

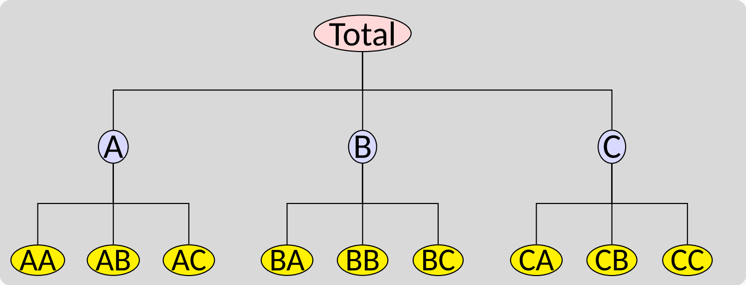
```r
tourism_fc %>%
  accuracy(tourism, measures = list(SS_MSE=skill_score(MSE), SS_CRPS=skill_score(CRPS))) %>%
  group_by(.model) %>%
  summarise(SS_MSE = mean(SS_MSE), SS_CRPS=mean(SS_CRPS)) %>%
  arrange(desc(SS_CRPS))
```

```
## # A tibble: 3 x 3
##   .model   SS_MSE SS_CRPS
##   <chr>     <dbl>   <dbl>
## 1 ets      0.155   0.138
## 2 ensemble 0.141   0.138
## 3 arima    0.0636  0.0999
```

Outline

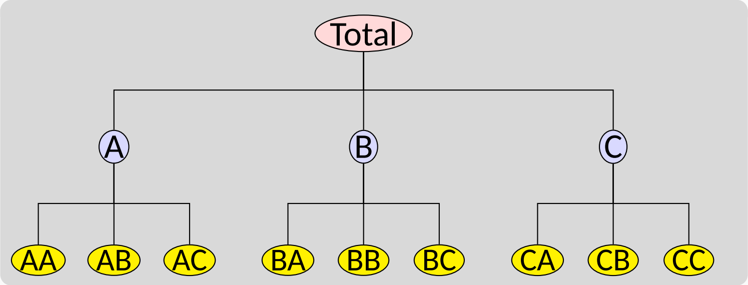
Hierarchical time series

A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.



Hierarchical time series

A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

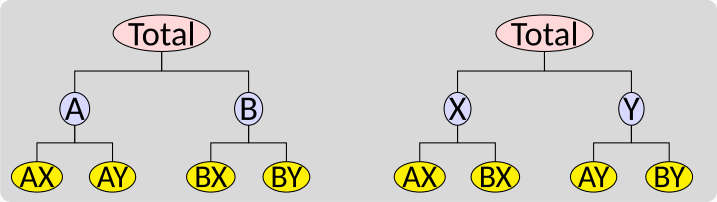


Examples

- Tourism demand by states, zones, regions

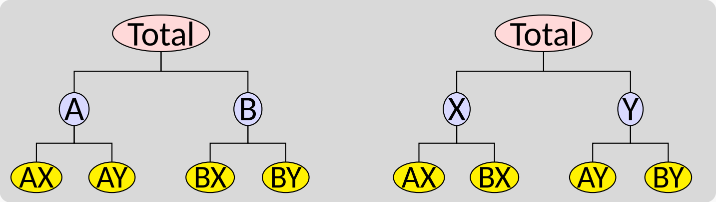
Grouped time series

A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.



Grouped time series

A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.



Examples

- Tourism by state and purpose of travel
- Retail sales by product groups/sub groups, and by countries/regions

Creating aggregates

```r
tourism %>%
  aggregate_key(Purpose * (State / Region), Trips = sum(Trips)) %>%
  filter(Quarter == yearquarter("1998 Q1")) %>%
  print(n = 15)
```

```
## # A tsibble: 425 x 5 [1Q]
## # Key:       Purpose, State, Region [425]
##    Quarter Purpose      State        Region        Trips
##      <qtr> <chr*>       <chr*>       <chr*>        <dbl>
##  1 1998 Q1 <aggregated> <aggregated> <aggregated> 23182.
##  2 1998 Q1 Business     <aggregated> <aggregated>  3599.
##  3 1998 Q1 Holiday      <aggregated> <aggregated> 11806.
##  4 1998 Q1 Other        <aggregated> <aggregated>   680.
##  5 1998 Q1 Visiting     <aggregated> <aggregated>  7098.
##  6 1998 Q1 <aggregated> ACT          <aggregated>   551.
##  7 1998 Q1 <aggregated> NSW          <aggregated>  8040.
##  8 1998 Q1 <aggregated> NT           <aggregated>   181.
##  9 1998 Q1 <aggregated> QLD          <aggregated>  4041.
## 10 1998 Q1 <aggregated> SA           <aggregated>  1735.
## 11 1998 Q1 <aggregated> TAS          <aggregated>   982.
## 12 1998 Q1 <aggregated> VIC          <aggregated>  6010.
## 13 1998 Q1 <aggregated> WA           <aggregated>  1641.
```

Creating aggregates

- A grouped structure is specified using `grp1 * grp2`
- A nested structure is specified via `parent / child`.
- Groups and nesting can be mixed:

```
(country/region/city) * (brand/product)
```

- All possible aggregates are produced.
- These are useful when forecasting at different levels of aggregation.

The problem

1. How to forecast time series at all nodes such that the forecasts add up in the same way as the original data?
2. Can we exploit relationships between the series to improve the forecasts?

The problem

1. How to forecast time series at all nodes such that the forecasts add up in the same way as the original data?
2. Can we exploit relationships between the series to improve the forecasts?

The solution

1. Forecast all series at all levels of aggregation using an automatic forecasting algorithm.
   (e.g., `ETS`, `ARIMA`, ...)
2. Reconcile the resulting forecasts so they add up correctly using least squares optimization (i.e., find closest reconciled forecasts to the original forecasts).
3. This is available using `reconcile()`.

Forecast reconciliation

```
tourism %>%
  aggregate_key(Purpose * (State / Region), Trips = sum(Trips)) %>%
  model(ets = ETS(Trips)) %>%
  reconcile(ets_adjusted = min_trace(ets)) %>%
  forecast(h = 2)
```

```
## # A fable: 1,700 x 7 [1Q]
## # Key:     Purpose, State, Region, .model [850]
##    Purpose  State  Region       .model       Quarter       Trips .mean
##    <chr*>   <chr*> <chr*>       <chr>          <qtr>      <dist> <dbl>
##  1 Business ACT    Canberra     ets          2018 Q1 N(144, 1119) 144.
##  2 Business ACT    Canberra     ets          2018 Q2 N(203, 2260) 203.
##  3 Business ACT    Canberra     ets_adjusted 2018 Q1  N(157, 539) 157.
##  4 Business ACT    Canberra     ets_adjusted 2018 Q2  N(214, 951) 214.
##  5 Business ACT    <aggregated> ets          2018 Q1 N(144, 1119) 144.
##  6 Business ACT    <aggregated> ets          2018 Q2 N(203, 2260) 203.
##  7 Business ACT    <aggregated> ets_adjusted 2018 Q1  N(157, 539) 157.
##  8 Business ACT    <aggregated> ets_adjusted 2018 Q2  N(214, 951) 214.
```

Hierarchical and grouped time series

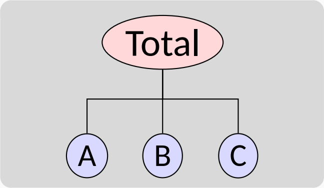
Every collection of time series with aggregation constraints can be written as

$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$$

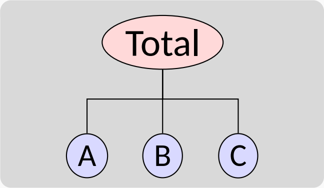
where

- $\boldsymbol{y}_t$ is a vector of all series at time t
- $\boldsymbol{b}_t$ is a vector of the most disaggregated series at time t
- $\boldsymbol{S}$ is a "summing matrix" containing the aggregation constraints.

Hierarchical time series



Hierarchical time series

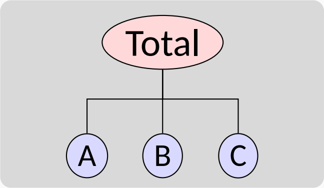


y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

Hierarchical time series



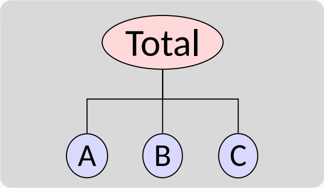
y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}$$

Hierarchical time series



y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

$$\boxed{\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t}$$

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.
(In general, they will not "add up".)

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.
(In general, they will not "add up".)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{G}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{G}$.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.
(In general, they will not "add up".)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{G}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{G}$.

- $\boldsymbol{G}$ extracts and combines base forecasts $\hat{\boldsymbol{y}}_n(h)$ to get bottom-level forecasts.
- $\boldsymbol{S}$ adds them up

Optimal combination forecasts

Main result

The best (minimum sum of variances) unbiased forecasts are obtained when $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}$, where $\boldsymbol{W}_h$ is the h-step base forecast error covariance matrix.

Optimal combination forecasts

Main result

The best (minimum sum of variances) unbiased forecasts are obtained when $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}$, where $\boldsymbol{W}_h$ is the h-step base forecast error covariance matrix.

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}\hat{\boldsymbol{y}}_n(h)$$

Problem: $\boldsymbol{W}_h$ hard to estimate, especially for $h > 1$.

Solutions:

- Ignore $\boldsymbol{W}_h$ (OLS)
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$ is diagonal (WLS)
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$ and use a shrinkage estimator (GLS)

Features

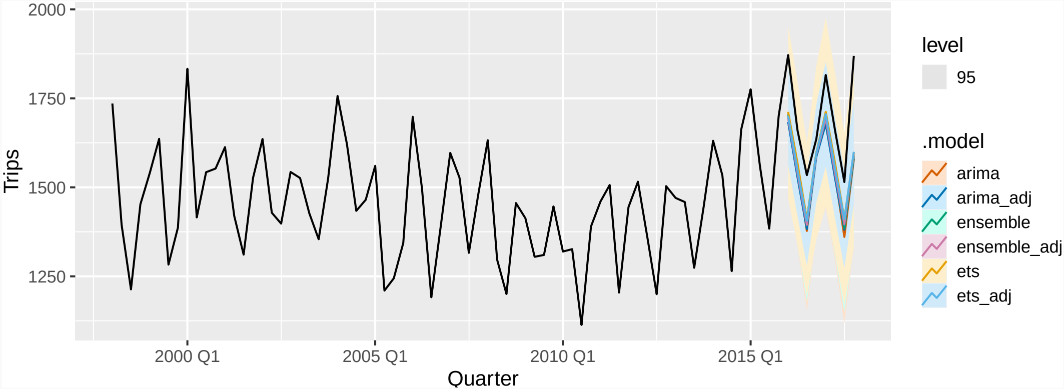
- Covariates can be included in initial forecasts.
- Adjustments can be made to initial forecasts at any level.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: regression of base forecasts on structure matrix.

Example: Australian tourism

```r
tourism_agg <- tourism %>%
  aggregate_key(Purpose * (State / Region),
    Trips = sum(Trips)
  )
fc <- tourism_agg %>%
  filter(year(Quarter) <= 2015) %>%
  model(ets = ETS(Trips), arima = ARIMA(Trips)) %>%
  mutate(ensemble = (ets + arima)/2) %>%
  reconcile(
    ets_adj = min_trace(ets, method="mint_shrink"),
    arima_adj = min_trace(arima, method="mint_shrink"),
    ensemble_adj = min_trace(ensemble, method="mint_shrink")
  ) %>%
  forecast(h = "2 years")
```

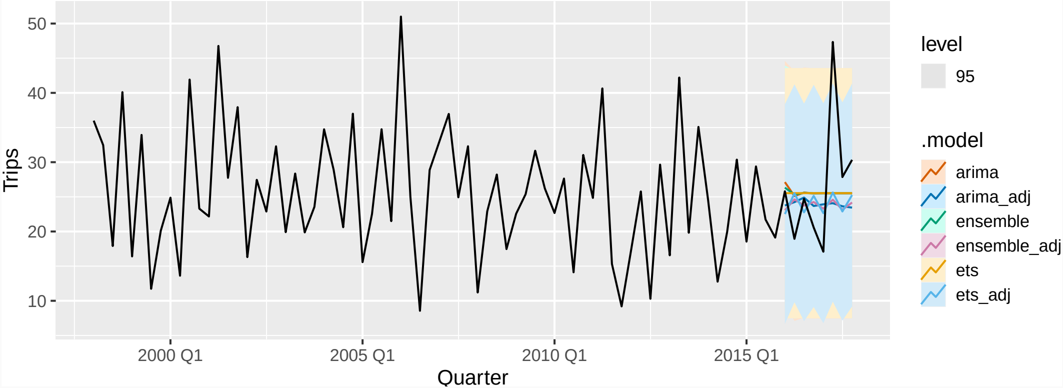
Example: Australian tourism

```r
fc %>%
  filter(State == "SA" & is_aggregated(Region) & is_aggregated(Purpose)) %>%
  autoplot(tourism_agg, level = 95)
```



Example: Australian tourism

```
fc %>%
  filter(Region == "Barossa" & Purpose == "Holiday") %>%
  autoplot(tourism_agg, level = 95)
```



Forecast evaluation

```
fc %>%
  accuracy(tourism_agg,
    measures = list(SS_MSE=skill_score(MSE), SS_CRPS=skill_score(CRPS)))
```

```
## # A tibble: 2,550 x 7
##    .model Purpose  State  Region                   .type   SS_MSE SS_CRPS
##    <chr>  <chr*>   <chr*> <chr*>                   <chr>    <dbl>   <dbl>
##  1 arima  Business ACT    Canberra                 Test   0.360     0.229
##  2 arima  Business ACT    <aggregated>             Test   0.360     0.229
##  3 arima  Business NSW    Blue Mountains           Test   0.439     0.293
##  4 arima  Business NSW    Capital Country          Test  -0.282    -0.0154
##  5 arima  Business NSW    Central Coast            Test  -0.0797   -0.101
##  6 arima  Business NSW    Central NSW              Test   0.426     0.245
##  7 arima  Business NSW    Hunter                   Test   0.187     0.0304
##  8 arima  Business NSW    New England North West Test  -0.200    -0.238
##  9 arima  Business NSW    North Coast NSW          Test  -0.00691 -0.0995
```

Forecast evaluation

```
fc %>%
  accuracy(tourism_agg,
    measures = list(SS_MSE=skill_score(MSE), SS_CRPS=skill_score(CRPS))) %>%
  group_by(.model) %>%
  summarise(SS_MSE = mean(SS_MSE), SS_CRPS=mean(SS_CRPS)) %>%
  arrange(desc(SS_CRPS))
```

```
## # A tibble: 6 x 3
##   .model        SS_MSE SS_CRPS
##   <chr>          <dbl>   <dbl>
## 1 ensemble_adj  0.195   0.146
## 2 ets_adj       0.199   0.143
## 3 arima_adj     0.120   0.113
## 4 ets           0.0904  0.0945
## 5 ensemble      0.0598  0.0893
## 6 arima        -0.0635  0.0405
```

More information

- Slides and papers: **robjhyndman.com**
- Packages: **tidyverts.org**
- Forecasting textbook using tidyverts package: **OTexts.com/fpp3**

Find me at ...

@robjhyndman

@robjhyndman

robjhyndman.com

rob.hyndman@monash.edu