Intro to tidyverts

Sri Seshadri

Conference training material:

https://github.com/rstudio-conf-2020/time-series-forecasting

Package website:

https://tidyverts.org

Technical reference:

https://otexts.com/fpp3/

This slide deck:

https://github.com/srivathsesh/IndyUseRTimeSeries

CSS stolen from:

https://github.com/rstudio-conf-2020/applied-ml

A package that loads it all

```r
library(fpp3)
```

```
## ── Attaching packages ────────────────────────────────────────────

## ✓ tibble      3.0.1     ✓ tsibble     0.8.6
## ✓ dplyr       0.8.5     ✓ tsibbledata 0.1.0
## ✓ tidyr       1.0.2     ✓ feasts      0.1.2
## ✓ lubridate   1.7.4     ✓ fable       0.1.1
## ✓ ggplot2     3.2.1

## ── Conflicts ──────────────────────────────────────────────────────
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x tsibble::id()            masks dplyr::id()
## x tsibble::interval()      masks lubridate::interval()
## x dplyr::lag()             masks stats::lag()
## x tsibble::new_interval() masks lubridate::new_interval()
```

```r
library(purrr)
```

Why `tidyverts`?

1. Packages like stats, forecast, astsa requires input data as a univariate time series data.
    - What if you have a large data comprised of many time series? many variables floating in R's envioronment?
2. Interface with many methods in `tidyverse`.
3. Awesome plotting methods.
4. Interface with `broom` to inspect model.
5. Easy evaluation of model metrics.

```
download.file("http://robjhyndman.com/data/tourism.xlsx", tourism_file <- tempfile())
my_tourism <- readxl::read_excel(tourism_file)
my_tourism %>% head()
```

```
## # A tibble: 6 x 5
##   Quarter    Region   State           Purpose  Trips
##   <chr>      <chr>    <chr>           <chr>    <dbl>
## 1 1998-01-01 Adelaide South Australia Business  135.
## 2 1998-04-01 Adelaide South Australia Business  110.
## 3 1998-07-01 Adelaide South Australia Business  166.
## 4 1998-10-01 Adelaide South Australia Business  127.
## 5 1999-01-01 Adelaide South Australia Business  137.
## 6 1999-04-01 Adelaide South Australia Business  200.
```

There are 80 `Quarters` - 20 years of data.

8 `States` nests 76 `Regions` There are 4 `Purposes`

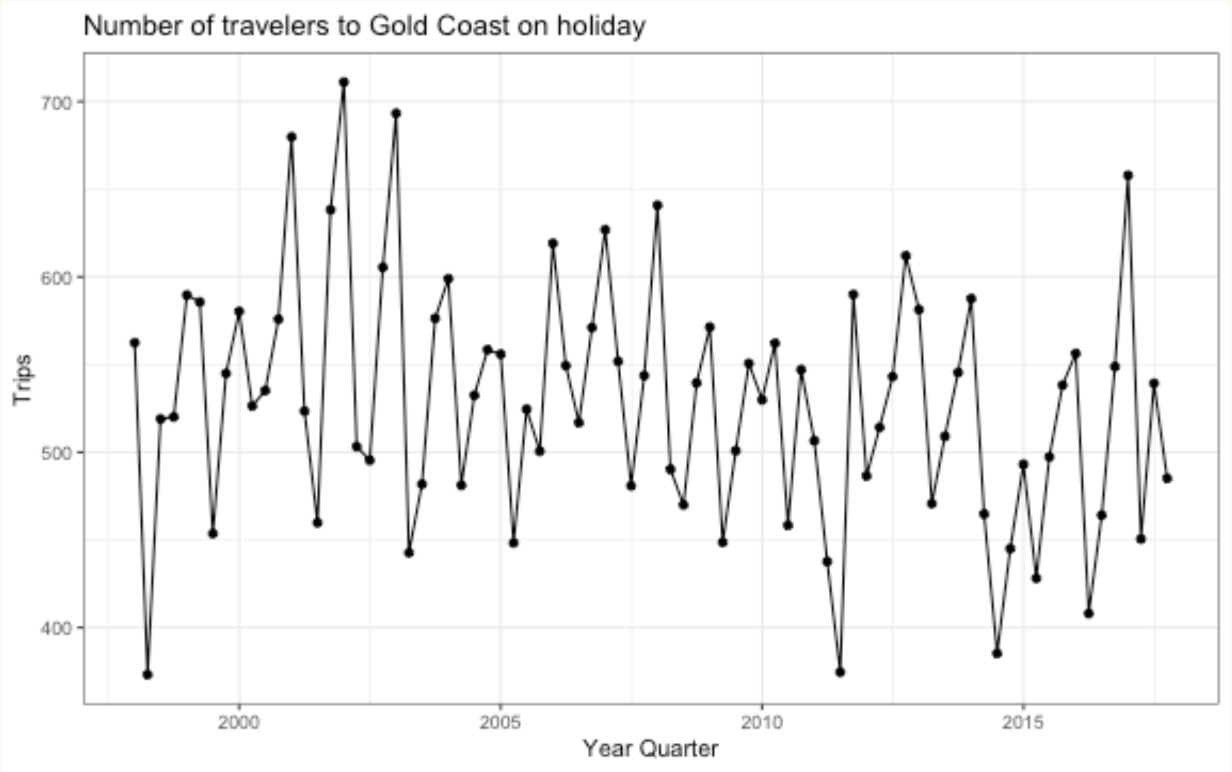
```
my_tourism %>%
  distinct(Purpose)
```

```
## # A tibble: 4 x 1
##   Purpose
##   <chr>
## 1 Business
## 2 Holiday
## 3 Other
## 4 Visiting
```

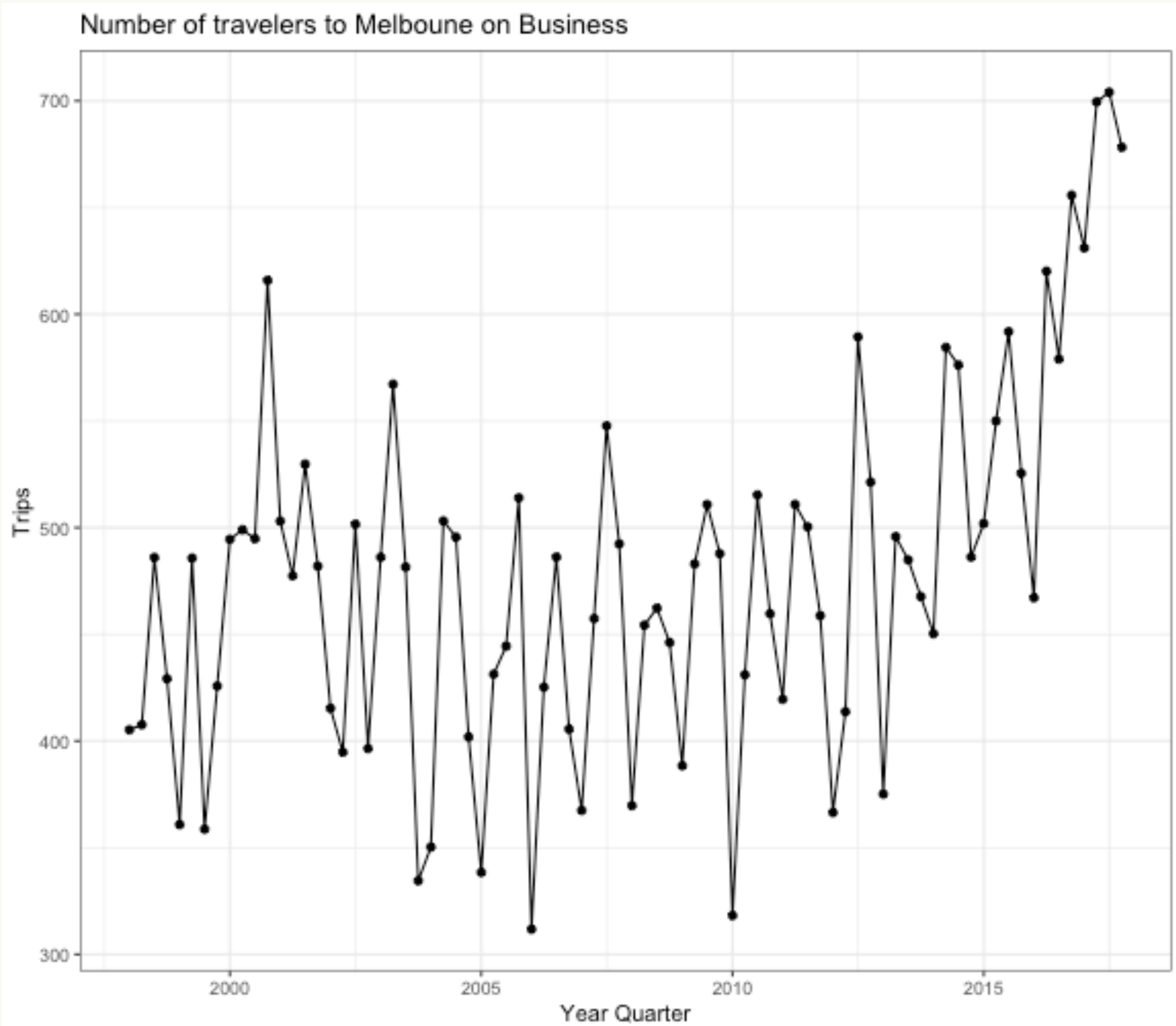
There are `76 * 4 = 304` unique time series. So, `76 * 4 * 80 = 24320` data points.

Visualize

```
my_tourism %>%
 filter(Region == 'Gold Coast', Purpose == 'Holiday') %>%
  ggplot(aes(x=yearquarter(Quarter), y = Trips)) +
  geom_point() + geom_line() +
  labs(x = "Year Quarter", y = "Trips", title = "Number of travelers to Gold Coast on holiday") +
  theme_bw()
```



```
my_tourism %>%
 filter(Region == 'Melbourne', Purpose == 'Business') %>%
  ggplot(aes(x=yearquarter(Quarter), y = Trips)) +
  geom_point() + geom_line() +
  labs(x = "Year Quarter", y = "Trips", title = "Number of travelers to Melboune on Business") +
  theme_bw()
```



Investigating time series

To...

- Decompose series into trend, season and remainders
- Fit models

We need the data to be as univariate time series object - `ts()`.

```
?decompose
?stl
?stats::arima
```

`Split the data into 304 time series?`

```r
RegionStatePurpose <- my_tourism %>%
  select(Region,State,Purpose) %>%
  distinct()

tssets <- RegionStatePurpose %>%
  pmap(function(Region,State,Purpose) my_tourism %>%
         filter(Region == !!Region,State == !!State, Purpose == !!Purpose) %>%
         pull(Trips) %>% ts(start = c(1998,1), frequency = 4))
str(tssets)
```

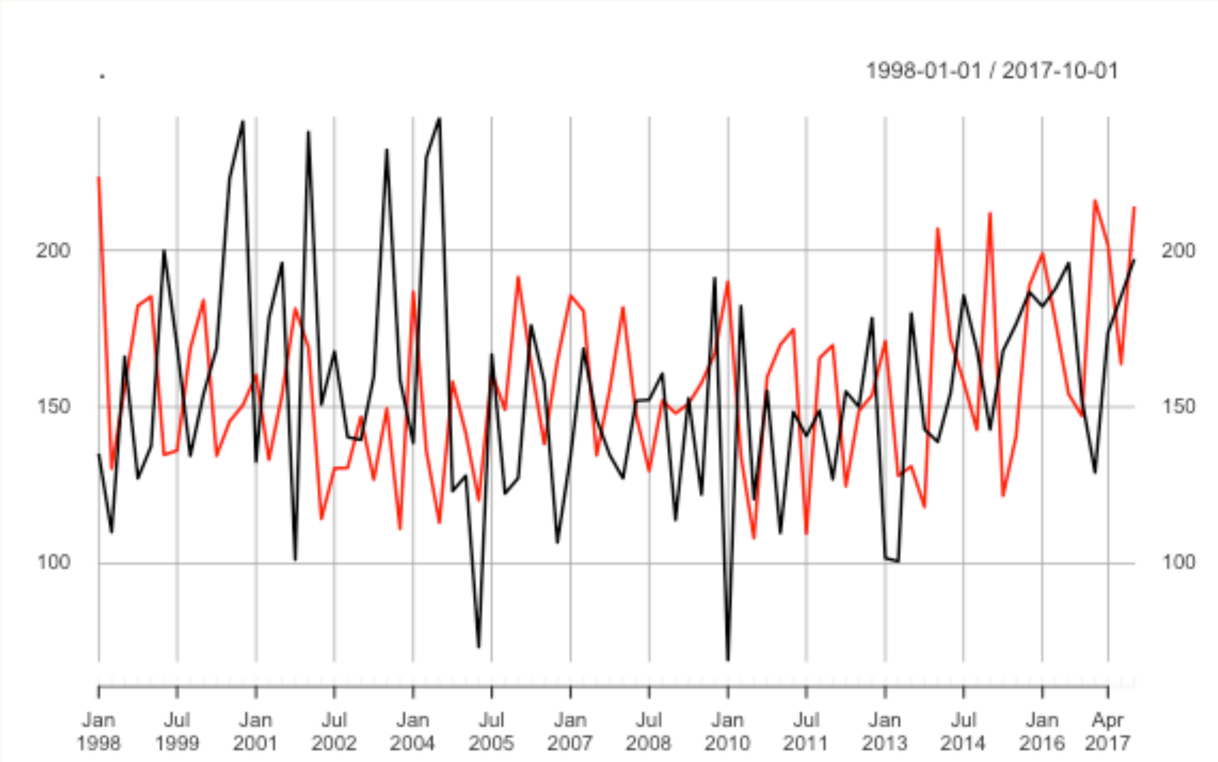
```
## List of 304
##  $ : Time-Series [1:80] from 1998 to 2018: 135 110 166 127 137 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 224 130 156 182 185 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 58.4 39.5 38.4 33.8 25.9 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 242 170 232 181 200 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 0 0 7.42 0 0.67 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 6.81 13.67 12.92 18.14 10.91 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 0 4.642 2.555 0.774 0 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 2.99 7.75 3.6 8.35 2.2 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 7.54 3.36 21.78 3.98 18.41 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 8.15 34.66 76.54 27.22 12.5 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 3.1 0 5.81 1.56 7.82 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 1.43 18.34 6.79 8.11 9.64 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 26.2 27.2 33.7 31 28 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 82.2 104.5 118.4 111.4 130.2 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 0.83 3.439 2.59 2.831 0.771 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 23.3 37.4 19.3 61.8 39.9 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 50.7 77.8 75.5 81.5 43.6 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 61.9 39.6 85.5 74 48.3 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 4.83 1.03 14.14 2.42 2.7 ...
##  $ : Time-Series [1:80] from 1998 to 2018: 44.3 46.6 31.8 40.6 45.6 ...
```

What about xts?

```
tourism_xts <- my_tourism %>%
  slice(1:160) %>%
  mutate(Key = rep(c("Business","Holiday"), each = 80)) %>%
  select(Quarter,Trips, Key) %>%
  pivot_wider(names_from = Key, values_from = Trips)

tourism_xts %>% head()
```

```
## # A tibble: 6 x 3
##   Quarter    Business Holiday
##   <chr>         <dbl>   <dbl>
## 1 1998-01-01     135.    224.
## 2 1998-04-01     110.    130.
## 3 1998-07-01     166.    156.
## 4 1998-10-01     127.    182.
## 5 1999-01-01     137.    185.
## 6 1999-04-01     200.    135.
```

```r
library(xts)
tourism_xts %>%
  select(-Quarter) %>%
  xts(order.by = ymd(tourism_xts$Quarter)) %>%
  plot()
```



It will be a `very wide` table for 304 timeseries.

What do we do? We use `tsibble!`

Modeling Process - Time series data.

1. `Inspect & Explore data.`
2. Determine transformation process (if required).
3. Choose training, validation &/ test data.
4. Fit models.
5. Inspect model metrics.

Components of tsibble data type

1. `Index`
2. `Key`
3. Measured

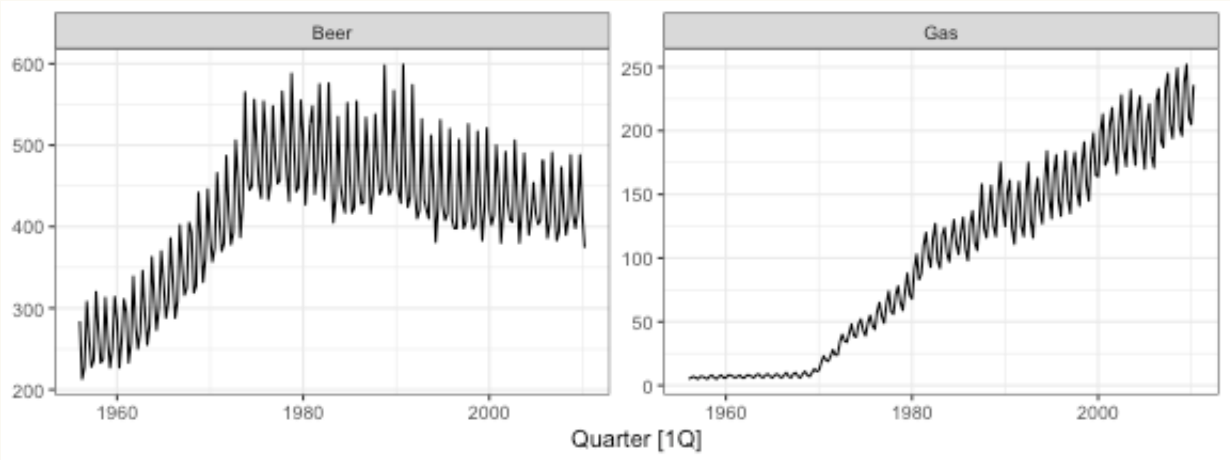
```
my_tourism %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(key = c(Region,State,Purpose),
             index = Quarter) -> tourism_tsbl

tourism_tsbl
```

```
## # 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 South Australia Business  135.
##  2 1998 Q2 Adelaide South Australia Business  110.
##  3 1998 Q3 Adelaide South Australia Business  166.
##  4 1998 Q4 Adelaide South Australia Business  127.
##  5 1999 Q1 Adelaide South Australia Business  137.
##  6 1999 Q2 Adelaide South Australia Business  200.
##  7 1999 Q3 Adelaide South Australia Business  169.
##  8 1999 Q4 Adelaide South Australia Business  134.
##  9 2000 Q1 Adelaide South Australia Business  154.
## 10 2000 Q2 Adelaide South Australia Business  169.
## # … with 24,310 more rows
```

autoplot

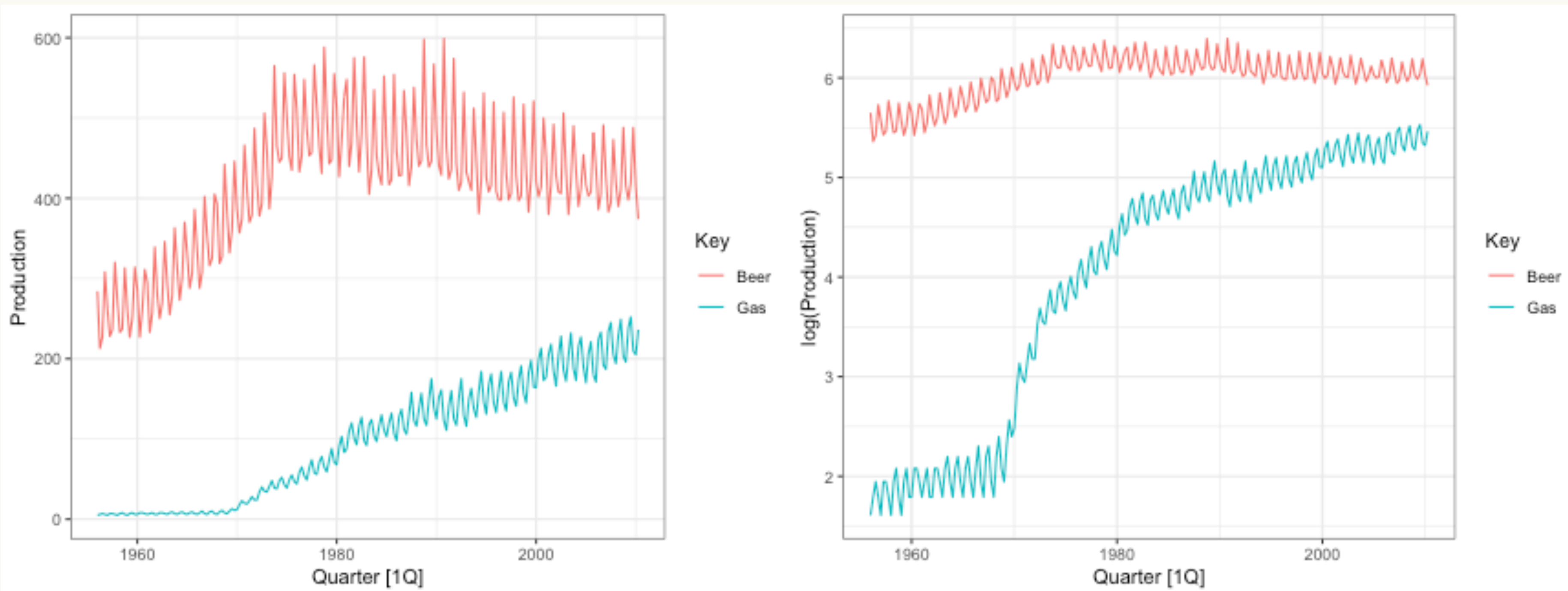
```r
tourism_tsbl %>%
  autoplot(Trips) + guides(color = "none") + theme_bw()
```


```r
tsibbledata::aus_production %>%
  autoplot(vars(Beer,Gas)) + theme_bw()
```



```r
tsibbledata::aus_production %>%
  select(Beer,Gas) %>%
  pivot_longer(-Quarter,names_to = "Key",values_to = "Production") %>%
  head(3)
```

```r
tsibbledata::aus_production %>%
  select(Beer,Gas) %>%
  pivot_longer(-Quarter,names_to = "Key",values_to = "Production") %>%
  as_tsibble(key = Key, index = Quarter) %>%
  autoplot() + theme_bw()  -> p1

tsibbledata::aus_production %>%
  select(Beer,Gas) %>%
  pivot_longer(-Quarter,names_to = "Key",values_to = "Production") %>%
  as_tsibble(key = Key, index = Quarter) %>%
  autoplot(log(Production)) + theme_bw()  -> p2

gridExtra::grid.arrange(p1,p2,ncol = 2)
```



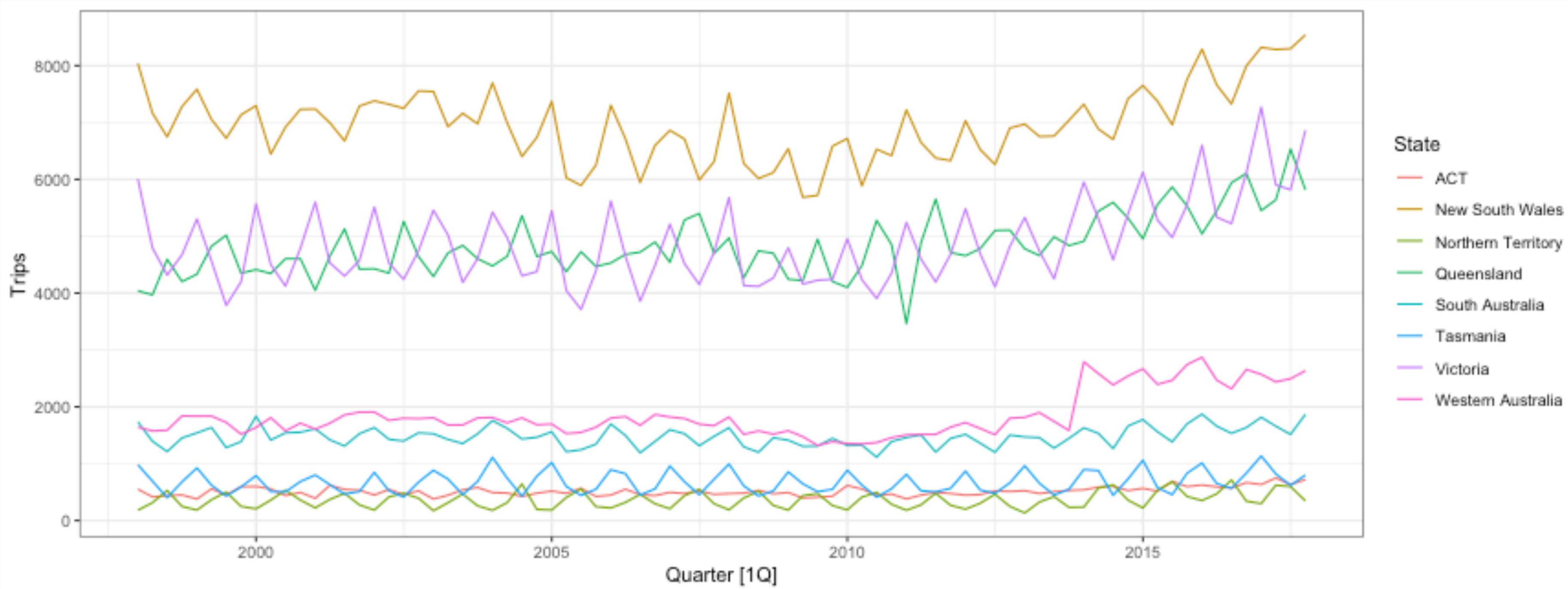
`group_by()` - making you lazy

You dont have to include `index` in the group

```
tourism_tsbl %>%
  group_by(State,Purpose) %>%
  summarise(Trips = sum(Trips)) %>%
  head()
```

```
## # A tibble: 6 x 4
## # Groups:   State [1]
##   State Purpose  Quarter Trips
##   <chr> <chr>      <qtr> <dbl>
## 1 ACT   Business 1998 Q1 150.
## 2 ACT   Business 1998 Q2  99.9
## 3 ACT   Business 1998 Q3 130.
## 4 ACT   Business 1998 Q4 102.
## 5 ACT   Business 1999 Q1  95.5
## 6 ACT   Business 1999 Q2 229.
```

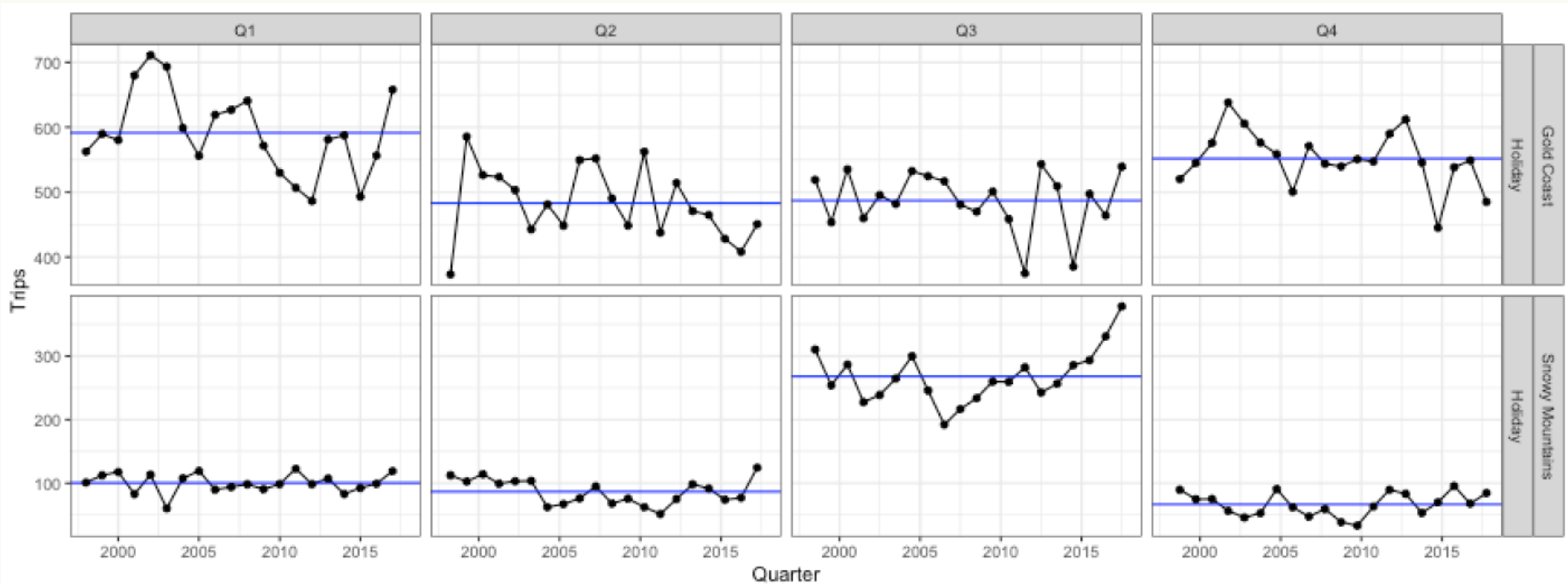
```
tourism_tsbl %>%
  group_by(State) %>%
  summarise(Trips = sum(Trips)) %>%
  autoplot(Trips) +
  theme_bw()
```



Exploring the series

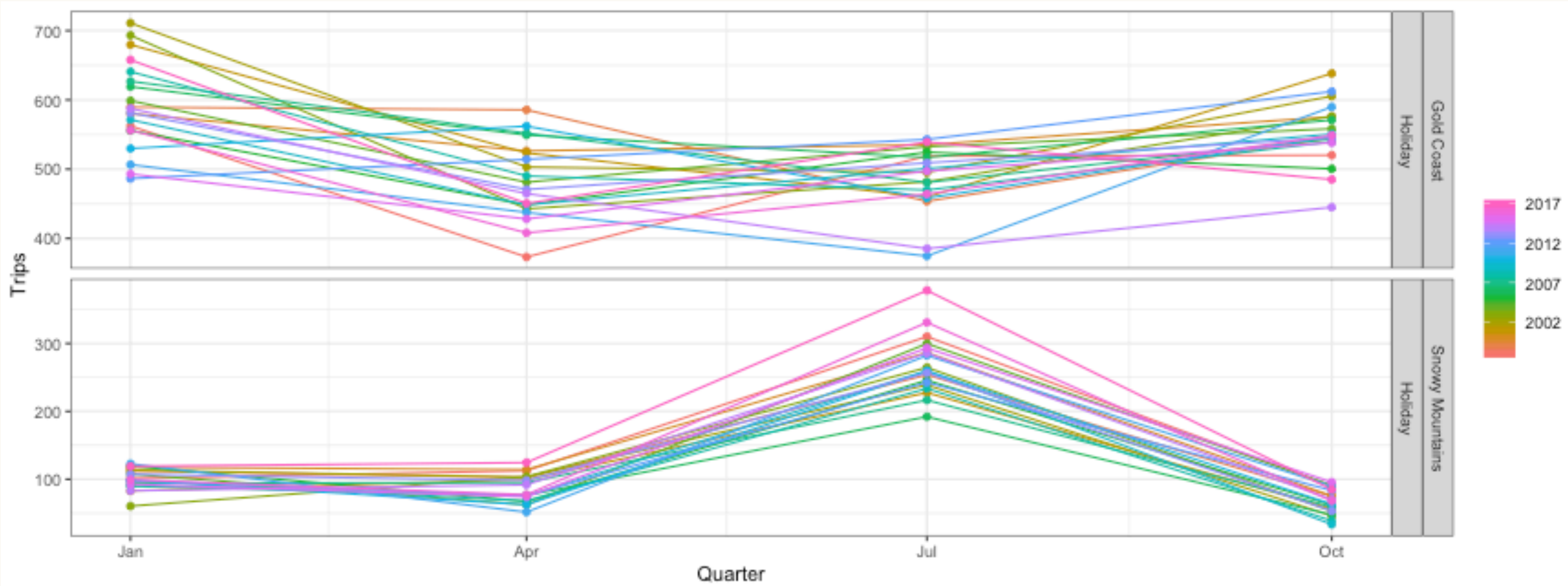
```
tourism_tsbl %>%
  select(-State) %>%
  filter(Region %in% c('Gold Coast','Snowy Mountains'), Purpose == 'Holiday') %>%
  gg_subseries() +
  theme_bw() +
  geom_point()
```

```
## Plot variable not specified, automatically selected `y = Trips`
```



Exploring the series

```
tourism_tsbl %>%
  select(-State) %>%
  filter(Region %in% c('Gold Coast','Snowy Mountains'), Purpose == 'Holiday') %>%
  gg_season(Trips) +
  theme_bw() +
  geom_point()
```

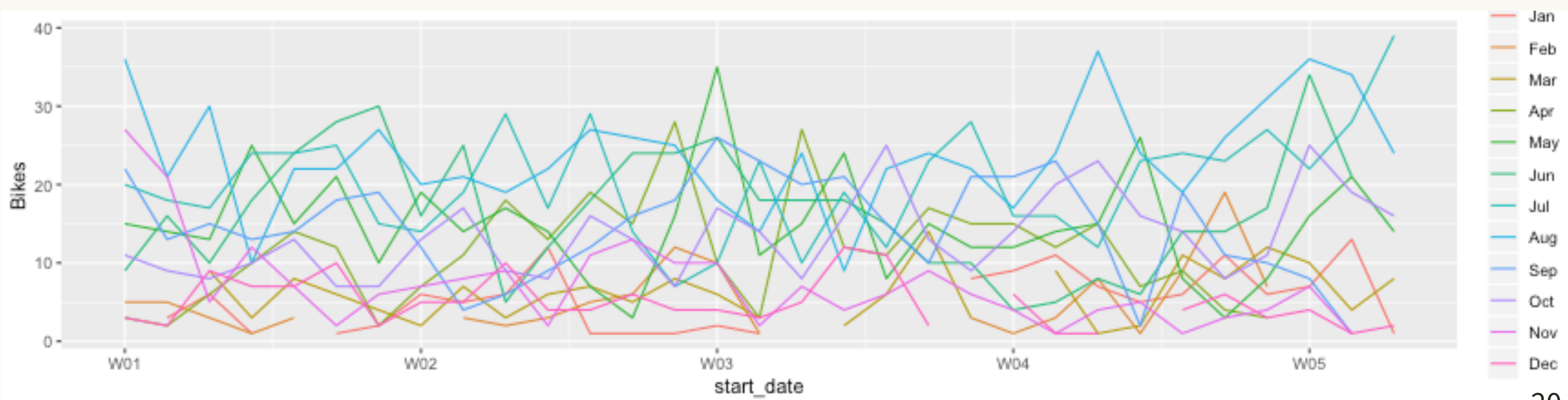


Season - none or many

```
tsibbledata::nyc_bikes %>%
  head(5)
```

```
## # A tsibble: 5 x 12 [!] <America/New_York>
## # Key:       bike_id [1]
##   bike_id start_time          stop_time           start_station start_lat
##   <fct>   <dttm>              <dttm>              <fct>             <dbl>
## 1 26301   2018-02-26 19:11:03 2018-02-26 19:15:40 3186               40.7
## 2 26301   2018-02-27 07:52:49 2018-02-27 07:58:13 3203               40.7
## 3 26301   2018-02-27 12:03:27 2018-02-27 12:04:54 3202               40.7
## 4 26301   2018-02-27 13:53:51 2018-02-27 14:21:04 3638               40.7
## 5 26301   2018-02-27 14:30:42 2018-02-27 14:33:11 3638               40.7
## # … with 7 more variables: start_long <dbl>, end_station <fct>, end_lat <dbl>,
## #   end_long <dbl>, type <fct>, birth_year <dbl>, gender <fct>
```

```
tsibbledata::nyc_bikes %>%
  as_tibble() %>%
  mutate(start_date = as.Date(start_time)) %>%
  group_by(start_date) %>%
  summarise(Bikes = n()) %>%
  as_tsibble(index = start_date) %>%
  fill_gaps() %>%
  gg_season(Bikes,period = "month")
```

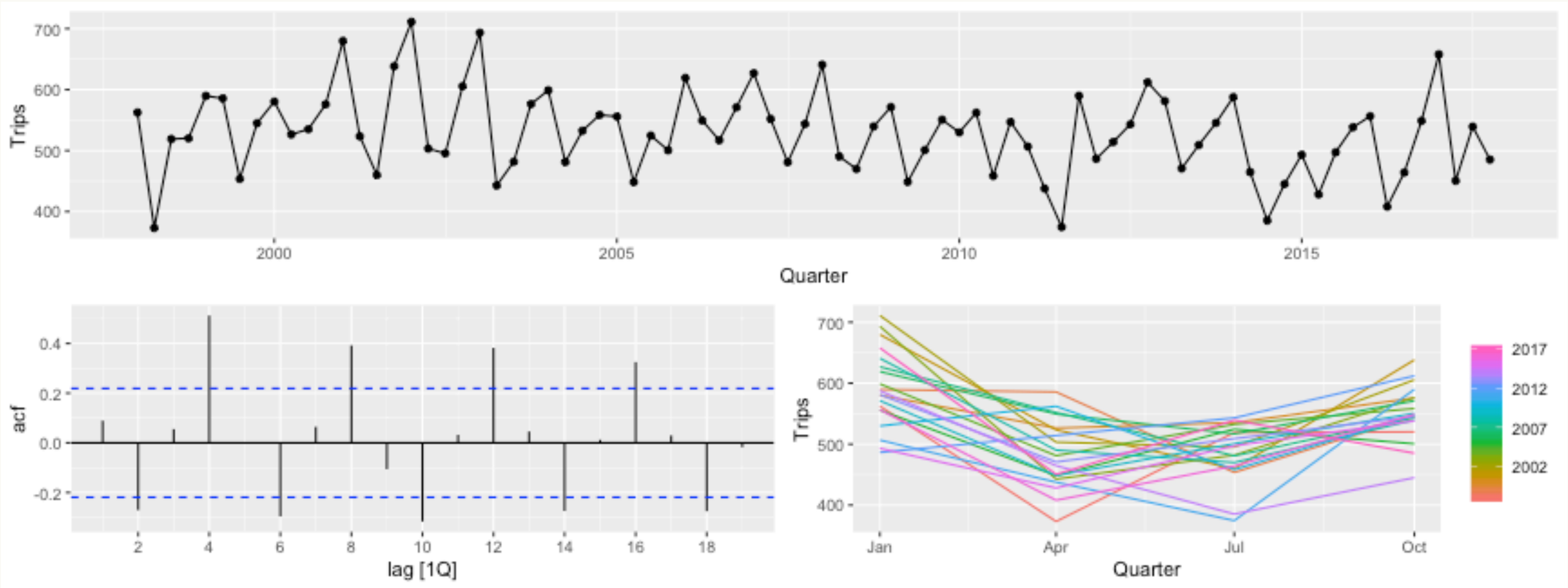


`fill_gaps()` can be dangerous

[read more][./appendix/fppvsts.html]

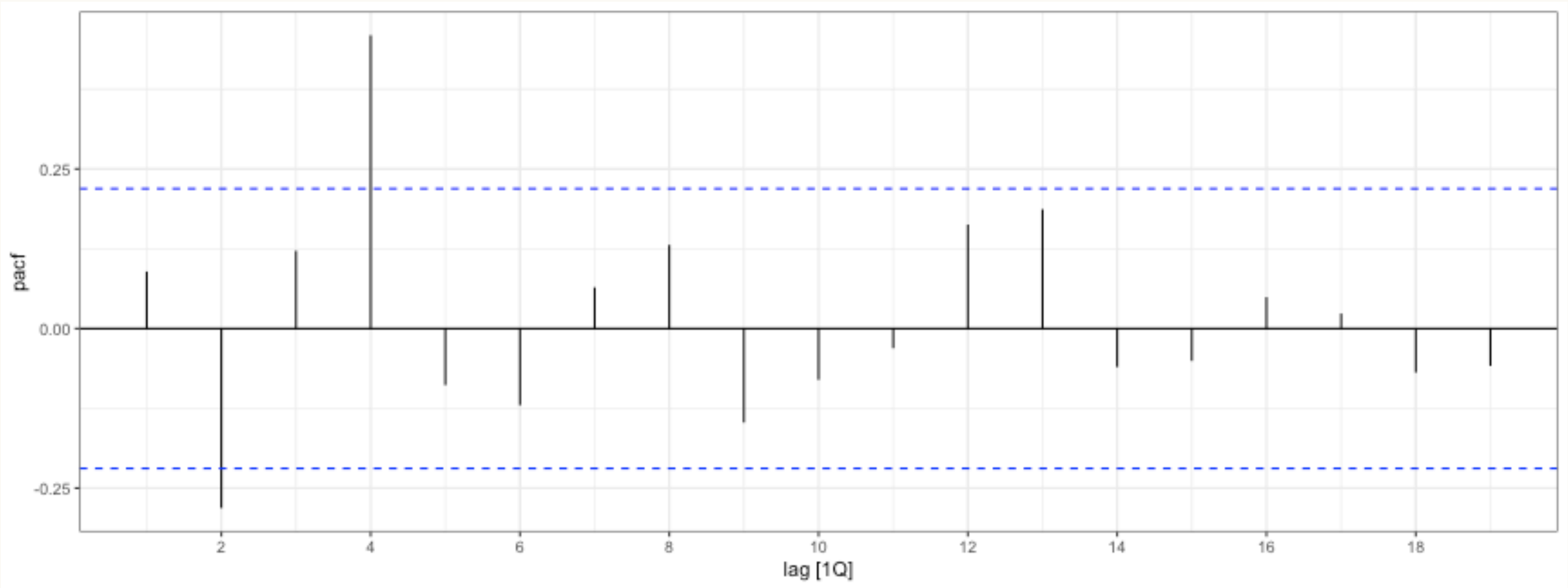
Autocorrelations

```
tourism_tsbl %>%
  select(-State) %>%
  filter(Region =='Gold Coast', Purpose == 'Holiday') %>%
  gg_tsdisplay(Trips)
```



ACF & PACF

```
tourism_tsbl %>%
  select(-State) %>%
  filter(Region =='Gold Coast', Purpose == 'Holiday') %>%
  PACF(Trips) %>% # Try ACF
  autoplot() + theme_bw()
```



gg_lag()

```
tourism_tsbl %>%
  select(-State) %>%
  filter(Region =='Gold Coast', Purpose == 'Holiday') %>%
  gg_lag(Trips, geom = "point") +
  theme_bw()
```

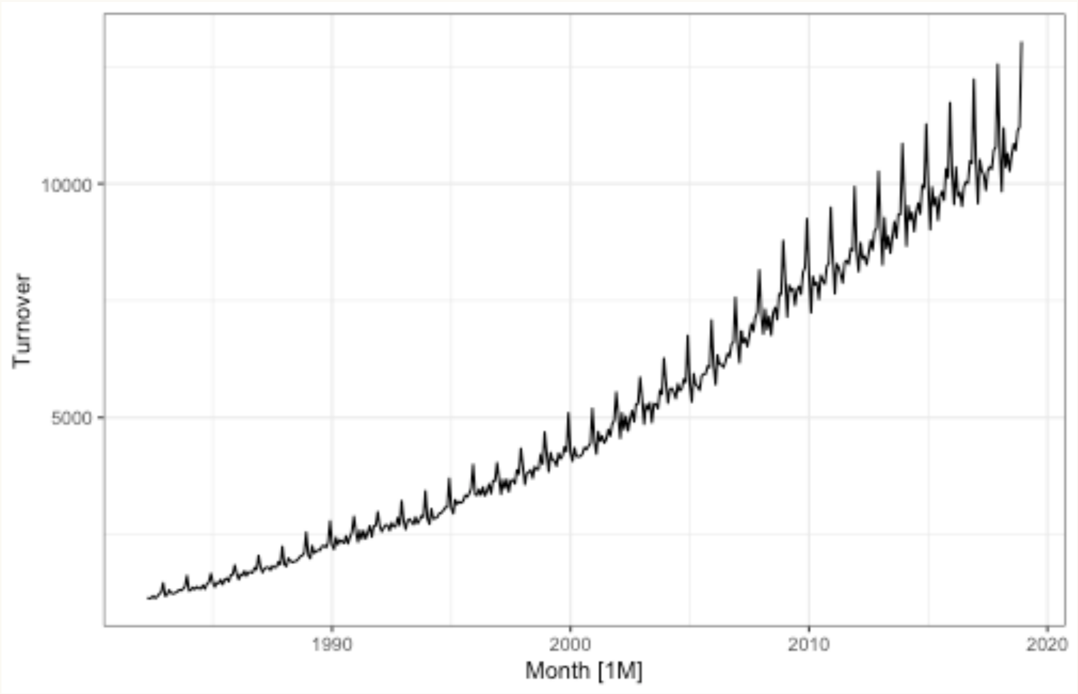
Modeling Process

1. Inspect & Explore data.
2. `Determine transformation process (if required).`
3. Choose training, validation &/ test data
4. Fit models.
5. Inspect model metrics.

Transformations

Variance stabilization

```
food <- aus_retail %>%
  filter(Industry == "Food retailing") %>%
  summarise(Turnover = sum(Turnover))
food %>%
  autoplot(Turnover) + theme_bw()
```

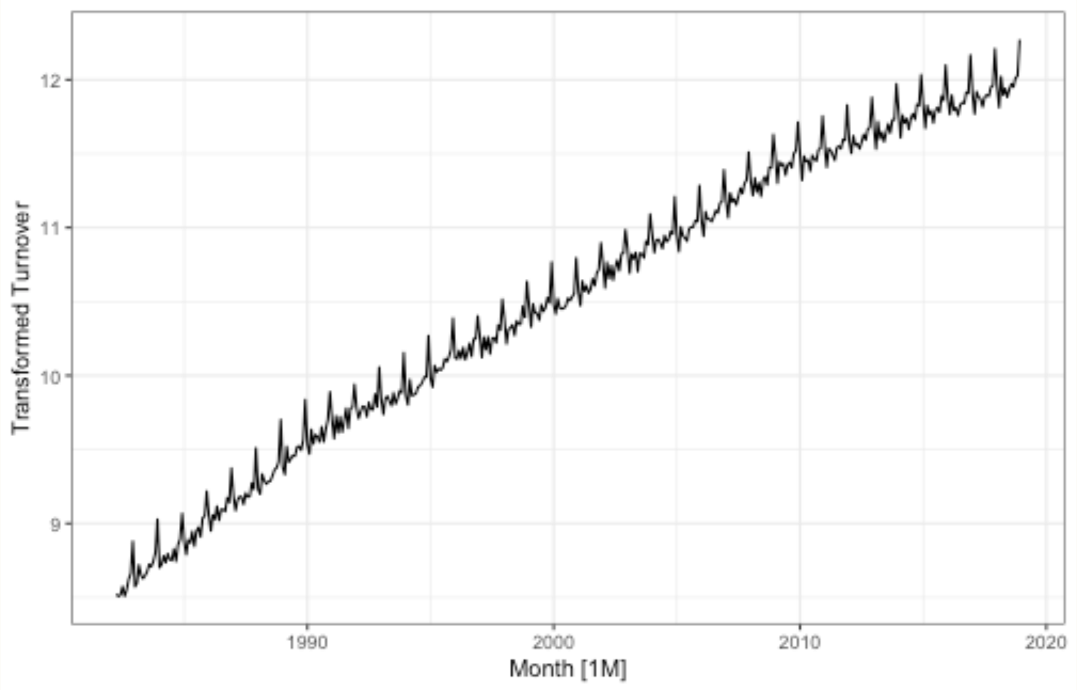


Varaince stabilization

```r
BoxCoxLambda <- food %>%
  features(Turnover, features = guerrero) %>%  #?features_by_tag
  pull(lambda_guerrero)

BoxCoxLambda
```

```
## [1] 0.05241123
```

```r
food %>%
  autoplot(box_cox(Turnover,BoxCoxLambda)) +
  labs( y = "Transformed Turnover") +
  theme_bw()
```

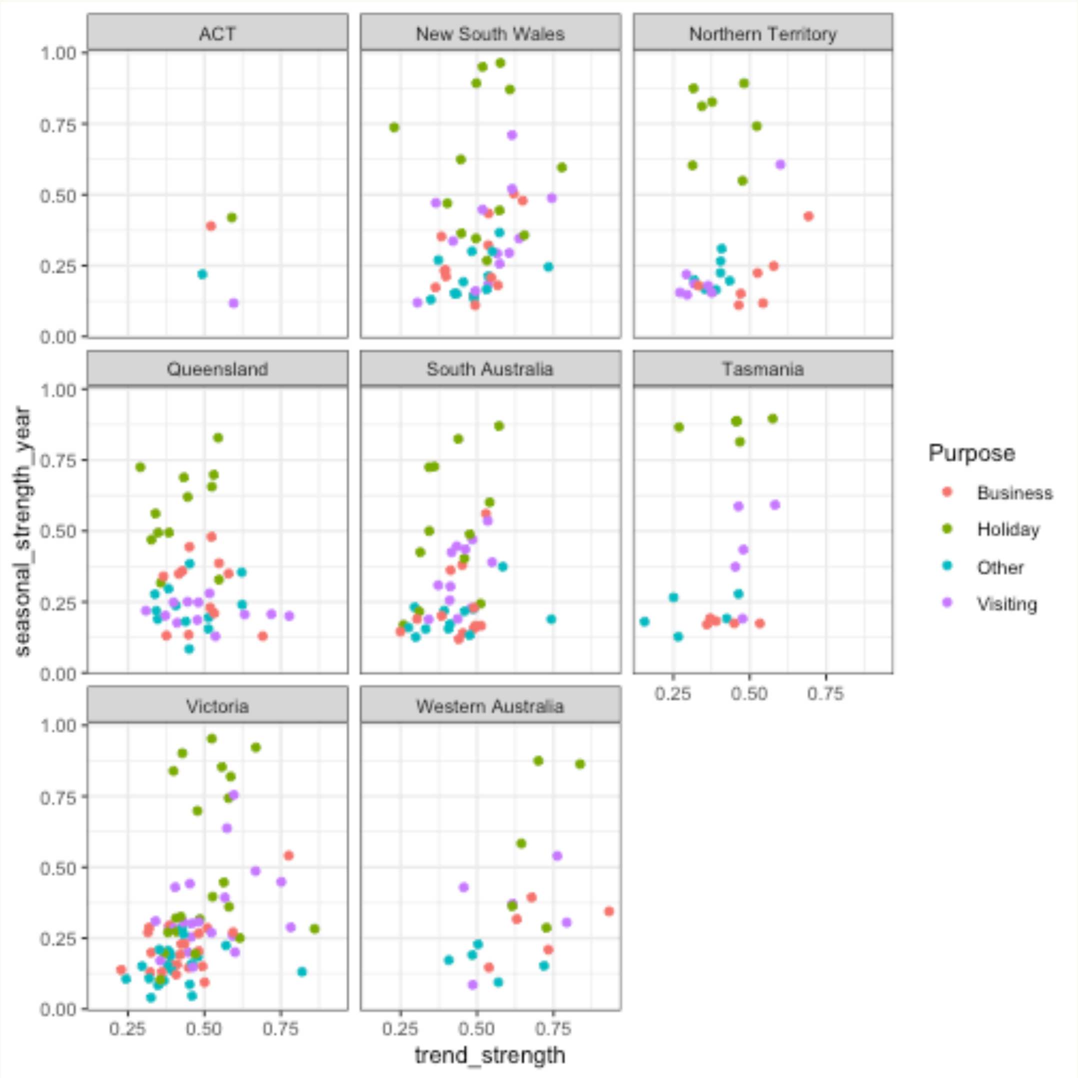


Feature Extraction

```
tourism_tsbl %>%
  features(Trips,feat_stl) %>%
  head()
```

```
## # A tibble: 6 x 12
##   Region State Purpose trend_strength seasonal_streng… seasonal_peak_y…
##   <chr>  <chr> <chr>            <dbl>            <dbl>            <dbl>
## 1 Adela… Sout… Busine…          0.451            0.380                3
## 2 Adela… Sout… Holiday          0.541            0.601                1
## 3 Adela… Sout… Other            0.743            0.189                2
## 4 Adela… Sout… Visiti…          0.433            0.446                1
## 5 Adela… Sout… Busine…          0.453            0.140                3
## 6 Adela… Sout… Holiday          0.512            0.244                2
## # … with 6 more variables: seasonal_trough_year <dbl>, spikiness <dbl>,
## #   linearity <dbl>, curvature <dbl>, stl_e_acf1 <dbl>, stl_e_acf10 <dbl>
```

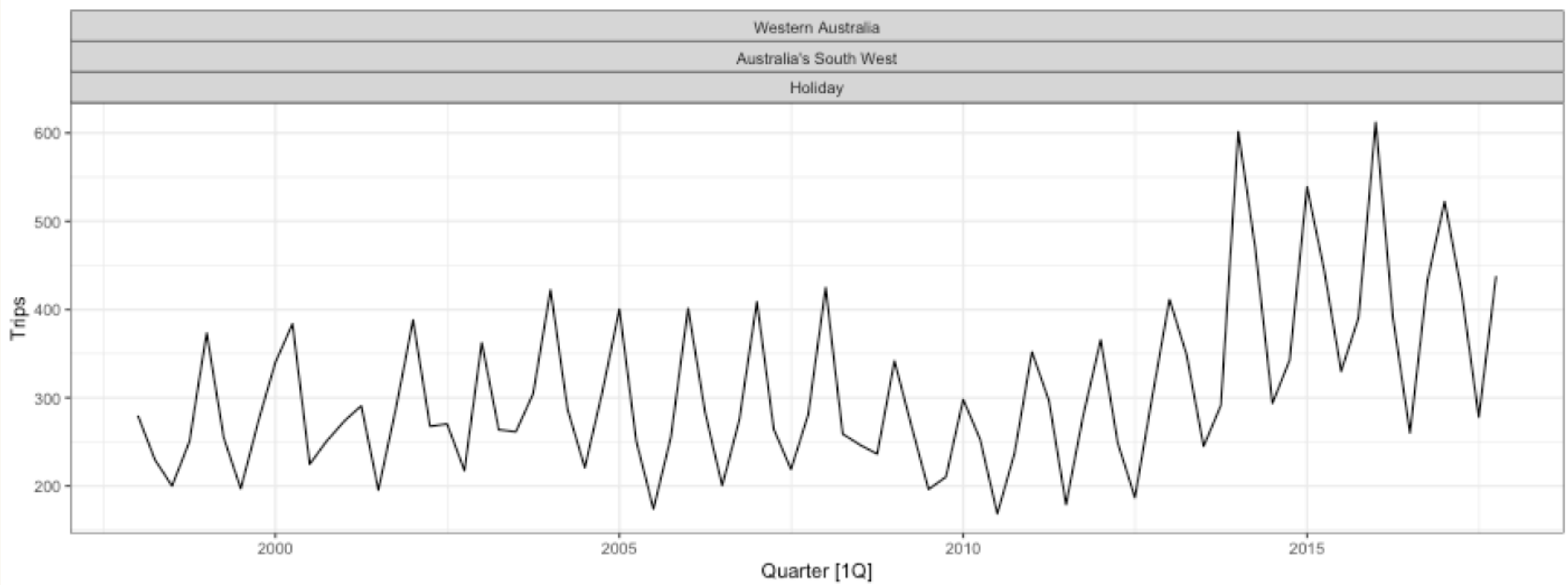
```
tourism_tsbl %>%
  features(Trips,feat_stl) %>%
  ggplot(aes(x=trend_strength, y = seasonal_strength_year, col = Purpose)) +
  facet_wrap(~State) +
  geom_point() + theme_bw()
```



Find the ost seasonal and trending series

```
tourism_tsbl %>%
  features(Trips,feat_stl) %>%
  mutate(trendSeasonScore = trend_strength * seasonal_strength_year) %>%
  filter(trendSeasonScore == max(trendSeasonScore)) -> TrendingSeasonal

tourism_tsbl %>%
  right_join(TrendingSeasonal) %>%
  autoplot(Trips) + facet_wrap(~State + Region + Purpose) + theme_bw()
```

```
## Joining, by = c("Region", "State", "Purpose")
```



Modeling Process

1. Inspect & Explore data.
2. Determine transformation process (if required).
3. `Choose training, validation &/ test data`
4. `Fit models.`
5. Inspect model metrics.

Fit model

```
# benchmark meodels
tourism_tsbl %>%
  filter(Region == "Australia's South West", Purpose == "Holiday") %>%
  filter(year(Quarter) < 2015) %>%  # get training data
  model(
    snaive = SNAIVE(Trips),
    arima = ARIMA(Trips),
    ets = ETS(Trips)
  ) -> fit_benchmark

fit_benchmark
```

```
## # A mable: 1 x 6
## # Key:     Region, State, Purpose [1]
##   Region          State         Purpose snaive   arima                     ets
##   <chr>           <chr>         <chr>   <model> <model>                    <model>
## 1 Australia's Sou… Western Aust… Holiday <SNAIV… <ARIMA(1,0,0)(0,1,2)… <ETS(M,N…
```

Model report & forecast

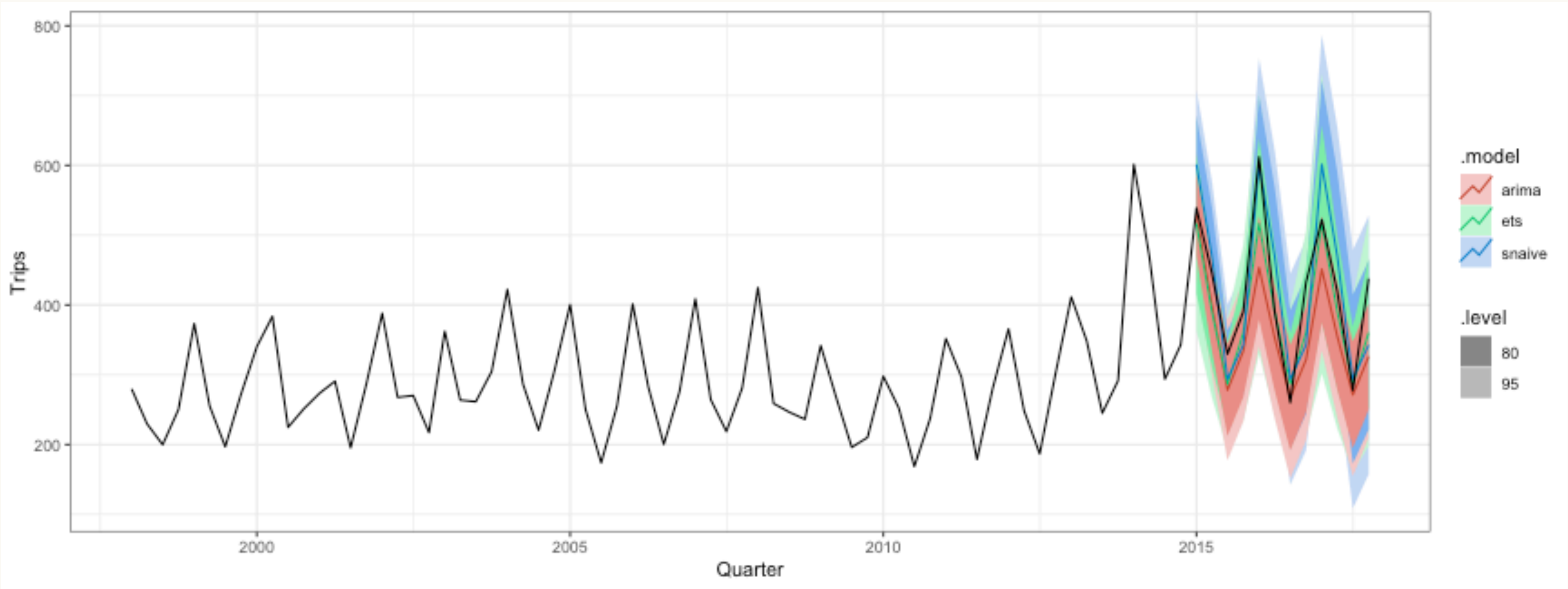
```
fit_benchmark %>%
  select(arima) %>%
  report()
```

```
## Series: Trips
## Model: ARIMA(1,0,0)(0,1,2)[4] w/ drift
##
## Coefficients:
##          ar1     sma1     sma2  constant
##       0.4370  -0.4740  -0.3989    3.0155
## s.e.  0.1156   0.1954   0.1637    1.6357
##
## sigma^2 estimated as 2124:  log likelihood=-336.55
## AIC=683.09   AICc=684.13   BIC=693.89
```

smart fable!

Fable is smart to figure the filtering and test data!

```
fit_benchmark %>%

  fabletools::forecast(h = "3 years") -> fcts

fcts %>%
  autoplot(tourism_tsbl) +
  theme_bw()
```



Modeling Process

1. Inspect & Explore data.
2. Determine transformation process (if required).
3. Choose training, validation &/ test data`
4. Fit models.`
5. `Inspect model metrics.`

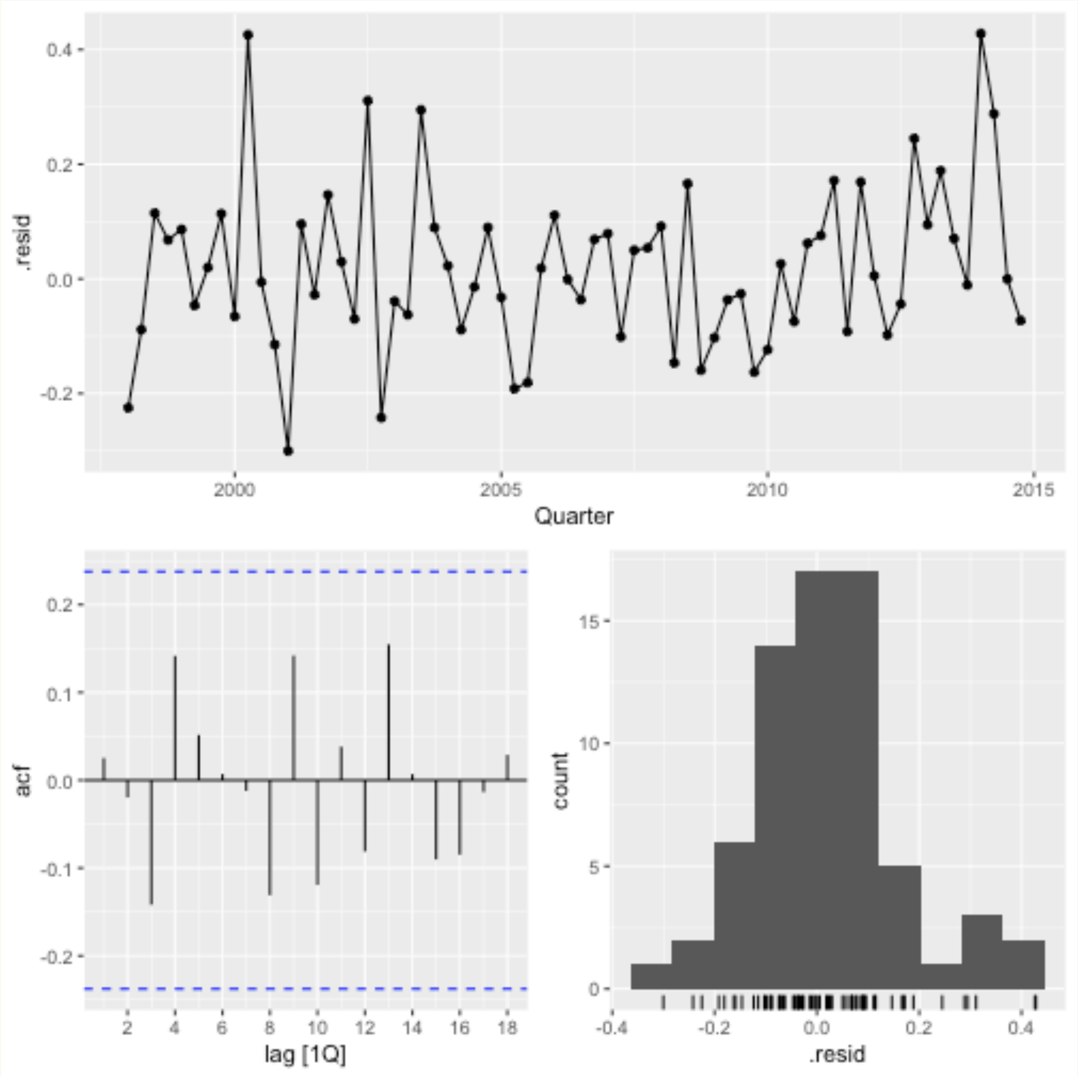
OOS error

```
fcts %>%
  accuracy(tourism_tsbl)
```

```
## # A tibble: 3 x 12
##   .model Region  State Purpose .type    ME  RMSE   MAE   MPE  MAPE  MASE    ACF1
##   <chr>  <chr>   <chr> <chr>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 arima  Austra… West… Holiday Test  59.8   75.7  61.0 13.1  13.6  1.48  -0.233
## 2 ets    Austra… West… Holiday Test  32.9   48.5  38.8  6.83  9.04 0.940 -0.349
## 3 snaive Austra… West… Holiday Test  -5.44  58.9  52.0 -1.30 12.5  1.26  -0.0663
```

Inspect model fit

```
fit_benchmark %>%
  select(ets) %>%
  gg_tsresiduals()
```



```r
augment(fit_benchmark) %>%
  features(.resid,ljung_box)
```

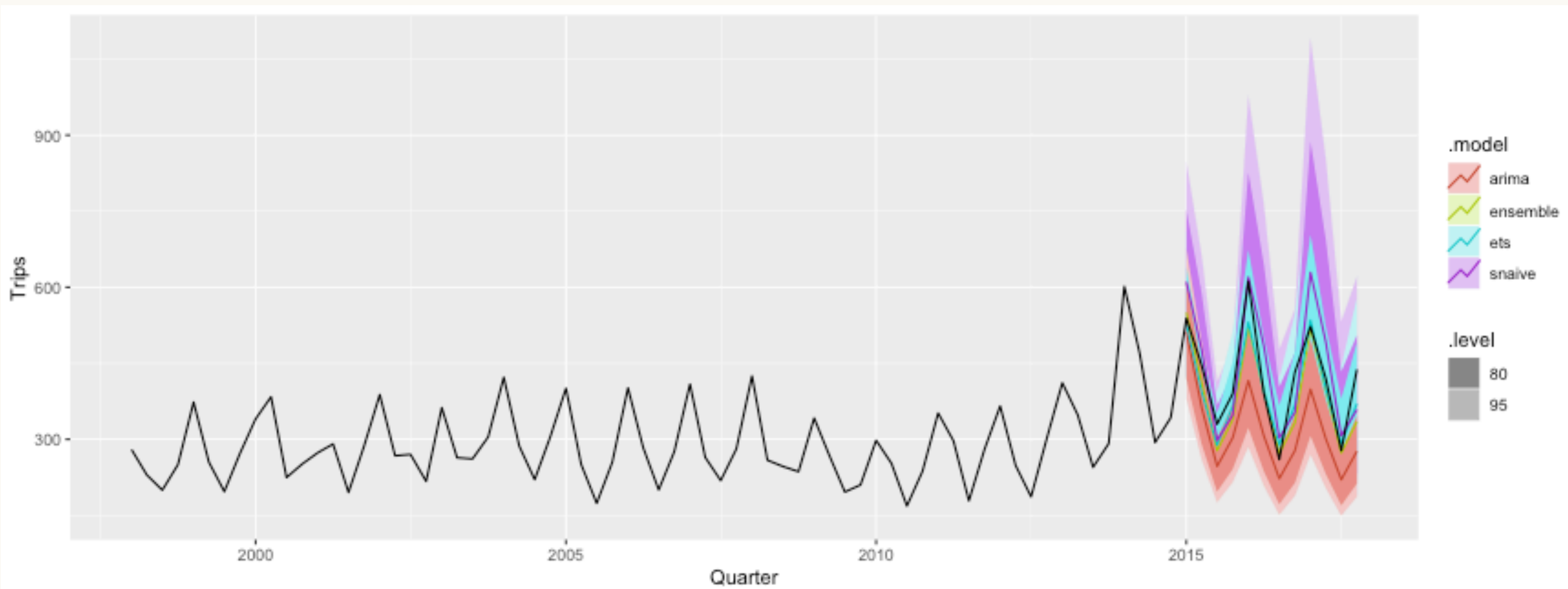
```
## # A tibble: 3 x 6
##   Region                  State             Purpose .model lb_stat lb_pvalue
##   <chr>                   <chr>             <chr>   <chr>    <dbl>     <dbl>
## 1 Australia's South West Western Australia Holiday arima   0.115    0.735
## 2 Australia's South West Western Australia Holiday ets     0.0454   0.831
## 3 Australia's South West Western Australia Holiday snaive  6.98     0.00825
```

Now the fun stuff!

Transformation of the data are handled

There is a way to `ensemble` models here

```
tourism_tsbl %>%
  filter(Region == "Australia's South West", Purpose == "Holiday") %>%
  filter(year(Quarter) < 2015) %>%  # get training data
  model(
    snaive = SNAIVE(log(Trips)),
    arima = ARIMA(log(Trips)),
    ets = ETS(log(Trips))
  ) %>%
  mutate(ensemble = (snaive + arima + ets)/3) -> fit_benchmark

fit_benchmark %>%
  fabletools::forecast(h = "3 years") %>%
  autoplot(tourism_tsbl)
```



Ensemble accuracy

```
fit_benchmark %>%
  fabletools::forecast(h = "3 years") %>%
  accuracy(tourism_tsbl)
```

```
## # A tibble: 4 x 12
##   .model  Region State Purpose .type    ME  RMSE   MAE   MPE  MAPE  MASE    ACF1
##   <chr>   <chr>  <chr> <chr>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 arima   Austr… West… Holiday Test  101.  112.  101.  23.7  23.7  2.45  -0.176
## 2 ensemb… Austr… West… Holiday Test   35.9  54.7  40.6  8.12  9.42 0.983 -0.273
## 3 ets     Austr… West… Holiday Test   25.2  43.0  35.5  5.09  8.46 0.859 -0.340
## 4 snaive  Austr… West… Holiday Test  -18.5  64.2  57.3 -4.43 13.9  1.39  -0.0445
```

Slap a model on 304 time series in tsibble

```r
tourism_tsbl %>%
  #filter(Region == "Australia's South West", Purpose == "Holiday") %>%
  filter(year(Quarter) < 2015) %>%  # get training data
  model(
    snaive = SNAIVE(Trips),
    arima = ARIMA(Trips),
    ets = ETS(Trips)
  ) -> fit_benchmark_All
```

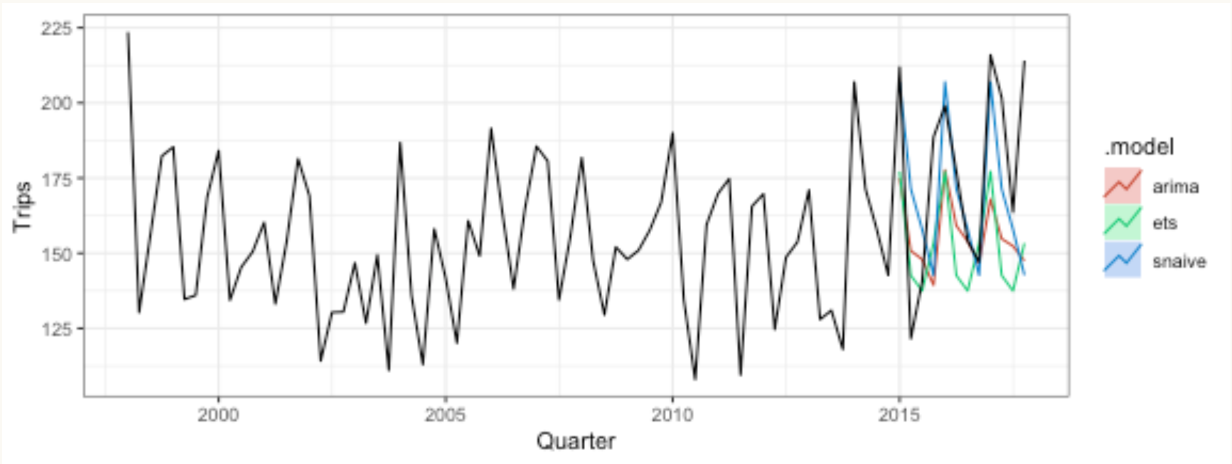
.code80[

```r
load("fit_benchmark_All.RData")
fit_benchmark_All
```

```
## # A mable: 304 x 6
## # Key:     Region, State, Purpose [304]
##    Region       State          Purpose snaive  arima                        ets
##    <chr>        <chr>          <chr>   <model> <model>                      <model>
##  1 Adelaide     South Austral… Busine… <SNAIV… <ARIMA(0,0,0)(1,0,1)[4… <ETS(M,N…
##  2 Adelaide     South Austral… Holiday <SNAIV… <ARIMA(0,0,0)(2,0,0)[4… <ETS(M,N…
##  3 Adelaide     South Austral… Other   <SNAIV… <ARIMA(0,1,1) w/ drift> <ETS(M,A…
##  4 Adelaide     South Austral… Visiti… <SNAIV… <ARIMA(0,0,0)(1,0,1)[4… <ETS(A,N…
##  5 Adelaide Hi… South Austral… Busine… <SNAIV… <ARIMA(1,0,0) w/ mean>  <ETS(A,N…
##  6 Adelaide Hi… South Austral… Holiday <SNAIV… <ARIMA(0,0,0) w/ mean>  <ETS(A,N…
##  7 Adelaide Hi… South Austral… Other   <SNAIV… <ARIMA(0,0,1)(1,0,0)[4… <ETS(A,N…
##  8 Adelaide Hi… South Austral… Visiti… <SNAIV… <ARIMA(0,0,0) w/ mean>  <ETS(M,A…
##  9 Alice Sprin… Northern Terr… Busine… <SNAIV… <ARIMA(0,0,0) w/ mean>  <ETS(A,N…
## 10 Alice Sprin… Northern Terr… Holiday <SNAIV… <ARIMA(0,0,0)(0,1,2)[4… <ETS(M,N…
## # … with 294 more rows
```

Get forecasts for each series on demand

```
fit_benchmark_All %>%
  filter(Region == 'Adelaide',
         Purpose == 'Holiday') %>%
  forecast(h = "3 years") %>%

  autoplot(tourism_tsbl, level = NULL) + theme_bw()
```

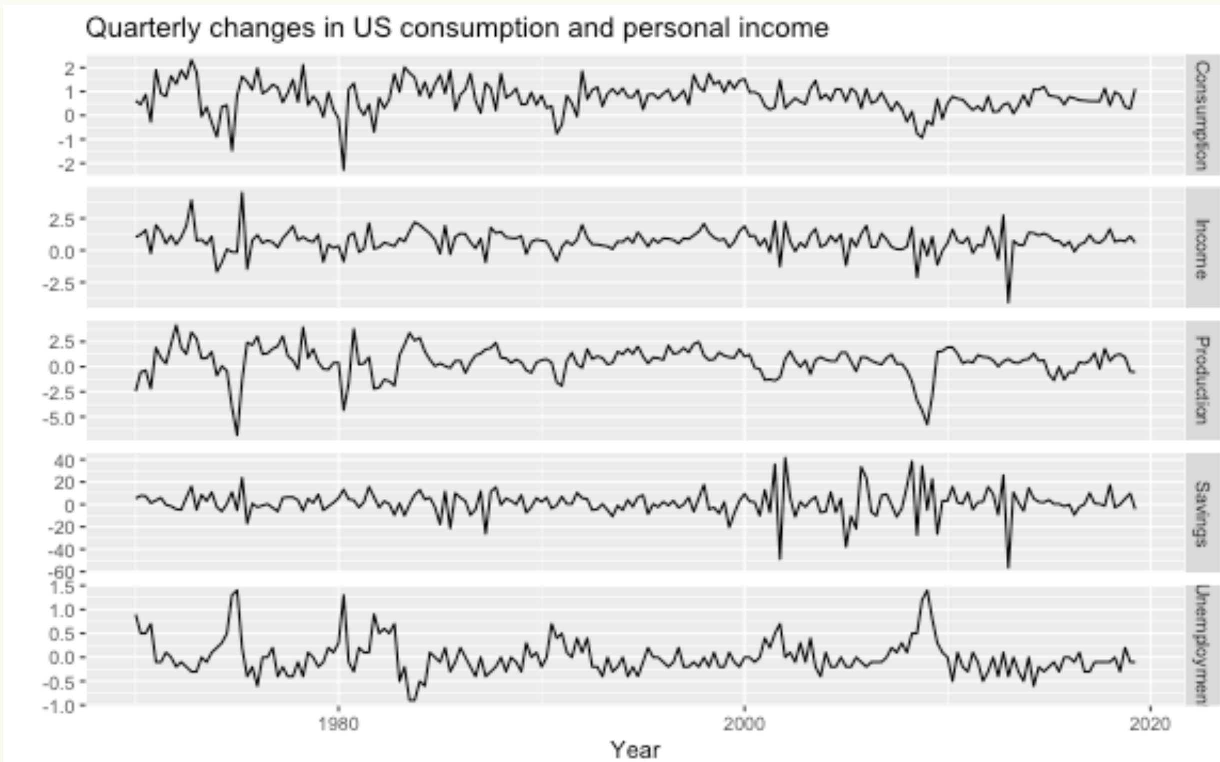



```
fit_benchmark_All %>%
  filter(Region == 'Adelaide',
         Purpose == 'Holiday') %>%
  forecast(h = "3 years") %>%
  accuracy(tourism_tsbl)
```

```
## # A tibble: 3 x 12
##   .model Region State Purpose .type    ME  RMSE   MAE   MPE  MAPE  MASE     ACF1
##   <chr>  <chr>  <chr> <chr>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 arima  Adela… Sout… Holiday Test  21.9   34.8  28.0  9.93  14.8  1.31 -0.0202
## 2 ets    Adela… Sout… Holiday Test  25.3   34.5  29.9 12.4   16.0  1.40  0.0900
## 3 snaive Adela… Sout… Holiday Test   8.22  30.5  21.4  2.44  12.4  1.00  0.00218
```

External regressors

```
us_change %>%
  pivot_longer(-Quarter, names_to = "variable", values_to = "value") %>%
  ggplot(aes(y = value, x = Quarter, group = variable)) +
  geom_line() + facet_grid(variable ~ ., scales = "free_y") +
  xlab("Year") + ylab("") +
  ggtitle("Quarterly changes in US consumption and personal income")
```

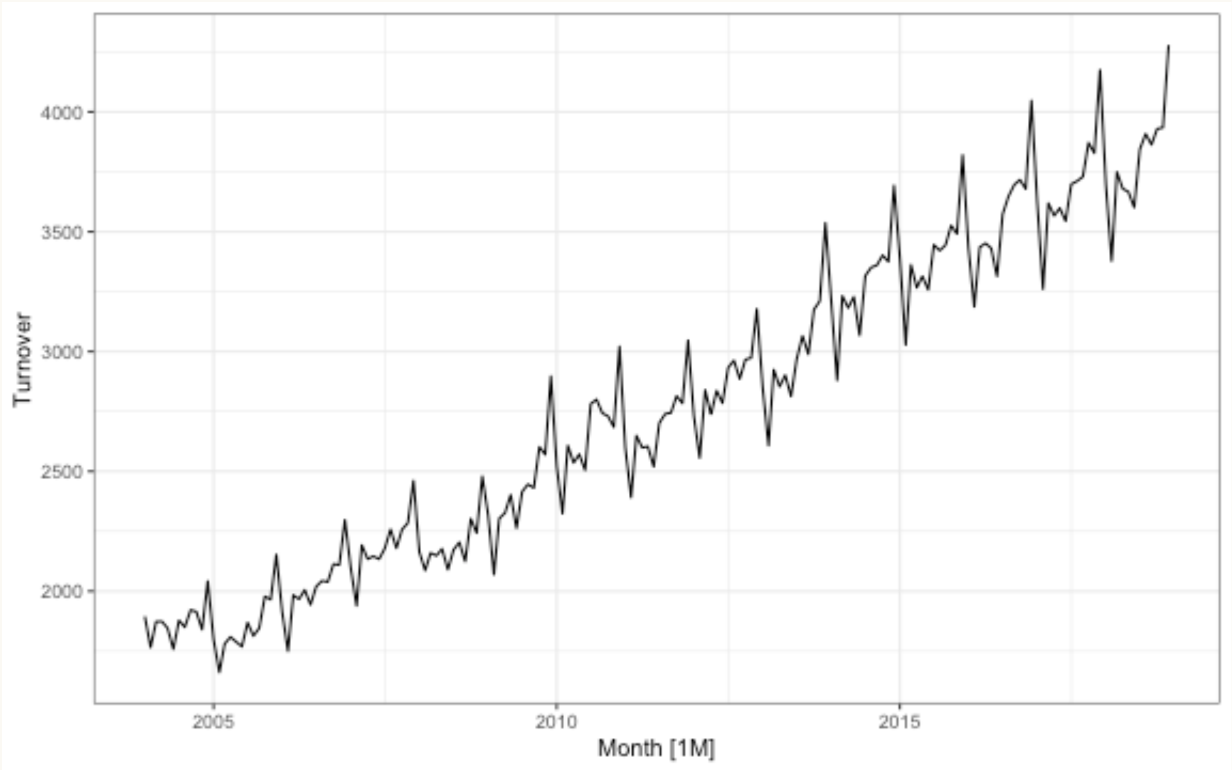


```r
fit <- us_change %>%
  model(regarima = ARIMA(Consumption ~ Income + Production + Savings + Unemployment))
report(fit)
```

```
## Series: Consumption
## Model: LM w/ ARIMA(0,1,2) errors
##
## Coefficients:
##           ma1     ma2  Income  Production  Savings  Unemployment
##       -1.0882  0.1118  0.7472      0.0370  -0.0531       -0.2096
## s.e.   0.0692  0.0676  0.0403      0.0229   0.0029        0.0986
##
## sigma^2 estimated as 0.09588:  log likelihood=-47.13
## AIC=108.27   AICc=108.86   BIC=131.25
```

Harmonic regression

```
aus_cafe <- aus_retail %>%
  filter(Industry == "Cafes, restaurants and takeaway food services",
         year(Month) %in% 2004:2018) %>%
  summarise(Turnover = sum(Turnover))

aus_cafe %>%
  autoplot(Turnover) + theme_bw()
```



Harmonic regression

```
fit <- aus_cafe %>%
  model(
   ` K = 1` = ARIMA(log(Turnover) ~ fourier(K = 1) + PDQ(0, 0, 0)),
    `K = 2 `= ARIMA(log(Turnover) ~ fourier(K = 2) + PDQ(0, 0, 0)),
    `K = 3 `= ARIMA(log(Turnover) ~ fourier(K = 3) + PDQ(0, 0, 0)),
    `K = 4` = ARIMA(log(Turnover) ~ fourier(K = 4) + PDQ(0, 0, 0)),
    `K = 5` = ARIMA(log(Turnover) ~ fourier(K = 5) + PDQ(0, 0, 0)),
    `K = 6` = ARIMA(log(Turnover) ~ fourier(K = 6) + PDQ(0, 0, 0))
  )

glance(fit)
```

```
## # A tibble: 6 x 8
##   .model     sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr>       <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 " K = 1" 0.00175     317. -616. -615. -588. <cpl [2]> <cpl [3]>
## 2 "K = 2 " 0.00107     362. -700. -698. -661. <cpl [5]> <cpl [1]>
## 3 "K = 3 " 0.000761    394. -763. -761. -725. <cpl [3]> <cpl [1]>
## 4 "K = 4"  0.000539    427. -822. -818. -771. <cpl [1]> <cpl [5]>
## 5 "K = 5"  0.000317    474. -919. -917. -875. <cpl [2]> <cpl [0]>
## 6 "K = 6"  0.000316    474. -920. -918. -875. <cpl [0]> <cpl [1]>
```

Harmonic regression

```
fit %>%
 select(`K = 6`) %>%
 report()
```

```
## Series: Turnover
## Model: LM w/ ARIMA(0,1,1) errors
## Transformation: log(.x)
##
## Coefficients:
##           ma1  fourier(K = 6)C1_12  fourier(K = 6)S1_12  fourier(K = 6)C2_12
##       -0.3953               0.0068              -0.0363              -0.0044
## s.e.   0.0619               0.0024               0.0024               0.0016
##       fourier(K = 6)S2_12  fourier(K = 6)C3_12  fourier(K = 6)S3_12
##                   -0.0215              -0.0070              -0.0355
## s.e.               0.0016               0.0014               0.0014
##       fourier(K = 6)C4_12  fourier(K = 6)S4_12  fourier(K = 6)C5_12
##                    0.0035              -0.0202              -0.0069
## s.e.               0.0013               0.0013               0.0013
##       fourier(K = 6)S5_12  fourier(K = 6)C6_12  intercept
##                   -0.0249               0.0014     0.0039
## s.e.               0.0013               0.0009     0.0008
##
## sigma^2 estimated as 0.0003163:  log likelihood=474.03
## AIC=-920.06   AICc=-917.5   BIC=-875.44
```

Harmonic regression

```
fit %>%
  select(`K = 6`) %>%
  forecast(h = "4 years") %>%
  autoplot(aus_cafe) + theme_bw()
```