MONASH University

MONASH BUSINESS SCHOOL

High dimensional time series analysis

tsibble feasts fable

3. Time series features

robjhyndman.com/hdtsa

Outline

1 Decompositions

2 Time series components

3 STL decomposition

feasts

Feature Extraction And Statistics for Time Series

- works with tidy temporal data provided by the tsibble package.
- produces time series features, decompositions, statistical summaries and visualisations.

Outline

1 Decompositions

2 Time series components

3 STL decomposition

Decompositions

The feasts package supports four common time series decomposition methods:

- Classical decomposition
- STL decomposition
- X11 decomposition
- X-13ARIMA-SEATS decomposition

Holidays by state

```r
holidays <- tourism %>%
  filter(Purpose=="Holiday") %>%
  group_by(State) %>%
  summarise(Trips = sum(Trips))
```

```
## # A tsibble: 640 x 3 [1Q]
## # Key:       State [8]
##    State Quarter Trips
##    <chr>   <qtr> <dbl>
##  1 ACT   1998 Q1  196.
##  2 ACT   1998 Q2  127.
##  3 ACT   1998 Q3  111.
##  4 ACT   1998 Q4  170.
##  5 ACT   1999 Q1  108.
##  6 ACT   1999 Q2  125.
##  7 ACT   1999 Q3  178.
##  8 ACT   1999 Q4  218.
##  9 ACT   2000 Q1  158.
## 10 ACT   2000 Q2  155.
```

Decompositions

```r
holidays %>% STL(Trips ~ season(window = "periodic")) %>%
  autoplot()
```

Outline

1 Decompositions

2 Time series components

3 STL decomposition

Time series patterns

Recall

Trend pattern exists when there is a long-term increase or decrease in the data.

Cyclic pattern exists when data exhibit rises and falls that are *not of fixed period* (duration usually of at least 2 years).

Seasonal pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, or day of the week).

Time series decomposition

$$y_t = f(S_t, T_t, R_t)$$

where $y_t =$ data at period t

$T_t =$ trend-cycle component at period t

$S_t =$ seasonal component at period t

$R_t =$ remainder component at period t

Time series decomposition

$$y_t = f(S_t, T_t, R_t)$$

where	$y_t =$	data at period t
	$T_t =$	trend-cycle component at period t
	$S_t =$	seasonal component at period t
	$R_t =$	remainder component at period t

Additive decomposition: $y_t = S_t + T_t + R_t$.

Multiplicative decomposition: $y_t = S_t \times T_t \times R_t$.

Time series decomposition

- Additive model appropriate if magnitude of seasonal fluctuations does not vary with level.
- If seasonal are proportional to level of series, then multiplicative model appropriate.
- Multiplicative decomposition more prevalent with economic series
- Alternative: use a Box-Cox transformation, and then use additive decomposition.
- Logs turn multiplicative relationship into an additive relationship:

$$y_t = S_t \times T_t \times E_t \quad \Rightarrow \quad \log y_t = \log S_t + \log T_t + \log R_t.$$

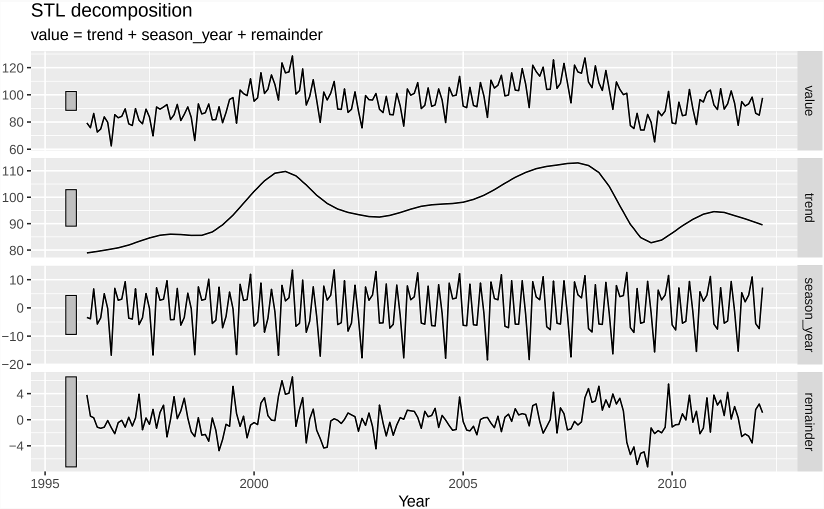
Decomposition dable

```r
elecequip <- as_tsibble(fpp2::elecequip)
dcmp <- elecequip %>% STL(value ~ season(window = 7))
dcmp
```

```
## # A dable:             195 x 6 [1M]
## # STL Decomposition: value = trend + season_year +
## #   remainder
##       index value trend season_year remainder season_adjust
##       <mth> <dbl> <dbl>       <dbl>     <dbl>         <dbl>
##  1 1996 Jan  79.4  78.9      -3.37     3.81           82.7
##  2 1996 Feb  75.8  79.1      -3.87     0.547          79.7
##  3 1996 Mar  86.3  79.3       6.73     0.301          79.6
##  4 1996 Apr  72.6  79.5      -5.74    -1.15           78.3
##  5 1996 May  74.9  79.7      -3.53    -1.31           78.4
##  6 1996 Jun  83.8  79.9       5.03    -1.14           78.8
##  7 1996 Jul  79.8  80.1      -0.222   -0.119          80.0
##  8 1996 Aug  62.4  80.4     -16.8     -1.21           79.2
##  9 1996 Sep  85.4  80.6       6.94    -2.15           78.5
```

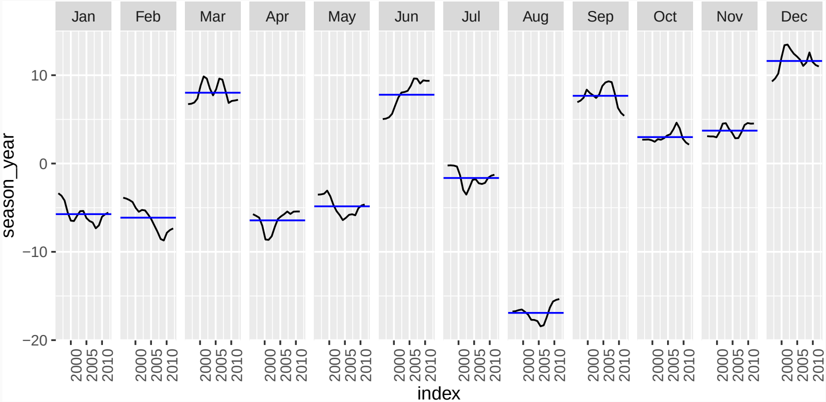
Euro electrical equipment

```r
autoplot(dcmp) + xlab("Year")
```



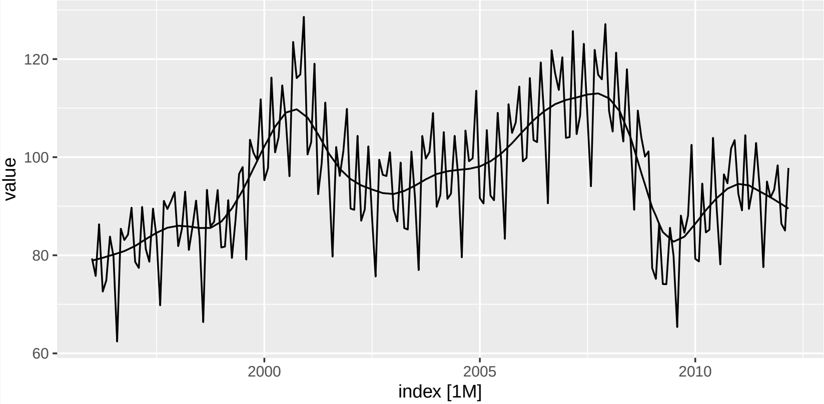
Euro electrical equipment

```
dcmp %>% gg_subseries(season_year)
```



Euro electrical equipment

```r
autoplot(elecequip, series="Data") +
  autolayer(dcmp, trend, series="Trend-cycle")
```



Your turn

Repeat the decomposition using

```
elecequip %>%
  STL(value ~ season(window=7) + trend(window=11)) %>%
  autoplot()
```

What happens as you change `season(window = ???)` and `trend(window = ???)`?

Outline

1 Decompositions

2 Time series components

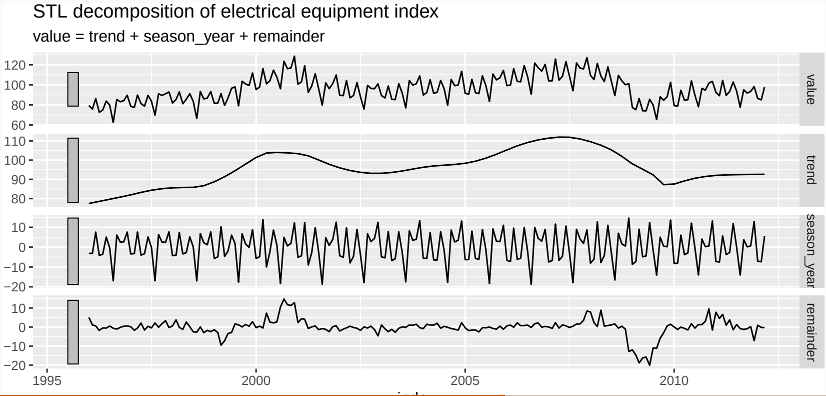
3 STL decomposition

STL decomposition

- STL: “Seasonal and Trend decomposition using Loess”
- Very versatile and robust.
- Unlike X-12-ARIMA, STL will handle any type of seasonality.
- Seasonal component allowed to change over time, and rate of change controlled by user.
- Smoothness of trend-cycle also controlled by user.
- Robust to outliers
- Not trading day or calendar adjustments.
- Only additive.
- Take logs to get multiplicative decomposition.
- Use Box-Cox transformations to get other decompositions.

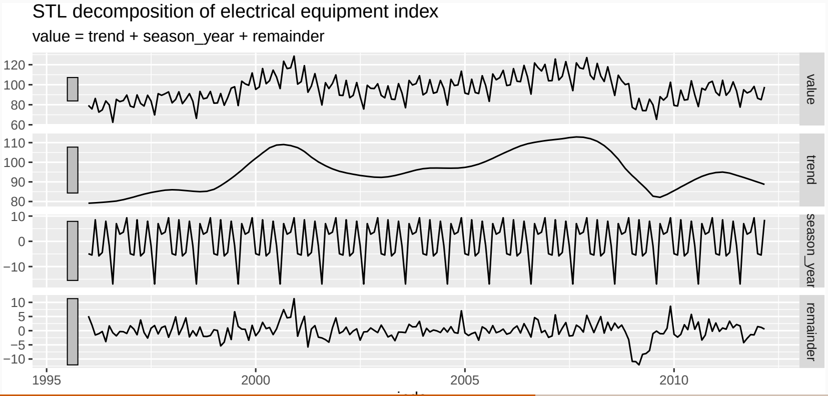
STL decomposition

```
dcmp <- elecequip %>%
  STL(value ~ season(window = 5), robust = TRUE)
autoplot(dcmp) +
  ggtitle("STL decomposition of electrical equipment index")
```



STL decomposition

```r
fit <- elecequip %>%
  STL(value ~ season(window="periodic"), robust=TRUE)
autoplot(fit) +
  ggtitle("STL decomposition of electrical equipment index")
```



STL decomposition

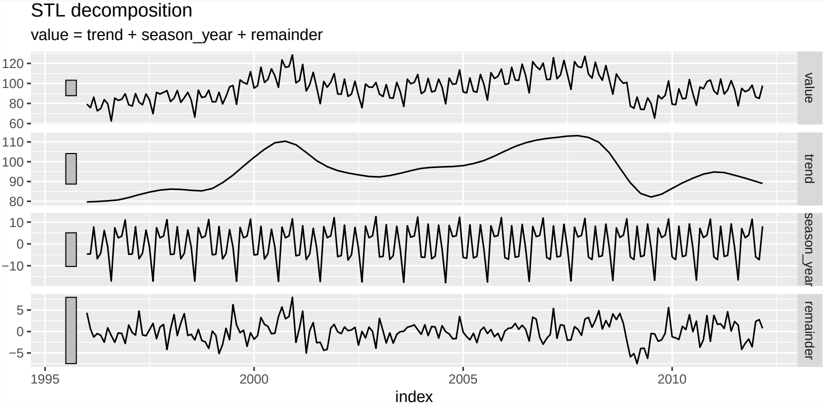
```
elecequip %>%
  STL(value ~ season(window = 5))

elecequip %>%
  STL(value ~ trend(window=15) + season(window="periodic"),
      robust = TRUE)
```

- `trend(window = ?)` controls wiggliness of trend component.
- `season(window = ?)` controls variation on seasonal component.

STL decomposition

```
elecequip %>% STL(value) %>% autoplot()
```



- `STL()` chooses `season(window=13)` by default
- Can include transformations. # Features

Feature extraction and statistics

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

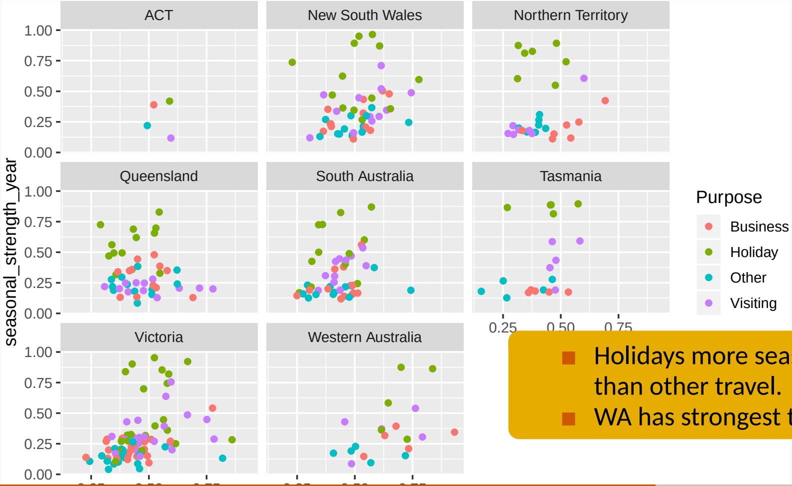
```
## # A tibble: 304 x 10
##    Region State Purpose trend_strength seasonal_streng~
##    <chr>  <chr> <chr>            <dbl>            <dbl>
##  1 Adela~ Sout~ Busine~          0.451            0.380
##  2 Adela~ Sout~ Holiday          0.541            0.601
##  3 Adela~ Sout~ Other            0.743            0.189
##  4 Adela~ Sout~ Visiti~          0.433            0.446
##  5 Adela~ Sout~ Busine~          0.453            0.140
##  6 Adela~ Sout~ Holiday          0.512            0.244
##  7 Adela~ Sout~ Other            0.584            0.374
##  8 Adela~ Sout~ Visiti~          0.481            0.228
##  9 Alice~ Nort~ Busine~          0.526            0.224
## 10 Alice~ Nort~ Holiday          0.377            0.827
## # ... with 294 more rows, and 5 more variables:
## #   spikiness <dbl>, linearity <dbl>, curvature <dbl>,
## #   seasonal_peak_year <dbl>, seasonal_trough_year <dbl>
```

Feature extraction and statistics

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

Feature extraction and statistics

```
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.

Feature extraction and statistics

Find the most seasonal time series:

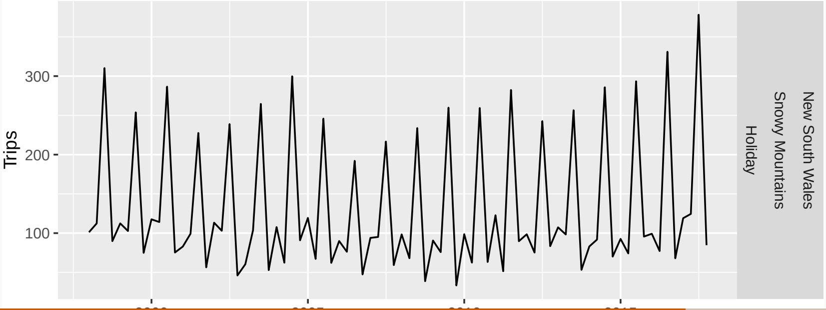
```
most_seasonal <- tourism %>%
  features(Trips, feat_stl) %>%
  filter(seasonal_strength_year == max(seasonal_strength_year))
```

Feature extraction and statistics

Find the most seasonal time series:

```
most_seasonal <- tourism %>%
  features(Trips, feat_stl) %>%
  filter(seasonal_strength_year == max(seasonal_strength_year))
```

```
tourism %>%
  right_join(most_seasonal, by = c("State","Region","Purpose")) %>%
  ggplot(aes(x = Quarter, y = Trips)) + geom_line() +
  facet_grid(vars(State,Region,Purpose))
```



Feature extraction and statistics

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

All features from the feasts package

```
## # A tibble: 304 x 45
##    Region State Purpose trend_strength seasonal_streng~
##    <chr>  <chr> <chr>            <dbl>            <dbl>
##  1 Adela~ Sout~ Busine~          0.451            0.380
##  2 Adela~ Sout~ Holiday          0.541            0.601
##  3 Adela~ Sout~ Other            0.743            0.189
##  4 Adela~ Sout~ Visiti~          0.433            0.446
##  5 Adela~ Sout~ Busine~          0.453            0.140
##  6 Adela~ Sout~ Holiday          0.512            0.244
##  7 Adela~ Sout~ Other            0.584            0.374
##  8 Adela~ Sout~ Visiti~          0.481            0.228
##  9 Alice~ Nort~ Busine~          0.526            0.224
## 10 Alice~ Nort~ Holiday          0.377            0.827
## # ... with 294 more rows, and 40 more variables:
## #   spikiness <dbl>, linearity <dbl>, curvature <dbl>,
## #   seasonal_peak_year <dbl>, seasonal_trough_year <dbl>,
## #   acf1 <dbl>, acf10 <dbl>, diff1_acf1 <dbl>,
## #   diff1_acf10 <dbl>, diff2_acf1 <dbl>, diff2_acf10 <dbl>,
```

Feature extraction and statistics

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

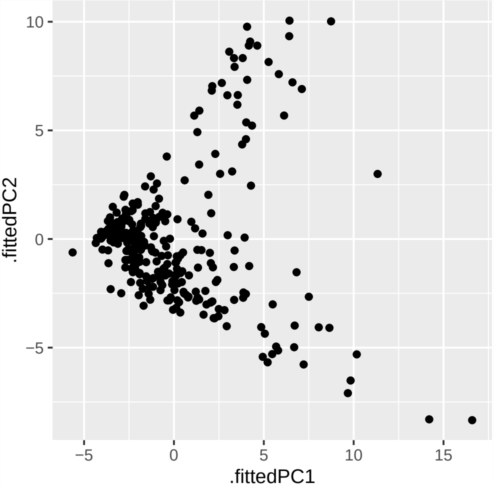
```
## # A tibble: 304 x 88
##    .rownames Region State Purpose trend_strength
##    <fct>     <chr>  <chr> <chr>            <dbl>
##  1 1         Adela~ Sout~ Busine~          0.451
##  2 2         Adela~ Sout~ Holiday          0.541
##  3 3         Adela~ Sout~ Other            0.743
##  4 4         Adela~ Sout~ Visiti~          0.433
##  5 5         Adela~ Sout~ Busine~          0.453
##  6 6         Adela~ Sout~ Holiday          0.512
##  7 7         Adela~ Sout~ Other            0.584
##  8 8         Adela~ Sout~ Visiti~          0.481
##  9 9         Alice~ Nort~ Busine~          0.526
## 10 10        Alice~ Nort~ Holiday          0.377
## # ... with 294 more rows, and 83 more variables:
## #   seasonal_strength_year <dbl>, spikiness <dbl>,
## #   linearity <dbl>, curvature <dbl>,
## #   seasonal_peak_year <dbl>, seasonal_trough_year <dbl>,
## #   acf1 <dbl>, acf10 <dbl>, diff1_acf1 <dbl>,
```

Principal components based on all features from the feasts package

Feature extraction and statistics

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

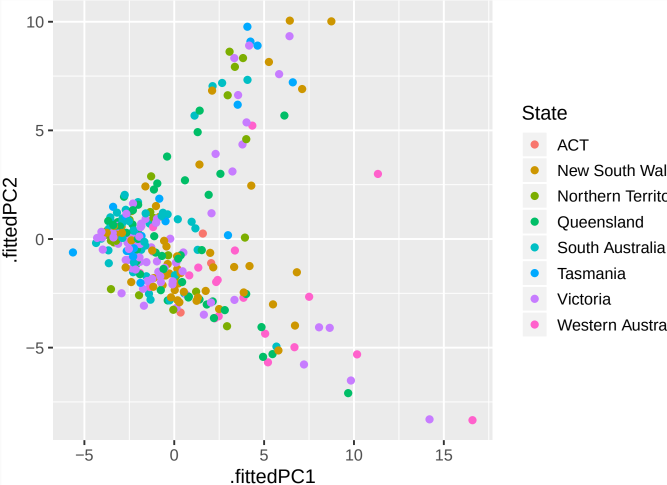
Principal components based on all features from the feasts package



Feature extraction and statistics

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

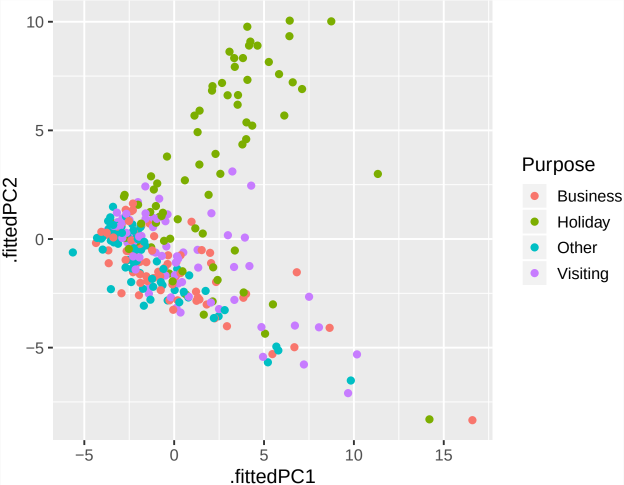
Principal components based on all features from the feasts package



Feature extraction and statistics

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

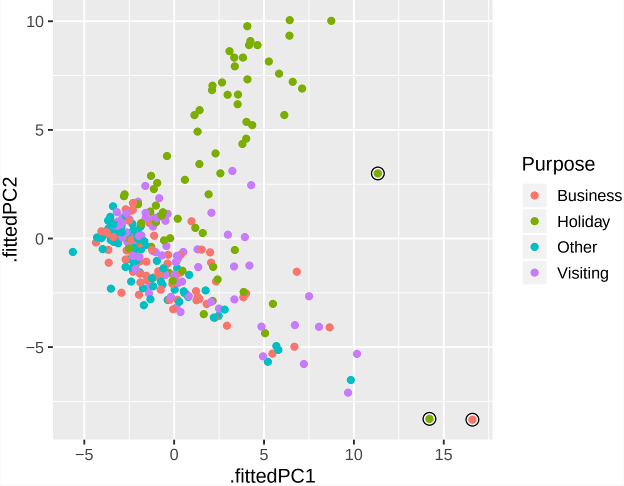
Principal components based on all features from the feasts package



Feature extraction and statistics

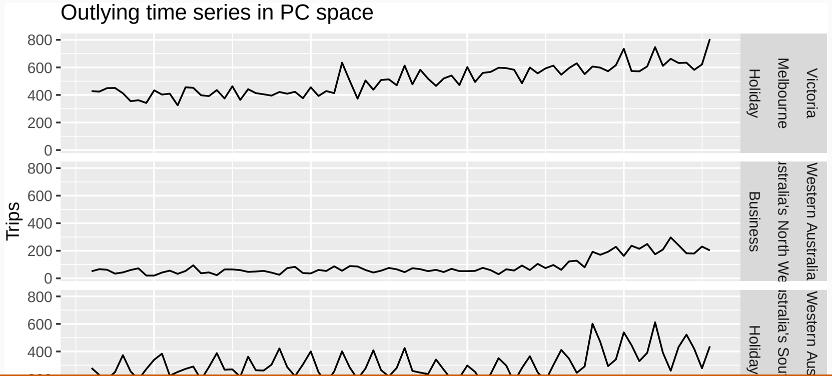
```
pcs %>% ggplot(aes(x=.fittedPC1, y=.fittedPC2, col=Purpose)) +
  geom_point() + theme(aspect.ratio=1)
```

Principal components based on all features from the feasts package



Feature extraction and statistics

```
outliers %>%
  left_join(tourism, by = c("State", "Region", "Purpose")) %>%
  ggplot(aes(x = Quarter, y = Trips)) +
    geom_line() +
    facet_grid(vars(State,Region,Purpose)) +
    ggtitle("Outlying time series in PC space") +
    theme(legend.position = "none")
```



Acknowledgements

Mitchell O'Hara-Wild

Earo Wang

feasts.tidyverts.org

robjhyndman.com