MONASH University

MONASH BUSINESS SCHOOL

High dimensional time series analysis

3. Time series features

robjhyndman.com/hdtsa

Outline

Outline

Strength of seasonality and trend

STL decomposition

$$y_t = T_t + S_t + R_t$$

Seasonal strength

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

Trend strength

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

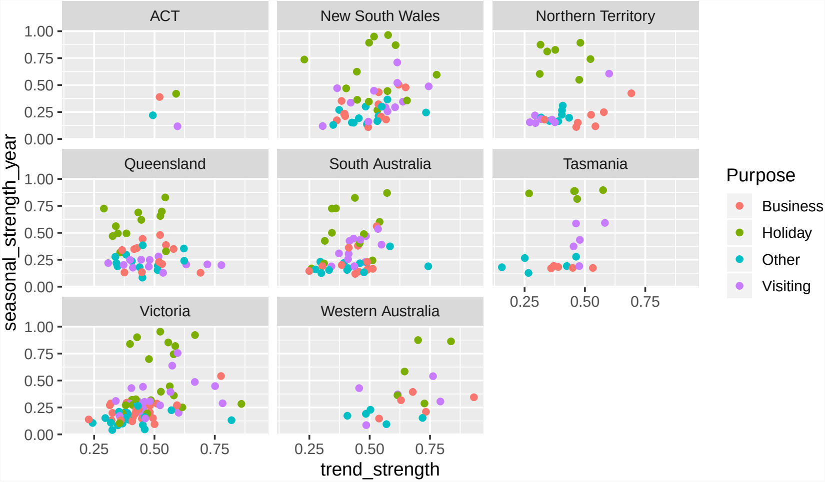
Feature extraction and statistics

```
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>
```

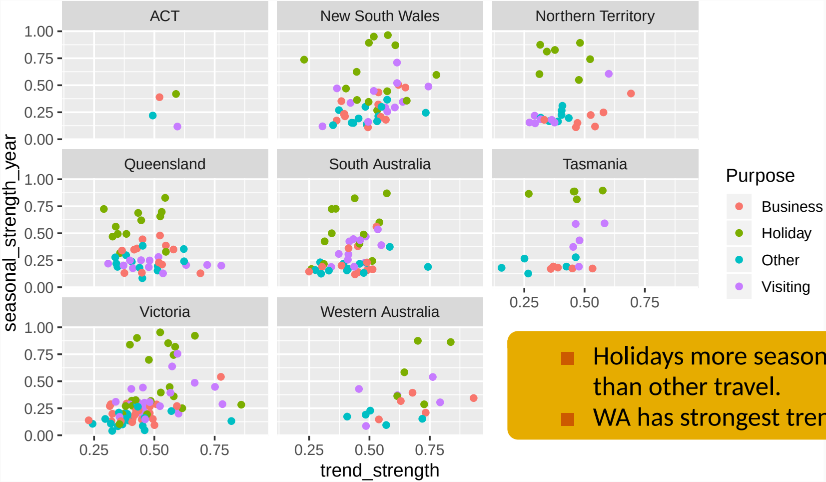
Feature extraction and statistics

```r
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:

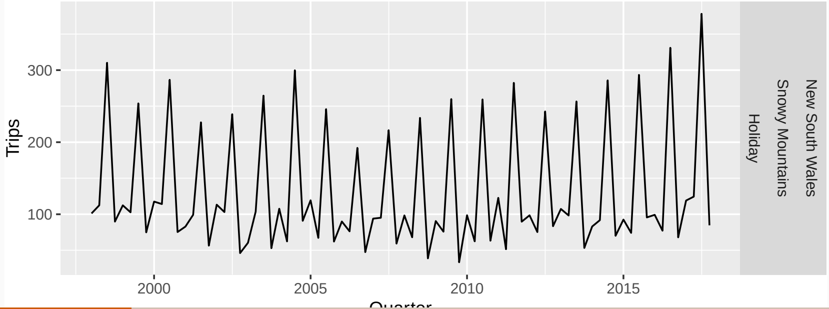
```
most_seasonal <- tourism %>%
  features(Trips, feat_stl) %>%
  filter(seasonal_strength_year == max(seasonal_strength_year))
```

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

Find the most trended time series:

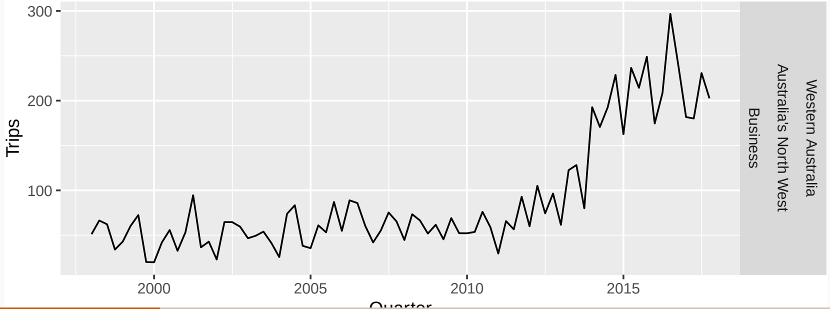
```
most_trended <- tourism %>%
  features(Trips, feat_stl) %>%
  filter(trend_strength == max(trend_strength))
```

Feature extraction and statistics

Find the most trended time series:

```
most_trended <- tourism %>%
  features(Trips, feat_stl) %>%
  filter(trend_strength == max(trend_strength))
```

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



Outline

Lab Session 5

- Use `GGally::ggpairs()` to look at the relationships between the STL-based features. You might wish to change `seasonal_peak_year` and `seasonal_trough_year` to factors.
- Which is the peak quarter for holidays in each state?

Outline

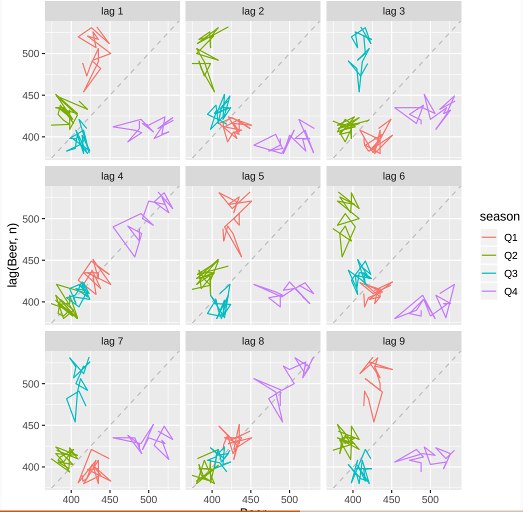
Example: Beer production

```
new_production <- aus_production %>%
  filter(year(Quarter) >= 1992)
new_production
```

```
## # A tsibble: 74 x 7 [1Q]
##     Quarter  Beer Tobacco Bricks Cement Electricity   Gas
##       <qtr> <dbl>   <dbl>  <dbl>  <dbl>       <dbl> <dbl>
##  1 1992 Q1   443    5777    383   1289       38332   117
##  2 1992 Q2   410    5853    404   1501       39774   151
##  3 1992 Q3   420    6416    446   1539       42246   175
##  4 1992 Q4   532    5825    420   1568       38498   129
##  5 1993 Q1   433    5724    394   1450       39460   116
##  6 1993 Q2   421    6036    462   1668       41356   149
##  7 1993 Q3   410    6570    475   1648       42949   163
##  8 1993 Q4   512    5675    443   1863       40974   138
##  9 1994 Q1   449    5311    421   1468       40162   127
```

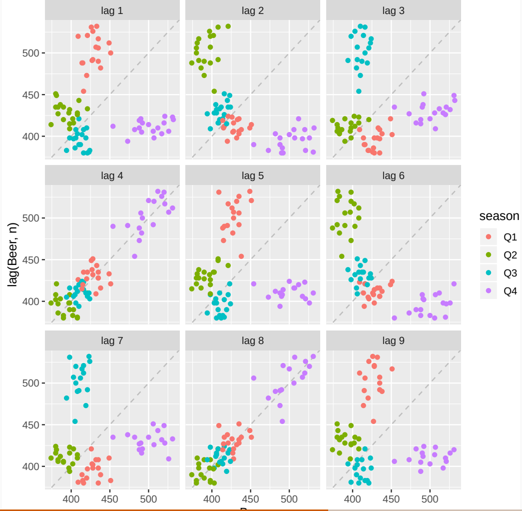
Example: Beer production

```
new_production %>% gg_lag(Beer)
```



Example: Beer production

```
new_production %>% gg_lag(Beer, geom='point')
```



Lagged scatterplots

- Each graph shows y_t plotted against y_{t-k} for different values of k.
- The autocorrelations are the correlations associated with these scatterplots.

Autocorrelation

Covariance and **correlation**: measure extent of **linear relationship** between two variables (y and X).

Autocorrelation

Covariance and **correlation**: measure extent of **linear relationship** between two variables (y and X).

Autocovariance and **autocorrelation**: measure linear relationship between **lagged values** of a time series y.

Autocorrelation

Covariance and **correlation**: measure extent of **linear relationship** between two variables (y and X).

Autocovariance and **autocorrelation**: measure linear relationship between **lagged values** of a time series y.

We measure the relationship between:

- y_t and y_{t-1}
- y_t and y_{t-2}
- y_t and y_{t-3}
- etc.

Autocorrelation

We denote the sample autocovariance at lag k by c_k and the sample autocorrelation at lag k by r_k. Then define

$$c_k = \frac{1}{T}\sum_{t=k+1}^{T}(y_t - \bar{y})(y_{t-k} - \bar{y})$$

and $$r_k = c_k/c_0$$

Autocorrelation

We denote the sample autocovariance at lag k by c_k and the sample autocorrelation at lag k by r_k. Then define

$$c_k = \frac{1}{T}\sum_{t=k+1}^{T}(y_t - \bar{y})(y_{t-k} - \bar{y})$$

and $$r_k = c_k/c_0$$

- r_1 indicates how successive values of y relate to each other
- r_2 indicates how y values two periods apart relate to each other
- r_k is *almost* the same as the sample correlation between y_t and y_{t-k}.

Autocorrelation

Results for first 9 lags for beer data:

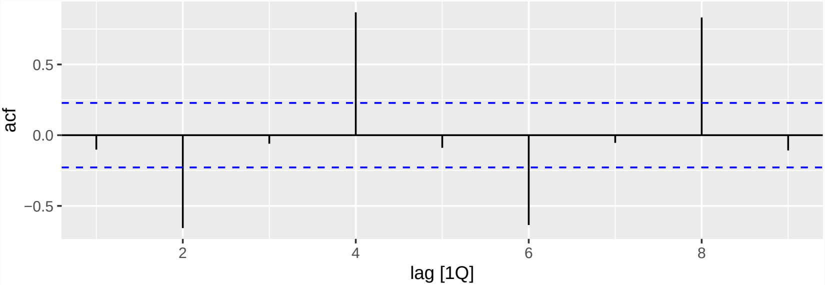
```
new_production %>% ACF(Beer, lag_max = 9)
```

```
## # A tsibble: 9 x 2 [1Q]
##      lag     acf
##    <lag>   <dbl>
## 1     1Q -0.102
## 2     2Q -0.657
## 3     3Q -0.0603
## 4     4Q  0.869
## 5     5Q -0.0892
## 6     6Q -0.635
## 7     7Q -0.0542
## 8     8Q  0.832
```

Autocorrelation

Results for first 9 lags for beer data:

```
new_production %>% ACF(Beer, lag_max = 9) %>% autoplot()
```

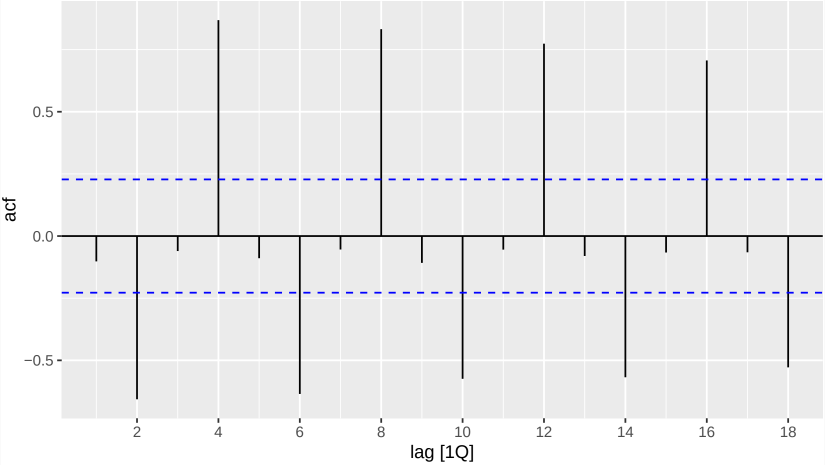


Autocorrelation

- r_4 higher than for the other lags. This is due to **the seasonal pattern in the data**: the peaks tend to be **4 quarters** apart and the troughs tend to be **2 quarters** apart.
- r_2 is more negative than for the other lags because troughs tend to be 2 quarters behind peaks.
- Together, the autocorrelations at lags 1, 2, ..., make up the *autocorrelation* or ACF.
- The plot is known as a **correlogram**

ACF

```
new_production %>% ACF(Beer) %>% autoplot()
```



Australian holidays

```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.
```

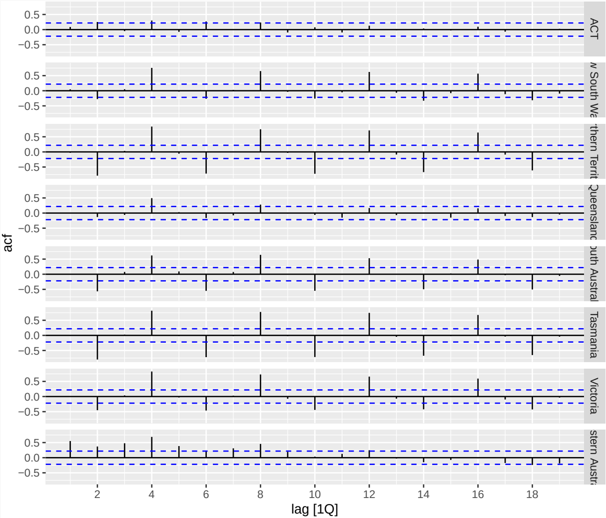
Australian holidays

```r
holidays %>% ACF(Trips)
```

```
## # A tsibble: 152 x 3 [1Q]
## # Key:        State [8]
##    State    lag      acf
##    <chr> <lag>    <dbl>
##  1 ACT       1Q  0.0877
##  2 ACT       2Q  0.252
##  3 ACT       3Q -0.0496
##  4 ACT       4Q  0.300
##  5 ACT       5Q -0.0741
##  6 ACT       6Q  0.269
##  7 ACT       7Q -0.00504
##  8 ACT       8Q  0.236
##  9 ACT       9Q -0.0953
## 10 ACT      10Q  0.0750
## # ... with 142 more rows
```

Australian holidays

```
holidays %>% ACF(Trips) %>% autoplot()
```



Feature extraction and statistics

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

```
## # A tibble: 304 x 10
##    Region State Purpose     acf1 acf10 diff1_acf1
##    <chr>  <chr> <chr>      <dbl> <dbl>      <dbl>
##  1 Adela~ Sout~ Busine~  0.0333  0.131     -0.520
##  2 Adela~ Sout~ Holiday  0.0456  0.372     -0.343
##  3 Adela~ Sout~ Other    0.517   1.15      -0.409
##  4 Adela~ Sout~ Visiti~  0.0684  0.294     -0.394
##  5 Adela~ Sout~ Busine~  0.0709  0.134     -0.580
##  6 Adela~ Sout~ Holiday  0.131   0.313     -0.536
##  7 Adela~ Sout~ Other    0.261   0.330     -0.253
##  8 Adela~ Sout~ Visiti~  0.139   0.117     -0.472
##  9 Alice~ Nort~ Busine~  0.217   0.367     -0.500
## 10 Alice~ Nort~ Holiday -0.00660 2.11      -0.153
## # ... with 294 more rows, and 4 more variables:
## #   diff1_acf10 <dbl>, diff2_acf1 <dbl>, diff2_acf10 <dbl>,
## #   season_acf1 <dbl>
```

Outline

Feature extraction and statistics

```
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

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

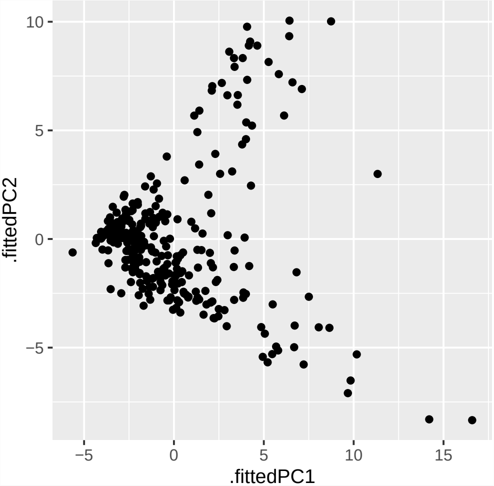
```
## # A tibble: 304 x 88
##    .rownames Region State Purpose trend_strength
##    <fct>     <chr>  <chr> <chr>            <dbl>
##  1 1         Adela~ Sout~ Busine~          0.451
##  2 2         Adela~ Sout~ Holiday          0.541
##  3 3         Adela~ Sout~ Other            0.743
##  4 4         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

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

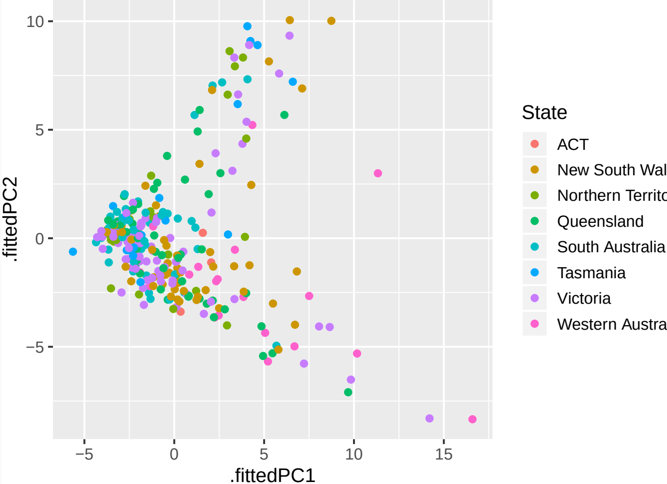
Principal components based on all features from the feasts package



Feature extraction and statistics

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

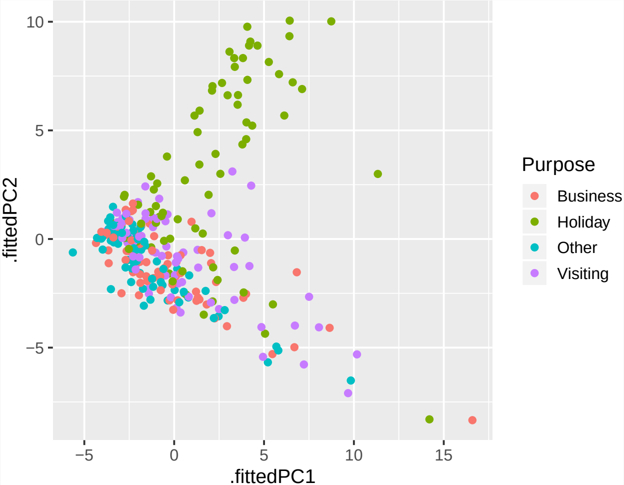
Principal components based on all features from the feasts package



Feature extraction and statistics

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

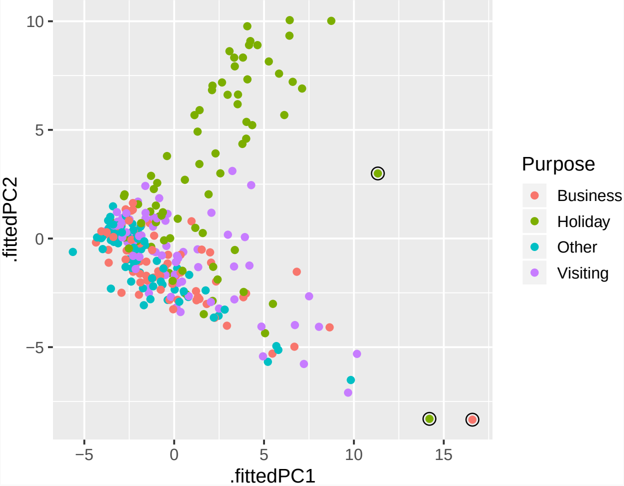
Principal components based on all features from the feasts package



Feature extraction and statistics

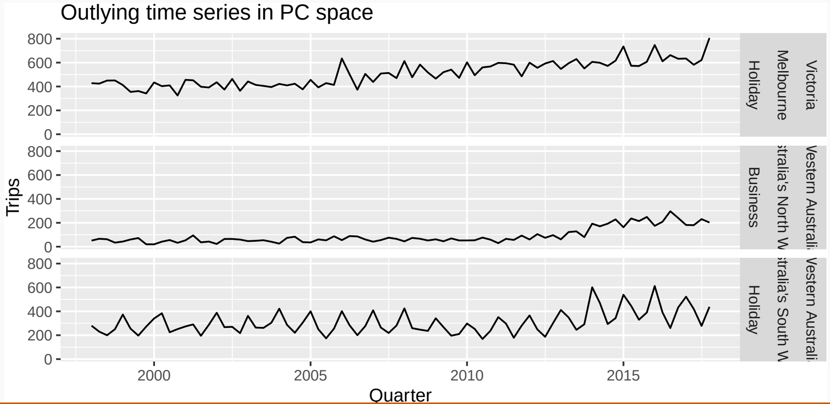
```
pcs %>% ggplot(aes(x=.fittedPC1, y=.fittedPC2, col=Purpose)) +
  geom_point() + theme(aspect.ratio=1)
```

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")
```



Outline

Lab Session 6

- Use a feature-based approach to look for outlying series in `PBS`.
- What is unusual about these series?