10.2 Grouped time series

Grouped time series involve more general aggregation structures than hierarchical time series. With grouped time series, the structure does not naturally disaggregate in a unique hierarchical manner, and often the disaggregating factors are both nested and crossed. For example, we could further disaggregate all geographic levels of the Australian tourism data by purpose of travel (such as holidays, business, etc.). So we could consider visitors nights split by purpose of travel for the whole of Australia, and for each state, and for each zone. Then we describe the structure as involving the purpose of travel "crossed" with the geographic hierarchy.

Figure 10.4 shows a $K = 2$-level grouped structure. At the top of the grouped structure is the Total, the most aggregate level of the data, again represented by y_t. The Total can be disaggregated by attributes (A, B) forming series $y_{\mathrm{A},t}$ and $y_{\mathrm{B},t}$, or by attributes (X, Y) forming series $y_{\mathrm{X},t}$ and $y_{\mathrm{Y},t}$. At the bottom level, the data are disaggregated by both attributes.

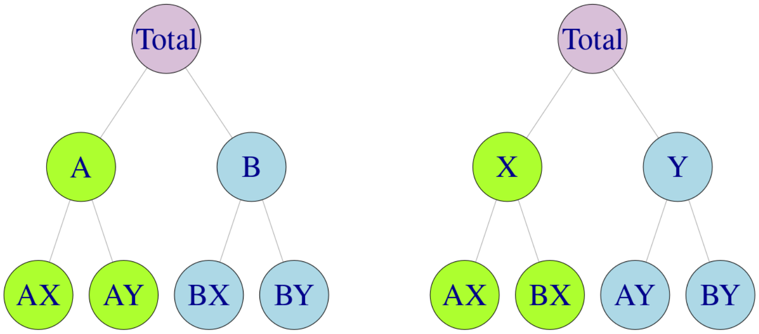

Figure 10.4: Alternative representations of a two level grouped structure.

This example shows that there are alternative aggregation paths for grouped structures. For any time t, as with the hierarchical structure,

$$y_t = y_{\mathrm{AX},t} + y_{\mathrm{AY},t} + y_{\mathrm{BX},t} + y_{\mathrm{BY},t}.$$

However, for the first level of the grouped structure,

$$y_{\text{A},t} = y_{\text{AX},t} + y_{\text{AY},t} \qquad y_{\text{B},t} = y_{\text{BX},t} + y_{\text{BY},t} \tag{10.4}$$

but also

$$y_{\text{X},t} = y_{\text{AX},t} + y_{\text{BX},t} \qquad y_{\text{Y},t} = y_{\text{AY},t} + y_{\text{BY},t}. \tag{10.5}$$

These equalities can again be represented by the $n \times m$ summing matrix $\boldsymbol{S}$. The total number of series is $n = 9$ with $m = 4$ series at the bottom-level. For the grouped structure in Figure 10.4 we write

$$\begin{bmatrix} y_t \\ y_{\text{A},t} \\ y_{\text{B},t} \\ y_{\text{X},t} \\ y_{\text{Y},t} \\ y_{\text{AX},t} \\ y_{\text{AY},t} \\ y_{\text{BX},t} \\ y_{\text{BY},t} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{\text{AX},t} \\ y_{\text{AY},t} \\ y_{\text{BX},t} \\ y_{\text{BY},t} \end{bmatrix},$$

or

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

where the second and third rows of $\boldsymbol{S}$ represent (10.4) and the fourth and fifth rows represent (10.5).

Grouped time series can sometimes be thought of as hierarchical time series that do not impose a unique hierarchical structure, in the sense that the order by which the series can be grouped is not unique.

Example: Australian prison population

The top row of Figure 10.5 shows the total number of prisoners in Australia over the period 2005 Q1 to 2016 Q4. This represents the top-level series in the grouping structure. The rest of the panels show the prison population disaggregated by (i) state[20] (ii) legal status, whether prisoners have already been sentenced or are in remand waiting for a sentence, and (iii) gender. In this example, the three factors are crossed, but none are nested within the others.

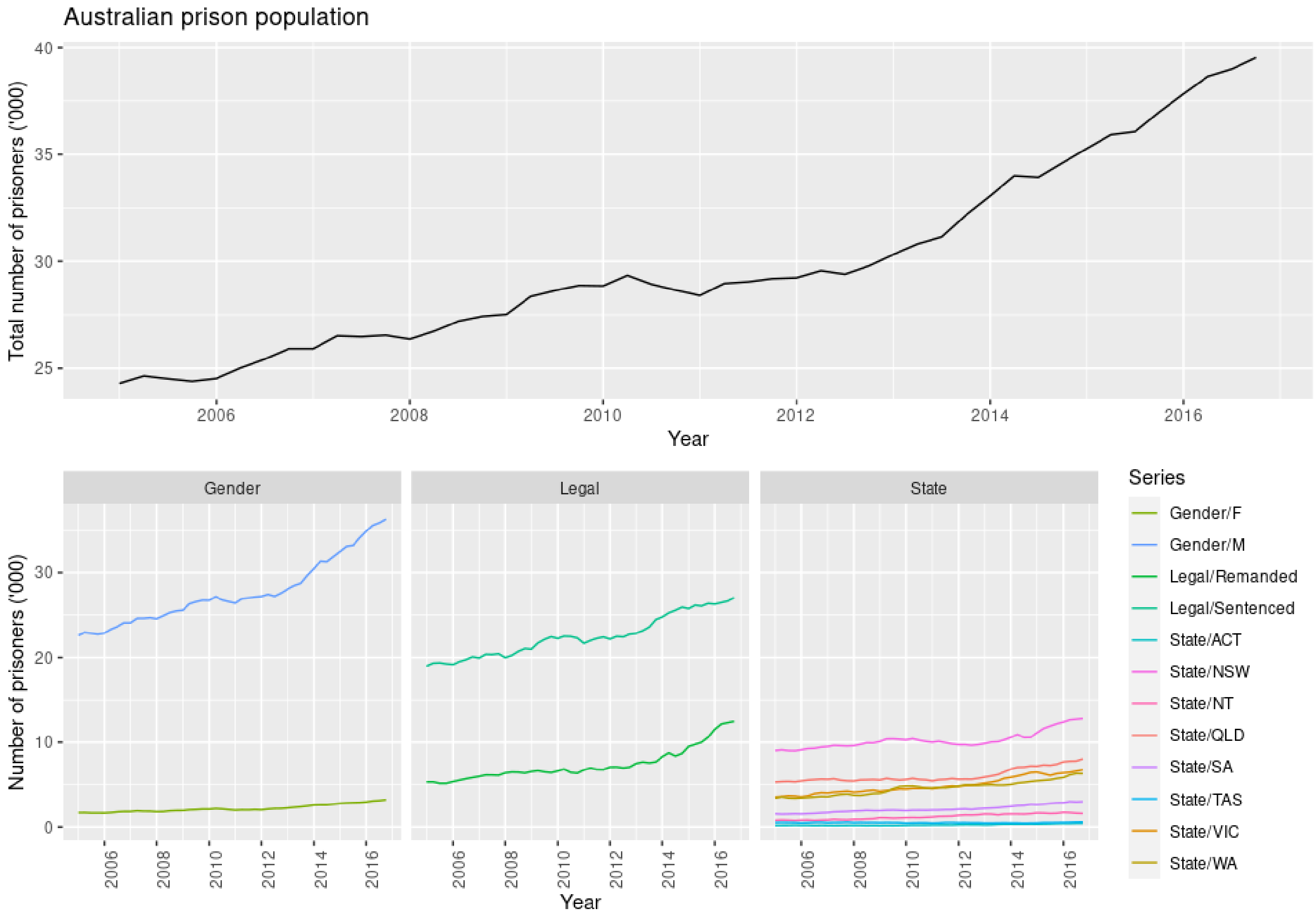

Figure 10.5: Total Australian quarterly adult prison population, disaggregated by state, by legal status and by gender.

To create a grouped time series, we use the `gts()` function. Similar to the `hts()` function, inputs to the `gts()` function are the bottom-level time series and information about the grouping structure. `prison` is a time series matrix containing the bottom-level time series. The information about the grouping structure can be passed in using the `characters` input. (An alternative is to be more explicit about the labelling of the series and use the `groups` input.)

```
prison.gts <- gts(prison/1e3, characters = c(3,1,9),
  gnames = c("State", "Gender", "Legal",
             "State*Gender", "State*Legal",
             "Gender*Legal"))
```

One way to plot the main groups is as follows.

```
prison.gts %>% aggts(level=0:3) %>% autoplot()
```

But with a little more work, we can construct Figure 10.5 using the following code.

```
p1 <- prison.gts %>% aggts(level=0) %>%
  autoplot() + ggtitle("Australian prison population") +
    xlab("Year") + ylab("Total number of prisoners ('000)")
groups <- aggts(prison.gts, level=1:3)
cols <- sample(scales::hue_pal(h=c(15,375),
          c=100,l=65,h.start=0,direction = 1)(NCOL(groups)))
p2 <- as_tibble(groups) %>%
  gather(Series) %>%
  mutate(Date = rep(time(groups), NCOL(groups)),
         Group = str_extract(Series, "([A-Za-z ]*)")) %>%
  ggplot(aes(x=Date, y=value, group=Series, colour=Series)) +
    geom_line() +
    xlab("Year") + ylab("Number of prisoners ('000)") +
    scale_colour_manual(values = cols) +
    facet_grid(.~Group, scales="free_y") +
    scale_x_continuous(breaks=seq(2006,2016,by=2)) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
gridExtra::grid.arrange(p1, p2, ncol=1)
```

Plots of other group combinations can be obtained similarly. Figure 10.6 shows the Australian prison population disaggregated by all possible combinations of two attributes at a time. The top plot shows the prison population disaggregated by state and legal status, the middle panel shows the disaggregation by state and gender and the bottom panel shows the disaggregation by legal status and gender.

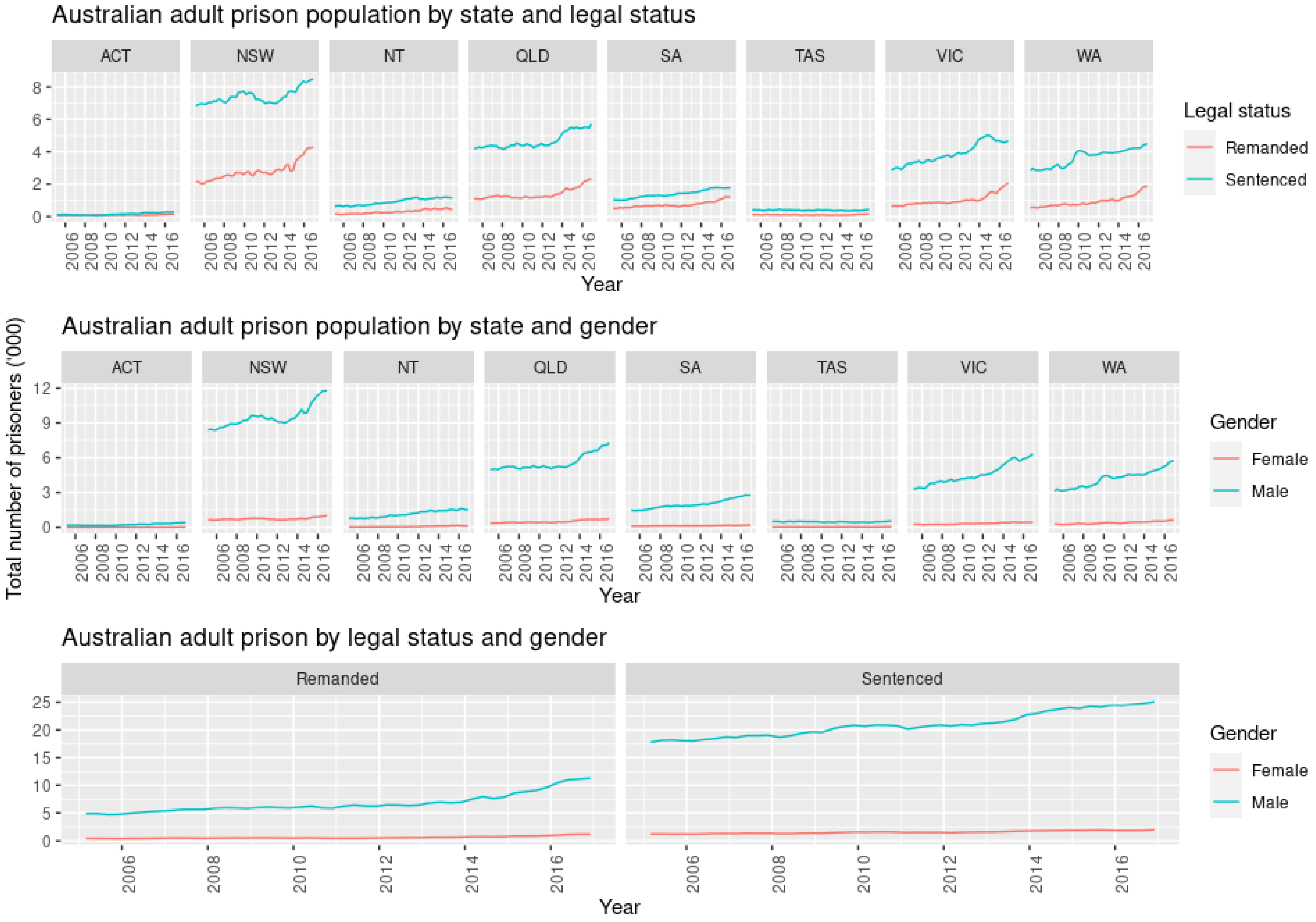

Figure 10.6: Australian adult prison population disaggregated by pairs of attributes.

Figure 10.7 shows the Australian adult prison population disaggregated by all three attributes: state, legal status and gender. These form the bottom-level series of the grouped structure.

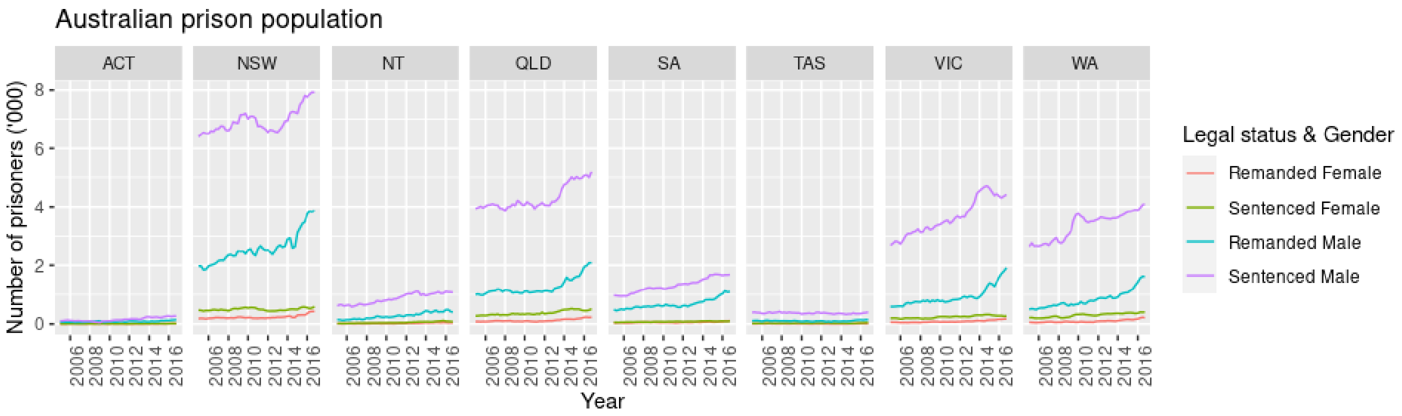

Figure 10.7: Bottom-level time series for the Australian adult prison population, grouped by state, legal status and gender.

20. Australia comprises eight geographic areas six states and two territories: Australian Capital Territory, New South Wales, Northern Territory, Queensland, South Australia, Tasmania, Victoria, Western Australia. In this example we consider all eight areas.↩