Hierarchical_2

Janice Hsu

2023-09-15

```
# Required Libraries
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(tsibble)
```

```
##
## Attaching package: 'tsibble'
```

```
## The following object is masked from 'package:lubridate':
##
##     interval
```

```
## The following object is masked from 'package:zoo':
##
##     index
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, union
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.2     v stringr 1.5.0
## v forcats 1.0.0     v tibble  3.2.1
## v purrr   1.0.1     v tidyr   1.3.0
## v readr   2.1.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x tsibble::interval() masks lubridate::interval()
## x dplyr::lag()        masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(fpp3)
```

```
## -- Attaching packages --------------------------------------------- fpp3 0.5 --
## v tsibbledata 0.4.1     v fable       0.3.3
## v feasts      0.3.1     v fabletools  0.3.3
## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::index()     masks zoo::index()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```r
library(hts)
```

```
## Loading required package: forecast
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
##
## Attaching package: 'forecast'
##
## The following object is masked from 'package:fabletools':
##
##     accuracy
```

```r
library(dplyr)
library(tidyr)
library(forecast)
```

```r
data <- read.csv("HLTH0037_ts_cleaned.csv")
```

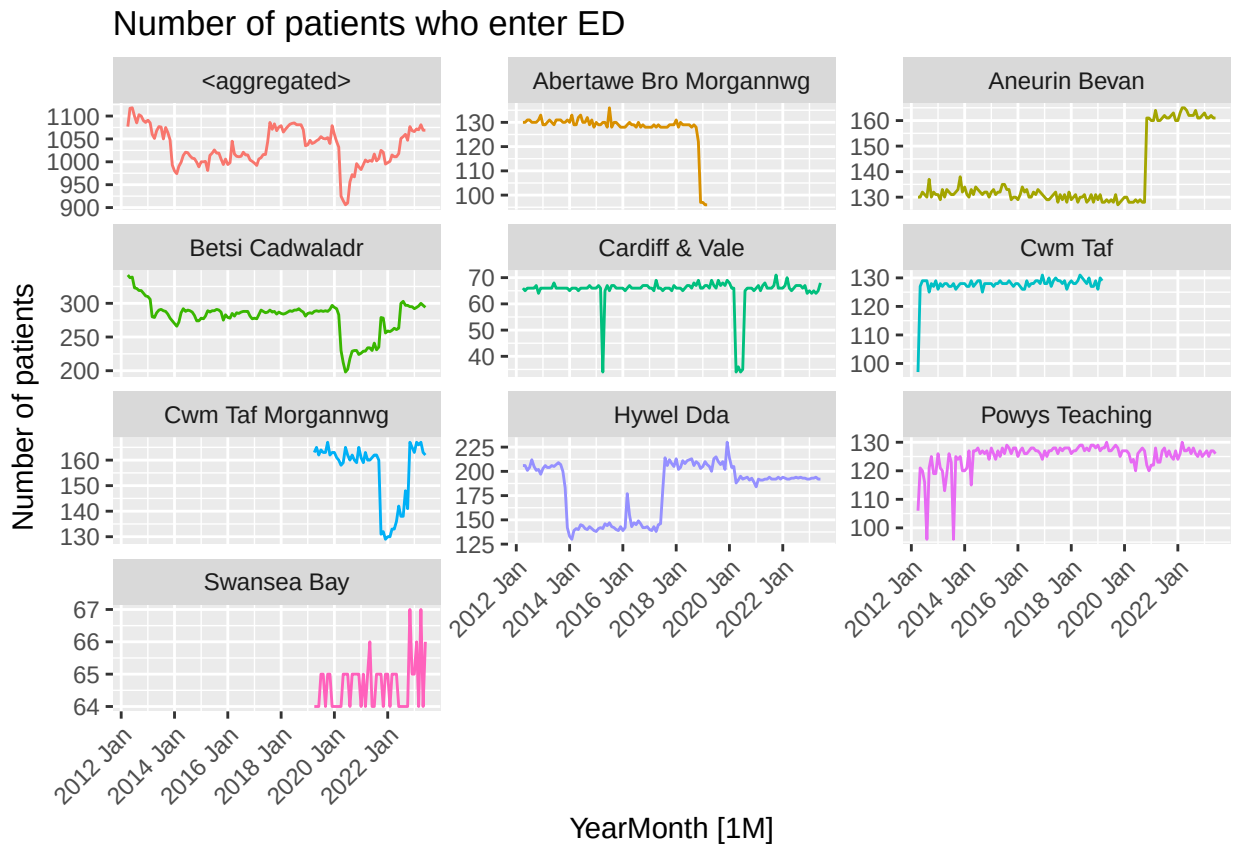
```
data1 <- data %>%
  mutate(YearMonth = yearmonth(YearMonth)) %>%
  as_tsibble(index = YearMonth, key = c(Age_Code, Sex_ItemName_ENG, Hospital_Code, Hospital_ItemName_EN

data1 <- data1 %>%
  mutate(Number = 1)
```

#Number of patients entering ED under different hospital hierarchy

```
data1_hts <- data1 %>%
  aggregate_key(Organisation/Hospital_ItemName_ENG, Number = sum(Number))

data1_hts |>
  filter(is_aggregated(Hospital_ItemName_ENG)) |>
  autoplot(Number) +
  labs(y = "Number of patients",
       title = "Number of patients who enter ED") +
  facet_wrap(vars(Organisation), scales = "free_y", ncol = 3) +
  theme(legend.position = "none")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



• A couple of Local Health Boards (LHBs) were redefined from the 1st of April 2019 onwards: Cwm Taf (27)–> Cwm Taf Morgannwg (30)// Abertawe Bro Morgannwg (26) –> Swansea Bay (31). Therefore, if you decide to forecast at LHB resolution, you might want to consider these 4 as a unique one. • A the Princess of Wales Hospital changed its Local Health Boards • So we analyse these 4 as one organisation

Group the changed Local Health Board together

```
data1_grouped <- data1 %>%
  mutate(Grouped_Organisation = case_when(
    Organisation %in% c("Cwm Taf", "Cwm Taf Morgannwg", "Abertawe Bro Morgannwg", "Swansea Bay") ~ "Grou
    TRUE ~ Organisation
  ))
```

There are 6 Local Health Boards

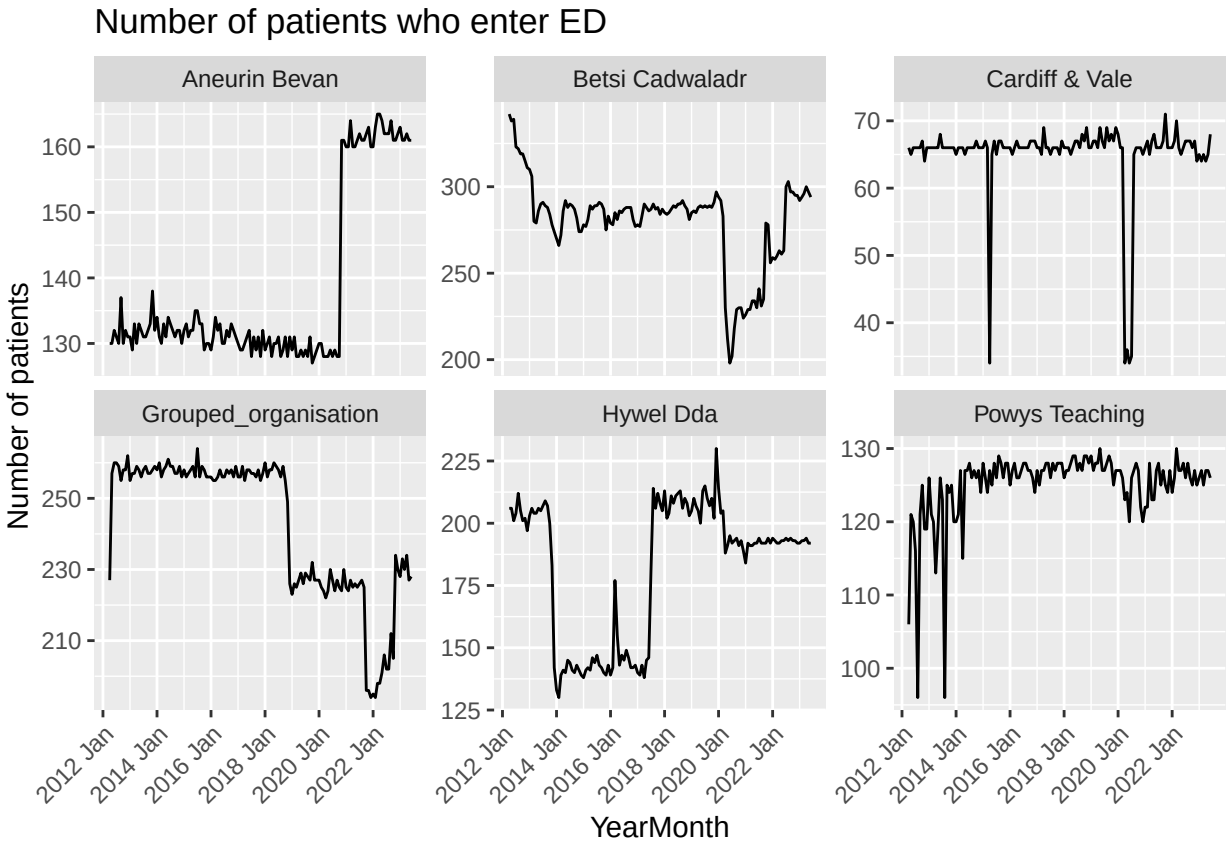
```
unique(data1_grouped$Grouped_Organisation)
```

```
## [1] "Betsi Cadwaladr"      "Hywel Dda"            "Grouped_organisation"
## [4] "Cardiff & Vale"       "Aneurin Bevan"        "Powys Teaching"
```

```
data2_hts <- data1_grouped %>%
  group_by(Grouped_Organisation) %>%
  summarise(Number = sum(Number))
```

Number of patients who enter ED under 6 different local health boards

```
data2_hts |>
  ggplot(aes(x = YearMonth, y = Number)) +
  geom_line(stat = "identity") +
  labs(y = "Number of patients",
       title = "Number of patients who enter ED") +
  facet_wrap(vars(Grouped_Organisation), scales = "free_y", ncol = 3) +
  theme(legend.position = "none") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Change the Age_Code structure into different groups

```
unique(data1_grouped$Age_Code)
```

```
##  [1] "0 to 4"   "18 to 24" "25 to 29" "30 to 34" "35 to 39" "40 to 44"
##  [7] "45 to 49" "5 to 17"  "50 to 54" "55 to 59" "60 to 64" "65 to 69"
## [13] "70 to 74" "75 to 79" "80 to 84" "85"       "Unknown"
```

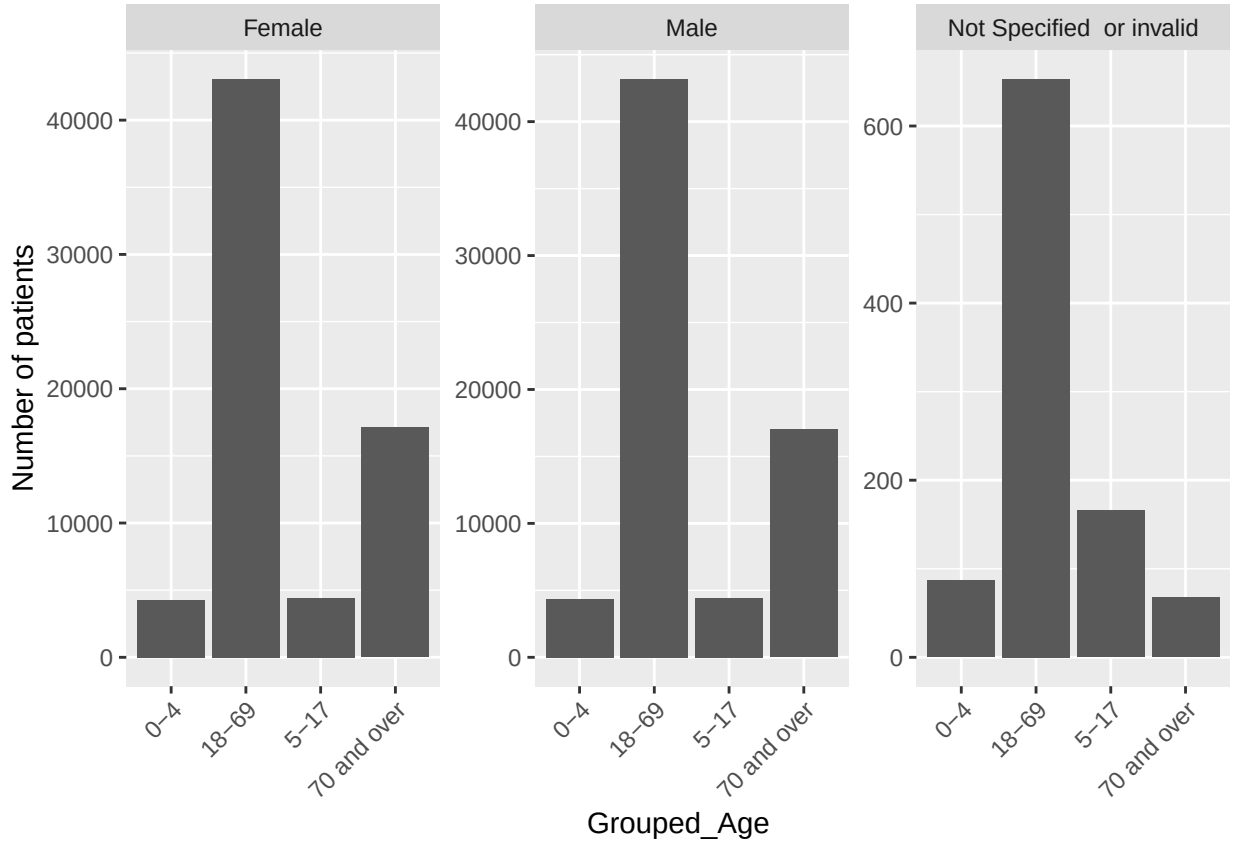
Age group: “0-4”, “5-17”, “18-69”, “70^”

```
data1_grouped_age <- data1_grouped %>%
  filter(Age_Code != "Unknown") %>%
  mutate(Grouped_Age = case_when(
    Age_Code == "0 to 4" ~ "0-4",
    Age_Code == "5 to 17" ~ "5-17",
    Age_Code %in% c("18 to 24", "25 to 29", "30 to 34", "35 to 39",
                    "40 to 44", "45 to 49", "50 to 54", "55 to 59",
                    "60 to 64", "65 to 69") ~ "18-69",
    Age_Code %in% c("70 to 74", "75 to 79", "80 to 84", "85") ~ "70 and over",
    TRUE ~ "Other"
  ))
```

Plot Number of Patients in different age groups

```
data1_gts <- data1_grouped_age %>%
  filter(Sex_ItemName_ENG != "Not Specified or invalid") %>%
  group_by(Grouped_Age, Sex_ItemName_ENG) %>%
  summarize(Number = sum(Number, na.rm = TRUE))

ggplot(data1_gts, aes(x = Grouped_Age, y = Number)) +
  geom_bar(stat = "identity") +
  labs(y = "Number of patients") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~ Sex_ItemName_ENG, scales = "free")
```



#Change the data into wide format

```
data2_wide <- data1_grouped %>%
  group_by(Grouped_Organisation) %>%
  index_by(YearMonth) %>%
  summarise(Number = sum(Number)) %>%
  pivot_wider(names_from = Grouped_Organisation, values_from = Number)
```

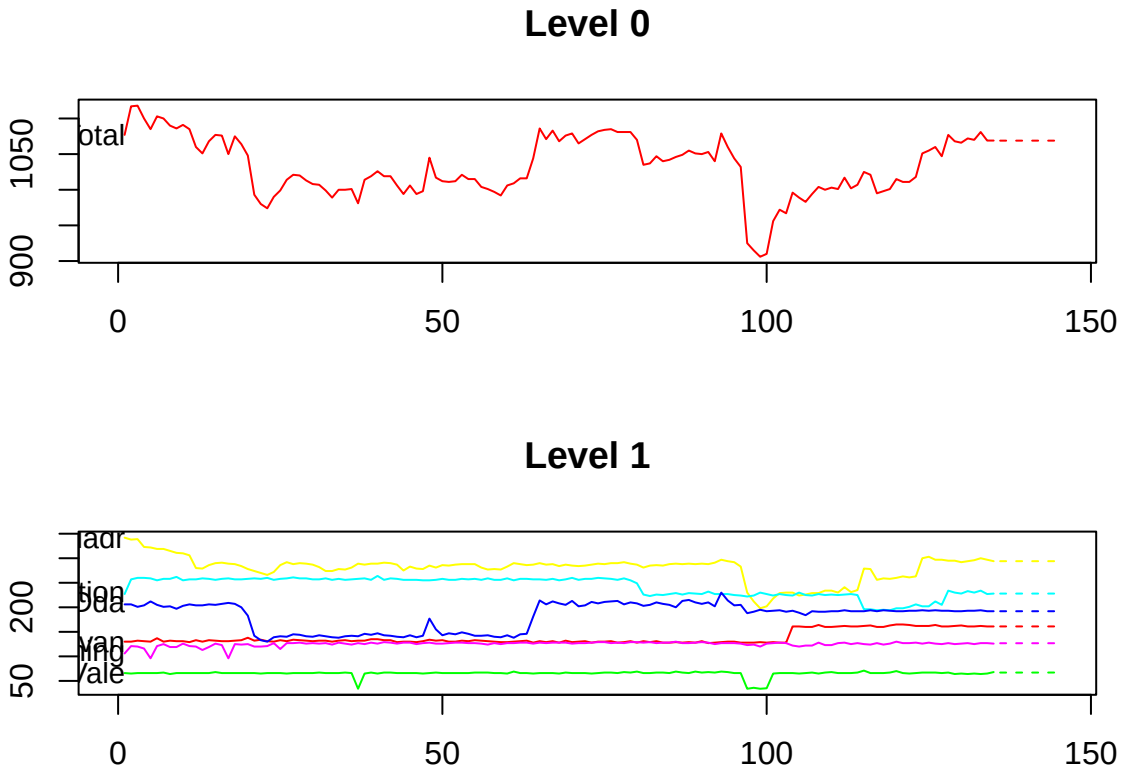
Forecast under Total (hierarchy 1) and Local Health Board (Hierarchy 2)

```
# Create the hierarchical time series object
hts_data <- hts(data2_wide[,-1])  # exclude the YearMonth column
```

```
## Since argument characters are not specified, the default labelling system is used.
```

```
# Forecast using the hierarchical model
forecasts_hts <- forecast(hts_data)

# Plot the forecasts
plot(forecasts_hts)
```




```
# If you want to inspect the forecasts for the aggregated level:
aggregated_forecast <- forecasts_hts$allfcasts$Total

# If you want to inspect the forecasts for an individual health board, say "Aneurin Bevan":
aneurin_bevan_forecast <- forecasts_hts$allfcasts$`Aneurin Bevan`
```

```
str(forecasts_hts)
```

```
## List of 6
##  $ bts    : Time-Series [1:10, 1:6] from 136 to 145: 161 161 161 161 161 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "Aneurin Bevan" "Betsi Cadwaladr" "Cardiff & Vale" "Grouped_organisation" ...
##  $ histy  : Time-Series [1:135, 1:6] from 1 to 135: 130 130 132 131 130 137 130 132 131 131 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "Aneurin Bevan" "Betsi Cadwaladr" "Cardiff & Vale" "Grouped_organisation" ...
##  $ labels :List of 2
##   ..$ Level 0: chr "Total"
##   ..$ Level 1: chr [1:6] "Aneurin Bevan" "Betsi Cadwaladr" "Cardiff & Vale" "Grouped_organisation" .
##  $ method : chr "comb"
##  $ fmethod: chr "ets"
##  $ nodes  :List of 1
##   ..$ Level 1: int 6
##  - attr(*, "class")= chr [1:2] "hts" "gts"
```

- bts: This contains the forecasts for the different series.It's a time-series matrix.
- histy: This represents the historical data for the different series.
- labels: This list contains labels for the different levels in the hierarchy.
- method, fmethod: These are metadata about the forecasting process, indicating the method used for forecasting and combination.
- nodes: Information about the nodes at different levels in the hierarchy.

```r
# Dates for Historical Data
start_date <- yearmonth("2012 Apr")
end_date <- yearmonth("2023 Jun")
hist_dates <- seq(as.Date(start_date), as.Date(end_date), by = "1 month") %>% yearmonth()

# Extract historical time series data from the hts object
historical_data <- zoo(forecasts_hts$histy, order.by = hist_dates)

# Adjusting the forecast start date
forecast_start_date <- as.Date(end_date) %>%
  add_with_rollback(months(1)) %>%
  yearmonth()

forecast_dates <- seq(as.Date(forecast_start_date), by = "1 month", length.out = 10) %>% yearmonth()

# Extract forecast data from the hts object
forecast_data <- zoo(forecasts_hts$bts, order.by = forecast_dates)

# Combine historical and forecast data for plotting
combined_data <- rbind(historical_data, forecast_data)

# Convert the combined data to a long data frame for ggplot
df <- as.data.frame(fortify(combined_data, melt = TRUE))
names(df) <- c("Date", "Health_Board", "Number")

# Plot
ggplot(df, aes(x = Date, y = Number, color = Health_Board)) +
  geom_line() +
  labs(title = "Hierarchical Time Series Forecast",
```

```
      x = "YearMonth",
      y = "Number",
      color = "Health Board") +
scale_x_yearmonth(name = "YearMonth",
                  labels = function(x) format(x, "%b %Y"),
                  breaks = yearmonth(seq(as.Date(start_date), by = "1 year", length.out = 12))) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```