# forecasting ED

Janice Hsu

2023-08-28

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(fpp3)
```

```
## -- Attaching packages --------------------------------------------- fpp3 0.5 --
## v tsibble     1.1.3     v fable       0.3.3
## v tsibbledata 0.4.1     v fabletools  0.3.3
## v feasts      0.3.1
## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## 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()
```

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

```r
data %>%
  select(Hospital_Hierarchy, Organisation) %>%
unique()
```

```
##      Hospital_Hierarchy           Organisation
## 1             W11000023        Betsi Cadwaladr
## 1171          W11000025              Hywel Dda
## 1931          W11000031            Swansea Bay
## 2033          W11000026 Abertawe Bro Morgannwg
## 2364          W11000029         Cardiff & Vale
## 2628          W11000030      Cwm Taf Morgannwg
## 2868          W11000027               Cwm Taf
## 3203          W11000028           Aneurin Bevan
## 3775          W11000024         Powys Teaching
```

```r
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)
```

```r
#Produce a table or plot to show the hierarchy between the organisation and hospital hierarchy

data2 <- data1 %>%
  select(YearMonth, Hospital_ItemName_ENG, Hospital_Hierarchy, Organisation, Number)
```

```r
# Convert to data.table
library(data.table)
```

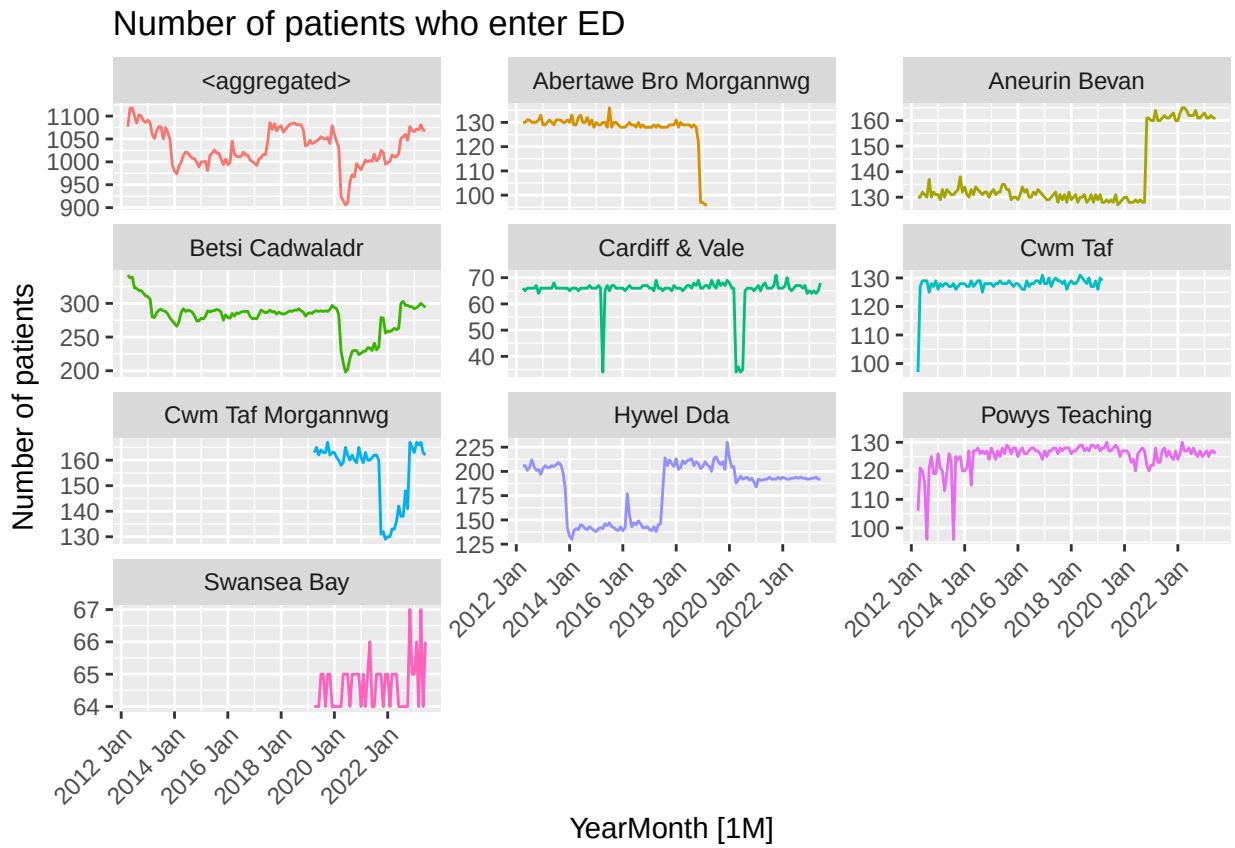
```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:tsibble':
##
##     key

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```r
setDT(data2)

# Create a hierarchical table using data.table operations
hierarchical_table <- data2[, .(Total = sum(Number)), by = .( Organisation, Hospital_ItemName_ENG)]

knitr::kable(hierarchical_table)
```

| Organisation | Hospital_ItemName_ENG | Total |
|---|---|---|
| Betsi Cadwaladr | Ysbyty Glan Clwyd | 4437 |
| Betsi Cadwaladr | Wrexham Maelor Hospital | 4478 |
| Betsi Cadwaladr | Colwyn Bay Community Hospital | 340 |
| Betsi Cadwaladr | Holywell Community Hospital | 394 |
| Betsi Cadwaladr | Mold Community Hospital | 682 |
| Betsi Cadwaladr | Ysbyty Gwynedd | 4381 |
| Betsi Cadwaladr | Llandudno General Hospital | 4342 |
| Betsi Cadwaladr | Bryn Beryl Hospital | 4058 |
| Betsi Cadwaladr | Dolgellau And Barmouth District Hospital | 3066 |
| Betsi Cadwaladr | Ffestiniog Memorial Hospital | 65 |
| Betsi Cadwaladr | Tywyn & District War Memorial Hospital | 2897 |
| Betsi Cadwaladr | Ysbyty Alltwen | 4315 |
| Betsi Cadwaladr | Ysbyty Penrhos Stanley | 4320 |
| Hywel Dda | Glangwili General Hospital | 4353 |
| Hywel Dda | Llandovery Hospital | 1320 |
| Hywel Dda | Bronglais General Hospital | 4339 |
| Hywel Dda | Cardigan And District Memorial Hospital | 921 |
| Hywel Dda | Prince Philip Hospital | 4339 |
| Hywel Dda | Withybush General Hospital | 4349 |
| Hywel Dda | S. Pembs Hosp. Health & Social Care Res Centre | 607 |
| Hywel Dda | New Tenby Cottage Hospital Outpatients | 2977 |
| Hywel Dda | Cardigan Integrated Care Centre | 1368 |
| Swansea Bay | Morriston Hospital | 1653 |
| Swansea Bay | Neath Port Talbot Hospital | 1646 |
| Abertawe Bro Morgannwg | Princess Of Wales Hospital | 2764 |
| Abertawe Bro Morgannwg | Singleton Hospital | 2558 |
| Abertawe Bro Morgannwg | Morriston Hospital | 2716 |
| Abertawe Bro Morgannwg | Neath Port Talbot Hospital | 2706 |
| Cardiff & Vale | University Hospital Of Wales | 4633 |
| Cardiff & Vale | The Barry Hospital | 4158 |
| Cwm Taf Morgannwg | Princess Of Wales Hospital | 1686 |
| Cwm Taf Morgannwg | The Royal Glamorgan Hospital | 1717 |
| Cwm Taf Morgannwg | Prince Charles Hospital | 1680 |
| Cwm Taf Morgannwg | Ysbyty Cwm Rhondda | 1602 |
| Cwm Taf Morgannwg | Ysbyty Cwm Cynon | 1250 |
| Cwm Taf | The Royal Glamorgan Hospital | 2712 |
| Cwm Taf | Prince Charles Hospital | 2718 |
| Cwm Taf | Aberdare General Hospital | 32 |
| Cwm Taf | Ysbyty Cwm Rhondda | 2610 |
| Cwm Taf | Ysbyty Cwm Cynon | 2652 |
| Aneurin Bevan | Nevill Hall Hospital | 4402 |
| Aneurin Bevan | Royal Gwent Hospital | 4465 |
| Aneurin Bevan | Ysbyty Aneurin Bevan | 4328 |
| Aneurin Bevan | Ysbyty Ystrad Fawr | 4387 |

| Organisation | Hospital_ItemName_ENG | Total |
|---|---|---|
| Aneurin Bevan | The Grange Hospital | 1055 |
| Powys Teaching | Llandrindod Wells Hospital | 4319 |
| Powys Teaching | Victoria Memorial Hospital | 4294 |
| Powys Teaching | Breconshire War Memorial Hospital | 4323 |
| Powys Teaching | Ystradgynlais Community Hospital | 3951 |

```
#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

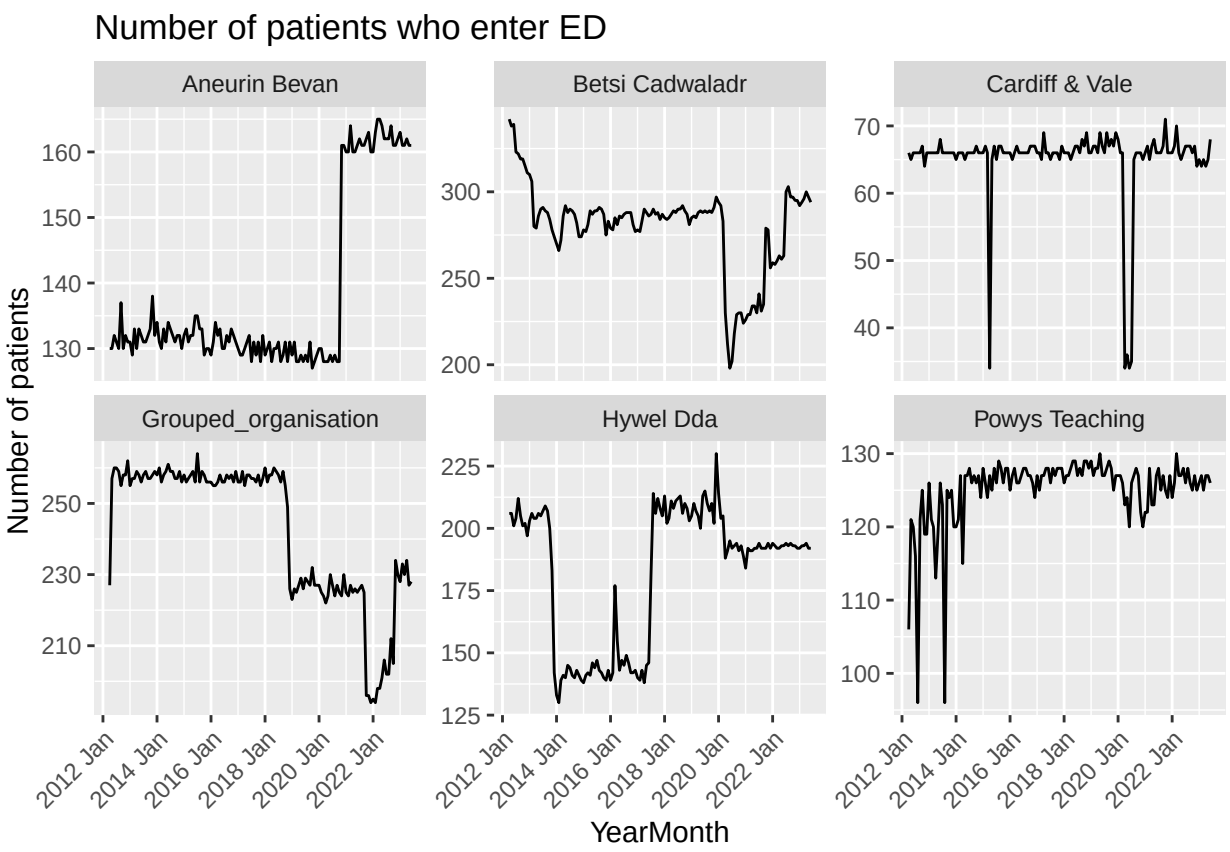
```
data1_grouped <- data1 %>%
  mutate(Grouped_Organisation = case_when(
    Organisation %in% c("Cwm Taf", "Cwm Taf Morgannwg", "Abertawe Bro Morgannwg", "Swansea Bay") ~ "Grou
    TRUE ~ Organisation
  ))
```

```
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))
```

```
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))
```



```r
#library(gt)

#data2_hts %>%
  #gt() %>%
  #tab_header(title = "Number of patients who enter ED") %>%
  #cols_label(
    #YearMonth = "Year/Month",
    #Number = "Number of Patients",
    #Grouped_Organisation = "Organisation Group"
  #)
```

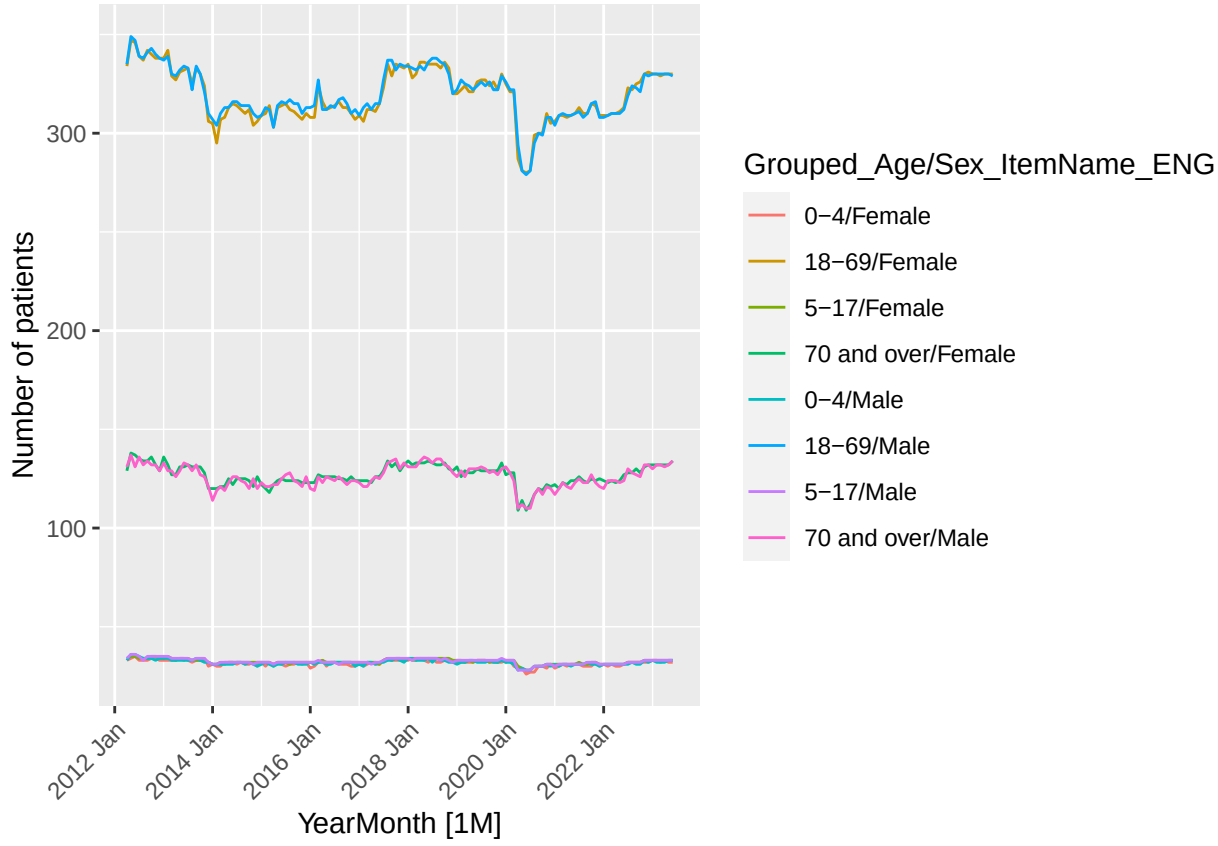
```r
unique(data1_grouped$Age_Code)
```

```
##  [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"
```

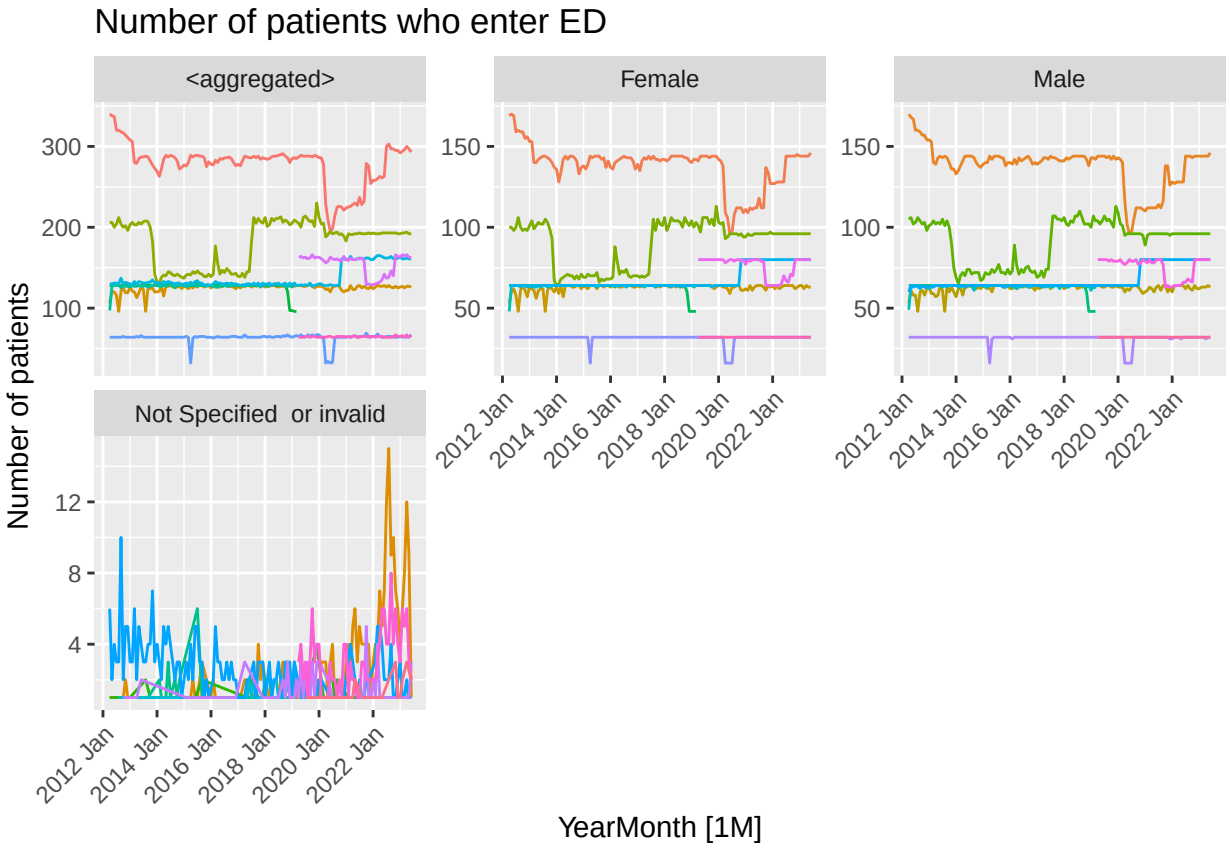
Age group: “0-4”, “5-17”, “18-69”, “70^”

```r
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"
  ))
```

```r
data1_gts <- data1_grouped_age %>%
  filter(!Sex_ItemName_ENG == "Not Specified  or invalid") %>%
  aggregate_key(Grouped_Age* Sex_ItemName_ENG , Number = sum(Number))

data1_gts |>
  filter(!is_aggregated(Sex_ItemName_ENG), !is_aggregated(Grouped_Age)) |>
  autoplot(Number) +
  labs(y = "Number of patients")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Number of patients entering ED, facet by sex

data3_ghts <- data1_grouped_age %>%
  aggregate_key((Grouped_Organisation/Hospital_Hierarchy)* Sex_ItemName_ENG , Number = sum(Number))

data3_ghts |>
  filter(!is_aggregated(Hospital_Hierarchy)) |>
  autoplot(Number) +
  labs(y = "Number of patients",
       title = "Number of patients who enter ED") +
  facet_wrap(vars(Sex_ItemName_ENG), scales = "free_y", ncol = 3) +
  theme(legend.position = "none")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




---

```
data1_full <- data1_grouped_age |>
  aggregate_key(Grouped_Organisation/Hospital_ItemName_ENG, Number = sum(Number))

data1_full$key_combined <- paste(data1_full$Grouped_Organisation, data1_full$Hospital_ItemName_ENG, sep
```

```
library(lubridate)


data1_wide <- data1_full %>%
   pivot_wider(names_from = key_combined, values_from = Number)
```

```
library(hts)

reconcile_data1_wide_paths <- function(data1_full) {

  # Convert to regular data frame to bypass tsibble constraints
  data1_full <- as.data.frame(data1_full)

  # Convert YearMonth to date
  data1_wide <- data1_full %>%
    unite("key_column", Grouped_Organisation, Hospital_ItemName_ENG, sep = "_") %>%
    pivot_wider(names_from = key_column, values_from = Number, values_fill = list(Number = 0))
```

```
  # Convert the data to a gts object
  ts_data <- ts(data1_wide[,-1], frequency = 12, start = c(year(min(ymd(paste0(data1_wide$YearMonth, "-
  gts_obj <- hts(ts_data)

  return(gts_obj)
}

# Calling the function
gts_obj = reconcile_data1_wide_paths(data1_full)
```

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

```
# First, forecast each individual series with ets
individual_forecasts <- lapply(1:ncol(gts_obj[[1]]), function(i) {
  ets_forecast <- forecast(ets(gts_obj[[1]][,i]), h = 6)
  return(ets_forecast$mean)
})

# Combine the forecasts into a matrix
forecast_matrix <- do.call(cbind, individual_forecasts)

# Now, reconcile the forecasts using hts
gts_forecast_matrix <- ts(forecast_matrix, frequency = 12)
gts_forecasts <- hts(gts_forecast_matrix)
```

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

```
reconciled_forecasts <- forecast(gts_forecasts, method = "bu")

# Print the reconciled forecasts
print(reconciled_forecasts)
```

```
## Hierarchical Time Series
## 2 Levels
## Number of nodes at each level: 1 50
## Total number of series: 51
## Number of observations in each historical series: 6
## Number of forecasts per series: 24
## Top level series of forecasts:
##        Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1                                                        81.52277 81.59813
## 2 81.97494 82.05030 82.12566 82.20102 82.27639 82.35175 82.42711 82.50247
## 3 82.87928 82.95464 83.03000 83.10536 83.18072 83.25609
##        Sep      Oct      Nov      Dec
## 1 81.67349 81.74885 81.82422 81.89958
## 2 82.57783 82.65319 82.72856 82.80392
## 3
```

Forecasting Each Series with ets:

```
individual_forecasts <- lapply(1:ncol(gts_obj[[1]]), function(i) {
  ets_forecast <- forecast(ets(gts_obj[[1]][,i]), h = 6)
  return(ets_forecast$mean)
})
```

Combine the Forecasts into a Matrix

```
forecast_matrix <- do.call(cbind, individual_forecasts)
```

Convert to HTS Structure:

```
gts_forecast_matrix <- ts(forecast_matrix, frequency = 12)
gts_forecasts <- hts(gts_forecast_matrix)
```

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

Reconcile the Forecasts:

```
reconciled_forecasts <- forecast(gts_forecasts, method = "bu", h = 6)
```

```
print(reconciled_forecasts)
```

```
## Hierarchical Time Series
## 2 Levels
## Number of nodes at each level: 1 50
## Total number of series: 51
## Number of observations in each historical series: 6
## Number of forecasts per series: 6
## Top level series of forecasts:
##        Jul      Aug      Sep      Oct      Nov      Dec
## 1 81.52277 81.59813 81.67349 81.74885 81.82422 81.89958
```

```
library(dplyr)
library(lubridate)

# Decide how many months you want to hold out for testing
h <- 6

# Split the data into training and test set
data1_train <- data1_full %>%
  filter(as.Date(YearMonth) < max(as.Date(YearMonth)) - months(h))

data1_test <- data1_full %>%
  filter(as.Date(YearMonth) >= max(as.Date(YearMonth)) - months(h))
```

```
str(reconciled_forecasts)
```

```
## List of 6
##  $ bts    : Time-Series [1:6, 1:50] from 1.5 to 1.92: 49.1 49.1 49.2 49.3 49.4 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : NULL
##   .. ..$ : chr [1:50] "structure(c(48.5891258616791, 48.6690226982572, 48.7489195348354, " "structure
##  $ histy  : Time-Series [1:6, 1:50] from 1 to 1.42: 48.6 48.7 48.7 48.8 48.9 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:50] "structure(c(48.5891258616791, 48.6690226982572, 48.7489195348354, " "structure
##  $ labels :List of 2
##   ..$ Level 0: chr "Total"
##   ..$ Level 1: chr [1:50] "structure(c(48.5891258616791, 48.6690226982572, 48.7489195348354, " "stru
##  $ method : chr "bu"
##  $ fmethod: chr "ets"
##  $ nodes  :List of 1
##   ..$ Level 1: int 50
##  - attr(*, "class")= chr [1:2] "hts" "gts"
```

Accessing the base forecasts:

```
base_forecasts <- reconciled_forecasts$bts
```

Accessing historical data:

```
historical_data <- reconciled_forecasts$histy
```

```
subset_data <- data1_test[, -c(1, which(names(data1_test) == "Grouped_Organisation"))]
```

```
sapply(subset_data, class)
```

```
## $Hospital_ItemName_ENG
## [1] "agg_vec"    "vctrs_rcrd" "vctrs_vctr"
##
## $Number
## [1] "numeric"
##
## $key_combined
## [1] "character"
```

```
sapply(subset_data, function(col) unique(class(col)))
```

```
## $Hospital_ItemName_ENG
## [1] "agg_vec"    "vctrs_rcrd" "vctrs_vctr"
##
## $Number
## [1] "numeric"
##
## $key_combined
## [1] "character"
```

```
subset_df <- as.data.frame(subset_data)
test_data_matrix <- matrix(subset_data$Number, nrow = nrow(subset_data))
```

```r
forecast_values <- matrix(reconciled_forecasts$bts, nrow = nrow(test_data_matrix))
```

```
## Warning in matrix(reconciled_forecasts$bts, nrow = nrow(test_data_matrix)):
## data length [300] is not a sub-multiple or multiple of the number of rows [282]
```

```r
# Ensure that the dimensions of the test data and forecast values match:
if(dim(test_data_matrix)[1] == dim(forecast_values)[1] & dim(test_data_matrix)[2] == dim(forecast_values
    accuracy_results <- accuracy(forecast_values, test_data_matrix)
    print(accuracy_results)
} else {
    cat("The dimensions of test data and forecast values do not match.")
}
```

```
## The dimensions of test data and forecast values do not match.
```

```r
if (nrow(test_data_matrix) > nrow(forecast_values)) {
    test_data_matrix <- test_data_matrix[1:nrow(forecast_values), ]
}
```

```r
# Convert this matrix into a time series object
test_data_ts <- ts(test_data_matrix, frequency = 12)


forecast_ts <- ts(forecast_values, start=start(test_data_ts), frequency=12)

# Calculate the Accuracy:
acc <- accuracy(forecast_ts, test_data_ts)

print(acc)
```

```
##                 ME    RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 77.98984 180.497 77.98984 97.55007 97.55007 0.8357321  1.127158
```