forecasting ED

Janice Hsu

2023-08-28

Preparation

```
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
```

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

```
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
```

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

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

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

```
# 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
```

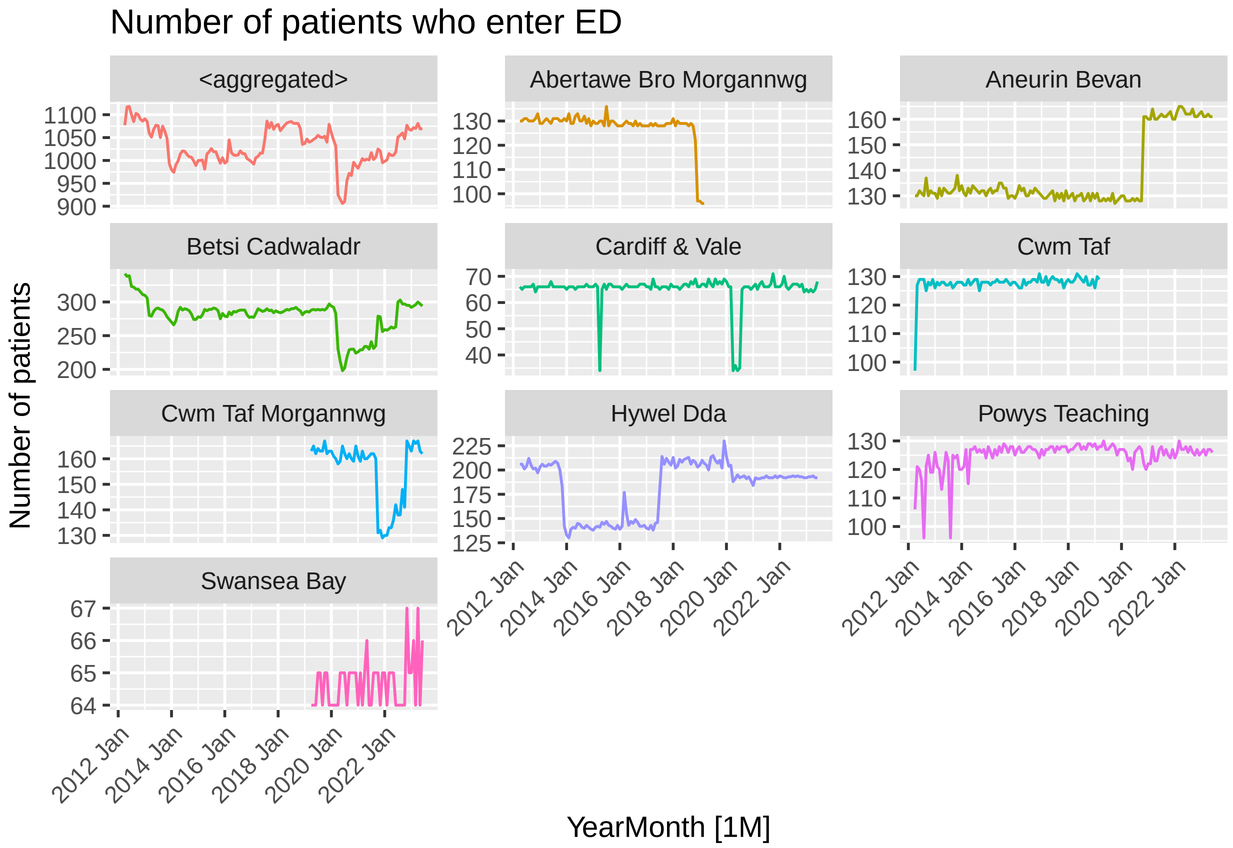
```r
setDT(data2)

# 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

Organisation	Hospital_ItemName_ENG	Total
Aneurin Bevan	Nevill Hall Hospital	4402
Aneurin Bevan	Royal Gwent Hospital	4465
Aneurin Bevan	Ysbyty Aneurin Bevan	4328
Aneurin Bevan	Ysbyty Ystrad Fawr	4387
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

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

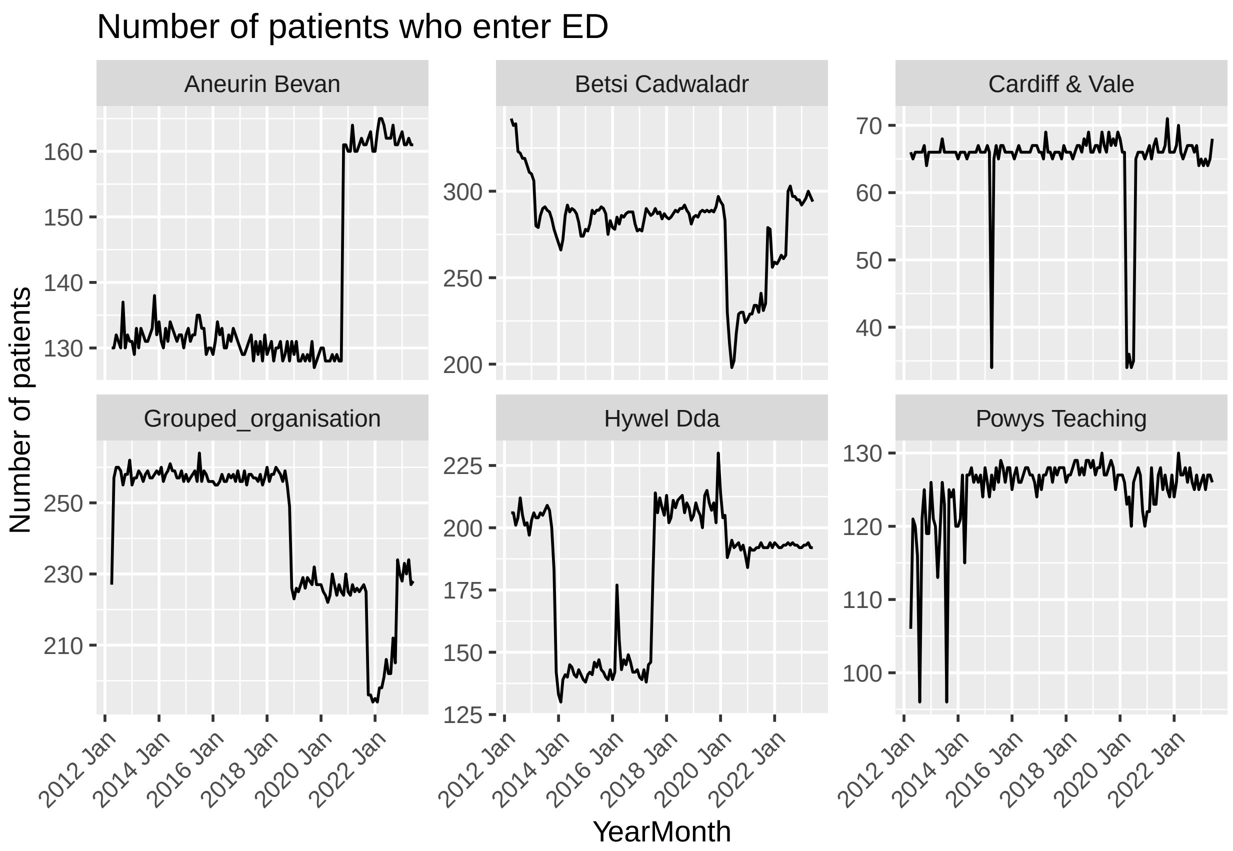
```
unique(data1_grouped$Grouped_Organisation)
```

```
## [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))
```



```
#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"
  #)
```

Change the Age_Code structure into different groups

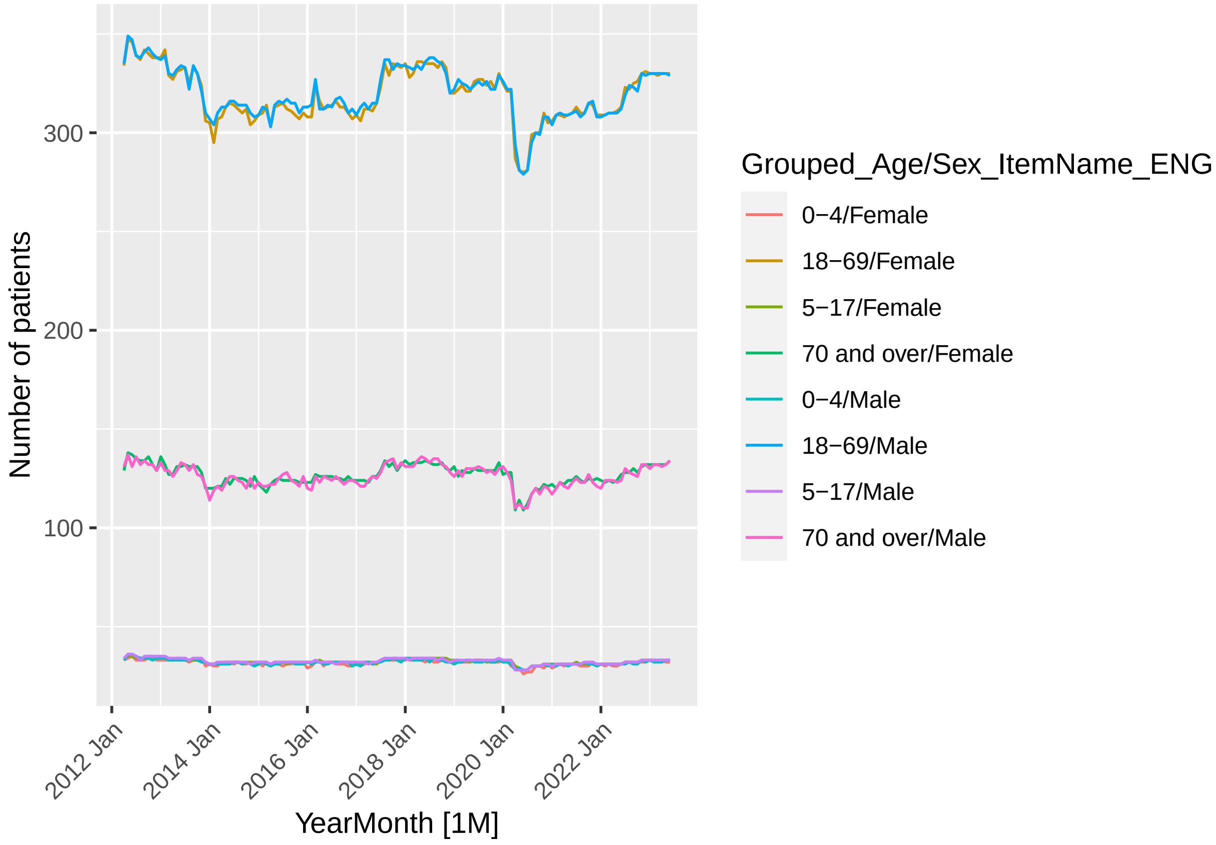
```
unique(data1_grouped$Age_Code)
```

```
##  [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^”

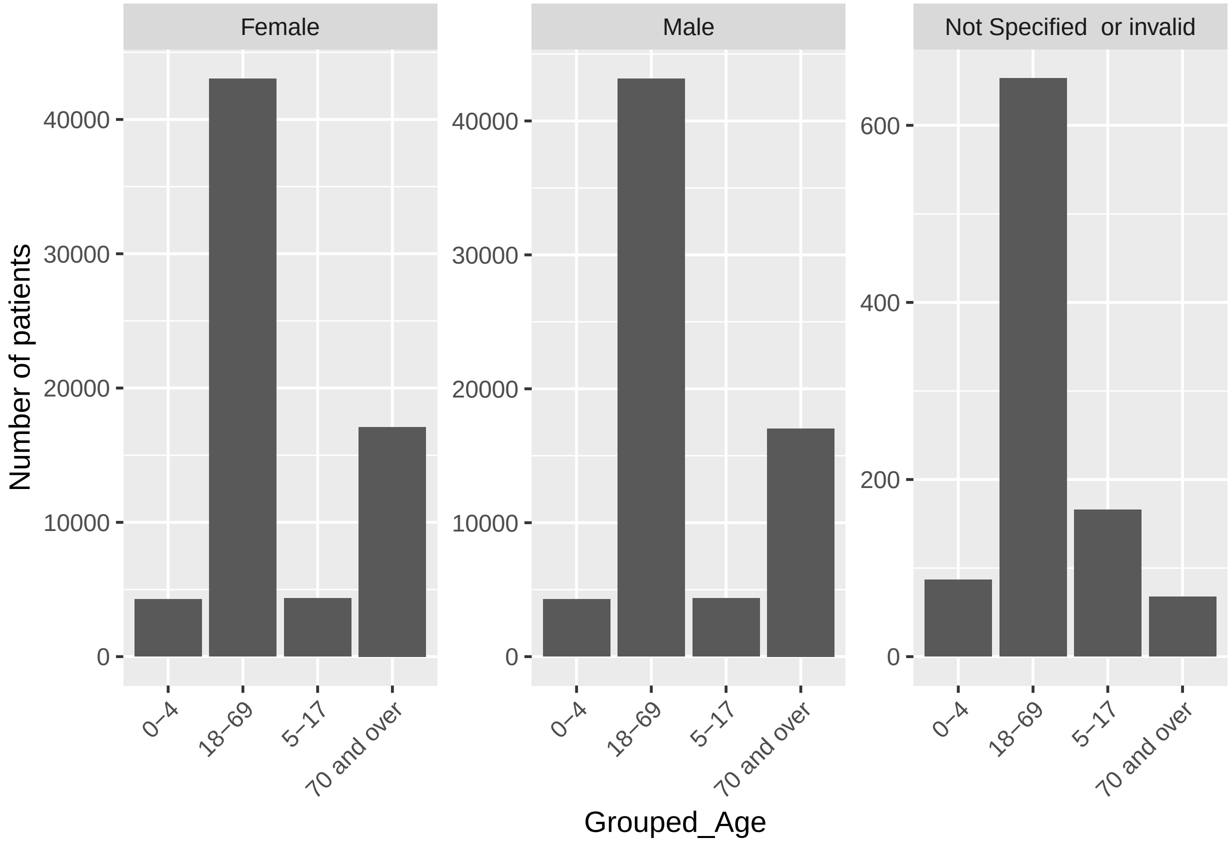
```r
data1_grouped_age <- data1_grouped %>%
  filter(Age_Code != "Unknown") %>%
  mutate(Grouped_Age = case_when(
    Age_Code == "0 to 4" ~ "0-4",
    Age_Code == "5 to 17" ~ "5-17",
    Age_Code %in% c("18 to 24", "25 to 29", "30 to 34", "35 to 39",
                    "40 to 44", "45 to 49", "50 to 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))
```

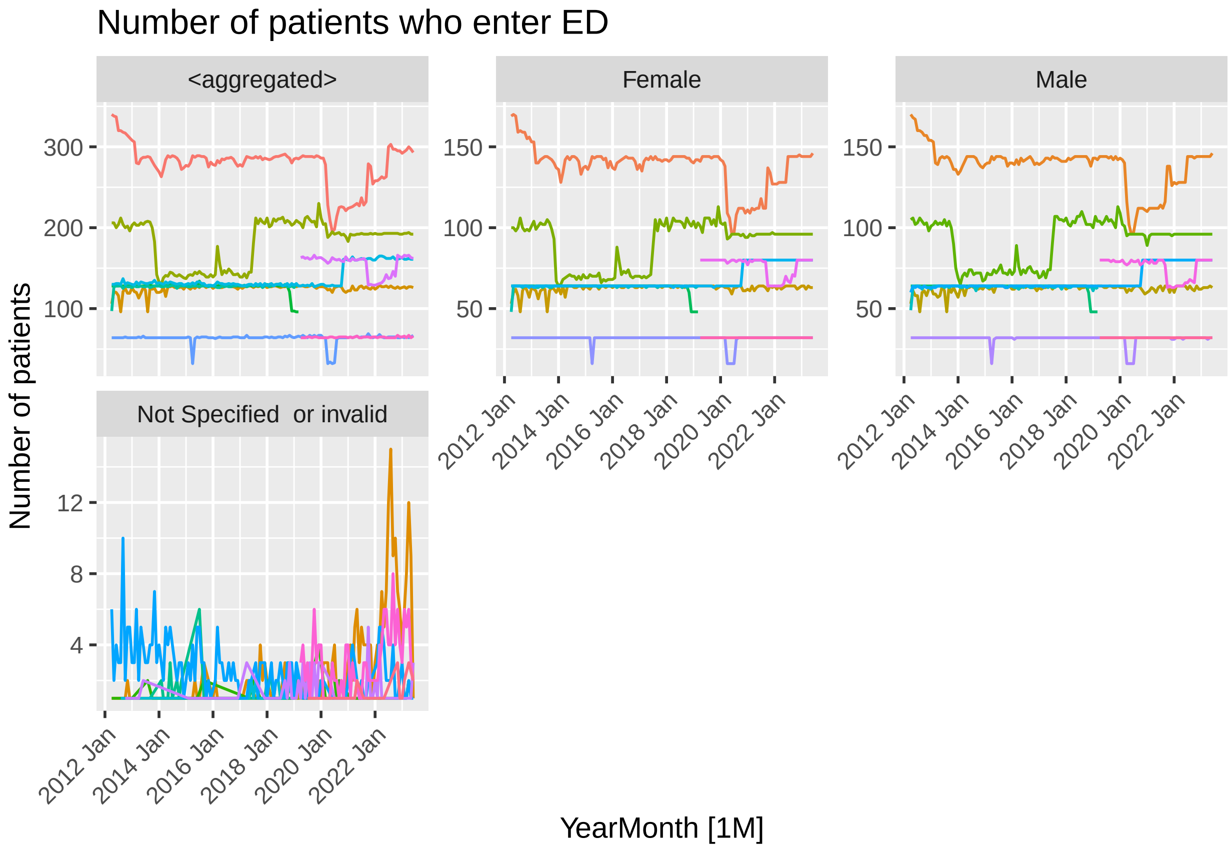


```r
library(dplyr)

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))
```

```r
library(ggplot2)

p <- 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")

print(p)
```



```r
# 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))
```



Forecast

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

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

```
library(lubridate)

data1_wide <- data1_full %>%
 # select(-Grouped_Organisation, -Hospital_ItemName_ENG) %>%
   pivot_wider(names_from = key_combined, values_from = Number)
#select does not work here, I reckon it is because of the aggregate function
```

use the reconcile function (however the gts turns out to be unexpected)

```
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)

  # Filter data to only include records up to 2028
  data1_full <- data1_full %>%
    filter(year(ymd(paste0(YearMonth, "-01"))) <= 2028)

  # 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, "-0
  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.
```

Forecast with ets

```
# 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)

# 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(reconciled_forecasts)
```

```
## Hierarchical Time Series
## 2 Levels
## Number of nodes at each level: 1 43
## Total number of series: 44
## 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
## 1                                                              80.71931
## 2  86.29176  87.22051  88.14925  89.07799  90.00674  90.93548  91.86422
## 3  97.43668  98.36542  99.29417 100.22291 101.15165 102.08040
##         Aug       Sep       Oct       Nov       Dec
## 1  81.64805  82.57679  83.50554  84.43428  85.36302
## 2  92.79297  93.72171  94.65045  95.57919  96.50794
## 3
```

Another try 1

Forecasting Each Series with ets:

```r
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

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

Convert to HTS Structure:

```r
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:

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

```r
print(reconciled_forecasts)
```

```
## Hierarchical Time Series
## 2 Levels
## Number of nodes at each level: 1 43
## Total number of series: 44
## 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 80.71931 81.64805 82.57679 83.50554 84.43428 85.36302
```

Compute accuracy

```r
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))
```

Accessing the base forecasts:(just to check)

```r
# base_forecasts <- reconciled_forecasts$bts
```

Accessing historical data: (just to check)

```r
# historical_data <- reconciled_forecasts$histy
```

Subset the data:

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

Turn the data into matrix:

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

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

```r
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))
```

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

```
## Warning in matrix(reconciled_forecasts$bts, nrow = nrow(test_data_matrix)):
## data length [258] is not a sub-multiple or multiple of the number of rows [233]
```

```
# 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.
```

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

Compute the accuracy metrics

```
# 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 30.09884 31.26194 31.05398 93.17353 96.13245 0.7282114  1.099215
```

Another try for forecasting 2

```r
library(dplyr)
library(tidyr)
library(hts)
library(lubridate)

# 1. Extracting and converting time information
time_info <- data1_full$YearMonth
date_converted <- as.Date(ymd(paste0(time_info, "-01")))

# 2. Convert data to wide format without YearMonth column
data1_wide <- data1_full %>%
  select(-YearMonth) %>%
  pivot_wider(names_from = key_combined, values_from = Number)

# 3. Convert to a matrix and ensure all values are numeric
data1_wide_numeric <- data1_wide[, sapply(data1_wide, is.numeric)]
data_matrix <- as.matrix(data1_wide_numeric)



# 4. Generate hierarchical time series
hierarchical_ts <- hts(data_matrix)
```

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

```r
# 5. Forecast using the "bu" method
forecast_results <- forecast(hierarchical_ts, h = 6, method = "bu")
```

```
## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series

## Warning in ets(x, lambda = lambda, ...): Missing values encountered. Using
## longest contiguous portion of time series
```

```r
# 6. Generate the new forecast dates
new_dates <- seq.Date(max(date_converted), by = "1 month", length.out = 7)[-1]

# 7. Extracting the forecasted values
```

```r
forecasted_values <- tail(forecast_results[[2]], 6)

# 8. Structure the forecasted data
forecast_df <- data.frame(
  YearMonth = rep(new_dates, times = ncol(forecasted_values)),
  key_combined = rep(colnames(forecasted_values), each = 6),
  Number = as.vector(t(forecasted_values))
)

# 9. Modify the separation process to handle unexpected underscores
forecast_df <- forecast_df %>%
  separate(key_combined,
           into = c("Grouped_Organisation", "Hospital_ItemName_ENG"),
           sep = "_(?=[^_]+$)")

# 10. Combine the historical and forecasted data
historical_df <- data1_full %>%
  select(YearMonth, Grouped_Organisation, Hospital_ItemName_ENG, Number)

combined_data <- dplyr::bind_rows(historical_df, forecast_df)
```

Use non-hierarchical forecasting

```r
summed_data <- data1_full %>%
  group_by(Grouped_Organisation) %>%
  summarise(Total_Number = sum(Number))
```

```r
# Convert summed_data explicitly to a data.frame
summed_data <- data.frame(summed_data)

results <- list()

# Extract unique organisations directly from summed_data
unique_organisations <- unique(summed_data$Grouped_Organisation)

for(org in unique_organisations) {

  # Explicitly match the rows
  single_org_data <- summed_data[which(summed_data$Grouped_Organisation %in% org), ]

  cat("\nProcessing:", org, "\n")
  cat("Number of rows:", nrow(single_org_data), "\n")

  # Only continue if there's data available
  if(nrow(single_org_data) > 0) {
    # Convert to time series
    ts_data <- ts(single_org_data$Total_Number, frequency = 12) # monthly data

    # Forecast
    forecasted <- forecast(auto.arima(ts_data), h=6)

    # Store the forecast in the results
    results[[org]] <- forecasted
  } else {
    warning(paste("No data available for", org))
  }
}
```

```
##
## Processing: Aneurin Bevan Betsi Cadwaladr Cardiff & Vale Grouped_organisation Hywel Dda Powys Teachin
## Number of rows: 0

## Warning: No data available for Aneurin BevanNo data available for Betsi
## CadwaladrNo data available for Cardiff & ValeNo data available for
## Grouped_organisationNo data available for Hywel DdaNo data available for Powys
## Teaching

##
## Processing: FALSE FALSE FALSE FALSE FALSE FALSE
## Number of rows: 0

## Warning: No data available for FALSENo data available for FALSENo data
## available for FALSENo data available for FALSENo data available for FALSENo
## data available for FALSE
```

```
# Display the results
results
```

```
## list()
```

```
summed_data$Date <- as.Date(summed_data$YearMonth)
```

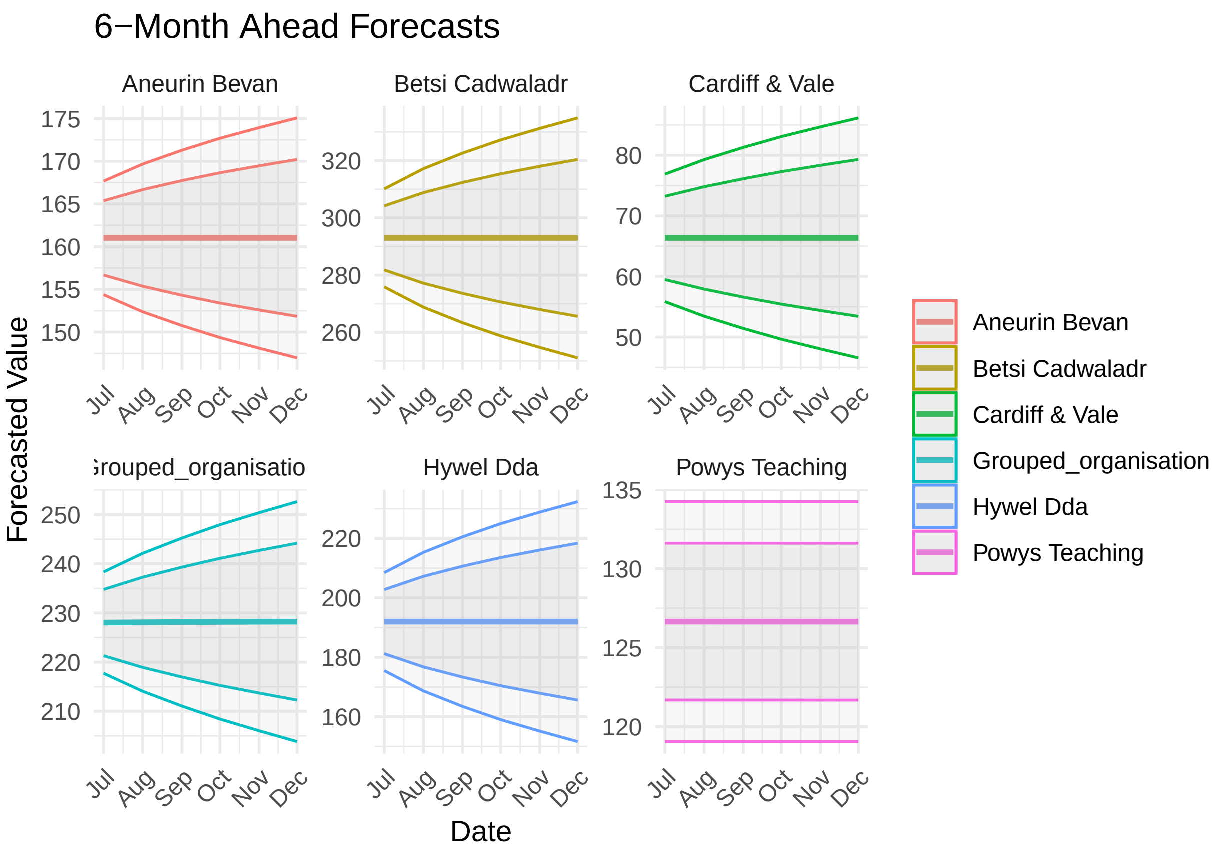
```
forecast_results_list <- summed_data %>%
  group_by(Grouped_Organisation) %>%
  do({
    # Convert to ts object
    time_series_data <- ts(.$Total_Number, frequency = 12, start = c(year(min(.$Date)), month(min(.$Dat

    # Fit ETS model
    fit <- tryCatch(ets(time_series_data), error = function(e) NULL)

    # Forecast 6 months ahead if fit is successful
    if (!is.null(fit)) {
      forecast_obj <- forecast(fit, h = 6)
      data.frame(
        Grouped_Organisation = rep(unique(.$Grouped_Organisation), times = 6),
        Forecast_Date = seq.Date(from = max(.$Date) + months(1), by = "month", length.out = 6),
        Forecasted_Value = forecast_obj$mean,
        Lower_80 = forecast_obj$lower[,1],
        Upper_80 = forecast_obj$upper[,1],
        Lower_95 = forecast_obj$lower[,2],
        Upper_95 = forecast_obj$upper[,2]
      )
    } else {
      data.frame()  # Empty data frame for organizations that couldn't be forecasted
    }
  })

# Convert to a more familiar data frame structure
forecast_df <- bind_rows(forecast_results_list)
```

use of rep(unique(.$Grouped_Organisation), times = 6) to ensure that the Grouped_Organisation column has the same number of rows as the forecasted values.

```
forecast_df$Grouped_Organisation <- as.character(forecast_df$Grouped_Organisation)
forecast_df$Grouped_Organisation <- factor(forecast_df$Grouped_Organisation)
```

```
library(ggplot2)

ggplot(forecast_df, aes(x = Forecast_Date, y = Forecasted_Value, group = Grouped_Organisation, color = G
  geom_line(size = 1) +

  # Highlighting the 80% confidence interval
  geom_ribbon(aes(ymin = Lower_80, ymax = Upper_80), alpha = 0.2, fill = "grey") +

  # Highlighting the 95% confidence interval
  geom_ribbon(aes(ymin = Lower_95, ymax = Upper_95), alpha = 0.1, fill = "grey") +
  labs(title = "6-Month Ahead Forecasts",
       y = "Forecasted Value",
       x = "Date",
       color = "Organisation") +
  theme_minimal() +
  theme(legend.title = element_blank()) +
  facet_wrap(~Grouped_Organisation, scales = "free", ncol = 3)+
  theme(legend.title = element_blank(),
        axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


6−Month Ahead Forecasts
Aneurin Bevan
Betsi Cadwaladr
Cardiff & Vale
Grouped_organisatio
Hywel Dda
Powys Teaching
Forecasted Value
Date
175
170
165
160
155
150
320
300
280
260
80
70
60
50
250
240
230
220
210
220
200
180
160
135
130
125
120
Jul
Aug
Sep
Oct
Nov
Dec
Aneurin Bevan
Betsi Cadwaladr
Cardiff & Vale
Grouped_organisation
Hywel Dda
Powys Teaching