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
```

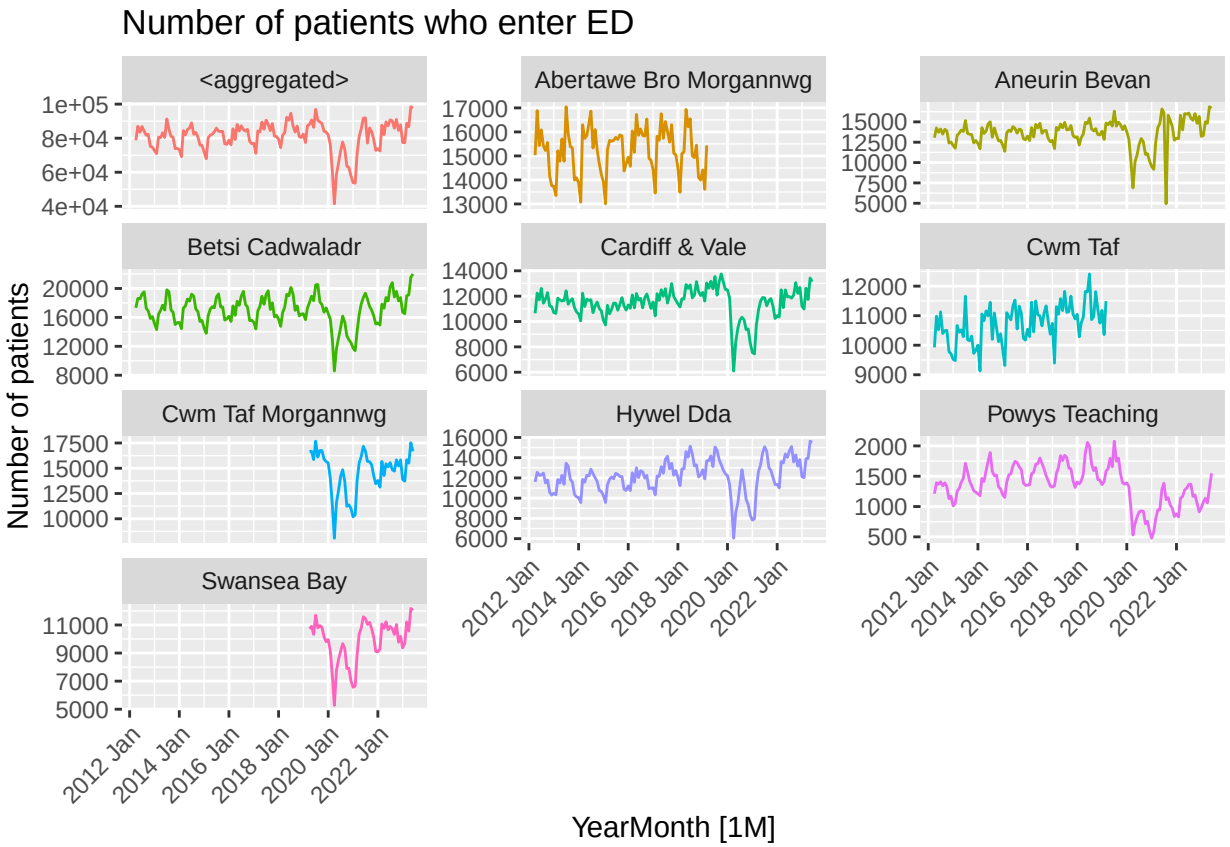
```r
library(dplyr)
library(tidyr)
library(forecast)
```

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

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

#Number of patients entering ED under different hospital hierarchy

```
data_hts <- data %>%
  aggregate_key(Organisation/Hospital_ItemName_ENG, Number = sum(Data))

data_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),
        axis.text.y = element_text(angle = 0, 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

```
data_grouped <- data %>%
  mutate(Aggregated_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

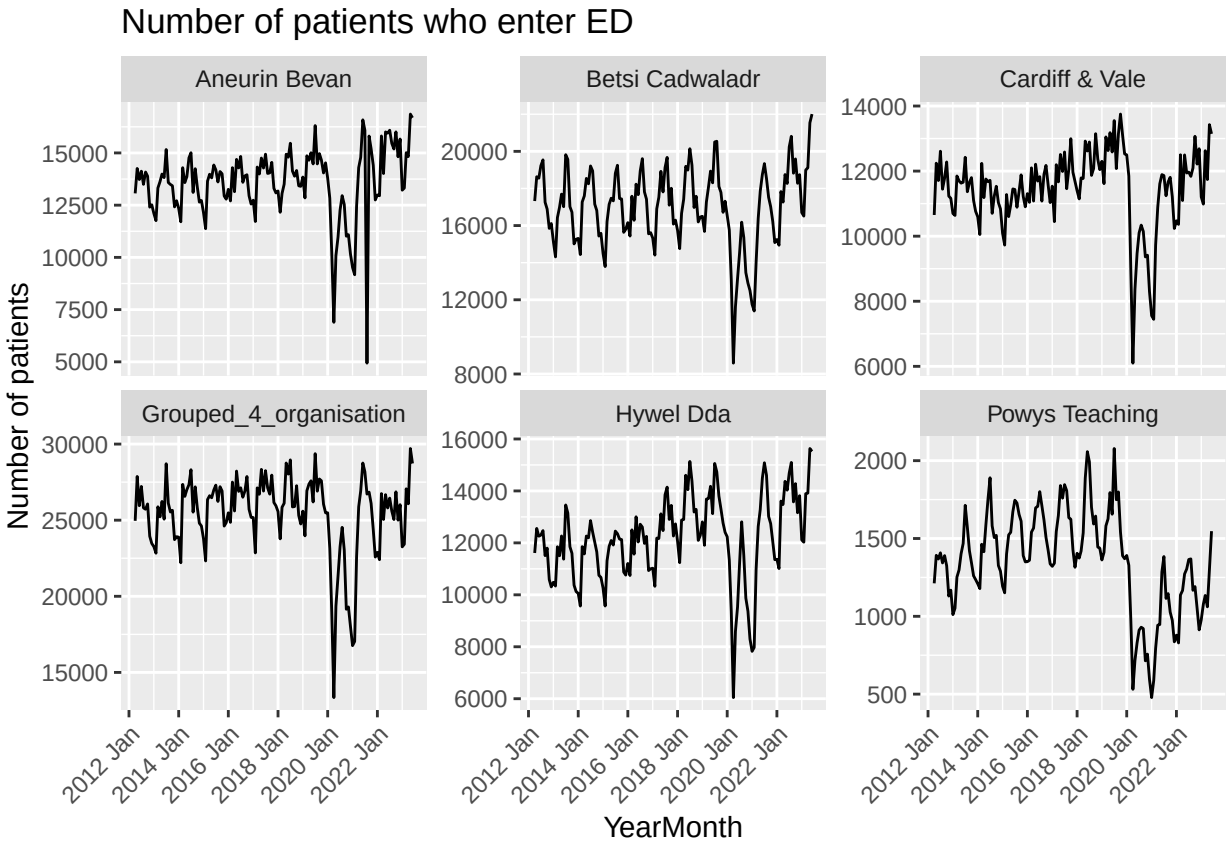
```
unique(data_grouped$Aggregated_Organisation)
```

```
## [1] "Betsi Cadwaladr"        "Hywel Dda"              "Grouped_4_organisation"
## [4] "Cardiff & Vale"         "Aneurin Bevan"          "Powys Teaching"
```

```
data2_hts <- data_grouped %>%
  group_by(Aggregated_Organisation) %>%
  summarise(Number = sum(Data))
```

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(Aggregated_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(data_grouped$Age_Code)
```

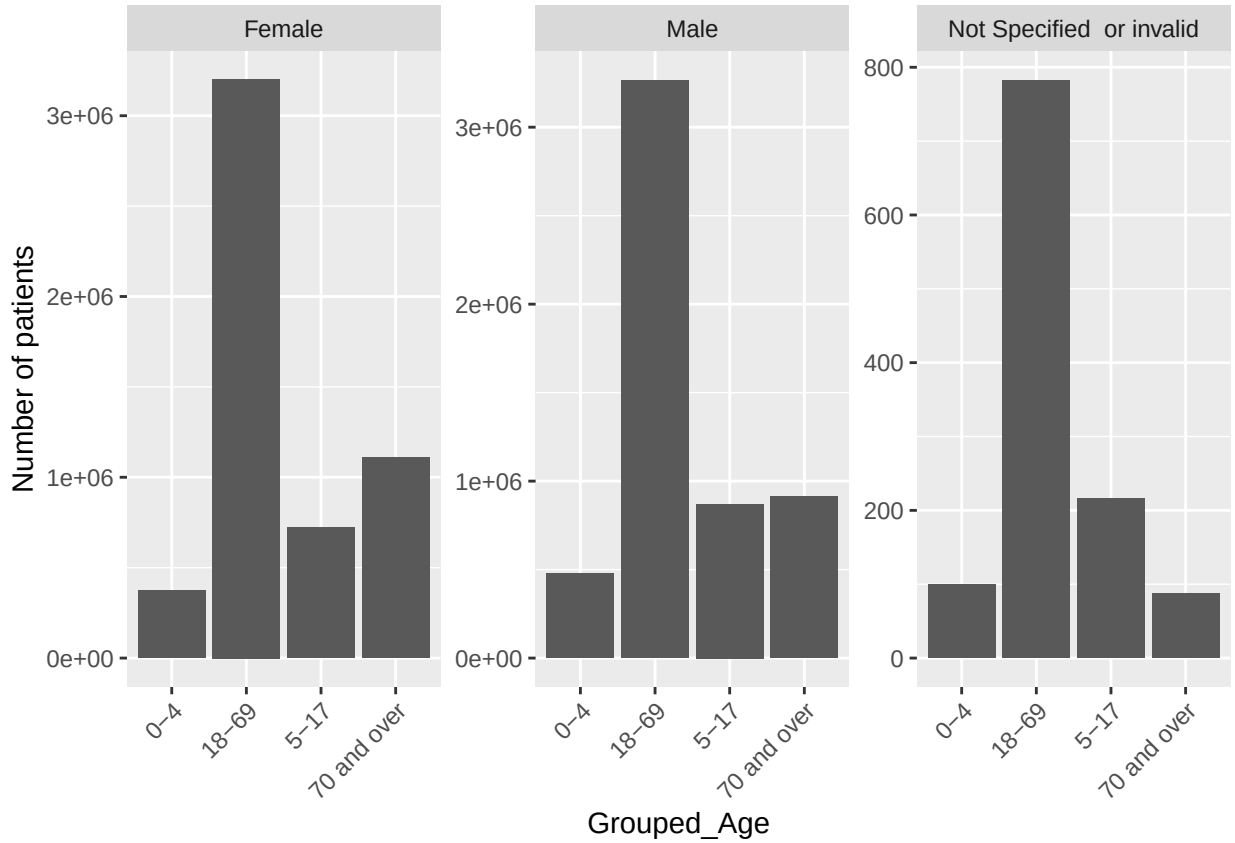
```
##  [1] "0 to 4"   "18 to 24" "25 to 29" "30 to 34" "35 to 39" "40 to 44"
##  [7] "45 to 49" "5 to 17"  "50 to 54" "55 to 59" "60 to 64" "65 to 69"
## [13] "70 to 74" "75 to 79" "80 to 84" "85"       "Unknown"
```

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

```
data_grouped_age <- data_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

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

ggplot(data_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 <- data_grouped %>%
  group_by(Aggregated_Organisation) %>%
  index_by(YearMonth) %>%
  summarise(Number = sum(Data)) %>%
  pivot_wider(names_from = Aggregated_Organisation, values_from = Number)
```

```
data2_wide <- as_tibble(data2_wide)
data2_wide <- data2_wide %>%
   mutate(Total = rowSums(select(., c("Aneurin Bevan", "Betsi Cadwaladr", "Cardiff & Vale", "Grouped_4_
```

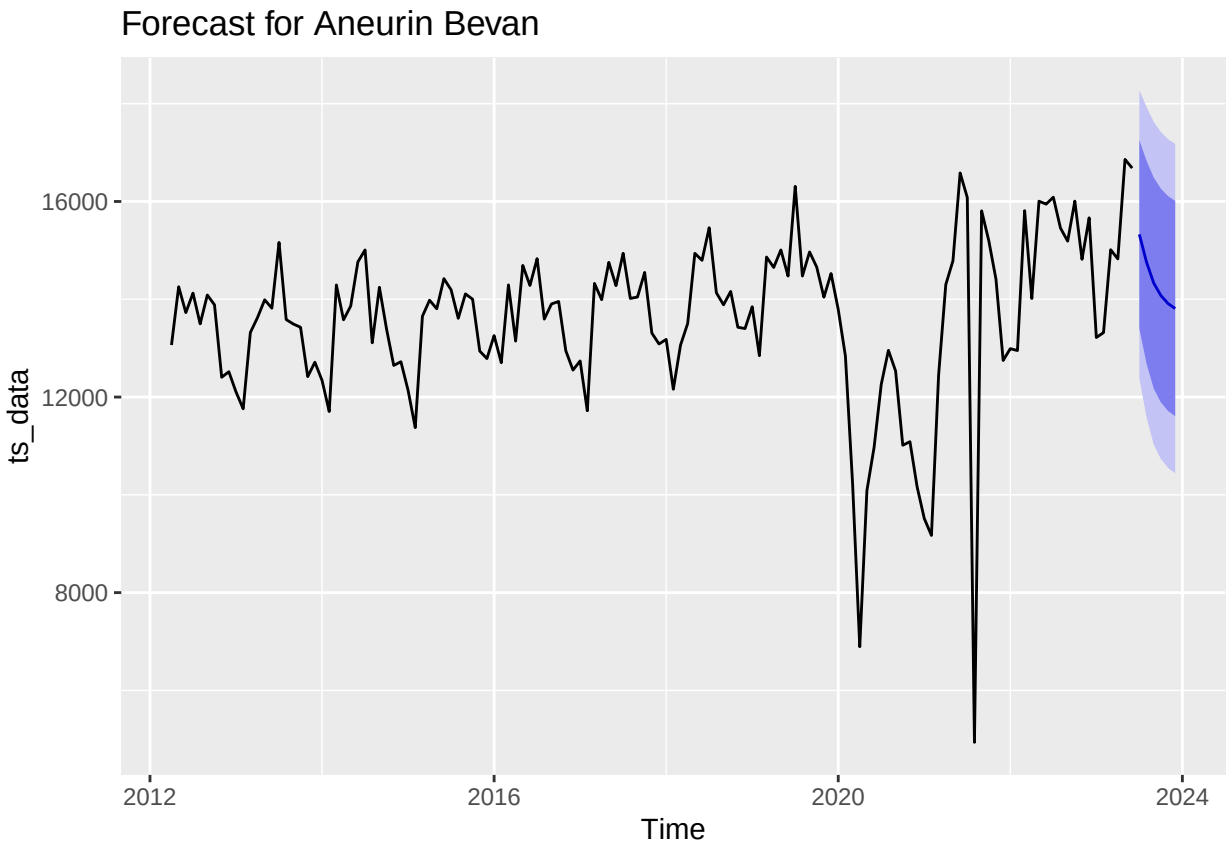
```r
library(forecast)

# Convert data into time series
start_year <- year(min(data2_wide$YearMonth))
start_month <- month(min(data2_wide$YearMonth))

# Convert each column to a ts object
cols_to_forecast <- c("Aneurin Bevan", "Betsi Cadwaladr", "Cardiff & Vale", "Grouped_4_organisation", "
forecast_list <- list()

for (col in cols_to_forecast) {
  ts_data <- ts(data2_wide[[col]], start = c(start_year, start_month), frequency = 12)

  # Forecast using auto.arima
  forecast_model <- auto.arima(ts_data)
  forecast_list[[col]] <- forecast(forecast_model, h = 6)
}
```

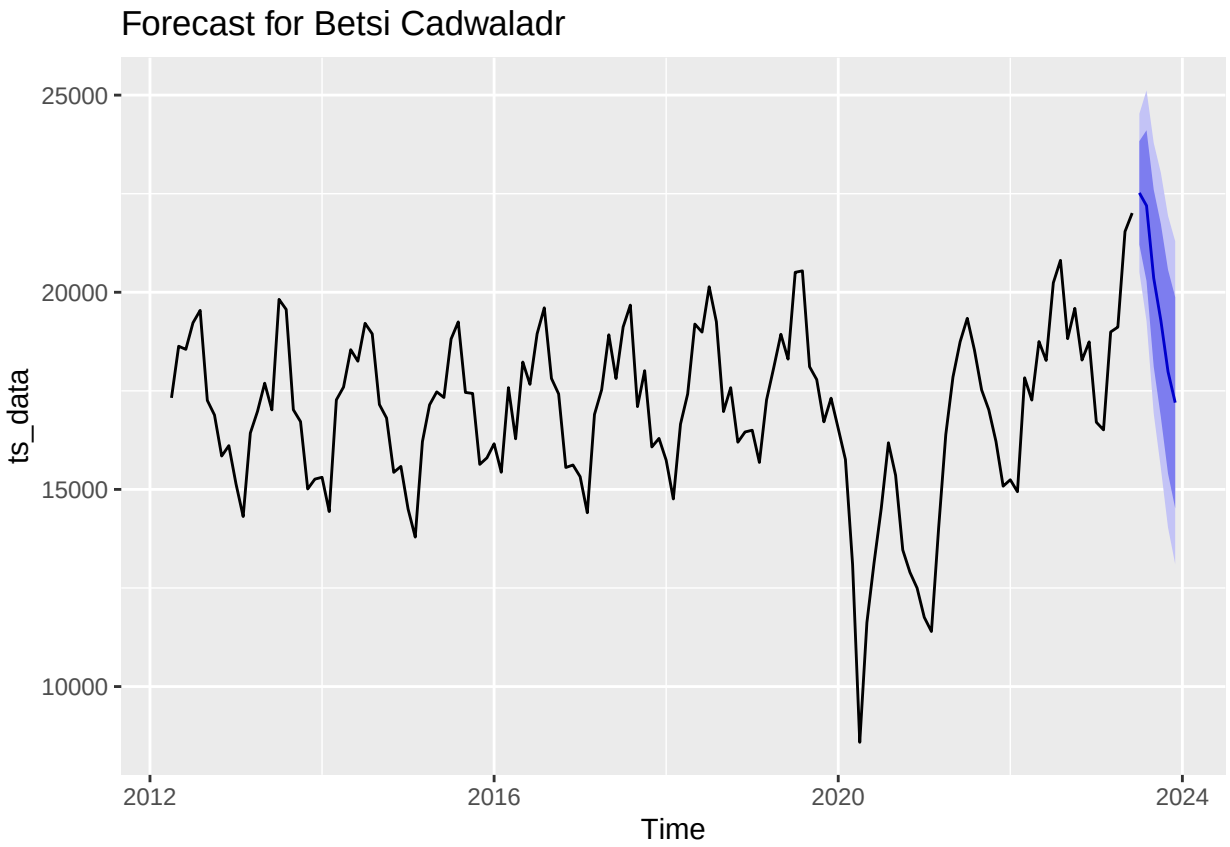
```r
print(forecast_list)
```

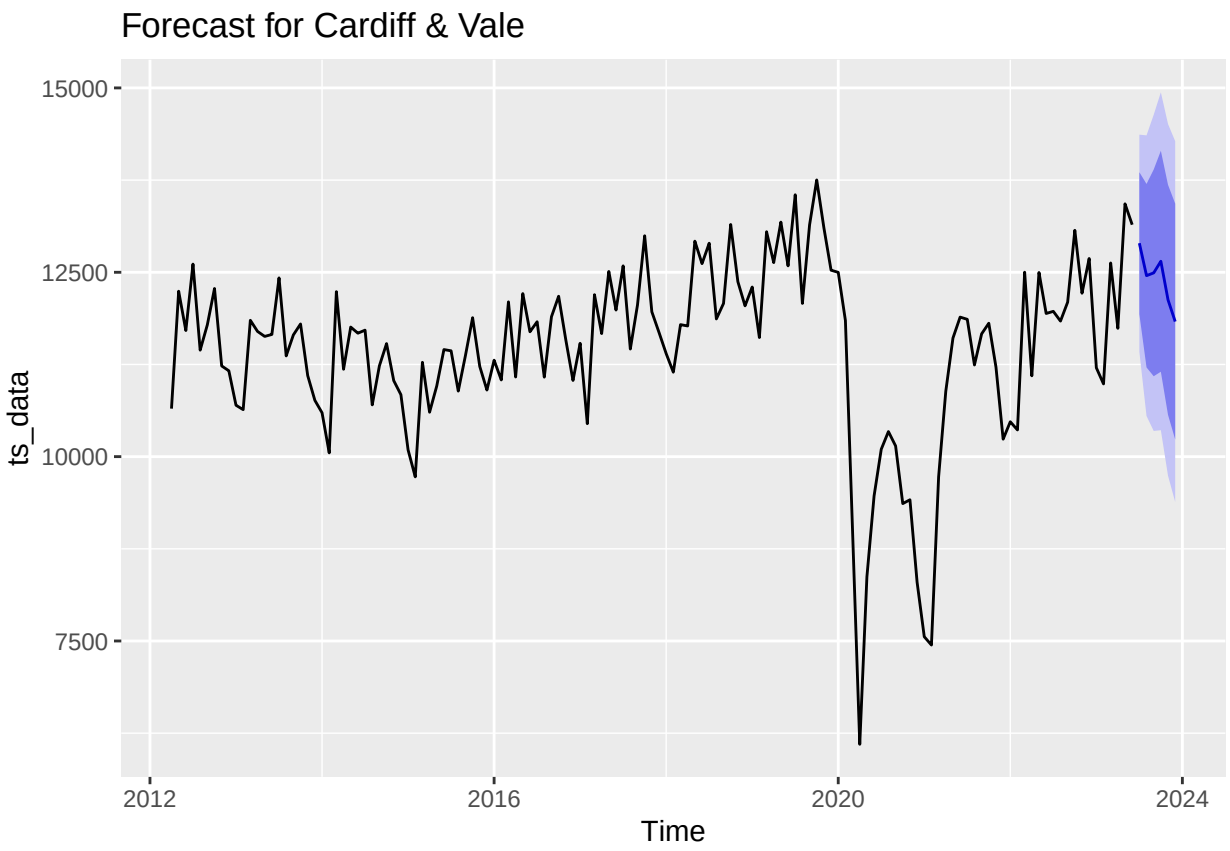
```
## $`Aneurin Bevan`
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jul 2023       15327.46 13401.32 17253.59 12381.69 18273.22
## Aug 2023       14758.62 12683.77 16833.47 11585.41 17931.83
## Sep 2023       14331.89 12174.36 16489.43 11032.23 17631.56
## Oct 2023       14077.53 11891.77 16263.28 10734.70 17420.35
## Nov 2023       13913.04 11715.52 16110.55 10552.23 17273.84
## Dec 2023       13809.83 11607.72 16011.95 10441.99 17177.68
##
## $`Betsi Cadwaladr`
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jul 2023       22520.72 21208.70 23832.74 20514.16 24527.28
## Aug 2023       22195.49 20282.25 24108.74 19269.44 25121.55
## Sep 2023       20359.95 18117.18 22602.73 16929.92 23789.98
## Oct 2023       19284.65 16835.70 21733.59 15539.31 23029.99
## Nov 2023       17985.58 15400.91 20570.24 14032.67 21938.49
## Dec 2023       17199.82 14523.44 19876.20 13106.64 21293.00
##
## $`Cardiff & Vale`
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Jul 2023       12894.49 11932.17 13856.81 11422.743 14366.23
## Aug 2023       12453.99 11210.08 13697.89 10551.600 14356.38
## Sep 2023       12491.75 11090.31 13893.20 10348.425 14635.08
## Oct 2023       12649.40 11151.53 14147.27 10358.603 14940.20
## Nov 2023       12125.90 10566.68 13685.12  9741.274 14510.52
## Dec 2023       11831.67 10232.62 13430.73  9386.128 14277.22
##
## $Grouped_4_organisation
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jul 2023       28201.58 26101.77 30301.39 24990.20 31412.96
## Aug 2023       27165.96 24527.91 29804.02 23131.41 31200.52
## Sep 2023       26742.12 23837.94 29646.29 22300.56 31183.67
## Oct 2023       26782.76 23735.25 29830.27 22122.00 31443.52
```

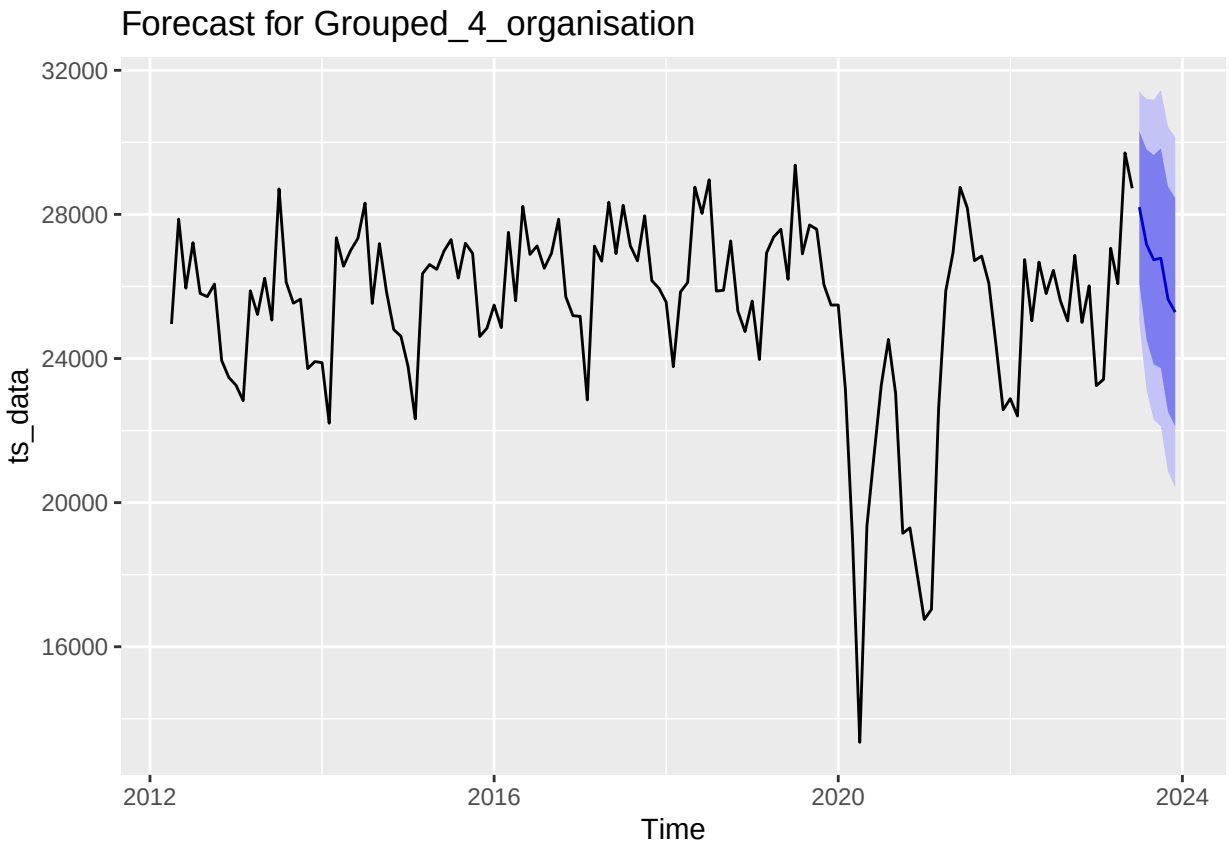
```
## Nov 2023       25645.29 22517.89 28772.70 20862.34 30428.25
## Dec 2023       25286.03 22113.33 28458.73 20433.81 30138.26
##
## $`Hywel Dda`
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Jul 2023       15984.72 14923.747 17045.70 14362.099 17607.35
## Aug 2023       16088.04 14582.199 17593.88 13785.055 18391.02
## Sep 2023       14280.20 12518.452 16041.94 11585.841 16974.55
## Oct 2023       13551.66 11633.763 15469.55 10618.491 16484.82
## Nov 2023       12548.97 10532.122 14565.81  9464.468 15633.47
## Dec 2023       11964.73  9883.705 14045.76  8782.076 15147.39
##
## $`Powys Teaching`
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Jul 2023       1593.479 1421.3226 1765.635 1330.1887 1856.768
## Aug 2023       1476.309 1242.9241 1709.693 1119.3776 1833.240
## Sep 2023       1391.293 1116.9363 1665.651  971.7004 1810.887
## Oct 2023       1260.690  956.2214 1565.159  795.0455 1726.335
## Nov 2023       1188.828  861.2557 1516.400  687.8494 1689.806
## Dec 2023       1020.005  674.2630 1365.747  491.2381 1548.772
##
## $Total
##          Point Forecast    Lo 80     Hi 80    Lo 95    Hi 95
## Jul 2023       96354.13 89558.55 103149.71 85961.19 106747.1
## Aug 2023       89505.44 80770.23  98240.66 76146.08 102864.8
## Sep 2023       90104.99 80309.35  99900.63 75123.86 105086.1
## Oct 2023       89432.47 79003.05  99861.90 73482.04 105382.9
## Nov 2023       85475.78 74652.91  96298.65 68923.62 102027.9
## Dec 2023       83342.11 72270.11  94414.11 66408.95 100275.3
```

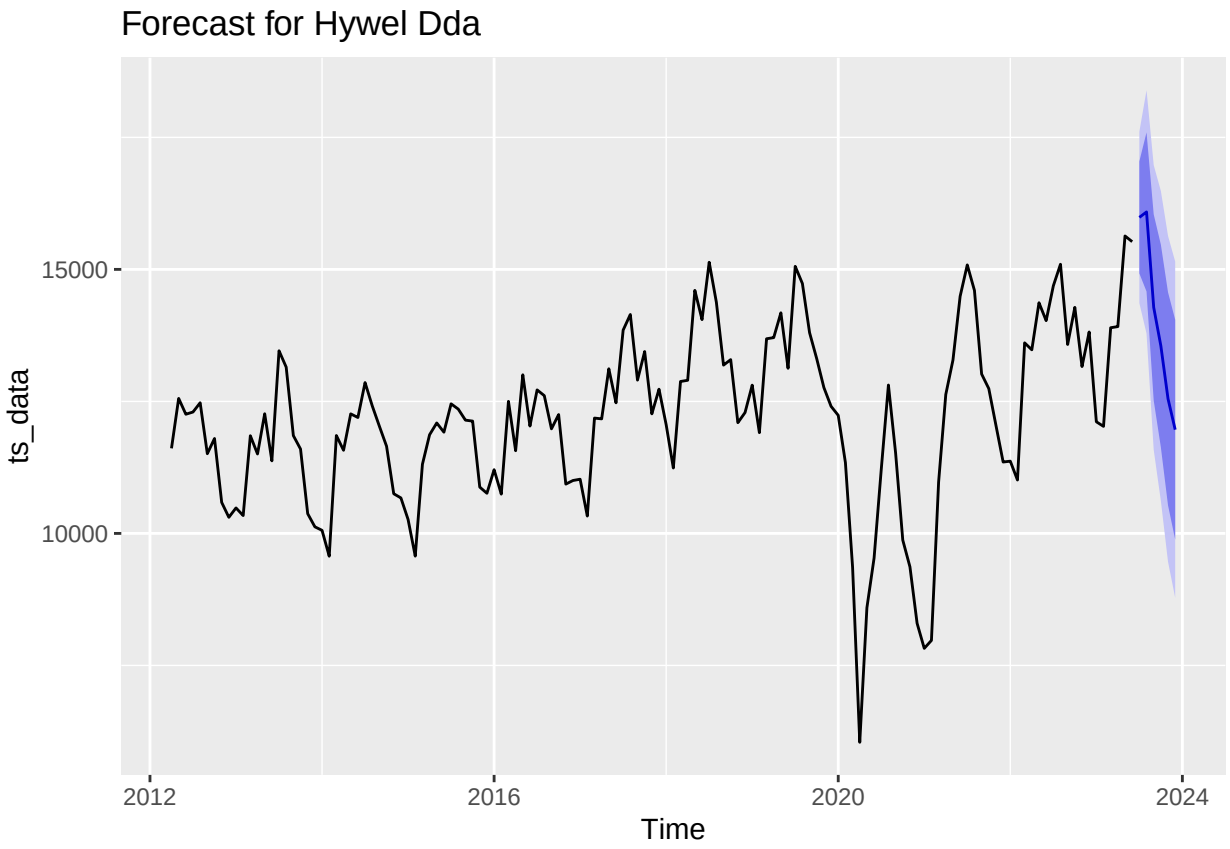
```
for (col in cols_to_forecast) {
  print(autoplot(forecast_list[[col]]) + ggtitle(paste("Forecast for", col)))
}
```

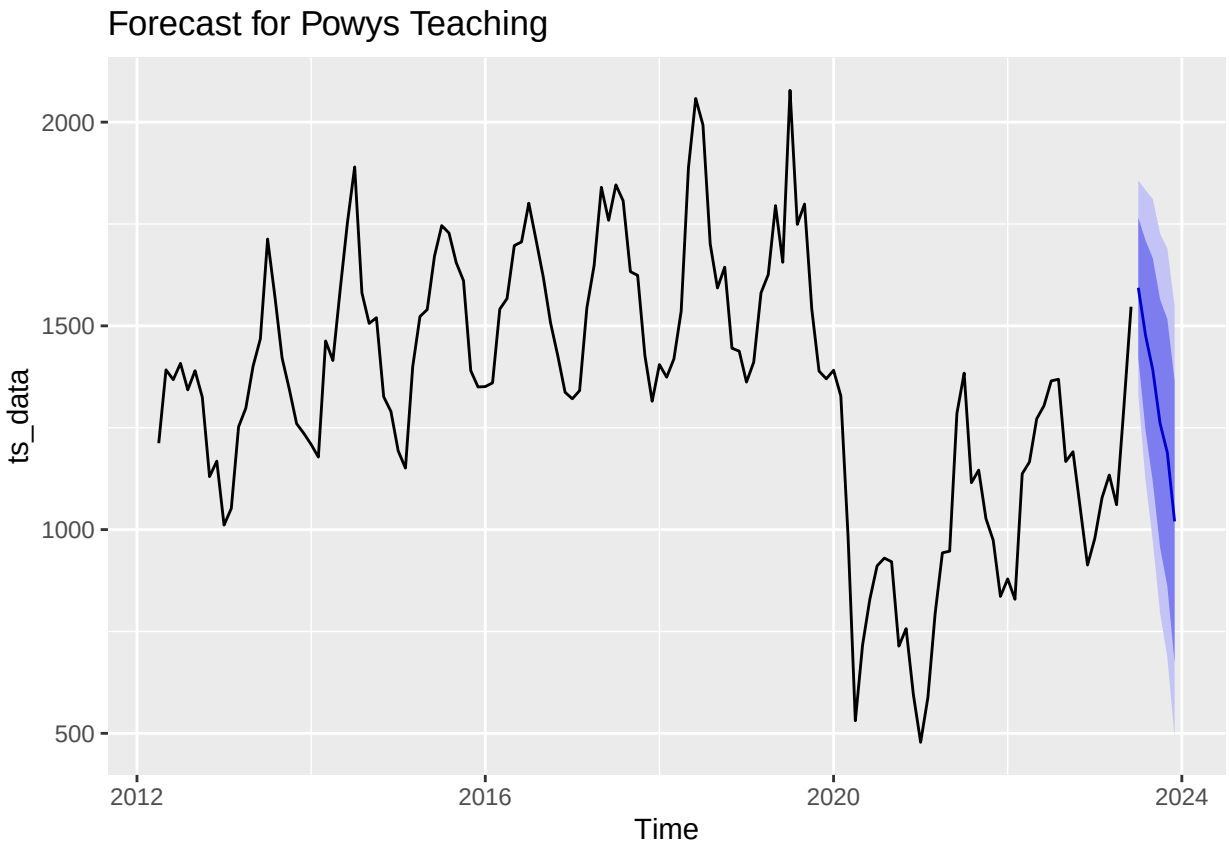

Forecast for Aneurin Bevan
ts_data
16000
12000
8000
2012
2016
2020
2024
Time

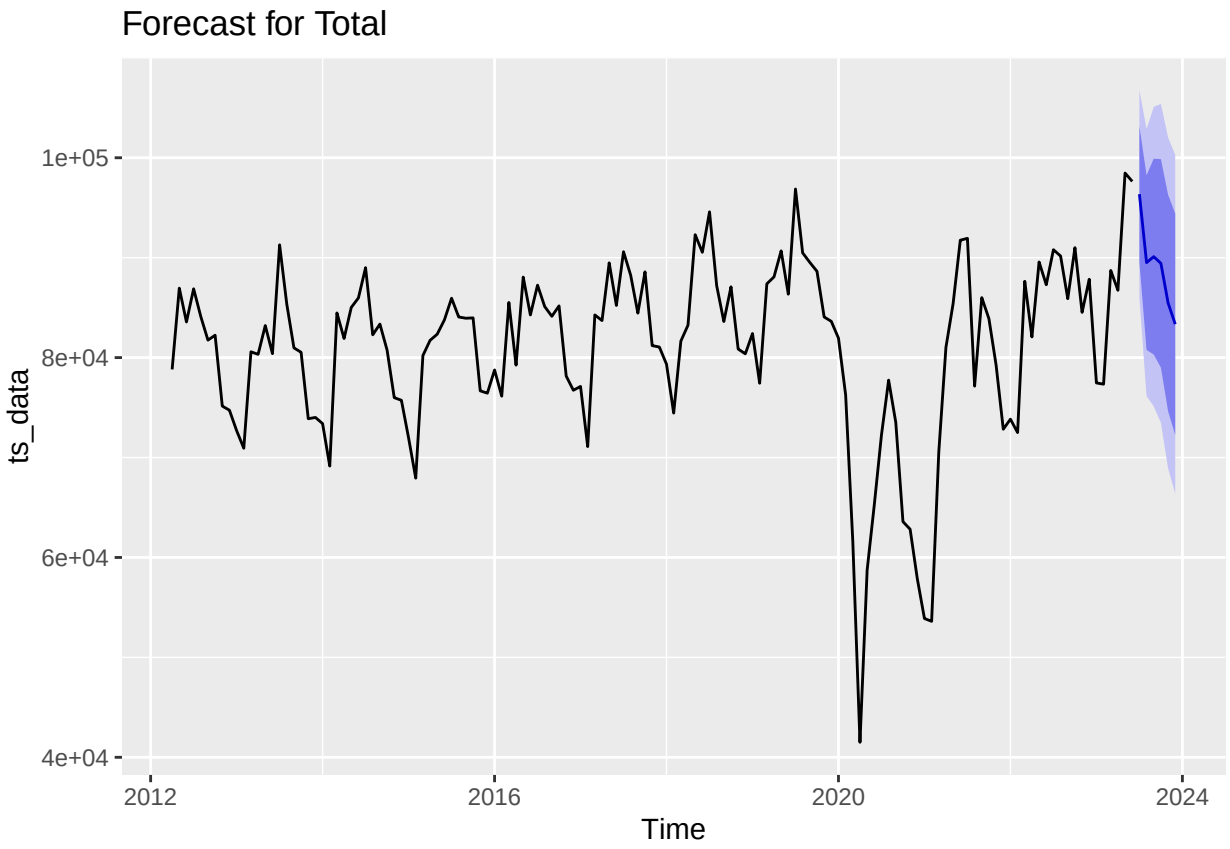

Forecast for Betsi Cadwaladr
25000
20000
15000
10000
ts_data
2012
2016
2020
2024
Time


Forecast for Cardiff & Vale
15000
12500
10000
7500
ts_data
2012
2016
2020
2024
Time


Forecast for Grouped_4_organisation
32000
28000
24000
20000
16000
ts_data
2012
2016
2020
2024
Time


Forecast for Hywel Dda
15000
10000
ts_data
2012
2016
2020
2024
Time


Forecast for Powys Teaching
2000
1500
1000
500
ts_data
2012
2016
2020
2024
Time


Forecast for Total
1e+05
8e+04
6e+04
4e+04
ts_data
2012
2016
2020
2024
Time