Forecasting Demo

Josh Tyler

Introduction

This is an example analysis report for the mpox forecasting competition. Here we outline an analysis workflow that starts at loading in the data and ends with the creation of the Estimates in the correct submission format.

Organising The Data

In order to create a forecast, we must first load and clean the raw Incidence data. The historic data was filtered to include only rows where there is no missing data and then joined to a table providing England regions. We have left the incidence rates as per 100,000 population.

```
library(tidyverse)
library(sf)
library(geodata)

dat <- read.csv("data/FULL-UKHSA-2017-2022-Lyme-Disease.csv") |>
  filter(!is.na(Value))

dat <- dat |>
  filter(Area.Type %in% c("UA","District")) |>
  select(Area.Code,Area.Name,Area.Type,
         Time.period,Value,Denominator,
         Lower.CI.95.0.limit,Upper.CI.95.0.limit)

auth_map<-st_read("data/Local_Authority_Districts_December_2024_Maps/LAD_DEC_24_UK_BFC.shp")
auth_extent<-ext(auth_map)


region_lookup<-read.csv("data/Region_Lookup.csv") |>
```

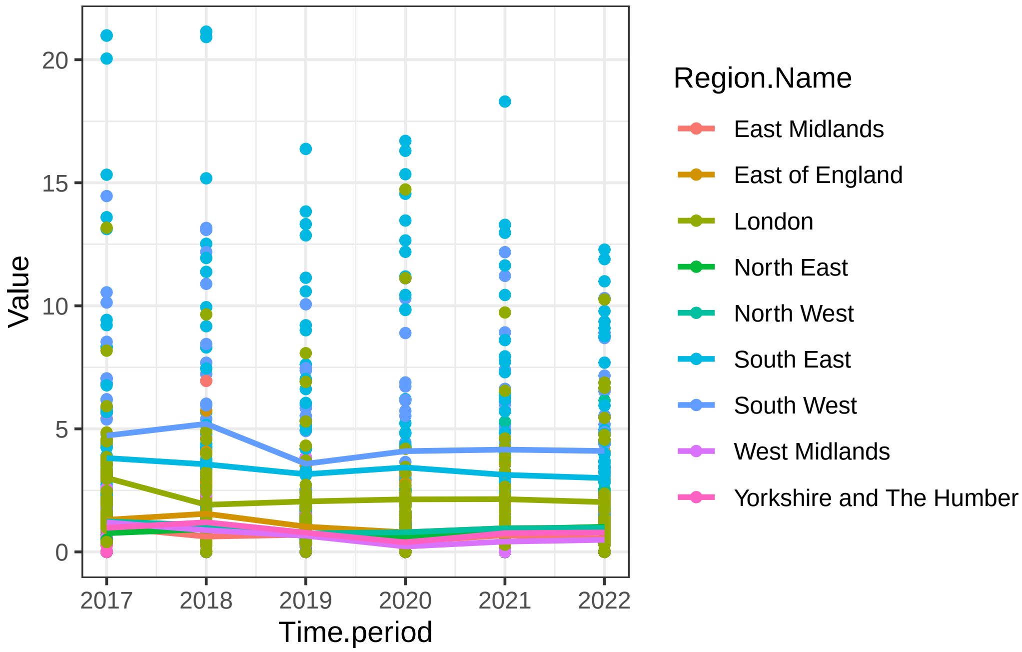
```
  select(LTLA22CD,LTLA22NM,UTLA22CD,UTLA22NM,RGN22CD,RGN22NM) |>
  distinct()

df<- dat |>
  left_join(region_lookup |>
              select("LTLA22CD","RGN22CD","RGN22NM") |>
              rename("Area.Code" = "LTLA22CD",
                     "Region.Code"= "RGN22CD",
                     "Region.Name" = "RGN22NM"),
            by = "Area.Code"
            )
```

Plotting The Data

In order to assess whether there is a significant difference between English Regions, we have plotted Incidence against time, with a mean value for each region as a line plot.

```
# Calculate average Value per Region.Name and Time.period
average_df <- df %>%
  group_by(Region.Name, Time.period) %>%
  summarise(Average_Value = mean(Value, na.rm = TRUE))
```

```
# Create the ggplot with points and lines
ggplot(data = df) +
  geom_point(aes(x = Time.period, y = Value, colour = Region.Name)) +
  geom_line(data = average_df,
            aes(x = Time.period, y = Average_Value,
                colour = Region.Name), size = 1) +
  theme_bw()
```



Creating a Forecast

This example uses a simple regression model whereby each area has its own coefficients.

```
results<-expand_grid(Area.Name=unique(dat$Area.Name),Time.period=c(2023,2024)) |>
  as.data.frame()

model<-lm(formula = Value ~ Time.period + Area.Name,data = df)

prediction<-predict(object = model,newdata = results)

prediction[which(prediction<0)]<-0

results<-cbind(results,prediction)
```

Creating the results table

We have constructed a data.frame to match the expected output columns, as given in the output template.

```r
library(gt)

forecast<-results |>
  select(Time.period,Area.Name,prediction) |>
  rename("Year"=Time.period,"Council"="Area.Name","Incidence"=prediction) |>
  mutate(Lower_95CI=Incidence) |>
  mutate(Upper_95_CI=Incidence) |>
  arrange(Year,Council)

write.csv(x = forecast,file = "example_forecast.csv")

gt(forecast)
```

Year	Council	Incidence	Lower_95CI	Upper_95_CI
2023	Adur	0.702155590	0.702155590	0.702155590
2023	Allerdale	2.558111403	2.558111403	2.558111403
2023	Amber Valley	0.329252256	0.329252256	0.329252256
2023	Arun	3.065688923	3.065688923	3.065688923
2023	Ashfield	0.000000000	0.000000000	0.000000000
2023	Ashford	0.567047256	0.567047256	0.567047256
2023	Babergh	0.023755590	0.023755590	0.023755590
2023	Barking and Dagenham	0.594777256	0.594777256	0.594777256
2023	Barnet	0.906413923	0.906413923	0.906413923
2023	Barnsley	0.622365590	0.622365590	0.622365590
2023	Barrow-in-Furness	0.222241403	0.222241403	0.222241403
2023	Basildon	0.470088923	0.470088923	0.470088923
2023	Basingstoke and Deane	8.556405590	8.556405590	8.556405590
2023	Bassetlaw	0.943608923	0.943608923	0.943608923
2023	Bath and North East Somerset	5.479268923	5.479268923	5.479268923
2023	Bedford	0.977353923	0.977353923	0.977353923
2023	Bexley	0.140745590	0.140745590	0.140745590
2023	Birmingham	0.263868923	0.263868923	0.263868923
2023	Blaby	0.000000000	0.000000000	0.000000000
2023	Blackburn with Darwen	0.214595590	0.214595590	0.214595590
2023	Blackpool	0.612205590	0.612205590	0.612205590
2023	Bolsover	0.923855590	0.923855590	0.923855590
2023	Bolton	0.300698923	0.300698923	0.300698923
2023	Boston	0.610803923	0.610803923	0.610803923
2023	Bournemouth, Christchurch and Poole	3.055533923	3.055533923	3.055533923
2023	Bracknell Forest	1.839478923	1.839478923	1.839478923
2023	Bradford	0.311613923	0.311613923	0.311613923
2023	Braintree	1.084927256	1.084927256	1.084927256
2023	Breckland	1.931475590	1.931475590	1.931475590
2023	Brent	0.362327256	0.362327256	0.362327256
2023	Brentwood	0.749478923	0.749478923	0.749478923
2023	Brighton and Hove	2.197852256	2.197852256	2.197852256
2023	Bristol	4.172058923	4.172058923	4.172058923
2023	Broadland	0.565078923	0.565078923	0.565078923
2023	Bromley	1.381728923	1.381728923	1.381728923
2023	Bromsgrove	0.679962256	0.679962256	0.679962256
2023	Broxbourne	0.688262256	0.688262256	0.688262256
2023	Broxtowe	0.000000000	0.000000000	0.000000000
2023	Buckinghamshire UA	1.350205590	1.350205590	1.350205590
2023	Burnley	0.423822256	0.423822256	0.423822256
2023	Bury	0.276375590	0.276375590	0.276375590
2023	Calderdale	0.789370590	0.789370590	0.789370590
2023	Cambridge	2.339382256	2.339382256	2.339382256
2023	Camden	5.123983923	5.123983923	5.123983923
2023	Cannock Chase	0.000000000	0.000000000	0.000000000
2023	Canterbury	0.082567256	0.082567256	0.082567256
2023	Carlisle	0.399691403	0.399691403	0.399691403
2023	Castle Point	0.038787256	0.038787256	0.038787256
2023	Central Bedfordshire	0.999818923	0.999818923	0.999818923