Health Data Science Project Report Workflow

2022-11-25

Load packages

```
# For data handling
library(tidyverse)
library(glue)

# For plotting
library(treemapify)
library(gganimate)
library(patchwork)
library(ggridges)
library(hrbrthemes)
library(scales)
```

Read in and view data

```
# Read in domestic abuse data from ScotPHO
data <- read_csv("rank_data.csv", show_col_types = FALSE)

# View and summarise data
head(data)
```

```
## # A tibble: 6 x 14
##   indicator area_~1 area_~2 area_~3  year period numer~4 measure lower~5 upper~6
##   <chr>     <chr>   <chr>   <chr>   <dbl> <chr>    <dbl>   <dbl>   <dbl>   <dbl>
## 1 Domestic~ Dundee~ S12000~ Counci~  2019 2019/~    2480    166.    160.    173.
## 2 Domestic~ Clackm~ S12000~ Counci~  2019 2019/~     781    152.    141.    162.
## 3 Domestic~ Glasgo~ S12000~ Counci~  2019 2019/~    9539    151.    148.    154.
## 4 Domestic~ West D~ S12000~ Counci~  2019 2019/~    1338    150.    142.    159.
## 5 Domestic~ North ~ S12000~ Counci~  2019 2019/~    4801    141.    137.    145.
## 6 Domestic~ Falkirk S12000~ Counci~  2019 2019/~    2123    132     126.    138.
## # ... with 4 more variables: comparator_value <dbl>, comparator_name <chr>,
## #   definition <chr>, data_source <chr>, and abbreviated variable names
## #   1: area_name, 2: area_code, 3: area_type, 4: numerator,
## #   5: lower_confidence_interval, 6: upper_confidence_interval
```

```
glimpse(data)
```

```
## Rows: 32
## Columns: 14
## $ indicator                <chr> "Domestic abuse", "Domestic abuse", "Domesti~
## $ area_name                <chr> "Dundee City", "Clackmannanshire", "Glasgow ~
## $ area_code                <chr> "S12000042", "S12000005", "S12000049", "S120~
## $ area_type                <chr> "Council area", "Council area", "Council are~
```

```
## $ year                        <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 20~
## $ period                      <chr> "2019/20 financial year", "2019/20 financial~
## $ numerator                   <dbl> 2480, 781, 9539, 1338, 4801, 2123, 1206, 174~
## $ measure                     <dbl> 166.1, 151.5, 150.7, 150.5, 140.6, 132.0, 13~
## $ lower_confidence_interval <dbl> 159.6, 141.1, 147.7, 142.5, 136.7, 126.4, 12~
## $ upper_confidence_interval <dbl> 172.8, 162.5, 153.7, 158.7, 144.7, 137.7, 13~
## $ comparator_value            <dbl> 124.39, 83.42, 133.49, 138.20, 88.81, 77.82,~
## $ comparator_name             <chr> "Dundee City", "Clackmannanshire", "Glasgow ~
## $ definition                  <chr> "Crude rate per 10,000 population", "Crude r~
## $ data_source                 <chr> "Scottish Government (Scottish Crime Statist~
```

The summarised views of the data suggest that there are a) columns with static information (across all rows) that might be relevant for display later on b) information necessary to perform analyses c) redundant information.

Prepare and clean data

```
# a) Save relevant static information
period <- data$period
indicator <- data$indicator
source <- data$data_source

# b) Select relevant data
data_clean <-
  data %>% select(
    area_name,
    area_code,
    abs_val = numerator,
    rel_2019 = measure,
    rel_2004 = comparator_value
  )

# Recover area population size from given values
# r = n/N * 10000 <=> N = 10000/r * n
# N: area population size
# r: incident rate per 10,000
# n: absolute number of incidents
data_clean <-
  data_clean %>% mutate(pop = round((10000 / rel_2019) * abs_val))
```

For the second approach, we need to add information on all remaining years in between 2004 and 2019.

```
# Add data from all other years
for (i in 2005:2018) {
  measure <- as.name("measure")
  data_clean <-
    read_csv(glue("rank_data_{i}.csv"), show_col_types = FALSE) %>%
    select(area_code, "rel_{{i}}" := {{measure}}) %>%
    right_join(data_clean, by = c("area_code" = "area_code"))
}
```

Next, we calculate the change in domestic abuse rates. For one council area and year, we define change as the domestic abuse rate of the council area during that year, divided by the rate for that council area referring to the baseline year (2004)).

```r
# Calculate "change" for each year
data_clean <- data_clean %>%
  rename_with(.cols = starts_with("rel"), ~ gsub("L", "", .x)) %>%
  mutate(across(
    starts_with("rel"),
    .names = "change_{col}",
    .fns = ~ . / rel_2004
  ))
```

Now, we can clean up and tidy our data to make plotting easier.

```r
# To reduce redundancy, only keep 2019 raw rate value data
data_clean <-
  data_clean %>% rename (value = "rel_2019") %>% select(-starts_with("rel"))

# Make data tidy (long)
data_clean <-
  data_clean %>% pivot_longer(
    cols = starts_with("change"),
    names_to = "year",
    values_to = "change",
    names_prefix = "change_rel_"
  ) %>% mutate_at(c("year"), as.integer)
```

We log-transform the change in rates to mitigate the skewness of the distribution due to outliers of very small population size.

```r
# Perform log-transformation
data_clean <- data_clean %>% mutate(change = log10(change))

head(data_clean)
```

```
## # A tibble: 6 x 7
##   area_code area_name   abs_val value    pop  year change
##   <chr>     <chr>         <dbl> <dbl>  <dbl> <int>  <dbl>
## 1 S12000042 Dundee City    2480  166. 149308  2018 0.0997
## 2 S12000042 Dundee City    2480  166. 149308  2017 0.0557
## 3 S12000042 Dundee City    2480  166. 149308  2016 0.0894
## 4 S12000042 Dundee City    2480  166. 149308  2015 0.108
## 5 S12000042 Dundee City    2480  166. 149308  2014 0.137
## 6 S12000042 Dundee City    2480  166. 149308  2013 0.122
```

In order to provide a form of grouping, we classify council areas as predominantly urban or rural. Using the Scottish Government Urban Rural Classification 2020, we categorized 4 council areas (whose vast majority of the population (>= 95%) lived in large urban areas) as urban.

```r
# Assign classification to urban areas
regions <- tribble(
  ~ area_name,
  ~ region,
  "Aberdeen City",
  "urban",
  "City of Edinburgh",
  "urban",
  "Dundee City",
  "urban",
```

```
  "Glasgow City",
  "urban",
)

# Add urban region info to main tibble
data_clean <- left_join(data_clean, regions, by = c("area_name"))

# Add rural region info to unassigned fields
data_clean <-
  data_clean %>% mutate_at(c("region"), ~ replace(., is.na(.), "rural"))
```

Plot data

Approach 1

For the first visualisation, we have to generate average data for each region group.

```
# Remove rows from redundant years
data_border_years <- data_clean %>% filter(year == 2019)

# Create average data points per region group (urban/rural)
data_avg <- data_border_years %>% group_by(region) %>%
  summarise(
    sum_pop = sum(pop),
    mean_value = mean(value),
    mean_change = mean(change)
  )
```

Then we can proceed to plot the data. First, we plot the abuse rate in the financial year 2019/2020 against the change in rates from 2004/2005 to 2019/2020 for each council area.

```
# Set limits for formatting
limits_pop <- c(0, 4000000)
limits_x <- c(-0.1, 0.6)
limits_y <- c(0, 175)

p1 <-
  ggplot(data_border_years,
         aes(
           x = change,
           y = value,
           size = pop,
           color = region
         )) +
  geom_point() +
  theme_bw() +
  ggtitle("Domestic Abuse over 15 Years in Scottish Council Areas") +
  labs(subtitle="Domestic Abuse Rate (DAR) = incidents per 10,000 population") +
  scale_size(limits = limits_pop,
             range = c(2, 12)) +
  scale_color_manual(values = c("#3199eb", "#fdcb47")) +
  xlim(limits_x) +
  xlab("Log Change in DAR from 2004 to 2019") +
  ylim(limits_y) +
```

```
  ylab("DAR per Council Area in 2019") +
  theme(
    aspect.ratio = 9 / 16,
    plot.title = element_text(size = 12, hjust = 0.5),
    axis.title.x = element_text(size = rel(0.7)),
    axis.title.y = element_text(size = rel(0.7)),
    legend.position = "none",
    plot.subtitle=element_text(size = 8, hjust = 0.5)
  )
```

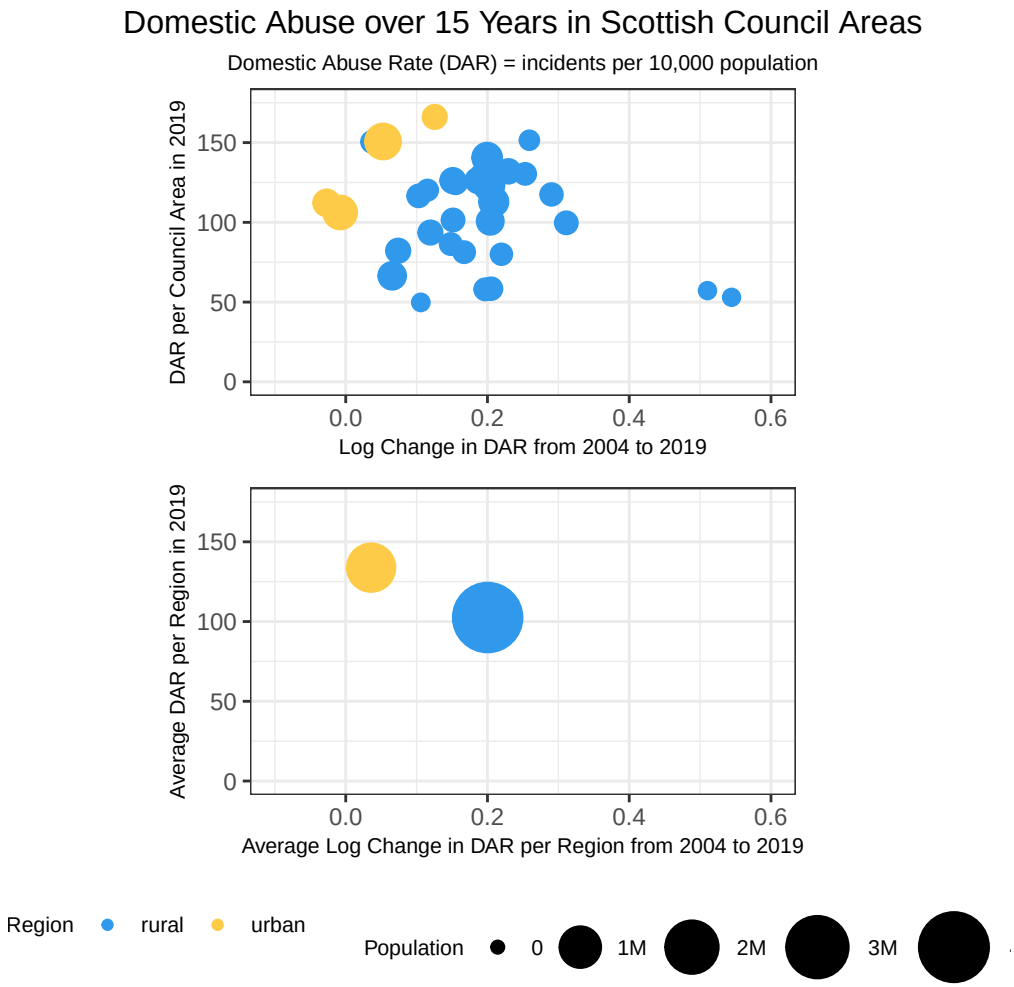
The calculation is straightforward, apart from the call of `scale_size()`. Here, we make sure to indicate the rough limits (`limits`) the population values take in this data set and how big we want the points to get (`range`). Additionally, we specify the notation (`labels`) of the big numbers the population value can take for the legend (generated in the chunk below).

Below, we plot the average values for each region group (summarised populations) and assemble the plot.

```
# Rename columns for display
data_avg <-
  data_avg %>% rename(Region = "region", Population = "sum_pop")

p2 <-
  ggplot(data_avg,
         aes(
           x = mean_change,
           y = mean_value,
           size = Population,
           color = Region
         )) +
  geom_point() +
  theme_bw() +
  scale_size_continuous(
    limits = limits_pop,
    range = c(2, 12),
    labels = label_number(scale_cut = cut_short_scale())
  ) +
  scale_color_manual(values = c("#3199eb", "#fdcb47")) +
  xlim(limits_x) +
  xlab("Average Log Change in DAR per Region from 2004 to 2019") +
  ylim(limits_y) +
  ylab("Average DAR per Region in 2019") +
  theme(
    aspect.ratio = 9 / 16,
    plot.title = element_text(size = 12, hjust = 0.5),
    axis.title.x = element_text(size = rel(0.7)),
    axis.title.y = element_text(size = rel(0.7)),
    legend.position = "bottom",
    legend.text = element_text(size = 8),
    legend.title = element_text(size = 8)
)

# Assemble whole plot
p1 / p2 + plot_layout()
```



Approach 2

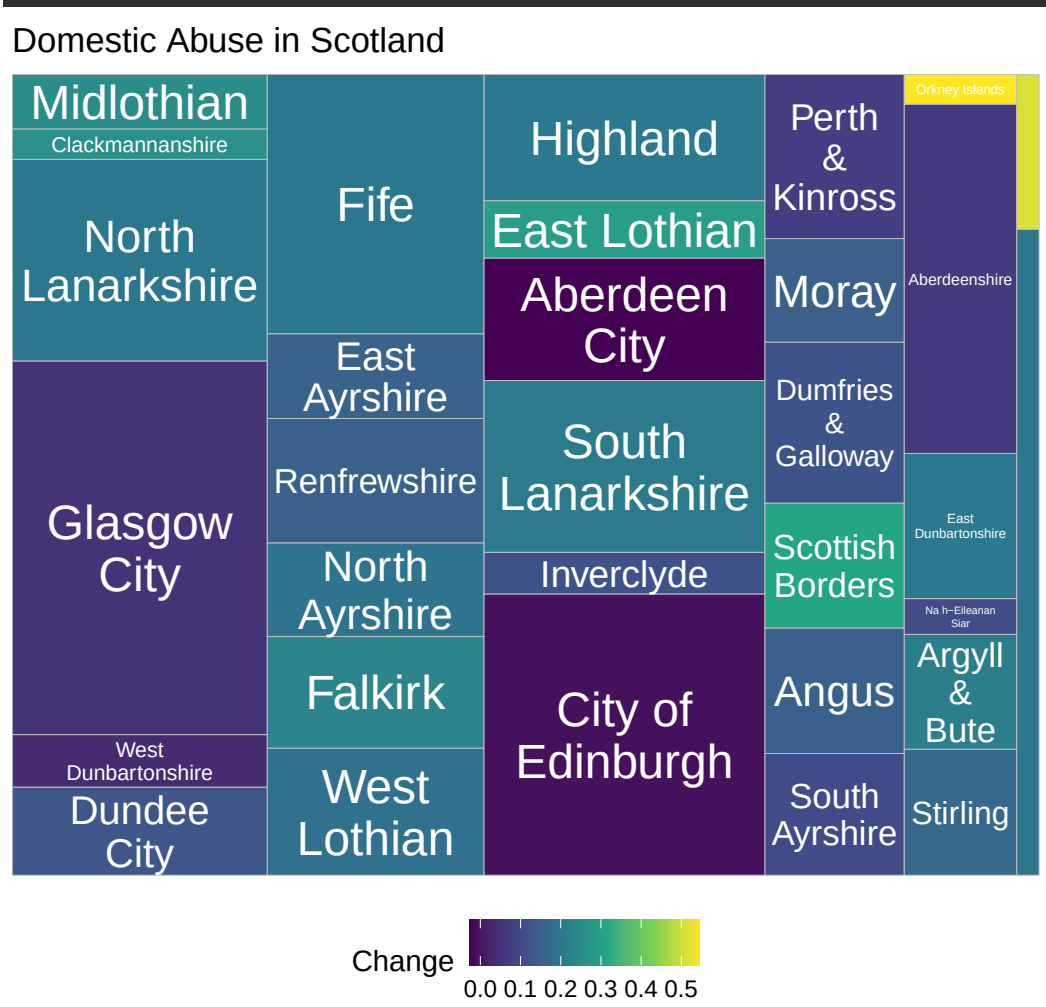
Next, we plot an animated treemap of the change in abuse rates from the baseline year (2004/2005) to each year through 2019/2020 for each council area. For this document, we plot a static version displaying only the frame for the 2019/2020 year.

```
# Rename columns for display
data_clean <- data_clean %>% rename(Change = "change")

p3 <- data_clean %>% filter(year == 2019) %>%
  ggplot(aes(
    area = pop,
    fill = Change,
    label = area_name,
  )) +
  geom_treemap(layout = "fixed") +
  geom_treemap_text(
    layout = "fixed",
    colour = "white",
    place = "centre",
    reflow = TRUE
  ) +
  scale_fill_viridis_c() +
```

```
  labs(title = "Domestic Abuse in Scotland",
       caption = paste("The change in domestic abuse rates from 2004 to",
                       " 2019, for each council area", sep="")) +
  theme(legend.position = "bottom")

p3
```



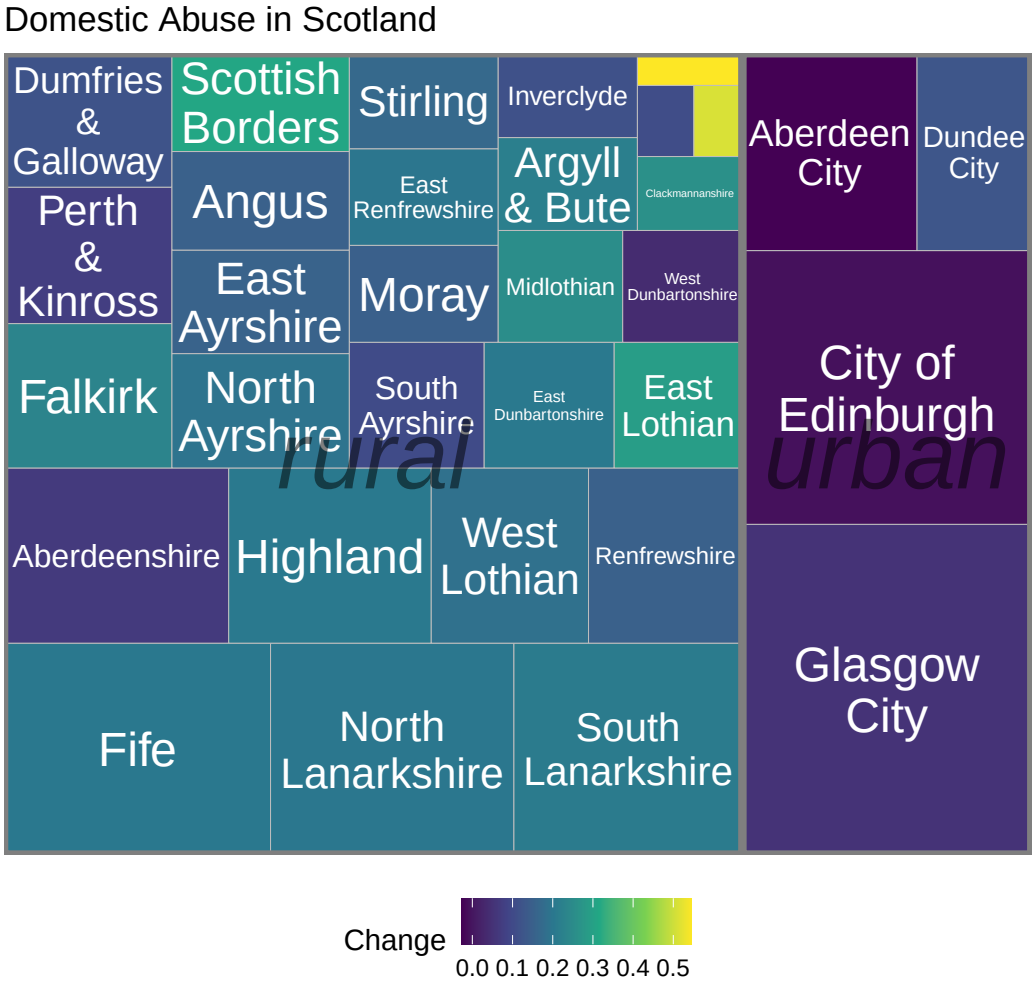
The change in domestic abuse rates from 2004 to 2019, for each council area

Again, we can make use of the region grouping, if we were to use the visualisation to support the scatterplots created above. The main message of Approach 2 ("Domestic abuse in Scotland on the rise") however does not require this information. This is what the plot would look like if we wanted to illustrate the key message of Approach 1:

```
p4 <- data_clean %>% filter(year == 2019) %>%
  ggplot(aes(
    area = pop,
    fill = Change,
    label = area_name,
    subgroup = region,
  )) +
  geom_treemap() +
  geom_treemap_text(colour = "white",
```

```
                    place = "centre",
                    reflow = TRUE) +
  geom_treemap_subgroup_border() +
  geom_treemap_subgroup_text(
    place = "centre",
    alpha = 0.5,
    colour =
      "black",
    fontface = "italic"
  ) +
  scale_fill_viridis_c() +
  labs(title = "Domestic Abuse in Scotland",
       caption = paste("The change in domestic abuse rates from 2004 to",
                       " 2019, for each council area", sep=""))  +
  theme(legend.position = "bottom")

p4
```



The change in domestic abuse rates from 2004 to 2019, for each council area

If we want to plot an animated version to further illustrate the key message of Approach 2, and safe it as a .gif file, we use the following code:

```r
p5 <-
  ggplot(data_clean, aes(area = pop, fill = Change, label = area_name)) +
  geom_treemap(layout = "fixed") +
  ggtitle("Domestic Abuse in the Past 15 Years in Scotland") +
  geom_treemap_text(layout = "fixed",
                    colour = "white",
                    place = "centre") +
  scale_fill_viridis_c() +
  transition_time(year) +
  ease_aes("linear") +
  labs(title = "Domestic Abuse in Scotland",
       caption = "The change in domestic abuse rates from 2004 to
                  {frame_time} f, for each council area") +
  theme(legend.position = "bottom")

anim_save("domestic-abuse_treemap.gif", p5)
```