Data report project

Data report on Population Prescribed Drug

This document was written by Irina Laidvee for data project (PU5058)

The data for the project was used form "Scottish Public Health Observatory" website via the ScotPHO Online Profiles tool access. The Rank tab was used to compare geographical variation for an indicator. First "Population prescribed drug for anxiety/depression/psychosis" was chosen for an indicator, then all Health Boards within Scotland were selected for geography level and compared by time. Areas were compared to each other over time since 2010/11 as a baseline year to 2021/22.

First, we load all packages and load the raw data.

```
library(here)
```

```
## here() starts at C:/Users/irina/Desktop/current work search/UoA Data Science 2023/GitHub/Population_
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
```

```
## -- 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(viridisLite)
library(latexpdf)

#read the data

rank_data <- read_csv(here("Inputs/rank_data.csv"))
```

```
## Rows: 14 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): indicator, area_name, area_code, area_type, period, comparator_name...
```

```
## dbl (6): year, numerator, measure, lower_confidence_interval, upper_confiden...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

This shows that we have 14 rows for each Health Board and 14 columns with variables and observations. A tidy dataset has to contain only 1 observation per row.

Cleaning the dataset.

1. *Remove unnecessary columns.* We can remove indicator and data_source, as we already mentioned this at the beginning of the file. We remove area_type column as it is the same for all rows. We will not be using lower and upper confidence_intervals in our data visualization. We remove the period column as it indicates the same information as year column. The measure column indicates the value for the 2021 year, therefore it can be omitted. Comparator_name repeats the area_name and the value definition can be omitted as well, as it is the same across all boards. In the code we will specify the index position of each column that needs to be removed.

```
rank_data_NEW = rank_data[, -c(1, 4, 5, 6, 9, 10, 12, 13, 14)]
glimpse(rank_data_NEW)
```

```
## Rows: 14
## Columns: 5
## $ area_name        <chr> "NHS Ayrshire & Arran", "NHS Lanarkshire", "NHS Fife"~
## $ area_code        <chr> "S08000015", "S08000032", "S08000029", "S08000031", "~
## $ numerator        <dbl> 83388, 147046, 79882, 251524, 31409, 86314, 23891, 61~
## $ measure          <dbl> 22.62, 22.14, 21.32, 21.22, 21.11, 20.67, 20.59, 20.2~
## $ comparator_value <dbl> 16.31, 16.07, 15.26, 16.61, 14.75, 15.21, 15.15, 14.8~
```

2. *Change columns' names.* Here we will change the names of some columns. measure -> finish_2021, comparator_value -> start_2010. Also we will omit the NHS from area_name column.

```
colnames(rank_data_NEW)[colnames(rank_data_NEW) %in% c("measure",
          "comparator_value")] <- c("finish_2021", "start_2010")
presc_data <- rank_data_NEW %>%
  mutate(area_name = gsub ("NHS", "", area_name)) %>%
  rename(health_board = area_name)
```

3. *Separate data sets.* We have area_name and area_code as an identification for each row. In a clean data set this should be separate. Therefore, we have created a separate csv file containing geographical identification. We remove area_code column from the table and save new table.

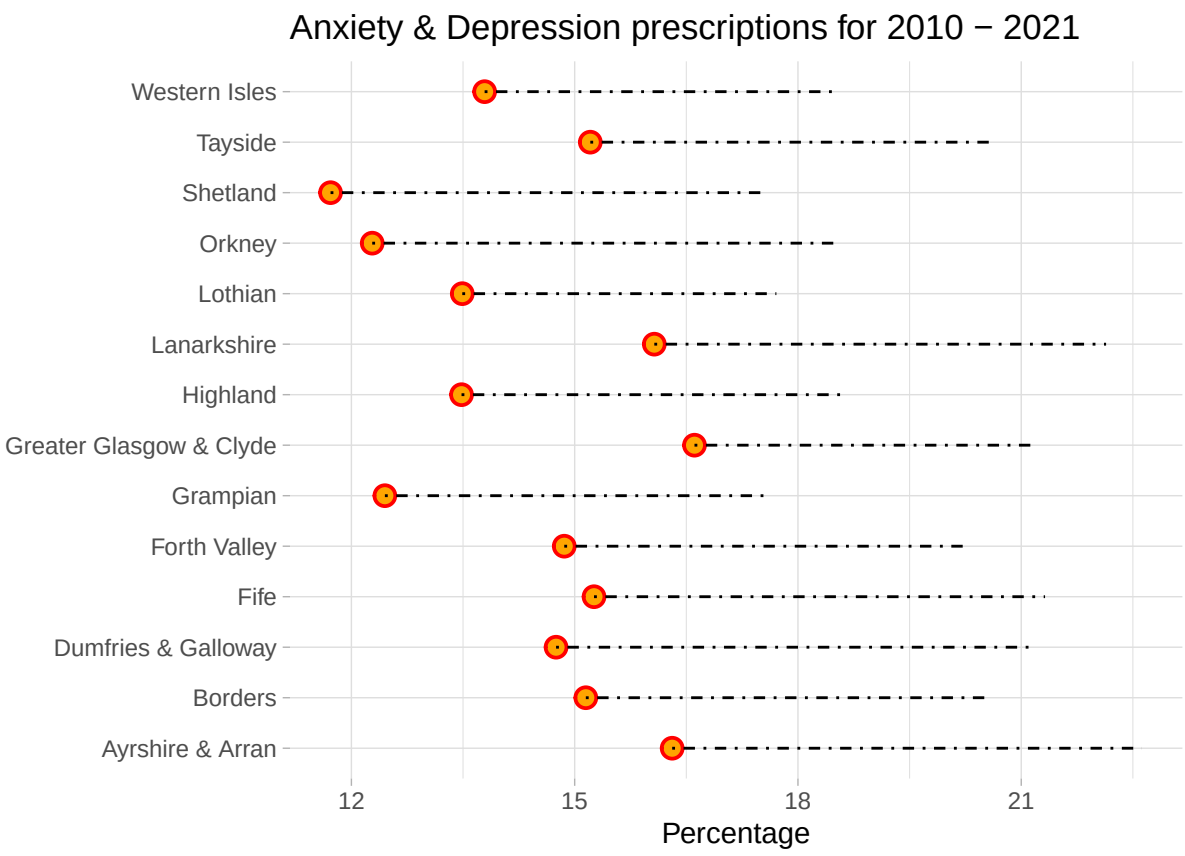
```
ID_rank_data <- presc_data[, c("health_board", "area_code")]
write_csv(ID_rank_data, "ID_rank_data.csv")

#remove area_code column
plot_data <- presc_data [, -c(2)]
write_csv(plot_data, "plot_data.csv")
```

Creating a data visualization

1. *Lollipop graph* We will be using library "forcats" and "ggplot2" from "tidyverse" and "viridisLite". We would like to create a lollipop plot, which is very close from both scatterplots and barplots.

```
ggplot(plot_data, aes(x=start_2010, y= health_board)) +
  #include point appearance
  geom_point(size = 3, color = "red", shape = 21, fill = "orange", stroke = 1) +
  #include segment appearance
  geom_segment(aes(x=start_2010, xend=finish_2021, y=health_board, yend=health_board) ,
               size = 0.5, color = "black", linetype = "dotdash") +
  #include plot theme
  theme_light() +
  theme(panel.border = element_blank() , panel.grid.minor.x = element_line() ) +
  #include title and labels
          ggtitle("Anxiety & Depression prescriptions for 2010 - 2021") +
xlab("Percentage") +
  ylab("")
```

```
## 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.
```

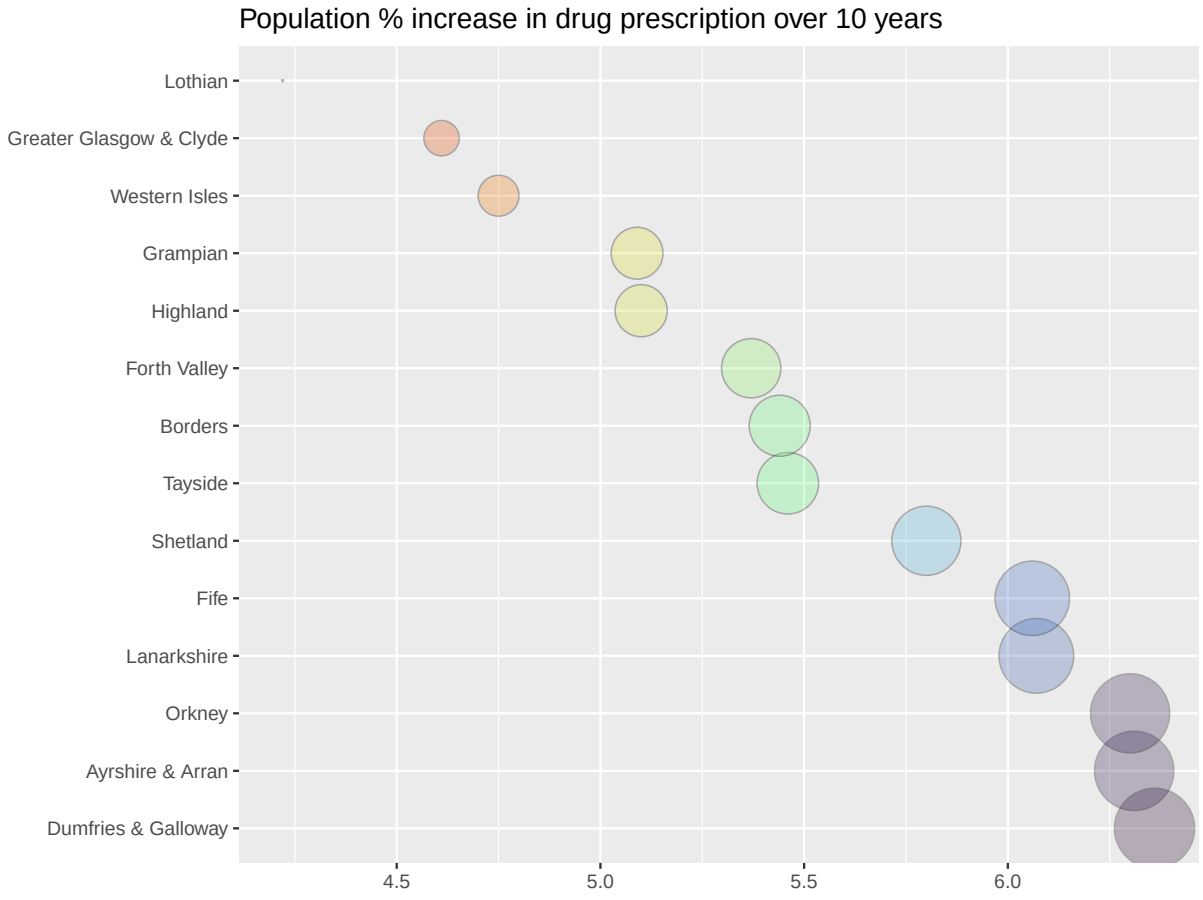


2. *Make some simple data analysis for further data visualization.* Let's add another column and calculate what is the total change in population % of prescribed medication during 2010 - 2021. We will organise the data in order, use bubble plot with colour changing scale in % difference.

```
plot_data2 <- plot_data  %>%
  mutate(total_change = finish_2021 - start_2010) %>%
  arrange(total_change)
write_csv(plot_data2, "plot_data2.csv")
```

Make a plot.

```
plot_data2 %>%
  arrange(desc(total_change)) %>%
  mutate(health_board = factor(health_board, health_board)) %>%
  ggplot(aes(x=total_change, y=health_board, size = total_change, fill= total_change)) +
  geom_point(alpha=0.3, color ="black", shape = 21) +
  scale_size(range = c(.1,17), name = "Percentage (%)") +
  scale_fill_viridis_c(alpha = 0.5, begin = 0, end = 1, direction = -1, option = "H",
                       values = NULL, space = "Lab", guide = "colourbar",
                       aesthetics = "fill") +
  xlab("") +
  ylab("") +
  theme(legend.position = "none") +
  ggtitle("Population % increase in drug prescription over 10 years")
```



3. *Some statistical analysis and data visualization.* Run a basic analysis with summary function.

```
summary(plot_data2)
```

```
##  health_board         numerator       finish_2021      start_2010
##  Length:14          Min.   :  4018   Min.   :17.52   Min.   :11.72
##  Class :character   1st Qu.: 25771   1st Qu.:18.55   1st Qu.:13.48
##  Mode  :character   Median : 70871   Median :20.41   Median :14.80
##                     Mean   : 78848   Mean   :19.88   Mean   :14.39
##                     3rd Qu.: 98743   3rd Qu.:21.19   3rd Qu.:15.25
##                     Max.   :251524   Max.   :22.62   Max.   :16.61
##   total_change
##  Min.   :4.220
##  1st Qu.:5.093
##  Median :5.450
##  Mean   :5.496
##  3rd Qu.:6.067
##  Max.   :6.360
```

Tidy the dataset for year comparison.

#column name change, pivot the table

```
colnames(plot_data)[colnames(plot_data) %in% c("finish_2021", "start_2010")] <- c("2021", "2010")
stat_test <- plot_data %>%
pivot_longer(3:4, names_to = "years", values_to = "per_cent") %>%
  select("years", "per_cent" )
```

#non-parametric test, shows there is a significant difference between the years

```
t.test(data = stat_test, per_cent ~ years, paired = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  per_cent by years
## t = -8.7795, df = 25.589, p-value = 3.395e-09
## alternative hypothesis: true difference in means between group 2010 and group 2021 is not equal to 0
## 95 percent confidence interval:
##  -6.783423 -4.208006
## sample estimates:
## mean in group 2010 mean in group 2021
##           14.38786           19.88357
```

```
#The negative t-value is due to the order of the groups.
#If the groups are switched, the answer will be the same, but with a positive sign.
```

#boxplot

```r
box_plot <- plot_data %>%
pivot_longer(3:4, names_to = "years", values_to = "per_cent")

ggplot(box_plot, aes(x = as.factor(years), y = per_cent, fill = as.factor(years))) +
  geom_boxplot(alpha=0.6, show.legend = FALSE, outlier.color = "red", outlier.shape = 19) +
  stat_boxplot(geom = "errorbar", width = 0.2) +
  geom_jitter(color = "black", size = 2, alpha = 0.8) +
 ylab("Percentage") +
  xlab("") +
  theme_bw() +
  theme(legend.position = "none", plot.title = element_text(size=18)) +
  ggtitle("Health Boards values overlay for 2010 and 2021")
```