Trend in the percentage of the Scottish population prescribed drugs for anxiety, depression or psychosis

Fionnuala Cousins 52319277

19 November, 2023, 21:48

```
# tidyverse includes dplyr and ggplot2 so I don't need to load them separately
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(here)
```

```
## here() starts at C:/Users/Fionnuala/OneDrive - University of Aberdeen/PU5063 Intro to HDS/Assessment
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

Step 0: Define the question

What are the regional trends for the number of people prescribed drugs for anxiety, depression and psychosis in Scotland over the last ten years? What might these mean for employers' allocation of support resources? The next sections follow the Health Data Science Workflow to address these questions.

Step 1: Data Acquisition

The data was downloaded from the Scottish Public Health Observatory https://scotland.shinyapps.io/ScotPHO_profiles_tool/ on 05/11/23 for the item "population prescribed drugs for anxiety/depression/psychosis" for all available years (2010-2021 inclusive) and all 14 health boards in Scotland, as well as an overall Scotland dataset. The downloaded file was called timetrend_data.csv

```r
#reading in the data and checking its columns:
adp_data <- read_csv(here("Inputs/timetrend_data.csv"))
```

```
## Rows: 180 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): indicator, area_name, area_code, area_type, period, definition, dat...
## dbl (5): 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.
```

```r
glimpse(adp_data)
```

```
## Rows: 180
## Columns: 12
## $ indicator                 <chr> "Population prescribed drugs for anxiety/dep~
## $ area_name                 <chr> "Scotland", "NHS Ayrshire & Arran", "NHS Bor~
## $ area_code                 <chr> "S00000001", "S08000015", "S08000016", "S080~
## $ area_type                 <chr> "Scotland", "Health board", "Health board", ~
## $ year                      <dbl> 2010, 2010, 2010, 2010, 2010, 2010, 2010, 20~
## $ period                    <chr> "2010/11 financial year", "2010/11 financial~
## $ numerator                 <dbl> 787040, 60822, 17226, 22280, 55334, 43976, 7~
## $ measure                   <dbl> 14.96, 16.31, 15.15, 14.75, 15.26, 14.86, 12~
## $ lower_confidence_interval <dbl> 14.93, 16.20, 14.94, 14.57, 15.14, 14.73, 12~
## $ upper_confidence_interval <dbl> 14.99, 16.43, 15.36, 14.92, 15.38, 14.98, 12~
## $ definition                <chr> "Percentage", "Percentage", "Percentage", "P~
## $ data_source               <chr> "Public Health Scotland (Prescribing Informa~
```

Step 2: Prepare/ clean data

Is it tidy?

Does each variable form a column?

Yes, for example, Year is a variable so has a single column rather than a column for each year.

Does each observation form a row?

Yes. While there are multiple columns, they are each just clarifying the observed value rather than additional observations themselves (for example, confidence intervals).

Is each cell a single value?

Yes, for example, there are no commas separating values. The period column data is in the format "YYYY/YYYY financial year" but this is still a single value because financial years always span two sequential calendar years.

```r
unique_columns <- c("area_code", "year")
duplicates <- adp_data[duplicated(adp_data[, unique_columns]) | duplicated
                       (adp_data[, unique_columns], fromLast = TRUE), ]
print(duplicates)
```

```
## # A tibble: 0 x 12
## # i 12 variables: indicator <chr>, area_name <chr>, area_code <chr>,
## #   area_type <chr>, year <dbl>, period <chr>, numerator <dbl>, measure <dbl>,
## #   lower_confidence_interval <dbl>, upper_confidence_interval <dbl>,
## #   definition <chr>, data_source <chr>
```

```r
# This chunk is for selecting and renaming columns and removing the NHS prefix.

clean_data <- adp_data %>%
  select('area_name','year','numerator', 'measure') %>%
  rename(no_prescriptions = 'numerator', NHS = 'area_name',
         percentage_pop = 'measure') %>%
  mutate(NHS = sub("^NHS ","", NHS)) %>%

  # the below line is necessary to calculate the national population for
# recalculating percentages later
  mutate(board_population = no_prescriptions / percentage_pop * 100)
glimpse(clean_data)
```

```
## Rows: 180
## Columns: 5
## $ NHS              <chr> "Scotland", "Ayrshire & Arran", "Borders", "Dumfries ~
## $ year             <dbl> 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010,~
## $ no_prescriptions <dbl> 787040, 60822, 17226, 22280, 55334, 43976, 70337, 187~
## $ percentage_pop   <dbl> 14.96, 16.31, 15.15, 14.75, 15.26, 14.86, 12.45, 16.6~
## $ board_population <dbl> 5260962.57, 372912.32, 113702.97, 151050.85, 362608.1~
```

Step 3: Analyse

14 Health Boards are too many to plot in the same visualisation; the audience would be overwhelmed. The boards need to be merged into a small number of groups. This will also mean that their values need to recalculated so each group can be understood as a percentage of the population of Scotland.

```r
#This chunk groups the boards into regions
group_data <- clean_data %>%
  mutate(Region = case_when(
    NHS %in% c("Ayrshire & Arran" , "Borders" , "Dumfries & Galloway")
    ~ "Borders",
    NHS %in% c("Fife" , "Forth Valley" , "Greater Glasgow & Clyde" ,
               "Lanarkshire" , "Lothian")
    ~ "Central Belt",
    NHS %in% c("Grampian" , "Tayside")
    ~ "North East",
    NHS %in% c("Highland" , "Western Isles" , "Orkney" , "Shetland")
    ~ "Highlands & Islands",
    NHS %in% c("Scotland") ~ "Scotland"))
glimpse(group_data)
```

```
## Rows: 180
## Columns: 6
## $ NHS              <chr> "Scotland", "Ayrshire & Arran", "Borders", "Dumfries ~
## $ year             <dbl> 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010,~
## $ no_prescriptions <dbl> 787040, 60822, 17226, 22280, 55334, 43976, 70337, 187~
## $ percentage_pop   <dbl> 14.96, 16.31, 15.15, 14.75, 15.26, 14.86, 12.45, 16.6~
## $ board_population <dbl> 5260962.57, 372912.32, 113702.97, 151050.85, 362608.1~
## $ Region           <chr> "Scotland", "Borders", "Borders", "Borders", "Central~
```

```
# The below is all one chunk because it would not work in separate chunks.
# This was explained by ChatGPT (2023)

# This code was difficult to arrive at and could be reworked.

# The below creates a total number of prescriptions per region
regional_data <- group_data %>%
  group_by(Region, year) %>%
  summarise(board_population = sum(board_population), regional_prescriptions =
              sum(no_prescriptions))
```

```
## `summarise()` has grouped output by 'Region'. You can override using the
## `.groups` argument.
```

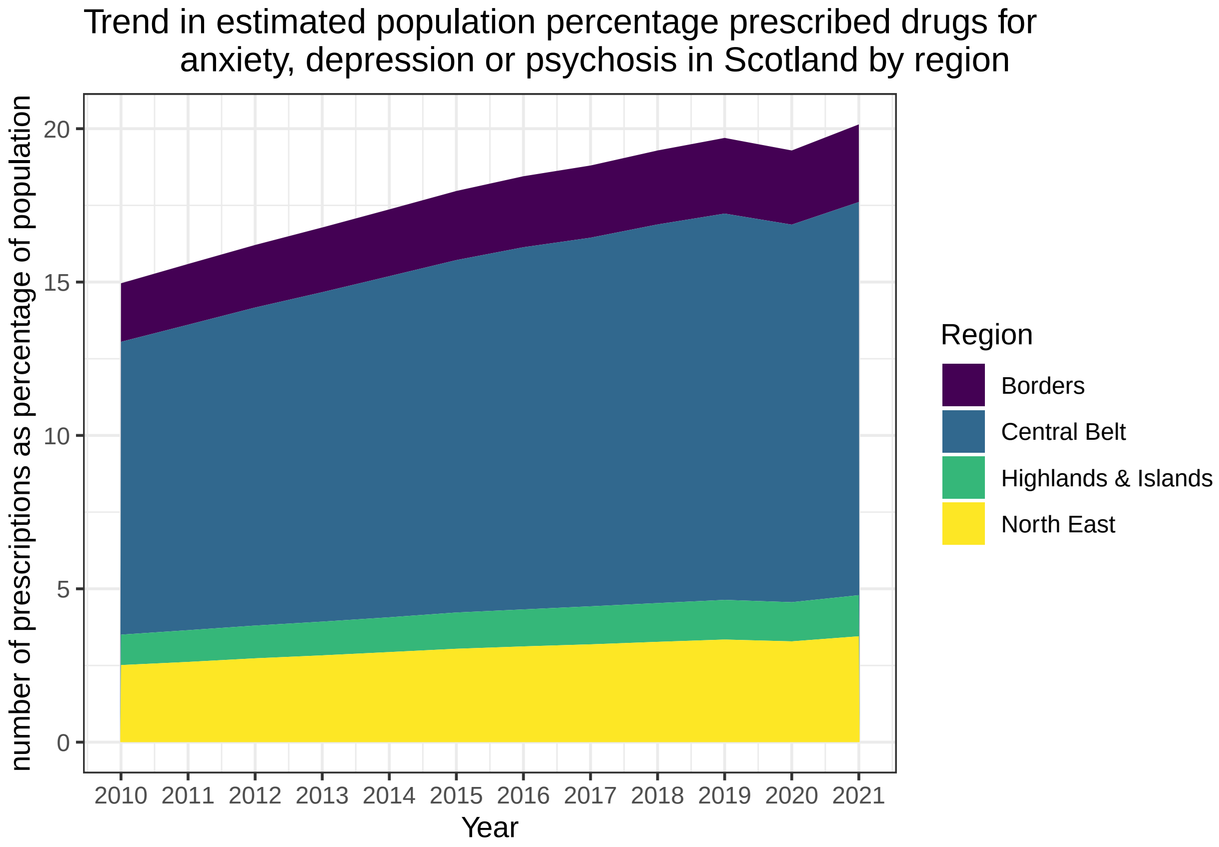
```
# The below creates a tibble for the annual calculated Scottish population
scotland_population <- regional_data %>% filter(Region == "Scotland") %>%
  rename(whole_population = board_population, scotland_prescriptions =
           regional_prescriptions)

# The below creates each regional number of prescriptions as a percentage of the
#scottish population for that year.
plot_data <- regional_data %>%
  left_join(scotland_population, by = 'year') %>%
  mutate(percentage_scot_pop = regional_prescriptions / whole_population * 100)
glimpse(plot_data)
```

```
## Rows: 60
## Columns: 8
## $ Region.x               <chr> "Borders", "Borders", "Borders", "Borders", "Bo~
## $ year                   <dbl> 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017,~
## $ board_population       <dbl> 637666.1, 639032.5, 637888.7, 636439.7, 635120.~
## $ regional_prescriptions <dbl> 100328, 104934, 108404, 112404, 116487, 121085,~
## $ Region.y               <chr> "Scotland", "Scotland", "Scotland", "Scotland",~
## $ whole_population       <dbl> 5260963, 5298679, 5314503, 5328123, 5347916, 53~
## $ scotland_prescriptions <dbl> 787040, 826064, 861481, 894059, 928933, 965638,~
## $ percentage_scot_pop    <dbl> 1.9070274, 1.9803805, 2.0397767, 2.1096361, 2.1~
```

Step 4: Communication

The final step is to generate the visualisation. This graph shows the increase in prescriptions for anxiety, depression and psychosis across Scotland from 2010-2021, broken down into four broad regions of Scotland.

```
plot_data %>%
  filter(Region.x != "Scotland") %>%
  ggplot(aes(x = year, y = percentage_scot_pop, fill = Region.x)) +
  geom_area()+
  labs(fill="Region")+
  xlab("Year")+
  ylab("number of prescriptions as percentage of population")+
  ggtitle("Trend in estimated population percentage prescribed drugs for
          anxiety, depression or psychosis in Scotland by region")+
  scale_fill_viridis(discrete=TRUE)+
  scale_x_continuous(breaks = unique(plot_data$year))+
  theme_bw()
```



Trend in estimated population percentage prescribed drugs for anxiety, depression or psychosis in Scotland by region

References

I acknowledge the use of ChatGTP (chat.openai.com) to troubleshoot coding issues on this assignment.

I entered the following prompt on 7 November 2023

> I am using ggplot to plot a stacked area chart. the plot is showing only select years' as ticks on the x-axis. how do i get it to label all the years? This is my ggplot code options(scipen = 999) summed_data %>% ggplot(aes(x = year, y = total_people, fill = Region)) + geom_area()+ xlab("Year")+ ylab("Estimated number of people")+

> ggtitle(“Trend in estimated population prescribed drugs for anxiety, depression or psychosis by Scottish region”)+ scale_fill_viridis(discrete=TRUE)+ theme_bw()

The content generated from this prompt was used to add all year labels to the axis.

I entered the following prompt on 19 November 2023:

> In the following sequence of coding, why am I getting an error that “object ‘regional_prescriptions’ not found” at the end, when that object is used earlier in the code? “‘{r} #This chunk creates a total number of prescriptions per region regional_data <- group_data %>% group_by(Region, year) %>% summarise(board_population = sum(board_population), regional_prescriptions = sum(no_prescriptions)) view(regional_data)

```
>>#This chunk creates a tibble for the annual calcuated Scottish population
scotland_population <- regional_data %>% filter(Region == "Scotland") %>%
  rename(whole_population = board_population)
view(scotland_population)
```

> plot_data <- regional_data %>% left_join(scotland_population, by = ‘year’) %>% mutate(percentage_scot_pop = regional_prescriptions / whole_population * 100) view(plot_data) Provide information on the ethical considerations when using generative artificial intelligence tools

The content generated from this prompt was used to fix errors in the code. The code was then further editeed.