Deaths from suicide in young people, aged 11-25 years

2024-11-21

The epidemic of suicide in Scotland, aged 11-25

Introduction

For my project I will be trying to answer the question: 'Is death by suicide rising in young people in Scotland?'. Suicide in young people is a tragic consequence of poor mental health, poor mental health services, and social impact on young people. Analyzing data from suicide rates could help us in the future to identify risk factors and decrease and prevent suicide.

Load packages

We first need to start by loading packages needed for the

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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
```

Data acquisition

The data I will be using in this report is the "Deaths by suicide in young people, aged 11-25 years" from Scotpho: https://scotland.shinyapps.io/ScotPHO_profiles_tool/

Read in data

Firstly the data is 'read in' as follows. The data I am using in this report contains a "period" column. This columns cells contain numbers that are a range of five years. To ease visual representation in my diagram I will be using the initial year the 'period' begins. Due to this code has been written to command only the 1-4 digits be used.

```
Suicide_data <- read.csv("~/Documents/Trends_ScotPHO_data_extract_2024-10-21.csv")

Suicide_data <- Suicide_data %>%
  mutate(year_start = substr(trend_axis, 1, 4))
```

Inspect Data

We use the glimpse function to initially inspect the data we have imported.

```
glimpse(Suicide_data)
```

```
## Rows: 17
## Columns: 12
## $ code                       <chr> "S00000001", "S00000001", "S00000001", "S000~
## $ areatype                   <chr> "Scotland", "Scotland", "Scotland", "Scotlan~
## $ areaname                   <chr> "Scotland", "Scotland", "Scotland", "Scotlan~
## $ indicator                  <chr> "Deaths from suicide in young people, aged 1~
## $ type_definition            <chr> "Crude rate per 100,000 population", "Crude ~
## $ definition_period          <chr> "2002 to 2006 calendar years; 5-year aggrega~
## $ trend_axis                 <chr> "2002-2006", "2003-2007", "2004-2008", "2005~
## $ numerator                  <dbl> 114.4, 112.6, 113.6, 113.8, 111.2, 116.2, 10~
## $ measure                    <dbl> 11.8, 11.5, 11.6, 11.6, 11.3, 11.7, 11.0, 10~
## $ upper_confidence_interval <dbl> 14.2, 13.9, 13.9, 13.9, 13.5, 14.1, 13.2, 12~
## $ lower_confidence_interval <dbl> 9.7, 9.5, 9.5, 9.5, 9.3, 9.7, 9.0, 8.4, 7.9,~
## $ year_start                 <chr> "2002", "2003", "2004", "2005", "2006", "200~
```

Data cleaning and preperation

Select trend_axis and measure

The data we are using in this report has 10 columns, we are most interested in two to make our graph from. Due to this we need to select and isolate the nesicary columns.

```
Deaths_from_suicide_in_young_people_aged_11_to_25_years <- Suicide_data %>%
  select(`trend_axis`,`measure`) %>%
  rename( `Frequency` = `trend_axis`,
          `Crude rate per 100000 population` = `measure`)
head(Deaths_from_suicide_in_young_people_aged_11_to_25_years)
```

```
##   Frequency Crude rate per 100000 population
## 1 2002-2006                              11.8
## 2 2003-2007                              11.5
## 3 2004-2008                              11.6
## 4 2005-2009                              11.6
## 5 2006-2010                              11.3
## 6 2007-2011                              11.7
```

Remove missing values

Missing values are then taken out from our data.In this data there have been no missing values or unique values.

```
Deaths_from_suicide_in_young_people_aged_11_to_25_years$`5-year period` %>% unique()
```

```
## NULL
```

```
Deaths_from_suicide_in_young_people_aged_11_to_25_years$`Crude rate per 100,000 population` %>% table()
```

```
## < table of extent 0 >
```

```
Deaths_from_suicide_in_young_people_aged_11_to_25_years %>% summary()
```

```
##   Frequency         Crude rate per 100000 population
##  Length:17          Min.   : 8.10
##  Class :character   1st Qu.: 9.70
```

```
##  Mode  :character   Median :10.90
##                     Mean   :10.46
##                     3rd Qu.:11.50
##                     Max.   :11.80
```

Check that data is tidy

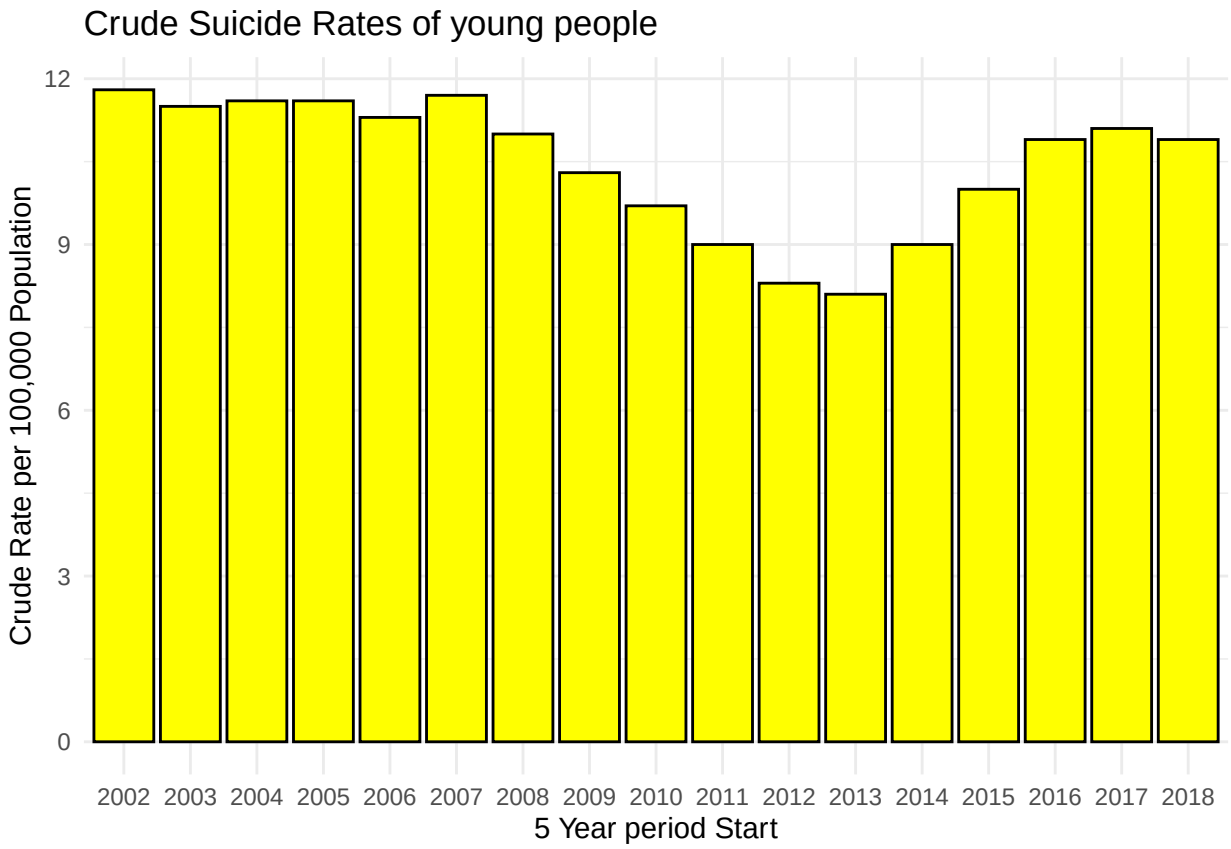
```
Suidicde_data <- Suicide_data %>%
  mutate(year_start = substr(trend_axis, 1, 4))
```

Data analysis and visualisation

Plot graph

The most efficient way to plot this graph is a bar chart, with colour to be more visually captivating. for this we use the ggplot and geom_col commands.

```
Suicide_data %>%
  ggplot(aes(x = year_start, y = measure)) +
  geom_col(fill = "yellow", colour = "black") +
  labs(
    title = "Crude Suicide Rates of young people",
    x = "5 Year period Start",
    y = "Crude Rate per 100,000 Population"
  ) +
  theme_minimal()
```

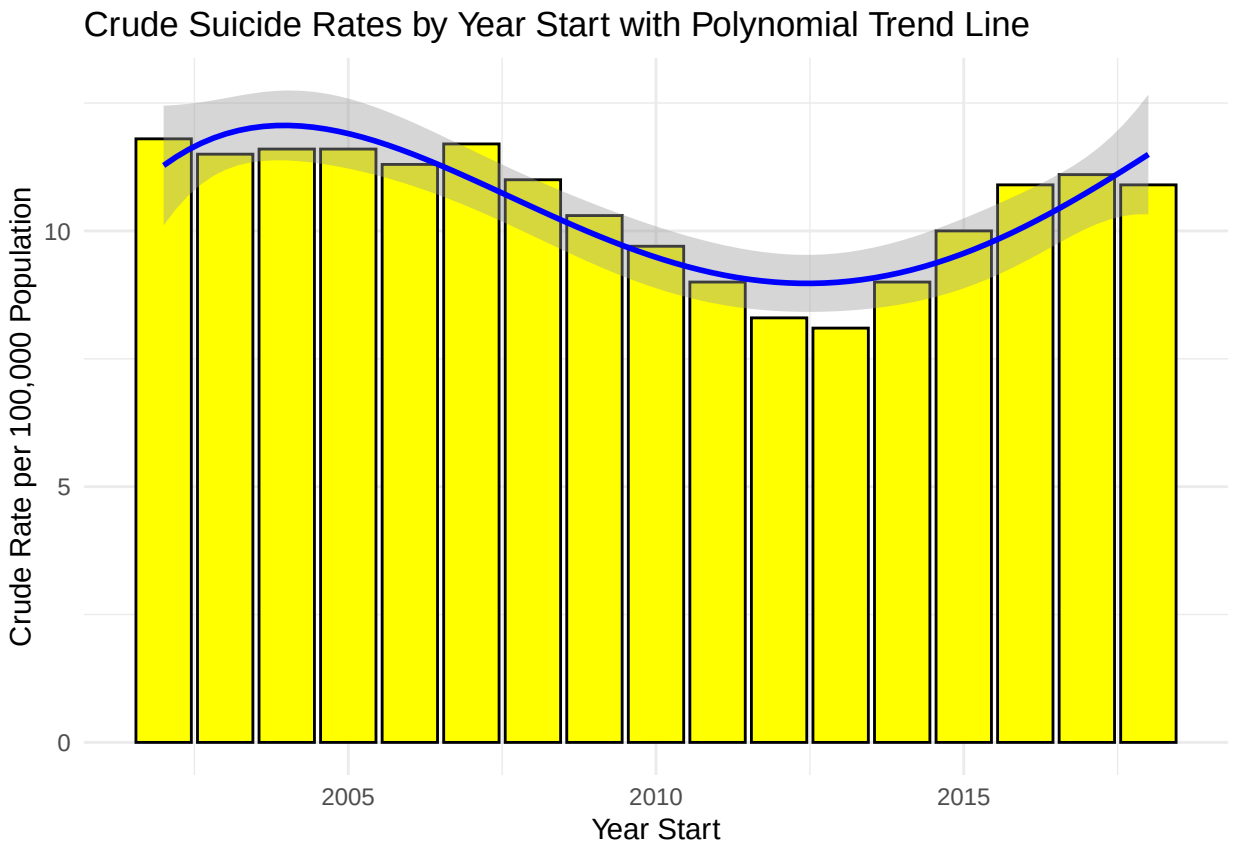


Graph with trend line

As the question of our report is; 'Is the rate of suicide in young people rising?' the most efficient way to visualise this in a graph is by adding a polynomial trend line. This way it is easy at first glance to appreciate the movement in values.

```
Suicide_data %>%
  ggplot(aes(x = as.numeric(year_start), y = measure)) +
  geom_col(fill = "yellow", colour = "black") +
  geom_smooth(method = "lm", formula = y ~ poly(x, 4), se = TRUE, colour = "blue", size = 1) +
  labs(
    title = "Crude Suicide Rates by Year Start with Polynomial Trend Line",
    x = "Year Start",
    y = "Crude Rate per 100,000 Population"
  ) +
  theme_minimal()
```

```
## 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.
```



Summary