Commom causes of unintentional injuries in Scotland

Patterns in demografic groups and death rates on 2011-2022 data

Inma Ruiz

24-04-2023

Contents

1 Overview

In this report we are going to analyse data on admissions to hospital and deaths in Scotland from unintentional injuries and assaults covering a ten year period (2011 - 2020). The analysis is going to be centered on the task of exploring which types of injures were the most common cause for admissions according to age and sex and on finding the death rates in admissions resulted for those main causes. Additionally, we will have a close look into the injury cause with the highest death ratio after admission.

2 Data Processing

We load the software packages we are going to use

```
# Mount packages
library(tidyverse)
library(janitor)
library(lubridate)
library(kableExtra)
library(formatR)
library(scales)
```

About the data

To perform this report, two datasets has been retrieved from The Scottish Health and Social Care Open Data platform for their analysis:

1. Admissions, Containing 391104 observations with information for 14 variables on emergency hospital admissions as a result of unintentional injuries and assaults.
2. Deaths, Containing 1825202 observations with information for 14 variables on deaths as a result of an unintentional injury.

```
#read in .csv files with the data

raw_admissions <- read_csv("https://www.opendata.nhs.scot/dataset/b0135993-3d8a-4f3b-afcf-e01f4d52137c/

raw_deaths <- read_csv("https://www.opendata.nhs.scot/dataset/b0135993-3d8a-4f3b-afcf-e01f4d52137c/reso
```

Exploratory Data Analysis

We started exploring the data by reading the Data Dictionary provided in the plataform and by opening both csv files in the R Studio environment, where we checked number of rows and columns, variable names and types of variables. We noticed a mixed naming style on variables and decided to use the clean_names() function right at the begging, creating two new objects: **admissions** and **deaths** that will be our main data feed for all the wrangling and analysis. After cleaning the names, we checked again our new created objects to see the new variable names. In this stage we also looked for variables susceptible to be used as the joining argument, and problems that we may encounter (eg: *financial_year* in *admissions* dataset is a character type variable where *year* in *deaths* dataset is numeric type).

```
#To have all variables with the same naming style
admissions <- raw_admissions %>%  clean_names()
deaths <- raw_deaths %>% clean_names()
```

Possible values for all categorical variables were systematically explored with the distinct(), count() or unique() functions. One example included in the code below, where we can see injury_types names varies in both datasets. We also found duplication in the data due to aggregation of categories in an additional categories named as "All" (or similar). Although we noticed that the category referred as "All" was not always equal to the sums of the other, in any case, we considered we can safely exclude those observations from the analysis by subseting only the categories we want to explore.

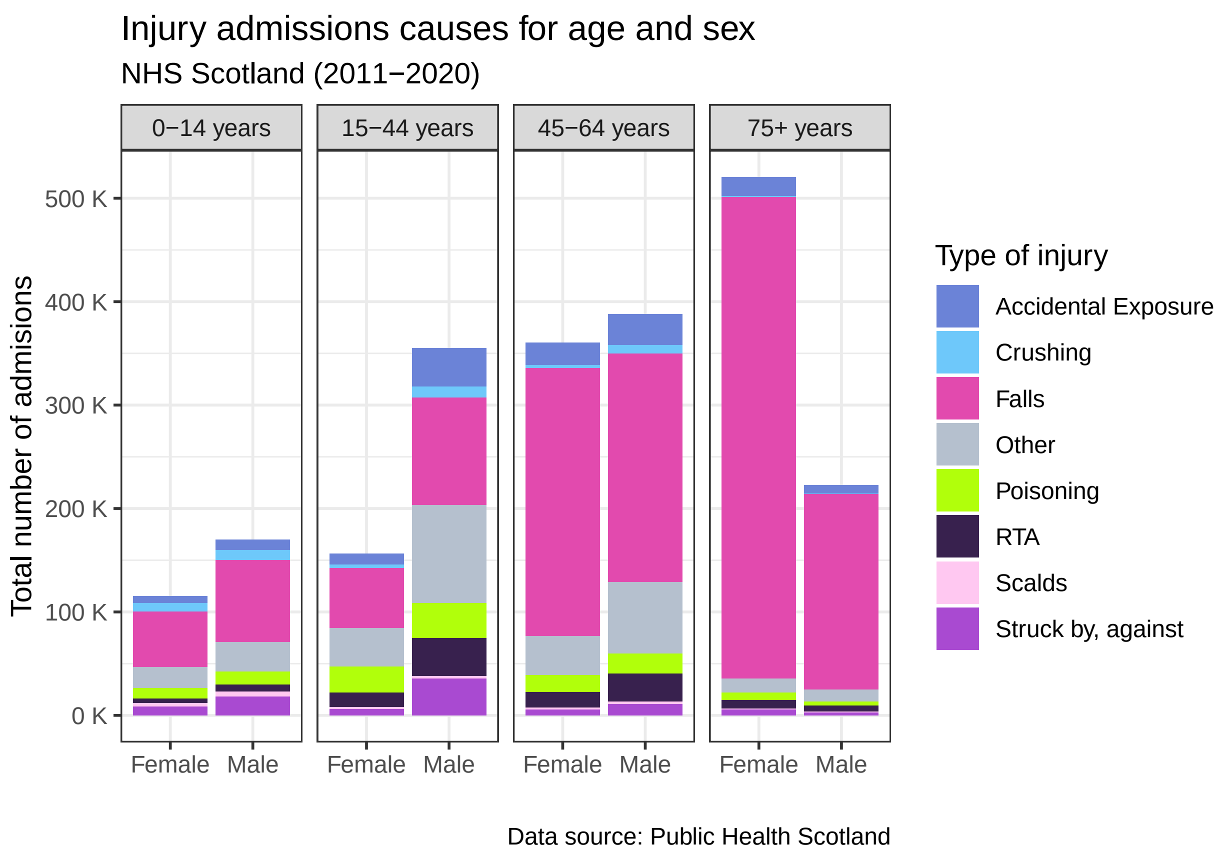
```
# example of how columns were explored to find its unique categories
admissions %>%
  distinct(injury_type)
```

```
## # A tibble: 9 x 1
##   injury_type
##   <chr>
## 1 All Diagnoses
## 2 RTA
## 3 Poisoning
## 4 Falls
## 5 Struck by, against
## 6 Crushing
## 7 Scalds
## 8 Accidental Exposure
## 9 Other
```

Most common admissions injuries for sex and age group.

For finding and visualizing the most common types injures by age group and sex we have use the data available for the full 10 years period. We have reduce the partitions for the age group aggregation into four groups, making its visualization and understanding more intuitive and less confusing: The wider groups created 0-14, 15-44, 45-65, 75+ can easily be related with common vulnerabilities, behaviors or risk exposures in children, young adults, mature adults or senior citizens.

```
#color palette for the plot (more contrasted that the default ggplot palette).
mycolors <- c(
  "#6b83d7","#6ec8fa","#e24aae", "#b5c0ce", "#b1ff0b", "#38214e", "#ffc8f1", "#a94ad1"
          )
#subseting the data for the plot, selecting and changing some variable types.
admissions %>%
  separate(financial_year, into = c("year", NA), sep = "/") %>%
  transmute(
    year,
    sex = as.factor(sex),
    age_group = as.factor(age_group),
    injury_type = as.factor(injury_type),
    number_of_admissions
  ) %>%
  # filtering to eliminate aggregated data
  filter(age_group != "All" &
           sex != "All" &  injury_type != "All Diagnoses") %>%
  # Reducing age groups into 4 with a wider age range resulted in a better
  # grasp of similar risky behaviors/exposure/vulnerabilities.
  mutate(
    age_group = case_when(
      age_group == "0-4 years" ~ "0-14 years",
      age_group == "5-9 years" ~ "0-14 years",
      age_group == "10-14 years" ~ "0-14 years",
      age_group == "15-24 years" ~ "15-44 years",
      age_group == "25-44 years" ~ "15-44 years",
      age_group == "45-64 years" ~ "45-64 years",
      age_group == "65-74 years" ~ "45-64 years",
      age_group == "75plus years" ~ "75+ years")) %>%
  group_by(injury_type, age_group, sex) %>%
  summarise(total = sum(number_of_admissions)) %>%
  ggplot(aes(sex, total, fill = injury_type)) +
  geom_col() + scale_fill_manual(values=mycolors) +
  facet_wrap(~  age_group, nrow = 1) +
  scale_y_continuous(labels = label_number(suffix = " K", scale = 1e-3)) +
  labs(
    title = "Injury admissions causes for age and sex",
    subtitle = "NHS Scotland (2011-2020)",
    caption = "Data source: Public Health Scotland",
    y = "Total number of admisions",
    x = " ",
    fill = "Type of injury" ) +
  theme_bw()
```



Rate of Death

In this piece of code we first created the table **"total_admissions"** with all admissions grouped by injury type. We had to change some names of the categories in order to match those on the second dataset. Then, we created a second table **"total_deaths"** with number of deaths by injury type. Both newly created tables were joined into a new one called **admissions_deaths**. Death rates in admissions were calculated for each injury cause with the mutate() function. The resulting ranking table is shown below.

```
#create a table with number of admission per type of injury
total_admissions <- admissions %>%
  separate(financial_year, into = c("year", NA), sep = "/") %>%
  select(sex, age_group, injury_type, number_of_admissions) %>%
  #categories in both datasets need to have same names
  mutate(injury_type = str_replace(injury_type, "RTA", "Land transport accidents"),
         injury_type = str_replace(injury_type, "Struck by, against", "Struck by,against"),
         injury_type = str_replace(injury_type, "Accidental Exposure", "Accidental exposure")
         ) %>%
  #filtering to eliminate duplicated data
  filter(age_group != "All" & sex != "All" & injury_type != "All Diagnoses") %>%
  group_by(injury_type) %>%
  summarise(total_admissions = sum(number_of_admissions, na.rm=TRUE))


#create a table with number of death per type of injury
total_deaths <- deaths %>%
  select(sex, age_group, injury_type, numberof_deaths) %>%
  #filtering to eliminate duplicated data
```

```
  filter(age_group != "All" &  sex != "All" &  injury_type != "All") %>%
  group_by(injury_type) %>%
  summarise(total_deaths = sum(numberof_deaths, na.rm=TRUE))


#join both tables and calculate rate of death in admissions
#per type of injury
admissions_deaths <- total_admissions %>%
  left_join(total_deaths, by = c("injury_type")) %>%
  mutate(death_rate = total_deaths/total_admissions) %>%
  #ordering the values in descending order by death ratio
  arrange(desc(death_rate))

#to display the table limiting decimals to 3
kable(admissions_deaths,
      caption = "Death rates in Scotland for injury types. 2011-2020",
      digits = 3) %>%
  kable_styling(latex_options = c("HOLD_position"), font_size = 12)
```

Table 1: Death rates in Scotland for injury types. 2011-2020

injury_type	total_admissions	total_deaths	death_rate
Poisoning	129056	36524	0.283
Land transport accidents	117876	6914	0.059
Accidental exposure	143044	3784	0.026
Falls	1429668	34816	0.024
Other	312396	6424	0.021
Scalds	18764	44	0.002
Struck by,against	92744	120	0.001
Crushing	44492	44	0.001

3 Results

1. **Having a fall was the most common reason for hospital admission** for all age groups and sexes.

2. **For under 75 years old, there have been more hospital admissions for males than for females**, being this difference greater for the age group 15-44 years. This may be due to males more prone to engage themselves in risky activities and behaviors than their females counterparts.

3. **For the 75+ years group, there have been more admissions of females than males**. Probably due to the higher proportion of female versus males in the total population within this age bracket.

4. Males between 15 to 44 years have the higher total number of admissions for poisoning, accidental exposure, other injuries, traffic accident and Struct by.

5. Among all the unintentional injury causes registered in admission, **poisoning have the higher death rate (0.267)**, calculated for the 10 years data. Followed by transport accidents (death rate = 0.062) and falls (death rate = 0.024).

4 Futher exploration based on results

As poisoning was by far the type of injury with the highest death rate in admissions among all injury types, further exploration focused in this cause were performed. We decided to explore how admissions, deaths and death rates differ across all the NHS health boards.

For this additional analysis we have used geographical spatial data for the Scottish Health Boards, a ESRI Shape file spatial data defining the boundaries of NHS Health Boards in Scotland, Available open source from the Spatial Data Metadata Portal, Scotland's catalogue of spatial data.

We load additional libraries for reading a dealing with this type of data

```
# Load packages
library(rgdal)
library(sp)
library(sf)
library(gridExtra)
library(latticeExtra)
```

Death rates in admission after poisoning by NHS Scottish Health Boards

We have calculated total admissions for poisoning for each health board form the data in the **admissions** dataset, and total deaths for poisoning for each health board from the data in the **deaths** dataset. Then, joined both together.

```
#create a table with total poison adissions by health board
poison_admissions_hb <- admissions %>%
  separate(financial_year, into = c("year", NA), sep = "/") %>%
  select ( year, hbr, sex, age_group, injury_type, number_of_admissions) %>%
  #filtering to eliminate aggregated data and subseting poisoning
  filter(age_group != "All" & sex != "All" & hbr != "S92000003" &
           injury_type == "Poisoning") %>%
  group_by(hbr)%>%
  summarise(total_poison_admissions = sum(number_of_admissions, na.rm=TRUE))

#create a table with number of deaths due to poisoning in each Health Board
poison_deaths_hb <- deaths %>%
  select (year, hbr, sex, age_group, injury_type, numberof_deaths) %>%
  #filtering to eliminate aggregated data and subseting poisoning
  filter(age_group != "All" & sex != "All" & hbr != "S92000003" &
           injury_type == "Poisoning") %>%
  group_by(hbr)%>%
  summarise(total_poison_deaths = sum(numberof_deaths, na.rm=TRUE))

#create a table by joining both tables
#and add column with rate of death calculation per Heath Board
poison_deaths_rates_hb <- poison_admissions_hb %>%
  left_join(poison_deaths_hb, by = c("hbr")) %>%
  mutate(death_rate = total_poison_deaths/total_poison_admissions)

#we are not printing the table due to space limitation (You can print it by deleting the hashtag)
#poison_deaths_rates_hb
```

In this chunk of code we read the .shp file containing the vector data with the health board boundaries and join our newly created table *poison_deaths_rates_hb* to it.

```r
#read in .shp file
scotland_hb <- st_read("C:/Users/Casa/Desktop/IRM/Injuries/SG_NHS_HealthBoards_2019.shp")
```

```
## Reading layer 'SG_NHS_HealthBoards_2019' from data source
##   'C:\Users\Casa\Desktop\IRM\Injuries\SG_NHS_HealthBoards_2019.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 14 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 5512.998 ymin: 530250.8 xmax: 470332 ymax: 1220302
## Projected CRS: OSGB36 / British National Grid
```

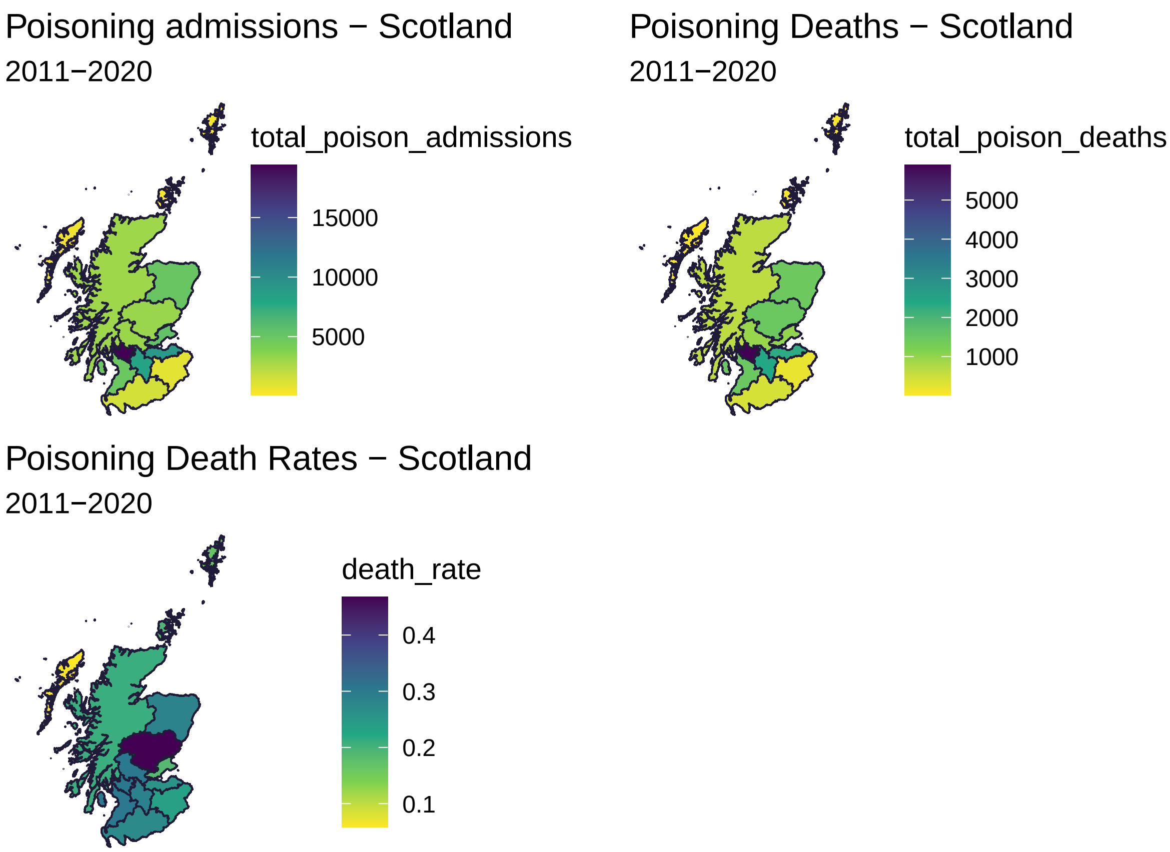
```r
#join the spatial data with the poisoning data
join_data <- scotland_hb %>%
inner_join(poison_deaths_rates_hb, by = c("HBCode" = "hbr"))
```

We create the 3 map plots

```r
map3 <- ggplot(join_data, aes(fill = death_rate)) +
  geom_sf(size = 0.3, color = "#1f1b39") +
  scale_fill_viridis_c(option = "viridis", direction = -1) +
    labs(
    title = "Poisoning Death Rates - Scotland",
    subtitle = "2011-2020") +
  coord_sf() +
  theme_void()

map2 <- ggplot(join_data, aes(fill = total_poison_deaths)) +
  geom_sf(size = 0.3, color = "#1f1b39") +
  scale_fill_viridis_c(option = "viridis", direction = -1) +
    labs(
    title = "Poisoning Deaths - Scotland",
    subtitle = "2011-2020") +
  coord_sf() +
  theme_void()

map1 <- ggplot(join_data, aes(fill = total_poison_admissions)) +
  geom_sf(size = 0.3, color = "#1f1b39") +
  scale_fill_viridis_c(option = "viridis", direction = -1) +
    labs(
    title = "Poisoning admissions - Scotland",
    subtitle = "2011-2020") +
  coord_sf() +
  theme_void()
```

We print the 3 maps plots together as one figure for comparing the visualization.

```r
#library containing hte function plot_grid()
library(cowplot)

#plotting the 3 maps together
plot_grid(map1, map2, map3, align = "hv")
```



Data sources: Public health Scotland and Scottish Government open data

Findings

Regarding to poisoning admissions during the period 2011 and 2020 in Scotland (Values extracted from the table that feeds the maps figure: **poison_deaths_rates_hb**):

1. NHS Greater Glasgow and Clyde has the highest number of admissions (18744)
2. NHS Orkney has the lowest number of admissions (110)
3. NHS Greater Glasgow and Clyde has the highest number of deaths (5414)
4. NHS Orkney has the lowest number of deaths (20)
5. NHS Tayside has the **highest death rate** (0.477)
6. NHS Western Isles has **the lowest death rate** (0.071)

5 Software and packages used

R: R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URLcitatiohttps://www.R-project.org/.

janitor: Firke S (2021). *janitor: Simple Tools for Examining and Cleaning Dirty Data*. R package version 2.1.0, https://CRAN.R-project.org/package=janitor..

Tidyverse: Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." *Journal of Open Source Software*, *4*(43), 1686. doi:10.21105/joss.01686 https://doi.org/10.21105/joss.01686.

Knitr: Yihui Xie (2022). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.40. H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016

kableExtra: Zhu H (2021). *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*. R package version 1.3.4, https://CRAN.R-project.org/package=kableExtra.

ggplot: H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016

formatR Xie Y (2023). *formatR: Format R Code Automatically*. R package version 1.14, https://CRAN.R-project.org/package=formatR.

lubridate Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL https://www.jstatsoft.org/v40/i03/.

rgdal Bivand R, Keitt T, Rowlingson B (2023). *rgdal: Bindings for the 'Geospatial' Data Abstraction Library*. R package version 1.6-4, https://CRAN.R-project.org/package=rgdal.

sp Pebesma, E.J., R.S. Bivand, 2005. Classes and methods for spatial data in R. R News 5 (2), https://cran.r-project.org/doc/Rnews/. Roger S. Bivand, Edzer Pebesma, Virgilio Gomez-Rubio, 2013. Applied spatial data analysis with R, Second edition. Springer, NY. https://asdar-book.org/

sf Pebesma, E., 2018. Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10 (1), 439-446, https://doi.org/10.32614/RJ-2018-00.

gridExtra Auguie B (2017). *gridExtra: Miscellaneous Functions for "Grid" Graphics*. R package version 2.3, https://CRAN.R-project.org/package=gridExtra.

laticeExtra Sarkar D, Andrews F (2022). *latticeExtra: Extra Graphical Utilities Based on Lattice*. R package version 0.6-30, https://CRAN.R-project.org/package=latticeExtra.

cowplot Wilke C (2020). *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*. R package version 1.1.1, https://CRAN.R-project.org/package=cowplot.

Spatial Data Metadata Portal, Scotland's catalogue of spatial data.