Observations of Injury Types Amongst NHS Patients Residing in Scotland (2011-2022)

Exam Number: B215161

23/06/2023

Contents

##Title: Observations of Injury Types Amongst NHS Patients Residing in Scotland (2011-2022)

##Overview: The data presented is a joined data set using three csv files provided by different NHS Services across Scotland. This data has been collected over the years of 2011 to 2022 and includes demographics defined by sex (male and female) and age groups (0-4 years, 5-9 years, 10-14 years, 15-24 years, 25-44 years, 45-64 years, 65-74 years, 75+ years))

#Most Common Injury Observed amongst Age and Sex: It can be observed from the data sets, which span from 2011-2022 the most frequent injury suffered is has been recorded as a falling injury, the demographic who has been particularly affected by this type of accident is specifically females aged 75+. Followed by male and females aged between 45 to 65 years. The graph has been plotted using a logarithmic scale due to the high volume of admission data collected concerning falling injury. It is clear to see that this type of injury is the most prevalent amongst the Scottish NHS services.

#Rate of Death by Admission for Falling Injuries: It can be observed that there is a positive correlation between the rate of death by quantity of admissions. This does not mean however the quantity of deaths is the same as the quantity of admissions, but that there is more deaths recorded through a greater number of recorded admissions across the NHS services of Scotland between the years 2011-2022.

##Data Processing:

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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(lubridate)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
```

```
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(dplyr)
library(ggplot2)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(viridis)
```

```
## Loading required package: viridisLite
##
## Attaching package: 'viridis'
##
## The following object is masked from 'package:scales':
##
##     viridis_pal
```

```
#Calling necessary packages for RMarkdown.

#getwd() if necessary
#setwd() if necessary

orig_hb = read.csv('hb_lookup.csv')
orig_uiadmissions = read.csv('ui_admissions_2022.csv')
```

```
orig_uideaths = read.csv('ui_deaths_2022.csv')

#CSV files read into RMarkdown and assigned to object names for ease of use.

#glimpse(orig_hb)
#Rows: 18 Columns: 5, HB numbers + name(location in Scotland). Couple of "NA"s in column 4, consistent

#glimpse(orig_uiadmissions)
#Rows: 391,104 Columns: 14, Breakdown of HB, Year, Age, Sex, Injurylocation, Injurytype, Number of Admi

#glimpse(orig_uideaths)
#Rows: 182,502 Columns: 14, Breakdown of HB, Year, Age, Sex, Injurylocation, Injurytype, Number of Deat

#summary(orig_hb)
#HB and HBName will be required for joining datasets after cleaning.

#summary(orig_uiadmissions)
#NumberOFAdmissions Max. 61279
#Column 1, Financial Year is a character string column not a integer: requires mutation.
#"HBR" column same as orig_hb co dataset column "HB"

#summary(orig_uideaths)
#NumberofDeaths Max. 2759

#Initial observations of data sets show a few discrepancies
```

```
cleaned_admission <- orig_uiadmissions %>%
  select( FinancialYear,
        HBR,
        CA,
        AgeGroup,
        Sex,
        InjuryType,
        InjuryLocation,
        NumberOfAdmissions) %>%
  clean_names()  %>%
  separate(financial_year, into = c("year"), sep = "/") %>%
  filter(injury_type != "All Diagnoses") %>%
  filter(injury_location != "All") %>%
  filter(age_group != "All") %>%
  filter(sex != "All") %>%
  mutate_at(c(1), as.numeric)
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 391104 rows [1, 2, 3, 4, 5,
## 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
#orig_uiadmissions file assigned to new object after cleaning
#select() function used to organise and select relevant columns from dataset: info, demographic, Injury
#clean_names() function used to make snake case, lower case names for uniformity
#separate() function to separate 2011/2012 and other values present in FinancialYear column into indivi
#filter() function used to omit (!=) information or highlight (==)
#mutate_at(c(1), as.numeric) to change character values to numeric ones as column represent "Financial

cleaned_deaths <- orig_uideaths %>%
```

```
  select( Year,
       HBR,
       CA,
       AgeGroup,
       Sex,
       InjuryType,
       InjuryLocation,
       NumberofDeaths) %>%
  clean_names() %>%
  filter(injury_type != "All") %>%
  filter(injury_location != "All") %>%
  filter(age_group != "All") %>%
  filter(sex != "All")
#Process from orig_uiadmissions repeated

cleaned_hb <- orig_hb %>%
  select( HB,
       HBName,
       Country) %>%
  clean_names()
#Process from orig_uiadmissions repeated
```

```
full_ui <- cleaned_admission %>%
  full_join(cleaned_deaths)
```

```
## Joining with `by = join_by(year, hbr, ca, age_group, sex, injury_type,
## injury_location)`
```

```
#full_join() used to merge cleaned datasets together. full_join() keeps values from both, easier to obs

full_ui_hb <- full_ui %>%
  full_join(cleaned_hb, by = c("hbr" = "hb"))

#full_join() of final dataset. All datasets are now merged.
# glimpse(full_ui_hb)
# is.na(full_ui_hb)

clean_full_ui <- full_ui_hb %>%
  select(year,
      hbr,
      hb_name,
      age_group,
      sex,
      injury_type,
      injury_location,
      number_of_admissions,
      numberof_deaths)  %>%
  mutate(numberof_deaths = coalesce(numberof_deaths, 0)) %>%
  mutate(across(c(injury_location), na_if, "Not Applicable")) %>%
  mutate(across(c(number_of_admissions), na_if, 0)) %>%
  na.omit(clean_full_ui) %>%
  group_by(sex, age_group, injury_type, injury_location, year, hbr, hb_name) %>%
  summarise(across(c(number_of_admissions, numberof_deaths), sum)) %>%
  arrange(age_group) %>%
```

```
  distinct() %>%
  rowid_to_column()
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(c(injury_location), na_if, "Not Applicable")`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'sex', 'age_group', 'injury_type',
## 'injury_location', 'year', 'hbr'. You can override using the `.groups`
## argument.
```

```
#Tidying and wrangling process to make data set ready for plotting.

#Final modifications and wrangling of data.
#Selected relevant columns for data plotting.
#Mutate() and Coalesce() to change NA values in numberof_deaths column to numerical value 0.
#Mutate(), across() to na_if functions to change character value "Not Applicable" to null value in colu
#na.omit() to omit null, incomplete data present in merged data.
#rowid_to_column() to number each line of the dataset for navigation and ease of use by adding integer
#group_by() to organise the columns into a desired order

#This dataset is cleaned and ready to be observed for any last discrepancies before being plotted.
```

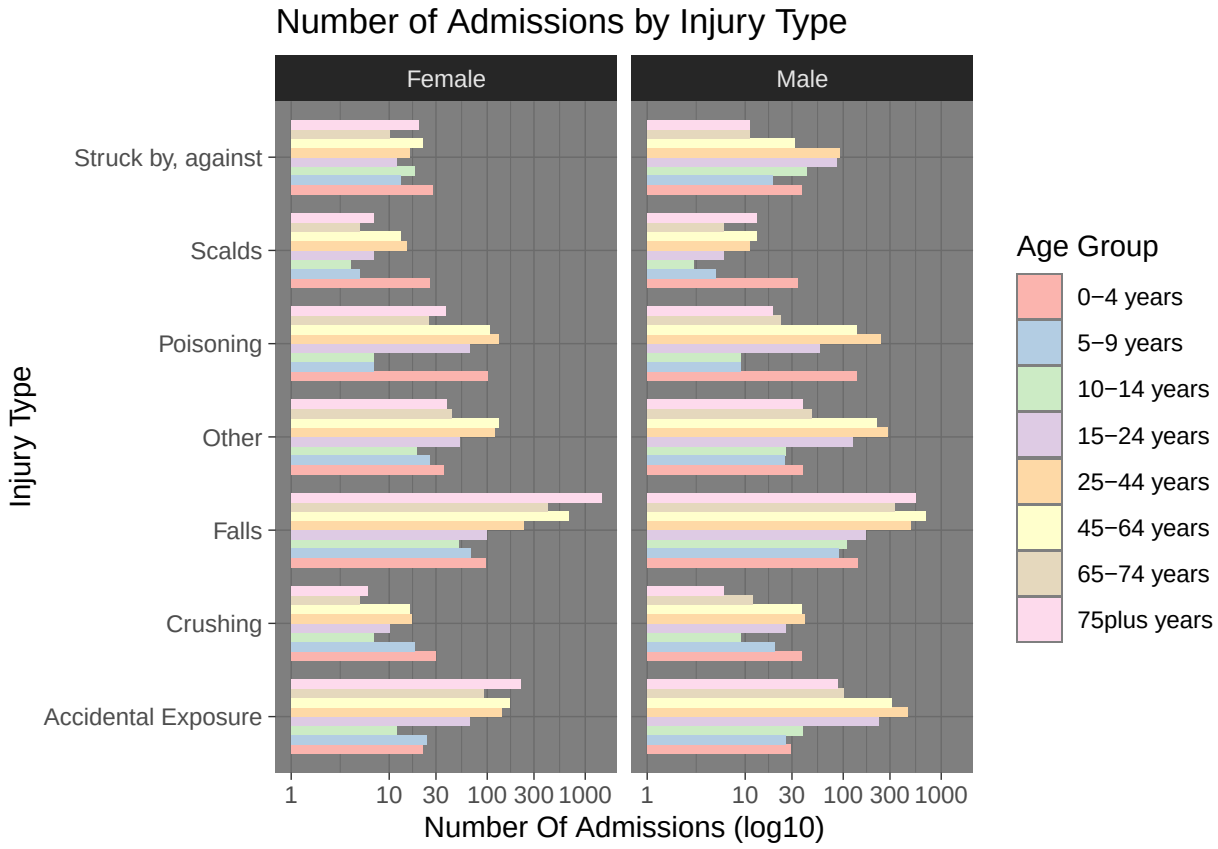
```
glimpse(clean_full_ui)

clean_full_ui %>%
  ungroup() %>%
  count(year)

clean_full_ui$year %>%
  is.numeric()
# Observations: range = 2011 to 2020, numeric values range = 2600 to 2900

clean_full_ui %>%
  ungroup() %>%
  count(sex)
# Female    13247, Male 14933

clean_full_ui %>%
  ungroup() %>%
  count(age_group)
# #0-4 years    3570
# 5-9 years 3008
# 10-14 years   2793
# 15-24 years   3551
# 25-44 years   4123
# 45-64 years   4217
```

```
# 65-74 years   3337
# 75plus years  3581

clean_full_ui %>%
  ungroup() %>%
  count(injury_type)
# #Accidental Exposure  3684
# Crushing  3120
# Falls 6246
# Other 5236
# Poisoning 3650
# Scalds    1832
# Struck by, against    4412

clean_full_ui %>%
  ungroup() %>%
  count(injury_location)
# Home  9903
# Other 7668
# Undisclosed   10609

clean_full_ui %>%
  ungroup() %>%
  count(hb_name)
#14 rows, Different NHS Services

clean_full_ui %>%
  ungroup() %>%
  count(number_of_admissions, sex)

clean_full_ui$number_of_admissions %>%
  is.numeric()

clean_full_ui %>%
  ungroup() %>%
  count(numberof_deaths, sex)

clean_full_ui$numberof_deaths %>%
  is.numeric()

# unique(clean_full_ui)
summary(clean_full_ui)

#Final inspection of dataset before plotting. count() is used to check and determine total variables pr
```

```
clean_full_ui$age_group <- factor(clean_full_ui$age_group, levels=c("0-4 years", "5-9 years", "10-14 ye

clean_full_ui_filtered <- clean_full_ui %>%
  filter(injury_type == "Falls")

#Final reorganizing of data as some discrepancies were observed in previous observations, including inc

#The final filtered data set is made after the penultimate data observations in the first graph have sh
```

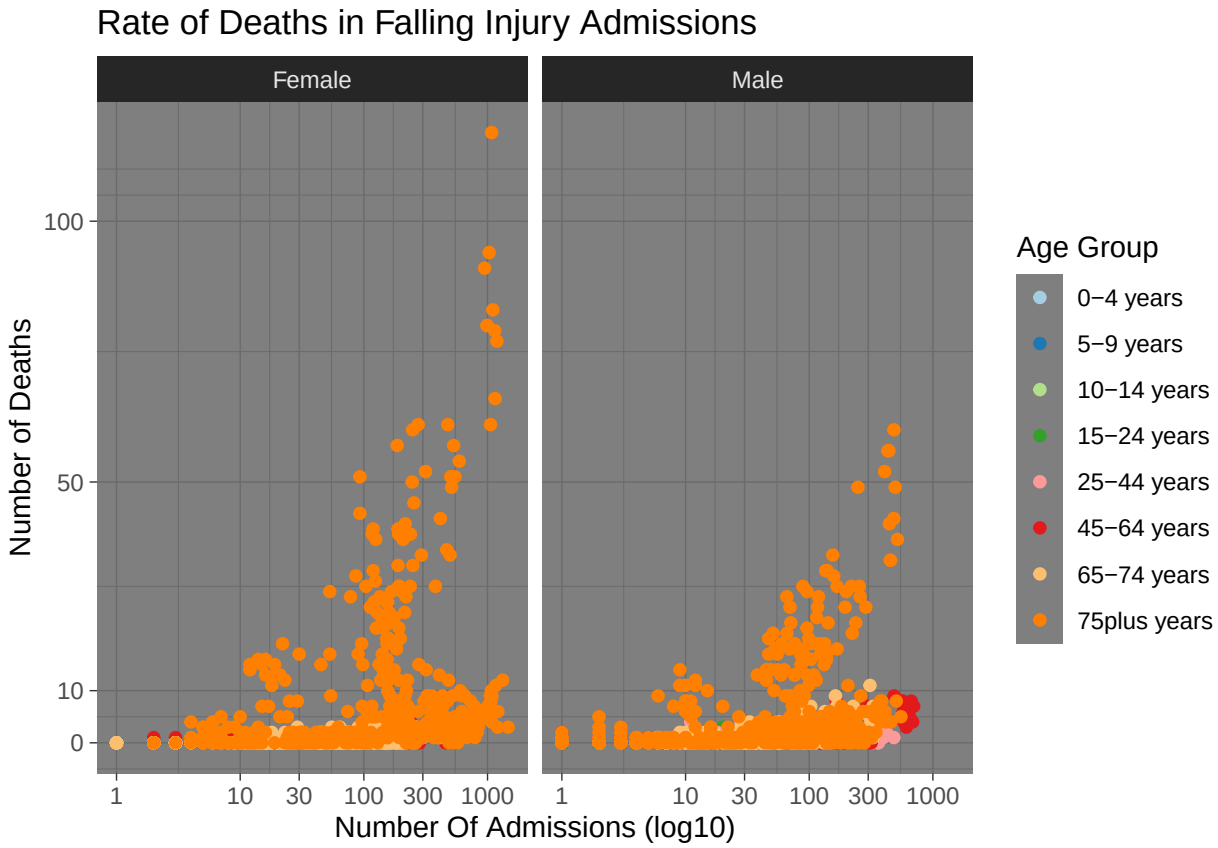
##Results:

```
#The final, cleaned and organised first dataset is piped through the ggplot function, with appropriate

clean_full_ui %>%
  ggplot(aes(x=injury_type, y=number_of_admissions,  fill=age_group)) +
  geom_bar(position = "dodge",
        stat = "identity",
        width = 0.8) +
  scale_fill_brewer(palette = "Pastel1") +
  theme_dark() +
  ggtitle("Number of Admissions by Injury Type") +
  xlab("Injury Type") +
  scale_y_log10(name = "Number Of Admissions (log10)",
          limits = c(),
          breaks = c(1, 10, 30, 100, 300, 1000, 10000)) +
  facet_wrap(~sex, ncol =10) +
  coord_flip() +
  guides(fill=guide_legend(title = "Age Group"))
```



Summary: It can be observed from this grouped bar chart, that the highest rate of admission over 2011 to 2020 was from females, aged 75+ years who had suffered from a falling injury. Followed by males and females of 45-65 years who had been admitted due to suffering from a falling injury. The X axis “Number of Admissions” was plotted using a logarithmic scale due to the extremely high volume of admissions recorded for falling injuries - as can be observed most other injuries, except “Accidental Exposure” and “Falls” do not scale above 300 in data concerning Number Of Admissions.

```
clean_full_ui_filtered %>%
  ggplot(aes(x=numberof_deaths, y=number_of_admissions, color=age_group)) +
  geom_point(alpha = 1, size = 1.7) +
  scale_color_brewer(palette = "Paired") +
  theme_dark() +
  ggtitle("Rate of Deaths in Falling Injury Admissions") +
  xlab("Number of Deaths") +
  scale_x_continuous(limits = c(),
          breaks = c(0, 10, 50, 100, 150, 200, 250, 300),
          minor_breaks = waiver()) +
  scale_y_log10(name = "Number Of Admissions (log10)",
          limits = c(),
          breaks = c(1, 10, 30, 100, 300, 1000, 10000)) +
  facet_wrap(~sex, ncol = NULL) +
  coord_flip() +
  guides(color=guide_legend(title = "Age Group"))
```



Summary: The amount of death for falling injury admission can be observed as low in comparison to quantity of admission. However, there is a positive correlation between the recordings of admissions and death by falling injury, in particular for females patients, aged 75+ years. This is an understandable observation and confirms the observations made in the previous plot. It can be therefore deduced that though the rate of deaths is low by quantity comparison of admission data, a positive correlation does exist proving that more deaths occur from falling injury if there are higher rates of admissions.

##Conclusion To conclude, these data observations are limited to the NHS services who have provided them and are subject to potential scrutiny regarding the circumstances upon which the data was recorded, ie potential human error. Most data processed has been compiled by omitting data that is not complete or

completely descriptive in its nature, this include “Non Applicable” data or NA results. Furthermore, the NHS services specified within the data set have registered at different times, which will mean collected data is not uniform in its range - data cleaning and wrangling has been performed as much as possible to compensate for this, resulting in unusable data. This has not however interfered with the results acquired through analyzing the data as the results show a clear spike in recorded information, showing observable patterns in injuries and demographics residing within Scotland.