Tutorial 4 - Week 8

Dimitrios Doudesis

04-11-2022

Contents

Practice Document

For this demonstration, we will be using the Heart Disease datasets which are openly available from Public Heath Scotland. In particular, we will be focusing on the on the mortality rates dataset.

```
# Note: the above works for code generally and comments, but does not work for
# strings such as the URL. Additionally, don't forget to install the formatR
# package if you plan to use this!
library(tidyverse)
library(janitor)
library(lubridate)
library(kableExtra)
library(formatR)

# very very very very very very very very very very very very very very very very very
# very very very very very very very very very very very very very very very very very
# very very very very very very very very very very very very very very very very very
# very very very very very very very very long comment
```

But first an aside...

When knitting to PDF, you can wrap the code and comments using the `formatR` package and the arguments `tidy.opts=list(width.cutoff=80), tidy = TRUE` (see the output for the very... very long comment when

knitted). However, this does not wrap strings, such as URLs due to LaTeX specific issues. There are very complicated ways around this, which we will not be covering as it requires other coding knowledge, but Kevin and Brittany have come up with 2 possible (though convoluted) solutions. For the Programming Assignment, the easiest solution may be to load the data from a saved file instead.

Brittany's solution

When you are writing code chunks and R leaves a blank space in the line number when the code is wrapped (e.g. line 64, blank space, line 65). When knitting to HTML the knitted document reflects this, unfortunately not the case when knitting to PDF. To unwrap the code, click enter at the beginning of the link without the number. This **however** means that the URL is not longer able to run without error. So, in a convoluted work around, you could have a chunk set to `eval=FALSE` meaning the code is not run but the knitted document shows the code. Then you could include a chunk below which will actually load the data (i.e., the code is run) but not show this in the knitted document (`echo=FALSE`)... convoluted as I said.

```
## Read in the 3 datasets
# Heart Disease Activity By Health Board
activity_raw <- read_csv("https://www.opendata.nhs.scot/dataset/0e17f3fc-9429-48aa-b1ba
                      -2b7e55688253/resource/748e2065-b447-4b75-99bd-f17f26f3eaef/downl
                      oad/hd_activitybyhbr.csv")

# Heart Disease Mortality By Health Board
mortality_raw <- read_csv("https://www.opendata.nhs.scot/dataset/0e17f3fc-9429-48aa
                      -b1ba-2b7e55688253/resource/dc0512a8-eb49-43b9-84f1-17ef95365d5
                      7/download/hd_mortalitybyhbr.csv")

# Health Board look up
hb <- read_csv("https://www.opendata.nhs.scot/dataset/9f942fdb-e59e-44f5-b534-
               d6e1729cc7b/resource/652ff726-e676-4a20-abda-435b98dd7bdc/download/hb1
               _hb21.csv")
```

Kevin's solution

Kevin's solution is slightly different: for each data set, add a variable such as `link1, echo=FALSE` and then use it in the code to be printed. If you adopt this method, for reproducibility sake, in the text you could include the full URL to be printed out. For example:

> The heart disease activity dataset was from the Public Health Scotland website: https://www.opendata.nhs.scot/dataset/0e17f3fc-9429-48aa-b1ba-2b7e55688253/resource/748e2065-b447-4b75-99bd-f17f26f3eaef/download/hd_activitybyhbr.csv

or

> The hearth disease activity dataset downloads from link1

> Data from https://www.opendata.nhs.scot/dataset/0e17f3fc-9429-48aa-b1ba-2b7e55688253/resource/748e2065-b447-4b75-99bd-f17f26f3eaef/download/hd_activitybyhbr.csv

```
activity_raw2 <- read_csv(link1)
```

Data Wrangling

```
activity_raw %>%
  glimpse()
```

```
## Rows: 43,200
## Columns: 15
## $ FinancialYear        <chr> "2011/12", "2011/12", "2011/12", "2011/12", "2011~
## $ HBR                  <chr> "S08000015", "S08000015", "S08000015", "S08000015~
## $ HBRQF                <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ AdmissionType        <chr> "All", "All", "All", "All", "All", "All", "All", ~
## $ AdmissionTypeQF      <chr> "d", "d", "d", "d", "d", "d", "d", "d", "d", "d",~
## $ AgeGroup             <chr> "0-44 years", "0-44 years", "45-64 years", "45-64~
## $ AgeGroupQF           <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Sex                  <chr> "Males", "Females", "Males", "Females", "Males", ~
## $ SexQF                <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Diagnosis            <chr> "Coronary Heart Disease", "Coronary Heart Disease~
## $ NumberOfDischarges   <dbl> 104, 55, 1087, 385, 760, 411, 663, 638, 35, 10, 3~
## $ NumberOfDischargesQF <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ CrudeRate            <dbl> 109.22649, 56.11901, 2078.07601, 683.64230, 4057.~
## $ CrudeRateQF          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ EASR                 <dbl> 109.75422, 53.33447, 2064.89207, 680.54949, 4071.~
```

```
activity <- activity_raw %>%
  left_join(hb, by = c("HBR" = "HB")) %>%
  select(FinancialYear,
         HBName,
         AdmissionType,
         AgeGroup,
         Sex,
         Diagnosis,
         NumberOfDischarges) %>%
  clean_names() %>%
  separate(financial_year, into = c("Year", NA), sep = "/", convert = TRUE) %>%
  mutate(sex = str_replace(sex, "Females", "Female"),
         sex = str_replace(sex, "Males", "Male")) %>%
  filter(sex            != "All",
         age_group      != "All",
         admission_type != "All",
         hb_name        != "S92000003")

mortality <- mortality_raw %>%
  left_join(hb, by = c("HBR" = "HB")) %>%
  select(Year,
         HBName,
         AgeGroup,
         Sex,
         Diagnosis,
         NumberOfDeaths) %>%
  clean_names() %>%
  mutate(sex = str_replace(sex, "Females", "Female"),
         sex = str_replace(sex, "Males", "Male")) %>%
```

```
  filter(sex       != "All",
         age_group != "All",
         hb_name   != "S92000003")
```

Is the activity dataset in long or wide format?

```
activity %>%
  head(n = 10)
```

```
## # A tibble: 10 x 7
##     Year hb_name                admission_type age_group   sex   diagn~1 numbe~2
##    <int> <chr>                  <chr>          <chr>       <chr> <chr>     <dbl>
##  1  2011 NHS Ayrshire and Arran Elective       0-44 years  Male  Corona~      18
##  2  2011 NHS Ayrshire and Arran Elective       0-44 years  Fema~ Corona~      10
##  3  2011 NHS Ayrshire and Arran Elective       45-64 years Male  Corona~     292
##  4  2011 NHS Ayrshire and Arran Elective       45-64 years Fema~ Corona~     102
##  5  2011 NHS Ayrshire and Arran Elective       65-74 years Male  Corona~     205
##  6  2011 NHS Ayrshire and Arran Elective       65-74 years Fema~ Corona~      94
##  7  2011 NHS Ayrshire and Arran Elective       75plus yea~ Male  Corona~      70
##  8  2011 NHS Ayrshire and Arran Elective       75plus yea~ Fema~ Corona~      55
##  9  2011 NHS Borders            Elective       0-44 years  Male  Corona~       4
## 10  2011 NHS Borders            Elective       0-44 years  Fema~ Corona~       1
## # ... with abbreviated variable names 1: diagnosis, 2: number_of_discharges
```

Data Visualization

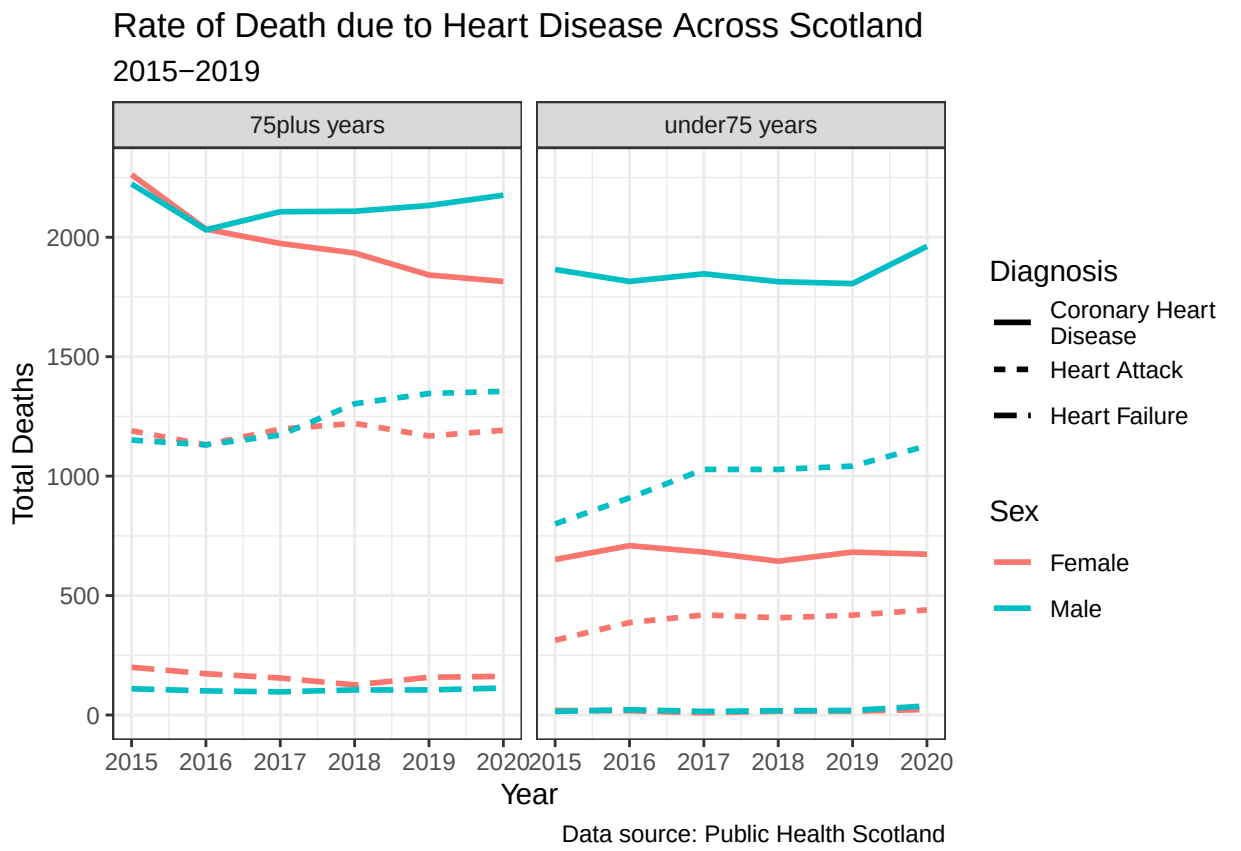
How might we visualize the following: Since 2010, what influence does gender have on the rate of death for different diagnoses for those aged 75+ vs those under 75?

There are at least 2 different coding approaches you can take to this:

1. Create a separate data frame for plotting, which you can then reuse for other plots or tables.

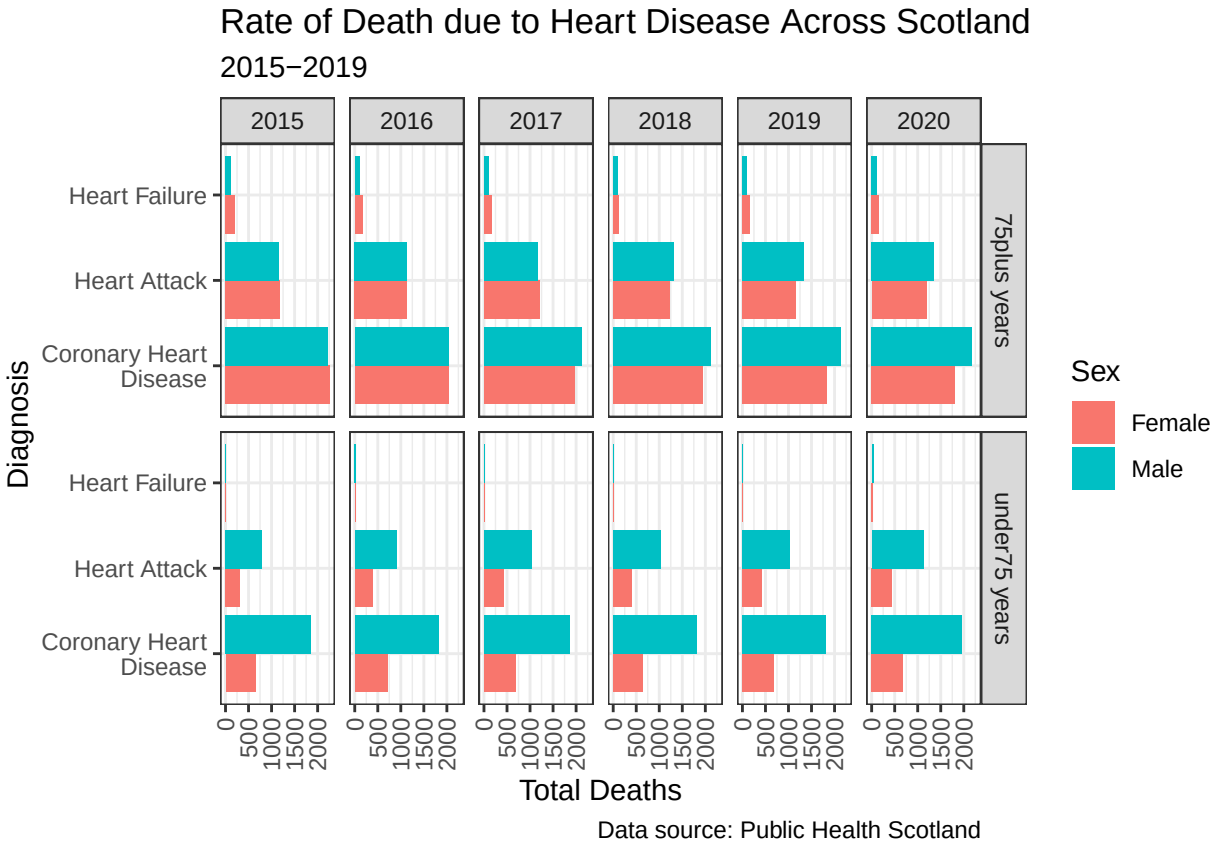
```
mortality_plot <- mortality %>%
  filter(year >= 2015,
         age_group %in% c("under75 years", "75plus years")) %>%
  group_by(diagnosis, sex, age_group, year) %>%
  summarise(total_deaths = sum(number_of_deaths, na.rm=TRUE)) %>%
  #wrap string "Coronary Heart Disease" for better plotting
  mutate(diagnosis = str_wrap(diagnosis, width = 15))

mortality_plot %>%
  ggplot(aes(x = year, y = total_deaths, color = sex)) +
  geom_line(aes(linetype = diagnosis), lwd = 1) +
  facet_wrap(~age_group) +
  labs(title = "Rate of Death due to Heart Disease Across Scotland",
       subtitle = "2015-2019",
       caption = "Data source: Public Health Scotland",
       y = "Total Deaths",
       x = "Year",
       color = "Sex",
       linetype = "Diagnosis") +
  theme_bw()
```



2. Pipe the data into the `ggplot` but do some data wrangling first.

```
mortality %>%
  filter(year >= 2015,
         age_group %in% c("under75 years", "75plus years")) %>%
  group_by(diagnosis, sex, age_group, year) %>%
  summarise(total_deaths = sum(number_of_deaths, na.rm=TRUE)) %>%
  #wrap string "Coronary Heart Disease" for better plotting
  mutate(diagnosis = str_wrap(diagnosis, width = 15)) %>%
  ggplot(aes(x = diagnosis, y = total_deaths, fill = sex)) +
  geom_col(position = "dodge") +
  facet_grid(age_group~year) +
  #compare facet_grid to facet_wrap
  #facet_wrap(~age_group~year) +
  labs(title = "Rate of Death due to Heart Disease Across Scotland",
       subtitle = "2015-2019",
       caption = "Data source: Public Health Scotland",
       y = "Total Deaths",
       x = "Diagnosis",
       fill = "Sex") +
  coord_flip() +
  theme_bw() +
  #adjust the text on the x axis to be more legible
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```


Rate of Death due to Heart Disease Across Scotland
2015−2019
2015
2016
2017
2018
2019
2020
75plus years
under75 years
Heart Failure
Heart Attack
Coronary Heart Disease
Diagnosis
0
500
1000
1500
2000
Total Deaths
Sex
Female
Male
Data source: Public Health Scotland

What about a table?

For analysis and data visualisation, tidy data (i.e., long data) is the ideal. However, for tables often wide data is more readable.

```
mortality_plot %>%
  pivot_wider(names_from = diagnosis, values_from = total_deaths) %>%
  ungroup()%>%
  kbl()
```

sex	age_group	year	Coronary Heart Disease	Heart Attack	Heart Failure
Female	75plus years	2015	2262	1190	200
Female	75plus years	2016	2034	1131	173
Female	75plus years	2017	1974	1197	155
Female	75plus years	2018	1934	1221	126
Female	75plus years	2019	1842	1168	158
Female	75plus years	2020	1815	1192	162
Female	under75 years	2015	651	313	19
Female	under75 years	2016	709	387	18
Female	under75 years	2017	682	419	9
Female	under75 years	2018	644	407	16
Female	under75 years	2019	682	418	16
Female	under75 years	2020	673	440	23
Male	75plus years	2015	2223	1151	110
Male	75plus years	2016	2031	1131	101
Male	75plus years	2017	2107	1172	97
Male	75plus years	2018	2109	1303	105
Male	75plus years	2019	2133	1346	105
Male	75plus years	2020	2176	1355	113
Male	under75 years	2015	1865	800	15
Male	under75 years	2016	1815	909	22
Male	under75 years	2017	1847	1028	15
Male	under75 years	2018	1814	1028	18
Male	under75 years	2019	1806	1042	19
Male	under75 years	2020	1962	1127	38

Heart Disease Mortality in Scotland 2015-2019					
Sex	Age Group	Year	Coronary Heart Disease	Heart Attack	Heart Failure
Female	75plus years	2015	2262	1190	200
Female	75plus years	2016	2034	1131	173
Female	75plus years	2017	1974	1197	155
Female	75plus years	2018	1934	1221	126
Female	75plus years	2019	1842	1168	158
Female	75plus years	2020	1815	1192	162
Female	under75 years	2015	651	313	19
Female	under75 years	2016	709	387	18
Female	under75 years	2017	682	419	9
Female	under75 years	2018	644	407	16
Female	under75 years	2019	682	418	16
Female	under75 years	2020	673	440	23
Male	75plus years	2015	2223	1151	110
Male	75plus years	2016	2031	1131	101
Male	75plus years	2017	2107	1172	97
Male	75plus years	2018	2109	1303	105
Male	75plus years	2019	2133	1346	105
Male	75plus years	2020	2176	1355	113
Male	under75 years	2015	1865	800	15
Male	under75 years	2016	1815	909	22
Male	under75 years	2017	1847	1028	15
Male	under75 years	2018	1814	1028	18
Male	under75 years	2019	1806	1042	19
Male	under75 years	2020	1962	1127	38

Note:

Source: Public Health Scotland

Aesthitics changes to the table

```
#remember the mortality plot data is grouped, so we may want to ungroup the data first before creating
mortality_plot %>%
  pivot_wider(names_from = diagnosis, values_from = total_deaths) %>%
  ungroup() %>%
  kbl(
    col.names = c(
      sex = "Sex",
      age_group = "Age Group",
      year = "Year",
      `Coronary Heart\nDisease` = "Coronary Heart Disease",
      `Heart Attack` = "Heart Attack",
      `Heart Failure` = "Heart Failure"
    )
  ) %>%
  kable_styling() %>%
  add_header_above(header = c("Heart Disease Mortality in Scotland 2015-2019" = 6)) %>%
  footnote("Source: Public Health Scotland")
```

Heart Disease Mortality in Scotland 2015-2019					
Year	Sex	Age Group	Coronary Heart Disease	Heart Attack	Heart Failure
2015	Female	75plus years	2262	1190	200
2016	Female	75plus years	2034	1131	173
2017	Female	75plus years	1974	1197	155
2018	Female	75plus years	1934	1221	126
2019	Female	75plus years	1842	1168	158
2020	Female	75plus years	1815	1192	162
2015	Female	under75 years	651	313	19
2016	Female	under75 years	709	387	18
2017	Female	under75 years	682	419	9
2018	Female	under75 years	644	407	16
2019	Female	under75 years	682	418	16
2020	Female	under75 years	673	440	23
2015	Male	75plus years	2223	1151	110
2016	Male	75plus years	2031	1131	101
2017	Male	75plus years	2107	1172	97
2018	Male	75plus years	2109	1303	105
2019	Male	75plus years	2133	1346	105
2020	Male	75plus years	2176	1355	113
2015	Male	under75 years	1865	800	15
2016	Male	under75 years	1815	909	22
2017	Male	under75 years	1847	1028	15
2018	Male	under75 years	1814	1028	18
2019	Male	under75 years	1806	1042	19
2020	Male	under75 years	1962	1127	38

Note:

Source: Public Health Scotland

Hint: If you want to rearrange the columns order, you can do with `select()`

```
mortality_plot %>%
  pivot_wider(names_from = diagnosis, values_from = total_deaths) %>%
  select(year, everything()) %>%
  ungroup() %>%
  kbl(
    col.names = c(
      "year" = "Year",
      "sex" = "Sex",
      "age_group" = "Age Group",
      `Coronary Heart\nDisease` = "Coronary Heart Disease",
      `Heart Attack` = "Heart Attack",
      `Heart Failure` = "Heart Failure"
    )
  ) %>%
  kable_styling() %>%
  add_header_above(header = c("Heart Disease Mortality in Scotland 2015-2019" = 6)) %>%
  footnote("Source: Public Health Scotland")
```