Introduction to R

Day 2

Transforming Publishing
phs.transformingpublishing@phs.scot

Public Health Scotland

Pathway

Intro
Intro to Data and Tools, Overview of R

Foundations
Commenting, Types, Variables, Statements, Data Structures, Packages

RStudio
Desktop/Server Version, Interface, Customisation, R Scripts, Hints/Tips

Workflow
Overview (Collect, Explore, Wrangle, Viz, Outputs), Git(Hub/ea), RMarkdown, RAP, Templates, Style Guide

Data Flow
Directories/File Paths, CSVs, SPSS (haven), SMRA/Other Databases

Explore
Mean, Median, Summary Function, Frequencies/Cross-Tabs

Wrangle
Tidyverse (dplyr/magrittr), Pipes, Functions (Filter, Mutate, Arrange, etc.), PHS Methods

Visualise
Intro to ggplot2, Line Graphs, Bar Plots, Scatterplots, Customisation

Output
Overview of RMarkdown, Shiny, etc.

Review
Overview, Next Steps, Q&A

Wrangle

Tidyverse

is a suite of packages for data exploration, manipulation, and visualisation; it's best practice to utilise these where possible.

- functions have a consistent format, i.e. `function(data, task)`

- gives us the package `dplyr`

dplyr

is a grammar of data manipulation, providing a set of "*verbs*" to help solve most data manipulation challenges

```
library(dplyr)
```

- `filter()`
- `mutate()`
- `arrange()`
- `select()`
- `group_by()`
- `summarise()`
- `count()`
- `rename()`
- `recode()`

Pipe Operator

- `%>%` is used to link functions together, passing the previous to the next
- Using the pipe operator makes R code more readable and prevents extensive parenthesis building up with multiple function calls
- Readable as "and then"
- Shortcut: (`ctrl` + `shift` + `M`)

```
arrange(filter(borders,
     HospitalCode == "B102H"), Dateofbirth)
```

```
borders %>%
     filter(HospitalCode == "B102H") %>%
     arrange(Dateofbirth)
```

Filter

`filter(<data>, <logical expression>)`

- picks cases based on their values

```
# all cases with E12 specialty
borders %>%
     filter(Specialty == "E12")

# B120H cases more than 10 days
borders %>%
     filter(HospitalCode ==
                "B120H" &
              LengthOfStay > 10)
```

Mutate

```
mutate(<data>, <new_col> =
<expression>)
```

- adds new variables that are functions of existing variables

```
# length of stay divided by 2
borders %>%
     mutate(los_div2 =
          LengthOfStay / 2)
```

Arrange

`arrange(<data>, <variables>)`

- changes the ordering of rows

- `desc()` to sort in descending order

```
# sort by Hospital Code
borders %>%
     arrange(HospitalCode)
```

Select

`select(<data>, <expression>)`

- picks variables based on their names
- prepend "–" to a variable to remove

```
# remove Postcode
borders %>%
     select(-Postcode)
```

Exercise 2

1. Read in "`Borders.csv`" (giving the data frame an appropriate name)
2. Which patients had a `LengthOfStay` of between 2 and 10 days?
3. Which of these patients were under `Specialty` E12 or C8?
4. Remove all columns other than `URI`, `Specialty`, and `LengthOfStay`
5. Complete all the above using pipes and write this to a CSV ordered by `LengthOfStay`

Group By

```
group_by(<data>,
<col_name>)
```

- groups variables to perform operations

- This doesn't visibly affect the data, but we can see the output shows the grouping. We can then perform other operations on the groups.

```
# sort by Hospital Code
borders %>%
     group_by(HospitalCode)
```

```
> ...
# Groups: HospitalCode [48]
...
```

Summarise

```
summarise(<data>, <name> = <expression>)
```

- reduced multiple values down to a single summary

```
# avg length of stay by hospital
borders %>%
     group_by(HospitalCode) %>%
     summarise(mean_los =
          mean(LengthOfStay))
```

Count

`count(<data>, <variables>)`

- useful for running frequencies, this calls group_by() and produces counts for a specified column

- sort by descending order using `sort = TRUE` as an argument

```
# counts of specialty
borders %>%
     count(Specialty, sort = TRUE)
```

Exercise 3

1. Read in "`Borders.csv`" (giving the data frame an appropriate name)
2. What is the earliest admission date by specialty?
3. What is the latest discharge date by specialty?
4. What are the number of admissions per hospital, per specialty?

Rename

```
rename(<data>, <new_name> = <existing_name>)
```

- renaming specific columns in a data frame

```
# rename Date of Birth column
borders %>%
    rename(date_of_birth = Dateofbirth)
```

Recode

```
mutate(<col> = recode(<col>,
<existing_code> =
<new_code>))
```

- for changing values within a column
- works best when used with `mutate()`

```
# change hospital code
borders %>%
     mutate(HospitalCode =
          recode(HospitalCode,
               "B120V" = "B120H"))
```

Exercise 4

1. Select the `URI`, `Specialty`, and `Dateofbirth` columns from the borders data and save to a new data frame.
2. Arrange this new data in ascending order by `Specialty` and check the results.
3. Extract the records with a missing `Dateofbirth` (hint: `?filter`)
4. Finally, recode `Specialty` "A1" to be "General Medicine"

Joining Tables

```
<type>_join(<data1>,
<data2>, by =
<common_variable>)
```

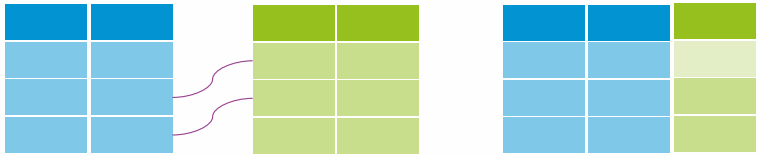
- for merging data by matching together using common variable(s)

```
# merge baby data
baby5 <- read_csv("data/Baby5.csv")
baby6 <- read_csv("data/Baby6.csv")
baby_joined <-
     left_join(baby5, baby6, by =
          c("FAMILYID", "DOB"))
```

Join Types

`left_join()`

`all rows from the 'left', any matches from the 'right'`

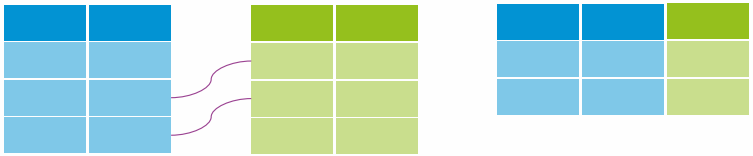


`right_join()`

`all rows from the 'right', any matches from the 'left'`

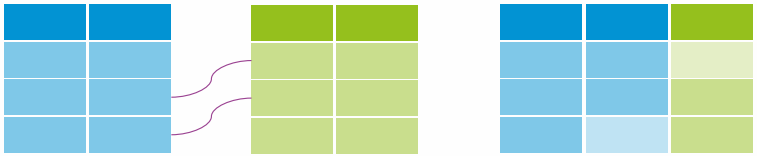
`inner_join()`

`rows of matched fields from data sets`



`full_join()`

`all rows, na for non-matched fields`



General Skills

Debugging

1. **Review warnings/errors** – these can appear cryptic but use Google and some will become familiar. Checking the functions could help – `?<function>`
2. **Narrow the problem** – step through the code, isolating the issue.
3. **Google/Stack Overflow** – this can be specific to the bug or more general to the problem you're trying to solve.
4. **Pair up** – sometimes a fresh pair of eyes makes the difference. Post a message on the R User Group Technical Queries Teams channel

Continuous Learning

- Data Science Knowledge Base – (People Development Hub) is the place for all content related to Data Science learning:
  - **Review, Follow, and Contribute to Guidance** – guidance is for sharing best practice, maintaining security, and improving efficiency.
  - **Expand your skills** – take another course to build your R skills or on related technologies (e.g. Git).
  - **Keep up to date** – our infrastructure is improving, we support knowledge sharing events, and so much more!

Review

Project

scotland.shinyapps.io/phs-rtraining-intro/

- Day 2 Project – Handwashing

Feel free to follow along for the project on the app or build a script on RStudio.

Next Steps

- Homework project & day 3
- Embed your new knowledge and skills!
- Expand your knowledge and skills with related technologies (e.g. git)
- Take R Further - look at other training opportunities (`phsmethods`)

Getting Help

- Vignettes (Help) / `` `?<function>` ``
- Google / Stack Overflow
  tag queries "[r] & [tidyverse]"
- R User Group Teams – Technical Queries
- Transforming Publishing