Trivia: Sailing the seven seas

Feature	Ocean
1. This ocean sits in the oldest existing ocean basin	A – Southern
2. This ocean has the highest salinity	B – Pacific
3. This ocean is the shallowest	C – Atlantic
4. This ocean has the longest continuous ocean current	D – Arctic

Share your answers in the chat box!

We will start at 9:00 Pacific / 10:00 Mountain / 11:00 Central / 12:00 Eastern / 13:00 Atlantic / 13:30 NFLD

Please take a moment to ensure that you have downloaded course materials for today, refresh your beverage, and / or network with us.

TDU

Introduction to R

Advanced Data Processing

Day 3

Public Health Agency of Canada
Agence de la santé publique du Canada

Canada

Course Learning Objectives

- Learners will be able to (in R):

- Carry out data cleaning and processing, and descriptive epidemiological analyses (including commonly used data visualizations);

- Create automated data products (e.g., epidemiologic summaries);

- Design and carry out a data collation plan that is consistent with proposed analysis plan;

- Explain when it is most appropriate to program analyses and automates tasks using R;

- Find and appraise possible solutions to R programming challenges

What we heard

Designed by Wayhomestudio - Freepik.com

Exercise Debrief

Exercise 2:

Tuberculosis outbreak

- Setup your workspace
- Clean and process data
- Create descriptive epi figures and summary tables relevant to the outbreak investigation
- Create a social network diagram
- Build an automated report using R markdown

1. What was your favorite function or component of this activity?
2. Did anything surprise you? Did you do anything differently?
3. Do you foresee using any parts of this activity in your workplace? How?

The Map

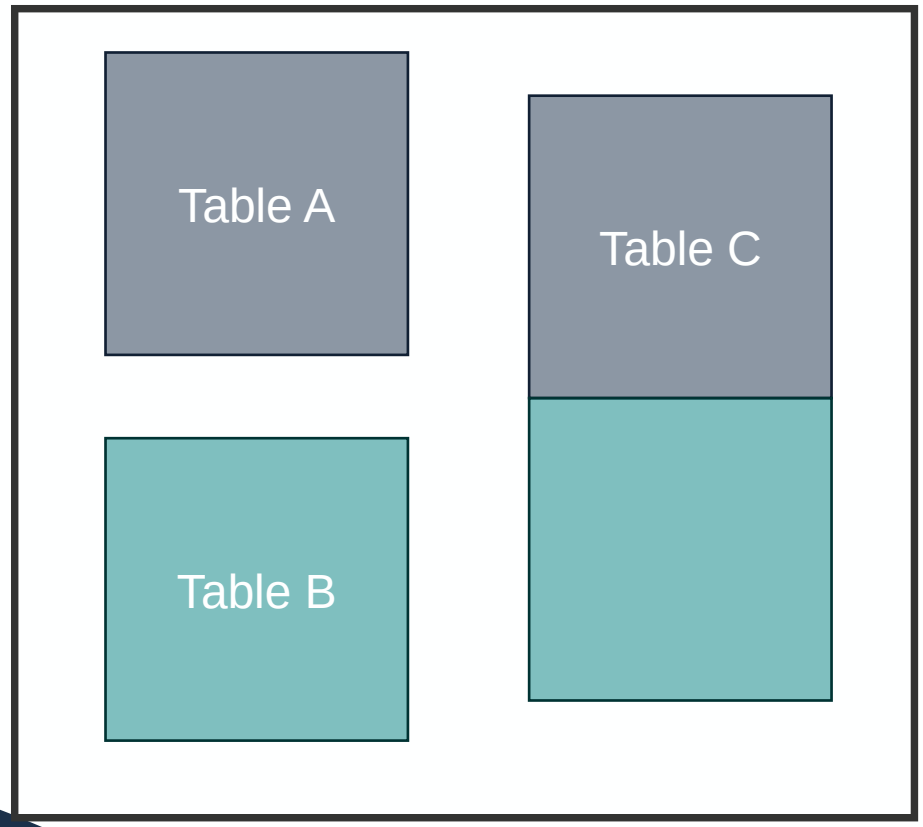
The treasure is in sight! We only have to navigate past

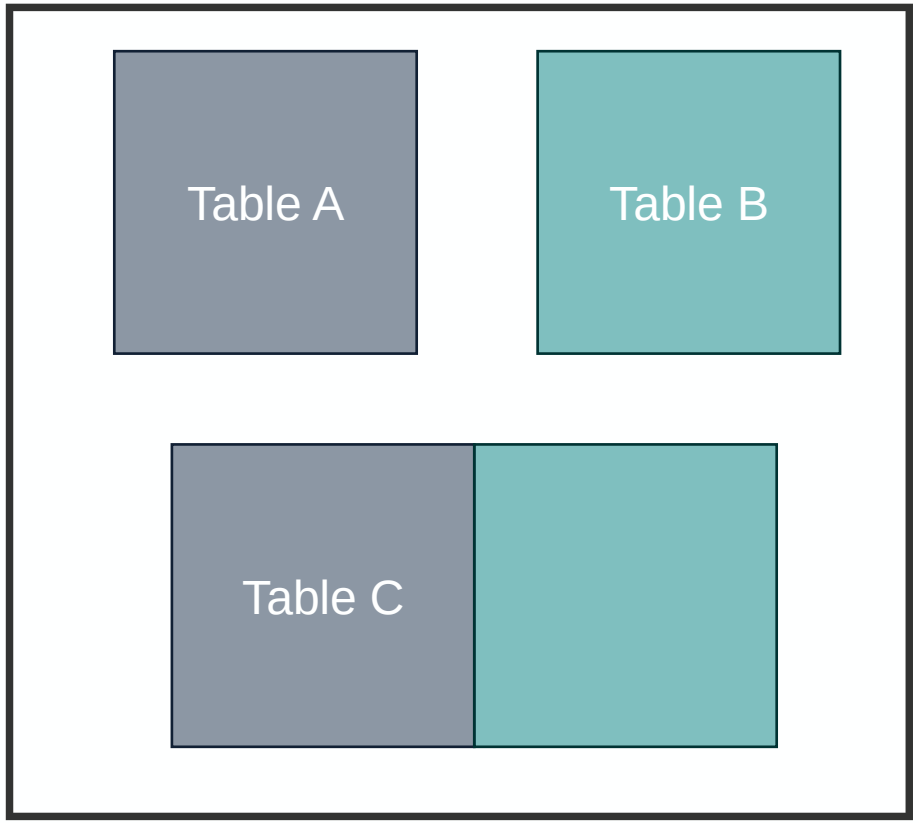
- “The forest of advanced data-processing palms”:
  - Appending and merging
- Test your analysis planning and troubleshooting skills in open-water!

Appending data

Review: Appending data

- *Appending* allows us to attach one table to another table:





Review: Considerations for appending data

- Important considerations for appending data:
  1. *The data classes are you appending*
  2. *The column headers*
  3. *The table sizes*
  4. *Ensuring that you have unique key variable(s) between data sets*
  5. *Checking to make sure the operation worked as expected*

- Which of the above considerations does **not** belong in the above list?

Appending data

- Appending data refers simply to adding *some data* to *some other data* to create longer/wider (more) data. (What could be better!?)

Appending data with the combine function: c()

- At its simplest, appending data in R can be done using the *combine function:* c()
- For example:

```
> "cat"
[1] "cat"
> "dog"
[1] "dog"
> c("cat", "dog")
[1] "cat" "dog"
```

Appending data objects

- You can also use c() to append **objects** together!
- For example:

```
> cats <- c("siamese", "tabby")
> dogs <- c("bulldog", "terrier")
> pets <- c(cats, dogs)
> pets
[1] "siamese" "tabby"   "bulldog" "terrier"
```

- What happens if you use c() to append different data types?
  - E.g., characters and numbers?

Considering changes to data types after c()

- What happens if you use c() to append different data types?
  - E.g., characters and numbers?

```
> dogs <- c("bulldog", "terrier")
> numbers <- c(1, 2, 3, 4)
> assorted <- c(dogs, numbers)
> assorted
[1] "bulldog" "terrier" "1"       "2"       "3"       "4"
```

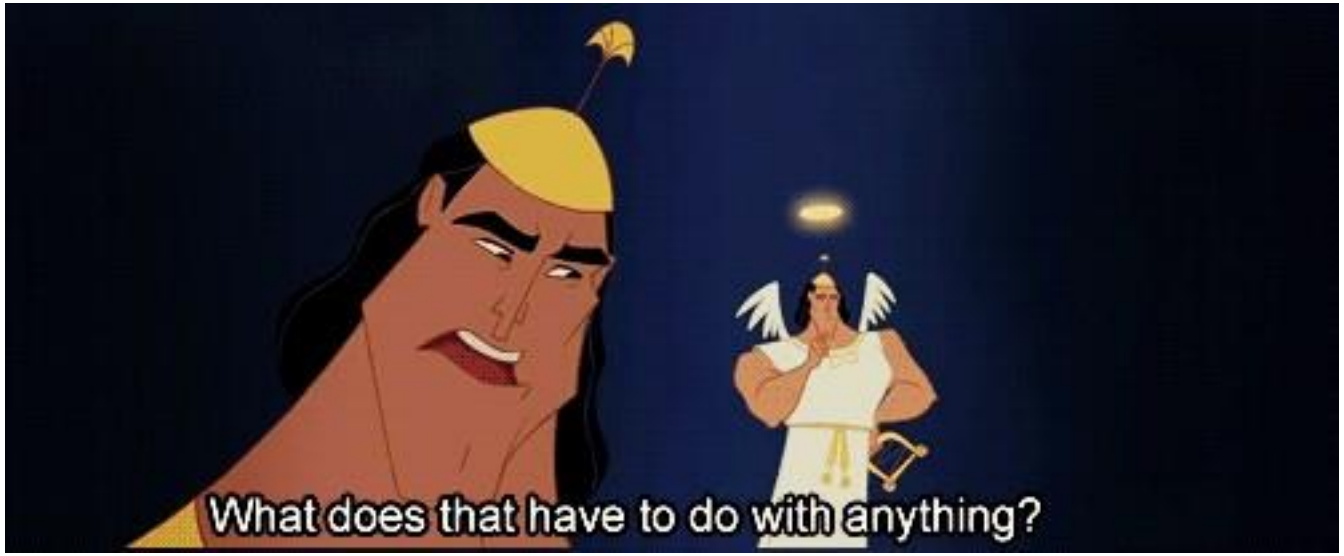
- This works! But what do you notice about the numbers?
  - Are they still numeric?
  - How could we test this?

Testing data types after c()

- Appending objects is useful, but it is important to be aware of what types of data you're left with

```
> is.numeric(assorted)
[1] FALSE
> is.numeric(assorted[3])
[1] FALSE
> is.numeric(assorted[4])
[1] FALSE
> is.numeric(assorted[5])
[1] FALSE
> is.numeric(assorted[6])
[1] FALSE
> class(assorted)
[1] "character"
```

Why does this matter?



Post in the chat an example of where you think changing data classes could lead to issues!

Appending tables using bind_rows()

- To append one table to the bottom of another in R, we can use the following function:
  - bind_rows()
  - E.g.,

```
> table1
  x1 x2 x3
1  1  4  7
2  2  5  8
3  3  6  9
> table2
  x1 x2 x3
1  3  6  9
2  2  5  8
3  1  4  7
```

```
> bind_rows(table1, table2)
  x1 x2 x3
1  1  4  7
2  2  5  8
3  3  6  9
4  3  6  9
5  2  5  8
6  1  4  7
```

- Column names (e.g., x1, x2, x3) are ***conserved*** in this operation
- What happens if we combine tables with different column names?

bind_rows() with different column headers (1)

- What happens if we try to bind_rows() two tables with different column names?

```
> table1
  x1 x2 x3
1  1  4  7
2  2  5  8
3  3  6  9
```

```
> table3
  x1 x3 x4
1  1  4  8
2  2  2  9
3  3  5  1
```

bind_rows() with different column headers (2)

- What happens if we try to bind_rows() two tables with different column names?

```
> table1
  x1 x2 x3
1  1  4  7
2  2  5  8
3  3  6  9
```

```
> table3
  x1 x3 x4
1  1  4  8
2  2  2  9
3  3  5  1
```

```
> bind_rows(table1, table3)
  x1 x2 x3 x4
1  1  4  7 NA
2  2  5  8 NA
3  3  6  9 NA
4  1 NA  4  8
5  2 NA  2  9
6  3 NA  5  1
```

Appending tables with bind_cols()

- The same way we can bind_rows() to make tables *longer*, we can also bind_cols() to make them *wider*

```
> table1
  x1 x2 x3
1  1  4  7
2  2  5  8
3  3  6  9
```

```
> table4
  x4 x5 x6
1  3  2  1
2  1  5  4
3  4  8  9
```

```
> bind_cols(table1, table4)
  x1 x2 x3 x4 x5 x6
1  1  4  7  3  2  1
2  2  5  8  1  5  4
3  3  6  9  4  8  9
```

- Just as column headers were preserved in bind_rows(), row names are preserved in bind_cols()

Question: bind_cols() with same column headers? (1)

- What do you think happens if you try to bind_cols() two tables that have the *same* column headers?

Question: bind_cols() with same column headers? (2)

- What do you think happens if you try to bind_cols() two tables that have the *same* column headers?

```
> bind_cols(table1, table5)
New names:
* x1 -> x1...1
* x2 -> x2...2
* x3 -> x3...3
* x1 -> x1...4
* x2 -> x2...5
* ...
  x1...1 x2...2 x3...3 x1...4 x2...5    x3...6
1      1      4      7    dog   blue chocolate
2      2      5      8    cat yellow   vanilla
3      3      6      9   fish    red   rainbow
```

Different data types with bind_rows() and bind_cols() (1)

- What happens when you try to append tables of different data types (*i.e.*, data *classes*) using bind_rows or bind_cols?

```
> table1
  x1 x2 x3
1  1  4  7
2  2  5  8
3  3  6  9
```

```
> table5
    x1     x2        x3
1  dog   blue chocolate
2  cat yellow   vanilla
3 fish    red   rainbow
```

Different data types with bind_rows() and bind_cols() (2)

- What happens when you try to append tables of different data types (*i.e.*, data *classes*) using bind_rows or bind_cols?

```
> bind_rows(table1, table5)
Error: Can't combine `..1$x1` <double> and `..2$x1` <character>.
```

```
> bind_cols(table1, table5)
New names:
* x1 -> x1...1
* x2 -> x2...2
* x3 -> x3...3
* x1 -> x1...4
* x2 -> x2...5
* ...
  x1...1 x2...2 x3...3 x1...4 x2...5    x3...6
1      1      4      7    dog   blue chocolate
2      2      5      8    cat yellow   vanilla
3      3      6      9   fish    red   rainbow
```

Data joins

Appending tables

- Often, you'll want to append one table with another to create a large, single table

x1	x2	x3	x4
1	5	1	4
8	3	6	4

+

x1	x2	x3	x4
5	8	7	9
1	4	8	2

x1	x2	x3	x4
1	5	1	4
8	3	6	4
5	8	7	9
1	4	8	2

Risks with appending data

Thinking to the work you've done in the past, or may be likely to do in the future, what are some of the challenges with using simple functions like **bind_cols()** and **bind_rows()** to combine datasets?

Combining tables for the same observations

- What if you want to *combine* two tables that contain different data for the same individuals?
  - Could you **append** these datasets? Would you try **bind_col()** or **bind_row()**?

Name	# Desserts eaten per week
MAB	4
Ben	8
Joanne	2

Name	# Cavities 2010-2019
Ben	14
Joanne	1
MAB	8

The order of data matters…

- Yes, you could.. But *should you?*
- When appending tables, *order matters.* Therefore, if your tables aren't sorted exactly the same, or contain different numbers of rows, they may combine *incorrectly.*

Name	# Desserts eaten per week	# Cavities 2010-2019
MAB	4	14
Ben	8	1
Joanne	2	8

The order of data matters… or does it?

- Using a merge (or join) operation ensures that your variables match up with the correct *key* variable
  - In this example, "*Name*" is the key we want to match on

Name	# Desserts eaten per week
MAB	4
Ben	8
Joanne	2

Name	# Cavities 2010-2019
Ben	14
Joanne	1
MAB	8

Joining on key values

- Using table joins makes sure that the information you're adding stays with the correct key identifier

Name	# Desserts eaten per week
MAB	4
Ben	8
Joanne	2

Name	# Cavities 2010-2019
Ben	14
Joanne	1
MAB	8

Tidy joining in R

- Using a join (or merge) operation ensures that your variables match up with the correct *key* variable
  - In this example, "*Name*" is the key we want to match on

```
> print(desserts1)
# A tibble: 3 × 2
  Name   Desserts.wk
  <chr>        <dbl>
1 MAB              4
2 Joanne           7
3 Ben             10
```

```
> print(cavities1)
# A tibble: 3 × 2
  Name   Cavities.10yr
  <chr>          <dbl>
1 MAB                8
2 Ben               14
3 Joanne             6
```

Tidy merging in r

- Using R (tidyverse) we can call the function full_join() to combine the two tables, matching on a key variable (e.g., “Name”)

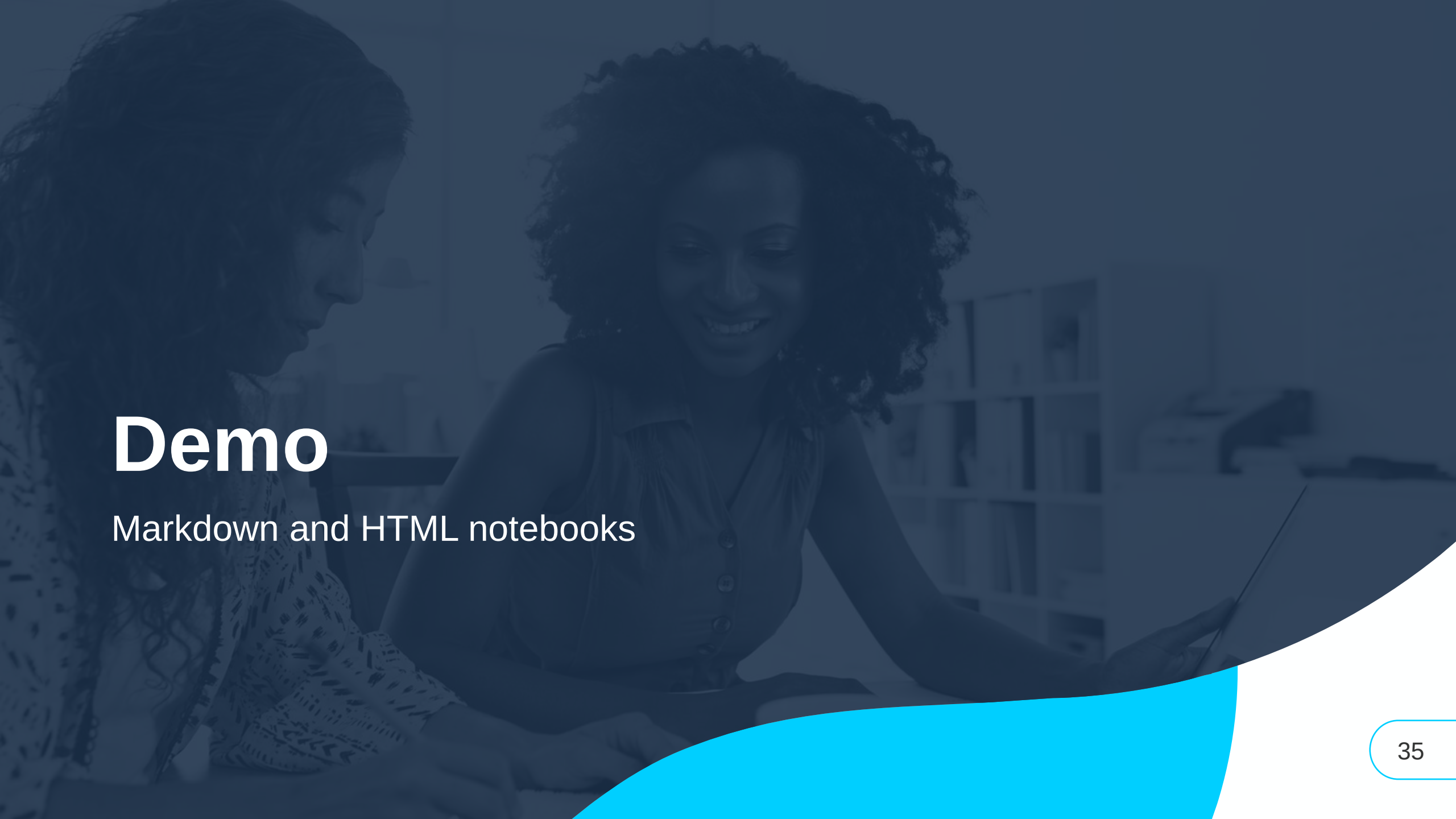
```
> full_join(desserts1, cavities1, by="Name")
# A tibble: 3 × 3
  Name    Desserts.wk Cavities.10yr
  <chr>         <dbl>         <dbl>
1 MAB               4             8
2 Joanne            7             6
3 Ben              10            14
```

Other examples of table joins

- Excel?
  - vlookup; hlookup
- Stata?
  - merge(1:m); merge(m:m)
- SAS?
  - Proc SQL
    - Select – From – Where
  - MERGE
    - Data statement option
- Others?

Types of tidy joins in R

	Join Type	Definition
Mutating joins	**Full**	Returns all data from the left and right side of the statement regardless of whether or not there are matches in the other table.
	Left	A link between two tables where all data from the left side of the statement is returned with only data that matches from the right.
	Right	A link between two tables where all data from the right side of the statement is returned with only data that matches from the left.
	Inner	A link between two tables based on equality between values (e.g., key variables) in a column of tables on the left and right side of the statement.
Filtering joins	**Anti**	Returns rows from the left side of the statement for which there is no match on the right (e.g., **not** in).
	Semi	Returns all rows from the left side of the statement where there are matching values in the right side, keeping only columns from the left



Demo

Markdown and HTML notebooks

Stranded on a des(s)ert Island:

Name/draw a dessert you'd bring with you

Please return by:

[TIME] Pacific / [TIME] Mountain / [TIME] Central / [TIME] Eastern / [TIME] Atlantic / [TIME] NFLD

Navigating tricky waters

Excellent R memes (1)

Excellent R memes (2)

Exercise: Navigating tricky waters (1)

Inspired by course participants in previous run-throughs

- What to do when you're met with an analysis task that you don't know how to complete?

Take 10 minutes to draft an action plan for how you would create or replicate a pirate's sea shanty in R.

Please review Q1, Q2, and Q3. We will work on Q4 as a group.

Exercise: Navigating tricky waters (2)

- **Q1.** What is a sea shanty? What resources will you require to find one and replicate it? How will you transcribe the music? Do you require additional resources to figure out how to do this?

- **Q2.** Can R play music? Are there packages that support this?

- **Q3.** Can you find examples of others' code to help with these tasks? Explain how you would go about this?

Exercise: Navigating tricky waters (3)

- **Q1.** What is a sea shanty? What resources will you require to find one and replicate it? How will you transcribe the music? Do you require additional resources to figure out how to do this?
  - A sea shanty is a simple folk song that was designed to be sung at sea while working without instrumental accompaniment.
  - You will need to find an example of a sea shanty; and search for the melody notes: e.g., “sea shanty sheet music free”
  - If you can’t find free resources: see the pdf sheet music for “Wellerman” on Dropbox
    - Note: may need a resource for how to read sheet music: this can be found here
    - https://www.instructables.com/How-to-Read-Sheet-Music-for-Beginners/

Exercise: Navigating tricky waters (4)

- **Q2.** Can R play music? Are there packages that support this?

- There are several packages available in R to help program and code music; check out the following links:
  - https://cran.r-project.org/web/packages/audio/
  - https://cran.r-project.org/web/packages/sound/
  - https://cran.r-project.org/web/packages/music/
  - https://cran.r-project.org/web/packages/tuneR/
  - https://cran.r-project.org/web/packages/gm/

Exercise: Navigating tricky waters (5)

- **Q3.** Can you find examples of others' code to help with these tasks? Explain how you would go about this?

- A great example/tutorial can be found here
- https://towardsdatascience.com/compose-and-play-music-in-r-with-the-rmusic-package-b2afa90761ea

- An older stack overflow post with instructions and explanation: https://stackoverflow.com/questions/31782580/how-can-i-play-birthday-music-using-r

Exercise: Navigating tricky waters (6)

- **Q4.** Briefly, using the answers to the prompts in Q1-3, list or sketch out what your analysis plan will look like for completing this task.

  - Step 1:
  - Step 2:
  - Step 3:
  - Etc.

Step 1:
Find an R package that can handle music; preferably one with a tutorial as well (e.g., "Rmusic").

Step 2:
Identify inputs for Rmusic package: E.g., (1) notes; (2) duration.

Step 3:
Investigate Sea Shanties; select one and find sheet music for this.

Step 4:
Use online resource to translate sheet music into something readable for R: E.g., notes and duration.

Step 5:
Install required R packages; translate sheet music into R script.

Step 6:

Independent Study and Drop-In Office Hours

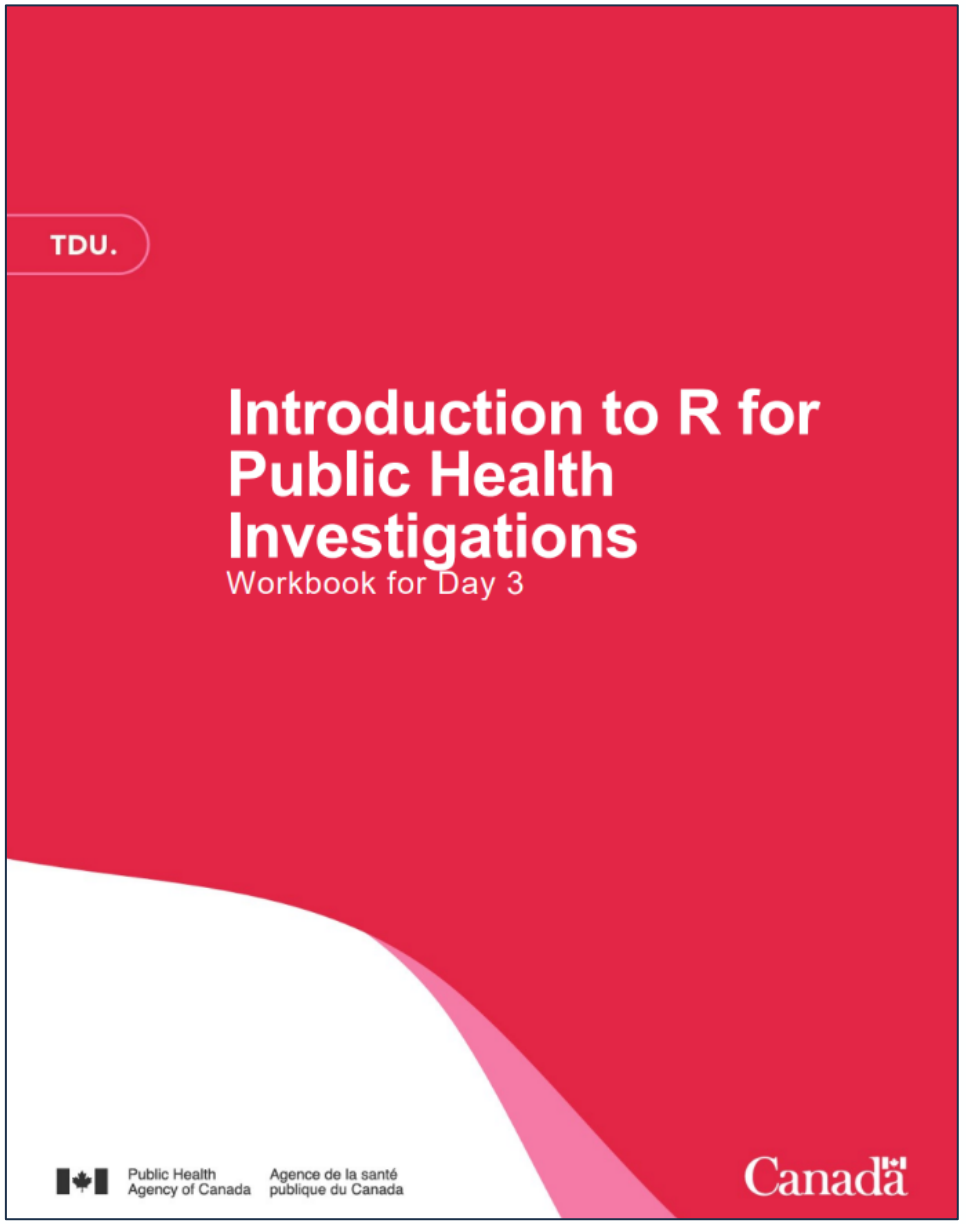
Practice makes perfect

Independent study

Scenario: You will combine provincial health insurance roster, physician billings, and hospitalization data so that you can identify cases of Asthma and describe its occurrence in a small province.

For this exercise you will:

- Clean and process data
- Append and merge the datasets
- Apply an administrative case definition
- Analyse and visualise the data
- Build an automated report in R markdown



Approach

- We recommend:
  - **Novice users:** Use the workbook and R script(s) provided on GitHub as a guide. Run the available scripts and prioritize your understanding what each chunk of code and functions used are doing. Do not worry about being able to write or debug code.
  - **Beginner/Intermediate users:** R code is provided as a screen capture image in the workbook. You should have sufficient understanding of coding to get a general sense of what the code is doing by reading it or doing a little research. We would like you write the code out from the guide as you progress through the scenario. Cross reference to the R script(s) provided on GitHub if you encounter any tangly problems.
  - **Advanced users:** We encourage you to try writing your own code where you like and contrast it with the code used for the exercise, and to help your peers as questions arise. Cross reference to the R script(s) provided on GitHub if you encounter any tangly problems.

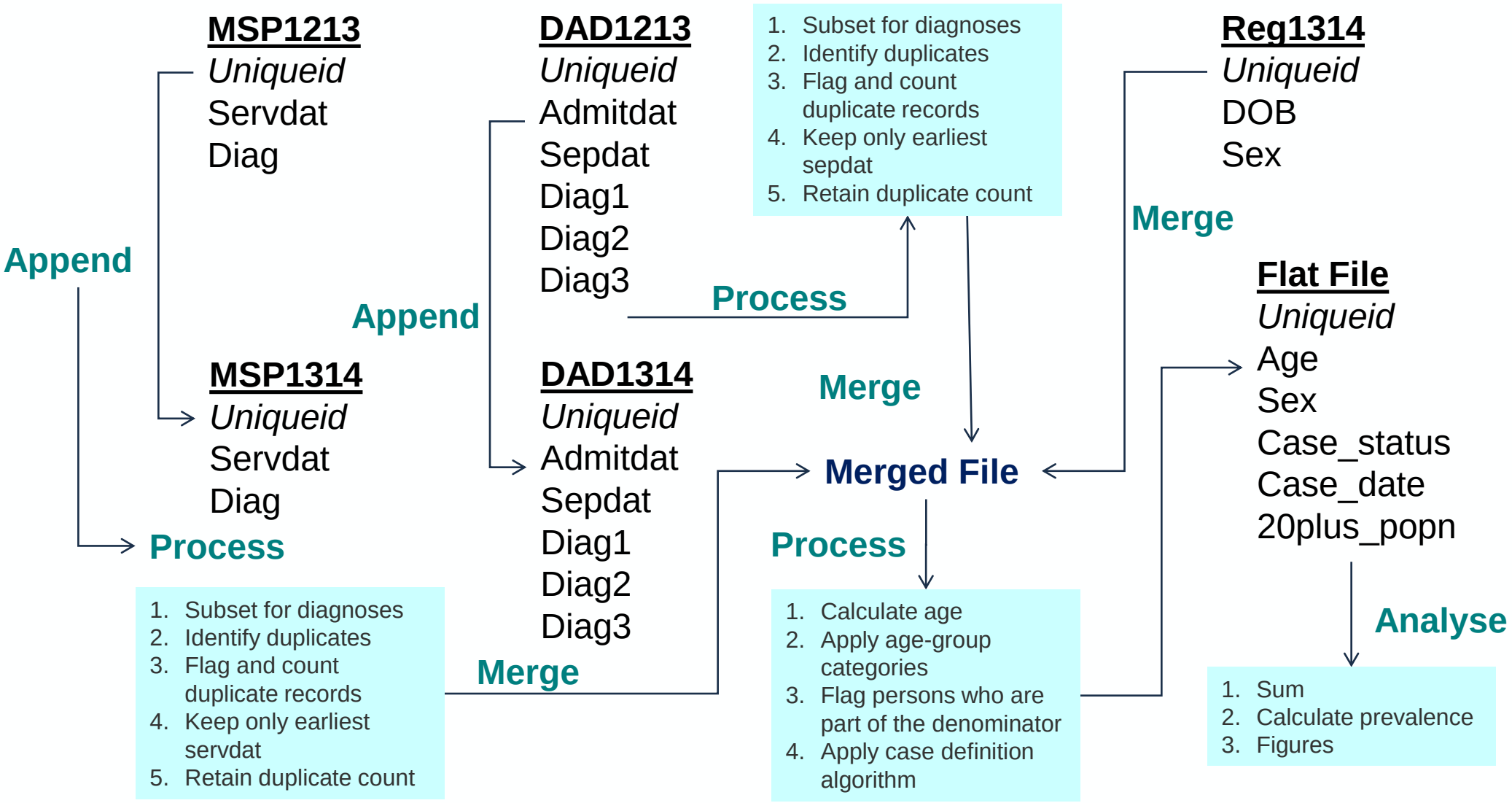
Independent Study and Drop-In Office Hours

- We will walk through the first steps of the exercise to ensure that everyone is able to get started (optional attendance)
- You are free to stay in the virtual classroom or leave while you work through the exercise
- We'll be here in the virtual classroom to answer your questions as they arise
- Please return by 12:15 Pacific / 13:15 Mountain / 14:15 Central / 15:15 Eastern / 16:15 Atlantic / 16:45 NFLD

Goal

uniqueid	dob	age	sex	asthma_case	asthma_casedate	popn_1plus
1	1936-09-21	76	1	0	*NA*	1
2	1916-04-08	97	2	1	2012-09-24	1
3	1924-10-17	88	1	0	*NA*	1
4	1995-04-01	18	2	0	*NA*	1
5	1997-06-28	16	1	0	*NA*	1

Data Collation



Trivia: Sailing the seven seas

The tallest recorded wave was observed in1958 following an earthquake off the coast of Alaska. How big was the tsunami?

A. Less than 50 meters
B. 50-100 meters
C. 100-200 meters
D. 300-400 meters
E. 400+ meters

We will start at 12:15 Pacific / 13:15 Mountain / 14:15 Central / 15:15 Eastern / 16:15 Atlantic / 16:45 NFLD

Please take a moment to ensure that you have downloaded course materials for today, refresh your beverage, and / or network with us.

Questions?

Wrap up

Independent study

- Complete exercise 3 as we will debrief when we meet tomorrow

- Please reach out to your course facilitators if you have questions

Feedback for day 3

- Please take a moment to answer a few questions
- https://www.slido.com/
- #IntroR2025