TDU

# Introduction to R for Public Health Investigations

Workbook for day 3

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

Canada

# Contents

## Practical Exercise

### Instructions

Learners are provided with a scenario, questions, and tasks with associated code to perform each task. We recommend:

> Novice users (Boatswains): Use this workbook and R script(s) provided on GitHub. Using this workbook as a guide, run the code we've provided piece by piece to understand what each chunk of code does, and what various functions are doing. At this point don't worry too much about being able to write or debug code.
>
> Beginner/Intermediate users (First Mates): R code is provided as a screen capture image in this workbook. You should have sufficient understanding of coding to get a general sense of what the code is doing by reading it (with the assistance of the help documentation, a few Google searches as needed, and comments in the R scripts we've provided on GitHub). It is our intention to have 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 (Master Mariners): 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.

Get as far as you can with this exercise within 4 hours and 10 mins (maximum). Don't worry if you need extra time. The learning curve for R is steep and learners will benefit most from dedicated time for practicing. Reach out to your course facilitators by Slack or by email if you require assistance with the course material.

## Introduction

In this session you will apply your R and data management skills to prepare and collate data from multiple datasets, apply a case identification algorithm, and conduct a basic descriptive epidemiological analysis. You have been provided with five datasets which are mock extracts from administrative datasets (physician billings, discharges from hospital, and provincial health insurance client roster) and have been simplified for training purposes. These datasets must be combined in order to efficiently apply administrative case definitions (algorithms) so that you can describe the occurrence of health service use for chronic conditions among residents of a small province. You've been provided with the following datasets and case definition:

Physician billings: The msp1213 and msp1314 datasets contain information from fee-for-service physician billings (general practitioner and specialists) in the 2012/13 and 2013/14 fiscal years respectively. Each record equates to a physician visit and a date on which the visit occurred and a single diagnosis code. While persons may have multiple conditions to discuss during their visit with the physician, the physician selects a "most responsible" condition for billing.

Discharges from hospital: The dad1213 and dad1314 datasets contain information from the discharge abstract produced upon discharge from hospital. The data includes acute care hospitalisations only as other hospitalisations, ambulatory or long-term care hospital services are excluded. Multiple diagnoses are included on the discharge abstract, along with the dates on which the individual was admitted to and discharged from the hospital.

Provincial health insurance client roster: The reg1314 dataset contains a listing of every person who is registered for provincial health insurance (e.g. MSP, OHIP, MCP, etc.). Enrolment is automatic upon issue of a birth certificate among residents of the province (in the case of new births) or applied for upon moving to the province. Registration is cancelled upon relocation and registration for provincial health insurance in a new province or death. Publically funded health care services are billed to the province of residence, or federal program providing health insurance coverage, or must be paid for out of pocket or by private insurance by those who are not registered. Regardless, persons who access health care services and who aren't registered with the provincial Medical Services Plan have been excluded from this dataset along with registrants who do not reside in the province.

Table 1: Contents of datasets provided for Introduction to R, Day 3

| Dataset | Fields | Type | Definition |
|---|---|---|---|
| msp1213.csv | uniqueid | String | Unique identifier (person) |

| Dataset | Fields | Type | Definition |
|---|---|---|---|
| | servdat | Date | Date on which physician visit occurred |
| | diag1 | String | Diagnosis (ICD9) code provided on billing |
| msp1314.csv | uniqueid | String | Unique identifier (person) |
| | servdat | Date | Date on which physician visit occurred |
| | diag1 | String | Diagnosis (ICD9) code provided on billing |
| dad1213.csv | uniqueid | String | Unique identifier (person) |
| | admitdat | Date | Date admitted to hospital |
| | sepdat | Date | Date discharged from hospital |
| | diag1 | String | Diagnosis (ICD9) code listed on discharge |
| | diag2 | String | Diagnosis (ICD9) code listed on discharge |
| | diag3 | String | Diagnosis (ICD9) code listed on discharge |
| dad1314.csv | uniqueid | String | Unique identifier (person) |
| | admitdat | Date | Date admitted to hospital |
| | sepdat | Date | Date discharged from hospital |
| | diag1 | String | Diagnosis (ICD9) code listed on discharge |
| | diag2 | String | Diagnosis (ICD9) code listed on discharge |
| | diag3 | String | Diagnosis (ICD9) code listed on discharge |
| reg1314.csv | uniqueid | String | Unique identifier (person) |
| | dob | Date | Date of birth |
| | sex | Numeric | F = 1, M = 2 |

**CCDSS Asthma definition**[1]: The case definition of diagnosed asthma is: an individual aged one year and older having at least two visits to a physician with a diagnosis of asthma in the first diagnostic field in a two-year period, or at least one hospital separation with a diagnosis of asthma ever in any diagnostic field, coded by the International Classification of Diseases (ICD), ninth revision or ICD-9-CM 493 or ICD-10-CA J45-46. Note that the application of this case definition in this exercise is highly simplified.

[1] https://www.canada.ca/en/public-health/services/publications/diseases-conditions/asthma-chronic-obstructive-pulmonary-disease-canada-2018.html#a1.2.1

## 1. Set up your workspace

1. Create new folders on your computer to organize the files for today’s exercises similar to how they were set up for exercise 1 and 2:
   a. Within the IntoToR folder, create a subfolder for Day 3 called: "Exercise_Day3".
   b. Create new folders within Exercise_Day3 called:
      i. "data";
      ii. "scripts"
      iii. “output”
   c. Move the files for Day 3 from to their corresponding folders.

   **Note: for Day 3 we’ll be working from a single .Rmd file in your scripts folder for this exercise, rather than writing several individual scripts.**

   Tips:

   - Use the same RProject from exercise 1 and 2
   - Clean up your environment in RStudio if you have previous work still in it with the following code: rm(list=ls())
   - Remember to save your work regularly

2. Now that you have set up your project folders:

   Open the datasets in excel (msp1213.csv, msp1314.csv, dad1213.csv, dad1314.csv, reg1314.csv).

   a. Is uniqueid really unique? What do you think each record represents?

   b. What field(s) do you think might be useful for linking data across tables?

   c. Refer back to the administrative case definition. What datasets and fields do you think might be needed in identifying cases of asthma and their case dates?

d. Given the administrative case definition and the data you have, what descriptive epidemiological calculations would you like to perform to describe the occurrence of asthma in this population?

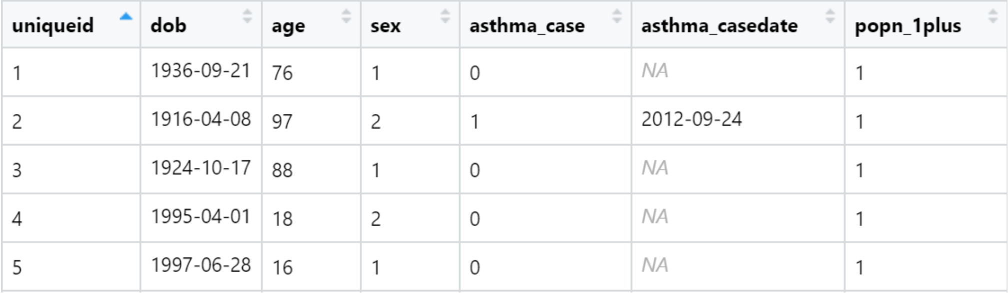
e. Considering that reg1314 is a Medical Services Plan client registry for the 2013/14 fiscal year, do you think that this is acceptable for use as a population registry? Why or why not? What limitations might there be if this were to be used as a population registry?

f. What additional information do you need from the other four tables to add to the reg1314 dataset to carry out your descriptive analysis of asthma?

g. How does this information need to be laid out as data fields in the reg1314 dataset so that you can perform your descriptive analysis? What steps do you need to take during data processing to include these data fields in the reg1314 dataset?

   *Hint – draw the resulting file. Having your starting point (i.e. datasets provided) and end point (i.e. ultimate flat file) clearly in mind will allow you to identify the specific steps you need to take during data processing.*

In this exercise, the steps below will be taken to create this flat file:



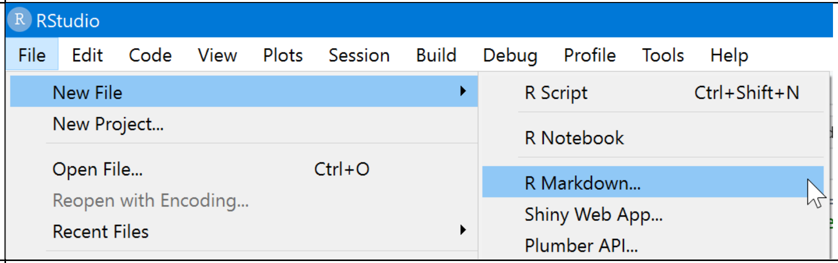
| uniqueid | dob | age | sex | asthma_case | asthma_casedate | popn_1plus |
|---|---|---|---|---|---|---|
| 1 | 1936-09-21 | 76 | 1 | 0 | NA | 1 |
| 2 | 1916-04-08 | 97 | 2 | 1 | 2012-09-24 | 1 |
| 3 | 1924-10-17 | 88 | 1 | 0 | NA | 1 |
| 4 | 1995-04-01 | 18 | 2 | 0 | NA | 1 |
| 5 | 1997-06-28 | 16 |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

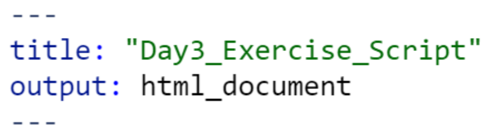
Overview of steps to be taken during this exercise:

- Append DAD
- Flag hospital separations with a diagnosis of asthma
    - keep only asthma-related hospitalisations
    - identify duplicates
    - flag and count duplicate records for each uniqueid, keep only earliest sepdat
- Save DAD data as new appended file
- Append MSP
- Flag physician visits with a diagnosis of asthma
    - keep only asthma-related physician visits
    - identify duplicates
    - flag and count duplicate records for each uniqueid, keep only earliest servdat
- Save MSP data as new appended file
- Merge registry to DAD and MSP data
    - Merge DAD and Registry (full join, uniqueid)
    - Merge MSP and Registry (full join, uniqueid)
    - Merge two resultant files (full join, uniqueid dob sex
- Save as new registry file with chronic disease status information
- Apply case definition: Flag persons who meet the case definition
- Calculate case date (earliest health service coded for asthma)
- Prepare registry for descriptive epi calculations
    - Calculate age at midyear 2013
    - Flag persons who are part of the denominator
    - Apply age group categories
- Analyse the data
    - Sum case definition (numerator) flag and denominator flag, calculate rate
    - Sum case definition (numerator) flag and denominator flag (by agegroup and/or sex), calculate rates
    - Create figures

## 2. Set your RStudio session

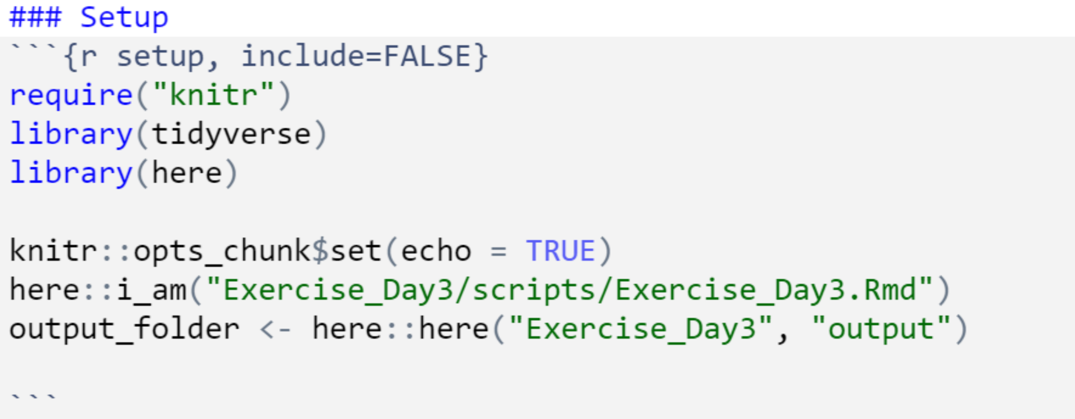
Note: Try adding your own comments to the code to keep track of critical things like counts of records and variables (to ensure operations are behaving as expected) and to clarify the code for your understanding (to ensure that you could understand weeks from now what you produced through the course of this exercise). Note that there is a master rmd file provided for this exercise (with comments) to help you should things get tangly. As this exercise is complicated, please do not hesitate to refer to the rmd file provided to run through the exercise, whether you look to develop your understanding by following along with the master rmd file, copying and pasting code into your own rmd file, or typing your own code out from the images provided into your own fresh rmd file. Regardless of your approach to this exercise, it is critical that you pay attention to changes in your files (number of variables and observations for instance) to ensure linkages behave as expected.

<table>
<tr><th>Task</th><th>Code</th></tr>
<tr><td>Open a new R Markdown file and select HTML as the default output format and call it:<br>“Day3_Exercise_script.Rmd”</td><td>


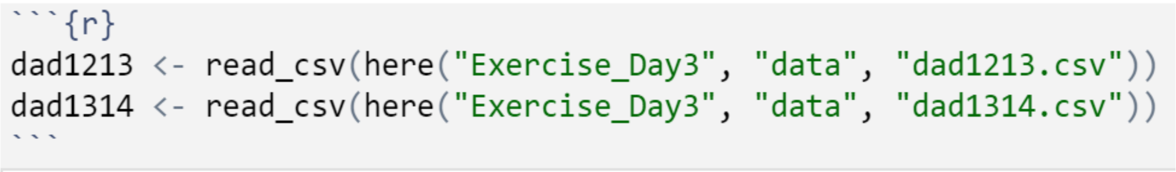
</td></tr>
<tr><td>Note that for the Day 3 exercise, we’ll be operating out of a single R-markdown file rather than sourcing several files all at once and treating our R-markdown file as a notebook in HTML format. First, you’ll need to</td><td>


```
---
title: "Day3_Exercise_Script"
output: html_document
---
```

</td></tr>
</table>

| | |
|---|---|
| specify this format in the YAML header at the top! | |
| Next, you'll need a code chunk to setup your document for analyses. Here, we'll directly load the libraries we plan to use, initialize the location of the present Rmd script, and set up the location of out 'output' folder pathway for use later. | <br>`### Setup`<br>`` ```{r setup, include=FALSE} ``<br>`require("knitr")`<br>`library(tidyverse)`<br>`library(here)`<br><br>`knitr::opts_chunk$set(echo = TRUE)`<br>`here::i_am("Exercise_Day3/scripts/Exercise_Day3.Rmd")`<br>`output_folder <- here::here("Exercise_Day3", "output")`<br><br>`` ``` `` |

3. Prepare the data:

| | |
|---|---|
| Load hospitalisation datasets (dad1213.csv and dad1314.csv) and store them as objects in memory. Note the number of observations and variables in each.<br><br>Note: If you need to add additional chunks to your R Markdown file you can either click on Insert -> R at the top of the Markdown toolbar or | <br>`` ```{r} ``<br>`dad1213 <- read_csv(here("Exercise_Day3", "data", "dad1213.csv"))`<br>`dad1314 <- read_csv(here("Exercise_Day3", "data", "dad1314.csv"))`<br>`` ``` `` |

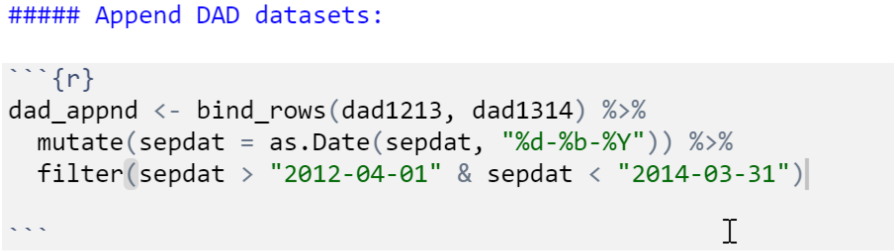
<table>
<tr>
<td>use the keyboard shortcut Ctrl + Alt + I ( Cmd + Option + I on macOS)</td>
<td></td>
</tr>
<tr>
<td>

Append hospitalisation datasets (using bind_rows). How many hospitalisations are there in each file?

Confirm separation dates fall in the correct fiscal year. How many records need to be dropped?

Note: If you only want to run part of the chunk of code, use the Run dropdown at the top of the markdown toolbar.

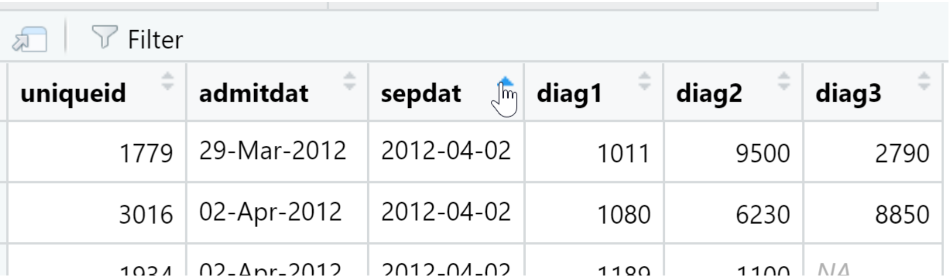
</td>
<td>



~~~
##### Append DAD datasets:

```{r}
dad_appnd <- bind_rows(dad1213, dad1314) %>%
  mutate(sepdat = as.Date(sepdat, "%d-%b-%Y")) %>%
  filter(sepdat > "2012-04-01" & sepdat < "2014-03-31")

```
~~~

Hint: try entering print(unique(dad_appnd$sepdat)) in the command window

Or, view and sort the file by double clicking on the file in the environment panel and clicking on the set of triangles next to the variable name:

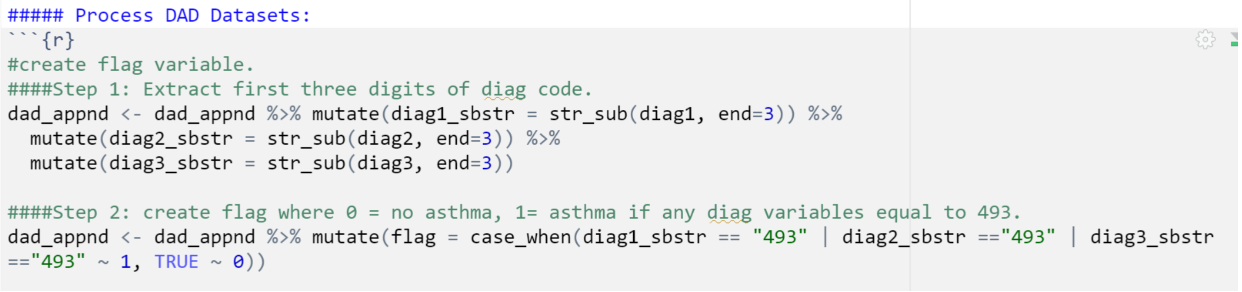


| uniqueid | admitdat | sepdat | diag1 | diag2 | diag3 |
|---|---|---|---|---|---|
| 1779 | 29-Mar-2012 | 2012-04-02 | 1011 | 9500 | 2790 |
| 3016 | 02-Apr-2012 | 2012-04-02 | 1080 | 6230 | 8850 |
| 1934 | 02-Apr-2012 | 2012-04-02 | 1189 | 1100 | NA |

</td>
</tr>
</table>

Flag and keep hospitalisations where a diagnosis of asthma was given. As we are interested in the whole diagnosis code block 493, we will take a substring of that code to filter upon.How many asthma-related hospitalisations were there in 2013/14 and 2012/13?

To practice: how many unique persons (hint: file size at the end of this code)? What is the maximum number of hospitalisations per person during the period of interest (hint: view and sort, or table or count functions)?

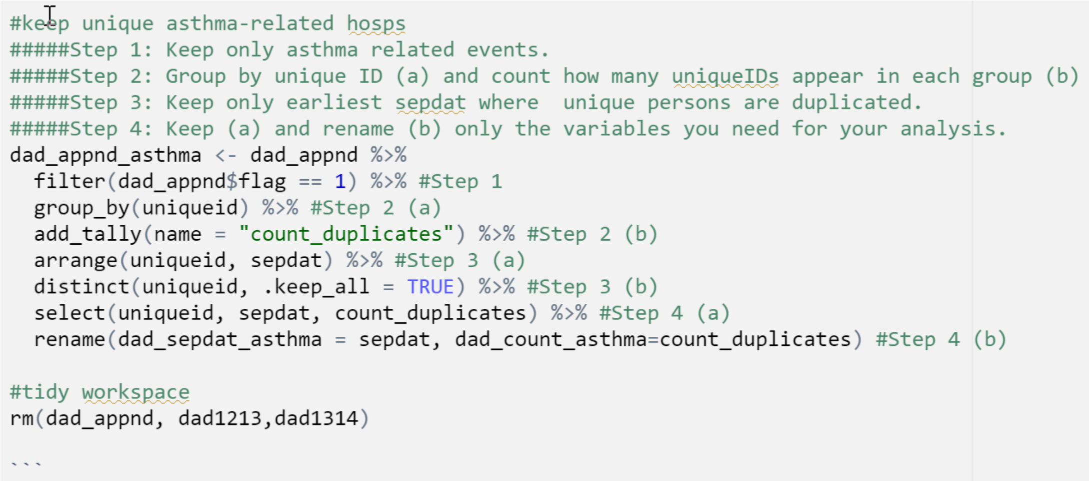


~~~
##### Process DAD Datasets:
```{r}
#create flag variable.
####Step 1: Extract first three digits of diag code.
dad_appnd <- dad_appnd %>% mutate(diag1_sbstr = str_sub(diag1, end=3)) %>%
  mutate(diag2_sbstr = str_sub(diag2, end=3)) %>%
  mutate(diag3_sbstr = str_sub(diag3, end=3))

####Step 2: create flag where 0 = no asthma, 1= asthma if any diag variables equal to 493.
dad_appnd <- dad_appnd %>% mutate(flag = case_when(diag1_sbstr == "493" | diag2_sbstr =="493" | diag3_sbstr
=="493" ~ 1, TRUE ~ 0))
~~~

As there are individuals who have been hospitalised more than once in the time period, keep only the earliest separation date. How does the distinct function work to remove duplicates? Do the data need to be sorted before using this function? Should the data be sorted ascending or descending, and on which variable(s)?

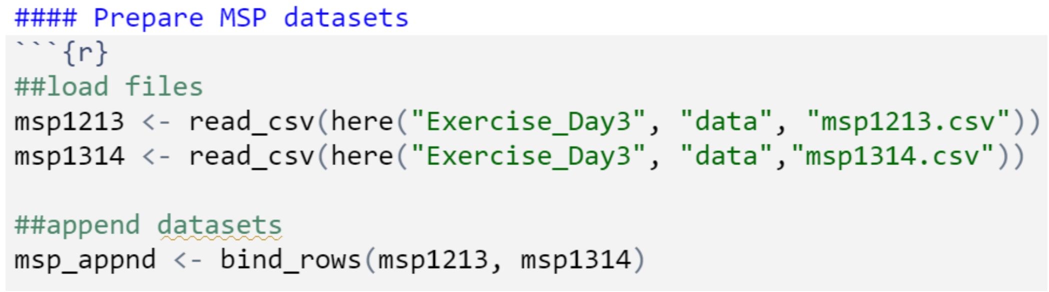
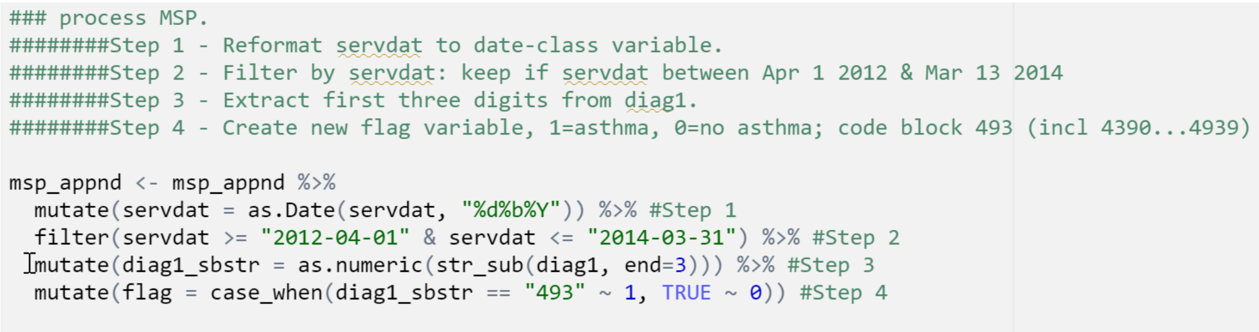
Keep and rename only the variables you need for later (i.e., uniqueid, sepdat, count_duplicates)

Tidy the data and work environment.



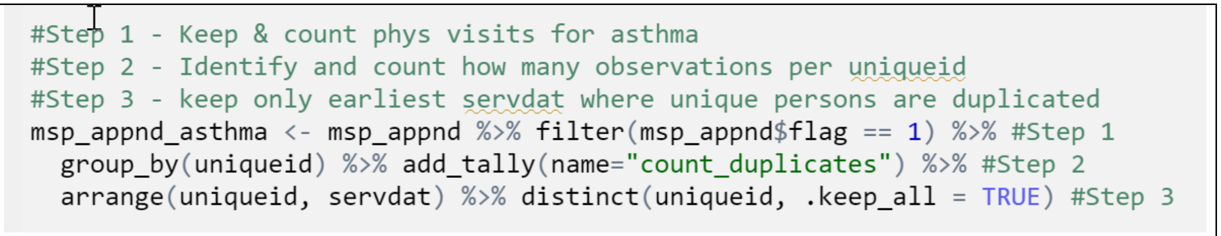
~~~
#keep unique asthma-related hosps
#####Step 1: Keep only asthma related events.
#####Step 2: Group by unique ID (a) and count how many uniqueIDs appear in each group (b)
#####Step 3: Keep only earliest sepdat where  unique persons are duplicated.
#####Step 4: Keep (a) and rename (b) only the variables you need for your analysis.
dad_appnd_asthma <- dad_appnd %>%
  filter(dad_appnd$flag == 1) %>% #Step 1
  group_by(uniqueid) %>% #Step 2 (a)
  add_tally(name = "count_duplicates") %>% #Step 2 (b)
  arrange(uniqueid, sepdat) %>% #Step 3 (a)
  distinct(uniqueid, .keep_all = TRUE) %>% #Step 3 (b)
  select(uniqueid, sepdat, count_duplicates) %>% #Step 4 (a)
  rename(dad_sepdat_asthma = sepdat, dad_count_asthma=count_duplicates) #Step 4 (b)

#tidy workspace
rm(dad_appnd, dad1213,dad1314)

```
~~~

*Note: The number of records remaining should equal the number of unique individuals obtained above*

<table>
<tr>
<td>Append physician visit datasets (msp1213.csv and msp1314.csv). How many physician visits are there in each file? How many in the appended file?</td>
<td>

<pre>
#### Prepare MSP datasets
```{r}
##load files
msp1213 <- read_csv(here("Exercise_Day3", "data", "msp1213.csv"))
msp1314 <- read_csv(here("Exercise_Day3", "data","msp1314.csv"))

##append datasets
msp_appnd <- bind_rows(msp1213, msp1314)
</pre>
<i>Tip: If writing from scratch, consider recycling DAD code</i>
</td>
</tr>
<tr>
<td>Similar to what we did for the DAD datasets, we now need to process the MSP datasets.<br><br>For this we want to reformat servdat to a date-class variable to be used for filtering; keep dates between Apr 1, 2012 and Mar 13, 2014; then use the physician codes starting with 493 to create a flag for asthma.</td>
<td>
<i>##### Process MSP Datasets:</i>

<pre>
### process MSP.
########Step 1 - Reformat servdat to date-class variable.
########Step 2 - Filter by servdat: keep if servdat between Apr 1 2012 & Mar 13 2014
########Step 3 - Extract first three digits from diag1.
########Step 4 - Create new flag variable, 1=asthma, 0=no asthma; code block 493 (incl 4390...4939)

msp_appnd <- msp_appnd %>%
  mutate(servdat = as.Date(servdat, "%d%b%Y")) %>% #Step 1
  filter(servdat >= "2012-04-01" & servdat <= "2014-03-31") %>% #Step 2
  mutate(diag1_sbstr = as.numeric(str_sub(diag1, end=3))) %>% #Step 3
  mutate(flag = case_when(diag1_sbstr == "493" ~ 1, TRUE ~ 0)) #Step 4
</pre>
</td>
</tr>
</table>
```

Drop all non-asthma related physician visits. How many observations remain (hint: file size)? What are the maximum number of physician visits coded for asthma per person (hint: view and sort, or table and count functions)?

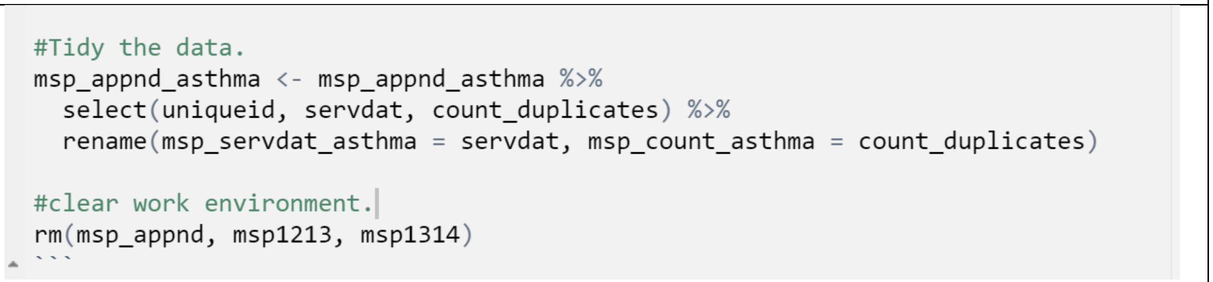
As there are individuals who have seen a physician more than once in the time period, keep only the earliest visit date. How many unique individuals have seen a physician where their visit was coded for asthma (hint: view and sort, or table and count functions)?



```
#Step 1 - Keep & count phys visits for asthma
#Step 2 - Identify and count how many observations per uniqueid
#Step 3 - keep only earliest servdat where unique persons are duplicated
msp_appnd_asthma <- msp_appnd %>% filter(msp_appnd$flag == 1) %>% #Step 1
  group_by(uniqueid) %>% add_tally(name="count_duplicates") %>% #Step 2
  arrange(uniqueid, servdat) %>% distinct(uniqueid, .keep_all = TRUE) #Step 3
```

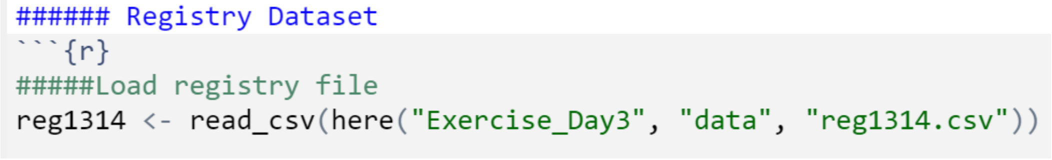
Tidy the data and work environment

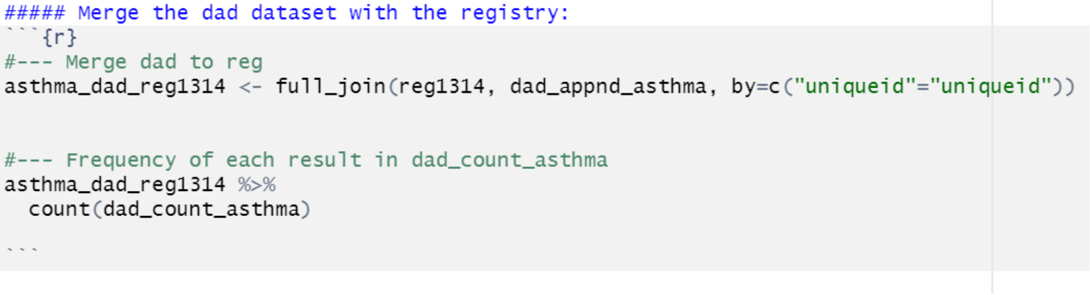
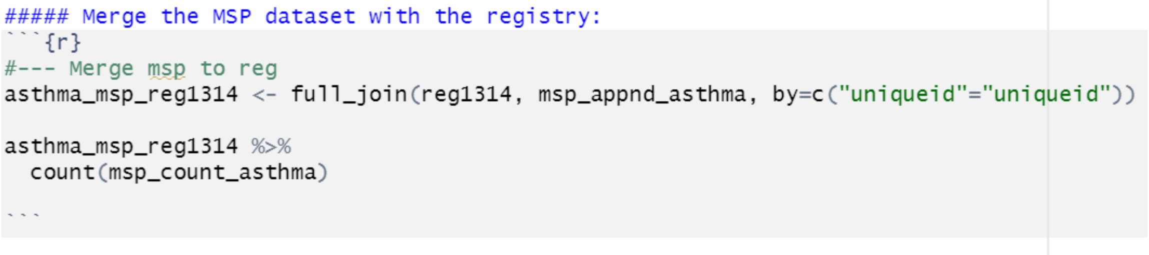
Keep and rename only the variables you need for later (i.e. uniqueid, servdat, count_duplicates)



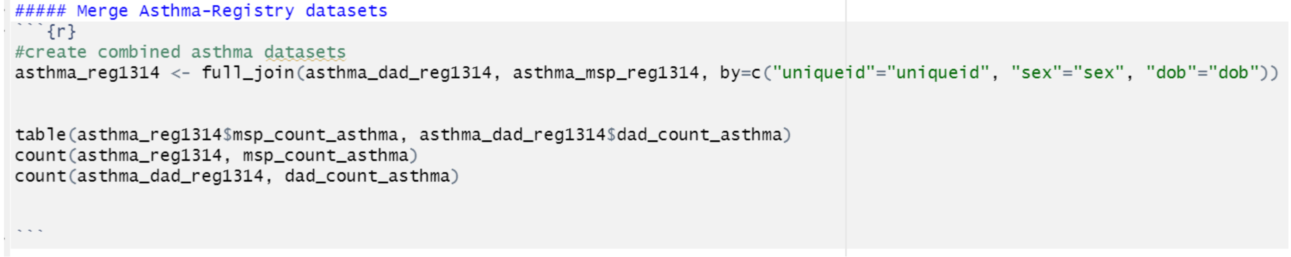
```
#Tidy the data.
msp_appnd_asthma <- msp_appnd_asthma %>%
  select(uniqueid, servdat, count_duplicates) %>%
  rename(msp_servdat_asthma = servdat, msp_count_asthma = count_duplicates)

#clear work environment.
rm(msp_appnd, msp1213, msp1314)
```

## 3. Clean, process, and assemble the data

| Task | Code |
|---|---|
| Load reg1314. How many registrants are there? | <br>###### Registry Dataset<br>```{r}<br>#####Load registry file<br>reg1314 <- read_csv(here("Exercise_Day3", "data", "reg1314.csv")) |
| In order to assemble MSP, DAD, and registry data, what type of merge(s) will you need? How will you merge all three files (i.e., what key variable or variables) and in what steps? Provide your justifications.<br>Tip: Review the join types in your participant guide | |

| | |
|---|---|
| Merge hospitalisation data with the registry. How many records are there in the merged datafile? How many registrants in 2013/14 were hospitalised where a diagnosis of asthma was given in the previous two years? | <br>##### Merge the dad dataset with the registry:<br>```{r}<br>#--- Merge dad to reg<br>asthma_dad_reg1314 <- full_join(reg1314, dad_appnd_asthma, by=c("uniqueid"="uniqueid"))<br><br>#--- Frequency of each result in dad_count_asthma<br>asthma_dad_reg1314 %>%<br>&nbsp;&nbsp;count(dad_count_asthma)<br><br>```<br><br>*Hint: You can find the count of observations in the resulting merged file in the environment window. View the merged file and see how many individuals have data pertaining to asthma hospitalisations. The number of persons should equal the number of persons in the dad_appnd_asthma file. The merged file also contains records pertaining to everyone else in the registry.* |
| Merge physician visit data with the registry. How many records are there in the merged data file? How many registrants in 2013/14 visited a physician where the billing was coded for asthma in the previous two years? | <br>##### Merge the MSP dataset with the registry:<br>```{r}<br>#--- Merge msp to reg<br>asthma_msp_reg1314 <- full_join(reg1314, msp_appnd_asthma, by=c("uniqueid"="uniqueid"))<br><br>asthma_msp_reg1314 %>%<br>&nbsp;&nbsp;count(msp_count_asthma)<br><br>```<br><br>*Hint: You can find the count of observations in the resulting merged file in the environment window. View the merged file and see how many individuals have data pertaining to asthma visits. The number of persons should equal the number of persons in the msp_appnd_asthma file. The merged file also contains records pertaining to everyone else in the registry.* |

Combine asthma datasets. How many observations are there in the resulting file? How many people are found in both datasets?



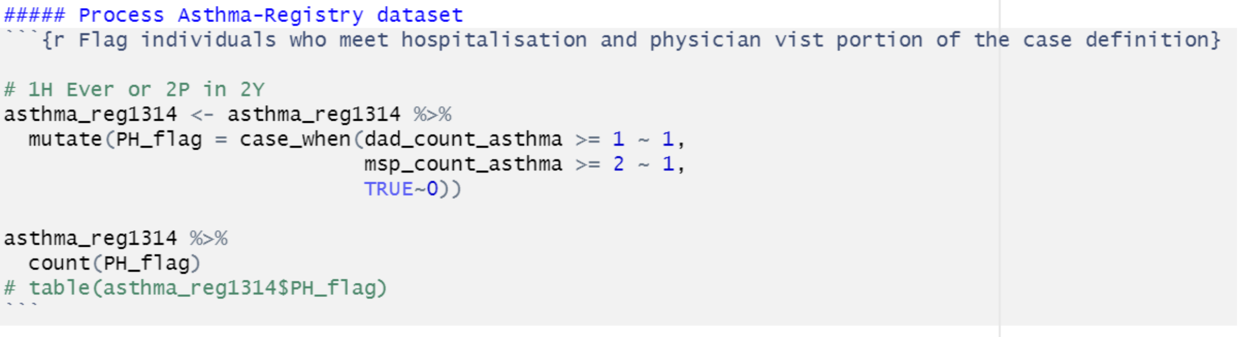
~~~
##### Merge Asthma-Registry datasets
```{r}
#create combined asthma datasets
asthma_reg1314 <- full_join(asthma_dad_reg1314, asthma_msp_reg1314, by=c("uniqueid"="uniqueid", "sex"="sex", "dob"="dob"))


table(asthma_reg1314$msp_count_asthma, asthma_dad_reg1314$dad_count_asthma)
count(asthma_reg1314, msp_count_asthma)
count(asthma_dad_reg1314, dad_count_asthma)


```
~~~

*Note: As sex and dob come from same file, mismatches would indicate bad things have happened - don't want duplicated columns.*

*Hint:*

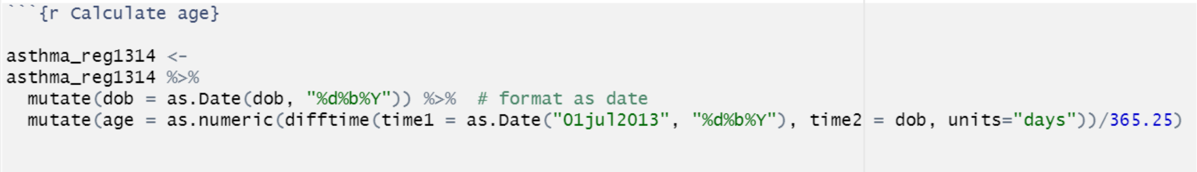
- *You can find the count of observations in the resulting merged file in the environment window.*
- *Using the table command, we can see a cross tabulation of count of hospitalisations by physician visits. This table will illustrate persons who have shown up in both files, and what their pattern of health service utlisation for asthma was in the period of interest.*
- *Using the count command we can see the number of persons in the dataset broken down by count of physician visits for asthma or hospitalisations for asthma.*

<table>
<tr>
<td>How many individuals meet the hospitalisation and physician component of the case definition for asthma? What are the other components of this definition?</td>
<td>



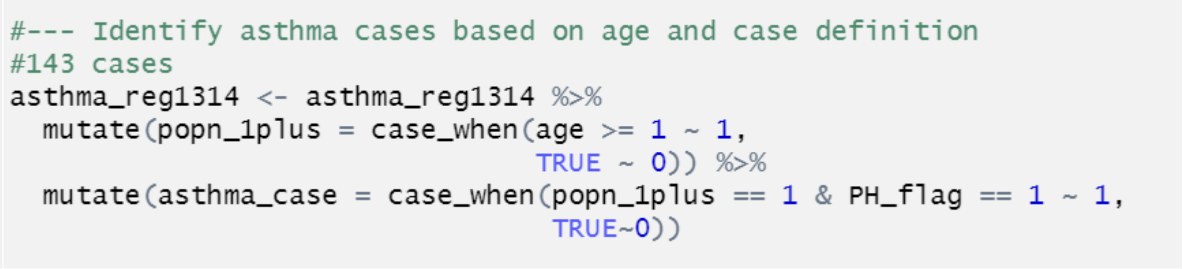
~~~
##### Process Asthma-Registry dataset
```{r Flag individuals who meet hospitalisation and physician vist portion of the case definition}

# 1H Ever or 2P in 2Y
asthma_reg1314 <- asthma_reg1314 %>%
  mutate(PH_flag = case_when(dad_count_asthma >= 1 ~ 1,
                             msp_count_asthma >= 2 ~ 1,
                             TRUE~0))


asthma_reg1314 %>%
  count(PH_flag)
# table(asthma_reg1314$PH_flag)
```
~~~

</td>
</tr>
<tr>
<td>Calculate case age in 2013/14. What reference date would you use for this calculation?</td>
<td>



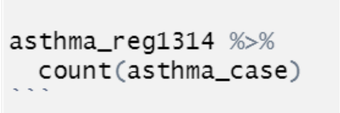
~~~
```{r Calculate age}

asthma_reg1314 <-
asthma_reg1314 %>%
  mutate(dob = as.Date(dob, "%d%b%Y")) %>%  # format as date
  mutate(age = as.numeric(difftime(time1 = as.Date("01jul2013", "%d%b%Y"), time2 = dob, units="days"))/365.25)
~~~

</td>
</tr>
<tr>
<td>How will you identify who should be included in the case definition based on age? Remember you will need to obtain denominator data from the registry for your descriptive epi calculations.</td>
<td>



```
#--- Identify asthma cases based on age and case definition
#143 cases
asthma_reg1314 <- asthma_reg1314 %>%
  mutate(popn_1plus = case_when(age >= 1 ~ 1,
                                TRUE ~ 0)) %>%
  mutate(asthma_case = case_when(popn_1plus == 1 & PH_flag == 1 ~ 1,
                                 TRUE~0))
```

</td>
</tr>
<tr>
<td>How many individuals met the case definition for asthma in 2013/14?</td>
<td>



~~~
asthma_reg1314 %>%
  count(asthma_case)
```
~~~

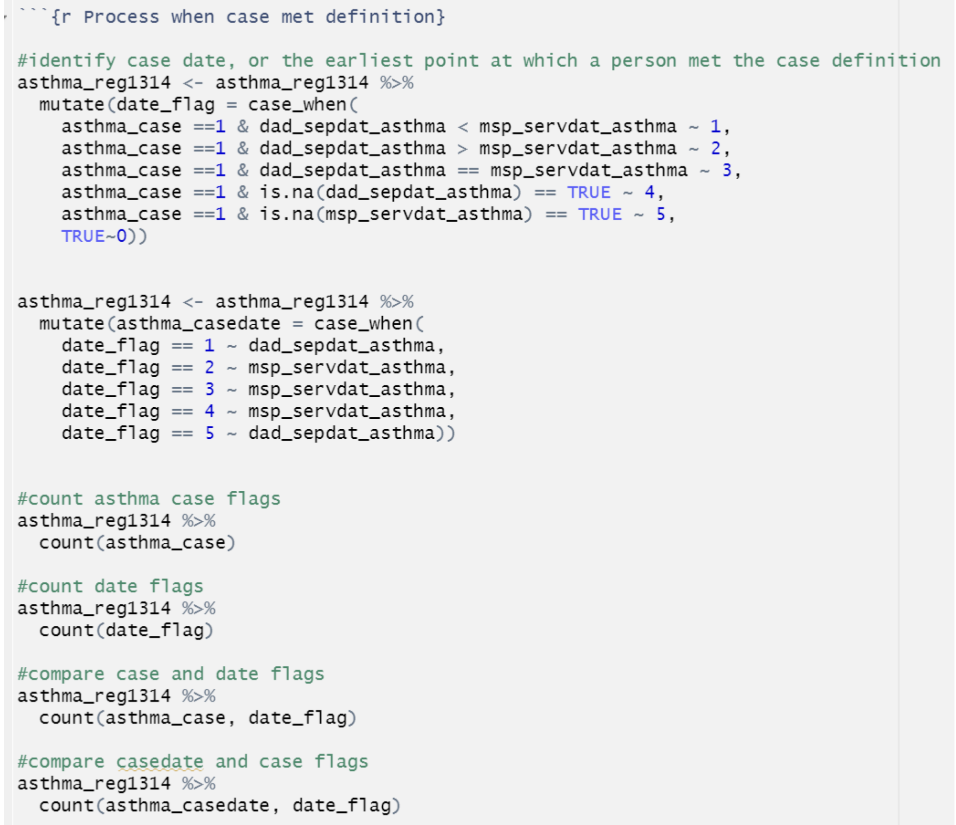
</td>
</tr>
</table>

<table>
<tr>
<td>

Specify the case date, or the date of the earliest health service coded for asthma. We will do this by creating a series of flags based on hospital separation date and physician visit date. We can use these flags to identify and subsequently recode to specify which date to use as the case date, or the earliest point a person met the case definition.

Check your logic statements through appropriate tabulations and cross tabulations (i.e., see count functions to the right and in the rmd file for this exercise).

</td>
<td>

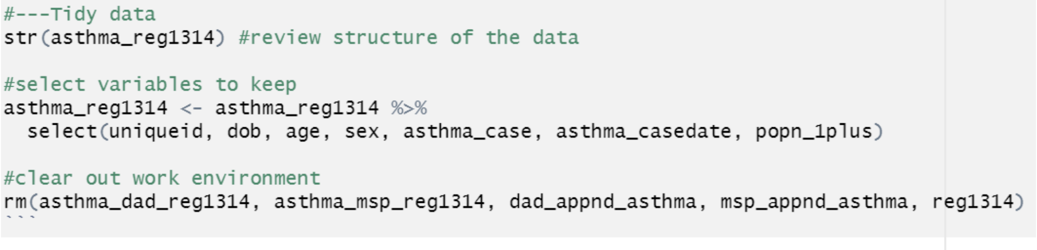


~~~
```{r Process when case met definition}

#identify case date, or the earliest point at which a person met the case definition
asthma_reg1314 <- asthma_reg1314 %>%
  mutate(date_flag = case_when(
    asthma_case ==1 & dad_sepdat_asthma < msp_servdat_asthma ~ 1,
    asthma_case ==1 & dad_sepdat_asthma > msp_servdat_asthma ~ 2,
    asthma_case ==1 & dad_sepdat_asthma == msp_servdat_asthma ~ 3,
    asthma_case ==1 & is.na(dad_sepdat_asthma) == TRUE ~ 4,
    asthma_case ==1 & is.na(msp_servdat_asthma) == TRUE ~ 5,
    TRUE~0))


asthma_reg1314 <- asthma_reg1314 %>%
  mutate(asthma_casedate = case_when(
    date_flag == 1 ~ dad_sepdat_asthma,
    date_flag == 2 ~ msp_servdat_asthma,
    date_flag == 3 ~ msp_servdat_asthma,
    date_flag == 4 ~ msp_servdat_asthma,
    date_flag == 5 ~ dad_sepdat_asthma))


#count asthma case flags
asthma_reg1314 %>%
  count(asthma_case)

#count date flags
asthma_reg1314 %>%
  count(date_flag)

#compare case and date flags
asthma_reg1314 %>%
  count(asthma_case, date_flag)

#compare casedate and case flags
asthma_reg1314 %>%
  count(asthma_casedate, date_flag)
~~~

</td>
</tr>
<tr>
<td>

Does the count of asthma cases count reflect incident cases? Why or why not?

</td>
<td></td>
</tr>
</table>

| | |
|---|---|
| Tidy the final product and workspace as needed.<br>Keep only the variables you need for later: uniqueid, dob, age, sex, asthma_case, asthma_casedate, popn_1plus | <br>`#---Tidy data`<br>`str(asthma_reg1314) #review structure of the data`<br><br>`#select variables to keep`<br>`asthma_reg1314 <- asthma_reg1314 %>%`<br>`  select(uniqueid, dob, age, sex, asthma_case, asthma_casedate, popn_1plus)`<br><br>`#clear out work environment`<br>`rm(asthma_dad_reg1314, asthma_msp_reg1314, dad_appnd_asthma, msp_appnd_asthma, reg1314)`<br>```` ``` ```` |

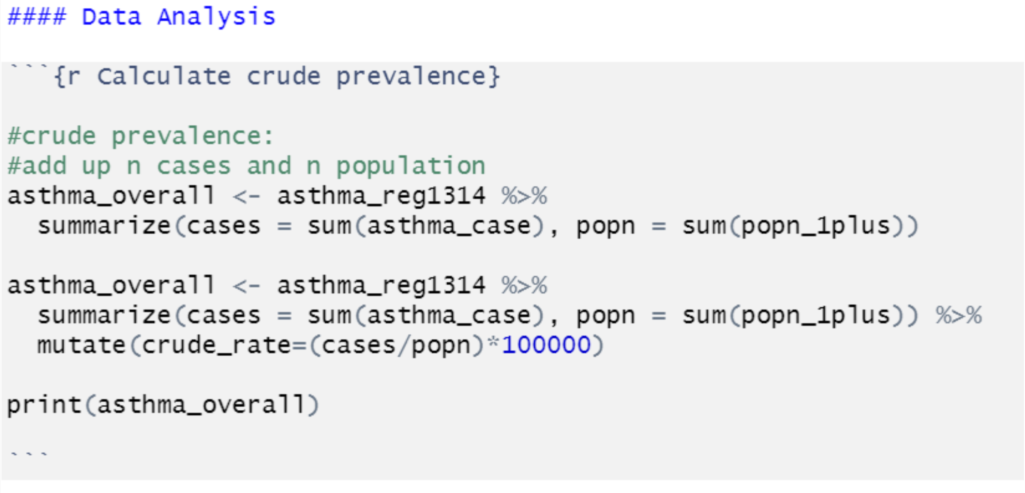
## 4. Analyze and visualize the data

<table>
<tr><th>Task</th><th>Code</th></tr>
<tr><td>

What was the crude asthma prevalence rate in 2013/14?

Note – code duplicates in this block for demonstrative purposes related to chaining statements together.

</td><td>

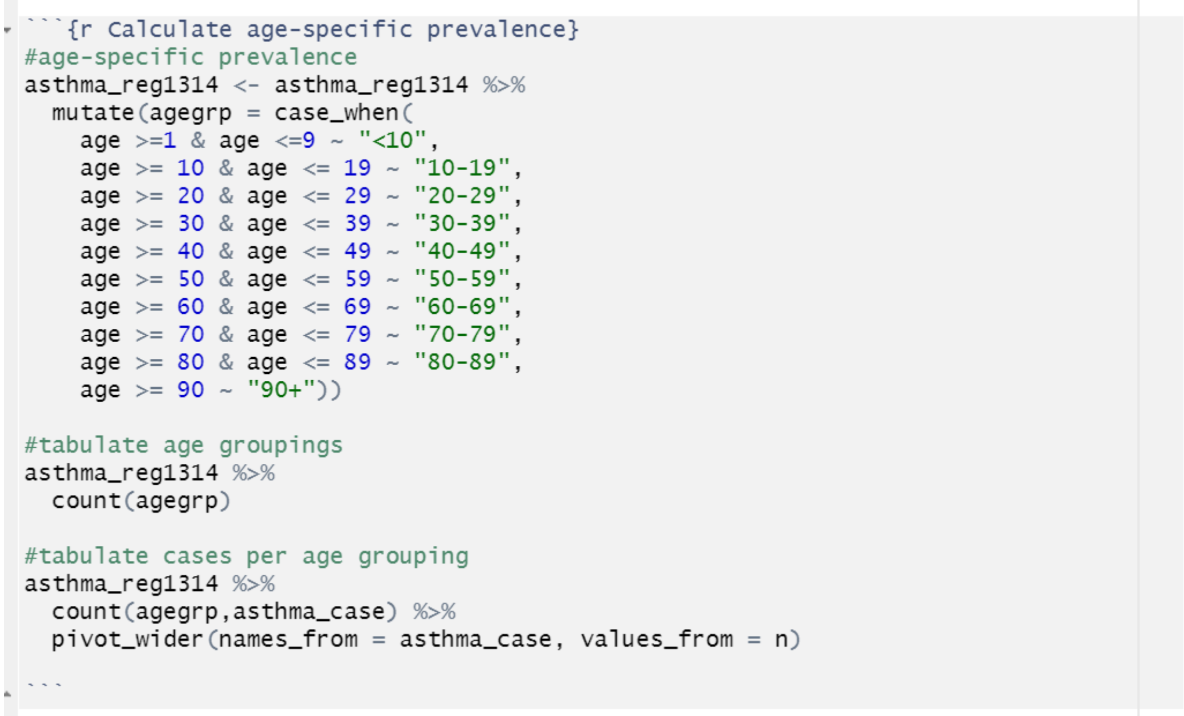


~~~
#### Data Analysis

```{r Calculate crude prevalence}

#crude prevalence:
#add up n cases and n population
asthma_overall <- asthma_reg1314 %>%
  summarize(cases = sum(asthma_case), popn = sum(popn_1plus))

asthma_overall <- asthma_reg1314 %>%
  summarize(cases = sum(asthma_case), popn = sum(popn_1plus)) %>%
  mutate(crude_rate=(cases/popn)*100000)

print(asthma_overall)

```
~~~

</td></tr>
</table>

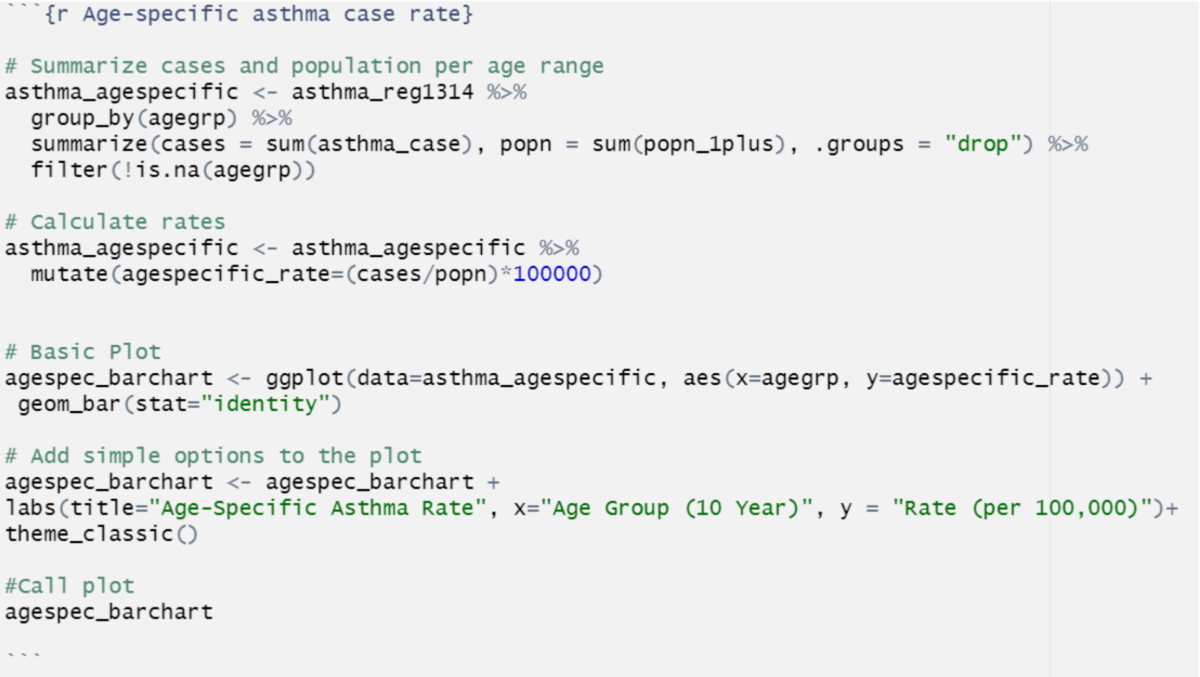
Calculate age-specific counts of prevalent asthma cases by first creating age categorisations and tabulating cases by those categories. What age groups will you use?

We will calculate rates and create a figure in the next step.

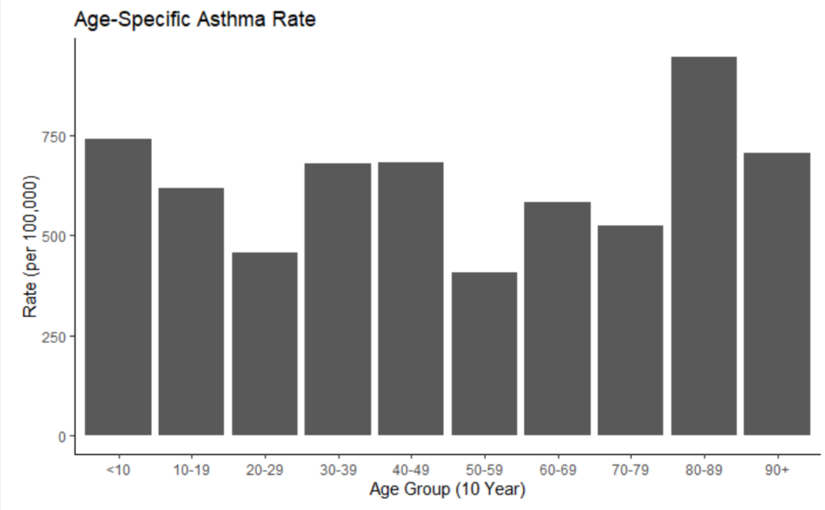


~~~
```{r Calculate age-specific prevalence}
#age-specific prevalence
asthma_reg1314 <- asthma_reg1314 %>%
  mutate(agegrp = case_when(
    age >=1 & age <=9 ~ "<10",
    age >= 10 & age <= 19 ~ "10-19",
    age >= 20 & age <= 29 ~ "20-29",
    age >= 30 & age <= 39 ~ "30-39",
    age >= 40 & age <= 49 ~ "40-49",
    age >= 50 & age <= 59 ~ "50-59",
    age >= 60 & age <= 69 ~ "60-69",
    age >= 70 & age <= 79 ~ "70-79",
    age >= 80 & age <= 89 ~ "80-89",
    age >= 90 ~ "90+"))

#tabulate age groupings
asthma_reg1314 %>%
  count(agegrp)

#tabulate cases per age grouping
asthma_reg1314 %>%
  count(agegrp,asthma_case) %>%
  pivot_wider(names_from = asthma_case, values_from = n)

```
~~~

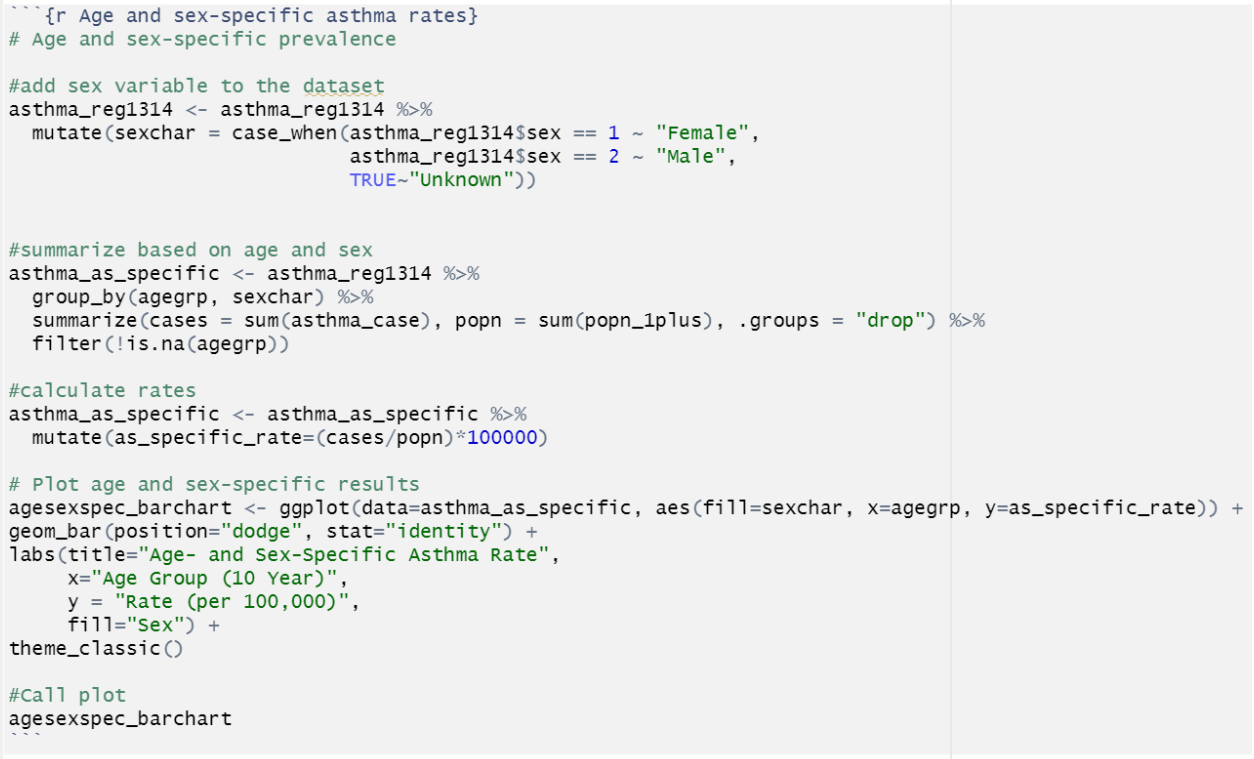
Calculate and plot age-specific asthma rates using the same 10 year age groups you created in the previous step..



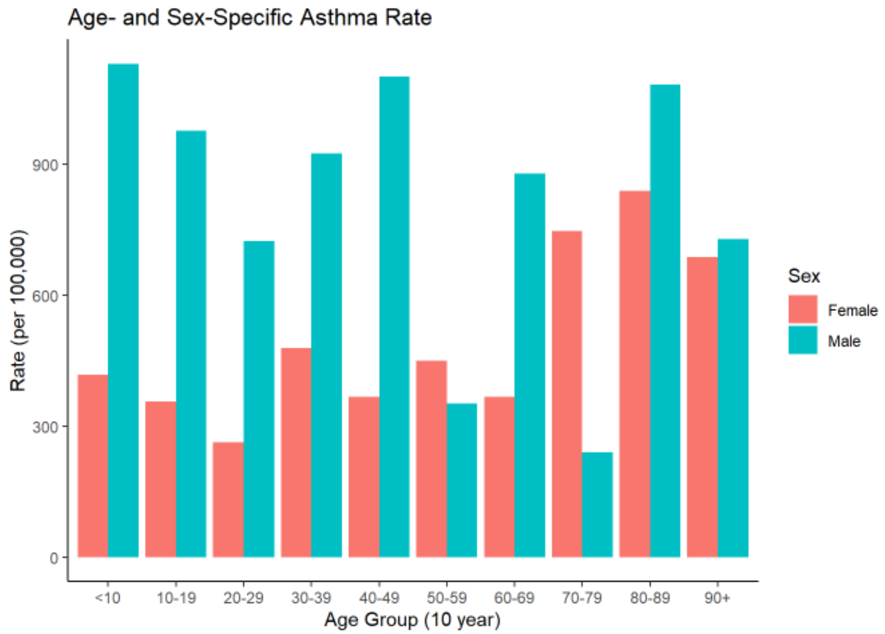
~~~
```{r Age-specific asthma case rate}

# Summarize cases and population per age range
asthma_agespecific <- asthma_reg1314 %>%
  group_by(agegrp) %>%
  summarize(cases = sum(asthma_case), popn = sum(popn_1plus), .groups = "drop") %>%
  filter(!is.na(agegrp))

# Calculate rates
asthma_agespecific <- asthma_agespecific %>%
  mutate(agespecific_rate=(cases/popn)*100000)


# Basic Plot
agespec_barchart <- ggplot(data=asthma_agespecific, aes(x=agegrp, y=agespecific_rate)) +
 geom_bar(stat="identity")

# Add simple options to the plot
agespec_barchart <- agespec_barchart +
labs(title="Age-Specific Asthma Rate", x="Age Group (10 Year)", y = "Rate (per 100,000)")+
theme_classic()

#Call plot
agespec_barchart

```
~~~

| | |
|---|---|
| |  Age-Specific Asthma Rate<br>Rate (per 100,000)<br>0<br>250<br>500<br>750<br><10<br>10-19<br>20-29<br>30-39<br>40-49<br>50-59<br>60-69<br>70-79<br>80-89<br>90+<br>Age Group (10 Year) |

Calculate and plot age- and sex-specific asthma rates.

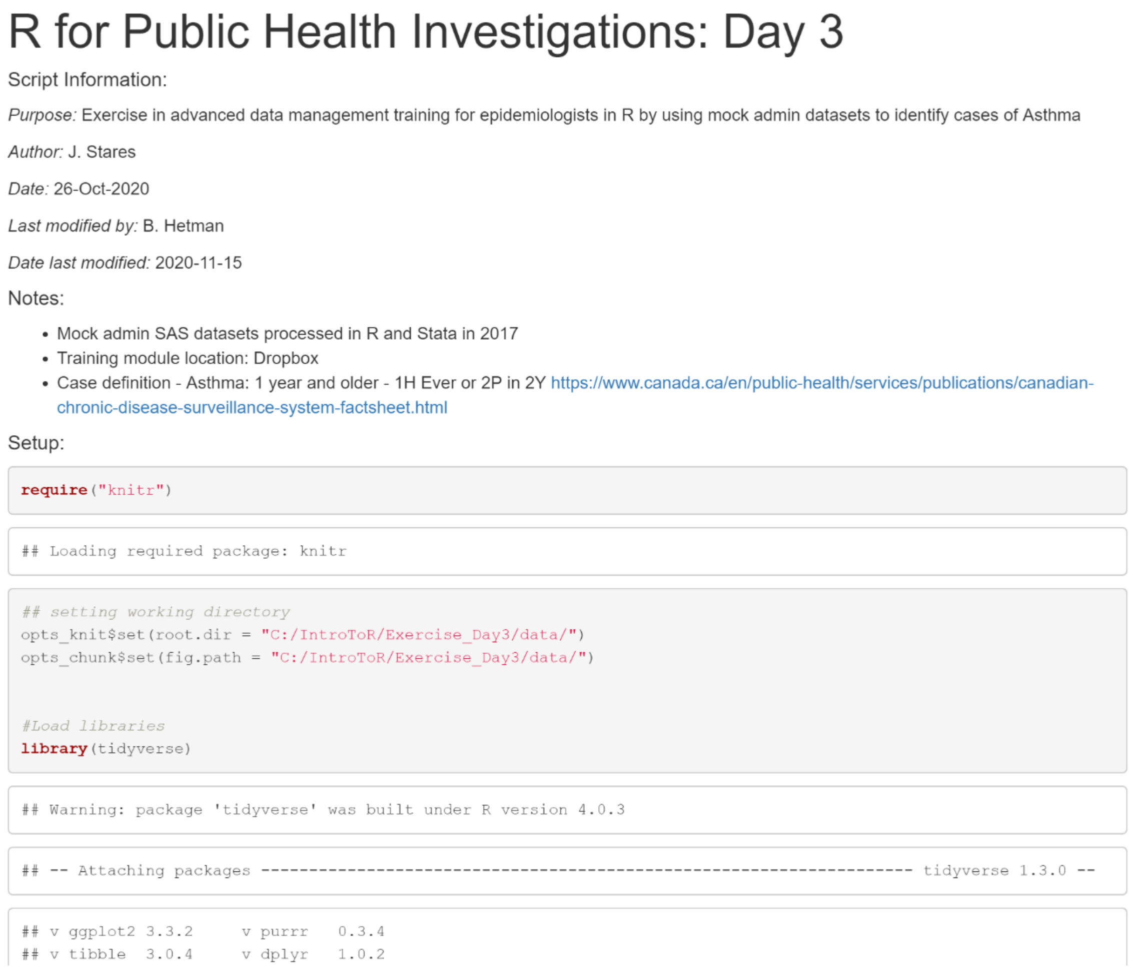


~~~
```{r Age and sex-specific asthma rates}
# Age and sex-specific prevalence

#add sex variable to the dataset
asthma_reg1314 <- asthma_reg1314 %>%
  mutate(sexchar = case_when(asthma_reg1314$sex == 1 ~ "Female",
                             asthma_reg1314$sex == 2 ~ "Male",
                             TRUE~"Unknown"))


#summarize based on age and sex
asthma_as_specific <- asthma_reg1314 %>%
  group_by(agegrp, sexchar) %>%
  summarize(cases = sum(asthma_case), popn = sum(popn_1plus), .groups = "drop") %>%
  filter(!is.na(agegrp))

#calculate rates
asthma_as_specific <- asthma_as_specific %>%
  mutate(as_specific_rate=(cases/popn)*100000)

# Plot age and sex-specific results
agesexspec_barchart <- ggplot(data=asthma_as_specific, aes(fill=sexchar, x=agegrp, y=as_specific_rate)) +
geom_bar(position="dodge", stat="identity") +
labs(title="Age- and Sex-Specific Asthma Rate",
     x="Age Group (10 Year)",
     y = "Rate (per 100,000)",
     fill="Sex") +
theme_classic()

#Call plot
agesexspec_barchart
```
~~~

| | |
|---|---|
| |  |
| Ensure pertinent details are captured in the script heading. | |
| Knit the document<br><br>Note: The document will be saved in your scripts folder or the folder you specified at the beginning of the markdown. | |

## 5. Example output

This is an example of what your output will look like when you open it in your web-browser:



# R for Public Health Investigations: Day 3

Script Information:

*Purpose:* Exercise in advanced data management training for epidemiologists in R by using mock admin datasets to identify cases of Asthma

*Author:* J. Stares

*Date:* 26-Oct-2020

*Last modified by:* B. Hetman

*Date last modified:* 2020-11-15

Notes:

- Mock admin SAS datasets processed in R and Stata in 2017
- Training module location: Dropbox
- Case definition - Asthma: 1 year and older - 1H Ever or 2P in 2Y https://www.canada.ca/en/public-health/services/publications/canadian-chronic-disease-surveillance-system-factsheet.html

Setup:

```
require("knitr")
```

```
## Loading required package: knitr
```

```
## setting working directory
opts_knit$set(root.dir = "C:/IntroToR/Exercise_Day3/data/")
opts_chunk$set(fig.path = "C:/IntroToR/Exercise_Day3/data/")


#Load libraries
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## -- Attaching packages -------------------------------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
```

## 6. Critical thinking

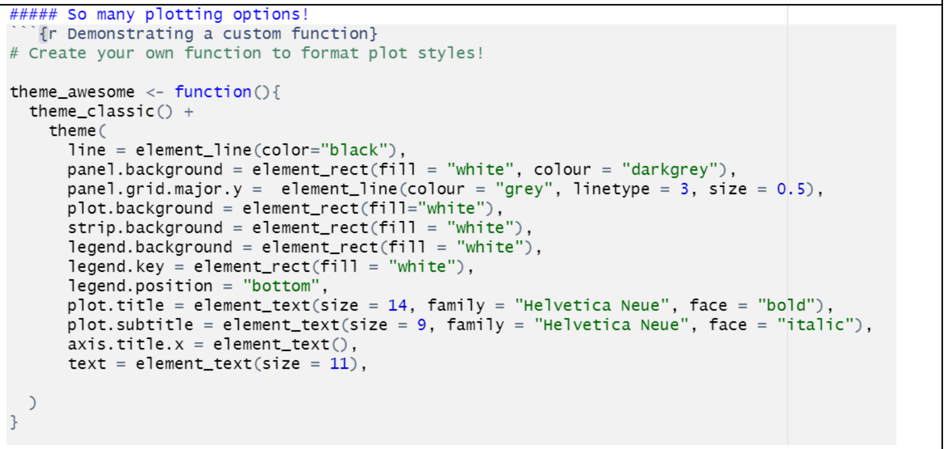
1. Why did we build the final product as a registry table that flags numerators and denominators for asthma? Is there a better way?

2. What are possible limitations to taking the approach of using administrative health records to identify cases of disease? What are the strengths with this approach?

3. What are your thoughts on how this exercise was structured in comparison with exercises from Day 1 and Day 2 (i.e., several short scripts vs. one long one, chunks of code vs. code broken into lines, and working directory structure)?

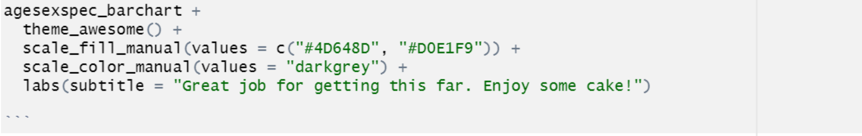
## 7. Bonus task 1!

Bonus advanced content! Demonstration of user defined functions:

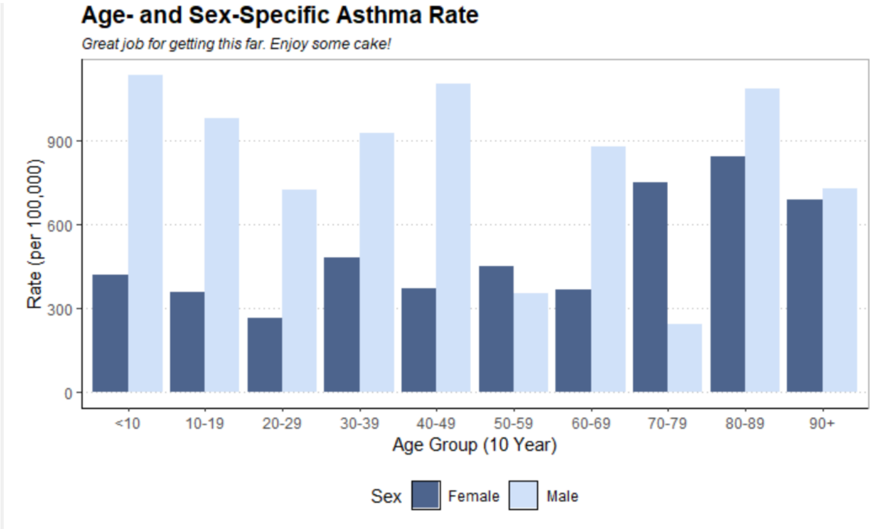
| Task | Code |
|---|---|
| Review any of the following material on Functions in R:<br>https://www.tutorialspoint.com/r/r_functions.htm<br>https://towardsdatascience.com/your-first-user-defined-function-in-r-1eedc634ead4<br>https://swcarpentry.github.io/r-novice-inflammation/02-func-R/<br>https://datasciencebeginners.com/2018/11/02/10-user-defined-functions-in-r/<br>Examine and run the demo code provided here to your rmd file.<br>What are the basic components of a user defined function? Which part of the code defines the name of the function? Are there any arguments input to this function? What does this function do? |  |

~~~
##### So many plotting options!
```{r Demonstrating a custom function}
# Create your own function to format plot styles!

theme_awesome <- function(){
  theme_classic() +
    theme(
      line = element_line(color="black"),
      panel.background = element_rect(fill = "white", colour = "darkgrey"),
      panel.grid.major.y =  element_line(colour = "grey", linetype = 3, size = 0.5),
      plot.background = element_rect(fill="white"),
      strip.background = element_rect(fill = "white"),
      legend.background = element_rect(fill = "white"),
      legend.key = element_rect(fill = "white"),
      legend.position = "bottom",
      plot.title = element_text(size = 14, family = "Helvetica Neue", face = "bold"),
      plot.subtitle = element_text(size = 9, family = "Helvetica Neue", face = "italic"),
      axis.title.x = element_text(),
      text = element_text(size = 11),

  )
}
~~~

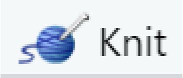
Apply your custom function to the agesexspec_barchart object created earlier.



~~~
agesexspec_barchart +
  theme_awesome() +
  scale_fill_manual(values = c("#4D648D", "#D0E1F9")) +
  scale_color_manual(values = "darkgrey") +
  labs(subtitle = "Great job for getting this far. Enjoy some cake!")

```
~~~



Knit the document



## 7. Bonus task 2!

Bonus advanced content! Better living through modern technology - demonstration of facets and loops:

<table>
<tr><th>Task</th><th>Code</th></tr>
<tr>
<td>

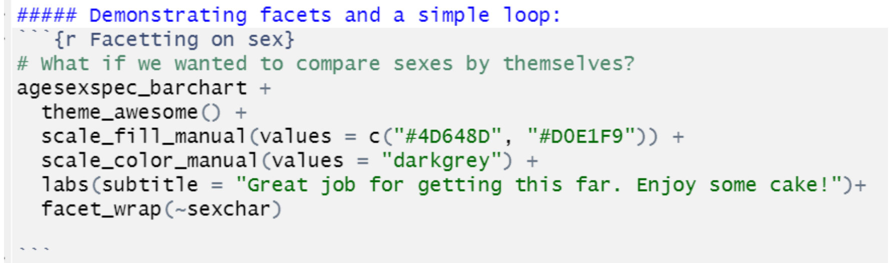
Review the following tutorial on faceting for ggplot in R.

http://www.sthda.com/english/wiki/ggplot2-facet-split-a-plot-into-a-matrix-of-panels

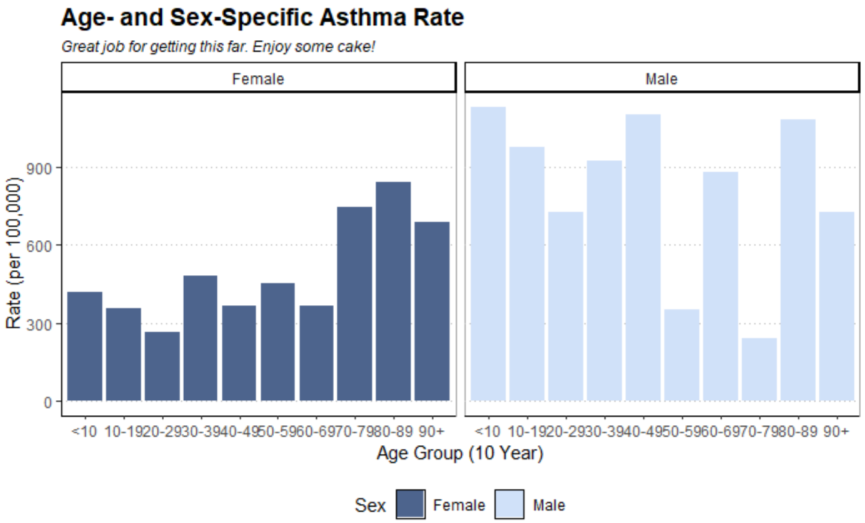
Examine and add the demo code provided here to your rmd file.

What is a facet? What does this facet do specifically? How are facets applicable to your work?

</td>
<td>



~~~
##### Demonstrating facets and a simple loop:
```{r Facetting on sex}
# What if we wanted to compare sexes by themselves?
agesexspec_barchart +
  theme_awesome() +
  scale_fill_manual(values = c("#4D648D", "#D0E1F9")) +
  scale_color_manual(values = "darkgrey") +
  labs(subtitle = "Great job for getting this far. Enjoy some cake!")+
  facet_wrap(~sexchar)

```
~~~



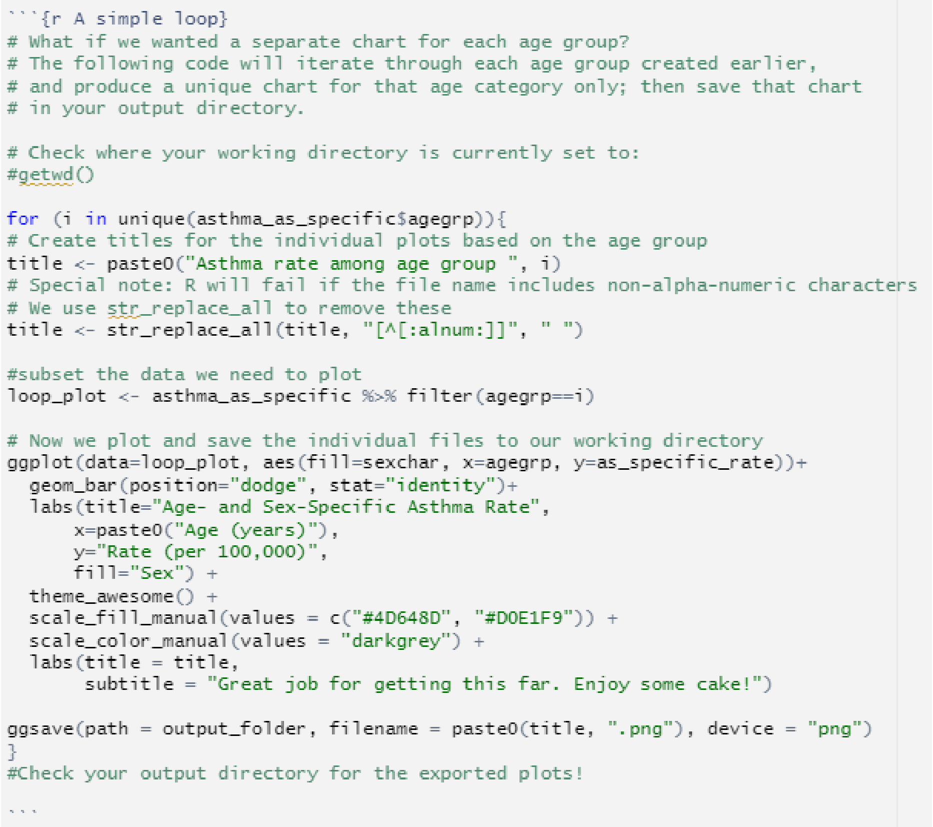
</td>
</tr>
</table>

Review the following tutorial on For Loops in R.

https://www.r-bloggers.com/2015/12/how-to-write-the-first-for-loop-in-r/

Examine and run the demo code provided here.

What are the basic components of a for loop? What is the purpose of the “i in unique” statement? What does the code block do specifically? What additional functionality does the loop provide? How are loops applicable to your work?



~~~
```{r A simple loop}
# What if we wanted a separate chart for each age group?
# The following code will iterate through each age group created earlier,
# and produce a unique chart for that age category only; then save that chart
# in your output directory.

# Check where your working directory is currently set to:
#getwd()

for (i in unique(asthma_as_specific$agegrp)){
# Create titles for the individual plots based on the age group
title <- paste0("Asthma rate among age group ", i)
# Special note: R will fail if the file name includes non-alpha-numeric characters
# We use str_replace_all to remove these
title <- str_replace_all(title, "[^[:alnum:]]", " ")

#subset the data we need to plot
loop_plot <- asthma_as_specific %>% filter(agegrp==i)

# Now we plot and save the individual files to our working directory
ggplot(data=loop_plot, aes(fill=sexchar, x=agegrp, y=as_specific_rate))+
  geom_bar(position="dodge", stat="identity")+
  labs(title="Age- and Sex-Specific Asthma Rate",
      x=paste0("Age (years)"),
      y="Rate (per 100,000)",
      fill="Sex") +
  theme_awesome() +
  scale_fill_manual(values = c("#4D648D", "#D0E1F9")) +
  scale_color_manual(values = "darkgrey") +
  labs(title = title,
       subtitle = "Great job for getting this far. Enjoy some cake!")

ggsave(path = output_folder, filename = paste0(title, ".png"), device = "png")
}
#Check your output directory for the exported plots!

```
~~~

Knit the document