Merging Australian postcodes and Federal election results

Rebecca Linigen

31 August 2018

Summary

In order to explore research question 3 'Does political preference have an impact on childhood immunisation coverage rates?'. Two datasets Postcodes by Electoral Division - Australia Parliament House & 2016 Federal Election results by electorate - Australian Electoral Commission need to be merged to create a singular datset with the key joining variable of **Postcode**, in order to merged with immunisation and demographic data that exists at a postcode level, and therefore conduct EDA to test the hypothesis that a relationship may exist between political preference and likelihood to immunise children.

Importing Dataset 1 - Australian Parliament House - Postcodes by Electoral Division

Load the `tidyverse` and `readxl` libraries;

```
library(tidyverse)
library(readxl)
```

Import the Data file using `read_excel`

```
APH_post_elec <- read_excel("Postcode by Electorate all AUS - 
fixed.xlsx")

glimpse(APH_post_elec)

## Observations: 3,035
## Variables: 3
## $ Postcode             <chr> "0800", "0810", "0812", "0820", "0822",
"...
## $ `Electoral division` <chr> "Solomon", "Solomon", "Solomon",
"Solomon...
## $ `Per cent`           <dbl> 100.0, 100.0, 100.0, 100.0, 96.1, 3.9,
10...
```

Using `glimpse` above, I can see that the number formats are readable,there are no characters surrounding the numbers, no leading zeroes have been dropped, and therefore no parsing is required at this stage.

Inspecting the contents of each dataset, it becomes clear that the electorate is the single variable that can be used to connect the two datasets, as shown below.

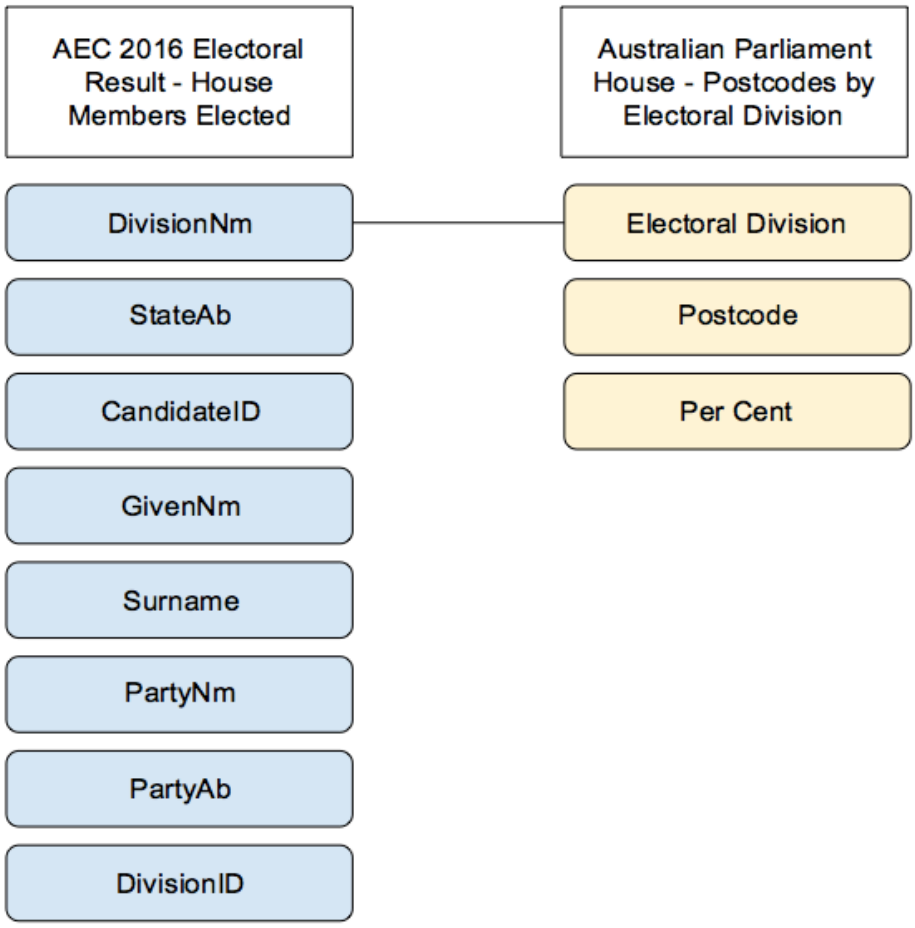

Fig.1. Relationship between variables in two Datasets

Importing Dataset 2 - Australian Electoral Commission - 2016 Federal Election results by electorate

```
AEC_2016_fed <- read_csv("HouseMembersElectedDownload-20499.csv")

## Parsed with column specification:
## cols(
##   `2016 Federal Election House of Representatives Members Elected
[Event:20499 Phase:FinalResults Generated:2017-05-11T17:34:37
Cycle:a45ea4bc-bff6-48cb-94c5-1bbaae0c7c0d Created:2017-05-11T17:29:15
Environment:PROD Site:CANBERRA Server:TALLYROOM Version:10.5.84.43192]`
= col_character()
## )
```

This produced a number of `parsing` and `rbind` warnings, let's `glimpse` the data to see if we can figure out what's going on.

```
glimpse(AEC_2016_fed)
```

```
## Observations: 151
## Variables: 1
## $ `2016 Federal Election House of Representatives Members Elected
[Event:20499 Phase:FinalResults Generated:2017-05-11T17:34:37
Cycle:a45ea4bc-bff6-48cb-94c5-1bbaae0c7c0d Created:2017-05-11T17:29:15
Environment:PROD Site:CANBERRA Server:TALLYROOM Version:10.5.84.43192]`
<chr> ...
```

The above output alerts us to the fact that the Header column in the csv is a character description of the data. We need to remove this using `skip`.

```
AEC_2016_fed <- read.csv(file = "HouseMembersElectedDownload-
20499.csv", skip = 1, head = TRUE)

glimpse(AEC_2016_fed)
```

```
## Observations: 150
## Variables: 8
## $ DivisionID  <int> 179, 197, 198, 103, 180, 104, 192, 199, 200,
105, ...
## $ DivisionNm  <fct> Adelaide, Aston, Ballarat, Banks, Barker,
Barton, ...
## $ StateAb     <fct> SA, VIC, VIC, NSW, SA, NSW, TAS, VIC, VIC, NSW,
NS...
## $ CandidateID <int> 29073, 28875, 28317, 28660, 29568, 28964, 28451,
2...
## $ GivenNm     <fct> Kate, Alan, Catherine, David, Tony, Linda, Ross,
D...
## $ Surname     <fct> ELLIS, TUDGE, KING, COLEMAN, PASIN, BURNEY,
HART, ...
## $ PartyNm     <fct> Australian Labor Party, Liberal, Australian
Labor ...
## $ PartyAb     <fct> ALP, LP, ALP, LP, LP, ALP, ALP, ALP, ALP, LP,
LP, ...
```

Now to merge the two datasets together. We know from using View on the data, that Electorate (*'Electoral division'* in **APH_post_elec** and *'DivisionNm'* in **AEC_2016_fed**) is our joining variable, but there are multiples observations of Electorate in **APH_post_elec** as there can be multiples postcodes in each electorate, let's use `count` to confirm that;

```
APH_post_elec %>% count(`Electoral division`) %>% filter(n>1)
```

```
## # A tibble: 150 x 2
##    `Electoral division`     n
##    <chr>                <int>
##  1 Adelaide                19
##  2 Aston                    9
##  3 Ballarat                17
```

```
##  4 Banks                     9
##  5 Barker                   63
##  6 Barton                   10
##  7 Bass                     20
##  8 Batman                    7
##  9 Bendigo                  24
## 10 Bennelong                 8
## # ... with 140 more rows
```

```
AEC_2016_fed %>% count(DivisionNm) %>% filter(n>1)
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: DivisionNm <fct>, n <int>
```

I suspect that each observation in **AEC_2016_fed** dataset is unique, therefore no results are returned. Let's try changing `n>1` to `N==1` and see what happens;

```
AEC_2016_fed %>% count(DivisionNm) %>% filter(n==1)
```

```
## # A tibble: 150 x 2
##    DivisionNm     n
##    <fct>      <int>
##  1 Adelaide       1
##  2 Aston          1
##  3 Ballarat       1
##  4 Banks          1
##  5 Barker         1
##  6 Barton         1
##  7 Bass           1
##  8 Batman         1
##  9 Bendigo        1
## 10 Bennelong      1
## # ... with 140 more rows
```

Now that we have the two datasets tidy, and we've sense checked the counts on the joining variable, we can combine the datasets using with a Mutating join using `leftjoin`.

```
elec_result_PC <- APH_post_elec %>% left_join(AEC_2016_fed, by =
"DivisionNm")
```

```
## Error: `by` can't contain join column `DivisionNm` which is missing
from LHS
```

This error was caused by a mismatch in column names. In order to join variables with different names, we need to apply a named vector in the format `c("a" = "b")` as follows;

```
elec_result_PC <- APH_post_elec %>% left_join(AEC_2016_fed, by =
c("Electoral division" = "DivisionNm"))
```

```
glimpse(elec_result_PC)
```

```
## Observations: 3,035
## Variables: 10
## $ Postcode             <chr> "0800", "0810", "0812", "0820", "0822",
"...
## $ `Electoral division` <chr> "Solomon", "Solomon", "Solomon",
"Solomon...
## $ `Per cent`           <dbl> 100.0, 100.0, 100.0, 100.0, 96.1, 3.9,
10...
## $ DivisionID           <int> 307, 307, 307, 307, 306, 307, 307, 307,
3...
## $ StateAb              <fct> NT, NT, NT, NT, NT, NT, NT, NT, NT, NT,
N...
## $ CandidateID          <int> 28737, 28737, 28737, 28737, 28735,
28737,...
## $ GivenNm              <fct> Luke, Luke, Luke, Luke, Warren, Luke,
Luk...
## $ Surname              <fct> GOSLING, GOSLING, GOSLING, GOSLING,
SNOWD...
## $ PartyNm              <fct> Australian Labor Party, Australian
Labor ...
## $ PartyAb              <fct> ALP, ALP, ALP, ALP, ALP, ALP, ALP, ALP,
A...
```

This combined dataset **elec_results_PC** is now ready to be merged with the immunisation dataset and demographic dataset - postcode has been identiifed as the joining variable across files.

After reviewing the other data files & discussing the regression model we plan on using, it was determined that we need to train the model on data from previous years. To this end, we decided to review data from the 2010 and 2013 federal elections, in addition to the 2016 federal elections results already merged above in **elec_results_PC**.

Repeat steps above to import the 2010 and 2013 election reuslt datasets.

Rename all headers so the different years are identifiable within the file (as all files are from the AEC, the filename conventions are the same in each file, therefore they need to be renamed to identify each years results within the merged file. I have shown one example below using the **AEC_2010_fed dataset**.

```
colnames(AEC_2010_fed)[colnames(AEC_2010_fed)=="DivisionNm"] <-
"DivisionNm2010"
```

After renaming all column headers to the desired names, I merge the existing dataset **elec_results_PC**, with the **AEC_2010_fed** datafile we just imported, usiing *"Electoral Division"* as the joining variable.

```
elec_result_PC_2016_2010 <- elec_result_PC %>% left_join(AEC_2010_fed,
by = c("Electoral division" = "DivisionNm2010"))

View(elec_result_PC_2016_2010)
```

I then repeat the same steps to join **AEC_2013_fed** to the merged 2010, 2016 and postcode level data saving as object **elect_reult_PC_all**.

```
elec_result_PC_all <- elec_result_PC_2016_2010 %>%
left_join(AEC_2013_fed, by = c("Electoral division" =
"DivisionNm2013"))

View(elec_result_PC_all)
```

This election result by postcode data is now ready to be merged to the PHN and demographic datasets using *Postcode* as the joiining variable.