The communication barriers that hamper our emergency response

Fiona Neilson | October 2020

Abstract: Victoria, Australia is a multicultural state, and yet recent emergencies have shown that important safety messages are failing to reach culturally and linguistically diverse (CALD) communities. This report creates English language and literacy profiles of these communities and shows which groups are the most disadvantaged and advantaged in terms of their ability to receive communications. Australian Bureau of Statistics (ABS) Census data for 2016 was used for one Local Government Area (LGA), the City of Greater Dandenong in southeast Melbourne (see map, Appendix B), and various socioeconomic indicators were combined with languages spoken to generate detailed profiles of cultural groups. Migrants who speak a language other than English at home were selected, and language has been used as a proxy for culture and ethnicity as it is more accurate than country of birth. The report finds that educational attainment is the strongest predictor of English proficiency, and the most disadvantaged group in terms of communication skills and connections to Australian institutions is women from Southeast Asia and Southern Europe who emigrated prior to 2006. Fortunately, there is a much larger section of the cohort that has much better literacy and social participation, across nearly all language groups. The analysis shows that the composition of each community group varies, and that communication strategies must be tailored to each group's profile. It provides a simple method for quickly creating a clear and informative profile of any ethnic community group, which could help councils and government put more successful communication plans into action.

Introduction: The past year has seen unprecedented disaster and emergency for the state of Victoria. Communicating with Victorians to warn them of danger is critical. While it is the responsibility of the government to provide warnings and manage response, it is clear that members of culturally and linguistically diverse (CALD) communities face communication barriers. The bushfires of 2019-2020 and the current pandemic have highlighted the need for government to partner with communities who cannot be easily reached by standard channels such as English-language mobile applications (Lazzaro, 2020), printed leaflets in local languages (Baker, 2020) or mainstream news media.

At one point in June 2020, all of the Victorian COVID-19 hotspots had 'above state-average diversity levels', with significant numbers who spoke a language other than English at home (Baker and Lee, 2020). While the Chief Health Officer acknowledged the overrepresentation of culturally and linguistically diverse communities amongst these cases (Grey, 2020) and the Victorian government undertook to improve communication with these groups, ethnic group representatives criticised the inaccessibility of translated materials and the lack of coverage of the over 300 languages spoken in Australia (Fryer, 2020). Some CALD community groups attempted to fill the gap by creating their own unofficial information, which risks being inaccurate, particularly when it comes to complex public health advice (Baker, 2020).

The National COVID-19 Health and Research Advisory Committee (NHRAC) report (2020) includes CALD communities amongst 'vulnerable and hard to reach groups' and emphasises the importance of 'maintaining clear communications on the status and reasons for public health measures' (NHRAC, p.2). The report finds that community representatives are best-placed to engage with at-risk groups (ibid, p.3). Wild et al (2020) emphasise that messages need to be tailored to match cultural context and values, and that trusted messengers and accessible channels are critical. The reception of these complex messages is also affected by a person's educational level and literacy in any language, not just English. A Community Four (2020) initiative saw Afghan Hazara women record translated pandemic information into voicemail messages and share them with families via Viber (a mobile phone messaging application), illustrating how tailored such approaches may need to be.

This investigation profiles the range of language groups in the highly multicultural Local Government Area (LGA) of the City of Greater Dandenong, where I live. The City of Greater Dandenong is home to a diverse range of migrants, particularly those from refugee backgrounds (City of Greater Dandenong Council, 2017). This study examines the relationship between English proficiency, language spoken and several socioeconomic indicators to identify migrant groups who may miss out on important English-language

health and safety advice. The aim is to show how many such groups there are within one local government area, so that state and local authorities can develop more targeted communication plans, and evaluate the extent to which they must partner with community groups.

Data: The data show migrants living in the City of Greater Dandenong at the time of the 2016 Census who speak a language other than English at home. Only people born outside Australia are included, as they have not had the same opportunity as Australian-born people to experience in-country cultural immersion and English language acquisition from birth. The 57,825 people in the dataset represent **36%** of the total population of 160,952 residents of Dandenong on the Census date (City of Greater Dandenong Council, 2018).

Data are from the *ABS 2016 Census of Population and Housing - Counting Persons, Place of Usual Residence* dataset. It was downloaded using the Census TableBuilder, which makes all variables available for selection into custom tables using the ABS Table Builder Pro online tool (Australian Bureau of Statistics, 2020). The Census is a national demographic survey run every 5 years and managed by the ABS. An initial report was generated from ABS data to list all languages other than English spoken in the LGA. All 151 languages on the list were then selected and downloaded individually as Comma Separated Value (CSV) files, each using a template I had created in Table Builder.

A range of indicators relating to person characteristics, education and qualifications, employment, cultural and language diversity, and dwelling characteristics were selected. The language other than English spoken at home was the filter criterion used. Languages were selected at the ABS 4-digit code level, as this is the most accurate. Table 1 lists these variables.

Table 1. *List of variables used in this analysis*

Variable	Type
Local Government Area (LGA)	Categorical nominal
Language Spoken at Home (LANP)	Categorical nominal
Proficiency in Spoken English (ENGP)	Categorical binary
Level of Highest Educational Attainment (HEAP)	Categorical ordinal
Dwelling Internet Connection (NEDD)	Categorical binary
Engagement in Employment, Education and Training (EETP)	Categorical ordinal
Sex (SEXP)	Categorical binary
Year of Arrival in Australia (ranges) (YARRP)	Categorical ordinal
Total (count of persons)	Numerical discrete
4 variables: Language group names and codes at ABS 1-digit and 2-digit levels	Categorical ordinal

Several pre-processing steps precede the ones described in this report:

a. *Variable selection* (by me). I selected all variables myself from ABS microdata to build custom tables for download. I did not use predefined data tables or data cubes.
b. *Creation of custom groups* (by me). The levels in four of the variables: English proficiency, education, employment/ training and year of arrival, were combined within their respective variable to create simpler categories and reduce the processing time by the ABS TableBuilder. For example, for English proficiency, categories *1 – Very well*, *2 – Well*, *3 - Not well* and *4 – Not at all* were combined into *Adequate* (1 and 2) and *Inadequate* (3 and 4).
c. *ABS creation of custom variables*. The ABS has derived the Employment and Training (EETP) variable from other variables. Educational attainment (HEAP) is a combination of two other variables (Australian Bureau of Statistics, 2016b).
d. I used the zero suppression function in Table Builder to remove observations with null values.
e. The ABS redacted data that could identify individuals. A number of downloaded tables were empty due to this, and were thus discarded.

Methods: Data science methods were used for collating the data, cleaning, pre-processing, exploratory visualisation and analysis. RStudio Team (2020) was used for all data processing and analysis unless stated otherwise.

1. Data representation - 112 tables (one per language) were uploaded to R using *read.csv*. A custom *tidy* function was created using commands *t, data.frame, colnames, remove_rownames* and *gather* in base R to transpose the data from wide to long format in each table, including variable names, and gather two columns into one (see below). Tables were joined into a master table using the *rbind* function. A reference table of ABS language classifications (ABS, 2016a) was added to the dataset using *left_join* from the **dplyr** package. While the downloaded data included the most detailed level of language and country code (the 4-digit level), the 2-digit and 1-digit codes and names were joined so that languages and countries could be grouped into higher-order groupings, eg language group.

2. Unstructured to Structured data – The raw data came in the form of 151 tables (CSV format). It was semi-structured in that it was a download from a database in a partially tabular format, but it needed formatting in order to be readable by R. The main feature of the downloaded tables was that many cells were empty: variable levels were printed hierarchically and the cells to the right in that row were left blank. I had to manually copy and paste values into large amounts of blank cells to make the data readable using Excel (see Appendix A for sample). I only filled in missing values in cells and did not undertake any pre-processing that could be done in R. I also formatted the ABS Languages classification into a basic readable format using Excel.

3. Data cleaning: Some data had one or two missing values due to errors in my preliminary formatting in Excel. *fct_recode* from **forcats** and *mutate* from **dplyr** were used to replace missing levels. These two functions were also used to clean up mismatches between Census data and reference data: for example, the language "Serbo-Croatian/Yugoslavian, so described" did not match with "Serbo-Croatian/Yugoslavian so described".

4. Type conversion – All variables were converted to factors using *as.factor*, except for total, which was converted to numeric with *as.numeric*. Education, Employment, Year of Arrival and Group Name 1DC were converted to ordered factors using *factor*.

6. Gathering/Spreading – Each table was in wide format and was transposed to long format using *t* before the tables were merged. The Gender variable (SEXP) was created by gathering two columns, Male and Female using *gather* from **tidyr**.

7. Data subset selection and/or subsampling – Rows with zero observations were discarded using the *filter* function from **dplyr**. Out of 5848 initial observations, 3696 were retained. Subsets for Advantage and Disadvantage were created using *filter* so that these groups could be investigated. A subset of counts greater than 30 was also created for both groups in vector format to remove very low counts from cluttering the graphs, using *subset* and *%in%* with ggplot functions. These two functions were also used to "bin" Advantage languages into four groups of ascending size for grid display.

8. Group-based data summarisation *Count* from **dplyr** was used extensively to calculate weighted counts, as the data was aggregated, and one observation did not equal one person. **Dplyr** functions s*elect, filter, mutate, summarise, group_by* and *arrange* were used extensively to explore the data and combine variables in different ways for exploration.

9. Variable selection and/or transformation – A Gender variable (SEXP) was created from Male and Female. Two levels of the Year of Arrival (YARRP) variable were merged into one.

10. Exploratory visualisation using ggplot2 – *ggplot* from **ggplot2** was used to create high-level heatmaps of the data to indicate areas for more detailed investigation. *Geom_point* was helpful for these scatterplots. Given that nearly all the variables were categorical, *geom_bar* proved a clearest way of showing interrelationships. *Stat = "identity"* was used to ensure weighted counts were used in bar charts, or *size="Count"* for scatterplots. The large number of levels for Language made simple graphs a challenge, however the *geom_treemap* proved useful for summarising this information, and faceted graphs (*facet_wrap* and *facet_grid,* **ggplot2**) proved a concise way of displaying levels or subsets of levels. **Ggpubr** was used for grid display. The **dplyr** functions were used extensively with pipes to select variables for plots.

Results and Discussion:

The linguistic profile of Greater Dandenong

Over 100 languages are spoken in the council areas by the 57,825 people in the dataset, making it culturally and linguistically diverse. More than 80% of residents have at least one overseas-born parent,

and in 2017, over 25% of asylum seekers in Victoria lived in Greater Dandenong (City of Greater Dandenong, 2017). With such a large spread of languages, the treemap graph provides a concise overview of the proportions (Figure 1).

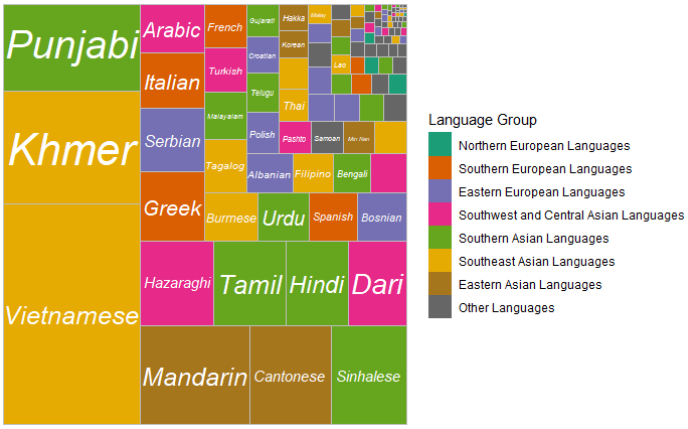


Figure 1. *Languages spoken in Greater Dandenong*

English proficiency

English proficiency was plotted with educational level by language group across the cohort as a proxy for general literacy levels (Figure 2). It shows clear correlations between higher levels of education (which may have attained been in any country) and adequate English proficiency. Eastern Asian and Southeast Asian language groups have the highest proportion of inadequate English, while the entire Northern European language group has adequate English.

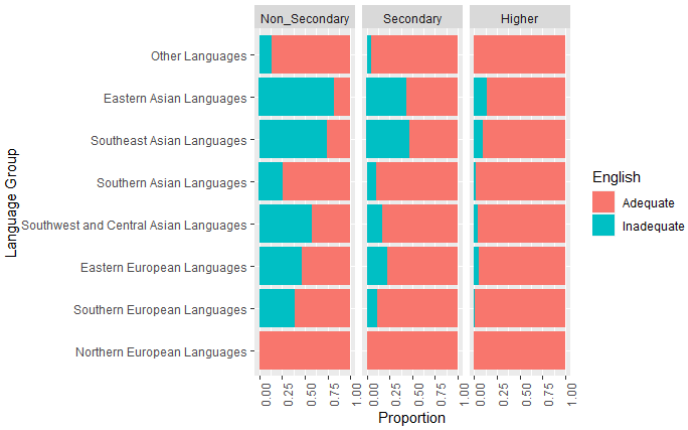


Figure 2. *Language, educational level and English proficiency*

A faceted scatterplot was then created to look in more detail at the Eastern Asian Languages group (Figure 3). While somewhat messy, this does confirm the relationship between adequate English proficiency and higher education. It also shows the differing education profiles within language groups: Japanese and Korean speakers are more highly educated, Tibetan speakers are the lowest educated group, and the rest have a fairly even distribution. There appears to be a slightly higher proportion of women than men with inadequate English.

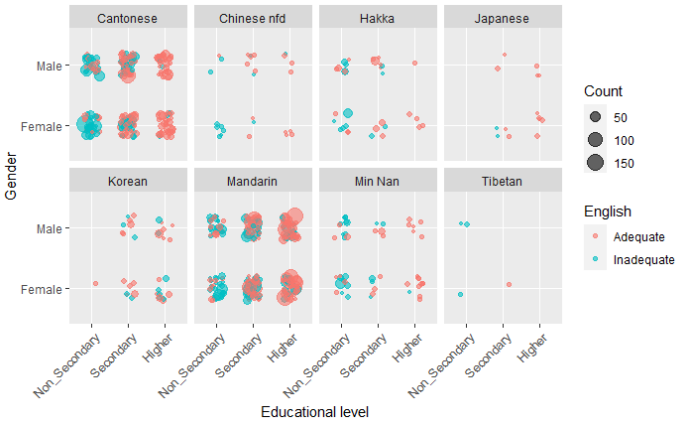


Figure 3. *Educational level, English and Gender by Language (Eastern Asian Languages)*

Disadvantaged groups

An alternative method was used to investigate who the most disadvantaged groups may be, and thus identify those most likely to be cut-off from mainstream information channels and social institutions such as place of work. A subset of the data was created of people with inadequate English, no internet access, incomplete secondary schooling and no participation in work or study. The data were then visualised to show language, gender and arrival period. Counts below 30 were not included. The data show that a clear majority of these people arrived before 2006, and that there are more women (**62%**) than men (**38%**). Southern European and Southeast Asian Languages dominate (Figure 4).

Figure 4. *Disadvantage by Gender, Language & Period of Arrival*

The bar chart presentation in Figure 4 allows us to make the connection more immediately between language and ethnic group: Italian and Greek

migrants from earlier waves of migration, Serbs and Croats displaced by the collapse of Yugoslavia, then Hazaras, Burmese and Rohingyas, more recently displaced by military juntas and persecution. Importantly, this graph shows clearly the prevalence of females and pre-2006 arrivals in the disadvantaged groups, suggesting that older women in these communities are the most removed from institutions that share emergency advice. Their lack of internet connectivity, which includes use of a mobile or smartphone, also removes them from local news or information in community languages, apart from television broadcasts or radio. Note however that these women may have Australian-born children and grandchildren who connect them to Australian society. Disadvantaged people comprise **2%** of the total cohort.

Advantaged groups

A similar approach can be taken to identify which groups have the most advantage, to understand which groups are most likely to receive communications. A subset of the data was created of people with adequate English, higher education, full participiation in work or study, and internet access. Data were then visualised to show language, gender and arrival period (Figure 5). Men comprise **60%** and women **40%** of the Advantaged group, although the proportions vary for individual languages, with women surpassing Thai men by a large margin. Some language groups, such as Southeast Asian, Eastern Asian, Southern European and Eastern European, see much more parity between men and women compared with Southern Asian, Southwest and Central Asian, where men outweigh women 2:1. Most groups saw arrivals across both periods, except for Bosnian, Croatian and Polish (pre-2006 only) and Korean (2006 onwards only). **57%** arrived from 2006 – 2016 and **43%** arrived pre-2006. Advantaged people comprise **17.5%** of the dataset. These time periods and gender markers may help identify generational groups within these communities, and could be useful if one subset of an ethnic group needed to be engaged to reach out to a less advantaged part.

Conclusions: This investigation shows the cultural diversity and richness of the City of Greater Dandenong, demonstrated by the spread of languages spoken and decade of arrival. It also shows that there are multiple demographic and social profiles within the community, and that it is necessary to drill down into individual ethnic groups to understand just how disconnected or not people may be from Australian institutions and the messages they share. The information in this report will help authorities understand the sizes of the communities, their level of English, and the range of languages that need to be engaged with, and provides a starting point for checking whether these community groups are receiving messages in a consumable way.

Additional variables such as age, and industry sector and type of role (eg professional, labourer) for people in employment, would help define the groups. Areas with Indigenous populations could also apply this approach for community languages, as long as Australian-born people were included in the data. The approach can easily be applied to any LGA, state or region defined by the ABS.

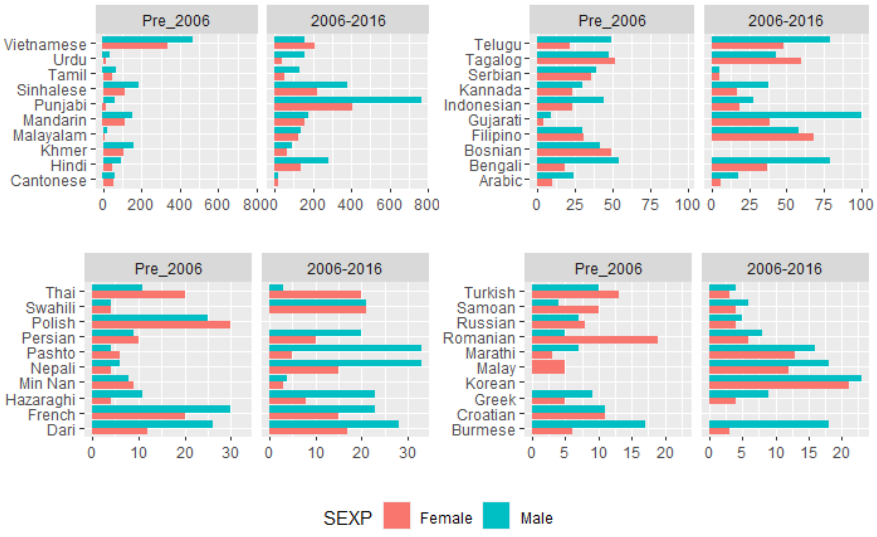


Figure 5. *Advantage by Gender, Language & Period of Arrival*

References

Australian Bureau of Statistics (2016a). *1267.0 - Australian Standard Classification of Languages (ASCL), 2016.* https://tinyurl.com/yxa9y32f

Australian Bureau of Statistics (2016b). *2901.0 - Census of Population and Housing: Census Dictionary, 2016*. https://www.abs.gov.au/ausstats/abs@.nsf/mf/2901.0

Australian Bureau of Statistics (2020, October 1). *ABS TableBuilder*. https://www.abs.gov.au/websitedbs/censushome.nsf/home/tablebuilder

Baker, N. and Lee, A. (2020, June 23). Greg Hunt warns Melbourne COVID-19 spike has potential to cause second wave. *SBS News*. https://www.sbs.com.au/news/greg-hunt-warns-melbourne-covid-19-spike-has-potential-to-cause-second-wave

Baker, N. (2020, June 16). Missing posters and 'fake' tweets: Pandemic communications strategy for multicultural Australia slammed. *SBS News*. https://www.sbs.com.au/news/missing-posters-and-fake-tweets-pandemic-communications-strategy-for-multicultural-australia-slammed

Baumer, B.S., Kaplan, D.T., & Horton, N.J. (2017). *Modern Data Science with R*. CRC Press.

City of Greater Dandenong Council (2017). *Cultural diversity*. https://greaterdandenong.com/document/10768/summaries-of-social-information-cgd

City of Greater Dandenong Council (2018, February). *Population Current and Forecast in Greater Dandenong and Suburbs*. https://greaterdandenong.com/document/10768/summaries-of-social-information

Community Four. (2020, June 30). *Refugee Women Lead Our COVID Response* [Video]. https://communityfour.org/uncategorized/refugee-women-lead-our-covid-response/

Fryer, B. (2020, June 23). Door knockers and on-the spot fines: how Victoria is tackling coronavirus community transmission. SBS News. https://www.sbs.com.au/news/door-knockers-and-on-the-spot-fines-how-victoria-is-tackling-coronavirus-community-transmission

Grey, A. (2020, June 29). Multilingual Australia is missing out on vital COVID-19 information. No wonder local councils and businesses are stepping in. *The Conversation*. https://theconversation.com/multilingual-australia-is-missing-out-on-vital-covid-19-information-no-wonder-local-councils-and-businesses-are-stepping-in-141362

Kassambara, A. (2020). *ggpubr: 'ggplot2' Based Publication Ready Plots*. R package version 0.4.0. https://CRAN.R-project.org/package=ggpubr

Lazzaro, K. (2020, October 1). Calls to make bushfire warnings easier to access for Victorians who don't speak English at home. *ABC News*. https://www.abc.net.au/news/2020-10-01/vic-emergency-app-only-in-english-investment-in-translation/12710414

Müller, K. & Wickham, W. (2020). *tibble: Simple Data Frames*. Rcpackage version 3.0.3. https://CRAN.R-project.org/package=tibble

National COVID-19 Health and Research Advisory Committee (2020). *Risks of resurgence of COVID-19 in Australia*. Australian Government National Health and Medical Research Council. https://www.nhmrc.gov.au/about-us/leadership-and-

governance/committees/national-covid-19-health-and-research-advisory-committee

RStudio Team (2020). *RStudio: Integrated Development Environment for R*. RStudio, PBC, Boston, MA. URL http://www.rstudio.com/. Mode: Desktop. Version: 1.3.1073

Wickham, H., & Grolemund, G. (2017). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.* O'Reilly Media.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H., Averick, M., Bryan. J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., LinPedersen, T., Miller, E., Bache, S. M., Müller,K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V. ... Yutani, H., (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., François, R., Henry, L. & Müller, K. (2020a). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.0. https://CRAN.R-project.org/package=dplyr

Wickham, H. (2020b). *forcats: Tools for Working with Categorical Variables (Factors)*. R package version 0.5.0. https://CRAN.R-project.org/package=forcats

Wild, A., Kunstler, B., Goodwin, D., & Skouteris, H. (2020, July 17). We asked multicultural communities how best to communicate COVID-19 advice. Here's what they told us. *The Conversation*. https://theconversation.com/we-asked-multicultural-communities-how-best-to-communicate-covid-19-advice-heres-what-they-told-us-142719

Wilkins, David (2019). *treemapify: Draw Treemaps in 'ggplot2'*. R package version 2.5.3. https://CRAN.R-project.org/package=treemapify

Appendix A – Raw data cleaning

This shows the semi-structured format of the data tables that were downloaded from the ABS Table Builder and the minimal formatting done in Excel to ensure the data was readable by R. In-principle advice was given via the LearnJCU Assessment discussion board that this was acceptable.

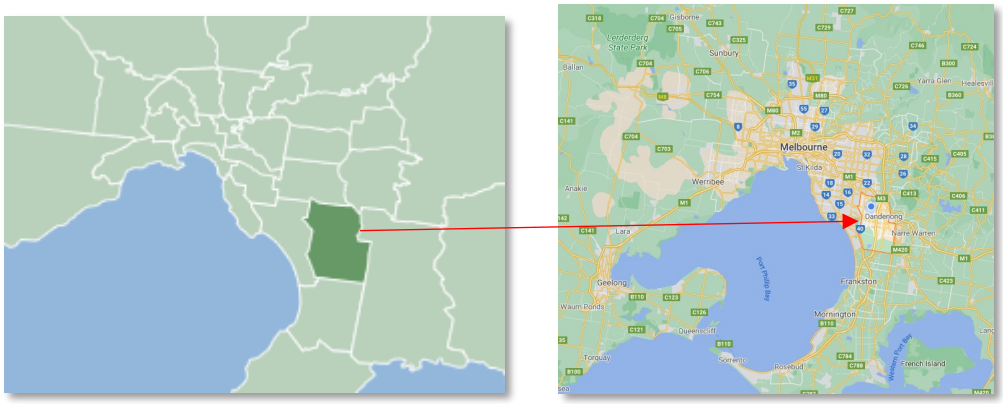
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Australian	Bureau of Statistics																	
2																			
3	2016 Census - Counting Persons, Place of Enumeration (MB)																		
4	SEXP Sex by HEAP Level of Highest Educational Attainment, EETP Engagement in Employment, Education and Training, NEDD Dwelling Internet Connection, ENGP Proficiency in Spoken English, BPLP -																		
5	Counting: Persons Location on Census Night																		
6																			
7	Filters:																		
8	Default Su	Persons Location on Census Night																	
9	LGA	Greater Dandenong (C)																	
10	LANP - 4 D	Albanian																	
11																			
12	HEAP Leve	Higher							Secondary										
13	EETP Enga	Partial	Not_Engaged				Fully		Partial			Not_Engaged							
14	NEDD Dwe	Internet a	Internet accessed from dwelling				Internet accessed fr		Internet accessed from dwellir			Internet accessed from dwelling							
15	ENGP Prof	Adequate	Adequate				Adequate		Adequate			Inadequate			Adequate				
16	BPLP - 4 D	The forme	Albania	The forme	Kosovo		Albania	The forme	Albania		The forme	Albania	The forme	Kosovo	Albania		The former	Yugoslav	Kosovo
17	YARRP Ye	Pre_2006	Pre_2006	Pre_2006	Pre_2006	2006-15	Pre_2006	Pre_2006	Pre_2006	2006-15	Pre_2006	Pre_2006	Pre_2006	Pre_2006	Pre_2006	2006-15	Pre_2006	2006-15	Pre_2006
18	SEXP Sex																		
19	Male	3	0	3	0	0	0	3	3	0	10	6	6	0	0	0	22	0	11
20	Female	0	0	0	0	0	4	0	5	3	4	0	4	3	8	3	17	3	6
21																			
22	Data Source: Census of Population and Housing, 2016, TableBuilder																		
23																			
24	INFO	Cells in this table have been randomly adjusted to avoid the release of confidential data. No reliance should be placed on small cells.																	
25																			
26																			
27	Copyright Commonwealth of Australia, 2018, see abs.gov.au/copyright																		
28	ABS data licensed under Creative Commons, see abs.gov.au/ccby																		
29																			

Figure A1. *Sample data table download using ABS Table Builder*

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	LGA	Greater D	Greater D	Greater D	Greater D	Greater D	Greater D	Greater D	Greater D	Greater D	Greater D	Greater D	Greater D	Greater D	Greater D	Grea
2	LANP	Albanian	Albanian	Albanian	Albanian	Albanian	Albanian	Albanian	Albanian	Albanian	Albanian	Albanian	Albanian	Albanian	Albanian	Alba
3	HEAP	Higher	Higher	Higher	Higher	Higher	Higher	Higher	Secondary	Secondary	Secondary	Secondary	Secondary	Secondary	Secondary	Seco
4	EETP	Partial	Not_Enga	Not_Enga	Not_Enga	Not_Enga	Fully	Fully	Partial	Partial	Partial	Not_Enga	Not_Enga	Not_Enga	Not_Enga	Not_
5	NEDD	Internet a	Internet a	Internet a	Internet a	Internet a	Internet a	Internet a	Internet a	Internet a	Internet a	Internet a	Internet a	Internet a	Internet a	Inter
6	ENGP	Adequate	Adequate	Adequate	Adequate	Adequate	Adequate	Adequate	Adequate	Adequate	Adequate	Inadequat	Inadequat	Inadequat	Adequate	Adec
7	BPLP	The forme	Albania	The forme	Kosovo	Kosovo	Albania	The forme	Albania	Albania	The forme	Albania	The forme	Kosovo	Albania	Alba
8	YARRP	Pre_2006	Pre_2006	Pre_2006	Pre_2006	2006-15	Pre_2006	Pre_2006	Pre_2006	2006-15	Pre_2006	Pre_2006	Pre_2006	Pre_2006	Pre_2006	2006
9	Male	3	0	3	0	0	0	3	3	0	10	6	6	0	0	
10	Female	0	0	0	0	0	4	0	5	3	4	0	4	3	8	
11																
12																

Figure A2. *Sample data table that has been formatted in Excel for upload to R*

Appendix B – Location of the City of Greater Dandenong, Victoria, Australia



https://en.wikipedia.org/wiki/City_of_Greater_Dandenong Google map excerpt

Appendix C – R code

```
library(tidyr)
library(dplyr)
library(tibble)
```

1_Import_Tidy

```
## READ IN THE DATA
DanSerbian <- read.csv("t3_final_Dan_Serbian.csv", stringsAsFactors =
TRUE, header = FALSE)
#Repeat for 111 more files (NB done in batches of 20)
## CREATE FUNCTION
#define tidy function to convert wide to long format, create variable
names and gather two cols (Male, Female) into one Gender variable
(NB I created this custom function)
tidy <- function(x) {
  x2 <- data.frame(t(x[-1]))
  colnames(x2) <- x[, 1]
  x2 <- remove_rownames(x2)
  x3 <- x2 %>% gather(Male, Female, key = "SEXP", value = "Count")
}
## APPLY FUNCTION
# tidy output & save with Dan prefix removed
Serbian <- tidy(DanSerbian)
# Repeat for 111 more files (NB done in batches of 20)
```

2_Join

```
## MERGE ALL TABLES INTO ONE MASTER TABLE
# Batches 1 & 2
AllLangs <- rbind(Vietnamese, Greek, Sinhalese, Hazaraghi, Tamil, Dari,
Arabic, Hindi, Italian, Khmer, Punjabi, Mandarin, Cantonese, Serbian,
Albanian, Turkish, Burmese, Spanish, Urdu, Bosnian, Malayalam, Tagalog,
French, Persian_ex_Dari, Croatian, Min_Nan, Pashto, Polish, Samoan,
Filipino, Bengali, Telugu, Gujarati, Indonesian, Rohingya, Thai, Russian,
Hungarian, Hakka)
# Incorporate Batches 3 – 6
AllLangs <-
rbind(AllLangs,Korean,Romanian,Kannada,Macedonian,Malay,Somali,Mao
riCookIsland,German,ChineseNfd,MauritianCreole,Oromo,Maltese,Nuer,N
epali,Armenian,SouthernAsianNfd,Lao,Portugese,Marathi,Japanese,Africa
nLangNfd,ChaldeanNeoAramaic,Kurdish,Mon,Slovak,Bisaya,IndoAryanNfd
,Tetum,Tulu,FijianHindustani,Timorese,Yoruba,Finnish,Bulgarian,Cebuano,
Hebrew,Igbo,Ndebele,PidginNfd,Hausa,Oriya,TokPisin,ChinHaka,Acholi,Tig
re,Akan,Ilokano,SerboCroatYugo,Dutch,Swahili,FrenchCreoleNfd,Karen,Cr
eoleNfd,Amharic,Dinka,Shona,MaoriNZ,Ukrainian,Harari,Afrikaans,Tongan
,Czech,Konkani,Krio,Tibetan,Uygur,Iranic,Shilluk,Kirundi,Fijian,AfricanLang
Nec,Slovene,Kinyarwanda)
# tidy column names (remove spaces)
names(AllLangs)<- str_replace_all(names(AllLangs), c(" " = ""))
colnames(AllLangs)
# Check
dim(AllLangs)
str(AllLangs)
```

3_Assign_Class

```
library(forcats)
# Remove 0 values
AllLangs <- AllLangs %>% filter( Count > 0)
# order factors: YARRP
years <- c("Pre_2006", "2006-15", "2016") # create vector in correct order
years <- as.factor(years)
AllLangs_2$YARRP <- factor(AllLangs_2$YARRP, levels = years, ordered =
TRUE)
levels(AllLangs_2$YARRP)
## ASSIGN CLASS
# set class
AllLangs$LGA <- as.factor(AllLangs$LGA)
AllLangs$LANP <- as.factor(AllLangs$LANP)
AllLangs$HEAP <- as.factor(AllLangs$HEAP)
AllLangs$EETP <- as.factor(AllLangs$EETP)

AllLangs$NEDD <- as.factor(AllLangs$NEDD)
AllLangs$ENGP <- as.factor(AllLangs$ENGP)
AllLangs$BPLP <- as.factor(AllLangs$BPLP)
AllLangs$YARRP <- as.factor(AllLangs$YARRP)
AllLangs$SEXP <- as.factor(AllLangs$SEXP)
AllLangs$Count <- as.numeric(AllLangs$Count)

## REVIEW FACTOR LEVELS
### LGA
# review factor levels
levels(AllLangs$LGA)
# recode
AllLangs <- AllLangs %>%  mutate(LGA = fct_recode(LGA,
"Greater_Dandenong" = "Greater Dandenong (C)"))
AllLangs %>% count(LGA)
### HEAP
levels(AllLangs$HEAP)
# reorder levels
education <- c("Non_Secondary", "Secondary", "Higher") # create vector
in correct order
education <- as.factor(education)
AllLangs$HEAP <- factor(AllLangs$HEAP, levels = education, ordered =
TRUE)
levels(AllLangs$HEAP)
### EETP
levels(AllLangs$EETP)
# reorder levels
engagement <- c("Not_Engaged", "Partial", "Fully") # create vector in
correct order
engagement <- as.factor(engagement)
AllLangs$EETP <- factor(AllLangs$EETP, levels = engagement, ordered =
TRUE)
levels(AllLangs$EETP)
### NEDD
levels(AllLangs$NEDD)
# identify obs with no level
AllLangs %>% filter(NEDD == "")
# add missing levels
# recode the obs with no level after checking in raw data then add to df
AllLangs <- AllLangs %>%  mutate(NEDD = fct_recode(NEDD,  "Internet
accessed from dwelling" = ""))
#simplify levels
levels(AllLangs$NEDD) <- c("Internet","No_Internet")
### ENGP
levels(AllLangs$ENGP)
### YARRP
levels(AllLangs$YARRP)
# order factors: YARRP
years <- c("Pre_2006", "2006-15", "2016") # create vector in correct order
years <- as.factor(years)
AllLangs$YARRP <- factor(AllLangs$YARRP, levels = years, ordered = TRUE)
### SEXP
levels(AllLangs$SEXP)
### LANP
# review factor levels
levels(AllLangs$LANP) # long
```

4_Summarise

```
# make a copy
table <- AllLangs # working table
## WEIGHTED COUNTS
sum(table$total) # sum of cases is 57,825
# compute counts per variable
table %>% count(HEAP, wt = Count)
table %>% count(EETP, wt = total)
table %>% count(NEDD, wt = total)
table %>% count(ENGP, wt = total)
table %>% count(YARRP, wt = total)
table %>% count(SEXP, wt = total)
table %>% count(LANP, wt = total) %>% arrange(desc(n))
## COMBINING VARIABLES IN SUMMARIES TO UNDERSTAND THE DATA
(EXPLORATION)
df <- table # make another copy
df %>% select(EETP, NEDD, total) #repeat as often as needed by changing
variables
# grouping and summarising - arranged by population
table %>% group_by(LANP, EETP) %>% summarise(total_pop = sum(total))
%>% arrange(desc(total_pop))
#
table %>% filter(LANP == "Urdu") %>%  group_by(ENGP, SEXP) %>%
summarise(Total = sum(total)) %>% mutate(Grand_Total = sum(Total))
%>% mutate(Perc = Total / Grand_Total * 100)
```

5_Join_Classification

```
# upload classification
LangsClass<- read.csv("langs-classification.csv", stringsAsFactors = TRUE, header = TRUE)
## JOIN CLASIFICATIONS BY KEY
# first need to remove the "," in LANP levels
levels(df1$LANP) <- gsub(", nfd"," nfd", levels(df1$LANP))
levels(df1$LANP) <- gsub(", nec"," nec", levels(df1$LANP))
levels(df1$LANP)
# join using inner_join
df2 <- df1 %>% left_join(LangsClass, c("LANP"= "Language4DC"))
head(df2)
# check for missing values
df2[!complete.cases(df2),]
# clean levels
df2 <- df2 %>%  mutate(LANP = fct_recode(LANP, "Serbo-Croatian/Yugoslavian so described" = "Serbo-Croatian/Yugoslavian, so described"))
df2 <- df2 %>%  mutate(LANP = fct_recode(LANP, "Southern Asian Languages" = "Southern Asian Languages nfd"))
# order levels
# create vector in correct order
LangGroup <- c("Northern European Languages","Southern European Languages","Eastern European Languages","Southwest and Central Asian Languages","Southern Asian Languages","Southeast Asian Languages", "Eastern Asian Languages", "Australian Indigenous Languages", "Other Languages") LangGroup <- as.factor(LangGroup)
df2$GroupName1DC<- factor(df2$GroupName1DC, levels = LangGroup, ordered = TRUE)
```

6_Visualise

```
library(ggplot2)
install.packages("treemapify")
library(treemapify)
# create treemap Fig 1
df2 %>% count(LANP, GroupName1DC, wt = total) %>%   ggplot(aes(area = n, fill = GroupName1DC, label = LANP)) +  geom_treemap() + geom_treemap_text(fontface = "italic", colour = "white", place = "centre", grow = TRUE) + labs(fill = "Language Group") + ggtitle("Languages other than English spoken in the City of Greater Dandenong")

# create bar chart Fig 2
g21 <- mutate(group_by(df2, HEAP, GroupName1DC), Prop = total / sum(total))
ggplot(g21) + geom_col(aes(GroupName1DC, Prop, fill = ENGP), position = "stack") + facet_wrap(~HEAP, ncol = 4) + coord_flip() + theme(axis.text.x = element_text(angle = 90, hjust = 1)) + labs(fill = "English") + xlab("Language Group") + ylab("Proportion")

# create faceted scatterplots Fig 3
df2 %>% filter(GroupName1DC == "Eastern Asian Languages") %>% ggplot +  geom_point(mapping = aes(x = HEAP, y = SEXP, colour = ENGP, size = total), position = position_jitter(width = 0.2, height = 0.2),  alpha = 0.6) + labs(colour = "English", size = "Count") + xlab("Educational level") + ylab("Gender") + facet_wrap(.~LANP, ncol = 4) + theme(axis.text.x = element_text(angle = 45, hjust = 1))

## CREATE DATASETS FOR SOCIAL ADVANTAGE/ DISADVANTAGED GROUPS
# create disadvantaged df
Disadvan <- df2 %>% filter(ENGP == "Inadequate", NEDD == "No_Internet", EETP == "Not_Engaged", HEAP == "Non_Secondary")
#
Disadvan %>% group_by(LANP, BPLP, SEXP) %>% count(LANP, wt = total) %>% arrange(desc(n))
#combine levels
Disadvan1 <- Disadvan #make a copy
levels(Disadvan1$YARRP) <- list("Pre_2006"=c("Pre_2006"), "2006-2016"=c("2006-15", "2016"))
# create a vector of the languages with min 30 count
DisadvanLang <- Disadvan1 %>% count(LANP, wt = total) %>%   filter(n >= 30)
# create graph of just the langs with >= 30 Fig 4
Disadvan1 %>% subset(LANP %in% DisadvanLang$LANP) %>% ggplot(aes(x=LANP, y=total, fill=SEXP)) + geom_bar(stat="identity", position="dodge") + coord_flip() + facet_grid(.~YARRP) + theme(legend.position = "top",  axis.text.x = element_text(angle = 45, hjust = 1, vjust = 0.5)) +  xlab("Language") + ylab("Count >= 30") + labs(fill = "Gender")

**EXPLORATION CODE**
# use this code to explore the counts by level; insert relevant variable
Disadvan1 %>% count(SEXP, wt = total)
# compute summary stats
Disadvanperc <- sum(Disadvan$total)/57825 * 100    #1140
Disadvanperc
Advanperc <- sum(Advan$total)/57825 * 100    #10116
Advanperc
# create advantaged df
Advan <- df2 %>% filter(ENGP == "Adequate", NEDD == "Internet", EETP == "Fully", HEAP == "Higher")
# compute summary stats
Advanperc_M <- 6063/ (6063 + 4053)
Advanperc_F <- 4053 / (6063 + 4053)
Advanperc_M
Advanperc_F
sum(df2$total) #57825
sum(Advan$total) #10116
Disadvan1 %>% count(LANP, wt = total)
sum(Disadvan1$total) #1140
# combine levels
Advan1 <- Advan #make a copy
levels(Advan1$YARRP) <- list("Pre_2006"=c("Pre_2006"), "2006-2016"=c("2006-15", "2016"))
head(Advan1)
# create a vector of the languages with min 30 count
AdvanLang <- Advan1 %>% count(LANP, wt = total) %>%   filter(n >= 30) %>% arrange(desc(n))
AdvanLang # This computes 40 languages

## SEPARATE LANGS INTO GROUPS OF 10 FOR BETTER PLOTTING
# "Binning' the languages into 4 groups by rank
OneTen <- AdvanLang[1:10,] #top 10
ElevenTwen <- AdvanLang[11:20,]
TwenOneThir <- AdvanLang[21:30,]
ThirOneForty <- AdvanLang[31:40,]

## CREATE PLOTS FOR EACH LANGUAGE 'BIN' GROUP
# recode persian as it is too long and messes up the alignment
Advan1 <- Advan1 %>% mutate(LANP = fct_recode(LANP,  "Persian" = "Persian (excluding Dari)"))
TwenOneThir <- TwenOneThir %>% mutate(LANP = fct_recode(LANP, "Persian" = "Persian (excluding Dari)"))
# create plots
# 1 – 10
a1 <- Advan1 %>% subset(LANP %in% OneTen$LANP) %>% ggplot(aes(x=LANP, y=total, fill=SEXP)) + geom_bar(stat="identity", position="dodge") + coord_flip() + facet_grid(.~YARRP) + ylab("") + theme(legend.position="none") + xlab(NULL)
# 11 – 20
a2 <- Advan1 %>% subset(LANP %in% ElevenTwen$LANP) %>% ggplot(aes(x=LANP, y=total, fill=SEXP)) + geom_bar(stat="identity", position="dodge") + coord_flip() + facet_grid(.~YARRP) + ylab("") + theme(legend.position="none") + xlab(NULL)
# 21 – 30
a3 <- Advan1 %>% subset(LANP %in% TwenOneThir$LANP) %>% ggplot(aes(x=LANP, y=total, fill=SEXP)) +  geom_bar(stat="identity", position="dodge") + coord_flip() +  facet_grid(.~YARRP) +  ylab("") + theme(legend.position="none") + xlab(NULL)
# 31 – 40
a4 <- Advan1 %>% subset(LANP %in% ThirOneForty$LANP) %>% ggplot(aes(x=LANP, y=total, fill=SEXP)) + geom_bar(stat="identity", position="dodge") + coord_flip() +  facet_grid(.~YARRP) + ylab("") + xlab(NULL) + labs(fill = "Gender")

Nb I tried to order the languages in each group in descending order by count but could not succeed

# display as a panel Fig 5
install.packages("ggpubr")
library(ggpubr)
ggarrange(a1, a2, a3, a4, ncol=2, nrow=2, common.legend = TRUE, legend="bottom")
```