1_import_tidy

Fiona Neilson

23/06/2021

1 DATA PRE-PROCESSING

1.1 Import libraries

```
library(tidyr)
library(dplyr)
library(tibble)
library(forcats)
library(stringr)
library(tidyverse)
```

1.2 Read in the data

I manually read in 112 CSV files with ABS microdata. The Dan prefix refers to 'City of Greater Dandenong', which is the Local Government Area or council name. Note: As I was not experienced in writing functions at the time of this work, I created a line of code per file.

```
# 1
DanSerbian <- read.csv("t3_final_Dan_Serbian.csv", stringsAsFactors = TRUE, header = FALSE)
DanSinhalese <- read.csv("t3_final_Dan_Sinhalese.csv", stringsAsFactors = TRUE, header = FALSE)
DanSpanish <- read.csv("t3_final_Dan_Spanish.csv", stringsAsFactors = TRUE, header = FALSE)
DanTamil <- read.csv("t3_final_Dan_Tamil.csv", stringsAsFactors = TRUE, header = FALSE)
DanTurkish <- read.csv("t3_final_Dan_Turkish.csv", stringsAsFactors = TRUE, header = FALSE)
DanUrdu <- read.csv("t3_final_Dan_Urdu.csv", stringsAsFactors = TRUE, header = FALSE)
DanAlbanian <- read.csv("t3_final_Dan_Albanian.csv", stringsAsFactors = TRUE, header = FALSE)
DanBurmese <- read.csv("t3_final_Dan_Burmese.csv", stringsAsFactors = TRUE, header = FALSE)
DanHindi <- read.csv("t3_final_Dan_Hindi.csv", stringsAsFactors = TRUE, header = FALSE)
DanItalian <- read.csv("t3_final_Dan_Italian.csv", stringsAsFactors = TRUE, header = FALSE)
DanArabic <- read.csv("t3_final_Dan_Arabic.csv", stringsAsFactors = TRUE, header = FALSE)
DanCantonese <- read.csv("t3_final_Dan_Cantonese.csv", stringsAsFactors = TRUE, header = FALSE)
DanDari <- read.csv("t3_final_Dan_Dari.csv", stringsAsFactors = TRUE, header = FALSE)
DanGreek <- read.csv("t3_final_Dan_Greek.csv", stringsAsFactors = TRUE, header = FALSE)
DanHazaraghi <- read.csv("t3_final_Dan_Hazaraghi.csv", stringsAsFactors = TRUE, header = FALSE)
DanKhmer <- read.csv("t3_final_Dan_Khmer.csv", stringsAsFactors = TRUE, header = FALSE)
DanMandarin <- read.csv("t3_final_Dan_Mandarin.csv", stringsAsFactors = TRUE, header = FALSE)
DanPunjabi <- read.csv("t3_final_Dan_Punjabi.csv", stringsAsFactors = TRUE, header = FALSE)
DanTigrinya <- read.csv("t3_final_Dan_Tigrinya.csv", stringsAsFactors = TRUE, header = FALSE)
DanVietnamese <- read.csv("t3_final_Dan_Vietnamese.csv", stringsAsFactors = TRUE, header = FALSE
)
```

```
# 2
DanBosnian <- read.csv("t3_final_Dan_Bosnian.csv", stringsAsFactors = TRUE, header = FALSE)
DanMalayalam <- read.csv("t3_final_Dan_Malayalam.csv", stringsAsFactors = TRUE, header = FALSE)
DanTagalog <- read.csv("t3_final_Dan_Tagalog.csv", stringsAsFactors = TRUE, header = FALSE)
DanFrench <- read.csv("t3_final_Dan_French.csv", stringsAsFactors = TRUE, header = FALSE)
DanPersian_ex_Dari <- read.csv("t3_final_Dan_Persian_ex_Dari.csv", stringsAsFactors = TRUE, head
er = FALSE)
DanCroatian <- read.csv("t3_final_Dan_Croatian.csv", stringsAsFactors = TRUE, header = FALSE)
DanMin_Nan <- read.csv("t3_final_Dan_Min_Nan.csv", stringsAsFactors = TRUE, header = FALSE)
DanPashto <- read.csv("t3_final_Dan_Pashto.csv", stringsAsFactors = TRUE, header = FALSE)
DanPolish <- read.csv("t3_final_Dan_Polish.csv", stringsAsFactors = TRUE, header = FALSE)
DanSamoan <- read.csv("t3_final_Dan_Samoan.csv", stringsAsFactors = TRUE, header = FALSE)
DanFilipino <- read.csv("t3_final_Dan_Filipino.csv", stringsAsFactors = TRUE, header = FALSE)
DanBengali <- read.csv("t3_final_Dan_Bengali.csv", stringsAsFactors = TRUE, header = FALSE)
DanTelugu <- read.csv("t3_final_Dan_Telugu.csv", stringsAsFactors = TRUE, header = FALSE)
DanGujarati <- read.csv("t3_final_Dan_Gujarati.csv", stringsAsFactors = TRUE, header = FALSE)
DanIndonesian <- read.csv("t3_final_Dan_Indonesian.csv", stringsAsFactors = TRUE, header = FALSE
)
DanRohingya <- read.csv("t3_final_Dan_Rohingya.csv", stringsAsFactors = TRUE, header = FALSE)
DanThai <- read.csv("t3_final_Dan_Thai.csv", stringsAsFactors = TRUE, header = FALSE)
DanRussian <- read.csv("t3_final_Dan_Russian.csv", stringsAsFactors = TRUE, header = FALSE)
DanHungarian <- read.csv("t3_final_Dan_Hungarian.csv", stringsAsFactors = TRUE, header = FALSE)
DanHakka <- read.csv("t3_final_Dan_Hakka.csv", stringsAsFactors = TRUE, header = FALSE)
# 3
DanKorean <- read.csv("t3_final_Dan_Korean.csv", stringsAsFactors = TRUE, header = FALSE)
DanRomanian <- read.csv("t3_final_Dan_Romanian.csv", stringsAsFactors = TRUE, header = FALSE)
DanKannada <- read.csv("t3_final_Dan_Kannada.csv", stringsAsFactors = TRUE, header = FALSE)
DanMacedonian <- read.csv("t3_final_Dan_Macedonian.csv", stringsAsFactors = TRUE, header = FALSE
)
DanMalay <- read.csv("t3_final_Dan_Malay.csv", stringsAsFactors = TRUE, header = FALSE)
DanSomali <- read.csv("t3_final_Dan_Somali.csv", stringsAsFactors = TRUE, header = FALSE)
DanMaoriCookIsland <- read.csv("t3_final_Dan_Maori_Cook_Island.csv", stringsAsFactors = TRUE, he
ader = FALSE)
DanGerman <- read.csv("t3_final_Dan_German.csv", stringsAsFactors = TRUE, header = FALSE)
DanChineseNfd <- read.csv("t3_final_Dan_Chinese_nfd.csv", stringsAsFactors = TRUE, header = FALS
E)
DanMauritianCreole <- read.csv("t3_final_Dan_Mauritian_Creole.csv", stringsAsFactors = TRUE, hea
der = FALSE)
DanOromo <- read.csv("t3_final_Dan_Oromo.csv", stringsAsFactors = TRUE, header = FALSE)
DanMaltese <- read.csv("t3_final_Dan_Maltese.csv", stringsAsFactors = TRUE, header = FALSE)
DanNuer <- read.csv("t3_final_Dan_Nuer.csv", stringsAsFactors = TRUE, header = FALSE)
DanNepali <- read.csv("t3_final_Dan_Nepali.csv", stringsAsFactors = TRUE, header = FALSE)
DanArmenian <- read.csv("t3_final_Dan_Armenian.csv", stringsAsFactors = TRUE, header = FALSE)
DanSouthernAsianNfd <- read.csv("t3_final_Dan_Southern_Asian_nfd.csv", stringsAsFactors = TRUE,
header = FALSE)
DanLao <- read.csv("t3_final_Dan_Lao.csv", stringsAsFactors = TRUE, header = FALSE)
DanPortugese <- read.csv("t3_final_Dan_Portugese.csv", stringsAsFactors = TRUE, header = FALSE)
DanMarathi <- read.csv("t3_final_Dan_Marathi.csv", stringsAsFactors = TRUE, header = FALSE)
DanJapanese <- read.csv("t3_final_Dan_Japanese.csv", stringsAsFactors = TRUE, header = FALSE)
```

```r
# 4
DanSerboCroatYugo <- read.csv("t3_final_Dan_Serbo_Croat_Yugo_sodescribed.csv", stringsAsFactors
= TRUE, header = FALSE)
DanDutch <- read.csv("t3_final_Dan_Dutch.csv", stringsAsFactors = TRUE, header = FALSE)
DanSwahili <- read.csv("t3_final_Dan_Swahili.csv", stringsAsFactors = TRUE, header = FALSE)
DanFrenchCreoleNfd <- read.csv("t3_final_Dan_French_Creole_nfd.csv", stringsAsFactors = TRUE, he
ader = FALSE)
DanKaren <- read.csv("t3_final_Dan_Karen.csv", stringsAsFactors = TRUE, header = FALSE)
DanCreoleNfd <- read.csv("t3_final_Dan_Creole_nfd.csv", stringsAsFactors = TRUE, header = FALSE)
DanAmharic <- read.csv("t3_final_Dan_Amharic.csv", stringsAsFactors = TRUE, header = FALSE)
DanDinka <- read.csv("t3_final_Dan_Dinka.csv", stringsAsFactors = TRUE, header = FALSE)
DanShona <- read.csv("t3_final_Dan_Shona.csv", stringsAsFactors = TRUE, header = FALSE)
DanMaori_NZ <- read.csv("t3_final_Dan_Maori_NZ.csv", stringsAsFactors = TRUE, header = FALSE)
DanUkrainian <- read.csv("t3_final_Dan_Ukrainian.csv", stringsAsFactors = TRUE, header = FALSE)
DanHarari <- read.csv("t3_final_Dan_Harari.csv", stringsAsFactors = TRUE, header = FALSE)
DanAfrikaans <- read.csv("t3_final_Dan_Afrikaans.csv", stringsAsFactors = TRUE, header = FALSE)
DanTongan <- read.csv("t3_final_Dan_Tongan.csv", stringsAsFactors = TRUE, header = FALSE)
DanCzech <- read.csv("t3_final_Dan_Czech.csv", stringsAsFactors = TRUE, header = FALSE)
DanKonkani <- read.csv("t3_final_Dan_Konkani.csv", stringsAsFactors = TRUE, header = FALSE)
DanKrio <- read.csv("t3_final_Dan_Krio.csv", stringsAsFactors = TRUE, header = FALSE)
DanTibetan <- read.csv("t3_final_Dan_Tibetan.csv", stringsAsFactors = TRUE, header = FALSE)
DanUygur <- read.csv("t3_final_Dan_Uygur.csv", stringsAsFactors = TRUE, header = FALSE)
DanIranic <- read.csv("t3_final_Dan_Iranic.csv", stringsAsFactors = TRUE, header = FALSE)

# 5
DanShilluk <- read.csv("t3_final_Dan_Shilluk.csv", stringsAsFactors = TRUE, header = FALSE)
DanKirundi <- read.csv("t3_final_Dan_Kirundi.csv", stringsAsFactors = TRUE, header = FALSE)
DanFijian <- read.csv("t3_final_Dan_Fijian.csv", stringsAsFactors = TRUE, header = FALSE)
DanAfricanLangNec <- read.csv("t3_final_Dan_African_Lang_nec.csv", stringsAsFactors = TRUE, head
er = FALSE)
DanSlovene <- read.csv("t3_final_Dan_Slovene.csv", stringsAsFactors = TRUE, header = FALSE)
DanAfricanLangNfd <- read.csv("t3_final_Dan_African_nfd.csv", stringsAsFactors = TRUE, header =
FALSE)
DanChaldeanNeoAramaic <- read.csv("t3_final_Dan_Chaldean_Neo_Aramaic.csv", stringsAsFactors = TR
UE, header = FALSE)
DanKurdish <- read.csv("t3_final_Dan_Kurdish.csv", stringsAsFactors = TRUE, header = FALSE)
DanMon <- read.csv("t3_final_Dan_Mon.csv", stringsAsFactors = TRUE, header = FALSE)
DanSlovak <- read.csv("t3_final_Dan_Slovak.csv", stringsAsFactors = TRUE, header = FALSE)
DanBisaya <- read.csv("t3_final_Dan_Bisaya.csv", stringsAsFactors = TRUE, header = FALSE)
DanIndoAryanNfd <- read.csv("t3_final_Dan_Indo_Aryan_nfd.csv", stringsAsFactors = TRUE, header =
FALSE)
DanTetum <- read.csv("t3_final_Dan_Tetum.csv", stringsAsFactors = TRUE, header = FALSE)
DanTulu <- read.csv("t3_final_Dan_Tulu.csv", stringsAsFactors = TRUE, header = FALSE)
DanFijianHindustani <- read.csv("t3_final_Dan_Fijian_Hindustani.csv", stringsAsFactors = TRUE, h
eader = FALSE)
DanTimorese <- read.csv("t3_final_Dan_Timorese.csv", stringsAsFactors = TRUE, header = FALSE)
DanYoruba <- read.csv("t3_final_Dan_Yoruba.csv", stringsAsFactors = TRUE, header = FALSE)
DanFinnish <- read.csv("t3_final_Dan_Finnish.csv", stringsAsFactors = TRUE, header = FALSE)
DanBulgarian <- read.csv("t3_final_Dan_Bulgarian.csv", stringsAsFactors = TRUE, header = FALSE)
DanCebuano <- read.csv("t3_final_Dan_Cebuano.csv", stringsAsFactors = TRUE, header = FALSE)
```

```
# 6
DanHebrew <- read.csv("t3_final_Dan_Hebrew.csv", stringsAsFactors = TRUE, header = FALSE)
DanKinyarwanda <- read.csv("t3_final_Dan_Kinyarwanda.csv", stringsAsFactors = TRUE, header = FAL
SE)
DanIgbo <- read.csv("t3_final_Dan_Igbo.csv", stringsAsFactors = TRUE, header = FALSE)
DanNdebele <- read.csv("t3_final_Dan_Ndebele.csv", stringsAsFactors = TRUE, header = FALSE)
DanPidginNfd <- read.csv("t3_final_Dan_Pidgin_nfd.csv", stringsAsFactors = TRUE, header = FALSE)
DanHausa <- read.csv("t3_final_Dan_Hausa.csv", stringsAsFactors = TRUE, header = FALSE)
DanOriya <- read.csv("t3_final_Dan_Oriya.csv", stringsAsFactors = TRUE, header = FALSE)
DanTokPisin <- read.csv("t3_final_Dan_Tok_Pisin.csv", stringsAsFactors = TRUE, header = FALSE)
DanChinHaka <- read.csv("t3_final_Dan_Chin_Haka.csv", stringsAsFactors = TRUE, header = FALSE)
DanAcholi <- read.csv("t3_final_Dan_Acholi.csv", stringsAsFactors = TRUE, header = FALSE)
DanTigre <- read.csv("t3_final_Dan_Tigre.csv", stringsAsFactors = TRUE, header = FALSE)
DanAkan <- read.csv("t3_final_Dan_Akan.csv", stringsAsFactors = TRUE, header = FALSE)
DanIlokano <- read.csv("t3_final_Dan_Ilokano.csv", stringsAsFactors = TRUE, header = FALSE)
```

1.3 Transpose data, including variable names, and gather columns

1.3.1 Create function

I created a custom function to reformat the ABS data files into tidy data format, including transposing (turning rows into columns) and combining separate Male and Female columns into one.

```
# define tidy 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 = "total")
}
```

1.3.2 Apply function

As mentioned above, I was not yet proficient in writing functions that would automate the upload and application of the tidying function, so I created one line of code for every data file.

```
# tidy and save output with Dan prefix removed
Serbian <- tidy(DanSerbian)
Sinhalese <- tidy(DanSinhalese)
Spanish <- tidy(DanSpanish)
Tamil <- tidy(DanTamil)
Turkish <- tidy(DanTurkish)
Urdu <- tidy(DanUrdu)
Albanian <- tidy(DanAlbanian)
Burmese <- tidy(DanBurmese)
Hindi <- tidy(DanHindi)
Italian <- tidy(DanItalian)
Arabic <- tidy(DanArabic)
Cantonese <- tidy(DanCantonese)
```

```
Dari<- tidy(DanDari)
Greek <- tidy(DanGreek)
Hazaraghi <- tidy(DanHazaraghi)
Khmer <- tidy(DanKhmer)
Mandarin <- tidy(DanMandarin)
Punjabi <- tidy(DanPunjabi)
Tigrinya <- tidy(DanTigrinya)
Vietnamese <- tidy(DanVietnamese)
#
Bosnian <- tidy(DanBosnian)
Malayalam <- tidy(DanMalayalam)
Tagalog <- tidy(DanTagalog)
French <- tidy(DanFrench)
Persian_ex_Dari <- tidy(DanPersian_ex_Dari)
Croatian <- tidy(DanCroatian)
Min_Nan <- tidy(DanMin_Nan)
Pashto <- tidy(DanPashto)
Polish <- tidy(DanPolish)
Samoan <- tidy(DanSamoan)
Filipino <- tidy(DanFilipino)
Bengali <- tidy(DanBengali)
Telugu <- tidy(DanTelugu)
Gujarati <- tidy(DanGujarati)
Indonesian <- tidy(DanIndonesian)
Rohingya <- tidy(DanRohingya)
Thai <- tidy(DanThai)
Russian <- tidy(DanRussian)
Hungarian <- tidy(DanHungarian)
Hakka <- tidy(DanHakka)
#
Korean <- tidy(DanKorean)
Romanian <- tidy(DanRomanian)
Kannada  <- tidy(DanKannada)
Macedonian <- tidy(DanMacedonian)
Malay <- tidy(DanMalay)
Somali <- tidy(DanSomali)
MaoriCookIsland  <- tidy(DanMaoriCookIsland)
German <- tidy(DanGerman)
ChineseNfd  <- tidy(DanChineseNfd)
MauritianCreole  <- tidy(DanMauritianCreole)
Oromo <- tidy(DanOromo)
Maltese  <- tidy(DanMaltese)
Nuer  <- tidy(DanNuer)
Nepali  <- tidy(DanNepali)
Armenian <- tidy(DanArmenian)
SouthernAsianNfd <- tidy(DanSouthernAsianNfd)
Lao <- tidy(DanLao)
Portugese <- tidy(DanPortugese)
Marathi <- tidy(DanMarathi)
Japanese <- tidy(DanJapanese)
```

```
#
SerboCroatYugo <- tidy(DanSerboCroatYugo)
Dutch <- tidy(DanDutch)
Swahili <- tidy(DanSwahili)
FrenchCreoleNfd <- tidy(DanFrenchCreoleNfd)
Karen <- tidy(DanKaren)
CreoleNfd <- tidy(DanCreoleNfd)
Amharic <- tidy(DanAmharic)
Dinka <- tidy(DanDinka)
Shona <- tidy(DanShona)
MaoriNZ <- tidy(DanMaori_NZ)
Ukrainian <- tidy(DanUkrainian)
Harari <- tidy(DanHarari)
Afrikaans <- tidy(DanAfrikaans)
Tongan <- tidy(DanTongan)
Czech <- tidy(DanCzech)
Konkani <- tidy(DanKonkani)
Krio <- tidy(DanKrio)
Tibetan <- tidy(DanTibetan)
Uygur <- tidy(DanUygur)
Iranic <- tidy(DanIranic)
Shilluk <- tidy(DanShilluk)
Kirundi <- tidy(DanKirundi)
Fijian <- tidy(DanFijian)
AfricanLangNec <- tidy(DanAfricanLangNec)
Slovene <- tidy(DanSlovene)
AfricanLangNfd <- tidy(DanAfricanLangNfd)
ChaldeanNeoAramaic <- tidy(DanChaldeanNeoAramaic)
Kurdish <- tidy(DanKurdish)
Mon <- tidy(DanMon)
Slovak <- tidy(DanSlovak)
Bisaya <- tidy(DanBisaya)
IndoAryanNfd <- tidy(DanIndoAryanNfd)
Tetum <- tidy(DanTetum)
Tulu <- tidy(DanTulu)
FijianHindustani <- tidy(DanFijianHindustani)
Timorese <- tidy(DanTimorese)
Yoruba <- tidy(DanYoruba)
Finnish <- tidy(DanFinnish)
Bulgarian <- tidy(DanBulgarian)
Cebuano <-tidy(DanCebuano)
Hebrew <- tidy(DanHebrew)
Kinyarwanda <- tidy(DanKinyarwanda)
Igbo <- tidy(DanIgbo)
Ndebele <- tidy(DanNdebele)
PidginNfd <- tidy(DanPidginNfd)
Hausa <- tidy(DanHausa)
Oriya <- tidy(DanOriya)
TokPisin <- tidy(DanTokPisin)
ChinHaka <- tidy(DanChinHaka)
```

```
Acholi <- tidy(DanAcholi)
Tigre <- tidy(DanTigrinya)
Akan <- tidy(DanAkan)
Ilokano <-tidy(DanIlokano)
```

1.4 Merge dataframes into master dataframe

I combined 112 separate files into one dataframe.

```
AllLangs <- rbind(Vietnamese, Greek, Sinhalese, Hazaraghi, Tamil, Dari, Arabi
c, Hindi, Italian, Khmer, Punjabi, Mandarin, Cantonese, Serbian, Albanian, Tu
rkish, Burmese, Spanish, Urdu, Bosnian, Malayalam, Tagalog, French, Persian_e
x_Dari, Croatian, Min_Nan, Pashto, Polish, Samoan, Filipino, Bengali, Telugu,
Gujarati, Indonesian, Rohingya, Thai, Russian, Hungarian, Hakka, Korean,Roman
ian,Kannada,Macedonian,Malay,Somali,MaoriCookIsland,German,ChineseNfd,Mauriti
anCreole,Oromo,Maltese,Nuer,Nepali,Armenian,SouthernAsianNfd,Lao,Portugese,Ma
rathi,Japanese,AfricanLangNfd,ChaldeanNeoAramaic,Kurdish,Mon,Slovak,Bisaya,In
doAryanNfd,Tetum,Tulu,FijianHindustani,Timorese,Yoruba,Finnish,Bulgarian,Cebu
ano,Hebrew,Igbo,Ndebele,PidginNfd,Hausa,Oriya,TokPisin,ChinHaka,Acholi,Tigre,
Akan,Ilokano,SerboCroatYugo,Dutch,Swahili,FrenchCreoleNfd,Karen,CreoleNfd,Amh
aric,Dinka,Shona,MaoriNZ,Ukrainian,Harari,Afrikaans,Tongan,Czech,Konkani,Krio
,Tibetan,Uygur,Iranic,Shilluk,Kirundi,Fijian,AfricanLangNec,Slovene,Kinyarwan
da)
```

1.4.1 Display dimensions

This shows that there are 5848 observations (rows) and 10 variables (columns).

```
dim(AllLangs)

## [1] 5848   10
```

1.4.2 Display dataframe structure

Here we see the 10 columns: * LGA - Local Government Area * LANP - Language Spoken at Home * HEAP - Level of Highest Educational Attainment * EETP - Engagement in Employment, Education and Training * NEDD - Dwelling Internet Connection * ENGP - Proficiency in Spoken English * BPLP - Country of Birth of Person * YARRP - Year of Arrival in Australia * SEXP - Sex * Count - Count of people

I have retained the ABS Census acronyms for consistency. As per my report, I combined some of the levels or sub-categories within each of these variables where I did not require the level of granularity provided.

```
str(AllLangs)

## 'data.frame':    5848 obs. of  10 variables:
##  $ LGA  : chr  "Greater Dandenong (C)" "Greater Dandenong (C)" "Greater Da
ndenong (C)" "Greater Dandenong (C)" ...
##  $ LANP : chr  "Vietnamese" "Vietnamese" "Vietnamese" "Vietnamese" ...
##  $ HEAP : chr  "Higher" "Higher" "Higher" "Higher" ...
##  $ EETP : chr  "Partial" "Partial" "Partial" "Partial" ...
##  $ NEDD : chr  "Internet accessed from dwelling" "Internet accessed from d
welling" "Internet accessed from dwelling" "Internet accessed from dwelling"
...
##  $ ENGP : chr  "Inadequate" "Inadequate" "Inadequate" "Adequate" ...
##  $ BPLP : chr  "Vietnam" "Vietnam" "Vietnam" "New Zealand" ...
##  $ YARRP: chr  "Pre_2006" "2006-15" "2016" "Pre_2006" ...
##  $ SEXP : chr  "Male" "Male" "Male" "Male" ...
##  $ total: chr  "5" "6" "0" "0" ...
```

1.5 Remove zero values

I removed all rows with a value of 0 in the Count column.

```
# remove zero values
#3696
AllLangs <- AllLangs %>% filter(
  total > 0
)
```

This reduces the number of observations (rows) to 3696.

```
# check reduced no. of rows
dim(AllLangs)

## [1] 3696   10
```

1.6 Clean column names

I removed spaces in column names as these kinds of 'invisible' spaces can impede analysis.

```
# tidy column names (remove spaces)
names(AllLangs) <- str_replace_all(names(AllLangs), c(" "=""))
colnames(AllLangs)

##  [1] "LGA"   "LANP"  "HEAP"  "EETP"  "NEDD"  "ENGP"  "BPLP"  "YARRP" "SEXP
"
## [10] "total"
```

1.7 Assign variable class

I classified all variables. All except for total are categorical variables.

```
# set class for variables
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$total <- as.numeric(AllLangs$total)
```

I then confirmed the class and looked at the levels within each categorical variable.

```
# check variable classification
str(AllLangs)

## 'data.frame':    3696 obs. of  10 variables:
##  $ LGA  : Factor w/ 1 level "Greater Dandenong (C)": 1 1 1 1 1 1 1 1 1 1 .
..
##  $ LANP : Factor w/ 112 levels "Acholi","African Languages, nec",..: 111 1
11 111 111 111 111 111 111 111 111 ...
##  $ HEAP : Factor w/ 3 levels "Higher","Non_Secondary",..: 1 1 1 1 1 1 1 1
1 1 ...
##  $ EETP : Factor w/ 3 levels "Fully","Not_Engaged",..: 3 3 3 3 2 2 2 2 2 1
...
##  $ NEDD : Factor w/ 2 levels "Internet accessed from dwelling",..: 1 1 1 1
1 1 1 1 2 1 ...
##  $ ENGP : Factor w/ 2 levels "Adequate","Inadequate": 2 2 1 1 2 2 1 1 1 2
...
##  $ BPLP : Factor w/ 106 levels "Afghanistan",..: 105 105 105 105 105 105 1
05 105 105 105 ...
##  $ YARRP: Factor w/ 3 levels "2006-15","2016",..: 3 1 3 1 3 1 3 1 3 3 ...
##  $ SEXP : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ total: num  5 6 70 33 12 18 111 21 9 19 ...
```

1.8 Order and clean factor levels

1.8.1 LGA - Local Government Area

```
# review factor levels
levels(AllLangs$LGA)

## [1] "Greater Dandenong (C)"
```

I tidied the name, removing white space and unnecessary text.

```
# recode
AllLangs <- AllLangs %>%
  mutate(LGA = fct_recode(LGA,
  "Greater_Dandenong" = "Greater Dandenong (C)"
  ))
AllLangs %>% count(LGA)

##                 LGA    n
## 1 Greater_Dandenong 3696
```

1.8.2 HEAP - Level of Highest Educational Attainment

```
# review factor levels - HEAP
levels(AllLangs$HEAP)

## [1] "Higher"        "Non_Secondary" "Secondary"
```

I ordered the educational levels from lowest to highest, so that subsequent analyses and visualisations follow this sequence: Non Secondary > Secondary > Higher

```
# reorder levels - HEAP
education <- c("Non_Secondary", "Secondary", "Higher") # create vector in cor
rect order
education <- as.factor(education)
AllLangs$HEAP <- factor(AllLangs$HEAP, levels = education, ordered = TRUE)
levels(AllLangs$HEAP)

## [1] "Non_Secondary" "Secondary"     "Higher"
```

1.8.3 EETP - Engagement in Employment, Education and Training

```
# review factor levels
levels(AllLangs$EETP)

## [1] "Fully"       "Not_Engaged" "Partial"
```

I ordered the employment/ education/ training variable from lowest to highest level of engagement: Not Engaged > Partial > Fully

```
# 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)

## [1] "Not_Engaged" "Partial"     "Fully"
```

1.8.4 NEDD - Dwelling Internet Connection

```
# review factor levels
levels(AllLangs$NEDD)

## [1] "Internet accessed from dwelling"     "Internet not accessed from dwel
ling"
```

I shortened these for display purposes and to help create more concise code.

```
#simplify levels
levels(AllLangs$NEDD) <- c("Internet","No_Internet")
levels(AllLangs$NEDD)

## [1] "Internet"    "No_Internet"
```

1.8.5 ENGP - Proficiency in Spoken English

```
# review factor levels
levels(AllLangs$ENGP)

## [1] "Adequate"   "Inadequate"
```

No changes required

1.8.6 YARRP - Year of Arrival in Australia

```
# review factor levels
levels(AllLangs$YARRP)

## [1] "2006-15"  "2016"     "Pre_2006"
```

I ordered these chronologically: Pre 2006 > 2006-15 > 2016

```
# order levels: 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)
levels(AllLangs$YARRP)

## [1] "Pre_2006" "2006-15"  "2016"
```

1.8.7 SEXP - Sex

```
# review factor levels
levels(AllLangs$SEXP)

## [1] "Female" "Male"
```

No changes required

1.8.8 LANP - Language Spoken at Home

```
# review factor levels
levels(AllLangs$LANP)

##   [1] "Acholi"
##   [2] "African Languages, nec"
##   [3] "African Languages, nfd"
##   [4] "Afrikaans"
##   [5] "Akan"
##   [6] "Albanian"
##   [7] "Amharic"
##   [8] "Arabic"
##   [9] "Armenian"
##  [10] "Bengali"
##  [11] "Bisaya"
##  [12] "Bosnian"
##  [13] "Bulgarian"
##  [14] "Burmese"
##  [15] "Cantonese"
##  [16] "Cebuano"
##  [17] "Chaldean Neo-Aramaic"
##  [18] "Chin Haka"
##  [19] "Chinese, nfd"
##  [20] "Creole, nfd"
##  [21] "Croatian"
##  [22] "Czech"
##  [23] "Dari"
##  [24] "Dinka"
##  [25] "Dutch"
##  [26] "Fijian"
##  [27] "Fijian Hindustani"
##  [28] "Filipino"
##  [29] "Finnish"
##  [30] "French"
##  [31] "French Creole, nfd"
##  [32] "German"
##  [33] "Greek"
##  [34] "Gujarati"
##  [35] "Hakka"
##  [36] "Harari"
##  [37] "Hausa"
##  [38] "Hazaraghi"
##  [39] "Hebrew"
##  [40] "Hindi"
##  [41] "Hungarian"
##  [42] "Igbo"
##  [43] "IIokano"
##  [44] "Indo-Aryan, nfd"
##  [45] "Indonesian"
##  [46] "Iranic, nfd"
##  [47] "Italian"
##  [48] "Japanese"
##  [49] "Kannada"
##  [50] "Karen"
##  [51] "Khmer"
##  [52] "Kinyarwanda (Rwanda)"
##  [53] "Kirundi (Rundi)"
##  [54] "Konkani"
##  [55] "Korean"
##  [56] "Krio"
##  [57] "Kurdish"
##  [58] "Lao"
##  [59] "Macedonian"
##  [60] "Malay"
##  [61] "Malayalam"
##  [62] "Maltese"
##  [63] "Mandarin"
##  [64] "Maori (Cook Island)"
##  [65] "Maori (New Zealand)"
##  [66] "Marathi"
##  [67] "Mauritian Creole"
##  [68] "Min Nan"
##  [69] "Mon"
##  [70] "Ndebele"
##  [71] "Nepali"
##  [72] "Nuer"
##  [73] "Oriya"
##  [74] "Oromo"
##  [75] "Pashto"
##  [76] "Persian (excluding Dari)"
##  [77] "Pidgin, nfd"
##  [78] "Polish"
##  [79] "Portuguese"
##  [80] "Punjabi"
##  [81] "Rohingya"
##  [82] "Romanian"
##  [83] "Russian"
##  [84] "Samoan"
##  [85] "Serbian"
##  [86] "Serbo-Croatian/Yugoslavian, so described"
##  [87] "Shilluk"
##  [88] "Shona"
##  [89] "Sinhalese"
##  [90] "Slovak"
##  [91] "Slovene"
##  [92] "Somali"
##  [93] "Southern Asian Languages, nfd"
##  [94] "Spanish"
##  [95] "Swahili"
##  [96] "Tagalog"
##  [97] "Tamil"
##  [98] "Telugu"
##  [99] "Tetum"
## [100] "Thai"
## [101] "Tibetan"
## [102] "Tigrinya"
## [103] "Timorese"
## [104] "Tok Pisin (Neomelanesian)"
## [105] "Tongan"
## [106] "Tulu"
## [107] "Turkish"
## [108] "Ukrainian"
## [109] "Urdu"
## [110] "Uygur"
## [111] "Vietnamese"
## [112] "Yoruba"
```

1.9 Copy dataframe and rename

I made a copy of the dataframe and assigned a new name. This way I retained a copy of the dataframe before the join operation in 1.10, in case I needed to revert to it.

```
# copy dataframe & rename
df<- AllLangs
```

1.10 Join classifications

1.10.1 Upload languages classification

This is the ABS languages classification, Table 1.3 showing broad groups, narrow groups and individual languages. I cleaned and tidied this file prior to upload.

```
# upload classifications
LangsClass<- read.csv("langs-classification.csv", stringsAsFactors = TRUE, he
ader = TRUE)
```

1.10.2 Clean levels

A preliminary check showed that a number of the language names in my dataframe did not match the ones in the classification, due to slight formatting differences in the source data. I made minor changes to match the dataframe terms to the ones in the classification, so they would join correctly.

```
# first need to remove the "," in LANP levels
levels(df$LANP) <- gsub(", nfd"," nfd", levels(df$LANP))
levels(df$LANP) <- gsub(", nec"," nec", levels(df$LANP))

# clean more levels. Specify replacement name first, then the name being repl
aced.
levels(df$LANP) <- gsub("Serbo-Croatian/Yugoslavian, so described","Serbo-Cro
atian/Yugoslavian so described", levels(df$LANP))
levels(df$LANP) <- gsub("Southern Asian Languages nfd","Southern Asian Langua
ges", levels(df$LANP))
```

1.10.3 Join classifications by key

I used the classification's 4-digit language code to join on the 4-digit language name in my dataframe and combine the two files.

```
# join using inner_join
df <- df %>%
  left_join(LangsClass, c("LANP"= "Language4DC"))
head(df)

##                  LGA       LANP   HEAP        EETP     NEDD       ENGP    BPLP
## 1 Greater_Dandenong Vietnamese Higher     Partial Internet Inadequate Vietnam
## 2 Greater_Dandenong Vietnamese Higher     Partial Internet Inadequate Vietnam
## 3 Greater_Dandenong Vietnamese Higher     Partial Internet   Adequate Vietnam
## 4 Greater_Dandenong Vietnamese Higher     Partial Internet   Adequate Vietnam
## 5 Greater_Dandenong Vietnamese Higher Not_Engaged Internet Inadequate Vietnam
## 6 Greater_Dandenong Vietnamese Higher Not_Engaged Internet Inadequate Vietnam
##      YARRP SEXP total GroupCode1DC GroupCode2DC LanguageCode4DC
## 1 Pre_2006 Male     5            6           63            6302
## 2  2006-15 Male     6            6           63            6302
## 3 Pre_2006 Male    70            6           63            6302
## 4  2006-15 Male    33            6           63            6302
## 5 Pre_2006 Male    12            6            63             6302
## 6  2006-15 Male    18            6            63             6302
##                 GroupName1DC GroupName2DC
## 1 Southeast Asian Languages    Mon-Khmer
## 2 Southeast Asian Languages    Mon-Khmer
## 3 Southeast Asian Languages    Mon-Khmer
## 4 Southeast Asian Languages    Mon-Khmer
## 5 Southeast Asian Languages    Mon-Khmer
## 6 Southeast Asian Languages    Mon-Khmer
```

1.10.4 Check for missing values

```
# check for missing values
df[!complete.cases(df),]
```

```
##  [1] LGA             LANP            HEAP            EETP
##  [5] NEDD            ENGP            BPLP            YARRP
##  [9] SEXP            total           GroupCode1DC    GroupCode2DC
## [13] LanguageCode4DC GroupName1DC    GroupName2DC
## <0 rows> (or 0-length row.names)
```

1.10.5 Order levels

I ordered the levels in my dataframe to reflect the order in the languages classification.

```
# create vector in correct order
LangGroup <- c("Northern European Languages","Southern European Languages","E
astern European Languages","Southwest and Central Asian Languages","Southern
Asian Languages","Southeast Asian Languages", "Eastern Asian Languages", "Aus
tralian Indigenous Languages", "Other Languages")
LangGroup <- as.factor(LangGroup)
df$GroupName1DC<- factor(df$GroupName1DC, levels = LangGroup, ordered = TRUE)
```

1.11 Export file

I exported the dataframe at this point as RStudio Cloud requires me to reload the data at the start of each new file. This step is not required in RStudio Desktop. I needed to use RStudio Cloud in order to knit these documents, as this function is not supported on my laptop.

```
write.csv(df, file="out_df.csv")
```