

3_visualise

Fiona Neilson

27/06/2021

3 VISUALISE DATA

3.1 Import libraries

```
library(ggplot2)
install.packages("treemapify")
library(treemapify)
install.packages("ggpubr")
library(ggpubr)
library(dplyr)
library(forcats)
library(tidyverse)
```

3.2 Read in dataframe

```
df = read.csv("out_2_df.csv", header=TRUE)
```

3.3 Create treemap

I chose a treemap to show the proportions of languages at the ABS 1DC level, which is the highest level language group name. I selected this to give an overall view of the balance of languages rather than a precise breakdown.

Joining the languages classification to my dataframe earlier made it possible for me to link individual languages back to their parent groups.

```
# create treemap
df %>% 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 \nCity
of Greater Dandenong")
```

Languages other than English spoken in the City of Greater Dandenong

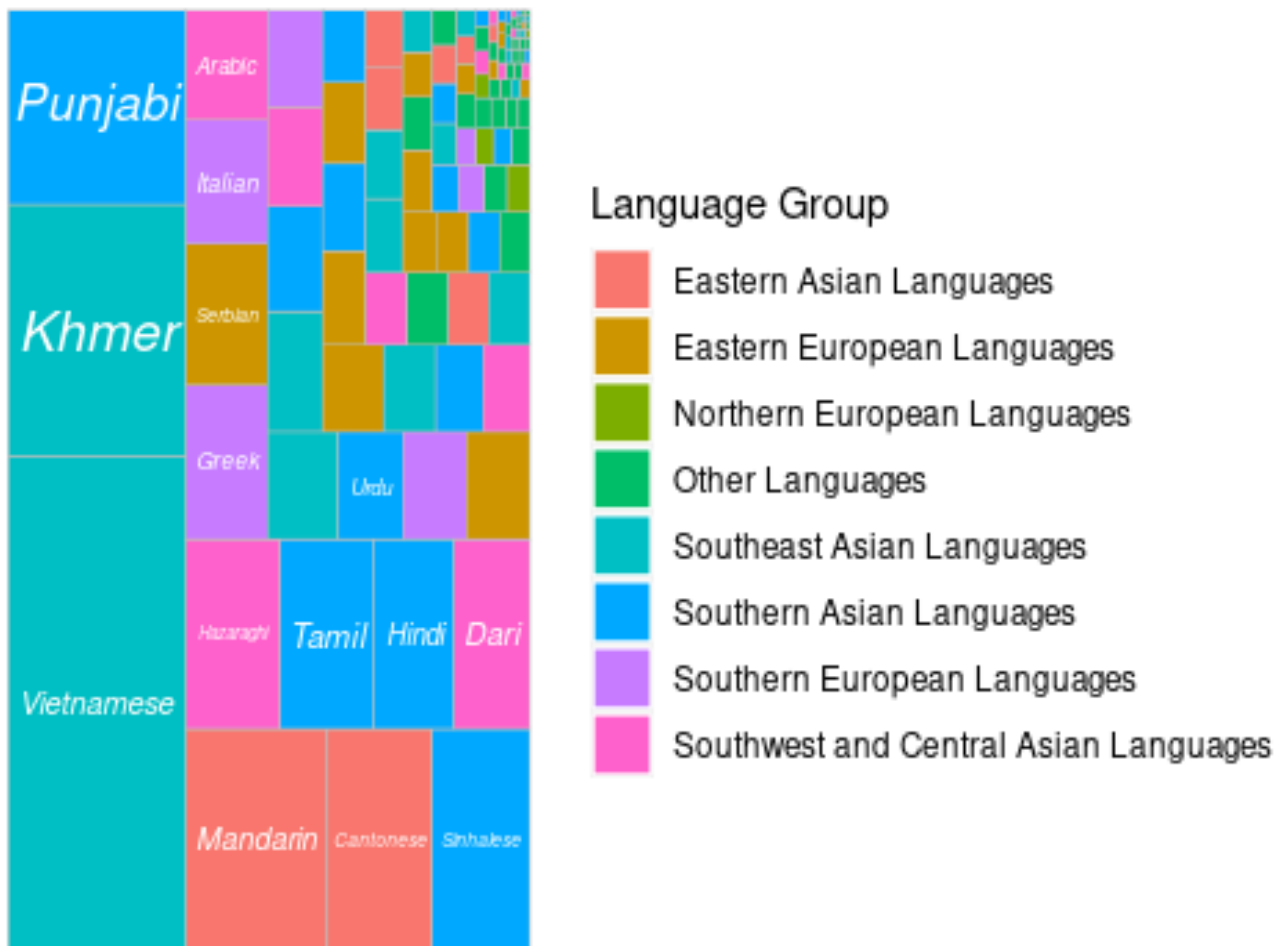


Fig 1

3.4 Create bar chart

I created a chart to show the level of education and English proficiency by language group, to look for any distinctive features.

3.4.1 Reorder levels for HEAP

The ordered factors that I created in the first file were dropped from the dataframe when I exported and then re-imported the data as a CSV file, as part of my recreation of the dataframe in inRStudio Cloud. I re-ordered some of these to ensure the correct display in plots.

```
# reorder levels
education <- c("Non_Secondary", "Secondary", "Higher") # create vector in correct order
```

```
education <- as.factor(education)
df$HEAP <- factor(df$HEAP, levels = education, ordered = TRUE)
levels(df$HEAP)

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

3.4.2 Create plot

```
# create bar chart
g21 <- mutate(group_by(df, HEAP, GroupName1DC), Prop = total / sum(total))
ggplot(g21) + geom_col(aes(Group Name1DC, 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")
```

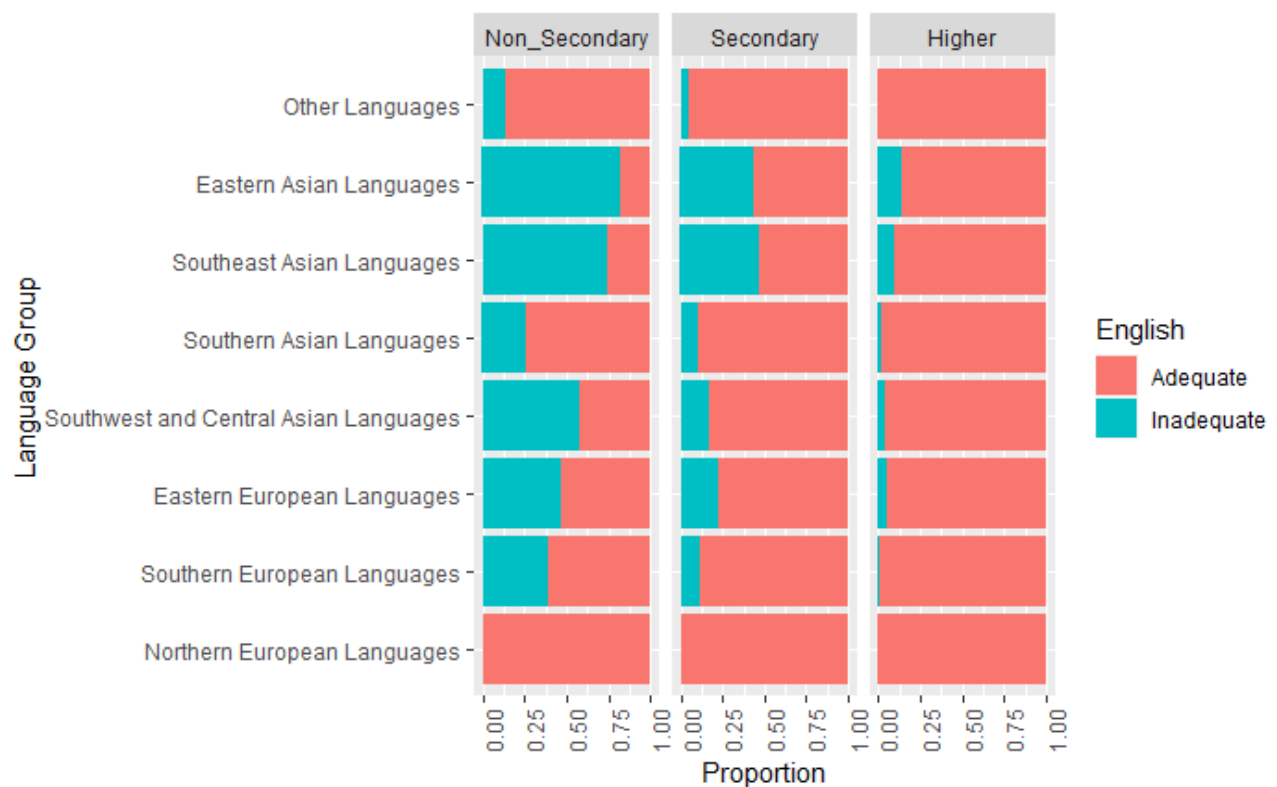


Fig 2

3.5 Create faceted scatterplots

I used faceted scatterplots to explore the relationship between language, education, English proficiency and gender within a sample language group. While this is the maximum number of variables I would normally include on one chart, this is a useful exploratory tool that provides a multi-faceted summary. I repeated this for every 1DC language group.

```
# create faceted scatterplots
df %>% filter(Group Name1DC == "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))
```



Fig 3

3.6 Explore socially advantaged/ disadvantaged groups

3.6.1 Create dataframe for disadvantaged groups

I created criteria for social advantage and disadvantage to explore the characteristics of these groups. While the choice of criteria is subjective and is not informed by any particular method, what I aim to show here is the ability to create criteria and then carry out analysis of a subset.

3.6.1.1 Combine year levels into two groups

I combined the single year 2016 into the 2006-2015 band as exploration showed no value in leaving this year as a standalone level.

```
#combine levels
```

```
df <- mutate(df, YARRP = fct_recode(YARRP, "2006-16" = "2016"))
df <- mutate(df, YARRP = fct_recode(YARRP, "2006-16" = "2006-15"))
head(df, 25)
```

##	X.1	X	LGA	LANP	HEAP	EETP	NEDD	
## 1	1	1	Greater_Dandenong	Vietnamese	Higher	Partial	Internet	
## 2	2	2	Greater_Dandenong	Vietnamese	Higher	Partial	Internet	
## 3	3	3	Greater_Dandenong	Vietnamese	Higher	Partial	Internet	
## 4	4	4	Greater_Dandenong	Vietnamese	Higher	Partial	Internet	
## 5	5	5	Greater_Dandenong	Vietnamese	Higher	Not_Engaged	Internet	
## 6	6	6	Greater_Dandenong	Vietnamese	Higher	Not_Engaged	Internet	
## 7	7	7	Greater_Dandenong	Vietnamese	Higher	Not_Engaged	Internet	
## 8	8	8	Greater_Dandenong	Vietnamese	Higher	Not_Engaged	Internet	
## 9	9	9	Greater_Dandenong	Vietnamese	Higher	Not_Engaged	No_Internet	
## 10	10	10	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 11	11	11	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 12	12	12	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 13	13	13	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 14	14	14	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 15	15	15	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 16	16	16	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 17	17	17	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 18	18	18	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 19	19	19	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 20	20	20	Greater_Dandenong	Vietnamese	Higher	Fully	Internet	
## 21	21	21	Greater_Dandenong	Vietnamese	Higher	Fully	No_Internet	
## 22	22	22	Greater_Dandenong	Vietnamese	Higher	Fully	No_Internet	
## 23	23	23	Greater_Dandenong	Vietnamese	Higher	Fully	No_Internet	
## 24	24	24	Greater_Dandenong	Vietnamese	Secondary	Partial	Internet	
## 25	25	25	Greater_Dandenong	Vietnamese	Secondary	Partial	Internet	
##	ENGP		BPLP		YARRP	SEXP	total	GroupCode1DC
## 1	Inadequate		Vietnam	Pre_2006	Male	5	6	
## 2	Inadequate		Vietnam	2006-16	Male	6	6	
## 3	Adequate		Vietnam	Pre_2006	Male	70	6	
## 4	Adequate		Vietnam	2006-16	Male	33	6	
## 5	Inadequate		Vietnam	Pre_2006	Male	12	6	
## 6	Inadequate		Vietnam	2006-16	Male	18	6	
## 7	Adequate		Vietnam	Pre_2006	Male	111	6	
## 8	Adequate		Vietnam	2006-16	Male	21	6	
## 9	Adequate		Vietnam	Pre_2006	Male	9	6	
## 10	Inadequate		Vietnam	Pre_2006	Male	19	6	
## 11	Inadequate		Vietnam	2006-16	Male	42	6	
## 12	Adequate		New Zealand	Pre_2006	Male	9	6	
## 13	Adequate		Thailand	Pre_2006	Male	7	6	

## 14	Adequate	Vietnam	Pre_2006	Male	466	6
## 15	Adequate	Vietnam	2006-16	Male	156	6
## 16	Adequate	Vietnam	2006-16	Male	4	6
## 17	Adequate	Indonesia	Pre_2006	Male	7	6
## 18	Adequate	Malaysia	Pre_2006	Male	6	6
## 19	Adequate	Philippines	Pre_2006	Male	4	6
## 20	Adequate	Hong Kong (SAR of China)	Pre_2006	Male	3	6
## 21	Inadequate	Vietnam	Pre_2006	Male	3	6
## 22	Adequate	Vietnam	Pre_2006	Male	10	6
## 23	Adequate	Vietnam	2006-16	Male	4	6
## 24	Inadequate	Cambodia	Pre_2006	Male	3	6
## 25	Inadequate	Vietnam	Pre_2006	Male	87	6
##	GroupCode2DC	LanguageCode4DC	GroupName1DC	GroupName2DC		
## 1	63	6302 Southeast Asian Languages	Mon-Khmer			
## 2	63	6302 Southeast Asian Languages	Mon-Khmer			
## 3	63	6302 Southeast Asian Languages	Mon-Khmer			
## 4	63	6302 Southeast Asian Languages	Mon-Khmer			
## 5	63	6302 Southeast Asian Languages	Mon-Khmer			
## 6	63	6302 Southeast Asian Languages	Mon-Khmer			
## 7	63	6302 Southeast Asian Languages	Mon-Khmer			
## 8	63	6302 Southeast Asian Languages	Mon-Khmer			
## 9	63	6302 Southeast Asian Languages	Mon-Khmer			
## 10	63	6302 Southeast Asian Languages	Mon-Khmer			
## 11	63	6302 Southeast Asian Languages	Mon-Khmer			
## 12	63	6302 Southeast Asian Languages	Mon-Khmer			
## 13	63	6302 Southeast Asian Languages	Mon-Khmer			
## 14	63	6302 Southeast Asian Languages	Mon-Khmer			
## 15	63	6302 Southeast Asian Languages	Mon-Khmer			
## 16	63	6302 Southeast Asian Languages	Mon-Khmer			
## 17	63	6302 Southeast Asian Languages	Mon-Khmer			
## 18	63	6302 Southeast Asian Languages	Mon-Khmer			
## 19	63	6302 Southeast Asian Languages	Mon-Khmer			
## 20	63	6302 Southeast Asian Languages	Mon-Khmer			
## 21	63	6302 Southeast Asian Languages	Mon-Khmer			
## 22	63	6302 Southeast Asian Languages	Mon-Khmer			
## 23	63	6302 Southeast Asian Languages	Mon-Khmer			
## 24	63	6302 Southeast Asian Languages	Mon-Khmer			
## 25	63	6302 Southeast Asian Languages	Mon-Khmer			

3.6.1.2 Reorder levels chronologically

```
# order Levels: YARRP
years2 <- c("Pre_2006", "2006-16") # create vector in correct order
years2 <- as.factor(years2)
df$YARRP <- factor(df$YARRP, levels = years2, ordered = TRUE)
levels(df$YARRP)

## [1] "Pre_2006" "2006-16"
```

3.6.1.3 Apply disadvantage criteria to create a subset

```
# create disadvantaged df
Disadvan <- df %>% 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))

## # A tibble: 80 x 4
## # Groups:   LANP, BPLP, SEXP [80]
##   LANP      BPLP      SEXP      n
##   <chr>    <chr>    <chr> <int>
## 1 Italian  Italy      Female 138
## 2 Vietnamese Vietnam Female 118
## 3 Khmer    Cambodia Female 100
## 4 Greek    Greece    Female  97
## 5 Italian  Italy      Male    79
## 6 Greek    Greece    Male    77
## 7 Vietnamese Vietnam Male    60
## 8 Khmer    Cambodia Male    55
## 9 Cantonese Vietnam Female  31
## 10 Mandarin China (excludes SARs and Taiwan) Female 22
## # ... with 70 more rows

# create a vector of the languages with min 30 count
DisadvanLang <- Disadvan %>% count(LANP, wt = total) %>% filter(n >= 30)

levels(Disadvan$YARRP)

## [1] "Pre_2006" "2006-16"
```

3.6.1.4 Create graph of languages with minimum count of 30

I removed languages with a total of less than 30 as sample sizes below this threshold are not statistically representative.

```
# create graph of just the langs with >= 30
Disadvan %>% 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")
```

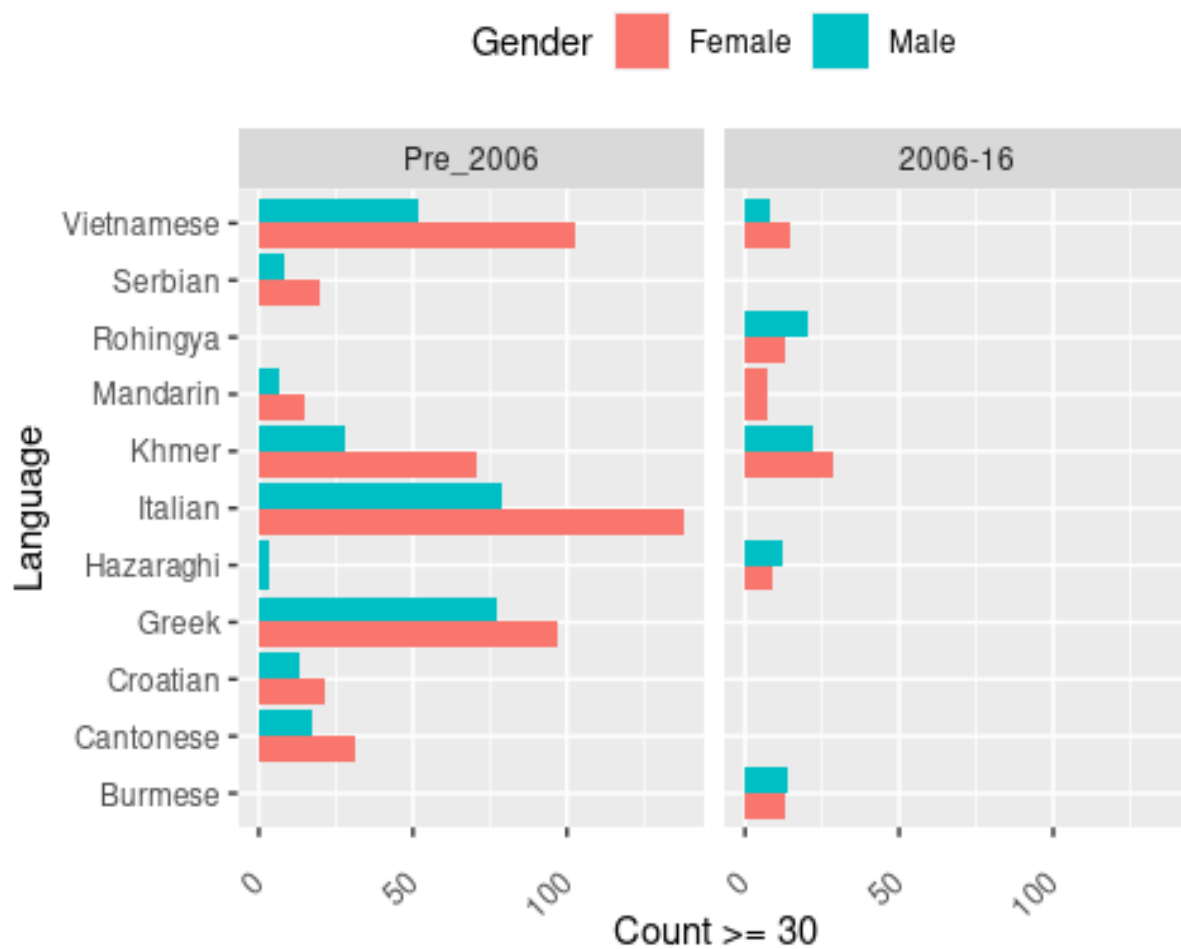


Fig 4

3.6.1.5 Exploration code

I created this code to explore the disadvantaged group by level; just insert the relevant variable.

```
# use this code to explore the counts by sex
Disadvan %>% count(SEXP, wt = total)

##      SEXP      n
## 1 Female  821
## 2   Male  506

# compute summary
Disadvanperc <- sum(Disadvan$total)/sum(df$total) * 100
round(Disadvanperc, 2)

## [1] 2.29

Disadvan %>% count(LANP, wt = total) %>% arrange(desc(n))
```



```
##          LANP      n
## 1      Italian 217
## 2      Greek 193
## 3    Vietnamese 187
## 4      Khmer 155
## 5    Cantonese 80
## 6      Serbian 73
## 7      Mandarin 43
## 8      Croatian 37
## 9      Burmese 33
## 10     Rohingya 33
## 11    Hazaraghi 31
## 12      Dari 27
## 13    Macedonian 21
## 14     Turkish 21
## 15     Bosnian 19
## 16     Pashto 18
## 17    Hungarian 16
## 18     Polish 16
## 19     Hakka 14
## 20     French 13
## 21     Min Nan 13
## 22     Spanish 12
## 23     Maltese 11
## 24    Mauritian Creole 9
## 25     Albanian 7
## 26      Tamil 5
## 27     Hindi 4
## 28      Thai 4
## 29     Arabic 3
## 30    Filipino 3
## 31     Romanian 3
## 32 Serbo-Croatian/Yugoslavian so described 3
## 33     Ukrainian 3
```

```
sum(Disadvan$total)
```

```
## [1] 1327
```

3.6.2 Create dataframe for advantaged groups

```
# create advantaged df
```

```
Advan <- df %>% filter(ENGP == "Adequate", NEDD == "Internet", EETP == "Fully",
HEAP == "Higher")
```

3.6.2.1 Compute sum and percentage

```
Advanperc_M <- 6063 / (6063 + 4053)
Advanperc_F <- 4053 / (6063 + 4053)
round(Advanperc_M, 2)
```

```
## [1] 0.6
```

```
round(Advanperc_F, 2)
```

```
## [1] 0.4
```

```
sum(df$total)
```

```
## [1] 57825
```

```
sum(Advan$total)
```

```
## [1] 10116
```

3.7 Plot advantaged groups

3.7.1 Create a vector of languages with minimum count of 30

```
AdvanLang <- Advan %>% count(LANP, wt = total) %>% filter(n >= 30) %>% arrange(desc(n))
```

AdvanLang # This computes 40 Languages

##	LANP	n
## 1	Punjabi	1305
## 2	Vietnamese	1252
## 3	Sinhalese	925
## 4	Mandarin	824
## 5	Hindi	676
## 6	Tamil	548
## 7	Khmer	468
## 8	Cantonese	389
## 9	Malayalam	303
## 10	Urdu	295
## 11	Bengali	221
## 12	Tagalog	207
## 13	Telugu	207
## 14	Serbian	201
## 15	Filipino	191
## 16	Gujarati	158
## 17	Arabic	150
## 18	Indonesian	114
## 19	Kannada	108
## 20	Bosnian	107
## 21	Dari	96
## 22	French	95
## 23	Pashto	70
## 24	Hazaraghi	69
## 25	Min Nan	58
## 26	Nepali	58
## 27	Polish	55
## 28	Thai	54
## 29	Swahili	53
## 30	Persian (excluding Dari)	52
## 31	Russian	46

## 32	Burmese	44
## 33	Korean	44
## 34	Croatian	39
## 35	Marathi	39
## 36	Romanian	38
## 37	Turkish	37
## 38	Malay	35
## 39	Greek	34
## 40	Samoan	30

3.7.3 Separate languages into groups of 10 for neater 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,]
```

3.7.4 Recode a long name

```
# recode long name Advan <- Advan %>% mutate(LANP = fct_recode(LANP, "Persian" = "Persian (excluding Dari)"))
TwenOneThir <- TwenOneThir %>% mutate(LANP = fct_recode(LANP, "Persian" = "Persian (excluding Dari)"))
```

3.7.5 Create plots for each language 'bin' group

```
# 1 - 10
a1 <- Advan %>% 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 <- Advan %>% 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 <- Advan %>% 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 <- Advan %>% 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")
```

3.7.6 Display as a panel

```
ggarrange(a1, a2, a3, a4, ncol=2, nrow=2, common.legend = TRUE, legend="bottom")
```

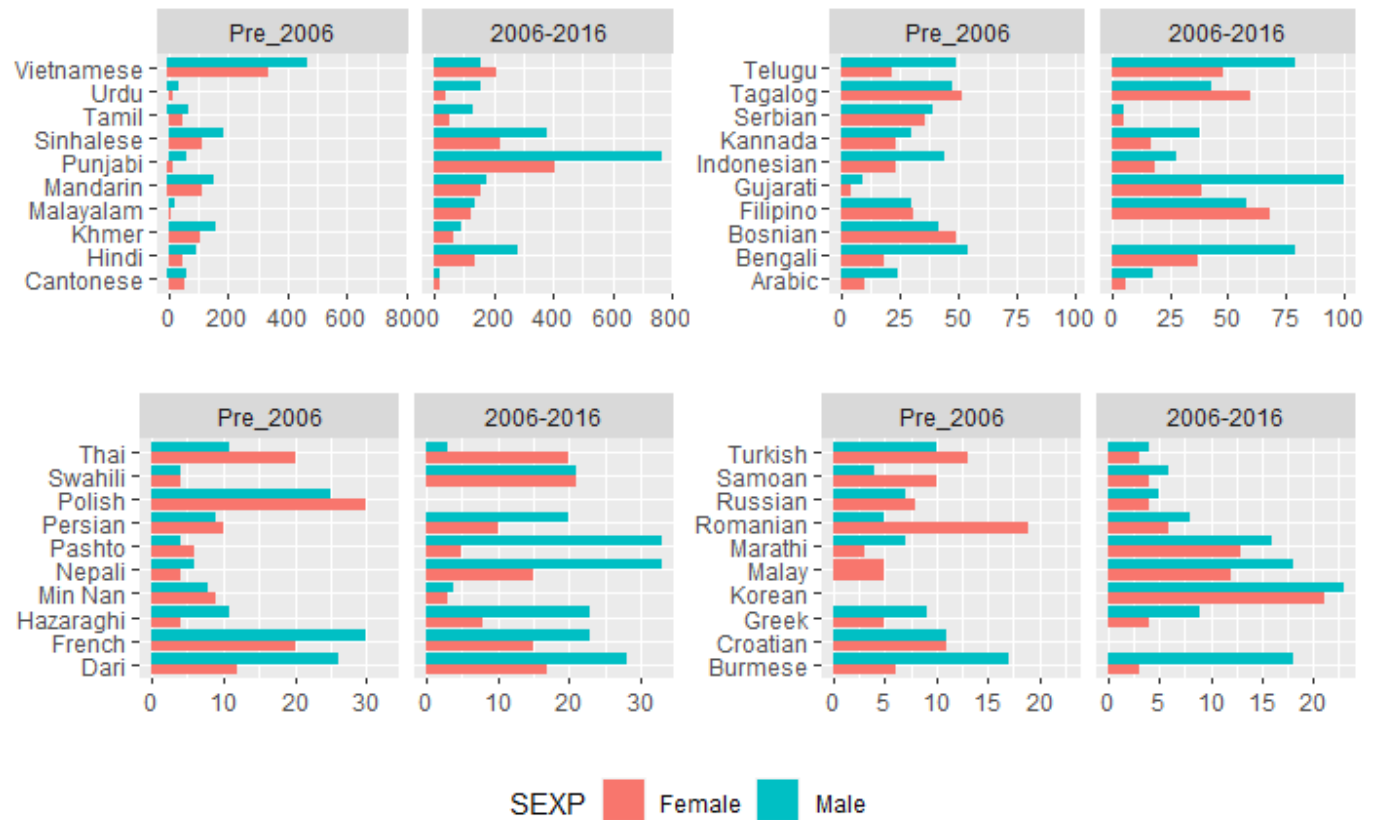


Fig 5