Take-Home Exercise 1: Creating Data Visualization that is enlightening and truthful

Liaw Ying Ting, Celin

2025-05-01

# 1. Overview

## 1.1 Setting the scene

A local online media company that publishes daily content on digital platforms is planning to release an article on demographic structures and distribution of Singapore in 2024.

## 1.2 The Task

Assuming the role of the graphical editor of the media company, you are tasked to prepare at most three data visualization for the article.

To accomplish the task, the data: *Singapore Residents by Planning Area/Subzone, Single Year of Age and Sex, June 2024* dataset shared by [Department of Statistics, Singapore(DOS)](https://www.singstat.gov.sg/find-data/search-by-theme/population/geographic-distribution/latest-data) is used.

## 1.3 Breaking down the question

Based on these information, the following questions will be answered:

* What is the proportion of males and females in Singapore and is gender-balance consistent across Singapore?
* What is the age distribution of Singapore’s population in 2024 and is there a decrease in birth rates over the years?
* Is the current number of working adults enough to support the number of dependents in Singapore?

# 2. Getting Started

## 2.1 Load packages

The following packages are loaded using the pacman::p\_load function:

* **tidyverse:** Core collection of R packages designed for data science
* **ggrepel:** An R package to provide geoms for gg plot2 to repel overlapping text labels
* **ggthemes:** An R package to provided extra themes, geoms and scales for ggplot 2
* **hrbrthemes:** an R packagae that provides typography-centric themes and theme componenets for ggplot2
* **patchwork:** to prepare composite figure created using ggplot2
* **dplyr:** An R package that provides a grammar of data manipulation, offering a set of functions for filtering, selecting, mutating, summarising and arranging data frames

pacman::p\_load(tidyverse, ggrepel,   
 ggthemes, hrbrthemes,  
 patchwork, dplyr)

package 'fontBitstreamVera' successfully unpacked and MD5 sums checked  
package 'fontLiberation' successfully unpacked and MD5 sums checked  
package 'extrafontdb' successfully unpacked and MD5 sums checked  
package 'Rttf2pt1' successfully unpacked and MD5 sums checked  
package 'fontquiver' successfully unpacked and MD5 sums checked  
package 'extrafont' successfully unpacked and MD5 sums checked  
package 'gdtools' successfully unpacked and MD5 sums checked  
package 'hrbrthemes' successfully unpacked and MD5 sums checked  
  
The downloaded binary packages are in  
 C:\Users\celin\AppData\Local\Temp\RtmpkT0xAs\downloaded\_packages

## 2.2 Import Data

The data that has been used has been mentioned in Chapter 1.2 above. This dataset will be imported as *resident\_data.*

resident\_data <- read\_csv("respopagesex2024.csv")

## 2.3 Data Wrangling

### 2.3.1 Check for duplicate entries

The data is being checked for any duplicate entries using any(duplicated()).

any(duplicated(resident\_data))

[1] FALSE

Since “FALSE” is returned, there are no duplicated entries

### 2.3.2 Check for missing values

The colSums function is being used to check for missing values.

colSums(is.na(resident\_data))

PA SZ Age Sex Pop Time   
 0 0 0 0 0 0

There are no missing values in any of the rows.

### 2.3.3 Filtering for relevant variables & renaming them

There are 60424 rows x 6 columns in the data, where the columns are labelled as:

* **PA** - Planning Area
* **SZ** - Subzone
* **Age** - From “0” to “89”, and those above 90 are labelled as “90\_and\_Over”
* **Sex** - Males or Females
* **Pop** - Resident Count (Has been rounded to the nearest 10)
* **Time** - Year (We will only be looking at 2024 data)

A new object called **resident\_data\_clean** is created where it will rename **PA, SZ** and **Pop** for easy reading by using the rename() function.

Those labelled as “90\_and\_Over” in the Age Column will be renamed as “90” using the ifelse() function for easier data plots.

Population and age will then be categorised as numerical data by using the as.numeric function.

For the Age column, it is slightly special as levels = as.character function in R sorts alphabetically by default and “10” will come before “2”. By using this function, R will be forced to treat the ages in true numeric order when plotting or tabulating them. A new column, **Age\_numeric**, was also created in the new object.

resident\_data\_clean <- resident\_data %>%  
 rename(  
 Planning\_Area = PA,  
 Subzone = SZ,  
 Population = Pop  
 ) %>%  
 # Convert Age to "90+" if it's "90\_and\_Over"  
 mutate(  
 Age = ifelse(Age == "90\_and\_Over", "90", Age),  
 Population = as.numeric(Population),  
 Age = factor(Age, levels = as.character(0:90)),  
 Age\_numeric = as.numeric(as.character(Age))   
 )

### 2.3.4 Grouping them into Regions for Easy Visualization

As there are too many planning areas, to make things easier to visualize, the planning areas have been further categorised into 5 areas: North, North-East, East, West and Central. This was done by using the categories on the [Urban Redevelopment Authority(URA)](https://eservice.ura.gov.sg/maps/?service=MP) website:

* **North**: Central Water Catchment, Lim Chu Kang, Mandai, Sembawang, Simpang, Sungei Kadut, Woodlands, Yishun
* **North-East**: Ang Mo Kio, Hougang, Punggol, Sengkang, Serangoon, Seletar
* **East**: Bedok, Changi, Changi Bay, Pasir Ris, Paya Lebar, Tampines
* **West**: Boon Lay, Bukit Batok, Bukit Panjang, Choa Chu Kang, Clementi, Jurong East, Jurong West, Pioneer, Tengah, Tuas, Western Water Cachement
* **Central**: Bishan, Bukit Merah, Bukit Timah, Central Area, Downtown Core, Marina East, Marina South, Museum, Newton, Novena, Orchard, Outram, River Valley, Rochor, Singapore River, Straits View, Tanglin
* Areas like **North-Eastern Islands** and **Western islands** were intentionally left out as they do not belong to any of the areas

**Steps:**

**Step 1:** A new list called **region\_map** was created to categorise them into the 5 categories

# 1. Step 1  
region\_map <- list(  
 North = c(  
 "Central Water Catchment", "Lim Chu Kang", "Mandai",  
 "Sembawang", "Simpang", "Sungei Kadut", "Woodlands", "Yishun"  
 ),  
 `North-East` = c(  
 "Ang Mo Kio", "Hougang", "Punggol", "Sengkang",   
 "Serangoon", "Seletar"  
 ),  
 East = c(  
 "Bedok", "Changi", "Changi Bay",   
 "Pasir Ris", "Paya Lebar", "Tampines"  
 ),  
 West = c(  
 "Boon Lay", "Bukit Batok", "Bukit Panjang", "Choa Chu Kang",  
 "Clementi", "Jurong East", "Jurong West", "Pioneer",   
 "Tengah", "Tuas", "Western Water Catchment"  
 ),  
 Central = c(  
 "Bishan", "Bukit Merah", "Bukit Timah", "Central Area",  
 "Downtown Core", "Marina East", "Marina South", "Museum",  
 "Newton", "Novena", "Orchard", "Outram", "River Valley",  
 "Rochor", "Singapore River", "Straits View", "Tanglin"  
 )  
)

**Step 2:** A new data object **resident\_regioned** was then created with their region matched to them and those who were not found are removed using filter() function.

# Step 2  
resident\_regioned <- resident\_data\_clean %>%  
 mutate(  
 Region = case\_when(  
 Planning\_Area %in% region\_map$North ~ "North",  
 Planning\_Area %in% region\_map$`North-East` ~ "North-East",  
 Planning\_Area %in% region\_map$East ~ "East",  
 Planning\_Area %in% region\_map$West ~ "West",  
 Planning\_Area %in% region\_map$Central ~ "Central",  
 TRUE ~ NA\_character\_  
 )  
 ) %>%  
 filter(!is.na(Region))

The data will then be grouped into their respective regions for the analysis.

### 2.3.5 Preview Processed Data

head(resident\_regioned)

# A tibble: 6 × 8  
 Planning\_Area Subzone Age Sex Population Time Age\_numeric Region  
 <chr> <chr> <fct> <chr> <dbl> <dbl> <dbl> <chr>   
1 Ang Mo Kio Ang Mo Kio Town… 0 Males 10 2024 0 North…  
2 Ang Mo Kio Ang Mo Kio Town… 0 Fema… 10 2024 0 North…  
3 Ang Mo Kio Ang Mo Kio Town… 1 Males 10 2024 1 North…  
4 Ang Mo Kio Ang Mo Kio Town… 1 Fema… 10 2024 1 North…  
5 Ang Mo Kio Ang Mo Kio Town… 2 Males 10 2024 2 North…  
6 Ang Mo Kio Ang Mo Kio Town… 2 Fema… 10 2024 2 North…

Now that data wrangling is complete, 3 key visualizations will be performed on this data set.

# 3. Data Visualization

## 3.1 Pie Chart By Regions

First, groups were being created using groupby() to group them by regions and sex (Eg. East Males and East Females) so that the total population of each group can be found using sum(). After that, .groups = “drop” will remove the groups as ggplot doesn’t use groupings to plot and some functions that “mutate” may not turn out as expected.

Next, ordered\_levels was created to define the stacking order for the pie slices in an anti-clockwise direction.

mutate() was used to add new columns to the pie\_all\_regions object.

ggplot() initializes a ggplot and the factors **y = Population** defines the height of each bar to be turned into pie slices and **fill = RegionSex** means the colour fill will be based on Sex.

geom\_col draws bars with heights corresponding to population with **width of 1** and **white borders between pie slices.**

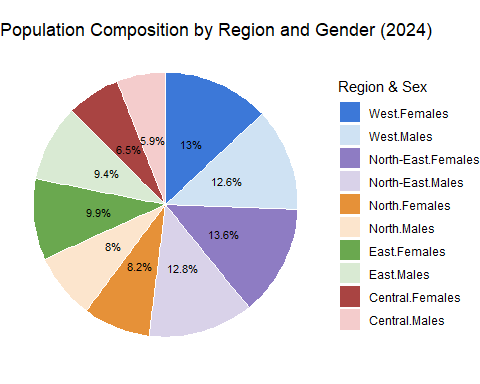
geom\_text was used to add percentage labels inside each pie slice with **size of 3** and **text colour black.**

coord\_polar(theta = “y”) converts stacked bar into a circular pie chart where the y-axis becomes the angle.

labs() help to add chart title and legend title.

scale\_fill\_manual was used to set custom colours for each Region-Sex.

### The Plot



### The Code

# Summarise population by Region and Sex  
pie\_all\_regions <- resident\_regioned %>%  
 group\_by(Region, Sex) %>%  
 summarise(Population = sum(Population), .groups = "drop")  
  
  
# Define the correct stacking order for pie slices  
ordered\_levels <- c(  
 "Central.Males", "Central.Females",  
 "East.Males", "East.Females",  
 "North.Males", "North.Females",  
 "North-East.Males", "North-East.Females",  
 "West.Males", "West.Females"  
)  
  
# Prepare the data  
pie\_all\_regions <- pie\_all\_regions %>%  
 mutate(  
 RegionSex = factor(paste(Region, Sex, sep = "."), levels = ordered\_levels),  
 label = paste0(round(Population / sum(Population) \* 100, 1), "%")  
 )  
  
# Plot: simple, clean pie with percentages inside  
ggplot(pie\_all\_regions, aes(x = "", y = Population, fill = RegionSex)) +  
 geom\_col(width = 1, color = "white") +  
 geom\_text(  
 aes(label = label),  
 position = position\_stack(vjust = 0.5),  
 size = 3,  
 color = "black"  
 ) +  
 coord\_polar(theta = "y") +  
 theme\_void(base\_size = 11) +  
 labs(  
 title = "Population Composition by Region and Gender (2024)",  
 fill = "Region & Sex"  
 ) +  
 scale\_fill\_manual(  
 values = c(  
 "Central.Males" = "#f4cccc",  
 "Central.Females" = "#a94442",  
 "East.Males" = "#d9ead3",  
 "East.Females" = "#6aa84f",  
 "North.Males" = "#fce5cd",  
 "North.Females" = "#e69138",  
 "North-East.Males" = "#d9d2e9",  
 "North-East.Females"= "#8e7cc3",  
 "West.Males" = "#cfe2f3",  
 "West.Females" = "#3c78d8"  
 ),  
 breaks = rev(ordered\_levels)  
 )

***Observations***: From the pie chart, it can be seen that the **proportion of males and females in each area is quite similar**, only differing by 1-2%. However, it can be seen from the pie chart that the proportion of **males are consistently lesser** that the proportion of females in each area. We can then conclude that there are **slightly less males than females in Singapore**. Overall, gender balance is consistent across Singapore.

Additionally, the **West and North-East Areas each made up 25%** of the total population, accounting for half the population in Singapore while the **North, Central and and East made up of the other half** of Singapore’s population.

## 3.2 Age-Sex Pyramid by Regions

For the age-sex pyramid, the data was then mutated using cut() and breaks() function to split them into Age categories: “0-9”, “10-19” and so on. In the cut() function, “right = FALSE” was used to exclude the upper bound of the bins so that they will consider 0-9 instead of 0-10 for each bin. A new column **Age\_bin** was created.

resident\_regioned <- resident\_regioned %>%  
 mutate(  
 Age\_bin = cut(  
 Age\_numeric,  
 breaks = c(seq(0, 90, by = 10), Inf),  
 labels = c(paste0(seq(0,80,by=10), "-", seq(9,89,by=10)), "90 and Above"),  
 right = FALSE  
 )  
 )

A new data object **pyramid\_data** was created to create the pyramid table. groupby() and summarise(Total = sum()) functions were used to group them by region(East, West etc), age group(Eg. 0-9, 10-19 etc) and sex(Males or females), before summing them up.

Value = ifelse(Sex == “Males, -Total, Total”) was then used to flip males to negative values so they will be plotted to the left, while the female values remain as positive.

Label = format(Total, big.mark = “,”) is used to convert the numbers from 52300 to 52,300. These labels will be shown beside the bar graph.

hjust = ifelse(Sex == “Males”, 1.1, -0.1) aligns the male label to the left of the graph and aligns the female label to the right of the graph.

pyramid\_data <- resident\_regioned %>%  
 group\_by(Region, Age\_bin, Sex) %>%  
 summarise(Total = sum(Population), .groups = "drop") %>%  
 mutate(  
 Value = ifelse(Sex == "Males", -Total, Total),  
 Label = format(Total, big.mark = ","),  
 hjust = ifelse(Sex == "Males", 1.1, -0.1) # push left or right based on sex  
 )

In this section, individual plots for each region will be created and put together with patchwork.

A reusable function called plot\_region\_pyramid will be created so that it can be called by its region name.

ggplot is then used next to plot the graph. aes( x = Value, y = Age\_bin, fill = Sex) uses x-axis as Values so that Males are negative and Females are positive, y-axis is using the Age\_bin which was created previously ranging from “0-9”, “10-19” etc and fill is coloured by the different Sex.

geom\_col(width = 0.8) makes the width slightly smaller than 1 so that they do not touch.

geom\_text(aes(label = Label, hjust = hjust), size = 2.8, color = “black”) is an aesthetic label that uses the pre-formatted functions in the code chunk before this.

scale\_x\_continuous() helps to adjust the labels of the axis to follow pre-defined values mentioned in the code.

coord\_cartesian() adjusts the actual values of the axis.

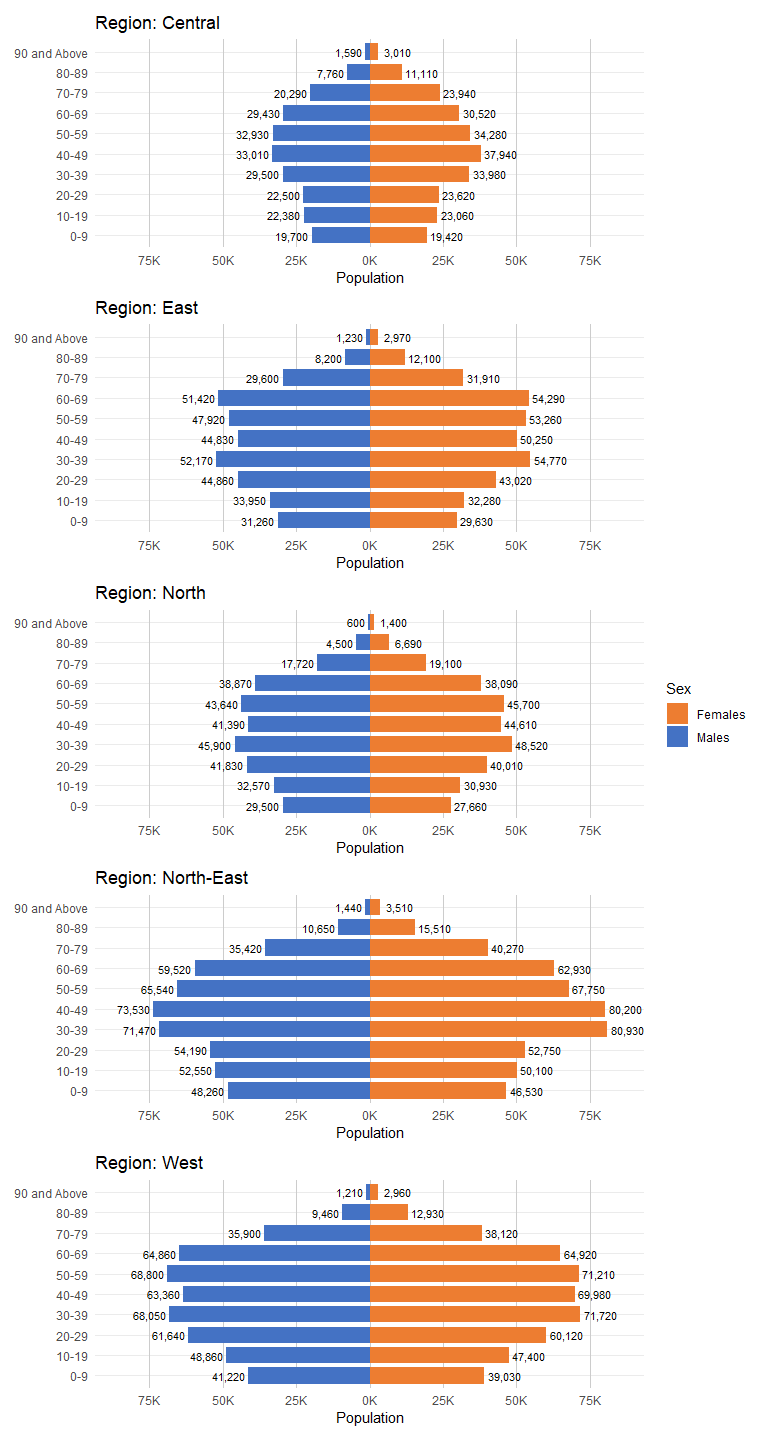
scale\_fill\_manual sets the fill colour for males and females.

labs() add a title for each pane with the region name.

theme\_minimal() uses the minimal theme.

(p1 / p2 / p3 / p4 / p5) + plot\_layout(guides = “collect”) helps to put the graphs together and show only 1 legend.

### The plot



### The code

plot\_region\_pyramid <- function(region\_name) {  
 ggplot(  
 data = filter(pyramid\_data, Region == region\_name),  
 aes(x = Value, y = Age\_bin, fill = Sex)  
 ) +  
 geom\_col(width = 0.8) +  
 geom\_text(aes(label = Label, hjust = hjust), size = 2.8, color = "black") +  
 scale\_x\_continuous(  
 breaks = c(-75000, -50000, -25000, 0, 25000, 50000, 75000),  
 labels = function(x) paste0(abs(x) / 1000, "K")  
 ) +  
 coord\_cartesian(xlim = c(-85000, 85000)) +  
 scale\_fill\_manual(values = c(Males = "#4472C4", Females = "#ED7D31")) +  
 labs(  
 title = paste("Region:", region\_name),  
 x = "Population",  
 y = "Age Group"  
 ) +  
 theme\_minimal(base\_size = 11) +  
 theme(  
 legend.position = "none",  
 strip.text = element\_text(face = "bold"),  
 axis.title.y = element\_blank(),  
 panel.grid.minor.x = element\_blank(),  
 panel.grid.major.x = element\_line(color = "grey80")  
 )  
}  
  
# Generate plots for each region  
p1 <- plot\_region\_pyramid("Central")  
p2 <- plot\_region\_pyramid("East")  
p3 <- plot\_region\_pyramid("North")  
p4 <- plot\_region\_pyramid("North-East")  
p5 <- plot\_region\_pyramid("West")  
  
# Arrange using patchwork (2 columns)  
(p1 / p2 / p3 / p4 / p5) + plot\_layout(guides = "collect")

***Observations:*** From the graphs, it can be seen that the population of **Ages** **30 to 69** are consistently larger than the population in the **Age Range** **0 to 29** and **Ages 70 and above** in most areas, resulting in a bulge in the middle of all the graphs in the five areas. This also shows that birth rates has decreased in the recent years.

Additionally, the population in the **Age range of 0 to 29** has consistently **more males than females** in all areas **except in the Central Area.** However, those in the **Age range 30 and above** has **more females than males** in all areas with those aged 60 to 69 in the North area being the exception.

## 3.3 Bar Graph of Dependency Ratio

The **dependency ratio** is a demographic measure that indicates the **ratio of dependents** (those typically not in the labour force) **to the working-age population**. **Dependents** are usually defined as those **under 15 and above 65 years old**.

With these information, the dependency ratio of the population in Singapore in the different regions will be calculated. This information will reflect the proportion of individuals who are likely to be economically dependent on others.

adjusted\_ageing\_ratio\_by\_region is the new object created to add new custom age groups.

mutate() is used to change the item to numeric so that <= and >= can be utilized to create the new custom **Age\_Group of 24 to 67** and **68 and above.**

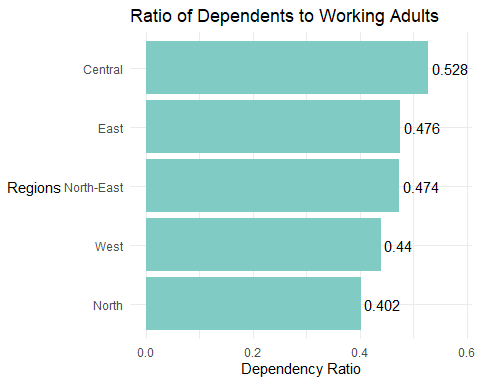
pivot\_wider() is used to transform data from long format to wide format so that the ageing ratio can be calculated correctly.

ggplot() is used to plot the horizontal bar chart, using reorder() to reorder the x-axis so that the bars are sorted by ratio.

geom\_col() is used to adjust the colour of the bars and geom\_text() is used to add labels to the bars, using **hjust** to push the labels to the right of each bar.

coord\_flip() is used to flip x and y axes and make it a horizontal bar chart.

### The Plot



### The Code

adjusted\_ageing\_ratio\_by\_region <- resident\_regioned %>%  
 mutate(Age\_numeric = as.numeric(as.character(Age))) %>%  
 filter(!is.na(Age\_numeric)) %>%  
 mutate(Age\_Group = case\_when(  
 Age\_numeric < 15 | Age\_numeric >= 65 ~ "< 15 or >=65",  
 Age\_numeric >= 15 & Age\_numeric <= 64 ~ "15-64",  
 TRUE ~ "Other"  
 )) %>%  
 filter(Age\_Group != "Other") %>%  
 group\_by(Region, Age\_Group) %>%  
 summarise(Population = sum(Population), .groups = "drop") %>%  
 pivot\_wider(names\_from = Age\_Group, values\_from = Population) %>%  
 mutate(Adjusted\_Ageing\_Ratio = `< 15 or >=65` / `15-64`)  
  
ggplot(adjusted\_ageing\_ratio\_by\_region, aes(x = reorder(Region, Adjusted\_Ageing\_Ratio), y = Adjusted\_Ageing\_Ratio)) +  
 geom\_col(fill = "#80CBC4") +  
 geom\_text(aes(label = round(Adjusted\_Ageing\_Ratio, 3)), hjust = -0.1) +  
 expand\_limits(y = max(adjusted\_ageing\_ratio\_by\_region$Adjusted\_Ageing\_Ratio) \* 1.1) +  
 labs(  
 title = "Ratio of Dependents to Working Adults",  
 x = "Regions",  
 y = "Dependency Ratio"  
 ) +  
 coord\_flip() +  
 theme\_minimal(base\_size = 11) +  
 theme(axis.title.y = element\_text(angle = 0, vjust = 0.5))

***Observations:*** From the bar graphs above, it can be seen that the **dependency ratio** (Population aged<15 and >65 / Population aged 15-65) **ranges from 0.402 to 0.528**. A ratio of 0.4 means that for every 2.5 working adults, there is 1 dependent, and a ratio of 0.528 means that for every 1.9 working adult, there is 1 dependent.

This **average** out to about **1 dependent to 2 working adults** in Singapore, showing a healthy demographic structure.

# 4. Conclusion

In this exercise, data cleaning and visualization of data was performed to examine the relationship between Age, Gender, Region and Population in Singapore. The key findings are:

* There is **gender-balance consistency across Singapore** with **slightly less males** than females **in all Regions** of Singapore
* There are **more males than females** in the **Age Range 0 to 29**, except in the Central Area
* There is consistently **more population in the Age Range 30 to 69** as compared to **Age Range 0 to 29**, showing a decrease in birth rate in Singapore
* There is a good average ratio of **1 dependent to 2 working adults** in Singapore, showing a healthy demographic

To conclude, Singapore has a balanced gender ratio with healthy dependency structure across the regions. However, a smaller proportion of young residents means a declining birth rate. If this trend continues, it may lead to a rising dependency in the future and policies should be adjusted for ageing related needs. Overall, current demographics reflect stability but call for forward looking strategies.

With relation to the codes used for analysis, **ggplot2** is a powerful package for exploratory data analysis when paired with **dplyr** which allows mutation of the data to usable formats. The original dataset consists of many areas which makes it difficult to find a direction and thus was further categorised into subgroups using the dplyr functions. Additional functions like **geom\_col, geom\_text, coord\_flip, coord\_cartesian and coor\_polar** were further utilised to create the visualizations for visual analysis.

Overall, this task was helpful to practice how to use R to mutate data and to present data effectively, conveying a clear message for the results obtained.

# 5. Key References

* [R for Visual Analytics](https://r4va.netlify.app/)
* [R for Data Science(2e)](https://r4ds.hadley.nz/)
* [ISSS608 AY2024-25 April Term](https://isss608-ay2024-25apr.netlify.app/)