Take-home Exercise 1

Ng Jin Yao

2025-05-01

# Overview

## 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**.

## The task

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

The data should be processed by using appropriate **tidyverse** family of packages and the data visualisation must be prepared using **ggplot2** and its extensions.

# Getting started

## Loading packages

The following R packages are loaded using pacman::p\_load() on top of the tidyverse packages:

* ggrepel: an R package provides geoms for ggplot2 to repel overlapping text labels.
* ggthemes: an R package provides some extra themes, geoms, and scales for ‘ggplot2’.
* patchwork: an R package for preparing composite figure created using ggplot2.
* dplyr: for data manipulation with the tidyverse package
* scales: for formatting and scaling data plots
* xml12: for working with HTML
* sf: for handling GIS file formats

The sf and xml12 packages were used as a dataset is extracted from a GEOJSON file later on.

pacman::p\_load(tidyverse, ggplot2, ggrepel, patchwork,   
 ggthemes,dplyr, xml12, sf, scales)

## Import data

The dataset “[Singapore Residents by Planning Area / Subzone, Single Year of Age and Sex, June 2024](https://www.singstat.gov.sg/-/media/files/find_data/population/statistical_tables/respopagesexfa2024.ashx)” retrieved from the [Department of Statistics, Singapore (DOS)](https://www.singstat.gov.sg/) will be used.

We will be importing this dataset as “sgResData24”

sgResData24 <- read\_csv("data/respopagesex2024.csv")

## Data pre-processing

We first take a look at the data and check for:

1. Any missing/NULL values or entries
2. If any column data types need adjustments
3. If any new columns need to be introduced to be used for visualisations
4. Any duplicate entries

## Preview of the dataset

The dim() and head() functions to view the dimensions and preview the first few rows of the dataset using the code below:

dim(sgResData24)

[1] 60424 6

head(sgResData24)

# A tibble: 6 × 6  
 PA SZ Age Sex Pop Time  
 <chr> <chr> <chr> <chr> <dbl> <dbl>  
1 Ang Mo Kio Ang Mo Kio Town Centre 0 Males 10 2024  
2 Ang Mo Kio Ang Mo Kio Town Centre 0 Females 10 2024  
3 Ang Mo Kio Ang Mo Kio Town Centre 1 Males 10 2024  
4 Ang Mo Kio Ang Mo Kio Town Centre 1 Females 10 2024  
5 Ang Mo Kio Ang Mo Kio Town Centre 2 Males 10 2024  
6 Ang Mo Kio Ang Mo Kio Town Centre 2 Females 10 2024

## Descriptive Statistics

The summary() function is used to view the basic descriptive statistics for each column, including the minimum, maximum, mean and quartiles.

summary(sgResData24)

PA SZ Age Sex   
 Length:60424 Length:60424 Length:60424 Length:60424   
 Class :character Class :character Class :character Class :character   
 Mode :character Mode :character Mode :character Mode :character   
   
   
   
 Pop Time   
 Min. : 0.0 Min. :2024   
 1st Qu.: 0.0 1st Qu.:2024   
 Median : 20.0 Median :2024   
 Mean : 69.4 Mean :2024   
 3rd Qu.: 90.0 3rd Qu.:2024   
 Max. :1180.0 Max. :2024

## Checking for duplicate values

The distinct() function is used to check if there are any duplicate values. The distinct() function returns unique rows.

distinct(sgResData24)

# A tibble: 60,424 × 6  
 PA SZ Age Sex Pop Time  
 <chr> <chr> <chr> <chr> <dbl> <dbl>  
 1 Ang Mo Kio Ang Mo Kio Town Centre 0 Males 10 2024  
 2 Ang Mo Kio Ang Mo Kio Town Centre 0 Females 10 2024  
 3 Ang Mo Kio Ang Mo Kio Town Centre 1 Males 10 2024  
 4 Ang Mo Kio Ang Mo Kio Town Centre 1 Females 10 2024  
 5 Ang Mo Kio Ang Mo Kio Town Centre 2 Males 10 2024  
 6 Ang Mo Kio Ang Mo Kio Town Centre 2 Females 10 2024  
 7 Ang Mo Kio Ang Mo Kio Town Centre 3 Males 10 2024  
 8 Ang Mo Kio Ang Mo Kio Town Centre 3 Females 10 2024  
 9 Ang Mo Kio Ang Mo Kio Town Centre 4 Males 30 2024  
10 Ang Mo Kio Ang Mo Kio Town Centre 4 Females 10 2024  
# ℹ 60,414 more rows

From the output of distinct() there are 60,424 x 6 rows returned, showing that there are no duplicated rows in the dataset.

Using the count() function, we can view the unique values/names from each column

## PA Column Count

count(sgResData24,PA)

# A tibble: 55 × 2  
 PA n  
 <chr> <int>  
 1 Ang Mo Kio 2184  
 2 Bedok 1456  
 3 Bishan 546  
 4 Boon Lay 728  
 5 Bukit Batok 1638  
 6 Bukit Merah 3094  
 7 Bukit Panjang 1274  
 8 Bukit Timah 1456  
 9 Central Water Catchment 182  
10 Changi 546  
# ℹ 45 more rows

There are 55 unique planning areas

## SZ Column Count

count(sgResData24,SZ)

# A tibble: 332 × 2  
 SZ n  
 <chr> <int>  
 1 Admiralty 182  
 2 Airport Road 182  
 3 Alexandra Hill 182  
 4 Alexandra North 182  
 5 Aljunied 182  
 6 Anak Bukit 182  
 7 Anchorvale 182  
 8 Ang Mo Kio Town Centre 182  
 9 Anson 182  
10 Bahar 182  
# ℹ 322 more rows

There are 332 unique subzones

## AG Column Count

count(sgResData24,Age)

# A tibble: 91 × 2  
 Age n  
 <chr> <int>  
 1 0 664  
 2 1 664  
 3 10 664  
 4 11 664  
 5 12 664  
 6 13 664  
 7 14 664  
 8 15 664  
 9 16 664  
10 17 664  
# ℹ 81 more rows

## Sex Column Count

count(sgResData24,Sex)

# A tibble: 2 × 2  
 Sex n  
 <chr> <int>  
1 Females 30212  
2 Males 30212

There are only 2 Sex groups - Male and Female

## Time Column Count

count(sgResData24,Time)

# A tibble: 1 × 2  
 Time n  
 <dbl> <int>  
1 2024 60424

There is only 1 unique group for time, the year 2024, for all the rows in the dataset

|  |
| --- |
| 🧐Observations and notes for this dataset |
| * There are 60,424 rows and 6 columns of data. * There are no duplicate rows/ values * Column Definition and Unique Values Count 6 Columns (5 Group and 1 Count):   + **PA** (chr type) - Planning Area (55 unique)   + **SZ** (chr type) - Subzone (332 unique)   + **Age** (chr type) - Age of Residents (Values: 1 to 89, 90\_and\_Over)   + **Sex** (chr type) - Sex (2 unique)   + **Time** (dbl type) - Time/ Period (1 unique)   + **Pop** (dbl type) - Resident Count |

## Transforming and Aggregating Age Column

Age column is a chr type column (with “90\_and\_Over” as one of the values in this column). From the Age column create 2 new columns so that we can use them for our data visualisations later on:

* Age\_num - introduced column that transforms the data type of the Age column from chr type to dbl type. Additionally, “90\_and\_Over” is converted to the value 90.
* Age\_group - aggregating the age groups of the population together using 10 year intervals

This is performed using the mutate() and case\_when() functions.

sgResData24 <- sgResData24 %>%  
 mutate(  
 # Convert age to numeric, handle "90\_and\_Over"  
 Age\_num = case\_when(  
 Age == "90\_and\_Over" ~ 90,  
 TRUE ~ suppressWarnings(as.numeric(Age)) # Avoid warnings from "90\_and\_Over"  
 ),  
   
 # Group into age bands  
 Age\_group = case\_when(  
 Age\_num >= 0 & Age\_num <= 9 ~ "0-9",  
 Age\_num >= 10 & Age\_num <= 19 ~ "10-19",  
 Age\_num >= 20 & Age\_num <= 29 ~ "20-29",  
 Age\_num >= 30 & Age\_num <= 39 ~ "30-39",  
 Age\_num >= 40 & Age\_num <= 49 ~ "40-49",  
 Age\_num >= 50 & Age\_num <= 59 ~ "50-59",  
 Age\_num >= 60 & Age\_num <= 69 ~ "60-69",  
 Age\_num >= 70 & Age\_num <= 79 ~ "70-79",  
 Age\_num >= 80 & Age\_num <= 89 ~ "80-89",  
 Age\_num >= 90 ~ "90+",  
 TRUE ~ NA\_character\_  
 )  
 )

We can then check our adjusted dataset using the print() function below:

print(sgResData24)

# A tibble: 60,424 × 8  
 PA SZ Age Sex Pop Time Age\_num Age\_group  
 <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <chr>   
 1 Ang Mo Kio Ang Mo Kio Town Centre 0 Males 10 2024 0 0-9   
 2 Ang Mo Kio Ang Mo Kio Town Centre 0 Females 10 2024 0 0-9   
 3 Ang Mo Kio Ang Mo Kio Town Centre 1 Males 10 2024 1 0-9   
 4 Ang Mo Kio Ang Mo Kio Town Centre 1 Females 10 2024 1 0-9   
 5 Ang Mo Kio Ang Mo Kio Town Centre 2 Males 10 2024 2 0-9   
 6 Ang Mo Kio Ang Mo Kio Town Centre 2 Females 10 2024 2 0-9   
 7 Ang Mo Kio Ang Mo Kio Town Centre 3 Males 10 2024 3 0-9   
 8 Ang Mo Kio Ang Mo Kio Town Centre 3 Females 10 2024 3 0-9   
 9 Ang Mo Kio Ang Mo Kio Town Centre 4 Males 30 2024 4 0-9   
10 Ang Mo Kio Ang Mo Kio Town Centre 4 Females 10 2024 4 0-9   
# ℹ 60,414 more rows

## Aggregate Planning Area (PA Column) into NSEW & Central Regions of Singapore

Referencing the URA 2019 masterplan map, we can group the Planning Areas into the different North, South, East, West and Central regions of Singapore.

We reference this from the GEOJSON data obtained from this data.gov.sg website [link](https://data.gov.sg/collections/2104/view).

### Extracting the Town(PA) and Region data from the GeoJSON file

The **sf** package is used here to help extract the spatial vector data from the GEOJSON data to extract the Town and Regions of Singapore.

#Install required packages if not already installed  
library(sf)  
library(xml2)  
library(dplyr)  
  
# Load the GeoJSON file  
geo\_data <- st\_read("data/MasterPlan2019PlanningAreaBoundaryNoSea.geojson")

Reading layer `MasterPlan2019PlanningAreaBoundaryNoSea' from data source   
 `C:\NJYao\ISSS608-VAA\Take-home\_Ex\Take-home\_Ex01\data\MasterPlan2019PlanningAreaBoundaryNoSea.geojson'   
 using driver `GeoJSON'  
Simple feature collection with 55 features and 2 fields  
Geometry type: MULTIPOLYGON  
Dimension: XY  
Bounding box: xmin: 103.6057 ymin: 1.158699 xmax: 104.0885 ymax: 1.470775  
Geodetic CRS: WGS 84

# Function to parse HTML and extract PLN\_AREA\_N and REGION\_N  
extract\_info <- function(html\_str) {  
 doc <- read\_html(html\_str)  
 rows <- xml\_find\_all(doc, ".//tr")  
  
 # Loop through rows and extract key-value pairs  
 data <- lapply(rows, function(row) {  
 th <- xml\_text(xml\_find\_first(row, ".//th"))  
 td <- xml\_text(xml\_find\_first(row, ".//td"))  
 if (!is.na(th) && !is.na(td)) {  
 return(setNames(list(td), th))  
 } else {  
 return(NULL)  
 }  
 })  
  
 # Combine and extract specific fields  
 info <- do.call(c, data)  
 list(  
 Town = info[["PLN\_AREA\_N"]],  
 Region = info[["REGION\_N"]]  
 )  
}  
  
  
# Apply the extraction function to each row  
info\_list <- lapply(geo\_data$Description, extract\_info)  
  
# Combine results into a data frame  
info\_df <- bind\_rows(info\_list) %>% distinct() %>% arrange(Region, Town)  
  
# View result  
print(info\_df)

# A tibble: 55 × 2  
 Town Region   
 <chr> <chr>   
 1 BISHAN CENTRAL REGION  
 2 BUKIT MERAH CENTRAL REGION  
 3 BUKIT TIMAH CENTRAL REGION  
 4 DOWNTOWN CORE CENTRAL REGION  
 5 GEYLANG CENTRAL REGION  
 6 KALLANG CENTRAL REGION  
 7 MARINA EAST CENTRAL REGION  
 8 MARINA SOUTH CENTRAL REGION  
 9 MARINE PARADE CENTRAL REGION  
10 MUSEUM CENTRAL REGION  
# ℹ 45 more rows

From the output of the above print(info\_df), we have a 55x2 table of “Town” and “Region” which we can then use for joining with our sgResData24 dataset.

To prepare our data for joining we first transform our sgResData24 dataset’s “PA” column data to uppercase to match the info\_df dataset “Town” data for joining later

sgResData24 %>% mutate(PA = toupper(PA))

# A tibble: 60,424 × 8  
 PA SZ Age Sex Pop Time Age\_num Age\_group  
 <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <chr>   
 1 ANG MO KIO Ang Mo Kio Town Centre 0 Males 10 2024 0 0-9   
 2 ANG MO KIO Ang Mo Kio Town Centre 0 Females 10 2024 0 0-9   
 3 ANG MO KIO Ang Mo Kio Town Centre 1 Males 10 2024 1 0-9   
 4 ANG MO KIO Ang Mo Kio Town Centre 1 Females 10 2024 1 0-9   
 5 ANG MO KIO Ang Mo Kio Town Centre 2 Males 10 2024 2 0-9   
 6 ANG MO KIO Ang Mo Kio Town Centre 2 Females 10 2024 2 0-9   
 7 ANG MO KIO Ang Mo Kio Town Centre 3 Males 10 2024 3 0-9   
 8 ANG MO KIO Ang Mo Kio Town Centre 3 Females 10 2024 3 0-9   
 9 ANG MO KIO Ang Mo Kio Town Centre 4 Males 30 2024 4 0-9   
10 ANG MO KIO Ang Mo Kio Town Centre 4 Females 10 2024 4 0-9   
# ℹ 60,414 more rows

Additionally, we rename the info\_df dataset’s “Town” column into “PA” using the **mutate()** function.

We can then perform a **left\_join** of the sgResData24 “PA” column with the info\_df dataset’s “PA” column to introduce the Region Column into sgResData24 dataset.

library(dplyr)  
  
# Rename Town to PA in the region info dataframe  
region\_info <- info\_df %>% rename(PA = Town)  
  
# left join sgResData24 with region\_info to get Region column  
sgResData24 <- sgResData24 %>% mutate(PA = toupper(PA)) %>%  
 left\_join(region\_info, by = "PA")  
  
print(sgResData24)

# A tibble: 60,424 × 9  
 PA SZ Age Sex Pop Time Age\_num Age\_group Region  
 <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <chr> <chr>   
 1 ANG MO KIO Ang Mo Kio Town … 0 Males 10 2024 0 0-9 NORTH…  
 2 ANG MO KIO Ang Mo Kio Town … 0 Fema… 10 2024 0 0-9 NORTH…  
 3 ANG MO KIO Ang Mo Kio Town … 1 Males 10 2024 1 0-9 NORTH…  
 4 ANG MO KIO Ang Mo Kio Town … 1 Fema… 10 2024 1 0-9 NORTH…  
 5 ANG MO KIO Ang Mo Kio Town … 2 Males 10 2024 2 0-9 NORTH…  
 6 ANG MO KIO Ang Mo Kio Town … 2 Fema… 10 2024 2 0-9 NORTH…  
 7 ANG MO KIO Ang Mo Kio Town … 3 Males 10 2024 3 0-9 NORTH…  
 8 ANG MO KIO Ang Mo Kio Town … 3 Fema… 10 2024 3 0-9 NORTH…  
 9 ANG MO KIO Ang Mo Kio Town … 4 Males 30 2024 4 0-9 NORTH…  
10 ANG MO KIO Ang Mo Kio Town … 4 Fema… 10 2024 4 0-9 NORTH…  
# ℹ 60,414 more rows

The output of print(sgResData24) shows the addition of the “Region” column.

We can perform the checks performed above to verify that the new columns introduced do not have any issues before proceeding with the visualisations.

## Preview of the dataset

The dim() and head() functions to view the dimensions and preview the first few rows of the dataset using the code below:

dim(sgResData24)

[1] 60424 9

head(sgResData24)

# A tibble: 6 × 9  
 PA SZ Age Sex Pop Time Age\_num Age\_group Region  
 <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <chr> <chr>   
1 ANG MO KIO Ang Mo Kio Town C… 0 Males 10 2024 0 0-9 NORTH…  
2 ANG MO KIO Ang Mo Kio Town C… 0 Fema… 10 2024 0 0-9 NORTH…  
3 ANG MO KIO Ang Mo Kio Town C… 1 Males 10 2024 1 0-9 NORTH…  
4 ANG MO KIO Ang Mo Kio Town C… 1 Fema… 10 2024 1 0-9 NORTH…  
5 ANG MO KIO Ang Mo Kio Town C… 2 Males 10 2024 2 0-9 NORTH…  
6 ANG MO KIO Ang Mo Kio Town C… 2 Fema… 10 2024 2 0-9 NORTH…

## Descriptive Statistics

The summary() function is used to view the basic descriptive statistics for each column, including the minimum, maximum, mean and quartiles.

summary(sgResData24)

PA SZ Age Sex   
 Length:60424 Length:60424 Length:60424 Length:60424   
 Class :character Class :character Class :character Class :character   
 Mode :character Mode :character Mode :character Mode :character   
   
   
   
 Pop Time Age\_num Age\_group   
 Min. : 0.0 Min. :2024 Min. : 0 Length:60424   
 1st Qu.: 0.0 1st Qu.:2024 1st Qu.:22 Class :character   
 Median : 20.0 Median :2024 Median :45 Mode :character   
 Mean : 69.4 Mean :2024 Mean :45   
 3rd Qu.: 90.0 3rd Qu.:2024 3rd Qu.:68   
 Max. :1180.0 Max. :2024 Max. :90   
 Region   
 Length:60424   
 Class :character   
 Mode :character

## Checking for duplicate values

The distinct() function is used to check if there are any duplicate values. The distinct() function returns unique rows.

distinct(sgResData24)

# A tibble: 60,424 × 9  
 PA SZ Age Sex Pop Time Age\_num Age\_group Region  
 <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <chr> <chr>   
 1 ANG MO KIO Ang Mo Kio Town … 0 Males 10 2024 0 0-9 NORTH…  
 2 ANG MO KIO Ang Mo Kio Town … 0 Fema… 10 2024 0 0-9 NORTH…  
 3 ANG MO KIO Ang Mo Kio Town … 1 Males 10 2024 1 0-9 NORTH…  
 4 ANG MO KIO Ang Mo Kio Town … 1 Fema… 10 2024 1 0-9 NORTH…  
 5 ANG MO KIO Ang Mo Kio Town … 2 Males 10 2024 2 0-9 NORTH…  
 6 ANG MO KIO Ang Mo Kio Town … 2 Fema… 10 2024 2 0-9 NORTH…  
 7 ANG MO KIO Ang Mo Kio Town … 3 Males 10 2024 3 0-9 NORTH…  
 8 ANG MO KIO Ang Mo Kio Town … 3 Fema… 10 2024 3 0-9 NORTH…  
 9 ANG MO KIO Ang Mo Kio Town … 4 Males 30 2024 4 0-9 NORTH…  
10 ANG MO KIO Ang Mo Kio Town … 4 Fema… 10 2024 4 0-9 NORTH…  
# ℹ 60,414 more rows

# Data Visualisation

## Population Pyramid Plot Visualisation

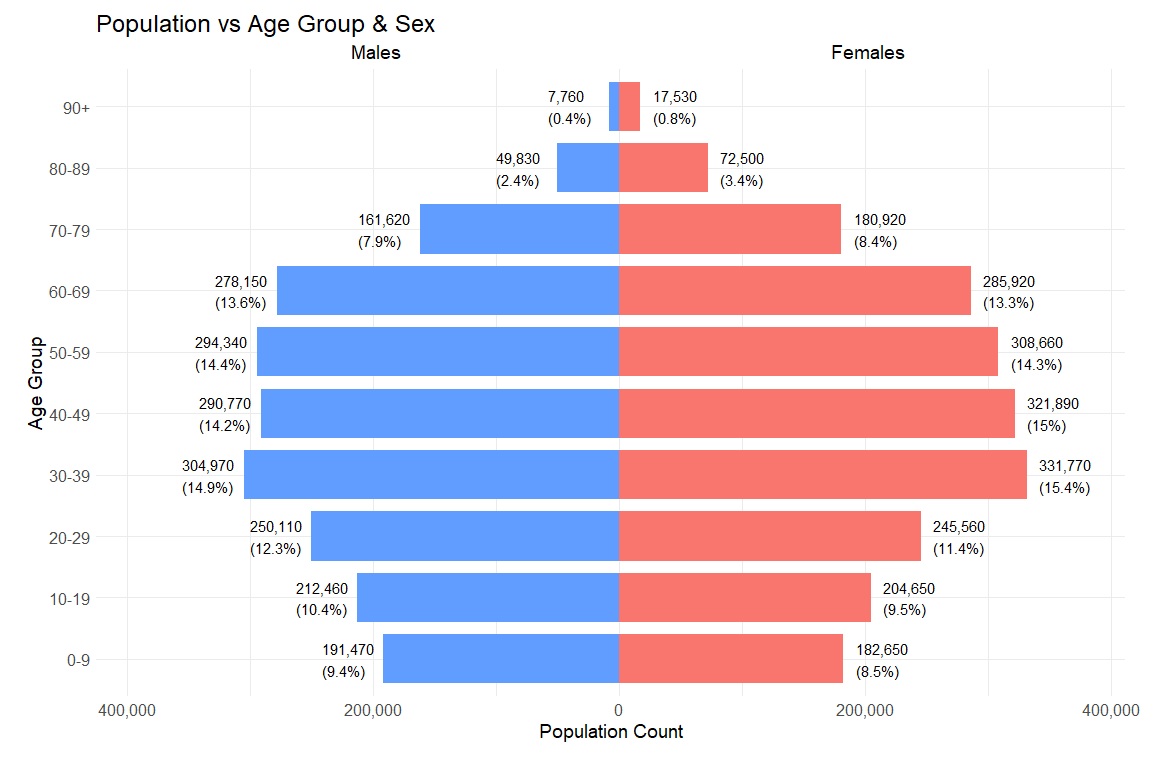
Plotting Age Group against the Population Count using a population pyramid

The **ggplot** and **dplyr** library will be used for this visualisation.

We first calculate the total population by summing the Pop and calculate other the values that will be used for labelling values for each group in the population pyramid using mutate(), summarise() and group\_by() functions. Then we set the levels for the Age\_groups so that the axis labels will be ordered in ascending order. The Pop values for the Males then have to be flipped into -ve values by using the **mutate()** and **ifelse()** functions.

The population pyramid is then plot using ggplot, with the respective packages to adjust the aesthetic aspects of the plot such as defining the fill to show segregate the plot by Sex.There are also data labels added for the % of total population and count for each group of the pyramid plot.

## The Plot



## The Code

library(ggplot2)  
library(dplyr)  
library(scales)  
  
# Summarise and compute plot values  
pop\_pyramid <- sgResData24 %>%  
 group\_by(Age\_group, Sex) %>%  
 summarise(Pop = sum(Pop), .groups = "drop") %>%  
 group\_by(Sex) %>%  
 mutate(Percent = Pop / sum(Pop) \* 100) %>%  
 ungroup()  
  
# Order Age\_group for proper vertical alignment  
age\_levels <- c("0-9", "10-19", "20-29", "30-39", "40-49",  
 "50-59", "60-69", "70-79", "80-89", "90+")  
pop\_pyramid$Age\_group <- factor(pop\_pyramid$Age\_group, levels = age\_levels)  
  
# Flip male values to negative  
pop\_pyramid <- pop\_pyramid %>%  
 mutate(Pop\_plot = ifelse(Sex == "Males", -Pop, Pop),  
 Label = paste0(comma(abs(Pop)), "\n(", round(abs(Percent), 1), "%)"))  
  
# Plot  
ggplot(pop\_pyramid, aes(x = Age\_group, y = Pop\_plot, fill = Sex)) +  
 geom\_bar(stat = "identity", width = 0.8, position = "identity") +  
 coord\_flip() +  
 geom\_text(aes(label = Label),  
 position = position\_nudge(y = ifelse(pop\_pyramid$Sex == "Males", -50000, 10000)),  
 hjust = ifelse(pop\_pyramid$Sex == "Male", 1, 0),  
 size = 4, color = 'black') +  
 scale\_y\_continuous(labels = function(x) comma(abs(x)), expand = expansion(mult = c(0.1, 0.1))) +  
 scale\_fill\_manual(values = c("Males" = "#619CFF", "Females" = "#F8766D")) +  
 labs(title = "Population vs Age Group & Sex",  
 subtitle = " Males Females",  
 x = "Age Group", y = "Population Count") +  
 theme\_minimal()+  
 theme(  
 plot.margin = margin(10, 20, 20, 20), # top, right, bottom, left  
 legend.position = "none",  
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12),  
 axis.title.x = element\_text(size = 14),  
 axis.title.y = element\_text(size = 14),  
 plot.title = element\_text(size = 18),  
 plot.subtitle = element\_text(size = 14)  
 )

### Observation from the Plot

From the population pyramid chart, we can observe that the Age group of 30-39 has the largest population (14.9% + 15.4% of the overall population), and the largest population for both males and females respectively as well.

We can also observe an aging population based on the shape of the population pyramid chart, with the 0-9, 10-19 and 20-29 Age group populations being lesser than the 30-39 Age group.

## Bar Chart Visualisation of Region vs Population and Sex

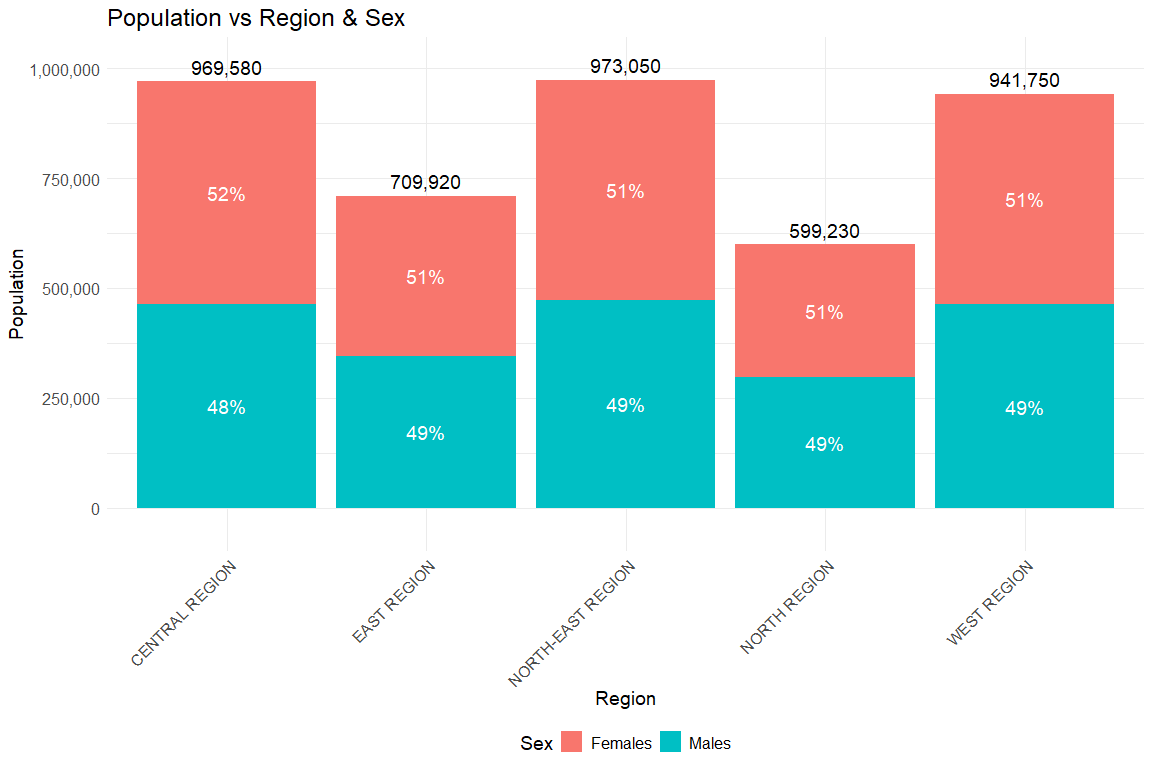
Plotting Region vs Population and Sex.

The **ggplot** and **dplyr** library will be used for this visualisation.

We first calculate the total population by summing the Pop and calculate other the values that will be used for labelling values (the % and total group count) for each group in the bar chart using mutate(), summarise() and group\_by() functions.

The bar chart is then plotted using ggplot, with the respective packages to adjust the aesthetic aspects of the plot such as the addition of the total count and percentages for Males and Females within each group.

## The Plot



## The Code

#Compute total population by Region and Sex  
stacked\_data\_region <- sgResData24 %>%  
 group\_by(Region, Sex) %>%  
 summarise(Pop = sum(Pop, na.rm = TRUE), .groups = "drop") %>%  
 group\_by(Region) %>%  
 mutate(Percent = Pop / sum(Pop) \* 100)  
  
#Compute total population for each Region group (for total label)  
labels\_data\_region <- stacked\_data\_region %>%  
 group\_by(Region) %>%  
 summarise(Pop = sum(Pop))  
  
#Plot stacked bars with percent labels inside and total on top  
 ggplot(stacked\_data\_region, aes(x = Region, y = Pop, fill = Sex)) +  
 geom\_bar(stat = "identity") +  
 geom\_text(aes(label = paste0(round(Percent), "%")),   
 position = position\_stack(vjust = 0.5), size = 5, color = "white") +  
 geom\_text(data = labels\_data\_region, aes(x = Region, y = Pop, label = comma(Pop)),   
 vjust = -0.5, size = 5, inherit.aes = FALSE) +  
 labs(x = "Region", y = "Population", title = "Population vs Region & Sex") +  
 scale\_y\_continuous(labels = function(x) comma(abs(x)), expand = expansion(mult = c(0.1, 0.1))) +  
 theme\_minimal() +  
 theme(  
 axis.text.x = element\_text(size = 12, angle = 45, hjust = 1),# Rotate x-axis labels and adjust alignment  
 axis.text.y = element\_text(size = 12),  
 axis.title.x = element\_text(size = 14),  
 axis.title.y = element\_text(size = 14),  
 plot.title = element\_text(size = 18),  
 legend.position = 'bottom', legend.text = element\_text(size=12),  
 legend.title = element\_text(size = 14)  
 )

### Observation from the Plot

From the Bar chart, we can observe that the North-East Region has the highest population, followed by the Central and then West region. Based on the percentages observed for all 5 regions, we can also tell that there are more Females than Male in the population, with Males having 48 to 49% and Females having 51 to 52%. It can also be observed that the West Region has a greater proportion of Females(52%) compared to Males(48%).

## Half-eye + Box Plot Visualisation of Age vs Region

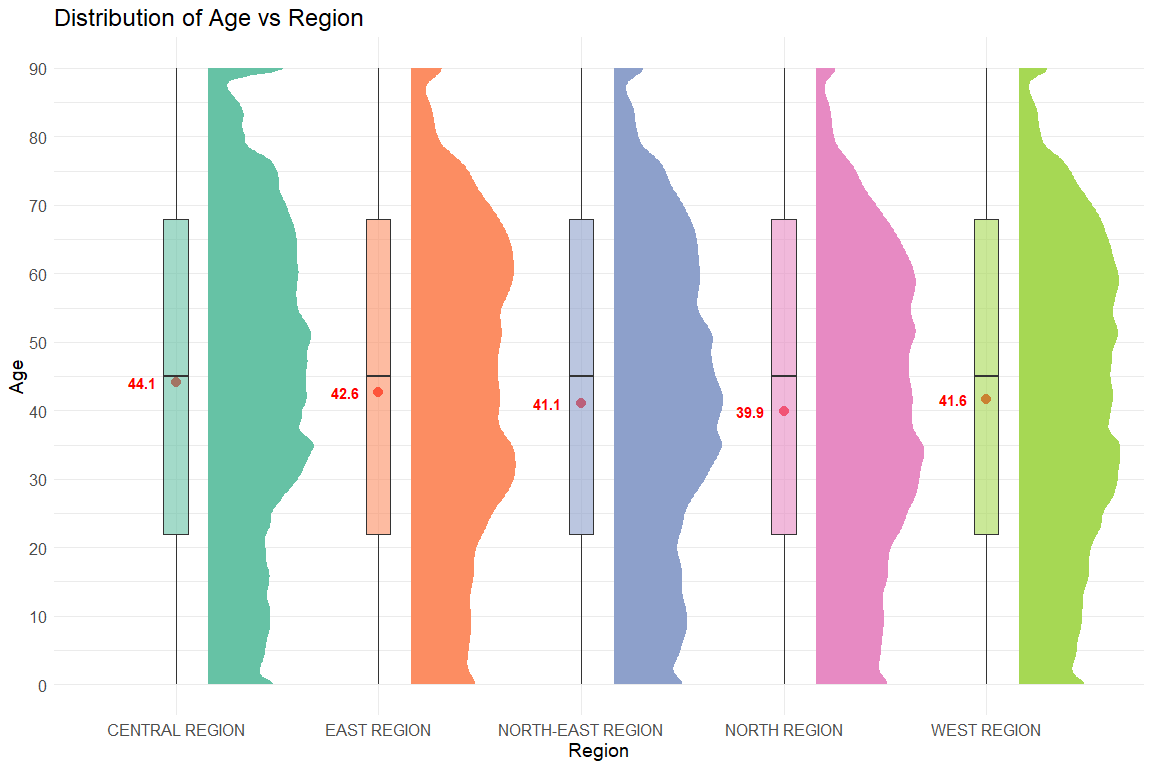
Plotting the Distribution of Age vs Region

The **ggplot**, **dplyr** and **ggdist** library will be used for this visualisation.

We first calculate the mean for each group to label the weighted means in the visualisation (using Age\_num and Pop as the weight).

The chart is then plotted using ggplot to plot a half-eye plot (together with ggdist) and boxplot together. The other respective packages are used to adjust the aesthetics of the plo such as ith the weighted mean calculated earlier indicated with a red point in each boxplot using geompoint(). Each Region group is also assigned a different colour to help differentiate each group visually.

## The Plot



## The Code

library(ggplot2)  
library(dplyr)  
library(ggdist)  
  
# Compute weighted mean for each region  
mean\_labels <- sgResData24 %>%  
 group\_by(Region) %>%  
 summarise(mean\_age = weighted.mean(Age\_num, Pop))  
  
# Plot with mean labels  
ggplot(sgResData24, aes(x = Region, y = Age\_num, fill = Region)) +  
 stat\_halfeye(  
 aes(weight = Pop, fill = Region),  
 adjust = 0.5,  
 width = 0.6,  
 justification = -0.3,  
 .width = 0,  
 point\_colour = NA  
 ) +  
 geom\_point (data = mean\_labels, aes(x = Region, y = mean\_age), shape = 20, size = 5, color = "red") +  
 geom\_boxplot(  
 width = .12,  
 outlier.shape = NA,  
 alpha = 0.6  
 ) +  
 geom\_text(  
 data = mean\_labels,  
 aes(x = Region, y = mean\_age, label = round(mean\_age, 1)),  
 #vjust = -0.8,  
 hjust = 1.7,  
 color = "red",  
 fontface = "bold",  
 size = 4  
 ) +  
 scale\_y\_continuous(breaks = seq(0, max(sgResData24$Age\_num), by = 10))+   
 scale\_fill\_brewer(palette = "Set2") +  
 theme\_minimal() +  
 theme(  
 legend.position = "none",  
 axis.text.x = element\_text(size = 12),# Rotate x-axis labels and adjust alignment  
 axis.text.y = element\_text(size = 12),  
 axis.title.x = element\_text(size = 14),  
 axis.title.y = element\_text(size = 14),  
 plot.title = element\_text(size = 18),  
 ) +  
 labs(x = "Region", y = "Age", title = "Distribution of Age vs Region")

|  |
| --- |
| Use The Correct Data Variable with Correct Data Type |
| Use Age\_num (dbl type) instead of Age (chr type) for this Distribution plot |

### Observation from the Plot

We can view that the distribution of the age of the population across the region is the same, with roughly the same median as well based on the boxplot median position. However, it can be observed that the mean age are different for each region, with Central having the highest(44.1), followed by East(42.6) and West(41.6). The higher mean in Central Region may indicate a greater proportion of older people living in these areas compared to other Regions.

This can also be observed from the distribution seen with the half-eye plot, where the end of the distribution graph shows a higher distribution of population above the age of around 80 compared to the other regions as shown in the screenshot below:

