Take-home Exercise 1b: Data Visualisation Makeover

Belinda Lim

2025-05-06

# **1 Introduction**

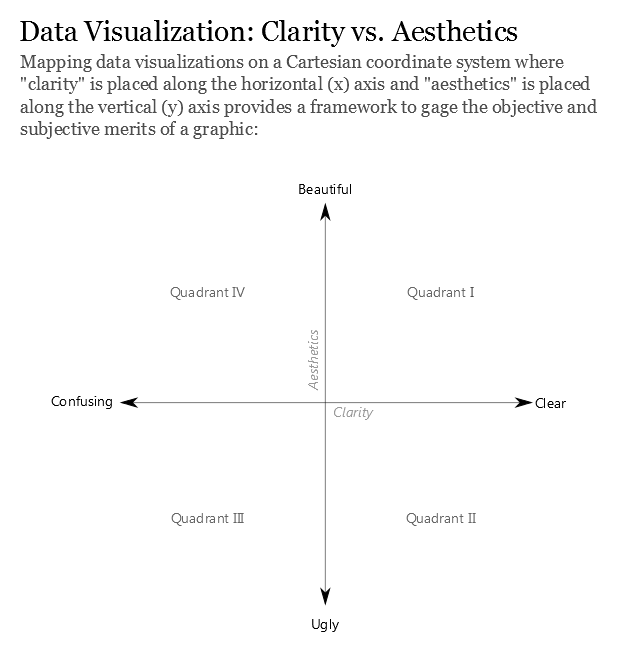
## **1.1 Overview**

Under Take-home Exercise 1b, we are required to select one submission from our classmate for Take-home Exercise 1 to critic 3 good design principles and 3 areas for further improvements. With reference to the comments, we will provide a makeover of the data visualisation.

## **1.2 Framework for Gauging Data Visualisation**

Under the article published by [Ben Jones](https://dataremixed.com/2012/05/data-visualization-clarity-or-aesthetics/), he shared the following framework to gauge the objective and subjective merits of a data visualisation. It will be used to review my classmate’s submission (see link to [original page](https://iss608-vaa.netlify.app/take-home_ex/take-home_ex01/take%20home%20exercise%201)).The submission was selected as the charts can be classified into the Quadrant II, III and IV:

* Quadrant I = [clear, beautiful]
* Quadrant II = [clear, ugly]
* Quadrant III = [confusing, ugly]
* Quadrant IV = [confusing, beautiful]



# **2 Critique**

## **2.1 Loading of libraries and data pre-processing based on classmate’s codes**

library("pacman")  
suppressWarnings(library("lubridate"))  
library(readr)

pacman::p\_load(tidyverse, readxl,   
 janitor, lubridate,   
 ggplot2, ggthemes,   
 scales, ggridges,   
 ggdist, patchwork)

Resident\_Data <- read.csv("data/Resident\_Data.csv")

glimpse(Resident\_Data)

Rows: 60,424  
Columns: 6  
$ PA <chr> "Ang Mo Kio", "Ang Mo Kio", "Ang Mo Kio", "Ang Mo Kio", "Ang Mo K…  
$ SZ <chr> "Ang Mo Kio Town Centre", "Ang Mo Kio Town Centre", "Ang Mo Kio T…  
$ Age <chr> "0", "0", "1", "1", "2", "2", "3", "3", "4", "4", "5", "5", "6", …  
$ Sex <chr> "Males", "Females", "Males", "Females", "Males", "Females", "Male…  
$ Pop <int> 10, 10, 10, 10, 10, 10, 10, 10, 30, 10, 20, 10, 20, 30, 30, 10, 3…  
$ Time <int> 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024,…

Resident\_Data[duplicated(Resident\_Data),]

[1] PA SZ Age Sex Pop Time  
<0 rows> (or 0-length row.names)

Note that my classmate has renamed SZ to Subzone and created two additional columns AG (Age Group) and FA (Functional Age) as shown in my classmate’s website (see extract in the Figure below). To ensure that the charts are reproducible, the data pre-processing steps and codes should be spelt out.

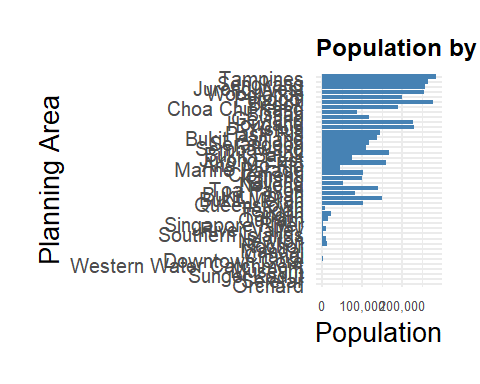
Figure 1: Extracts from classmate’s website

## **2.2 Bar chart based on Population Area**

### **2.2.1 Original Chart**

The original chart is reproduced below using the codes provided by my classmate.

filtered\_data <- Resident\_Data[!is.na(Resident\_Data$Pop) & Resident\_Data$Pop > 0, ]  
  
ggplot(data = filtered\_data, aes(x = reorder(`PA`, Pop), y = Pop)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 coord\_flip() +  
 labs(  
 title = "Population by Planning Area (Singapore, 2024)",  
 x = "Planning Area",  
 y = "Population"  
 ) +  
 scale\_y\_continuous(labels = comma) +  
 theme\_minimal(base\_size = 20) + # Increase base font size  
   
 theme(  
 plot.title = element\_text(size = 18, face = "bold"),  
 axis.text.y = element\_text(size = 15),  
 axis.text.x = element\_text(size = 10),  
 plot.margin = margin(1, 1, 1, 1, "cm") # Add breathing space  
 )



However, the chart produced did not look like the one published on my classmate’s website:

Figure 2: Extracts from classmate’s website

Good Design Principles

* The bar chart summarised the population numbers in each all PA in Singapore.
* It also has a labelled x-axis scale.
* It uses gridlines that are greyed-out which helps with the visual mapping to the x-axis.

Areas of Improvements

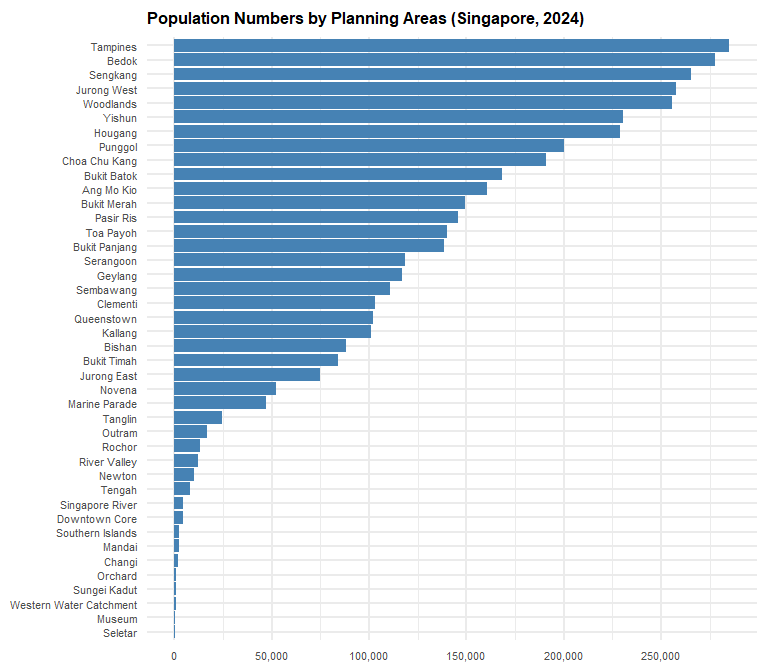
However, the chart can be classified under Quadrant II = [clear, ugly].

To improve the chart, it is recommended to:

* Sort bar chart by population numbers
* Remove x-axis and y-axis label since the information is found in the chart title.
* Adjust size of the x-axis numbers and add more intervals for the ticks
* Adjust size of the title
* Remove plot margins to allow chart to be bigger

### **2.2.2 Makeover of Original Chart**

filtered\_data <- Resident\_Data %>%  
 filter(!is.na(Pop) & Pop > 0) %>%  
 group\_by(PA) %>%  
 summarise(Pop = sum(Pop), .groups = "drop")  
  
ggplot(data = filtered\_data, aes(x = reorder(PA, Pop), y = Pop)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 coord\_flip() +  
 labs(  
 title = "Population Numbers by Planning Areas (Singapore, 2024)",  
 x = NULL,   
 y = NULL   
 ) +  
 scale\_y\_continuous(breaks = c(0, 50000, 100000, 150000, 200000, 250000, 300000), labels = comma) +  
 theme\_minimal(base\_size = 16) +  
 theme(  
 plot.title = element\_text(size = 12, face = "bold"),  
 axis.text.y = element\_text(size = 8),   
 axis.text.x = element\_text(size = 8)  
 )



## **2.3 Pyramid Bar chart based on Population Area**

### **2.3.1 Original Chart**

The attempt to reproduce the original chart using the code below has failed:

pyramid\_data \<- aggregate(Pop \~ `AG` + Sex, data = Resident\_Data, sum)  
  
age\_order \<- c("0\_to\_4", "5\_to\_9", "10\_to\_14", "15\_to\_19", "20\_to\_24", "25\_to\_29", "30\_to\_34", "35\_to\_39", "40\_to\_44", "45\_to\_49", "50\_to\_54", "55\_to\_59", "60\_to\_64", "65\_to\_69", "70\_to\_74", "75\_to\_79", "80\_to\_84", "85\_to\_89", "90\_and\_over")  
  
pyramid\_data$AG <- factor(pyramid\_data$AG, levels = age\_order)  
  
pyramid\_data$Pop[pyramid\_data$Sex == "Males"\] \<- -pyramid\_data$Pop[pyramid\_data$Sex == "Males"\]  
  
ggplot(data = pyramid\_data, aes(x = AG, y = Pop, fill = Sex)) + geom\_bar(stat = "identity") + coord\_flip() + labs(title = "Population Pyramid (2024)", x = "Age Group", y = "Population") + scale\_y\_continuous(labels = comma) + scale\_fill\_manual(values = c("skyblue", "salmon")) + theme\_minimal() + theme(axis.text.y = element\_text(size = 10))

A check was done on the column names.

colnames(Resident\_Data)

[1] "PA" "SZ" "Age" "Sex" "Pop" "Time"

There was an error as the original data does not have AG (or Age Group) but AG was cited in the codes above.

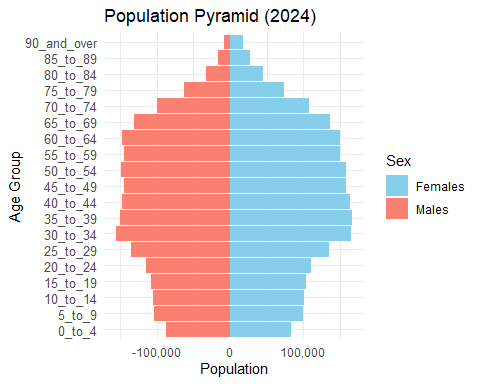
To rectify, the following codes for AG column were inserted

Resident\_Data <- Resident\_Data %>%  
 mutate(  
 Age = recode(Age, "90\_and\_Over" = "90"),   
 Age = as.numeric(Age),   
 Pop = as.numeric(Pop)   
 )

Resident\_Data <- Resident\_Data %>%  
 mutate(  
 AG = case\_when(  
 Age >= 0 & Age <= 4 ~ "0\_to\_4",  
 Age >= 5 & Age <= 9 ~ "5\_to\_9",  
 Age >= 10 & Age <= 14 ~ "10\_to\_14",  
 Age >= 15 & Age <= 19 ~ "15\_to\_19",  
 Age >= 20 & Age <= 24 ~ "20\_to\_24",  
 Age >= 25 & Age <= 29 ~ "25\_to\_29",  
 Age >= 30 & Age <= 34 ~ "30\_to\_34",  
 Age >= 35 & Age <= 39 ~ "35\_to\_39",  
 Age >= 40 & Age <= 44 ~ "40\_to\_44",  
 Age >= 45 & Age <= 49 ~ "45\_to\_49",  
 Age >= 50 & Age <= 54 ~ "50\_to\_54",  
 Age >= 55 & Age <= 59 ~ "55\_to\_59",  
 Age >= 60 & Age <= 64 ~ "60\_to\_64",  
 Age >= 65 & Age <= 69 ~ "65\_to\_69",  
 Age >= 70 & Age <= 74 ~ "70\_to\_74",  
 Age >= 75 & Age <= 79 ~ "75\_to\_79",  
 Age >= 80 & Age <= 84 ~ "80\_to\_84",  
 Age >= 85 & Age <= 89 ~ "85\_to\_89",  
 Age >= 90 ~ "90\_and\_over",  
 TRUE ~ NA\_character\_  
 )  
 )

The orignal code can now be reproduced:

pyramid\_data <- aggregate(Pop ~ `AG` + Sex, data = Resident\_Data, sum)  
  
age\_order <- c("0\_to\_4", "5\_to\_9", "10\_to\_14", "15\_to\_19", "20\_to\_24", "25\_to\_29",   
 "30\_to\_34", "35\_to\_39", "40\_to\_44", "45\_to\_49", "50\_to\_54", "55\_to\_59",   
 "60\_to\_64", "65\_to\_69", "70\_to\_74", "75\_to\_79", "80\_to\_84", "85\_to\_89",   
 "90\_and\_over")  
  
pyramid\_data$AG <- factor(pyramid\_data$AG, levels = age\_order)  
  
pyramid\_data$Pop[pyramid\_data$Sex == "Males"] <- -pyramid\_data$Pop[pyramid\_data$Sex == "Males"]  
  
ggplot(data = pyramid\_data, aes(x = AG, y = Pop, fill = Sex)) +  
 geom\_bar(stat = "identity") +  
 coord\_flip() +  
 labs(title = "Population Pyramid (2024)", x = "Age Group", y = "Population") +  
 scale\_y\_continuous(labels = comma) +  
 scale\_fill\_manual(values = c("skyblue", "salmon")) +   
 theme\_minimal() +  
 theme(axis.text.y = element\_text(size = 10))



Good Design Principles

* The chart is an age pyramid which illustrated the age structure of Singapore’s population well as we can immediately see that the main bulk of the population is between 30 to 69.
* Instead of doing a simple bar chart where Males and Females population are both on the right, the scale of the bar chart has been adjusted to move Males to the left.
* The gridlines also help with the visual mapping to the x-axis numbers.

However, the age pyramid can be classified under Quadrant IV = [confusing, beautiful]. This is because males are coded as red and females are coded as blue. There is also a negative number in the x-axis for population.

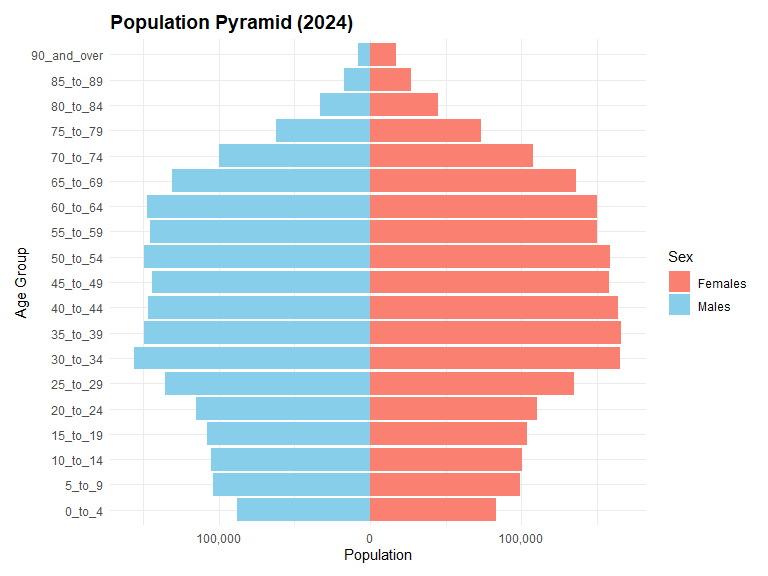
Areas of Improvements

To improve the chart, it is recommended to:

* Correct the codes for binning as mentioned above.
* Swop the colour code for gender
* Update the scale to be positive number
* Compress the chart to be shorter

### **2.3.2 Makeover of Original Chart**

ggplot(data = pyramid\_data, aes(x = AG, y = Pop, fill = Sex)) +  
 geom\_bar(stat = "identity", width = 0.9) +  
 coord\_flip() +  
 labs(title = "Population Pyramid (2024)", x = "Age Group", y = "Population") +  
 scale\_y\_continuous(labels = function(x) comma(abs(x))) +  
 scale\_fill\_manual(values = c("Males" = "skyblue", "Females" = "salmon")) +  
 theme\_minimal() +  
 theme(  
 axis.text.y = element\_text(size = 9),  
 axis.text.x = element\_text(size = 9),  
 plot.title = element\_text(size = 14, face = "bold"),  
 plot.margin = margin(10, 10, 10, 10)  
 )



## **2.4 Box Plot based on Age Range**

### **2.4.1 Original Chart**

The attempt to reproduce the original chart using the code below has failed too as there was no FA column created.

FA\_order <- c("<= 60", ">60 to 80", ">80 to 100", ">100 to 120", ">120", "Not Available")  
Resident\_Data$FA <- factor(Resident\_Data$FA, levels = FA\_order)  
  
ggplot(Resident\_Data, aes(x = FA, y = Pop)) +  
 geom\_boxplot(fill = "lightgreen") +  
 labs(title = "Population by Functional Age Range (FA)", x = "FA Band", y = "Population") +  
 theme\_minimal()

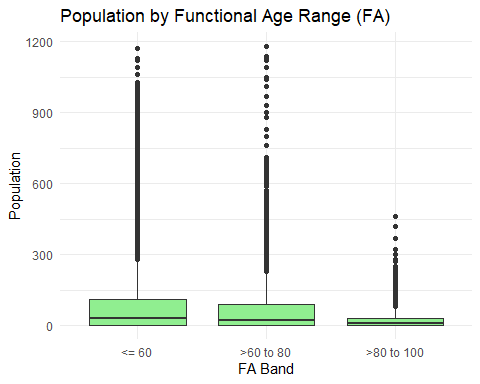
To rectify, it is assumed that FA refers to Functional Age binned based on Age <= 60, >60 to 80, >80 to 100, >100 to 120, >120, Not Available.

The following codes for FA column were inserted

Resident\_Data <- Resident\_Data %>%  
 mutate(  
 FA = case\_when(  
 Age <= 60 ~ "<= 60",  
 Age > 60 & Age <= 80 ~ ">60 to 80",  
 Age > 80 & Age <= 100 ~ ">80 to 100",  
 Age > 100 & Age <= 120 ~ ">100 to 120",  
 Age > 120 ~ ">120",  
 TRUE ~ "Not Available"  
 )  
 )

I have attempted to reproduce the original chart using the codes provided but was unable to do so as it is not possible to distinguish population aged “>100 to 120”, “>120” and “Not Available” since all population aged 90 and above will be classified under “>80 to 100”. Hence, I am unable to critique on the good design principles for this chart.

FA\_order <- c("<= 60", ">60 to 80", ">80 to 100", ">100 to 120", ">120", "Not Available")  
Resident\_Data$FA <- factor(Resident\_Data$FA, levels = FA\_order)  
  
ggplot(Resident\_Data, aes(x = FA, y = Pop)) +  
 geom\_boxplot(fill = "lightgreen") +  
 labs(title = "Population by Functional Age Range (FA)", x = "FA Band", y = "Population") +  
 theme\_minimal()



Areas of Improvement

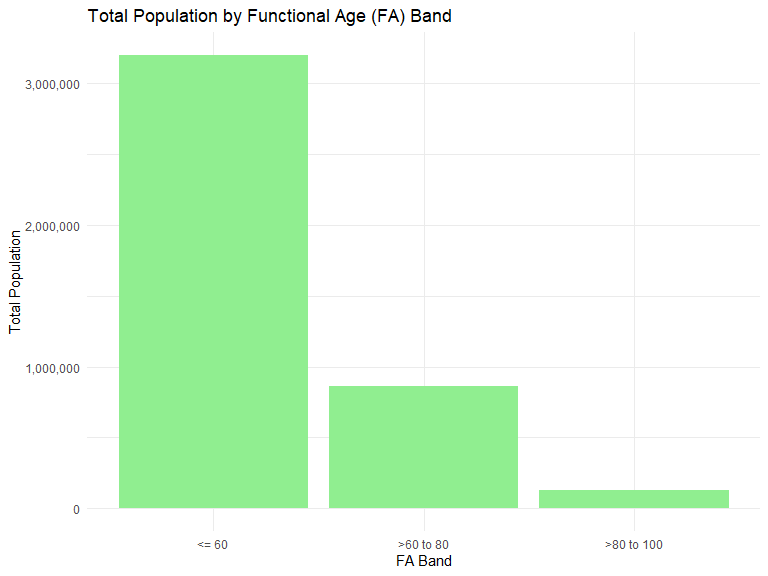
* The chart is a box plot based on age range. However, box plots are used to show distributions of numeric continuous data values along y-axis, and compare them between multiple groups.
* As mentioned earlier, the data grouping seems to be erroneous and needs corrections to the code.
* Given that the population numbers are by subzones, the box plot shows large number of outliers which is not useful for storytelling.

Hence, the chart can be classified under Quadrant III = [confusing, ugly].

### **2.4.2 Makeover of Original Chart**

If the intent is to show the total population number in the 3 age groups as per the writeup, it is recommended to use a bar graph instead as shown below. The total population numbers in each age group are added together first, followed by the plotting.

Resident\_FA\_summary <- Resident\_Data %>%  
 group\_by(FA) %>%  
 summarise(TotalPop = sum(Pop, na.rm = TRUE))  
  
ggplot(Resident\_FA\_summary, aes(x = FA, y = TotalPop)) +  
 geom\_col(fill = "lightgreen") +  
 labs(title = "Total Population by Functional Age (FA) Band", x = "FA Band", y = "Total Population") +  
 scale\_y\_continuous(labels = comma) +  
 theme\_minimal()



# **3 References**

**Jones, B.** (2012, May 28). *Data visualization: Clarity or aesthetics?* <https://dataremixed.com/2012/05/data-visualization-clarity-or-aesthetics/>