Take-home Exercise 1–Phase 2

Evaluation and improvement of classmates’ visual works

Li Yuquan

2025-04-30

# 1 Overview

For the second phase of Take-home Exercise 1, we are required to critically review a data visualization submitted by one of our classmates. This involves identifying and explaining three effective design principles demonstrated in their work, as well as highlighting three areas for potential improvement. Based on this critique, we are expected to develop a revised or “makeover” version of their visualization that addresses the identified weaknesses while preserving the original message and intent.

In this part, I will conduct a critical analysis of the visualization submitted by my classmate [**Tai Qiuyan**](https://isss608-ay2025-qiuyan.netlify.app/takehome_ex/takehome_ex01/takehome_ex01), discussing three strengths in design and proposing improvements to enhance clarity, effectiveness, and visual communication.

# 2 Load packages and import data

## Load packages

| name | purpose |
| --- | --- |
| scales | Nicely formats axis ticks and legends. |
| ggrepel | Adds non‑overlapping text labels to ggplots. |
| patchwork | Combines multiple ggplots into a single figure. |
| ggthemes | Supplies predefined aesthetic themes. |
| hrbrthemes | Extends ggthemes with modern, typographically friendly themes and font support. |
| tidyverse | Meta‑package that loads ggplot2, dplyr, tidyr, readr, stringr, forcats, etc. — the backbone for data import, transformation, and visualisation. |
| tidytext | Tokenises and analyses text. |
| treemapify | Creates ggplot‑compatible treemaps. |
| dplyr | Provides a fast, consistent, and intuitive set of functions for data manipulation, including filtering, selecting, grouping, summarising, and joining data frames. |
| RColorBrewer | Provides color palettes from ColorBrewer for categorical and sequential data visualisation. |
| grid | Offers low-level functions for building and arranging graphical objects in R. |
| gghalves | Adds half‑geoms such as half‑violin and half‑boxplots to ggplot2 for compact comparisons. |

pacman::p\_load(scales, ggrepel, patchwork,   
 ggthemes, hrbrthemes,tidyverse,   
 tidytext,treemapify, dplyr, RColorBrewer, grid, gghalves)

## Import data

The code imports the dataset respopagesex2024.csv and load the CSV file into a dataframe named population\_data.

After import, the mutate() function is applied to convert the Age column from character to numeric format.

population\_data <- read\_csv("data/respopagesex2024.csv") %>%  
 mutate(Age = as.numeric(Age))

## Data overview

 glimpse() function provides a concise overview of the dataset’s structure, displaying column names, data types, and sample values.

glimpse(population\_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 <dbl> 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 1…  
$ Sex <chr> "Males", "Females", "Males", "Females", "Males", "Females", "Male…  
$ Pop <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 30, 10, 20, 10, 20, 30, 30, 10, 3…  
$ Time <dbl> 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024,…

# 3 Original visualization

The original visualization is from my classmate Tai Qiuyan. Please click on this [**link**](https://isss608-ay2025-qiuyan.netlify.app/takehome_ex/takehome_ex01/takehome_ex01) to view her original visualization report or obtain more information. In this section, I will make comments and revisions on her visualization 3.

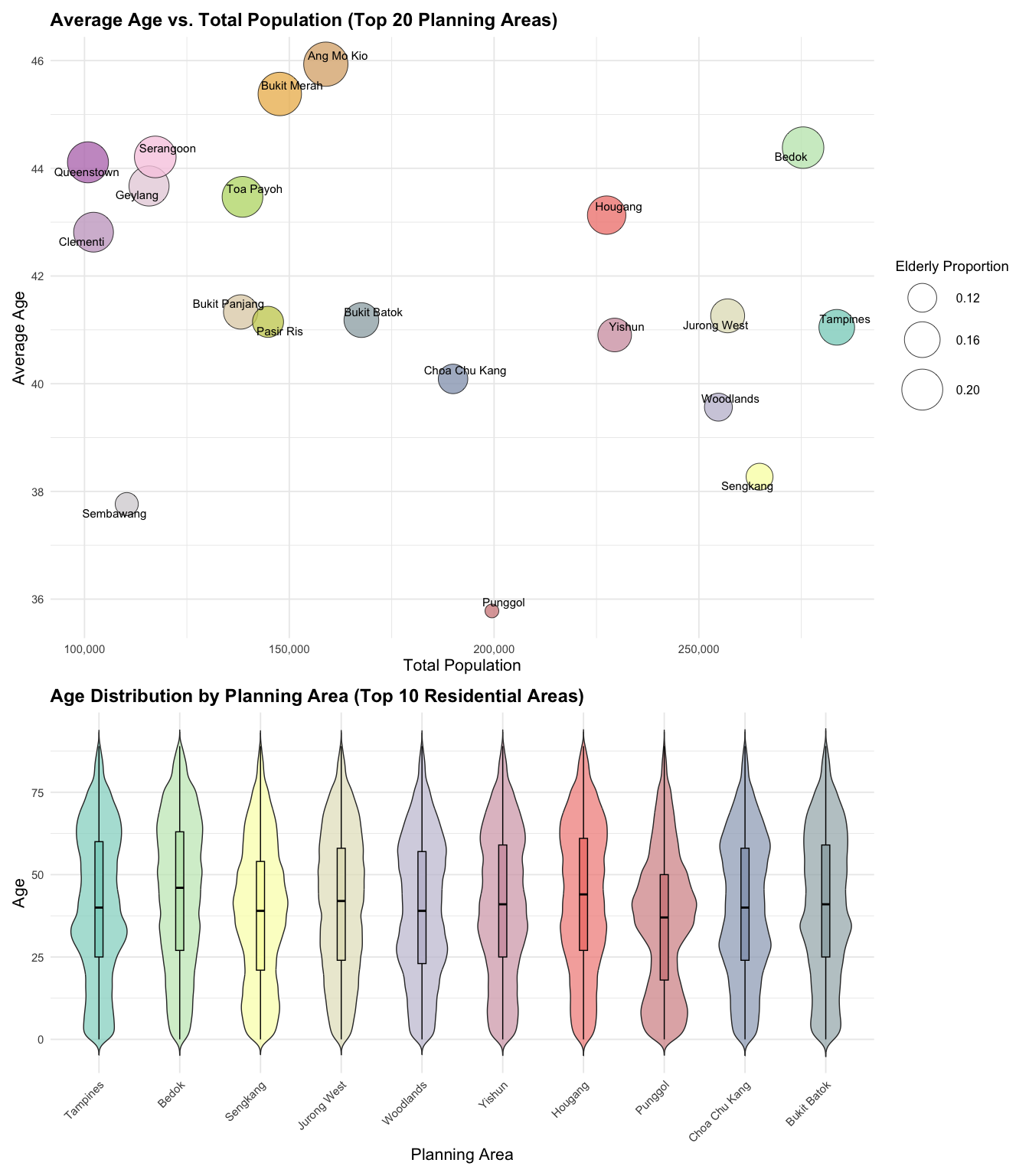
## 3.1 Data cleaning and preparation

|  |
| --- |
| Tip |
| All data import, data processing, and variable naming in the original visualisation strictly follow the code submitted by the classmate. |

violin\_data <- population\_data %>%  
 mutate(Age = as.numeric(Age)) %>%  
 filter(!is.na(Age), !is.na(PA)) %>%  
 group\_by(PA, Age) %>%  
 summarise(Pop = sum(Pop), .groups = "drop") %>%  
 uncount(weights = Pop)  
  
top10\_PAs <- population\_data %>%  
 group\_by(PA) %>%  
 summarise(Total\_Pop = sum(Pop), .groups = "drop") %>%  
 arrange(desc(Total\_Pop)) %>%  
 slice\_head(n = 10) %>%  
 pull(PA)  
  
violin\_filtered <- violin\_data %>%  
 filter(PA %in% top10\_PAs) %>%  
 mutate(PA = factor(PA, levels = top10\_PAs))  
  
  
top20\_PAs <- population\_data %>%  
 group\_by(PA) %>%  
 summarise(Total\_Pop = sum(Pop), .groups = "drop") %>%  
 arrange(desc(Total\_Pop)) %>%  
 slice\_head(n = 20) %>%  
 pull(PA)  
  
pa\_summary20 <- violin\_data %>%  
 filter(PA %in% top20\_PAs) %>%  
 mutate(AgeGroup = if\_else(Age > 65, "Elderly", "Other")) %>%  
 group\_by(PA) %>%  
 summarise(  
 Avg\_Age = mean(Age),  
 Total\_Pop = n(),  
 Elderly\_Prop = mean(AgeGroup == "Elderly"),  
 .groups = "drop"  
 )

## 3.2 Visualization

### The Plot



### The Code

# palette for all 20 PAs  
area\_colors\_all <- colorRampPalette(  
 RColorBrewer::brewer.pal(10, "Set3")  
)(20)  
names(area\_colors\_all) <- top20\_PAs  
  
# Violin plot  
p1 <- ggplot(violin\_filtered, aes(x = PA, y = Age, fill = PA)) +  
 geom\_violin(trim = FALSE, scale = "area", width = 0.8, alpha = 0.7) +  
 geom\_boxplot(width = 0.1, outlier.shape = NA, color = "black") +  
 scale\_fill\_manual(values = area\_colors\_all[top10\_PAs]) +  
 labs(  
 title = "Age Distribution by Planning Area (Top 10 Residential Areas)",  
 x = "Planning Area",  
 y = "Age"  
 ) +  
 theme\_minimal(base\_size = 14) +  
 theme(  
 legend.position = "none",  
 axis.text.x = element\_text(angle = 45, hjust = 1),  
 axis.title = element\_text(size = 16),  
 plot.title = element\_text(face = "bold", size = 18)  
 )  
  
# Scatter plot  
p2 <- ggplot(pa\_summary20, aes(  
 x = Total\_Pop,  
 y = Avg\_Age,  
 size = Elderly\_Prop,  
 fill = PA  
 )) +  
 geom\_point(shape = 21, alpha = 0.8, color = "black") +  
 geom\_text\_repel(aes(label = PA), size = 4, max.overlaps = 30) +  
   
 scale\_fill\_manual(values = area\_colors\_all, guide = FALSE) +  
 scale\_size\_continuous(  
 name = "Elderly Proportion",  
 range = c(6, 20)  
 ) +  
 scale\_x\_continuous(labels = scales::comma) +  
 labs(  
 title = "Average Age vs. Total Population (Top 20 Planning Areas)",  
 x = "Total Population",  
 y = "Average Age"  
 ) +  
 theme\_minimal(base\_size = 14) +  
 theme(  
 legend.position = "right",  
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.title = element\_text(size = 16),  
 plot.title = element\_text(face = "bold", size = 18)  
 )  
  
# combine  
(p2 / p1) + plot\_layout(heights = c(1, 0.6))

# 4 Evaluation and Improvement

## 4.1 Good design principles

1. This visualization effectively integrates total population, average age, and elderly proportion into a single bubble chart, enabling users to compare the scale and aging level of multiple planning areas at a glance through a multidimensional view.
2. The use of consistent and vibrant color schemes across both charts ensures that the same planning areas are easily recognizable, enhancing readability and supporting a visually coherent interpretation across different visual elements.
3. The two charts complement each other: the upper bubble chart provides a macro-level overview of population characteristics, while the lower violin plot reveals detailed age distribution patterns, together offering a more comprehensive and in-depth understanding of demographic structures.

## 4.2 Further improvement

1. The y-axis of the violin plot lacks a clear label indicating that it represents age, and the wide scale range (0–75) without finer graduations makes it difficult for viewers to interpret the exact distribution. Adding a descriptive axis title and refining the scale would improve clarity.
2. In the bubble chart, some labels—such as those for Geylang and Serangoon—overlap with the bubbles or appear too densely packed, reducing legibility. This could be improved by repositioning the labels or abbreviating long area names to avoid visual clutter.
3. The violin plot does not include median lines or reference markers such as the age of 65, making it harder to quickly identify central tendencies or distinguish age structure differences across areas. Including such markers and enlarging the box plots would enhance interpretability.
4. The legend for the bubble chart is positioned too close to the bottom chart, disrupting the overall layout balance and drawing attention away from the data. Relocating the legend to the upper-right empty space and resizing it appropriately would create a more harmonious and visually balanced design.

## 4.3 Makeover version

### 4.3.1 Data Prepare

### The Description

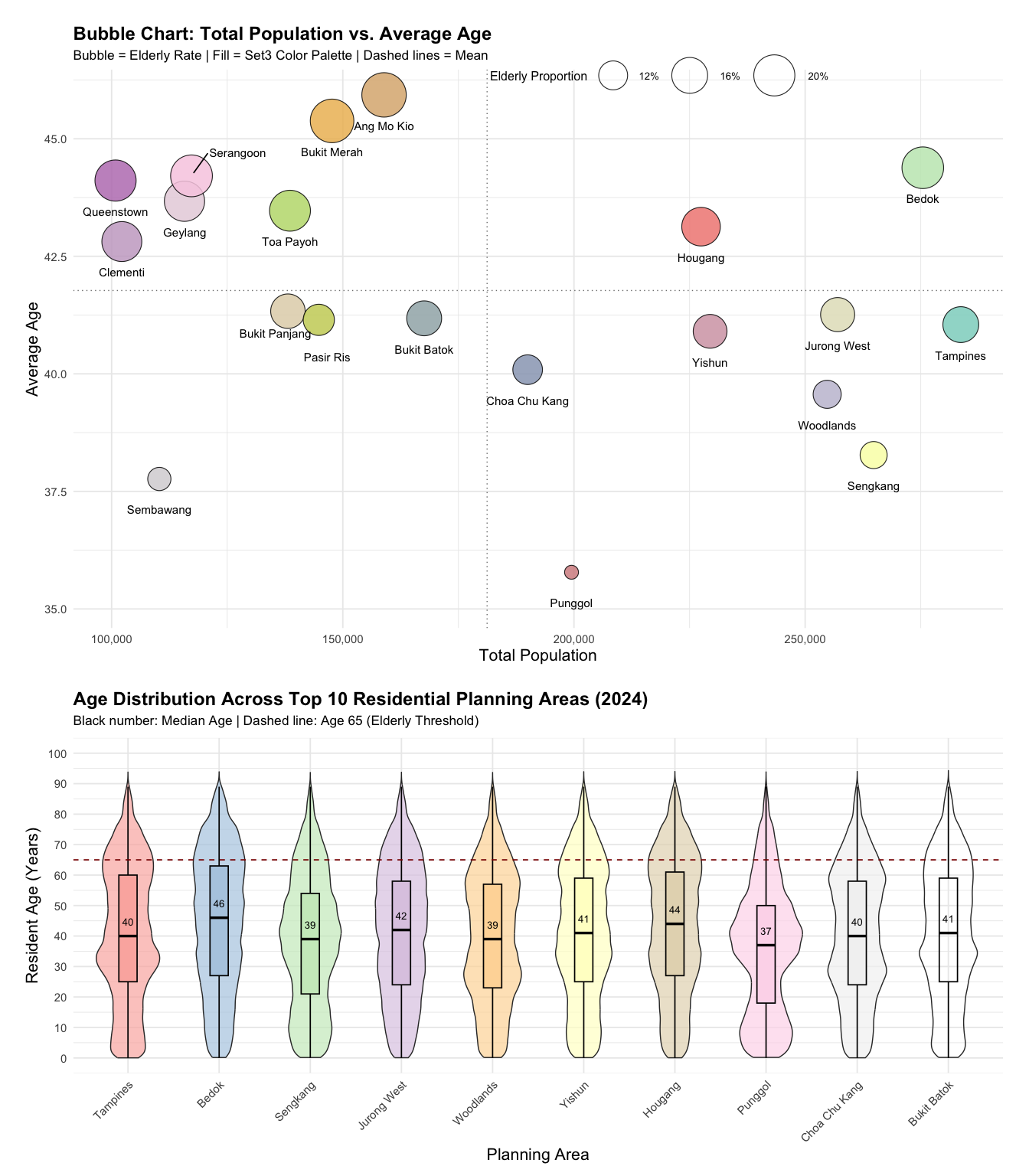
1. The population\_data was preprocessed specifically for violin plot visualization, aiming to generate age distribution data grouped by gender, in order to show the age structure of different genders across the top 10 most populous planning areas.
2. This line converts the PA variable in the pa\_summary20 data frame into an ordered factor, following the population-based ranking specified in top20\_PAs, to ensure that the x-axis in the subsequent plot displays planning areas sorted by population size.

### The Code

violin\_gender\_split <- population\_data %>%  
 filter(PA %in% top10\_PAs) %>%  
 mutate(Age = as.numeric(Age)) %>%  
 filter(!is.na(Sex), !is.na(Age)) %>%  
 group\_by(PA, Sex, Age) %>%  
 summarise(Pop = sum(Pop), .groups = "drop") %>%  
 uncount(weights = Pop) %>%  
 mutate(  
 Sex = factor(Sex, levels = c("Females", "Males")),  
 PA = factor(PA, levels = top10\_PAs)  
 )  
  
pa\_summary20$PA <- factor(pa\_summary20$PA, levels = top20\_PAs)

### 4.3.2 Markover Version 1

#### The Plot



#### The Code

area\_colors\_all <- colorRampPalette(RColorBrewer::brewer.pal(10, "Set3"))(20)  
names(area\_colors\_all) <- top20\_PAs  
  
p5 <- ggplot(pa\_summary20, aes(x = Total\_Pop,   
 y = Avg\_Age,   
 size = Elderly\_Prop)) +  
 geom\_point(  
 fill = area\_colors\_all[pa\_summary20$PA],  
 shape = 21,  
 stroke = 0.6,  
 alpha = 0.85,  
 color = "black",  
 show.legend = TRUE  
 ) +  
 geom\_text\_repel(  
 data = subset(pa\_summary20,   
 PA == "Serangoon"),  
 aes(label = PA),  
 size = 4,  
 box.padding = 0.5,  
 point.padding = 0.6,  
 segment.color = "black",  
 segment.size = 0.6,  
 min.segment.length = 0,  
 force = 4,  
 nudge\_y = 0.5,  
 nudge\_x = 10000  
 ) +  
 geom\_text\_repel(  
 data = subset(pa\_summary20,   
 PA != "Serangoon"),  
 aes(label = PA),  
 size = 4,  
 box.padding = 0.5,  
 point.padding = 0.5,  
 segment.alpha = 0,  
 force = 3,  
 nudge\_y = -0.65  
 ) +  
 scale\_size\_continuous(  
 name = "Elderly Proportion",  
 range = c(6, 20),  
 labels = percent\_format(accuracy = 1)  
 ) +  
 geom\_vline(xintercept = mean(pa\_summary20$Total\_Pop), linetype = "dotted", color = "grey50") +  
 geom\_hline(yintercept = mean(pa\_summary20$Avg\_Age),   
 linetype = "dotted",   
 color = "grey50") +  
 scale\_x\_continuous(labels = comma) +  
 labs(  
 title = "Bubble Chart: Total Population vs. Average Age",  
 subtitle = "Bubble = Elderly Rate | Fill = Set3 Color Palette | Dashed lines = Mean",  
 x = "Total Population",  
 y = "Average Age"  
 ) +  
 theme\_minimal(base\_size = 14) +  
 theme(  
 legend.position = c(0.82, 1.05),  
 legend.direction = "horizontal",  
 legend.justification = c("right", "top"),  
 legend.key.size = unit(0.8, "lines"),  
 legend.title = element\_text(size = 12),  
 legend.text = element\_text(size = 10),  
 axis.title = element\_text(size = 16),  
 plot.title = element\_text(size = 18, face = "bold"),  
 plot.subtitle = element\_text(size = 13),  
 plot.margin = margin(20, 20, 20, 20)  
 )  
  
  
pd <- ggplot(violin\_filtered, aes(x = PA,   
 y = Age,   
 fill = PA)) +  
 geom\_violin(trim = FALSE,   
 scale = "area",   
 width = 0.8,   
 alpha = 0.7) +  
 geom\_boxplot(width = 0.2,   
 outlier.shape = NA,   
 color = "black",   
 linewidth = 0.6) +  
 stat\_summary(  
 fun = median,  
 geom = "text",  
 aes(label = round(..y.., 0)),  
 color = "black",  
 size = 3.5,  
 vjust = -1.5  
 ) +  
 geom\_hline(yintercept = 65,   
 linetype = "dashed",   
 color = "darkred",   
 linewidth = 0.6) +  
 scale\_fill\_brewer(palette = "Pastel1") +  
 scale\_y\_continuous(  
 limits = c(0, 100),  
 breaks = seq(0, 100, by = 10)  
 ) +  
 labs(  
 title = "Age Distribution Across Top 10 Residential Planning Areas (2024)",  
 subtitle = "Black number: Median Age | Dashed line: Age 65 (Elderly Threshold)",  
 x = "Planning Area",  
 y = "Resident Age (Years)"  
 ) +  
 theme\_minimal(base\_size = 14) +  
 theme(  
 legend.position = "none",  
 axis.text.x = element\_text(angle = 45,   
 hjust = 1),  
 axis.title = element\_text(size = 16),  
 plot.title = element\_text(face = "bold",   
 size = 18),  
 plot.subtitle = element\_text(size = 13,   
 margin = margin(b = 10))  
 )  
  
(p5 / pd) + plot\_layout(heights = c(1, 0.6))

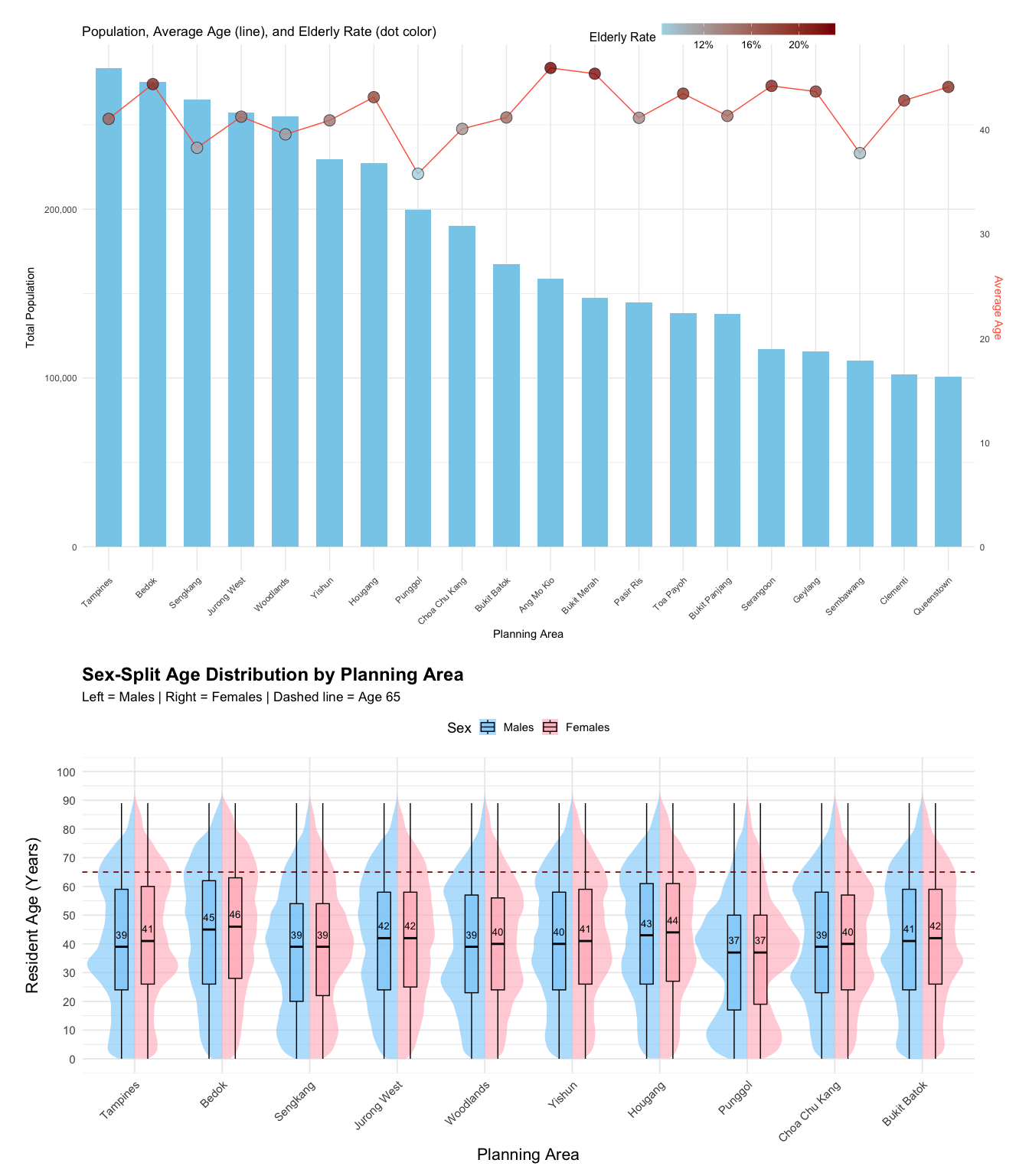
|  |
| --- |
| Changes |
| Bubble Chart   1. Adjusted the position of area labels to be closer to the corresponding bubbles and added guiding lines for specific densely packed areas. 2. Moved the legend from the right side to the top of the chart to create a more balanced and visually coordinated layout.   Violin Plot   1. Added dashed reference lines to represent average values (mean age and total population), enhancing comparative analysis. 2. Included numerical labels for median values within the boxplots. 3. Enlarged the boxplot elements for better visibility and clarity. 4. Revised the y-axis scale to range from 0 to 100, changing the tick intervals from 25 to 10 years for improved granularity. |

### 4.3.3 Markover Version 2

After making improvements to the original visualization in the Markover Version 1 section, I believe there are still some methods that can achieve better visualization results by changing the original chart type.

Therefore, while retaining the modified parts of the Markover Version 1 section, I made the following second version improvements.

#### The Plot



#### The Code

max\_pop <- max(pa\_summary20$Total\_Pop)  
  
pb <- ggplot(pa\_summary20, aes(x = PA)) +  
 geom\_col(aes(y = Total\_Pop), fill = "skyblue",   
 width = 0.6) +  
 geom\_line(aes(y = Avg\_Age \* max\_pop / max(pa\_summary20$Avg\_Age),   
 group = 1),   
 color = "tomato",   
 size = 0.5) +  
 geom\_point(aes(y = Avg\_Age \* max\_pop / max(pa\_summary20$Avg\_Age),   
 fill = Elderly\_Prop),  
 shape = 21, size = 5,   
 stroke = 0.4,   
 color = "black",   
 alpha = 0.8) +  
 scale\_fill\_gradient(low = "lightblue",   
 high = "darkred",   
 name = "Elderly Rate",   
 labels = percent) +  
 scale\_y\_continuous(  
 name = "Total Population",  
 labels = comma,  
 sec.axis = sec\_axis(~ . \* max(pa\_summary20$Avg\_Age) / max\_pop,   
 name = "Average Age")  
 ) +  
 labs(  
 title = "Population, Average Age (line), and Elderly Rate (dot color)",  
 x = "Planning Area"  
 ) +  
 theme\_minimal() +  
 theme(  
 axis.text.x = element\_text(angle = 45, hjust = 1),  
 axis.title.y.right = element\_text(color = "tomato"),  
 legend.position = c(0.85, 1.05),  
 legend.direction = "horizontal",  
 legend.justification = c("right", "top"),  
 legend.key.width = unit(1.2, "cm"),  
 legend.key.height = unit(0.4, "cm"),  
 legend.title = element\_text(size = 12),  
 legend.text = element\_text(size = 10),  
 plot.margin = margin(20, 20, 20, 20)  
 )  
  
  
offset <- 0.15  
  
pa <- ggplot(violin\_gender\_split, aes(x = PA,   
 y = Age,   
 fill = Sex)) +  
 geom\_half\_violin(data = subset(violin\_gender\_split,   
 Sex == "Males"),  
 side = "l",   
 alpha = 0.6,   
 trim = FALSE,   
 color = NA,   
 width = 1.2) +  
 geom\_half\_violin(data = subset(violin\_gender\_split,   
 Sex == "Females"),  
 side = "r",   
 alpha = 0.6,   
 trim = FALSE,   
 color = NA,   
 width = 1.2) +  
 geom\_boxplot(data = subset(violin\_gender\_split,   
 Sex == "Males"),  
 width = 0.15,   
 outlier.shape = NA,   
 alpha = 0.7,  
 position = position\_nudge(x = -offset),   
 color = "black") +  
 geom\_boxplot(data = subset(violin\_gender\_split,  
 Sex == "Females"),  
 width = 0.15, outlier.shape = NA,   
 alpha = 0.7,  
 position = position\_nudge(x = offset),   
 color = "black") +  
 stat\_summary(data = subset(violin\_gender\_split,   
 Sex == "Males"),  
 fun = median,   
 geom = "text",  
 aes(label = round(..y.., 0)),  
 position = position\_nudge(x = -offset),  
 vjust = -1.2, size = 3.5,   
 color = "black") +  
 stat\_summary(data = subset(violin\_gender\_split,   
 Sex == "Females"),  
 fun = median,   
 geom = "text",  
 aes(label = round(..y.., 0)),  
 position = position\_nudge(x = offset),  
 vjust = -1.2,   
 size = 3.5,   
 color = "black") +  
 scale\_fill\_manual(values = c("Males" = "#87CEFA",   
 "Females" = "#FFB6C1")) +  
 geom\_hline(yintercept = 65,   
 linetype = "dashed",   
 color = "darkred",   
 linewidth = 0.6) +  
 scale\_y\_continuous(  
 limits = c(0, 100),  
 breaks = seq(0, 100, by = 10)  
 ) +  
 labs(  
 title = "Sex-Split Age Distribution by Planning Area",  
 subtitle = "Left = Males | Right = Females | Dashed line = Age 65",  
 x = "Planning Area",  
 y = "Resident Age (Years)",  
 fill = "Sex"  
 ) +  
 theme\_minimal(base\_size = 14) +  
 theme(  
 axis.text.x = element\_text(angle = 45, hjust = 1),  
 axis.title = element\_text(size = 16),  
 plot.title = element\_text(size = 18, face = "bold"),  
 plot.subtitle = element\_text(size = 13),  
 legend.position = "top"  
 )  
(pb / pa) + plot\_layout(heights = c(1, 0.6))

|  |
| --- |
| Changes |
| Bubble Chart   1. The original bubble chart was replaced with a bar chart combined with a line chart, resulting in a clearer visual structure while still presenting the same three variables. 2. Instead of using the y-axis to directly show average age, the revised chart displays it as a red line with a corresponding secondary y-axis on the right, enhancing multidimensional comparison. 3. The elderly proportion, originally represented by bubble size, is now shown through the color intensity of the dots, accompanied by a continuous color legend that improves readability and comparison. 4. The x-axis, previously loosely sorted by total population, is now explicitly arranged from the most to the least populous planning area—from Tampines to Queenstown—making the visualisation more structured and interpretable.   Violin Plot   1. A gender dimension was added to the violin plot, making the visualisation more comprehensive and clearer by revealing differences in age distribution between males and females. |