Take-home\_Ex01\_Phase2

# Take-home Exercise 1 - Phase 2

I‘d like to provide my personal thoughts to our friend **Zhang Jinghan**’s work.

## Getting start

### **Installing and loading the required libraries**

pacman::p\_load(ggrepel, patchwork,   
 ggthemes, hrbrthemes,  
 tidyverse, ggiraph, plotly,   
 DT, readxl, gifski, gapminder,  
 gganimate, scales)

### **Importing data**

pop\_data <- read.csv("data/respopagesex2024.csv")  
  
sgresidence <- read.csv("data/respopagesex2024.csv")

## Good design principles

Here are the areas across **all three visuals** where she did very well and I should learn from:

1. Her insights into the data were very profound, and she did a great job transforming the data to uncover more observations. For example, she derived the Youth Dependency Ratio (YDR) and the Elderly Dependency Ratio (EDR) to provide deeper insights into the issue of population aging. In comparison, my own assignment, which only compared population distributions, was not as insightful.
2. She used colors very skillfully in her visualization—not just to make her work more visually appealing, but more importantly, to highlight contrasts between different groups. For example, she used red and blue to distinguish between male and female groups, and orange and purple to differentiate between younger and older age groups. These thoughtful color choices made her entire visualization much clearer and easier to understand.
3. Her choice of visualization types was highly appropriate—for instance, using bar charts to represent population distribution and a scatter quadrant map to show the distribution of YDR and EDR across subzones. Personally, I couldn’t think of a better choice of charts to convey the data more effectively.

## Further improvement

Following are my own ideas to adjust the visuals:

1. In her Visualization 1, the Population Pyramid, she presented a comprehensive view of the population structure of Singapore residents in 2024, broken down by gender and age group. However, from a reader’s perspective, the visualization contains too much information in a single chart, which may require extra effort to understand the relationships among population size, gender, age group, and percentage of the total population. In another word, the visual is not very straightforward. If it were up to me, I would present these pieces of information separately for greater clarity.

For the code, I’d like to simply “borrow” the approach used by our friend **Luo Yuming**, who streamlined the amount of information in the chart and represented the gender ratio comparison using an additional pie chart.

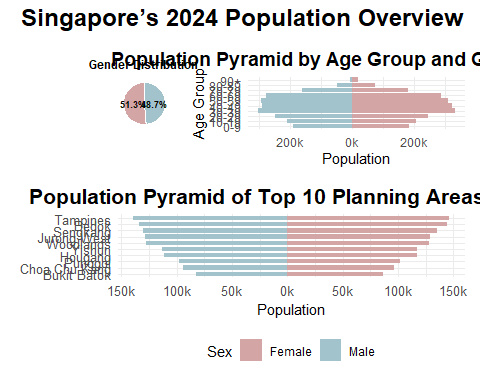
# Standardise column names  
pop\_data <- pop\_data %>%  
 rename(  
 Planning\_Area = PA,  
 Subzone = SZ,  
 Age = Age,  
 Sex = Sex,  
 Population = Pop,  
 Year = Time  
 ) %>%  
 filter(Year == 2024) %>%  
 mutate(  
 Age\_numeric = ifelse(Age == "90\_and\_Over", 90, as.integer(Age)),   
 Sex = ifelse(Sex == "Males", "Male", "Female")  
 )

Warning: There was 1 warning in `mutate()`.  
ℹ In argument: `Age\_numeric = ifelse(Age == "90\_and\_Over", 90,  
 as.integer(Age))`.  
Caused by warning in `ifelse()`:  
! NAs introduced by coercion

age\_distribution <- pop\_data %>%  
 filter(Sex %in% c("Male", "Female")) %>%  
 group\_by(Planning\_Area, Age\_numeric) %>%  
 summarise(Population = sum(Population), .groups = "drop") %>%  
 rename(Age = Age\_numeric)  
  
  
weighted\_median <- function(df) {  
 df <- df[order(df$Age), ]  
 cum\_pop <- cumsum(df$Population)  
 cutoff <- sum(df$Population) / 2  
 df$Age[which(cum\_pop >= cutoff)[1]]  
}  
  
  
median\_age\_summary <- age\_distribution %>%  
 group\_by(Planning\_Area) %>%  
 summarise(Median\_Age = weighted\_median(cur\_data\_all()))

Warning: There was 1 warning in `summarise()`.  
ℹ In argument: `Median\_Age = weighted\_median(cur\_data\_all())`.  
ℹ In group 1: `Planning\_Area = "Ang Mo Kio"`.  
Caused by warning:  
! `cur\_data\_all()` was deprecated in dplyr 1.1.0.  
ℹ Please use `pick()` instead.

gender\_summary <- pop\_data %>%  
 group\_by(Sex) %>%  
 summarise(Population = sum(Population), .groups = "drop") %>%  
 mutate(Percent = Population / sum(Population),  
 Label = paste0(percent(Percent, accuracy = 0.1)))   
  
  
p1 <- ggplot(gender\_summary, aes(x = "", y = Population, fill = Sex)) +  
 geom\_col(width = 1, color = "white") +  
 coord\_polar(theta = "y") +  
 geom\_text(aes(label = Label), position = position\_stack(vjust = 0.5),  
 size = 2.5, fontface = "bold", color = "black") +   
 scale\_fill\_manual(values = c("Male" = "#a3c3cc", "Female" = "#d4a5a5")) +  
 labs(title = "Gender Distribution") +  
 theme\_void() +  
 theme(  
 plot.margin = margin(10, 10, 10, 10),   
 legend.position = "none",  
 plot.title = element\_text(size = 9, face = "bold", hjust = 0.5)  
 )  
  
  
  
pyramid\_data <- pop\_data %>%  
 filter(Sex %in% c("Male", "Female")) %>%  
 mutate(AgeGroup = cut(Age\_numeric,  
 breaks = c(seq(0, 90, 10), Inf),  
 right = FALSE,  
 labels = c("0-9", "10-19", "20-29", "30-39",  
 "40-49", "50-59", "60-69", "70-79",  
 "80-89", "90+"))) %>%  
 group\_by(AgeGroup, Sex) %>%  
 summarise(Population = sum(Population), .groups = "drop") %>%  
 mutate(Population = ifelse(Sex == "Male", -Population, Population))  
  
  
  
p2 <- ggplot(pyramid\_data, aes(x = Population, y = AgeGroup, fill = Sex)) +  
 geom\_col(width = 0.9) +  
 scale\_x\_continuous(labels = function(x) paste0(abs(x / 1000), "k")) +  
 scale\_fill\_manual(values = c("Male" = "#a3c3cc", "Female" = "#d4a5a5")) +  
 labs(  
 title = "Population Pyramid by Age Group and Gender (2024)",  
 x = "Population",  
 y = "Age Group"  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(size = 14, face = "bold", hjust = 0.5),  
 legend.position = "none"  
 )  
  
  
top10\_pa\_names <- pop\_data %>%  
 group\_by(Planning\_Area) %>%  
 summarise(Total = sum(Population), .groups = "drop") %>%  
 arrange(desc(Total)) %>%  
 slice\_head(n = 10) %>%  
 pull(Planning\_Area)  
  
  
top10\_pyramid <- pop\_data %>%  
 filter(Planning\_Area %in% top10\_pa\_names, Sex %in% c("Male", "Female")) %>%  
 group\_by(Planning\_Area, Sex) %>%  
 summarise(Population = sum(Population), .groups = "drop") %>%  
 mutate(  
 Population = ifelse(Sex == "Male", -Population, Population),  
 Planning\_Area = factor(Planning\_Area, levels = rev(top10\_pa\_names))  
 )  
  
  
  
p3 <-ggplot(top10\_pyramid, aes(x = Population, y = Planning\_Area, fill = Sex)) +  
 geom\_col(width = 0.7) +  
 scale\_x\_continuous(labels = function(x) paste0(abs(x / 1000), name = "k")) +  
 scale\_fill\_manual(values = c("Male" = "#a3c3cc", "Female" = "#d4a5a5")) +  
 labs(  
 title = "Population Pyramid of Top 10 Planning Areas (2024)",  
 y = NULL  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(face = "bold", size = 16, hjust = 0.5),  
 axis.text = element\_text(size = 10),  
 legend.position = "bottom"  
 )  
  
  
(p1 + p2) / p3 +  
 plot\_layout(  
 widths = c(1, 2),   
 heights = c(1, 1.2)   
 ) +  
 plot\_annotation(  
 title = "Singapore’s 2024 Population Overview",  
 theme = theme(  
 plot.title = element\_text(size = 18, face = "bold", hjust = 0.5)  
 )  
 )



1. For visualization 2, I’d like to make a very simple adjustment which can directly hit the point, which is add a label to each zone on the chart. If required, we may also make this to an interactive chart.

My shot:

sgresidence <- sgresidence %>%  
 mutate(age = as.integer(Age))

Warning: There was 1 warning in `mutate()`.  
ℹ In argument: `age = as.integer(Age)`.  
Caused by warning:  
! NAs introduced by coercion

sgresidence %>%  
 filter(is.na(as.integer(Age))) %>%  
 distinct(Age)

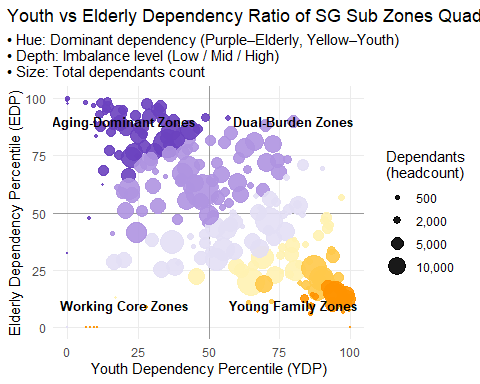
Warning: There was 1 warning in `filter()`.  
ℹ In argument: `is.na(as.integer(Age))`.  
Caused by warning:  
! NAs introduced by coercion

Age  
1 90\_and\_Over

sgresidence %>%  
 filter(Age == "90\_and\_Over") %>%  
 summarise(total\_90plus = sum(Pop)) %>%  
 mutate(total\_pop = sum(sgresidence$Pop),  
 percent = total\_90plus / total\_pop \* 100)

total\_90plus total\_pop percent  
1 25290 4193530 0.6030719

sgresidence <- sgresidence %>%  
 mutate(  
 Age = ifelse(Age == "90\_and\_Over", "90", Age),  
 age = as.integer(Age)  
 )  
  
sgresidence <- sgresidence %>%  
 filter(Pop > 0)  
  
## 1 ────────── Calculate ratios  
dep\_sz <- sgresidence %>%   
 mutate(group = case\_when(  
 age < 15 ~ "Youth",  
 age >= 65 ~ "Elderly",  
 TRUE ~ "Working")) %>%   
 group\_by(SZ, group) %>%   
 summarise(Pop = sum(Pop), .groups = "drop") %>%   
 pivot\_wider(names\_from = group, values\_from = Pop, values\_fill = 0) %>%   
 mutate(  
 YDR = Youth / Working,  
 EDR = Elderly / Working,  
 Dependants = Youth + Elderly,  
 grp = if\_else(YDR > EDR, "Y\_dom", "O\_dom"),  
 log\_ratio = log(pmax(EDR, 1e-6) / pmax(YDR, 1e-6)), # avoid log(0)  
 YDR\_pct = percent\_rank(YDR) \* 100,  
 EDR\_pct = percent\_rank(EDR) \* 100  
 )  
  
## 2 ────────── Discretize log\_ratio into "Low" / "Mid" / "High"  
dep\_plot <- dep\_sz %>%  
 mutate(  
 imbalance = cut(abs(log\_ratio),  
 breaks = quantile(abs(log\_ratio), probs = c(0, 1/3, 2/3, 1)),  
 labels = c("Low", "Mid", "High"),  
 include.lowest = TRUE))  
  
## 3 ────────── Define stepped color table  
col\_tbl <- tribble(  
 ~grp, ~imbalance, ~col,  
 "Y\_dom", "Low", "#FFF2B2",  
 "Y\_dom", "Mid", "#FFC94A",  
 "Y\_dom", "High", "#FF9300",  
 "O\_dom", "Low", "#E3DFF5",  
 "O\_dom", "Mid", "#AF94E1",  
 "O\_dom", "High", "#6B42BF"  
)  
  
## 4 ────────── Merge color  
dep\_plot <- dep\_plot %>%  
 left\_join(col\_tbl, by = c("grp", "imbalance"))  
  
## 5 ────────── Plot  
ggplot(dep\_plot, aes(YDR\_pct, EDR\_pct)) +  
 geom\_hline(yintercept = 50, color = "gray60") +  
 geom\_vline(xintercept = 50, color = "gray60") +  
 geom\_point(aes(size = Dependants, color = col), alpha = 0.9) +  
 scale\_colour\_identity() +  
 scale\_size\_area(max\_size = 12,  
 breaks = c(500, 2000, 5000, 10000),  
 labels = scales::comma,  
 name = "Dependants\n(headcount)") +  
 labs(  
 title = "Youth vs Elderly Dependency Ratio of SG Sub Zones Quadrants (2024)",  
 subtitle = "• Hue: Dominant dependency (Purple–Elderly, Yellow–Youth) \n• Depth: Imbalance level (Low / Mid / High) \n• Size: Total dependants count",  
 x = "Youth Dependency Percentile (YDP)",  
 y = "Elderly Dependency Percentile (EDP)"  
 ) +  
   
 annotate("text", x = 80, y = 90, label = "Dual-Burden Zones", fontface = "bold", size = 3.5, hjust = 0.5) +  
 annotate("text", x = 80, y = 10, label = "Young Family Zones", fontface = "bold", size = 3.5, hjust = 0.5) +  
 annotate("text", x = 20, y = 90, label = "Aging-Dominant Zones", fontface = "bold", size = 3.5, hjust = 0.5) +  
 annotate("text", x = 20, y = 10, label = "Working Core Zones", fontface = "bold", size = 3.5, hjust = 0.5) +  
 theme\_minimal(base\_size = 11) +  
 theme(  
 panel.grid.minor = element\_blank(),  
 plot.title.position = "plot"  
 )



1. For Visualization 3, I noticed that a): Median lines for both the EDR and YDR are presented, they are not labeled, which may cause confusion. I would recommend adding clear labels to these reference lines. b): It would be helpful to highlight planning areas where EDR exceed the median values, as these zones may warrant closer attention. c): It’s good to show the median EDR and YDR in value so that readers can understand the situation more clearly.

My shot:

# 1. Categorize by Age Group and summarize population  
df\_agegroup <- sgresidence %>%  
 mutate(AgeGroup = case\_when(  
 Age < 15 ~ "Youth",  
 Age <= 64 ~ "WorkingAge",  
 TRUE ~ "Elderly"  
 )) %>%  
 group\_by(PA, AgeGroup) %>%  
 summarise(Pop = sum(Pop), .groups = "drop") %>%  
 pivot\_wider(names\_from = AgeGroup, values\_from = Pop, values\_fill = 0) %>%  
 mutate(  
 TotalPop = Youth + WorkingAge + Elderly,  
 ElderlyRatio = Elderly / WorkingAge,  
 YouthRatio = Youth / WorkingAge  
 )  
  
# 2. Reshape to long format for stacked bar chart  
df\_age\_long <- df\_agegroup %>%  
 pivot\_longer(cols = c("Youth", "WorkingAge", "Elderly"),  
 names\_to = "AgeGroup", values\_to = "Pop") %>%  
 mutate(PA = fct\_reorder(PA, -Pop, .fun = sum))  
  
# 3. Compute median values and max population  
median\_edr <- median(df\_agegroup$ElderlyRatio, na.rm = TRUE)  
median\_ydr <- median(df\_agegroup$YouthRatio, na.rm = TRUE)  
max\_pop <- max(df\_agegroup$TotalPop, na.rm = TRUE)  
  
# 4. Add star marker to EDR above median  
df\_agegroup <- df\_agegroup %>%  
 mutate(EDR\_star = if\_else(ElderlyRatio > median\_edr, "\*", ""))  
  
# 5. Define color palette  
custom\_colors <- c(  
 "Youth" = "#FFD700", # Gold  
 "WorkingAge" = "#B0B0B0", # Gray  
 "Elderly" = "#800080" # Purple  
)  
  
# 6. Get rightmost x position  
last\_x <- length(unique(df\_age\_long$PA))  
  
# 7. Create the plot  
ggplot(df\_age\_long, aes(x = PA, y = Pop, fill = AgeGroup)) +  
 geom\_bar(stat = "identity", width = 0.95) +  
  
 # Elderly Dependency Ratio line  
 geom\_line(  
 data = df\_agegroup,  
 aes(x = PA, y = ElderlyRatio \* max\_pop / 2, group = 1),  
 inherit.aes = FALSE, color = "#9932CC", size = 1.2  
 ) +  
  
 # Youth Dependency Ratio line  
 geom\_line(  
 data = df\_agegroup,  
 aes(x = PA, y = YouthRatio \* max\_pop / 2, group = 1),  
 inherit.aes = FALSE, color = "#E67E22", size = 1.2  
 ) +  
  
 # Red stars for high EDR  
 geom\_text(  
 data = df\_agegroup %>% filter(EDR\_star == "\*"),  
 aes(x = PA, y = ElderlyRatio \* max\_pop / 2, label = EDR\_star),  
 inherit.aes = FALSE, color = "red", size = 5, vjust = -0.8  
 ) +  
  
 # Median lines  
 geom\_hline(yintercept = median\_edr \* max\_pop / 2,  
 color = "black", linetype = "solid", size = 0.5) +  
 geom\_hline(yintercept = median\_ydr \* max\_pop / 2,  
 color = "black", linetype = "solid", size = 0.5) +  
  
 # Median line labels (right)  
 annotate("text", x = last\_x, y = median\_edr \* max\_pop / 2,  
 label = "Median EDR", vjust = -1, hjust = 1, size = 3, color = "black") +  
 annotate("text", x = last\_x, y = median\_ydr \* max\_pop / 2,  
 label = "Median YDR", vjust = 1.5, hjust = 1, size = 3, color = "black") +  
  
 # ➕ EDR/YDR values in red box area  
 annotate("text", x = last\_x - 10, y = max\_pop \* 0.9,  
 label = paste0("Median EDR = ", round(median\_edr, 2)),  
 hjust = 0, size = 2, color = "#9932CC") +  
 annotate("text", x = last\_x - 10, y = max\_pop \* 0.8,  
 label = paste0("Median YDR = ", round(median\_ydr, 2)),  
 hjust = 0, size = 2, color = "#E67E22") +  
  
 # Final touches  
 scale\_fill\_manual(values = custom\_colors) +  
 scale\_y\_continuous(  
 name = "Population Count",  
 sec.axis = sec\_axis(~ . \* 2 / max\_pop, name = "Dependency Ratio"),  
 labels = label\_number(scale\_cut = cut\_short\_scale())  
 ) +  
 labs(  
 title = "Population by Planning Area with Youth & Elderly Dependency Ratios",  
 subtitle = "\* indicates Planning Areas with Elderly Dependency Ratio above the national median",  
 x = "Planning Area",  
 fill = "Age Group"  
 ) +   
   
 theme\_minimal() +  
 theme(  
 axis.text.x = element\_text(angle = 45, hjust = 1),  
 legend.position = "right"  
 )

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.

