2022 Australian Federal Elections Analysis

Bhavya Bhargava

2024-09-19

The 2022 Australian federal election marked a turning point in voter behavior, with nearly one in three voters supporting minor parties or independent candidates, the highest level since the 1930s. This surge in independent support, especially as traditional parties like the Liberals and Labor experienced historic declines, highlights a growing shift towards alternative voices in politics.

For Environmental groups like the Climate2000 group, understanding this shift is critical for analyzing how independents can influence key issues like climate policy.

Assuming the role of an independent analyst for the group, below is my analysis of the voter demographics through various electorates and the state of Victoria.

Importing the packages that were utilised

```r
library(tidyverse)
library(sf)
library(ggplot2)
library(readxl)
```

Analysis of the electorates won by indeendent candidates in 2022 federal elections

For performing the analysis of the 2022 federal elections, data was sourced from the Australian Electoral Commission’s Tally Room for the required elections. Filtering of the data was done on the basis of the understanding that ‘PartyAb’ i.e. Party Abbreviation for the Independent candidates is “IND” and they got elected.

The data was sourced the following way: 1. Visited results.ac.gov.au and under full federal elections in the Tally room archive section clicked on the 2022 federal elections. 2. On the page which opened I clicked on the downloads button in the left pane and so a list of different data sets came up. 3. Then under the Distribution and flow of preferences section, I clicked on “Distribution of preferences by candidate by division” to get the relevant data.

The link to the resource is attached below: https://results.aec.gov.au/27966/Website/Downloads/HouseDopByDivisionDownload-27966.csv (https://results.aec.gov.au/27966/Website/Downloads/HouseDopByDivisionDownload-27966.csv)

Please find the code and result for the same below:

```r
# Loading election results data from local repo
votes <- read_csv(here::here("data/HouseDopByDivisionDownload-27966.csv"), skip = 1)

# Filtering rows where independents got elected
independent_wins <- votes %>%
    filter(PartyAb == "IND" & Elected == "Y")

# Printing just the unique electorate names
print(paste("2022 election Independent won elctorates:",unique(independent_wins$DivisionNm)))
```

```
##  [1] "2022 election Independent won elctorates: Fowler"
##  [2] "2022 election Independent won elctorates: Mackellar"
##  [3] "2022 election Independent won elctorates: North Sydney"
##  [4] "2022 election Independent won elctorates: Warringah"
##  [5] "2022 election Independent won elctorates: Wentworth"
##  [6] "2022 election Independent won elctorates: Clark"
##  [7] "2022 election Independent won elctorates: Goldstein"
##  [8] "2022 election Independent won elctorates: Indi"
##  [9] "2022 election Independent won elctorates: Kooyong"
## [10] "2022 election Independent won elctorates: Curtin"
```

Analysis and count of the Liberal seats lost to independents in 2022 federal elections

In continuing our examination of the 2022 federal election results, we focus on the electorates that were formerly held by the Liberal Party (represented by the abbreviation “LP”) but were not retained in the recent election, and whether they were lost to Independent candidates.

The data was handeled in the following way:

1. The same dataset from the Australian Electoral Commission’s Tally Room was utilized, as it offers a comprehensive overview of the election results, including the historical context of which party previously held each seat.
2. The analysis is based on the assumption that electorates where the HistoricElected field is “Y” (indicating a prior victory) and the PartyAb for the Liberal party is “LP” reflect seats formerly held by the Liberal party.

Furthermore, to identify these changes, the dataset was filtered for those electorates where the Liberal Party did not win in the 2022 elections (Elected == “N”), yet had historically been successful (HistoricElected == “Y”).

Please find the code and result for the same below:

```
# For demonstration, assuming `HistoricElected` shows previous winners being Liberal
formerly_liberal <- votes %>%
    filter(Elected == "N" & PartyAb == "LP" & HistoricElected == "Y")  # PartyAb must be confirmed

#Comparing ex_liberal and currently independent electorates
ex_liberal_electorates <- unique(formerly_liberal$DivisionNm)

independent_electorates <- unique(independent_wins$DivisionNm)

lost_electorates <- intersect(ex_liberal_electorates, independent_electorates)

# Printing the number of electorates Liberals lost to Independents
print(paste("Number of electorates lost to Independents:", length(lost_electorates)))
```

```
## [1] "Number of electorates lost to Independents: 6"
```

```
print(paste("Names of electorates:", lost_electorates))
```

```
## [1] "Names of electorates: Mackellar"    "Names of electorates: North Sydney"
## [3] "Names of electorates: Wentworth"    "Names of electorates: Goldstein"
## [5] "Names of electorates: Kooyong"      "Names of electorates: Curtin"
```

Relative analysis of an Independent Electorate:

For this analysis I've chosen the independent electorate of Kooyong which has proven to be a major win for independence from liberals in 2022.

Steps taken to acquire the data:

For getting the data, I visited the ABS website and acquired the relevant census data pack. The specifications are as below: Census year: 2021 DataPack type: General Community Profile Geography: Commonwealth Electoral Divisions (CED) From here I downloaded the data for Victoria. The data was downloaded from this link: https://www.abs.gov.au/census/find-census-data/datapacks/download/2021_GCP_CED_for_VIC_short-header.zip (https://www.abs.gov.au/census/find-census-data/datapacks/download/2021_GCP_CED_for_VIC_short-header.zip)

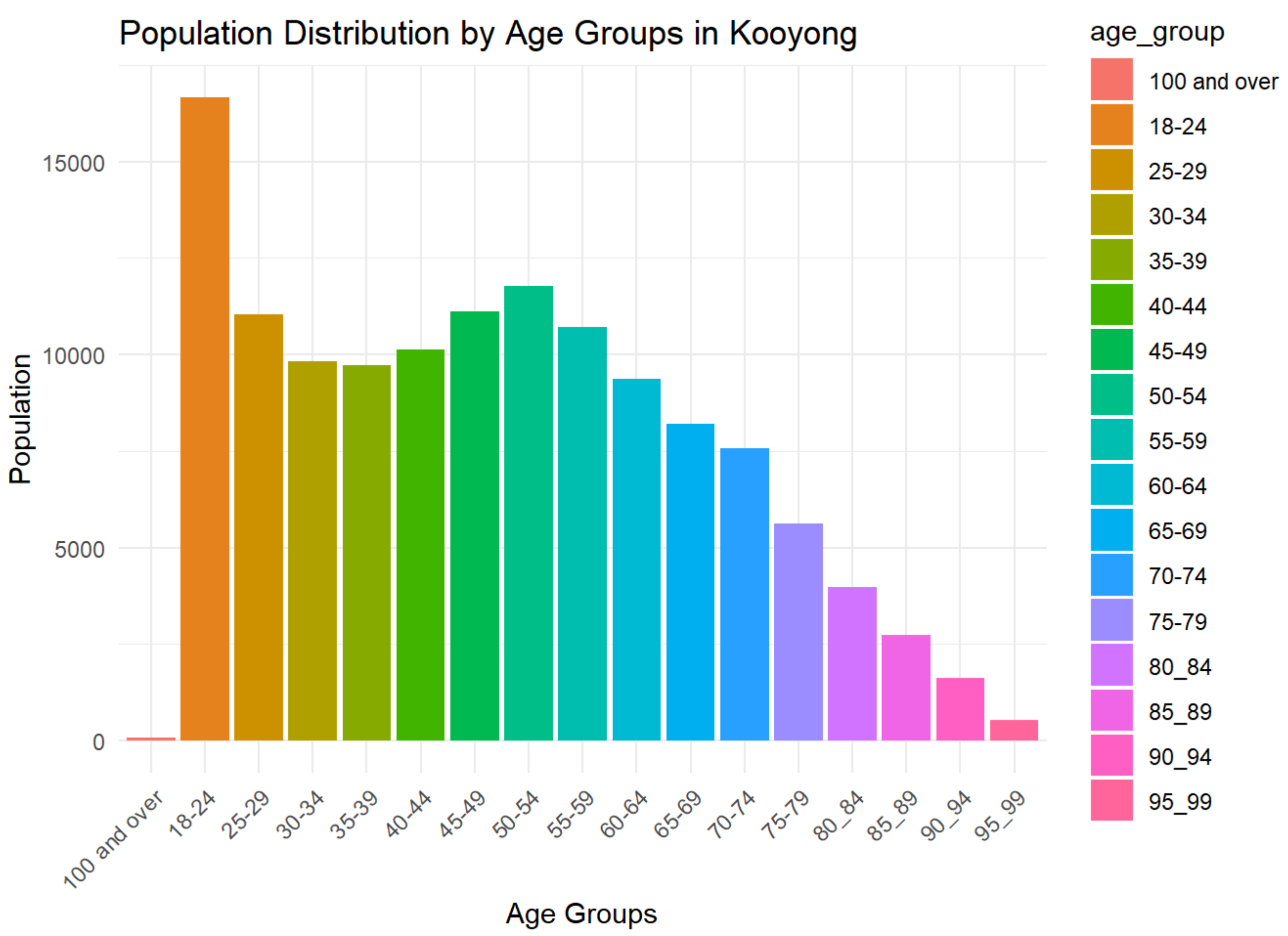
Furthermore to do the required analysis I utilized the G04 and G017 data sets from the archive for the state and the CED of Kooyong which I found from: https://abs.gov.au/census/find-census-data/quickstats/2021/CED226 (https://abs.gov.au/census/find-census-data/quickstats/2021/CED226)

Plot for age distribution of voters across the electorate and state

```
# Loading data for the dataset from the local repo
g04_part1 <- read.csv(here::here("data/2021_Census_GCP_Commonwealth_Electroral_Division_for_VIC/2021Census_G04A_VIC_CED.cs
v"))
g04_part2 <- read.csv(here::here("data/2021_Census_GCP_Commonwealth_Electroral_Division_for_VIC/2021Census_G04B_VIC_CED.cs
v"))
```

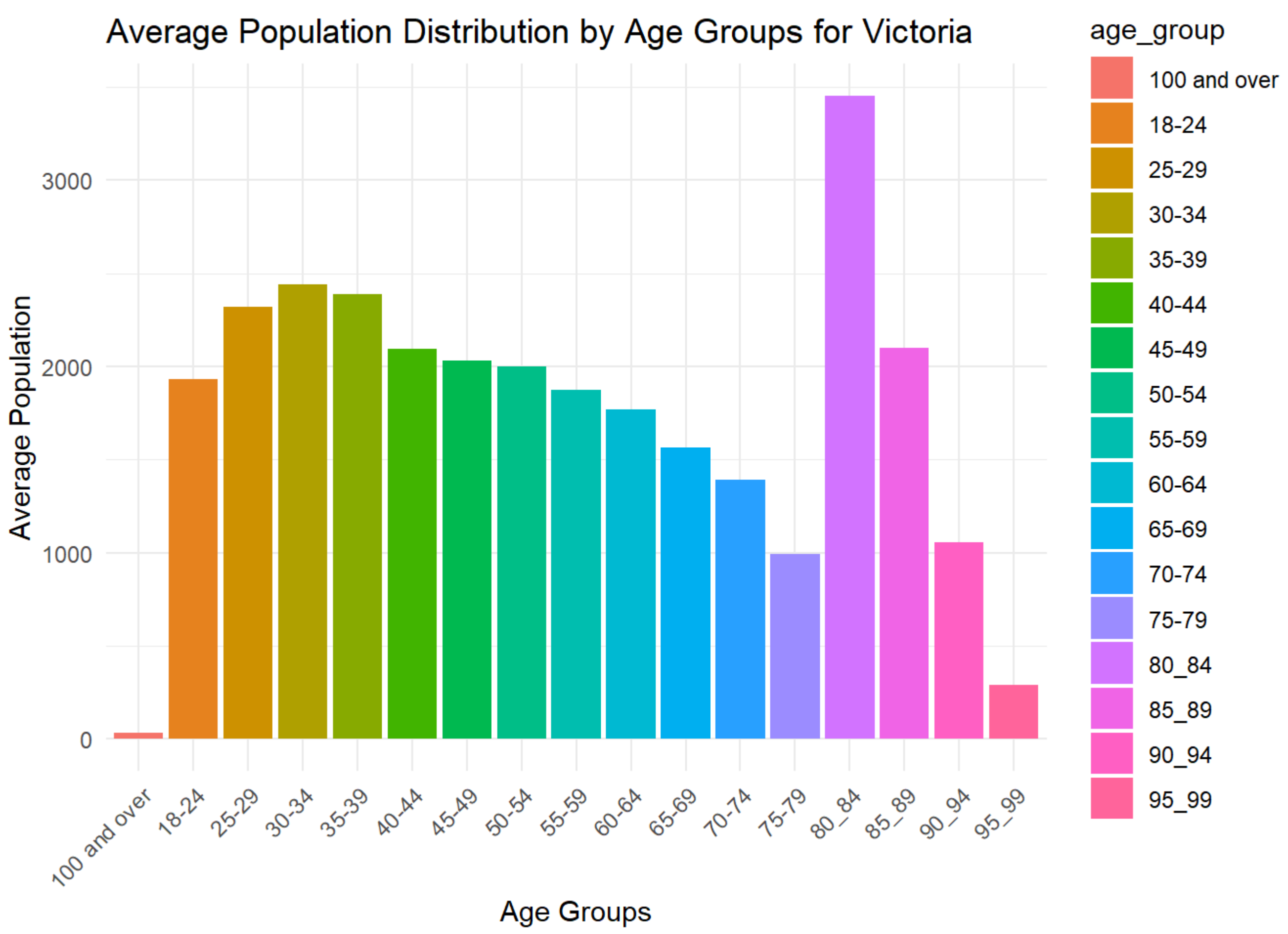
For Kooyong

```
# For the Kooyong Age plot
# Filtering and binding the datasets
g04_data_combined <- bind_rows(
  g04_part1 %>% filter(CED_CODE_2021 == "CED226"),
  g04_part2 %>% filter(CED_CODE_2021 == "CED226")
)

# Transforming data to long format and filtering out unnecessary columns
g04_data_long <- g04_data_combined %>%
  pivot_longer(cols = starts_with("Age_yr_"), names_to = "age_group", values_to = "population") %>%
  mutate(age_group = str_remove(age_group, "_P"),  # Remove suffix
         age_group = str_replace(age_group, "Age_yr_100_yr_over", "100 and over"),
         age_group = str_replace(age_group, "Age_yr_", "")) %>%
  # Removing unnecessary columns
  filter(!str_detect(age_group, "_F|_M"),
         !str_detect(age_group, "0_17|18_")) %>%
  mutate(age_group = case_when(
    age_group == "18" ~ "18-24",
    age_group %in% c("19", "20", "21", "22", "23", "24") ~ "18-24",
    age_group %in% c("25", "26", "27", "28", "29") ~ "25-29",
    age_group %in% c("30", "31", "32", "33", "34") ~ "30-34",
    age_group %in% c("35", "36", "37", "38", "39") ~ "35-39",
    age_group %in% c("40", "41", "42", "43", "44") ~ "40-44",
    age_group %in% c("45", "46", "47", "48", "49") ~ "45-49",
    age_group %in% c("50", "51", "52", "53", "54") ~ "50-54",
    age_group %in% c("55", "56", "57", "58", "59") ~ "55-59",
    age_group %in% c("60", "61", "62", "63", "64") ~ "60-64",
    age_group %in% c("65", "66", "67", "68", "69") ~ "65-69",
    age_group %in% c("70", "71", "72", "73", "74") ~ "70-74",
    age_group %in% c("75", "76", "77", "78", "79") ~ "75-79",
    str_detect(age_group, "80_84|85_89|90_94|95_99|100 and over") ~ age_group
  )) %>%
  # Remove rows where age_group is NA due to the case_when conditions
  drop_na(age_group)

# Sum up populations within each age group
kooyong_age_data <- g04_data_long %>%
  group_by(age_group) %>%
  summarise(total_population = sum(population, na.rm = TRUE))

# Plotting the data
ggplot(kooyong_age_data, aes(x = age_group, y = total_population, fill = age_group)) +
  geom_col() +
  labs(title = "Population Distribution by Age Groups in Kooyong", x = "Age Groups", y = "Population") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



For Victoria

```r
# For the Victoria Age plot
# Combining the datasets without filtering for a specific CED_CODE_2021
g04_data_combined <- bind_rows(
  g04_part1,
  g04_part2
)

# Transforming data to long format
g04_data_long <- g04_data_combined %>%
  pivot_longer(cols = starts_with("Age_yr_"), names_to = "age_group", values_to = "population") %>%
  mutate(age_group = str_remove(age_group, "_P"),  # Remove suffix
         age_group = str_replace(age_group, "Age_yr_100_yr_over", "100 and over"),
         age_group = str_replace(age_group, "Age_yr_", "")) %>%
  # Removing gender-specific and under-18 columns
  filter(!str_detect(age_group, "_F|_M"),
         !str_detect(age_group, "0_17|18_")) %>%
  mutate(age_group = case_when(
    age_group == "18" ~ "18-24",
    age_group %in% c("19", "20", "21", "22", "23", "24") ~ "18-24",
    age_group %in% c("25", "26", "27", "28", "29") ~ "25-29",
    age_group %in% c("30", "31", "32", "33", "34") ~ "30-34",
    age_group %in% c("35", "36", "37", "38", "39") ~ "35-39",
    age_group %in% c("40", "41", "42", "43", "44") ~ "40-44",
    age_group %in% c("45", "46", "47", "48", "49") ~ "45-49",
    age_group %in% c("50", "51", "52", "53", "54") ~ "50-54",
    age_group %in% c("55", "56", "57", "58", "59") ~ "55-59",
    age_group %in% c("60", "61", "62", "63", "64") ~ "60-64",
    age_group %in% c("65", "66", "67", "68", "69") ~ "65-69",
    age_group %in% c("70", "71", "72", "73", "74") ~ "70-74",
    age_group %in% c("75", "76", "77", "78", "79") ~ "75-79",
    str_detect(age_group, "80_84|85_89|90_94|95_99|100 and over") ~ age_group
  ))%>%
  # Filter out rows with NA in the population
  filter(!is.na(age_group), !is.na(population))

# Calculate the average population within each age group
victoria_average_age_data <- g04_data_long %>%
  group_by(age_group) %>%
  summarise(average_population = mean(population, na.rm = TRUE))%>%
  # Remove groups with NA average_population
  filter(!is.na(average_population))

# Plotting the data
ggplot(victoria_average_age_data, aes(x = age_group, y = average_population, fill = age_group)) +
  geom_col() +
  labs(title = "Average Population Distribution by Age Groups for Victoria", x = "Age Groups", y = "Average Population") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

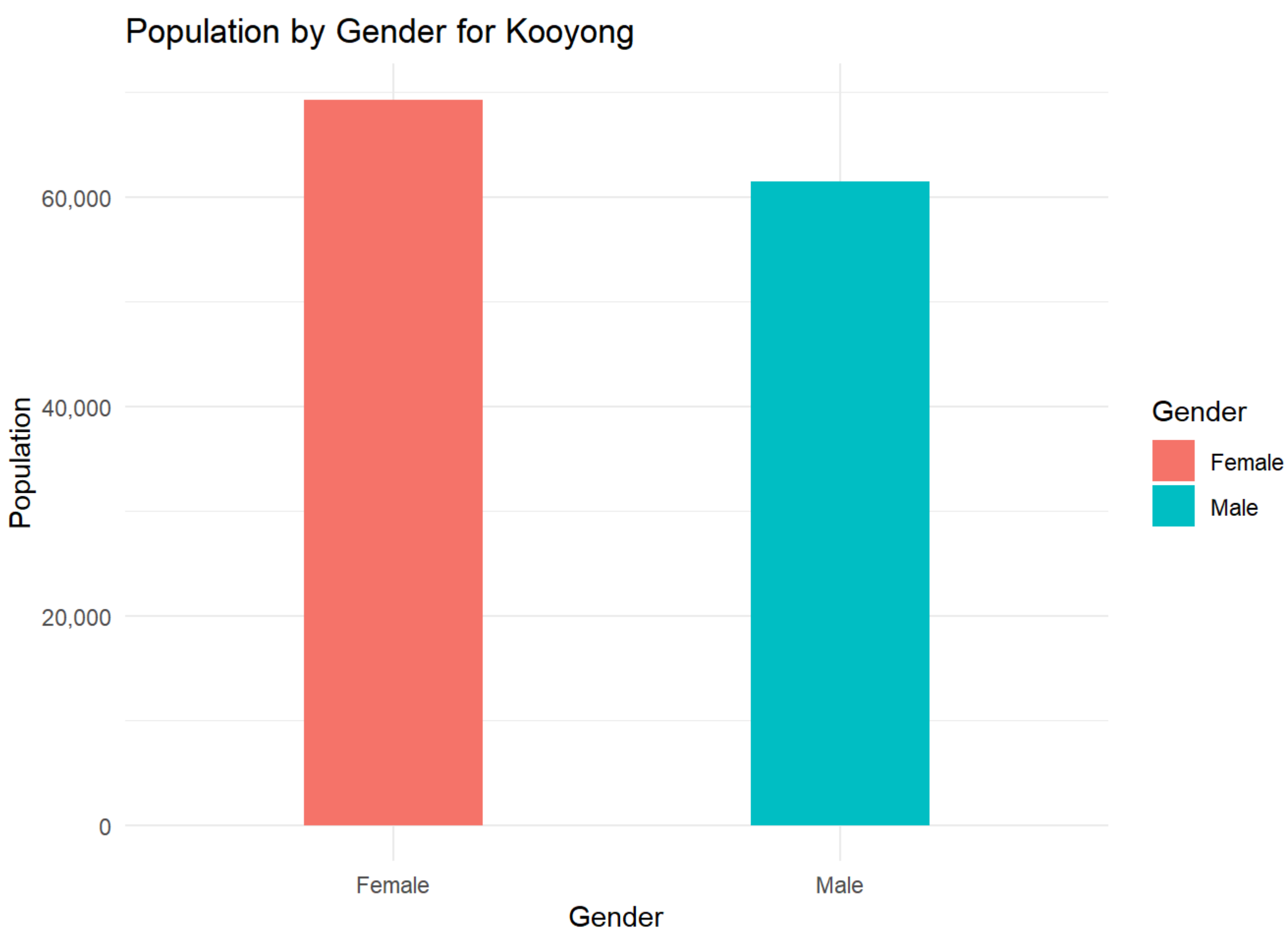


Analysis: In Kooyong as it is a bustling part of Melbourne there are more people of younger ages whereas generally is Victoria there are more older people.

Plot for age distribution of voters across the electorate and state

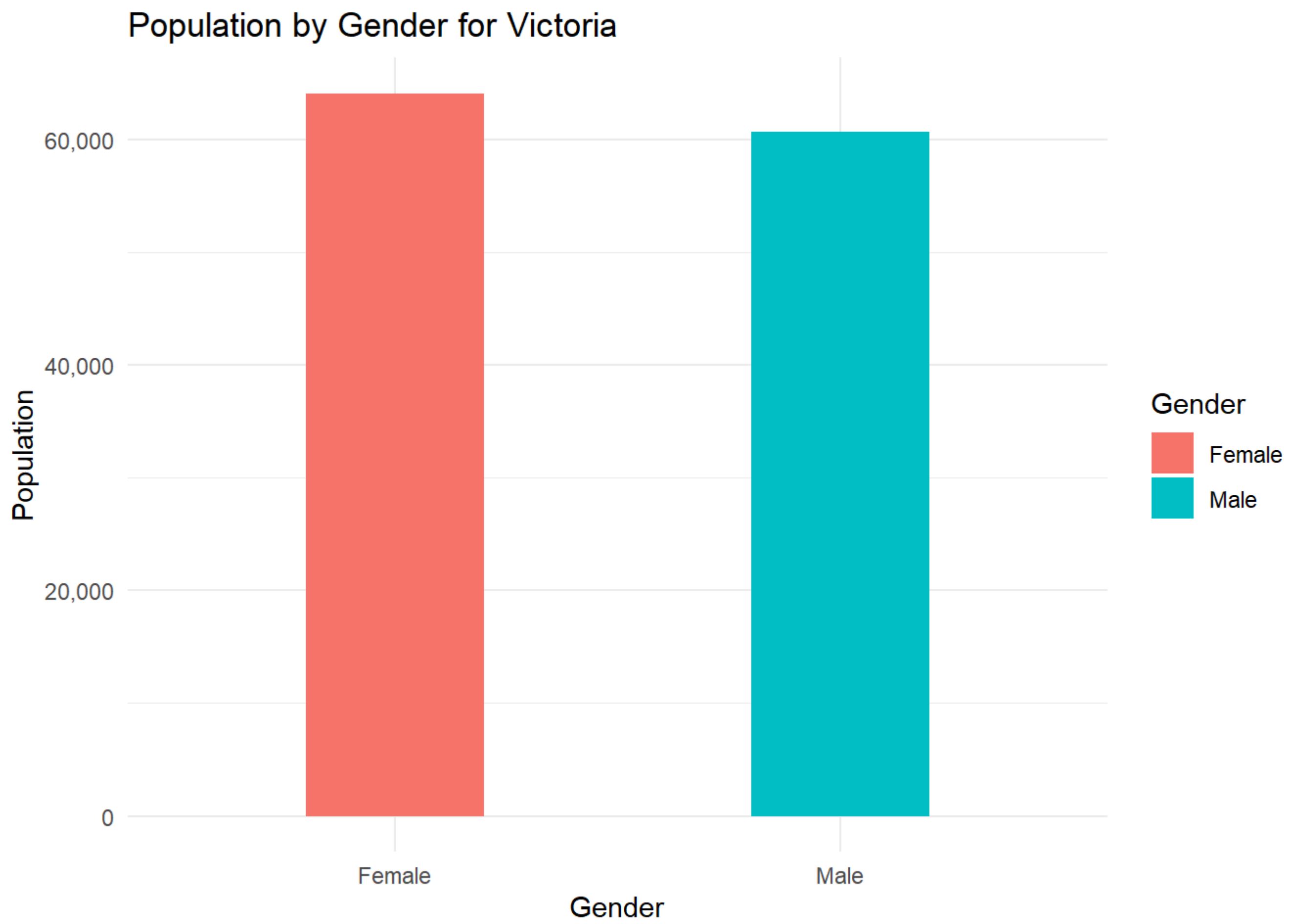
For Kooyong

```
# Data taken out for Kooyong
kooyong_population_data_1 <- g04_part1 %>%
  filter(CED_CODE_2021 == "CED226")
kooyong_population_data_2 <- g04_part2 %>%
  filter(CED_CODE_2021 == "CED226")

# Summarizing the population data for both the genders
population_up_to_17_male <- kooyong_population_data_1 %>%
  select(matches("^Age_yr_([0-9]|1[0-7])_M$")) %>%
  unlist() %>%
  sum(na.rm = TRUE)

population_up_to_17_female <- kooyong_population_data_1 %>%
  select(matches("^Age_yr_([0-9]|1[0-7])_F$")) %>%
  unlist() %>%
  sum(na.rm = TRUE)

# Calculating the sum of population for each gender
total_male_population <- sum(kooyong_population_data_2 %>% select(starts_with("Tot_M")) %>% unlist(), na.rm = TRUE)
total_female_population <- sum(kooyong_population_data_2 %>% select(starts_with("Tot_F")) %>% unlist(), na.rm = TRUE)

# Adjusting the total populations by subtracting the populations up to age 17
adjusted_total_male_population <- total_male_population - population_up_to_17_male
adjusted_total_female_population <- total_female_population - population_up_to_17_female

# Creating a data frame for plotting
kooyong_gender_population <- data.frame(
  Gender = c("Male", "Female"),
  Adjusted_Population_Data = c(adjusted_total_male_population, adjusted_total_female_population)
)

# Plotting the bar chart
ggplot(kooyong_gender_population, aes(x = Gender, y = Adjusted_Population_Data, fill = Gender)) +
  geom_col(width = 0.4) +
  scale_y_continuous(labels = scales::comma) + # Adding comma formatting to the y-axis labels
  labs(x = "Gender", y = "Population", title = "Population by Gender for Kooyong") +
  theme_minimal()
```



For Victoria

```
# Data taken out for Victoria
victoria_population_data_1 <- g04_part1
victoria_population_data_2 <- g04_part2

# Summarizing the population data for both the genders
population_up_to_17_male <- victoria_population_data_1 %>%
  select(matches("^Age_yr_([0-9]|1[0-7])_M$")) %>%
  unlist() %>%
  sum(na.rm = TRUE)

population_up_to_17_female <- victoria_population_data_1 %>%
  select(matches("^Age_yr_([0-9]|1[0-7])_F$")) %>%
  unlist() %>%
  sum(na.rm = TRUE)

# Calculating the sum of population for each gender
total_male_population <- sum(victoria_population_data_2 %>% select(starts_with("Tot_M")) %>% unlist(), na.rm = TRUE)
total_female_population <- sum(victoria_population_data_2 %>% select(starts_with("Tot_F")) %>% unlist(), na.rm = TRUE)

# Adjusting the total populations by subtracting the populations up to age 17
adjusted_total_male_population <- total_male_population - population_up_to_17_male
adjusted_total_female_population <- total_female_population - population_up_to_17_female

# Calculating the average of the adjusted total populations for the whole of Victoria
# Dividing by the number of data entries for a true average
avg_adjusted_male_population <- adjusted_total_male_population / nrow(g04_part2)
avg_adjusted_female_population <- adjusted_total_female_population / nrow(g04_part2)

# Creating a data frame for plotting
victoria_gender_population <- data.frame(
  Gender = c("Male", "Female"),
  Adjusted_Average_Population = c(avg_adjusted_male_population, avg_adjusted_female_population)
)

# Plot the bar chart
ggplot(victoria_gender_population, aes(x = Gender, y = Adjusted_Average_Population, fill = Gender)) +
  geom_col(width = 0.4) +
  scale_y_continuous(labels = scales::comma) +
  labs(x = "Gender", y = "Population", title = "Population by Gender for Victoria") +
  theme_minimal()
```



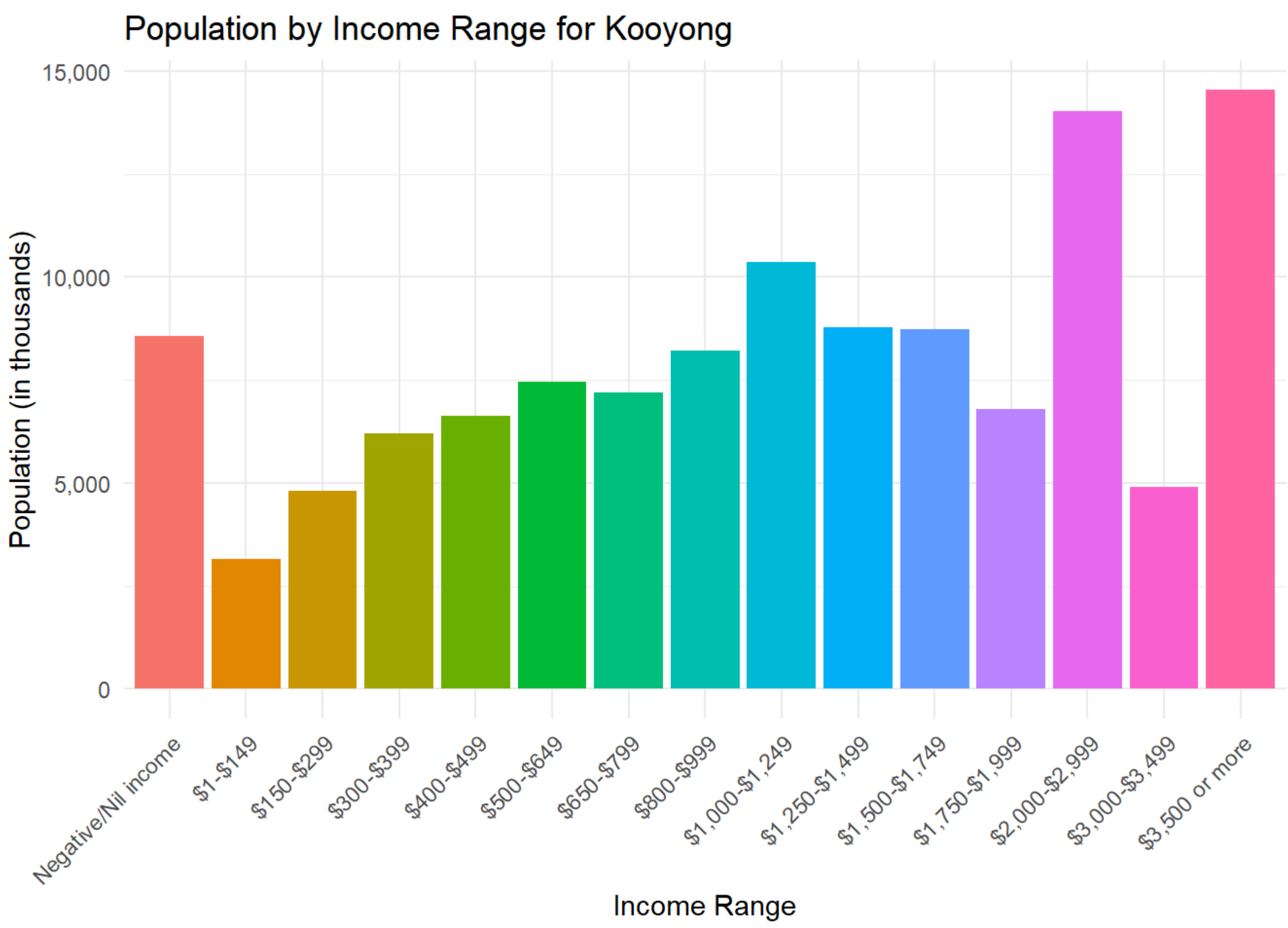
Analysis: Here we see Kooyong follows suite with rest of Victoria and has more female population and compared to male.

Plot for income distribution of voters across the electorate and state

```
# Taking relevant income data from the local repo
g17_part1 <- read.csv(here::here("data/2021_Census_GCP_Commonwealth_Electoral_Division_for_VIC/2021Census_G17B_VIC_CED.cs
v"))
g17_part2 <- read.csv(here::here("data/2021_Census_GCP_Commonwealth_Electoral_Division_for_VIC/2021Census_G17C_VIC_CED.cs
v"))
```

For Kooyong

```r
# Filtering data relevant to Kooyong
kooyong_data_combined <- bind_rows(
  g17_part1 %>%
  filter(CED_CODE_2021 == "CED226"),
  g17_part2 %>%
  filter(CED_CODE_2021 == "CED226")
)

# Constructing a mapping based on the income ranges
ranges <- c("Negative/Nil income", "$1-$149", "$150-$299", "$300-$399", "$400-$499",
            "$500-$649", "$650-$799", "$800-$999", "$1,000-$1,249", "$1,250-$1,499",
            "$1,500-$1,749", "$1,750-$1,999", "$2,000-$2,999", "$3,000-$3,499", "$3,500 or more")

column_mappings <- c("Neg_Nil_income", "1_149", "150_299", "300_399", "400_499",
                     "500_649", "650_799", "800_999", "1000_1249", "1250_1499",
                     "1500_1749", "1750_1999", "2000_2999", "3000_3499", "3500_more")

# Initializing a data frame to store the results
result_df <- tibble(Range = ranges, Population = numeric(length(ranges)))

# Subtracting and aggregating population values
for (i in seq_along(ranges)) {
  age_col <- paste0("P_", column_mappings[i], "_15_19_yrs")
  total_col <- paste0("P_", column_mappings[i], "_Tot")
  # Taking mean here just for 'na' values resolution
  result_df$Population[i] <- mean(kooyong_data_combined[[total_col]] - kooyong_data_combined[[age_col]], na.rm = TRUE)
}

result_df$Range <- factor(result_df$Range, levels = ranges)

# Plotting the bar chart
ggplot(result_df, aes(x = Range, y = Population, fill = Range)) +
  geom_col(show.legend = FALSE) +
  labs(title = "Population by Income Range for Kooyong",
       x = "Income Range",
       y = "Population (in thousands)") +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



For Victoria

```
# Filtering data relevant to all Victoria
victoria_data_combined <- bind_rows(
  g17_part1,
  g17_part2
)

# Constructing a mapping based on the income ranges
ranges <- c("Negative/Nil income", "$1-$149", "$150-$299", "$300-$399", "$400-$499",
            "$500-$649", "$650-$799", "$800-$999", "$1,000-$1,249", "$1,250-$1,499",
            "$1,500-$1,749", "$1,750-$1,999", "$2,000-$2,999", "$3,000-$3,499", "$3,500 or more")

column_mappings <- c("Neg_Nil_income", "1_149", "150_299", "300_399", "400_499",
                     "500_649", "650_799", "800_999", "1000_1249", "1250_1499",
                     "1500_1749", "1750_1999", "2000_2999", "3000_3499", "3500_more")

# Initializing a data frame to store the results
result_df <- tibble(Range = ranges, Population = numeric(length(ranges)))

# Subtracting and aggregating population values
for (i in seq_along(ranges)) {
  age_col <- paste0("P_", column_mappings[i], "_15_19_yrs")
  total_col <- paste0("P_", column_mappings[i], "_Tot")
  result_df$Population[i] <- mean(victoria_data_combined[[total_col]] - victoria_data_combined[[age_col]], na.rm = TRUE)
}

result_df$Range <- factor(result_df$Range, levels = ranges)

# Plotting the bar chart
ggplot(result_df, aes(x = Range, y = Population, fill = Range)) +
  geom_col(show.legend = FALSE) +
  labs(title = "Population by Income Range for Victoria",
       x = "Income Range",
       y = "Population (in thousands)") +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Analysis: These distribution charts are bit different and show that the average income in Kooyong is much higher as compared to generally in the state.

G02 data based analysis of Kooyong Electorate

Below are the 2 variables that I would select to do the analysis of the G02 Census data and which seemed most relevant in an electoral context:

Median Rent (Median_rent_weekly) Description- This represents the median rent paid by households on a weekly basis in a given area. Source- From the census data, which usually contains detailed socioeconomic information about regions. Interest in Election Politics- Housing affordability is often a critical issue in elections. Regions with higher rents might have populations more concerned with policies related to housing, rent control, and real estate market regulation. An analysis of this variable can indicate how housing costs could influence the political leanings of a region or how receptive an electorate might be to certain housing policies.

Average Household Size (Average_household_size) Description- This measures the average number of people living in a household within an area. Source- Like the other, same from the census data, which provides demographic and household composition statistics. Interest in Election Politics- Average household size can reflect various demographic and cultural characteristics of an electorate. For instance, areas with larger households may have different priorities, such as family services, education, and community facilities. Understanding the household size can provide insights into the potential concerns and policy preferences of an electorate, such as those related to childcare, schooling, and tax benefits for families.

Data acquisition steps:

For this analysis, we've used the most recent census data from Australian Bureau of Statistics which was conducted in 2021. This data includes specific medians and averages under the general community.

Visited the Census GeoPackages page of ABS and entered the following specifications to get the required data- Census year - 2021 Selected state or territory - Victoria Selected the table data for G02 - Selected medians and averages GDA type - GDA2020 Downloaded the zip archive which came up from the following link- https://www.abs.gov.au/census/find-census-data/geopackages/download/Geopackage_2021_G02_VIC_GDA2020.zip (https://www.abs.gov.au/census/find-census-data/geopackages/download/Geopackage_2021_G02_VIC_GDA2020.zip)

Here the selected variables from the G02 census are analysed based on the basis of their SA1 codes to analyze their distribution for the selected electorate region and the state.

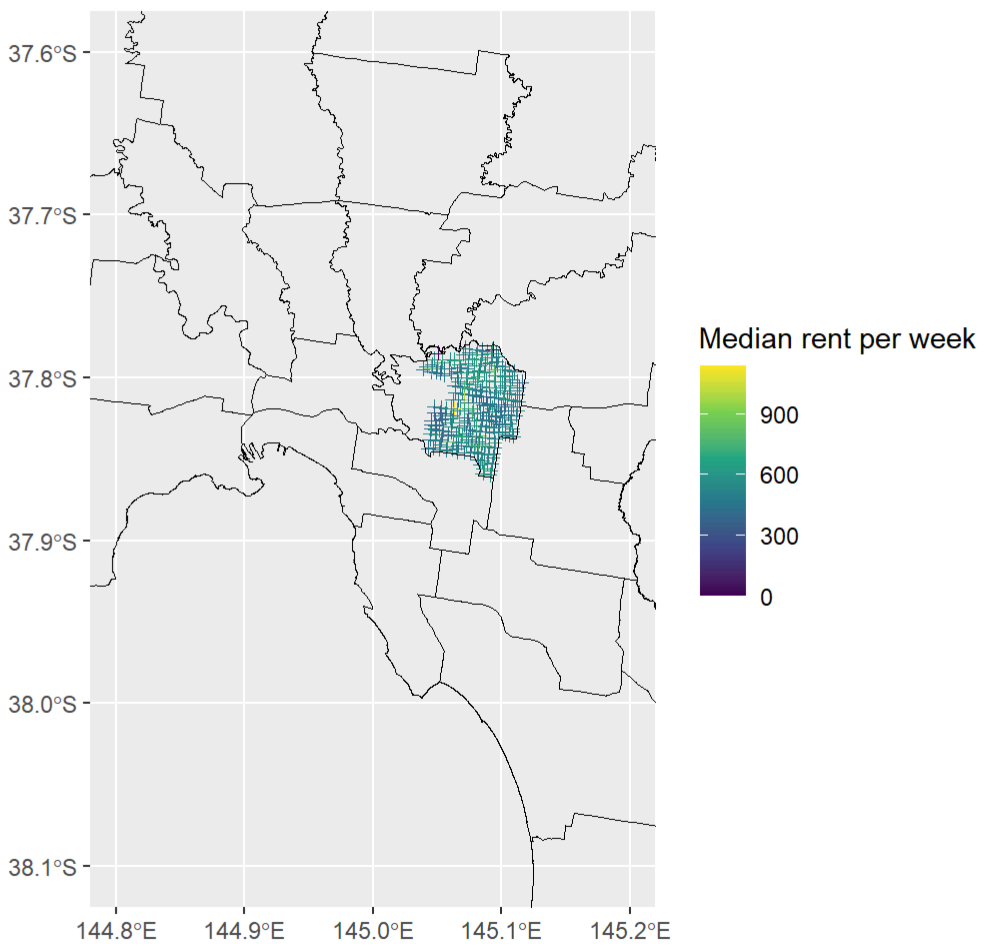
To acquire the SA1 codes for the electorate and the boundary data for Victoria we need to visit the "Maps and data of Victorian electoral divisions" page of Australian Electoral Commission (AEC) wherein from the electoral divisions by SA2 and SA1 section we can acquire the SA1 codes by converting the 11 digit code into 7 digits for our chosen electorate by accessing the Victoria electoral divisions SA1 and SA2 data set. Furthermore, by accessing the GIS data section of the page we can acquire the boundary data for Victoria named 'ESRI shapefile'. This will help us plot our data on maps for better visualization and understanding.

```
# reading the SA1 Geopackage data
G02_SA1_data <- read_sf(here::here("data/Geopackage_2021_G02_VIC_GDA2020/G02_VIC_GDA2020.gpkg"), layer = "G02_SA1_2021_VIC")

# Filtering out relevant columns to selected variables
G02_SA1_filtered <- G02_SA1_data %>%
  select(SA1_CODE_2021, Median_rent_weekly, Average_household_size, SA1_NAME_2021)

# manipulate SA1 codes for extraction
G02_SA1_filtered <- G02_SA1_filtered %>%
  mutate(SA1_CODE = paste0(substr(SA1_CODE_2021, 1, 1),
                           substr(SA1_CODE_2021, 6, 9),
                           substr(SA1_CODE_2021, 10, 11)))

#Converting SA1 code to character data
G02_SA1_filtered$SA1_CODE <- as.character(G02_SA1_filtered$SA1_CODE)

# calculate centroids for each SA1 region
G02_SA1_filtered <- G02_SA1_filtered |>
  mutate(centroid = st_centroid(geom))

# reading SA1 codes for Kooyong from the Victoria electoral divisions SA1 and SA2 data set
Kooyong_SA1_codes <- read_excel("data/vic-by-SA2-and-SA1-by-division.xlsx", sheet = "KOOYONG")

# renaming columns for ease of understanding
colnames(Kooyong_SA1_codes) <- c("SA1_CODE", "New_Division", "Old_Division", "SA2_Name", "Actual_enrolment", "Projected_enro
lment")



# joining SA1 GeoPackage data with Kooyong SA1 codes
G02_SA1_filtered_Kooyong <- G02_SA1_filtered %>%
  inner_join(Kooyong_SA1_codes, by = c("SA1_CODE" = "SA1_CODE"), copy = TRUE)

# further filter out zero values for visualization
G02_SA1_filtered <- G02_SA1_filtered %>%
   mutate(centroid = st_centroid(geom)) %>%
   filter(Median_rent_weekly != 0) %>%
  filter(Average_household_size != 0)

vic_elec_map <- read_sf(here::here("data/E_VIC21_region.shp")) %>%
   mutate(DivisionNm = toupper(Elect_div)) %>%
  mutate(geometry = st_zm(geometry))

# set projection to GDA1994 using EPSG:4283
st_crs(vic_elec_map$geometry,4283)
```

```
## Coordinate Reference System:
##   User input: GDA94
##   wkt:
## GEOGCRS["GDA94",
##     DATUM["Geocentric Datum of Australia 1994",
##         ELLIPSOID["GRS 1980",6378137,298.257222101,
##             LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["geodetic latitude (Lat)",north,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["geodetic longitude (Lon)",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     USAGE[
##         SCOPE["Horizontal component of 3D system."],
##         AREA["Australia including Lord Howe Island, Macquarie Island, Ashmore and Cartier Islands, Christmas Island, Coco
s (Keeling) Islands, Norfolk Island. All onshore and offshore."],
##         BBOX[-60.55,93.41,-8.47,173.34]],
##     ID["EPSG",4283]]
```

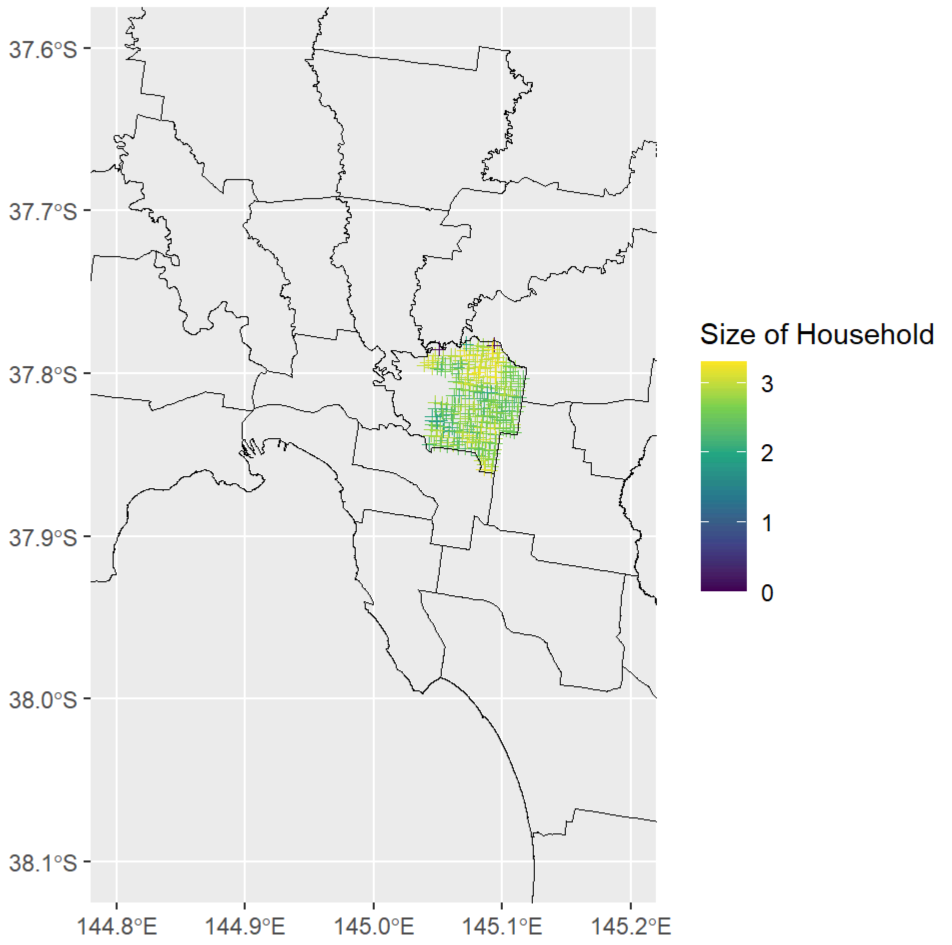
Median Weekly Rent mapped for Kooyong

```
# transform projection from GDA1994 to GDA2020 using EPSG:7844
vic_elec_map$geometry = st_transform(vic_elec_map$geometry, 7844)

# plotting for household size and median weekly rent of Kooyong electorate

ggplot(G02_SA1_filtered_Kooyong) +
   geom_sf(aes(geometry = centroid, color = Median_rent_weekly), shape = 3) +
   geom_sf(data = vic_elec_map, aes(geometry = geometry), fill = "transparent", size = 1.2, color = "black") +
   coord_sf(xlim = c(144.8, 145.2), ylim = c(-38.1, -37.6)) +
   scale_color_viridis_c(name = "Median rent per week", option = "viridis")
```



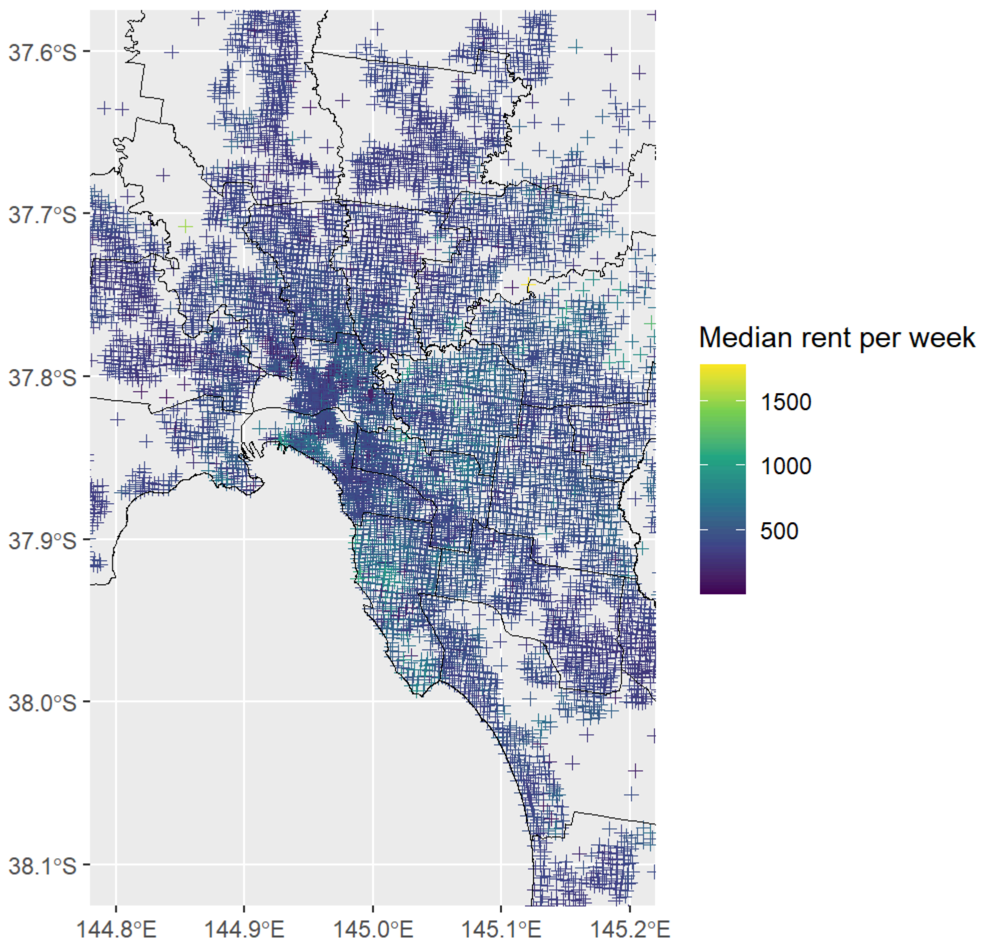
Average household size mapped for Kooyong

```
ggplot(G02_SA1_filtered_Kooyong) +
   geom_sf(aes(geometry = centroid, color = Average_household_size), shape = 3) +
   geom_sf(data = vic_elec_map, aes(geometry = geometry), fill = "transparent", size = 1.2, color = "black") +
   coord_sf(xlim = c(144.8, 145.2), ylim = c(-38.1, -37.6)) +
   scale_color_viridis_c(name = "Size of Household", option = "viridis")
```



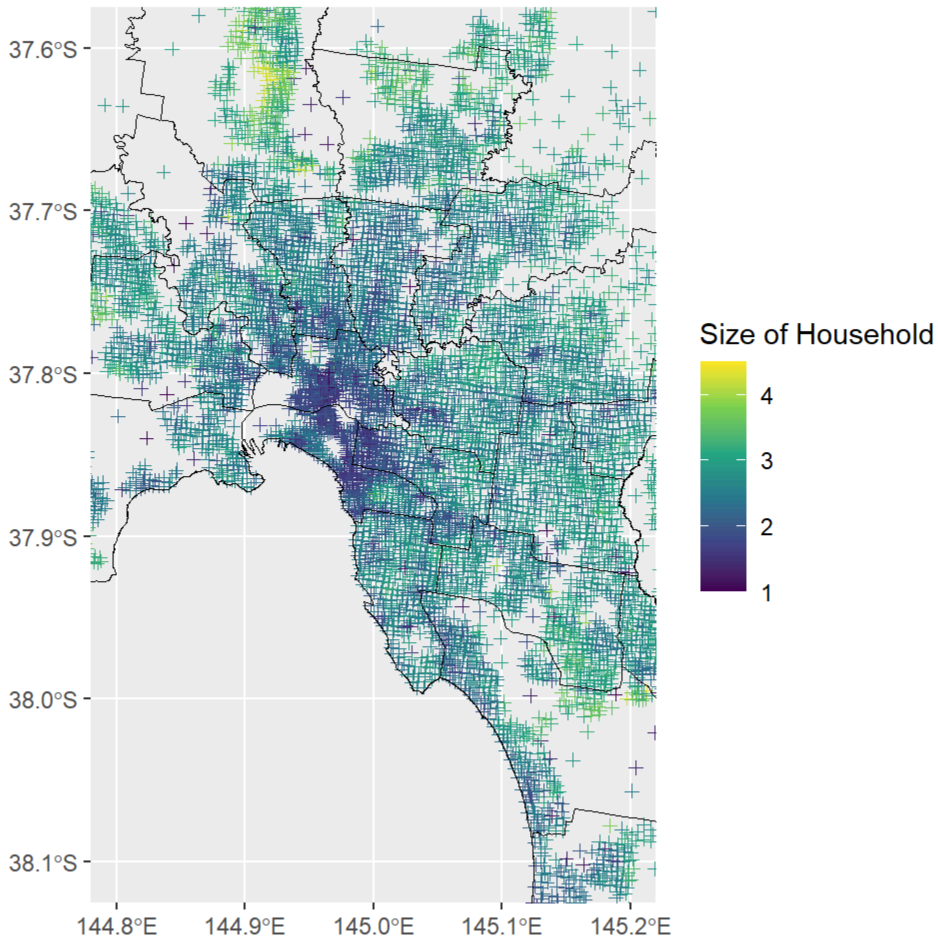
Median Weekly Rent mapped for Victoria

```
ggplot(G02_SA1_filtered) +
   geom_sf(aes(geometry = centroid, color = Median_rent_weekly), shape = 3) +
   geom_sf(data = vic_elec_map, aes(geometry = geometry), fill = "transparent", size = 1.2, color = "black") +
   coord_sf(xlim = c(144.8, 145.2), ylim = c(-38.1, -37.6)) +
   scale_color_viridis_c(name = "Median rent per week", option = "viridis")
```



Average household size mapped for Victoria

```
ggplot(G02_SA1_filtered) +
   geom_sf(aes(geometry = centroid, color = Average_household_size), shape = 3) +
   geom_sf(data = vic_elec_map, aes(geometry = geometry), fill = "transparent", size = 1.2, color = "black") +
   coord_sf(xlim = c(144.8, 145.2), ylim = c(-38.1, -37.6)) +
   scale_color_viridis_c(name = "Size of Household", option = "viridis")
```

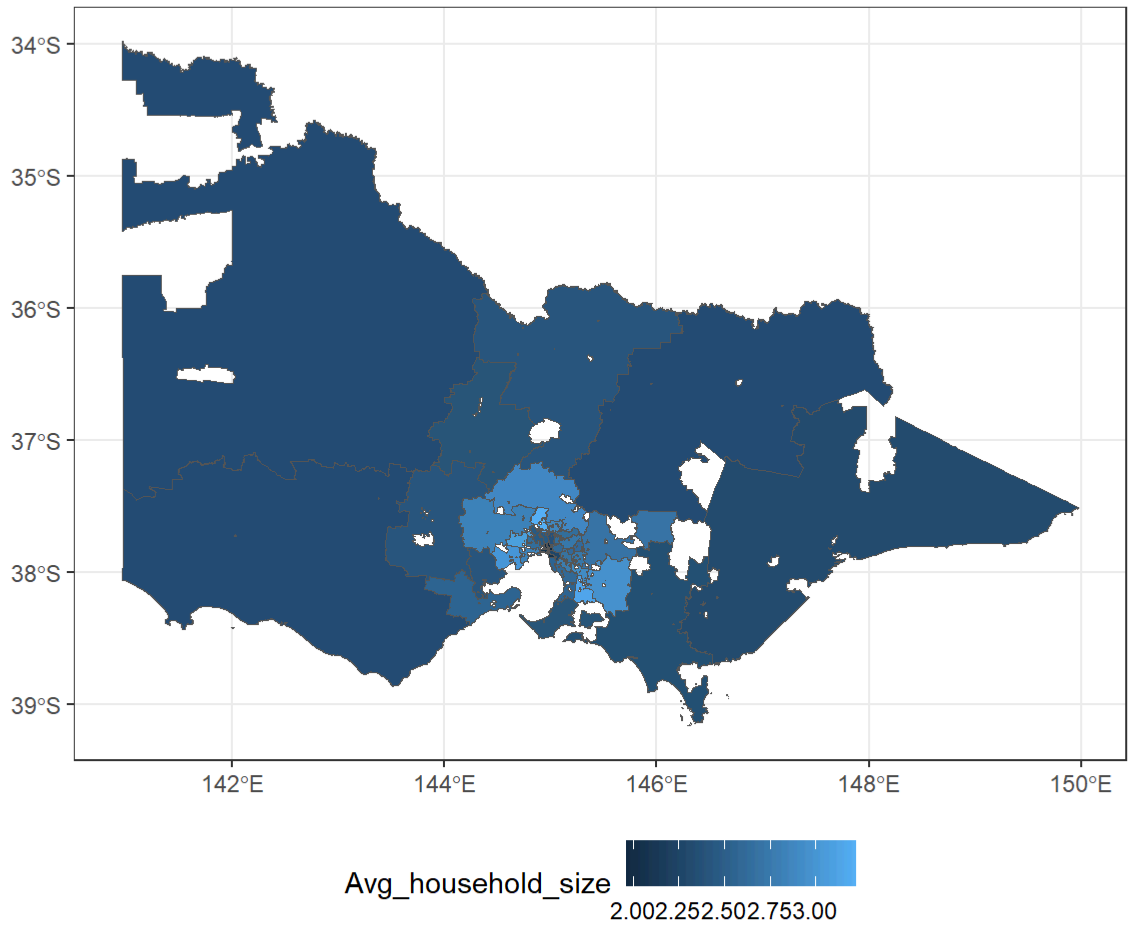


Compairing and analysing Kooyong with other electorates

Checking for the entire state

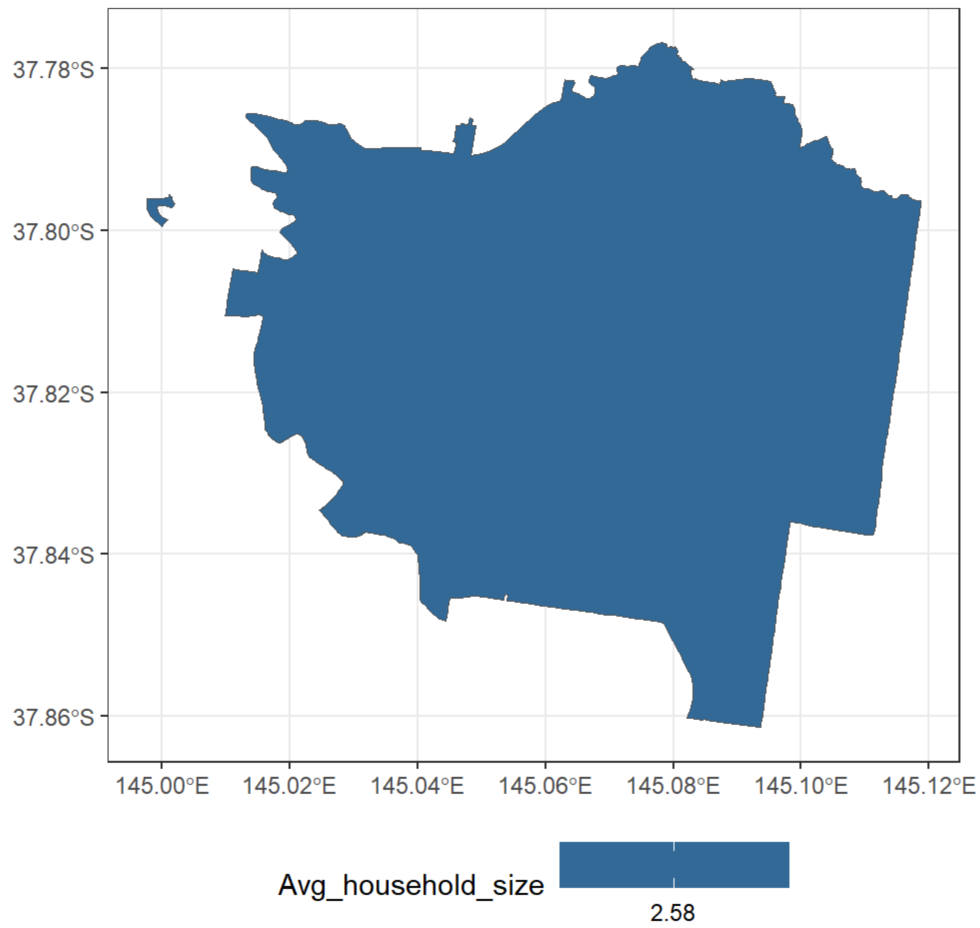
When analyzing the above mappings the median weekly rent suggests that the average rent in the electorate is quite high as compared to the rest of the state and this the average household size might be big to afford this. But in contrast the average household size for the electorate is pretty much average as compared to other places. This suggests that despite the household size being average the people living in this electorate are mostly paying higher rents and thus might be earning higher as compared to the other electorates.

```
# determining with SA1 centroids overlap with which electoral regions
electoral_intersects = st_intersects(G02_SA1_filtered$centroid,
                                vic_elec_map$geometry,
                                sparse = FALSE)
arr_ind = which(electoral_intersects == TRUE, arr.ind = TRUE)
sa1_ind = arr_ind[,1]
division_ind = arr_ind[,2]
division_name = vic_elec_map$Elect_div[division_ind]
sa1_name = G02_SA1_filtered$SA1_NAME_2021[sa1_ind]

# save the SA1 region and its matching electorate into a data frame
sa1_divisions = data.frame(SA1_NAME_2021 = sa1_name,
                            DivisionNm = division_name)

# join SA1 data and divisions intersected
vicmap_sa1_G02_electorates  <- G02_SA1_filtered %>%
  right_join(sa1_divisions)

# Combine the data from electorates together using group_by
vicmap_sa1_G02_electorates_grouped =  vicmap_sa1_G02_electorates |>
  group_by(DivisionNm) |>
  summarise(Avg_household_size = mean(Average_household_size)) |>
  ungroup()

# Plot results to check assignment
ggplot() +
   geom_sf(data = vicmap_sa1_G02_electorates_grouped,
           aes(geometry = geom, fill = Avg_household_size)) +
  theme_bw() +
  theme(legend.position = "bottom")
```



Let's checkout just for Kooyong

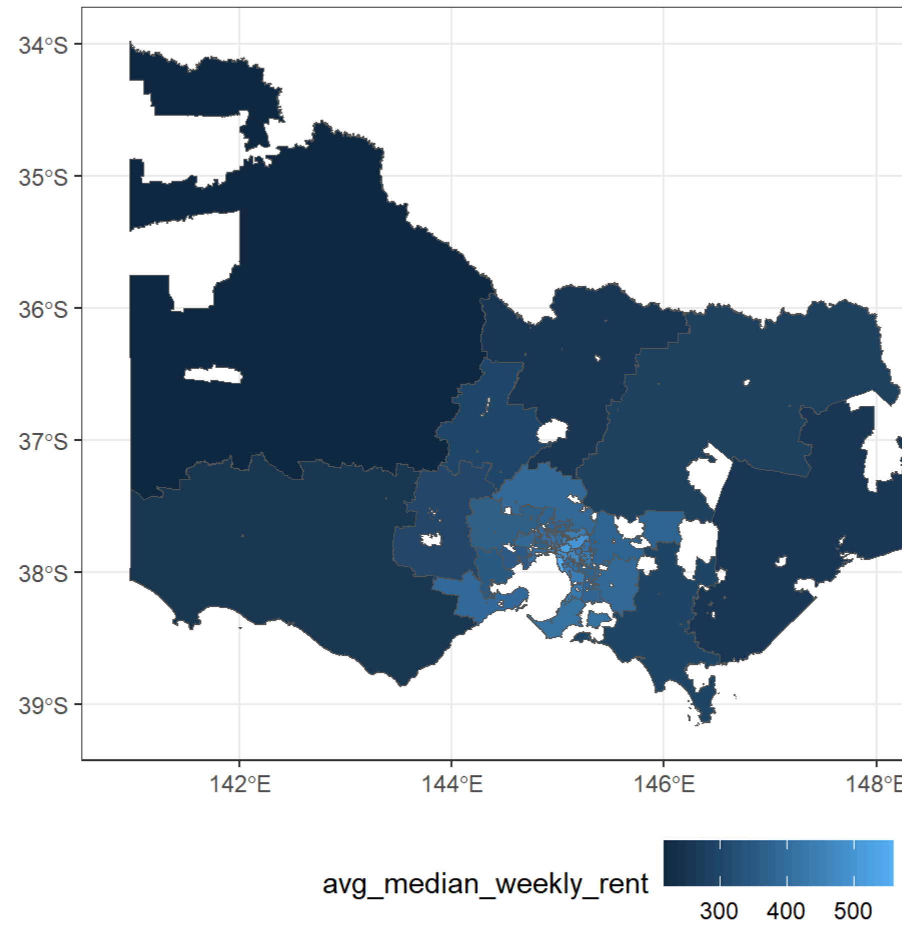
```
# Filtering out Kooyong data
Kooyong_household <- vicmap_sa1_G02_electorates %>%
  filter(DivisionNm == "Kooyong") %>%
  summarise(Avg_household_size = mean(Average_household_size), na.rm = TRUE) %>%
  ungroup()

# Plotting
ggplot() +
   geom_sf(data = Kooyong_household,
           aes(geometry = geom, fill = Avg_household_size)) +
  theme_bw() +
  theme(legend.position = "bottom")
```



For the median weekly rent in Kooyong

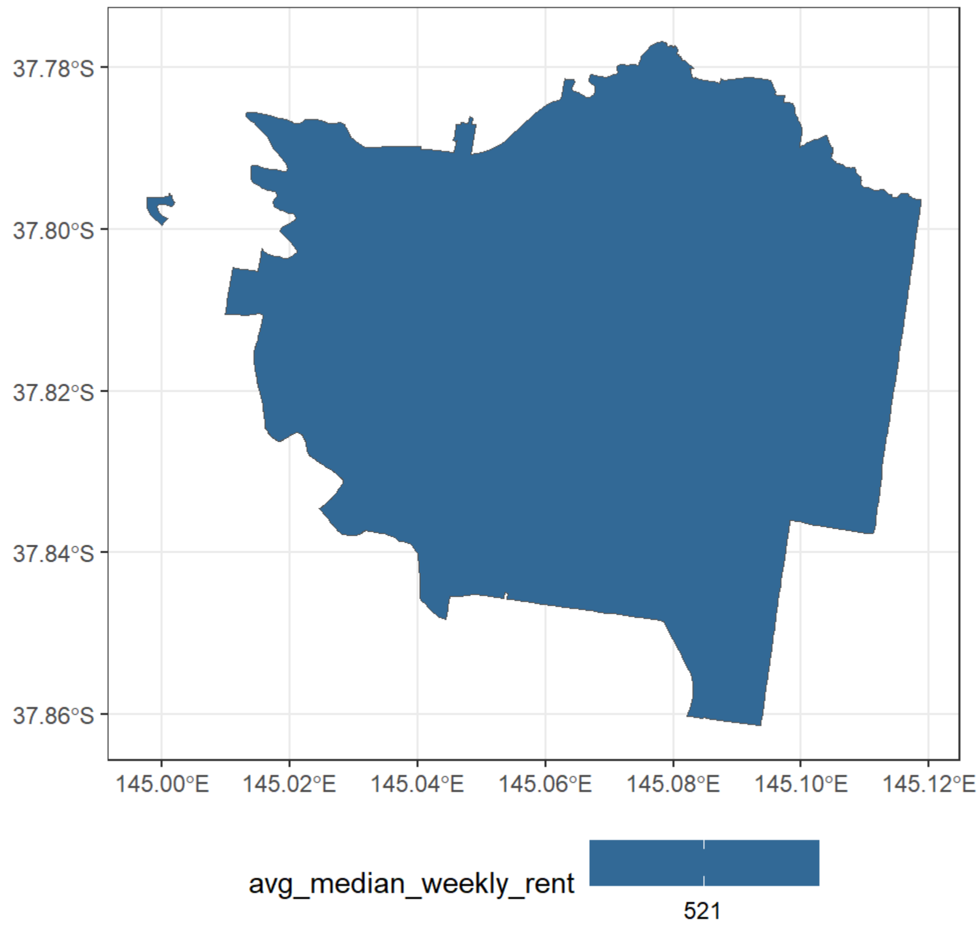
For whole state

```
# Combine the data from electorates together using group_by
vicmap_sa1_G02_electorates_grouped =  vicmap_sa1_G02_electorates |>
  group_by(DivisionNm) |>
  summarise(avg_median_weekly_rent = mean(Median_rent_weekly)) |>
  ungroup()

# Plot results to check assignment
ggplot() +
   geom_sf(data = vicmap_sa1_G02_electorates_grouped,
           aes(geometry = geom, fill = avg_median_weekly_rent)) +
  theme_bw() +
  theme(legend.position = "bottom")
```



let's checkout for Kooyong

```
# Filtering Koooyong relevant data
kooyong_rent <- vicmap_sa1_G02_electorates %>%
  filter(DivisionNm == "Kooyong") %>%
  summarise(avg_median_weekly_rent = mean(Median_rent_weekly, na.rm = TRUE)) %>%
  ungroup()

ggplot() +
   geom_sf(data = kooyong_rent,
           aes(geometry = geom, fill = avg_median_weekly_rent)) +
  theme_bw() +
  theme(legend.position = "bottom")
```



Conclusive Remarks

From the analysis of g02 census data for the Kooyong electorate it can be concluded that while the region has a normal household size as compared to the rest of the state, its higher weekly rent suggests that the population in this region earns higher than the latter. These observations can prove useful when formulating the election campaign for the said region, where more focus should be laid on other factors as compared to the improvement of its population's income. This also suggests that there are some things that are being done right in the region which should also be implemented in the other parts of the state.

Effect of The 2023 Parliament Referendum in Kooyong

As the 2023 referendum was a significant event for the aboriginal population of Australia, analysis of its acceptance can prove to be a vital source of information for understanding the political landscape of an area. This might prove useful for my chosen electorate of Kooyong, so I visited the tally room webpage of the Australian electoral commission and acquired the relevant data for understanding the amount of yes votes in the region.

Below are the steps of how I proceeded: 1. Visited results.ac.gov.au and under federal referendums in the Tally room archive section clicked on the 2023 referendum. 2. On the page which opened I clicked on the downloads button in the left pane and so a list of different data sets came up. 3. Then under the Votes section, I clicked on "Votes by polling place - VIC" to get the relevant data for Victoria as Kooyong lies here.

The link for the downloaded CSV is attached below:
https://results.aec.gov.au/29581/Website/Downloads/ReferendumPollingPlaceResultsByStateDownload-29581-VIC.csv
(https://results.aec.gov.au/29581/Website/Downloads/ReferendumPollingPlaceResultsByStateDownload-29581-VIC.csv)

Furthermore here is the R code for the analysis of the yes votes in Kooyong:

```
# Reading the polls result
yes_votes <- read_csv(here::here("data/ReferendumPollingPlaceResultsByStateDownload-29581-VIC.csv"),skip = 1)

# Filtering data out for Kooyong and summarizing
kooyong_yes_votes <- yes_votes%>%
  filter(DivisionName=="Kooyong")%>%
  summarise(
    yes_votes_total = sum(YesVotes),
    votes_total = sum(TotalVotes),
    kooyong_yes_perc = yes_votes_total/votes_total * 100
  )

# Printing the percentage for the electorate
print(paste("Following is the percentage of Yes votes for Kooyong:",kooyong_yes_votes$kooyong_yes_perc,"%"))
```

```
## [1] "Following is the percentage of Yes votes for Kooyong: 60.2314182201106 %"
```

As it can be seen from the output of the analysis, around 60.23 percent of the population of Kooyong voted for yes to an indigenous voice in the parliament, which is a majority.

For getting the relevant census variables necessary for identifying aboriginal and Torres Strait people of the region, I visited the ABS website and acquired the relevant census data pack. The specifications are as below: Census year: 2021 DataPack type: Aboriginal and/or Torres Strait Islander Peoples (Indigenous) Profile Geography: Statistical Area 2 (SA2)

From here I downloaded the data for Victoria as per the requirements of my case.

On opening the downloaded archive and going through the files in it, I came across the following data set which contains appropriate variables for identifying the percentage of Aboriginal and Torres Strait islanders in Kooyong. The data set is as follows:

```r
# Reading and viewing the population data for the SA2 region
sa2_pop_data <- read_csv(here::here("data/2021_Census_IP_Statistical_Area_2_for_VIC/2021Census_I02_VIC_SA2.csv"))
head(sa2_pop_data)
```

```
## # A tibble: 6 × 22
##   SA2_CODE_2021 Indigenous_Aboriginal_M Indigenous_Aboriginal_F
##           <dbl>                   <dbl>                   <dbl>
## 1     201011001                      84                     109
## 2     201011002                      64                      44
## 3     201011005                      39                      44
## 4     201011006                     110                     117
## 5     201011007                      42                      38
## 6     201011008                     193                     205
## # ℹ 19 more variables: Indigenous_Aboriginal_P <dbl>,
## #   Indigenous_Torres_Strait_Is_M <dbl>, Indigenous_Torres_Strait_Is_F <dbl>,
## #   Indigenous_Torres_Strait_Is_P <dbl>, Indig_Bth_Abor_Torr_Strt_Is_M <dbl>,
## #   Indig_Bth_Abor_Torr_Strt_Is_F <dbl>, Indig_Bth_Abor_Torr_Strt_Is_P <dbl>,
## #   Indigenous_TotM <dbl>, Indigenous_TotF <dbl>, Indigenous_TotP <dbl>,
## #   Non_Indig_M <dbl>, Non_Indig_F <dbl>, Non_Indig_P <dbl>,
## #   Indigenous_status_nsM <dbl>, Indigenous_status_nsF <dbl>, …
```

From this data set the following columns should prove appropriate for calculating the percentage of aboriginal people. The description for them is as follows: Indigenous_Aboriginal_P <- Aboriginal people count for that SA2 region. Indigenous_Torres_Strait_Is_P <- Torres Strait islanders count for that SA2 region. Indigenous_TotP <- Total indigenous population count for that region. Tot_P <- Total overall population count for that region.

Download link for the SA2 data used: https://www.abs.gov.au/census/find-census-data/datapacks/download/2021_IP_SA2_for_VIC_short-header.zip (https://www.abs.gov.au/census/find-census-data/datapacks/download/2021_IP_SA2_for_VIC_short-header.zip)

Using the SA_2 codes for Kooyong and these variables we can get an idea of how many people identify as aboriginal and Torres strait islanders and what is there total population, further helping us calculate their percentage share in the population of the region.

Coming to information on the Indigenous population for the SA1 regions, it is currently unavailable in the datapacks. This is likely due to privacy and confidentiality concerns, especially in areas with small populations where disclosing sensitive demographic details like Indigenous status could be risky. Furthermore, data collecting reliability may be poorer in rural and regional locations with fewer populations, reducing the accuracy of demographic data. Even though SA1 regions are the foundation of statistical reporting, information is often aggregated at higher geographical levels to guarantee reliability and safeguard privacy. As a result, there is a lack of detailed demographic data for Indigenous populations at the SA1 level.

Broader population represetativness analysis check of Kooyong

It should be noted that Kooyong in some area represented the trends of the state in terms of gender but otherwise for income and age distribution it was much different. This could be one of the reasons for the yes vote to voice in parliament for the Indigenous people. Verified from: https://www.theguardian.com/australia-news/ng-interactive/2023/oct/17/live-voice-referendum-results-aec-2023-poll-tracker-by-electorate-state-australia-yes-no-vote-polls-percentage-the-australian-indigenous-voice-to-parliament-who-won-result-winner-map-counts (https://www.theguardian.com/australia-news/ng-interactive/2023/oct/17/live-voice-referendum-results-aec-2023-poll-tracker-by-electorate-state-australia-yes-no-vote-polls-percentage-the-australian-indigenous-voice-to-parliament-who-won-result-winner-map-counts)

References

-> Websites utilized as data source:

- Australian Bureau of Statistics - https://www.abs.gov.au/ (https://www.abs.gov.au/)
- Australian Electoral Commision - https://www.aec.gov.au/ (https://www.aec.gov.au/)
- The Guardian (major news network) - https://www.theguardian.com/ (https://www.theguardian.com/)

-> Individual data sources added in their respective blocks

-> Citing packages used:

```r
citation("tidyverse")
```

```
## To cite package 'tidyverse' in publications use:
##
##   Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R,
##   Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller
##   E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V,
##   Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to
##   the tidyverse." _Journal of Open Source Software_, *4*(43), 1686.
##   doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Welcome to the {tidyverse}},
##     author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang and Lucy D'Agostino McGowan and Romain François and Garrett Grolemund and Alex Hayes and Lionel Henry and Jim Hester and Max Kuhn and Thomas Lin Pedersen and Evan Miller and Stephan Milton Bache and Kirill Müller and Jeroen Ooms and David Robinson and Dana Paige Seidel and Vitalie Spinu and Kohske Takahashi and Davis Vaughan and Claus Wilke and Kara Woo and Hiroaki Yutani},
##     year = {2019},
##     journal = {Journal of Open Source Software},
##     volume = {4},
##     number = {43},
##     pages = {1686},
##     doi = {10.21105/joss.01686},
##   }
```

```
citation("sf")
```

```
## To cite package sf in publications, please use:
##
##   Pebesma, E., & Bivand, R. (2023). Spatial Data Science: With
##   Applications in R. Chapman and Hall/CRC.
##   https://doi.org/10.1201/9780429459016
##
##   Pebesma, E., 2018. Simple Features for R: Standardized Support for
##   Spatial Vector Data. The R Journal 10 (1), 439-446,
##   https://doi.org/10.32614/RJ-2018-009
##
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```

```
citation("ggplot2")
```

```
## To cite ggplot2 in publications, please use
##
##   H. Wickham. ggplot2: Elegant Graphics for Data Analysis.
##   Springer-Verlag New York, 2016.
##
## A BibTeX entry for LaTeX users is
##
##   @Book{,
##     author = {Hadley Wickham},
##     title = {ggplot2: Elegant Graphics for Data Analysis},
##     publisher = {Springer-Verlag New York},
##     year = {2016},
##     isbn = {978-3-319-24277-4},
##     url = {https://ggplot2.tidyverse.org},
##   }
```

```
citation("readxl")
```

```
## To cite package 'readxl' in publications use:
##
##   Wickham H, Bryan J (2023). _readxl: Read Excel Files_. R package
##   version 1.4.3, <https://CRAN.R-project.org/package=readxl>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {readxl: Read Excel Files},
##     author = {Hadley Wickham and Jennifer Bryan},
##     year = {2023},
##     note = {R package version 1.4.3},
##     url = {https://CRAN.R-project.org/package=readxl},
##   }
```