# MATH2270/MATH2237/MATH2404 Assignment 3

## Name

Jiacong Zheng

## Student number

S3913565

## Assessment declaration checklist

Please carefully read the statements below and check each box if you agree with the declaration. If you do not check all boxes, your assignment will not be marked. If you make a false declaration on any of these points, you may be investigated for academic misconduct. Students found to have breached academic integrity may receive official warnings and/or serious academic penalties. Please read more about academic integrity [here](https://www.rmit.edu.au/students/my-course/assessment-results/academic-integrity). If you are unsure about any of these points or feel your assessment might breach academic integrity, please contact your course coordinator for support. It is important that you DO NOT submit any assessment until you can complete the declaration truthfully.

By checking the boxes below, I declare the following:

|  |  |
| --- | --- |
|  | I have not impersonated, or allowed myself to be impersonated by, any person for the purposes of this assessment |
|  | This assessment is my original work and no part of it has been copied from any other source except where due acknowledgement is made. |
|  | No part of this assessment has been written for me by any other person except where such collaboration has been authorised by the lecturer/teacher concerned. |
|  | I have not used generative “AI” tools for the purposes of this assessment. |
|  | Where this work is being submitted for individual assessment, I declare that it is my original work and that no part has been contributed by, produced by or in conjunction with another student. |
|  | I give permission for my assessment response to be reproduced, communicated, compared and archived for the purposes of detecting plagiarism. |
|  | I give permission for a copy of my assessment to be retained by the university for review and comparison, including review by external examiners. |

I understand that:

|  |  |
| --- | --- |
|  | Plagiarism is the presentation of the work, idea or creation of another person or machine as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited. |
|  | Plagiarism includes the act of assisting or allowing another person to plagiarise or to copy my work. |

I agree and acknowledge that:

|  |  |
| --- | --- |
|  | I have read and understood the Declaration and Statement of Authorship above. |
|  | If I do not agree to the Declaration and Statement of Authorship in this context and all boxes are not checked, the assessment outcome is not valid for assessment purposes and will not be included in my final result for this course. |

## Assignment URL

<https://radar.cloudflare.com/scan/b769e9ea-8fc3-42f0-a1eb-edc47886cdfa>

## References

Australian Electoral Commission. (2022). 2022 federal election - House of Representatives downloads. <https://results.aec.gov.au/27966/Website/HouseDownloadsMenu-27966-Csv.htm>

Australian Electoral Commission. (2025). 2025 federal election - House of Representatives downloads. <https://tallyroom.aec.gov.au/HouseDownloadsMenu-31496-Csv.htm>

Australian Electoral Commission. (n.d.). Federal electoral boundary GIS data. <https://www.aec.gov.au/electorates/gis/gis_datadownload.htm>

## Assignment Code

library(shiny)

library(dplyr)

library(ggplot2)

library(plotly)

library(sf)

library(leaflet)

library(DT)

# 2022 Election Data

seats\_2022 <- read.csv("HouseCandidatesDownload-27966.csv",skip = 1, header = TRUE)

first\_prefs\_party\_2022 <- read.csv("HouseFirstPrefsByPartyDownload-27966.csv",skip = 1, header = TRUE)

tpp\_div\_2022 <- read.csv("HouseTppByDivisionDownload-27966.csv",skip = 1, header = TRUE)

changed\_seats\_2022 <- read.csv("HouseSeatsWhichChangedHandsDownload-27966.csv",skip = 1, header = TRUE)

# 2025 Election Data

seats\_2025 <- read.csv("HouseCandidatesDownload-31496.csv",skip = 1, header = TRUE)

first\_prefs\_party\_2025 <- read.csv("HouseFirstPrefsByPartyDownload-31496.csv",skip = 1, header = TRUE)

tpp\_div\_2025 <- read.csv("HouseTppByDivisionDownload-31496.csv",skip = 1, header = TRUE)

changed\_seats\_2025 <- read.csv("HouseSeatsWhichChangedHandsDownload-31496.csv",skip = 1, header = TRUE)

# Load shapefiles

divisions\_2022 <- st\_read("CED\_2016\_AUST.shp")

divisions\_2025 <- st\_read("CED\_2016\_AUST.shp")

seats\_2022\_winners <- seats\_2022 %>%

filter(Elected == "Y") %>%

group\_by(PartyNm) %>%

summarise(Seats = n(), .groups = "drop")

seats\_2025\_winners <- seats\_2025 %>%

filter(Elected == "Y") %>%

group\_by(PartyNm) %>%

summarise(Seats = n(), .groups = "drop")

tpp\_div\_2022\_agg <- tpp\_div\_2022 %>%

mutate(TPP\_Percentage = ifelse(PartyAb == "ALP", `Australian.Labor.Party.Votes` / `TotalVotes` \* 100,

`Liberal.National.Coalition.Votes` / `TotalVotes` \* 100)) %>%

group\_by(StateAb, PartyAb, Year = "2022") %>%

summarise(TPP\_Percentage = mean(TPP\_Percentage, na.rm = TRUE), .groups = "drop")

tpp\_div\_2025\_agg <- tpp\_div\_2025 %>%

mutate(TPP\_Percentage = ifelse(PartyAb == "ALP", `Australian.Labor.Party.Votes` / `TotalVotes` \* 100,

`Liberal.National.Coalition.Votes` / `TotalVotes` \* 100)) %>%

group\_by(StateAb, PartyAb, Year = "2025") %>%

summarise(TPP\_Percentage = mean(TPP\_Percentage, na.rm = TRUE), .groups = "drop")

# UI

ui <- fluidPage(

titlePanel("Australian Federal Elections 2022-2025: Political Changes"),

tabsetPanel(

# Tab 1: Party Performance and Seat Changes

tabPanel("Party Performance",

fluidRow(

column(4, style = "height: 100vh; overflow-y: auto;",

selectInput("party\_seats", "Select Parties:",

choices = unique(c(seats\_2022$PartyNm, seats\_2025$PartyNm)),

multiple = TRUE, selected = c("Australian Labor Party", "Liberal")),

selectInput("year\_map", "Select Year:", choices = c("2022", "2025")),

textOutput("party\_performance\_summary")

),

column(8, style = "height: 100vh; overflow-y: auto;",

plotlyOutput("seats\_bar", height = "400px"),

leafletOutput("seats\_map", height = "calc(100vh - 400px)")

))),

# Tab 2: Voting Patterns and Preferences

tabPanel("Voting Patterns",

fluidRow(

column(4, style = "height: 100vh; overflow-y: auto;",

selectInput("party\_votes", "Select Parties:",

choices = unique(c(first\_prefs\_party\_2022$PartyNm, first\_prefs\_party\_2025$PartyNm)),

multiple = TRUE, selected = c("Australian Labor Party", "Liberal")),

selectInput("year\_pie", "Select Year:", choices = c("2022", "2025")),

textOutput("voting\_patterns\_summary")

),

column(8,style = "height: 100vh; overflow-y: auto;",

plotlyOutput("votes\_bar", height = "400px"),

plotlyOutput("votes\_pie", height = "calc(100vh - 400px)")

))),

# Tab 3: Regional Analysis

tabPanel("Seats That Changed Hands",

fluidRow(

column(4, style = "height: 100vh; overflow-y: auto;",

selectInput("state\_changed", "Select State:",

choices = c("All", unique(changed\_seats\_2025$StateAb),selected ="VIC")),

textOutput("seat\_changes\_summary")),

column(8,style = "height: 100vh; overflow-y: auto;",

leafletOutput("changed\_seats\_map", height = "500px"),

plotlyOutput("changed\_seats\_chart", height = "400px"),

DT::dataTableOutput("changed\_seats\_table", height = "calc(100vh - 900px)")

))),

# Tab 4: References

tabPanel("References",

HTML("

<h3>Data Sources</h3>

<p>Australian Electoral Commission. (2022). 2022 federal election - House of Representatives downloads. <a href='https://results.aec.gov.au/27966/Website/HouseDownloadsMenu-27966-Csv.htm'>https://results.aec.gov.au/27966/Website/HouseDownloadsMenu-27966-Csv.htm</a></p>

<p>Australian Electoral Commission. (2025). 2025 federal election - House of Representatives downloads. <a href='https://tallyroom.aec.gov.au/HouseDownloadsMenu-31496-Csv.htm'>https://tallyroom.aec.gov.au/HouseDownloadsMenu-31496-Csv.htm</a></p>

<p>Australian Electoral Commission. (n.d.). Federal electoral boundary GIS data. <a href='https://www.aec.gov.au/electorates/gis/gis\_datadownload.htm'>https://www.aec.gov.au/electorates/gis/gis\_datadownload.htm</a></p>

")

)

)

)

# Server

server <- function(input, output) {

# Tab 1: Party Performance

output$party\_performance\_summary <- renderText({

"TThis project utilizes interactive visualizations to showcase the political changes in Australia between the 2022 and 2025 federal elections, focusing on party seat performance and electoral division distribution. By selecting parties and years and clicking on the map, you can explore how shifts in voter preferences and regional differences have shaped the political landscape."

})

output$seats\_bar <- renderPlotly({

data <- rbind(

seats\_2022\_winners %>% mutate(Year = "2022"),

seats\_2025\_winners %>% mutate(Year = "2025")

) %>% filter(PartyNm %in% input$party\_seats)

plot\_ly(data, x = ~PartyNm, y = ~Seats, color = ~Year, type = "bar") %>%

layout(barmode = "group", title = "Seats Won by Party")

})

output$seats\_map <- renderLeaflet({

if (input$year\_map == "2022") {

data <- merge(divisions\_2022, seats\_2022 %>% filter(Elected == "Y") %>% distinct(DivisionNm, .keep\_all = TRUE),

by.x = "CED\_NAME16", by.y = "DivisionNm", all.x = TRUE)

data$PartyAb[is.na(data$PartyAb)] <- "No Data"

} else {

data <- merge(divisions\_2025, seats\_2025 %>% filter(Elected == "Y") %>% distinct(DivisionNm, .keep\_all = TRUE),

by.x = "CED\_NAME16", by.y = "DivisionNm", all.x = TRUE)

data$PartyAb[is.na(data$PartyAb)] <- "No Data"

}

# Dynamically generate colors based on unique parties

unique\_parties <- unique(data$PartyAb)

party\_colors <- colorFactor(

palette = rainbow(length(unique\_parties)), domain = unique\_parties)

# Create map

map <- leaflet(data) %>%

addTiles() %>%

addPolygons(

fillColor = ~party\_colors(PartyAb),

fillOpacity = 0.7,

weight = 1.5,

color = "black",

popup = ~paste(CED\_NAME16, PartyNm)

)

map <- map %>% addControl(html = "<h4>Electoral Division Winners</h4>", position = "topright",className = "map-title") })

output$changed\_seats <- DT::renderDataTable({

if (input$year\_map == "2022") {

changed\_seats\_2022

} else {

changed\_seats\_2025

}

})

# Tab 2: Voting Patterns

output$voting\_patterns\_summary <- renderText({

"This tab displays voting patterns based on first preference votes. The bar chart shows the vote share for selected parties in 2022 and 2025, allowing comparison of party support over time. The pie chart illustrates the overall distribution of first preference votes for the selected year, highlighting shifts in voter preferences in Australia's political landscape."

})

output$votes\_bar <- renderPlotly({

data <- rbind(

first\_prefs\_party\_2022 %>% mutate(Year = "2022"),

first\_prefs\_party\_2025 %>% mutate(Year = "2025")

) %>% filter(PartyNm %in% input$party\_votes)

plot\_ly(data, x = ~PartyNm, y = ~TotalPercentage, color = ~Year, type = "bar") %>%

layout(barmode = "group", title = "First Preference Vote Share")

})

output$votes\_pie <- renderPlotly({

data <- if(input$year\_pie == "2022") first\_prefs\_party\_2022 else first\_prefs\_party\_2025

data <- data %>% filter(TotalPercentage > 1)

plot\_ly(data, labels = ~PartyNm, values = ~TotalPercentage, type = "pie") %>%

layout(title = paste("Vote Distribution", input$year\_pie))

})

# Tab 3: Seats Change Analysis

output$seat\_changes\_summary <- renderText({

total\_changed <- nrow(changed\_seats\_2025)

gained\_by\_party <- changed\_seats\_2025 %>%

group\_by(PartyAb) %>%

summarise(Gained = n(), .groups = "drop")

lost\_by\_party <- changed\_seats\_2025 %>%

group\_by(PreviousPartyAb) %>%

summarise(Lost = n(), .groups = "drop")

changes\_by\_state <- changed\_seats\_2025 %>%

group\_by(StateAb) %>%

summarise(Changes = n(), .groups = "drop")

summary\_text <- paste(

"Between the 2022 and 2025 federal elections, a total of", total\_changed, "seats changed hands, reflecting significant shifts in Australia's political landscape.\n",

"The Australian Labor Party (ALP) gained",

ifelse(any(gained\_by\_party$PartyAb == "ALP"), gained\_by\_party$Gained[gained\_by\_party$PartyAb == "ALP"], 0), "seats.\n",

"The Liberal Party (LP) gained",

ifelse(any(gained\_by\_party$PartyAb == "LP"), gained\_by\_party$Gained[gained\_by\_party$PartyAb == "LP"], 0), "seats.\n",

"The states with the most seat changes were:\n",

paste(changes\_by\_state$StateAb, "with", changes\_by\_state$Changes, "changes", sep = " ", collapse = "\n"),

paste("\nPositive Aspects:\n",

"- The Australian Labor Party (ALP) gained seats in states like Victoria (VIC) and New South Wales (NSW), indicating voter support for its policies.\n",

"- The rise of independent candidates (e.g., IND) suggests a reevaluation of traditional parties, fostering political diversity.\n",

"Negative Risks:\n",

"- The Liberal Party (LP) and National Party (NP) lost seats in traditional strongholds, potentially weakening the coalition's influence.\n",

"- Uneven regional distribution of seat changes, with stability in Queensland (QLD) and Western Australia (WA) but volatility in VIC and NSW, could lead to parliamentary division or policy-making challenges.\n",

"Overall, these changes highlight both the vitality of Australian politics and the challenges traditional parties face in adapting to voter demands."

))

summary\_text

})

data <- merge(divisions\_2025, changed\_seats\_2025, by.x = "CED\_NAME16", by.y = "DivisionNm", all.x = TRUE)

output$changed\_seats\_map <- renderLeaflet({

if (input$state\_changed == "All") {

data\_filtered <- data

} else {

data\_filtered <- data %>% filter(StateAb == input$state\_changed)

}

leaflet(data\_filtered) %>%

addTiles() %>%

addPolygons(

fillColor = ~ifelse(!is.na(PreviousPartyAb), "red", "gray"),

fillOpacity = 0.7,

weight = 1,

color = "black",

popup = ~paste(CED\_NAME16, ifelse(!is.na(PreviousPartyNm), paste("Changed from", PreviousPartyNm, "to", PartyNm), "No change"))

)

})

output$changed\_seats\_chart <- renderPlotly({

changed\_by\_state <- changed\_seats\_2025 %>%

group\_by(StateAb) %>%

summarise(Count = n(), .groups = "drop")

plot\_ly(changed\_by\_state, x = ~StateAb, y = ~Count, type = "bar") %>%

layout(title = "Number of Seats Changed Hands by State", xaxis = list(title = "State"), yaxis = list(title = "Count"))

})

output$changed\_seats\_table <- DT::renderDataTable({

if (input$state\_changed == "All") {

changed\_seats\_2025 %>% select(DivisionNm, StateAb, PreviousPartyAb, PartyAb)

} else {

changed\_seats\_2025 %>% filter(StateAb == input$state\_changed) %>% select(DivisionNm, StateAb, PreviousPartyAb, PartyAb)

}

})

}

# Run the app

shinyApp(ui, server)