MATH2404 Assignment 3

Name

Timm Rahrt

Student number

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. 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.
- ☒ 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 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://timmrahrt.shinyapps.io/Assessment3/

References

- Queensland Government. (2022). Queensland offence rates—monthly from July 1997. Open Data Portal. Retrieved November 27, 2024, from https://www.data.qld.gov.au/dataset/offence-rates-monthly-from-july-1997
- Bureau of Crime Statistics and Research (BOCSAR). (2024). Criminal incident data. Retrieved from https://bocsar.nsw.gov.au/statistics-dashboards/open-datasets/criminal-offences-data.html
- Western Australian Government. (2024). Crime statistics. Retrieved from https://www.wa.gov.au/organisation/western-australia-police-force/crime-statistics
- Crime Statistics Agency Victoria. (2023). Recorded offences. Retrieved from https://www.crimestatistics.vic.gov.au/crime-statistics/latest-victorian-crime-data/recorded-offences-2
- Australian Institute of Health and Welfare. (2024). Alcohol, tobacco & other drugs in Australia: Data. Retrieved from https://www.aihw.gov.au/reports/alcohol/alcohol-tobacco-other-drugs-australia/data
- Australian Bureau of Statistics. (2020). Labour force, Australia, detailed, August 2020. Retrieved from https://www.abs.gov.au/statistics/labour/employment-and-unemployment/labour-force-australia-detailed/latest-release
- RDocumentation.org. (2017). Lubridate package (Version 1.9.3). Retrieved from https://www.rdocumentation.org/packages/lubridate/versions/1.9.3
- RDocumentation.org. (2024). Cell_cols function (Version 1.1.0). Retrieved from https://www.rdocumentation.org/packages/cellranger/versions/1.1.0/topics/cell_cols
- Tidyverse. (2024). Apply a function (or functions) across multiple columns — across. Retrieved from https://dplyr.tidyverse.org/reference/across.html
- Fjelstul, J. C. (2022). Regular expressions in R. Retrieved November 28, 2024, from https://jfjelstul.github.io/regular-expressions-tutorial/
- Tidyverse. (2024). Bind multiple data frames by row — bind_rows. Retrieved from https://dplyr.tidyverse.org/reference/bind_rows.html

- Zach. (2022). How to calculate lagged values in R (with examples). Statology. Retrieved from https://www.statology.org/r-lag/
- Tidyverse. (2024). Replace NAs with specified values — replace_na. Retrieved from https://tidyr.tidyverse.org/reference/replace_na.html
- Loopple. (2024). AI color palette generator. Retrieved November 28, 2024, from https://www.loopple.com/color-palette-generator/
- NNK. (2022). R - Replace NA values with 0 (zero). Spark by {Examples}. Retrieved from https://sparkbyexamples.com/r-programming/replace-na-values-with-zero-in-r-dataframe/.
- Tidyverse. (2024). Mutating joins — mutate-joins. Retrieved from https://dplyr.tidyverse.org/reference/mutate-joins.html
- Australian Institute of Health and Welfare. (2023). National drug strategy household survey: Summary. Retrieved from https://www.aihw.gov.au/reports/illicit-use-of-drugs/national-drug-strategy-household-survey/contents/summary
- Grolemund, G., Cheng, J., & Cetinkaya-Rundel, M. (2017). Shiny - Customize your UI with HTML. Retrieved from https://shiny.posit.co/r/articles/build/html-tags/
- Shiny.posit.co. (2015). Shiny - Using action buttons. Retrieved from https://shiny.posit.co/r/articles/build/action-buttons/
- Shiny. (2019). Shiny - Interactive plots. Retrieved November 28, 2024, from https://shiny.posit.co/r/articles/build/plot-interaction/
- Plotly.com. (2024). Plotly. Retrieved from https://plotly.com/ggplot2/

Assignment Code

Shiny Code to for data preprocessing and the Shinydashboard

```
## 0. Setup - Load all libraries required

library(readr)

library(shiny)

library(shinydashboard)

library(plotly)

library(ggplot2)

library(tidyr)

library(dplyr)

library(magrittr)

library(readxl)

library(lubridate)
```

```
library(gganimate)


# Set your working directory

#setwd("~/Desktop/RMIT/Data Visualisation and Communication/Assessments/Assignment 3")


## 1. Build df Graph 1: Lineplot - Drug Possession in Australia's biggest States


# Load the first dataset for the first visualisation - QLD

Crime_QLD <- read_csv("QLD_Reported_Offences_Rates.csv")

# Split the `Month Year` column into two separate columns

Crime_QLD %<>% separate(`Month Year`, into = c("Month", "Year"), sep = 3)

# Drop 204 rows to start at Year JUL 2014 (to align with later df's)

Crime_QLD <- Crime_QLD[-c(1:204), ]

# Next, remove the column Month and fix Year

Crime_QLD %<>% select(-Month)

Crime_QLD$Year <- as.numeric(Crime_QLD$Year) + 2000

# Select necessary columns, sum values by year and adjust their type

subset_Crime_QLD <- Crime_QLD %>% select(Year, `Drug Offences`,`Possess Drugs`)

subset_Crime_QLD <- subset_Crime_QLD %>% group_by(Year) %>%

  summarise(`Drug Offences` = sum(`Drug Offences`,  na.rm=TRUE),

            `Possess Drugs` = sum(`Possess Drugs`, na.rm=TRUE))

subset_Crime_QLD$State <- as.factor('QLD')

# Drop Year 2014 to align with other df's

subset_Crime_QLD %<>%  filter(Year != 2014)

# check results

head(subset_Crime_QLD, 20)
```

```
# Load the second dataset for the first visualisation - NSW

Crime_NSW <- read_excel('NSW_Crime.xlsx', sheet = "Summary of offences",
range=cell_cols("Q:Z"))

# Amend the variable names and select all needed rows

colnames(Crime_NSW) <- Crime_NSW[1, ]

Crime_NSW <- Crime_NSW[-1, ]

subset_Crime_NSW <- Crime_NSW[32:37, ]

# Sum all values by Year

subset_Crime_NSW %<>% summarise(across(everything(), ~ sum(as.numeric(.),
na.rm=TRUE)))

# Change the wide format to long and fix the Year column

subset_Crime_NSW %<>% pivot_longer(cols = starts_with("July"),

                                   names_to = "Year",

                                   values_to = "Possess Drugs")

subset_Crime_NSW %<>% mutate(Year = sub("July ([0-9]{4}) -.*", "\\1", Year))

subset_Crime_NSW$Year <- as.numeric(subset_Crime_NSW$Year)

subset_Crime_NSW$Year <- subset_Crime_NSW$Year + 1

# Adjust the State variable and check the result

subset_Crime_NSW$State <- factor("NSW")

head(subset_Crime_NSW,20)



# Load the third dataset for the first visualisation - WA

Crime_WA <- read_excel('wa-police-force-crime-timeseries.xlsx', sheet = 'Western
Australia', skip = 7)
```

```
# Again, select columns and filter by rows - add rows to have the total

subset_Crime_WA <- Crime_WA %>% select(`Month and Year`, `Drug Possession`,`Possession of Drug Paraphernalia`)

subset_Crime_WA <- subset_Crime_WA %>% filter(`Month and Year` >= "2014-07-01" & `Month and Year` < "2024-07-01")

subset_Crime_WA %<>% mutate(`Drug Possession` = `Drug Possession` + `Possession of Drug Paraphernalia`) %>%

  select(`Month and Year`, `Drug Possession`)

subset_Crime_WA %<>% mutate(Year = format(`Month and Year`, "%Y")) %>%

  group_by(Year) %>% summarise(`Possess Drugs` = sum(`Drug Possession`, na.rm=TRUE))

subset_Crime_WA$`Possess Drugs` <- subset_Crime_WA$`Possess Drugs` / 100

# Adjust the column types and delete Year == 2014

subset_Crime_WA$State <- factor('WA')

subset_Crime_WA$`Possess Drugs` <- as.numeric(subset_Crime_WA$`Possess Drugs`)

subset_Crime_WA$Year <- as.numeric(subset_Crime_WA$Year)

subset_Crime_WA %<>%  filter(Year != 2014)

# Check results

head(subset_Crime_WA, 20)



# Load the last dataset for the first visualisation - VIC

Crime_VIC <- read_excel("Victoria_Crime.xlsx", sheet="Table 01")

# Subset by needed rows and sort by descending Year

subset_Crime_VIC <- Crime_VIC %>% filter(`Offence Subgroup` == "C32 Drug possession") %>% arrange(Year)

subset_Crime_VIC %<>% select(Year, `Rate per 100,000 population`, `Offence Subgroup`)

# Rename column to match all other dataframes variables `Possess Drugs`
```

```
subset_Crime_VIC %<>% rename(`Possess Drugs` = `Rate per 100,000 population`)

# Fix the column type and display the results

subset_Crime_VIC$State <- factor("VIC")

head(subset_Crime_VIC, 20)



# Merge all dataframes to `merged_df`

merged_df <- bind_rows(subset_Crime_NSW %>% select(Year, `Possess Drugs`, State),

                       subset_Crime_VIC %>% select(Year, `Possess Drugs`, State),

                       subset_Crime_QLD %>% select(Year, `Possess Drugs`, State),

                       subset_Crime_WA %>% select(Year, `Possess Drugs`, State))

# Add a column to display the yearly index change (in %)

merged_df %<>% group_by(State) %>% arrange(Year) %>%

  mutate(`Percentage Change` = (100 * (`Possess Drugs` - lag(`Possess Drugs`)) /
lag(`Possess Drugs`))) %>%

  ungroup()

merged_df$`Percentage Change` <- replace_na(merged_df$`Percentage Change`, 0)

# Check for missing data

sum(is.na(merged_df)) # Will Return 0



## 2. Build df Graph 2: Scatterplot - Drug Possession and Usage in NSW



# Load the dataset

drug_cat_NSW <- read_excel('NSW_Crime.xlsx', sheet = "Summary of offences",
range=cell_cols("G:P"))

# Rename the columns, Subset rows and factorise `Drug`
```

```
colnames(drug_cat_NSW) <- drug_cat_NSW[1, ]

drug_cat_NSW <- drug_cat_NSW[-1, ]

drug_cat_NSW <- drug_cat_NSW[32:37, ]

drug_cat_NSW$Drug <- factor(c("Cocaine", "Narcotics", "Cannabis", "Amphetamines",
"Ecstasy", "Other drugs"))

drug_cat_NSW %<>% pivot_longer(cols = starts_with("July"),

                               names_to = "Year",

                               values_to = "Drug Usage")

# Fix the Year variable and column types

drug_cat_NSW %<>% mutate(Year = sub("July ([0-9]{4}) -.*", "\\1", Year))

drug_cat_NSW$Year <- as.numeric(drug_cat_NSW$Year)

drug_cat_NSW$Year <- drug_cat_NSW$Year + 1

drug_cat_NSW$`Drug Usage` <- as.numeric(drug_cat_NSW$`Drug Usage`)

# Add `Max Year` column for each `Drug Usage`

drug_cat_NSW <- drug_cat_NSW %>%

  group_by(Drug) %>%

  mutate(`Max Year` = Year[which.max(`Drug Usage`)])

# Check the results

print(drug_cat_NSW, n =20)

# Check for missing data

sum(is.na(drug_cat_NSW)) # Will Return 0
```

3. Build df Graph 3: Barplot - Drug Deaths by major cities

```
# Load the dataset

Drug_deaths <- read_excel('Drug_Death.xlsx', sheet = "Table 1", skip =1)
```

```
# Remove row 1, select necessary columns and subset for rows

Drug_deaths <- Drug_deaths[-1,]

Drug_deaths %<>% select(Year, Alcohol, Cocaine, Cannabinoids, `All opioids`, `All antidepressants`)

Drug_deaths <- Drug_deaths[1:14,]

# Amend column types and change the wide dataframe to long

Drug_deaths$Cocaine <- as.numeric(Drug_deaths$Cocaine)

Drug_deaths$Cannabinoids <- as.numeric(Drug_deaths$Cannabinoids)

Drug_deaths2 <- Drug_deaths %>%

  pivot_longer(cols = -Year, names_to = "Drug", values_to = "Deaths") %>%

  mutate(Year = as.numeric(Year)) %>%

  group_by(Drug) %>%

  arrange(Year) %>%

  mutate(`Percentage Change` = (Deaths - lag(Deaths)) / lag(Deaths) * 100) %>% ungroup()
# Calculate percentage change of `Deaths`

# Create custom colours

custom_c <- c("Alcohol" = "#FF6F61", "All antidepressants" = "#A9A52B", "All opioids" = "#2AA198",

              "Cannabinoids" = "#3B99FC", "Cocaine" = "#E67FB2")

Drug_deaths2 %<>% mutate(color = custom_c[Drug])

# Check results

head(Drug_deaths2, 20)

# Check for missing data

sum(is.na(Drug_deaths2)) # Will Return 5 (first 5 `Percentage Change` values)

Drug_deaths2$`Percentage Change`[is.na(Drug_deaths2$`Percentage Change`)] <- 0 # Replace NA with 0
```

4. Build df Graph 4: Scatterplot - Unemployment Rate vs Drug Possession by State

```
# Create the Unemployment Rate (by State) df

# Load Unemployment rate NSW and VIC

Unemployment_NSW_VIC <- read_excel("Unemployment_rate.xlsx", sheet = "Data1")

Unemployment_NSW_VIC %<>% select(`...1`, `> New South Wales ;  Unemployment rate ;  Persons ;`,

                                 `> Victoria ;  Unemployment rate ;  Persons ;`)

# Load Unemployment rate NSW and VIC

Unemployment_QLD <- read_excel("Unemployment_rate.xlsx", sheet = "Data2")

Unemployment_QLD %<>% select(`...1`, `> Queensland ;  Unemployment rate ;  Persons ;`)

# Load Unemployment rate NSW and VIC

Unemployment_WA <- read_excel("Unemployment_rate.xlsx", sheet = "Data3")

Unemployment_WA %<>% select(`...1`, `> Western Australia ;  Unemployment rate ;  Persons ;`)

# Merge all together, rename columns and drop the first 9 rows

Unemployment_AU <- Unemployment_NSW_VIC %>% full_join(Unemployment_QLD, by = '...1') %>% full_join(Unemployment_WA, by = '...1')

colnames(Unemployment_AU) <- c("Year", "NSW", "VIC", "QLD", "WA")

Unemployment_AU <- Unemployment_AU[-c(1:9), ]

# Change the numeric Year column back to a date ("1899-12-30" is Excels `origin` date), extract just the year

Unemployment_AU %<>% mutate(Year = format(as.Date(as.numeric(Year), origin = "1899-12-30"), "%Y"))

# Calculate the mean Unemployment rate of each year

Unemployment_AU %<>% group_by(Year) %>%

  summarise(across(NSW:WA, ~mean(as.numeric(.), na.rm=TRUE)))

# Drop 24 rows to start from year 2015 to match merged_df
```

```
Unemployment_AU <- Unemployment_AU[-c(1:24), ]

# Transform from wide to long format

Unemployment_AU %<>% pivot_longer(cols = NSW:WA,

                                  names_to = "State",

                                  values_to = "Unemployment Rate")

# Add a new column `Unemployment Rate % Change` for hover information

Unemployment_AU %<>% group_by(State) %>% arrange(Year) %>%

  mutate(`Unemployment Rate % Change` = ((`Unemployment Rate` - lag(`Unemployment Rate`)) / lag(`Unemployment Rate`)) *100) %>%

  ungroup()

# Correct data types

Unemployment_AU$Year <- as.numeric(Unemployment_AU$Year)

Unemployment_AU$State <- factor(Unemployment_AU$State)

# Merge `merged_df` to `Unemployment_AU` by Year and State

final_merged_df <- merged_df %>% left_join(Unemployment_AU, by = c("Year", "State"))

# Check result

head(final_merged_df,20)

# Check for missing data

sum(is.na(final_merged_df)) # Will Return 4 (first 4 `Unemployment Rate % Change` values)

final_merged_df$`Unemployment Rate % Change`[is.na(final_merged_df$`Unemployment Rate % Change`)] <- 0 # Now starting from 0




## 1. Define UI

ui <- fluidPage(
```

```
  titlePanel("The Shift in Drug Usage in Australia"),

  sidebarLayout( # Incl. our sidebar and main panel with the graphs

    sidebarPanel( # All content for the sidebar

      h3("About this Dashboard"), # HTML heading

      HTML("Is Australia facing a quiet drug epidemic?"),

      br(),

      HTML("This dashboard aims to provide insights into drug possession and usage trends including drug-related deaths across major cities. It also highlights how societal factors can impact the trend including unemployment or epidemics such as Covid. You can hover over each data point to gain further insights!"),

      br(),

      HTML("The visualisation addresses those interested in drug-related trends across Australian's major cities as well as policy makers and public health officials who allocate resources for drug prevention and support programs."),

      tags$ul( # Creates a list

        br(),

        tags$li("How does drug possession shift in Australian's major cities?"), # Pullet points

        tags$li("Did the Covid Lockdown increase drug consumption across NSW?"),

        tags$li("Which drug is killing the most Australians and is one substance taking more lives each year?"),

        tags$li("Is there a relationship between a high Unemployment Rate and Drug Possession?")),

      br(),

      h4("Data Sources"), # HTML Sub-heading,

      tags$ul(

        tags$li("Queensland Government (2022). Queensland offence rates. Retrieved from https://www.data.qld.gov.au."),

        tags$li("BOCSAR (2024). Criminal incident data. Retrieved from https://bocsar.nsw.gov.au."),
```

```
        tags$li("Western Australian Government (2024). Crime statistics. Retrieved from https://www.wa.gov.au."),

        tags$li("Crime Statistics Agency Victoria (2023). Recorded offences. Retrieved from https://www.crimestatistics.vic.gov.au."),

        tags$li("AIHW (2024). Alcohol, tobacco & other drugs in Australia. Retrieved from https://www.aihw.gov.au."),

        tags$li("Australian Bureau of Statistics. (2020). Labour force, Australia, detailed, August 2020. Retrieved from https://www.abs.gov.au/statistics/labour/employment-and-unemployment/labour-force-australia-detailed/latest-release")),

      br(), # Adds a line break in our panel

      actionButton("fact_button", "Did you know? CLICK ME"), # Adds a button we can press

      br(), br(), # Two more breaks

      textOutput("content")), # Will output the fact

    mainPanel( # Content of our main panel

      fluidRow(

        plotlyOutput("linePlot", height = "47vh", width = "100%")), # Line plot with amended height and width

      div(style = "margin-top: 20px;", # Adds empty space

          tabsetPanel(tabPanel("Drug Usage", plotlyOutput("drugUsagePlot", height = "100%")), # Creates our tabs and creates the first for the scatterplot

                      tabPanel("Drug-Related Deaths", plotlyOutput("barPlot", height = "100%")), # Creates a second tab for the barplot

                      tabPanel("Unemployment vs Drug Possession", plotlyOutput("UnemploymentDrugUsagePlot", height = "100%")))))) # Creates the third tab for the scatterplot

)




## 2. Define server logic

server <- function(input, output, session) {
```

```
  # Graph 1: Lineplot - Drug Possession in Australia's biggest States
  output$linePlot <- renderPlotly({ # Renders the plot defined below
    p5 <- ggplot(data = merged_df, aes(x = Year, y = `Possess Drugs`, colour = State)) +
      geom_line(aes(group = State)) + # Line plot with each State
      geom_point(aes(text = paste0("Percentage Change: ",round(`Percentage Change`, 2), "%"))) + # Adds the points with the hover text
      labs(title = "Drug Possession in Australian's major States", # title and labels
           x = "Year (July - June)",
           y = "Drug Possession per 100k Persons") +
      scale_y_continuous(limits = c(0, NA)) + # x and y scale to not deceive the audience with wrong scaling
      scale_x_continuous(breaks = seq(2015, 2024, by = 3),
                         labels = seq(2015, 2024, by = 3)) +
      theme_minimal() + # style
      theme(strip.text = element_text(size = 9, face = "bold"),
            legend.position = "bottom", # move the legend to the bottom
            plot.title = element_text(size = 20, face = "bold", hjust = 0),
            plot.subtitle = element_text(size = 12, hjust = 0))

    ggplotly(p5) %>% layout( # Convert ggplot to plotly
      annotations = list( # adjust layout
        x = 0.0001, y = 1.0455,
        text = "Drug Possession per 100k Persons in the Last 9 Years",
        showarrow = FALSE,
        xref = "paper", yref = "paper",
        xanchor = "left", yanchor = "top",
```

```
      font = list(size = 18)),
    legend = list(orientation = "h", # Adjust legend position
                  x = 0.5, y = -0.2,
                  xanchor = "center"),
    margin = list(l = 100)) # Adjust the plot margins
})

# Graph 2: Scatterplot - Drug Possession and Usage in NSW
output$drugUsagePlot <- renderPlotly({
  ani_plot <- plot_ly(data = drug_cat_NSW,
                      x = ~Year,
                      y = ~`Drug Usage`,
                      color = ~Drug,
                      type = "scatter", # displays scatterplot
                      mode = "markers",
                      frame = ~Year,
                      text = ~paste0(round(`Drug Usage`, 1), " per 100k<br>",
                                     "Year: ", Year,"<br>",
                                     "Max Drug Usage: ", max(`Drug Usage`), " in ", `Max Year`),
                      textposition = "top center", # Position of the created annotations
                      hoverinfo = "text", # Show only the text in hover
                      marker = list(size = 12, opacity = 0.7)) %>% # Reduce transparency
    layout(
      title = "Drug Usage in NSW (2015-2024)",
      xaxis = list(title = "Year"),
      yaxis = list(title = "Drug Usage per 100k Persons", range = c(0,
max(drug_cat_NSW$`Drug Usage`, na.rm = TRUE))), # customise y label size
```

```
      showlegend = TRUE, # Include a legend
      shapes = list( # Vertical dotted (red) line for 2019.5
        list(type = "line",
             x0 = 2019.5, x1 = 2019.5,
             y0 = 0, y1 = max(drug_cat_NSW$`Drug Usage`, na.rm = TRUE),
             line = list(color = "red", width = 1, dash = "dot")), # Color of line
        list(type = "line", # Vertical dotted (red) line for 2020.5
             x0 = 2021.5, x1 = 2021.5,
             y0 = 0, y1 = max(drug_cat_NSW$`Drug Usage`, na.rm = TRUE),
             line = list(color = "red", width = 1, dash = "dot"))),
      annotations = list( # Vertical dotted (red) line for 2020.5
        list(x = 2020.5,
             y = max(drug_cat_NSW$`Drug Usage`, na.rm = TRUE) * 0.9, # Adjust the vertical
position
             text = "CoVid Lockdown",
             showarrow = FALSE,
             font = list(color = "red", size = 12))), # Color of text
      updatemenus = NULL) # Removes the upper left Play and Pause button
    ani_plot
  })

  # Graph 3: Barplot - Drug Deaths by major cities
  output$barPlot <- renderPlotly({
    ani_barplot <- plot_ly(data = Drug_deaths2,
                           x = ~Deaths,
                           y = ~Drug,
                           type = "bar", # displays barplot
```

```
                frame = ~Year,
                orientation = 'h',
                marker = list(color = ~color),
                text = ~paste0(round(Deaths), " deaths<br>",
                               ifelse(`Percentage Change` == 0, "No previous data", # if value of `Percentage Change` == 0 text is displayed
                                      paste0("Change: ", round(`Percentage Change`, 2), "%"))), # if != 0 the value + %
                textposition = "outside", # Annotation is not inside the bar
                hoverinfo = "text", # Keep our hover text
                hovertext = ~paste("Year: ", Year, "<br>", # Customises the text inside the hover
                                   "Deaths: ", round(Deaths), "<br>",
                                   ifelse(`Percentage Change` == 0, "Change: No previous data",
                                          paste0("Change: ", round(`Percentage Change`, 2), "%"))) ) %>%
    layout(title = "Drug-Related Deaths in Major Cities (2009 to 2022)",
           xaxis = list(title = "Total Deaths"),
           yaxis = list(title = "Drug Type", categoryorder = "total ascending"),
           showlegend = FALSE) # Deactive the legend
  ani_barplot
 })

 # Graph 4: Scatterplot - Unemployment Rate vs Drug Possession by State
 output$UnemploymentDrugUsagePlot <- renderPlotly({
  ani_unemployment_plot <- plot_ly(data = final_merged_df,
                                   x = ~`Unemployment Rate`,
                                   y = ~`Possess Drugs`,
```

```
                color = ~State,
                frame = ~Year,
                type = 'scatter',
                mode = 'markers+text', # Enables markers and also annotations
                marker = list(size = 12, opacity = 0.7),
                text = ~round(`Possess Drugs`, 1), # Annotation shows only Drug Possession value
                textposition = "top center", # Position annotation above markers
                hoverinfo = "text", # Show detailed information on hover
                hovertext = ~paste(
                  "<b>State: </b>", State, "<br>",
                  "<b>Year: </b>", Year, "<br>",
                  "<b>Drug Possession: </b>", round(`Possess Drugs`, 1), "<br>",
                  "<b>Drug Possession % Change: </b>", round(`Percentage Change`, 2), "%<br>",
                  "<b>Unemployment Rate: </b>", round(`Unemployment Rate`, 2), "%<br>",
                  "<b>Unemployment Rate % Change: </b>", round(`Unemployment Rate % Change`, 2), "%")) %>%
    layout(title = "Drug Possession by Unemployment Rate",
           xaxis = list(
             title = "Unemployment Rate (%)",
             zeroline = FALSE, # Hides the default y axis line
             range = c(0, max(final_merged_df$`Unemployment Rate`, na.rm = TRUE))),
           yaxis = list(
             title = "Drug Possession per 100k Persons",
             zeroline = FALSE), # Hides the default x axis line
           showlegend = TRUE,
```

```
            updatemenus = NULL)
    ani_unemployment_plot
  })


  # Displaying the fact-button when clicked
  output$content <- renderText({ # Renders the below text
    req(input$fact_button) # Registered button click
    "Australia ranks third in world-wide cocaine consumption. Approximately 4.5% of Australians population used cocaine in the last 12 months. Source: Australian Institute of Health and Welfare. (2023)"
  })
}



# 5. Run the application
shinyApp(ui = ui, server = server)
```