Project 1: Graphing

TOTAL: 5pts

GOAL

Make a graph using Washington, DC crime data.

Step 0: Prep

Install and load packages:

- We are specifically using the ggplot2 library within this package.
- install.packages('tidyverse')
- library(tidyverse)

Step 1: Get data

All of the datas come from here.

Washington, D.C. 2024 crime data:

- dc.data2024 <- read.csv("https://opendata.arcgis.com/datasets/c5a9f33ffca546babbd91de1969e742d_6.csv", stringsAsFactors = FALSE)

Washington, D.C. for the current year, and all the earlier years:

- dc.data2025 <- read.csv("https://opendata.arcgis.com/datasets/74d924ddc3374e3b977e6f002478cb9b_7.csv", stringsAsFactors = FALSE)
-
- dc.data2023 <- read.csv("https://opendata.arcgis.com/datasets/89561a4f02ba46cca3c42333425d1b87_5.csv", stringsAsFactors = FALSE)

- dc.data2022 <- read.csv("https://opendata.arcgis.com/datasets/f9cc541fc8c04106a05a1a4f1e7e813c_4.csv", stringsAsFactors = FALSE)
- dc.data2021 <- read.csv("https://opendata.arcgis.com/datasets/619c5bd17ca2411db0689bb0a211783c_3.csv", stringsAsFactors = FALSE)
- dc.data2020 <- read.csv("https://opendata.arcgis.com/datasets/f516e0dd7b614b088ad781b0c4002331_2.csv", stringsAsFactors = FALSE)
- dc.data2019 <- read.csv("https://opendata.arcgis.com/datasets/f08294e5286141c293e9202fcd3e8b57_1.csv", stringsAsFactors = FALSE)
- dc.data2018 <- read.csv("https://opendata.arcgis.com/datasets/38ba41dd74354563bce28a359b59324e_0.csv", stringsAsFactors = FALSE)
- dc.data2017 <- read.csv("https://opendata.arcgis.com/datasets/6af5cb8dc38e4bcbac8168b27ee104aa_38.csv", stringsAsFactors = FALSE)
- dc.data2016 <- read.csv("https://opendata.arcgis.com/datasets/bda20763840448b58f8383bae800a843_26.csv", stringsAsFactors = FALSE)
- dc.data2015 <- read.csv("https://opendata.arcgis.com/datasets/35034fcb3b36499c84c94c069ab1a966_27.csv", stringsAsFactors = FALSE)
- dc.data2014 <- read.csv("https://opendata.arcgis.com/datasets/6eaf3e9713de44d3aa103622d51053b5_9.csv", stringsAsFactors = FALSE)
- dc.data2013 <- read.csv("https://opendata.arcgis.com/datasets/5fa2e43557f7484d89aac9e1e76158c9_10.csv", stringsAsFactors = FALSE)
- dc.data2012 <- read.csv("https://opendata.arcgis.com/datasets/010ac88c55b1409bb67c9270c8fc18b5_11.csv", stringsAsFactors = FALSE)
- dc.data2011 <- read.csv("https://opendata.arcgis.com/datasets/9d5485ffae914c5f97047a7dd86e115b_35.csv", stringsAsFactors = FALSE)
- dc.data2010 <- read.csv("https://opendata.arcgis.com/datasets/fdacfbdda7654e06a161352247d3a2f0_34.csv", stringsAsFactors = FALSE)
- dc.data2009 <- read.csv("https://opendata.arcgis.com/datasets/73cd2f2858714cd1a7e2859f8e6e4de4_33.csv", stringsAsFactors = FALSE)
- dc.data2008 <- read.csv("https://opendata.arcgis.com/datasets/180d56a1551c4e76ac2175e63dc0dce9_32.csv", stringsAsFactors = FALSE)

Step 2: Clean data

Merge the annual data into one big table:

- dc.data.temp <- rbind(dc.data2008, dc.data2009, dc.data2010, dc.data2011, dc.data2012, dc.data2013, dc.data2014, dc.data2015, dc.data2016, dc.data2017, dc.data2018, dc.data2019, dc.data2020, dc.data2021, dc.data2022, dc.data2023, dc.data2024, dc.data2025)

Separate the datetime field into ‘date’ and ‘time’:

- dc.data <- separate(dc.data.temp, REPORT_DAT, into = c("date", "time"), sep = " ")

Format the date column as a date:

- dc.data$date <- as.Date(dc.data$date, format = "%Y/%m/%d")

Step 3: Enrich data

Calculate year, month, day of year, day of week, and hour of day:

- dc.data$year <- substr(dc.data$date, 0, 4)
- dc.data$month <- month(dc.data$date)
- dc.data$day <- day(dc.data$date)
- dc.data$dow <- weekdays(dc.data$date)
- dc.data$hour <- substr(dc.data$time, 0, 2)

Step 4: Graph data

Bar graphs:

- Follow these iterative steps to make progressively better bar graphs:
    - ggplot()
    - ggplot(dc.data, aes(SHIFT))
    - ggplot(dc.data, aes(SHIFT)) + geom_bar(stat = "count")
    - ggplot(dc.data, aes(SHIFT)) + geom_bar(stat = "count", aes(fill = OFFENSE))
    - ggplot(dc.data, aes(SHIFT)) + geom_bar(stat = "count", aes(fill = OFFENSE), position = position_stack(reverse = TRUE))
    - ggplot(dc.data, aes(SHIFT)) + geom_bar(stat = "count", aes(fill = OFFENSE), position = position_stack(reverse = TRUE)) + theme_classic()
    - ggplot(dc.data, aes(SHIFT)) + geom_bar(stat = "count", aes(fill = OFFENSE), position = position_stack(reverse = TRUE)) + theme_classic() + theme(legend.position = "bottom")

```
ggplot(dc.data, aes(SHIFT)) + geom_bar(stat = "count", aes(fill = OFFENSE), position = position_stack(reverse = TRUE)) + theme_classic() + facet_wrap(~DISTRICT)
```

Formatting:

- Make your graph an object, and then you can customize the formatting:'

```
ggplot(dc.data, aes(SHIFT)) + geom_bar(stat = "count", aes(fill = OFFENSE)) +
 labs(
  title = "Your title here",
  x = "x axis label here",
  y = "y axis label here"
 ) +
 theme(
  panel.grid.minor = element_blank(),
  panel.grid.major.y = element_blank(),
  panel.grid.major.x = element_line(),
  axis.ticks = element_blank()
 )
```

- For further customizations, including adjusting axis labels, this article has you covered

HERE IS THE SCRIPT WE CREATED IN CLASS ON 22 JANUARY 2025

Better Graphs:

These aren't mandatory, but get you thinking better about the data and what's possible…

- First, some data prep to group crime by 'Person' and 'Property':

```
dc.data$TYPE <- case_when(
 dc.data$OFFENSE %in%
  c(
   "ARSON",
   "BURGLARY",
   "MOTOR VEHICLE THEFT",
   "THEFT F/AUTO",
   "THEFT/OTHER") ~ "Property",
 dc.data$OFFENSE %in%
  c("ASSAULT W/DANGEROUS WEAPON", "HOMICIDE", "ROBBERY", "SEX ABUSE")
 ~ "Person")
```

- Second, a cleaner crimes by hour graph:

```
dc.data %>% ggplot() +
  geom_line(aes(x = xxxx), stat = "count", group = 1, color = "blue", size = 1) +
  labs(title = "DC Crimes by Hour Reported \n2008 - 202x", x = "Hour Reported", y =
"Number of Crimes (Thousands)", fill = "Type of Crime") +
  theme(plot.title = element_text(hjust = 0.5, size = 14),
        axis.text.x = element_text(color = "black", size = 10),
        axis.text.y = element_text(color = "black", size = 10),
        axis.ticks.y = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_line(color = "gray"),
        panel.background = element_blank()) +
  scale_y_continuous(limits = c(0, xxxxx), breaks = c(0,
5000,10000,15000,20000,25000,30000,35000), labels = c(0,5,10,15,20,25,30,35))
```

- Third, graph crime type by district:

```
dc.data %>% filter(DISTRICT != "NA") %>%
  ggplot() +
  geom_bar(aes(x = as.factor(xxx), fill = TYPE), stat = "count") +
  labs(title = "Crimes by DC Police District \n2010 - 202x", x = "District", y = "Number of
Crimes (Thousands)", fill = "Type of Crime") +
  theme(plot.title = element_text(hjust = 0.5, size = 14),
        axis.text.x = element_text(color = "black", size = 10),
        axis.text.y = element_text(color = "black", size = 10),
        axis.ticks.y = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_line(color = "gray"),
        panel.background = element_blank()) +
  scale_y_continuous(limits = c(0, 125000), breaks = c(0,
25000,50000,75000,100000,125000), labels = c(0,25,50,75,100,125))
```

- Fourth, graph a crime type as a percentage of weapon type/method:

```
dc.data %>% filter(TYPE == "xxx") %>%
  ggplot() +
  geom_bar(aes(x = OFFENSE, fill = METHOD), position = "fill") +
  labs(title = "DC xxx Crimes by Method \n2010 - 202x", x = "Crime", y = "Proportion of
Crime", fill = "Method") +
  theme(plot.title = element_text(hjust = 0.5, size = 14),
        axis.text.x = element_text(color = "black", size = 10),
        axis.text.y = element_text(color = "black", size = 10),
        axis.ticks.y = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_line(color = "gray"),
        panel.background = element_blank()) +
  scale_x_discrete(labels = c("Assault", "Homicide", "Robbery", "Sex Abuse")) +
  scale_y_continuous(labels = c("0%", "25%", "50%", "75%", "100%")) +
  scale_fill_manual(values = c("#1b9e77", "#d95f02","#7570b3"), labels = c("Gun",
"Knife", "Other"))
```

HERE IS THE SCRIPT WE BUILT IN CLASS ON 29 JANUARY 2025

Part II: SUBMISSION

Transform your script as a usable HTML file, via RMarkdown. You will want to copy/paste the most relevant parts of the script you've built into a clean, consolidated file. You create this output as a .Rmd file in RStudio, but what you submit will be an .HTML file.

Use this file as a template

1. Adjust the template with your code accordingly
2. Use the 'knit' button in RStudio to produce the output
    a. The .HTML is automatically saved to the same folder as your script
3. Click 'Open in Browser' to display in your web browser.

HERE IS MARKDOWN FILE WE MADE IN CLASS ON 29 JANUARY 2025

GRADES

- **Part I: Graph (4.5pts)**
    - Create three graphs using the data provided (**1.5pts per graph**)
    - Each graph should analyze a different variable in the data
        - □ *One graph should analyze **behavior***
        - □ *One graph should analyze **space***
        - □ *One graph should analyze **time***
        - □ NONE of the graphs should analyze SHIFT
    - Each graph should include a title and x/y axis labels
- **Part II: Submission (0.5pts)**
    - Project created in RMarkdown and submitted as an HTML or PDF file via Canvas

Please email me with any questions.