Activity #14 - First QMD File

Zhan Wang

2025-11-20

Armed Forces Data Wrangling Redux (Activities #08 and #10)

The table shows the number of Air Force officers by sex across each officer rank in the June 2025 Armed Forces dataset. Each row represents an officer rank, and the columns show how many men and women fall into that rank, letting us compare how the distribution of sex changes as rank increases. Because the data were expanded to one row per soldier, the counts reflect actual numbers of individual officers. From the table, men consistently appear in much higher numbers than women across all officer ranks, and the size of the difference varies by rank. Since these proportions are not consistent across the ranks, sex and rank do not appear to be independent for this subgroup of the Air Force.

	Rank	Male
1	Brigadier General	99
2	Captain	15715
3	Colonel	2663
4	First Lieutenant	5045
5	General	11
6	Lieutenant Colonel	7373
7	Lieutenant General	30
8	Major	9682
9	Major General	63
10	Second Lieutenant	5048

Popularity of Baby Names (Activity #13)

I chose the names Ashley, Matthew, Paula, and Sean because they represent people who are close to me, including friends and family, and I was curious to see how their popularity changed over time. The line graph shows yearly counts of each name from the late 1800s to the early 2000s, with different colors and line styles to make the trends easy to compare. The name Paula peaks around the 1950s before declining, Sean rises and falls a few times throughout the mid-1900s, Matthew grows steadily and peaks around the 1980s, and Ashley rises sharply after the mid-1970s and has the highest overall peak. Overall, the visualization makes it easy to see how each name follows a very different pattern across the decades.

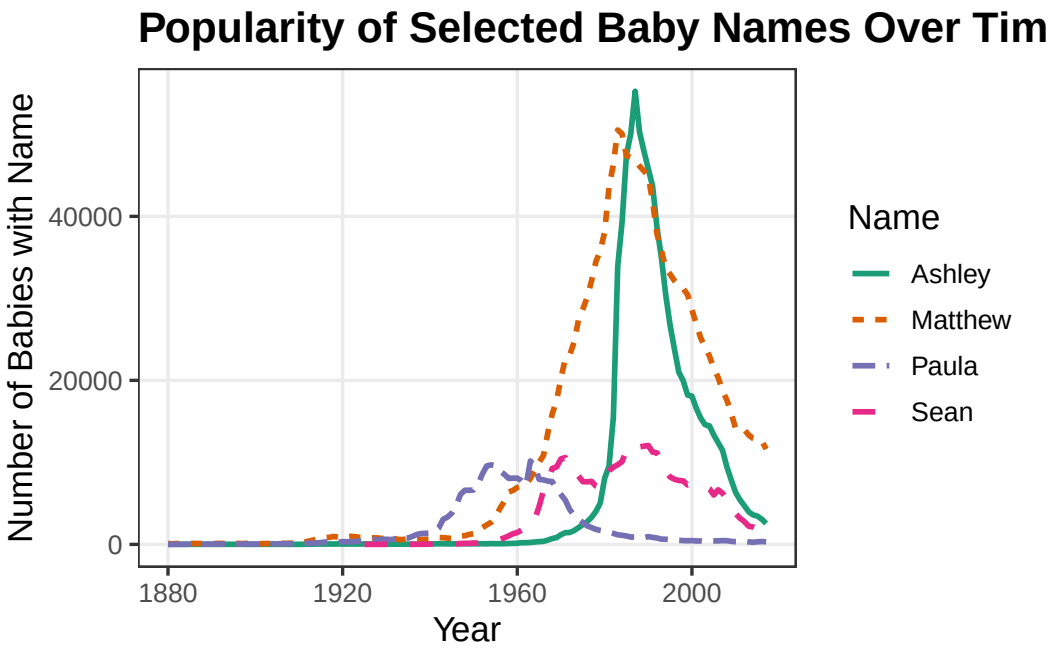


Figure 1: Popularity of Names Ashley, Matthew, Paula, and Sean Over Time.

Plotting a Mathematical Function (Activity #04)

This is the Box Problem, where you take a 36" $\times$ 48" sheet of paper, cut out squares of side length x from each corner, and fold up the sides to form an open-top box. The volume depends on how big the cutouts are, so the plot shows how the volume changes as x increases. The curve starts at zero, rises to a single peak, and then drops back down as the cutouts get too large to form a usable box. This visualization makes it easy to see where the maximum happens and how quickly the volume changes based on small differences in the cutout size.

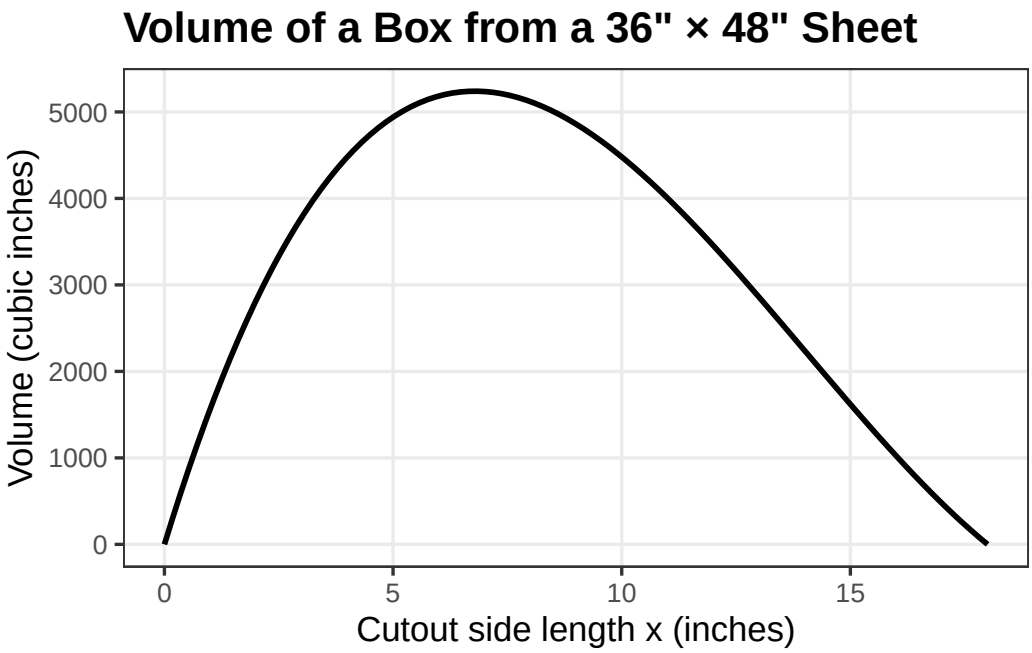


Figure 2: Volume of an open-top box made from a 36" × 48" sheet as a function of the cutout side length.

For this 36" × 48" sheet, the maximum volume occurs when the cutout side length is approximately 6.79 inches, which produces a volume of about 5,240 cubic inches.

What You Feel You've Learned So Far

When I think back to the start of this course, I feel like I went from very little R experience to being able to wrangle data, clean it up, and make helpful and interesting visualizations. In the beginning, even simple tasks like filtering or reshaping tables felt confusing, especially when datasets were messy or had extra header rows. Over time, the tidyverse workflow started to make more sense, and I learned how to build pipelines that move from raw data to polished results.

I especially enjoyed the visualization topics, since it was fun to design plots that are both readable and visually appealing. Writing in Quarto also pushed me to separate narrative from code, add informative figure captions, and include alt text so the graphics are more accessible. Overall, I now feel much more confident writing reproducible code, thinking through each step of an analysis, and explaining what my plots show. I still have more to learn, but I feel like I have a solid foundation for working with data in R beyond this course.

Code Appendix

Below is the code used to generate all results in this document.

Armed Forces Data Wrangling Code

```
library(tidyverse)
library(rvest)
library(stringr)

build_groups <- function(file_path) {
  raw <- readr::read_csv(
    file_path,
    col_names = FALSE,
    show_col_types = FALSE
  )

  raw <- raw[-1, ]

  branch_row <- as.character(raw[1, ])
  sex_row    <- as.character(raw[2, ])

  new_names <- ifelse(
    is.na(sex_row) | sex_row == "",
    branch_row,
    paste(branch_row, sex_row, sep = "_")
  )
  new_names[is.na(new_names)] <- "unknown"

  data <- raw[-c(1, 2), ]
  colnames(data) <- new_names
  colnames(data)[1] <- "pay_grade"

  groups <- data |>
    tidyr::pivot_longer(
      cols = -pay_grade,
      names_to = "branch_sex",
      values_to = "count_raw"
    ) |>
    dplyr::filter(str_detect(count_raw, "\\d")) |>
    dplyr::mutate(count = readr::parse_number(count_raw)) |>
    tidyr::separate(
      branch_sex,
      into = c("branch", "sex"),
      sep = "_",
      fill = "right",
      extra = "merge"
    ) |>
    dplyr::mutate(
      branch = str_to_title(branch),
      sex = dplyr::coalesce(str_to_title(sex), "Unknown")
```

```
    ) |>
    dplyr::select(pay_grade, branch, sex, count)

  groups
}

attach_ranks <- function(groups_df) {
  webRanks <- rvest::read_html(
    "https://neilhatfield.github.io/Stat184_PayGradeRanks.html"
  ) |>
    rvest::html_elements(css = "table") |>
    rvest::html_table()

  rawRanks <- webRanks[[1]]

  rawRanks[1, 1] <- "Type"
  rankHeaders <- rawRanks[1, ]
  names(rawRanks) <- rankHeaders[1, ]

  rawRanks <- rawRanks[-c(1, 26), ]

  cleanRanks <- rawRanks |>
    dplyr::select(!Type) |>
    tidyr::pivot_longer(
      cols = !`Pay Grade`,
      names_to = "Branch",
      values_to = "Rank"
    ) |>
    dplyr::mutate(
      Rank = dplyr::na_if(x = Rank, y = "--")
    )

  groups_df |>
    dplyr::mutate(
      pay_grade = as.character(pay_grade),
      branch = str_to_title(branch)
    ) |>
    dplyr::left_join(
      cleanRanks |>
        dplyr::rename(
          pay_grade = `Pay Grade`,
          branch = Branch
        ),
      by = c("pay_grade", "branch")
    )
}
```

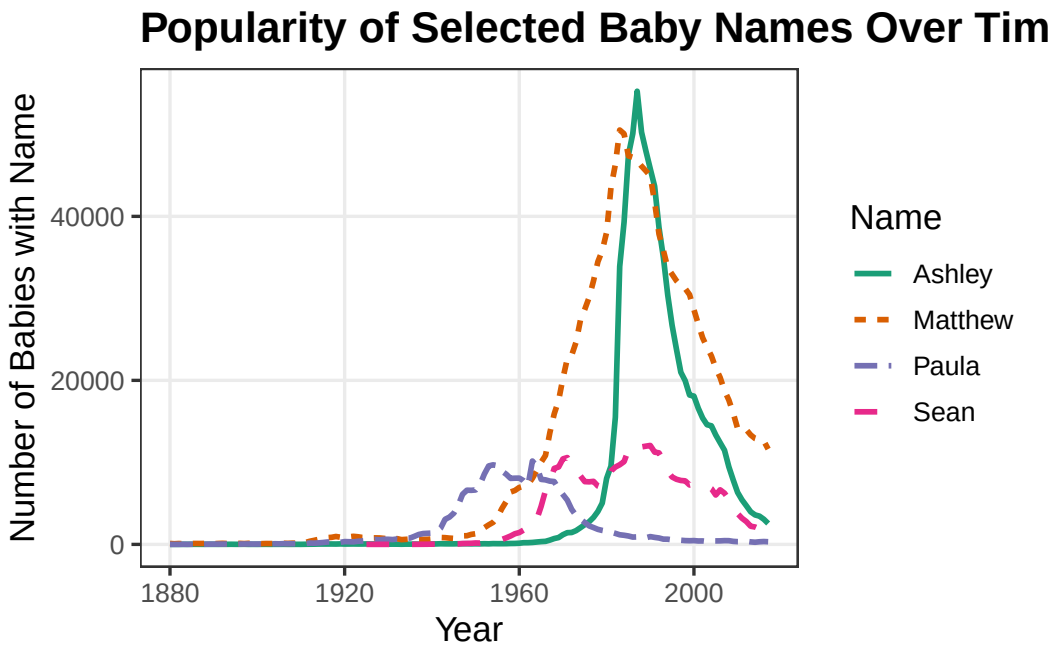
```r
to_individuals <- function(groups_with_rank_df) {
  groups_with_rank_df |>
    tidyr::uncount(weights = count, .remove = TRUE)
}

groups <- build_groups("~/Downloads/US_Armed_Forces_(6_2025) - Sheet1 (1).csv") |>
  attach_ranks()

individuals <- to_individuals(groups)

airforce_officers <- individuals |>
  dplyr::filter(
    branch == "Air Force",
    !is.na(Rank),
    str_starts(pay_grade, "O")
  )

airforce_officer_table <- airforce_officers |>
  dplyr::count(Rank, sex, name = "Count") |>
  tidyr::pivot_wider(
    names_from = sex,
    values_from = Count,
    values_fill = 0
  ) |>
  dplyr::arrange(Rank)

as.data.frame(airforce_officer_table)
```

```
                 Rank  Male
1   Brigadier General    99
2             Captain 15715
3             Colonel  2663
4    First Lieutenant  5045
5             General    11
6  Lieutenant Colonel  7373
7  Lieutenant General    30
8               Major  9682
9       Major General    63
10  Second Lieutenant  5048
```

Popularity of Baby Names Code

```r
library(babynames)
library(dplyr)
library(ggplot2)
```

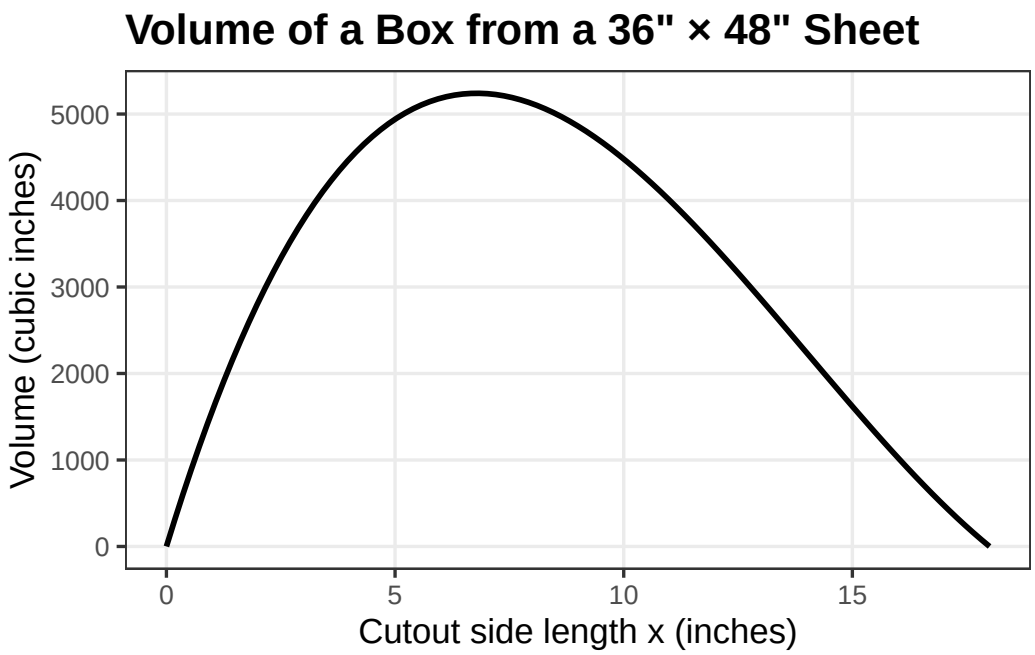
```r
names_selected <- c("Ashley", "Matthew", "Paula", "Sean")

baby_yr <- babynames |>
  dplyr::filter(name %in% names_selected) |>
  dplyr::group_by(name, year) |>
  dplyr::summarise(count = sum(n), .groups = "drop") |>
  dplyr::arrange(name, year)

baby_plot <- ggplot(
  data = baby_yr,
  mapping = aes(
    x = year,
    y = count,
    color = name,
    linetype = name
  )
) +
  geom_line(linewidth = 1) +
  scale_color_brewer(palette = "Dark2") +
  labs(
    title = "Popularity of Selected Baby Names Over Time",
    x = "Year",
    y = "Number of Babies with Name",
    color = "Name",
    linetype = "Name",
    alt = "Line chart shows the number of babies given the names Ashley, Matthew, Paula, and Se
  ) +
  theme_bw(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold"),
    panel.grid.minor = element_blank()
  )

baby_plot
```



Box Problem Code

```
library(ggplot2)

getVolume <- function(sideLength, paperLength = 48, paperWidth = 36) {
  sideLength * (paperWidth - 2 * sideLength) * (paperLength - 2 * sideLength)
}

volume_function <- function(x) {
  getVolume(sideLength = x)
}

box_plot <- ggplot(data = data.frame(x = c(0, 18)), aes(x = x)) +
  stat_function(
    fun = volume_function,
    linewidth = 1
  ) +
  labs(
    title = "Volume of a Box from a 36\" × 48\" Sheet",
    x = "Cutout side length x (inches)",
    y = "Volume (cubic inches)",
    alt = "Curve showing the volume of an open-top box made from a 36 by 48 inch sheet as a fu
  ) +
  theme_bw(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold"),
```

```
    panel.grid.minor = element_blank()
  )

max_result <- optimize(
  f = volume_function,
  interval = c(0, 18),
  maximum = TRUE
)

x_max <- max_result$maximum
volume_max <- max_result$objective

box_plot
```




```
x_max
```

```
[1] 6.788902
```

```
volume_max
```

```
[1] 5239.819
```