Activity #14 – A First QMD File

Jayadeep Raghav Vadlapati

2025-11-18

Armed Forces Data Wrangling Redux

In this section, I revisit the Armed Forces data wrangling from Activity #08 and demonstrate that I can produce a fully reproducible pipeline that starts from the PDF table and ends with a clean data frame where each row corresponds to an individual service member. Using `rvest` and `tidyverse`, I tidy the pay-grade and rank reference table, reshape the PDF counts into long format, merge in rank and category information, and then expand the grouped counts to an individual-level data frame with over 1.2 million rows. All steps can be re-run on any computer without needing local data files, so another analyst could reproduce the results without alteration.

Frequency Table for the Armed Forces

To study how sex and rank relate within a specific sub-group, I focus on **Army enlisted personnel**. From the individual-level data, I build a two-way frequency table that summarizes the number of male and female soldiers at each enlisted rank (E1–E9). This table lets me see how representation changes as rank increases.

Looking at the table, there are many more enlisted men than women at every rank, but the gap is especially large at the lower ranks where total counts are highest. As we move up the rank structure toward E8 and E9, the absolute counts shrink for both groups, but women remain a relatively small fraction of the total. This pattern suggests that **sex and rank are not independent** within this Army-enlisted sub-group: men are far more prevalent at all levels, and the gender imbalance persists into the senior enlisted ranks rather than disappearing with promotion.

Table 1: Counts of male and female Army enlisted personnel by rank.

[

Army Enlisted Personnel by Rank and Sex] Army Enlisted Personnel by Rank and Sex

Rank

Popularity of Baby Names

In Activity #13 I examined the popularity of four names—**Aiden, Jacob, Emma, and Sophia**—using the `babynames` data. For this QMD document, I reuse that work to create a polished visualization that tracks how these names rise and fall over time, separated by sex. This helps highlight both long-term naming trends and how strongly each name is associated with boys versus girls.

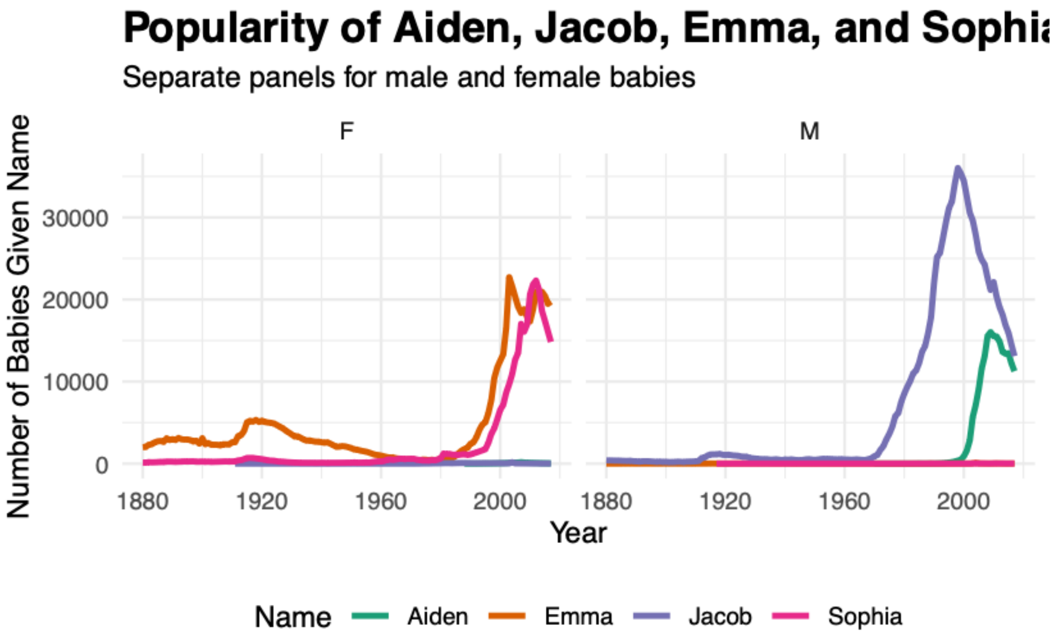

Figure 1: Popularity of Aiden, Jacob, Emma, and Sophia over time, separated by sex.

This visualization shows that **Emma** and **Sophia** have dramatic surges in popularity for girls in the late 1990s and 2000s, while **Jacob** and **Aiden** dominate among boys during overlapping but distinct periods. The faceting by sex makes it immediately clear that each name is overwhelmingly used for one gender, and the colored lines allow quick comparison of when each name reaches its peak. Overall, the plot highlights how naming fashions shift over time and how differently the same name can behave for boys versus girls.

Plotting a Mathematical Function (Box Problem)

For the classic box-from-a-sheet problem, I consider a rectangular sheet of paper that is **36 inches by 48 inches**. If we cut squares of side length (x) from each corner and fold up the sides, the resulting open-top box has base dimensions ((36 - 2x)) by ((48 - 2x)) and height (x). The volume function is

[V(x) = x(36 - 2x)(48 - 2x),]

where (x) is measured in inches and must be between 0 and 18 to keep the base dimensions positive.

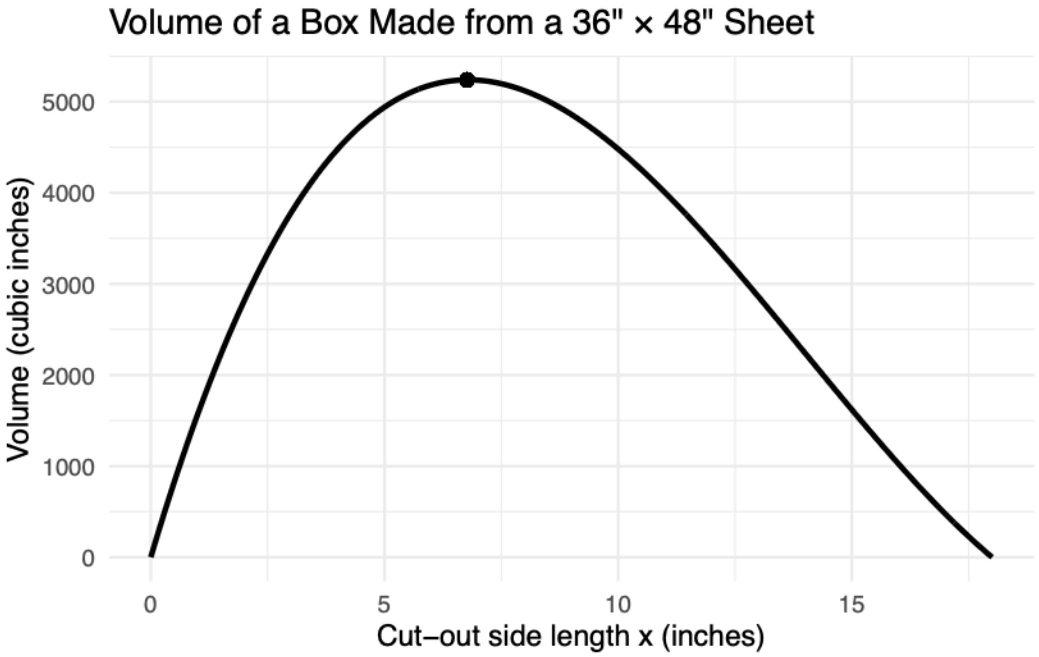

Figure 2: Volume of the 36” × 48” box as a function of the cut-out size x.

From the graph and numerical search, the volume rises from zero at (x = 0), reaches a clear maximum, and then declines back toward zero as the cut-out becomes so large that the base shrinks. The maximizing cut size is approximately 6.77 inches, which yields a maximum volume of about 5240 cubic inches. So, the box built from a 36 by 48 sheet achieves its **maximum volume** when the corner squares have side length about **6.77 inches**, producing a volume of roughly **5240 cubic inches**.

What I Feel I’ve Learned So Far

Over the course of this class, I’ve learned how to move from raw data and vague questions to clean, reproducible analyses that actually answer something. Early on, I treated R code as a collection of individual commands; now I think more in terms of pipelines—loading, wrangling, visualizing, and interpreting as a connected workflow. Activities like the Armed Forces project taught me that careful reshaping and validation are just as important as the final plot. At the same time, the visualization projects with diamonds, penguins, and baby names helped me see how design choices—like transparency, faceting, and color palettes—directly affect how easy it is for someone else to understand the story I’m trying to tell.

I’ve also improved in commenting and organizing my code so that another person could reproduce my results from scratch. That has changed the way I think about my own work: instead of writing quick, one-off scripts, I’m more deliberate about structure, naming conventions, and documenting my decisions. Overall, I feel like I’ve moved from just “using R” to actually thinking like a data analyst who can justify and communicate each step of an analysis.

Code Appendix

Below I include clearly labelled code sections for all of the analyses in this document. This appendix is the only place where code appears in the rendered PDF; the main body focuses on results and interpretation.

Armed Forces Data Wrangling Code

```
library(tidyverse)
library(rvest)

## Load and Prepare Pay Grade/Rank Reference Data ----
pay_grade_url <- "https://neilhatfield.github.io/Stat184_PayGradeRanks.html"
pay_grade_page <- read_html(pay_grade_url)

tables <- html_table(pay_grade_page, fill = TRUE)
pay_grade_table <- tables[[1]]

# Fix duplicate column names
names(pay_grade_table) <- make.unique(names(pay_grade_table))

pay_grade_ranks_long <- pay_grade_table %>%
  rename(PayGrade = 1) %>%
  pivot_longer(
    cols = -PayGrade,
    names_to = "ServiceBranch",
    values_to = "Rank"
  ) %>%
  mutate(
    Category = case_when(
      str_starts(PayGrade, "E") ~ "Enlisted",
      str_starts(PayGrade, "W") ~ "Warrant Officer",
      str_starts(PayGrade, "O") ~ "Officer",
      TRUE ~ NA_character_
    ),
    Rank = na_if(Rank, "--")
  ) %>%
  filter(ServiceBranch != "Coast Guard")

## Load Armed Forces Data ----
armed_forces_raw <- tribble(
  ~PayGrade, ~Army.Male, ~Army.Female, ~Navy.Male, ~Navy.Female, ~MarineCorps.Male, ~MarineCorp
  "E1", 7429, 1326, 8903, 3434, 7849, 655, 8537, 1933, 179, 38,
  "E2", 22338, 4336, 17504, 5833, 15034, 1684, 7343, 2019, 186, 41,
  "E3", 43775, 10229, 25436, 9103, 35239, 4174, 37324, 10369, 1015, 194,
  "E4", 79234, 15143, 33859, 9959, 28519, 2961, 53185, 15055, 541, 179,
  "E5", 54803, 10954, 58142, 16169, 22262, 2670, 40614, 10762, 859, 173,
  "E6", 49502, 7363, 45833, 9950, 12225, 1529, 31400, 6679, 853, 147,
  "E7", 30264, 4410, 19046, 3434, 7720, 747, 18309, 4807, 535, 114,
  "E8", 9482, 1472, 6007, 850, 3495, 293, 3876, 1221, 112, 25,
  "E9", 2865, 394, 2574, 368, 1515, 82, 1903, 523, 47, 16,
  "W1", 3727, 460, 44, 4, 494, 44, 27, 1, NA, NA,
  "W2", 6024, 692, 641, 91, 725, 53, 33, 1, NA, NA,
  "W3", 2794, 346, 744, 115, 518, 32, 0, 0, NA, NA,
  "W4", 1378, 137, 432, 41, 265, 12, 0, 0, NA, NA,
  "W5", 494, 43, 69, 6, 104, 3, 0, 0, NA, NA,
  "O1", 7122, 2400, 5497, 1766, 2412, 366, 5048, 1985, 412, 152,
  "O2", 9550, 3006, 5544, 1716, 3162, 525, 5045, 2037, 437, 155,
  "O3", 20986, 6053, 14480, 4830, 5385, 707, 15715, 5485, 997, 280,
  "O4", 12350, 3044, 7983, 2306, 3637, 338, 9682, 3440, 941, 209,
  "O5", 6939, 1531, 5525, 1151, 1830, 137, 7373, 1890, 657, 124,
  "O6", 3161, 452, 2644, 452, 656, 54, 2663, 569, 206, 42,
  "O7", 100, 18, 101, 5, 36, 2, 99, 18, 11, 2,
  "O8", 80, 8, 62, 6, 28, 2, 63, 6, 10, 0,
  "O9", 46, 5, 32, 2, 17, 1, 30, 7, 4, 1,
  "O10", 11, 0, 8, 0, 3, 0, 11, 0, 3, 0
)

group_level_data <- armed_forces_raw %>%
  pivot_longer(
    cols = -PayGrade,
    names_to = "Branch_Sex",
    values_to = "Count"
  ) %>%
  mutate(
    Sex = str_extract(Branch_Sex, "(Male|Female)$"),
    ServiceBranch = str_remove(Branch_Sex, "\\.(?:Male|Female)$")
  ) %>%
  mutate(
    ServiceBranch = case_when(
      ServiceBranch == "MarineCorps" ~ "Marine Corps",
      ServiceBranch == "AirForce" ~ "Air Force",
      ServiceBranch == "SpaceForce" ~ "Space Force",
      TRUE ~ ServiceBranch
    )
  ) %>%
  select(-Branch_Sex) %>%
  filter(!is.na(Count)) %>%
  left_join(
    pay_grade_ranks_long,
    by = c("PayGrade", "ServiceBranch")
  ) %>%
```

Rank

```
  select(PayGrade, ServiceBranch, Sex, Count, Rank, Category) %>%
  arrange(PayGrade, ServiceBranch, Sex)

individual_level_data <- group_level_data %>%
  uncount(Count)
```

Frequency Table Code

```
library(gt)

army_enlisted <- individual_level_data %>%
  filter(ServiceBranch == "Army", Category == "Enlisted")

army_enlisted_table <- army_enlisted %>%
  count(Rank, Sex) %>%
  pivot_wider(
    names_from = Sex,
    values_from = n,
    values_fill = 0
  ) %>%
  arrange(match(Rank, pay_grade_ranks_long$Rank))

army_enlisted_table %>%
  gt()
```

Baby Names Code

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

BabyNames <- babynames %>%
  rename(count = n)

my_names <- c("Aiden", "Jacob", "Emma", "Sophia")

selected_names <- BabyNames %>%
  filter(name %in% my_names) %>%
  group_by(year, name, sex) %>%
```

```r
  summarise(total_count = sum(count), .groups = "drop")

ggplot(selected_names, aes(x = year, y = total_count, color = name)) +
  geom_line(linewidth = 1.1) +
  facet_wrap(~ sex) +
  labs(
    title = "Popularity of Aiden, Jacob, Emma, and Sophia Over Time",
    subtitle = "Separated by Sex to Show How Popularity Trends Differ",
    x = "Year",
    y = "Number of Babies Given Name",
    color = "Name"
  ) +
  scale_color_brewer(palette = "Dark2") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    legend.position = "bottom"
  )
```

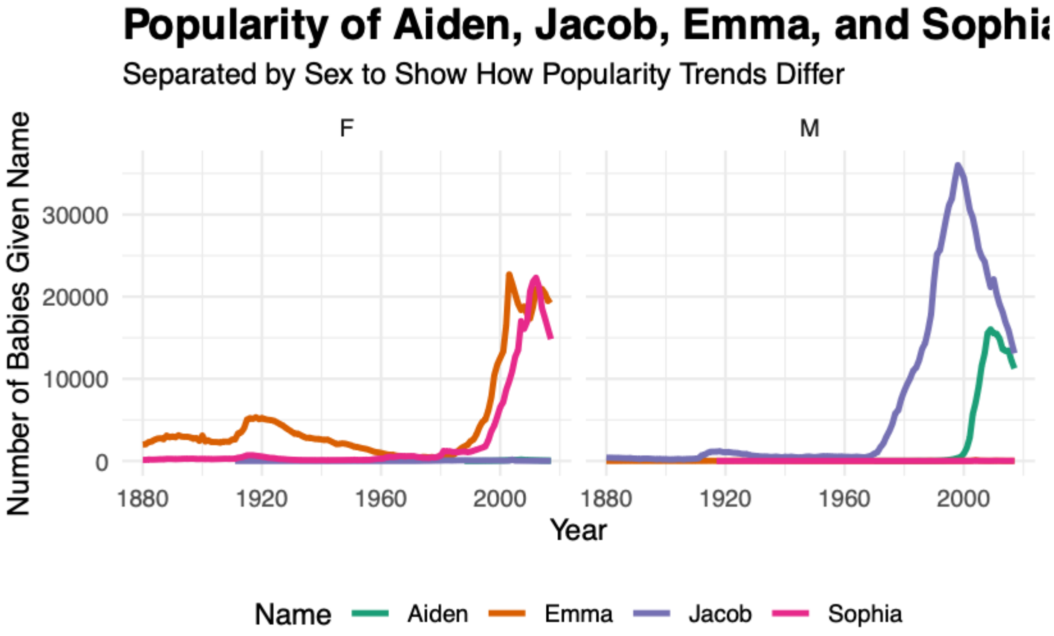

Figure 3: Popularity of Aiden, Jacob, Emma, and Sophia over time, separated by sex.

Box Problem Code

```r
library(ggplot2)
library(tibble)
library(dplyr)

box_volume <- function(x) {
  x * (36 - 2 * x) * (48 - 2 * x)
}

x_vals <- seq(0, 18, length.out = 400)
volume_df <- tibble(x = x_vals, volume = box_volume(x_vals))

opt_index <- which.max(box_volume(x_vals))
x_max <- x_vals[opt_index]
v_max <- box_volume(x_max)

ggplot(volume_df, aes(x = x, y = volume)) +
  geom_line(linewidth = 1) +
  geom_point(aes(x = x_max, y = v_max), size = 2) +
  labs(
    title = "Volume of a Box Made from a 36\" × 48\" Sheet",
    x = "Cut-out side length x (inches)",
    y = "Volume (cubic inches)"
  ) +
  theme_minimal()
```

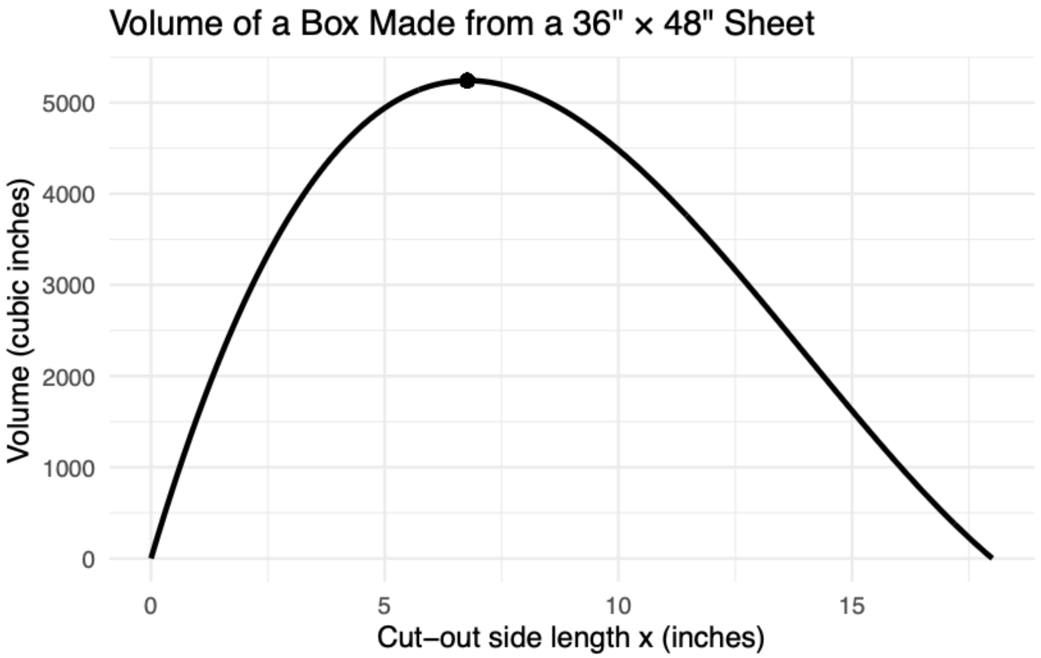

Figure 4: Volume of the 36” × 48” box as a function of the cut-out size x.