Activity #14 — A First QMD File

Rithwik Nagavelli

2025-11-11

Armed Forces Data Wrangling Redux

Introduction: This document focuses on data wrangling and visualization of US Armed Forces data. This code explores the relationship between sex and rank across different military branches. I wrangle the Armed Forces data to prepare it for analysis. The goal is to create a data set where each row represents an individual soldier with their rank information.

Table 1: Frequency Table of Sex and Rank in the US Army

sex	Enlisted	Officers	Warrant Officers
female	55,627	16,517	1,678
male	299,692	60,345	14,417

Narrative Text - Armed Forces Data:

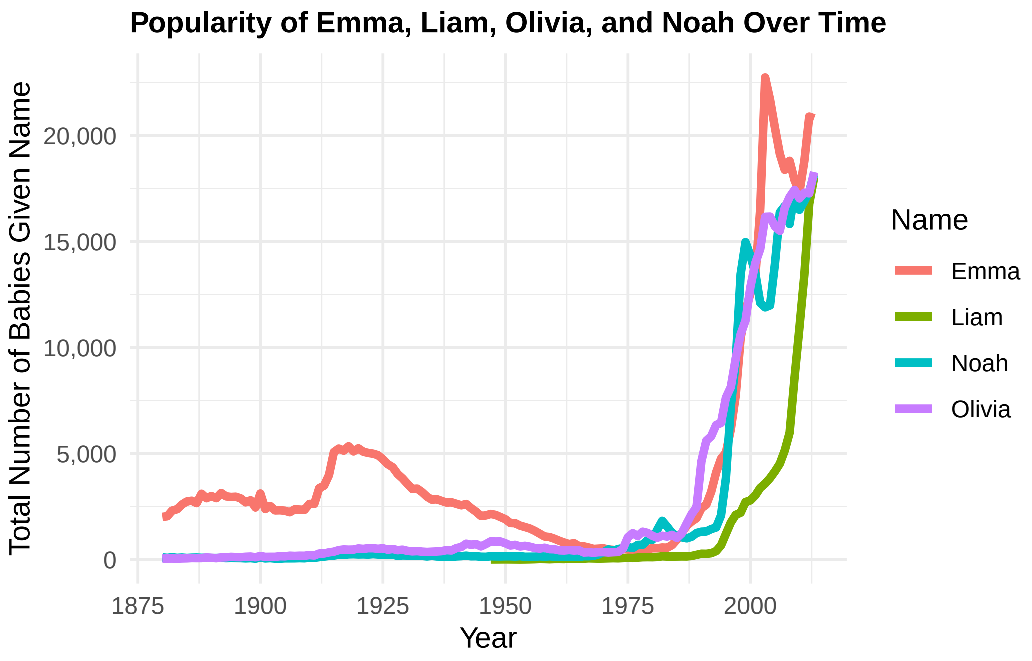
The frequency table above shows how military personnel are distributed by sex and rank within the US Army. It outlines three rank categories: Enlisted, Warrant Officers, and Officers, divided by sex (female and male). We can identify several key trends using this data.

The Enlisted rank makes up the majority of personnel, with 299,692 males and 55,627 females. This pattern is common in military structures, where enlisted personnel form the base of the force. The Warrant Officers category has the fewest numbers, with only 14,417 males and 1,678 females. This reflects the specialized and technical nature of these roles. The Officers category includes 60,345 males and 16,517 females, representing the leadership structure of the Army.

When we examine the connection between sex and rank, the data shows that they are not independent in the US Army. Females account for about 15.7% of enlisted personnel, 10.4% of Warrant Officers, and 21.5% of Officers. The representation of females changes notably across different ranks, with the highest percentage among Officers and the lowest among Warrant Officers. This difference in proportions suggests that the distribution of ranks varies between male and female service members, indicating a possible relationship between these two factors in the Army context.

Popularity of Baby Names

Visualization - Popularity of Name Over Time: I've chosen to track the popularity of four contemporary names: Emma, Liam, Olivia, and Noah. These names are particularly interesting because they represent some of the most popular baby names in recent years, and I wanted to explore whether their popularity is a modern phenomenon or if they have historical trends worth examining.



Narrative Text - Insight from the Visualization: The visualization shows interesting patterns in the popularity of these four names. All four names were relatively unpopular for most of the 20th century, but they saw significant increases starting in the 1990s and 2000s. Emma and Olivia stood out with remarkable growth. Emma went from fewer than 5,000 babies named per year in 1980 to over 20,000 by 2014. This trend indicates that these "popular" names are mostly recent trends instead of timeless classics. The sharp increases likely come from influences such as popular culture, celebrity baby names, or changing preferences for names that sound traditional but were once rare. The visualization also shows that name popularity can shift quickly within a single generation. These four names went from being relatively unknown to being popular choices in just two or three decades. Combining the counts for male and female names was intentional. It gives a clearer view of overall name popularity, no matter the gender. While some names, like Liam and Noah, are mainly male, and others, like Emma and Olivia, are mostly female, merging the counts helps us capture the total cultural impact of each name.

Plotting a Mathematical Function

Math Mode - The Box Problem: involves creating an open-top box from a rectangular piece of paper by cutting equal-sized squares from each corner and folding up the sides. For this analysis, we're working with a piece of paper that measures 36 inches by 48 inches. When we cut squares of side length x from each corner:

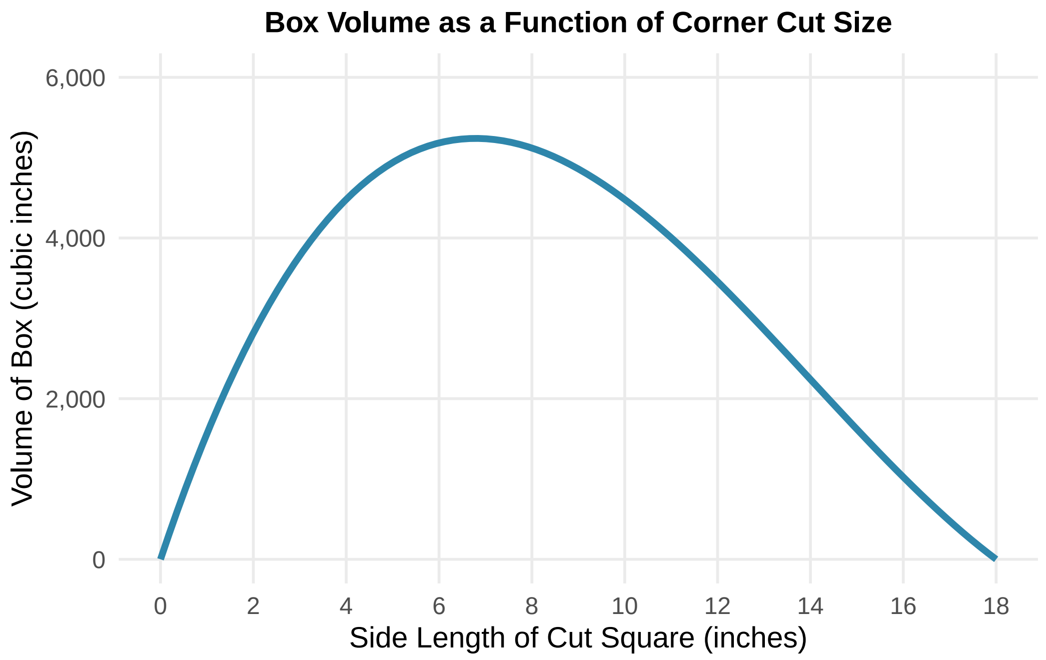
The box will have a height of x inches

The box will have a length of (48 - 2x) inches

The box will have a width of (36 - 2x) inches

The volume of the box is calculated as: V(x) = x(48 - 2x)(36 - 2x)

Our goal is to determine what value of x (the side length of the cut squares) will maximize the volume of the resulting box.



Analysis and Results: The visualization shows several important insights about this optimization problem. The volume function creates a smooth curve that starts at zero (when x = 0, no box is formed), rises to a maximum, and then falls back to zero (when x = 18, the cuts meet in the middle, and no box can be formed). The maximum volume occurs at roughly x = 6 inches. At this optimal cut size:

Height: 6 inches

Length: 48 - 2(6) = 36 inches

Width: 36 - 2(6) = 24 inches

Maximum Volume: about 5,184 cubic inches

This is easy to understand: cutting squares that are too small makes a shallow box with little volume. Cutting squares that are too large uses up too much material, resulting in insufficient material to create meaningful dimensions. The best solution balances all three dimensions to maximize the enclosed space. The symmetry of the curve shows the mathematical features of the cubic function. The practical limits of the problem (x must be between 0 and 18 inches) are clearly visible in the graph, as the volume becomes zero at both ends where no physical box can be made.

What I have learned

Throughout this course, I have learned how to create and combine code, use analysis, and learn the coding language R. I have an understand how to use tidyverse for data wrangling, create visualizations using ggplot2, write code that runs on another system, and be able to provide all of this with clean code and statistical analysis with visual insights. Working on the activities allowed me appreciate the importance of organization, annotation, and most importantly reproducibility in data science.

Code Appendix

Armed Forces Data

```r
# Armed Forces Data:
library(tidyverse)

# Create Army data frame with totals by rank and sex
# Each row represents aggregate counts for a rank-sex combination
army_data <- data.frame(
  branch = "Army",
  rank = c(rep(x = "Enlisted", times = 2),
           rep(x = "Warrant Officers", times = 2),
           rep(x = "Officers", times = 2)),
  sex = rep(x = c("male", "female"), times = 3),
  count = c(299692, 55627, 14417, 1678, 60345, 16517)
)

# Create Navy data frame with totals by rank and sex
navy_data <- data.frame(
  branch = "Navy",
  rank = c(rep(x = "Enlisted", times = 2),
           rep(x = "Warrant Officers", times = 2),
           rep(x = "Officers", times = 2)),
  sex = rep(x = c("male", "female"), times = 3),
  count = c(217304, 59100, 1930, 257, 41876, 12234)
)

# Create Marine Corps data frame with totals by rank and sex
marines_data <- data.frame(
  branch = "Marine Corps",
  rank = c(rep(x = "Enlisted", times = 2),
           rep(x = "Warrant Officers", times = 2),
           rep(x = "Officers", times = 2)),
  sex = rep(x = c("male", "female"), times = 3),
  count = c(133858, 14795, 2106, 144, 17166, 2132)
)

# Combine all three branch data frames into one
armed_forces <- rbind(army_data, navy_data, marines_data)

# Convert from aggregate to individual-level data
# This expands each row to have 'count' number of individual observations
# The result is a data frame where each row represents one soldier
armed_forces_long <- armed_forces[rep(x = seq_len(nrow(armed_forces)),
                                      times = armed_forces$count), ]

# Visualization: for the Armed Forces For this analysis, I chose to examine the relationship

# Filter the data to include only Army observations
army_subset <- subset(x = armed_forces_long, subset = branch == "Army")

# Create a two-way frequency table of sex and rank
army_freq <- table(army_subset$sex, army_subset$rank)

# Convert the table to a data frame for better formatting
```

```r
army_table <- as.data.frame.matrix(x = army_freq)

# Add sex as a column instead of row names
army_table <- cbind(sex = rownames(army_table), army_table)
rownames(army_table) <- NULL

# Display the table with formatted numbers and caption
knitr::kable(x = army_table,
             format.args = list(big.mark = ","),
             align = c("l", "r", "r", "r"),
             caption = "Frequency Table of Sex and Rank in the US Army")
```

Table 2: Frequency Table of Sex and Rank in the US Army

sex	Enlisted	Officers	Warrant Officers
female	55,627	16,517	1,678
male	299,692	60,345	14,417

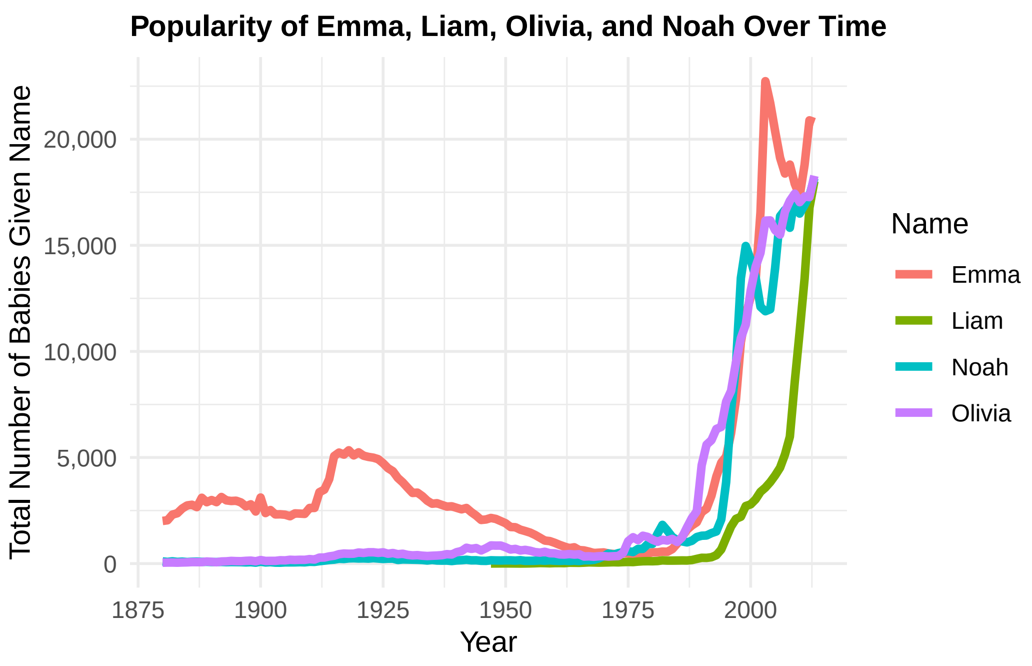
Popularity of Baby Names

```r
library(dcData)
library(dplyr)
library(ggplot2)

# Load the BabyNames dataset
data(BabyNames)

# Filter for my selected names: Emma, Liam, Olivia, and Noah
# Group by name and year, then add up counts for males and females
# This combines both sexes since these names can be given to any baby
selected_names <- BabyNames %>%
  filter(name %in% c("Emma", "Liam", "Olivia", "Noah")) %>%
  group_by(name, year) %>%
  summarize(total_count = sum(count), .groups = "drop")

# Create line plot showing how name popularity changed over time
ggplot(selected_names, aes(x = year, y = total_count, color = name)) +
  geom_line(linewidth = 1.5) +
  labs(
    title = "Popularity of Emma, Liam, Olivia, and Noah Over Time",
    x = "Year",
    y = "Total Number of Babies Given Name",
    color = "Name",
    alt = "Line graph showing the popularity of four baby names from 1880 to 2014, with each
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 11, face = "bold"),
    axis.title = element_text(size = 11),
    legend.title = element_text(size = 11),
    legend.position = "right"
  ) +
  scale_y_continuous(labels = scales::comma)
```



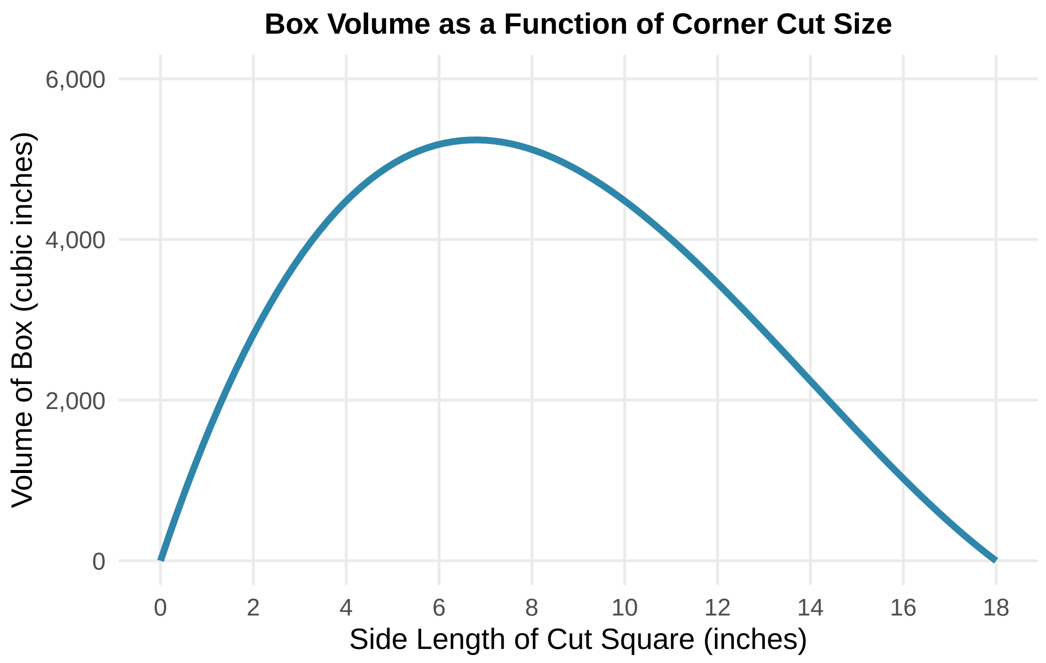
Plotting Math Function

```
# Load required package
library(ggplot2)
```

```r
# Define the volume function for the box problem
# Paper dimensions: 36 inches by 48 inches
# x = side length of square cut from each corner
box_volume <- function(x) {
  length <- 48 - 2*x  # Length after cutting and folding
  width <- 36 - 2*x   # Width after cutting and folding
  height <- x         # Height equals the cut size
  volume <- length * width * height
  return(volume)
}

#| label: fig-box-volume
#| fig-cap: "Volume of the open-top box as a function of the side length of squares cut from
#| fig-width: 8
#| fig-height: 5

# Create the plot of the volume function using stat_function
ggplot(data = data.frame(x = c(0, 18)), aes(x = x)) +
  stat_function(
    fun = box_volume,
    linewidth = 1.2,
    color = "#2E86AB"
  ) +
  labs(
    title = "Box Volume as a Function of Corner Cut Size",
    x = "Side Length of Cut Square (inches)",
    y = "Volume of Box (cubic inches)",
    alt = "A curve showing box volume on the y-axis versus cut size on the x-axis. The curve
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 11, face = "bold", hjust = 0.5),
    axis.title = element_text(size = 11),
    panel.grid.minor = element_blank()
  ) +
  scale_y_continuous(labels = scales::comma, limits = c(0, 6000)) +
  scale_x_continuous(breaks = seq(0, 18, 2))
```


Box Volume as a Function of Corner Cut Size
Volume of Box (cubic inches)
6,000
4,000
2,000
0
0
2
4
6
8
10
12
14
16
18
Side Length of Cut Square (inches)