Data Wrangling and Visualization

Tim Damasco

2025-11-11

Armed Forces Data Wrangling

This section demonstrates the data wrangling process for the U.S. Armed Forces personnel data, creating both group and individual level data frames.

Data Preparation

The data wrangling process transforms the original wide format Armed Forces data into a tidy format suitable for analysis. The raw data contains personnel counts organized by pay grade, branch, and sex.

Results

The final datasets include:

- **groups_df**: A group level data frame with personnel counts by pay grade, branch, sex, and rank
- **individuals_df**: An individual level data frame where each row represents one soldier with their associated characteristics

Frequency Table Analysis

To explore the relationship between sex and rank in the U.S. Armed Forces, I created a two-way frequency table focusing on Army enlisted personnel (pay grades E-1 through E-9). This subset was chosen because it represents a substantial portion of the military with sufficient observations across all rank levels and both sexes.

Table 1: Two-way frequency table showing the distribution of Army enlisted personnel by rank and sex

	F	M
Corporal OR Specialist	15143	79234
First Sergeant OR Master Sergeant	1472	9482
Private	5662	29767

	F	M
Private First Class	10229	43775
Sergeant	10954	54803
Sergeant First Class	4410	30264
Sergeant Major OR Command Sergeant Major	394	2865
Staff Sergeant	7363	49502

Analysis of the Frequency Table

The frequency table displays eight enlisted ranks in rows (from Private to Staff Sergeant) with counts separated by sex in two columns (F for female, M for male). Several patterns emerge from examining these data. First, males substantially outnumber females across all enlisted ranks, with the male-to-female ratio appearing relatively consistent throughout the rank structure. For example, at the Private rank, there are 29,767 males compared to 5,662 females (approximately a 5:1 ratio), while at the Sergeant rank, there are 54,803 males compared to 10,954 females (approximately a 5:1 ratio as well).

Second, the distribution across ranks shows that mid-level enlisted positions contain the largest numbers of personnel. The Sergeant rank (E-5) has the highest total count with 65,757 soldiers, while the highest enlisted ranks like Sergeant Major/Command Sergeant Major (E-9) have the smallest counts with only 3,259 total soldiers. This pyramid structure reflects the nature of military organization, where fewer positions exist at higher ranks.

Regarding the independence of sex and rank, the data suggest these variables may be approximately independent within the Army enlisted population. The male to female ratio remains relatively stable across different rank levels, rather than showing dramatic shifts that would indicate strong dependence. If sex and rank were strongly dependent, we would see substantially different ratios at different ranks. This would mean that women were disproportionately concentrated at lower ranks or if advancement rates differed dramatically by sex.

Popularity of Baby Names

Name Selection and Rationale

For this analysis, I selected four names to track over time: Tim, Connor, Mildred, and Sophia. These names were chosen to represent diverse temporal popularity patterns. Tim and Mildred represent names with historical peaks that have declined in recent decades, while Connor and Sophia demonstrate more contemporary popularity trends. Additionally, these names show interesting gender-specific patterns, with some names being predominantly used for one sex while others show more balanced usage.

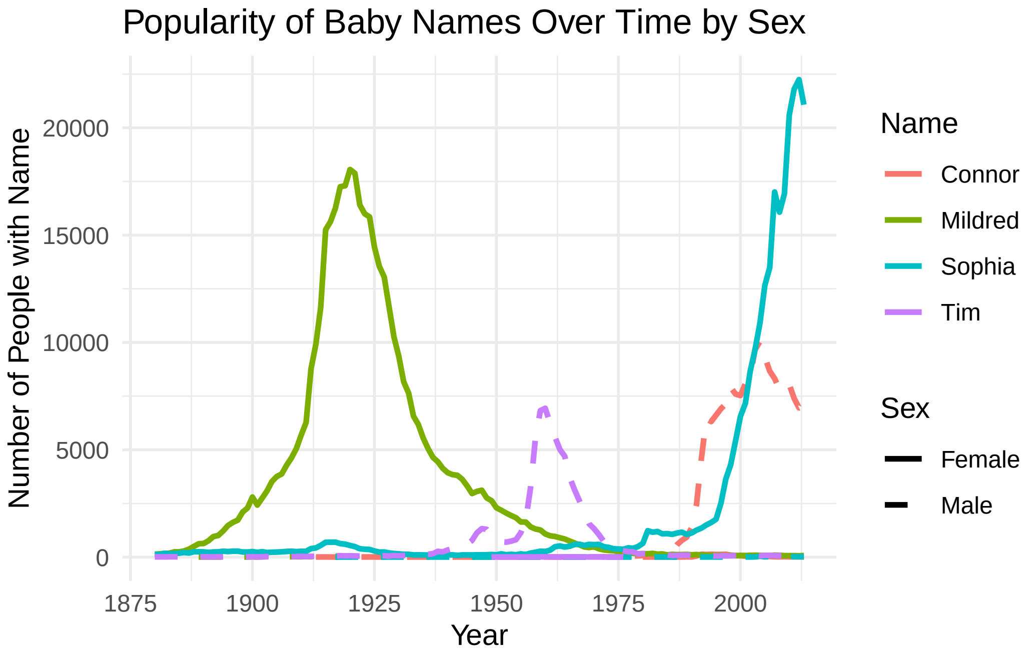

Figure 1: Time series showing the popularity of selected baby names from 1880 to 2013, with line type indicating sex

Interpretation of Trends

This visualization shows the popularity of the names Tim, Connor, Mildred, and Sophia from the years of 1880 to 2015. There are separate lines distinguishing between sexes, with dashed lines being males and solid lines being females. The graph uses different colors for each name and showcases that the sex is a factor in the popularity of a name. For example, the name Mildred shown in the green line shows exclusively female naming and peaked around the early 20th century. Comparatively, the name Tim peaked in the 1960s and is exclusively a male name. Sex matters significantly when talking about the popularity of a name as these four names show virtually no sex-name sharing patterns.

The visualization employs both color and line type to encode information effectively. Color distinguishes between the four different names, while line type (solid versus dashed) indicates sex, allowing viewers who are color blind to still distinguish between male and female usage patterns. This dual encoding ensures the graphic remains accessible while clearly communicating temporal trends in naming patterns across different genders and generations.

Plotting a Mathematical Function

The Box Problem

The classic box problem involves cutting equal squares from each corner of a rectangular piece of paper and folding up the sides to create an open-top box. For this analysis, I used a piece of paper measuring 36 inches by 48 inches. The objective is to determine what size square cut (x) maximizes the volume of the resulting box.

Mathematical Function

The volume function for this box can be derived as follows: - Length of box: 48 - 2x (original length minus two cuts) - Width of box: 36 - 2x (original width minus two cuts)
- Height of box: x (the size of the square cut) - Volume: V(x) = (48 - 2x)(36 - 2x)(x)

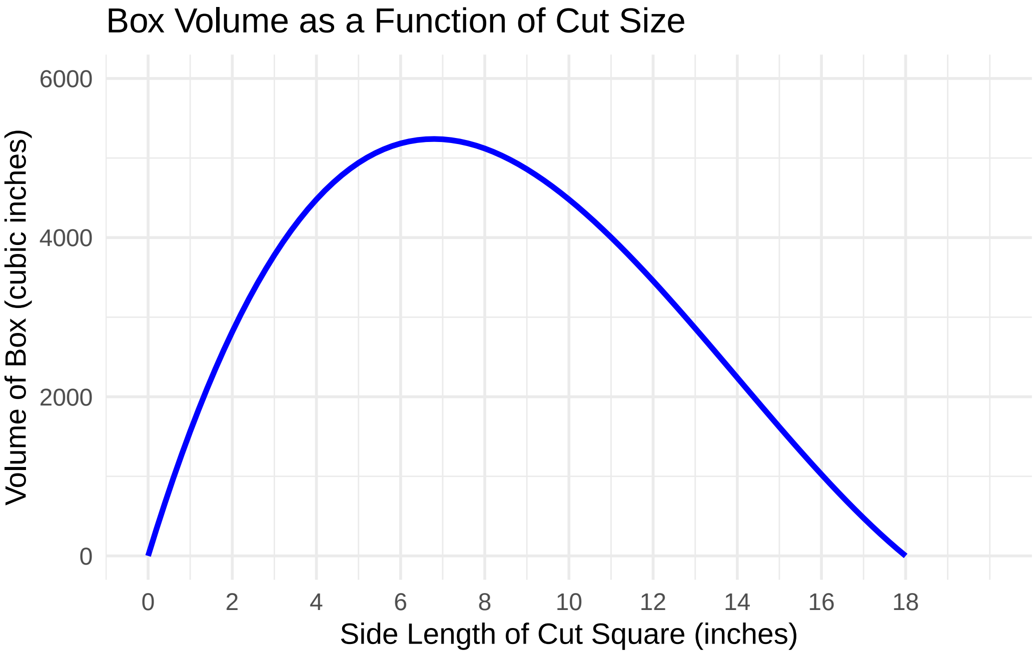

Figure 2: Graph showing the volume of a box as a function of the side length of cut squares, for a 36 inch by 48 inch piece of paper

Analysis and Optimal Cut Size

The graph reveals that the box volume follows a smooth parabolic relationship with the cut size. Starting from zero volume when no cuts are made (x = 0), the volume increases as the cut size grows, reaching a maximum at approximately **x = 6.86 inches**, where the volume peaks at **5,239.82 cubic inches**. Beyond this optimal point, the volume decreases as the cuts become too large, eventually returning to zero when x = 18 inches (at which point the width would be reduced to zero).

Answering the Original Questions: - **Maximum volume:** 5,239.82 cubic inches - **Optimal cut size:** 6.86 inches

This mathematical relationship demonstrates an optimization problem where the maximum occurs at a point along our domain. The optimal cut size of approximately 6.86 inches balances the competing effects of box height (which increases with larger cuts) and base dimensions (which decrease with larger cuts). We can observe from the graph that cuts smaller or larger than the optimal value result in progressively smaller volumes.

###What I have learned so far in stat 184

Throughout this course, I have developed a comprehensive understanding of workflows using R and the tidyverse ecosystem. One of the most valuable skills I have acquired is data wrangling, which I demonstrated in the Armed Forces analysis by transforming wide-format data into tidy long-format data using `pivot_longer()`, filtering rows with `filter()` and `str_detect()`, and joining multiple data sources with `left_join()`. This process taught me that that careful manipulation of the data is essential before any meaningful analysis can occur.

I've also gained proficiency in web scraping techniques, specifically using the `rvest` package to extract structured data from HTML tables. In the Armed Forces project, I scraped rank information from a webpage using `read_html()` and `html_table()`, which allowed me to enrich my dataset with contextual information that was not available in the original CSV file.

Additionally, I have become comfortable with the data manipulation through dplyr's verb functions. Functions like `mutate()`, `select()`, `arrange()`, and `uncount()` have been regurlary used, and I now understand how to chain these operations together using the pipe operator (`%>%`) to create clear, readable data processing pipelines.

Finally, I have developed skills in data visualization using ggplot2, where I have learned to create quality graphics that effectively communicate insights. Through projects like the baby names time series and the box volume optimization, I have learned that effective visualizations require thoughtful choices about aesthetics like color, line type, size, proper labeling, and accessibility considerations. I now understand that visualization is not just about making pretty charts, but about designing visual arguments that help readers understand patterns and relationships in data.

Code Appendix

Armed Forces Data Wrangling and Frequency Table Creation

```r
# Load packages
library(tidyverse)
library(janitor)
library(rvest)
library(knitr)

# Read and clean the Armed Forces data
military <- read_csv(
  "US_Armed_Forces_(6_2025) - Sheet1 (1).csv",
  skip = 2,
  col_types = cols(.default = "c")
) %>%
  # Select and name only the pay grade + male/female columns
  rename(
    PayGrade   = 1,
    Army_M     = 2,  Army_F     = 3,
    Navy_M     = 5,  Navy_F     = 6,
    Marines_M  = 8,  Marines_F  = 9,
    AirForce_M = 11, AirForce_F = 12,
    Space_M    = 14, Space_F    = 15
  ) %>%
  # Keep only true pay grades (drop totals / blank lines)
  filter(str_detect(PayGrade, "^[EWO]")) %>%
  # Turn counts into numeric, remove commas and N/A
  mutate(
    across(
      -PayGrade,
      ~ as.numeric(str_remove_all(., ",|N/A\\*"))
    )
  )

# Group-level tidy data (case = group of soldiers)
groups_df <- military %>%
  pivot_longer(
    cols = -PayGrade,
    names_to = c("Branch", "Sex"),
    names_sep = "_",
    values_to = "Count"
  ) %>%
  drop_na(Count)

# Get pay grade and rank titles from Pay Grade and Ranks webpage
pg_url <- "https://neilhatfield.github.io/Stat184_PayGradeRanks.html"

ranks_raw <- pg_url %>%
  read_html() %>%
  html_table(fill = TRUE) %>%
  .[[1]] %>%
  clean_names() %>%
  rename(
    PayGrade   = pay_grade,
    Army       = ranks_by_branch_of_service,
    Navy       = ranks_by_branch_of_service_2,
    Marines    = ranks_by_branch_of_service_3,
    AirForce   = ranks_by_branch_of_service_4,
    Space      = ranks_by_branch_of_service_5,
    CoastGuard = ranks_by_branch_of_service_6
  ) %>%
  select(-x) %>%
  filter(
    !is.na(PayGrade),
    PayGrade != "Pay Grade",
    !str_starts(PayGrade, "Note")
  )

ranks_long <- ranks_raw %>%
  pivot_longer(
    cols = -PayGrade,
    names_to = "Branch",
    values_to = "RankTitle"
  ) %>%
  mutate(
    RankTitle = ifelse(RankTitle == "--", NA, RankTitle)
  )

# Join rank titles onto group level data
groups_df <- groups_df %>%
  left_join(ranks_long, by = c("PayGrade", "Branch"))

# groups_df structure:
# PayGrade, Branch, Sex, Count, RankTitle
# (case = group of soldiers)

# Individual-level data (case = individual soldier)
individuals_df <- groups_df %>%
  filter(!is.na(Count), Count > 0) %>%
  uncount(weights = Count)

# individuals_df structure:
# PayGrade, Branch, Sex, RankTitle
# (one row per soldier)

# Two-way frequency table (Sex x Rank) for Army Enlisted personnel
army_enlisted <- individuals_df %>%
  filter(Branch == "Army",
         str_detect(PayGrade, "^E"))

army_enlisted_table <- table(
  Rank = army_enlisted$RankTitle,
  Sex  = army_enlisted$Sex
)
```

Baby Names Visualization

```
# Alt text that is accessible-"Line graph showing the popularity of four baby
#names (Tim, Connor, Mildred, and Sophia) from 1880 to 2013. The x-axis
#represents years, and the #y-axis shows the count of babies given each name."
```

```r
# Load required package
library(dcData)

# Load the BabyNames data
data(BabyNames)

# Filter for selected names
selected_names <- BabyNames %>%
  filter(name %in% c("Tim", "Connor", "Mildred", "Sophia"))

# Create time series plot with sex indicated by line type
ggplot(data = selected_names, aes(x = year, y = count, color = name, linetype = sex)) +
  geom_line(size = 1) +
  scale_linetype_manual(values = c("F" = "solid", "M" = "dashed"),
                        labels = c("F" = "Female", "M" = "Male"),
                        name = "Sex") +
  labs(title = "Popularity of Baby Names Over Time by Sex",
       x = "Year",
       y = "Number of People with Name",
       color = "Name") +
  theme_minimal() +
  theme(legend.position = "right")
```

Box Problem Function and Visualization

```r
# Alternative text for accessibility:
# A line graph showing box volume on the y-axis versus cut size on the x-axis.
# The curve starts at zero, increases to a maximum around x equals 6 inches where
# volume is approximately 3456 cubic inches, then decreases back toward zero as
# x approaches 18 inches.

# Define the volume function for a 36 inch by 48 inch piece of paper
calculate_volume <- function(x) {
  length <- 48 - 2*x  # Length after cutting squares from both ends
  width <- 36 - 2*x   # Width after cutting squares from both sides
  height <- x         # Height equals the side of the cut square
  volume <- length * width * height
  return(volume)
}

# Create sequence of x values from 0 to 18 inches
x_values <- seq(from = 0, to = 18, by = 0.01)

# Calculate corresponding y values (volumes)
y_values <- calculate_volume(x = x_values)
```

```r
# Create the plot using ggplot2 with stat_function
ggplot(data = data.frame(x = x_values), aes(x = x)) +
  stat_function(fun = calculate_volume, color = "blue", size = 1) +
  labs(title = "Box Volume as a Function of Cut Size",
       x = "Side Length of Cut Square (inches)",
       y = "Volume of Box (cubic inches)") +
  theme_minimal() +
  scale_x_continuous(limits = c(0, 20), breaks = seq(0, 18, 2)) +
  scale_y_continuous(limits = c(0, 6000))
```