Activity 14 - A First QMD File

Arhaan Keshwani

2025-11-10

1 Armed Forces Data Wrangling Redux

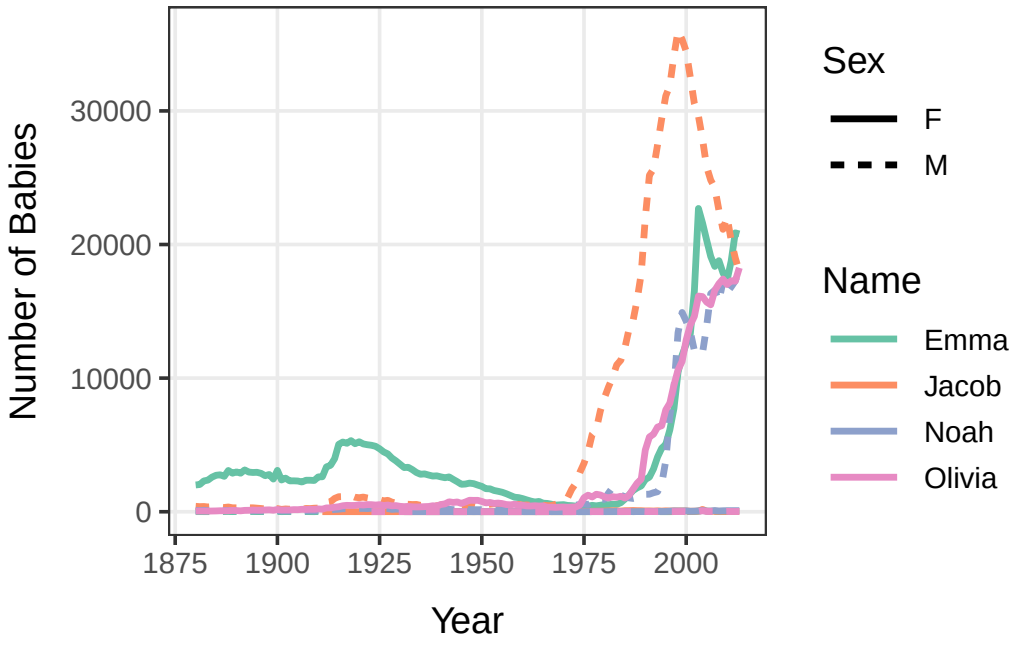
Table 1: Two-Way Frequency Table of Gender by Enlisted Rank in the U.S. Army

Army Enlisted Rank/Gender	Female	Male	Total
Corporal OR Specialist	15,143 (27.2%)	79,234 (26.4%)	94,377 (26.6%)
First Sergeant OR Master Sergeant	1,472 (2.6%)	9,482 (3.2%)	10,954 (3.1%)
Private	5,662 (10.2%)	29,767 (9.9%)	35,429 (10.0%)
Private First Class	10,229 (18.4%)	43,775 (14.6%)	54,004 (15.2%)
Sergeant	10,954 (19.7%)	54,803 (18.3%)	65,757 (18.5%)
Sergeant First Class	4,410 (7.9%)	30,264 (10.1%)	34,674 (9.8%)
Sergeant Major OR Command Sergeant Major	394 (0.7%)	2,865 (1.0%)	3,259 (0.9%)
Staff Sergeant	7,363 (13.2%)	49,502 (16.5%)	56,865 (16.0%)
Total	55,627 (100.0%)	299,692 (100.0%)	355,319 (100.0%)

The table summarizes the distribution of enlisted Army personnel by rank and gender, showing both the counts and percentages each gender contributes within a rank. For both males and females, the largest concentrations occur in the lower enlisted ranks, particularly Specialist/Corporal, Private First Class, and Sergeant. This shows that most enlisted soldiers have early to mid-career positions. However, the female proportion varies across ranks: females make up 27.2% of Specialists but only 7.9% of Sergeants First Class and less than 1% of Sergeant Majors. These shifts in percentages show that sex and rank are not independent among enlisted Army personnel. Instead, the chance of having higher enlisted ranks looks like it differs between males and females, with female representation decreasing at more senior ranks.

2 Popularity of Baby Names

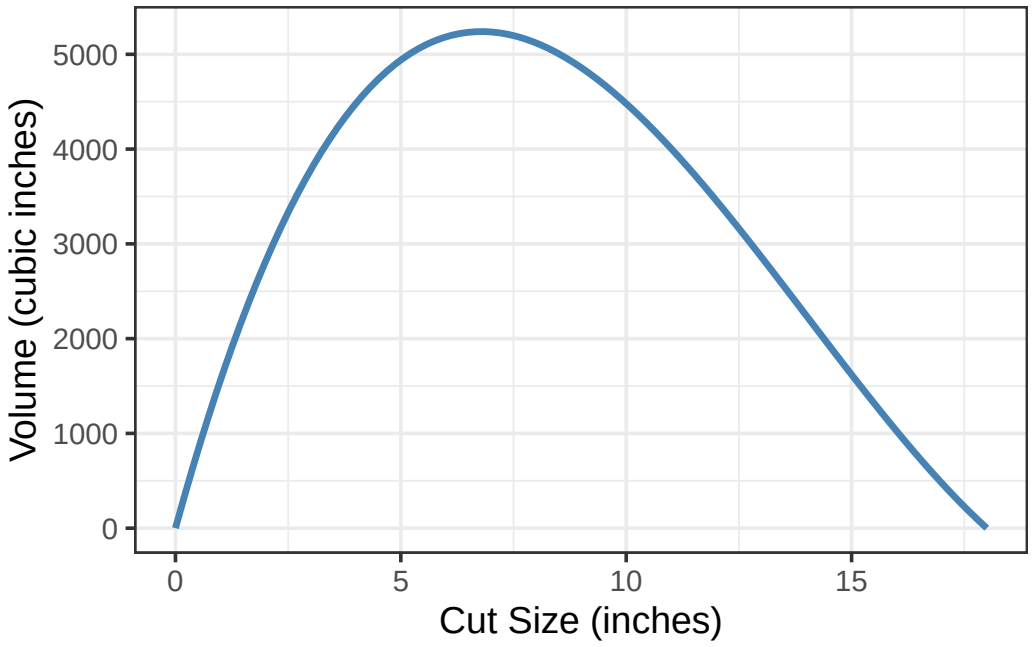
Figure 1: Popularity of Selected Baby Names Over Time.



The time series visualization displays the popularity of the names Emma, Olivia, Jacob, and Noah in the United States from the late 1800s through the early 2000s, with solid and dashed lines distinguishing female and male babies. These names were selected since they represent several of the most culturally recognizable and historically popular names in the U.S., and they also allow for a clear comparison between traditionally male and female naming patterns. From the plot, we can see that Jacob and Noah show dramatic rises in male popularity beginning around the 1980s and continuing into the early 2000s, with Jacob in particular reaching very high counts. In contrast, Emma and Olivia show steady long-term female usage that increases steeply in the late 1990s, showing modern naming trends. The clear separation of line types and colors helps us see that each name's trajectory is tied to its associated sex. There is no big crossover between male and female usage.

3 Plotting a Mathematical Function

Figure 2: Volume of an Open-Top Box Made From a 36 × 48 Inch Sheet.



The figure displays the volume of an open-top box created by cutting equal squares of side length x from each corner of a 36 × 48 inch sheet and folding up the sides. The curve shows how the box's volume changes as the cut size increases: volume first rises steeply, reaches a single peak, and then declines as the cuts become too large to leave a usable base. This pattern shows the trade-off in the box-making process, larger cuts create taller sides but also shrink the length and width of the base. From the graph, the maximum volume occurs at approximately x 6 inches, where the box reaches a peak of a little above 5,200 cubic inches, and then decreases afterward. Therefore, based on the visualization, the optimal cut size for constructing the largest possible open-top box from a 36 × 48 inch piece of paper is about 6 inches, producing a maximum volume of roughly 5,200 cubic inches.

4 What I Feel I Have Learned So Far

Throughout STAT 184, I've gained a much better understanding of how to plan, structure, and execute data analysis tasks effectively, especially through the PCIP method. Learning to pause and Plan, Construct, Implement, and Present has made my coding more intentional and my workflow more organized, and I've already noticed how this approach prevents confusion later in a project. I also learned to write and debug R code more confidently, particularly when creating graphs that communicate patterns clearly rather than just displaying data. Beyond the technical skills, I have learned the principles of good data visualization, specifically how choices in color, scale, layout, and labeling affect the way information is interpreted. These concepts have helped me think more

critically about how to design plots that are both accurate and understandable, and I know I'll continue applying these ideas in future data science work.

5 Code Appendix

```
# ------------------------------------------------
# Activity 8 Revision and Visualization
# ------------------------------------------------
# Import necessary libraries
library(googlesheets4)
library("dplyr")
library("tidyr")
library(rvest)
library(janitor)
library(knitr)
library(kableExtra)

# Import US Armed Forces Data from Google Sheet
gs4_deauth()
usaf_raw <- read_sheet(
  ss =
   ↪  'https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qbwb_E/edit?gid=597536282#gid=597536282'
)

# Import Pay Grade Ranks from HTML
rankTable <- read_html(x = "https://neilhatfield.github.io/Stat184_PayGradeRanks.html") %>%
  html_elements(css = "table") %>%
  html_table()

# Wrangle Ranks table to have 3 columns for pay grade, branch, and rank
ranks = rankTable[[1]]
names(ranks) <- c("Rank Type", "PayGrade", "Army", "Navy", "Marine Corps", "Air Force", "Space Force", "Coast
↪  Guard")
ranks_long <- ranks %>%
  select(-1) %>%
  slice(-c(1, 26)) %>%
  pivot_longer(
    cols = -PayGrade,
    names_to = "Branch",
    values_to = "Rank"
  )

# Wrangle US Armed Forces data to have each case be group of soldiers
usaf_group <- usaf_raw %>%
  select(-4, -7, -10, -13, -16, -17, -18, -19) %>%
  slice(-c(12, 18, 29, 30, 31)) %>%
  mutate(
    ...3 = ifelse(row_number() == 1, ...2, ...3),
    ...6 = ifelse(row_number() == 1, ...5, ...6),
    ...9 = ifelse(row_number() == 1, ...8, ...9),
    ...12 = ifelse(row_number() == 1, ...11, ...12),
    ...15 = ifelse(row_number() == 1, ...14, ...15)
  ) %>%
  {
    new_names <- c(
      "PayGrade",
      paste(
        as.character(unlist(.[1, -1])),
        as.character(unlist(.[2, -1])),
        sep = "/"
      )
    )
    setNames(., new_names)
  } %>%
  slice(-(1:2)) %>%
  pivot_longer(
    cols = -PayGrade,
    names_to = "Branch_Gender",
    values_to = "Count"
  ) %>%
```

```
  separate(Branch_Gender, into = c("Branch", "Gender"), sep = "/") %>%
  mutate(
    Count = as.character(Count),
    Count = na_if(Count, "N/A*"),
  ) %>%
  filter(!is.na(Count)) %>%
  mutate(
    Count = as.numeric(Count)
  ) %>%
  left_join(ranks_long, by = c("PayGrade", "Branch"))

# Wrangle US Armed Forces Grouped Data to have each case be an individual soldier
usaf_individual <- usaf_group %>%
  uncount(weights = Count)

# Filter to Enlisted Army personnel only
army_enlisted <- usaf_individual %>%
  filter(Branch == "Army", grepl("^E", PayGrade))

# Create two-way frequency table of Rank by Gender
army_enlisted_table_pct <- army_enlisted %>%
  tabyl(Rank, Gender) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_percentages("col") %>%
  adorn_pct_formatting(digits = 1) %>%
  adorn_title(
    placement = "combined",
    row_name = "Army Enlisted Rank",
    col_name = "Gender"
  )

formatNsArmyEnlisted <- attr(army_enlisted_table_pct, "core") %>%
  adorn_totals(where = c("row", "col")) %>%
  mutate(across(where(is.numeric), format, big.mark=","))

army_enlisted_table <- army_enlisted_table_pct %>%
  adorn_ns(position = "front", ns = formatNsArmyEnlisted)

# Display formatted table
army_enlisted_table %>%
  kable(
    format = "latex",
    caption = "Two-Way Frequency Table of Gender by Enlisted Rank in the U.S. Army",
    booktabs = TRUE,
    align = c("l", "c", "c", "c")
  ) %>%
  kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 16,
    latex_options = c("H", "scale_down")
  )
# ------------------------------------------------
# Activity 13: BabyNames Revision and Visualization
# ------------------------------------------------
# Load required packages
library(tidyverse)
library(dcData)  # Contains the BabyNames dataset

# Filter the BabyNames dataset for the selected names
selected_names <- c("Emma", "Olivia", "Jacob", "Noah")

babynames_filtered <- BabyNames %>%
  filter(name %in% selected_names)

# Create a time series plot for the selected names
ggplot(
  data = babynames_filtered,
  mapping = aes(
    x = year,
```

```
    y = count,
    color = name,
    linetype = sex
  )
) +
  # Use lines to show changes in popularity over time
  geom_line(linewidth = 1.2) +

  # Color-blind-friendly palette
  scale_color_brewer(palette = "Set2") +

  # Ensure linetype & color both encode info
  guides(
    color = guide_legend(
      title = "Name",
      order = 2
    ),
    linetype = guide_legend(
      title = "Sex",
      order = 1
    )
  ) +

  # Add title and axis labels
  labs(
    x = "Year",
    y = "Number of Babies"
  ) +

  # Theme that is clean and readable
  theme_bw(base_size = 14) +
  theme(
    plot.title = element_text(
      size = 16,
      hjust = 0.5
    ),
    axis.title.x = element_text(
      margin = margin(t = 10)
    ),
    axis.title.y = element_text(
      margin = margin(r = 10)
    ),
    legend.title = element_text(),
    legend.key.width = unit(1.1, "cm"),
    panel.grid.minor = element_blank()
  )
# ------------------------------------------------
# Activity 4: Box Problem Revision and Visualization
# ------------------------------------------------
# Load required package
library(ggplot2)

# Define function to compute volume of the box
volume_of_box <- function(x, paper_length = 48, paper_width = 36) {
  # l = length after cutting out squares
  length_after_cut <- paper_length - 2 * x

  # w = width after cutting out squares
  width_after_cut <- paper_width - 2 * x

  # h = height of the box, equal to cut size
  height_of_box <- x

  # Volume = l * w * h
  volume <- length_after_cut * width_after_cut * height_of_box
  return(volume)
}

# Domain of x: must be less than half the smaller dimension (36/2 = 18)
```

```r
x_range <- seq(from = 0, to = 18, by = 0.01)

# Create ggplot visualization using stat_function
ggplot(data.frame(x = x_range), aes(x = x)) +
  stat_function(
    fun = volume_of_box,
    args = list(paper_length = 48, paper_width = 36),
    linewidth = 1.2,
    color = "steelblue"
  ) +
  labs(
    x = "Cut Size (inches)",
    y = "Volume (cubic inches)"
  ) +
  theme_bw(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.title = element_text()
  )
```