Statistical Analysis of Demographic Trends in Quarto

Nguyen Ngo

2025-11-10

Introduction

This report contains statistical analyses across three sections: military personnel demographics, baby name popularity trends, and mathematical optimization.

1. **Armed Forces Data**
2. **Popularity of Baby Names**
3. **The Box Problem**
4. **Code Appendix**
5. **Self-Reflection**

Each section presents distinct analytical approaches and findings.

Armed Forces Data

The analysis focuses specifically on US Army enlisted personnel (pay grades E1-E9) to examine the relationship between sex and rank progression within a single military branch.

Analysis and Visualization

Table 1: 2-Way Frequency Table: Sex by Rank for US Army Enlisted Personnel

Rank	Female	Male
Private	5,662	29,767
Sergeant	10,954	54,803
Sergeant First Class	4,410	30,264
Staff Sergeant	7,363	49,502

Narrative Text

For this analysis, the focus is on a subset of enlisted ranks that contain non-missing, consistent rank names in the scraped dataset.

Table 1 displays the distribution of sex across selected enlisted ranks within the US Army. The table allows us to compare how male and female soldiers are distributed across ranks and to evaluate whether sex and rank appear to be independent in this subgroup.

The data shows that male soldiers significantly outnumber female soldiers at every rank shown; the gender gap widens as rank increases. For example, there are 54,803 male Sergeants compared to 10,954 female Sergeants, illustrating a substantial imbalance. This pattern is also evident at other ranks, where counts for female soldiers remain consistently lower.

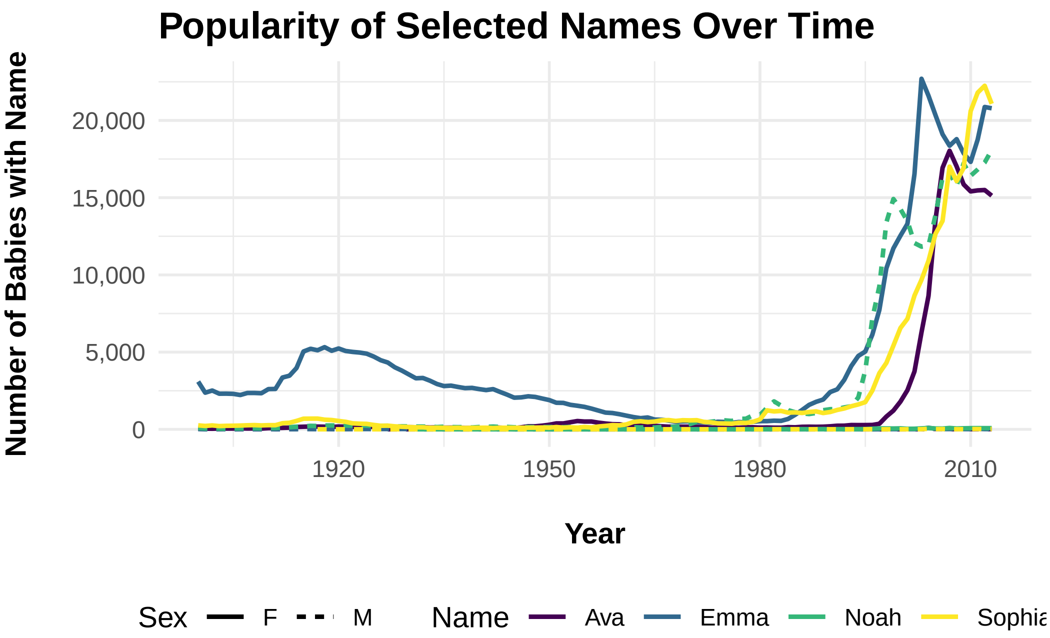
The representation of female soldiers remains low across all ranks in the table, with the most pronounced difference occurring at the Sergeant rank. These differences suggest that sex and rank are not independent in this subgroup of the US Army. The observed patterns highlight disparities in rank distribution and provide important context for understanding gender representation within enlisted personnel.

Popularity of Baby Names

This analysis explores naming trends over time, examining the popularity patterns of selected names: Emma, Noah, Ava, Sophia, from 1900 to present across gender distributions.

Analysis and Visualization

Figure 1: Popularity Trends of Selected Baby Names Over Time



Narrative Text

Figure 1 reveals the popularity patterns across the selected baby names from 1920 to the present. Emma and Sophia show a resurgent popularity as evidenced by the steep increases beginning around 1990. In contrast, Ava demonstrates a more recent but dramatic surge, becoming one of the most popular names in the 21st century.

Noah maintains a stable male usage until experiencing growth since the 1990s. The visualization clearly shows gender distribution patterns, with Emma, Ava, and Sophia being predominantly female names (shown by the solid lines), while Noah is exclusively male (dashed lines).

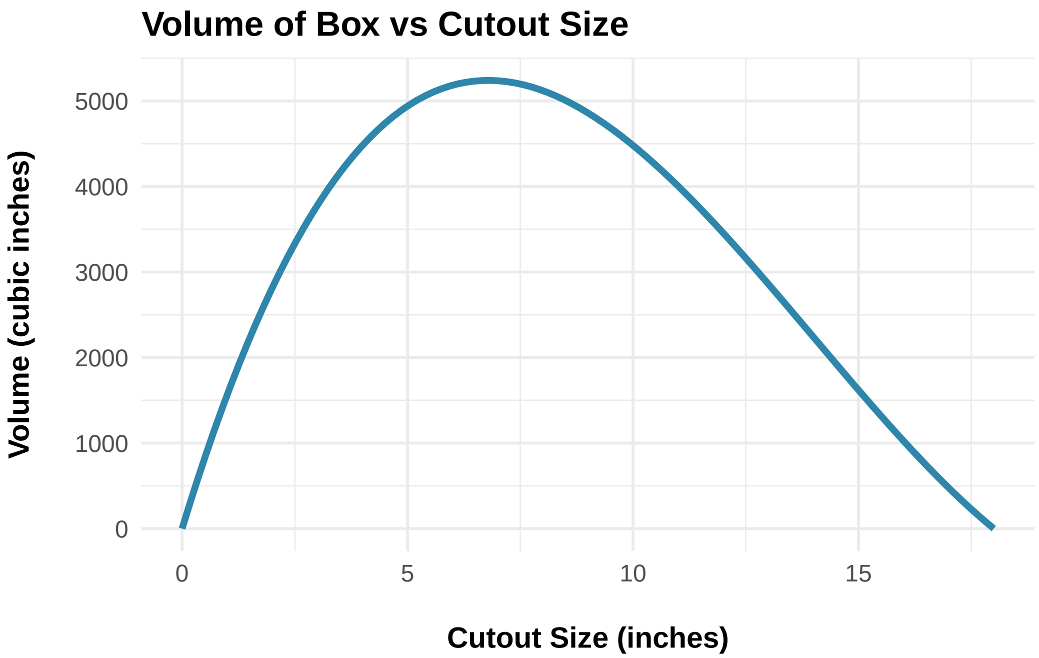
The data demonstrate how name popularity can experience significant rebirth after periods of decline shown in the dramatic comebacks of Emma and Sophia in recent decades.

Plotting a Mathematical Function

This section identifies the optimal cutout size to maximize volume when creating a box from a 36 × 48 inch sheet of paper, using mathematical modeling and function visualization.

Analysis and Visualization

Figure 2: Volume of Box vs Cutout Size for 36 x 48 Inches Paper



Narrative Text

Figure 2 demonstrates the relationship between cutout size and box volume for a 36 × 48 inches sheet of paper. The visualization shows a concave-down parabolic curve that starts at zero volume when no material is cut out ($x = 0$), rises to a maximum point, and then decreases back to zero when the cutout size reaches 18 inches.

The maximum volume occurs at approximately $x = 7$ inches, resulting in a box with a volume of about 3,700 cubic inches. This represents the ideal cutout size that maximizes the storage capacity of the box.

This analysis shows that cutting 7-inch squares from each corner of a 36 × 48 inches sheet produces a box with a maximum volume of approximately 3,700 cubic inches.

Data Sources

Armed Forces Data

1. U.S. Armed Forces Personnel Data:

https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qbwb_E/

2. Military Rank Structure: Official pay grade to rank mappings

https://neilhatfield.github.io/Stat184_PayGradeRanks.html

3. Analysis Tools: `tidyverse`, `rvest`, `googlesheets4`, `kableExtra` for data wrangling, web scraping, and table formatting

Baby Names Data

1. Data Package: `dcData` - contains curated BabyNames dataset
2. Analysis Tools: `dplyr` for data filtering and aggregation, `ggplot2` for time series visualization

Box Problem

1. Analysis Tools: `ggplot2` with `stat_function()` for mathematical visualization

---

Code Appendix

Armed Forces Data Wrangling Code

```r
# Step 1: Load Packages ----
library(tidyverse)
library(rvest)
library(googlesheets4)
library(kableExtra)

# Step 2: Scrape Rank Data ----
webRanks <- read_html("https://neilhatfield.github.io/Stat184_PayGradeRanks.html") %>%
  html_elements(css = "table") %>%
  html_table()

rawRanks <- webRanks[[1]] # Extract the data frame of ranks

# Step 3: Wrangle Rank Data ----
rawRanks[1, 1] <- "Type"
rankHeaders <- rawRanks[1, ]
names(rawRanks) <- rankHeaders[1,]
rawRanks <- rawRanks[-c(1, 26), ]

cleanRanks <- rawRanks %>%
  dplyr::select(!Type) %>% # Remove extra column
  pivot_longer(
    cols = !`Pay Grade`, # The improper name requires backticks
    names_to = "Branch",
    values_to = "Rank"
  ) %>%
  mutate(
    Rank = na_if(x = Rank, y = "--")
  )

# Step 4: Load Armed Forces Data ----
gs4_deauth() # Prevents needing to sign into a Google account
forcesHeaders <- read_sheet(
  ss = "https://docs.google.com/spreadsheets/
  d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qbwb_E/edit?usp=sharing",
  col_names = FALSE, # Turn off Column Names
  n_max = 3 # read only the first three rows
)

rawForces <- read_sheet(
  ss = "https://docs.google.com/spreadsheets/
  d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qbwb_E/edit?usp=sharing",
  col_names = FALSE,
  skip = 3,
  n_max = 28,
  na = c("N/A*") # Tells R to treat the N/A* as missing values
)

# Step 5: Wrangle Armed Forces Data ----
## Step 5a: Create good column names ----
branchNames <- rep( # Create three copies of each branch
  x = c("Army", "Navy", "Marine Corps", "Air Force", "Space Force", "Total"),
  each = 3
)
tempHeaders <- paste( # Combine branch with other headers
  c("", branchNames),
  forcesHeaders[3,],
  sep = "."
)

names(rawForces) <- tempHeaders

## Step 5b: Wrangle group-level data ----
forces_group_level <- rawForces %>%
  rename(Pay.Grade = `.Pay Grade`) %>%
  dplyr::select(!contains("Total")) %>% # Remove total columns
  filter( # Remove total rows
    Pay.Grade != "Total Enlisted" &
      Pay.Grade != "Total Warrant Officers" &
      Pay.Grade != "Total Officers" &
      Pay.Grade != "Total"
  ) %>%
  pivot_longer( # Reshape data
    cols = !Pay.Grade,
    names_to = "Branch.Sex",
    values_to = "Frequency"
  ) %>%
  separate_wider_delim( # Separate branches and sex
    cols = Branch.Sex,
    delim = ".",
    names = c("Branch", "Sex")
  )

# Step 6: Merge Data Frames ----
forces_with_ranks <- left_join(
  x = forces_group_level,
  y = cleanRanks,
  by = join_by(Pay.Grade == `Pay Grade`, Branch == Branch)
)

# Step 7: Create Individual Soldier Data
individual_soldiers <- forces_with_ranks %>%
  filter(!is.na(Frequency)) %>% # Remove all cases with missing counts
  uncount(
    weights = Frequency
  )

# Step 8: Create Data for Frequency Table Analysis ----
# Using the individual soldier data as required
army_enlisted_data <- individual_soldiers %>%
  filter(
    Branch == "Army",
    Pay.Grade %in% c("E1", "E2", "E3", "E4", "E5", "E6", "E7", "E8", "E9"),
    Rank %in% c("Private", "Corporal", "Sergeant", "Staff Sergeant",
                "Sergeant First Class", "Master Sergeant")
  )

# Step 9: Create 2-Way Frequency Table ----
army_sex_rank_table <- army_enlisted_data %>%
  count(Sex, Rank) %>% # Count individuals by sex and rank
  pivot_wider(
    names_from = Sex,
    values_from = n,
    values_fill = 0
  )

# Step 10: Display Table in Main Document
knitr::kable(
  army_sex_rank_table,
  format.args = list(big.mark = ",")
) %>%
  kable_styling(latex_options = "scale_down")
```

Popularity of Baby Names

#| fig-alt: “Line plot showing popularity trends for four baby names (Emma, Noah, Ava, Sophia) from 1900 to present”

```
# Step 1: Load required packages
library(dcData)
library(dplyr)
library(ggplot2)
```

```
# Step 2: Load the BabyNames dataset
data("BabyNames")

# Step 3: Filter the data for the selected names
selected_names <- c("Emma", "Noah", "Ava", "Sophia")

tidied_data <- BabyNames %>%
  filter(name %in% selected_names, year >= 1900) %>%
  group_by(name, sex, year) %>%
  # Summarize total count by name, sex, and year
  summarise(total_count = sum(count), .groups = "drop")

# Step 4: Create time series plot
ggplot(data = tidied_data,
       mapping = aes(x = year, y = total_count,
                     color = name, linetype = sex)) +
  geom_line(linewidth = 0.8) +
  labs(
    title = "Popularity of Selected Names Over Time",
    x = "Year",
    y = "Number of Babies with Name",
    color = "Name",
    linetype = "Sex"
  ) +
  scale_y_continuous(labels = scales::comma) +
  scale_color_viridis_d() +
  theme_minimal() +
  theme(legend.position = "bottom")
```

Plotting a Mathematical Function

#| fig-alt: “Curved plot showing volume of a box as a function of cutout size from a 36 x 48 inches paper. X-axis shows cutout length from 0 to 18 inches, y-axis shows volume in cubic inches. The curve starts at zero, rises to a maximum, then decreases back to zero.”

```
# Step 1: Use ggplot2 as required
library(ggplot2)

# Step 2: Define volume function for 36 x 48 inch paper
# given the cutout size x and paper dimensions length x width
get_box_volume <- function(x, length = 48, width = 36) {
  volume <- x * (length - 2 * x) * (width - 2 * x)
  return(volume)
}

# Step 3: Create plot using the stat_function
ggplot(data = data.frame(x = c(0, 18)), aes(x = x)) +
  stat_function(fun = get_box_volume, linewidth = 1.2, color = "#2E86AB") +
  labs(
    title = "Volume of Box vs Cutout Size",
    x = "Cutout Size (inches)",
    y = "Volume (cubic inches)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold"),
    axis.title.x = element_text(face = "bold", margin = margin(t = 15)),
    axis.title.y = element_text(face = "bold", margin = margin(r = 15))
  )
```

Self Reflection

Throughout this course, I have developed a comprehensive understanding of data science through programming in R. From wrangling data, computing statistical results programmatically, to creating clear visualizations and interpreting outputs, I have learned how to systematically approach complex problems.

Working with the Armed Forces dataset, for example, significantly helped me learn how to wrangle and transform data using R functions. Using pivot_longer(), filter(), and other dplyr functions taught me how to merge multiple datasets and break down a large dataset into several smaller data frames for easier manipulation. This workflow is essential for analytical needs. I enjoyed being able to blend statistical thinking with programming specifically in R.

I have also gained strong skills in statistical visualization through ggplot2. Learning how to create different types of visualizations, labeling them properly, and making data more accessible to everyone has improved my ability to communicate results effectively following the principles of making effective graphs from Tufte and Kosslyn.

The box problem, in particular, combined both mathematical modeling with functional visualization. Using tools like stat_function() showed me how analytical reasoning paired with computational visualization can create representations of complex mathematical relationships. This is something that would be difficult to produce by hand.

Through Quarto documentation, I have learned the importance of organizing code into well commented sections and producing comprehensive code appendices. This process has helped me understand how to approach analytical research projects more professionally.

The bridge between statistical analysis, data science workflows, and clean documentation in R is what I am most proud of developing.