Statistical Analysis: Armed Forces, Baby Names, and the Box Problem

Jasmine Randhawa

2025-11-11

Armed Forces Data Wrangling

Enlisted Marines Frequency Table by Gender and Rank

Rank	Male	Female	Difference	Total
Lance Corporal	35239	4174	31065	39413
Corporal	28519	2961	25558	31480
Sergeant	22262	2670	19592	24932
Private First Class	15034	1684	13350	16718
Staff Sergeant	12225	1529	10696	13754
Private	7849	655	7194	8504
Gunnery Sergeant	7720	747	6973	8467
Captain	5385	707	4678	6092
Major	3637	338	3299	3975
First Sergeant OR Master Sergeant	3495	293	3202	3788
First Lieutenant	3162	525	2637	3687
Second Lieutenant	2412	366	2046	2778
Lieutenant Colonel	1830	137	1693	1967
Chief Warrant Officer	1612	100	1512	1712
Sergent Major OR Master Gunnery Sergeant	1515	82	1433	1597
Colonel	656	54	602	710
Warrant Officer	494	44	450	538
Brigadier General	36	2	34	38
Major General	28	2	26	30
Lieutenant General	17	1	16	18
General	3	0	3	3

Differences in Male and Female Enlistment in the Marines by Rank

Marine Frequency Table Narrative Text

When deciding what rank to focus on, I chose the Marines as it is viewed as an elite branch of the US Military. I also assumed that it would have the least amount of females serving. Using the US Armed Forces data that I wrangled, I created a frequency table that would look at the relationship between sex and rank in the Marine Corps.

The frequency table that I created displayed the number of females and males that were enlisted for each rank of the Marines. It also included the total number of people enlisted, as well as the difference between the number of males enlisted and the number of females enlisted. I created the difference column, so it would be easier to see the disparity in every rank. I chose to include every rank of the Marines that had observations, and I arranged the table so it would go from the ranks with the most people to the least. I believe that this makes it a lot easier to see how the ratio of men to women in each rank gets larger, as the ranks get more prestigious.

Even in the lowest rank of the Marines, there are a lot more men than women that are enlisted. At almost every rank, there are at least 10 male Marines for every female Marine. This disparity gets even larger at the higher ranks. There is not a single female in the highest rank that a Marine can hold, General. The second highest rank, Lieutenant General, only has one female officer.

The lack of women throughout the Marines, but especially at the higher rank levels demonstrates that sex and rank are dependent on each other. In the Marines, it is less likely for a female to even enlist and even less likely for them to hold a high rank. Being a female in the Marines means it's less likely for you to get a rank promotion than your male colleagues.

Popularity of Baby Names

Name Trends Visualization

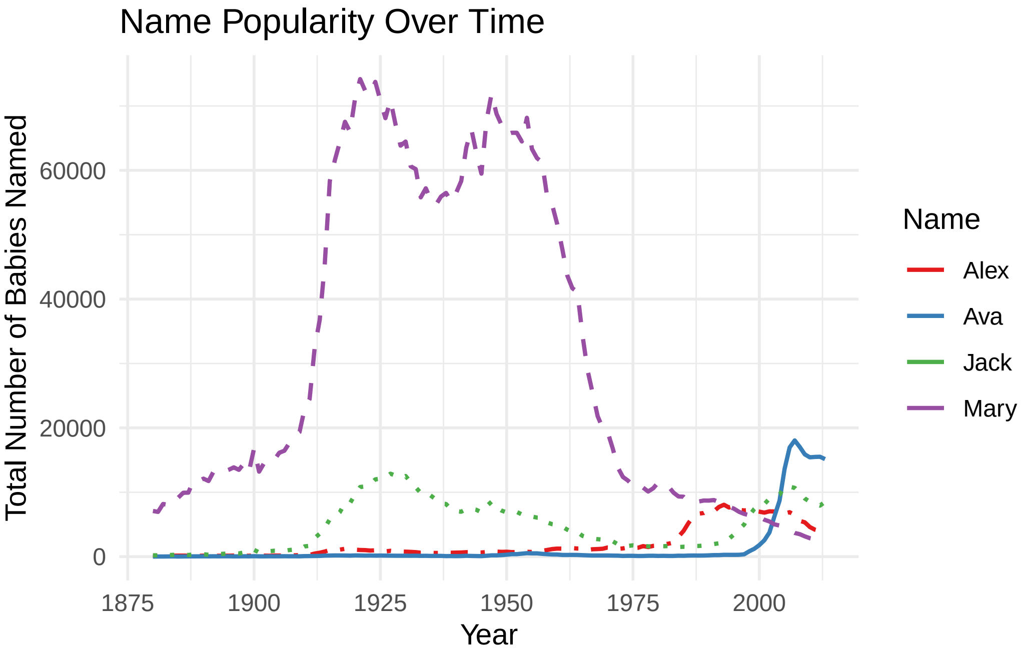


Figure 1: Trends in Baby Name Popularity (1880-2013)

Baby Name Narrative Text

When doing this assignment, I somewhat randomly chose the names ‘Mary’, ‘Ava’, ‘Jack’, and ‘Alex’. I’ve met a lot of people with those names and I thought that they would be interesting to analyze. I knew that Ava was a more trendy name compared to the rest, as my parents were considering naming me Ava when I was born.

Still, when I generated the graph, I was surprised to see the results. The name Mary peaked in popularity around 1920 and again in around 1945. I knew that Mary would’ve been a popular name since it’s a biblical name, but I didn’t realize that it would be so much more popular than the other names. The name Jack was never as popular as Mary but it also peaked around 1920 and then again in the 2000s. Alex also had stable popularity, but became more popular in the 1990s where it also remained stable. Ava was more clearly a trendy name due to it being rare until the early 2000s when it rapidly rose upwards.

All of these choices show how trends affect how we name babies in the United States. Some names will always have a moderate level of popularity, while others will be trends that will fall off after some time. While some trends might last for decades, others are much more short lived. However, due to some names having multiple peaks in the graph, we can see that trendy names can always come back and become popular again.

Plotting a Mathematical Function

Box Problem Visualization

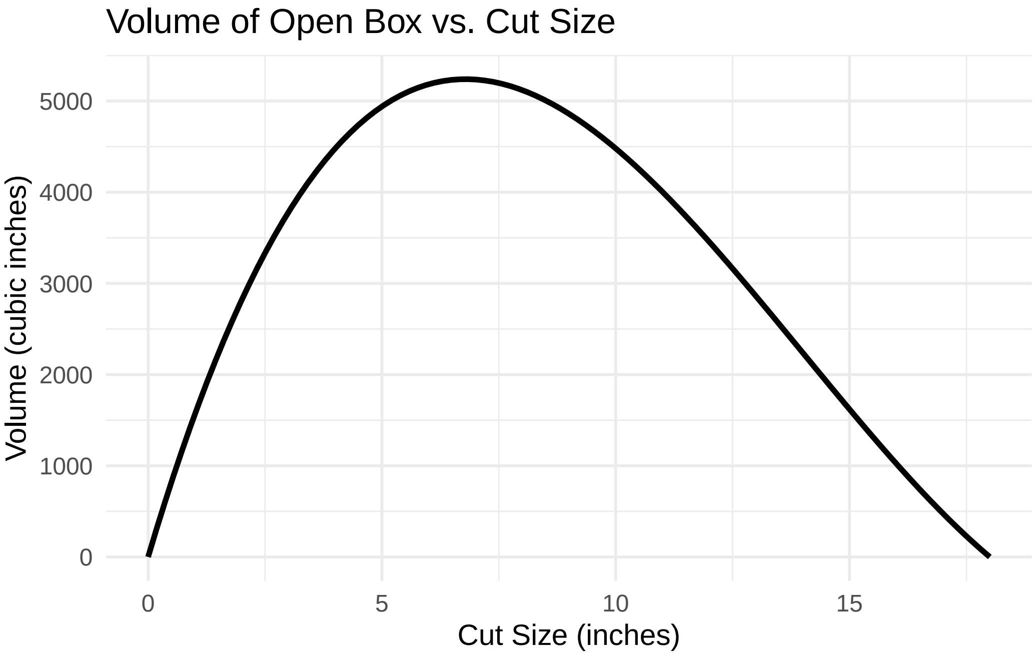


Figure 2: Volume of Open-Faced Box vs Cut Size for 36x48 inch Paper

Box Plot Narrative Text

The graph shows the relationship between the size of square cuts on a sheet of paper and the volume of an open-faced box created from the paper. The downwards parabola shows that the volume of the box initially grows as the cut size grows, then reaches a maximum volume, and after that decreases as the cuts become too big. The parabolic curve means that an increase in height correlates to a decrease in base area. That means there is an optimal cut, where the trade-off between height and base area is utilized to its fullest extent.

By looking at the graph, we can tell that the maximum volume is when the cut size is around 7 inches. The volume of the open-faced box would be approximately 5,200 cubic inches. The dimensions should be around 22x34x7 inches. The graph also shows that an open-faced box can be created as long as the cuts are greater than 0 inches (you can't create a box with no height), as well as less than 18 inches (you can't create a box if you have no material left).

What I've Learned So Far

Through taking STAT 184, I've developed an introductory understanding of R. I know that I'll be taking more courses to build on my skills in the future, which is why I won't say that I have a comprehensive understanding of the language. However, I feel confident using everything that we have learned in class (with the exception of getting Quarto to properly run on my laptop).

Taking this course made me feel more confident with my coding, and also gave me a deeper understanding of how my computer works (this however did stem from my issues with Quarto). When going into the semester, I was apprehensive to learn a new coding language however I have come to enjoy using R. This has inspired me to learn new coding languages once this semester is over, and to enhance my R skills.

Prior to taking this course, I had created data visualizations using python but it wasn't something that I was very skilled at. Now, my favorite thing to create is graphs and plots by using ggplot2. The teaching style of this class really helped me master this, along with the extensive documentation of R and ggplot2. I also knew how to wrangle data before taking this course, but only at a more basic level. I remember taking a lot of time to work on the Armed Forces Data Wrangling homework, but the difficulty of the assignment forced me to really understand how wrangling worked. Now, when I have to wrangle data, I can clearly understand why I'm cleaning the data instead of just trying to replicate examples and past works.

Taking this course also made me realize that I need to know how to explain my code and the visualizations that I create with it. I remember that I used to dislike commenting when I first started coding (I also hated creating docstrings). Creating longer and more complicated pieces of code showed me that comments are crucial for understanding code when you read it again or show it to another person. Not everyone understands how coding works, and even if you do understand it can be hard to follow someone's (or your own) thought process. Visualizations are the same, as context is valuable to understand what a graph or plot is trying to display. If a person were to just look at code, they wouldn't understand everything that you were trying to convey. By taking this class I learned that alternative text, narrative descriptions, and other things like it are equally as important as the code that I created. Realizing this made me change my thought process when I code, and my mindset now is to create and explain code in a way that can be easily understood by many people.

Overall, STAT 184 has been an incredibly valuable class. The concepts that I learned can be applied not only to R, but to other coding languages and when I'm writing reports. I'm thankful that I've taken this course as it's made me a better data scientist and coder.

Code Appendix

```r
# Armed Forces Data Wrangling
## Load Packages
library(tidyverse)
library(rvest)
library(knitr)
library(kableExtra)

# Step 1: Load data from Google Sheets
sheet_url <- "https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qbw
```

```
raw_army <- read_csv(sheet_url,
                     skip = 2,
                     na = c("N/A*", "N/A", "NA", ""))
```

```
New names:
Rows: 29 Columns: 19
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(1): Pay Grade num (18): Male...2, Female...3, Total...4, Male...5, Female...6,
Total...7, ...
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `Male` -> `Male...2`
* `Female` -> `Female...3`
* `Total` -> `Total...4`
* `Male` -> `Male...5`
* `Female` -> `Female...6`
* `Total` -> `Total...7`
* `Male` -> `Male...8`
* `Female` -> `Female...9`
* `Total` -> `Total...10`
* `Male` -> `Male...11`
* `Female` -> `Female...12`
* `Total` -> `Total...13`
* `Male` -> `Male...14`
* `Female` -> `Female...15`
* `Total` -> `Total...16`
* `Male` -> `Male...17`
* `Female` -> `Female...18`
* `Total` -> `Total...19`
```

```
# Step 2: Change column names
army_col_names <- c("Pay_Grade",
                    "Army__Male", "Army__Female", "Army__Total",
                    "Navy__Male", "Navy__Female", "Navy__Total",
                    "Marine_Corps__Male", "Marine_Corps__Female", "Marine_Corps__Total",
                    "Air_Force__Male", "Air_Force__Female", "Air_Force__Total",
                    "Space_Force__Male", "Space_Force__Female", "Space_Force__Total",
                    "Total__Male", "Total__Female", "Total__Total")
names(raw_army) <- army_col_names

# Step 3: Remove total rows
pay_grades <- c("E1", "E2", "E3", "E4", "E5", "E6", "E7", "E8", "E9",
                "W1", "W2", "W3", "W4", "W5",
                "O1", "O2", "O3", "O4", "O5", "O6", "O7", "O8", "O9", "O10")
cleaned_army <- raw_army %>%
  filter(Pay_Grade %in% pay_grades) %>%
  mutate(across(-Pay_Grade, ~ as.numeric(.))) # convert to numeric

# Step 4: Convert data from wide to narrow
long_army <- cleaned_army %>%
  pivot_longer(
    cols = -Pay_Grade, # Keep pay_grade
    names_to = "Demographic", # For service_branch and sex
    values_to = "Count"
  )

# Step 5: Finish grouped data frame
grouped_army <- long_army %>%
  separate(
    col = Demographic,
    into = c("Service_Branch", "Sex"),
    sep = "__"
  ) %>%
  filter(Sex %in% c("Male", "Female"),
         Service_Branch != "Total",
         Sex != "Total")

# Step 6: Start making individual frame, create individual rows
individual_army <- grouped_army %>%
  filter(!is.na(Count)) %>%  # Filters out NA values so uncount can run
  uncount(weights = Count, .id = "Soldier_ID")

# Step 7: Scrape and clean the rank information page
rank_url <- "https://neilhatfield.github.io/Stat184_PayGradeRanks.html"
rank_page <- read_html(rank_url)

## Extracts the table
rank_table <- rank_page %>%
  html_nodes("table") %>%
  html_table(fill = TRUE) %>%
  .[[1]]

## Clean column names
colnames(rank_table) <- c("Extra_Info", "Pay_Grade", "Army", "Navy", "Marine_Corps",
                          "Air_Force", "Space_Force", "Coast_Guard")

## Remove the header row and the bottom row
rank_table <- rank_table[-1, ]
rank_table <- rank_table[-25, ]

## Remove the "Extra_Info" column
rank_table <- rank_table[ ,-1]
```

```r
## Move to narrow format
rank_long <- rank_table %>%
  pivot_longer(
    cols = -Pay_Grade,
    names_to = "Service_Branch",
    values_to = "Rank"
  ) %>%
  filter(Rank != "" & !is.na(Rank) & Rank != "--") %>%
  mutate(Service_Branch = case_when(
    Service_Branch == "Army" ~ "Army",
    Service_Branch == "Navy" ~ "Navy",
    Service_Branch == "Marine_Corps" ~ "Marine_Corps",
    Service_Branch == "Air_Force" ~ "Air_Force",
    Service_Branch == "Space_Force" ~ "Space_Force",
    Service_Branch == "Coast_Guard" ~ "Coast_Guard",
  ))


# Step 8: Add rank information to both tables
grouped_army_final <- grouped_army %>%
  left_join(rank_long, by = c("Pay_Grade", "Service_Branch"))

individual_army_final <- individual_army %>%
  left_join(rank_long, by = c("Pay_Grade", "Service_Branch"))

# Step 9:  Create a function to make the frequency tables
create_freq_table <- function(data, gender) {
  data %>%
    filter(Sex == gender) %>%
    count(Service_Branch, Rank) %>%
    pivot_wider(
      names_from = Service_Branch,
      values_from = n,
      values_fill = 0
    ) %>%
    # Calculate row totals
    mutate(Total = rowSums(across(where(is.numeric)))) %>%
    # Add total row
    bind_rows(
      summarise(., across(where(is.numeric), sum)) %>%
        mutate(Rank = "Total")
    )
}

# Step 10: Create male and female tables using individual_army_final
male_freq_final <- create_freq_table(individual_army_final, "Male")
female_freq_final <- create_freq_table(individual_army_final, "Female")
```

```r
# Step 11: Compare male vs female enlistment by rank in Marine Corps
marine_comparison <- full_join(
  male_freq_final %>% select(Rank, Marine_Corps) %>% rename(Male = Marine_Corps),
  female_freq_final %>% select(Rank, Marine_Corps) %>% rename(Female = Marine_Corps),
  by = "Rank"
) %>%
  mutate(
    Male = replace_na(Male, 0),
    Female = replace_na(Female, 0),
    Difference = Male - Female,
    Total = Male + Female
  ) %>%
  filter(Rank != "Total", Total != 0) %>%
  arrange(desc(Total))  # Sorts the table by how many people are given a rank

## View the result
knitr::kable(marine_comparison)
```

Rank	Male	Female	Difference	Total
Lance Corporal	35239	4174	31065	39413
Corporal	28519	2961	25558	31480
Sergeant	22262	2670	19592	24932
Private First Class	15034	1684	13350	16718
Staff Sergeant	12225	1529	10696	13754
Private	7849	655	7194	8504
Gunnery Sergeant	7720	747	6973	8467
Captain	5385	707	4678	6092
Major	3637	338	3299	3975
First Sergeant OR Master Sergeant	3495	293	3202	3788
First Lieutenant	3162	525	2637	3687
Second Lieutenant	2412	366	2046	2778
Lieutenant Colonel	1830	137	1693	1967
Chief Warrant Officer	1612	100	1512	1712
Sergent Major OR Master Gunnery Sergeant	1515	82	1433	1597
Colonel	656	54	602	710
Warrant Officer	494	44	450	538
Brigadier General	36	2	34	38
Major General	28	2	26	30
Lieutenant General	17	1	16	18
General	3	0	3	3

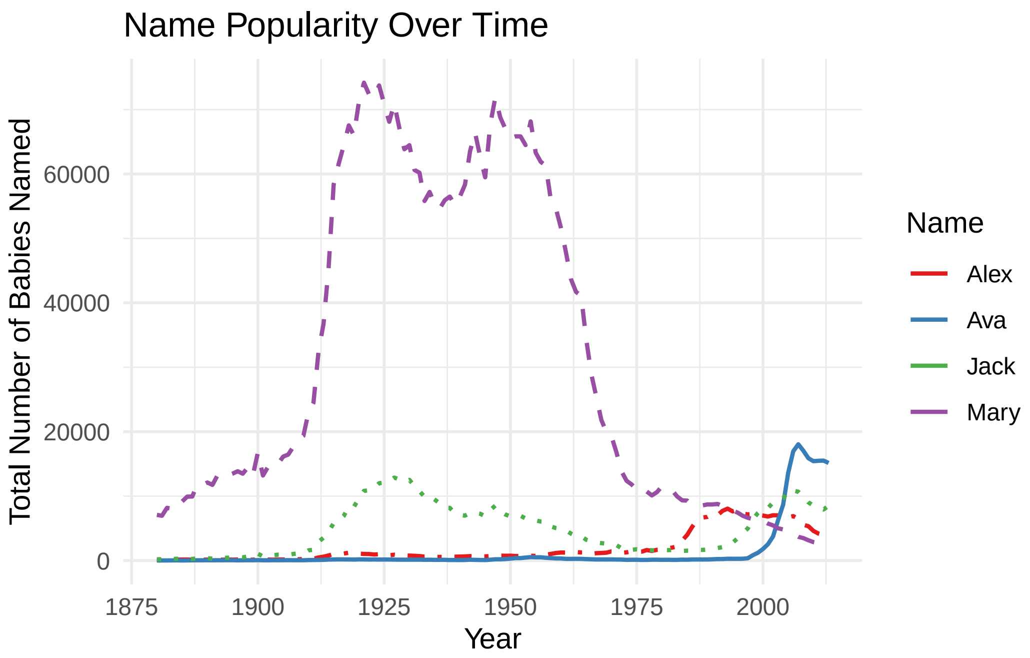
```r
# Baby Names Analysis Code
## Load required packages and data
library(dcData)
library(ggplot2)
```

```
data(BabyNames)

# Step 1: Filter for selected names
selected_names <- c("Jack", "Alex", "Ava", "Mary")
names_filtered <- BabyNames %>% filter(name %in% selected_names)

# Step 2: Summarize data by year and name
names_summary <- names_filtered %>%
  group_by(year, name) %>%
  summarise(total = sum(count), .groups = 'drop')

# Step 3:  Create time plot for the names
ggplot(names_summary, aes(x = year, y = total, color = name, linetype = name)) +
  geom_line(linewidth = 0.75) +
  scale_linetype_manual(values = c("Ava" = "solid", "Jack" = "dotted",
                                   "Mary" = "dashed", "Alex" = "dotdash")) +
  labs(title = "Name Popularity Over Time",
       x = "Year",
       y = "Total Number of Babies Named",
       color = "Name",
       alt = "Line plot showing popularity trends of four baby names from 1880 to 2013") +
  guides(linetype = "none") +
  theme_minimal() +
  scale_color_brewer(palette = "Set1")
```

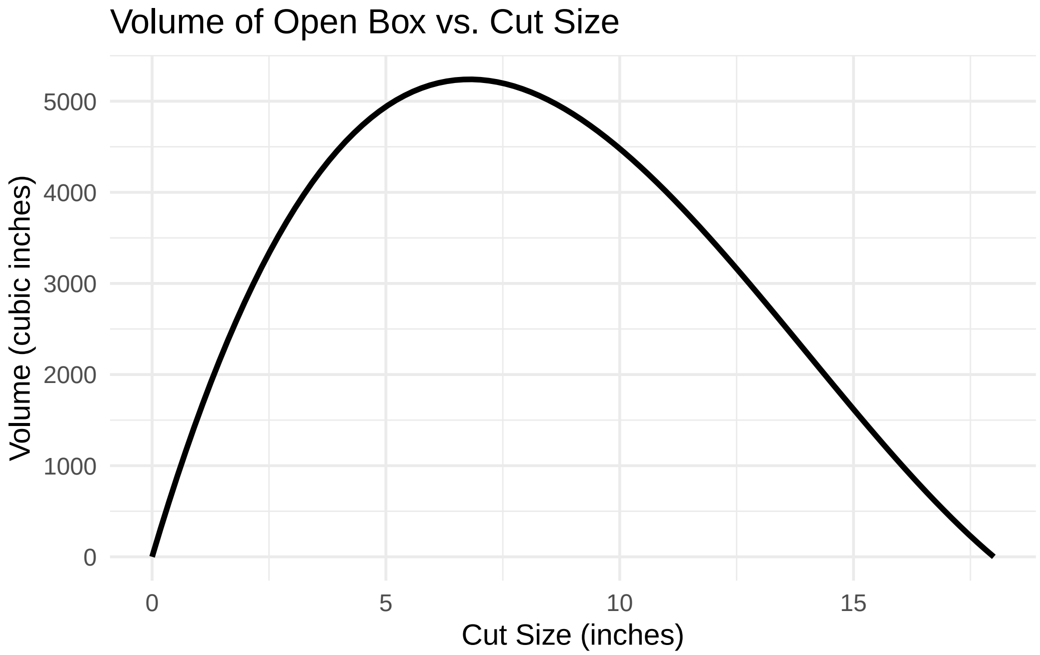


```
# Box Problem Code
## Load required package
```

```r
library(ggplot2)

# Step 1: Define box volume function for 36x48 inch paper
box_volume <- function(cut_size) {
  length <- 36 - (2 * cut_size)
  width <- 48 - (2 * cut_size)
  height <- cut_size
  volume <- length * width * height
  return(volume)
}

# Step 2:  Create plot using stat_function
ggplot(data.frame(x = c(0, 18)), aes(x = x)) +
  stat_function(fun = box_volume, linewidth = 1) +
  labs(title = "Volume of Open Box vs. Cut Size",
       x = "Cut Size (inches)",
       y = "Volume (cubic inches)",
       alt = "Parabolic curve showing box volume versus cut size for 36x48 inch paper") +
  theme_minimal()
```


Volume of Open Box vs. Cut Size
Volume (cubic inches)
Cut Size (inches)