Activity 14 QMD File

Shriyans Nellutla

2025-11-16

1 Armed Forces Data Wrangling Redux (Activities #08 and #10)

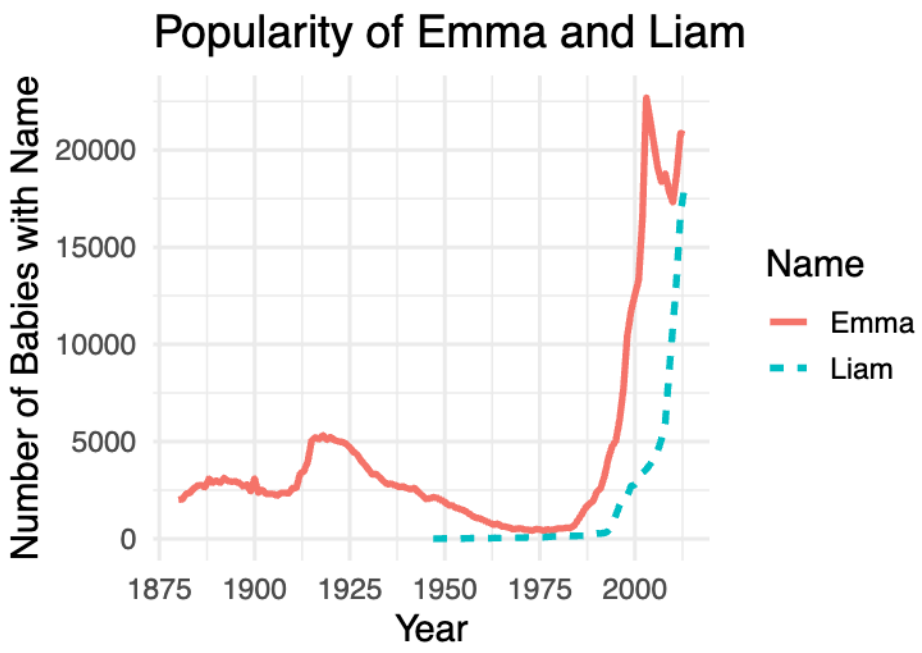
Table 1: Two-Way Frequency Table: Army Enlisted by Sex and Rank

Sex	Rank	Frequency
Female	Corporal OR Specialist	14619
Male	Corporal OR Specialist	81278
Female	First Sergeant OR Master Sergeant	1426
Male	First Sergeant OR Master Sergeant	9287
Female	Private	4705
Male	Private	26672
Female	Private First Class	8169
Male	Private First Class	38802
Female	Sergeant	11111
Male	Sergeant	55671
Female	Sergeant First Class	4322
Male	Sergeant First Class	30367
Female	Sergeant Major OR Command Sergeant Major	413
Male	Sergeant Major OR Command Sergeant Major	2908
Female	Staff Sergeant	7432
Male	Staff Sergeant	50030

The data show that sex and rank are not independent among enlisted Army personnel. After wrangling the dataset so each row represented an individual soldier with their branch, sex, and rank, I focused on the Army enlisted subgroup and created a two-way frequency table. While men outnumber women at every rank, the gap becomes much wider as the ranks increase. Female representation is noticeably higher at the lower enlisted levels, like Private or Specialist, but it steadily drops off in the senior enlisted positions, such as First Sergeant or Sergeant Major. If sex and rank were independent, we'd expect the proportion of men and women to remain fairly consistent across all ranks, but that clearly isn't the case here. Instead, the pattern suggests that women are less likely to occupy higher enlisted roles, indicating that rank advancement in the Army is associated with sex rather than being evenly distributed across genders.

2 Popularity of Baby Names (Activity #13)

Figure 1: Popularity of Emma and Liam vs Time



I chose the names Emma and Liam because I wanted to compare a name that has been common in the United States for a long time with one that seemed more recent. Emma has been used for generations, so I expected it to appear consistently in the data, while Liam felt like a name that only became popular in the modern era. The time series supports this idea because Emma stays fairly steady for many decades, and Liam barely shows up until the late twentieth century. After that point, both names rise quickly, especially during the 2000s. This increase may be related to cultural influences such as trends in entertainment or changing naming preferences, along with growth in the overall U.S. population. The visualization clearly shows how naming patterns can shift over time and how a name can become widely popular in a relatively short period.

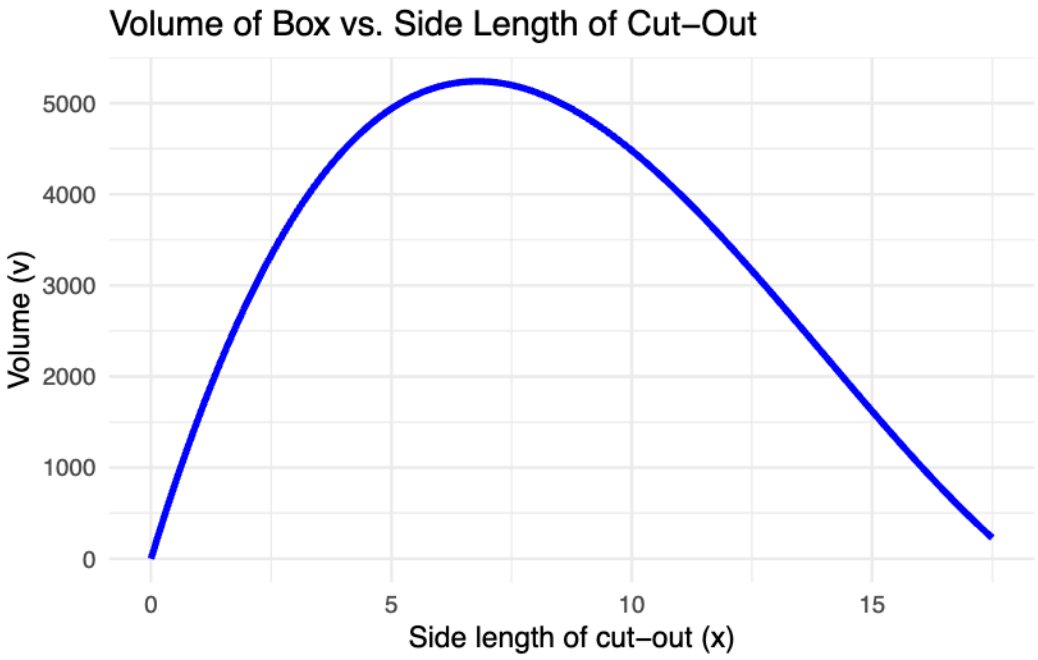
3 Plotting a Mathematical Function (Activity #04)

We want to visualize how the volume of an open-top box changes as we vary the side length of the square cutouts made from each corner of an 36" by 48" sheet of paper.

The formula for the volume is:

$V(x) = (36 - 2x)(48 - 2x)x$

Figure 2: Plot of Volume vs Cut-Out Length



The graph shows how the volume of the box changes as the cut out size increases, and it helps illustrate why there is an optimal value of x. At very small cut outs, the volume is low because the box is not tall enough. As x increases, the box becomes deeper, and the volume rises quickly until it reaches a clear peak. The plot makes this maximum easy to see because the curve tops out at around $x = 7$ inches, where the volume is roughly 5250 cubic inches. After that point, the volume starts to fall again since cutting out too much shrinks the base of the box. Overall, the visualization shows that the maximum volume for a box made from a 36 by 48 inch sheet occurs when the cut out side length is about 7 inches.

4 What I Feel I've Learned So Far

In STAT 184, I feel like I've really started to understand how to use R in a practical way. At the beginning of the course, I didn't fully know how RStudio worked, but now I'm a lot more comfortable using the interface, writing code, and keeping my work organized in projects and Quarto files. A big part of what I've learned is how to bring data into R and get it into a format that actually makes sense to work with. I've learned how to read CSV files, scrape tables from websites, and even pull data from Google Sheets. After that, using tidyverse functions like mutate, filter, pivot_longer, and separate has helped me get used to cleaning and reshaping messy datasets. I've realized that data wrangling is one of the most important steps, because nothing else works well unless the data is cleaned up first. I've also gotten better at exploring data once it is tidy. I can create summaries, calculate simple statistics, and make clear visuals with ggplot2. I'm starting to understand not just how to make the graphs, but how to explain what the graphs are showing and why it matters. Another thing I've picked up is how important it is to make my work reproducible. Writing Quarto documents that mix my code and explanations helps me stay organized and makes

it easier for someone else to follow what I did. Overall, I feel like I've built a solid base in R. I can take a dataset, clean it, analyze it, visualize it, and explain what I found. It feels like the course has helped me think more like a data analyst and less like someone who is just trying random code to see what works.

5 Code Appendix

```
### Armed Forces Data Wrangling Redux

# Load Packages
library(tidyverse)
library(rvest)
library(googlesheets4)
library(knitr)

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

rawRanks <- webRanks[[1]]

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

cleanRanks <- rawRanks %>%
  dplyr::select(!Type) %>%
  pivot_longer(
    cols = !`Pay Grade`,
    names_to = "Branch",
    values_to = "Rank"
  ) %>%
  mutate(
    Rank = na_if(x = Rank, y = "--")
  )

# Load Armed Forces Data
gs4_deauth()
forcesHeaders <- read_sheet(
  ss = "https://docs.google.com/spreadsheets/d/1cn4i0-ymB1ZytWXCwsJiq6fZ9PhGLUvbMBHlzqG4bwo/ed
```

```
  col_names = FALSE,
  n_max = 3
)

rawForces <- read_sheet(
  ss = "https://docs.google.com/spreadsheets/d/1cn4i0-ymB1ZytWXCwsJiq6fZ9PhGLUvbMBHlzqG4bwo/ed
  col_names = FALSE,
  skip = 3,
  n_max = 28,
  col_types = "c"
)

# Wrangle Armed Forces Data
branchNames <- rep(
  x = c("Army", "Navy", "Marine Corps", "Air Force", "Space Force", "Total"),
  each = 3
)
tempHeaders <- paste(
  c("", branchNames),
  forcesHeaders[3,],
  sep = "."
)

names(rawForces) <- tempHeaders

cleanForces <- rawForces %>%
  rename(Pay.Grade = `.Pay Grade`) %>%
  dplyr::select(!contains("Total")) %>%
  filter(Pay.Grade != "Total Enlisted" &
           Pay.Grade != "Total Warrant Officers" &
           Pay.Grade != "Total Officers" &
           Pay.Grade != "Total") %>%
  pivot_longer(
    cols = !Pay.Grade,
    names_to = "Branch.Sex",
    values_to = "Frequency"
  ) %>%
  separate_wider_delim(
    cols = Branch.Sex,
    delim = ".",
    names = c("Branch", "Sex")
  ) %>%
  mutate(
    Frequency = na_if(Frequency, y = "N/A*"),
    Frequency = parse_number(Frequency)
  )
```

```
# Merge Data Frames
key_forcesRanks <- left_join(
  x = cleanForces,
  y = cleanRanks,
  by = join_by(Pay.Grade == `Pay Grade`, Branch == Branch)
)

# Transform Group into Individual
key_individualRanks <- key_forcesRanks %>%
  filter(!is.na(Frequency)) %>%
  uncount(
    weights = Frequency
  )

# Focus on Army Enlisted personnel
army_enlisted <- key_individualRanks %>%
  filter(Branch == "Army", str_detect(Pay.Grade, "^E"))

# Create two-way frequency table of Sex vs Rank
table_df <- as.data.frame(table(army_enlisted$Sex, army_enlisted$Rank)) %>%
  rename(
    Sex = Var1,
    Rank = Var2,
    Frequency = Freq
  )

kable(table_df, caption="Two-Way Frequency Table: Army Enlisted by Sex and Rank")

### Popularity of Baby Names

# Load Packages
library(tidyverse)
library(dcData)
library(ggplot2)

# Data Wrangling
selected_names <- BabyNames %>%
  filter(name %in% c("Emma", "Liam")) %>%
  filter((name == "Emma" & sex == "F") | (name == "Liam" & sex == "M"))

# Data Visualization
ggplot(selected_names, aes(x=year,
                           y=count,
                           color=name,
                           linetype=name,
                           group=interaction(name, sex))) +
  geom_line(size = 1.2) +
```

```
  labs(
    title = "Popularity of Emma and Liam",
    x = "Year",
    y = "Number of Babies with Name",
    color = "Name",
    linetype = "Name"
  ) +
  theme_minimal(base_size = 14)

### Plotting a Mathematical Function

library(ggplot2)

# Function Definition
volume = function(x) {
      v = (36 - 2*x)*(48 - 2*x)*x
      return(v)
}

x_range <- data.frame(x = c(0, 17.5))

# Function Visualization Plot
ggplot(x_range, aes(x = x)) +
  stat_function(fun = volume, color = "blue", size = 1.2) +
  labs(
    title = "Volume of Box vs. Side Length of Cut-Out",
    x = "Side length of cut-out (x)",
    y = "Volume (v)"
  ) +
  theme_minimal()
```

tinytex::install_tinytex() install.packages("tinytex") library(tinytex)