

# **Activity 14**

**STAT 184**

Abigail Weinstein

2025-11-12

## **Table of contents**

|   |   |
|---|---|
| Assignment Introduction . . . . .             | 1 |
| Part 1: Wrangling Armed Forces Data . . . . . | 1 |
| Part 2: Baby Name Visualization . . . . .     | 3 |
| Part 3: Paper Box Function . . . . .          | 4 |
| Part 4: Course Reflection . . . . .           | 6 |
| Code Appendix . . . . .                       | 6 |

## **Assignment Introduction**

Activity #14 provides students with an opportunity to reflect upon past assignments, revise code, and introduces working with Quarto - a version of R Markdown that allows users to write and run code simultaneously. This assignment consists of four sections: 1) wrangle data and create a frequency table (Activity #8 and #10), 2) incorporate existing visualizations (Activity #13), 3) generate new visualizations (Activity #4), and 4) reflect upon the course material. Final submissions of the assignment should be in a **pdf** format.

## **Part 1: Wrangling Armed Forces Data**

This assignment portion is targeted towards efficiently wrangling data from the US Armed Forces data set and the producing a frequency table that demonstrates the relationship between sex and rank within a subset of the armed forces within a particular branch of the military. Data is based off of *pay grade* in the military and needs to be transformed appropriately. The data is in the form of a frequency table initially, which requires transformation to a typical dataframe in order to successfully work with the data.

## Cleaning Data and Creating Table

After scraping our datasets from the provide Google Sheet and html file path, we begin cleaning them. Full code can be seen in the Code Appendix. A brief description of the process will be described here. To work with both datasets, they were transposed so that the columns became rows and rows become columns. This situated the data into a more manageable framework. From here, the columns were renamed to represent the gender, branch, and pay grade (Soldier data) or pay grade and rank (Rank data). Unnecessary rows and columns were dropped from each dataset. Then, each data frame was pivoted and merged. Data was then separated from the groups they were in and organized as individual observations. After this was done, we filtered the dataset so that it only included data for those who were **officers** in the **Army**. With this done, we are able to begin working on creating the frequency table (shown in Table 1 ).

## Exploring the Table

Table 1: Rank and Gender of Army Officers

| Rank/Gender       | Female                 | Male                   | Total                   |
|-------------------|------------------------|------------------------|-------------------------|
| Second Lt.        | 2,358 (3.05%)          | 7,153 (9.26%)          | <b>9,511 (12.31%)</b>   |
| First Lt.         | 3,093 (4.00%)          | 10,013 (12.96%)        | <b>13,106 (16.96%)</b>  |
| Captain           | 5,739 (7.43%)          | 20,694 (26.78%)        | <b>26,433 (34.21%)</b>  |
| Major             | 3,002 (3.89%)          | 12,758 (16.51%)        | <b>15,760 (20.40%)</b>  |
| Lt. Colonel       | 1,539 (1.99%)          | 6,969 (9.02%)          | <b>8,508 (11.01%)</b>   |
| Colonel           | 588 (0.76%)            | 3,084 (3.99%)          | <b>3,672 (4.75%)</b>    |
| Brigadier General | 18 (0.02%)             | 87 (0.11%)             | <b>105 (0.14%)</b>      |
| Major General     | 7 (0.01%)              | 100 (0.13%)            | <b>107 (0.14%)</b>      |
| Lt. General       | 8 (0.01%)              | 41 (0.05%)             | <b>49 (0.06%)</b>       |
| General           | 1 (0.00%)              | 12 (0.02%)             | <b>13 (0.02%)</b>       |
| <b>Total</b>      | <b>16,353 (21.17%)</b> | <b>60,911 (78.83%)</b> | <b>77,264 (100.00%)</b> |

Table 1 shows the frequency table relating military rank and gender within Army officers. Military rank is shown along the vertical axis and ranges from the rank of “Second Lieutenant” (O1) to “General” (O10). Gender is ordered “female” and “male” across the horizontal axis. Initially, the table had been formatted such that military rank was on the horizontal axis, but due to formatting issues, it has been flipped. The last column and row of the table represent the total count and proportion of individuals in each category. There are a total of 77,264 observations within this subset of our data, 21.17% of which are female officers and 78.83% of which are male. Knowing the percentages and totals allow us to draw conclusions about the relationship between rank and gender among high-ranking Army officials.

Unsurprisingly, we see a high concentration of our population (nearly 98%) found with in the first 6 ranks of the Army - in the lieutantents and colonels. The gender divide across these ranks are fairly similar to the general gender divide within this subset. We see between 20% to 25% of First and Second Lieutenants, Captains, Majors, and Lieutenant Colonels being women. At this point though, we see a gradual decline in the proportion of women within these higher

ranks. Only about 12.5% of Colonels are women. Around 15% of Brigadier Generals and Lt. Generals are women. After this, we see a steep decline, with only 7% of Major Generals and Generals - our highest positions in the Army - are women. Out of the 13 Generals, only one is a woman. Seeing this decline in proportion of women in high ranking positions indicates to us that there is a relationship between rank and gender within the Army.

There is already a fairly universal gender divide in the Army. Again, only 21% of Army officers are women. This necessitates us shifting the idea we have of “gender equality” within the military - it will not necessarily be a 50/50 split regarding gender. Part of this is due to the military being significantly male-dominated. It was only 10 years ago that women were allowed to serve in active combat. In those 10 years, women have earned positions in every rank of the Army - including General. As such, it is understandable that there is a smaller concentration of women among Army officers, especially in the higher ranks. So, despite the glaring gender gap in rank (and likely participation) within the Army, these statistics are indicative of progress within the centuries-old institution, even if they don’t necessarily appear to be.

## **Part 2: Baby Name Visualization**

The second task we were given was to incorporate a line plot displaying the popularity of baby names over time. To do so, we needed to load and subset data from the Baby Names dataset. The names selected were that from my friend group (and my own): Abigail, Genevieve, Kara, and Jillian. Before we discuss the trends displayed in Figure 1 though, we will briefly go over how the plot was constructed.

### **Plot Construction**

After importing data, we filtered the BabyNames dataset to include our four names (Abigail, Jillian, Genevieve, and Kara), as well as to only include those associated with the female gender. This is to eliminate any possibly inflated measures due to gender. With this subset, we were able to create the line plot. To highlight the differences between each name, different colors and line styles were used. It was also important to set axis scales on this visual to ensure that values at varying points could be seen clearly.

### **Exploring the Visualization**

In Figure 1 , we see the line plot displaying the popularity of the names Abigail, Genevieve, Jillian, and Kara from 1880 to 2013. The y-axis of the visualization measures popularity in frequency of use annually and ranges from 0 uses to 16,000 uses, measured at increments of 2,000. This is data collected from the record of individual baby names for a given year. The year of analysis is displayed on the x-axis, ranging from 1880 to 2020 at 20 year intervals. Note that data collection is only through 2013. This visualization highlights three primary trends across this set of four names. The first trend we notice is that the name “Genevieve” was in

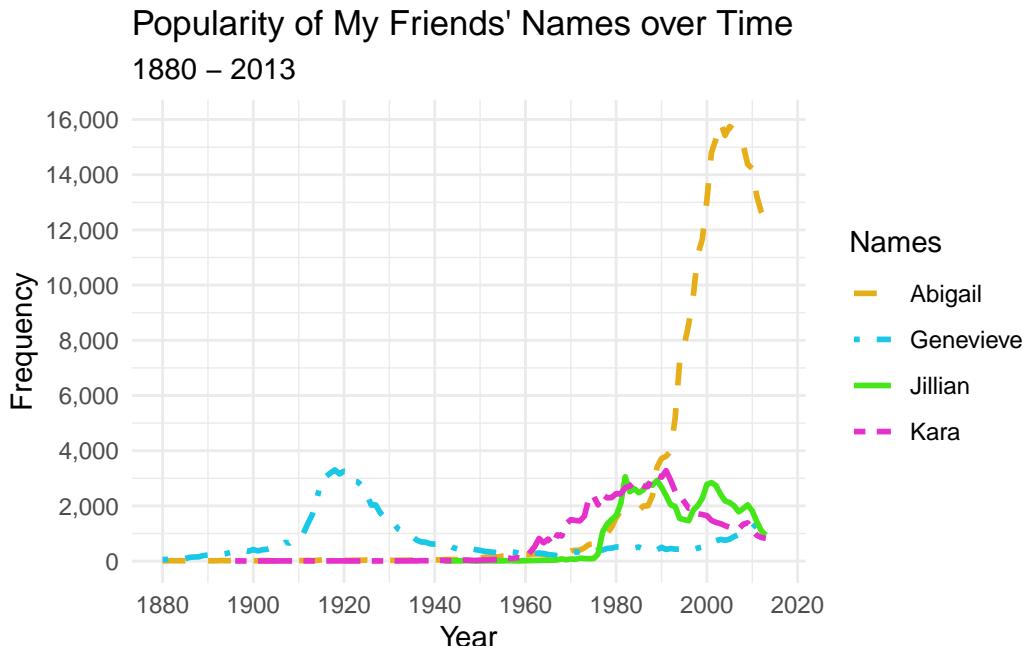


Figure 1

use nearly 70 years before any of the others, with use peaking in 1920. It was not a “popular” name by any means, with about 3,000 uses in its peak year. However, it is noticed as popular due to the lack of activity in the other names. “Genevieve” is represented by the teal dashed line. The second trend we note is the growth of the usage of the names “Kara” (pink dashed line) and “Jillian” (lime solid line), which both peaked in around 1990. Both have a peak popularity of around 3,000 as well. It should be noted that Kara saw gradual growth in use while Jillian experience a more rapid increase in relative popularity. The third trend we see in this visualization is the explosive growth in use of the name “Abigail” (yellow dashed line), which went mostly unused until the 1990s. The name saw a peak around 2000 with nearly 16,000 annual uses. This indicates that Abigail was incredibly popular name during this time - and, generally speaking, is the most popular name of the group.

### Part 3: Paper Box Function

The final coding task of this assignment was to modify a function and plot the output of that function. This function is derived from Activity #4, which required us to generate a function that produced the volume of a box given a 8.5x11 piece of paper and a continuous set of cuts to make from a corner. The function has since been modified to be a 36x48 piece of paper.

### Function and Plot Generation

To create our function, we utilized the volume formula: length x width x height. These values are reliant upon the size of the square cut taken from the paper, so it was necessary to reassign the length and width values to the constant dimensions minus the dimensions of the two square cuts. These values are plugged into the formula and the output is the corresponding volume.

To plot this, we utilize the function *stat\_function*, which allows us to plot a graph by only providing a set of x-values and a function. In this particular iteration, we opted to have our x-vector range from 0 to 12 on intervals of .1. After plotting the curve of our function, we also opted to highlight the peak of plot, which indicate the cut size that produces the greatest volume. We this in Figure 2 .

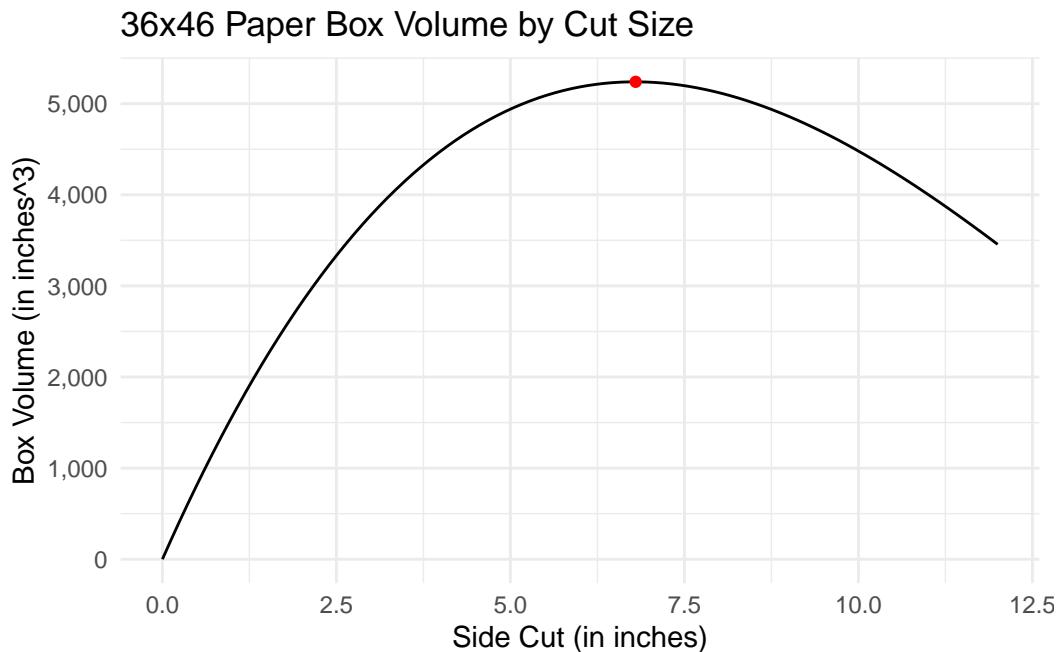


Figure 2

### Exploring the Visualization

Figure 2 shows the relationship between the size of a cut from a 36in x 48in piece of paper in inches and the volume of the box that could be produced from that cut. Our x-axis shows the side cut measures, which range from 0.0 to 12.0 inches and are incremented at 2.5 inch segments. The box volume is represented on the y-axis, which ranges from 0 cubic inches to 5,000 cubic inches on an interval of 1,000. The curve plotted rises rapidly from its point of origin at (0.0 in, 0.0 cubic in). As both cut length and volume increase, we begin to see a parabola shape form. It reaches its peak between 6.5 and 7.5 inches, with a volume of just over 5,000 cubic inches. After this, the volume begins to decline, even as the size of the cut

increases. This indicates that the relationship between the size of the piece cut from a paper and the volume of the resulting paper box are not necessarily positively or negatively linear, but rather quadratic. It should be noted that the decline in volume is more gradual than the increase in volume on the other side of the parabola. This is reasonable, as the more you remove from the length and width of the paper, you lose paper to make a box with. So, as height of the box increases, its other dimensions will eventually decrease such that if you cut too much, there will be a smaller box.

## Part 4: Course Reflection

The final section of this assignment is to reflect on the class itself. When I began this course, I had no experience working with RStudio. While I did have background with similar platforms, such as Stata, it was especially important that I learn how to use RStudio, as it is not only widely used in my field, but it is also being used in two of my other classes this semester. Since beginning this class, I feel significantly more competent in using not only R, but in working with other interfaces as well. My coding style is significantly more organized and thought out now than it was at the beginning of the semester. Even between a month ago and now, I feel stronger in my work in R. The data wrangling task from Activity #8, for example, was significantly more daunting last month than it would be now - and data wrangling is perhaps the most stressful part of working with R for me. There are certain aspects of the course I anticipate implementing more than others, such as the tools for visualization creation and developing unique functions. These will become especially useful as I move into data modeling. The visualization unit has also been beneficial outside of the context of learning how to make them, as the material coincided with work from my other courses this semester. The readings provided definitely supplemented not only what we were doing in class, but what we were covering in my visual analytics course. It has been exciting seeing how the skills learned in this course apply to both the higher-level methods courses I am taking and the theory-based courses.

## Code Appendix

### Part 1: Wrangling Armed Forces Data

```
###TASK 1:  
###WRANGLING US ARMED FORCES DATA  
  
#import packages  
library(tidyverse)  
library(rvest)  
library(googlesheets4)  
library(dplyr)
```

```

library(data.table)
library(janitor)
library(knitr)
library(kableExtra)
library(baizer)

#import data - soldier frequencies
gs4_deauth()
soldierGroup <- read_sheet(ss = 'https://docs.google.com/spreadsheets/d/1cn4i0-ymB1ZytWXCwsJ')

#import data - rank data
web_page <- "https://neilhatfield.github.io/Stat184_PayGradeRanks.html"
rank <- web_page %>% read_html() %>%
  html_nodes(css = "table") %>% html_table(fill = TRUE)
rank <- rank[[1]]

#flip soldier dataframe
newSoldier <- transpose(soldierGroup)

#update column names
colnames(newSoldier) <- as.character(unlist(newSoldier[1,]))
colnames(newSoldier)[1] <- c("Branch")
colnames(newSoldier)[2] <- c("Gender")

#fill in missing data
newSoldier$Branch <- c("Branch", rep("Army", times = 3),
                       rep("Navy", times = 3), rep("Marine Corps", times = 3),
                       = 3), rep("Total", times = 3))
newSoldier[newSoldier == "N/A*"] <- 0

#drop unnecessary columns and rows
newSoldier <- newSoldier[-c(1, 17:19), -c(31)]
newSoldier <- newSoldier %>%
  select(-contains("Total")) %>% filter(Gender != "Total")
#consistent data type
newSoldier <- newSoldier %>% mutate(
  across(
    .cols = E1:O10,
    as.numeric
  ))
  
#flip rank dataframe

```

```

rank <- transpose(rank)

#update column names
colnames(rank) <- as.character(unlist(rank[2,]))
colnames(rank)[1] <- "Branch"

#drop unnecessary rows and columns
rank <- rank[-c(1:2), -c(26)]

#pivot soldier data
newSoldier <- newSoldier %>% pivot_longer(
  cols = E1:O10,
  names_to = "PayGrade",
  values_to = "Count"
)

#pivot rank data
rank <- rank %>% pivot_longer(
  cols = E1:O10,
  names_to = "PayGrade",
  values_to = "Rank"
)

#merge data
newSoldier <- left_join(newSoldier, rank)

#uncount data
newSoldier <- uncount(newSoldier, Count, .remove = TRUE)
soldierIndivid <- newSoldier

#filter full dataframe
soldierFilter <- soldierIndivid %>%
  filter/Branch == "Army" & grepl("0", PayGrade))

#create relative frequencies
relArmy <- soldierFilter %>% tabyl(PayGrade, Gender) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_percentages(denominator = "all") %>%
  adorn_pct_formatting(digits = 2) %>%
  adorn_title(
    placement = "combined",
    row_name = "Rank",

```

```

col_name = "Gender")

#create absolute frequencies
formatTbl <- attr(relArmy, "core") %>%
  adorn_totals(where = c("row", "col")) %>%
  mutate(
    across(where(is.numeric), format, big.mark = ","))
  )

#create table base
armyTable <- relArmy %>%
  adorn_ns(position = "front", ns = formatTbl)
#sort table
armyTable <- armyTable %>% move_row(2, .after = 10)

#rename columns
newHeaders <- unique(soldierFilter$Rank)
newHeaders <- gsub("Lieutenant", "Lt.", newHeaders)
armyTable$`Rank/Gender` <- c(newHeaders, "Total")

#format table
armyTable %>%
  kable(
    caption = "Rank and Gender of Army Officers",
    booktabs = TRUE,
    align = c("l", rep("c", 3))
  ) %>% kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 6,
  ) %>% row_spec(11, bold = TRUE) %>% column_spec(4, bold = TRUE)

```

## Part 2: Baby Name Visualization

```

###TASK 2:
###BABY NAME LINE PLOT

#clear environment
rm(list=ls())

#import packages
library(devtools)

```

```

library(dcData)
library(ggplot2)
library(dplyr)
library(scales)

#import data
data('BabyNames', package = 'dcData')

#filter data
selectNames <- BabyNames %>%
  filter((name == 'Abigail' | name == 'Jillian' |
          name == 'Genevieve' | name == 'Kara') & sex == 'F')

#create plot
nameplot <- ggplot(data = selectNames, mapping = aes(
  x = year,
  y = count,
  color = name,
  linetype = name)) +
  geom_line(linewidth = 1) +
  scale_linetype_manual(name = "Names", values = c('dashed', 'dotdash','solid', 'twodash')) +
  scale_color_manual(name = "Names", values = c('#E6AE19', '#19C7E6', '#45E619', '#E630CB')) +
  theme_minimal() +
  scale_x_continuous(n.breaks = 10, limits = c(1880, 2015)) +
  scale_y_continuous(n.breaks = 10, labels = comma) +
  labs(x = "Year", y = "Frequency",title = "Popularity of My Friends' Names over Time", subtitle = "Top 10 Names")

```

### Part 3: Paper Box Function

```

####TASK 3:
####PAPER BOX FUNCTION

#clear environment
rm(list=ls())

#import libraries
library(ggplot2)
library(scales)
library(ggpmisc)

```

```

#creating function
Volume <- function(sideLength){
  INITIAL_LENGTH = 36
  INITIAL_WIDTH = 48
  length = INITIAL_LENGTH - (2 * sideLength)
  width = INITIAL_WIDTH - (2 * sideLength)
  height = sideLength

  volume = length * width * height
  return(volume)
}

#create data sequence
x = seq(from = 0, to = 12, by = .1)
y <- Volume(x)

#generate plot
volPlot <- ggplot(data.frame(x = x, y = y), mapping = aes(x, y)) +
  stat_function(fun = Volume) + theme_minimal() +
  labs(x = "Side Cut (in inches)", y = "Box Volume (in inches^3)",
       title = "36x46 Paper Box Volume by Cut Size") +
  scale_y_continuous(labels = comma) + stat_peaks(col = "red")
volPlot

```