Activity #14 - First QMD File

Ashley Song

2025-11-12

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

The table shows the number of Air Force Warranted Officers by sex across each officer rank in the June 2025 Armed Forces dataset. Each row represents an officer rank, and the columns show how many men and women fall into that rank, letting us compare how the distribution of sex changes as rank increases. Because the data were expanded to one row per soldier, the counts reflect actual numbers of individual officers. From the table, men consistently appear in much higher numbers than women across all officer ranks, and the size of the difference varies by rank. Since these proportions are not consistent across the ranks, sex and rank do not appear to be independent for this subgroup of the Air Force.

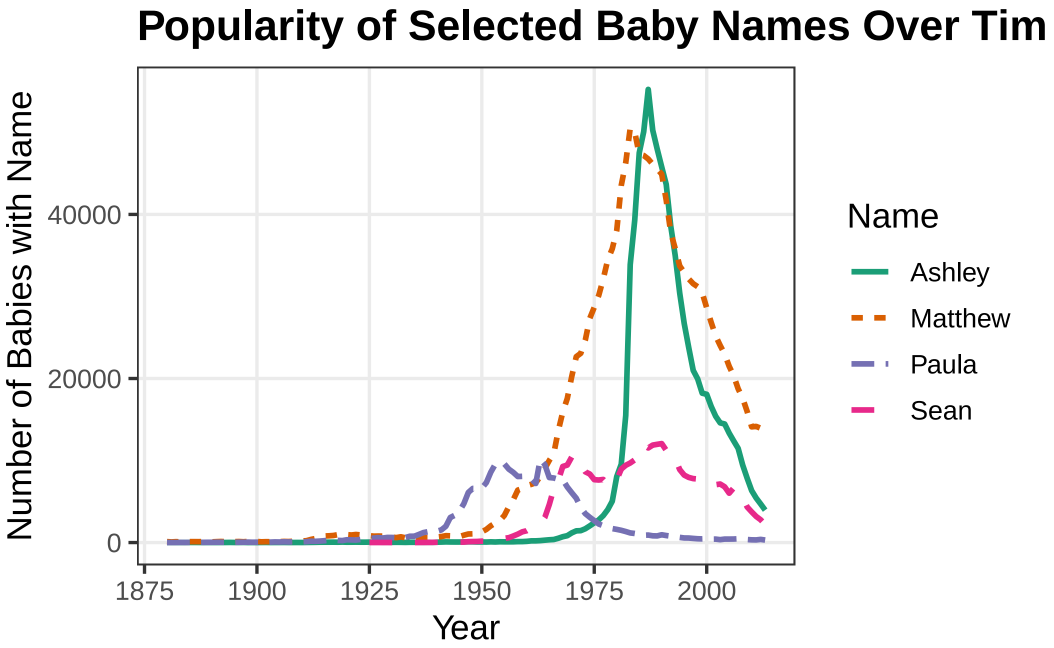
Table 1. Frequency of Sex by Rank among Air Force Warranted Officers in the June 2025 Armed Forces data.

```
# A tibble: 10 x 3
   Rank               Female  Male
   <chr>               <int> <int>
 1 Brigadier General      18    99
 2 Captain              5485 15715
 3 Colonel               569  2663
 4 First Lieutenant     2037  5045
 5 General                 0    11
 6 Lieutenant Colonel   1890  7373
 7 Lieutenant General      7    30
 8 Major                3440  9682
 9 Major General           6    63
10 Second Lieutenant    1985  5048
```

Popularity of Baby Names (Activity #13)

I chose the names Ashley, Matthew, Paula, and Sean because they represent the people closest to me, my best friends Matthew and Paula, and my twin brother Sean, and included my own name, so I was curious to see how their popularity changed over time. The line graph shows yearly counts of each name from the late 1800s to the early 2000s, with different colors and line styles to make the trends easy to compare. The name Paula peaks around the 1950s before declining, Sean rises and falls a few times throughout the mid-1900s, Matthew grows steadily and peaks around the 1980s, and Ashley rises sharply after the mid-1970s and has the highest overall peak. Overall, the visualization makes it easy to see how each name follows a very different pattern across the decades.

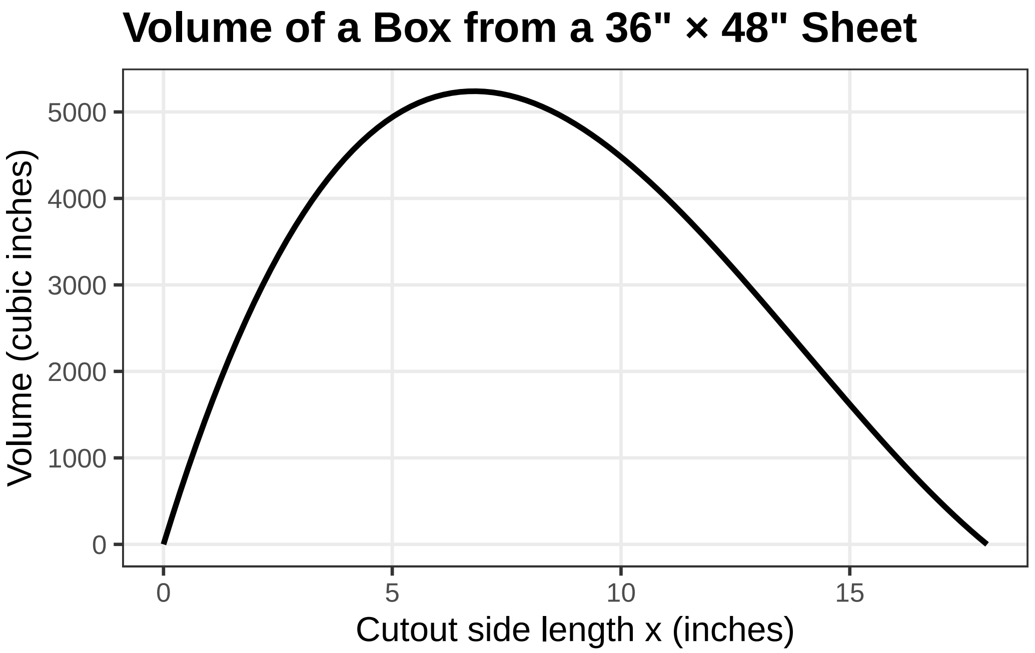
Figure 1. Popularity of Names Ashley, Matthew, Paula, and Sean Over Time.



Plotting a Mathematical Function (Activity #04)

This is the Box Problem, where you take a 36” × 48” sheet of paper, cut out squares of side length x from each corner, and fold up the sides to form an open-top box. The volume depends on how big the cutouts are, so the plot shows how the volume changes as x increases. The curve starts at zero, rises to a single peak, and then drops back down as the cutouts get too large to form a usable box. This visualization makes it easy to see where the maximum happens and how quickly the volume changes based on small differences in the cutout size.

Figure 2. Volume of an open-top box made from a 36” × 48” sheet as a function of the cutout side length.



For this 36” × 48” sheet, the maximum volume occurs when the cutout side length is approximately 6.79 inches, which produces a volume of about 5,240 cubic inches.

What You Feel You’ve Learned So Far

When I think back to the start of this course, I feel like I went from zero R knowledge to actually being able to wrangle data, clean it up, and make helpful and interesting visualizations. In the beginning, even simple things like filtering or reshaping tables felt extremely confusing, but over time everything started to click. I especially liked the visualization topics since I enjoy graphic design, and it was fun learning how to make plots that not only work but also look good and communicate data interpretations clearly to audiences. Now I feel much more confident writing reproducible code, thinking through each step, and explaining what my plots show. Overall, though still not 100% confident in my abilities, I feel like I have a much better understanding of how to work with data in R, and I hope to continue growing my R skills for the rest of the course and even after the course ends.

Code Appendix

Below is the code used to generate all results in this document.

Armed Forces Data Wrangling Code

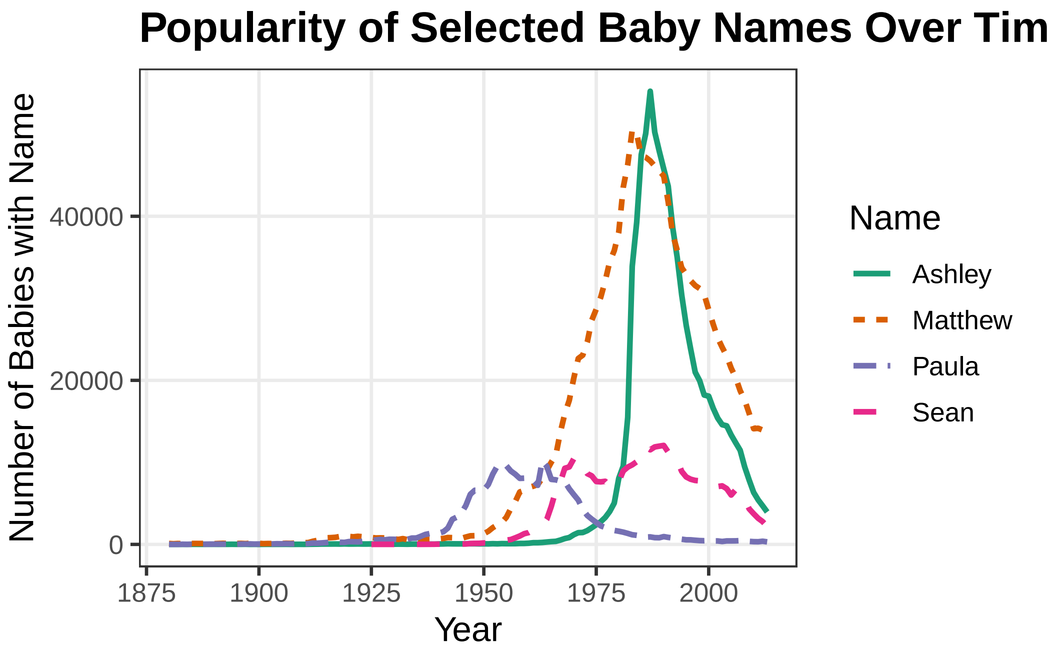
```r
# import packages
library(tidyverse)
library(rvest)
library(googlesheets4)

# scrape lookup table of pay grades and ranks
webRanks <- rvest::read_html(
  x = "https://neilhatfield.github.io/Stat184_PayGradeRanks.html"
) |>
  rvest::html_elements(css = "table") |>
  rvest::html_table()

rawRanks <- webRanks[[1]]

# clean rank table
rawRanks[1, 1] <- "Type"
rankHeaders <- rawRanks[1, ]
names(rawRanks) <- rankHeaders[1, ]

rawRanks <- rawRanks[-c(1, 26), ]

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

# load Armed Forces sheet from Google Sheets
googlesheets4::gs4_deauth()

forcesHeaders <- googlesheets4::read_sheet(
  ss = "https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qbwb_E/edi
  col_names = FALSE,
  n_max = 3
)

rawForces <- googlesheets4::read_sheet(
  ss = "https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qbwb_E/edi
  col_names = FALSE,
  skip = 3,
  n_max = 28,
  na = c("N/A*")
)

# create column names for 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

# wrangle Armed Forces data to grouped format
cleanForces <- rawForces |>
  dplyr::rename(Pay.Grade = `.Pay Grade`) |>
  dplyr::select(!dplyr::contains("Total")) |>
  dplyr::filter(
    Pay.Grade != "Total Enlisted",
    Pay.Grade != "Total Warrant Officers",
    Pay.Grade != "Total Officers",
    Pay.Grade != "Total"
  ) |>
  tidyr::pivot_longer(
    cols = !Pay.Grade,
    names_to = "Branch.Sex",
    values_to = "Frequency"
  ) |>
  tidyr::separate_wider_delim(
    cols = Branch.Sex,
    delim = ".",
    names = c("Branch", "Sex")
  )

# merge in rank names
key_forcesRanks <- dplyr::left_join(
  x = cleanForces,
  y = cleanRanks,
  by = dplyr::join_by(Pay.Grade == `Pay Grade`, Branch == Branch)
)

# expand to individual-level data
key_individualRanks <- key_forcesRanks |>
  dplyr::filter(!is.na(Frequency)) |>
  tidyr::uncount(weights = Frequency)

# subset to Air Force commissioned officers (Pay.Grade starting with "O")
airforce_officers <- key_individualRanks |>
  dplyr::mutate(Branch = trimws(Branch)) |>
  dplyr::filter(
    Branch == "Air Force",
    !is.na(Rank),
    grepl("^O", Pay.Grade)      # matches O1, O2, ..., O10
  )

# frequency table: Sex by Rank for Air Force officers
airforce_officer_table <- airforce_officers |>
  dplyr::count(Rank, Sex, name = "Count") |>
  dplyr::arrange(Rank, Sex) |>
  tidyr::pivot_wider(
    names_from = Sex,
    values_from = Count,
    values_fill = 0
  )

airforce_officer_table
```

```
# A tibble: 10 x 3
   Rank               Female  Male
   <chr>               <int> <int>
 1 Brigadier General      18    99
 2 Captain              5485 15715
 3 Colonel               569  2663
 4 First Lieutenant     2037  5045
 5 General                 0    11
 6 Lieutenant Colonel   1890  7373
 7 Lieutenant General      7    30
 8 Major                3440  9682
 9 Major General           6    63
10 Second Lieutenant    1985  5048
```

Popularity of Baby Names Code

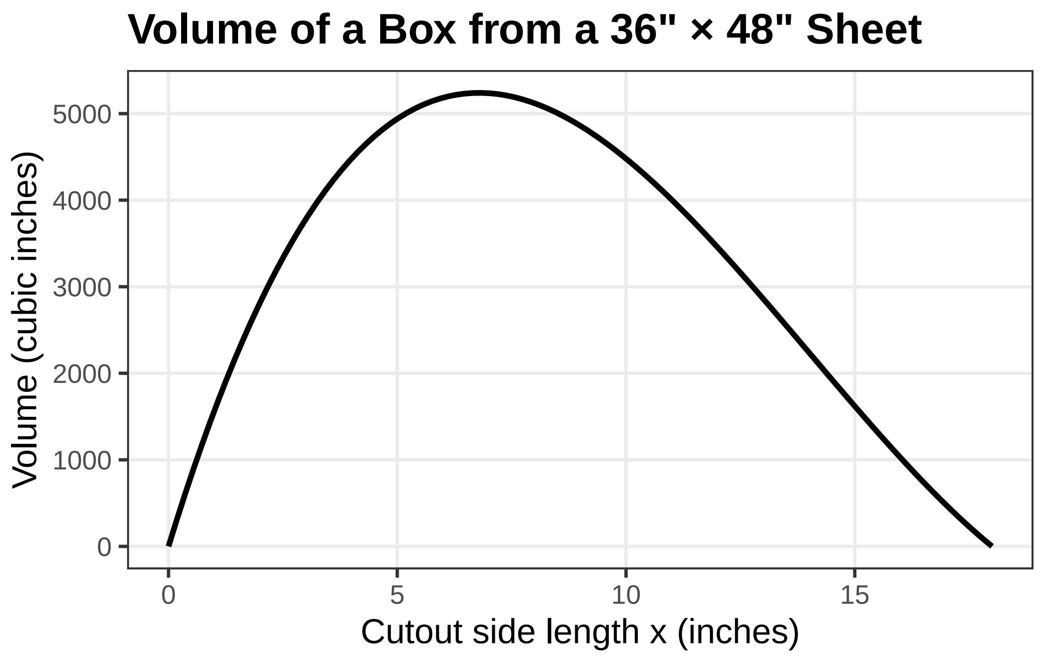
```
# load packages
library(dplyr)
library(ggplot2)
library(dcData)

# load data
```

```r
data(BabyNames, package = "dcData")

# select names
names_selected <- c("Ashley", "Matthew", "Paula", "Sean")

# filter to selected names and summarize by year
baby_yr <- BabyNames |>
  dplyr::filter(name %in% names_selected) |>
  dplyr::mutate(year = as.integer(year)) |>
  dplyr::group_by(name, year) |>
  dplyr::summarise(count = sum(count), .groups = "drop") |>
  dplyr::arrange(name, year)

# create time series plot
baby_plot <- ggplot(
  data = baby_yr,
  mapping = aes(
    x = year,
    y = count,
    color = name,
    linetype = name
  )
) +
  geom_line(linewidth = 1) +
  # colorblind-friendly palette
  scale_color_brewer(palette = "Dark2") +
  labs(
    title = "Popularity of Selected Baby Names Over Time",
    x = "Year",
    y = "Number of Babies with Name",
    color = "Name",
    linetype = "Name",
    alt = "Line chart shows the number of babies given the names Ashley, Matthew, Paula, and Se
  ) +
  theme_bw(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold"),
    panel.grid.minor = element_blank()
  )

baby_plot
```



Box Problem Code

```
# load packages
library(ggplot2)

# volume function for 36" × 48" sheet
getVolume <- function(sideLength, paperLength = 48, paperWidth = 36) {
  sideLength * (paperWidth - 2 * sideLength) * (paperLength - 2 * sideLength)
}

volume_function <- function(x) {
  getVolume(sideLength = x)
}

# volume plot
box_plot <- ggplot(data = data.frame(x = c(0, 18)), aes(x = x)) +
  stat_function(
    fun = volume_function,
    linewidth = 1
  ) +
  labs(
    title = "Volume of a Box from a 36\" × 48\" Sheet",
    x = "Cutout side length x (inches)",
    y = "Volume (cubic inches)",
    alt = "Curve showing the volume of an open-top box made from a 36 by 48 inch sheet as a fu
  ) +
  theme_bw(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold"),
    panel.grid.minor = element_blank()
  )

# find x for max volume
max_result <- optimize(
  f = volume_function,
  interval = c(0, 18),
  maximum = TRUE
)

x_max <- max_result$maximum                # optimal cutout size
volume_max <- max_result$objective         # maximum volume

box_plot
```



```
x_max
```

```
[1] 6.788902
```

```
volume_max
```

```
[1] 5239.819
```