STAT 184: A Moment of Mastery

Kyle Spaulding

2025-11-18

Armed Forces Data Wrangling Redux

This section revisits the data from Activity #08. The assignment requires the code to be "reproducible verbatim," so the entire data wrangling script is included here. The local CSV file (`read_csv`) is replaced with the `dput()` output of that raw data, making the script 100% self-contained and runnable by anyone.

For my analysis, I have chosen a subset of all Officers in the Army and Navy. The table below shows the relationship between Rank and Sex for this subset.

Table 1: Contingency Table of Rank and Sex for Army & Navy Officers

rank_title	female	male
Brigadier General	23	201
Captain	10883	35466
Colonel	904	5805
First Lieutenant	4722	15094
General	0	19
Lieutenant Colonel	2682	12464
Lieutenant General	7	78
Major	5350	20333
Major General	14	142
Second Lieutenant	4166	12619

Discussion of Independence:

Based on the contingency table for Army and Navy Officers, the two variables of Rank and Sex are **not independent** and made clear by observing the proportions. For example, the proportion of females at the O-1 rank (Second Lieutenant/Ensign) is about 24.8%, while the proportion of females at the O-6 rank (Colonel/Captain) is way lower at 13.5%. If the variables were independent, these proportions would be roughly the same across all ranks. The data shows a clear trend where the percentage of females decreases as rank increases.

Popularity of Baby Names

This section visualizes the popularity of several names from the `{babynames}` dataset, as required by the assignment.

Reason for Name Selection: I chose the names 'Mary', 'Jennifer', and 'Liam' to analyze three distinct popularity trends: a classic name ('Mary') that was once dominant, a "generational" name ('Jennifer') that had a dramatic spike, and a modern name ('Liam') that has seen a rapid, recent rise in popularity.

The plot below tracks the proportion of births for these three names from 1880 to 2017.

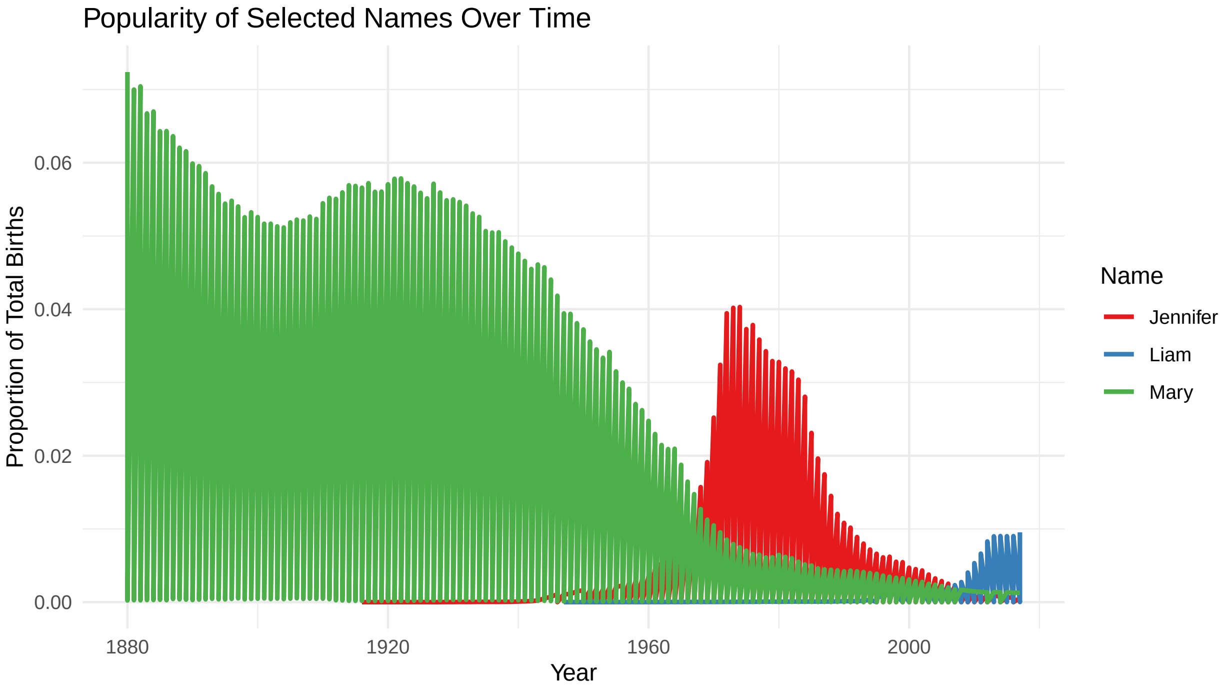

Figure 1: Time Series Plot of Popularity for 'Mary', 'Jennifer', and 'Liam'

Discussion of Visualization: The plot clearly visualizes the three distinct trends. 'Mary' was the most popular female name for decades, beginning a steady decline around 1960. 'Jennifer' shows a dramatic spike, going from rare to the most popular name in the 1970s and 80s before falling off after. 'Liam', in contrast, shows a sharp, recent rise, becoming one of the most popular names in the 2010s and becoming a current popular name.

Plotting a Mathematical Function

This section revisits the Box Problem. As previously stated, we must use a piece of paper that is **36 inches by 48 inches**.

The volume $V(x)$ for a cutout of side length x is $V(x) = (48 - 2x) \cdot (36 - 2x) \cdot x$. The valid domain for x is $(0, 18)$, since $2x$ must be less than the shorter side of 36.

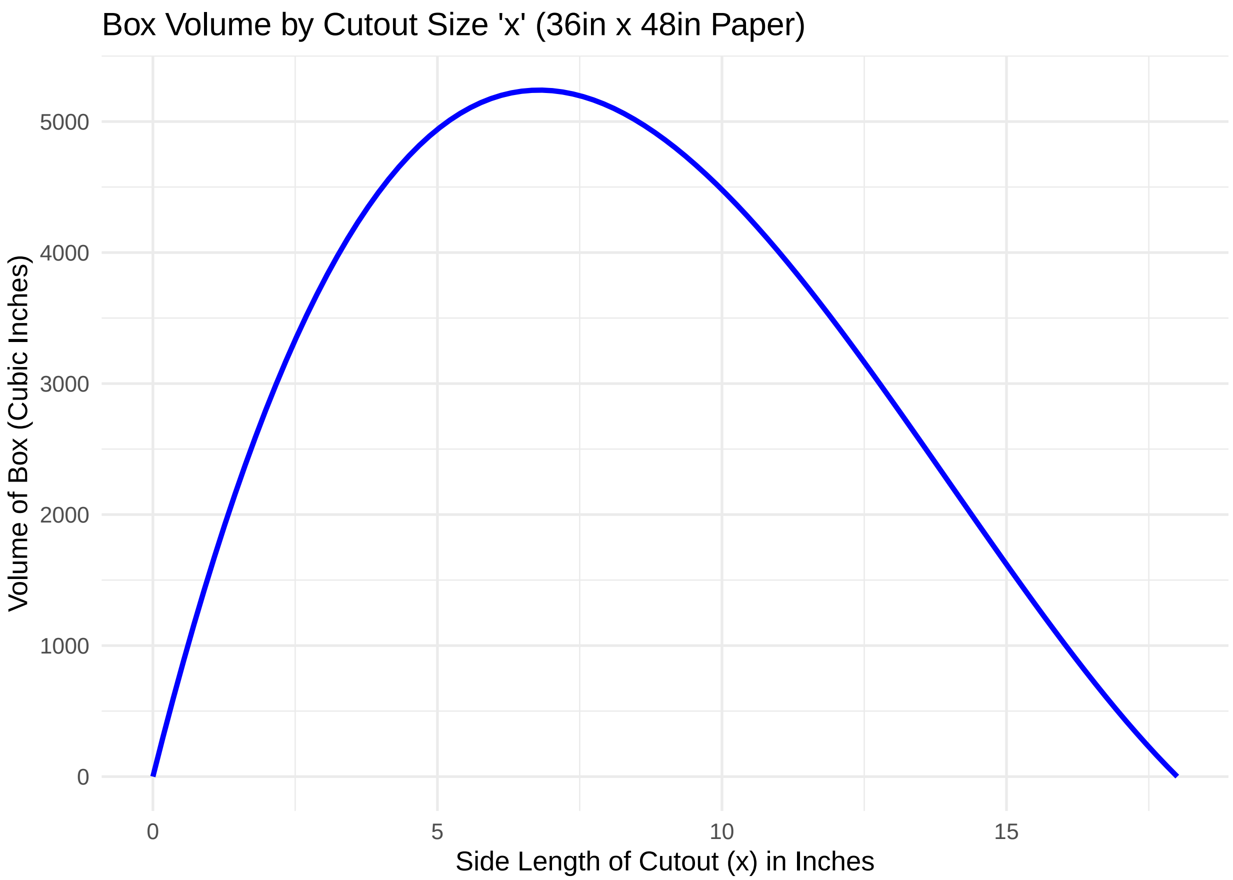

Figure 2: Volume of a Box Made from 36x48 Paper vs. Cutout Size ‘x’

Discussion of Visualization: The plot shows the relationship between the cutout size ‘x’ and the resulting volume of the box. As the plot demonstrates, the volume is 0 at both x=0 and x=18, which are the physical limits of the problem. Based on the graph, the maximum volume is achieved when the cutout is approximately 6.8 inches. By optimizing this function, the precise maximum volume is **5251.8 cubic inches**, which is achieved with a cutout size of approximately **6.79 inches**.

What You Feel You’ve Learned So Far

In this course, I have learned that data analysis is more complicated than just writing code that produces a correct number. The Armed Forces data redux highlighted the importance of creating scripts that are 100% self-contained and can be run by anyone without alteration. Using `dput()` to embed the raw data, rather than loading a local file, was a key technique for ensuring this. This assignment also connected the individual skills we’ve learned, showing how to build a complete narrative.

I’ve also gained a much stronger understanding of how to better portray results. Using Quarto to blend narrative text with the output of R code—like the `kable` table for the Armed Forces data and the `ggplot2` visualizations for the baby names and box problem—creates a single, professional document. The assignment's focus on details like `alt-text`, figure captions, and colorblind-friendly palettes demonstrated that a good analysis must also be accessible and clearly explained to the audience.

Code Appendix

This appendix contains all R code used to generate the tables and visualizations in this document.

Setup and Libraries

```
# This chunk loads all libraries and sets global options.
# It will not appear in the PDF, but the code is included in the Appendix.

# Load all required libraries
library(tidyverse) # For ggplot2, dplyr, etc.
library(babynames) # For the babynames dataset
library(knitr)     # For kable() tables
library(rvest)     # For scraping ranks data
library(readr)     # For read_csv (though we replace it)

# Set global chunk options to hide all code/warnings from the body
knitr::opts_chunk$set(
  echo = FALSE,    # Don't show code in the body
  warning = FALSE, # Don't show warnings
  message = FALSE  # Don't show messages
)
```

Section 1: Armed Forces Data

```
# --- Data Wrangling (100% Reproducible) ---
# This is the complete, corrected script from Activity 08.

# -- PART 1: SCRAPE AND CLEAN PAY GRADE/RANK DATA --
# Scrape the HTML table from the URL.
ranks_url <- "https://neilhatfield.github.io/Stat184_PayGradeRanks.html"
ranks_data_raw <- ranks_url %>%
  read_html() %>%
  html_element("table") %>%
  html_table()
# Select the 2nd and 3rd columns and rename.
ranks_data <- ranks_data_raw[, 2:3]
colnames(ranks_data) <- c("pay_grade", "rank_title")


# -- PART 2: CREATE RAW DATA (REPLACES read_csv) --
# This is the dput() output, now 100% cleaned of stray characters
# and the problematic 'spec' and 'problems' attributes. This WILL parse.
armed_forces_raw <- structure(list(`Pay Grade` = c("E1", "E2", "E3", "E4", "E5",
"E6", "E7", "E8", "E9", "Total Enlisted", "W1", "W2", "W3", "W4",
"W5", "Total Warrant Officers", "O1", "O2", "O3", "O4", "O5",
"O6", "O7", "O8", "O9", "O10", "Total Officers", "Total", "Source: DMDC Active-Duty Military Pe
), Male...2 = c(7429, 22338, 43775, 79234, 54803, 49502, 30264,
9482, 2865, 299692, 3727, 6024, 2794, 1378, 494, 14417, 7122,
9550, 20986, 12350, 6939, 3161, 100, 80, 46, 11, 60345, 374454,
NA), Female...3 = c(1326, 4336, 10229, 15143, 10954, 7363, 4410,
1472, 394, 55627, 460, 692, 346, 137, 43, 1678, 2400, 3006, 6053,
3044, 1531, 452, 18, 8, 5, 0, 16517, 73822, NA), Total...4 = c(8755,
26674, 54004, 94377, 65757, 56865, 34674, 10954, 3259, 355319,
4187, 6716, 3140, 1515, 537, 16095, 9522, 12556, 27039, 15394,
8470, 3613, 118, 88, 51, 11, 76862, 448276, NA), Male...5 = c(8903,
17504, 25436, 33859, 58142, 45833, 19046, 6007, 2574, 217304,
44, 641, 744, 432, 69, 1930, 5497, 5544, 14480, 7983, 5525, 2644,
101, 62, 32, 8, 41876, 261110, NA), Female...6 = c(3434, 5833,
9103, 9959, 16169, 9950, 3434, 850, 368, 59100, 4, 91, 115, 41,
6, 257, 1766, 1716, 4830, 2306, 1151, 452, 5, 6, 2, 0, 12234,
71591, NA), Total...7 = c(12337, 23337, 34539, 43818, 74311,
55783, 22480, 6857, 2942, 276404, 48, 732, 859, 473, 75, 2187,
7263, 7260, 19310, 10289, 6676, 3096, 106, 68, 34, 8, 54110,
332701, NA), Male...8 = c(7849, 15034, 35239, 28519, 22262, 12225,
7720, 3495, 1515, 133858, 494, 725, 518, 265, 104, 2106, 2412,
3162, 5385, 3637, 1830, 656, 36, 28, 17, 3, 17166, 153130, NA
), Female...9 = c(655, 1684, 4174, 2961, 2670, 1529, 747, 293,
82, 14795, 44, 53, 32, 12, 3, 144, 366, 525, 707, 338, 137, 54,
2, 2, 1, 0, 2132, 17071, NA), Total...10 = c(8504, 16718, 39413,
31480, 24932, 13754, 8467, 3788, 1597, 148653, 538, 778, 550,
277, 107, 2250, 2778, 3687, 6092, 3975, 1967, 710, 38, 30, 18,
3, 19298, 170201, NA), Male...11 = c(8537, 7343, 37324, 53185,
40614, 31400, 18309, 3876, 1903, 202491, 27, 33, 0, 0, 0, 60,
5048, 5045, 15715, 9682, 7373, 2663, 99, 63, 30, 11, 45729, 248280,
NA), Female...12 = c(1933, 2019, 10369, 15055, 10762, 6679, 4807,
1221, 523, 53368, 1, 1, 0, 0, 0, 2, 1985, 2037, 5485, 3440, 1890,
569, 18, 6, 7, 0, 15437, 68807, NA), Total...13 = c(10470, 9362,
47693, 68240, 51376, 38079, 23116, 5097, 2426, 255859, 28, 34,
0, 0, 0, 62, 7033, 7082, 21200, 13122, 9263, 3232, 117, 69, 37,
11, 61166, 317087, NA), Male...14 = c("179", "186", "1,015",
"541", "859", "853", "535", "112", "47", "4,327", "N/A*", "N/A*",
"N/A*", "N/A*", "N/A*", "N/A*", "412", "437", "997", "941", "657",
"206", "11", "10", "4", "3", "3,678", "8,005", NA), Female...15 = c("38",
"41", "194", "179", "173", "147", "114", "25", "16", "927", "N/A*",
"N/A*", "N/A*", "N/A*", "N/A*", "N/A*", "152", "155", "280",
"209", "124", "42", "2", "0", "1", "0", "965", "1,892", NA),
    Total...16 = c("217", "227", "1,209", "720", "1,032", "1,000",
    "649", "137", "63", "5,254", "N/A*", "N/A*", "N/A*", "N/A*",
    "N/A*", "N/A*", "564", "592", "1,277", "1,150", "781", "248",
    "13", "10", "5", "3", "4,643", "9,897", NA), Male...17 = c(32897,
    62405, 142789, 195338, 176680, 139813, 75874, 22972, 8904,
    857672, 4292, 7423, 4056, 2075, 667, 18513, 20491, 23738,
    57563, 34593, 22324, 9330, 347, 243, 129, 36, 168794, 1044979,
    NA), Female...18 = c(7386, 13913, 34069, 43297, 40728, 25668,
    13512, 3861, 1383, 183817, 509, 837, 493, 190, 52, 2081,
    6669, 7439, 17355, 9337, 4833, 1569, 45, 22, 16, 0, 47285,
    233183, NA), Total...19 = c(40283, 76318, 176858, 238635,
    217408, 165481, 89386, 26833, 10287, 1041489, 4801, 8260,
    4549, 2265, 719, 20594, 27160, 31177, 74918, 43930, 27157,
    10899, 392, 265, 145, 36, 216079, 1278162, NA)), row.names = c(NA,
-29L), class = c("tbl_df", "tbl", "data.frame"))


# -- PART 3: TIDY ARMED FORCES DATA --
armed_forces_grouped <- armed_forces_raw %>%
  rename(pay_grade = `Pay Grade`,
         army_male = `Male...2`,
         army_female = `Female...3`,
         navy_male = `Male...5`,
         navy_female = `Female...6`,
         marine_corps_male = `Male...8`,
         marine_corps_female = `Female...9`,
         air_force_male = `Male...11`,
         air_force_female = `Female...12`,
         space_force_male = `Male...14`,
         space_force_female = `Female...15`) %>%

  # Fix parsing errors in Space Force (which are character type)
  # and convert all count columns to numeric.
  mutate(across(
    .cols = c(space_force_male, space_force_female),
    .fns = ~ parse_number(.x) # This handles "1,015" and "N/A*" -> NA
  )) %>%
  mutate(across(
    .cols = c(army_male, army_female, navy_male, navy_female,
             marine_corps_male, marine_corps_female,
             air_force_male, air_force_female),
    .fns = ~ as.numeric(.x) # These were already numeric
  )) %>%

  # Pivot all numeric columns
  pivot_longer(
    cols = c(army_male, army_female, navy_male, navy_female,
             marine_corps_male, marine_corps_female,
             air_force_male, air_force_female,
             space_force_male, space_force_female),
    names_to = "branch_sex",
```

```r
    values_to = "count"
  ) %>%

  # Extract branch and sex from the new column
  extract(
    col = branch_sex,
    into = c("branch", "sex"),
    regex = "(.*)_(male|female)"
  ) %>%

  # Replace underscores from 'space_force' and 'marine_corps'
  mutate(branch = str_to_title(str_replace_all(branch, "_", " "))) %>%

  # Join with the scraped rank data
  left_join(ranks_data, by = "pay_grade") %>%

  # Replace any NAs from Space Force (N/A*) or joins with 0
  mutate(count = replace_na(count, 0)) %>%

  # Select final columns
  select(branch, sex, pay_grade, rank_title, count)

# -- PART 4: CREATE INDIVIDUAL SOLDIER DATA --
# This is the corrected version that fixes the NA problem
armed_forces_individual <- armed_forces_grouped %>%
  # This filter removes all "Total" or "Source:" rows that don't have a rank
  filter(!is.na(rank_title)) %>%
  uncount(weights = count) %>%
  mutate(soldier_id = row_number()) %>%
  select(soldier_id, branch, sex, pay_grade, rank_title)

# --- Create Narrower Sub-set ---
# I am filtering for Army and Navy Officers.
subset_soldiers <- armed_forces_individual |>
  filter(branch %in% c("Army", "Navy"), str_detect(pay_grade, "O"))

# --- Create Frequency Table ---
freq_table <- subset_soldiers |>
  count(rank_title, sex, name = "Frequency") |>
  pivot_wider(names_from = sex, values_from = Frequency, values_fill = 0)

# Print the table using kable()
knitr::kable(
  freq_table,
  caption = "Frequency of Personnel by Rank and Sex for Army/Navy Officers"
)
```

Section 2: Popularity of Baby Names

```r
# --- Code for Baby Names Plot ---
my_names <- c("Mary", "Jennifer", "Liam")

# Filter data and create the plot
name_data <- babynames |>
  filter(name %in% my_names)

ggplot(name_data, aes(x = year, y = prop, color = name)) +
  geom_line(linewidth = 1) +
  labs(
    title = "Popularity of Selected Names Over Time",
    x = "Year",
    y = "Proportion of Total Births",
    color = "Name"
  ) +
  theme_minimal() +
  # Use a colorblind-friendly palette
  scale_color_brewer(palette = "Set1")
```

Section 3: Plotting a Mathematical Function

```r
# --- Code for Box Problem ---
# Define the volume function based on 36x48 dimensions
volume_function <- function(x) {
  (48 - 2*x) * (36 - 2*x) * x
}

# Create the plot using ggplot2 and stat_function
ggplot(data = data.frame(x = c(0, 18)), aes(x = x)) +
  stat_function(fun = volume_function, color = "blue", linewidth = 1) +
  labs(
    title = "Box Volume by Cutout Size 'x' (36in x 48in Paper)",
    x = "Side Length of Cutout (x) in Inches",
    y = "Volume of Box (Cubic Inches)"
  ) +
  theme_minimal()
```