Multi Data Visualization Comparison

Karol Bucon

2025-11-17

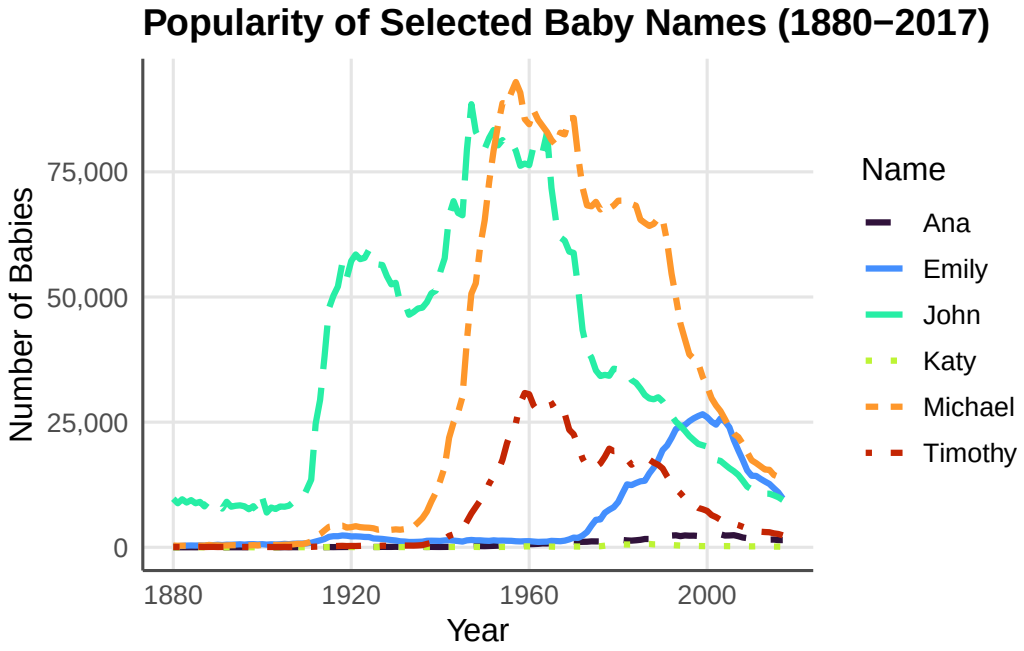
Sex–Rank Relationship for Enlisted Members of the U.S. Marine Corps

Table 1: Two-way frequency table showing the distribution of Marine Corps enlisted personnel by sex and rank.

Rank	Male	Female	Total
E2	14,688	1,604	16,292
E3	35,047	3,787	38,834
E4	28,946	2,942	31,888
E5	21,481	2,723	24,204
E6	11,667	1,370	13,037
E7	8,191	760	8,951
E8	3,559	275	3,834
E9	1,518	83	1,601
Total	**125,097**	**13,544**	**138,641**

Interpretation:
This table shows the distribution of 138,641 enlisted Marines across nine different pay-grades separated by **sex** (columns) and **rank** (rows). Furthermore, we can see that most of the Marine Corps is made up of males (125,097) compared to the much smaller female count (13,544). Another detail we can see is that the higher the rank the larger the difference is between males and females in the Marine Corps. For example, in pay-grade **E9** the male count comes in at **1518** while the female count comes in at only **83**. This really shows the contrast between the two genders. Lastly, discussing whether sex and rank are independent is quite simple. If these two metrics were independent then the female count would be consistent throughout instead, we see that the higher the rank the lower the female count is. Meaning that this is not independent.

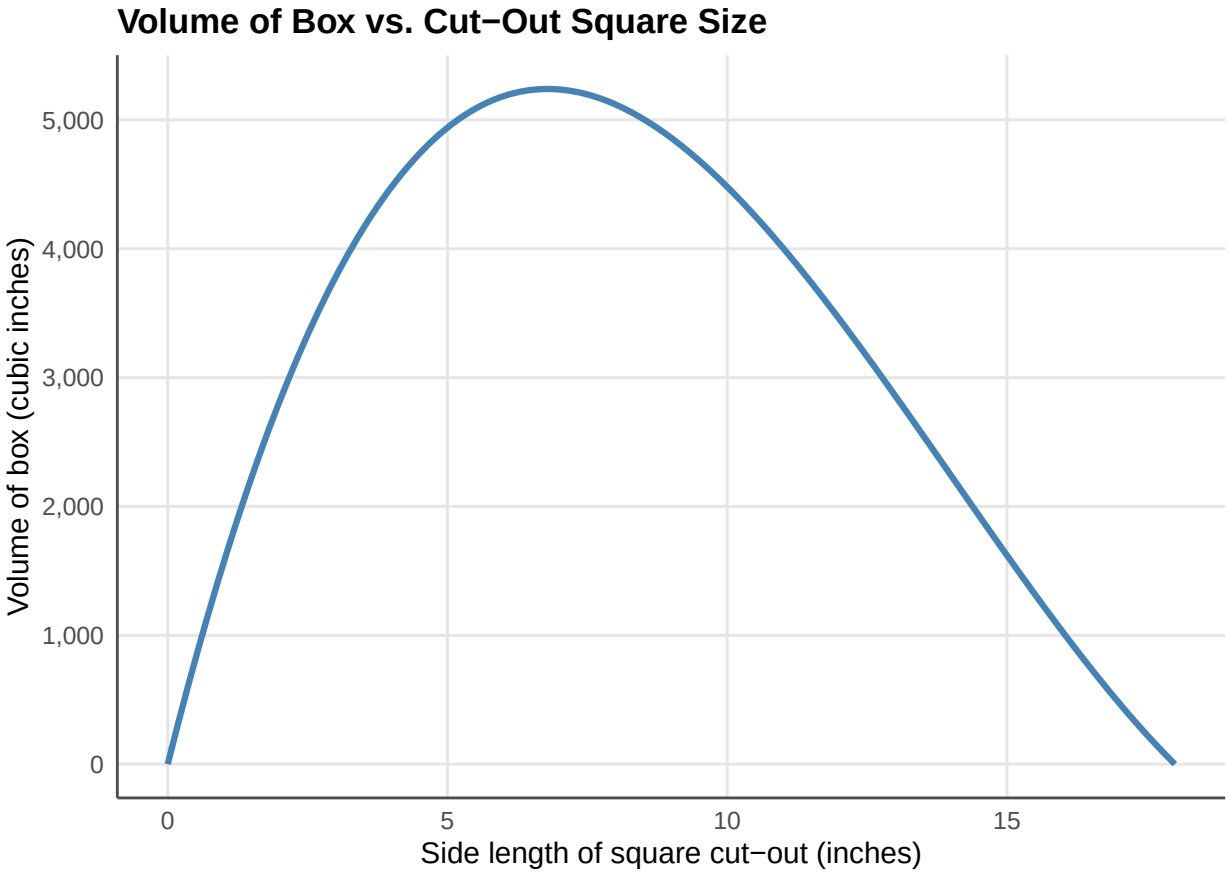
Figure 1: Popularity of Selected Baby Names (1880–2017). Each line shows total births per year for each given name, for both male and female names.



Interpretation:
This plot visualizes six baby names (3 male, 3 female) tracked from **1880** to **2017**. Each name is represented by its own line and color for easier readability. This plot reveals to the reader how name popularity increased and decreased throughout the years. Starting with the name **John** we can see that it peaked around **60,000** births from the **20s** to **40s** before seeing a decline. Looking at the name **Michael** we see a peak of **90,000** births (highest out of male names) in the **50s** and **60s**. Next, **Timothy** peaks in the **60s** at roughly **25,000** births before declining. Moving onto the female names starting with **Emily** this name sees **25,000** births by the **90s** to **2000** before declining. Lastly, **Ana** and **Katy** remained pretty rare throughout the years never really exceeding **5,000** births annually. Overall, all of these names see a drop-off just after the year 2000 showing that parents are choosing much more different or "rare" names.

Box Problem

Figure 2: Volume of an open-top box formed from a 36 × 48 inch sheet as a function of the square cut-out side length. The volume reaches a max at approximately x = 6 inches.



Interpretation:
This visualization displays a smooth curve representing the volume function of an open-top box. This is because equal squares are cut from each corner of a 36 × 48 inch sheet and folded up. The x axis shows the side length of square cut out and the y shows Volume of box The curve rises to a peak then goes back toward zero. Using the `stat_function` the optimal value is apparent. From this visualization we can identify that the **maximum volume occurs at approximately x = 6 inches** with volume being around **5,300 cubic inches**. To maximize capacity 6-inch squares should be cut from each corner. Overall, what we can learn from this visualization is that the curve tells us how much too cut and whether it's too much or too little.

What have I learned?

This course has taught me many things related to data wrangling and visualization using R that have seriously improved the way I write and look at code. What helped me the most were the principles of effective graphics from Tufte and Kosslyn. I never really thought specific graphics and various other visuals were as important until taking this course. It opened up my mind to the many different ways one can organize and display data creatively and effectively. Being able to display data is one thing but doing it in a way many people even those who aren't familiar with the concept or data is a good tool/strength to have. After reading through the different principles I can now go onto create meaningful visuals as I find them essential to learning especially in the field of data science. In addition, completing these assignments showed me how effective these principles are and how specific data may not fit every type of visual. Overall, principles like the ones discussed as well as other rules/tips significantly improved my visuals and expanded my view on visuals themselves.

Code Appendix

Setup and Libraries

```
library(tidyverse)  # includes readr for read_csv
library(janitor)
library(knitr)
library(kableExtra)
library(babynames)
library(viridis)
library(scales)
```

Load and Process Armed Forces Data

```
# Read the CSV file - force all columns to character to avoid type conflicts
rawForces <- read_csv(
  '/Users/karolbucon/Downloads/US Armed Forces (6_2024) - Sheet1.csv',
  skip = 3,
  n_max = 28,
  col_types = cols(.default = "c")  # Read everything as character
)

# Get headers from the CSV
forcesHeaders <- read_csv(
  '/Users/karolbucon/Downloads/US Armed Forces (6_2024) - Sheet1.csv',
  col_names = FALSE,
  n_max = 3,
  col_types = cols(.default = "c")
)

# Create column names
branchNames <- rep(
  x = c("Army", "Navy", "Marine Corps", "Air Force", "Space Force", "Total"),
  each = 3
)

tempHeaders <- paste(
  c("", branchNames),
  as.character(forcesHeaders[3,]),
  sep = "."
)

names(rawForces) <- tempHeaders
```

```r
# Clean the data
cleanForces <- rawForces %>%
  rename(Pay.Grade = `.Pay Grade`) %>%
  dplyr::select(!contains("Total")) %>%
  filter(!is.na(Pay.Grade),
         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)
  ) %>%
  filter(!is.na(Frequency))
```

Filter Marine Corps Enlisted Data

```r
# Filter for Marine Corps Enlisted (E1 to E9)
marine_enlisted <- cleanForces %>%
  filter(Branch == "Marine Corps", grepl("^E[1-9]", Pay.Grade)) %>%
  rename(Rank = Pay.Grade)
```

Create Marine Corps Frequency Table

```r
# Create two-way frequency table using the Frequency values
marine_enlisted_table <- marine_enlisted %>%
  select(Rank, Sex, Frequency) %>%
  pivot_wider(names_from = Sex, values_from = Frequency, values_fill = 0)

# Add row totals
marine_enlisted_table <- marine_enlisted_table %>%
  mutate(Total = rowSums(select(., -Rank)))
```

```r
# Calculate column totals
col_totals <- marine_enlisted_table %>%
  summarise(across(where(is.numeric), sum)) %>%
  mutate(Rank = "Total") %>%
  select(Rank, everything())

# Bind the total row
marine_enlisted_table <- bind_rows(marine_enlisted_table, col_totals)

# Display formatted table
knitr::kable(
  marine_enlisted_table,
  align  = c("l", rep("r", ncol(marine_enlisted_table) - 1)),
  digits = 0,
  format.args = list(big.mark = ",")
) %>%
  kableExtra::kable_classic(full_width = FALSE) %>%
  kableExtra::row_spec(nrow(marine_enlisted_table), bold = TRUE)
```

Process Baby Names Data

```r
# Load babynames data (already loaded in setup)
# Filter for selected names, combining both sexes
selected_names <- c("Katy", "Ana", "Emily", "John", "Timothy", "Michael")

babynames_filtered <- babynames %>%
  filter(name %in% selected_names) %>%
  group_by(year, name) %>%
  summarise(total_births = sum(n), .groups = "drop")
```

Create Baby Names Visualization

```r
ggplot(data = babynames_filtered, aes(x = year, y = total_births,
                                      color = name, linetype = name)) +
  geom_line(linewidth = 1.1) +
  scale_color_viridis_d(option = "turbo", end = 0.9) +
  scale_linetype_manual(values = c("Katy" = "dotted",
                                   "Ana" = "dashed",
                                   "Emily" = "solid",
                                   "John" = "longdash",
                                   "Timothy" = "dotdash",
                                   "Michael" = "twodash")) +
  scale_y_continuous(labels = scales::comma) +
```

```
  labs(
    title    = "Popularity of Selected Baby Names (1880-2017)",
    x        = "Year",
    y        = "Number of Babies",
    color    = "Name",
    linetype = "Name"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.minor = element_blank(),
    plot.title       = element_text(face = "bold", size = 14),
    legend.position  = "right",
    panel.grid.major = element_line(color = "gray90"),
    axis.line        = element_line(color = "gray30")
  )
```

Define Box Volume Function

```
# Define the volume function for a box made from a 36 × 48 inch sheet
# Arguments:
#   x = side length of the square cut from each corner (in inches)
# Returns:
#   volume = volume of the resulting open-top box (in cubic inches)
volume_box <- function(x) {
  length <- 48 - 2 * x  # Length after cutting corners
  width <- 36 - 2 * x   # Width after cutting corners
  height <- x           # Height is the cut-out size
  volume <- length * width * height
  return(volume)
}
```

Create Box Volume Plot

```
# Create the plot using ggplot2 with stat_function
ggplot(data = data.frame(x = c(0, 18)), mapping = aes(x = x)) +
  stat_function(
    fun = volume_box,
    linewidth = 1.2,
    color = "steelblue"
  ) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Volume of Box vs. Cut-Out Square Size",
```

```
  x = "Side length of square cut-out (inches)",
  y = "Volume of box (cubic inches)"
) +
theme_minimal(base_size = 12) +
theme(
  panel.grid.minor = element_blank(),
  plot.title = element_text(face = "bold", size = 14),
  panel.grid.major = element_line(color = "gray90"),
  axis.line = element_line(color = "gray30")
)
```