

Activity 14 - Quarto Document

STAT 184

Jacob Koewler

2025-11-07

1 Armed Forces Data Wrangling

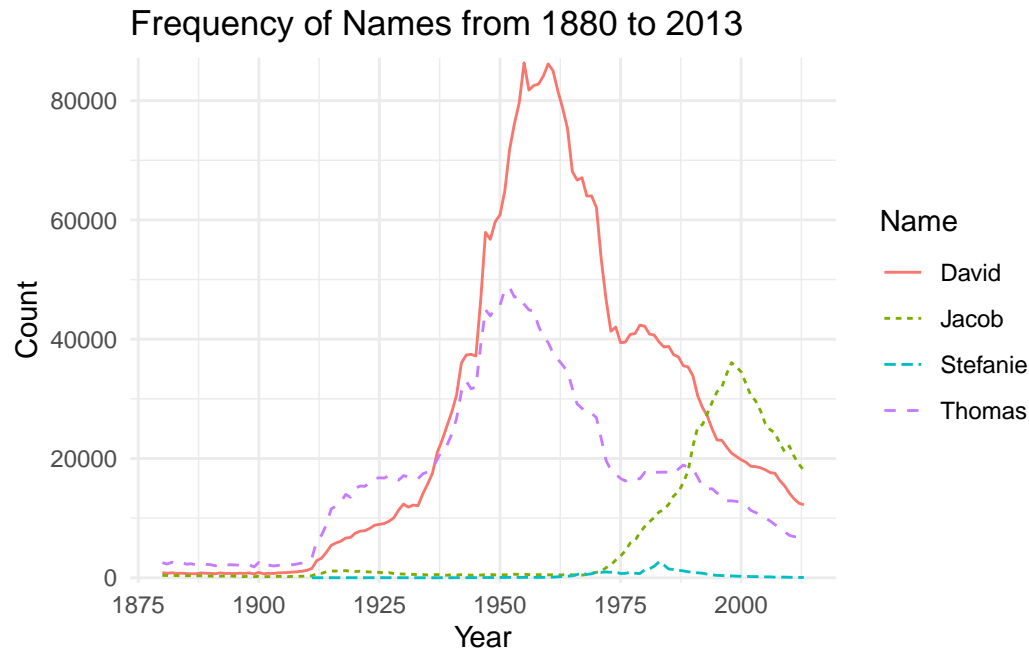
This is code used to wrangle [this armed forces data](#) into usable data for making a Frequency Table. We then created a frequency table comparing Sex and Rank for Officers within the US Air Force.

Sex	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Female	1985	2037	5485	3440	1890	569	18	6	7	0
Male	5048	5045	15715	9682	7373	2663	99	63	30	11

This data visualization shows the differences in Officer rank for men and women in the US Air Force. We can see that women make up about one in four service members until the rank of Lieutenant Colonel, where it drops to one in five. Once we reach the higher commissioned officer ranks, it drops again, with only one in twelve Major Generals being women. However, 18% of Lieutenant Generals are women, changing the existing trends. There are currently no women in the Air Force with the rank of General.

2 Popularity of Baby Names

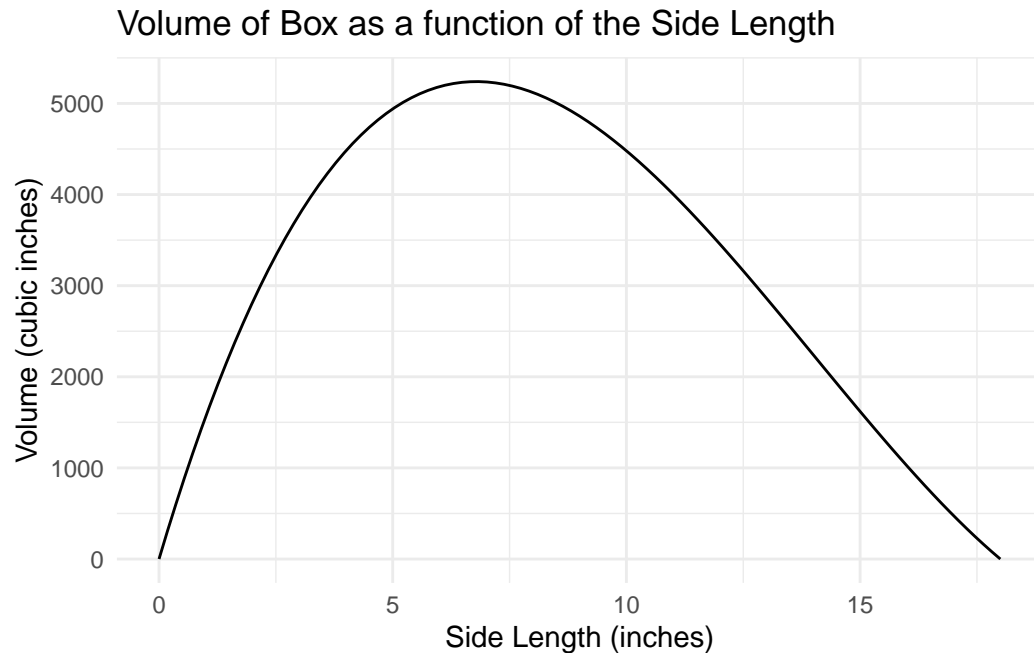
Here, we used code to wrangle the babynames data until we got just the names we wanted. I choose the names of my immediate family, those being David and Stefanie (my parents), Jacob (myself), and Thomas (my brother).



This data visualization shows the frequency of first names of my immediate family from 1880 to 2013 for a data set of baby names. The x-axis is in years, and is from 1880 to 2013. The y-axis is counts, and is from 0 to 86000. Overall, David is the most popular name, and Stefanie is the least popular name. Thomas was the most popular name from 1880 to 1937. David was then the most popular name from 1937 to 1990. Finally, Jacob is the most popular name from 1990 to 2013.

3 Plotting a Mathematical Function

Here, we used code to create a function to find the maximum volume of a box from a flat sheet with the dimensions of 36x48 inches.



This data visualization shows the maximum volume of a box with varying side lengths. The side lengths range from 0.01 to 17.99 inches, and the volume ranges from 0.01 to 5239.7. Overall, the volume increases from 0 to about 7, where it reaches a local maximum. From about 7 to 18, the volume decreases over time. The maximized volume of the box occurs at the maximum of 5239.7, which corresponds to a side length of approximately 6.76.

4 What I've Learned So Far

Overall, I am pretty satisfied with the amount I have learned in this class and about R and coding in general. This is the first coding class I have ever taken. My coding experience before this class consisted of less than an hour in python. Through many trials and tribulations, I now understand the basics of coding and how to create better visualizations. I also learned how to create Function, which were especially difficult for me to understand, but talking through them with my father helped tremendously. Through this class, I have also learned a lot of good habits for coding in general, like planning before you start to code and also to add labels to each section of code to keep it organized. Altogether, I am fairly satisfied with where I am right now and I feel like I am in a good spot to improve in the future.

5 Code Appendix:

```
#| label: tbl-frequency-armed-forces
#| tbl-cap: "Frequency Table comparing Sex and Rank for Officers within the US Air Force"
#| tbl-alt: "Frequency Table comparing Sex and Rank for Officers within the US Air Force"
```

```

# Step 1: Load Packages ----
library(tidyverse)
library(rvest)
library(google sheets4)
library(janitor)
library(knitr)

# Step 2: Scrape Rank Data ----
JDK_Full_Raw_Ranks <- read_html("https://neilhatfield.github.io/Stat184_PayGradeRanks.html") %>%
  html_elements(css = "table") %>%
  html_table()

JDK_Raw_Ranks <- JDK_Full_Raw_Ranks[[1]]

# Step 3: Wrangle Rank Data ----
JDK_Raw_Ranks[1, 1] <- "Type"
JDK_Rank_Headers <- JDK_Raw_Ranks[1, ]
names(JDK_Raw_Ranks) <- JDK_Rank_Headers[1,]
JDK_Raw_Ranks <- JDK_Raw_Ranks[-c(1, 26),]

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

# Step 4: Load Armed Forces Data ----
gs4_deauth()
JDK_Forces_Headers <- read_sheet(
  ss = "https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qbwb_E/ed.
  col_names = FALSE,
  n_max = 3
)

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

```

```

# Step 5: Wrangle Armed Forces Data ----
JDK_Branch_Names <- rep(
  x = c("Army", "Navy", "Marine Corps", "Air Force", "Space Force", "Total"),
  each = 3
)

JDK_Temporary_Headers <- paste(
  c("", JDK_Branch_Names),
  JDK_Forces_Headers[3,],
  sep = "."
)

names(JDK_Raw_Forces) <- JDK_Temporary_Headers

JDK_Clean_Forces <- JDK_Raw_Forces %>%
  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")
  )

# Step 6: Merge Data Frames ----
JDK_Joint_Forces_Ranks <- left_join(
  x = JDK_Clean_Forces,
  y = JDK_Clean_Ranks,
  by = join_by(`Pay.Grade` == `Pay Grade`, `Branch` == `Branch`)
)

# Step 7: Transform Group into Individual ----
JDK_Armed_Forces_Individuals <- JDK_Joint_Forces_Ranks %>%
  filter(!is.na(Frequency)) %>%
  uncount(
    weights = Frequency
  )

```

```

# Step 8: Create Data Table
JDK_Air_Force_Combined <- JDK_Joint_Forces_Ranks %>%
  filter(Branch == "Air Force")

JDK_Air_Force_Officer <- JDK_Air_Force_Combined %>%
  filter(Pay.Grade %in% c("01", "02", "03", "04", "05", "06", "07", "08", "09", "010"))

Frequency_Table <- JDK_Air_Force_Officer %>%
  group_by(Sex, Pay.Grade) %>%
  summarise(Frequency = sum(Frequency)) %>%
  pivot_wider(names_from = Pay.Grade, values_from = Frequency, values_fill = 0)

Frequency_Table <- Frequency_Table[c("Sex", "01", "02", "03", "04", "05", "06", "07", "08", "09", "010")]

Frequency_Table %>%
  kable() %>%
  kableExtra::kable_classic()

#| label: fig-koewler-family-names
#| fig-cap: "Frequency of Names within the Koewler Family from 1880 to 2013"
#| fig-alt: "Linegraph of Frequency of First Names within the Koewler Family from 1880 to 2013"

# Step 1: Load packages
library(dplyr)
library(ggplot2)
library(dcData)

# Step 2: Load data
data("BabyNames")

# Step 3: Create Koewler Family Names
Koewler_Family_Names <- c("David", "Stefanie", "Jacob", "Thomas")

# Step 4: Create Count of Names
Koewler_Family <- BabyNames %>%
  filter(name %in% Koewler_Family_Names) %>%
  group_by(name, year) %>%
  summarize(
    total = sum(count),
    .groups = "drop"
  )

# Step 5: Create Plot
ggplot(Koewler_Family,
  mapping = aes(

```

```

        x = year,
        y = total,
        color = name,
        linetype = name)) +
geom_line() +
theme_minimal() +
labs(title = "Frequency of Names from 1880 to 2013",
      x = "Year",
      y = "Count",
      color = "Name",
      linetype = "Name") +
scale_y_continuous(
  expand = expansion(mult = 0.01)) +
scale_x_continuous(limits = c(1880, 2013))

#| label: fig-box-problem
#| fig-cap: "Volume of a Box as a function of the Side length"
#| fig-alt: "Linegraph of Volume of a box as a function of the Side length"

# Step 1: Load Packages
library(ggplot2)

# Step 2: Create Function
getVolume <- function(sideLength, paperLength = 48, paperWidth = 36){
  vol <- sideLength * (paperWidth - 2*sideLength) * (paperLength - 2 * sideLength)
  return(vol)
}

# Step 3: Limit Length of X
sides <- seq(from = 0.01, to = 17.99, by = 0.01)

# Step 4: Create Plot

base <-
  ggplot() +
  xlim(0, 18)

base + geom_function(fun = getVolume) +
  labs(title = "Volume of Box as a function of the Side Length",
        x = "Side Length (inches)",
        y = "Volume (cubic inches)") +
  theme_minimal()

```

““