

# Activity 14

Ankita Bhattacharyya

2025-11-07

## Armed Forces Data

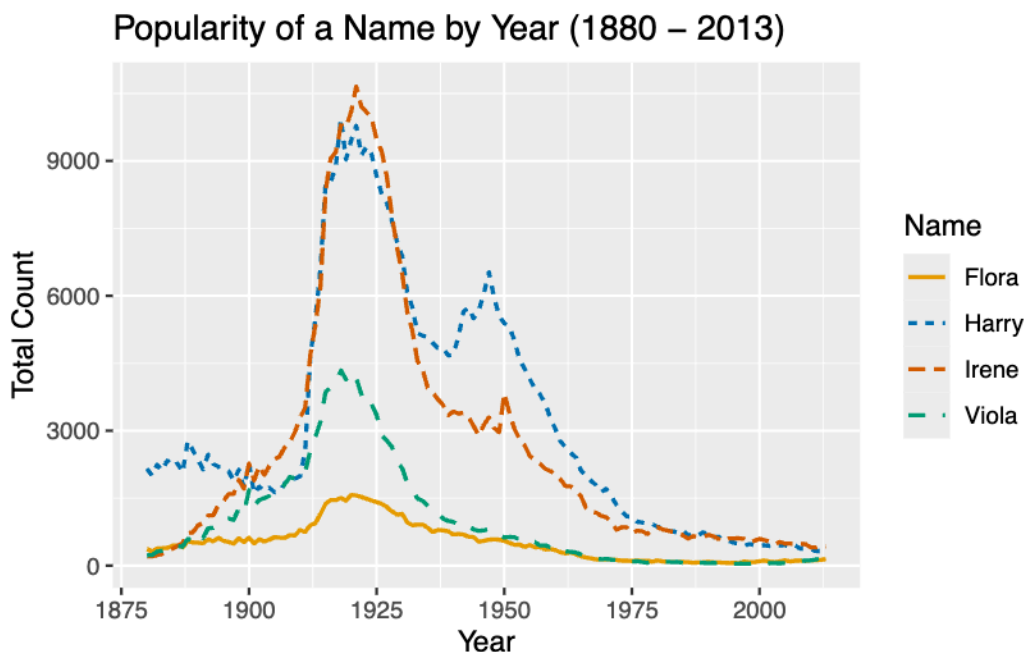
Table 1: Totals of Marine Corps Privates by Sex

Rank/Sex	Female	Male	Total
Corporal	2942 (2.01%)	28946 (19.75%)	31888 (21.76%)
First Sergeant OR Master Sergeant	275 (0.19%)	3559 (2.43%)	3834 (2.62%)
Gunnery Sergeant	760 (0.52%)	8191 (5.59%)	8951 (6.11%)
Lance Corporal	3787 (2.58%)	35047 (23.92%)	38834 (26.50%)
Private	659 (0.45%)	7233 (4.94%)	7892 (5.39%)
Private First Class	1604 (1.09%)	14688 (10.02%)	16292 (11.12%)
Sergeant	2723 (1.86%)	21481 (14.66%)	24204 (16.52%)
Sergeant Major OR Master Gunnery Sergeant	83 (0.06%)	1518 (1.04%)	1601 (1.09%)
Staff Sergeant	1370 (0.93%)	11667 (7.96%)	13037 (8.90%)
Total	14203 (9.69%)	132330 (90.31%)	146533 (100.00%)

Table 1 gives the relative and absolute frequencies for the Marine Corps' ranks separated by sex. Looking at just the absolute frequencies, within the Marine Corps, regardless of rank, there are fewer females in comparison to males. However, when you look at the relative frequencies, females in the higher ranks, like Corporal and Sergeant, have higher frequencies than females in the lower ranks. Overall, looking at just the totals, there doesn't seem to be any positive or negative relationship with Rank and Sex, as, regardless of rank, there are fewer females than males. But comparing within females, it can be seen that the higher the rank, the higher the percentage of females there is, with the exception of the highest ranks.

## Popular Baby Names Project

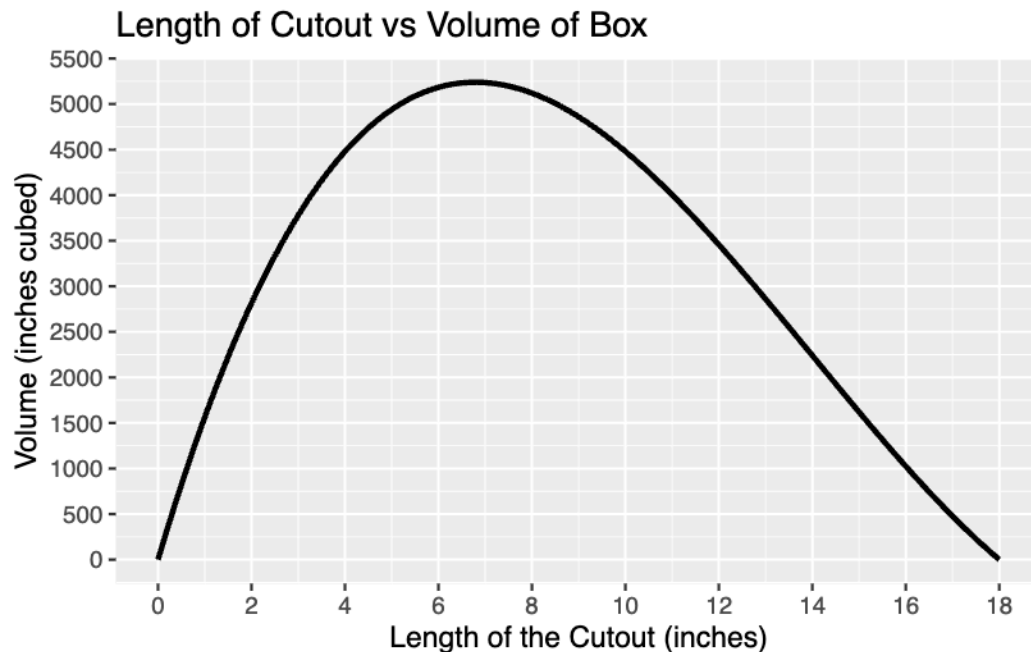
Figure 1: Line-plot of popularity of Baby Names



The Figure 1 shows the popularity trend of four names from 1880 to 2013. I chose the names I did because they are all fictional characters, and I was interested in their popularity. Overall, Flora has the fewest counts across all years compared to the other three names. Flora shows only one noticeable increase, with the name peaking around 1920, but it then decreases slowly right after. Around 1950, the trends for Viola and Flora merge and look almost identical through 2013. Viola shows a steady increase from 1880, with popularity in the name shooting up around 1913 and peaking around 1917. After around 1923, the name has a steady decrease. The name Irene has the highest peak among all the other names, with a total count of around 10,500 in 1923. Irene had a steady increase since 1880, but shot up around 1910. The name Harry has a peak around the same time as Irene's, but also shows another smaller peak around 1948. All of the chosen names after 1975 show very similar, steady, decreasing trends. Interestingly, around 1912, all the names had show a sharp increase. The reason could be that there were more babies around that time. The Figure 1 also shows another sharp increase, specifically for Irene and Harry around 1947, which aligns with the baby boom after World War II.

## The Box Problem

Figure 2: A line plot of the max volume of a box.



The Figure 2 shows the maximum total volume, with each input specifying the length of the cutout for a paper size of 36 inches by 48 inches. The maximum volume of the paper is around 5,500 cubic inches, which occurs when the cutout length is slightly less than 7 inches. Figure 2 displays that as the cutout length increases from 0 to 7, there is a positive relationship between the length of the cutout and the volume of the box made from the piece of paper. But after reaching the max volume of 5,500 inches cubed at a cutout length of 7 inches, the relationship becomes negative and ultimately reaches 0 inches cubed at a cutout length of 18 inches. Ultimately, the page should have a side length of around 22 inches, a width of around 34 inches, and a height of a little less than 7 inches, for a maximum volume of 5,500 cubic inches.

## Self Reflection

I've definitely learned throughout all the activities the importance of commenting and documenting the code. This also goes in hand with creating plans as well. When I look back at my older code, like activity 4 and 8, there aren't any comments, which makes following the code quite tough. But activities afterward, I made it a point to comment out the code along the way. Which made it way easier to follow what I'm doing. As mentioned before, I have improved on my plans as well. Earlier activities my plans were very vague, like having multiple steps combined into one step. Which was quite unhelpful when turning the steps into code. I would have to figure out more, while also confused about the initial code. Now, I try to make multiple smaller goals to break up the code. I also edit the plan as I write and fix the code. Coding wise, I have learned how to make nicely formatted tables and graphs that comply to accessibility standards. As well as principles and practices (Tufte and Kosslyn) to create impactful tables and well made graphs.

## Code Appendix

```
#Armed Forces Data Wrangling -----

#load packages
library(tidyverse)
library(rvest)
library(google sheets4)

# scraping the Rank data from html
rank_raw <- read_html("https://neilhatfield.github.io/Stat184_PayGradeRanks.html") %>%
  html_elements(css = "table") %>%
  html_table()
# taking out only needed table
Rank_needed_raw <- rank_raw[[1]]

#removing unwanted columns and rows
Rank_needed_raw = Rank_needed_raw[,-c(1,8)]
Rank_needed_raw = Rank_needed_raw[-c(1),]

#Renaming columns - couldn't use rename because it was one whole same name column
new_headers = c("Pay_Grade", "Army", "Navy", "MC", "AF", "SF")
colnames(Rank_needed_raw) <- new_headers

# stacking columns
Rank_clean <- Rank_needed_raw %>%
  pivot_longer(
    cols = Army:SF,
    names_to = "Branch",
    values_to = "Rank"
  )

gs4_deauth()

#reading the google sheet into the raw data
armynRaw <- read_sheet(
  ss = 'https://docs.google.com/spreadsheets/d/1cn4i0-ymB1ZytWXCwsJiq6fZ9PhGLUvbMBHlZqG4bwo/ed
)

ArmyBase <- armynRaw %>%
  rename( # renaming column names
    Pay_Grade = `Active-Duty Personnel by Service Branch, Sex, and Pay Grade`,
    Army_Male = `...2`,
    Army_Female = `...3`,
    Army_Total = `...4`,
    Navy_Male = `...5`,
```

```

Navy_Female = `...6`,
Navy_Total = `...7`,
MC_Male = `...8`,
MC_Female = `...9`,
MC_Total = `...10`,
AF_Male = `...11`,
AF_Female = `...12`,
AF_Total = `...13`,
SF_Male = `...14`,
SF_Female = `...15`,
SF_Total = `...16`,
junk = `...17`,
junk2 = `...18`,
junk3 = `...19`
) %>% # removing junk columns
select(-c(junk, junk2, junk3, Army_Total, Navy_Total, MC_Total, AF_Total, SF_Total))
#removing junk rows
ArmyBase = ArmyBase[-c(1,2,12, 18, 29, 30,31), ]
ArmyGroup_Clean <- ArmyBase %>%
mutate( #change value from string to numeric
  Army_Male = as.numeric(Army_Male),
  Army_Female = as.numeric(Army_Female),
  Navy_Male = as.numeric(Navy_Male),
  Navy_Female = as.numeric(Navy_Female),
  MC_Male = as.numeric(MC_Male),
  MC_Female = as.numeric(MC_Female),
  AF_Male = as.numeric(AF_Male),
  AF_Female = as.numeric(AF_Female),
  SF_Male = as.numeric(SF_Male),
  SF_Female = as.numeric(SF_Female)
) %>%
mutate( # changing NAs into 0
  SF_Male = case_match(
    .x = SF_Male,
    NA ~ 0,
    .default = SF_Male
  )
) %>%
mutate(
  SF_Female = case_match(
    .x = SF_Female,
    NA ~ 0,
    .default = SF_Female
  )
) %>%
pivot_longer( # stacking columns
  cols = Army_Male:SF_Female,

```

```

    names_to = "Branch",
    values_to = "Total"
  ) %>%
  separate_wider_delim( # separating branch from Sex
    cols = Branch,
    delim = "_",
    names = c("Branch", "Sex")
  ) %>%
  mutate(
    Total = case_match(
      .x = Total,
      NA ~ 0,
      .default = Total
    )
  )
)

#Army Clean with totals data set
US_Army_Clean <- left_join( # Combining the two data frames
  x = ArmyGroup_Clean,
  y = Rank_clean,
  by = join_by(Branch == Branch, Pay_Grade == Pay_Grade)
)

#Army clean with Individual data set
US_Army_Ind_Clean <- US_Army_Clean %>%
  uncount( # uncounting
    weights = Total
  )

#Armed Forces (MC) Table -----

#loading packages
library(tidyverse)
library(knitr)
library(kableExtra)
library(janitor)

#filtering to get subset
US_Armed_Stats <- US_Army_Ind_Clean %>%
  filter(Pay_Grade == "E1" |
    Pay_Grade == "E2" |
    Pay_Grade == "E3" |
    Pay_Grade == "E4" |
    Pay_Grade == "E5" |
    Pay_Grade == "E6" |
    Pay_Grade == "E7" |
    Pay_Grade == "E8" |

```

```

Pay_Grade == "E9",
Branch == "MC") %>%
tabyl(Rank, Sex) %>%
adorn_totals(where = c("row", "col")) %>%
adorn_percentages(denominator = "all") %>%
adorn_pct_formatting(digits = 2) %>%
adorn_title(
  placement = "combined",
  row_name = "Rank",
  col_name = "Sex"
)

#formatting
formatsN <- attr(US_Armed_Stats, "core") %>%
  adorn_totals(
    where = c("row", "col")
  )
MCFreq <- US_Armed_Stats %>%
  adorn_ns(position = "front", ns = formatsN)

#styling the frequency table
MCFreq %>%
  kable(
    caption = "Totals of Marine Corps Privates by Sex",
    booktabs = TRUE,
    align = c("l", rep("c",6)),
    format.args = list(big.mark = ',')
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped","condensed"),
    font_size = 10,
    stripe_color = "gray!10"
  )

#Baby Names Project -----

#loading in packages
library(ggplot2)
library(tidyverse)
data(data = "BabyNames", package = "dcData")

baby_clean <- BabyNames %>%
  group_by(
    name, year # grouping variables
  ) %>%
  summarise(total_count = sum(count)) %>% #summarizes counts of each name per year
  filter(

```

```

    name == 'Flora' |
    name == "Harry" |
    name == "Viola" |
    name == 'Irene' # pulling out the names
  )

ggplot(
  data = baby_clean,
  mapping = aes(
    x = year, #x-position
    y = total_count, # y-position
    color = name,
    linetype = name
  )
) + geom_line(
  size = 0.75 # thickness of lines
) + labs(
  title = "Popularity of a Name by Year (1880 - 2013)", # title of graph
  x = xlab("Year"), # change x-axis title
  y = ylab("Total Count"), # change y-axis title
  color = "Name",
  linetype = "Name",
  alt = "A line plot of the popularity of four names from the year 1880 to 2013." # alt text
) + scale_color_manual( #changing color of lines
  values = c(
    "Flora" = "#E69F00",
    "Harry" = "#0072B2",
    "Irene" = "#D55E00",
    "Viola" = "#009E73"
  )
) + theme_grey() # theme

#The Box Problem Code -----

#loading packages
library(ggplot2)

#creating function to find volume of the box
volume_of_box <- function(L_PAPER = 36, W_PAPER = 48, x){
  LBox <- L_PAPER - (x * 2)
  WBox <- W_PAPER - (x * 2)
  HBox <- x
  VBox <- LBox * WBox * HBox
  return(VBox)
}

#inputs for x

```

```

length_cutout <- data.frame(x = seq(from = 0, to = 18, by = 1))

#creating visualization
ggplot(
  data = length_cutout,
  mapping = aes(
    x = x
  )
) + stat_function(
  geom = "line", #type of graph
  linewidth = 1,
  fun = volume_of_box, #function x is passed through
  args = list(L_PAPER = 36, W_PAPER = 48)
) + labs( #labels for the plot
  title = "Length of Cutout vs Volume of Box", #title
  x = xlab("Length of the Cutout (inches)"), # x axis label
  y = ylab("Volume (inches cubed)"), # y axis label
  alt = "Line plot showing the change in max volume for a box." # alt text
) + theme_grey( # theme

) + scale_x_continuous( #changing the scale increments
  breaks = seq(from = 0, to = 18, by = 2)
) + scale_y_continuous(
  breaks = seq(from = 0, to = 6000, by = 500)
)

```