Activity 14

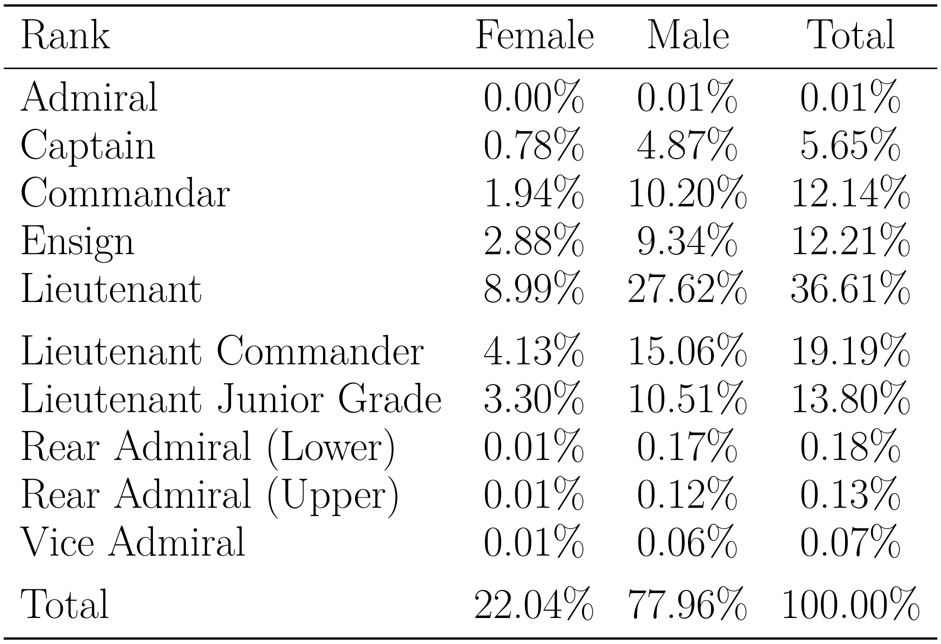
Data Visualizations from Stat 184

Brady Kelly

2025-11-13

Navy Data Table

Table 1: Frequency table of each sex in a specific rank when accounting for all officers in the Navy.



Rank	Female	Male	Total
Admiral	0.00%	0.01%	0.01%
Captain	0.78%	4.87%	5.65%
Commandar	1.94%	10.20%	12.14%
Ensign	2.88%	9.34%	12.21%
Lieutenant	8.99%	27.62%	36.61%
Lieutenant Commander	4.13%	15.06%	19.19%
Lieutenant Junior Grade	3.30%	10.51%	13.80%
Rear Admiral (Lower)	0.01%	0.17%	0.18%
Rear Admiral (Upper)	0.01%	0.12%	0.13%
Vice Admiral	0.01%	0.06%	0.07%
Total	22.04%	77.96%	100.00%

Figure 1: Frequency table of each sex in a specific rank when accounting for all officers in the Navy.

Figure 1 Displays a table that shows each sex in a specific rank, while accounting fro every officer in the Navy. ## Popularity of Baby Names

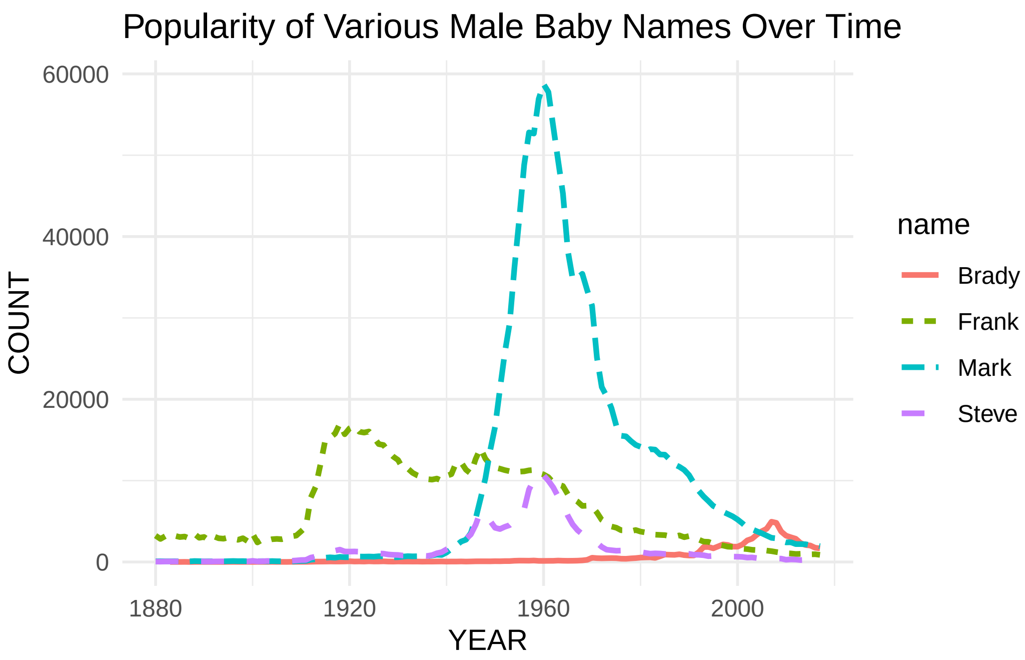


Figure 2: Plot of Baby Names Popularity Over Time

Figure 2 Displays a line chart representing four names. The x axis is the year while the y axis is the amount of babies named. ## Box Volume Function With Graph

Figure 3 Displays volume of a box over heights from 0 to 20. Using a function I created to find the maximum dimensions of a box.

What I've Learned So Far

In this class I learned how to be proficient in R. I now have strong skills in using it for statistical analysis. Writing basic R code as well as importing datasets and creating visualizations are all things that I am now able to do because of this class. Throughout the class I was able to practice writing code and using basic concepts to complete certain tasks. Having experience with R and certain data types is very helpful for my future. This course strengthened my programming skills and my ability to analyze data.

Code Appendix

```r
library(tidyverse)
library(rvest)
```

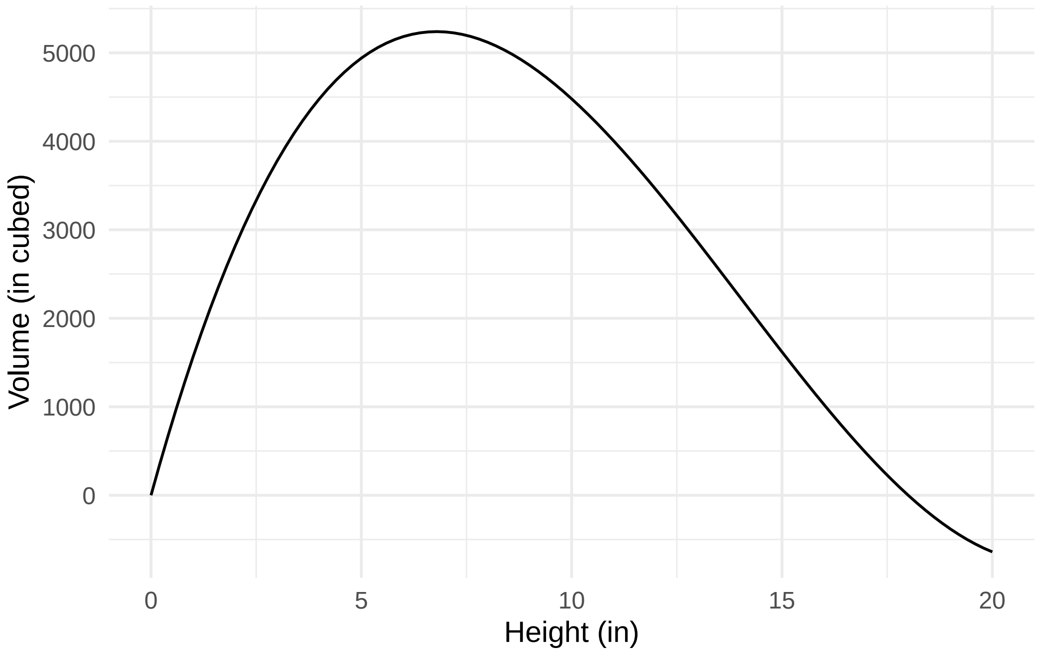


Figure 3: Plot of Volume vs Height

```
library(googlesheets4)
library(dplyr)
library(janitor)
library(kableExtra)

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

rawRanks <- webRanks[[1]] # Extract the data frame of ranks

# Wrangle Rank Data ----
## Enter a value in the first cell (1, 1)
rawRanks[1, 1] <- "Type"
## Extract actual column headers
rankHeaders <- rawRanks[1, ]
## Apply headers as column names
names(rawRanks) <- rankHeaders[1,]
## Remove redundant first row and last row
```

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

cleanRanks <- rawRanks %>%
  dplyr::select(!Type) %>% # Remove extra column
  pivot_longer(
    cols = !`Pay Grade`, # The improper name requires backticks
    names_to = "Branch",
    values_to = "Rank"
  ) %>%
  mutate(
    Rank = na_if(x = Rank, y = "--")
  )

# Load Armed Forces Data ----
gs4_deauth()
forcesHeaders <- read_sheet(
  ss = "https://docs.google.com/spreadsheets/d/1cn4i0-ymB1ZytWXCwsJiq6fZ9PhGLUvbMBHlzqG4bwo/e
  col_names = FALSE, # Turn off Column Names
  n_max = 3 # read only the first three rows
)

rawForces <- read_sheet(
  ss = "https://docs.google.com/spreadsheets/d/1cn4i0-ymB1ZytWXCwsJiq6fZ9PhGLUvbMBHlzqG4bwo/e
  col_names = FALSE, # Turn off Column Names
  skip = 3, # Skip the first three rows
  n_max = 28, # Read only the next 28 rows; drops footer
  col_types = "c" # Tells R to read everything as character data
)

# Wrangle Armed Forces Data ----
## Create good column names ----
### Pattern is Pay Grade followed by 3 columns for each branch in the order
### Army, Navy, Marine Corp, Air Force, Space Force, and Total
branchNames <- rep( # Create three copies of each branch
  x = c("Army", "Navy", "Marine Corps", "Air Force", "Space Force", "Total"),
  each = 3
)
tempHeaders <- paste( # Combine branch with other headers
  c("", branchNames),
  forcesHeaders[3,],
  sep = "."
)
```

```r
names(rawForces) <- tempHeaders

cleanForces <- rawForces %>%
  rename(Pay.Grade = `.Pay Grade`) %>%
  dplyr::select(!contains("Total")) %>% # Remove total columns
  filter(Pay.Grade != "Total Enlisted" &
           Pay.Grade != "Total Warrant Officers" &
           Pay.Grade != "Total Officers" &
           Pay.Grade != "Total") %>% # Remove total rows; see note below
  pivot_longer( # Reshape data
    cols = !Pay.Grade,
    names_to = "Branch.Sex",
    values_to = "Frequency"
  ) %>%
  separate_wider_delim( # Separate branches and sex
    cols = Branch.Sex,
    delim = ".",
    names = c("Branch", "Sex")
  ) %>%
  mutate(
    Frequency = na_if(Frequency, y = "N/A*"), # Convert N/A* to missing
    Frequency = parse_number(Frequency) # Parse values as numbers
  )

# Merge Data Frames ----
key_forcesRanks <- left_join(
  x = cleanForces,
  y = cleanRanks,
  by = join_by(Pay.Grade == `Pay Grade`, Branch == Branch)
) %>%
  mutate(
    RankClass = case_when(
      grepl(pattern = "^E", x = Pay.Grade) ~ "Enlisted",
      grepl(pattern = "^W", x = Pay.Grade) ~ "Warrent Officer",
      grepl(pattern = "^O", x = Pay.Grade) ~ "Officer",
    )
  )
# Transform Group into Individual ----
key_individualRanks <- key_forcesRanks %>%
  filter(!is.na(Frequency)) %>% # Remove all cases with missing counts
  uncount(
    weights = Frequency
  )

Frequency_data <- key_individualRanks %>%
  filter(Branch == "Navy" & RankClass == "Officer")%>%
  tabyl(Rank, Sex)%>%
  adorn_totals(where = c("row", "col"))%>%
  adorn_percentages(denominator = "all")%>%
  adorn_pct_formatting(digits = 2)

Frequency_data %>%
  kable(
    caption = "Frequency table of each sex in a specific rank when accounting for all office
    booktabs = TRUE,
    align = c("l", rep("c", 6))
  ) %>%
  kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 16
  )

#Baby Names Code
library(babynames)
library(dplyr)
library(ggplot2)

Baby_Name <- babynames %>%
  filter(name %in% c("Brady", "Mark", "Frank", "Steve"),
         sex == "M")

ggplot(Baby_Name) +
  aes(x = year, y = n, colour = name, linetype = name) +
  geom_line(size = 1) +
  labs(
    x = "YEAR",
    y = "COUNT",
    title = "Popularity of Various Male Baby Names Over Time"
  ) +
  theme_minimal()
# Box Problem Code
V <- function(height, width, length){
  (length-2*height)*(width-2*height)*height
}
```

```
library(ggplot2)
ggplot()+
  stat_function(
    fun = V,
    args = list(length = 36, width = 48),
    xlim = c(0, 20)
  ) +
  labs(
    x = "Height (in)",
    y = "Volume (in cubed)"
  ) +
  theme_minimal()
```