Activity 10

Aarush Bansal

2025-04-16

1 Armed Forces Data

First I created a data frame for individual cases by calling the data from a Google sheets. Then I proceeded to clean the data as it was a double header and R doesn't support that format. I also continued to tidy the data. After that I pivoted multiple times to create a data frame where all the soldiers were grouped by Pay Grade. Then utilizing uncount(), I created a data frame where the cases were individual soldiers separated by Pay Grade, Gender, Branch, and Rank.

Here is a two-way frequency table that shows the relationship between sex and rank in the United States Military highlighting specifically the Marine Corps's Sergeant rank as example.

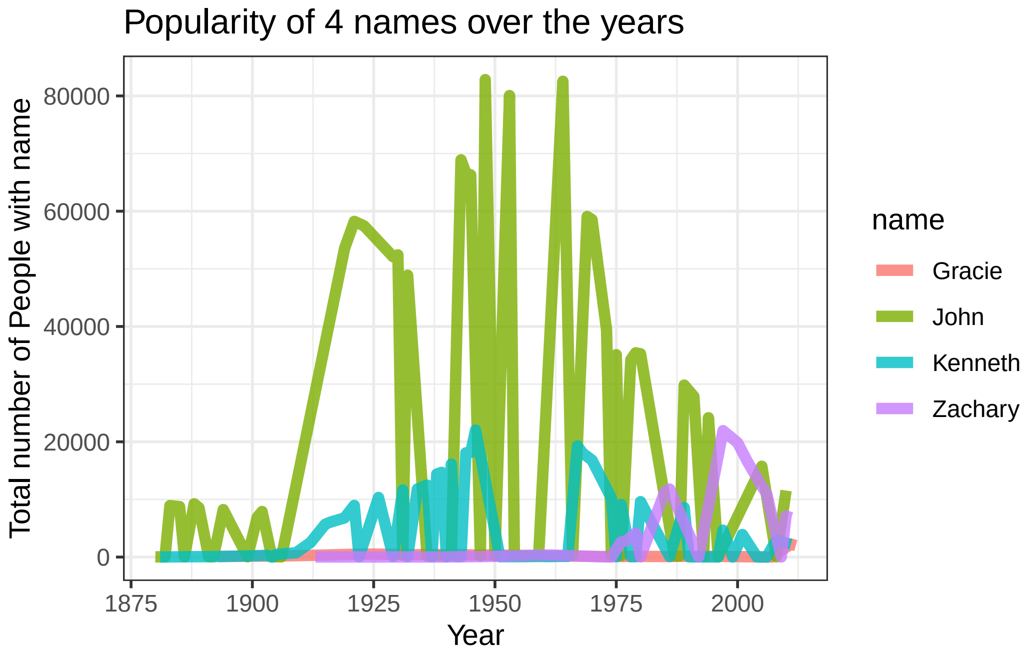
Table 1: Frequency Table of the Sex and Rank of Officers in the Marine Corps

Gender	Captain	First Lieutenant	Second Lieutenant	Total
Female	208	192	123	523
Male	1,766	1,125	764	3,655
Total	1,974	1,317	887	4,178

Looking at the visualization we can see the title, various names of officer ranks in the Marine Corps, and totals for Males and Females alongside totals of ranks to see the makeup of Marine Corps Officers. Here we can see a large disparity between the number of male officers and female officers in all ranks. However what's interesting to note is that the variance of the number of officers in each rank seem to be a lot less for females when compared to men. This could mean that the women that do make it to the upper chain of Marine Corps Command are able to push all the way to the highest rank consistently as the barrier of entry seems to be much higher for them due to physical and cultural limitations of the United States Military.

2 Baby Names

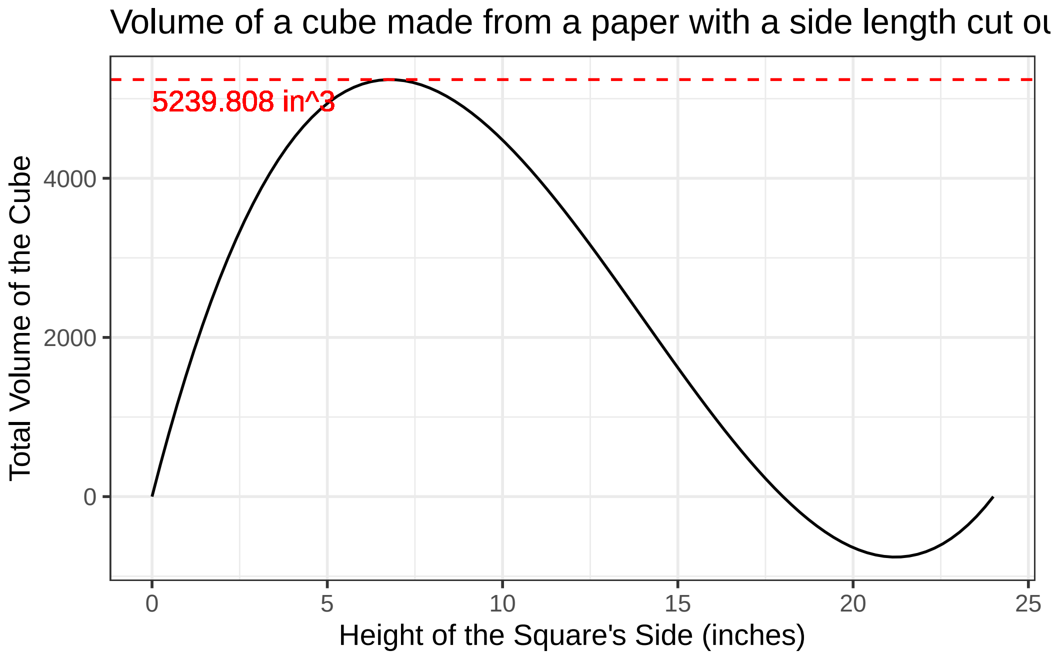
Here utilizing data of various baby names, I created a time series plot looking at the names John, Kenneth, Gracie, and Zachary. The only vague reasoning behind the selection of names was to try to pick names that are currently popular and all names I've personally known multiple people to have.



Taking a look at the visualization, there is clear labeling and scaling alongside a key to help understand the various colors being utilized to look at the popularity of the various names. The graph also has a light transparency to help see overlapping values that may occur. Finally, we can see without a doubt that John is historically the most popular name due to what I assume is the reference to Biblical lore but I'm not 100 percent sure about the exact reasoning why. The names Gracie and Zachary seemed to become more popular after the 2000s and 1980s respectively.

3 The Box Problem

Utilizing a sheet of paper with 36 inches of length and 48 inches of width we can create a cube by cutting the corners off with various side lengths. Here I am creating a visualization of the various volumes achievable by the various sizes we can cut off the corners of the paper.



Taking a look at this visualization we can see that the max volume possible is 5239.808 inches cubed with a square side length of roughly 6.4 inches. The curve also demonstrates the various possible cube volumes that can be created from this singular sheet of paper and it seems that a little past 18 inches it becomes impossible to create a cube anymore from the paper with the given dimensions.

4 Reflection

Looking back on the course I have realized that I not only have learned how to use R but also how to teach myself how to keep using R and teaching myself new syntax as time goes on. Due to the expansiveness of the language and it's public libraries, there are many tools that I still haven't encountered and will utilize in the future. For example even though we went over tidyverse and rvest in class, I still needed to read through the documentation to understand how to utilize it for the various activities we had assigned, and I am now capable of data wrangling most data sets I will come across for future use. I also have also learned how to improve my code and to make it understandable for other readers as my past programming classes never emphasized such issues and allowed just the raw code to be submitted as long as it ran and gave the proper output. Lastly, I learned how to create concise and effect visualizations through ggplot and the Kossyln and Tufte readings assigned in class, utilizing them to create time-series plots, histograms, line graphs, scatter plots, etc. I can with 100 percent certainty say that I have grown as an individual in my knowledge and experience about R from this class and I appreciate everything you have taught us thus far.

5 Code Appendix

```r
# Creating the data frame where each case is
#an individual soldier that includes rank names
library(tidyverse)
library(googlesheets4)
library(rvest)
#import data from the html and google sheet into 2 separate data frames
#gs4_deauth(), read_sheet, read_html, html_elements, html_table
gs4_deauth()
armyRaw <- read_sheet(ss = "https://docs.google.com/spreadsheets/d/1cn4i0-ymB1ZytWXCwsJiq6fZ9P
                      skip = 2,
                      col_types = c("?"),
                      guess_max = 10,
                      .name_repair = "minimal"
                      )

ranksRawList <- read_html(x = "https://neilhatfield.github.io/Stat184_PayGradeRanks.html") %>%
  html_elements(css = "table") %>%
  html_table(convert = TRUE, header = TRUE)



#tidy data

#remove totals from the google sheet data frame and
#removes coast guard from the html data frame if necessary

# Converts the tibble from HTML functions to a dataframe that is tidy
ranksRaw <- data.frame(ranksRawList[1])
colnames(ranksRaw) <- unlist(ranksRaw[1, ])
ranksRaw <- ranksRaw[-c(1,26), ] %>%
  subset(select = -c(1, 8))

# Created Army Tidy
armyTidy <- armyRaw[-c(10, 16, 27, 28, 29), ] %>%
  subset(select =  -c(4, 7 ,10, 13, 16, 17, 18, 19))

#Wrangle data into 2 seperate data frames

armyGrp <- pivot_longer(armyTidy,
                        cols = c("Male", "Female"),
                        names_to = "Gender",
                        values_drop_na = TRUE
                        )
Branch <- data.frame(Branch = rep(c("Army", "Army",
                                    "Navy", "Navy",
                                    "Marine Corps", "Marine Corps",
                                    "Air Force", "Air Force",
                                    "Space Force", "Space Force"),
                                  times = 22
                                  )
                     )
armyGrp <- bind_cols(armyGrp, Branch, .name_repair = "minimal")
ranksTidy <- pivot_longer(ranksRaw,
                          cols = c("Army",
                                   "Navy",
                                   "Marine Corps",
                                   "Air Force",
                                   "Space Force"
                                   ),
                          names_to = "Branch",
                          values_to = "Rank"
                          )
armyGrp <- left_join(armyGrp,
                     ranksTidy,
                     by = c("Pay Grade", "Branch"),
                     copy = TRUE)
#ArmyGrp is the final Data Frame for grouped cases in the Army Data

armyInd <- uncount(armyGrp, weights = value)

#ArmyInd is the final Data Frame for individual cases in the Army Data
library(janitor)
library(knitr)
library(kableExtra)
library(tidyverse)
library(tidyr)
#IMPORTANT: armyInd is the individual case table from above

colnames(armyInd)[1] <- "PayGrade"

IndOfficer <- armyInd %>%
  filter(PayGrade == c("O1", "O2", "O3"), Branch == "Marine Corps")

#Creates the dataframe inside R to work with for Male and Female cases
freqOfficer <- IndOfficer %>%
  tabyl(Gender, Rank) %>%
  adorn_totals(c("row", "col"))

#Prints the data in a simple but clean format
#for the reader/anyone else who may want these frequency tables
freqOfficer %>%
  kable(caption = "Frequency Table of the Sex and Rank
        of Officers in the Marine Corps",
        format.args = list(big.mark = ",")) %>%
  kable_classic()

#Start of baby plot
#| label: BabyPlot

#Create subset of baby names that just includes John, Kenneth, Gracie, and Zachary
library(tidyverse)
library(dplyr)
library(dcData)

subBabyNames <- BabyNames %>%
#filtering only the necessary names,
#then creating a frequency table after changing the case
#from names to individual people
  filter(name == c("John", "Kenneth", "Gracie", "Zachary")) %>%
  uncount(weights = count) %>%
  group_by(year, name) %>%
  summarize(count=sum(n()))

#Begin Plotting using year and count as the x and y
ggplot(
  data = subBabyNames,
  mapping = aes(
    x = year,
    y = count,
    color = name
  )
) +
  #Thickening the lines slightly
  #while adding some transparency to help see the stacked lines
  geom_line(size = 2, alpha = 0.8) +
  #Labeling to help with visualization and to title the graph
  labs(
    x = "Year",
    y = "Total number of People with name",
    title = "Popularity of 4 names over the years"
  ) +
  theme_bw()
#Volume Function for the box with length = 36 inches and width = 48 inches
findVolume <- function(h){
  v <- (36-2*h)*(48-2*h)*h
  return(v)
}
```

```
#Find the max value for the function
maxVal <- max(findVolume(h = seq(0,48/1.999999,0.1)))

#Create plot for the volume function
#utilizing a vector of values for h instead of a manual command.
data_frame(h = seq(0,48/1.999999,0.1)) %>%
ggplot(aes(h)) +
  stat_function(fun = findVolume, geom = "line") +
  #labels the maximum value with a label and a dashed line
  #that show the vertex in red.
  geom_hline(aes(yintercept = maxVal), linetype = "dashed", color = "red") +
  geom_text(aes(0,maxVal,label = paste(maxVal, "in^3"),
                vjust = 1.5,
                hjust = -0.0001),
            color = "red"
            ) +
  labs(
    x = "Height of the Square's Side (inches)",
    y = "Total Volume of the Cube",
    title = "Volume of a cube made from a paper with a side length cut out"
  ) +
  theme_bw()
```