Activity 14 - First QMD File

Max Lick

2025-11-10

1 Armed Forces Data Wrangling Redux

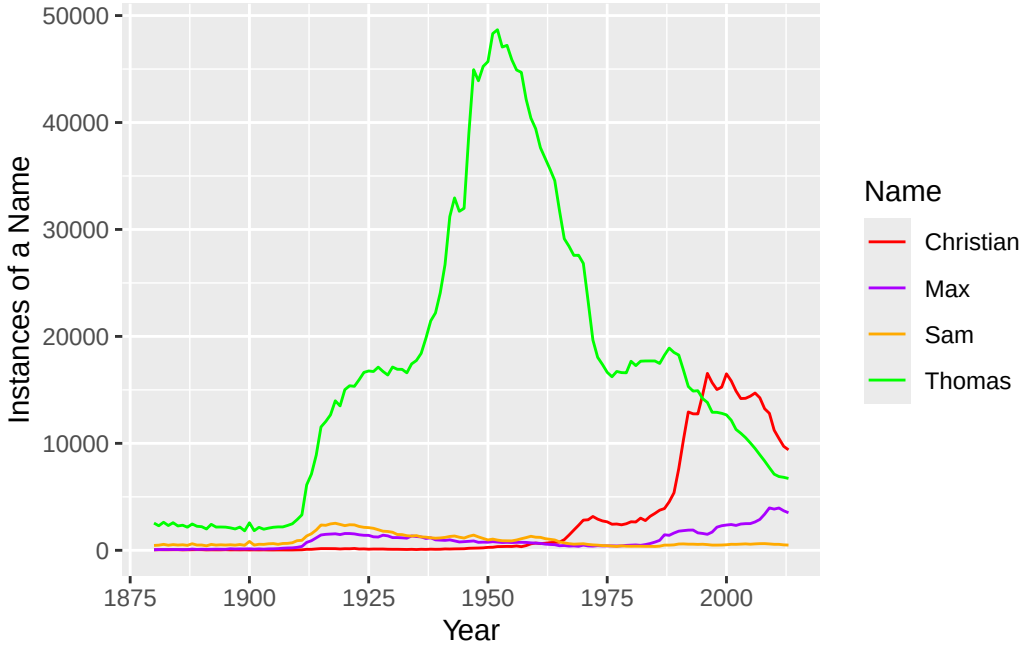
Table 1: Gender Distribution By Rank in the Army

Rank	Female	Male	Total
Second Lieutenant	2,400 (3.1225%)	7,122 (9.2660%)	9,522 (12.3884%)
First Lieutenant	3,006 (3.9109%)	9,550 (12.4249%)	12,556 (16.3358%)
Captain	6,053 (7.8752%)	20,986 (27.3035%)	27,039 (35.1786%)
Major	3,044 (3.9603%)	12,350 (16.0678%)	15,394 (20.0281%)
Lieutenant Colonel	1,531 (1.9919%)	6,939 (9.0279%)	8,470 (11.0197%)
Colonel	452 (0.5881%)	3,161 (4.1126%)	3,613 (4.7006%)
Brigadier General	18 (0.0234%)	100 (0.1301%)	118 (0.1535%)
Major General	8 (0.0104%)	80 (0.1041%)	88 (0.1145%)
Lieutenant General	5 (0.0065%)	46 (0.0598%)	51 (0.0664%)
General	0 (0.0000%)	11 (0.0143%)	11 (0.0143%)
Total	16,517 (21.4892%)	60,345 (78.5108%)	76,862 (100.0000%)

The above table shows the frequency of each rank in the army, separated by gender. As you can clearly see, females make up just under a quarter of the army population. Looking through the rank distribution, however, females are not proportionately represented in higher up ranks. In the first four ranks, females make up 20-25 percent of the population. For the final four ranks, they make up nearly 10 percent of the population.

2 Popularity of Baby Names

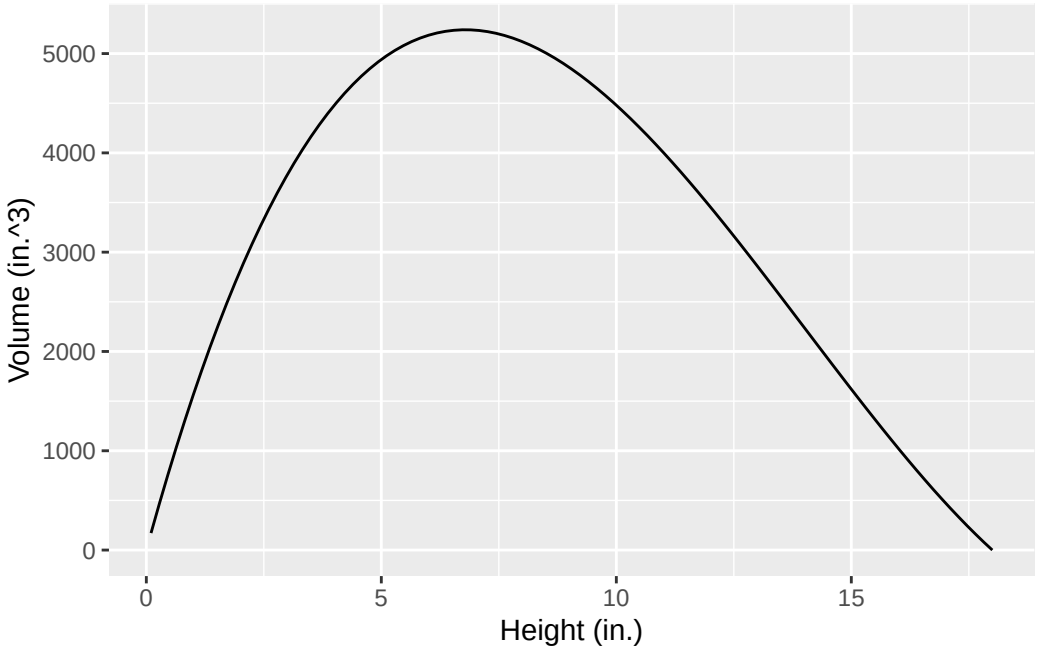
Table 2: Baby Name Distribution Throughout Time



The graph above shows the yearly name distribution of the names of my three roommates and I. This visualization shows a trend of an overall increase in births each year. Following Thomas through time, the name hit a massive spike in both 1915 and 1935, hitting its peak in the 1950's. More recently, it has dropped down more similarly to the others. Christian was not a common name for nearly a century, and then gained popularity in the 1990's and remains the most popular of the four today. Both Max and Sam have not had much of a peak, but Max has a slight increase whereas Sam has not seen much change.

3 Plotting a Mathematical Function

Table 3: Volume of Box



The graph above shows the relationship between the chosen height of the box versus the box's volume. As the graph shows, it peaks around 6-7 before it begins to drop off again. The peak volume is 5,239.8 cubic inches, with a chosen height of 6.8 inches.

4 Self-Reflection

Throughout the course, I have gained valuable information about the ins and outs of R, as well as some valuable insight into statistics. I have also gained thorough knowledge of how to create an effective data visualization from the works of Tufte and Kosslyn, as seen in the three examples above. As shown by this document in particular, I have learned how to create good documentation through the usage of Quarto and its implementation of YAML for the header. This class has provided me with an overall valuable basis in statistical analysis via R that I will be able to use going forward in my career as a Data Scientist.

5 Code Appendix

```
# Step 1: Load packages ----
library(tidyverse)
library(rvest)

# Step 2: Import necessary datasets ----
armedForcesRaw <- read.table( #Import Armed Forces
  "C:/Users/pacad/Downloads/US_Armed_Forces_(6_2025) - Sheet1.csv",
  header = TRUE,
  sep =',',
  skip = 2)

rankData <- read_html( #Import Pay Range and Rank
  x = "https://neilhatfield.github.io/Stat184_PayGradeRanks.html"
) %>%
  html_elements(css = "table") %>%
  html_table()

# Step 3: Tidy Armed Forces Dataframe ----
#  select, filter, rename, pivot_longer, seperate_wider_delim, unite
dataClean <- armedForcesRaw %>%
  select(c(-Total, -Total.1, -Total.2, -Total.3, -Total.4,
           -Total.5, -Male.5, -Female.5)) %>% #Remove "Total" columns
  #The next five lines remove summary statistics
  filter(Pay.Grade != "Total Officers") %>%
  filter(Pay.Grade != "Total") %>%
  filter(Pay.Grade != "Total Enlisted") %>%
  filter(Pay.Grade != "Total Warrant Officers") %>%
  filter(Pay.Grade !=
    "Source: DMDC Active-Duty Military Personnel Master File (June 2025)") %>%
  rename( #Change names to recognize gender and branch
    Male.Army = Male,
    Female.Army = Female,
    Male.Navy = Male.1,
    Female.Navy = Female.1,
    Male.Marine_Corps = Male.2,
    Female.Marine_Corps = Female.2,
    Male.Air_Force = Male.3,
    Female.Air_Force = Female.3,
    Male.Space_Force = Male.4,
    Female.Space_Force = Female.4
  ) %>%
  pivot_longer(
    #Pivot Gender/Branch rows into a column where amount has respective
    #amounts as values
    cols = Male.Army:Female.Space_Force,
    names_to = "Gender",
    values_to = "Amount"
  ) %>%
  separate_wider_delim( #Separate Gender/Branch column into respective columns
    cols = Gender,
```

```
    delim = '.',
    names = c('Gender', 'Branch')
  ) %>%
  unite( #Unite Pay Grade and Branch Columns
    col = "PG_Branch",
    c("Pay.Grade", "Branch"),
    sep = '/'
  )

# Step 4: Tidy Pay Grade Dataframe ----
#  as.data.frame, gsub, make.unique, select, rename, filter, pivot_longer, unite
#Turn data into dataframe
cleanRankData <- as.data.frame(rankData[[1]])
#Remove spaces in column names
colnames(cleanRankData) <- gsub(" ", "", colnames(cleanRankData))
#Change column names so all are unique
colnames(cleanRankData) <- make.unique((colnames(cleanRankData)))
#Remove empty first column
cleanRankData <- select(cleanRankData, -1)
#Change column names to recognize branch
cleanRankData <- rename(cleanRankData,
                        Pay.Grade = PayGrade,
                        Army = RanksbyBranchofService,
                        Navy = RanksbyBranchofService.1,
                        Marine_Corps = RanksbyBranchofService.2,
                        Air_Force = RanksbyBranchofService.3,
                        Space_Force = RanksbyBranchofService.4,
                        Coast_Guard = RanksbyBranchofService.5
)
#Remove coast guard column
cleanRankData <- select(cleanRankData, -Coast_Guard)
#Remove first row
cleanRankData <- filter(cleanRankData, Pay.Grade != "Pay Grade")
#Remove last row
cleanRankData <- filter(cleanRankData, Pay.Grade !=
  "Note: -- indicates that a pay grade is not currently used by a service branch")
#Pivot Branch data into Branch and Rank columns to identify Rank based on Branch
cleanRankData <- pivot_longer(cleanRankData,
                              cols = Army:Space_Force,
                              names_to = 'Branch',
                              values_to = 'Rank'
)
#Join Pay Grade and Branch columns
cleanRankData <- unite(cleanRankData,
                       col = "PG_Branch",
                       c("Pay.Grade", "Branch"),
                       sep = '/')

# Step 5: Join Tidied Dataframes ----
#  left_join, join_by
#join dataframes together by Pay Grade/Branch column
dataPolished <- left_join(
```

```
  x = dataClean,
  y = cleanRankData,
  by = join_by(PG_Branch == PG_Branch)
)

# Step 6: Tidy Combined Dataframe
#  separate_wider_delim, filter, mutate, uncount
dataPolished <- dataPolished %>%
  #Separate Pay Grade and Branch columns
  separate_wider_delim(
    cols = PG_Branch,
    delim = '/',
    names = c("Pay.Grade", "Branch")
  ) %>%
  #Remove null values
  filter(
    Amount != "N/A*"
  ) %>%
  #Remove commas from amounts and turn amounts into numeric values
  mutate(
    Amount = gsub(",","",Amount),
    Amount = as.numeric(Amount)
  ) %>%
  #Create case for each individual soldier
  uncount(
    Amount
  )



#------------------------------------------------------------------------
# Step 1: Load Necessary Packages
library(janitor)
library(kableExtra)
# Step 2: Create Data Table of Officers in the Army ----
rankDataTable <- dataPolished %>%
  filter(
    Branch == 'Army',
    Pay.Grade == 'O1'|
    Pay.Grade == 'O2'|
    Pay.Grade == 'O3'|
    Pay.Grade == 'O4'|
    Pay.Grade == 'O5'|
    Pay.Grade == 'O6'|
    Pay.Grade == 'O7'|
    Pay.Grade == 'O8'|
    Pay.Grade == 'O9'|
    Pay.Grade == 'O10') %>%
  tabyl(Rank, Gender) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_percentages(denominator = "all") %>%
  adorn_pct_formatting(digits = 4) %>%
  arrange(factor(Rank, levels = c("Second Lieutenant",
```

```
                                    "First Lieutenant",
                                    "Captain",
                                    "Major",
                                    "Lieutenant Colonel",
                                    "Colonel",
                                    "Brigadier General",
                                    "Major General",
                                    "Lieutenant General",
                                    "General")))

# Step 3: Turn Data Table into Frequency Table ----
rankFreqTable <- rankDataTable %>%
  adorn_ns(
    position = "front",
    format_func = function(x) {
      format(x, big.mark = ",")
    }
  )

# Step 4: Create an Organized and Easily Digestible Table ----
rankFreqTable %>%
  kable( # Make Pretty Table
    format = "latex",
    align = c("l", rep("c", 6)) # Control the text alignment in each column
  ) %>%
  kable_classic( # Pre-built styling
    font_size = 16 # Control Font Size
  )


#------------------------------------------------------------------------
# Step 1: Load necessary packages ----
library(dcData)
# Step 2: Wrangle Data ----
babyData <- BabyNames %>%
  #Select only the name, count, and year statistics
  select(
    name, count, year
  ) %>%
  #Choose four names
  filter(
    name == "Christian" | name == "Sam" | name == "Thomas" | name == "Max"
  ) %>%
  #Remove gender disparity
  group_by(name, year) %>%
  summarise(count = sum(count))


#------------------------------------------------------------------------
# Step 1: Load necessary packages ----
library(ggplot2)
# Step 2: Create Visualization ----
```

```r
ggplot(
  #Select Data
  data = babyData,
  #Create mapping
  mapping = aes(
    x = year,
    y = count,
    color = name
  )
) +
  #Create line graph
  geom_line() +
  #Add Labels
  labs(
    x = 'Year',
    y = 'Instances of a Name',
    color = 'Name',
    alt = "Line graph showing the distribution of four different baby names since 1875"
  ) +
  #Change line colors for better differentiation
  scale_color_manual(
    values = c("#FF0000",
               "#AA00FF",
               "#FFAA00",
               "#00FF00")
  )


#-------------------------------------------------------------------------
# Step 1: Initialize all possible height values ----
nums <- seq(from = 0, to = 18, by = 0.1)

# Step 2: Create function to find volume ----
getVolume <- function(x,length = 48, width = 36){
  #Volume Equation
  volume <- (length - 2*x) * (width - 2*x) * x
  return(volume)
}


#-------------------------------------------------------------------------
# Step 1: Load necessary packages ----
library(ggplot2)
# Step 2: Create Visualization ----
ggplot(
  data = NULL
) +
  #Create graph based on function
  stat_function(
    geom = 'line', #Use line graph geometry
    fun = getVolume, #Use getVolume function
    xlim = c(0.1, 18) #Establish bounds of x values
```

```
  ) +
  labs( #Create Labels for axes
    x = 'Height (in.)',
    y = 'Volume (in.^3)',
    alt = "A line graph showing the relationship between height and volume of a box"
  )


#-------------------------------------------------------------------------
```