

Activity 14A

Liam Bengston

2026-03-15

Table 1: Two-way frequency table of Sex by Rank for Marine Corps Officers

Table: Two-way frequency table of Sex by Rank for Marine Corps Officers

Introduction

This plot will show the ebbs and flows in the popularity of certain names over time. The difference in genders and how the popularity of the choices in name might act differently accordingly is shown by the split in the graph. I chose the names I did because they all fit within the same scale so one name's values wouldn't completely drown out the rest.

Figure 1: Line graph of Baby Names Over the Years

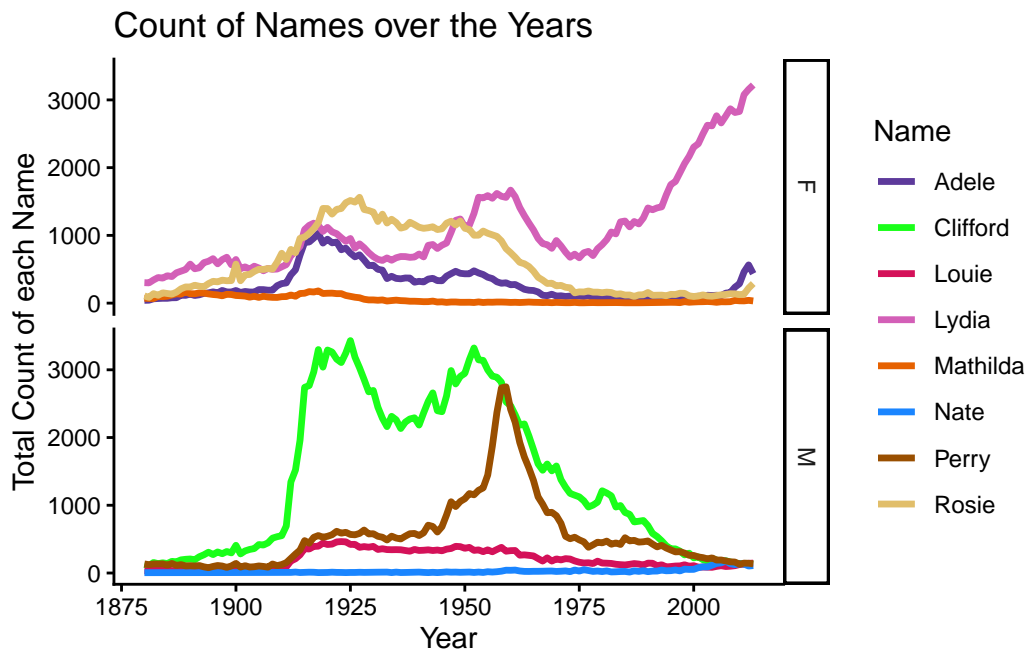
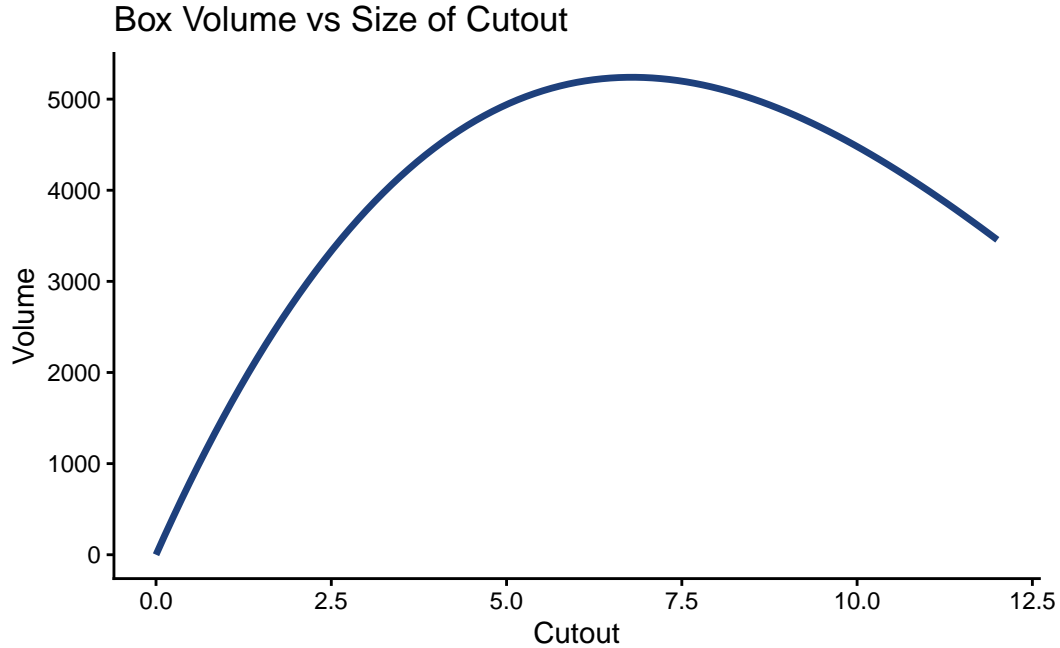


Figure 2: Box Volume vs Cutout Height



Explanation

This plot will present the peak of the box's volume and at which cutout size it appears. The box problem will have its answers seen in the plot as it is the plot that showcases the point at which the target cutout size and target, or max, volume are found.

Code Appendix

```
##| echo: true ##| eval: false
```

Appendix 1: Armed Forces Data Wrangling

```
##| label: tbl-airForceOfficers ##| tbl-cap: "Two-way frequency table of Sex by Rank for Marine  
Corps Officers" ##| tbl-pos: "H" ##| include: true ##| fig-alt: "Code that tidies and merges Armed  
Forces personnel data with rank information"
```

Make two tied data frames for the June 2025 armed forces data

1 where case is a group of soliders and 1 where the case is

an individual solider. Both need to have rank added.

Step 1: Load Packages —

```
library(tidyverse) library(rvest) library(google Sheets4)
```

Step 2: 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
```

Step 3: 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 = "-") )
```

Step 4: Load Armed Forces Data —

```
gs4_deauth() # Prevents needing to sign into a Google account forcesHeaders <- read_sheet(  
ss = "https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicMIVDF7Gr-  
nXCb5qbwb_E/edit?usp=sharing", 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/19xQnI1cBh6Jkw7eP8YQuuicMIVDF7Gr-  
nXCb5qbwb_E/edit?usp=sharing", 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 na = c("N/A ")  
# Tells R to treat the N/A as missing values )
```

Step 5: Wrangle Armed Forces Data —

Step 5a: 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 = "" )  
names(rawForces) <- tempHeaders
```

Step 5b: Wrangle Armed Forces Data —

```
cleanForces <- rawForces %>% rename(Pay.Grade = .Pay Grade) %>% dplyr::select(!contains("Total"))  
%>% # Remove total columns filter( # Remove total rows; see note below Pay.Grade != "Total  
Enlisted" & Pay.Grade != "Total Warrant Officers" & Pay.Grade != "Total Officers" & Pay.Grade  
!= "Total" ) %>% 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") )
```

Step 6: Merge Data Frames —

```
key_forcesRanks <- left_join( x = cleanForces, y = cleanRanks, by = join_by(Pay.Grade == Pay  
Grade, Branch == Branch) )
```

Step 7: Transform Group into Individual —

```
key_individualRanks <- key_forcesRanks %>% filter(!is.na(Frequency)) %>% # Remove all cases  
with missing counts uncount( weights = Frequency )
```

```
marine_officers <- key_individualRanks %>% filter(Branch == "Marine Corps", str_starts(Pay.Grade,  
"O-"))
```

```
knitr::kable( table(marine_officers$Sex, marine_officers$Rank), caption = "Two-way frequency ta-  
ble of Sex by Rank for Marine Corps Officers" )
```

Appendix 2: Baby Names Graph

```
#| label: fig-nameLineGraph #| fig-cap: "Line graph of Baby Names Over the Years" #|
fig-pos: "H" #| fig-alt: "Line graph that shows the selected baby names over the years" #|
echo: false # Make a line graph of baby names by gender — library(tidyverse) library(janitor)
library(knitr) library(kableExtra) library(dcData) library(rvest) library(dplyr) library(readr)
library(googlesheets4)
```

```
data(BabyNames)
```

```
BabyNames_untidy <- BabyNames %>% #this holds the data that will be graphed later fil-
ter(name %in% c("Mathilda", "Adele", "Nate", "Louie", "Rosie", "Lydia", "Perry", "Clifford"))
```

selects only the female versions of these names

```
BabyNames_female_wrangled <- BabyNames_untidy %>% filter(name %in% c("Mathilda",
"Adele", "Rosie", "Lydia") & sex == "F")
```

selects only the male versions of these names

```
BabyNames_male_wrangled <- BabyNames_untidy %>% filter(name %in% c("Nate", "Louie",
"Clifford", "Perry") & sex == "M")
```

recombines the male and female information so that it can be graphed

```
BabyNames_wrangled <- bind_rows(BabyNames_female_wrangled, BabyNames_male_wrangled)
%>% rename(Name = name) # capitalizes name to be more presentable
```

sets up the line graph pulling from BabyNames_wrangled

```
ggplot( data = BabyNames_wrangled, mapping = aes( x = year, y = count, color = Name ) ) +
geom_line(size = 1.2) + # increased line size a little to improve readability facet_grid(vars(sex))
+ # splits graph by sex so both only have 4 lines scale_color_manual(values = c( "Mathilda"
= "#E66100", "Adele" = "#5D3A9B", "Nate" = "#1A85FF", "Louie" = "#D41159", "Rosie"
= "#E1BE6A", "Lydia" = "#D35FB7", "Clifford" = "#1AFF1A", "Perry" = "#994F00" )) +
labs( title = "Count of Names over the Years", x = "Year", y = "Total Count of each Name" ) +
theme_classic() # a visual polish adjustment “
```

Appendix 3: Box Volume Function

```
#| label: fig-boxVolumeFunction #| fig-cap: "Box Volume vs Cutout Height" #| fig-pos: "H" #|
fig-alt: "Line graph showing how box volume changes with cutout height" # function that finds
the volume at a specific x cutout box_volume <- function(x) { length = 36 width = 48 v = (length
- (2 x)) (width - (2 x)) x return(v) } # plots the the volume at certain x cutouts and the peak is
the max ggplot(data.frame(x = c(0, 12)), aes(x = x)) + stat_function(fun = box_volume, color
= "#1E407C", size = 1.2) + labs( title = "Box Volume vs Size of Cutout", x = "Cutout", y =
"Volume" ) + theme_classic()
```