

Activity14

Alek Melenski

2025-11-17

Armed Forces Data Wrangling Redux (Activities #08 and #10)

Activity #8

For activity #8, I fixed my code by changing the names of the x_1, x_2, x_n. They were initially the wrong names and it caused the entire script to mess up and the end data frame (military_individual) ended up being blank.

see appendix for code

Activity #10

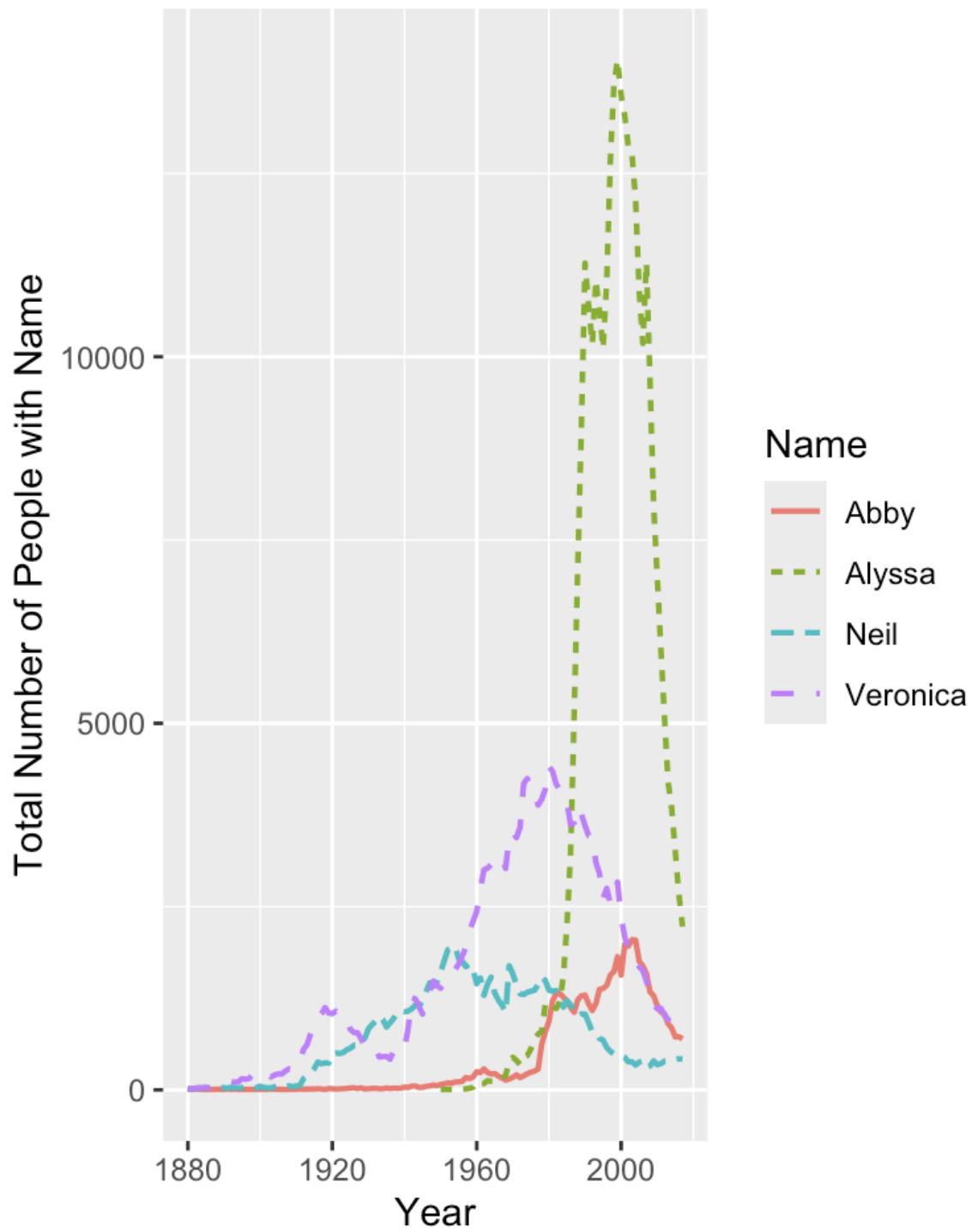
Creates a 2 way frequency table for a chosen branch and rank of the US Armed Forces data set.

see appendix for code

Popularity of Baby Names (Activity #13)

I chose the name that I did because they are the names of the stat 184 professors. They also have a good variety of popularity so it makes the visualization a little more interesting.

Popularity of the Stat 184 Instructors First Names



Plotting a Mathematical Function (Activity #04)

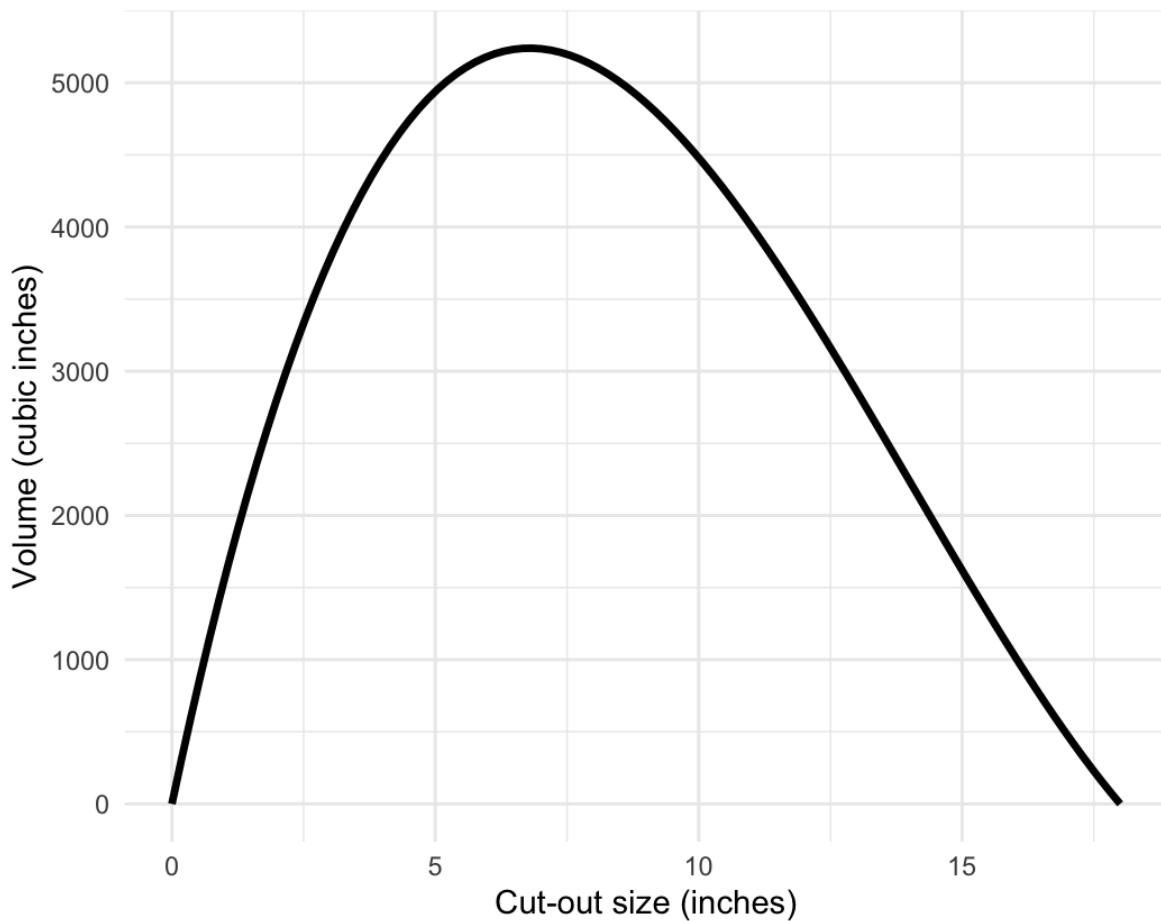
a.

see appendix for code

b.

see appendix for code

Figure 1. Volume of Box vs. Cut-Out Size



What You Feel You've Learned So Far

Coming into this course, I had absolutely no knowledge of R. I feel like I have learned a lot about R and I honestly prefer the visuals of ggPlot over Matplotlib in python which is my most well known language. I like the easier customization that it offers and ability to wrangle data more effectively. I also just learned about quarto markdown files and I think they are pretty cool and useful aswell.

Appendix

Activity 8

```
library(tidyverse)
library(janitor)

# Load data
armedForces <- read.csv("/Users/alekmelenski/Downloads/US_Armed_Forces_(6_2025) - Sheet1.csv")

# Clean names
military_clean <- armedForces %>%
  clean_names() %>%
  rename(
    pay_grade = active_duty_personnel_by_service_branch_sex_and_pay_grade,
    army_male = x,
    army_female = x_1,
    navy_male = x_2,
    navy_female = x_3,
    marine_corps_male = x_4,
    marine_corps_female = x_5,
    air_force_male = x_6,
    air_force_female = x_7,
    space_force_male = x_8,
    space_force_female = x_9
  ) %>%
  filter(pay_grade %in% c(
    "E1", "E2", "E3", "E4", "E5", "E6", "E7", "E8", "E9",
    "W1", "W2", "W3", "W4", "W5",
    "01", "02", "03", "04", "05", "06", "07", "08", "09", "010"
  ))
  
# Convert to long format
```

```

military_long <- military_clean %>%
  pivot_longer(
    cols = -pay_grade,
    names_to = "branch_gender",
    values_to = "count"
  ) %>%
  mutate(
    count = case_when(
      count == "N/A*" ~ 0,
      TRUE ~ as.numeric(str_remove_all(count, ","))
    )
  ) %>%
  mutate(
    branch = case_when(
      str_detect(branch_gender, "army") ~ "Army",
      str_detect(branch_gender, "navy") ~ "Navy",
      str_detect(branch_gender, "marine") ~ "Marine Corps",
      str_detect(branch_gender, "air") ~ "Air Force",
      str_detect(branch_gender, "space") ~ "Space Force"
    ),
    gender = case_when(
      str_detect(branch_gender, "male") ~ "male",
      str_detect(branch_gender, "female") ~ "female"
    )
  ) %>%
  filter(!is.na(count), count > 0) %>%
  select(branch, pay_grade, gender, count)

# Create individual records
military_individual <- military_long %>%
  uncount(count, .id = "soldier_id")

# Preview
head(military_individual)
glimpse(military_individual)

# Totals
cat("Total personnel:", sum(military_long$count), "\n")
cat("Individual records:", nrow(military_individual), "\n")

# Count summary
military_individual %>%

```

```
count(branch, pay_grade, gender) %>%
arrange(branch, pay_grade) %>%
head(20)
```

Activity 10

```
library(tidyverse)

#Choose a branch and rank group
branch_choice <- "Army"
rank_group <- c("E1","E2","E3","E4","E5","E6","E7","E8","E9")

#subset the data
subset_data <- military_individual %>%
  filter(branch == branch_choice,
         pay_grade %in% rank_group)

#create frequency table
freq_table <- subset_data %>%
  count(pay_grade, gender) %>%
  pivot_wider(
    names_from = gender,
    values_from = n,
    values_fill = 0
  ) %>%
  arrange(match(pay_grade, rank_group))

#print
print(freq_table)
```

Activity 4 (a)

```
{volume <- function(x) {}
  x * (11 - 2*x) * (8.5 - 2*x)
}
```

Activity 4 b

```
library(ggplot2)

#Box volume function for a 36 x 48 inch sheet
get_volume <- function(side_length, paper_length = 48, paper_width = 36) {
  volume <- side_length *
    (paper_width - 2 * side_length) *
    (paper_length - 2 * side_length)
  return(volume)
}

#cannot exceed half the shorter side (36/2 = 18)
max_side <- 18

#plot using stat_function
box_plot <- ggplot(data = data.frame(x = c(0, max_side)),
                     aes(x = x)) +
  stat_function(
    fun = get_volume,
    args = list(paper_length = 48, paper_width = 36),
    linewidth = 1.2
  ) +
  labs(
    title = "Figure 1. Volume of Box vs. Cut-Out Size",
    x = "Cut-out size (inches)",
    y = "Volume (cubic inches)"
  ) +
  theme_minimal()

box_plot
```