

Activity 14

IST 184

Nicholas Finn

2025-11-10

Table of contents

0.1	Armed Forces Data Redux (Activities #08 and #10)	1
0.2	Popularity of Baby Names (Activity #13)	2
0.3	Plotting a Mathematical Function (Activity #04)	3
0.4	What I've Learned	4
0.5	Code Appendix	5

This is a QMD document containing a review and analysis of three previous activities that have been done earlier in the semester, along with a section stating the most important topic(s) that I have learned throughout the semester.

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

The armed forces data that is scraped from a Google sheet starts with 29 rows, and columns for each gender, branch, and total in a very raw and untidy way. The data is tidied into a much more readable table and in the case of the table below, the focus is on the difference between rank and sex in the count of officers in the Marine Corps.

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

PayGrade	Branch	Sex	Count
O1	Marine	Male	2412
O1	Marine	Female	366
O2	Marine	Male	3162
O2	Marine	Female	525
O3	Marine	Male	5385
O3	Marine	Female	707

O4	Marine	Male	3637
O4	Marine	Female	338
O5	Marine	Male	1830
O5	Marine	Female	137
O6	Marine	Male	656
O6	Marine	Female	54
O7	Marine	Male	36
O7	Marine	Female	2
O8	Marine	Male	28
O8	Marine	Female	2
O9	Marine	Male	17
O9	Marine	Female	1
O10	Marine	Male	3
O10	Marine	Female	0

The table above shows a comparatively larger number of males in all ranks of the Marine Corps compared to women. The rank in which the ratio of males to women in the Marine Corps is the closest to 1 is the paygrade O2, where the ratio is around 6 men for each woman. In perspective of rank, the general trend is that as rank increases, both genders see a decline in count, however this is the opposite between ranks O1-O3, as the trend is the opposite and numbers climb. The rank that sees the largest numbers is the O3 rank, with 5,385 males and 707 females, and after that rank, the counts begin to decrease.

0.2 Popularity of Baby Names (Activity #13)

The Baby Names data set from the dcData package contains the popularity of baby names from 1880 - 2013 and is useful for tracking the trends of names for babies born from year to year. The data from this set is already in a tidied state and does not need to be wrangled.

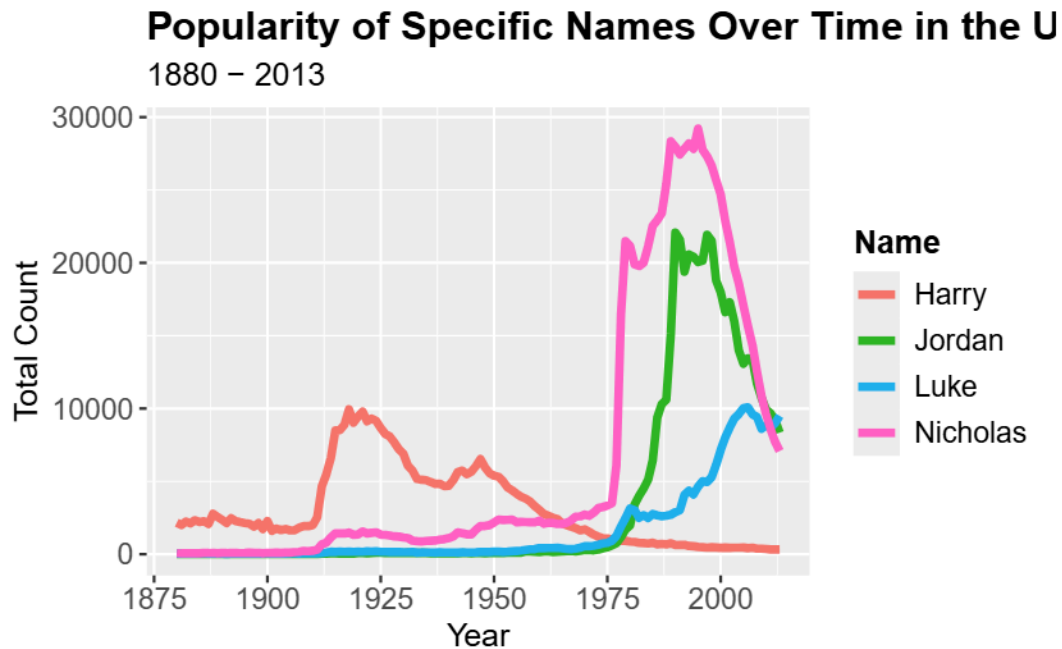


Figure 1: Visual of the Baby Names Data from the dcData Package

The visual above is a line graph that shows the popularity of four given names (Harry, Jordan, Luke, and Nicholas) from 1880-2013 in the United States. Each count is the number of babies each year given each name, regardless of gender. In the visual, the names Jordan, Luke, and Nicholas all began to see significant increases in popularity following 1975, meanwhile Harry saw continuous decline since the 1920s when it reached its peak. Reasons for the names Jordan and Luke becoming increasingly popular largely are due to the rise in fame of professional basketball player Michael Jordan and Star Wars icon Luke Skywalker, respectively. Unlike Jordan and Luke, popular culture did not result in an increase in popularity of the name Harry in the case of Harry Potter. Finally, Nicholas, which saw the greatest rise in popularity, has no definitive link to pop culture in the same sense as the others that would cause its sharp incline in use.

0.3 Plotting a Mathematical Function (Activity #04)

The mathematical function used in this activity is based off of an example for determining the largest possible volume of a box that is made by cutting the corners in of a sheet of paper and folding the sheet into an open faced box.

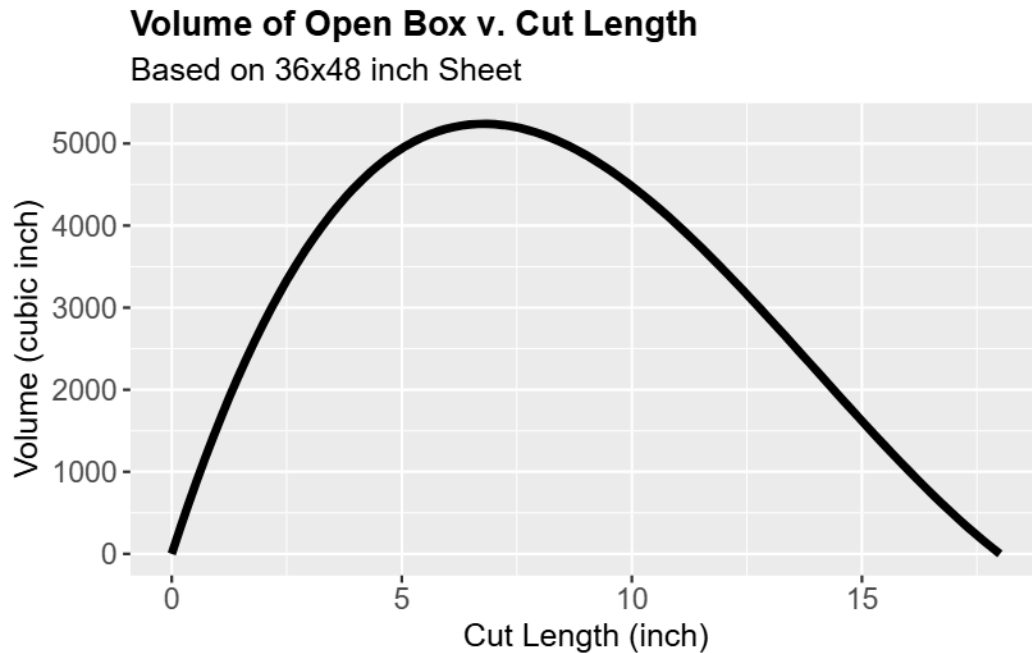


Figure 2: Visual of the getVolume Function

The graph above shows the volume of an open box versus the cut length on the corners of the sheets, and forms a hill shaped graph. The volume of the box begins at 0 as there is no cut and hence no folds to make the sides of the box. Then, as the cut size begins to increase, so does the volume of the box, where it reaches a peak at a cut length of $x = 6.789$ inches resulting in a volume of 5230.8 cubic inches.

0.4 What I've Learned

In this course, I feel that I have learned the most in how best to not only wrangle and tidy data, but also how to present it once it is tidied. This is my first time working with R and much of it has been new to me, however I have experience with programming and somethings I am familiar with. What I am not used to is having to adjust a data table that has been given to me to work with and then analyze how best to present the data down to the finest details, including the text size and line colors on the visuals once they are made. For example, I learned the most about tidying when dealing with the Armed Forces data and turning it into a tidied table, and then with the visuals I learned the most adapting the diamonds tidy data into an effective visual.

0.5 Code Appendix

```
#Step 1 Load Packages
library(googleheets4)
library(dplyr)
library(tidyr)
library(kableExtra)

#Step 2 Import Data
gs4_deauth()
armedRaw <- read_sheet(
  ss = 'https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicM1VDF7Gr-nXCb5qbwb_E/'
  skip = 2
)

#Step 3 Group Frame
groupArmed <- armedRaw %>%
  dplyr::select(-"Total...4", -"Total...7", -"Total...10", -"Total...13", -"Total...16", -"M")
#Step 4 Rename Columns
rename(
  PayGrade = "Pay Grade",
  Army_Male = Male...2,
  Army_Female = Female...3,
  Navy_Male = Male...5,
  Navy_Female = Female...6,
  Marine_Male = Male...8,
  Marine_Female = Female...9,
  AirForce_Male = Male...11,
  AirForce_Female = Female...12,
  SpaceForce_Male = Male...14,
  SpaceForce_Female = Female...15
) %>%
#Step 5 Mutate Data
mutate(across(-"PayGrade", \(x) as.numeric(as.character(x))))
) %>%
#Step 6 Pivot Columns and Group by PayGrade, Branch, Sex
pivot_longer(
  cols = -"PayGrade",
  names_sep = "_",
  names_to = c("Branch", "Sex"),
  values_to = "Count",
  values_drop_na = TRUE
)
```

```

) %>%
#Step 7 Get Rid of Extra Columns for Totals
  filter(PayGrade %in% c("01", "02", "03", "04", "05", "06", "07", "08", "09", "010"))
) %>%
  filter(Branch %in% c("Marine"))
# Make group armed forces frequency table ----
kable(groupArmed) %>%
  kable_styling(full_width = FALSE,
                position = "center",
                font_size = 12)
# Step 1 Load Packages
library(dcData)
library(ggplot2)

# Step 2 Extract Data
data(BabyNames, package = "dcData");
babyData <- data.frame(BabyNames)

# Step 3 Make New Modified Data Set
newBabyData <- babyData %>%
  group_by(year, name) %>%
  summarise(total_count = sum(count), .groups = "drop") %>%
  filter(name %in% c('Nicholas', 'Jordan', 'Luke', 'Harry'))
# Step 4 Make Visual
ggplot(newBabyData) +
  aes(x = year, y = total_count, colour = name) +
  geom_line(linewidth = 1.5) +
  scale_color_manual(
    values = c(Harry = "#F8766D",
              Jordan = "#31B425",
              Luke = "#20AFEC",
              Nicholas = "#FF61C3")
  ) +
  labs(
    x = "Year",
    y = "Total Count",
    title = "Popularity of Specific Names Over Time in the US",
    subtitle = "1880 - 2013",
    color = "Name"
  ) +
  theme_gray() +
  theme(

```

```

    plot.title = element_text(size = 15L,
                              face = "bold"),
    plot.subtitle = element_text(size = 12L),
    axis.title.y = element_text(size = 12L),
    axis.title.x = element_text(size = 12L),
    axis.text.y = element_text(size = 11L),
    axis.text.x = element_text(size = 11L),
    legend.text = element_text(size = 11L),
    legend.title = element_text(face = "bold",
                                size = 12L)
  )
# Create Function
getVolume <- function(x) {
  (36 - 2*x) * (48 - 2*x) * x
}
# Create Plot
ggplot(data.frame(x = c(0:18)), aes(x = x)) +
  stat_function(fun = getVolume, linewidth = 1.5) +
  labs(
    title = "Volume of Open Box v. Cut Length",
    subtitle = "Based on 36x48 inch Sheet",
    x = "Cut Length (inch)",
    y = "Volume (cubic inch)"
  ) +
  theme_gray() +
  theme(
    plot.title = element_text(face = "bold"),
    plot.subtitle = element_text(size = 12L),
    axis.title.y = element_text(size = 12L),
    axis.title.x = element_text(size = 12L),
    axis.text.y = element_text(size = 11L),
    axis.text.x = element_text(size = 11L)
  )
)

```