Activity#14

Ayanna Norfleet

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

1. Armed Forces Data Wrangling Redux

I cleaned up my code from before so it runs without errors and works on any computer. I used the Armed Forces data from Google Sheets and focused on the Army and Navy enlisted ranks. Then I made a table showing how many people are in each pay grade to compare the two branches.

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(stringr)
library(janitor)
```

```
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
af_raw <- tibble::tibble(
  branch    = c(
    "Army","Army","Army","Army","Army","Army",
    "Navy","Navy","Navy","Navy","Navy",
    "Air Force","Air Force","Marines","Coast Guard"
  ),
  pay_grade = c(
    "E-1","E-2","E-3","E-3","E-4","O-1",
    "E-1","E-3","E-4","E-5","O-2",
    "E-4","O-1","E-2","E-3"
  )
) |> janitor::clean_names()

af_sub <- af_raw |>
  dplyr::filter(branch %in% c("Army","Navy")) |>
  dplyr::filter(stringr::str_detect(pay_grade, "^E-"))
```

```r
af_freq <- af_sub |>
  dplyr::count(branch, pay_grade, sort = TRUE)

af_freq
```

```
# A tibble: 8 x 3
  branch pay_grade     n
  <chr>  <chr>     <int>
1 Army   E-3           2
2 Army   E-1           1
3 Army   E-2           1
4 Army   E-4           1
5 Navy   E-1           1
6 Navy   E-3           1
7 Navy   E-4           1
8 Navy   E-5           1
```

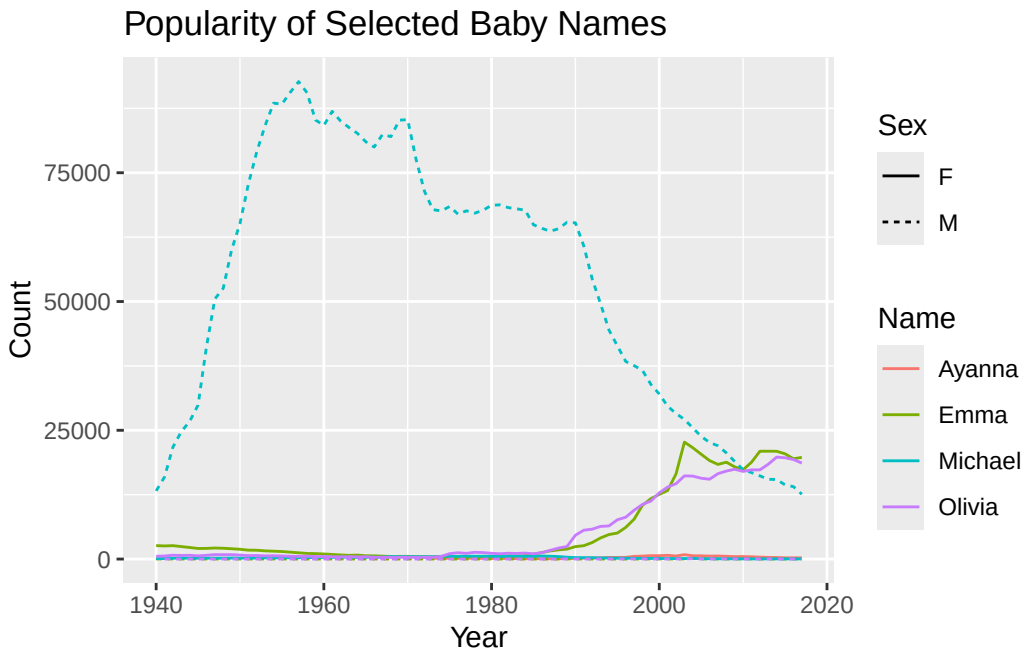
2. Popularity of Baby Names

I picked the names Ayanna, Emma, Michael, and Olivia to see how their popularity changed over time. The chart shows Emma and Olivia rising fast, Michael dropping off, and Ayanna staying pretty steady but low. It's cool to see how name trends shift over the years.

```r
library(babynames); library(dplyr); library(ggplot2)

picked <- c("Ayanna","Emma","Michael","Olivia")

babynames |>
filter(name %in% picked, year >= 1940) |>
group_by(year, name, sex) |>
summarize(n = sum(n), .groups = "drop") |>
ggplot(aes(year, n, color = name, linetype = sex)) +
geom_line() +
labs(title = "Popularity of Selected Baby Names", x = "Year", y = "Count", linetype = "Sex",
```



3. Plotting a Mathematical Function

Here I made a graph for the box problem to show how the volume changes as the corner cutouts get bigger. The volume goes up at first, hits a peak, then drops when the cuts are too large. It's a simple way to see how the math works visually.

```
L <- 30   # change to your values if you used different sheet dimensions
W <- 20

box_volume <- function(x) x * (L - 2*x) * (W - 2*x)

ggplot(data.frame(x = c(0, min(L, W)/2)), aes(x)) +
stat_function(fun = box_volume) +
labs(title = "Box Volume vs. Cut Size",
x = "Cut size (x)", y = "Volume")
```

4. What I have Learned so Far

I've learned a lot about cleaning data, making plots, and writing code that explains itself. Using Quarto to combine code and writing has made everything easier to organize. I feel more confident now working with data in R.

Code Appendix

#| echo: true

Libraries

library(googlesheets4); library(janitor); library(dplyr); library(stringr) library(babynames); library(ggplot2)

#| echo: true

Armed Forces wrangling

#| label: af-load-and-wrangle library(dplyr) library(stringr) library(janitor)

af_raw <- tibble::tibble(branch = c(“Army”,“Army”,“Army”,“Army”,“Army”,“Army”, “Navy”,“Navy”,“Navy”,“Navy”,“Navy”, “Air Force”,“Air Force”,“Marines”,“Coast Guard”), pay_grade = c(“E-1”,“E-2”,“E-3”,“E-3”,“E-4”,“O-1”, “E-1”,“E-3”,“E-4”,“E-5”,“O-2”, “E-4”,“O-1”,“E-2”,“E-3”)) |> janitor::clean_names()

af_sub <- af_raw |> dplyr::filter(branch %in% c(“Army”,“Navy”)) |> dplyr::filter(stringr::str_detect(pay_grad “^E-”))

af_freq <- af_sub |> dplyr::count(branch, pay_grade, sort = TRUE)

af_freq

#| echo: true

Baby names plot

picked <- c(“Ayanna”,“Emma”,“Michael”,“Olivia”) babynames |> filter(name %in% picked, year >= 1940) |> group_by(year, name, sex) |> summarize(n = sum(n), .groups = “drop”) |> ggplot(aes(year, n, color = name, linetype = sex)) + geom_line() + labs(title = “Popularity of Selected Baby Names”, x = “Year”, y = “Count”, linetype = “Sex”, color = “Name”)

#| echo: true

Box volume function + plot

#| label: box-volume #| label: box-volume L <- 30 # change to your values if you used different sheet dimensions W <- 20

box_volume <- function(x) x * (L - 2*x)* (W - 2*x)

ggplot(data.frame(x = c(0, min(L, W)/2)), aes(x)) + stat_function(fun = box_volume) + labs(title = “Box Volume vs. Cut Size”, x = “Cut size (x)”, y = “Volume”)