Activity 14

Ekansh Bharadwaj

2025-11-09

Activity 14

Armed Forces Data

Question 3: Data Wrangling Code for Armed Forces Data

code and make sure that codes meets expectation. This item is worth a large number of points so that it can off-set poor performance on Question 5 of Activity #08.

For scoring this item, look at the Code Appendix for clearly labelled section of code for wrangling the Armed Forces data, to build a data frame where the case is an individual solider and contains rank names. This portion of the code MUST be 100% reproducible by someone else who will run the code verbatim (no alterations or edits will be made).

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```r
library(rvest)
```

```

Attaching package: 'rvest'

The following object is masked from 'package:readr':

    guess_encoding
```

```r
library(googlesheets4)

ranks_html <- read_html("https://neilhatfield.github.io/Stat184_PayGradeRanks.html")

ranks_raw_tbls <- ranks_html |>
  html_elements(css = "table") |>
  html_table()

ranks_raw <- ranks_raw_tbls[[1]]

ranks_raw[1, 1] <- "Type"
header_row <- ranks_raw[1, ]
names(ranks_raw) <- header_row[1, ]
```

```
Warning: The `value` argument of `names<-()` must be a character vector as of tibble
3.0.0.
```

```r
ranks_raw <- ranks_raw[-c(1, 26), ]

rank_lookup <- ranks_raw |>
  dplyr::select(!Type) |>
  pivot_longer(
    cols = !`Pay Grade`,
    names_to = "branch",
    values_to = "rank"
  ) |>
  mutate(rank = na_if(rank, "--")) |>
  mutate(branch = as.character(branch)) |>
  arrange(branch, `Pay Grade`)

gs4_deauth()

forces_url <- "https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qbw

forces_head <- read_sheet(
  ss = forces_url,
  col_names = FALSE,
  n_max = 3
)
```

```
v Reading from "US_Armed_Forces_(6/2025)".
v Range 'Sheet1'.
New names:
```

```r
forces_raw <- read_sheet(
  ss = forces_url,
  col_names = FALSE,
  skip = 3,
  n_max = 28,
  na = c("N/A*")
)
```

```
v Reading from "US_Armed_Forces_(6/2025)".
v Range '4:10000000'.
New names:
```

```r
branch_names <- rep(
  c("Army", "Navy", "Marine Corps", "Air Force", "Space Force", "Total"),
  each = 3
)

temp_headers <- paste(
  c("", branch_names),
  forces_head[3, ],
  sep = "."
)

names(forces_raw) <- temp_headers

af_groups_jun2025 <- forces_raw |>
  rename(pay_grade = `.Pay Grade`) |>
  dplyr::select(!starts_with("Total.")) |>
  filter(!grepl("^Total", pay_grade)) |>
  pivot_longer(
    cols = !pay_grade,
    names_to = "branch_sex",
    values_to = "n"
  ) |>
  separate_wider_delim(
    cols = branch_sex,
    delim = ".",
    names = c("branch", "sex")
  ) |>
  filter(sex %in% c("Female", "Male")) |>
  left_join(
    y = rank_lookup,
    by = join_by(pay_grade == `Pay Grade`, branch == branch)
  ) |>
  mutate(
    category = case_when(
      grepl("^E\\d+", pay_grade) ~ "Enlisted",
      grepl("^W\\d+", pay_grade) ~ "Warrant",
      grepl("^O\\d+", pay_grade) ~ "Officer",
      TRUE ~ "Other"
    )
  ) |>
  relocate(branch, sex, category, pay_grade, rank, .after = pay_grade) |>
  arrange(branch, sex, pay_grade)

af_individuals_jun2025 <- af_groups_jun2025 |>
  filter(!is.na(n)) |>
  tidyr::uncount(weights = n, .remove = FALSE, .id = "row_in_group") |>
  group_by(branch, sex, pay_grade) |>
  mutate(seq_in_group = dplyr::row_number()) |>
  ungroup() |>
  mutate(
    soldier_id = stringr::str_c(
      stringr::str_replace_all(branch, "\\s+", ""),
      "_", sex, "_", pay_grade, "_", seq_in_group
    )
  ) |>
  dplyr::select(soldier_id, branch, sex, category, pay_grade, rank)
```

Question 4: Visualization for the Armed Forces

For this section, we want you to create a two-way frequency table to explore the impact of sex and rank in the US Armed Forces. You may choose to which rank to focus on (Enlisted, Warrant Offices, or Officers) as well as which particular branch (Army, Navy, Marine Corps, Air Force, Space Force). You may not choose a combination that yields zero observations (e.g., Warrent Officers in the Space Force).

Does the student create a frequency table that would allow them to explore whether sex and rank are independent of each other with respect to their chosen sub-group of the armed forces?

```
army_enlisted_jun2025 <- af_individuals_jun2025 |>
filter(
branch    == "Army",
category == "Enlisted",
!is.na(rank)
)

army_enlisted_rank_sex_tbl <- army_enlisted_jun2025 |>
count(rank, sex, name = "n") |>
arrange(rank, sex) |>
pivot_wider(
names_from  = sex,
values_from = n,
```

```r
values_fill = 0
) |>
arrange(rank)

army_enlisted_rank_sex_tbl
```

```
# A tibble: 8 x 3
  rank                                        Female  Male
  <chr>                                        <int> <int>
1 Corporal OR Specialist                       15143 79234
2 First Sergeant OR Master Sergeant             1472  9482
3 Private                                       5662 29767
4 Private First Class                          10229 43775
5 Sergeant                                     10954 54803
6 Sergeant First Class                          4410 30264
7 Sergeant Major OR Command Sergeant Major       394  2865
8 Staff Sergeant                                7363 49502
```

Question 5: Narrative Text for the Armed Forces Section

Does the student include text describing what we can learn from their data visualization? Do they address whether sex and rank are independent of each other for their selected sub-group of the US Armed Forces?

The table shows that men consistently outnumber women across all enlisted ranks in the Army, indicating a strong relationship between rank and sex rather than independence.

Popularity od Baby Names

Question 6: Code for the Popular Baby Names Project

For scoring this item, we will focus on the student's code to 1) clean and prepare the baby names data, and 2) create the data visualization. (Focus just on the code for creating the data visualization for this item.)

Look at the Code Appendix for clearly labelled code for the Popular Names/Baby Names section. There should be code that 1) loads the babyNames data, 2) filters that data to their chosen sub-set of names, regardless of the sex, and 3) creates a times series plot for their chosen sub-set of names.

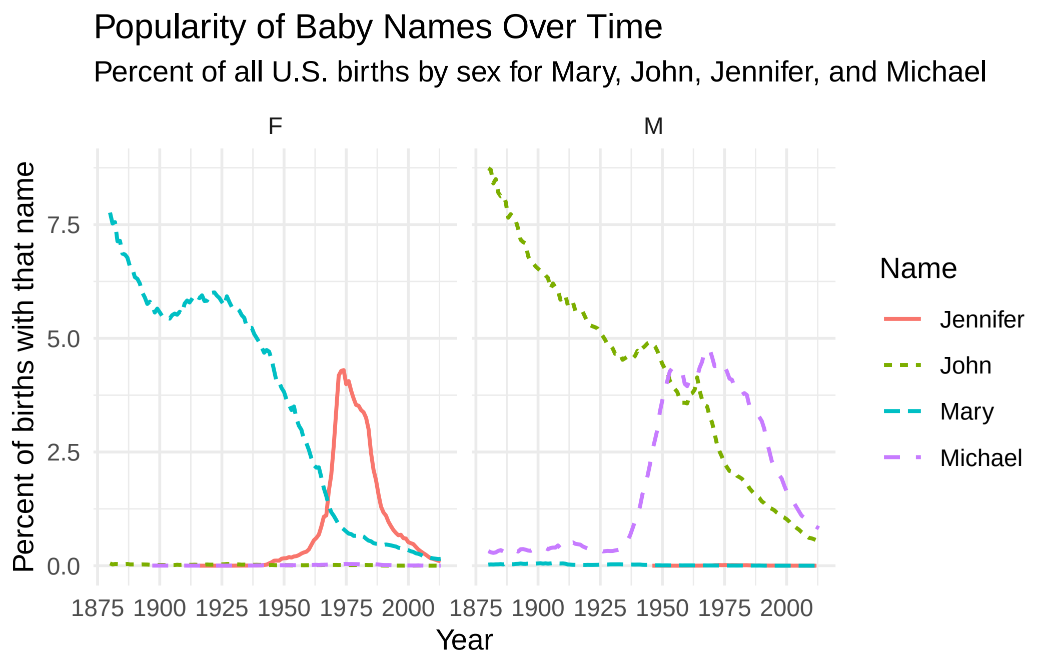
```r
library(dcData)
library(dplyr)
library(tidyr)
library(ggplot2)
data("BabyNames")
```

```r
names_vec <- c("Mary", "John", "Jennifer", "Michael")

baby_selected <- BabyNames |>
filter(name %in% names_vec)

totals_by_sex_year <- BabyNames |>
group_by(sex, year) |>
summarise(
total_births = sum(count),
.groups      = "drop"
)

baby_pct <- baby_selected |>
group_by(name, sex, year) |>
summarise(
count   = sum(count),
.groups = "drop"
) |>
left_join(
y  = totals_by_sex_year,
by = join_by(sex, year)
) |>
mutate(
pct_of_births = 100 * count / total_births
) |>
ungroup()

baby_names_ts_plot <- ggplot(
data    = baby_pct,
mapping = aes(
x        = year,
y        = pct_of_births,
color    = name,
linetype = name
)
) +
geom_line(linewidth = 0.7) +
facet_wrap(vars(sex)) +
labs(
title    = "Popularity of Baby Names Over Time",
subtitle = "Percent of all U.S. births by sex for Mary, John, Jennifer, and Michael",
x        = "Year",
y        = "Percent of births with that name",
color    = "Name",
linetype = "Name"
) +
theme_minimal()
```

Question 7: Visualization for Popular Baby Names Project

Does the student create a time series plot for their chosen sub-set of baby names? Does this visualization show evidence of applying what they have learned about EPTs, and the principles of effective graphics from Tufte and Kosslyn? Focus on the visual appearance of their visualization(s), not their generating code here.

```
baby_names_ts_plot
```



Question 8: Narrative Text for the Popular Baby Names Project

Does the student include text describing what we can learn from their data visualization? Do they explain why they chose the names that they did?

The plot shows how the popularity of Mary, John, Jennifer, and Michael has changed over time, with each name rising and falling during different generations. Faceting by sex makes it easy to compare trends, and the lines show clear peaks followed by declines.

Plotting a Mathematical Function

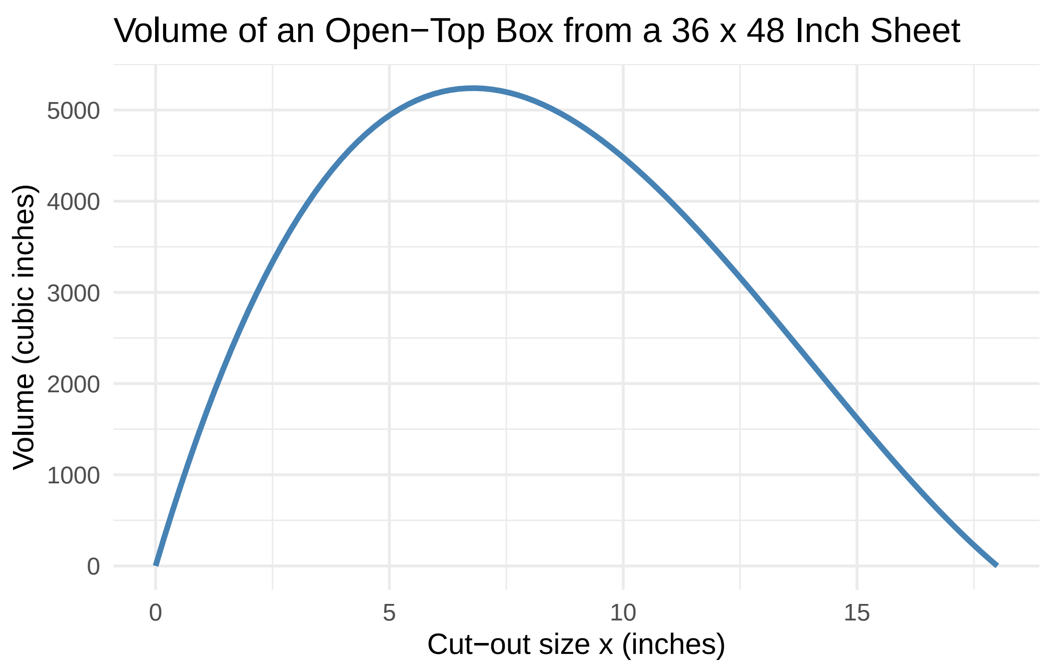
Question 9: Code for the Box Problem

For this assignment, please adjust your function to use a piece of paper that is 36 inches by 48 inches. There are two parts to this item: code and the resulting visualization. Both will be used for scoring purposes.

Code Look at the Code Appendix for clearly labelled code for the Box Problem. There should be a defined function for the box problem AND code to create a plot of the function using the {ggplot2} package. The hint is to look at the documentation of stat_function.

Visualization Look in the body of their submission for a clearly labelled plot for the Box Problem. The visualization should show the graph of their function, be properly labelled, clear, appropriately sized, and have a numbered figure caption.

```
library(ggplot2)
library(dplyr)

box_volume <- function(x) {
x * (36 - 2*x) * (48 - 2*x)
}

x_vals <- seq(0, 18, length.out = 500)


box_plot <- ggplot(data.frame(x = x_vals), aes(x = x)) +
stat_function(
fun  = box_volume,
linewidth = 1,
color = "steelblue"
) +
labs(
title    = "Volume of an Open-Top Box from a 36 x 48 Inch Sheet",
x        = "Cut-out size x (inches)",
y        = "Volume (cubic inches)"
) +
theme_minimal()

box_plot
```



Question 10: Narrative Text for the Box Problem

Does the student include text describing what we can learn from their data visualization AND answer the question(s) originally posed about the Box Problem (what's the max volume and for what side length) for the 36 inch X 48 inch piece of paper.

The plot shows that the box's volume increases as the cut-out size grows until it reaches a clear peak and then declines as the folds become too large. For a 36 $\times$ 48 inch sheet, the maximum volume occurs at about $x \approx 6$ inches, producing a volume of roughly 5,200 cubic inches.

Self Reflection

Throughout this course I have learnt how to wrangle and reshape unclean data which helped me turn real datasets into tidy useable forms, for example converting armed forces data into a clean dataset, also developed strong data visualization skills from readings such as Tufte and Kosslyn. I learnt hoe to mathematically model problems in R such as the box problem and interpreting the graphs it vreates

Code Appendix

###Problem 3

```
library(tidyverse)
library(rvest)
library(googlesheets4)
```

```r
ranks_html <- read_html("https://neilhatfield.github.io/Stat184_PayGradeRanks.html")

ranks_raw_tbls <- ranks_html |>
  html_elements(css = "table") |>
  html_table()

ranks_raw <- ranks_raw_tbls[[1]]

ranks_raw[1, 1] <- "Type"
header_row <- ranks_raw[1, ]
names(ranks_raw) <- header_row[1, ]

ranks_raw <- ranks_raw[-c(1, 26), ]

rank_lookup <- ranks_raw |>
  dplyr::select(!Type) |>
  pivot_longer(
    cols = !`Pay Grade`,
    names_to = "branch",
    values_to = "rank"
  ) |>
  mutate(rank = na_if(rank, "--")) |>
  mutate(branch = as.character(branch)) |>
  arrange(branch, `Pay Grade`)

gs4_deauth()

forces_url <- "https://docs.google.com/spreadsheets/d/19xQnI1cBh6Jkw7eP8YQuuicMlVDF7Gr-nXCb5qb

forces_head <- read_sheet(
  ss = forces_url,
  col_names = FALSE,
  n_max = 3
)
```

```
v Reading from "US_Armed_Forces_(6/2025)".
```

```
v Range 'Sheet1'.
```

```
New names:
* `` -> `...1`
* `` -> `...2`
* `` -> `...3`
* `` -> `...4`
* `` -> `...5`
* `` -> `...6`
* `` -> `...7`
* `` -> `...8`
* `` -> `...9`
* `` -> `...10`
* `` -> `...11`
* `` -> `...12`
* `` -> `...13`
* `` -> `...14`
* `` -> `...15`
* `` -> `...16`
* `` -> `...17`
* `` -> `...18`
* `` -> `...19`
```

```r
forces_raw <- read_sheet(
  ss = forces_url,
  col_names = FALSE,
  skip = 3,
  n_max = 28,
  na = c("N/A*")
)
```

```
v Reading from "US_Armed_Forces_(6/2025)".
v Range '4:10000000'.
New names:
```

```r
branch_names <- rep(
  c("Army", "Navy", "Marine Corps", "Air Force", "Space Force", "Total"),
  each = 3
)

temp_headers <- paste(
  c("", branch_names),
  forces_head[3, ],
  sep = "."
)

names(forces_raw) <- temp_headers

af_groups_jun2025 <- forces_raw |>
  rename(pay_grade = `.Pay Grade`) |>
  dplyr::select(!starts_with("Total.")) |>
  filter(!grepl("^Total", pay_grade)) |>
  pivot_longer(
    cols = !pay_grade,
    names_to = "branch_sex",
    values_to = "n"
  ) |>
  separate_wider_delim(
    cols = branch_sex,
    delim = ".",
    names = c("branch", "sex")
  ) |>
  filter(sex %in% c("Female", "Male")) |>
  left_join(
    y = rank_lookup,
    by = join_by(pay_grade == `Pay Grade`, branch == branch)
  ) |>
  mutate(
    category = case_when(
      grepl("^E\\d+", pay_grade) ~ "Enlisted",
      grepl("^W\\d+", pay_grade) ~ "Warrant",
      grepl("^O\\d+", pay_grade) ~ "Officer",
      TRUE ~ "Other"
    )
  ) |>
  relocate(branch, sex, category, pay_grade, rank, .after = pay_grade) |>
  arrange(branch, sex, pay_grade)

af_individuals_jun2025 <- af_groups_jun2025 |>
  filter(!is.na(n)) |>
  tidyr::uncount(weights = n, .remove = FALSE, .id = "row_in_group") |>
  group_by(branch, sex, pay_grade) |>
  mutate(seq_in_group = dplyr::row_number()) |>
  ungroup() |>
  mutate(
    soldier_id = stringr::str_c(
      stringr::str_replace_all(branch, "\\s+", ""),
      "_", sex, "_", pay_grade, "_", seq_in_group
    )
  ) |>
  dplyr::select(soldier_id, branch, sex, category, pay_grade, rank)
```

Problem 4

```
army_enlisted_jun2025 <- af_individuals_jun2025 |>
filter(
branch   == "Army",
category == "Enlisted",
!is.na(rank)
)
```

```r
army_enlisted_rank_sex_tbl <- army_enlisted_jun2025 |>
count(rank, sex, name = "n") |>
arrange(rank, sex) |>
pivot_wider(
names_from  = sex,
values_from = n,
values_fill = 0
) |>
arrange(rank)

army_enlisted_rank_sex_tbl
```

```
# A tibble: 8 x 3
  rank                                      Female  Male
  <chr>                                      <int> <int>
1 Corporal OR Specialist                     15143 79234
2 First Sergeant OR Master Sergeant           1472  9482
3 Private                                     5662 29767
4 Private First Class                        10229 43775
5 Sergeant                                   10954 54803
6 Sergeant First Class                        4410 30264
7 Sergeant Major OR Command Sergeant Major     394  2865
8 Staff Sergeant                              7363 49502
```

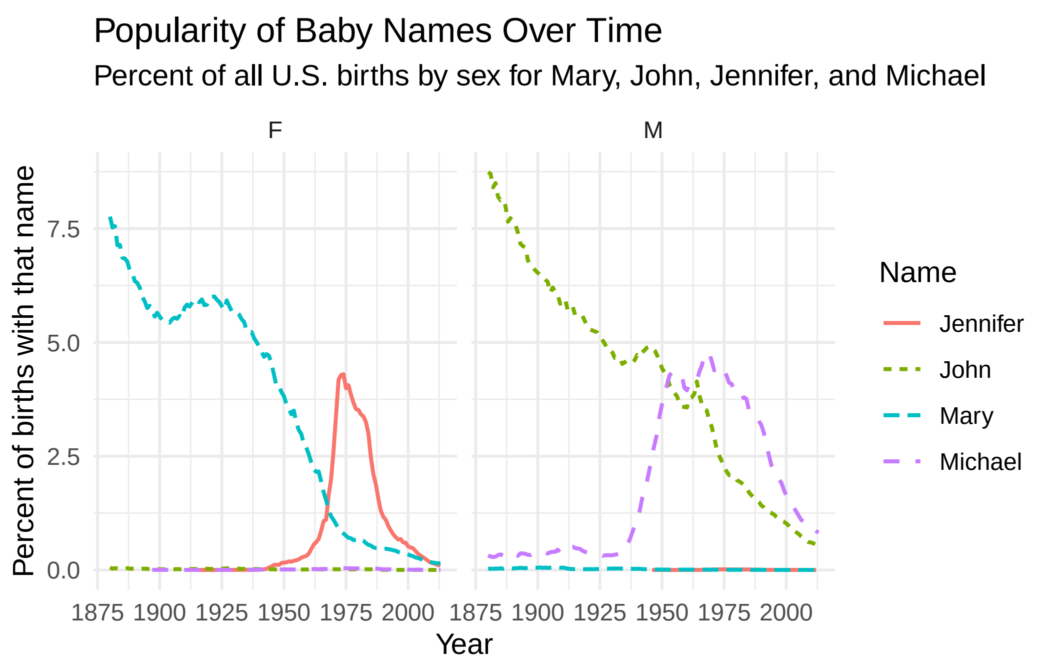
Problem 6

```r
library(dcData)
library(dplyr)
library(tidyr)
library(ggplot2)
data("BabyNames")


names_vec <- c("Mary", "John", "Jennifer", "Michael")

baby_selected <- BabyNames |>
filter(name %in% names_vec)

totals_by_sex_year <- BabyNames |>
group_by(sex, year) |>
summarise(
total_births = sum(count),
.groups      = "drop"
)
```

```
baby_pct <- baby_selected |>
group_by(name, sex, year) |>
summarise(
count   = sum(count),
.groups = "drop"
) |>
left_join(
y  = totals_by_sex_year,
by = join_by(sex, year)
) |>
mutate(
pct_of_births = 100 * count / total_births
) |>
ungroup()

baby_names_ts_plot <- ggplot(
data    = baby_pct,
mapping = aes(
x        = year,
y        = pct_of_births,
color    = name,
linetype = name
)
) +
geom_line(linewidth = 0.7) +
facet_wrap(vars(sex)) +
labs(
title    = "Popularity of Baby Names Over Time",
subtitle = "Percent of all U.S. births by sex for Mary, John, Jennifer, and Michael",
x        = "Year",
y        = "Percent of births with that name",
color    = "Name",
linetype = "Name"
) +
theme_minimal()
```

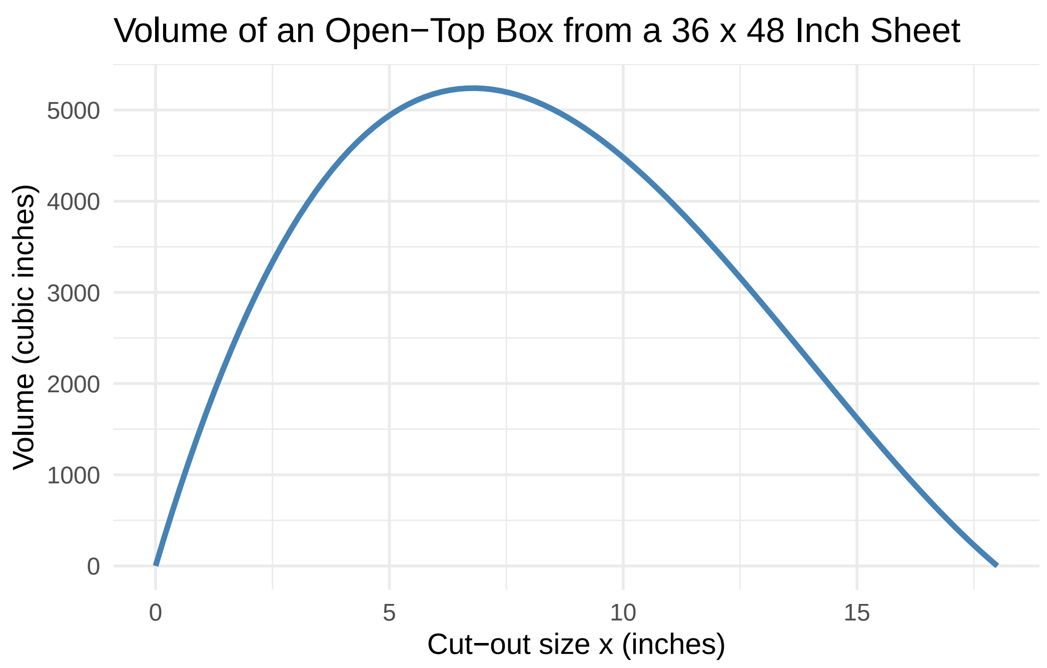
Problem 7

```
baby_names_ts_plot
```



Problem 9

```r
library(ggplot2)
library(dplyr)

box_volume <- function(x) {
x * (36 - 2*x) * (48 - 2*x)
}

x_vals <- seq(0, 18, length.out = 500)


box_plot <- ggplot(data.frame(x = x_vals), aes(x = x)) +
stat_function(
fun   = box_volume,
linewidth = 1,
color = "steelblue"
) +
labs(
title    = "Volume of an Open-Top Box from a 36 x 48 Inch Sheet",
x        = "Cut-out size x (inches)",
y        = "Volume (cubic inches)"
) +
theme_minimal()

box_plot
```


Volume of an Open−Top Box from a 36 x 48 Inch Sheet
Volume (cubic inches)
5000
4000
3000
2000
1000
0
0
5
10
15
Cut−out size x (inches)