# Activity #10: Reproducible RMDs

Maeric Barrows

2023-11-21

## Prologue: Introducing the Assignment

This R Markdown file is an extension of Activity 9: A First RMD File. This project was made in conjunction with GitHub. The link to the GitHub repository is provided here: https://github.com/maeric75/STAT184_Activity_10

(Note for Prof. Hatfield: In the GitHub repository, the file name is Assignment9. This is the file I initially linked and worked with for this assignment. For the purpose of submission, I have made a copy of this assignment called Activity10. The knitted html and pdf files will have this name, too.)

(Note for me: I (Maeric as of 11/21/2023) did not change any of the material from Activity 9. I just made the RMD file more accessible for future me. I hope I helped you.)

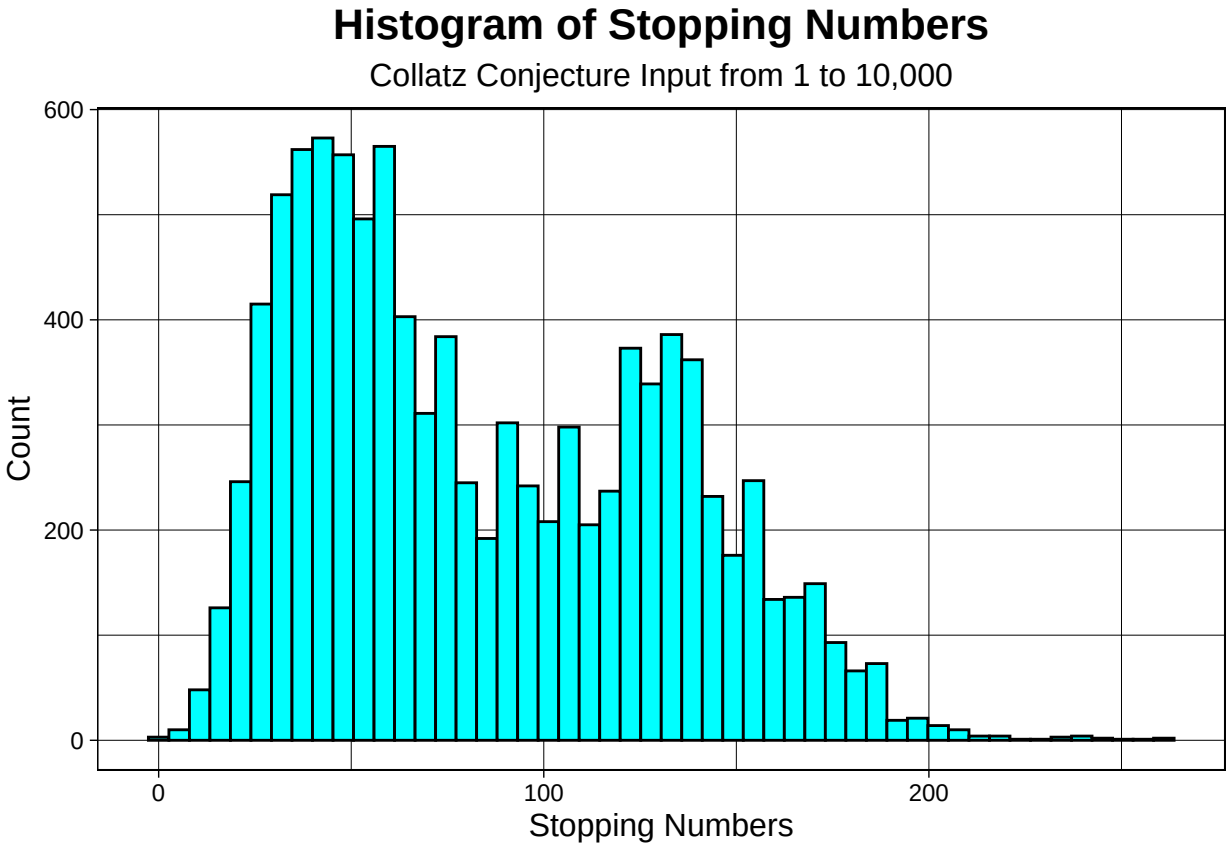
## Section 1: Collatz Conjecture

The Collatz conjecture, a famous function in mathematics, tests whether every integer eventually becomes the number 1 by running the number through two arithmetic operations. These are the operations present in the conjecture:

- If n (the integer being tested) is even: n = n / 2
- If n is odd: n = 3n + 1
- If n = 1: Stop the conjecture

In Assignment 3 of STAT 184, Prof. Hatfield asked the students to find the "stopping times" for the first 10,000 integers greater than 0, and create a histogram of these stopping times. A stopping time is how many times a number goes through the Collatz conjecture before it becomes the number 1. To find this, we must first create code which will properly evaluate the Collatz conjecture. Then, we will use the sapply function so that the Collatz conjecture runs from integers 1 to 10,000. After this, we will turn our Collatz data into a data frame.

Then, we will use ggplot2 to create a histogram using the data frame:

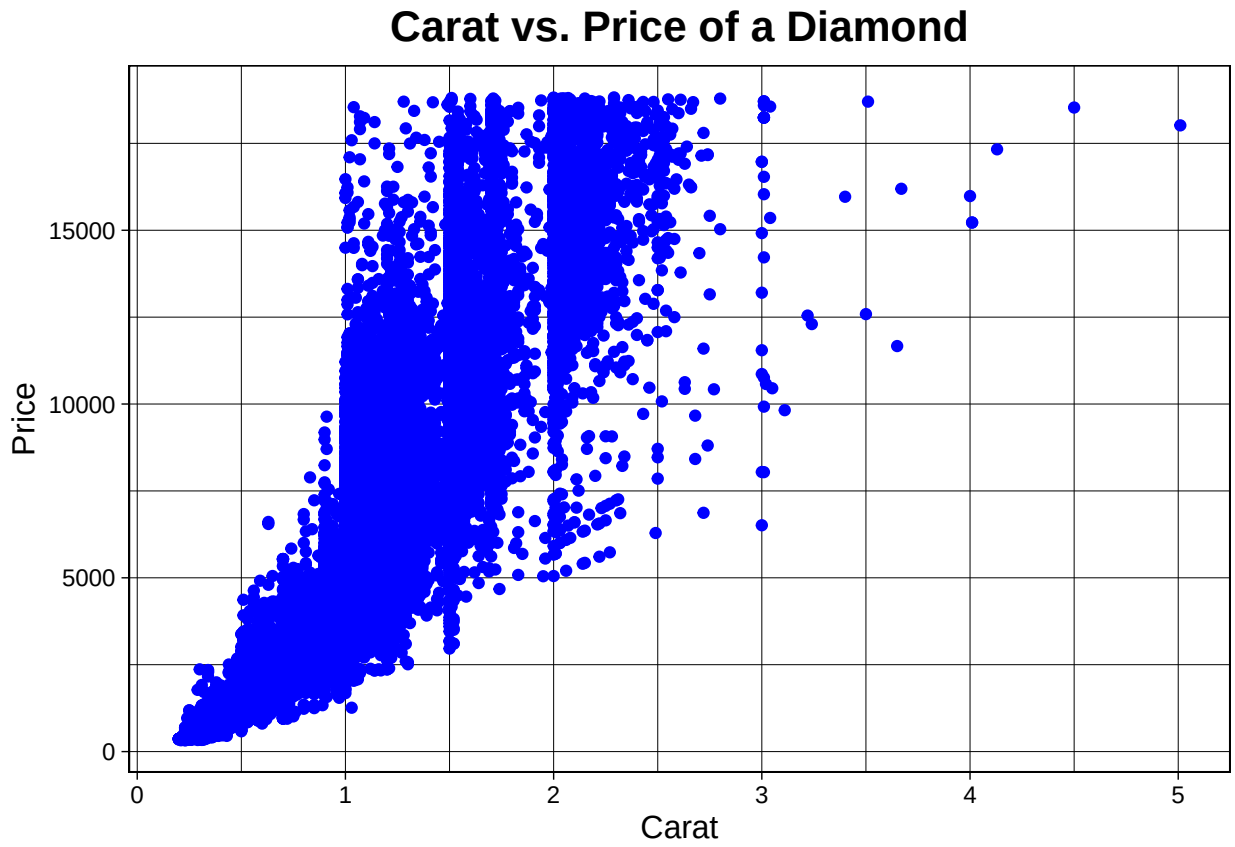



To make a conclusion about the data, we can observe the "flow" of the histogram. The bars close to the stopping number 0 are very short, followed by taller and taller bars until we get to just above the stopping number 50, where the bars start to get shorter again. In between stopping numbers 100 and 150, the bars generally increase in height for a bit, but go back to steadily decreasing in height after. The height of a bar corresponds to how many times a stopping number occurs. Our observations tell us that the very smallest and very largest stopping numbers do not occur very often, and that a significant amount of integers between 1 and 10,000 stop after about 50 recursions of the Collatz conjecture. The amount of integers that stop after about 125 recursions is also greater than the numbers which stop after 100 or 150 recursions.

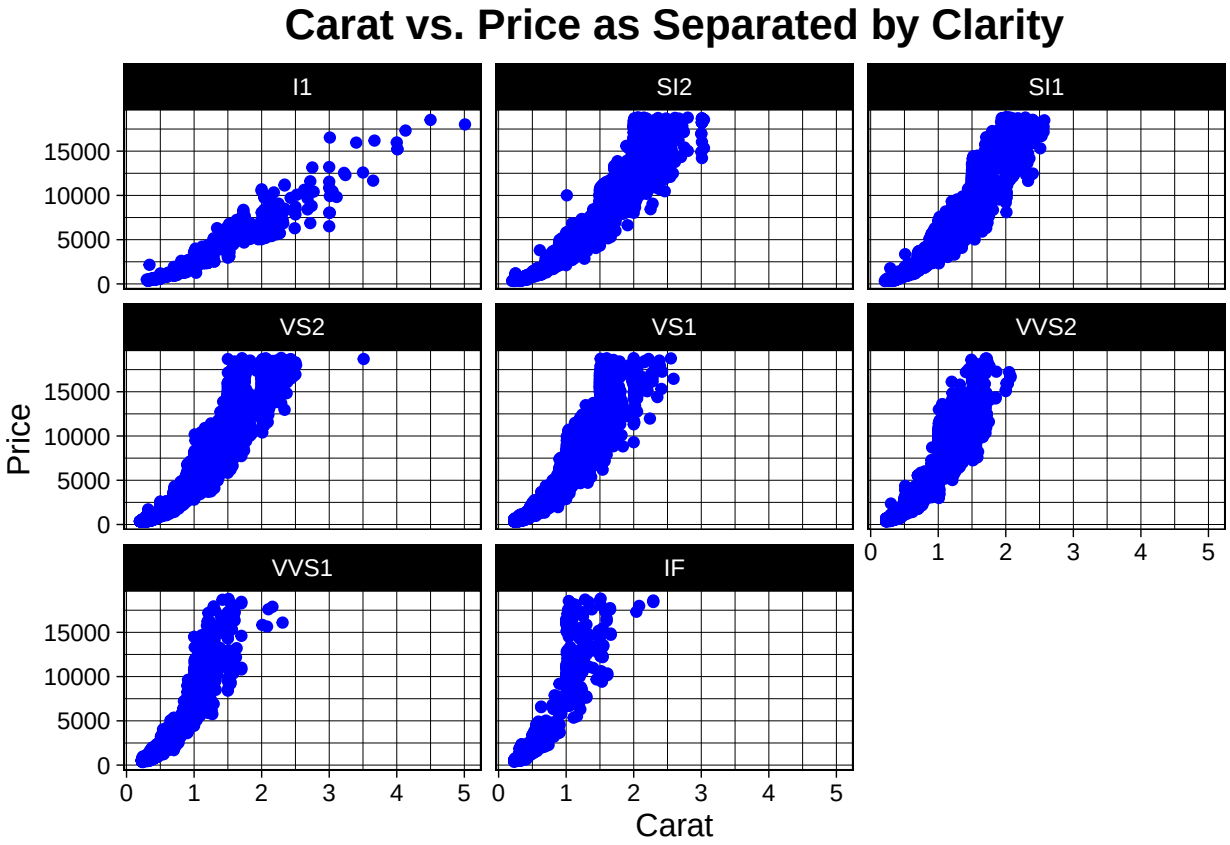
## Section 2: The Price of Diamonds

In addition to creating data visualizations, the ggplot2 package features a data set about diamonds. The data set features data for 53,940 diamonds, measuring them based on many variables, like carat (weight), cut quality, clarity, color, width (x), height (y), and depth (z), among others. All of these attributes contribute to a diamond's price. We can see the relationship between these variables and price by creating data visualizations and tables.

First, we'll create a data visualization comparing the carat of a diamond and its price. Since there are so many diamonds in the data set, and we are looking for a general trend when comparing carat and price, a scatter plot is a good way to demonstrate this data.




This visualization shows that as the carat, or weight, of a diamond increases, the price also generally increases. This comparison is simple. If we want to add more variables to this graph, we can split all of the diamonds by their clarity. The scatter plot below still compares carat and price, but also compares those two variables to a diamond's clarity:




This graph also shows that price increases as carat increases. This visualization adds another comparison point with the clarity of the diamond. We can see that the plots that represent lower clarity diamonds (grouped in the top row of scatter plots) reach the highest prices further to the right than the plots with higher clarity (grouped in the middle and bottom row). This proves that as a diamond's clarity increases, its price generally increases as well.

Visualizations are not the only way to demonstrate data and make conclusions about what results in a diamond's price. We can also create summary tables. Here is code that will create a summary table of the cut of a diamond and its price:

```
##         cut price_min price_Q1 price_median price_Q3 price_max price_mean
## 1      Fair       337  2050.25       3282.0  5205.50     18574   4358.758
## 2      Good       327  1145.00       3050.5  5028.00     18788   3928.864
## 3 Very Good       336   912.00       2648.0  5372.75     18818   3981.760
## 4   Premium       326  1046.00       3185.0  6296.00     18823   4584.258
## 5     Ideal       326   878.00       1810.0  4678.50     18806   3457.542
##   price_stdev count
## 1    3560.387  1610
## 2    3681.590  4906
## 3    3935.862 12082
## 4    4349.205 13791
## 5    3808.401 21551
```

This summary table provides a lot of statistics, some of which are confusing to decipher. Before creating the table, I assumed that diamonds with an "Ideal" cut would have a higher price than those with lower quality cuts, like "Fair" or "Good." In reality, the data does not support that hypothesis. "Ideal" diamonds actually

have the lowest minimum, first quartile, median, third quartile, and average price, the exact opposite of what I would expected. The lowest quality cut, "Fair," has the highest minimum, first quartile, and median price, while "Premium" has the highest third quartile and mean price. "Ideal" is very close to first in maximum price, but is still slightly behind "Very Good" and "Premium." This does not necessarily mean that "Ideal" diamonds absolutely have the lowest prices. Since there are much more "Ideal" diamonds than any other in this study, as shown by the "count" column, it is possible that the data of the "Ideal" column is more accurate than the other cut types. It is difficult to make a concrete conclusion, but using this data, we can infer that cut quality has little to no effect on the price of a diamond.

## Section 3: Course Takeaways (So Far)

For Section 3 of this RMD assignment, Prof. Hatfield asked us to reflect on what we have learned in the course. Something that I have always thought, since the beginning of the course, is that R is an incredibly overwhelming software to use. There are so many libraries of functions that could be used to solve any data-related problem you could think of. I am sure that I have not touched more than 2% of what R has to offer in this course, and the 2% I have seen is still filled with so many options for what to do that I find myself extremely overwhelmed in every assignment. Thus, with there being a lot to learn about R, I have already learned a lot. What follows is my attempt to bring up some things not touched on in this assignment already.

**Learned Thing 1. PCIP**

When first introduced to PCIP, or "Plan, Code, Improve, Polish," I was doubtful that it would help me code in R. From the beginning, I liked to throw code at the wall and see what stuck. This worked in Python classes, but I found R to be an entirely different beast. As I mentioned, everything was so overwhelming for me. To fix this, I needed to slow my process down, and organize it a little better. PCIP has gotten me out of many ruts in this class, and I am thankful that I started taking the process seriously.

**Learned Thing 2. Wrangling and Cleaning**

When I first heard of data wrangling, I was excited, because I hoped it would pertain to a hobby of mine. I create my own Top 40 hit songs chart every week, in the style of the Billboard Hot 100, which involves taking data and reinterpret it. Making one of these charts typically takes hours, and I thought that data wrangling would automate the process a little. Unfortunately, I have not learned of a way to speed up my personal process yet, as the data I use comes in the form of an image, but if I can find data in other forms, like spreadsheets, I will be able to use my new-found data wrangling and cleaning skills to automate my Top 40 chart more than I already have. Below is the result of data wrangling I did from a previous assignment. I was very proud of figuring this one out.

```
## New names:
## * `` -> `...1`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...15`
## * `` -> `...16`
```

```
## # A tibble: 192 x 4
##    `Pay Grade` `Marital Status`        Sex    Count
##    <chr>       <chr>                   <chr>  <dbl>
```

```
##  1 E-1         Single Without Children Male    9456
##  2 E-1         Single Without Children Female  1309
##  3 E-1         Single With Children    Male     365
##  4 E-1         Single With Children    Female    80
##  5 E-1         Joint Service Marriage  Male      40
##  6 E-1         Joint Service Marriage  Female    38
##  7 E-1         Civilian Marriage       Male    2579
##  8 E-1         Civilian Marriage       Female   358
##  9 E-2         Single Without Children Male   21600
## 10 E-2         Single Without Children Female  3324
## # i 182 more rows
```

### Learned Thing 3. Perseverance

This will sound corny, but this class has taught me to persevere and work hard. In high school, I was used to things coming easy to me. I would pick up concepts quickly, and classes felt like a breeze. Here, things changed. I often felt like I was making no progress and that I was hitting a wall in my learning. Most assignments in this class take hours for me to complete, with more than a few taking days, something I was not accustomed to. Taking this class, and many others in my first semester, has been humbling. Even though I do not enjoy the stressful process of struggling, the feeling of pride at the end for completing a tough assignment is ultimately worth it. It is difficult to convey my level of perseverance in a data visualization, but this class has certainly made me realize how much it rewards me, so a visualization for my attitude about perseverance might look like a line graph trending upwards.

## Epilogue: Screenshot for Prof. Hatfield

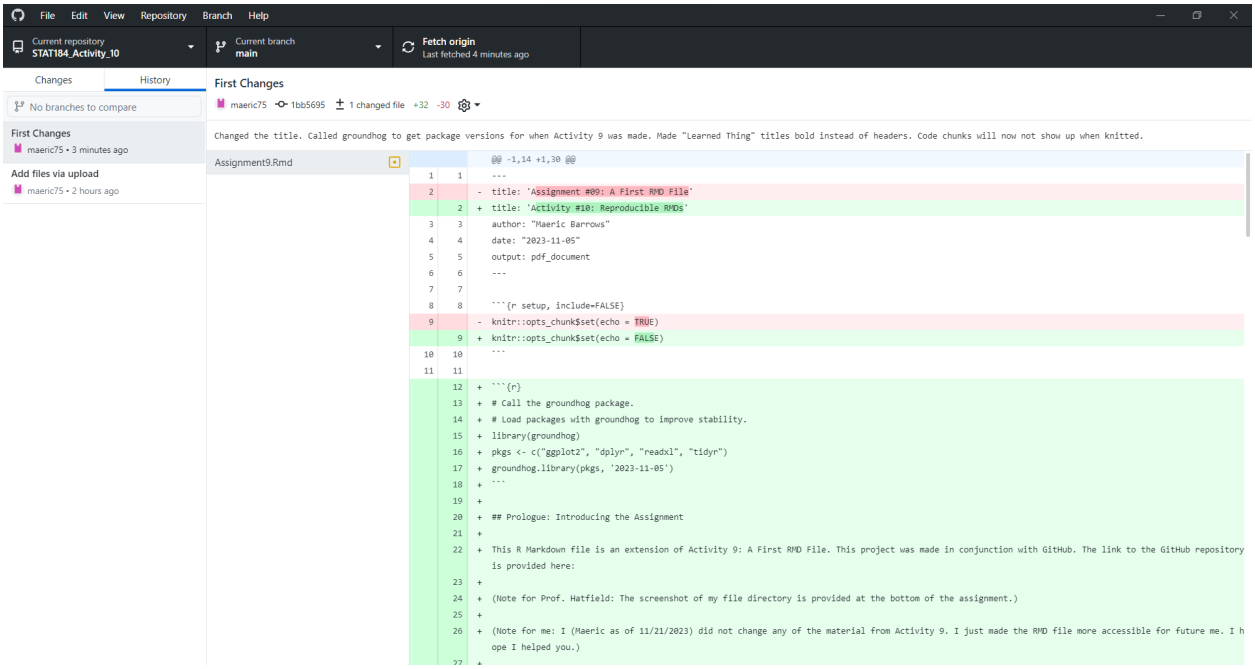



Figure 1: Screenshot of GitHub desktop

## Code Appendix

```
knitr::opts_chunk$set(echo = FALSE)
# Call the groundhog package.
library(groundhog)
# Load packages with groundhog to improve stability.
pkgs <- c("ggplot2", "dplyr", "readxl", "tidyr")
groundhog.library(pkgs, '2023-11-05')

# ------ Collatz Conjecture Code ------

### Calculate the Collatz conjecture and return the stopping number
calculate_collatz <- function(number, count = 0){
  if(number == 1){
    return(count)
  }else if(number %% 2 == 0){
    new_number <- number / 2
    calculate_collatz(number = new_number, count = count + 1)
  }else{
    new_number <- 3*number + 1
    calculate_collatz(number = new_number, count = count + 1)
  }
}

### Collects the stopping numbers when calculate_collatz is run using numbers 1 to 10,000.
sapply_collatz <- sapply(
  X = seq(1,10000,1),
  FUN = calculate_collatz
)

### Creates a data frame to store the values of sapply_collatz.
collatz_dataframe <- as.data.frame(
  x = sapply_collatz
)

### Creates a histogram using collatz_dataframe.
ggplot(
  data = collatz_dataframe,
  mapping = aes(x = `sapply_collatz`)
) +
  geom_histogram(
    bins = 50,
    fill = "cyan",
    color = "black"
  ) +
  labs(
    x = "Stopping Numbers",
    y = "Count",
    title = "Histogram of Stopping Numbers",
    subtitle = "Collatz Conjecture Input from 1 to 10,000"
  ) +
  theme_linedraw() +
  theme(
```

```
    plot.title = element_text(size = 16L,
                              face = "bold",
                              hjust = 0.5),
    plot.subtitle = element_text(size = 12L,
                                 hjust = 0.5),
    axis.title.x = element_text(size = 12L),
    axis.title.y = element_text(size = 12L)
  )

# ------ Price of Diamonds Code ------

### Creates a scatter plot to compare carat and price of a diamond.
ggplot(
  data = diamonds,
  mapping = aes(x = carat,
                y = price)
) +
  geom_point(shape = "circle",
             size = 1.5,
             color = "blue") +
  labs(
    x = "Carat",
    y = "Price",
    title = "Carat vs. Price of a Diamond"
  ) +
  theme_linedraw() +
  theme(
    plot.title = element_text(size = 16L,
                              face = "bold",
                              hjust = 0.5),
    axis.title.x = element_text(size = 12L),
    axis.title.y = element_text(size = 12L)
    )

### Creates a graph that groups all diamonds by clarity, and compares carat and price.
ggplot(
  data = diamonds,
  mapping = aes(x = carat,
                y = price)
) +
  geom_point(shape = "circle",
             size = 1.5,
             color = "blue") +
  labs(
    x = "Carat",
    y = "Price",
    title = "Carat vs. Price as Separated by Clarity"
  ) +
  theme_linedraw() +
  theme(
    plot.title = element_text(size = 16L,
                              face = "bold",
                              hjust = 0.5),
```

```
    axis.title.x = element_text(size = 12L),
    axis.title.y = element_text(size = 12L)
    ) +
  facet_wrap(vars(clarity))

### Create a summary table using cut and price.
# Load data
data(diamonds)

# Summarize data
cut_price_summary <- diamonds %>%
  group_by(cut) %>%
  select(cut, price) %>%
  summarize(
    across(
      .cols = where(is.numeric),
      .fns = list(
        min = ~min(price, na.rm = TRUE),
        Q1 = ~quantile(price, probs = 0.25, na.rm = TRUE),
        median = ~median(price, na.rm = TRUE),
        Q3 = ~quantile(price, probs = 0.75, na.rm = TRUE),
        max = ~max(price, na.rm = TRUE),
        mean = ~mean(price, na.rm = TRUE),
        stdev = ~sd(price, na.rm = TRUE)
      )
    ),
    count = n()
  )

# Convert to table
cut_price_table <- as.data.frame(cut_price_summary)

cut_price_table

# ------ Wrangling Army Data Code ------

### Wrangles the army data from previous assignments.
## Getting Stuff
# Set working directory
setwd("C:/Users/Maeric/OneDrive/Documents")

# Read in data from Excel file using a relative file path
army_data <- read_xlsx(
  path = "~/RMarkdown Files/Army_MaritalStatus.xlsx",
  range = "Sheet1!B8:Q37"
)

## Wrangling
# Removing unnecessary rows and columns
army_data <- army_data %>%
  select(c(1:3, 5,6, 8,9, 11,12))%>%
  slice(c(1:10, 12:21, 23:27))
View(army_data)
```

```r
# Creating the group case data frame
wrangled_army_data <- army_data %>%
  rename(c("Pay Grade" = 1,
           "Single Without Children_Male" = 2,
           "Single Without Children_Female" = 3,
           "Single With Children_Male" = 4,
           "Single With Children_Female" = 5,
           "Joint Service Marriage_Male" = 6,
           "Joint Service Marriage_Female" = 7,
           "Civilian Marriage_Male" = 8,
           "Civilian Marriage_Female" = 9)) %>%
  slice(-1) %>%
  pivot_longer(
    cols = !"Pay Grade",
    names_to = "marital_status",
    values_to = "Count"
  ) %>%
  tidyr::separate_wider_delim(
    cols = marital_status,
    delim = "_",
    names = c("Marital Status", "Sex")
  ) %>%
  mutate(
    Count = as.numeric(Count)
  )

wrangled_army_data
```