# Activity 09

Collin M. Lu

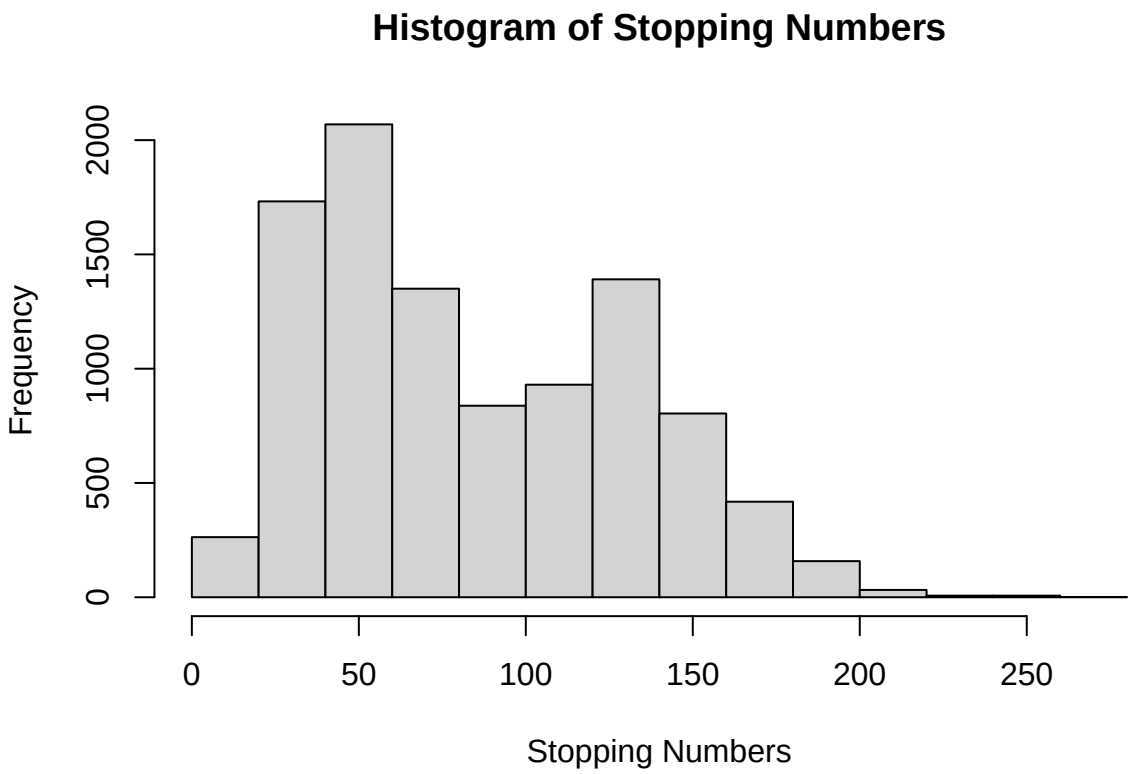
2023-11-30

## The Collatz Conjecture

The Collatz Conjecture is a famous mathematical sequence that contains two relatively simple equations, aimed to transform any positive integer into the number 1. The operation states if the number is even, divide by two. Then if the number is odd, triple it and add one. What we learned from Activity #3 and the code chuck below is that although the histogram is skewed left, larger integers don't necessarily have a larger stopping time.

```
get_collatz_conjecture <- function(start_int, stop_num = 0){
  if (start_int == 1) {
    return(stop_num)
  } else if (start_int %% 2 == 0) {
    get_collatz_conjecture(start_int = start_int / 2, stop_num = stop_num + 1)
  } else {
    get_collatz_conjecture(start_int = 3 * start_int + 1, stop_num = stop_num + 1)
  }
}
```

```
collatz_conjecture_vec <- Vectorize(
  FUN = get_collatz_conjecture,
  vectorize.args = "start_int"
)
hist(x = collatz_conjecture_vec(start_int = 1:10000),
     main = "Histogram of Stopping Numbers",
     xlab = "Stopping Numbers")
```



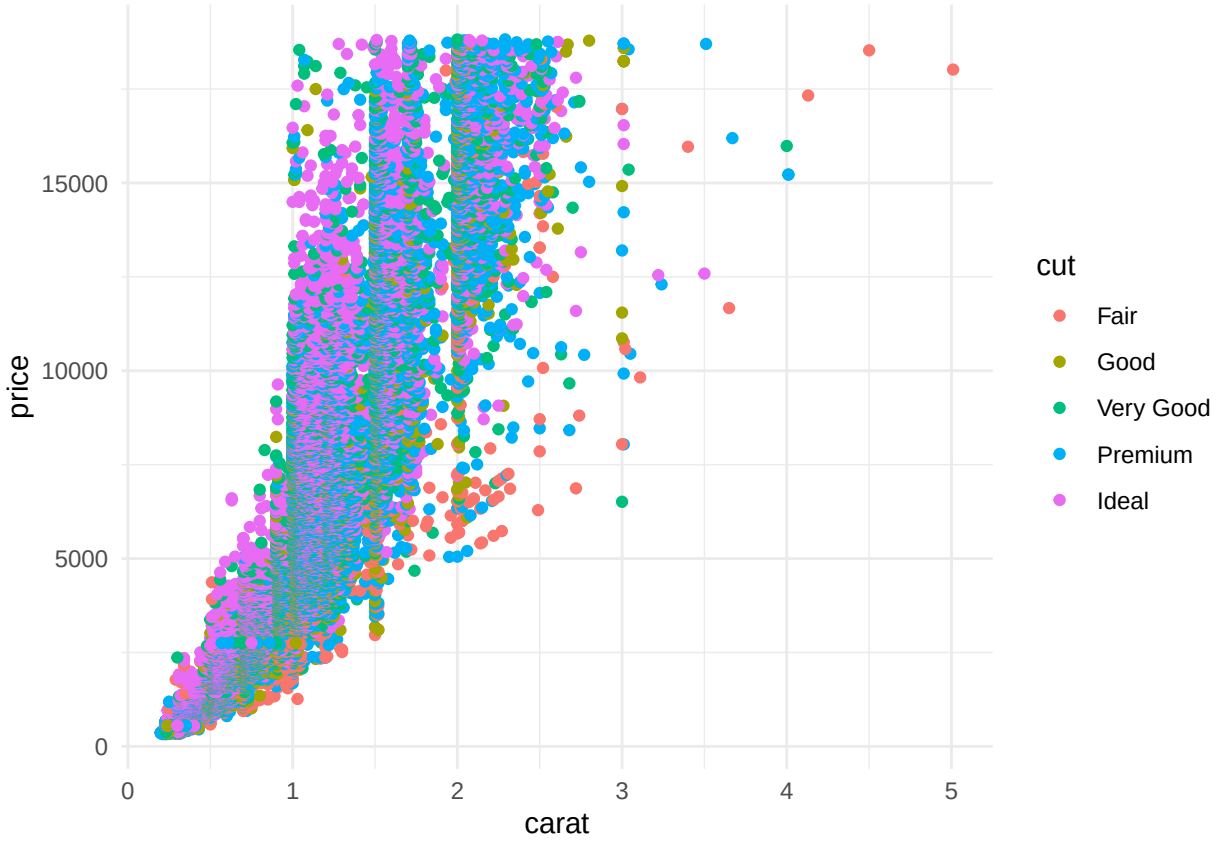

### Price of Diamonds

The price of diamonds can be seen as a positive correlation when considering the carat. In other words, as the carat increases, the prices of the diamonds tend to increase as well. When looking at the visualization below, the different types of cuts of diamonds can be seen as each color. The major takeaway from this scatter plot is that prices of diamonds increase when the carat is high, no matter the cut.

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## -- 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 errors
```

```
ggplot(diamonds) +
  aes(x = carat, y = price, colour = cut) +
  geom_point(shape = "circle", size = 1.5) +
  scale_color_hue(direction = 1) +
  theme_minimal()
```




From looking at the table below, we can tell that the centers (median and mean) of each type of cut are relatively similar. As the cut becomes better quality, the length's tend to decrease. The standard deviation also remains somewhat constant across all cuts, meaning outliers aren't playing a huge factor in calculating these statistics. Lastly, the count attribute shows that there are significantly more higher quality cut diamonds than lower quality cut diamonds in the sample.

```
library(knitr)


diamonds_table <- diamonds %>%
  group_by(cut) %>%
  select(cut, z) %>%
  summarize(
    across(
      .cols = where(is.numeric),
      .fns = list(
        min = ~min(.x, na.rm = TRUE),
        Q1 = ~quantile(.x, probs = 0.20, na.rm = TRUE),
        Q2 = ~quantile(.x, probs = 0.40, na.rm = TRUE),
        median = ~median(.x, na.rm = TRUE),
        Q3 = ~quantile(.x, probs = 0.60, na.rm = TRUE),
        Q4 = ~quantile(.x, probs = 0.80, na.rm = TRUE),
        max = ~max(.x, na.rm = TRUE),
        sam = ~mean(.x, na.rm = TRUE),
        sasd = ~sd(.x, na.rm = TRUE)
      )
```

```
    ),
    count = n()
  )

kable(diamonds_table)
```

| cut | z_min | z_Q1 | z_Q2 | z_median | z_Q3 | z_Q4 | z_max | z_sam | z_sasd | count |
|---|---|---|---|---|---|---|---|---|---|---|
| Fair | 0 | 3.47 | 3.836 | 3.97 | 4.07 | 4.49 | 6.98 | 3.982770 | 0.6516384 | 1610 |
| Good | 0 | 2.99 | 3.480 | 3.70 | 3.87 | 4.08 | 5.79 | 3.639507 | 0.6548925 | 4906 |
| Very Good | 0 | 2.83 | 3.320 | 3.56 | 3.82 | 4.10 | 31.80 | 3.559801 | 0.7302281 | 12082 |
| Premium | 0 | 2.85 | 3.440 | 3.72 | 3.94 | 4.26 | 8.06 | 3.647124 | 0.7311610 | 13791 |
| Ideal | 0 | 2.75 | 3.060 | 3.23 | 3.52 | 4.05 | 6.03 | 3.401448 | 0.6576481 | 21551 |

## What I've Learned

So far this semester, I've learned lots about R and coding overall. Prior to taking this course, my knowledge about creating visualizations and tables by coding was slim to none. One of major the things I've learned is the importance of functions and naming functions to create effective lines of code. Naming functions is crucial to create organized, as well as effective, code. Another major concept that I've learned about is data wrangling. Data wrangling is a huge part of the statistics industry, so I'm glad to have been exposed to it before I begin taking higher level statistics and coding courses.