A First RMD File

Milo Schmitt

2023-11-07

The Collatz Conjecture

The Collatz Conjecture presents two arithmetic operations and a question: if these two operators are applied, will we be able to transform any positive integer into the number 1?

Consider the following piecewise function:

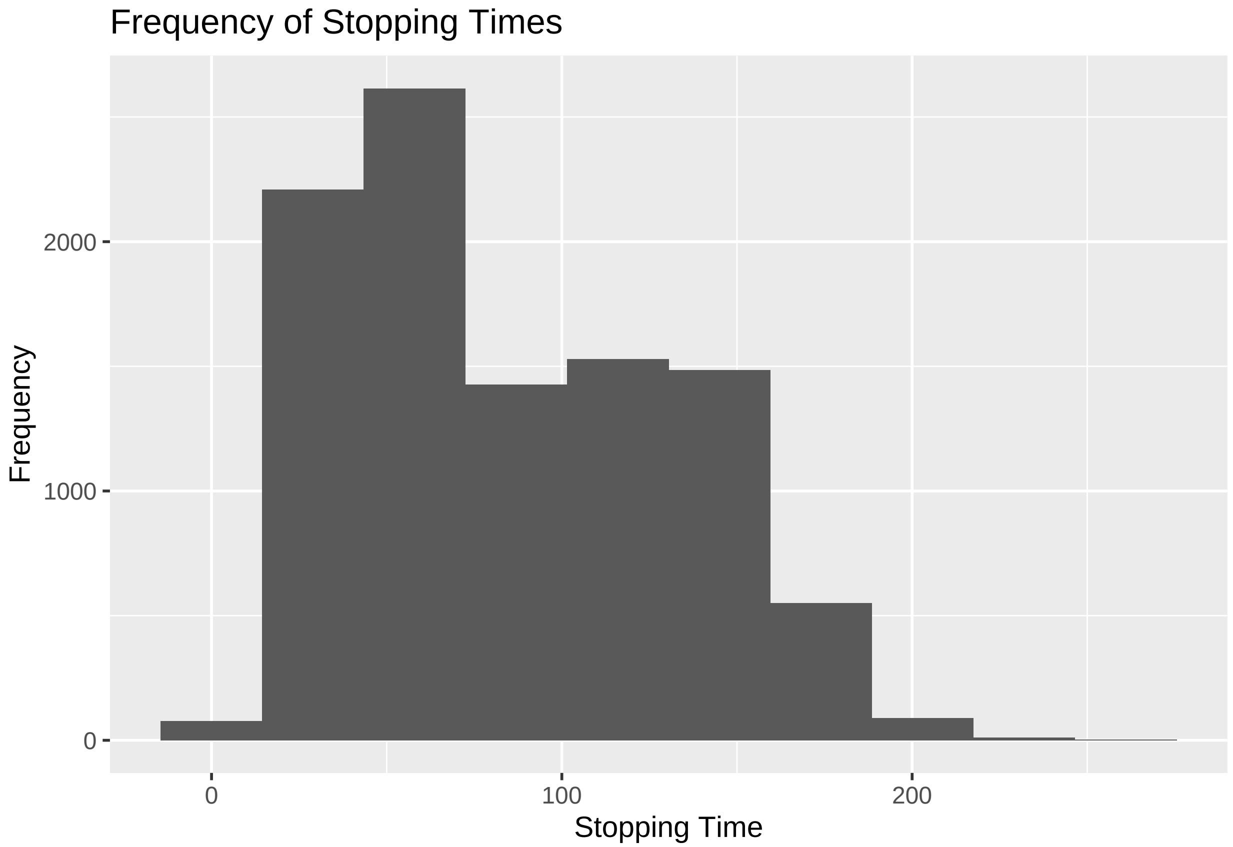
```
\begin{equation}
 f(n) =
   \left\{\begin{array}{lr}
       f(n/2) & if\ n\ is\ even \\
       f(3n+2) & if\ n\ is\ odd \\
       stop & if\ n\ is\ one
    \end{array}\right.
 \end{equation}
```

(The above was created in LateX, and I did not realize that it can only be used with an HTML document)

f(n) = { f(n/2) if n is even f(3n+1) if n is odd stop if n is one }

The number of times a positive integer n must be passed through this function to transform into one is known as the stopping time. For example, the stopping time of 1 is 0, 2 is 1, and 3 is 7.

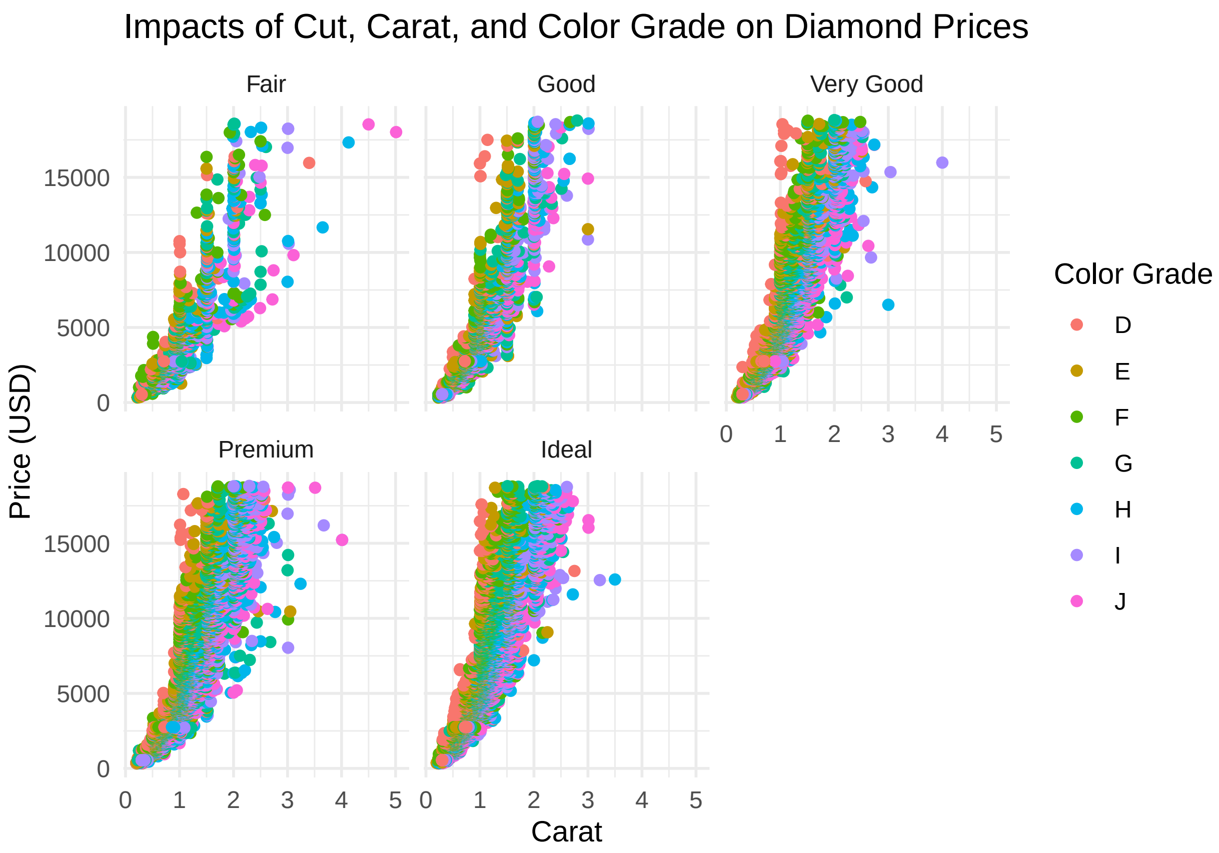
The goal presented was to find a visualization to represent the first stopping numbers of the first 10,000 positive integers.



The above histogram displays one bar for a certain range of stopping times. For each of these ranges, the y axis represents the frequency at which the stopping time appears. It should be noted that a majority of the first 10,000 positive integers have stopping times between 0 and 100.

Diamonds

The Diamonds dataset provided by R is a large dataset representing a group of diamonds. Each of these diamonds has many attributes: cut, color, price, length, width, height, and more. The goal of this activity is to create a visualization which displays a relationship between a certain diamond characteristic and the price of diamonds.



The above visualizations are an overview of the impacts of Cut, Carat, and Color Grade on the price of diamonds. It is clear from the graphs that Color Grade does not have much of an impact on price. Carat and Cut, however, seem to have some positive correlation with price.

```
## # A tibble: 5 x 28
##   cut       x_count x_min  x_Q1 x_median  x_Q3 x_max x_mad x_mean  x_sd y_count
##   <ord>       <int> <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>   <int>
## 1 Fair         1610     0  5.63     6.18  6.7  10.7  0.808   6.25 0.964    1610
## 2 Good         4906     0  5.02     5.98  6.42  9.44 1.10    5.84 1.06     4906
## 3 Very Good   12082     0  4.75     5.74  6.47 10.0  1.25    5.74 1.10    12082
## 4 Premium     13791     0  4.8      6.11  6.8  10.1  1.42    5.97 1.19    13791
## 5 Ideal       21551     0  4.54     5.25  6.44  9.65 1.19    5.51 1.06    21551
## # i 17 more variables: y_min <dbl>, y_Q1 <dbl>, y_median <dbl>, y_Q3 <dbl>,
## #   y_max <dbl>, y_mad <dbl>, y_mean <dbl>, y_sd <dbl>, z_count <int>,
## #   z_min <dbl>, z_Q1 <dbl>, z_median <dbl>, z_Q3 <dbl>, z_max <dbl>,
## #   z_mad <dbl>, z_mean <dbl>, z_sd <dbl>
```

The above table is a summary of various statistics derived from the x, y, and z (length, width, height) values from the diamonds data set.

The above table displays frequencies of various attributes within the diamond table. It can be noted that a significant portion of the diamonds are of the premium or ideal quality.

Table 1: Cut and Color of Diamonds

Color/Cut	Fair	Good	Very Good	Premium	
D	163 (0.30%)	662 (1.23%)	1,513 (2.80%)	1,603 (2.97%)	2,83
E	224 (0.42%)	933 (1.73%)	2,400 (4.45%)	2,337 (4.33%)	3,90
F	312 (0.58%)	909 (1.69%)	2,164 (4.01%)	2,331 (4.32%)	3,82
G	314 (0.58%)	871 (1.61%)	2,299 (4.26%)	2,924 (5.42%)	4,88
H	303 (0.56%)	702 (1.30%)	1,824 (3.38%)	2,360 (4.38%)	3,11
I	175 (0.32%)	522 (0.97%)	1,204 (2.23%)	1,428 (2.65%)	2,09
J	119 (0.22%)	307 (0.57%)	678 (1.26%)	808 (1.50%)	89
Total	1,610 (2.98%)	4,906 (9.10%)	12,082 (22.40%)	13,791 (25.57%)	21,55

Reflections

The most significant lesson I've learned throughout this course is the ability to plan out and describe my code. Being able to present material in a meaningful and comprehensible manner is very important in all aspects of life, not just data science. I feel that I've also learned how to manipulate and visualize data frames, which is a useful skill to have.

One of my concerns is that data recovered from real-world activities is not always presented in a clean or enjoyable manner. While I have learned various data wrangling skills to account for this, I think I still need to polish these skills for the future.

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE)

#Load packages with groundhog to improve stability
library("groundhog")
pkgs <- c("ggplot2", "dplyr", "janitor", "kableExtra", "knitr")
groundhog.library(pkgs, '2023-11-07') #Use the date that you started the project here


# The following code creates a function to find the stopping number of any input with regard to the Col

updateEven <- function(number) # create function for evens
{
return(number/2)
}

updateOdd <- function(number) # create function for odds
{
return(number*3 + 1)
}

countCollatz <- function(number) # create final function
{
count = 0
result = 0
while(number != 1) # stop if number is one
{
if(number%%2 == 0)
{
result <- updateEven(number) # even function if # is even
count <- count + 1
}

else
{
result <- updateOdd(number) # odd function if not even
count <- count + 1
}
number = result
}
return(count)
}

inputs <- seq(1, 10000, 1) # input numbers from 1 to 10000
countCollatzVectorized <- Vectorize(countCollatz)
stopping_times <- countCollatzVectorized(inputs)
stopping_times <- as.data.frame(stopping_times) # turn vector into dataframe

ggplot(stopping_times, aes(x = stopping_times)) +
  geom_histogram(bins=10) + # 10 bins to prevent overcrowding
  labs( # set labels
    title = "Frequency of Stopping Times",
    x = "Stopping Time",
    y = "Frequency"
  )

# The following code creates a plot which utilizes cut, color, and carat data to visualize changes in d

ggplot(diamonds) + aes(x = carat, y = price, colour = color) + # x carat, y price, color by color grade

geom_point(shape = "circle", size = 1.5) + # scatter plot

labs( # more detailed labels

x = "Carat",

y = "Price (USD)",

color = "Color Grade",

title = "Impacts of Cut, Carat, and Color Grade on Diamond Prices"

) +

scale_color_hue(direction = 1) +

theme_minimal() +

facet_wrap(vars(cut)) # five graphs for cut

# The following code utilizes the across function to find various statistics for each dimension of the

diamonds %>%
  group_by(cut) %>%
  summarise(
    across(c(x, y, z), list(
               count = ~sum(!is.na(.)),
               min = ~min(., na.rm = TRUE),
               Q1 = ~quantile(., probs = 0.25, na.rm = TRUE),
               median = ~median(., na.rm = TRUE),
               Q3 = ~quantile(., probs = 0.75, na.rm = TRUE),
               max = ~max(., na.rm = TRUE),
               mad = ~mad(., na.rm = TRUE),
               mean = ~mean(., na.rm = TRUE),
               sd = ~sd(., na.rm = TRUE)
  )))

# The following code creates a frequency table using the color and cut statistics of the diamonds data.

diamondTable <- diamonds %>%
  tabyl(color, cut) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_percentages(denominator = "all") %>%
  adorn_pct_formatting(digits = 2) %>%
  adorn_title(
    placement = "combined",
    row_name = "Color",
    col_name = "Cut")

formatNs <- attr(diamondTable, "core") %>%
  adorn_totals(where = c("row", "col")) %>%
  mutate(
    across(where(is.numeric), format, big.mark = ",")
  )

diamondFreqTab <- diamondTable %>%
  adorn_ns(position = "front", ns = formatNs)

diamondFreqTab %>%
  kable(
    caption = "Cut and Color of Diamonds",
    booktabs = TRUE,
    align = c('l', rep("c", 6))
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 16
  )
```