Phase 1 and 2 Analysis Including Collatz Conjecture and Diamonds Data

Katie Clayton

2022-11-04

Collatz Conjecture

Intro to Collatz Conjecture The collatz conjecture is used to transform every positive integer into one. Two arithmetic equations are input using if statements and else if statements in order to get to the integer one. If the number is even, the function will use the number and divide it by two. If the number is odd, it will be multiplied by three and added to one. Once the number gets to one, the function will stop. The purpose is to find out the most frequent stopping time based off this function.

```
#name function and add nouns
getCollatz <- function(number,count=0)
  #first part of collatz conjecture, when number is 1
{if (number == 1){
  return (count)}
  #when number is even, divide by two
  else if (number%%2==0)
    {getCollatz(number/2,count+1)}
  #when number is odd, multiply by 3 and add 1
  else{
    getCollatz(number*3+1,count+1)}}
# create function for histogram
stoppingNumbers <- sapply(
# label x axis and incldue function that needs to be called
X = 1:10000,
FUN = getCollatz
)
# make the histogram
hist(stoppingNumbers)
```

Figure 1. Histogram of The Most Frequent Stopping Numbers For The Collatz Conjecture The Collatz Conjecture distribution is focused from numbers 0 to 250. As shown, there are modal clumps from around 25 to 75 and also 100 to roughly 155. This proves that the distribution was highly impacted at these certain numbers. According to the histogram, the most frequent stopping number is 50. This means that the function, on average, runs through the arithmetic operations 50 times before outputting the number one. A viewer may also gather other numbers that are most frequent when running through the Collatz Conjecture from this histogram.

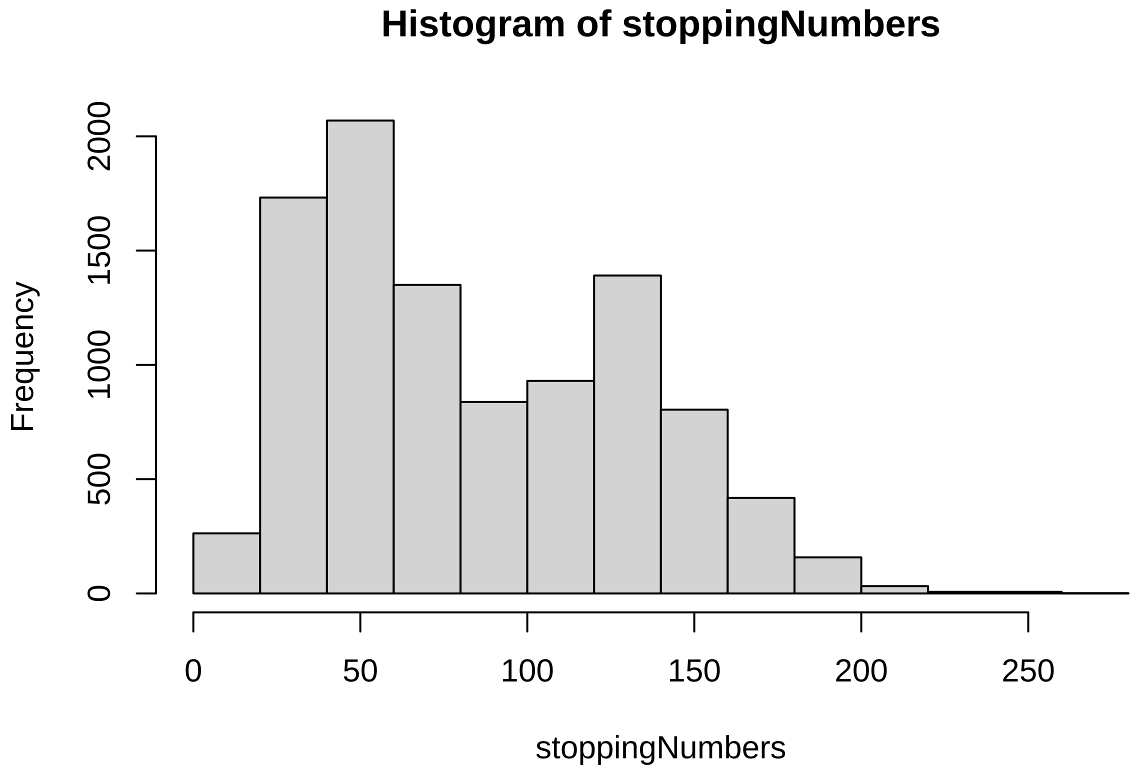

Figure 1: Collatz Conjecture

Diamonds Data Visualizations

Scatter Plots A viewer can gain the most knowledge from the diamonds data through a data visualization. A scatter plot is most appropriate for this set of data because it can include each observation and also a line of best fit to show the linear regression as the weight of the diamond increases.

```
# filter data so there is no more than four groupings on one graph
library(ggplot2)
data(diamonds)
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
```

```
filtered_diamonds <- diamonds %>%
  # pull a sample from the data
  slice_sample(prop = 0.1) %>%
  filter(color %in% c("D", "F", "G", "H"))

  # create graph using ggplot and assign variables
  ggplot(
    data = filtered_diamonds,
    mapping = aes(x = carat, y = price, colour = color)
  ) +

  # establish size for the line of best fit and the points, as well as shape
  # for points
  geom_point(shape = "circle", size = 1.5) +
  geom_smooth(span = 0.75, se = FALSE) +

  # give each color a color which will display the legend
  scale_color_manual(
    values = c(D = "#0D0B0C",
               F = "#888386",
               G = "#00A7FF",
               H = "#FF8E00",
               J = "#000000",
               E = "#8A8591",
               I = "#DC9423")
  ) +

  # label the axis, add a title
  labs(
    x = "Weight of Diamond (carats)",
    y = "Price of Diamond (dollars)",
    title = "Price of Diamond (dollars) vs. Color and Weight (carats)"
  )+
  theme_minimal()
```

```
## ‘geom_smooth()‘ using method = ’gam’ and formula ’y ~ s(x, bs = "cs")’
```

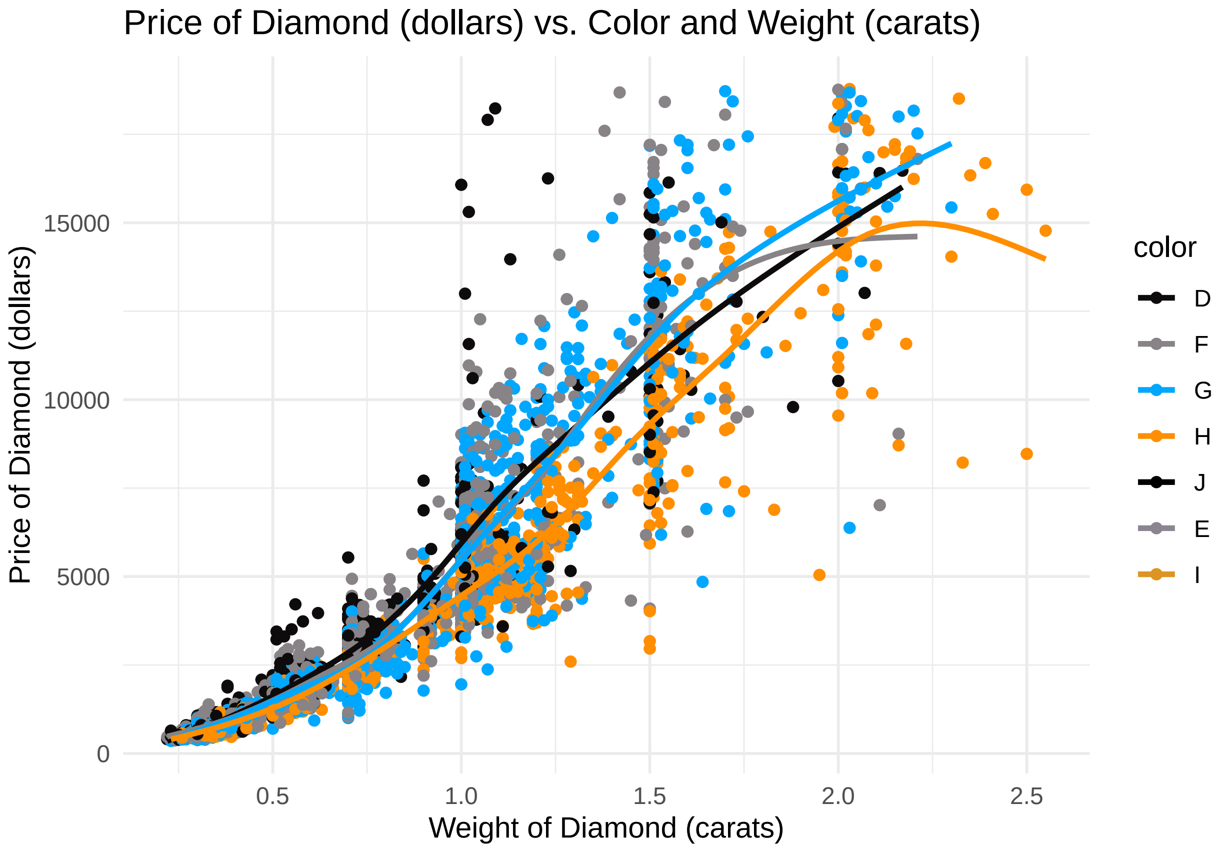

Figure 2: Diamonds Data Scatterplot 1

Figure 2. Price of Diamond vs Weight of Diamond The scatter plot above shows the comparison between price of diamond and weight of diamond. A reader is able to compare the diamond colors, “E”, “J”, and “I” in this visualization. The color “E” increases in price the fastest as the weight increases compared to the other two colors; However, all of the colors have a steep increase in price initially. The main difference is shown where the lines of best fit begin to drop off or level out after hitting their peak point. Although, the color “E” increases the fastest, it also decreases after hitting its peak the fastest, unlike the color “J” which continues to increase in price as the weight increases.

```
# filter data so there is no more than four groupings on one graph
library(ggplot2)
data(diamonds)
library(dplyr)
filtered_diamonds <- diamonds %>%
  # pull a sample from the data
  slice_sample(prop = 0.1) %>%
  filter(color %in% c("J", "E", "I"))

  # create graph using ggplot and assign variables
  ggplot(
    data = filtered_diamonds,
    mapping = aes(x = carat, y = price, colour = color)
  ) +

  # establish size for the line of best fit and the points, as well as shape
  # for points
  geom_point(shape = "circle", size = 1.5) +
  geom_smooth(span = 0.75, se = FALSE) +

  # give each color a color which will display the legend
  scale_color_manual(
    values = c(D = "#0D0B0C",
               F = "#888386",
               G = "#00A7FF",
               H = "#FF8E00",
               J = "#000000",
               E = "#8A8591",
               I = "#DC9423")
  ) +

  # label the axis, add a title
  labs(
    x = "Weight of Diamond (carats)",
    y = "Price of Diamond (dollars)",
    title = "Price of Diamond (dollars) vs. Color and Weight (carats)"
  )+
  theme_minimal()
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```

Figure 3. Price of Diamond vs Weight of Diamond The scatter plot above shows the differences in price increase compared to weight for the diamond colors “D”, “F”, “G”, “H”. The specific colors are labeled along the side of the graph while the axes are labeled pertaining to the price and weight. The lines of best fit show that these four specific diamond colors all increase around the same at first, but “D” is the one with the highest peak and “H” continues to have a price until the end of the graph unlike the others.

Summary Table Summary tables are used to organize data in a way that displays numerical values that can be easily gathered and compared by viewers for their personal use.

```
## load necessary packages
library(ggplot2)
library(dplyr)
library(knitr)
library(kableExtra)
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : ’length(x) = 3 > 1’ in coercion to ’logical(1)’
```

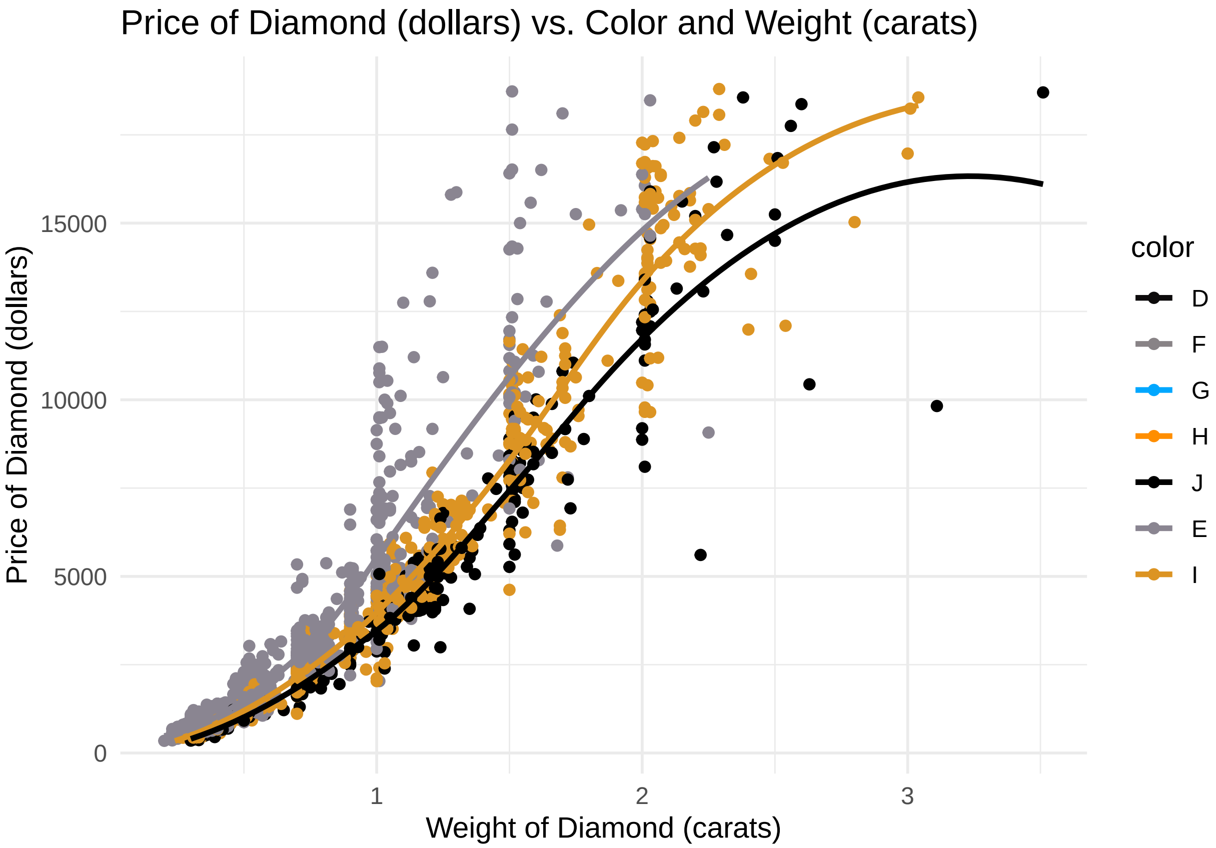

Figure 3: Diamonds Data Scatterplot 2

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
# load diamonds data
data(diamonds)
# name variable
depthDiamonds <- diamonds %>%
  # choose cut and the depth which is z
  group_by(cut) %>%
  select(cut, price) %>%
  # input the 10 statistics needed, not forgetting na.rm = TRUE at the end of the line
  summarize(
    across(
      .cols = where(is.numeric),
      .fns = list(
        min = ~min(price, na.rm = TRUE),
        Q1 = ~quantile(price, probs = 0.20, na.rm = TRUE),
        Q2 = ~quantile(price, probs = 0.40, na.rm = TRUE),
        median = ~median(price, na.rm = TRUE),
        Q3 = ~quantile(price, probs = 0.60, na.rm = TRUE),
        Q4 = ~quantile(price, probs = 0.80, na.rm = TRUE),
        max = ~max(price, na.rm = TRUE),
        sam = ~mean(price, na.rm = TRUE),
        sasd = ~sd(price, na.rm = TRUE)
      ),
      round(digits = 2)
    ),
    # format the numbers so the big ones have a , if needed
    count = format(n(), big.mark = ","),
  )
# give the columns names
colnames(depthDiamonds) <- c("Cut", "Min", "1st Quintile", "2nd Quintile", "Median", "3rd Quintile", "4
# use piping and kable function to output the table
depthDiamonds %>%
  kable(
    # add title, grid lines and align the numbers with words
    caption = "Statistics of the Price vs Cut of a Diamond",
    booktabs = TRUE,
    align = c("l", rep("c", 6))
```

Table 1: Statistics of the Price vs Cut of a Diamond

Cut	Min	1st Quintile	2nd Quintile	Median	3rd Quintile	4th Quintile
Fair	337	1790.6	2805.0	3282.0	3947.4	6090.4
Good	327	876.0	2176.0	3050.5	3888.0	5834.0
Very Good	336	760.0	1892.4	2648.0	3751.6	6288.0
Premium	326	924.0	2100.0	3185.0	4408.0	7485.0
Ideal	326	803.0	1243.0	1810.0	2529.0	5613.0

```
) %>%
kableExtra::kable_styling(
  bootstrap_options = c("striped", "condensed"),
  font_size = 16
) %>%
kableExtra::kable_classic()
```

Figure 4. Diamonds Data Summary Table The summary table of the diamonds data includes each cut of diamond and 10 different statistics pertaining to the price of the specific cut. A viewer can gain a sense of how many diamonds there are per cut, which type of cut has the highest maximum and much more. This summary table would be most useful when trying to gain specific statistics about the price of a certain cut of a diamond.

Knowledge from STAT 184

Stat 184 has taught me a lot of different things so far. Coming into this class, I had very little prior coding experience so I was very nervous. Although, as the class has continued, I have become more comfortable with reading code and typing it. I have also really enjoyed using the different packages within R, especially ggplot2. I used the esquisse package and ggplot2 in order to create the visualizations for the diamonds and penguins data. I have also gained other knowledge about data visualizations from Kosslyn and Tufte. They specified things to avoid, like ducks, as well as things to include that will help make the visualization most pleasing to the eye. It has been very helpful to have all the slides and other material posted in order to look back on when getting confused. I have learned the importance of using tidy data and also how to wrangle the data so it is easier to work with. One example that we did in class was tidying the family data and the army data. The lesson that I remember most and will continue to apply is the idea of PCIP. By following those steps, it has led me to solutions quicker. Starting with the first assignment of planning, I have continued to use PCIP for every assignment. Before taking this class, I had zero experience in statistical coding and now I feel as if I have increased my knowledge on this topic.