Activity 10

Otis Murray

2023-11-03

Section 1: Collatz Conjecture

This is an R Markdown document that goes over the Collatz Conjecture, the mathematical formulation associated of the function, and answer a question asked by Dr. Neil Hatfield.

The Collatz Conjecture is a mathematical phenomenon that asks whether repeating two arithmetic operations will eventually transform every positive integer into the value 1, it is defined by a piecewise function f(n). The mathematical formulation associated with this problem is as follows: if your n value is 1 then you stop, else if your n value is even divide n by 2, if your n value is odd you need to multiply n by 3 and then add 1. Eventually, through enough iterations, you will reach the number 1. Dr. Neil Hatfield's question was about the distribution of stopping times for the first 10,000 positive integers. "Stopping times" is defined as the number of times the function is recursively invoked until the value 1 is reached. Below is a histogram of the distribution of the first 10,000 positive integers and their stopping times.

Histogram For Collatz Conjecture

The data visualization above appears to have the majority of its distribution at lower values and it is skewed right towards higher values. The graph has very few values where the stopping time is higher than 200 but it does have some values on the both extremes. In other words, the collatz conjecture is able to turn most values into 1 in a relatively short amount of recursions, but some values take an extra number of recursions to achieve the value 1. To answer Dr. Neil's question

Section 2: Diamonds

In section 2 we examine the diamonds data set from the ggplot2 package, this data set consists of around 54,000 diamond cases. The table has 10 different attributes: carat, cut, color, clarity, depth, table, price, and x, y, z values. While all the attributes may have some influence on price, below I will be examining the carats and how they effect the price of the diamonds. I will also exploring the relationship between the x-values (length) of the diamonds and their effect on the price of said diamonds.

In the data visualization above we can see that for every cut of diamonds the price of the diamonds increases as the weight in carats increases. Therefore, the two variables have a positive association as changing one will cause the same type of change in the other. Additionally, very little separates the different cuts of diamonds except for the distributions as the fair diamonds appear to have the most extremes and looks as though it has the least amount of points on the plot.

In the data visualization above we can see the length of the diamonds and its effect on the price of the diamonds. Like the example before, the two variables are directly associated with one another, as when the length increases, the price is expected to also increase. Despite there being a few outliers, the graphs, organized by color of diamonds appear to be relatively similar to one another. In short, this visualization tells us that there is a correlation between the length of a diamond and its resulting price.

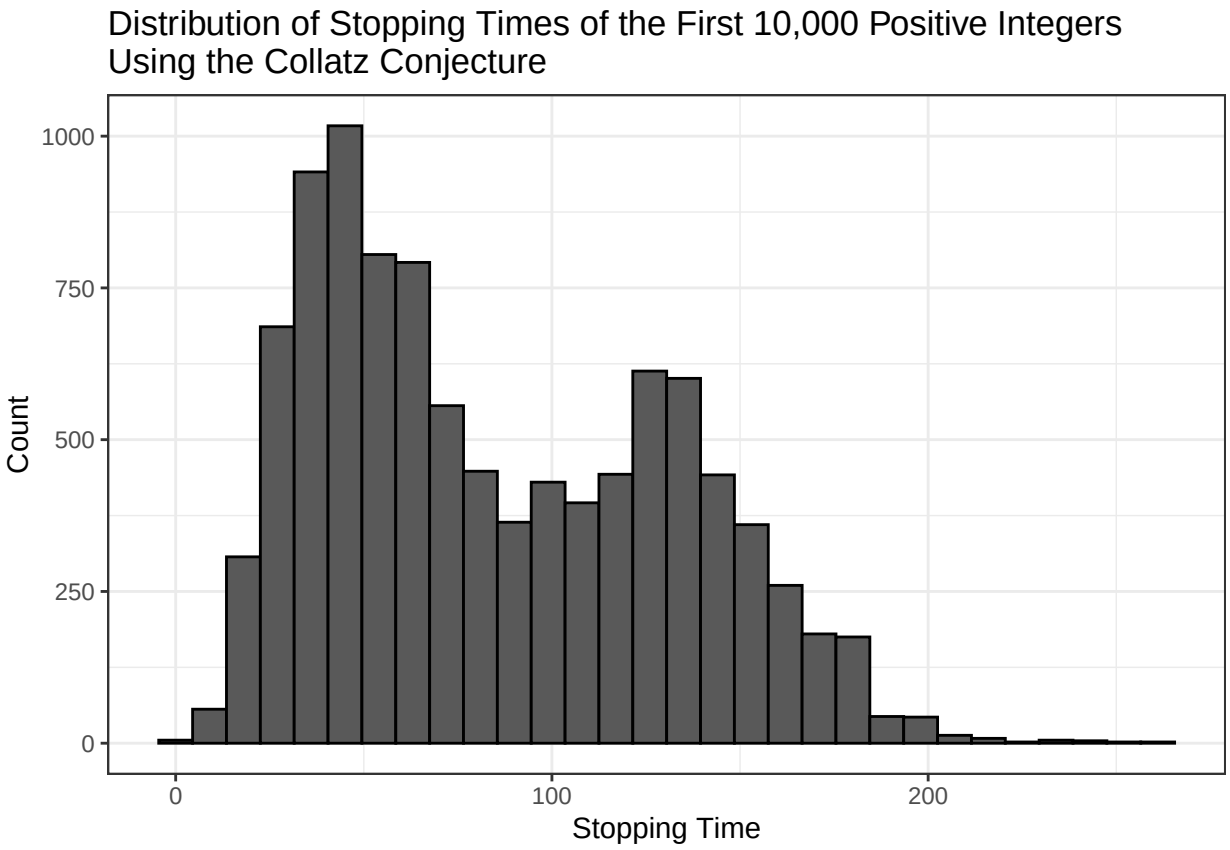


Figure 1: Histogram of Stopping Times

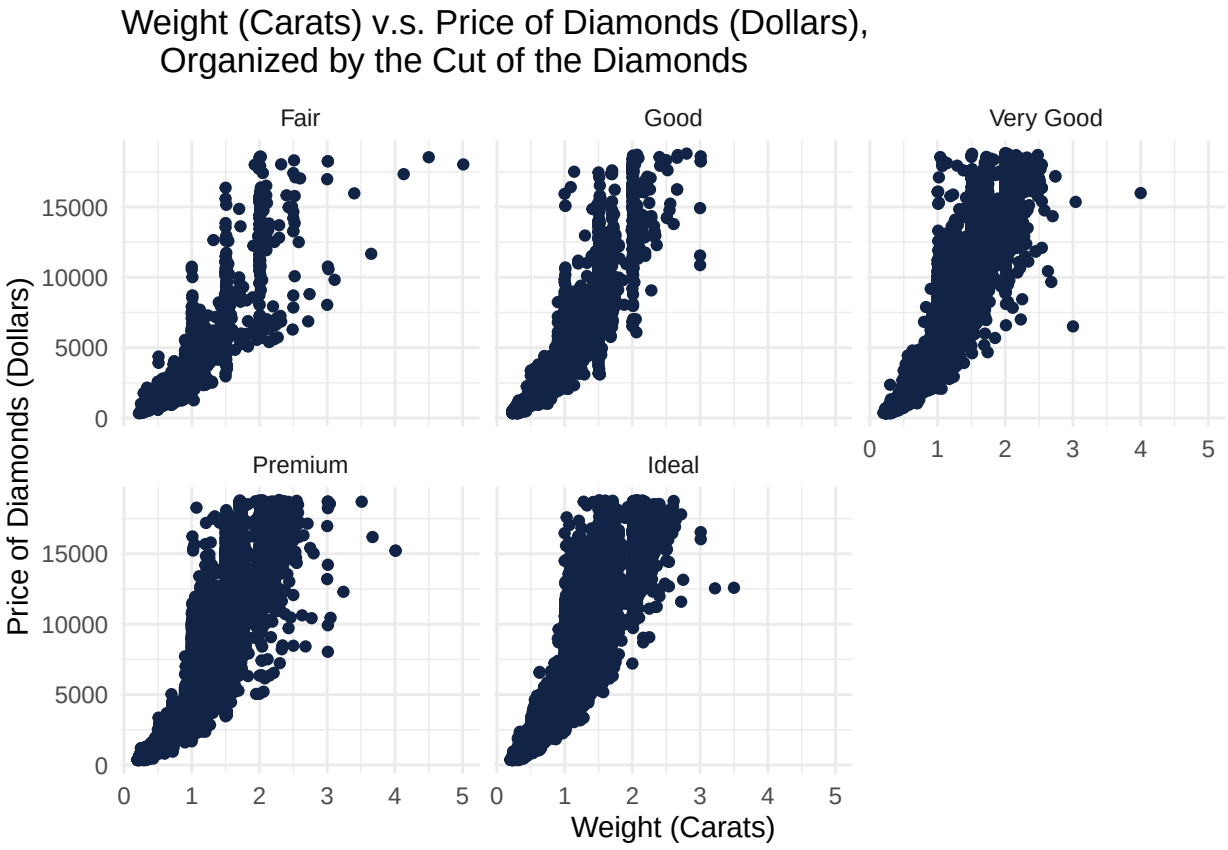


Figure 2: Weight (Carats) v.s. Price of Diamonds (Dollars), Organized by the Cut of the Diamonds

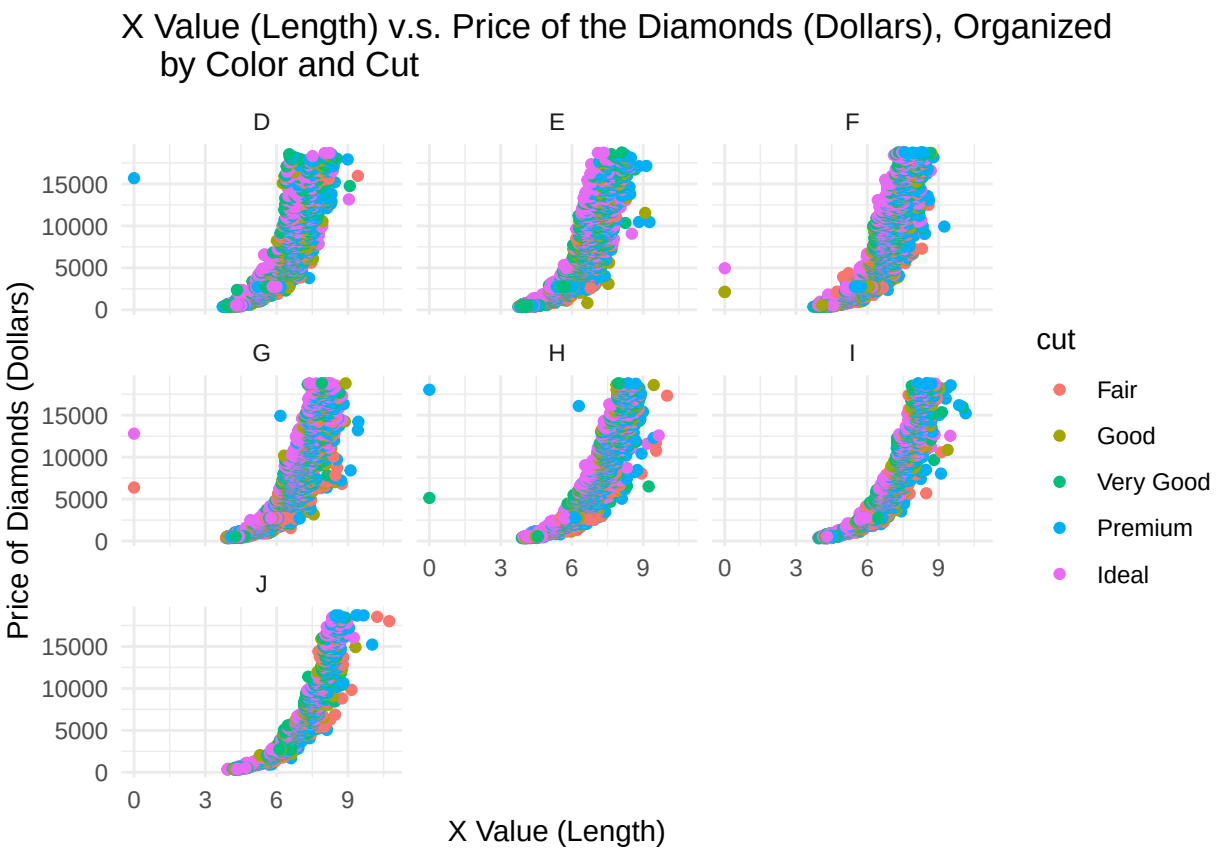


Figure 3: X Value (Length) v.s. Price of the Diamonds (Dollars), Organized by Color and Cut

Table 1: Summary Statistics for the Price of Diamonds by Cut

cut	min	Q1	median	Q3	max	sam	sasd	count
Fair	337	2050.25	3282.0	5205.50	18574	4358.758	3560.387	1610
Good	327	1145.00	3050.5	5028.00	18788	3928.864	3681.590	4906
Very Good	336	912.00	2648.0	5372.75	18818	3981.760	3935.862	12082
Premium	326	1046.00	3185.0	6296.00	18823	4584.258	4349.205	13791
Ideal	326	878.00	1810.0	4678.50	18806	3457.542	3808.401	21551

Below you can see a summary table followed by a description about how the price of diamonds is effected by the cut of the diamond.

The summary table above examines certain statistics for each of the 5 different cuts of diamonds and places them side by side to allow for a comparison between the groups. For example, using the table above we can deduce that Q1 for fair diamonds has the highest price on average while Q3 for the premium diamonds has the highest price. Additionally, we can infer that the lower tiers of diamonds are not as representative of their true population's statistics as they have much lower sample counts (1610 and 4906) when compared to the ideal cut of diamond that has a sample size of 21,551 diamonds.

Through the two data visualizations and the summary table above we can see that Price isn't heavily influenced by cut because all 5 of the different cuts of diamonds had pretty similiar values and graphs. While this could be due to the fact that the sample sizes are very different, the data doesn't allow us to conclude any strong correlations between the cut of the diamond and its price.

Section 3: Reflections

Over the course of Stat 184 I have learned a lot about how to not only program in R, but important coding guidelines that I should learned to follow no matter the language I am working with. I have learned everything from establishing a plan, to wrangling the data, to tidying that data, to presenting that data in an effective and visually pleasing way. More specifically, I have learned how important it is to create and follow a plan before jumping straight into your code. While it may be easy to convince yourself that because you know how to get started you should just jump right in, however, this may lead to you getting lost or losing the projects purpose sometime down the road. For that reason and many others, I have realized planning is very helpful to add to your coding lifestyle. As for wrangling and tidying data I have learned a lot about what kind of data is good and what kind is bad. Pretty much everything can be quantified and transformed into numbers, but the difference between the numbers and data is that data has a purpose. It's important to find data that relates to the purpose of your project. Using that data you now have to tidy it because it is very challenging to present your data when it looks fragmented. Through this idea, I have realized that audiences are very judgmental and they're looking for an opportunity to pick apart your work. It is your job as a statistician to make sure there is nothing for them to criticize. Moreover, tidying data doesn't only improve how the raw data looks, but it will also improve how the data visualizations look because those are based on the tidied data set. Tidying data also increases the reputability of the author because it is very common practice, besides, it comes off as lazy if someone just presents their raw wrangled data. Last but certainly not least, I have learned through Kosslyn and Tufte that there are many things to consider when presenting your data. From making sure your visualizations remain simple, yet effective, to not forgetting to include narrative text under your visualizations. What interested me the most in this course, however, was the process of data wrangling. I always wondered how one could just take large amounts of data and transfer it into an environment where code could be used to alter it. Below you will see some examples of the data wrangling methods I have learned using the Minneapolis2013 data set.

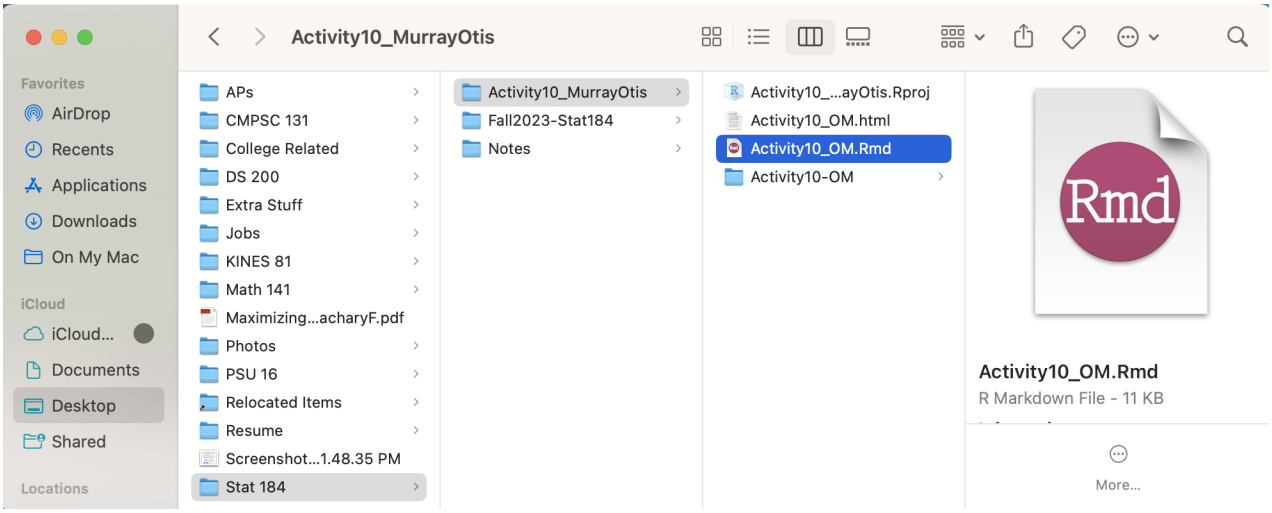
```r
#load packages
library(dplyr)
library(dcData)

#load babynames data set and wrangle specific data from the data set
data(BabyNames)
#define names you want to find
chosen_names <- c("Justin", "Eric", "Zach", "Nick", "Bob", "Andrew")
Tmp6 <- BabyNames %>%
filter(name %in% chosen_names) %>%
group_by(name, year, sex) %>%
summarise(Freq2 = n()) %>%
arrange(desc(Freq2))
```

```
## `summarise()` has grouped output by 'name', 'year'. You can override using the
## `.groups` argument.
```

```r
#view your wrangled data
View(Tmp6)
```

File Path Screenshot

```
## [1] "/Users/otismurray/Desktop/Stat 184/Activity10_MurrayOtis"
```



Code Appendix

```
knitr::opts_chunk$set(echo = FALSE, dpi=300)
#Load packages with groundhog
library("groundhog")
pkgs <- c('ggplot2', 'dplyr', 'kableExtra', 'rvest', 'readr', 'tidyr', 'googlesheets4')
groundhog.library(pkgs, '2023-11-03') #Using the date when project was started
#load diamonds data
data("diamonds")
gs4_deauth
#collatz function involving conditional statements
get_collatz <- function(n, counter=0){
  if (n==1){
    return(counter)
  }else if (n%%2==0){
    counter <<- counter+1
    get_collatz(n=n/2,counter=counter+1)
  }else(
    get_collatz(n=3*n+1,counter=counter+1)
  )
}
#create a vector of all the stopping times
collatzVector <- Vectorize(
  FUN = get_collatz,
  vectorize.args = "n")

#create a histogram of the stopping times
ggplot(
  mapping=aes(collatzVector(n=1:10000))
#Plug in values 1 to 10000
)+
  geom_histogram(
    bins=30,
    col=I("black")
  )+
  labs(
#labels of graph
    x="Stopping Time",
    y="Count",
    title="Distribution of Stopping Times of the First 10,000 Positive Integers
Using the Collatz Conjecture"
  )+
  theme_bw()
#Data visualization 1 of Diamond Data Set (Carat v.s. Price)
ggplot(diamonds) +
  aes(x = carat, y = price) +
  geom_point(shape = "circle", size = 1.5, colour = "#112446") +
  labs(
#labels of graph
    x = "Weight (Carats)",
    y = "Price of Diamonds (Dollars)",
    title = "Weight (Carats) v.s. Price of Diamonds (Dollars),
    Organized by the Cut of the Diamonds"
```

```
  ) +
  theme_minimal() +
  facet_wrap(vars(cut))
# Data Visualization 2 of the diamond data set (X value (length) v.s. price)
ggplot(diamonds) +
  aes(x = x, y = price, colour = cut) +
  geom_point(shape = "circle", size = 1.5) +
  scale_color_hue(direction = 1) +
  theme_minimal() +
  facet_wrap(vars(color)) +
  labs(
  #labels of graph
    x = "X Value (Length)",
    y = "Price of Diamonds (Dollars)",
    title = "X Value (Length) v.s. Price of the Diamonds (Dollars), Organized
    by Color and Cut"
  )
#load packages
library(knitr)
library(kableExtra)

#create basic data frame
diamondTable <- diamonds%>%
  group_by(cut) %>%
  select(cut,price) %>%
  summarise(
    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),
        sam=~mean(price, na.rm=TRUE),
        sasd=~sd(price, na.rm=TRUE)
      )
    ),
    count=n()
  )

#change the names
names(diamondTable) = gsub(pattern = "price_", replacement = "", x = names(diamondTable))

#improve and polish table
diamondTable%>%
  kable(
    caption="Summary Statistics for the Price of Diamonds by Cut",
    booktabs=TRUE,
    align=c("l", rep("c", 6))
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options=c("striped", "condensed"),
```

```
    font_size=16
  )
#load packages
library(dplyr)
library(dcData)

#load babynames data set and wrangle specific data from the data set
data(BabyNames)
#define names you want to find
chosen_names <- c("Justin", "Eric", "Zach", "Nick", "Bob", "Andrew")
Tmp6 <- BabyNames %>%
filter(name %in% chosen_names) %>%
group_by(name, year, sex) %>%
summarise(Freq2 = n()) %>%
arrange(desc(Freq2))
#view your wrangled data
View(Tmp6)
#Screenshot of file path
getwd()
setwd("~/Desktop/Stat 184/Activity10_MurrayOtis")
knitr::include_graphics("Screenshot 2023-11-29 at 5.16.52 PM.png")
```