First RMD File Activity

David Bednarczyk

2023-12-01

Collatz Conjecture

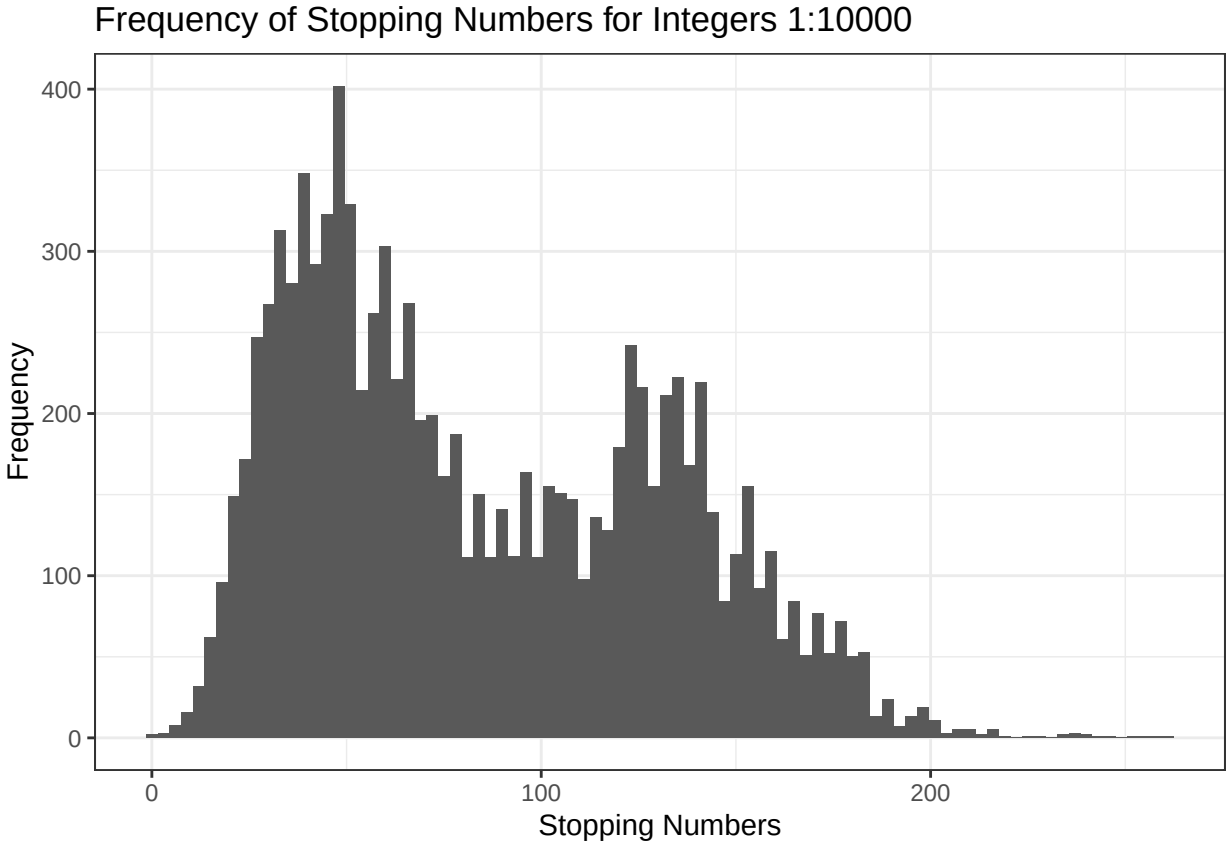
The Collatz Conjecture is a famous mathematics problem that has famously never been solved. It asks whether repeating the same two operations will eventually turn every positive integer into 1. The Collatz Conjecture is expressed by the following functions:

If x is even: f(x) = x/2

If x is odd: f(x) = 3x+1

If x is 1: STOP

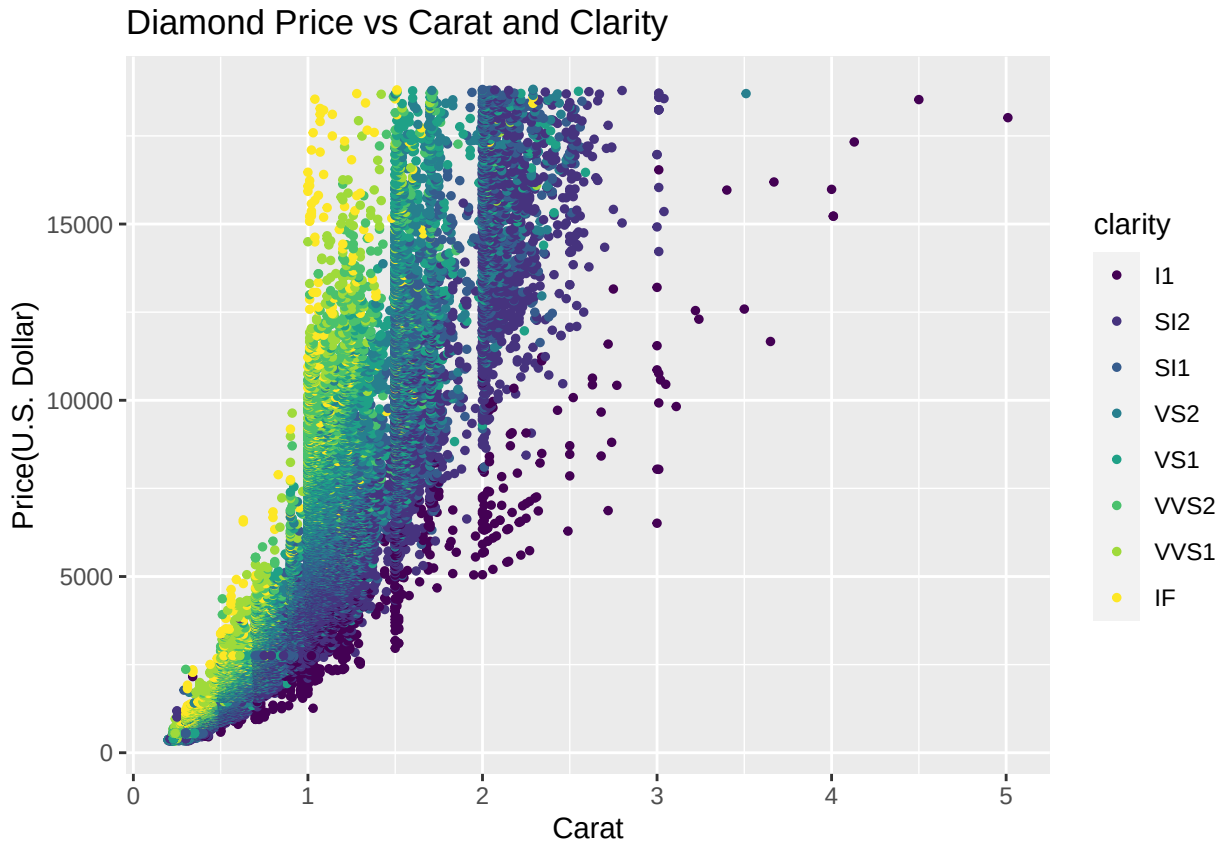
"Stopping Numbers" represent the number of steps each positive integer has to take from its original value to 1. Professor Hatfield gave us a challenge: use a histogram to plot the first 10,000 stopping numbers of the Collatz Conjecture. The following is the described histogram:



We can see that a majority of the inputs fell between 25 and 100. We also see a dip in frequency around 100, with a spike in frequency around stopping number 125. This shows the unpredictability of the stopping numbers. There are a small number of integers that have stopping numbers above 200. Though we cannot tell what integers give such high stopping number outputs, we can assume that there are integers between 1 and 10,000 that behave strangely within the constraints of the Collatz Conjecture.

Diamond Data

In Activity #5, we explored data visualizations using ggplot2. In Activity #7, we explored creating tables using dplyr and the kableExtra packages. In both of these activities, we used the diamonds data set found in the ggplot2 package. This diamonds data set houses information about ~54,000 diamonds including cut, price, carats, clarity, color, and size. In Activity #5, we were asked to create data visualizations that represent the relationship between two or more variables in the diamonds data set. The following scatter plot shows the relationship between price, cut, and carat of all ~54,000 diamonds.



The clarity scale goes from I1, the lowest quality to IF, the highest quality. As you can see, the higher quality diamonds reach the highest price values at lower carat values than lower quality diamonds. The entire right half of the graph displays low clarity diamonds with high carat values, but they still do not compare in price to the high clarity diamonds.

This table shows us some interesting things about the relationship between clarity and price. We can see that both the median and sample arithmetic mean for the two highest clarity diamonds are significantly lower than that of all lower clarity diamonds. The explanation for this is the carat values for these diamonds. Because higher clarity diamonds are less abundant than mediocre clarity diamonds, higher clarity diamonds weigh less than lesser clarity diamonds. This means that although the clarity is better, the diamond can weigh much less, causing it to cost much less.

Table 1: Statistics About the Price of Diamonds

clarity	Count	Minimum	FirstQ	ThirdQ	Median	Maximum	SA_Mean	SA_Sd
I1	741	345	2080.00	3344	3344	18531	3924.169	2806.778
SI2	9194	326	2264.00	4072	4072	18804	5063.029	4260.459
SI1	13065	326	1089.00	2822	2822	18818	3996.001	3799.484
VS2	12258	334	900.00	2054	2054	18823	3924.989	4042.303
VS1	8171	327	876.00	2005	2005	18795	3839.455	4011.748
VVS2	5066	336	794.25	1311	1311	18768	3283.737	3821.648
VVS1	3655	336	816.00	1093	1093	18777	2523.115	3334.839
IF	1790	369	895.00	1080	1080	18806	2864.839	3920.248

Reflection

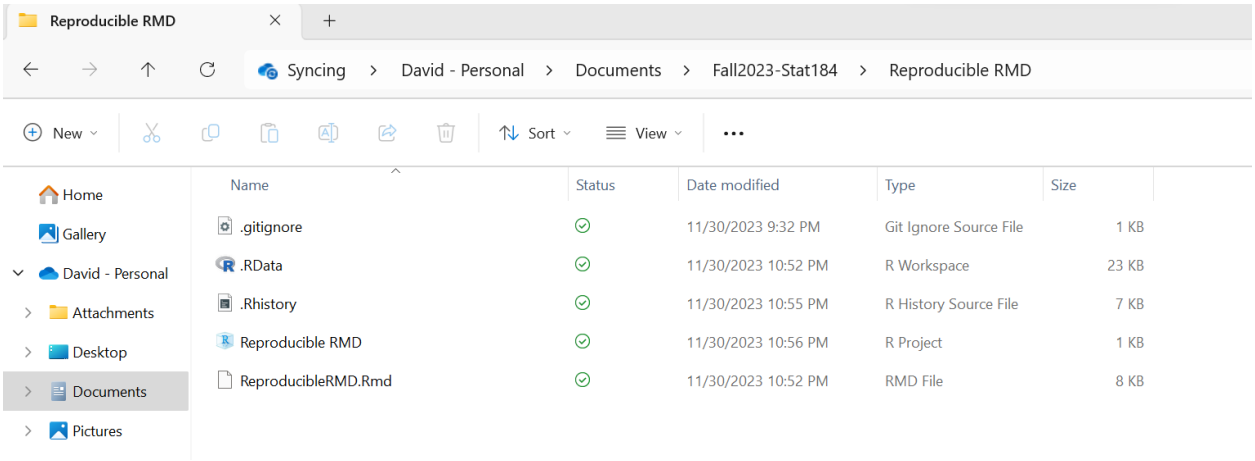
In Phase 1, our STAT 184 class talked about the basics of R and fundamental concepts that would help us throughout the course. We started by discussing the layout of R, including the console, source, environments, packages, and places to find help with anything in R. We also discussed data structures and types. Activity #2 was all about planning out our code and creating meaningful names to objects. In Activity #3, we learned about vectors and logic functions. The next section of Phase 1 was about tidy data, creating data frames, and sub-setting data frames.

In Phase 2, we moved past the very basic attributes of R and began discuss how we can use R to explore statistics. We talked about Exploratory Data Analysis and Confirmatory Data Analysis and the elementary perceptual tasks that we use to decipher data visualizations. We read about the different aspects of data visualizations according to Tufte and Kosslyn. Professor Hatfield introduced the PCIP system (Plan, Code, Improve, Polish) and we practiced it by creating our first data visualizations with ggplot2. We then moved into data wrangling using packages such as dplyr and explored how we can expand R with packages. Next, we learned how to create tables in R using the janitor, knitr, and kableExtra packages. We wrapped up Phase 2 by talking about Data Scraping with rvest.

We are now in Phase 3 and are learning about R Markdown, putting an emphasis on communication and conveying ideas to others in an easy to understand way. We discussed YAML headers, inline code and code chunks, knitting, and debugging in RMD.

Github Link

DavidBednarczyk/Reproducible_RMD: This Repo was created for Activity #10 in STAT 184 Section 4 (github.com)


Reproducible RMD
Syncing
David - Personal
Documents
Fall2023-Stat184
Reproducible RMD
New
Sort
View
Home
Gallery
David - Personal
Attachments
Desktop
Documents
Pictures
Name
Status
Date modified
Type
Size
.gitignore
11/30/2023 9:32 PM
Git Ignore Source File
1 KB
.RData
11/30/2023 10:52 PM
R Workspace
23 KB
.Rhistory
11/30/2023 10:55 PM
R History Source File
7 KB
Reproducible RMD
11/30/2023 10:56 PM
R Project
1 KB
ReproducibleRMD.Rmd
11/30/2023 10:52 PM
RMD File
8 KB

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE)
options(error = recover)
#Load packages with groundhog to improve stability
library("groundhog")
groundhog.library("ggplot2", '2023-11-28')
groundhog.library("tidyverse", '2023-11-28')
groundhog.library("conflicted", '2023-11-28')
groundhog.library("kableExtra", '2023-11-28')
conflict_prefer("filter", "dplyr")
conflict_prefer("lag", "dplyr")

# The following code creates a function to show both the end value(1), and the Stopping Number
run_collatz <- function(num, count = 0) {
  if (num == 1) {
    return(list(res = num, count = count))
  } else if (num %% 2 == 0) {
    num <- num / 2
  } else {
    num <- 3 * num + 1
  }

  return(run_collatz(num, count + 1))
}
#This function just returns the Stopping Number
get_stopping_num <- function(num) {
  result <- run_collatz(num)
  return(result$count)
}
#The following code creates a vector of starting numbers 1:10000
starting_numbers <- 1:10000
#The following creates a vector that houses the stopping numbers for integers 1:10000
stopping_numbers <- sapply(starting_numbers, get_stopping_num)
#The following turns the stopping numbers vector into a data frame that can be used to create a visuali
stopping_frame <- data.frame(StoppingNumbers = stopping_numbers)
#This ggplot code creates a histogram of the frequency of stopping numbers for the first 10000 integers
ggplot(stopping_frame, aes(x = StoppingNumbers))+
  geom_histogram(binwidth = 3)+
  labs(title = "Frequency of Stopping Numbers for Integers 1:10000",
       x= "Stopping Numbers",
       y= "Frequency")+
  theme_bw()
# The following code pipes the diamonds data plot into the ggplot function
diamonds %>%
# This ggplot code creates a scatter plot with carat on the x axis and price on the y axis, with clarity
ggplot(aes(x = carat, y = price, color = clarity))+
geom_point(size = 1)+
labs(x = "Carat",
y = "Price(U.S. Dollar)",
title = "Diamond Price vs Carat and Clarity")+
theme_gray()
#The following code creates a table about price, grouped by cut, using the diamonds data set. It includ
```

```r
diamondTable <- diamonds %>%
group_by(clarity) %>%
summarise(Count = n(),
Minimum = min(price),
FirstQ = quantile(price, .25),
ThirdQ = quantile(price, .5),
Median = median(price),
Maximum = format(max(price), nsmall = 3),
SA_Mean = format(mean(price), nsmall = 3),
SA_Sd = sd(price)
)
#The following code pipes the diamondTable into the kable function and makes it more presentable by cre
diamondTable %>%
kable(
caption = "Statistics About the Price of Diamonds",
booktabs = TRUE
) %>%
kable_styling(bootstrap_options = c("striped", "condensed"))
knitr::include_graphics("C:/Users/david/OneDrive/Documents/Fall2023-Stat184/Reproducible RMD/Reproducibl
```