Activity 10

Conor Long

2023-11-30

GitHub Link

Link to GitHub

Collatz Conjecture

The collatz conjecture asks if repeating the same arithmetic operations will turn every positive integer into 1. These operations are: divide the integer by 2 if even, multiply the integer by 3 and add 1 if it's odd. The question is, what is the distribution of "stopping times" for the first 10,000 positive integers. "Stopping times" refers to the number of times the function needs to be recursively invoked. Below is a histogram of 10,000 positive integers and the stopping times on the x-axis with their frequency on the y-axis. Based on the histogram, we can see that very few numbers require over 200 stopping times. The graph shows that the highest frequency of stopping times is around 50, with just over a frequency of 2,000. So out of 10,000 numbers, close to a quarter of them require around 50 stopping times according to the collatz conjecture.

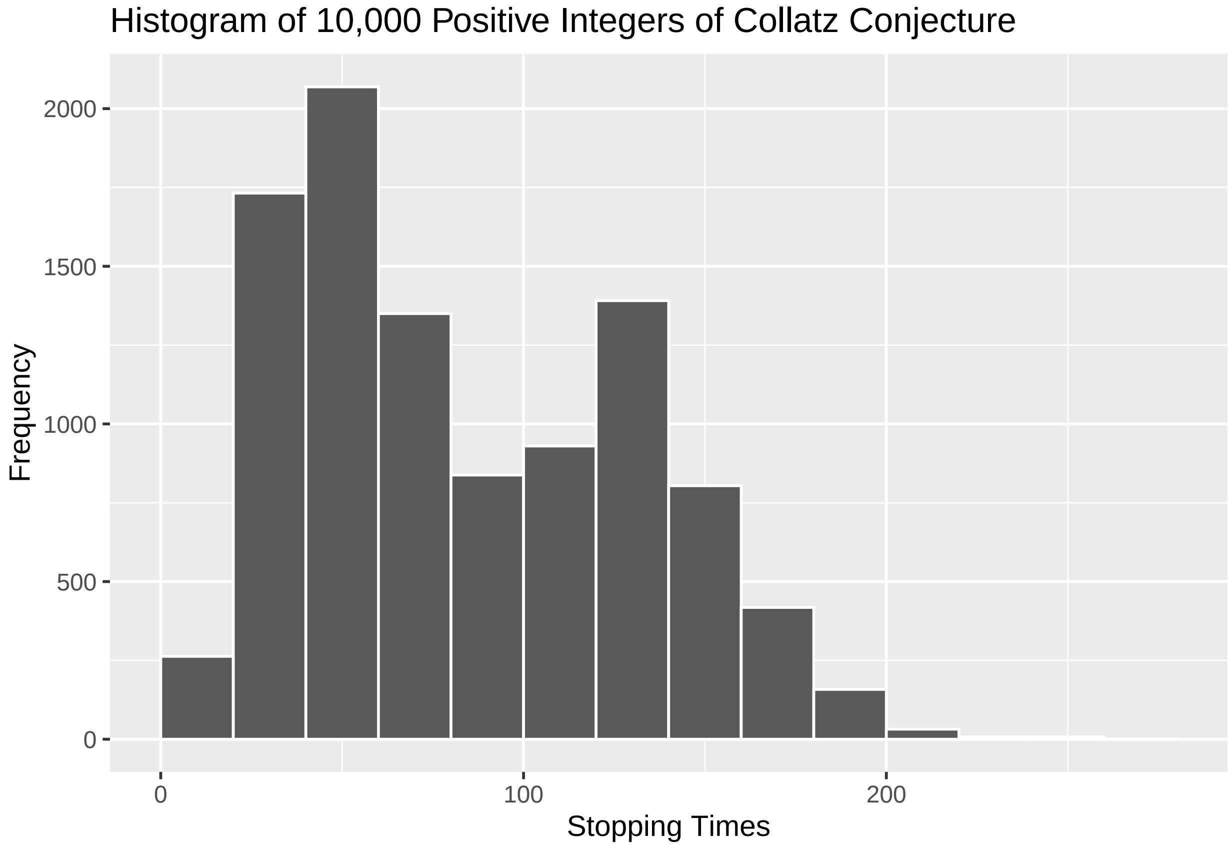
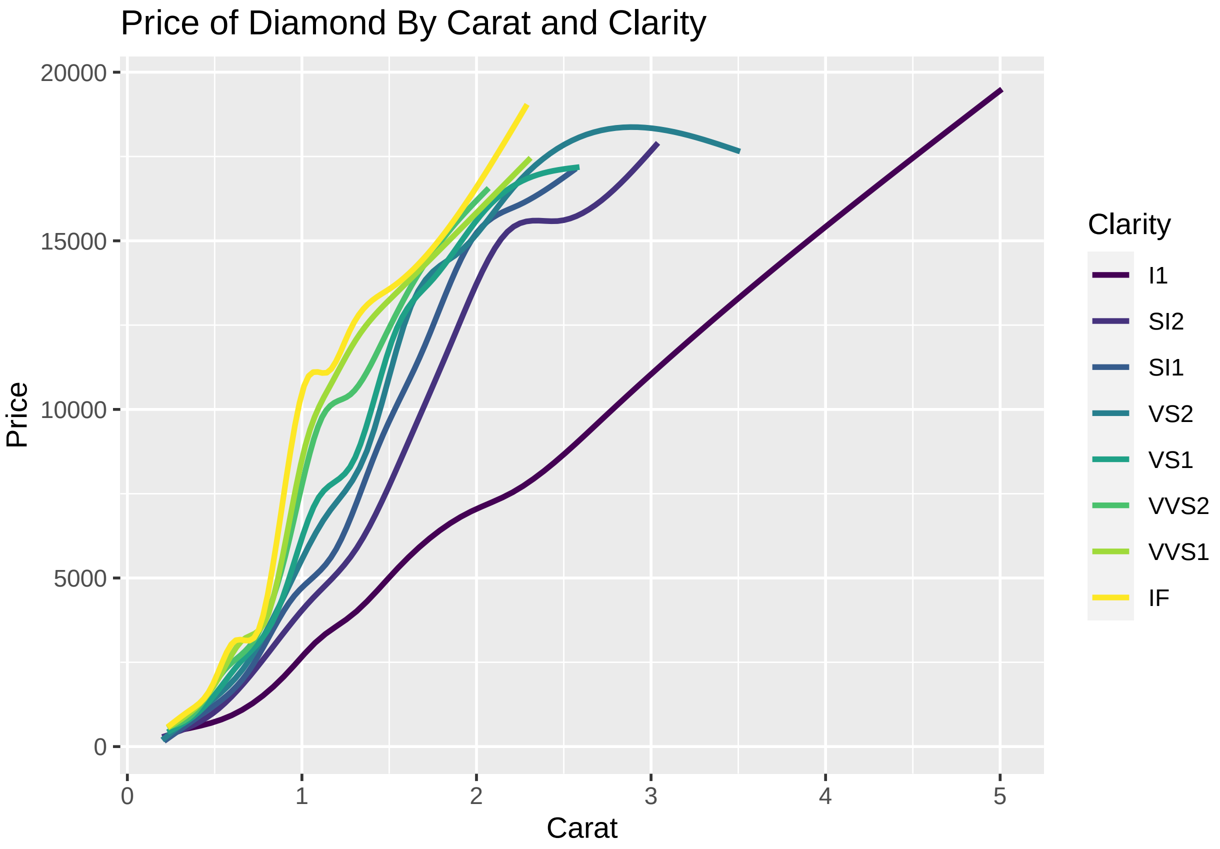


Table 1: Diamonds Summary Statistics Based on Cut and Price

Cut	Minimum	1st Quintile	2nd Quintile	Median	3rd Quintile	4th Quintile	Maximum	Mean	Standard Deviation	Count
Fair	337	1790.6	2805.0	3282.0	3947.4	6090.4	18574	4358.76	3560.39	1610
Good	327	876.0	2176.0	3050.5	3888.0	5834.0	18788	3928.86	3681.59	4906
Very Good	336	760.0	1892.4	2648.0	3751.6	6288.0	18818	3981.76	3935.86	12082
Premium	326	924.0	2100.0	3185.0	4408.0	7485.0	18823	4584.26	4349.20	13791
Ideal	326	803.0	1243.0	1810.0	2529.0	5613.0	18806	3457.54	3808.40	21551

Diamonds

By using the diamonds data set, we can use data visualizations to see what impacts the price of a diamond. By looking at the price, carat, and clarity of a diamond, we can use ggplot2 to create a data visualization showing what effect the carat and clarity of a diamond has on the price. Below shows a visualization with price on the y-axis, carat on the x-axis, and each color represents a different clarity. Based on the graph, we can see that diamonds with a clarity of IF do not need a high carat in order for the price to be high, compared to the clarity I1, which shows the highest carat measurement out of all the diamonds in this data set.



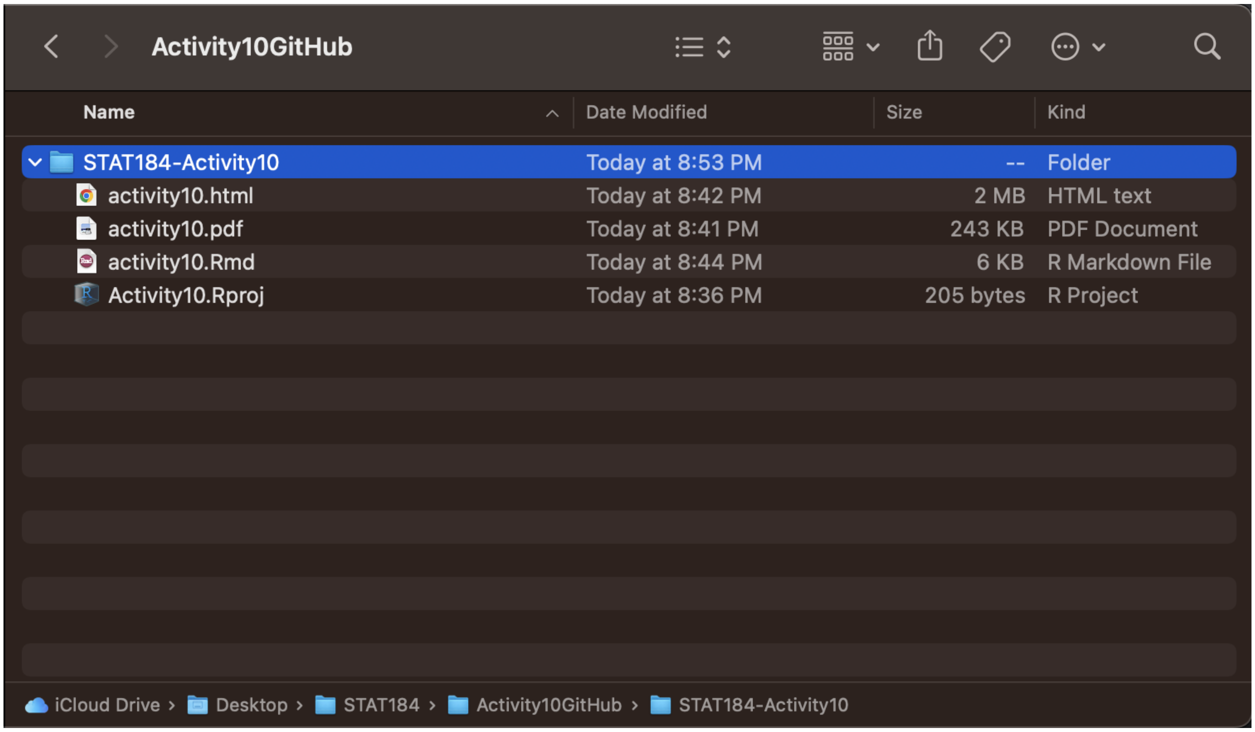
Using a summary table, we can compare price with cut. Based on the table, we can see that minimum and maximum prices are compareable between each cut. We can also see that the premium cut has the highest mean and standard deviation.

Course Takeaways

Throughout this course, I have learned how to use R to make data visualizations, summary tables, and use R for statistical analysis, just to name a few. I have learned what makes a bad data visualization from a great one, and how I should construct these visualizations using packages like ggplot2. I have learned how to create tables, specifically summary and frequency tables, using various R packages. One thing that I really enjoyed was getting data from a website to use. I found that to be useful in any personal projects that I might make in my free time. Overall, I feel more comfortable using R after being in this course.

Adding onto this, I now understand and know how to use GitHub, something that I've wanted to know how to use but never actually looked into. This is a good tool to have for the future.

Image of File Directory



Activity10GitHub

Name	Date Modified	Size	Kind
STAT184-Activity10	Today at 8:53 PM	--	Folder
activity10.html	Today at 8:42 PM	2 MB	HTML text
activity10.pdf	Today at 8:41 PM	243 KB	PDF Document
activity10.Rmd	Today at 8:44 PM	6 KB	R Markdown File
Activity10.Rproj	Today at 8:36 PM	205 bytes	R Project

iCloud Drive › Desktop › STAT184 › Activity10GitHub › STAT184-Activity10

Code Appendix

```r
knitr::opts_chunk$set(echo = TRUE, dpi = 300)
#Load packages with groundhog to improve stability
library("groundhog")
pkgs <- c("ggplot2", "knitr", "janitor", "dplyr", "kableExtra", "here")
groundhog.library(pkgs, '2023-11-28')
## Code for collatz conjecture
runCollatz <- function(num, count = 0) {
  if(num == 1) {
    # Since num equals 1, return the final stopping time
    return(count)
  }
  else if(num %% 2 == 0) {
    # Since even, divide num by 2, recursively call runCollatz
    # Increase count by one since it counts as a stopping time
    return(runCollatz(num/2, count = count + 1))
  }
  else {
    # Since odd, multiply num by 3 and add one, recursively call runCollatz
    # Increase count by one since it counts as a stopping time
    return(runCollatz((3*num)+1, count = count + 1))
  }
}

# Vectorize the function so we can get the 10,000 integers
vectorizeRunCollatz <- Vectorize(FUN = runCollatz)

## Code for collatz conjecture visualization

# Create data frame with 10,000 numbers using the Collatz Conjecture
value <- vectorizeRunCollatz(num = seq(1,10000))
collatzData <- data.frame(value)

# Calculates the bins using the Sturges method
numBreaks <- pretty(range(value), n = nclass.Sturges(value), min.n = 1)

# Use ggplot2 to graph histogram
ggplot(data = collatzData, mapping = aes(x = value)) +
  geom_histogram(breaks = numBreaks, color = "white") +
  labs(x = "Stopping Times",
       y = "Frequency",
       title = "Histogram of 10,000 Positive Integers of Collatz Conjecture")

## Diamonds data set visualization

data(diamonds)
ggplot(data = diamonds, mapping = aes(x=carat, y=price, color=clarity)) +
  geom_smooth(se = FALSE) +
  labs(x = "Carat",
       y = "Price",
       color = "Clarity",
       title = "Price of Diamond By Carat and Clarity")
```

```r
## Diamonds data set table

# Import data
data("diamonds", package = "ggplot2")

# Create the data table
priceStats <- diamonds %>%
  group_by(cut) %>%
  select(cut, price) %>%
  summarize(
    across(
      .cols = where(is.numeric),
      .fns = list(
        min = ~min(price, na.rm = TRUE),
        q1 = ~quantile(price, probs = 0.2, na.rm = TRUE),
        q2 = ~quantile(price, probs = 0.4, na.rm = TRUE),
        median = ~median(price, na.rm = TRUE),
        q3 = ~quantile(price, probs = 0.6, na.rm = TRUE),
        q4 = ~quantile(price, probs = 0.8, na.rm = TRUE),
        max = ~max(price, na.rm = TRUE),
        smean = ~mean(price, na.rm = TRUE),
        sasd = ~sd(price, na.rm = TRUE)
      )
    ),
    count = n()
  )

# Format the table with kable

priceStats %>% kable(
      col.names = c("Cut", "Minimum", "1st Quintile",
                    "2nd Quintile", "Median", "3rd Quintile",
                    "4th Quintile", "Maximum", "Mean",
                    "Standard Deviation", "Count"),
      align = c("l", rep("c", 10)),
      digits = 2,
      caption = "Diamonds Summary Statistics Based on Cut and Price"
      ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 16, latex_options = "scale_down"
  )
# include_graphics will insert an image into the rmd file, using the relative path
knitr::include_graphics("../filePathProof.png")
```