Gaurang Kakade's First RMD File (Activity 8)

Gaurang Kakade

2022-10-28

Collatz Conjecture

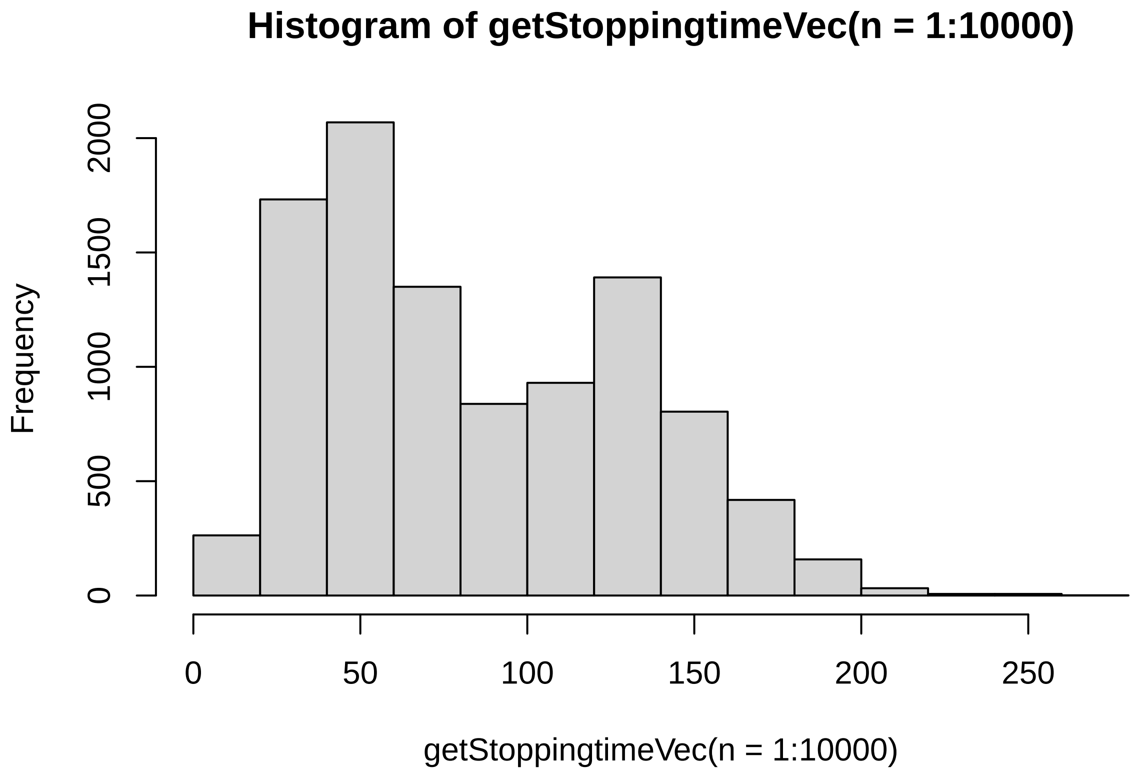
The Collatz conjecture says that for every number n that is a positive integer, a sequence created with the function above will eventually fall to the number one. So, according to Collatz Conjecture, when we take any natural number n, if n is even divide it by 2 to get n/2, if n is odd, multiply it by 3 and add 1 to obtain 3n+1. Then repeat this process until you reach 1.

```
counter <<- 0
### creates a function called getStopping time to find the stopping number
getStoppingtime <- function(n) {

### If the starting interger is even, iterate stopping number by 1 and then call collatz counter now sta
if(n %% 2 == 0){
total<- n/2
counter <<-counter+1

return (getStoppingtime(total))

}
### if starting integer is 1, then stop and return 0
else if (n == 1){
count2 <- counter
counter <<- 0
return (count2)

return (getStoppingtime(total))

}
### If the starting integer is odd, iterate the stopping number by 1 and then call collatz counter on 3
else {
total <- (3*n) + 1
counter <<- counter + 1
return (getStoppingtime(total))
}
}
getStoppingtime(3)
```

```
## [1] 7
```

```r
### Using Vectorize
### Create a vectorized form of getstoppingtime
getStoppingtimeVec <- Vectorize(
  FUN = getStoppingtime,
  vectorize.args = "n"
)
```

```r
### Create a histogram of stopping numbers for the first 10,000 postive integers
hist ( x = getStoppingtimeVec(n=1:10000))
```



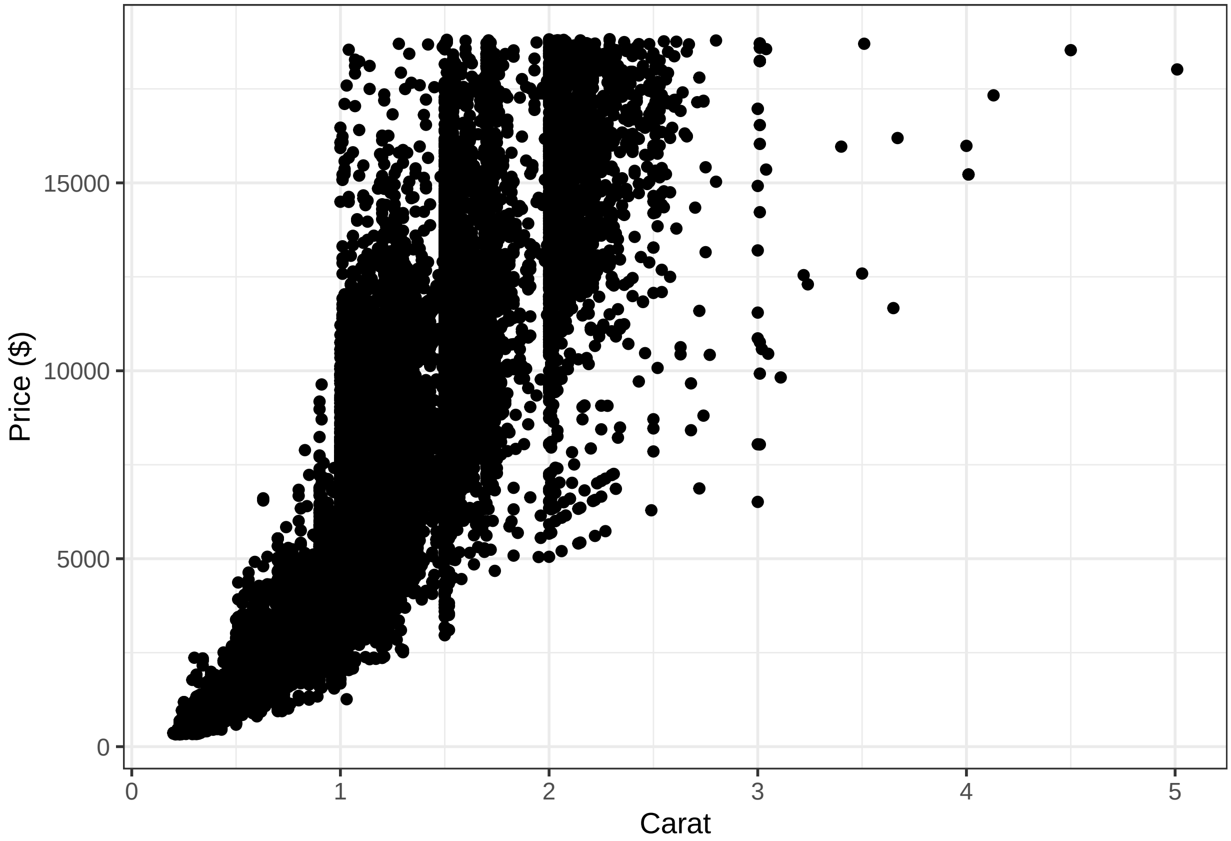
Explanation

From the data visualization we can infer that the majority of the stopping time takes place between 50 and 200.The frequency between 0 and 50 have values between half of 500 to a little more than 1500. At 50, the frequency reaches to more than 2000. We then observe a dip and a rise (sort of a valley structure) in the frequency between 50 and 150. The frequency after 150 seems to be gradually decreasing and eventually becoming equal to 0 after 200.

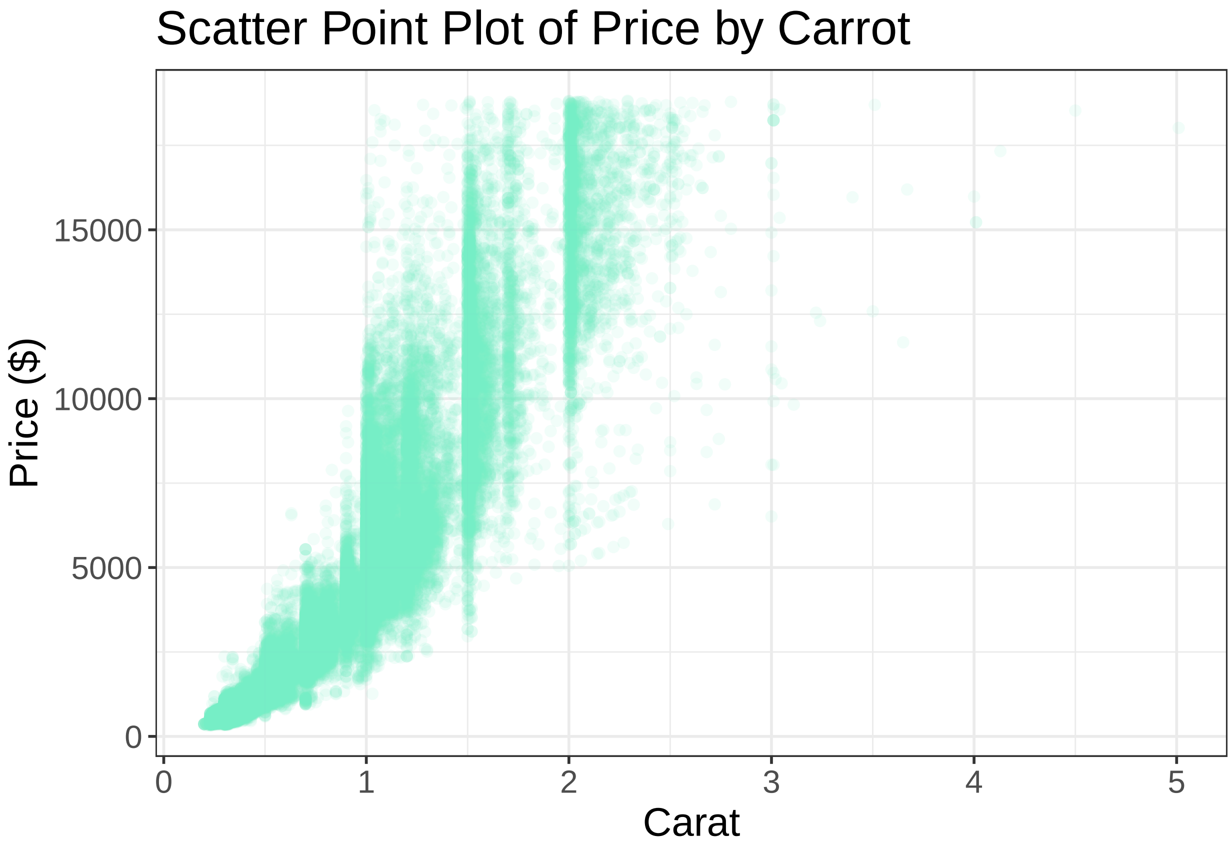
Diamonds Data set (Activity 5 and Activity 7)

In the diamonds data set we observe that there are 10 different attributes such as carat, cut, color, clarity, depth, table, price, x, y and z which describe an individual case which in the current scenario is an individual diamond. From the diamonds data set, we can observe and learn a lot about the price of the diamonds and the tendencies through different comparisons between the various other attributes and the price of the diamond. I have chosen to use the price of the diamond with the attribute carat to observe unique trend that it showcases.

```
## Improve ---
### The default fill makes seeing the midline impossible
### The labels could use improvement and units of measure
### Perhaps a different theme might be useful
### The code could be better organized and commented

### Improved Code ----
# Create the framework and map carat hole to horizontal, price to vertical

### Loading ggplot2
library(ggplot2)
ggplot(
  data = diamonds,
  mapping = aes(x = carat,y = price)
)+
  geom_point() + # Make a pointplot to show the ordered structure, no fill
  theme_bw() + #  try the black and white theme
  ylab("Price ($)") + # improve axis label
  xlab('Carat')
```



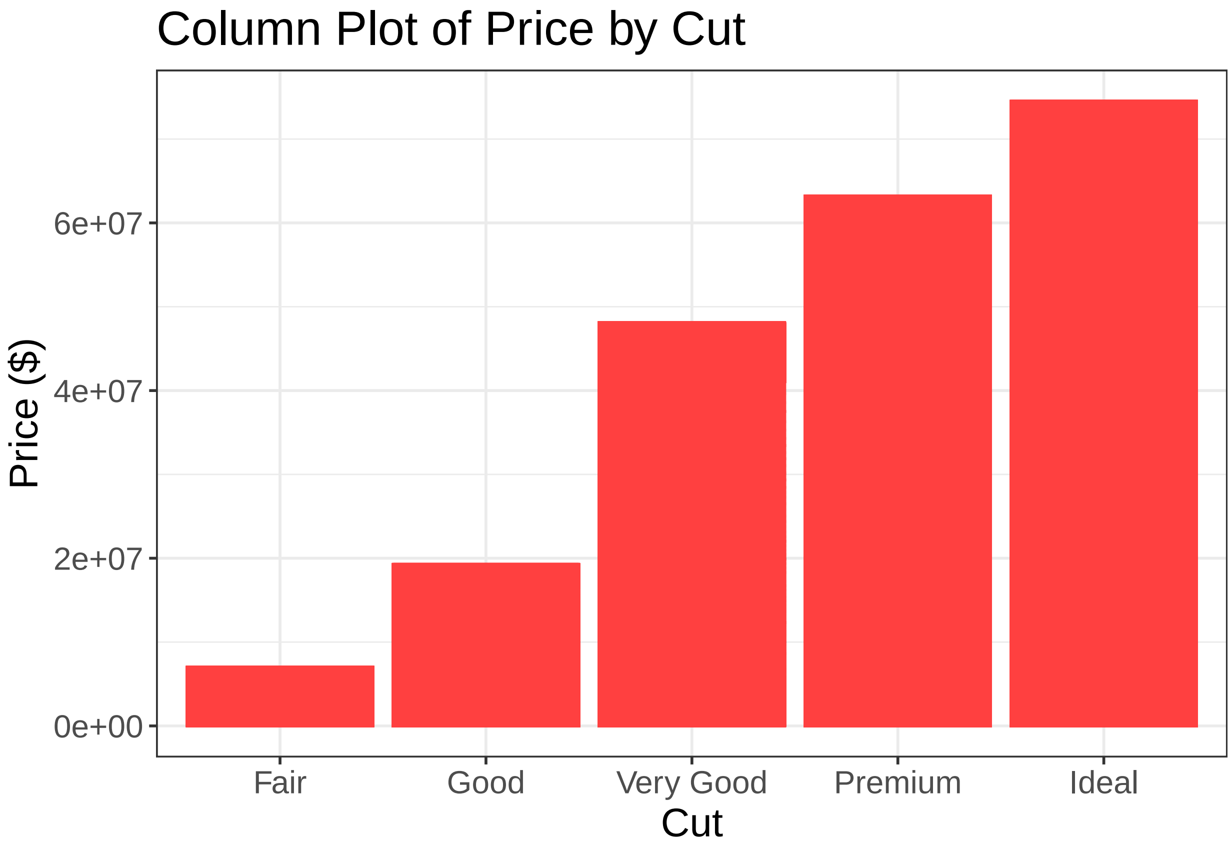
```
## Polish ----
### The font size is a bit small
### A fill color would help make the visualization pop
### Add a title
### Add transparency

ggplot(
  data = diamonds,
  mapping = aes(x = carat,y = price) ### aes is basically telling it to plot carat on the x-axis and pr
) +
  geom_point(alpha = 0.1, color = "aquamarine2") + # Make a pointplot to show the ordered structure, no
  theme_bw() + #  try the black and white theme
  ylab("Price ($)") + # improve axis label
  xlab('Carat') +
  theme(
    text = element_text(size = 15) # Change the base font size
  ) +
  ggtitle("Scatter Point Plot of Price by Carrot") # Add title
```



Explanation of the Visualization 1

From the data visualization, we are trying to how does the ordered structure of carat, cut, colour, clarity, depth, table, price, x, y, and z varies by each diamond. There are a total of 53940 diamonds and our aim is that we are trying to look at the price by carat. So we have made a point plot to show the ordered structure where Price in dollars is the y label and carat is the x label. So we can observe how the price of a diamond varies depending on the carat. This tells us the curvature (EPT) of the curve of the scatter plot. So we observe that as the carat increase so does the price of the diamond increases. By the notion of Position EPT along an aligned scale with grid lines, we can see that a 1-carat diamond roughly costs around 4900, 2-carat costs roughly between 5000 and 15000 and 3-carat surpasses the price of 15000. Another EPT that can be observed is the length. When the diamond is of 1 carat we can see the range of length of price(2500 to 5000) is less than the range of length of price for the 2-carat diamond(6000 to 18000). So we see the length of the price is not aligned and the grid lines help us to figure out the length of the price range.

```
ggplot(
  data = diamonds,
  mapping = aes(x = cut,y = price) ### aes is basically telling it to plot cut on the x-axis and price
) +
  geom_col(alpha = 0.1, color = "brown1") + # Make a pointplot to show the ordered structure, no fill
  theme_bw() + #  try the black and white theme
  ylab("Price ($)") + # improve axis label
  xlab('Cut') +
  theme(
    text = element_text(size = 15) # Change the base font size
  ) +
  ggtitle("Column Plot of Price by Cut") # Add title
```



Explanation of the Visualization 2

From the data visualization we can observe that visualization is a column plot of price by cut. So we have made a column plot to show the ordered structure where Price in dollars is the y label and cut is the x label. So we can observe how the price of a diamond varies depending on the cut. When the cut is equal to Fair, we can observe that the price in dollars is less than half of 2e+07. When the cut is Good, the price in dollars is nearly equal to 2e+07. Now, when the cut is Very Good, the price jumps to more than 4e+07. When the cut is Premium and Ideal, the price in dollars is the most as it surpasses 6e+07. Hence, we can perceive that as the cut becomes better and better in terms of quality the price in dollars goes on increasing.

```
# Load Packages ----
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
```

```
library(knitr)
library(kableExtra)
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : ’length(x) = 2 > 1’ in coercion to ’logical(1)’

##
## Attaching package: ’kableExtra’

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

```
library(psych)
```

```
##
## Attaching package: ’psych’

## The following objects are masked from ’package:ggplot2’:
##
##     %+%, alpha
```

```
# Load data ----
data("diamonds", package = "ggplot2")
# Build Summary table ----
## Dplyr Approach
```

Table 1: Summary Statistics for Diamond length (mm)

Cut	Count	Minimum	20%-tile	40%-tile	Median	60%-tile	80%-tile	Max	SAM	SASD
Fair	1,610	0	5.56	6.04	6.18	6.29	7.06	10.74	6.25	0.96
Good	4,906	0	4.72	5.63	5.98	6.22	6.61	9.44	5.84	1.06
Very Good	12,082	0	4.60	5.37	5.74	6.13	6.64	10.01	5.74	1.10
Premium	13,791	0	4.68	5.67	6.11	6.44	6.98	10.14	5.97	1.19
Ideal	21,551	0	4.45	4.95	5.25	5.70	6.56	9.65	5.51	1.06

```r
sumTable1 <- diamonds %>%
  group_by(cut) %>%
  summarize(
    count = n(),
    minimum = min(x, na.rm = TRUE),
    firstQuin = quantile(x, probs = 0.2, na.rm = TRUE),
    secondQuin = quantile(x, probs = 0.4, na.rm = TRUE),
    median = median(x, na.rm = TRUE),
    thirdQuin = quantile(x, probs = 0.6, na.rm = TRUE),
    fourthQin = quantile(x, probs = 0.8, na.rm = TRUE),
    max = max(x, na.rm = TRUE),
    SAM = mean(x, na.rm = TRUE),
    SASD = sd(x, na.rm = TRUE)
)
# Create the pretty table ----
sumTable1 %>%
  kable(
    digits = 2,
    format.args = list(big.mark = ","),
    caption = "Summary Statistics for Diamond length (mm)",
    col.names = c("Cut", "Count", "Minimum", "20%-tile", "40%-tile", "Median",
                  "60%-tile", "80%-tile", "Max", "SAM", "SASD"),
    align = c("l", rep("c", 10)),
    booktabs = TRUE
  ) %>%
  kableExtra::kable_classic()
```

Explanation of the summary table

We can infer that table consists of calculated values of the following statistics cut count, minimum, first QUIN-tile, second QUIN-tile, median, third QUIN-tile, fourth QUIN-tile, maximum, arithmetic mean, and the arithmetic standard deviation by the type of cut. The table has 5 rows differentiated on the basis of cut (Fair, Good, Very Good, Premium, and Ideal).

Reflections

The course so far has given me a lot of knowledge about applications of R in data science. I have become knowledgeable with respect to how thousands of user-created packages are publicly available to extend the capabilities of R. The course is categorized in 4 phases. In the first phase, I learned about the basic principles of R, data types and structures, Functions in R, and how to create tidy Data. We had respective activities for each of the sub parts to get a clear understanding and good grasp over the concepts. Activity 3 was based on Functions where I practiced defining and working with functions in R. Activity 4 was based on Tidy Data where I was asked to tidy a given set of data with keeping in mind the rules for the tidying data. In the second phase, I learned about the difference between Elementary Data Analysis and Confirmatory Data Analysis and the notion of their data narratives, Elementary Perceptual Tasks, Data Visualizations and their respective grammer, The PCIP (plan, code, improve and polish) process, Data wrangling and Tables. HW 5.1 and 5.2 introduced us to the topic of data visualizations according to Kosslyn and Tufte. For PCIP process, I learned how to implement it by working on the Galton Family Data and the Army Marital Status Data. I then learned how to create different types of visualizations using the ggplot2 package in the tidyverse collection. We also covered the notion of facets through facet_wrap and facet_grid. For Data Wrangling, I was introduced to three different ways to classify wrangling verbs. For creating tables, I learned about different packages such as janitor for creating frequency tables, knitr package's kable function which helps in creating nice looking tables that can be further improved/customized with the functions of the kableExtra package. In phase 3, I learning about R markdown, different coding styles that are used and activity 8 covers creating my first R markdown file.