Homework-3: ’Preston Robertson

All of these homework problems are from the *R for Data Science* book. The section numbers (e.g., “3.2.4 Exercises”) refer to sections in this book. Although the questions are based on those in the book, some questions ask for additional details or analysis.

When solving these problems, you are allowed to use any method from the book or class, even if that method wasn’t yet covered when the exercise was presented in the book.

Write answers that are as complete as possible. If a graph is helpful for formalizing the solution, provide the graph. If a table is helpful, provide a table. In the text part of the answer, outline the progression in your thinking as you perform the analysis.

Note that you should type your answers in RStudio, by typing into the file **Homework-3.sa.Rmd**.

flights

# A tibble: 336,776 x 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
1 2013 1 1 517 515 2 830 819  
2 2013 1 1 533 529 4 850 830  
3 2013 1 1 542 540 2 923 850  
4 2013 1 1 544 545 -1 1004 1022  
5 2013 1 1 554 600 -6 812 837  
6 2013 1 1 554 558 -4 740 728  
7 2013 1 1 555 600 -5 913 854  
8 2013 1 1 557 600 -3 709 723  
 arr\_delay carrier flight tailnum origin dest air\_time distance hour minute  
 <dbl> <chr> <int> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
1 11 UA 1545 N14228 EWR IAH 227 1400 5 15  
2 20 UA 1714 N24211 LGA IAH 227 1416 5 29  
3 33 AA 1141 N619AA JFK MIA 160 1089 5 40  
4 -18 B6 725 N804JB JFK BQN 183 1576 5 45  
5 -25 DL 461 N668DN LGA ATL 116 762 6 0  
6 12 UA 1696 N39463 EWR ORD 150 719 5 58  
7 19 B6 507 N516JB EWR FLL 158 1065 6 0  
8 -14 EV 5708 N829AS LGA IAD 53 229 6 0  
 time\_hour   
 <dttm>   
1 2013-01-01 05:00:00  
2 2013-01-01 05:00:00  
3 2013-01-01 05:00:00  
4 2013-01-01 05:00:00  
5 2013-01-01 06:00:00  
6 2013-01-01 05:00:00  
7 2013-01-01 06:00:00  
8 2013-01-01 06:00:00  
# ... with 336,768 more rows

## 5.6.7 Exercises

### (1) 5.6.7 Exercise 1 (25 pts; 5 each)

Brainstorm at least 4 different ways to assess the typical delay characteristics of a group of flights. Consider the following scenarios:

* A flight is 15 minutes early 50% of the time, and 15 minutes late 50% of the time.
* A flight is always at least 10 minutes late.
* A flight is 30 minutes early 50% of the time, and 30 minutes late 50% of the time.
* 99% of the time a flight is on time. 1% of the time it’s 2 hours late.

For each scenario, using the flights dataset, give the analysis, give the answer, and then discuss the findings.

## If the plane is always late or early by only 15mins then it could be due to the how the plane is loaded and if one person  
  
## The crew probably starts loading the plane too late, or have issues keeping the schedule due to where the hangar is.  
  
## A difference of 30 minutes is very large and makes me question how a team could manage to consistently be so far apart.  
  
## This would be considered an outline than is probably caused by uncontrollable factors such as lag from other planes.

Then discuss this question: *Which is more important: arrival delay or departure delay?* Why? Explain your reasoning.

### (2) 5.6.7 Exercise 3 (5 pts)

Our definition of cancelled flights (is.na(dep\_delay) | is.na(arr\_delay)) is slightly suboptimal. Why? Which is the most important column?

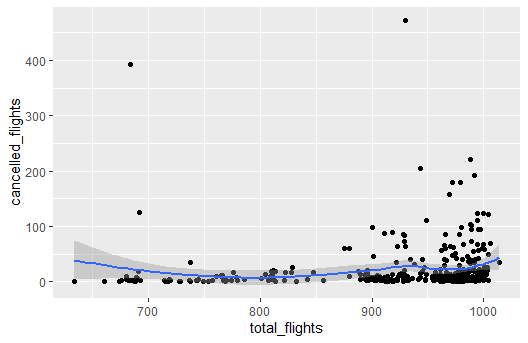
## The most important column is the arrival delay since it gives more information such as possible crash or re-routing flight paths (for fuel or because there is no room in the airport.)

### (3) 5.6.7 Exercise 4 (10 pts)

Look at the number of cancelled flights per 24-hour day. Is there a pattern? Is the proportion of cancelled flights related to the average delay?

## Plot Below, the answer is yes there seems to be a pattern  
  
daily\_cancels <-   
 flights %>%  
 mutate(cancels = (is.na(arr\_delay) > 1 | is.na(dep\_delay))) %>%  
 group\_by(year, month, day) %>%  
 summarise(  
 cancelled\_flights = sum(cancels),  
 total\_flights = n(), # Counts the number of flights  
 )  
  
# Each point is a single day's cancelled and total number of flights  
  
ggplot(data = daily\_cancels, mapping = aes(x = total\_flights, y = cancelled\_flights)) +  
 geom\_point() +  
 geom\_smooth()

`geom\_smooth()` using method = 'loess' and formula 'y ~ x'



### (4) 5.6.7 Challenging Exercise (10 pts)

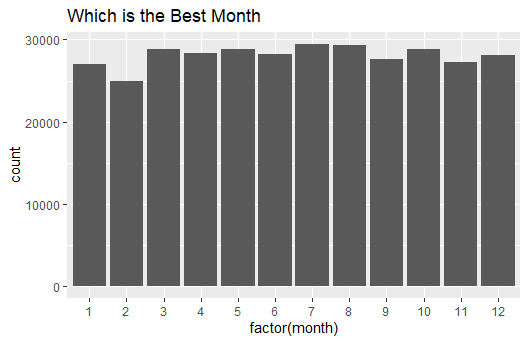
For each plane, count the number of flights before the first delay of greater than 1 hour.

*Hint*: I found this exercise, which has since been removed from the book, to be challenging but rewarding. A solution involving a for loop is easy to conceptualize. But, the Tidyverse way, and also the more efficient way, is to avoid explicit loops. Among various loop-less elegant solutions is to use row\_number() to number the flights for each plane.

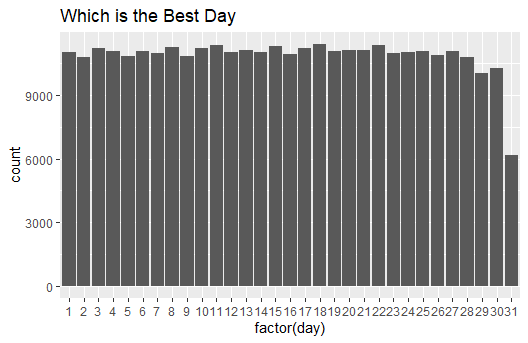
## Plot Below, the answer is yes there seems to be a pattern  
  
before\_delay <- flights %>%  
 mutate(cancels = (is.na(arr\_delay) > 1 | is.na(dep\_delay))) %>%  
 group\_by(month, day) %>%  
 summarise(  
 count = n(),  
 agg\_dep\_delay = sum(cumsum(dep\_delay > 60) < 1)  
 )  
  
before\_delay

# A tibble: 365 x 4  
# Groups: month [12]  
 month day count agg\_dep\_delay  
 <int> <int> <int> <int>  
1 1 1 842 NA  
2 1 2 943 NA  
3 1 3 914 NA  
4 1 4 915 NA  
5 1 5 720 NA  
6 1 6 832 NA  
7 1 7 933 NA  
8 1 8 899 NA  
# ... with 357 more rows

ggplot(data = before\_delay,mapping = aes(x = factor(month), y = count)) +  
 geom\_col() + ggtitle("Which is the Best Month")



ggplot(data = before\_delay,mapping = aes(x = factor(day), y = count)) +  
 geom\_col() + ggtitle("Which is the Best Day")



## The day graph is skewed since some months do not have 30 days and even less have 31 days.

## 5.7.1 Exercises

### (5) 5.7.1 Exercise 2 (10 pts)

Which plane (tailnum) has the worst on-time record?

flights %>%  
 mutate(cancels = (is.na(arr\_delay) > 1 | is.na(dep\_delay))) %>%  
 group\_by(tailnum) %>%  
 summarise(  
 count = n(),  
 max\_arrival\_delay = max(arr\_delay),  
 percentage\_on\_time = mean(arr\_delay <= 0, na.rm = TRUE)  
 ) %>%  
 filter(count > 50) %>% #to keep from having the flights with a small number of trips messing up the analysis  
 arrange(desc(percentage\_on\_time))

# A tibble: 2,093 x 4  
 tailnum count max\_arrival\_delay percentage\_on\_time  
 <chr> <int> <dbl> <dbl>  
1 N553AA 52 NA 0.863  
2 N4YNAA 70 NA 0.841  
3 N854VA 87 201 0.839  
4 N363NB 86 274 0.826  
5 N3753 130 NA 0.822  
6 N852VA 78 NA 0.818  
7 N579AA 51 NA 0.816  
8 N4XHAA 53 NA 0.808  
# ... with 2,085 more rows

## Going to the last page and bottom shows the worst, while the best is at the top, a weird bug was happening when I was attempting to sort the other way.

### (6) 5.7.1 Exercise 3 (10 pts)

What time of day should you fly if you want to avoid delays as much as possible?

flights %>%  
 mutate(cancels = (is.na(arr\_delay) > 1 | is.na(dep\_delay))) %>%  
 mutate(dep\_time\_hr = dep\_time %/% 100 ) %>%  
 group\_by(dep\_time\_hr) %>%  
 summarise(  
 count = n(),  
 avg\_arrive\_delay = mean(arr\_delay,na.rm = TRUE)  
 ) %>%  
 filter(count > 30)

# A tibble: 24 x 3  
 dep\_time\_hr count avg\_arrive\_delay  
 <dbl> <int> <dbl>  
1 0 881 121.   
2 1 223 196.   
3 2 64 225.   
4 4 305 -8.88  
5 5 7246 -8.85  
6 6 22699 -7.09  
7 7 21461 -7.35  
8 8 26424 -4.03  
# ... with 16 more rows

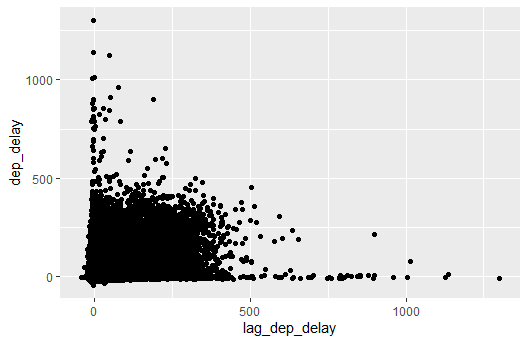
## Hour 11 is the best based on average delay.

### (7) 5.7.1 Exercise 5 (10 pts)

Delays are typically temporally correlated: even once the problem that caused the initial delay has been resolved, later flights are delayed to allow earlier flights to leave. Using lag(), explore how the delay of a flight is related to the delay of the immediately preceding flight.

flights %>%  
 mutate(cancels = (is.na(arr\_delay) > 1 | is.na(dep\_delay))) %>%  
 group\_by(origin) %>%  
 arrange(year, month,day,dep\_time) %>%  
 mutate(lag\_dep\_delay = lag(dep\_delay)) %>% #Lag shows the progressive delay in time series data  
 ggplot(aes(lag\_dep\_delay,dep\_delay))+  
 geom\_point()

Warning: Removed 9127 rows containing missing values (geom\_point).



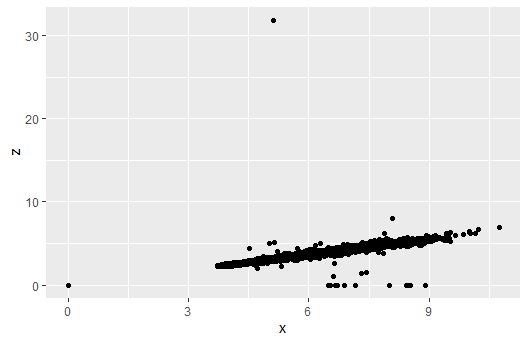
# There is a positive correlation between lag and departure delay. This makes sense since there is a logical reason for the next delays.

## 7.3.4 Exercises

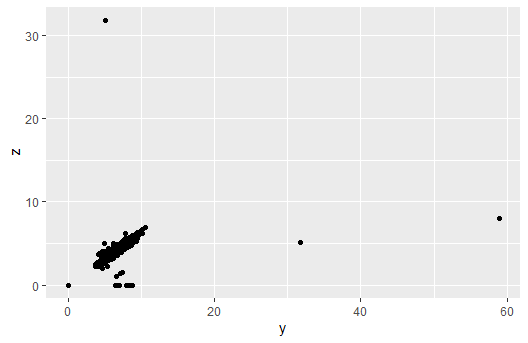
### (8) 7.3.4 Exercise 1 (10 pts)

Explore the distribution of each of the x, y, and z variables in diamonds. What do you learn? Think about a diamond and how you might decide which dimension is the length, width, and depth.

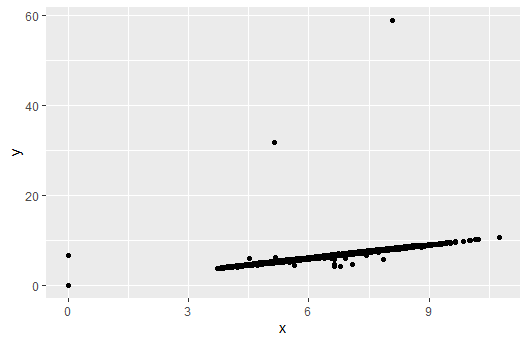
# Something to be learned from the plots below are that the "x" component is always the smallest size which is probably how they determine which side is length, width, and depth.  
  
ggplot(diamonds, aes(x = x, y = z)) +  
 geom\_point()



ggplot(diamonds, aes(x = y, y = z)) +  
 geom\_point()



ggplot(diamonds, aes(x = x, y = y)) +  
 geom\_point()



### (9) 7.3.4 Exercise 3 (5 pts)

How many diamonds are 0.99 carat? How many are 1 carat? What do you think is the cause of the difference?

diamonds %>%  
 filter(carat >= 0.99, carat <= 1) %>%  
 count(carat)

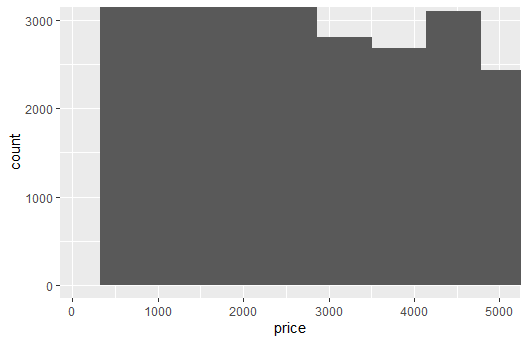
# A tibble: 2 x 2  
 carat n  
 <dbl> <int>  
1 0.99 23  
2 1 1558

# The difference is due to the goal behind the diamond grinder to not reduce a diamond below a specific size. All the .99 carat diamonds might have been damaged later or a mistake when grinding.

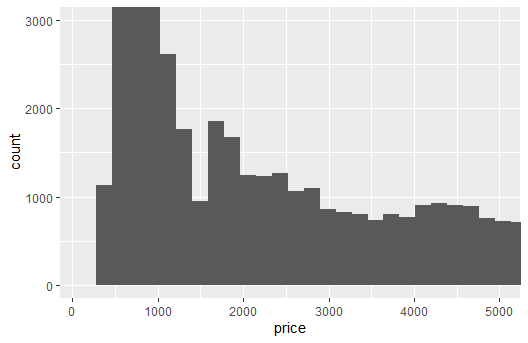
### (10) 7.3.4 Exercise 4 (5 pts)

Compare and contrast coord\_cartesian() vs xlim() or ylim() when zooming in on a histogram. What happens if you leave binwidth unset? What happens if you try and zoom so only half a bar shows?

ggplot(diamonds) +  
 geom\_histogram(mapping = aes(x = price), bins = 30) +  
 coord\_cartesian(xlim = c(100, 5000), ylim = c(0, 3000))



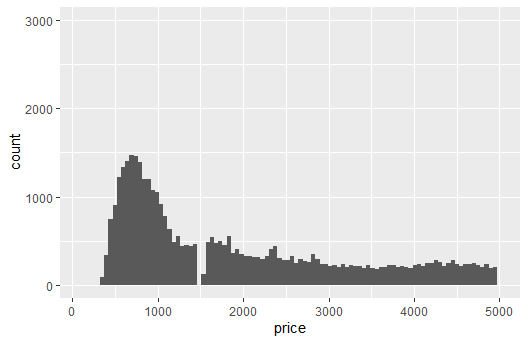
ggplot(diamonds) +  
 geom\_histogram(mapping = aes(x = price), bins = 100) +  
 coord\_cartesian(xlim = c(100, 5000), ylim = c(0, 3000))



ggplot(diamonds) +  
 geom\_histogram(mapping = aes(x = price), bins = 100) +  
 xlim(100, 5000) +  
 ylim(0, 3000)

Warning: Removed 14714 rows containing non-finite values (stat\_bin).

Warning: Removed 2 rows containing missing values (geom\_bar).



## The zoom function helps when trying to look at the graphs, looking only half a bar would show nothing but a filled in histogram box.

## 7.4.1 Exercises

### (11) 7.4.1 Exercise 1 (5 pts)

What happens to missing values in a histogram? What happens to missing values in a bar chart? Why is there a difference?

## Depends on the histogram, it is stated below as: "Removed ... rows containing non-finite values (stat\_bin)"  
  
## An N/A column is created.   
  
## The difference is in categorical variables vs continuous variables.

### (12) 7.4.1 Exercise 2 (5 pts)

What does na.rm = TRUE do in mean() and sum()?

## Removes N/A values from the calculations before.

## 7.5.1.1 Exercises

### (13) 7.5.1.1 Exercise 1 (10 pts)

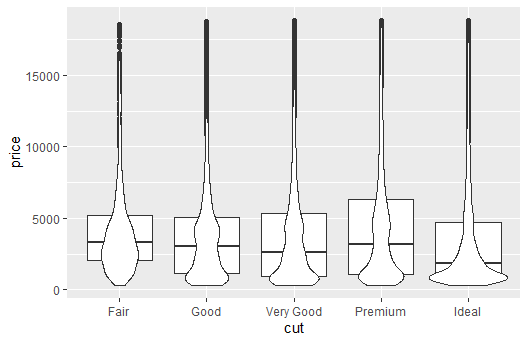
Use what you’ve learned to improve the visualization of the departure times of canceled vs. non-canceled flights. Explain the ways in which the visualization is an improvement.

## Come back to after done.

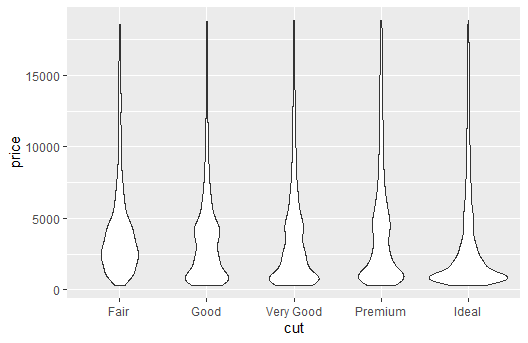
### (14) 7.5.1.1 Exercise 4 (10 pts)

One problem with boxplots is that they were developed in an era of much smaller datasets and tend to display a prohibitively large number of outlying values. One approach to remedy this problem is the violin plot. For the diamonds dataset, examine the behavior of the different diamond cuts, using both boxplots and violin plots. Compare the resulting graphs. For this analysis, which is better?

ggplot(diamonds, aes(x = cut, y = price)) +  
 geom\_boxplot() +  
 geom\_violin()



ggplot(diamonds, aes(x = cut, y = price)) +  
 geom\_violin()



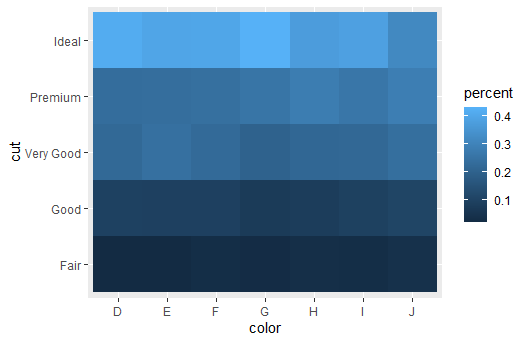
## Violin plot is more helpful since you can visualize the data and see the distribution better. As you can see in the overlapping graph, the boxplot causes a lot of information to be missed.

## 7.5.2.1 Exercises

### (15) 7.5.2.1 Exercise 1 (10 pts)

How could you rescale the count dataset above to more clearly show the distribution of cut within color, or color within cut?

diamonds %>%  
 count(color, cut) %>%  
 group\_by(color) %>%  
 mutate(percent = n / sum(n)) %>%  
 ggplot(mapping = aes(x = color, y = cut)) +  
 geom\_tile(mapping = aes(fill = percent))

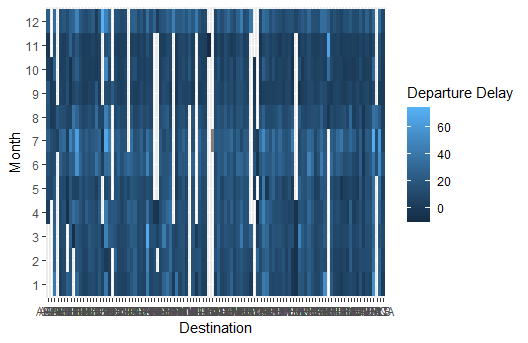


## Percent represents the distributions in this tile plot

### (16) 7.5.2.1 Exercise 2 (10 pts)

Use geom\_tile() together with dplyr to explore how average flight delays vary by destination and month of year. What makes the plot difficult to read? How could you improve it?

flights %>%  
 group\_by(month, dest) %>%  
 summarise(dep\_delay = mean(dep\_delay, na.rm = TRUE)) %>%  
 ggplot(aes(x = dest, y = factor(month), fill = dep\_delay)) +  
 geom\_tile() +  
 labs(x = "Destination" , y = "Month", fill = "Departure Delay")



# I'm unsure how to improve it. Maybe taking out missing values? the main problem is the amount of destinations.

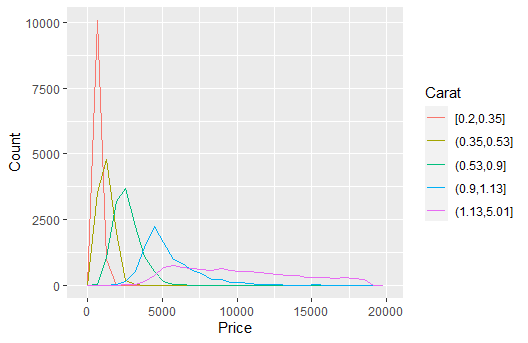
## 7.5.3.1 Exercises

### (17) 7.5.3.1 Exercise 1 (10 pts)

Instead of summarizing the conditional distribution with a boxplot, you could use a frequency polygon. What do you need to consider when using cut\_width() vs cut\_number()? How does that impact a visualization of the 2D distribution of carat and price?

ggplot(diamonds, mapping = aes(color = cut\_number(carat, 5), x = price)) +  
 geom\_freqpoly() +  
 labs(x = "Price", y = "Count", color = "Carat")

`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

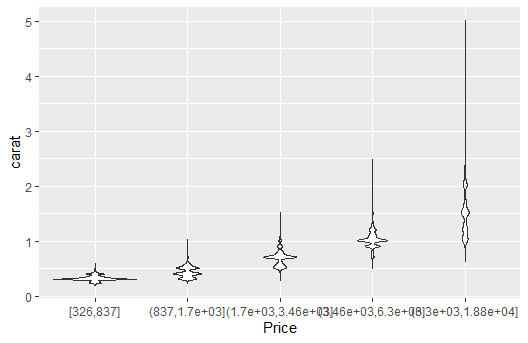


## cut\_number is split into sections based on the number of groups specified, where cut\_width separates based on the continuous variable.

### (18) 7.5.3.1 Exercise 2 (10 pts)

Visualize the distribution of carat, partitioned by price.

ggplot(diamonds, aes(x = cut\_number(price, 5), y = carat)) +  
 geom\_violin() +  
 xlab("Price")



### (19) 7.5.3.1 Exercise 3 (5 pts)

How does the price distribution of very large diamonds compare to small diamonds. Is it as you expect, or does it surprise you?

## As Carat increases the price increases and it is not proper distribution since the higher carat diamonds are more rare. This means less data points leading to less obvious distributions. Also some diamonds are just excellently cut, meaning smaller carat diamonds will skew the distributions.