Computing Club Presentation

Safiya Sirota

October 15, 2024

Introduction

Now that we've explored some basics with Lupe, we can see how it works in R Markdown. If you haven't before, you will want to install `tinytex::install_tinytex()` in order for RMarkdown documents to be rendered as .pdfs using LaTeX.

Let's pretend to do some work on the `tidyverse` dataset `diamonds`. This was we can also show how to incorporate nice tables and plots into our RMarkdown document.

Formatting

- When working in an R Markdown file, there are multiple ways to produce the same formatting. For example, I use section{} above, but I could also use a hashtag, like I did to make the introduction header. Similarly subsection{} will correlate to a double hashtag, and so on.

- There are also multiple way to italicize and bold your text. You can either do *this*, or you can use the *R Markdown* way. And for boldface, you can use the **LaTeX** way or the **R Markdown** way.
- Finally, there are 2 ways to bullet point. One if by using the asterisk, and the other is by using an `\itemize{}` environment, as is done above.

Just be careful when trying to use R Markdown conventional formatting *inside* LaTeX environments! This can lead to issues you may have to troubleshoot.

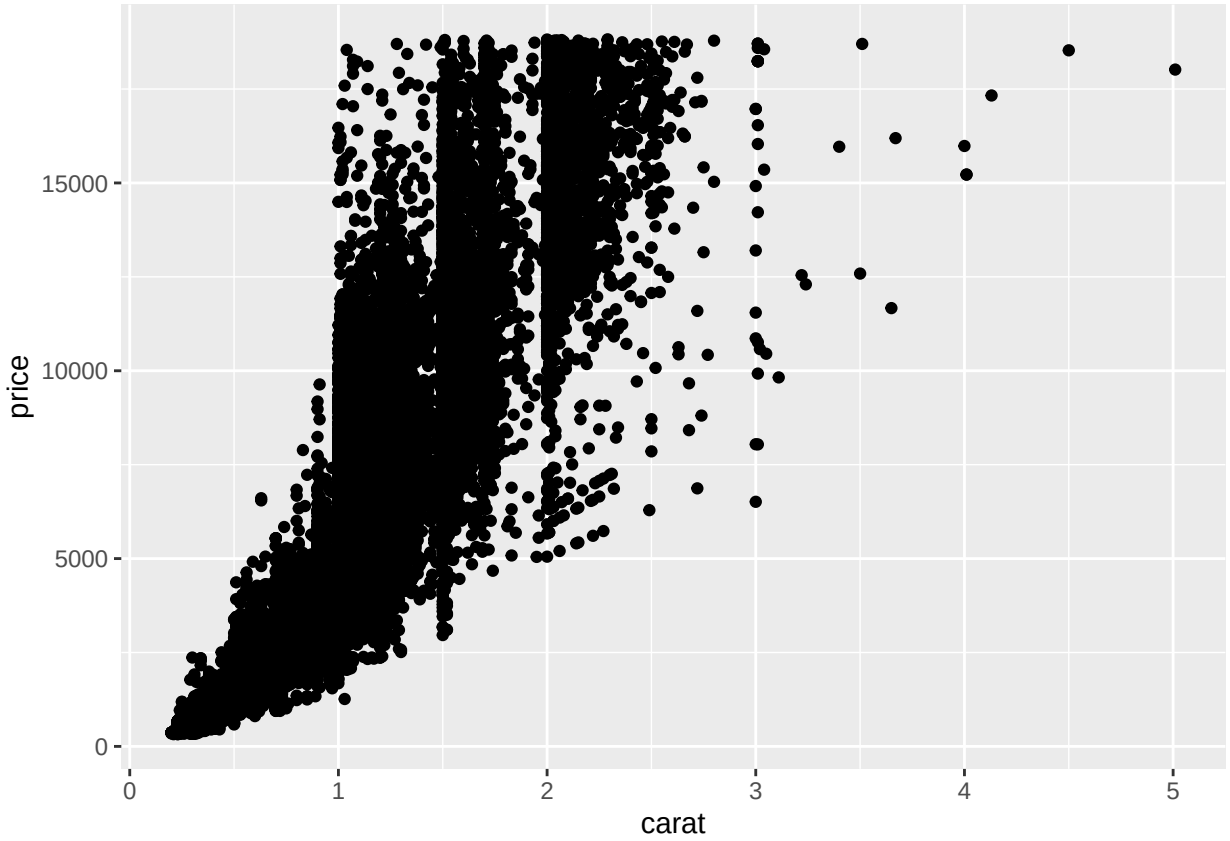
Tables and Plots

Plots

```
library(tidyverse)
library(gtsummary)
```

Including plots into your homework is easy enough and you may already know how to do it. One way is to simply just print the plot out from an R Markdown chunk.

```
diamonds %>%
  ggplot() +
  geom_point(aes(x = carat, y = price))
```



However, you may not know that you can format your plot using LaTeX-ish commands in the code chunk header...

```
diamonds %>%
  ggplot() +
  geom_point(aes(x = carat, y = price))
```

And if I take a screenshot of my plot, or have a different picture. I can include it using this technique as well

And there's even one more technique!

```
library(knitr)
include_graphics("plot_pic.png")
```

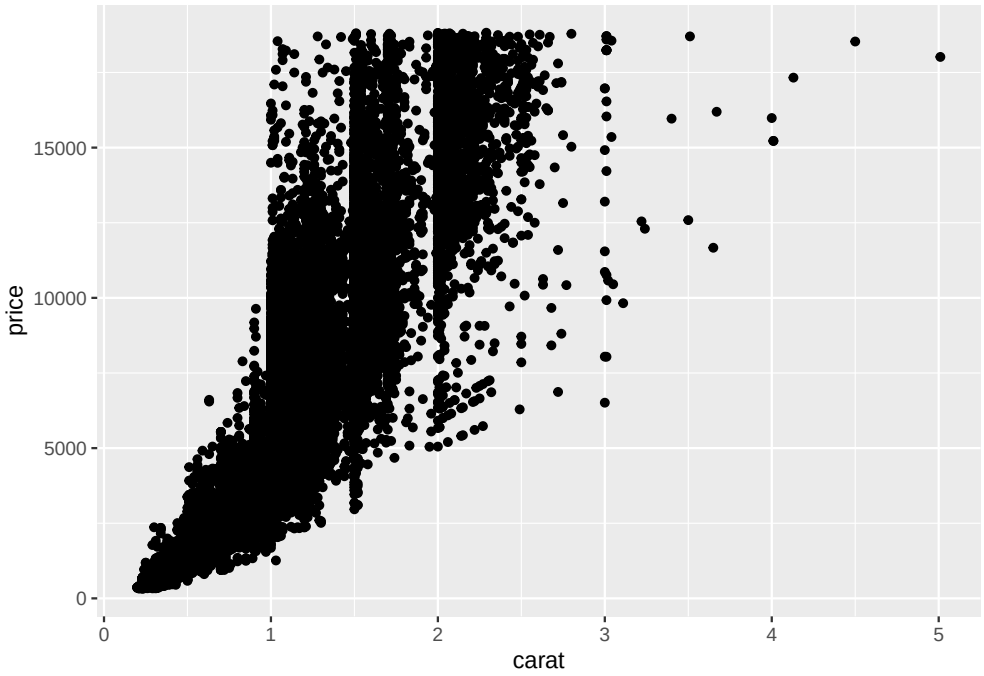


Figure 1: Diamonds by carat and price

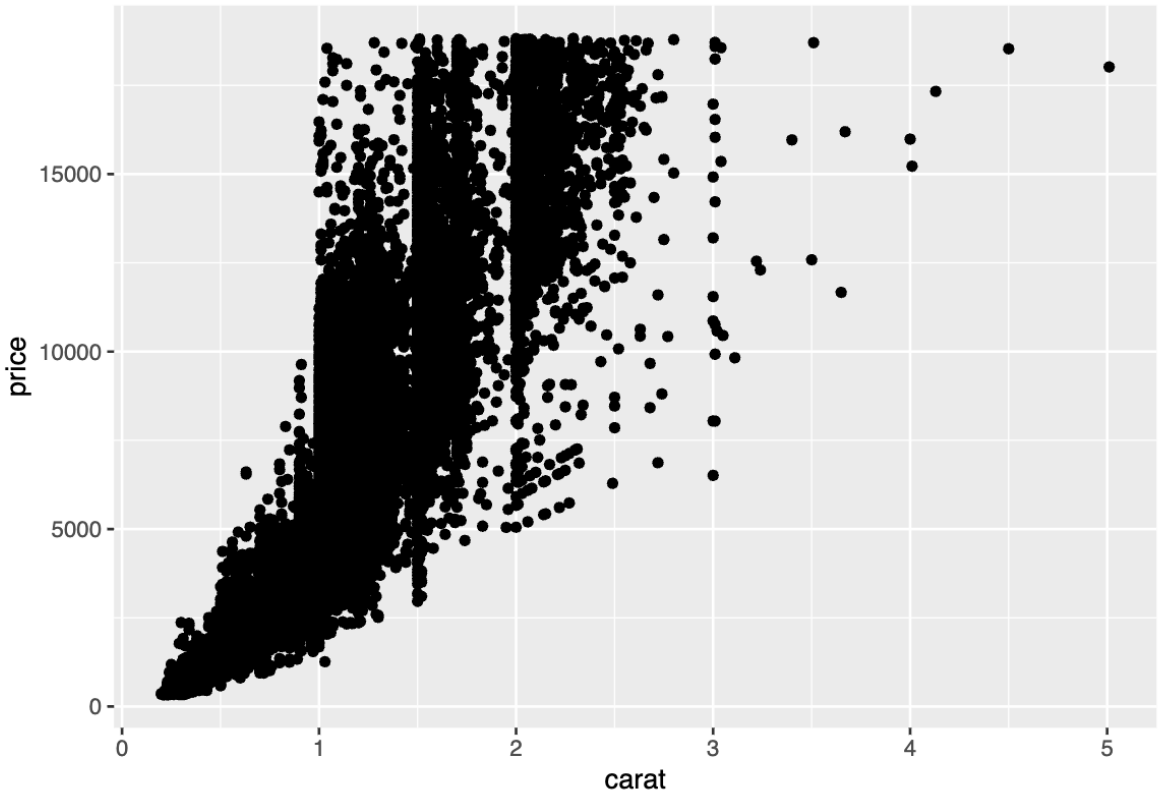
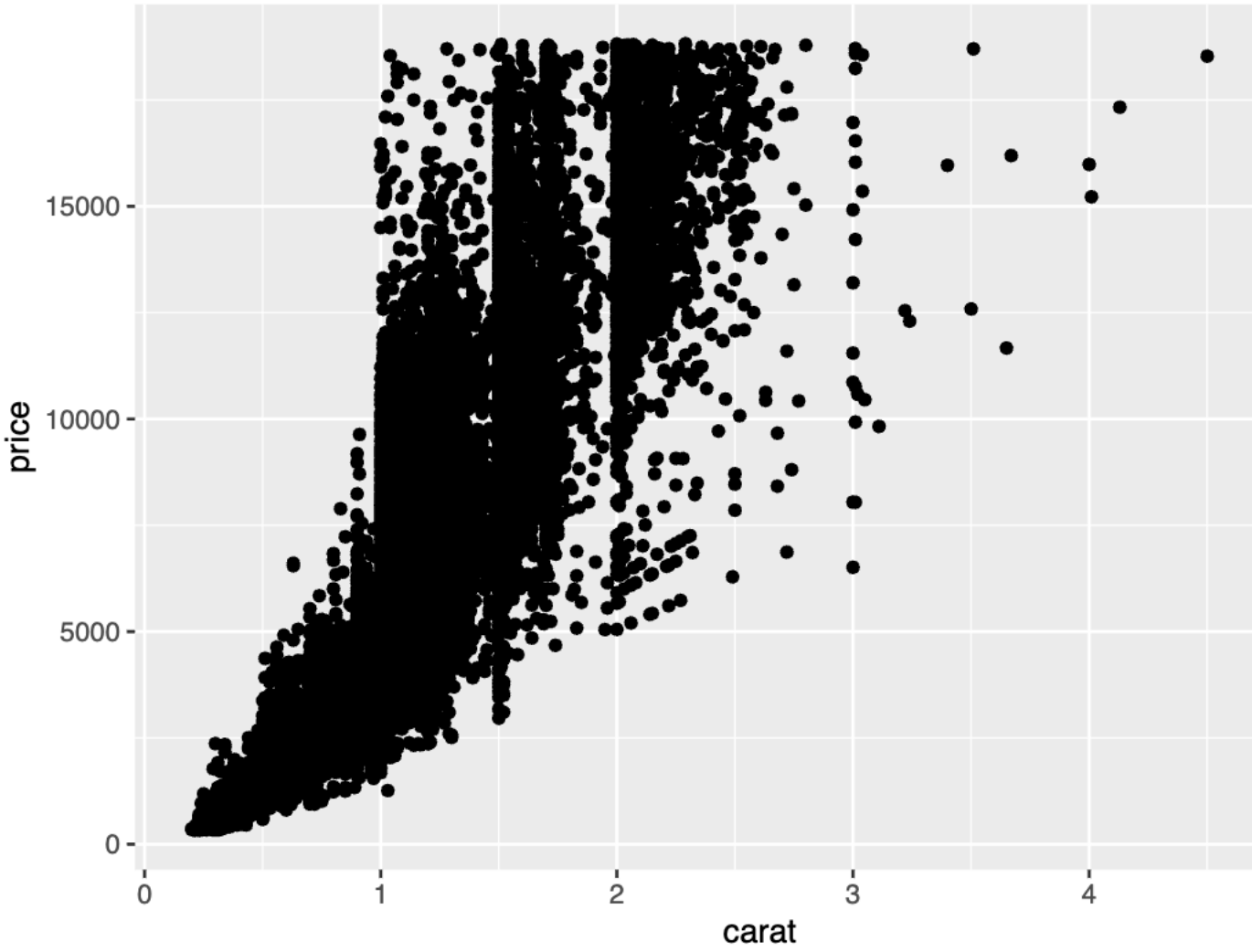


Figure 2: Diamonds by carat and price



Tables

```
diamonds %>% head() %>% gt::gt()
```

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48

```r
dim(diamonds)
```

```
## [1] 53940    10
```

```r
diamonds_data <- diamonds %>%
  mutate(
    quality2 = case_when(
      cut %in% c("Fair", "Good") & carat < 3 ~ "okay",
      cut %in% c("Good", "Very Good") & carat > 3 & carat <= 4.5 ~ "good",
      cut %in% c("Very Good", "Premium", "Ideal") & carat > 4.5 ~ "great",
      TRUE ~ "neither"
    )
  )
diamonds_data %>% head() %>% gt::gt()
```

carat	cut	color	clarity	depth	table	price	x	y	z	quality2
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43	neither
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31	neither
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31	okay
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63	neither
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75	okay
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48	neither

group_by() and summarise()

```r
diamonds %>%
  group_by(clarity) %>% # group_by has to come before summarise to get group-level summary statistics
  summarise(
    mean_price = mean(price),
    count = n()
  )  %>% # I can create +1 summary statistics at a time
  gt::gt()
```

clarity	mean_price	count
I1	3924.169	741
SI2	5063.029	9194
SI1	3996.001	13065
VS2	3924.989	12258
VS1	3839.455	8171
VVS2	3283.737	5066
VVS1	2523.115	3655
IF	2864.839	1790

gtsummary

Here I am making a table one using `gtsummary`'s `tbl_summary` function. I am grouping by cut and adding an overal column and p values. The bolded labels are just to make it look pretty.

```r
diamonds %>%
  tbl_summary(by = cut) %>% # calling the table one summary compand
  add_overall() %>% # adding overall column
  bold_labels() %>% # making labels look nice
  add_p() # adding p values
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**Overall**, N = 53,940	**Fair**, N = 1,610	**Good**, N = 4,906	**Very Good**, N = 12,082	**Premium**, N = 13,791	**Ideal**, N = 21,551	**p-value**
carat	0.70 (0.40, 1.04)	1.00 (0.70, 1.20)	0.82 (0.50, 1.01)	0.71 (0.41, 1.02)	0.86 (0.41, 1.20)	0.54 (0.35, 1.01)	<0.001
color							<0.001
D	6,775 (13%)	163 (10%)	662 (13%)	1,513 (13%)	1,603 (12%)	2,834 (13%)	
E	9,797 (18%)	224 (14%)	933 (19%)	2,400 (20%)	2,337 (17%)	3,903 (18%)	
F	9,542 (18%)	312 (19%)	909 (19%)	2,164 (18%)	2,331 (17%)	3,826 (18%)	
G	11,292 (21%)	314 (20%)	871 (18%)	2,299 (19%)	2,924 (21%)	4,884 (23%)	
H	8,304 (15%)	303 (19%)	702 (14%)	1,824 (15%)	2,360 (17%)	3,115 (14%)	
I	5,422 (10%)	175 (11%)	522 (11%)	1,204 (10.0%)	1,428 (10%)	2,093 (9.7%)	
J	2,808 (5.2%)	119 (7.4%)	307 (6.3%)	678 (5.6%)	808 (5.9%)	896 (4.2%)	
clarity							<0.001
I1	741 (1.4%)	210 (13%)	96 (2.0%)	84 (0.7%)	205 (1.5%)	146 (0.7%)	
SI2	9,194 (17%)	466 (29%)	1,081 (22%)	2,100 (17%)	2,949 (21%)	2,598 (12%)	
SI1	13,065 (24%)	408 (25%)	1,560 (32%)	3,240 (27%)	3,575 (26%)	4,282 (20%)	
VS2	12,258 (23%)	261 (16%)	978 (20%)	2,591 (21%)	3,357 (24%)	5,071 (24%)	
VS1	8,171 (15%)	170 (11%)	648 (13%)	1,775 (15%)	1,989 (14%)	3,589 (17%)	
VVS2	5,066 (9.4%)	69 (4.3%)	286 (5.8%)	1,235 (10%)	870 (6.3%)	2,606 (12%)	
VVS1	3,655 (6.8%)	17 (1.1%)	186 (3.8%)	789 (6.5%)	616 (4.5%)	2,047 (9.5%)	
IF	1,790 (3.3%)	9 (0.6%)	71 (1.4%)	268 (2.2%)	230 (1.7%)	1,212 (5.6%)	
depth	61.80 (61.00, 62.50)	65.00 (64.40, 65.90)	63.40 (61.30, 63.80)	62.10 (60.90, 62.90)	61.40 (60.50, 62.20)	61.80 (61.30, 62.20)	<0.001
table	57.00 (56.00, 59.00)	58.00 (56.00, 61.00)	58.00 (56.00, 61.00)	58.00 (56.00, 59.00)	59.00 (58.00, 60.00)	56.00 (55.00, 57.00)	<0.001

Characteristic	Overall, N = 53,940	Fair, N = 1,610	Good, N = 4,906	Very Good, N = 12,082	Premium, N = 13,791	Ideal, N = 21,551	p-value
price	2,401 (950, 5,324)	3,282 (2,050, 5,206)	3,051 (1,145, 5,028)	2,648 (912, 5,373)	3,185 (1,046, 6,296)	1,810 (878, 4,679)	<0.001
x	5.70 (4.71, 6.54)	6.18 (5.63, 6.70)	5.98 (5.02, 6.42)	5.74 (4.75, 6.47)	6.11 (4.80, 6.80)	5.25 (4.54, 6.44)	<0.001
y	5.71 (4.72, 6.54)	6.10 (5.57, 6.64)	5.99 (5.02, 6.44)	5.77 (4.77, 6.51)	6.06 (4.79, 6.76)	5.26 (4.55, 6.45)	<0.001
z	3.53 (2.91, 4.04)	3.97 (3.61, 4.28)	3.70 (3.07, 4.03)	3.56 (2.95, 4.02)	3.72 (2.94, 4.16)	3.23 (2.80, 3.98)	<0.001

This is a more complex gt summary example. Here we are controlling the type of statistic variables that are going to be displayed.

```
diamonds %>%
  tbl_summary(by = cut,
              type = all_continuous() ~ "continuous2",
              statistic = list( # we need to specify the stats by the type of variable
                all_continuous() ~ c("{mean} ({sd})",
                                     "{median} ({p25}, {p75})",
                                     "{min}, {max}"),
                all_categorical() ~ "{n} / {N} ({p}%)"
              )) %>%
  add_overall() %>%
  bold_labels() %>%
  add_p()
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	Overall, N = 53,940	Fair, N = 1,610	Good, N = 4,906	Very Good, N = 12,082	Premium, N = 13,791	Ideal, N = 21,551	p-value
carat							<0.001
Mean (SD)	0.80 (0.47)	1.05 (0.52)	0.85 (0.45)	0.81 (0.46)	0.89 (0.52)	0.70 (0.43)	
Median (IQR)	0.70 (0.40, 1.04)	1.00 (0.70, 1.20)	0.82 (0.50, 1.01)	0.71 (0.41, 1.02)	0.86 (0.41, 1.20)	0.54 (0.35, 1.01)	
Range	0.20, 5.01	0.22, 5.01	0.23, 3.01	0.20, 4.00	0.20, 4.01	0.20, 3.50	
color							<0.001
D	6,775 / 53,940 (13%)	163 / 1,610 (10%)	662 / 4,906 (13%)	1,513 / 12,082 (13%)	1,603 / 13,791 (12%)	2,834 / 21,551 (13%)	
E	9,797 / 53,940 (18%)	224 / 1,610 (14%)	933 / 4,906 (19%)	2,400 / 12,082 (20%)	2,337 / 13,791 (17%)	3,903 / 21,551 (18%)	
F	9,542 / 53,940 (18%)	312 / 1,610 (19%)	909 / 4,906 (19%)	2,164 / 12,082 (18%)	2,331 / 13,791 (17%)	3,826 / 21,551 (18%)	

Characteristic	**Overall**, N = 53,940	**Fair**, N = 1,610	**Good**, N = 4,906	**Very Good**, N = 12,082	**Premium**, N = 13,791	**Ideal**, N = 21,551	**p-value**
G	11,292 / 53,940 (21%)	314 / 1,610 (20%)	871 / 4,906 (18%)	2,299 / 12,082 (19%)	2,924 / 13,791 (21%)	4,884 / 21,551 (23%)	
H	8,304 / 53,940 (15%)	303 / 1,610 (19%)	702 / 4,906 (14%)	1,824 / 12,082 (15%)	2,360 / 13,791 (17%)	3,115 / 21,551 (14%)	
I	5,422 / 53,940 (10%)	175 / 1,610 (11%)	522 / 4,906 (11%)	1,204 / 12,082 (10.0%)	1,428 / 13,791 (10%)	2,093 / 21,551 (9.7%)	
J	2,808 / 53,940 (5.2%)	119 / 1,610 (7.4%)	307 / 4,906 (6.3%)	678 / 12,082 (5.6%)	808 / 13,791 (5.9%)	896 / 21,551 (4.2%)	
clarity							<0.001
I1	741 / 53,940 (1.4%)	210 / 1,610 (13%)	96 / 4,906 (2.0%)	84 / 12,082 (0.7%)	205 / 13,791 (1.5%)	146 / 21,551 (0.7%)	
SI2	9,194 / 53,940 (17%)	466 / 1,610 (29%)	1,081 / 4,906 (22%)	2,100 / 12,082 (17%)	2,949 / 13,791 (21%)	2,598 / 21,551 (12%)	
SI1	13,065 / 53,940 (24%)	408 / 1,610 (25%)	1,560 / 4,906 (32%)	3,240 / 12,082 (27%)	3,575 / 13,791 (26%)	4,282 / 21,551 (20%)	
VS2	12,258 / 53,940 (23%)	261 / 1,610 (16%)	978 / 4,906 (20%)	2,591 / 12,082 (21%)	3,357 / 13,791 (24%)	5,071 / 21,551 (24%)	
VS1	8,171 / 53,940 (15%)	170 / 1,610 (11%)	648 / 4,906 (13%)	1,775 / 12,082 (15%)	1,989 / 13,791 (14%)	3,589 / 21,551 (17%)	
VVS2	5,066 / 53,940 (9.4%)	69 / 1,610 (4.3%)	286 / 4,906 (5.8%)	1,235 / 12,082 (10%)	870 / 13,791 (6.3%)	2,606 / 21,551 (12%)	
VVS1	3,655 / 53,940 (6.8%)	17 / 1,610 (1.1%)	186 / 4,906 (3.8%)	789 / 12,082 (6.5%)	616 / 13,791 (4.5%)	2,047 / 21,551 (9.5%)	
IF	1,790 / 53,940 (3.3%)	9 / 1,610 (0.6%)	71 / 4,906 (1.4%)	268 / 12,082 (2.2%)	230 / 13,791 (1.7%)	1,212 / 21,551 (5.6%)	
depth							<0.001
Mean (SD)	61.75 (1.43)	64.04 (3.64)	62.37 (2.17)	61.82 (1.38)	61.26 (1.16)	61.71 (0.72)	
Median (IQR)	61.80 (61.00, 62.50)	65.00 (64.40, 65.90)	63.40 (61.30, 63.80)	62.10 (60.90, 62.90)	61.40 (60.50, 62.20)	61.80 (61.30, 62.20)	
Range	43.00, 79.00	43.00, 79.00	54.30, 67.00	56.80, 64.90	58.00, 63.00	43.00, 66.70	
table							<0.001
Mean (SD)	57.46 (2.23)	59.05 (3.95)	58.69 (2.85)	57.96 (2.12)	58.75 (1.48)	55.95 (1.25)	
Median (IQR)	57.00 (56.00, 59.00)	58.00 (56.00, 61.00)	58.00 (56.00, 61.00)	58.00 (56.00, 59.00)	59.00 (58.00, 60.00)	56.00 (55.00, 57.00)	

Characteristic	**Overall**, N = 53,940	**Fair**, N = 1,610	**Good**, N = 4,906	**Very Good**, N = 12,082	**Premium**, N = 13,791	**Ideal**, N = 21,551	**p-value**
Range	43.00, 95.00	49.00, 95.00	51.00, 66.00	44.00, 66.00	51.00, 62.00	43.00, 63.00	
price							<0.001
Mean (SD)	3,933 (3,989)	4,359 (3,560)	3,929 (3,682)	3,982 (3,936)	4,584 (4,349)	3,458 (3,808)	
Median (IQR)	2,401 (950, 5,324)	3,282 (2,050, 5,206)	3,051 (1,145, 5,028)	2,648 (912, 5,373)	3,185 (1,046, 6,296)	1,810 (878, 4,679)	
Range	326, 18,823	337, 18,574	327, 18,788	336, 18,818	326, 18,823	326, 18,806	
x							<0.001
Mean (SD)	5.73 (1.12)	6.25 (0.96)	5.84 (1.06)	5.74 (1.10)	5.97 (1.19)	5.51 (1.06)	
Median (IQR)	5.70 (4.71, 6.54)	6.18 (5.63, 6.70)	5.98 (5.02, 6.42)	5.74 (4.75, 6.47)	6.11 (4.80, 6.80)	5.25 (4.54, 6.44)	
Range	0.00, 10.74	0.00, 10.74	0.00, 9.44	0.00, 10.01	0.00, 10.14	0.00, 9.65	
y							<0.001
Mean (SD)	5.73 (1.14)	6.18 (0.96)	5.85 (1.05)	5.77 (1.10)	5.94 (1.26)	5.52 (1.07)	
Median (IQR)	5.71 (4.72, 6.54)	6.10 (5.57, 6.64)	5.99 (5.02, 6.44)	5.77 (4.77, 6.51)	6.06 (4.79, 6.76)	5.26 (4.55, 6.45)	
Range	0.00, 58.90	0.00, 10.54	0.00, 9.38	0.00, 9.94	0.00, 58.90	0.00, 31.80	
z							<0.001
Mean (SD)	3.54 (0.71)	3.98 (0.65)	3.64 (0.65)	3.56 (0.73)	3.65 (0.73)	3.40 (0.66)	
Median (IQR)	3.53 (2.91, 4.04)	3.97 (3.61, 4.28)	3.70 (3.07, 4.03)	3.56 (2.95, 4.02)	3.72 (2.94, 4.16)	3.23 (2.80, 3.98)	
Range	0.00, 31.80	0.00, 6.98	0.00, 5.79	0.00, 31.80	0.00, 8.06	0.00, 6.03	