R Bootcamp

Day 1 Exercises — ggplot

September 14, 2015

Create a new RMarkdown document in RStudio (File -> New -> RMarkdown -> Document -> HTML)

Create a code chunk to load the `ggplot2` package (via `library(ggplot2)`). Use the `echo` argument to the code chunk to avoid printing the output of this unexciting command. (Hint: this is quite similar to the `eval` argument that appears in the template document.)

For all of the following exercises, write a brief description of what you intend to do, show the code used to do it and the output generated, and and write a brief description of your interpretation.

There is a dataset on the price and attributes of diamonds that comes with the ggplot2 package. Examine the `diamonds` data.frame using `str`, `summary`, `head`, and/or `View`. What relationships might be worth exploring graphically?

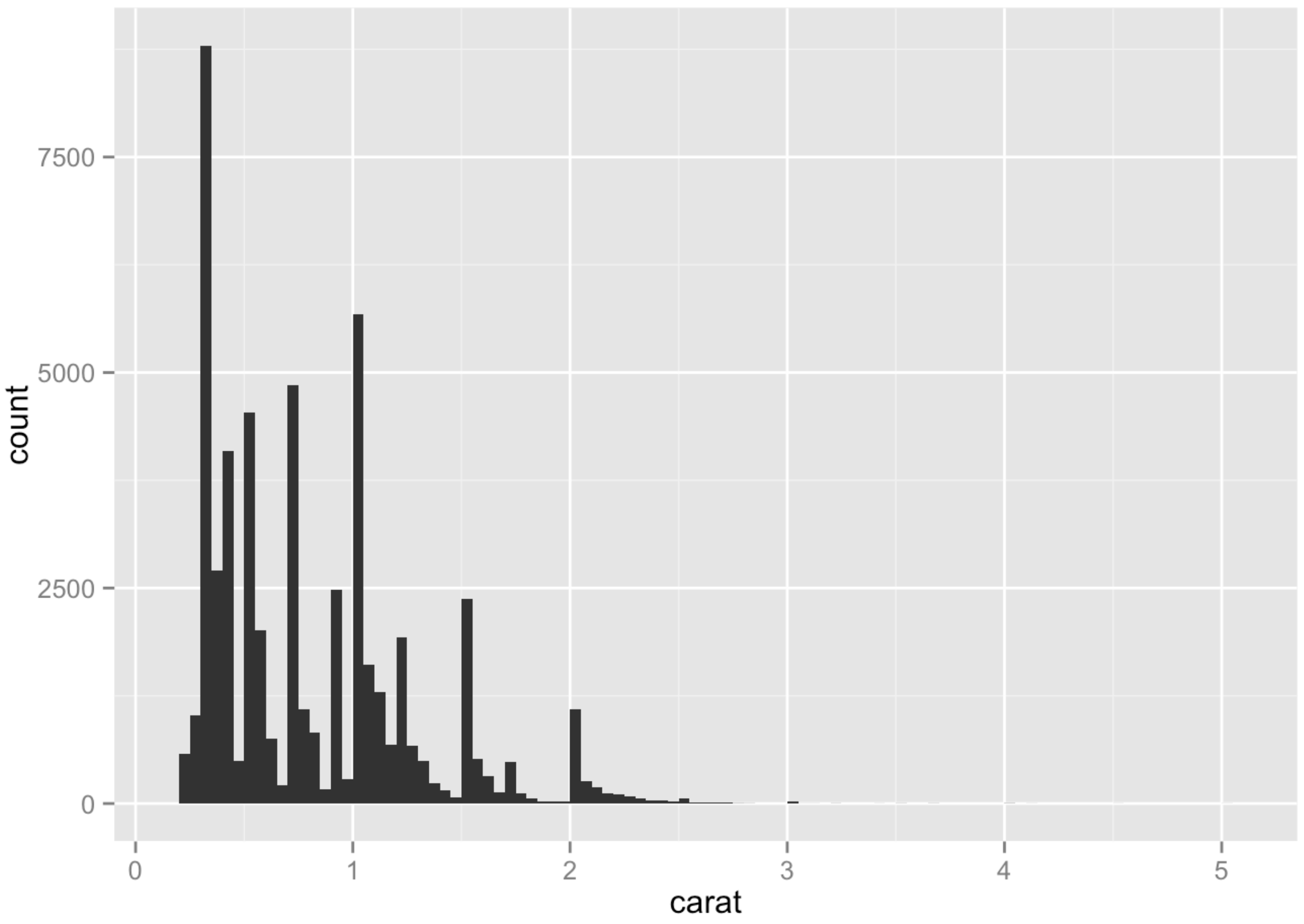
```
str(diamonds)
```

```
## 'data.frame':    53940 obs. of  10 variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

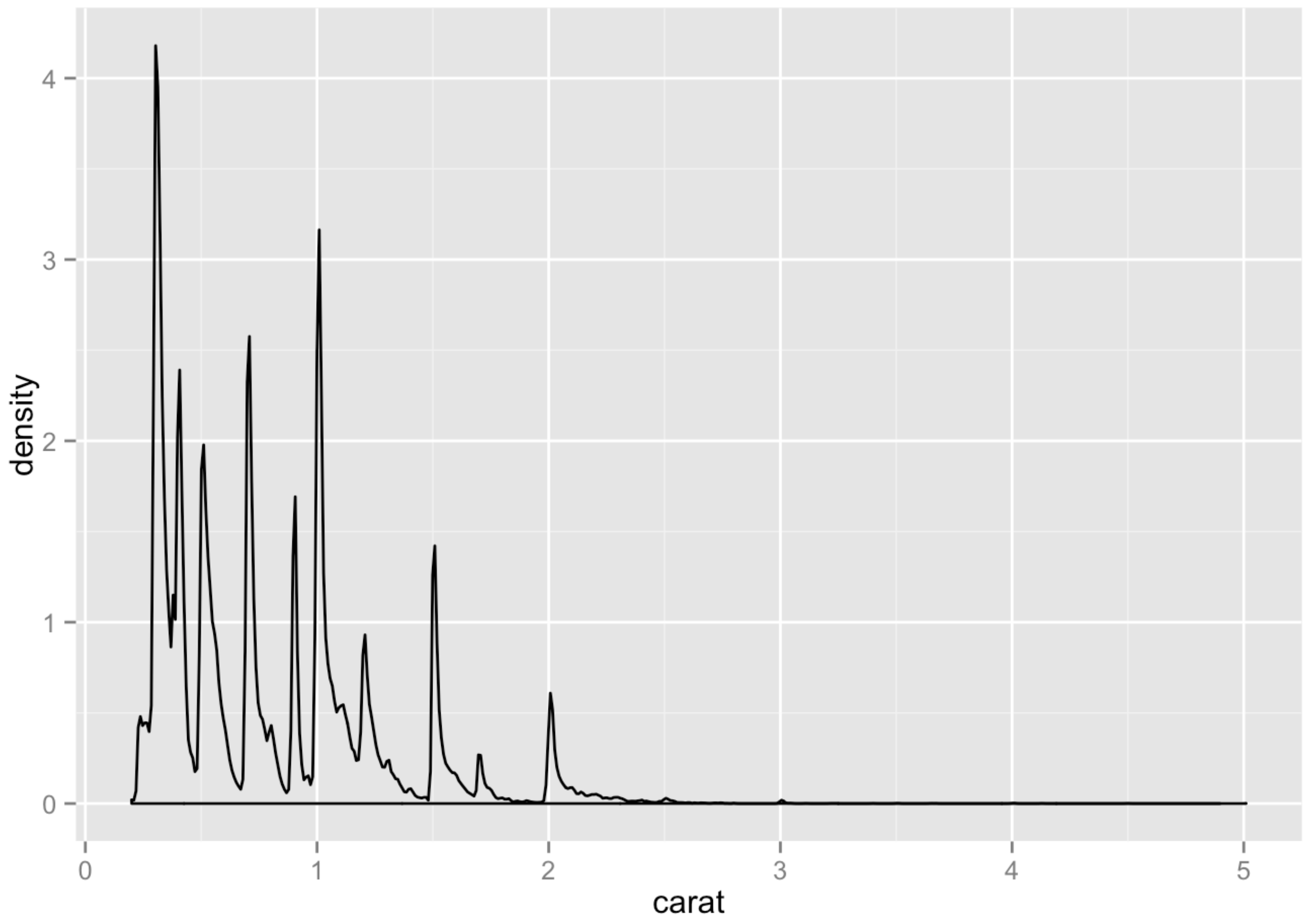
1. Examine the distribution of diamond size (`carat`) via
   a. a histogram
   b. density curve

Be sure to adjust the bin (via `binwidth`) or kernal (via `adjust`) to capture any interesting patterns.

```
ggplot(diamonds, aes(x = carat)) +
    geom_histogram(binwidth = .05)
```



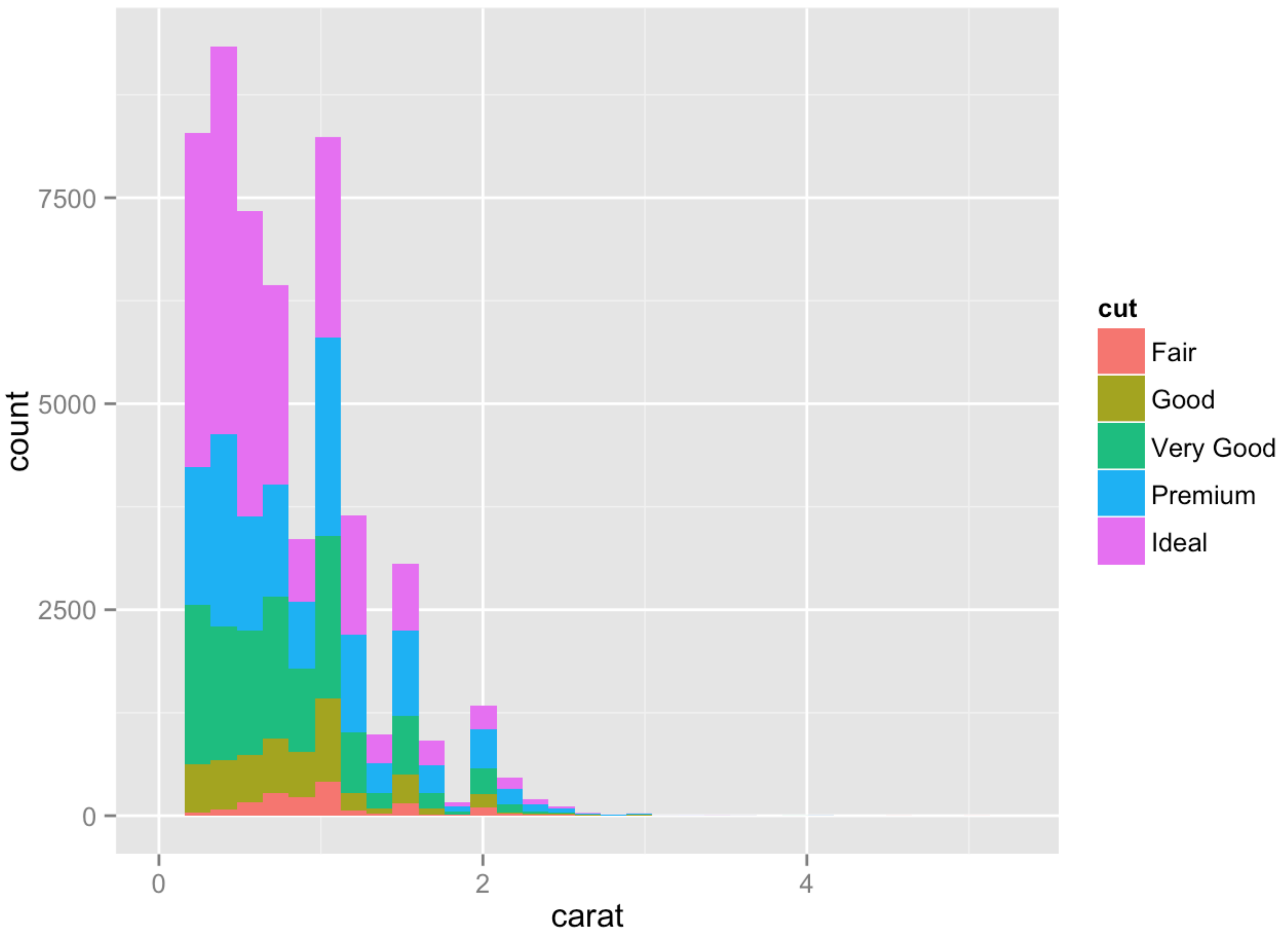
```
ggplot(diamonds, aes(x = carat)) +
    geom_density(adjust = .1)
```



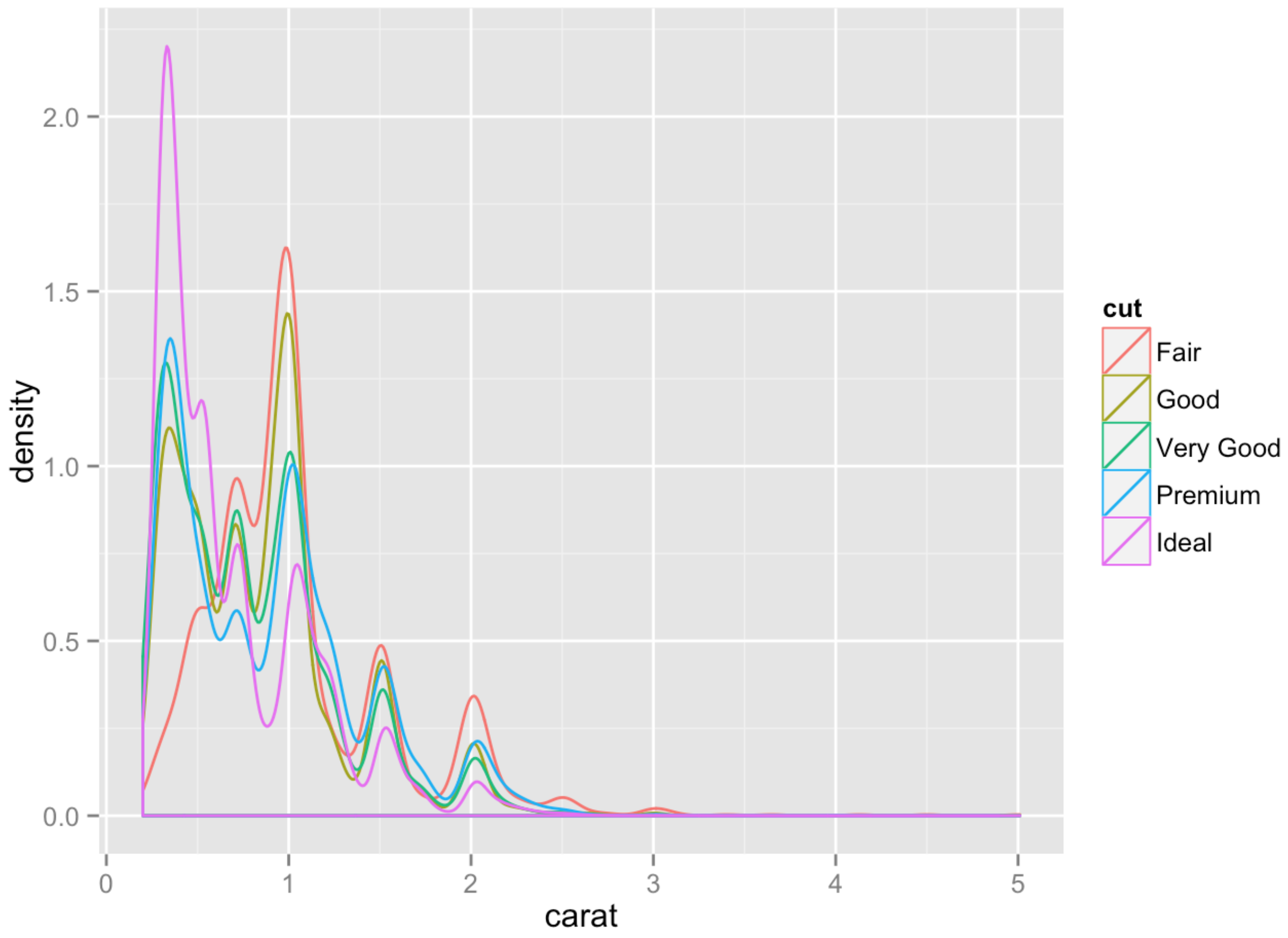
2. a. It appears there are several qualities of diamond `cut` in this dataset (Can you tell how many from `str(diamonds)` . Create a histogram and density curve to explore the distribution of sizes across the different cut qualities. You could separate cuts by color or fill or facet (or `linetype` , or ???).

```
ggplot(diamonds, aes(x = carat, fill = cut)) +
    geom_histogram()
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

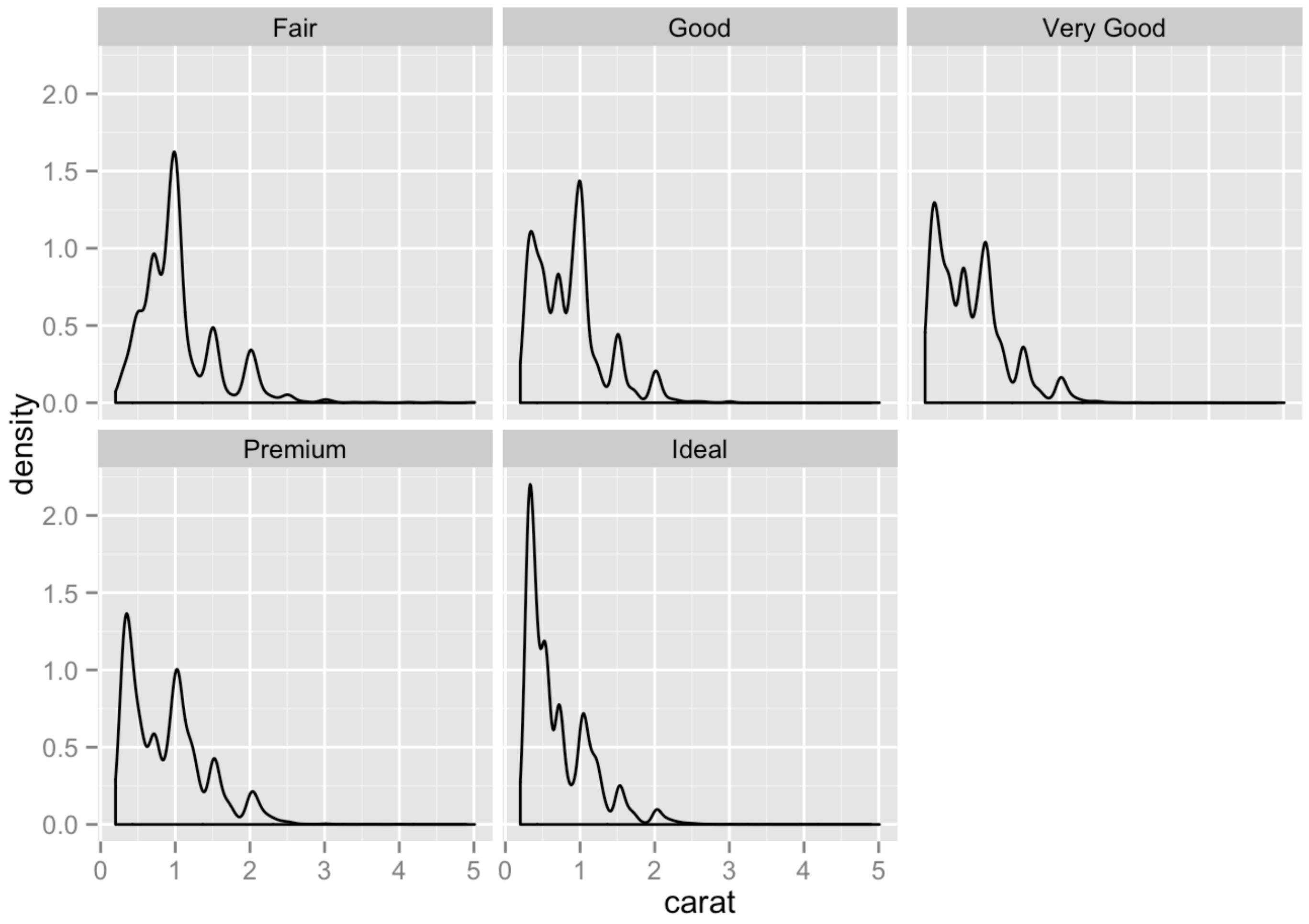


```
ggplot(diamonds, aes(x = carat, color = cut)) +
    geom_density()
```



```
# Note that you can get rid of the line(s) across the bottom of density plot like
so:
# geom_line(stat = "density")

## or ##
ggplot(diamonds, aes(x = carat)) +
    geom_density() +
    facet_wrap(~cut)
```

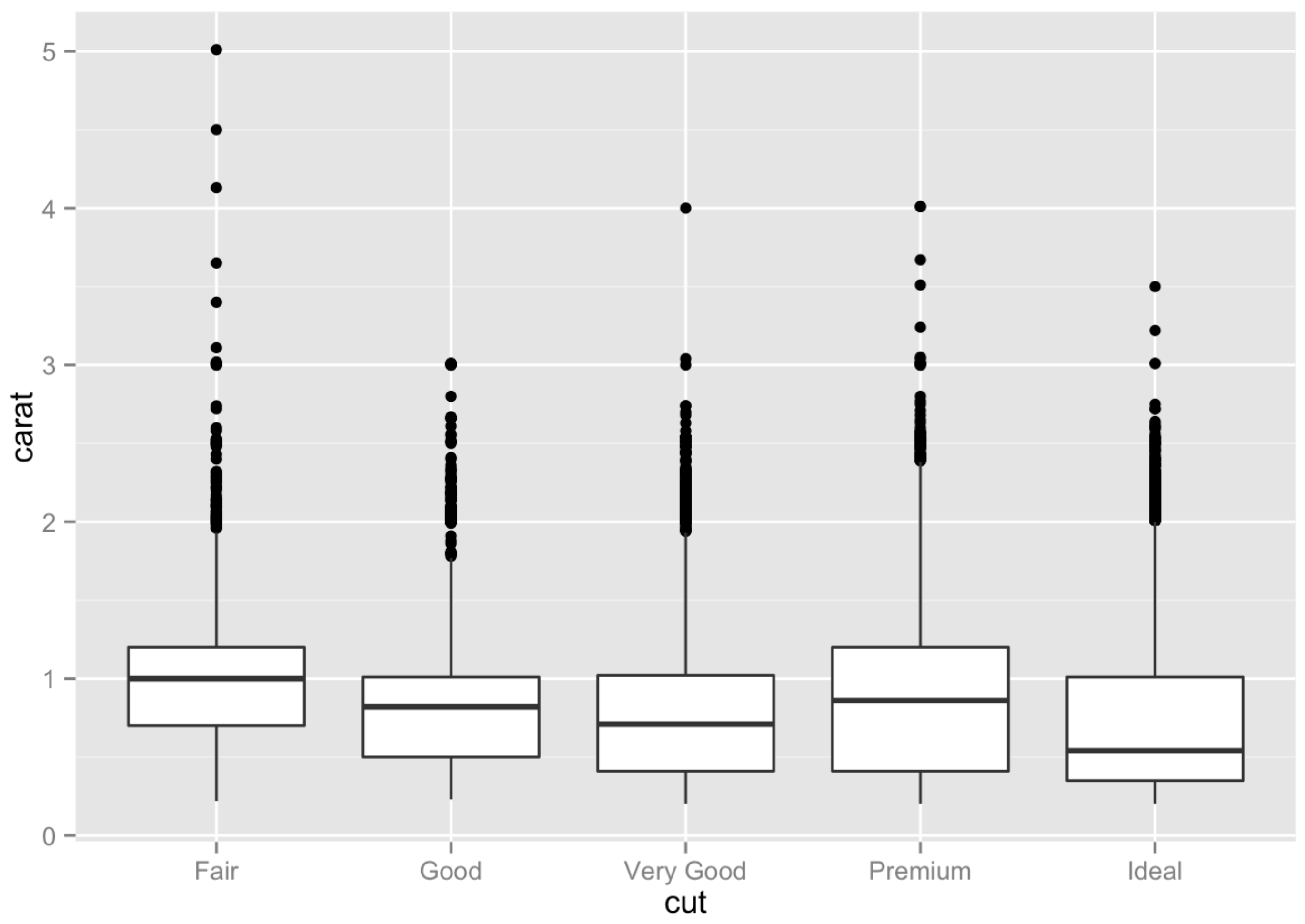


```
b. Which of the above plots would be more appropriate if we were interested in abs
olute quantity of diamonds of each cut, and which would be more appropriate if we
were interested in the relative distribution of sizes within each cut? Why?
```

```
# The density curves are better for relative comparisons because the area under ea
ch is normalized
# (so information about the number of diamonds within each cut-class is lost).
# The histograms retain this information and so are preferable for comparisons of
total quantity within each cut-class.
```

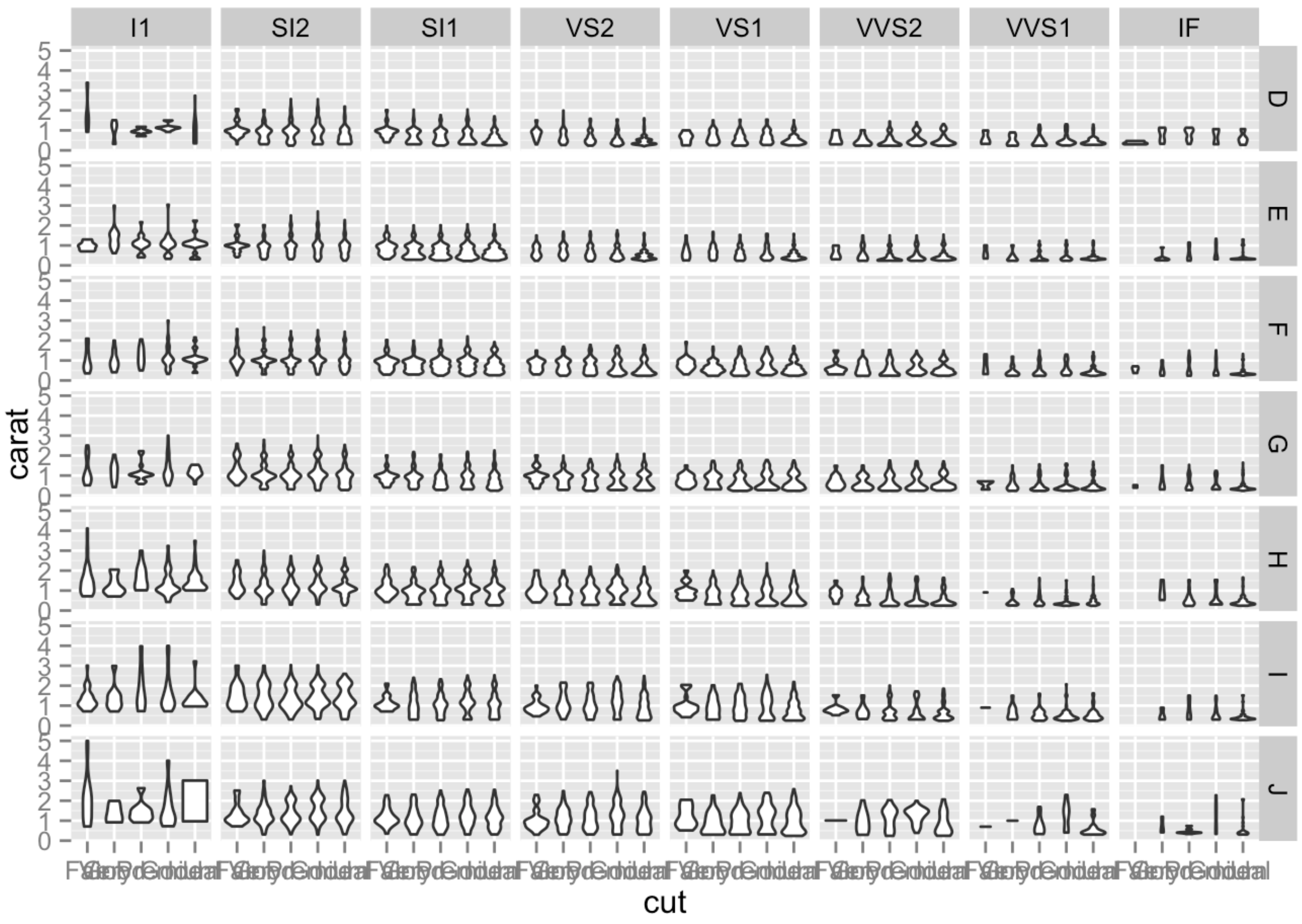
3. Create a boxplot that compares the distribution of sizes within each `cut`.

```
ggplot(diamonds, aes(x = cut, y = carat)) +
    geom_boxplot()
```



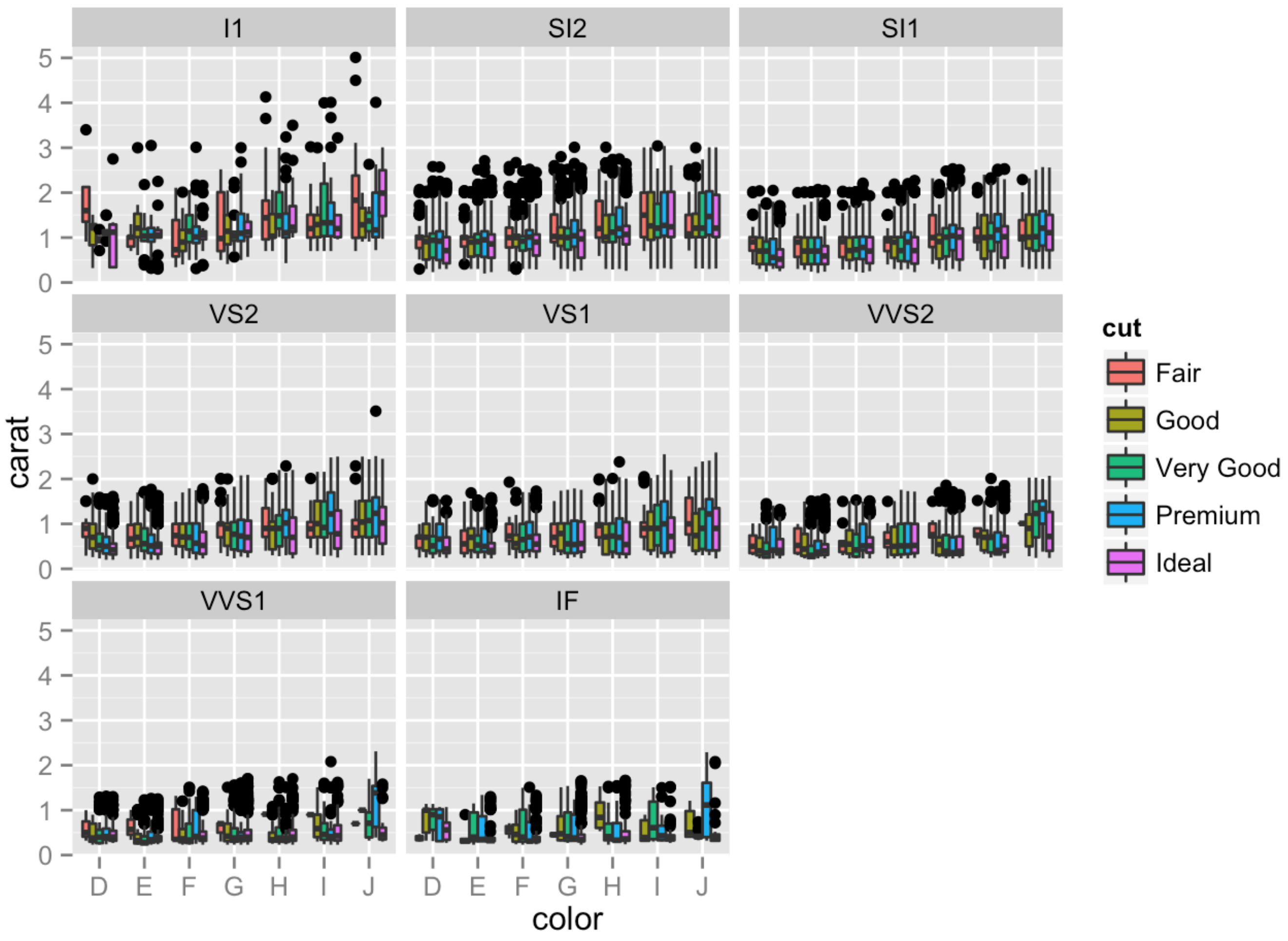
4. Change the above plot to be a violin plot. Add a single line to facet the plots by both color and clarity. Don't worry about the legibility of the resulting plot.

```
ggplot(diamonds, aes(x = cut, y = carat)) +
    geom_violin() +
    facet_grid(color ~ clarity)
```



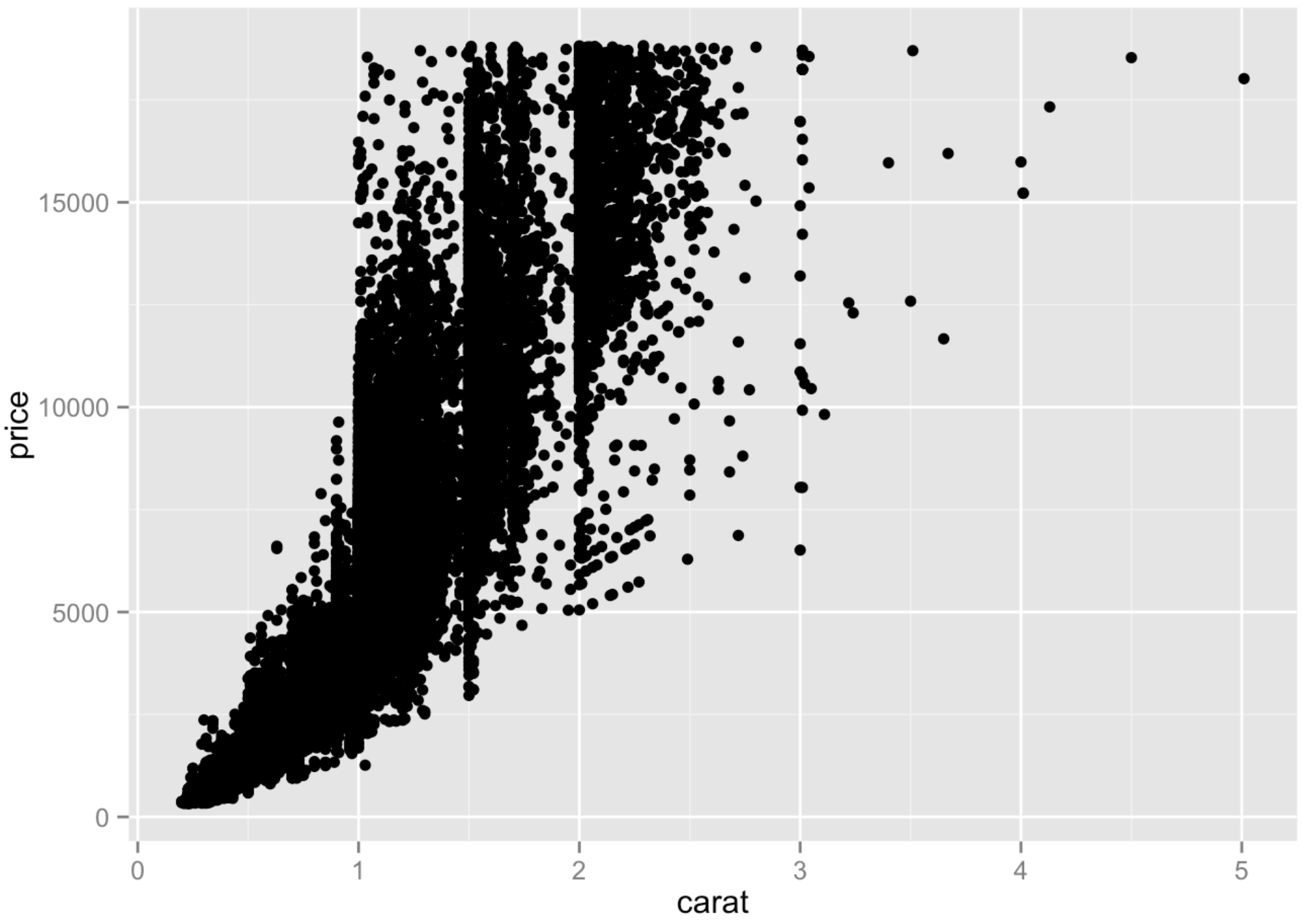
5. Let’s try to get all that information in a more comprehensible plot. Leave `carat` on the y-axis, and experiment with mapping `cut` , `clarity` , and `color` to x, color/fill, and facets to produce a reasonable plot. Use whichever of boxplots or violin plots is more legible.

```
ggplot(diamonds, aes(x = color, y = carat)) +
    geom_boxplot(aes(fill = cut)) +
    facet_wrap(~clarity)
```



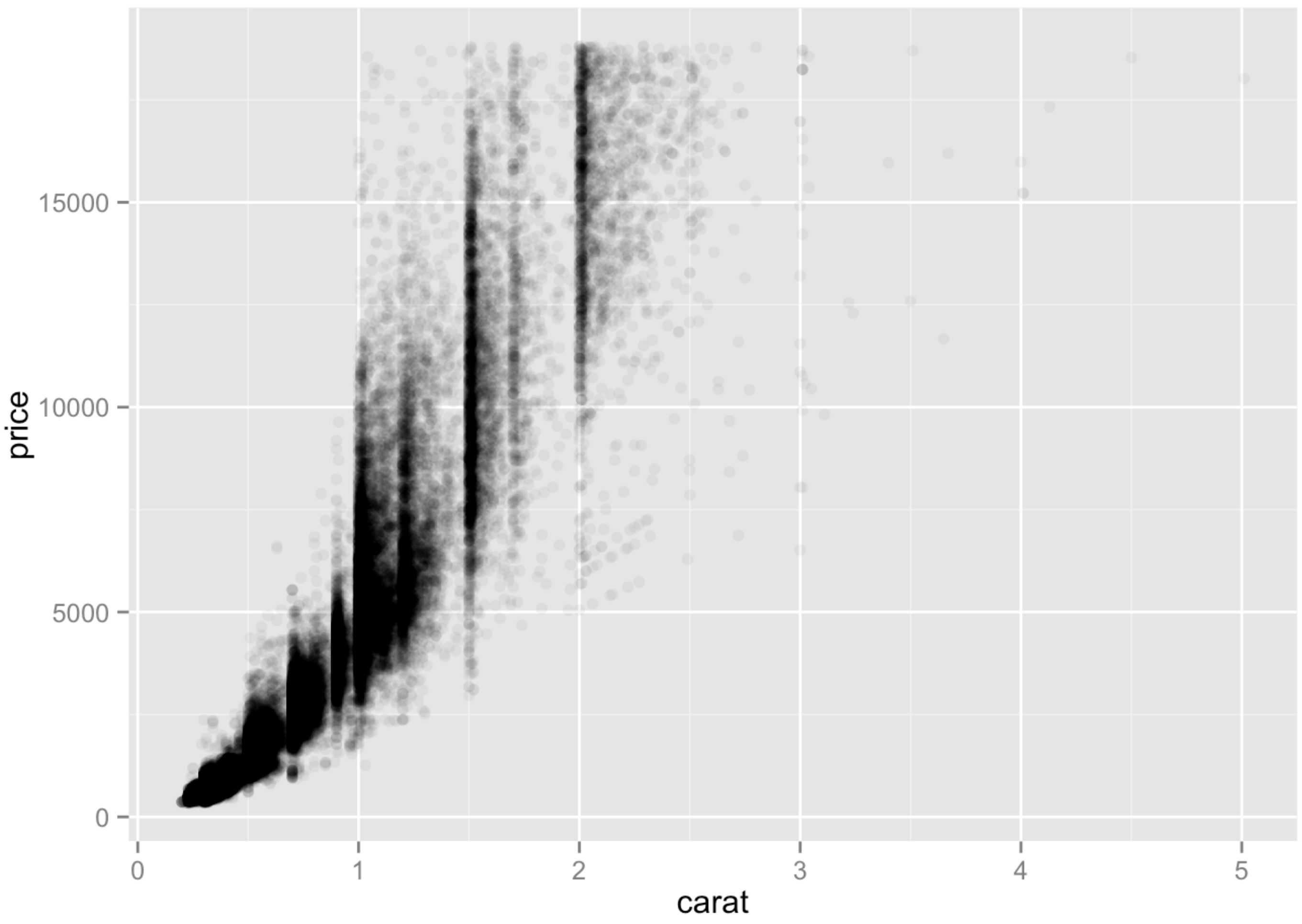
6 a. Create a scatterplot of diamond price as a function of size.

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



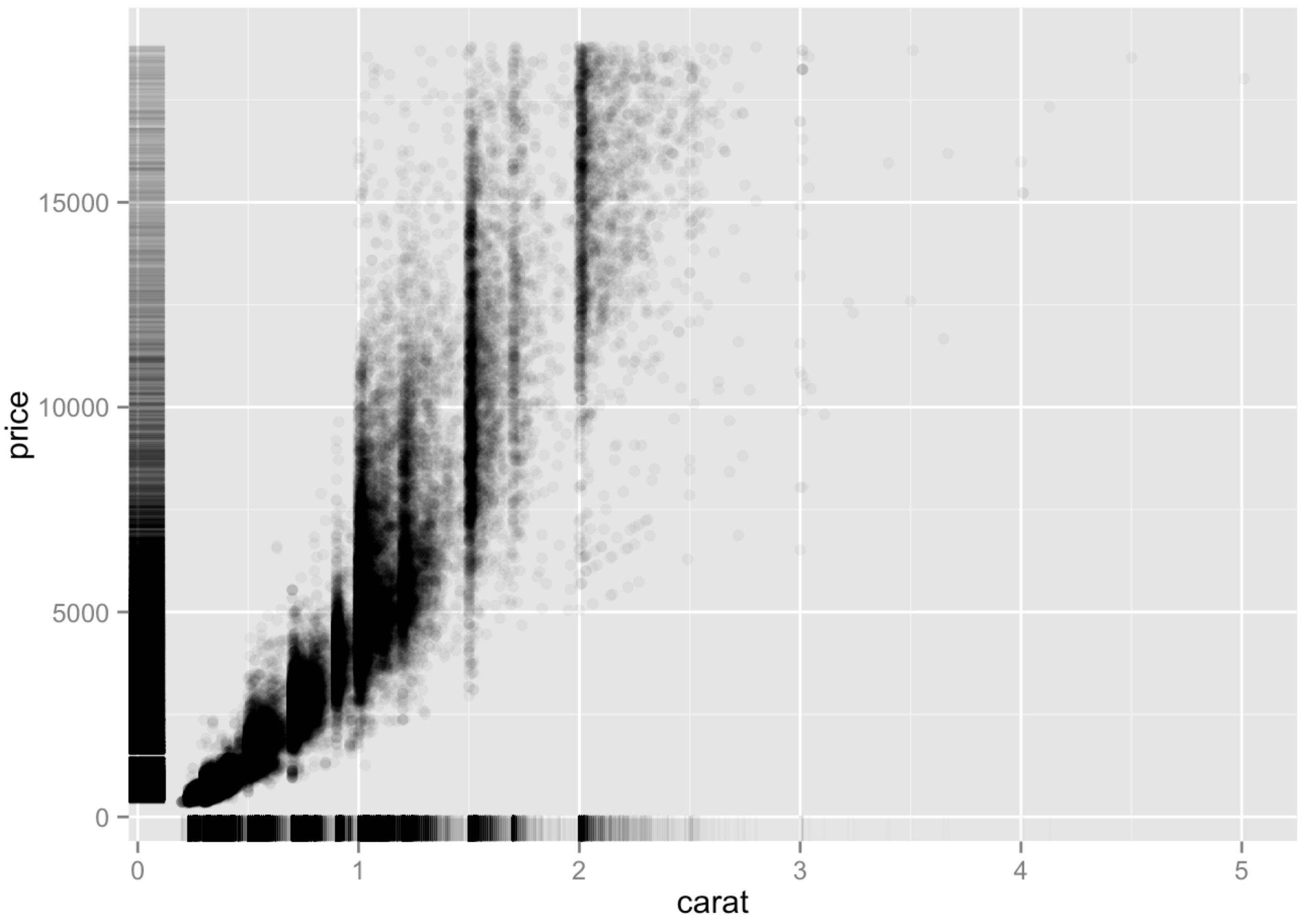
```
b. There is a lot of overplotting there, so information is being lost. Adjust the
alpha level of the points to fix this.
```

```
ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(alpha = .05)
```



```
c. Add marginal rug plots to provide information about the distribution of both va
riables. Consider adjusting the alpha level of the rug plots too.
```

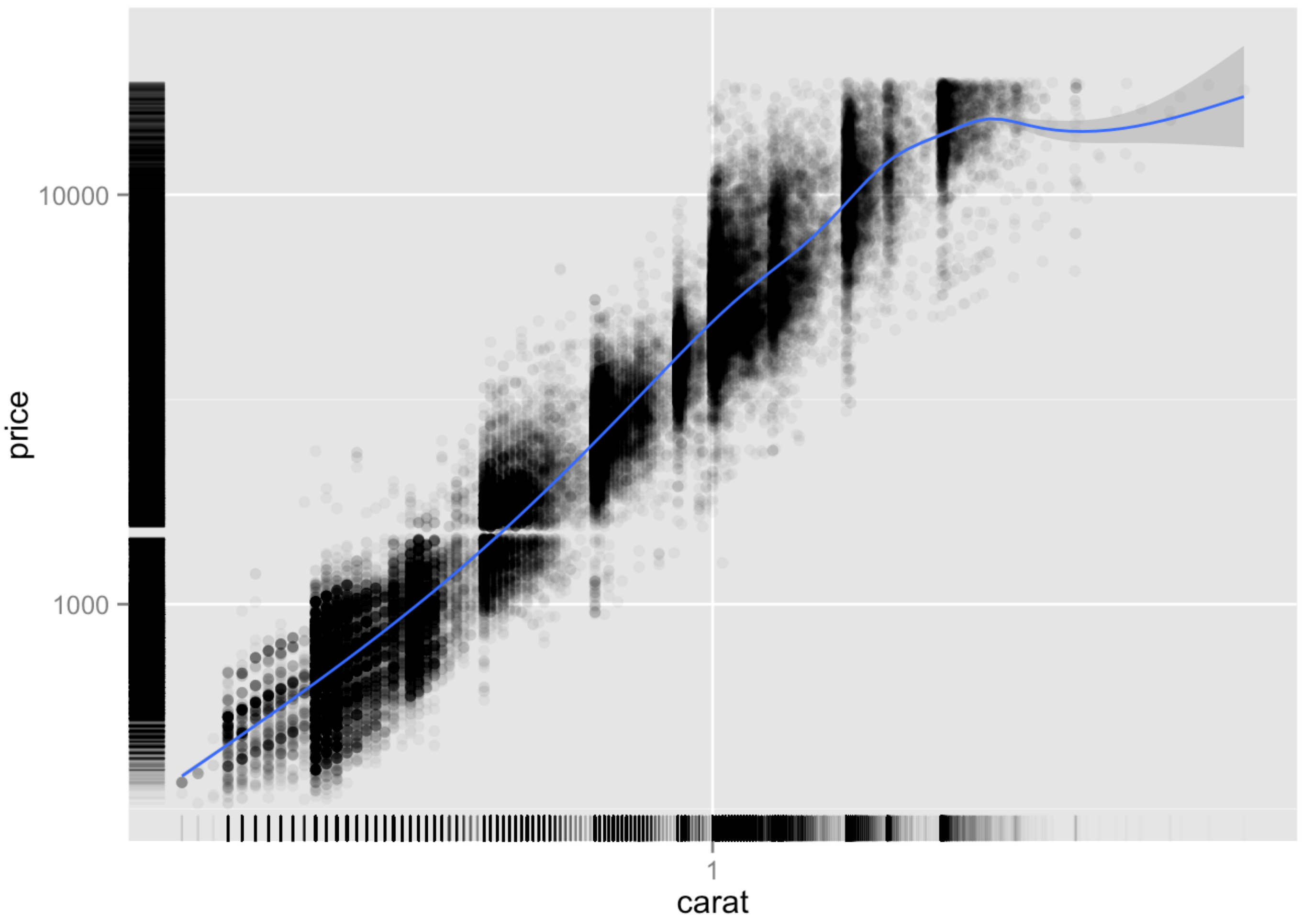
```
ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(alpha = .05) +
    geom_rug(alpha = .01)
```



```
d. Add a smoothing curve to plot. What do you think about the right side of the cu
rve? Consider that and the marginal distributions of each variable. Should you tra
nsform either or both of the axes? Experiment.
```

```
ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(alpha = .05) +
    geom_rug(alpha = .01) +
    geom_smooth() +
    scale_x_log10() +
    scale_y_log10()
```

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using gam wi
th formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.
```

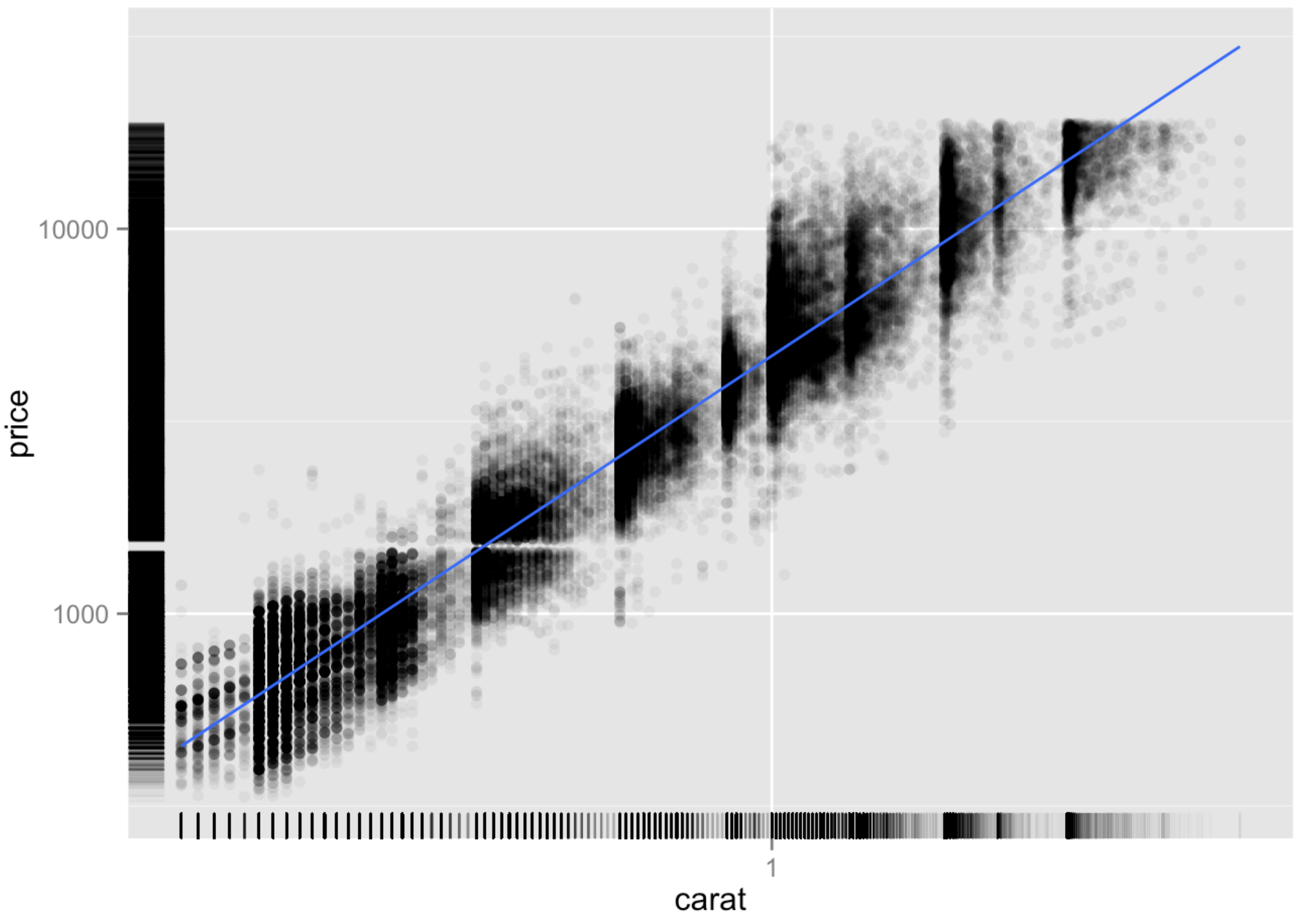


```
e. An expert tells you that diamonds larger than 3 carats or smaller than 0.25 car
ats are outliers. Set the x-axis limits to exclude those outliers. Also, change th
e smoothing algorithm to a linear fit.
```

```
ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(alpha = .05) +
    geom_rug(alpha = .01) +
    geom_smooth(method = 'lm') +
    scale_x_log10(limits = c(.25, 3)) +
    # or xlim(c(.1, 3))
    scale_y_log10()
```

```
## Warning: Removed 605 rows containing missing values (stat_smooth).
```

```
## Warning: Removed 605 rows containing missing values (geom_point).
```



```
f. The same expert tells you the relationship between price and size depends on th
e quality of the `cut`. Test her statement by coloring according to `cut`. Should
the rugs be colored by cut? How about the fitted lines? Does her statement appear
to be true?
```

```
ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(alpha = .05, aes(color = cut)) +
    geom_rug(alpha = .01) +
    geom_smooth(method = 'lm', aes(color = cut)) +
    scale_x_log10(limits = c(.25, 3)) +
    scale_y_log10()
```

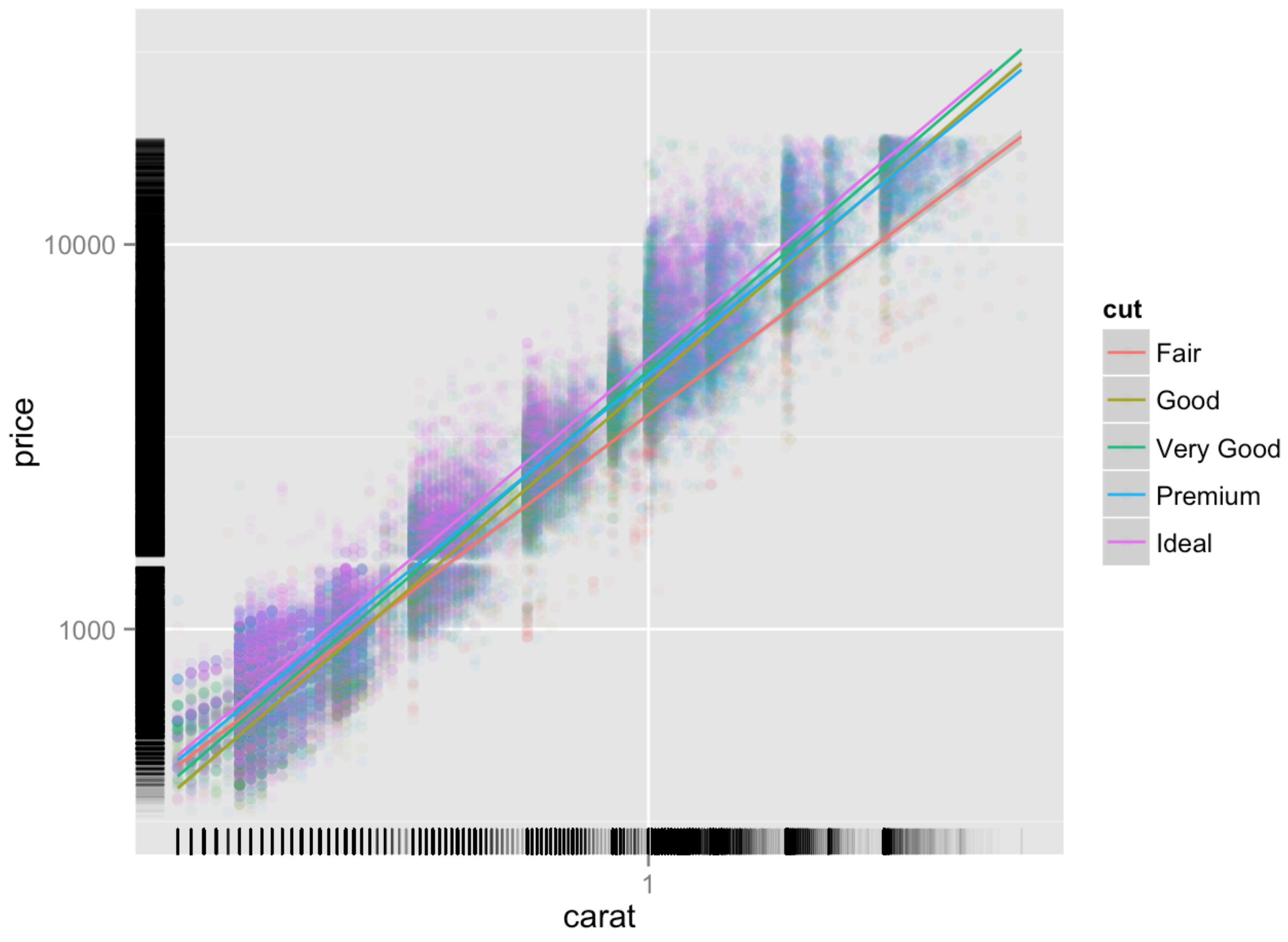
```
## Warning: Removed 12 rows containing missing values (stat_smooth).
```

```
## Warning: Removed 48 rows containing missing values (stat_smooth).
```

```
## Warning: Removed 355 rows containing missing values (stat_smooth).
```

```
## Warning: Removed 70 rows containing missing values (stat_smooth).
## Warning: Removed 120 rows containing missing values (stat_smooth).
```

```
## Warning: Removed 605 rows containing missing values (geom_point).
```



```
# Color gets lost in the rug plots, so they probably shouldn't be colored
# (but perhaps you can see patterns where I don't).
# The lines absolutely should be colored because we want to see if the slope
# 3of the lines are different for the various `cut`s.
```

```
g. Challenge question: Time to get the plot ready for publication! Replace the ugl
y gray background by using the `theme_bw`. Remove the rug plots and replace them w
ith log-tick marks if either variable is log-transformed. Change the axis labels t
o variable (unit). Reverse the order of the legend entries so that they are in the
same order as the lines in the plot, change the legend title to "Cut quality", and
move the legend to the empty space in the upper-left part of the graph. The "R Coo
kbook" <http://www.cookbook-r.com/Graphs/> may be helpful here; ggplot Google quer
ies often lead to it.
```

```
ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(alpha = .05, aes(color = cut)) +
    geom_smooth(method = 'lm', aes(color = cut)) +
    scale_x_log10(name = "Size (carat)", limits = c(.25, 3), breaks = c(.5, 1, 2,
3)) +
    scale_y_log10(name = "Price (USD)") +
    annotation_logticks() +
    scale_color_discrete(guide = guide_legend(title = "Cut quality", reverse = TRU
E)) +
    theme_bw() +
    theme(legend.position = c(0.1, 1),
          legend.justification = c(0, 1))
```

```
## Warning: Removed 12 rows containing missing values (stat_smooth).
```

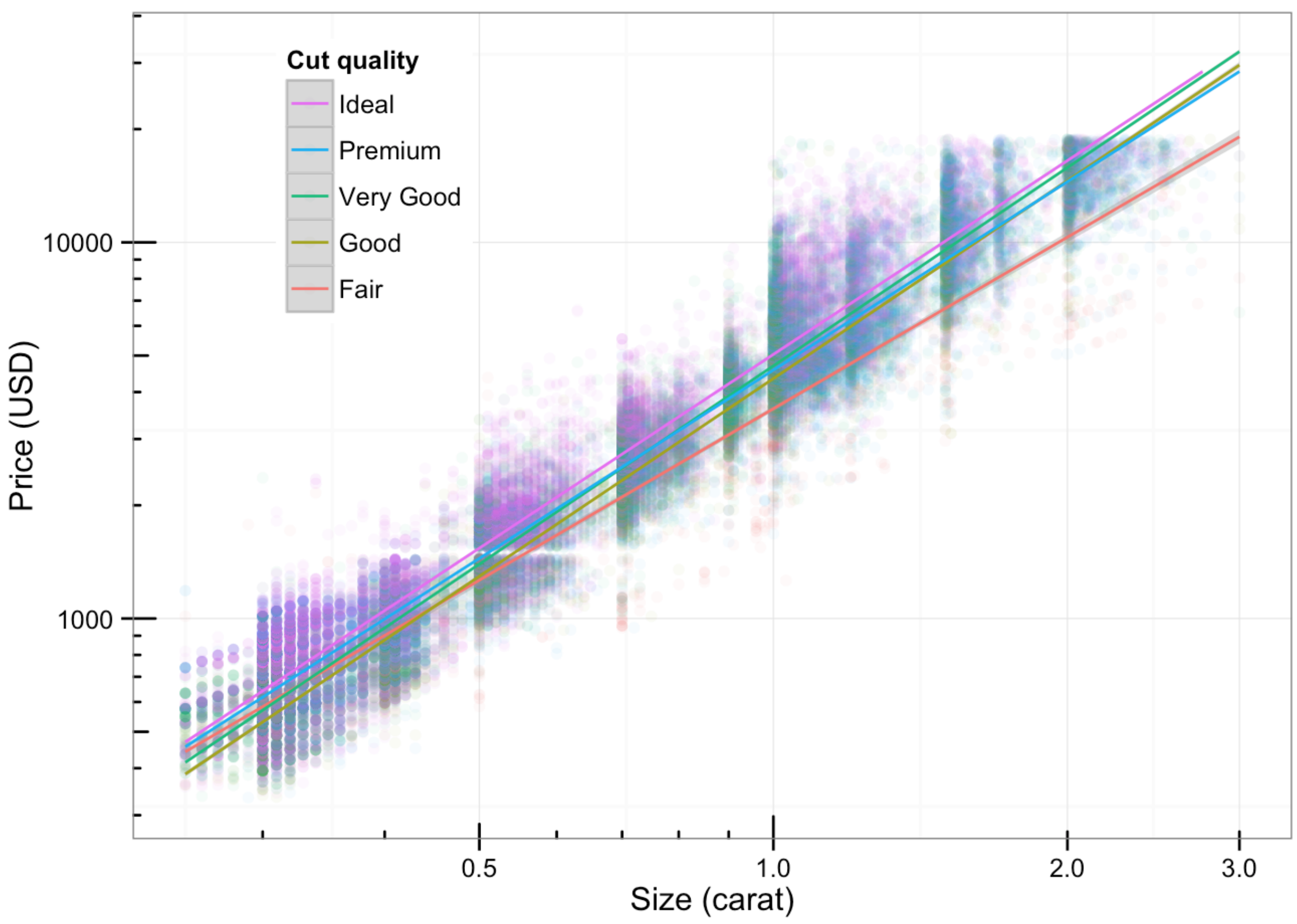
```
## Warning: Removed 48 rows containing missing values (stat_smooth).
```

```
## Warning: Removed 355 rows containing missing values (stat_smooth).
```

```
## Warning: Removed 70 rows containing missing values (stat_smooth).
```

```
## Warning: Removed 120 rows containing missing values (stat_smooth).
```

```
## Warning: Removed 605 rows containing missing values (geom_point).
```



7. There are several other variables in the `diamonds` dataset. Explore their distributions, marginal distributions, relationships with each other, and effects on the relationship between price and size. More information on the dataset is available via `?diamonds` . Consider mapping variables to color/fill, shape/linetype, axes, facets, etc.