MSCS 264: Homework #1

Due Tuesday, Feb 14 at 11pm

Al Ashir Intisar

Data Visualization

You will be using the `diamonds` dataset in the `ggplot2` package to examine factors that influence the price of a diamond.

A few quick notes on submission.

- Make sure I can see your code in addition to output and commentary
- Upload only your knitted pdf, but make sure your (appropriately named) RMarkdown file is available in your personal Submit folder
- Get started early! Remember these are not intended to be completed in one night
- Put your name in the author line!
- The RMarkdown file that produced this pdf is on the RStudio server under `Mscs 264b S21` > `Class` > `Homework` > `hw01.Rmd`. You should start by saving a copy of `hw01.Rmd` in your Submit folder, and then you can use this as a template for filling in R code and written answers.

IMPORTANT: The `diamonds` data set is very large, so we will only be working with a subset of the data called `smaller`. The two chunks below load the packages you need and create the `smaller` dataset.

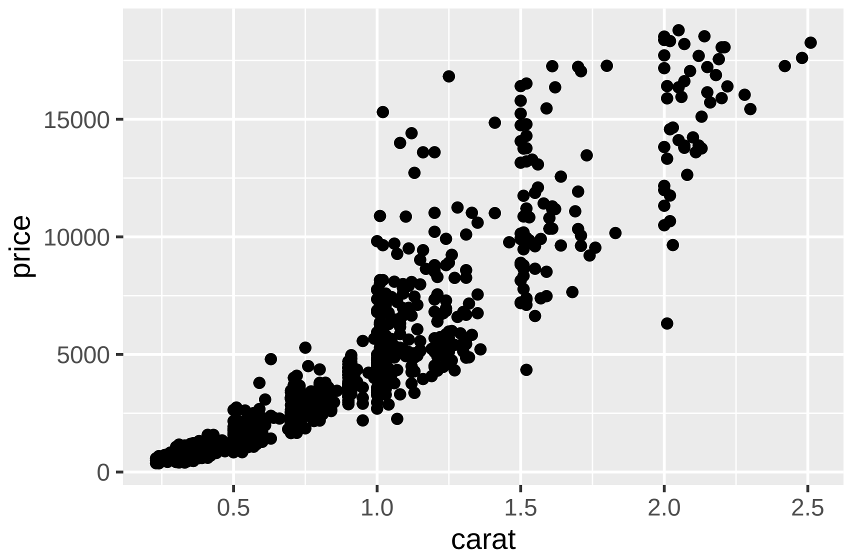
```r
set.seed(1)
smaller <- diamonds %>%
  filter(carat <= 3) %>%
  slice_sample(n = 1000)
```

(a) First, examine the documentation for the diamonds data using `?diamonds`. How many observations are there? How many variables?

Ans: This dataset contains the prices and other sttributes of almost 54,000 diamonds. It is a dataframe with 53940 rows and 10 variables.

(b) Generate a scatterplot of diamond price versus size in carats, where price is the response and size is the explanatory variable. Describe the relationship in one sentence.

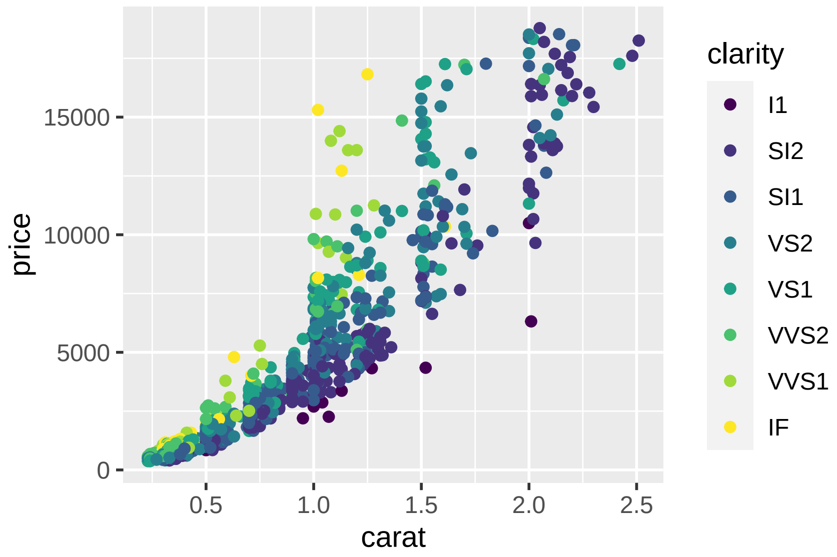
```r
ggplot(data = smaller) +
geom_point(mapping = aes(x = carat, y = price))
```



Ans: As the size and the carats variable has a strong positive correlation.

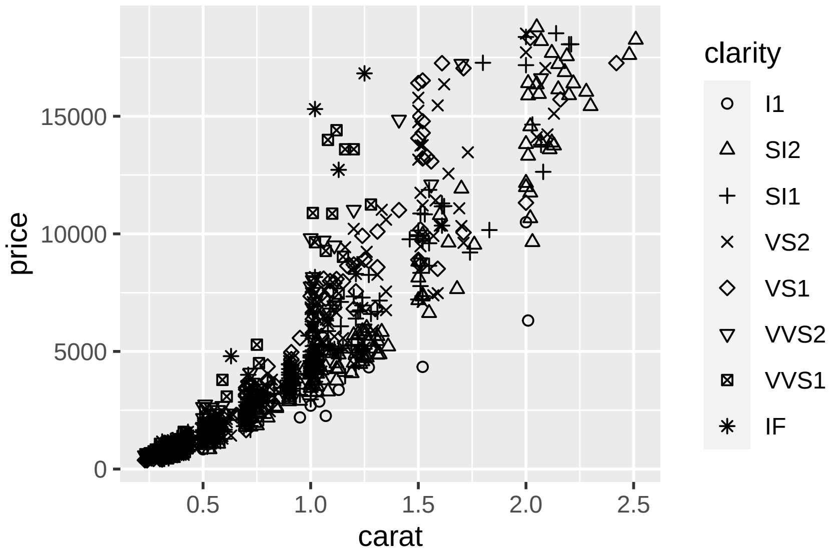
(c) Modify your plot in (b) so that point color is determined by diamond clarity.

```
ggplot(data = smaller) +
geom_point(mapping = aes(x = carat, y = price, color = clarity))
```



(d) Modify your plot in (b) so that point shape is determined by diamond clarity. Be sure to assign each clarity a unique shape.

```
ggplot(data = smaller) +
geom_point(mapping = aes(x = carat, y = price, shape = clarity)) +
scale_shape_manual(values=c(1,2,3,4,5,6,7,8))
```

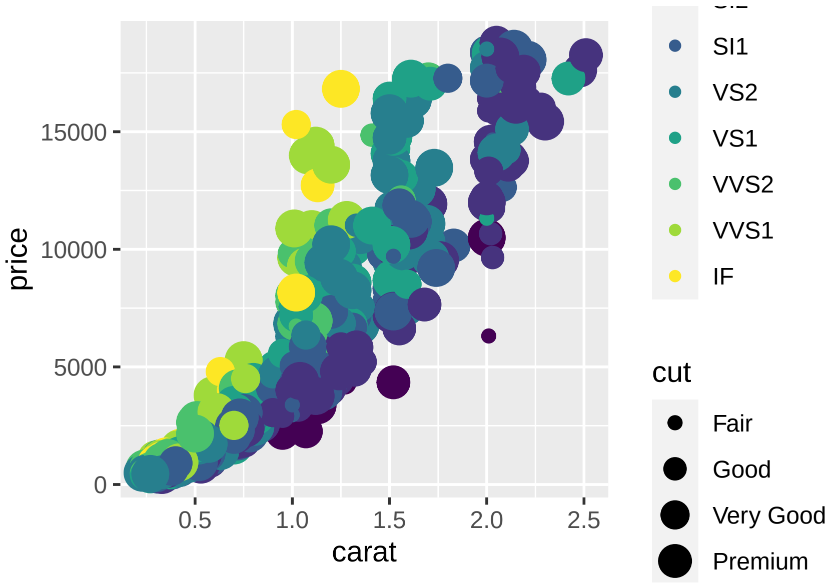


(e) Does plot (c) or (d) tell a better story? Explain briefly.

Ans: Plot (c) tells a better story than plot (d) because the overlapping of the shapes makes it harder to distinguish one point from the other and thus makes a visual representation of the data harder.

(f) Modify your plot in (c) so that point size is determined by diamond cut.

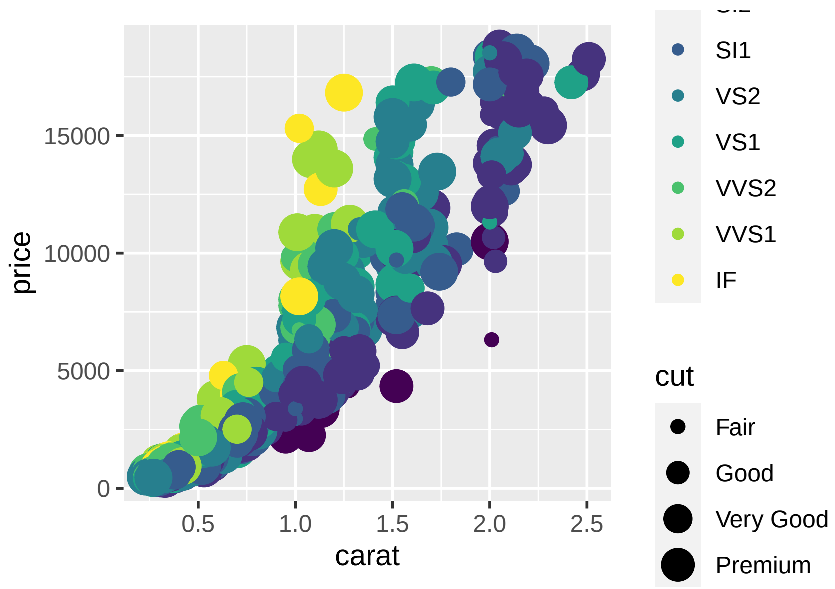
```
ggplot(data = smaller) +
geom_point(mapping = aes(x = carat, y = price, color = clarity, size = cut))
```



1. Notice in your plot in (f) the default size is too large. Reduce the size of the points, while still allowing better cuts to have larger sizes. You might explore `scale_size_discrete` or `scale_size_manual`.

```
ggplot(data = smaller) +
geom_point(mapping = aes(x = carat, y = price, color = clarity, size = cut)) +
scale_size_discrete()
```

```
## Warning: Using size for a discrete variable is not advised.
```



```
scale_size_manual(values=c(1,2,3,4,5))
```

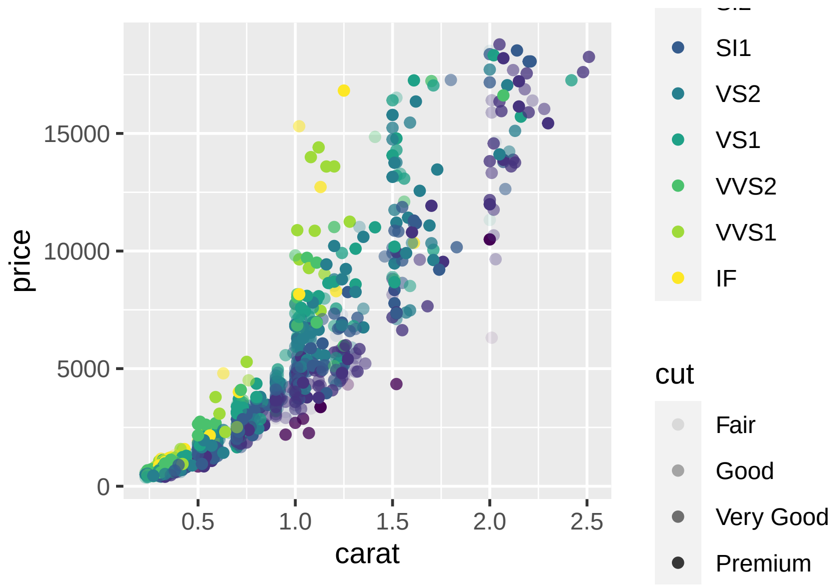
```
## <ggproto object: Class ScaleDiscrete, Scale, gg>
##     aesthetics: size
##     axis_order: function
##     break_info: function
##     break_positions: function
##     breaks: waiver
##     call: call
##     clone: function
##     dimension: function
##     drop: TRUE
##     expand: waiver
##     get_breaks: function
##     get_breaks_minor: function
##     get_labels: function
##     get_limits: function
##     guide: legend
##     is_discrete: function
##     is_empty: function
##     labels: waiver
##     limits: NULL
##     make_sec_title: function
##     make_title: function
##     map: function
##     map_df: function
##     n.breaks.cache: NULL
##     na.translate: TRUE
##     na.value: NA
##     name: waiver
##     palette: function
##     palette.cache: NULL
##     position: left
##     range: <ggproto object: Class RangeDiscrete, Range, gg>
##         range: NULL
##          reset: function
##          train: function
##          super:  <ggproto object: Class RangeDiscrete, Range, gg>
##      rescale: function
##      reset: function
##      scale_name: manual
##      train: function
##      train_df: function
##      transform: function
##      transform_df: function
##      super:  <ggproto object: Class ScaleDiscrete, Scale, gg>
```

2. R is giving you a warning when you use `scale_size_discrete`. What is going on here? What is going on here?

Ans: The aesthetic size is better suited for continuous values whereas 'cut' is a discrete variable. In the graph in (1) we can see that it is very hard to distinguish one group of cut from other group of cut even though individually they are easy to distinguish.

3. Does adjusting alpha produce a better plot versus reducing the size of each point?

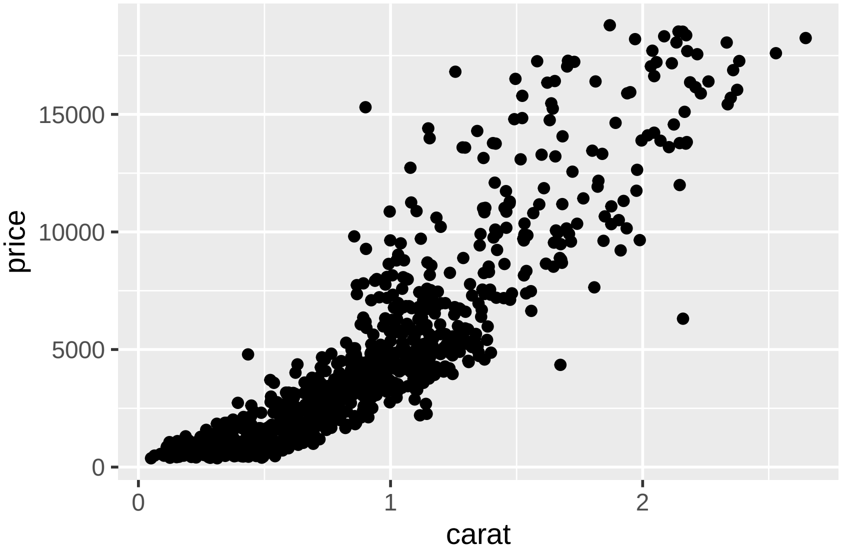
```r
ggplot(data = smaller) +
geom_point(mapping = aes(x = carat, y = price, color = clarity, alpha = cut))
```



Ans: Adjusting alpha does help better visualize the points compared to size where they are overlapped and hard to distinguish.

(g) Return again to (b). Does jittering points produce a better plot? Don't use just the default amount of jittering - play with more and less and print the plot you think looks best.

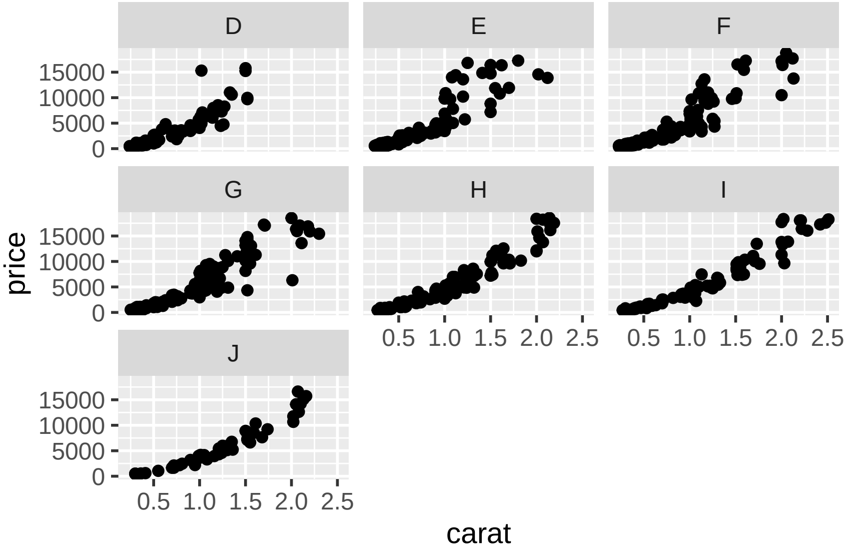
```r
ggplot(data = smaller) +
geom_jitter(mapping = aes(x = carat, y = price), width = 0.2, height = 10)
```



Ans: Yes, jittering creates a nicer plot where more data points are visible. The carat and rpice variable has a steep positive correlation. Therefore, it is better to use a smaller value as width and a higher value for height so that the plot is not too distorted.

(h) Examine the relationship between price and carats by color, creating one plot per color. Describe what story this visualization is telling. Note that colors D, E, and F are colorless (more radiant and valuable), while G, H, I, and J are nearly colorless (it's hard to tell these apart unless the diamonds are very large).

```
ggplot(data = smaller) +
geom_point(mapping = aes(x = carat, y = price)) +
facet_wrap(~color)
```



Ans: The diamonds with colors D, E, and F are more valuable compared to diamonds with colors G, H, I, and J in terms of their weights. That is a diamonds in the first group usually cost more than the diamonds in the second group even when their are of the same carat.