DS_Nov_22_1

Srijan Kundu

2022-11-22

Loading `diamonds` dataset:

```
diamonds
```

```
## # A tibble: 53,940 x 10
##    carat cut       color clarity depth table price     x     y     z
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
##  2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
##  3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
##  4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
##  5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
##  6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
##  7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
##  8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
##  9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
## 10  0.23 Very Good H     VS1      59.4    61   338  4     4.05  2.39
## # ... with 53,930 more rows
```

```
glimpse(diamonds)
```

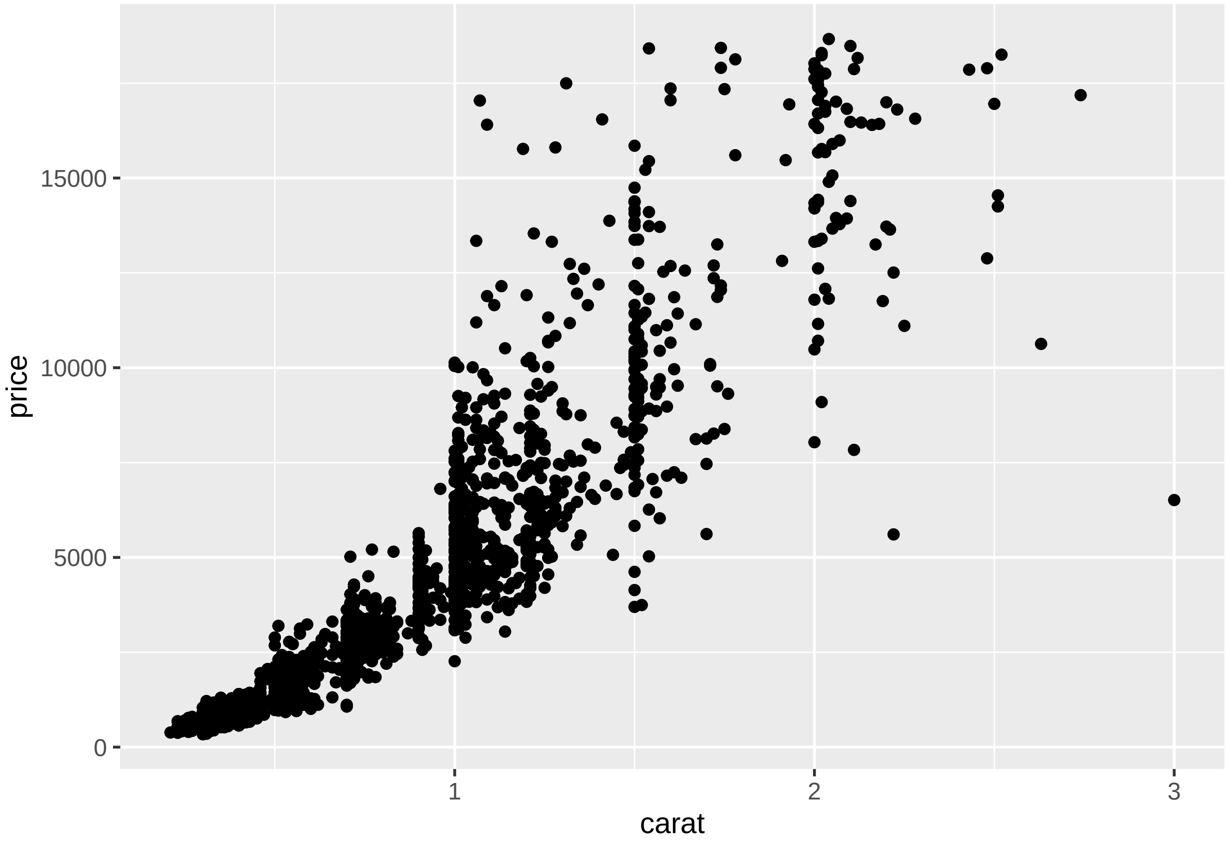
```
## Rows: 53,940
## Columns: 10
## $ carat   <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22, 0.23, 0.~
## $ cut     <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very Good, Ver~
## $ color   <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J, J, J, I,~
## $ clarity <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1, SI1, VS1, ~
## $ depth   <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1, 59.4, 64~
## $ table   <dbl> 55, 61, 65, 58, 58, 57, 57, 55, 61, 61, 55, 56, 61, 54, 62, 58~
## $ price   <int> 326, 326, 327, 334, 335, 336, 336, 337, 337, 338, 339, 340, 34~
## $ x       <dbl> 3.95, 3.89, 4.05, 4.20, 4.34, 3.94, 3.95, 4.07, 3.87, 4.00, 4.~
## $ y       <dbl> 3.98, 3.84, 4.07, 4.23, 4.35, 3.96, 3.98, 4.11, 3.78, 4.05, 4.~
## $ z       <dbl> 2.43, 2.31, 2.31, 2.63, 2.75, 2.48, 2.47, 2.53, 2.49, 2.39, 2.~
```

Random sampling 2000 rows from `diamonds`:

```
data_dia = slice_sample(diamonds, n = 2000)
```

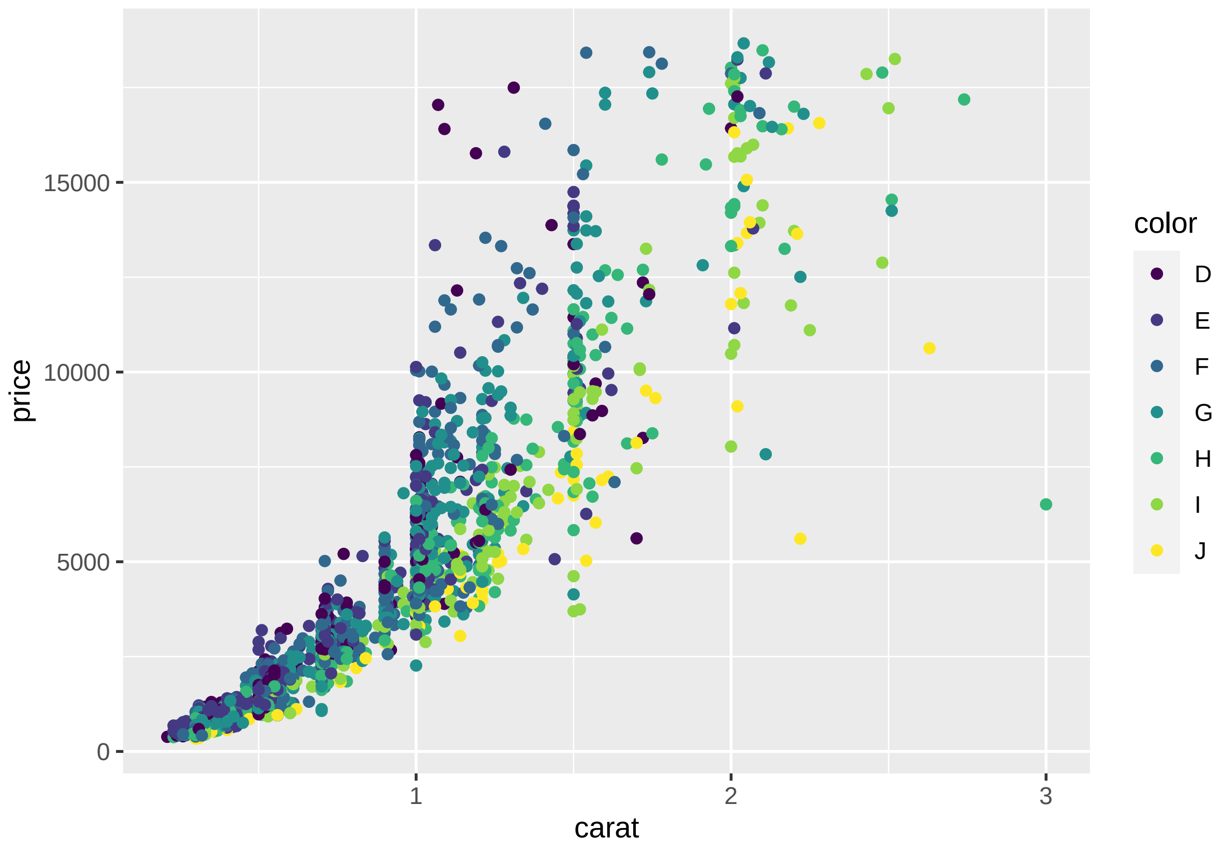
Obtain a scatterplot of the carat and the price:

```
p1 = ggplot(data_dia, aes(x = carat, y = price))
p1 + geom_point()
```



Represent the points in the scatterplot by the specific colors of the specific diamonds:

```
p1 + geom_point(aes(color = color))
```



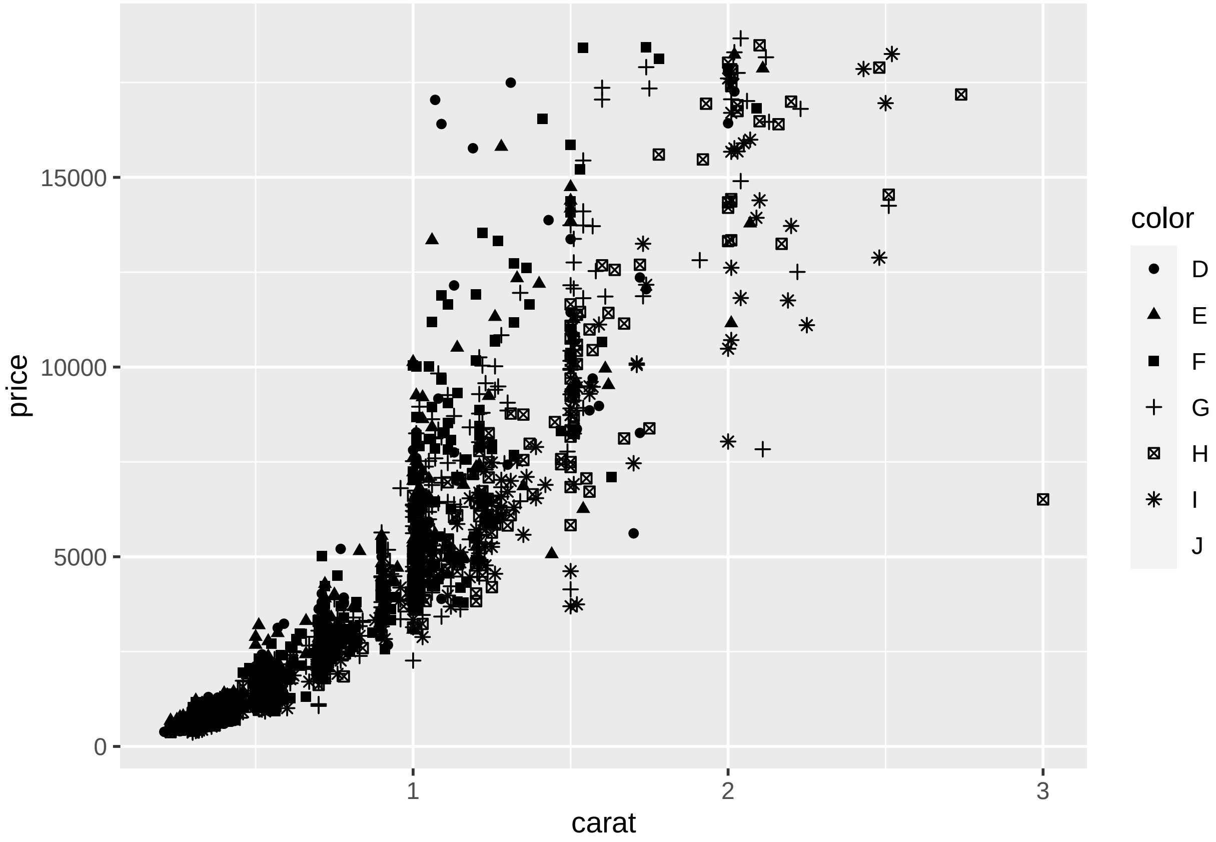
Indicate different colors of different diamonds by different shapes:

```
p1 + geom_point(aes(shape = color))
```

```
## Warning: Using shapes for an ordinal variable is not advised
```

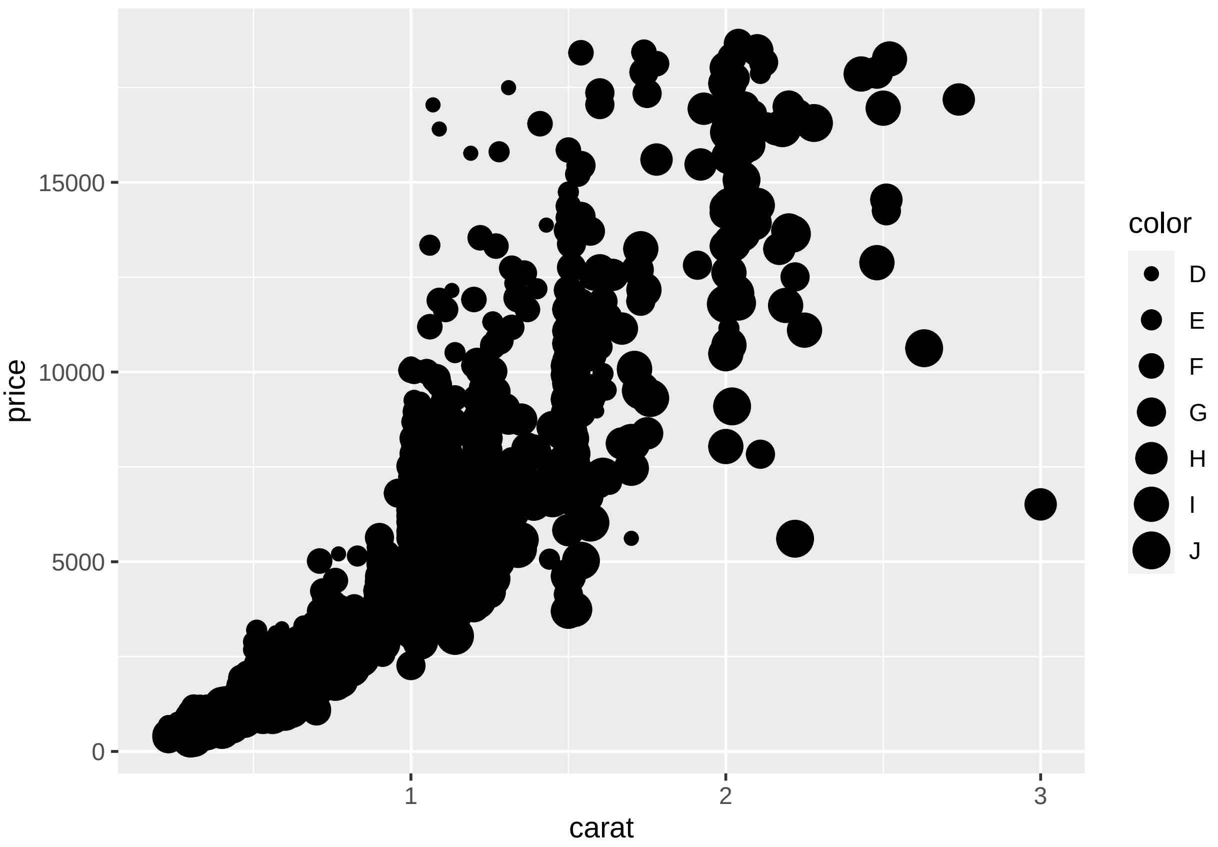
```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 79 rows containing missing values (geom_point).
```



Try size to distinguish as well:

```
p1 + geom_point(aes(size = color))
```



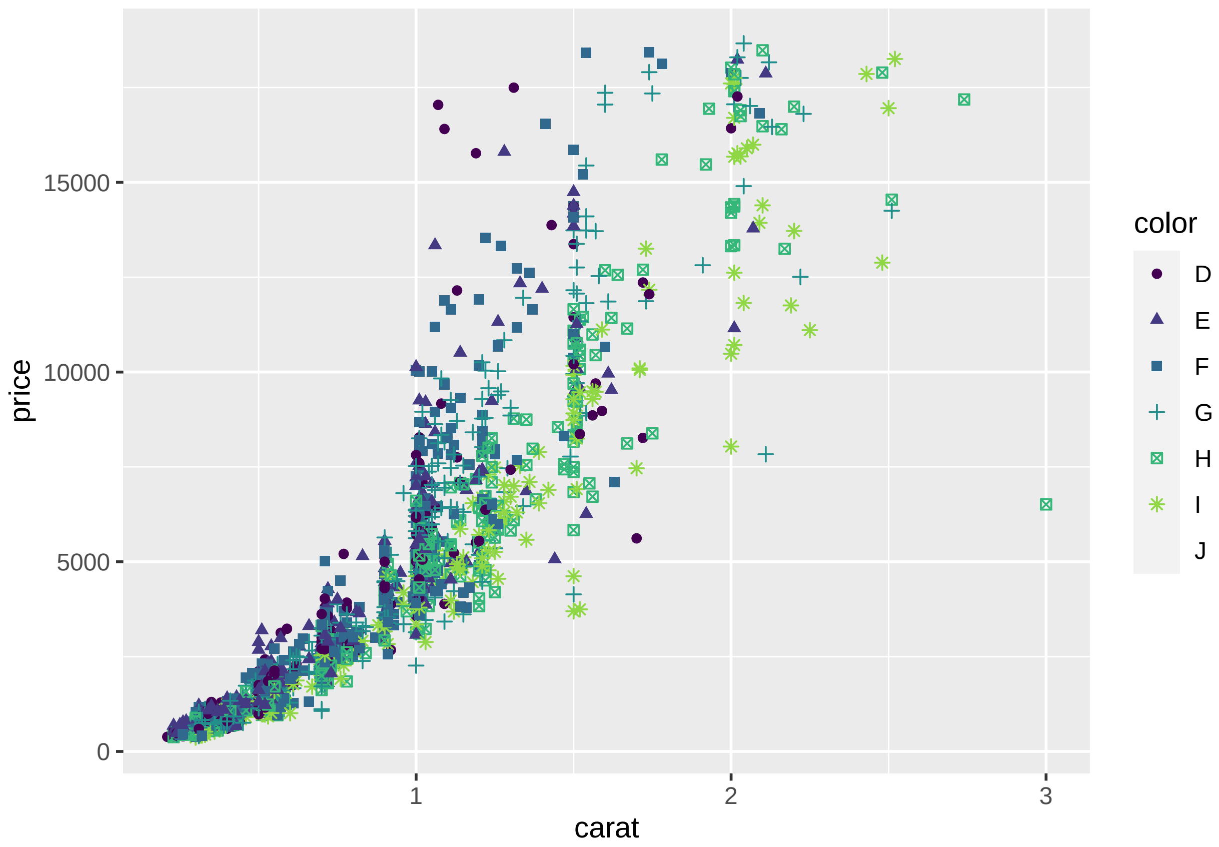
Use both color and shape to distinguish:

```
p1 + geom_point(aes(shape = color, color = color))
```

```
## Warning: Using shapes for an ordinal variable is not advised
```

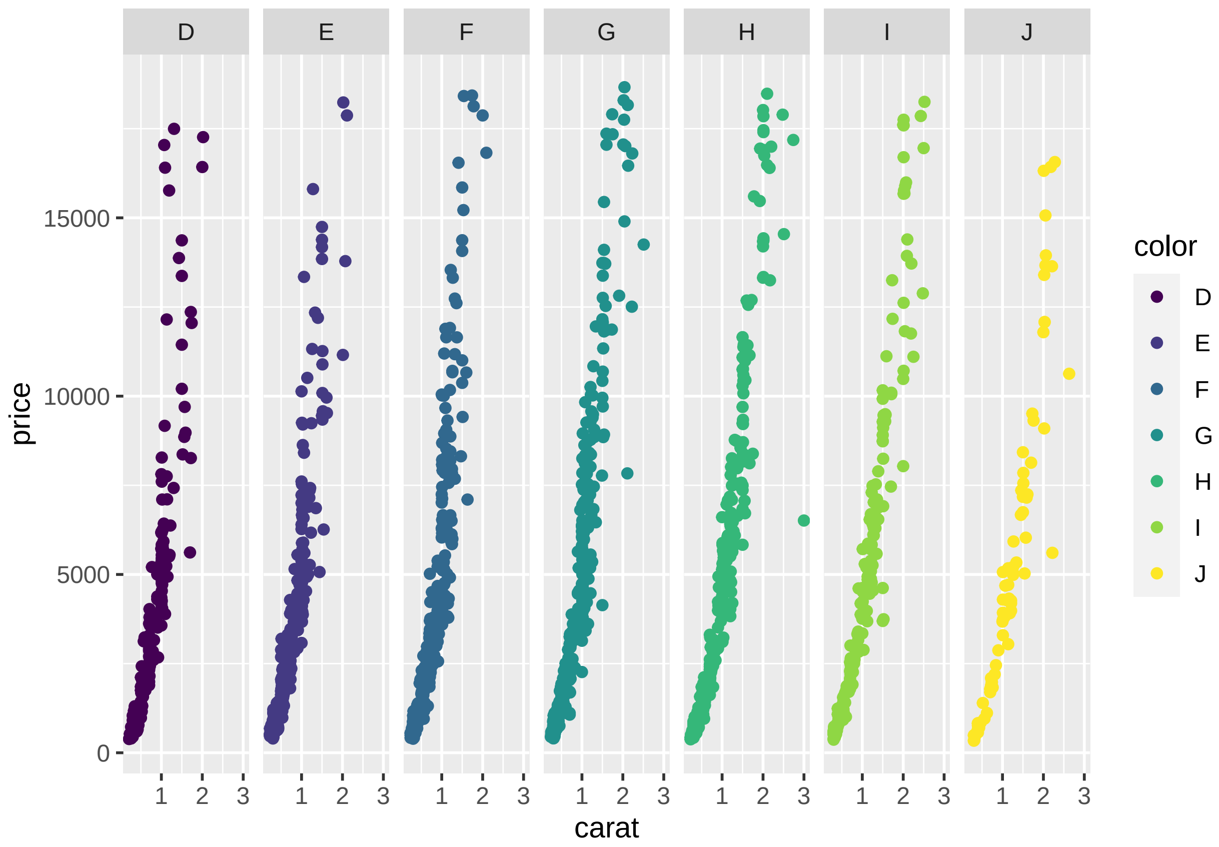
```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 79 rows containing missing values (geom_point).
```

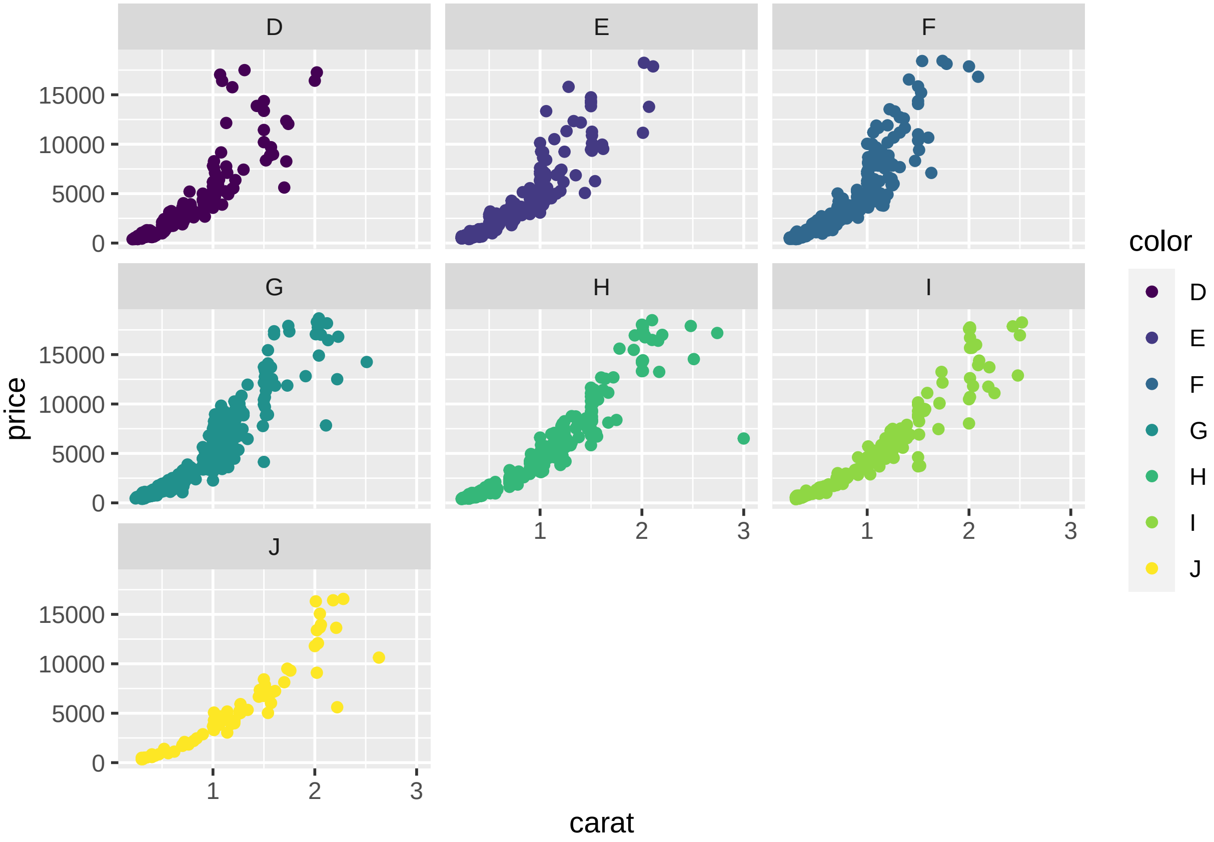


Different scatterplots on the basis of different colors of diamonds:

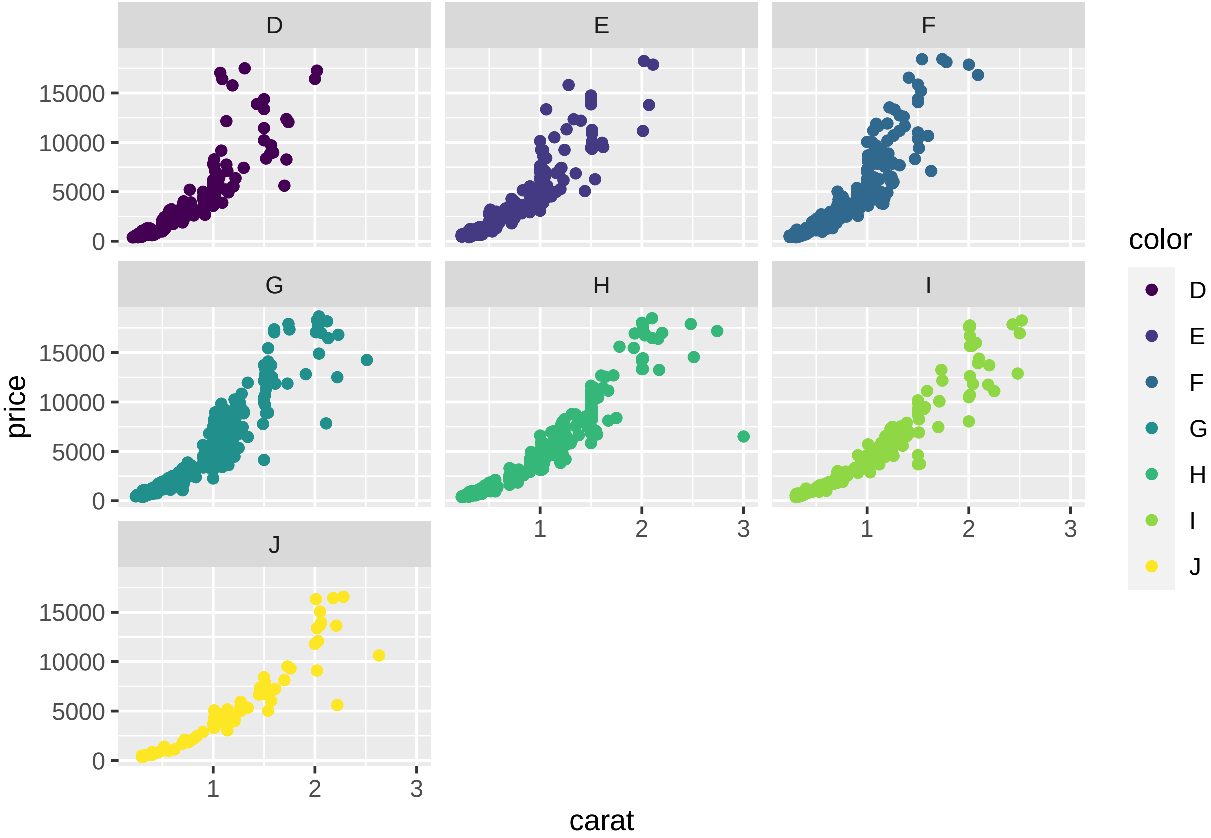
```
p1 + geom_point(aes(color = color)) + facet_grid(~color)
```



```
p1 + geom_point(aes(color = color)) + facet_wrap(~color)
```

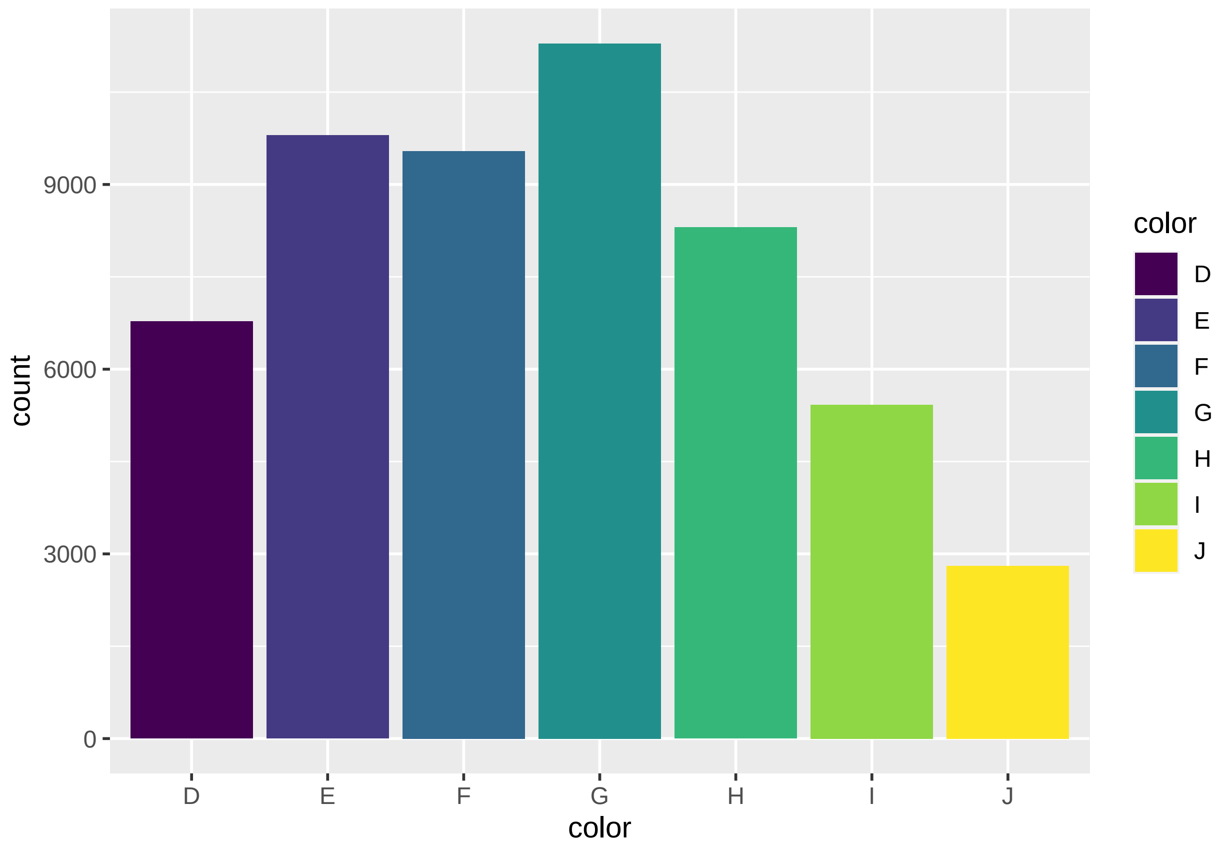


```
p1 + geom_point(aes(color = color)) + facet_wrap(vars(color))
```



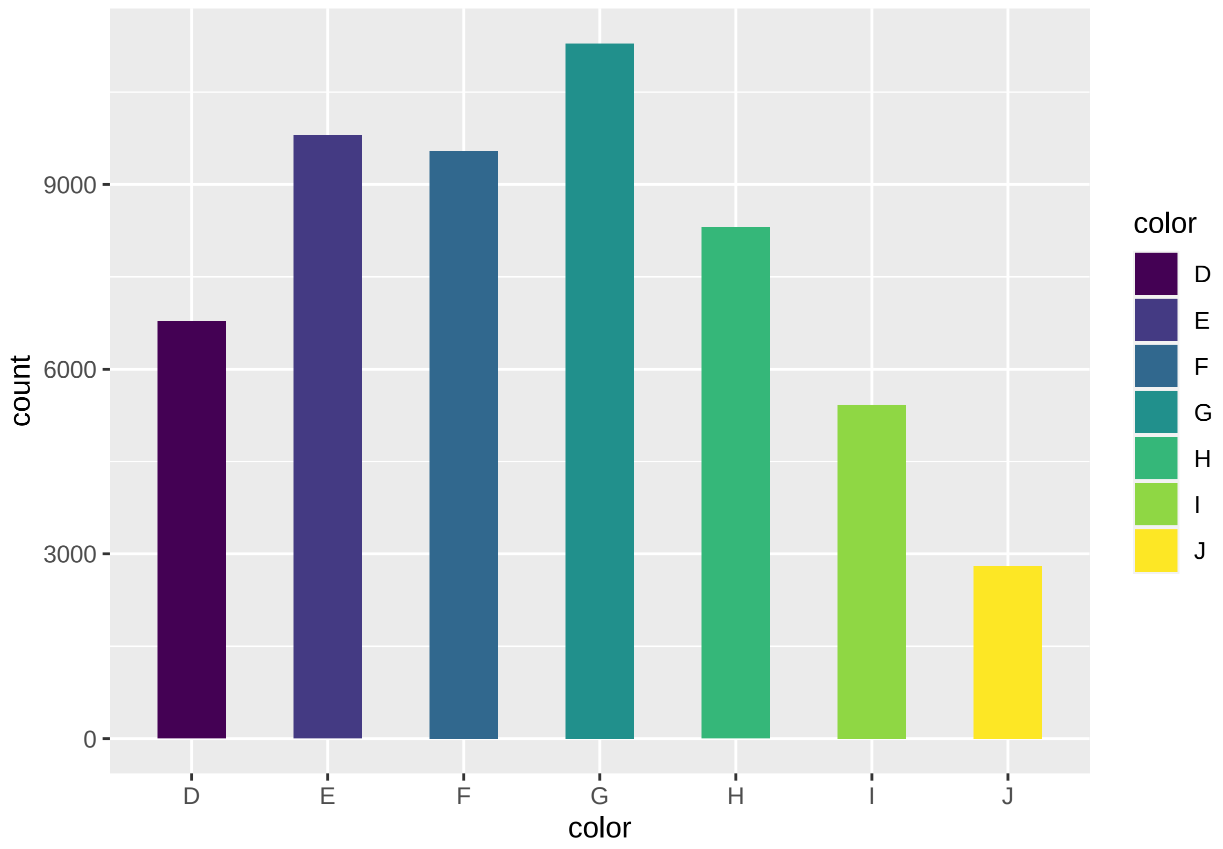
Obtain barplot of the frequency distribution of the colors:

```
p2 = ggplot(diamonds, aes(x = color, fill = color))
p2 + geom_bar()
```

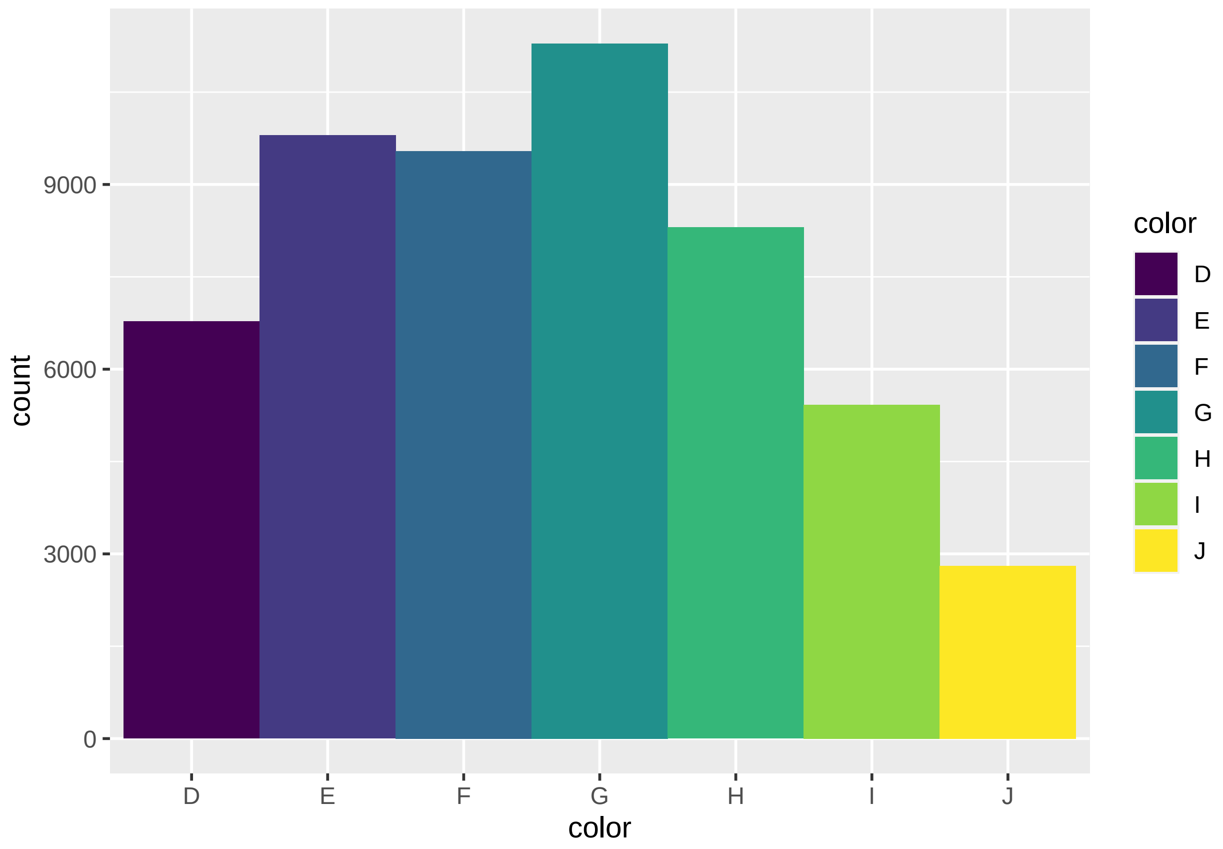


Modify the bar width:

```
p2 + geom_bar(width = 0.5)
```

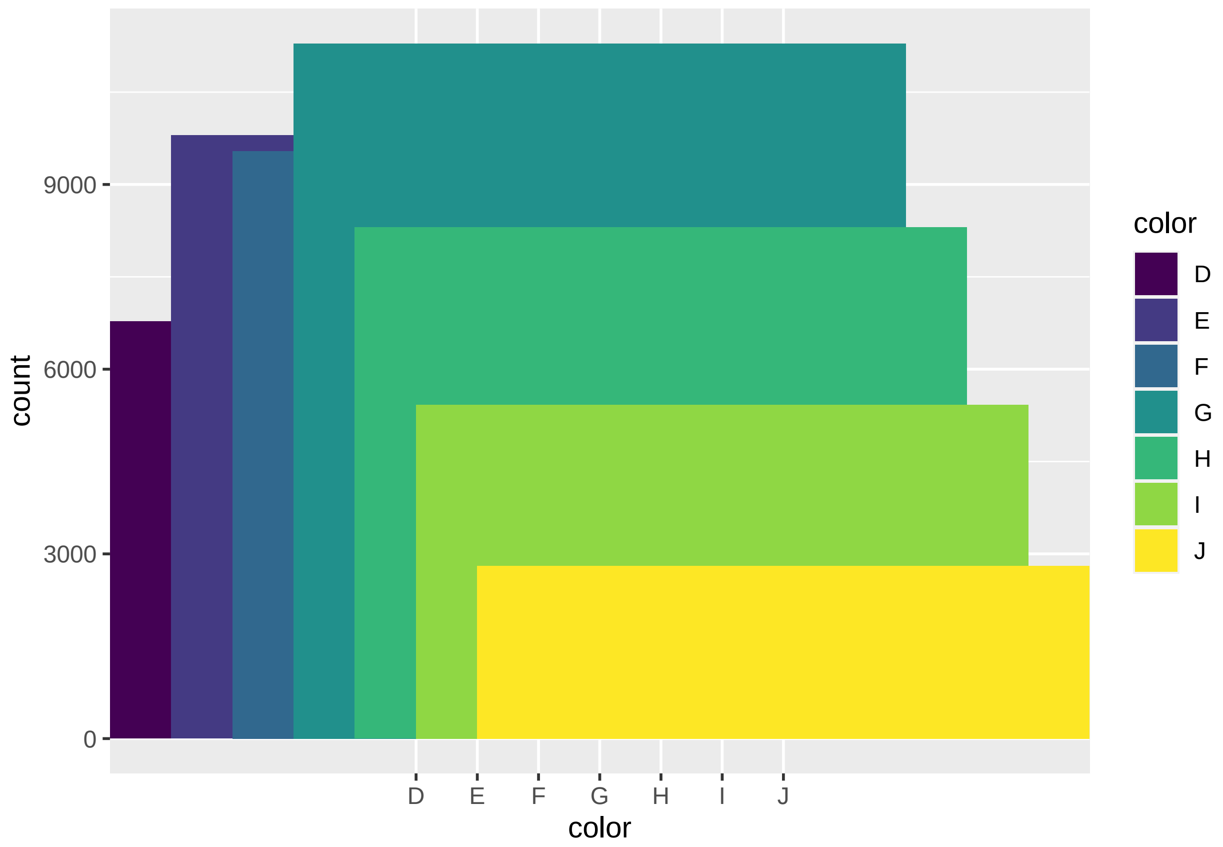


```
p2 + geom_bar(width = 1)
```

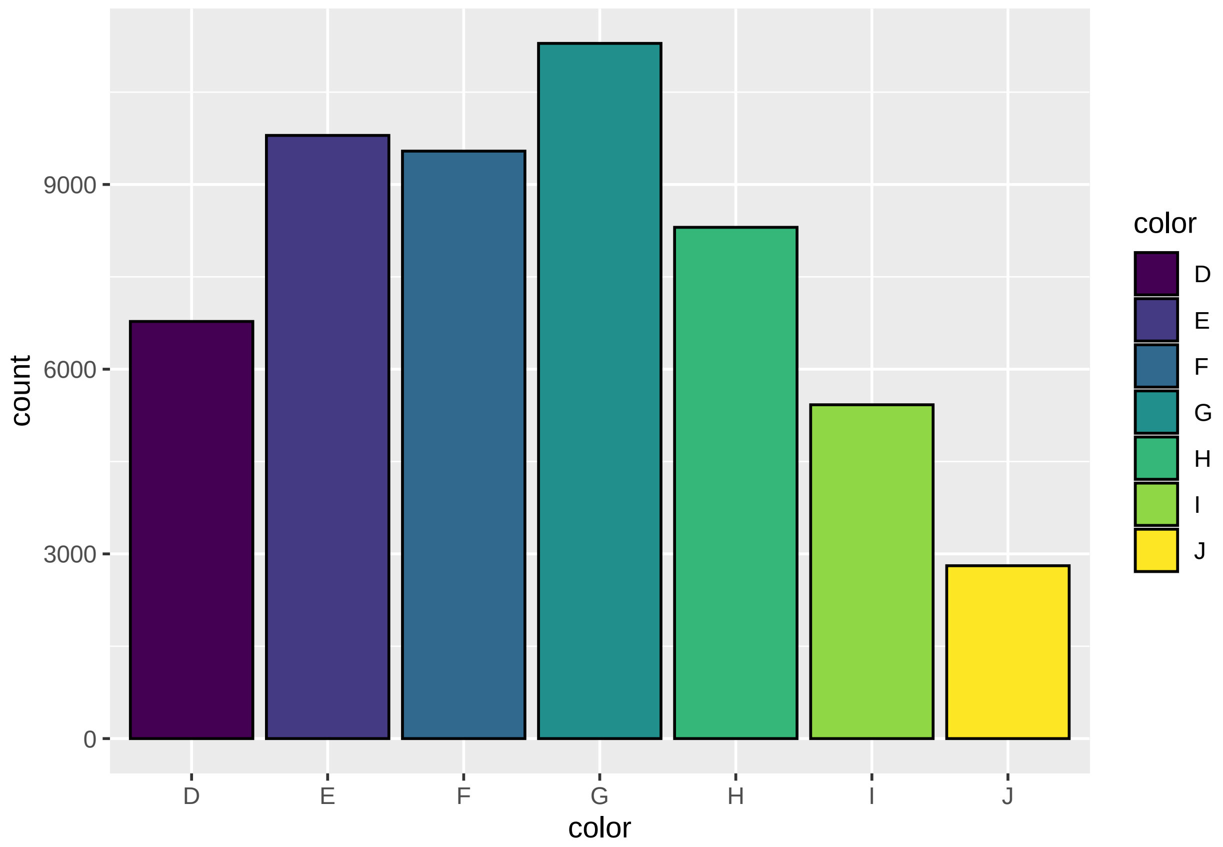


```
p2 + geom_bar(width = 10)
```

```
## Warning: position_stack requires non-overlapping x intervals
```

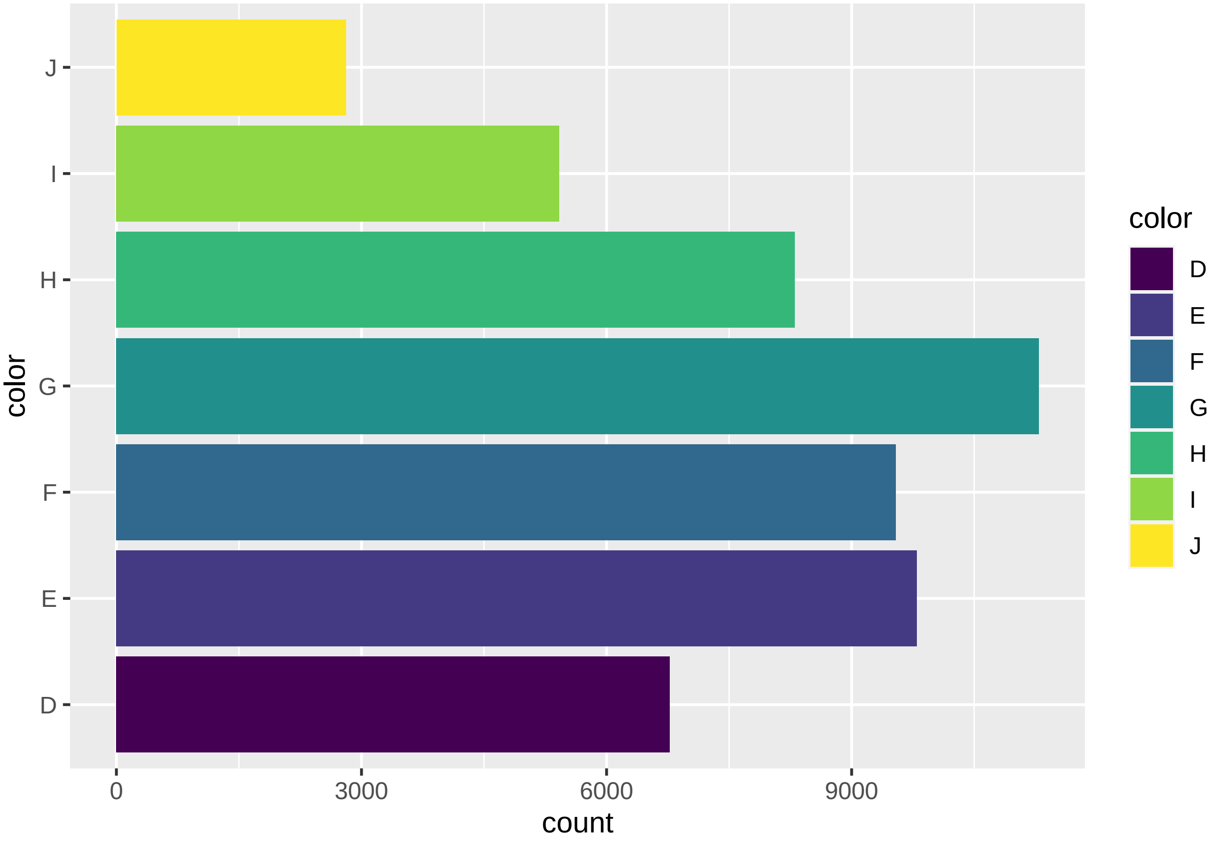


```
p2 + geom_bar(color = "black")
```



Make the bar diagram horizontal:

```
p2 + geom_bar() + coord_flip()
```



Within each color, consider the frequency distribution of different clarity (do in same graph, first in horizontally placed bars and then for stacked for each color):