EDA_FA3_KHAFAJI

FA3 Exploring the Diamonds dataset

Let's explore the diamonds dataset from the tidyverse library

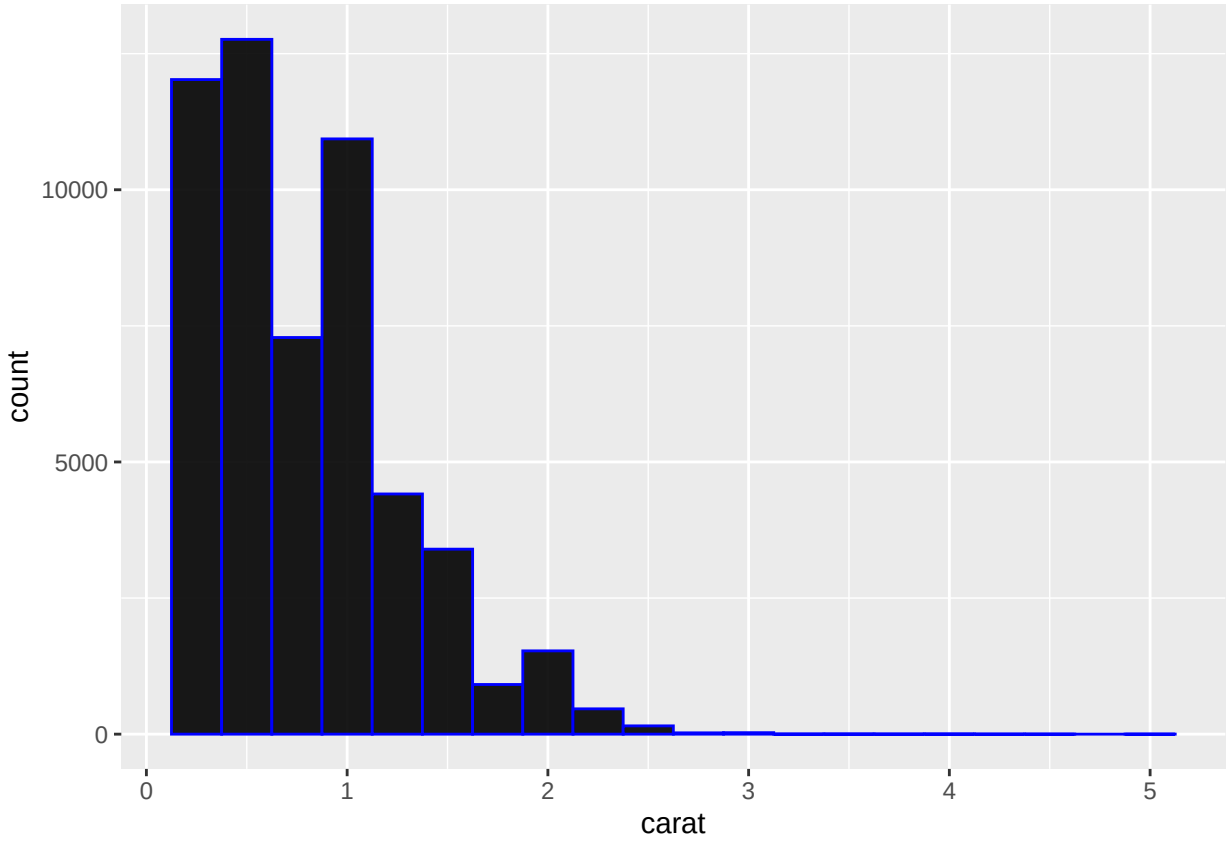
```
diamonds
```

```
## # 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
## # i 53,930 more rows
```

1. Create a histogram

Let's create a histogram of the dataset, but using the layers function of ggplot.

```
diamonds %>% ggplot(aes(x=carat)) +
  layer(
    geom = "bar",
    stat = "bin",
    position = "identity",
    params = list(
      binwidth = 0.25,
      color = "blue",
      fill = "black",
      alpha = 0.9
    )
  )
```

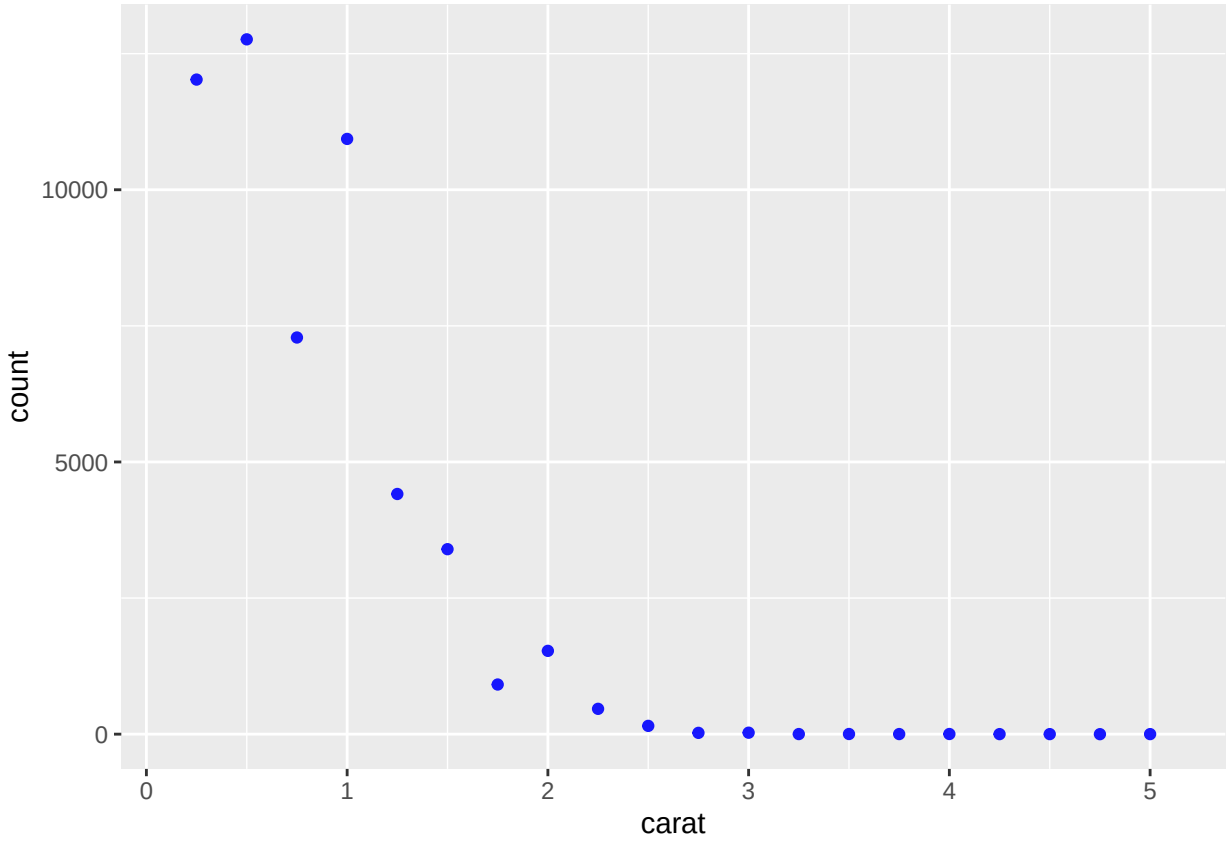


This is a classic histogram, containing details on the amount of diamonds with a given carat, and hence it's distribution.

2. Create a Histogram with point or line geom

We can also do a histogram, but with a line or point geom, by customizing the geom parameter on the layer function parameters.

```
diamonds %>% ggplot(aes(x=carat)) +
  layer(
    geom = "point",
    stat = "bin",
    position = "identity",
    params = list(
      binwidth = 0.25,
      color = "blue",
      fill = "black",
      alpha = 0.9
    )
  )
```

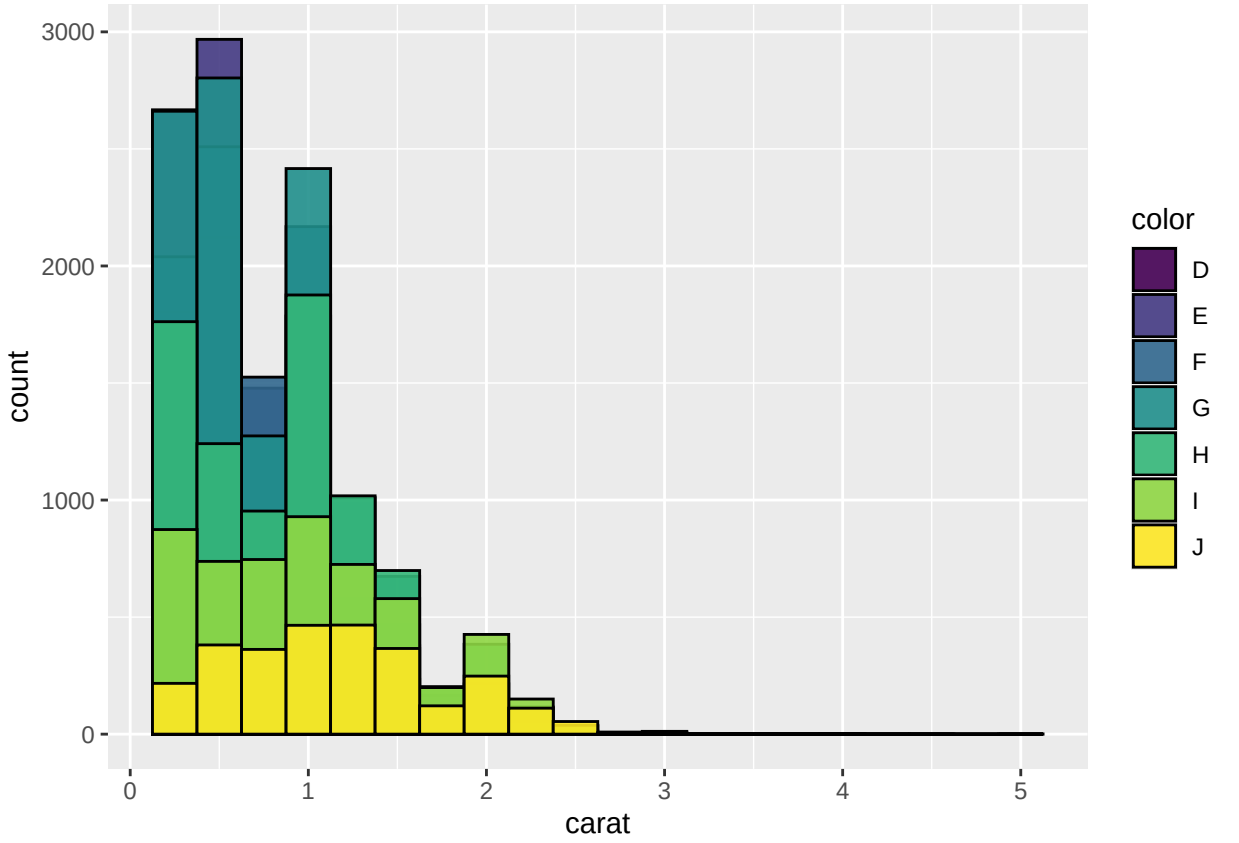


This shows that we can get the same information as a classic histogram, with a different geometry. Here, using points gives us a better view of the distribution.

3. To your original histogram (1.) add an aesthetic mapping from one of the factor variables to the fill or color aesthetic.

Let's add the color values as an aesthetic mapping, specifically for the color aesthetic.

```
diamonds %>% ggplot(aes(x=carat, fill=color)) +
  layer(
    geom = "bar",
    stat = "bin",
    position = "identity",
    params = list(
      binwidth = 0.25,
      color = "black",
      alpha = 0.9
    )
  )
```



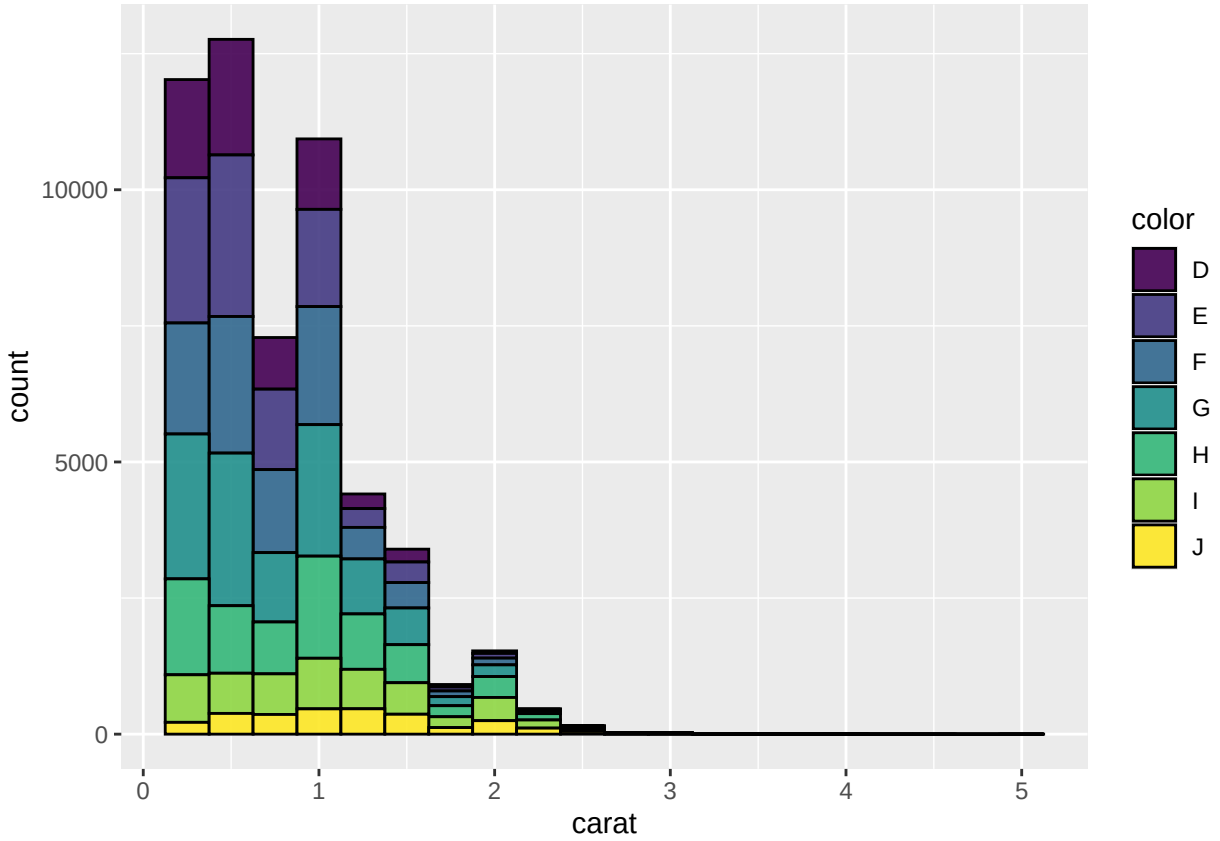
Notice how the y-axis of this histogram is now less than our first histogram. This is because each subgroup, our diamond's color, is layered on top of each other. For example, there are around 400 to 500 diamonds of carat 1 with a color of J, while there are around 900 diamonds of carat 1 with a color of I.

When the identity position is used, the subgroup with the least count (or least value in the y-axis) is at the front, while the subgroup with the most count is layered at the back.

4. Changing the position parameter in the layer function

We have the "identity" position as the default, but let's find out what will happen if we change that. Let's try nudge.

```
diamonds %>% ggplot(aes(x=carat, fill=color)) +
  layer(
    geom = "bar",
    stat = "bin",
    position = "stack",
    params = list(
      binwidth = 0.25,
      color = "black",
      alpha = 0.9
    )
  )
```



After stacking them, we see the same y-axis values from this graph and our very first histogram. This means that the subgroups of color are stacked against each other when using ' position = "stack." ' This is in contrast to when the position is set to identity, which layers the subgroups on top of each other rather than stacking them.