FA3_EDA

ABLIAN

2025-02-27

```r
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

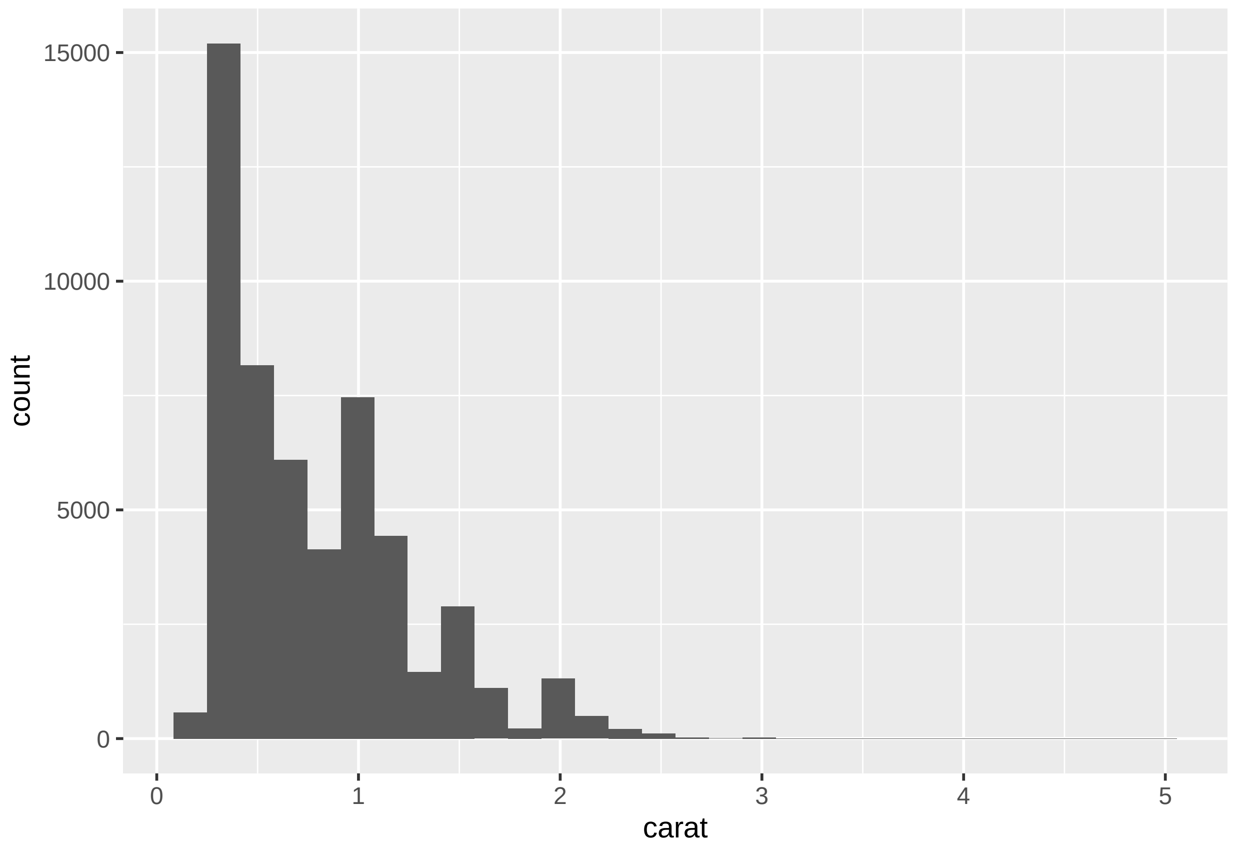
```r
library("ggplot2")
data("diamonds")
```

```r


histo_layer <- ggplot(diamonds, aes(x = carat)) +
  geom_bar(stat = "bin", position = "stack", aes(y = after_stat(count)))
print(histo_layer)
```

Create a histogram on the diamonds dataset, for example with ggplot() + geom_histogram(aes(x = carat), data = diamonds)

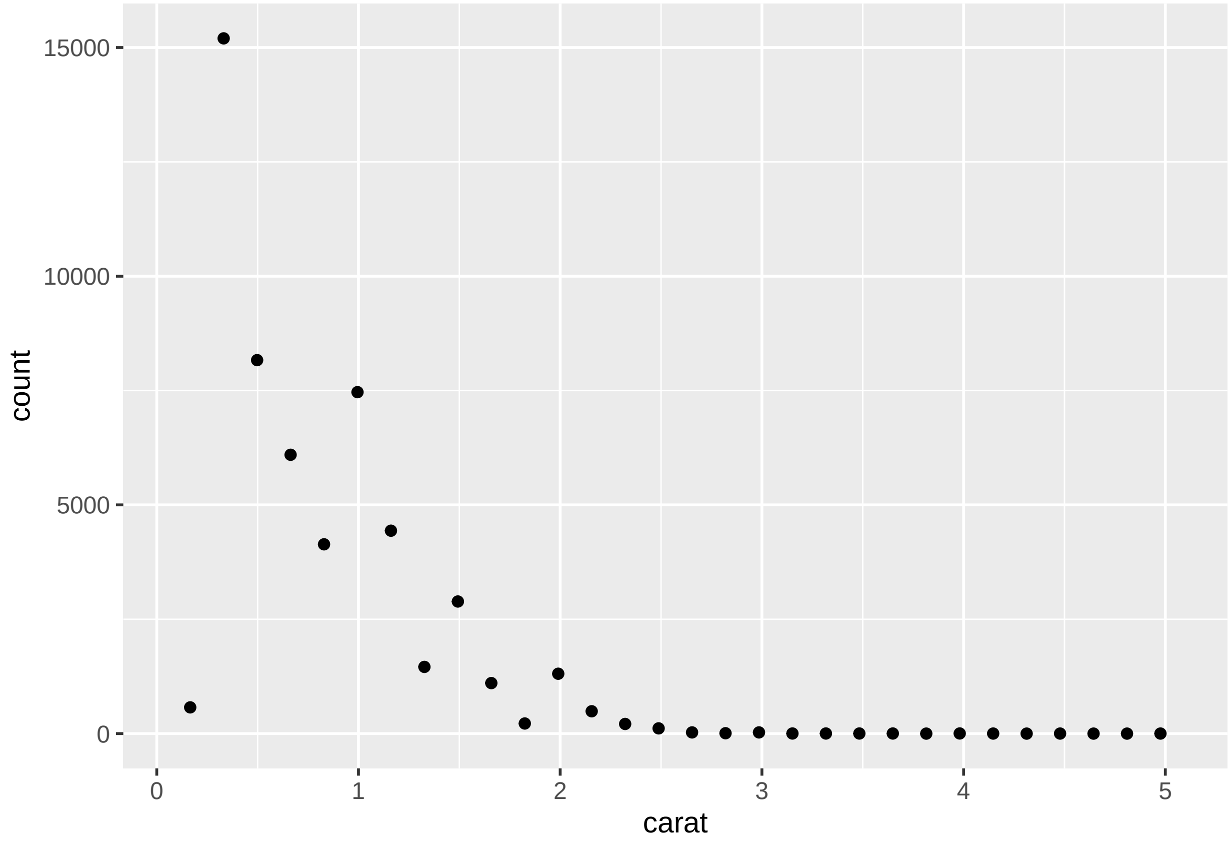
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



Remember that a histogram is a plot with stat_bin and geom_bar. Modify your histogram code so that it uses a different geom, for example geom_line or geom_point. This should be simple once you have the layer specification of a histogram.

```
histo_layer2 <- ggplot(diamonds, aes(x = carat)) +
  geom_point(stat = "bin", position = "identity", aes(y = after_stat(count)))
print(histo_layer2)
```

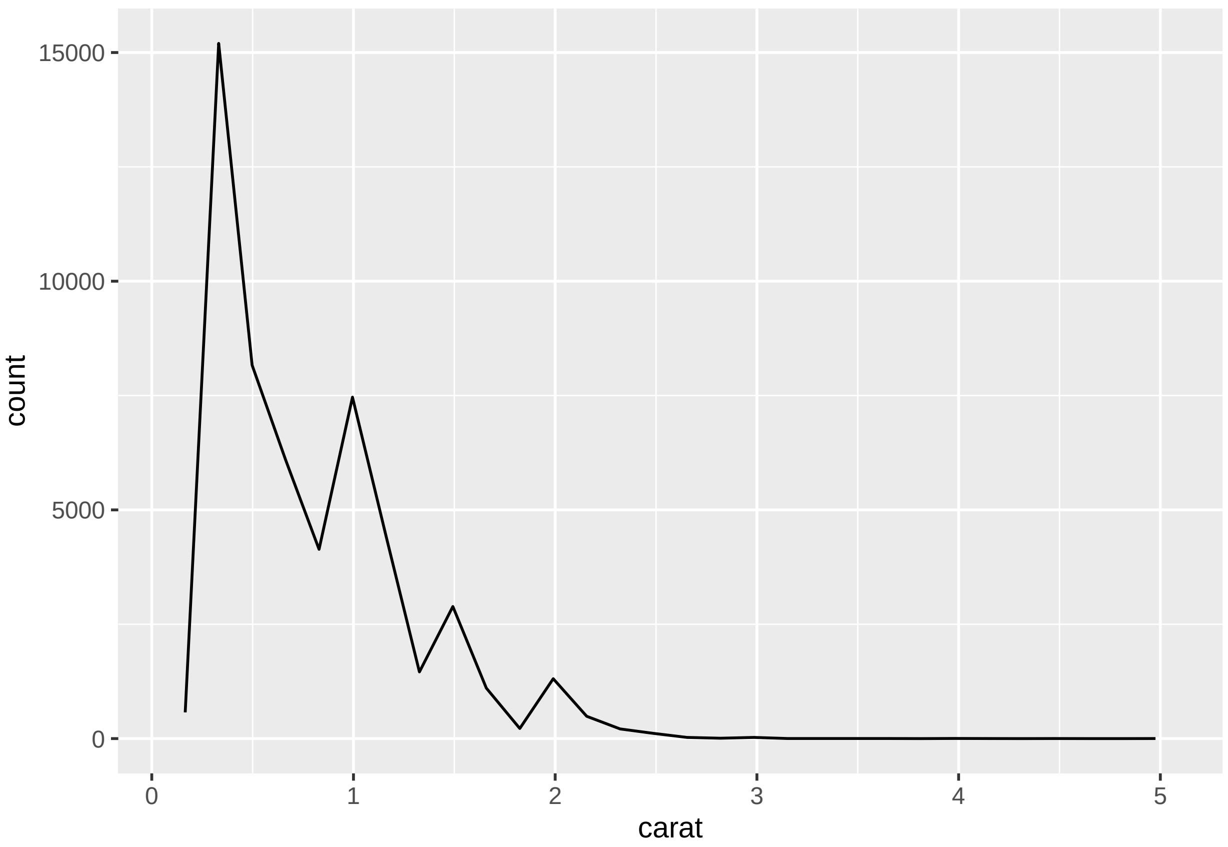
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



We can also use lines for visualization.

```
histo_layer3 <- ggplot(diamonds, aes(x = carat)) +
  geom_line(stat = "bin", position = "identity", aes(y = after_stat(count)))
print(histo_layer3)
```

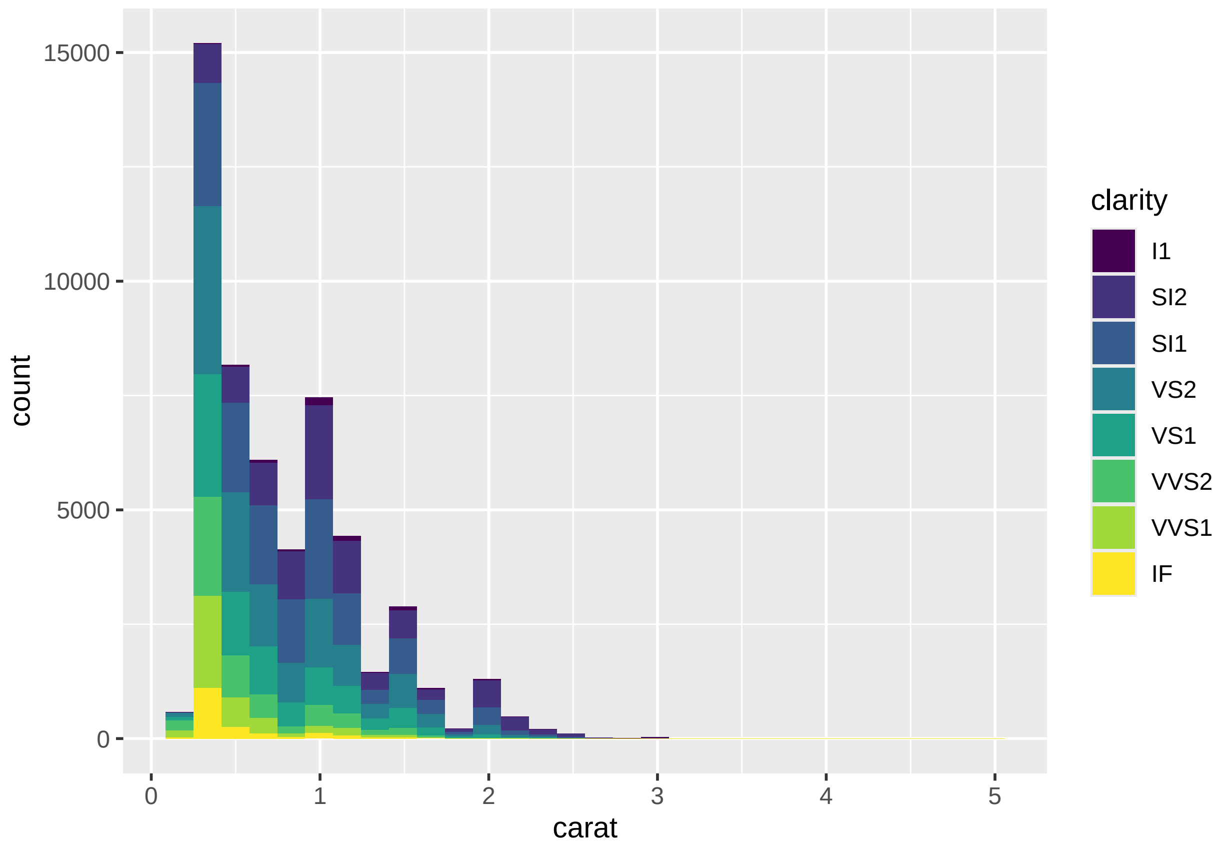
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



In your histogram (the one plotted with bars that you created in question 1), add an aesthetic mapping from one of the factor variables (maybe color or clarity) to the fill or color aesthetic.

```
histo_layer4 <- ggplot(diamonds, aes(x = carat, fill = clarity)) +
  geom_bar(stat = "bin", position = "stack", aes(y = after_stat(count)))
print(histo_layer4)
```

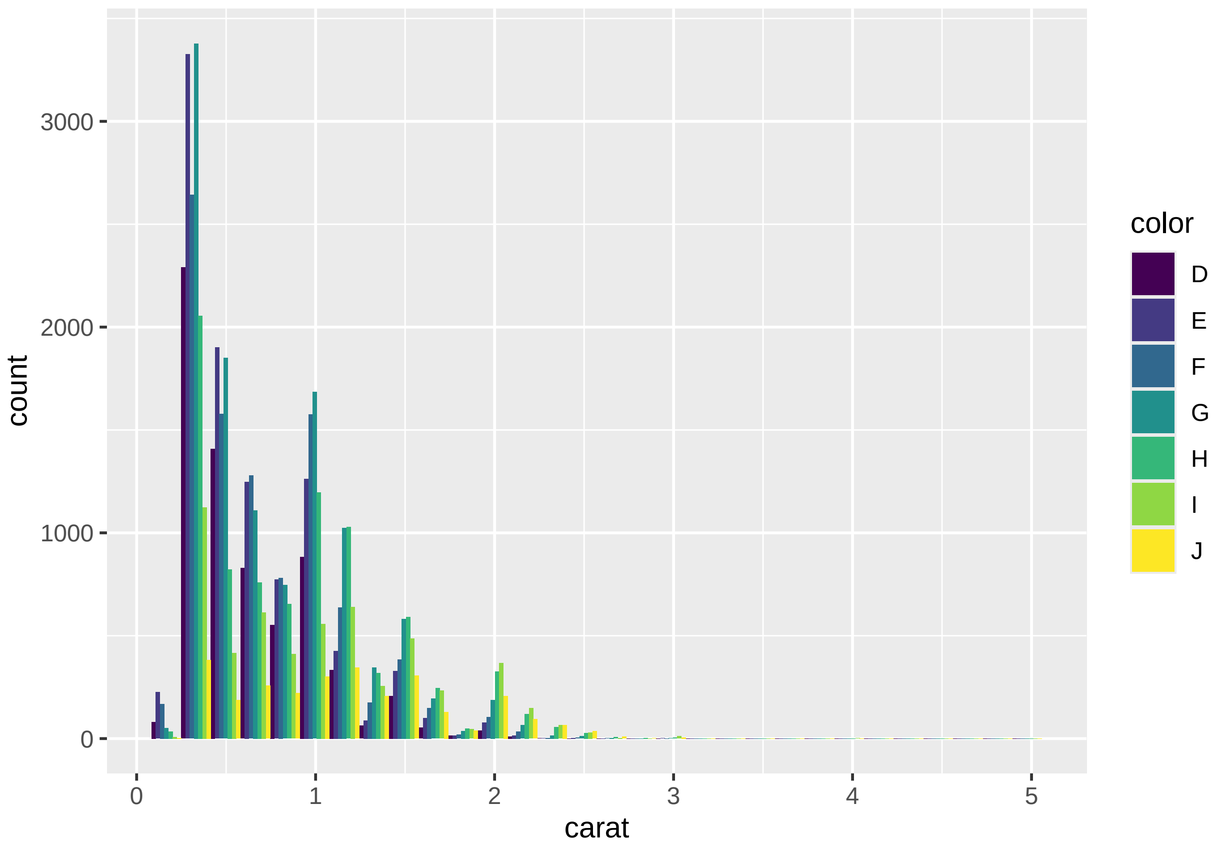
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



What is the default position adjustment for a histogram? Try changing the position adjustment in the histogram you created in question 3 to something different (hint: try dodge).

```
histo_layer5 <- ggplot(diamonds, aes(x = carat, fill = color)) +
  geom_bar(stat = "bin", position = "dodge", aes(y = after_stat(count)))
print(histo_layer5)
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```


3000
2000
1000
0
count
0
1
2
3
4
5
carat
color
D
E
F
G
H
I
J