**3.7.1 Exercises**

1. What is the default geom associated with stat\_summary()? How could you rewrite the previous plot to use that geom function instead of the stat function?

# The stat\_summary is associated with geom\_pointrange

ggplot(data = diamonds) +

geom\_pointrange(

mapping = aes(x = cut, y = depth),

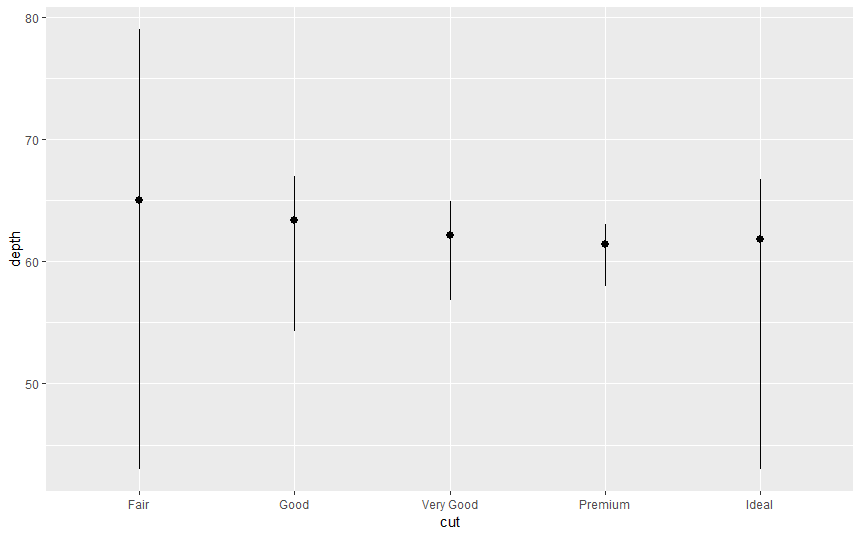
stat = "summary",

fun.ymin = min,

fun.ymax = max,

fun.y = median

)



1. What does geom\_col() do? How is it different to geom\_bar()?

geom\_bar makes the height of the bar proportional to the number of cases in each group while [geom\_col](http://127.0.0.1:39984/help/library/ggplot2/help/geom_col) makes the heights of the bars to represent values in the data.

The geom\_col () function has different default stat than geom\_bar() . The default stat of geom\_col() is stat\_identity(), which leaves the data as is. The geom\_col function expects that the data contains x values and y values which represent the bar height.

The default stat of geom\_bar is stat\_bin. The geom\_bar() function only expects an x variable.The stat, stat\_bin(), preprocess input data counting the number of observations for each value of x. The y aesthetic uses the values of these counts.

1. Most geoms and stats come in pairs that are almost always used in concert. Read through the documentation and make a list of all the pairs. What do they have in common?

|  |  |
| --- | --- |
| Geom | Stat |
| `geom\_bar()` | `stat\_count()` |
| `geom\_bin2d()` | `stat\_bin\_2d()` |
| `geom\_boxplot()` | `stat\_boxplot()` |
| `geom\_contour()` | `stat\_contour()` |
| `geom\_count()` | `stat\_sum() |
| `geom\_density()` | `stat\_density()` |
| `geom\_density\_2d()` | `stat\_density\_2d()` |
| `geom\_hex()` | `stat\_hex()` |
| `geom\_freqpoly()` | `stat\_bin()` |
| `geom\_histogram()` | `stat\_bin()` |
| `geom\_qq\_line()` | `stat\_qq\_line()` |
| `geom\_qq()` | `stat\_qq()` |
| `geom\_quantile()` | `stat\_quantile()` |
| `geom\_smooth()` | `stat\_smooth()` |
| `geom\_violin()` | `stat\_violin()` |
| `geom\_sf()` | `stat\_sf()` |

1. What variables does stat\_smooth() compute? What parameters control its behaviour?

The function `stat\_smooth()` calculates the following variables:

- `y`: predicted value

- `ymin`: lower value of the confidence interval

- `ymax`: upper value of the confidence interval

- `se`: standard error

The "Computed Variables" section of the `stat\_smooth()` documentation contains these variables.

The parameters that control the behavior of `stat\_smooth()` include

- `method`: the method used to

- `formula`: the formula are parameters such as `method` which determines which method is used to calculate the predictions and confidence interval, and some other arguments that are passed to that.

- `na.rm`:

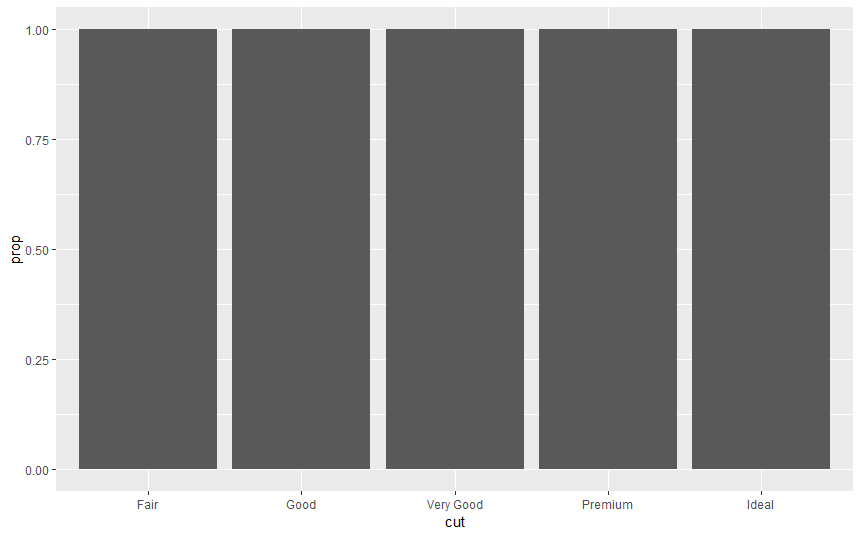
1. In our proportion bar chart, we need to set group = 1. Why? In other words what is the problem with these two graphs?

If `group = 1` is not included, then all the bars in the plot will have the same height, a height of 1.

The function `geom\_bar()` assumes that the groups are equal to the `x` values since the stat computes the counts within the group.

**ggplot**(data = diamonds) +

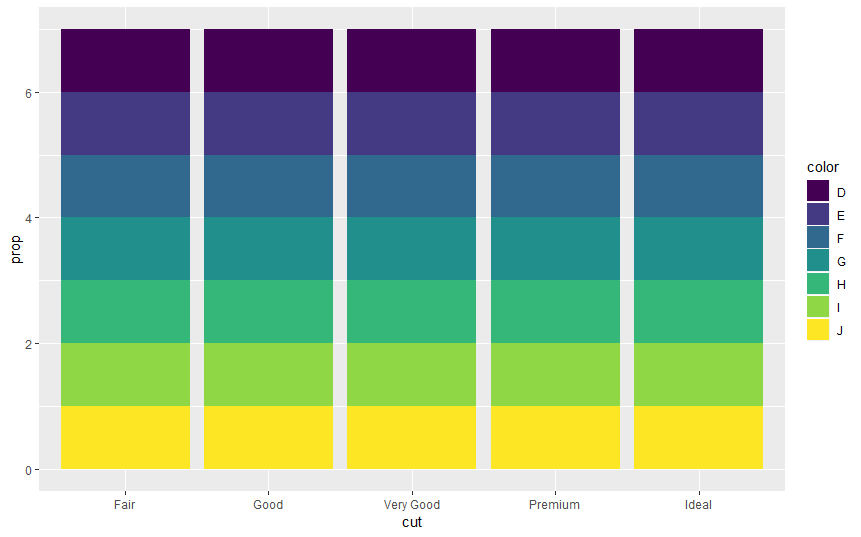
**geom\_bar**(mapping = **aes**(x = cut, y = ..prop..))



The problem with these two plots is that the proportions are calculated within the groups.

**ggplot**(data = diamonds) +

**geom\_bar**(mapping = **aes**(x = cut, fill = color, y = ..prop..))

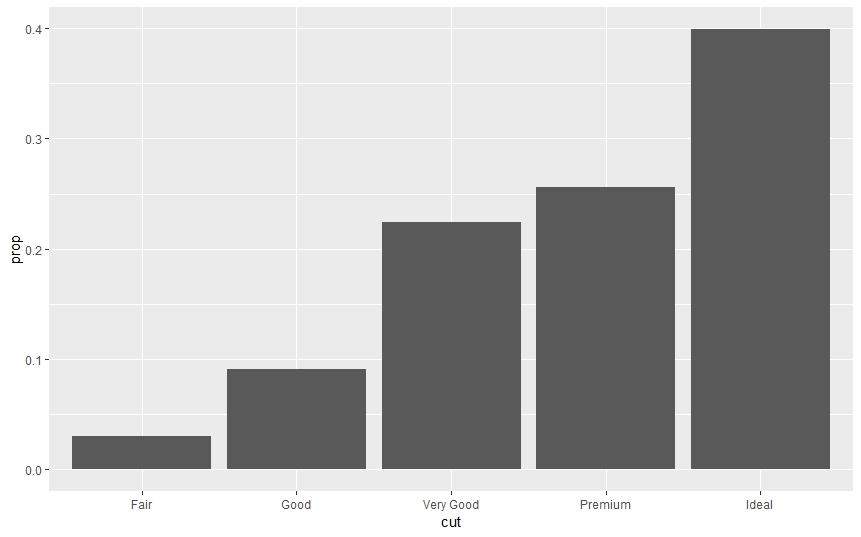


The following code will produce the intended stacked bar charts.

With group

ggplot(data = diamonds) +

geom\_bar(mapping = aes(x = cut, y = ..prop.., group = 1))



ggplot(data = diamonds) +

geom\_bar(mapping = aes(x = cut, fill = color, y = ..prop.., group = color))

