

# r4ds Ex 3.6.1

MW

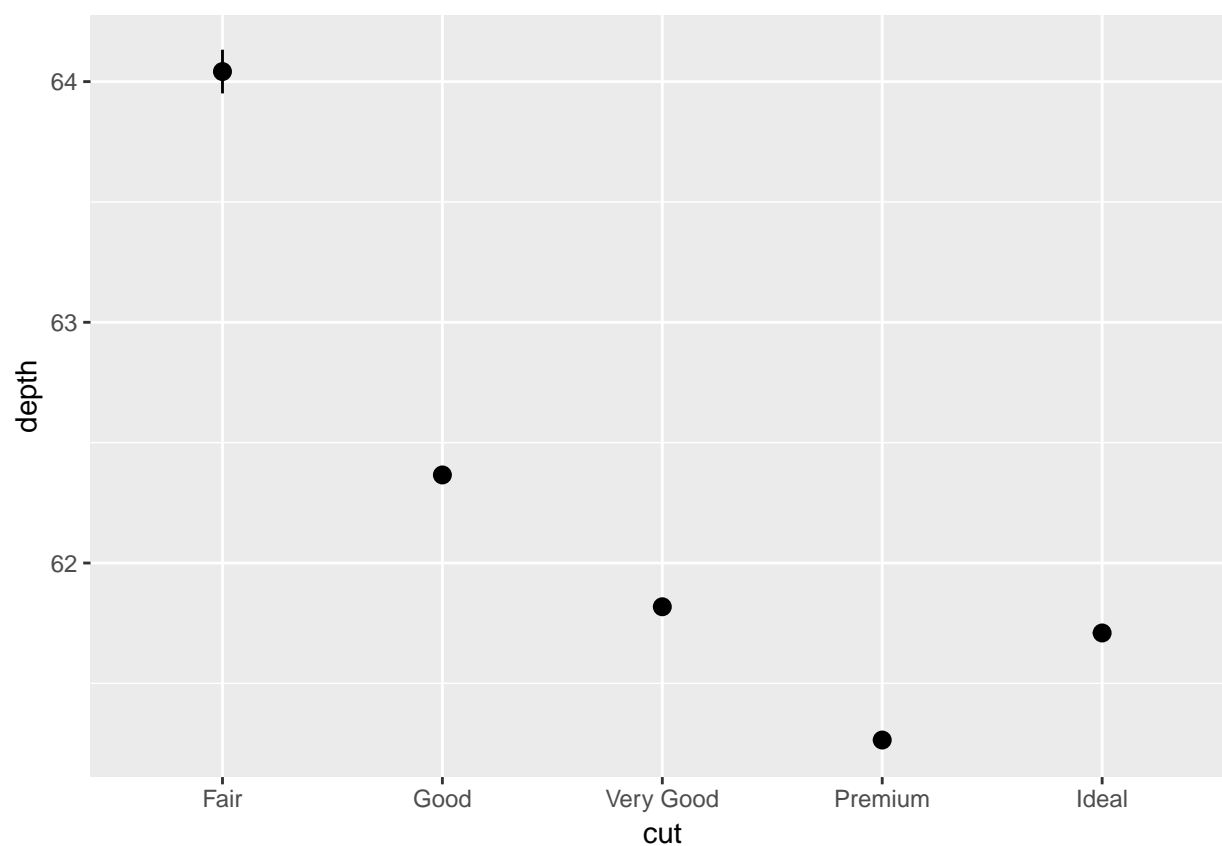
2019/05/15

## 3.7.1

1

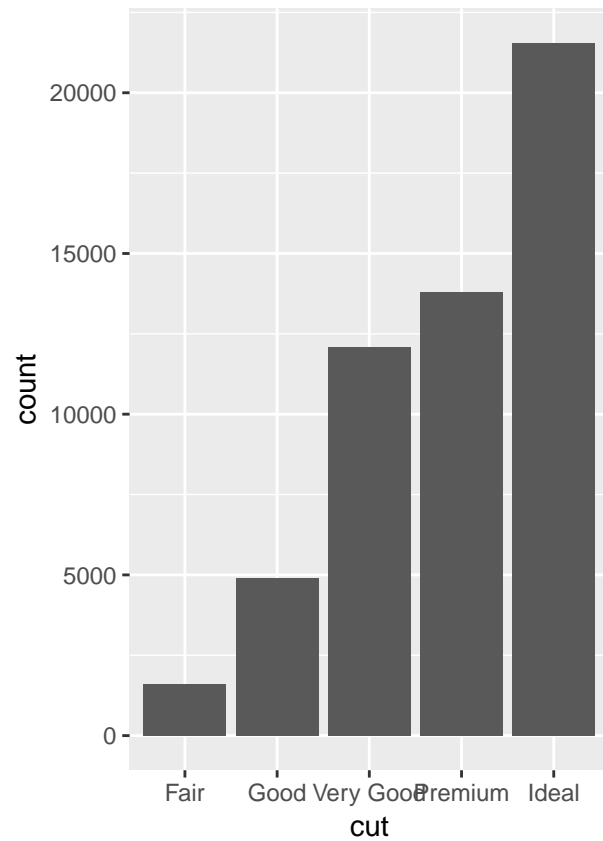
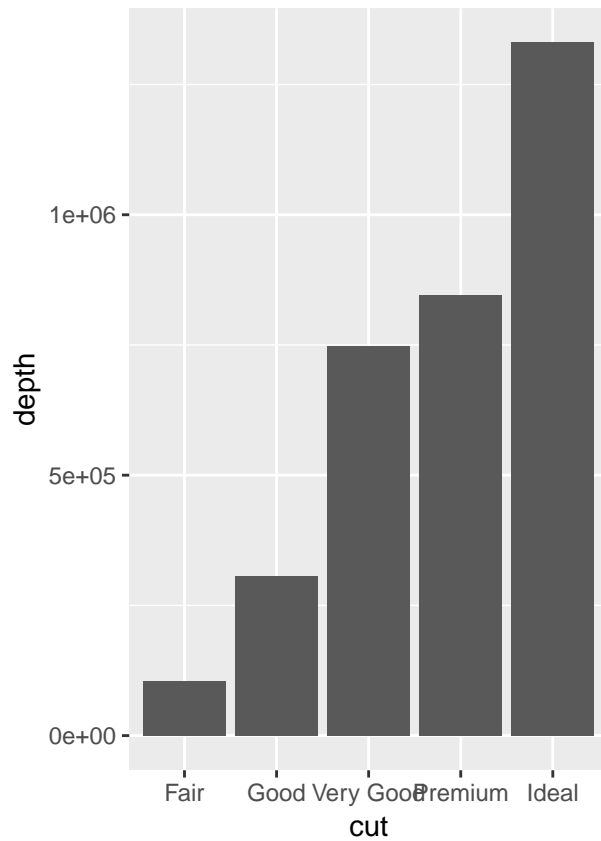
```
ggplot(data=diamonds) +  
  geom_pointrange(aes(x=cut, y=depth), stat="summary")
```

## No summary function supplied, defaulting to `mean\_se()`



2

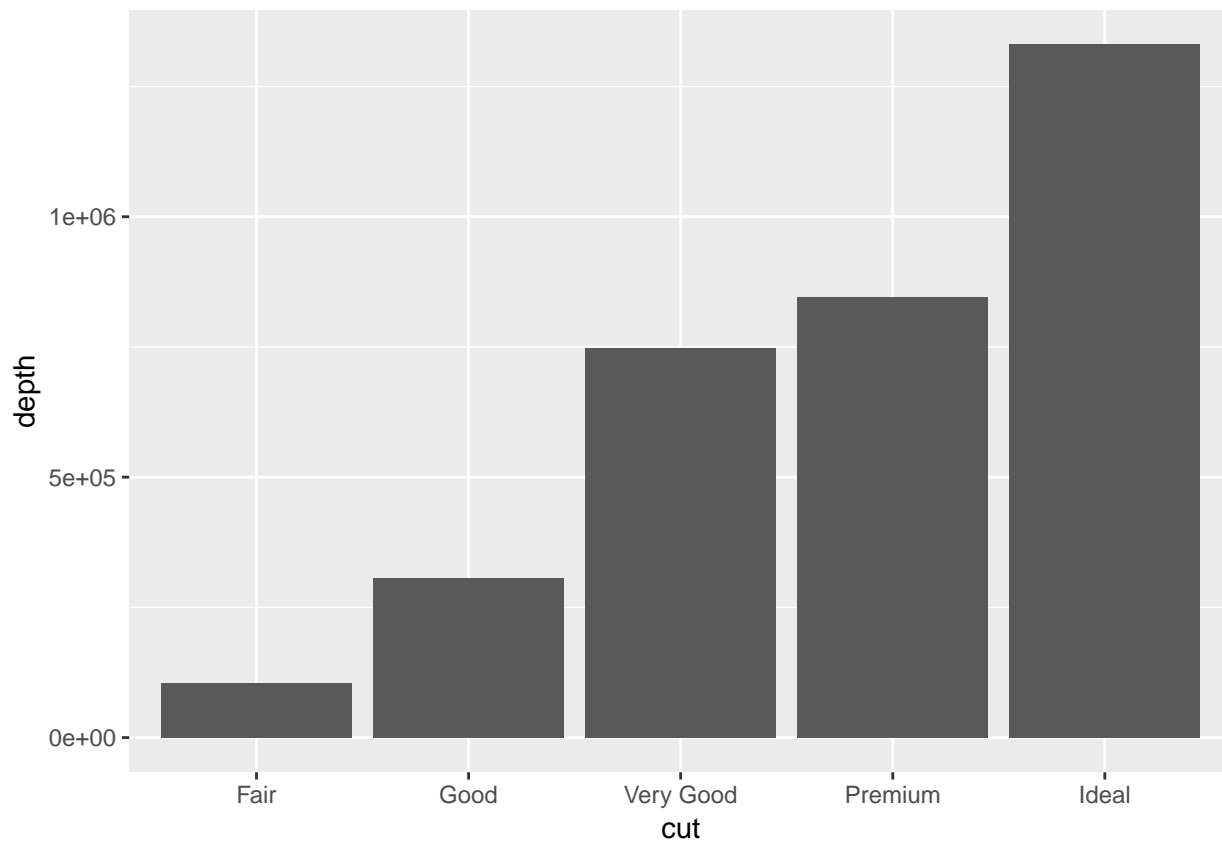
```
library(gridExtra)  
a <- ggplot(data=diamonds) +  
  geom_col(aes(x=cut, y=depth))  
  
b <- ggplot(data=diamonds) +  
  geom_bar(aes(x=cut))  
  
grid.arrange(a, b, nrow = 1)
```



`geom_col` uses `stat_identity`, so it requires `x` and `y` in aesthetics. `geom_bar` uses `stat_count` by default, so it has possibility to change the `y` by specifying `stat`.

Example is as follows:

```
ggplot(data=diamonds) +  
  geom_bar(aes(x=cut, y=depth), stat="identity")
```



### 3

Correspondence list of `geom_` and `stat_`. - `geom_bar` `stat_count` - `geom_contour` `stat_contour`  
- `geom_count` `stat_sum` - `geom_density` `stat_density` - `geom_histogram` `stat_bin` - `geom_qq_line`  
`stat_qq_line` - `geom_qq` `stat_qq` - `geom_smooth` `stat_smooth` - `geom_violin` `stat_violin` - `geom_bin2d`  
`stat_bin_2d` - `geom_boxplot` `stat_boxplot` - `geom_density_2d` `stat_density_2d` - `geom_hex` `stat_hex` -  
`geom_freqpoly` `stat_bin` - `geom_quantile` `stat_quantile` - `geom_sf` `stat_sf`

### 4

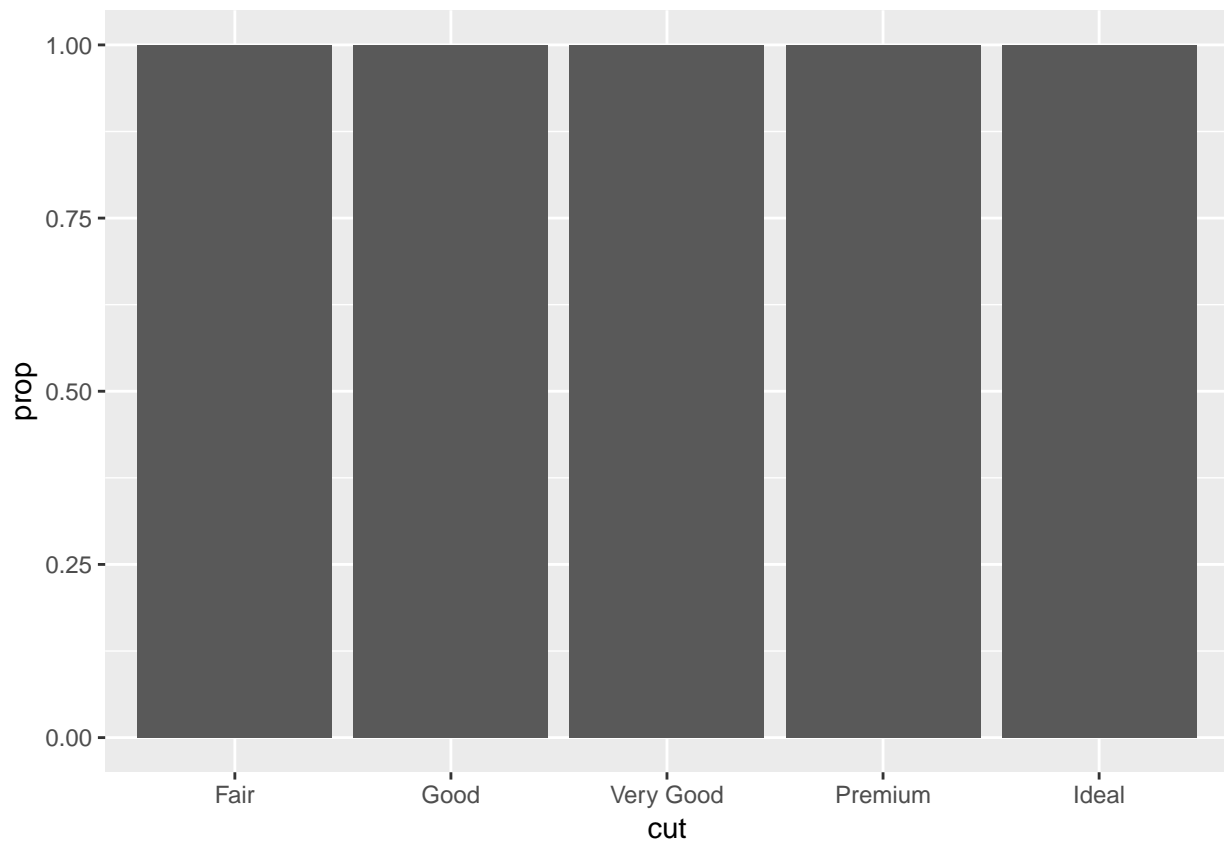
`stat_smooth`

- `y`: predicted value
- `ymin`: lower value of the confidence interval
- `ymax`: upper value of the confidence interval
- `se`: standard error

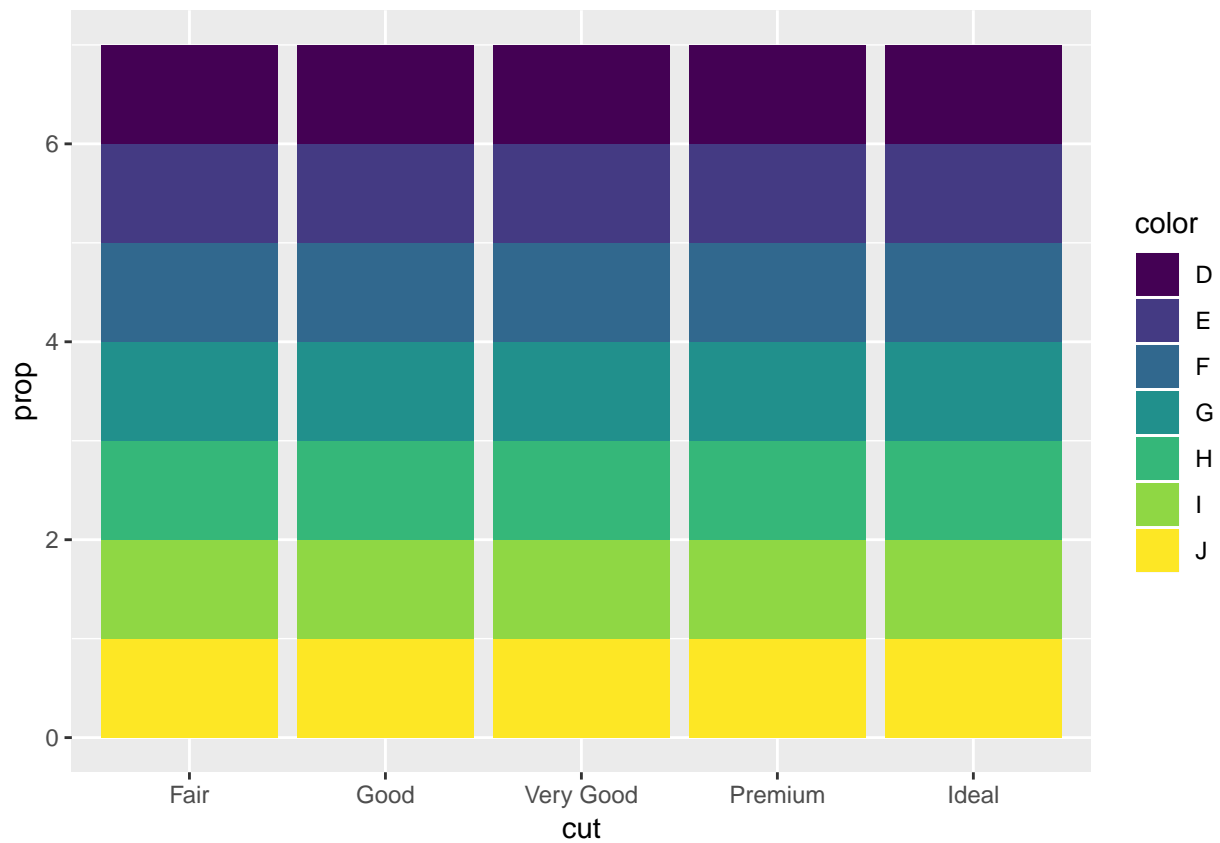
The list of parameters that control the behavior is as follows: - `method` - `formula` - `se` - `na.rm` - `show.legend`  
- `inherit.aes` - `n` - `span` - `fullrange` - `level` - `method.args`

### 5

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = ..prop..))
```

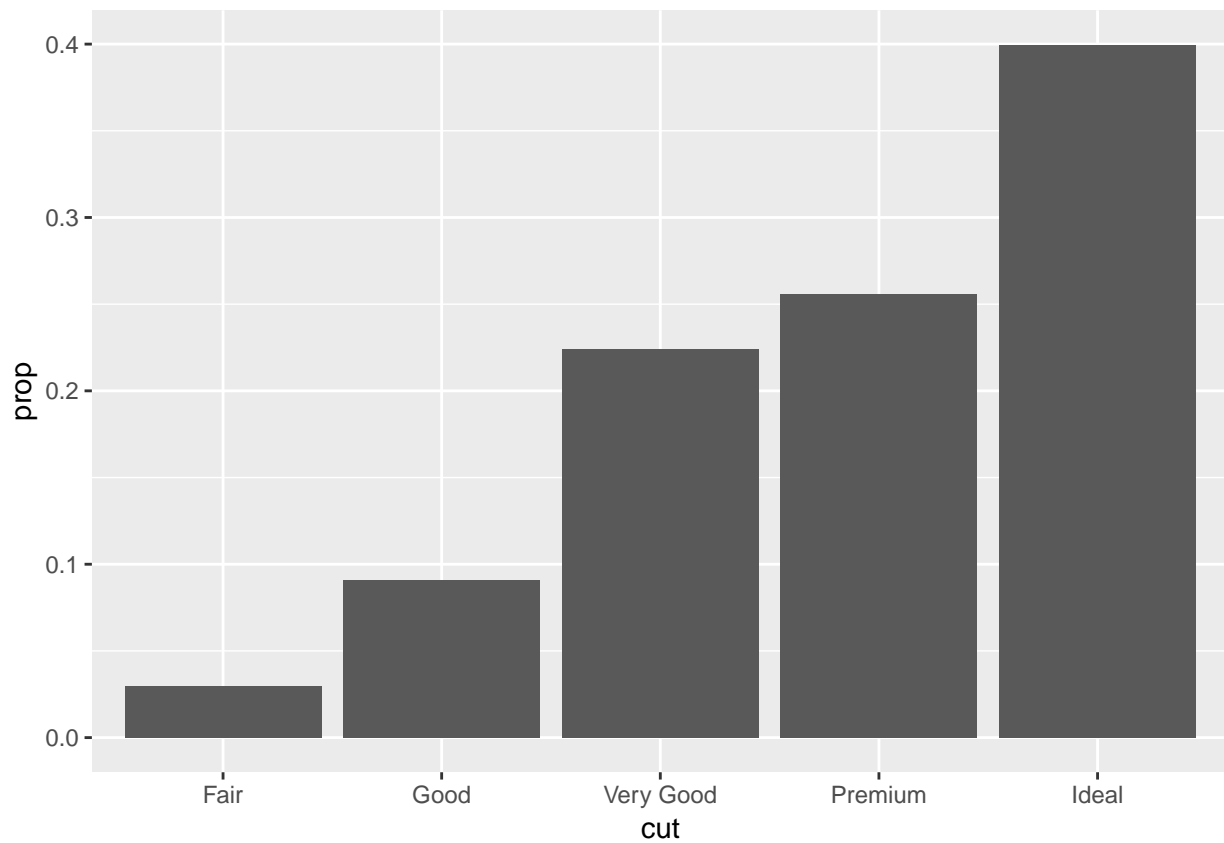


```
ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = color, y = ..prop..))
```



The problem is that the ratio calculation is done within the group. Solutions;

```
ggplot(data=diamonds) +  
  geom_bar(aes(x=cut, y=..prop.., group=1))
```



```
diamonds %>%  
  group_by(cut) %>% nest() %>%  
  inner_join(count(diamonds %>% group_by(cut))) %>%  
  mutate(s=.$n %>% sum) %>% unnest() %>%  
  ggplot(aes(x=cut, y=n/s, fill=color)) +  
    geom_bar(stat="identity")
```

```
## Joining, by = "cut"
```

