

Histogram gaps and spikes in digital photography

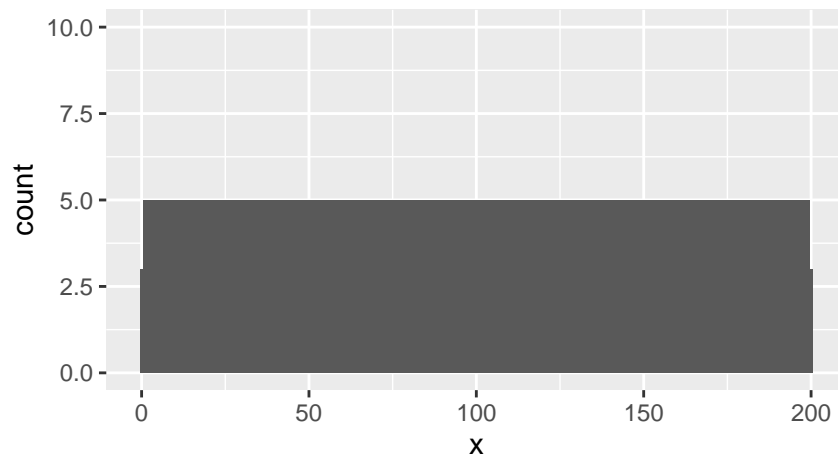
Michael A. Covington

2023 November 26

Some digital cameras scale their digital output more than once before delivering it to you. This results in gaps or spikes in the histogram, especially if the scale factor is close to 1. Here are examples showing how this happens.

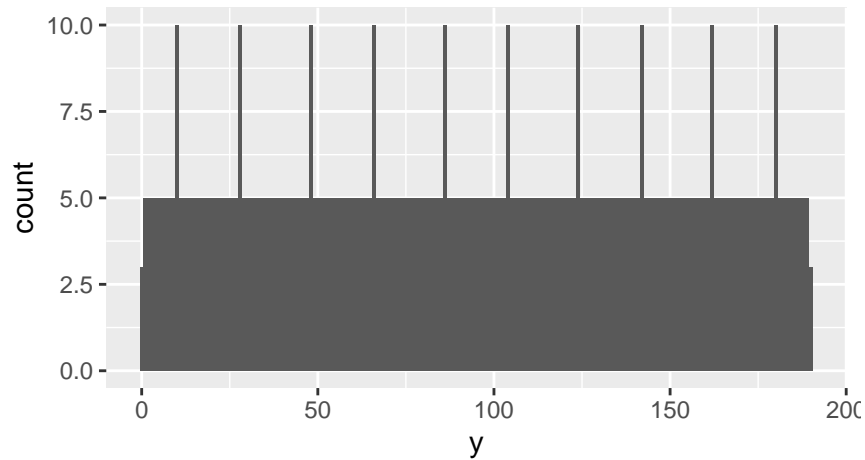
Let's start with a smooth distribution of random floating-point numbers, then round each of them to the nearest integer. The histogram comes out smooth, of course.

```
x = round(seq(0,200,0.2))  
ggplot(data.frame(x),aes(x)) + geom_histogram(binwidth=1) + ylim(c(0,10))
```



Now multiply each of those integers by 0.95 and round to the nearest integer again. You get spikes:

```
y = round(x*0.95)
ggplot(data.frame(y),aes(y)) + geom_histogram(binwidth=1) + ylim(c(0,10))
```



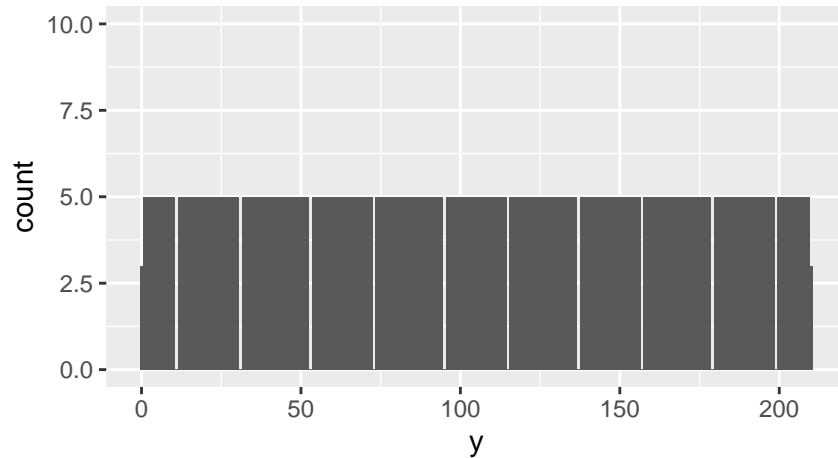
The reason is that some integers occur more than once. An example:

Original value	$\times 0.95$	Rounded
26	24.70	25
27	25.65	26
28	26.60	27
29	27.55	28
30	28.50	29
31	29.45	29
32	30.40	30

There is a spike at 29 because there are two ways to get 29. (Actually, with R's rounding rules, 28.50 rounds down, and there is a spike at 28.)

If we scale up rather than down, we get gaps in the histogram:

```
y = round(x*1.05)
ggplot(data.frame(y), aes(y)) + geom_histogram(binwidth=1) + ylim(c(0,10))
```



Now there are integers that never occur because adjacent integers in the input round to integers that are not adjacent in the output. Gaps are bothersome because they cause jumps (posterization) in gradients in underexposed areas of the picture.

I am grateful to Mark Shelley, who pointed out this effect to me in this discussion on Cloudy Nights Forums: <https://www.cloudynights.com/topic/896181-canon-eos-r-iso-1600-colour-gradients-and-rings/>