Note on “normalizing bins” in histograms.

(my thoughts after feedback from Ilya Mandel)

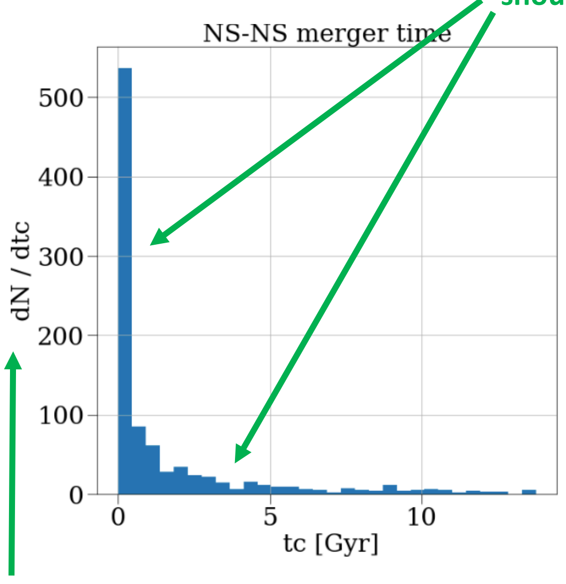
There are different ways of plotting a histogram. Here’s a quick note on “how to scale the bin widths of the histogram bins”, and how this affects your plot. I wrote it after spending 2 hours trying to figure out why my distribution looks different from a result in Alejandro’s paper.. Finding out later that it’s because of the way the histogram is plotted.

In all the plots, I’m looking at the distribution of the coalescence time, tc, of NS-NS systems. (so how long it takes from the formation of the NS-NS to spiral in.

If you would plot the distribution of the coalescence time, tc, it would look like this:

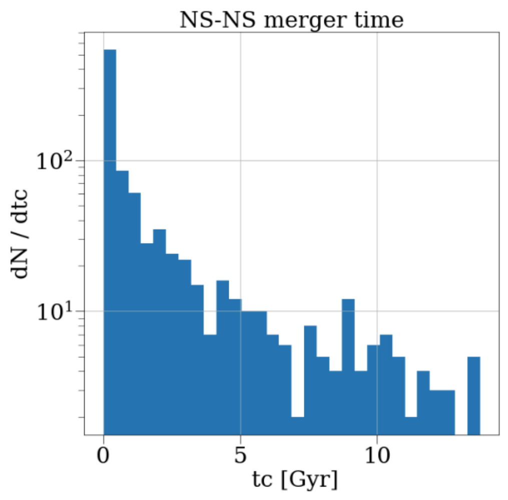
1) np.histogram(tc)

y-axis value changes a lot. So maybe we should plot log-scale



Don’t use just “number of systems” since the y-axis is number of systems per ”bin-interval” which is given by dtc (here the bin size dtc is ~ 0.5Gyr)

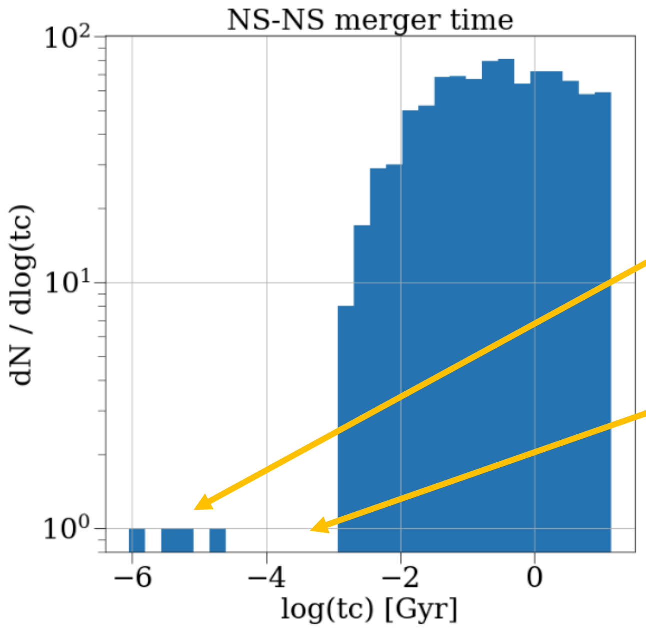
2) np.histogram(tc) + plt.yscale(‘log’)



The y-scale is now in log. The total number of systems that are plotted can still be obtained by adding all the y-value of each bin. To get the number of systems with short merger time (say tc< 1Gyrs) you add all the dN/dtc of all the bins within tc < 1Gyr.

For cosmology interpertations, often a tc of 1 Gyr or 10 Gyr is both interpertated as “long”. We might therefore be interested in the distribution of log(tc) instead to zoom in on the small tc values.

3) np.histogram(np.log(tc)) + plt.yscale(‘log’)



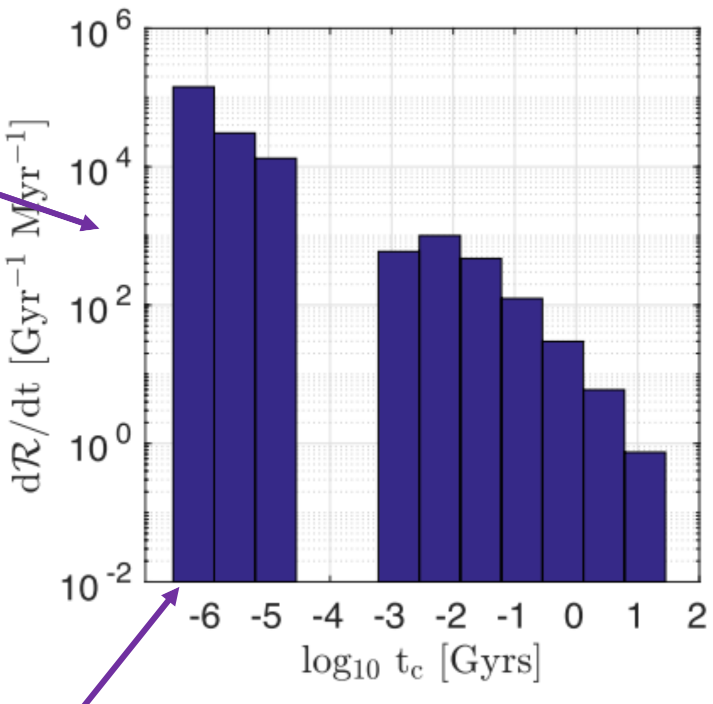
Now some interesting things start to be visible: we have ~4 systems with merger times of log(tc) < -4

The fact that we have a gap around log(tc) = -4, is from Poisson noise in the sampling.

We now basically do a change of variable from tc -> log(tc). The bins are also dlog(tc) ~ 0.4 Gyr and have fixed width in log(tc). It is important to put the dlog(tc) on the y-axis!!

4) My confusion + perhaps slightly misleading plot:

This plot was made with the exact same data for tc as the plots above, and is shown in Vigna-Gomez+2018. I hadn’t realized until today that I always mis-interpertated this distribution.



They plot the log(tc) on the x-axis, but the bins (and y-axis) are dN / dtc (not dlog(tc). In other words. Even though the bins look the same size (in log(tc), this is not true since they are defined in log(tc). Therefore the bin in log(tc)~-6 is something like [1*10^-6 – 5*10^-6] whereas the bin around log(tc) ~ 1 = [1,1.5]. So more than 6 orders of magnitude difference in the binsizes!

This is why the bins around log(tc) ~ -6 look so high. They are the same “4 samples”, but now to get the number of events in these bins , you have to do ~10^5 *10^-6 = .1 (for the lowest bin) and ~10^0~10^1 = 10 for the highest . Which gives back the rates. PS: I find this representation misleading because it now does look as if there are many more samples around log(tc) ~ -6 and therefore the gap looks physical.
(note that on the yaxis R = my N, but multiplied by SFR etc. ..)

Anyways, Fig 4 is not wrong. But hopefully this shows that histograms can be misleading. And that you should always quote on the y-axis in what unit the bins are (tc vs log(tc)) (which they luckily do in Fig 4)

Personally, I prefer that the binsize is scaled similar to the x-axis (so Fig. 3 instead of Fig 4)