7.5 V_{max} method

Some test data for this exercise are provided – this is the simulated data that was used for the Figure 7.8 in the text.
The file contains data for 194 "galaxies" and for each galaxy are given, in order, two independent "luminosities" and a "distance". The galaxies were uniformly distributed in a sphere with radius $2^{1/3}$, and the first luminosity for each "galaxy" was selected by the criterion $L/R^2 \geq 20$; this gives the flux limit for the sample. The second luminosity has not been selected on the basis of any flux limit, and will not be used in this example. The univariate luminosity functions in the Chapter were built from the first luminosity of each triple of numbers.
Here are our results for the 194 galaxies, as numbers. The luminosity function ρ has been normalized to unity, so it is like a probability density.

log power	ρ
−0.75	0.327
−0.25	0.336
0.25	0.145
0.75	0.132
1.25	0.058
1.75	0.00252
2.25	0.000103

Try to set up your own simulation.
The luminosities were drawn from a Schechter function, starting at a lower limit of 1/10 units and with a characteristic luminosity of 10 units. The index was just 1. This is rather fiddly to do, as there isn't an analytic form for the integral of the Schechter function (unless your computer is happy with the exponential integral).
In setting up these simulations, it is important to be sure that reasonably common bright galaxies do not fall outside the simulated volume at the flux limit. In this case, galaxies of luminosity $202^{2/3}$ and greater will be artificially under-represented. On the other hand, to get the 194 galaxies in the sample, 5000 had to be created.
These test data will allow you to check one realization of V/ V_{max}. A Monte Carlo simulation will require engagement with the aforementioned Schechter functions. We found it easiest to fit a polynomial to an integration of the Schechter function, and then obtain the inverse of this cumulative distribution by a simple Newton-Raphson search or similar. Having to find the inverse function numerically does slow down the generation of random numbers quite a lot.