Distribution Tests

16

We can relate the VaR model's prediction (tomorrow's PnL distribution) with actual, later outcomes (the realized and experienced PnL) in a way that is more expressive than the backtesting with its focus on relatively rare VaR violations. We can view the prediction—a set of 1000 equally likely PnL values—as a skeleton or proxy for the infinitely many possible PnL outcomes. More specifically, the *sorted* PnL predictions $\Delta p_{(i)}$ delimit 1001 adjacent intervals into one of which the actual PnL must later perforce fall:

$$(-\infty, \Delta p_{(1)}], (\Delta p_{(1)}, \Delta p_{(2)}], \ldots, (\Delta p_{(1000)}, \infty).$$

To depict these interval boundaries, we simply reprise a graph we've encountered at the beginning of the book in Fig. 16.1.

Note how the left- and right-most intervals tend to be larger than the central ones, as the predicted PnL values or interval boundaries are typically spaced further apart at the fringes. In a leap of faith, we declare that the probabilities for an actual PnL to fall in any of those intervals—if the model is correct—should be just about the same (in this case, 1/1001). Intuitively, dense regions represent areas where we predict the actual PnL to materialize more often—but the intervals in those regions are also narrower, which keeps their hit probability at bay. Conversely, sparse regions indicate areas where we generally expect few PnLs to happen—but if a stray PnL should ever appear nearby, the larger intervals there can scoop it up more easily. In both cases, hit frequencies and interval sizes offset each other, yielding uniform probabilities for the various intervals.

We can have a go at this from another direction: the VaR, i.e., the 1%-quantile, delimits the 10 most negative PnLs, and the probability of hitting that area is 1%. The 2%-quantile fences in the 20 most negative PnLs; the probability of hitting that area is 2%. Consequently, the probability of falling *between* those two

M. Auer, *Hands-On Value-at-Risk and Expected Shortfall*, Management for Professionals, https://doi.org/10.1007/978-3-319-72320-4_16

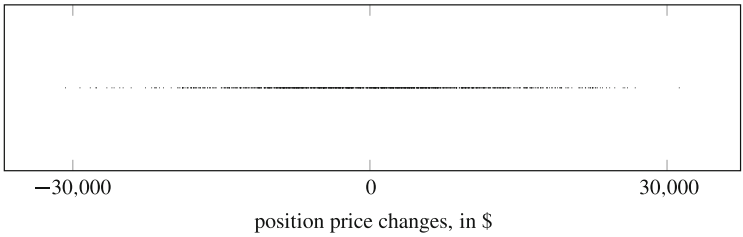

Fig. 16.1 Price changes as interval boundaries

quantiles—the 10th and 20th most negative PnLs—must again be 1%. The reasoning above just breaks this down to a more granular level.[1]

We can proceed to a fairly straightforward test setup. Each day, an actual PnL will fall into one of the 1001 predicted, hypothetical PnL intervals, and we can assign that interval's index to that PnL. If we furthermore divide that index by 1001, we get, for each day, a value u between 0 and 1. We can repeat that for a certain number of days—typically 40—and obtain as many u-values. As each index is equally likely, the normalized u values are—or at least should be in case of a correct model—uniformly distributed. And that's something we can test.

The following steps should help build up some intuition; you can easily retrace them using Excel's `RAND` formula, which generates a $[0, 1]$-uniform. An example of 40 such artificially generated uniforms is given in Fig. 16.2.

These values sure look random, but are they uniform? If they are, there should be some small, some medium, and some large u-values. To visually check for that, we sort them by size and graph them again in Fig. 16.3.

Clearly, sorted uniforms should form a diagonal, as they do in this example. Now, by pure chance, such uniforms might also deviate from the diagonal (the smaller the sample size, the larger the potential deviation). To give you an impression of that deviation range in our case of a 40-sized sample, and sticking with this setup of artificially created uniforms, Fig. 16.4 depicts ten such sorted uniform samples.

It turns out that this range can be described with so-called *confidence intervals* based on the beta distribution. The kth largest uniform $u_{(k)}$ of a size-n sample is

[1] The probability of a random number falling between two quantiles $q_d < q_u$ is $u-d$ (see Sect. A.7). The PnL predictions (our interval bounds) correspond to $k/1001$-quantiles; the probability of a random number falling between two adjacent ones, e.g., $d = 17/1001$ and $u = 18/1001$, is $u - d = 1/1001$.

You will notice that this treatment of quantiles is slightly different from our previous one, where we defined the 10th most negative PnL value to correspond to the 1%-quantile instead of the $10/1001 = 0.99\%$-quantile. This is largely inconsequential here due to the large number of PnL values involved, and our respective definition choices are solely motivated by convenience and concise, short form notation. In any case, empirical quantiles like these are often handled slightly differently according to a given problem at hand, for example, when it comes to dealing with the very first and last intervals or with questions of quantile interpolation.

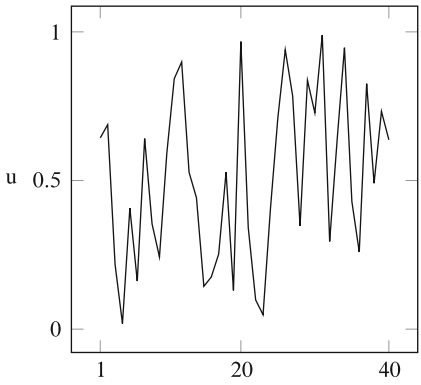

Fig. 16.2 A sample of 40 uniforms

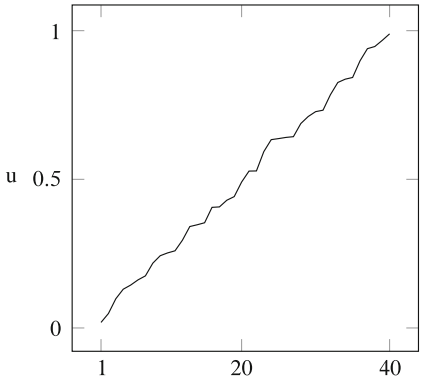

Fig. 16.3 The same sample of 40 uniforms, sorted

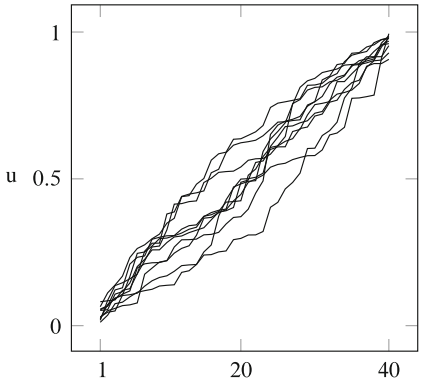

Fig. 16.4 Ten samples of 40 uniforms, sorted

Beta$(k, n + 1 - k)$-distributed, which allows us to gauge plausible ranges for those uniforms.[2]

[2]The confidence interval delimits the outcome of each u between an upper and lower bound (it is thus two-sided); it uses the 5% and the 95% quantiles of the Beta distribution for this, spanning the 90% of outcomes in between (hence it is a 90% confidence interval). You can use Excel's

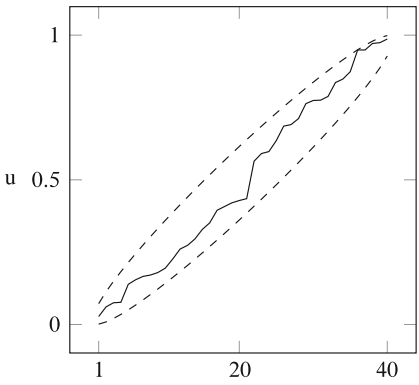

Fig. 16.5 Good distribution test

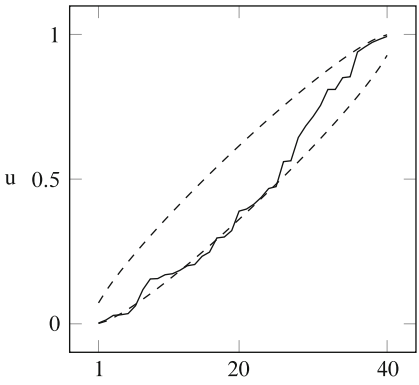

Fig. 16.6 Bad distribution test

We are now ready to look at series of normalized PnL interval indices, derived from actual PnLs versus their respective PnL distribution predictions. Here are several real-world examples, along with two-sided 90%-confidence intervals. The first example, given in Fig. 16.5, shows a portfolio for which the model works well and makes predictions within the confidence intervals.

A second example, given in Fig. 16.6, shows a model that does not seem to work that well—it seems to tend to underestimate the actual risk. Why is that so? The lower left part of the graph is the region of interest, for it roughly corresponds to negative PnL values, i.e., losses. That's because PnLs close to zero should usually hover around interval index 500 (or $u \approx 0.5$), positive PnLs or gains lie in the upper intervals above it, and negative ones or losses are below. Like for all PnLs, we expect the losses (represented by the observed u-values) to lie on the main diagonal. If they are below, it means that realized PnLs happened to fall in lower-than-expected intervals—the losses are more severe than predicted. The model thus underestimates these losses' magnitude and hereby the risk.

`BETA.INV(0.95;k;n+1-k)` and `BETA.INV(0.05;k;n+1-k)` functions to obtain each day's upper and lower bounds.

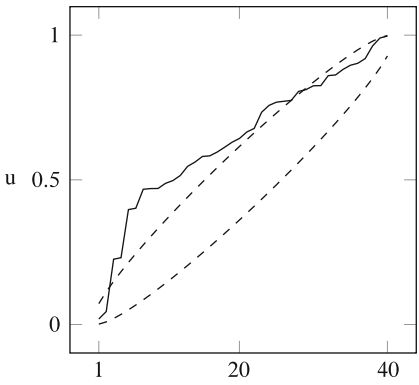

Fig. 16.7 Ugly distribution test

Finally, another example of a model that doesn't work well, albeit with less dire consequences, is given if Fig. 16.7. In this case, the model overestimates the risk and the magnitude of losses. Mathematically, this is just as poor a model as the previous one. In practice, though, this model errs on the conservative side, which for the regulator usually proves fine. Bearing the increased costs, i.e., higher capital requirements, of slightly risk-averse models may be preferable to constantly having the validity of a model hovering around the lower confidence band questioned.

Using such distribution test graphs for various portfolios is a quick and comprehensive way to check a model's quality. I personally prefer such graphical, expressive tests to the blind use of statistical tests and their resulting—mono-dimensional and highly condensed—p-values (like the ones we encountered in the backtesting chapter, supplementing the VaR/PnL time series graphs). Some don't deem such tests to be strictly quantitative—I disagree: they represent a wealth of quantitative data and do so in an interpretable context. Nonetheless, some stakeholders will insist on p-values, and they can easily be created. We can, for instance, check for the uniformity of our u-values with standard tests like Kolmogorov-Smirnow or Anderson-Darling, which readily yield those comforting numbers.[3] In my view, p-values best serve to unceremoniously seal an argument you are ready to win with a more transparent reasoning anyway, much like tossing in a PhD title into an email signature.

Distribution tests are more powerful than backtesting, in the sense that they use more information than mere binary and therefore abridged "violation yes/no" flags. They require fewer days to become meaningful, and we use 40 to 60 days as opposed to the 250 days common in backtesting. As the number of days increases,

[3]While we generally prefer Python and NumPy when doing statistics, the software suite R offers some good support for such tests. The Kolmogorov-Smirnov test is built-in (`ks.test(u, "punif")`), while Anderson-Darling is available as a separate library (called `ADGofTest`). Please note that some tests do not allow for identical u-values, which in our case can happen because of the discrete intervals. You can simply add/subtract tiny and different offsets to each u-value to disentangle them, e.g., $10^{-8}k(2\mathbb{1}_{u(k)<1/2} - 1)/n$.

the confidence bands will get narrower, and as is the case with all statistical tests, too large a sample size will eventually guarantee a p-value violation. Using 2–3 months of business days seems to be a reasonable compromise: the time series is long enough to avoid too large a leeway in terms of confidence bands, and it is short enough to not artificially cause spurious rejections due to an oversized sample.

Like some other modeling choices, the number of days to use in distribution tests is a matter of preference rather than optimality, which can lead to inquiries with remarkable follow-up explanation efforts. Defending such choices will be the topic of Sect. 17.4.