Appendix G Gehan Ranking Example

G.1 Rank for uncensored datasets

To avoid distributional assumptions, some statistical methods are based on the *ranks* of observations rather than on their measured values. Using ranks instead of measured values allows the methods to be more robust to outlying observations, which can be especially useful when working with skewed data. As a simple example, consider measured values {0.01, 2.30, 78.1, 0.14, 0.05 }. The rank of each observation is simply its position when the list is sorted from least to greatest is shown in Example 1.

Example 1 dataset with ranks

Value	0.01	0.05	0.14	2.30	78.1
Rank	1	2	3	4	5

When there are ties in the measured or observed values, and a repeated value has multiple positions in the ordered dataset, the simplest ranking scheme assigns the average position to ties. In the example shown in below, the average of positions 1 and 2, namely 1.5, is assigned as the rank for both 0.01 values, and the average of positions 3, 4, and 5, namely 4, is assigned as the rank of all three 0.82 values.

Example 2 dataset with ranks

Value	0.01	0.01	0.82	0.82	0.82
Position	1	2	3	4	5
Rank	1.5	1.5	4	4	4

G.2 Gehan Rank for censored datasets

When there are non-detects in a dataset, Gehan rank assigns the average of the positions a value could take in the ordered dataset if the censored value were known. The table below shows a dataset with ten values, four of which are censored. The first step in determining the Gehan ranks is to order the values in the natural way, and assign a starting position to each observation.

Example 3 dataset with starting positions

Value	<10	<10	20	<30	30	30	<40	50	60	90
Starting Position	1	2	3	4	5	6	7	8	9	10

The next step is to understand what positions each value could take if the true values for the censored observations were known. The table below illustrates some of the possible orderings. The first row shows the basic initial ordering. To understand the next three rows, note that a value known to be <30 could be anything less than 30, and so could take any of the earlier positions, potentially forcing the observed value of 20 and/or the censored values of <10 to higher positions. Similarly, the censored value represented by <40 could fall in any position between 1 and 7.

Consider the censored values and the positions they could take. For instance, <30 could take any of the first four positions, and could also take the fifth position if <40 represents a value that is less than the value represented by <30. In general, a censored value can take any position up to and including its starting position, and can take positions beyond its starting position when there are non-detects with larger starting positions. The possible positions for a non-detect in starting position k are $1, 2, \ldots, k, k+1, \ldots, k+r$, where r is the number of non-detects with starting positions greater than k. The average of these is $(k + r + 1)/2$.

Possible orderings of Example 3 dataset

Ordering	Position									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	<10	<10	20	<30	30	30	<40	50	60	90
2	<10	<10	<30	20	30	30	<40	50	60	90
3	<10	<30	<10	20	30	30	<40	50	60	90
4	<30	<10	<10	20	30	30	<40	50	60	90
5	<10	<10	20	<30	30	<40	30	50	60	90
6	<10	<10	<30	20	30	<40	30	50	60	90
7	<10	<30	<10	20	30	<40	30	50	60	90
8	<30	<10	<10	20	30	<40	30	50	60	90
9	<10	<10	20	<30	<40	30	30	50	60	90
10	<10	<10	<30	20	<40	30	30	50	60	90
11	<10	<30	<10	20	<40	30	30	50	60	90
12	<30	<10	<10	20	<40	30	30	50	60	90
...										

Next consider the uncensored values. The possible positions of uncensored values are bounded below by the starting position. The measured value of 20, for instance, will always be larger than the two censored values of <10 and <10, so its smallest possible position is 3rd. However the uncensored value of 20 might also be larger than censored values with larger initial positions than 20 itself, meaning 20 could take the 4th position if one of the censored values <30 or <40 is in fact smaller than 20, or the 5th position if both <30 and <40 are smaller than 20. Similarly, 30 could take its initial starting positions (5th or 6th) or one larger (7th), where the latter happens if the censored value <40 is in fact less than 30. The uncensored values 50, 60, and 90 will always have the 8th, 9th, and 10th positions respectively. In general, the possible positions for a non-detect in starting position k are $k, k+1, \ldots, k+r$, where r is the number of non-detects with starting positions greater than k. The average of these is $k + r/2$, unless the uncensored value in position k is a tie, in which case the starting position k is replaced by the average starting position of the tied values.

A summary of the values, their starting positions, their possible positions, and their Gehan ranks (the average of the possible positions) is shown below.

Gehan ranks for Example 3 dataset

Value	<10	<10	20	<30	30	30	<40	50	60	90
Starting Position	1	2	3	4	5	6	7	8	9	10
Possible Positions	1,2,3,4	1,2,3,4	3,4,5	1,2,3,4,5	5,6,7	5,6,7	1,2,3,4, 5,6,7	8	9	10
Gehan Rank	2.5	2.5	4	3	6	6	4	8	9	10

G.3 Percentiles for uncensored datasets

There are many algorithms for estimating quantiles for uncensored datasets. If the $p * 100$ percentile of a dataset of size n is desired for $0 \leq p \leq 1$, each algorithm will specify the $p * 100$ percentile as an observation in the dataset or as an interpolation between two consecutive values in the ordered dataset. Here, the primary interest is in the situation where $p = 0.90$.

Let $\lfloor \cdot \rfloor$ denote the floor function, so $\lfloor k \rfloor$ is the largest integer less than or equal to k, and let $\lceil \cdot \rceil$ denote the ceiling function, so $\lceil k \rceil$ is the smallest integer greater than or equal to k. When k is an integer, the floor and ceiling of k are the same; when k is not an integer, the floor and ceiling are the integers immediately below and above k, respectively.

One simple quantile/percentile algorithm takes the $p * 100$ of a dataset of size n to be the $\lceil p * n \rceil$th observation in the ordered dataset. For a dataset of size 10, the 90th percentile is then the 9th observation in the ordered dataset. See the Example 4 (ordered) dataset with the 90th percentile in bold.

Example 4 dataset, with 90th percentile in bold.

Example 4 dataset	8	10	20	22	30	30	36	40	**60**	90

G.4 Percentiles for censored datasets using Gehan Ranking

The Example 3 dataset and the Gehan ranks of the observations are repeated below.

Example 3 dataset with Gehan ranks

Value	**<10**	**<10**	**20**	**<30**	**30**	**30**	**<40**	**50**	**60**	**90**
Gehan Rank	2.5	2.5	4	3	6	6	4	8	9	10

Reordering the values according to the Gehan rank assignment results in the ordered dataset shown below.

Example 3 dataset, ordered according to Gehan rank

Value	**<10**	**<10**	**<30**	**20**	**<40**	**30**	**30**	**50**	**60**	**90**
Gehan Rank	2.5	2.5	3	4	4	6	6	8	9	10

In this case, the 90th percentile would be estimated as the observation in the 9th spot, where the ordering is dictated by Gehan ranking, namely 60. When the censored values are all in the lower 80% of the ordered dataset, which is the most common situation, the 90th percentile computed using Gehan ranking coincides with the 90th percentile computed using the detection limit for censored values.

Note that there are several algorithms for computing, or estimating, quantiles/percentiles from data. The one used here is simple, but it is biased high (the ten data points in the example correspond to the 10^{th} through 100^{th} percentiles). An alternative is to assume these sampels represent the 0^{th} through the 90^{th} percentiles, or to provide symmetry, and make them the 5^{th}, 15^{th}, …. , 95^{th} percentiles, or another form of symmetry as the 0^{th}, 11.1%, 22.2%, …, 100% quantiles. When there are many data points, as there are for most analyte and stratigraphic layer combinations in the Colstrip dataset, the actual method for calculation does not have a huge effect on the estimated quantile values.

G.5 Boostrapped UTLs

The upper tolerance limit (UTL) of interest for this investigation is the 95% upper confidence limit on the 90th percentile of the data for a specific analyte and stratigraphic layer combination. This is denoted as the 95/90 UTL, and can be computed by bootstrapping 90th percentiles, and then taking the 95th percentile from the bootstrapped data. The process is illustrated with the Example 3 dataset.

For the Example 3 dataset, each bootstrap iteration involves drawing 10 observations, with replacement, from the original dataset (the number of bootstrap samples must be the same as the number of actual samples). Usually, several thousand bootstrap samples are generated. An example bootstrap sample from the Example 3 dataset is shown in the table below:

Bootstrap Sample 1 for Example 3 dataset

Bootstrap Sample 1	**20**	**<30**	**30**	**<40**	**<40**	**<40**	**50**	**60**	**90**	**90**

The Gehan ranks are computed and shown below.

Bootstrap Sample 1 with Gehan ranks for Example 3 dataset

Bootstrap Sample 1	**20**	**<30**	**30**	**<40**	**<40**	**<40**	**50**	**60**	**90**	**90**
Gehan Rank	**3**	**3**	**4.5**	**3.5**	**3.5**	**3.5**	**7**	**8**	**9**	**10**

When the dataset is reordered according to the Gehan ranks, the 9th position is 90, which is the estimated 90th percentile.

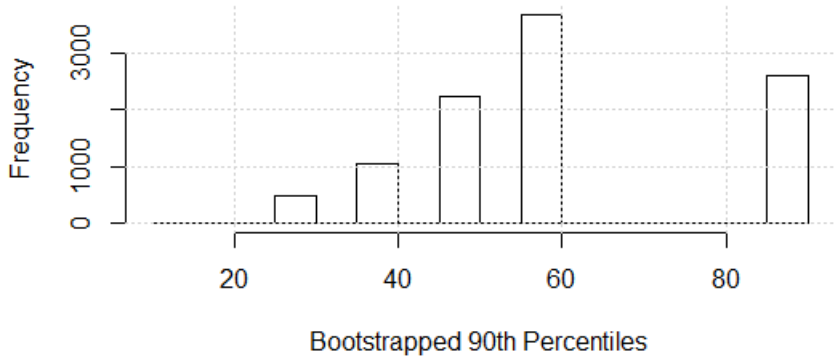
Taking another bootstrap sample, and computing Gehan ranks, gives the sample and ranks shown:

Bootstrap Sample 2 with Gehan ranks for Example 3 dataset

Bootstrap Sample 2	**<10**	**<10**	**<10**	**20**	**<30**	**30**	**30**	**50**	**50**	**60**
Gehan Rank	**2.5**	**2.5**	**2.5**	**4.5**	**3**	**6.5**	**6.5**	**8.5**	**8.5**	**10**

When the second bootstrap dataset is reordered according to the Gehan ranks, the 9th position is 60, which provides the second estimate of the 90th percentile.

The process of resampling and finding the 90th percentile of the resampled dataset is repeated many times (10,000 in the computations made here), and the 90th percentile from each iteration is recorded, leading to a set of 10,000 estimates of the 90th percentile, the first two of which, from the examples above, are 90 and 60. A set of 10,000 results are shown in the figure below.



Histogram of bootstrapped 90th percentile values for Example 3 dataset

The 95th percentile from the bootstrapped statistics is 90. Thus, the 90/95 UTL, which is taken to be the BSL, is 90 for this dataset.

The finite discrete nature of the possibilities is clear when there are so few samples in the original dataset. However, when there are many samples, the bootstrapped distribution tends to "fill in" a lot more, and the 95th percentile of the bootstrapped data can appear to come from a near-continuous distribution. This also depends on the number of non-detects in the original dataset and the positions of those non-detects relative to the detected values.