Estimating Population Range by Recurring Online Chunk Bootstrap with Non-cumulative Data in Streaming Data Environment

***Abstract*—**

***Index Terms*—Article submission, IEEE, IEEEtran, journal, LaTeX, paper, template, typesetting.**

I. Introduction

The bootstrap method is a powerful and widely used statistical tool for estimating the uncertainty from finite samples of any parameter of interest, such as standard error, confidence intervals, and accuracy, etc. Efron [**?**] introduced bootstrap methods as a generalization of the jackknife method, where the jackknife was mathematically expressed as a linear approximation of the bootstrap. Empirical results demonstrated that these methods are capable of estimating the standard error of complex estimators effectively. Efron and Tibshirani [**?**] further explored the basic concepts and applications of bootstrap methods for estimating standard errors, confidence intervals, and other measures of statistical accuracy. For standard error estimation, data from a population is resampled with replacement to create multiple bootstrap samples, allowing for the calculation of statistics and their standard deviations for each sample. The results showed that bootstrap estimations provided reasonably accurate and efficient alignments with theoretical density curves. In terms of confidence intervals, several methods were empirically tested, showing that bootstrap confidence intervals can be more accurate than standard methods, particularly when the distribution of the statistic is non-normal. Achieving accurate results requires a sufficient number of bootstrap replications. Carpenter and Bithell [**?**] presented a practical guide for applying bootstrap confidence intervals in healthcare data, addressing when, which, and how to implement these methods. They assessed various bootstrap methods for confidence intervals across three families: pivotal, non-pivotal, and test-inversion. The experiments concluded that bootstrap confidence intervals are a suitable alternative when the assumptions of the underlying distribution do not hold, such as asymptotic normality, especially with small sample sizes or complex data structures.

Range approximation in one-dimensional data plays an important role in statistics, data analysis, and computational sciences, aiming to estimate the interval between a dataset's minimum and maximum values.

II. Studied Problem and Objectives

Let a chunk of numeric data ($\mathbf{C}$), a bin ($\mathbf{B}$), and a set of numeric data ($\mathbf{D}$) be defined as follows:

1) $\mathbf{D} = \{d_j \in R |\ a \leq d_j \leq b \text{ for } j = 1, 2, \ldots, m \text{ and } a, b, \in R\}$
2) The chunk $\mathbf{C}$ is divided into n smaller chunks $\mathbf{c}_j \subset D$ for $j = 1, 2, \ldots, n$ and $\cup_{i=1}^{n} \mathbf{c}_i \subset D$.
3) A set of bins $\mathbf{B}$ for containing all numeric values of a single chunk $\mathbf{c}_i$. There are two attributes, namely v_i^{min} and v_i^{max} such that for all $d_j \in \mathbf{c}_i$, $d_j \in [v_i^{min}, v_i^{max}]$.

Each chunk $\mathbf{c}_i$ flows sequentially into the bin. If some values are less than $v^{(min)}$ or greater than $v^{(max)}$, then the width of B must be expanded. In the beginning, the size of the bin $\mathbf{B}$ is made large enough to contain all values in the first incoming chunk $\mathbf{c}_1$. Then, the size of $\mathbf{B}$ is occasionally expanded so that all integers in the following incoming chunks can be assigned to the intervals of $\mathbf{B}$. The bin $\mathbf{B}$ is expanded if the values of some integers in some incoming chunks are either less than $v^{(min)}$ or greater than $v^{(max)}$. Therefore, the problem studied in this research is to determine the most effective and efficient way to expand the bin bfB when necessary.

As aforementioned, a and b are the minimum and maximum values of $\mathbf{D}$, respectively. After capturing of all elements in the first chunk $\mathbf{c}_1$, how to achieve the minimum number of expansions of $\mathbf{B}$ so that

1) All incoming values in the following chunks can be assigned to $\mathbf{B}$.
2) $v^{(min)}$ is closed to a as much as possible.
3) $v^{(max)}$ is closed to b as much as possible.

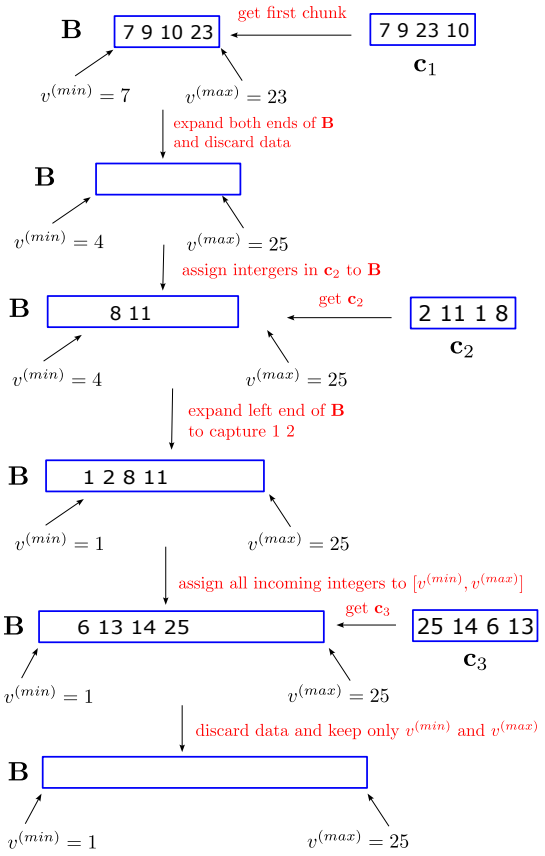


Fig. 1: An example of studied problem.

Figure 1 illustrates an example scenario of capturing chunks. Assumed that three sequentially incoming chunks contain these integer elements: $\mathbf{c}_1 = \{7, 9, 23, 10\}$, $\mathbf{c}_2 = \{2, 11, 1, 8\}$, $\mathbf{c}_3 = \{25, 14, 6, 13\}$. After capturing $\mathbf{c}_1$, the values of the left and right ends ($v^{(min)}$ and $v^{(max)}$)of bin $\mathbf{B}$ are set to 7 and 23, respectively, and all data are discarded. Then, both ends are expanded in advance to 4 and 25 for $v^{(min)}$ and $v^{(max)}$, preparing for $\mathbf{c}_2$. When $\mathbf{c}_2$ enters, all integers except 2 can be captured because the value at the left-end value $v^{(min)} = 4$ is larger than 2. Thus, the left end $v^{(min)}$ is expanded to 2. There is no need for the right-end expansion. All data in $\mathbf{c}_2$ are discarded. Then, get $\mathbf{c}_3$. The interval of $\mathbf{B}$ is large enough to capture all the integers of $\mathbf{c}_3$. After capturing $\mathbf{c}_3$, all integers are discarded. In this example, the number of expansions is 2, after chunks 1 and 2. Generally, how to achieve the minimum number of expansions of $\mathbf{B}$ in advance so that both expanded values are also minimum.

A. Constraints

Let $|\mathbf{c}_i|$ be the number of elements in a chunk $\mathbf{c}_i$. The bin B can be considered an interval with a left end denoted by L and a right end denoted by R. The following constraints are imposed.

1) The probability of distribution of each integer d_j in $\mathbf{c}_i$ is unknown.
2) $|\mathbf{c}_i| \neq |\mathbf{c}_{i+1}|$ or $|\mathbf{c}_i| = |\mathbf{c}_{i+1}|$.
3) After assigning all integers of $\mathbf{c}_i$ inside the bin, chunk $\mathbf{c}_i$ is completely discarded and never reentered the bin assignment.

III. Concept of Recurring Online Bootstrap in Streaming Environment with Unrecorded Data

Figure 2 illustrates the concept of recurring online bootstrap in a streaming environment with unrecorded data. A standard histogram with eight bins highlighted in pink was created based on a statistical distribution. The height of each box represents the theoretical number of elements derived from the statistical distribution. As shown at the top of Figure 2, it assumed that the first data chunk 1 arrived. Suppose that the number of incoming data satisfied by the leftmost or rightmost interval is larger than that of a standard histogram noted as blue colour. In that case, expansion of the current standard histogram is required. In our work, the recurring online bootstrap is proposed to extend incrementally to the streaming data chunks. The middle part of Figure 2 shows after the extension is performed, the numbers of the incoming data chunks of the leftmost and rightmost bins are not larger than the numbers of the leftmost and rightmost bins of the standard histogram. The bottom part of Figure 2 shows after the extension is performed, the numbers of the incoming data chunks of the leftmost and rightmost bins are not larger than the numbers of the leftmost and rightmost bins of the standard histogram. The bottom part of Figure 2 shows the summary of the extension in which the upper subfigure shows the right-end extension and the lower subfigure shows the left-end extension.

A. Algorithm

The proposed algorithms of the recurring online bootstrap in a streaming environment are given as follows:
Algorithm 1: Capturing a data chunk $\mathbf{c}_i$ and expanding $\mathbf{B}$ when it is necessary.

Input: Stream of data chunks, $\mathbf{c}_i$ for $1 \leq i < \infty$.
Output: Minimum and maximum values denoted by $v^{(min)}$ and $v^{(max)}$

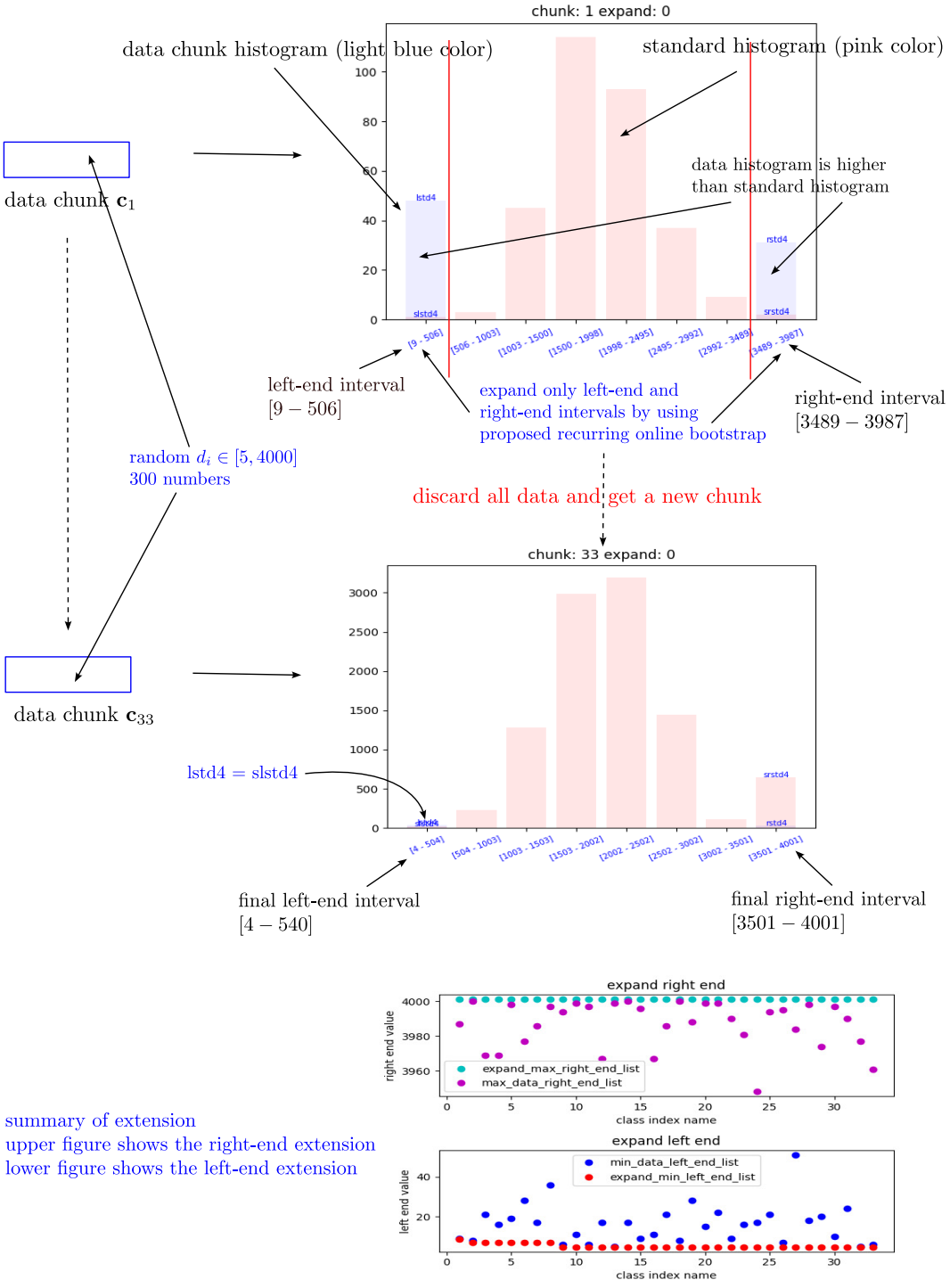


Fig. 2: Framework.

1. Get the first chunk $\mathbf{c}_1$ and set $total\_data = |\mathbf{c}_1|$.
2. **If** $i = 1$ **then**
3. $v^{(min)} = \min(\mathbf{c}_1)$ and $v^{(max)} = \max(\mathbf{c}_1)$.
4. **Else**

5. **If** $v^{(min)} > min(\mathbf{c}_i)$ **then**
6. $v^{(min)} = min(\mathbf{c}_i)$.
7. **EndIf**
8. **If** $v^{(max)} < max(\mathbf{c}_i)$ **then**
9. $v^{(max)} = max(\mathbf{c}_i)$.
10. **EndIf**
11. **EndIf**
12. Construct a set of bins $\mathbf{B}$ with 8 equal sub-intervals $\mathbf{b}_i$ for $1 \leq i \leq 8$ by
$\min(\mathbf{b}_i) = v^{(min)} + width * (i-1)$,
$\max(\mathbf{b}_i) = v^{(min)} + width * (i)$,
where $width = (v^{(max)} - v^{(min)})/8$.
13. Assign the elements in $\mathbf{c}_i$ to the sub-intervals $\mathbf{b}_1$ and $\mathbf{b}_8$.
14. Let $avg = (v^{(max)} + v^{(min)})/2$ be the central value of $\mathbf{B}$.
15. Find the types of probability distribution in list P best fitting the data in $\mathbf{b}_1$, and $\mathbf{b}_8$ by using **Alg. 2.1** with its arguments $total\_data$, $\mathbf{b}_1$, and $\mathbf{b}_8$.
16. Compute the theoretical number of elements in $\mathbf{b}_1$ denoted by $lstd$, and in $\mathbf{b}_8$ denoted by $rstd$ from the best fitted probability distribution by using **Alg. 2.2**.
17. **While** $|\mathbf{b}_1| > lstd$ or $|\mathbf{b}_8| > rstd$ **do**
18. **If** $|\mathbf{b}_1| > lstd$ **then**
19. Compute the left-expanded and right-expanded values denoted by $expand^{(min)}$ and $expand^{(max)}$ by using **Alg. 3** for all elements in $\mathbf{b_1}$, and **Alg. 4** for all elements in $\mathbf{b}_8$.
20. **if** $expand^{(min)} \geq v^{(min)}$ **and** $expand^{(max)} \leq v^{(max)}$ **then**
21. **break** and go to Line 32 .
22. **Else**
23. **If** $v^{(min)} > expand^{(min)}$ **then**,
24 Replace $v^{(min)}$ by $expand^{(min)}$.
25. **EndIf**
26. **If** $v^{(max)} < expand^(max)$ **then**,
27. Replace $v^{(max)}$ by $expand^{(max)}$.
28. **EndIf**
29. Go to Line 12.
30. **EndIf**
31. **EndWhile**
32. Discard $\mathbf{c}_i$ and all elements in $\mathbf{b}_1$ and $\mathbf{b}_8$.
33. **Return** $v^{(min)}$ and $v^{(max)}$.

Algorithm 2.1: Find the types of probability distribution in the list P that best fit the data in $\mathbf{b}_1$, and $\mathbf{b}_8$.

Input: (1) A list of the standard probability distribution P. (2) $total\_data$. (3) $|\mathbf{b}_1|$. (4) $|\mathbf{b}_8|$.
Output: $lname$ and $rname$.

1. **For** each type probability distribution $p \in P$ **do**
2. Divide the area under standard probability distribution p into 8 stripes of equal width.
3. Let $l_4^{(p)}$ be the percentage of data in the 4^{th} stripe to the left of the mean.
4. Let $r_4^{(p)}$ be the percentage of data in the 4^{th} stripe to the right of mean.
5. Compute the difference between standard number of elements in the 4^{th} left stripe and $|\mathbf{b}_1|$:
$ld^{(p)} = abs(l_4^{(p)} * total\_data - |\mathbf{b}_1|)$.
6. Compute the difference between standard number of elements in the 4^{th} right stripe and $|\mathbf{b}_8|$:
$rd^{(p)} = abs(r_4^{(p)} * total\_data - |\mathbf{b}_8|)$.
7. **EndFor**
8. Let $lname = \arg\min_{p \in P}(ld^{(p)})$.
9. Let $rname = \arg\max_{p \in P}(rd^{(p)})$.
10. **Return** $lname$ and $rname$.

Algorithm 2.2: Computing $lstd$ and $rstd$.

Input: (1) A list of the standard probability distribution P. (2) $total\_data$. (3) $lname$. (4) $rname$. (5) $l_4^{(p)}$ and $r_4^{(p)}$; $\forall p \in P$ from Algorithm 2.1.
Output: $lstd$ and $rstd$.

1. **If** $lname$ is the same as $rname$ **then**
2. Set $lstd = l_4^{(lname)} * total\_data$.
3. Set $rstd = r_4^{(rname)} * total\_data$.
4. **EndIf**
5. **If** $lname$ is different from $rname$ **then**
6. Set $lstd = \max_{p \in P}(l_4^{(p)}) * total\_data$.
7. Set $rstd = \max_{p \in P}(r_4^{(p)}) * total\_data$.
8. **EndIf**
9. **Return** $lstd$ and $rstd$.

Algorithm 3: Recurring online chunk bootstrap for $\mathbf{b}_1$.

Input: (1) Present set of incoming elements in $\mathbf{b}_1$; (2) Number of bootstrap iterations N; (3) $mean(\mathbf{a})$ is a function computing the mean of set $\mathbf{a}$; (4) $std(\mathbf{a})$ is a function computing the standard deviation of set $\mathbf{a}$; (5)

$abs(x)$ is the absolute value of constant x.
Output: $v^{(min)}$.

1. Let $\mathbf{S} = \emptyset$ be a set of bootstrapped samples.
2. Let $\mathbf{M} = \emptyset$ be a set of mean values of each bootstrapped sample. bootstrapped samples.
3. Let $\mathbf{P} = \emptyset$ be a set of standard deviation of each bootstrapped samples.
4. $prev\_mean = 0$.
5. **For** $1 \leq i \leq N$ **do**:
6. Let $\mathbf{s}_i$ be a set of randomly sampled integers of size $|\mathbf{b}_1|$ from $\mathbf{b}_1$ with replacement.
7. $\mathbf{S} = \mathbf{S} \cup \{\mathbf{s}_i\}$.
8. $present\_mean = (mean(\mathbf{s}_i) + prev\_mean)/2$.
9. $prev\_mean = present\_mean$.
10. $\mathbf{M} = \mathbf{M} \cup \{present\_mean\}$.
11. $\mathbf{P} = \mathbf{P} \cup \{std(\mathbf{s}_i)\}$.
12. **EndFor**.
13. $\mu^{(boot)} = mean(\mathbf{M})$.
14. $\sigma^{(boot)} = mean(\mathbf{P})$.
15. $\mu^{(diff)} = abs(mean(\mathbf{b}_1) - \mu^{(boot)})$.
16. $\sigma^{(diff)} = abs(std(\mathbf{b}_1) - \sigma^{(boot)})$.
17. **If** $\mu^{(boot)} < mean(\mathbf{b}_1)$ **then**
18. $v^{(min)} = min(\mathbf{b}_1) - \mu^{(diff)}$.
19. **EndIf**
20. **If** $mean(\mathbf{b}_1) < \mu^{(boot)}$ **then**
21. $v^{(min)} = min(\mathbf{b}_1) - \sigma^{(diff)}$.
22. **EndIf**
23. **Return** $v^{(min)}$

Algorithm 4: Recurring online chunk bootstrap for $\mathbf{B}_8$.

Input: (1) Present set of incoming integers in $\mathbf{B}_1$; (2) Number of bootstrap iterations N; (3) $mean(\mathbf{a})$ is a function that compute the mean of the set $\mathbf{a}$; (4) $std(\mathbf{a})$ is a function that compute the standard deviation of the set $\mathbf{a}$; (5) $abs(x)$ is the absolute value of constant x.
Output: $v^{(max)}$.

1. Let $\mathbf{S} = \emptyset$ be a set of bootstrapped sample set $\mathbf{s}$.
2. Let $\mathbf{M} = \emptyset$ be a set of mean values of each bootstrapped sample set.
3. Let $\mathbf{P} = \emptyset$ be a set of standard deviation of each bootstrapped sample set.
4. $prev\_mean = 0$.
5. **For** $1 \leq i \leq N$ **do**:
6. Let $\mathbf{s}_i$ be a set of randomly sampled integers of size $|\mathbf{b}_8|$ from $\mathbf{b}_8$ with replacement.
7. $\mathbf{S} = \mathbf{S} \cup \{\mathbf{s}_i\}$.
8. $present\_mean = (mean(\mathbf{s}_i) + prev\_mean)/2$.
9. $prev\_mean = present\_mean$.
10. $\mathbf{M} = \mathbf{M} \cup \{present\_mean\}$.
11. $\mathbf{P} = \mathbf{P} \cup \{std(\mathbf{s}_i)\}$.
12. **EndFor**.
13. $\mu^{(boot)} = mean(\mathbf{M})$.
14. $\sigma^{(boot)} = mean(\mathbf{P})$.
15. $\mu^{(diff)} = abs(mean(\mathbf{b}_8) - \mu^{(boot)})$.
16. $\sigma^{(diff)} = abs(std(\mathbf{b}_8) - \sigma^{(boot)})$.
17. **If** $\mu^{(boot)} < mean(\mathbf{b}_8)$ **do**
18. $v^{(max)} = max(\mathbf{b}_8) + \sigma^{(diff)}$.
19. **EndIf**
20. **If** $mean(\mathbf{B}_1) < \mu^{(boot)}$ **do**
21. $v^{(max)} = max(\mathbf{b}_8) + \mu^{(diff)}$.
22. **EndIf**
23. **Return** $v^{(max)}$.

IV. Experiments

A. Data for experiments

This study comprehensively evaluated the range approximation using two scenarios of one-dimensional data populations: simulated and real-world data. As detailed in Table I, we used ten datasets from five established statistical distributions, each with distinct parameter settings. The simulation framework encompasses four fundamental distributions: F distribution $F(\nu_1, \nu_2)$, Chi-square $\chi 2(\nu_1)$, Normal distribution $N(\mu, \sigma^2)$, Wald distribution $Wald(\mu, \lambda)$, and Weibull distribution $Weibull(shape)$. We implemented two configurations for the F distribution, including $F(5, 100)$ and $F(5, 20)$, which exhibit notable differences in their statistical characteristics. Although both share the same numerator degrees of freedom ($\nu_1 = 5$), the increase in the denominator degrees of freedom from 10 to 15 results in significant changes in the distribution properties. The second configuration demonstrates a substantially wider range [0.0199, 83.5992] compared to the first [0.0083, 52.4746], with approximately 59% larger maximum value, indicating greater potential for extreme values. This configuration also exhibits more pronounced right skewness, heavier tails, and increased variability in its upper tail. These distinct characteristics between the two configurations provide a comparative study for evaluating the robustness of range approximation methods under different distributional conditions and tail behaviours. For the Chi-square distribution, the two chi-squared distributions, $\chi 2(\nu = 2)$ and $\chi 2(\nu = 2)$, were simulated. The $\chi 2(\nu = 2)$ distribution shows a highly right-skewed shape with its peak near zero and a range of [0.0000, 21.9290], while $\chi 2(\nu = 10)$ displays a more symmetric shape, with a range of [0.7468, 33.6357]. The $\chi 2(\nu = 10)$ distribution also shows greater spread and variability, reflected in its broader range and larger maximum value.

To assess the performance under normal distributions, we maintained a consistent mean of zero while varying the standard deviation $\sigma^2 = 16$ and $\sigma^2 = 1$, with ranges spanning from -3.6943 to 4.2593 and -14.2604 to 18.0044 respectively. Both normal distributions share the same mean but have significantly different variances, allowing us to investigate the impact of different spread patterns with the same mean parameter but differ in their λ parameters, leading to distinct characteristics. With the smaller λ value, Wald exhibits a more spread-out distribution with a range of [0.0323, 15.1948] and shows heavier right-tail behaviour, indicating a higher probability of extreme values. In contrast, with the larger λ value, Wald demonstrates a more concentrated distribution around the mean with a range of [0.1090, 12.8305] and a lighter tail. For Weibull distribution, the different shape parameters 1 and 5 exhibit distinct characteristics in their behaviour and applications. Weibull with shape=1 behaves like an exponential distribution with a monotonically decreasing pattern and a wide range of [0.1090, 12.2323], characterized by a constant hazard rate, making it suitable for modelling random failures and exponential decay processes. In contrast, Weibull with shape=5 presents a bell-shaped, nearly symmetric distribution with a narrower range of [0.1209, 1.5980], featuring increased failure rates over time and more concentrated probability density around the characteristic life. This significant difference in their ranges and probability density distributions reflects their reliability engineering and failure analysis applications. In addition, we set the number of simulated data to 10,000 samples for all datasets. This number is considered the population size for construction streaming data environment.

TABLE I: Description of statistical distribution's parameters, minimum and maximum values of the simulation scenario.

No.	Distribution	Minimum	Maximum
1	$F(\nu_1 = 5, \nu_2 = 10)$	0.0083	52.4746
2	$F(\nu_1 = 5, \nu_2 = 15)$	0.0199	83.5992
3	$\chi^2(\nu = 2)$	0.0000	21.9290
4	$\chi^2(\nu = 10)$	0.7468	33.6357
5	$N(\mu = 0, \sigma^2 = 16)$	-3.6943	4.2593
6	$N(\mu = 0, \sigma^2 = 1)$	-14.2604	18.0044
7	$Wald(\mu = 1, \lambda = 0.5)$	0.0323	15.1948
8	$Wald(\mu = 1, \lambda = 2)$	0.1090	12.8305
9	$Weibull(shape = 1)$	0.1090	12.2323
10	$Weibull(shape = 5)$	0.1209	1.5980

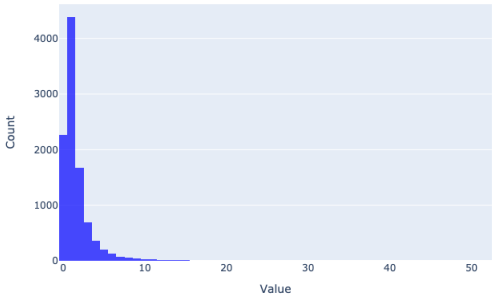
In our experiments on real-world scenarios, we examined four distinct datasets representing various economic sectors, as detailed in Table II. These data sets were selected to evaluate our methodology in different real-world applications and value distributions. The first data set comprises laptop price data, with values ranging from 174.00 to $6,099.00$. This substantial price range encompasses the diverse laptop market, from entry-level consumer devices to high-end professional systems, providing a comprehensive representation of price distribution in the technology retail sector. The second data set focuses on electronic sales transactions from 20.75 to $11,396.80$. This extensive range captures the full spectrum of electronic retail activity, from minor accessory purchases to significant investments in premium electronic equipment. Our third data set examines e-commerce sales data, with transactions ranging from 100.30 to $9,995.62$. This range reflects the diverse nature of online retail transactions, demonstrating the breadth of consumer purchasing behaviour in the digital marketplace. The dataset's distribution provides valuable insights into e-commerce transaction patterns and consumer spending behaviours. The fourth dataset investigates world tourism economy indicators, ranging from 0.1578 to 28.1923. This dataset's unique scale and distribution characteristics offer an important perspective on our methodology's applicability to macroeconomic indicators.

TABLE II: Description of minimum and maximum values of the real-world scenario.

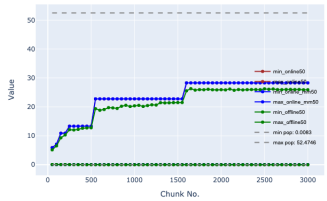
No.	Distribution	Minimum	Maximum
1	Laptop prices	174.00	6,099.00
2	Electronic sales	20.75	11,396.80
3	E-commerce sales	100.30	9,995.62
4	world tourism economy	0.1578	28.1923

B. Experimental results

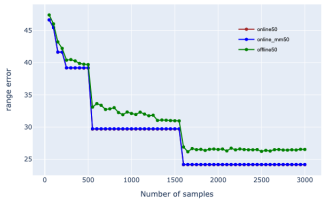
For creating the streaming data chunks environment, given a population $\mathbb{P}$ with size N, let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ be the set of samples randomly selected from the population $\mathbb{P}$ as the set of samples. We created the streaming data chunks with size L.



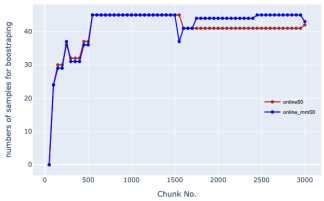
(a)



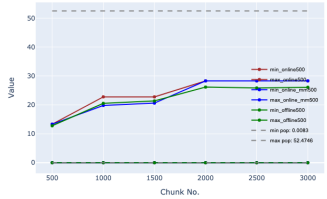
(b)



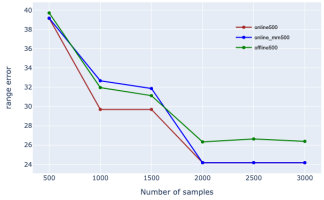
(c)



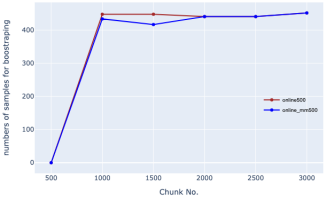
(d)



(e)

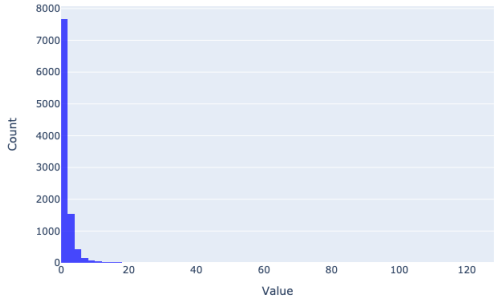


(f)

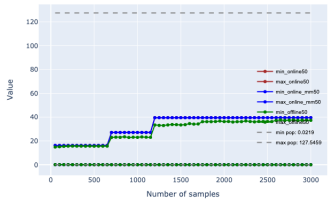


(g)

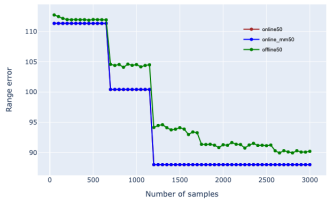
Fig. 3: Range approximation results of $F(\nu_1 = 5, \nu_2 = 10)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



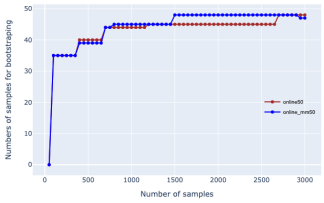
(a)



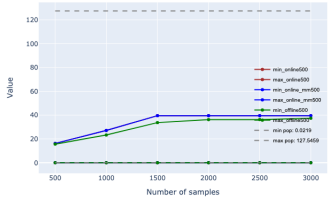
(b)



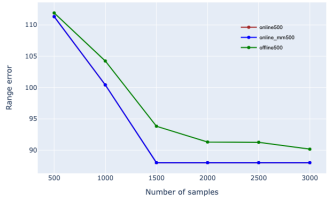
(c)



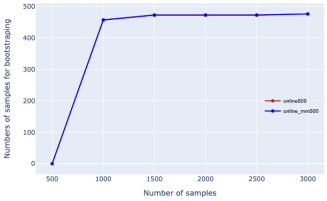
(d)



(e)

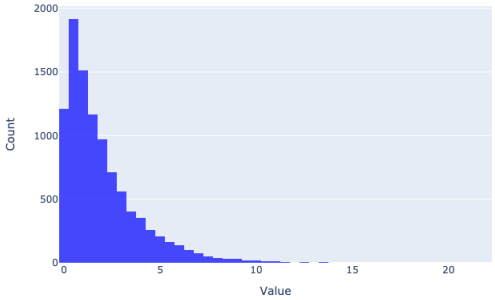


(f)

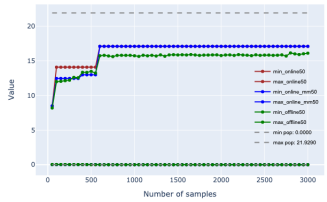


(g)

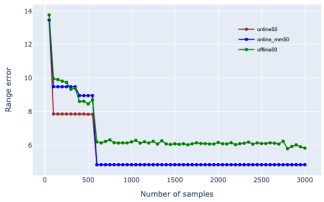
Fig. 4: Range approximation results of $F(\nu_1 = 5, \nu_2 = 20)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



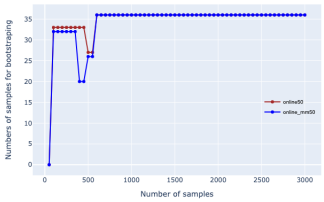
(a)



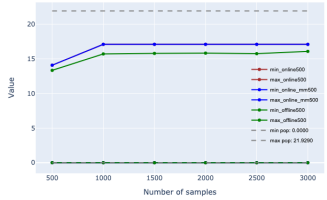
(b)



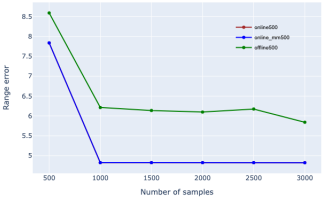
(c)



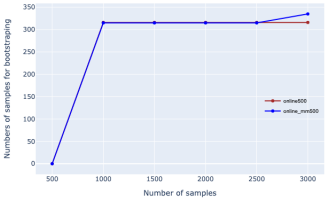
(d)



(e)

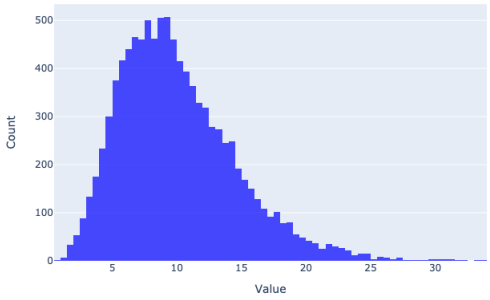


(f)

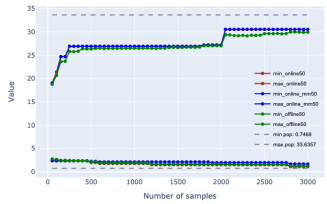


(g)

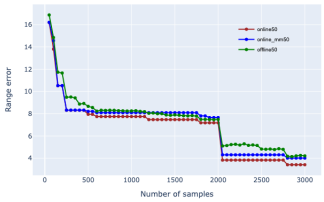
Fig. 5: Range approximation results of $\chi^2(\nu = 2)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



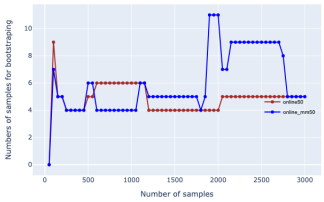
(a)



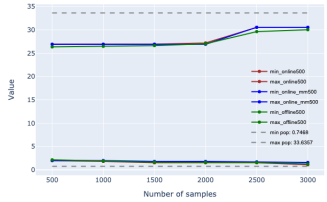
(b)



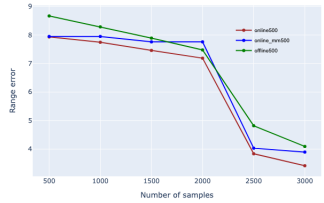
(c)



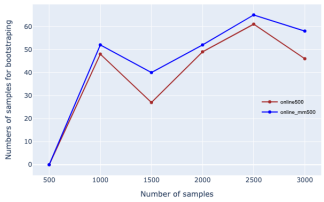
(d)



(e)

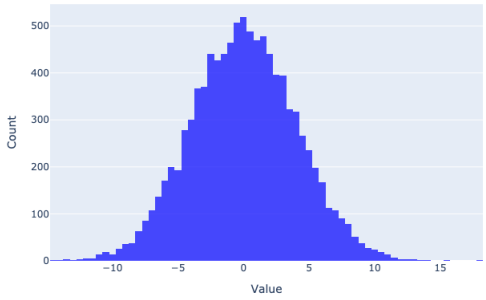


(f)

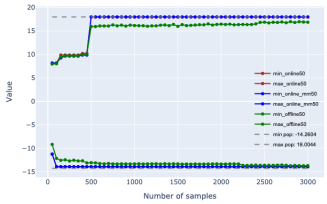


(g)

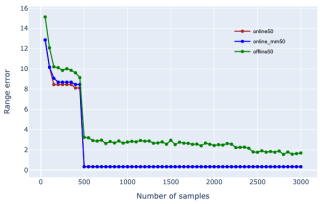
Fig. 6: Range approximation results of $\chi^2(\nu = 10)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



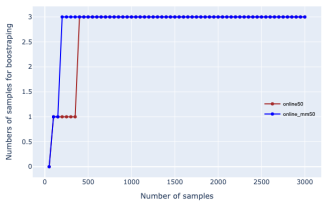
(a)



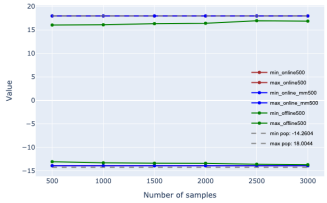
(b)



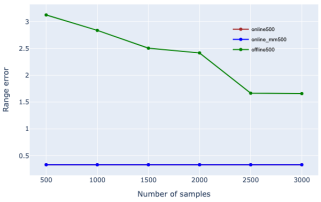
(c)



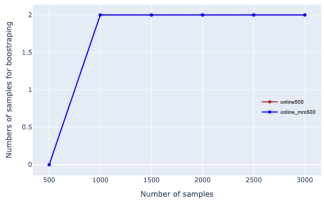
(d)



(e)

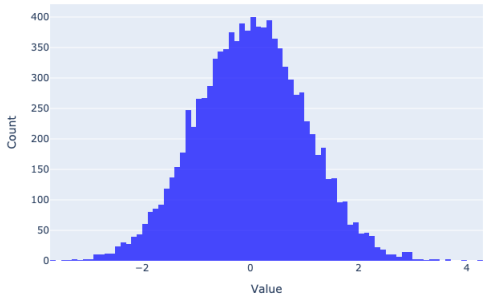


(f)

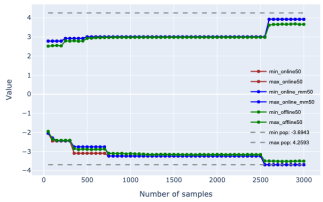


(g)

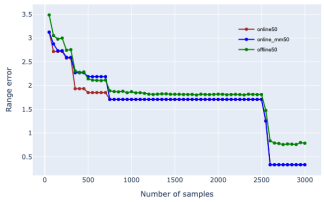
Fig. 7: Range approximation results of $N(\mu = 0, \sigma^2 = 16)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



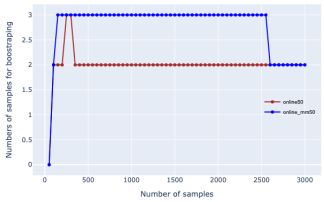
(a)



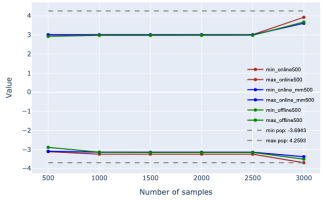
(b)



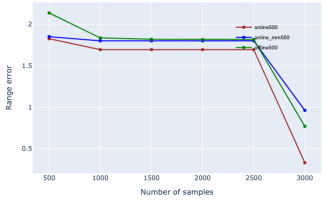
(c)



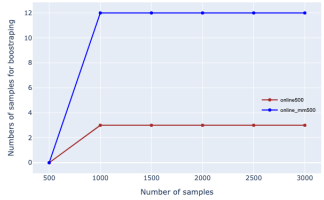
(d)



(e)

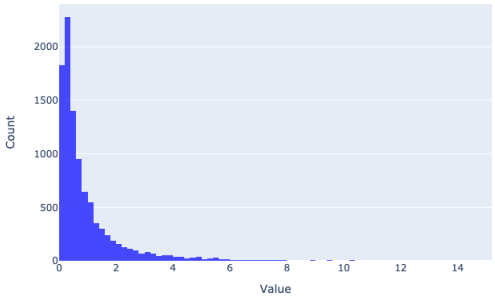


(f)

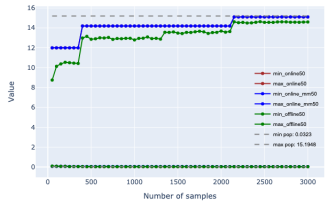


(g)

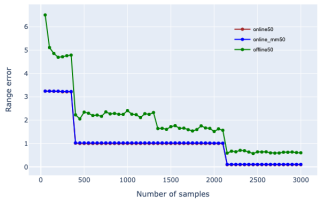
Fig. 8: Range approximation results of $N(\mu = 0, \sigma^2 = 1)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



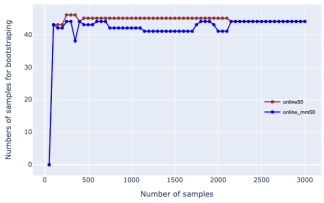
(a)



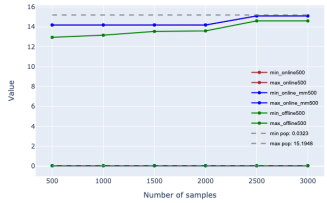
(b)



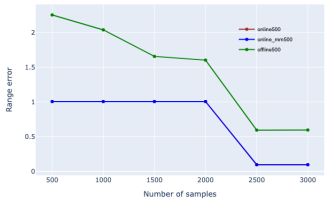
(c)



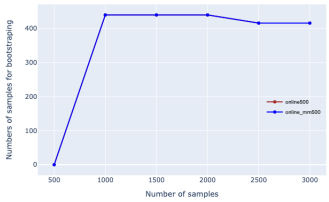
(d)



(e)

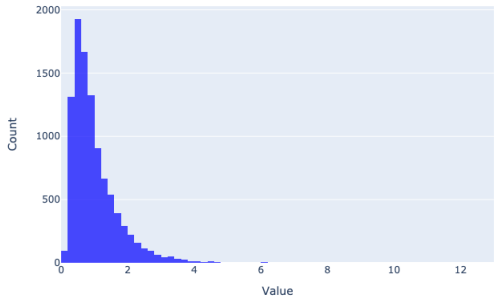


(f)

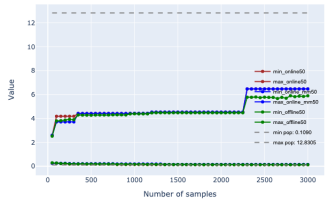


(g)

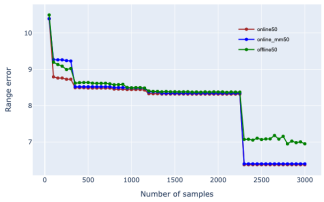
Fig. 9: Range approximation results of $Wald(\mu = 1, \lambda = 0.5)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



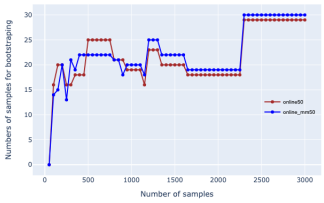
(a)



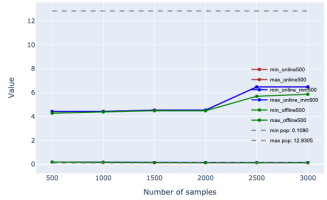
(b)



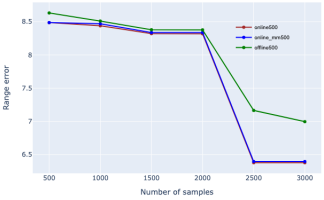
(c)



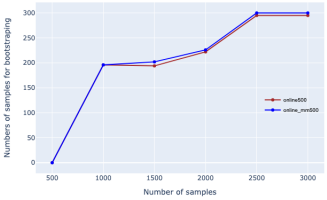
(d)



(e)

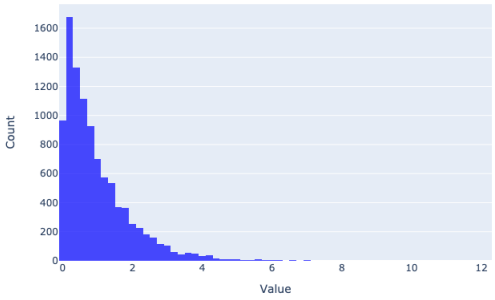


(f)

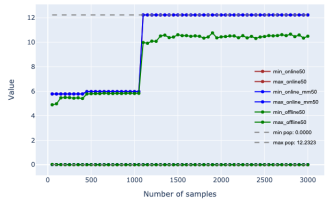


(g)

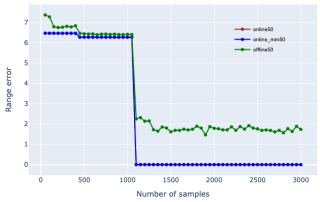
Fig. 10: Range approximation results of $Wald(\mu = 1, \lambda = 2.0)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



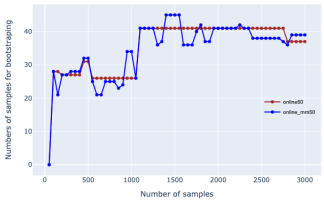
(a)



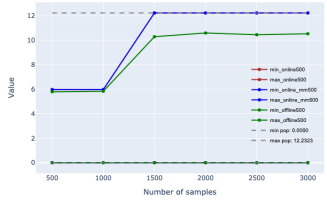
(b)



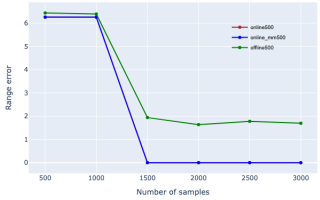
(c)



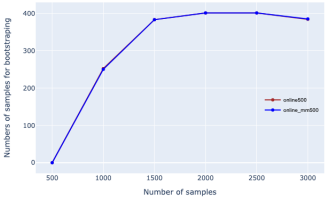
(d)



(e)

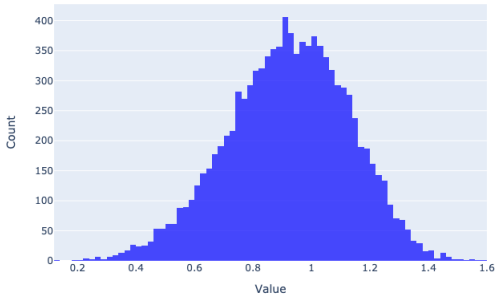


(f)

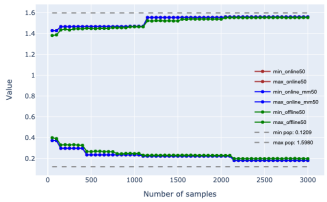


(g)

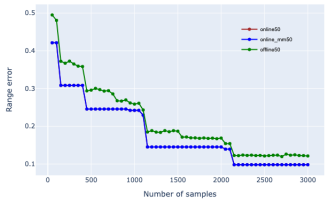
Fig. 11: Range approximation results of $Weibull(shape = 1)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



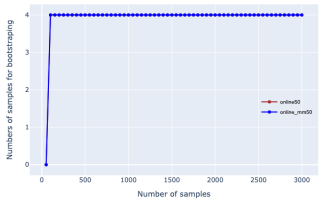
(a)



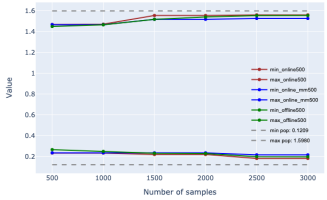
(b)



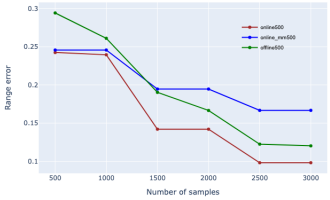
(c)



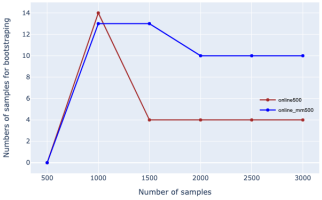
(d)



(e)

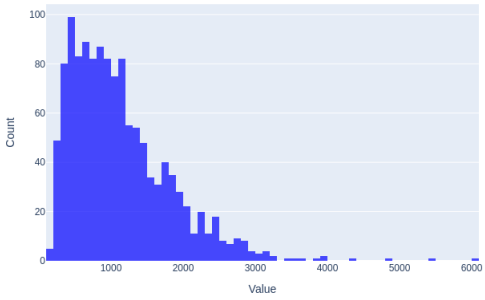


(f)

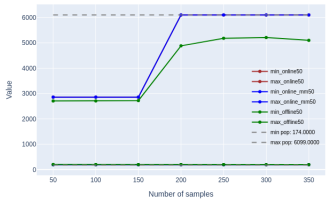


(g)

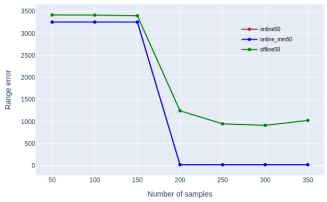
Fig. 12: Range approximation results of $Weibull(shape = 5)$ including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



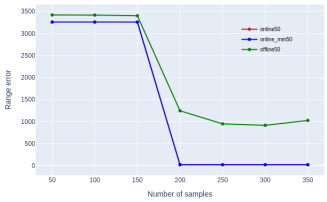
(a)



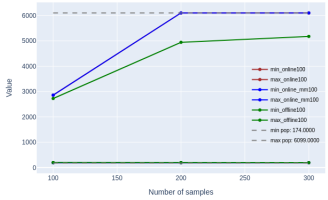
(b)



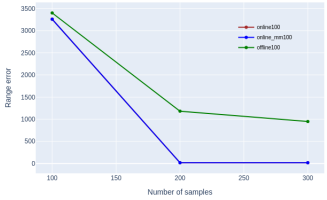
(c)



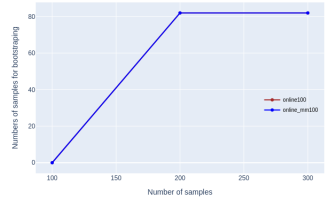
(d)



(e)

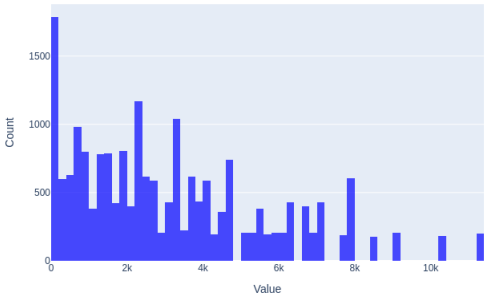


(f)

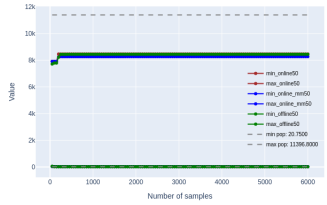


(g)

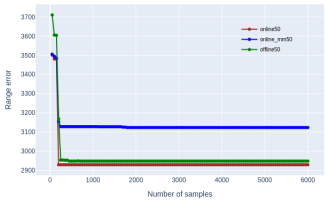
Fig. 13: Range approximation results of laptop prices including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



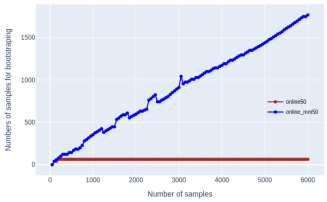
(a)



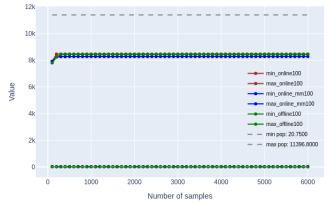
(b)



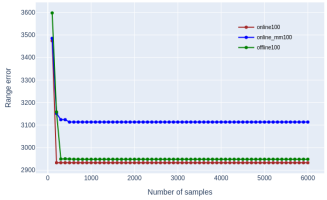
(c)



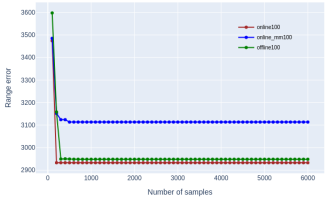
(d)



(e)

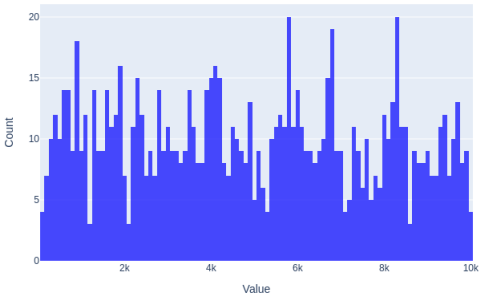


(f)

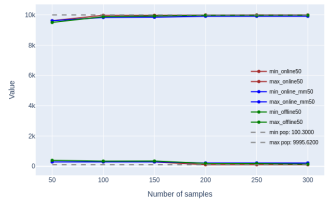


(g)

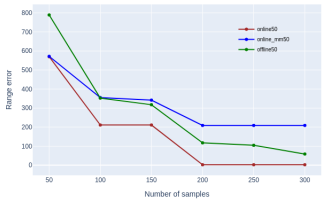
Fig. 14: Range approximation results of Electronic sales including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



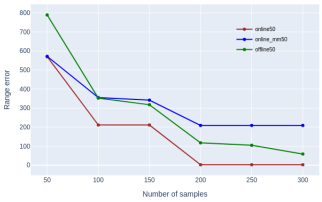
(a)



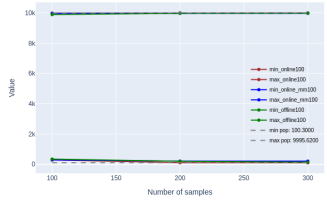
(b)



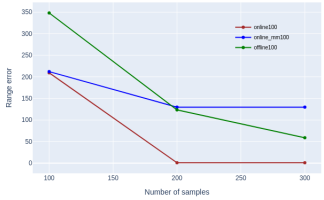
(c)



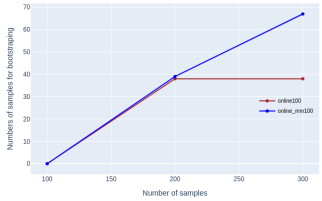
(d)



(e)

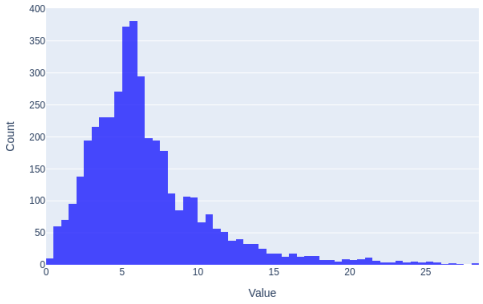


(f)

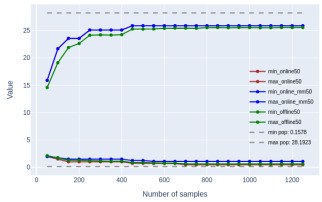


(g)

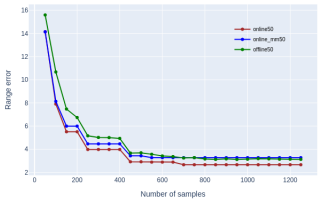
Fig. 15: Range approximation results of e-commerce sales including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.



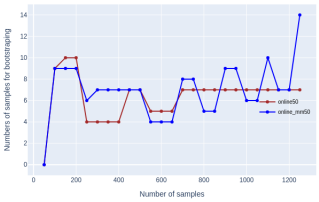
(a)



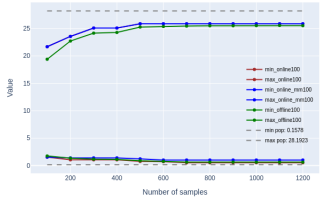
(b)



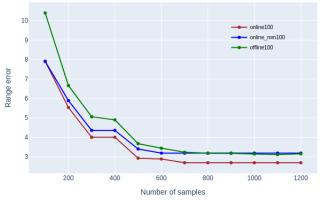
(c)



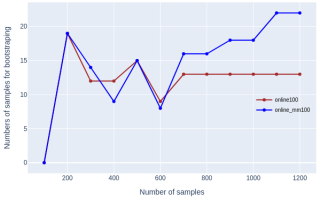
(d)



(e)



(f)



(g)

Fig. 16: Range approximation results of world tourism economy including i) histogram of population data as shown in (a), ii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 50 samples/chunk, as shown in (b), (c), and (d), respectively, and iii) min-max approximation, range error and number of leftmost and rightmost bins for bootstrapping, for 500 samples/chunk, as shown in (e), (f), and (g), respectively.