2021

Upper-Sided EWMA-Based Distribution-Specific Tolerance Limits

Owen Visser
University of North Florida, n00894010@unf.edu

UPPER-SIDED EWMA-BASED DISTRIBUTION-SPECIFIC TOLERANCE LIMITS

by

Owen Visser

A Thesis submitted to the Department of Mathematics & Statistics
in partial fulfillment of the requirements for the degree of
Master of Science
UNIVERSITY OF NORTH FLORIDA
COLLEGE OF ART AND SCIENCES
April, 2021

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

ABSTRACT

Tolerance limits are constructed from sample data to ascertain if a proportion of a process is within specification limits. There exists multiple methods of calculating the sample size requirements for tolerance limits under various assumptions. In this research, a distribution-specific algorithm that utilizes the exponentially weighted moving average technique (EWMA), first introduced by Sa and Razaila (2004), is reconstructed. The algorithm is used to calculate the required sample sizes for continuous construction of upper-sided tolerance limits. The sample sizes and intervals constructed from them are compared to three existing methods for various distributions. The distribution-specific algorithm was observed to reduce the sample size requirements more rapidly, and to a greater extent, than the competitors.

CHAPTER 1

INTRODUCTION

When creating a product or conducting a process it is important to know the desired specifications are met. Tolerance limits are calculated to assess the capabilities of a process based on how close the limits fall within, or on, the predetermined specification limits. The most widely known tolerance limits are created under the assumption that the population is normally distributed. Given a sample of n observations, $X_1, X_2, ..., X_n$, denote the sample mean and sample standard deviation $\overline{X}$ and s, respectively. For a constant k, determined by the specification proportion and probability requirement, the two-sided tolerance limits of a process can be defined as $\overline{X} \pm ks$ (Armstrong, 1982) where k must satisfy

$$P(P(\overline{X} - ks < X < \overline{X} + ks) \geq q) = 1 - \alpha \tag{1.1}$$

where q is the specification percentage and $1 - \alpha$ is the probability requirement.

While these tolerance limits have their uses, the assumption of normality may not be possible. Often skewed distributions are valid descriptors of processes or populations.

In this case, non-parametric tolerance limits are a common choice. From a sample, the non-parametric tolerance method uses the minimum order statistic (X_{min}), maximum order statistic (X_{max}), or both to calculate lower, upper, or two-sided tolerance intervals, respectively. This is written as,

$$P(P(X_{min} \leq X) \geq q) = 1 - \alpha \tag{1.2}$$

$$P(P(X \leq X_{max}) \geq q) = 1 - \alpha \tag{1.3}$$

$$P(P(X_{min} \leq X \leq X_{max}) \geq q) = 1 - \alpha \tag{1.4}$$

In this method, a sample is taken of required size n, such that X_{max} and X_{min} can satisfy the required specification percentage with probable accuracy. The use of these tolerance limits and their required sample sizes are described in detail by Conover (1998). When a process or production is not subject to change, tolerance limits, such as these, can be constructed at the beginning of a process and remain unchanged without repercussion. Tolerance limits created in this fashion can provide useful inferences to the process capabilities, and recalculation of the tolerance limits is not necessary. However, if the process or production is changed, multiple samples should be taken to ensure that the specifications are met. The overall increase in sample size for the duration of a process can result in the occasional outlier, especially for processes defined by skewed distributions. The maximum and minimum values that define Conover's standard non-parametric tolerance limits are subject to large variation when outliers are present. Amin and Lee (1999) described the ill effects of outliers on non-parametric tolerance limits.

Razaila and Sa (2003) introduced an improved method of generating non-parametric tolerance limits that reduces the effect of outliers, and as a result reduces the mean and standard deviation of the interval length. In Sa and Razaila's method the same sample sizes from Conover's method are used, then they are compiled into a cumulative sample. Afterward, the cumulative sample is ordered and the specific sample unit(s), at the quantile position that is required to satisfy probability conditions, is taken as the upper, lower, or two-sided tolerance limit(s). While this method has its benefits, the reoccurring large samples can diminish resources and may have large costs.

Another approach to calculating continuous tolerance intervals is to employ the exponentially weighted moving average (EWMA) technique. This technique is used to continuously update statistics for any process. Lee and Sa (2001) illustrated the use of the EWMA technique as a tool for constructing tolerance limits, as well as the sample sizes required to construct them. So long as the distribution has the increasing failure rate (IFR) property, this EWMA-based method showed reduced necessary sample sizes and decreased averages and standard deviations of tolerance interval lengths for the family of distributions that follow the IFR property. While Lee and Sa's method (2001) reduced those features by a considerable amount, they can still be reduced further.

Using an EWMA-based method, Sa and Razaila (2004) formulated a general algorithm to solve for sample size requirements for both lower and upper sided tolerance intervals, for any specific distribution. They also provided tables of the sample sizes required to construct the lower sided tolerance intervals of several distributions at varying parameter values.

The goal of this thesis is to reconstruct the distribution-specific algorithm in order to calculate the sample sizes for upper-sided tolerance limits. In addition, the distribution-specific general algorithm was compared to three other commonly used methods of constructing tolerance intervals. More specifically, the averages and standard deviations of intervals constructed along with the sample sizes used for these methods are simulated and compared. Chapter two consists of an introduction to the EWMA technique, followed by the construction of the upper-sided general algorithm with the use of EWMA in the third chapter. The fourth chapter discusses the coding of the algorithm, and what was observed from the distributions used. The fifth chapter compares the four methods in sample sizes and averages and standard deviations of interval lengths across all distributions and methods. Next, an illustrative example which shows the use of the algorithm and EWMA technique in a real world scenario is in the sixth chapter. This leads to the conclusion and finally a discussion of decision making in chapter seven.

CHAPTER 2

INTRODUCING EWMA

Assume samples are taken from some known process independently over time at times $t = 1, 2, ...$ Let

$$\mathcal{X}_t = \{X_{1t}, X_{2t}, \ldots, X_{n_t t}\}, \tag{2.1}$$

where n_t is the sample size taken at time t. Let $\hat{X}_t$ be some statistic calculated from $\mathcal{X}_t$. Define the EWMA statistic, Z_t, for $t > 0$ as

$$Z_t = \lambda \hat{X}_t + (1 - \lambda) Z_{t-1}, \tag{2.2}$$

where λ is a pre-specified constant such that $0 < \lambda < 1$. By solving recursively for Z_t, it can be represented as a sum of all past statistics. Define the constant c_1 as the initial set-up value in the process, such that $\hat{X}_0 = Z_0 = c_1$.

$$\begin{aligned}
Z_t &= \lambda \hat{X}_t + (1 - \lambda) Z_{t-1} \\
&= \lambda \hat{X}_t + (1 - \lambda)(\lambda \hat{X}_{t-1} + (1 - \lambda) Z_{t-2}) \\
&= \lambda \hat{X}_t + (1 - \lambda)\lambda \hat{X}_{t-1} + (1 - \lambda)^2 Z_{t-2} \\
&\vdots \\
Z_t &= \lambda \hat{X}_t + (1 - \lambda)\lambda \hat{X}_{t-1} + (1 - \lambda)^2 \lambda \hat{X}_{t-2} + \cdots + (1 - \lambda)^{t-1} \lambda \hat{X}_1 + (1 - \lambda)^t c_1
\end{aligned}$$

Define w_i for $i = 0, 1, 2, ..., t,$ as

$$w_i = \begin{cases} (1-\lambda)^t & \text{when } i = 0 \\ \lambda(1-\lambda)^{t-i} & \text{when } i > 0 \end{cases} \qquad \text{where} \quad \sum_{i=0}^{t} w_i = 1. \tag{2.3}$$

Z_t can then be written as

$$Z_t = c_1(1-\lambda)^t + \sum_{i=1}^{t} \hat{X}_i \lambda (1-\lambda)^{t-i} = \sum_{i=0}^{t} w_i \hat{X}_i. \tag{2.4}$$

The expected value and variance of the EWMA statistic are as follows,

$$E(Z_t) = c_1(1-\lambda)^t + \sum_{i=1}^{t} E(\hat{X}_i)\lambda(1-\lambda)^{t-i}$$
$$V(Z_t) = \sum_{i=1}^{t} V(\hat{X}_i)\lambda^2(1-\lambda)^{2(t-i)}.$$

Moreover, since the X_i's are identically distributed, the expected value and variance become

$$E(Z_t) = c_1(1-\lambda)^t + E(\hat{X})(1-(1-\lambda)^t) \tag{2.5}$$

$$V(Z_t) = V(\hat{X}) \left(\frac{\lambda}{2-\lambda}\right) (1-(1-\lambda)^{2t}). \tag{2.6}$$

The choice of λ has great impact on the practicality of the EWMA technique. With $\lambda \in (0, 1)$, as $t \to \infty$ the expected value of the EWMA statistic, $E(Z_t)$, approaches $E(\hat{X})$ and variance slowly decreases to become a constant reliant only on λ.

$$V(Z_t)_{t\to\infty} = V(\hat{X}) \left(\frac{\lambda}{2-\lambda}\right) \tag{2.7}$$

Therefore, as the number of samples taken increases, the convergence of the variance depends on λ. The smaller λ is, the slower the rate of convergence will be. At its boundaries, λ impacts the results of the continuous sampling. On the right boundary, as $\lambda \to 1$, the weights will become more focused toward the present. That is, when $\lambda = 1$, $w_i = 0$ for $i = 0, 1, ..., t-1$, and $w_t = 1$. The expectation becomes $E(\hat{X})$, and variance $V(\hat{X})$. In this case only the most recent sample is considered. On the other end of the boundary it can be seen that as $\lambda \to 0$ only the initial sample has significance. When $\lambda = 0$, $w_i = 0$ for $i = 1, 2, ..., t$, and $w_0 = 1$. The expectation becomes the initial estimate, $\hat{X}_0$, and variance becomes zero.

If the X_i's are not identically distributed, the above properties of convergence may not hold. Although, the weighting effect on present or past samples can be shown to affect rate of convergence. Commonly used values for economic data are $\lambda = 0.2 \pm 0.1$ as suggested by Hunter (1986); these values will be implemented in this research.

CHAPTER 3

THE UPPER-SIDED DISTRIBUTION-SPECIFIC ALGORITHM

Consider a continuous process, with known p.d.f. $f(\cdot)$ and c.d.f. $F(\cdot)$, that is sampled independently over time. The samples taken are of size n_i for $i = 1, 2, ..., t$. Let $X_{min\ i}$ and $X_{max\ i}$ be the smallest and largest values of sample i, respectively. The p.d.f. and c.d.f. of the smallest and largest order statistic at time i are as follows:

$$h_i(x) = n_i[1 - F(x)]^{n_i - 1} f(x) \qquad g_i(x) = n_i[F(x)]^{n_i - 1} f(x) \tag{3.1}$$

$$H_i(x) = 1 - [1 - F(x)]^{n_i} \qquad G_i(x) = [F(x)]^{n_i} \tag{3.2}$$

where $h_i(\cdot)$ and $H_i(\cdot)$ specifies p.d.f. and c.d.f of the smallest order statistic, and $g_i(\cdot)$ and $G_i(\cdot)$ specifies the p.d.f. and c.d.f. of the largest order statistic.

The EWMA minimum and maximum are defined as $Z_{min\ t}$ and $Z_{max\ t}$, respectively.

$$Z_{min\ t} = \sum_{i=0}^{t} w_i X_{min\ i} \tag{3.3}$$

$$Z_{max\ t} = \sum_{i=0}^{t} w_i X_{max\ i} \tag{3.4}$$

In particular, the one-sided upper tolerance interval at time t can then be defined as

$$(-\infty, Z_{max\ t})$$

and must satisfy

$$P(P(X \leq Z_{max\ t}) \geq q) = 1 - \alpha. \tag{3.5}$$

That is, with probability $1-\alpha$ at least the proportion, q, of the process can be found within the interval $(-\infty, Z_{max\ t})$. From (3.5), the sample sizes can be calculated to satisfy the probability requirement for the proportion of the process.

Consider the one-sided upper tolerance interval at time t, $(-\infty, Z_{max\ t})$, with $Z_{max\ 0} = X_{max\ 0} = c_2$ such that $F(c_2) = q$, where F is the c.d.f. of known process. The first required sample size at $t = 1$ can be calculated by manipulating (3.5). By applying the known distribution it can be shown that,

$$\begin{aligned} 1 - \alpha &= P(P(X \leq Z_{max\ 1}) \geq q) \\ &= P(F(Z_{max\ 1}) \geq q) \\ &= P(Z_{max\ 1} \geq F^{-1}(q)) \\ &= P(Z_{max\ 1} \geq c_2). \end{aligned}$$

Using the definition of the EWMA statistic from equation (2.2) it can be seen that

$$P(\lambda X_{max\ 1} + (1-\lambda)c_2 \geq c_2) = 1 - \alpha.$$

The left hand side can then be manipulated,

$$\begin{aligned} & P(\lambda X_{max\,1} \geq c_2 - c_2(1-\lambda)) \\ = \; & P(\lambda X_{max\,1} \geq c_2\lambda)) \\ = \; & P(X_{max\,1} \geq c_2) \\ = \; & 1 - P(X_{max\,1} < c_2). \end{aligned}$$

By implementing the order statistic distribution of $X_{max\,1}$,

$$1 - \alpha = 1 - P(X_{max\,1} < c_2) = 1 - G_1(c_2) = 1 - [F(c_2)]^{n_1}. \tag{3.6}$$

The value of n_1 is calculated to satisfy equation (3.6). Since sample size is an integer, the smallest n_1 is used such that $[F(c_2)]^{n_1} \approx \alpha$, while still ensuring the approximation is conservative; i.e. $[F(c_2)]^{n_1} \leq \alpha$. The sample size, n_1, calculated in this way is equal to the sample size required for the construction of the standard non-parametric tolerance interval, which can be found in the text of Conover (1998).

Equation (3.5) can be manipulated further to calculate the sample size requirements for $t > 1$. Consider the left hand side of equation (3.5) at time t. $X_{max\,i}$ will be denoted as x_i, let $c = c_2 - w_0c_2$. Then,

$$\begin{aligned} 1 - \alpha &= P(P(X \leq Z_{max,t}) \geq q) \\ &= P\left(\sum_{i=0}^{t} w_i x_i \geq c_2\right) \end{aligned}$$

$$
\begin{aligned}
&= P\left(\sum_{i=1}^{t} w_i x_i + w_0 c_2 \geq c_2\right) \\
&= P\left(\sum_{i=1}^{t} w_i x_i \geq c_2 - w_0 c_2\right) \\
&= P\left(\sum_{i=1}^{t-1} w_i x_i + w_t x_t \geq c\right) \\
&= 1 - P\left(\sum_{i=1}^{t-1} w_i x_i + w_t x_t < c\right) \\
&= 1 - \int_{-\infty}^{c/w_t} g_t(x_t) P\left(\sum_{i=1}^{t-1} w_i x_i < c - w_t x_t\right) dx_t \\
&= 1 - \int_{-\infty}^{c/w_t} g_t(x_t) P\left(\sum_{i=1}^{t-2} w_i x_i + w_{t-1} x_{t-1} < c - w_t x_t\right) dx_t \\
&= 1 - \int_{-\infty}^{c/w_t} g_t(x_t) \int_{-\infty}^{(c - w_t x_t)/w_{t-1}} g_{t-1}(x_{t-1}) P\left(\sum_{i=1}^{t-2} w_i x_i < c - \sum_{j=t-1}^{t} w_j x_j\right) dx_{t-1} dx_t.
\end{aligned}
$$

Continuing this process will give the following general algorithm

$$
1 - \alpha = 1 - \int_{-\infty}^{c/w_t} g_t(x_t) \left(\cdots \left(\int_{-\infty}^{(c - \sum_{i=2}^{t} w_i x_i)/w_1} g_1(x_1) dx_1 \right) \cdots \right) dx_t. \tag{3.7}
$$

CHAPTER 4

CALCULATING SAMPLE SIZES FOR SPECIFIC DISTRIBUTIONS

The upper-sided distribution specific algorithm was used to calculate the required sample sizes to construct the upper sided tolerance limits for various distributions. For each distribution considered, the method of calculating required sample size follows an identical procedure. Moreover, the same code structure was used for all distributions, with the only changes occurring in the equations of the p.d.f. and c.d.f.. The raw code can be found in the appendix (section B.1). The Python packages NumPy (Harris et al., 2020) and SciPy (Virtanen et al., 2020) were used to compute integration and generate random variables. The integration techniques used within the SciPy package follow a technique from the Fortran library QUADPACK (Piessens et al., 1983), and each integral was computed within 0.0005 absolute and relative errors. The generation of random numbers within both the NumPy and SciPy packages is done with the Mersenne Twister Sequence; a well documented and commonly used approach to generate random numbers in scientific research (Matsumoto and Nishimura, 1998).

For each distribution studied, it is assumed that the process is in a state of statistical control. It is also assumed that the process can be independently sampled over time. A general description of the code for a distribution that holds these assumptions, at α, λ, and q, is as follows;

1. The order statistic p.d.f. and c.d.f. along with their supports are defined.
2. The desired quantile, x_q, of the distribution is calculated.
3. For $t = 1$, the sample size is calculated based on equation (3.6).
4. For $t > 1$, the weights are calculated.
5. Given the set of previous sample sizes $\{n_1, ..., n_{t-1}\}$, SciPy is used to evaluate the required n_t from the integral (3.7) at time t. n_t is then recorded.
6. Steps 4 and 5 are repeated until required t is reached.

The values taken for α, λ, and q are as follows: $1 - \alpha$ at 0.90, 0.95, 0.98, and 0.99, q at 0.85, 0.90, 0.95, and 98%, and λ at 0.10, 0.20, and 0.30.

The distributions considered in this method were:

- Exponential ($\beta = 1$) with support $x \in [0, \infty)$ and p.d.f. and c.d.f.

$$f(x) = \frac{1}{\beta} e^{-x\left(\frac{1}{\beta}\right)}$$

$$F(x) = 1 - e^{-x\left(\frac{1}{\beta}\right)},$$

  with skewness and kurtosis $(2, 6)$.

- Standard Normal ($\mu = 0$, $\sigma = 1$), with support $x \in (-\infty, \infty)$ and p.d.f. and c.d.f.

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \frac{1 + \mathrm{erf}(x/\sqrt{2})}{2},$$

  with skewness and kurtosis $(0, 0)$.

- Inverse Gaussian ($\mu = 1,\ \theta = 0.5$), ($\mu = 1,\ \theta = 5$), ($\mu = 5,\ \theta = 1$), with support $x \in (0, \infty)$ and p.d.f. and c.d.f.

$$f(x) = \sqrt{\frac{\theta}{2\pi x^3}} \exp\left(-\frac{\theta(x-\mu)^2}{2\mu^2 x}\right)$$
$$F(x) = \Phi\left(\sqrt{\frac{\theta}{x}}\left(\frac{x}{\mu} - 1\right)\right) + \exp\left(\frac{2\theta}{\mu}\right)\Phi\left(-\sqrt{\frac{\theta}{x}}\left(\frac{x}{\mu} + 1\right)\right),$$

where

$$\Phi(x) = \frac{1 + \mathrm{erf}(x/\sqrt{2})}{2},$$

with skewness and kurtosis $(4.25, 30)$, $(1.34, 3)$, and $(6.71, 75)$ respectively.

- Weibull ($\eta = 1.5,\ \beta = 1$), ($\eta = 0.5,\ \beta = 1$), with support $x \in (0, \infty)$ and p.d.f. and c.d.f.

$$f(x) = \frac{\eta}{\beta}\left(\frac{x}{\beta}\right)^{\eta - 1} \exp\left(-(x/\beta)^{\eta}\right)$$
$$F(x) = 1 - \exp\left(-(x/\beta)^{\eta}\right),$$

with skewness and kurtosis $(1.07, 1.39)$ and $(6.62, 84.72)$ respectively.

- Log Normal ($\mu = 0,\ \sigma = 0.25$), ($\mu = 8,\ \sigma = 2$), with support $x \in (0, \infty)$ and p.d.f. and c.d.f.

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(\frac{(\ln x - \mu)^2}{2\sigma^2}\right)$$
$$F(x) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\ln(x - \mu)}{\sqrt{2}\sigma}\right),$$

with skewness and kurtosis $(0.78, 1.1)$ and $(414, 9220556)$ respectively.

- Log-logistic ($\alpha = 1$, $\beta = 2$), ($\alpha = 2$, $\beta = 10$), with support $x \in (0, \infty)$ and p.d.f. and c.d.f.

$$f(x) = \frac{(\beta/\alpha)(x/\alpha)^{\beta-1}}{(1 + (x/\alpha)^{\beta})^2}$$
$$F(x) = \frac{1}{1 + (x/\alpha)^{-\beta}},$$

with skewness and kurtosis and $(0.94, 3.51)$ for $\alpha = 2$, $\beta = 10$ only (the former parameter values have no defined skewness or kurtosis (Tadikamalla, 1980)).

- Chi Squared ($k = 2$), ($k = 6$), with support $x \in (0, \infty)$ and p.d.f. and c.d.f.

$$f(x) = \frac{1}{2^{(k/2)}\Gamma(k/2)} x^{(k/2)-1} e^{-x/2}$$
$$F(x) = \frac{1}{\Gamma(k/2)} \gamma\left(\frac{k}{2}, \frac{x}{2}\right),$$

where

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$$
$$\gamma(x, y) = \int_0^y t^{x-1} e^{-t} dt,$$

with skewness and kurtosis $(2, 6)$ and $(1.15, 2)$ respectively.

- Cauchy ($x_0 = 0$, $\gamma = 0.5$), ($x_0 = 0$, $\gamma = 1.5$), with support $x \in (-\infty, \infty)$, $\gamma \in (0, -\infty)$, and p.d.f. and c.d.f.

$$f(x) = \left[\pi\gamma\left(1 + \frac{(x - x_0)^2}{\gamma^2}\right)\right]^{-1}$$
$$F(x) = \frac{1}{\pi} \arctan\left(\frac{x - x_0}{\gamma}\right) + \frac{1}{2}.$$

with no defined skewness and kurtosis.

4.1 Observations of Sample Sizes Obtained

Tables A.1-A.15 contain the sample sizes calculated from these distributions via the upper-sided distribution specific algorithm. They can be found in the Appendix, section A.1.

It can be seen from every table, as t increases there is a reduction in sample size. As λ increases, there was a slower reduction in sample sizes. Increasing λ directly increases the variance of the EWMA maximum causing increased sample size requirements. This is due to the variance of the EWMA statistic having a direct correlation to λ and t as seen in equation (2.7). The only observable effect of q and α are increases of the initial sample size required at $t = 1$.

After a few increments, the sample sizes reduction nearly stabilizes. This is more apparent in lower α and q values where the sample size is the same for the last few ts in most tables. While the larger αs and qs show slower convergence, the sample sizes calculated for $t = 5$ can still be used as if they had converged. If the sample size of $t = 5$ is repeated, the tolerance limits constructed from them will still be within statistical accuracy but more conservative.

From the distributions considered, the mean and variance have little, or no, effect in the overall result of sample sizes. This can be seen in the distributions Chi Squared ($k = 2$) and Exponential ($\beta = 1$) in Tables A.12 and A.1, respectively. The two distributions share the same sample sizes, but differ in their mean and variance. It can also be seen that both distributions share the same skewness and kurtosis values.

The ratio of the skewness and kurtosis (skewness/kurtosis) was observed to influence the reduction of sample sizes. Individually, skewness and kurtosis were not observed to affect the sample sizes in any pattern. However, when the tables of distributions with defined skewness and kurtosis values were ordered by the ratio, (Tables A.6, A.8, A.13, A.4, A.1, A.11, A.3, A.5, A.7, A.9) the reduction in sample size follows a pattern; as the ratio decreases, so does the sample size requirement at $t = 2, 3, 4$, and 5. However, if skewness and kurtosis does not exist then there no discernible pattern. For example, Cauchy and Log-logistic ($\alpha = 1,\ \beta = 2$) do not have defined skewness and kurtosis values and no pattern can be identified.

The calculation of sample sizes through the algorithm can be very expensive; many distributions sample sizes took more than 24 hours to compute for just $t = 5$. As t increases linearly, the process time increases exponentially. Each new t forces the calculation of another layer of integration (equation 3.7). To maintain the precision of 0.005 in both absolute and relative error, the computation time extends well past any reasonable amount. Other methods of integration were tried when coding the algorithm such as Monte Carlo integration or using less restrictive errors, none of which resulted in accurate sample sizes in reasonable times. However, the sample sizes calculated at $t = 5$ are low enough that they can be repeated while still maintaining lower sample sizes than other continuous sampling methods.

CHAPTER 5

COMPARISONS BASED ON SIMULATION STUDIES

Four existing methods were compared: Standard Non-parametric method (SNP) (Conover, 1998), Continuous Non-Parametric method (CNP) (Razaila and Sa, 2003), EWMA-based method for distributions with IFR property (EWMA-IFR) (Lee and Sa, 2001), and EWMA-based Distribution Specific method (EWMA-DS) (Sa and Razaila, 2004). Two studies were conducted to compare the methods of constructing upper-sided tolerance intervals. Each study was conducted with varying t, q, λ, and α values.

In the first study, the sample size requirements from each method were used to calculate the average and standard deviation of the interval lengths, based on simulated sample data.

The second study compares the sample sizes required for construction of upper-sided tolerance intervals between the methods.

The distributions considered include,

- Inverse Gaussian ($\mu = 1$, $\theta = 5$ and $\mu = 5$, $\theta = 1$)
- Weibull ($\eta = 1.5$, $\beta = 1$ and $\eta = 0.5$, $\beta = 1$)
- Chi Squared ($k = 2$ and $k = 6$)
- Log Normal ($\mu = 0$, $\sigma = 0.25$ and $\mu = 8$, $\sigma = 2$)
- Log Logistic ($\alpha = 1$, $\beta = 2$ and $\alpha = 2$, $\beta = 10$)
- Cauchy ($x_0 = 0$, $\gamma = 0.5$ and $x_0 = 0$, $\gamma = 1.5$)
- Exponential ($\beta = 1$).

The following distributions were not used in comparisons for the EWMA-IFR method due to not satisfying the increasing failure rate property.

- Log Normal ($\mu = 8$, $\sigma = 2$)
- Log Logistic ($\alpha = 1$, $\beta = 2$ and $\alpha = 2$, $\beta = 10$)
- Cauchy ($x_0 = 0$, $\gamma = 0.5$ and $x_0 = 0$, $\gamma = 1.5$)

5.1 Interval-Length Comparisons

The calculation of the upper-sided tolerance limits for the EWMA-DS method was done by using equation (3.4). Tables A.16-A.28 contain the averages and standard deviations of the lengths of the upper-sided tolerance intervals constructed. Note that at $t = 1$ the upper-sided tolerance intervals of all methods are equal.

The procedure for calculating the average and standard deviation of interval length is as follows:

1. Given q, λ, and α, a set of data is generated from a distribution.
2. Samples are taken from the data set to match the required size of each method.
3. The proper statistic is used to compute the upper-sided tolerance limit for their respective method.
4. Steps 1-3 are done for $t = 1, 2, 3, 4, 5$. The tolerance intervals for each t are recorded.
5. For each distribution, step 4 is repeated 10000 times. The average and standard deviation of the recorded intervals are then calculated for each distribution and method.
6. Steps 1-5 are repeated for each choice of q, λ, and α

5.1.1 Interval-Length Comparisons Results

Tables A.16 - A.28 contain the averages and standard deviations of interval lengths from the distributions considered. It can be seen from the the tables that the EWMA-DS method out preformed the SNP and EWMA-IFR method for all distributions considered in both reduction of interval length and reduction in standard deviation.

The EWMA-DS method was observed to be relatively equal to the CNP method in average and standard deviation of interval length for most distributions. However, this was dependant on the value of λ. When $\lambda = 0.3$ the CNP method out preformed the EWMA-DS for almost all distributions. When $\lambda = 0.1$ the EWMA-DS method out preformed the CNP method in almost all distributions.

It should be noted that neither the Cauchy distribution or the Log-logistic distribution (for $\alpha = 1$, $\beta = 2$ specifically) have defined variances, which prevented the EWMA-DS and SNP methods from creating intervals with reasonable length.

5.2 Sample-Size Comparisons

It should be noted that the sample sizes of both the CNP and SNP methods are identical. Furthermore, Lee and Sa (2001) have already compared the sample sizes of the SNP method and EWMA-IFR method; resulting in far lower sample sizes for the EWMA-IFR method. The EWMA-DS method only needs to be compared to the EWMA-IFR method for distributions that satisfy the IFR property. For distributions that do not satisfy the IFR property the EWMA-DS method will be compared to the SNP sample sizes.

The sample sizes required from all methods were compared at $\alpha = 90\%$ at varying values of t, q, and λ. The percent reduction of sample size is calculated to compare the methods (see Table 5.1 and Table 5.2). Note that there is no reduction at $t = 1$ as both methods share the same required sample size.

5.2.1 Sample-Size Comparisons Results

The reduction of sample size by using the EWMA-DS method over the EWMA-IFR method for distributions that satisfy the IFR property was at least 20%. The reduction of sample size by using the EWMA-DS method over the SNP or CNP method for distributions that do not satisfy the IFR property was at least 50%.

In relation to the sample size calculations from the previous chapter, the value of the skewness and kurtosis ratio is inversely proportional to the reduction of sample size. The smallest ratio of skewness and kurtosis values offer the most reduction in sample sizes for the EWMA-DS method; 50% or more reduction in sample size when compared to the EWMA-IFR method for some distributions. The required sample sizes of the EWMA-DS method at $t = 5$ are nearly equivalent to, and often less than, the convergent required sample sizes (as $t \to \infty$) of the EWMA-IFR method (Lee and Sa, 2001).

Table 5.1: Upper sided tolerance sample size comparison EWMA-DS vs. EWMA-IFR

		$q = 90\%$			$q = 95\%$		
Distribution	t	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$
Inverse Gaussian	3	33.3	37.5	31.2	35.5	34.4	30.3
$\mu = 1$, $\theta = 5$	5	35.7	35.7	26.7	35.7	36.7	29.0
Inverse Gaussian	3	46.7	50.0	43.8	48.4	46.9	42.4
$\mu = 5$, $\theta = 1$	5	50.0	50.0	46.7	50.0	46.7	41.9
Weibull	3	33.3	31.2	31.2	35.5	31.2	30.3
$\eta = 1.5$, $\beta = 1$	5	35.7	28.6	26.7	32.1	33.3	29.0
Weibull	3	46.7	43.8	43.8	45.2	43.8	39.4
$\eta = 0.5$, $\beta = 1$	5	50.0	50.0	46.7	50.0	43.3	41.9
Chi-Squared	3	33.3	37.5	31.2	38.7	34.4	33.3
$k = 2$	5	35.7	28.6	26.7	35.7	36.7	32.3
Chi-Squared	3	33.3	31.2	31.2	35.5	31.2	30.3
$k = 6$	5	35.7	35.7	26.7	35.7	33.3	32.3
Exponential	3	33.3	37.5	31.2	38.7	34.4	33.3
$\beta = 1$	5	35.7	28.6	26.7	39.3	36.7	32.3
Log Normal	3	33.3	31.2	31.2	35.5	34.4	30.3
$\mu = 0$, $\sigma = 0.25$	5	35.7	28.6	26.7	35.7	33.3	29.0

Entries are the %Reduction in sample sizes for the EWMA-DS method compared to the EWMA-IFR method for $\alpha = 0.01$. The %Reduction was calculated as such

$$100 \left| \frac{n_{ifr} - n_{ds}}{n_{ifr}} \right| = \%\text{Reduction},$$

where n_{ifr} is the sample size required by the EWMA-IFR method, and n_{ds} is the EWMA-DS method.

Table 5.2: Upper sided tolerance sample size comparison EWMA-DS vs. SNP

		$q = 90\%$			$q = 95\%$		
Distribution	t	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$
Cauchy	3	68.2	68.2	63.6	71.1	68.9	64.4
$x_0 = 0$, $\gamma = 0.5$	5	77.3	72.7	68.2	75.6	73.3	68.9
Cauchy	3	68.2	68.2	63.6	71.1	68.9	64.4
$x_0 = 0$, $\gamma = 1.5$	5	77.3	72.7	68.2	75.6	73.3	68.9
Log Logistic	3	63.6	59.1	59.1	64.4	62.2	57.8
$\alpha = 1$, $\beta = 2$	5	68.2	63.6	63.6	68.9	66.7	62.2
Log Logistic	3	54.5	54.5	50.0	57.8	53.3	51.1
$\alpha = 2$, $\beta = 10$	5	63.6	54.5	54.5	62.2	57.8	55.6
Log Normal	3	68.2	68.2	63.6	68.9	64.4	62.2
$\mu = 8$, $\sigma = 2$	5	72.7	72.7	63.6	73.3	71.1	66.7

Entries are the %Reduction in sample sizes for the EWMA-DS method compared to the SNP method for $\alpha = 0.01$. The %Reduction was calculated as such

$$100 \left| \frac{n_{snp} - n_{ds}}{n_{snp}} \right| = \%\text{Reduction},$$

where n_{snp} is the sample size required by the SNP method, and n_{ds} is the EWMA-DS method.

CHAPTER 6

AN ILLUSTRATIVE EXAMPLE

Suppose that a manufacturer is producing break-away cables which are attachments that automatically trigger safety features on a towed load if they become disconnected with the towing vehicle. The product is designed to release at pressures under 19 newtons (N). The manufacturer is interested in showing that at least 95% of the product will not exceed the upper limit of 19N with at least a probability of 0.95. Samples of the cables breaking points were collected for 65 consecutive weeks, with 5 measurements per sample and can be seen in Table 6.1.

To test the null hypothesis of normality, a Kolmogorov-Smirnov (KS) test was conducted via python (Chakravarti et al., 1967) with a significance of 0.01. The KS test critical value at this significance for this sample size was recorded as

$$KS_c \approx \frac{1.63862}{\sqrt{n}} \approx 0.0904$$

A normal distribution was fitted to the data resulting in a KS statistic of 0.4554, which is greater than KS_c. Therefore, the null hypothesis was rejected. It can concluded that the data does not follow a normal distribution.

While fitting the model it was found that the data is best described with a Weibull distribution. Using python, the parameters of the Weibull distribution were estimated (with

$\hat{\eta} = 2.454,\ \hat{\beta} = 11.251$) and compared to a histogram (Figure 1). It can be seen that data follows the contour of the distributions p.d.f.. Furthermore, a Q-Q plot shows that the majority of data fits closely with the estimated distribution.

A KS test was conducted on the new null hypothesis that the data are from the estimated Weibull distribution. The resulting KS statistic of 0.030469 is less than the KS_c of 0.0904. Therefore, the null hypothesis cannot be rejected. It is reasonable to assume that the data comes from Weibull distribution with estimated parameters $\hat{\eta} = 2.454,\ \hat{\beta} = 11.251$.

An X-bar and R chart were created to check if the process is under statistical control. It can be observed from Figure 2 and 3 that the process is under control.

Table 6.1: Weekly data of breaking points of cables.

Week	Measurement					Week	Measurement				
1	1.9	6.3	7.9	16.0	16.5	2	6.8	7.7	7.8	8.3	16.3
3	0.6	8.0	10.5	11.2	12.3	4	3.2	3.8	9.4	9.7	11.5
5	1.4	6.1	6.3	10.8	13.4	6	3.4	9.5	9.8	14.1	18.4
7	6.6	7.3	8.2	10.8	19.4	8	6.5	6.5	13.2	18.1	21.1
9	9.1	14.1	14.1	15.5	22.0	10	5.6	6.1	6.2	9.5	16.6
11	4.8	7.5	9.8	14.5	15.3	12	8.3	9.2	11.1	14.2	17.7
13	3.9	6.5	8.4	12.9	15.0	14	2.9	7.2	7.4	10.1	11.9
15	6.7	8.3	9.7	10.4	17.1	16	2.3	5.0	5.1	11.3	12.9
17	0.5	6.0	9.6	10.5	15.7	18	2.8	5.3	5.8	10.3	10.4
19	1.9	8.9	11.2	11.2	11.3	20	5.7	8.5	9.1	10.5	11.0
21	4.2	11.1	13.1	17.0	17.6	22	6.1	7.9	9.3	16.8	17.4
23	1.9	5.9	9.6	10.6	10.9	24	4.8	5.6	7.1	7.3	12.9
25	3.9	5.7	7.5	9.2	15.7	26	2.5	4.5	4.8	6.7	18.2
27	5.3	7.4	7.8	8.7	11.2	28	8.9	11.5	11.9	12.3	14.9
29	5.9	8.1	8.3	11.2	15.3	30	3.4	9.3	9.9	13.6	17.5
31	2.1	6.6	9.1	10.4	18.0	32	10.0	10.6	11.6	12.7	12.8
33	8.2	8.6	12.0	12.6	13.8	34	5.3	7.8	9.8	10.0	13.2
35	2.8	4.2	8.1	8.6	14.2	36	7.4	8.0	8.2	12.5	17.7
37	6.6	8.4	9.7	10.1	16.1	38	7.1	8.6	9.3	10.6	14.3
39	6.8	10.0	10.2	10.9	13.9	40	6.0	8.0	10.4	13.1	15.8
41	3.5	8.9	9.4	9.7	15.0	42	6.0	6.6	8.7	11.0	14.8
43	4.3	8.2	9.2	11.2	13.4	44	7.6	9.3	9.5	10.6	16.0
45	3.0	7.6	8.2	13.7	15.7	46	5.9	7.0	12.5	14.6	17.0
47	3.5	5.7	8.0	9.8	12.5	48	6.5	9.7	9.9	15.8	18.7
49	4.4	7.4	14.8	15.0	18.1	50	7.3	7.8	8.8	12.5	15.2
51	5.6	5.9	9.0	9.2	12.6	52	7.6	8.3	9.7	13.6	13.6
53	2.7	8.0	9.3	12.5	13.8	54	4.1	6.6	13.0	13.5	16.1
55	9.8	10.3	13.5	13.6	14.9	56	3.7	5.0	5.9	7.5	17.9
57	7.5	10.5	12.3	12.4	15.0	58	4.9	7.1	8.7	13.5	13.5
59	8.2	11.2	11.3	15.6	16.9	60	2.2	2.9	10.2	14.1	19.8
61	0.3	8.4	10.8	12.6	18.2	62	6.4	10.2	12.5	15.6	17.9
63	5.9	7.4	12.5	13.3	20.2	64	5.7	6.1	10.7	18.5	21.6
65	7.4	7.9	9.7	13.0	14.4						

Note: The data was generated from a Weibull distribution (with $\eta = 2.478$ and $\beta = 10.78$) via inverse CDF method, where the random number generation was done with the package NumPy (Harris et al., 2020).

Figure 6.1: Histogram of data with fitted Weibull Distribution ($\hat{\eta} = 2.454,\ \hat{\beta} = 11.251$).

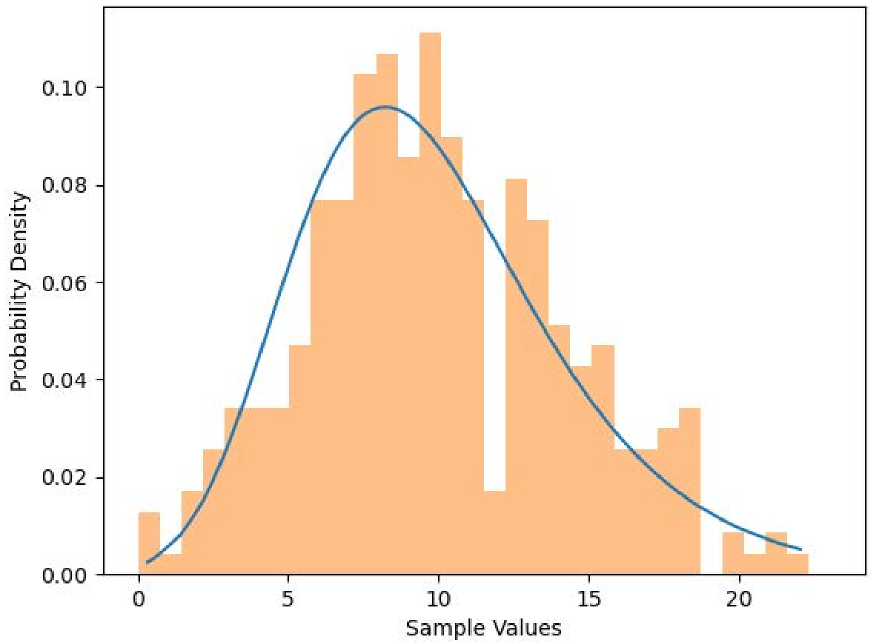


Figure 6.2: QQ-plot of data for fitted Weibull Distribution ($\hat{\eta} = 2.454,\ \hat{\beta} = 11.251$).

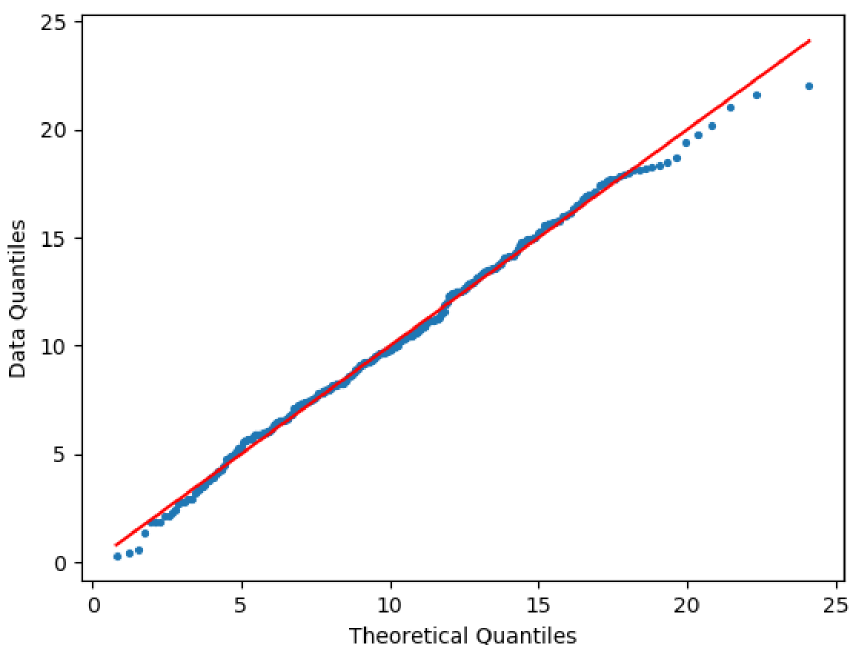


Figure 6.3: X-bar chart for the breaking point data.

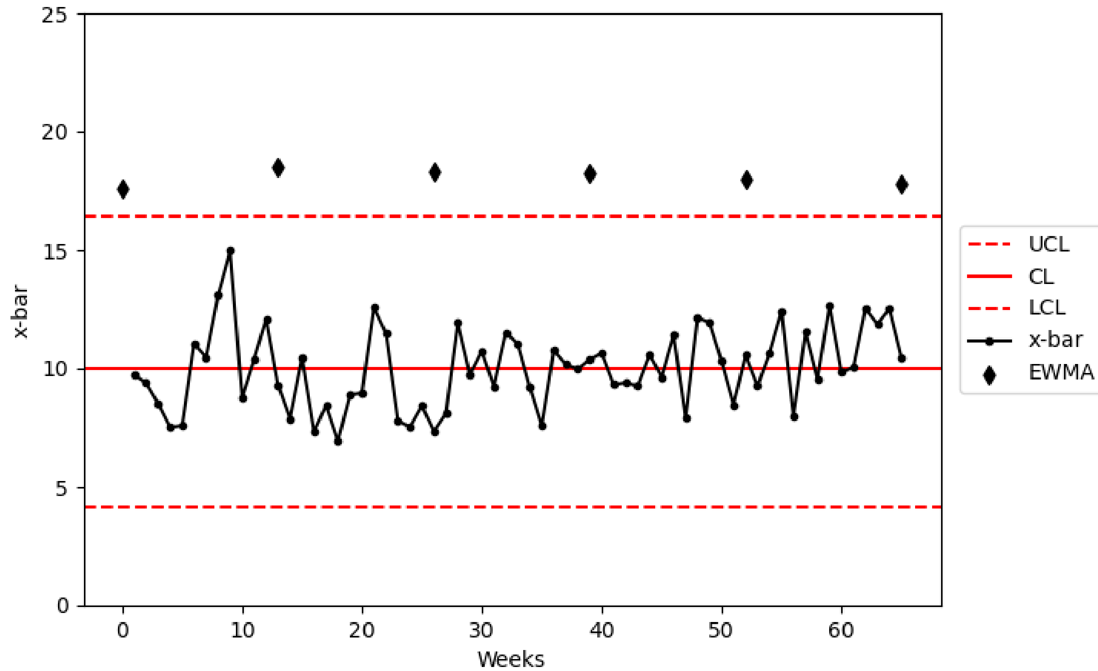
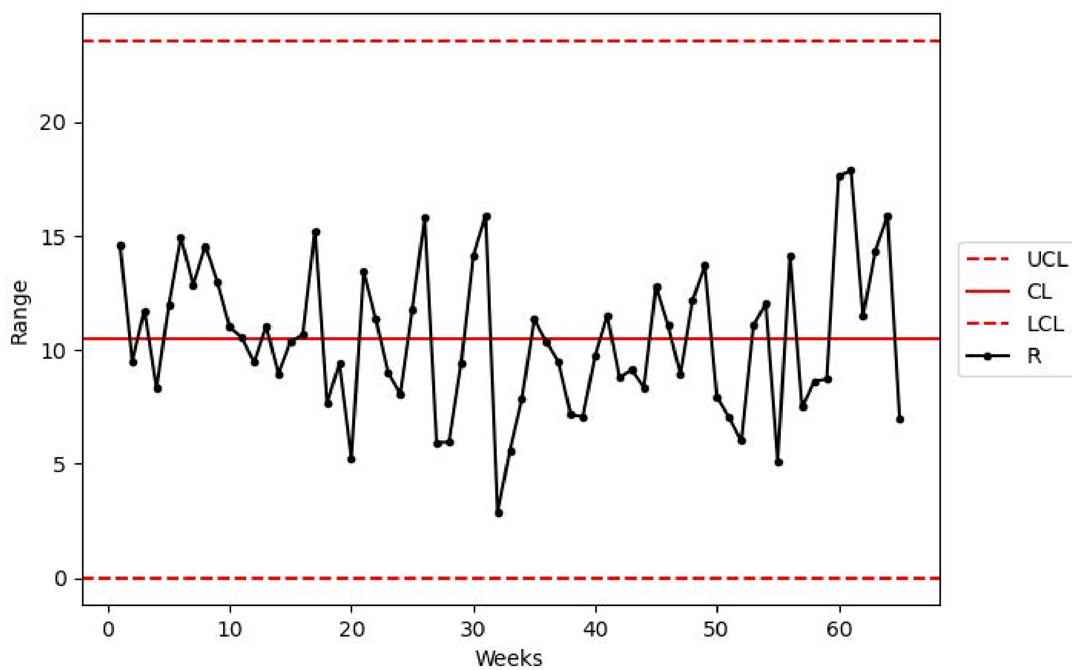


Figure 6.4: R-chart for the breaking point data



With the requirement of statistical control satisfied, the upper-sided distribution specific algorithm was used to calculate the required samples sizes for the EWMA-DS method. The data was segmented into five sections of thirteen weeks each, with $t = 1$ relating to weeks 1-13, $t = 2$ relating to weeks 14-26, and so on. For $t = 1, 2, 3, 4, 5$ the required sample sizes are $n_1 = 59, n_2 = 30, n_3 = 27, n_4 = 24, n_5 = 24$, respectively. A typical lambda value of 0.2 is used here. The upper bound of the tolerance limits were then calculated for each time period:

Time Period 0: The initial value, $X_{max\ 0}$, is equal to the estimated distributions quantile value,

$$\mathrm{EWMA}_0 = c_1 = 17.59.$$

Time Period 1: The upper bound of the tolerance limit in the first time period is

$$\mathrm{EWMA}_1 = 0.2 \cdot X_{max\ 1} + 0.8 \cdot \mathrm{EWMA}_0 = 18.48,$$

where the maximum value is 22.04 in the first 13 weeks.

Time Period 2: The upper bound is

$$\mathrm{EWMA}_2 = 0.2 \cdot X_{max\ 2} + 0.8 \cdot \mathrm{EWMA}_1 = 18.31,$$

where the maximum of the 30 samples selected is 17.61 in the second set of 13 weeks.

Time Period 3: The upper bound is

$$\text{EWMA}_3 = 0.2 \cdot X_{max\ 3} + 0.8 \cdot \text{EWMA}_2 = 18.25,$$

where the maximum of the 27 samples selected is 18.03 in the third set of 13 weeks.

Time Period 4: The upper bound is

$$\text{EWMA}_4 = 0.2 \cdot X_{max\ 4} + 0.8 \cdot \text{EWMA}_3 = 18.00,$$

where the maximum of the 24 samples selected is 17.00 in the fourth set of 13 weeks.

Time Period 5: The upper bound is

$$\text{EWMA}_5 = 0.2 \cdot X_{max\ 5} + 0.8 \cdot \text{EWMA}_4 = 17.79,$$

where the maximum of the 24 samples selected is 16.94 in the fifth set of 13 weeks.

In Figure 2 each EWMA_t is plotted at the 13 week time periods, with EWMA_0 being plotted initially. The upper-sided EWMA tolerance limits display clear variation in the tolerance of the process over the six time periods; however the tolerance limit is still within the requirement.

Now that the distribution of the process has been estimated from the initial samples, the EWMA-DS sample sizes can be used to take samples instead of the regular 5 samples per week. If the sampling schedule is consistent with time periods, every thirteen weeks will only require 24 samples or less, rather than 65 every thirteen weeks.

CHAPTER 7

CONCLUSION AND DISCUSSION

In this research, the EWMA distribution-specific algorithm was reconstructed and used to calculate sample sizes required for the construction of upper-sided tolerance intervals for various distribution. Two studies were conducted to compare the sample sizes required and the length of the tolerance intervals. The comparison was done with three existing methods: SNP, CNP, and EWMA-IFR. Of the four methods, EWMA-DS always resulted in the lowest required sample sizes. Furthermore, it has produced the shortest average interval lengths, and the smallest standard deviations of interval lengths for most of the distributions considered.

Given a process, the EWMA-DS method requires that the process has a known distributions and is in control. If the process is not in control, the method cannot be used. The ability to maintain the knowledge of the distribution of the process is lost when control is lost. However, if control is lost and then regained, the use of the EWMA-DS method can function again. After a short number of iterations, the method will offer its reduced sample sizes and small interval lengths.

If the distribution of the process remains completely unidentified, the EWMA-DS method cannot be used despite the processes state of control. However, if the process maintains its control and can be potentially identified by a number of distributions, the method can still be utilized. If the distribution with the most conservative sample sizes is chosen from the potentially identified distributions (such that the sample sizes required by it are larger than the required sample sizes of the other distributions being considered), the EWMA-DS method will still have the lowest required sample sizes of its competitors.

When a process is under control and the distribution of the process can be identified, the intervals constructed via the EWMA-DS method require the lowest samples sizes, maintain their accuracy, and are in most cases better than the intervals constructed by the competing methods.

APPENDIX A

TABLES

A.1 Sample Size Tables

Table A.1: Sample sizes required for $X \sim exp(\beta = 1)$.

		q											
		85% (1.89711998) λ			90% (2.30258509) λ			95% (2.99573227) λ			98% (3.91202301) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	7	7	8	11	12	12	23	24	25	58	60	63
90%	3	6	7	7	10	10	11	19	21	22	50	53	57
	4	6	6	7	8	9	10	19	19	22	46	50	54
	5	5	6	6	9	10	11	17	19	21	45	49	54
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	9	9	9	13	14	15	27	28	30	69	72	76
95%	3	7	8	9	11	12	13	23	25	26	58	62	67
	4	6	7	8	10	11	12	21	22	25	53	58	64
	5	6	7	7	10	10	12	19	22	25	50	56	62
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	10	11	11	16	17	18	33	34	36	83	88	93
98%	3	8	9	10	13	14	15	26	29	32	68	73	80
	4	8	8	9	11	13	14	24	26	29	61	67	75
	5	7	8	9	11	12	14	23	26	29	57	65	73
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	12	12	13	18	19	20	37	39	41	94	99	105
99%	3	9	10	11	14	15	17	29	32	35	75	82	89
	4	8	9	10	13	14	16	27	29	33	67	74	83
	5	8	9	10	12	14	15	24	28	32	62	71	81

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.2: Sample sizes required for $X \sim Normal(\mu = 0, \sigma = 1)$.

		q											
		85% (1.03643339) λ			90% (1.28155157) λ			95% (1.64485363) λ			98% (2.05374891) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
90%	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	8	8	9	12	13	13	25	26	27	61	64	67
	3	7	8	8	11	11	12	21	23	24	54	56	60
	4	7	7	7	10	11	12	20	21	23	50	54	58
95%	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	10	10	11	15	15	16	29	31	32	73	76	80
	3	8	9	9	12	13	14	25	26	29	62	66	71
	4	7	8	9	11	12	14	23	25	27	56	62	68
98%	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	11	12	13	17	18	19	35	37	39	88	93	98
	3	10	10	11	15	16	17	29	31	34	72	78	85
	4	8	9	10	13	14	16	26	29	32	65	71	79
99%	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	13	14	15	20	21	22	40	42	45	100	102	111
	3	10	11	12	16	17	19	32	35	38	79	88	95
	4	10	10	12	14	16	17	28	31	35	71	79	88

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.3: Sample sizes required for $X \sim invGaussian(\mu = 1, \theta = 0.5)$.

		q											
		85% (1.76636) λ			90% (2.35826) λ			95% (3.5526) λ			98% (5.43708) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	6	6	7	10	11	11	21	22	23	55	57	60
90%	3	5	6	6	8	9	10	18	19	21	46	50	54
	4	5	5	6	8	8	9	16	18	19	43	47	51
	5	5	5	6	7	8	9	16	17	20	42	45	50
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	8	8	8	12	12	13	25	26	28	65	68	72
95%	3	6	7	8	10	11	12	21	22	24	54	59	63
	4	5	6	7	8	10	11	19	21	23	50	54	60
	5	5	6	6	9	9	10	17	20	23	47	52	59
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	9	9	10	14	15	16	30	32	34	79	83	88
98%	3	7	8	9	12	12	14	24	26	29	64	69	76
	4	7	7	8	10	12	13	22	24	27	57	64	71
	5	6	7	8	9	11	12	21	24	27	53	60	69
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	10	11	11	16	17	18	34	36	38	89	94	100
99%	3	8	9	10	13	14	15	27	30	33	71	77	84
	4	7	8	9	11	13	15	24	26	30	63	70	79
	5	7	7	9	11	12	14	22	26	29	58	67	77

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.4: Sample sizes required for $X \sim invGaussian(\mu = 1, \theta = 5)$.

		q											
		85% (1.42960) λ			90% (1.58836) λ			95% (1.85279) λ			98% (2.19511) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	7	8	8	11	12	13	23	24	25	58	61	64
90%	3	6	6	7	10	10	11	20	21	23	50	53	57
	4	6	7	7	9	10	10	18	20	21	47	50	55
	5	6	7	7	9	9	11	18	19	22	44	49	54
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	9	9	10	13	14	15	27	29	30	69	72	76
95%	3	7	8	8	12	12	13	23	24	27	58	63	67
	4	7	7	8	10	11	12	21	23	25	53	58	64
	5	6	7	8	10	11	12	20	22	25	51	56	63
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	10	11	12	16	17	18	33	35	37	84	88	93
98%	3	9	9	10	13	14	15	27	29	32	68	74	80
	4	8	9	9	12	13	15	24	27	29	61	67	76
	5	7	8	9	11	13	14	23	25	29	57	65	73
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	12	13	13	18	19	21	37	39	42	94	100	106
99%	3	9	10	11	15	16	17	30	33	35	76	81	89
	4	9	9	11	13	14	16	26	29	33	67	75	84
	5	8	9	10	12	14	16	25	28	32	62	71	81

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.5: Sample sizes required for $X \sim invGaussian(\mu = 5, \theta = 1)$.

		q											
		85% (7.94159) λ			90% (11.9578) λ			95% (21.3689) λ			98% (38.6246) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	5	6	6	9	10	10	20	21	22	52	55	58
90%	3	5	5	6	8	8	9	16	17	19	45	47	51
	4	4	4	5	6	7	8	15	17	18	41	45	49
	5	4	5	5	7	7	8	14	16	18	39	43	48
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	7	7	8	11	11	12	23	25	26	62	66	69
95%	3	5	6	6	9	10	10	19	20	23	52	55	61
	4	5	5	6	7	9	10	18	20	21	47	52	58
	5	5	6	6	8	8	10	16	19	21	44	50	56
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	8	8	9	13	14	15	28	30	32	76	80	85
98%	3	6	7	8	10	11	12	23	25	27	61	66	72
	4	6	7	7	9	10	12	20	22	25	54	61	68
	5	5	6	7	9	10	11	19	22	25	51	58	67
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	9	10	10	15	16	17	32	34	36	85	90	96
99%	3	7	8	9	11	12	14	25	27	30	68	74	81
	4	7	7	8	11	12	13	22	25	29	60	67	76
	5	6	7	8	10	11	13	21	24	27	56	64	73

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.6: Sample sizes required for $X \sim Weibull(\eta = 1.5, \beta = 1)$.

		q											
		85% (1.53248606) λ			90% (1.74372151) λ			95% (2.07811064) λ			98% (2.4827573) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	8	8	8	12	12	13	24	25	26	60	62	65
90%	3	6	7	8	10	11	11	20	22	23	52	55	59
	4	6	7	7	9	10	11	20	20	23	48	52	56
	5	6	6	7	9	10	11	19	20	22	46	51	56
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	9	10	10	14	15	15	28	30	31	71	74	78
95%	3	8	8	9	12	12	14	24	25	28	60	64	69
	4	7	7	8	10	12	13	22	24	26	55	60	66
	5	6	8	8	10	11	12	20	23	25	52	58	64
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	11	11	12	17	18	19	34	36	38	86	90	96
98%	3	9	10	10	13	14	16	28	30	33	70	76	82
	4	8	9	10	13	14	15	25	28	31	63	70	78
	5	7	8	10	12	13	15	23	26	30	60	67	75
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	12	13	14	19	20	21	39	41	43	97	102	109
99%	3	10	11	12	15	16	18	30	33	37	78	84	92
	4	9	9	11	13	15	17	28	31	34	69	77	86
	5	8	10	10	13	14	16	25	29	33	64	73	83

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.7: Sample sizes required for $X \sim Weibull(\eta = 0.5, \beta = 1)$.

		q											
		85% (3.59906424) λ			90% (5.30189811) λ			95% (8.97441185) λ			98% (15.3039239) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	6	6	6	10	10	11	20	21	22	53	55	58
90%	3	5	5	6	8	9	9	17	18	20	44	48	51
	4	4	5	6	7	8	9	16	17	19	41	44	49
	5	4	5	5	7	7	8	14	17	18	39	44	48
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	7	8	8	11	12	13	24	25	27	63	66	69
95%	3	6	6	7	9	10	11	20	22	23	52	56	61
	4	5	6	6	9	9	10	18	19	22	47	52	58
	5	5	5	7	8	9	10	17	19	22	44	50	56
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	8	9	10	14	14	15	29	31	33	76	80	85
98%	3	7	7	8	11	12	13	23	25	27	61	66	73
	4	6	7	8	9	11	12	21	23	26	55	61	68
	5	6	7	7	9	10	12	19	22	26	51	58	66
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	10	10	11	16	16	18	33	35	37	86	91	96
99%	3	7	9	9	12	14	14	26	28	31	68	74	81
	4	7	7	9	11	12	14	23	26	29	60	67	76
	5	6	7	8	10	11	13	21	24	28	56	64	73

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.8: Sample sizes required for $X \sim Lognormal(\mu = 0, \sigma = 0.25)$.

		q											
		85% (1.29577419) λ			90% (1.37766204) λ			95% (1.50864728) λ			98% (1.67102503) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	8	8	8	12	12	13	24	25	26	59	61	64
90%	3	6	7	8	10	11	11	20	21	23	51	54	58
	4	6	6	7	9	10	11	19	21	22	47	51	56
	5	6	7	7	9	10	11	18	20	22	46	50	54
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	9	10	10	14	14	15	28	29	31	70	73	77
95%	3	8	8	9	11	13	14	23	25	27	59	63	68
	4	7	7	8	11	11	12	22	24	26	54	59	65
	5	6	8	8	10	11	13	20	22	25	51	57	63
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	11	11	12	17	17	18	34	35	38	85	89	94
98%	3	9	10	10	13	15	16	27	30	32	69	75	81
	4	8	9	10	12	13	15	25	27	30	62	68	77
	5	7	8	10	12	13	15	23	26	30	58	66	74
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	12	13	14	19	20	21	38	40	43	96	101	107
99%	3	10	11	11	15	16	18	30	33	36	76	83	91
	4	9	9	11	13	15	16	27	30	34	68	76	85
	5	8	9	11	13	14	16	26	29	32	64	72	82

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.9: Sample sizes required for $X \sim Lognormal(\mu = 8, \sigma = 2)$.

		q											
		85% (23691.385) λ			90% (38680.974) λ			95% (79994.027) λ			98% (181225.57) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	5	5	6	8	9	9	18	18	19	45	48	50
90%	3	4	5	5	7	7	8	14	16	17	38	40	44
	4	4	4	4	6	7	7	13	14	16	33	37	41
	5	3	4	5	6	6	8	12	13	15	32	36	40
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	6	7	7	10	10	11	21	22	23	54	57	61
95%	3	5	5	6	8	9	9	17	18	20	44	48	52
	4	5	5	6	7	8	9	15	17	19	40	43	49
	5	4	5	5	6	7	9	14	16	18	36	42	47
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	8	8	8	12	13	13	25	27	28	66	70	74
98%	3	6	6	8	9	10	12	20	22	24	53	57	63
	4	5	6	6	9	9	10	18	19	22	46	52	58
	5	5	6	7	8	9	10	16	19	22	42	48	56
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	9	9	10	14	15	16	30	31	33	76	79	84
99%	3	6	7	8	10	11	12	21	24	27	58	64	70
	4	6	7	7	9	10	12	20	22	24	51	57	65
	5	5	6	7	9	10	11	18	20	24	47	54	63

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.10: Sample sizes required for $X \sim Loglog(\alpha = 1, \beta = 2)$.

		q											
		85% (2.38047614) λ			90% (3.0) λ			95% (4.35889894) λ			98% (7.0) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
90%	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	6	6	7	10	10	11	19	20	21	49	51	53
	3	5	6	6	8	9	9	16	17	19	40	43	47
	4	5	5	6	7	7	9	15	16	17	37	40	44
	5	4	5	5	7	8	8	14	15	17	34	39	43
95%	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	7	8	8	11	12	13	23	24	26	58	61	64
	3	6	6	7	10	10	11	19	20	22	47	51	56
	4	6	6	7	8	9	10	17	19	20	43	47	52
	5	5	6	6	8	9	10	15	18	20	39	44	50
98%	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	9	9	10	14	14	15	28	29	31	70	74	79
	3	7	8	8	11	12	13	22	24	26	56	61	66
	4	6	7	8	9	11	12	20	22	25	50	55	62
	5	6	7	8	9	10	12	18	21	24	46	52	60
99%	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	10	11	11	16	17	18	32	33	35	80	84	89
	3	8	8	10	12	13	14	24	27	30	62	68	75
	4	7	8	9	10	12	14	22	24	27	54	61	69
	5	6	7	8	10	11	13	20	23	26	51	57	66

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.11: Sample sizes required for $X \sim Loglog(\alpha = 2, \beta = 10)$.

		q											
		85% (2.37882647) λ			90% (2.49146188) λ			95% (2.6847593) λ			98% (2.95154632) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	7	8	8	11	12	12	23	23	25	56	59	61
90%	3	6	6	7	10	10	11	19	21	22	48	51	55
	4	6	7	7	9	9	10	17	19	21	45	48	53
	5	6	7	7	8	10	10	17	19	20	42	47	51
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	9	9	10	13	14	15	27	28	29	67	70	74
95%	3	7	8	8	11	12	13	22	24	26	56	60	65
	4	7	7	8	10	11	12	20	22	24	51	56	61
	5	6	7	8	10	10	12	19	21	24	48	53	60
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	10	11	12	16	17	18	32	34	36	81	85	90
98%	3	9	9	10	13	14	15	26	28	31	65	71	77
	4	7	8	9	11	13	14	23	26	28	59	65	73
	5	7	8	9	11	12	14	22	24	28	55	62	70
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	12	12	13	18	19	20	36	38	41	91	96	102
99%	3	9	10	11	14	15	17	29	32	34	73	79	86
	4	8	10	11	13	14	16	26	28	32	64	72	81
	5	8	9	10	12	14	15	23	27	31	60	68	78

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.12: Sample sizes required for $X \sim \chi^2(df = 2)$.

		q											
		85% (3.79423997) λ			90% (4.60517019) λ			95% (5.99146455) λ			98% (7.82404601) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	7	7	8	11	12	12	23	24	25	58	60	63
90%	3	6	7	7	10	10	11	19	21	22	50	53	57
	4	6	6	7	8	9	10	19	19	22	46	50	54
	5	5	6	6	9	10	11	18	19	21	45	49	54
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	9	9	9	13	14	15	27	28	30	69	72	76
95%	3	7	8	9	11	12	13	23	25	26	58	62	67
	4	6	7	8	10	11	12	21	22	25	53	58	64
	5	6	7	7	10	10	12	19	22	25	50	56	62
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	10	11	11	16	17	18	33	34	36	83	88	93
98%	3	8	9	10	13	14	15	26	29	32	68	73	80
	4	8	8	9	11	13	14	24	26	29	61	67	75
	5	7	8	9	11	12	14	23	26	29	57	65	73
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	12	12	13	18	19	20	37	39	41	94	99	105
99%	3	9	10	11	14	15	17	29	32	35	75	82	89
	4	8	9	10	13	14	16	27	29	33	67	74	83
	5	8	9	10	12	14	15	24	28	32	62	71	81

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.13: Sample sizes required for $X \sim \chi^2(df = 6)$.

		q											
		85% (9.44610313) λ			90% (10.64464068) λ			95% (12.5915872) λ			98% (15.0332078) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	7	8	8	12	12	13	24	24	26	59	61	64
90%	3	7	7	7	10	11	11	20	22	23	51	54	58
	4	6	6	7	9	10	11	19	20	22	47	51	56
	5	6	6	7	9	9	11	18	20	21	46	50	54
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	9	9	10	14	14	15	28	29	31	70	73	77
95%	3	7	8	9	11	12	13	23	25	27	59	63	68
	4	7	8	8	11	12	13	21	23	26	54	59	65
	5	7	7	8	10	11	12	21	23	25	51	57	63
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	11	11	12	16	17	18	33	35	37	85	89	94
98%	3	8	10	10	14	15	16	28	30	32	69	75	81
	4	8	8	10	12	13	15	24	27	30	62	68	77
	5	7	9	9	11	13	14	23	26	30	58	66	74
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	12	13	14	19	20	21	38	40	42	96	101	107
99%	3	10	10	11	15	16	18	30	33	36	76	83	91
	4	8	10	11	13	15	16	27	30	34	68	75	84
	5	8	9	10	12	14	16	25	28	32	63	72	82

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.14: Sample sizes required for $X \sim Cauchy(x_0 = 0, \gamma = 0.5)$.

		q											
		85% (0.9813) λ			90% (1.5388) λ			95% (3.1569) λ			98% (7.9472) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	5	6	6	8	9	9	17	17	18	41	43	46
90%	3	5	5	5	7	7	8	13	14	16	33	36	38
	4	4	4	5	6	6	7	11	13	14	29	32	36
	5	4	4	5	5	6	7	11	12	14	27	31	35
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	7	7	7	10	10	11	20	21	22	49	52	55
95%	3	5	6	6	8	9	9	15	17	18	40	42	47
	4	5	5	6	7	7	9	14	15	17	34	39	43
	5	4	5	6	6	8	8	13	14	17	32	36	41
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	8	8	9	12	13	13	25	25	27	64	64	68
98%	3	6	7	7	9	10	11	18	20	22	44	51	56
	4	6	6	7	8	9	10	16	19	21	41	46	52
	5	5	6	7	8	8	10	15	17	19	37	43	49
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	9	10	10	14	14	15	30	30	31	76	76	77
99%	3	7	7	8	10	12	13	19	22	25	47	55	63
	4	6	7	8	9	10	11	18	20	23	45	51	58
	5	6	6	7	8	9	11	16	19	22	42	47	55

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

Table A.15: Sample sizes required for $X \sim Cauchy(x_0 = 0, \gamma = 1.5)$.

		q											
		85% (2.9439) λ			90% (4.6165) λ			95% (9.4706) λ			98% (23.8418) λ		
1-α	t	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
	1	15	15	15	22	22	22	45	45	45	114	114	114
	2	5	6	6	8	9	9	17	17	18	41	43	46
90%	3	5	5	5	7	7	8	13	14	16	33	36	38
	4	4	4	5	6	6	7	11	13	14	29	32	36
	5	4	4	5	5	6	7	11	12	14	27	31	35
	1	19	19	19	29	29	29	59	59	59	149	149	149
	2	7	7	7	10	10	11	20	21	22	49	52	55
95%	3	5	6	6	8	9	9	15	17	18	40	42	47
	4	5	5	6	7	7	9	14	15	17	34	39	43
	5	4	5	6	6	8	8	13	14	17	32	36	41
	1	25	25	25	38	38	38	77	77	77	194	194	194
	2	8	8	9	12	13	13	25	25	27	64	64	68
98%	3	6	7	7	9	10	11	18	20	22	44	51	56
	4	6	6	7	8	9	10	16	19	21	41	46	52
	5	5	6	7	8	8	10	15	17	19	37	43	49
	1	29	29	29	44	44	44	90	90	90	228	228	228
	2	9	10	10	14	14	15	30	30	31	76	76	77
99%	3	7	7	8	10	12	13	19	22	25	47	55	63
	4	6	7	8	9	10	11	18	20	23	45	51	58
	5	6	6	7	8	9	11	16	19	22	42	47	55

Note: Entries are sample sizes required to calculate the Upper-Sided EWMA Tolerance Limit for the specified distribution, at time t. Numbers in parentheses are the q^{th} quantiles of the specified distribution.

A.2 Simulation Study Tables

Table A.16: Mean and standard deviation comparison for IG and $Weibull$ at $\alpha = 0.1$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Inverse Gaussian	EWMA-DS	1.677	1.697	1.812	1.828	1.941	1.959	2.069	2.085
($\mu = 1,\ \theta = 5$)		(0.074)	(0.087)	(0.184)	(0.197)	(0.073)	(0.087)	(0.183)	(0.194)
	EWMA-IFR	1.707	1.753	1.873	1.928	1.969	2.017	2.134	2.189
		(0.074)	(0.086)	(0.184)	(0.195)	(0.073)	(0.087)	(0.182)	(0.196)
	CNP	1.833	1.791	1.832	1.791	2.097	2.059	2.095	2.056
		(0.195)	(0.143)	(0.198)	(0.145)	(0.193)	(0.142)	(0.194)	(0.144)
	SNP	2.119	2.103	2.105	2.111	2.367	2.374	2.366	2.37
		(0.473)	(0.465)	(0.467)	(0.468)	(0.464)	(0.475)	(0.464)	(0.473)
Inverse Gaussian	EWMA-DS	16.149	16.716	21.913	21.768	26.437	27.291	33.705	34.07
($\mu = 5,\ \theta = 1$)		(4.166)	(4.53)	(9.899)	(9.721)	(4.763)	(5.319)	(11.808)	(12.189)
	EWMA-IFR	18.023	20.398	26.524	29.018	28.708	31.677	38.95	42.392
		(4.607)	(5.266)	(11.178)	(11.472)	(5.183)	(5.959)	(12.717)	(13.457)
	CNP	21.491	19.364	21.303	19.243	33.94	31.504	33.973	31.511
		(8.644)	(5.877)	(8.476)	(5.792)	(10.793)	(7.678)	(10.902)	(7.719)
	SNP	38.211	38.335	38.4	37.783	54.102	53.093	53.178	53.742
		(30.708)	(30.594)	(29.813)	(29.599)	(34.819)	(33.352)	(33.83)	(34.646)
Weibull	EWMA-DS	1.852	1.872	2.006	2.034	2.178	2.203	2.327	2.346
($\eta = 1.5,\ \beta = 1$)		(0.087)	(0.104)	(0.216)	(0.233)	(0.083)	(0.098)	(0.208)	(0.217)
	EWMA-IFR	1.884	1.938	2.082	2.148	2.211	2.263	2.4	2.459
		(0.085)	(0.101)	(0.212)	(0.229)	(0.081)	(0.095)	(0.205)	(0.213)
	CNP	2.045	1.997	2.046	2.0	2.369	2.324	2.365	2.322
		(0.238)	(0.177)	(0.239)	(0.178)	(0.224)	(0.168)	(0.229)	(0.167)
	SNP	2.364	2.357	2.361	2.368	2.663	2.661	2.666	2.66
		(0.531)	(0.53)	(0.531)	(0.525)	(0.505)	(0.504)	(0.502)	(0.494)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.17: Mean and standard deviation comparison for IG and $Weibull$ at $\alpha = 0.05$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Inverse Gaussian	EWMA-DS	1.698	1.721	1.857	1.876	1.96	1.984	2.121	2.139
($\mu = 1,\ \theta = 5$)		(0.074)	(0.087)	(0.186)	(0.197)	(0.074)	(0.088)	(0.187)	(0.194)
	EWMA-IFR	1.725	1.778	1.924	1.988	1.99	2.043	2.187	2.245
		(0.074)	(0.087)	(0.185)	(0.195)	(0.074)	(0.088)	(0.187)	(0.193)
	CNP	1.932	1.792	1.934	1.793	2.2	2.063	2.195	2.058
		(0.195)	(0.125)	(0.196)	(0.127)	(0.197)	(0.126)	(0.196)	(0.125)
	SNP	2.209	2.209	2.217	2.214	2.479	2.469	2.466	2.475
		(0.465)	(0.468)	(0.479)	(0.471)	(0.478)	(0.46)	(0.474)	(0.476)
Inverse Gaussian	EWMA-DS	16.95	17.8	23.811	24.269	27.493	28.621	36.626	36.876
($\mu = 5,\ \theta = 1$)		(4.099)	(4.612)	(10.466)	(10.41)	(4.945)	(5.595)	(12.535)	(12.519)
	EWMA-IFR	18.874	21.612	29.064	31.819	29.826	33.044	42.232	45.641
		(4.567)	(5.367)	(11.661)	(11.952)	(5.305)	(6.16)	(13.517)	(13.764)
	CNP	25.923	19.354	25.937	19.284	39.641	31.517	39.739	31.43
		(9.597)	(5.116)	(9.583)	(5.108)	(11.76)	(6.745)	(11.85)	(6.814)
	SNP	43.8	44.217	43.655	43.232	59.632	60.071	60.235	59.524
		(31.375)	(32.383)	(31.39)	(31.168)	(34.879)	(36.403)	(36.25)	(35.531)
Weibull	EWMA-DS	1.875	1.9	2.069	2.087	2.201	2.227	2.387	2.406
($\eta = 1.5,\ \beta = 1$)		(0.085)	(0.102)	(0.213)	(0.228)	(0.081)	(0.098)	(0.202)	(0.212)
	EWMA-IFR	1.905	1.966	2.14	2.212	2.232	2.289	2.456	2.522
		(0.083)	(0.099)	(0.209)	(0.221)	(0.08)	(0.095)	(0.199)	(0.207)
	CNP	2.168	1.998	2.173	1.999	2.483	2.324	2.483	2.325
		(0.234)	(0.156)	(0.235)	(0.158)	(0.224)	(0.148)	(0.224)	(0.147)
	SNP	2.476	2.467	2.476	2.481	2.771	2.771	2.767	2.771
		(0.518)	(0.509)	(0.515)	(0.516)	(0.494)	(0.493)	(0.492)	(0.495)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.18: Mean and standard deviation comparison for IG and $Weibull$ at $\alpha = 0.01$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Inverse Gaussian	EWMA-DS	1.73	1.762	1.938	1.965	1.993	2.025	2.203	2.229
($\mu = 1,\ \theta = 5$)		(0.074)	(0.086)	(0.186)	(0.197)	(0.074)	(0.087)	(0.188)	(0.196)
	EWMA-IFR	1.759	1.822	2.012	2.078	2.022	2.084	2.273	2.344
		(0.074)	(0.086)	(0.187)	(0.195)	(0.075)	(0.087)	(0.187)	(0.195)
	CNP	1.922	1.836	1.922	1.836	2.188	2.104	2.188	2.104
		(0.157)	(0.108)	(0.158)	(0.108)	(0.155)	(0.107)	(0.154)	(0.108)
	SNP	2.369	2.356	2.364	2.359	2.633	2.639	2.628	2.625
		(0.475)	(0.464)	(0.47)	(0.47)	(0.479)	(0.479)	(0.47)	(0.475)
Inverse Gaussian	EWMA-DS	18.461	19.78	28.083	28.43	29.429	31.07	41.37	42.143
($\mu = 5,\ \theta = 1$)		(4.485)	(4.979)	(11.527)	(11.196)	(5.17)	(5.827)	(13.284)	(13.265)
	EWMA-IFR	20.721	23.804	33.747	36.938	31.915	35.724	47.626	51.76
		(4.945)	(5.646)	(12.738)	(12.581)	(5.536)	(6.392)	(14.052)	(14.59)
	CNP	24.959	20.964	24.933	20.957	38.729	33.843	38.766	33.758
		(7.406)	(4.625)	(7.404)	(4.558)	(9.236)	(5.935)	(9.241)	(5.912)
	SNP	53.219	53.055	52.49	52.12	70.363	70.839	70.653	70.888
		(34.018)	(34.169)	(33.834)	(32.416)	(36.84)	(37.617)	(37.282)	(38.008)
Weibull	EWMA-DS	1.912	1.952	2.161	2.193	2.237	2.274	2.472	2.507
($\eta = 1.5,\ \beta = 1$)		(0.082)	(0.099)	(0.208)	(0.222)	(0.078)	(0.095)	(0.198)	(0.211)
	EWMA-IFR	1.944	2.018	2.243	2.317	2.268	2.337	2.543	2.624
		(0.081)	(0.096)	(0.204)	(0.216)	(0.078)	(0.093)	(0.194)	(0.205)
	CNP	2.159	2.056	2.16	2.056	2.474	2.377	2.472	2.377
		(0.187)	(0.132)	(0.185)	(0.132)	(0.18)	(0.126)	(0.179)	(0.125)
	SNP	2.652	2.645	2.657	2.643	2.938	2.935	2.936	2.934
		(0.513)	(0.51)	(0.503)	(0.506)	(0.481)	(0.476)	(0.481)	(0.476)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.19: Mean and standard deviation comparison for $Weibull$ and χ^2 at $\alpha = 0.1$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Weibull	EWMA-DS	6.888	7.139	9.284	9.26	10.871	11.221	13.737	13.852
($\eta = 0.5,\ \beta = 1$)		(1.537)	(1.717)	(3.922)	(3.923)	(1.878)	(2.129)	(4.69)	(4.769)
	EWMA-IFR	7.532	8.451	10.865	11.886	11.682	12.798	15.573	16.828
		(1.68)	(1.958)	(4.326)	(4.537)	(2.059)	(2.388)	(5.05)	(5.235)
	CNP	8.915	8.168	8.99	8.225	13.549	12.673	13.589	12.693
		(3.162)	(2.188)	(3.268)	(2.262)	(4.015)	(2.825)	(4.044)	(2.795)
	SNP	14.919	15.302	15.325	15.231	20.688	20.937	20.744	21.02
		(11.03)	(11.759)	(11.522)	(11.485)	(13.621)	(13.175)	(13.372)	(13.72)
Chi-Square	EWMA-DS	5.084	5.174	5.774	5.858	6.466	6.573	7.169	7.272
($df = 2$)		(0.394)	(0.46)	(0.995)	(1.031)	(0.397)	(0.462)	(1.004)	(1.081)
	EWMA-IFR	5.235	5.487	6.122	6.404	6.631	6.879	7.552	7.86
		(0.398)	(0.464)	(0.994)	(1.039)	(0.402)	(0.468)	(1.002)	(1.074)
	CNP	5.876	5.656	5.874	5.658	7.317	7.093	7.324	7.113
		(1.04)	(0.754)	(1.039)	(0.758)	(1.046)	(0.765)	(1.059)	(0.773)
	SNP	7.377	7.403	7.351	7.352	8.824	8.778	8.811	8.833
		(2.554)	(2.555)	(2.499)	(2.539)	(2.609)	(2.538)	(2.565)	(2.586)
Chi-Square	EWMA-DS	11.297	11.432	12.25	12.39	13.209	13.337	14.16	14.251
($df = 6$)		(0.529)	(0.622)	(1.316)	(1.382)	(0.51)	(0.604)	(1.3)	(1.363)
	EWMA-IFR	11.492	11.831	12.705	13.061	13.407	13.729	14.608	14.991
		(0.527)	(0.613)	(1.307)	(1.362)	(0.507)	(0.594)	(1.287)	(1.347)
	CNP	12.419	12.132	12.415	12.11	14.333	14.065	14.341	14.076
		(1.43)	(1.057)	(1.43)	(1.05)	(1.382)	(1.018)	(1.387)	(1.031)
	SNP	14.397	14.367	14.371	14.349	16.183	16.159	16.228	16.252
		(3.302)	(3.23)	(3.284)	(3.265)	(3.176)	(3.188)	(3.186)	(3.212)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.20: Mean and standard deviation comparison for $Weibull$ and χ^2 at $\alpha = 0.05$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Weibull	EWMA-DS	7.201	7.553	9.996	10.029	11.317	11.768	14.762	15.065
($\eta = 0.5,\ \beta = 1$)		(1.635)	(1.794)	(4.12)	(4.096)	(1.909)	(2.15)	(4.875)	(4.907)
	EWMA-IFR	7.923	8.921	11.703	12.772	12.151	13.336	16.793	18.162
		(1.781)	(2.089)	(4.505)	(4.648)	(2.072)	(2.374)	(5.341)	(5.418)
	CNP	10.643	8.196	10.585	8.139	15.635	12.729	15.72	12.72
		(3.579)	(1.972)	(3.557)	(1.942)	(4.186)	(2.466)	(4.408)	(2.467)
	SNP	17.336	17.391	17.172	17.014	23.366	23.417	23.531	23.383
		(11.995)	(12.214)	(12.262)	(11.842)	(13.895)	(14.287)	(14.46)	(14.049)
Chi-Square	EWMA-DS	5.178	5.305	6.04	6.138	6.57	6.688	7.425	7.529
($df = 2$)		(0.395)	(0.457)	(1.0)	(1.046)	(0.401)	(0.47)	(1.005)	(1.047)
	EWMA-IFR	5.338	5.616	6.395	6.729	6.73	7.005	7.811	8.116
		(0.397)	(0.464)	(0.998)	(1.049)	(0.402)	(0.476)	(1.009)	(1.048)
	CNP	6.435	5.685	6.422	5.681	7.846	7.095	7.861	7.108
		(1.044)	(0.668)	(1.042)	(0.66)	(1.06)	(0.671)	(1.06)	(0.671)
	SNP	7.95	7.958	7.908	7.932	9.358	9.308	9.284	9.303
		(2.547)	(2.555)	(2.537)	(2.513)	(2.598)	(2.547)	(2.544)	(2.517)
Chi-Square	EWMA-DS	11.426	11.616	12.548	12.703	13.34	13.505	14.469	14.609
($df = 6$)		(0.525)	(0.626)	(1.307)	(1.383)	(0.51)	(0.6)	(1.273)	(1.341)
	EWMA-IFR	11.625	12.007	13.036	13.469	13.547	13.906	14.949	15.34
		(0.516)	(0.618)	(1.303)	(1.368)	(0.51)	(0.595)	(1.264)	(1.323)
	CNP	13.17	12.15	13.157	12.15	15.052	14.076	15.047	14.076
		(1.404)	(0.921)	(1.406)	(0.923)	(1.367)	(0.896)	(1.348)	(0.891)
	SNP	15.098	15.048	15.039	15.061	16.931	16.907	16.924	16.901
		(3.239)	(3.271)	(3.286)	(3.258)	(3.205)	(3.179)	(3.209)	(3.152)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.21: Mean and standard deviation comparison for $Weibull$ and χ^2 at $\alpha = 0.01$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Weibull	EWMA-DS	7.808	8.284	11.448	11.741	12.07	12.625	16.525	16.832
($\eta = 0.5,\ \beta = 1$)		(1.731)	(1.917)	(4.365)	(4.408)	(2.028)	(2.253)	(5.183)	(5.165)
	EWMA-IFR	8.567	9.74	13.564	14.823	12.938	14.318	18.793	20.401
		(1.891)	(2.181)	(4.888)	(5.034)	(2.165)	(2.481)	(5.601)	(5.778)
	CNP	10.246	8.775	10.335	8.853	15.358	13.524	15.36	13.549
		(2.75)	(1.719)	(2.769)	(1.742)	(3.338)	(2.149)	(3.376)	(2.186)
	SNP	20.639	20.631	20.673	20.704	27.408	27.328	27.499	27.759
		(13.182)	(13.35)	(13.062)	(13.214)	(14.893)	(15.098)	(15.141)	(15.187)
Chi-Square	EWMA-DS	5.338	5.521	6.448	6.575	6.734	6.907	7.842	7.989
($df = 2$)		(0.398)	(0.465)	(1.005)	(1.04)	(0.4)	(0.468)	(1.017)	(1.06)
	EWMA-IFR	5.507	5.843	6.864	7.217	6.901	7.233	8.242	8.616
		(0.399)	(0.465)	(0.997)	(1.049)	(0.404)	(0.471)	(1.007)	(1.06)
	CNP	6.366	5.906	6.366	5.901	7.783	7.331	7.788	7.339
		(0.838)	(0.572)	(0.838)	(0.57)	(0.84)	(0.577)	(0.838)	(0.578)
	SNP	8.73	8.736	8.761	8.731	10.147	10.11	10.175	10.153
		(2.559)	(2.514)	(2.518)	(2.53)	(2.542)	(2.513)	(2.559)	(2.563)
Chi-Square	EWMA-DS	11.653	11.879	13.158	13.355	13.585	13.817	15.027	15.197
($df = 6$)		(0.515)	(0.601)	(1.298)	(1.36)	(0.509)	(0.597)	(1.282)	(1.322)
	EWMA-IFR	11.846	12.288	13.662	14.141	13.776	14.212	15.502	15.977
		(0.512)	(0.6)	(1.289)	(1.357)	(0.506)	(0.596)	(1.268)	(1.316)
	CNP	13.06	12.459	13.095	12.473	14.994	14.398	14.988	14.394
		(1.133)	(0.794)	(1.13)	(0.791)	(1.102)	(0.765)	(1.108)	(0.765)
	SNP	16.124	16.073	16.172	16.097	17.969	17.954	17.991	17.938
		(3.252)	(3.161)	(3.207)	(3.173)	(3.133)	(3.136)	(3.191)	(3.123)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.22: Mean and standard deviation comparison for *Loglog* and *Lognormal* at $\alpha = 0.1$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Log Normal	EWMA-DS	1.421	1.429	1.485	1.494	1.55	1.559	1.613	1.619
($\mu = 0,\ \sigma = .25$)		(0.035)	(0.042)	(0.088)	(0.092)	(0.035)	(0.041)	(0.086)	(0.09)
	EWMA-IFR	1.434	1.456	1.515	1.539	1.563	1.585	1.642	1.666
		(0.034)	(0.042)	(0.087)	(0.091)	(0.034)	(0.041)	(0.084)	(0.089)
	CNP	1.497	1.477	1.496	1.478	1.624	1.606	1.625	1.606
		(0.095)	(0.07)	(0.095)	(0.07)	(0.094)	(0.069)	(0.092)	(0.068)
	SNP	1.626	1.625	1.63	1.629	1.747	1.752	1.753	1.749
		(0.217)	(0.218)	(0.222)	(0.22)	(0.216)	(0.215)	(0.215)	(0.212)
Log Normal	EWMA-DS	71843	76995	117859	119915	132877	141645	203425	202925
($\mu = 8,\ \sigma = 2$)		(63865)	(68383)	(159871)	(146854)	(90724)	(101118)	(220637)	(231226)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	84535	72273	84055	72066	159110	139273	158380	139175
		(48632)	(29630)	(48673)	(29126)	(81018)	(50031)	(79512)	(50118)
	SNP	254782	264911	266580	262484	437340	421639	414916	430191
		(479275)	(631612)	(554531)	(565151)	(915579)	(875953)	(748590)	(1298330)
Log Logistic	EWMA-DS	3.855	4.008	5.021	5.032	5.723	5.902	7.251	7.378
($\alpha = 1,\ \beta = 2$)		(1.397)	(1.65)	(3.249)	(3.58)	(16.804)	(13.703)	(6.11)	(10.978)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	4.416	4.086	4.411	4.083	6.35	5.907	6.387	5.926
		(1.354)	(0.86)	(1.354)	(0.875)	(1.895)	(1.237)	(1.929)	(1.266)
	SNP	8.45	8.551	8.219	8.29	12.331	12.095	11.841	12.272
		(11.879)	(26.131)	(18.187)	(12.486)	(26.296)	(23.847)	(18.687)	(37.319)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.23: Mean and standard deviation comparison for *Loglog* and *Lognormal* at $\alpha = 0.05$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Log Normal	EWMA-DS	1.429	1.442	1.509	1.518	1.559	1.57	1.633	1.643
($\mu = 0,\ \sigma = .25$)		(0.035)	(0.042)	(0.087)	(0.094)	(0.033)	(0.04)	(0.086)	(0.091)
	EWMA-IFR	1.443	1.469	1.538	1.566	1.572	1.596	1.664	1.692
		(0.035)	(0.041)	(0.086)	(0.092)	(0.033)	(0.04)	(0.086)	(0.09)
	CNP	1.548	1.48	1.547	1.479	1.673	1.607	1.671	1.607
		(0.094)	(0.061)	(0.093)	(0.061)	(0.09)	(0.059)	(0.09)	(0.059)
	SNP	1.677	1.673	1.67	1.672	1.798	1.791	1.797	1.795
		(0.22)	(0.213)	(0.213)	(0.217)	(0.215)	(0.211)	(0.219)	(0.21)
Log Normal	EWMA-DS	79492	85236	141312	140009	142997	153595	233411	234557
($\mu = 8,\ \sigma = 2$)		(107470)	(100402)	(398401)	(254884)	(98877)	(128231)	(319410)	(280283)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	108529	71835	109129	71954	198190	138440	196530	137586
		(59605)	(25187)	(58971)	(25342)	(94404)	(43384)	(94910)	(42391)
	SNP	320132	311819	315468	316799	517891	507441	511785	512924
		(700486)	(700307)	(713411)	(607602)	(1002136)	(870426)	(1081553)	(1235571)
Log Logistic	EWMA-DS	4.01	4.18	5.529	5.481	5.791	6.026	7.809	7.796
($\alpha = 1,\ \beta = 2$)		(1.604)	(1.609)	(4.377)	(3.494)	(2.145)	(2.282)	(5.082)	(4.781)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	5.051	4.064	5.073	4.076	7.318	5.896	7.333	5.894
		(1.505)	(0.744)	(1.523)	(0.752)	(2.168)	(1.064)	(2.244)	(1.064)
	SNP	9.341	9.419	9.446	9.353	13.755	13.382	13.631	13.485
		(14.748)	(15.0)	(12.115)	(11.849)	(19.441)	(15.839)	(18.624)	(18.508)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.24: Mean and standard deviation comparison for *Loglog* and *Lognormal* at $\alpha = 0.01$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Log Normal	EWMA-DS	1.445	1.462	1.546	1.558	1.574	1.59	1.67	1.683
($\mu = 0,\ \sigma = 0.25$)		(0.034)	(0.041)	(0.086)	(0.092)	(0.034)	(0.04)	(0.084)	(0.088)
	EWMA-IFR	1.459	1.489	1.578	1.611	1.588	1.616	1.701	1.734
		(0.034)	(0.04)	(0.085)	(0.09)	(0.033)	(0.039)	(0.083)	(0.088)
	CNP	1.542	1.5	1.541	1.5	1.669	1.63	1.667	1.628
		(0.074)	(0.052)	(0.075)	(0.053)	(0.074)	(0.052)	(0.073)	(0.05)
	SNP	1.748	1.744	1.744	1.749	1.867	1.864	1.865	1.864
		(0.217)	(0.213)	(0.212)	(0.215)	(0.216)	(0.208)	(0.209)	(0.21)
Log Normal	EWMA-DS	94254	102524	167616	163823	168079	180918	292401	288153
($\mu = 8,\ \sigma = 2$)		(87511)	(95079)	(221740)	(186673)	(166082)	(159923)	(378182)	(285799)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	101035	79166	100970	78927	187985	151300	188119	151727
		(41615)	(23005)	(41933)	(23135)	(70279)	(39870)	(71573)	(39082)
	SNP	416116	427079	411005	415179	687556	652156	679034	666402
		(723498)	(991157)	(807787)	(738361)	(1548554)	(1217323)	(1243182)	(1069816)
Log Logistic	EWMA-DS	4.382	4.631	6.264	6.353	6.245	6.576	8.963	8.937
($\alpha = 1,\ \beta = 2$)		(2.234)	(2.357)	(4.54)	(4.416)	(2.393)	(3.362)	(6.559)	(5.806)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	4.909	4.311	4.913	4.299	7.114	6.245	7.072	6.235
		(1.117)	(0.673)	(1.138)	(0.665)	(1.636)	(0.956)	(1.602)	(0.958)
	SNP	11.76	12.016	11.519	12.068	16.866	18.48	16.79	16.684
		(16.241)	(19.078)	(14.764)	(31.631)	(25.575)	(111.317)	(36.044)	(25.739)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.25: Mean and standard deviation comparison for *Cauchy* and *Loglog* at $\alpha = 0.1$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Cauchy	EWMA-DS	5.862	6.293	11.518	13.072	13.872	14.515	27.724	23.531
($\mu = 0,\ \sigma = 0.5$)		(42.52)	(37.83)	(105.26)	(181.86)	(183.46)	(155.28)	(304.3)	(178.42)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	3.467	2.898	3.423	2.859	7.097	5.94	7.13	5.918
		(2.38)	(1.299)	(2.456)	(1.231)	(4.755)	(2.701)	(4.988)	(2.594)
	SNP	30.133	34.081	28.356	38.992	76.852	88.35	116.382	64.701
		(339.57)	(395.79)	(251.08)	(711.45)	(1370.04)	(1500.96)	(3689.18)	(767.209)
Cauchy	EWMA-DS	11.258	14.021	31.256	29.438	33.84	37.763	69.382	72.536
($\mu = 0,\ \sigma = 1.5$)		(161.4)	(139.05)	(237.4)	(182.01)	(270.1)	(252.9)	(528.74)	(564.32)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	10.337	8.634	10.275	8.661	21.186	17.67	21.703	17.889
		(6.958)	(3.689)	(6.796)	(3.948)	(14.464)	(7.71)	(15.804)	(7.94)
	SNP	109.531	107.57	76.982	90.945	522.534	200.438	180.955	227.391
		(1836.9)	(2030.8)	(551.3)	(1237.8)	(29271.1)	(2465.5)	(1781.6)	(3676.9)
Log Logistic	EWMA-DS	2.563	2.576	2.664	2.674	2.761	2.774	2.869	2.875
($\alpha = 2,\ \beta = 10$)		(0.063)	(0.072)	(0.15)	(0.16)	(0.067)	(0.077)	(0.163)	(0.168)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	2.671	2.639	2.67	2.64	2.882	2.848	2.875	2.844
		(0.149)	(0.107)	(0.146)	(0.107)	(0.158)	(0.114)	(0.156)	(0.114)
	SNP	2.915	2.901	2.901	2.906	3.126	3.128	3.119	3.122
		(0.417)	(0.406)	(0.402)	(0.421)	(0.444)	(0.435)	(0.436)	(0.438)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.26: Mean and standard deviation comparison for *Cauchy* and *Loglog* at $\alpha = 0.05$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Cauchy	EWMA-DS	15.364	16.804	15.03	17.378	20.764	21.222	63.546	46.212
($\mu = 0,\ \sigma = 0.5$)		(691.04)	(590.37)	(140.08)	(204.32)	(532.01)	(434.47)	(2672.6)	(1335.3)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	4.563	2.859	4.569	2.829	9.299	5.813	9.321	5.842
		(3.194)	(1.081)	(3.422)	(1.061)	(6.351)	(2.151)	(6.292)	(2.187)
	SNP	93.895	74.786	39.496	43.01	184.36	90.356	80.447	113.68
		(4901.3)	(2056.9)	(400.86)	(652.15)	(10175.8)	(1571.3)	(945.68)	(3185.2)
Cauchy	EWMA-DS	29.888	31.243	52.791	43.266	42.667	46.351	118.50	94.099
($\mu = 0,\ \sigma = 1.5$)		(572.44)	(476.25)	(674.47)	(361.69)	(321.94)	(279.25)	(1667.4)	(899.44)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	13.795	8.571	13.918	8.564	27.951	17.532	28.286	17.524
		(9.349)	(3.253)	(9.494)	(3.17)	(19.042)	(6.48)	(20.021)	(6.668)
	SNP	605.472	828.871	183.26	102.61	637.847	238.356	226.155	265.482
		(46909.2)	(65664.8)	(4120.2)	(824.41)	(38124.2)	(2624.5)	(2237.9)	(6036.6)
Log Logistic	EWMA-DS	2.575	2.595	2.705	2.716	2.775	2.793	2.908	2.92
($\alpha = 2,\ \beta = 10$)		(0.062)	(0.072)	(0.155)	(0.159)	(0.067)	(0.077)	(0.166)	(0.173)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	2.749	2.641	2.75	2.641	2.958	2.843	2.958	2.846
		(0.15)	(0.094)	(0.153)	(0.093)	(0.162)	(0.1)	(0.16)	(0.099)
	SNP	2.989	2.985	2.986	2.993	3.213	3.217	3.214	3.216
		(0.425)	(0.423)	(0.415)	(0.42)	(0.452)	(0.454)	(0.458)	(0.459)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.27: Mean and standard deviation comparison for *Cauchy* and *Loglog* at $\alpha = 0.01$

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Cauchy	EWMA-DS	10.184	12.684	22.609	49.546	25.009	26.995	77.304	66.648
($\mu = 0,\ \sigma = 0.5$)		(129.27)	(167.6)	(236.97)	(3062.4)	(479.6)	(416.6)	(1818.3)	(1515.2)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	4.149	3.144	4.165	3.145	8.598	6.53	8.557	6.5
		(1.989)	(0.982)	(2.038)	(0.984)	(4.196)	(2.077)	(4.151)	(2.041)
	SNP	106.23	207.74	60.352	64.078	135.73	237.84	197.77	154.08
		(2802.9)	(12281)	(702.94)	(762.45)	(1649.3)	(11986)	(5381.6)	(4160.3)
Cauchy	EWMA-DS	29.777	41.037	72.664	89.186	83.763	91.642	149.87	136.4
($\mu = 0,\ \sigma = 1.5$)		(242.36)	(974.39)	(715.02)	(2933.4)	(1593.2)	(1416.7)	(2779.2)	(1680.5)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	12.491	9.446	12.528	9.466	25.715	19.403	25.959	19.432
		(5.985)	(2.936)	(6.156)	(2.975)	(12.439)	(6.012)	(13.58)	(6.09)
	SNP	179.28	288.49	273.31	223.16	989.84	649.08	1007.6	318.01
		(2493.6)	(7571.9)	(5446.4)	(3448.0)	(45100)	(22584)	(62022)	(2966.6)
Log Logistic	EWMA-DS	2.602	2.628	2.765	2.783	2.802	2.827	2.972	2.992
($\alpha = 2,\ \beta = 10$)		(0.063)	(0.073)	(0.162)	(0.166)	(0.069)	(0.078)	(0.17)	(0.176)
	EWMA-IFR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	CNP	2.741	2.674	2.738	2.674	2.947	2.88	2.948	2.879
		(0.12)	(0.081)	(0.12)	(0.082)	(0.128)	(0.085)	(0.128)	(0.087)
	SNP	3.115	3.117	3.121	3.114	3.346	3.353	3.349	3.354
		(0.442)	(0.435)	(0.437)	(0.436)	(0.466)	(0.466)	(0.469)	(0.473)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

Table A.28: Mean and standard deviation comparison for exponential distribution.

		$q = 90\%$				$q = 95\%$			
		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.1$		$\lambda = 0.3$	
Distribution	Procedure	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$	$t = 3$	$t = 5$
Exp	EWMA-DS	2.543	2.585	2.894	2.933	3.232	3.28	3.58	3.624
$(\beta = 1)$		(0.193)	(0.229)	(0.486)	(0.509)	(0.199)	(0.234)	(0.498)	(0.523)
$\alpha = 0.1$	EWMA-IFR	2.617	2.742	3.071	3.207	3.313	3.44	3.768	3.914
		(0.197)	(0.233)	(0.498)	(0.515)	(0.199)	(0.235)	(0.501)	(0.527)
	CNP	2.938	2.831	2.942	2.831	3.651	3.545	3.652	3.545
		(0.517)	(0.375)	(0.522)	(0.381)	(0.531)	(0.388)	(0.528)	(0.386)
	SNP	3.672	3.676	3.671	3.679	4.413	4.393	4.392	4.407
		(1.25)	(1.245)	(1.262)	(1.248)	(1.271)	(1.282)	(1.275)	(1.29)
Exp	EWMA-DS	2.586	2.653	3.021	3.061	3.285	3.343	3.701	3.763
$(\beta = 1)$		(0.199)	(0.232)	(0.502)	(0.517)	(0.201)	(0.234)	(0.496)	(0.526)
$\alpha = 0.05$	EWMA-IFR	2.664	2.807	3.2	3.354	3.365	3.502	3.897	4.058
		(0.199)	(0.234)	(0.498)	(0.518)	(0.203)	(0.237)	(0.5)	(0.529)
	CNP	3.202	2.839	3.212	2.837	3.937	3.554	3.915	3.551
		(0.519)	(0.332)	(0.523)	(0.332)	(0.538)	(0.341)	(0.528)	(0.341)
	SNP	3.948	3.993	3.947	3.963	4.682	4.648	4.645	4.672
		(1.267)	(1.3)	(1.261)	(1.274)	(1.298)	(1.246)	(1.267)	(1.261)
Exp	EWMA-DS	2.675	2.764	3.225	3.281	3.372	3.46	3.915	3.995
$(\beta = 1)$		(0.198)	(0.233)	(0.507)	(0.525)	(0.199)	(0.236)	(0.498)	(0.53)
$\alpha = 0.01$	EWMA-IFR	2.76	2.928	3.431	3.613	3.454	3.621	4.119	4.312
		(0.199)	(0.234)	(0.507)	(0.527)	(0.2)	(0.236)	(0.501)	(0.532)
	CNP	3.186	2.956	3.184	2.955	3.891	3.665	3.889	3.667
		(0.416)	(0.291)	(0.421)	(0.292)	(0.425)	(0.29)	(0.422)	(0.294)
	SNP	4.386	4.419	4.37	4.373	5.078	5.076	5.071	5.073
		(1.289)	(1.288)	(1.284)	(1.287)	(1.274)	(1.267)	(1.285)	(1.281)

EWMA-DS: Distribution specific EWMA-based method.
EWMA-IFR: EWMA-based method for distributions with IFR property (Lee and Sa, 2001).
SNP: Standard Non-parametric method (Conover, 1998).
CNP: Continuous non-parametric method (Sa and Razaila, 2004).

APPENDIX B

CODE

B.1 Sample sizes calculation for upper-side distribution specific tolerance limits

```
import math
import scipy.integrate as integrate
from scipy.special import erfinv
import scipy.stats as stats

def getw(t, lam): # using t and lambda values
    w = [(1 - lam) ** t] # create W_0
    for i in range(1, t + 1): # from t and lambda create the rest of the W's
        w.append(lam * (1 - lam) ** (t - i))
    return w

def getc(setofws, xqval):
    c = (1 - setofws[0]) * xqval
    return c

AlphaValues = [0.1, 0.05, 0.02, 0.01]
qValues = [0.15, 0.1, 0.05, 0.02]
LambdaValues = [0.1, 0.2, 0.3]
Tvalues = [1, 2, 3, 4, 5]
options = {'epsabs': 0.0005, 'epsrel': 0.0005}

# Edit boundries here.
def t3bd2(x3): return [0, (c - x3 * w[3]) / w[2]]
def t3bd3(): return [0, c / w[3]]
def t4bd2(x3, x4): return [0, (c - x4 * w[4] - x3 * w[3]) / w[2]]
def t4bd3(x4): return [0, (c - x4 * w[4]) / w[3]]
def t4bd4(): return [0, c / w[4]]
def t5bd2(x3, x4, x5): return [0, (c - x5 * w[5] - x4 * w[4] - x3 * w[3]) / w[2]]
def t5bd3(x4, x5): return [0, (c - x5 * w[5] - x4 * w[4]) / w[3]]
def t5bd4(x5): return [0, (c - x5 * w[5]) / w[4]]
def t5bd5(): return [0, c / w[5]]

# Edit quantiles here.
xqs = []
for i in range(len(qValues)):
    mu = 0
    sig = 0.25
    xqs.append(math.exp(mu + math.sqrt(2 * (sig**2))*erfinv(2*(1 - qValues[i]) - 1)))
print(xqs)
```

```python
# Edit pdf here.
def pdf(x, n):
    mu = 0
    sig = 0.25
    return n * (((1 / 2) + (1 / 2)*math.erf((math.log(x)-mu)/(math.sqrt(2) * sig)))**(n-1)) * ((1 / (x
        ↪ * sig * math.sqrt(2 * math.pi))) * math.exp(-((math.log(x)-mu)**2)/(2 * (sig**2))))

# Edit cdf here.
def cdf(x, n):
    mu = 0
    sig = 0.25
    return ((1 / 2) + (1 / 2)*math.erf((math.log(x)-mu)/(math.sqrt(2) * sig)))**n

# Code Start.
f2 = lambda x2: pdf(x2, n2) * cdf((c - w[2]*x2)/w[1], n1)
f23 = lambda x2, x3: pdf(x2, n2) * pdf(x3, n3) * cdf(((c - x3 * w[3] - x2 * w[2]) / w[1]), n1)
f234 = lambda x2, x3, x4: pdf(x2, n2) * pdf(x3, n3) * pdf(x4, n4) * cdf(((c - x4 * w[4] - x3 * w[3] - x2
    ↪ * w[2]) / w[1]), n1)
f2345 = lambda x2, x3, x4, x5: pdf(x2, n2) * pdf(x3, n3) * pdf(x4, n4) * pdf(x5, n5) * cdf(((c - x5 * w
    ↪ [5] - x4 * w[4] - x3 * w[3] - x2 * w[2]) / w[1]), n1)

# t = 1
t = Tvalues[0]
Nt1 = []
for k in range(len(AlphaValues)):
    alpha = AlphaValues[k]
    n1 = 1
    for j in range(len(xqs)):
        xq = xqs[j]
        for i in range(len(LambdaValues)):
            lam = LambdaValues[i]
            w = getw(t, lam)
            c = getc(w, xq)
            soln = 1 - cdf(c / w[1], n1)
            while soln < 1-alpha:
                n1 += 1
                soln = 1 - cdf(c / w[1], n1)
            print(n1)
            Nt1.append(n1)

# t = 2
t = Tvalues[1]
Nt2 = []
for k in range(len(AlphaValues)):
    alpha = AlphaValues[k]
    n2 = 1
    for j in range(len(xqs)):
        xq = xqs[j]
        for i in range(len(LambdaValues)):
            n1 = Nt1[k * 12 + j * 3 + i]
            n2 = max(int(n1 / 3), n2)
            lam = LambdaValues[i]
            w = getw(t, lam)
            c = getc(w, xq)
```

```
            integral = integrate.nquad(f2, [[0, c / w[2]]], opts=options)
            soln = 1 − integral[0]
            print(integral)
            while soln < 1 − alpha:
                n2 += 1
                integral = integrate.nquad(f2, [[0, c / w[2]]], opts=options)
                soln = 1 − integral[0]
                print(integral)
            print(n2)
            Nt2.append(n2)

# t = 3
t = Tvalues[2]
Nt3 = []
for k in range(len(AlphaValues)):
    alpha = AlphaValues[k]
    n3 = 1
    for j in range(len(xqs)):
        xq = xqs[j]
        for i in range(len(LambdaValues)):
            lam = LambdaValues[i]
            w = getw(t, lam)
            n1 = Nt1[k * 12 + j * 3 + i]
            n2 = Nt2[k * 12 + j * 3 + i]
            n3 = max(int(n2 / 2), n3)
            c = getc(w, xq)
            integral = integrate.nquad(f23, [t3bd2, t3bd3], opts=options)
            soln = 1 − integral[0]
            print(integral)
            while soln < (1−alpha):
                n3 += 1
                integral = integrate.nquad(f23, [t3bd2, t3bd3], opts=options)
                soln = 1 − integral[0]
                print(integral)
            print(n3)
            Nt3.append(n3)

# t = 4
t = Tvalues[3]
Nt4 = []
for k in range(len(AlphaValues)):
    alpha = AlphaValues[k]
    n4 = 1
    for j in range(len(xqs)):
        xq = xqs[j]
        for i in range(len(LambdaValues)):
            lam = LambdaValues[i]
            w = getw(t, lam)
            n1 = Nt1[k * 12 + j * 3 + i]
            n2 = Nt2[k * 12 + j * 3 + i]
            n3 = Nt3[k * 12 + j * 3 + i]
            n4 = max(2*n3 − n2 − 2, n4 − 1, 1)
            c = getc(w, xq)
            integral = integrate.nquad(f234, [t4bd2, t4bd3, t4bd4], opts=options)
```

```
            soln = 1 - integral[0]
            print(integral)
            while soln < (1-alpha):
                n4 += 1
                integral = integrate.nquad(f234, [t4bd2, t4bd3, t4bd4], opts=options)
                soln = 1 - integral[0]
                print(integral)
            print(n4)
            Nt4.append(n4)

# t = 5
t = Tvalues[4]
Nt5 = []
for k in range(len(AlphaValues)):
    alpha = AlphaValues[k]
    print(alpha)
    n5 = 1
    for j in range(len(xqs)):
        xq = xqs[j]
        print(j)
        for i in range(len(LambdaValues)): # set i = zero to get the first lambda value :::: for i in range
            ↪ (len(LambdaValues)):
            lam = LambdaValues[i]
            print(lam)
            w = getw(t, lam)
            n1 = Nt1[k * 12 + j * 3 + i]
            n2 = Nt2[k * 12 + j * 3 + i]
            n3 = Nt3[k * 12 + j * 3 + i]
            n4 = Nt4[k * 12 + j * 3 + i]
            n5 = max(2*n4 - n3 - 1, n5 - 1, 1)
            c = getc(w, xq)
            integral = integrate.nquad(f2345, [t5bd2, t5bd3, t5bd4, t5bd5], opts=options)
            soln = 1 - integral[0]
            print(integral)
            while integral[0] < 0.005:
                print(n5, "too_small_continue")
                n5 += 1
                pizza = integral[0]
                integral = integrate.nquad(f2345, [t5bd2, t5bd3, t5bd4, t5bd5], opts=options)
                soln = 1 - integral[0]
                print(integral)
                if integral[0] < pizza:
                    soln = 0.9 - alpha
                    integral = (0.5, 1)
            while soln < (1-alpha):
                n5 += 1
                integral = integrate.nquad(f2345, [t5bd2, t5bd3, t5bd4, t5bd5], opts=options)
                soln = 1 - integral[0]
                print(integral)
            print(n5)
            Nt5.append(n5)

print(Nt1)
print(Nt2)
```

```
print(Nt3)
print(Nt4)
print(Nt5)
```

B.2 Simulation study - Intervals

```
import math
import scipy.stats as sc
import random
from scipy.special import erfinv
import statistics as stat

# define distribution functions to match sample sizes code.
def weibull_sg(k, lam, n): # parameters: k, lam || n = samp.size
    sampleset = []
    for i in range(n):
        u = random.uniform(0, 1)
        sampleset.append(math.exp(math.log(lam) + (1 / k) * math.log(-math.log(u))))
    return sampleset

def weibull_icdf(k, lam, qval): # parameters: k, lam || n = samp.size
    return math.exp(math.log(lam) + (1 / k) * math.log(-math.log(qval)))

def invgaussian_sg(mu, theta, n): # parameters: mu, lam( or theta) || n = samp.size
    sampleset = []
    for i in range(n):
        v = random.normalvariate(0, 1)
        y = v * v
        x = mu + (mu*mu*y/(2*theta)) - (mu/(2*theta))*math.sqrt(4*mu*theta*y + mu*mu*y*y)
        z = random.uniform(0, 1)
        if z <= mu / (mu + x):
            sol = x
        else:
            sol = mu*mu/x
        sampleset.append(sol)
    return sampleset

def invgauss_icdf(mu, theta, qval):
    if mu == 1 and theta == 5:
        if qval == 0.15: return 1.42960
        if qval == 0.1: return 1.58836
        if qval == 0.05: return 1.85279
        if qval == 0.02: return 2.19511
    if mu == 5 and theta == 1:
        if qval == 0.15: return 7.94159
        if qval == 0.1: return 11.9578
        if qval == 0.05: return 21.3689
        if qval == 0.02: return 38.6246

def chisquare_sg(df, n): # parameters: df = degrees of freedom || n = samp.size
    sampleset = []
    for i in range(n):
```

```
        u = random.uniform(0, 1)
        sampleset.append(sc.chi2.ppf(u, df))
    return sampleset

def chisquare_icdf(df, qval):
    return sc.chi2.ppf(1 − qval, df)

def lognormal_sg(mu, sigma, n): # parameters: mu, sigma || n = samp.size
    sampleset = []
    for i in range(n):
        u = random.lognormvariate(mu, sigma)
        sampleset.append(u)
    return sampleset

def lognormal_icdf(mu, sig, qval):
    return math.exp(mu + math.sqrt(2 ∗ (sig∗∗2))∗erfinv(2∗(1 − qval) − 1))

def loglogistic_sg(alpha, beta, n): # parameters: alpha, beta || n = samp.size
    sampleset = []
    for i in range(n):
        u = random.uniform(0, 1)
        sol = alpha ∗ ((u / (1 − u))∗∗(1 / beta))
        sampleset.append(sol)
    return sampleset

def loglogistic_icdf(alpha, beta, qval):
    return alpha ∗ (((1 − qval) / qval)∗∗(1 / beta))

def exponential_sg(beta, n): # parameters: beta || n = samp.size
    sampleset = []
    for i in range(n):
        u = random.expovariate(beta)
        sampleset.append(u)
    return sampleset

def exp_icdf(beta, qval):
    return −math.log(qval)/beta

def cauchy_sg(x_0, gamma, n): # parameters: df = degrees of freedom || n = samp.size
    sampleset = []
    for i in range(n):
        u = random.uniform(0, 1)
        sampleset.append(sc.cauchy.ppf(u, x_0, gamma))
    return sampleset

def cauchy_icdf(x_0, gamma, q):
    return sc.cauchy.ppf(1−q, x_0, gamma)

def icdf_dist(dist_ind, q):
    if dist_ind == 1: return invgauss_icdf(1, 5, q)
    if dist_ind == 2: return invgauss_icdf(5, 1, q)
    if dist_ind == 3: return weibull_icdf(1.5, 1, q)
    if dist_ind == 4: return weibull_icdf(0.5, 1, q)
    if dist_ind == 5: return chisquare_icdf(2, q)
```

```
    if dist_ind == 6: return chisquare_icdf(6, q)
    if dist_ind == 7: return lognormal_icdf(0, 0.25, q)
    if dist_ind == 8: return lognormal_icdf(8, 2, q)
    if dist_ind == 9: return loglogistic_icdf(1, 2, q)
    if dist_ind == 10: return loglogistic_icdf(2, 10, q)
    if dist_ind == 11: return exp_icdf(1, q)
    if dist_ind == 12: return cauchy_icdf(0, 0.5, q)
    if dist_ind == 13: return cauchy_icdf(0, 1.5, q)
    return print("Indicator not available")

def sample_dist(dist_ind, ss):
    if dist_ind == 1: return invgaussian_sg(1, 5, ss)
    if dist_ind == 2: return invgaussian_sg(5, 1, ss)
    if dist_ind == 3: return weibull_sg(1.5, 1, ss)
    if dist_ind == 4: return weibull_sg(0.5, 1, ss)
    if dist_ind == 5: return chisquare_sg(2, ss)
    if dist_ind == 6: return chisquare_sg(6, ss)
    if dist_ind == 7: return lognormal_sg(0, 0.25, ss)
    if dist_ind == 8: return lognormal_sg(8, 2, ss)
    if dist_ind == 9: return loglogistic_sg(1, 2, ss)
    if dist_ind == 10: return loglogistic_sg(2, 10, ss)
    if dist_ind == 11: return exponential_sg(1, ss)
    if dist_ind == 12: return cauchy_sg(0, 0.5, ss)
    if dist_ind == 13: return cauchy_sg(0, 1.5, ss)
    return print("Indicator not available")

# sample sizes
sa_lee_ewma_sets = [...]
ewma_sets = [...]
conover_n = [...]

def conover_utl(sampset, alpha, q):
    conover = [...]
    n = 0
    for i in range(len(AlphaValues)):
        for j in range(len(qValues)):
            if AlphaValues[i] == alpha and qValues[j] == q:
                n = conover[i * 12 + j * 3]
    if n > len(sampset):
        return print("sample size is too small")
    holdset = random.sample(sampset, n)
    return [holdset, max(holdset)]

def ewma_utl(set_of_sets, alpha, q, lam, dist_ind, t):
    ewma_dist = ewma_sets[dist_ind - 1]
    n = []
    for num in range(t):
        ewma_set = ewma_dist[num]
        for k in range(len(AlphaValues)):
            for j in range(len(qValues)):
                for i in range(len(LambdaValues)):
                    if AlphaValues[k] == alpha and qValues[j] == q and LambdaValues[i] == lam:
                        n.append(ewma_set[k * 12 + j * 3 + i])
    holdsamp = []
```

```
    maxsamples = []
    for i in range(len(n)):
        holdsamp.append(random.sample(set_of_sets[i], n[i]))
    for i in range(len(n)):
        maxsamples.append(max(holdsamp[i]))
    ewma_0 = icdf_dist(dist_ind, q)
    ewma_1 = lam * maxsamples[0] + (1 - lam) * ewma_0
    ewma_2 = lam * maxsamples[1] + (1 - lam) * ewma_1
    ewma_3 = lam * maxsamples[2] + (1 - lam) * ewma_2
    ewma_4 = lam * maxsamples[3] + (1 - lam) * ewma_3
    ewma_5 = lam * maxsamples[4] + (1 - lam) * ewma_4
    return [ewma_1, ewma_2, ewma_3, ewma_4, ewma_5]

def sa_lee_ewma_utl(set_of_sets, alpha, q, lam, dist_ind, t):
    n = []
    diff_qvalues = [0.1, 0.05, 0.01]
    for num in range(t):
        ewma_set = sa_lee_ewma_sets[num]
        for k in range(len(AlphaValues)):
            for j in range(len(diff_qvalues)):
                for i in range(len(LambdaValues)):
                    if AlphaValues[k] == alpha and diff_qvalues[j] == q and LambdaValues[i] == lam:
                        n.append(ewma_set[k * 9 + j * 3 + i])
    holdsamp, maxsamples = []
    for i in range(len(n)): holdsamp.append(random.sample(set_of_sets[i], n[i]))
    for i in range(len(n)): maxsamples.append(max(holdsamp[i]))
    ewma_0 = icdf_dist(dist_ind, q)
    ewma_1 = lam * maxsamples[0] + (1 - lam) * ewma_0
    ewma_2 = lam * maxsamples[1] + (1 - lam) * ewma_1
    ewma_3 = lam * maxsamples[2] + (1 - lam) * ewma_2
    ewma_4 = lam * maxsamples[3] + (1 - lam) * ewma_3
    ewma_5 = lam * maxsamples[4] + (1 - lam) * ewma_4
    return [holdsamp, [ewma_1, ewma_2, ewma_3, ewma_4, ewma_5]]

def sa_utl(set_of_sets, alpha, q, t): # [0.1, 0.05, 0.02, 0.01]
    for i in range(len(AlphaValues)):
        if AlphaValues[i] == alpha: choo = i
    for i in range(len(qValues)):
        if qValues[i] == q: chichi = i
    sa_big_set = [...]
    hold_sa_1 = sa_big_set[choo]
    hold_sa_2 = hold_sa_1[chichi]
    t_samp_sizes = []
    for k in range(t): t_samp_sizes.append(hold_sa_2[k])

    sa_utl_hold, s_o_s_, setofsethold = []
    for w in range(len(set_of_sets)):
        setofsethold = set_of_sets[w] + setofsethold
        setofsethold.sort()
        s_o_s_.append(setofsethold)
    sa_utl_hold.append(max(s_o_s_[0]))
    for j in range(1, len(set_of_sets)):
        sosh = s_o_s_[j]
        sa_utl_hold.append(sosh[len(sosh) - t_samp_sizes[j]])
```

```
    return [t_samp_sizes, sa_utl_hold]

dist = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
# dist number value indicators
# 1 = Inverse Gaussian: mu = 1, theta = 5 -
# 2 = Inverse Gaussian: mu = 5, theta = 1 -
# 3 = weibull: k = 1.5, lambda = 1 -
# 4 = weibull: k = 0.5, lambda = 1 -
# 5 = Chi Squared: df = 2 -
# 6 = Chi Squared: df = 6 -
# 7 = Log Normal: mu = 0 sig = 0.25 -
# 8 = Log Normal: mu = 8 sig = 2 -
# 9 = Log Logistic: alpha = 1 beta = 2 -
# 10 = Log Logistic: alpha = 2 beta = 10 -
# 11 = Exponential: b = 1 -
# 12 = cauchy location 0 gamma 0.5 -
# 13 = cauchy location 0 gamma 1.5 -

AlphaValues = [0.1, 0.05, 0.02, 0.01]
qValues = [0.15, 0.1, 0.05, 0.02]
LambdaValues = [0.1, 0.2, 0.3]
Tvalues = [1, 2, 3, 4, 5]
simulationsize = 10000
distribution = 11
qnum = 2 # 0 = 0.15, 1 = 0.1, 2 = 0.5, 3 = 0.02
alphanum = 3 # 0 = 0.1, 1 = 0.05, 2 = 0.02, 3 = 0.01
lambdanum = 2 # 0 = 0.1, 1 = 0.2, 2 = 0.3
samplesize = conover_n[alphanum * 12 + qnum * 3]
print(samplesize)
conover_len_t2 = []
ewma_len_t2 = []
sa_lee_len_t2 = []
sa_raz_len_t2 = []
conover_len_t4 = []
ewma_len_t4 = []
sa_lee_len_t4 = []
sa_raz_len_t4 = []

for zzz in range(simulationsize):
    conover_samples = []
    conover_utls = []
    for c in range(max(Tvalues)):
        hold = conover_utl(sample_dist(distribution, samplesize), AlphaValues[alphanum], qValues[qnum])
        conover_samples.append(hold[0])
        conover_utls.append(hold[1])
    sa_lee_ewma_utls_hold = sa_lee_ewma_utl(conover_samples, AlphaValues[alphanum], qValues[qnum],
        ↪ LambdaValues[lambdanum], distribution, max(Tvalues))
    ewma_utls = ewma_utl(sa_lee_ewma_utls_hold[0], AlphaValues[alphanum], qValues[qnum],
        ↪ LambdaValues[lambdanum], distribution, max(Tvalues))
    sa_utls = sa_utl(conover_samples, AlphaValues[alphanum], qValues[qnum], max(Tvalues))
    sa_raz_utl = sa_utls[1]
    sa_lee_ewma_utls = sa_lee_ewma_utls_hold[1]
    conover_len_t2.append(conover_utls[2])
    ewma_len_t2.append(ewma_utls[2])
```

```
    sa_lee_len_t2.append(sa_lee_ewma_utls[2])
    sa_raz_len_t2.append(sa_raz_utl[2])
    conover_len_t4.append(conover_utls[4])
    ewma_len_t4.append(ewma_utls[4])
    sa_lee_len_t4.append(sa_lee_ewma_utls[4])
    sa_raz_len_t4.append(sa_raz_utl[4])
    print(zzz)
# repeat 10,000 times.
print("For alpha value ", AlphaValues[alphanum])
print("For q value ", qValues[qnum], "The x-q for the distribution is", icdf_dist(distribution, qValues[
    ↪ qnum]))
print("For lambda value ", LambdaValues[lambdanum])
print("For distribution indicator ", distribution)

ewma_mean_t2 = stat.mean(ewma_len_t2)
ewma_sd_t2 = stat.stdev(ewma_len_t2, ewma_mean_t2)
ewma_mean_t4 = stat.mean(ewma_len_t4)
ewma_sd_t4 = stat.stdev(ewma_len_t4, ewma_mean_t4)
print(" my method MEAN ", round(ewma_mean_t2, 3), " && ", round(ewma_mean_t4, 3), " ")
print(" my method SD (", round(ewma_sd_t2, 3), ") && (", round(ewma_sd_t4, 3), ") ")

sa_lee_mean_t2 = stat.mean(sa_lee_len_t2)
sa_lee_sd_t2 = stat.stdev(sa_lee_len_t2, sa_lee_mean_t2)
sa_lee_mean_t4 = stat.mean(sa_lee_len_t4)
sa_lee_sd_t4 = stat.stdev(sa_lee_len_t4, sa_lee_mean_t4)
print(" Sa & Lee method MEAN ", round(sa_lee_mean_t2, 3), " && ", round(sa_lee_mean_t4, 3), " ")
print(" Sa & Lee method SD (", round(sa_lee_sd_t2, 3), ") && (", round(sa_lee_sd_t4, 3), ") ")
# from these sets, calculate the mean and SD from the values.

sa_raz_mean_t2 = stat.mean(sa_raz_len_t2)
sa_raz_sd_t2 = stat.stdev(sa_raz_len_t2, sa_raz_mean_t2)
sa_raz_mean_t4 = stat.mean(sa_raz_len_t4)
sa_raz_sd_t4 = stat.stdev(sa_raz_len_t4, sa_raz_mean_t4)
print(" Sa & Raz method MEAN ", round(sa_raz_mean_t2, 3), " && ", round(sa_raz_mean_t4, 3), " ")
print(" Sa & Raz method SD (", round(sa_raz_sd_t2, 3), ") && (", round(sa_raz_sd_t4, 3), ") ")

conover_mean_t2 = stat.mean(conover_len_t2)
conover_sd_t2 = stat.stdev(conover_len_t2, conover_mean_t2)
conover_mean_t4 = stat.mean(conover_len_t4)
conover_sd_t4 = stat.stdev(conover_len_t4, conover_mean_t4)
print(" Conover method MEAN ", round(conover_mean_t2, 3), " && ", round(conover_mean_t4, 3), " 
    ↪ ")
print(" Conover method SD (", round(conover_sd_t2, 3), ") && (", round(conover_sd_t4, 3), ") ")
```

B.3 Simulation study - Sample Sizes

```
# lee sa, ewma ifr
x = [[17, 17, 17, 34, 35, 35],[15, 16, 16, 31, 32, 33],[14, 14, 15, 28, 30, 31]]
# ewma_sets =
# Inverse Gaussian: mu = 1, theta = 5 -
x1 = [[11, 12, 13, 23, 24, 25], [10, 10, 11, 20, 21, 23], [9, 9, 11, 18, 19, 22]]
# weibull: k = 1.5, lambda = 1 -
x2 = [[12, 12, 13, 24, 25, 26], [ 10, 11, 11, 20, 22, 23], [ 9, 10, 11, 19, 20, 22]]
# Chi Squared: df = 2 -
x3=[[11, 12, 12, 23, 24, 25], [10, 10, 11, 19, 21, 22], [9, 10, 11, 18, 19, 21]]
# Exponential: b = 1 -
x4=[[11, 12, 12, 23, 24, 25], [10, 10, 11, 19, 21, 22], [9, 10, 11, 17, 19, 21]]
# Cauchy location 0 gamma 0.5 -
x5=[[8, 9, 9, 17, 17, 18] , [7, 7, 8, 13, 14, 16] , [5, 6, 7, 11, 12, 14]]
# 8 = Log Normal: mu = 8 sig = 2 -
x6 =[[ 8, 9, 9, 18, 18, 19], [ 7, 7, 8, 14, 16, 17], [ 6, 6, 8, 12, 13, 15]]
# 9 = Log Logistic: alpha = 1 beta = 2 -
x7 = [[ 10, 10, 11, 19, 20, 21], [ 8, 9, 9, 16, 17, 19], [ 7, 8, 8, 14, 15, 17]]
# 2 = Inverse Gaussian: mu = 5, theta = 1 -
x8 = [[9, 10, 10, 20, 21, 22], [8, 8, 9, 16, 17, 19], [7, 7, 8, 14, 16, 18]]
# 4 = weibull: k = 0.5, lambda = 1 -
x9 =[[10, 10, 11, 20, 21, 22], [8, 9, 9, 17, 18, 20], [7, 7, 8, 14, 17, 18]]
# 6 = Chi Squared: df = 6 -
x10 =[[12, 12, 13, 24, 24, 26], [10, 11, 11, 20, 22, 23], [9, 9, 11, 18, 20, 21]]
# 7 = Log Normal: mu = 0 sig = 0.25 -
x11 =[[12, 12, 13, 24, 25, 26], [10, 11, 11, 20, 21, 23], [9, 10, 11, 18, 20, 22]]
# 10 = Log Logistic: alpha = 2 beta = 10 -
x12 =[[11, 12, 12, 23, 23, 25], [10, 10, 11, 19, 21, 22], [8, 10, 10, 17, 19, 20]]
# 13 = cauchy location 0 gamma 1.5 -
x13 =[[8, 9, 9, 17, 17, 18], [7, 7, 8, 13, 14, 16], [5, 6, 7, 11, 12, 14]]

def solver(sets, sets2): # sets = my stuff, sets2 = lee sa
    setone=list()
    for i in range(len(sets)):
        settwo = list()
        hold = sets[i]
        hold2 = sets2[i]
        for j in range(len(hold)):
            settwo.append(round(100*((hold2[j] - hold[j])/hold2[j]), 1))
        setone.append(settwo)
    return setone

z = [x1,x2,x3,x4,x5, x6, x7, x8, x9, x10, x11, x12, x13]
for k in range(len(z)):
    print(k+1, solver(z[k], x))
```

B.4 Illustration

```
import math
import numpy as np
import random
import matplotlib.pyplot as plt
from scipy import stats

def inverse_cdf_weibull(k_, lam_, n_):
    sampleset, uset = [], []
    for i in range(n_):
        u = random.uniform(0, 1)
        uset.append(1 - u)
        sampleset.append(math.exp(math.log(lam_) + (1 / k_)
                * math.log(-math.log(u))))
    return [sampleset, uset]

def delta_fun(x):
    if x >= 0:
        y = 1
    else: y = 0
    return y

def weib(k__, lam__, x):
    return (k__ / lam__) * (x / lam__) ** (k__ - 1)
        * np.exp(-(x / lam__) ** k__)

def weib_cdf(x, _lam, _k):
    return 1 - math.exp(-(x / _lam)**_k)

def getxq(_k, _lam, _q):
    return math.exp(math.log(_lam) + (1 / _k)
        * math.log(-math.log(1 - _q)))

def mean(list_):
    return sum(list_) / len(list_)

def round_to_decimal(places, _list_):
    return [round(numb, places) for numb in _list_]

def weibull_mean(__lam, __k):
    return __lam * math.gamma(1 + (1 / __k))

def weibull_var(__lam, __k):
    return (__lam**2) * (math.gamma(1 + (2 / __k)) -
        math.gamma(1 + (1 / __k))**2)

# Edit distribution here.
k = 2.378 # shape
lam = 10.78 # scale
ewma_q = 0.95
ewma_lamda = 0.2
ewma_alpha = 0.05
n_week = 5 # sample size per week
```

```python
quarters = 5
num_weeks = 13*quarters # number of weeks to survey
quarter = int(num_weeks/13) # number of yearly quarters test is being done
weeks_ = [...] # Data goes here

# Separating the data into quarterly segments ( by 13 weeks )
r_value = []
weeks_mean = []
for i in range(len(weeks_)):
    weeks_by_week = weeks_[i]
    r_value.append(weeks_by_week[n_week - 1] - weeks_by_week[0])
    weeks_mean.append(mean(weeks_[i]))
for num in range(len(weeks_)):
    print("week", num + 1, ": average ", "%.4f" % weeks_mean[num], ", list: ", round_to_decimal(4,
        weeks_[num]))

data_sets_total = []
data_by_quarter = []
for i in range(quarter):
    data_sets_hold = []
    for z in range(13*i, 13*(i+1)):
        hold = weeks_[z]
        for v in range(len(hold)):
            data_sets_hold.append(hold[v])
            data_sets_total.append(hold[v])
    data_by_quarter.append(data_sets_hold)

# Estimating the Parameters for the Q-Q plot.
data = data_sets_total
data.sort()
a_out, Kappa_out, loc_out, Lambda_out = stats.exponweib.fit(data, f0=1, floc=0)
print("K = ", Kappa_out, " and Lambda =", Lambda_out)
# Formulating a q-q plot for the data.
data_sets_total.sort() # This q-q plot data from MLE parameter estimates.
y_axis_qq = []
y_axis_quantiles = []
for b in range(1, len(data_sets_total)+1):
    p = (b - 0.5) / len(data_sets_total)
    y_axis_quantiles.append(p)
    y_axis_qq.append(Lambda_out * (-math.log(1-p))**(1 / Kappa_out))

plt.scatter(data_sets_total, y_axis_quantiles, s=7) # Scatterplot of the Data.
plt.ylabel('F(x)')
plt.xlabel('Sample Values')
plt.show()
plt.plot(data, stats.exponweib.pdf(data, * stats.exponweib.fit(data, 1, 1, scale=0, loc=0)))
_ = plt.hist(data, bins=np.linspace(0, data[len(data)-1]+1, 33), density=True, alpha=0.5)
plt.annotate("Shape: $k = %.3f$ \n Scale: $\lambda = %.3f$" % (Kappa_out, Lambda_out), xy=(15,
    .08))
plt.ylabel('Probability Density')
plt.xlabel('Sample Values')
plt.show() # Plot of the fitted data in a histogram.
plt.scatter(y_axis_qq, data_sets_total, s=7.5) # Q-Q plot of the data.
plt.plot(y_axis_qq, y_axis_qq, 'red')
```

```
plt.ylabel('Data Quantiles')
plt.xlabel('Theoretical Quantiles')
plt.title('Q-Q plot')
plt.show()

# Kolmogorov-Smirnov Test.
ks_test = []
for c in range(len(data_sets_total)):
    testval = weib_cdf(data_sets_total[c], Lambda_out, Kappa_out)
    test1 = testval - (c+1)/len(data_sets_total)
    test2 = testval - ((c+1) - 1/len(data_sets_total))
    if test1 >= test2:
        ks_test.append(test1)
    else: ks_test.append(test2)
ks_test.sort()
print("The Kolmogorov-Smirnov Test results in a D value of ", ks_test[len(ks_test) - 1])
print("The Kolmogorov-Smirnov Test has a critical value, D_a, for alpha = 0.01 for a sample of size"
    ↪ , len(data_sets_total), "of approximately ", 1.63/math.sqrt(len(data_sets_total)))

x_bar = weeks_mean
x_bar_bar = mean(weeks_mean)
r_bar = mean(r_value)
p_x = 0
for i in range(num_weeks):
    week_i = weeks_[i]
    for j in range(n_week):
        p_x += delta_fun(x_bar_bar - week_i[j])
px = p_x / (len(data_sets_total))
print(px)

w_u = 0.61
w_l = 0.56
v_u = 2.25
v_l = 0
x_ucl = x_bar_bar + r_bar*w_u
x_lcl = x_bar_bar - r_bar*w_l
r_ucl = r_bar*v_u
r_lcl = r_bar*v_l
samplenumbers = [59, 30, 27, 24, 24]
max_samples = []
for i in range(len(data_by_quarter)):
    randsample = random.sample(data_by_quarter[i], samplenumbers[i])
    randsample.sort()
    max_samples.append(randsample[len(randsample)-1])
ewma_0 = getxq(Kappa_out, Lambda_out, ewma_q)
ewma_1 = ewma_lamda * max_samples[0] + (1 - ewma_lamda) * ewma_0
ewma_2 = ewma_lamda * max_samples[1] + (1 - ewma_lamda) * ewma_1
ewma_3 = ewma_lamda * max_samples[2] + (1 - ewma_lamda) * ewma_2
ewma_4 = ewma_lamda * max_samples[3] + (1 - ewma_lamda) * ewma_3
ewma_5 = ewma_lamda * max_samples[4] + (1 - ewma_lamda) * ewma_4
ewma_ucl = [ewma_0, ewma_1, ewma_2, ewma_3, ewma_4, ewma_5]
quarters_x_axis = [0, 1 * 13, 2 * 13, 3 * 13, 4 * 13, 5 * 13]
print("The ewma UCL's are ", ewma_ucl, "for t = [0, 1, 2, 3, 4, 5] respectively.")
```

```
# x-bar chart
plt.figure(figsize=(9, 5))
x_bar_list = list(range(1, len(x_bar)+1))
plt.axhline(y=x_ucl, linestyle='dashed', label='UCL', color='red')
plt.axhline(y=x_bar_bar, linestyle='solid', label='CL', color='red')
plt.axhline(y=x_lcl, linestyle='dashed', label='LCL', color='red')
plt.plot(x_bar_list, x_bar, marker='.', label='x-bar', color='black')
plt.scatter(quarters_x_axis, ewma_ucl, label='EWMA', marker='d', color='black')
plt.xlabel('Weeks')
plt.ylabel('x-bar')
plt.ylim(0, 25)
ax = plt.subplot(111)
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width*0.8, box.height])
plt.legend(loc='center_left', bbox_to_anchor=(1.01, 0.5))
plt.show()

# r chart
plt.figure(figsize=(9, 5))
r_value_list = list(range(1, len(r_value)+1))
plt.axhline(y=r_ucl, linestyle='dashed', label='UCL', color='red')
plt.axhline(y=r_bar, linestyle='solid', label='CL', color='red')
plt.axhline(y=r_lcl, linestyle='dashed', label='LCL', color='red')
plt.plot(r_value_list, r_value, marker='.', label='R', color='black')
plt.xlabel('Weeks')
plt.ylabel('Range')
ax = plt.subplot(111)
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width*0.8, box.height])
plt.legend(loc='center_left', bbox_to_anchor=(1.01, 0.5))
plt.show()
```

REFERENCES

Amin, R. and Lee, S. (1999). The effects of autocorrelation and outliers on two-sided tolerance limits. *Journal of Quality Technology*, 31:286–300.

Armstrong, W. (1982). Tables for normal tolerance limits, sampling plans, and screening. *Journal of the Royal Statistical Society Series C*, 31(3):306–307.

Chakravarti, I., Laha, R., and Roy, J. (1967). *Handbook of Methods of Applied Statistics*. Number v. 1 in Handbook of Methods of Applied Statistics. Wiley.

Conover, W. J. (1998). *Practical nonparametric statistics*, volume 350. John Wiley & Sons.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del R'ıo, J. F., Wiebe, M., Peterson, P., G'erard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Hunter, J. S. (1986). The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–210.

Lee, S. and Sa, P. (2001). Tolerance limits as a continuous assessment tool. *Total Quality Management*, 12:491–510.

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30.

Piessens, R., De Doncker-Kapenga, E., de, E., Ueberhuber, C., and Kahaner, D. (1983). *Quadpack: A Subroutine Package for Automatic Integration*. Computational Mathematics Series. Springer-Verlag.

Razaila, L. and Sa, P. (2003). The continuous non-parametric tolerance limits. *COMMUNICATIONS IN STATISTICS Simulation and Computation® Vol. 32*, No. 1:191–204.

Sa, P. and Razaila, L. (2004). One-sided continuous tolerance limits and their accompanying sample size problem. *Journal of Applied Statistics*, 31:419–434.

Tadikamalla, P. R. (1980). A look at the burr and related distributions. *International Statistical Review / Revue Internationale de Statistique*, 48(3):337–344.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.