University of Alberta

Sequential Analysis with Applications to Clinical Trials

by

Evgenia Samuylova

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Statistics

Department of Mathematical and Statistical Sciences

Spring 2012
Edmonton, Alberta

To my family

Abstract

The topic of the thesis is an overview of some sequential and change-point detection methods with applications to clinical trials. Performing sequential monitoring is important for ethical, economical and other reasons. It is important to terminate a study as soon as possible when potentially harmful treatments are used or when financial resources are limited. The modern theory of sequential testing of hypotheses started with works of Wald and Barnard on quality control of military supplies during World War II. Since then sequential methods received a lot of attention. In this thesis we consider application of truncated sequential methods to four different models. First, we consider sequential testing of composite hypotheses in the presence of nuisance parameters. Second, we describe sequential procedures for binary data with risk-adjustment. Then, we consider non-parametric methods for sequential monitoring of longitudinal data. We finish the thesis with an example of monitoring proportions in the context of waiting time at emergency departments in hospitals.

Acknowledgments

I would like to express my gratitude to my supervisor Professor Edit Gombay for her patience and kind support throughout my master's program.

I wish to thank Jason Scarlett, a health system simulation engineer at Data Integration, Measurement & Reporting at Alberta Health Services, for compiling and providing the data set on emergency departments lengths of stay.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sequential Analysis Motivation

This thesis describes sequential analysis methodology. We illustrate the methods in the context of clinical trials. However, applications can be found in almost any area where an experiment is carried out sequentially. This includes quality control and reliability in industry and sequential monitoring of optimal portfolio weights in finance. Another area of applications is monitoring hospital performance, for example, with respect to infection rates or patient falls. One of the recent important applications is monitoring of wait times including emergency department lengths of stay, wait times for hip replacement and radiation oncology services. Long wait times at emergency departments and walk-in clinics is a problem that almost everyone encounters every year. The problem was resolved in Ontario, but is still present in Alberta. Statistical monitoring of wait times at emergency departments in Alberta is a part of the Five-Year Health Action Plan [1].

In experiments, where data accumulate over time, it is natural to monitor results as they occur. An analysis of the data at one or more time points prior to the official close of the study is called interim analysis. The purpose of such analysis is to assess the possibility of terminating the study early. There are several reasons for the interim analyses. The most important is the ethical reason. For example, in clinical trials, it is necessary to monitor the results to ensure that individuals are not exposed to unsafe treatments. If a new treat-

ment is beneficial or harmful compared to the placebo or a current standard treatment, it is an ethical need to perform interim analysis and to terminate the study earlier than planned if sufficient evidence is gained. However, originally sequential methods were developed for economic reasons. These methods typically lead to savings in sample size, time and cost in comparison to the fixed sample procedures.

The randomized clinical trials have become a gold standard for medical research. It has led to a great development of methodology for the design, conduct and analysis of these studies. The process of interim monitoring of the accumulating data in such trials are receiving more and more attention. It is even a formal requirement for some studies. For example, the US Food and Drug Administration (FDA) guidelines contain "E9 Statistical Principles for Clinical Trials" [2] that says "Careful conduct of a clinical trial according to the protocol has a major impact on the credibility of the results (see ICH E6). Careful monitoring can ensure that difficulties are noticed early and their occurrence or recurrence minimized... The goal of such an interim analysis is to stop the trial early if the superiority of the treatment under study is clearly established, if the demonstration of a relevant treatment difference has become unlikely or if unacceptable adverse effects are apparent". The ICH guidelines were adopted by Health Canada in 2003 [3].

Often the interim monitoring is performed by an independent data monitoring committee (DMC), sometimes called data and safety monitoring board (DSMB). The committee is formed to assess the progress of a clinical trial, patient safety and treatment efficacy, and recommend whether to continue, modify or terminate a trial. The operation of data monitoring committees is described in [4]. The book provides instructions on how to establish DMC, describes its purpose and responsibilities as well as statistical approaches to monitoring accumulating data.

To stress the importance of the interim analysis we would like to mention that there are many conferences held on this topic, and corresponding training

is offered to researches in clinical trials trough workshops, such as the Workshop on Current Advances in Interim Analysis and Design Modifications held in Rockville, USA in September 2000. Also, the interim analysis is one of the topics at FDA/Industry Statistics Workshop every year.

Although the formal requirements mentioned above were written with the accent on the more practical group sequential methods, continuous monitoring is often desirable. In this thesis we focus on sequential methods, the methods where analysis is performed after each new observation.

History of Sequential Methods

Even though interim analysis is common in many fields of science, it did not receive much attention in the classical theory and the sequential procedures cannot be found in most of the widely used statistical software packages. Armitage [5] said: "The classical theory of experimental design deals predominantly with experiments of predetermined size, presumably because the pioneers of the subject, particularly R. A. Fisher, worked in agricultural research, where the outcome of a field trial is not available until long after the experiment has been designed and started. It is interesting to speculate how differently statistical theory might have evolved if Fisher had been employed in medical or industrial research."

The formal application of sequential procedures started in the late 1920s in the area of statistical quality control in industry. Walter A. Shewhart [6] introduced control charts for process control while working for Bell Labs, formerly known as the American Telephone & Telegraph Company (AT&T). The control chart is nowadays one of the seven basic tools of Statistical Process Control (SPC) and the most technically sophisticated, out of the seven. By the 1920s the AT&T's engineers had realized the importance of reducing variation in a manufacturing process in order to keep the frequency of failures and repairs low. Shewhart formulated the problem in terms of common- and special-causes of variation and introduced the control chart as a tool for dis-

tinguishing between the two. Common-cause is the usual, chance cause of variation in a system. A certain amount of natural variability always exists, no matter how well a production process is designed or maintained. This natural variability, "background noise", was named the common-cause of variation. Special-causes are unusual, not previously observed causes of variation. The special-cause variability might occur, for example, in case of improperly adjusted or controlled machines, operator errors or defective raw materials. Shewhart emphasized that bringing a production process into a state of statistical control, where there is only common-cause variation, and keeping it in control, is necessary to predict future output and to manage a process economically.

The Shewhart control chart is a graphical display of a quality characteristic (e.g. mean) that has been measured on a sample versus the sample number. A typical control chart is shown in Fig. 1.1. The control chart consists of a

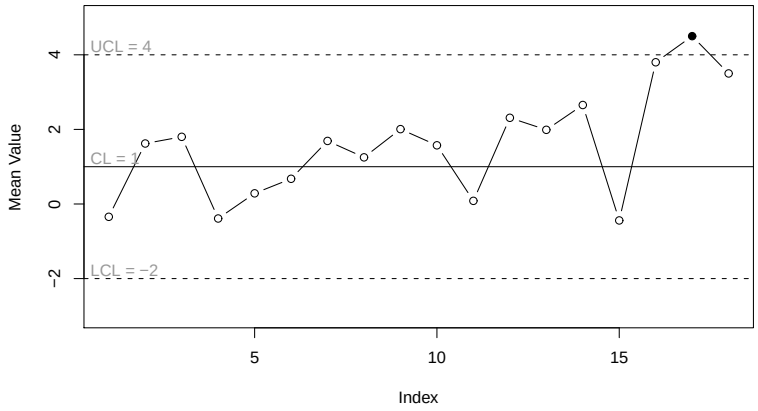

Figure 1.1: Shewhart Control Chart.

center line that represents the mean value of the quality characteristic when the process is in-control state and two other horizontal lines, called the upper control limit (UCL) and the lower control limit (LCL). The UCL and LCL lines are typically drawn 3 standard errors from the center line. The chart may optionally have upper and lower warning limits, drawn as separate lines,

typically 2 standard errors from the central line. The warning and control limits are also sometimes referred to as alert and alarm thresholds, respectively.

If the process is in control, almost all, 99.73% for the three-sigma limits, points will plot within control limits. A point that falls outside of the control limits indicates that the process is out of control, and investigation and correction are required.

Although the control chart was not motivated by the Neyman-Pearson lemma, it is closely connected to the hypothesis testing. In a sense, the control chart is a test of the hypothesis that the process is in a state of statistical control. A point plotting within the control limits is equivalent to failing to reject the null hypothesis of statistical control, while a point outside the control limits is equivalent to rejecting the null hypothesis. The main difference with the hypothesis testing is that when testing hypotheses the validity of assumptions is usually checked, whereas control charts are used to detect departures from the assumed in control state. In general, one should not worry too much about assumptions such as the form of the distribution or independence when applying control charts with the purpose of reducing variability about the state of statistical control. The hypothesis testing framework is useful in analyzing the performance of control charts. It brings in the probability of type I error, concluding the process is out of control when it is really in control, and the probability of type II error, concluding the process is in control when it is really out of control.

A general model for a Shewhart control chart is as follows. Let S be a sample statistic that measures some quality characteristic of interest, the mean of S is θ_0 and the standard deviation of S is σ_S. Then the center line, the upper control limit and the lower control limit are defined by the equations

$$\text{UCL} = \theta_0 + L\sigma_S,$$

$$\text{center line} = \theta_0,$$

$$\text{UCL} = \theta_0 - \text{L}\sigma_\text{S},$$

where L is the distance of the control limits from the center line. If the process is in control with the mean θ_0 and S is the sample average, the by using the central limit theorem to assume that S is approximately normally distributed, it is expected that $100(1-\alpha)\%$ of the sample means S will fall between $\theta_0 + Z_{\alpha/2}\sigma_S$ and $\theta_0 - Z_{\alpha/2}\sigma_S$, where $L = Z_{\alpha/2}$ is the $\alpha/2$-th quantile of a standard normal distribution.

The design of the Shewhart control chart is given by the sample size, control limits (L) and frequency of sampling. The control limits are chosen to minimize the risk of type I and type II errors. Moving control limits away from the center line decreases the risk of type I error while increasing the risk of type II error. If the control lines are moved closer to the center line the effect on errors is opposite. It is important to note here that we are talking about individual type I and type II errors, i.e. corresponding to testing $H_0 : \theta_i = \theta_0$ for sample number i. Keeping type I error small prevents unnecessary process adjustments and is consistent with the "if it isn't broken don't fix it" philosophy. For the control chart where average as a sample statistic and three-sigma control limits are used, from the standard normal table the probability of type one error is 0.0027 under the assumption of normality. That is, a false alarm will be generated on average in 0.27 out of 100 samples.

In addition to control limits, sample size and the frequency of sampling must be specified to design a control chart. Ideally, large samples very frequently should be taken. In this situation it is easier to detect small shifts very early. However this is usually not economically feasible. A compromise between small samples at short intervals and larger samples at longer intervals is usually sought.

Another way to make the decisions about sample size is through the average running length (ARL). ARL is the average number of points that must be plotted before a point indicates that a process if out of control. This is an important characteristic of sequential tests, especially open ended. Shewhart procedure stops and declares an out-of-control condition with probability one.

Thus it is not possible to control the traditional overall type I error at a certain level. Instead, the sample size and/or L are chosen to control ARL as large as possible when the process is in control, since stopping represents a false alarm, while if the process has shifted an undesirable setting, we would like the procedure to stop shortly. ARL is also called average sample number (ASN) and average stopping time.

When the observations are uncorrelated the ARL for any Shewhart control chart is

$$\text{ARL} = \frac{1}{p},$$

where p is the individual type I error, the probability that any point exceeds the control limits. ARL is used to evaluate the performance of sequential procedures. For the control chart with the average as a sample statistic and three-sigma control limits, $p = 0.0027$ when the process is in control and therefore in control ARL=370. That is, even if the process remains in control, an out-of-control signal will be generated every 370 samples on average. The use of average run lengths for comparing control chart performance is usually criticized, because that average usually follows a geometric distribution, which has high variability and is very skewed. The standard deviation of the geometric distribution is $\sqrt{1-p}/p$, which is approximately 370 for the three-sigma limits. Because the distribution is quite skewed to the right, it might be more appropriate to report percentiles of the run length distribution instead of just the ARL.

There is a number of generalizations of Shewhart control charts to accommodate for correlation between the observations and different types of data (e.g. binary, counts). Shewhart control charts are still widely used in industry.

The modern theory of sequential testing of hypotheses was initially developed independently by Wald in 1943 [7] in the US and Barnard [8] in the UK. Wald and Barnard developed the sequential methods for quality control of military supplies in the World War II. In 1947 Wald [9] proposed very powerful and well-known sequential probability ratio test (SPRT), where for a sequence of

independent and identically distributed random variables $Y_1, \ldots, Y_k, \ldots$ with a density function $f(\cdot;\theta)$ with respect to a σ-finite measure, where $\theta \in \Omega \subset \mathbb{R}$ is the parameter of interest, the null hypothesis

$$H_0 : \theta = \theta_0$$

is tested versus

$$H_A : \theta = \theta_A. \tag{1.1}$$

The value θ_0 is typically determined by the current process performance, while θ_A represents an alternative value of interest, corresponding typically to worse performance. The test is defined as follows: Two positive constants A and B ($B < A$) are chosen so that the test has the prescribed error probabilities $\alpha = P(\text{type I error})$ and $\beta = P(\text{type II error})$. At each stage of the experiment, the likelihood ratio $\prod_{i=1}^{k} f(y_i;\theta_0) / \prod_{i=1}^{k} f(y_i;\theta_A)$ is computed. If

$$\prod_{i=1}^{k} f(y_i;\theta_0) / \prod_{i=1}^{k} f(y_i;\theta_A) \leq B$$

the sampling is stopped with the acceptance of H_0. If

$$\prod_{i=1}^{k} f(y_i;\theta_0) / \prod_{i=1}^{k} f(y_i;\theta_A) \geq A$$

the sampling is stopped with the acceptance of H_A. If

$$B < \prod_{i=1}^{k} f(y_i;\theta_0) / \prod_{i=1}^{k} f(y_i;\theta_A) < A$$

the sampling is continued by taking an additional observation. The sample size at which the boundary A or B is crossed is a random variable and the mean of this random variable is the average running length (ARL). It is also called the average sample number (ASN) and we use the ASN notation in the rest of the thesis. The Wald's procedure is optimal in the sense of minimizing the average sample number among all the tests with a finite ASN and error probabilities α and β. Note that the theory can essentially be used only in case both H_0 and H_A are simple hypotheses.

One of the other approaches to quality control problems is change-point detection. In statistics these problems can usually be modeled as follows. We have a sequence of independent and identically distributed random variables $Y_1, \ldots, Y_k, \ldots$ and would like to detect whether a change at time τ occurred in the sequence and that after time τ Y's have another distribution. In contrast with the sequential testing described above, in change-point detection we test

$$\begin{array}{ll} H_0: & \theta_i = \theta_0 \text{ for all } i \geq 1 \text{ versus} \\ H_A: & \theta_i = \theta_0 \text{ for all } i < \tau, \\ & \theta_i = \theta_A \text{ for all } i \geq \tau, \end{array} \tag{1.2}$$

where τ is unknown. Wald's SPRT was originally developed to for testing the hypotheses when τ is fixed and equals 1.

Sequential procedures that test against an alternative hypothesis that all observations $Y_1, \ldots, Y_k, \ldots$ come from $f(\cdot; \theta_A)$ cannot be expected to be very powerful if the change in distribution happens late in the sample. The few observations after the change in parameter would be dominated by the many from the original distribution. In the case when both $f(y; \theta_0)$ and $f(y; \theta_A)$ are completely specified, that is θ_0 and θ_A are both known constants, there are some popular and well developed change-point detection procedures available, such as Shewhart's control charts, Moving Average control charts, Page's CUSUM procedure and V-mask (see [10]). The basic change-point detection problem was first considered by Page in 1954 [11]. If the interest is in the mean parameter, Page's CUSUM test monitors the partial sums

$$S_k - \min_{1 \leq j \leq k} S_j, \quad k = 1, 2, \ldots,$$

where $S_k = \sum_{i=1}^{k} w_i$ and w_i is the sample score assigned to the i -th observation. There are different implementations of this procedure. Two-sided implementation suggested by Barnard [12] involved the use of a graphical device, called a V-mask. This approach is inconvenient to use in practice. In situations where concern is focused on only detecting increases (or decreases) of the parameter, the tabular form of the CUSUM is easier to use. That is, if the alternative hypothesis is one-sided, $H_A: \theta_i = \theta_A > \theta_0$ for all $i \geq \tau$, a

standard tabular CUSUM involves monitoring:

$$S_k = \max(0, S_{k-1} + w_k), \quad k = 1, 2, 3, \ldots,$$

where $S_0 = 0$, and w_k is the sample weight (see [10]). The null hypothesis of no change is rejected at the first k when $S_k \geq h$.

The design of the CUSUM is given by w_k and h. Lorden (1971) [13] suggested the log-likelihood ratio weights

$$w_k = \log \frac{f(Y_k; \theta_A)}{f(Y_k; \theta_0)}. \tag{1.3}$$

Moustakides [14] showed that this choice of weights is optimal in the sense that, among all schemes with the same ASN under H_0, the log-likelihood ratio weights give the smallest ASN under H_A given that the change point has been past.

The choice of a control limit h is based on the Average Sample Number (ASN) of the CUSUM under H_0 and H_A. Similar to Shewhart control chart, it is not possible to control the traditional type I error at a certain level. Instead, h is chosen to control ASN as large as possible under H_0 and as small as possible under H_A. After the control limit h and weights w_k are selected the CUSUM algorithm can be formulated as follows.

CUSUM Test. Stop and conclude that H_0 is not supported by the data at the first k, when

$$S_k \geq h.$$

The other method, that is extensively used and performs similar to CUSUM, is the exponentially weighted moving average (EWMA). It is another alternative to the Shewhart control chart when detecting small shifts is of interest. Similar to the CUSUM, the EWMA is typically used with individual observations, rather than samples.

The exponentially weighted moving average control chart was originally proposed by Roberts (1959) [15]. The theoretical properties of the EWMA

were studied later (Crowder [16],Lucas and Saccucci [17]). The EWMA for monitoring the process mean is defined as

$$S_k = \lambda Y_k + (1-\lambda)\, S_{k-1}, \tag{1.4}$$

where $0 < \lambda \leq 1$ is a constant, sometimes called a smoothing parameter, and $S_0 = \theta_0$. Sometimes the average of the preliminary data or the historical value is used for S_0. Rewriting S_{k-1} in terms of previous Y_i's in (1.4), demonstrates that the EWMA test statistic S_k equals an exponentially weighted average of all previous observations

$$S_k = \lambda \sum_{i=0}^{k-1} (1-\lambda)^i Y_{k-i} + (1-\lambda)^k S_0.$$

The weights $\lambda(1-\lambda)^i$ decrease geometrically so that the most recent observations have heavier weights while the first observations contribute very little. In contrast, the original CUSUM control chart uses equal weights. The EWMA is widely used in time series modeling and in forecasting (Box et al. [18])).

Assuming Y_i's are independent random variables with variance σ^2, the variance of S_k is

$$\sigma^2_{S_k} = \sigma^2 \left(\frac{\lambda}{2-\lambda} \right) \left[1 - (1-\lambda)^{2k} \right].$$

Therefore the center line and control limits for the EWMA control chart are defined by the equations

$$\begin{aligned} \text{UCL} &= \theta_0 + L\sigma \sqrt{\tfrac{\lambda}{2-\lambda}\left[1-(1-\lambda)^{2k}\right]}, \\ \text{center line} &= \theta_0, \\ \text{UCL} &= \theta_0 - L\sigma \sqrt{\tfrac{\lambda}{2-\lambda}\left[1-(1-\lambda)^{2k}\right]}, \end{aligned} \tag{1.5}$$

where L is the width of the control limits [10]. The process is considered out-of-control whenever the EWMA test statistic is out of the control limits. Note that the term $\left[1-(1-\lambda)^{2k}\right]$ in the above equations tends to 1 as $k \to \infty$. The EWMA chart is identical to a Shewhart control chart when $\lambda = 1$. Therefore, given the width of the control limits L and the smoothing parameter λ the EWMA algorithm can be formulated as follows.

EWMA. Stop and conclude that $H_0 : \theta = \theta_0$ is not supported by the data at the first k, when

$$|S_k - \theta_0| \geq L\sigma\sqrt{\frac{\lambda}{2-\lambda}\left[1-(1-\lambda)^{2k}\right]}. \tag{1.6}$$

The design of the EWMA is given by L and λ. Similar to the CUSUM the choice of these values depends on the desired ASN. Values of λ between 0.05 and 0.25 were found to work well in practice. It was also found that $L = 3$, the usual three-sigma limits, works reasonably well, especially with the larger values of λ ($\lambda > 0.1$).

Although the normal distribution is the basis of the EWMA chart, the chart is relatively robust to the normality assumption violations. The chart, however, can be adapted to some other distributions, like Poisson and Bernoulli, and to monitoring parameters other than the mean [10].

The EWMA chart is sensitive to small shifts in the process mean, but is very slow in detecting large shifts. The ASN performance is equivalent to that of CUSUM.

Truncated Sequential Tests

The stopping rules for the above procedures, e.g. Wald's sequential probability ratio test or the CUSUM, do not rely on asymptotic distributions. These open-ended procedures stop and reject the null hypothesis of the process being in control with probability one. The disadvantage of this is that all these procedures do not allow to control type I error, which may lead to many false alarms, especially if the change in the process happens not soon after monitoring begins. The hypotheses that are tested, (1.1) or (1.2), can also be generalized by not fully specifying the alternative parameter θ_A. We now describe three truncated sequential procedures that allow to control type I

error and test the generalized hypotheses

$$\begin{aligned} H_0: &\quad \theta_i = \theta_0 \text{ for all } i \geq 1 \text{ versus} \\ H_A: &\quad \theta_i = \theta_0 \text{ for all } i < \tau, \\ &\quad \theta_i = \theta_A > \theta_0 \text{ for all } i \geq \tau, \end{aligned}$$

where θ_A and the change-point τ are unknown. The term truncated means that there is a maximum sample size n that is specified before the monitoring begins and testing is performed after each new observation until the maximum sample size is reached.

These sequential tests are based on the efficient score vector, that is defined as

$$V_k(\theta) = \sum_{i=1}^{k} \frac{\partial}{\partial \theta} \log f(Y_i; \theta).$$

To obtain sequential test statistics we standardize $V_k(\theta_0)$ with $I^{-1/2}(\theta_0)$, where $I(\theta)$ is the information matrix

$$I(\theta) = -E\left[\frac{\partial^2}{\partial \theta^2} \log f(Y; \theta)\right].$$

Let $S_k(\theta) = I^{-1/2}(\theta) V_k(\theta)$ and n be the truncation point, i.e. the maximal allowed sample size. The truncated level α tests are as follows.

Test 1.1. Stop and conclude that H_0 is not supported by the data at the first k, $1 < k \leq n$ when

$$\frac{1}{\sqrt{k}} S_k(\theta_0) \geq C_1(\alpha, n),$$

where

$$C_1(\alpha, n) = \frac{-\log\left(-\log\left(1-\alpha\right)\right) + 2\log\log n + \frac{1}{2}\log\log\log n - \frac{1}{2}\log\pi}{\sqrt{2\log\log n}};$$

otherwise do not reject H_0.

For the two-sided version of Test 1.1 monitor $k^{-\frac{1}{2}}|S_k(\theta_0)|$ and reject H_0 if it is greater than

$$C_1^*(\alpha, n) = \frac{-\log\left(-\frac{1}{2}\log\left(1-\alpha\right)\right) + 2\log\log n + \frac{1}{2}\log\log\log n - \frac{1}{2}\log\pi}{\sqrt{2\log\log n}}. \tag{1.7}$$

Asymptotic distribution of the test statistic in Test 1.1 was calculated by Darling and Erdős [19]. However, critical values $C_1(\alpha, n)$ and $C_1^*(\alpha, n)$, corresponding to the asymptotic distribution, give conservative tests. They can be improved by using Vostrikova's results [20]. See Appendix A for details.

Test 1.2. Stop and conclude that H_0 is not supported by the data at the first k, $1 < k \leq n$ when

$$\frac{1}{\sqrt{n}} S_k(\theta_0) \geq C_2(\alpha),$$

where C_2 is obtained from the distribution of $|N(0,1)|$, $N(0,1)$ a standard normal random variable; otherwise do not reject H_0.

For the two-sided version of Test 1.2 monitor $n^{-1/2}|S_k(\theta_0)|$ and reject H_0 if it is greater than $C_2^*(\alpha)$, where $C_2^*(\alpha)$ satisfies

$$\frac{4}{\pi} \sum_{m=0}^{\infty} \frac{(-1)^m}{2m+1} \exp\left(-\frac{\pi^2(2m+1)^2}{8C_2^*(\alpha)^2}\right) = 1 - \alpha. \tag{1.8}$$

Test 1.1 and Test 1.2 were compared in Gombay [21] when $\tau = 1$, i.e. in sequential testing. Test 1.1 is conservative and stops early for large changes. However Test 1.2 is more powerful at the expense of a greater average stopping time.

Tests 1.1 and 1.2 were originally developed for the case when $\tau = 1$ in H_A. When the problem of interest is the change point detection a sequential test motivated by cumulative sum procedure may be considered.

Test 1.3. Stop and conclude that H_0 is not supported by the data at the first k, $1 < k \leq n$ when

$$\max_{0 \leq j < k} \frac{1}{\sqrt{n}} \left(S_k(\theta_0) - S_j(\theta_0)\right) \geq C_2^*(\alpha);$$

otherwise do not reject H_0.

The two-sided version of Test 1.2 has the same critical value as the one-sided Test 1.3. Test 1.3 has more power when a change in a distribution

happens late. The distributions of test statistics in Test 1.2 and Test 1.3 under H_0 are provided in Appendix A.

The tests described in the Introduction can be applied to many different models. We illustrate the application of the sequential and change detection tests to some of the models. In Chapter 2 we consider sequential methods for testing composite hypotheses, in particular in the presence of nuisance parameters. In Chapter 3 we present tests for binary data and a risk-adjustment procedure and illustrate them with a surgeon performance monitoring example. In Chapter 4 we consider nonparametric sequential methods for longitudinal clinical trials in terms of comparison of several treatment groups. In Chapter 5 we provide an example of monitoring proportions in the context of waiting times at emergency departments at the Stollery Hospital and the University of Alberta Hospital.

Chapter 2

Composite Hypotheses

Let $Y_1, \ldots, Y_i, \ldots$ be independent random variables with density $f(y; \xi_i)$ with respect to a σ-finite measure, where $\xi_i = (\theta_i, \eta_i)$ denotes the parameter vector. We assume that $\theta_i \in \Omega_1 \subset \mathbb{R}^d$, $d \geq 1$, $\eta_i \in \Omega_2 \subset \mathbb{R}^p$, $p \geq 0$ and consequently $\xi_i \in \Omega = \Omega_1 \times \Omega_2$. In this notations θ is the parameter of interest and η is the nuisance parameter. In this section we consider sequential tests of the composite hypotheses

$$\begin{array}{ll} H_0: & \theta_i = \theta_0,\ \eta_i = \eta \in \Omega_2 \text{ for all } i \geq 1 \text{ versus} \\ H_A: & \theta_i = \theta_0,\ \eta_i = \eta \in \Omega_2 \text{ for all } i < \tau, \\ & \theta_i = \theta_A,\ \eta_i = \eta \in \Omega_2 \text{ for all } i \geq \tau, \end{array} \tag{2.1}$$

where $\eta,\ \theta_A,\ \tau \geq 1$ are unknown. These hypotheses are composite for two reasons: first, the alternative parameter space for θ is not a single point as θ_A is unknown, second, there is a nuisance parameter η.

Wald was the first who tried to generalize his open-ended likelihood-ratio test by introducing weight functions. He suggested integrating it over the range of nuisance parameters to reduce a composite hypothesis to a simple one in terms of the parameter of interest. This approach was found not successful in applications.

Another approach suggested using Wald-type procedure and asymptotics developed by Barlett (1946) [22] and Cox (1963) [23] under contiguous alternatives, that is, assuming that $\|\theta_A - \theta_0\| = O(N^{-1/2})$, where N is the sample size, which is a random variable. It was shown that the likelihood ratio has optimal power in fixed sample size tests (see Bahadur [24], Brown [25]). In

Gombay (2002) [26] truncated sequential tests based on the likelihood ratio were compared to the sequential t-test of Barnard [27] and Rushton [28], [29] with the variance as a nuisance parameter. The procedure based on the likelihood ratio was found to have greater power at the expense of greater average sample number (ASN). See [21] and [26] for more discussions.

Tests 1.1-1.3 formulated in the previous section do not require specifying θ_A. In this section we give extensions of these tests to testing hypotheses in the presence of nuisance parameters.

The efficient score vector is defined as

$$V_k(\xi) = \sum_{i=1}^{k} \nabla_\xi \log f(Y_i; \xi)$$

and the information matrix is

$$I(\xi) = -E\left[\frac{\partial^2}{\partial\xi_i \partial\xi_j} \log f(Y; \xi)\right].$$

The information matrix I can be partitioned with respect to θ and η as

$$I = \begin{pmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{pmatrix},$$

where I_{11} is $d \times d$, $I_{12} = I_{21}^T$ is $d \times p$ and I_{22} is $p \times p$. The inverse of I is then

$$I^{-1} = \begin{pmatrix} I^{11} & I^{12} \\ I^{21} & I^{22} \end{pmatrix}.$$

The parameter of interest is θ, so the nuisance parameter η is replaced by its restricted maximum likelihood estimator $\hat{\eta}_k$, i.e. the solution of

$$\sum_{i=1}^{k} \nabla_\eta \log f(Y_i; \theta_0, \eta) = 0. \tag{2.2}$$

When we replace η with $\hat{\eta}_k$ the efficient vector score with respect to the parameter of interest simplifies to

$$V_k(\theta_0, \hat{\eta}_k) = \sum_{i=1}^{k} \nabla_\theta \log f(Y_i; \theta_0, \hat{\eta}_k).$$

Similar to Tests 1.1 and 1.2 the efficient score vector $V_k(\theta_0, \hat{\eta}_k)$ is standardized with $(I^{11}(\theta_0, \hat{\eta}_k))^{1/2}$, where $I^{11}(\theta_0, \hat{\eta}_k) = \left(I_{11} - I_{12}I_{22}^{-1}I_{21}\right)^{-1}$. The test statistic for monitoring is then

$$S_k(\theta_0, \hat{\eta}_k) = (I^{11}(\theta_0, \hat{\eta}_k))^{1/2} V_k(\theta_0, \hat{\eta}_k).$$

The approximating process for S_k is a multivariate Gaussian process with independent standard Wiener process components. Hence each component can be monitored with a level α^* test which gives an overall level of significance $\alpha = 1 - (1 - \alpha^*)^d$. Therefore it is enough to define the tests for $d = 1$.

Sequential tests analogous to Test 1.1 and 1.2 are then as follows.

Test 2.1. Stop and conclude that H_0 is not supported by the data at the first k, $1 < k \leq n$ when

$$\frac{1}{\sqrt{k}} S_k(\theta_0, \hat{\eta}_k) \geq C_1(\alpha, n),$$

otherwise do not reject H_0.

The critical value $C_1(\alpha, n)$ is the same as in Test 1.1. For the two-sided version of Test 2.1 monitor $k^{-1/2}|S_k(\theta_0)|$ and reject H_0 if it is greater than $C_1^*(\alpha, n)$. Vostrikova's improvements of the critical values are also applicable here.

Test 2.2. Stop and conclude that H_0 is not supported by the data at the first k, $1 < k \leq n$ when

$$\frac{1}{\sqrt{n}} S_k(\theta_0, \hat{\eta}_k) \geq C_2(\alpha),$$

otherwise do not reject H_0.

For the two-sided version of Test 2.2 monitor $n^{-1/2}|S_k(\theta_0, \hat{\eta}_k)|$ and reject H_0 if it is greater than $C_2^*(\alpha)$.

Test 2.3. Stop and conclude that H_0 is not supported by the data at the first k, $1 < k \leq n$ when

$$\max_{0 \leq j < k} \frac{1}{\sqrt{n}} \left(S_k(\theta_0, \hat{\eta}_k) - S_j(\theta_0, \hat{\eta}_k)\right) \geq C_2^*(\alpha),$$

otherwise do not reject H_0.

Test statistics in Tests 2.1-2.3 have the same asymptotic distributions as the ones in Tests 1.1-1.3, respectively (see Gombay [30]).

Example. Let $Y_1, \ldots, Y_i, \ldots$ be independent normal random variables with mean μ and variance σ^2. Consider the problem of testing the hypotheses (2.1), which are in terms of μ and σ^2

$$\begin{aligned} H_0: &\quad \mu_i = \mu_0,\ \sigma_i^2 = \sigma^2 \in \Omega_2 \text{ for all } i \geq 1, \\ H_A: &\quad \mu_i = \mu_0,\ \sigma_i^2 = \sigma^2 \in \Omega_2 \text{ for all } i < \tau, \\ &\quad \mu_i = \mu_A,\ \sigma_i^2 = \sigma^2 \in \Omega_2 \text{ for all } i \geq \tau, \end{aligned}$$

where $\mu \in \Omega_1 = \mathbb{R}$ and $\sigma \in \Omega_2 = (0, \infty) \subset \mathbb{R}$, hence $p = d = 1$. Here μ is the parameter of interest and σ^2 is a nuisance parameter. Without loss of generality we can assume that $\mu_0 = 0$.

For the normal random variables

$$\log f(y; \mu, \sigma^2) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log\sigma^2 - \frac{(y-\mu)^2}{2\sigma^2}.$$

The restricted maximum likelihood estimator of σ^2 is the solution of (2.2):

$$\sum_{i=1}^{k}\left(\frac{1}{2\sigma^2} - \frac{Y_i^2}{2(\sigma^2)^2}\right) = 0.$$

Then $\hat{\sigma}_k^2 = \sum_{i=1}^{k} Y_i^2/k$. We have

$$V_k(\mu, \hat{\sigma}_k^2) = \sum_{i=1}^{k}\frac{\partial}{\partial\mu}\log f(Y_i; \mu, \hat{\sigma}_k^2) = \sum_{i=1}^{k}\frac{Y_i - \mu}{\hat{\sigma}_k^2}$$

The components of the informations matrix are

$$I_{11} = -E\left[\frac{\partial^2}{\partial\mu^2}\log f(y; \mu, \sigma^2)\right] = \frac{1}{\sigma^2},$$

$$I_{12} = I_{21} = -E\left[\frac{\partial^2}{\partial\mu\partial\sigma^2}\log f(y; \mu, \sigma^2)\right] = 0,$$

$$I_{22} = -E\left[\frac{\partial^2}{\partial(\sigma^2)^2}\log f(y; \mu, \sigma^2)\right] = \frac{1}{2\sigma^4}$$

and therefore $I^{11} = \sigma^2$. We would like to compare two-sided Test 2.1, two-sided Test 2.2 ans Test 2.3. The two-sided Test 2.1 statistic is

$$\frac{1}{\sqrt{k}}|S_k| = \frac{|\sum_{i=1}^{k} Y_i|}{\sqrt{\sum_{i=1}^{k} Y_i^2}}.$$

The two-sided Test 2.2 statistic is

$$\frac{1}{\sqrt{n}}|S_k| = \frac{|\sum_{i=1}^{k} Y_i|}{\sqrt{\frac{n}{k}\sum_{i=1}^{k} Y_i^2}}.$$

Test 2.3 statistic is

$$\max_{0\le j<k} \frac{1}{\sqrt{n}}\left(S_k(0,\hat{\sigma}_k^2) - S_j(0,\hat{\sigma}_k^2)\right) = \max_{0\le j<k} \frac{\sum_{i=j+1}^{k} Y_i}{\sqrt{\frac{n}{k}\sum_{i=1}^{k} Y_i^2}}.$$

Tables 2.1 and 2.2 show the results of small simulation study on the power and average sample number. Test 2.1 is prone to early stopping at the beginning due to the weight of $1/\sqrt{k}$. So, it is recommended to start monitoring only after 10 observations have been obtained. This is not an issue, since the theoretical results are asymptotic. In the simulations presented below testing started after 10 observations, truncation points are $n = 100, 200, 400$, the level of significance $\alpha = 0.05$. The critical value for Test 2.1 is $C_1^*(0.05, 100) = 3.07$, $C_1^*(0.05, 200) = 3.12$, $C_1^*(0.05, 400) = 3.16$ using Vostrikova's approximation, $C_2^*(0.05) = C_3(0.05) = 2.24$ for Tests 2.2 and 2.3. There were 5000 simulations performed for each scenario.

Table 2.1 shows that for all the three tests the power is increasing in n and is decreasing in τ. Between Test 2.1 and 2.2, Test 2.2 always has higher power. Test 2.3 shows very good performance. It has high power even when a change happens late.

Table 2.2 provides average sample numbers. ASN is increasing in τ. When $\tau = 1$, Test 2.1 has the smallest ASN, while reaching high power for large changes. Test 2.3 has the smallest ASN when n and τ are large.

Table 2.1: Monitoring Normal Data: Simulated Power. The initial mean parameter $\mu_0 = 0$, nuisance parameter $\sigma = 1$, μ_A varies from 0 to 0.6. Number of simulations is 5000.

n	τ	μ_A	Test 2.1	Test 2.2	Test 2.3
100	1	0.0	0.01	0.04	0.04
		0.2	0.20	0.45	0.53
		0.4	0.83	0.96	0.97
		0.6	1.00	1.00	1.00
	50	0.0	0.01	0.04	0.04
		0.2	0.03	0.13	0.22
		0.4	0.15	0.42	0.65
		0.6	0.41	0.76	0.95
200	1	0.0	0.02	0.05	0.04
		0.2	0.48	0.76	0.82
		0.4	1.00	1.00	1.00
		0.6	1.00	1.00	1.00
	100	0.0	0.01	0.04	0.04
		0.2	0.07	0.24	0.41
		0.4	0.37	0.72	0.93
		0.6	0.84	0.97	1.00
400	1	0.0	0.02	0.05	0.04
		0.2	0.86	0.97	0.98
		0.4	1.00	1.00	1.00
		0.6	1.00	1.00	1.00
	200	0.0	0.02	0.05	0.04
		0.2	0.15	0.43	0.68
		0.4	0.80	0.97	1.00
		0.6	1.00	1.00	1.00

Table 2.2: Monitoring Normal Data: Average Sample Number. The initial mean parameter $\mu_0 = 0$, nuisance parameter $\sigma = 1$, μ_A varies from 0 to 0.6. Number of simulations is 5000.

n	τ	μ_A	Test 2.1	Test 2.2	Test 2.3
100	1	0.0	99	99	99
		0.2	91	84	87
		0.4	53	52	60
		0.6	25	35	44
	50	0.0	99	99	99
		0.2	99	97	97
		0.4	96	90	90
		0.6	90	80	81
200	1	0.0	198	198	198
		0.2	155	145	146
		0.4	58	77	85
		0.6	26	52	61
	100	0.0	198	198	198
		0.2	196	190	187
		0.4	185	169	162
		0.6	161	146	143
400	1	0.0	395	396	397
		0.2	216	221	221
		0.4	60	112	120
		0.6	27	78	87
	200	0.0	396	396	397
		0.2	387	368	354
		0.4	333	304	287
		0.6	282	267	259

Chapter 3

Binary Data

Identical Bernoulli random variables.

Binary data arise when an observation takes one of two possible values. For example, "yes", if a person dies after surgery, of "no", if the person survives. If the two outcomes are coded as zero and one, the data can be modeled by Bernoulli random variables with probability mass function $f(y;p) = p^y(1-p)^{1-y}$. This means that the value 1, "success" occurs with probability p and the value 0, "failure", with probability $1-p$. First, we will consider a simple case of monitoring parameter p by testing the null hypothesis $H_0 : p = p_0$. Let $Y_1, \ldots, Y_i, \ldots$ be independent identically distributed Bernoulli random variables with parameter p. Then the standardized score statistic

$$S_k(p_0) = \sum_{i=1}^{k} \frac{y_i - p_0}{\sqrt{p_0(1-p_0)}}.$$

Tables 3.1 and 3.2 contain small simulation results when $p_0 = 0.07$, truncation point $n = 5000$ and $\tau = 1, 500, 2500$. Test 1.3 has the highest power and Test 1.1 has the lowest. However, when the change in parameter is large, all tests have power 1, while the average sample numbers of Tests 1.2 and 1.3 are considerably larger when the change happens early.

Adjusting for covariates.

In monitoring it is often necessary to take into account other sources of variation that are specific to each observation. For example, a very sick person has

Table 3.1: Monitoring Binary Data: Simulated Power. The initial parameter $p_0 = 0.07$, p_A varies from 0.07 to 0.13. Number of simulations is 5000.

n	p_A	Test 1.1	Test 1.2	Test 1.3
1	0.07	0.05	0.05	0.06
	0.09	0.99	1.00	1.00
	0.11	1.00	1.00	1.00
	0.13	1.00	1.00	1.00
500	0.07	0.05	0.05	0.05
	0.09	0.96	1.00	1.00
	0.11	1.00	1.00	1.00
	0.13	1.00	1.00	1.00
2500	0.07	0.05	0.05	0.06
	0.09	0.39	0.73	0.92
	0.11	0.98	1.00	1.00
	0.13	1.00	1.00	1.00

Table 3.2: Monitoring Binary Data: Average Sample Number. The initial parameter $p_0 = 0.07$, p_A varies from 0.07 to 0.13. Number of simulations is 5000.

n	p_A	Test 1.1	Test 1.2	Test 1.3
1	0.07	4782	4942	4935
	0.09	1455	2031	1953
	0.11	362	1008	996
	0.13	160	672	672
500	0.07	4786	4939	4935
	0.09	2470	2519	2291
	0.11	1196	1508	1405
	0.13	894	1169	1105
2500	0.07	4797	4938	4932
	0.09	4563	4320	3977
	0.11	3644	3512	3272
	0.13	3184	3169	3011

a higher risk of dying after a surgery than a healthy person. So, for monitoring in medicine the baseline risks should be taken into account to prevent false alarms, accusations in adverse outcomes that are due to patients prior risks rather than treatment or doctor performance.

The procedure that accounts for other covariates is called risk-adjustment. Consider the situation where $Y_1, \ldots, Y_i, \ldots$ are independent Bernoulli random variables with success probabilities depending on covariates. Logistic models are used to account for the dependence, that is for $i = 1, 2, \ldots$ success

probabilities are given by logit linear functions

$$\log \frac{p_i}{1-p_i} = \alpha + \beta_1 x_{i1} + \cdots + \beta_r x_{ir} = \alpha + \beta x_i^T,$$

where $(x_{i1}, \ldots, x_{ir})$ represents an r-vector of covariates associated with the i-th subject, α is an intercept and $\beta = (\beta_1, \ldots, \beta_r)$ is a vector of parameters.

There are many reasons for considering the logit function $\log(p/(1-p))$. One of them is that p itself cannot be a linear function of covariates as it is bounded by 0 and 1 and $\alpha + \beta x^T$ is unbounded. The odds function $p/(1-p)$ also cannot take negative values and is not symmetric with respect to 1. For example, if odds of success is 1/2, then odds of failure is 2/1=2. The logit function can take negative and positive values and odds of success have the same distance from 0 as odds of failure ($\log(1/2) = -\log(2)$), which is helpful for interpretations. Also, the Bernoulli distribution is an exponential family distribution and the logit is its natural parameter.

The Odds Ratio.

When adjustment for covariates is done, each subject has a different baseline score. We need a different measure of risk in order to formulate statistical hypotheses and to perform monitoring. For Bernoulli random variables it is natural to use the odds ratio.

The classical definition of the odds ratio is that it is the ratio of the odds of an event occurring in one group to the odds of it occurring in another group. If a probability of an event is p_i^1 if the subject i is in one group and p_i^2 if the subject is in the second group, then the odds ratio is

$$R_i = \frac{p_i^1/(1-p_i^1)}{p_i^2/(1-p_i^2)}.$$

The definition can also be used if p_i^2 represents a historical risk for subject i and p_i^1 is the risk under the new condition like new treatment or a particular doctor. This means that the odds ratio measures an effect, e.g. of a new treatment or doctor's performance, on the subject's baseline risk.

Defining the hypotheses H_0 and H_A based on the odds ratio allows monitoring risks in terms of a single parameter while taking into account the estimated baseline risks for each subject. Let the estimated risk based on the current conditions be p_i. Then $R_0 = 1$ may mean no effect of new treatment or it may mean that the risk of death after surgery does not depend on whether the surgery was performed by this particular surgeon. Given an estimated risk of an event equals to p_i, the odds of the event equals $R_0 p_i/(1-p_i)$. Under H_A the odds of the event after the change is $R_A p_i/(1-p_i)$. This corresponds to a probability of the event equal to $R_A p_i/(1-p_i+R_A p_i)$ after the change. If $R_0 = 1$ and $R_A = 2$ (detecting a doubling of the odds of event), then, for example, a subject with the estimated risk $p_i = 1/2$ would have a risk of $p_i = 2/3$ after the change under the alternative hypothesis. In order to detect increases in R we test

$$\begin{array}{ll} H_0: & R_i = R_0 \text{ for all } i \geq 1 \text{ versus} \\ H_A: & R_i = R_0 \text{ for all } i < \tau, \\ & R_i = R_A > R_0 \text{ for all } i \geq \tau. \end{array}$$

In this case $\theta = R$ and we denote the corresponding density function of the Bernoulli random variables $f(y;R)$. Then

$$\log f(y;R) = y \log p(R) + (1-y) \log (1-p(R)),$$

where $p(R) = Rp/(1-p+Rp)$ is the probability of failure. Then

$$\frac{\partial}{\partial R} \log f = \left(\frac{\partial}{\partial p} \log f \right) p',$$

where p' is a derivative of p with respect to R. To calculate $I(R)$ we need $\frac{\partial^2}{\partial R^2} \log f(y;R)$. By taking the second derivative we obtain

$$\begin{array}{rcl} \frac{\partial^2}{\partial R^2} \log f(y;R) = \frac{\partial}{\partial R} \left(\frac{\partial}{\partial R} \log f \right) & = & \frac{\partial}{\partial R} \left((\frac{\partial}{\partial p} \log f) p' \right) \\ & = & \left(\frac{\partial^2}{\partial p^2} \log f \right) p' + \left(\frac{\partial}{\partial p} \log f \right) p''. \end{array}$$

Here

$$\frac{\partial}{\partial p} \log f = \frac{y-p}{p(1-p)}.$$

Hence

$$I(R) = -E\frac{\partial^2}{\partial R^2}\log f(y;R) = I(p)p' + p''E\frac{y-p}{p(1-p)} = I(p)p'.$$

It is well known that $I(p) = (Var(y))^{-1} = p(1-p)$. Therefore the standardized score statistic for the truncated sequential tests is

$$S_k(R_0) = I^{-1/2}(R_0)V_k(R_0) = \sum_{i=1}^{k}\frac{y_i - p_i^0}{\sqrt{p_i^0(1-p_i^0)}},$$

where $p_i^0 = R_0 p_i/(1 - p_i + R_0 p_i)$.

Example: Monitoring Surgical Performance.

Sequential testing for monitoring surgical outcomes is popular nowadays. The necessity of such monitoring became especially clear after doctor Harold Shipman's case [31]. He murdered over 200 of his patients during the period of 1975 to 1998 in England, UK and became one of the most prolific serial killers. Another case that was a starting point for monitoring of clinical performance is the Bristol Royal Infirmary Inquiry [32] concerning mortality of children under one year of age after open-heart surgeries. Various industrial quality control procedures were employed for the monitoring and adjusted for case-mix. The popular procedures include risk-adjusted sequential probability ratio test (SPRT) [33], risk-adjusted cumulative sum chart (CUSUM) [34], variable life-adjusted display (VLAD) [35], [36] and truncated sequential and change-point detection tests [37]. In this example we would like to compare Tests 1.1-1.3 to the risk-adjusted CUSUM in the context of monitoring surgical performance. We apply the tests to cardiac surgery data [34] and use Monte Carlo simulations for comparison of empirical power and average sample numbers.

The data were previously analyzed by various methods in [34], [33] and [37]. The data are the 30-day post-operative mortality from individual coronary artery bypass graft operations at cardiac surgery center in UK for the period 1992-1998. Let y_i be the outcome for patient i and $y_i = 1$ if patient i dies within 30 days of operation and $y_i = 0$ otherwise. This is modeled by Bernoulli

random variables, $f(y; p_i) = p_i^y(1-p_i)^{1-y}$. For risk-adjustment, the data for the years 1993 and 1994 were used to find that the logistic model

$$\log \frac{p_i}{1-p_i} = -3.68 + 0.077x_i, \tag{3.1}$$

with only x_i, a Parsonnet score, as an explanatory variable, was appropriate [34]. The Parsonnet score is based on a combination of other explanatory variates such as age, hypertension and a number of reoperations. Based on this model, the lowest risk patients, with zero Parsonnet score, were estimated to have a risk of death of just 2.5%, while the patients with the highest risk, with Parsonnet score higher than 70, had an estimated mortality rate of 86%. If the Parsonnet score is higher than 20, the patient is considered to have an extremely high risk of mortality. The Parsonnet score is itself an estimated risk of mortality within 30 days of cardiac surgery. The logistic model that is used to compute the Parsonet score was built by Parsonnet (1989) [38] using 3500 operations and 14 risk factors that are easy to asess. The model was tested on different hospitals.

Tables 3.3-3.5 contain simulation results on type I error, power and average sample number (ASN) for the risk-adjusted two-sided Tests 1.1 and 1.2, Test 1.3 and CUSUM. The data is simulated as follows. First, Parsonnet scores are generated as random numbers following Poisson distribution with mean 13. This implies an average mortality risk of 6.4%, which is close to the mortality rate of 6.5% estimated from the data for 1992 and 1993. Then, mortality risks are predicted using the logistic model (3.1) and random Bernoulli numbers are generated using the estimated risks. These Bernoulli numbers and the estimated risks are used to form the test statistics. The control limit for CUSUM is set to $h = 4.5$. It was shown in [34] that when $h = 4.5$ ASN is around 9600 under the null hypothesis. We stop the CUSUM algorithm after 9600 observations. For the CUSUM we also need to specify an odds ratio under the alternative hypotheis. We consider detecting doubling of the odds of death and therefore set $R_A = 2$ ($R_0 = 1$). The truncation points for Tests 1.1-1.3 are

$n = 5000$ and 9600. The truncation point of $n = 5000$ approximately corresponds to the number of patients that were operated between 1994 and 1998 in the cardiac surgery data. For Tests 1.1-1.3 the significance level is set to $\alpha = 0.05$ and the critical values are $C_1^*(0.05, 5000) = 3.28$, $C_1^*(0.05, 9600) = 3.3$ (using Vostrikova's approximation), $C_2^*(0.05) = C_3(0.05) = 2.24$. The monitoring starts after 10 observations for all tests as in the previous example. There are 5000 simulations performed for each scenario.

Table 3.3 compares the tests in terms of type I error. Tests 1.1-1.3 have empirical type I error close to 0.05 as expected. The CUSUM procedure with $h = 4.5$ stopped with rejection of H_0 79% of the time under the null hypothesis of no change. This implies that most likely there will be a false alarm when no change happened.

Table 3.3: Monitoring Odds Ratio: Simulated Type I Error. The initial odds ratio $R_0 = 1$, $\alpha = 0.05$. The CUSUM algorithm is stopped after 9600 observations, the truncation points are $n = 5000, 9600$. Number of simulations is 5000.

n	CUSUM	Test 1.1	Test 1.2	Test 1.3
5000	NA	0.0426	0.0550	0.0570
9600	0.7918	0.0456	0.0436	0.0530

Tables 3.4 and 3.5 analyze the power as follows. The odds ratio after change varies over $R_A = 1.25, 1.5, 1.75, 2$. The columns *Total* and *Before* τ represent the overall proportion of stops (power) and the proportion of stops before the change point τ, respectively. This separation is necessary because the CUSUM test stops with probability one even under the hypothesis of no change. For the same reason the empirical power of CUSUM is one in almost all cases. It is not 1 in one scenario, because we force the algorithm to stop after 9600 observations. All other tests also show very high power in most situations. From *Before* τ column of Table 3.5 we see that for the CUSUM test the probability of stopping before the change point increases as τ increases and can be as high as 56% when the change point is far from the beginning. In other words, there are too many false alarms. The power of Test 1.3 is almost

always high (> 0.95). Tests 1.1 and 1.2 have smaller power due to the extra terms in the test statistics corresponding to the before change observations. Tests 1.1-1.3 have lower power when the change is small ($R_A = 1.25$), especially Test 1.1 with power of 0.21 when $\tau = 2500$ and $n = 5000$. As the value of truncation point n increases the power increases.

Table 3.4: Monitoring Odds Ratio: Simulated Power of Tests 1.1-1.3. The initial odds ratio $R_0 = 1$, R_A varies from 1.25 to 2, the truncation points are $n = 5000, 9600$. Number of simulations is 5000.

n	τ	R_A	Test 1.1		Test 1.2		Test 1.3	
			Total	Before τ	Total	Before τ	Total	Before τ
5000	1	1.25	0.88	0.00	0.98	0.00	0.99	0.00
		1.50	1.00	0.00	1.00	0.00	1.00	0.00
		1.75	1.00	0.00	1.00	0.00	1.00	0.00
		2.00	1.00	0.00	1.00	0.00	1.00	0.00
	500	1.25	0.78	0.03	0.95	0.00	0.98	0.00
		1.50	1.00	0.03	1.00	0.00	1.00	0.00
		1.75	1.00	0.03	1.00	0.00	1.00	0.00
		2.00	1.00	0.03	1.00	0.00	1.00	0.00
	2500	1.25	0.20	0.05	0.50	0.00	0.75	0.00
		1.50	0.84	0.04	0.98	0.00	1.00	0.00
		1.75	1.00	0.04	1.00	0.00	1.00	0.00
		2.00	1.00	0.04	1.00	0.00	1.00	0.00
9600	1	1.25	1.00	0.00	1.00	0.00	1.00	0.00
		1.50	1.00	0.00	1.00	0.00	1.00	0.00
		1.75	1.00	0.00	1.00	0.00	1.00	0.00
		2.00	1.00	0.00	1.00	0.00	1.00	0.00
	960	1.25	0.98	0.03	1.00	0.00	1.00	0.00
		1.50	1.00	0.03	1.00	0.00	1.00	0.00
		1.75	1.00	0.04	1.00	0.00	1.00	0.00
		2.00	1.00	0.04	1.00	0.00	1.00	0.00
	4800	1.25	0.45	0.04	0.79	0.00	0.95	0.00
		1.50	0.99	0.03	1.00	0.00	1.00	0.00
		1.75	1.00	0.04	1.00	0.00	1.00	0.00
		2.00	1.00	0.04	1.00	0.00	1.00	0.00

Table 3.6 analyzes the average sample number of the three truncated tests. Column *Total* represents the overall average sample number, while column *When Stopped* is the average sample number when an algorithm stops before the truncation point ($n = 5000$ or 9600). Table 3.5 provides average sample numbers for the CUSUM. Out of the four tests, CUSUM has the smallest ASN,

Table 3.5: Monitoring Odds Ratio: Simulated Power, Average Sample Number and Conditional Average Sample Number for CUSUM. *Before* τ is the proportion of stops before τ. The initial odds ratio $R_0 = 1$, R_A varies from 1.25 to 2. The algorithm is stopped after 9600 observations. Number of simulations is 5000.

τ	R_A	Total Power	Before τ	ASN	Conditional ASN
1	1.25	1.00	0.00	1221	1221
	1.50	1.00	0.00	457	457
	1.75	1.00	0.00	260	260
	2.00	1.00	0.00	174	174
500	1.25	1.00	0.06	1573	1661
	1.50	1.00	0.07	888	931
	1.75	1.00	0.07	705	735
	2.00	1.00	0.07	634	658
960	1.25	1.00	0.14	1908	2137
	1.50	1.00	0.13	1281	1397
	1.75	1.00	0.13	1108	1195
	2.00	1.00	0.13	1035	1115
2500	1.25	1.00	0.34	2860	3679
	1.50	1.00	0.33	2365	2933
	1.75	1.00	0.35	2204	2737
	2.00	1.00	0.33	2187	2653
4800	1.25	0.99	0.54	3878	5932
	1.50	1.00	0.53	3614	5240
	1.75	1.00	0.55	3435	5028
	2.00	1.00	0.56	3347	4959

but when the change happens after 500 observations the small ASN is due to erroneous stopping, that is before τ. When change happens right at the first observation, CUSUM detects it faster unless the the change is large ($R_A \geq 2$). In this case Test 1.1 has a little smaller ASN. When the change happens late the average sample number for CUSUM is smaller than τ. For late changes, out of Tests 1.1-1.3, Test 1.3 stops earlier and has higher power. In general, when n increases average sample number also increases for the truncated tests.

Table 3.5 contains conditional average sample numbers of CUSUM, given the change point has been passed. It shows that conditional ASN for CUSUM is the smallest ASN among the tests except for the cases when $\tau = 1$ and $R_A = 2$. In those situations Test 1.1 has the smallest ASN.

In general, one should decide on what is more important for a given prob-

Table 3.6: Monitoring Odds Ratio: Average Sample Number for Tests 1.1-1.3. The initial odds ratio $R_0 = 1$, R_A varies from 1.25 to 2, the truncation points are $n = 5000, 9600$. Number of simulations is 5000.

n	τ	R_A	Test 1.1		Test 1.2		Test 1.3	
			Total	When Stopped	Total	When Stopped	Total	When Stopped
5000	1	1.25	2337	1976	2640	2589	2513	2478
		1.50	629	629	1328	1328	1300	1300
		1.75	289	289	897	897	888	888
		2.00	169	169	680	680	679	679
	500	1.25	3298	2826	3079	2985	2785	2745
		1.50	1546	1546	1835	1835	1689	1689
		1.75	1090	1090	1398	1398	1307	1307
		2.00	904	904	1182	1182	1115	1115
	2500	1.25	4677	3349	4582	4170	4287	4051
		1.50	4055	3869	3811	3781	3500	3498
		1.75	3494	3490	3399	3399	3188	3188
		2.00	3193	3193	3179	3179	3017	3017
9600	1	1.25	2499	2470	3646	3646	3511	3511
		1.50	639	639	1849	1849	1821	1821
		1.75	298	298	1251	1251	1242	1242
		2.00	166	166	942	942	942	942
	960	1.25	4467	4343	4613	4602	4192	4192
		1.50	2209	2209	2806	2806	2609	2609
		1.75	1687	1687	2209	2209	2081	2081
		2.00	1462	1462	1902	1902	1807	1807
	4800	1.25	8657	7485	8157	7768	7468	7363
		1.50	6906	6880	6633	6631	6189	6189
		1.75	6043	6043	6044	6044	5744	5744
		2.00	5678	5678	5753	5753	5522	5522

lem, an early detection at the expense of very possible false alarms, in which case CUSUM performs better, or a guaranteed probability of type I error at the expense of delays in the change detection. In the latter case, for detecting small changes it is better to use Test 1.3 or 1.2, for detecting early large changes it is recommended to use Test 1.1 as it stops faster. When it is expected that the change will happen late in a sequence Test 1.3 has the highest power with the smallest average sample number. For detecting large changes in situations similar to the one in the example it might also be more optimal to set a smaller truncation point. In that case the power of the truncated sequential tests is still very high, but the delay in detection is much shorter.

Finally, we apply Tests 1.1-1.3 and CUSUM to the cardiac surgery data, the data for seven surgeons for the period 1994-1998. Risk-adjustment is performed using model (3.1). We set $n = 5000, 9600$ and $\alpha = 0.05$ for the truncated tests. We assume $R_A = 2$ and $h = 4.5$ for the CUSUM. Plots of the test statistics when $n = 5000$ are presented in Figures 3.1- 3.4. Test 1.1 and CUSUM signal a change for surgeons 1 and 2. Test 1.1 stops at 355 for surgeon 1 and at 236 for surgeon 2. CUSUM stops at 260 and 220, respectively. CUSUM also signals a change for surgeon 7 after 111 observations.

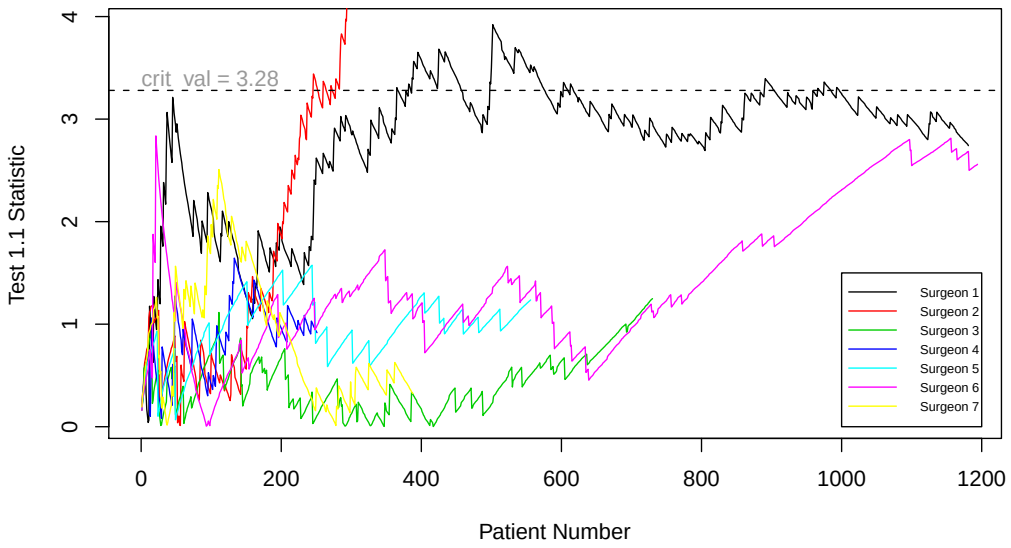

Figure 3.1: Test 1.1 statistic for the seven surgeons ($n = 5000$, $C_1^*(0.05, 5000) = 3.28$).

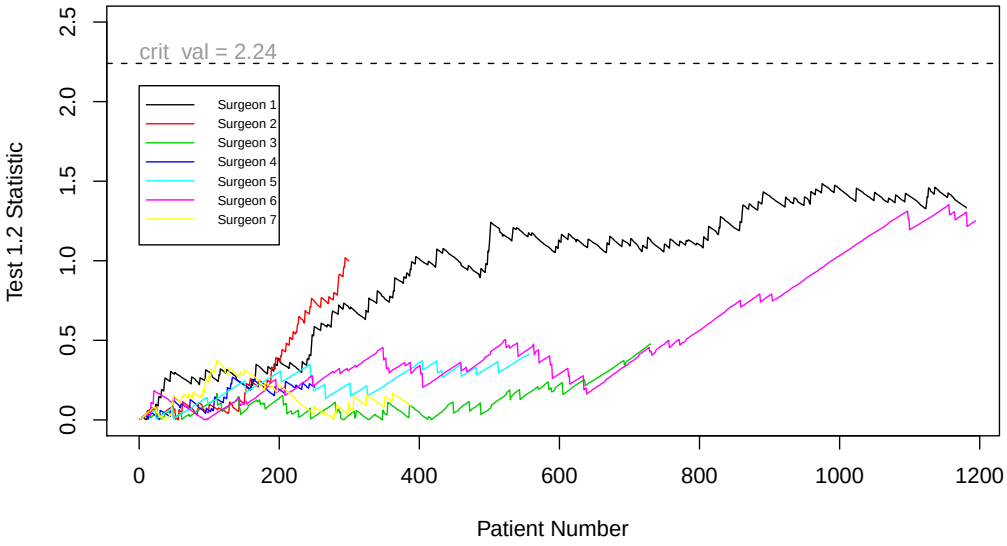

Figure 3.2: Test 1.2 statistic for the seven surgeons ($n = 5000$, $C_2^*(0.05) = 2.24$).

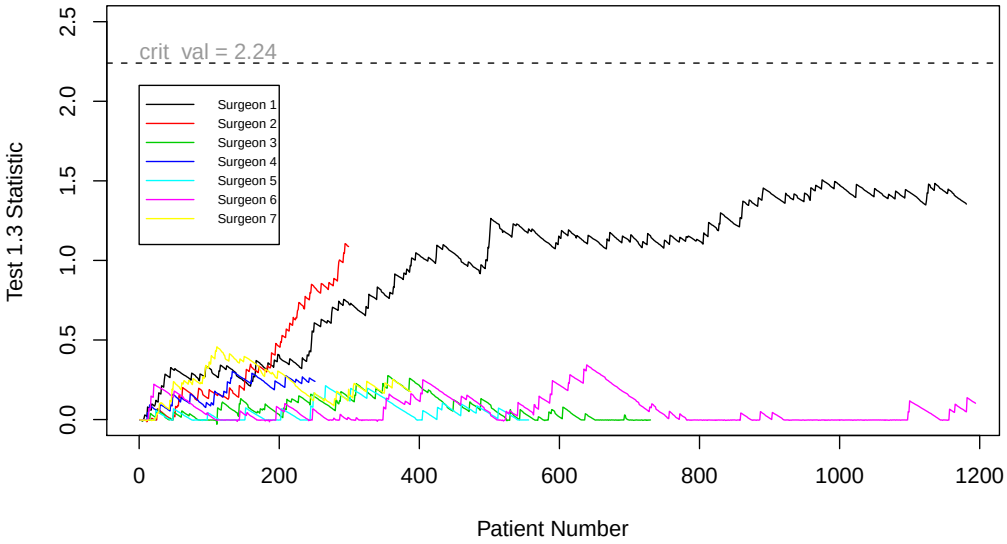

Figure 3.3: Test 1.3 statistic for the seven surgeons ($n = 5000$, $C_3(0.05) = 2.24$).

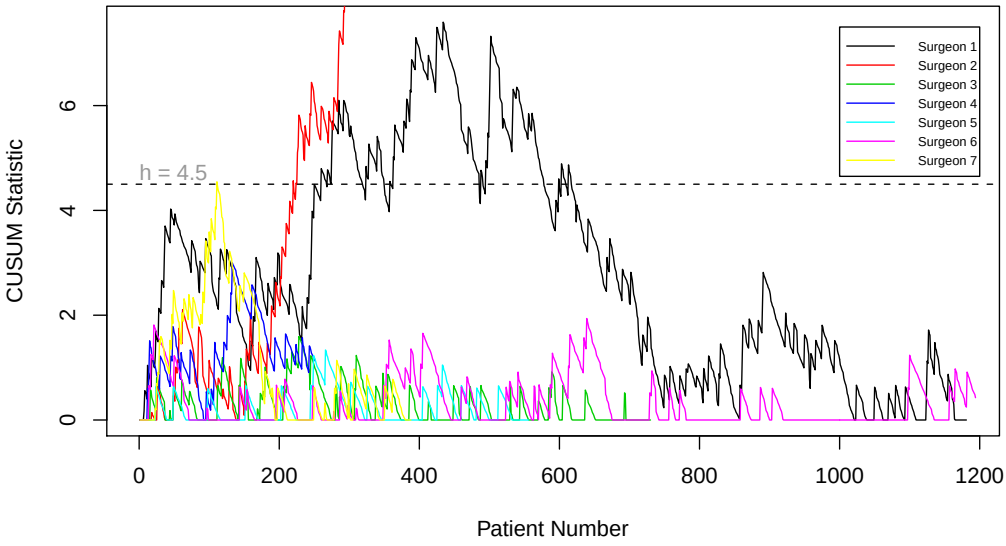

Figure 3.4: CUSUM test statistics for the seven surgeons ($h = 4.5$).

Chapter 4

Longitudinal Data

Sometimes the outcome for an individual subject consists of a series of repeated measurements. Longitudinal studies are often used in psychology to study developmental trends across the life span, in sociology to study life events throughout lifetimes or generations, in medicine to find predictors of certain diseases. In contrast to a cross-sectional study repeated several times to observe an outcome a longitudinal study looks at the outcome on the same people. That is the sample of a longitudinal study contains the same individuals at different time points. In this case the differences observed in those people are less likely to be the result of differences due to time or differences across generations.

In this chapter we consider sequential methods for longitudinal data in treatments comparison set up. Parametric approaches for longitudinal data are often complicated and computationally expensive. We consider nonparametric approaches here.

Suppose there are I treatment groups with J_i subjects in i-th group. Let $Y^i_{j1}, \ldots, Y^i_{jP}$ be P observations on subject j in i-th group. Assume $Y^i_j = (Y^i_{j1}, \ldots, Y^i_{jP})^T$, $j = 1, \ldots, J_i$ are random vectors with density function f^i, $i = 1, \ldots, I$. We are interested in testing

$$\begin{aligned} H_0 &: \quad f^1 = \cdots = f^I = f \text{ versus} \\ H_A &: \quad \text{not } H_0. \end{aligned}$$

For a fixed p the sequential rank of Y_{jp}^i is defined as

$$\begin{aligned} R_{jp}^i &= 1 + \#\{Y_{kp}^l |\ Y_{kp}^l \leq Y_{jp}^i, Y_{kp}^l \text{ has already been observed,} \\ &\quad \text{not including } Y_{jp}^i \text{ itself, for } l = 1, \ldots, I \text{ and } k = 1, \ldots, J_l\}. \end{aligned}$$

For example, if Y_{jp}^i is the first p-th measurement, $R_{jp}^i = 1$. The second p-th measurement is of sequential rank 2 if it is larger or equal to the first one (Y_{jp}^i) and 1 otherwise. The third p-th observation may have sequential rank 1, 2 or 3. The sequential ranks that have already been assigned do not change when a new observation is available. Under H_0 and for fixed p, R_{jp}^i follow independent discrete uniform distributions [39], that is R_{jp}^i can take values $1, \ldots, m_{jp}^i$ with equal probability, where m_{jp}^i is the number of available p-th measurements

$$\begin{aligned} m_{jp}^i &= 1 + \#\{Y_{kp}^l |\ Y_{kp}^l \text{ has been observed before } Y_{jp}^i, \\ &\quad \text{for } l = 1, \ldots, I \text{ and } k = 1, \ldots, J_l\}. \end{aligned}$$

The mean and the variance of R_{jp}^i are then given by

$$E\left(R_{jp}^i\right) = \frac{m_{jp}^i + 1}{2}$$

and

$$Var\left(R_{jp}^i\right) = \frac{\left(m_{jp}^i\right)^2 - 1}{12}.$$

At time t, subjects with all P repeated measurements observed are called complete. For each of the complete subjects the sum of standardized sequential ranks is introduced

$$S_j^i = \sum_{p=1}^{P} Z_{jp}^i,$$

where

$$Z_{jp}^i = \frac{R_{jp}^i - (m_{jp}^i + 1)/2}{\sqrt{\left(\left(m_{jp}^i\right)^2 - 1\right)/12}}.$$

Note that S_j^i, $i = 1, \ldots, I$, $j = 1, \ldots, J_i$, are independent.

To formulate sequential tests that use Weiner process approximation, like Tess 1.1-1.3, S_j^i needs to be standardized. Under H_0, the mean of S_j^i is zero. To find the variance, we note that S_j^i is the sum of dependent term, as ranks of repeated measurements on the same subject may be dependent. Then

$$\begin{aligned} Var\left(S_j^i\right) &= \sum_{p=1}^{P} \sum_{q=1}^{P} Cov\left(Z_{jp}^i, Z_{jq}^i\right) \\ &= \sum_{p=1}^{P} Cov\left(Z_{jp}^i, Z_{jp}^i\right) + 2 \sum_{p,q:p<q} Cov\left(Z_{jp}^i, Z_{jq}^i\right). \end{aligned}$$

Let $J^p(t)$ denote the number of subjects with observed p-th measurement at time t in all groups. Then, at time t the empirical pooled covariance estimator for $Cov\left(Z^i_{jp}, Z^i_{jq}\right)$, $1 \le p < q \le P$, is

$$\rho_{p,q}(t) = \frac{1}{J^q\left(t\right)} \sum_{i,j:q\text{th measurement observed}} \left(Z^i_{jp} - \bar{Z}_p\right)\left(Z^i_{jq} - \bar{Z}_q\right),$$

where

$$\bar{Z}_p = \frac{1}{J^p\left(t\right)} \sum_{i,j:p\text{th measurement observed}} Z^i_{jp}, \quad p = 1, \ldots, P.$$

Thus an estimator of $Var\left(S^i_j\right)$, $i = 1, \ldots, I$, at time t is

$$(\hat{\sigma}(t))^2 = \sum_{p=1}^{P} \rho_{p,p}(t) + 2 \sum_{p,q:p<q} \rho_{p,q}(t).$$

Let $J_i(t)$ denote the number of complete subjects at time t. Now we can define the test statistic

$$S^i(t) = \sum_{j=1}^{J_i(t)} \frac{S^i_j}{\hat{\sigma}(t)}$$

for each group, $i = 1, \ldots, I$. The statistical process $S^i(t)$ has $J_i(t)$ independent standardized components. The test statistic is updated after each subject that became complete. The level α tests are as follows.

Test 4.1. (aka Pocock) Stop and conclude that H_0 is not supported by the data at the first time t, when for some $i = 1, \ldots, I-1$,

$$\left|\frac{1}{\sqrt{J_i\left(t\right)}} S^i\left(t\right)\right| \ge C^*_1(\alpha^*, J_i),$$

where $C^*_1(\alpha^*, J_i)$ is defined in (1.7); otherwise do not reject H_0.

Test 4.2. (aka O'Brien-Flemming) Stop and conclude that H_0 is not supported by the data at the first time t, when for some $i = 1, \ldots, I-1$,

$$\left|\frac{1}{\sqrt{J_i}} S^i\left(t\right)\right| \ge C^*_2(\alpha^*),$$

where $C^*_2(\alpha^*)$ is defined in (1.8); otherwise do not reject H_0.

The significance level α is adjusted to comparing of I groups through α^*. By the independence of the $S^i(t)$, $i = 1, \ldots, I-1$, if α^* satisfies

$$1 - \alpha = (1 - \alpha^*)^{I-1}$$

the overall type I error rate is α. Notice that it is enough to monitor $I-1$ treatment groups, which is proved in [40]. Again, Test 4.1 can be less conservative with the use of other critical values.

Performance of Tests 4.1 and 4.2 is similar to that of Tests 1.1 and 1.2, respectively. Test 4.1 stops early for large treatment differences when maximal sample sizes are large, while Test 4.2 is more powerful when deviations from H_0 are small.

It is also important to remind that the statistics are not updated after each new observation, but rather when this observation is the P-th measurement on some subject. That is, at time t an analysis performed in case $n(t) = J_1(t)+\cdots+J_I(t)$ has increased by 1. Observations on incomplete subjects are used only in the calculations of sequential ranks R^i_{jp} and in the estimation of the covariance structure $\rho_{p,q}(t)$, $1 \le p \le q \le P$.

The above tests work best when the maximal sample size is large. An alternative test for small maximal sample sizes is based rather on signs of the sums of P standardized sequential ranks. Let

$$\mathrm{sgn}(S^i_j) = \begin{cases} 1, & \text{if } S^i_j > 0, \\ 0, & \text{if } S^i_j = 0, \\ -1, & \text{if } S^i_j < 0 \end{cases}$$

be the sign function. Under H_0 for fixed I, $\mathrm{sgn}(S^i_j)$, $j = 1,\ldots,J_i(t)$ are independent identically distributed, as functions of i.i.d. random variables, with $P(\mathrm{sgn}(S^i_j) = 1) = P(\mathrm{sgn}(S^i_j) = 1) = 1/2$. Therefore the statistical processes

$$S^i(t) = \sum_{j=1}^{J_i(t)} \mathrm{sgn}(S^i_j)$$

are simple symmetric random walks, $i = 1,\ldots,I$. The level α test based on $S^i(t)$ is as follows.

Test 4.3. Stop and conclude that H_0 is not supported by the data at the first time t, when for some $i = 1,\ldots,I-1$,

$$\left|S^i(t)\right| \ge C(\alpha^*, J_i),$$

where $C(\alpha^*, J_i) \in \{1, \ldots, J_i\}$ is the value for which

$$\frac{1}{C(\alpha^*, J_i)} \sum_{j=0}^{2C(\alpha^*, J_i)-1} \left(\frac{1-(-1)^j}{2} \right) \sin \frac{j\pi}{2} (\cos x_j)^{J_i} \frac{1+\cos x_j}{\sin x_j}$$

is closest to $1-\alpha^*$, where $x_j = j\pi/(2C(\alpha^*, J_i))$; otherwise do not reject H_0.

For small treatment differences, the power of Test 4.3 is larger than the power of Test 4.1 but smaller than that of Test 4.2 and the average sample number of Test 4.3 is similar to that of Test 4.1 and Test 4.2. For large treatment differences, ASN of Test 4.3 is larger. In general [40] recommend Test 4.2 for large sample sizes, unless early stopping is very important and large treatment differences are expected. Test 4.3 should be used when $J_i < 20$.

Example. Consider the following simulation study. Let $Y_j^i = (Y_{j1}^i, \ldots, Y_{j3}^i)$ denote the correlated observations for each subject in one of two groups $j = 1, \ldots, J_i$, $i = 1, 2$. We induce an autoregressive correlation structure by

$$Y_j^i = \mu_k + \Omega^{\frac{1}{2}} W_j^i,$$

where $\mu_1 = 1$ and μ_2 varies over the set $\{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$ and they represent the difference in means between groups, W_j^i is a vector of independent unit exponential variables and

$$\Omega = \begin{pmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{pmatrix},$$

ρ is a parameter that is less than one. An autoregressive correlation structure is very natural in longitudinal data or repeated measures designs. It indicates that two observations taken close in time within an individual tend to be more highly correlated than two observations taken far apart in time from the same individual. We set $\rho = 0.5$ in the example. We assume equal group sizes. In simulations we also assume that first J_1 vectors of repeated measures correspond to subjects from the first group and the next J_2 vectors of repeated measures correspond to subjects from the second group. Then we perform

a random permutation of numbers $1, \ldots, n$ to simulate the order in which observations were obtained. For simplicity, all three repeated measures for each subject are assumed to be observed in succession

Tables 4.1 and 4.2 show results of the simulation study on the power and average sample number. To avoid erroneous early stopping testing is started after 10 observations. The level of significance $\alpha = 0.05$. Since there are two groups in the study, $\alpha^* = \alpha = 0.05$. Let $J_1 = J_2 = 20, 50$. The critical values for the tests are $C_1^*(0.05, 20) = 2.93$, $C_1^*(0.05, 50) = 3.02$, $C_2^*(0.05) = 2.24$ and $C(0.05, 20) = 10$, $C(0.05, 50) = 16$ for Tests 4.1-4.3, respectively. There are 5000 simulations performed for each scenario.

Table 4.1: Monitoring Longitudinal Data: Simulated Power. The parameters are $\mu_1 = 1$, μ_2 varies from 1 to 2, $J_1 = J_2 = 20, 50$. Number of simulations is 5000.

J_1	μ_2	Test 4.1	Test 4.2	Test 4.3
20	1.0	0.03	0.05	0.04
	1.2	0.05	0.09	0.06
	1.4	0.10	0.21	0.12
	1.6	0.20	0.39	0.25
	1.8	0.34	0.59	0.42
	2.0	0.48	0.74	0.57
50	1.0	0.04	0.05	0.04
	1.2	0.08	0.16	0.09
	1.4	0.30	0.54	0.30
	1.6	0.62	0.85	0.61
	1.8	0.86	0.96	0.83
	2.0	0.95	0.99	0.95

Tables 4.1 and 4.2 demonstrate the performance of Tests 4.1-4.3 in terms of empirical power and average sample number. Power and ASN increase when sample size J_1 increases. The power is quite low for all three tests when the sample size is small. The power is the highest for Test 4.2 for both sample sizes. Test 4.3 has an average power and Test 4.1 has the smallest power when $J_1 = 20$. Test 4.3 and 4.1 have very similar powers when $J_1 = 50$. The average sample numbers are similar for all three tests when $J_1 = 20$. When $J_1 = 50$, Test 4.3 has the largest ASN. Average sample numbers of Tests 4.1 and 4.2 are

similar to ASN of Test 4.3 for small treatment differences. When treatment differences are moderate and large, average sample numbers of Tests 4.1 and 4.2 are smaller, but similar to each other. Test 4.1 has the smallest ASN when the treatment difference is the largest ($\mu_2 - \mu_1 = 1$).

Table 4.2: Monitoring Longitudinal Data: Average Sample Number. The parameters are $\mu_1 = 1$, μ_2 varies from 1 to 2, $J_1 = J_2 = 20, 50$. Number of simulations is 5000.

J_1	μ_2	Test 4.1	Test 4.2	Test 4.3
20	1.0	39.31	39.46	39.68
	1.2	39.18	39.19	39.53
	1.4	38.41	38.23	39.09
	1.6	37.20	36.53	38.04
	1.8	35.25	34.24	36.44
	2.0	33.23	32.29	34.78
50	1.0	97.54	98.96	99.07
	1.2	95.56	96.45	98.29
	1.4	87.38	86.49	93.45
	1.6	73.60	72.86	84.30
	1.8	59.39	61.82	73.71
	2.0	48.65	54.49	64.57

Chapter 5

Monitoring Wait Time

The long wait times at Emergency departments and not only there has become a problem, especially in Alberta, in the last 10 years. The median waiting time in Edmonton was 11.6 hours in March 2011, which is a somewhat smaller than a couple of years earlier. According to the Worlds Health Organization 70% of patients should be accesed or discharged withtin 4 hours and nobody should wait longer than 12 hours in emergency rooms.

As a part of the Five-Year Health Action Plan, Alberta Health Services (AHS) and Alberta Health and Wellness (AH&W) are tracking the progress being made at some of the largest emergency departments. They are monitoring a proportion of discharged patients with the length of stay at a particular emergency department no longer than four hours [41]. The target that 90% of patients are seen, assesed, treated and discharged within four hours has been set for March 2015.

An example used in this chapter is based on the data from emergency departments of University of Alberta and Stollery Hospitals for the period of Jan 4, 2011 to June 25, 2011. The data were provided by the Data Integration, Measurements & Reporting. For each of the two emergency departments we have weekly numbers of visits and weekly proportions of people that were discharged within the target time. The data are modeled by binomial random variables $Y_i \sim B(n_i, p_i)$. That is, y_i is the number of people discharged within four hours during i-th week. We are interested in monitoring "p_i". That is we

are interested in testing

$$\begin{aligned} H_0: &\quad p_i = p_0 \text{ for all } i \geq 1 \text{ versus} \\ H_A: &\quad p_i = p_0 \text{ for all } i < \tau, \\ &\quad p_i = p_A > p_0 \text{ for all } i \geq \tau. \end{aligned}$$

We apply Tests 1.1 and 1.2, two-sided, and Test 1.3 to verify if the data support the null hypothesis. The moving average plots are presented in the weekly summary reports. We compare the performance of Tests 1.1-1.3 to that of the exponentially weighted moving average procedure (EWMA). We monitor the following test statistics of Tests 1.1-1.3, respectively, $\frac{1}{\sqrt{k}}|S_k(p_0)|$, $\frac{1}{\sqrt{n}}|S_k(p_0)|$ and $\max\limits_{0\leq j<k\leq n} \frac{1}{\sqrt{n}}\left(S_k(p_0) - S_j(p_0)\right)$. In the case of binomial distribution

$$S_k(p_0) = \sum_{i=1}^{k} \frac{y_i - E(Y_i)}{\sqrt{Var(Y_i)}} = \sum_{i=1}^{k} \frac{y_i - n_i p_0}{\sqrt{n_i p_0(1-p_0)}}.$$

For EWMA we have to make some modifications to adjust for not identical distributions. Each week there is a different number of people that were seen at an emergency department. Therefore the formula for the control limits (1.5), that were given in the first chapter, need to be changed. Since

$$S_k = \lambda Y_k + (1-\lambda) S_{k-1}$$

is calculated recurrently, then

$$S_k = \lambda \sum_{i=1}^{k} (1-\lambda)^{k-i} Y_i + (1-\lambda)^k S_0,$$

where $S_0 = p_0 n_0$ and one has to choose the initial p_0 and n_0. Then the expected weighted average is

$$ES_k = \lambda p_0 \sum_{i=1}^{k} (1-\lambda)^{k-i} n_i + (1-\lambda)^k p_0 n_0,$$

and

$$VarS_k = \lambda^2 p_0(1-p_0) \sum_{i=1}^{k} (1-\lambda)^{2(k-i)} n_i.$$

Therefore we keep monitoring until

$$|S_k - ES_k| \geq L\lambda \sqrt{p_0(1-p_0) \sum_{i=1}^{k} (1-\lambda)^{2(k-i)} n_i}.$$

as in (1.6). This idea is also used to peform risk-adjustment (see, for example, [42], [43]).

Tables 5.1 and 5.2 provide results of Tests 1.1-1.3 and EWMA applied to data from the Stollery and University of Alberta Hospitals. The first 29 weeks were used to evaluate the historical "p_0". The tests were performed on the values for the weeks 30 to 129. The width of the control limits for EWMA was set to $L = 3$, the usual three-sigma limits. The truncation point was $n = 260$ (about the number of weeks in five years). If a truncated test did not stop after 130 available observations, and hence did not reach the truncation point $n = 260$, we write "did not stop" for stopping time.

Table 5.1: Monitoring proportion of discharged patients within four hours: Stollery Hospital. Stopping times are provided for the tests that signaled a change. The initial proportions are $p_0 = 0.687$, the historical average, and $p_0 = 0.7, 0.75, 0.8$. The sample size is 130, the truncation point is n=260.

p_0	Test 1.1	Test 1.2	Test 1.3	EWMA			
				$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$
0.687	10	21	21	10	10	10	10
0.700	10	23	23	10	10	10	10
0.750	14	35	75	14	13	13	13
0.800	10	13	did not stop	10	10	10	10

Table 5.2: Monitoring proportion of discharged patients within four hours: University of Alberta Hospital. Stopping times are provided for the tests that signaled a change. The initial proportions are $p_0 = 0.4$, the historical average, and $p_0 = 0.5$. The sample size is 130, the truncation point is n=260.

p_0	Test 1.1	Test 1.2	Test 1.3	EWMA			
				$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$
0.4	10	51	87	10	10	14	14
0.5	10	5	did not stop	10	10	10	10

From Tables 5.1 and 5.2 we see that all the tests detect a change from historical proportions. Simulation studies, presented below, show that all the tests are very sensitive to even small changes in proportions. For this reason, we provide results for other than historical values of p_0 in Tables 5.1 and 5.2. Tests 1.1, 1.2 and EWMA are two-sided. Test 1.3 is one-sided and this might

explain why it did not signal a change in certain cases. For example, it might mean that the proportion of patients discharged from the emergency departments within four hours has not reached 80% for the Stollery Hospital and has not reached 50% for the University of Alberta Hospital.

Tables 5.3-5.6 contain simulation results for Tests 1.1-1.3 and EWMA. For simplicity we assume that all $n_i = 1000$. We set $p_0 = 0.4$, $\alpha = 0.05$, $n = 260$, each scenario is repeated 5000 times. In Table 5.3 we report the average sample number (ASN) and the maximum sample number (max SN) of EWMA over 5000 simulations under the null hypothesis of $p_i = p_0 = 0.4$, $i \geq 1$. The maximum ASN of EWMA is 1391 when $\lambda = 0.05$. Since the distribution of sample numbers of EWMA is skewed to the right, EWMA will stop before 1500 most of the time. In all simulations below we stop EWMA algorithm after 1500 observations. Table 5.4 shows that Tests 1.1-1.3 have empirical power 1 when the alternative parameter is larger than 0.5. EWMA is known to stop with probability 1, and even truncated at 1500 it has an empirical power of 1 (Table 5.5). Moreover, Table 5.5 shows that EWMA stops before the change even happened in up to 35% times (see column *Before* τ). Table 5.6 provides average sample numbers for all the tests considered here. EWMA has the smallest ASN, but in many cases they are even smaller than τ. This inconsistency is due to many early false alarms.

Table 5.3: Monitoring proportion of discharged patients within four hours: Average and Maximum Sample Numbers of EWMA under $H_0 : \ p_i = 0.4, \ i \geq 1$. Number of simulations is 5000.

Statistic	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$
ASN	1391	831	553	474
max SN	10930	7880	5538	3810

Tables 5.7-5.9 provide simulation results for smaller gradation of p_A. We set $p_0 = 0.4$, $\alpha = 0.05$, $n = 260$, p_A changes from 0.401 to 0.41. We see that behavior of Tests 1.1- 1.3 is similar to behavior of these tests in the previous chapters. Test 1.3 has the highest power and Test 1.1 has the lowest power. Again, the power of EWMA is always 1 (see Table 5.8). Table 5.8 also

Table 5.4: Monitoring proportion of discharged patients within four hours: Simulated Power for Tests 1.1-1.3. The initial proportion is $p_0 = 0.4$, p_A varies from 0.5 to 0.9, the truncation point is n=260. Number of simulations is 5000.

τ	p_A	Test 1.1	Test 1.2	Test 1.3
26	0.5	1	1	1
	0.6	1	1	1
	0.7	1	1	1
	0.8	1	1	1
	0.9	1	1	1
130	0.5	1	1	1
	0.6	1	1	1
	0.7	1	1	1
	0.8	1	1	1
	0.9	1	1	1
200	0.5	1	1	1
	0.6	1	1	1
	0.7	1	1	1
	0.8	1	1	1
	0.9	1	1	1

Table 5.5: Monitoring proportion of discharged patients within four hours: Simulated Power for EWMA. The initial proportion is $p_0 = 0.4$, p_A varies from 0.5 to 0.9, $\lambda = 0.05, 0.1, 0.2, 0.3$. The algorithm is stopped after 1500 observations. Number of simulations is 5000.

τ	p_A	EWMA							
		$\lambda = 0.05$		$\lambda = 0.1$		$\lambda = 0.2$		$\lambda = 0.3$	
		Total	Before τ	Total	Before τ	Total	Before τ	Total	Before τ
26	0.5	1	0.02	1	0.02	1	0.03	1	0.04
	0.6	1	0.02	1	0.02	1	0.03	1	0.03
	0.7	1	0.02	1	0.02	1	0.03	1	0.04
	0.8	1	0.01	1	0.02	1	0.03	1	0.03
	0.9	1	0.02	1	0.02	1	0.03	1	0.03
130	0.5	1	0.08	1	0.14	1	0.19	1	0.22
	0.6	1	0.09	1	0.13	1	0.20	1	0.24
	0.7	1	0.09	1	0.13	1	0.20	1	0.23
	0.8	1	0.08	1	0.13	1	0.20	1	0.23
	0.9	1	0.09	1	0.13	1	0.20	1	0.23
200	0.5	1	0.14	1	0.20	1	0.29	1	0.35
	0.6	1	0.14	1	0.21	1	0.29	1	0.34
	0.7	1	0.14	1	0.21	1	0.28	1	0.35
	0.8	1	0.13	1	0.20	1	0.29	1	0.34
	0.9	1	0.14	1	0.21	1	0.30	1	0.32

shows a lot of early false alarms (up to 34%). The number of early false alarms

Table 5.6: Monitoring proportion of discharged patients within four hours: Average Sample Number. The initial proportion is $p_0 = 0.4$, p_A varies from 0.5 to 0.9. The truncation point is $n = 260$, EWMA algorithm is stopped after 1500 observations. Number of simulations is 5000.

τ	p_A	Test 1.1	Test 1.2	Test 1.2	EWMA (Total)			
					$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$
26	0.5	28	31	31	27	26	26	26
	0.6	27	28	28	26	26	26	26
	0.7	26	27	27	26	26	26	26
	0.8	26	27	27	26	26	26	26
	0.9	26	27	27	26	26	26	26
130	0.5	134	135	134	126	122	118	116
	0.6	131	132	132	124	122	117	115
	0.7	130	131	131	124	122	117	115
	0.8	129	131	131	125	122	118	115
	0.9	129	131	130	124	121	117	115
200	0.5	204	204	203	187	181	171	164
	0.6	199	202	201	187	180	171	166
	0.7	199	201	200	187	179	172	165
	0.8	198	200	200	186	180	170	165
	0.9	197	200	200	186	179	170	167

increases with λ and does not depend on p_A, when we compare columns *Before* τ in Tables 5.8 and 5.5. Table 5.9 shows that all average sample numbers are quite large. For small changes ($p_A \leq 0.404$), the open-ended EWMA has larger average sample numbers than the truncated tests. When a change is large, EWMA has the smallest ASN. Out of Tests 1.1-1.3, Test 1.3 has the smallest ASN unless a large change happend early in a sequence, in which case Test 1.1 has the smallest ASN.

Tables 5.10-5.12 contain simulated power and ASN when p_A is fixed at 0.4. The tables summarize power and ASN behavior for different settings of p_0. Tables 5.10 and 5.11 show that all tests detect a change in proportion in 1%. They also show that decrease in the proportion is detected by Test 1.1, Test 1.2 and EWMA. This is due to the fact, that those tests are two-sided. An adequate choice of a value of p_0 may help to detect wheather the historical value reached the desired level or not. For this purpose one-sided tests, such as Test 1.3, are a better option.

Table 5.7: Monitoring proportion of discharged patients within four hours: Simulated Power for Tests 1.1-1.3. The initial proportion is $p_0 = 0.4$, p_A varies from 0.401 to 0.41, the truncation point is $n = 260$. Number of simulations is 5000.

τ	p_A	Test 1.1	Test 1.2	Test 1.3
26	0.401	0.06	0.14	0.21
	0.402	0.18	0.42	0.55
	0.403	0.47	0.77	0.86
	0.404	0.81	0.94	0.98
	0.405	0.96	0.99	1.00
	0.406	1.00	1.00	1.00
	0.407	1.00	1.00	1.00
	0.408	1.00	1.00	1.00
	0.409	1.00	1.00	1.00
	0.410	1.00	1.00	1.00
130	0.401	0.03	0.07	0.10
	0.402	0.05	0.15	0.27
	0.403	0.09	0.28	0.49
	0.404	0.18	0.47	0.71
	0.405	0.34	0.67	0.90
	0.406	0.55	0.84	0.98
	0.407	0.73	0.93	1.00
	0.408	0.87	0.98	1.00
	0.409	0.96	0.99	1.00
	0.410	0.99	1.00	1.00
200	0.401	0.02	0.05	0.06
	0.402	0.02	0.07	0.09
	0.403	0.03	0.09	0.15
	0.404	0.04	0.13	0.22
	0.405	0.05	0.18	0.32
	0.406	0.06	0.24	0.43
	0.407	0.10	0.32	0.57
	0.408	0.14	0.41	0.71
	0.409	0.20	0.51	0.81
	0.410	0.26	0.59	0.91

Table 5.8: Monitoring proportion of discharged patients within four hours: Simulated Power for EWMA. The initial proportion is $p_0 = 0.4$, p_A varies from 0.401 to 0.41, $\lambda = 0.05, 0.1, 0.2, 0.3$. The algorithm is stopped after 1500 observations. Number of simulations is 5000.

τ	p_A	EWMA							
		$\lambda = 0.05$		$\lambda = 0.1$		$\lambda = 0.2$		$\lambda = 0.3$	
		Total	Before τ	Total	Before τ	Total	Before τ	Total	Before τ
26	0.401	0.81	0.02	0.90	0.02	0.96	0.03	0.97	0.03
	0.402	0.98	0.01	0.98	0.02	0.99	0.03	0.99	0.04
	0.403	1.00	0.01	1.00	0.02	1.00	0.03	1.00	0.04
	0.404	1.00	0.02	1.00	0.02	1.00	0.03	1.00	0.04
	0.405	1.00	0.02	1.00	0.02	1.00	0.03	1.00	0.04
	0.406	1.00	0.02	1.00	0.03	1.00	0.03	1.00	0.03
	0.407	1.00	0.02	1.00	0.02	1.00	0.03	1.00	0.04
	0.408	1.00	0.02	1.00	0.02	1.00	0.03	1.00	0.03
	0.409	1.00	0.02	1.00	0.02	1.00	0.03	1.00	0.03
	0.410	1.00	0.01	1.00	0.02	1.00	0.03	1.00	0.04
130	0.401	0.80	0.09	0.90	0.12	0.95	0.19	0.97	0.23
	0.402	0.97	0.08	0.98	0.13	0.98	0.20	0.99	0.23
	0.403	1.00	0.09	1.00	0.14	1.00	0.19	1.00	0.23
	0.404	1.00	0.08	1.00	0.13	1.00	0.20	1.00	0.23
	0.405	1.00	0.09	1.00	0.13	1.00	0.20	1.00	0.23
	0.406	1.00	0.09	1.00	0.14	1.00	0.20	1.00	0.23
	0.407	1.00	0.08	1.00	0.14	1.00	0.19	1.00	0.23
	0.408	1.00	0.09	1.00	0.13	1.00	0.19	1.00	0.23
	0.409	1.00	0.09	1.00	0.13	1.00	0.18	1.00	0.22
	0.410	1.00	0.09	1.00	0.13	1.00	0.20	1.00	0.23
200	0.401	0.81	0.13	0.90	0.21	0.95	0.29	0.97	0.33
	0.402	0.96	0.13	0.97	0.20	0.98	0.30	0.99	0.34
	0.403	1.00	0.14	1.00	0.21	1.00	0.29	1.00	0.32
	0.404	1.00	0.14	1.00	0.21	1.00	0.29	1.00	0.34
	0.405	1.00	0.14	1.00	0.22	1.00	0.29	1.00	0.34
	0.406	1.00	0.13	1.00	0.21	1.00	0.30	1.00	0.33
	0.407	1.00	0.13	1.00	0.21	1.00	0.28	1.00	0.34
	0.408	1.00	0.14	1.00	0.21	1.00	0.28	1.00	0.33
	0.409	1.00	0.13	1.00	0.21	1.00	0.28	1.00	0.34
	0.410	1.00	0.13	1.00	0.20	1.00	0.29	1.00	0.33

Table 5.9: Monitoring proportion of discharged patients within four hours: Average Sample Number. The initial proportion is $p_0 = 0.4$, p_A varies from 0.401 to 0.41. The truncation point is $n = 260$, EWMA algorithm is stopped after 1500 observations. Number of simulations is 5000.

τ	p_A	Test 1.1	Test 1.2	Test 1.3	EWMA (Total)			
					$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$
26	0.401	252	253	248	742	586	468	419
	0.402	243	234	222	423	394	346	339
	0.403	216	200	184	228	239	243	257
	0.404	174	166	152	145	157	169	183
	0.405	137	139	127	105	111	124	137
	0.406	109	120	110	81	84	95	106
	0.407	92	107	99	67	68	76	85
	0.408	80	96	89	59	58	64	72
	0.409	71	88	82	52	51	55	62
	0.410	65	82	77	49	47	49	54
130	0.401	255	257	255	760	623	479	427
	0.402	255	254	249	490	442	389	357
	0.403	253	249	240	316	309	305	295
	0.404	251	240	228	236	238	238	242
	0.405	244	230	213	198	195	198	203
	0.406	235	219	201	175	172	174	182
	0.407	225	210	192	163	159	161	165
	0.408	214	200	184	154	151	149	152
	0.409	204	192	178	149	145	144	144
	0.410	196	186	173	145	141	137	138
200	0.401	256	257	257	771	619	493	430
	0.402	256	257	257	533	475	402	383
	0.403	256	257	255	364	350	333	321
	0.404	256	256	254	290	282	277	272
	0.405	255	255	252	254	247	244	244
	0.406	255	254	251	235	226	219	222
	0.407	254	252	247	224	213	208	205
	0.408	254	251	245	215	206	200	197
	0.409	253	249	242	210	199	193	190
	0.410	252	247	238	208	197	188	185

Table 5.10: Monitoring proportion of discharged patients within four hours: Simulated Power for Tests 1.1-1.3. The initial proportion values are $p_0 = 0.3, 0.39, 0.399, 0.4, 0.401, 0.41, 0.5$, the alternative parameter p_A is fixed at 0.4, the truncation point is $n = 260$. Number of simulations is 5000.

τ	p_0	Test 1.1	Test 1.2	Test 1.3
26	0.300	1.00	1.00	1.00
	0.390	1.00	1.00	1.00
	0.399	0.05	0.13	0.22
	0.400	0.02	0.04	0.04
	0.401	0.06	0.14	0.00
	0.410	1.00	1.00	0.00
	0.500	1.00	1.00	0.00
130	0.300	1.00	1.00	1.00
	0.390	0.98	1.00	1.00
	0.399	0.03	0.06	0.12
	0.400	0.03	0.05	0.04
	0.401	0.03	0.07	0.02
	0.410	0.98	1.00	0.00
	0.500	1.00	1.00	0.00
200	0.300	1.00	1.00	1.00
	0.390	0.27	0.60	0.91
	0.399	0.02	0.05	0.07
	0.400	0.03	0.04	0.05
	0.401	0.03	0.05	0.03
	0.410	0.26	0.59	0.02
	0.500	1.00	1.00	0.02

Table 5.11: Monitoring proportion of discharged patients within four hours: Simulated Power for EWMA. The initial proportion values are $p_0 = 0.3, 0.39, 0.399, 0.4, 0.401, 0.41, 0.5$, the alternative parameter p_A is fixed at 0.4, $\lambda = 0.05, 0.1, 0.2, 0.3$. The algorithm is stopped after 1500 observations. Number of simulations is 5000.

τ	p_0	EWMA							
		$\lambda = 0.05$		$\lambda = 0.1$		$\lambda = 0.2$		$\lambda = 0.3$	
		Total	Before τ	Total	Before τ	Total	Before τ	Total	Before τ
26	0.300	1.00	0.02	1.00	0.02	1.00	0.03	1.00	0.04
	0.390	1.00	0.02	1.00	0.02	1.00	0.03	1.00	0.03
	0.399	0.82	0.02	0.90	0.02	0.95	0.03	0.97	0.03
	0.400	0.67	0.02	0.84	0.02	0.93	0.03	0.96	0.03
	0.401	0.81	0.01	0.90	0.02	0.95	0.04	0.97	0.04
	0.410	1.00	0.02	1.00	0.02	1.00	0.03	1.00	0.03
	0.500	1.00	0.02	1.00	0.02	1.00	0.03	1.00	0.04
130	0.300	1.00	0.09	1.00	0.14	1.00	0.20	1.00	0.22
	0.390	1.00	0.08	1.00	0.14	1.00	0.20	1.00	0.23
	0.399	0.81	0.09	0.90	0.14	0.96	0.20	0.97	0.23
	0.400	0.68	0.08	0.83	0.14	0.93	0.20	0.96	0.23
	0.401	0.80	0.09	0.89	0.14	0.95	0.20	0.97	0.23
	0.410	1.00	0.09	1.00	0.13	1.00	0.18	1.00	0.24
	0.500	1.00	0.08	1.00	0.14	1.00	0.20	1.00	0.23
200	0.300	1.00	0.13	1.00	0.20	1.00	0.29	1.00	0.32
	0.390	1.00	0.13	1.00	0.21	1.00	0.28	1.00	0.34
	0.399	0.80	0.13	0.90	0.21	0.95	0.29	0.97	0.34
	0.400	0.66	0.13	0.85	0.21	0.93	0.29	0.96	0.33
	0.401	0.80	0.13	0.88	0.20	0.95	0.29	0.97	0.34
	0.410	1.00	0.13	1.00	0.20	1.00	0.29	1.00	0.33
	0.500	1.00	0.14	1.00	0.21	1.00	0.28	1.00	0.33

Table 5.12: Monitoring proportion of discharged patients within four hours: Average Sample Number. The initial proportion values are $p_0 = 0.3, 0.39, 0.399, 0.4, 0.401, 0.41, 0.5$, the alternative parameter p_A is fixed at 0.4. The truncation point is $n = 260$, EWMA algorithm is stopped after 1500 observations. Number of simulations is 5000.

τ	p_0	Test 1.1	Test 1.2	Test 1.3	EWMA (Total)			
					$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$
26	0.300	28	31	30	27	26	26	26
	0.390	65	82	77	48	47	49	53
	0.399	252	252	248	723	593	470	412
	0.400	256	257	258	916	701	531	456
	0.401	253	252	260	740	600	472	429
	0.410	65	83	260	49	47	50	54
	0.500	28	31	260	27	26	26	26
130	0.300	133	135	133	125	122	118	116
	0.390	195	186	173	146	140	137	137
	0.399	255	257	255	758	608	474	424
	0.400	256	257	258	897	708	526	448
	0.401	255	257	259	768	615	487	433
	0.410	196	186	260	145	141	139	138
	0.500	133	135	260	126	122	117	116
200	0.300	203	204	202	187	181	171	166
	0.390	252	247	238	207	196	189	185
	0.399	256	257	257	782	623	492	426
	0.400	255	258	258	912	692	518	459
	0.401	255	257	258	781	641	494	430
	0.410	252	247	258	207	197	188	187
	0.500	204	205	258	187	179	171	166

Bibliography

[1] Alberta Health and Wellness *Becoming the Best: Alberta's 5-Year Health Action Plan 2010-2015*, Alberta Health Services, 2010. Available from http://www.health.alberta.ca/documents/Becoming-the-Best-2010.pdf.

[2] Food and Drug Administration, *E9: Statistical Principals for Clinical Trials*, Federal Register, **63(179)** (1998), 49583-49598.

[3] Health Canada, *Guidance for Industry: Statistical Principals for Clinical Trials: ICH Topic E9*, 2003. Available from http://www.hc-sc.gc.ca/dhp-mps/prodpharma/applic-demande/guide-ld/ich/efficac/e9-eng.php.

[4] S. Ellenberg, T. Fleming and D. DeMets, *Data Monitoring Committees in Clinical Trials: A Practical Perspective*, John Wiley & Sons, England, 2002.

[5] P. Armitage, *Interim analysis in clinical trials,* Multiple Comparisons, Selection, and Applications in Biometry, (ed. by F. M. Hoppe), Marcel Dekker, New York, 1993, 391402.

[6] W. A. Shewhart, *Economic control of quality of manufactured product*, D. Van Nostrand, New York, 1931.

[7] A. Wald, *Sequential tests of statistical hypotheses*, Ann Math Statist **6** (1945), 117-186.

[8] G. A. Barnard, *Sequential tests in industrial statistics (with discussion)*, J R Statist Soc **8(suppl.)** (1946), 1-26.

[9] A. Wald, *Sequential Analysis*, Wiley, New York, 1947.

[10] D. C. Montgomery, *Introduction to Statistical Quality Control*, 6th edn, John Wiley & Sons, New York, 2009.

[11] E. S. Page, *Continuous inspection schemes*, Biometrika **41** (1954), 100114.

[12] G. A. Barnard, *Control charts and stochastic processes*, J R Statist Soc Series B **21** (1959), 239-271.

[13] G. Lorden, *Procedures for reacting to a change in distribution*, Ann Math Statist **42** (1971), 1897-1908.

[14] G. V. Moustakides, *Optimal stopping times for detecting changes in distributions*, Annals of Statistics **14(4)** (1986), 13791387.

[15] S. W. Roberts, *Control chart tests based on geometric moving averages*, Technometrics **1** (1959), 239-250.

[16] S. V. Crowder, *A simple method for studying run length distributions of exponentially weighted moving average charts*, Technometrics **29** (1987), 401-407.

[17] J. M. Lucas and M. S. Saccucci, *Exponentially weighted moving average control schemes: Properties and enhancements (with discussion)*, Technometrics **32** (1990), 1-29.

[18] G. Box, G. Jenkins, G. Reinsel, *Time Series Analysis: Forecasting and Control*, 3rd ed, Prentice Hall, Englewood Cliffs, 1994.

[19] D. A. Darling, P. Erdős, *A limit theorem for the maximum of normalized sums of independent random variables*, Duke Mathematical Journal **23** (1956), 143155.

[20] L. J. Vostrikova, *Detection of a "disorder" in a Wiener process*, Theory of Probability and its Applications **26** (1981), 356362.

[21] E. Gombay, *Parametric sequential tests in the presence of nuisance parameters*, Theory of Stochastic Processes **8(24)** (2002), 107118.

[22] M. S. Bartlett, *The large-sample theory of sequential tests*, Proc Camb Phil Soc **42** (1946), 239-244.

[23] D. R. Cox, *Large sample sequential tests for composite hypotheses,* Sankhya **25** (1963), 5-12.

[24] R. R. Bahadur, *An optimal property of the likelihood ratio statistic*, Proc Fifth Berkeley Symp Math Statist **1** (1966), 13-26.

[25] L. D. Brown, *Non-local asymptotic optimality of appropriate likelihood ratio tests*, Ann Math Statist **42** (1971), 1206-1240.

[26] E. Gombay, *Sequential testing of composite hypothesis*, Limit Theorems in Probability and Statistics, Vol. II, (ed. by I. Berkes, E. Csaki and M. Csorgo), Budapest, 2002, 107125.

[27] G. A. Barnard, *Review of A. Wald's sequential analysis*, J Amer Statist Ass **42** (1947), 658-664.

[28] S. Rushton, *On a sequential t-test*, Biometrika **37** (1950), 326-333.

[29] S. Rushton, *On a two-sided sequential t-test*, Biometrika **39** (1952), 302-308.

[30] E. Gombay, *Sequential change-point detection and estimation*, Sequential Analysis **22** (2003), 203222.

[31] *Shipman Inquiry: The First Report*, London, UK: The Stationery Office, 2002. Available from http://www.the-shipman-inquiry.org.uk/reports.asp.

[32] BRI Inquiry Panel, *Learning from Bristol: The Report of the Public Inquiry into Children's Heart Surgery at the Royal Infirmary 1984-1995*, London, UK: The Stationery Office, 2001. Available from http://www.bristol-inquiry.org.uk/final_report.

[33] D. Spiegelhalter, O. Grigg, R. Kinsman, T. Treasure, *Risk-adjusted sequential probability ratio tests: applications to Bristol, Shipman and adult cardiac surgery*, International Journal for Quality in Health Care **15(1)** (2003), 7-13.

[34] S. H. Steiner, R. J. Cook, V. T. Farewell, T. Treasure, *Monitoring surgical performance using risk-adjusted cumulative sum charts*, Biostatistics **1(4)** (2000), 441-452.

[35] J. Lovegrove, O. Valencia, T. Treasure, C. Sherlaw-Johnson, S. Gallivan, *Monitoring the results of cardiac surgery by variable life-adjusted display*, Lancet **350** (1997), 1128-1130.

[36] J. Lovegrove, C. Sherlaw-Johnson, O. Valencia, T. Treasure, S. Gallivan, *Monitoring the performance of cardiac surgeons*, Journal of the Operational Research Society **50(7)** (1999), 684-689.

[37] E. Gombay, A. A. Hussein, S. H. Steiner, *Monitoring binary outcomes using risk-adjusted: a comparative study* Stat Med **30(23)** (2011), 2815-2826.

[38] V. Parsonnet, D. Dean, A. Bernstein, *A method Of uniform stratification of risk for evaluating the results of surgery in acquired adult heart-disease*, Circulation **79(6)** (1989), 3-12.

[39] E. A. Parent, *Sequential ranking procedures*, PhD Thesis, Stanford University, 1965.

[40] P. Bogowicz, E. Gombay, G. Heo, *Nonparametric sequential monitoring of longitudinal trials*, Stat Med **29(24)** (2010), 2469-2479.

[41] Alberta Health and Wellness *Weekly Emergency Department Length of Stay Summary*, Alberta Health Services, 2010-present. Available from http://www.albertahealthservices.ca/Data/ahs-data-er-wait-times-edmonton.pdf.

[42] O. Grigg, D. Spiegelhalter, *A simple risk-adjusted exponentially weighted moving average*, J Am Stat Assoc **102** (2007), 140-152.

[43] D. A. Cook, M. Coory, R. A. Webster, *Exponentially weighted moving average charts to compare observed and expected values for monitoring risk-adjusted hospital indicators*, BMJ Qual Saf **20(6)** (2011), 469-474.

Appendix A

Asymptotics

Here we provide some theoretical background to the truncated tests described in the thesis. Let

$$a(x) = (2\log x)^{\frac{1}{2}}, \quad b(x) = 2\log x + \frac{1}{2}\log\log x - \frac{1}{2}\log\pi$$

and $W(t)$ denote a standard Brownian motion. Under certain conditions the test statistics have the following asymptotic distributions under the null hypothesis of no change.

- The asymptotic distributions of the one-sided and two-sided Test 1.1 statistics were calculated by Darling and Erdős (1956) and are given by

$$\lim_{n\to\infty} P\left(a(\log n)\max_{1<k\leq n}\frac{1}{\sqrt{k}}S_k \leq u + b(\log n)\right) = \exp(-e^{-u})$$

and

$$\lim_{n\to\infty} P\left(a(\log n)\max_{1<k\leq n}\frac{1}{\sqrt{k}}|S_k| \leq u + b(\log n)\right) = \exp(-2e^{-u}),$$

for $-\infty < u < \infty$. The two-sided test statistic $\max_{1<k\leq n} 1/\sqrt{k}|S_k|$ can be better approximated by a diffusion process

$$\begin{aligned} P\left(\max_{1<k\leq n}\tfrac{1}{\sqrt{k}}|S_k| > u\right) &\cong P\left(\sup_{1\leq t\leq n} W^{\frac{1}{2}}(t) > u\right) \\ &\cong P\left(\sup_{1\leq t\leq n}|U(\log t)| > u\right), \end{aligned}$$

where $U(t)$ is a stationary diffusion process. Vostrikova (1981) showed that for all $N > 0$

$$P\left(\sup_{0\leq t\leq N}|U(t)| > u\right) = \frac{u\exp(-u^2/2)}{\sqrt{2\pi}}\left(N - N\frac{1}{u^2} + \frac{4}{u^2} + O(\frac{1}{u^4})\right)$$

as $u \to \infty$. Hence the improved critical value of the two-sided Test 1.1 is obtained using

$$P\left(\max_{1<k\leq n} \frac{1}{\sqrt{k}}|S_k| > u\right) \cong \frac{u\exp(-u^2/2)}{\sqrt{2\pi}}\left(\log n(1-\frac{1}{u^2}) + \frac{4}{u^2} + O(\frac{1}{u^4})\right).$$

- A critical value for the Test 1.2 can be obtained from

$$P\left(\max_{1<k\leq n} \frac{1}{\sqrt{n}}S_k > u\right) \cong P\left(\sup_{0\leq t\leq 1} W(t) > u\right) = 2(1-\Phi(u)),$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. The two-sided test statistic have the following tail probability

$$\begin{aligned} P\left(\max_{1<k\leq n} \tfrac{1}{\sqrt{n}}|S_k| > u\right) &\cong P\left(\sup_{0\leq t\leq 1} |W(t)| > u\right) \\ &= 1 - \tfrac{4}{\pi}\sum_{m=0}^{\infty} \tfrac{(-1)^m}{2m+1}\exp\left(-\tfrac{\pi^2(2m+1)^2}{8u^2}\right). \end{aligned} \tag{A.1}$$

- The Test 1.3 statistic can also be approximated by the supremum of a Brownian motion. This follows from the fact that

$$\max_{0\leq j<k\leq n} \frac{1}{\sqrt{n}}(S_k - S_j) \to^D \sup_{0\leq s<t\leq 1}(W(t) - W(s)) =^D \sup_{0\leq t\leq 1}|W(t)|.$$

Therefore a critical value for Test 1.3 can be obtained from the last equality in (A.1).

Appendix B

R Scripts

A function that finds α for Test 1, given critical value h. It is used to find an approximate critical value using Vostrikova's improvement.

```
##########################
### Test 1 critic val ####
##########################

f=function(h,n)
{
 return(exp(-h^2/2)*h/sqrt(2*pi)*(log(n)*(1-1/h^2)+4/h^2))
}
```

Script for Tables 2.1 and 2.2.

```
############################
### Test 1, 2-sided #######
############################

test1=function(n,s,c1)
{
t=abs(s)/sqrt(1:n)
k=which.min(t[11:n]<c1)
if(k==1 && t[11]<c1)  return(c(n,0)) else return(c(k,1))
}

############################
### Test 2, 2-sided #######
############################

test2=function(n,s)
{
t=abs(s)/sqrt(n)
k=which.min(t[11:n]<2.24)
if(k==1 && t[11]<2.24) return(c(n,0)) else return(c(k,1))
}

############################
### Test 3 ################
############################

test3=function(n,y,s2k)
{
d=rep(0,n)
for(k in 1:10)
{
 d[1:k]=d[1:k]+y[k]
}
t=0
k=11
while(t<2.24 && k<=n)
{
 d[1:k]=d[1:k]+y[k]
 t=max(d[1:k]/sqrt(s2k[k])/sqrt(n))
 k=k+1
}
if(t<2.24) return(c(n,0)) else return(c(k-1,1))
```

```
}


############################
### Saving results ########
############################

sink("outputnormal.txt")
set.seed(9600)

xx=format(round(c(0.0,0.2,0.4,0.6),1),nsmall=1)
xx=rep(xx,3)
y=c(50,rep(999,3),100,rep(999,3),200,rep(999,3))
z=c(100,rep(999,3),200,rep(999,3),400,rep(999,3))
xx=cbind(z,y,xx)

power=NULL
asn=NULL
for (n in c(100,200,400))
{
if (n==100) {c1=3.07} else {if(n==200) {c1=3.12} else c1=3.16}
for (tau in c(0.5))
{
 for (i in c(0.0,0.2,0.4,0.6))
 {
  a1=0
  asn1=0
  a2=0
  asn2=0
  a3=0
  asn3=0

  for (m in 1:5000)
  {
   y=c(rnorm(round(tau*n)-1,0,1),rnorm(round(tau*n)+1,i,1))
   st=cumsum(y)
   s2=cumsum(y^2)
   s2k=s2/1:n
   s=st/sqrt(s2k)

   test=test1(n,s,c1)
   a1=a1+test[2]
   asn1=asn1+test[1]

   test=test2(n,s)
```

```
   a2=a2+test[2]
   asn2=asn2+test[1]

   test=test3(n,y,s2k)
   a3=a3+test[2]
   asn3=asn3+test[1]
  }
  power=rbind(power,c(a1/m,a2/m,a3/m))
  asn=rbind(asn,c(asn1/m,asn2/m,asn3/m))
 }
}
}
power=cbind(xx,format(round(power,2),nsmall=2))
asn=cbind(xx,format(round(asn,2),nsmall=2))

cat("Simulated Power")
cat("\n")
cat("\n")
write(t(power),"",ncolumns=6,sep=" & ")
cat("\n")
cat("ASN")
cat("\n")
cat("\n")
write(t(asn),"",ncolumns=6,sep=" & ")

sink()
```

Script for Tables 3.1 and 3.2.

```
############################
### Test 1, 2-sided #######
############################

test1=function(n,s,c1)
{
t=abs(s)/sqrt(1:n)
k=which.min(t[11:n]<c1)
if(k==1 && t[11]<c1)  return(c(n,0)) else return(c(k,1))
}

############################
### Test 2, 2-sided #######
############################

test2=function(n,s)
{
t=abs(s)/sqrt(n)
k=which.min(t[11:n]<2.24)
if(k==1 && t[11]<2.24) return(c(n,0)) else return(c(k,1))
}

############################
### Test 3 ################
############################

test3=function(n,y,p)
{
d=rep(0,n)
for(k in 1:10)
{
 d[1:k]=d[1:k]+st[k]
}
t=0
k=11
while(t<2.24 && k<=n)
{
 d[1:k]=d[1:k]+st[k]
 t=max(d[1:k])/sqrt(n)
 k=k+1
}
if(t<2.24) return(c(n,0)) else return(c(k-1,1))
```

```
}

############################
### Saving results ########
############################

library(Rlab)
sink("outputbinary.txt")
set.seed(9600)

xx=format(round(c(0.07,0.09,0.11,0.13),2),nsmall=2)
xx=rep(xx,3)
y=c(1,rep(999,3),500,rep(999,3),2500,rep(999,3))
xx=cbind(y,xx)

power=NULL
asn=NULL
for (n in c(5000))
{
c1=3.3*0^(9600-n)+3.28*0^(n-5000)
for (tau in c(1/n,0.1,0.5))
{
 for (i in c(0.07,0.09,0.11,0.13))
 {
  a1=0
  asn1=0
  a2=0
  asn2=0
  a3=0
  asn3=0

  for (m in 1:5000)
  {
   p=rep(0.07,n)
   ptrue=p
   ptrue[round(tau*n):n]=i
   y=rbern(n,ptrue)
   st=(y-p)/sqrt(p*(1-p))
   s=cumsum(st)

   test=test1(n,s,c1)
   a1=a1+test[2]
   asn1=asn1+test[1]

   test=test2(n,s)
```

```
   a2=a2+test[2]
   asn2=asn2+test[1]

   test=test3(n,y,p)
   a3=a3+test[2]
   asn3=asn3+test[1]
  }
  power=rbind(power,c(a1/m,a2/m,a3/m))
  asn=rbind(asn,c(asn1/m,asn2/m,asn3/m))
 }
}
}
power=cbind(xx,format(round(power,2),nsmall=2))
asn=cbind(xx,format(round(asn,2),nsmall=2))

cat("Simulated Power")
cat("\n")
cat("\n")
write(t(power),"",ncolumns=5,sep=" & ")
cat("\n")
cat("ASN")
cat("\n")
cat("\n")
write(t(asn),"",ncolumns=5,sep=" & ")

sink()
```

Script for Tables 3.3-3.6.

```
############################
### Test 1, 2-sided #######
############################

test1=function(n,s,c1)
{
t=abs(s)/sqrt(1:n)
k=which.min(t[11:n]<c1)
if(k==1 && t[11]<c1)  return(c(n,0)) else return(c(k,1))
}

############################
### Test 2, 2-sided #######
############################

test2=function(n,s)
{
t=abs(s)/sqrt(n)
k=which.min(t[11:n]<2.24)
if(k==1 && t[11]<2.24) return(c(n,0)) else return(c(k,1))
}

############################
### Test 3 ################
############################

test3=function(n,y,p)
{
d=rep(0,n)
for(k in 1:10)
{
 d[1:k]=d[1:k]+st[k]
}
t=0
k=11
while(t<2.24 && k<=n)
{
 d[1:k]=d[1:k]+st[k]
 t=max(d[1:k])/sqrt(n)
 k=k+1
}
if(t<2.24) return(c(n,0)) else return(c(k-1,1))
```

```
}

############################
### CUSUM ##################
############################

cusum=function(n,wk,h)
{
xt=0
k=1
 while(xt<h && k<=n)
 {
  xt=max(c(0,xt+wk[k]))
  k=k+1
 }
if(xt<h) return(c(n,0)) else return(c(k-1,1))
}

############################
### Saving results ########
############################

library(Rlab)
sink("SurgeonH0.txt")
set.seed(9600)

power=NULL
asn=NULL
asncusum=NULL
for (n in c(5000,9600))
{
c1=3.3*0^(9600-n)+3.28*0^(n-5000)
  a1=0
  asn1=0
  a2=0
  asn2=0
  a3=0
  asn3=0
  a4=0
  asn4=c()

  for (m in 1:5000)
  {
   x=rpois(n,13)
   logitp=-3.68+0.077*x
```

```
   p=exp(logitp)/(1+exp(logitp))
   y=rbern(n,p)
   st=(y-p)/sqrt(p*(1-p))
   s=cumsum(st)
   wk=log(2^y/(1-p+2*p))

   test=test1(n,s,c1)
   a1=a1+test[2]
   asn1=asn1+test[1]

   test=test2(n,s)
   a2=a2+test[2]
   asn2=asn2+test[1]

   test=test3(n,y,p)
   a3=a3+test[2]
   asn3=asn3+test[1]

   test=cusum(n,wk,4.5)
   a4=a4+test[2]
   asn4=c(asn4,test[1])
  }
  power=rbind(power,c(a4/m,a1/m,a2/m,a3/m))
  asncusum=rbind(asncusum,asn4)
  asn=rbind(asn,c(sum(asn4)/m,asn1/m,asn2/m,asn3/m))
}
power=cbind(c(5000,9600),format(round(power,4),nsmall=4))
asn=cbind(c(5000,9600),format(round(asn,2),nsmall=2))

cat("Simulated Type I Error")
cat("\n")
cat("\n")
write(t(power),"",ncolumns=5,sep=" & ")
cat("\n")
cat("ASN")
cat("\n")
cat("\n")
write(t(asn),"",ncolumns=5,sep=" & ")

sink()

sink("Surgeon.txt")
set.seed(9600)

xx=format(round(c(1.25,1.50,1.75,2.00),2),nsmall=2)
```

```
xx=rep(xx,6)
y=c(1,rep(999,3),500,rep(999,3),2500,rep(999,3))
y=c(y,1,rep(999,3),960,rep(999,3),4800,rep(999,3))
z=c(5000,rep(999,11),9600,rep(999,11))
xx=cbind(z,y,xx)

power=NULL
asn=NULL
for (n in c(5000,9600))
{
c1=3.3*0^(9600-n)+3.28*0^(n-5000)
for (tau in c(1/n,0.1,0.5))
{
 for (i in c(1.25,1.50,1.75,2.00))
 {
  a1=0
  asn1=0
  ab1=0
  asnstopped1=0
  a2=0
  asn2=0
  ab2=0
  asnstopped2=0
  a3=0
  asn3=0
  ab3=0
  asnstopped3=0

  for (m in 1:5000)
  {
   x=rpois(n,13)
   logitp=-3.68+0.077*x
   p=exp(logitp)/(1+exp(logitp))
   ptrue=p
   ptrue[round(tau*n):n]=i*p[round(tau*n):n]/
   (1-p[round(tau*n):n]+i*p[round(tau*n):n])
   y=rbern(n,ptrue)
   st=(y-p)/sqrt(p*(1-p))
   s=cumsum(st)

   test=test1(n,s,c1)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
```

```
    ab1=ab1+1
   }
   asnstopped1=asnstopped1+test[2]*test[1]

   test=test2(n,s)
   a2=a2+test[2]
   asn2=asn2+test[1]
   if (test[1]<tau*n)
   {
    ab2=ab2+1
   }
   asnstopped2=asnstopped2+test[2]*test[1]

   test=test3(n,y,p)
   a3=a3+test[2]
   asn3=asn3+test[1]
   if (test[1]<tau*n)
   {
    ab3=ab3+1
   }
   asnstopped3=asnstopped3+test[2]*test[1]
  }
  power=rbind(power,c(a1/m,ab1/m,a2/m,ab2/m,a3/m,ab3/m))
  asn=rbind(asn,c(asn1/m,asnstopped1/a1,asn2/m,asnstopped2/a2,
      asn3/m,asnstopped3/a3))
 }
}
}
power=cbind(xx,format(round(power,2),nsmall=2))
asn=cbind(xx,format(round(asn),nsmall=0))

cat("Simulated Power")
cat("\n")
cat("\n")
write(t(power),"",ncolumns=9,sep=" & ")
cat("\n")
cat("ASN")
cat("\n")
cat("\n")
write(t(asn),"",ncolumns=9,sep=" & ")

sink()

sink("CUSUMresults.txt")
set.seed(9600)
```

```
xx=format(round(c(1.25,1.50,1.75,2.00),2),nsmall=2)
xx=rep(xx,5)
y=c(1,rep(999,3),500,rep(999,3),960,rep(999,3),
  2500,rep(999,3),4800,rep(999,3))
xx=cbind(y,xx)
result=NULL
n=9600
for (tau in c(1,500,960,2500,4800))
{
 for (i in c(1.25,1.50,1.75,2.00))
 {
  a4=0
  asn4=0
  ab4=0
  asncond=0
  for (m in 1:5000)
  {
   x=rpois(n,13)
   logitp=-3.68+0.077*x
   p=exp(logitp)/(1+exp(logitp))
   ptrue=p
   ptrue[tau:n]=i*p[tau:n]/(1-p[tau:n]
                          +i*p[tau:n])
   y=rbern(n,ptrue)
   wk=log(2^y/(1-p+2*p))

   test=cusum(n,wk,4.5)
   a4=a4+test[2]
   asn4=asn4+test[1]
   if (test[1]>=tau)
   {
    asncond=asncond+test[1]
    ab4=ab4+1
   }
  }
  result=rbind(result,c(a4/m,(m-ab4)/m,asn4/m,asncond/ab4))
 }
}
result=cbind(xx,format(round(result[,1:2],2),nsmall=2),
format(round(result[,3:4],0),nsmall=0))
cat("CUSUM results")
cat("\n")
cat("\n")
write(t(result),"",ncolumns=6,sep=" & ")
sink()
```

Application of tests to the cardiac surgery data.

```
a=read.csv("data.csv",header=T)
a=a[2087:6685,]

test1=function(n,s,c1)
{
 t=abs(s)/sqrt(1:length(s))
 k=which.min(t[11:length(s)]<c1)
 if(k==1 && t[11]<c1)  return(c(n,0)) else return(c(k,1))
}

test2=function(n,s)
{
t=abs(s)/sqrt(n)
k=which.min(t[11:length(s)]<2.24)
if(k==1 && t[11]<2.24) return(c(n,0)) else return(c(k,1))
}

test3=function(n,y,p)
{
d=rep(0,n)
for(k in 1:10)
{
 d[1:k]=d[1:k]+st[k]
}
t=0
k=11
while(t<2.24 && k<=length(y))
{
 d[1:k]=d[1:k]+st[k]
 t=max(d[1:k])/sqrt(n)
 k=k+1
}
if(t<2.24) return(c(n,0)) else return(c(k-1,1))
}

cusum=function(n,wk,h)
{
xt=0
k=1
 while(xt<h && k<=length(wk))
 {
  xt=max(c(0,xt+wk[k]))
```

```
  k=k+1
 }
if(xt<h) return(c(n,0)) else return(c(k-1,1))
}


for(i in 1:7)
{
 x=a[a[,2]==i,4]
 logitp=-3.68+0.077*x
 p=exp(logitp)/(1+exp(logitp))
 y=a[a[,2]==i,3]
 st=(y-p)/sqrt(p*(1-p))
 s=cumsum(st)
 wk=log(2^y/(1-p+2*p))

 print(c(test1(5000,s,3.28),test2(5000,s),
 test3(5000,y,p),cusum(5000,wk,4.5)))
}
```

Script for Tables 4.1 and 4.2.

```
#the table with J_i=50 is obtained by the same script with n1=50

###############
### Test 1 ###
###############

test1=function(n,k,z,s,c1)
{
 nt=0
 ska=0
 for(t in 1:10)
 {
  if(k[t]==1)
  {
   nt=nt+1
   ska=ska+s[t]
  }
 }
 t=11
 test=0
 while(test<c1 && t<=n)
 {
  if(k[t]==1)
  {
   sd2=(t-1)/t*sum(cov(z[1:t,]))
   nt=nt+1
   ska=ska+s[t]
   st=ska/sqrt(sd2)
   test=abs(st/sqrt(nt))
  }
  t=t+1
 }
 if(test<c1)  return(c(n,0)) else return(c(t-1,1))
}

###############
### Test 2 ###
###############

test2=function(n,k,z,s,c2)
{
 ska=0
```

```
 for(t in 1:10)
 {
  if(k[t]==1)
  {
   ska=ska+s[t]
  }
 }
 t=11
 test=0
 while(test<c2 && t<=n)
 {
  if(k[t]==1)
  {
   sd2=(t-1)/t*sum(cov(z[1:t,]))
   ska=ska+s[t]
   st=ska/sqrt(sd2)
   test=abs(st/sqrt(n1))
  }
  t=t+1
 }
 if(test<c2)  return(c(n,0)) else return(c(t-1,1))
}

###############
### Test 3 ###
###############

test3=function(n,k,s,c3)
{
 ska=0
 for(t in 1:10)
 {
  if(k[t]==1)
  {
   ska=ska+sign(s[t])
  }
 }
 t=11
 while(abs(ska)<c3 && t<=n)
 {
  if(k[t]==1)
  {
   ska=ska+sign(s[t])
  }
  t=t+1
```

```
 }
 if(abs(ska)<c3)  return(c(n,0)) else return(c(t-1,1))
}

######################
### Saving results ###
######################
sink("longit20.txt")
set.seed(9600)

n1=20
c1=2.93
c2=2.24
c3=10
n=2*n1

#creating correlated data
times=1:3
rho=0.5
sigma=1

H=abs(outer(times, times, "-"))
V=sigma*rho^H
p=nrow(V)
V[cbind(1:p, 1:p)]=V[cbind(1:p, 1:p)]*sigma
s=svd(V)
D=diag(sqrt(s$d))
om=s$u %*% D %*% t(s$v)

power=NULL
asn=NULL
ll=5000

for (mu in c(1,1.2,1.4,1.6,1.8,2))
{
 a1=0
 asn1=0
 a2=0
 asn2=0
 a3=0
 asn3=0

 for(l in 1:ll)
 {
  yy=NULL
```

```
for(k in 1:2)
{
 for(j in 1:n1)
 {
  yy=rbind(yy,c(0,0,0))
 }
}

for(k in 1:2)
{
 for(j in 1:n1)
 {
  w=rexp(3)
  yy[(k-1)*n1+j,]=om %*% w+rep(mu^(k-1),3)
 }
}

z=NULL
ind=sample(1:n)
k=c(rep(1,n1),rep(2,n1))
a=c(1:n1,1:n1)
k=k[order=ind]
for(i in 1:3)
{
 y=yy[,i]
 y=y[order=ind]
 r=c()
 for(j in 1:n)
 {
  r[j]=sum(y[1:j]<=y[j])
 }
 m=1:n
 zz=rep(0,n)
 zz[-1]=(r[-1]-(m[-1]+1)/2)/sqrt((m[-1]^2-1)/12)
 z=cbind(z,zz)
}
s=apply(z,1,sum)

test=test1(n,k,z,s,c1)
a1=a1+test[2]
asn1=asn1+test[1]

test=test2(n,k,z,s,c2)
a2=a2+test[2]
asn2=asn2+test[1]
```

```
  test=test3(n,k,s,c3)
  a3=a3+test[2]
  asn3=asn3+test[1]
 }
 power=rbind(power,c(a1/ll,a2/ll,a3/ll))
 asn=rbind(asn,c(asn1/ll,asn2/ll,asn3/ll))
}

power=cbind(format(round(c(1,1.2,1.4,1.6,1.8,2),1),
      nsmall=1),format(round(power,2),nsmall=2))
asn=cbind(format(round(c(1,1.2,1.4,1.6,1.8,2),1),
    nsmall=1),format(round(asn,2),nsmall=2))

cat("Simulated Power")
cat("\n")
cat("\n")
write(t(power),"",ncolumns=4,sep=" & ")
cat("\n")
cat("ASN")
cat("\n")
cat("\n")
write(t(asn),"",ncolumns=4,sep=" & ")

sink()
```

Script for Tables 5.1 and 5.2.

```
############################
### Test 1, 2-sided #######
############################

test1=function(n,p0,p,np,c1)
{
 d=0
 for (k in 1:10)
 {
  d=d+(p[k]-p0)*sqrt(np[k]/p0/(1-p0))
 }
 t=abs(d)/sqrt(k)
 k=11
 while (t<c1 && k<=n)
 {
  d=d+(p[k]-p0)*sqrt(np[k]/p0/(1-p0))
  t=abs(d)/sqrt(k)
  k=k+1
 }
 if(t<c1) return(c(n,0)) else return(c(k-1,1))
}
############################
### Test 2, 2-sided #######
############################

test2=function(n,p0,p,np)
{
 d=0
 for (k in 1:10)
 {
  d=d+(p[k]-p0)*sqrt(np[k]/p0/(1-p0))
 }
 t=0
 k=11
 while (t<2.24 && k<=n)
 {
  d=d+(p[k]-p0)*sqrt(np[k]/p0/(1-p0))
  t=abs(d)/sqrt(n)
  k=k+1
 }
 if(t<2.24) return(c(n,0)) else return(c(k-1,1))
}
```

```
############################
### Test 3 ################
############################

test3=function(n,p0,p,np)
{
 d=rep(0,n)
 for(k in 1:10)
 {
  d[1:k]=d[1:k]+(p[k]-p0)*sqrt(np[k]/p0/(1-p0))
 }
 t=0
 k=11
 while(t<2.24 && k<=n)
 {
  d[1:k]=d[1:k]+(p[k]-p0)*sqrt(np[k]/p0/(1-p0))
  t=max(d[1:k])/sqrt(n)
  k=k+1
 }
 if(t<2.24) return(c(n,0)) else return(c(k-1,1))
}
############################
### EWMA for not iid ######
############################
ewma=function(n,p0,y,np,np0,L,lambda)
{
  s=lambda*y[1]+(1-lambda)*p0*np0
  t=np[1]
  h=L*lambda*sqrt(p0*(1-p0)*t)
  es=lambda*p0*np[1]+(1-lambda)*p0*np0
  for(k in 2:10)
  {
   s=lambda*y[k]+(1-lambda)*s
   es=lambda*p0*np[k]+(1-lambda)*es
   t=(1-lambda)^2*t+np[k]
   h=L*lambda*sqrt(p0*(1-p0)*t)
  }
  k=11
  while (abs(s-es)<h && k<=n)
  {
   s=lambda*y[k]+(1-lambda)*s
   es=lambda*p0*np[k]+(1-lambda)*es
   t=(1-lambda)^2*t+np[k]
   h=L*lambda*sqrt(p0*(1-p0)*t)
   k=k+1
```

```
  }
 if(abs(s-es)<h) return(c(n,0)) else return(c(k-1,1))
}
############################
### Printing results ######
############################
a=read.csv("ToPlot.csv",header=F)
np=a[30:129,4]
p=as.numeric(gsub("\\%","",a[,5]))/100
p00=mean(p[1:29])
np0=mean(a[1:29,4])
y=p[30:129]*np
tab=NULL
for (p0 in c(p00,0.7,0.75,0.8))
{
 k=test1(260,p0,p,np,3.14)[1]
 k=c(k,test2(260,p0,p,np)[1])
 k=c(k,test3(260,p0,p,np)[1])
  for (lambda in c(0.05,0.1,0.2,0.3))
 {
  k=c(k,ewma(260,p0,y,np,np0,3,lambda)[1])
 }
 tab=rbind(tab,k)
}
write(t(tab),"",ncolumns=7,sep="\t")

np=a[30:129,8]
p=as.numeric(gsub("\\%","",a[,9]))/100
p00=mean(p[1:29])
np0=mean(a[1:29,8])
y=p[30:129]*np

tab=NULL
for (p0 in c(p00,0.5))
{
 k=test1(260,p0,p,np,3.14)[1]
 k=c(k,test2(260,p0,p,np)[1])
 k=c(k,test3(260,p0,p,np)[1])
 for (lambda in c(0.05,0.1,0.2,0.3))
 {
  k=c(k,ewma(260,p0,y,np,np0,3,lambda)[1])
 }
 tab=rbind(tab,k)
}
write(t(tab),"",ncolumns=7,sep="\t")
```

Script for Table 5.3.

```
############################
### EWMA ###################
############################

ewma=function(p0,np,L,lambda)
{
 p=rbinom(1,np,0.4)/1000
 s=(lambda*p+(1-lambda)*p0)*np
 h=L*sqrt(lambda/(2-lambda)*np*p0*(1-p0)*(1-(1-lambda)^2))
 for (k in 2:10)
 {
  p=rbinom(1,np,0.4)/1000
  s=lambda*p*np+(1-lambda)*s
  h=L*sqrt(lambda/(2-lambda)*np*p0*(1-p0)*(1-(1-lambda)^(2*k)))
 }
 k=11
 while(abs(s-p0*np)<h )
 {
  p=rbinom(1,np,0.4)/1000
  s=lambda*p*np+(1-lambda)*s
  h=L*sqrt(lambda/(2-lambda)*np*p0*(1-p0)*(1-(1-lambda)^(2*k)))
  k=k+1
 }
 return(k-1)
}

############################
### Printing Results ######
############################

set.seed(9600)

asn005=c()
asn01=c()
asn02=c()
asn03=c()
np=1000
for (m in 1:10)
{
 test=ewma(0.4,np,3,0.05)
 asn005=c(asn005,test)
```

```
 test=ewma(0.4,np,3,0.1)
 asn01=c(asn01,test)

 test=ewma(0.4,np,3,0.2)
 asn02=c(asn02,test)

 test=ewma(0.4,np,3,0.3)
 asn03=c(asn03,test)
}

print(c(mean(asn005),max(asn005)))
cat("\n")
print(c(mean(asn01),max(asn01)))
cat("\n")
print(c(mean(asn02),max(asn02)))
cat("\n")
print(c(mean(asn03),max(asn03)))
```

Script for Tables 5.4-5.6.

```
#use test1, test2, test3 functions from the previous script

############################
### EWMA ###################
############################

ewma=function(n,p0,p,np,L,lambda)
{
 s=(lambda*p[1]+(1-lambda)*p0)*np[1]
 h=L*sqrt(lambda/(2-lambda)*np[1]*p0*(1-p0)*
 (1-(1-lambda)^2))
 for (k in 2:10)
 {
  s=lambda*p[k]*np[1]+(1-lambda)*s
  h=L*sqrt(lambda/(2-lambda)*np[1]*p0*(1-p0)*
  (1-(1-lambda)^(2*k)))
 }
 k=11
 while(abs(s-p0*np[1])<h && k<=n)
 {
  s=lambda*p[k]*np[1]+(1-lambda)*s
  h=L*sqrt(lambda/(2-lambda)*np[1]*p0*(1-p0)*
  (1-(1-lambda)^(2*k)))
  k=k+1
 }
 if(abs(s-p0*np[1])<h) return(c(n,0)) else return(c(k-1,1))
}

############################
### Saving Results ########
############################

sink("Table53.txt")
set.seed(9600)

power1=NULL
power2=NULL
asn=NULL
np=rep(1000,260)
n=260
nn=1500
p=c()
```

```
xx=format(round(5:9/10,1),nsmall=1)
xx=rep(xx,3)
y=c(26,rep(999,4),130,rep(999,4),200,rep(999,4))
xx=cbind(y,xx)
ll=5000

pow=c()
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 5:9)
 {
  a1=0
  asn1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):n]=rbinom(n-(round(tau*n)-1),1000,i/10)/1000

   test=test1(n,0.4,p,np,3.14)
   a1=a1+test[2]
   asn1=asn1+test[1]
  }
  pow=c(pow,a1/m)
  asn0=c(asn0,asn1/m)
 }
}
power1=cbind(power1,pow)
asn=cbind(asn,asn0)

pow=c()
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 5:9)
 {
  a1=0
  asn1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):n]=rbinom(n-(round(tau*n)-1),1000,i/10)/1000
```

```
   test=test2(n,0.4,p,np)
   a1=a1+test[2]
   asn1=asn1+test[1]
  }
  pow=c(pow,a1/m)
  asn0=c(asn0,asn1/m)
 }
}
power1=cbind(power1,pow)
asn=cbind(asn,asn0)

pow=c()
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 5:9)
 {
  a1=0
  asn1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):n]=rbinom(n-(round(tau*n)-1),1000,i/10)/1000

   test=test3(n,0.4,p,np)
   a1=a1+test[2]
   asn1=asn1+test[1]
  }
  pow=c(pow,a1/m)
  asn0=c(asn0,asn1/m)
 }
}
power1=cbind(power1,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 5:9)
 {
  a1=0
  asn1=0
  b1=0
```

```
  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):nn]=rbinom(nn-(round(tau*n)-1),1000,i/10)/1000

   test=ewma(nn,0.4,p,np,3,0.05)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 5:9)
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):nn]=rbinom(nn-(round(tau*n)-1),1000,i/10)/1000

   test=ewma(nn,0.4,p,np,3,0.1)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
```

```
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 5:9)
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):nn]=rbinom(nn-(round(tau*n)-1),1000,i/10)/1000

   test=ewma(nn,0.4,p,np,3,0.2)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 5:9)
 {
  a1=0
  asn1=0
  b1=0
```

```
  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):nn]=rbinom(nn-(round(tau*n)-1),1000,i/10)/1000

   test=ewma(nn,0.4,p,np,3,0.3)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

power1=cbind(xx,power1)
power2=cbind(xx,format(round(power2,2),nsmall=2))
asn=cbind(xx,format(round(asn),nsmall=0))

cat("Simulated Power Tests1-3")
cat("\n")
cat("\n")
write(t(power1),"",ncolumns=5,sep=" & ")
cat("Simulated Power EWMA")
cat("\n")
cat("\n")
write(t(power2),"",ncolumns=10,sep=" & ")
cat("\n")
cat("ASN")
cat("\n")
cat("\n")
write(t(asn),"",ncolumns=9,sep=" & ")

sink()
```

Script for Tables 5.7-5.9.

```
#use test1, test2, test3, ewma functions from the previous
#script

############################
### Saving Results ########
############################

sink("Table56.txt")
set.seed(9600)

power1=NULL
power2=NULL
asn=NULL
np=rep(1000,260)
n=260
nn=1500
p=c()
xx=format(round(401:410/1000,3),nsmall=3)
xx=rep(xx,3)
y=c(26,rep(999,9),130,rep(999,9),200,rep(999,9))
xx=cbind(y,xx)

ll=5000

pow=c()
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 401:410)
 {
  a1=0
  asn1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):n]=rbinom(n-(round(tau*n)-1),1000,i/1000)/1000

   test=test1(n,0.4,p,np,3.14)
   a1=a1+test[2]
   asn1=asn1+test[1]
  }
```

```
  pow=c(pow,a1/m)
  asn0=c(asn0,asn1/m)
 }
}
power1=cbind(power1,pow)
asn=cbind(asn,asn0)

pow=c()
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 401:410)
 {
  a1=0
  asn1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):n]=rbinom(n-(round(tau*n)-1),1000,i/1000)/1000

   test=test2(n,0.4,p,np)
   a1=a1+test[2]
   asn1=asn1+test[1]
  }
  pow=c(pow,a1/m)
  asn0=c(asn0,asn1/m)
 }
}
power1=cbind(power1,pow)
asn=cbind(asn,asn0)

pow=c()
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 401:410)
 {
  a1=0
  asn1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):n]=rbinom(n-(round(tau*n)-1),1000,i/1000)/1000
```

```
   test=test3(n,0.4,p,np)
   a1=a1+test[2]
   asn1=asn1+test[1]
  }
  pow=c(pow,a1/m)
  asn0=c(asn0,asn1/m)
 }
}
power1=cbind(power1,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 401:410)
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):nn]=
   rbinom(nn-(round(tau*n)-1),1000,i/1000)/1000

   test=ewma(nn,0.4,p,np,3,0.05)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
```

```
for (tau in c(0.1,0.5,200/260))
{
 for (i in 401:410)
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):nn]=
   rbinom(nn-(round(tau*n)-1),1000,i/1000)/1000

   test=ewma(nn,0.4,p,np,3,0.1)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 401:410)
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):nn]=
   rbinom(nn-(round(tau*n)-1),1000,i/1000)/1000
```

```
   test=ewma(nn,0.4,p,np,3,0.2)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in 401:410)
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,0.4)/1000
   p[round(tau*n):nn]=
   rbinom(nn-(round(tau*n)-1),1000,i/1000)/1000

   test=ewma(nn,0.4,p,np,3,0.3)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)
```

```
power1=cbind(xx,format(round(power1,2),nsmall=2))
power2=cbind(xx,format(round(power2,2),nsmall=2))
asn=cbind(xx,format(round(asn),nsmall=0))

cat("Simulated Power Tests1-3")
cat("\n")
cat("\n")
write(t(power1),"",ncolumns=5,sep=" & ")
cat("Simulated Power EWMA")
cat("\n")
cat("\n")
write(t(power2),"",ncolumns=10,sep=" & ")
cat("\n")
cat("ASN")
cat("\n")
cat("\n")
write(t(asn),"",ncolumns=9,sep=" & ")

sink()
```

Script for Tables 5.10-5.12.

```
#use test1, test2, test3, ewma functions from the previous
#script

############################
### Saving Results ########
############################

sink("Table59.txt")
set.seed(9600)

power1=NULL
power2=NULL
asn=NULL
np=rep(1000,260)
n=260
nn=1500
p=c()
xx=format(round(c(0.3,0.39,0.399,0.4,0.401,0.41,0.5),3),
nsmall=3)
xx=rep(xx,3)
y=c(26,rep(999,6),130,rep(999,6),200,rep(999,6))
xx=cbind(y,xx)

ll=5000

pow=c()
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in c(0.3,0.39,0.399,0.4,0.401,0.41,0.5))
 {
  a1=0
  asn1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,i)/1000
   p[round(tau*n):n]=rbinom(n-(round(tau*n)-1),1000,0.4)/1000

   test=test1(n,i,p,np,3.14)
   a1=a1+test[2]
   asn1=asn1+test[1]
```

```
  }
  pow=c(pow,a1/m)
  asn0=c(asn0,asn1/m)
 }
}
power1=cbind(power1,pow)
asn=cbind(asn,asn0)

pow=c()
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in c(0.3,0.39,0.399,0.4,0.401,0.41,0.5))
 {
  a1=0
  asn1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,i)/1000
   p[round(tau*n):n]=rbinom(n-(round(tau*n)-1),1000,0.4)/1000

   test=test2(n,i,p,np)
   a1=a1+test[2]
   asn1=asn1+test[1]
  }
  pow=c(pow,a1/m)
  asn0=c(asn0,asn1/m)
 }
}
power1=cbind(power1,pow)
asn=cbind(asn,asn0)

pow=c()
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in c(0.3,0.39,0.399,0.4,0.401,0.41,0.5))
 {
  a1=0
  asn1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,i)/1000
```

```
   p[round(tau*n):n]=rbinom(n-(round(tau*n)-1),1000,0.4)/1000

   test=test3(n,i,p,np)
   a1=a1+test[2]
   asn1=asn1+test[1]
  }
  pow=c(pow,a1/m)
  asn0=c(asn0,asn1/m)
 }
}
power1=cbind(power1,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in c(0.3,0.39,0.399,0.4,0.401,0.41,0.5))
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,i)/1000
   p[round(tau*n):nn]=rbinom(nn-(round(tau*n)-1),1000,0.4)/1000

   test=ewma(nn,i,p,np,3,0.05)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
```

```
for (tau in c(0.1,0.5,200/260))
{
 for (i in c(0.3,0.39,0.399,0.4,0.401,0.41,0.5))
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,i)/1000
   p[round(tau*n):nn]=rbinom(nn-(round(tau*n)-1),1000,0.4)/1000

   test=ewma(nn,i,p,np,3,0.1)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in c(0.3,0.39,0.399,0.4,0.401,0.41,0.5))
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,i)/1000
   p[round(tau*n):nn]=rbinom(nn-(round(tau*n)-1),1000,0.4)/1000

   test=ewma(nn,i,p,np,3,0.2)
   a1=a1+test[2]
```

```
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

pow=NULL
asn0=c()
for (tau in c(0.1,0.5,200/260))
{
 for (i in c(0.3,0.39,0.399,0.4,0.401,0.41,0.5))
 {
  a1=0
  asn1=0
  b1=0

  for (m in 1:ll)
  {
   p[1:(round(tau*n)-1)]=rbinom((round(tau*n)-1),1000,i)/1000
   p[round(tau*n):nn]=rbinom(nn-(round(tau*n)-1),1000,0.4)/1000

   test=ewma(nn,i,p,np,3,0.3)
   a1=a1+test[2]
   asn1=asn1+test[1]
   if (test[1]<tau*n)
   {
    b1=b1+1
   }
  }
  pow=rbind(pow,c(a1/m,b1/m))
  asn0=c(asn0,asn1/m)
 }
}
power2=cbind(power2,pow)
asn=cbind(asn,asn0)

power1=cbind(xx,format(round(power1,2),nsmall=2))
power2=cbind(xx,format(round(power2,2),nsmall=2))
```

```
asn=cbind(xx,format(round(asn),nsmall=0))

cat("Simulated Power Tests1-3")
cat("\n")
cat("\n")
write(t(power1),"",ncolumns=5,sep=" & ")
cat("Simulated Power EWMA")
cat("\n")
cat("\n")
write(t(power2),"",ncolumns=10,sep=" & ")
cat("\n")
cat("ASN")
cat("\n")
cat("\n")
write(t(asn),"",ncolumns=9,sep=" & ")

sink()
```