A Minimal-Maximal Correlation-Type Goodness-of-Fit Test

by

Josie Ann White

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
In partial fulfillment of the requirements for the degree of

Master of Science

Department of Statistics
University of Manitoba
Winnipeg

Copyright © 2014 by Josie A. White

Abstract

In goodness-of-fit testing, the goal is to determine if data come from a particular distribution. One graphical approach to test goodness-of-fit is a probability plot. Two probability plots typically used are the probability-probability plot and the quantile-quantile plot, but to use these plots, plotting points are needed. Balakrishnan et al. (2010) proposed a new plotting point based on simultaneous closeness probabilities. This was followed up by a correlation-type goodness-of-fit test based on these plotting points.

In this thesis, two tests based on the correlation coefficient test are proposed; in particular, a maximal-correlation coefficient test and a minimal-correlation coefficient test which are based on simultaneous closeness probabilities are developed. Two approaches are considered to investigate these two tests: a grid search method and an averaging method. Numerical results, including illustrative examples, critical values and a power study are also provided.

Acknowledgments

I would like to express my very great appreciation to my advisor Dr. Katherine Davies for all of the guidance and support she has provided me with through out my time in the graduate program. Without her valuable assistance and enthusiast encouragement my thesis would not have been possible. I would also like to thank my examining committee members, Dr. Po Yang and Dr. Narayanaswamy Balakrishnan.

I would like to thank my professors at Brandon University for encouraging me to further my education in statistics. I would also like to acknowledge University of Manitoba for providing funding for me through the University of Manitoba Graduate Scholarship.

I would especially like to thank my husband Mike Kshymensky for his endless support, encouragement and optimism during my schooling. I am also thankful for my parents Mike and Sue White, and my brothers and sisters for their continuous support and reassurance.

Dedication

To my husband and family.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The Pitman closeness criterion allows for the comparison of two estimators of a statistical parameter. It can determine if one estimator is better than another. A lot of work has been completed on Pitman closeness to ordered data including Pitman closeness to the population median of a distribution (Balakrishnan et al., 2009b) and Pitman closeness related to population quantiles (Balakrishnan et al., 2009a). This work led to the introduction of simultaneous Pitman closeness probabilities which can be found in Balakrishnan et al. (2010). This work has the advantage of being able to compare all order statistics at once as opposed to the previous pairwise comparisons.

Goodness-of-fit tests are generally used to determine how well a set of observations can be fit to a model. There are two types of plots that are typically used to visually assess goodness-of-fit; these two plots are the probability-probability plot and the quantile-quantile plot. A probability-probability plot compares an empirical cumulative distribution function to a specified theoretical distribution while a

quantile-quantile plot compares quantiles of the two probability distributions. Both type of plots require the use of plotting points, which is a controversial topic. Balakrishnan et al. (2012a) recently introduced plotting points based on simultaneous Pitman closeness probabilities.

The correlation coefficient goodness-of-fit test was introduced by Filliben (1975). This test is based on the linearity of probability plots and is therefore relatively easy to understand as well as to calculate. Balakrishnan et al. (2012a) proposed a correlation-type goodness-of-fit test statistic based on simultaneous closeness probability plotting points. However, optimal plotting points could not be found for the first and last order statistic.

In this thesis, I will introduce a minimal-maximal correlation test which has two forms: one based on the maximum correlation and the other based on the minimum correlation. Both of these tests will use simultaneous closeness probability plotting points. Furthermore, the tests will be carried out in two ways: a grid search method and an averaging method. A comparison between the two methods will be carried out, which will come in the form of tables of critical values of the associated test statistics and power properties of the proposed tests for both methods. I will also provide a few examples to illustrate these tests.

1.2 Thesis Organization

In Chapter 2, I provide some background material. First, I describe Pitman closeness probabilities and their use in the analysis of ordered data in more detail. Then I will provide an overview of simultaneous Pitman closeness probabilities and their

applications. This will be followed up by an overview of goodness-of-fit assessments, including descriptions of probability-probability plots and quantile-quantile plots. Subsequently I will discuss a correlation-type goodness-of-fit test and its use in probability plots. Finally, I will outline plotting points based on simultaneous closeness probabilities and their use in a correlation-type goodness-of-fit test.

In Chapter 3, I will introduce the proposed correlation goodness-of-fit tests based on maximal and minimal correlations. For each, I will first consider a grid search method, and secondly I will consider a method where the average plotting points are found from the first method and used in each test. I will provide critical values for the maximal- and minimal-correlation tests for both methods. I will also carry out a power study to investigate their performance. I will follow up with a demonstration of the tests using several data sets.

Important segments of the R code used to produce the results in this thesis are provided in the Appendix.

Chapter 2

Preliminaries

2.1 Pitman Closeness

2.1.1 History

The concept of *Pitman closeness* was introduced by Pitman (1937). This criterion aims to compare two estimators within a class where both estimators are based on a sample size of n. It can be defined as follows: the Pitman closeness of T_1 relative to T_2 is the probability that the estimator T_1 produces an estimate that is closer to a real-valued parameter θ than the one which is produced by the estimator T_2 (Balakrishnan et al., 2011b). Formally, it is given by

$$\mathbb{P}(T_1, T_2|\theta, n) = Pr(|T_1 - \theta| < |T_2 - \theta|). \tag{2.1}$$

This is also known as the Pitman closeness probability or Pitman nearness. Furthermore, if

$$Pr(|T_1 - \theta| < |T_2 - \theta|) \geq \frac{1}{2} \tag{2.2}$$

for all values of θ with strict inequality holding for at least one value of θ, then it can be said that T_1 is a closer estimator of θ than T_2 (Balakrishnan et al., 2009a). It should be noted that Pitman closeness does not measure how much closer T_1 is to θ than the competing estimator T_2 (Keating et al., 1993).

A related criteria for comparing two estimators is *Banks' criterion*. Introduced in 1997, it states that an estimator T_1 is closer to a parameter θ than another estimator T_2 if

$$\mathbb{P}(|T_1 - \theta| < \epsilon) > \mathbb{P}(|T_2 - \theta| < \epsilon) \tag{2.3}$$

for some $\epsilon > 0$. This differs from Pitman closeness since T_1 is preferred over a competing estimator T_2 if it has a greater probability of being within ϵ of θ than T_2. This criterion can be thought to compare the clustering of T_1 to that of T_2 in an ϵ - neighborhood of the parameter θ. Banks' criterion can also be generalized to a simultaneous comparison of multiple estimators of θ.

Pitman closeness has also been used in the area of Bayesian statistics. Ghosh and Sen (1991) introduced the notion of posterior Pitman closeness and identified the differences between this new Bayesian approach and traditional Pitman closeness. They found that this new criterion avoided the drawbacks of the usual notion of Pitman closeness involving transitivity.

2.1.2 Some Issues and Controversies

One major concern with the Pitman closeness criterion that has been noted in the literature is that it lacks the transitive property. If a set of items is transitive, it relates A to B, B to C and also A to C. A relation is said to be intransitive if A

cannot be related to C. A simple example is in sports: Team A beats Team B, and Team B beats Team C, but Team A does not beat Team C. Formally, intransitivity of random variables can be defined as: for any real-valued random variables A, B, and C, stochastic intransitiveness occurs whenever $Pr(A < B)$, $Pr(B < C)$, and $Pr(C < A)$ all exceed 0.50. These probabilities imply that there is a better than 50% chance of each of the events occurring: A is less than B, B is less than C and C is less than A. In the context of Pitman closeness, this occurs, for instance, when three estimators are to be compared, say $\hat{\theta}_1$, $\hat{\theta}_2$, and $\hat{\theta}_3$ and $\mathbb{P}(\hat{\theta}_1, \hat{\theta}_2|\theta)$, $\mathbb{P}(\hat{\theta}_2, \hat{\theta}_3|\theta)$ and $\mathbb{P}(\hat{\theta}_3, \hat{\theta}_1|\theta)$ are all greater than 0.50. When this occurs it raises the question as to which of the estimators is the best choice (Balakrishnan et al., 2011b).

Another controversy with the Pitman closeness criterion is the pairwise-worst simultaneous best paradox. This paradox can be defined as follows: for any real-valued random variables X_1, X_2, and X_3, it is possible for $Pr(X_i = min\{X_1, X_2, X_3\})$ to be the largest for $i = 3$, even though $Pr(X_1 < X_3)$ and $Pr(X_2 < X_3)$ exceed 0.50. Hence, while X_3 is preferred over X_1 and X_2 in any simultaneous comparison, in the pairwise comparisons it is worst (Balakrishnan et al., 2011b). Similar to this, a pairwise-best simultaneous-worst paradox can also occur. This can be defined analogously as follows: for any real-valued random variables X_1, X_2, and X_3, it is possible for $Pr(X_i = min\{X_1, X_2, X_3\})$ to be the smallest for $i = 1$, even though $Pr(X_1 < X_2)$ and $Pr(X_1 < X_3)$ exceed 0.50. Here, for the simultaneous comparison, X_1 is the least preferred over both X_2 and X_3, even though X_1 is preferred over each one in the pairwise comparison.

2.1.3 Pitman Closeness and Ordered Data: Recent Developments

A lot of work has been done recently on Pitman closeness related to ordered data. One of the initial works on this was completed by Balakrishnan et al. (2009b) who studied the Pitman closeness of order statistics to the population median of a distribution. Let $X_{1:n} < \cdots < X_{n:n}$ be the order statistics from a random sample of size n from an arbitrary absolutely continuous distribution. The sample median is defined as $M_n = X_{m:n}$ if $n = 2m - 1$ and $M_n = \frac{1}{2}(X_{m:n} + X_{m+1:n})$ if $n = 2m$, and denote the population median by μ. In some general situations, the authors showed that the sample median M_n was the Pitman-closest order statistic to the population median. In this context, the Pitman closeness of order statistics to the population median is

$$\pi_i = Pr(|M_n - \mu| < |X_{i:n} - \mu|), \text{ for } i = 1, \ldots n,$$

except for $i = m$ in the case $n = 2m - 1$. From this, it was found that for all $i = 1, \ldots, n$ (except $i = m$ in the case $n = 2m - 1$), $\pi_i > \frac{1}{2}$. It was also shown that π_i is decreasing in i for $i = 1, \ldots, m$, and increasing in i for $i = m + 1, \ldots, n$; this was shown to hold for any distribution if $n = 2m - 1$ and for any symmetric distribution if $n = 2m$.

For a sample size that is odd, in the case of symmetric distributions and taking $\mu = 0$, the authors found that the probabilities of closeness π_i were symmetric, i.e., $\pi_i = \pi_{n-i+1}$, and the probabilities π_i, for $i = 1, \ldots, n$, were distribution-free. They also found a large sample approximation for π_{m+1} using Stirling's approximation and showed that $\pi_{m+1} \to \frac{1}{2}$ as $m \to \infty$. An expression was also derived for π_l for

$m+1 < l < 2m-1$ and a table of values was produced supporting the results that $\pi_l > \frac{1}{2}$. This table also supported the observation that π_l increases as l increases.

For a sample size that is even, the authors defined, for $\mu = 0$, $\pi_i^* = Pr(|X_{m+1:n}| < |X_{i:n}|)$ for $i = m+2, \ldots, n$. An explicit expression was found for π_{m+2}^* and also for π_l^* for $m+2 < l < n$. Symmetry was also shown, i.e., for $\pi_i^{**} = Pr(|X_{m:n}| < |X_{i:n}|)$ then for $i = 1, \ldots, m-1$, $\pi_i^{**} = \pi_{n-i+2}^*$. A table of the probabilities of closeness for π_l^* demonstrated that $\pi_l^* > \frac{1}{2}$. This table also confirmed that as l increases, so does π_l^*.

Further work has also been done on comparing estimators under the criterion based on censored samples. Balakrishnan et al. (2011b) compared the Best Linear Unbiased Estimator (BLUE) and Best Linear Invariant Estimator (BLIE) under the Pitman closeness criterion based on a Type-II right censored sample from the exponential distribution. Suppose that a Type-II right censored sample is taken from the Exp(θ) distribution and let $X_{1:n} \leq \cdots \leq X_{r:n}$ be the order statistics, with the largest $n-r$ order statistics being censored. The BLUE and BLIE are then defined as $\theta_r^* = (1/r)T_r$ and $\hat{\theta}_r = (1/(r+1))T_r$, respectively, where $T_r = \sum_{i=1}^{r} X_{i:n} + (n-r)X_{r:n}$, which is often known as the total time on test. Noting that $2T_r/\theta \sim \chi_{2r}^2$, the exact Pitman closeness probability between $\hat{\theta}_r$ and θ_r^* was calculated as

$$\pi_r = Pr(|\theta_r^* - \theta| \leq |\hat{\theta}_r - \theta|) = Pr\left(\chi_{2r}^2 \leq \frac{4r+4}{2r+1}r\right), \tag{2.4}$$

where χ_a^2 denotes a central chi-square random variable with a degrees of freedom. As can be seen in Equation 2.4 , it was determined that the probability of closeness

only depended on r and not on either θ or n. Furthermore, they also established that the BLIE is Pitman closeness (PC) inadmissible compared with the BLUE, where an estimator T_j is PC-admissible if there is an estimator T_i in a class $\mathcal{C}$ which is closer than T_j.

Similar work has been completed on comparing the Best Linear Unbiased Predictor (BLUP) and Best Linear Invariant Predictor (BLIP) of censored order statistics from an exponential distribution in one-sample and two-sample situations under the Pitman closeness criterion; this was done by Balakrishnan et al. (2012b). In the one sample case, consider a Type-II censored sample of size r from n variables from Exp(θ) and let $X_{1:n} \leq \cdots \leq X_{r:n}$ denote the r order statistics. It may be of use to an experimenter to be able to predict how long the experiment would have lasted had it not terminated at the rth failure, i.e., the prediction of $X_{s:n}$ for $r < s \leq n$. In this case, there is no real valued parameter but instead a random quantity, $X_{s:n}$. In the case of the exponential distribution, it is known that the BLUP and BLIP of $X_{s:n}$ based on a Type-II right censored sample of size r are

$$X^*_{s:n} = X_{r:n} + \theta^*_r(\alpha_{s:n} - \alpha_{r:n}),$$

$$X^{**}_{s:n} = X_{r:n} + \hat{\theta}_r(\alpha_{s:n} - \alpha_{r:n}),$$

respectively, where θ^*_r and $\hat{\theta}_r$ are the BLUE and BLIE of θ, and

$$\alpha_{i:n} = \frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-i+1} \text{ for } i = 1, 2, \ldots, n.$$

The Pitman closeness probability between $X^*_{s:n}$ and $X^{**}_{s:n}$ in the prediction of $X_{s:n}$ was then found to be

$$\pi_{r,s:n} = Pr(|X_{s:n} - X^*_{s:n}| < |X_{s:n} - X^{**}_{s:n}|)$$

$$= b_{r,s,n} \sum_{i=0}^{s-r-1} (-1)^i \binom{s-r-1}{i} \left(\frac{1}{n-s+i+1}\right) \left(\frac{1}{[1+(n-s+i+1)D]}\right)^r, \tag{2.5}$$

where $D = \frac{(\alpha_{s:n}-\alpha_{r:n})(2r+1)}{2r(r+1)}$ and $b_{r,s,n} = \frac{(n-r)!}{(s-r-1)!(n-s)!}$.

Using this expression, for $n = 10$ and 15 for $r = 1(1)n-1$ and $s = r+1(1)n$, the authors found exact Pitman closeness probabilities. These results showed that when $r = 1$, the BLUP is Pitman closer than the BLIP. However when $s = r+1$ except when $r = 1$, the BLIP is always Pitman closer. For small values of r, the BLUP is generally Pitman closer, however the reverse is true for larger r, i.e., the BLIP is Pitman closer.

For the two special cases when $r = 1$ and $s = r+1$, they found that for $r = 1$, it did not matter what the choice of s and n were since the BLUP was always Pitman closer than the BLIP. However for $s = r+1$, the BLIP was Pitman closer than the BLUP for all $r > 1$ and when $r = 1$, the BLUP was Pitman closer.

For the two sample case, suppose the interest lies in predicting future sample lifetimes $Y_{1:m}, \ldots, Y_{m:m}$ from Exp(θ). Let $X_{1:n} \leq \cdots \leq X_{r:n}$ be a Type-II right censored sample from the Exp(θ) distribution and let θ^*_r and $\hat{\theta}_r$ be the BLUE and BLIE of θ as defined earlier. For predicting $Y_{s:m}$, the BLUP and BLIP of $Y_{s:m}$ are, $Y^*_{s:m} = \alpha_{s:m}\theta^*_r$ and $Y^{**}_{s:m} = \alpha_{s:m}\hat{\theta}_r$, respectively, and $\alpha_{s:m} = \frac{1}{m} + \frac{1}{m-1} + \cdots + \frac{1}{m-s+1}$. The authors found the Pitman closeness between $Y^*_{s:m}$ and $Y^{**}_{s:m}$ as predictors of $Y_{s:m}$ to be

$$\pi_{s:m(r:n)} = Pr(|Y^{*}_{s:m} - Y_{s:m}| < |Y^{**}_{s:m} - Y_{s:n}|)$$

$$= \frac{m!}{(s-1)!(m-s)!} \sum_{j=0}^{s-1} (-1)^{s-1-j} \binom{s-1}{j} \frac{1}{m-j}$$

$$\times \left[\frac{2r(r+1)}{2r(r+1) + \alpha_{s:m}(2r+1)(m-j)} \right]^{r}. \tag{2.6}$$

For $r = 1, 5$ and 10 with $m = 5$ to 15 when $s \geq 2$, the authors found that the BLUP is always Pitman closer when $r = 1$ or 5. However, when $r = 10$, the outcome depended on m and s. The BLIP was found to be Pitman closer than the BLUP for small values of s, and for larger values of s the BLUP was Pitman closer than the BLIP. For the special case when $s = 1$, the authors found that the BLIP was uniformly better than the BLUP.

Another type of censoring that has been considered is progressive censoring. A progressively Type-II right censoring experiment is when n identical units from an absolutely continuous distribution are placed on a test and r complete failures are to be observed, where $1 \leq r \leq n$. To reduce the total time on test or the number of failed items, the $n-r$ remaining lifetimes are to be progressively censored such that R_1 surviving units are to be withdrawn at random from the test at the time of the first failure, R_2 surviving units are to be withdrawn at random from the test at the time of the second failure and so forth. This continues until all remaining R_r surviving units are to be withdrawn at the time of the rth failure. The lifetime data observed in this way produce Type-II right progressively censored order statistics (PCOS) and we denote the progressive censoring scheme (PCS) by $R = (R_1, R_2, \ldots, R_r)$. With this in mind, in the case of the exponential distribution

with mean lifetime θ, Volterman et al. (2012) wished to determine the optimal progressive censoring scheme to estimate θ under the Pitman closeness criterion. For this, they considered two different progressive censoring schemes, $\tilde{R} = (R_1, \ldots, R_m)$ and $\tilde{S} = (S_1, \ldots, S_m)$, such that $R_i \neq S_i$ for at least two i. They then defined the Pitman closeness probability as

$$Pr(\theta^*_{\tilde{R}}, \theta^*_{\tilde{S}}|\theta) = Pr(|\theta^*_{\tilde{R}} - \theta| < |\theta^*_{\tilde{S}} - \theta|), \tag{2.7}$$

where $\theta^*_{\tilde{R}}$ and $\theta^*_{\tilde{S}}$ are the BLUEs based on PCSs $\tilde{R}$ and $\tilde{S}$, respectively. If this probability exceeds 0.50, it can be said that $\tilde{R}$ is a better censoring scheme than $\tilde{S}$ for the purpose of estimating θ. The authors then compared various progressive censoring schemes to the right censoring scheme for different samples of size n and showed that the right censoring scheme was indeed optimal compared to the others. They did this by computing the probability exactly for small sample sizes and for larger samples they used $1,000,000$ Monte Carlo simulations.

2.1.4 Pitman Closeness and Population Quantiles

2.1.4.1 Pitman Closeness of Order Statistics to Population Quantiles

The work on Pitman closeness of order statistics to the population median was extended to comparing order statistics to population quantiles in Balakrishnan et al. (2009a). Let $Y_1, \ldots, Y_n$ be a random sample of size n with cumulative distribution function (cdf) $G(y)$ and probability distribution function (pdf) $g(y)$. The authors goal was to determine the closest order statistic to a particular population quantile. The authors denoted ξ^*_p as the pth quantile of $G(y)$, where the pth quantile, ξ_p, is

defined as a point satisfying the equation $Pr(Y \leq \xi_p^*)$. Furthermore, they assumed $G(\cdot)$ belongs to the location-scale family of distributions such that

$$G(y) = F\left(\frac{y-\mu}{\sigma}\right) \text{ and } g(y) = \frac{1}{\sigma} f\left(\frac{y-\mu}{\sigma}\right) \forall y \in \Re$$

with location parameter $\mu \in \Re$ and scale parameter $\sigma > 0$. Let $X_{1:n} \leq \cdots \leq X_{n:n}$ denote the corresponding order statistics from the standard distribution with cdf $F(x)$ and pdf $f(x)$ from a random sample of size n. For any two order statistics $X_{i:n}$ and $X_{l:n}$, they defined the probability of Pitman closeness to the population quantile ξ_p, where $\xi_p = (\xi_p^* - \mu)/\sigma$, as

$$\pi_{(l)i}(p) = Pr(|X_{l:n} - \xi_p| < |X_{i:n} - \xi_p|) \text{ for } i = \{1, \ldots, n\}/l.$$

For $i = l+1, l+2, \ldots, n$, using properties of order statistics, the general expression for the Pitman closeness between two order statistics, $X_{l:n}$ and $X_{i:n}$, to a population quantile ξ_p, was then found to be

$$\pi_{(l)i}(p) = 1 - I_p(l, n-l+1) + k_{l,i,n} \sum_{j=0}^{i-l-1} (-1)^{i-l-1-j} \binom{i-l-1}{j} \frac{1}{n-l-j}$$
$$\times \int_{-\infty}^{\xi_p} \{F(x)\}^{l-1} \{1-F(x)\}^j \{1-F(-x+2\xi_p)\}^{n-l-j} f(x)dx, \quad (2.8)$$

and for $i = 1, 2, \ldots l-1$,

$$\pi_{(l)i}(p) = I_p(l, n-l+1) + k_{l,i,n} \sum_{j=0}^{l-i-1} (-1)^{l-i-1-j} \binom{l-i-1}{j} \frac{1}{l-j-1}$$

$$\times \int_{\xi_p}^{\infty} \{F(y)\}^j \{1-F(y)\}^{n-l} \{F(2\xi_p - y)\}^{l-j-1} f(y) dy, \quad (2.9)$$

where

$$I_x(\alpha, \beta) = \frac{1}{B(\alpha, \beta)} \int_0^x u^{\alpha-1}(1-u)^{\beta-1} du, \;\; 0 < x < 1$$

is the incomplete beta ratio, $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta)$ is the complete beta function, and

$$k_{l,i,n} = \frac{n!}{(l-1)!(i-l-1)!(n-i)!} \;\; \text{for } 1 \leq l < i \leq n.$$

For comparing two contiguous order statistics, i.e., $X_{i:n}$ and $X_{i+1:n}$, the authors defined $\pi_i(p) = Pr(|X_{i:n} - \xi_p| < |X_{i+1:n} - \xi_p|)$ for $i = 1, \ldots, n-1$, for which they found a general expression for $\pi_i(p)$ as

$$\pi_i(p) = 1 - I_p(l, n-i+1) + a_{i,n} \int_{-\infty}^{\xi_p} [F(x)]^{i-1} [1 - F(-x + 2\xi_p)]^{n-i} f(x) dx, \quad (2.10)$$

where $a_{i:n} = \frac{n!}{(i-1)!(n-i)!}$ for $i = 1, \ldots, n-1$.

It was shown that if $X_{l:n}$, for some $l \in 1, \ldots, n$, is the Pitman-closest order statistic to ξ_p, then $Y_{l:n}$ is the Pitman-closest order statistic to ξ_p^*. For a distribution

that is symmetric about the origin, suppose that $X_{l:n}$ is the Pitman-closest order statistic to the pth quantile ξ_p. The authors also showed that $X_{n-l+1:n}$ is the Pitman-closest order statistic to the $(1-p)$th quantile ξ_{1-p}.

To infer about a specified quantile, if the Pitman closeness probability for a given l is greater than 0.50 for all i, then it can be said that the lth order statistic is the Pitman-closest order statistic to the specified population quantile and hence can be used as an estimator. The authors demonstrated this procedure for the uniform, exponential and power function distributions.

For the uniform distribution, the authors found an expression for the Pitman closeness probability associated with two order statistics and tables were created to provide closeness probabilities for $p = 0.1, 0.25, 0.75$ and 0.90 and for sample sizes $n = 10$ and $n = 15$. Consider estimating $p = 0.25$ quantile, then for the uniform distribution with $n = 10$, looking at $l = 3$, it can be seen that for all i, $\pi_{i:10}$ is greater than 0.50. So it can be said that $X_{3:10}$ is the Pitman-closest estimator to the 25th percentile. Also, tables were constructed to provide a summary of the Pitman closest order statistic to the pth quantile of the distribution for sample sizes $n = 5(5)20$. These tables allow for one to identify the closest order statistic to a particular quantile for a specified sample size.

Similar to what was done for the uniform distribution, the Pitman closeness probabilities for the standard exponential and the power function distributions were derived and calculated. Tables for the Pitman-closest order statistics to the pth quantile of the exponential distribution turned out to be the exact same results as those found for the uniform distribution. For the power function distribution it was found that the Pitman closest order statistics do change but are still quite similar

to results for both the uniform and exponential distribution. These differences were seen for for extreme values of α, i.e., $0 < \alpha < 0.25$.

2.1.4.2 Simultaneous Closeness among Order Statistics to Population Quantiles

In order to determine the most frequently closest order statistic among all other order statistics to a population quantile under Pitman closeness, a pairwise comparison could be used but it is both time consuming and cumbersome. To overcome these concerns, Balakrishnan et al. (2010) suggested a simultaneous comparison.

With the same idea in mind, Blyth (1972) suggested the reduction of the comparisons by considering the joint distributions of the estimators being compared with respect to loss functions. For this, Blyth suggested two criteria. Blyth's first criterion suggests choosing the estimator within a class $\mathcal{C}$ which is most frequently closest to the value of the unknown parameter θ. In other words, select $\hat{\theta}_i$ from among estimators in a class $\mathcal{C}$ for which

$$\max Pr_{i \in \mathcal{K}} \lceil \mathcal{L}_i = \min_{j \in \mathcal{K}} (\mathcal{L}_j) \rceil, \tag{2.11}$$

where k is an index set for $\mathcal{C}$, and $\mathcal{L}_i = |\hat{\theta}_i - \theta|$. This can be thought of as a max-min criterion, in that the probability that $\hat{\theta}_i$ has the smallest loss among all estimators in $\mathcal{C}$ is maximized. The second criterion is to choose the estimator within the class $\mathcal{C}$ which is least-frequently farthest from θ. This minimizes the probability that $\hat{\theta}_i$ has the maximum loss among the estimators in $\mathcal{C}$ and hence can be thought

of as a min-max criterion. This can be seen as choosing $\hat{\theta}_i$ from among estimators in $\mathcal{C}$ for which

$$\min Pr_{i\in\mathcal{K}}\lceil\mathcal{L}_i = \max_{j\in\mathcal{K}}(\mathcal{K}_j)\rceil. \tag{2.12}$$

Note that whenever the size of the index set is two, both these criteria are equivalent to the Pitman closeness criterion.

Using Blyth's first criterion, in the context of order statistics as estimators, Balakrishnan et al. (2010) defined the *simultaneous closeness probability (SCP)* of $X_{i:n}$ for $i \in 1, \ldots, n$ among the order statistics $X_{1:n} \leq \cdots \leq X_{n:n}$, in the estimation of a population parameter θ as

$$\pi_{i:n}(\theta) = Pr\left(|X_{i:n} - \theta| < \min_{j,j\neq i} |X_{j:n} - \theta|\right) \tag{2.13}$$

for all $i \in 1 \ldots n$. This simplifies the probability computations in Blyth's first criterion and allows for the partitioning of a random vector of observations into regions for which each order statistic is the best. The idea of this comparison is to determine the probability that each order statistic is simultaneously closest to θ when being compared to the remaining order statistics in the sample. For this, geometric arguments are needed, and in particular, the concept of Voronoi regions or tessellations. A Voronoi tessellation is a way of dividing space into different regions and since the estimators are ordered in this case it makes the computation less complex. The best estimator is chosen by looking for the order statistic with the highest probability of being closest to the parameter of interest. Let $\mathcal{A}_{i:n}$ be

the pairwise Voronoi region associated with $X_{i:n}$ when compared with the previous order statistic $X_{i-1:n}$, i.e.,

$$\mathcal{A}_{i:n} = \{\mathbf{x} \in \Re_n : X_{i-1:n} + X_{i:n} \leq 2\theta\} \text{ for each } i \in \{2, \ldots, n\},$$

and let $\mathcal{B}_{i:n}$ be the simultaneous Voronoi region associated with $X_{i:n}$. Then, $X_{i:n}$ is closer to θ than all other order statistics in this region, i.e.,

$$\mathcal{B}_{i:n} = \left\{\mathbf{x} \in \Re_n : |X_{i:n} - \theta| \leq \min_{j,j\neq i} |X_{j:n} - \theta|\right\} \text{ for each } i \in \{1, \ldots, n\}.$$

For support $\Re$, for $i = 2, \ldots, n-1$, the authors derived the simultaneous closeness probability $\pi_{i:n}(\theta)$ of $X_{i:n}$ in the estimation of θ as

$$\begin{aligned}
\pi_{i:n}(\theta) &= Pr\{\mathcal{B}_{i:n}\} \\
&= Pr\{\mathcal{A}_{i:n}\} - Pr\{\mathcal{A}_{i+1:n}\} = \binom{n}{i-1}[F(\theta)]^{i-1}[1-F(\theta)]^{n-i+1} \\
&+ \int_0^{F(\theta)} \frac{n!}{(i-1)!(n-1)!}[\bar{F}(2\theta - F^{-1}(u))]^{n-i}u^{i-1}du \\
&- \int_0^{F(\theta)} \frac{n!}{(i-2)!(n-i+1)!}[\bar{F}(2\theta - F^{-1}(u))]^{n-i+1}u^{i-2}du. \qquad (2.14)
\end{aligned}$$

From this, the authors found the simultaneous closeness probabilities for the special cases $\pi_{1:n}(\theta)$ and $\pi_{n:n}(\theta)$, for the smallest and largest order statistics, which can be expressed as

$$\pi_{1:n}(\theta) = Pr\{\mathcal{B}_{1:n}\} = 1 - Pr\{\mathcal{A}_{2:n}\}$$

and

$$\pi_{n:n}(\theta) = Pr\{\mathcal{B}_{n:n}\} = 1 - Pr\{\mathcal{A}_{n:n}\}.$$

With Blyth's second criterion in mind, $\pi_{i:n}(\theta)$ can be interpreted as the probability that the loss due to $X_{i:n}$ in the estimation of θ is less than the losses due to all the other order statistics.

For X with a bounded support on the interval (a, b), $\pi_{i:n}(\theta)$ was given as

$$\pi_{i:n}(\theta) = n\binom{n-1}{i-2}\int_a^{b^*} f(x)[F(x)]^{i-2}\{[F(h_2(x))]^{n-i+1} - [F(h_1(x))]^{n-i+1}\}dx, \tag{2.15}$$

where $b^* = min(b, 2\theta - a)$, $h_1(x) = max(a, x)$ and $h_2(x) = min(b, 2\theta - x)$.

Considering a location-scale family and ξ_p^* as the pth quantile of $F(x)$, for a complete support, it was found that the simultaneous closeness probabilities do not depend on location and scale parameters but rather depend on p, where $p \in (0, 1)$, and n. For $i = 2, \ldots, n-1$, setting $\theta = \xi_p^*$, the authors found the SCPs to be

$$\begin{aligned}\pi_{i:n}(p) = &\binom{n}{i-1}p^{i-1}(1-p)^{n-i+1} \\ &+ \frac{n!}{(i-1)!(n-i)!}\int_0^p \{1 - G[2G^{-1}(p) - G^{-1}(u)]\}^{n-i}u^{i-1}du \\ &- \frac{n!}{(i-2)!(n-i+1)!}\int_0^p \{1 - G[2G^{-1}(p) - G^{-1}(u)]\}^{n-i+1}u^{i-2}du \end{aligned} \tag{2.16}$$

where, once again,

$$F(x) = G\left(\frac{x-u}{\sigma}\right) \text{ and } f(x) = \frac{1}{\sigma} g\left(\frac{x-u}{\sigma}\right) \ \forall x \in \Re.$$

If the distribution is symmetric, it was also found that

$$\pi_{i:n}(p) = \pi_{n-i+1:n}(1-p) \text{ for } i = 1, \ldots, n. \tag{2.17}$$

Equation 2.17 certainly makes tabulating simultaneous closeness probabilities easier since the probabilities only need to be found for $p \leq 0.50$ and the others can be computed by using the symmetry relation. To demonstrate, the authors derived and calculated SCPs for the normal distribution for various values of n.

For X with a bounded support on the interval (a, b), the authors found the simultaneous closeness probability of $X_{i:n}$ to ξ_p^* to be

$$\pi_{i:n}(p) = n\binom{n-1}{i-2} \int_{a'}^{b^{**}} g(z)[G(z)]^{i-2}\{[G(h_2(z))]^{n-i+1} - [G(h_1(z))]^{n-i+1}\}dz, \tag{2.18}$$

where $b^{**} = min(b', 2z_p - a')$, $h_1(z) = max(a', z)$ and $h_2(z) = min(b', 2z_p - z)$. With $b' = (b-\mu)/\sigma$, $a' = (a-\mu)/\sigma)$, and $\xi_p = (\xi_p^* - \mu)/\sigma$.

The authors derived and calculated SCPs for the exponential distribution for various values of n. They found that the simultaneous closest order statistic and the SCPs to the pth quantile were similar to that of the normal distribution.

Simultaneous closeness probabilities have also been found for progressively Type-II right censored order statistics to population quantiles; this was done in Volterman et al. (2013). For a bounded support, assume there exists Type-II right PCOS from a continuous pdf $f(x)$ and cdf $F(x)$ for a sample of size n with censoring scheme

$R = (R_1, \ldots, R_r)$. For a bounded support, $l = 1, 2, \ldots, r-1$ and fixed quantiles $\xi_p \in (a, b)$, the probability that $X^R_{l+1:r:n}$ is Pitman closer to ξ_p than $X^R_{l:r:n}$ was given as

$$Pr\{A_{l+1:r:n}\} = F^{X^R_{l:r:n}}(\xi_p) - c_{l-1} \sum_{i=1}^{l} a_i(l) \times \int_0^p (1-u)^{\gamma_1 - \gamma_{l+1}^{-1}} [1 - F(min[b, 2\xi_p - F^{-1}(u)])]^{\gamma_{l+1}} du. \quad (2.19)$$

The constants $\gamma_1, \ldots, \gamma_r$ are defined as $\gamma_l = \sum_{i=l}^{r}(R_i + 1) = n - (l-1) - \sum_{i=1}^{l-1} R_i$ for $l = 1, \ldots, r$. Here, γ_l represents the number of remaining units between the $(l-1)$th and lth failures. For simplification of notation, the authors let $c_{l-1} = \prod_{i=1}^{l} \gamma_i$ and $a_i(l) = \prod_{\substack{k=1 \\ k \neq i}}^{l} 1/\gamma_k - \gamma_i$. From Equation 2.19, the authors noted that $F^{X^R_{l:r:n}}(\xi_p)$ depends only on p and the PCS, and does not depend on the underlying distribution F.

Using Equation 2.18 and the results in Balakrishnan et al. (2010), for $l = 2, 3, \ldots, r-1$ and any fixed quantile ξ_p, the SCP of $X^R_{l:r:n}$ to ξ_p, $\pi_{l:r:n}(\xi_p)$, was found to be

$$\begin{aligned}
\pi_{l:r:n}(\xi_p) &= Pr\{B_{l:r:n}\} \\
&= Pr\{A_{l:r:n}\} - Pr\{A_{l+1:r:n}\} \\
&= F^{X^R_{l-1:r:n}}(\xi_p) - F^{X^R_{l:r:n}}(\xi_p) \\
&+ c_{l-1}\sum_{i=1}^{l} a_i(l)\int_0^p (1-u)^{\gamma_1-\gamma_{l+1}+1^{-1}}[1-F(min[b, 2\xi_p - F^{-1}(u)])]^{\gamma_{l+1}}du \\
&- c_{l-2}\sum_{i=1}^{l-1} a_i(l-1)\int_0^p (1-u)^{\gamma_1-\gamma_{l+1}-1}[1-F(min[b, 2\xi_p - F^{-1}(u)])]^{\gamma_l}du.
\end{aligned} \tag{2.20}$$

For the special cases of $l = 1$ and $l = r$, the authors found

$$\pi_{1:r}(\xi_p) = Pr(\mathcal{B}_{1:n}) = 1 - Pr(\mathcal{A}_{2:r}), \tag{2.21}$$

$$\pi_{r:r}(\xi_p) = Pr(\mathcal{B}_{r:n}) = Pr(\mathcal{A}_{r:r}). \tag{2.22}$$

Together, Equations 2.19 - 2.22, give the SCP of PCOS to any population quantile. The authors demonstrated their procedure for the exponential, uniform and normal distributions.

2.1.4.3 Pitman Closeness of Record Values to Population Quantiles

Another type of data considered in the context of Pitman closeness and quantiles is record data. Let $\{X_i, i \geq 1\}$ be a sequence of independent and identically distributed random variables with cdf $F(x)$ and pdf $f(x)$. For every $i < j$, X_j is defined as an upper record if $X_j > X_i$. In other words, X_j is an upper record if its value is

greater than all observations before it. Similarly, lower record values can be defined. Furthermore, the zero-th upper and lower records are given as $U_0 \equiv L_0 \equiv X_1$, and for $n \geq 1$ the nth upper and lower records are taken as U_n and L_n. With this in mind, Ahmadi and Balakrishnan (2009) aimed to study the Pitman closeness of record values to population quantiles. The Pitman closeness probability for any two upper record values U_i and U_j to the parameter ξ_p for $0 < p < 1$ was defined as

$$\pi_U(i, j : p) = Pr(|U_i - \xi_p| < |U_j - \xi_p|), \tag{2.23}$$

which was alternatively written as

$$\pi_U(i, j : p) = \begin{cases} Pr(U_i < \xi_p) + Pr(U_i > \xi_p, U_i + U_j < 2\xi_p) \text{ for } 0 \leq j < i \\ Pr(U_i > \xi_p) + Pr(U_i < \xi_p, U_i + U_j > 2\xi_p) \text{ for } j > i. \end{cases} \tag{2.24}$$

From Equation 2.24, for $0 \leq j < i$, the authors showed that U_i is the Pitman-closest estimator of ξ_p among the class $C_* = \{U_0, U_1, \ldots, U_i\}$ assuming that $Pr(U_i < \xi_p) \geq 0.50$. Similarly, for $j > i$, U_i is Pitman-closest estimator of ξ_p among the class $C^* = \{U_i, U_{i+1}, \ldots, \}$ provided $Pr(U_i > \xi_p) \geq 0.50$. An explicit expression for the probability of $\pi_U(i, j : p)$ was given as

$$\pi_U(i, j : p) = q \sum_{k=0}^{i} \frac{(-\log q)^k}{k!} + \sum_{k=0}^{j-i-1} \sum_{r=o}^{j-i-k-1} \frac{(-1)^k}{i!k!r!(j-i-1)!} C_1(i, k, r; p) \tag{2.25}$$

for $0 \leq i < j$, $p \in (0, 1)$ and $q = 1 - p$, where

$$C_1(i, k, r; p) = \int_0^{-\log q} u^{k+i} \left\{ -\log \bar{F}(-F^{-1}(1 - e^{-u}) + 2\xi_p) \right\}^r \bar{F}(-F^{-1}(1-e^{-u})+2\xi_p) du.$$

For two contiguous upper record values, the Pitman closeness probability was found to be

$$\pi_U(i,i+1;p) = (1-p)\sum_{k=0}^{i}\frac{(-\log q)^k}{k!} + \int_0^{-\log q}\bar{F}(-F^{-1}(1-e^{-u})+2\xi_p)\frac{u^i}{i!}du. \quad (2.26)$$

For the special case when $p = 0.50$, it was found that the probabilities of closeness of upper records to the population median are distribution-free for a parent distribution which is symmetric about $\xi_{0.5}$. For $i \geq 1$, when $p = 0.50$, they found

$$Pr(U_i > \xi_{0.5}) = \frac{1}{2}\sum_{k=0}^{i}\frac{(log2)^k}{k!} > \frac{1}{2} = Pr(U_o = X_1 > \xi_{0,5}).$$

Furthermore, the authors found that U_0 was the Pitman-closest estimator of the population median among the class of all upper record values. Generally, they established that U_i is a Pitman closer estimator of the population median than U_j for $j > i \geq 0$, and for the class $\mathcal{C} = \{U_i, U_{i+1}, U_{i+2}, \dots\}$, U_i is the Pitman-closest estimator of $\xi_{0.5}$. These results were demonstrated for the uniform and exponential distributions.

2.1.4.4 Simultaneous Closeness of k-Records

Another work on Pitman closeness and quantiles is based on k-record data. For this, assume that $X_{1:n}, \dots X_{n:n}$ form a sequence of random variables with cdf $F(x)$ and pdf $f(x)$. Then for a sample of size m, let $X_{i:m}$ denote the ith order statistic. Ahmadi and Balakrishnan (2013) considered the following setting: let $T_{0:k} = k$, $U_{0,k} = X_{1:k}$, and for $n \geq 1$ let $T_{n,k} = min\{j : j > T_{n-1,k}, X_j > X_{T_{n-1,k}-k+1:T_{n-1,k}}\}$.

An upper k-record is defined as the kth largest X that is yet unseen. This sequence of upper k-records is then defined as $U_{n,k} = X_{T_{n,k}-k+1:T_{n,k}}$ for $n \geq 0$, and is known as a Type-2 k-record sequence. A similar definition can be given for lower k-records, denoted by $L_{n,k}$. With this in mind, Ahmadi and Balakrishnan (2013) extended previous work on Pitman closeness to population quantiles and examined the simultaneous closeness probability for record data. The simultaneous closeness probability of $U_{i,k}$ to the parameter of interest θ among the class of upper k-records was expressed as

$$\pi_U(i,k;\theta) = \begin{cases} Pr(|U_{i,k} - \theta| < \min_{j=i-1,i+1} |U_{j,k} - \theta|) & \text{for } i \geq 1 \\ Pr(|U_{0,k} - \theta| < |U_{1,k} - \theta|) & \text{for } i = 0. \end{cases} \tag{2.27}$$

For a sequence of upper k-records $U_{0,k}, U_{1,k}, \ldots$ with bounded support on the interval (a,b), for fixed k and $i \geq 1$, the simultaneous closeness probability of $U_{i,k}$ to θ was shown by the authors to be

$$\pi_U(i,k;\theta) = [\bar{F}(\theta)]^k \frac{[-k \log \bar{F}(\theta)]^i}{i!} + \frac{k^i}{(i-1)!} \int_{-\log \bar{F}(max\{a,2\theta-b|\})}^{-\log \bar{F}(\theta)} \times u^{i-1} [\bar{F}(-F^{-1}(1-e^{-u}) + 2\theta)]^k \left(\frac{ku}{i} - 1\right) du. \tag{2.28}$$

For a sequence with unbounded support, the simultaneous closeness probability of $U_{i,k}$ was found to be

$$\pi_U(i,k;\theta) = k^i \int_0^{-\log \bar{F}(\theta)} \frac{u^{i-1}}{(i-1)!}[\bar{F}(-F^{-1}(1-e^{-u})+2\theta]^k \left(\frac{ku}{i}-1\right) du + [\bar{F}(\theta)]^k \frac{[-k\log \bar{F}(\theta)]^i}{i!}. \quad (2.29)$$

The Pitman closeness probability for the lower k-records was found in a similar manner. Assuming $F(\cdot)$ belongs to a location-scale family and replacing θ with ξ_p in Equation 2.29 gives the simultaneous closeness probability of k-records to population quantiles.

In the case of unbounded support, for fixed k, the SCP of $U_{i,k}$ to the pth quantile ξ_p, for $i \geq 1$, was given as

$$\pi_U(i,k;\xi_p) = \int_0^{-\log q} [\bar{F}(-F^{-1}(1-e^{-u})+2\xi_p]^k \frac{k^i u^{i-1}}{(i-1)!}\left(\frac{ku}{i}-1\right) du + q^k \frac{-k\log q^i}{i!} \quad (2.30)$$

and for $i = 0$,

$$\pi_U(0,k;\xi_p) = q^k + k\int_0^{-\log q} [\bar{F}(-F^{-1}(1-e^{-u}) + 2\xi_p]^k du, \quad (2.31)$$

where $q = 1-p$. Clearly these probabilities depend on p, i, and k but not on location and scale parameters. Assuming F is symmetric about $\xi_{0.5}$, the authors found the probabilities $\pi_U(i,k;\xi_{0.5})$, for all $i \geq 0$, do not depend on F and are given

by

$$\pi_U(i,k;\xi_{0.5}) = \frac{(k\log 2)^i}{i!}\left(2^{-k} - 1 + \frac{k\log 2}{i+1} - k\sum_{r=1}^{k}\binom{k}{r}\frac{(-1)^r}{r2^r}\right)$$
$$+\sum_{r=1}^{k}\binom{k}{r}(-1)^r\left(\frac{k}{r}-1\right)\left(\frac{k}{r}\right)\left(1-\frac{1}{2^r}\sum_{t=0}^{i-1}\frac{(r\log 2)^t}{t!}\right) \quad (2.32)$$

for $i \geq 1$, and for $i = 0$ they found that

$$\pi_U(0,k;\xi_{0.5}) = \frac{1}{2^k} + k\log 2\sum_{r=1}^{k}\binom{k}{r}\frac{(-1)^r}{r}\left(1-\frac{1}{2^r}\right). \quad (2.33)$$

Furthermore, if $U_{i,k}$ is the simultaneous closest k-record to ξ_p among the class of upper k-records, then they found that $L_{i,k}$ is the simultaneously closest k-record to ξ_{1-p} among the class of lower k-records.

2.2 Goodness-of-fit Assessments

2.2.1 Basics

The goodness-of-fit of a statistical model determines how well the model fits a set of observations. It is common to assume that a set of data follows a particular distribution and being able to test the validity of the model assumptions of that particular distribution is desirable. In goodness-of-fit tests, the most important part involves checking for different departures from a set of standard conditions (Huber-Carol, 2002). With this goal, there are both parametric and non-parametric goodness-of-fit tests available. Parametric statistics assume that data come from a specific

probability distribution and inferences about the parameters of the distribution are made. Non-parametric statistics refers to statistics whose sampling distribution does not depend on either the explicit form of the distribution of the population or the values of certain parameters in the distribution of the population (Massey Jr, 1951). Parametric methods typically make more assumptions than non-parametric methods.

The first parametric goodness-of-fit test was introduced by Karl Pearson during the 1900s in a paper that described an objective way of assessing adequacy of fit. This test was the chi-squared test, and the corresponding test statistic was used to compare observed values to theoretical ones (Huber-Carol, 2002). For this test, it is first assumed that underlying probability distribution of the data is multinomial. Then the null hypothesis is that the multinomial probabilities are equal to the hypothesized probabilities, p_i. The test statistic is defined as

$$\chi^2 = \sum_{i=n}^{k} \frac{(x_i - m_i)^2}{m_i},$$

where x_i comes from a multinomial distribution, n is the number of observations, p_i are the hypothesized probabilities and $m_i = np_i$. Here, x_i can be interpreted as the observed counts and m_i the expected numbers or expected cell counts (Cochran, 1952). As n approaches infinity, the test statistic, χ^2, has a chi-squared distribution with $k - 1$ degrees of freedom under the null hypothesis (Cochran, 1952).

A popular non-parametric goodness-of-fit test is the Kolomogorov-Smirnov test, which is based on the maximum difference between an empirical cumulative distribution function and a hypothetical cumulative distribution function (Lilliefors,

1967). The test statistic is

$$d = maximum|F_0(x) - S_n(x))|,$$

where $F_0(x)$ is the specified cumulative frequency distribution from a population and $S_n(x)$ is the cumulative step function of the sample (Massey Jr, 1951). For this test, the null hypothesis is that the sample which comes from an unknown distribution is equal to a common known distribution. If the sample comes from the hypothesized distribution, then d will be small. Over the years, there have been many parametric and non-parametric tests introduced. Some of these include non-parametric tests such as the Wilcoxon signed rank test, the Kruskal-Wallis test and the Mann-Whitney U test, while some parametric tests include t-tests and analysis of variance tests.

2.2.2 Graphical Assessments

Probability plots are used as a graphical procedure of testing the goodness-of-fit of a hypothesized distribution to given data (Arnold et al., 1992). By looking at a plot of two sets of values, an empirical set against a theoretical set or two theoretical data sets, a decision can be reached as to whether there is agreement. The most commonly used plots are the probability-probability plot and the quantile-quantile plot.

2.2.2.1 Probability-Probability Plots

In general, a probability-probability (PP) plot typically has two purposes: it can be used to see if two data sets agree or it can be used to compare a data set to

a theoretical distribution. In the latter case, it is used to determine how well a specific distribution fits the observed data. It does this by comparing an empirical cumulative distribution function to a specified theoretical cumulative distribution function $F(x)$. To do this, the location and scale parameters of $F(x)$ are required in order to evaluate the cdf at the ordered data values; if they are not specified, they need to be estimated. Let $X_1, X_2, \ldots, X_n$ be a random sample from an absolutely continuous distribution with cdf $F(x)$, then a PP plot, more precisely, is a plot of $F(x_{i:n})$ versus p_i, where $x_{1:n} \leq \cdots \leq x_{n:n}$ denote the ordered observations and p_i is a plotting point associated with $x_{i:n}$. If the specified theoretical distribution fits the data well, then the plot will be exhibit a 45 degree line. A PP plot will not remain linear if there are changes to either the location or scale parameters. Furthermore, these type of plots are able to detect discrepancies in the middle of a distribution rather than in the tails. PP plots can also be extended to multivariate situations (Wilk and Gnanadesikan, 1968).

2.2.2.2 Quantile-Quantile Plots

Another visual method of goodness-of-fit is the quantile-quantile (QQ) plot. QQ plots are generally more widely used than PP plots. Introduced by Wilk and Gnanadesikan (1968), a QQ plot compares the quantiles of one probability distribution with similar quantiles of another using a graphical approach. This type of plot can be used to compare collections of data or theoretical distributions. Also, this plot allows one to identify outliers as well as expose location and scale differences (Marden, 2004). Furthermore, it can be used to check whether an assumed linear regression model's errors behave like a random sample from the normal distribution

(Arnold et al., 1992).

Commonly, a QQ plot is used to compare a data set to a theoretical model. For the univariate case, let $X_1, X_2, \ldots X_n$ be a random sample from an absolutely continuous cdf $F(x)$ with location and scale parameters. The observed order statistics, $x_{1:n} \leq \cdots \leq x_{n:n}$ can be thought to represent the values taken by the sample quantile function, $F^{-1}(p)$ (Arnold et al., 1992). The QQ plot is made by plotting the points $(F^{-1}(p_i), x_{i:n})$ where $x_{i:n}$ are ordered observed values of the sample and p_i is again the plotting point associated with $x_{i:n}$. Normally the population quantiles are on the x-axis and the sample quantiles on the y-axis. If the plotted points lie close to a 45 degree line, then this leads to the conclusion that the model fits the data well (Castillo et al., 2005).

A QQ plot has many benefits, for instance, it is a good detector of distributional discrepancies and also provides a useful tool for examining the sufficiency of a composite hypotheses in which there are unspecified location and scale parameters. Unlike PP plots, however, QQ plots cannot easily be applied in multivariate situations (Wilk and Gnanadesikan, 1968). One benefit over the PP plot is that the parameters do not need to be estimated and, in fact, a QQ plot is also unaffected by changes in location or scale. Since the location and scale parameters are not required for this type of plot, it is better at comparing the data distribution with a family of distributions that change only in location or scale.

Like PP plots, QQ plots need plotting points and since there are a multitude of different plotting points to choose from, the best choice usually depends on the purpose of the research as well as the distribution of the variable being considered.

2.2.3 Plotting Points

Plotting points are needed in various goodness-of-fit graphical assessments. The first plotting point introduced was

$$P_i = \frac{i}{n},$$

where i is the rank of the ordered data and n is the number of observations. The problem with such plotting points, however, is that the largest value cannot be used since it corresponds to the 100th percentile of the theoretical distribution. To rectify this, $\pi = \frac{i-1}{n}$ was suggested, however, in this case the smallest value cannot be plotted for similar reasons. Over the years, many different plotting points have been suggested. A distribution-free plotting point was proposed by Weibull (1939) as

$$P_{i:n} = \frac{i}{n+1}. \tag{2.34}$$

For a random variable, X, with pdf $f(x)$ and cdf $F(x)$, then $F(X_{i:n})$ is a new variable related to X by order ranking from the smallest to the largest value and it will have a probability density $f_{i:n}(F(X_{i:n}))$ given by

$$f_{i:n}(F(X_{i:n})) = \frac{n!}{[(i-1)!(n-i)!]}(F(x_{i:n}))^{i-1}(1-F(x_{i:n}))^{n-i}f(x_{i:n}). \tag{2.35}$$

If X has a continuous distribution then $X_{i:n}$ is such that $F(X_{i:n}) \sim B(i, n-i+1)$, where $B(\alpha, \beta)$ is a beta distribution with shape parameters α and β. Based on this,

one can obtain the plotting point in Equation 2.34 which is also known as the *mean-rank* plotting point, denoted $e_{i:n}$. In this way, we see

$$e_{i:n} = \mathbb{E}[F(X_{i:n})] = \frac{i}{n+1}. \tag{2.36}$$

Also based on this, another plotting point that can be used is the median of $F(X_{i:n})$. Let $\mathbf{M}(x)$ denote the median of $F(X_{i:n})$, then the median-rank, $m_{i:n}$ is given by

$$m_{i:n} = \mathbf{M}[F(X_{i:n})] = b_{0.5;i,n-i+1}, \tag{2.37}$$

where $b_{0.5;\alpha,\beta}$ is the median of $B(\alpha, \beta)$. This plotting point is referred to as the *median-rank* plotting point.

The median-rank plotting point is typically used for skewed distributions such as extreme value, since it is thought to be more robust than the mean-rank. Since $F(X_{i:n}) \sim B(i, n-i+1)$, it has been established that

$$e_{i:n} < m_{i:n} \ \forall i < \frac{n}{2} \text{ and } e_{i:n} > m_{i:n}, \ \forall i > \frac{n}{2}.$$

For more details on these plotting points, see Castillo et al. (2005).

The choice of plotting point is a controversial topic. Recently Makkonen (2008) presented the idea that for any analysis of the cdf, observed order statistics must be plotted at $i/(n+1)$ and that the mean of $F(X_{i:n})$ is the best plotting point for extreme value analysis, which contradicts the previous assumption that the median-rank is more robust than the mean-rank. New plotting points have been recently introduced which are based on the Pitman closeness criterion.

A distribution approach that chooses plotting points based on simultaneous closeness probabilities of order statistics to population quantiles was recently done by Balakrishnan et al. (2012a). The authors established that these simultaneous closeness probabilities are independent of the location and scale parameters for location-scale families and just depend on the sample size n as well as p. For a given i and n, the goal is to identify a plotting point for $x_{i:n}$. If one considers SCPs, which are functions of p, one may find a p for which the SCP is maximized. This p, where the SCP is maximized, could be used as a plotting point for that $x_{i:n}$. For $n = 10$ and $n = 15$, Figures 2.1 and 2.2 respectively, plot the SCPs, $\pi_{i:n}$, at different values of p for each order statistic $x_{i:n}$, for $i = 1, \ldots, n$ for the normal distribution. From both Figures 2.1 and 2.2, for $i = 2, \ldots, n-1$, it is clear that optimal plotting position can be found by taking the mode of each curve since for $X_{2:n}, \ldots, X_{n-1:n}$, the curves are unimodal. Also note the SCP plots for the ith and $(n-i+1)$th order statistic are symmetric about $p = 0.50$. Since the normal distribution is symmetric, in order to maximize $\pi_{i:n}(p)$, the partial derivatives of $\pi_{i:n}(p)$ with respect to p were taken and equated to zero. Then, let the solution to this equation be denoted by $s_{i:n}$, which the authors referred to as a SCP plotting points.

In Balakrishnan et al. (2012a), SCP plotting points for values of p ranging over $0.001(0.001)0.999$ for all values of i corresponding to different values of n were first found for the case of the normal distribution. The authors noted that for $n = 10$ and $n = 15$, the simultaneous closeness plotting points were higher than the mean-ranks, which were higher than the median-ranks for larger order statistics. For smaller order statistics this inequality was reversed. Simultaneous closeness plotting points were also found for the logistic, Laplace, and Cauchy distributions for values of

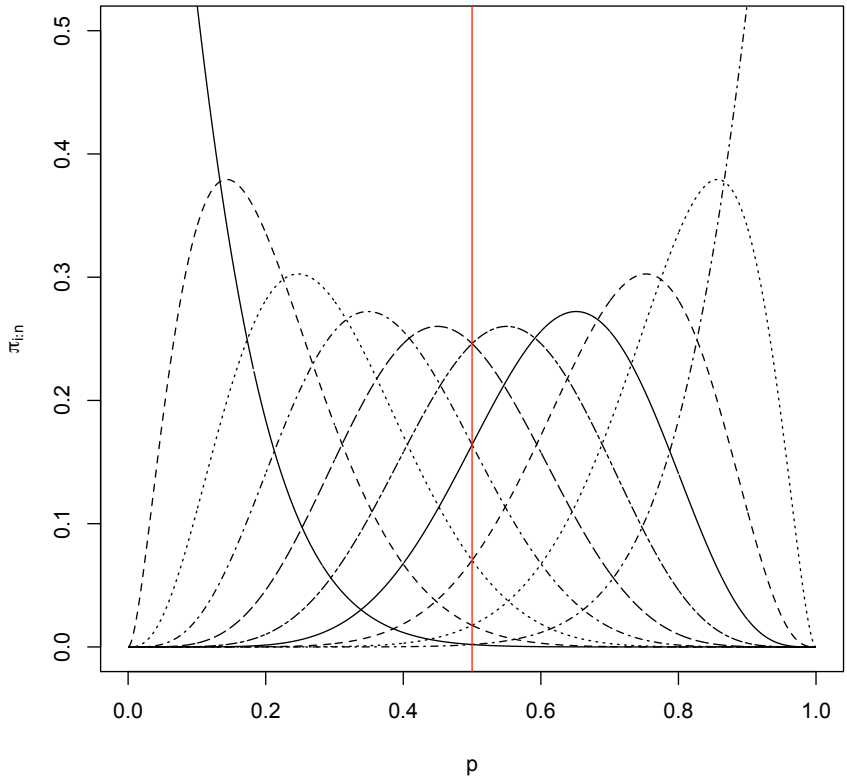

Figure 2.1: SCPs for normal order statistics when $n = 10$

$n = 10$ and $n = 15$. From the tables, it was noted that the uniform distribution had the highest SCP plotting points for the smaller order statistics and lowest for the largest order statistics, this was followed in order by the normal, logistic, Laplace and Cauchy. The Cauchy distribution had the smallest SCP plotting points for the lower order statistics and the highest for the higher order statistics. Distribution-free SCP plotting points were also found using the uniform distribution for samples of size $n = 10$ and $n = 15$.

From Figures 2.1 and 2.2, it should be noted that the plots are not unimodal

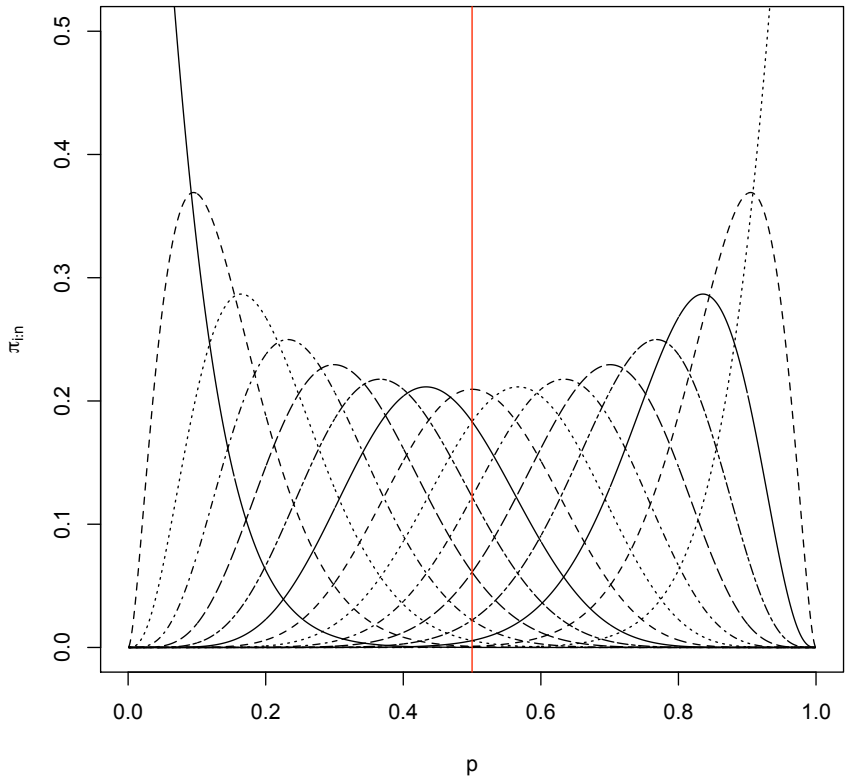

Figure 2.2: SCPs for normal order statistics when $n = 15$

for $i = 1$ and n; they are monotonically decreasing and increasing, respectively. This means that SCP plotting points cannot be found by maximizing the SCP in these two cases. In order to overcome this, the authors used the midpoints between $(0, s_{2:n})$ and $(s_{n-1:n}, 1)$ for the plotting points for $x_{1:n}$ and $x_{n:n}$, respectively.

2.2.4 Correlation Test

One common goodness-of-fit test is the correlation test. The normal probability plot correlation coefficient test was first introduced by Filliben (1975). It was introduced to test for the hypothesis of normality because it was easy to understand and also relatively easy to calculate since it only centers on the linearity of a probability plot. The null hypothesis being tested is that the sample data come from a specific distribution, i.e.,

$$H_o : F(x) = F_0(x)$$

$$H_a : F(x) \neq F_0(x),$$

or equivalently,

$$H_o : \rho \geq \rho_0$$

$$H_a : \rho < \rho_0,$$

where ρ_o is a percentile point for the correlation coefficient under the null distribution $F_0(x)$. Let $x_{i:n} \leq \cdots \leq x_{n:n}$ be the observed order statistics from a sample of size n. In its original form, the normal probability plot correlation coefficient test

computes the product moment correlation coefficient between the ordered observations $x_{i:n}$ and the median-ranks $\mathbf{M}(x_{i:n})$ (Filliben, 1975), where the observations come from the standard normal distribution. The product moment correlation coefficient was introduced by Karl Pearson in 1895 and can be written as

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}}.$$

A product moment correlation coefficient being close to one implies that the normal probability plot of $x_{i:n}$ against the median-ranks, $\mathbf{M}(x_{i:n})$, is roughly linear and that the sample was generated from the hypothesized normal distribution. The distribution of r depends only on the sample size n and the standardized cdf; it does not depend on either location or scale parameters. Due to the simplicity of this test, some advantages of this test include that it is not limited to any sample size and that it can be easily extended to other distributions besides the normal distribution (Filliben, 1975).

Based on Filliben's work, Kinnison (1989) developed a correlation coefficient type test for a Type-I extreme value distribution, the Gumbel distribution. For this test, the correlation coefficient was calculated between the ordered data values and their mean-ranks. The mean-ranks were found by substituting the rank percentile of a data value, where the rank percentiles were calculated as the rank of each data divided by the sample size plus one, into the inverse of the extreme value cumulative distribution function (Kinnison, 1989). Through simulation, the empirical distribution of the statistic was found and was used to get the critical values at different percentage points. Kinnison also examined the power properties of this

test for alternative models. The alternate distributions chosen were the Weibull and Cauchy distributions since they are a part of extreme value family. The normal and log-normal distributions were also considered. Several different sample sizes were used for each distribution and it was found that the test had good power to reject samples from the alternate distributions. It was also found that as the sample size increased, the power of the test also did Kinnison (1989). This test is useful since it is one of the easiest available goodness-of-fit tests for the extreme value distribution.

2.2.5 Correlation Coefficient-Type Test Based on SCPs

Following along the lines of Kinnison, a correlation goodness-of-fit was proposed for the test of normality using the new SCP plotting positions introduced by Balakrishnan et al. (2012a) and described in Section 2.2.3. For observed order statistics $x_{1:n} \leq \cdots \leq x_{n:n}$, let $s_{1:n}, \ldots, s_{n:n}$ be the SCP plotting points. Since optimal plotting points could not be found for smallest and largest order statistics, as previously mentioned, the authors used the midpoint between $(0, s_{2:n})$ for $x_{1:n}$ and the midpoint between $(s_{n-1:n}, 1)$ for $x_{n:n}$. First proposed as a test for normality, the tests considers the correlation between the values $x_{i:n}$ and $\Phi^{-1}(s_{i:n})$ for $i = 1, \ldots, n$ where Φ is the standard cdf of the standard normal distribution. For this test, small values of the correlation coefficient indicate the rejection of the hypothesis of normality. Monte Carlo simulations were used to find the average value, variance and both 5% and 10% critical values of the new test statistic for sample sizes of $n = 10, 15$ and 25. The power of the test for these sample sizes was found using Monte Carlo simulations for various alternatives such as the exponential and Cauchy. The purpose of this power study was to evaluate the proposed correlation goodness-of-fit

test for normality. The authors also compared the performance of the test when other plotting points were used, such as distribution-free, mean- and median-rank plotting points. These comparisons showed that the power of the test based on the distribution-free SCP plotting method and median-rank method were not as good as the SCP plotting method based on the normal. An advantage of the SCP plotting method is that optimal plotting points can be found for other distributions. The authors found that this could result in the possibility of better power properties and better plotting methods for that model compared to the mean- and median-ranks.

This work was followed by a correlation-type goodness-of-fit test for the extreme value distribution based on simultaneous closeness probabilities. To test for extreme value, $G^{-1}(s_{i:n}) = log(-log(1-s_{i:n})$ is used in place of the inverse cdf of the standard normal distribution, where $s_{i:n}$ now denotes the SCP plotting points for the extreme value distribution. Since SCP plotting points once again cannot be found for $i = 1$ and n, the authors again took the midpoints of the intervals $(0, s_{2:n})$ and $(s_{n-1:n}, 1)$ which were used for $x_{1:n}$ and $x_{n:n}$, respectively. By using Monte Carlo simulation, the distribution of the correlation coefficient statistic was estimated, as well as the critical values of its distribution. Also, the distribution of the correlation coefficient was estimated based on both the mean-rank plotting points and the median-rank plotting points for the sake of comparison.

A power study comparing the proposed test with the correlation coefficient test based on the mean- and median-ranks was then conducted to judge the performance of the goodness-of-fit test for extreme value distribution based on SCP plotting points. Samples were simulated from distributions such as exponential, gamma, lognormal, and Students t, and the power of the test was determined based on

100,000 simulations at significance levels of 5% and 10%. The authors found that the test based on the SCP plotting points gave consistently better power than the other methods. It was also found, in particular, that the test based on SCP plotting points performed better than the median-ranks for small sample sizes. As n increased, the power of the tests based on the median-ranks and SCP plotting points became very similar in all cases.

The authors also demonstrated the use of their test in the form of a Weibull analysis. Suppose that an experimenter wants to know if a particular set of data comes from the Weibull(θ, α) distribution. To estimate the location and scale parameters, a least-squares regression could be used on a QQ plot. Based on this idea, the correlation coefficient test could be used. In order to use the correlation coefficient test, the log of the data is taken so it can be transformed to data from the extreme value distribution with location parameter $\log(\theta)$ and scale parameter $1/\alpha$. For an ordered sample $Y_{1:n} \leq \cdots \leq Y_{n:n}$ from the extreme value distribution, it can be seen that

$$\frac{Y_{i:n} - \log(\theta)}{1/\alpha} \approx \log(-\log(1 - a_{i:n})) \tag{2.38}$$

which implies $Y_{i:n} \approx \log(\theta) + \frac{1}{\alpha}[\log(-\log(1 - a_{i:n}))]$, where $a_{i:n}$ is a plotting point associated with $Y_{i:n}$. By substituting the mean-rank, median-rank and SCP plotting points for $a_{i:n}$, the authors were able to get the least-squares estimates of the Weibull parameters. They then computed the bias and mean square error of these estimates by using $10,000$ simulated Weibull samples of size $n = 10, 40$ and 50 for all three methods. They found that the estimates of θ based on the SCP plotting points and median-ranks were very near unbiased and became almost identical as n increased.

For the estimate of α, the mean-rank method ended up being the best based on the mean squared error.

Chapter 3

Proposed Test

3.1 Motivation

In order to assess goodness-of-fit, QQ plots comparing an observed sample to a theoretical distribution can be used. QQ plots do this by comparing the quantiles of the two probability distributions and checking for linearity. In order to use these type of plots, plotting points are needed. A new plotting point was introduced by Balakrishnan et al. (2011a) that is based on simultaneous closeness probabilities. These SCP plotting points were the basis for the correlation coefficient test described in Section 2.2.5. The correlation coefficient test involves finding the correlation between a sample data and the inverse cdf of the hypothesized model evaluated at the corresponding plotting points. If the correlation is close to one, it indicates the linearity and it can be said that the data come from the hypothesized distribution. One of the drawbacks of the correlation test using SCP plotting points is that SCP plotting points for $x_{1:n}$ and $x_{n:n}$ had to be interpolated. That is, the midpoint between 0 and $s_{2:n}$ was used for $x_{1:n}$ and similarly for $x_{n:n}$, the midpoint between $s_{n-1:n}$

and 1 was used. While this provides a valid test, we now consider modifications to this procedure.

3.2 Construction

We propose a minimal-maximal correlation-type goodness-of-fit test, which will have two forms: a minimal-correlation test and a maximal-correlation test. In particular, in conjunction with typical SCP plotting points, we will consider a space of SCP plotting points $(0, s_{2:n}) \times (s_{n-1:n}, 1)$ for $x_{1:n}$ and $x_{n:n}$, and identify where the correlation coefficient is minimized (maximized). Both tests can be used to determine if the model comes from a specific distribution.

Considering the hypotheses given in Section 2.2.4, under H_0, the largest correlation that can be achieved with a given data set should be high. Therefore, for the test which identifies maximum correlation, which we shall call the *maximal-correlation* test, if the maximum correlation test statistic is below the critical value, then the null hypothesis is rejected. This implies that the data does not come from the proposed distribution, i.e., there is a lack-of-fit. If the test statistic is above the critical value then we have insufficient evidence to conclude that the data come from the alternative distribution. It is set up in this way because the maximum correlation needs to be high enough to be a good fit.

For a given data set, under the null hypothesis, the lowest correlation that can be obtained should not be too low. Therefore, for the test which determines minimum correlation, which we shall call the *minimal-correlation* test, if the minimum correlation is high then it suggests that the data come from the hypothesized model. That is, if the correlation test statistic is lower than some critical value then it

can be concluded that the null hypothesis should be rejected and the hypothesized model is not a good fit. In order to investigate these two tests, we considered two approaches, which we will call the grid search method and the averaging method.

For a given data set, the first method consists of a grid search which finds where the correlation value is minimized (maximized). For $x_{2:n}, \ldots, x_{n-1:n}$ the SCP plotting points $s_{2:n}, \ldots, s_{n:n}$ found by Balakrishnan et al. (2012a) are used. Since we do not have optimal plotting points for $x_{1:n}$ and $x_{n:n}$ (which would be 0 and 1, respectively), we consider a range of the plotting point for $x_{1:n}$ that is from $(0.0001, s_{2:n})$ and similarly the range considered for $x_{n:n}$ is from $(s_{n-1:n}, 0.9999)$, thus giving a space of possible values defined by $(0, s_{2:n}) \times (s_{n-1:n}, 1)$. The correlation is then found between the sample data and inverse cdf of the hypothesized distribution evaluated at each point in this grid and the known plotting points $s_{i:n}$. For instance, for a given data set, $x_{1:n}, \ldots x_{n:n}$, the first correlation evaluated is between $(x_{1:n}, \ldots, x_{n:n})$ and $(0.0001, s_{2:n}, \ldots, s_{n-1:n}, s_{n-1:n})$ and the last one is evaluated between $(x_{1:n}, \ldots, x_{n:n})$ and $(s_{2:n}, s_{2:n}, \ldots, s_{n-1:n}, 0.9999)$. This will create a grid of correlation values. The grid is then searched for the highest correlation value which corresponds to the maximal-correlation test statistic and the lowest correlation value which corresponds to the minimal-correlation test statistic. The so found plotting points corresponding to minimal-correlation test will be denoted as $s^*_{1:n}$ and $s^*_{n:n}$ for $x_{1:n}$ and $x_{n:n}$ respectively. For the maximal-correlation test, the plotting points for $x_{1:n}$ and $x_{n:n}$ will be denoted to as $s^{**}_{1:n}$ and $s^{**}_{n:n}$. In the remainder of this thesis, we will refer to these new plotting points as *modified SCP plotting points*. Note, for both tests, we use the traditional SCP plotting points, $s_{i:n}$, for $i = 2, \ldots, n-1$, i.e., $s^*_{i:n} = s^{**}_{i:n} = s_{i:n}$ for $i = 2, \ldots, n-1$. To make this procedure more clear, consider

the following data set:

0.5465	0.2021	0.6995	0.0099	0.2293	0.7782	0.6547	0.3516	0.8880	0.1313

Suppose we wish to test for the uniform distribution. Then the procedure entails evaluating the correlation between these ordered data values and all sets of plotting points of the form $(0.0001, 0.1584, \ldots, 0.8416)$, where the first and last entries are increased in their support over the space $(0, s_{2:n}) \times (s_{n-1}, 1)$. The points $0.1584, \ldots, 0.8416$ are the SCP plotting points found for $s_{2:10}, \ldots, s_{9:10}$ by Balakrishnan et al. (2010). This gives a matrix of correlation values and these are plotted in Figure 3.1. This visualizes our objective function which we can minimize and maximize. From Figure 3.1 it can be seen that the minimal correlations occur at the boundaries of the space, i.e., 0.0001 and 0.1584 for $x_{1:10}$ and 0.8416 and 0.9999 for $x_{10:10}$. The maximal correlation occurs in the middle of the space.

The second method does not require a search for a given data set, but instead it finds the maximal (minimal) correlation using the averages of the plotting points that were found in the first method, i.e., it uses the averages of $s^*_{1:n}$, $s^*_{n:n}$, $s^{**}_{1:n}$ and $s^{**}_{n:n}$. We will denote these by $\bar{s}^*_{1:n}, \bar{s}^*_{n:n}, \bar{s}^{**}_{1:n}$ and $\bar{s}^{**}_{n:n}$. In other words, this method uses plotting points which are not data dependent. The test proceeds in the same fashion; it finds the correlation between the ordered data and the inverse cdf of the hypothesized distribution evaluated at the modified SCP plotting points.

3.3 Numerical Results

In this section, we compare the two tests across both methods by looking at their distributional properties, carrying out a power study and considering several illus-

trative examples.

3.3.1 Distributional Properties

Through simulation we investigated the distribution of our four test statistics for both the uniform$(0, 1)$ and standard normal for both methods. For the grid search method, uniform$(0, 1)$ random observations were simulated for samples of size, $n = 10, 15, 20$ and 30. Within each sample, the correlation was found between the ordered data and the inverse cdf evaluated at points in the space considered. These values were then stored in a matrix for each of the four sample sizes. A grid search was then conducted by scanning each of the four matrices for the highest and lowest correlation, thus finding the set of plotting points at which each occurred, and identifying maximum and minimum correlation values. This grid search method was repeated $10,000$ times. The 5% and 10% critical values of the test statistic were then found by ordering both the maximum and minimum correlation values for each sample size and finding the 500th and 1000th value. For further information, the number of times that the four new plotting points were the same as their corresponding boundary values was calculated and stored, i.e., for $s^*_{1:n}$ it was found how many times that it was equal to either 0.0001 or $s_{2:n}$. This procedure was repeated for the standard normal distribution. The average SCP plotting points to be used in the second method, i.e., $\bar{s}^*_{1:n}$, $\bar{s}^*_{n:n}$, $\bar{s}^{**}_{1:n}$ and $\bar{s}^{**}_{n:n}$, can be seen in Tables 3.1 and 3.2. These two tables compare the previously used plotting points for $x_{1:n}$ and $x_{n:n}$ and our modified SCP plotting points. It can be noted from these two tables that as n increases, the SCP plotting points for the minimal- and maximal-test become quite different from the SCP plotting points using the midpoint method, though

are very similar for small values such as $n = 10$. The critical values found for the test statistics for the distributions considered can be seen in Tables 3.3 and 3.6.

For the uniform distribution with $n = 10$, from Table 3.7, it can be seen that for the minimal-correlation test, the bounds are being hit 100% of the time. This was the case for all the sample sizes considered. From Table 3.8 we can see this pattern did not continue for the normal distribution. It can be seen from Table 3.8 that the proportions for $s^*_{n:n}$ and $s^{**}_{n:n}$ occurring at the boundary values total to one. For the normal distribution, the proportion of times the plotting points $s^{**}_{1:n}$ occurs at the bound 0.0001 seems to decrease as n increases, but for the plotting point $s^*_{1:n}$ it increases as n increases. This is also the case for the plotting point $s^{**}_{n:n}$; the proportion of times it hits the bound 0.9999 decreases as n increases but increases for $s^*_{n:n}$ as n increases. For $s^{**}_{1:n}$ and $s^{**}_{n:n}$, at the bounds $s_{2:n}$ and $s_{n-1:n}$ respectively, the proportion increases as n increases but the reverse is true for $s^*_{1:n}$ and $s^*_{n:n}$. In the case of the uniform distribution, all the proportions increase for both $s^{**}_{1:n}$ and $s^{**}_{n:n}$. However for $s^*_{1:n}$ and $s^*_{n:n}$ the proportions decrease as n increases for the bounds $s_{2:n}$ and $s_{n-1:n}$, but increase for the bounds 0.0001 and 0.9999.

The second method took the average of the SCP plotting points from the grid search method and used them as plotting points for $x_{1:n}$ and $x_{n:n}$ for both tests. The maximum correlation was found between the ordered data values and the average SCPs found for the maximal test in the grid search method, and the minimum correlation was found between the ordered data and the average SCPS found the for the minimal test. This procedure was repeated $10,000$ times. Critical values were found at the 5% and 10% level; these results can be seen in Tables 3.4 and 3.5.

Histograms of the correlation statistic were made for both methods for the max-

imal and minimal correlation tests for the two distributions considered. Figure 3.2 -3.9 show that for both methods, the majority of correlation values were between 0.95 and 1 for the uniform distribution. These results were very similar for the normal distribution. The histograms also show that the distribution of the correlation statistic is always skewed to the left in both tests across both methods.

3.3.2 Power Study

The power of a test is the probability that the test will reject the null hypothesis when the null hypothesis is false. The power of the proposed tests was investigated by replacing the uniform (normal) distribution with an alternative distribution in the simulations. For a given alternative, 10,000 random samples were generated for sample sizes $n = 10, 15, 20$ and 30, and the correlation was found using the modified SCP plotting points. The proportion of the correlations that were less than the critical values was determined; this is the power of the test at the associated level. The alternative distributions chosen for this power study were the $B(1, 3)$, $B(3, 1)$, $B(3, 3)$, $B(2, 1)$ and $B(2, 2)$. The choice of significance levels were 5% and 10%. The results are presented in Tables 3.9 and 3.10. Table 3.9 indicates that all four correlation tests for the uniform had good power to reject samples from the majority of alternative distributions considered; this is more evident as n increases. The averaging method appears to have consistently better power than the grid search method for the maximal-correlation test in all cases. For the minimal-correlation test, the grid search method had better power than the averaging method for all alternative distributions for $n = 15, 20$ and 30. In terms of the two tests, the minimal-correlation test had higher power in almost all cases. For the normal, the

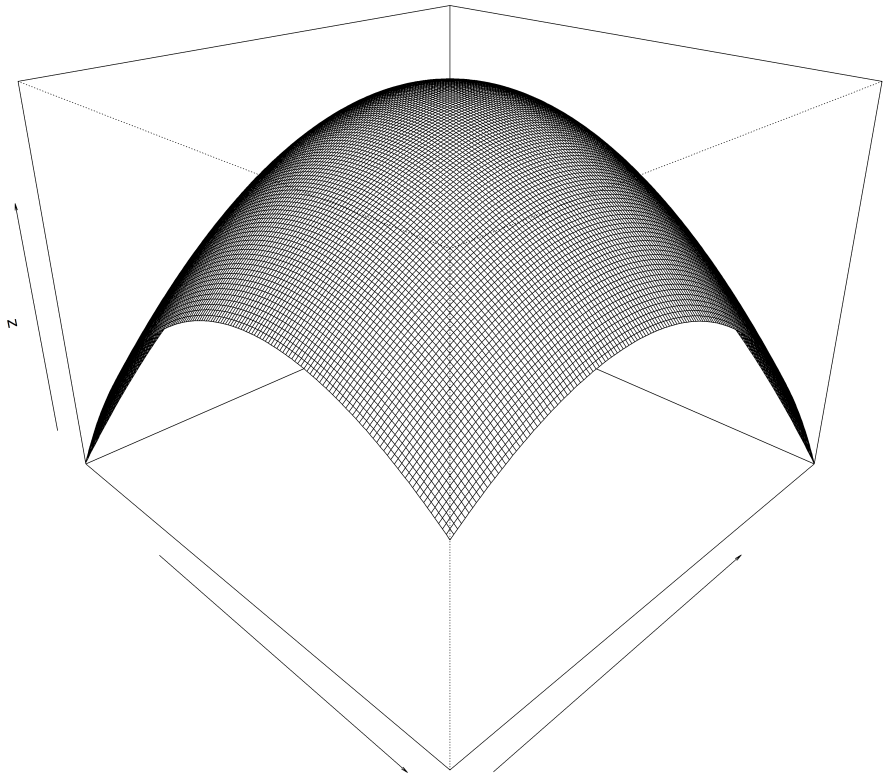

Figure 3.1: Plot of the correlation coefficient over a grid of plotting points for $x_{1:n}$ and $x_{n:n}$

i/n	10			15			20			30		
	$s_{i:n}$	$\bar{s}^{**}_{i:n}$	$\bar{s}^{*}_{i:n}$	$s_{i:n}$	$\bar{s}^{**}_{i:n}$	$\bar{s}^{*}_{i:n}$	$s_{i:n}$	$\bar{s}^{**}_{i:n}$	$\bar{s}^{*}_{i:n}$	$s_{i:n}$	$\bar{s}^{**}_{i:n}$	$\bar{s}^{*}_{i:n}$
1	0.0792	0.0779	0.0847	0.0529	0.0516	0.0554	0.0398	0.0385	0.0415	0.0265	0.0255	0.0277
2	0.1584	0.1584	0.1584	0.1058	0.1058	0.1058	0.0795	0.0795	0.0795	0.0530	0.0530	0.0530
3	0.2542	0.2542	0.2542	0.1699	0.1699	0.1699	0.1275	0.1275	0.1275	0.0851	0.0851	0.0851
4	0.3521	0.3521	0.3521	0.2353	0.2353	0.2353	0.1766	0.1766	0.1766	0.1178	0.1178	0.1178
5	0.4506	0.4506	0.4506	0.3012	0.3012	0.3012	0.2261	0.2261	0.2261	0.1508	0.1508	0.1508
6	0.5494	0.5494	0.5494	0.3674	0.3674	0.3674	0.2758	0.2758	0.2758	0.1840	0.1840	0.1840
7	0.6479	0.6479	0.6479	0.4337	0.4337	0.4337	0.3256	0.3256	0.3256	0.2172	0.2172	0.2172
8	0.7458	0.7458	0.7458	0.5000	0.5000	0.5000	0.3754	0.3754	0.3754	0.2504	0.2504	0.2504
9	0.8416	0.8416	0.8416	0.5663	0.5663	0.5663	0.4252	0.4252	0.4252	0.2837	0.2837	0.2837
10	0.9208	0.9225	0.9149	0.6326	0.6326	0.6326	0.4751	0.4751	0.4751	0.3169	0.3169	0.3169
11	-	-	-	0.6988	0.6988	0.6988	0.5249	0.5249	0.5249	0.3502	0.3502	0.3502
12	-	-	-	0.7647	0.7647	0.7647	0.5748	0.5748	0.5748	0.3835	0.3835	0.3835
13	-	-	-	0.8301	0.8301	0.8301	0.6246	0.6246	0.6246	0.4168	0.4168	0.4168
14	-	-	-	0.8942	0.8942	0.8942	0.6744	0.6744	0.6744	0.4501	0.4501	0.4501
15	-	-	-	0.9471	0.9484	0.9445	0.7242	0.7242	0.7242	0.4834	0.4834	0.4834
16	-	-	-	-	-	-	0.7739	0.7739	0.7739	0.5166	0.5166	0.5166
17	-	-	-	-	-	-	0.8234	0.8234	0.8234	0.5499	0.5499	0.5499
18	-	-	-	-	-	-	0.8725	0.8725	0.8725	0.5832	0.5832	0.5832
19	-	-	-	-	-	-	0.9205	0.9205	0.9205	0.6165	0.6165	0.6165
20	-	-	-	-	-	-	0.9603	0.9610	0.9591	0.6498	0.6498	0.6498
21	-	-	-	-	-	-	-	-	-	0.6831	0.6831	0.6831
22	-	-	-	-	-	-	-	-	-	0.7163	0.7163	0.7163
23	-	-	-	-	-	-	-	-	-	0.7496	0.7496	0.7496
24	-	-	-	-	-	-	-	-	-	0.7828	0.7828	0.7828
25	-	-	-	-	-	-	-	-	-	0.8160	0.8160	0.8160
26	-	-	-	-	-	-	-	-	-	0.8492	0.8492	0.8492
27	-	-	-	-	-	-	-	-	-	0.8822	0.8822	0.8822
28	-	-	-	-	-	-	-	-	-	0.9149	0.9149	0.9149
29	-	-	-	-	-	-	-	-	-	0.9470	0.9470	0.9470
30	-	-	-	-	-	-	-	-	-	0.9735	0.9743	0.9724

Table 3.1: SCP plotting points for the uniform distribution using the averaging method

i/n	10			15			20			30		
	$s_{i:n}$	$\bar{s}^{**}_{i:n}$	$\bar{s}^{*}_{i:n}$	$s_{i:n}$	$\bar{s}^{**}_{i:n}$	$\bar{s}^{*}_{i:n}$	$s_{i:n}$	$\bar{s}^{**}_{i:n}$	$\bar{s}^{*}_{i:n}$	$s_{i:n}$	$\bar{s}^{**}_{i:n}$	$\bar{s}^{*}_{i:n}$
1	0.0717	0.0671	0.0717	0.0476	0.0450	0.0457	0.0356	0.0349	0.0298	0.0237	0.0238	0.0159
2	0.1433	0.1433	0.1433	0.0951	0.0951	0.0951	0.0712	0.0712	0.0712	0.0473	0.0473	0.0473
3	0.2472	0.2472	0.2472	0.1644	0.1644	0.1644	0.1231	0.1231	0.1231	0.0820	0.0820	0.0820
4	0.3487	0.3487	0.3487	0.2320	0.2320	0.2320	0.1738	0.1738	0.1738	0.1158	0.1158	0.1158
5	0.4496	0.4496	0.4496	0.2992	0.2992	0.2992	0.2242	0.2242	0.2242	0.1494	0.1494	0.1494
6	0.5504	0.5504	0.5504	0.3662	0.3662	0.3662	0.2745	0.2745	0.2745	0.1829	0.1829	0.1829
7	0.6513	0.6513	0.6513	0.4331	0.4331	0.4331	0.3246	0.3246	0.3246	0.2164	0.2164	0.2164
8	0.7528	0.7528	0.7528	0.5000	0.5000	0.5000	0.3748	0.3748	0.3748	0.2497	0.2497	0.2497
9	0.8567	0.8567	0.8567	0.5669	0.5669	0.5669	0.4249	0.4249	0.4249	0.2831	0.2831	0.2831
10	0.9208	0.9342	0.9271	0.6338	0.6338	0.6338	0.4750	0.4750	0.4750	0.3165	0.3165	0.3165
11	-	-	-	0.7008	0.7008	0.7008	0.5250	0.5250	0.5250	0.3499	0.3499	0.3499
12	-	-	-	0.7680	0.7680	0.7680	0.5751	0.5751	0.5751	0.3832	0.3832	0.3832
13	-	-	-	0.8356	0.8356	0.8356	0.6252	0.6252	0.6252	0.4166	0.4166	0.4166
14	-	-	-	0.9049	0.9049	0.9049	0.6754	0.6754	0.6754	0.4500	0.4500	0.4500
15	-	-	-	0.9525	0.9545	0.9549	0.7255	0.7255	0.7255	0.4833	0.4833	0.4833
16	-	-	-	-	-	-	0.7758	0.7758	0.7758	0.5167	0.5167	0.5167
17	-	-	-	-	-	-	0.8262	0.8262	0.8262	0.5500	0.5500	0.5500
18	-	-	-	-	-	-	0.8769	0.8769	0.8769	0.5834	0.5834	0.5834
19	-	-	-	-	-	-	0.9288	0.9288	0.9288	0.6168	0.6168	0.6168
20	-	-	-	-	-	-	0.9644	0.9656	0.9695	0.6501	0.6501	0.6501
21	-	-	-	-	-	-	-	-	-	0.6835	0.6835	0.6835
22	-	-	-	-	-	-	-	-	-	0.7169	0.7169	0.7169
23	-	-	-	-	-	-	-	-	-	0.7503	0.7503	0.7503
24	-	-	-	-	-	-	-	-	-	0.7836	0.7836	0.7836
25	-	-	-	-	-	-	-	-	-	0.8171	0.8171	0.8171
26	-	-	-	-	-	-	-	-	-	0.8506	0.8506	0.8506
27	-	-	-	-	-	-	-	-	-	0.8842	0.8842	0.8842
28	-	-	-	-	-	-	-	-	-	0.9180	0.9180	0.9180
29	-	-	-	-	-	-	-	-	-	0.9527	0.9527	0.9527
30	-	-	-	-	-	-	-	-	-	0.9868	0.9766	0.9841

Table 3.2: SCP plotting points for the normal distribution using the averaging method

	Minimal		Maximal	
n	5%	10%	5%	10%
10	0.7287	0.7548	0.9607	0.9689
15	0.8064	0.8251	0.9666	0.9729
20	0.8510	0.8648	0.9710	0.9765
30	0.8990	0.9079	0.9771	0.9814

Table 3.3: Critical values of the correlation statistics using the grid search method for normal data

	Minimal		Maximal	
n	5%	10%	5%	10%
10	0.9084	0.9285	0.9109	0.9301
15	0.9355	0.9480	0.9355	0.9481
20	0.9504	0.9589	0.9489	0.9598
30	0.9650	0.9712	0.9630	0.9701

Table 3.4: Critical values of the correlation statistics using the averaging method for normal data

	Minimal		Maximal	
n	5%	10%	5%	10%
10	0.9164	0.9334	0.9174	0.9341
15	0.9438	0.9545	0.9439	0.9546
20	0.9563	0.9653	0.9564	0.9653
30	0.9704	0.9765	0.9704	0.9765

Table 3.5: Critical values of the correlation statistics using the averaging method for uniform data

	Minimal		Maximal	
n	5%	10%	5%	10%
10	0.8802	0.9036	0.9362	0.9501
15	0.9307	0.9442	0.9510	0.9613
20	0.9507	0.9598	0.9612	0.9687
30	0.9682	0.9747	0.9719	0.9779

Table 3.6: Critical values of the correlation statistics using the grid search method for uniform data

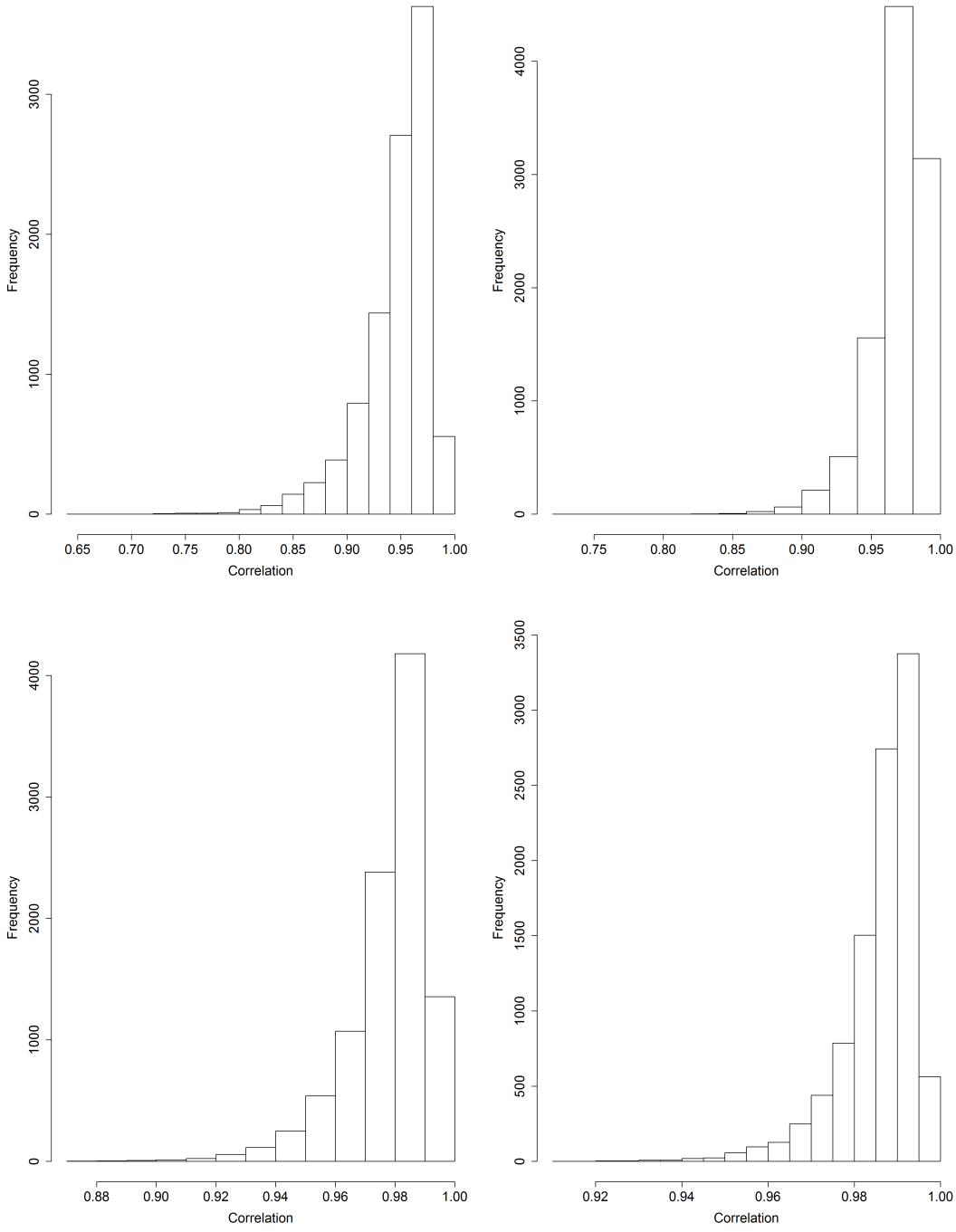

Figure 3.2: Histograms of the minimal-correlation statistic found using the grid search method for uniform(0,1): on the top row from left to right are the plots for $n = 10$ and 15; on the bottom row from left to right are the plots for $n = 20$ and 30.

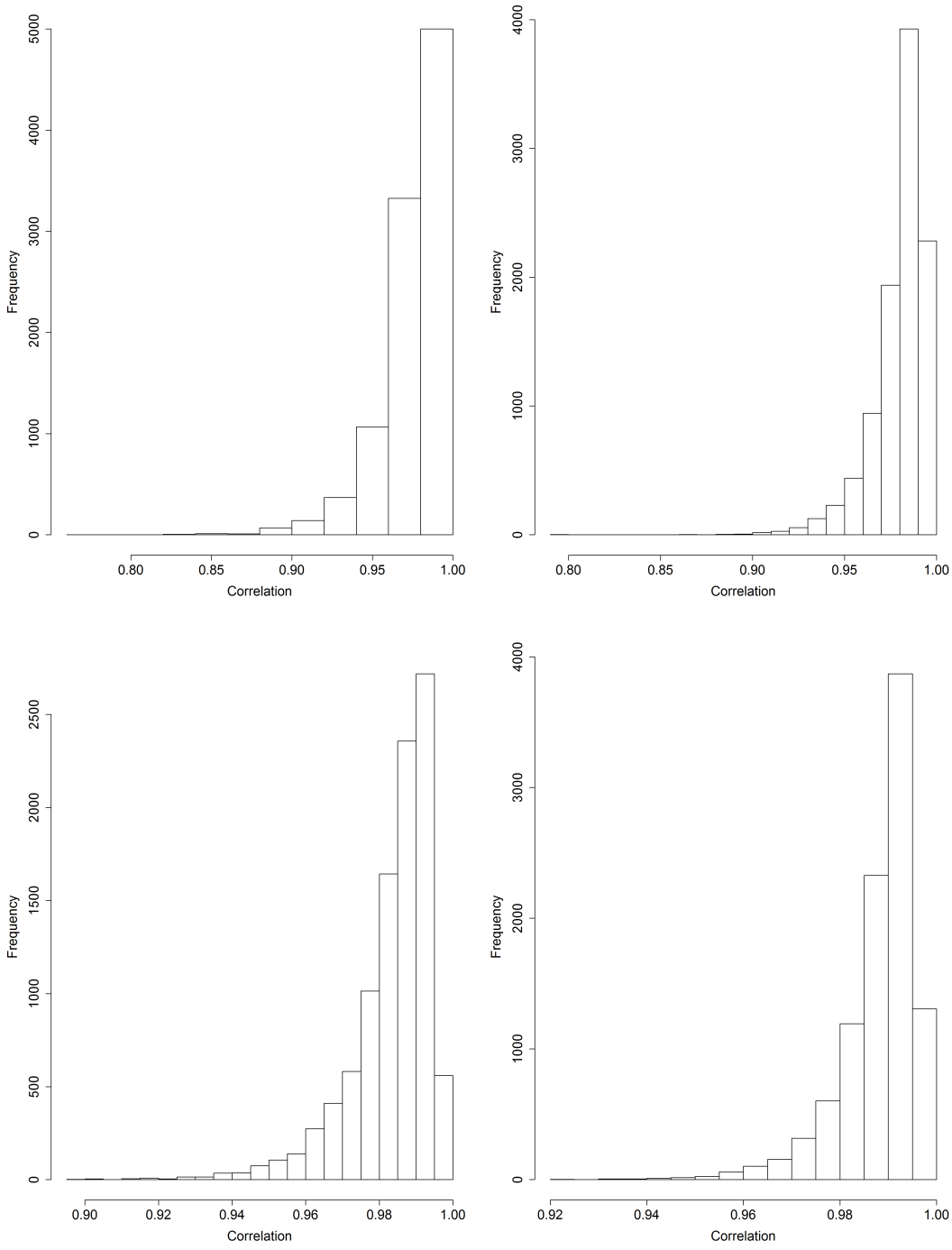

Figure 3.3: Histograms of the maximal-correlation statistic found using the grid search method for uniform(0,1): on the top row from left to right are the plots for $n = 10$ and 15; on the bottom row from left to right are the plots for $n = 20$ and 30.

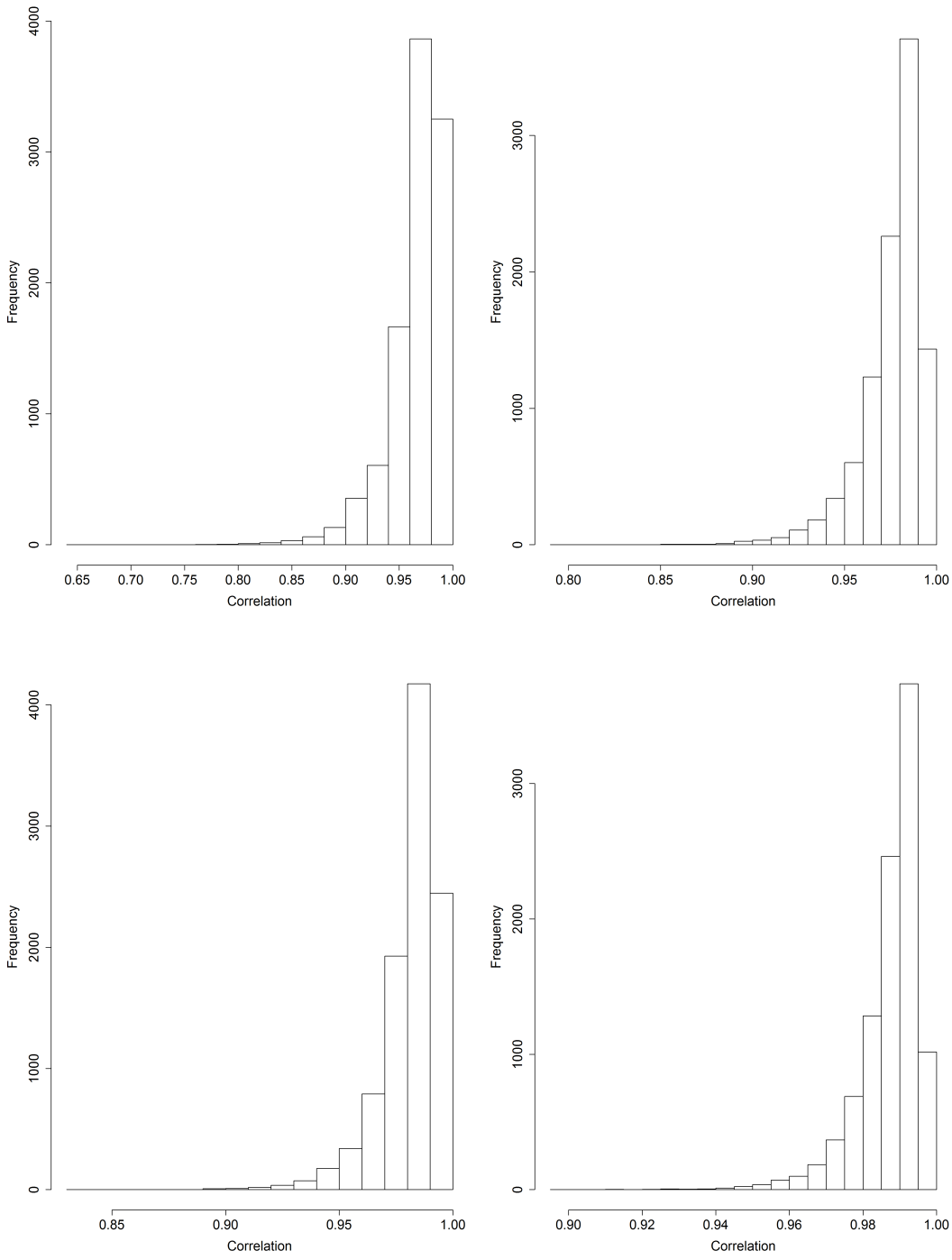

Figure 3.4: Histograms of the minimal-correlation statistic found using the averaging method for uniform(0,1): on the top row from left to right are the plots for $n = 10$ and 15; on the bottom row from left to right are the plots for $n = 20$ and 30.

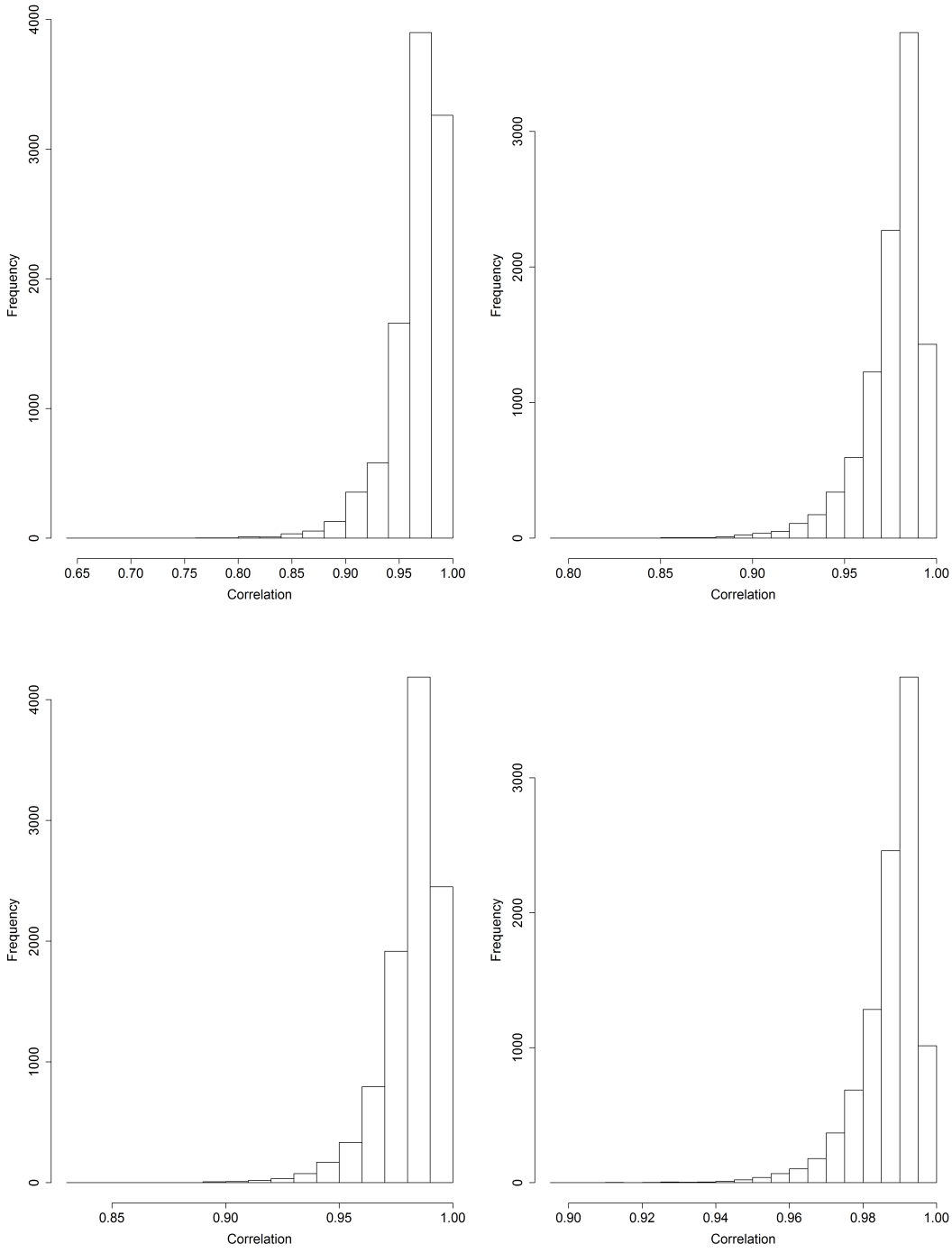

Figure 3.5: Histograms of the maximal-correlation statistic found using the averaging method for uniform(0,1): on the top row from left to right are the plots for $n = 10$ and 15; on the bottom row from left to right are the plots for $n = 20$ and 30.

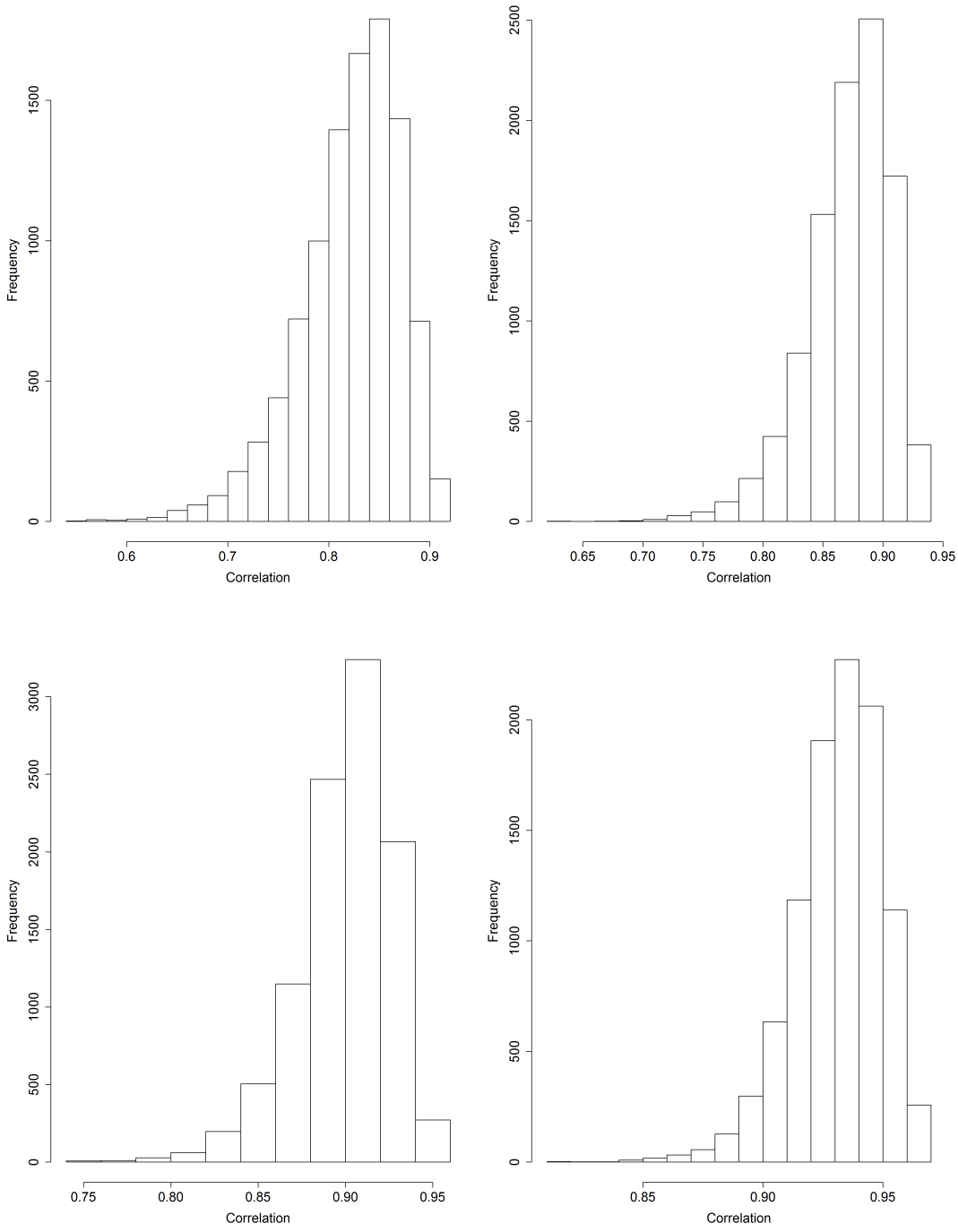

Figure 3.6: Histograms of the minimal-correlation statistic found using the grid search method for the standard normal distribution: on the top row from left to right are the plots for $n = 10$ and 15; on the bottom row from left to right are the plots for $n = 20$ and 30.

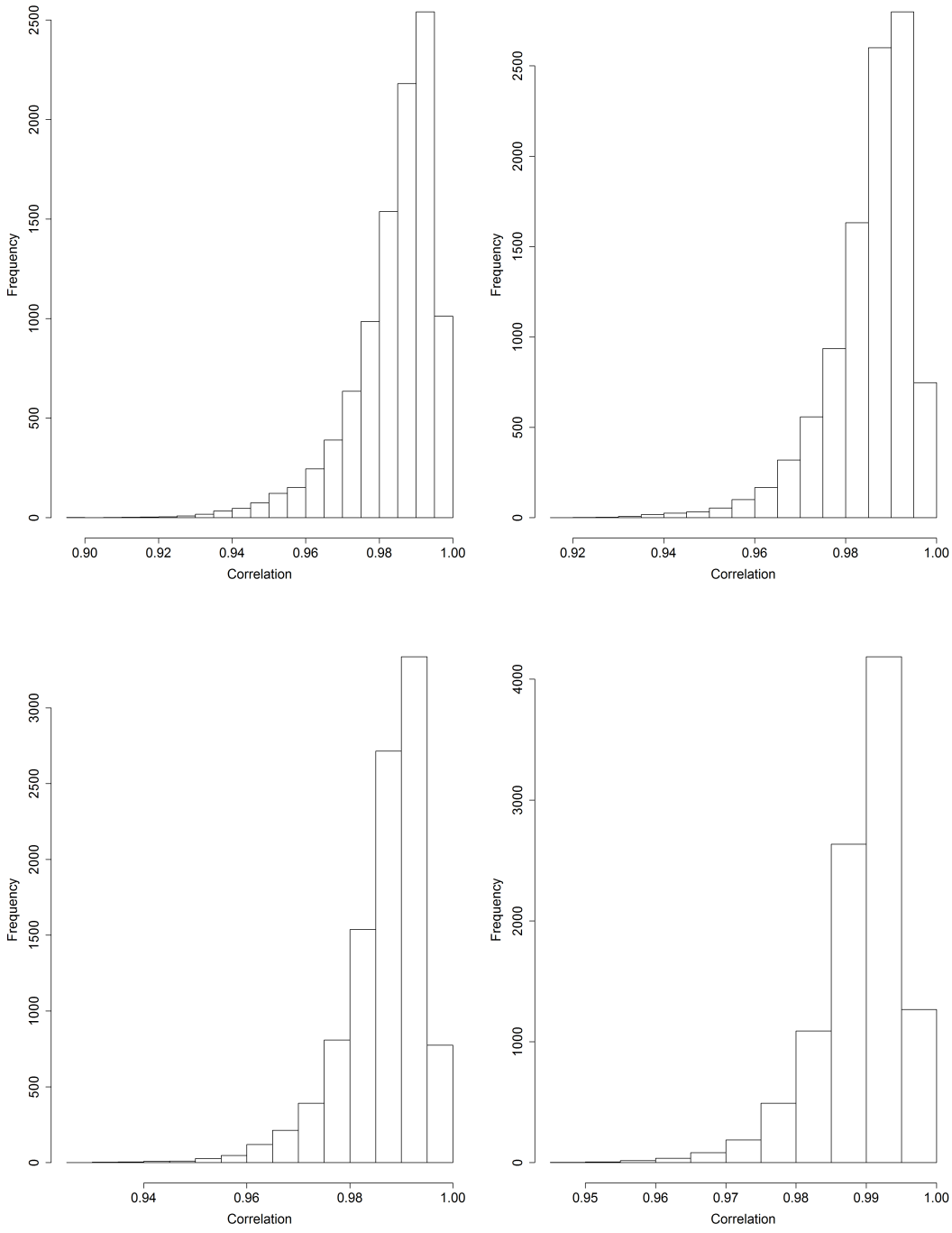

Figure 3.7: Histograms of the maximal-correlation statistic found using the grid search method for the standard normal distribution: on the top row from left to right are the plots for $n = 10$ and 15; on the bottom row from left to right are the plots for $n = 20$ and 30.

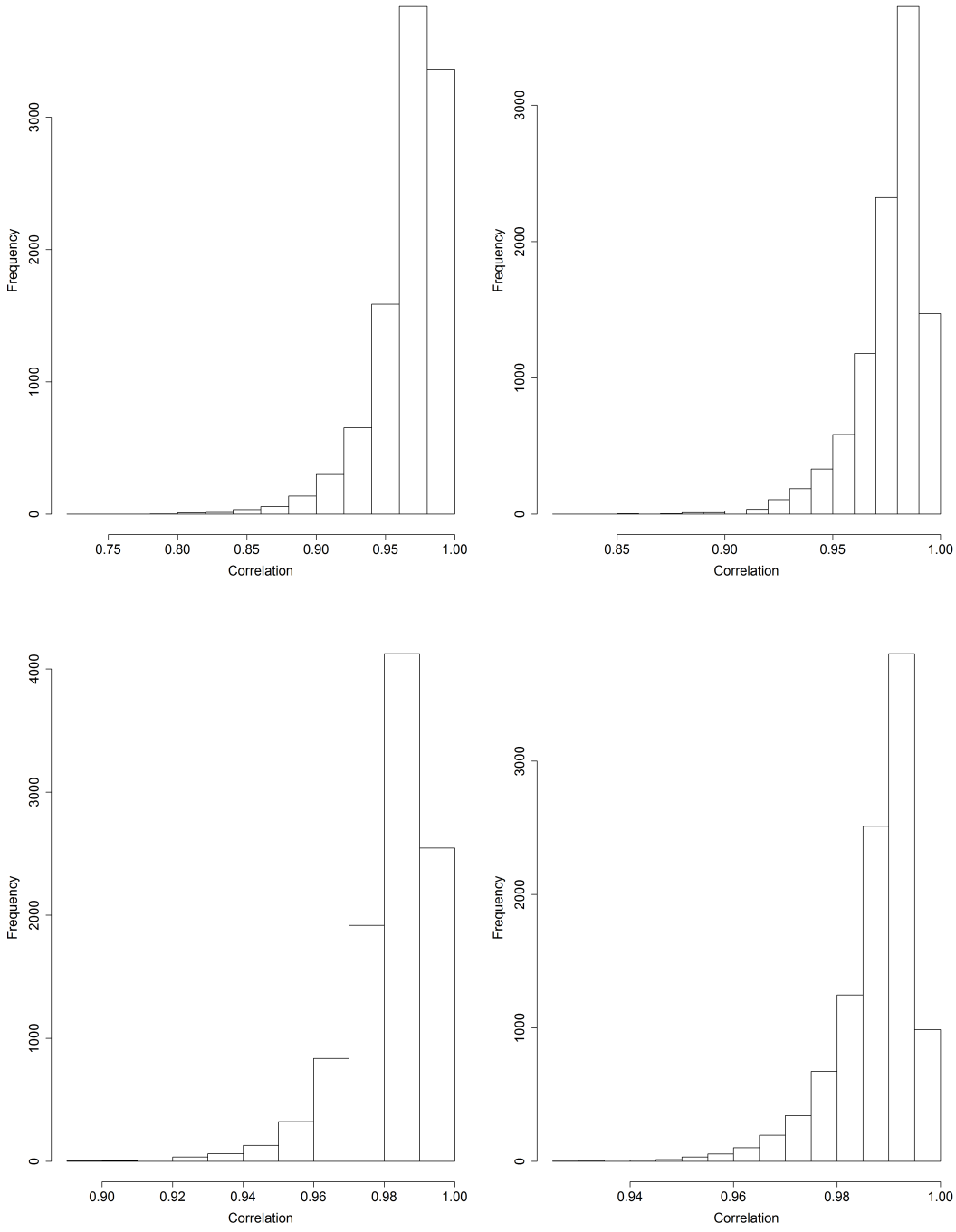

Figure 3.8: Histograms of the minimal-correlation statistic found using the averaging method for the standard normal distribution: on the top row from left to right are the plots for $n = 10$ and 15; on the bottom row from left to right are the plots for $n = 20$ and 30.

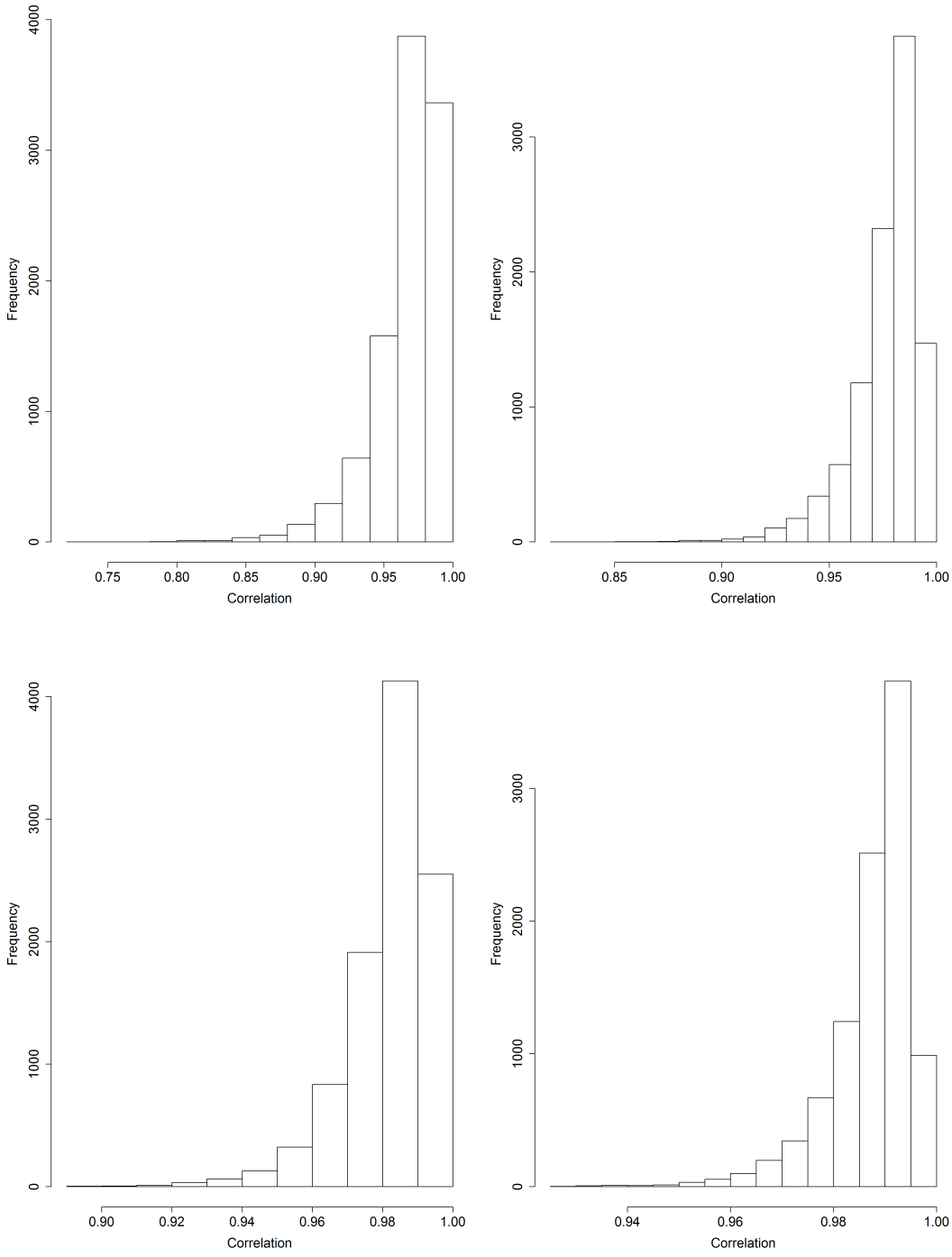

Figure 3.9: Histograms of the maximal-correlation statistic found using the averaging method for the standard normal distribution: on the top row from left to right are the plots for $n = 10$ and 15; on the bottom row from left to right are the plots for $n = 20$ and 30.

n/ Boundary	Maximal				Minimal			
	$i = 1$		$i = n$		$i = 1$		$i = n$	
	0.0001	$s_{2:n}$	$s_{n-1:n}$	0.9999	0.0001	$s_{2:n}$	$s_{n-1:n}$	0.9999
10	0.2824	0.2195	0.2180	0.2862	0.4655	0.5345	0.5367	0.4633
15	0.3196	0.2650	0.2618	0.3182	0.4770	0.5230	0.5238	0.4762
20	0.3354	0.2910	0.2951	0.3324	0.4782	0.5218	0.5145	0.4855
30	0.3726	0.3229	0.3313	0.3678	0.4788	0.5212	0.5204	0.4796

Table 3.7: Proportion of times the modified SCP plotting points occurred at the boundary for uniform$(0, 1)$

n / Boundary	Maximal				Minimal			
	$i = 1$		$i = n$		$i = 1$		$i = n$	
	0.0001	$s_{2:n}$	$s_{n-1:n}$	0.9999	0.0001	$s_{2:n}$	$s_{n-1:n}$	0.9999
10	0.0204	0.1144	0.4998	0.5002	0.1094	0.0203	0.5086	0.4914
15	0.0119	0.1381	0.5273	0.4727	0.1341	0.0132	0.4777	0.5223
20	0.0106	0.1503	0.5818	0.4182	0.1532	0.0117	0.4269	0.5731
30	0.0107	0.1724	0.6657	0.3343	0.1692	0.0107	0.3351	0.6649

Table 3.8: Proportion of times the modified SCP plotting points occurred at the boundary for the standard normal distribution

grid search had higher power for all cases for the minimal-correlation test, and almost all cases for the maximal-correlation test. In terms of the two tests, the minimal-correlation test generally had higher power than the maximal-correlation test for the grid search, except for case when the alternative was $B(3, 3)$. For the averaging method, the minimal-correlation test again had consistently higher power for $n = 20$ and 30 than the maximal-correlation test. Also we observe that the power increases with sample size.

These observations lead us to believe the optimal test, in general, is the minimal-correlation test based on the grid search method.

		Grid Search Method				Averaging Method			
		Maximal		Minimal		Maximal		Minimal	
n	Alternative	5%	10%	5%	10%	5%	10%	5%	10%
10	$B(1,2)$	0.1217	0.1935	0.1358	0.2191	0.1308	0.2114	0.1318	0.2130
	$B(1,3)$	0.1951	0.2934	0.2302	0.3438	0.2293	0.3249	0.2313	0.3265
	$B(2,1)$	0.1211	0.1921	0.1382	0.2251	0.1398	0.2203	0.1400	0.2198
	$B(3,1)$	0.1951	0.2902	0.2252	0.3304	0.2302	0.3232	0.2300	0.3241
	$B(3,3)$	0.0594	0.1188	0.0691	0.1360	0.0729	0.1340	0.0745	0.1374
15	$B(1,2)$	0.1928	0.2878	0.2210	0.3297	0.2144	0.3092	0.2153	0.3316
	$B(1,3)$	0.3466	0.4557	0.3945	0.5136	0.3824	0.4925	0.3844	0.4938
	$B(2,1)$	0.1929	0.2905	0.2257	0.3358	0.2198	0.3123	0.2212	0.3145
	$B(3,1)$	0.3382	0.4490	0.3895	0.5057	0.3816	0.4907	0.3821	0.4927
	$B(3,3)$	0.0839	0.1564	0.1071	0.1900	0.0951	0.1704	0.0979	0.1752
20	$B(1,2)$	0.2735	0.3778	0.3035	0.4204	0.2813	0.4012	0.2814	0.4018
	$B(1,3)$	0.4815	0.5840	0.5230	0.6345	0.5130	0.6259	0.5124	0.6260
	$B(2,1)$	0.2726	0.3816	0.3069	0.4178	0.2787	0.3976	0.2804	0.4002
	$B(3,1)$	0.4838	0.5892	0.5271	0.6346	0.5510	0.6223	0.5124	0.6249
	$B(3,3)$	0.1133	0.1958	0.1357	0.2355	0.1131	0.2043	0.1135	0.2081
30	$B(1,2)$	0.4097	0.5421	0.4446	0.5813	0.4257	0.5589	0.4268	0.5591
	$B(1,3)$	0.6936	0.7950	0.7287	0.8242	0.7187	0.8082	0.7192	0.8085
	$B(2,1)$	0.4071	0.5409	0.4406	0.5770	0.4271	0.5573	0.4283	0.5590
	$B(3,1)$	0.6943	0.7915	0.7287	0.8221	0.7129	0.8058	0.7142	0.8068
	$B(3,3)$	0.1540	0.2727	0.1837	0.3209	0.1727	0.2876	0.1753	0.2908

Table 3.9: Power of the minimal- and maximal-correlation tests for uniform$(0,1)$

		Grid Search Method				Averaging Method			
		Maximal		Minimal		Maximal		Minimal	
n	Alternative	5%	10%	5%	10%	5%	10%	5%	10%
10	$B(1,2)$	0.1263	0.2111	0.1284	0.2224	0.0740	0.1564	0.0742	0.1561
	$B(1,3)$	0.1408	0.2306	0.2185	0.3371	0.1532	0.2519	0.1519	0.2511
	$B(2,1)$	0.1220	0.2093	0.1259	0.2233	0.0896	0.1633	0.0818	0.1519
	$B(3,1)$	0.1396	0.2328	0.2246	0.3458	0.1474	0.2493	0.1382	0.2345
	$B(3,3)$	0.0615	0.1272	0.0343	0.0738	0.0248	0.0611	0.0241	0.0599
15	$B(1,2)$	0.1955	0.3008	0.2069	0.3501	0.1231	0.2173	0.1257	0.2214
	$B(1,3)$	0.2572	0.3688	0.3747	0.5294	0.2502	0.3798	0.2539	0.3826
	$B(2,1)$	0.1921	0.2971	0.2088	0.3510	0.1291	0.2255	0.1268	0.2191
	$B(3,1)$	0.2562	0.3704	0.3798	0.5375	0.2488	0.3772	0.2452	0.3706
	$B(3,3)$	0.0614	0.1268	0.0273	0.0726	0.0211	0.0499	0.0209	0.0493
20	$B(1,2)$	0.2747	0.3931	0.3127	0.4878	0.1750	0.3055	0.2040	0.3436
	$B(1,3)$	0.3787	0.5103	0.5350	0.6906	0.3613	0.5144	0.3855	0.5378
	$B(2,1)$	0.2740	0.4032	0.3143	0.5010	0.1765	0.3090	0.1964	0.3386
	$B(3,1)$	0.3866	0.5193	0.5420	0.7042	0.3615	0.5250	0.3997	0.5600
	$B(3,3)$	0.0549	0.1179	0.0268	0.0734	0.0159	0.0463	0.0202	0.0533
30	$B(1,2)$	0.4492	0.5969	0.5752	0.7475	0.3002	0.4785	0.4238	0.6014
	$B(1,3)$	0.6259	0.7546	0.7955	0.8960	0.5895	0.7420	0.6874	0.8161
	$B(2,1)$	0.4541	0.5995	0.5757	0.7517	0.3151	0.4923	0.4369	0.6132
	$B(3,1)$	0.6398	0.7565	0.8058	0.9045	0.5858	0.7388	0.6767	0.8089
	$B(3,3)$	0.0572	0.1190	0.0525	0.1243	0.0126	0.0430	0.0295	0.0747

Table 3.10: Power of minimal- and maximal-correlation tests for the standard normal distribution

3.4 Illustrative Examples

3.4.1 Example 1

To illustrate these methods, data from Nelson (2005) were used. This set of data is the time to failure of specimens of a new Class H electrical insulation at a temperature of 260 °C. Since the test items were inspected periodically and the lifetimes were assigned as the midpoint of the intervals between inspections, the data set contains some equal failure times. The experiment was done in order to estimate the life at 180 °C with an expected nominal life of 20,000 h. The observations, in hours, are

600 744 744 744 912 1228 1320 1464 1608 1896

The purpose of this test was to assess normality of the data. From Figure 3.10 we can see that the data looks approximately normal since the points fall closely about a 45 degree line for all of the plotting points except for the minimal grid search plotting points. For the grid search method, the minimal-correlation statistic was found to be 0.8502 (p-value of 0.6841) and the plotting points for $x_{1:n}$ and $x_{n:n}$ were 0.0001 and 0.8567, respectively; the maximal-correlation statistic was found to be 0.9745 (p-value of 0.1679) and the plotting points for $x_{1:n}$ and $x_{n:n}$ were 0.1014 and 0.9136, respectively. For the averaging method, the minimal-correlation statistic was 0.9730 (p-value of 0.5612) and the maximal-correlation statistic was 0.9722 (p-value of 0.5378).

In both methods, the p-values are similar for the minimal-correlation test, however for the maximal-correlation test the p-values are quite different. For both

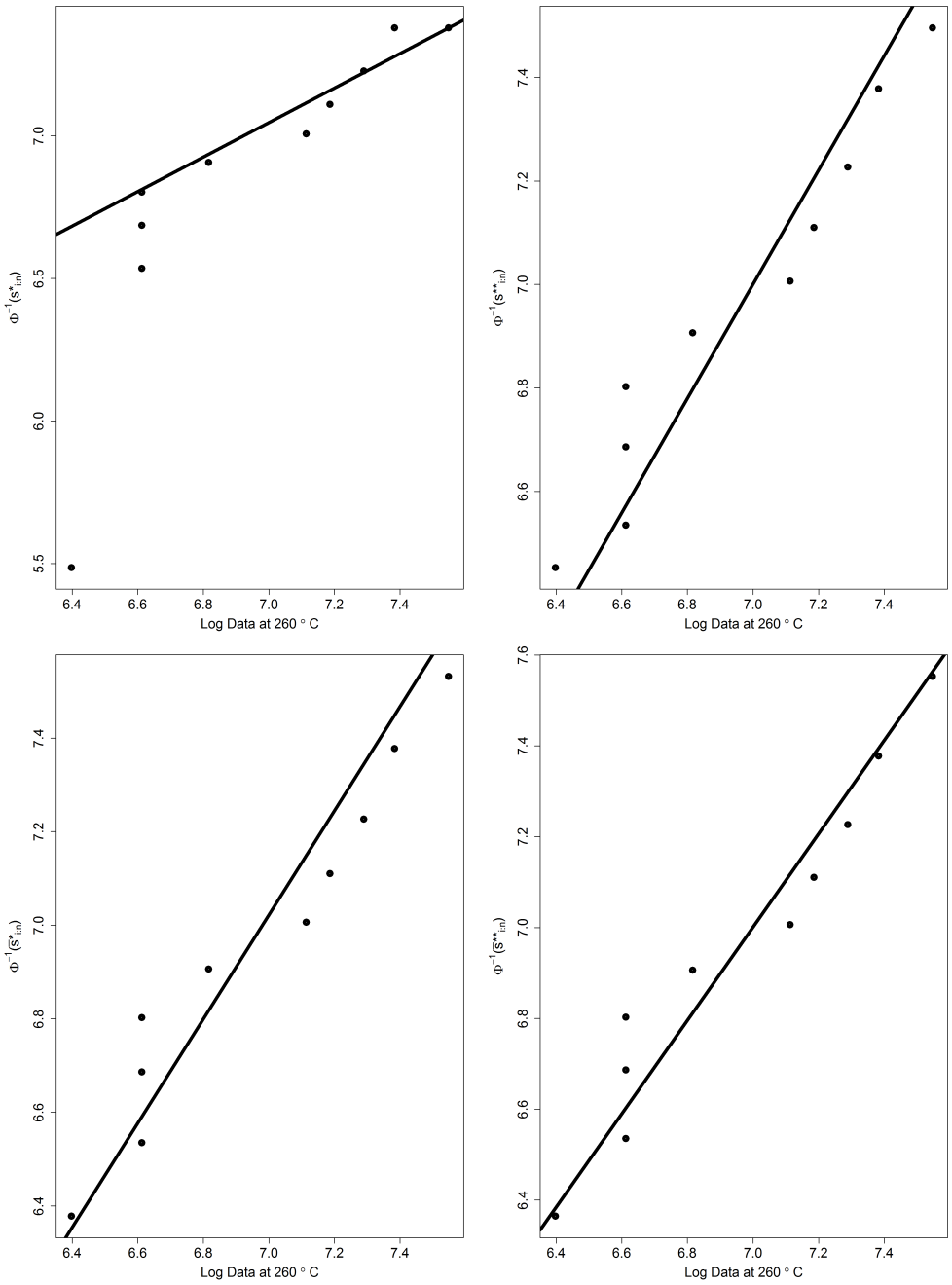

Figure 3.10: Starting at the top left and going from left to right, the plots are of the log of the data vs the inverse cdf of the normal distribution evaluated at the modified SCP plotting points for the minimal- and maximal-correlation tests for normality for the grid search method, and the minimal- and maximal-correlation tests for normality for the averaging method.

methods and tests, the p-values agree, i.e., the null hypothesis cannot be rejected and we can assume that the data come from a normal distribution. Also, since the p-value is higher for the minimal-correlation test in both methods, it is clear that the minimal-correlation test is better in detecting the normality of the data. From the p-values we can see that individually, between the methods, the minimal-correlation test is better at discriminating the data for the grid search method and the maximal-correlation was better at discriminating the data for the averaging method. These results are similar to those found by Balakrishnan et al. (2012a), since the correlation they reported was 0.9730 with a p-value of 0.5634.

3.4.2 Example 2

The following data were a simple random sample of size $n = 30$ generated according to the method given by Stephens and D'Agostino (1986):

79.89	88.13	90.03	92.56	95.97	99.62	103.56	105.48	111.38	113.90
85.29	89.33	91.46	95.14	96.20	102.56	103.60	106.82	112.97	115.95
87.83	89.35	92.55	95.94	98.70	103.22	104.21	108.39	113.75	118.52

Again, we wish to test for normality. For the grid search method, the minimal-correlation statistic was found to be 0.9349 (p-value of 0.5334) and the plotting points found for $x_{1:n}$ and $x_{n:n}$ were 0.0001 and 0.9999 respectively. The maximal-correlation statistic was found to be 0.9905 (p-value of 0.4921) and the plotting points for $x_{1:n}$ and $x_{n:n}$ were 0.0282 and 0.9638 respectively. For the averaging method, the correlation statistic for the minimal-correlation test was 0.9882 (p-value

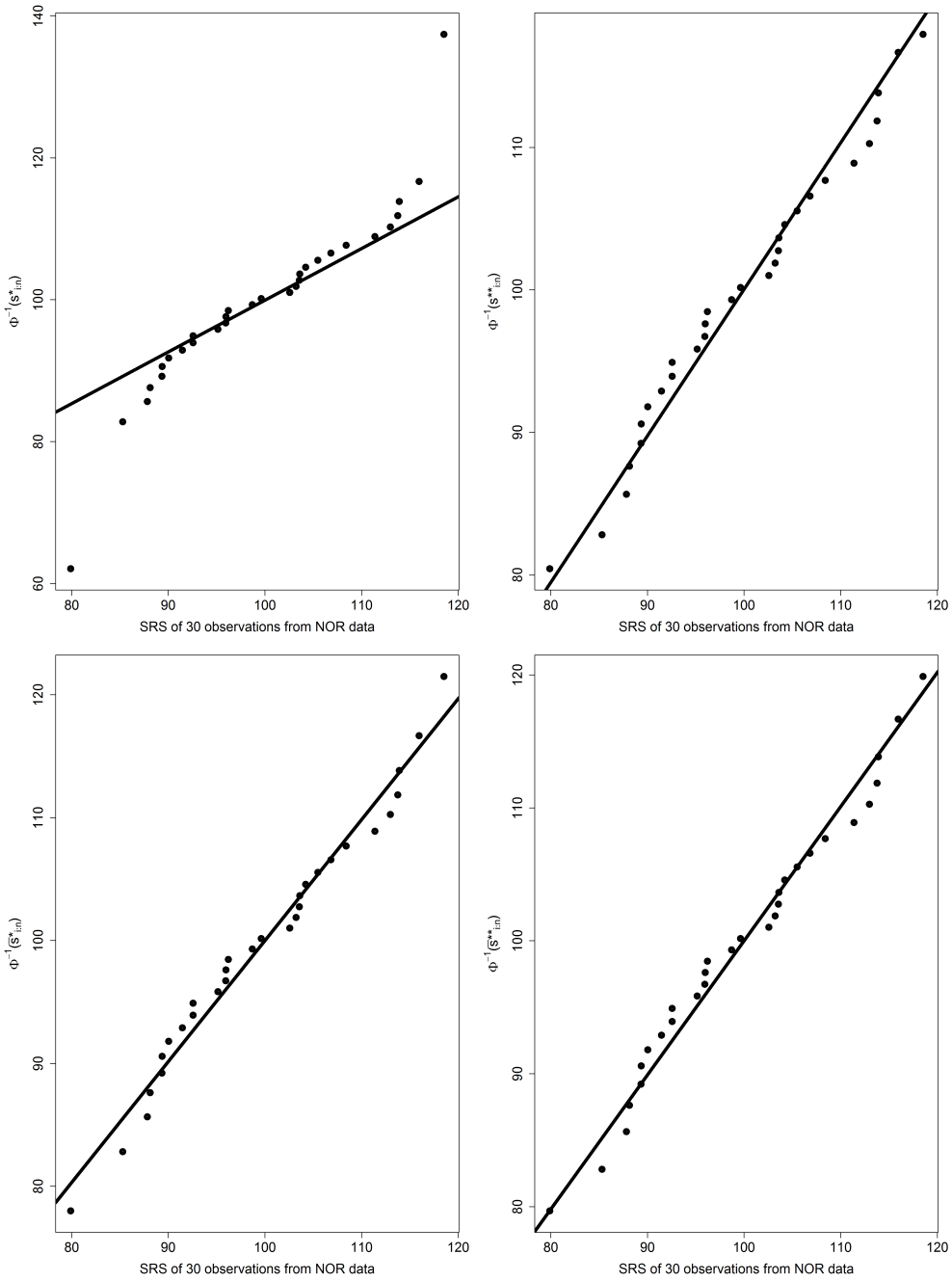

Figure 3.11: Starting at the top left and going from left to right, the plots are of the data vs the inverse cdf of the normal distribution evaluated at the modified SCP plotting points. The top two are for the minimal- and maximal-correlation tests for the grid search method and the bottom two are for the minimal- and maximal-correlation tests for the averaging method.

of 0.6605) and the maximal-correlation statistic was found to be 0.9899 (p-value of 0.7362).

All four tests agreed that the data came from a normal distribution. The p-values for the grid search method are lower than those found for the averaging method suggesting that the averaging method was better able to reach the conclusion that the data is normal. The p-value for the minimal-correlation test for the grid search method is higher than the maximal-correlation test; this was reversed for the averaging method. This shows that within the two tests, the minimal-correlation test is again better at discriminating than the maximal correlation test for the grid search and vice versa for the averaging method. All these results are consistent with those found by Balakrishnan et al. (2012a) whom, based on the traditonal SCP plotting points, found a correlation of 0.9899 and a p-value of 0.7362.

3.4.3 Example 3

Data from Stephens and D'Agostino (1986), given below, consist of order statistics of a random sample of values which in the literature have been tested for uniformity:

0.004 0.304 0.304 0.612 0.748 0.806 0.850 0.885 0.906 0.977

The minimal-correlation statistic was found to be 0.8314 (p-value of 0.0097) for the grid search method and the plotting points for $x_{1:n}$ and $x_{n:n}$ were 0.1584 and 0.9999 respectively. The maximal-correlation statistic was 0.9086 (p-value of 0.0138) and the plotting points for $x_{1:n}$ and $x_{n:n}$ where 0.001 and 0.8416 respectively. For the averaging method the correlation statistic was found to be 0.8735

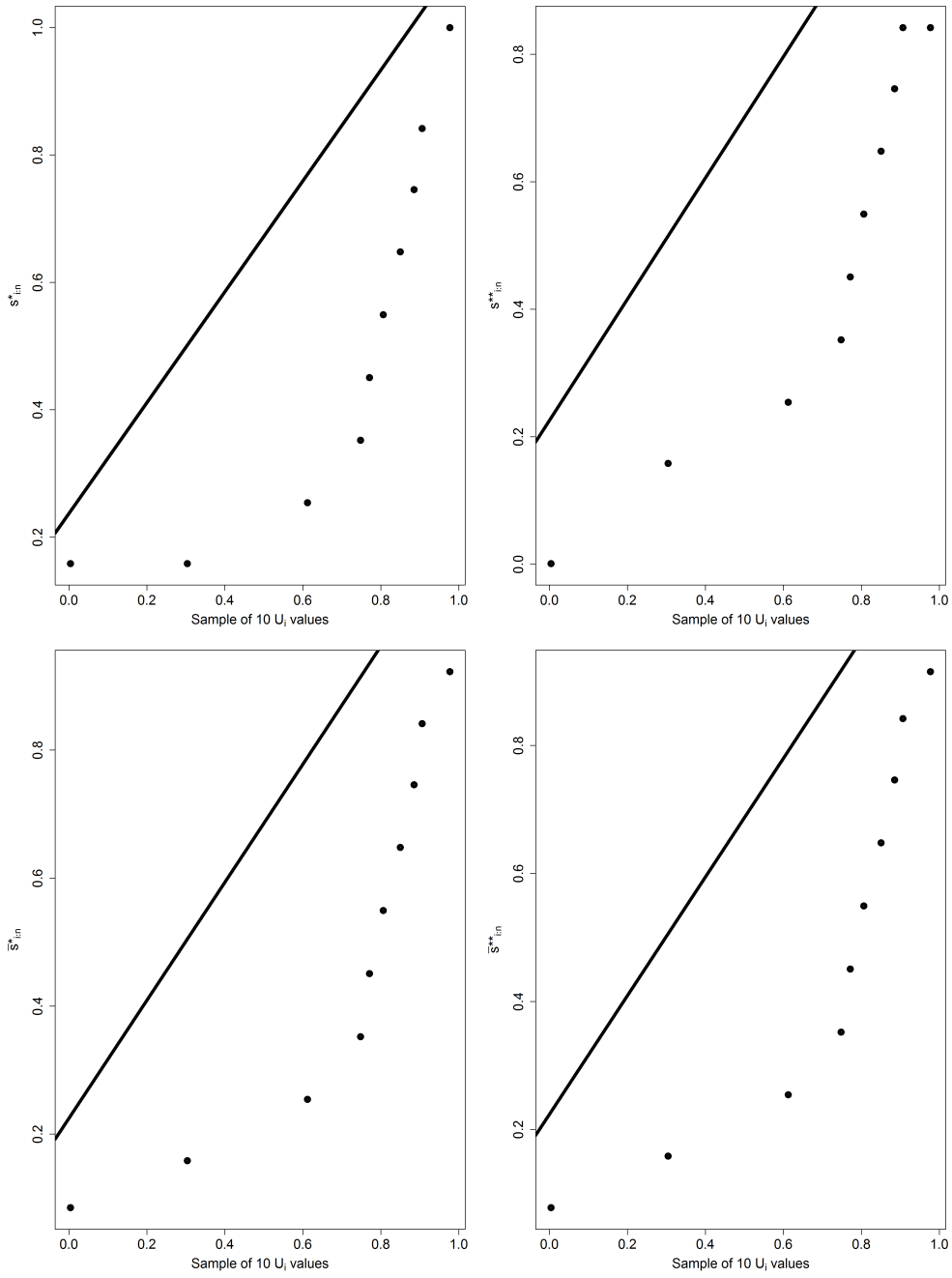

Figure 3.12: Starting at the top left and going from left to right, the plots are of the data vs the inverse cdf of the uniform distribution evaluated at the modified SCP plotting points. The top two are for the minimal- and maximal- correlation tests for uniformity for the grid search method. The bottom two are the minimal- and maximal-correlation tests for uniformity for the averaging method.

for the minimal-correlation test (p-value of 0.0098). For the maximal-correlation test the correlation was found to be 0.8770 (p-value of 0.0100). The p-values for both maximal-correlation tests are similar, as is the case for the minimal-correlation test p-values across both methods. In both methods, the minimal-correlation test has a lower p-value. All four p-values suggest that the data do not come from the uniform distribution; this is also evident from the QQ plots in Figure 3.12. The conclusion is consistent with the results of the various tests carried out in Stephens and D'Agostino (1986).

Chapter 4

Conclusion

In this thesis, two froms of a correlation-based test have been proposed to test for goodness-of-fit: a maximal-correlation test and a minimal-correlation test, each carried out using two methods. For the maximal-correlation test, we wanted to achieve the highest possible correlation to infer goodness-of-fit. For the minimal-correlation test we wanted to obtain a test statistic that was not too low so as to also infer goodness-of-fit. This was done using two methods: a grid search method and an averaging method. These two methods were used to simulate the distribution of test statistics for two different distributions: uniform$(0, 1)$ and the standard normal. The tests were compared through distributional properties and a power study. Through the comparison of the critical values of each test, it was noted that the averaging method had higher critical values than the grid search method for both the maximal and minimal test for both distributions.

Through a power study of the uniform$(0, 1)$, it was found that in general, the minimal-correlation test had consistently better power for the grid search method than the averaging method. In terms of the maximal-correlation test, it was found that it generally had better power for the averaging method than the grid search

method. Based on the power study for the standard normal, we found that the grid search method had consistently better power for both the maximal- and minimal-correlation tests. We also found that the minimal-correlation test had better power in most cases than the maximal-correlation test. The only exception to this was the for the alternative distribution $B(3,3)$, for this we noted that the maximal-correlation test had higher power than the minimal-correlation test for both methods. In general, for both methods the minimal test had higher power than the maximal test. We also observed that in all cases that as the sample size increased, the power also increased.

In the future, it would be interesting to compare of the power properties of the proposed tests to the correlation test that used the midpoints of $(0, s_{2:n})$ and $(s_{n-1:n}, 1)$ as the SCP plotting points for $x_{1:n}$ and $x_{n:n}$ as well as the correlation test using the mean- and median-ranks as plotting points. Another future work would be to extend the maximal- and minimal-correlation tests to test for goodness-of-fit to the extreme value distribution. Lastly, a more extensive power study could be carried to include more alternative distributions.

Appendix A

R code

```
#-------------------------------------------------------------------
# Maximal-Minimal Correlation Test for Uniform(0,1)
#-------------------------------------------------------------------
#-------------------------------------------------------------------
# Grid Search Method
#-------------------------------------------------------------------

#Initilizing the lists and vectors
ind1max <- c()
ind1min <- c()
r.max <- list()
r.min <- list()
pp.maxr <- list()
pp.maxc <- list()
pp.minr <- list()
pp.minc <- list()

# Creating vectors for each list
for(h in 1:4){
  r.max[[h]] <- c(length(1000))
  r.min[[h]] <- c(length(1000))
  pp.maxr[[h]] <- c(length(1000))
  pp.maxc[[h]] <- c(length(1000))
  pp.minr[[h]] <- c(length(1000))
  pp.minc[[h]] <- c(length(1000))
}
```

```
# Starting the counts at zero
lower.countmx1 <- list(0,0,0,0)
lower.countmx2 <- list(0,0,0,0)
lower.countmn1 <- list(0,0,0,0)
lower.countmn2 <- list(0,0,0,0)
upper.countmx1 <- list(0,0,0,0)
upper.countmx2 <- list(0,0,0,0)
upper.countmn1 <- list(0,0,0,0)
upper.countmn2 <- list(0,0,0,0)

for (l in  1:1000) {

# generating 30 random uniforms and allocating the data
  data <- runif(30)
  datag <- list()
  os.data <- list()
  datag[[1]] <- data[1:10]
  os.data[[1]] <- sort(datag[[1]])
  datag[[2]] <- data[1:15]
  os.data[[2]] <- sort(datag[[2]])
  datag[[3]] <- data[1:20]
  os.data[[3]] <- sort(datag[[3]])
  datag[[4]] <- data[1:30]
  os.data[[4]] <- sort(datag[[4]])

# allocating x_n, x_1 and spcs based on the size of n
  for (h in 1:4){
    if (length(os.data[[h]]) == 10) {
      scp1 <-  0.1584
      scp2 <- 0.8416
      scps3 <-  c(0.1584, 0.2542, 0.3521, 0.4506, 0.5494, 0.6479,
                  0.7458, 0.8416)
  }
    if (length(os.data[[h]]) == 15) {
      scp1 <- 0.1058
      scp2 <- 0.8942
      scps3 <- c(0.1058, 0.1699, 0.2353, 0.3012, 0.3674, 0.4337,
                 0.5000, 0.5663, 0.6326, 0.6988, 0.7647, 0.8301,
                 0.8942)
```

```
    }
    if(length(os.data[[h]]) == 20) {
      scp1 <- 0.0795
      scp2 <- 0.9205
      scps3 <- c(0.0795, 0.1275, 0.1766, 0.2261, 0.2758, 0.3256,
                 0.3754,0.4252, 0.4751, 0.5249, 0.5748, 0.6246,
                 0.6744, 0.7242, 0.7739,0.8234, 0.8725, 0.9205)
    }
    if(length(os.data[[h]]) == 30) {
      scp1 <- 0.0530
      scp2 <- 0.9470
      scps3 <- c(0.0530, 0.0851, 0.1178, 0.1508, 0.1840, 0.2172,
                 0.2504, 0.2837, 0.3169, 0.3502, 0.3835, 0.4168,
                 0.4501, 0.4834, 0.5166, 0.5499, 0.5832, 0.6165,
                 0.6498, 0.6831, 0.7163, 0.7496, 0.7828, 0.8160,
                 0.8492, 0.8822, 0.9149, 0.9470)
    }
    k <- 0

# Sequence for finding the max/min for the correlation
    scps1 <-seq(0.0001, scp1, by=0.0001)
    scps2 <- seq(scp2, 0.9999, by=0.0001)
    n1 <- length(scps1)
    n2 <- length(scps2)
    x = matrix(nrow = n1, ncol = n2)

# For loops to find the correlation at each i and j value
    for(i in scps1){
      k <- k+1
      m <-0
      for(j in scps2) {
        m <- m+1
        x[k,m] = cor(os.data[[h]], qunif(c(i, scps3, j)))
      }
    }

# Finding the counts, max/min and the plotting points
    ind1max <- which(x==max(x), TRUE)
    ind1min <- which(x==min(x), TRUE)
    r.max[[h]][l] <- x[ind1max]
```

```
      r.min[[h]][l] <- x[ind1min]
      pp.maxr[[h]][l] <- scps1[ind1max[1]]
      pp.maxc[[h]][l] <- scps2[ind1max[2]]
      pp.minr[[h]][l] <- scps1[ind1min[1]]
      pp.minc[[h]][l] <- scps2[ind1min[2]]

      if(pp.maxr[[h]][l] == scps1[1]){
        lower.countmx1[[h]] <- lower.countmx1[[h]]+1
      }

      if(pp.maxr[[h]][l] == scps1[n1]) {
        lower.countmx2[[h]] <- lower.countmx2[[h]]+1
      }

      if(pp.minr[[h]][l] == scps1[1]){
        lower.countmn1[[h]] <- lower.countmn1[[h]]+1
      }

      if (pp.minr[[h]][l]== scps1[n1]) {
        lower.countmn2[[h]] <- lower.countmn2[[h]]+1
      }
      if(pp.maxc[[h]][l] == scps2[1]){
        upper.countmx1[[h]] <- upper.countmx1[[h]]+1
      }

      if(pp.maxc[[h]][l]== scps2[n2]) {
        upper.countmx2[[h]] <- upper.countmx2[[h]]+1
      }
      if(pp.minc[[h]][l] == scps2[1]){
        upper.countmn1[[h]] <- upper.countmn1[[h]]+1
      }

      if (pp.minc[[h]][l] == scps2[n2]) {
        upper.countmn2[[h]] <- upper.countmn2[[h]]+1
      }
    }
  }
```

```
#  finding the proportions
prop_counts_10 <- lapply(lapply(combinedcounts_10, sum),
                         function(r) {r/10000})
prop_counts_15 <- lapply(lapply(combinedcounts_15, sum),
                         function(r) {r/10000})
prop_counts_20 <- lapply(lapply(combinedcounts_20, sum),
prop_counts_30 <- lapply(lapply(combinedcounts_30, sum),
                         function(r) {r/10000})

# sorting the correlations
maxminsort_10 <- lapply(combinedmaxmin_10,sort)
maxminsort_15 <- lapply(combinedmaxmin_15,sort)
maxminsort_20 <- lapply(combinedmaxmin_20,sort)
maxminsort_30 <- lapply(combinedmaxmin_30,sort)

# Extracting the critical values
max_10 <- maxminsort_10$V1
min_10 <- maxminsort_10$V2
max_15 <- maxminsort_15$V1
min_15 <- maxminsort_15$V2
max_20 <- maxminsort_20$V1
min_20 <- maxminsort_20$V2
max_30 <- maxminsort_30$V1
min_30 <- maxminsort_30$V2


# Finding the critical values
max_10[500]
min_10[500]
max_10[1000]
min_10[1000]
max_15[500]
min_15[500]
max_15[1000]
min_15[1000]
max_20[500]
min_20[500]
max_20[1000]
min_20[1000]
max_30[500]
```

```
min_30[500]
max_30[1000]
min_30[1000]

# Finding the plotting points
combinedpp_10 <- do.call(rbind, pp_10)
combinedpp_15 <- do.call(rbind, pp_15)
combinedpp_20 <- do.call(rbind, pp_20)
combinedpp_30 <- do.call(rbind, pp_30)

# Averaging the plotting points
mean_pp_10 <- lapply(combinedpp_10, mean)
mean_pp_15 <- lapply(combinedpp_15, mean)
mean_pp_20 <- lapply(combinedpp_20, mean)
mean_pp_30 <- lapply(combinedpp_30, mean)

#-------------------------------------------------------------
# Average Method
#-------------------------------------------------------------

# Initilizing the lists and vectors
corr_min <- list()
corr_max <- list()

# Creating vectors for each list
for(h in 1:4){
  corr_min[[h]] <- c(length(10000))
  corr_max[[h]] <- c(length(10000))
}

# Starting the counts at zero
for (l in  1:10000) {

# generating 30 random uniforms and allocating the data
  data <- runif(30)
  datag <- list()
  os.data <- list()
  datag[[1]] <- data[1:10]
  os.data[[1]] <- sort(datag[[1]])
  datag[[2]] <- data[1:15]
```

```
  os.data[[2]] <- sort(datag[[2]])
  datag[[3]] <- data[1:20]
  os.data[[3]] <- sort(datag[[3]])
  datag[[4]] <- data[1:30]
  os.data[[4]] <- sort(datag[[4]])

# allocating x_n, x_1 and spcs based on the size of n
  for (h in 1:4){
    if (length(os.data[[h]]) == 10) {
      scp1 <-  0.1584
      scp2 <- 0.8416
      scps3 <-  c(0.1584, 0.2542, 0.3521, 0.4506, 0.5494, 0.6479,
                  0.7458, 0.8416)
    }
    if (length(os.data[[h]]) == 15) {
      scp1 <- 0.1058
      scp2 <- 0.8942
      scps3 <- c(0.1058, 0.1699, 0.2353, 0.3012, 0.3674, 0.4337,
                 0.5000, 0.5663, 0.6326, 0.6988, 0.7647, 0.8301,
                 0.8942)
    }
    if(length(os.data[[h]]) == 20) {
      scp1 <- 0.0795
      scp2 <- 0.9205
      scps3 <- c(0.0795, 0.1275, 0.1766, 0.2261, 0.2758, 0.3256,
                 0.3754,0.4252, 0.4751, 0.5249, 0.5748, 0.6246,
                 0.6744, 0.7242, 0.7739,0.8234, 0.8725, 0.9205)
    }
    if(length(os.data[[h]]) == 30) {
      scp1 <- 0.0530
      scp2 <- 0.9470
      scps3 <- c(0.0530, 0.0851, 0.1178, 0.1508, 0.1840, 0.2172,
                 0.2504, 0.2837, 0.3169, 0.3502, 0.3835, 0.4168,
                 0.4501, 0.4834, 0.5166, 0.5499, 0.5832, 0.6165,
                 0.6498, 0.6831, 0.7163, 0.7496, 0.7828, 0.8160,
                 0.8492, 0.8822, 0.9149, 0.9470)
    }
    k <- 0

# Storing the correlations
```

```
corr_min[[h]][l] <- cor(os.data[[h]],
                        qunif(c(min_mean_1, scps3, min_mean_2)))corr_max
                        qunif(c(max_mean_1, scps3, max_mean_2))
    }
}

# Sorting the correlations
m2_min_10 <- sort(unlist(corr_min[1]))
m2_max_10 <- sort(unlist(corr_max[1]))
m2_min_15 <- sort(unlist(corr_min[2]))
m2_max_15 <- sort(unlist(corr_max[2]))
m2_min_20 <- sort(unlist(corr_min[3]))
m2_max_20 <- sort(unlist(corr_max[3]))
m2_min_30 <- sort(unlist(corr_min[4]))
m2_max_30 <- sort(unlist(corr_max[4]))

#Getting the critical Values
m2_max_10[500]
m2_min_10[500]
m2_max_10[1000]
m2_min_10[1000]
m2_max_15[500]
m2_min_15[500]
m2_max_15[1000]
m2_min_15[1000]
m2_max_20[500]
m2_min_20[500]
m2_max_20[1000]
m2_min_20[1000]
m2_max_30[500]
m2_min_30[500]
m2_max_30[1000]
m2_min_30[1000]

#-------------------------------------------------
#                    Power Study
#-------------------------------------------------

#-------------------------------------------------
# Grid Search Method
```

```
#-------------------------------------------------------------

# Initilizing the lists and vectors
ind1max <- c()
ind1min <- c()
r.max <- list()
r.min <- list()

# Creating vectors for each list
for(h in 1:4){
  r.max[[h]] <- c(length(500))
  r.min[[h]] <- c(length(500))
}

for (l in  1:500) {

# generating 30 random beta's and allocating the data
  data <- rbeta(30,1,2)
  datag <- list()
  os.data <- list()
  datag[[1]] <- data[1:10]
  os.data[[1]] <- sort(datag[[1]])
  datag[[2]] <- data[1:15]
  os.data[[2]] <- sort(datag[[2]])
  datag[[3]] <- data[1:20]
  os.data[[3]] <- sort(datag[[3]])
  datag[[4]] <- data[1:30]
  os.data[[4]] <- sort(datag[[4]])

# allocating x_n, x_1 and spcs based on the size of n
  for (h in 1:4){
    if (length(os.data[[h]]) == 10) {
      scp1 <-  0.1584
      scp2 <- 0.8416
      scps3 <-  c(0.1584, 0.2542, 0.3521, 0.4506, 0.5494,
                  0.6479, 0.7458, 0.8416)
    }
      if (length(os.data[[h]]) == 15) {
      scp1 <- 0.1058
      scp2 <- 0.8942
```

```
    scps3 <- c(0.1058, 0.1699, 0.2353, 0.3012, 0.3674,
               0.4337, 0.5000, 0.5663, 0.6326, 0.6988,
               0.7647, 0.8301, 0.8942)
  }
  if(length(os.data[[h]]) == 20) {
    scp1 <- 0.0795
    scp2 <- 0.9205
    scps3 <- c(0.0795, 0.1275, 0.1766, 0.2261, 0.2758,
               0.3256, 0.3754,0.4252, 0.4751, 0.5249,
               0.5748, 0.6246, 0.6744, 0.7242, 0.7739,
               0.8234, 0.8725, 0.9205)
  }
  if(length(os.data[[h]]) == 30) {
    scp1 <- 0.0530
    scp2 <- 0.9470
    scps3 <- c(0.0530, 0.0851, 0.1178, 0.1508, 0.1840,
               0.2172, 0.2504, 0.2837, 0.3169, 0.3502,
               0.3835, 0.4168, 0.4501, 0.4834, 0.5166,
               0.5499, 0.5832, 0.6165, 0.6498, 0.6831,
               0.7163, 0.7496, 0.7828, 0.8160,0.8492,
               0.8822, 0.9149, 0.9470)
  }
  k <- 0

# Creating the sequneces for finding the correlation
  scps1 <-seq(0.0001, scp1, by=0.0001)
  scps2 <- seq(scp2, 0.9999, by=0.0001)

  n1 <- length(scps1)
  n2 <- length(scps2)

  x = matrix(nrow = n1, ncol = n2)

# For loops to find the correlation at each i and j value
  for(i in scps1){
    k <- k+1
    m <-0
    for(j in scps2) {
      m <- m+1
      x[k,m] = cor(os.data[[h]], qunif(c(i, scps3, j)))
```

```
      }
    }

# Finding the counts, max/min and the plotting points
    ind1max <- which(x==max(x), TRUE)
    ind1min <- which(x==min(x), TRUE)
    r.max[[h]][l] <- x[ind1max]
    r.min[[h]][l] <- x[ind1min]
  }
}

# Finding the power
# n = 10
gs_countmax10_5 <- which(r.max[[1]] <= 0.9362388 )
gs_countmin10_5 <- which(r.min[[1]] <=  0.8802414 )
gs_countmax10_10 <- which(r.max[[1]] <= 0.9500979 )
gs_countmin10_10 <- which(r.min[[1]] <=  0.903626 )

# n = 15
gs_countmax15_5 <- which(r.max[[2]] <= 0.9509959 )
gs_countmin15_5 <- which(r.min[[2]] <= 0.9306624 )
gs_countmax15_10 <- which(r.max[[2]] <= 0.9612513 )
gs_countmin15_10 <- which(r.min[[2]] <= 0.944174 )

# n = 20
gs_countmax20_5 <- which(r.max[[3]] <= 0.9612149 )
gs_countmin20_5 <- which(r.min[[3]] <= 0.9507521 )
gs_countmax20_10 <- which(r.max[[3]] <= 0.9686859 )
gs_countmin20_10 <- which(r.min[[3]] <= 0.9598361 )

# n = 30
gs_countmax30_5 <- which(r.max[[4]] <= 0.9719308)
gs_countmin30_5 <- which(r.min[[4]] <= 0.9681896 )
gs_countmax30_10 <- which(r.max[[4]] <= 0.9779393 )
gs_countmin30_10 <- which(r.min[[4]] <= 0.974732 )

# Combining the Data
combined_count_10 <- cbind(length(gs_countmax10_5),
                           length(gs_countmin10_5),
                           length(gs_countmax10_10),
```

```
                              length(gs_countmin10_10),
                                )
combined_count_15 <- cbind(length(gs_countmax15_5),
                              length(gs_countmin15_5),
                              length(gs_countmax15_10),
                              length(gs_countmin15_10),
                               )
combined_count_20 <- cbind(length(gs_countmax20_5),
                              length(gs_countmin20_5),
                              length(gs_countmax20_10),
                              length(gs_countmin20_10),
                              )
combined_count_30 <- cbind(length(gs_countmax30_5),
                              length(gs_countmin30_5),
                              length(gs_countmax30_10),
                              length(gs_countmin30_10),
                              )

# finding the proportions
prop_counts_10 <- lapply(combined_props_10, sum)
prop_counts_15 <- lapply(combined_props_15, sum)
prop_counts_20 <- lapply(combined_props_20, sum)
prop_counts_30 <- lapply(combined_props_30, sum)

ulprops_10 <- unlist(prop_counts_10)
ulprops_15 <- unlist(prop_counts_15)
ulprops_20 <- unlist(prop_counts_20)
ulprops_30 <- unlist(prop_counts_30)

# Final results for the power study

cbind(ulprops_10, ulprops_15, ulprops_20, ulprops_30)

#-------------------------------------------------------------
# Average Method
#-------------------------------------------------------------

# Initilizing the lists and vectors
corr_min <- list()
corr_max <- list()
```

```
# Creating vectors for each list
for(h in 1:4){
  corr_min[[h]] <- c(length(10000))
  corr_max[[h]] <- c(length(10000))
}

# Starting the counts at zero
for (l in  1:10000) {

  # generating 30 random uniforms and allocating the data
  data <- runif(30)
  datag <- list()
  os.data <- list()
  datag[[1]] <- data[1:10]
  os.data[[1]] <- sort(datag[[1]])
  datag[[2]] <- data[1:15]
  os.data[[2]] <- sort(datag[[2]])
  datag[[3]] <- data[1:20]
  os.data[[3]] <- sort(datag[[3]])
  datag[[4]] <- data[1:30]
  os.data[[4]] <- sort(datag[[4]])

# allocating x_n, x_1 and spcs based on the size of n
  for (h in 1:4){
    if (length(os.data[[h]]) == 10) {
      scp1 <-  0.1584
      scp2 <- 0.8416
      scps3 <-  c(0.1584, 0.2542, 0.3521, 0.4506, 0.5494, 0.6479,
                  0.7458, 0.8416)
    }
    if (length(os.data[[h]]) == 15) {
      scp1 <- 0.1058
      scp2 <- 0.8942
      scps3 <- c(0.1058, 0.1699, 0.2353, 0.3012, 0.3674, 0.4337,
                 0.5000, 0.5663, 0.6326, 0.6988, 0.7647, 0.8301,
                 0.8942)
    }
    if(length(os.data[[h]]) == 20) {
      scp1 <- 0.0795
```

```
      scp2 <- 0.9205
      scps3 <- c(0.0795, 0.1275, 0.1766, 0.2261, 0.2758, 0.3256,
                 0.3754,0.4252, 0.4751, 0.5249, 0.5748, 0.6246,
                 0.6744, 0.7242, 0.7739,0.8234, 0.8725, 0.9205)
    }
    if(length(os.data[[h]]) == 30) {
      scp1 <- 0.0530
      scp2 <- 0.9470
      scps3 <- c(0.0530, 0.0851, 0.1178, 0.1508, 0.1840, 0.2172,
                 0.2504, 0.2837, 0.3169, 0.3502, 0.3835, 0.4168,
                 0.4501, 0.4834, 0.5166, 0.5499, 0.5832, 0.6165,
                 0.6498, 0.6831, 0.7163, 0.7496, 0.7828, 0.8160,
                 0.8492, 0.8822, 0.9149, 0.9470)
    }
    k <- 0

# Storing the correlations
    corr_min[[h]][l] <- cor(os.data[[h]],
                            qunif(c(min_mean_1, scps3, min_mean_2)))
    corr_max[[h]][l] <- cor(os.data[[h]],
                              qunif(c(max_mean_1, scps3, max_mean_2)))
     }
}

# Finding the Power
# n = 10
nm_countmax10_5 <- which(corr_max[[1]] <= 0.9174137)
nm_countmin10_5 <- which(corr_min[[1]] <=  0.9164834 )
nm_countmax10_10 <- which(corr_max[[1]] <= 0.9341992 )
nm_countmin10_10 <- which(corr_min[[1]] <= 0.933432 )

# n = 15
nm_countmax15_5 <- which(corr_max[[2]] <= 0.9439232)
nm_countmin15_5 <- which(corr_min[[2]] <= 0.9436713 )
nm_countmax15_10 <- which(corr_max[[2]] <= 0.9546443 )
nm_countmin15_10 <- which(corr_min[[2]] <= 0.9545003 )

# n = 20
nm_countmax20_5 <- which(corr_max[[3]] <= 0.9564804)
nm_countmin20_5 <- which(corr_min[[3]] <= 0.956344 )
```

```
nm_countmax20_10 <- which(corr_max[[3]] <= 0.9653213 )
nm_countmin20_10 <- which(corr_min[[3]] <= 0.9652664 )

#n = 30
nm_countmax30_5 <- which(corr_max[[4]] <= 0.9704636)
nm_countmin30_5 <- which(corr_min[[4]] <= 0.9704358 )
nm_countmax30_10 <- which(corr_max[[4]] <= 0.9764956 )
nm_countmin30_10 <- which(corr_min[[4]] <=  0.976452 )

# Combining the data
combined_count_10 <- c(length(nm_countmax10_5),
                       length(nm_countmin10_5),
                       length(nm_countmax10_10),
                       length(nm_countmin10_10) )
combined_count_15 <- c( length(nm_countmax15_5),
                       length(nm_countmin15_5),
                       length(nm_countmax15_10),
                       length(nm_countmin15_10) )
combined_count_20 <- c(length(nm_countmax20_5),
                      length(nm_countmin20_5),
                      length(nm_countmax20_10),
                      length(nm_countmin20_10) )
combined_count_30 <- c(length(nm_countmax30_5),
                      length(nm_countmin30_5),
                      length(nm_countmax30_10),
                      length(nm_countmin30_10) )

#-------------------------------------------------
#Maximal-Minimal Correlation Test for Normal(0,1)
#-------------------------------------------------
#-------------------------------------------------
# Grid Search Method
#-------------------------------------------------
# Initilizing the lists and vectors
ind1max <- c()
ind1min <- c()
r.max <- list()
r.min <- list()
pp.maxr <- list()
pp.maxc <- list()
```

```
pp.minr <- list()
pp.minc <- list()

# Creating vectors for each list
for(h in 1:4){
  r.max[[h]] <- c(length(1000))
  r.min[[h]] <- c(length(1000))
  pp.maxr[[h]] <- c(length(1000))
  pp.maxc[[h]] <- c(length(1000))
  pp.minr[[h]] <- c(length(1000))
  pp.minc[[h]] <- c(length(1000))
}

# Starting the counts at zero
lower.countmx1 <- list(0,0,0,0)
lower.countmx2 <- list(0,0,0,0)
lower.countmn1 <- list(0,0,0,0)
lower.countmn2 <- list(0,0,0,0)
upper.countmx1 <- list(0,0,0,0)
upper.countmx2 <- list(0,0,0,0)
upper.countmn1 <- list(0,0,0,0)
upper.countmn2 <- list(0,0,0,0)

for (l in  1:500) {

# generating 30 random uniforms and allocating the data
  data <- rnorm(30, mean = 0, sd = 1)
  datag <- list()
  os.data <- list()
  datag[[1]] <- data[1:10]
  os.data[[1]] <- sort(datag[[1]])
  datag[[2]] <- data[1:15]
  os.data[[2]] <- sort(datag[[2]])
  datag[[3]] <- data[1:20]
  os.data[[3]] <- sort(datag[[3]])
  datag[[4]] <- data[1:30]
  os.data[[4]] <- sort(datag[[4]])

# allocating x_n, x_1 and spcs based on the size of n
for (h in 1:4){
```

```
    if (length(os.data[[h]]) == 10) {
      scp1 <-  0.1433
      scp2 <-  0.8567
      scps3 <-  c(0.1433, 0.2472, 0.3487, 0.4496, 0.5504, 0.6513,
       0.7528, 0.8567)
    }
    if (length(os.data[[h]]) == 15) {
      scp1 <- 0.0951
      scp2 <- 0.9049
      scps3 <- c(0.0951, 0.1644, 0.2320, 0.2992, 0.3662,
      0.4331, 0.5000,  0.5669, 0.6338, 0.7008, 0.7680,
      0.8356, 0.9049)
    }
    if(length(os.data[[h]]) == 20) {
      scp1 <- 0.0712
      scp2 <- 0.9288
      scps3 <- c(0.0712, 0.1231, 0.1738, 0.2242, 0.2745,
      0.3246, 0.3748, 0.4249, 0.4750, 0.5250, 0.5751,
      0.6252, 0.6754, 0.7255, 0.7758, 0.8262, 0.8769,
       0.9288)
    }
    if(length(os.data[[h]]) == 30) {
      scp1 <- 0.0473
      scp2 <- 0.9527
      scps3 <- c(0.0473, 0.0820, 0.1158, 0.1494, 0.1829,
       0.2164, 0.2497, 0.2831, 0.3165, 0.3499, 0.3832,
        0.4166, 0.4500, 0.4833,  0.5167, 0.5500, 0.5834,
        0.6168, 0.6501, 0.6835, 0.7169, 0.7503, 0.7836,
        0.8171, 0.8506, 0.8842, 0.9180, 0.9527)
    }
    k <- 0

# Creating the sequences for finding the correlation
    scps1 <-seq(0.0001, scp1, by=0.0001)
    scps2 <- seq(scp2, 0.9999, by=0.0001)

    n1 <- length(scps1)
    n2 <- length(scps2)

    x = matrix(nrow = n1, ncol = n2)
```

```
# For loops to find the correlation at each i and j value
    for(i in scps1){
      k <- k+1
      m <-0
      for(j in scps2) {
        m <- m+1
        x[k,m] = cor(os.data[[h]], qnorm(c(i, scps3, j))
      }
    }

# Finding the counts, max/min and the plotting points
    ind1max <- which(x==max(x), TRUE)
    ind1min <- which(x==min(x), TRUE)
    r.max[[h]][l] <- x[ind1max]
    r.min[[h]][l] <- x[ind1min]
    pp.maxr[[h]][l] <- scps1[ind1max[1]]
    pp.maxc[[h]][l] <- scps2[ind1max[2]]
    pp.minr[[h]][l] <- scps1[ind1min[1]]
    pp.minc[[h]][l] <- scps2[ind1min[2]]

    if(pp.maxr[[h]][l] == scps1[1]){
      lower.countmx1[[h]] <- lower.countmx1[[h]]+1
    }
    if(pp.maxr[[h]][l] == scps1[n1]) {
      lower.countmx2[[h]] <- lower.countmx2[[h]]+1
    }
    if(pp.minr[[h]][l] == scps1[1]){
      lower.countmn1[[h]] <- lower.countmn1[[h]]+1
    }
    if (pp.minr[[h]][l]== scps1[n1]) {
      lower.countmn2[[h]] <- lower.countmn2[[h]]+1
    }
    if(pp.maxc[[h]][l] == scps2[1]){
      upper.countmx1[[h]] <- upper.countmx1[[h]]+1
    }
    if(pp.maxc[[h]][l]== scps2[n2]) {
      upper.countmx2[[h]] <- upper.countmx2[[h]]+1
    }
    if(pp.minc[[h]][l] == scps2[1]){
```

```
      upper.countmn1[[h]] <- upper.countmn1[[h]]+1
    }
    if (pp.minc[[h]][l] == scps2[n2]) {
      upper.countmn2[[h]] <- upper.countmn2[[h]]+1
    }
  }
}


#  finding the proportions
prop_counts_10 <- lapply(lapply(combinedcounts_10, sum),
                         function(r) {r/10000})
prop_counts_15 <- lapply(lapply(combinedcounts_15, sum),
                         function(r) {r/10000})
prop_counts_20 <- lapply(lapply(combinedcounts_20, sum),
prop_counts_30 <- lapply(lapply(combinedcounts_30, sum),
                         function(r) {r/10000})


# sorting the correlations
maxminsort_10 <- lapply(combinedmaxmin_10,sort)
maxminsort_15 <- lapply(combinedmaxmin_15,sort)
maxminsort_20 <- lapply(combinedmaxmin_20,sort)
maxminsort_30 <- lapply(combinedmaxmin_30,sort)

# Extracting the critical values
max_10 <- maxminsort_10$V1
min_10 <- maxminsort_10$V2
max_15 <- maxminsort_15$V1
min_15 <- maxminsort_15$V2
max_20 <- maxminsort_20$V1
min_20 <- maxminsort_20$V2
max_30 <- maxminsort_30$V1
min_30 <- maxminsort_30$V2

# Finding the critical values
max_10[500]
min_10[500]
max_10[1000]
min_10[1000]
```

```
max_15[500]
min_15[500]
max_15[1000]
min_15[1000]
max_20[500]
min_20[500]
max_20[1000]
min_20[1000]
max_30[500]
min_30[500]
max_30[1000]
min_30[1000]

# Finding the plotting points
combinedpp_10 <- do.call(rbind, pp_10)
combinedpp_15 <- do.call(rbind, pp_15)
combinedpp_20 <- do.call(rbind, pp_20)
combinedpp_30 <- do.call(rbind, pp_30)

# Averaging the plotting points
mean_pp_10 <- lapply(combinedpp_10, mean)
mean_pp_15 <- lapply(combinedpp_15, mean)
mean_pp_20 <- lapply(combinedpp_20, mean)
mean_pp_30 <- lapply(combinedpp_30, mean)

#-------------------------------------------------
# Average Method
#-------------------------------------------------

# Initializing the lists and vectors
corr_min <- list()
corr_max <- list()

# Creating vectors for each list

for(h in 1:4){
  corr_min[[h]] <- c(length(10000))
  corr_max[[h]] <- c(length(10000))
}
```

```
for (l in  1:10000) {

# generating 30 random uniforms and allocating the data

  data <- rnorm(30, mean = 0, sd = 1)
  datag <- list()
  os.data <- list()
  datag[[1]] <- data[1:10]
  os.data[[1]] <- sort(datag[[1]])
  datag[[2]] <- data[1:15]
  os.data[[2]] <- sort(datag[[2]])
  datag[[3]] <- data[1:20]
  os.data[[3]] <- sort(datag[[3]])
  datag[[4]] <- data[1:30]
  os.data[[4]] <- sort(datag[[4]])

  # allocating x_n, x_1 and spcs based on the size of n
    for (h in 1:4){
    if (length(os.data[[h]]) == 10) {
      scp1 <-  0.1433
      scp2 <-  0.8567
      scps3 <-  c(0.1433, 0.2472, 0.3487, 0.4496, 0.5504, 0.6513,
       0.7528, 0.8567)
    }

    if (length(os.data[[h]]) == 15) {
      scp1 <- 0.0951
      scp2 <- 0.9049
      scps3 <- c(0.0951, 0.1644, 0.2320, 0.2992, 0.3662,
      0.4331, 0.5000,  0.5669, 0.6338, 0.7008, 0.7680,
      0.8356, 0.9049)
    }
    if(length(os.data[[h]]) == 20) {
      scp1 <- 0.0712
      scp2 <- 0.9288
      scps3 <- c(0.0712, 0.1231, 0.1738, 0.2242, 0.2745,
      0.3246, 0.3748, 0.4249, 0.4750, 0.5250, 0.5751,
      0.6252, 0.6754, 0.7255, 0.7758, 0.8262, 0.8769,
       0.9288)
    }
```

```
    if(length(os.data[[h]]) == 30) {
      scp1 <- 0.0473
      scp2 <- 0.9527
      scps3 <- c(0.0473, 0.0820, 0.1158, 0.1494, 0.1829,
       0.2164, 0.2497, 0.2831, 0.3165, 0.3499, 0.3832,
        0.4166, 0.4500, 0.4833,  0.5167, 0.5500, 0.5834,
        0.6168, 0.6501, 0.6835, 0.7169, 0.7503, 0.7836,
        0.8171, 0.8506, 0.8842, 0.9180, 0.9527)
    }
    k <- 0

    # Storing the correlations

    corr_min[[h]][l] <- cor(os.data[[h]],
        qnorm(c(min_mean_1, scps3, min_mean_2)))
    corr_max[[h]][l] <- cor(os.data[[h]],
         qnorm(c(max_mean_1, scps3, max_mean_2)))
  }
}

# Sorting the data

m2_min_10 <- sort(unlist(corr_min[1]))
m2_max_10 <- sort(unlist(corr_max[1]))
m2_min_15 <- sort(unlist(corr_min[2]))
m2_max_15 <- sort(unlist(corr_max[2]))
m2_min_20 <- sort(unlist(corr_min[3]))
m2_max_20 <- sort(unlist(corr_max[3]))
m2_min_30 <- sort(unlist(corr_min[4]))
m2_max_30 <- sort(unlist(corr_max[4]))

# To get the critical Values

m2_max_10[500]
m2_min_10[500]
m2_max_10[1000]
m2_min_10[1000]
m2_max_15[500]
m2_min_15[500]
m2_max_15[1000]
```

```
m2_min_15[1000]
m2_max_20[500]
m2_min_20[500]
m2_max_20[1000]
m2_min_20[1000]
m2_max_30[500]
m2_min_30[500]
m2_max_30[1000]
m2_min_30[1000]

#-------------------------------------------------
#                  Power Study
#-------------------------------------------------

#-------------------------------------------------
# Grid search Method
#-------------------------------------------------

# Initializing the lists and vectors

ind1max <- c()
ind1min <- c()
r.max <- list()
r.min <- list()

# Creating vectors for each list

for(h in 1:4){
  r.max[[h]] <- c(length(500))
  r.min[[h]] <- c(length(500))
}

for (l in  1:500) {

# generating 30 random Betas and allocating the data
  data <- rbeta(30,1,2)
  datag <- list()
  os.data <- list()
  datag[[1]] <- data[1:10]
  os.data[[1]] <- sort(datag[[1]])
```

```
  datag[[2]] <- data[1:15]
  os.data[[2]] <- sort(datag[[2]])
  datag[[3]] <- data[1:20]
  os.data[[3]] <- sort(datag[[3]])
  datag[[4]] <- data[1:30]
  os.data[[4]] <- sort(datag[[4]])

# allocating x_n, x_1 and spcs based on the size of n

  for (h in 1:4){
    if (length(os.data[[h]]) == 10) {
      scp1 <-  0.1433
      scp2 <-  0.8567
      scps3 <-  c(0.1433, 0.2472, 0.3487, 0.4496, 0.5504, 0.6513,
       0.7528, 0.8567)
    }

    if (length(os.data[[h]]) == 15) {
      scp1 <- 0.0951
      scp2 <- 0.9049
      scps3 <- c(0.0951, 0.1644, 0.2320, 0.2992, 0.3662,
      0.4331, 0.5000,  0.5669, 0.6338, 0.7008, 0.7680,
      0.8356, 0.9049)
    }
    if(length(os.data[[h]]) == 20) {
      scp1 <- 0.0712
      scp2 <- 0.9288
      scps3 <- c(0.0712, 0.1231, 0.1738, 0.2242, 0.2745,
      0.3246, 0.3748, 0.4249, 0.4750, 0.5250, 0.5751,
      0.6252, 0.6754, 0.7255, 0.7758, 0.8262, 0.8769,
       0.9288)
    }
    if(length(os.data[[h]]) == 30) {
      scp1 <- 0.0473
      scp2 <- 0.9527
      scps3 <- c(0.0473, 0.0820, 0.1158, 0.1494, 0.1829,
       0.2164, 0.2497, 0.2831, 0.3165, 0.3499, 0.3832,
        0.4166, 0.4500, 0.4833,  0.5167, 0.5500, 0.5834,
        0.6168, 0.6501, 0.6835, 0.7169, 0.7503, 0.7836,
        0.8171, 0.8506, 0.8842, 0.9180, 0.9527)
```

```
    }
    k <- 0

# Creating the sequences for finding the correlation
 scps1 <-seq(0.0001, scp1, by=0.0001)
    scps2 <- seq(scp2, 0.9999, by=0.0001)

    n1 <- length(scps1)
    n2 <- length(scps2)

    x = matrix(nrow = n1, ncol = n2)

# For loops to find the correlation at each i and j value
for(i in scps1){
       k <- k+1
      m <-0
      for(j in scps2) {
        m <- m+1
        x[k,m] = cor(os.data[[h]], qnorm(c(i, scps3, j)))
      }
    }

# Finding the counts, max/min and the coordinates at the max and min
    ind1max <- which(x==max(x), TRUE)
    ind1min <- which(x==min(x), TRUE)
    r.max[[h]][l] <- x[ind1max]
    r.min[[h]][l] <- x[ind1min]
  }
}

# Finding the Power
# n = 10
gs_countmax10_5 <- which(r.max[[1]] <=  0.9607204 )
gs_countmin10_5 <- which(r.min[[1]] <=  0.7286672 )
gs_countmax10_10 <- which(r.max[[1]] <= 0.9688554)
gs_countmin10_10 <- which(r.min[[1]] <=  0.7547894 )

# n = 15
gs_countmax15_5 <- which(r.max[[2]] <= 0.9666295 )
gs_countmin15_5 <- which(r.min[[2]] <=  0.8063892)
```

```
gs_countmax15_10 <- which(r.max[[2]] <=  0.9729472)
gs_countmin15_10 <- which(r.min[[2]] <= 0.8250993 )

# n = 20
gs_countmax20_5 <- which(r.max[[3]] <= 0.9709903 )
gs_countmin20_5 <- which(r.min[[3]] <= 0.8509791 )
gs_countmax20_10 <- which(r.max[[3]] <= 0.976458 )
gs_countmin20_10 <- which(r.min[[3]] <= 0.8648301 )

# n = 30
gs_countmax30_5 <- which(r.max[[4]] <= 0.977135)
gs_countmin30_5 <- which(r.min[[4]] <= 0.8990045 )
gs_countmax30_10 <- which(r.max[[4]] <= 0.9813577 )
gs_countmin30_10 <- which(r.min[[4]] <= 0.9078754)

# Combining the Values
combined_count_10 <- cbind(length(gs_countmax10_5),
                           length(gs_countmin10_5),
                           length(gs_countmax10_10),
                           length(gs_countmin10_10),
                            )
combined_count_15 <- cbind(length(gs_countmax15_5),
                           length(gs_countmin15_5),
                           length(gs_countmax15_10),
                           length(gs_countmin15_10),
                            )
combined_count_20 <- cbind(length(gs_countmax20_5),
                           length(gs_countmin20_5),
                           length(gs_countmax20_10),
                           length(gs_countmin20_10),
                            )

combined_count_30 <- cbind(length(gs_countmax30_5),
                           length(gs_countmin30_5),
                           length(gs_countmax30_10),
                           length(gs_countmin30_10),
                            )

# Combing the power
combined_props_10 <- do.call(rbind, props_10)
```

```
combined_props_15 <- do.call(rbind, props_15)
combined_props_20 <- do.call(rbind, props_20)
combined_props_30 <- do.call(rbind, props_30)

# Summing the results
prop_counts_10 <- lapply(combined_props_10, sum)
prop_counts_15 <- lapply(combined_props_15, sum)
prop_counts_20 <- lapply(combined_props_20, sum)
prop_counts_30 <- lapply(combined_props_30, sum)

ulprops_10 <- unlist(prop_counts_10)
ulprops_15 <- unlist(prop_counts_15)
ulprops_20 <- unlist(prop_counts_20)
ulprops_30 <- unlist(prop_counts_30)

# creating a table of results
cbind(ulprops_10, ulprops_15, ulprops_20, ulprops_30)
#-----------------------------------------
#Average method
#-----------------------------------------

# Initializing the lists and vectors
corr_min <- list()
corr_max <- list()

# Creating vectors for each list
for(h in 1:4){
  corr_min[[h]] <- c(length(10000))
  corr_max[[h]] <- c(length(10000))
}

for (l in  1:10000) {

# generating 30 random Beta's and allocating the data
  data <- rbeta(30,1,2)
  datag <- list()
  os.data <- list()
  datag[[1]] <- data[1:10]
  os.data[[1]] <- sort(datag[[1]])
  datag[[2]] <- data[1:15]
```

```
  os.data[[2]] <- sort(datag[[2]])
  datag[[3]] <- data[1:20]
  os.data[[3]] <- sort(datag[[3]])
  datag[[4]] <- data[1:30]
  os.data[[4]] <- sort(datag[[4]])

# allocating x_n, x_1 and spcs based on the size of n  for (h in 1:4){
    if (length(os.data[[h]]) == 10) {
      scp1 <-  0.1433
      scp2 <-  0.8567
      scps3 <-  c(0.1433, 0.2472, 0.3487, 0.4496, 0.5504, 0.6513,
       0.7528, 0.8567)
    }
     if (length(os.data[[h]]) == 15) {
      scp1 <- 0.0951
      scp2 <- 0.9049
      scps3 <- c(0.0951, 0.1644, 0.2320, 0.2992, 0.3662,
      0.4331, 0.5000,  0.5669, 0.6338, 0.7008, 0.7680,
      0.8356, 0.9049)
    }
    if(length(os.data[[h]]) == 20) {
      scp1 <- 0.0712
      scp2 <- 0.9288
      scps3 <- c(0.0712, 0.1231, 0.1738, 0.2242, 0.2745,
      0.3246, 0.3748, 0.4249, 0.4750, 0.5250, 0.5751,
      0.6252, 0.6754, 0.7255, 0.7758, 0.8262, 0.8769,
       0.9288)
    }
    if(length(os.data[[h]]) == 30) {
      scp1 <- 0.0473
      scp2 <- 0.9527
      scps3 <- c(0.0473, 0.0820, 0.1158, 0.1494, 0.1829,
       0.2164, 0.2497, 0.2831, 0.3165, 0.3499, 0.3832,
        0.4166, 0.4500, 0.4833,  0.5167, 0.5500, 0.5834,
        0.6168, 0.6501, 0.6835, 0.7169, 0.7503, 0.7836,
        0.8171, 0.8506, 0.8842, 0.9180, 0.9527)
    }
    k <- 0

# Finding the Maximum and minimum correlation
```

```
    corr_min[[h]][l] <- cor(os.data[[h]],
        qnorm(c(min_mean_1, scps3, min_mean_2)))
    corr_max[[h]][l] <- cor(os.data[[h]],
        qnorm(c(max_mean_1, scps3, max_mean_2)))
      }
}

# Finding the Power
# n = 10
nm_countmax10_5 <- which(corr_max[[1]] <=  0.9108808)
nm_countmin10_5 <- which(corr_min[[1]] <=  0.908479)
nm_countmax10_10 <- which(corr_max[[1]] <= 0.9300996)
nm_countmin10_10 <- which(corr_min[[1]] <= 0.9285467)

# n = 15
nm_countmax15_5 <- which(corr_max[[2]] <=  0.9355293)
nm_countmin15_5 <- which(corr_min[[2]] <= 0.93552)
nm_countmax15_10 <- which(corr_max[[2]] <= 0.9480486)
nm_countmin15_10 <- which(corr_min[[2]] <= 0.947958 )

# n = 20
nm_countmax20_5 <- which(corr_max[[3]] <= 0.9487769)
nm_countmin20_5 <- which(corr_min[[3]] <= 0.9503622 )
nm_countmax20_10 <- which(corr_max[[3]] <= 0.9588988)
nm_countmin20_10 <- which(corr_min[[3]] <=  0.9598521 )

# n = 30
nm_countmax30_5 <- which(corr_max[[4]] <= 0.9630239)
nm_countmin30_5 <- which(corr_min[[4]] <= 0.9650149)
nm_countmax30_10 <- which(corr_max[[4]] <= 0.9701206)
nm_countmin30_10 <- which(corr_min[[4]] <=  0.9712342 )

# Combining the data
combined_count_10 <- c(length(nm_countmax10_5),
                       length(nm_countmin10_5),
                       length(nm_countmax10_10),
                       length(nm_countmin10_10) )
combined_count_15 <- c( length(nm_countmax15_5),
                       length(nm_countmin15_5),
                       length(nm_countmax15_10),
```

```
                          length(nm_countmin15_10) )
combined_count_20 <- c(length(nm_countmax20_5),
                       length(nm_countmin20_5),
                       length(nm_countmax20_10),
                       length(nm_countmin20_10) )
combined_count_30 <- c(length(nm_countmax30_5),
                      length(nm_countmin30_5),
                      length(nm_countmax30_10),
                      length(nm_countmin30_10) )

#-------------------------------------------------
#                 Data Example 1
#-------------------------------------------------


# Grid Search Method
#-------------------------------------------------

# Inputting the data
chlifed <- c(600,744,744,744,912,1228,1320,1464,1608,1896)

# Taking the log
chlifedlog <- log(chlifed)

# Putting in the SCP values
scps3 <-  c(0.1433, 0.2472, 0.3487, 0.4496, 0.5504, 0.6513,
                0.7528, 0.8567)

# Creating the sequence for the grid
scp1 <- 0.1433
scp2 <- 0.8567
scps1 <-seq(0.0001, scp1, by=0.0001)
scps2 <- seq(scp2, 0.9999, by=0.0001)

n1 <- length(scps1)
n2 <- length(scps2)

x = matrix(nrow = n1, ncol = n2)

# For loops to find the correlation at each i and j value
```

```
k <- 0
for(i in scps1){

  k <- k+1
  m <-0

  for(j in scps2) {
    m <- m+1
    x[k,m] = cor(chlifedlog, qnorm(c(i, scps3, j)))
  }
}

ind1max <- which(x==max(x), TRUE)
ind1min <- which(x==min(x), TRUE)
r.max<- x[ind1max]
r.min<- x[ind1min]


# Finding the P-value for the maximum
combinedmaxmin_10 <- do.call(rbind, maxmin_10)
min_10 <- combinedmaxmin_10$V2
min_gs_chlifed <- which(min_10 <= 0.8502242)
length_min_gs <- length(min_gs_chlifed)
p_value_min10 <- length_min_gs/10000

# Finding the P-value for the minimum
max_10 <- combinedmaxmin_10$V1
max_gs_chlifed <- which(max_10 <= 0.9744889)
length_max_gs <- length(max_gs_chlifed)
p_value_max10 <- length_max_gs/10000

#-----------------------------------------------
# Avergae Method
#-----------------------------------------------

# Putting in the modified SCP plotting points
min_mean_1 <- 0.07172864
min_mean_2 <- 0.9270685
max_mean_1 <- 0.06707189
max_mean_2 <- 0.9342107
```

```
# Finding the Correlation
corr_min <- cor(chlifedlog,
                          qnorm(c(min_mean_1, scps3, min_mean_2)))
corr_max <- cor(chlifedlog,
                          qnorm(c(max_mean_1, scps3, max_mean_2)))

# Finding the P-value for the maximum
average_min <- read.table("corr_min_10.csv")
min_a_10 <- which(average_min <= 0.9729964)
length_min_a <- length(min_a_10)
p_valuea_min10 <- length_min_a/10000

# Finding the P-value for the minimum
average_max <- read.table("corr_max_10.csv")
max_a_10 <- which(average_max <= 0.972244)
length_max_a <- length(max_a_10)
p_valuea_max10 <- length_max_a/10000


#-----------------------------------------------
# Data Example 2
#-----------------------------------------------

#-----------------------------------------------
# Grid Search Method
#-----------------------------------------------

# inputting the data
srs30 <- c(79.89, 88.13, 90.03, 92.56, 95.97, 99.62, 103.56,
        105.48, 111.38, 113.90, 85.29, 89.33, 91.46, 95.14,
        96.20, 102.56, 103.60, 106.82, 112.97, 115.95, 87.83,
        89.35, 92.55, 95.94, 98.70, 103.22, 104.21, 108.39,
        113.75, 118.52)

# SCPs for n= 30
scps3 <- c(0.0473, 0.0820, 0.1158, 0.1494, 0.1829, 0.2164, 0.2497,
           0.2831, 0.3165, 0.3499, 0.3832, 0.4166, 0.4500, 0.4833,
           0.5167, 0.5500, 0.5834, 0.6168, 0.6501, 0.6835, 0.7169,
           0.7503, 0.7836, 0.8171, 0.8506, 0.8842, 0.9180, 0.9527)
```

```
# Creating the sequence for the grid search
scp1 <- 0.0473
scp2 <- 0.9527
scps1 <-seq(0.0001, scp1, by=0.0001)
scps2 <- seq(scp2, 0.9999, by=0.0001)

n1 <- length(scps1)
n2 <- length(scps2)

x = matrix(nrow = n1, ncol = n2)

# For loops to find the correlation at each i and j value
k <- 0
for(i in scps1){

  k <- k+1
  m <-0

  for(j in scps2) {
    m <- m+1
    x[k,m] = cor(sort(srs30), qnorm(c(i, scps3, j)))

  }
}

ind1max <- which(x==max(x), TRUE)
ind1min <- which(x==min(x), TRUE)
r.max<- x[ind1max]
r.min<- x[ind1min]


# Finding the maximum p-value
combinedmaxmin_30 <- do.call(rbind, maxmin_30)
min_30 <- combinedmaxmin_30$V2
min_gs_chlifed <- which(min_30 <= 0.9349074 )
length_min_gs <- length(min_gs_chlifed)
p_value_min30 <- lapply(length_min_gs, function(r) {r/10000})

# Finding the Minimum p-value
```

```
max_30 <- combinedmaxmin_30$V1
max_gs_chlifed <- which(max_30 <= 0.9905121)
length_max_gs <- length(max_gs_chlifed)
p_value_max30 <- lapply(length_max_gs, function(r) {r/10000})

#-------------------------------------------------
# Average method
#-------------------------------------------------
# Putting the modified SCP Values
min_mean_1 <- 0.01587896
min_mean_2 <- 0.9840833
max_mean_1 <- 0.02377405
max_mean_2 <- 0.9765869

srs30sort <- sort(srs30)

# Finding the maximal and minimal correlation
corr_min <- cor(srs30sort,
                qnorm(c(min_mean_1, scps3, min_mean_2)))
corr_max <- cor(srs30sort,
               qnorm(c(max_mean_1, scps3, max_mean_2)))

# Finding the maximum P-value
average_min <- read.table("corr_min_30.csv")
min_a_30 <- which(average_min <= 0.988209)
length_min_a <- length(min_a_30)
p_valuea_min30 <- lapply(length_min_a, function(r) {r/10000})

# Finding the minimum p-value
average_max <- read.table("corr_max_30.csv")
max_a_30 <- which(average_max <= 0.9899287)
length_max_a <- length(max_a_30)
p_valuea_max30 <- lapply(length_max_a, function(r) {r/10000})



#-------------------------------------------------
# Data Example 3
#-------------------------------------------------
```

```
#-------------------------------------------------------
# Grid Search Method
#-------------------------------------------------------

# Inputting the data
data <- c(0.004, 0.304, 0.612, 0.748, 0.771, 0.806, 0.850,
                0.885, 0.906, 0.977)

# SCP values for n = 10
scps3 <-  c(0.1584, 0.2542, 0.3521, 0.4506, 0.5494, 0.6479,
 0.7458, 0.8416)

# Creating the sequence for the grid search
scp1 <- 0.1584
scp2 <- 0.8416
scps1 <-seq(0.0001, scp1, by=0.0001)
scps2 <- seq(scp2, 0.9999, by=0.0001)

n1 <- length(scps1)
n2 <- length(scps2)

x = matrix(nrow = n1, ncol = n2)

# For loops to find the correlation at each i and j value
k <- 0
for(i in scps1){

  k <- k+1
  m <-0

  for(j in scps2) {
    m <- m+1
    x[k,m] = cor(sort(data), qunif(c(i, scps3, j)))

  }
}

ind1max <- which(x==max(x), TRUE)
ind1min <- which(x==min(x), TRUE)
r.max<- x[ind1max]
```

```
r.min<- x[ind1min]


# Finding the p-value for the maximum
combinedmaxmin_10 <- do.call(rbind, maxmin_10)
min_10 <- combinedmaxmin_10$V2
min_gs_data<- which(min_10 <= 0.8314281)
length_minu_gs <- length(min_gs_data)
p_value_min10 <- length_minu_gs/10000

# Finding the p-value for the minimum#
max_10 <- combinedmaxmin_10$V1
max_gs_chlifed <- which(max_10 <=  0.9085895)
length_max_gs <- length(max_gs_chlifed)
p_value_max10<- length_max_gs/10000

#-----------------------------------------
# Average Method
#-----------------------------------------

# Putting in the modified SCP Values
min_mean_1 <- 0.0847
min_mean_2 <- 0.9225
max_mean_1 <- 0.0779
max_mean_2 <- 0.9149

# Finding the Correlation
corr_min <- cor(sort(data),
        qunif(c(min_mean_1, scps3, min_mean_2)))
corr_max <- cor(sort(data),
        qunif(c(max_mean_1, scps3, max_mean_2)))

# Finding the p-value for the maximum
average_min <- read.table("corr_min_10.csv")
min_a_10 <- which(average_min <= 0.8735113)
length_min_a <- length(min_a_10)
p_valuea_min10 <- length_min_a/10000

# Finding the p-value for the minimum
average_max <- read.table("corr_max_10.csv")
```

```
max_a_10 <- which(average_max <= 0.8769523)
length_max_a <- length(max_a_10)
p_valuea_max10 <- length_max_a/10000
```

Bibliography

Ahmadi, J. and Balakrishnan, N. (2009). Pitman closeness of record values to population quantiles. *Statistics & Probability Letters*, 79(19):2037–2044. (Cited on page 23.)

Ahmadi, J. and Balakrishnan, N. (2013). Simultaneous closeness of k-records. *Communications in Statistics-Simulation and Computation*, 42(1):107–120. (Cited on pages 24 and 25.)

Arnold, B. C., Balakrishnan, N., and Nagaraja, H. N. (1992). *A first course in order statistics*. John Wiley & Sons, New York. (Cited on pages 29 and 31.)

Balakrishnan, N., Davies, K., and Keating, J. P. (2009a). Pitman closeness of order statistics to population quantiles. *Communications in Statistics-Simulation and Computation*, 38(4):802–820. (Cited on pages 1, 5 and 12.)

Balakrishnan, N., Davies, K., Keating, J. P., and Mason, R. L. (2012a). Computation of optimal plotting points based on Pitman closeness with an application to goodness-of-fit for location-scale families. *Computational Statistics & Data Analysis*, 56(9):2637–2649. (Cited on pages 2, 34, 39, 45, 67 and 69.)

Balakrishnan, N., Davies, K. F., Keating, J. P., and Mason, R. (2010). Simultaneous closeness among order statistics to population quantiles. *Journal of Statistical Planning and Inference*, 140(9):2408–2415. (Cited on pages 1, 16, 17, 21 and 46.)

Balakrishnan, N., Davies, K. F., Keating, J. P., and Mason, R. L. (2011a). Correlation-type goodness of fit test for extreme value distribution based on simultaneous closeness. *Communications in Statistics - Simulation and Computation*, 40(7):1074–1095. (Cited on page 43.)

Balakrishnan, N., Davies, K. F., Keating, J. P., and Mason, R. L. (2011b). Pitman closeness, monotonicity and consistency of best linear unbiased and invariant estimators for exponential distribution under type II censoring. *Journal of Statistical Computation and Simulation*, 81(8):985–999. (Cited on pages 4, 6 and 8.)

Balakrishnan, N., Davies, K. F., Keating, J. P., and Mason, R. L. (2012b). Pitman closeness comparison of best linear unbiased and invariant predictors for exponential distribution in one-and two-sample situations. *Communications in Statistics-Theory and Methods*, 41(1):1–15. (Cited on page 9.)

Balakrishnan, N., Iliopoulos, G., Keating, J., and Mason, R. (2009b). Pitman closeness of sample median to population median. *Statistics & Probability Letters*, 79(16):1759–1766. (Cited on pages 1 and 7.)

Blyth, C. R. (1972). Some probability paradoxes in choice from among random alternatives. *Journal of the American Statistical Association*, 67(338):366–373. (Cited on page 16.)

Castillo, E., Hadi, A., Balakrishnan, N., and Sarabia, J. (2005). Extreme value

and related models in engineering and science applications. *John Wiley & Sons, Hoboken, New Jersey*, 179:180. (Cited on pages 31 and 33.)

Cochran, W. G. (1952). The $\chi 2$ test of goodness of fit. *The Annals of Mathematical Statistics*, pages 315–345. (Cited on page 28.)

Filliben, J. J. (1975). The probability plot correlation coefficient test for normality. *Technometrics*, 17(1):111–117. (Cited on pages 2, 37 and 38.)

Ghosh, M. and Sen, P. K. (1991). Bayesian Pitman closeness. *Communications in Statistics-Theory and Methods*, 20(11):3659–3678. (Cited on page 5.)

Huber-Carol, C. (2002). *Goodness-of-fit Tests and Model Validity*. Birkhauser, Boston. (Cited on pages 27 and 28.)

Keating, J. P., Mason, R. L., and Sen, P. K. (1993). *Pitman's measure of closeness: a comparison of statistical estimators*. Siam, Philadelphia. (Cited on page 5.)

Kinnison, R. (1989). Correlation coefficient goodness-of-fit test for the extreme-value distribution. *The American Statistician*, 43(2):98–100. (Cited on pages 38 and 39.)

Lilliefors, H. W. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402. (Cited on page 28.)

Makkonen, L. (2008). Bringing closure to the plotting position controversy. *Communications in Statistics-Theory and Methods*, 37(3):460–467. (Cited on page 33.)

Marden, J. I. (2004). Positions and QQ plots. *Statistical Science*, 19(4):606–614. (Cited on page 30.)

Massey Jr, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78. (Cited on pages 28 and 29.)

Nelson, W. B. (2005). *Applied life data analysis*. John Wiley & Sons, Hoboken, New Jersey. (Cited on page 65.)

Pitman, E. J. (1937). The closest estimates of statistical parameters. In *Proceedings of the Cambridge Philosophical Society*, volume 33, pages 212–222. (Cited on page 4.)

Stephens, M. A. and D'Agostino, R. B. (1986). *Goodness-of-fit techniques*. Marcel Dekker. (Cited on pages 67, 69 and 71.)

Volterman, W., Davies, K. F., and Balakrishnan, N. (2012). Pitman closeness as a criterion for the determination of the optimal progressive censoring scheme. *Statistical Methodology*, 9(6):563–572. (Cited on page 12.)

Volterman, W., Davies, K. F., and Balakrishnan, N. (2013). Simultaneous Pitman closeness of progressively type-II right-censored order statistics to population quantiles. *Statistics*, 47(2):439–452. (Cited on page 20.)

Weibull, W. (1939). A statistical theory of strength of materials', iva-handl. (Cited on page 32.)

Wilk, M. B. and Gnanadesikan, R. (1968). Probability plotting methods for the

analysis for the analysis of data. *Biometrika*, 55(1):1–17. (Cited on pages 30 and 31.)

Index