جامعة بيرزيت

BIRZEIT UNIVERSITY

Machine learning approach to test the normality of the data

Prepared by:

Hussein M. Soboh

1185061

Committee:

Dr. Hassan Abu Hassan	Advisor
Dr. Tareq Sadeq	Member
Dr. Radi Jarrar	Member

Submitted in partial fulfillment of the requirements of the "Master Degree in Applied Statistic and Data Science" from the faculty of Graduate Studies at Birzeit University - Palestine

جامعة بيرزيت

BIRZEIT UNIVERSITY

Machine learning approach to test the normality of the data

Prepared by:

Hussein Soboh

1185061

Committee:

Dr. Hassan Abu Hassan	Advisor
Dr. Tareq Sadeq	Member
Dr. Radi Jarrar	Member

Submitted in partial fulfillment of the requirements of the "Master Degree in Applied Statistic and Data Science" from the faculty of Graduate Studies at Birzeit University - Palestine

To my family

Table of Content

List of Tables

List of Figures

Abstract

Normality tests are very important in statistical inference; their purpose is to know if the data is sampled from the normal population. The normality of the data is a prerequisite for several parametric statistics such as t-test, ANOVA, and regression analysis. Violation of the normality assumption may yield incorrect results and wrong decisions. There are many tests available to detect departure from normality for a random sample. But these tests sometimes lead to contradicting results. Moreover, some of them can be applied under certain conditions. In this research, we build a machine learning classification model to predict the "normality" of the data using several features: size, skewness, kurtosis, median, and percentage of data lies within 1, 2, and 3 standard deviations. To find the best classification technique that fits our data, three models created using three classification techniques: Random Forest (RF), Gradient Boosting Machines (GBM), and Support Vector Machines (SVM). The evaluation phase showed high accuracy and ROC_AUC for the three models with few points in favor of the (RF) model. Power comparison was also executed for (RF) model against seven statistical tests: Shapiro-Wilk (SW), Anderson-Darling (AD), Jarque-Bera (JB), Shapiro-Francia (SF), Kolmogorov-Smirnov (KS), Cramer-von Mises (CVM), and Lilliefors (Lillie). The comparison concluded using a Monte Carlo simulation on 25 alternative distributions on different sample sizes. The results showed significantly higher power for the model comparing to the other normality tests.

ملخص

تعد اختبارات التوزيع الطبيعي(Normal distributions tests) مهمة للغاية في الاستدلال الاحصائي، والغرض منها هو معرفة ما إذا كانت البيانات مأخوذه من مجتمع توزيعه يتبع للتوزيع الطبيعي. التوزيع الطبيعي للبيانات هو شرط أساسي لعدة إحصاءات مثل: t-test, ANOVA, regression analysis. عدم تحقق هذا الشط يمكن أن يؤدي الى نتائج و قرارات خاطئة. توجد العديد من االاختبارات التي تستخدم لهذا الغرض ولكنها في اغلب الاحيان تؤدي الى نتائج متناقضة. وبعضها فعاليتها مشروطة على ظروف عدة للعينة مثل حجم العينة. الهدف الرئيس من هذا البحث هو استخدام تقنيات تعلم الالة لبناء نموذج يمكن أن يكون ذا جوده جيدة مقارنة باالاختبارات الحالية. يحاول هذا البحث إنشاء نموذج تصنيف باستخدام صفات عدة للبيانات مثل حجم العينة والانحراف والتفلطح والوسيط والنسبة المئوية للبيانات التي تقع ضمن 1 و 2 و 3 انحرافات معيارية. انحرافات معيارية و 2 و 3. للعثور على أفضل أسلوب تصنيف يناسب بياناتنا ، تم إنشاء ثلاثة نماذج باستخدام ثلاث تقنيات تصنيف: Random Forest (RF)، Gradient Boosting Machines (GBM), and Support Vector Machines (SVM). أظهرت النتائج دقة تصنيف عالية و قيم ROC_AUC عالية للنماذج الثلاثة مع أفضلية بسيطة لصالح نموذج (RF). تمت مقارنة الاختبار الناتج من هذا البحث مع عدة اختبارات اخرى تستخدم لهذا الغرض مثل: Shapiro-Wilk (SW), Anderson-Darling (AD), Jarque-Bera (JB), Shapiro-Francia (SF), Kolmogorov-Smirnov (KS), Cramer-von Mises (CVM)، و Lilliefors (Lillie). تمت المقارنة بأسلوب ال "Power test" باستخدام محاكاة "MonteCarlo Simulation" على 25 توزيع لا ينتمون للتوزيع الطبيعي في أحجام عينة مختلفة, وأظهرت النتائج بشكل ملحوظ القدرة الأعلى للاختبار الجديد مقارنة بالاختبارات الأخرى.

Chapter One

Introduction

The normal distribution is an underlying assumption of many statistical procedures. Parametric tests such as correlation, regression, t-tests, and analysis of variance are based on the assumption that the data follows a normal distribution. When the assumption does not hold, it is hard to draw accurate and reliable conclusions about the data (Ghasemi & Zahediasl, 2012). Visual plots such as P-P plot and statistical tests such as Shapiro-Wilk, Chi-square, D'Agostino-Pearson, Jarque-Bera, and others are the classical methods usually used to detect non-normality (Das & Imon, 2016).

Some of the existing normality tests can only be applied under certain conditions. For example, the Shapiro-Wilk test has a limitation on the size of the sample where it does not perform well on samples with size more than 50 (Shapiro, Wilk, & Chen, 1968). Moreover, different tests of normality often produce different results[1]. The contradicting results are misleading and often confuse statisticians.

In this research, we try to leverage the power of machine learning techniques to build a new test that could be with comparable performance with the existing tests. Machine learning offers the ability to build a model that learns from experience. By providing examples of normal (negative) and non-normal (positive) examples, the model can learn the characteristics of each of these classes to a level that it can classify correctly new examples to the correct normality class (Bishop, 2006).

[1] Several comparisons between the normality test described in section 2.2

1.1 Background

The normal distribution, also known as Gaussian distribution is one type of continuous probability distributions. It appears as a bell curve (**Figure 1**) where it is symmetric about its mean, which is identical to its mode and median. 68%, 95%, and 99% of the data fall within 1, 2, and 3 standard deviations respectively (Patel & Read, 1996).

(Forbes, Evans, Hasting, & Peacock, 2011) The normal distributions have the following density function, usually noted as $N(\mu, \sigma^2)$:

$$f(x;\ \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, -\infty \leq x \leq \infty$$

Where μ is the mean, and σ is the standard deviation. **Figure 1** shows the p.d.f of the distribution of multiple examples of μ and σ^2.

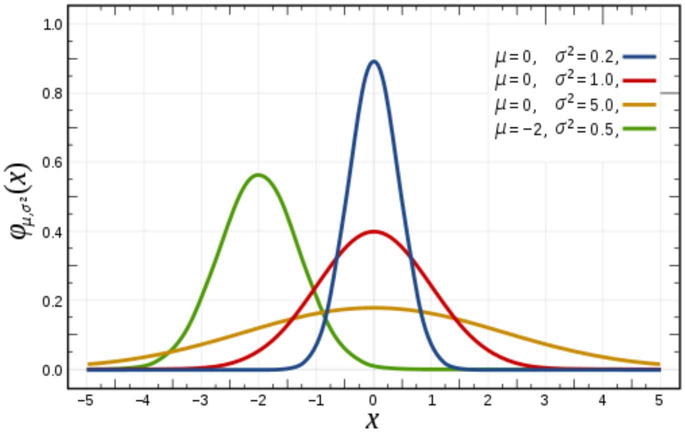

Figure 1: PDF of Normal distribution (Normal distribution, 2020)

The normal distribution is special as its two parameters (μ and σ^2) are mutually independent and provide us with complete information on the shape and location of the distribution (Casella & Berger, 2001). The independence of the two parameters characterizes the normal distribution from other distributions (Lukas, 1942). The normal distribution is unimodal and it has two inflection points located 1 standard deviation from the mean (Patel & Read, 1996).

If $X \sim N(\mu, \sigma^2)$, then the random variable $Z = \frac{X-\mu}{\sigma}$ has an N(0,1) distribution, known as standard normal distribution and it is described by p.d.f

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

This function is symmetric around $x = 0$, where it attains its maximum value $1/\sqrt{2\pi}$ and has inflection points at $x = 1$ and $x = -1$ (Casella & Berger, 2001).

Normal distributions are the most importantly used in natural and social sciences to represent random variables. Quantities such as examination grades, snowflakes sizes, and other phenomena are approximated the normal probability density function (Lyon, 2014). The importance is mainly due to the central limit theorem, which states that the sum of independent and identically distributed random variables converges to a normal distribution as the number of samples increases regardless of the type of distribution of the sampled variables. This theorem provides theoretical bases for why so many variables we see in nature appear to have approximately a normal probability distribution (Hazewinkel, 1994).

Normality tests are used to determine if the data is sampled from Normal distribution. The normality of the data is an assumption need to be verified before applying several parametric statistics such as t-test, linear regression analysis, discernment analysis, and analysis of Variance (ANOVA). When the assumption is violated, the accuracy of the conclusions about the data is questionable and not reliable (Ghasemi & Zahediasl, 2012).

The normality test assess the likelihood that a given data set $\{x_1, \dots, x_n\}$ comes from the normal distribution. The null hypothesis $H_0$is that the observations are distributed normally versus the alternative $H_a$that the observations are not distributed normally. There are two sets of methods that can be used to examine normality, visual methods, and statistical test methods (Ghasemi & Zahediasl, 2012).

Visual plots such as the P-P plot are useful to visualize the distribution of the data but they usually not enough to conclude decisions about the normality of the data. Hence, a variety of statistical tests have been developed in this area such as Shapiro-Wilk, Anderson Darling, Kolmogorov-Smirnov tests, and others. These tests are parametric aims to measure the probability of departure from normality for the data set on different significant levels.

1.2 Problem definition

The departure from normality is very critical in statistical inference. Biased interpretation can be inferred if the normality assumption is violated. Normality tests have traditionally been designed as classical statistical hypothesis testing procedures and, to the best of our knowledge, this has been the only way used so far to find a departure from normality.

The long list of tests developed in the literature can make it hard for statisticians to select the appropriate test to use[2]. Moreover, these statistical tests are sensitive to the size of the data as shown in the study of (Oztuna, Elhan, & Tuccar, 2006).

In this research, we are proposing a new approach to testing normality. In this approach, we use the Machine Learning tools to develop a classification model that can classify the sample data to the correct underlying distribution with less sensitivity to the nature of the underlying distribution of the data.

[2] The tests and their details are explained in section 2.1 in this document

1.3 Research Objectives

In this research, we propose a new approach to testing the normality of the data using Machine Learning (ML). Machine learning algorithms build a mathematical model based on sample data, known as "training data", to make predictions or decisions without being explicitly programmed to do so. This approach is known as supervised machine learning. Classification is one type of supervised machine learning where the human provides the algorithm with pairs of inputs and desired outputs, and the algorithm learn a general rule to produce the desired output given an input it has never seen before (Mueller & Guido, S, 2016).

The idea of using machine learning in testing the normality was not explored in previous literature we read as of the date of writing this research. In this research, we build a model that learns the properties and the characteristics of both the normal and the alternative distributions by providing examples of both classes. We expect the model to get enough experience to be able to correctly classify the normality of the data regardless of the sample size and the underlying distribution. One advantage of this approach compared to the classical tests is that it can provide us with additional measures to the power of the test. The power measures the ability to detect one type of the classes – the non-normality- while in the classification models, additional quality metrics are available to measure the performance on detecting the two classes, such as Accuracy, Specificity, and Sensitivity.

1.4 Limitations of the study

Results and conclusions from a Monte Carlo simulation in comparing powers across various distributions are seriously limited in generalizability beyond those distributions. The generalizability of the results depends on the design and how much coverage of different probability distributions is included in the study. In chapter 3, we show a wide range of alternative distributions added to the scope of the research by which we expect this offers a greater potential for generalizing results comparing to the distributions used in previous studies.

Related to that, the generalizability of the proposed model could be questionable; results and conclusions of any classification model are limited to the data set it trained with (Cai, et al., 2020). In chapter 3 we try to overcome this limitation by having enough representations of the distributions in the training and by building a model from a set of features resilience to the change in the type of distribution such as skewness and kurtosis.

Another limitation of this study is the choice of power as the base measure to compare our model against other statistical tests. This comparison is limited to only one of the two sides of the quality of any classification model. The Power which stands for "Recall" in machine learning terminology, evaluates the performance of detecting the positive class –alternative class in our use case- and does not evaluate how the model performs in detecting negative class -normal class in our use case. This is because the classical normality tests are statistical tests; if the test does not have evidence to reject the null hypothesis (the sample has normal distribution), it does not mean it accept it. This limitation prevents us from using other quality measures such as Accuracy and F-Measure to compare the quality of the classifier against other tests on both normal and alternative classes.

Chapter Two

Literature review

2.1 Normality tests

A large number of methods and tests available to detect departure from normality where each test has its characteristics and power. We can look for departure from normality using two ways: Visual methods of normal plots or significant tests (Ghasemi & Zahediasl, 2012).

2.1.1 Visual tests

The researcher can validate the normality of the data using graphical methods such as P-P plot, Q-Q plot, histogram, box plot, or stem-and-leaf plot. These plots are useful to visualize the distribution of the data but they often do not provide reliable evidence about the normality of the data. The plots are subjective, a plot can be interpreted into different levels of “normality” by different people. Moreover, judging using these visual methods required enough statistical experience of the researcher to take a correct decision. These imply to use more formal and reliable tests (Yap & Sim, 2011).

2.1.2 Statistical tests

The effort of developing normality tests was initiated by (Pearson, 1895) who used the skewness and kurtosis as indicators of departure from normality. The number of different tests for normality seems to be boundless. The researchers classified the tests in different ways. In this section we present the tests by classifying them into four main groups as following:

- **Empirical Distribution Function (EDF) tests**: These tests involve measuring the discrepancy between the cumulative distribution function of the normal distribution and the empirical distribution function of the sample (D’Agostino & Stephens, 1986). The most popular tests of this type: Kolmogorov-Smirnov (KS) test (1933), Cramer-von Mises (CVM) test, and Anderson-Darling (AD) test. The Anderson-Darling (AD) test (1974) is the recommended one in this family (D’Agostino & Stephens, 1986). KS test is highly sensitive to extreme values, and it has low power and it should not be used in testing normality (Throde, 2002).
- **Moments tests**: These tests use the skewness and the kurtosis (the second and the third moments respectively) of the sample to calculate the test statistic (D’Agostino & Stephens, 1986). Popular tests are Jarque-Bera (JB) test (1975) and the D’Agostino-Pearson Omnibus test (DP) (1973).
- **Regression and correlation tests**: The tests are based on the correlation between the empirical data and corresponding scores under normality (D’Agostino & Stephens, 1986). Shapiro-Wilk (SW) (1965) test is the popular one in this family. It has good power for sample sizes up to 50. For large samples, the computation of its test statistic is much complicated (Das & Imon, 2016). Other tests in this group are the Shapiro-Francia (SF) test and Ryan-Joiner test
- **Chi-Squared test**: It is not recommended for continuous distributions as it computes the number of observations instead of the observations themselves when calculating the test statistic. The chi-Squared test should not be used (D’Agostino & Stephens, 1986).

2.2 Previous comparisons

The literature shows many attempts to compare different normality tests trying to find the best performing one. Most of the comparisons are based on comparing the power of the tests on the alternative distributions using Monte Simulation on different alternatives with different sample sizes and levels of significance. The results have a lot of variation.

(Shapiro, Wilk, & Chen, 1968) Indicates that SW (Shapiro and Wilk 1965) has the best power comparing to $\sqrt{b_1}$ (statndard third moment), b_2(standard fourth moment), Kolmogorov-Smirnov, Cramer-Von Mises, Weighted CM, Modified KS, chi-squared, and u (Studentized range) on alternatives of sample size (10, 15, 20, 35, 50).

In (Muyombya, 2017) study that examined the power of the tests on large sample sizes, Kolmogorov-Smirnov was the most powerful normality test regardless of the nature of the distribution. Followed by Shapiro-Wilk, Shapiro-Francia, Anderson-Darling, Jaque-Bera, and D'Agostino-Pearson.

(Alizadeh & Arghami, 2011) Compared the power of several tests and concluded that Jaque-Bera is the most powerful test for symmetric distributions and Shapiro-Wilk is the most powerful for asymmetric distributions with support$(-\infty, \infty)$. It also reveals Kolmogorov-Smirnov and Shapiro-Wilk have the best power for alternatives supported by $(0, \infty)$

A study by (Islam, Normality Testing- A New Direction, 2011) compared tests to ensure the validity of the t-statistic used to assessing the significance of the regressors. It shows that Anderson-Darling is the best option comparing to Jarque-Bera, D'Agostino and Pearson, and Lilliefors (a modification of Kolmogorov-Smirnov test).

(Razali & Wah, 2011) Compared the power of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors, and Anderson-Darling. Shapiro-Wilk was the most powerful test then Anderson-

Darling, Lilliefors, and Kolmogorov-Smirnov on both symmetric and asymmetric alternatives. This research also reveals that these tests have low power in a small sample size (less than 30).

(Islam, Ranking of Normality Tests: An Appraisal through Skewed Alternative Space, 2019) Evaluated the performance of several tests by using a proposed stringency framework of comparing tests. The research compares Kolmogorov-Smirnov, Anderson-Darling, Jaque-Bera, Shapiro-Wilk, D'Agostino, Coin (COIN), Bonett, and Seier test (T_w). And he recommends to use T_w test for slightly skewed, Anderson-Darling and Shapiro-Wilk for moderately skewed, and all except COIN and T_w for highly skewed alternatives.

(Afeez, 2018) Compared several tests on five classes of alternatives: Near Normal, Symmetric long-tailed, Symmetric short-tailed, Asymmetric long-tailed, and Asymmetric short-tailed. SW had good power in a wide range of alternatives comparing to Anderson-Darling, Cramer–von Mises, Jaque-Bera, Chi-Square tests. Jaque-Bera was poor for symmetric short tails, but it is appropriate for symmetric long-tailed distributions.

(Seier, 2002) Claimed that Tests based on skewness and kurtosis are not powerful against symmetric alternative distributions where the kurtosis is close to that of the normal distribution. These tests are more powerful when the alternative is more peaked than normal.

Some of the studies and investigations share similar results. For example, Shapiro-Wilk was in a good rank in some of them, but it was not recommended in others. Having a clear answer to the best performing test seems a very complicated task.

2.3 Limitations of the statistical tests

A large number of comparisons with different results confuse the researcher on which normality test to apply where dozens of tests are available to use. Based on what we show from some of the previous literature, no single test is uniformly more powerful than others.

Comparing the tests based on their power using simulation didn’t succeed having an answer on what is the best test to use, as each test has its area of strengths and weaknesses. The power of the tests depends critically on two factors: The alternative, which can’t be specified when doing the test, and as we saw that the same test has different powers when applied on different distributions. The other factor is the sample size, which is critical as well since the normality tests will always reveal non-normality as the sample size grows. (Oztuna, Elhan, & Tuccar, 2006) Show that for small sample size, the normality tests have small power to reject the null hypothesis when it should be rejected. And for large sample sizes, the normality tests become much sensitive and the test can be significant even in case of a small deviation from normality.

Chapter Three

Methodology

In this research, we propose a new approach to testing normality using state of the art ML techniques. In this chapter, we will start explaining the different steps to be executed to build and evaluate the classification model. Then we describe the method that is used in comparing the quality of the “new test” against other popular statistical tests of normality.

3.1 Alternative Distributions

Alternative distributions can be classified into five major families based on the distribution skewness and kurtosis: asymmetric long-tailed (ALT), asymmetric short-tailed (AST), symmetric long-tailed (SLT), symmetric short-tailed (SST), and close to normal (CTN) (Shapiro, S. & Wilk, B. & Chen, J. 1968). The alternative distributions used in this study were selected from these families on different levels of parameters to cover a wide range of data. Five instances from each family are chosen as shown in **Table 1**, and an overview of the corresponding probability distributions is provided later in this section. The alternatives will be used in the proposed model as positive examples, and also used in the later phase of comparing the power of the new test against other statistical tests.

Table 1: Alternative distributions used in the research

Family	Alternatives				
Asymmetric_Long_Tailed (ALT)	Weibull(0.5, 1)	Weibull(2, 1)	LogNormal (0, 1)	$\chi^2(4)$	$\chi^2(10)$
Asymmetric_Short_Tailed (AST)	Beta(2, 1)	Beta(3, 2)	LogNormal (0, 0.15)	LogNormal (0, 0.25)	LogNormal (0, 0.35)
Symmetric_Long_Tailed (SLT)	t(1)	t(2)	t(4)	t(7)	Tukey (10)
Symmetric_Short_Tailed (SST)	Uniform(0,1)	Beta(1.3, 1.3)	Beta(1.5, 1.5)	Tukey(1.5)	Truncated normal (-2, 2)
Close_To_Normal (CTN)	Tukey (0.1)	Tukey (0.2)	Tukey (5)	t (10)	Laplace(0, 10)

3.1.1 Beta Distribution

(Walck, 2007) (Forbes, Evans, Hasting, & Peacock, 2011) The Beta distribution denoted by $Beta(\alpha, \beta)$ is a continuous distribution given by:

$$f(x; v, \omega) = \frac{x^{v-1}(1-x)^{\omega-1}}{B(v, \omega)} \; 0 \leq x \leq 1$$

Where the quality $B(v, \omega)$ is the Beta function defined in terms of Gamma function as:

$$B(v, \omega) = \frac{\Gamma(v)\Gamma(\omega)}{\Gamma(v+\omega)}$$

For v = ω = 1, the Beta distribution simply becomes a uniform distribution between zero and one.

The mean and the variance of the Beta distribution given by

$$E(X) = \frac{v}{v+\omega}$$

$$Var(X) = \frac{v\omega}{(v+\omega)^2(v+\omega+1)}$$

Figure 2 shows the Beta distribution on different levels of v and ω.

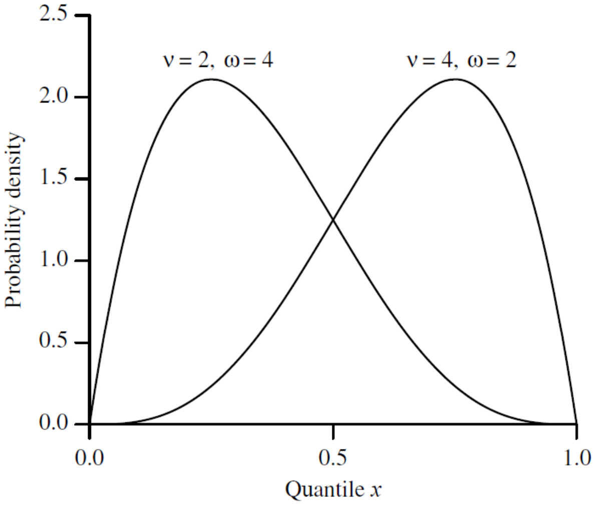

Figure 2: Probability density function for Beta variate ***β****: ν, ω*

3.1.2 Student t-distribution

(Walck, 2007) (Forbes, Evans, Hasting, & Peacock, 2011) The Student's t-distribution (or simply the t-distribution) denoted by $t(v)$ is given by

$$f(t; v) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{v\pi}\Gamma\left(\frac{\mathrm{v}}{2}\right)}(1 + \frac{x^2}{v})^{-\frac{v+1}{2}}; v > 0$$

Where $v = n - 1$ is the degrees of freedom and t is a real number. The functions of Γ and β are the usual Gamma and Beta functions. The mean of t-distribution is 0 for $v > 1$, otherwise undefined. The variance is given by

$$Var(X) = \begin{cases} \frac{v}{v-2}, & v > 2 \\ \infty, & 1 < v \leq 2 \\ undefined, & otherwise \end{cases}$$

Figure 3 shows the t distribution of different values of v.

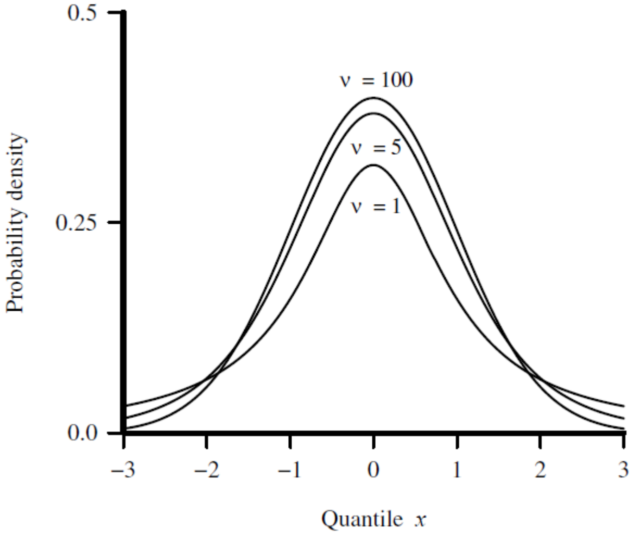

Figure 3: Probability density function for Student's t variate, ***t:*** *ν*

3.1.3 Chi-squared Distribution

(Walck, 2007) (Forbes, Evans, Hasting, & Peacock, 2011) The chi-squared distribution denoted by $\chi^2(v)$ with v degrees of freedom is the distribution of a sum of the squares of v independent standard normal random variables. Where a set of data is represented by a theoretical model, the chi-squared distribution can be used to test the goodness of fit between the observed data points and the values predicted by the model, subject to the differences being normally distributed. It is given by

$$f(x;v) = \frac{\left(\frac{x}{2}\right)^{\frac{v}{2}-1} e^{-\frac{x}{2}}}{2\Gamma\left(\frac{v}{2}\right)}; x \geq 0$$

The mean of the chi-squared distribution is equal to the degrees of freedom v, and the variance is double the mean = $2v$. **Figure 4** shows the distribution of different values of v.

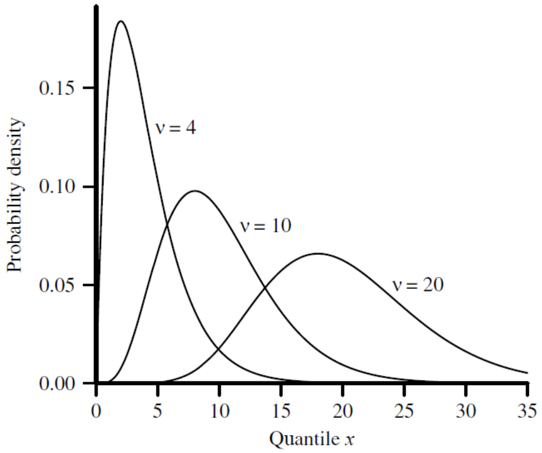

Figure 4: Probability density function for the Chi-Squared variate χ^2: v

3.1.4 Log-normal Distribution

(Walck, 2007) (Forbes, Evans, Hasting, & Peacock, 2011) The Log-normal distribution denoted by $Lognormal(\mu, \sigma^2)$ is a continuous distribution of a random variable whose logarithm is normally distributed. The lognormal distribution applies to random variables that are constrained by zero but have a few very large values. The resulting distribution is asymmetrical and positively skewed. It is given by

$$f(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln x-\mu}{\sigma}\right)^2}; \qquad x,\mu,\sigma > 0$$

An Alternative parameter of scale is m where $m = e^{\mu}$. The mean and the variance of the Log-normal distribution are given by

$$E(X) = e^{\mu+\frac{\sigma^2}{2}}$$

$$Var(X) = (e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$$

Figure 5 shows the Log-normal distribution on different values m and σ.

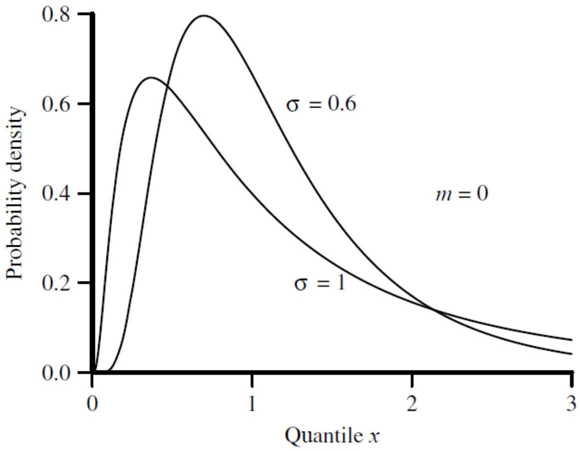

*Figure 5: Probability density function for the Log-normal variate **L**: m, σ*

3.1.5 Weibull Distribution:

(Walck, 2007) (Forbes, Evans, Hasting, & Peacock, 2011) The Weibull distribution denoted by $weibull(k, \lambda)$ is named after the Swedish physicist Waloddi Weibull (1887-1979) who described it in detail in 1951. Weibull variate is commonly used as a lifetime distribution in reliability applications. The two-parameter Weibull distribution can represent decreasing, constant, or increasing failure rates. The β parameter is the shape parameter, and η is simply a scale parameter and the variable $y = x/\eta$ has distribution

$$g(y) = \beta y^{\beta-1} e^{-y^{\beta}}$$

The Weibull distribution is given by

$$f(x; \beta, \eta) = \frac{\beta}{\eta}\left(\frac{x}{\eta}\right)^{\beta-1} e^{-\left(\frac{x}{\eta}\right)^{\beta}}; x \geq 0$$

The mean and the variance of the Weibull distribution are given by

$$E(X) = \eta\, \Gamma\left(1 + \frac{1}{\beta}\right)$$

$$Var(X) = \eta^2 \left[\Gamma\left(1+\frac{2}{\beta}\right) - \left(\Gamma\left(1+\frac{1}{\beta}\right)\right)^2 \right]$$

Figure 6 shows the Weibull distribution on different levels of β and η.

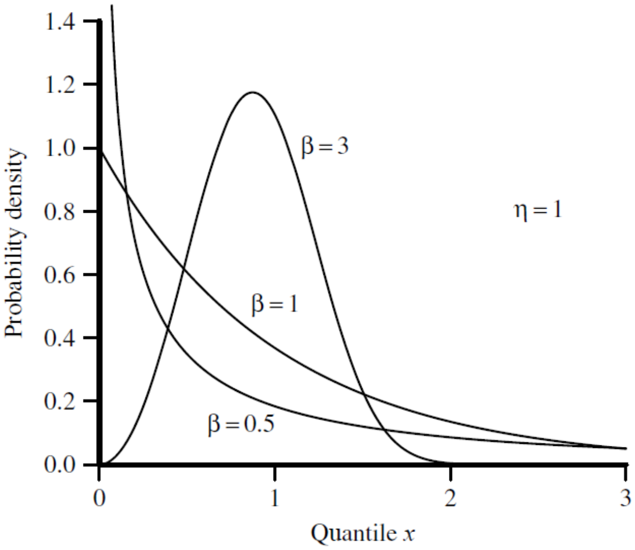

Figure 6: Probability density function for Weibull variate $W: \eta, \beta$

3.1.6 Tukey Distribution

(Stephanie, 2015) (Joiner & Rosenblatt, 1971) Tukey lambda distribution denoted by $Tukey(\lambda)$ is a continuous symmetric probability distributed defined in terms of its quantile function, named after the American mathematician John Wilder Tukey (1915-2000). Unlike most other probability distributions, there isn't a "one size fits all" formula for probability density function. It is defined in terms of quantiles where the quantile function $Q(p)$ (i.e. the inverse of the cumulative distribution function) and the quantile density function ($q = dQ/dp$) are:

$$Q(p;\lambda) = \begin{cases} \frac{1}{\lambda}\left[p^\lambda - (1-p)^\lambda\right], & \lambda \neq 0 \\ \log\left(\frac{p}{1-p}\right), & \lambda = 0 \end{cases}$$

$$q(p;\lambda) = p^{\lambda-1} + (1-p)^{\lambda-1}$$

Figure 7 shows the Tukey lambda distribution on different levels parameters.

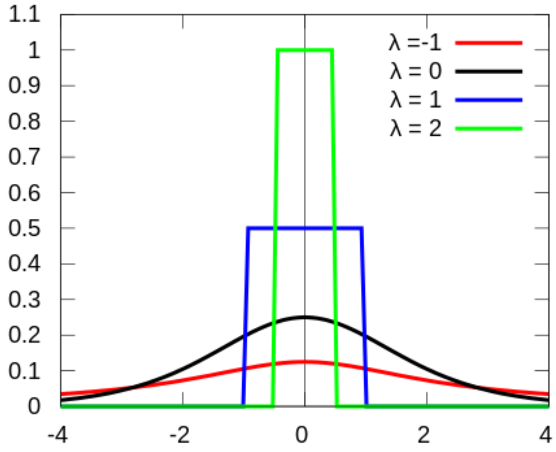

Figure 7: PDF of Tukey lambda distribution (Tukey lambda distribution, 2019)

3.1.7 Laplace Distribution

(Walck, 2007) (Forbes, Evans, Hasting, & Peacock, 2011) Laplace distribution sometimes called double exponential distribution is a continuous probability distribution named after Pierre-Simon Laplace (1749-1827). It is a symmetric distribution whose tails fall off less sharply than the Gaussian distribution but faster than the Cauchy distribution. The distribution has an interesting feature as the best estimator for the mean μ is the median and not the sample mean. The distribution is given by

$$f(x;a,b) = \frac{1}{2b}\begin{cases} \exp\left(-\frac{a-x}{b}\right), & x < \mu \\ \exp\left(-\frac{x-a}{b}\right), & x \geq \mu \end{cases}$$

Where a is the location parameter, and b > 0 is the scale parameter. The variance of the distribution is $2b^2$. **Figure 8** shows the Laplace distribution on different levels of a and b.

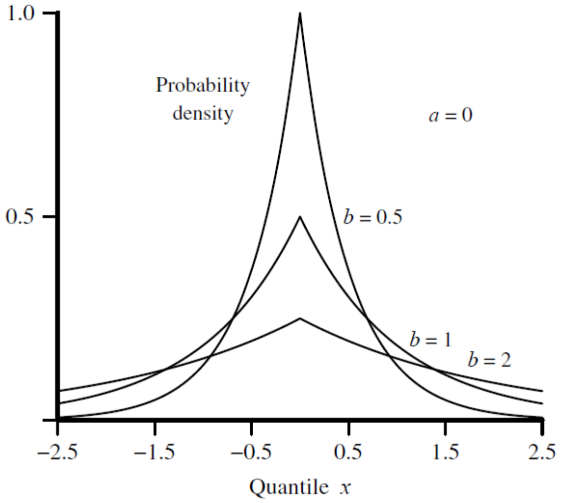

Figure 8: Probability density function for the Laplace variate $\boldsymbol{L}: a, b$

3.1.8 Uniform (Rectangular) Distribution

(Walck, 2007) (Forbes, Evans, Hasting, & Peacock, 2011) Uniform distribution denoted by $U(a, b)$ is a symmetric probability distribution defined by two parameters a and b where a the location parameter is and $(b - a)$ is the scale parameter. It is widely used as the basis for the generation of random numbers for other statistical distributions. Where every value in the range of the distribution is equally likely to occur. This is the distribution taken by uniform random numbers. It is given by

$$f(x; a, b) = \frac{1}{b - a};\ a \leq x \leq b$$

The mean and the variance of the Uniform distribution are given by:

$$E(X) = \frac{1}{2}(a + b)$$

$$Var(X) = \frac{1}{12}(b - a)^2$$

Figure 9 shows the p.d.f of the Uniform distribution.

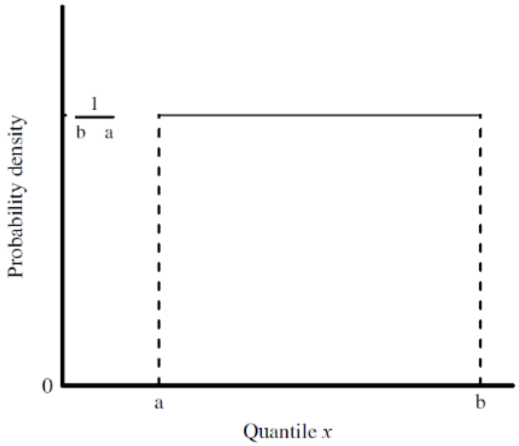

Figure 9: Probability density function for the rectangular variate ***R***: a, b

3.1.9 Truncated Normal Distribution

(Burkardt, 2014)The truncated normal probability density function is defined in two steps. We choose a general normal PDF by specifying parameters μ and σ^2, and then a truncation range (a, b). The p.d.f associated with the general normal distribution is modified by setting values outside the range to zero, and uniformly scaling the values inside the range so that the total integral is 1. Suppose X has a normal distribution with mean μ and variance $\sigma^2$and lies within the interval (a, b), with $-\infty \leq a \leq b \leq \infty$. Then the p.d.f of X truncated on $a < X < b$ is given by:

$$f(x;\ \mu, \sigma, a, b) = \frac{1}{\sigma} \frac{\emptyset\left(\frac{x-\mu}{\sigma}\right)}{\emptyset\left(\frac{b-\mu}{\sigma}\right) - \emptyset\left(\frac{a-\mu}{\sigma}\right)}; a \leq x \leq$$

Figure 10 shows the p.d.f of the Truncated normal distribution on different levels of a and b.

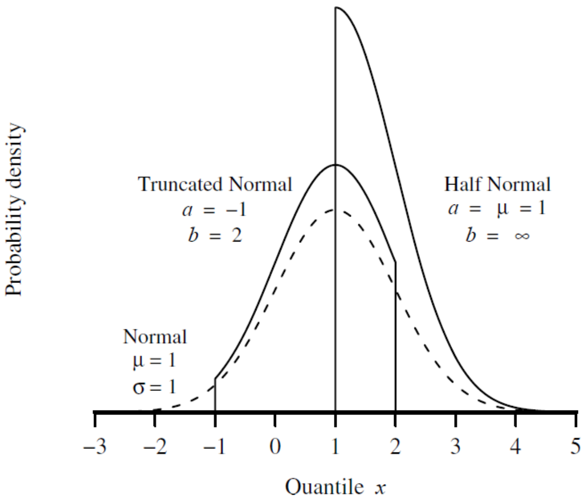

Figure 10: Probability density function for the truncated normal variate $\boldsymbol{X}$: μ, σ, a, b

3.2 Model construction

In this section, we describe the process of building the classification model of testing the normality. We start by describing the normality test as a classification problem. Then we describe the data that we will use for this purpose and the steps of training and evaluating the model.

3.2.1 Process

This problem is a binary classification problem, we predict if the sample data has departure from normality based on some properties such as skewness and kurtosis. The target variable in this classification problem is the type of distribution where “alternative” represents the positive class and “normal” represents the negative class. We did not choose the positive class to represent the normality – which could make more sense for others– because we need to compare the power of this model with other tests that try to check if the sample is significantly departing from normality and not vice versa.

In the process of creating the model, we are following the steps of Train-Validate-Test. In the training phase, we train a model using training data from positive and negative classes. Where in the validate stage we run the model on the validation data set and tune the model parameters to yield the best quality that can be achieved by the model. In this stage, we set in the model the threshold (cut off) points that have optimal quality. The tuned model is then tested on different test sets and the quality of this test represents the final quality of the model.

3.2.2 Classification Techniques

In this research, we aim to find the best classification technique that has the best performance in our use case, so we tend to build models using several methods and choose the one with the best quality. There are plenty of classification algorithms available to use. For example, the caret[3] (Classification And REgression Training) package in R has more than 180 classification techniques from different families. It is hard to examine all of the techniques to find the best one that fits our data, and it is not straight forward to select from this long list. So, in this research, I refer to previous studies that compared these techniques and evaluated their quality on several data sets.

(Fernandez-Delgado, Cernadas, Barro, & Amorim, 2014) Compared 179 classifiers from 17 families in 121 data sets, and (Wainer, 2016) compared 14 techniques on 115 binary datasets. The two studies show that Random Forest (RF) (Breiman, 2001), Gradient Boosting Machines (GBM) (Friedman, Greedy Function Approximation: A Gradient Boosting Machine, 2001), and Support Vector Machines (SVM) (Boser, Guyon, & Vapnik, 1992) classifiers are the most performing ones and they are not significantly different from each other. As a result, we will build

[3] http://topepo.github.io/caret/index.html

three models from these classifier types and compare their quality as part of this research. In the subsections below, we are providing a brief explanation of these classifiers.

3.2.2.1 Random forest (RF)

Random forest is an effective model for both classification and regression problems. In classification learning, it is an ensemble classifier constructed from a collection of decision trees that improve the prediction over a single decision tree. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Random Forests are supervised machine learning algorithms. As opposed to other machine learning models like neural networks, Random Forests make it easy to see the features that contribute to regression or classification and the importance of the variable to the decision (Breiman, 2001).

The data set is split randomly with replacement into different bags – this is called data bagging- each one represents a decision tree. For each bag, a different set of features with size $\sqrt{n}$ or $\frac{n}{3}$ is selected from feature set n and cross-validation are used to select which feature set is most appropriate for this specific bag – this is called Feature Bagging. So each data bag will have a different set of features chosen for creating the Decision Tree (Breiman, 2001).

In Random Forests, cross-validation is estimated internally during the run because of the bagging procedure. For each bag, based on sampling, about 62% unique samples from the original dataset are used. Hence, about a third of the samples are not present in ith tree construction. These left-out data will be used for cross-validation, to identify the most useful feature set combination, among the multiple randomly chosen feature bags for that data bag. The proportion of samples classified incorrectly from the cross-validation set (for that data bag) overall classifications of the cross-validation set is the error estimate of the system. This error estimate is also referred to as the out-of-bag (OOB) error estimate. OOB is a mean prediction error on each training sample j, using only the trees that did not have sample j in the bootstrap sample (Rebala, Ravi, & Churiwala, 2019).

For a new data point, the prediction of its class is the aggregation of the predicted classes from all trees. The final result is aggregated using the general voting technique as shown in the below equation:

$$f = \frac{1}{B}\sum_{n=1}^{B} f_n(x)$$

Where:

f = the final prediction from RF

B = number of trees

n = the index of decision tree

f_n = the result from decision tree n

x = the vector of the new data point to predict

Random Forest enables us to see which features are important for the variable to the decision. The intuitive notion in determining the variable importance is that if the variable is important, then rearranging the values of the variable in constructing the trees will not reduce the prediction accuracy. For a variable m, compute the number of correct classifications of the tree for out-of-bag cases. Permute values of variable m and then compute the classification of the out-of-bag case for the tree. Compute the difference in the number of correct classifications after permutation and before permutation. The average difference over all the trees is the importance score of the variable m (Ayyadevara, 2018).

3.2.2.2 Gradient boosting machines (GBM)

Gradient boosting is a machine learning classification technique based on creating an ensemble model from different models built sequentially as follows. It starts by creating an initial model using a tree or linear regression that fits the data. The second model is built and its objective is to accurately predict the cases where the first model performs poorly. The combination of these two models should have higher performance than either model alone. This booting process repeated many times until reaching the minimum prediction error. Gradient refers to the error, or residual, obtained after building a model. Boosting refers to improving. The technique is known as gradient boosting machine, or GBM. Gradient boosting is a way to gradually improve (reduce) error (Friedman, Greedy Function Approximation: A Gradient Boosting Machine, 2001)

To see how GBM works, let's begin with an easy example. Assume you're given a model M (which is based on decision tree) to improve upon. Let's say the current model accuracy is 80%. We want to improve on that. We'll express our model as follows:

$$Y = M(x) + error$$

Y is the dependent variable and $M(x)$ is the decision tree using the x independent variables. Now we'll predict the error from the previous decision tree:

$$error = G(x) + error2$$

$G(x)$ is another decision tree that tries to predict the error using the x independent variables. In the next step, similar to the previous step, we build a model that tries to predict $error2$ using the x independent variables:

$$error2 = H(x) + error3$$

Now we combine all these together:

$$Y = M(x) + G(x) + H(x) + error3$$

The preceding equation is likely to have an accuracy that is greater than 80% as individually model M (single decision tree) had 80% accuracy, while in the above equation we are considering 3 decision trees (Friedman, Hastie, & Tibshirani, Additive logistic regression: a statistical view of boosting, 2000)

3.2.2.3 Support vector machines with Radial Basis Function Kernel (RBF SVM)

Support vector machines (SVM) is a binary classifier, it classifies the data points by creating the optimal hyperplane boundary that has the maximum margin for the data points. SVM can handle linear separable data points as shown in the previous figure and can handle data points that are not linear separable by mapping data points into higher dimensional space using “kernel” functions. SVM classifier creates a hyperplane of N-1 dimensions for n-dimensional feature vectors to separate the data into two classes. For example, for feature vector of size 2 the hyperplane is a line and can be represented by the following equation (Boser, Guyon, & Vapnik, 1992):

$$y = w.f(x) + b$$

Where:

$f(x)$ = the feature vector

w = the weight assigned to feature vector

b = the bias term

All values of y greater than the function value are classified as class 1, and all other values are classified as class 2.

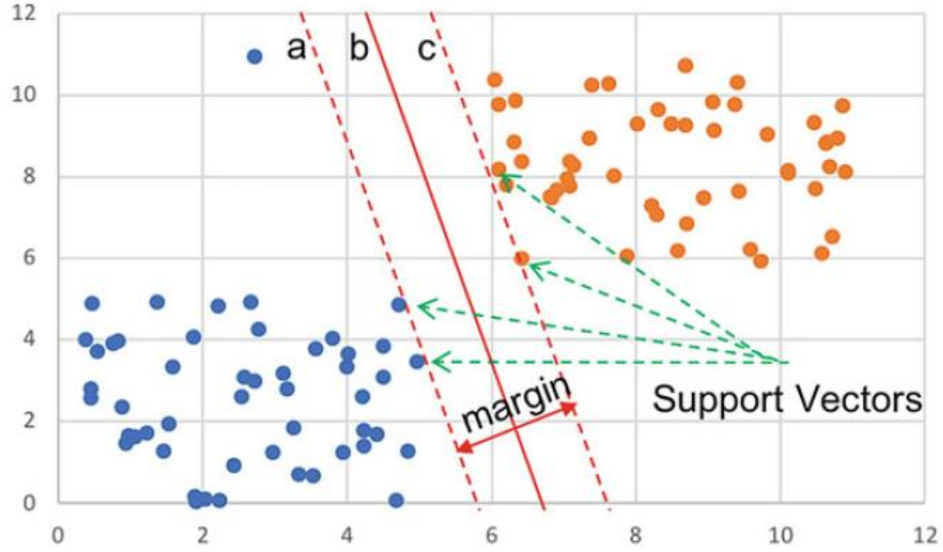

Figure 11: Support vectors and margin representation

Figure 11 shows valid class boundaries of a, b, and c represented by red lines. All of these boundaries classify the data points into two classes correctly. However, line b provides the largest margin for both the classes. SVM looks for boundaries that maximize the margin for the data points using sophisticated quadratic programming algorithms. The points closest to the lines a and c represent the support vectors that provide the boundary lines for the classes (Rebala, Ravi, & Churiwala, 2019).

For practical problems, there is higher noise in the data, or the data points may not be linearly separable. For complex nonlinear boundaries, it can be shown that by converting a nonlinear lower-dimensional space into a higher-dimensional space, the feature space can become linearly separable. For example, **Figure 12**Error! Reference source not found. shows data points that cannot be classified with linear classifiers in two-dimensional space. For these data points, using quadratic terms in a function to represent the data in an alternate dimension allows a linear classifier to draw a linear boundary.

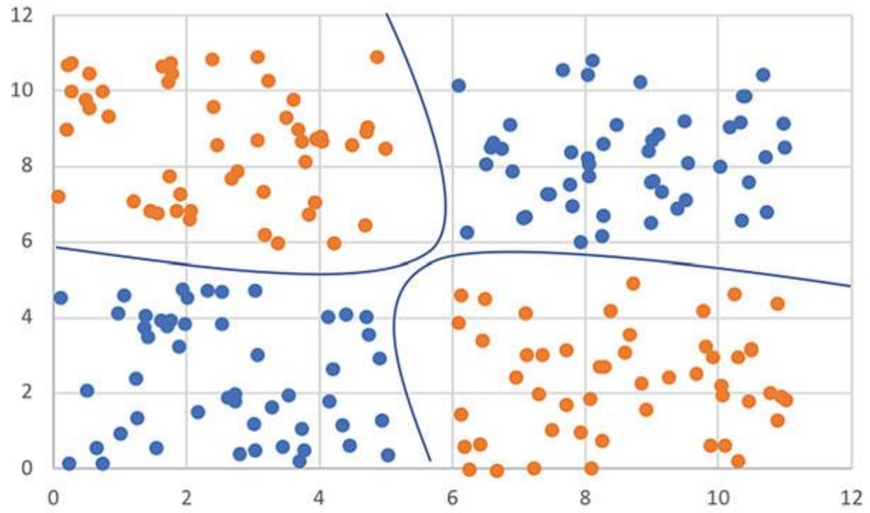

Figure 12: Nonlinear decision boundary

Converting data points to higher dimensional space requires mapping them to a function ϕ(x), where the function ϕ(x) has a higher number of variables or higher-order variables that can represent the feature variables in the input feature vector. This requires computing distance or a similarity measure between each pair of data points using the dot product. This can be very computationally intensive since you have to compute $O(N^2)$ dot products for N data points. Moreover, finding the right mapping function can be tricky. To solve this problem, a trick based on Mercer's theorem is used. A simplified explanation of Mercer's theorem states that a positive definite kernel function can be decomposed to a dot product. Conversely, instead of computing a mapping function, one can simply use a kernel function to represent the dot product of the mapping function. Hence, simply by choosing a kernel function, computing similarity between data points for a high dimensional feature space is very efficient. This allows for computing distance measures without actually computing ϕ(x) (Boser, Guyon, & Vapnik, 1992).

Kernels are similarity functions or distance functions that support certain dot product properties. Kernels allow substitution of a single function for a higher dimensional feature vector.

This kernel function would fit the data model for the input sample space as if it is based on the high dimensional feature vector. This makes computing distance or similarity very easy and also allows constructing the model with small input spac. Many kernels exist such as Polynomial and sigmoid functions. It is not obvious which kernel works best. Radial bias kernel (RBF) is one of the commonly used kernels in SVM and can be represented by the below equation (Bishop, 2006):

$$K(x, y) = e^{-(x-y)^2/2\sigma^2}$$

3.2.3 Data set

The first stage of building a classification model is to prepare a data set for training and testing. We used the simulation code to generate samples from alternative and normal distributions, where each generated sample represents a data point in our data set. Several statistics and features were calculated on each sample, where the features and the sample underlying distribution represent the data point/vector. The samples to be scaled before calculating the features to improve the scalability of the model and avoid biasedness toward a specific set of sample sizes or distributions.

Data set of 10,000 data points to be generated from both alternative and normal distributions having a 1:1 ratio between positives and negatives aiming to balanced data sets. The positive labels generated from the alternatives are listed in **Table 1**. The negative labels to be generated from the normal distribution on different levels of mean and standard deviation. Both sets of labels will be generated from 200 different sample sizes randomly selected from the range of [5, 2000].

We will divide the data set into “seen” and “unseen” sets. The “unseen” data represents samples from specific distributions that will not be used in the process of building the model. This set will be kept as hold out data to measure the quality of the model on data that it didn't see before which can give us an indicator on the generalizability of the model.

3.2.4 Training

Different models will be trained using the three techniques: Random Forest (RF), Gradient Descent Boosting (GBM), and Support vector machines with Radial Basis Function Kernel (RBF SVM). Each model will be evaluated and the best performing one will be considered and used in the later stage of comparing against statistical normality tests. The features of the model are calculated from each sample and saved in CSV format. The features to be used in the model are properties of the sample data such as sample size, median, skewness, kurtosis, Sigma_1_ratio (Percentage of data lies within 1 standard deviation), sigma_2_ration, sigma_3_ratio. This is an initial set of features we can start with to build a baseline. Other features probably will be added during the time of building the model.

Feature selection techniques could be applied to the model to find the most significant features and drop the non-important ones. Techniques such as Feature Importance of Random Forest, Recursive Feature Elimination (RFE), and ANOVA F-test could be used in this study. The goal is to keep the model with a minimal set of features that gives the highest possible quality.

3.2.5 Evaluation

Several metrics are available to use for evaluating the quality of a classification model. We prefer to use the Accuracy measure in this problem more than other measures like F-Measure. The Accuracy represents the combination of Specificity (1- α) and Sensitivity (Power) which are the measures we will use in comparing the quality of the model with other statistical tests.

The validation data set will be used to evaluate the quality of the models from several classification techniques. The models will be tuned by applying different model parameters such as the number of trees in the Random Forest classifier and sigma in the SVM classifier. The model with the best performance to be chosen for the next steps.

The selected model will be evaluated on the test set and the unseen data sets. Different quality measures and charts to be used to report and analyze the performance of the proposed test.

3.3 Power comparison test

A power comparison test to be concluded between different normality tests including the new proposed model using Monte Carlo simulation. The alternative distributions considered are the ones listed in **Table 1**. The comparison will be on three levels of significance $\alpha = 0.01$, $\alpha = 0.05$, and $\alpha=0.10$ to investigate the effect of the significance level on the power of the test. Corresponding thresholds of the proposed test on each level of significance can be calculated by choosing the thresholds that give specificity of 0.99, 0.95, and 0.90 for 0.01, 0.05, and 0.10 level of significance respectively. Samples of size n = 10, 20, 30, 50, 100, 200, 500, and 1000 will be used in the simulation from each alternative with 1,000 repetitions.

3.4 Toolbox

We will use R as the main programming language in this research. It offers data scientists and statisticians a vast toolbox and libraries for data loading, modeling, visualization, and analysis. RStudio with R 3.6.2 is used. We use the caret[4] package to build and evaluate the classification

[4] http://topepo.github.io/caret/index.html

models as it provides the data scientists with a simple interface for executing many classifiers with automatic parameter tuning for the models . This enables the researcher to use the state of the art classification techniques with minimal knowledge of the underlying algorithms (Kuhn, 2008). We will use the caret package to train and tune the models, feature selection, and variable importance estimation. MonteCarlo[5] library will be used to simulate the power of the model and the statistical normality tests.

[5] https://cran.r-project.org/web/packages/MonteCarlo/vignettes/MonteCarlo-Vignette.html

Chapter Four

Simulation and Results

4.1 Classification model

In this section, we show the process of building the normality classification model. We start by describing the data and the set of features used in training. Then we describe the training process and show the quality of the generated models. The last part of this section analyzes the errors generated from the models.

4.1.1 Data generation

Data set of size 10,000 data points to be used in training and evaluation was simulated using R code from both normal and alternative distributions. Each data point represents a sample of size n generated from alternative (positive) or normal (negative) distribution. Of the data, 50% of the data points are simulated from the positive class (“class_1”) and the other 50% are simulated from the negative class (“class_2”). The data intended to have a 1:1 ratio between positives and negatives aiming to balanced data sets.

The positive labels were generated from the alternatives are listed in **Table 1**. The negative labels were generated from the normal distribution on different levels of mean and standard deviation. Both sets of labels are generated from 200 different sample sizes selected from the range [5, 2000] listed in **Table 18** in the appendix 3. Total of 50 samples sampled from each size, 25 created from the alternative distributions, and another 25 samples created from the normal distribution. The negative 25 labels on each size generated as following: Five means were randomly selected from the range [-1000, 1000]. For each mean, five samples generated from a normal distribution with a coefficient of variation equals to 0.01, 0.1, 0.3, 0.6, and 1.0. Using different levels of variation aims to train the model on representative data set to decrease the biasedness to specific distributions. **Code snippet 1** in the appendix 1 shows the code used to generate the data.

4.1.2 Exploring data

Features are calculated on each data point and during training. Many features were examined and the following set is the ones that selected to build the model. Function **calc_stats()** in **Code snippet 1** shows how these features are calculated in R

- **Size**: The size of the sample. The smallest sample has size 8, and the largest sample has size 1998. See **Table 18** in the appendix 3.
- **Median**: The midpoint of the values that divide the set into two groups after they have been ordered from the smallest to the largest, or the largest to the smallest (Mulholland & Jones, 1968). The median for the normal distribution should be equal to the mean (Patel & Read, 1996). And because we scaled the samples, the median should be 0 for normal samples. The bigger the departure of the median from 0, the more likely the sample has departed from normality.
- **Skewness**: It is the measure of the symmetry of a probability distribution. A data set is symmetric if it looks the same to the left and the right of the center point. The skewness for a sample of size n is calculated using the formula:

$$\boldsymbol{skewness} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^3 / n}{s^3}$$

Where $\bar{x}$ = mean, s = standard deviation.

The skewness for the normal distribution is zero. Negative values for skewness indicate the data is skewed to the left, and a positive value indicates a skewness to the right (Hazewinkel, 1994).

- **Kurtosis**: It is a measure of whether the data is heavy-tailed or light-tailed relative to the normal distribution. Distributions with large kurtosis exhibit tail data exceeding the tails of the normal distribution. The formula for calculating the kurtosis is:

$$\boldsymbol{kurtosis = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^4 \,/n}{s^4}}$$

Where $\bar{x}$ = mean, s = standard deviation.

The kurtosis for the normal distribution is 3, it is less or greater than 3 for other distributions (Hazewinkel, 1994).

- **Sigma_1_ratio**: The percentage of the data that is located within 1 standard deviation. Normal distribution should have 68% of the data falls within 1 standard deviation (Patel & Read, 1996).
- **Sigma_2_ratio**: The percentage of the data that is located within 2 standard deviations. Normal distribution should have 95% of the data falls within 2 standard deviations (Patel & Read, 1996).
- **Sigma_3_ratio**: The percentage of the data that is located within 2 standard deviations. Normal distribution should have 99% of the data falls within 3 standard deviations (Patel & Read, 1996).

The target variable is **“dist_type”**, it has two possible values:

- **“class_1”**: The positive class; the class of the alternative distribution
- **“class_0”**: The negative class; the class of the normal distribution

As we see from the explanations above, the features are expected to be highly correlated with the target variable **"dist_type"**. **Table 2** to **Table 6** shows the descriptive statistic for the features for each class of the target variable. **Figure 13** to **Figure 19** shows the density plot for each feature by **dist_type**. **Figure 30** to **Figure 36** in the Appendix 2 show also the box plots for these variables. By looking at the statistics we can observe:

- The **size** has the same statistic for class_0 and class_1 as expected and it is uniformly distributed according to the density plot.

Table 2: Features descriptive statistics for size per dist_type

Feature	"class_0" (Normal)	"class_1" (Alternative)
size (sample size)		
minimum	8	8
median (IQR)	1,007.00 (459.25, 1,462.50)	1,007.00 (459.25, 1,462.50)
mean (sd)	993.25 ± 593.97	993.25 ± 593.97
maximum	1,998	1,998

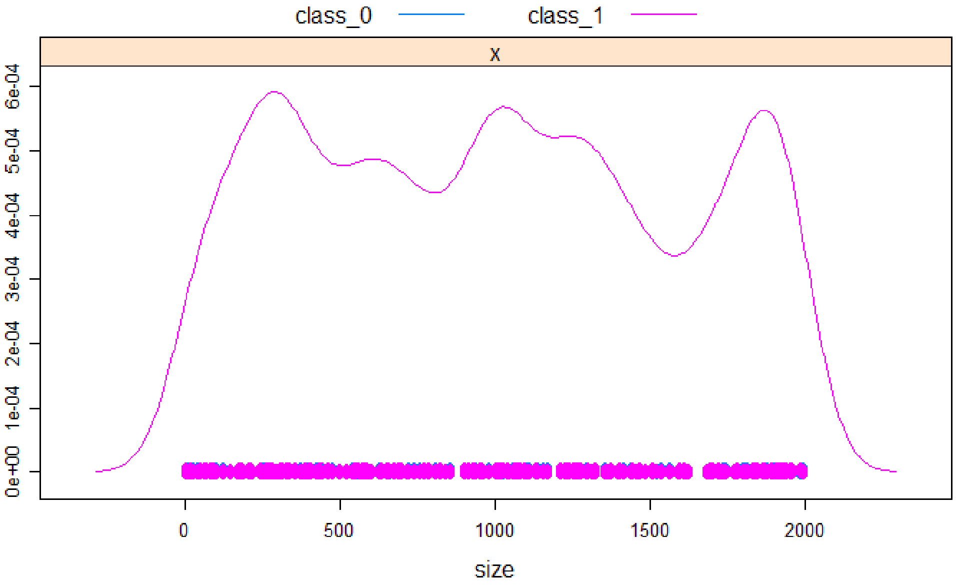

Figure 13: Density plot for "size"

- The **median** feature has a similar statistic of minimum, median, mean, and maximum for both classes. But the density plot shows that the median for class_0 is denser around zero where it is more flatten on the range of the distribution for class_1.

Table 3: Features descriptive statistics for median per dist_type

Feature	“class_0” (Normal)	“class_1” (Alternative)
median		
minimum	-0.59	-0.51
median (IQR)	0.00 (-0.02, 0.02)	-0.02 (-0.11, 0.02)
mean (sd)	-0.00 ± 0.05	-0.05 ± 0.12
maximum	0.45	0.35

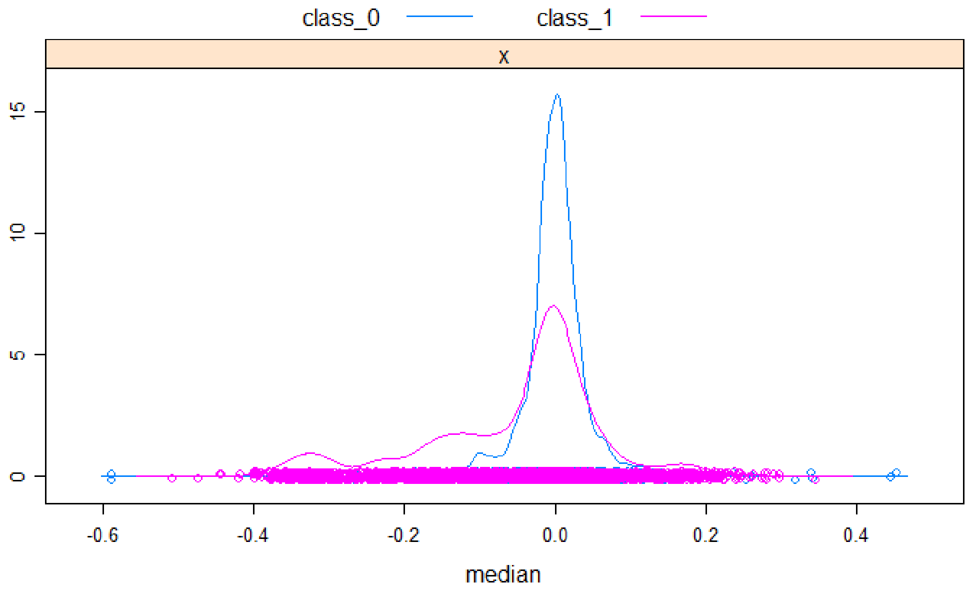

Figure 14: Density plot for "median"

- **Skewness** has a different range of values in the classes. Skewness ranges from [-1.3, 1.25] for class_0 while its range for class_1 is much bigger [-30.22, 29.26].

Table 4: Features descriptive statistics for skewness per dist_type

Feature	"class_0" (Normal)	"class_1" (Alternative)
skewness		
minimum	-1.30	-30.22
median (IQR)	-0.02 (-0.07, 0.05)	0.05 (-0.07, 0.78)
mean (sd)	-0.01 ± 0.12	0.60 ± 3.95
maximum	1.25	29.26

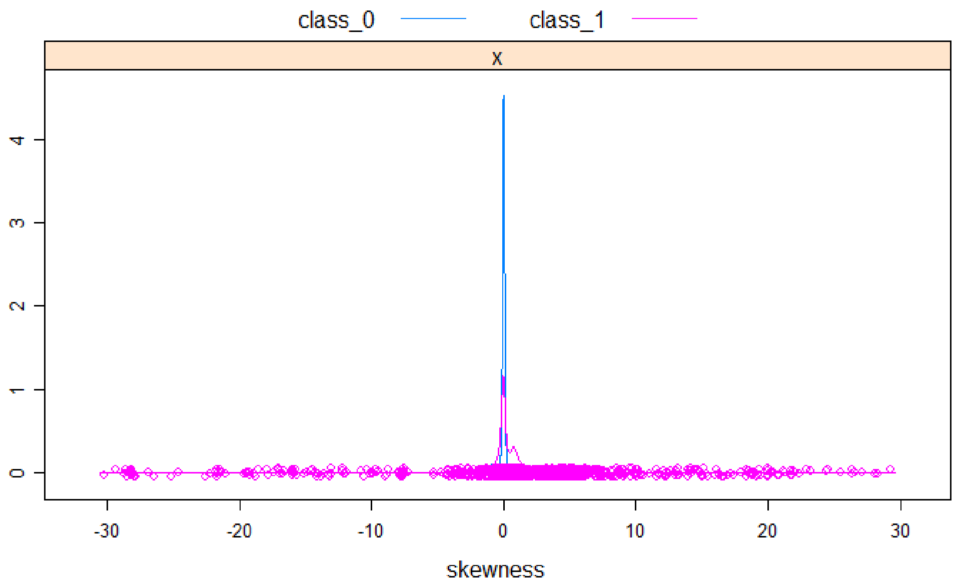

Figure 15: Density plot for "skewness"

- **Kurtosis** has very low values for class_0 comparing to class_1. The maximum value in class_1 is 4.83 while it spans from 1.36 to 986 for class_1.

Table 5: Features descriptive statistics for kurtosis per dist_type

Feature	"class_0" (Normal)	"class_1" (Alternative)
kurtosis		
minimum	1.43	1.36
median (IQR)	2.96 (2.87, 3.07)	3.55 (2.58, 5.62)
mean (sd)	2.96 ± 0.21	28.95 ± 104.02
maximum	4.83	985.85

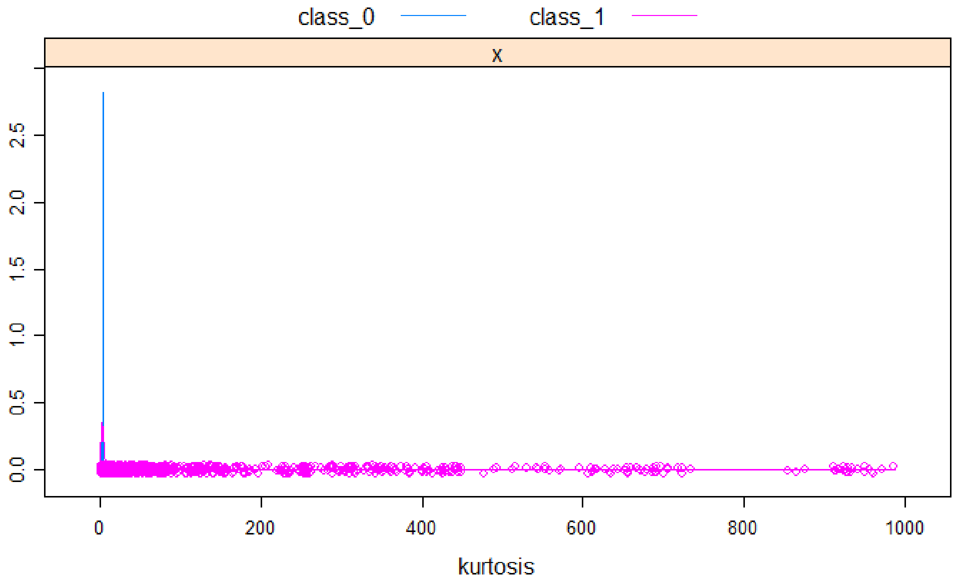

Figure 16: Density plot for "kurtosis"

- The density plots for **sigma_1_ratio**, **sigma_2_ratio,** and **sigma_3_ratio** show different distributes of these features between class_0 and class_1. The range of the values is almost similar but the values are significantly dense around the expected ratios - explained in the section above - in class_0 more than class_1.

Table 6 Features descriptive statistics for sigma_n_ratio per dist_type

Feature	**"class_0" (Normal)**	**"class_1" (Alternative)**
sigma_1_ratio		
minimum	0.50	0.38
median (IQR)	0.68 (0.68, 0.69)	0.70 (0.66, 0.76)
mean (sd)	0.68 ± 0.02	0.72 ± 0.11
maximum	0.88	1.00
sigma_2_ratio		
minimum	0.87	0.88
median (IQR)	0.95 (0.95, 0.96)	0.96 (0.95, 0.97)

Feature	“class_0” (Normal)	“class_1” (Alternative)
mean (sd)	0.95 ± 0.01	0.96 ± 0.02
maximum	1.00	1.00
sigma_3_ratio		
minimum	0.98	0.94
median (IQR)	1.00 (1.00, 1.00)	0.99 (0.99, 1.00)
mean (sd)	1.00 ± 0.00	0.99 ± 0.01
maximum	1.00	1.00

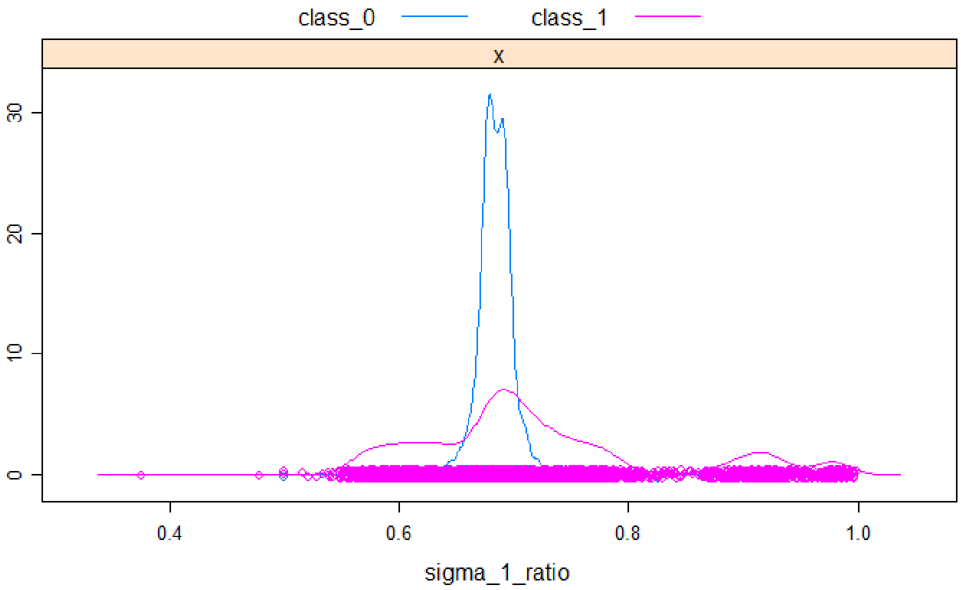

Figure 17: Density plot for "sigma_1_ratio"

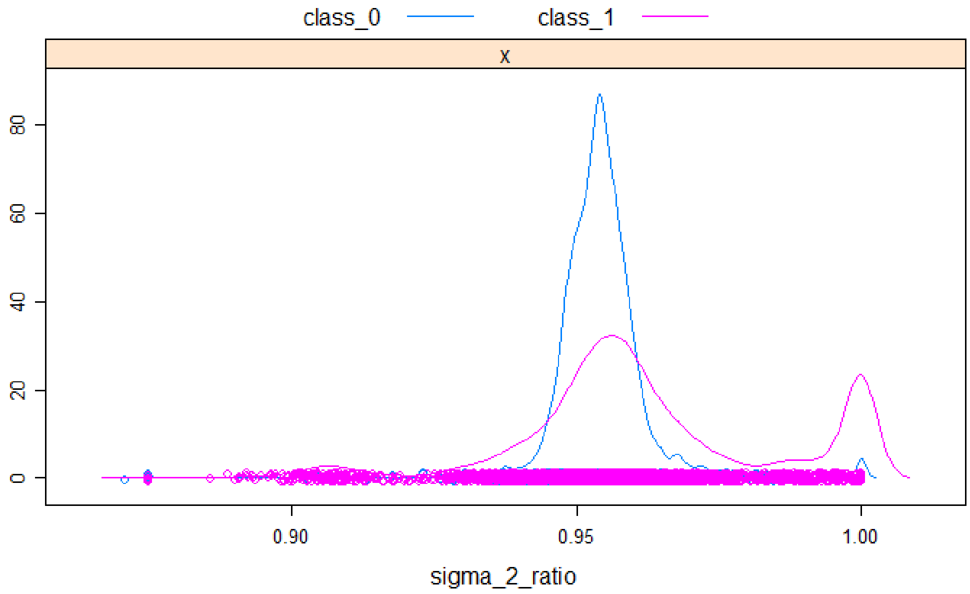

Figure 18: Density plot for "sigma_2_ratio"

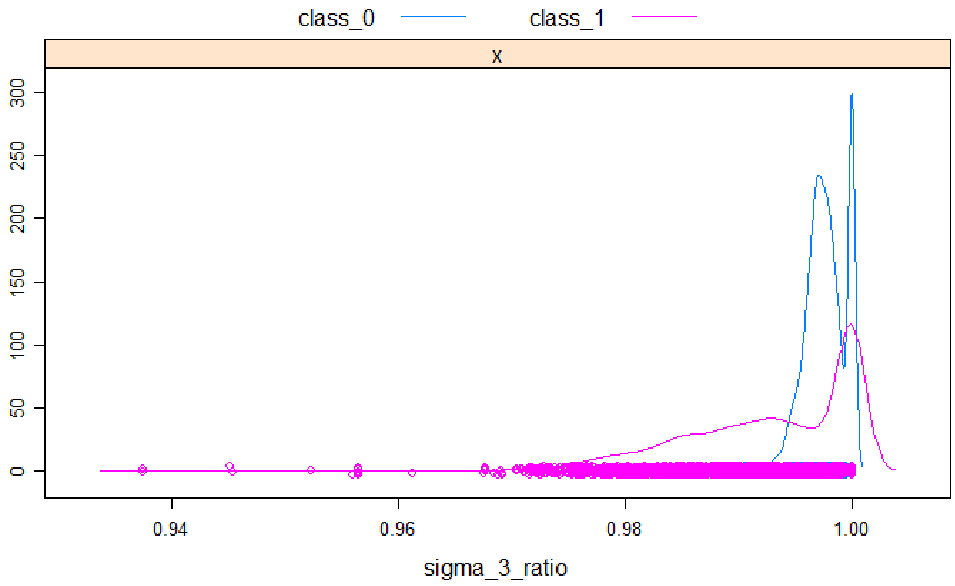

Figure 19: Density plot for "sigma_3_ratio"

These observations indicate that these features are very good candidates to be used in the model and predict the distribution type of a sample. In the next section, we will start the process of building the model.

4.1.3 Splitting data (train, validate, test, unseen)

"Unseen" data set were created by selecting one distribution for each of the five alternative families and all samples generated from 0.6 coefficient of variation normal family. These represent 20% of the data in which it split the data into so this process results in 8,000 "seen" and 2,000 "unseen" data sets. The reason of having the unseen data is to test the final model on distributions that the model didn't see before, to validate the scalability of the model and its ability to generalize to new data by predicting how our model will perform on other distributions not included in this research. The remaining data points of the other four types of both alternative and normal distributions represent the data set that will be used in training and testing, they are randomly divided into 60% train, 20% validate, 20% test.

Table 7 shows the distribution of the data sets after splitting.

Table 7: Data distribution after splitting

Seen **Size = 8000** **Train = 4800** **Validate = 1600** **Test = 1600**	Close_To_Normal	tukey(0.2), tukey(5), t(10), laplace(0, 10)
	Symmetric_Long_Tailed	t(2), t(4), t(7), tukey(10)
	Symmetric_Short_Tailed	beta(1.3, 1.3), beta(1.5, 1.5), tukey(1.5), truncatednormal(2, 2)
	Asymmetric_Long_Tailed	weibull(2, 1), lognormal(0, 1), chisq (4), chisq (10)
	Asymmetric_Short_Tailed	beta(3, 2), lognormal(0, 0.15), lognormal(0, 0.25), lognormal(0, 0.35)
	Normal c.o.v = 0.01	All samples
	Normal c.o.v = 0.1	All samples

	Normal c.o.v = 0.3	All samples
	Normal c.o.v = 1.0	All samples
Unseen **Size = 2000**	Close_To_Normal	tukey(0.1)
	Symmetric_Long_Tailed	t(1)
	Symmetric_Short_Tailed	uniform(0, 1)
	Asymmetric_Long_Tailed	weibull(0.5, 1)
	Asymmetric_Short_Tailed	beta(2, 1)
	Normal c.o.v = 0.6	All samples

4.1.4 Training

In this section, we describe the stage of training the classification models and tuning them using the validation set to find the appropriate parameters, mainly the optimal cutoff point. In later sections, we will evaluate the tuned models on different test sets.

4.1.4.1 Model generation

We tried to train three models using different classification techniques: Random Forest (RF), Gradient Boosting Machines (GBM), and Support Vector Machines (SVM). Experimentally, we trained other classifiers such as Naïve Bayes, K Neighbors, logistic regression, decision tree, neural networks (nnet). And we found they have lower quality than the former three so we excluded them from the study and focused only on the two boosting classifiers (RF and GBM) and the geometrical classifier (SVM). The goal of using multiple classifiers is - as we stated earlier in the methodology - is to find the best technique that can fit our data. “caret” package used to train the three models. The **x** vector consists of the seven features we described above: **size**, **median**, **skewness**, **kurtosis**, **sigma_1_ratio**, **sigma_2_ratio**, and **sigma_3_ratio**. The binary variable **dist_type** is the **y** target variable which has two possible values: class_1 (the positive class/alternative), class_0 (the negative class/normal).

To train a model using “caret“, we can pass different options to the training process through “train” and “trainControl” APIs to enable finding optimal parameters for the models. The main options we set in training are:

- “**Method**”: the resampling method. It specifies which technique “caret” will use for resampling the training data while it searches for the best tuning parameters. We choose “cv” with 10 number of folds as the resampling method.
- “**Metric**”: The summary metric to use in selecting the optimal model. Possible values are “Accuracy" and "Kappa", we choose “Accuracy”.
- Other options can be found at **Code Snippet 2** that shows the source code used to train the three modes.

Three models (rf, gbm, and svmRadial) were generated and the best parameters found by “caret” tuning for each of them can be found in **Table 8**.

Table 8: Model parameters

Model	Parameter	Value	Description
rf	mtry	2	Number of variable is randomly collected to be sampled at each split time.
	n.trees	500	Number of branches will grow after each time split

Model	Parameter	Value	Description
gbm	n.trees	400	Number of trees (the number of gradient boosting iteration)
	Interaction.depth	10	number of splits it has to perform on a tree (starting from a single node)
	shrinkage	0.1	It is considered as a learning rate
	n.minobsinnode	10	The minimum number of observations in trees' terminal nodes.
svmRadial	sigma	0.5257	The radial kernel smoothing parameter
	C	64	The penalty parameter of the error term. It controls the trade off between smooth decision boundary and classifying the training points correctly.

4.1.4.2 Model validation

Classification models by default apply 0.5 as a threshold, where prediction scores above this threshold are considered positive and predictions below this threshold are considered negative (Mueller & Guido, S, 2016). To find the optimal quality, the “validation” set is used to find the threshold that produces the highest accuracy of the models. **Table 9** and **Figure 20** show the best thresholds found in each model. We will use these thresholds to evaluate the quality of the models using the “**test**” and “**unseen**” data sets. We will refer to it as “**applied_threshold**” in the following text in this document.

Table 9: Best thresholds based on the validation set

Model	Best threshold	Accuracy
rf	0.6290000	0.940000
gbm	0.6952947	0.934375
svmRadial	0.3833224	0.919375

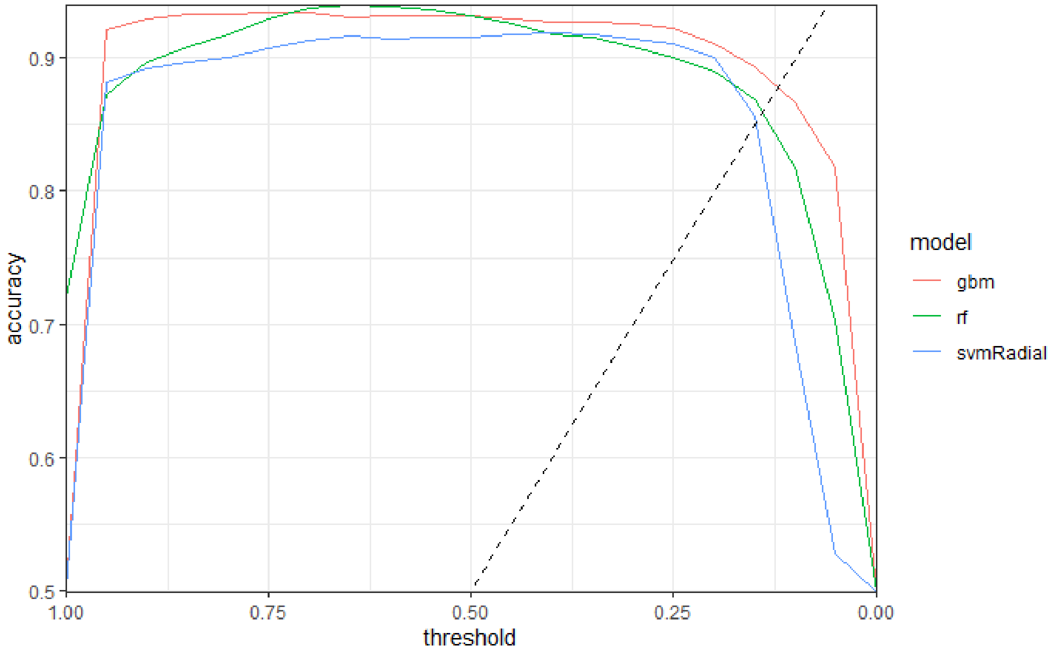

Figure 20: Accuracy on each threshold

The next section is to evaluate the quality of the models using different techniques, one of them is the "Accuracy" metric in which we will use the "applied_threshold" as a cut-off point to assign a positive and negative tag for each prediction.

4.1.5 Evaluation

4.1.5.1 Evaluation Metrics

ROC (Receiver Operating Characteristics) is a graph that shows the performance of a classification model at all classification thresholds. The ROC curve is created by plotting the true positive rate (TPR) against false positive rate (FPR). Below formulas show the calculation of these rates (Tharwat, 2020).

$$TPR = Recall = Sensitivity = Power \ = \frac{TP}{TP + FN}$$

$$FPR = 1 - Specificity = \frac{FP}{TN + FP}$$

$$Specificity = TNR = \frac{TN}{TN + FP}$$

Where:

- TP: The instance is labeled positive (alternative), and correctly classified by the model as positive (alternative)
- TN: The instance is labeled negative (normal), and correctly classified by the model as negative (normal)
- FP: The instance is labeled negative (normal), and incorrectly classified by the model as positive (alternative)
- FN: The instance is labeled positive (alternative), and incorrectly classified by the model as negative (normal)

ROC-AUC (Area under the ROC Curve) is a metric that represents a degree or measure of separability that tells how much the model can distinguish between classes. AUC ranges from 0 to 1. The closest the AUC toward 1, the better the performance of the model. A poor model has AUC near to 0. **Figure 21** and **Figure 22** show the ROC-AUC graphs of each model on the "test" and "unseen" sets. ROC-AUC is very high and close to 1 for the three models on the "test" set and a bit lower on the "unseen" set than the "test" set. **Table 10** indicates that the ROC-AUC on the "test" set is 0.978, 0.966, and 0.957 for **rf**, **gbm**, and **svmRadial** models respectively.

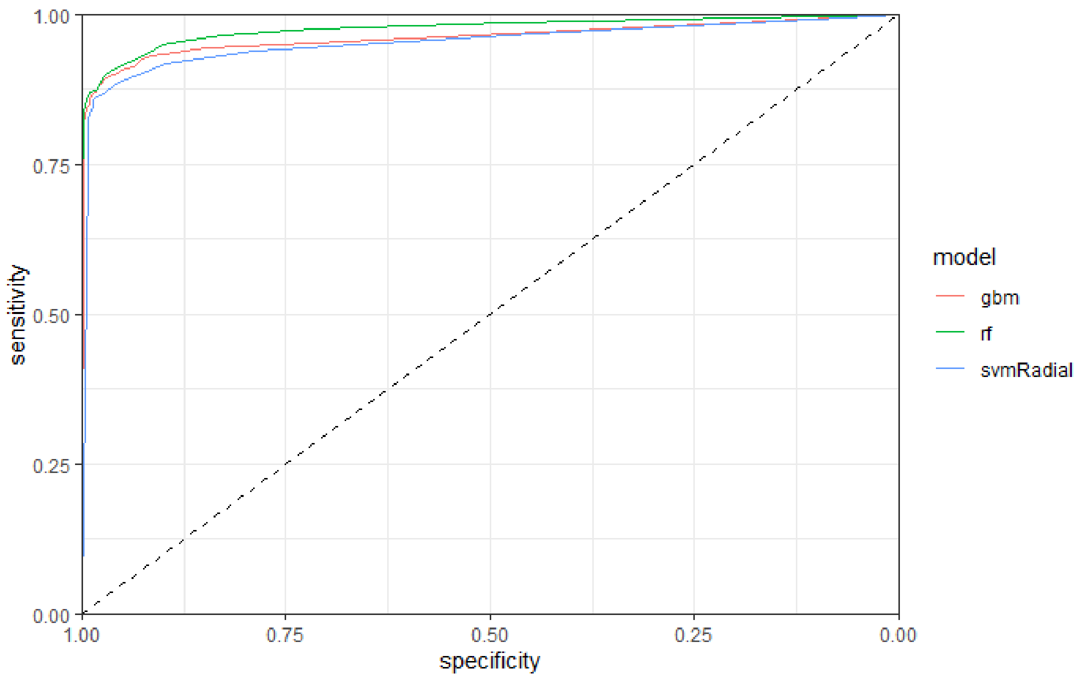

Figure 21: ROC on the Test set

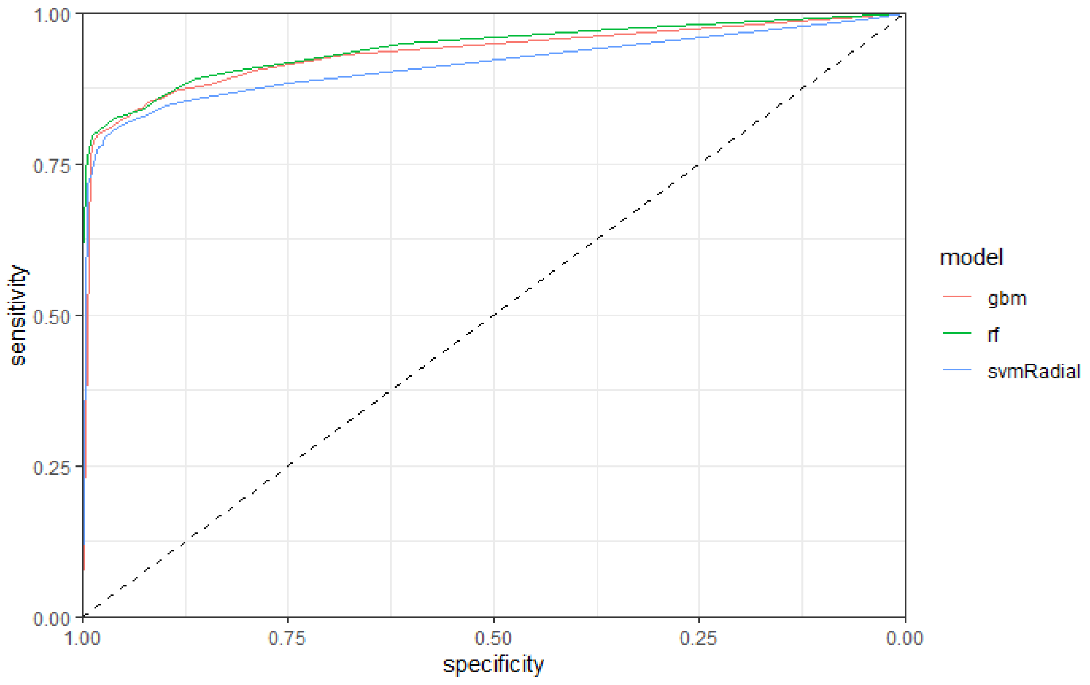

Figure 22: ROC on the Unseen set

Using the “applied_threshold” of each model found in the validation step, we calculated the accuracy of the models on the test set and also on the “unseen” set. The reason behind choosing the threshold on a data set (“validation”) and test the model on a different set (“test”) is to make sure the threshold chosen is not biased toward a specific set and keeps valid in other data. The “unseen” set is useful to examine how the model behaves on predicting new data points not seen in training nor validation. **Table 10** contains statistics of each model on “validation”, “test”, and “unseen” sets. We can notice that the three models have high “Accuracy” and “ROC-AUC”. And also we can observe from this table the following:

- The “Accuracy” is almost identical for both “validation” and “test” set. This means the “applied_threshold” does not over-fit the “validation” set.
- The “Accuracy” on “unseen” data is not significantly different from what is reported for the “validation” and “test” sets. This implies that the models are reasonably valid to make predictions on new data sets not included in the process.
- ROC-AUC is high and almost the same in “validation” and “test” sets. It is less by 2-3 points in the “unseen” set but it is a very minor degradation.
- Models are “specificity” oriented. The specificity is higher than the sensitivity by 12-20 points on different data sets. This implies that the model's ability to avoid predicting alternative data as normal is higher than its power/recall to correctly recognize an alternative data set. This is important as the impact of making a mistake of assigning an “alternative” class to a “normal” data set could be worse than missing an “alternative” instance. We aim to decrease the former error as much as possible as many statistical analysis tools assume normality of the underlying distribution of the data.

- Random Forest (“rf”) has a bit higher quality than the other two classifiers. We will nominate it as the final model we choose to represent our solution of predicating normality.

Table 10: Quality statistic of the models on applied_threshold

Model	Test set	Applied threshold	ROC_AUC	Sensitivity	Specificity	Accuracy
rf	validation	0.6290000	0.9736311	0.8872180	0.9925187	0.940000
rf	test	0.6290000	0.9777190	0.8694030	0.9899497	0.929375
rf	unseen	0.6290000	0.9437715	0.7960000	0.9880000	0.892000
gbm	validation	0.6952947	0.9672998	0.8922306	0.9763092	0.934375
gbm	test	0.6952947	0.9659413	0.8706468	0.9824121	0.926250
gbm	unseen	0.6952947	0.9401380	0.8090000	0.9650000	0.887000
svmRadial	validation	0.3833224	0.9552716	0.8872180	0.9513716	0.919375
svmRadial	test	0.3833224	0.9570052	0.8681592	0.9723618	0.920000
svmRadial	unseen	0.3833224	0.9165810	0.8100000	0.9580000	0.884000

(Quality evaluation code can be found in **Code Snippet 3** in Appendix 1)

To find the important features in the model, feature importance plots generated in **Figure 23**, **Figure 24**, and **Figure 25** for the three models that show the importance score on a scale of 0 to 100. Feature importance is useful to assess the features used in the model by identifying the predictive power of the features. A score assigned to each of them that indicates the relative importance of the feature when making a prediction.

We can observe that the most important feature in predicting the normality at the three models is “kurtosis”. “Skewness” is also at the top three features at all models which indicates its significance in prediction. The importance of other features varies from model to model. This is expected as each model applies different techniques in learning the problem.

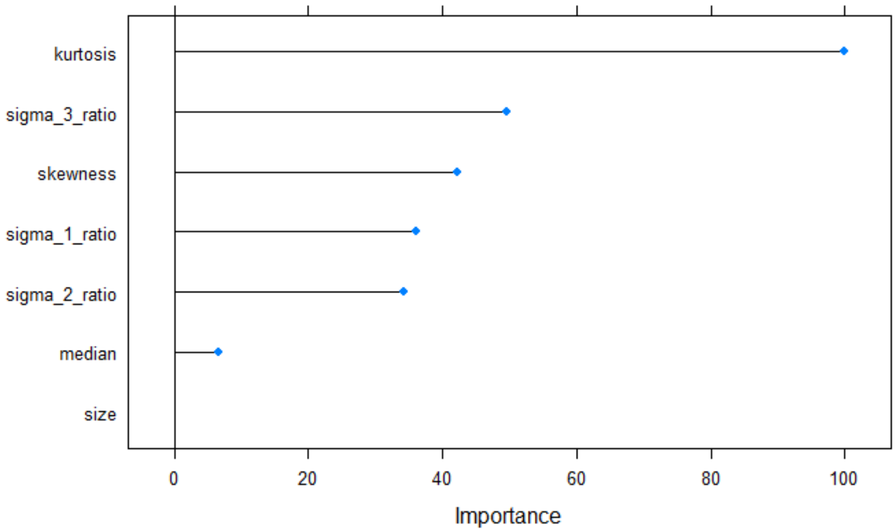

Figure 23: Feature importance in "rf" model

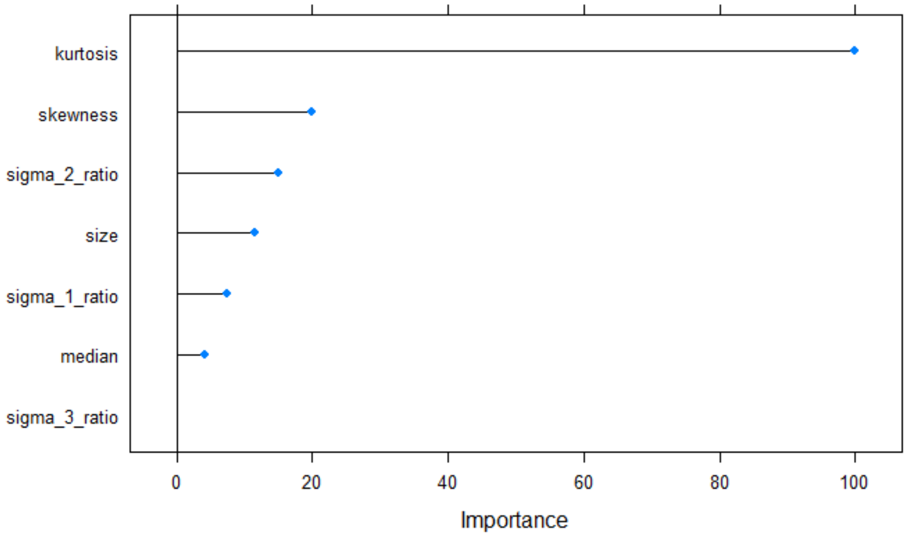

Figure 24: Feature importance in "gbm" model

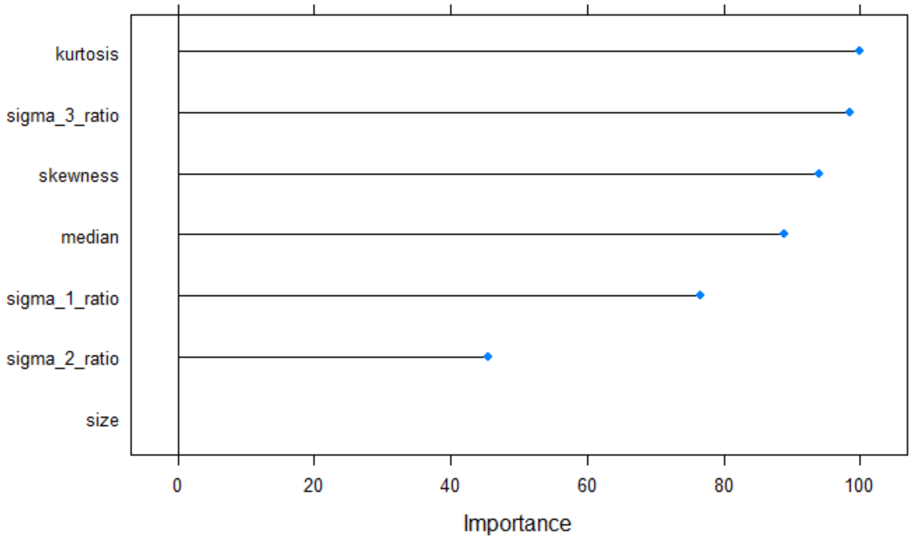

Figure 25: Feature importance in "svmRadial" model

4.1.5.2 Error analysis

Error analysis is an important stage in evaluating a classification model. We summarize the errors generated by the model trying to understand more the areas we can improve. We generated an instance report that contains the status of each instance of type FN or FP in the "validation". FP and FN are the specificity and sensitivity errors respectively. We are executing error analysis on the "validation" set so that any further improvement in the model can be tested on another test (the test set) to avoid biasedness. **Table 11** shows a summary of the instances of the three models. As we can notice, most of the errors are sensitivity errors where the models did classify incorrectly alternative distributions as a normal class. **Table 12** shows the sensitivity errors for the models at each alternative distributing family. We can observe that the majority of the errors happen on Symmetric_Short_Tailed and Close_To_Normal distributions. **Figure 26** shows the distribution of FN errors per sample size. The errors are distributed almost uniformly with a little skewness to the right that may indicate less power in small size but it is not clear enough to induce such a conclusion. **Table 19** in the appendix 3 shows 10 instances that got the lowest score by the RF model, these instances have features similar to what we can expect for normal distributions. These findings point to the areas that we can start investigating if more improvement is required for the quality of the model.

Table 11: Summary instance report

Classifier	TP	FP	FN	TN
Rf	710	8	88	794
Gbm	713	19	85	783
svmRadial	711	40	87	762

Table 12: Sensitivity errors (FN) per alternative family

Alternative family	"rf"	"gbm"	"svmRadial"
Asymmetric_Long_Tailed	2	2	1
Asymmetric_Short_Tailed	5	6	5
Symmetric_Long_Tailed	1	1	4
Symmetric_Short_Tailed	46	44	44
Close_To_Normal	34	32	33

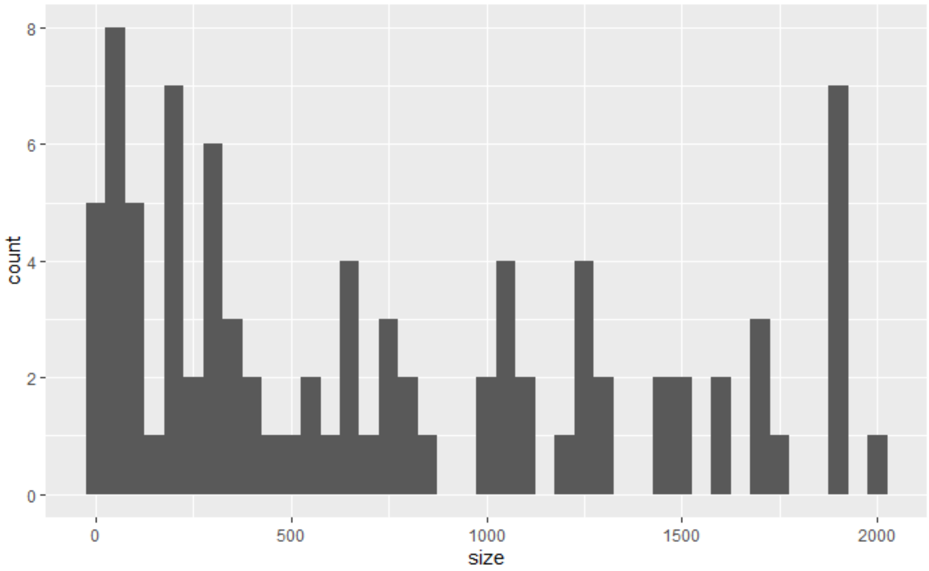

Figure 26: Frequency of FN errors per size

4.2 Power comparison

In previous sections, we showed how we generated a classification model to predict the "non-normality" of a sample data. In the evaluation step, we tested the models on different test sets and the results indicate a high quality (accuracy) of the models. In this section, we will evaluate the classification model in terms of the "normality test" and will examine its quality by comparing its power with other statistic tests. We choose in this comparison the "rf" model to represent the new machine learning approach of testing normality. It was the one with the highest performance as we saw in the evaluation steps before. For the next sections, we will call the created classification model (rf) as a "new_test" when we compare it with other tests.

4.2.1 Procedure

A power comparison test was concluded between the "new_test" model and other statistical tests using the Monte Carlo simulation procedure. Monte Carlo is a method to estimate the probability and the expected value of a random variable by repeating a random process many times. If we find that we are unable to compute a probability or an expected value exactly with mathematics, we can still attempt to estimate it by making the computer repeat the random experiment many times, keeping track of the result of the experiment each time. This technique is known as Monte Carlo simulation, after the famous Monte Carlo casino in the Principality of Monaco (Hasting, 1970). For example, to find the integral between 3 and 6 in a normal distribution of mean 1 and standard deviation 10, one could use the probability tables. But it can be simulated by sampling from that distribution 100,000 times and see how many values are between 3 and 6.

Related to our problem, we will run a Monte Carlo simulation to estimate the power of the normality tests. The power of the test is the probability that the test rejects the null hypothesis ($H_0: Sample\ is\ normal$) when it should be rejected (sample is actually "not-normal"). We will estimate the power of the tests participating in this comparison by letting each test to detect the departure from normality on a set of samples from "alternative" distributions. We repeat this process many times and the ratio of detected samples out of all examined samples represents the power of the test.

Statistical normality tests were chosen to be in this comparison based on their popularity. Seven tests included in this research are listed in **Table 13**. We used a wide range of alternative distributions in this comparison. A total of 25 different distributions were chosen from the main five families shown in **Table 1**. The power was estimated on three levels of significance: α = 0.01, α = 0.05, and α=0.10. Samples of size 10, 20, 30, 50, 100, 200, 500, and 1000 were used in the simulation from each alternative with 1,000 repetitions.

Table 13: List of normality tests used in the power comparison

Shapiro-Wilk (SW)	Anderson-Darling (AD)
Jarque-Bera (JB)	Shapiro-Francia (SF)
Kolmogorov-Smirnov (KS)	Cramer-von Mises (CVM)
Lilliefors (Lillie)	

The "new_test" is a binary classification model that was tuned in the "validation" stage with a fixed threshold of 0.629 in which a prediction above this threshold is considered as "alternative" a prediction below this threshold as "normal". The question is how we will run this model on three different levels of significance. We can answer this question simply if we know that:

$$\alpha = FPR = 1 - Specificity$$

So, to run the model on $\alpha = 0.05$ level of significance, we need to apply the threshold that gives $specificity = 1 - 0.05 = 0.95$. **Table 14** shows the thresholds used in the “new_test” – “rf” model - on each significance level using the “validation” set.

Table 14: "new_test" Threshold used on each significance level

α	Specificity	Threshold
0.01	0.99	0.65
0.05	0.95	0.45
0.1	0.9	0.35

MonteCarlo R package[6] used to run the simulation, complete code can be found in **Code Snippet 4** in the Appendix 1.

4.2.2 Results

In this section we discuss the results of the power of normality tests including the “new_test” we propose in this research. We will show the power of the tests from different perspectives. First, we calculate the overall power of each test on all alternative distribution per each sample size. Then, we calculate the power of each test per each alternative family. And finally, we show the power of each test on each of the 25 alternative distributions.

By looking at **Figure 27**, **Figure 28**, and **Figure 29** that show the overall power of the tests on 0.01, 0.05, and 0.1 significance levels, we can see that the “new_test” is the most powerful compared to other statistical tests on all level of significance. It has high power on a small sample

[6] https://cran.r-project.org/web/packages/MonteCarlo/vignettes/MonteCarlo-Vignette.html

size especially on 0.05 and 0.1 significance levels. **Table 20**, **Table 21**, and **Table 22** in the Appendix 2 show the power in tabular format.

Table 15, **Table 16**, and **Table 17** show the power of the tests per alternative family on the three levels of significance. The three tables indicate that the "new_test" significantly has higher power than other tests in every family. It is different than other tests where it gives high power on small sample sizes. On the other hand, it is consistent with other tests that they are most powerful in detecting Asymmetric-Long-Tailed alternatives and they give the lowers power on Close-To-Normal distributions. But the "new_test" has relatively high power on all families comparing with other tests. Power results on every distribution can be found in **Table 23**, **Table 24**, and **Table 25** in Appendix 3.

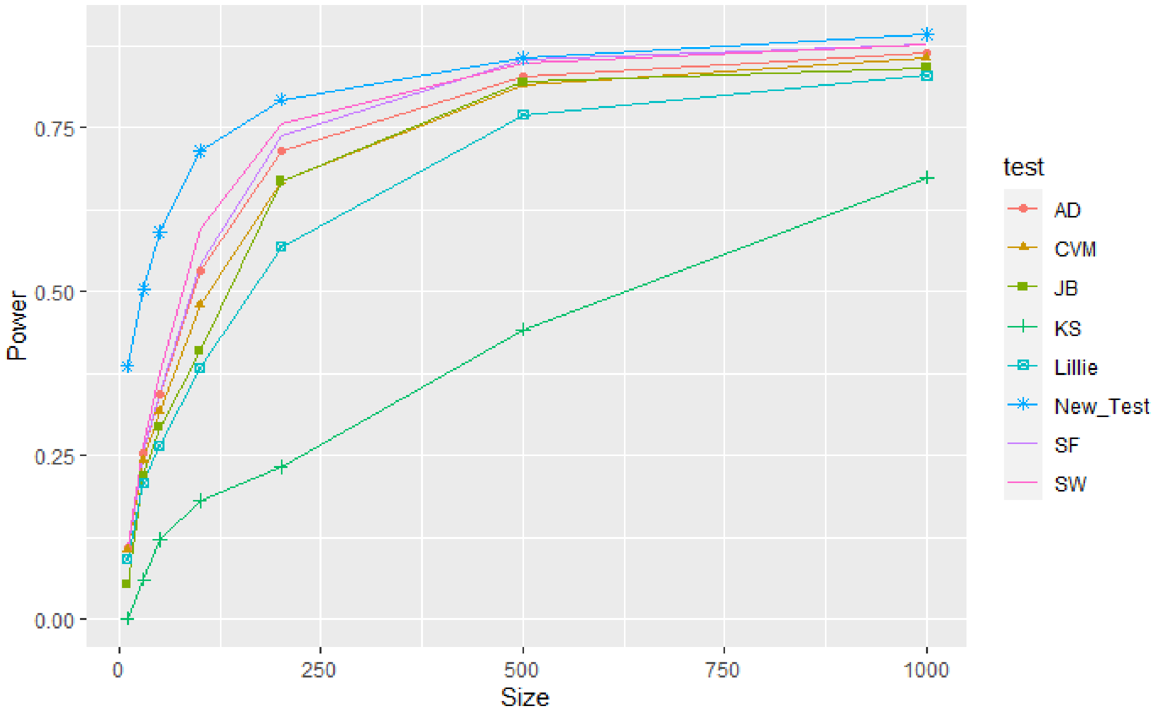

Figure 27: Overall power comparison on 1% significance level

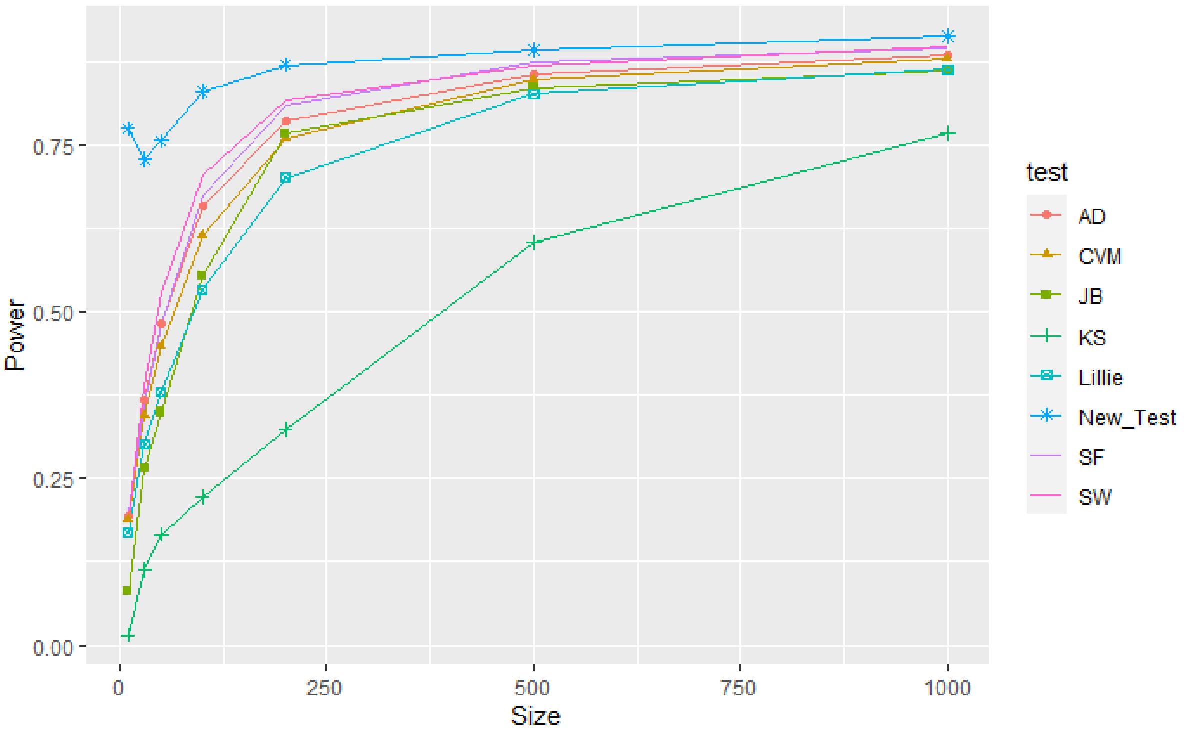

Figure 28: Overall power comparison on 5% significance level

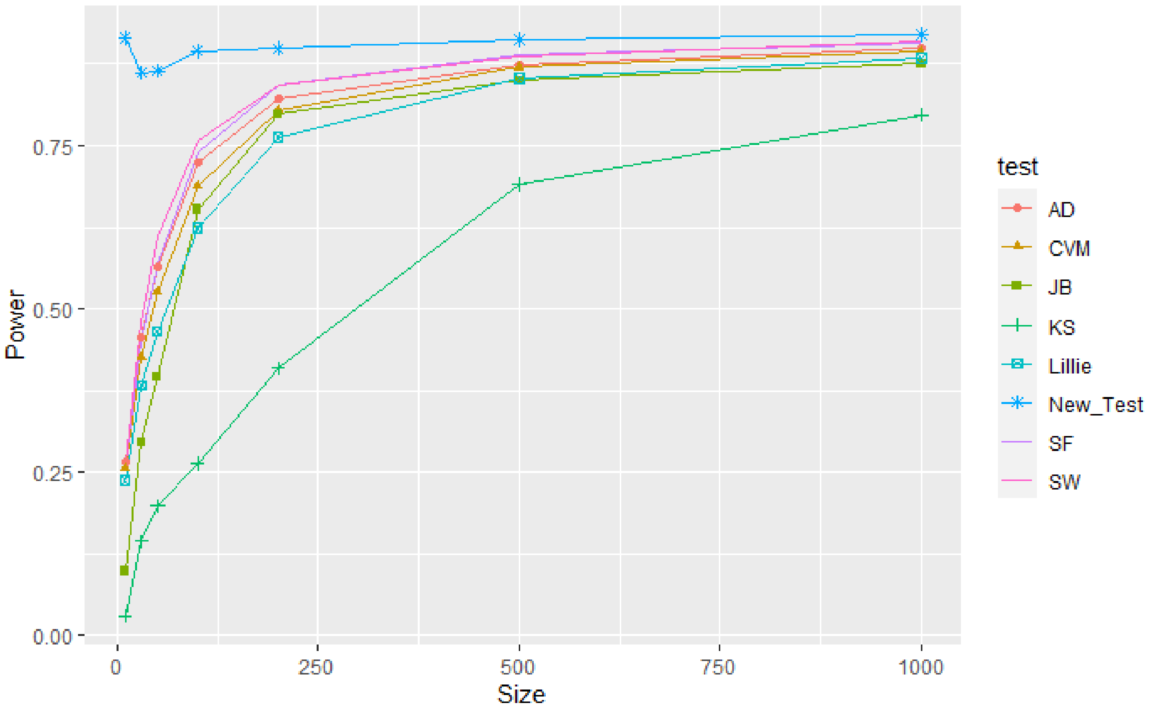

Figure 29: Overall power comparison on 10% significance level

Table 15: Tests power per alternative family on 1% level of significance

Family	Size	“new_test”	SW	AD	CVM	JB	KS	Lillie	SF
CTN	10	**0.29**	0.02	0.02	0.03	0.01	0	0.02	0.03
	30	**0.26**	0.06	0.05	0.06	0.08	0	0.05	0.07
	50	**0.27**	0.1	0.1	0.1	0.11	0	0.07	0.1
	100	**0.33**	0.19	0.2	0.21	0.19	0	0.15	0.19
	200	**0.41**	0.34	0.36	0.34	0.26	0.02	0.28	0.31
	500	**0.54**	0.51	0.47	0.45	0.35	0.18	0.42	0.52
	1000	**0.66**	0.59	0.53	0.51	0.41	0.38	0.45	0.59
ALT	10	**0.5**	0.27	0.25	0.23	0.11	0	0.19	0.25
	30	**0.69**	0.55	0.52	0.49	0.47	0.15	0.43	0.53
	50	**0.82**	0.67	0.62	0.59	0.6	0.3	0.53	0.65
	100	**0.95**	0.87	0.78	0.74	0.77	0.4	0.67	0.83
	200	**0.99**	**0.99**	0.94	0.91	0.93	0.51	0.83	0.98
	500	**1**	**1**	**1**	**1**	**1**	0.73	0.98	**1**
	1000	**1**	**1**	**1**	**1**	**1**	0.94	**1**	**1**
AST	10	**0.33**	0.03	0.03	0.03	0.01	0	0.02	0.03
	30	**0.36**	0.13	0.1	0.09	0.09	0	0.06	0.11
	50	**0.44**	0.28	0.22	0.17	0.15	0	0.11	0.24
	100	**0.64**	0.56	0.47	0.4	0.31	0	0.28	0.49
	200	**0.8**	**0.8**	0.71	0.64	0.66	0.04	0.54	0.76
	500	0.94	0.97	0.94	0.92	0.97	0.4	0.83	**0.98**
	1000	0.99	**1**	**1**	0.99	**1**	0.64	0.96	**1**
SLT	10	**0.45**	0.21	0.24	0.24	0.13	0.01	0.22	0.23
	30	**0.68**	0.52	0.53	0.53	0.46	0.15	0.48	0.55
	50	**0.79**	0.64	0.61	0.61	0.6	0.31	0.57	0.67
	100	**0.89**	0.76	0.72	0.7	0.78	0.5	0.66	0.78
	200	**0.95**	0.86	0.81	0.79	0.89	0.59	0.74	0.87
	500	**0.99**	0.96	0.92	0.91	0.98	0.67	0.85	0.97
	1000	**1**	0.99	0.99	0.98	**1**	0.78	0.93	**1**
SST	10	**0.38**	0.01	0.01	0.01	0	0	0.01	0.01
	30	**0.53**	0.06	0.05	0.04	0	0	0.02	0.02
	50	**0.64**	0.2	0.16	0.11	0	0	0.05	0.07
	100	**0.76**	0.61	0.49	0.35	0.01	0	0.16	0.43
	200	**0.81**	0.8	0.75	0.66	0.6	0	0.46	0.77

Family	Size	“new_test”	SW	AD	CVM	JB	KS	Lillie	SF
	500	**0.81**	0.8	0.8	0.8	0.8	0.24	0.77	0.8
	1000	**0.81**	0.8	0.8	0.8	0.8	0.64	0.8	0.8

Table 16: Tests power per alternative family on 5% level of significance

Family	Size	“new_test”	SW	AD	CVM	JB	KS	Lillie	SF
CTN	10	**0.69**	0.08	0.09	0.08	0.02	0	0.08	0.1
	30	**0.5**	0.13	0.14	0.15	0.11	0	0.12	0.15
	50	**0.47**	0.18	0.19	0.2	0.15	0	0.17	0.2
	100	**0.54**	0.31	0.33	0.32	0.24	0.02	0.27	0.3
	200	**0.59**	0.47	0.45	0.43	0.31	0.08	0.39	0.45
	500	**0.65**	0.57	0.54	0.52	0.39	0.31	0.49	0.59
	1000	**0.75**	0.67	0.62	0.6	0.49	0.4	0.55	0.66
ALT	10	**0.87**	0.38	0.37	0.36	0.17	0.02	0.3	0.39
	30	**0.89**	0.67	0.63	0.6	0.55	0.27	0.55	0.64
	50	**0.93**	0.79	0.74	0.7	0.68	0.38	0.64	0.77
	100	**0.99**	0.94	0.88	0.85	0.85	0.49	0.79	0.91
	200	**1**	**1**	0.99	0.96	0.99	0.63	0.92	**1**
	500	**1**	**1**	**1**	**1**	**1**	0.87	**1**	**1**
	1000	**1**	**1**	**1**	**1**	**1**	0.99	**1**	**1**
AST	10	**0.76**	0.1	0.09	0.09	0.02	0	0.08	0.1
	30	**0.68**	0.29	0.25	0.21	0.13	0	0.17	0.26
	50	**0.72**	0.48	0.41	0.36	0.23	0.01	0.27	0.42
	100	**0.84**	0.7	0.64	0.57	0.52	0.04	0.48	0.66
	200	**0.93**	0.91	0.83	0.79	0.83	0.21	0.71	0.88
	500	**0.99**	**0.99**	0.97	0.96	**0.99**	0.57	0.92	**0.99**
	1000	**1**	**1**	**1**	**1**	**1**	0.83	0.99	**1**
SLT	10	**0.75**	0.33	0.35	0.35	0.18	0.05	0.33	0.36
	30	**0.8**	0.62	0.61	0.61	0.53	0.29	0.57	0.64
	50	**0.86**	0.71	0.69	0.68	0.67	0.44	0.64	0.73
	100	**0.93**	0.81	0.78	0.77	0.83	0.56	0.73	0.84
	200	**0.98**	0.9	0.87	0.85	0.91	0.64	0.81	0.92
	500	**1**	0.98	0.96	0.95	0.99	0.74	0.92	0.98
	1000	**1**	**1**	**1**	0.99	**1**	0.84	0.97	**1**
SST	10	**0.8**	0.06	0.06	0.06	0	0	0.05	0.05

Family	Size	“new_test”	SW	AD	CVM	JB	KS	Lillie	SF
	30	**0.77**	0.25	0.21	0.16	0.01	0	0.1	0.11
	50	**0.79**	0.49	0.38	0.3	0.01	0	0.18	0.3
	100	**0.85**	0.76	0.66	0.56	0.33	0	0.39	0.66
	200	**0.86**	0.81	0.8	0.77	0.8	0.07	0.67	0.81
	500	**0.83**	0.81	0.81	0.81	0.81	0.53	0.81	0.81
	1000	**0.81**	**0.81**	**0.81**	**0.81**	**0.81**	0.78	**0.81**	**0.81**

Table 17: Tests power per alternative family on 10% level of significance

Family	Size	“new_test”	SW	AD	CVM	JB	KS	Lillie	SF
CTN	10	**0.86**	0.13	0.15	0.14	0.04	0	0.14	0.16
	30	**0.71**	0.2	0.22	0.22	0.13	0.01	0.2	0.22
	50	**0.69**	0.26	0.28	0.27	0.19	0.02	0.25	0.28
	100	**0.68**	0.39	0.4	0.4	0.25	0.05	0.37	0.38
	200	**0.67**	0.53	0.51	0.5	0.34	0.15	0.47	0.53
	500	**0.72**	0.63	0.59	0.58	0.45	0.37	0.54	0.64
	1000	**0.78**	0.73	0.68	0.66	0.55	0.42	0.61	0.71
ALT	10	**0.97**	0.48	0.45	0.43	0.21	0.06	0.4	0.47
	30	**0.96**	0.73	0.7	0.67	0.59	0.33	0.62	0.72
	50	**0.97**	0.85	0.8	0.77	0.73	0.43	0.71	0.84
	100	**1**	0.97	0.93	0.9	0.91	0.54	0.85	0.95
	200	**1**	**1**	0.99	0.98	0.99	0.71	0.95	1
	500	**1**	**1**	**1**	**1**	**1**	0.94	**1**	**1**
	1000	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
AST	10	**0.92**	0.17	0.18	0.16	0.03	0	0.13	0.17
	30	**0.85**	0.4	0.37	0.32	0.18	0.01	0.27	0.37
	50	**0.86**	0.57	0.51	0.45	0.31	0.03	0.38	0.51
	100	**0.93**	0.8	0.72	0.67	0.64	0.11	0.6	0.75
	200	**0.96**	0.94	0.89	0.85	0.91	0.33	0.79	0.93
	500	0.99	**1**	0.98	0.98	0.99	0.68	0.96	0.99
	1000	**1**	**1**	**1**	**1**	**1**	0.9	**1**	**1**
SLT	10	**0.89**	0.39	0.43	0.41	0.21	0.08	0.41	0.44
	30	**0.9**	0.66	0.67	0.65	0.57	0.37	0.62	0.7
	50	**0.92**	0.76	0.73	0.72	0.72	0.5	0.69	0.79
	100	**0.96**	0.84	0.83	0.81	0.86	0.59	0.78	0.87
	200	**0.98**	0.92	0.9	0.89	0.94	0.67	0.86	0.94

Family	Size	“new_test”	SW	AD	CVM	JB	KS	Lillie	SF
	500	**1**	0.99	0.98	0.97	0.99	0.79	0.95	0.99
	1000	**1**	**1**	**1**	**1**	**1**	0.87	0.99	**1**
SST	10	**0.93**	0.15	0.13	0.13	0.01	0	0.11	0.1
	30	**0.89**	0.41	0.33	0.28	0.01	0	0.21	0.24
	50	**0.88**	0.62	0.5	0.42	0.03	0.01	0.3	0.44
	100	**0.9**	0.8	0.74	0.66	0.6	0.03	0.52	0.74
	200	**0.88**	0.82	0.81	0.8	0.81	0.19	0.74	0.82
	500	**0.84**	0.82	0.81	0.82	0.82	0.67	0.82	0.82
	1000	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	0.8	**0.82**	**0.82**

Chapter Five

Discussion and conclusion

5.1 Thesis summary

The results showed that using Machine Learning techniques is a valid solution for the problem of detecting departure from normality for a data sample. We managed to build a classification model using a minimal number of features that had a high ability to predict departure from normality on all families of alternative probability distributions with high resilience to the sample size. The “Accuracy” of the three classification models built in this research was high on all data sets, including the “unseen” data which is a set of probability distributions held out from the process to assess the ability of the model to generalize to new data sets not seen before. The performance of the classification model as a normality test (“new_test”) was validated by comparing its power against the state of the art statistical normality tests. The results showed that the “new_test” has significantly better power than the other tests on different levels of significance and sample sizes. Another advantage of this approach is that the classical normality tests are hypothesis tests, i.e. if a test failed to reject the null hypothesis (sample is normal) it does not mean we can accept it, it can only give predictions on one of the directions. However, the "new_test" is a classification model that can predict both cases, either normal or not.

5.2 Future research

The work done in this research could be a starting point for further development in the direction of using machine learning models to solve statistical problems we used to solve by classical statistical tests. Learning from data seems a very promising approach that could help in solving problems related to the characteristics of the data. Future research on this topic could be:

- Improve the quality -mainly sensitivity- of the models by doing thorough error analysis. We can examine the areas that had low accuracy and try to understand the characteristics of these areas aiming to find more features that could decrease the number of errors and improve the quality.
- Explore other ML techniques to solve this problem such as deep learning. We can also try an ennsemble classifier of several statistical tests.
- It is important to assess the applicability of converging a theory into real-life applications. The next step is to encapsulate the model in a new R library published to the public to let this functionality available for use by statisticians and data scientists as a new method for testing the normality.
- The model created in this research can be extended from binary (normal, alternative) to a multiclass classification model to classify the sample into its underlying probability distribution.
- Explore Machine Learning techniques in other parametric and non-parametric tests

5.3 Resources

A live demo of the new_test is available to use[7]. The demo demonstrates the ability of the classification model to predict the normality status of a sample and compare its results with the seven classical normality tests. The source code used for this research is available as a GitHub open source project[8].

[7] http://ec2-50-112-220-245.us-west-2.compute.amazonaws.com:3838/thesis-demo/

[8] https://github.com/hsoboh/hussein-soboh-ms-thesis

References

Afeez, B. (2018). Selection and Validation of Comparative Study of Normality Test. *American Journal of Mathematics and Statistics, 8*(6), 190-201.

Alizadeh, H., & Arghami, N. (2011). Monte Carlo comparison of seven normality tests. *Journal of Statistical Computation and Simulation, 81*(8), 965-972.

Ayyadevara, V. K. (2018). *Pro Machine Learning Algorithms.* Apress. doi:10.1007/978-1-4842-3564-5

Bishop, C. (2006). *Pattern Recognizition and Machine Learning.* Springer.

Boser, B., Guyon, I., & Vapnik, V. (1992). A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144-152. doi:10.1145/130385.130401

Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 5-32. doi:10.1023/A:1010933404324

Burkardt, J. (2014, October 17). The Truncated Normal Distribution. Retrieved May 9, 2020, from http://people.sc.fsu.edu/~jburkardt/presentations/truncated normal.pdf

Cai, X.-L., Xie, D.-J., Madsen, K., Wang, Y., Bogemann, S., Cheung, E., . . . Chan, R. (2020). Generalizability of machine learning for classification of schizophrenia based on resting-state functional MRI data. *Human Brain Mapping, 41*(1), 172-184. doi:https://doi.org/10.1002/hbm.24797

Casella, L., & Berger, R. (2001). *Statistical inference* (2nd ed.). Cengage Learning.

D'Agostino, B., & Stephens, A. (1986). *Goodness-of-fit Techniques.* Marcel Dekker.

Das, R., & Imon, M. (2016). A Brief Review of Tests for Normality. *American Journal of Theoretical and Applied Statistics, 5*(1), 5-12. doi:10.0.45.128/j.ajtas.20160501.12

Fernandez-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we Need Hundreds of Classifiers to Solve Real WorldClassification Problems? *Journal of Machine Learning Research*(15), 3133-3181.

Forbes, C., Evans, M., Hasting, N., & Peacock, B. (2011). *Statistical Distributions* (4th ed.). Hoboken, New Jersey: John Wiley & Sons, Inc.

Friedman, J. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics, 29*(5), 1189-1232.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics, 28*(2), 337-407. doi:10.1214/aos/1016218223

Ghasemi , A., & Zahediasl, S. (2012). Normality Tests for Statistical Analysis: A Guide for Non-Statisticians. *International Journal of Endocrinology and Metabolism, 10*(2), 486-489. doi:10.5812/ijem.3505

Hasting, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika, 57*(1), 97-109.

Hazewinkel, M. (1994). *Encyclopedia of Mathematics.* Springer.

Islam, T. (2011). Normality Testing- A New Direction. *International Journal of Business and Social Sciences, 2*, 115-118.

Islam, T. (2019). Ranking of Normality Tests: An Appraisal through Skewed Alternative Space. *Symmetry, 11*(827). doi:10.20944/preprints201905.0190.v1

Joiner, B., & Rosenblatt, J. (1971). Some Properties of the Range in Samples from Tukey's Symmetric Lambda Distributions. *Journal of the American Statistical Association, 66*(334), 394–399.

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software, 28*(5), 1-26.

Lukas, E. (1942). A characterization of the normal distribution. *Annals of Mathematical statistics, 13*, 91-83.

Lyon, A. (2014). Why are Normal Distributions Normal. *The British Journal for The Philosophy of Science, 65*, 621-649.

Mueller, C., & Guido, S, S. (2016). *Introduction to Machine Learning with Python.* O'Reilly Media.

Mulholland, H., & Jones, C. (1968). *Fundamentals of statistics.* Boston: Springer. doi:10.1007/978-1-4899-6507-3

Muyombya, M. (2017). On empirical power of univariate normality tests under symmetric, asymmetric and scaled distributions. *International Journal of Scientific and Engineering Research, 8*(3).

Normal distribution. (2020, April 29). Retrieved May 9, 2020, from Wikipedia: https://en.wikipedia.org/w/index.php?title=Normal_distribution&oldid=953810074

Oztuna, D., Elhan, A., & Tuccar, E. (2006). Investigation of four different normality tests in terms of type 1 error rate and power under different distributions. *Turkish Journal of Medical Sciences, 36*(3), 171-176.

Patel, K., & Read, B. (1996). *Handbook of the Normal Distribution* (2nd ed.). CRC Press.

Pearson, K. (1895). Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous. *Philosophical Transactions of the Royal Society of London. A, 186*, 343-414. Retrieved from https://www.jstor.org/stable/90649

Ramasubramanian, K., & Singh, A. (2019). *Machine Learning Using R.* Apress. doi:10.1007/978-1-4842-4215-5

Razali, N., & Wah, Y. (2011). Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling Tests. *Journal of Statistical Modeling and Analytics, 2*(1), 21-33.

Rebala, G., Ravi, A., & Churiwala, S. (2019). *An introduction to machine learning.* Springer. doi:10.1007/978-3-030-15729-6

Seier, E. (2002). Comparison of tests of univariate normality. *InterStat Statistical Journal, 1*, 1-17.

Shapiro, S., Wilk, B., & Chen, J. (1968). Acomparison study. *Journal of American, 63*(324), 1343-1372.

Stephanie. (2015, August 6). *Tukey Lambda Distribution: Definition*. Retrieved May 9, 2020, from Statistics How To: https://www.statisticshowto.com/tukey-lambda-distribution/

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*. doi:10.1016/j.aci.2018.08.003

Throde, C. (2002). *Testing For Normality.* Marcel Dekker.

Tukey lambda distribution. (2019, June 6). Retrieved May 9, 2020, from Wikipedia: https://en.wikipedia.org/w/index.php?title=Tukey_lambda_distribution&oldid=900491544

Wainer, J. (2016). Comparison of 14 different families of classification.

Walck, C. (2007, September 10). Hand-book on statistical distributions for expirements. University of Stockholms. Retrieved May 9, 2020, from http://www.stat.rice.edu/~dobelman/textfiles/DistributionsHandbook.pdf

Yap, W., & Sim, H. (2011). Comparison of various types of normality tests. *Journal of Statistical Computation and Simulation, 81*(12), 2141-2155. doi:10.1080/00949655.2010.520163

Appendix 1: Code

Code snippet 1: Data generation

```
library("extraDistr") #rtlambda
library("truncnorm") #rtruncnorm
library("moments")#skewness and kurtosis

#"Generate samples"
set.seed(666)
population <- unique(as.integer(runif(n = 1000000, min = 5, max = 2000)))
sizes <- sort(sample(x= population, size = 200, replace = FALSE))
generate_samples(sizes)

generate_samples<-function(sizes){

  process_sample<-function(s, family, dist, alternative){

    #Calc features
    scaled_s <- scale(s)

    size <- length(scaled_s)

    stats = calc_stats(s, scaled_s)

    #sample_id
    sample_id <- paste(family, dist, size, sep = "-")

    data_set[nrow(data_set) + 1, ] = c(sample_id, family, dist, stats,
alternative)

    write(x = scaled_s, file = paste(data_files_dir, "/", sample_id,
"_scaled", sep = ""), ncolumns = 1)
    write(x = s, file = paste(data_files_dir, "/", sample_id, sep = ""),
ncolumns = 1)

    return(data_set)
  }


  data_set <- data.frame(sample_id=character(0), dist_family=character(0),
dist=character(0), size=integer(0), mean=numeric(0), median=numeric(0),
mean_median_diff=numeric(0), sd=numeric(0), skewness=numeric(0),
kurtosis=numeric(0), outliers_minor_ratio=integer(0),
outliers_extream_ratio=integer(0), sigma_1_ratio=numeric(0),
sigma_2_ratio=numeric(0), sigma_3_ratio=numeric(0), alternative=integer(0),
stringsAsFactors = FALSE)


  for(size in sizes){
    print(paste("Processing samples of size ", size, sep = ""))

    ############ Close to normal
    family <- "close_normal"
    for(dist in dist_ctn){
      s <- create_alternative_sample(dist, size)
      data_set <- process_sample(s, family, dist, 1)
    }


    ############ Symmetric long-tailed
    family <- "sym_long_tail"

    for(dist in dist_slt){
      s <- create_alternative_sample(dist, size)
      data_set <- process_sample(s, family, dist, 1)
    }

    ############# Symmetric short-tailed
    family <- "sym_short_tail"
    for(dist in dist_sst){
      s <- create_alternative_sample(dist, size)
      data_set <- process_sample(s, family, dist, 1)
    }


    ############# Asymmetric long-tailed
    family <- "asym_long_tail"

    for(dist in dist_alt){
      s <- create_alternative_sample(dist, size)
      data_set <- process_sample(s, family, dist, 1)
    }


    ############# Asymmetric short-tailed
    family <- "asym_short_tail"

    for(dist in dist_ast){
      s <- create_alternative_sample(dist, size)
      data_set <- process_sample(s, family, dist, 1)
    }


    ############ Normal

    #To make sure we get in all runs the same means and sd
    #I don't why the set.seed out the loop does not work here
    set.seed(666)
    norm_mean = as.integer(runif(5, -1000, 1000))

    norm_cov <- c(0.01, 0.1, 0.3, 0.6, 1.0) #coeffienent of variation
    for(mu in norm_mean){
      for(cov in norm_cov){
        family <- paste("normal(", cov, ")", sep = "")

        sd = round(abs(cov*mu), 3)

        s <- rnorm(n = size, mean = mu, sd = sd)
    data_set <- process_sample(s, family, paste("Normal", mu, sd, sep =
"_"), 0)
   }
  }
 }


 write.csv(data_set, file=data_file)
}

do_scaling <- TRUE

calc_stats<-function(sample, scaled_sample = NULL){
 if(do_scaling == TRUE){
  if(is.null(scaled_sample)){
   s <- scale(sample)
  }else{
   s <- scaled_sample
  }
 }else{
  s <- sample
 }
 #Calc features
 size <- length(s)
 mean_ <- round(mean(s), 5)
 median_ <- round(median(s),5)
 sd_ <- sd(s)
 skewness_ <- round(skewness(s), 5)
 kurtosis_ <- round(kurtosis(s), 5)
 outliers <- find_outliers(s)

 sigma_1_ratio <- length(which(abs(s - mean_) <= 1*sd_ ))/size
 sigma_2_ratio <- length(which(abs(s - mean_) <= 2*sd_ ))/size
 sigma_3_ratio <- length(which(abs(s - mean_) <= 3*sd_ ))/size

 outliers_minor_ratio <- length(outliers$minor) / size
 outliers_extream_ratio <- length(outliers$extreme) / size

 stats <- list("size" = size,
        "mean" = mean_,
        "median" = median_,
        "mean_median_diff" = abs(mean_ - median_) / sd_,
        "sd" = sd_,
        "skewness" = skewness_,
        "kurtosis" = kurtosis_,
        "outliers_minor_ratio" = outliers_minor_ratio,
        "outliers_extream_ratio" = outliers_extream_ratio,
        "sigma_1_ratio" = sigma_1_ratio,
        "sigma_2_ratio" = sigma_2_ratio,
        "sigma_3_ratio" = sigma_3_ratio)
 return(stats)
}

find_outliers<-function(data){
 lowerq = quantile(data)[2]
 upperq = quantile(data)[4]
 iqr = upperq - lowerq #Or use IQR(data)
```

```
  minor_threshold_upper = (iqr * 1.5) + upperq
  minor_threshold_lower = lowerq - (iqr * 1.5)

  extreme_threshold_upper = (iqr * 3) + upperq
  extreme_threshold_lower = lowerq - (iqr * 3)

  outliers = list()
  outliers[["minor"]] = data[(data < minor_threshold_lower) | (data >
minor_threshold_upper )]
  outliers[["extreme"]] = data[(data < extreme_threshold_lower) | (data >
extreme_threshold_upper )]

  return(outliers)
}

create_alternative_sample<-function(dist, size){
  #Close to normal dist
  if(dist == "tukey(0.1)"){
    return(rtlambda(n = size, lambda = 0.1))
  }
  if(dist == "tukey(0.2)"){
    return(rtlambda(n = size, lambda = 0.2))
  }
  if(dist == "tukey(5)"){
    return(rtlambda(n = size, lambda = 5))
  }
  if(dist == "t(10)"){
    return(rt(n = size, df = 10))
  }
  if(dist == "laplace(0,10)"){
    return(rlaplace(n = size, mu = 0, sigma = 10))
  }


  #Sym Long Tail
  if(dist == "t(1)"){ #Cachy
    return(rt(n = size, df = 1))
  }
  if(dist == "t(2)"){
    return(rt(n = size, df = 2))
  }
  if(dist == "t(4)"){
    return(rt(n = size, df = 4))
  }
  if(dist == "t(7)"){
    return(rt(n = size, df = 7))
  }
  if(dist == "tukey(10)"){
    return(rtlambda(n = size, lambda = 10))
  }

  #Sym Short Tail
  if(dist == "uniform(0,1)"){
    return(runif(n = size, min = 0, max = 10))
  }
  if(dist == "beta(1.3,1.3)"){ #alpa=1.3, beta=1.3
```

```
    return(rbeta(n = size, shape1 = 1.3, shape2 = 1.3))
  }
  if(dist == "beta(1.5,1.5)"){ #alpa=1.5, beta=1.5
    return(rbeta(n = size, shape1 = 1.5, shape2 = 1.5))
  }
  if(dist == "tukey(1.5)"){
    return(rtlambda(n = size, lambda = 1.5))
  }
  if(dist == "truncatednormal(2,2)"){
    return(rtruncnorm(n = size, mean = -2, sd = 2))
  }


  #Asym Long Tail
  if(dist == "Weibull(0.5,1)"){ # shape = k
    return(rweibull(n = size, shape = 0.5, scale = 1))
  }
  if(dist == "Weibull(2,1)"){ # shape = k
    return(rweibull(n = size, shape = 2, scale = 1))
  }
  if(dist == "lognormal(0,1)"){
    return(rlnorm(n = size, meanlog = 0, sdlog = 1))
  }
  if(dist == "chisquared(4)"){
    return(rchisq(n = size, df = 4))
  }
  if(dist == "chisquared(10)"){
    return(rchisq(n = size, df = 10))
  }

  #Asym Short Tail
  if(dist == "beta(2,1)"){ #alpa=2, beta=1
    return(rbeta(n = size, shape1 = 2, shape2 = 1))
  }
  if(dist == "beta(3,2)"){
    return(rbeta(n = size, shape1 = 3, shape2 = 2))
  }
  if(dist == "lognormal(0,0.15)"){
    return(rlnorm(n = size, meanlog = 0, sdlog = 0.15))
  }
  if(dist == "lognormal(0,0.25)"){
    return(rlnorm(n = size, meanlog = 0, sdlog = 0.25))
  }
  if(dist == "lognormal(0,0.35)"){
    return(rlnorm(n = size, meanlog = 0, sdlog = 0.35))
  }

  stop(paste("Not handled dist:", dist, sep = " "))
}


dist_ctn <- list("tukey(0.1)", "tukey(0.2)", "tukey(5)", "t(10)",
"laplace(0,10)")
dist_slt <- list("t(1)", "t(2)", "t(4)", "t(7)", "tukey(10)")
dist_sst <- list("uniform(0,1)", "beta(1.3,1.3)", "beta(1.5,1.5)",
"tukey(1.5)", "truncatednormal(2,2)")
```

```
dist_alt <- list("Weibull(0.5,1)", "Weibull(2,1)", "lognormal(0,1)",
"chisquared(4)", "chisquared(10)")
dist_ast <- list("beta(2,1)", "beta(3,2)", "lognormal(0,0.15)",
"lognormal(0,0.25)", "lognormal(0,0.35)")
dist_alternatives = c(dist_ctn, dist_slt, dist_sst, dist_alt, dist_ast)
```

Code Snippet 2: Training Code

```
# Define the control
trControl <- trainControl(method = "cv",
                        number = 10,
                        classProbs = TRUE,
                        savePredictions = "all",
                        search = "grid",
                        allowParallel = TRUE)

# Run training
models <- caretList(dist_type ~
                    size
                  + median
                  + skewness
                  + kurtosis
                  + sigma_1_ratio
                  + sigma_2_ratio
                  + sigma_3_ratio,
  data = train_data,
  methodList = c("rf", "gbm", "svmRadial"),
  metric = "Accuracy",
  tuneLength = 10,
  continue_on_fail = FALSE,
  trControl = trControl)
```

Code Snippet 3: Models evaluation code

```
library("ggplot2")
library("pROC")
library(dplyr)

predict_score<-function(model, sample){
  x <- calc_stats(sample)
  pred <- predict( model, type = "prob", newdata = x)
  return(pred$class_1)
}

get_power_threshold<-function(model, alpha){
   thr <- power_thresholds_df[power_thresholds_df$model==model$method &
power_thresholds_df$test_set=="dev" , c(paste("th_", alpha, sep = ""))]
   return(thr)
}

n_grid<-c(10, 30, 50, 100, 200, 500, 1000)
alph_grid<-c(0.01, 0.05, 0.1)
test_grid<-c(model_names, "SW", "KS","AD", "CVM", "Lillie", "SF", "JB")


#Function to caclulate the sensitivity, specificity, accuracy
calc_statistics<-function(actual_classes, predictions){

  df <- data.frame(threshold = numeric(0), tp = numeric(0), fp = numeric(0),
fn = numeric(0), tn = numeric(0), sensitivity = numeric(0), specificity =
numeric(0), accuracy = numeric(0), stringsAsFactors = FALSE )

  #Create sequence of thresholds
  thresholds<-seq(0.0,1,by=0.05)

  for(threshold in thresholds){
    tp<-0
    fp<-0
    tn<-0
    fn<-0
    for(i in 1:length(predictions)){
      pred<-predictions[i]
      actual <- actual_classes[i]
      if(actual == "class_1"){
        if(pred >= threshold){
          tp = tp + 1
        }else{
          fn = fn + 1
        }
      }else{
        if(pred >= threshold){
          fp = fp + 1
        }else{
          tn = tn + 1
        }
      }
    }
    sensitivity<-ifelse(tp+fn==0,0,tp/(tp+fn))
    specificity<-ifelse(tn+fp==0,0,tn/(tn+fp))
    accuracy<-(tp+tn)/(tp+fp+tn+fn)

    df[nrow(df) + 1, ] = c(threshold, tp, fp, fn, tn, sensitivity,
specificity, accuracy)
  }
  return(df)

}
```

```
calc_power_thresholds<-function(statistics){
  #Calculate thresholds for 0.1, 0.05, 0.01 alpha
  # alpha  = fpr = 1 - Specificity => we search for Specificity 0.99, 0.95,
0.90
  # power = recall = sensitivity = tpr
  thr_0.10 <- statistics[which.min(abs(0.90 - statistics$specificity)),
c("threshold")]
  thr_0.05 <- statistics[which.min(abs(0.95 - statistics$specificity)),
c("threshold")]
  thr_0.01 <- statistics[which.min(abs(0.99 - statistics$specificity)),
c("threshold")]
  thresholds <- list("th_0.1" = thr_0.10, "th_0.05" = thr_0.05, "th_0.01" =
thr_0.01)
  return(thresholds)
}

run_test<-function(test_set, test_set_name, model, applied_threshold){
  pred <- predict(model,  newdata = test_set, type = "prob")
  pred$class <- apply(pred, MARGIN=1, FUN = function(x) ifelse(x["class_1"]
>= applied_threshold, "class_1", "class_0"))
  pred$class <- as.factor(pred$class)

  matrix <-confusionMatrix(pred$class, test_set$dist_type, positive =
"class_1")
  print(matrix)

  # Compute roc
  test.roc <- roc(test_set$dist_type, pred$class_1 )

  stats <- calc_statistics(test_set$dist_type, pred$class_1)
  stats$model <- model$method
  stats$test_set <- test_set_name

  instance_report_df <- test_set
  instance_report_df$model <- model$method
  instance_report_df$test_set <- test_set_name
  instance_report_df$actual <- instance_report_df$dist_type
  instance_report_df <- instance_report_df[, !(names(instance_report_df)
%in% c("dist_type"))]
  instance_report_df$predicted <- pred$class
  instance_report_df$score <- pred$class_1
  instance_report_df$threshold <- applied_threshold

  determine_error_type<-function(row){
    actual_class <- row["actual"]
    predicted_class<-row["predicted"]

    if(actual_class == "class_1"){
      if(predicted_class == "class_1"){
        return("TP")
      }else{
        return("FN")
      }
    }else{
      if(predicted_class == "class_1"){
        return("FP")
      }else{
        return("TN")
      }
    }
  }

  instance_report_df$type <- apply(instance_report_df,  MARGIN=1, FUN =
function(x) determine_error_type(x))

  return(list(stats=stats, roc_auc = test.roc$auc, instance_report =
instance_report_df))
}

all_summary_df <- data.frame(model=character(0), test_set=character(0),
threshold=numeric(0), roc_auc = numeric(0), sensitivity = numeric(0),
specificity = numeric(0), accuracy = numeric(0), stringsAsFactors = FALSE)

all_statistics_df <- data.frame(model=character(0), test_set = character(0),
threshold = numeric(0), tp = numeric(0), fp = numeric(0), fn = numeric(0),
tn = numeric(0), sensitivity = numeric(0), specificity = numeric(0),
accuracy = numeric(0), stringsAsFactors = FALSE )

##### Test models
power_thresholds_df <- data.frame(model=character(0), test_set =
character(0), "th_0.1" = numeric(0), "th_0.05" = numeric(0), "th_0.01" =
numeric(0), stringsAsFactors = FALSE)
power_thresholds_matrix <- matrix(ncol= 5, )

for(model in model_list){
  model_name <- model$method
  print(paste("Calculating quality on model", model_name, sep = " "))

  #Retrieve best threshold based on dev set
  predDev_prob <- predict(model,  newdata = dev_set, type = "prob")
  dev.roc <- roc(dev_set$dist_type, predDev_prob$class_1)
  applied_threshold <- coords(dev.roc, "best", ret = "threshold")$threshold
  print(paste("Best threshold ", applied_threshold))

  #Calculate quality on dev, test, unseen data
  sets <- list("dev"=dev_set, "test"=test_set, "unseen"=unseen_set)
  for(set_name in names(sets)){
    print(paste("Calculating quality on", set_name, "set", sep = " "))
    set <-sets[[set_name]]

    #Run test and get back statistics
    stats_and_roc <- run_test(set, set_name, model, applied_threshold)
    all_statistics_df <- rbind(all_statistics_df, stats_and_roc$stats)
    instance_report <- stats_and_roc$instance_report
    write.csv(x = instance_report, file = paste(stats_dir,
"/instance_report-", model_name, "-", set_name, ".csv", sep = ""))

    #Calculate thresholds on severals alpha
    power_thresholds<-calc_power_thresholds(stats_and_roc$stats)
    power_thresholds_df <- rbind(power_thresholds_df, data.frame(model =
model_name, test_set=set_name, "th_0.1"=power_thresholds$th_0.1,
"th_0.05"=power_thresholds$th_0.05, "th_0.01"=power_thresholds$th_0.01))
```

```
    #Calculate quality on applied threshold
    stat <- stats_and_roc$stats[which.min(abs(applied_threshold -
stats_and_roc$stats$threshold)) ,]
    all_summary_df <- rbind(all_summary_df, data.frame(model=model_name,
test_set=set_name, threshold=applied_threshold,
roc_auc=stats_and_roc$roc_auc, sensitivity=stat$sensitivity,
specificity=stat$specificity, accuracy=stat$accuracy))
  }
}

dev_stats<- all_statistics_df[all_statistics_df$test_set=="dev",]
ggplot(dev_stats, aes(specificity, sensitivity)) +
  geom_path(aes(color = model))+
  scale_x_reverse(expand = c(0,0))+
  scale_y_continuous(expand = c(0,0))+
  geom_abline(intercept = 1, slope = 1, linetype = "dashed")+
  ggtitle(paste("ROC of dev set")) +
  theme_bw()

test_stats<- all_statistics_df[all_statistics_df$test_set=="test",]
ggplot(test_stats, aes(specificity, sensitivity)) +
  geom_path(aes(color = model))+
  scale_x_reverse(expand = c(0,0))+
  scale_y_continuous(expand = c(0,0))+
  geom_abline(intercept = 1, slope = 1, linetype = "dashed")+
  ggtitle(paste("ROC of test set")) +
  theme_bw()

unseen_stats<- all_statistics_df[all_statistics_df$test_set=="unseen",]
ggplot(unseen_stats, aes(specificity, sensitivity)) +
  geom_path(aes(color = model))+
  scale_x_reverse(expand = c(0,0))+
  scale_y_continuous(expand = c(0,0))+
  geom_abline(intercept = 1, slope = 1, linetype = "dashed")+
  ggtitle(paste("ROC of unseen data")) +
  theme_bw()


print(power_thresholds_df)
print(all_summary_df)
print(all_statistics_df)

ggplot(dev_stats, aes(threshold, accuracy)) +
  geom_path(aes(color = model))+
  scale_x_reverse(expand = c(0,0))+
  scale_y_continuous(expand = c(0,0))+
  geom_abline(intercept = 1, slope = 1, linetype = "dashed")+
    theme_bw()

write.csv(x = power_thresholds_df, file = power_thresholds_file)
write.csv(x = all_summary_df, file = summary_stats_file)
write.csv(x = all_statistics_df, file = stats_file)
```

Code Snippet 4: MonteCarlo simulation code

```
library(MonteCarlo)
library("nortest") #AD, cvm, lillie #https://cran.r-
project.org/web/packages/nortest/nortest.pdf
library("extraDistr") #rtlambda
library("truncnorm") #rtruncnorm
library("tseries") #jarque.bera.test
library(parallel)
library(MASS)

test_is_alternative<-function(model, sample, alpha){
  pred <- predict_score(model, sample)
  thr <- get_power_threshold(model, alpha)
  return(pred > thr)
}

normality_test<-function(n, dist, test, alpha, family){
  sample <- create_alternative_sample(dist = dist, size = n)
  if(test == "SW"){ #Shapiro
    test_result <- shapiro.test(sample)
    decision <- test_result$p.value <= alpha
  }else if (test == "KS"){#KS
    test_result <- ks.test(sample, "pnorm", mean=mean(sample), sd =
sd(sample))
    decision <- test_result$p.value <= alpha
  }else if (test == "AD"){ #Anderson Darling
    test_result <- ad.test(sample)
    decision <- test_result$p.value <= alpha
  }else if(test == "CVM"){#Cramer-von Mises Test
    test_result <- cvm.test(sample)
    decision <- test_result$p.value <= alpha
  }else if(test == "Lillie"){ #Lilliefors
    test_result <- lillie.test(sample)
    decision <- test_result$p.value <= alpha
  }else if(test == "SF"){ "Shapiro-Francia"
    test_result <- sf.test(sample)
    decision <- test_result$p.value <= alpha
  }else if(test == "JB"){ #Jarque-Bera
    test_result <- jarque.bera.test(sample)
    decision <- test_result$p.value <= alpha
  }else if(test %in% model_names){ #Proposed tests
    decision = test_is_alternative(model_list[[test]], sample, alpha) == 1
  }else {
    stop(paste("normality_test: Not handled test", test))
  }

  # return result:
  return(list("power"=decision))
}

run_test<-function(family, dists){
  set.seed(100)

  dist_grid<-dists
  family_grid <- list(family)
  param_list=list("n"=n_grid, "dist"=dist_grid, "test"=test_grid,
"alpha"=alph_grid, "family"=family_grid)

  system.time({
    MC_result<-MonteCarlo(func=normality_test, nrep=1000,
param_list=param_list, ncpus =1, max_grid = 5000)
    saveRDS(MC_result, paste(power_dir, "/",family, ".rds", sep = ""))

    df<-MakeFrame(MC_result)
    write.csv(df, paste(power_dir, "/",family, ".csv", sep = ""))
  })
}

run_test("ctn", dist_ctn)
run_test("alt", dist_alt)
run_test("slt", dist_slt)
run_test("ast", dist_ast)
run_test("sst", dist_sst)
```

Appendix 2: Figures

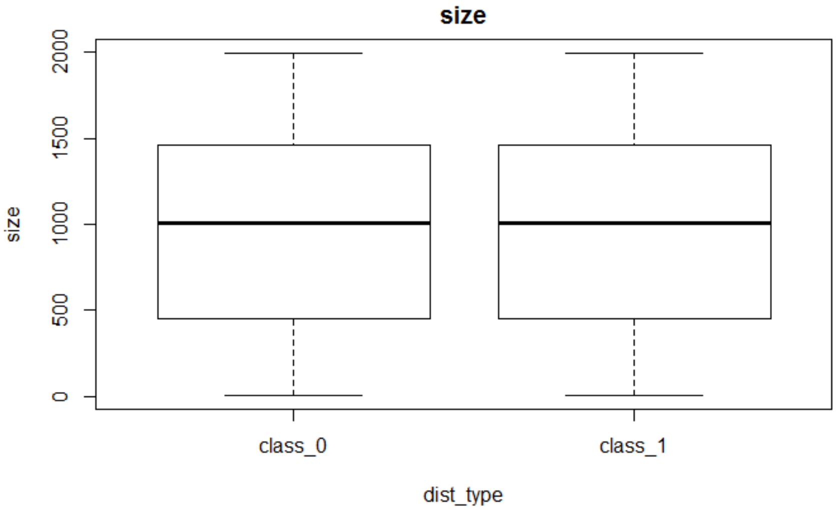

Figure 30 Boxplot for size per dist_type

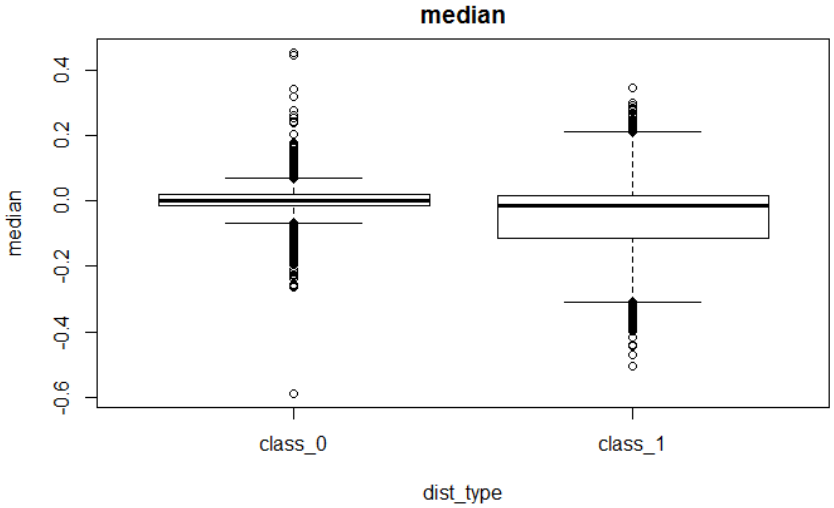

Figure 31 Boxplot for median per dist_type

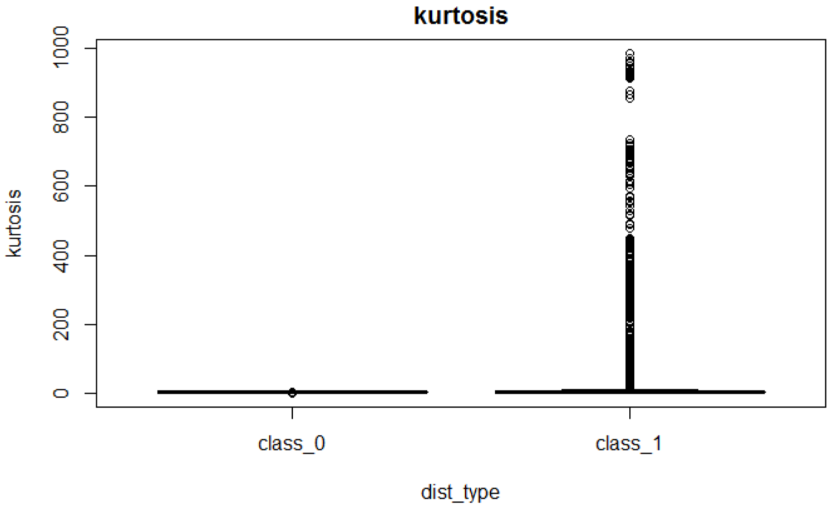

Figure 32 Boxplot for kurtosis per dist_type

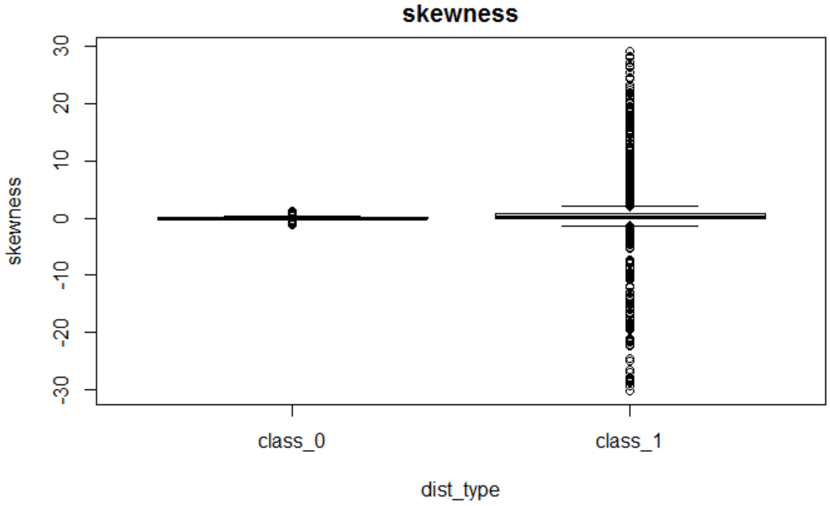

Figure 33 Boxplot for skewness per dist_type

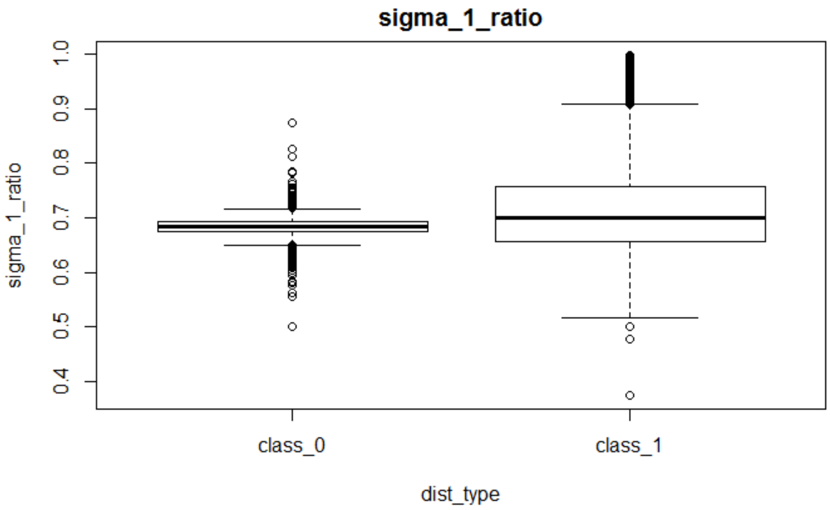

Figure 34 Boxplot for sigma_1_ratio per dist_type

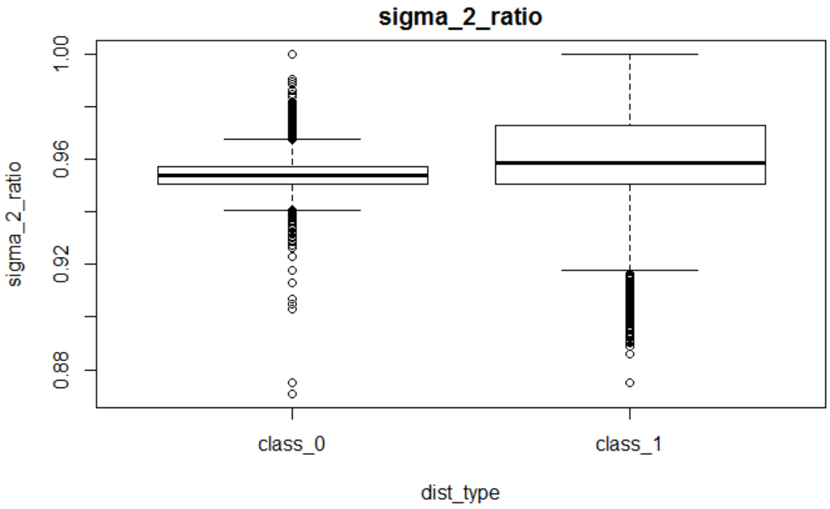

Figure 35 Boxplot for sigma_2_ratio per dist_type

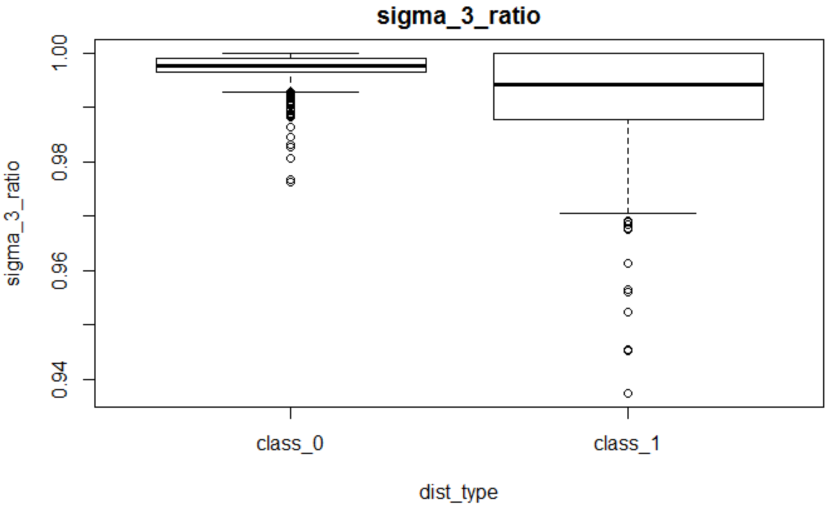

Figure 36 Boxplot for sigma_3_ratio per dist_type

Appendix 3: Tables

Table 18 Sample sizes

8	16	23	31	42	55	65	72	73
86	91	95	103	129	151	173	178	185
194	212	214	215	219	246	247	257	265
266	272	282	284	292	298	306	310	317
330	332	338	339	360	376	382	390	394
406	409	428	436	445	464	480	503	504
525	543	549	552	563	568	578	580	591
593	602	631	634	654	657	659	664	679
693	702	718	723	744	761	762	773	775
788	790	811	825	844	859	902	906	917
920	933	937	943	954	959	962	963	981
1005	1009	1010	1016	1023	1024	1028	1045	1047
1060	1064	1077	1081	1087	1099	1106	1113	1135
1137	1138	1157	1174	1216	1230	1231	1242	1243
1244	1254	1256	1263	1265	1269	1275	1279	1297
1302	1317	1323	1358	1361	1369	1376	1398	1404
1405	1420	1440	1444	1445	1460	1470	1489	1512
1525	1536	1565	1594	1599	1613	1621	1625	1688
1697	1698	1703	1721	1724	1730	1738	1746	1773
1783	1800	1811	1815	1833	1841	1845	1847	1867
1870	1881	1892	1895	1901	1904	1907	1908	1910
1914	1924	1929	1933	1942	1955	1959	1960	1987
1995	1998							

Table 19: FN instances with the lowest score from the validation set on rf model

Distribution family	Distribution	Size	Median	Skewness	Kurtosis	Sigma ratio 1	Sigma ratio 2	Sigma ratio 3	Score
sym_short_tail	truncatednormal (2,2)	1895	0.009	0.008	2.982	0.678	0.955	0.998	0.014
sym_short_tail	truncatednormal (2,2)	1724	-0.004	0.025	2.958	0.684	0.955	0.997	0.026
sym_short_tail	truncatednormal (2,2)	1892	0.018	-0.002	3.010	0.683	0.954	0.996	0.028
sym_short_tail	truncatednormal (2,2)	844	-0.014	0.068	2.914	0.692	0.953	0.999	0.032
sym_short_tail	truncatednormal (2,2)	1243	-0.010	0.061	2.990	0.683	0.957	0.998	0.032
sym_short_tail	truncatednormal (2,2)	543	0.008	-0.117	3.080	0.680	0.948	0.998	0.034
sym_short_tail	truncatednormal (2,2)	552	0.000	-0.042	2.853	0.672	0.951	1.000	0.038
sym_short_tail	truncatednormal (2,2)	1525	0.019	-0.078	2.971	0.679	0.950	0.997	0.042
sym_short_tail	truncatednormal (2,2)	1621	0.011	-0.050	2.995	0.682	0.958	0.998	0.042
sym_short_tail	truncatednormal (2,2)	1077	0.032	-0.080	3.011	0.685	0.948	0.998	0.048
sym_short_tail	truncatednormal (2,2)	1995	0.010	-0.024	2.960	0.695	0.951	0.999	0.05
sym_short_tail	truncatednormal (2,2)	679	0.017	0.075	3.199	0.700	0.951	0.996	0.052
sym_short_tail	truncatednormal (2,2)	1323	-0.007	-0.115	3.091	0.683	0.953	0.996	0.052
sym_short_tail	truncatednormal (2,2)	654	-0.009	-0.038	2.948	0.680	0.951	1.000	0.054
close_normal	tukey(0.2)	409	0.054	-0.145	2.914	0.675	0.963	0.998	0.058
close_normal	tukey(0.2)	811	-0.012	0.030	3.022	0.692	0.956	0.998	0.064
sym_short_tail	truncatednormal (2,2)	1698	-0.013	0.078	2.975	0.674	0.953	0.998	0.068
sym_short_tail	truncatednormal (2,2)	1064	0.025	0.001	3.054	0.685	0.949	0.997	0.074
sym_short_tail	truncatednormal (2,2)	360	-0.059	0.014	3.041	0.681	0.953	0.997	0.082
sym_short_tail	truncatednormal (2,2)	659	0.020	0.035	2.889	0.684	0.959	0.998	0.084

Table 20: Overall tests power on 1% significance level

Size	“new_test”	SW	AD	CVM	JB	KS	Lillie	SF
10	**0.39**	0.11	0.11	0.11	0.05	0	0.09	0.11
30	**0.5**	0.26	0.25	0.24	0.22	0.06	0.21	0.26
50	**0.59**	0.38	0.34	0.32	0.29	0.12	0.27	0.35
100	**0.71**	0.6	0.53	0.48	0.41	0.18	0.38	0.54
200	**0.79**	0.76	0.71	0.67	0.67	0.23	0.57	0.74
500	**0.86**	0.85	0.83	0.81	0.82	0.44	0.77	0.85
1000	**0.89**	0.88	0.86	0.86	0.84	0.67	0.83	0.88

Table 21: Overall tests power on 5% significance level

Size	“new_test”	SW	AD	CVM	JB	KS	Lillie	SF
10	**0.78**	0.19	0.19	0.19	0.08	0.02	0.17	0.2
30	**0.73**	0.39	0.37	0.34	0.26	0.11	0.3	0.36
50	**0.76**	0.53	0.48	0.45	0.35	0.17	0.38	0.48
100	**0.83**	0.71	0.66	0.62	0.55	0.22	0.53	0.67
200	**0.87**	0.82	0.79	0.76	0.77	0.32	0.7	0.81
500	**0.89**	0.87	0.86	0.85	0.84	0.6	0.83	0.87
1000	**0.91**	0.9	0.88	0.88	0.86	0.77	0.86	0.89

Table 22: Overall tests power on 10% significance level

Size	“new_test”	SW	AD	CVM	JB	KS	Lillie	SF
10	**0.91**	0.26	0.27	0.26	0.1	0.03	0.24	0.27
30	**0.86**	0.48	0.46	0.43	0.3	0.14	0.38	0.45
50	**0.86**	0.61	0.56	0.52	0.4	0.2	0.46	0.57
100	**0.89**	0.76	0.72	0.69	0.65	0.26	0.62	0.74
200	**0.9**	0.84	0.82	0.8	0.8	0.41	0.76	0.84
500	**0.91**	0.89	0.87	0.87	0.85	0.69	0.85	0.89
1000	**0.92**	0.91	0.9	0.89	0.88	0.8	0.88	0.91

Table 23: Tests power per distribution at 1% significance level

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	laplace(0,10)	10.00	**0.32**	0.05	0.07	0.06	0.02	0.00	0.05	0.07
		30.00	**0.55**	0.19	0.18	0.20	0.27	0.00	0.15	0.25
		50.00	**0.68**	0.35	0.35	0.33	0.38	0.00	0.22	0.39
		100.00	**0.88**	0.66	0.66	0.67	0.68	0.01	0.46	0.68
		200.00	**0.98**	0.92	0.95	0.93	0.93	0.08	0.84	0.94

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.68	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	t(10)	10.00	**0.28**	0.01	0.02	0.02	0.01	0.00	0.01	0.02
		30.00	**0.29**	0.06	0.03	0.03	0.09	0.00	0.02	0.07
		50.00	**0.34**	0.08	0.04	0.04	0.14	0.00	0.02	0.07
		100.00	**0.44**	0.14	0.06	0.06	0.21	0.00	0.03	0.16
		200.00	**0.58**	0.20	0.11	0.08	0.32	0.00	0.04	0.28
		500.00	**0.80**	0.47	0.27	0.19	0.64	0.00	0.10	0.54
		1000.00	0.88	0.79	0.57	0.46	**0.89**	0.00	0.22	0.84
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	tukey(0.1)	10.00	**0.29**	0.02	0.01	0.01	0.00	0.00	0.01	0.01
		30.00	**0.20**	0.02	0.02	0.01	0.02	0.00	0.02	0.02
		50.00	**0.19**	0.01	0.01	0.01	0.04	0.00	0.02	0.02
		100.00	**0.22**	0.02	0.02	0.01	0.04	0.00	0.01	0.04
		200.00	**0.22**	0.03	0.02	0.01	0.06	0.00	0.02	0.02
		500.00	**0.20**	0.04	0.04	0.02	0.10	0.00	0.02	0.06
		1000.00	**0.11**	0.08	0.04	0.04	0.12	0.00	0.03	0.07
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	tukey(0.2)	10.00	**0.27**	0.01	0.00	0.00	0.00	0.00	0.01	0.01
		30.00	**0.14**	0.00	0.01	0.01	0.00	0.00	0.00	0.00
		50.00	**0.08**	0.01	0.01	0.01	0.00	0.00	0.00	0.00
		100.00	**0.06**	0.00	0.01	0.01	0.00	0.00	0.00	0.00
		200.00	**0.04**	0.01	0.01	0.01	0.00	0.00	0.01	0.00
		500.00	**0.04**	0.03	0.03	0.03	0.00	0.00	0.01	0.01
		1000.00	**0.46**	0.11	0.06	0.04	0.05	0.00	0.02	0.03
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	tukey(5)	10.00	**0.27**	0.03	0.02	0.03	0.00	0.00	0.01	0.02
		30.00	**0.11**	0.01	0.04	0.06	0.01	0.00	0.04	0.02
		50.00	0.06	0.02	0.08	**0.12**	0.00	0.00	0.06	0.01
		100.00	0.05	0.12	0.27	**0.29**	0.00	0.00	0.22	0.06
		200.00	0.23	0.53	**0.69**	**0.69**	0.00	0.00	0.52	0.32
		500.00	0.66	**1.00**	**1.00**	**1.00**	0.00	0.21	0.97	0.99
		1000.00	0.85	**1.00**	**1.00**	**1.00**	0.00	0.87	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	Average CTN	10.00	**0.29**	0.02	0.02	0.02	0.01	0.00	0.02	0.03
		30.00	**0.26**	0.06	0.06	0.06	0.08	0.00	0.05	0.07
		50.00	**0.27**	0.09	0.10	0.10	0.11	0.00	0.06	0.10
		100.00	**0.33**	0.19	0.20	0.21	0.19	0.00	0.14	0.19
		200.00	**0.41**	0.34	0.36	0.34	0.26	0.02	0.29	0.31

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500.00	**0.54**	0.51	0.47	0.45	0.35	0.18	0.42	0.52
		1000.00	**0.66**	0.60	0.53	0.51	0.41	0.37	0.45	0.59
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	chisquared(10)	10.00	**0.37**	0.04	0.04	0.03	0.01	0.00	0.03	0.04
		30.00	**0.50**	0.19	0.14	0.11	0.14	0.00	0.08	0.17
		50.00	**0.69**	0.38	0.25	0.20	0.29	0.00	0.15	0.32
		100.00	**0.93**	0.78	0.60	0.50	0.61	0.00	0.35	0.73
		200.00	**1.00**	0.99	0.94	0.88	0.95	0.03	0.71	0.98
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.51	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.98	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	chisquared(4)	10.00	**0.47**	0.10	0.09	0.07	0.03	0.00	0.05	0.10
		30.00	**0.65**	0.52	0.44	0.37	0.35	0.00	0.25	0.46
		50.00	**0.90**	0.83	0.74	0.66	0.62	0.01	0.44	0.78
		100.00	**1.00**	**1.00**	0.99	0.96	0.95	0.07	0.83	**1.00**
		200.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.50	0.99	**1.00**
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	lognormal(0,1)	10.00	**0.61**	0.43	0.38	0.37	0.19	0.00	0.30	0.38
		30.00	0.92	**0.97**	0.95	0.94	0.81	0.14	0.81	0.95
		50.00	0.99	**1.00**	**1.00**	**1.00**	0.98	0.50	0.98	**1.00**
		100.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.95	**1.00**	**1.00**
		200.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	Weibull(0.5,1)	10.00	0.70	**0.78**	0.74	0.69	0.33	0.00	0.56	0.69
		30.00	0.98	**1.00**	**1.00**	**1.00**	0.97	0.60	**1.00**	**1.00**
		50.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.98	**1.00**	**1.00**
		100.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		200.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	Weibull(2,1)	10.00	**0.34**	0.02	0.02	0.01	0.00	0.00	0.02	0.02
		30.00	**0.39**	0.07	0.07	0.05	0.06	0.00	0.04	0.08
		50.00	**0.52**	0.16	0.11	0.10	0.11	0.00	0.06	0.15
		100.00	**0.84**	0.57	0.31	0.26	0.28	0.00	0.17	0.43
		200.00	**0.95**	**0.95**	0.78	0.65	0.70	0.01	0.43	0.90

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500.00	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.14	0.92	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.71	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	Average ALT	10.00	**0.50**	0.27	0.25	0.23	0.11	0.00	0.19	0.25
		30.00	**0.69**	0.55	0.52	0.49	0.47	0.15	0.44	0.53
		50.00	**0.82**	0.67	0.62	0.59	0.60	0.30	0.53	0.65
		100.00	**0.95**	0.87	0.78	0.74	0.77	0.40	0.67	0.83
		200.00	**0.99**	**0.99**	0.94	0.91	0.93	0.51	0.83	0.98
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.73	0.98	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.94	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	beta(2,1)	10.00	**0.30**	0.02	0.03	0.03	0.00	0.00	0.03	0.03
		30.00	**0.26**	0.22	0.18	0.16	0.02	0.00	0.09	0.12
		50.00	0.28	**0.52**	0.44	0.34	0.03	0.00	0.20	0.34
		100.00	0.44	**0.98**	0.91	0.78	0.17	0.00	0.56	0.90
		200.00	0.67	**1.00**	**1.00**	0.99	0.99	0.11	0.96	**1.00**
		500.00	0.87	**1.00**	**1.00**	**1.00**	**1.00**	0.95	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	beta(3,2)	10.00	**0.29**	0.01	0.01	0.01	0.00	0.00	0.01	0.01
		30.00	**0.21**	0.02	0.02	0.02	0.00	0.00	0.02	0.01
		50.00	**0.21**	0.04	0.05	0.03	0.00	0.00	0.02	0.02
		100.00	**0.39**	0.20	0.16	0.12	0.00	0.00	0.08	0.08
		200.00	0.65	**0.73**	0.54	0.37	0.10	0.00	0.22	0.51
		500.00	0.94	**1.00**	0.99	0.95	**1.00**	0.02	0.73	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.29	0.99	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	lognormal(0,0.15)	10.00	**0.30**	0.02	0.02	0.01	0.00	0.00	0.02	0.01
		30.00	**0.31**	0.04	0.04	0.04	0.06	0.00	0.02	0.04
		50.00	**0.36**	0.10	0.05	0.05	0.09	0.00	0.03	0.08
		100.00	**0.59**	0.19	0.12	0.08	0.18	0.00	0.07	0.18
		200.00	**0.71**	0.39	0.25	0.20	0.36	0.00	0.17	0.38
		500.00	**0.90**	0.87	0.73	0.65	0.85	0.01	0.46	0.88
		1000.00	0.97	**1.00**	0.98	0.96	**1.00**	0.08	0.83	0.99
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	lognormal(0,0.25)	10.00	**0.37**	0.02	0.04	0.03	0.01	0.00	0.02	0.04
		30.00	**0.46**	0.13	0.10	0.07	0.14	0.00	0.06	0.12
		50.00	**0.59**	0.25	0.19	0.14	0.21	0.00	0.09	0.27
		100.00	**0.84**	0.55	0.42	0.36	0.44	0.00	0.23	0.48
		200.00	**0.97**	0.90	0.78	0.69	0.85	0.01	0.47	0.88

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500.00	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.22	0.95	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.82	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	lognormal(0,0.35)	10.00	**0.38**	0.06	0.05	0.04	0.02	0.00	0.03	0.05
		30.00	**0.57**	0.25	0.17	0.16	0.24	0.00	0.12	0.27
		50.00	**0.78**	0.47	0.38	0.30	0.41	0.00	0.22	0.48
		100.00	**0.97**	0.87	0.75	0.67	0.77	0.01	0.48	0.80
		200.00	**1.00**	**1.00**	0.99	0.96	0.99	0.09	0.87	**1.00**
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.78	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	Average AST	10.00	**0.33**	0.03	0.03	0.02	0.01	0.00	0.02	0.03
		30.00	**0.36**	0.13	0.10	0.09	0.09	0.00	0.06	0.11
		50.00	**0.44**	0.28	0.22	0.17	0.15	0.00	0.11	0.24
		100.00	**0.65**	0.56	0.47	0.40	0.31	0.00	0.28	0.49
		200.00	**0.80**	**0.80**	0.71	0.64	0.66	0.04	0.54	0.75
		500.00	0.94	**0.97**	0.94	0.92	**0.97**	0.40	0.83	0.98
		1000.00	0.99	**1.00**	**1.00**	0.99	**1.00**	0.64	0.96	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(1)	10.00	**0.67**	0.44	0.50	0.47	0.36	0.03	0.43	0.48
		30.00	**0.98**	0.94	0.93	0.94	0.90	0.55	0.89	0.95
		50.00	**1.00**	0.99	**1.00**	**1.00**	0.99	0.83	0.99	**1.00**
		100.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**
		200.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(2)	10.00	**0.45**	0.17	0.19	0.20	0.12	0.00	0.17	0.21
		30.00	**0.79**	0.58	0.57	0.54	0.61	0.09	0.43	0.60
		50.00	**0.92**	0.78	0.76	0.75	0.81	0.21	0.65	0.83
		100.00	**0.99**	0.97	0.96	0.95	0.97	0.49	0.91	0.97
		200.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.88	**1.00**	**1.00**
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(4)	10.00	**0.33**	0.06	0.04	0.05	0.03	0.00	0.06	0.07
		30.00	**0.51**	0.20	0.15	0.14	0.26	0.00	0.10	0.24
		50.00	**0.64**	0.34	0.24	0.24	0.41	0.02	0.17	0.37
		100.00	**0.86**	0.57	0.49	0.45	0.69	0.02	0.32	0.64
		200.00	**0.97**	0.86	0.80	0.72	0.92	0.08	0.58	0.88

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500.00	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.36	0.94	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.85	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(7)	10.00	**0.30**	0.03	0.03	0.02	0.01	0.00	0.02	0.03
		30.00	**0.33**	0.08	0.07	0.04	0.12	0.00	0.04	0.11
		50.00	**0.44**	0.13	0.08	0.07	0.19	0.00	0.04	0.16
		100.00	**0.58**	0.23	0.15	0.10	0.34	0.00	0.08	0.27
		200.00	**0.78**	0.44	0.25	0.22	0.53	0.00	0.12	0.48
		500.00	**0.96**	0.81	0.62	0.53	0.88	0.00	0.30	0.84
		1000.00	**1.00**	0.97	0.94	0.89	0.99	0.02	0.65	0.99
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	tukey(10)	10.00	**0.48**	0.36	0.42	0.46	0.14	0.00	0.41	0.36
		30.00	0.80	0.81	0.95	**0.97**	0.40	0.13	0.91	0.85
		50.00	0.93	0.98	**1.00**	**1.00**	0.60	0.50	0.99	0.98
		100.00	0.99	**1.00**	**1.00**	**1.00**	0.88	0.99	**1.00**	**1.00**
		200.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	Average SLT	10.00	**0.45**	0.21	0.24	0.24	0.13	0.01	0.22	0.23
		30.00	**0.68**	0.52	0.53	0.53	0.46	0.15	0.47	0.55
		50.00	**0.79**	0.64	0.62	0.61	0.60	0.31	0.57	0.67
		100.00	**0.88**	0.75	0.72	0.70	0.78	0.50	0.66	0.78
		200.00	**0.95**	0.86	0.81	0.79	0.89	0.59	0.74	0.87
		500.00	**0.99**	0.96	0.92	0.90	0.98	0.67	0.85	0.97
		1000.00	**1.00**	0.99	0.99	0.98	1.00	0.77	0.93	1.00
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	beta(1.3,1.3)	10.00	**0.36**	0.00	0.00	0.01	0.00	0.00	0.01	0.00
		30.00	**0.56**	0.03	0.04	0.02	0.00	0.00	0.02	0.01
		50.00	**0.69**	0.12	0.12	0.10	0.00	0.00	0.04	0.02
		100.00	**0.88**	0.69	0.46	0.26	0.00	0.00	0.12	0.32
		200.00	0.99	**1.00**	0.93	0.78	0.68	0.00	0.41	0.97
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.05	0.97	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.77	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	beta(1.5,1.5)	10.00	**0.38**	0.01	0.01	0.01	0.00	0.00	0.01	0.01
		30.00	**0.46**	0.02	0.03	0.03	0.00	0.00	0.01	0.00
		50.00	**0.57**	0.06	0.06	0.04	0.00	0.00	0.03	0.01
		100.00	**0.79**	0.44	0.27	0.18	0.00	0.00	0.07	0.18
		200.00	0.96	**0.98**	0.79	0.56	0.36	0.00	0.26	0.85

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.02	0.86	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.45	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	truncatednormal(2,2)	10.00	**0.30**	0.01	0.01	0.01	0.00	0.00	0.01	0.01
		30.00	**0.18**	0.01	0.01	0.01	0.02	0.00	0.01	0.01
		50.00	**0.14**	0.01	0.01	0.01	0.02	0.00	0.01	0.01
		100.00	**0.14**	0.01	0.01	0.01	0.02	0.00	0.01	0.01
		200.00	**0.10**	0.01	0.01	0.01	0.01	0.00	0.01	0.01
		500.00	**0.05**	0.01	0.01	0.02	0.02	0.00	0.01	0.01
		1000.00	**0.04**	0.01	0.01	0.01	0.01	0.00	0.00	0.01
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	tukey(1.5)	10.00	**0.42**	0.01	0.01	0.02	0.00	0.00	0.01	0.01
		30.00	**0.75**	0.14	0.10	0.09	0.00	0.00	0.04	0.03
		50.00	**0.90**	0.46	0.36	0.23	0.00	0.00	0.10	0.21
		100.00	**1.00**	0.99	0.88	0.70	0.01	0.00	0.35	0.88
		200.00	**1.00**	**1.00**	**1.00**	0.98	0.99	0.02	0.85	**1.00**
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.66	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	uniform(0,1)	10.00	**0.42**	0.02	0.01	0.02	0.00	0.00	0.01	0.00
		30.00	**0.69**	0.09	0.08	0.07	0.00	0.00	0.04	0.03
		50.00	**0.87**	0.36	0.27	0.16	0.00	0.00	0.09	0.12
		100.00	**0.99**	0.93	0.80	0.59	0.00	0.00	0.24	0.75
		200.00	**1.00**	**1.00**	**1.00**	0.97	0.97	0.01	0.75	**1.00**
		500.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.44	**1.00**	**1.00**
		1000.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	Average SST	10.00	**0.38**	0.01	0.01	0.01	0.00	0.00	0.01	0.01
		30.00	**0.53**	0.06	0.05	0.04	0.00	0.00	0.02	0.02
		50.00	**0.63**	0.20	0.16	0.11	0.00	0.00	0.05	0.07
		100.00	**0.76**	0.61	0.48	0.35	0.01	0.00	0.16	0.43
		200.00	**0.81**	0.80	0.75	0.66	0.60	0.01	0.46	0.77
		500.00	**0.81**	0.80	0.80	0.80	0.80	0.23	0.77	0.80
		1000.00	**0.81**	0.80	0.80	0.80	0.80	0.64	0.80	0.80

Table 24: Tests power per distribution at 5% significance level

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	laplace(0,10)	10	**0.68**	0.16	0.18	0.16	0.06	0.00	0.13	0.17
		30	**0.69**	0.35	0.37	0.36	0.36	0.02	0.28	0.43
		50	**0.80**	0.51	0.53	0.50	0.52	0.02	0.45	0.59
		100	**0.94**	0.80	0.83	0.81	0.79	0.09	0.71	0.86
		200	**1.00**	0.97	0.99	0.98	0.97	0.33	0.94	0.99
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.95	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	t(10)	10	**0.73**	0.06	0.08	0.06	0.02	0.00	0.07	0.10
		30	**0.51**	0.12	0.09	0.09	0.13	0.00	0.09	0.15
		50	**0.49**	0.14	0.11	0.12	0.17	0.00	0.08	0.20
		100	**0.63**	0.24	0.15	0.13	0.30	0.00	0.10	0.28
		200	**0.76**	0.36	0.24	0.21	0.46	0.00	0.16	0.42
		500	**0.87**	0.64	0.48	0.42	0.75	0.00	0.29	0.73
		1000	0.92	0.90	0.76	0.69	**0.95**	0.03	0.51	0.91
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	tukey(0.1)	10	**0.69**	0.06	0.06	0.06	0.01	0.00	0.05	0.07
		30	**0.46**	0.07	0.06	0.06	0.06	0.00	0.06	0.07
		50	**0.43**	0.07	0.05	0.05	0.06	0.00	0.05	0.08
		100	**0.44**	0.10	0.07	0.07	0.10	0.00	0.05	0.09
		200	**0.43**	0.09	0.08	0.07	0.12	0.00	0.06	0.11
		500	**0.34**	0.11	0.12	0.10	0.16	0.00	0.08	0.16
		1000	0.23	0.18	0.15	0.12	**0.25**	0.00	0.11	0.24
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	tukey(0.2)	10	**0.70**	0.05	0.05	0.04	0.01	0.00	0.04	0.05
		30	**0.44**	0.03	0.05	0.05	0.00	0.00	0.04	0.03
		50	**0.30**	0.05	0.04	0.05	0.01	0.00	0.04	0.03
		100	**0.30**	0.04	0.05	0.05	0.00	0.00	0.06	0.02
		200	**0.26**	0.05	0.06	0.04	0.01	0.00	0.05	0.02
		500	**0.28**	0.11	0.09	0.09	0.07	0.00	0.07	0.05
		1000	**0.65**	0.29	0.20	0.17	0.27	0.00	0.11	0.17
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	tukey(5)	10	**0.66**	0.06	0.09	0.10	0.02	0.00	0.10	0.09
		30	**0.40**	0.07	0.15	0.17	0.02	0.00	0.13	0.07
		50	**0.34**	0.13	0.24	0.27	0.01	0.00	0.23	0.11
		100	0.37	0.36	0.53	**0.56**	0.00	0.01	0.43	0.26
		200	0.53	**0.89**	0.88	0.86	0.00	0.08	0.77	0.74
		500	0.77	**1.00**	**1.00**	**1.00**	0.00	0.61	**1.00**	**1.00**
		1000	0.96	**1.00**	**1.00**	**1.00**	0.00	0.99	**1.00**	**1.00**

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	Average CTN	10	**0.69**	0.08	0.09	0.08	0.02	0.00	0.08	0.10
		30	**0.50**	0.13	0.14	0.15	0.11	0.00	0.12	0.15
		50	**0.47**	0.18	0.19	0.20	0.15	0.00	0.17	0.20
		100	**0.54**	0.31	0.33	0.32	0.24	0.02	0.27	0.30
		200	**0.60**	0.47	0.45	0.43	0.31	0.08	0.40	0.46
		500	**0.65**	0.57	0.54	0.52	0.40	0.31	0.49	0.59
		1000	**0.75**	0.67	0.62	0.60	0.49	0.40	0.55	0.66
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	chisquared(10)	10	**0.82**	0.12	0.10	0.10	0.04	0.00	0.08	0.11
		30	**0.80**	0.36	0.33	0.28	0.24	0.00	0.20	0.32
		50	**0.88**	0.56	0.47	0.40	0.44	0.02	0.33	0.57
		100	**0.99**	0.90	0.80	0.73	0.80	0.06	0.61	0.88
		200	**1.00**	**1.00**	0.99	0.96	0.99	0.20	0.89	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.86	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	chisquared(4)	10	**0.87**	0.22	0.22	0.23	0.07	0.00	0.16	0.25
		30	**0.94**	0.76	0.67	0.59	0.48	0.02	0.49	0.68
		50	**0.97**	0.96	0.90	0.83	0.75	0.08	0.71	0.92
		100	**1.00**	**1.00**	**1.00**	0.99	0.99	0.36	0.96	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.87	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	lognormal(0,1)	10	**0.91**	0.59	0.58	0.54	0.27	0.03	0.47	0.59
		30	**1.00**	0.99	0.97	0.98	0.92	0.41	0.93	0.99
		50	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.80	**0.99**	**1.00**
		100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	Weibull(0.5,1)	10	**0.96**	0.89	0.86	0.86	0.44	0.09	0.75	0.89
		30	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.90	**1.00**	**1.00**
		50	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	Weibull(2,1)	10	**0.79**	0.08	0.08	0.07	0.02	0.00	0.06	0.09
		30	**0.72**	0.22	0.18	0.14	0.12	0.00	0.12	0.19
		50	**0.80**	0.42	0.34	0.27	0.22	0.00	0.19	0.34
		100	**0.97**	0.80	0.62	0.51	0.48	0.01	0.40	0.68
		200	**1.00**	0.99	0.94	0.86	0.94	0.06	0.73	0.98
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.51	0.99	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.96	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	Average ALT	10	**0.87**	0.38	0.37	0.36	0.17	0.02	0.30	0.39
		30	**0.89**	0.67	0.63	0.60	0.55	0.27	0.55	0.64
		50	**0.93**	0.79	0.74	0.70	0.68	0.38	0.64	0.77
		100	**0.99**	0.94	0.88	0.85	0.85	0.49	0.79	0.91
		200	**1.00**	**1.00**	0.99	0.96	0.99	0.63	0.92	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.87	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	beta(2,1)	10	**0.72**	0.13	0.14	0.13	0.01	0.00	0.11	0.10
		30	**0.62**	0.48	0.44	0.34	0.04	0.00	0.27	0.36
		50	0.68	**0.86**	0.74	0.63	0.12	0.01	0.45	0.71
		100	0.78	0.99	0.98	0.94	0.74	0.08	0.82	**1.00**
		200	0.91	**1.00**	**1.00**	**1.00**	**1.00**	0.50	0.99	**1.00**
		500	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	beta(3,2)	10	**0.74**	0.04	0.04	0.05	0.00	0.00	0.05	0.04
		30	**0.59**	0.10	0.10	0.08	0.00	0.00	0.07	0.06
		50	**0.54**	0.20	0.19	0.16	0.01	0.00	0.13	0.11
		100	**0.69**	0.50	0.38	0.30	0.05	0.00	0.23	0.33
		200	0.88	**0.95**	0.79	0.66	0.64	0.01	0.50	0.84
		500	0.99	**1.00**	**1.00**	0.99	**1.00**	0.21	0.93	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.79	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	lognormal(0,0.15)	10	**0.75**	0.08	0.07	0.06	0.02	0.00	0.05	0.09
		30	**0.60**	0.12	0.11	0.11	0.07	0.00	0.09	0.14
		50	**0.64**	0.19	0.15	0.14	0.16	0.00	0.11	0.20
		100	**0.77**	0.33	0.29	0.23	0.31	0.00	0.19	0.33
		200	**0.85**	0.62	0.46	0.44	0.55	0.01	0.36	0.60
		500	**0.96**	**0.96**	0.86	0.83	0.94	0.07	0.70	0.95
		1000	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.36	0.94	**1.00**

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	lognormal(0,0.25)	10	**0.80**	0.10	0.10	0.08	0.04	0.00	0.07	0.11
		30	**0.75**	0.26	0.21	0.21	0.20	0.00	0.17	0.27
		50	**0.83**	0.44	0.34	0.31	0.33	0.00	0.25	0.42
		100	**0.96**	0.74	0.64	0.56	0.65	0.02	0.45	0.72
		200	**0.99**	0.97	0.91	0.86	0.94	0.11	0.77	0.96
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.61	0.99	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.98	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	lognormal(0,0.35)	10	**0.81**	0.16	0.13	0.14	0.05	0.00	0.11	0.15
		30	**0.86**	0.46	0.39	0.33	0.32	0.01	0.26	0.47
		50	**0.93**	0.70	0.62	0.54	0.55	0.02	0.43	0.66
		100	**1.00**	0.95	0.90	0.83	0.87	0.10	0.73	0.94
		200	**1.00**	**1.00**	**1.00**	0.98	**1.00**	0.40	0.95	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.97	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	Average AST	10	**0.76**	0.10	0.10	0.09	0.02	0.00	0.08	0.10
		30	**0.68**	0.28	0.25	0.21	0.13	0.00	0.17	0.26
		50	**0.72**	0.48	0.41	0.36	0.23	0.01	0.27	0.42
		100	**0.84**	0.70	0.64	0.57	0.52	0.04	0.48	0.66
		200	**0.93**	0.91	0.83	0.79	0.83	0.21	0.71	0.88
		500	**0.99**	**0.99**	0.97	0.96	0.99	0.57	0.92	0.99
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.83	0.99	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(1)	10	**0.83**	0.61	0.61	0.60	0.43	0.18	0.58	0.64
		30	**0.99**	0.96	0.96	0.96	0.94	0.74	0.96	0.97
		50	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.91	0.99	**1.00**
		100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(2)	10	**0.75**	0.29	0.32	0.30	0.18	0.03	0.24	0.34
		30	**0.88**	0.69	0.67	0.68	0.66	0.18	0.58	0.72
		50	**0.96**	0.87	0.85	0.85	0.87	0.36	0.79	0.88
		100	**1.00**	0.98	0.98	0.99	0.99	0.72	0.95	0.99
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.97	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(4)	10	**0.72**	0.13	0.14	0.14	0.06	0.00	0.14	0.17
		30	**0.66**	0.34	0.31	0.27	0.32	0.02	0.22	0.37
		50	**0.78**	0.48	0.43	0.40	0.51	0.04	0.29	0.52
		100	**0.92**	0.71	0.66	0.61	0.76	0.07	0.52	0.79
		200	**0.99**	0.93	0.89	0.86	0.93	0.20	0.75	0.95
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.66	0.98	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.98	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(7)	10	**0.69**	0.08	0.09	0.08	0.04	0.00	0.07	0.10
		30	**0.56**	0.16	0.14	0.14	0.17	0.00	0.12	0.20
		50	**0.61**	0.22	0.19	0.18	0.26	0.00	0.12	0.26
		100	**0.74**	0.38	0.27	0.26	0.45	0.01	0.19	0.42
		200	**0.90**	0.58	0.48	0.40	0.64	0.01	0.29	0.63
		500	**0.98**	0.90	0.82	0.77	0.94	0.04	0.62	0.92
		1000	0.99	**1.00**	0.98	0.96	**1.00**	0.20	0.86	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	tukey(10)	10	**0.74**	0.55	0.61	0.63	0.19	0.04	0.63	0.57
		30	0.91	0.94	0.98	**0.99**	0.54	0.52	0.98	0.96
		50	0.98	**1.00**	**1.00**	**1.00**	0.74	0.87	**1.00**	**1.00**
		100	**1.00**	**1.00**	**1.00**	**1.00**	0.95	**1.00**	**1.00**	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	Average SLT	10	**0.75**	0.33	0.35	0.35	0.18	0.05	0.33	0.36
		30	**0.80**	0.62	0.61	0.61	0.53	0.29	0.57	0.64
		50	**0.87**	0.71	0.69	0.69	0.67	0.44	0.64	0.73
		100	**0.93**	0.81	0.78	0.77	0.83	0.56	0.73	0.84
		200	**0.98**	0.90	0.87	0.85	0.91	0.64	0.81	0.92
		500	**1.00**	0.98	0.96	0.95	0.99	0.74	0.92	0.98
		1000	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.84	0.97	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	beta(1.3,1.3)	10	**0.82**	0.05	0.06	0.06	0.00	0.00	0.04	0.04
		30	**0.84**	0.22	0.18	0.14	0.00	0.00	0.10	0.07
		50	**0.87**	0.45	0.35	0.26	0.00	0.00	0.14	0.24
		100	**0.98**	0.94	0.73	0.57	0.21	0.00	0.36	0.73
		200	**1.00**	**1.00**	0.99	0.94	0.99	0.02	0.76	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.47	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	beta(1.5,1.5)	10	**0.81**	0.04	0.05	0.04	0.00	0.00	0.05	0.04
		30	**0.74**	0.16	0.13	0.10	0.00	0.00	0.06	0.04
		50	**0.80**	0.32	0.26	0.19	0.00	0.00	0.09	0.14
		100	**0.94**	0.81	0.59	0.44	0.08	0.00	0.27	0.54
		200	**1.00**	**1.00**	0.95	0.86	0.95	0.01	0.60	0.99
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.25	0.98	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.93	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	truncatednormal(2,2)	10	**0.69**	0.06	0.05	0.06	0.01	0.00	0.05	0.06
		30	**0.43**	0.05	0.05	0.04	0.02	0.00	0.04	0.06
		50	**0.36**	0.05	0.05	0.05	0.04	0.00	0.05	0.04
		100	**0.35**	0.06	0.05	0.04	0.06	0.00	0.05	0.06
		200	**0.28**	0.05	0.05	0.04	0.04	0.00	0.05	0.04
		500	**0.14**	0.03	0.04	0.06	0.06	0.00	0.05	0.05
		1000	**0.07**	0.05	0.04	0.05	0.05	0.00	0.05	0.05
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	tukey(1.5)	10	**0.86**	0.08	0.09	0.06	0.00	0.00	0.06	0.06
		30	**0.93**	0.46	0.36	0.27	0.00	0.00	0.16	0.22
		50	**0.98**	0.85	0.66	0.55	0.00	0.00	0.33	0.60
		100	**1.00**	**1.00**	0.98	0.90	0.71	0.02	0.68	0.99
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.18	0.98	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.98	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	uniform(0,1)	10	**0.84**	0.08	0.07	0.08	0.00	0.00	0.06	0.05
		30	**0.91**	0.39	0.32	0.24	0.00	0.00	0.15	0.17
		50	**0.96**	0.76	0.56	0.44	0.00	0.00	0.26	0.46
		100	**1.00**	**1.00**	0.95	0.85	0.57	0.01	0.57	0.96
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.13	0.95	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.92	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	Average SST	10	**0.80**	0.06	0.06	0.06	0.00	0.00	0.05	0.05
		30	**0.77**	0.26	0.21	0.16	0.00	0.00	0.10	0.11
		50	**0.79**	0.49	0.38	0.30	0.01	0.00	0.17	0.30
		100	**0.85**	0.76	0.66	0.56	0.33	0.01	0.39	0.66
		200	**0.86**	0.81	0.80	0.77	0.80	0.07	0.67	0.81
		500	**0.83**	0.81	0.81	0.81	0.81	0.52	0.81	0.81
		1000	**0.81**	**0.81**	**0.81**	**0.81**	**0.81**	**0.78**	**0.81**	**0.81**

Table 25: Tests power per distribution at 10% significance level

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	laplace(0,10)	10	**0.84**	0.21	0.23	0.24	0.10	0.00	0.22	0.31
		30	**0.82**	0.45	0.49	0.46	0.39	0.05	0.42	0.53
		50	**0.89**	0.60	0.63	0.61	0.56	0.07	0.55	0.70
		100	**0.96**	0.87	0.89	0.88	0.79	0.21	0.81	0.90
		200	**1.00**	0.99	0.99	**1.00**	0.97	0.55	0.97	0.99
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	t(10)	10	**0.86**	0.12	0.13	0.12	0.03	0.00	0.12	0.14
		30	**0.74**	0.17	0.16	0.16	0.15	0.00	0.13	0.22
		50	**0.74**	0.24	0.18	0.17	0.25	0.00	0.15	0.28
		100	**0.75**	0.28	0.25	0.22	0.34	0.01	0.18	0.38
		200	**0.76**	0.45	0.38	0.31	0.53	0.01	0.26	0.53
		500	**0.91**	0.72	0.59	0.58	0.80	0.03	0.41	0.80
		1000	**0.96**	0.94	0.85	0.80	**0.96**	0.07	0.67	0.95
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	tukey(0.1)	10	**0.87**	0.09	0.11	0.11	0.02	0.00	0.10	0.11
		30	**0.67**	0.12	0.10	0.11	0.08	0.00	0.10	0.13
		50	**0.62**	0.10	0.13	0.10	0.12	0.00	0.11	0.14
		100	**0.60**	0.14	0.14	0.11	0.11	0.00	0.12	0.15
		200	**0.53**	0.15	0.15	0.14	0.15	0.00	0.11	0.18
		500	**0.42**	0.20	0.17	0.16	0.25	0.00	0.15	0.25
		1000	0.26	0.28	0.24	0.24	**0.33**	0.00	0.19	0.31
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	tukey(0.2)	10	**0.90**	0.10	0.10	0.09	0.01	0.00	0.09	0.10
		30	**0.66**	0.08	0.10	0.09	0.02	0.00	0.11	0.06
		50	**0.60**	0.09	0.10	0.09	0.02	0.00	0.11	0.07
		100	**0.55**	0.08	0.10	0.13	0.02	0.00	0.11	0.05
		200	**0.42**	0.11	0.11	0.12	0.04	0.00	0.12	0.07
		500	**0.45**	0.22	0.18	0.17	0.19	0.00	0.13	0.13
		1000	**0.71**	0.44	0.33	0.24	0.46	0.00	0.19	0.29
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	tukey(5)	10	**0.85**	0.13	0.16	0.16	0.02	0.00	0.17	0.15
		30	**0.66**	0.16	0.25	0.27	0.02	0.01	0.23	0.18
		50	**0.57**	0.25	0.38	0.40	0.02	0.01	0.32	0.21
		100	0.57	0.59	0.63	**0.65**	0.00	0.05	0.60	0.43
		200	0.65	**0.96**	0.94	0.92	0.00	0.20	0.88	0.89

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500	0.83	**1.00**	**1.00**	**1.00**	0.00	0.85	**1.00**	**1.00**
		1000	0.98	**1.00**	**1.00**	**1.00**	**0.01**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
CTN	Average CTN	10	**0.86**	0.13	0.15	0.14	0.04	0.00	0.14	0.16
		30	**0.71**	0.20	0.22	0.22	0.13	0.01	0.20	0.22
		50	**0.68**	0.26	0.28	0.27	0.19	0.02	0.25	0.28
		100	**0.69**	0.39	0.40	0.40	0.25	0.05	0.36	0.38
		200	**0.67**	0.53	0.51	0.50	0.34	0.15	0.47	0.53
		500	**0.72**	0.63	0.59	0.58	0.45	0.37	0.54	0.64
		1000	**0.78**	0.73	0.68	0.66	0.55	0.41	0.61	0.71
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	chisquared(10)	10	**0.94**	0.21	0.18	0.15	0.05	0.00	0.18	0.20
		30	**0.94**	0.46	0.41	0.39	0.28	0.02	0.31	0.45
		50	**0.96**	0.72	0.62	0.55	0.51	0.05	0.46	0.69
		100	**1.00**	0.95	0.90	0.84	0.86	0.12	0.74	0.93
		200	**1.00**	**1.00**	0.99	0.98	**1.00**	0.42	0.94	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.96	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	chisquared(4)	10	**0.98**	0.35	0.29	0.30	0.09	0.01	0.26	0.36
		30	**0.97**	0.82	0.80	0.71	0.58	0.07	0.59	0.80
		50	**1.00**	0.98	0.94	0.90	0.84	0.19	0.80	0.97
		100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.55	0.98	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.96	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	lognormal(0,1)	10	**0.99**	0.72	0.70	0.67	0.36	0.07	0.56	0.70
		30	**1.00**	**1.00**	0.99	0.99	0.95	0.59	0.96	1.00
		50	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
		100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	Weibull(0.5,1)	10	**1.00**	0.95	0.91	0.90	0.52	0.24	0.86	0.93
		30	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.96	**1.00**	**1.00**
		50	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	Weibull(2,1)	10	**0.95**	0.16	0.16	0.14	0.02	0.00	0.13	0.15
		30	**0.90**	0.36	0.28	0.26	0.16	0.01	0.23	0.33
		50	**0.92**	0.55	0.44	0.38	0.29	0.02	0.30	0.53
		100	**0.99**	0.88	0.76	0.65	0.68	0.05	0.53	0.83
		200	**1.00**	**1.00**	0.97	0.90	0.97	0.18	0.81	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.75	0.99	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
ALT	Average ALT	10	**0.97**	0.48	0.45	0.43	0.21	0.06	0.40	0.47
		30	**0.96**	0.73	0.70	0.67	0.59	0.33	0.62	0.72
		50	**0.98**	0.85	0.80	0.77	0.73	0.43	0.71	0.84
		100	**1.00**	0.97	0.93	0.90	0.91	0.54	0.85	0.95
		200	**1.00**	**1.00**	0.99	0.98	0.99	0.71	0.95	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.94	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	beta(2,1)	10	**0.89**	0.23	0.24	0.22	0.02	0.00	0.15	0.20
		30	**0.84**	0.68	0.61	0.50	0.08	0.02	0.40	0.55
		50	0.87	**0.94**	0.83	0.74	0.26	0.05	0.61	0.83
		100	0.93	**1.00**	**1.00**	0.97	0.95	0.25	0.91	**1.00**
		200	0.97	**1.00**	**1.00**	**1.00**	**1.00**	0.73	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	beta(3,2)	10	**0.88**	0.11	0.10	0.10	0.00	0.00	0.09	0.10
		30	**0.78**	0.19	0.20	0.17	0.02	0.00	0.14	0.14
		50	**0.74**	0.36	0.26	0.23	0.02	0.00	0.19	0.20
		100	**0.85**	0.71	0.58	0.46	0.21	0.01	0.40	0.54
		200	0.92	**0.99**	0.90	0.78	0.88	0.05	0.65	0.95
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.40	0.98	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.94	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	lognormal(0,0.15)	10	**0.92**	0.14	0.14	0.11	0.03	0.00	0.12	0.14
		30	**0.81**	0.18	0.18	0.17	0.14	0.00	0.15	0.20
		50	**0.79**	0.26	0.23	0.22	0.22	0.01	0.21	0.26
		100	**0.86**	0.47	0.36	0.34	0.37	0.01	0.28	0.45
		200	**0.93**	0.71	0.60	0.57	0.67	0.04	0.50	0.70

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500	**0.98**	**0.98**	0.92	0.90	0.97	0.20	0.82	0.97
		1000	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.57	0.98	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	lognormal(0,0.25)	10	**0.92**	0.18	0.18	0.17	0.05	0.00	0.13	0.16
		30	**0.86**	0.39	0.32	0.29	0.26	0.01	0.26	0.36
		50	**0.90**	0.53	0.49	0.44	0.41	0.03	0.36	0.50
		100	**0.99**	0.83	0.73	0.68	0.75	0.06	0.59	0.80
		200	**1.00**	0.98	0.95	0.92	0.98	0.24	0.84	0.97
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.79	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	lognormal(0,0.35)	10	**0.96**	0.21	0.25	0.20	0.06	0.00	0.17	0.22
		30	**0.94**	0.58	0.52	0.48	0.39	0.03	0.41	0.58
		50	**0.98**	0.78	0.71	0.65	0.66	0.07	0.52	0.76
		100	**1.00**	0.97	0.94	0.91	0.93	0.21	0.82	0.96
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.60	0.98	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
AST	Average AST	10	**0.91**	0.17	0.18	0.16	0.03	0.00	0.13	0.16
		30	**0.85**	0.40	0.37	0.32	0.18	0.01	0.27	0.37
		50	**0.86**	0.57	0.50	0.46	0.31	0.03	0.38	0.51
		100	**0.93**	0.80	0.72	0.67	0.64	0.11	0.60	0.75
		200	**0.96**	0.94	0.89	0.85	0.91	0.33	0.79	0.92
		500	**1.00**	**1.00**	0.98	0.98	0.99	0.68	0.96	0.99
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(1)	10	**0.93**	0.64	0.67	0.66	0.46	0.26	0.66	0.66
		30	**0.99**	0.97	0.98	0.98	0.95	0.79	0.96	0.98
		50	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.97	**1.00**	**1.00**
		100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(2)	10	**0.88**	0.34	0.36	0.34	0.22	0.04	0.33	0.41
		30	**0.93**	0.74	0.75	0.72	0.70	0.28	0.68	0.78
		50	**0.99**	0.89	0.90	0.89	0.90	0.51	0.83	0.91
		100	**1.00**	0.99	0.99	0.99	0.99	0.79	0.98	0.99
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.98	**1.00**	**1.00**

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(4)	10	**0.87**	0.20	0.22	0.19	0.08	0.00	0.20	0.25
		30	**0.84**	0.40	0.39	0.35	0.37	0.03	0.31	0.46
		50	**0.87**	0.57	0.52	0.47	0.57	0.07	0.42	0.63
		100	**0.96**	0.76	0.75	0.72	0.82	0.13	0.63	0.83
		200	**0.99**	0.94	0.93	0.90	0.96	0.33	0.86	0.96
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.84	0.99	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	t(7)	10	**0.88**	0.14	0.13	0.13	0.05	0.00	0.13	0.18
		30	**0.78**	0.24	0.22	0.20	0.21	0.01	0.17	0.30
		50	**0.77**	0.32	0.26	0.22	0.29	0.01	0.20	0.39
		100	**0.85**	0.43	0.41	0.34	0.50	0.02	0.28	0.54
		200	**0.90**	0.68	0.59	0.54	0.73	0.03	0.43	0.72
		500	**0.99**	0.93	0.90	0.83	0.96	0.12	0.75	0.95
		1000	**1.00**	**1.00**	0.99	0.98	**1.00**	0.37	0.95	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	tukey(10)	10	**0.89**	0.61	0.75	0.73	0.24	0.12	0.75	0.70
		30	0.96	0.96	**0.99**	0.99	0.61	0.72	0.98	0.98
		50	0.99	**1.00**	**1.00**	**1.00**	0.84	0.96	**1.00**	**1.00**
		100	**1.00**	**1.00**	**1.00**	**1.00**	0.97	**1.00**	**1.00**	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SLT	Average SLT	10	**0.89**	0.39	0.43	0.41	0.21	0.08	0.41	0.44
		30	**0.90**	0.66	0.67	0.65	0.57	0.37	0.62	0.70
		50	**0.92**	0.76	0.74	0.72	0.72	0.50	0.69	0.79
		100	**0.96**	0.84	0.83	0.81	0.86	0.59	0.78	0.87
		200	**0.98**	0.92	0.90	0.89	0.94	0.67	0.86	0.94
		500	**1.00**	0.99	0.98	0.97	0.99	0.79	0.95	0.99
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.87	0.99	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	beta(1.3,1.3)	10	**0.94**	0.14	0.12	0.13	0.01	0.00	0.11	0.07
		30	**0.93**	0.36	0.33	0.24	0.00	0.00	0.18	0.21
		50	**0.93**	0.66	0.50	0.37	0.00	0.01	0.30	0.37
		100	**0.98**	**0.98**	0.86	0.72	0.64	0.02	0.55	0.88
		200	**1.00**	**1.00**	**1.00**	0.98	**1.00**	0.12	0.87	**1.00**

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.79	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	beta(1.5,1.5)	10	**0.94**	0.11	0.11	0.10	0.01	0.00	0.09	0.08
		30	**0.89**	0.30	0.23	0.21	0.00	0.00	0.17	0.14
		50	**0.88**	0.50	0.38	0.33	0.00	0.00	0.22	0.28
		100	**0.97**	0.89	0.74	0.60	0.42	0.01	0.40	0.75
		200	**1.00**	**1.00**	0.98	0.93	0.99	0.05	0.78	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.56	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	truncatednormal(2,2)	10	**0.88**	0.11	0.09	0.10	0.01	0.00	0.08	0.11
		30	**0.68**	0.11	0.10	0.12	0.05	0.00	0.10	0.11
		50	**0.60**	0.11	0.09	0.09	0.06	0.00	0.10	0.10
		100	**0.55**	0.11	0.11	0.12	0.07	0.00	0.11	0.10
		200	**0.40**	0.11	0.10	0.11	0.08	0.00	0.10	0.09
		500	**0.22**	0.09	0.07	0.09	0.08	0.00	0.09	0.09
		1000	0.09	0.08	0.09	0.10	**0.11**	0.00	**0.11**	0.12
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	tukey(1.5)	10	**0.95**	0.21	0.17	0.18	0.00	0.00	0.13	0.12
		30	**0.98**	0.69	0.53	0.46	0.00	0.00	0.32	0.43
		50	**1.00**	0.93	0.80	0.68	0.05	0.01	0.48	0.78
		100	**1.00**	**1.00**	0.99	0.96	0.96	0.07	0.82	**1.00**
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.45	0.99	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	uniform(0,1)	10	**0.95**	0.17	0.16	0.14	0.00	0.00	0.12	0.12
		30	**0.96**	0.59	0.48	0.36	0.00	0.00	0.27	0.32
		50	**0.98**	0.88	0.72	0.61	0.04	0.01	0.41	0.68
		100	**1.00**	**1.00**	0.98	0.91	0.91	0.04	0.75	0.99
		200	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.30	0.98	**1.00**
		500	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**
		1000	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
		Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
SST	Average SST	10	**0.93**	0.15	0.13	0.13	0.01	0.00	0.11	0.10
		30	**0.89**	0.41	0.33	0.28	0.01	0.00	0.21	0.24
		50	**0.88**	0.62	0.50	0.42	0.03	0.01	0.30	0.44
		100	**0.90**	0.80	0.74	0.66	0.60	0.03	0.53	0.74
		200	**0.88**	0.82	0.82	0.80	0.81	0.18	0.74	0.82

Family	Distribution	Size	new_test	SW	AD	CVM	JB	KS	Lillie	SF
		500	**0.84**	0.82	0.81	0.82	0.82	0.67	0.82	0.82
		1000	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	0.80	**0.82**	**0.82**