Journal of Modern Applied Statistical Methods

Volume 16 | Issue 2 Article 30

December 2017

# JMASM 49: A Compilation of Some Popular Goodness of Fit Tests for Normal Distribution: Their Algorithms and MATLAB Codes (MATLAB)

Metin Öner
*Manisa Celal Bayar University, Manisa, Turkey*, metin.oner@cbu.edu.tr

İpek Deveci Kocakoç
*Dokuz Eylul University, İzmir, Turkey*, ipek.deveci@deu.edu.tr

Journal of Modern Applied Statistical Methods
November 2017, Vol. 16, No. 2, 547-575.
doi: 10.22237/jmasm/1509496200

ISSN 1538 − 9472

# JMASM 49: A Compilation of Some Popular Goodness of Fit Tests for Normal Distribution: Their Algorithms and MATLAB Codes (MATLAB)

**Metin Öner**
Manisa Celal Bayar University
Manisa, Turkey

**İpek Deveci Kocakoç**
Dokuz Eylul University
İzmir, Turkey

The main purpose of this study is to review calculation algorithms for some of the most common non-parametric and omnibus tests for normality, and to provide them as a compiled MATLAB function. All tests are coded to provide *p*-values for those normality tests, and the proposed function gives the results as an output table.

*Keywords:* Normality test, non-parametric test, MATLAB function, *p*-value

## Introduction

One of the most important assumptions for parametric statistical methods is that the sample data come from a normally-distributed population. As this assumption holds, *t*-tests, variance analysis, factor analysis, and many more methods gain power. Normality of error terms is one of the most important assumptions for regression analysis. For many statistical analyses, a normality test is one of the most important and necessary things to do along with the examination of distributional features of data with descriptive statistics, outlier detection, and heteroscedasticity tests.

Hypotheses for fitness testing if sample data or variable(s) follow a normal distribution (i.e. goodness-of-fit testing) are as follows:

$$H_0: x \in N(\mu, \sigma)$$
$$H_1: x \notin N(\mu, \sigma)$$

Most popular normality tests can be listed as follows:

*Metin Öner is an Associate Professor in the School of Applied Sciences. Email him at: metin.oner@cbu.edu.tr.*

(a) Empirical Distribution Function (EDF) tests:
- Kolmogorov-Smirnov test
    - Limiting form (KS-Lim)
    - Stephens Method (KS-S)
    - Marsaglia Method (KS-M)
    - Lilliefors test (KS-L)
- Anderson-Darling (AD) test
- Cramer-Von Mises (CvM) test

(b) Tests based on Regression and Correlation:
- Shapiro-Wilk (SW) test
- Shapiro-Francia (SF) test

(c) Moment $\left(\sqrt{b_1}, b_2\right)$ tests:
- Jarque-Bera (JB) test
- D'Agostino and Pearson (DAP) test

Shown in Table 1, normality tests covered by some commercial and non-commercial statistical packages/software. Only some common tests are computed by these packages, and researchers are obliged to use these ready-made tests, or code them in any language. Some tests are not covered by any of the packages, and most packages do not give any hints on which test to use under which assumptions.

Therefore, the purpose of this article is to review the most common algorithms used for the computation of $p$-values for normality testing and compile them in one MATLAB routine for researchers who want to examine their data via different normality tests.

**Table 1.** Normality tests covered by some commercial and non-commercial statistical software

| | EDF Based | | | | | | Reg-Corr | | Moment | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Package** | **KS-Lim** | **KS-S** | **KS-M** | **KS-L** | **AD** | **CvM** | **SW** | **SF** | **JB** | **DAP** |
| Minitab | X | | | | X | | | | | |
| NCSS | X | | | | X | | | | | |
| SAS | X | | | | | | X | | | |
| SPSS | X | | | X | | | X | | | |
| Stata | | | | | | | X | X | | X |
| Statistica | X | | | X | | | X | | | |
| XL-Stat | | | | X | X | | X | | X | |
| R Project | | | | X | X | X | X | X | | X |
| Analyse-it | X | | | | X | | X | | | |

## Empirical Distribution Function Tests

Let $y_1, y_2, \ldots, y_n$ be ordered values of $x_1, x_2, \ldots, x_n$ sample values. If $i$ denotes the frequency of $y_k$ in $k^{th}$ order, the empirical distribution function is defined as the following step function:

$$\mathrm{F}_n(y) = \begin{cases} 0, & y < y_1 \\ i/n, & y_k \le y < y_{k+1}, \quad k = 1, 2, \ldots n-1 \\ 1, & y \ge y_n \end{cases} \tag{1}$$

The following EDF tests are based on this function. Computation procedures are given here for these tests. For further information about the formulas and the interpretation of EDF statistics, see Hollander and Wolfe (1999) and Gibbons and Chakraborti (1992). For details about the $k$-sample analogs of the Kolmogorov-Smirnov and Cramer-von Mises statistics used by NPAR1WAY, see Kiefer (1959).

### Kolmogorov-Smirnov Normality Test

The Kolmogorov-Smirnov (KS) test statistic is computed with the help of the $D_n$ statistic, which is defined as follows:

$$D_n = \sup_x \left| \mathrm{F}_n(x) - \mathrm{F}_0(x) \right| \tag{2}$$

where "sup" in equation (2) denotes the supremum, that is the maximum of the values in a given interval.

The $D_n$ test statistic will be the greatest vertical distance between $\mathrm{F}(x)$ and $\mathrm{F}_0(x)$ (Kolmogorov, 1933):

$$D_n = \max\left\{ \left| D_n^- \right|, \left| D_n^+ \right| \right\} \tag{3}$$

where $D_n^- = \mathrm{F}(x_{k-1}) - \mathrm{F}_0(x_k)$ and $D_n^+ = \mathrm{F}(x_k) - \mathrm{F}_0(x_k)$. The KS$z$ statistic is computed as follows:

$$\mathrm{KS}z = \sqrt{n} D_n \tag{4}$$

The stepwise procedure given below covers the mutual steps required for calculation of *p*-values for four KS-type tests. The first eight steps are also identical for the AD and CvM tests. The rest of the calculation steps are given for each EDF test.

Step 1: Enter sample **x** vector

Step 2: $y_i = \text{sort}(\mathbf{x})$

Step 3: $i = 1, 2, \ldots, n$

Step 4: $F_n(\mathbf{y}) = i / n$

Step 5: $\text{mean}(\mathbf{y}) = \Sigma\, y_i / n$

Step 6: $\sigma(\mathbf{y}) = \sqrt{\sum \left(y_i - \text{mean}(\mathbf{y})\right)^2 / n - 1}$

Step 7: $z_i = (y_i - \text{mean}(\mathbf{y})) / \sigma(\mathbf{y})$

Step 8: $F_0(\mathbf{y}) = u_i = \phi(z_i)$

Step 9: $D_{\text{plus}} = \text{abs}(F_0(\mathbf{y}) - i / n)$

Step 10: $D_{\text{minus}} = \text{abs}(F_0(\mathbf{y}) - (i - 1) / n)$

Step 11: $D_n = \max(D_{\text{plus}}, D_{\text{minus}})$

Step 12: $\text{KS}z = \sqrt{n} D_n$

***Limiting Form***

Most statistical packages use this method to calculate the KS*z* test statistic. In this method, if $n \to \infty$ the distribution of the KS*z* statistic $\left(\sqrt{n} D_n\right)$ is asymptotically Kolmogorov distributed. This statistic has the following formula (Facchinetti, 2009):

$$\lim_{n\to\infty} \Pr\left(\sqrt{n} D_n \le x\right) = 1 - 2\sum_{k=1}^{\infty} (-1)^{k-1} e^{-2k^2 x^2} \tag{5}$$

This method is suitable for cases where the sample size is large and the distribution parameters are known. However, it is being used in cases where *n* is small and parameters are not known. The *p*-value for this test can be calculated by the following step:

$$\text{Step 13: } p\text{-value} = 1 - \sum_{k=-20}^{20} (-1)^k \exp\left(-2k^2 \left(\sqrt{n} D_n\right)^2\right)$$

***Marsaglia Method***

This method was introduced by Marsaglia, Tsang, and Wang (2003). $\Pr(D_n \le d)$ is calculated by this formula:

$$\Pr(D_n \le d) = \frac{n!}{n^n} t_{kk} \tag{6}$$

Here, $t_{kk}$ is the $(k, k)^{th}$ element of the matrix $\mathbf{H}^n$, $\mathbf{H}$ is an $m \times m$ matrix, $m = 2k - 1$, $d = (k - h) / n$, with $k$ (a positive integer), and $0 \le h < 1$.

Although this method has a complicated algorithm, it provides 13-15 digit accuracy for $n$ ranging from 2 to at least 16000 for one-tailed $p$-value calculation.

Step 13: Calculate $k$, $m$, and $h$ values:
$k$ = Round up $(nD_n)$
$m = 2k - 1$
$h = k - nD_n$

Step 14: Get the **H** matrix using the following procedure:
Get the first column of the **H** matrix except the element Hmatrix $(m, 1)$.

```
Loop i = 1: m – 1
  Hmatrix (i, 1) = (1 – h^i) / i!
End
```

Get the $m^{th}$ row of the **H** matrix except the element Hmatrix $(m, 1)$.
Hmatrix $(m, c)$ = Hmatrix $(r, 1)^T$, $r = 1, \ldots, m - 1$, and $c = m - r + 1$
Get the other elements of the **H** matrix.

```
Loop i = 1: m – 1
  Loop j = 2: m
      If i – j + 1 ≥ 0
          Hmatrix (i, j) = 1 / (i – j + 1)!
      Else
          Hmatrix (i, j) = 0
      End
  End
End
```

Get the element Hmatrix $(m, 1)$.

```
  If h ≤ 0.5
```

Hmatrix ($m$, 1) = $(1 – 2h^m) / m!$

Else

Hmatrix ($m$, 1) = $(1 – 2h^m + \max(0, 2h – 1)^m) / m!$

Step 15: Calculate the $p$-value

$$p\text{-value} = \Pr\left(D_n > d\right) = 1 - \Pr\left(D_n \le d\right) = 1 - \frac{n!}{n^n}\mathbf{H}^n\left(k,k\right)$$

### *Stephens' Method*

Stephens' method uses a $D^*$ test statistic, which is revised by using $D_n$, to test normality for cases where parameters are not known. The test statistic $D^*$ is calculated based on $n$ via the following equation:

$$D^* = D_n\left(\sqrt{n} - 0.01 + \frac{0.85}{\sqrt{n}}\right) \tag{7}$$

Stephens (1986, p. 123) obtained upper tail critical $D^*$ values by Monte Carlo simulations and tabulated them. Here, the $p$-value is calculated by linear interpolation based on the values of Stephens' table and the calculation steps are as follows:

Step 13: Calculate the modified $D^*$ statistic from equation (7).

Step 14: Calculate the $p$-value via linear interpolation by using Stephens' critical value table:

$$p\text{-value} = \begin{cases} \text{report } p\text{-value} > 0.15 & D^* < 0.775 \\ 0.15 + \left(D^* - 0.775\right)\left[\left(0.10 - 0.15\right)/\left(0.819 - 0.775\right)\right] & 0.775 \le D^* < 0.819 \\ 0.10 + \left(D^* - 0.819\right)\left[\left(0.05 - 0.10\right)/\left(0.895 - 0.819\right)\right] & 0.819 \le D^* < 0.895 \\ 0.05 + \left(D^* - 0.895\right)\left[\left(0.025 - 0.05\right)/\left(0.995 - 0.895\right)\right] & 0.895 \le D^* < 0.995 \\ 0.025 + \left(D^* - 0.995\right)\left[\left(0.01 - 0.025\right)/\left(1.035 - 0.995\right)\right] & 0.995 \le D^* < 1.035 \\ \text{report } p\text{-value} < 0.01 & D^* \ge 1.035 \end{cases}$$

***Lilliefors Test for Normality***

The Lilliefors (LF) test was presented as a correction of the Kolmogorov-Smirnov test by Lilliefors (1967). Dellal and Wilkinson (1986) provided a numerical approximation to calculate $p$-values for this method. The LF test is an extension of the Kolmogorov-Smirnov test to the case where the parameters of the hypothesized normal distribution are unknown and estimated by a sample data set. If the mean and variance parameters of the hypothesized normal distribution are not known, the LF test or Stephens' method should be used instead of the Kolmogorov-Smirnov test. The LF test as a correction of the Kolmogorov-Smirnov test should not be confused with the original Kolmogorov-Smirnov test. This study adopts the algorithm used by the statistical software SPSS to calculate the $p$-value of the LF test, which is based on the use of the critical value table and formulation of Dellal and Wilkinson. The algorithm is given below:

Step 13: Find the critical values $D20$ and $D15$ corresponding to $n$ from the table given by Dellal and Wilkinson (1986). This table is included in the MATLAB codes in the Appendix. $D20$ and $D15$ are the critical values corresponding to $\alpha = 0.20$ and $\alpha = 0.15$ for the LF normality test, respectively. If $n$ lies between two lower and upper $n$ values, find the critical values $D20$ and $D15$ by linear interpolation. The equations of linear interpolation for the critical values $D20$ and $D15$ are given below:

$$D20 = D20_{\text{lower}} + \left(n - n_{\text{lower}}\right)\frac{D20_{\text{upper}} - D20_{\text{lower}}}{n_{\text{upper}} - n_{\text{lower}}}$$

$$D15 = D15_{\text{lower}} + \left(n - n_{\text{lower}}\right)\frac{D15_{\text{upper}} - D15_{\text{lower}}}{n_{\text{upper}} - n_{\text{lower}}}$$

Step 14: Find the values of $a_1$, $b_1$, $c_1$, $a_2$, $b_2$, and $c_2$ via the following equations:

$$a_1 = -7.01256(n + 2.78019)$$
$$b_1 = 2.99587\sqrt{n + 2.78019}$$
$$c_1 = 2.1804661 + 0.974598/\sqrt{n} + 1.67997/n$$
$$a_2 = -7.90289126054n^{0.98}$$
$$b_2 = 3.180370175721n^{0.49}$$
$$c_2 = 2.2947256$$

Step 15: Calculate the critical value $D10$ corresponding to $\alpha = 0.10$.
If $n \leq 100$, $a = a_1$, $b = b_1$, $c = c_1$, with

$$D10 = \frac{-b_1 - \sqrt{b_1^{\,2} - 4a_1c_1}}{2a_1}$$

If $n > 100$, $a = a_2$, $b = b_2$, $c = c_2$, with

$$D10 = \frac{-b_2 - \sqrt{b_2^{\,2} - 4a_2c_2}}{2a_2}$$

Step 16: Calculate the $p$-value according to the following formula:

$$p\text{-value} = \begin{cases} 0.10 & \text{if} \quad D_n = D10 \\ \exp\left(aD_n^2 + bD_n + c - 2.3025851\right) & \text{if} \quad D_n > D10 \\ 0.15 + \left(D_n - D15\right)\left[\left(0.10 - 0.15\right)/\left(D10 - D15\right)\right] & \text{if} \quad D_n \geq D15 \\ 0.20 + \left(D_n - D20\right)\left[\left(0.15 - 0.20\right)/\left(D15 - D20\right)\right] & \text{if} \quad D_n \geq D20 \\ \text{report } p\text{-value} > 0.20 & \text{if} \quad D_n \leq D10 \end{cases}$$

## Anderson-Darling Normality Test

The Anderson-Darling (AD) statistic, $A^2$, was derived from following integral function:

$$A^2 = n\int_{-\infty}^{+\infty}\left[\mathrm{F}_n(x) - \mathrm{F}(x)\right]^2 \psi(x)\, d\mathrm{F}(x) \tag{8}$$

where ψ($x$) is the weight function of the squared difference. The AD test statistic turns into following formula by taking the weight function to be $\psi(x) = [F(x)(1 - F(x)]^{-1}$, where F($x$) is the underlying theoretical cumulative distribution (Anderson & Darling, 1952, 1954).

$$A^2 = -n - (1/n)\sum_{i=1}^{n}(2i-1)\left[\ln(u_i) + \ln(1-u_{n-i+1})\right] \qquad (9)$$

where $u_i = \varphi(z_i) = \varphi\left((y_i - \bar{x})/\hat{\sigma}\right)$ is the cumulative probability of the standard normal distribution. Mean and variance are unknown and estimated from the sample.

Stephens (1986) modified the $A^2$ test statistic to obtain critical values for different sample sizes:

$$A^{*2} = A^2\left(1.0 + 0.75/n + 2.25/n^2\right) \qquad (10)$$

The algorithm for calculating the $p$-value is given below for Anderson-Darling test (Stephens, 1986, p. 127).

Step 9: Loop $i$ = 1: $n$
Sum = Sum + (2$i$ – 1)(ln $u_i$ + ln(1 – $u_i$)
End
Step 10: Calculate the test statistic.
$A$ = -$n$ – (1 / $n$) ∗ Sum
Step 11: Calculate the modified test statistic.
$A^*$ = $A$(1 + 0.75 / $n$ + 2.25 / $n^2$)
Step 12: Calculate the $p$-value.

$$p\text{-value} = \begin{cases} 1-\exp\left(-13.436+101.14A^* - 223.73A^{*2}\right) & \text{if} \quad A^* \le 0.2 \\ 1-\exp\left(-8.318+42.796A^* - 59.938A^{*2}\right) & \text{if} \quad 0.2 < A^* \le 0.34 \\ \exp\left(0.9177 - 4.279A^* - 1.38A^{*2}\right) & \text{if} \quad 0.34 < A^* \le 0.60 \\ \exp\left(1.2937 - 5.709A^* + 0.0186A^{*2}\right) & \text{if} \quad 0.60 < A^* \le 153.467 \\ 0 & \text{if} \quad A^* > 153.467 \end{cases}$$

## Cramer-von Mises Test

Csörgö and Faraway (1996) noted the Cramer-von Mises (CvM) test was independently presented by Cramer (1928) and von Mises (1931). The CvM test is derived from the same integral function as the AD statistic:

$$W^2 = n\int_{-\infty}^{+\infty}\left[\mathrm{F}_n(x)-\mathrm{F}(x)\right]^2\psi(x)\,d\mathrm{F}(x) \tag{11}$$

where $\psi(x)$ is a weight function of squared differences. When $\psi(x) = 1$, the statistic is referred to as the CvM statistic $W^2$ and, when $\psi(x) = [\mathrm{F}(x)(1 - \mathrm{F}(x)]^{-1}$, the statistic is referred to as the AD statistic $A^2$.

The CvM test statistic $W^2$ can be written explicitly as

$$W^2 = \frac{1}{12n} + \sum_{i=1}^{n}\left[\mathrm{F}_0(y_i) - \frac{2i-1}{2n}\right]^2 \tag{12}$$

where $\mathrm{F}_0(y_i)$ is the cumulative distribution function of the specified distribution and the $y_i$ are the sorted values of the $x_i$ data set (Scott & Stewart, 2011).

Stephens (1986, p.127) provided the calculation procedure for a modified statistic $W^{*2} = W^2(1 + 0.5 / n)$ as follows:

Step 9: Loop $i = 1: n$
Sum = Sum + $(u_i - (2i - 1) / 2n)^2$
End
Step 10: Calculate the test statistic.
$W = (1 / 12n)$ + Sum
Step 11: Calculate the modified test statistic.
$W^* = W(1 + 0.5 / n)$
Step 12: Calculate the $p$-value

$$p\text{-value} = \begin{cases} 1-\exp\left(13.953+775.5W^{*}-12542.61W^{*2}\right) & \text{if} \quad W^{*} < 0.0275 \\ 1-\exp\left(-5.903+179.546W^{*}-1515.29W^{*2}\right) & \text{if} \quad 0.0275 \le W^{*} < 0.051 \\ \exp\left(0.886-31.62W^{*}+10.897W^{*2}\right) & \text{if} \quad 0.051 \le W^{*} < 0.092 \\ 1.111-34.242W^{*}+12.832W^{*2} & \text{if} \quad W^{*} \ge 0.093 \end{cases}$$

## Regression and Correlation Based Tests

### Shapiro-Wilk Test

Let $y_1, y_2, \ldots, y_i, \ldots, y_n$ be ordered values of $n$ independent and identically distributed random samples $(x_1, x_2, \ldots, x_i, \ldots, x_n)$ coming from a population with unknown mean $\mu \in \Re$ and unknown $\sigma > 0$. The Shapiro-Wilk statistic $W$ for testing normality is then defined as

$$W = \frac{\left(\sum_{i=1}^{n} a_i y_i\right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2} \tag{13}$$

where $a_i$ are elements of the vector

$$\mathbf{a}' = \frac{\mathbf{m}'\mathbf{V}^{-1}}{\left(\mathbf{m}'\mathbf{V}^{-1}\mathbf{V}^{-1}\mathbf{m}\right)^{1/2}}$$

where $\mathbf{m}' = (m_1, m_2, \ldots, m_n)$ is the vector of expected values of normal order statistics and $\mathbf{V} = [\mathrm{cov}(y_i, y_j)]$ is the covariance matrix of order statistics (Shapiro & Wilk, 1965).

Royston (1982) provided an approximation method for the calculation of the $p$-value for a normalized $W$ statistic. This method calculates the $p$-value of the test as the upper tail of the standard normal distribution. The algorithm is given below:

Step 1: Sort the sample observations in ascending order, i.e. $\vec{\mathbf{x}} = \left(x_1, \ldots, x_n\right)$ into $\vec{\mathbf{y}} = \left(y_1, \ldots, y_n\right)$.

Step 2: Calculate Blom scores $\left(\tilde{\mathbf{m}}\right)$ (Solomon & Sawilowsky, 2009).

$$\tilde{m}_i = \varphi^{-1}\left(\frac{i - 0.375}{n + 0.25}\right)$$

Step 3: $m = \tilde{\mathbf{m}}^{\mathrm{T}}\tilde{\mathbf{m}} = \sum_{i=1}^{n} \tilde{m}_i^2$

Step 4: Calculate the coefficients $a_i$ from the following equations:
For $i = 1, 2, n - 1, n$, and with $u = 1/\sqrt{n}$:

$$a_n = -2.70606u^5 + 4.43469u^4 - 2.07119u^3 - 0.14798u^2 + 0.22116u + \tilde{m}_n \big/ \sqrt{m}$$
$$a_{n-1} = -3.58263u^5 + 5.68263u^4 - 1.75246u^3 - 0.29376u^2 + 0.04298u + \tilde{m}_n \big/ \sqrt{m}$$
$$a_1 = -a_n$$
$$a_2 = -a_{n-1}$$

For $3 \le \underline{i} \le n - 2$:

$$a_i = \tilde{m}_i \big/ \sqrt{\varepsilon}, \text{ where } \varepsilon = \frac{m - 2\tilde{m}_n^2 - 2\tilde{m}_{n-1}^2}{1 - 2a_n^2 - 2a_{n-1}^2}$$

Step 5: Calculate the Shapiro-Wilk test statistic $W$ from equation (13).

Step 6: Normalize the test statistic $W$:

$$Z = \frac{\ln(1 - W) - \mu}{\sigma}$$

where, for $x = \ln n$,

$$\mu = -1.5861 - 0.31082x - 0.083751x^2 + 0.0038915x^3$$
$$\sigma = \exp\left(-0.4803 - 0.082676x + 0.0030302x^2\right)$$

Step 7: Calculate the $p$-value as the upper tail from the standard normal distribution:

$p$-value = $\Pr(Z \ge z) = 1 - \phi(|z|)$

## Shapiro Francia Test

The Shapiro Francia test statistic $W'$ is given by

$$W' = \frac{\left(\sum_{i=1}^{n} b_i y_i\right)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = \frac{\left(\sum_{i=1}^{n} \tilde{m}_i X_{(i)}\right)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n} \tilde{m}_i^2} \tag{14}$$

where $y_1 \leq \ldots \leq y_n$ are ordered statistics, the $\tilde{m}_i$ are Blom scores, and

$$\mathbf{b} = (b_1, \ldots, b_i, \ldots, b_n) = \frac{\tilde{\mathbf{m}}^{\mathrm{T}}}{\sqrt{(\tilde{\mathbf{m}}^{\mathrm{T}}\tilde{\mathbf{m}})}} = \frac{\tilde{\mathbf{m}}^{\mathrm{T}}}{\sqrt{\sum_{i=1}^{n} \tilde{\mathbf{m}}^2}}$$

which is called the Shapiro Francia statistic for normality tests (Shapiro & Francia, 1972). If the sample data are leptokurtic, the Shapiro-Francia test is recommended; whereas for platycurtic data, the Shapiro-Wilk test is preferred.

Royston (1993) proposed an approximation for the Shapiro-Francia test to calculate the $p$-value. Mbah and Paothong (2015) use Royston's approximation algorithm for $p$-value calculation when they compared the Shapiro-Francia test with other tests. The algorithm for Royston's approximation (for sample sizes $5 \leq n \leq 5000$) is given below:

Step 1: Sort the sample observations in ascending order, i.e. $\vec{\mathbf{x}} = (x_1, \ldots, x_n)$ into $\vec{\mathbf{y}} = (y_1, \ldots, y_n)$.

Step 2: Calculate the Shapiro Francia test statistic $W'$ from equation (14).

Step 3: Normalize the test statistic $W'$.

$$Z = \frac{\ln(1 - W') - \mu}{\sigma}$$

where, for $u = \ln(n)$, $v = \ln(u)$,

$$\mu = -1.2725 + 1.0521(v - u)$$
$$\sigma = 1.0308 + 0.26758(v + 2/u)$$

Step 4: Calculate the $p$-value as the upper tail from the standard normal distribution:

$p\text{-value} = \Pr(Z \geq z) = 1 - \phi(|z|)$

## Moment Tests

### Jarque-Bera Test

The Jarque-Bera (JB) test is a goodness of fit measure calculated from sample kurtosis and skewness (Jarque & Bera, 1987). The normal distribution has a skewness coefficient of zero and a kurtosis of three. The test statistic JB is then defined by:

$$\mathrm{JB} = n\left[\frac{\left(\sqrt{b_1}\right)^2}{6} - \frac{\left(b_2 - 3\right)^2}{24}\right] \tag{15}$$

where the sample skewness is $\sqrt{b_1} = \mu_3 / \mu_2^{3/2}$ and the sample kurtosis is $b_2 = \mu_4 / \mu_2^2$, where $\mu_2$, $\mu_3$, and $\mu_4$ are the second, third, and fourth central moments, respectively. The $j^{\text{th}}$ moment is calculated by

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^j, \quad j = 2,3,4 \tag{16}$$

The JB statistic has an asymptotic chi-square distribution with two degrees of freedom and $H_0$ should be rejected at a significance level $\alpha$ if $\mathrm{JB} \geq \chi_\alpha^2(2)$.

The algorithm for calculating the JB test statistic is as follows:

Step 1: The sample vector $\mathbf{x}$ is entered.
Step 2: $\mathrm{mean}(\mathbf{x}) = \sum x_i / n$.
Step 3: Second moment $\mu_2 = \sum_{i=1}^{n}\left(x_i - \mathrm{mean}(\mathbf{x})\right)^2 / n$.
Step 4: Third moment $\mu_3 = \sum_{i=1}^{n}\left(x_i - \mathrm{mean}(\mathbf{x})\right)^3 / n$.
Step 5: Fourth moment $\mu_4 = \sum_{i=1}^{n}\left(x_i - \mathrm{mean}(\mathbf{x})\right)^4 / n$.
Step 6: Skewness $\sqrt{b_1} = \mu_3 / \mu_2^{3/2}$.
Step 7: Kurtosis $b_2 = \mu_4 / \mu_2^2$.
Step 8: Calculate the JB test statistic from equation (16).
Step 9: Calculate the *p*- value from the chi-square distribution with two degrees of freedom.

## D'Agostino and Pearson Test

The D'Agostino and Pearson (DAP) test aggregates the skewness and kurtosis tests. The test statistic is defined by

$$\mathrm{DAP} = \mathrm{Z}^2\left(\sqrt{b_1}\right) + \mathrm{Z}^2\left(b_2\right) \tag{17}$$

The skewness test statistic $\mathrm{Z}\left(\sqrt{b_1}\right)$ and kurtosis test statistic $\mathrm{Z}\left(b_2\right)$ are approximately normally distributed, and the DAP test statistic has an asymptotic chi-square distribution with two degrees of freedom (D'Agostino & Pearson, 1973). The algorithm for calculating the *p*-value of DAP test is given below:

Step 1: Compute the sample skewness $\sqrt{b_1} = \mu_3 / \mu_2^{3/2}$.

Step 2: Compute the following values:

$$Y = \sqrt{b_1}\sqrt{\frac{(n+1)(n+3)}{6(n-2)}}$$

$$\beta_2\left(\sqrt{b_1}\right) = \frac{3\left(n^2+27n-70\right)(n+1)(n+3)}{(n-2)(n+5)(n+7)(n+9)}$$

$$W^2 = -1 + \left[2\beta_2\left(\sqrt{b_1}\right) - 1\right]^{1/2}$$

$$\alpha = \left[2 / \left(W^2 - 1\right)\right]^{1/2}$$

Step 3: Compute the skewness test statistic $Z\left(\sqrt{b_1}\right)$:

$$\mathrm{Z}\left(\sqrt{b_1}\right) = \frac{1}{\sqrt{\ln W}} \ln\left[\frac{Y}{\alpha} + \left[\left(\frac{Y}{\alpha}\right)^2 + 1\right]^{1/2}\right]$$

Step 4: Compute the sample kurtosis $b_2 = \mu_4 / \mu_2^2$.

Step 5: Compute the following values:

$$\mathrm{E}(b_2) = \frac{3(n-1)}{(n+1)}$$

$$\mathrm{Var}(b_2) = \frac{24n(n-2)(n-3)}{(n+1)^2(n+3)(n+5)}$$

$$X = \frac{b_2 - \mathrm{E}(b_2)}{\mathrm{Var}(b_2)}$$

$$\sqrt{\beta_1(b_2)} = \frac{6(n^2-5n+2)}{(n+7)(n+9)}\left[\frac{6(n+3)(n+5)}{n(n-2)(n-3)}\right]^{1/2}$$

$$A = 6 + \frac{8}{\sqrt{\beta_1(b_2)}}\left[\frac{2}{\sqrt{\beta_1(b_2)}} + \left(1+\frac{4}{\beta_1(b_2)}\right)^{1/2}\right]$$

Step 6: Compute the kurtosis test statistic $Z(b_2)$.

$$\mathrm{Z}(b_2) = \frac{1}{\sqrt{2/(9A)}}\left[\left(1-\frac{2}{9A}\right)-\left(\frac{1-2/A}{1+X\sqrt{2/(A-4)}}\right)^{1/3}\right]$$

Step 7: Compute the DAP test statistic from equation (17).

$$\mathrm{DAP} = \mathrm{Z}^2\left(\sqrt{b_1}\right) + \mathrm{Z}^2(b_2)$$

Step 8: Calculate the $p$-value from the chi-square distribution with two degrees of freedom.

## Codes, Execution, and Output

All algorithms for these ten normality tests are coded in the Matlab2015 environment and presented as a function (normalitytest.m). Data should be a $1 \times n$ row vector (in “**x** = […]” format) and entered as a variable in the workspace. The function gives a display of results, as well as a $10 \times 3$ matrix named “Results,” including test statistics in the first column, $p$-values in the second, and test result in the last. The code file is available both as an Appendix and as an .m file in the MathWorks File Exchange under the name “normality test package” (Öner & Deveci Kocakoç, 2016).

Here, utilization of the function is shown on two sample data sets of 20 data. The first data set is normally distributed:

```
x1=[66 53 154 76 73 118 106 69 87 84 41 33 78 56 55 35 44 135 75 58];
```

After the command "Results=normalitytest(x1)," the output in Figure 1 can be obtained. By using the common threshold of $\alpha = 0.05$, data is found to be normally distributed by all tests since all *p*-values are above 0.05.

```
Command Window
>> x1=[66 53 154 76 73 118 106 69 87 84 41 33 78 56 55 35 44 135 75 58];
>> Results=normalitytest(x1)

Test Name                  Test Statistic   p-value   Normality (1:Normal,0:Not Normal)
------------------------   --------------   ---------   --------------------------------
KS Limiting Form                 0.7188       0.6797                   1
KS Stephens Modification         0.7478       0.1500                   1
KS Marsaglia Method              0.7188       0.6229                   1
KS Lilliefors Modification       0.1607       0.1876                   1
Anderson-Darling Test            0.5951       0.1204                   1
Cramer- Von Mises Test           0.0956       0.1295                   1
Shapiro-Wilk Test                0.9186       0.0930                   1
Shapiro-Francia Test             0.9206       0.0935                   1
Jarque-Bera Test                 2.9892       0.2243                   1
DAgostino & Pearson Test         4.6285       0.0988                   1
```

**Figure 1.** Output for the first data set

```
Command Window
>> x2=[66 253 154 276 73 118 106 69 87 84 41 33 78 56 55 35 44 135 75 58];
>> Results=normalitytest(x2)

Test Name                  Test Statistic   p-value   Normality (1:Normal,0:Not Normal)
------------------------   --------------   ---------   --------------------------------
KS Limiting Form                 1.1035       0.1750                   1
KS Stephens Modification         1.1479       0.0100                   0
KS Marsaglia Method              1.1035       0.1473                   1
KS Lilliefors Modification       0.2468       0.0024                   0
Anderson-Darling Test            1.6833       0.0006                   0
Cramer- Von Mises Test           0.2751       0.0007                   0
Shapiro-Wilk Test                0.7758       0.0004                   0
Shapiro-Francia Test             0.7719       0.0007                   0
Jarque-Bera Test                13.1210       0.0014                   0
DAgostino & Pearson Test        14.7783       0.0006                   0
```

**Figure 2.** Output for the second data set.

The second data set is disturbed by changing some of the values in the first data set:

```
x2=[66 253 154 276 73 118 106 69 87 84 41 33 78 56 55 35 44 135 75 58];
```

After the command "Results=normalitytest(x2)," the output in Figure 2 can be obtained. By using the common threshold of $\alpha = 0.05$, data is found to be normally distributed by only two of the tests. The KS test is known to have some problems with outliers and small sample sizes (Steinskog, Tjøstheim, Kvamstø, 2007). The Stephens' and Lilliefors modifications seem to overcome these problems, while the limiting form and Marsaglia method do not. Since it is not in the scope of this article to discuss the powers of the tests, the rest of the interpretation is left to the reader.

The function also gives these results as a matrix to use for other purposes. By this function, a gap in statistical computing can be filled for users of MATLAB as well as for researchers who would like to know the calculation details of these test statistics. This MATLAB function can be used to compute all ten test statistics and to have the results of normality tests with just one command. The next step after this study is to build an API with a user-friendly interface to perform the tests without the need of any MATLAB knowledge.

## References


Anderson, T. W., & Darling, D. A. (1952). Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, *23*(2), 193-212. doi: 10.1214/aoms/1177729437

Anderson, T. W., & Darling, D. A. (1954). A test of goodness of fit. *Journal of the American Statistical Association*, *49*(268), 765-769. doi: 10.1080/01621459.1954.10501232

Csörgö, S., & Faraway, J. J. (1996).The exact and asymptotic distributions of Cramer-von Mises statistics. *Journal of Royal Statistical Society*. *Series B (Methodological)*, *58*(1), 221-234. Available from http://www.jstor.org/stable/2346175

D'Agostino, R. B., & Pearson, E. S. (1973). Tests for departures from normality. Empirical results for the distribution of $b^2$ and $\sqrt{b^1}$. *Biometrika*, *60*(3), 613-622. doi: 10.1093/biomet/60.3.613

Dellal, G. E., & Wilkinson, L. (1986). An analytic approximation to the distribution of Lilliefors's test statistic for normality. *The American Statistician, 40*(4), 294-296. doi: 10.1080/00031305.1986.10475419

Facchinetti, S. (2009). A procedure to find exact critical values of Kolmogorov-Smirnov test. *Statistica Applicata – Italian Journal of Applied Statistics, 21*(3-4), 337-359. Available from http://sa-ijas.stat.unipd.it/sites/sa-ijas.stat.unipd.it/files/IJAS_3-4_2009_07_Facchinetti.pdf

Jarque, C. M., & Bera, A. K. (1987). A test for normality of observations and regression residuals. *International Statistical Review, 55*(2), 163-172. doi: 10.2307/1403192

Kolmogorov, A. (1933). Sulla determinazione empirica di una leggi di distribuzione [On the empirical determination of a distribution]. *Giornale dell'Istituto Italiano degli Attuari, 4*, 83-91.

Lilliefors, H. W. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association, 62*(318), 399-402. doi: 10.1080/01621459.1967.10482916

Marsaglia, G., Tsang, W. W., & Wang, J. (2003). Evaluating Kolmogorov's distribution. *Journal of Statistical Software, 8*(18). doi: 10.18637/jss.v008.i18

Mbah, A. K., & Paothong, A. (2015). Shapiro-Francia test compared to other normality test using expected *p*-value. *Journal of Statistical Computation and Simulation, 85*(15), 3002-3016. doi: 10.1080/00949655.2014.947986

Öner, M., Deveci Kocakoç, I. (2016). *Normality test package* [MATLAB function]. Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/60147-normality-test-package

Royston, P. (1982). An extension of Shapiro and Wilk's *W* test for normality to large samples. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 31*(2), 115-124. doi: 10.2307/2347973

Royston, P. (1993). A pocket-calculator algorithm for the Shapiro-Francia test for non-normality: An application to medicine. *Statistics in Medicine, 12*(2), 181-184. doi: 10.1002/sim.4780120209

Scott, W. F., & Stewart, B. (2011). Tables for the Lilliefors and modified Cramer-von Mises tests of normality. *Communications in Statistics – Theory and Methods, 40*(4), 726-730. doi: 10.1080/03610920903453467

Shapiro, S. S., & Francia, R. S. (1972). An approximate analysis of variance test for normality. *Journal of the American Statistical Association, 67*(337), 215-216. doi: 10.1080/01621459.1972.10481232

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika, 52*(3-4), 591-611. doi: 10.1093/biomet/52.3-4.591

Solomon, S. R., & Sawilowsky, S. S. (2009). Impact of rank-based normalizing transformations on the accuracy of test scores. *Journal of Modern Applied Statistical Methods, 8*(2), 448-462. doi: 10.22237/jmasm/1257034080

Steinskog, D. J., Tjøstheim, D. B., & Kvamstø, N. G. (2007). A cautionary note on the use of the Kolmogorov-Smirnov test for normality. *Monthly Weather Review, 135*(3), 1151-1157. doi: 10.1175/mwr3326.1

Stephens, M. A. (1986). Tests based on EDF statistics. In R. B. D'Agostino & M. A. Stephens (Eds.), *Goodness-of-fit techniques* (pp. 97-194). New York, NY: Marcel Dekker.

## Appendix: MATLAB Function (normalitytest.m)

```
function Results=normalitytest(x)

% Enter the data as a row vector in the workspace

%example1: Normally distributed data
%x1=[66 53 154 76 73 118 106 69 87 84 41 33 78 56 55 35 44 135 75 58];

%example1: Disturbed data
%x2=[66 253 154 276 73 118 106 69 87 84 41 33 78 56 55 35 44 135 75 58];

% Alpha value can be changed as required
alpha=0.05;

% KOLMOGOROV-SMIRNOV TEST- LIMITING FORM

n=length(x);
i=1:n;
y=sort(x);
fx=normcdf(zscore(y));
dplus=max(abs(fx-i/n));
dminus=max(abs(fx-(i-1)/n));
Dn=max(dplus,dminus);
KSz=sqrt(n)*Dn;
s=-20:1:20;
a=(-1).^s.*exp(-2*(s.*KSz).^2);
pvalue=1-sum(a);
Results(1,1)=KSz;
Results(1,2)=pvalue;

% KOLMOGOROV-SMIRNOV TEST - STEPHENS MODIFICATION

dKSz=Dn*(sqrt(n)-0.01+0.85/sqrt(n));

if dKSz<0.775
    pvalue=0.15;
```

```
elseif dKSz<0.819
    pvalue=((0.10-0.15)/(0.819-0.775))*(dKSz-0.775)+0.15;
elseif dKSz<0.895
    pvalue=((0.05-0.10)/(0.895-0.819))*(dKSz-0.819)+0.10;
elseif dKSz<0.995
    pvalue=((0.025-0.05)/(0.995-0.895))*(dKSz-0.895)+0.05;
elseif dKSz<1.035
    pvalue=((0.01-0.025)/(1.035-0.995))*(dKSz-0.995)+0.025;
else
    pvalue=0.01;
end
Results(2,1)=dKSz;
Results(2,2)=pvalue;

% KOLMOGOROV-SMIRNOV TEST - MARSAGLIA METHOD

k=ceil(n*Dn);
m=2*k-1;
h=k-n*Dn;

Hmatrix=zeros(m,m);

for i=1:m-1
   for j=2:m
      if i-j+1>=0
      Hmatrix(i,j)=1/factorial(i-j+1);
    else
      Hmatrix(i,j)=0;
    end
    end
end

for i=1:m-1
    Hmatrix(i,1)=(1-h^i)/factorial(i);
end

Hmatrix(m,:)=fliplr(Hmatrix(:,1)');
```

```
if h<=0.5
Hmatrix(m,1)=(1 - 2*h^m)/factorial(m);
else
Hmatrix(m,1)=(1 - 2*h^m + max(0,2*h-1)^m)/factorial(m);
end
    lmax = max(eig(Hmatrix));
    Hmatrix = (Hmatrix./lmax)^n;
    pvalue  =  (1  -  exp(gammaln(n+1)  +  n*log(lmax)  -  n*log(n))  *
Hmatrix(k,k));
Results(3,1)=KSz;
Results(3,2)=pvalue;

% KOLMOGOROV-SMIRNOV TEST - LILLIEFORS MODIFICATION

% P = [n D20 D15]
P=[5 0.289 0.303;
   6 0.269 0.281;
   7 0.252 0.264;
   8 0.239 0.250;
   9 0.227 0.238;
   10 0.217 0.228;
   11 0.208 0.218;
   12 0.200 0.210;
   13 0.193 0.202;
   14 0.187 0.196;
   15 0.181 0.190;
   16 0.176 0.184;
   17 0.171 0.179;
   18 0.167 0.175;
   19 0.163 0.170;
   20 0.159 0.166;
   25 0.143 0.150;
   30 0.131 0.138;
   40 0.115 0.120;
   100 0.074 0.077;
   400 0.037 0.039;
   900 0.025 0.026];
```

```
aaa=P(:,1)';
subind=max(find(aaa<n));
upind=subind+1;
xxx=P(subind:upind,:);

if aaa(upind)==n
   D20=xxx(2,2);
   D15=xxx(2,3);
else
    D20=xxx(1,2)+(n-aaa(subind))*((xxx(2,2)-xxx(1,2))/(xxx(2,1)-
xxx(1,1)));
    D15=xxx(1,3)+(n-aaa(subind))*((xxx(2,3)-xxx(1,3))/(xxx(2,1)-
xxx(1,1)));
end

a1=-7.01256*(n+2.78019);
b1=2.99587*sqrt(n+2.78019);
c1=2.1804661+0.974598/sqrt(n)+1.67997/n;

a2=-7.90289126054*(n^0.98);
b2=3.180370175721*(n^0.49);
c2=2.2947256;

if n>100
   D10=(-b2-sqrt(b2^2-4*a2*c2))/(2*a2);
   a=a2;
   b=b2;
   c=c2;
else
   D10=(-b1-sqrt(b1^2-4*a1*c1))/(2*a1);
   a=a1;
   b=b1;
   c=c1;
end

if Dn==D10
        pvalue=0.10;
    elseif Dn>D10
```

```
        pvalue=exp(a*Dn^2+b*Dn+c-2.3025851);
    elseif Dn>=D15
        pvalue=((0.10-0.15)/(D10-D15))*(Dn-D15)+0.15;
    elseif Dn>=D20
       pvalue=((0.15-0.20)/(D15-D20))*(Dn-D20)+0.20;
    else
       pvalue=0.20;
end
Results(4,1)=Dn;
Results(4,2)=pvalue;

% ANDERSON-DARLING TEST

adj=1+0.75/n+2.25/(n^2);
i=1:n;
ui=normcdf(zscore(y),0,1);
oneminusui=sort(1-ui);
lastt=(2*i-1).*(log(ui)+log(oneminusui));
asquare=-n-(1/n)*sum(lastt);
AD=asquare*adj;

if AD<=0.2
    pvalue=1-exp(-13.436+101.14*AD-223.73*AD^2);
elseif AD<=0.34
    pvalue=1-exp(-8.318+42.796*AD-59.938*AD^2);
elseif AD<=0.6
    pvalue=exp(0.9177-4.279*AD-1.38*AD^2);
elseif AD<=153.467
    pvalue=exp(1.2937*AD-5.709*AD+0.0186*AD^2);
else
    pvalue=0;
end
Results(5,1)=AD;
Results(5,2)=pvalue;

% CRAMER-VON MISES TEST

adj=1+0.5/n;
```

```
i=1:n;
fx=normcdf(zscore(y),0,1);
gx=(fx-((2*i-1)/(2*n))).^2;
CvMteststat=(1/(12*n))+sum(gx);
AdjCvM=CvMteststat*adj;

if AdjCvM<0.0275
    pvalue=1-exp(-13.953+775.5*AdjCvM-12542.61*(AdjCvM^2));
elseif AdjCvM<0.051
    pvalue=1-exp(-5.903+179.546*AdjCvM-1515.29*(AdjCvM^2));
elseif AdjCvM<0.092
    pvalue=exp(0.886-31.62*AdjCvM+10.897*(AdjCvM^2));
elseif AdjCvM>=0.093
    pvalue=exp(1.111-34.242*AdjCvM+12.832*(AdjCvM^2));
end
Results(6,1)=AdjCvM;
Results(6,2)=pvalue;

% SHAPIRO-WILK TEST

a=[];
i=1:n;
mi=norminv((i-0.375)/(n+0.25));
u=1/sqrt(n);
m=mi.^2;

a(n)=-2.706056*(u^5)+4.434685*(u^4)-2.07119*(u^3)-
0.147981*(u^2)+0.221157*u+mi(n)/sqrt(sum(m));
a(n-1)=-3.58263*(u^5)+5.682633*(u^4)-1.752461*(u^3)-
0.293762*(u^2)+0.042981*u+mi(n-1)/sqrt(sum(m));
a(1)=-a(n);
a(2)=-a(n-1);
eps=(sum(m)-2*(mi(n)^2)-2*(mi(n-1)^2))/(1-2*(a(n)^2)-2*(a(n-1)^2));
a(3:n-2)=mi(3:n-2)./sqrt(eps);
    ax=a.*y;
    KT=sum((x-mean(x)).^2);
    b=sum(ax)^2;
    SWtest=b/KT;
```

```
mu=0.0038915*(log(n)^3)-0.083751*(log(n)^2)-0.31082*log(n)-1.5861;
sigma=exp(0.0030302*(log(n)^2)-0.082676*log(n)-0.4803);
z=(log(1-SWtest)-mu)/sigma;
pvalue=1-normcdf(z,0,1);
Results(7,1)=SWtest;
Results(7,2)=pvalue;

% SHAPIRO-FRANCIA TEST

mi=norminv((i-0.375)/(n+0.25));
micarp=sqrt(mi*mi');
weig=mi./micarp;
pay=sum(y.*weig)^2;
payda=sum((y-mean(y)).^2);
SFteststa=pay/payda;

u1=log(log(n))-log(n);
u2=log(log(n))+2/log(n);
mu=-1.2725+1.0521*u1;
sigma=1.0308-0.26758*u2;

zet=(log(1-SFteststa)-mu)/sigma;
pvalue=1-normcdf(zet,0,1);
Results(8,1)=SFteststa;
Results(8,2)=pvalue;

% JARQUE-BERA TEST

E=skewness(y);
B=kurtosis(y);
JBtest=n*((E^2)/6+((B-3)^2)/24);
pvalue=1-chi2cdf(JBtest,2);
Results(9,1)=JBtest;
Results(9,2)=pvalue;

% D'AGOSTINO-PEARSON TEST

beta2=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
```

```
wsquare=-1+sqrt(2*(beta2-1));
delta=1/sqrt(log(sqrt(wsquare)));
alfa=sqrt(2/(wsquare-1));

expectedb2=(3*(n-1))/(n+1);
varb2=(24*n*(n-2)*(n-3))/(((n+1)^2)*(n+3)*(n+5));
sqrtbeta=((6*(n^2-5*n+2))/((n+7)*(n+9)))*sqrt((6*(n+3)*(n+5))/(n*(n-
2)*(n-3)));
A=6+(8/sqrtbeta)*(2/sqrtbeta+sqrt(1+4/(sqrtbeta^2)));

squarerootb=skewness(y);
Y=squarerootb*sqrt(((n+1)*(n+3))/(6*(n-2)));
zsqrtbtest=delta*log(Y/alfa+sqrt((Y/alfa)^2+1));

b2=kurtosis(y);
zet=(b2-expectedb2)/sqrt(varb2);
ztestb2=((1-2/(9*A))-((1-2/A)/(1+zet*sqrt(2/(A-
4))))^(1/3))/sqrt(2/(9*A));

DAPtest=zsqrtbtest^2+ztestb2^2;

pvalue=1-chi2cdf(DAPtest,2);
Results(10,1)=DAPtest;
Results(10,2)=pvalue;

% Compare p-value to alpha
for i=1:10
    if Results(i,2)>alpha
        Results(i,3)=1;
    else
        Results(i,3)=0;
    end
end

% Output display

disp(' ')
```

```
disp('Test Name                     Test Statistic    p-value    Normality
(1:Normal,0:Not Normal)')
disp('------------------------    --------------  ---------  ------------
--------------------')
fprintf('KS Limiting Form            %6.4f \t    %6.4f              %1.0f
\r',KSz,Results(1,2),Results(1,3))
fprintf('KS Stephens Modification    %6.4f \t    %6.4f              %1.0f
\r',dKSz,Results(2,2),Results(2,3))
fprintf('KS Marsaglia Method         %6.4f \t    %6.4f              %1.0f
\r',KSz,Results(3,2),Results(3,3))
fprintf('KS Lilliefors Modification  %6.4f \t    %6.4f              %1.0f
\r',Dn,Results(4,2),Results(4,3))
fprintf('Anderson-Darling Test       %6.4f \t    %6.4f              %1.0f
\r',AD,Results(5,2),Results(5,3))
fprintf('Cramer-Von Mises Test       %6.4f \t    %6.4f              %1.0f
\r',AdjCvM,Results(6,2),Results(6,3))
fprintf('Shapiro-Wilk Test           %6.4f \t    %6.4f              %1.0f
\r',SWtest,Results(7,2),Results(7,3))
fprintf('Shapiro-Francia Test        %6.4f \t    %6.4f              %1.0f
\r',SFteststa,Results(8,2),Results(8,3))
fprintf('Jarque-Bera Test            %6.4f \t    %6.4f              %1.0f
\r',JBtest,Results(9,2),Results(9,3))
fprintf('DAgostino & Pearson Test    %6.4f \t    %6.4f              %1.0f
\r',DAPtest,Results(10,2),Results(10,3))
```