Determining the Statistical Power of the Kolmogorov-Smirnov and Anderson-Darling Goodness-of-Fit Tests via Monte Carlo Simulation

Brad M. Boyerinas

December 2016

CNA
ANALYSIS & SOLUTIONS

Approved by: **December 2016**

Dr. Timothy Roberts, Director
Aviation Systems and Technology
Advanced Technology and Systems Analysis

Abstract

Metrics are often used to compare the performance of newly developed systems with the performance of their predecessors. Metrics can also be used to compare the output of a simulator with real-world data to test the accuracy of the simulation. Statistical comparison of these metrics can be necessary when making such a determination. There are different methods of statistical comparison that are sensitive to the various types of underlying distribution of the metric data. Distribution type can affect the performance of these tests, and, fortunately, the distributions of many common metrics are well known. For example, mean time to repair (MTTR) and mean flight hours between critical failures (MFHBCF), generally follow a log-normal and an exponential distribution, respectively. This paper presents the effects of distribution type and parameters on the statistical power of two common goodness-of-fit tests (Kolmogorov–Smirnov and Anderson-Darling) via Monte Carlo simulation.

This page intentionally left blank.

Contents

This page intentionally left blank.

List of Figures

This page intentionally left blank.

List of Tables

This page intentionally left blank.

Introduction

The results of a Monte Carlo simulation that calculates the statistical power of two common goodness-of-fit (GoF) tests are presented and analyzed in this paper. Various distribution types are considered, including the normal, lognormal, and exponential distributions. The results of this study provide required sample size as a function of statistical power. The presented data can be used to determine the minimum required sample size for a desired level of power. The simulation methodology can be adapted to calculate statistical power for the same distributions with different parameters or other distribution types.

Goodness-of-Fit Testing

Goodness-of-fit (GoF) testing is a technique used to determine how well a statistical model fits a data set. Single-sample GoF tests consider a null and an alternative hypothesis to confirm whether a sample could have been drawn from a population with a particular distribution. Multi-sample GoF tests determine whether the samples could have been drawn from populations with the same distribution. Thus, GoF tests are useful for validating whether simulation output is similar to real-world data, and for comparing the performance of a new system to that of a previous generation. Two such tests, Kolmogorov–Smirnov (KS) and Anderson-Darling (AD), are the subjects of discussion in this paper, and their behaviors in terms of statistical power are analyzed and presented. Determining statistical power is important for test design because it enables the designer to choose a minimum sample size required to detect a difference between samples (i.e., the GoF result may be too unreliable if the required sample size is not used for the test).

Two-Sample KS and AD Tests

The two-sample KS and AD tests are GoF tests used to infer whether two samples were drawn from populations with the same distribution. In both tests, the empirical distribution function (EDF) of each sample is used to calculate the test statistic. The EDF is a step function that steps by 1/n for each occurrence of n, as shown in Figure 1. for the case of two normally distributed samples. If the value of the test statistic is

larger than a critical value for a given significance, or if the p-value is less than the given level of significance, the null hypothesis is rejected and one can infer that the samples were drawn from populations with dissimilar distributions. Both tests can accommodate equal or unequal sample sizes among the two samples being considered. The test statistics for the KS and AD tests are shown below, respectively, in Equations 1 and 2 [1].

Figure 1.   Empirical distribution functions of two randomly collected and normally distributed samples with µ and σ of 1
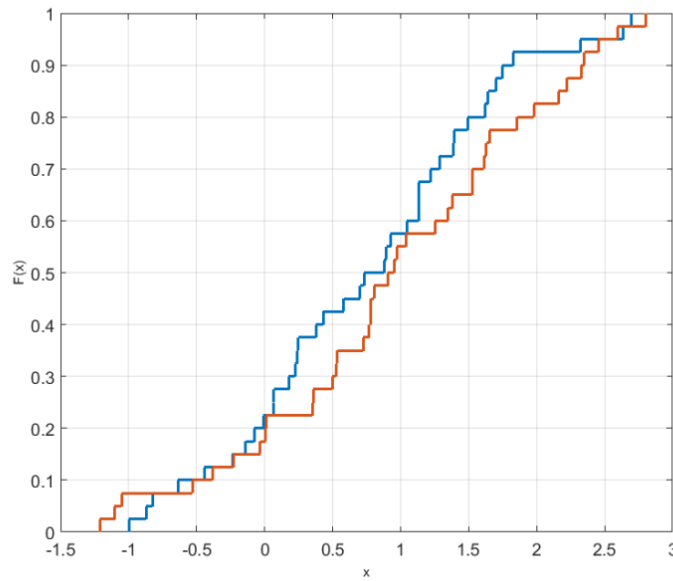
$$KS = max\left|F_{n_1}(x) - G_{n_2}(x)\right| \tag{1}$$

where $F_{n_1}(x)$ and $G_{n_2}(x)$ are EDFs of the two samples. The equations used to determine the KS critical values for varying levels of significance are shown in Appendix B as a function of c(α), n$_1$, and n$_2$.

$$AD = \frac{n_1 n_2}{N} \int_{-\infty}^{\infty} \frac{\left\{F_{n_1}(x) - G_{n_2}(x)\right\}^2}{H_N(x)\{1 - H_N(x)\}} dH_N(x) \tag{2}$$

where $F_{n_1}(x)$ and $G_{n_2}(x)$ are EDFs of the two samples with sample sizes n_1 and n_2, $n_1 + n_2 = N$ and $H_N(x) = \left\{n_1 F_{n_1}(x) + n_2 G_{n_2}(x)\right\}/N$.

2

Equation 2 can be generalized in discrete form, as shown in Equation 3 [2]:

$$AD = \frac{N-1}{N^2}\left[\frac{1}{n_1}\sum_{i=1}^{L} h_i \frac{\left(NF_{n_1 i} - n_1 H_i\right)^2}{H_i(N-H_i) - N\frac{h_i}{4}} + \frac{1}{n_2}\sum_{i=1}^{L} h_i \frac{\left(NF_{n_2 i} - n_2 H_i\right)^2}{H_i(N-H_i) - N\frac{h_i}{4}}\right] \qquad (3)$$

where z_i is the array with length L of the distinct values of the two samples ordered from smallest to largest, N is the total number of data points of the two samples (N=n_1+n_2), h_i is the number of values in the combined samples equal to z_i, H_i is the number of values in the combined samples less than z_i plus one half the number of values in the combined samples equal to z_i, and F_{n1i} and F_{n2i} are the number of values in group n_1 or n_2 that are less than z_i plus one half the number of values in the specific group equal to z_i.

The method to determine the p-value of the two-sample AD test statistic is shown in Appendix B. It was adapted from reference [2]. Alternatively, a critical value can be calculated for a direct comparison to the test statistic when performing a hypothesis test.

It should be noted that the KS test is less complex than the AD test, both on an intuitive and a computational level. The KS test statistic simply looks for the maximum distance between EDFs for the two samples along their entire range, and is more sensitive to discrepancies between EDFs toward the median, while the AD test statistic integrates over their entirety and includes a weighting term $[H(x) * (1 - H(x))]^{-1}$ that places greater emphasis on the tails of the EDFs.

Understanding KS and AD Statistical Power via Monte Carlo Simulation

Statistical power is the probability of correctly rejecting the null hypothesis when the alternative hypothesis is true. It is dependent on sample size and also the difference in parameters (means and variance) between the samples being compared. Because of this, experiment designers must choose a minimal necessary sample size to maintain a minimally acceptable level of power. However, since power is sensitive to differences in sample distribution parameters, assigning an accurate estimation of power to a statistical test is often nontrivial and could require *a-priori* knowledge of sample distribution and parameters. Without this knowledge, or best estimation, one cannot assign a meaningful level of power to a GoF test. The statistical power of KS and AD tests can be analyzed for a given range of parameters and sample sizes to provide insight into relative adequacy of the tests.

Below, the powers of the KS and AD tests are calculated via Monte Carlo simulation for varying sample sizes and distribution parameters. Normal, lognormal, and exponential distributions are considered. For normal and lognormal distributions, a simulation parameter defined as $\Delta\mu/\sigma$ is used to observe the effect of distribution parameters on test power. This parameter is the difference in means between the two samples divided by the sample standard deviation (the standard deviation is assumed to be constant=1 for all cases in the simulation). It is often referred to as the "signal-to-noise ratio" when determining statistical power for normal and lognormal data. For power estimation of the exponential distribution, parameter $\Delta\mu$ is considered, where $\Delta\mu$ is the difference in means between the samples and $\mu=\mu_o-\Delta\mu$. For brevity, the simulation parameters $\Delta\mu/\sigma$ for normal and lognormal and $\Delta\mu$ for exponential are both referred to generally as δ in the schematic in Figure 3.

Since the shape of the exponential probability density function (PDF) relies on the mean parameter, the relative difference in means between two exponential distributions cannot be used alone to sufficiently determine power. In other words, for example, one cannot expect similar power when considering one set of exponential samples with means of 0.2 and 1.2 and another set with means of 4 and 5, even though the difference between both is 1. This observation is displayed in Figure 2, where the relative shapes of two sets of exponential PDFs vary drastically, despite the same difference between means.

Figure 2.   Exponential PDFs with means of $\mu = 0.2$ and 1.2 and $\mu = 4$ and 5
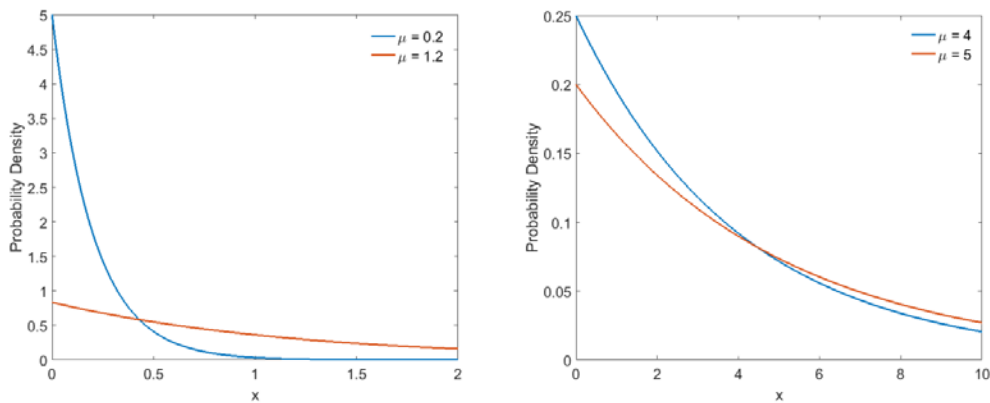
Simulation Methodology

The method used in this paper to simulate statistical power as a function of sample size and δ is shown in Figure 3. The simulation starts by considering a specific distribution type. Then, two samples with chosen sample sizes n_1 and n_2 and parameter δ are randomly drawn. The GoF test is applied at a significance of 0.2 and the result of the hypothesis test is stored. This sequence is iterated for a total of 10,000 times. Then, power is calculated by dividing the number of times the test rejected the null hypothesis by the total number of iterations. For example, if the null hypothesis is rejected 9,500 times out of 10,000 total iterations, the calculated power is 95 percent. This scheme is repeated for varying sample sizes from 4 through 150 with an increment of 2, and δ from 0 through 1 with an increment of 0.1. The Matlab script used to perform the simulation is in Appendix C. VBA code capable of running the AD and KS tests is in Appendix D.

All AD test simulations use an identical sample size for each iteration ($n_1=n_2$), whereas the KS simulation uses a sample size offset by 1 ($n_1=n_2+1$, $N=n_1+n_2$). Using this offset accounts for how the critical value of the two-sample KS test does not increase monotonically for increasing sample size and $n_1=n_2$, especially for small N [3]. This behavior does not significantly affect the results of this study, considering that acceptable levels of power (80 percent or greater) are generally achieved with N>50. However, tables in references [3] and [4] should be consulted when performing the KS test for N<<50 to achieve acceptable accuracy.

Figure 3.   Monte Carlo simulation flowchart for estimating AD and KS statistical power
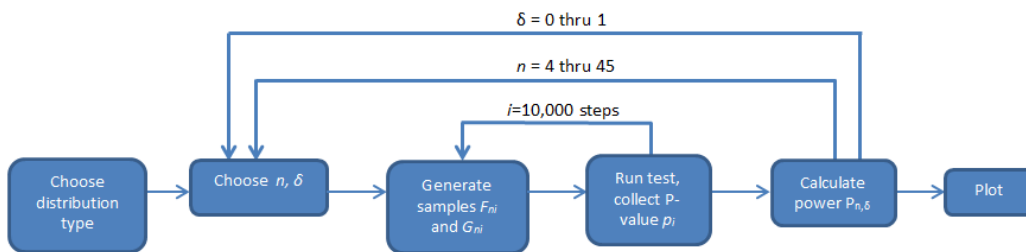
Simulation Results and Discussion

The AD test is generally known to be more sensitive than KS, as shown in Figure 4, due to its greater emphasis on the tails of the data [1], and the results of the simulation in this study reaffirm this for all distributions considered. Statistical power is displayed in Figure 5 through Figure 8 as a function of δ and sample size. The x and y axes represent δ and sample size, respectively, and the color contour in each plot displays the corresponding level of power calculated in the simulation for a given δ and sample size. The legend next to each plot correlates the numeric value of power to the color displayed. Numeric values of power are located in Table 1 through Table 6 in Appendix A for all distributions except the exponential distribution with μ_o=5 because its statistical power is below 0.80 for all values of sample size and δ.

Figure 4.   Simulated statistical power for AD and KS tests with normal distribution (μ=5, σ=1) and δ =4.5
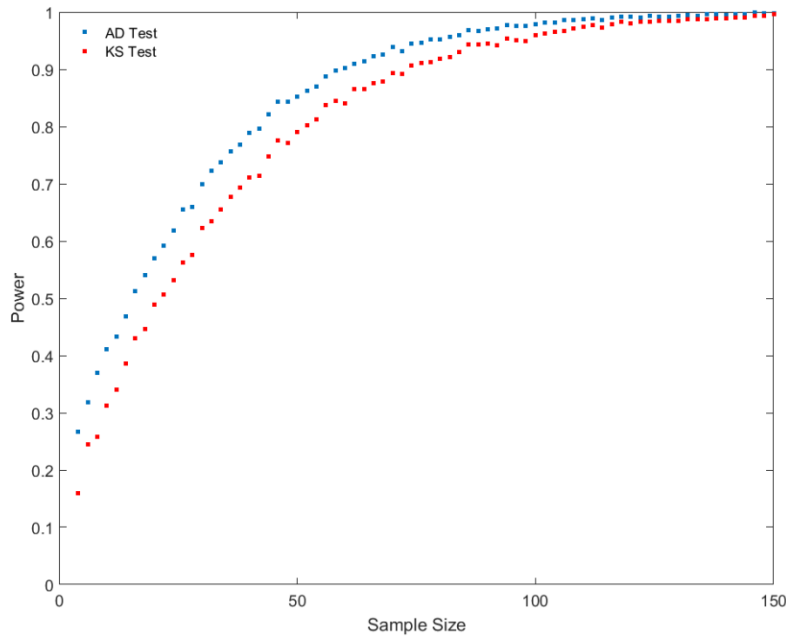
Figure 5.   Simulated statistical power for normal distribution using AD (left) and KS (right) tests
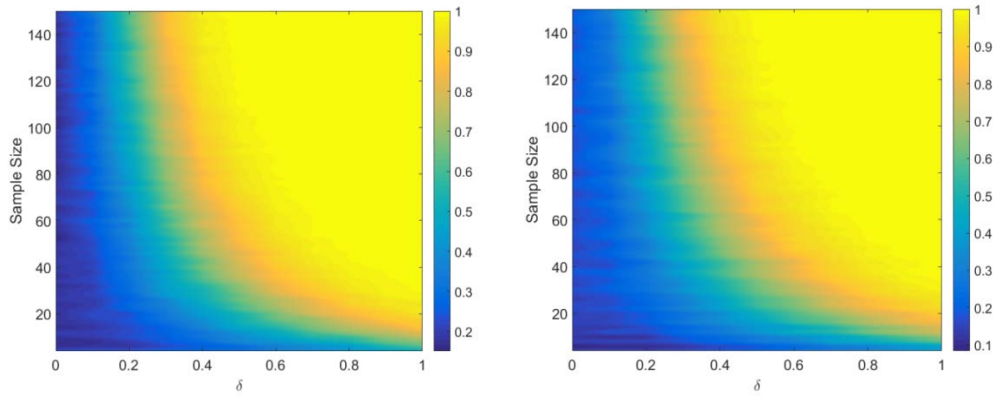
Figure 6.   Simulated statistical power for lognormal distribution using AD (left) and KS (right) tests
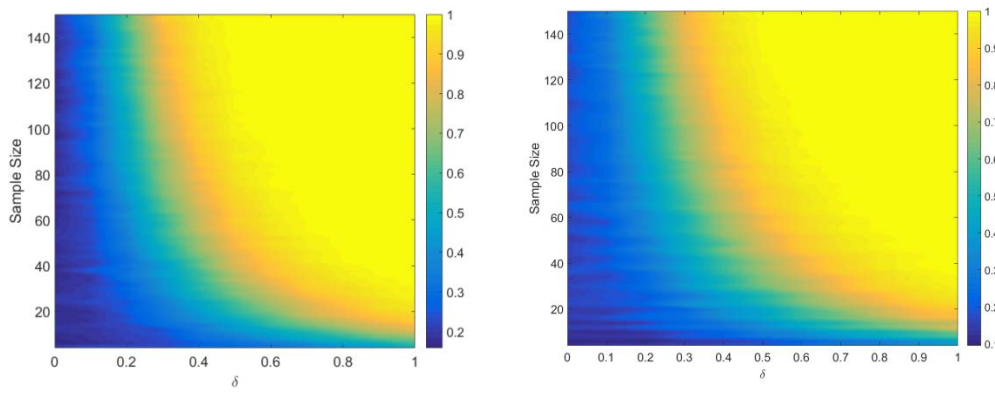
Figure 7.   Simulated statistical power for exponential distribution with $\mu_o=1$ using AD (left) and KS (right) tests
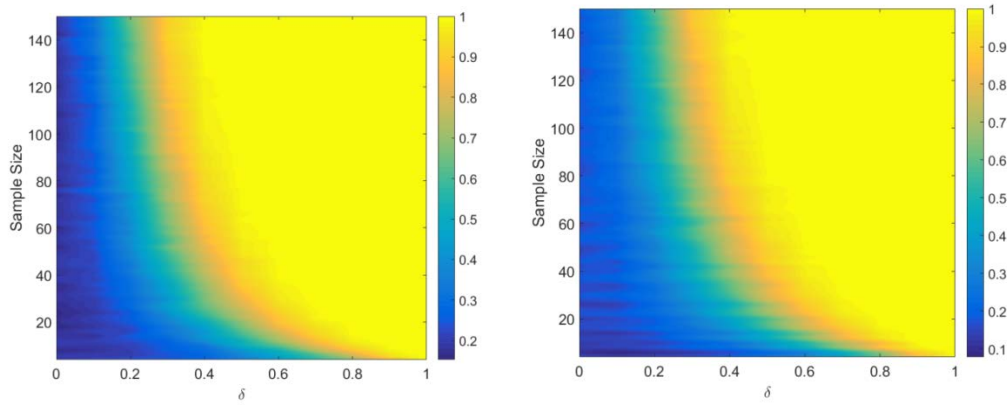
Figure 8.   Simulated statistical power for exponential distribution with $\mu_o=5$ using AD (left) and KS (right) tests
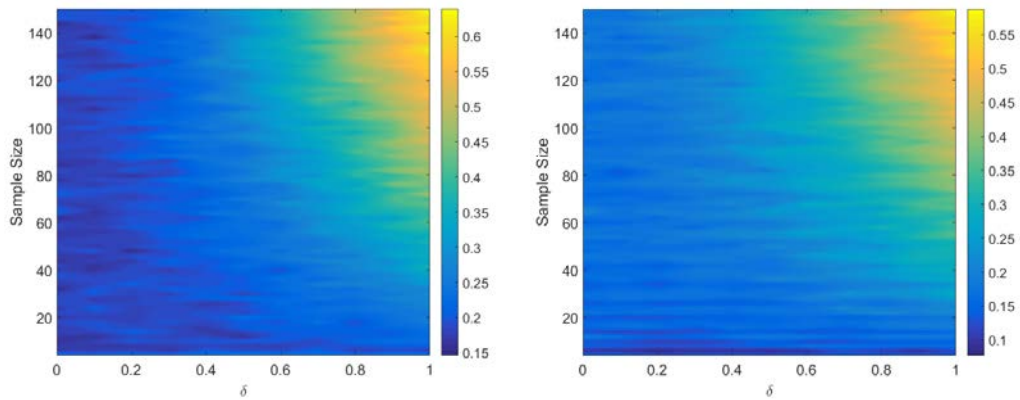
Summary

The results from this study affirm that distribution type and parameters control the statistical power of the AD and KS tests. Larger sample sizes will generally increase power for normal, lognormal, and exponential distributions. The statistical power of exponentially distributed data depends on both the difference in means between samples and the values of the means when using GoF testing. Depending on exponential parameter μ_o, the AD and KS tests may not be able to achieve desirable levels of power regardless of sample size.

This page intentionally left blank.

Appendix A: Tabulated Data from AD and KS Simulations

Table 1.   Required sample size for given power and δ obtained from AD simulation with normal distribution

δ=0.2			δ=0.5			δ=0.7			δ=0.9	
P	n		P	n		P	n		P	n
80	-		80	43		80	22		80	14
90	-		90	64		90	32		90	20
95	-		95	78		95	40		95	23
99	-		99	110		99	54		99	39

Table 2.   Required sample size for given power and δ obtained from AD simulation with lognormal distribution

δ=0.2			δ=0.5			δ=0.7			δ=0.9	
P	n		P	n		P	n		P	n
80	-		80	41		80	23		80	14
90	-		90	58		90	30		90	19
95	-		95	76		95	41		95	26
99	-		99	104		99	56		99	36

Table 3.   Required sample size for given power and δ obtained from AD simulation with exponential distribution and $\mu_o=0$

δ=0.2			δ=0.5			δ=0.7			δ=0.9	
P	n		P	n		P	n		P	n
80	-		80	27		80	9		80	-
90	-		90	36		90	13		90	5
95	-		95	47		95	18		95	6
99	-		99	65		99	26		99	9

Table 4.   Required sample size for given power and δ obtained from KS simulation with normal distribution

δ=0.2		δ=0.5		δ=0.7		δ=0.9	
P	n	P	n	P	n	P	n
80	-	80	52	80	28	80	17
90	-	90	71	90	38	90	25
95	-	95	87	95	49	95	30
99	-	99	138	99	67	99	44

Table 5.   Required sample size for given power and δ obtained from KS simulation with lognormal distribution

δ=0.2		δ=0.5		δ=0.7		δ=0.9	
P	n	P	n	P	n	P	n
80	-	80	53	80	27	80	16
90	-	90	73	90	36	90	23
95	-	95	92	95	46	95	30
99	-	99	128	99	68	99	45

Table 6.   Required sample size for given power and δ obtained from KS simulation with exponential distribution and $\mu_o=0$

δ=0.2		δ=0.5		δ=0.7		δ=0.9	
P	n	P	n	P	n	P	n
80	-	80	31	80	11	80	-
90	-	90	46	90	17	90	7
95	-	95	59	95	21	95	7
99	-	99	77	99	29	99	10

Appendix B: Calculating p-Values for the AD Test and Critical Values for the KS Test

The method detailed below to calculate the p-value from the k-sample AD test is drawn from reference [2]. For the case of two samples (k=2), the method begins by calculating

$$T = \frac{AD - 1}{\sigma_n} \tag{4}$$

where

$$\sigma_n = \sqrt{var(AD)} = \sqrt{\frac{aN^3 + bN^2 + cN + d}{(N-1)(N-2)(N-3)}} \tag{5}$$

with

$$a = (4g - 6)(k - 1) + (10 - 6g)H \tag{6}$$

$$b = (2g - 4)k^2 + 8hk + (2g - 14h - 4)H - 8h + 4g - 6 \tag{7}$$

$$c = (6h + 2g - 2)k^2 + (4h - 4g + 6)k + (2h - 6)H + 4h \tag{8}$$

$$d = (2h + 6)k^2 = 4hk \tag{9}$$

where

$$H = \sum_{i=1}^{k} \frac{1}{n_i}, h = \sum_{i=1}^{N-1} \frac{1}{i} \tag{10}$$

and

$$g = \sum_{i=1}^{N-2} \sum_{j=i+1}^{N-1} \frac{1}{(N-i)j} \tag{11}$$

where $N = n_1 + n_2$.

Once the value of T is obtained, the value of ln(p) can be interpolated from Table 7. The log-transformed values of p must be used since they increase linearly with T. This result is then transformed to obtain p.

Table 7.   Percentiles and Log-Transformed Percentiles of the T distribution [2]

T	0.326	0.545	1.225	1.96	2.719	3.752
P	0.25	0.2	0.1	0.05	0.025	0.01
log(P)	-1.386	-1.609	-2.303	-2.996	-3.689	-4.605

The critical value for the two-sample KS test is

$$D_{mn} = c(\alpha) * \sqrt{\frac{n_1 + n_2}{n_1 n_2}} \tag{12}$$

with values of $c(\alpha)$ shown in Table 8.

Table 8.   KS critical value parameters for various levels of significance

c(0.2)	1.07
c(0.1)	1.22
c(0.05)	1.36
c(0.01)	1.63

Appendix C: Matlab Simulation Code

```matlab
clear all
close all
clc
format long g


numsamp=2;
variance=1;
w=1;

for s= [4:.1:5]

clearvars -EXCEPT s ADPower KSPower numsamp variance w ratio

for ssize=4:2:150

for k=1:10000


x1=normrnd(s,variance,ssize,1);
x1=sort(x1);
x2=normrnd(5,variance,ssize,1);
x2=sort(x2);
xtot=sort([x1;x2]);
xtotu=unique(sort([x1;x2]));
l=length(unique(sort([x1;x2])));

n=length(x1);
m=length(x2);
tot=n+m;
smallh=0;
countf=0;
countg=0;
critical=1.07 * sqrt((n+m)/(n*m));


for i=1:(length(xtotu)-1)
```

```matlab
smallh=0;
countf=0;
countg=0;
bighcount=0;
bigfcount=0;
biggcount=0;

%%%%%%%%%%%%%%%%%%%%%%%% start AD test
 for j=1:length(xtot)

  if xtotu(i)==xtot(j)
   smallh=smallh+1;
  end

 end

 for j=1:length(xtot)

  if xtotu(i)<xtot(j)

   bighcount=bighcount + 1;

  end

 end

 bigh=bighcount + .5*smallh;


 for j=1:length(x1)

  if xtotu(i)==x1(j)
   countf=countf+1;
  end

 end

 for j=1:length(x1)

  if xtotu(i)<x1(j)

   bigfcount=bigfcount + 1;

  end

 end
```

```matlab
bigf= bigfcount + .5*countf;


 for j=1:length(x2)

  if xtotu(i)==x2(j)
   countg=countg+1;
  end

 end

 for j=1:length(x2)

  if xtotu(i)<x2(j)

    biggcount=biggcount + 1;

  end

 end

 bigg= biggcount + .5*countg;


 ff(i,1)=smallh * (( (tot)*bigf -length(x1)*bigh)^2) / (bigh*(tot
- bigh) - .25*smallh*tot);
 gg(i,1)=smallh * (( (tot)*bigg -length(x2)*bigh)^2) / (bigh*(tot
- bigh) - .25*smallh*tot);

end

 A2= (tot-1)/(tot^2) * [(1/length(x1)) * sum(ff) + (1/length(x2))
* sum(gg)];

g = 0;

for r=1:(tot-2)

 for v=(r + 1):(tot - 1)

  g = g + (1 / ((tot - r) * v));

 end

end

T = 0;
```

```matlab
for d = 1:(tot - 1)

 T = T + (1 / d);

end


S = (1/n) + (1/m);

a = (4 * g - 6) * (numsamp - 1) + (10 - 6 * g) * S;
b = (2 * g - 4) * numsamp ^ 2 + 8 * T * numsamp + (2 * g - 14 * T
- 4) * S - 8 * T + 4 * g - 6;
c = (6 * T + 2 * g - 2) * numsamp ^ 2 + (4 * T - 4 * g + 6) *
numsamp + (2 * T - 6) * S + 4 * T;
d = (2 * T + 6) * numsamp ^ 2 - 4 * T * numsamp;

sigma = ((a * tot ^ 3 + b * tot ^ 2 + c * tot + d) / ((tot - 1) *
(tot - 2) * (tot - 3) * (numsamp - 1) ^ 2)) ^ 0.5;
critval20 = 1 + sigma * (0.877 - 0.08 / ((numsamp - 1) ^ 0.5) -
0.171666 / (numsamp - 1));

%%%%%%%%%%%%%%%T values from table
T25 = 0.326;
T20 = 0.625666;
T10 = 1.225;
T05 = 1.96;
T025 = 2.719;
T01 = 3.752;

T = (A2 - 1) / sigma;

%%%%%%%%%%%%%%%%%%%%%%log transformed P values
P25 = -1.386;
p20 = -1.609;
p10 = -2.303;
p05 = -2.996;
p025 = -3.689;
p01 = -4.605;

if T < T20 & T > T25

 P = (((T - T25) * (p20 - P25)) / (T20 - T25)) + P25;

elseif T < T10 & T > T20

 P = (((T - T20) * (p10 - p20)) / (T10 - T20)) + p20;

elseif T < T05 & T > T10
```

```matlab
 P = (((T - T10) * (p05 - p10)) / (T05 - T10)) + p10;

elseif T < T025 & T > T05

 P = (((T - T05) * (p025 - p05)) / (T025 - T05)) + p05;

elseif T < T01 & T > T025

 P = (((T - T025) * (p01 - p025)) / (T01 - T025)) + p025;

elseif T < T25

 P = ((P25 - p20) / (T25 - T20)) * (T - T25) + P25;

elseif T > T01

 P = ((p01 - p025) / (T01 - T025)) * (T - T01) + p01;

else
 P = 9;

end

A2;
P=exp(P);
Pall(k,1)=P;

counter=0;
for v=1:k

 if Pall(v,1)<.20
  counter=counter+1;
 end
end

ADPower(ssize-3,w,1)=counter/k;


%%%%%%%%%%%%%%%%%%%%%%%% start KS test
for ii=1:length(xtot)

 for r=1:n
  if xtot(ii)==x1(r)
   cdf1(ii)=(1/n);
   break
  else
   cdf1(ii)=0;
```

```
  end
 end

 for r=1:m
  if xtot(ii)==x2(r)
   cdf2(ii)=(1/m);
   break
  else
   cdf2(ii)=0;
  end
 end

end

for i=2:length(cdf1)

 cdf1(i)=cdf1(i)+cdf1(i-1);

end

for i=2:length(cdf2)

 cdf2(i)=cdf2(i)+cdf2(i-1);

end

ks=max(abs([cdf1-cdf2]));

mmax(k,1)=ks;

end

counter=0;
for v=1:k

 if mmax(v,1)>critical
  counter=counter+1;
 end
end

KSPower(ssize-3,w)=counter/k;
ratio(w,1)=abs(s-4)/sqrt(variance);


end

w=w+1;
s
```

```
end


ADPower(2:2:end,:)=[];
KSPower(2:2:end,:)=[];

index=4:2:150;
contourf(flipud(ratio),index,ADPower)
contourf(flipud(ratio),index,KSPower)
```

This page intentionally left blank.

Appendix D: VBA Code for AD and KS Tests

```
Sub adtest()

'''''''''''''''''''''''''''''''''''''
'IMPORTANT NOTE:  INSERT SAMPLE DATA IN COLUMNS STARTING AT A3 AND B3
'''''''''''''''''''''''''''''''''''''

Dim LastRow1 As Double
Dim LastRow2 As Double
Dim lastrowtot As Double
Dim r1 As Range
Dim r2 As Range
Dim x1()
Dim x2()
Dim xtot()
Dim x1length As Double
Dim x2length As Double
Dim xtotlength As Double
Dim smallh As Double
Dim countf As Double
Dim countg As Double
Dim bighcount As Double
Dim bigfcount As Double
Dim biggcount As Double
Dim ff()
Dim gg()
Dim g As Double
Dim S As Double
Dim T As Double
Dim a As Double
Dim b As Double
Dim c As Double
Dim d As Double
Dim sigma As Double
Dim significance As Double
Dim P As Double
Dim cdf1()
Dim cdf2()
Dim dcdf()

ActiveSheet.Range("H3:H9999").ClearContents
ActiveSheet.Range("I3:I9999").ClearContents
```

```
numsamp = 2

With ActiveSheet
  LastRow1 = .Cells(.Rows.Count, "A").End(xlUp).Row
  x1 = Range("A3:A" & LastRow1).Value

End With

With ActiveSheet
  LastRow2 = .Cells(.Rows.Count, "B").End(xlUp).Row
  x2 = Range("B3:B" & LastRow2).Value

End With

x1length = Application.CountA(x1)
x2length = Application.CountA(x2)
xtotlength = x1length + x2length

x1 = Application.Transpose(x1)
x2 = Application.Transpose(x2)
xtot = x1

i = 1

While i <= x2length

 ReDim Preserve xtot(1 To x1length + i)
 xtot(i + x1length) = x2(i)
 i = i + 1

Wend

xtot = Application.Transpose(xtot)
Sheets("AD test").Range("H3:H" & xtotlength + 2).Value = xtot
Range("H3:H"   &   xtotlength   +   2).Sort   key1:=Range("H3"),
order1:=xlAscending, Header:=xlNo
xtot = Range("H3:H" & xtotlength + 2).Value
xtot = Application.Transpose(xtot)
ActiveSheet.Range("H2:H"   &   (2   +   xtotlength)).AdvancedFilter
Action:=xlFilterCopy,        CopyToRange:=ActiveSheet.Range("I2"),
Unique:=True

With ActiveSheet
  lastrowtot = .Cells(.Rows.Count, "I").End(xlUp).Row
  xtotu = Range("I3:I" & lastrowtot).Value


End With
xtotulength = Application.CountA(xtotu)
xtotu = Application.Transpose(xtotu)
'MsgBox xtotlength

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
i = 1
```

```
While i <= (xtotulength - 1)


ReDim Preserve ff(1 To i)
ReDim Preserve gg(1 To i)

smallh = 0
countf = 0
countg = 0
bighcount = 0
bigfcount = 0
biggcount = 0


 For j = 1 To xtotlength

  If xtotu(i) = xtot(j) Then

   smallh = smallh + 1

  End If

 Next


 For o = 1 To xtotlength

  If xtotu(i) < xtot(o) Then
   bighcount = bighcount + 1

  End If

 Next

 bigh = bighcount + 0.5 * smallh


 For r = 1 To x1length

  If xtotu(i) = x1(r) Then
   countf = countf + 1
  End If

 Next

 For v = 1 To x1length

  If xtotu(i) < x1(v) Then
   bigfcount = bigfcount + 1
  End If

 Next
```

```
 bigf = bigfcount + 0.5 * countf

 For w = 1 To x2length

  If xtotu(i) = x2(w) Then
   countg = countg + 1
  End If

 Next

 For q = 1 To x2length

  If xtotu(i) < x2(q) Then
   biggcount = biggcount + 1

  End If

 Next

 bigg = biggcount + 0.5 * countg


 ff(i) = smallh * ((xtotlength * bigf - x1length * bigh) ^ 2 / (bigh
* (xtotlength - bigh) - 0.25 * smallh * xtotlength))
 gg(i) = smallh * ((xtotlength * bigg - x2length * bigh) ^ 2 / (bigh
* (xtotlength - bigh) - 0.25 * smallh * xtotlength))

i = i + 1

Wend

A2 = ((xtotlength - 1) / (xtotlength ^ 2)) * ((1 / x1length) *
Application.WorksheetFunction.Sum(ff)   +   (1   /   x2length)   *
Application.WorksheetFunction.Sum(gg))

'''''''''''''''''''''''''''''''' critical value
g = 0

For i = 1 To (xtotlength - 2)

 For j = (i + 1) To (xtotlength - 1)

  g = g + (1 / ((xtotlength - i) * j))

 Next

Next

T = 0

For d = 1 To (xtotlength - 1)

 T = T + (1 / d)
```

```
Next

S = 0
S = 1 / x1length + 1 / x2length

a = (4 * g - 6) * (numsamp - 1) + (10 - 6 * g) * S
b = (2 * g - 4) * numsamp ^ 2 + 8 * T * numsamp + (2 * g - 14 * T -
4) * S - 8 * T + 4 * g - 6
c = (6 * T + 2 * g - 2) * numsamp ^ 2 + (4 * T - 4 * g + 6) * numsamp
+ (2 * T - 6) * S + 4 * T
d = (2 * T + 6) * numsamp ^ 2 - 4 * T * numsamp




sigma = ((a * xtotlength ^ 3 + b * xtotlength ^ 2 + c * xtotlength +
d) / ((xtotlength - 1) * (xtotlength - 2) * (xtotlength - 3) *
(numsamp - 1) ^ 2)) ^ 0.5
critval25 = 1 + sigma * (0.675 - 0.245 / ((numsamp - 1) ^ 0.5) -
0.105 / (numsamp - 1))
critval20 = 1 + sigma * (0.877 - 0.08 / ((numsamp - 1) ^ 0.5) -
0.171666 / (numsamp - 1))
critval10 = 1 + sigma * (1.281 + 0.25 / ((numsamp - 1) ^ 0.5) - 0.305
/ (numsamp - 1))
critval05 = 1 + sigma * (1.645 + 0.678 / ((numsamp - 1) ^ 0.5) -
0.362 / (numsamp - 1))
critval025 = 1 + sigma * (1.96 + 1.149 / ((numsamp - 1) ^ 0.5) -
0.391 / (numsamp - 1))
critval01 = 1 + sigma * (2.326 + 1.822 / ((numsamp - 1) ^ 0.5) -
0.396 / (numsamp - 1))


'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''P
value computation

''''''''''''''''''T values from table
T25 = 0.326
T20 = 0.625666
T10 = 1.225
T05 = 1.96
T025 = 2.719
T01 = 3.752

T = (A2 - 1) / sigma

'''log transformed P values
P25 = -1.386
p20 = -1.609
p10 = -2.303
p05 = -2.996
p025 = -3.689
p01 = -4.605
```

```
If T < T20 And T > T25 Then

 P = (((T - T25) * (p20 - P25)) / (T20 - T25)) + P25

ElseIf T < T10 And T > T20 Then

 P = (((T - T20) * (p10 - p20)) / (T10 - T20)) + p20

ElseIf T < T05 And T > T10 Then

 P = (((T - T10) * (p05 - p10)) / (T05 - T10)) + p10

ElseIf T < T025 And T > T05 Then

 P = (((T - T05) * (p025 - p05)) / (T025 - T05)) + p05

ElseIf T < T01 And T > T025 Then

 P = (((T - T025) * (p01 - p025)) / (T01 - T025)) + p025

ElseIf T < T25 Then

 P = ((P25 - p20) / (T25 - T20)) * (T - T25) + P25

ElseIf T > T01 Then

 P = ((p01 - p025) / (T01 - T025)) * (T - T01) + p01


Else: P = 9


End If

P = Exp(P)

''''''''''''''''''''''''''''''

MsgBox "Anderson Darling Test Results" & vbNewLine & vbNewLine &
"Test statistic value: " & A2 & vbNewLine & vbNewLine & "Critical
value (0.25 significance): " & critval25 & vbNewLine & "Critical
value (0.20 significance): " & critval20 & vbNewLine & "Critical
value (0.10 significance): " & critval10 & vbNewLine & "Critical
value (0.05 significance): " & critval05 & vbNewLine & "Critical
value (0.025 significance): " & critval025 & vbNewLine & "Critical
value (0.01 significance): " & critval01 & vbNewLine & vbNewLine & "P
value = " & Format(P, "0.0000000000000")


''''''''''''''''''''''''''''''

If A2 < critval25 Then
 h25 = "accept"
```

```
 Else
  h25 = "reject"

End If

If A2 < critval20 Then
 h20 = "accept"
 Else
 h20 = "reject"

End If

If A2 < critval10 Then
 h10 = "accept"
 Else
 h10 = "reject"

End If

If A2 < critval05 Then
 h05 = "accept"
 Else
 h05 = "reject"

End If

If A2 < critval025 Then

 h025 = "accept"
 Else
 h025 = "reject"

End If

If A2 < critval01 Then

 h01 = "accept"
 Else
 h01 = "reject"

End If


MsgBox "Anderson Darling Test Results" & vbNewLine & vbNewLine & "At
0.25 significance, " & h25 & " the null hypothesis." & vbNewLine &
"At 0.20 significance, " & h20 & " the null hypothesis." & vbNewLine
& "At 0.10 significance, " & h10 & " the null hypothesis." &
vbNewLine & "At 0.05 significance, " & h05 & " the null hypothesis."
& vbNewLine & "At 0.025 significance, " & h025 & " the null
hypothesis." & vbNewLine & "At 0.01 significance, " & h01 & " the
null hypothesis."

'''''''''''''' start KS test
For ii = 1 To xtotlength
```

```
 For r = 1 To x1length
  If xtot(ii) = x1(r) Then
   ReDim Preserve cdf1(1 To ii)
   cdf1(ii) = (1 / x1length)
   Exit For
  Else
   ReDim Preserve cdf1(1 To ii)
   cdf1(ii) = 0
  End If
 Next r

 For r = 1 To x2length
  If xtot(ii) = x2(r) Then
   ReDim Preserve cdf2(1 To ii)
   cdf2(ii) = (1 / x1length)
   Exit For
  Else
   ReDim Preserve cdf2(1 To ii)
   cdf2(ii) = 0
  End If
 Next r

Next ii


For i = 2 To Application.CountA(cdf1)

 cdf1(i) = cdf1(i) + cdf1(i - 1)

Next i


For i = 2 To Application.CountA(cdf2)

 cdf2(i) = cdf2(i) + cdf2(i - 1)

Next i

For i = 1 To Application.CountA(cdf1)

 ReDim Preserve dcdf(1 To i)
 dcdf(i) = Abs(cdf1(i) - cdf2(i))

Next i

ks = Application.Max(dcdf)

critical20 = 1.07 * ((x1length + x2length) / (x1length * x2length)) ^
0.5
critical10 = 1.22 * ((x1length + x2length) / (x1length * x2length)) ^
0.5
critical05 = 1.36 * ((x1length + x2length) / (x1length * x2length)) ^
0.5
```

```
critical01 = 1.63 * ((x1length + x2length) / (x1length * x2length)) ^
0.5

If ks < critical20 Then
 h20 = "accept"
 Else
 h20 = "reject"

End If

If ks < critical10 Then
 h10 = "accept"
 Else
 h10 = "reject"

End If

If ks < critical05 Then
 h05 = "accept"
 Else
 h05 = "reject"

End If

If ks < critical01 Then
 h01 = "accept"
 Else
 h01 = "reject"

End If


MsgBox "KS Test Results" & vbNewLine & vbNewLine & "Test statistic
value: " & ks & vbNewLine & vbNewLine & "At 0.20 significance, " &
h20 & " the null hypothesis." & vbNewLine & "At 0.10 significance, "
& h10 & " the null hypothesis." & vbNewLine & "At 0.05 significance,
" & h05 & " the null hypothesis." & vbNewLine & "At 0.01
significance, " & h01 & " the null hypothesis."


End Sub
```

This page intentionally left blank.

References

[1] R. B. D'Agostino, M. A. Stephens. "Goodness-of-Fit Techniques", Marcel Dekker, New York, 1986, pp.97-101.

[2] F. W. Scholz and M. A. Stephens. "K-Sample Anderson-Darling Tests", Journal of the American Statistical Association, 1987, Vol. 82, 339

[3] M. H. Gail and S. B. Green. "Critical Values for the One-Sided Two-Sample Kolmogorov-Smirnov Statistic", Journal of the American Statistical Association, 1976, Vol. 71, 355

[4] P. J. Kim and R. I. Jennich, "Table of the Exact Sampling Distribution of the Kolmogorov-Smirnov Criterion", Selected Table in Mathematical Statistics I, Markham Publishing, Chicago, 1970, pp. 79-170

This page intentionally left blank.

CNA is a not-for-profit research organization
That serves the public interest by providing
in-depth analysis and result-oriented solutions
to help government leaders choose
the best course of action
in setting policy and managing operations.

Nobody gets closer—
to the people, to the data, to the problem.