IID Testing in SP 800 90B

Meltem Sonmez Turan
meltem.turan@nist.gov
Random Bit Generation Workshop 2016
National Institute of Standards and Technology

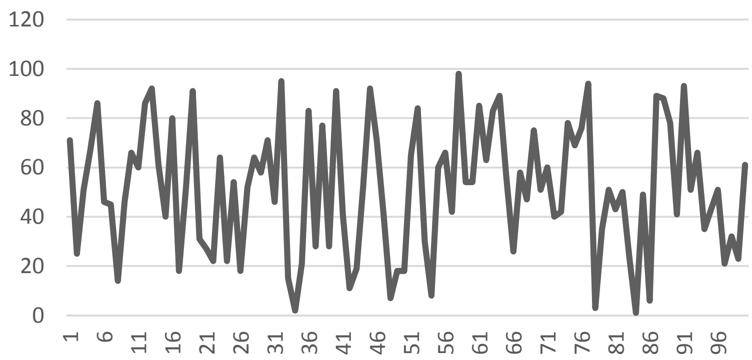
What is the IID Assumption?

Critical assumption in statistics, machine learning theory, entropy estimation, etc.

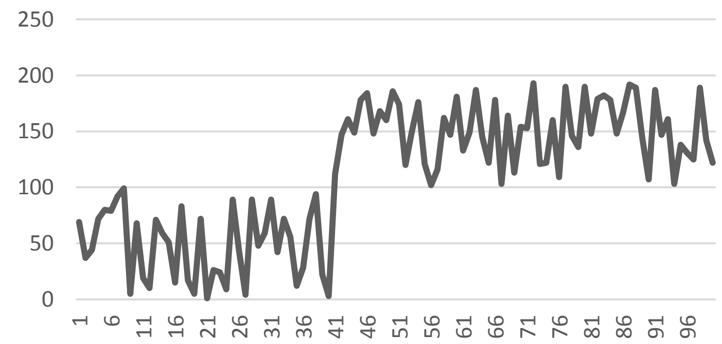
In probability theory, a collection of random variables is independent and identically distributed (IID or *i.i.d.*), if

- each sample has the same probability distribution as every other sample, and
- all samples are mutually independent.

Examples: dice rolls, coin flips



IID - Uniformly distributed.



Non-IID behavior.

Why is IID testing important for SP 800-90B?

SP 800-90B has two tracks for entropy estimation:

- *IID track:* If the noise source is IID, the entropy is estimated using the *most common value* estimate.
- *Non-IID track:* If the noise source is not IID, the entropy estimation is more complex. We use ten estimators.

Determining the track:

The track is IID only if *all* of the conditions are satisfied;

1. The following datasets are tested, and the IID assumption is verified
    - *Sequential dataset*
    - *Row* and *column datasets*
    - *Conditioned sequential dataset* (if a non-vetted conditioning component is used).
2. IID claim by the submitter

IID Testing

Input: The sequence $S=(s_1,\ldots,s_L)$ where $s_i \in A = \{x_1,\ldots,x_k\}$ and $L \geq 1{,}000{,}000$.

Output: Decision regarding the IID assumption: *The samples are not IID* OR *There is no evidence that data is not IID*.

Two types of tests:

1. *Permutation testing (shuffling tests):* based on test statistics with unknown distributions.
2. *Chi-square tests:* based on test statistics with approximated distributions.

If the hypothesis is rejected by any of the tests, the values in S are assumed to be non-IID.

Permutation Testing

Input sequence S → Test statistics T

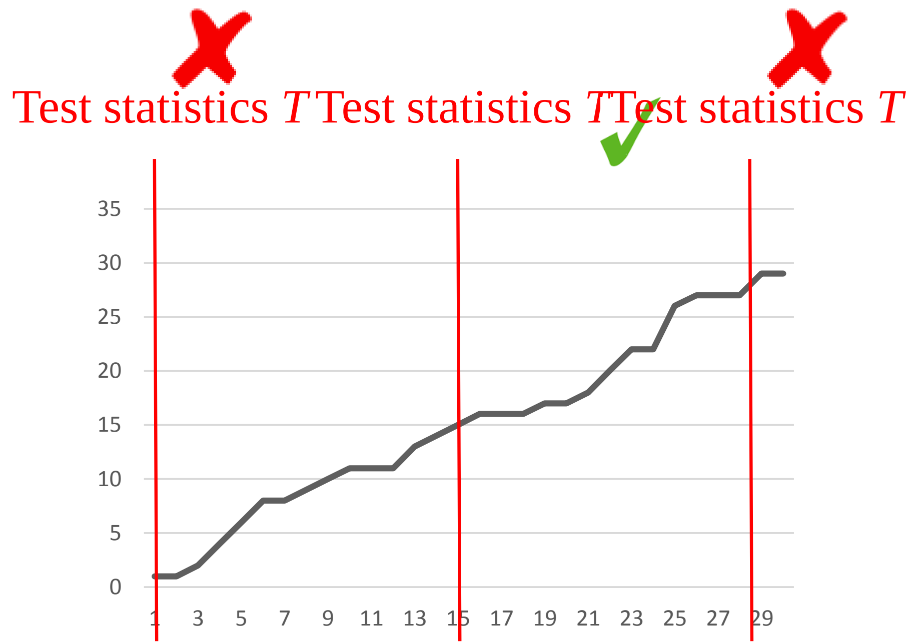
Shuffled S → T_1

Shuffled S → T_2

Shuffled S → T_3

...

Shuffled S → $T_{10,000}$



Permutation Testing

Input: $S = (s_1, \ldots, s_L)$

Output: Decision on the IID assumption

Assign the counters C_0 and C_1 to zero.

Calculate the test statistic T on S: denote the result as t.

For j = 1 to 10,000

- Permute S using the Fisher-Yates shuffle algorithm.
- Cal
- If (t

- If ((C_0+
  noise so

Input: $S = (s_1, \ldots, s_L)$

Output: Shuffled $S = (s_1, \ldots, s_L)$

1. $i = L$
2. While ($i \geq 1$)
   a. Generate a random integer j that is uniformly distributed between 0 and i.
   b. Swap s_j and s_i

$i = i - 1$

Test statistics for Permutation Testing

Eleven test statistics:

1. Excursion
2. Number of directional runs
3. Length of directional runs
4. Number of increases and decreases
5. Number of runs based on the median
6. Length of runs based on median
7. Average collision
8. Maximum collision
9. Periodicity (5 parameters)
10. Covariance (5 parameters)
11. Compression

Binary vs. non-binary samples

The number of distinct sample values, (size of A), significantly affects the distribution of the test statistics.

Two conversions for binary data:

- *Conversion I* partitions the sequences into 8-bit non-overlapping blocks, and counts the number of ones in each block.

  S = (1,0,0,0,1,1,1,0,1,1,0,1,1,0,1,1,0,0,1,1) becomes (4, 6, 2).

- *Conversion II* partitions the sequences into 8-bit non-overlapping blocks, and calculates the integer value of each block.

  S = (1,0,0,0,1,1,1,0, 1,1,0,1,1,0,1,1,0,0,1,1) becomes (142, 219, 48).

1. Excursion Test Statistics

Based on how far the running sum of sample values deviates from its average at each point in the dataset.

Pseudocode:

1. Find $\bar{X} = (s_1 + s_2 + \ldots + s_L) / L$.

2. For $i = 1$ to L, find

$$d_i = \left| \sum_{j=1}^{i} s_j - i \times \bar{X} \right|.$$

3. $T = \max(d_1, \ldots, d_L)$.

Example:

Let $S = (2, 15, 4, 10, 9)$.

The average = 8.

$d_1 = |2-8| = 6$

$d_2 = |(2+15) - (2\times8)| = 1$

$d_3 = |(2+15+4) - (3\times8)| = 3$

$d_4 = |(2+15+4+10) - (4\times8)| = 1$

$d_5 = |(2+15+4+10+9) - (5\times8)| = 0$

$T = \max(6, 1, 3, 1, 0) = 6$.

2. Number of Directional Runs

Based on the number of runs constructed using the relations between consecutive samples.

Pseudocode:

1. Construct $S' = (s'_1, \ldots, s'_{L-1})$, where

$$s'_i = \begin{cases} -1, & \text{if } s_i > s_{i+1} \\ +1, & \text{if } s_i \leq s_{i+1} \end{cases}$$

for $i = 1, \ldots, L-1$.

2. T = # runs in S'.

Binary data: Apply Conversion *I*.

Example:

Let S = (2, 2, 2, 5, 7, 7, 9, 3, 1, 4, 4);

S' = (+1, +1, +1, +1, +1, +1, –1, –1, +1, +1).

There are three runs:

(+1, +1, +1, +1, +1, +1), (–1, –1) and (+1, +1).

T = 3.

3. Length of Directional Runs

Based on the length of the longest run constructed using the relations between consecutive samples.

Pseudocode:

1. Construct $S' = (s'_1, \ldots, s'_{L-1})$, where

$$s'_i = \begin{cases} -1, & \text{if } s_i > s_{i+1} \\ +1, & \text{if } s_i \leq s_{i+1} \end{cases}$$

for i =1, …, L-1.

2. T = length of the longest run in S'.

Binary data: Apply Conversion I.

Example:

Let S = (2, 2, 2, 5, 7, 7, 9, 3, 1, 4, 4).

S'= (+1, +1, +1, +1, +1, +1, –1, –1, +1, +1).

There are three runs:

(+1, +1, +1, +1, +1, +1), (–1, –1) and (+1, +1)

Longest run has length T = 6.

4. Number of Increases and Decreases

Based on the maximum number of increases or decreases between consecutive sample values.

Pseudocode:

1. Construct $S' = (s'_1, \ldots, s'_{L-1})$, where

$$s'_i = \begin{cases} -1, & \text{if } s_i > s_{i+1} \\ +1, & \text{if } s_i \leq s_{i+1} \end{cases}$$

for $i = 1, \ldots, L$-1.

2. T = max (number of -1's in S', number of +1's in S').

Binary data: Apply Conversion I.

Example:

Let S = (2, 2, 2, 5, 7, 7, 9, 3, 1, 4, 4).

S'= (+1, +1, +1, +1, +1, +1, –1, –1, +1, +1).

There are eight +1's and two –1's in S',

T = max (8, 2) = 8.

5. Number of Runs Based on the Median

Based on the number of runs that are constructed with respect to the median of the input data.

Pseudocode:

1. Find the median $\tilde{X}$ of S.
2. Construct $S' = (s'_1, \dots, s'_L)$ where

$$s'_i = \begin{cases} -1, & \text{if } s_i < \tilde{X} \\ +1, & \text{if } s_i \geq \tilde{X} \end{cases}$$

for $i = 1, \dots, L$.

3. T = # runs in S'.

Binary data: The median is assumed to be 0.5.

Example:

Let $S = (5, 15, 12, 1, 13, 9, 4)$.

The median is 9.

$S' = (-1, +1, +1, -1, +1, +1, -1)$.

There are five runs: (-1), $(+1, +1)$, (-1), $(+1, +1)$, and (-1).

$T = 5$

6. Length of Runs Based on Median

Based on the length of the longest run that is constructed with respect to the median of the input data.

Pseudocode:

1.Find the median $\tilde{X}$ of $S = (s_1, \ldots, s_L)$.

2.Construct $S' = (s'_1, \ldots, s'_L)$

$$s'_i = \begin{cases} -1, & \text{if } s_i < \tilde{X} \\ +1, & \text{if } s_i \geq \tilde{X} \end{cases}$$

for $i = 1, \ldots, L$.

3. T = length of the longest run S'.

Binary data: The median of the input data is assumed to be 0.5.

Example:

Let $S = (5, 15, 12, 1, 13, 9, 4)$.

The median is 9.

$S' = (-1, +1, +1, -1, +1, +1, -1)$.

Runs: (-1), $(+1, +1)$, (-1), $(+1, +1)$, and (-1).

The length of longest run is 2; T =2.

7. Average Collision Test Statistics

Based on the number of successive sample values until a duplicate is found.

Pseudocode:

1. C is an empty list. $i = 1$.
2. While $i < L$,

   Find the smallest j such that $(s_i, \ldots, s_{i+j-1})$ contains two identical values. If no such j exists, break.

   Add j to the list C.

   $i = i + j + 1$

3. T = average of all values in C.

Binary data: Apply Conversion II.

Example:

Let $S = (2, 1, 1, 2, 0, 1, 0, 1, 1, 2)$.

The first collision occurs for $j = 3$. Add 3 to C.

In remaining sequence (2, 0, 1, 0, 1, 1, 2), next collision occurs for $j = 4$. Add 4 to C.

The third sequence is (1,1,2), and $j = 2$.

$C = [3,4,2]$. The average is 3, $T = 3$.

8. Maximum Collision Test Statistics

Based on the number of successive sample values until a duplicate is found.

Pseudocode:

1. C is an empty list. $i = 1$

3. While $i < L$

   Find the smallest j such that $(s_i, \ldots, s_{i+j-1})$ contains two identical values.

   If no such j exists, break.

   Add j to the list C.

   $i = i + j + 1$

4. T = the maximum value in the list C.

Binary data: Apply Conversion II.

Example:

Let S= (2, 1, 1, 2, 0, 1, 0, 1, 1, 2).

C = [3,4,2] is computed as in previous example.

T = max(3,4,2) = 4

9. Periodicity Test Statistics

Based on the periodic relations in the data. The test takes a lag parameter p as input.

The test is repeated for five different values of p: 1, 2, 8, 16, and 32.

Example:

Let S = (2, 1, 2, 1, 0, 1, 0, 1, 1, 2), and let $p = 2$.

Since $s_i = s_{i+p}$ for five values of i (1, 2, 4, 5 and 6)

$T = 5$

Pseudocode:

1. Initialize T to zero.
2. For $i = 1$ to $L - p$

   If ($s_i = s_{i+p}$), increment T by one.

Binary data: Apply Conversion I.

10. Covariance Test Statistics

Based on the strength of the lagged correlation.

Pseudocode:

1. Initialize T to zero.

2. For $i = 1$ to $L - p$

$$T = T + (s_i \times s_{i+p})$$

Handling Binary data: Apply Conversion I.

The test is repeated for five values of p: 1, 2, 8, 16, and 32.

Previous version:

$T = T + (s_i - \mu)(s_{i-1} - \mu)$, where μ = mean.

Example:

Let S = (5, 2, 6, 10, 12, 3, 1).

Let p = 2.

T is calculated as $(5\times6) + (2\times10) + (6\times12) + (10\times3) + (12\times1) = 164$.

11. Compression Test Statistics

Based on the size of the data subset after the samples are encoded into a character string and processed by a general-purpose compression

Pseudocode:

1. Encode the input data as a character string containing a list of values separated by a single space, e.g., "S = (144, 21, 139, 0, 0, 15)" becomes "144 21 139 0 0 15".

2. Compress the character string with the bzip2 compression algorithm.

3. T = length of the compressed string, in bytes.

Additional Chi-Square Statistical Tests

1. Testing independence for non-binary data
2. Testing goodness-of-fit for non-binary data
3. Testing independence for binary data
4. Testing goodness-of-fit for binary data
5. Length of the Longest Repeated Substring (LRS) Test

Testing independence for non-binary data

Based on the frequencies of pairs.

Pseudocode:

1. Find the proportion p_i of each x_i in S.
2. Calculate expected # of occurrences of pairs. $e_{i,j} = p_i p_j (L-1)$
3. Allocate (i,j) pairs into bins.
4. Apply the chi-square test.

Example:

Let S = (2, 2, 3, 1, 3, 2, 3, 2, 1, 3, 1, 1, 2, 3, 1, 1, 2, 2, 2, 3, 3, 2, 3, 2, 3, 1, 2, 2, 3, 3, 2, 2, 2, 1, 3, 3, 3, 2, 3, 2, 1, 3, 2, 3, 1, 2, 2, 3, 1, 1, 3, 2, 3, 2, 3, 1, 2, 2, 3, 3, 2, 2, 2, 1, 3, 3, 3, 2, 3, 2, 1, 2, 2, 3, 3, 3, 2, 3, 2, 1, 2, 2, 2, 1, 3, 3, 3, 2, 3, 2, 1, 3, 2, 3, 1, 2, 2, 3, 1, 1), L=100.

$A=\{1, 2, 3\}$; $p_1=0.21$, $p_2=0.41$ and $p_3=0.38$.

Bin	Pairs	Exp	Obs.
1	(1,1) (1,3)	12.39	13
2	(3,1)	7.98	9
3	(1,2)	8.61	8
4	(2,1)	8.61	8
5	(3,3)	14.44	10
6	(2,3)	15.58	19
7	(3,2)	15.58	18
8	(2,2)	16.81	14

Test statistics=3.20 < 23.322.
Not rejected!

Testing goodness-of-fit for non-binary data

Based on the frequencies of samples in different parts of the input.

Pseudocode:

1. c_i = # of x_i in S. $e_i = c_i/10$.
2. Construct a chi-square table based on expected values, starting from smallest.
3. Partition the input sequence into 10 non-overlapping parts and apply the chi-square test with 9 (#bins – 1).

Example:

Let $A=\{1, 2, 3\}$, and let $c_1=43$, $c_2=55$, $c_3=52$, $c_4=10$.

$e_1=4.3$, $e_2=5.5$, $e_3=5.2$, $e_4=1$.

30 bins,

Bin	Pairs	Exp	Obs.
1	1, 4	5.3	7
2	2	5.5	7
3	3	5.2	1
4	1, 4	5.3	5
5	2	5.5	3
6	3	5.2	8
…	…	…	…
30	3	5.2	2

Test statistics=37.08 < 42.312.
Not rejected!

Testing independence for binary data

Based on the independence between adjacent bits.

Pseudocode:

1. p_0, p_1:proportion of zeroes and ones.
2. For each P=(a_1,a_2,…, a_m),

   o = # of occurrences P in S.

   e= expected number of P in S, *based on* p_0, p_1.

   T=T + $\frac{(o-e)^2}{e}$.

Example:

Let S = (1,1,0,1,0,1,1,0,1,1,1,1,0,0,1,1, 0,0,1,0,0,0,1,0,1,1,0,0,1,1).

$p_0 = \frac{17}{30}, p_1 = \frac{13}{30}, m = 2$

Bin	Pairs	Exp	Obs.
1	(0,0)	9.32	5
2	(0,1)	7.12	8
3	(1,0)	7.12	8
4	(1,1)	5.44	8

Test statistics=3.42 < 11.345

Not rejected!

Testing goodness-of-fit for binary data

Based on the distribution of ones throughout the sequence.

Pseudocode:

1. p :proportion of ones.
2. Partition S into 10 non-overlapping subsequences S_i. For each S_i

   o = # of ones in S_i.

   $e = p \left\lfloor \frac{L}{10} \right\rfloor$.

   $T = T + \frac{(o-e)^2}{e}$.

Example: Let S = (1,1,0,1,0,1,1,0,1,1, 1,1,0,0,1,1,1,1,1,0,0,1,0,0,1,0,0,0,1,0,1, 1,0,0,1,1,0,1,0,1,0,1,1,0,1,0,1,0,1,1,1,0, 0,1,1,0,0,1,0,0,0,1,0,1,1,0,0,1,1,0,1,1,0, 1,0,1,1,0,1,1,1,1,0,0,1,1,0,0,1,1,1,1,1,0, 1,1,0,0,1,1).

$p = 0.58$.

Test statistics=3.03 < 21.666
Not rejected!

Bin	Exp	Obs.
1	5.8	7
2	5.8	7
3	5.8	3
4	5.8	6
5	5.8	6
6	5.8	4
7	5.8	5
8	5.8	7
9	5.8	6
10	5.8	7

Length of the Longest Repeated Substring Test

Based on the length of the longest repeated substring (W).

Pseudocode:

1. Collision pr. $p_{col} = \sum p_i^2$
2. Let E be a Binomially distr. r.v.

with parameters $N=\binom{L-W+1}{2}$ and $(p_{col})^W$.

3. If Pr $(E \geq 1) = 1 -$ Pr $(E = 0) = 1 - (1 - p_{col})^N$ is less than 0.001, the test fails.

Example: Let S = (1,1,0,1,0,1,1,0,1,1, 1,1,0,0,1,1,1,1,1,0,0,**1,0,0,1,0,0,0,1,0,1, 1,0,0,1,1,0,1**,0,1,0,1,1,0,1,0,1,0,1,1,1,0, 0,1,**1,0,0,1,0,0,0,1,0,1,1,0,0,1,1,0,1**,1,0, 1,0,1,1,0,1,1,1,1,0,0,1,1,0,0,1,1,1,1,1,0, 1,1,0,0,1,1).

$W = 17$

Collision probability = $0.42^2 + 0.58^2$ = 0.5128

N= 3486, $p_{col}{}^W$ = 0.000012.

Pr $(E \geq 1) = 1 - (1 - p_{col}{}^W)^N = 0.04$.

0.04 > 0.001 ! Not rejected!

Summary

- The shuffling tests were restructured; we call them permutation testing. More extensive and requires more time.
- Removed some of the tests that were not very effective (variant of directional runs and collision tests)
- Added new Periodicity test with five parameters.
- Added new parameters to the covariance test.