


7-17-2020

JMASM 53: MicceriRD

Michael Lance

ExceLance, LLC, michael.lance@gmail.com

 Part of the [Applied Statistics Commons](#), [Social and Behavioral Sciences Commons](#), and the [Statistical Theory Commons](#)

Recommended Citation

Lance, M. (2019). JMASM 53: MicceriRD. *Journal of Modern Applied Statistical Methods*, 18(2), eP3391.
doi: 10.22237/jmasm/1604190720

JMASM 53: MicceriRD

Michael Lance

ExceLance, LLC
Plymouth, MI

Fortran 77 and 90 modules (REALPOPS.lib) exist for invoking the 8 distributions estimated by Micceri (1989). These respective modules were created by Sawilowsky et al. (1990) and Sawilowsky and Fahome (2003). The MicceriRD (Micceri's Real Distributions) Python package was created because Python is increasingly used for data analysis and, in some cases, Monte Carlo simulations.

Keywords: Micceri, real distributions, Python, Monte Carlo simulation, robustness

In 1989, Theodore Micceri published "The Unicorn, the Normal Curve, and Other Improbable Creatures" with the purpose of illustrating the variety of real distributions found in real datasets that are often not normally-distributed. His article had important implications for the use of statistical procedures on non-normal datasets. When simulating data in the "real world", it helps to sample from such distributions. Thus, REALPOPS.lib was created by Sawilowsky et al. (1990) and Sawilowsky and Fahome (2003) for this very purpose. Using these, many subsequent articles have been published on what happens when data are not normal and parametric tests are used (and how accuracy and power can be improved when using non-parametric tests on data from such distributions, even with small samples).

The MicceriRD (Micceri's Real Distributions) Python package was created because Python is increasingly used for data analysis and, in some cases, Monte Carlo simulations. Arrays of samples are generated in order to produce arrays of results as this is much more efficient than generating a sample and result within each nested loop of a Monte Carlo simulation. The random number generator used in drawing samples via NumPy is from the Parallel Congruent Generator (SciPy Community, 2019) family (PCG XSL RR 128/64), which has a period of 2^{126} (O'Neill, 2014).

MicceriRD has the following functions for each of Micceri's 8 real distributions:

1. It invokes each distribution by name as a NumPy array.
2. It draws a NumPy array of k random samples of n size from each distribution, by name.

Sample Code

Installation from command prompt:

```
pip install MicceriRD
```

(current version at the time of this writing: 0.2)

Invoke the Digit Preference distribution as a NumPy array:

```
In[1]: import MicceriRD as mrd
```

```
In[2]: mrd.DigitPreference
```

```
Out[2]: array([420, 420, 425,..., 635, 635, 635])
```

Draw 4 random samples of $n = 6$ as a NumPy array from the Digit Preference distribution:

```
In[3]: mrd.Sampling.DigitPreference(4,6)
```

```
Out[3]:
```

```
array([[520, 460, 515, 465, 525, 560],  
       [585, 575, 585, 570, 475, 470],  
       [470, 550, 475, 580, 500, 590],  
       [525, 595, 535, 515, 525, 470]])
```

MicceriRD on the Python Package Index:

<https://pypi.org/project/MicceriRD/>

MicceriRD homepage:

<https://bitbucket.org/mikeronni/miccerird/>

References

Micceri, T. (1989). The unicorn, the normal curve, and other improbable creatures. *Psychological Bulletin*, 105(1), 156-166. doi: 10.1037/0033-2909.105.1.156

MICCERIRD

O'Neill, M. E. (2014, September 5). *PCG: A family of simple fast space-efficient statistically good algorithms for random number generation* (Technical report no. HMC-CS-2014-0905). Claremont, CA: Harvey Mudd College Computer Science Department. Retrieved from <https://www.cs.hmc.edu/tr/hmc-cs-2014-0905.pdf>

Sawilowsky, S. S., Blair, R. C., & Micceri, T. (1990). REALPOPS.LIB: APC Fortran library of eight real distributions in psychology and education. *Psychometrika*, 55(4), 729.

Sawilowsky, S. S., & Fahoome, G. F. (2003). *Statistics via Monte Carlo simulation with Fortran*. Rochester Hills, MI: JMASM.

SciPy Community. (2019, July 26). Parallel congruent generator (64-bit, PCG64) [NumPy routine]. Retrieved from https://docs.scipy.org/doc/numpy/reference/random/bit_generators/pcg64.html