UCL

Generating normally distributed pseudo-random numbers

David Duemig

Abstract

In this report, we examine three different methods for generating pseudo-random numbers. Specifically, we generate numbers that are distributed with the Gaussian (normal) distribution, starting from the random number generator of a computer. In order to check whether what we have generated is actually Gaussian or Pareto distributed, we apply various normality tests. Finally, to illustrate the applications of these methods, we perform a Monte Carlo simulation in which we calculate the Value-at-Risk (VaR) and Expected Shortfall (ES) for a given portfolio.

1 Introduction

We often want to generate random draws from a certain probability distribution in the course of writing code for data analysis or for simulating the dynamics of the stock market. The problem is that most programming languages are only equipped with simple random number generators that generate uniformly-distributed numbers between 0 and RAND_MAX, where RAND_MAX depends on the implementation and the language. The uniform distribution is sufficient when we need to simulate something simple, like rolling dice, but it is inadequate when more sophisticated probabilities are crucial. That is, often nonlinear distributions like the Gaussian, Poisson and exotic decisions, like the Gumbel distribution, are required.

This report describes three different methods for generating numbers that are distributed with the Gaussian (normal) distribution, based on the simple random number generator of a computer.[1]

For comparison, we also generate random samples with the ´numpy.random` function from the numpy library, which is based on the Mersenne Twister, a high-quality pseudo-random number generator.[2]

In order to quantify the probability that what we have generated comes from a normal distribution, several normality tests are introduced.

Finally, to illustrate the applications of our methods, we perform a Monte Carlo simulation to calculate the Value at Risk and Expected Shortfall of a portfolio consisting of the S&P 500 and the FTSE 100.

[1] The rand() function.
[2] The code for this report can be found in the GitHub Repository.

2 Methods

In this section, we present three different approaches to sample from a normal distribution.

2.1 Inverse Transformation Sampling

The inverse sampling algorithm allows us to generate random samples from a given probability distribution function (PDF) by using random samples from the uniform distribution and the cumulative distribution function (CDF) of the given PDF. The algorithm can be divided into two cases:

- **Case 1: CDF is invertible**

  Let F: $\mathbb{R} \to (0,1)$ be an invertible CDF (i. e. $\exists\ F^{-1}: (0,1) \to \mathbb{R}$) such that
  $F^{-1}\big(F(x)\big) = x\ \forall\ x\ \in\ \mathbb{R}$
  $F\big(F^{-1}(a)\big) = a\ \forall\ a \in (0,1)$
  Then if $U \sim Uniform(0,1)$ then $F^{-1}(U) \sim F$ (i. e. $F^{-1}(U)$ has CDF F)

- **Case 2: General Case**

  Let F: $\mathbb{R} \to (0,1)$ be a CDF and define $G: (0,1) \to \mathbb{R}$ by $G(a) = inf\{x \in \mathbb{R}: a \leq F(x)\}$

  Now, if $U \sim Uniform(0,1)$ then $G(U)$ has CDF F

The proof for both cases can be found in the appendix.

If we want to generate samples for the random variable X, let X be a random variable with PDF $N(\mu, \sigma^2)$

The PDF of X is

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

The CDF is given by

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx$$

Now the CDF of $N(\mu, \sigma^2)$ is a one-one and onto function from R to (0, 1). Thus, F is invertible, and we can apply case 1 to generate random samples of $N(\mu, \sigma^2)$.

2.2 Rejection Sampling

It is not always possible to find an explicit formula for the CDF of a random variable X we want to generate samples for. Furthermore, there may be alternative methods for generating samples with a distribution F that are more efficient than the inverse transformation method.

In this section, we present a method known as the acceptance rejection method. We assume that the distribution from which we want to sample has a probability density function $f(x)$. The basic idea is to find an alternative probability distribution with the density function $g(x)$, for which we have already developed an efficient algorithm to generate samples (e.g., inverse transform method). The function $g(x)$ should be "close" to $f(x)$; more specifically, we assume that the ratio $f(x)/g(x)$ is bounded by a constant $c > 0; sup_x\{f(x)/g(x)\} \leq c$. The complete algorithm and a proof can be found in the appendix.

If we want to generate samples for the random variable X, distributed as $N(\mu, \sigma^2)$, let $X = \sigma Z + \mu$, where Z denotes a random variable with distribution $N(0, 1)$. It is therefore sufficient to find an algorithm for generating $Z \sim N(0, 1)$. Moreover, if we can generate from the absolute value $|Z|$, then by symmetry Z can be obtained by independently generating a random variable S (for sign) that is ± 1 with a probability of 0.5 and setting $Z = S|Z|$. In other words, we generate U and set Z = $|Z|$ if $U \leq 0.5$ and set $Z = -|Z|$ if U > 0.5.

$|Z|$ is non-negative and has the following density:

$$f(x) = \frac{2}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right), x \geq 0$$

For our alternative, we choose $g(x) = e^{-x}, x \geq 0$, the exponential density with rate 1, something we already know how to sample from; e.g., using the inverse transform method. Noting that $h(x) \stackrel{\text{def}}{=} \frac{f(x)}{g(x)} = e^{x - x^2/2}\sqrt{2/\pi}$, we calculate its maximum (solve h′(x) = 0); which must occur at a value of x that maximizes the exponent.[3] Therefore,

$c = \sqrt{2e/\pi} \approx 1.32$, and so f(y)/cg(y) = $e^{-(y-1)^2/2}$

Accordingly, the algorithm for generating Z looks as follows:

1. Generate Y with an exponential distribution at rate 1; i.e. generate U and set $Y = -\ln(U)$ [4]
2. Generate U
3. If $U \leq e^{-(Y-1)^2/2}$, set $|Z| = Y$; otherwise back to 1
4. Generate U. Set $Z = |Z|$ if $U \leq 0.5$, set $Z = -|Z|$ if $U > 0.5$

2.3 Box-Muller Transformation

The usual Box-Mueller algorithm uses uniform random deviates to generate $Z \sim N(0, 1)$. In this report, we will modify the algorithm to directly produce $Z \sim N(\mu, \sigma^2)$.

Suppose u_1 and u_2 are independent random variables that are uniformly distributed on [0, 1] and let

$$z_1 = \sqrt{-2\ln(u_1)}\cos(2\pi u_2)$$
$$z_2 = \sqrt{-2\ln(u_1)}\sin(2\pi u_2)$$

Then z1 and z2 are independent random variables with a standard normal distribution. The proof that this transformation results in the desired Gaussian distribution is a special case of the proof of the modified transformation and can be found in the appendix.

The modification we make to use the Box-Mueller transformation for directly generating $Z \sim N(\mu, \sigma^2)$ is to apply the invariant relation.[5] Specifically, setting:

$$x = z\sigma + \mu$$

We have

$$x_1 = \sqrt{-2\ln(u_1)}\cos(2\pi u_2)\sigma + \mu$$
$$x_2 = \sqrt{-2\ln(u_1)}\sin(2\pi u_2)\sigma + \mu$$

[3] The exponent $x - x^2/2$ is maximized at x=1.
[4] Or -ln(1-U)
[5] Invariant relation: $G(x, \mu, \sigma) = \frac{G(y,0,1)}{\sigma}$, where $y = \frac{x-\mu}{\sigma}$.

3 Normality Tests

There are a number of techniques to determine whether what we have generated is Gaussian. In this report, we will look at two types of techniques to check whether a data sample is normally distributed.

3.1 Graphical Methods[6]

These are methods for displaying the data and for qualitatively evaluating whether the data sample looks Gaussian.

- **Histogram**
  One simple and frequently used graphical representation to check whether our data sample is normally distributed is the histogram. A Gaussian distribution is indicated when the histogram shows the well-known bell shape.

- **Box Plot**
  If our data sample is drawn from a Gaussian distribution, the box will be symmetrical with the mean and median in the center. Moreover, if the data sample corresponds to the assumption of normality, we should observe only a few outliers.

- **Probability-Probability Plot**
  The probability-probability plot, or P-P plot for short, is a probability plot that can be used to assess how closely our data sample agrees with the normal distribution. A P-P plot is a scatterplot created by plotting the probabilities of the sample distribution and the probabilities of the theoretical distribution against each other. The resulting points are displayed with the theoretical value on the x-axis and the data sample on the y-axis. If both sets of quantiles came from the same distribution, we should observe a line of points at an angle of 45 degrees from the bottom left to the top right.

- **Quantile-Quantile Plot**
  The quantile-quantile plot, or Q-Q plot for short, is similar to the P-P plot and is another graphical method we use to determine whether our data sample comes from the normal distribution. A Q-Q plot is a scatterplot created by plotting the quantiles of the sample distribution and the quantiles of the theoretical distribution against each other. The resulting points are displayed with the theoretical value on the x-axis and the data sample on the y-axis. If both sets of quantiles came from the same distribution, we should observe a line of points at an angle of 45 degrees from the bottom left to the top right.

- **Empirical CDF Plot**
  An empirical CDF plot has a similar function to a P-P plot. However, unlike a probability plot, the empirical CDF plot has scales that are not transformed, and the fitted distribution does not form a straight line, but an S-shaped curve under normality. The empirical cumulative probabilities close to this S-shaped curve fulfill the normality assumption.

3.2 Statistical Tests

These are methods to quantify the probability that the data sample was drawn from a normal distribution. There are numerous statistical tests that can be applied, with each test making different assumptions and considering different aspects of the data. All of these tests return at least two things:

[6] The graphical representations for our data samples can be found in the GitHub Repository.

Statistic
A quantity calculated by the test that can be compared to critical values from the distribution of the test statistic.

p-value
A value calculated by the test that can be used to interpret the test; in our case, to check whether the data sample was taken from a normal distribution.

The tests assume that the data sample comes from a Gaussian distribution; technically, this assumption is called the null hypothesis, or H0. A threshold value, called alpha, of typically 5% is chosen, which is used to interpret the p-value. More specifically, the p-values can be interpreted as follows:

p <= alpha: reject H0; i.e., not normally distributed
p > alpha: fail to reject H0; i.e., normally distributed

Thus, we generally aim for results with a higher p-value to confirm that our data sample is likely to be normally distributed. However, a result above 5% does not mean that the null hypothesis is true; it only means that it is very likely, given the available evidence. Moreover, the p-value is not the probability that the data sample corresponds to a Gaussian distribution, but can be seen as a value that helps to interpret the statistical test.

3.2.1 Empirical Distribution Function (EDF) Tests[7]

Based on the measure of discrepancy between empirical and theoretical distribution, we can define the following tests:

- **Kolmogorov-Smirnov Test**
  The Kolmogorov-Smirnov test (KS test) was first derived by Kolmogorov (1933) and later modified by Smirnov (1948). It is a statistical test for the agreement of two probability distributions. More specifically, it compares the cumulative distribution of the sample data with the expected cumulative normal distribution and bases its p-value on the largest discrepancy. The KS test can be modified to serve as a normality test; in this case, the samples are standardized and compared to a standard normal distribution.

- **Anderson-Darling Test**
  The Anderson-Darling test is another statistical test that can be used to determine whether the distribution of the data sample deviates from the Gaussian distribution. The test is named after Theodore Wilbur Anderson and Donald Allan Darling, who first described it in 1952. A feature of this test is that it provides a list of critical values, not just a single p-value, which allows a more thorough interpretation of the result.

- **Shapiro-Wilk Test**
  The Shapiro-Wilk test was developed by Samuel Shapiro and Martin Wilk and first presented in 1965. The test verifies the hypothesis that the underlying population of a data sample is normally distributed. More precisely, for the nominator the test statistic calculates what the variance of a data sample would look like if it came from a normal distribution (first estimator) and compares this "expected" variance with the actual variance of the data sample for the denominator (second estimator). If the population of the data sample is normally distributed, then both estimators for the variance should be approximately the same.

[7] The results of the statistical tests for our data samples can be found on GitHub.

3.2.2 Tests Based on Descriptive Measures

Based on moments, the most popular tests are the D'Agostino-Pearson Omnibus test and the Jarque-Bera test:

- **D'Agostino-Pearson Omnibus Test**
  The D'Agostino-Pearson Omnibus Test (1973) computes summary statistics from the sample data to determine whether the data distribution deviates from the Gaussian distribution. Specifically, it first calculates the skewness and kurtosis to quantify how far the sample distribution deviates from the normal distribution in terms of asymmetry and shape. It then calculates to what extent each of these values differs from the value expected for a Gaussian distribution and calculates a single p-value from the sum of these deviations.

- **Jarque-Bera-Test**
  The Jarque-Bera test was proposed by Carlos M. Jarque and Anil K. Bera (1982) and also calculates summary statistics from the sample data to verify whether the data sample differs from the normal distribution. Concretely, the test determines whether the skewness and kurtosis of the data sample are unusual compared to their expected values, as measured by a chi-square statistic.

4 Monte Carlo Simulation in Excel[8]

Using the Inverse Transformation Sampling method, we can now perform a Monte Carlo simulation to compute the VaR and the ES for a portfolio of two ETFs.

We start by generating random drawings from a normal distribution[9]. Thereafter, we can use the Cholesky decomposition of the correlation matrix C to obtain correlated asset returns. More specifically, the Cholesky decomposition of C is a decomposition of the form:[10]

$$C = LL^T$$

where L is a lower triangular matrix with real and positive diagonal entries, and L^T denotes the conjugate transpose of L.

Correlated asset returns can then be obtained by

$$LX = Y$$

where X are uncorrelated asset returns and Y represent correlated asset returns.[11]

To arrive at our final simulated asset returns, we add the expected returns to the correlated asset returns. Finally, we can calculate our desired risk measures, which can be defined as follows:

VaR: For a position V with random changes in value ΔV over a time horizon $[0, \Delta t]$ the VaR at the confidence level $\alpha \in (0, 1)$ is given by the smallest number I, such that the probability that the loss -ΔV exceeds I is no larger than $(1 - \alpha)$.

ES: For a position V with random changes in value ΔV over a time horizon $[0, \Delta t]$ the ES (ES_α) at the confidence level $\alpha \in (0, 1)$ is defined as the conditional expected value of all losses exceeding VAR_α.

---

[8] The complete calculation can be found in the Excel file located in the Dropbox folder.
[9] In Excel this can be achieved by STANDNORMINV(normalRand()).
[10] Correlation matrices are positive semidefinite.
[11] For example, if we want two correlated asset returns, then $L = \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho} \end{bmatrix}$.

References

[1] Anderson, T.W. & Darling, D.A. (1952), 'Asymptotic theory of certain goodness-of-fit criteria based on stochastic processes.', in: The Annals of Mathematical Statistics, Vol. 23, No. 2, pp. 193-212.

[2] Bera, A. K. & Jarque, C. M. (1982), 'Model specification tests: A simultaneous approach.', in: Journal of Econometrics, Vol. 20, pp. 59-82.

[3] Chhikara, R.S. & Folks, J.L. (1974), 'Estimation of the Inverse Gaussian Distribution Function," in: Journal of the Royal Statistical Society, Vol. 69, pp. 250-254.

[4] D'Agostino, R. B. (1971.), 'An omnibus test of normality for moderate and large sample sizes.', in: Biometrika, Vol. 58, No. 2, pp. 341-348.

[5] D'Agostino, R. B. (1986), 'Tests for normal distribution.', in: Goodness-of-fit Techniques, edited by D'Agostino, R. B., and Stephens, M. A., pp. 367-420, New York. Marcel Dekker.

[6] D'Agostino R, & Pearson E. S. (1973), 'Tests for departure from normality. Empirical results for the distributions of b_2 and $\sqrt{b_1}$', in: Biometrika, Vol. 60, No. 3, pp. 613-622.

[7] Devroye, L. (1986). 'Non-uniform Random Variate Generation', Springer-Verlag, New York.

[8] Karl Sigman (2007), 'Acceptance-Rejection Method', Columbia University.

[9] Kolmogorov, A. (1933), 'Sulla determinazione empirica di una legge di distribuzione.', in: Giornale dell'Istituto Italiano degli Attuari, Vol. 4, pp. 83–91.

[10] Murison, M.A. (2000), 'A Method for Directly Generating a Gaussian Distribution with Nonunit. Variance and Nonzero Mean from Uniform Random Deviates', Washington.

[11] Miller, F., Vandome, A., & John, M. (2010), 'Inverse Transform Sampling', VDM Verlag Dr. Mueller e.K.

[12] Shapiro, S.S. & Wilk, M.B. (1965), 'Ananalysis of variance test for normality (complete samples).', in: Biometrika, Vol.52, No. 3/4, pp. 591-611.

[13] Smirnov, N. (1948), 'Table for estimating the goodness of fit of empirical distributions.', in: Annals of Mathematical Statistics, Vol. 19, No. 2, pp. 279–281.

[14] Wichura, M.J. (1988), 'Algorithm as 241: The percentage points of the normal distribution', in: Applied Statistics, Vol. 37, No. 3, pp. 477-484.

Dropbox

https://www.dropbox.com/home/UCL

GitHub Repository

https://nbviewer.jupyter.org/github/dduemig/UCL-Project-Generating-normally-distributed-pseudo-random-numbers-/blob/master/UCL%20Project.ipynb

Appendix

I. Proofs of the Inverse Sampling Algorithm

- **Proof of Case 1:**

  Notice the following facts:

  - $x \leq y,\ x, y \in \mathbb{R} \implies F(x) \leq F(y)$

    This is because F is CDF and CDF is non $-$ decreasing

  - $x < y,\ x, y \in \mathbb{R} \implies F(x) < F(y)$

    Since F is invertible for $x \neq y, F(x) \neq F(y)$

    Combining this and the previous statement, we can conclude that $F(x) < F(y)$

  - $a \leq b,\ a, b \in (0,1) \implies F^{-1}(a) \leq F^{-1}(b)$

    We will prove this by contradiction. Let us suppose $a \leq b$ and if at all possible $F^{-1}(a) > F^{-1}(b)$

    $\implies F\big(F^{-1}(a)\big) > F\big(F^{-1}(b)\big)$

    $\implies a > b$ which is a contradiction

    Hence Proved

  Now let $U \sim Uniform(0,1)$

  Let $H(x)$ denote the CDF of the $Uniform$ Distribution

  $$H(x) = \begin{cases} 0 & x \leq 0 \\ x & 0 < x < 1 \\ 1 & x \geq 1 \end{cases}$$

  Now:

  $$\begin{aligned} P(F^{-1}(U) \leq x) &= P\big(U \leq F(x)\big) \\ &= H\big(F(x)\big) \\ &= F(x) \end{aligned}$$

  as range of F is $(0,1)$

  This implies $F^{-1}(U) \sim F$

  Hence Proved

- **Proof of Case 2:**

First, we notice that

- If $F: \mathbb{R} \to (0,1)$ is invertible then $G = F^{-1}$
- $G(a) \in (0,1)$ since $F(x) \to 1 \; as \; x \to \infty \; and \; F(x) \to 0 \; as \; x \to -\infty$

Now we will prove the following claim:

Claim: $\forall\, a \in (0,1), \forall\, y \in \mathbb{R}: a \leq F(y)$ iff $G(a) \leq y$

Proof: Let us suppose a $\leq F(y)$ for some $a \in (0,1)$ and $y \in \mathbb{R}$

Now $y \in \{x: a \leq F(x)\}$

$\implies y \geq inf\{x: a \leq F(x)\} = G(a)$

$\implies y \geq G(a)$

Now to prove the converse we prove the contrapositive statement, i. e.

$a > F(y)$ implies that $G(a) > y$

So, now assume $a > F(y)$

By right continuity, $\exists y_o > y$ such that $F(y_o) < a$

$\implies \{x: a \leq F(x)\} \subset [y_o, \infty)$

$\implies G(a) \geq y_o > y$

$\implies G(a) > y$

Hence Proved

Now, repeating the last argument of proof of Case 1 we can prove that G(U) has CDF F

Let $H(x)$ be the CDF of $uniform\ (0,1)$ as defined before and let $U \sim Uniform(0,1)$

$$\begin{aligned} P(G(U) \leq x) &= P\big(U \leq F(x)\big) \\ &= H\big(F(x)\big) \\ &= F(x) \end{aligned}$$

Hence, $G(U) \sim F$

Hence Proved

II. Acceptance-Rejection Algorithm

1. Generate a random variable Y distributed as G

2. Generate U (independet from Y)

3. If $U \leq \frac{f(Y)}{cg(Y)}$ then set $X = Y$ ("accept"); otherwise go back to 1 ("reject")

Before proofing this, we notice that

- $f(Y)$ and $g(Y)$ are random variables, and so is the ratio $\frac{f(Y)}{cg(Y)}$

  This ratio is independent of U in Step 2
- The ratio is bounded between 0 and 1; $0 < \frac{f(Y)}{cg(Y)} \leq 1$
- The number of times N, steps 1 and 2 need to be called (e. g. the number of iterations needed to successfully generate X) is itself a random variable and has geometric distribution with "success" probability

  $p = P(U \leq \frac{f(Y)}{cg(Y)}$; $\mathrm{P(N = n)} = (1-p)^{n-1}p, n \geq 1$

  Thus, on average the number of iterations required is given by $E(N) = \frac{1}{P}$
- In the end we obtain our X as having the conditional distribution of a Y given that the event $\left\{U \leq \frac{f(Y)}{cg(Y)}\right\}$ occurs

A direct calculation yields that $p = \frac{1}{c}$, by first conditioning on Y:

$P\left(U \leq \frac{f(Y)}{cg(Y)} \middle| Y = y\right) = \frac{f(y)}{cg(y)}$; thus, unconditioning and recalling that Y has density $g(y)$ yields $p = \int_{-\infty}^{\infty} \frac{f(y)}{cg(y)} \times g(y)dy$

$$= \frac{1}{c}\int_{-\infty}^{\infty} f(y)dy$$

$$= \frac{1}{c}$$

where the last equality follows since f is a density function (hence by definition integrates to 1). Thus, $E(N) = c$, the bounding constant, and we can now indeed see that it is desirable to choose our alternative density g so as to minimize this constant

$c = \sup_x \left\{\frac{f(x)}{g(x)}\right\}$. Of course the optimal function would be $g(x) = f(x)$ which is not

what we have in mind since the whole point is to choose a different alternative form *f.*

We summarize with:

The expected number of iterations of the algorithm required until an X is successfully generated is exactly the bounding constant $c = \sup_x \left\{\frac{f(x)}{g(x)}\right\}$

Proof that the algorithm works

We must show that the conditional distribution of Y given that $U \leq \frac{f(Y)}{cg(Y)}$,

is indeed F; i. e. that $P\left(Y \leq y \mid U \leq \frac{f(Y)}{cg(Y)}\right) = F(y)$. Letting $B = \left\{U \leq \frac{f(Y)}{cg(Y)}\right\}$,

$A = \{Y \leq y\}$, recalling that $P(B) = p = \frac{1}{c}$, and then using the basic fact that

$P(A|B) = P(B|A)\frac{P(A)}{P(B)}$ yields

$$P\left(U \leq \frac{f(Y)}{cg(Y)}\middle| Y \leq y\right) \times \frac{G(y)}{\frac{1}{c}} = \frac{F(y)}{cG(y)} \times \frac{G(y)}{\frac{1}{c}} = F(y)$$

where we used the following computation:

$$P\left(U \leq \frac{f(Y)}{cg(Y)}\middle| Y \leq y\right) = \frac{P\left(U \leq \frac{f(Y)}{cg(Y)},\ Y \leq y\right)}{G(y)}$$

$$= \int_{-\infty}^{y} \frac{P\left(U \leq \frac{f(y)}{cg(y)}\middle| Y = w \leq y\right)}{G(y)} g(w)dw$$

$$= \frac{1}{G(y)} \int_{-\infty}^{y} \frac{f(w)}{cg(w)} g(w)dw$$

$$= \frac{1}{cG(y)} \int_{-\infty}^{y} f(w)dw$$

$$= \frac{F(y)}{cG(y)}$$

III. Proof of Box-Mueller Transformation

We have

$$\sqrt{\frac{(x_1-\mu)^2}{\sigma^2}+\frac{(x_2-\mu)^2}{\sigma^2}}=\sqrt{2}\sqrt{-\ln(u_1)}$$

which we solve for u_1

$$u_{1=}e^{\left(-\frac{1}{2}\frac{(x_2-\mu)^2}{\sigma^2}\right)}\,e^{\left(-\frac{1}{2}\frac{(x_1-\mu)^2}{\sigma^2}\right)}$$

and we have

$$\frac{x_2}{x_1}=\frac{\sqrt{-2\ln(u_1)}\sin(2\pi u_2)\,\sigma+\mu}{\sqrt{-2\ln(u_1)}\cos(2\pi u_2)\,\sigma+\mu}$$

which we solve for u_2

$$u_2=-\frac{1}{2}\arctan\left(\frac{\left(\left(\sqrt{-\ln(u_1)}\,x_1(-x_1+x_2)\mu-\sqrt{\ln(u_1)\,x_2^2(-x_1+x_2)^2\mu^2+2\ln(u_1)^2x_2^2(x_2^2+x_1^2)\sigma^2}\right)x_2\right)}{\left(\sqrt{-\ln(u_1)}\,x_2^2(-x_1+x_2)\mu+x_1\sqrt{\ln(u_1)\,x_2^2(-x_1+x_2)^2\mu^2+2\ln(u_1)^2x_2^2(x_2^2+x_1^2)\,\sigma^2}\right)}\right)/\pi$$

Substituting for u_1 and simplifying, we find

$$u_2=\frac{1}{2}\frac{arctan\left(\frac{-x_2+\mu}{-x_1+\mu}\right)}{\pi}$$

Now, the fundamental transformation law of probabilities states that

$$|p(y)dy|=|p(x)dx|,\text{ or }p(y)=p(x)\left|\frac{d_x}{d_y}\right|,$$

where $p(x)dx$ is the probability that x lies between x and $x+dx$,

and $p(y)dy$ is the probability that y lies between y and y + dy.

The extension of this to n dimenstion is $p(Y)dY=p(x)J(X,Y)dY$,

where $p(X)$ is the joint probability distribution of X,

$p(Y)$ is the joint probability distribution of Y,

$X=(x_1,x_2..x_n),Y=(y_1,y_2..y_n),dY=dy_1,dy_2,\dots dy_n$

and $J(X,Y)$ is the Jacobi determinant. For two dimensions, $J(X,Y)$

$$=det\left(\begin{bmatrix}\frac{\partial}{\partial y_1}x_1 & \frac{\partial}{\partial y_2}x_1\\ \frac{\partial}{\partial y_1}x_2 & \frac{\partial}{\partial y_2}x_2\end{bmatrix}\right)\ Hence,$$

$$J(Z,X) = -\frac{1}{2}\frac{e^{\left(-\frac{1}{2}\frac{(-x_1+\mu)^2}{\sigma^2}\right)}\, e^{\left(-\frac{1}{2}\frac{(-x_2+\mu)^2}{\sigma^2}\right)}}{\pi\, \sigma^2}$$

Thus, for a uniform distribution (u_1, u_2) the distribution of (x_1, x_1)

is a symmetric bivariate Gaussian distribution with mean μ and variance σ^2

Hence Proved