Random Variate Generation

Christos Alexopoulos and Dave Goldsman

Georgia Institute of Technology, Atlanta, GA, USA

5/28/18

Outline

Goal: Use $\mathcal{U}(0,1)$ numbers to generate observations (variates) from other distributions, and even stochastic processes.

Try to be fast, reproducible.

- Discrete distributions, like Bernoulli, Binomial, Poisson, and empirical
- Continuous distributions like exponential, normal (many ways), and empirical
- Multivariate normal
- Nonhomogeneous Poisson process
- Autoregressive moving average time series
- Waiting times
- Brownian motion

Outline

Inverse Transform Method

Inverse Transform Theorem: Let X be a continuous random variable with c.d.f. $F(x)$. Then $F(X) \sim \mathcal{U}(0,1)$.

Proof: Let $Y = F(X)$ and suppose that Y has c.d.f. $G(y)$. Then

$$\begin{aligned} G(y) &= \mathrm{P}(Y \le y) = \mathrm{P}(F(X) \le y) \\ &= \mathrm{P}(X \le F^{-1}(y)) = F(F^{-1}(y)) = y. \quad \square \end{aligned}$$

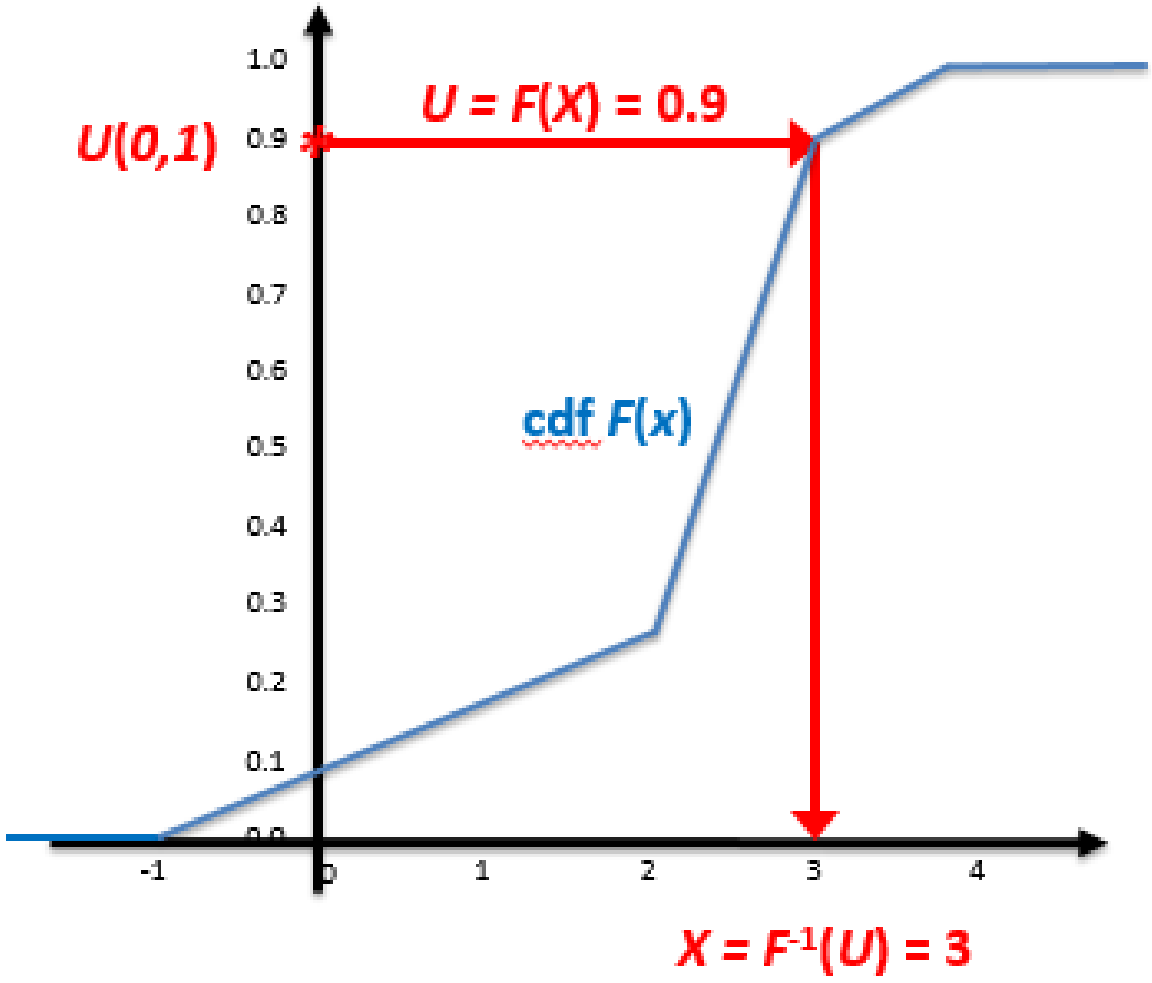
In the above, we can define the inverse c.d.f. by

$$F^{-1}(u) = \min[x : F(x) \ge u] \quad u \in [0,1].$$

This representation can be applied to continuous or *discrete* or mixed distributions (see figure).

Inverse Transform Method

(generate X from U)



How do we use this result?

Let $U \sim \mathcal{U}(0,1)$. Then $F(X) = U$ means that the random variable $F^{-1}(U)$ has the same distribution as X.

So here is the *inverse transform method* for generating a RV X having c.d.f. $F(x)$:

1. Sample U from $\mathcal{U}(0,1)$.
2. Return $X = F^{-1}(U)$.

We'll do some continuous examples first, then discrete.

Example: The $\mathcal{U}(a,b)$ distribution, with $F(x) = \frac{x-a}{b-a}$, $a \leq x \leq b$.

Solving $(X-a)/(b-a) = U$ for X, we get $X = a + (b-a)U$. $\square$

Example: The Exp(λ) distribution, with $F(x) = 1 - e^{-\lambda x}, x > 0$.

Solving $F(X) = U$ for X,

$$X = -\frac{1}{\lambda}\ell\mathrm{n}(1-U) \quad \text{or} \quad X = -\frac{1}{\lambda}\ell\mathrm{n}(U). \quad \square$$

Example: The Weibull distribution, $F(x) = 1 - e^{-(\lambda x)^{\beta}}, x > 0$.

Solving $F(X) = U$ for X,

$$X = \frac{1}{\lambda}\left[-\ell\mathrm{n}(1-U)\right]^{1/\beta} \quad \text{or} \quad X = \frac{1}{\lambda}\left[-\ell\mathrm{n}(U)\right]^{1/\beta} \quad \square.$$

Example: The triangular (0,1,2) distribution has p.d.f.

$$f(x) = \begin{cases} x & \text{if } 0 \le x < 1 \\ 2 - x & \text{if } 1 \le x \le 2. \end{cases}$$

The c.d.f. is

$$F(x) = \begin{cases} x^2/2 & \text{if } 0 \le x < 1 \\ 1 - (x-2)^2/2 & \text{if } 1 \le x \le 2. \end{cases}$$

If $U < 1/2$, we solve $X^2/2 = U$ to get $X = \sqrt{2U}$.
If $U \ge 1/2$, the only root of $1 - (X-2)^2/2 = U$ in $[1, 2]$ is

$$X = 2 - \sqrt{2(1-U)}.$$

Thus, for example, if $U = 0.4$, we take $X = \sqrt{0.8}$. $\square$

Remark: Do not replace U by $1 - U$ here!

Example: The standard normal distribution. Unfortunately, the inverse c.d.f. $\Phi^{-1}(\cdot)$ does not have an analytical form. *This is often a problem with the inverse transform method.*

Easy solution: Do a table lookup. E.g., If $U = 0.975$, then $Z = \Phi^{-1}(U) = 1.96$. $\square$

Crude portable approximation (BCNN): The following approximation gives at least one decimal place of accuracy for $0.00134 \leq U \leq 0.98865$:

$$Z = \Phi^{-1}(U) \approx \frac{U^{0.135} - (1-U)^{0.135}}{0.1975}. \quad \square$$

Here's a better portable solution: The following approximation has absolute error $\leq 0.45 \times 10^{-3}$:

$$Z \;=\; \text{sign}(U - 1/2)\left(t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3}\right),$$

where $\text{sign}(x) = 1, 0, -1$ if x is positive, zero, or negative, respectively,

$$t \;=\; \{-\ell\text{n}[\min(U, 1 - U)]^2\}^{1/2},$$

and

$$c_0 = 2.515517, \quad c_1 = 0.802853, \quad c_2 = 0.010328,$$
$$d_1 = 1.432788, \quad d_2 = 0.189269, \quad d_3 = 0.001308.$$

In any case, if $Z \sim \text{Nor}(0,1)$ and you want $X \sim \text{Nor}(\mu, \sigma^2)$, just take $X \leftarrow \mu + \sigma Z$. $\square$

Easy Example (Inverse Transform): Suppose you want to generate $X \sim \text{Nor}(3, 16)$, and you start with $U = 0.59$. Then

$$X = \mu + \sigma Z = 3 + 4\Phi^{-1}(0.59) = 3 + 4(0.2275) = 3.91. \quad \square$$

For discrete distributions, it's often best to construct a table.

Baby Discrete Example: The Bernoulli(p) distribution.

x	$P(X = x)$	$F(x)$	$\mathcal{U}(0,1)$'s
0	$1-p$	$1-p$	$[0,\ 1-p]$
1	p	1	$(1-p, 1]$

If $U \leq 1-p$, then take $X = 0$; otherwise, $X = 1$. For instance, if $p = 0.75$ and we generate $U = 0.13$, we take $X = 0$. $\square$

Alternately, we can construct the following "backwards" table (which isn't strictly inverse transform, but it's the one that I usually use).

x	$P(X = x)$	$\mathcal{U}(0,1)$'s
1	p	$[0,\ p]$
0	$1-p$	$(p, 1]$

If $U \leq p$, take $X = 1$; otherwise, $X = 0$. $\square$

Example: Suppose we have a slightly less-trivial discrete p.m.f.

x	$P(X = x)$	$F(x)$	$\mathcal{U}(0,1)$'s
-1	0.6	0.6	[0.0,0.6]
2.5	0.3	0.9	(0.6,0.9]
4	0.1	1.0	(0.9,1.0]

Thus, if $U = 0.63$, we take $X = 2.5$. $\square$

Sometimes there's an easy way to avoid constructing a table.

Example: The discrete uniform distribution on $\{1, 2, \ldots, n\}$,

$$P(X = k) = \frac{1}{n}, \quad 1, 2, \ldots, n.$$

Clearly, $X = \lceil nU \rceil$, where $\lceil \cdot \rceil$ is the ceiling function.

So if $n = 10$ and $U = 0.376$, then $X = \lceil 3.76 \rceil = 4$. $\square$

Example: The geometric distribution with p.m.f. and c.d.f.

$$f(k) = q^{k-1}p \quad \text{and} \quad F(k) = 1 - q^k,\ k = 1, 2, \ldots,$$

where $q = 1 - p$. Thus, after some algebra,

$$X = \min[k : 1 - q^k \geq U] = \left\lceil \frac{\ell\mathrm{n}(1-U)}{\ell\mathrm{n}(1-p)} \right\rceil \sim \left\lceil \frac{\ell\mathrm{n}(U)}{\ell\mathrm{n}(1-p)} \right\rceil.$$

For instance, if $p = 0.3$ and $U = 0.72$, we obtain

$$X = \left\lceil \frac{\ell\mathrm{n}(0.28)}{\ell\mathrm{n}(0.7)} \right\rceil = 4. \quad \square$$

Remark: Can also generate Geom(p) by counting Bern(p) trials until you get a success.

Easy Example: Generate $X \sim$ Geom(1/6). This is the same thing as counting the number of dice tosses until a 3 (or any particular number) comes up, where the Bern(1/6) trials are the i.i.d. dice tosses. For instance, if you toss 6,2,1,4,3, then you stop on the Bernoulli trial $X = 5$, and that's your answer.

But life isn't always dice tosses. A general way to generate a Geom(p) is to count the number of trials until $U_i \leq p$. For example, if $p = 0.3$, then $U_1 = 0.71$, $U_2 = 0.96$, and $U_3 = 0.12$ implies that $X = 3$. $\square$

Remark: If you have a discrete distribution like Pois(λ) with an infinite number of values, you could write out table entries until the c.d.f. is nearly one, generate exactly one U, and then search until you find $X = F^{-1}(U)$, i.e., x_i such that $U \in \big(F(x_{i-1}),\ F(x_i)\big]$.

x	$P(X = x)$	$F(x)$	$\mathcal{U}(0,1)$'s
x_1	$f(x_1)$	$F(x_1)$	$\big[0,\ F(x_1)\big]$
x_2	$f(x_2)$	$F(x_2)$	$\big(F(x_1),\ F(x_2)\big]$
x_3	$f(x_3)$	$F(x_3)$	$\big(F(x_2),\ F(x_3)\big]$
$\vdots$			

Example: Suppose $X \sim \text{Pois}(2)$, so that $f(x) = \frac{e^{-2}2^x}{x!}$, $x = 0, 1, 2, \ldots$.

x	$f(x)$	$F(x)$	$\mathcal{U}(0,1)$'s
0	0.1353	0.1353	$[0,\ 0.1353]$
1	0.2707	0.4060	$(0.1353,\ 0.4060]$
2	0.2707	0.6767	$(0.4060,\ 0.6767]$
3	0.1804	0.8571	$(0.6767,\ 0.8571]$
$\vdots$			

For instance, if $U = 0.313$, then $X = 1$ $\square$.

Continuous Empirical Distributions

If you can't find a good theoretical distribution to model a certain RV, you may want to use the *empirical c.d.f.* of the data, $X_1, X_2, \ldots, X_n$,

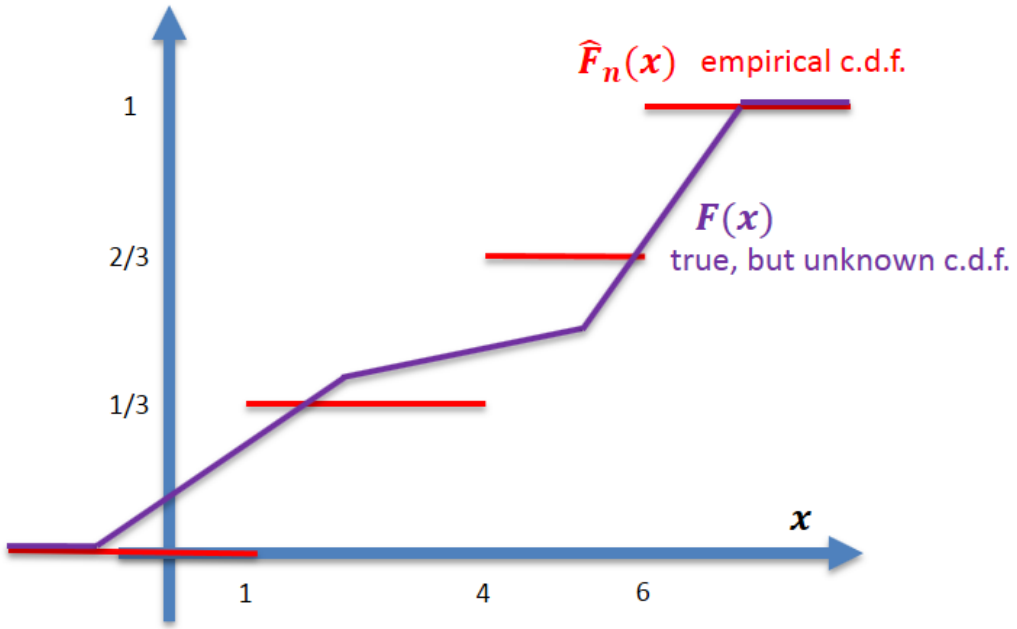
$$\hat{F}_n(x) \equiv \frac{\text{number of } X_i\text{'s} \leq x}{n}.$$

Note that $\hat{F}_n(x)$ is a step function with jumps of height $1/n$ (every time an observation occurs).

Good news: Even though X is continuous, the *Glivenko-Cantelli Lemma* says that $\hat{F}_n(x) \to F(x)$ for all x as $n \to \infty$. So $\hat{F}_n(x)$ is a good approximation to the true c.d.f., $F(x)$.

The ARENA functions `DISC` and `CONT` can be used to generate RV's from the empirical c.d.f.'s of discrete and continuous distributions, respectively.

To do so ourselves, we first define the *ordered* points, $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$. For example, if $X_1 = 4$, $X_2 = 1$, and $X_3 = 6$, then $X_{(1)} = 1$, $X_{(2)} = 4$, and $X_{(3)} = 6$.



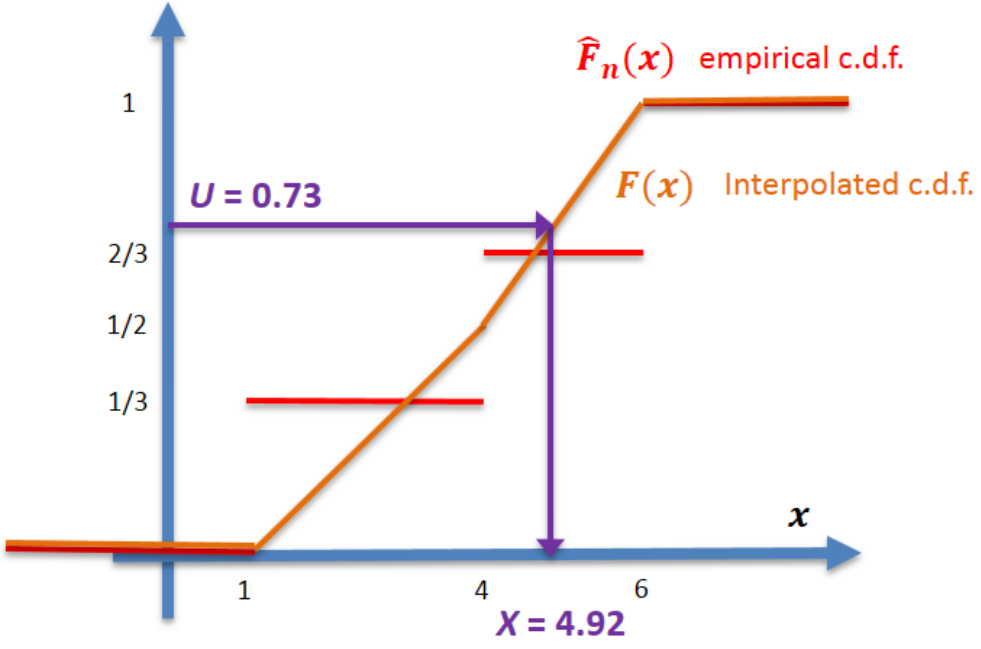
Given that you only have a finite number n of data points, we can turn the empirical c.d.f. into a continuous RV by using linear interpolation between the $X_{(i)}$'s.

$$F(x) = \begin{cases} 0 & \text{if } x < X_{(1)} \\ \frac{i-1}{n-1} + \frac{x - X_{(i)}}{(n-1)(X_{(i+1)} - X_{(i)})} & \text{if } X_{(i)} \leq x < X_{(i+1)}, \forall i \\ 1 & \text{if } x \geq X_{(n)} \end{cases}$$

1. Set $F(X) = U \sim \mathcal{U}(0,1)$. Let $P = (n-1)U$ and $I = \lceil P \rceil$.
2. Solve to get $X = X_{(I)} + (P - I + 1)\big(X_{(I+1)} - X_{(I)}\big)$.

Example: Suppose $X_{(1)} = 1$, $X_{(2)} = 4$, and $X_{(3)} = 6$. If $U = 0.73$, then $P = (n-1)U = 1.46$ and $I = \lceil P \rceil = 2$. So

$$\begin{aligned} X &= X_{(I)} + (P - I + 1)\big(X_{(I+1)} - X_{(I)}\big) \\ &= X_{(2)} + (1.46 - 2 + 1)\big(X_{(3)} - X_{(2)}\big) \\ &= 4 + (0.46)\big(6 - 4\big) \;=\; 4.92. \quad \square \end{aligned}$$



Check (slightly different way):

$$F(x) = \begin{cases} 0 + \frac{x-1}{2(4-1)} & \text{if } 1 \le x < 4 \quad (i = 1 \text{ case}) \\ \frac{1}{2} + \frac{x-4}{2(6-4)} & \text{if } 4 \le x < 6 \quad (i = 2 \text{ case}) \end{cases}$$

Setting $F(X) = U$ and solving for the two cases, we have

$$X = \begin{cases} 1 + 6U & \text{if } U < 1/2 \\ 2 + 4U & \text{if } U \ge 1/2 \end{cases}$$

Then $U = 0.73$ implies $X = 2 + 4(0.73) = 4.92$. $\square$

Example (BCNN): We can use an *approximate* empirical c.d.f. when dealing with *grouped* data. Let's look at a sample of 100 repair times.

interval	freq	rel freq	approx $\hat{F}(x)$
$0.25 \leq x \leq 0.5$	31	0.31	0.31
$0.5 < x \leq 1.0$	10	0.10	0.41
$1.0 < x \leq 1.5$	25	0.25	0.66
$1.5 < x \leq 2.0$	34	0.34	1.00

How to construct a realization X from $\hat{F}(x)$? Use the fact that $\hat{F}(X) \sim \mathcal{U}(0,1)$ and do inverse transform, i.e., $X = \hat{F}^{-1}(U)$.

For instance, if $U \in (0.66,\ 1.00]$ (the last two $\hat{F}(x)$ entries), then

$$\begin{aligned} X &= \hat{F}^{-1}(0.66) + \Big(\frac{\hat{F}^{-1}(1.00) - \hat{F}^{-1}(0.66)}{1.00 - 0.66}\Big)(U - 0.66) \\ &= 1.5 + \Big(\frac{2.0 - 1.5}{1.00 - 0.66}\Big)(U - 0.66) \ = \ 1.5 + 1.471(U - 0.66). \end{aligned}$$

Thus, e.g., if $U = 0.83$, then $X = 1.75$. $\square$

To do this in general,

1. Generate U.
2. Find the $\hat{F}(x)$ interval in which U lies, i.e., i such that $r_i < U \leq r_{i+1}$. In the above example,

r_1	r_2	r_3	r_4	r_5
0	0.31	0.41	0.66	1.0

3. Let x_i be the left endpoint of the ith X-interval, and let a_i be the reciprocal of the slope of the ith interval.

x_1	x_2	x_3	x_4	x_5
0.25	0.50	1.0	1.5	2.0

a_1	a_2	a_3	a_4	a_5
0.81	5.0	2.0	1.47	–

4. $X = x_i + a_i(U - r_i)$.

Outline

Cutpoint Method

Suppose we want to generate from the discrete distribution

$$P(X = k) \;=\; p_k, \quad k = a, a+1, \ldots, b$$

with large $b - a$. In this case, inverse transform may have to search a lot of possible values, and may be inefficient. Let

$$q_k \;=\; P(X \leq k), \quad k = a, a+1, \ldots, b.$$

For fixed m, the cutpoint method of Fishman and Moore computes and stores the cutpoints

$$I_j \;=\; \min \left[k : q_k > \frac{j-1}{m} \right], \quad j = 1, \ldots, m.$$

These cutpoints help us scan through the list of possible k-values much more quickly than regular inverse transform.

Here is the algorithm that computes the cutpoints . . .

Algorithm CMSET

$j \leftarrow 0$, $k \leftarrow a - 1$, and $A \leftarrow 0$

While $j < m$:

    While $A \leq j$:

        $k \leftarrow k + 1$

        $A \leftarrow mq_k$

    $j \leftarrow j + 1$

    $I_j \leftarrow k$

Once the cutpoints are computed, we can use the cutpoint method.

Algorithm CM

Generate U from $\mathcal{U}(0,1)$

$L \leftarrow \lfloor mU \rfloor + 1$

$X \leftarrow I_L$

While $U > q_X$: $X \leftarrow X + 1$

In short, this algorithm selects an integer $L = \lfloor mU \rfloor + 1$ and starts the search at the value I_L. Its correctness results from the fact that

$$P(I_L \leq X \leq I_{L+1}) = 1 \quad (I_{m+1} = b).$$

Let $\mathrm{E}(C_m)$ be the expected number of comparisons until Algorithm CM terminates. Given L, the maximum number of required comparisons is $I_{L+1} - I_L + 1$. Hence,

$$\begin{aligned} \mathrm{E}(C_m) &\leq \frac{I_2 - I_1 + 1}{m} + \cdots + \frac{I_{m+1} - I_m + 1}{m} \\ &= \frac{b - I_1 + m}{m}. \end{aligned}$$

Note that if $m \geq b$, then $\mathrm{E}(C_m) \leq (2m-1)/m \leq 2$.

Example: Consider the distribution

k	1	2	3	4	5	6	7	8
p_k	0.01	0.04	0.07	0.15	0.28	0.19	0.21	0.05
q_k	0.01	0.05	0.12	0.27	0.55	0.74	0.95	1

For $m = 8$, we have the following cutpoints:

$$\begin{aligned}
I_1 &= \min[i : q_i > 0] = 1 \\
I_2 &= \min[i : q_i > 1/8] = 4 \\
I_3 &= \min[i : q_i > 2/8] = 4 \\
I_4 &= \min[i : q_i > 3/8] = 5 = I_5 \\
I_6 &= \min[i : q_i > 5/8] = 6 \\
I_7 &= \min[i : q_i > 6/8] = 7 = I_8 \\
I_9 &= 8
\end{aligned}$$

For $U = 0.219$, we have $L = \lfloor 8(0.219) \rfloor + 1 = 2$, and

$$X = \min[i : I_2 \le i \le I_3, q_i \ge 0.219] = 4. \quad \square$$

Outline

Convolution Method

Convolution refers to adding things up.

Example: Binomial(n, p).

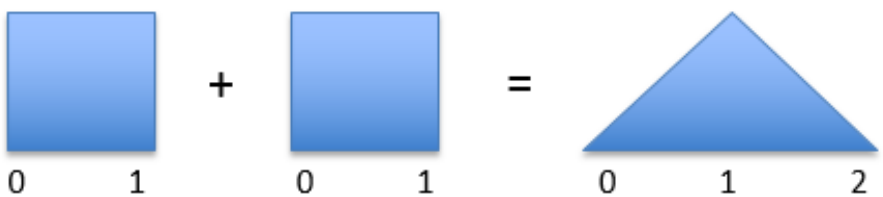
If $X_1, \ldots, X_n \sim$ i.i.d. Bern(p), then $Y = \sum_{i=1}^{n} X_i \sim \text{Bin}(n, p)$.

We already know how to get Bernoulli RVs via Inverse Transform: Suppose $U_1, \ldots, U_n$ are i.i.d. $\mathcal{U}(0,1)$. If $U_i \leq p$, set $X_i = 1$; otherwise, set $X_i = 0$. Repeat for $i = 1, \ldots, n$. Add up to get Y. $\square$

For instance, if $Y \sim \text{Bin}(3, 0.4)$ and $U_1 = 0.63$, $U_2 = 0.17$, and $U_3 = 0.81$, then $Y = 0 + 1 + 0 = 1$. $\square$

Example: Triangular(0,1,2).

It can be shown that if U_1 and U_2 are i.i.d. $\mathcal{U}(0,1)$, then $U_1 + U_2$ is Tria(0,1,2). (This is easier — but maybe not faster — than our inverse transform method.) $\square$



Example: Erlang$_n(\lambda)$. If $X_1, \ldots, X_n \sim$ i.i.d. Exp(λ), then $Y = \sum_{i=1}^n X_i \sim$ Erlang$_n(\lambda)$. By inverse transform,

$$Y = \sum_{i=1}^n X_i = \sum_{i=1}^n \left[\frac{-1}{\lambda}\ell\mathrm{n}(U_i)\right] = \frac{-1}{\lambda}\ell\mathrm{n}\left(\prod_{i=1}^n U_i\right).$$

This only takes one natural log evaluation, so it's pretty efficient. □

Example: A *crude* "desert island" Nor(0,1) approximate generator (which I wouldn't use).

Suppose that $U_1, \ldots, U_n$ are i.i.d. $\mathcal{U}(0,1)$, and let $Y = \sum_{i=1}^n U_i$. For large n, the CLT implies that $Y \approx \mathrm{Nor}(n/2, n/12)$.

In particular, let's choose $n = 12$, and assume that it's "large." Then

$$Y - 6 = \sum_{i=1}^{12} U_i - 6 \approx \mathrm{Nor}(0,1). \quad \square$$

Other convolution-related tidbits:

Did you know. . . ?

If $X_1, \ldots, X_n$ are i.i.d. Geom(p), then $\sum_{i=1}^{n} X_i \sim \text{NegBin}(n, p)$.

If $Z_1, \ldots, Z_n$ are i.i.d. Nor(0,1), then $\sum_{i=1}^{n} Z_i^2 \sim \chi^2(n)$.

If $X_1, \ldots, X_n$ are i.i.d. Cauchy, then $\bar{X} \sim$ Cauchy (this is kind of like getting nowhere fast!).

Outline

Acceptance-Rejection Method

Motivation: The majority of c.d.f.'s cannot be inverted efficiently. A-R samples from a distribution that is "almost" the one we want, and then adjusts by "accepting" only a certain proportion of those samples.

Baby Example: Generate a $\mathcal{U}(\frac{2}{3}, 1)$ RV. (You would usually do this via inverse transform, but what the heck!) Here's the A-R algorithm:

1. Generate $U \sim \mathcal{U}(0, 1)$.

2. If $U \geq \frac{2}{3}$, ACCEPT $X \leftarrow U$. O'wise, REJECT and go to Step 1.

Notation: Suppose we want to simulate a continuous RV X with p.d.f. $f(x)$, but that it's difficult to generate directly. Also suppose that we can easily generate a RV having p.d.f. $h(x) \equiv t(x)/c$, where $t(x)$ *majorizes* $f(x)$, i.e.,

$$t(x) \geq f(x), \quad x \in \mathbb{R},$$

and

$$c \equiv \int_{\mathbb{R}} t(x)\,dx \geq \int_{\mathbb{R}} f(x)\,dx = 1,$$

where we assume that $c < \infty$.

Theorem (von Neumann 1951): Define $g(x) \equiv f(x)/t(x)$ and note that $0 \leq g(x) \leq 1$ for all x. Let $U \sim \mathcal{U}(0,1)$, and let Y be a RV (independent of U) with p.d.f. $h(y) = t(y)/c$. If $U \leq g(Y)$, then Y has (conditional) p.d.f. $f(y)$.

This suggests the following "acceptance-rejection" algorithm ...

Algorithm A-R

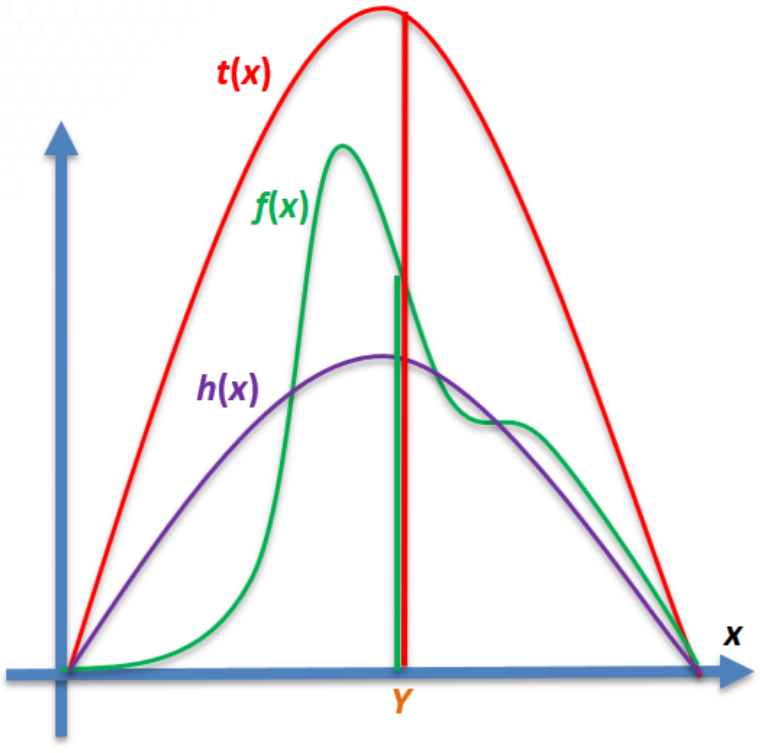
Repeat

Generate U from $\mathcal{U}(0,1)$

Generate Y from $h(y)$ (independent of U)

until $U \leq g(Y) = \frac{f(Y)}{t(Y)} = \frac{f(Y)}{c\,h(Y)}$

Return $X \leftarrow Y$



Generate a point Y uniformly under $t(x)$ (equivalently, sample Y from p.d.f. $h(x)$).

Accept the point with probability $f(Y) / t(Y) = f(Y) / [c\, h(Y)]$.

If you accept, then set $X = Y$ and stop.

Proof that X has p.d.f. $f(x)$.

Let A be the "Acceptance" event. The c.d.f. of X is

$$P(X \leq x) \;=\; P(Y \leq x|A) \;=\; \frac{P(A,\; Y \leq x)}{P(A)}. \tag{1}$$

Then

$$\begin{aligned} P(A|Y=y) &= P\left(U \leq g(Y)|Y=y\right) \\ &= P\left(U \leq g(y)|Y=y\right) \\ &= P\left(U \leq g(y)\right) \quad (U \text{ and } Y \text{ are independent}) \\ &= g(y) \quad (U \text{ is uniform}). \end{aligned} \tag{2}$$

By the law of total probability,

$$\begin{aligned}
P(A,\ Y \le x) &= \int_{-\infty}^{\infty} P(A,\ Y \le x | Y = y) h(y)\, dy \\
&= \int_{-\infty}^{x} P(A | Y = y) h(y)\, dy \\
&= \frac{1}{c} \int_{-\infty}^{x} P(A | Y = y) t(y)\, dy \\
&= \frac{1}{c} \int_{-\infty}^{x} g(y) t(y)\, dy \quad \text{(by (2))} \\
&= \frac{1}{c} \int_{-\infty}^{x} f(y)\, dy. \qquad (3)
\end{aligned}$$

Letting $x \to \infty$, we have

$$P(A) \;=\; \frac{1}{c}\int_{-\infty}^{\infty} f(y)\,dy \;=\; \frac{1}{c}. \tag{4}$$

Then (1), (3), and (4) imply

$$P(X \le x) \;=\; \frac{P(A,\; Y \le x)}{P(A)} \;=\; \int_{-\infty}^{x} f(y)\,dy,$$

so that, by the Fundamental Theorem of Calculus, the p.d.f. of X is

$$\frac{d\,P(X \le x)}{dx} \;=\; \frac{d}{dx}\int_{-\infty}^{x} f(y)\,dy \;=\; f(x). \quad \square$$

There are two main issues:

- The ability to quickly sample from $h(y)$.
- $c \geq 1$ ought to be as close to 1 as possible — i.e., $t(x)$ must be "close" to $f(x)$. That's because

$$P(U \leq g(Y)) \; = \; \frac{1}{c}$$

and the number of trials until "success" $[U \leq g(Y)]$ is Geom$(1/c)$, so that the mean number of trials is c.

Example (Law 2015): Generate a RV with p.d.f. $f(x) = 60x^3(1-x)^2$, $0 \le x \le 1$. Can't invert this analytically.

Note that the maximum occurs at $x = 0.6$, and $f(0.6) = 2.0736$.

Using the majorizing function

$$t(x) \;=\; 2.0736, \quad 0 \le x \le 1$$

(which isn't actually very efficient), get $c = \int_0^1 t(x)\,dx = 2.0736$, so

$$h(x) \;=\; \frac{t(x)}{c} \;=\; 1, \quad 0 \le x \le 1 \quad \text{(i.e., a } \mathcal{U}(0,1) \text{ p.d.f.)}$$

and

$$g(x) \;=\; \frac{f(x)}{t(x)} \;=\; 60x^3(1-x)^2/2.0736.$$

E.g., if we generate $U = 0.13$ and $Y = 0.25$, then it turns out that $U \le g(Y) = 60Y^3(1-Y)^2/2.0736$, so we take $X \leftarrow 0.25$. $\square$

Example (Ross): Generate a standard *half-normal* RV, with p.d.f.

$$f(x) \;=\; \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \quad x \geq 0.$$

Using the majorizing function

$$t(x) \;=\; \sqrt{\frac{2e}{\pi}} e^{-x} \;\geq\; f(x) \text{ for all } x \geq 0,$$

we get

$$c \;=\; \sqrt{\frac{2e}{\pi}} \int_0^\infty e^{-x}\, dx = \sqrt{\frac{2e}{\pi}} = 1.3155,$$

$$h(x) \;=\; \frac{t(x)}{c} \;=\; e^{-x} \quad \text{(easy Exp(1) p.d.f.)},$$

and

$$g(x) \;=\; \frac{f(x)}{t(x)} \;=\; e^{-(x-1)^2/2}. \quad \square$$

We can use the half-normal result to generate a $\mathrm{Nor}(0, 1)$ variate.

Generate U from $\mathcal{U}(0, 1)$.

Generate X from the half-normal distribution.

Return

$$Z = \begin{cases} -X & \text{if } U \leq 1/2 \\ X & \text{if } U > 1/2. \end{cases}$$

Reminder: We can then generate $\mathrm{Nor}(\mu, \sigma^2)$ RV by using the obvious transformation $\mu + \sigma Z$.

Example: The gamma distribution with p.d.f.:

$$f(x) = \frac{\lambda^\beta x^{\beta-1}}{\Gamma(\beta)} e^{-(\lambda x)^\beta}, \quad x > 0.$$

We'll split the task of generating gamma RV's via the A-R algorithm into two cases depending on the magnitude of the shape parameter: $\beta < 1$ and $\beta \geq 1 \ldots$

If $\beta < 1$, we'll use the following A-R algorithm with $c \leq 1.39$:

Algorithm GAM1

$b \leftarrow (e + \beta)/e$ (e is the base of ℓn)

While (True)

Generate U from $\mathcal{U}(0, 1)$; $W \leftarrow bU$

If $W < 1$

$Y \leftarrow W^{1/\beta}$; Generate V from $\mathcal{U}(0, 1)$

If $V \leq e^{-Y}$: Return $X = Y/\lambda$

Else

$Y \leftarrow -\ell\text{n}[(b - W)/\beta]$

Generate V from $\mathcal{U}(0, 1)$

If $V \leq Y^{\beta-1}$: Return $X = Y/\lambda$

If $\beta \geq 1$, the value of c for the following A-R algorithm decreases from $4/e = 1.47$ to $\sqrt{4/\pi} = 1.13$ as β increases from 1 to ∞.

Algorithm GAM2

$a \leftarrow (2\beta - 1)^{-1/2}$; $b \leftarrow \beta - \ell n(4)$; $c \leftarrow \beta + a^{-1}$; $d \leftarrow 1 + \ell n(4.5)$

While (True)

Generate U_1, U_2 from $\mathcal{U}(0, 1)$

$V \leftarrow a\ell n[U_1/(1 - U_1)]$

$Y \leftarrow \beta e^V$; $Z \leftarrow U_1^2 U_2$

$W \leftarrow b + cV - Y$

If $W + d - 4.5Z \geq 0$: Return $X = Y/\lambda$

Else

If $W \geq \ell n(Z)$: Return $X = Y/\lambda$

Example: The Poisson distribution with probability mass function

$$P(X = n) = e^{-\lambda} \frac{\lambda^n}{n!}, \quad n = 0, 1, \ldots$$

We'll use a variation of A-R to generate a realization of X. The algorithm will go through a set of equivalent statements to arrive at a rule that gives $X = n$.

Recall that, by definition, $X = n$ if we observe exactly n arrivals from a Poisson(λ) process in one time unit.

Define A_i as the ith interarrival time from a Pois(λ) process.

$$
\begin{aligned}
X = n \quad &\Leftrightarrow \quad \text{See exactly } n \text{ Pois}(\lambda) \text{ arrivals by } t = 1 \\
&\Leftrightarrow \quad \sum_{i=1}^{n} A_i \leq 1 < \sum_{i=1}^{n+1} A_i \\
&\Leftrightarrow \quad \sum_{i=1}^{n} \left[\frac{-1}{\lambda} \ell\mathrm{n}(U_i) \right] \leq 1 < \sum_{i=1}^{n+1} \left[\frac{-1}{\lambda} \ell\mathrm{n}(U_i) \right] \\
&\Leftrightarrow \quad \frac{-1}{\lambda} \ell\mathrm{n} \left(\prod_{i=1}^{n} U_i \right) \leq 1 < \frac{-1}{\lambda} \ell\mathrm{n} \left(\prod_{i=1}^{n+1} U_i \right) \\
&\Leftrightarrow \quad \prod_{i=1}^{n} U_i \geq e^{-\lambda} > \prod_{i=1}^{n+1} U_i. \qquad (5)
\end{aligned}
$$

The following A-R algorithm samples $\mathcal{U}(0,1)$'s until (5) becomes true, i.e., until the first time n such that $e^{-\lambda} > \prod_{i=1}^{n+1} U_i$.

Algorithm POIS1

$a \leftarrow e^{-\lambda}$; $p \leftarrow 1$; $X \leftarrow -1$

Until $p < a$

Generate U from $\mathcal{U}(0,1)$

$p \leftarrow pU$; $X \leftarrow X + 1$

Return X

Example (BCNN): Apply Algorithm POIS1 to obtain a Pois(2) variate.

Sample until $e^{-\lambda} = 0.1353 > \prod_{i=1}^{n+1} U_i$.

n	U_{n+1}	$\prod_{i=1}^{n+1} U_i$	Stop?
0	0.3911	0.3911	No
1	0.9451	0.3696	No
2	0.5033	0.1860	No
3	0.7003	0.1303	Yes

Thus, we take $X = 3$. □

Remark: An easy argument says that the expected number of U's that are required to generate one realization of X is $\mathrm{E}[X + 1] = \lambda + 1$.

Remark: If $\lambda \geq 20$, we can use the normal approximation

$$\frac{X - \lambda}{\sqrt{\lambda}} \approx \text{Nor}(0, 1).$$

Algorithm POIS2 (for $\lambda \geq 20$)

$\alpha \leftarrow \sqrt{\lambda}$

Generate Z from $\text{Nor}(0, 1)$

Return $X = \max(0, \lfloor \lambda + \alpha Z + 0.5 \rfloor)$ (Note that this employs a "continuity correction.")

E.g., if $\lambda = 30$ and $Z = 1.46$, then $X = \lfloor 30.5 + \sqrt{30}(1.46) \rfloor = 38$.

Remark: Of course, another way to generate a Pois(λ) is simply to table the c.d.f. values like we did in an earlier discrete inverse transform example. This may be more efficient and accurate than the above methods — which is not to say that the A-R method isn't clever and pretty!

Outline

Composition Method

Idea: Suppose a RV actually comes from *two* RV's (sort of on top of each other). E.g., your plane can leave the airport gate late for two reasons — air traffic delays and maintenance delays, which compose the overall delay time.

The goal is to generate a RV with c.d.f.

$$F(x) = \sum_{j=1}^{\infty} p_j F_j(x),$$

where $p_j > 0$ for all j, $\sum_j p_j = 1$, and the $F_j(x)$'s are "easy" c.d.f.'s to generate from.

- Generate a positive integer J such that $P(J = j) = p_j$ for all j.
- Return X from c.d.f. $F_J(x)$.

Proof that X has c.d.f. $F(x)$: By the law of total probability,

$$\begin{aligned} P(X \leq x) &= \sum_{j=1}^{\infty} P(X \leq x | J = j) P(J = j) \\ &= \sum_{j=1}^{\infty} F_j(x) p_j = F(x). \quad \square \end{aligned}$$

Example: Laplace distribution (exponential distribution reflected off of the y-axis)

$$f(x) \equiv \begin{cases} \frac{1}{2}e^{x}, & x < 0 \\ \frac{1}{2}e^{-x}, & x > 0 \end{cases} \quad \text{and} \quad F(x) \equiv \begin{cases} \frac{1}{2}e^{x}, & x < 0 \\ 1 - \frac{1}{2}e^{-x}, & x > 0 \end{cases}$$

Meanwhile, let's decompose X into "negative exponential" and regular exponential distributions:

$$F_1(x) \equiv \begin{cases} e^{x} & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases} \quad \text{and} \quad F_2(x) \equiv \begin{cases} 0 & \text{if } x < 0 \\ 1 - e^{-x} & \text{if } x > 0 \end{cases}$$

Then

$$F(x) \;=\; \frac{1}{2}F_1(x) + \frac{1}{2}F_2(x),$$

so that we generate from $F_1(x)$ half the time, and $F_2(x)$ half.

We'll use inverse transform to solve $F_1(X) = e^X = U$ for X half the time, and $F_2(x) = 1 - e^{-X} = U$ the other half. Then

$$X \;\leftarrow\; \begin{cases} \ell\mathrm{n}(U) & \text{w.p. 1/2} \\ -\ell\mathrm{n}(U) & \text{w.p. 1/2} \end{cases} \qquad \square$$

Outline

Special-Case Techniques

Box–Muller Method: Here's a nice, easy way to generate standard normals.

Theorem: If U_1, U_2 are i.i.d. $\mathcal{U}(0,1)$, then

$$\begin{aligned} Z_1 &= \sqrt{-2\ell\mathrm{n}(U_1)}\cos(2\pi U_2) \\ Z_2 &= \sqrt{-2\ell\mathrm{n}(U_1)}\sin(2\pi U_2) \end{aligned}$$

are i.i.d. Nor(0,1).

Note that the trig calculations must be done in radians.

Proof Someday soon. □

Some interesting corollaries follow directly from Box–Muller.

Example: Note that

$$Z_1^2 + Z_2^2 \sim \chi^2(1) + \chi^2(1) \sim \chi^2(2).$$

But

$$\begin{aligned} Z_1^2 + Z_2^2 &= -2\ell\mathrm{n}(U_1)(\cos^2(2\pi U_2) + \sin^2(2\pi U_2)) \\ &= -2\ell\mathrm{n}(U_1) \\ &\sim \mathrm{Exp}(1/2). \end{aligned}$$

Thus, we've just proven that

$$\chi^2(2) \sim \mathrm{Exp}(1/2). \quad \square$$

Example: Note that

$$Z_2/Z_1 \sim \text{Nor}(0,1)/\text{Nor}(0,1) \sim \text{Cauchy} \sim t(1).$$

But

$$Z_2/Z_1 = \frac{\sqrt{-2\ell\text{n}(U_1)}\sin(2\pi U_2)}{\sqrt{-2\ell\text{n}(U_1)}\cos(2\pi U_2)} = \tan(2\pi U_2).$$

Thus, we've just proven that

$$\tan(2\pi U) \sim \text{Cauchy} \quad \text{(and, similarly, } \cot(2\pi U) \sim \text{Cauchy).}$$

Similarly,

$$Z_2^2/Z_1^2 = \tan^2(2\pi U) \sim t^2(1) \sim F(1,1).$$

(Did you know that?)

Polar Method — a little faster than Box–Muller.

1. Generate U_1, U_2 i.i.d. $\mathcal{U}(0,1)$.

   Let $V_i = 2U_i - 1$, $i = 1, 2$, and $W = V_1^2 + V_2^2$.

2. If $W > 1$, reject and go back to Step 1.

   O'wise, let $Y = \sqrt{-2\ell\mathrm{n}(W)/W}$, and accept $Z_i \leftarrow V_i Y$, $i = 1, 2$.

Then Z_1, Z_2 are i.i.d. Nor(0,1).

Order Statistics

Suppose that $X_1, X_2, \ldots, X_n$ are i.i.d. from some distribution with c.d.f. $F(x)$, and let $Y = \min\{X_1, \ldots, X_n\}$ with c.d.f. $G(y)$. (Y is called the first order stat.) Can we generate Y using just *one* $\mathcal{U}(0,1)$?

Yes! since the X_i's are i.i.d., we have

$$\begin{aligned} G(y) &= 1 - P(Y > y) = 1 - P(\min_i X_i > y) \\ &= 1 - P(\text{all } X_i\text{'s} > y) = 1 - [P(X_1 > y)]^n \\ &= 1 - [1 - F(y)]^n. \end{aligned}$$

Now do Inverse Transform: set $G(Y) = U$ and solve for Y. After a little algebra, get (don't be afraid)...

$$Y \;=\; F^{-1}\Big(1 - (1-U)^{1/n}\Big).$$

Example: Suppose $X_1, \ldots, X_n \sim \text{Exp}(\lambda)$. Then

$$G(y) \;=\; 1 - (e^{-\lambda y})^n \;=\; 1 - e^{-n\lambda y}.$$

Thus, $Y = \min_i\{X_i\} \sim \text{Exp}(n\lambda)$. (This is not a surprising result if you've taken a stochastic processes class.) In any case,
$Y = -\frac{1}{n\lambda}\ell\text{n}(U)$. $\quad\square$

We can do the same kind of thing for $Z = \max_i X_i$.

Other Quickies

If $X \sim \text{Nor}(\mu, \sigma^2)$, then e^X has the lognormal distribution.

$\chi^2(n)$ distribution: If $Z_1, Z_2, \ldots, Z_n$ are i.i.d. Nor(0,1), then $\sum_{i=1}^n Z_i^2 \sim \chi^2(n)$.

$t(n)$ distribution: If $Z \sim \text{Nor}(0,1)$ and $Y \sim \chi^2(n)$, and Z and Y are independent, then

$$\frac{Z}{\sqrt{Y/n}} \sim t(n).$$

Note that $t(1)$ is the Cauchy distribution.

$F(n,m)$ distribution: If $X \sim \chi^2(n)$ and $Y \sim \chi^2(m)$ and X and Y are independent, then $(X/n)/(Y/m) \sim F(n,m)$.

Outline

Bivariate Normal Distribution

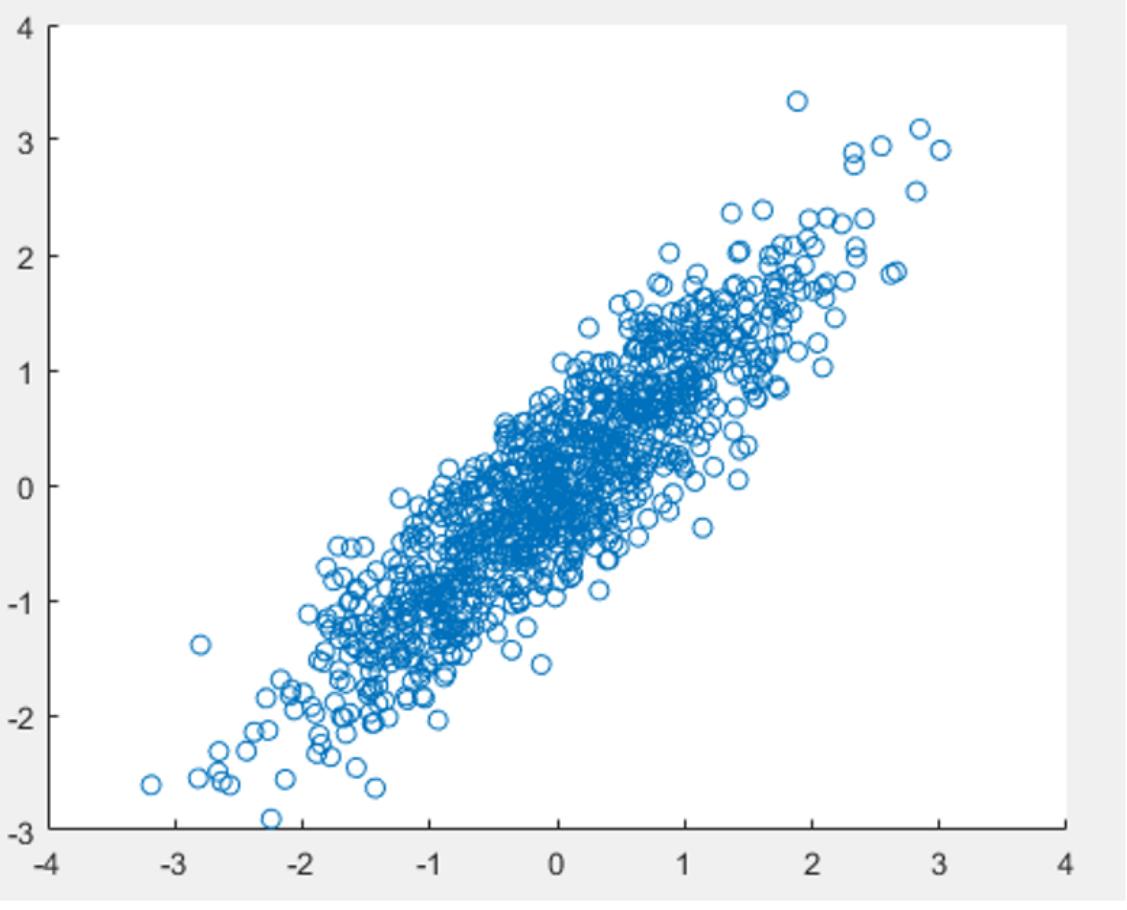
The random vector (X, Y) has the *bivariate normal distribution* with means $\mu_X = \mathrm{E}[X]$ and $\mu_Y = \mathrm{E}[Y]$, variances $\sigma_X^2 = \mathrm{Var}(X)$ and $\sigma_Y^2 = \mathrm{Var}(Y)$, and correlation $\rho = \mathrm{Corr}(X, Y)$ if it has joint p.d.f.

$$f(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left\{\frac{-\left[z_X^2(x) + z_Y^2(y) - 2\rho z_X(x) z_Y(y)\right]}{2(1-\rho^2)}\right\},$$

where $z_X(x) \equiv (x - \mu_X)/\sigma_X$ and $z_Y(y) \equiv (y - \mu_Y)/\sigma_Y$.

For example, heights and weights of people can be modeled as bivariate normal.

This distribution is easily generalized to the multivariate case.



MATLAB example:
Bivariate normal
means = 0,
variances = 1,
covariance = 0.9

Multivariate Normal Distribution

The random vector $\boldsymbol{X} = (X_1, \ldots, X_k)^{\mathrm{T}}$ has the *multivariate normal distribution* with mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)^{\mathrm{T}}$ and $k \times k$ covariance matrix $\Sigma = (\sigma_{ij})$ if it has p.d.f.

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left\{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right\}, \quad \boldsymbol{x} \in \mathbb{R}^k.$$

It turns out that

$$\mathrm{E}[X_i] = \mu_i, \quad \mathrm{Var}(X_i) = \sigma_{ii}, \quad \mathrm{Cov}(X_i, X_j) = \sigma_{ij}.$$

Notation: $\boldsymbol{X} \sim \mathrm{Nor}_k(\boldsymbol{\mu}, \Sigma)$.

In order to generate $\boldsymbol{X}$, let's start with a vector $\boldsymbol{Z} = (Z_1, \ldots, Z_k)$ of i.i.d. Nor(0,1) RV's. That is, suppose $\boldsymbol{Z} \sim \text{Nor}_k(\mathbf{0}, I)$, where I is the $k \times k$ identity matrix, and $\mathbf{0}$ is simply a vector of 0's.

Suppose we can find the (lower triangular) Cholesky matrix C such that $\Sigma = CC'$.

Similar to the one-dimensional result in which a linear function of a normal RV is still normal, it can be shown that $\boldsymbol{X} = \boldsymbol{\mu} + C\boldsymbol{Z}$ is multivariate normal with mean $\boldsymbol{\mu}$ and covariance matrix

$$\begin{aligned} \Sigma &\equiv \text{Cov}(\boldsymbol{X}) = \text{Cov}(\boldsymbol{\mu} + C\boldsymbol{Z}) \\ &= C\,\text{Cov}(\boldsymbol{Z})C' = CIC' = CC'. \end{aligned}$$

For $k = 2$, we can derive after a teensy bit of algebra,

$$C = \begin{pmatrix} \sqrt{\sigma_{11}} & 0 \\ \frac{\sigma_{12}}{\sqrt{\sigma_{11}}} & \sqrt{\sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}}} \end{pmatrix}.$$

Since $\boldsymbol{X} = \boldsymbol{\mu} + C\boldsymbol{Z}$, we have

$$X_1 = \mu_1 + c_{11} Z_1 = \mu_1 + \sqrt{\sigma_{11}}\, Z_1$$

$$X_2 = \mu_2 + c_{21} Z_1 + c_{22} Z_2 = \mu_2 + \frac{\sigma_{12}}{\sqrt{\sigma_{11}}} Z_1 + \sqrt{\sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}}}\, Z_2$$

Example: Suppose we want to generate a bivariate normal random vector (X_1, X_2) with mean $\boldsymbol{\mu} = (2, 5)$ and covariance matrix

$$\Sigma = \begin{pmatrix} 2 & -1 \\ -1 & 4 \end{pmatrix}.$$

First of all, let's find the Cholesky matrix C such that $\Sigma = CC'$,

$$C = \begin{pmatrix} \sqrt{2} & 0 \\ \frac{-1}{\sqrt{2}} & \sqrt{4 - \frac{1}{2}} \end{pmatrix} = \begin{pmatrix} 1.4142 & 0 \\ -0.7071 & 1.8708 \end{pmatrix}.$$

Suppose we sample $Z_1 = 1.2$ and $Z_2 = -0.3$. Then

$$\begin{aligned} X_1 &= \mu_1 + c_{11} Z_1 = 2 + 1.414(1.2) = 3.697 \\ X_2 &= \mu_2 + c_{21} Z_1 + c_{22} Z_2 \\ &= 5 - 0.707(1.2) + 1.871(-0.3) = 2.742. \quad \square \end{aligned}$$

The following algorithm computes C for general k:

Algorithm LTM

For $i = 1, \ldots, k$,

For $j = 1, \ldots, i-1$,

$$c_{ij} \leftarrow \left(\sigma_{ij} - \sum_{\ell=1}^{j-1} c_{i\ell} c_{j\ell}\right) / c_{jj}$$

$$c_{ji} \leftarrow 0$$

$$c_{ii} \leftarrow \left(\sigma_{ii} - \sum_{\ell=1}^{i-1} c_{i\ell}^2\right)^{1/2}$$

Once C has been computed, the multivariate normal RV $\boldsymbol{X} = \boldsymbol{\mu} + C\boldsymbol{Z}$ can easily be generated:

1. Generate $Z_1, Z_2, \ldots, Z_k \sim$ i.i.d. $\mathrm{Nor}(0, 1)$.

2. Let $X_i \leftarrow \mu_i + \sum_{j=1}^{i} c_{ij} Z_j$, $i = 1, 2, \ldots, k$.

3. Return $\boldsymbol{X} = (X_1, X_2, \ldots, X_k)$.

Outline

Generating Stochastic Processes

We'll now talk about...

- Markov Chains
- Poisson Arrivals
- Nonhomogeneous Poisson Arrivals
- Time Series
  - MA(1)
  - AR(1)
  - ARMA(p, q)
  - EAR(1)
  - Autoregressive Pareto
- $M/M/1$ Queue Waiting Times
- Brownian Motion

Markov Chains

Consider a time series having a certain number of states (e.g., sun / rain) that can transition from day to day.

Example: On Monday it's sunny, on Tues and Weds, it's rainy, etc.

Informally speaking, if today's weather only depends on yesterday, then you have a *Markov chain*.

Just do a simple example. Let $X_i = 0$ if it rains on day i; otherwise, $X_i = 1$. Denote the day-to-day transition probabilities by

$$P_{jk} \;=\; P(\text{state } k \text{ on day } i \mid \text{state } j \text{ on day } i-1), \quad j,k = 0,1.$$

Suppose that the probability state transition matrix is

$$\mathbf{P} = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix},$$

e.g., $P_{01} = P(\text{R today} \,|\, \text{S yesterday}) = 0.3$.

If it rains on Monday (rainy days and Mondays always get me down), let's simulate the rest of the work week.

To do so, we'll run daily Bern(P_{j0}) trials to determine if it's going to rain today given yesterday's weather, where

$$P_{j0} \equiv P(\text{R today} \,|\, \text{yesterday's weather was } X_{i-1} = j).$$

	$j = X_{i-1}$	P_{j0}	U_i	$U_i < P_{j0}$?	R/S
M	–	–	–	–	R
Tu	0	$P_{00} = 0.7$	0.62	Y	R
W	0	$P_{00} = 0.7$	0.03	Y	R
Th	0	$P_{00} = 0.7$	0.77	N	S
F	1	$P_{10} = 0.4$	0.91	N	S

Generating Poisson Arrivals

When the arrival rate is a *constant* λ, the interarrivals of a Poisson(λ) process are i.i.d. Exp(λ), and the arrival times are:

$$T_0 \leftarrow 0 \quad \text{and} \quad T_i \leftarrow T_{i-1} - \frac{1}{\lambda}\ell\mathrm{n}(U_i), \quad i \geq 1.$$

Now suppose that we want to generate a *fixed number* n of PP(λ) arrivals in a *fixed time interval* $[a, b]$. To do so, we note a theorem stating that the joint distribution of the n arrivals is the same as the joint distribution of the order statistics of n i.i.d. $\mathcal{U}(a, b)$ RV's.

Generate i.i.d. $U_1, \ldots, U_n$ from $\mathcal{U}(0, 1)$

Sort the U_i's: $U_{(1)} < U_{(2)} < \cdots < U_{(n)}$

Set the arrival times to $T_i \leftarrow a + (b - a)U_{(i)}$

Nonhomogeneous Poisson Process — *nonstationary* arrivals

Same assumptions as regular Poisson process except the arrival rate λ isn't a constant, so stationary increments doesn't apply.

Let

$$\lambda(t) = \text{ rate (intensity) function at time } t,$$

$$N(t) = \text{ number of arrivals during } [0, t].$$

Then

$$N(b) - N(a) \sim \text{ Poisson} \left(\int_a^b \lambda(t)\, dt \right).$$

Example: Suppose that the arrival pattern to the Waffle House over a certain time period is a NHPP with $\lambda(t) = t^2$. Find the probability that there will be exactly 4 arrivals between times $t = 1$ and 2.

First of all, the number of arrivals in that time interval is

$$N(2) - N(1) \sim \text{Pois}\left(\int_1^2 t^2\, dt\right) \sim \text{Pois}(7/3).$$

Thus,

$$P\big(N(2) - N(1) = 4\big) = \frac{e^{-7/3}(7/3)^4}{4!} = 0.120. \quad \square$$

Incorrect NHPP Algorithm [it can "skip" intervals with large $\lambda(t)$]

$T_0 \leftarrow 0$; $i \leftarrow 0$

Repeat

Generate U from $\mathcal{U}(0,1)$

$T_{i+1} \leftarrow T_i - \frac{1}{\lambda(T_i)}\ell\mathrm{n}(U)$

$i \leftarrow i + 1$

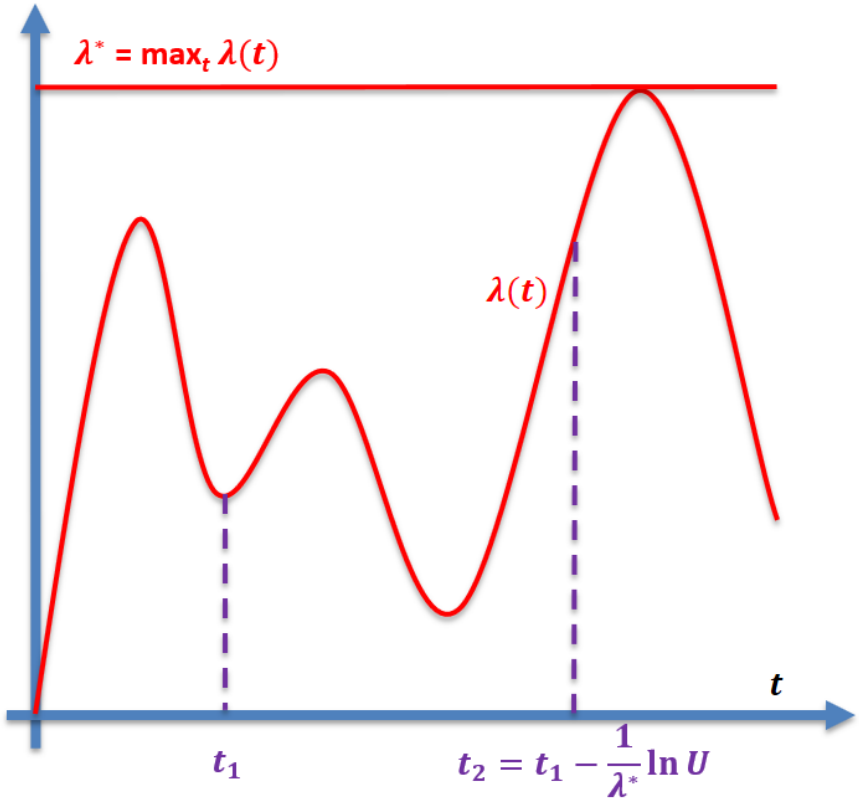
Don't use this algorithm! — the arrival rate $\lambda(T_i)$ doesn't keep pace with changes in $\lambda(t)$ that might occur between the current arrival at time T_i and the next arrival at time T_{i+1}.

Whatever shall we do?

The *Thinning Algorithm*...

- Assumes that $\lambda^\star \equiv \max_t \lambda(t) < \infty$;
- Generates *potential arrivals* at the rate $\lambda^\star$; and
- Accepts (keeps) a potential arrival at time t with probability $\lambda(t)/\lambda^\star$.

The figure below illustrates two *potential arrivals* at times t_1 and t_2. We keep potential arrival i with probability $\lambda(t_i)/\lambda^\star$.



Let T_i denote the ith arrival that we *actually keep*. E.g., if we reject the first potential arrival but keep the second, then $T_1 \leftarrow t_2$.

Thinning Algorithm

$T_0 \leftarrow 0; i \leftarrow 0$

Repeat

$t \leftarrow T_i$

Repeat

Generate U, V from $\mathcal{U}(0,1)$

$t \leftarrow t - \frac{1}{\lambda^\star}\ell\mathrm{n}(U)$

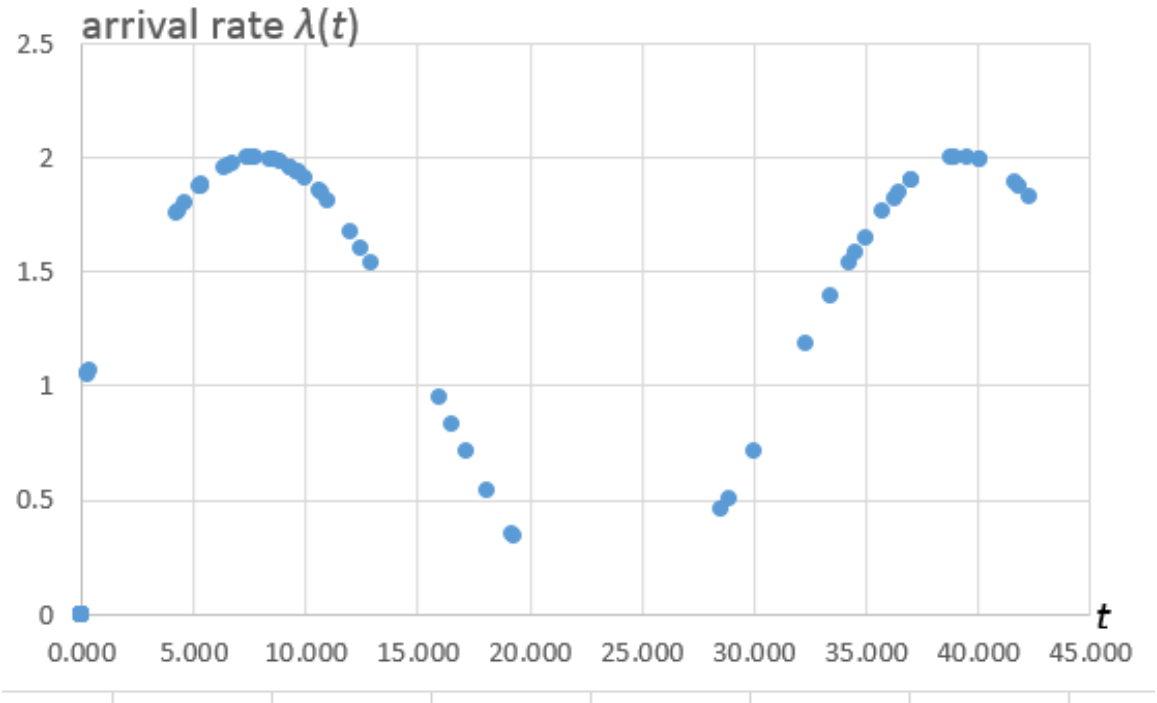
until $V \leq \lambda(t)/\lambda^\star$

$i \leftarrow i + 1$

$T_i \leftarrow t$

Example: Let's simulate NHPP arrivals in which $\lambda(t) = 1 + \sin(t/5)$, so that $\lambda^\star = 2$. The figure below plots arrival points.

The t-values of the blue dots represent actual arrival times; and the y-values are placed to illustrate the values of $\lambda(t)$ at those points.



Remark: Thinning is an Acceptance-Rejection method. Very nice, but if $\lambda^\star$ is significantly higher than most of the values that $\lambda(t)$ takes, then thinning can be inefficient.

Remark: Arena and various other simulation languages use a different method when generating NHPP arrivals from an Arrival Schedule.

That's because such a schedule gives $\lambda(t)$ as a *step function* that only changes occasionally, say, every hour. In this case, it's possible to take clever advantage the exponential's memoryless property to avoid the use of thinning.

First-Order Moving Average Process

An MA(1) is a time series process is defined by

$$Y_i \;=\; \varepsilon_i + \theta\varepsilon_{i-1}, \quad \text{for } i = 1, 2, \ldots,$$

where θ is a constant and the ε_i's are i.i.d. Nor$(0, 1)$ RV's that are independent of Y_0.

The MA(1) is a popular tool for modeling and detecting trends.

It's easy to show that $Y_1, Y_2, \ldots$ are all $\text{Nor}(0, 1 + \theta^2)$. But the Y_i's aren't independent!

Define the *covariance function*, $R_k \equiv \text{Cov}(Y_i, Y_{i+k})$, $k = 0, \pm 1, \pm 2, \ldots$.

For the MA(1), we have $R_0 = \text{Var}(Y_i) = 1 + \theta^2$, $R_1 = \text{Cov}(Y_i, Y_{i+1}) = \theta$, and $R_k = 0$ for $k \geq 2$. So the covariances die off pretty quickly.

How to generate? Start with $\varepsilon_0 \sim \text{Nor}(0, 1)$. Then generate $\varepsilon_1 \sim \text{Nor}(0, 1)$ to get Y_1, $\varepsilon_2 \sim \text{Nor}(0, 1)$ to get Y_2, etc.

First-Order Autoregressive Process

An AR(1) process is defined by

$$Y_i = \phi Y_{i-1} + \varepsilon_i, \quad \text{for } i = 1, 2, \ldots,$$

where $-1 < \phi < 1$, $Y_0 \sim \text{Nor}(0, 1)$, and the ε_i's are i.i.d. $\text{Nor}(0, 1 - \phi^2)$ RV's that are independent of Y_0.

This is used to model lots of real-world stuff.

As defined, the Y_i's are all Nor(0,1), but (similar to the MA(1)), they aren't independent.

The AR(1) has covariance function

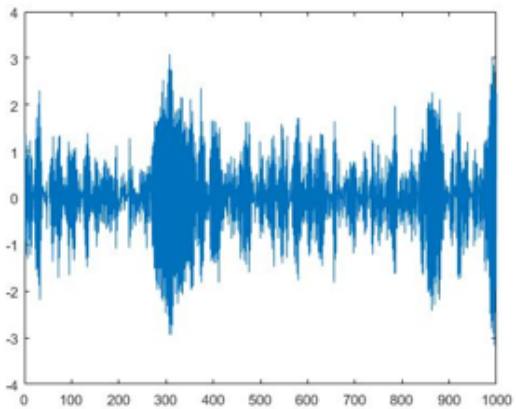
$$R_k \;=\; \mathrm{Cov}(Y_i, Y_{i+k}) \;=\; \phi^{|k|}. \quad \text{for all } k = 0, \pm 1, \pm 2, \ldots.$$

If ϕ is close to one, you get highly positively correlated Y_i's. If ϕ is close to zero, the Y_i's are nearly independent.

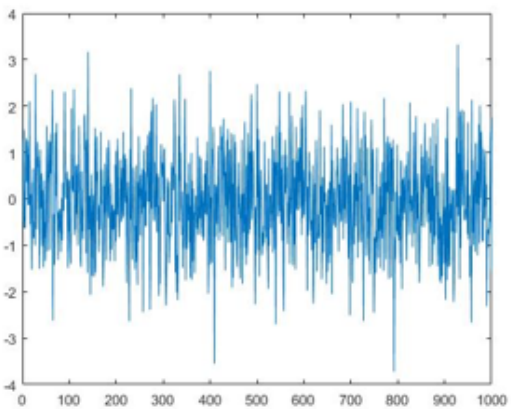
How to generate? Start with $Y_0 \sim \mathrm{Nor}(0,1)$ and $\varepsilon_1 \sim \sqrt{1-\phi^2}\,\mathrm{Nor}(0,1)$ to get $Y_1 = \phi Y_0 + \varepsilon_1$.

Then generate $\varepsilon_2 \sim \sqrt{1-\phi^2}\,\mathrm{Nor}(0,1)$ to get $Y_2 = \phi Y_1 + \varepsilon_2$, etc.

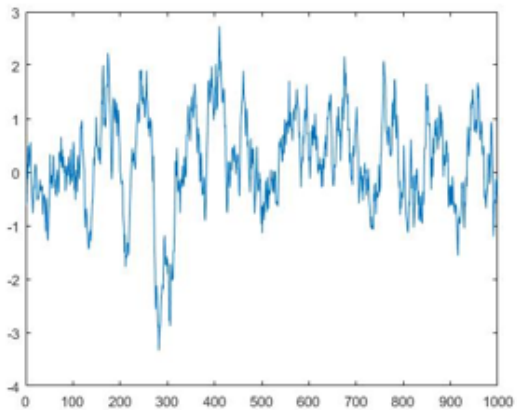
AR(1) pix



AR(1), $\phi = -0.95$



AR(1), $\phi = 0$



AR(1), $\phi = 0.95$

ARMA(p, q) Process

An obvious generalization of the MA(1) and AR(1) processes is the ARMA(p, q), which consists of a pth order AR and a qth order MA, which we will simply define (without stating properties):

$$Y_i = \sum_{j=1}^{p} \phi_j Y_{i-j} + \varepsilon_i + \sum_{j=1}^{q} \theta_j \varepsilon_{i-j}, \quad i = 1, 2, \ldots,$$

where the ϕ_j's, θ_j's, and $\mathrm{Var}(\varepsilon_i)$, as well as the initial RVs $Y_0, Y_{-1}, \ldots, Y_{1-p}$, are chosen so as to assure that the process doesn't explode.

Such processes are used in a variety of modeling and forecasting applications.

First-Order Exponential Autoregressive Process

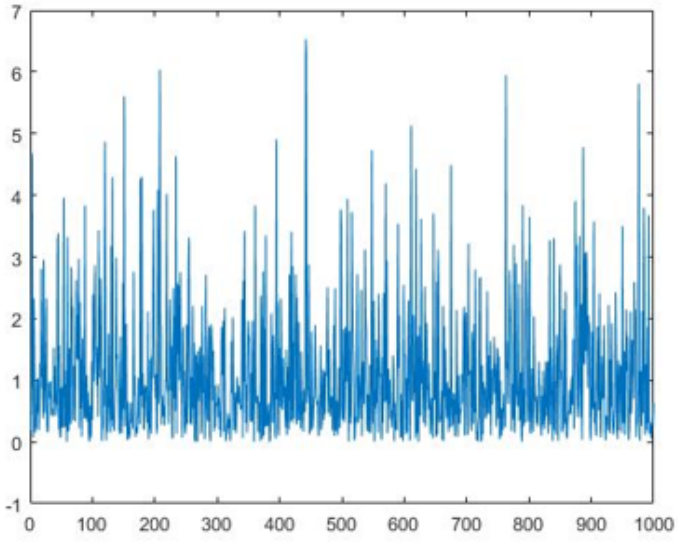
An EAR(1) process (Lewis 1980) is defined by

$$Y_i = \begin{cases} \phi Y_{i-1}, & \text{w.p. } \phi \\ \phi Y_{i-1} + \varepsilon_i, & \text{w.p. } 1-\phi \end{cases},$$

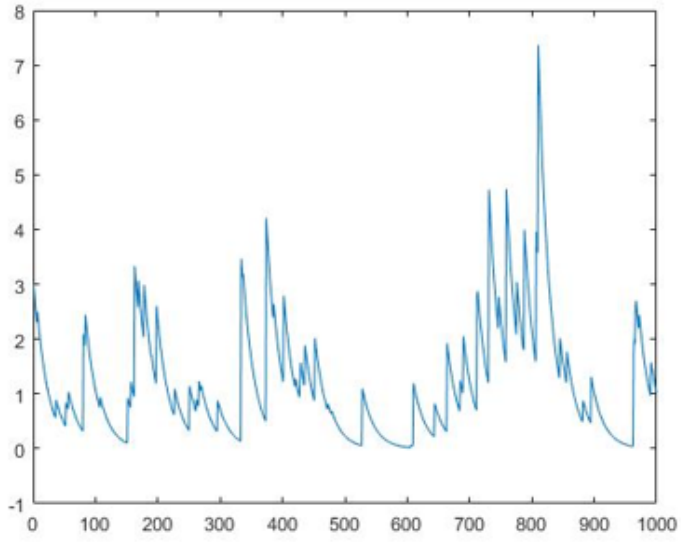
for $i = 1, 2, \ldots$, where $0 \leq \phi < 1$, $Y_0 \sim \text{Exp}(1)$, and the ε_i's are i.i.d. Exp(1) RV's that are independent of Y_0.

The EAR(1) has the same covariance structure as the AR(1), except that $0 \leq \phi < 1$, that is, $\text{Cov}(Y_i, Y_{i+k}) = \phi^{|k|}$ for all $k = 0, \pm 1, \pm 2, \ldots$.

EAR(1) pix



EAR(1), $\phi = 0.0$



EAR(1), $\phi = 0.95$

Autoregressive Pareto (ARtoP) Process

Now let's see how to generate a series of correlated Pareto RV's. First of all, a RV X has the *Pareto distribution* with parameters $\lambda > 0$ and $\beta > 0$ if it has c.d.f.

$$F_X(x) \;=\; 1 - (\lambda/x)^\beta, \quad \text{for } x \geq \lambda.$$

The Pareto is a heavy-tailed distribution that has a variety of uses in statistical modeling, and for which

$$\mathrm{E}[X] \;=\; \frac{\beta\lambda}{\beta - 1} \quad \text{for } \beta > 1 \quad \text{and}$$

$$\mathrm{Var}(X) \;=\; \frac{\beta\lambda^2}{(\beta-1)^2(\beta-2)} \quad \text{for } \beta > 2.$$

In order to obtain the ARP process, let's start off with a regular AR(1) with normal noise,

$$Y_i = \rho Y_{i-1} + \varepsilon_i, \quad \text{for } i = 1, 2, \ldots,$$

where $-1 < \rho < 1$, $Y_0 \sim \text{Nor}(0, 1)$, and the ε_i's are i.i.d. $\text{Nor}(0, 1 - \rho^2)$ and independent of Y_0. Note that $Y_0, Y_1, Y_2, \ldots$ are marginally Nor(0,1) but correlated.

Feed this process into the Nor(0,1) c.d.f. $\Phi(\cdot)$ to obtain correlated Unif(0,1) RV's, $U_i = \Phi(Y_i)$, $i = 1, 2, \ldots$.

Now feed the correlated U_i's into the inverse of the Pareto c.d.f. to obtain correlated Pareto RV's:

$$X_i = F_X^{-1}(U_i) = F_X^{-1}(\Phi(Y_i)) = \frac{\lambda}{[1 - \Phi(Y_i)]^{1/\beta}}, \quad i = 1, 2, \ldots.$$

M/M/1 Queue

Consider a single-server queue with customers arriving according to a Poisson(λ) process, standing in line with a FIFO discipline, and then getting served in an $\mathrm{Exp}(\mu)$ amount of time.

Let I_{i+1} denote the interarrival time between the ith and $(i+1)$st customers; let S_i be the ith customer's service time; and let W_i^Q denote the ith customer's wait before service.

Lindley gives a very nice way to generate a series of waiting times for this simple example:

$$W_{i+1}^Q \;=\; \max\{W_i^Q + S_i - I_{i+1},\; 0\}.$$

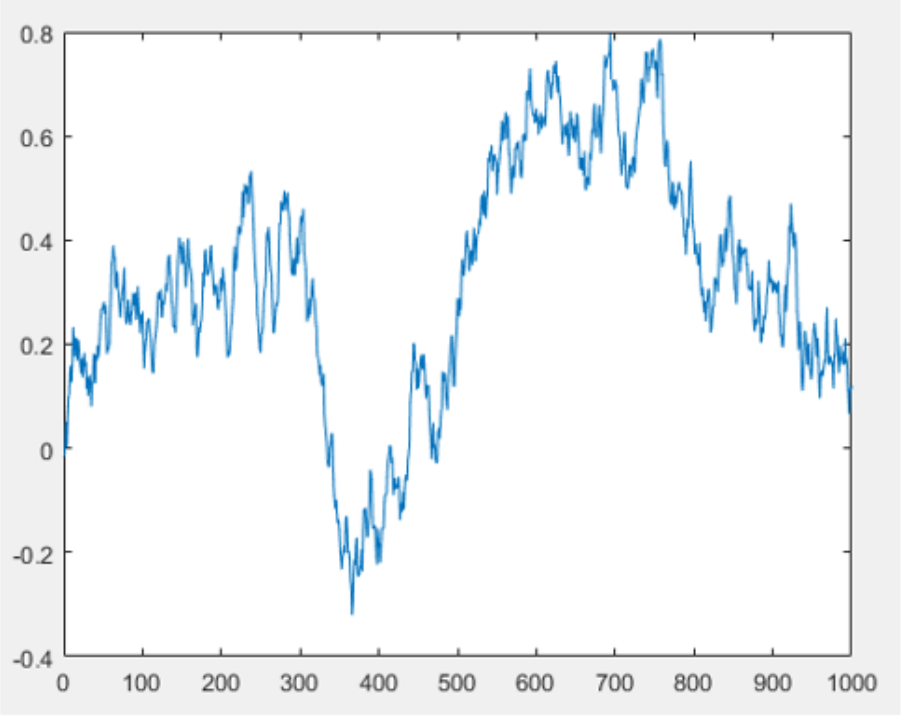
And similarly, the total time in system, $W_i = W_i^Q + S_i$, is

$$W_{i+1} \;=\; \max\{W_i - I_{i+1},\; 0\} + S_{i+1}.$$

Brownian Motion

Discovered by Robert Brown; analyzed rigorously by Einstein; and mathematics established by Wiener (also called *Wiener process*).

Widely used in everything from financial analysis to queueing theory to statistics to other OR/IE application areas. Incredibly important.



Definition: The continuous-time stochastic process $\{\mathcal{W}(t), t \geq 0\}$ is a *standard Brownian motion* process if:

1. $\mathcal{W}(0) = 0$.
2. $\mathcal{W}(t) \sim \text{Nor}(0, t)$.
3. $\{\mathcal{W}(t),\ t \geq 0\}$ has stationary and independent increments.

Increments: Anything like $\mathcal{W}(b) - \mathcal{W}(a)$.

Stationary increments informally means that the distribution of $\mathcal{W}(t+h) - \mathcal{W}(t)$ only depends on h.

Independent increments: If $a < b < c < d$, then $\mathcal{W}(d) - \mathcal{W}(c)$ is indep of $\mathcal{W}(b) - \mathcal{W}(a)$.

The definition gives rise to some cool properties.

Theorem: $\mathrm{Cov}(\mathcal{W}(s), \mathcal{W}(t)) = \min(s, t)$.

Proof: Suppose that $s < t$. Then

$$\begin{aligned}
\mathrm{Cov}(\mathcal{W}(s), \mathcal{W}(t)) &= \mathrm{Cov}\Big(\mathcal{W}(s), \mathcal{W}(t) - \mathcal{W}(s) + \mathcal{W}(s)\Big) \\
&= \mathrm{Cov}\Big(\mathcal{W}(s), \mathcal{W}(t) - \mathcal{W}(s)\Big) + \mathrm{Var}(\mathcal{W}(s)) \\
&= 0 + s \;=\; s,
\end{aligned}$$

by independent increments and the fact that $\mathcal{W}(s) \sim \mathrm{Nor}(0, s)$. $\square$

How do you generate BM on a computer?

Suppose that $Y_1, Y_2, \ldots$ is any sequence of i.i.d. RV's with mean zero and variance 1. (To some extent, the Y_i's don't even have to be independent!) *Donsker's Central Limit Theorem* says that

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{\lfloor nt \rfloor} Y_i \xrightarrow{d} \mathcal{W}(t) \quad \text{as } n \to \infty,$$

where $\xrightarrow{d}$ denotes convergence in distribution as n gets big.

Remark: The regular CLT is just for the case $t = 1$, so that the thing converges to $\mathcal{W}(1) \sim \text{Nor}(0, 1)$.

Donsker is a *much* more general *process CLT*. Now the thing converges to an entire BM process $\{\mathcal{W}(t)\}$ for *all* t — not just a single, wussy Nor(0,1) RV.

Here's a way to construct BM:

One choice that works well is to take $Y_i = \pm 1$, each with probability 1/2. Take n at least 100, $t = 1/n, 2/n, \ldots, n/n$, and calculate $\mathcal{W}(1/n), \mathcal{W}(2/n), \ldots, \mathcal{W}(n/n)$.

Another choice is simply to take $Y_i \sim \text{Nor}(0, 1)$, $i = 1, 2, \ldots$.

Exercise: Let's construct some BM! First, pick some "large" value of n and start with $\mathcal{W}(0) = 0$. Then

$$\mathcal{W}\Big(\frac{i}{n}\Big) \;=\; \mathcal{W}\Big(\frac{i-1}{n}\Big) + \frac{Y_i}{\sqrt{n}}, \quad i = 1, 2, \ldots.$$

Here are some miscellaneous properties of Brownian Motion:

- BM is continuous everywhere, but has no derivatives! (Pretty deep result.)
- Area under $\mathcal{W}(t)$ is normal: $\int_0^1 \mathcal{W}(t)\,dt \sim \text{Nor}(0, \frac{1}{3})$.
- A *Brownian bridge*, $\mathcal{B}(t)$, is conditioned BM such that $\mathcal{W}(0) = \mathcal{W}(1) = 0$.
- $\text{Cov}(\mathcal{B}(s), \mathcal{B}(t)) = \min(s,t) - st$.
- $\int_0^1 \mathcal{B}(t)\,dt \sim \text{Nor}(0, \frac{1}{12})$.

Geometric Brownian Motion

The GBM process

$$S(t) \;=\; S(0)\exp\left\{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma\mathcal{W}(t)\right\}, \quad t \geq 0,$$

is often used to model stock prices, where μ is related to the "drift" of the stock price, σ is its volatility, and $S(0)$ is the initial price.

In addition, we can use GBM to estimate option prices. E.g., a European call option C permits its owner, who pays an up-front fee for the privilege, to purchase the stock at a pre-agreed strike price k, at a pre-determined expiry date T. Its "value" is

$$\mathrm{E}[C] \;=\; e^{-rT}\mathrm{E}\Big[\big(S(T) - k\big)^+\Big],$$

where $x^+ = \max\{0, x\}$ and $\mu \leftarrow r$, the "risk-free" interest rate.

Exercise: Let's estimate the value $\mathrm{E}[C]$ of a stock option. Pick your favorite values of r, σ, T, k, and off you go!

Lots of ways to actually do this. I would recommend that you directly simulate multiple simulation replications of the BM $\mathcal{W}(T)$, and then take the sample average of the $e^{-rT}(S(T) - k)^+$ values.

But there are other ways: You can just simulate the distribution of $S(T)$ directly (it's lognormal), or you can actually look up the exact "Black–Scholes" answer (see below).

There are also many, many generalizations of this problem that we can talk about some other time.

How to Win a Nobel Prize

The Black–Scholes European call option value is

$$\begin{aligned}
\mathrm{E}[C] &= e^{-rT}\mathrm{E}[S(T)-k]^+ \\
&= e^{-rT}\mathrm{E}\left[S(0)\exp\left\{\left(r-\frac{\sigma^2}{2}\right)T+\sigma\mathcal{W}(T)\right\}-k\right]^+ \\
&= e^{-rT}\int_{-\infty}^{\infty}\left[S(0)\exp\left\{\left(r-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}z\right\}-k\right]^+\phi(z)\,dz \\
&= S(0)\Phi(b+\sigma\sqrt{T})-ke^{-rT}\Phi(b) \quad \text{(after lots of algebra)},
\end{aligned}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the Nor(0,1) p.d.f. and c.d.f., and

$$b \equiv \frac{rT-\frac{\sigma^2T}{2}-\ell\mathrm{n}(k/S(0))}{\sigma\sqrt{T}}. \quad \square$$

Now get your tickets to Norway or Sweden or wherever they give out the Nobel Prize...