# Statistics 1 Unit 3: Simulation

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Kurt Hornik

EQUIS ACCREDITED AACSB ACCREDITED ASSOCIATION OF MBAS AMBA ACCREDITED

# Outline

- Sums and mixtures
- Stochastic processes
- Monte Carlo estimation

# Mixtures

Suppose that $Y$ is a discrete random variable such that

$$(X|Y = i) \sim F_i.$$

I.e., conditional on $Y = i$, $X$ has distribution function $F_i$.

What is the distribution of $X$?

# Mixtures

Suppose that $Y$ is a discrete random variable such that

$$(X|Y = i) \sim F_i.$$

I.e., conditional on $Y = i$, $X$ has distribution function $F_i$.

What is the distribution of $X$?

We easily find

$$\mathbb{P}(X \leq x) = \sum_i \mathbb{P}(X \leq x|Y = i)\mathbb{P}(Y = i) = \sum_i \mathbb{P}(Y = i)F_i(x).$$

# Mixtures

Suppose that $Y$ is a discrete random variable such that

$$(X|Y = i) \sim F_i.$$

I.e., conditional on $Y = i$, $X$ has distribution function $F_i$.

What is the distribution of $X$?

We easily find

$$\mathbb{P}(X \leq x) = \sum_i \mathbb{P}(X \leq x|Y = i)\mathbb{P}(Y = i) = \sum_i \mathbb{P}(Y = i)F_i(x).$$

This is the *discrete mixture* of $F_1, \ldots, F_k$ with mixture probabilities (weights) $\theta_1 = \mathbb{P}(Y = 1), \ldots, \theta_k = \mathbb{P}(Y = k)$.

# Mixtures

The above just write the marginal distribution of $X$ in terms of the conditional distribution of $X$ given $Y$ and the marginal distribution of $Y$.

We can do something similar in case $Y$ has a continuous distribution with density $g$:

$$\mathbb{P}(X \leq x) = \int \mathbb{P}(X \leq x | Y = y) g(y) \, dy.$$

This write the CDF of $X$ as a *continuous mixture* of the conditional distributions of $X$ given $Y$, weighted by the marginal density of $Y$.

# Mixtures

How can we draw from a mixture distribution?

# Mixtures

How can we draw from a mixture distribution?

Simple:

- First draw $Y$ from its marginal distribution, giving $y$.

# Mixtures

How can we draw from a mixture distribution?

Simple:

- First draw $Y$ from its marginal distribution, giving $y$.
- Then draw $X$ from its conditional distribution given $y$.

# Convolutions

Suppose $X \sim F$ and $Y \sim G$ are independent.

What is the distribution of $Z = X + Y$?

# Convolutions

Suppose $X \sim F$ and $Y \sim G$ are independent.

What is the distribution of $Z = X + Y$?

If $G$ has density $g$, we find that

$$\begin{aligned}
\mathbb{P}(X + Y \leq z) &= \int \mathbb{P}(X + Y \leq z | Y = y)\, g(y)\, dy \\
&= \int \mathbb{P}(X \leq z - y | Y = y)\, g(y)\, dy \\
&= \int F(z - y)\, g(y)\, dy.
\end{aligned}$$

# Convolutions

If $F$ has density $f$ and everything is fine, the distribution of $Z$ has density

$$\frac{d}{dz}\mathbb{P}(Z \leq z) = \int f(z-y)\,g(y)\,dy = \int g(z-x)f(x)\,dx.$$

This is the *convolution* of the densities $f$ and $g$.

# Convolutions

If $F$ has density $f$ and everything is fine, the distribution of $Z$ has density

$$\frac{d}{dz}\mathbb{P}(Z \le z) = \int f(z-y)\,g(y)\,dy = \int g(z-x)\,f(x)\,dx.$$

This is the *convolution* of the densities $f$ and $g$.

In general, the CDF of $Z$ is given by the convolution

$$\int F(z-y)\,dG(y) = \int G(z-x)\,dF(x)$$

with these integrals as explained in the probability course.

# Convolutions

How can we draw from the convolution of two distributions $F$ and $G$?

# Convolutions

How can we draw from the convolution of two distributions $F$ and $G$?

Simple:

- *Independently* draw $X \sim F$ and $Y \sim G$

# Convolutions

How can we draw from the convolution of two distributions $F$ and $G$?

Simple:

- *Independently* draw $X \sim F$ and $Y \sim G$
- Return $Z = X + Y$.

# Convolutions

How can we draw from the convolution of two distributions $F$ and $G$?

Simple:

- *Independently* draw $X \sim F$ and $Y \sim G$
- Return $Z = X + Y$.

(Well, that’s what convolutions are.)

# Example

Take $X_1 \sim \text{gamma}(2, 2)$ and $X_2 \sim \text{gamma}(2, 4)$ and compare the distribution of the sum to the discrete mixture with weights 1/2.

# Example

Take $X_1 \sim \mathrm{gamma}(2, 2)$ and $X_2 \sim \mathrm{gamma}(2, 4)$ and compare the distribution of the sum to the discrete mixture with weights 1/2.

E.g., using a sample size of $n = 1000$:

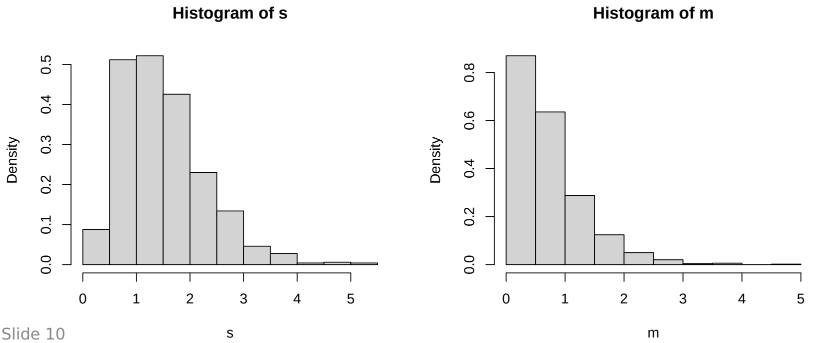
```
R> n <- 1000
R> x1 <- rgamma(n, 2, 2)
R> x2 <- rgamma(n, 2, 4)
R> ## Convolution:
R> s <- x1 + x2
R> ## Mixture:
R> u <- runif(n)
R> g <- (u > 0.5)
R> m <- g * x1 + (1 - g) * x2
```

Note how the mixture is done: if $g = 0$, we take $x2$; otherwise, $x1$.

# Example

Compare:

```
R> op <- par(mfcol = c(1, 2))
R> hist(s, probability = TRUE)
R> hist(m, probability = TRUE)
R> par(op)
```



# Note 1

Note that mixtures and convolutions are two fundamentally different things!

# Note 1

Note that mixtures and convolutions are two fundamentally different things!

Mixtures are weighted sums (actually, means) of CDFs, whereas convolutions relate to the CDFs of independent sums of random variables!

# Note 2

Of course, for discrete mixtures, we can always do

$$X = \sum_{i=1}^{k} I(Y = i) X_i, \qquad X_1 \sim F_1, \ldots, X_k \sim F_k$$

but that's actually not very efficient (only need to draw that from the $F_i$ where $y = i$.

# Note 2

Of course, for discrete mixtures, we can always do

$$X = \sum_{i=1}^{k} I(Y = i) X_i, \qquad X_1 \sim F_1, \dots, X_k \sim F_k$$

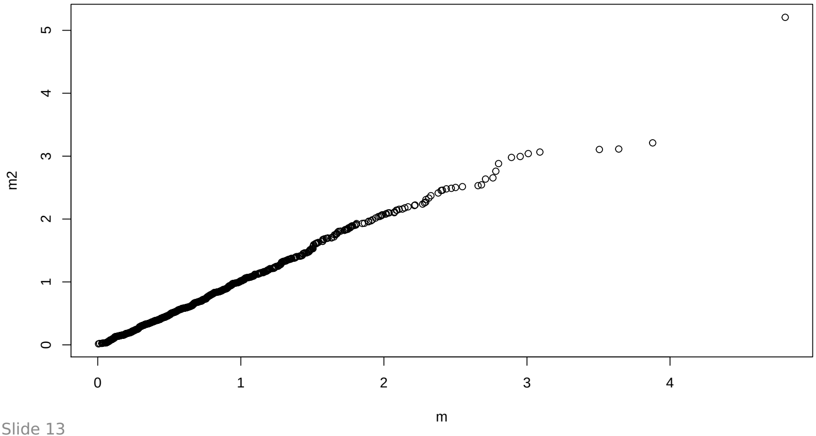
but that's actually not very efficient (only need to draw that from the $F_i$ where $y = i$.

In R, often can conveniently use the fact the the d-p-q-r functions are vectorized. E.g., for our example:

```
R> m2 <- rgamma(n, 2, ifelse(u > 0.5, 2, 4))
```

# Note 2

```
R> qqplot(m, m2)
```



# Outline

- Sums and mixtures
- Stochastic processes
- Monte Carlo estimation

# Introduction

For QFin, we also need to be able to simulate continuous time stochastic processes, i.e., families of random variables $(X(t))$ indexed by a continuous time parameter $t$.

# Introduction

For QFin, we also need to be able to simulate continuous time stochastic processes, i.e., families of random variables $(X(t))$ indexed by a continuous time parameter $t$.

Of course, the most important such process is the Wiener process (Brownian motion): but that's not so easy, so will be discussed in later courses.

# Introduction

For QFin, we also need to be able to simulate continuous time stochastic processes, i.e., families of random variables $(X(t))$ indexed by a continuous time parameter $t$.

Of course, the most important such process is the Wiener process (Brownian motion): but that’s not so easy, so will be discussed in later courses.

Here, we discuss the (homogeneous) Poisson process (and in the homeworks, compound Poisson processes).

# Renewal processes

Suppose we have light bulbs with random (technical) life times $\rho_i$.

# Renewal processes

Suppose we have light bulbs with random (technical) life times $\rho_i$.

At $t = 0$, we start using the first light bulb, which lasts time $\rho_1$.

# Renewal processes

Suppose we have light bulbs with random (technical) life times $\rho_i$.

At $t = 0$, we start using the first light bulb, which lasts time $\rho_1$.

At $t = \rho_1$, light bulb 1 stops working, and we replace it by light bulb 2, which lasts time $\rho_2$, i.e., until $\rho_1 + \rho_2$.

# Renewal processes

Suppose we have light bulbs with random (technical) life times $\rho_i$.

At $t = 0$, we start using the first light bulb, which lasts time $\rho_1$.

At $t = \rho_1$, light bulb 1 stops working, and we replace it by light bulb 2, which lasts time $\rho_2$, i.e., until $\rho_1 + \rho_2$.

At $t = \rho_1 + \rho_2$, light bulb 2 stops working, and we replace it by light bulb 3, which lasts time $\rho_3$, i.e., until $\rho_1 + \rho_2 + \rho_3$.

# Renewal processes

Suppose we have light bulbs with random (technical) life times $\rho_i$.

At $t = 0$, we start using the first light bulb, which lasts time $\rho_1$.

At $t = \rho_1$, light bulb 1 stops working, and we replace it by light bulb 2, which lasts time $\rho_2$, i.e., until $\rho_1 + \rho_2$.

At $t = \rho_1 + \rho_2$, light bulb 2 stops working, and we replace it by light bulb 3, which lasts time $\rho_3$, i.e., until $\rho_1 + \rho_2 + \rho_3$.

At $t = \rho_1 + \cdots + \rho_n$, light bulb $n$ stops working, and we replace it by light bulb $n + 1$, which lasts time $\rho_{n+1}$, i.e., until $\rho_1 + \cdots + \rho_{n+1}$.

# Renewal processes

Suppose we have light bulbs with random (technical) life times $\rho_i$.

At $t = 0$, we start using the first light bulb, which lasts time $\rho_1$.

At $t = \rho_1$, light bulb 1 stops working, and we replace it by light bulb 2, which lasts time $\rho_2$, i.e., until $\rho_1 + \rho_2$.

At $t = \rho_1 + \rho_2$, light bulb 2 stops working, and we replace it by light bulb 3, which lasts time $\rho_3$, i.e., until $\rho_1 + \rho_2 + \rho_3$.

At $t = \rho_1 + \cdots + \rho_n$, light bulb $n$ stops working, and we replace it by light bulb $n + 1$, which lasts time $\rho_{n+1}$, i.e., until $\rho_1 + \cdots + \rho_{n+1}$.

What is the number $N(t)$ of replacements we need to make up to time $t$?

# Renewal processes

Write $\tau_0 = 0$ and

$$\tau_n = \rho_1 + \cdots + \rho_n$$

for the time when we replace light bulb $n$.

# Renewal processes

Write $\tau_0 = 0$ and

$$\tau_n = \rho_1 + \cdots + \rho_n$$

for the time when we replace light bulb $n$.

Clearly,

$$N(t) = \begin{cases} 0, & \tau_0 \leq t < \tau_1, \\ 1, & \tau_1 \leq t < \tau_2, \\ \vdots & \vdots \\ n, & \tau_n \leq t < \tau_{n+1}, \\ \vdots & \vdots \end{cases}$$

# Renewal processes

Equivalently,

$$N(t) = \sum_n I(\tau_n \leq t)$$

($N(t)$ is the number of replacements up to $t$), and

$$N(t) \geq n \Leftrightarrow \tau_n \leq t$$

(there were at least $n$ replacements up to $t$ if and only if the time of replacement $n$ is not after $t$).

# Renewal processes

Equivalently,

$$N(t) = \sum_n I(\tau_n \leq t)$$

($N(t)$ is the number of replacements up to $t$), and

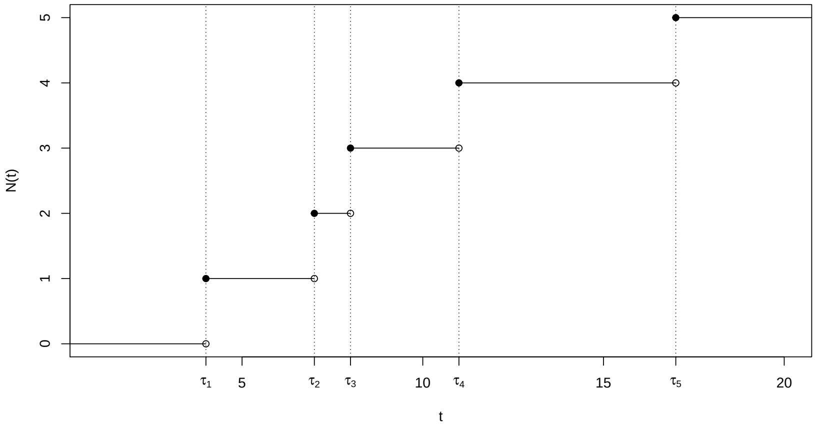
$$N(t) \geq n \Leftrightarrow \tau_n \leq t$$

(there were at least $n$ replacements up to $t$ if and only if the time of replacement $n$ is not after $t$).

If the $\rho_i$ are i.i.d., we call the discrete-time sequence $(\tau_n, n = 0, 1, \ldots)$ or equivalently, the continuous-time process $(N(t), t \geq 0)$ a *renewal process*.

# Renewal processes

Graphical illustration:



# Renewal processes

Of course, there is nothing specific to light bulbs or renewals in the above.

# Renewal processes

Of course, there is nothing specific to light bulbs or renewals in the above.

Consider a sequence of events, with event $n$ at time $\tau_n$.

# Renewal processes

Of course, there is nothing specific to light bulbs or renewals in the above.

Consider a sequence of events, with event $n$ at time $\tau_n$.

Write $\tau_0 = 0$, and $\rho_n = \tau_n - \tau_{n-1}$ for the time between event $n-1$ and event $n$.

# Renewal processes

Of course, there is nothing specific to light bulbs or renewals in the above.

Consider a sequence of events, with event $n$ at time $\tau_n$.

Write $\tau_0 = 0$, and $\rho_n = \tau_n - \tau_{n-1}$ for the time between event $n-1$ and event $n$.

Then if the $(\rho_n)$ are i.i.d., the $(\tau_n, n = 0, 1, \ldots)$ or the equivalent (event) counting process $(N(t) = \sum_n I(\tau_n \le t), t \ge 0)$ is a renewal process.

# Poisson process

As a special case, if the $(\rho_n)$ are i.i.d. exponentially distributed with rate parameter $\lambda$, $(\tau_n)$ or $(N(t))$ is a *Poisson process* with rate parameter $\lambda$.

# Poisson process

As a special case, if the $(\rho_n)$ are i.i.d. exponentially distributed with rate parameter $\lambda$, $(\tau_n)$ or $(N(t))$ is a *Poisson process* with rate parameter $\lambda$.

But why is it called a “Poisson” process?

# Poisson process

As a special case, if the $(\rho_n)$ are i.i.d. exponentially distributed with rate parameter $\lambda$, $(\tau_n)$ or $(N(t))$ is a *Poisson process* with rate parameter $\lambda$.

But why is it called a “Poisson” process?

One can show: if $\lambda$ is the rate parameter of the $\rho_n$, then

- The number of events/points in a set is Poisson with rate equal to $\lambda$ times the size of the set.

# Poisson process

As a special case, if the $(\rho_n)$ are i.i.d. exponentially distributed with rate parameter $\lambda$, $(\tau_n)$ or $(N(t))$ is a *Poisson process* with rate parameter $\lambda$.

But why is it called a “Poisson” process?

One can show: if $\lambda$ is the rate parameter of the $\rho_n$, then

- The number of events/points in a set is Poisson with rate equal to $\lambda$ times the size of the set.
- The numbers of events/points in non-overlapping sets are independent of each other.

For starters, read “set” as “interval”.

# Poisson process

As a special case, if the $(\rho_n)$ are i.i.d. exponentially distributed with rate parameter $\lambda$, $(\tau_n)$ or $(N(t))$ is a *Poisson process* with rate parameter $\lambda$.

But why is it called a “Poisson” process?

One can show: if $\lambda$ is the rate parameter of the $\rho_n$, then

- The number of events/points in a set is Poisson with rate equal to $\lambda$ times the size of the set.
- The numbers of events/points in non-overlapping sets are independent of each other.

For starters, read “set” as “interval”.

In particular,

$$N(t) \sim \text{Poisson}(\lambda t).$$

# Simulating a Poisson process A

Suppose we want to simulate the (times of the) first $n$ events of a Poisson process with rate $\lambda$.

# Simulating a Poisson process A

Suppose we want to simulate the (times of the) first $n$ events of a Poisson process with rate $\lambda$.

This is easy: we need to simulate i.i.d. $\rho_i \sim \mathrm{exponential}(\lambda)$, and then compute the cumulative sums of these:

```
R> rppA <- function(n, lambda) {
+     cumsum(rexp(n, lambda))
+ }
R> (x <- rppA(20, 1.5))
```

```
 [1]  0.267819  1.368932  1.630125  1.957936  2.382470  2.546440
 [7]  3.864321  4.121348  5.719247  5.822406  6.742708  6.866921
[13]  6.978267  7.298553  7.677173  8.026610  9.159534  9.693273
[19] 11.488227 11.865750
```

# Simulating a Poisson process B

Suppose we want to simulate the events of a Poisson process with rate $\lambda$ up to time $t$.

# Simulating a Poisson process B

Suppose we want to simulate the events of a Poisson process with rate $\lambda$ up to time $t$.

This is not so straightforward, as we don't know $n$.

# Simulating a Poisson process B

Suppose we want to simulate the events of a Poisson process with rate $\lambda$ up to time $t$.

This is not so straightforward, as we don't know $n$.

We could try to iteratively draw a single $\rho_n$ until $\tau_n > t$, but that's perhaps not very efficient. Is there a better way?

# Simulating a Poisson process B

Suppose we want to simulate the events of a Poisson process with rate $\lambda$ up to time $t$.

This is not so straightforward, as we don't know $n$.

We could try to iteratively draw a single $\rho_n$ until $\tau_n > t$, but that's perhaps not very efficient. Is there a better way?

Well, we already know that $N(t) \sim \text{Poisson}(\lambda t)$.

One can show: given $N(t) = n$, the points $\tau_1, \ldots, \tau_n$ are distributed as an ordered sample of size $n$ from the uniform distribution on $[0, t]$!

# Simulating a Poisson process B

Thus, we can do:

```
R> rppB <- function(t, lambda) {
+     sort(runif(rpois(1, t * lambda), max = t))
+ }
```

E.g.,

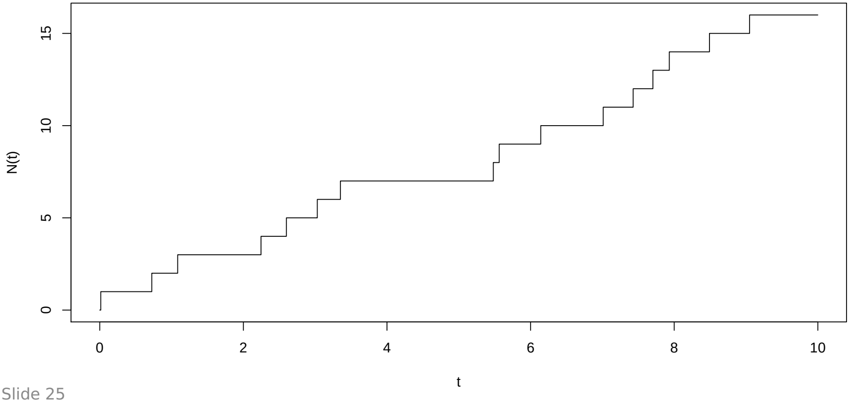
```
R> (x <- rppB(10, 1.5))
```

```
 [1] 0.01409716 0.72473996 1.08543110 2.24573281 2.59902724 3.02953019
 [7] 3.35133575 5.48057882 5.56310538 6.14135012 7.01132067 7.42835590
[13] 7.70294162 7.93157549 8.49079620 9.04964288
```

# Simulating a Poisson process B

To visualize, simple variant:

```
R> plot(c(0, x, 10), c(0, seq_along(x), length(x)),
+       type = "s", xlab = "t", ylab = "N(t)")
```



# Outline

- Sums and mixtures
- Stochastic processes
- Monte Carlo estimation

# Basic idea

Suppose we want to (approximately) compute an integral of the form

$$\theta = \int g(x) f(x)\, dx$$

where $f$ is a density function.

Clearly, if $X \sim f$,

$$\mathbb{E}(g(X)) = \int g(x) f(x)\, dx = \theta$$

and we know from the Law of Large numbers that if $n$ is large and $X_1, \ldots, X_n$ are drawn i.i.d. from $f$,

$$\frac{1}{n} \sum_{i=1}^{n} g(X_i) \approx \mathbb{E}(g(X)) = \theta.$$

# MC estimation

This suggests we could estimate $\theta$ via the following Monte Carlo method:

- Draw an i.i.d. sample $x_1, \ldots, x_n$ from $f$.

# MC estimation

This suggests we could estimate $\theta$ via the following Monte Carlo method:

- Draw an i.i.d. sample $x_1, \ldots, x_n$ from $f$.
- Approximate $\theta$ by

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} g(x_i).$$

# MC estimation

How well can this work?

# MC estimation

How well can this work?

If $X_1, \ldots, X_n$ are i.i.d. from $f$,

$$\mathbb{E}(\hat{\theta}) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} g(X_i)\right) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(g(X_i)) = \theta$$

and (using independence!),

$$\operatorname{var}(\hat{\theta}) = \operatorname{var}\left(\frac{1}{n}\sum_{i=1}^{n} g(X_i)\right) = \frac{1}{n^2}\sum_{i=1}^{n} \operatorname{var}(g(X_i)) = \frac{\operatorname{var}(g(X))}{n}.$$

# MC estimation

So the *standard error* of the MC estimate is

$$\mathrm{sd}(\hat{\theta}) = \frac{\mathrm{sd}(g(X))}{\sqrt{n}}.$$

# MC estimation

So the *standard error* of the MC estimate is

$$\mathrm{sd}(\hat{\theta}) = \frac{\mathrm{sd}(g(X))}{\sqrt{n}}.$$

Qualitatively, this scales like $1/\sqrt{n}$.

# MC estimation

So the *standard error* of the MC estimate is

$$\mathrm{sd}(\hat{\theta}) = \frac{\mathrm{sd}(g(X))}{\sqrt{n}}.$$

Qualitatively, this scales like $1/\sqrt{n}$.

So to improve the precision by a factor of 10, we need to increase the sample size by a factor of 100!

# MC estimation

So the *standard error* of the MC estimate is

$$\mathrm{sd}(\hat{\theta}) = \frac{\mathrm{sd}(g(X))}{\sqrt{n}}.$$

Qualitatively, this scales like $1/\sqrt{n}$.

So to improve the precision by a factor of 10, we need to increase the sample size by a factor of 100!

Of course, we do not know $\mathrm{sd}(g(X))$, but we can estimate it from the MC sample (using the sample sd). This gives the estimated precision

$$\widehat{\mathrm{sd}}(\hat{\theta}) = \frac{\mathrm{sd}(g(x_1), \ldots, g(x_n))}{\sqrt{n}}.$$

# Example 1

Compute (approximate)

$$\theta = \int_0^1 e^{-x}\,dx$$

via MC estimation. (Of course, we know that $\theta = 1 - e^{-1}$.)

# Example 1

Compute (approximate)

$$\theta = \int_0^1 e^{-x}\,dx$$

via MC estimation. (Of course, we know that $\theta = 1 - e^{-1}$.)

For MC estimation, we need to write

$$\theta = \int_0^1 e^{-x}\,dx = \int g(x) f(x)\,dx$$

for some density $f$.

# Example 1

Compute (approximate)

$$\theta = \int_0^1 e^{-x}\, dx$$

via MC estimation. (Of course, we know that $\theta = 1 - e^{-1}$.)

For MC estimation, we need to write

$$\theta = \int_0^1 e^{-x}\, dx = \int g(x) f(x)\, dx$$

for some density $f$.

If we take $g(x) = e^{-x}$, we get $f(x) = 1$ for $0 < x < 1$. I.e., we need to sample from the standard uniform!

# Example 1

We can thus do

```
R> n <- 1234
R> x <- runif(n)
R> theta_hat <- mean(exp(-x))
R> c(theta_hat, 1 - exp(-1))
```

```
[1] 0.6306469 0.6321206
```

and estimate the standard error as

```
R> sd(exp(-x)) / sqrt(n)
```

```
[1] 0.005215123
```

# Example 1

Was this the only possible way to get MC estimates for $\theta$?

# Example 1

Was this the only possible way to get MC estimates for $\theta$?

We could also write

$$\theta = \int_0^1 e^{-x}\, dx = \int I_{(0,1)}(x)\, e^{-x}\, dx.$$

Now $g(x) = I_{(0,1)}(x)$ and $f$ is the density of the standard exponential distribution!

# Example 1

Was this the only possible way to get MC estimates for $\theta$?

We could also write

$$\theta = \int_0^1 e^{-x}\,dx = \int I_{(0,1)}(x)\,e^{-x}\,dx.$$

Now $g(x) = I_{(0,1)}(x)$ and $f$ is the density of the standard exponential distribution!

Which approach do you expect to work better?

# Example 1

E.g.,

```
R> x <- rexp(n)
R> theta_hat <- mean(x <= 1)
R> c(theta_hat, 1 - exp(-1))
```

```
[1] 0.6345219 0.6321206
```

with an estimated standard error of

```
R> sd(x <= 1) / sqrt(n)
```

```
[1] 0.01371426
```

# Example 1

E.g.,

```
R> x <- rexp(n)
R> theta_hat <- mean(x <= 1)
R> c(theta_hat, 1 - exp(-1))
```

```
[1] 0.6345219 0.6321206
```

with an estimated standard error of

```
R> sd(x <= 1) / sqrt(n)
```

```
[1] 0.01371426
```

This seems to be much worse!

# Example 1

What does this example show?

# Example 1

What does this example show?

- For a given $\theta$, there may be several ways to write it as $\theta = \mathbb{E}(g(X))$, $X \sim f$.

# Example 1

What does this example show?

- For a given $\theta$, there may be several ways to write it as $\theta = \mathbb{E}(g(X))$, $X \sim f$.
- Better MC estimates have smaller standard errors $\mathrm{sd}(g(X))$, $X \sim f$.

# Importance sampling

Equivalently, suppose we want to compute/approximate

$$\theta = \int g(x)\,dx.$$

For arbitrary $f$, we can do

$$\theta = \int \frac{g(x)}{f(x)} f(x)\,dx$$

and hence estimate $\theta$ via

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} \frac{g(x_i)}{f(x_i)}, \qquad x_1, \dots, x_n \text{ i.i.d.} \sim f.$$

# Importance sampling

By suitably choosing $f$, we can reduce the variance/sd of the MC estimate: *importance sampling*.

# Importance sampling

By suitably choosing $f$, we can reduce the variance/sd of the MC estimate: *importance sampling*.

What is the best $f$ we can find?

# Importance sampling

By suitably choosing $f$, we can reduce the variance/sd of the MC estimate: *importance sampling*.

What is the best $f$ we can find?

Need to choose $f$ to minimize

$$\begin{aligned}
\operatorname{var}\left(\frac{g(X)}{f(X)}\right) &= \int \left(\frac{g(x)}{f(x)} - \theta\right)^2 f(x)\,dx \\
&= \int \left(\frac{g(x)^2}{f(x)} - 2\theta g(x) + \theta^2 f(x)\right) dx \\
&= \int \frac{g(x)^2}{f(x)}\,dx - \theta^2.
\end{aligned}$$

# Importance sampling

Write

$$I = \int |g(x)|\,dx, \qquad g_0(x) = |g(x)|/I.$$

Then $\theta_0 = \int g_0(x)\,dx = 1$,

$$\int \frac{g(x)^2}{f(x)}\,dx = \int \frac{(g_0(x)\cdot I)^2}{f(x)}\,dx = I^2 \int \frac{g_0(x)^2}{f(x)}\,dx.$$

and from the above computation with $g_0$ instead of $g$,

$$\int \frac{g_0(x)^2}{f(x)}\,dx = 1 + \int \left(\frac{g_0(x)}{f(x)} - 1\right)^2 f(x)\,dx.$$

# Importance sampling

Clearly,

$$\int \left( \frac{g_0(x)}{f(x)} - 1 \right)^2 f(x)\, dx$$

is minimal for $f = g_0$, and hence the same is true for

$$\int \frac{g(x)^2}{f(x)}\, dx.$$

# Importance sampling

We have thus proved: optimal variance reduction in importance sample is achieved for

$$f(x) = \frac{|g(x)|}{\int |g(x)|\, dx},$$

i.e., by sampling proportionally to $|g(x)|$.

# Importance sampling

We have thus proved: optimal variance reduction in importance sample is achieved for

$$f(x) = \frac{|g(x)|}{\int |g(x)|\, dx},$$

i.e., by sampling proportionally to $|g(x)|$.

(Provided of course that $\int |g(x)|\, dx < \infty$.)

# Example 2

For $x > 0$, compute (approximate)

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt = \frac{\theta}{\sqrt{2\pi}} + \frac{1}{2}, \qquad \theta = \int_{0}^{x} e^{-t^2/2} dt$$

via MC estimation.

# Example 2

For $x > 0$, compute (approximate)

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt = \frac{\theta}{\sqrt{2\pi}} + \frac{1}{2}, \qquad \theta = \int_0^x e^{-t^2/2} dt$$

via MC estimation.

The obvious idea is using $g(t) = xe^{-t^2/2}$ and $f$ uniform on $(0, x)$.

# Example 2

For $x > 0$, compute (approximate)

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt = \frac{\theta}{\sqrt{2\pi}} + \frac{1}{2}, \qquad \theta = \int_0^x e^{-t^2/2} dt$$

via MC estimation.

The obvious idea is using $g(t) = xe^{-t^2/2}$ and $f$ uniform on $(0, x)$.

Alternatively, substituting $y = t/x$,

$$\theta = \int_0^x e^{-t^2/2} dt = \int_0^1 xe^{-(xy)^2/2} dy.$$

So we could also draw from the standard uniform, and use the same sample to do MC estimation for several $x$!

# Example 2

A possible implementation of this idea:

```
R> mypnorm <- function(x, n = 10000) {
+     u <- runif(n)
+     p <- numeric(length(x))
+     for(i in seq_along(x)) {
+         g <- x[i] * exp(-(u * x[i])^2 / 2)
+         p[i] <- mean(g) / sqrt(2 * pi) + 0.5
+     }
+     p
+ }
```

# Example 2

This gives e.g.

```
R> x <- seq(.1, 2.5, length.out = 10)
R> (p <- mypnorm(x))

 [1] 0.5398272 0.6430338 0.7365815 0.8155038 0.8774419 0.9226071
 [7] 0.9531367 0.9721733 0.9830047 0.9884799

R> p - pnorm(x)

 [1] -6.672781e-07 -3.231938e-05 -1.605042e-04 -4.360894e-04
 [5] -8.856296e-04 -1.511640e-03 -2.297791e-03 -3.215848e-03
 [9] -4.231838e-03 -5.310464e-03
```

# Antithetic variables

As everyone knows,

$$\mathrm{var}(X + Y) = \mathrm{var}(X) + \mathrm{var}(Y) + 2\mathrm{cov}(X, Y)$$

So maybe we could reduce the variance of MC estimates even more if we used pairs of negatively correlated random variables from the same distribution?

# Antithetic variables

As everyone knows,

$$\mathrm{var}(X+Y) = \mathrm{var}(X) + \mathrm{var}(Y) + 2\mathrm{cov}(X,Y)$$

So maybe we could reduce the variance of MC estimates even more if we used pairs of negatively correlated random variables from the same distribution?

Well, yes, but how could we get such pairs?

# Antithetic variables

As everyone knows,

$$\mathrm{var}(X+Y)=\mathrm{var}(X)+\mathrm{var}(Y)+2\mathrm{cov}(X,Y)$$

So maybe we could reduce the variance of MC estimates even more if we used pairs of negatively correlated random variables from the same distribution?

Well, yes, but how could we get such pairs?

Remember the quantile transform method $X_i = Q(U_i)$. We could also use $Y_i = Q(1-U_i)$, and $U_i$ and $1-U_i$ are negatively correlated.

Maybe the same is true for $X_i$ and $Y_i$?

# Antithetic variables

One can show: if $g = g(x_1, \ldots, x_k)$ is monotone, then

$$g(Q(U_1), \ldots, Q(U_k)), \qquad g(Q(1-U_1), \ldots, Q(1-U_k))$$

are negatively correlated.

# Antithetic variables

One can show: if $g = g(x_1, \ldots, x_k)$ is monotone, then

$$g(Q(U_1), \ldots, Q(U_k)), \qquad g(Q(1-U_1), \ldots, Q(1-U_k))$$

are negatively correlated.

For MC estimation: if we can draw from $f$ via its quantile function, generate $n/2$ replicates each $X_i$ and $Y_i$ using the same $U_{i1}, \ldots, U_{ik}$, and use

$$\hat{\theta} = \frac{2}{n} \sum_{i=1}^{n/2} \frac{X_i + Y_i}{2}.$$

# Antithetic variables

One can show: if $g = g(x_1, \ldots, x_k)$ is monotone, then

$$g(Q(U_1), \ldots, Q(U_k)), \qquad g(Q(1-U_1), \ldots, Q(1-U_k))$$

are negatively correlated.

For MC estimation: if we can draw from $f$ via its quantile function, generate $n/2$ replicates each $X_i$ and $Y_i$ using the same $U_{i1}, \ldots, U_{ik}$, and use

$$\hat{\theta} = \frac{2}{n} \sum_{i=1}^{n/2} \frac{X_i + Y_i}{2}.$$

This requires $nk/2$ instead of $nk$ uniform variates, and reduces estimation variance by using antithetic variables.

# Antithetic variables

To illustrate, continue MC estimation of $\Phi(x)$.

We had

$$\theta = \mathbb{E}_U(xe^{-(xU)^2/2}),$$

for $U$ standard uniform.

When restricting to $x > 0$, $g(u) = xe^{-(ux)^2/2}$ is monotone.

Hence, we can use

$$X_i = xe^{-(xU_i)^2/2}, \qquad Y_i = xe^{-(x(1-U_i))^2/2}.$$

# Antithetic variables

We provide a function which has a flag for toggling the use of anithetic sampling

```
R> mypnorm2 <- function(x, n = 10000, antithetic = TRUE) {
+     u <- runif(n / 2)
+     v <- if(!antithetic) runif(n / 2) else 1 - u
+     u <- c(u, v)
+     p <- numeric(length(x))
+     for(i in seq_along(x)) {
+         g <- x[i] * exp(-(u * x[i])^2 / 2)
+         p[i] <- mean(g) / sqrt(2 * pi) + 0.5
+     }
+     p
+ }
```

and perform the following MC experiment:

# Antithetic variables

```
R> x <- seq(.1, 2.5, length.out = 10)
R> Phi <- pnorm(x)
R> set.seed(123)
R> system.time(p1 <- mypnorm2(x, antithetic = FALSE))

   user  system elapsed
  0.013   0.000   0.013

R> set.seed(123)
R> system.time(p2 <- mypnorm2(x))

   user  system elapsed
  0.002   0.000   0.002
```

So clearly, the antithetic variant is *faster*. It is also *more precise*:

# Antithetic variables

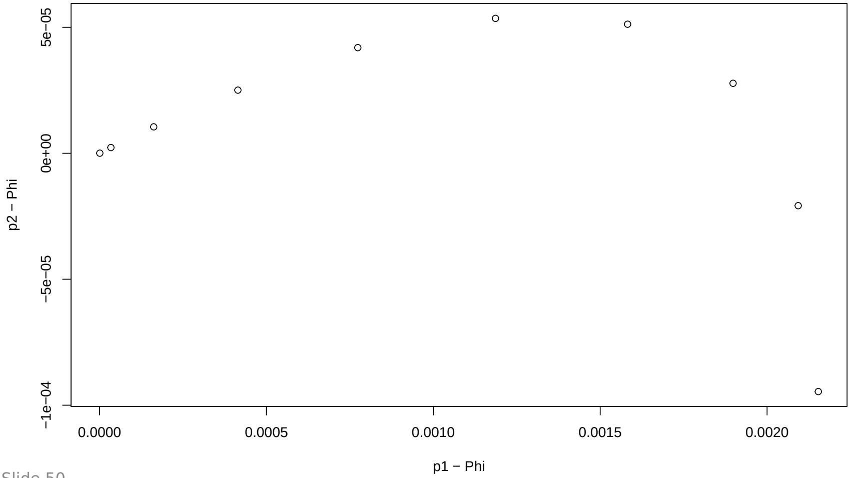
```
R> print(round(cbind(x, Delta1 = p1 - Phi, Delta2 = p2 - Phi), 5))
```

```
            x  Delta1 Delta2
 [1,] 0.10000 0.00000  0e+00
 [2,] 0.36667 0.00003  0e+00
 [3,] 0.63333 0.00016  1e-05
 [4,] 0.90000 0.00041  3e-05
 [5,] 1.16667 0.00077  4e-05
 [6,] 1.43333 0.00119  5e-05
 [7,] 1.70000 0.00158  5e-05
 [8,] 1.96667 0.00190  3e-05
 [9,] 2.23333 0.00209 -2e-05
[10,] 2.50000 0.00215 -9e-05
```

Graphically:

# Antithetic variables

```
R> plot(p1 - Phi, p2 - Phi)
```


5e−05
0e+00
−5e−05
−1e−04
p2 − Phi
p1 − Phi