The Need for pRNG Streams

Recall the *Simple Inventory System* where demands for inventory items slowly depletes an initial maximum store of S. Periodically, inventory size l is evaluated and if $l < s$ a restocking order is placed for $S - l$ items.

Demands placed when $l < 0$ incur a penalty for the seller, so keeping $l > 0$ should be correlated with increased profits. But each order has an overhead cost diminishing profit — the optimal re-ordering policy (s) is not obvious in this case.

Additionally, we should consider non-instaneous delivery of ordered items, aka *delivery lag*.

We want to assess the difference in two different re-ordering policies s_A and s_B

A Naive Approch to s_A vs s_B

The `SIM` will use a sequence of random values in two different **stochastic processes** in the simulation: **demands** and **delivery lag**.

```
$ cat SIM-AvsB.sh
set -e
for seed in $(cat SEEDS) ; do
    # collect policy specific results for a given random seed
    ./SIM A ${seed} >A-output-${seed}.out
    ./SIM B ${seed} >B-output-${seed}.out
done
$ ./ANALYZE.sh A-output-*.out -- B-output-*.out
n=500 mean=0.266 popstddev=1.530 ci90=(0.230,0.302)
```

Clearly, we detected a difference — but is this accurate? Does it really exist? Is the claimed confidence interval correct?

Discuss. . .

A Naive Approch to s_A vs s_B

That was a trick question — the answer to both is **YES**. It's a statistical test of independent samples, the only "claim" the confidence interval asserts is that

> *If we ran the experiment,* ***with a different set of seeds*** *we expect the mean difference between the two scenarios to fall within* $(0.230, 0.320)$ *about 90% of the time.*

The **real** question is **qualatitive**:

> WAS THE *experimental design* OPTIMIZED TO MEASURE THE **MOST ACCURATE** DIFFERENCE IN POLICIES s_A AND s_B?

A Naive Approch to s_A vs s_B

No.

The re-order polices will move around the (simultation time) of reorders. The demand seen by s_A and s_B will be the same **up until one of the policies places an order**. The policy that orders first (suppose it is s_A) will use a draw from the pRNG (call it x_k) to determine delivery lag.

But x_k will produce a demand for s_B. And some time after this, s_B will use an x_m, $m > k$ to determine a delivery lag — but most likely x_m will generate a demand arrival for s_A.

A Naive Approch to s_A vs s_B

As the two policies **diverge** from their initial syncronized use of the pRNG sequence $x_0, x_1, \ldots$, they will see different patterns in both **demands** and **delivery lags**.

Our naive comparison of the two policies **is not flawed** — but we can perform a more sensitive experiment using **independent sequences of random values** for our two stochastic processes in the simulation: **demands**, and **delivery lags**.
In this way, the same demand profile is seen by both s_A and s_B experiments, and both experiments see the same sequence of random delivery lags.

This is an example of ***Variance Reduction*** in experimental design.

Problematic Solutions

i. Use different seeds for each stochastic element.

Problematic Solutions

i. Use different seeds for each stochastic element.
   You may inadvertently choose two seeds that are one after another in the pRNG. **Really, really, bad:** you **will induce serial dependence** between the stochastic elements of your simulation that are supposed to be independent!
ii. Use a different pRNG for each stocastic element, eg: choose different (a, m) pairs for several distinct Lehmer generators.

Problematic Solutions

i. Use different seeds for each stochastic element.
   You may inadvertently choose two seeds that are one after another in the pRNG. **Really, really, bad:** you **will induce serial dependence** between the stochastic elements of your simulation that are supposed to be independent!
ii. Use a different pRNG for each stocastic element, eg: choose different (a, m) pairs for several distinct Lehmer generators.
   Not much better, the quality of pRNGs is an assessment of their apparent randomness *internally* within their own generated values (for a particular seed). There have been **no studies** showing that two different pRNGs are random and indpendent of each other.

   Why not? Because this is universally considered the wrong use of pRNGs in simulations.

Do the Right Thing: pRNG (sub) Streams

We can mathematically break of up the **whole sequence** of pRNG generated values into disjoint subsequences.

These are called ***streams*** of the pRNG.

How? Suppose you want to break up an (a, m) Lehmer generator into 5 distinct streams (indexed by j):

$$\text{elements per streams} = e = \left\lfloor \frac{m}{5} \right\rfloor$$

Do the Right Thing: pRNG (sub) Streams

Given a seed x_0, the $j = 0$ stream begins at $x_0 = x_{0,0}$. The $j = 1$ stream begins at

$$x_{1,0} = a^e x_0 \bmod m$$

the $j = 2$ stream begins at

$$x_{2,0} = a^{2e} x_0 \bmod m$$

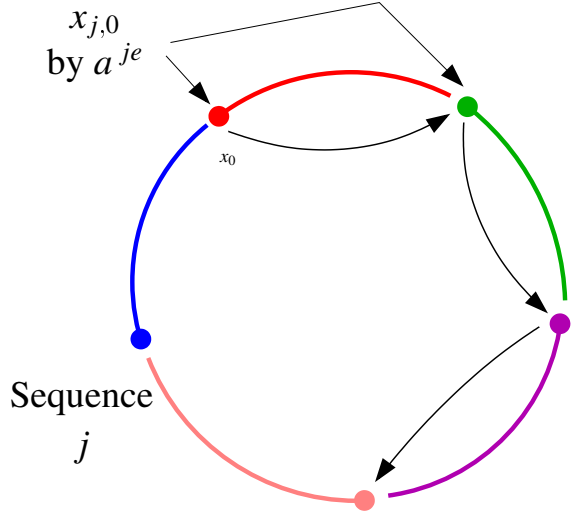
and the jth stream begins at

$$x_{j,0} = a^{je} x_0 \bmod m$$

The "next value" function remains the same:

$$f(x_{j,i}) = a x_{j,i-1} \bmod m$$

You just have j different states to store in memory.



A Better Approch to s_A vs s_B

```
$ cat SIM-AvsB-streams.sh
set -e
for seed in $(cat SEEDS) ; do
    # collect policy specific results for a given random seed
    # Use 2 streams, stream 0 for demands (-d), 1 for lags (-l)
    ./SIM A ${seed} --streams 2 -d 0 -l 1 >A-output-${seed}.out
    ./SIM B ${seed} --streams 2 -d 0 -l 1 >B-output-${seed}.out
done
$ ./ANALYZE.sh A-output-*.out -- B-output-*.out
n=500 mean=0.315 popstddev=0.352 ci90=(0.307,0.324)
```

Note the substantially better confidence interval due to ***variance reduction***

— improved by a factor of 18.

A Paired Approch to s_A vs s_B (Ideal)

```
$ cat SIM-AvsB-streams-paired.sh
set -e
for seed in $(cat SEEDS) ; do
    # collect policy specific results for a given random seed
    # Use 2 streams, stream 0 for demands (-d), 1 for lags (-l)
    ./SIM A ${seed} --streams 2 -d 0 -l 1 >A-output-${seed}.out
    ./SIM B ${seed} --streams 2 -d 0 -l 1 >B-output-${seed}.out
done
# a little more effort is required in ANALYZE-PAIRED.sh to pair
# same seed results together correctly
$ ./ANALYZE-PAIRED.sh "%s-output-%d.out" A B SEEDS
n=500 mean=0.353 popstddev=0.341 ci90=(0.328,0.378)
```

We treat the s_A vs s_B as a **placebo/treatment** or **pre/post** experimental design — the difference between same seed results are treated as the original data set δ_i.

By treating differences as accurately as possible in our statistical tests, we can again improve the result.

A Paired Approch to s_A vs s_B (Ideal)

```
$ cat SIM-AvsB-streams-paired.sh
set -e
for seed in $(cat SEEDS) ; do
    # collect policy specific results for a given random seed
    # Use 2 streams, stream 0 for demands (-d), 1 for lags (-l)
    ./SIM A ${seed} --streams 2 -d 0 -l 1 >A-output-${seed}.out
    ./SIM B ${seed} --streams 2 -d 0 -l 1 >B-output-${seed}.out
done
# a little more effort is required in ANALYZE-PAIRED.sh to pair
# same seed results together correctly
$ ./ANALYZE-PAIRED.sh "%s-output-%d.out" A B SEEDS
n=500 mean=0.353 popstddev=0.341 ci90=(0.328,0.378)
```

Using a paired experimental design **isn't usually considered an example of *variance reduction*** — it's simply best practice in statistical analysis.

You aren't **always guaranteed** of a better result vs. a non-paired analysis — but you do have the most accurate result!