

# Tücken bei der Erstellung von Zufallszahlen mit „RANUNI“

Rainer Muche, Tillmann Babik  
Institut für Biometrie  
Schwabstraße 13  
89075 Ulm  
rainer.muche@uni-ulm.de

## Zusammenfassung

In der Statistik werden oft Zufallszahlen benötigt, um z.B. Simulationen oder auch Randomisierungen in Klinischen Studien durchzuführen. Dabei wird Wert auf die Programmierung des jeweiligen Algorithmus gelegt, den zugrunde liegenden Zufallszahlen wird aber in der Regel vertraut. Speziell bei der Randomisierung wird in den gültigen Regelwerken festgelegt, dass die zufällige Zuteilung der Patienten nachprüfbar und wiederholbar sein muss, was für die Ziehung der Zufallszahlen und den Ablauf des ganzen Randomisationsprozesses eine wichtige und notwendige Einschränkung darstellt.

In SAS werden gleichverteilte Zufallszahlen mit der Funktion RANUNI zur Verfügung gestellt. In diesem Beitrag soll auf die Tücken bei der Verwendung von RANUNI, speziell der Möglichkeit der Verwendung eines Seed-Startwertes, eingegangen werden. Dabei gibt es Probleme beim gleichzeitigen Aufruf von mehreren RANUNI-Funktionen in einem Data-Step sowie innerhalb von Makros.

Es werden verschiedene Verwendungen von RANUNI in SAS getestet, um die auftretenden Problematiken aufzuzeigen. Es wird mit der Funktion und der Call Routine gearbeitet, welche jeweils in einem SAS Data Step und einem SAS Makro angewendet und evaluiert werden. Es kann gezeigt werden, dass die Verwendung der Funktion unter bestimmten Voraussetzungen zu unerwünschten Ergebnissen führen kann und dass die Call Routine eine bessere Alternative ist. Speziell, wenn mehrere Zufallsprozesse in einem DataStep und/oder Makro ablaufen sollen, ist die Nutzung der Call Routine dringend angezeigt.

**Schlüsselwörter:** RANUNI, Funktion RANUNI, Call Routine, Makro

## 1 Einleitung

In der Statistik werden oft Zufallszahlen benötigt, um z.B. Simulationen oder auch Randomisierungen in Klinischen Studien durchzuführen. Dabei wird Wert auf die Programmierung des jeweiligen Algorithmus gelegt, den zugrunde liegenden Zufallszahlen wird aber in der Regel vertraut. Speziell bei der Randomisierung wird in den gültigen Regelwerken (ICH-E9 [1]) zum Beispiel festgelegt, dass die zufällige Zuteilung der Patienten nachprüfbar und wiederholbar sein muss, was für die Ziehung der Zufallszahlen und den Ablauf des ganzen Randomisationsprozesses eine wichtige und notwendige Einschränkung darstellt. Dafür wird typischerweise ein Seed als Startwert für die Auswahl der Zufallszahl genutzt, mit dem der Zufallsprozess wiederholbar ist.

In SAS werden gleichverteilte Zufallszahlen mit der Funktion "RANUNI" zur Verfügung gestellt. In diesem Beitrag soll auf die Tücken bei der Verwendung von "RANUNI", speziell der Möglichkeit der Verwendung eines Seed-Startwertes, eingegangen werden. Dabei gibt es Probleme beim gleichzeitigen Aufruf von mehreren „RANUNI“-Funktionen in einem Data-Step sowie innerhalb von Makros. Auf diese wird näher eingegangen werden.

## 2 Unterschied RANUNI als Funktion und RANUNI als Call Routine laut SAS OnlineDoc V9

Laut SAS OnlineDoc [2] erlaubt die Call Routine eine größere Kontrolle über den Seedwert und damit über die Erstellung von Zufallsvariablen. Zudem liefert die Call Routine nach jedem Ausführen einen neuen Seedwert zurück. Die SAS OnlineDoc bezieht sich dabei aber ausschließlich auf die Verwendung innerhalb eines DataSteps.

In der OnlineDoc gibt es unter der folgenden URL dazu auch ein SAS Beispiel:

<http://support.sas.com/onlinedoc/913/getDoc/de/lrdict.hlp/a000127852.htm#a000127854>.

```
options nodate pageno=1 linesize=80 pagesize=60;

data case;
  retain Seed_1 Seed_2 Seed_3 45;
  do i=1 to 10;
    call ranuni(Seed_1,X1);
    call ranuni(Seed_2,X2);
    X3=ranuni(Seed_3);
    if i=5 then
      do;
        Seed_2=18;
        Seed_3=18;
      end;
    output;
  end;
run;

proc print;
  id i;
  var Seed_1-Seed_3 X1-X3;
run;
```

**Tabelle 1:** Das Ergebnis zu obigem SAS Beispiel

i	Seed_1	Seed_2	Seed_3	X1	X2	X3
1	694315054	694315054	45	0.32332	0.32332	0.32332
2	1404437564	1404437564	45	0.65399	0.65399	0.65399
3	2130505156	2130505156	45	0.99209	0.99209	0.99209
4	1445125588	1445125588	45	0.67294	0.67294	0.67294
5	1013861398	18	18	0.47212	0.47212	0.47212
6	1326029789	707222751	18	0.61748	0.32933	0.61748
7	932142747	991271755	18	0.43406	0.46160	0.43406
8	1988843719	422705333	18	0.92613	0.19684	0.92613
9	516966271	1437043694	18	0.24073	0.66918	0.24073
10	2137808851	1264538018	18	0.99549	0.58885	0.99549

X1 und X2 sind Zufallszahlen, die mit der Call Routine erzeugt wurden, X3 wurde mit der Funktion erzeugt. Alle drei Zufallszahlen wurden mit einem Seedwert von "45" initialisiert. In Seed\_1-Seed\_3 werden die Seedwerte für die Zufallszahlen gespeichert. Die Nummerierung der Spaltenüberschriften korrespondiert mit der der Variablen. In Seed\_1 und Seed\_2 sieht man die von der Call Routine zurückgegebenen Seedwerte, welche in den nächsten Schleifendurchläufen als neuer Seed benutzt werden. Die Frage stellt sich, wieso bei der Call Routine und der Funktion dieselbe Zufallszahl ausgegeben wird, obwohl die Seedwerte so unterschiedlich sind. Man kann daher annehmen, dass die Funktion (X3) im Hintergrund die gleiche Seedveränderungen vornimmt, wie die Call Routine, diese aber nicht zurückgibt. Hier kommt auch der Unterschied zum tragen. Verändert man den Seedwert (Durchlauf 5), so zeigt dies auf die Funktion keine Auswirkung, die Funktion läuft immer noch synchron mit X1. Man hat also bei der Funktion nach der einmaligen Initialisierung keine Kontrolle mehr über die Erstellung der Zufallszahlen. Aber die Veränderung des Seedwertes der Call Routine (X2) zeigt Wirkung, die Ergebnisse weichen jetzt von X1 und X3 ab. Soweit zur Erläuterung der Angaben in der SAS OnlineDoc. Nun zu den Problemen.

### 3 Problembeschreibung

Es ist nicht möglich, mehrere unabhängige RANUNI Funktionen in einem DataStep zu nutzen. Die einmalige Initialisierung der ersten RANUNI Funktion hat auch Einfluss auf nachfolgende RANUNI Funktionen. Zum Beispiel ergibt

```
DATA Beispiel;
  Y=RANUNI(45);
  Z=RANUNI(45);
RUN;
```

nicht, dass Y und Z übereinstimmen, obwohl beide Funktionen denselben Seedwert zu scheinen haben. Dem ist nämlich nicht so, stattdessen erhält Y den Wert 0.32332 und Z den Wert 0.65399. Zum Vergleich die ersten beiden Zeilen in Spalte X3 des SAS Beispiels. Was passiert also? RANUNI wird mit 45 initialisiert und man kann annehmen,

dass der Seedwert intern auf 694315054 (siehe Seed\_1 im SAS Beispiel) gesetzt und als Seed in der zweiten RANUNI Funktion benutzt wird. Das bedeutet, für die Erstellung zweier unabhängiger Zufallsvariablen innerhalb eines DataSteps **muss** die Call Routine verwendet werden. Es ist allerdings zu beachten, dass bei der Call Routine keine Konstanten als Seed verwendet werden. Hier müssen **zwingend** Variablen verwendet werden, da SAS diese Werte verändern möchte und dies nur mit Variablen möglich ist. Im nachfolgenden Beispiel stimmen die Zufallszahlen in Y und Z überein:

```
DATA Beispiel;  
  RETAIN SEED_1 SEED_2 45;  
  CALL RANUNI(SEED_1, Y);  
  CALL RANUNI(SEED_2, Z);  
RUN;
```

Die Ergebnisse der Funktion innerhalb eines DataSteps sind reproduzierbar, was bedeutet, dass die Seedwerte der Funktion bei jedem erneuten DataStep Aufruf zurückgesetzt werden.

Hinweis: Die Initialisierung der RANUNI Methoden kann auch mit 0 erfolgen, dadurch wird die aktuelle Systemzeit als Startseed genutzt. Die Ergebnisse werden dadurch "zufälliger", sind aber auch nicht mehr reproduzierbar.

## 4 RANUNI in SAS-Makros

Damit wäre das Thema RANUNI als Funktion und als Call Routine, innerhalb von DataSteps abgehandelt. Nun zu der Thematik wenn man Zufallszahlen innerhalb eines Makros erstellen möchte. Hierzu ebenfalls zwei Tests:

```
Test_01  
RANUNI Funktion in einem SAS Makro:  
%MACRO Test_01();  
  %DO i=1 %TO 10;  
    %PUT %SYSFUNC(RANUNI(45));  
  %END;  
%MEND;  
%Test_01;
```

Beim ersten Durchlauf erhalten wir hier dieselben Zufallszahlen wie in X3 des SAS Beispiels. Aber die Ergebnisse sind innerhalb einer SAS Session nicht reproduzierbar, weil die intern erstellten Seedwerte (siehe Problembeschreibung DataStep) nicht bei jedem Makroaufruf zurückgesetzt werden, sondern erst nach dem Schließen von SAS. Eine Möglichkeit zum vorzeitigen Leeren derartiger "Buffer" ist uns nicht bekannt. Auch wenn man statt dem festen Wert "45" eine Makrovariable einfügt und diese bei jedem Makroaufruf auf "45" setzt, ist das Ergebnis nicht reproduzierbar.

## Test\_02

RANUNI Call Routine in einem SAS Makro:

```
%MACRO Test_02();  
    %LET X=0;  
    %LET FIX=45;  
    %DO i=1 %TO 10;  
        %SYSCALL RANUNI(FIX, X);  
        %PUT &X;  
    %END;  
%MEND;  
%Test_02;
```

Bei der Call Routine muss als Seedwert eine Variable verwendet werden (FIX=45), der Grund wurde bereits oben in der DataStep Abhandlung erklärt. Zudem muss der Wert, in welche die Zufallszahl geschrieben werden soll, deklariert werden (X=0). Da man bei der Call Routine die Möglichkeit hat, aktiv in die Seedvariable einzugreifen, hat man bei diesem Test reproduzierbare Ergebnisse, da bei jedem Aufruf der Startseed auf 45 gesetzt wird. Soll mit der Systemzeit initialisiert werden, dann muss die Variable, hier FIX, auf 0 gesetzt werden: "%LET FIX=0". Die Zuweisung der Seedvariablen muss außerhalb der Schleife geschehen, da ansonsten derselbe Seedwert für jeden Durchlauf genutzt wird, was natürlich gleiche Zufallszahlen zur Folge hat.

## Fazit

Es wurden verschiedene Verwendungen von "RANUNI" in SAS getestet, um die auftretenden Problematiken aufzuzeigen. Es wurde mit der Funktion [3] und der Call Routine [2] gearbeitet, welche jeweils in einem SAS Data Step und einem SAS Makro angewendet und evaluiert wurden. Es kann gezeigt werden, dass die Verwendung der Funktion unter bestimmten Voraussetzungen zu unerwünschten Ergebnissen führen kann und dass die Call Routine in diesen Fällen eine bessere Alternative ist. Speziell, wenn mehrere Zufallsprozesse in einem DataStep und/oder Makro ablaufen sollen, ist die Nutzung der Call Routine dringend angezeigt.

## Literatur

- [1] ICH-E9: Statistical principals for clinical trials; Online unter <http://www.emea.europa.eu/pdfs/human/ich/036396en.pdf>
- [2] SAS OnlineDoc 9.1.3: RANUNI als Call Routine; Online unter <http://support.sas.com/onlinedoc/913/getDoc/de/lrdict.hlp/a000127852.htm>
- [3] SAS OnlineDoc 9.1.3: RANUNI als Funktion; Online unter <http://support.sas.com/onlinedoc/913/getDoc/de/lrdict.hlp/a000202926.htm>