# AGENDA

**Simulationen und Mathematische Programmierung mit SAS–** Dr. Mihai Paunescu

**Die SAS Communities - Hilfe in der Not zu jeder Zeit, wenn SAS nicht tut, was man will –** Kurt Bremser, Data Warehouse Administrator und SAS Communities Super-User, AMOS AUSTRIA

**Die besten Tipps und Tricks aus meinen liebsten SAS Press Büchern – und warum Sie überlegen sollten, selbst ein Buch für SAS zu schreiben–** DI Rainer Sternecker

**„Höher, schneller, weiter“ – SAS Architekturtrends Cloud Computing, Event Streaming, Machine Learning und mehr–** Ing. Phillip Manschek BSc.

# SIMULATIONEN UND MATHEMATISCHE PROGRAMMIERUNG MIT SAS

sas® club

Der Business Analytics Club für SAS User

**GERHARD SVOLBA**
**MIHAI PAUNESCU**

Credits to Rick Wicklin, SAS Cary, NC

# EINLEITUNG

- Grundidee von Simulationen

- 10 Tipps und Tricks für Simulationen mit SAS
  - Simulationsmöglichkeiten in SAS
  - Zufallszahlen, Verteilungen und Analysemöglichkeiten
  - Mathematische Programmierung mit der SAS® IML Software
  - Optimierung Ihrer Simulationen

- Vorgehensweise zur Analyse von Systemen
  - Zu komplex für die formelmäßige oder theoretische Behandlung
  - Simulationsmodell als Basis für (viele) Experimente
  - Gewinnung von Erkenntnissen über das reale System

- Anwendung typischerweise für
  - Analyse von Spiel- und Investitionsstrategien
  - Analytisch unlösbare Probleme
  - Theoretisch lösbare Probleme, die aber einen hohen Komplexitätsgrad aufweisen
  - Nachbildung von komplexen Prozessen

# TIPP #1 VERWENDEN SIE SAS FÜR SIMULATIONS-STUDIEN

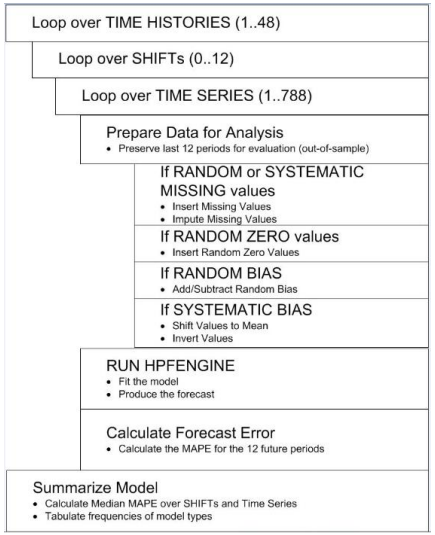
- **SAS Datastep** für die Simulation von Daten aus univariaten und unkorrelierten multivariaten Verteilungen
- **SAS®IML** für die Simulation von Daten aus vielen Verteilungen z.B: korrelierten multivariaten Verteilungen, Definition neuer Funktionen für das Erzeugen von Verteilungen, die in SAS nicht vorhanden sind
- **SAS®STAT und SAS®ETS** Procedures (SIMNORMAL, SIM2D, COPULA) zur Simulation von Daten mit speziellen Eigenschaften.

- Simulationsmöglichkeiten in SAS, die in diesem Vortrag nicht behandelt werden
  - **SAS Simulation Studio** (OR) für die Simulation von diskreten Ereignissen
  - **Proc MCMC** (STAT) Markov-Chain Monte Carlo Procedure zum Schätzen Bayesianischer Modelle
  - **Proc Risk** und **SAS Risk Management** zur Simulation von Risiko Parametern
  - **Proc Model** (ETS) Monte Carlo Simulation von Zeitreihenmodellen

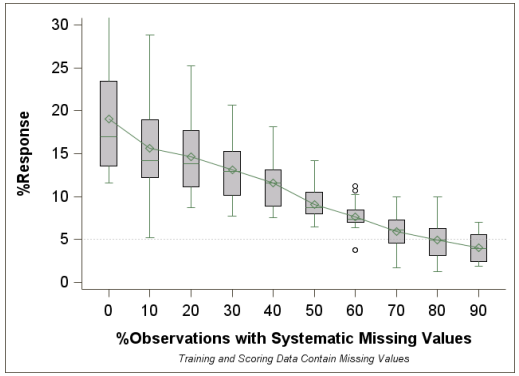
- Auswerte- und Darstellungsmöglichkeiten im SAS®System
  - SAS®Visual Analytics
  - SAS Reporting, SAS Graphiken, Geo-Maps

# TIPP #1

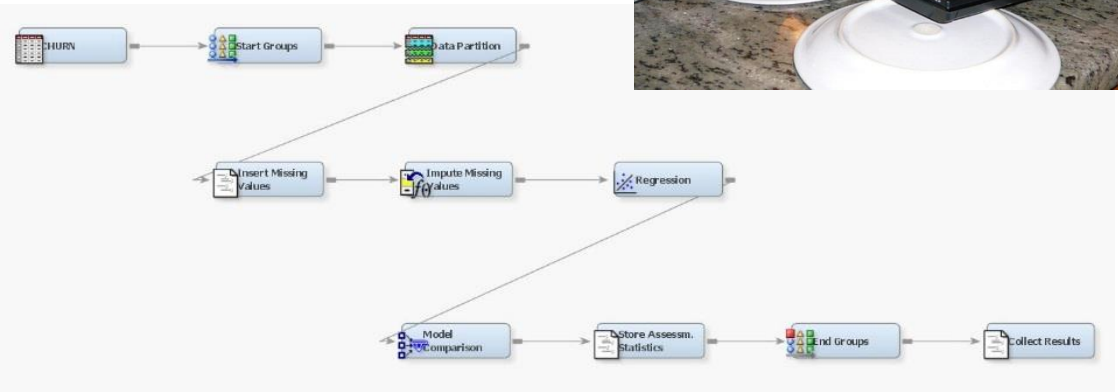
## BEISPIEL: SIMULATION DER KONSEQUENZ SCHLECHTER DATENQUALITÄT AUF DIE MODELLGÜTE


- Verschachtelte Datastep Schleifen
- Datenmanagement Anweisungen
- Analytic Procedures
- Aggregationen, Auswertungen




Kombination unterschiedlicher SAS Tools (SAS®Enterprise Miner, Datastep Code, Macro Code, Auswertungen)



# VERWENDEN SIE DIE „NEUEN“ GENERATOREN FÜR ZUFALLSZAHLEN

## Allgemeines Template für die Simulation univariater Daten in einem SAS Datastep

```
%let N=100;
data Sample;
  call streaminit(12345);
  do i=1 to &n;
   x=rand("DistributionName",parm1,parm2,...);
   output;
  end;
run;
```

Simulationsparameter als Macro Variable(n)

Kein SET-Statement, Daten werden erst erzeugt

Zufallszahlen-Generator zur Erzeugung der Daten

Do-Loop für die Iterationen

# TIPP #2 VERWENDEN SIE DIE „NEUEN“ GENERATOREN FÜR ZUFALLSZAHLEN

- Die „alten“ Zufallszahlengeneratoren in SAS Base (RANUNI, RANNOR, RANPOI, …) verwenden einen älteren Algorithmus aus den 1970ern. (genauso wie PROBxxx, xxxINV)
- Kein Problem bei kleinen Samples (1000, …)
- Der Mersenne-Twister Algorithmus hat den Vorteil einen extrem langen Periode („Wann wiederholt sich die Sequenz“)
- Dieser Algorithmus ist in der RAND Funktion in SAS im Einsatz (seit SAS 9)

# TIPP #2 BEISPIELE FÜR DIE „RAND“ FUNKTION

## SAS Datastep

```
call streaminit(12345);
x1=rand("Bernoulli",0.5);       *** Münzwurf;
x2=rand("Binomial",0.5,10);     *** Anzahl der Erfolge bei 10 Versuchen;
x3=rand("Geometric",0.5);       *** Anzahl der Versuche bis zum Erfolg;
x4=ceil(6*rand("Uniform"));     *** Ergebnisse eines Würfels;
x5=rand("Table",0.5,0.3,0.2); *** Häufigkeiten mit Zurücklegen;
x6=rand("Poisson",4);           *** Anzahl der Ereignisse pro Zeitintervall;
x7=rand("Uniform");             *** Gleichverteilung im Interval [0,1];
x8=rand("Normal",24,6);         *** Normalverteilung mit mu=24 und sigma=6;
```

## SAS IML

```
call randseed(7654);
call randgen(x_bern,"Bernoulli",0.5);
call randgen(x_binom,"Bimonial",0,5,10);
Table_prob={0.5 0.3 0.2};
call randgen(x_table,"Table",table_prob);
```

# TIPP #3

## SIMULIERE DATEN AUS EINER KOMBINATION VON VERTEILUNGEN

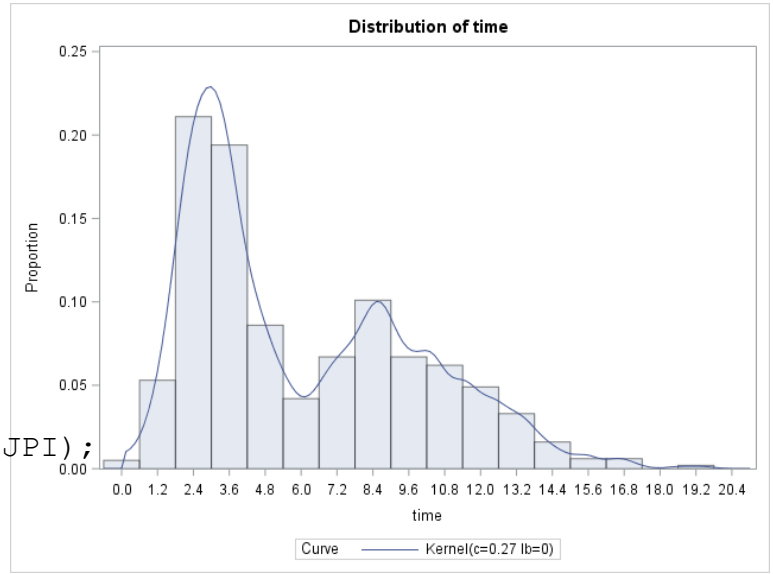
In einem Call Center werden die Anrufe in 3 Gruppen geteilt: 50% sind einfache Anfragen, 20% sind spezialisierte Anfragen, und 30 % sind „harte Fälle“. Die Erfahrungswerte bzgl. der Bearbeitungsdauer sind in folgender Tabelle dargestellt.

| Question | Mean | Standard Deviation |
|---|---|---|
| Easy | 3 | 1 |
| Specialized | 8 | 2 |
| Hard | 10 | 3 |

```
data Calls(drop=i);
call streaminit(0);
array prob [3] _temporary_ (0.5 0.2 0.3); /* mixing probabilities */

do i = 1 to 1000;
  Type = rand("Table", of prob[*]);
   if      Type=1 then time = rand("Normal", 3, 1);
   else if Type=2 then time = rand("Normal", 8, 2);
   else                time = rand("Normal", 10, 3);
   output;
end;
run;

proc univariate data=Calls;
 ods select Histogram;
 histogram time / vscale=proportion kernel(lower=0 c=SJPI);
run;
```



# TIPP #3

## SIMULIERE DATEN AUS EINER KOMBINATION VON VERTEILUNGEN

- Rick Wicklin beschreibt in seinem SGF2015 Paper wie Daten aus komplexen Verteilungen simuliert werden können, auch wenn diese im Basis-Set der 20 Verteilungen für RAND Funktion nicht enthalten sind.
  - Löschen von bestimmten Wertebereichen einer Verteilungen ergibt eine Truncated Distribution
  - Verschiebung und Skalierung von Zufallsvariablen innerhalb der gleichen Verteilungsfamilie
  - Anwendung von Transformation um eine Verteilung in eine andere zu transformieren

# TIPP #4

## MACHEN SIE SICH MIT SAS®IML VERTRAUT

- SAS hat auch eine Matrixsprache (SAS IML Software) → PROC IML
- Diese ist voll in das SAS System integriert
  - Verwenden von SAS Datasets, Ausgeben von Ergebnissen nach SAS
  - SAS Funktionen, SAS Formate, ….
- SAS IML bietet: Matrizen, Matrixmultiplikationen, Vektoren, Skalare, Teilmatrizen, Indizes
- SAS®IML bietet auch eine Integration zwischen SAS und dem R Open Source Project

# TIPP #4

## MACHEN SIE SICH MIT SAS®IML VERTRAUT

| Operator | Description |
|---|---|
| ` (accent grave) | Transpose (postfix) |
| - *(prefix)* | Negative prefix |
| [ ] | Subscript |
| ** | Matrix exponentiation |
| ## | Element-wise exponentiation |
| * | Matrix multiplication |
| # | Element-wise multiplication |
| / | Element-wise division |
| @ | Direct (Kronecker) product |
| + | Addition |
| - | Subtraction |
| \|\| | Horizontal concatenation |
| // | Vertical concatenation |

Beispiele

- A+B : matrix addition
- A*B : matrix multiplication,
- A#B : element-wise multiplication
- A[5,2] : Element aus der 5. Zeile, 2. Spalte
- A[1:3,2:10] : die ersten drei Spalten für die 2. bis 10. Zeile
- W = INV(T(x)*x);

# TIPP #4

## MACHEN SIE SICH MIT SAS®IML VERTRAUT

Projekte mit ihren Gewinn und Auftrittswahrscheinlichkeiten

| Obs | Gewinn | Probability |
|---|---|---|
| 1 | 500 | 0,2 |
| 2 | 100 | 0,4 |
| 3 | 10 | 0,6 |

Alle Gewinn/Verlust Kombinationen mit Gesamt-Gewinn und dazugehörigen Wahrscheinlichkeit

| ID | Gesamt Gewinn | Probability | Profil |
|---|---|---|---|
| 1 | 500 | 0,048 | 001 |
| 2 | 100 | 0,128 | 010 |
| 3 | 600 | 0,032 | 011 |
| 4 | 10 | 0,288 | 100 |
| 5 | 510 | 0,072 | 101 |
| 6 | 110 | 0,192 | 110 |
| 7 | 610 | 0,048 | 111 |
| 8 | 0 | 0,192 | 000 |

# TIPP #4

## MACHEN SIE SICH MIT SAS®IML VERTRAUT


```
proc iml;
   use projects; read all; close;

   N = nrow(prob);
   ScenarioID = t(1:2**N);
   format = "binary" + strip(char(N));
   bin = putn(ScenarioID,format);


   bt=j(2**N,N,99);
   do i = 1 to N;
      bt[,i]=num(substr(bin,N-i+1,1));
   end;



   prob_m  = abs(1-bt-t(prob));

   ScenarioSum = bt * value;
   ScenarioProb   = prob_m[,#];



    create FullCalc_Outcomes_IML var {Scena
   append;
   close FullCalc_Outcomes_IML;
quit;
```

| ScenarioID | bin |
|---|---|
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |
| 8 | 000 |

| Obs | VALUE | PROB |
|---|---|---|
| 1 | 500 | 0,2 |
| 2 | 100 | 0,4 |
| 3 | 10 | 0,6 |

| | | | bt | | |
|---|---|---|---|---|---|
| | | | 1 | 0 | 0 |
| | | | 0 | 1 | 0 |
| | | | 1 | 1 | 0 |
| | | | 0 | 0 | 1 |
| | | | 1 | 0 | 1 |
| | | | 0 | 1 | 1 |
| | | | 1 | 1 | 1 |
| | | | 0 | 0 | 0 |

| | prob_m | | |
|---|---|---|---|
| | 0,2 | 0,6 | 0,4 |
| | 0,8 | 0,4 | 0,4 |
| | 0,2 | 0,4 | 0,4 |
| | 0,8 | 0,6 | 0,6 |
| | 0,2 | 0,6 | 0,6 |
| | 0,8 | 0,4 | 0,6 |
| | 0,2 | 0,4 | 0,6 |
| | 0,8 | 0,6 | 0,4 |

| ScenarioSum | ScenarioProb |
|---|---|
| 500 | 0,048 |
| 100 | 0,128 |
| 600 | 0,032 |
| 10 | 0,288 |
| 510 | 0,072 |
| 110 | 0,192 |
| 610 | 0,048 |
| 0 | 0,192 |

# TIPP #4

# MACHEN SIE SICH MIT SAS®IML VERTRAUT

## Codevergleich: Berechnung und Häufigkeitsgewichtung aller möglichen Ereignis-Kombination einer Projektliste

### SAS Datastep

```
*** Calculate Number of Projects;
proc sql noprint;
 select strip(put(count(*),8.)) into :n_proj from work.projects;
quit;
*** Create Row of Project Probs;
proc transpose data=work.projects prefix=prob out=tp_prob(drop=_name_);
 var prob;
 id ProjectID;
run;
*** Create Row of Project Values;
proc transpose data=work.projects prefix=value out=tp_value(drop=_name_);
 var value;
 id ProjectID;
run;
*** Create Matrix with [#Scenarios,value+probs];
data value_prob;
 format ScenarioID 8.;
 set tp_value;
 set tp_prob;
 do ScenarioID = 1 to 2**&n_proj; output; end;
run;
*** FullCalc_Outcome Mart;
data FullCalc_Outcomes_Datastep;
 set value_prob;
 *** Define Arrays;
 array ind[&n_proj] ind1-ind&n_proj;     ** Project y/n Indicator;
 array prob[&n_proj] prob1-prob&n_proj;  ** Project Success Probability;
 array value[&n_proj] value1-value&n_proj; ** Project Value;
 array scn_value[&n_proj] scn_value1-scn_value&n_proj;      ** Scenario Project Value (with YN);
 array scn_prob[&n_proj] scn_prob1-scn_prob&n_proj; ** Scenario Probabilty (with YN);
 bin = put(ScenarioID,binary32.);     ** Derive Binary String of ID;
 ScenarioProb = 1;                    ** Initialize Scenario Prob;
 do i = 1 to &n_proj;
  ind[i]=substr(bin,32+1-i,1);        ** Fill Indicator with respective Char of String;
  scn_value[i] = ind[i]*value[i];        ** Calculate Scenario Value;
  if ind[i] = 1 then scn_prob[i] = prob[i];  ** Calculate Scenario Probability;
  else scn_prob[i] = 1-prob[i];
  ScenarioProb = ScenarioProb * scn_prob[i]; ** Iteratively Multiply Scenario Probs ("AND" Probability);
 end;
  ScenarioSum = sum(of scn_value1-scn_value&n_proj);   ** Sum over Scenario Values;
  drop i;
run;
```

### SAS IML

```
proc iml;
      use sim.projects;
      read all;
      close;

      ScenarioID = t(1:2**nrow(prob));
      bin = putn(ScenarioID,"binary32.");

      pt=repeat(t(prob),2**nrow(prob),1);
      vt=repeat(t(value),2**nrow(prob),1);
      bt=j(2**nrow(prob),nrow(prob),1);
      do i = 1 to nrow(prob);
          bt[,i]=num(substr(bin,32-i+1,1));
      end;

      value = vt#bt;
      prob  = abs(abs(1-bt)-pt);

      ScenarioSum    = value[,+];
      ScenarioProb   = prob[,#];

      create FullCalc_Outcomes_IML var {ScenarioID ScenarioSum ScenarioProb};
      append;
      close FullCalc_Outcomes_IML;

quit;
```

# TIPP #5

## SO SIMULIEREN SIE DATEN AUS EINER MULTIVARIATEN VERTEILUNG

Verwendet den Mean Vektor und die Cov-Matrix

```
proc iml;
Mean = {42, 5200, 280}; /* population means */
Cov = {12 48 25,          /* population covariances */
       48 420 0,
       25 0 100};

N = 1000; /* sample size */
call randseed(123);
X = RandNormal(N, Mean, Cov); /* x is a 1000 x 3 matrix */

SampleMean = mean(X);
SampleCov = cov(X);
varNames = {Alter Volumen Events};
print SampleMean[colname=varNames],
SampleCov[colname=varNames rowname=VarNames];

/* write sample to SAS data set for plotting */
create MVN from X[colname=varNames]; append from X; close
MVN;
quit;
```

# SO SIMULIEREN SIE DATEN AUS EINER MULTIVARIATEN VERTEILUNG

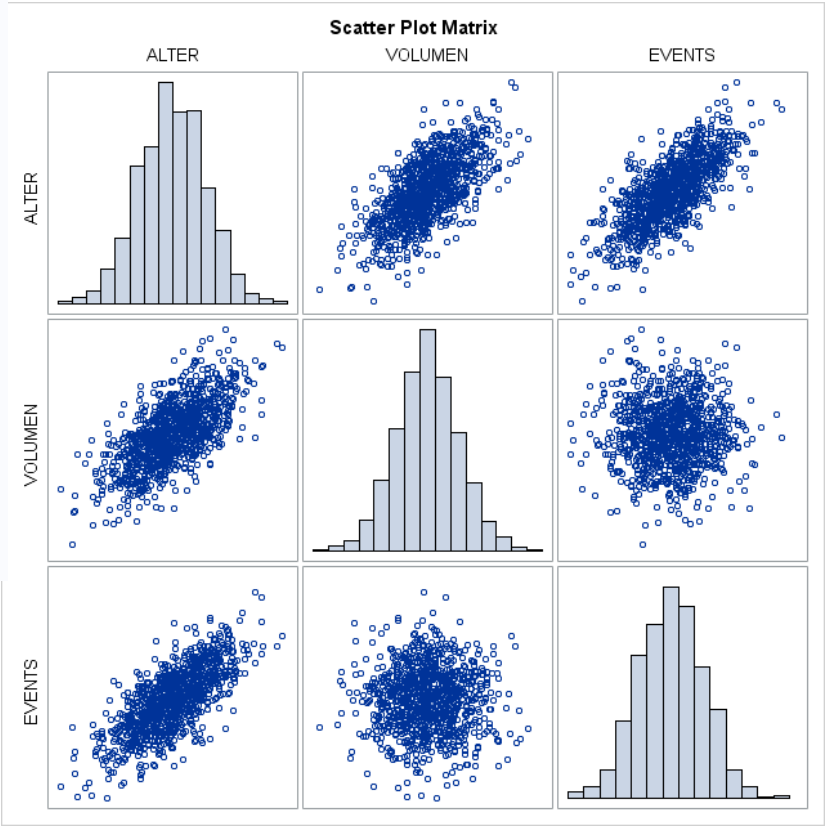
## Ergebnisse des Programm-Codes

**Simple Statistics**

| Variable | N | Mean | Std Dev | Sum | Minimum | Maximum |
|---|---|---|---|---|---|---|
| ALTER | 1000 | 41.96466 | 3.50861 | 41965 | 30.27959 | 53.47730 |
| VOLUMEN | 1000 | 5200 | 20.59750 | 5199748 | 5129 | 5266 |
| EVENTS | 1000 | 280.08244 | 10.29895 | 280082 | 248.16537 | 314.96263 |

**Pearson Correlation Coefficients, N = 1000**
**Prob > |r| under H0: Rho=0**

| | ALTER | VOLUMEN | EVENTS |
|---|---|---|---|
| ALTER | 1.00000 | 0.66484<br><.0001 | 0.72743<br><.0001 |
| VOLUMEN | 0.66484<br><.0001 | 1.00000 | -0.00653<br>0.8365 |
| EVENTS | 0.72743<br><.0001 | -0.00653<br>0.8365 | 1.00000 |



# TIPP #5

## SO SIMULIEREN SIE DATEN AUS EINER MULTIVARIATEN VERTEILUNG

- Die RAND Funktion im Datastep ist sehr mächtig für die Simulation von Daten für univariate Verteilungen.
- SAS IML ist das Werkzeug der Wahl für die Simulation von korrelierten Daten von multivariaten Verteilungen.
- SAS IML beinhält viele built-in Funktionen für die Simulation unterschiedlicher univariater und multivariater Verteilungen.
- SAS IML unterstützt auch die Matrix-Berechnungen um Datensamples von weniger häufig verwendeten Verteilungen zu ziehen.

- Hinweis: für den Spezialfall der multivariaten Normalverteilung bietet auch SAS STAT mit der SIMNORMAL Procedure eine Simulationsmöglichkeit.

# TIPP #6 VERMEIDEN SIE MACRO-LOOPS

Für jeden Durchlauf ein eigener Datastep

Einzelberechnung der Ergebnisse

Append der Ergebnisse

```
/*****************************************/
/* THIS CODE IS INEFFICIENT. DO NOT USE. */
/*****************************************/
%macro Simulate(N, NumSamples);

proc datasets nolist;
delete OutStats; /* delete data if it exists */
run;

%do i = 1 %to &NumSamples;
data Temp; /* create one sample */
 call streaminit(0);
 do i = 1 to &N;
  x = rand("Uniform");
  output;
 end;
run;
proc means data=Temp noprint; /* compute one statistic */
 var x;
 output out=Out mean=SampleMean;
run;
proc append base=OutStats data=Out; /* accumulate statistics */
run;
%end;
%mend;
/* call macro to simulate data and compute ASD. VERY SLOW! */
%Simulate(10, 1000) /* means of 1000 samples of size 10 */
```

# TIPP #6 VERMEIDEN SIE MACRO-LOOPS

Verschachtelter Do-Loop

Analyse: „BY“ SimulationRun

```
%macro Simulate(N, NumSamples);

   data Temp;
   call streaminit(0);
   do SimulationRun = 1 to &NumSamples;
            do i = 1 to &N;
               x = rand("Uniform");
               output;
            end;
    end;
   run;

   proc means data=Temp nway noprint;
      class SimulationRun;
      var x;
      output out=Out mean=SampleMean;
   run;

%mend;

/* call macro to simulate data and compute ASD */
%Simulate(10, 100)                /* means of 100 samples of size 10 */
```

# TIPP #7 BESCHLEUNIGEN SIE IHRE SIMULATIONS-LÄUFE DURCH UNTERDRÜCKUNG VON OUTPUT

- Bei der Simulations-Iteration sind wir typischerweise an den erzeugten Daten im SAS Dataset und weniger an den Ergebnissen im Output-Fenster oder den Graphiken interessiert.
- Optionen wie NOPRINT oder PLOTS=NONE können hier hilfreich sein.
- Weiters
  - Erstellung der Graphiken abschalten: ODS GRAPHICS OFF;
  - Über ODS alle Ergebnisse unterdrücken: ODS EXCLUDE ALL;
  - Den Tree-View im Results-Fenster nicht befüllen: ODS RESULTS OFF;
  - Die Notes im Log unterdrücken: OPTIONS NONOTES;
- Rick Wicklin präsentiert in seinem Paper „**Ten Tips for Simulating Data with SAS®”** folgende beiden Macros:

```
%macro ODSOff(); /* call prior to BY-
group processing */
     ods graphics off;
     ods exclude all;
     ods results off;
     options nonotes;
%mend;
```

```
%macro ODSOn(); /* call after BY-group
processing */
ods graphics on;
ods exclude none;
ods results on;
options notes;
%mend;
```

# TIPP #8

## PLANEN SIE IHRE SIMULATIONS-STUDIE BEVOR SIE STARTEN

- Starten Sie den Testlauf (Programmverifikation) mit 2-5 Iterationen.
- Starten Sie den Performancetest (Laufzeitermittlung) mit 100 – 1000 Iterationen.
- Stellen Sie sicher, dass Sie Ihr Programm vor dem „Run“ speichern!!! Damit Sie notfalls die Session ohne Verluste vollständig abbrechen können.
- Starten Sie mit einem „groben“ Grid und verfeinern Sie dort, wo Sie mehr Details benötigen
  - ein 20x20 Grid, benötigt die vierfache Laufzeit eines 10x10 Grids

**TIPP #9**

# NUTZEN SIE DAS WISSEN AUS SAS BLOGS, LITERATUR VON SAS PRESS, SAS GLOBAL FORUMS, PAPERS

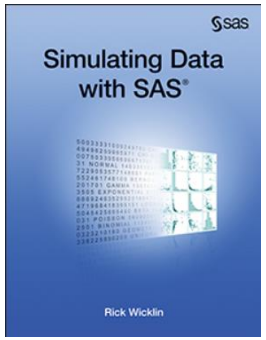
## Do-Loop Blog von Rick Wicklin http://blogs.sas.com/content/iml/

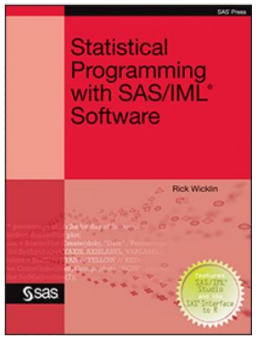
The DO Loop
Statistical programming in SAS with an emphasis on SAS/IML programs

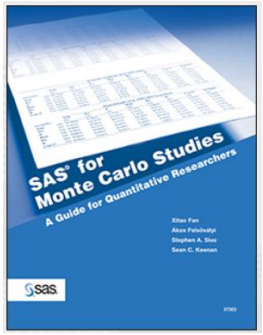
SAS GLOBAL FORUM
SAS GLOBAL USERS GROUP 2016 CONFE
ONLINE PROCEEDINGS

http://support.sas.com/events/sasglobalforum/previous/online.html

Paper SAS1387-2015 Ten Tips for Simulating Data with SAS®, Rick Wicklin
http://support.sas.com/resources/papers/proceedings15/SAS1387-2015.pdf



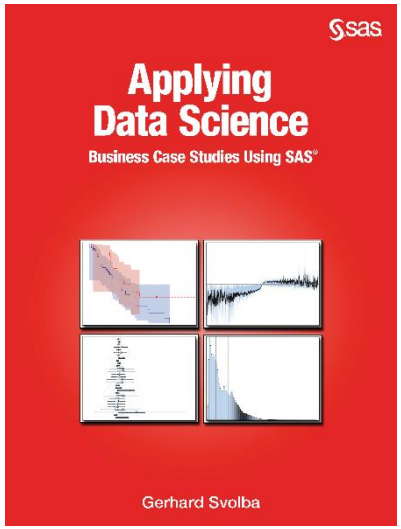

**SAS® for Monte Carlo Studies: A Guide for Quantitative Researchers**
Xitao Fan, Ph.D.
Akos Felsovalyi, M.S.
Stephen A. Sivo, Ph.D.
Sean C. Keenan, Ph.D.
**http://support.sas.com/publishing/authors/felsovalyi.html**

## Bücher von Rick Wicklin

http://support.sas.com/publishing/authors/wicklin.html

# TIPP #10

# NEUES BUCH IN SAS PRESS „APPLYING DATA SCIENCE: BUSINESS CASE STUDIES USING SAS“ MIT ZWEI SIMULATIONS-STUDIEN




## Applying Data Science:

### Business Case Studies Using SAS

Data Science and Analytics helps you to solve your business questions

The SAS® Analytic Plattform is perfectly suited to perform these analyses

**Eight Case Studies with Business Background, Results, Interpretation and SAS Code**

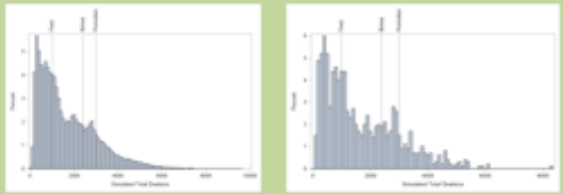
SAS Press (expected 2017)

http://www.sascommunity.org/wiki/Applying_Data_Science_-_Business_Case_Studies_Using_SAS

Gerhard Svolba

## Using Monte Carlo Methods to Simulate the Most Likely Outcome

*Will the Sales Manager keep his job (when we look at his sales pipeline)?*



## Simulation of the Processes of the Monopoly® Board Game

*How can we simulate complex environments to get insight in the most frequent processes?*