

Empirische Poweranalyse

Stefan Englert
Institut für Medizinische Biometrie und Informatik
Im Neuenheimer Feld 305
Heidelberg
englert@imbi.uni-heidelberg.de

Zusammenfassung

Dieser Vortrag stellt verschiedene Verfahren zur Durchführung und Optimierung von empirischen Poweranalysen vor. Diese sind gewöhnlich sehr zeit- und rechenaufwendig, so dass sich Techniken zur Optimierung der Laufzeit entsprechender SAS-Programme lohnen.

Auf folgende Aspekte wird dabei eingegangen:

- Möglichkeiten der Realisierung empirischer Powerberechnungen in SAS
- Optimierung des Aufbaus der Simulation
- Ausgabe notwendiger Kenngrößen mittels ODS OUTPUT
- Allgemeine laufzeitoptimierende Programmtechniken

Die Beispiele basieren auf SAS in der Version 9.2 und 9.1. Zur Beurteilung der Performance wurden Echtzeitmessungen durchgeführt.

Schlüsselwörter: Empirische Poweranalyse, Simulation, Performance, ODS OUTPUT

1 Einleitung

Für eine Vielzahl von statistischen Modellen und Tests existieren bisher entweder keine Methoden die Power analytisch zu berechnen, oder die entwickelten Methoden sind nur approximativ gültig und somit möglicherweise unzuverlässig. Empirische Powerberechnungen halten deshalb immer weiter Einzug in die Planung klinischer Studien.

Teilweise wird bereits in der Planung einer klinischen Studie vermutet, dass die Voraussetzungen eines statistischen Verfahrens verletzt sind. In dieser Situation können empirische Powerberechnungen verwendet werden, um die Stabilität des Verfahrens zu untersuchen. Dabei werden für spezielle Werte der Modellparameter eine große Anzahl hypothetischer Datensätze zufällig generiert. Auf jeden dieser Datensätze wird anschließend ein geeigneter statistischer Test angewendet und die Power anhand der Rate, wie oft die Nullhypothese abgelehnt wurde, geschätzt. Die Durchführung dieser Simulation ist dabei sehr zeit- und rechenaufwendig. Möchte man eine angestrebte Power von 80% mit einer Genauigkeit von 0,2% schätzen, so sind bereits 100'000 Wiederholungen notwendig (siehe Tabelle 1). Überlegungen, wie die Performance in diesen Fällen positiv beeinflusst werden kann, lohnen sich deshalb sehr schnell.

Tabelle 1: Geschätzte Genauigkeit einer empirischen Poweranalyse in Abhängigkeit vom Stichprobenumfang (Power = 0,8) [1]

Stichprobenumfang	1'000	10'000	100'000	1'000'000
Genauigkeit	±0,025	±0,008	±0,002	±0,001

Diese Problematik verschärft sich noch weiter, wenn durch empirische Poweranalysen der optimale Stichprobenumfang bestimmt werden soll. Hierbei wird für eine gegebene Ausgangssituation gewöhnlich der Stichprobenumfang so lang erhöht, bis eine gewünschte Power erreicht ist.

Eine weitere Problematik bei der praktischen Durchführung empirischer Powerberechnungen ist, dass nur wenige in SAS integrierte Prozeduren eine direkte tabellarische Ausgabe der zur Beurteilung notwendigen Kennziffern (bsp. p-Werte) ermöglichen. Meist werden diese nur im Output-Fenster als Listing wiedergegeben und können somit nicht direkt weiterverwendet werden.

In diesem Beitrag werden diese beiden Herausforderungen angegangen. Es werden unterschiedliche Szenarien zur Durchführung einer empirischen Poweranalyse vorgestellt, wobei jeweils Vorschläge gemacht werden, wie man in diesen Fällen mit einfachen Mitteln die Performance positiv beeinflussen kann. In den dargestellten Methoden wurden die verwendeten Testprozeduren bewusst nicht neu programmiert, sondern stets auf in SAS implementierte Prozeduren zurückgegriffen. Somit ermöglicht die vorgestellte Methode schnelle und einfache empirische Poweranalysen aller in SAS implementierten Prozeduren.

Abschließend werden zusammenfassend einige Tipps gegeben, die man immer berücksichtigen kann, da sie stets Vorteile bringen.

Die Performancemessungen wurden durch zwei SAS Makros realisiert, mit denen eine Echtzeitmessung durchgeführt werden kann. Das erste Makro initialisiert die Zeitmessung und das zweite berechnet die Zeit, die seit der Initialisierung vergangen ist, und gibt diese Information aus.

2 Aufbau der Simulation

In SAS können empirische Powerberechnungen grundsätzlich mit zwei verschiedenen Herangehensweisen durchgeführt werden [2-6]. Einerseits ist es möglich die für die Simulation notwendigen Datensätze einzeln zu erstellen und zu analysieren und dies über ein Makro wiederholt auszuführen (Iteration im Makro-Schritt). Andererseits können die verschiedenen hypothetischen Datensätze direkt gemeinsam generiert und ausgewertet werden (Iteration im Data-Schritt). Zu jedem der Ansätze ist im Folgenden ein Beispielprogramm angegeben.

2.1 Iteration im Makro-Schritt

Abbildung 1 zeigt den Aufbau eines Simulationsprogrammes, das die empirische Power durch eine Iteration in einem Makro-Schritt realisiert. Zuerst wird ein hypothetischer Datensatz erzeugt und anschließend eine passende Prozedur zur Auswertung angewendet.

det. Die erhaltenen p-Werte werden ausgegeben, an einen weiteren Datensatz angefügt und dort gesammelt. Abschließend wird die relative Häufigkeit der Ablehnung der Nullhypothese (empirische Power) bestimmt.

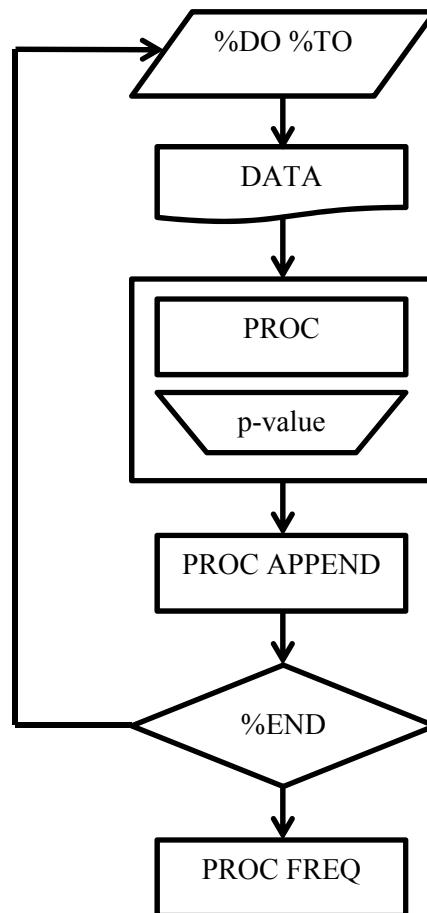


Abbildung 1: Flow-Chart einer Simulation mit Iteration in einem Makro-Schritt

Möchte man beispielsweise die Power einer ANOVA mit zwei Gruppen und 63 Beobachtungen pro Gruppe simulieren, so lässt sich dies in SAS wie folgt implementieren.

```

%MACRO ANOVA1 (n) ;

%DO k = 1 %TO &n;
  DATA Simulation;
  DO i=1 TO 63;
  method=1; x=RANNOR(-1); OUTPUT;
  method=2; x=RANNOR(-1)+1; OUTPUT; END;

  PROC ANOVA DATA=Simulation NOPRINT OUTSTAT=Out;
  CLASS method; MODEL x=method;
  RUN; QUIT;

PROC APPEND DATA=OUT BASE=Results; RUN;

```

```
%END;
```

```
DATA Results; SET Results(WHERE=( _SOURCE_='method' ));  
IF PROB > 0.05 then reject = 'no '; ELSE reject = 'yes';  
RUN;
```

```
PROC FREQ DATA=Results; TABLE reject; RUN; QUIT;  
%MEND ANOVA1;
```

Bei dieser Variante fällt insbesondere negativ auf, dass von SAS sehr viel in den Log geschrieben wird, was den Programmablauf merkbar ausbremst. Dem kann man entgegenwirken, indem man die Ausgabe sämtlicher Notes unterdrückt (OPTIONS NONOTES;). In dem angegebenen Beispiel führt diese Vorgehensweise mindestens zu einer Verdopplung der Geschwindigkeit. Da die erhaltenen p-Werte automatisch mit OUTSTAT in einen Datensatz geschrieben werden, sollte außerdem die Ausgabe im Output-Fenster mittels NOPRINT unterdrückt werden. Dies wirkt sich ebenfalls positiv auf die Performance aus.

2.2 Iteration im DATA-Schritt

In SAS ist es ebenfalls möglich alle für die Simulation notwendigen Teildatensätze gemeinsam zu erstellen und diese anschließend durch ein BY-Kommando auszuwerten. Einen Flow-Chart des Programmablaufs zeigt Abbildung 2.

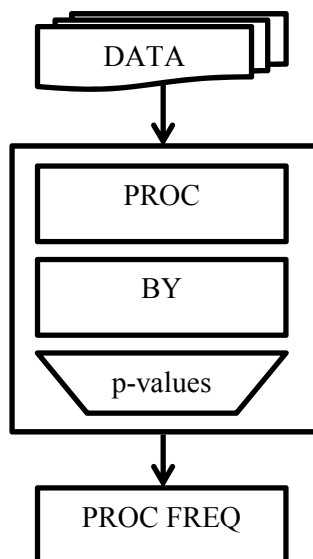


Abbildung 2: Flow-Chart einer Simulation mit Iteration im Data-Schritt

Für das Beispiel einer ANOVA mit zwei Gruppen und 63 Beobachtungen pro Gruppe lässt sich dies in SAS wie folgt realisieren.

```

%MACRO ANOVA2 (n) ;
DATA Simulation;
    DO k=1 TO &n;
        DO i=1 TO 63;
            method=1; x=RANNOR(-1); OUTPUT;
            method=2; x=RANNOR(-1)+1; OUTPUT; END;
        END;
END;

PROC ANOVA DATA=Simulation NOPRINT OUTSTAT=Out;
    CLASS method; MODEL x=method;
    BY k;
RUN;

DATA Results; SET Out(WHERE=( _SOURCE_='method' ));
IF PROB > 0.05 then reject = 'no '; ELSE reject = 'yes';
RUN;

PROC FREQ DATA=Results; TABLE reject; RUN; QUIT;
%MEND ANOVA2;

```

Diese Variante erzeugt initial einen sehr großen Datensatz, der anschließend in einer entsprechenden Prozedur ausgewertet wird. Bei 1'000'000 Wiederholungen erreicht der Datensatz im angegebenen Beispiel eine Gesamtgröße von ca. 4 GB, sodass ausreichend Hauptspeicher zur Verfügung stehen muss.

Wichtig ist demnach auch, dass der Datensatz bereits passend für die Testprozedur erstellt wird, sodass zuvor keine Sortierung notwendig ist.

2.3 Performancevergleich

Während in Variante 1 (Iteration durch ein Makro) die Programmierung linear aufgebaut ist und deshalb sehr einfach zu programmieren ist, laufen in Variante 2 sämtliche Berechnungen parallel ab. Eine Performancemessung zeigt deutliche Unterschiede zwischen beiden Verfahren und zwischen den SAS Versionen 9.1 und 9.2.

Tabelle 2: Performancebewertung der unterschiedlichen Herangehensweisen
(SAS Version 9.2)

Wiederholungen	Gesamtdauer der Simulation in Sekunden (Echtzeitmessung)		
	Variante 1	Variante 1 (nonotes)	Variante 2
10	0,53	0,38	0,18
100	4,23	2,42	0,20
1'000	40,55	23,25	0,36
10'000	372,03	232,03	4,47
100'000		2082,83	39,72
1'000'000			529,30

Tabelle 3: Performancebewertung der unterschiedlichen Herangehensweisen
(SAS Version 9.1)

Wiederholungen	Gesamtdauer der Simulation in Sekunden(Echtzeitmessung)		
	Variante 1	Variante 1 (nonotes)	Variante 2
10	1,34	0,23	0,05
100	12,69	2,50	0,25
1'000	113,23	20,67	0,25
10'000	1077,00	206,53	3,98
100'000		1996,15	27,42
1'000'000			372,88

In beiden Varianten steigt die benötigte Gesamtzeit linear mit der Anzahl der Simulationsdurchläufe an. Allerdings ist Variante 2 etwa 100-mal schneller als Variante 1 in SAS 9.2 und etwa 250-mal schneller als in SAS 9.1. Ab 10'000 Wiederholungen erfolgen bei Variante 1 so viele Ausgaben in den Log, dass SAS die Berechnung unterbricht und eine Reaktion des Nutzers anfordert. Unterdrückt man sämtliche Notes, so halbiert sich die Simulationsdauer in SAS 9.2. In SAS 9.1 verringert sich die Simulationsdauer sogar um den Faktor 5, sodass sich keine Performanceunterschiede mehr zwischen SAS in Version 9.2 und 9.1 zeigen. Diese Variante ist jedoch immer noch etwa 50-mal langsamer als Variante 2.

Eine genauere Analyse des Programmablaufs in Variante 1 zeigt, dass etwa die Hälfte der Gesamtzeit (bei 100'000 Wiederholungen 1011,59 Sekunden) darauf verwendet wird, um die erhaltenen p-Werte an den Ergebnisdatensatz anzufügen. Scheinbar stellt das wiederholte Öffnen und Schließen von verschiedenen sehr kleinen Datensätzen SAS vor größere Herausforderungen als die Handhabung und Auswertung sehr großer Datensätze.

In der hier durchgeführten Simulation zeigt sich zusätzlich, dass SAS 9.1 in Variante 2 geringfügig schneller arbeitet als SAS 9.2.

3 Ausgabe statistischer Kenngrößen

Zur Beurteilung der statistischen Power ist es notwendig die von einer Prozedur berechneten p-Werte in Tabellen zur weiteren Bearbeitung abzuspeichern. Bei einer ANOVA ist dies durch OUTSTAT bereits vorimplementiert und deshalb sehr komfortabel möglich. Nur wenige in SAS integrierte Prozeduren bieten jedoch diese Möglichkeit. Meistens werden die p-Werte nur im Output-Fenster als Listing wiedergegeben und können somit nicht direkt weiterverwendet werden. Eine komfortable Möglichkeit diese Informationen auszulesen bietet das Output Delivery System (ODS). Mithilfe von ODS OUTPUT ist es möglich den Output direkt in eine Tabelle umzuleiten.

Wir illustrieren dies anhand einer Survivalanalyse mittels Log-Rank-Test, bei der eine direkte tabellarische Ausgabe der zur Beurteilung notwendigen Kennziffern (p-Werte) nicht möglich ist.

Eine Abfrage mittels `ODS TRACE` ergibt, dass die Analyseergebnisse der Prozedur `LIFETEST` unter `HomTests` abgespeichert werden. Über `ODS OUTPUT HomTests` ist es deshalb möglich die Ergebnisse direkt in eine angegebene Tabelle (hier `logrankstat`) umzuleiten.

```
PROC LIFETEST DATA=Simulation;
...
ODS OUTPUT HomTests = logrankstat;
RUN;
```

Diese Vorgehensweise hat jedoch den unangenehmen Nebeneffekt, dass die Ausgabe im Output-Fenster nicht mehr mit `NOPRINT` unterdrückt werden darf, da andernfalls auch die Tabelle nicht mehr ausgegeben wird. Dies würde jedoch zu Performanceeinbußen führen. Deshalb ist es notwendig speziell die Ausgabe im Listing zu unterdrücken.

```
ODS LISTING CLOSE;
PROC LIFETEST DATA=Simulation;
...
ODS OUTPUT HomTests = logrank;
RUN;
ODS LISTING;
```

Die Verwendung von `ODS OUTPUT` ist in beiden vorgestellten Varianten des Programmierablaufs problemlos möglich. Wird innerhalb der Prozedur eine `BY`-Anweisung verwendet (Variante 2), so enthält der ausgegebene Datensatz die ermittelten Kenngrößen automatisch getrennt nach der `BY`-Variablen.

4 Schlussbemerkungen

Unser Vergleich der beiden Möglichkeiten zur Umsetzung empirischer Poweranalysen in SAS zeigt deutlich, wie wichtig es ist, den Aufbau der Simulation zu berücksichtigen. Iterationen sollten, wann immer es möglich ist, in einem Data- und nicht in einem Makro-Schritt durchgeführt werden.

Grundsätzlich sind zusätzlich folgende Mittel zur Steigerung der Performance einer empirischen Poweranalyse hilfreich

- Nur unbedingt notwendige Messgrößen berechnen lassen
- Datensätze bereits so erstellen, dass vor der Auswertung keine Sortierung notwendig ist
- Ausgaben im Log-Fenster einschränken, beispielsweise mittels `OPTIONS NONOTES;`
- Output soweit möglich mit `NOPRINT` bzw. `ODS LISTING CLOSE;` einschränken

Literatur

- [1] W.J. Dixon, F.J. Massey: Introduction to Statistical Analysis. McGraw-Hill, 1983.
- [2] SAS Institute, Inc., SAS/STAT User's Guide, Version 9.2, Cary, NC, USA 2009
- [3] J. Zhou, B. Coate, Y. He: Empirical Power Computation Using SAS for Schuirman's Two One-Sided Tests Procedure in Clinical Pharmacokinetic Drug-Drug Interaction Studies, Paper 154-29, SAS Users Group International 29, 2004
- [4] J. Zhou, Y. Yuan, M. Chen, B. Coate: Empirical Power for Higher-Order Cross-over Designs in Comparative Bioavailability Clinical Trials, Paper 184-30, SAS Users Group International 30, 2005
- [5] J. Zhou, J. Le, B. Coate: Empirical Power Estimation for Phase I Dose Proportionality Studies Based on Power-Law Model Using Confidence Interval Criteria, Paper 182-31, SAS Users Group International 31, 2006
- [6] X. Fan, A. Felsovalyi, S.A. Sivo, S.C. Keenan: SAS for Monte Carlo Studies: A Guide for Quantitative Researchers, SAS Publishing, Cary, NC, 2001