

# Modularisierung und Kapselung von SAS-Makros in einem Auswertungspaket für die Evaluation von Klausurergebnissen

Michael Volk, Beate Einsiedler, Rainer Muche

Institut für Epidemiologie und Med. Biometrie, Universität Ulm, Deutschland;  
[rainer.muche@uni-ulm.de](mailto:rainer.muche@uni-ulm.de)

Bei der Routineauswertung von immer wieder zu erhebenden Daten ist es üblich, entsprechende Standardprogramme zu erstellen. In SAS werden diese hauptsächlich als SAS-Makros mit der Übergabe von Standardparametern erstellt. Dabei stellt sich bei der Überarbeitung und Erweiterung dieser Makros oft das Problem, dass die Struktur der Parameterübergabe sowie der Beziehungen der entsprechenden Makrovariablen nur schwer nachzuvollziehen ist. Daraus ergibt sich oft ein großer Überarbeitungsaufwand. Für eine Überarbeitung des 2009 auf der KSFE in Halle vorgestellten SAS-Programms zur (teil-)automatisierten Analyse von Klausurergebnissen [2] wurde im Rahmen einer Bachelorarbeit [4] die Programmierung so umgestellt, dass eine Überarbeitung und Erweiterung in Zukunft viel einfacher möglich ist.

Die dafür eingesetzten Methoden sind die der Modularisierung und Kapselung. Mit der Hilfe der SAS Macro Language ist es möglich ein Programm in einzelne Module aufzuteilen. Dies hat den Vorteil einer besseren Übersichtlichkeit. Da jedes Makro nur ein Teil des gesamten Programms ist, können hier die einzelnen Schritte besser nachvollzogen werden. Außerdem können Fehler so schneller gefunden werden, da bekannt ist, in welchem Modul er zu suchen ist. Ein wichtiges Werkzeug dazu ist die Definition von lokalen und globalen Makrovariablen (MVAR): in jedem Makro können lokale MVARs definiert werden. Wenn z.B Makro A das Makro B aufruft, hat Makro B vollen Zugriff auf alle lokalen MVARs von Makro A. Dies ist so lange der Fall, bis eine lokale MVAR eines aufgerufenen Makros den gleichen Namen besitzt, wie eine MVAR des aufrufenden Makros. Das hat den Vorteil, dass Werte, die ein Makro schon berechnet hat, nicht nochmal im aufgerufenen Makro berechnet werden müssen. Allerdings ist hier sehr darauf zu achten, wann welche MVAR in welchem Makro geändert wird. Um sicher zu sein, dass keine Probleme auftreten können, besteht die Möglichkeit alle benötigten Werte mittels Makro-Übergabeparameter in einer gleichnamigen lokalen MVAR zu übergeben, somit kann auf den Wert im aufrufenden Makro nicht mehr zugegriffen werden. Dies Vorgehen wird auch Kapselung genannt: Jedes Makro hat nur Zugriff auf die eigenen MVARs (vgl. [3]).

Mit Zugriffen auf Datasets ist es etwas anders: Hier gibt es keine Hierarchie wie bei den MVARs, jedes Makro kann auf jedes Dataset in jeder Library zugreifen und die Daten dort manipulieren. Hier hilft die Modularisierung den Überblick zu behalten. Wenn ein Makro abgeschlossen wird, werden z.B. bei den Kenngrößen die Ergebnisse in ein existierendes permanentes Dataset gespeichert und anschließend alle selbst erzeugten Datasets gelöscht.

Eine Ausnahme bilden festgelegte kleine Hilfsmakros, die berechnete Zwischenwerte in einem temporären Dataset für die weitere Ausarbeitung hinterlegen.

In der Phase der Kenngrößenberechnung [1] wird jede Kenngröße von einem eigenen Makro berechnet, was den Vorteil der Austauschbarkeit mit sich bringt: wenn eine Kenngröße nicht mehr benötigt wird bzw. wenn es eine bessere Kenngröße für eine bestimmte Auswertung gibt, kann diese als neues Modul einfach ausgetauscht werden. Es müssen lediglich geringe Teile des Hauptmakros und im Modul für die Ausgabe geändert werden.

In dem Beitrag wird anhand der Auswertungssituation der Analyse von Klausurergebnissen in der Medizin das Vorgehen beispielhaft und nachvollziehbar aufgezeigt.

Literatur:

[1] Möltner A, Schnellberg D, Jünger J: Grundlegende quantitative Analysen medizinischer Prüfungen. GMS Z. Med. Ausbildung. 2006;23(3): Doc53

[2] Muche R, Janz B, Einsiedler B: Quantitative Analysen medizinischer Prüfungen mittels eines (teil-) automatisierten SAS-Programms. In: Spilke J, Becker C, Haerting J, Schumacher E (Hrsg.): KSFE 2009 – Proceedings der 13. Konferenz für SAS-Anwender in Forschung und Entwicklung, Shaker Verlag, Aachen, 2009, 187-194

[3] Pfister G: Fortgeschrittene Makro-Programmierung. Tutoriumsunterlagen, KSFE 2013, 2013

[4] Volk M: Automatisierte Analyse medizinischer Prüfungen mit SAS. BSc.-Arbeit, Hochschule Ulm; 2013