

Eine Vorlage – zahlreiche Outputs: Entwicklung einer effizienten Strategie für die Studiauswertung

Edelbert Arnold
Accovion GmbH
Helfmann-Park 10
65760 Eschborn
edelbert.arnold@accovion.com

Zusammenfassung

Im Rahmen der Auswertung von klinischen Studien gibt es vielfach Anforderungen, dass eine Vielzahl von Outputs basierend auf einer Vorlage erstellt werden muss. Das trifft für alle Tabellen, Listen und Grafiken zu, bei denen es entweder nur eine Vorlage in einem Berichtserstellungsplan gibt, oder auch mehrere Vorlagen dieselbe Struktur aufweisen. Bei Accovion wurde eine Strategie entwickelt, wie effizient mit solchen Wiederholungen der Ausgabestruktur umgegangen wird.

Häufig unterscheiden sich Outputs nur durch die Datenselektion, bei studienübergreifenden Auswertungen durch die darzustellenden Behandlungsgruppen. Der gewählte Prozess zielt darauf ab, den dazugehörigen Programmcode nur einmal zu erstellen, mehrfach aufzurufen, dabei aber dessen Übersichtlichkeit zu bewahren. Die Programmierer der einzelnen Outputs können sich dann auf deren spezifischen Inhalt konzentrieren. Wird die Darstellung mehrerer Subgruppen gewünscht, so wird die Wiederverwendbarkeit durch standardisierte Variablen optimiert. Sind verschiedene Behandlungskombinationen darzustellen, so lässt sich der damit verbundene Anpassungsaufwand durch Verwendung dazugehöriger Formatlisten minimieren. Möchte man Tabellen für weitere Populationen oder Subgruppen zusätzlich erstellen, so müssen in einer externen Datei lediglich entsprechende Kürzel hinzugefügt und nachfolgend die Programme neu gestartet werden.

Der gewählte Prozess steht im klaren Gegensatz zu dem zunächst einfach erscheinenden Weg, Programmcode zu kopieren und an wenigen Stellen anzupassen. Für alle Schritte nach der ersten Programmierung liefert dieses Vorgehen allerdings sichtliche Nachteile. Änderungswünsche an der Struktur sind mehrfach umzusetzen, Fehler mehrfach zu korrigieren. Validierung und Dokumentation sind aufwändig, die Einheitlichkeit der Darstellung gestaltet sich als schwierig. Dem entgegen wirkt die bei Accovion gewählte Strategie, die den Einsatz von einigen wenigen zentralen, übersichtlichen Makros mit projekt- und studienspezifischen Anpassungen beinhaltet. Der Planungsaufwand, der zu Beginn der Auswertung betrieben wird, zahlt sich im Nachhinein vielfach aus. In dieser Phase gilt es abzuwägen, für welche Wiederholungen sich die Integration im projekt- bzw. studienspezifischen Makrosystem lohnt. Möchte man etwa mehrere Listen mit verschiedenen Typen von Adverse Events erstellen (all, serious etc.), so kann eine effiziente Umsetzung in nur einem Programm außerhalb des Makrosystems erfolgen. Das System ist aber dahingehend erweiterbar, dass eine Vielzahl zunächst ungeplanter Wiederholungen hinzugefügt werden kann, z.B. für eine ganz neue Population, ohne dass die bisher erstellten Programme zur Outputerstellung geändert werden müssen.

Der vorgestellte Prozess ist an bestehende Tools bei Accovion angelehnt. Er liefert einen umso größeren Vorteil im Hinblick auf Effizienz und Qualität, je mehr Wiederholungen es im Rahmen einer Auswertung gibt.

Schlüsselwörter: Studienauswertung, SAS-Output, Wiederholung, Repeated Output, Effizienz, Änderungsfreundlichkeit, Ausgabe

1 Einleitung

Bei der Auswertung klinischer Studien wird eine Vielzahl von Outputs in Form von Tabellen, Listen und Grafiken erstellt. Dabei sind Effizienz und Qualität wesentliche Ansprüche, die an statistische Programmierer in Pharmafirmen und Auftragsforschungsunternehmen herangetragen werden. Betrachtet man sich die statistischen Analysepläne und die in der Regel zusätzlich erstellten Vorlagen zur Erstellung von Outputs, wird schnell deutlich, dass viele dieser geplanten Outputs auf ein und derselben Vorlage basieren oder Vorlagen die gleiche Struktur aufweisen.

Das bei Accovion entwickelte Makrosystem GENREPOUT (**generate repeated outputs**) ermöglicht es, effizient mit diesen Wiederholungen umzugehen. Der damit verbundene Programmcode wird nur einmal entwickelt und vielfach innerhalb des Studienteams verwendet. Dadurch verbleibt jedem Programmierer mehr Zeit für den spezifischen Inhalt des von ihm zu erstellenden Outputs, womit ein Beitrag zur Erhöhung der Qualität geleistet wird.

Die Konzipierung eines solchen Makrosystems ist von mehreren Faktoren abhängig. Für Accovion als CRO müssen neben den eigenen Prozessen und Standards die Vorgaben des Kunden berücksichtigt werden. Die Auswahl der Selektionsgrößen im Makrosystem kann von einer Studie zur nächsten ganz unterschiedlich ausfallen. Die am häufigsten vorkommende Größe ist sicher die im statistischen Analyseplan (SAP) definierte Population. Änderungen der gewählten Größen und deren Bezeichnungen im Laufe einer Studie sind allerdings nicht selten. Dafür besteht mit GENREPOUT ein änderungsfreundliches System.

2 Voraussetzungen

2.1 Anforderungen des Kunden

Bevor GENREPOUT für die Studienauswertung verwendet und angepasst wird, müssen einige grundlegende Anforderungen des Kunden geklärt werden:

- Auswertung im Accovionsystem oder im System des Kunden
- Relevante SOPs des Kunden, Good Programming Practice Guideline
- Grundsätzliche Vorgaben des Kunden für die Outputs
- Projekt- und studienspezifische Anforderungen des Kunden

Unabhängig davon, ob die Auswertung im System von Accovion oder in demjenigen des Kunden durchgeführt wird, muss geklärt werden, ob GENREPOUT angewandt werden kann. Folgende Faktoren können dagegen sprechen:

- Kunde hat sein eigenes System, mit solchen Wiederholungen umzugehen
- GENREPOUT passt nicht zu vorhandenen und zu verwendenden Tools

Ist beides nicht der Fall, dann muss überprüft werden, ob das Makrosystem keiner Kunden-SOP widerspricht, was allerdings sehr unwahrscheinlich ist. Die Erfahrung zeigt, dass Accovion in den meisten Fällen die Freiheit besitzt, die Programm- und Makrostruktur selbst zu definieren und erst die Ausgabe fest definiert ist. Folgende grundsätzliche Vorgaben bekommt Accovion häufig mitgeteilt:

- Ausgabetypp: Am häufigsten sind lrt, rtf und pdf für Tabellen und Listen, cgm, (e)ps und pdf für Grafiken
- Layout: Schriftgröße, Seitenränder, Anzahl der Spalten und Zeilen
- Allgemeine Vorgaben für Titel, Fußnoten, Nummerierung

Die Details dieser Vorgaben können nicht gegen die Verwendung von GENREPOUT sprechen, sie beeinflussen aber selbstverständlich dessen Details. An dieser Stelle sei betont, dass es sich bei GENREPOUT nicht um ein starres Makrosystem handelt, sondern eine Sammlung von wenigen, übersichtlichen Makros, die für jedes Projekt bzw. jede Studie, so weit es sinnvoll erscheint, angepasst wird. Diese Anpassungen sind schließlich im Wesentlichen von folgenden Punkten abhängig:

- Auswertung eine Studie oder studienübergreifende Auswertung
- SAP: Beschreibung der durchzuführenden Analysen
- Genaue Spezifikation der Outputs mit darzustellenden Spalten/ Kurven etc.

Zu Beginn einer jeden Studienaushwertung macht man sich mit den entsprechenden Dokumenten vertraut. Im Anschluss daran kann abgeschätzt werden, für welche der Selektionsgrößen wie Population und Studie die Einbindung in das Makrosystem lohnt. Mehr Details zur Auswahl der Selektionsgrößen befinden sich im Kapitel 3.1.

2.2 Situation bei Accovion

Die Programmierumgebung bei Accovion bildet die Basis für die Verwendung von GENREPOUT. Im folgenden sind die wesentlichen Eckpunkte gelistet:

- System: Red Hat Enterprise Linux Release 5 (bis Januar 2010 HP-UX V11)
- SAS® als Softwarepaket für die Auswertung, aktuell V9.2
- Header-Header-Makrosystem HFS: Grundlage ist eine Excel-Tabelle, mit der Titel und Fußnoten verwaltet werden
- TABMAX: Makrosystem für Standardauswertungen (Auszahlungen mit PROC FREQ, deskriptive Statistiken und Standardausgaben)
- ROCMAX: Makrosystem von Accovion zur Generierung, Archivierung und Änderungskontrolle von Datensätzen und Outputs
- Relevante SOPs zur Entwicklung und Validierung von Programmen

GENREPOUT wurde im Wesentlichen unabhängig vom Betriebssystem und der SAS®-Version erstellt. Ob die Programme zur Output-Erstellung von Accovion erstellte TABMAX-Makros enthalten, ist für GENREPOUT völlig unerheblich. Auch spielt es keine Rolle, ob die Auswertungsprogramme später in das ROCMAX-System eingebunden werden, mit dem man alle Programme mit Hilfe eines einzigen Programmlaufs erstellen und falls gewünscht, mit älteren Versionen vergleichen kann.

Allerdings besteht eine enge Verzahnung zum Header-Header-Makrosystem HFS von Accovion, welches daher im folgenden Abschnitt in seinen Grundzügen vorgestellt wird.

Das Makrosystem ist konform zu den Accovion-SOPs und unterstützt die Vorgabe, Programmcode nicht beliebig oft zu wiederholen und ihn stattdessen nur einmal zu definieren und beliebig oft zu verwenden.

2.3 HFS-Makrosystem

Die wesentliche Aufgabe von HFS ist es, Titel und Fußnoten für die zu erstellenden Outputs zu verwalten und diese dann in Form von TITLE und FOOTNOTE Statements zur Verfügung zu stellen. Dazu kommt die Bereitstellung weiterer Attribute, die zu den Outputs gehören, wie der Programmname, der Name des Programmierers und die Tabellennummer. Diese Attribute können dann in Titel oder Standardfußnoten der Outputs integriert werden.

Im Folgenden werden die wesentlichen Bestandteile des HFS-Systems erläutert.

Ausgangspunkt ist eine Excel-Tabelle, in die bestimmte Attribute aller zur Auswertung gehörenden Tabellen, Listen und Grafiken eingetragen werden. Für ein paar wenige Einträge wird in Abbildung 1 die Struktur aufgezeigt:

_NAME	_CHAR	_TNUM	_TITLE	_FNUM	_FOOT	TBLNUM	PROGRAM	Respsble
dm001	t	1	Demographic data: Sex, age and race			1.1	dm001.sas	XY
ef001	f	1	Plot of least square means and corresponding 95% confidence intervals for the primary efficacy variable PEFF	1	Note: For details about PEFF derivation, see SAP, chapter 7.5.10	2.1	ef_lsm.sas	EA
ef001	f	2	Timepoint: After 24 h					
ae006	d	1	All adverse events	1	Note: MedDRA = Medical Dictionary for Regulatory Activities, Adverse events according to MedDRA version 12.1	3.1	ae_list.sas	SM
ae006	d			2	TEAE = Treatment Emergent Adverse Event			
ae007	d	1	Serious adverse events	1	Note: MedDRA = Medical Dictionary for Regulatory Activities, Adverse events according to MedDRA version 12.1	3.2	ae_list.sas	SM
ae007	d			2	TEAE = Treatment Emergent Adverse Event			

Abbildung 1: Beispiel für die Verwendung von titles.xls

Die Excel-Tabelle titles.xls wird anschließend als csv-Datei in der Programmierumgebung gespeichert. Durch den folgenden einfachen Makroaufruf wird diese csv-Datei in den permanenten Datensatz HFS_META umgewandelt und im gleichen Verzeichnis gespeichert:

```
%hfs_get;
```

Abbildung 2: Makroaufruf zur Umwandlung von titles.csv in HFS_META

Bevor der Output erstellt wird, muss ein Aufruf, wie in Abb. 3 dargestellt, erfolgen:

```
hfs_gid=dm001;
%hfs_head(...Parameter...);
```

Abbildung 3: Beispielaufruf zur Erstellung der TITLE und FOOTNOTE Statements für die Demographietabelle dm001; mit Makroparametern im Aufruf von %hfs_head kann global für alle Outputs der Studie eine Standardausgabe definiert werden.

Bei diesem Aufruf erfolgt zunächst die Selektion derjenigen Einträge aus HFS_META, die zu dem in &hfs_gid enthaltenen Output gehören. Diese Einträge werden dann in TITLE und FOOTNOTE Statements umgewandelt. Nach der Ausgabeprozedur folgt das HFS-Postprocessing mit 2 kurzen Makroaufrufen, die für GENREPOUT von keiner besonderen Bedeutung sind. Letztendlich sieht der Output folgendermaßen aus:

```
KSFE2010/01

Demographic data: Sex, age and race

-----

*** Hier kommt der Hauptteil mit den Daten ***;

-----

Accovion GmbH: 25FEB2010 / dm00100t.lst / dm001.sas      Page 1 of 1
```

Abbildung 4: Beispiel für die Ausgabe des Titels im Tabellenkopf. In der Fußnote findet man eine mögliche Standardfußnote mit Firmennamen, Datum der Erstellung, Namen der Ausgabedatei und des Programms sowie die Seitennummerierung.

In manchen Fällen macht es Sinn, dass Makrovariablen in der Excel-Tabelle verwendet und später aufgelöst werden. Möchte man etwa in der obigen Tabelle „Table 1.1: “ in der Haupttitelzeile verwenden, so müsste in titles.xls `_TITLE=Table &tblnum: Demographic data ...` gesetzt werden (&tblnum in HFS definiert). In GENREPOUT wird intensiv mit der Verwendung von Makrovariablen gearbeitet, siehe Kapitel 3.3. Der Aufruf von %hfs_get ist unabhängig von der Existenz von Makrovariablen. Bevor hfs_head durchgeführt wird, müssen allerdings alle Makrovariablen aufgelöst sein.

3 Aufsetzen von GENREPOUT

In diesem Kapitel wird zunächst dargestellt, wie auf Basis der Vorgaben für die Auswertung ein sinnvolles Konzept für den effizienten Umgang mit Wiederholungen aufgebaut wird. Als nächstes werden die wenigen, beteiligten Programme und Makros dargestellt und wie diese je nach Ausgangssituation aufgesetzt werden.

3.1 Auswahl der Selektionsgrößen

Welche Selektionsgrößen verwendet werden, kann je nach Art der Auswertung und deren Details sehr unterschiedlich ausfallen. Die bisherige Erfahrung hat gezeigt, dass sich GENREPOUT fast immer lohnt.

In den meisten Fällen liegt für eine Studie oder auch eine studienübergreifende Auswertung die Anzahl der gewünschten Outputs im dreistelligen Bereich. Die Anzahl der unterschiedlichen Vorlagen liegt dabei sehr häufig weit unter 50% der Gesamtzahl.

Betrachtet man diese Vorlagen, so können dementsprechend die Selektionsgrößen gewählt werden. Im Folgenden sind die am häufigsten vorkommenden Größen angegeben:

- Population (z.B. Safety, ITT, Per-Protocol)
- Subgruppen (z.B. Geschlecht, Altersgruppe, Land)
- Studienauswahl bei studienübergreifenden Auswertungen, z.B. einer Integrated Summary of Efficacy/Safety (ISE/ISS)
- Studienphase (z.B. Titration, Maintenance)

Am häufigsten kommt die Population in GENREPOUT vor. Bei fast jeder Studienauswertung ist deren Aufnahme in das Makrosystem sinnvoll. Ausnahmen sind Safety Updates und ISS-Analysen, wo bei allen Analysen nur eine Population verwendet wird. Vor allem bei studienübergreifenden Auswertungen muss berücksichtigt werden, dass die Auswahl der Behandlungsgruppen unterschiedlich sein wird. Dies sollte beim Aufsetzen von GENREPOUT bedacht werden.

Grundsätzlich ist die Aufnahme einer Selektionsgröße in GENREPOUT dann sinnvoll, wenn es basierend auf dieser Größe zahlreiche Repeated Outputs gibt. Das bedeutet, es ist bei vielen Outputs der Fall, dass Vorlagen wiederholt verwendet werden und sich die Outputs nur durch diese Größe unterscheiden. Gibt es etwa nur 2 oder 3 solcher Repeated Outputs, wird sich die Aufnahme der entsprechenden Selektionsgröße nicht lohnen.

Zusätzlich macht die Aufnahme einer Größe in GENREPOUT dann Sinn, wenn Wiederholungen bei vielen Vorlagen oder bei mehreren Auswertungsgruppen, etwa sowohl in der Safety als auch in der Efficacy, gewünscht sind. Die Aufnahme von Selektionsgrößen wie AE-Typen (all, serious, possibly related etc.) oder verschiedene Efficacy-Parameter in GENREPOUT lohnt sich eher nicht, da deren Implementierung leicht innerhalb eines Programms erfolgen kann.

Der Vorteil von GENREPOUT im Hinblick auf die Effizienz wird umso größer sein, je mehr Repeated Outputs es bei einer geringen Anzahl von Selektionsgrößen gibt.

3.2 Einfluss der Selektionsgrößen auf die Outputs

Die Auswahl der Größen für GENREPOUT kann sich an folgenden Stellen im Output auswirken:

- Titel und Fußnoten, z.B. durch den Zusatz „Safety Population“ oder eine Unter Nummerierung als Teil der Tabellenummer im Titel
- Texte innerhalb des Outputs, z.B. „Male“ und „Female“ für die Subgruppe „Sex“

- Bezeichnung der Behandlungsgruppen
- Datenselektion, sichtbar durch die Anzahl der Patienten
- Dateiname, z.B. „s“ für „Safety Population“ als Bestandteil des Dateinamens


3.3 Vorstellung von GENREPOUT

Unabhängig von der Situation im Unternehmen kann eine effiziente Strategie für den Umgang mit diesen Selektionsgrößen gefunden werden. Der im Folgenden vorgestellte Weg ist eng mit der Situation bei Accovion verknüpft, insbesondere mit dem HFS-Makrosystem, kann aber stark abhängig vom Kunden variieren.

Zunächst wird die Implementierung der am häufigsten vorkommenden Selektionsgröße, der Population beschrieben.

3.4 Implementierung der Population als Selektionsgröße

In der Excel-Tabelle **titles.xls** wird eine Spalte für die Population hinzugefügt, in die man Kürzel für die gewünschte Population einträgt, die gleichzeitig auch als Bestandteil des Dateinamens verwendet werden:



_NAME	_CHAR	_TNUM	_TITLE	_FNUM	_FOOT	_POP	TBLNUM	PROGRAM	Respsble
dm001	t	1	Demographic data			0si	1.1	dm001.sas	XY

Abbildung 5: Beispiel für die Integration der Population in titles.xls

Durch die Populationsspalte wird der Wunsch mitgeteilt, dass die Demographietabelle für die 3 Populationen „All Subjects“, „Safety Population“ und „Intention-To-Treat“ erstellt werden soll. Dieser Text soll als zusätzliche Zeile im Titel erscheinen und die 0, s bzw. i als Bestandteil des Dateinamens verwendet werden.

In der als csv-Datei gespeicherten Excel-Datei werden als nächstes durch das Programm **titles_add_mv.sas** Makrovariablen hinzugefügt. Im folgenden Beispiel soll auch eine Tabellenummer ausgegeben werden, wobei für die Population eine Unternummerierung in der Form .1, .2 und .3 erscheinen soll. Danach sieht titles.csv so aus:

```
_NAME;_CHAR;_TNUM;_TITLE;_FNUM;_FOOT;_POP;TBLNUM;PROGRAM;Respsble;
dm001;t;1;Table &tbl_id: Demographic data;;;0si;1.1;dm001.sas;XY;
dm001;t;2;&pop_txt; ; ; ; ; ;
```

Abbildung 6: Beispiel für die Integration von Makrovariablen in titles.csv

Innerhalb des Makros **%def_pop** werden die Makroparameter für alle Populationen definiert. Der wesentliche Makrocode könnte für die 3 Populationen wie folgt aussehen:

```
%LET n_pop = %LENGTH(&_pop);

%DO i = 1 %TO &n_pop;
  %LET chpop&i = %SUBSTR(&_pop, &i, 1);
  %IF &&chpop&i EQ 0 %THEN %DO;
    %LET whpop&i = usubjid NE " ";
    %LET tpop&i = All subjects;
  %END;
  %IF &&chpop&i EQ s %THEN %DO;
    %LET whpop&i = saffl EQ "Y";
    %LET tpop&i = Safety population;
  %END;
  %IF &&chpop&i EQ i %THEN %DO;
    %LET whpop&i = ittfl EQ "Y";
    %LET tpop&i = Intention-to-treat population;
  %END;
%END;
```

Abbildung 7: Beispiel für die Integration der Population im Makro %def_pop

Mit diesem Programmcode werden für alle 3 Populationen Makroparameter für die Datensелеktion und den Teil des Titels erstellt, der die Population beschreibt. Damit dieser Programmcode funktioniert, müssen vorher im Makro die Einträge in HFS_META selektiert werden, die zum entsprechenden Output gehören, und dieser in Makroparameter umgewandelt werden. Das bedeutet, dass %def_pop im Programmcode zur Output-Erstellung nach %hfs_get folgen muss.

Damit wiederum der Aufruf von %hfs_head zur Erstellung der TITLE und FOOTNOTE Statements funktioniert, der wegen der unterschiedlichen Titel pro Population einmal erfolgen muss, müssen vorher die Makroparameter definiert sein, die in HFS_META enthalten sind. Dies erfolgt im Makro %def_mv_hfs. Der wesentliche Programmcode lautet im vorliegenden Fall wie folgt:

```
*** &pop_add enthält die Unternummerierung für die Population ***;
%IF &n_pop GT 1 %THEN %LET pop_add = .&popn;
%ELSE %LET pop_add = ;;

*** Definition von &tbl_id und &pop_txt ***;
%LET tbl_id = %TRIM(&tblnum)&pop_add;
%LET pop_txt = &&tpop&popn;
```

Abbildung 8: Beispiel für die Integration der Population im Makro %def_mv_hfs (siehe Abbildung 9 für die Definition von &popn)

Die Funktion von HFS_META, %hfs_get und %hfs_head wurde bereits kurz im Abschnitt 2.3. beschrieben. Dort wurde auch der wesentliche Programmcode zur Outputerstellung bei der Standardverwendung von HFS vorgestellt. Wie dieser mit GENREPOUT aussieht, wird als nächstes aufgezeigt:

```
%LET hfs_gid = dm001;
%hfs_get;
%def_pop;

%MACRO all_tab(&popn);
    ...
    SET demo(WHERE=&&whpop&popn);
    ...
    %LET outname = &hfs_gid.0&&chpop&popn..t.lst;
    %def_mv_hfs;
    %hfs_head(...Makroparameter...);
    ...inklusive Ausgabeprozedur...
%MEND all_tab;

%MACRO call_all_tab;
    %DO _popn = 1 %TO &n_pop;
        %all_tab(&_popn);
    %END;
%MEND call_all_tab;
%call_all_tab;
```

Abbildung 9: Beispiel für die Integration der Population im Programmcode für die Erstellung der Demophietabellen für 3 Populationen

HFS_META wird durch den Aufruf von %hfs_get erzeugt. Mit %hfs_gid wird gesteuert, wie durch %def_pop die Makroparameter für die Populationen definiert werden und welche Einträge aus HFS_META in %hfs_head verwendet werden. Durch Aufruf von %def_mv_hfs werden die Makroparameter &tbl_id und &pop_txt definiert. Der Aufruf von %all_tab innerhalb der Schleifenstruktur besitzt den Vorteil, dass unabhängig von der Anzahl der gewünschten Populationen das Programm bestehen bleiben kann. Möchte man eine Population nicht mehr haben oder hinzufügen, so muss lediglich die Populationsspalte in der Excel-Tabelle durch Löschen oder Hinzufügen des entsprechenden, in %def_pop definierten Zeichens modifiziert werden, während Programme unverändert bleiben können.

Resultat sind die 3 Outputs mit den Dateinamen dm00100t.lst, dm0010st.lst und dm0010jt.lst, in denen über die Makroparameter &whpop1, &whpop2 und &whpop3 die Patienten selektiert wurden, die Tabellennummern 1.1.1, 1.1.2 und 1.1.3 lauten und in einer separaten Titelzeile die Population entsprechend &tpop1, &tpop2 und &tpop3 enthalten ist.

Anstelle einer separaten Zeile für die Population im Titel kann der entsprechende Text natürlich alternativ an den Haupttitel angehängt werden. Etliche Variationen sind denkbar, auch unter Beteiligung der Fußnoten oder des Behandlungscode, falls nicht für alle Populationen die gleiche zugrunde liegt.

3.4.1 Implementierung von Subgruppenauswertungen

Die Implementierung entspricht zu großen Teilen derjenigen für die Populationen. In titles.xls wird eine zusätzliche Spalte `_SG` definiert. Darin werden Zeichen eingetragen, die eindeutig einer Subgruppenauswertung zugeordnet sind, z.B. der Altersgruppe, des Landes oder eines Krankheitsstatus zu Beginn der Studie.

Im Programm `titles_add_mv.sas` wird der Makroparameter `&sg_txt` zur Beschreibung im Titel definiert.

In den Makros `%def_pop` und `%def_mv_hfs` werden weitere Makroparameter analog zu den Populationen definiert.

Auch die Erstellung der Tabelle ohne Subgruppen kann integriert werden.

Zusätzlich zu den Makros, die schon für die Populationen beschrieben worden sind, wurde das kleine Makro `%det_sg` erstellt, das eine standardisierte Auswertung aller Subgruppen durch die Verwendung von Standardvariablen ermöglicht. Bei der Verwendung der Altersgruppe und des Geschlechts kann der wesentliche Teil des Codes so aussehen:

```
*** &sgvar: Variable, die in &&whsg&sgn verwendet wird ***;  
*** sgnum: Schleifenzähler aus dem aufrufenden Programm ***;  
  
%MACRO det_sg(sgvar, sgnum);  
  LENGTH sgtext $100.;  
  %IF &&chsg&sgnum EQ a %THEN %DO;          *** a: Altersgruppe ***;  
    sgnum = &sgvar.n;  
    sgtext = &sgvar;  
  %END;  
  %IF &&chsg&sgnum EQ b %THEN %DO;          *** b: Geschlecht ***;  
    IF &sgvar EQ "M" THEN sgnum = 1;  
    IF &sgvar EQ "F" THEN sgnum = 2;  
    IF &sgvar EQ "M" THEN sgtext = "Male Subjects";  
    IF &sgvar EQ "F" THEN sgtext = "Female Subjects";  
  %END;  
%MEND det_sg;
```

Abbildung 10: Beispiel für die Standardisierung von Subgruppenvariablen

Der Aufruf des Makros `%det_sg` innerhalb der Subgruppenschleife im Outputerstellungsprogramm ermöglicht dann die standardisierte Auswertung über die Subgruppen Altersgruppe und Geschlecht. Beim obigen Beispiel wird davon ausgegangen, dass es in den Ausgangsdaten für die Altersgruppe je eine numerische Variable mit Postfix `~N`

und den Werten 1,2 usw. und eine Textvariable gibt, etwa AGEGRPN und AGEGRP. Für das Geschlecht geht man von einer Variablen mit der Belegung „M“ oder „F“ aus. Nach der Ausführung von %det_sg werden stattdessen im weiteren Verlauf des Programms immer die Variablen sgnum und sgtext verwendet.

Im Outputerstellungsprogramm werden innerhalb %all_tab mit dem zusätzlichen Parameter &sgn die Makroparameter für die Datenselektion und das Zeichen für den Dateinamen verwendet, in %call_all_tab ist eine weitere Schleife für die Subgruppen enthalten. Auswertungen können innerhalb der Schleife mit der Zeile „BY sgnum sgtext;“ erfolgen. Eine Auswertung über Adverse Events mit dem Geschlecht als Subgruppe könnte dann folgendermaßen aussehen:

```
KSFE2010/01
```

```
Table 3.1.2: All Adverse Events  
Safety Population, subgroup sex
```

```
Male Subjects
```

```
...
```

```
Female Subjects
```

```
...
```

```
Note: ...
```

Abbildung 11: Beispiel einer Subgruppentabelle unter Verwendung des Geschlechts

3.4.2 Implementierung der Studie als Selektionsgröße

Bei studienübergreifenden Auswertungen gibt es häufig separate Tabellen für die Studie oder eine Gruppe von Studien. Die Implementierung ist wiederum derjenigen für die Populationen ähnlich.

In titles.xls wird die Spalte _STUDY hinzugefügt, titles_add_mv.sas wird durch &STUDY_TXT ergänzt, %def_pop erhält weitere Makroparameter, %def_mv_hfs erwartet wenige Anpassungen und das Outputerstellungsprogramm neben der Verwendung der definierten Makroparameter eine Schleife für die Studien. Als Startwert der Schleife kann die 0 verwendet werden, um eine kombinierte Auswertung aller Studien zu erhalten.

Häufig ist es bei diesen Auswertungen so, dass sich auch die Behandlungsgruppen unterscheiden. Dabei kann es unter bestimmten Umständen hilfreich sein, mehrere Behandlungsvariablen zu verwenden, welche wiederum in %def_pop definiert werden könnten. Die Spaltenüberschriften der verschiedenen Studien definiert man am besten extern in Formatlisten, damit diese nicht in jedem Programm und für jede Studie separat definiert werden müssen.

3.4.3 Weitere Selektionsgrößen

Im Grunde genommen lässt sich jede Größe in GENREPOUT integrieren, für die es als sinnvoll erachtet wird. Die Implementierung kann ähnlich erfolgen wie in den vorangegangenen Abschnitten beschrieben.

Falls dies als möglich und hilfreich erscheint, kann das System noch weiter automatisiert werden. Dabei sollte jedoch beachtet werden, dass die Programme so verständlich bleiben, dass der klare Zeitvorteil gegenüber dem Weglassen im Makrosystem gewahrt bleibt. Alle Programmierer eines Teams müssen sich zunächst mit GENREPOUT vertraut machen, dessen Vorteil auch umso größer ist, je häufiger damit gearbeitet wird.

Bereits im Abschnitt 3.1 wurde erwähnt, dass sich die Aufnahme von Größen in GENREPOUT eher nicht lohnt, wenn die Implementierung effizienter durch Wiederholungen innerhalb eines Programms erfolgen kann. Will man etwa mehrere Listen mit Adverse Events basierend auf derselben Vorlage erstellen, so kann dies durch Schleifenbildung innerhalb eines Outputerstellungsprogramms erfolgen. Dann muss innerhalb dieser Schleife die jeweilige Datenselektion erfolgen. Titel, Tabellenummer und Dateiname müssen dann auch innerhalb der Schleife angepasst werden. Entsprechend könnte der Ablauf innerhalb eines Programms erfolgen, wenn mehrere Efficacy-Parameter auf Basis der gleichen Vorlage erstellt werden sollen.

4 GENREPOUT während der Auswertungsphase

In diesem Abschnitt wird beschrieben, wie GENREPOUT in den Verlauf einer Auswertung integriert wird. Dies beginnt mit der Phase des Aufsetzens, wird mit der eigentlichen Auswertungsphase fortgesetzt und endet mit der Validierung des Makrosystems vor der Abgabe der Outputs.

4.1 Integration der Phase des Aufsetzens

Der richtige Zeitpunkt für das Aufsetzen von GENREPOUT liegt dann vor, wenn statistischer Analyseplan und Vorlagen für die Outputs im Wesentlichen stabil sind und die Programmierung der Outputs noch nicht begonnen wurde. Zusätzlich ist es von Vorteil, wenn der Teil der Analysedatensätze von der Struktur her feststeht, in dem die Variablen und deren Attribute enthalten sind, die in GENREPOUT für die Datenselektion verwendet werden sollen. Ausgehend von der CDISC-Struktur sollten diese Variablen im Subject-Level Analysis Dataset ADSL definiert sein.

Wichtig ist zusätzlich zu der auswertungsspezifischen Anpassung der Module von GENREPOUT die Dokumentation, wie in den Outputerstellungsprogrammen damit gearbeitet wird. Das System wird in dieser Phase dadurch getestet, indem für verschiedene Kombinationen der Selektionsgrößen Vorlagenprogramme erstellt werden.

Diese können dann im weiteren Verlauf der Programmierung der Outputs von den Teammitgliedern verwendet werden. In der Regel kann derjenige Teil der Programme unverändert bleiben, der sich auf die Verwendung von GENREPOUT bezieht.

4.2 GENREPOUT während der Auswertungsphase


Der einfachste Fall ist ein zu 100% finaler statistischer Analyseplan mitsamt den Vorlagen für die Outputs. In diesem Fall kann auch GENREPOUT und dessen Verwendung bis zum Ende der Auswertungsphase unverändert bleiben. Dies ist aber in den seltensten Fällen der Fall.

Möchte man neue Outputs erstellen, müssen neue Einträge in titles.xls erfolgen und neuer Programmcode generiert werden. Dies ist in einer Auswertung mit oder ohne GENREPOUT gleichermaßen der Fall.

Werden einfach nur Wiederholungen bereits existierender Outputs für andere Datenselktionen gewünscht, so reicht mit GENREPOUT die Angabe der entsprechenden Zeichen in titles.xls und die Ausführung der Programme. Dies wird auch dadurch unterstützt, dass im Auswertungsprogramm von Anfang an immer alle zur Selektion nötigen Variablen behalten werden.

Manchmal wird für zahlreiche Outputs eine ganz neue Population oder eine neue Subgruppe gewünscht, die in der Phase des Aufsetzens von GENREPOUT noch nicht bekannt war. Dann reichen wenige Änderungen in GENREPOUT, das Hinzufügen der entsprechenden Zeichen in titles.xls und die Ausführung des Outputerstellungsprogramms.

Etwas aufwändiger wird die Umsetzung, wenn sich die Auswertung so ändert, dass das Hinzufügen einer neuen Selektionsgröße notwendig erscheint. Wird zum Beispiel bei einer studienübergreifenden Auswertung bei den bereits integrierten Größen Population und der Subgruppen für einen Teil der Outputs zusätzlich die Auswertung nach der Studie gewünscht, so müssen die Module dementsprechend angepasst werden. In titles.xls wird die `_STUDY`-Spalte hinzugefügt:



<code>_NAME</code>	<code>_CHAR</code>	<code>_TNUM</code>	<code>_TITLE</code>	<code>_FNUM</code>	<code>_FOOT</code>	TBLNUM	<code>_POP</code>	<code>_SG</code>	<code>_STUDY</code>	PROGRAM	Respsble
ae002	t	1	All adverse events	1	Note: ...	3.1	s	All	0123	ae002.sas	XY

Abbildung 12: Integration der Studie als zusätzliche Selektionsgröße in titles.xls

Die Abbildung 12 zeigt ein Beispiel zur Erzeugung von Tabellen mit Adverse Events für die Safety Population, allen in `%def_pop` definierten Subgruppen und für 3 Studien kombiniert und separat. Im Outputerstellungsprogramm wird eine Schleife zur Analyse der Studien ergänzt und an wenigen Stellen die Makroparameter hinzugefügt. Diese Ergänzungen können natürlich auf die Programme begrenzt werden, für die es separate Auswertungen nach der Studie geben soll.

Gibt es mehrere Selektionsgrößen in GENREPOUT, so kommt es häufig vor, dass nicht alle möglichen Kombinationen dieser Selektionen gewünscht werden. Dies kann durch das Hinzufügen von `%IF`-Bedingungen in `%call_all_tab` gesteuert werden.

Bei Subgruppenauswertungen kommt es mitunter vor, dass es sich erst im Laufe der Auswertung herausstellt, dass die Verteilung der Ausprägungen nicht wie zunächst erwartet aussieht. Gibt es etwa eine Auswertung nach der Rasse, sind aber 99% in der Gruppe White, dann wird durch diese Auswertung keine neue Aussage ermöglicht. Die Verteilung kann, falls gewünscht, in GENREPOUT programmtechnisch festgestellt

werden, und damit auch Programmfehler vermieden werden, wenn eine Subgruppenausprägung gar nicht auftritt.

4.3 Validierung von GENREPOUT

GENREPOUT ist nicht als fixes generisches Makrosystem gedacht. Es soll im Gegenteil offen für eine Vielzahl von möglichen Kombinationen von Selektionsgrößen mit samt der Möglichkeit weiterer Automatisierungen gehalten werden. Daher muss GENREPOUT für jede Auswertung neu validiert werden, was wegen der Übersichtlichkeit der Module jedoch nicht viel Zeit in Anspruch nimmt. Zusätzlich zur „Validierung durch Anwendung“ wird Source Code Review reichen.

5 Fazit

GENREPOUT ist ein Konzept für den effizienten Umgang mit Wiederholungen von Outputs, dass dazu beiträgt, dass vielfach benötigter Programmcode nur einmal definiert und vielfach verwendet wird. Bei Accovion wurde ein System von wenigen, übersichtlichen, mit Hilfe von SAS[®] generierten Modulen erzeugt, das sich nahezu bei jeder Auswertung als sinnvoll erweist. Das hat die Erfahrung der vergangenen Jahre immer wieder gezeigt.

Der Planungsaufwand zu Beginn der Auswertungsphase zahlt sich nachher mehrfach aus. Die Programmierer eines Teams können sich mehr auf den spezifischen Inhalt ihrer Outputs konzentrieren. Ferner ist GENREPOUT änderungsfreundlich. Das Hinzufügen, Ändern und Löschen von bestimmten Auswertungen wird im Hinblick auf Wiederholungen von Outputs basierend auf gleichen Vorlagen stark unterstützt. Dies trägt sowohl zu erhöhter Effizienz als auch Qualität der Auswertung bei.