

# Excel gesteuerte TFL Erstellung mittels SAS-Makros: Erfahrung aus der kontinuierlichen Auswertung wiederholter Querschnittstudien

Marius Sieverding  
ARGUS Statistik GmbH  
Karl-Heinrich-Ulrichs-Str. 20A  
10785 Berlin  
Marius.sieverding@argus-statistik.de

Raffaella Matteucci Gothe  
ARGUS Statistik GmbH  
Karl-Heinrich-Ulrichs-Str. 20A  
10785 Berlin  
raffaella.gothe@argus-statistik.de

## Zusammenfassung

Mit dem Ziel, aufwändige, wiederkehrende Studien zu automatisieren, wurden Programme und Makros zusammengestellt, welche über eine Excel-Datei gesteuert werden. Dies macht es möglich, viele Einstellungen im Layout und Struktur der TFLs zu ändern, ohne den Programmcode anzupassen.

**Schlüsselwörter:** Makro, TFLs, Excel, Automatisierung

## 1 Problemstellung

Das Ziel des Projektes bestand in der Auswertung mehrerer sich wiederholenden Querschnittstudien. Hierbei müssen alle Inhalte des Fragebogens, sowie alle abgeleiteten Variablen des SAP in Tabellenform dargestellt werden. Dies beläuft sich auf > 200 Tabellen. Die Daten werden im CISC Standard aufbereitet, was bedeutet, erstellen der Study Data Tabulation Model (SDTM) und Analysis Data Model (ADaM) aus den Rohdaten. Diese Umfrage wurde in 2 verschiedenen Studiensamples durchgeführt, über 3 Jahre wiederholt, mit insgesamt 6 Auswertungen, die durchgeführt werden, um den zeitlichen Verlauf bestimmter Merkmale abzubilden. Zusätzlich sollten einige Auswertungen stratifiziert werden (7 Stratifizierungsvariablen mit bis zu 9 Ausprägungen jeweils) sowie teilweise standardisiert. Dieses Paket wiederum wurde zeitversetzt in 4 Ländern gestartet. Es ergeben sich somit

200 TFLs \* 2 Studiensamples \* 6 Auswertungen \* 4 Länder (\*7 Stratifizierungsvariablen \* bis zu 9 Ausprägungen jeweils)

Dies stellte uns vor die Herausforderung, sehr große Menge an Tabellen wiederholt zu erstellen, wobei zwischen den Ländern und den zeitlich versetzten Auswertungen immer wieder kleine Änderungen möglich sein mussten.

## 2 Lösungsansatz

Hierfür haben wir ein Set von standardisierten und validierten TFL Programmen entwickelt/angepasst, welches über eine Excel-Datei gesteuert wird. In dieser Datei werden alle Informationen gespeichert, welche zur Erstellung einer bestimmten Tabelle notwendig sind: Angabe der Analysevariable, Subgruppe der Beobachtungseinheiten in der Tabelle, Titel, Fußnoten und Algorithmen für weitere Möglichkeiten zur individuellen Anpassung der Tabelle. Somit lassen sich Änderungen schnell durchführen, ohne den SAS-Programmcode direkt ändern zu müssen. In der Excel-Datei sind zusätzlich Informationen zu Deckblatt, Stratifizierung und Standardisierung gespeichert.

### 2.1 Beschreibung

Die Art und Weise, wie die Programme zu TFL Erzeugung strukturiert sind und in den übergeordneten Prozess eingebunden werden, ist in Abbildung 1 dargestellt. Für jede Studie werden fixe Standardparameter definiert, welche i.d.R keine Anpassungen benötigen und daher direkt im SAS-Code festgeschrieben werden können. Diese beinhalteten z.B. Studienkürzel und Name des Sponsors. Die Standard-Makros und Tabellen-Makros werden geladen, die Excel-Datei über ein `PROC IMPORT` eingelesen sowie die Makro-Variablen aus dieser Datei erzeugt über den Code, der in SAS-Code 1 **Fehler! Verweisquelle konnte nicht gefunden werden.** gezeigt wird.

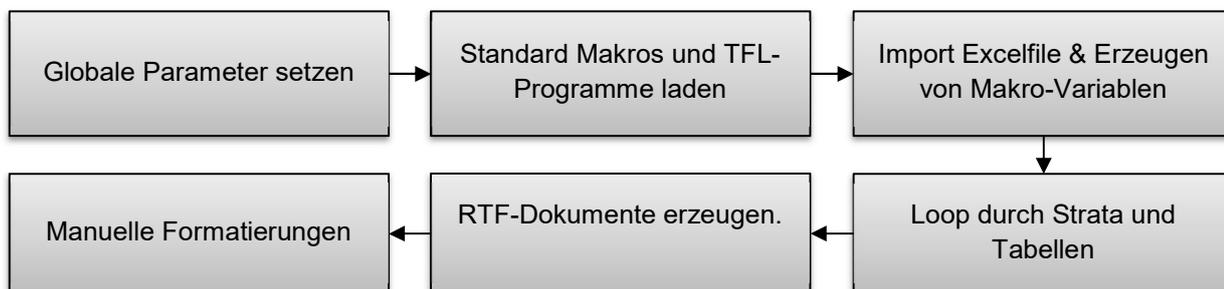


Abbildung 1: Flowchart Ablauf der TFL Erzeugung

```
%let id=%sysfunc(open(temp));  
%syscall set(id);  
%let rc=%sysfunc(fetchobs(&id,&j.));  
%let id=%sysfunc(close(&id));
```

#### SAS-Code 1: Erzeugen und Abrufen von Makro-Variablen

Mehrere verschachtelte Schleifen laufen durch alle ausgewählten Strata und ausgewählten Tabellen. Für die Spezifizierung der Tabellen wird in einem SAS-Dokument für alle gewünschten Tabellen das entsprechende Makro eingefügt, mit einem Verweis auf die entsprechende Zeile im Excel-Dokument (siehe SAS-Code 2). Hierbei können auch mehrere aufeinander folgende Tabellen mit einem Programm erstellt werden (siehe erstes Beispiel im genannten SAS-Code).

```

%tables(Minimum = 1, Maximum = 15);
%tables_subvalue(Minimum = 16, Maximum = 20);
%tables(Minimum = 21, Maximum = 21);
%tables(Minimum = 22, Maximum = 22);
%numeric_2_strata(tab_num = 23);
%numeric_4_strata(tab_num = 24);

```

### SAS-Code 2: Auflistung Tabellen-Programme mit Verweis auf Excel-Dokument

Jedes in SAS-Code 2 genannte Standard-TFL-Makro durchläuft eine Schleife von der Starttabelle (Minimum) bis zur Endtabelle (Maximum) und ruft weitere Makros auf, welche die eigentliche Tabelle aus den Makro-Variablen erzeugen (siehe auch SAS-Code 3).

## 2.2 Syntax

Im SAS-Code 3 **Fehler! Verweisquelle konnte nicht gefunden werden.** ist die Abfolge der verschiedenen Prozeduren dargestellt. Das zentrale `%categ_table()` Makro soll hier nur exemplarisch zeigen, wie alle notwendigen Parameter als Makro-Variablen an die TFL-Makros weitergegeben werden. Die im vorherigen SAS-Code 2 genannten Makros, wie z.B. `%tables()` haben im Prinzip die Aufgabe, eine Schleife zu erstellen und das `%categ_table()` Makro für alle genannten Tabellennummern nacheinander auszuführen.

Nicht weiter aufgeführt ist der Code, welcher aus Daten den Report generiert. Hier sind verschiedene Vorgehensweisen möglich. Im bestehenden Fall wird die ausgewählte Variable mit der ausgewählten Subgruppe aus der ADQS gebildet. Durch ein `merge` an die entsprechende Subgruppe der ADSL werden eventuell fehlende Werte festgestellt. Über eine `PROC FREQ` oder `PROC SUMMARY` werden die Häufigkeiten erzeugt und in einem `Datastep` formatiert (in diesem Fall z.B. spezielle Vorgaben des Sponsors zur Auf- und Abrundung).

```
/*Tabellenliste einlesen (temp) und Makro-Variablen
erzeugen*/
%let id=%sysfunc(open(temp));
%syscall set(id);

/*Stratifizierungsliste einlesen (temp2) und Makro-
Variablen erzeugen*/
%let id2=%sysfunc(open(temp2));
%syscall set(id2);

/*Loop durch Anzahl an bestimmten Tabellen aus der
Tabellenliste durchführen, Makro-Variablen der
entsprechenden Zeile laden*/
%do j=&Minimum. %to &Maximum.;
%let rc=%sysfunc(fetchobs(&id,&j.));

/*Gleiches für Stratifizierungen*/
%do i=1 %to &MaxStrata.;
%let rc2=%sysfunc(fetchobs(&id2,&i.));

/*Erzeugen von Tabellen mit Informationen aus den Makro-
Variablen*/
%categ_table(
    Strata_Label = %sysfunc(TRIM(&Strata_Var_Label.)),
    Strata_Value = %sysfunc(TRIM(&Strata_Value.)),
    table = %title,
    programName = &programName.,
    footnote1_tab = &Footnote.,
    footnote2_tab = &Footnote2.,
    footnote3_tab = &Footnote3.,
    datasource = &Dataset.,
    subgroup_m = &subgroup1. ,
    subgroup_Cond = &subgroup_Cond1. ,
    variable_Cond = &variable_Cond1.,
    sample_Cond = &sample_Cond1.,
    Strata_Cond = &Strata_Cond.,
    varname = &Variable_name.,
    N_subGr = &N_subGr.,
    pght=&pght.,
    Multiple=&Multiple.);

/*Schließen der Schleifen*/
%end;
%end;

/*Schließen der SAS-Files temp und temp2*/
%let id=%sysfunc(close(&id));
%let id2=%sysfunc(close(&id2));
```

**SAS-Code 3:** Schematische Darstellung des Codes zur Erstellung der TFLs

Mit einem `PROC REPORT` werden die formatierten Daten nach Standard Layout des Sponsors ausgegeben und mittels `ODS RTF` zu einem Report zusammengefügt. Hierzu ist in SAS-Code 4 das Schema der ODS RTF Konfiguration zu sehen. Zum Start eines jeden Dokumentes wird ein Deckblatt mittels des `%cover` Makros erzeugt. Dies beinhaltet globale Parameter wie Studientitel, Kennzeichnung, Namen der verantwortlichen Personen und die Versionsnummer des Dokumentes. Mittels

`options(toc_data="No")` wird ein Eintrag des Titelblatts in das Inhaltsverzeichnis verhindert. Mittels `ODS PROCLABEL` werden aus den entsprechenden Einträgen der Excel-Datei die Tabellenummer und der Titel definiert.

Einige Formatierungen müssen später im \*.rtf-Dokument vorgenommen werden. Das Inhaltsverzeichnis ist im Dokument immer das allererste Objekt und damit vor dem Titelblatt. Dieses muss manuell vor das Inhaltsverzeichnis geschoben werden. Außerdem ist das Verzeichnis unsichtbar, bis es markiert und aktualisiert wurde. Erst danach sind alle Einträge zu sehen. Hier haben wir uns für den Einsatz eines VBA-Makros entschieden, da jede Stratifizierungsvariable in einem eigenen \*.rtf-Dokument gespeichert wird und daher bis zu 14 Dokumente pro Land und Auswertung bearbeitet werden müssen. Mittels des VBA-Makros lassen sich diese Aufgaben automatisiert ausführen.

```
ods tagsets.rtf file="&filepath.&rft_name..rtf"
  style=styles.client
  options(contents='Yes' toc_data='yes' TOC_LEVEL ='1')
  NOGTITLE nogfootnote ;

ods noproctitle;

/*Die Titelseite sollte kein Eintrag im Inhaltsverzeichnis
haben*/
ods tagsets.rtf options(toc_data='No');

/*Makro für Titelseite*/
%cover;

/*Inhaltsverzeichnis wieder aktivieren*/
ods tagsets.rtf options(toc_data='Yes');

/*Titel und Nummer werden durch Makro-Variablen aus der
Excel-Datei erstellt*/
ods rtf startpage=now;
ODS PROCLABEL = "Table %sysfunc(TRIM(&Table_Number.))
                %sysfunc(TRIM(&Table_Name.))";

/*PROC REPORT erstellt die Tabelle*/

ods rtf startpage=now;
ODS PROCLABEL = "Table %sysfunc(TRIM(&Table_Number.))
                %sysfunc(TRIM(&Table_Name.))";

/*PROC REPORT erstellt die Tabelle*/

...

ods tagsets.rtf close;
```

#### SAS-Code 4: Schemata der ODS RTF Struktur

### 2.3 Beispiel

Als kurzes Beispiel zur Vorgehensweise bei Veränderungen im Programm Code bei z.B. Änderung der Nachkommastellen. Es könnte eine neue Spalte in der Excel-Datei eingefügt werden, welche eine Y/N-Flag mit dem Namen „round“ beinhaltet (siehe Abbildung 2). Nach dem Einlesen der Excel-Datei steht diese nun als Makro-Variable &round zur Verfügung und kann entsprechend dem SAS-Code 4 genutzt werden, um zwischen Tabellen mit 2 Nachkommastellen (&round. = "Y") und Tabellen mit einer Nachkommastelle (%else) zu differenzieren.

I	J	K	L
Footnote2	Footnote3	round	TYPE
Missing refers to unknown smoking		N	categorical
		Y	continuous
		N	categorical
Current CC smoker is defined as someone who at the time of the		N	categorical

**Abbildung 2:** Neue Y/N-Flag in Excel-Datei zur Erzeugung von Makro-Variablen

```
%if %EVAL("%sysfunc(TRIM(&round.))" = "Y") %then %do;
    /*Runden auf 2 Nachkommastellen*/
%end;
%else %do;
    /*Runden auf 1 Nachkommastellen*/
%end;
```

**SAS-Code 5:** Makro-Variablen %round zur Differenzierung von Nachkommastellen

### 3 Fazit

Das vorgestellte System ist ein Beispiel für eine maximale Automatisierung der TFL-Erstellung bei gleichzeitig vorhandener Flexibilität für Anpassungen und Veränderungen.

#### 3.1 Vorteile des Systems

Ein wesentlicher Vorteil des Systems ist es, schnell auf Änderungswünsche des Kunden reagieren zu können. Durch die Möglichkeit, für alle Tabellen eine Makro-Variablen zu erstellen und mit Werten zu füllen, können Veränderungen an den Tabellen schnell auf globalem Maßstab angelegt werden. Es reicht, eine neue Spalte in der Excel Datei anzulegen und entsprechend im Programm zu referenzieren. Zusätzlich lassen sich schnell verschiedene Tabellen erzeugen und hinzufügen. Durch Hinzufügen von Zeilen und Verweis mit einem Standard-TFL-Makro auf die entsprechende Zeilennummer, können Tabellen dem Report hinzugefügt werden. Ad hoc Anfragen, welche sich aus vorhandenen Variablen und Subgruppen beantworten lassen, sind schnell und optisch konsistent erstellt.

Stratifizierungen, als Modul programmiert, können einfach hinzugeschaltet und variiert werden. Solange eine Gruppe aus der ADSL definierbar ist, kann diese als Stratifizierung in der Excel-Datei definiert und im Programmcode abgerufen werden.

Aufgrund der schier Masse des Outputs, stellt sich an vielen Stellen nicht die übliche Frage, ob eine Programmierung „per-hand“ sich besser eignet oder eine Prozedur dafür geschrieben werden soll. Dieses System verlangt nach einer maximalen Automatisierung.

rung, um den gewünschten Output zu bewerkstelligen. Sofern dieses modular aufgebaut ist, können Teile davon gut in andere Projekte integriert werden, in welchen eventuell weniger Automatisierung gefragt ist.

### **3.2 Nachteile des Systems**

Das erste Einrichten der Struktur ist zeitaufwändig. Viele einzelne Aspekte sind schneller „per-hand“ für eine spezifische Situation programmiert als ein allgemeingültiges globales Programm entwickelt ist. Durch die lange Laufzeit der Studie und die geplanten Wiederholungen, war die Automatisierung jedoch ein klares Ziel und eine Notwendigkeit.

Ein weiterer Nachteil ist, dass gewisse Vorgaben an die Struktur der Daten gegeben sind. Dies ist zum einen dem auf das CDISC ADaM Standard zurück zu führen und zum anderen der Tatsache, dass Subgruppen immer über die ADSL definierbar sein sollten. Dies kann bei hierarchischen Fragen zu Problemen führen, da alle Personen, welche in der Subgruppe sind und die Frage jedoch nicht gestellt bekommen haben (Aufgrund ihrer Antwort auf der ersten Ebene), fälschlicherweise als „Missing“ kategorisiert werden. Es ist in den TFL-Programmen möglich, die Variablen der zwei Fragen zu kombinieren, bei Fragen mit mehr als 2 Ebenen ist dies jedoch nicht mehr praktikabel, da die Übersichtlichkeit der Tabelle stark darunter leidet.

Hinzufügen und Anpassen von komplett neuen Tabellenstrukturen sind komplexer als in Tabellen ohne weitreichende Automatisierung. Hierbei müssen alle Makro-Variablen und Schleifen beachtet werden, um z.B. eine reibungslose Stratifizierung zu gewährleisten. Nach der Erstellung steht diese Struktur jedoch global für alle Variablen zur Verfügung.

Während es einfach ist, Änderungen auf globalem Maßstab anzuwenden, macht dieses System es aufwändiger, individuelle Änderungen an einzelnen Tabellen durchzuführen. Das System ist zu empfehlen bei der Produktion von vielen ähnlichen Tabellen. Wenn jede Tabelle individuell angepasst werden soll, ist dies schwieriger. Der Import von Makro-Variablen aus Excel kann aber auch hier von großem Vorteil sein.