

Das Beste aus zwei Welten Aufruf von R-Funktionen mit PROC IML

Peter Beyer
HMS Analytical Software GmbH
Rohrbacher-Str. 16
D-69115 Heidelberg
peter.beyer@analytical-software.de

Zusammenfassung

Dieser Beitrag stellt die neue von SAS bereitgestellte Schnittstelle zur Open Source Software R vor. Die Schnittstelle ist seit SAS 9.22 verfügbar und ist in der SAS Prozedur PROC IML implementiert.

Es wird sowohl auf die Installationsvoraussetzungen als auch auf die Start-Optionen eingegangen. Anschließend wird gezeigt, wie R Code mit SAS ausgeführt werden kann und auf welchem Wege SAS-Datasets exportiert und importiert werden können.

Ebenfalls werden Antworten auf Fragen wie z.B. „gehen SAS Formate auf dem Weg zu R verloren“ oder „ist es möglich, mit R erzeugte Grafiken in SAS-Reports einzubinden“, gegeben.

Abschließend wird noch eine Best-Practice Methode für die Behandlung von Fehlern vorgestellt.

Schlüsselwörter: SAS, R, IML, PROC IML

1 Was ist SAS IML

Die SAS Interactive Matrix Language (IML) ist eine Programmiersprache für die explorative Datenanalyse. Sie implementiert eine breite Palette an Funktionen für die meisten Standard-Matrix-Operationen. Auch lineare algebraische und statistische Funktionen, wie z.B. die Lösung gewöhnlicher Differentialgleichungen oder linearer Systeme sind Teil von SAS IML. Direkter Zugriff auf SAS-Datasets zum Bearbeiten oder Neuerstellen sind gegeben. Zusätzlich können eigene SAS IML-Module entwickelt werden, die sich in Anwendungen integrieren lassen. Darüber hinaus implementiert SAS IML die Schnittstelle zu R, auf die von SAS via PROC IML zugegriffen werden kann.

Dieser Artikel konzentriert sich auf die SAS Prozedur PROC IML und die darin enthaltenen Funktionalitäten, um dadurch Zugriff auf R zu erhalten. Nicht berücksichtigt werden in diesem Artikel Details zu SAS IML oder das SAS IML Studio.

2 Was ist R

Das R-Environment mit der gleichnamigen Programmiersprache R ist ein frei verfügbares Open-Source Software Tool. Spezialgebiete sind statistische Berechnungen und die Erstellung von Grafiken. Die Syntax wie auch die Palette an grundlegenden statistischen Funktionen ist vergleichbar mit SAS IML.

Eine weitere Charakteristik von R ist die große Anzahl an Entwicklern und Anwendern, die sich in sogenannten Communities organisieren und dort Algorithmen zu unterschiedlichen Spezialgebieten kostenlos bereitstellen.

Die Community www.bioconductor.org ist z.B. speziell für Biologen und Bioinformatiker interessant.

3 Warum R aufrufen, wenn man doch auch SAS IML nutzen könnte?

Es stellt sich die berechtigte Frage, warum man PROC IML nutzen sollte, um R-Funktionen aufzurufen, wenn doch R ähnlichen Charakter wie SAS IML besitzt?

Das wohl schwerwiegendste Argument besteht in der fast unzähligen Anzahl an *State of the Art* Algorithmen, die als Zusatzpakete für R frei verfügbar herunterladbar sind. Insbesondere ist R sehr beliebt in Forschungsabteilungen und wird somit dort für die Umsetzung von komplexen Algorithmen und anspruchsvollen Grafiken verwendet. Hinzu kommt, dass viele Fachzeitschriften im Bereich Life Science voraussetzen, dass die mit einer Veröffentlichung in Zusammenhang stehenden Algorithmen ebenfalls öffentlich zugänglich gemacht werden (z. B. Instructions to Authors, *Bioinformatics*: „Software or data must be freely available to non-commercial users”¹).

Ein weiterer Vorteil ergibt sich für Anwender, die von R auf SAS umstellen. In diesem Fall muss der R-Code nicht in SAS-Code überführt werden, sondern kann mittels PROC IML weiterverwendet werden.

Zum Schluss sei noch genannt, dass mit Hilfe dieser Schnittstelle die Validierung von Ergebnissen möglich ist. Da statistische SAS-Prozeduren oft auch in vergleichbarer Weise in R implementiert sind, können Ergebnisse aus beiden Software-Systemen auf Gleichheit untersucht werden.

4 Installationsvoraussetzungen

Um auf die Schnittstelle zu R mittels PROC IML zugreifen zu können, muss mindestens SAS 9.22 (Maintenance-Release 3 für SAS 9.2) installiert sein und eine gültige

¹ http://www.oxfordjournals.org/our_journals/bioinformatics/for_authors/general.html

Lizenz für SAS IML vorhanden sein. SAS IML kann sowohl mit 32-bit als auch mit 64-bit Windows Computern genutzt werden.

Voraussetzung ist, dass sowohl der SAS Workspace Server als auch R auf dem gleichen Computer installiert sind. Ebenfalls bietet sich die Möglichkeit z.B. via Enterprise Guide auf andere SAS-Server zuzugreifen. In diesem Fall muss R auf dem jeweiligen SAS-Server installiert sein.

R wird nicht zusammen mit der SAS-Installation ausgeliefert, sondern muss zusätzlich heruntergeladen werden (<http://cran.r-project.org/>). Laut einer SAS Note (<http://support.sas.com/kb/42/079.html>) unterstützt SAS IML 9.22 keine R-Versionen ab 2.12. Die Version R-2.11.1 wird empfohlen.

Die aktuelle Version SAS 9.3 bringt einige neue Features im Bezug auf PROC IML, siehe Tabelle 1 Hinzu kommt, dass nun auch Linux-Systeme und die 64-bit Version von R unterstützt werden. Laut Dokumentation können alle R-Versionen nach R-2.6.1 genutzt werden.

Tabelle 1: Charakteristiken der Schnittstelle PROC IML zu R.

	SAS 9.22	SAS 9.3
Lizenz für SAS-IML	x	x
32-bit and 64-bit Windows	x	x
32-bit and 64-bit Linux	-	x
32-bit version of R	x	x
64-bit version of R	-	x
Supported R Versions	R-2.6.1 bis R-2.11.1	R-2.6.1 und höher

4.1 SAS mit der RLANG-Option starten

Standardmäßig ist es nicht möglich, von SAS auf R zuzugreifen. Dieses Verhalten lässt sich durch die System Option NORLANG steuern. Um nun den Zugriff auf R zu ermöglichen, startet man SAS mit der RLANG Option (C: \ ... \ sas.exe -RLANG) oder nimmt einen entsprechenden Eintrag in der SAS Config-Datei vor.

Sobald SAS gestartet wurde, empfiehlt es sich zu prüfen, ob R installiert ist und ob der Zugriff von SAS aus möglich ist. Dies kann durch einen PROC OPTIONS Schritt abfragt werden:

```
PROC OPTIONS OPTION = RLANG ;
RUN ;
```

Eine von vier möglichen Meldungen wird in das SAS Logfenster geschrieben und gibt somit Auskunft über die aktuellen Zugriffsmöglichkeiten auf R.

1. NORLANG: Do not support access to R language interfaces
2. ERROR: The RLANG system option must be specified in the SAS configuration file or on the SAS invocation command line to enable the submission of R language statements.
3. WARNING: SAS option RLANG is not supported on this host.
4. RLANG: Support access to R language interfaces

5 Ausführen von R Code

Die mit PROC IML aufgerufene Schnittstelle ist recht komfortabel. Auszuführender R Code wird in die Ausdrücke SUBMIT und ENDSUBMIT eingebettet. Das folgende Beispiel zeigt eine simple Multiplikation. Das Ergebnis wird mit der *paste()*-Funktion ausgegeben.

```
PROC IML;  
  SUBMIT / R;  
    a <- 2  
    paste("a:",a)  
    b <- 3  
    paste("b:",b)  
    c <- a * b  
    paste("c:",c)  
  ENDSUBMIT;  
QUIT;
```

Abbildung 1 zeigt die Ausgabe von R, die direkt in das SAS Outputfenster geschrieben wurde.

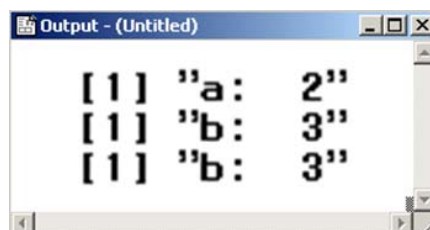


Abbildung 1: Der ins SAS Outputfenster umgeleitete R Output.

6 Integrierte IML-Subroutinen zum Datentransfer nutzen

Ein Datenformat eines Softwaresystems in ein vergleichbares Datenformat eines anderen Softwaresystems zu konvertieren, ist eine der großen Herausforderungen bei der

Entwicklung von Schnittstellen. Gleichwohl ist dies ein wichtiges Merkmal einer leistungsfähigen Schnittstelle.

SAS bietet eine komfortable Möglichkeit an, ein SAS-Dataset als R-Dataframe in R zur Verfügung zu stellen und anschließend wieder als Dataset nach SAS zu importieren. Sowohl der Weg von SAS zu R als auch der rückläufige Weg werden im Folgenden beschrieben.

6.1 Export von SAS-Datasets zu R

Für den Fall, dass man Daten von SAS zu R transferieren möchte, bedeutet dies, ein SAS-Dataset in ein R-Dataframe zu konvertieren. Ein Dataframe in R ist wahrscheinlich der am häufigsten verwendete Objekttyp. Unter einem Dataframe in R versteht man eine Matrix, die aus unterschiedlichen Datentypen aufgebaut sein kann.

Im folgenden Beispiel wird mit Hilfe der IML-Subroutine *ExportDataSetToR* das SAS-Dataset *SasHelp.Class* in ein R-Dataframe mit dem Namen *RDataFrame* konvertiert. Die Subroutine wird mit der IML-Kontrollanweisung *RUN* ausgeführt:

```
PROC IML;
  RUN ExportDataSetToR( "SASHELP.Class" , "RDataFrame" );
  SUBMIT / R;
    RDataFrame[ 1:4 , 1:5 ]
  ENDSUBMIT;
QUIT;
```

Der ausgeführte R Code gibt die Zeilen 1-4 und Spalten 1-5 des Dataframes aus. Diese Ausgabe wird wiederum in das SAS Outputfenster geschrieben, siehe Abbildung 2.

```
      Name Sex Age Height Weight
1 Alfred  M  14   69.0  112.5
2 Alice   F  13   56.5   84.0
3 Barbara F  13   65.3   98.0
4 Carol   F  14   62.8  102.5
```

Abbildung 2: Der ins SAS Outputfenster umgeleitete R Output.

Tabelle 2 soll die Möglichkeiten verdeutlichen, welche Datentypen von SAS zu R exportiert werden können.

Tabelle 2: Von der Schnittstelle PROC IML unterstützte Datenkonvertierungen von SAS zu R.

Subroutine	SAS-Quelle	R-Zielobjekt
ExportDataSetToR	SAS-Datasets	R-Dataframe
ExportMatrixToR	SAS/IML matrix	R-Matrix

Wie bereits erwähnt, ist die Schnittstelle zu R in SAS IML implementiert. Deswegen ist es prinzipiell ebenfalls möglich, IML Matrizen in R Matrizen zu konvertieren.

6.2 Import von R-Datframes als SAS-Datasets

Auch der umgekehrte Weg der Datenübertragung von R nach SAS ist als IML- Subroutine implementiert. Die Subroutine *ImportDataSetFromR* importiert ein R-Dataframe als SAS-Dataset in eine angegebene SAS-Bibliothek.

Im folgenden Beispiel wird das Dataframe Iris (Länge und Breite von Kelch- und Blütenblättern von drei unterschiedlichen Spezies der Blumenart Iris) verwendet. Es sollen nur Blumen der Spezies "Setosa" in einem neuen Dataframe namens "iris.setosa" gespeichert werden. Anschließend wird der neue Dataframe in die SAS-Bibliothek WORK geschrieben.

Die SAS Prozedur PROC MEANS wird abschließend auf die importierte SAS-Datei angewendet:

```
PROC IML;  
  SUBMIT / R;  
    iris.setosa = iris[(iris$Species=="setosa"),]  
  ENDSUBMIT;  
RUN ImportDataSetFromR("WORK.IrisSetosa"  
  , "iris.setosa");  
QUIT;  
PROC MEANS data = WORK.IrisSetosa N SUM MEAN;  
RUN;
```

Ein Blick in den SAS Output, Abbildung 3 der Prozedur PROC MEANS bestätigt, dass erfolgreich auf das R-Dataframe in der SAS Bibliothek WORK zugegriffen werden konnte.

The MEANS Procedure			
Variable	N	Sum	Mean
Sepal_Length	50	250.3000000	5.0060000
Sepal_Width	50	171.4000000	3.4280000
Petal_Length	50	73.1000000	1.4620000
Petal_Width	50	12.3000000	0.2460000

Abbildung 3: SAS Output von PROC MEANS.

Tabelle 3 verdeutlicht die Möglichkeiten, welche Datentypen von R nach SAS importiert werden können.

Tabelle 3: Von der Schnittstelle PROC IML unterstützte Datenkonvertierungen von R zu SAS.

Subroutine	R-Quelle	SAS-Zielobjekt
ImportDataSetFromR	R-Dataframe	SAS-Datasets
ImportMatrixFromR	R-Matrix	SAS/IML Matrix

Auch hier sei noch mal erwähnt, dass die Schnittstelle zu R in SAS IML implementiert ist und aus diesem Grund auch R-Matrizen importiert werden können. Allerdings sind diese dann nur innerhalb der PROC IML Prozedur verfügbar und können nicht in einer SAS Bibliothek gespeichert werden.

7 Umwandlung von Formaten

SAS und R implementieren nicht nur unterschiedliche Datentypen, sondern unterscheiden sich auch bei den Formaten. R besitzt kein vergleichbares Konzept zu den SAS-Formaten. Vielmehr handelt es sich bei R um Klassen bzw. Objekte. Die Schnittstelle zu R unterstützt lediglich drei SAS-Formate, siehe Tabelle 4. Diese sind *Date*, *Datetime* und *Time*. Die Datentypen *Numeric* und *Character* werden in gleichnamige R-Klassen konvertiert.

Tabelle 4: Angabe der SAS-Formate, die in eine entsprechende R-Klasse konvertiert werden.

SAS-Format	R-Klasse
Date (z.B. Datew.d)	Date
Datetime (z.B. Datetimew.d)	Klassen, die von Posixtct erben
Time (z.B. Timew.d)	Klassen, die von Posixtlt erben
Numeric	Numeric
Character	Character

Die R Klassen "Posixtct" „Posixtlt“ repräsentieren dabei das Datum bzw. die Zeit. Alle anderen Formate (auch benutzerdefinierte Formate) werden nicht umgewandelt.

Der umgekehrte Weg, R-Klassen in SAS-Formate umzuwandeln, wird ebenfalls unterstützt, siehe Tabelle 5. Dies setzt voraus, dass die Daten in einem R-Dataframe vorliegen.

Tabelle 5: Angabe der R-Klassen, die in ein entsprechendes SAS-Format konvertiert werden.

R	SAS
Date	Date9.
Posixt	Time19.
Numeric	Numeric
Character	Character
Complex	Fehlermeldung
Raw	Fehlermeldung

Werte der Klasse *Numeric* oder *Character* werden in die entsprechenden SAS-Datentypen konvertiert. Werte der Klasse *Complex* oder *Raw*, die ebenfalls gültige Werte eines Dataframes darstellen, können nicht konvertiert werden.

8 Import von in R erstellten Graphiken nach SAS

Mit der Prozedur PROC IML können R-Grafiken erzeugt werden. Allerdings funktioniert dies nur, wenn man auf eine lokale SAS-Installation zugreift. Als Ergebnis wird man eine Grafik erhalten, die dann in einem R-Fenster auf dem gleichen Computer angezeigt wird. Grafiken, die mit einer R-Installation auf einem Server erzeugt wurden, können auf diese Weise nicht angezeigt werden.

Wie man eine R-Graphik in einen SAS-Report einbettet und zwar unabhängig davon, ob man lokal oder auf einem Server arbeitet, wird in dem folgenden Verfahren beschrieben. Dazu wird eine R-Graphik mit Hilfe der R-Funktion *png()* in einem Verzeichnis abgespeichert und anschließend diese Graphik mit ODS-Befehlen in einen SAS-Report integriert:

```
PROC IML;
  SUBMIT / R;
    png(file="c:/tmp/swiss.png",width=600,height=400)
    source("http://addictedtor.free.fr/graphiques/sources/source_2.R")
    dev.off()
  ENDSUBMIT;
QUIT;
```

Mit Hilfe von *png()* wird eine Graphikausgabe im png-Format geöffnet. Im Anschluss wird mit *source()* ein öffentlich verfügbares R-Skript geladen, welches den Code für die Erzeugung einer bestimmten Graphik enthält. Die Funktion *dev.off()* schließt die aktuelle Graphikausgabe wieder.

Um nun die mit R erzeugte Graphik in einen SAS-Report zu integrieren, werden folgende ODS-Statements abgesetzt:

```

ODS ESCAPECHAR='^';
ODS HTML FILE='c:\tmp\swiss.html' STYLE=minimal
GPATH='c:\temp\' GTITLE GFOOTNOTE;

DATA _NULL_;
  FILE PRINT;
  PUT "<IMG SRC='c:\tmp\swiss.png' BORDER='0'>";
RUN;
ODS _ALL_ CLOSE;

```

Der erzeugte SAS-HTML Report, Abbildung 4, enthält nun die mit R erzeugte Graphik. Diese Graphik hat 2004 die „R Homepage graphics competition“ gewonnen und befindet sich seitdem auf der Startseite der R-Homepage (<http://www.r-project.org>).

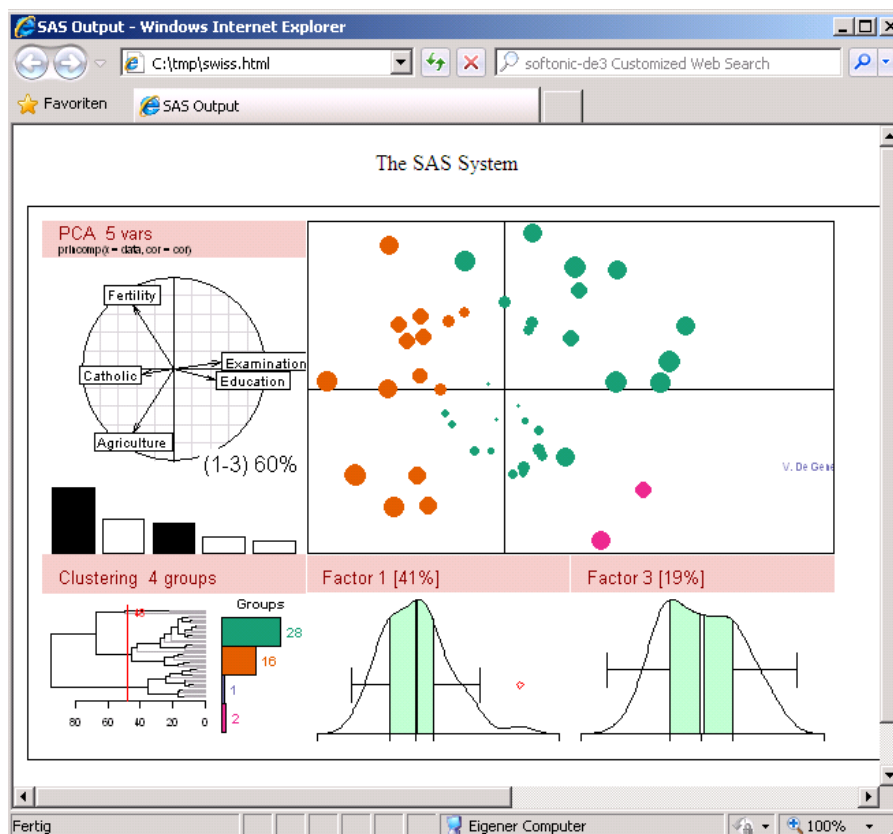


Abbildung 4: Eine mit R erzeugte Graphik, die in einen SAS-Report eingebettet wurde. Die Graphik zeigt unterschiedliche Abbildungen der deskriptiven Statistik. Die Grafik findet sich auf der offiziellen R-Homepage wieder.

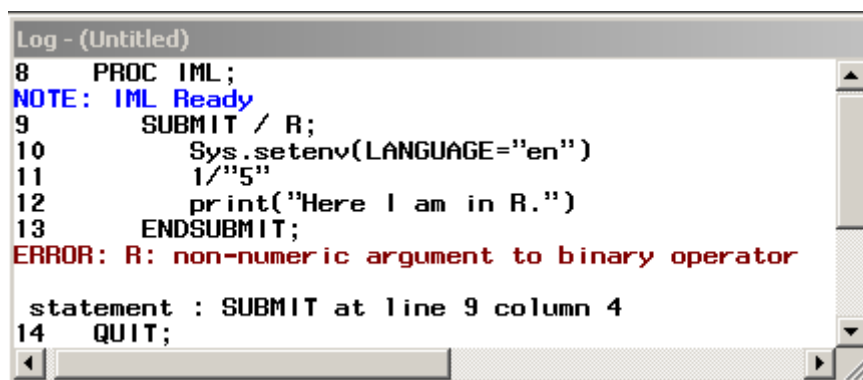
9 Fehlerbehandlung

Fehlerbehandlung wird u.a. dazu eingesetzt, mögliche Fehler abzufangen, die ohne eine solche Vorgehensweise zum unkontrollierten Abbruch des Programms führen würden.

SAS und R bieten unterschiedliche Mechanismen für die Behandlung von Fehlern an. Zwei Integrationsmöglichkeiten dieser beiden Mechanismen werden im Folgenden vorgestellt.

9.1 Globale Makrovariable & SysErrorText auslesen

Kommt es auf der Seite von R zu einer Fehlermeldung, so wird diese zurück an SAS geleitet und dort als SAS-Error ins Logfenster geschrieben, siehe Abbildung 5.



```
Log - (Untitled)
8 PROC IML;
NOTE: IML Ready
9 SUBMIT / R;
10 Sys.setenv(LANGUAGE="en")
11 1/"5"
12 print("Here I am in R.")
13 ENDSUBMIT;
ERROR: R: non-numeric argument to binary operator

statement : SUBMIT at line 9 column 4
14 QUIT;
```

Abbildung 5: Eine Fehlermeldung von R, die im SAS Logfenster ausgegeben wird.

Da in SAS die letzte Fehlermeldung in der globalen Makrovariablen &SysErrorText abgespeichert wird, kann man diese Fehlermeldung auslesen und das eigene Programm entsprechend reagieren lassen:

```
DATA _NULL_;
  IF ("&SYSERRORTEXT." NE "") THEN DO;
    PUT "Error in R: &SYSERRORTEXT.";
  END;
RUN;
```

9.2 TryCatch-Block in R nutzen

R besitzt einen Try-Catch-Mechanismus zur Fehlerbehandlung. Dieser erlaubt es einen potenziell fehlerverursachenden Codeblock auszuführen und beim tatsächlichen Eintreten des Fehlers, kontrolliert auf diesen zu reagieren. Dadurch wird ein unkontrolliertes Beenden eines Programms verhindert.

Im folgenden Beispiel wird eine Division durch eine alphanumerische Variable (1 / "5") in R vorgenommen. Diese Division ist in den ersten Teil des Try-Catch-Blocks eingebettet (zu erkennen an { }). Da die Division durch eine alphanumerische Variable nicht

erlaubt ist, wird nach dem Eintreten des Fehlers der Teil ausgeführt, der für den Fehlerfall programmiert wurde. In dem Codebeispiel wird die letzte von R erzeugte Fehlermeldung ausgelesen (*geterrmessage()*) und in einem Dataframe gespeichert:

```
PROC IML;
  SUBMIT / R;
    Sys.setenv(LANGUAGE="en")
    errMess = ""
    tryCatch({
      x = 1/"x";
    },
    error = function(ex) {
      errMess = geterrmessage()
      errMess <- as.data.frame(errMess)
    }
  )
ENDSUBMIT;
RUN ImportDataSetFromR("WORK.ErrMessFromR",
"errMess");
QUIT;
```

Innerhalb der PROC IML Prozedur wird nun mit Hilfe der IML-Subroutine *ImportDataSetFromR* das erzeugte Dataframe mit der Fehlermeldung nach SAS importiert. Zurück in SAS kann man nun diese Fehlermeldung auslesen und z.B. in das Log-Fenster schreiben oder, wenn gewollt, das Programm kontrolliert beenden lassen.

```
DATA _NULL_;
  SET WORK.ErrMessFromR END=LAST;
  IF LAST THEN DO;
    CALL SYMPUT('errMessInSAS', errMess);
  END;
RUN;
%PUT Error-Message from R: &errMessInSAS.;
```

10 X-Command als Zugriffsmöglichkeit auf R

Ganz ohne SAS IML funktioniert der Zugriff auf R natürlich auch. Unter Verwendung des X-Commands, ist es möglich Betriebssystembefehle abzusetzen. Liegt nun ein R-Skript in einem bekannten Verzeichnis, so kann dieses wie folgt ausgeführt werden:

```
OPTIONS XWAIT XSYNC;
X "C:\Programme\R\R-2.14.1\bin\r.exe --no-save --quit;
  <"c:\temp\program.r"
  >"c:\temp\program.log" ";
```

Zunächst wird der Pfad zu der R-Installation angegeben, gefolgt von dem Befehl, keinen sogenannten Workspace (Gesamtheit aller in R, während dieser Session, erzeugte Objekte) zu speichern (*no-save*). Anschließend wird der Befehl abgesetzt, die geöffnete R-Session nach Abschluss des Skripts wieder zu schließen (*quit*). Die zweite Zeile verweist auf das R-Skript und die dritte Zeile gibt an, in welcher Datei das Log von R gespeichert werden soll.

Ruft man ein R-Skript mit Hilfe eines X-Commands auf, so verliert man allerdings die in diesem Artikel genannten Vorzüge der in PROC IML realisierten Schnittstelle zu R (z.B. SAS Dataset Export/Import).

11 Einschränkungen

Die von SAS als PROC IML-Prozedur bereitgestellte Schnittstelle zu R stellt, wie bereits erwähnt, einen leistungsfähigen Mechanismus zur Verfügung, um SAS Datasets zu exportieren und zu importieren. Dabei muss man sich allerdings der Einschränkung bewusst sein, dass R Daten im Arbeitsspeicher vorhält. Die Größe der Datasets [kB], die in R bearbeitet werden können, steht also in Abhängigkeit zu der Größe des Arbeitsspeichers des verwendeten Systems.

Greift man auf eines der unzähligen frei verfügbaren Zusatzpakete von R zu, so gilt es zu prüfen, ob das erwartete Ergebnis erzielt werden kann. Denn nicht nur R „kommt ohne jegliche Garantie“ sondern gerade auch die Zusatzpakete. An dieser Stelle sei aber noch einmal darauf hingewiesen, dass die Vorteile einer Open Source Software ebenfalls für R gültig sind. Ein wichtiger Aspekt dabei ist, dass der Code frei verfügbar ist und dadurch von einer großen Entwicklergemeinschaft gesichtet wird. Bugs, so die Annahme, sollten somit frühzeitig erkannt und gefixt werden.

12 Zusammenfassung

SAS stellt mit der Prozedur PROC IML eine leistungsstarke Schnittstelle zur Open-Source-Software R zu Verfügung. Dadurch bietet sich für SAS-Anwender die Möglichkeit, auf die in R implementierten *State of the Art* Algorithmen zuzugreifen.

Die SUBMIT- und ENDSUBMIT-Anweisungen der PROC IML Prozedur erlauben auf einfache Weise R Code auszuführen. Auch der Anspruch SAS-Datasets nach R zu exportieren, dort Berechnungen durchführen zu können und SAS-Datasets anschließend wieder importieren zu können, wird mit dieser Schnittstelle erfüllt.

Weiterhin besteht die Möglichkeit, SAS-Formate zu exportieren und wieder zu importieren. Allerdings beschränkt sich dies auf Datum- und Zeitformate.

Eine große Stärke von R ist die Erstellung von Graphiken. Hier wurde gezeigt, wie über die Schnittstelle eine Graphik mit R erzeugt wird und wie diese in einem weiteren Schritt mit Hilfe von ODS-Befehlen in einen SAS-Report integriert werden kann.

Demonstriert wurde außerdem, wie sich die unterschiedlichen Methoden der Fehlerbehandlung von SAS und R miteinander integrieren lassen. Fehler, die auf Seiten von R verursacht wurden, konnten mit SAS abgefangen werden, und eine kontrollierte Reaktion darauf konnte folgen.

Literatur

- [1] SAS-Version 9.22:
http://support.sas.com/documentation/cdl/en/imlug/63541/HTML/default/viewer.htm#r_toc.htm
- [2] SAS-Version 9.3:
http://support.sas.com/documentation/cdl/en/imlug/64248/HTML/default/viewer.htm#r_toc.htm