SAS-MAKROS ZUR ENTWICKLUNG UND VALIDIERUNG VON PROGNOSEMODELLEN AUF BASIS DER LOGISTISCHEN REGRESSION

GETESTET IN SAS VERSION 9

(AUSZUG AUS EINEM PRAKTIKUMSBERICHT)

Praktikumsstelle:	
Institution:	Universität Ulm
Abteilung:	Biometrie und Medizinische Dokumentation
	Schwabstraße 13
	89075 Ulm

Praktikumsleiter:	
Name:	PD Dr. Rainer Muche
Telefon:	0731 – 50 26903

Zeitraum:	
Beginn:	01.06.2005
Ende:	29.08.2005

Praktikant:	
Name:	Felix Ruthenberg
Studiengang:	Medizinische Dokumentation und Informatik, FH Ulm

1 PROGNOSEMODELLE MIT HILFE VON SAS-MAKROS

In der Abteilung Biometrie entstand im Rahmen einer Habilitationsschrift im Jahre 2004 [1] eine Auswertungsstrategie, mit deren Hilfe es möglich ist, komplexe Prognosen, die auf logistische Regressionsmodelle aufbauen, schnell und einfach mit SAS zu erstellen.

Dafür wurden mehrere SAS-Makros entwickelt, die alle zusammen genommen ein umfangreiches Paket darstellen, das von der deskriptiven Analyse der Variablen bis hin zur kompletten Modellierung und Validierung des Prognosemodells alle wichtigen Punkte abdeckt.

Die Makros wurden an einem Beispieldatensatz aus dem Reha-Forschungs-Verbund Ulm [3] entwickelt und getestet, der auch mir für meine Arbeit zu Grunde lag. Dieser Datensatz [4] enthält Informationen über Reha-Aufenthalte von Patienten, wie zum Beispiel Dauer des Aufenthaltes und Art der Reha. Die erfasste Zielgröße ist das Beziehen einer Erwerbsunfähigkeitsrente bis zu 2 Jahre nach dem stationären Reha-Aufenthalt. Mit Hilfe der Makros soll ein Prognosemodell erstellt werden, das zuverlässig Angaben über einen neuen Reha-Patienten machen kann um durch intensivere Reha oder Nachbeobachtungen die Zeit bis zum Erhalt der Rente zu verlängern oder diese gar ganz zu umgehen.

In einem Praktikum, dass Bestandteil des Studienganges "medizinische Dokumentation und Informatik" der Fachhochschule Ulm ist, wurde mir die Aufgabe gestellt diese Makros, die in SAS Version 8 erstellt wurden, in der SAS Version 9 zu testen. Eventuell auftretende Probleme, die durch den Versionswechsel verursacht werden sollten gefunden und behoben werden, damit die Makros auch in der nächsten SAS-Generation benutzt werden können. Im Vorfeld war bereits klar, dass einige Makros Probleme verursachen könnten, da z. B. bekannt war, dass die PROC LOGISTIC in SAS Version 9 geändert wurde. Bei den "einfacheren" Makros wie zum Beispiel dem Makro zur Deskription des Datensatzes wurden hingegen keine Probleme erwartet.

Das komplette Paket besteht aus 102 Makro-Dateien, davon enthalten 14 Dateien die eigentlichen Makroaufrufe (Hauptmakros). In 65 Dateien sind Fehlermeldungen gespeichert, die verbleibenden 23 Makros sind Hilfsmakros, die von anderen Makros aufgerufen werden. Die Hilfsmakros kommen teilweise von externen Stellen, anderen Universitäten etc.

Während meines dreimonatigen Praktikums habe ich mich hauptsächlich auf die Hauptmakros konzentriert. Diese werde ich im Folgenden hier vorstellen, kurz die Funktionen erläutern und dann aufgetretene Probleme mit entsprechenden Lösungsvorschlägen darstellen.

Zum Testen der Makros habe ich denselben Datensatz benutzt, an dem auch die Makros entwickelt wurden. Beim ersten Erproben wurde die Situation dargestellt, die auch in der Habilitationsschrift [1] angegeben wird. Das bedeutet, dass das Makro mit allen dort beschriebenen Einstellungen der Parameter gestartet wurde. Verglichen wurde der Output des Makros mit dem Output, der in der Habilitationsschrift abgedruckt ist, bzw. der in Version 8 als Vergleich erzeugt wurde. Außerdem wurden die Meldungen im LOG-Fenster nach Warnungen und Fehler untersucht und unter Umständen erzeugten Dateien auf ihre Korrektheit überprüft.

Aufgetretene Fehler wurden untersucht um Ursachen herauszufinden und zu beheben. Die Vorgehensweise dabei kann sehr unterschiedlich sein, hilfreich ist aber in den meisten Fällen die Isolation von einzelnen Befehlen und Aufrufen (z. B. PROC LOGISTIC) um an einfachen Programmen Änderungen schnell feststellen zu können.

Da die Makros in der Praxis aber nicht nur mit ihrer Standardeinstellung verwendet werden, ist es notwendig auch alle weiteren Parameter, mit denen das Makro aufgerufen werden kann, zu testen. Dazu habe ich jeden einzelnen Parameter verstellt und untersucht, ob das ausgegebene Ergebnis den Erwartungen entsprach. Wenn das nicht der Fall war hab ich auch hier nach den Problemen gesucht und diese (wenn möglich bzw. notwendig) auch behoben.

Bei den Testläufen wurde auch die Laufzeit der Makros protokolliert um evtl. Veränderungen in der Rechengeschwindigkeit festzustellen. Getestet wurde auf einem PC mit Intel Celeron Prozessor (895 MHz), 265 MB RAM und Windows XP als Betriebssystem. Es gab bei keinem Makro Probleme mit den Laufzeiten, der Vollständigkeit halber sind diese hier aber trotzdem aufgeführt.

1.1 VORBEREITUNGEN ZUR REGRESSIONSANALYSE

Bevor auf einen Datensatz eine Regressionsanalyse angewandt wird ist es sinnvoll diesen nach verschiedenen Gesichtspunkten zu untersuchen um inhaltliche Fehler zu vermeiden und mögliche Über- oder Unterschätzungen der Regressionsparameter zu verhindern. Die folgenden Makros beinhalten also Auswertungsstrategien, um den vorhanden Datensatz auf die multivariate Regressionsanalyse vorzubereiten.

1.1.1 PM_DESRIPTION.MAC.SAS

Bevor überhaupt mit irgendeiner statistischen Auswertung begonnen wird empfiehlt es sich, dass man sich Klarheit über die vorhandene Datenstruktur verschafft. Dazu dient dieses Makro, das (wie der Name schon sagt) eine deskriptive Darstellung des Datensatzes erzeugt.

Das Makro erstellt Kreuztabellen, die die Häufigkeiten der kategoriellen Merkmale gruppiert nach der Zielgröße enthalten. Für die Darstellung der stetigen Merkmale werden mit Hilfe der PROC UNIVARIATE Stichprobenumfang, Mittelwert und Standardabweichung sowie die Perzentile in 20er-Schritten und der Median ausgegeben. Klassiert nach den Perzentilen und gruppiert nach der Zielgröße werden zusätzlich Kreuztabellen erstellt. Für alle stetigen Variablen kann man sich zudem noch Histogramme generieren lassen, die Aussagen zur Verteilung zulassen.

Die Ausgabe des Makros liefert daneben auch Informationen über fehlende Werte im Datensatz, gruppiert nach stetigen und kategoriellen Merkmalen und der Zielgröße, die für das Makropaket immer dichotom sein muss (in diesem Fall Erwerbsunfähigkeitsrente ja oder nein). Außerdem kann man sich anzeigen lassen welche Beobachtung die meisten fehlenden Werte hat. Diese Berechnung ist sehr rechen- und damit zeitintensiv, daher werden per default nur die fünf Beobachtungen mit der maximalen Anzahl an fehlenden Werten ausgegeben werden.

Bei diesem Makro sind keine Auffälligkeiten gegenüber der Version 8 aufgetreten. Die Laufzeiten sind vergleichbar mit denen in Version 8. Ohne Ausgabe der most_extreme-Werte zwischen 11 und 17 Sekunden, mit Ausgabe zwischen knapp 8 und 8 1/2 Minuten.

1.1.2 PM_MULTICOLLIN.MAC.SAS

Multikollinearität untersucht die Zusammenhänge zwischen einzelnen Variablen um Zusammenhänge im Datensatz zu erkennen und unter Umständen Variablen, die durch andere beschrieben werden, aus dem Modell nehmen zu können. Wichtige Kenngrößen die dazu ermittelt werden sind zum Beispiel die Korrelationskoeffizienten nach Spearman. Das Makro nutzt in der Hauptsache die SAS-Prozedur PROC REG um diese Werte zu berechnen. Zur Übersichtlichkeit werden nur Variablen angezeigt, die in diesen Werten über einer (vom Benutzer) festgelegten Schranke liegen. Im Reha-Datensatz wurde zum Beispiel eine starke Korrelation zwischen der Anzahl der bewilligten Tage und der Heilverfahrensart festgestellt.

Auch bei diesem Makro sind keine Änderungen zu verzeichnen gewesen. Die Laufzeit liegt mit Werten zwischen 13 und 16 Sekunden im erwarteten Bereich.

1.1.3 PM_MISSING.MAC.SAS

Dieses Makro ersetzt fehlende Werte im Datensatz. Dazu kann der Benutzer verschiedene Möglichkeiten einstellen. Die Single Imputation ersetzt fehlende Werte stetiger Variablen zum Beispiel durch den Mittelwert oder den Median der anderen vorhandenen Werte innerhalb einer Variablen. Fehlende Werte kategorieller Variablen werden durch eine Missing-Kategorie, die vom Benutzer angegeben werden kann, ersetzt. (Bei der Multiple Imputation werden mehrere Datensätze erstellt, die fehlende Werte über eine zu Grunde gelegte Verteilung (zum Beispiel Normalverteilung) ersetzt. So entstehen mehrere verschiedene Datensätze, die getrennt voneinander ausgewertet werden können um dann am Schluss diese Ergebnisse wieder zusammenzuführen. Die SAS-Prozedur PROC MI bietet mit seinen Statements die erforderlichen Ersetzungsmethoden.

Die Ausgabe dieses Makros besteht außer aus den Datensätzen auch aus einer Übersicht über die Verteilung der fehlenden Werte und einer Zusammenfassung über die Art der Ersetzung(en).

Mit den in der Habilitationsschrift angegebenen Parametern wurde der Datensatz korrekt ausgewertet. Die Überprüfung der einzelnen Parameter jedoch ergab, dass die multiple Imputation auch ausgeführt wird, wenn der entsprechende Parameter dies verhindern sollte (*imputation_art=0*). Allerdings ist das kein Problem, das auf den Versionswechsel zurückzuführen ist und wurde von anderer Stelle bereits beseitigt.

Die Laufzeiten liegen bei minimal 52 Sekunden wenn nur 2 Datensätze bei der Multiple Imputation erzeugt werden sollen und ca. 1 Minute 40 Sekunden bei in der Praxis üblichen Parameterangaben.

1.1.4 PM_MI_ANALYZE.MAC.SAS

Die Datensätze, die über die multiple Imputation des Makros PM_MISSING.MAC.SAS erzeugt wurden, können mit diesem Makro ausgewertet werden. Zuvor muss jedoch für jeden dieser Datensätze mit Hilfe der Makros PM_LOGREG.MAC.SAS (siehe Kapitel 2.2.2) sowie PM_ROC.MAC.SAS (siehe Kapitel 2.3.2) ein Regressionsmodell entworfen werden. Diese Modelle lassen sich mit dem PM_MI_ANALYZE.MAC.SAS-Makro zusammenfassend darstellen. In SAS gibt es dazu die Prozedur PROC MIANALYZE.

Das Makro hat in Version 9 keine Auffälligkeiten gezeigt und hat eine Laufzeit von ca. 20 Sekunden bei 5 auszuwertenden Datensätzen.

1.1.5 PM_INFLUENCE.MAC.SAS

Dieses Makro ermittelt die einflussreichste Beobachtung im Datensatz. Diese Beobachtungen können unter Umständen durch ihre extreme Abweichung zu anderen Beobachtungen großen Einfluss auf das zu ermittelnde Modell haben, und dieses verzerren. Daher kann es sinnvoll sein diese Beobachtungen bei der weiteren Modellbetrachtung nicht weiter zu berücksichtigen. Die Ermittlung solcher Beobachtungen erfolgt durch Methoden der Ausreißer- und / oder Residualanalyse. Zusätzlich können die Ergebnisse graphisch ausgegeben werden. Dieses Makro benutzt zur Berechnung seiner Ausgabe die Methoden der PROC LOGISTIC mit dem Parameter 'difchisq'. Das steht für die Veränderung der Pearson-Chi-Quadrat-Statistik bei Entfernung einer Beobachtung.

Die Ausführung dieses Makros mit den verschiedenen Testparametern ergab keine Hinweise auf Versionskonflikte. Die Laufzeiten lagen mit 14 Sekunden im Mittel und maximal 34 Sekunden bei Erzeugen der Grafiken im normalen Bereich.

1.2 REGRESSIONSANALYSE-MAKROS

Nachdem der Datensatz inhaltlich so weit aufgeschlossen ist, dass fehlenden Werte ersetzt, unbedeutende Variablen und einflussreiche Beobachtungen erkannt bzw. eliminiert wurden, kann mit der Berechnung des eigentlichen Regressionsmodells begonnen werden. Dazu bietet das Paket die folgenden Makros an.

1.2.1 PM_UNI_LOGREG.MAC.SAS

Bei der univariaten logistischen Regression wird der Einfluss jeweils einer Variablen auf die Zielgröße untersucht. Mit diesem Vorgehen werden zwei Ziele verfolgt: Einerseits können die Voraussetzungen für den Einsatz einer Variablen untersucht werden, andererseits werden so Variablen, die nur geringen Einfluss auf die Zielgröße haben, gefunden und müssen im Modell nicht weiter berücksichtigt werden. Das hat zum Vorteil, dass sich die Variablenanzahl im Modell reduziert, was auch nach Vorauswahl der inhaltlich relevanten Daten nicht immer optimal der Fall ist.

In diesem Makro wird die Prozedur PROC LOGISTIC benutzt, die in der SAS Version 9 einige Änderungen erfahren hat. Dadurch müssen im Makro Veränderungen zur Anpassung auf Version 9 durchgeführt werden. Das Laufzeitverhalten liegt mit knapp anderthalb Minuten im Durchschnitt im normalen Bereich.

Fehlerbeschreibung TYPEIII:

Bei der ersten Ausführung des Makros bemerkt man sofort, dass die kategoriellen Variablen nicht korrekt ausgewertet werden, dies zeigt der Ausschnitt aus dem entsprechenden Output:

```
Univariate Logistische Regression. Zielgröße eu_rente . Ohne missing values

              Prob
variable     ChiSq

 alter      <.0001
 bmi        0.0135
```

(hier sollten die kategoriellen Variablen aufgeführt sein)

```
Univariate Logistische Regression. Zielgröße eu_rente . Mit missing values

              Prob
variable     ChiSq          _0          _1

 alter      <.0001         664         120
 bmi        0.0015         710         130
               .           710         130
               .           710         130
               .           710         130
               .           710         130
               .           710         130
               .           684         124
               .           706         129
               .           660         119
               .           710         130
               .           710         130
               .           710         130
               .           710         130
               .           710         130
               .           710         130
```

(hier sind keine Namen für die kategoriellen Variablen aufgelistet und fehlende Werte für die entsprechenden p-Werte)

Bei einem Blick ins Log-Fenster fällt sofort diese Fehlermeldung auf:

```
WARNING: Output 'TypeIII' was not created.  Make sure that the output objectname,
         label, or path is spelled correctly.  Also, verify that the appropriate
         procedure options are used to produce the requested output object.  For
         example, verify that the NOPRINT option is not used.
```

Das liegt daran, dass der Name der ODS-Tabelle zur Ausgabe der p-Werte in den TYPEIII-Statistiken geändert wurde (TYPE 3 statt TYPEIII).

Fehlerbehebung allgemein:

Das bedeutet für den ODS-Aufruf, dass statt wie bisher mit

ODS OUTPUT TYPEIII

nun

ODS OUTPUT TYPE3

aufgerufen werden muss.

Fehlerbehebung im Makro:

Gesucht wurde nach Vorkommen von:

ODS OUTPUT TypeIII

In Zeile 1278 und Zeile 1589 wurde ODS OUTPUT TypeIII durch ODS OUTPUT Type3 ersetzt.

Allerdings wurde nach dieser Maßnahme der Datensatz zwar erstellt, der Fehler in Makro-Output war aber noch nicht behoben. Beim Ansehen des Datensatzes wird man auf eine zweite Änderung in der Ausgabe der TYPEIII-Tabellen aufmerksam:

Fehlerbeschreibung Variable / Effect:

Bisher bestand der ODS-Output des TypeIII-Moduls aus den Variablen

'Variable', 'DF', 'Wald Chi-square' und 'Pr > Chi-Square'

	Variable	DF	Wald Chi-square	Pr > Chi-Square
1	geschlecht	1	2.0984	0.1475

Ab Version 9 wird die Variable 'Variable' durch eine neue Variable 'Effect' ersetzt

	Effect	DF	Wald Chi-square	Pr > Chi-Square
1	geschlecht	1	2.0984	0.1475

Fehlerbehebung allgemein:

Die Variable 'Effect' muss in 'Variable' umbenannt werden. Dafür bietet sich die Data Set Option *Rename=* an.

Fehlerbehebung im Makro:

Es muss nach der Stelle gesucht werden, in der die TYPEIII-Datensätze erstellt werden. Am sinnvollsten sucht man also nach dem String.

```
ODS OUTPUT TypeIII=TypeIII&ip;
```

Diesen String gibt es zweimal im Makro, von Interesse ist allerdings nur der Teil ab Zeile 1589, denn nur dieser Datensatz wird mit den relevanten Variablen weiterverarbeitet. Die Umbenennung der Variablen muss vor der Weiterverarbeitung aber erst nach der endgültigen Erstellung des Datensatzes erfolgen. Ab Zeile 1661 steht also nun folgender Code:

```
/*Umbenennung der neuen Variablen Effect in Variable*/

DATA TypeIII&ip (RENAME=(Effect=Variable));
      SET TypeIII&ip;
RUN;
```

Der geänderte Datensatz hat dann wieder folgenden Inhalt:

	Variable	DF	Wald Chi-square	Pr > Chi-Square
1	geschlecht	1	2.0984	0.1475

und kann wie bisher weiter verarbeitet werden.

1.2.2 PM_LOGREG.MAC.SAS

Da hier das eigentliche Regressionsmodell entwickelt wird ist dieses Makro das wichtigste im ganzen Paket. Die eigentlichen Regressionsberechnungen (über die SAS-Prozedur PROC LOGISTIC) finden im Untermakro LOGREG2.MAC.SAS statt. Deswegen bezog sich meine Fehlersuche hauptsächlich auf dieses Untermakro. Nach den Erfahrungen mit dem PM_UNI_LOGERG.MAC.SAS-Makro war davon auszugehen, dass ähnliche Probleme auch hier auftreten würden. Dass dies tatsächlich der Fall war hat sich dadurch gezeigt, dass die unter *out_pred=*, *out_roc=*, *out_est=* und *out_mi=* angegeben Datensätze nicht erstellt wurden. Deswegen wurden vor dem eigentlichen Testen gezielt die Fehler durch die TYPEIII-Änderungen behoben.

Fehlerbeschreibung TYPEIII:

Folgende Fehlermeldung kommt beim Aufruf des Makros PM_LOGREG.MAC.SAS:

```
WARNING: Output 'TypeIII' was not created.  Make sure that the output object name,
         label, or path is spelled correctly.  Also, verify that the appropriate
         procedure options are used to produce the requested output object.  For
        example, verify that the NOPRINT option is not used.
```

Das liegt daran, dass der Name der ODS-Tabelle zur Ausgabe der p-Werte in den TYPEIII-Statistiken geändert wurde (TYPE 3 statt TYPEIII). Siehe auch Kapitel 2.2.1 univariate logistische Regression.

Fehlerbehebung im Makro LOGREG2.MAC.SAS:

Gesucht wurde nach Vorkommen von ODS OUTPUT TypeIII.

In Zeile 348 und Zeile 850 wurde ODS OUTPUT TypeIII durch ODS OUTPUT Type3 ersetzt.

Fehlerbeschreibung Variable / Effect:

Bisher bestand der ODS-Output des TypeIII-Moduls aus den Variablen

'Variable', 'DF', 'Wald Chi-square' und 'Pr > Chi-Square'

	Variable	DF	Wald Chi-square	Pr > Chi-Square
1	geschlecht	1	2.0984	0.1475

Ab Version 9 wird die Variable 'Variable' durch eine neue Variable 'Effect' ersetzt

	Effect	DF	Wald Chi-square	Pr > Chi-Square
1	geschlecht	1	2.0984	0.1475

Fehlerbehebung im Makro:

Es muss nach der Stelle gesucht werden, in der die TYPEIII-Datensätze erstellt werden. Am sinnvollsten sucht man also nach dem String

ODS OUTPUT TypeIII=

Diesen String gibt es zweimal im Makro LOGREG2.MAC.SAS, nämlich in Zeile 348 und 850.

Die Umbenennung der Variablen muss vor der Weiterverarbeitung, aber erst nach schließen des ODS-Outputs eingefügt werden. Ab Zeile 554 und 1010 steht also nun folgender Code:

Ergänzung in Zeile 554:

```
/*Umbenennung der neuen Variablen Effect in Variable*/

DATA TypeIII (RENAME=(Effect=Variable));
      SET TypeIII;
RUN;
```

Auch in Zeile 1010 musste diese Änderung durchgeführt werden, die IF-Bedingung ist notwendig, weil der Datensatz nur geändert wird, wenn er auch vorhanden ist; kreiert wurde er nur, wenn die Bedingung 'true' ist:

```
/*Umbenennung der neuen Variablen Effect in Variable*/

%IF (&mi=1 && &count ne 0) %THEN %DO;
      DATA &out_mi (RENAME=(Effect=Variable));
      SET &out_mi;
      RUN;
%END;
```

Der neu erstellte Datensatz hat dann wieder folgenden Inhalt:

	Variable	DF	Wald Chi-square	Pr > Chi-Square
1	geschlecht	1	2.0984	0.1475

und kann wie bisher weiter verarbeitet werden.

Beim folgenden eigentlichen Testen des Makros wurden noch 2 weitere Probleme entdeckt, die zwar beide nichts mit dem Versionswechsel zu tun haben, die aber dennoch untersucht wurden um den Fehler auszubessern oder wenigstens mögliche Fehlerbehebungen vorzuschlagen.

Fehlerbeschreibung Reihenfolge der Angabe der Variablen in xvar/cvar und fxvar/fcvar:

Dieses Phänomen ist beim Testen der Makros in Version 8 und 9 gleichermaßen aufgetreten und somit kein versionsabhängiges Problem, dennoch sollte es hier der Vollständigkeit halber aufgeführt werden: Das Makro bietet die Funktion an bestimmte Variablen, trotz ungünstiger Einflüsse im Endmodell, immer mit aufzunehmen. Realisiert wir das über die Parameter *fxvar=* bzw. *fcvar=*. Bei z. B. zwei stetigen Variablen (Alter und BMI) von denen nur eine unbedingt im Modell bleiben soll (z. B. BMI) muss diese bei der Variablenübergabe (*xvar=*) unbedingt an erster Stelle stehen. Sollte das nicht so sein (*xvar= alter bmi, fxvar=bmi,*), reagiert das Makro mit einer unspezifischen Fehlermeldung:

```
Fehler!

Eine oder mehrere unter 'fxvar' / 'fcvar' übergebene
Variable(n) konnten nicht unter 'xvar' bzw. 'cvar'
gefunden werden. Dies ist jedoch zwingend erforderlich.



Bitte die ENTER- Taste drücken, um fortzufahren.
```

Fehlerbehebung allgemein:

Wenn die Reihenfolge der Variablendeklaration nach oben beschriebenem Muster eingehalten wird treten diese Probleme nicht auf.

"Komisches" Verhalten bei Nichterstellen des MI-Datensatzes in PM_LOGREG.MAC.SAS:

Wenn das Makro angewiesen wird den MI-Datensatz (*mi=1, out_mi=...,*) zu erstellen, so läuft es problemlos durch und liefert die erwarteten Ergebnisse. Wenn diese Option jedoch ausgeschaltet wird (*mi=0,*) entstehen zahlreiche Fehlermeldungen und das Makro beginnt mit langen Rechenzeiten.

Untersucht werden muss auch hier das Makro LOGREG2.MAC.SAS, weil sich darin die eigentlich ausgeführte PROC LOGISTIC befindet.

Ein erster Fehler, der auftaucht, betrifft die Erstellung des RSQUARE-Datensatzes, die Fehlermeldung lässt ein ähnliches Problem wie bei TYPEIII vermuten (geänderter Name im Modulaufruf), dies ist jedoch nachweislich nicht der Fall, der Aufruf ist gleich geblieben. Ähnliches geschieht auch mit dem Datensatz, der unter *c_wert=1,* und *c_data=...,* erzeugt werden soll. Im Folgenden wird nur auf das RQUARE-Problem eingegangen, weil mit diesem wohl auch das c_data-Problem gelöst wird.

Der Log-Output, der die aufgetretenen Fehler beschreibt, sieht folgendermaßen aus:

```
SYMBOLGEN:  Macro variable C_DATA resolves to sasdatei.association
MPRINT(LOGREG2):   ODS OUTPUT Association=sasdatei.association;
SYMBOLGEN:  Macro variable RSQUARE resolves to 1
MLOGIC(LOGREG2):  %IF condition &rsquare=1 is TRUE
MPRINT(LOGREG2):   ODS OUTPUT rsquare=rsquare;
SYMBOLGEN:  Macro variable MI resolves to 0
MLOGIC(LOGREG2):  %IF condition &mi=1 is FALSE
```

An dieser Stelle fällt auf, dass ein großer Makroblock einfach übersprungen wird, weil die Variable *&mi* ungleich 1 ist. Die IF-Bedingung, die das Ganze verursacht, schließt den Aufruf der PROC LOGISTIC ein. Folglich erklärt sich der Fehler, dass der Datensatz nicht erzeugt werden kann. Rsquare sollte aber auch bei *mi=0,* Output einer PROC LOGISTIC sein.

```
SYMBOLGEN:  Macro variable C_WERT resolves to 1
MLOGIC(LOGREG2):  %IF condition &c_wert = 1 is TRUE
SYMBOLGEN:  Macro variable C_DATA resolves to sasdatei.association
MPRINT(LOGREG2):   DATA sasdatei.association (DROP=Label1 cValue1 nValue1);
SYMBOLGEN:  Macro variable C_DATA resolves to sasdatei.association
MPRINT(LOGREG2):   SET sasdatei.association;
MPRINT(LOGREG2):   RUN;

[...]

NOTE: There were 4 observations read from the data set SASDATEI.ASSOCIATION.
NOTE: The data set SASDATEI.ASSOCIATION has 4 observations and 3 variables.
NOTE: DATA statement used (Total process time):
      real time           0.17 seconds
      cpu time            0.01 seconds


SYMBOLGEN:  Macro variable RSQUARE resolves to 1
MLOGIC(LOGREG2):  %IF condition &rsquare=1 is TRUE
MPRINT(LOGREG2):   DATA rsquare_max (DROP=Label1 nValue1 cValue1);
MPRINT(LOGREG2):   SET rsquare;
ERROR: File WORK.RSQUARE.DATA does not exist.
MPRINT(LOGREG2):   RUN;

[...]
```

```
WARNING: Output 'rsquare' was not created.  Make sure that the output object name,
         label, or path is spelled correctly.  Also, verify that the appropriate
         procedure options are used to produce the requested output object.  For
         example, verify that the NOPRINT option is not used.
WARNING: Output 'Association' was not created.  Make sure that the output object name,
         label, or path is spelled correctly.  Also, verify that the appropriate
         procedure options are used to produce the requested output object.  For
         example, verify that the NOPRINT option is not used.
```

Zum Makroteil, der diese Meldungen verursacht (ab Zeile 838):

Hier wird die RSQUARE-Ausgabe verlangt:

```
%IF &rsquare=1 %THEN %DO;
      ODS OUTPUT rsquare=rsquare;
%END;

[...]

%IF &mi=1 %THEN %DO;
```

Wenn *&mi=0* ist, dann wird dieser komplette Block **nicht** ausgeführt:

```
/*überprüfen, ob CLASS- Variablen angegeben wurden*/

%IF &count ne 0 %THEN %DO;

      ODS OUTPUT Type3=&out_mi;

%END;
%ELSE %DO;

      ODS OUTPUT ParameterEstimates=&out_mi;

%END;

PROC LOGISTIC DATA=&data
```

[... / in diesem Teil des Makros werden die Parameter für die PROC LOGISTIC gesetzt]

```
RUN;

ODS OUTPUT CLOSE;
```

[...]

```
%END;
```

Ab Zeile 1019 setzt das Makro wieder ein und führt alle weiteren Schritte aus.

Dass mit diesem Makro also eine Fehlermeldung erzeugt wird und die entsprechenden Datensätze nicht erstellt werden können erscheint nachvollziehbar. Der RSQUARE-Datensatz kann ohne PROC LOGISTIC nicht erstellt werden, diese wird aber nicht ausgeführt, in SAS Version 8 läuft alles anstandslos.

Fehlerbehebung:

Das Problem wurde an den Programmierer der Makros (Christoph Ziegler) weitergeleitet, der das Problem erkannte und vorschlug das *%END;* unmittelbar nach den für *mi=1,* relevanten Makroteilen zu setzen:

```
%IF &mi=1 %THEN %DO;    /*Bedingungen der 'mi'-Abfrage, die bei &mi=0 nicht*/
                        /*ausgeführt werden*/

/*überprüfen, ob CLASS- Variablen angegeben wurden*/

%IF &count ne 0 %THEN %DO;

      ODS OUTPUT Type3=&out_mi;

%END;
%ELSE %DO;

      ODS OUTPUT ParameterEstimates=&out_mi;

%END;
%END;                   /*%END; zum Beenden der 'mi'-Abfrage damit die fol-*/
                        /*gende PROC LOGISTIC ausgeführt werden kann*/
```

Somit hat der Parameter *mi=0,* keinen Einfluss mehr auf die nachfolgende PROC LOGISTIC, die immer ausgeführt wird und die gewünschten Datensätze erstellt.

In den SAS-Makros V8 ist dies so realisiert und es bleibt zu klären, wodurch der Fehler entstanden ist. Leider gibt es keine befriedigende Antwort auf diese Frage, denn die Makros wurden komplett so kopiert wie sie in Version 8 vorlagen und auf Version 9 getestet. Mögliche Erklärungen für den abgeänderten Programmablauf können sein:

Durch ständiges Ändern der Makros zu Testzwecken hätte es passieren können, dass Teile vom Code (zum Beispiel auch durch Copy and Paste) an eine falsche Stelle verschoben wurden. Dies ist in sofern unwahrscheinlich, weil dieser Fehler ziemlich früh beim Testen des Makros auf Version 9 aufgetreten ist (nachdem die TYPEIII-Geschichten behoben wurden) und zudem nie Veranlassung dazu bestand den Code an dieser Stelle und mit solcher Wirkung zu ändern.

Eine andere Erklärung könnte sein, dass zufällig eine alte Version der Makros (mit diesem noch nicht korrigierten Fehler) kopiert und verwendet wurde und der Fehler sich also aus früheren Zeiten fortgesetzt hat. Diese Erklärung ist aber auch nicht wahrscheinlich.

Das Makro FL_HEINZE.MAC.SAS:

Bei manchen (meist kleinen) Datensätzen kann es vorkommen, dass die Zielgröße allein über eine bestimmte Wertekonstellation der Einflussgrößen vorhergesagt werden kann (ein Regressionsmodell ist dann eigentlich nicht mehr notwendig), man spricht dabei von einer (quasi) complete separation. Dieser Fall bringt Probleme beim Erstellen des Regressionsmodells mit sich. Zum gezielten Abfangen dieses Falles wird das FL-Makro von Georg Heinze [6] aktiviert.

Diese Situation kommt beim vorliegenden Datensatz nicht vor, zum Testen habe ich ein kleines Hilfsprogramm geschrieben, und einen weiteren Datensatz aus der Abteilung verwendet, der das entsprechende Problem erzeugt. Das so manuell aufgerufene Makro zeigte keinen Versionskonflikt.

1.3 MAKROS ZUR MODELLGÜTE

Nachdem das Regressionsmodell erstellt wurde gilt es hier nun zu prüfen wie gut es denn zu den zu den beobachteten Daten passt. Hierzu dienen die nächsten beiden Makros.

1.3.1 PM_GOF.MAC.SAS

Für die Anpassungsgüte (Goodness of Fit) wird der Zusammenhang zwischen den beobachteten Werten y und den durch das Modell geschätzten Werten $\hat{y}$ untersucht. Auch hier kommt wieder die PROC LOGISTIC zum Einsatz.

Dieses Makro greift auf mehrere Untermakros zu, das Makro von Kuss [2] stellt 5 verschiedene statistische Tests zur Prüfung der Goodness of Fit zur Verfügung. Da dieses Makro allerdings in SAS Version 6 erstellt wurde, und es damals noch kein CLASS-Statement in der PROC LOGISTIC gab, ist das Makro von Friendly notwendig, das eine manuelle Dummy-Kodierung der kategoriellen Variablen durchführt. [5]

Das Makro funktioniert in SAS Version 9 aber problemlos und liefert dieselben Ergebnisse wie in Version 8. Die Laufzeiten des Makros liegen mit einer Minute und 36 Sekunden im erwarteten Bereich.

1.3.2 PM_ROC.MAC.SAS

Die ROC-Analyse bestimmt die Prognosegüte, unter anderem Sensitivität und Spezifität bei unterschiedlichem Cut-Point (Der Cut-Point bestimmt die Merkmalsausprägungen einer Beobachtung mit der diese in eine der Gruppen der Zielgröße einsortiert wird.). Das Makro liefert Auskunft über die Verteilung von Sensitivität und Spezifität und stellt diese grafisch dar. Hauptsächlich wird hier mit DATA-Steps gearbeitet.

Die Ausführung dieses Makros mit verschiedenen Testparametern ergab keine Hinweise auf Versionskonflikte. Die Laufzeiten liegen mit etwas mehr als zwei Minuten im normalen Bereich und sind gegenüber Version 8 nicht auffällig.

1.4 MAKROS ZUR MODELLVALIDIERUNG

Bei einem an einem Beispieldatensatz gewonnenem Regressionsmodell muss geprüft werden mit welcher Güte (Prognosegüte) dieses Modell auf neue Daten angepasst werden kann. Es gibt mehrere Arten der Validierung, man kann diese einteilen in interne, temporale und externe Validierungsmethoden.

Bei den **internen** Validierungsmethoden wird der Datensatz, der auch zur Erstellung des Modells zur Verfügung stand, benutzt um das Modell zu testen. Beispielsweise werden mehrere zufällige Datensätze aus dem Orignaldatensatz gewonnen, die dann auf das Modell angewendet werden um seine Güte zu ermitteln. Diese Vorgehensweise muss in der Regel mehrere hundert Mal angewandt werden um zu einem richtigen Ergebnis zu kommen.

Die **temporale** Validierung nutzt den Orignaldatensatz, der mit einem Zufallsverfahren in mehrere, meistens zwei, Datensätze gespalten wird. An einem Teildatensatz wird das Modell entwickelt und an dem anderen validiert. Allerdings stammen beide Datensätze immer noch aus einer gemeinsamen Quelle.

Wenn eine Validierung auf einem zweiten, unabhängigen Datensatz beruht, spricht man von einer **externen** Validierung. Da hier mit komplett neuen Daten gearbeitet wird ist dies die weitestgehende Form der Validierung.

1.4.1 PM_EXTERNAL_VALIDATION.MAC.SAS

Bei der externen Validierung ist, wie oben bereits erwähnt, die Anwendung eines zweiten unabhängigen Datensatzes gefordert. Dieser muss für das Makro strukturgleich sein, das heißt, dass dieselben Variablennamen und Formatierungen in beiden Datensätzen gegeben sein müssen. Das Makro untersucht mit Hilfe der Makros PM_LOGREG.MAC.SAS und PM_ROC.MAC.SAS die Prognosegüte eines festen Modells auf Basis neuer Daten. Dabei unterscheidet das Makro nicht zwischen rein externer und temporaler Validierung, da dies Fragen inhaltlicher Natur sind.

Beim Testen des Makro sind keine weiteren Probleme aufgetreten. Dabei ist allerdings zu beachten, dass zuvor bereits die beiden Makros, auf die die externe Validierung zugreift, geändert und an die Version 9 angepasst wurden. Wäre das nicht der Fall gewesen, so würden spätestens hier die weiter oben beschriebenen Probleme auftauchen. Mit einer Laufzeit von 52 Sekunden steht das Makro der Laufzeit in Version 8 in nichts nach.

1.4.2 PM_DATASPLITTING.MAC.SAS

Um für die temporale Validierung einen zweiten Datensatz mit den geforderten Eigenschaften (gleiche Variablen und Formatierungen) zu erhalten erzeugt dieses Makro aus einem Quelldatensatz auf Basis eines angegebenen Prozentsatzes zwei zufällige Teil-Datensätze. Dabei greift das Makro auf die SAS-Funktion RANUNI zu.

Dieses Makro ist eins der "einfachen" Makros und wie erwartet sind hier keine Probleme aufgetreten. Aufgrund der einfachen Struktur des Makros sind 13 Sekunden Laufzeit vollkommen ausreichend.

1.4.3 PM_CROSSVALIDATION.MAC.SAS

Die Kreuzvalidierung ist eine Weiterführung des Data-Splittings. Dabei wird der Datensatz mehrfach in Gruppen unterteilt und die Auswertungen jeweils auf diese Untergruppen durchgeführt. Es gibt 4 verschiedene Kreuzvalidierungsarten [1], die von dem Makro durchgeführt werden können. Für die einzelnen Modellerstellungen und –validierungen werden wie bei dem Makro für die externe Validierung die Makros PM_LOGREG.MAC.SAS und PM_ROC.MAC.SAS verwendet.

Weil aber eben genau diese beiden Makros zuvor schon auf die Version 9 angepasst wurden, sind hier keine weiteren Probleme aufgetreten. Durch die Erstellung mehrer Gruppen und dem Berechnen der jeweiligen Parameter besitzt dieses Makro eine etwas längere Laufzeit, die aber mit gut 18 Minuten für 10 Gruppen und 50 Wiederholungen im Normbereich liegt.

1.4.4 PM_BOOTSTRAP_VALIDATION.MAC.SAS

Das Prinzip des Ziehens mit Zurücklegen kommt bei der Bootstrap-Validierung zum Einsatz. Es werden neue Datensätze gebildet, die dieselbe Größe wie der ursprüngliche Datensatz haben, und ähnliche statistische Eigenschaften. Dabei kann es vorkommen, dass einzelne Beobachtungen aus dem Quelldatensatz einfach, mehrfach oder gar nicht vorkommen. Durch wiederholte Erzeugung solcher Bootstrap-Samples und der Ermittlung der einzelnen Bootstrap-Schätzer kann nun ein validierter Schätzer der Prognosegüte gewonnen werden.

Beim Testen des Makros PM_BOOTSTRAP_VALIDATION.MAC.SAS stellt man fest, dass im Output für die Emax-Werte NM, DIF, D und OPT fehlende Werte angezeigt werden:

Emax O	Emax B	Emax NM	Emax DIF	Emax D	Emax OPT	%
0	0	.	.	.	.	.

Ein Blick ins Log-Fenster bringt eine Fehlermeldung über einen Datensatz (X.VEKTOR.DATA), der nicht gefunden werden konnte:

```
SYMBOLGEN:  Macro variable VALIDATION resolves to BOOTSTRAP
MLOGIC(PM_ROC):  %IF condition %UPCASE(&validation) ne NONE is TRUE
SYMBOLGEN:  Macro variable MACRO_PATH resolves to 'H:\Felix\PM_Makros_V9'
MPRINT(PM_ROC):   LIBNAME x 'H:\Felix\PM_Makros_V9';
NOTE: Libref X was successfully assigned as follows:
      Engine:        V9
      Physical Name: H:\Felix\PM_Makros_V9
MPRINT(PM_ROC):   DATA vektor;
MPRINT(PM_ROC):   SET x.vektor;
ERROR: File X.VEKTOR.DATA does not exist.
MPRINT(PM_ROC):   logit_p_stern=log(p_stern/(1-p_stern));
SYMBOLGEN:  Macro variable RESP_VAR resolves to eu_rente
MPRINT(PM_ROC):   eu_rente=.;
MPRINT(PM_ROC):   RUN;
```

```
NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.VEKTOR may be incomplete.  When this step was stopped there
         were 0 observations and 3 variables.
NOTE: DATA statement used (Total process time):
      real time           0.04 seconds
      cpu time            0.03 seconds
```

Behebung des Fehlers fehlender X.VEKTOR - Datensatz:

In SAS Version 8 enden die permanenten SAS-Dateien auf .sd7, ab Version 9 erkennt SAS diese Endung nicht mehr als DATASET, sondern verlangt die Endung .sas7bdat. Eine Umbenennung des permanenten Datensatzes im Makroverzeichnis (Libref X im Log) von vektor.sd7 nach vektor.sas7bdat schafft Abhilfe für dieses Problem.

Die Laufzeit bei diesem Makro kann aufgrund der Erstellung von mehreren (z. B. 200) Samples sehr lange dauern. Sie liegt aber gegenüber Version 8 mit ca. 8 Stunden für 200 Samples beim getesteten Datensatz im normalen Bereich.

1.4.5 PM_SHRINKAGE_VALIDATION.MAC.SAS

Dieses Verfahren ist ein internes Validierungsverfahren, es handelt sich um eine so genannte Kalibrierung. Es wird versucht einen Überoptimismus bei Bestimmung der Regressionskoeffizienten zu verhindern. Dazu werden die beobachteten Werte mit den im Modell vorhergesagten Werten aufgetragen und die Steigung der Geraden beobachtet. Bei gleichen Datensätzen beträgt diese 1, sobald allerdings ein anderer Datensatz angewandt wird, ist die Geradensteigung üblicherweise kleiner 1. Diese Steigung (Shrinkage) kann zur Korrektur der Regressionskoeffizienten, und demnach auch der Prognosegüte, herangenommen werden.

Beim Testen des Makros sind keine Probleme aufgetaucht, die Laufzeit liegt bei knapp 2 Minuten.

1.5 ZUSAMMENFASSUNG UND AUSBLICK

Das Makropaket wurde in SAS Version 8 geschrieben und getestet. Meine Aufgabe war diese auf SAS Version 9.1 zu testen und eventuelle Fehler zu korrigieren. Einige Fehler waren sofort gefunden, andere wiederum waren etwas kniffeliger zu entdecken und zu beheben. Das Testen der Makros bezog sich während meines Praktikums auch nur auf die Hauptmakros, die vom Benutzer selbst aufgerufen werden. Im Folgenden findet sich eine Übersicht über die einzelnen Makros mit einer kurzen Bemerkung über eventuell aufgetretene Fehler.

Makroname	**SAS Version 8**	**SAS Version 9**
DESCRIPTION	Keine Fehler aufgetreten	Keine Fehler aufgetreten
MULTICOLLIN	Keine Fehler aufgetreten	Keine Fehler aufgetreten
MISSING	Auch bei *imputation_art=0,* wird die multiple Imputation ausgeführt.	Auch bei *imputation_art=0,* wird die multiple Imputation ausgeführt.
MI_ANALYZE	Keine Fehler aufgetreten	Keine Fehler aufgetreten
INFLUENCE	Keine Fehler aufgetreten	Keine Fehler aufgetreten
UNI_LOGREG	Keine Fehler aufgetreten	TYPEIII-Fehler: Geänderter Aufruf: Type3 statt TypeIII Umbenennung 'Variable' in 'Effect'
LOGREG	Keine Fehler aufgetreten	TYPEIII-Fehler: Geänderter Aufruf: Type3 statt TypeIII Umbenennung 'Variable' in 'Effect'
GOF	Keine Fehler aufgetreten	Keine Fehler aufgetreten
ROC	Keine Fehler aufgetreten	Keine Fehler aufgetreten
EXTERNAL_VALIDATION	Keine Fehler aufgetreten	Keine Fehler aufgetreten
DATASPLITTING	Keine Fehler aufgetreten	Keine Fehler aufgetreten
CROSSVALIDATION	Keine Fehler aufgetreten	Keine Fehler aufgetreten
BOOTSTRAP_VALIDATION	Keine Fehler aufgetreten	X.VEKTOR.DATA: Umbenennung des permanenten Datensatzes vektor.sd7 zu vektor.sas7bdat
SHRINKAGE_VALIDATION	Keine Fehler aufgetreten	Keine Fehler aufgetreten

Die Hilfsmakros, auf die diese Makros zugreifen, kommen teilweise von externen Stellen, anderen Universitäten etc. Sie sind zum Teil in SAS Version 6 geschrieben und stellen daher die Frage, ob die Kompatibilität zu SAS Version 9 gewährleistet werden kann. In meinen Tests sind keine Versionskonflikte aufgefallen, allerdings muss das nicht heißen, dass es die nicht gibt. Vor dem unbedachten Gebrauch der Makros sollte also geklärt werden, ob die Untermakros zu SAS Version 9 kompatibel sind. Hier können in den meisten Fällen die Autoren der entsprechenden Makros am besten Auskunft geben.

1.6 LITERATURVERZEICHNIS

[1] MUCHE, R.: Entwicklung und Validierung von Prognosemodellen auf Basis der logistischen Regression, Habilitationsschrift (2004)

[2] KUSS, O.: Global goodness-of-fit tests in logistic regression with sparse data. Statist. Med. 21: 3789 – 3801 (2002)

[3] JACOBI, E., RÖSCH, M. ALT, B.: Rehabilitationswissenschaftlicher Forschungsverbund Ulm - "Bausteine der Reha". Die Rehabilitation 37 Suppl. 2: 111 – 116 (1998)

[4] KALUSCHA, R., JACOBI, E.: Eine Datenbank zur Effektivitätsbeurteilung: Das Datenbankkonzept des rehabilitationswissenschaftlichen Forschungsverbundes Ulm. *DRV-Schriften* **20:** 218 - 219 (2000)

[5] FRIENDLY, H.: SAS-Makro dummy.sas http://www.psych.yorku.ca/friendly/lab/file/macros/dummy.sas (aufgerufen am 16.1.2004) (2001)

[6] HEINZE, G., SCHEMPER, M.: A solution to the problem of separation in logistic regression. *Statist. Med.* **21:** 2409 - 2419 (2002)