SAS® PROC COMPARE - Vergleich von SAS®-Datensätzen leicht gemacht?

Cordula Massion
Accovion GmbH
Softwarecenter 3
35037 Marburg
cordula.massion@accovion.com

Beate Jakobi-Plöhn
Accovion GmbH
Helfmann-Park 10
65760 Eschborn
beate.jakobi-ploehn@accovion.com

Jörg Müller
Accovion GmbH
Helfmann-Park 10
65760 Eschborn
joerg.mueller@accovion.com

Zusammenfassung

Insbesondere im Rahmen der Validierung von SAS®-Programmen mittels Doppelprogrammierung ist der SAS Programmierer häufig mit der Aufgabe konfrontiert, den Inhalt zweier Datensätze auf Gleichheit zu überprüfen. Hierzu bietet sich der Einsatz der SAS® Prozedur PROC COMPARE an.
Das Ergebnis „Note: No unequal values were found. All values compared are exactly equal“ heißt allerdings nicht zwangsläufig, dass die zu vergleichenden Datensätze wirklich identisch sind. Die richtige Anwendung von Optionen und Anweisungen ist die Voraussetzung für belastbare Ergebnisse und deren transparente und übersichtliche Dokumentation.
Dieser Beitrag wird die wesentliche Funktionalität sowie die Grenzen und Risiken der Prozedur PROC COMPARE beleuchten.

Schlüsselwörter: PROC COMPARE, ODS PDF, SYSINFO, Vergleich von Datensätzen

1 Einführung

In der täglichen Arbeit von SAS®-Programmierern stellt sich häufig die Frage nach der Gleichheit von zwei Datensätzen. Hierzu bietet sich die SAS® Prozedur PROC COMPARE an, weil sie es ermöglicht, Datensätze bezüglich verschiedener Aspekte zu vergleichen und die Ergebnisse übersichtlich darzustellen.
Auch der Vergleich von Ergebnissen, z.B. Datenlistings oder Tabellen, ist denkbar, indem man sie zusätzlich als Datensätze generiert. Diese Datensätze können dann eingelesen und verglichen werden [1].

1.1 Was macht PROC COMPARE?

PROC COMPARE vergleicht Datensatzinhalte indem passende Beobachtungen und Variablenpaare identifiziert und bezüglich ihrer Metadaten und Inhalte verglichen werden. Neben dem Vergleich des kompletten Inhalts zweier Datensätze ist es auch möglich, gezielt einzelne Variablen innerhalb eines Datensatzes miteinander zu vergleichen. Die Ausführungen in diesem Beitrag beschränken sich jedoch auf den Vergleich des kompletten Inhalts zweier Datensätze.

Im Rahmen des Metadatenvergleichs werden die Datensatz-Labels sowie die Formate, Länge, Labels, der Datentyp und die Position der Variablen in den jeweiligen Datensätzen verglichen.

Desweiteren wird analysiert, ob die Datensätze Beobachtungspaare bzw. Variablenpaare enthalten, die einen Vergleich der Variableninhalte ermöglichen. Beobachtungen oder Variablen, die vom inhaltlichen Vergleich ausgeschlossen wurden, werden identifiziert.

Schließlich gibt PROC COMPARE detaillierte Auskunft über Unterschiede bei Variableninhalten.

1.2 Anwendungsgebiete von PROC COMPARE

Im Rahmen von klinischen Studien spielt PROC COMPARE insbesondere bei der unabhängigen Doppelprogrammierung von Analyse-Datensätzen eine wichtige Rolle. Hier kommt es darauf an, die von einem unabhängigen, zweiten Programmierer anhand von Spezifikationen erstellten Datensätze gegen die Original-Datensätze zu überprüfen und eventuelle Unterschiede zu erkennen [2].

Ein weiteres Einsatzgebiet ist die Überprüfung und Dokumentation von Datensatzänderungen. So können Datenänderungen bei regelmäßigen Datenupdates kontrolliert und dokumentiert werden. Außerdem ist es möglich, klar definierte Datenänderungen mithilfe eines Vorher/Nachher-Vergleichs zu dokumentieren. Dies könnte z.B. bei notwendigen Datenänderungen nach Datenbankschluss sinnvoll sein. In diesem Fall muss die im SAS® Programm durchgeführte manuelle Datenänderung transparent und unzweifelhaft dokumentiert werden, was neben einer ausführlichen Dokumentation im SAS® Programm mit dem entsprechenden PROC COMPARE Output untermauert werden kann.

Auch die Auswirkungen von Programmänderungen auf einen Datensatz können mit PROC COMPARE überprüft und dokumentiert werden. Eine gezielte Programmänderung, z.B. die Herleitung eines Baseline Flags, sollte sich nur auf Variablen auswirken, die mit diesem Baseline Flag in Zusammenhang stehen. Mit Hilfe eines Vergleiches des Datensatzes vor versus nach der Programmänderung lässt sich belegen, dass wirklich nur die gewünschten Variablen von der Änderung betroffen sind. Als Validierung der Programmänderung reicht dieses Verfahren allein aber nicht aus, weil es lediglich die

Auswirkungen der Programmänderung auf den gerade aktuellen Datensatz zeigt – mit anderen Datenkonstellationen könnten bei fehlerhafter Programmierung unerwartete Variablen betroffen sein.

1.3 Wie geht's: Einfache Syntax

Mit Hilfe der folgenden Syntax wird der Standardvergleich zweier Datensätze durchgeführt:

```
PROC COMPARE
     BASE=class1a
     COMP=class1a;
RUN;
```

In diesem Fall wird ein Datensatz mit sich selbst verglichen, was zu folgendem Ergebnis führt:

```
                        The COMPARE Procedure
              Comparison of WORK.CLASS1A with WORK.CLASS1A
                            (Method=EXACT)

(1)                          Data Set Summary
      Dataset                 Created          Modified  NVar    NObs
      WORK.CLASS1A  28FEB14:12:58:03  28FEB14:12:58:03     3       3
      WORK.CLASS1A  28FEB14:12:58:03  28FEB14:12:58:03     3       3

(2)                          Variables Summary
                  Number of Variables in Common: 3.

(3)                         Observation Summary
                     Observation      Base  Compare
                     First Obs           1        1
                     Last  Obs           3        3

     Number of Observations in Common: 3.
     Total Number of Observations Read from WORK.CLASS1A: 3.
     Total Number of Observations Read from WORK.CLASS1A: 3.

     Number of Observations with Some Compared Variables Unequal: 0.
     Number of Observations with All Compared Variables Equal: 3.

     NOTE: No unequal values were found. All values compared are exactly equal.
```

Der Output gliedert sich in drei Hauptabschnitte:

- „Data Set Summary" (1): Detailinformationen zu den Datensätzen, wobei insbesondere die Anzahl der Variablen und Beobachtungen pro Datensatz interessant sind.
- „Variable Summary" (2): Informationen bezüglich der Vergleichbarkeit der Variablen.
- „Observation Summary" (3): Informationen bezüglich der Vergleichbarkeit von Beobachtungen.

Wie bei dem Vergleich eines Datensatzes mit sich selbst erwartet stimmt die Anzahl der Variablen und Beobachtungen („Data Set Summary“), die Anzahl der gemeinsamen Variablen („Variables Summary“) sowie die Anzahl der gemeinsamen Beobachtungen („Observation Summary“) jeweils überein und es werden keine Unterschiede bei den Variableninhalten gefunden („Note: No unequal values found. All values compared are exactly equal“).

2 Ergänzungen der Syntax

Die oben beschriebene einfache Syntax für den Vergleich zweier Datensätze kann durch eine Vielzahl von Optionen ergänzt werden, die das Ergebnis auf einzelne Aspekte beschränken bzw. einzelne Aspekte besonders ausführlich beleuchten. Auch wenn es in vielen Fällen ausreichend und sinnvoll ist, den Standardoutput zu verwenden, so kann die Einschränkung auf Teilaspekte oder eine andere Darstellungsform der Ergebnisse in bestimmten Situationen sinnvoll sein. Im Folgenden werden einige nützliche und häufig zum Einsatz kommende Optionen und Statements anhand von Beispielen vorgestellt.

2.1 „No unequal values found“, aber kein kompletter Vergleich

Bei dem Vergleich zweier Datensätze erhält man den folgenden Output:

```
(1)                          Data Set Summary
      Dataset                  Created          Modified  NVar    NObs
      WORK.CLASS1A   08MAR14:07:37:40  08MAR14:07:37:40     3       3
      WORK.CLASS1B   08MAR14:07:37:40  08MAR14:07:37:40     3       1

(2)                          Variables Summary
                   Number of Variables in Common: 3.

(3)                         Observation Summary
                       Observation      Base  Compare
                       First Obs           1        1
                       Last  Match         1        1
                       Last  Obs           3

    Number of Observations in Common: 1.
    Number of Observations in WORK.CLASS1A but not in WORK.CLASS1B: 2.
    Total Number of Observations Read from WORK.CLASS1A: 3.
    Total Number of Observations Read from WORK.CLASS1B: 1.

    Number of Observations with Some Compared Variables Unequal: 0.
    Number of Observations with All Compared Variables Equal: 1.

    NOTE: No unequal values were found. All values compared are exactly equal.
```

Der erste Blick fällt vermutlich auf die Note “No unequal values found. All values compared are exactly equal“ am Ende des Outputs in der „Observation Summary“ (3), die vermuten lässt, dass die beiden verglichenen Datensätze komplett identisch sind. Bei genauerer Analyse des Ergebnisses stellt man aber fest, dass nicht alle Beobachtungen

in den Vergleich einbezogen wurden, was hier in der „Data Set Summary“ (1) oder noch deutlicher in der „Observation Summary“ (2) erkennbar ist.

Mit Hilfe der Option LISTALL im folgenden Programmcode

```
PROC COMPARE
     BASE=class1a
     COMP=class1b LISTALL;
RUN;
```

wird der Output durch den Abschnitt „Comparison Results for Observations“ ergänzt, in dem alle Beobachtungen gelistet werden, die nur in einem der beiden Datensätze vorkommen, also vom Vergleich ausgeschlossen werden:

```
            Comparison Results for Observations

  Observation 2 in WORK.CLASS1A not found in WORK.CLASS1B.
  Observation 3 in WORK.CLASS1A not found in WORK.CLASS1B.
```

Durch die Benutzung der Option ist die Gefahr, vom Vergleich ausgeschlossene Beobachtungen zu übersehen, deutlich geringer.

Trotz identischer Anzahl der Beobachtungen in beiden Datensätzen laut „Data Set Summary“ kann es, z.B. beim Vergleich einzelner BY-Gruppen, Beobachtungen geben, die in einem der beiden Datensätze nicht vorkommen. Aus diesem Grund ist es empfehlenswert, die Option LISTALL zu verwenden, um auf diese Fälle aufmerksam zu werden und die betreffenden Beobachtungen eindeutig identifizieren zu können.

In ähnlicher Weise werden durch LISTALL auch Variablen, die nur in einem der beiden Datensätze vorhanden sind, gelistet.

2.2 „Unequal values were found …“

Die folgende „Observation Summary“ zeigt, dass die beiden verglichenen Datensätze gleich viele Beobachtungen enthalten. Außerdem erhält man eine Aussage darüber wie viele Beobachtungen identisch bzw. ungleich sind.

```
                    Observation Summary
                  Observation      Base  Compare

                  First Obs           1        1
                  First Unequal       1        1
                  Last  Unequal       3        3
                  Last  Obs           3        3

  Number of Observations in Common: 3.
  Total Number of Observations Read from WORK.CLASS2A: 3.
  Total Number of Observations Read from WORK.CLASS2B: 3.

  Number of Observations with Some Compared Variables Unequal: 2.
  Number of Observations with All Compared Variables Equal: 1.
```

In der zugehörigen „Value Comparison Summary“ sieht man, wie viele (1) und welche (2) Variablen Unterschiede aufweisen.

```
                      Values Comparison Summary

(1)  Number of Variables Compared with All Observations Equal: 1.
     Number of Variables Compared with Some Observations Unequal: 2.
     Total Number of Values which Compare Unequal: 4.
     Maximum Difference: 1.

(2)                    Variable  Type  Len   Ndif   MaxDif

                       NAME      CHAR    8      2
                       AGE       NUM     8      2    1.000
```

Die inhaltlichen Unterschiede finden sich im Abschnitt „Value Comparison Results for Variables“. Sie werden variablenweise angezeigt, d.h. pro Variable werden alle Beobachtungen mit Unterschieden standardmäßig in der folgenden Form gelistet:

```
           Value Comparison Results for Variables
_____________________________________________________________
             ||  Base Value           Compare Value
        Obs  ||  NAME                  NAME
  _________  ||  ________              ________
             ||
         1   ||  Alfred                William
         3   ||  William               Alfred
_____________________________________________________________


 ____________________________________________________________
             ||        Base    Compare
        Obs  ||         AGE        AGE      Diff.     % Diff
  _________  ||  _________  _________  _________  _________
             ||
         1   ||    14.0000    15.0000     1.0000     7.1429
         3   ||    15.0000    14.0000    -1.0000    -6.6667
_____________________________________________________________
```

Eine weitere Möglichkeit diese Unterschiede darzustellen bietet die Option TRANSPOSE.

```
PROC COMPARE
     BASE=class2a
     COMP=class2b TRANSPOSE;
RUN;
```

mit der alle Unterschiede sortiert nach einzelnen Beobachtungen gelistet werden:

```
              Comparison Results for Observations

_OBS_1=1 _OBS_2=1:
Variable    Base Value       Compare         Diff.        % Diff
     AGE     14.000000     15.000000      1.000000      7.142857
    NAME        Alfred       William

_OBS_1=3 _OBS_2=3:
Variable    Base Value       Compare         Diff.        % Diff
     AGE     15.000000     14.000000     -1.000000     -6.666667
    NAME       William        Alfred
```

In beiden Auflistungen sieht man, dass die Unterschiede durch eine fehlende bzw. unterschiedliche Sortierung der Datensätze auftreten.
Aus diesem Grund sollten die Datensätze vor einem Vergleich nach einem oder mehreren möglichst eindeutigen Schlüsseln sortiert werden.

```
PROC SORT DATA=class2a;
   BY name;
RUN;
PROC SORT DATA=class2b;
   BY name;
RUN;

PROC COMPARE
     BASE=class2a
     COMP=class2b LISTALL;
RUN;
```

Der Vergleich nach der Sortierung zeigt an, dass die beiden Datensätze inhaltlich identisch sind.

```
                    The COMPARE Procedure
          Comparison of WORK.CLASS2A with WORK.CLASS2B
                       (Method=EXACT)

                      Data Set Summary

 Dataset                 Created          Modified  NVar    NObs

 WORK.CLASS2A  19FEB14:10:31:37  19FEB14:10:31:37      3       3
 WORK.CLASS2B  19FEB14:10:31:37  19FEB14:10:31:37      3       3

                      Variables Summary

             Number of Variables in Common: 3.

                      Observation Summary

                Observation      Base  Compare

                First Obs           1        1
                Last  Obs           3        3

 Number of Observations in Common: 3.
 Total Number of Observations Read from WORK.CLASS2A: 3.
 Total Number of Observations Read from WORK.CLASS2B: 3.

 Number of Observations with Some Compared Variables Unequal: 0.
 Number of Observations with All Compared Variables Equal: 3.

 NOTE: No unequal values were found. All values compared are exactly equal.
```

2.3 Verwendung des ID-Statements, Option CRITERION

Mehr Informationen bezüglich der gefundenen Unterschiede bekommt man durch Hinzufügen eines ID-statements:

```
PROC COMPARE
     BASE=class3a
     COMP=class3b;
     ID name;
RUN;
```

Die ID-variablen (hier: `name`) werden anstelle der Observation number zur Identifizierung einzelner Beobachtungen verwendet und erleichtern dadurch das Auffinden in großen Datensätzen.

```
                    Values Comparison Summary

 Number of Variables Compared with All Observations Equal: 5.
 Number of Variables Compared with Some Observations Unequal: 1.
 Total Number of Values which Compare Unequal: 3.
 Maximum Difference: 0.00004903.
```

```
          Value Comparison Results for Variables
_____________________________________________________________
             ||        Base    Compare
 NAME        ||         BMI        BMI      Diff.     % Diff
 ________    ||   _________  _________  _________  _________
             ||
 Alfred      ||       26.47      26.47  0.0000284   0.000107
 Alice       ||       29.47      29.47  -0.000024  -0.000081
 William     ||       28.37      28.37  -0.000049  -0.000173
_____________________________________________________________
```

Im obigen PROC COMPARE Ergebnis erkennt man außerdem, dass die angezeigten Unterschiede nur minimal sind und hier keine Relevanz haben. Um diese Anzeige zu vermeiden empfehlen wir die Option CRITERION.

```
PROC COMPARE
     BASE=class3a
     COMP=class3b CRITERION=0.01;
     ID name;
RUN;
```

Mit Hilfe dieser Option wird die Genauigkeit des Vergleichs festgelegt. Dies kann insbesondere bei Vergleichen von Datensätzen, die z.B. auf unterschiedlichen Betriebssystemen erstellt wurden, relevant sein. Dadurch wird die Erkennung von technisch bedingten Unterschieden vermieden. In der „Value Comparison Summary“ werden die Unterschiede immer noch erwähnt, aber der Abschnitt „Value Comparison Results for Variables“ entfällt.

```
                    Values Comparison Summary

  Number of Variables Compared with All Observations Equal: 6.
  Number of Variables Compared with Some Observations Unequal: 0.
  Total Number of Values which Compare Unequal: 0.
  Total Number of Values not EXACTLY Equal: 3.
  Maximum Difference Criterion Value: 0.0000017285.
```

2.4 Verwendung der Option OUT

Bei Vergleichen von Zeichenketten kann es passieren, dass Variableninhalte als unterschiedlich in der „Value Comparison Results for Variables“ gelistet werden, diese Unterschiede aber nicht sichtbar sind. Dies liegt an der auf 20 Stellen begrenzten Darstellung von Zeichenketten. Bei der Verwendung von TRANSPOSE werden sogar nur die ersten 12 Stellen angezeigt. Das Zeichen „+“ oberhalb der Anzeige des Variableninhalts weist daraufhin, dass der komplette Inhalt der Variablen nicht angezeigt wird.

```
                ||  Base Value            Compare Value
          Obs   ||  MODEL                  MODEL
                ||  ____________________+  ____________________+
                ||
          22    ||   TT 1.8 convertible     TT 1.8 convertible
          23    ||   TT 1.8 Quattro 2dr     TT 1.8 Quattro 2dr
```

Hier empfiehlt sich die Erstellung eines Output Datensatzes mit Hilfe der Option OUT= und entsprechenden Zusatzoptionen:

```
PROC COMPARE
   BASE=cars1
   COMP=cars2 OUT=carsdiff
   OUTNOEQUAL OUTBASE OUTCOMP OUTDIF;
RUN;
```

In Abhängigkeit von den folgenden Optionen werden Beobachtungen in den Output Datensatz geschrieben:

OUTBASE: Beobachtungen aus BASE (Type of Observation=BASE)
OUTCOMP: Beobachtungen aus COMP (Type of Observation =COMPARE)
OUTDIF: Beobachtungen mit den Unterschieden (Type of Observation =DIF)

Die Option OUTNOEQUAL bewirkt, dass nur Beobachtungen, die unterschiedliche Werte haben, in dem Output Datensatz berücksichtigt werden.

	Type of Observation	Observation Number	MAKE	MODEL
1	BASE	22	Audi	TT 1.8 convertible 2dr (coupe)
2	COMPARE	22	Audi	TT 1.8 convertible 2dr coupe
3	DIF	22	.....	.....X....X.....
4	BASE	23	Audi	TT 1.8 Quattro 2dr (convertible)
5	COMPARE	23	Audi	TT 1.8 Quattro 2dr convertible
6	DIF	23	.....	.....X.....X.....

Die ersten beiden Variablen im Output-Datensatz beinhalten den Typ der Beobachtung und die Observation number. Danach folgen, falls vorhanden, die ID-Variablen und alle anderen Variablen aus den zu vergleichenden Datensätzen.
In der DIF Zeile kann es folgende Einträge geben:

1. Zeichenkette identisch: entsprechend der Länge der Variablen werden Punkte angezeigt
2. Zeichenkette ungleich: alle Stellen, die unterschiedlich sind, werden mit einem „X“ markiert
3. Numerischer Wert identisch: Anzeige „E“
4. Numerischer Wert ungleich: die Differenz der beiden Variablen wird angezeigt.

Die Verwendung von OUT= erleichtert auch das Auffinden von Beobachtungen, die nur in einem der beiden Datensätze vorhanden sind. In diesen Fällen gibt es nur eine Beobachtung vom Typ BASE bzw. COMP, Typ DIF entfällt da ein Vergleich nicht möglich ist.

2.5 Betrachtung von fehlenden Werten

Der folgende Outputauszug zeigt einen Vergleich von Datensätzen, die fehlende Werte enthalten. Lediglich fehlende Werte gleichen Typs, also missing values oder gleichartige special missings werden als identisch erkannt.

```
                    Observation Summary

                Observation      Base  Compare

                First Obs           1        1
                First Unequal       1        1
                Last  Unequal       2        2
                Last  Obs           2        2

 Number of Observations in Common: 2.
 Total Number of Observations Read from WORK.CARS1: 2.
 Total Number of Observations Read from WORK.CARS2: 2.

 Number of Observations with Some Compared Variables Unequal: 2.
 Number of Observations with All Compared Variables Equal: 0.

            Value Comparison Results for Variables
_____________________________________________________________
             ||       Base    Compare
        Obs ||  CYLINDERS  CYLINDERS      Diff.     % Diff
  ________   ||  _________  _________  _________  _________
             ||
          1  ||          M          A
          2  ||                4.0000
_____________________________________________________________
```

Ist der Vergleich von Beobachtungen, in denen mindestens einer der beiden Werte fehlt, nicht relevant, so kann man diesen durch die Verwendung der Option NOMISS unterdrücken.

```
PROC COMPARE
   BASE=cars1
   COMP=cars2 NOMISS;
RUN;
```

Der resultierende (auszugsweise) Output

```
                    Observation Summary

                Observation      Base  Compare

                First Obs           1        1
                Last  Obs           2        2

  Number of Observations in Common: 2.
  Total Number of Observations Read from WORK.CARS1: 2.
  Total Number of Observations Read from WORK.CARS2: 2.

  Number of Observations with Some Compared Variables Unequal: 0.
  Number of Observations with All Compared Variables Equal: 2.

  NOTE: No unequal values were found. All values compared are exactly equal.
```

führt zu der „NOTE: No unequal values were found. All values compared are exactly equal.“ führt. Dennoch stimmen die Datensätze nicht exakt überein, weil die Beobachtungen mit missing values von dem Vergleich ausgeschlossen wurden.

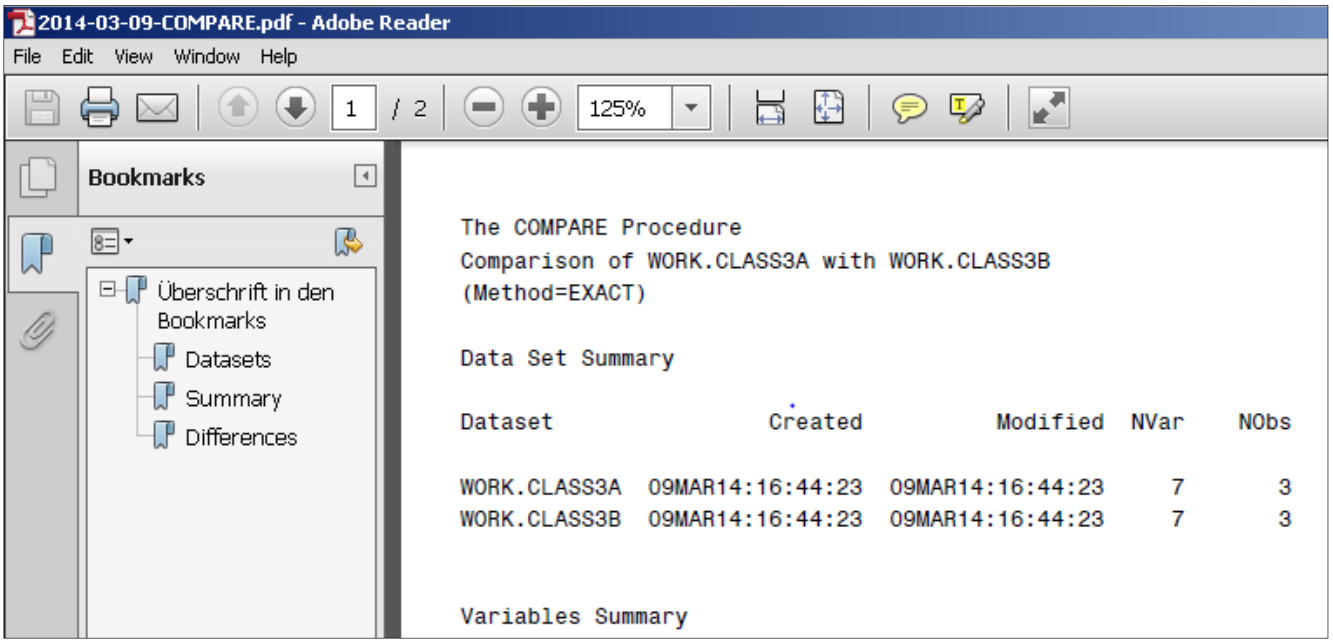
2.6 Output als PDF-file

Neben der Definition des Output Umfangs kann es auch sinnvoll sein, nicht den standardmäßigen Text- oder Datensatz-Output zu verwenden, sondern andere Output Formate zu wählen.
Für Dokumentationszwecke bietet sich die Ausgabe der Ergebnisse in einem PDF-file an, der mit ODS PDF leicht erzeugt werden kann. Der Programmcode

```
*Create macro variable with actual date in ISO8601 format;
DATA _null_;
  CALL SYMPUT ("actual",left(put("&sysdate"d,is8601da.)));
RUN;

ODS LISTING CLOSE;
*Use actual date in ISO8601 format in filename;
ODS PDF file="&actual._<filename>";
ODS PROCLABEL "Überschrift in den Bookmarks";
     PROC COMPARE
     BASE=class3a
     COMP=class3b;
     RUN;
ODS PDF CLOSE;
ODS LISTING;
```

erzeugt den folgende PDF file



Anhand der Bookmarks im linken Teil des PDF-files erhält man einen Überblick über den PROC COMPARE Output und ein schnelles Navigieren zwischen den verschiedenen Ergebnisabschnitten wird ermöglicht.
Mithilfe von ODS PROCLABEL können im Fall von mehreren PROC COMPARE Aufrufen in einem Programmlauf Bookmark Überschriften für die einzelnen Vergleiche definiert werden, was ebenfalls der Übersichtlichkeit des Outputs dient.

Bei regelmäßiger, z.B. monatlicher Erstellung des Outputs erleichtert die Verwendung des aktuellen Datums in ISO8601 Format im Filenamen die chronologische Sortierung der Dokumente in einem Ablage-Verzeichnis.

3 Ergebnisdarstellung außerhalb des Output-files oder Output Datensatzes

Während die in Kapitel 2 beschriebenen Beispiele sich im Wesentlichen mit dem Umfang und Layout des generierten Outputs beschäftigten, sowie einige Fallstricke bei der Interpretation aufzeigten, werden in diesem Kapitel weitere Ergebnisausgaben außerhalb des Output-files oder Datensatzes vorgestellt.

3.1 Ergebnisübersicht im Log-file

Um einen schnellen Überblick über das Vergleichsergebnis zu erhalten, können die Optionen ERROR oder WARNING nützlich sein, die eine kurze Ergebniszusammenfassung als ERROR oder WARNING in den Log-file schreiben.

Die folgende Ergänzung des Programmcodes aus Kapitel 2.1

```
PROC COMPARE
   BASE=class1a
   COMP=class1b LISTALL WARNING;
RUN;
```

führt aufgrund der unterschiedlichen Anzahl von Beobachtungen in den zu vergleichenden Datensätzen zu der Log-WARNING:

```
WARNING: Data set WORK.CLASS1A contains 2 observations not in WORK.CLASS1B.
WARNING: The data sets WORK.CLASS1A and WORK.CLASS1B do not contain the same
         data. One or both data sets contain variables or observations not in
         the other. However, all comparisons are equal for the data in common.
```

Mithilfe von Programmroutinen, die standardmäßig die Log-files nach möglichen WARNINGs oder ERRORs scannen, kann ohne Ansicht des Outputs schnell festgestellt werden, ob der Vergleich Unterschiede erkannt hat. Erst wenn entsprechende WARNINGs oder ERRORs aufgetreten sind, wird eine genauere Durchsicht des Outputs notwendig.

3.2 Macrovariable SYSINFO

Neben den Ergebnissen im Output oder als Datensatz liefert die automatische Macrovariable SYSINFO Informationen über das Ergebnis des durchgeführten Vergleichs.

Die nachfolgende Tabelle zeigt eine Auswahl der für die verschiedenen Teilaspekte des Vergleichs verwendeten Codes:

Tabelle 1: Codes für Teilergebnisse des Vergleichs (Auswahl)

Compare Ergebnis	**Code**
Keinerlei Unterschiede	0
Unterschiede bzgl. - Datensatzmetadaten* - Variablenmetadaten*	 1, 2 4, 8, 16, 32
Unterschiede bzgl. - nicht vergleichbarer Beobachtungen* - nicht vergleichbarer Variablen*	 64, 128 1024, 2048
Unterschiede bzgl. der Inhalte von Variablen	4096

*verschiedene Codes je nach Art des Unterschieds

Der Wert der Variablen SYSINFO ergibt sich aus der Code-Summe der jeweils im Vergleich aufgetretenen Unterschiede. Gibt es z.B. Unterschiede im Datensatzlabel (Code 1) und mindestens einen unterschiedlichen Variablenwert (Code 4096), so wäre SYSINFO= 4097 (1 + 4096).

SYSINFO kann zur Steuerung des weiteren Programmablaufs verwendet werden. Dabei muss allerdings beachtet werden, dass die Macrovariable mit Beginn des nächsten Data Steps oder Prozeduraufrufs überschrieben wird.

Der folgende Programmcode zeigt beispielhaft, wie der Programmablauf in Abhängigkeit vom Ergebnis des vorherigen Datensatzvergleichs gesteuert werden kann:

```
PROC COMPARE
     BASE=data1
     COMP=data2;
RUN;
%LET rc=&sysinfo.;
DATA _null_;
  %IF &rc >= 4096 %THEN
      PUT "Es gibt Beobachtungen mit unterschiedlichen Werten";
  %ELSE %DO;
        ....
  %END;
RUN;
```

So kann z.B. mit Hilfe von SYSINFO bei mehreren, hintereinander durchzuführenden Vergleichen der Umfang des Outputs verringert werden, indem nur Vergleiche ausgegeben werden, die Unterschiede identifiziert haben. Zunächst wird PROC COMPARE mit der Option NOPRINT (kein gedruckter Output) ausgeführt. Nur wenn SYSINFO einen Unterschied identifiziert, wird anschließend der komplette Output in einem weiteren PROC COMPARE Lauf erzeugt. Für Datensatzvergleiche ohne Unterschiede wird kein gedruckter Output generiert.

4 Programmiervorlage

Der folgende Programmcode kann als Grundlage für eine PROC COMPARE-Aufruf dienen. Er enthält sinnvolle und häufig verwendete Optionen und Statements, die bei der Anwendung von PROC COMPARE bedacht werden sollten:

```
PROC COMPARE BASE=<dataset1>  COMP=<dataset2>
     LISTALL /* TRANSPOSE */
     /* CRITERION=<0.00001> */
     /* NOMISS */
     /* OUT=<dataset3> OUTNOEQUAL OUTBASE OUTCOMP OUTDIF */
     WARNING /* ERROR */;
     ID <var1 … varn>;
RUN;
```

Durch Hinzufügen der entsprechenden ODS PDF Syntax (siehe Kap. 2.6) kann ein PDF-file des Outputs erzeugt werden.

5 Fazit

Die SAS® Prozedur PROC COMPARE ermöglicht mit geringem Aufwand den Vergleich von SAS® Datensätzen bezüglich Metadaten und Variableninhalten. Durch die gezielte Anwendung von Prozeduranweisungen und Optionen kann der Fokus des Vergleiches auf bestimmte Aspekte konzentriert werden und damit die Lesbarkeit und Übersichtlichkeit der Ergebnisse erhöht werden. Die hier vorgestellten Optionen und Statements stellen dabei nur einen Auszug der Möglichkeiten von PROC COMPARE dar.

Als Nutzer sollte man sich immer bewusst sein, dass durch die Wahl einschränkender Optionen Teilaspekte nicht dargestellt werden, die vielleicht doch von Relevanz sein könnten, aber deren Auftreten nicht erwartet wurde. Eine gute Kenntnis der zu vergleichenden Daten ist also ebenso unerlässlich wie eine gezielte Anwendung der verschiedenen Optionen.

Wenn der inhaltliche Vergleich der Variablen von Interesse ist, dann sollte bei der Note „No unequal values found. All values compared exactly equal“ immer überprüft werden, ob alle Daten wie erwartet verglichen wurden, oder eventuell einige Beobachtungen oder Variablen vom Vergleich ausgeschlossen waren.

Desweiteren sollte man bedenken, dass Unterschiede in Zeichenketten nach der zwanzigsten Stelle zwar erkannt, im Output aber nicht dargestellt werden. Hier muss immer auf die Generierung eines Output Datensatzes mit anschließender Ausgabe zurückgegriffen werden.

Insgesamt lässt sich sagen: ja, PROC COMPARE macht den Vergleich von SAS®-Datensätzen leicht - aber nur wenn man die passenden Optionen einsetzt oder kombiniert und das Ergebnis korrekt interpretiert.

Literatur

Informationen zur Prozedur PROC COMPARE wurde dem Base SAS® 9.3 Procedures Guide, Second Edition entnommen.
(https://support.sas.com/documentation/cdl/en/proc/65145/HTML/default/viewer.htm#n1nwxbchh5hpu1n1h28kmici2awd.htm) [12.03.2014]

Weitere Literatur:

[1] Lara E.H. Guttadauro: VALIDATION:
Let SAS® do the comparisons for you. PharmSUG Proceedings 2001, Boston.

[2] B. Hientzsch, G. Lückel, J. Müller, N. Tambascia:
Unabhängige Doppelprogrammierung – das non plus ultra der Validierung?
Proceedings der 17. KSFE Ulm. Shaker-Verlag, 2013.