# Ersetzung fehlender Datums- und Zeitangaben im Standardformat ISO 8601

Matthias Lehrkamp
Bayer AG
Müllerstraße 178
13353 Berlin
matthias.lehrkamp@bayer.com

**Zusammenfassung**

Der ISO 8601 Standard beschreibt verschiedene Standardformate zur Darstellung von Datumsformaten und Zeitangaben für Computersysteme, herausgegeben von der International Organization for Standardization (ISO). Das ISO 8601 Format eignet sich besonders, um Weltweit Kalenderdaten auszutauschen. Das größte Manko dieses Formates ist jedoch, dass der Bindestrich gleich zwei Bedeutungen hat. Es findet Verwendung als Trennzeichen zwischen zwei Zeitelementen und als Ersatzzeichen für fehlende Zeitangaben. Das wiederum erschwert das Auslesen der einzelnen Zeitelemente. In dieser Präsentation wird ein simpler Weg aufgezeigt, um die einzelnen Zeitelemente auszulesen und fehlende Elemente, mit den frühestmöglichen oder letztmöglichen Zeitpunkt zu ersetzen. Nach diesem Beitrag haben sie garantiert einen sicheren Lösungsweg, um fehlende Angaben im ISO 8601 formatiertem Datum mit Uhrzeitangabe zu ersetzen.

**Schlüsselwörter:** ISO 8601, date, datetime, regulärer Ausdruck, Perl Regular Expression, Makro erstellen

## 1 ISO 8601

Die „International Organization for Standardization“ (ISO) gibt weltweit geltende Standards heraus, auch Normen genannt. Darunter fällt auch die Norm ISO 8601 „Data elements and interchange formats - Information interchange - Representation of dates and times“, diese beschreibt die Darstellung von Datums- und Urzeitangaben. Der 28. Februar 2018 wird laut Norm mit 2018-02-28 angeben, die Uhrzeit (Sekundengenau) mit 15:16:12 und beides Zusammen mit 2018-02-28T15:16:12.

Seit einiger Zeit ist eine neue Version in der Entwicklung, die in zwei Teile (Part 1 und Part 2) herauskommen soll. Vom 2016-12-30 bis 2017-03-23 gab es eine Vorabversion zum Review, bisher (Stand 2018-02-12) gab es jedoch keine Veröffentlichung dieser Versionen. Der erste Teil (Part 1) beschreibt die grundlegenden Konzepte und Prinzipien. Im zweiten Teil werden Erweiterungen der Standardformate beschrieben. Laut Wikipedia wurde das Datumsformat in die DIN EN 28601 in Deutschland am 1. Mai 1996 übernommen und war somit das einzige normgerechte numerische Datumsformat. Da viele Menschen im Alltag weiterhin das alte Format nach DIN 1355-1 benutzten,

wurde dieses Format über eine Neuregelung der DIN 5008 im Jahr 2001 wieder zugelassen. Beide Angaben sind somit heute gültig.

Bei klinischen Daten haben sich die Datenstrukturen, wie von der „Clinical Data Interchange Standards Consortium“, kurz CDISC, vorgeschrieben, durchgesetzt. CDISC verlangt bei der Speicherung der klinischen Daten für alle Datumsangaben das Format nach ISO 8601. Für alle Studien, die nach dem Datum 2016-12-17 gestartet sind, müssen die Daten bei der amerikanischen Behörde „FDA“ in der CDISC Struktur eingereicht werden. Es kann also davon ausgegangen werden, das in Zukunft alle einzureichenden Studien den ISO 8601 Standard folgen.

Dieser Beitrag konzentriert sich ausschließlich auf den Umgang mit dem Datumsformat (2018-02-28) und dem Datumsformat mit sekundengenauer Uhrzeitangabe (2018-02-28T15:16:12). Im Folgendem eine kurze Erklärung der Zusammensetzung der Zeichen.

**Zeichenerklärung:**

Standardformat: *YYYY-MM-DD*T*hh*:*mm*:*ss*

Zeitkomponenten
*Y*: Jahr (eng. year), 4 Ziffern, kann aber erweitert werden
*M*: Monat (engl. month), Zahl zwischen 01 und 12
*D*: Tag (engl. day), Zahl zwischen 01 und 31
*h*: Stunden (engl. hours), Zahl zwischen 00 und 23
*m*: Minuten (engl. minutes), Zahl zwischen 00 und 59
*s*: Sekunden (engl. seconds), Zahl zwischen 00 und 59

Sonstige Zeichen

- \- Trennzeichen zwischen den Zeitkomponenten Jahr, Monat und Tag, sowie als Ersatzzeichen für fehlende Angaben
- T Trennzeichen zwischen Datum und Uhrzeit
- : Trennzeichen zwischen den Zeitkomponenten Stunden, Minuten und Sekunden

Die Angabe der Zeitkomponenten von der größten zur kleinsten Zeiteinheit entspricht der üblichen mathematischen Schreibweise. Im Zusammenspiel mit der festen Ziffernanzahl jeder Zeitkomponente, sortieren sich die Zeitpunkte automatisch vom frühesten bis zum spätesten Zeitpunkt. Wobei fehlende Angaben sich vor den vollständigen einsortieren.

2017-03-09
2017-03-10
2017-03-12
2018-02
2018-02--T14:00
2018-02-28
2018-03-01

SAS verwendet unterschiedliche Methoden, wie das Datum mit oder ohne Zeitangabe behandelt wird. Dies ist in der Tabelle 1 kurz zusammengefasst.

**Tabelle 1:** Übersicht für das Datum und dem Datum mit Zeitangabe

| | **Datum** | **Datum mit Urzeit** |
|---|---|---|
| Allgemeines Format | *YYYY-MM-DD* | *YYYY-MM-DD*T*hh*:*mm*:*ss* |
| Vollständige Daten | 2017-05-30 | 2017-05-30T19:59:30 |
| Mit fehlenden Angaben<br>• Am Ende<br><br>• Vor dem Ende | <br>2017<br>2017-05<br>2017---15<br>----30 | <br>2017-05-30T19<br>2017-05-30T19:59<br>2017-05-30T-:59:30<br>2017-05--T-:59:30 |
| SAS In-/Formate<br>*w*: gesamte Zeichenlänge<br>*d*: Anzahl der Zahlen für die optionalen Sekundenbruchteile [0-6] | is8601da*w*.<br>e8601da*w*.<br>b8601da*w*. (ohne Trennzeichen -) | is8601dt*w*.*d*<br>e8601dt*w*.*d*<br>b8601dt*w*.*d* (ohne Trennzeichen - und :) |
| SAS numerischer Wert | Tage nach dem<br>1960-01-01<br>1=1960-01-02 | Sekunden nach<br>1960-01-01T00:00:00<br>1=1960-01-01T00:00:01 |

Bei Formaten, die mit dem Datum ohne Zeitangabe arbeiten, gibt der numerische Wert die vergangenen Tage nach dem 1. Januar 1960 an. Die 1 entspricht somit dem 2. Januar 1960, da am diesem Tag bereits 1 Tag seit dem 1. Januar 1960 vergangen ist. Im Unterschied zum Datum ohne Uhrzeitangabe, gibt der numerische Wert eines Datums mit Uhrzeitangabe die vergangenen Sekunden nach dem 1. Januar 1960, 00:00:00 Uhr an. Die 1 wäre somit 1 Sekunde nach Mitternacht des 1ten Januars 1960. Eine weitere Besonderheit tritt für das Jahr bei der Verwendung von SAS Formaten auf. Die Formate für das Datum mit oder ohne Zeitangabe sind auf eine Jahreszahl mit 4 Ziffern beschränkt, wobei die kleinste Jahreszahl der Beginn des gregorianischen Kalenders (1582) darstellt. Die Norm ISO 8601 erlaubt auch negative Jahreszahlen und kann mehr als 4 Ziffern verwenden. Da die Analyse der Daten numerische Werte benötigt, ist das Jahr auf das Intervall [1582-9999] in SAS beschränkt.

Bei dem Vergleich von zwei Zeitpunkten ist es erforderlich, fehlende Zeitkomponenten zu ersetzen. Die einfachste Methode ist die Bildung eines Zeitintervalls von dem frühestmöglichen Zeitpunkt bis zum spätmöglichsten Zeitpunkt. Beispielsweise kann das Datum 2018-02 mit einem Intervall [2018-02-01, 2018-02-28] ersetzt werden.

Das nachfolgende Kapitel beschreibt eine sichere Möglichkeit, die einzelnen Zeitkomponenten in SAS auszulesen, um danach das mögliche Zeitintervall zu bilden. Da das Problem in fast jeder Studie auftritt, wird die Lösung als Makro umgesetzt. Somit ist es flexibel anwendbar und kann wiederholt angewendet werden.

# 2 Der einfache Umgang mit vollständigen Daten

Bei vollständigen Daten bietet SAS verschiedene Möglichkeiten die Daten ins Numerische umzuwandeln oder auch gezielt auf einzelne Zeitkomponenten zuzugreifen. Beispielsweise bietet SAS verschiedene Informate an, um ein Datum mit oder ohne Zeitangabe vom Text ins Numerische zu konvertieren (siehe Tabelle 1). Weiterhin kann die SCAN Funktion benutzt werden, um bestimmte Zeitkomponenten zu extrahieren.

```
DATA dates;
  x1= "2017-05-30";
  x2= "2017-05-30T19:59:30";
  * conversion to numeric values;
  y1= input(x1,is8601da.);
  y2= input(x2,is8601dt.);
  PUT "1) " y1= y2=; * print numeric values;
  PUT "2) " y1= is8601da. y2= is8601dt.; * print formatted values;
  * get specific time components, z1 is month and z2 is hour;
  z1= input(scan(x1,2,'-'),4.);
  z2= input(scan(x2,4,'-T:'),4.);
  PUT "3) " z1= z2=;
RUN;
```

Log:

```
1) y1=20969 y2=1811793570
2) y1=2017-05-30 y2=2017-05-30T19:59:30
3) z1=5 z2=19
```

Bei unvollständigen Daten können die zuvor gezeigten Methoden leider nicht angewendet werden. Die Formate unterstützen fehlende Zeitkomponenten nicht und die SCAN Funktion ist ebenfalls nicht anwendbar, da der Bindestrich doppelt belegt ist. Dieser wird sowohl als Trennzeichen, als auch für fehlende Werte eingesetzt. Die SUBSTR Funktion kommt bei fehlenden Zeitkomponenten leider auch nicht in Frage, da sich die Positionen der Zeitkomponenten, durch die Ersetzung mehrerer Ziffern durch einen Bindestrich, verändern. Sollen alle Möglichkeiten in Betracht gezogen werden, ist es sinnvoll einen regulären Ausdruck (in SAS Perl Regular Expression) zu verwenden.

Der nächste Abschnitt beschäftigt sich mit einem Anwendungsbeispiel aus dem klinischen Bereich. Im Anschluss wird dann die Lösung mit SAS in einem Makro umgesetzt.

# 3 Anwendungsbeispiel aus dem klinischen Umfeld

Enno ist Proband in einer klinischen Studie zum Wirksamkeitsnachweis eines Medikaments zur Behandlung von akuten Bronchitisanfällen. Der Arzt fragt den Probanden, ob es im letzten halben Jahr irgendwelche Beschwerden gab. Unser Proband gab an, dass er einen halben Tag lang starke Kopfschmerzen hatte. Er wisse genau, dass das Ereignis am 15. eines Monats war, da er immer genau am 15. eines Monats seinen Lohn bekom-

me. Enno erinnere sich aber nicht mehr an den genauen Monat. Dem Arzt bleibt nichts anderes übrig, als den Monat als fehlend einzutragen. Auf dem Fragebogen wird also 2017---15 notiert.

Die Daten auf den Fragebögen werden in eine Datenbank übertragen, anonymisiert und irgendwann gelangen die Daten zu unserer Statistikerin Teresa. Schon nach kurzer Zeit fällt ihr das kuriose Datum von dem Ereignis auf. Sie stellt eine Rückfrage an den behandelnden Arzt, um den fehlenden Monat bei gegebenem Tag zu bestätigen. Der Arzt bestätigt das Datum und gibt an, aus welchen Gründen das Datum so festgehalten wurde. Teresa muss also mit dem gegebenen Datum weiterarbeiten. Jetzt gilt es zu beantworten, ob die starken Kopfschmerzen etwas mit dem neuen Medikament zu tun haben können? Das Medikament wurde am 2017-12-17 das erste Mal vom Patienten eingenommen. Teresa überlegt sich eine Lösung, wie sie mit dem Datum zurechtkommt. Sie notiert sich das frühestmögliche Datum 2017-01-15 und das spätmöglichste Datum 2017-12-15. Jetzt kann sie überprüfen, ob die Medikamentengabe in dem möglichen Intervall der starken Kopfschmerzen stattfand, beziehungsweise kurz davor gegeben wurde.

**Tabelle 2:** Beispieldaten ersetzen

| | Erste Medikation | Start der Beschwerden | |
|---|---|---|---|
| Original wert | 2017-12-17 | 2017---15 | |
| Ersetzung für die Analyse | 2017-12-17 | 2017-01-15 | 2017-12-15 |

Die Medikation wurde erstmals nach dem möglichen Start der Beschwerden genommen, dadurch steht ganz sicher fest, dass das Medikament nicht mit den starken Kopfschmerzen im Zusammenhang steht.

# 4 Lösung als SAS Makro

In diesem Abschnitt soll eine Lösung zur Ersetzung fehlender Zeitkomponenten entwickelt werden. Die Lösung wird dabei als SAS Makro umgesetzt. Das Makro selbst soll splitIso8601Dtc heißen. Es soll die einzelnen Zeitkomponenten in verschiedenen Variablen abspeichern, fehlende Zeitkomponenten ersetzen und das Ergebnis als numerisches Intervall abspeichern.

Bevor es mit der Programmierung losgeht, sollte eine Makro Spezifikation geschrieben werden. Folgende Punkte sind festzuhalten.

- Beschreibung des Makros, was macht es und welche Besonderheiten gibt es?
- In welcher Systemumgebung soll es laufen (Betriebssystem, SAS Version, …)?
- Welche Ein- und Ausgaben gibt es?
- Welche Fehlerchecks müssen implementiert werden, damit das Makro fehlerfrei läuft (Parameterchecks, Systemumgebung prüfen, …)?

- Den Programmablauf entweder in einem Ablaufdiagramm oder in schriftlicher Form beschreiben.

Nachdem das Makro spezifiziert ist, kann mit der Programmierung angefangen werden. Zu Beginn empfiehlt es sich die Programmierung auf einen bestimmten Spezialfall so weit zu beschränken, dass vorerst ohne Macroschleifen (%IF … %THEN …) oder Makrobedingungen (%DO …) gearbeitet werden kann. Der Vorteil dabei ist, dass der Code ausführbar ist und das Ergebnis schneller betrachtet werden kann. Wenn beispielsweise ein Makro alle Laborparameter auswerten soll, wird zunächst die Lösung für nur einen ausgewählten Parameter programmiert. Dadurch wird die Programmierung bedeutend einfacher. Damit die spätere Umwandlung zu einem SAS Makro schnell möglich ist, sollten die Makroparameter als Makrovariablen festgelegt werden und ausschließlich im Programmcode verwendet werden.

| 1. Schritt: Programmieren eines Beispielszenarios mit ausführbaren SAS Code (ohne Verwendung von Makrobedingungen und Makroschleifen). Die geplanten Makroparameter werden dabei als Makrovariablen festgelegt und ausschließlich im Code benutzt. |
|---|

Für das Makro splitIso8601Dtc sind folgende Makroparameter geplant.

**Tabelle 3:** Makroparameter des Makros splitIso8601Dtc

| **Makroparameter** | **Funktion** |
|---|---|
| inDat | Eingangsdatensatz |
| outDat | Ausgangsdatensatz |
| Var | Character-Variable deren Datum im ISO 8601 Format in die einzelnen Zeitkomponenten getrennt werden soll. |

Mit den gegebenen Makroparametern ergibt sich für den Anfang folgender Programmcode.

```
%LET inDat= ae;
%LET outDat= ksfe_2018;
%LET var= aestdtc;

DATA &outDat;
  SET &inDat;
  ...
```

Das erste zu lösende Problem ist, eine sichere Methode zu finden, die die einzelnen Zeitkomponenten auslesen kann. Da die Textfunktionen %SCAN und %SUBSTR aus bereits erwähnten Gründen nicht anwendbar sind, muss eine andere Lösung gefunden werden. Die wohl sicherste Methode stellt die Verwendung von regulären Ausdrücken dar.

## 4.1 Regulärer Ausdruck zum Auslesen der einzelnen Zeitkomponenten

Der reguläre Ausdruck beschreibt die Zeichenkette über syntaktische Regeln. SAS bietet im Basispaket die Perl Regular Expression an. Eine gute Übersicht über die Anwendung und den syntaktischen Regeln enthält das Tip Sheet von SAS „SAS 9 - Perl Regular Expressions Tip Sheet“. Im Folgenden soll kurz der Ausdruck zum Auslesen der einzelnen Zeitkomponenten erläutert werden.

Der reguläre Ausdruck wird stets in Schrägstriche /…/ eingeschlossen. Der Anfang eines Variablenwertes kann mit einem Dach ^ festgelegt werden, das Ende mit einem Dollarzeichen $. Sollen die Buchstaben „en“ überall im Text gefunden werden, so werden beide Zeichen ^ und $ weggelassen. Sollen dagegen die Buchstaben nur am Ende stehen, muss /en$/ als Ausdruck verwendet werden. In unserem Fall wollen wir den zu suchenden Text, das ISO 8601 formatierte Datum, vom Beginn bis zum Ende festlegen. Am Ende können aufgrund der Variablenlänge ein oder mehrere Leerzeichen auftreten. Das Leerzeichen wird mit \s abgebildet, das Auftreten beliebig vieler Zeichen [0,*n*] wird mit einem Stern * gekennzeichnet. Damit sieht der Beginn des Ausdrucks wie folgt aus.

```
/^…\s*$/
```

Gruppen werden durch runde Klammern (…) gekennzeichnet. Mit Hilfe von Gruppen können Textelemente zusammengefasst werden. Auf den Inhalt einer Gruppe kann direkt zugegriffen werden. Gruppen können auch als optional definiert werden. Für das ISO 8601 Format werden die Gruppen benutzt, um auf die einzelnen Zeitkomponenten zuzugreifen. Zusätzlich werden weitere Gruppen benötigt, um die Zeitkomponenten als optional festzulegen. Unser erstes Element ist das Jahr, bestehend aus 4 Ziffern \d. Die Anzahl eines Elements kann über geschweifte Klammern festgelegt werden. Dabei kann gezielt ein Intervall {*n*,*m*} bestimmt werden, eine Mindestanzahl {n,} oder Höchstanzahl {,m} oder eine genaue Anzahl {n} festgelegt werden. Der vorhergehende Ausdruck wird um (\d{4}) ergänzt, welches das Jahr als nicht optionale Gruppe darstellt.

```
/^(\d{4})…\s*$/
```

Der Trennstrich - und der Monat, bestehend aus zwei Ziffern \d{2} oder einen Bindestrich - (fehlende Angabe), sind optional. Ein optionales Element wird mit einem Fragezeichen gekennzeichnet. Eine Auswahl zwischen zwei Elementen wird über den senkrechten Strich A|B angezeigt. Für die Darstellung des Monats werden also zwei Gruppen benötigt. Die erste Gruppe kennzeichnet, dass der Trennstrich und der Monat optional sind (…)?. Innenliegend befindet sich der Trennstrich und die nächste Zeitkomponente, der Monat (-(\d{2}|-))?.

```
/^(\d{4})(-(\d{2}|-)…)?\s*$/
```

Alle weiteren Zeitkomponenten werden jetzt genauso hinzugefügt, es ändern sich lediglich die Trennzeichen.

/^(\d{4})(-(\d{2}|-)(-(\d{2}|-)(T(\d{2}|-)(:(\d{2}|-)(:(\d{2}))?)?)?)?)?\s*$/

Beachtet werden sollte, dass bei diesem Ausdruck das Datum mit einem Bindestrich enden kann, laut ISO 8601 muss aber das Datum, mit oder ohne Uhrzeit, auf eine Zahl enden. Das kann aber als zusätzliches Feature in der Makrospezifikation verkauft werden oder es wird im Makro über eine weitere Abfrage eine Fehlermeldung erzeugt.

Der reguläre Ausdruck wird der Funktion PRXPARSE übergeben, die diesen dann kompiliert und eine Pattern ID für die Verwendung in anderen PRX-Funktionen zurückgibt. Mit der Funktion PRXMATCH wird zunächst überprüft, ob der aktuelle Variablenwert zu dem Ausdruck passt. Danach können die einzelnen Zeitkomponenten ausgelesen werden, indem über die Funktion PRXPOSN auf die jeweilige Gruppe zugegriffen wird. Für die Umwandlung des Textwertes in eine Zahl wird die Funktion INPUT verwendet. Damit keine Fehler bei fehlenden Werten ausgegeben werden, wird der Bindestrich über die Funktion COMPRESS gelöscht.

```
* set regular expression ID for is8601dt format;
prxid= prxparse('/^(\d{4})(-(\d{2}|-)(-(\d{2}|-)(T(\d{2}|-
)(:(\d{2}|-)(:(\d{2}))?)?)?)?)?\s*$/');
* extract the date/time parts;
IF prxmatch(prxid,&var) THEN DO;
  &var._year= input(compress(prxposn(prxid,1,&var),'-'), 4.);
  &var._min_month=input(compress(prxposn(prxid,3,&var),'-'), 2.);
  &var._min_day= input(compress(prxposn(prxid,5,&var),'-'), 2.);
  &var._min_hour= input(compress(prxposn(prxid,7,&var),'-'), 2.);
  &var._min_minute= input(compress(prxposn(prxid,9,&var),'-'), 2.);
  &var._min_second= input(compress(prxposn(prxid,11,&var),'-'), 2.);
  ... Weitere SAS Statements
```

Die Funktion PRXPOSN bekommt die Pattern ID des regulären Ausdrucks, die Gruppe, welche ausgelesen werden soll und den Eingabewert. Die auszulesende Gruppe stimmt mit der Position der zu lesenden Zeitkomponente nicht überein. Das liegt daran, dass in dem regulären Ausdruck zusätzliche Klammern verwendet werden, um optionale Komponenten zu kennzeichnen. Im Folgendem eine Übersicht dazu.

```
Gruppe:    1         3         5         7         9         11
       /^(\d{4})(-(\d{2}|-)(-(\d{2}|-)(T(\d{2}|-)(:(\d{2}|-)(:(\d{2}))?)?)?)?)?\s*$/
Eingabe: 2018  -  03    -  01    T  14    :  35    :  15
```

## 4.2 Ersetzen fehlender Zeitkomponenten

Nachdem die Werte der einzelnen Zeitkomponenten in verschiedenen Variablen abgelegt sind, können die fehlenden Komponenten ersetzt werden. Die Tabelle 4 gibt die

Werte an, die für die Bildung des frühestmöglichen Zeitpunktes (Minimum) und des spätmöglichen Zeitpunktes (Maximum) einzusetzen sind.

**Tabelle 4:** Ersetzung fehlender Werte

| | **Früheste(s) Datum/Zeit** | **Späteste(s) Datum/Zeit** |
|---|---|---|
| **Monat** | 1 | 12 |
| **Tag** | 1 | letzter Tag des Monats im gegebenen Jahr |
| **Stunden** | 0 | 23 |
| **Minuten** | 0 | 59 |
| **Sekunden** | 0 | 59 |

Die Ersetzung fehlender Werte beginnt von der größten bis zur kleinsten Zeiteinheit. Bis auf den letzten Tag des Monats, können alle fehlenden Werte, wie in der Tabelle aufgeführt über eine entsprechende IF THEN Abfrage, ersetzt werden.

```
* impute month when missing;
IF missing(&var._min_month) THEN DO;
   &var._min_month= 1;
   &var._max_month= 12;
END;
ELSE DO;
   &var._max_month= &var._min_month;
END;
```

Da die Daten bereits in den Variablen eingelesen wurden, die das Minimum beinhalten soll (`&var._min_month`), müssen vorhandene Werte auch für das Maximum übernommen werden. Um den letzten Tag des Monats zu ermitteln, eignet sich die Funktion INTNX. Diese Funktion beachtet automatisch den Monat und die Schaltjahre. Komplizierte Algorithmen sind also nicht nötig.

```
* impute day when missing;
IF missing(&var._min_day) THEN DO;
  &var._min_day= 1;
  * Last day of the respective month and year;
  &var._max_day= day(intnx('month'
                          ,input(cats(put(&var._year,z4.),'-'
                                    ,put(&var._max_month,z2.),'-01')
                                         ,is8601da.)
                          ,0
                          ,'E'));
END;
ELSE DO;
   &var._max_day= &var._min_day;
END;
```

Die INTNX Funktion benötigt folgende Eingaben. Als ersten Parameter muss der Zeittyp (`'month'`) angegeben werden. Der zweite Parameter muss ein numerisches SAS Datum sein. Da der Tag noch fehlt, wird zunächst der 1. Tag des gegebenen oder bereits

ersetzten Monats (Maximum) und des gegebenen Jahres erzeugt. Über den 3. Parameter kann eine Zeitverschiebung in Tagen übergeben werden. Diese Funktionalität wird aber in diesem Fall nicht benötigt, weshalb die Verschiebung auf 0 gesetzt wird. Über dem letzten Parameter (optional) kann eine Justierung des Datums vorgenommen werden. Unter anderem kann hier das `'E'` übergeben werden, um zum letzten Tag des Monats zu springen. Mit der Funktion DAY wird nur der Tag des Datums in unserer Variable übertragen.

Alle anderen Zeitkomponenten können, wie an dem Beispiel des Monats gezeigt, entsprechend ersetzt werden. Somit sind alle Daten ausgelesen und die fehlenden Zeitkomponenten sind bereits ersetzt. Die Daten liegen auch in zweifacher Ausführung vor, nämlich das Minimum und das Maximum des Eingangsdatums mit Uhrzeit. Im nächsten Kapitel werden die einzelnen Zeitkomponenten zu einem numerischen Datum mit Uhrzeit zusammengefasst.

## 4.3 Speichern des Datums als numerischen Wert

Für die Umwandlung der einzelnen Werte in ein numerisches SAS Datum mit Uhrzeit werden zwei Funktionen verwendet. Die Funktion DHMS benötigt ein numerisches Datum und die einzelnen Komponenten der Uhrzeit als Eingabe. Somit können die Stunden, Minuten und Sekunden direkt zugeordnet werden. Das Datum kann mit der Funktion MDY erzeugt werden, das wiederum die einzelnen Komponenten des Datums, in der Reihenfolge Monat, Tag und Jahr, zugewiesen bekommt.

```
* create the full date/time value;
FORMAT &var._min &var._max is8601dt.;
&var._min= dhms(mdy(&var._min_month, &var._min_day, &var._year)
                ,&var._min_hour, &var._min_minute, &var._min_second);
&var._max= dhms(mdy(&var._max_month, &var._max_day, &var._year)
                ,&var._max_hour, &var._max_minute, &var._max_second);
```

Als Ergebnis ergibt sich ein numerisches Datum mit Uhrzeit für die frühestmögliche Zeit (Minimum), und ein weiteres numerisches Datum mit Uhrzeit für die spätmöglichste Zeit (Maximum).
Der Programmcode kann jetzt ausgiebig getestet und angepasst werden. Im nächsten Schritt wird der Code dann in ein SAS Makro umgewandelt.

## 4.4 Umwandlung in ein SAS Makro

Sofern stets die zuvor angelegten Makrovariablen benutzt wurden, ist die Umwandung hin zu einem SAS Makro schnell umsetzbar. Die Definition der Makrovariablen wird einfach in Makroparameter umgeschrieben und das zuvor festgelegte Beispiel kann sogleich als Testaufruf für das Makro verwendet werden.

```
%LET inDat= ae;
%LET outDat= ksfe_2018;
%LET var= aestdtc;
```

```
%MACRO splitIso8601dtc( inDat=
                      , outDat=
                      , var=
        ) / DES='separate datetime components';
```

```
DATA &outDat;
  SET &inDat;
  ...
```

```
DATA &outDat;
  SET &inDat;
  ...

%MEND splitIso8601dtc;

%splitIso8601dtc( inDat= ae
                , outDat= ksfe_2018
                , var= aestdtc )
```

Der Testaufruf sollte zum gleichen Ergebnis führen, wie der Programmcode vorher.

2. Schritt: Den Programmcode in ein Makro umwandeln. Dabei werden die Makrodefinitionen als Eingangsparameter verwendet und das gewählte Szenario als Beispielaufruf verwendet.

In einem nächsten Schritt sollten die Eingabeparameter und weitere Checks innerhalb des Makros eingebaut werden.

```
%MACRO splitIso8601dtc( inDat=
                      , outDat=
                      , var=
      ) / DES='separate datetime components';

  /* Parameter checks */
  %IF %length(&inDat)=0 %THEN %DO;
    %PUT ERROR: Parameter inDat must be filled.;
    %ABORT;
  %END;
  ...

  DATA &outDat;
    SET &inDat;
    ...

%MEND splitIso8601dtc;
```

Zusätzlich sollten weitere Testaufrufe ausprobiert werden. Sobald die Funktionsweise des bisherigen Codes sichergestellt ist, kann damit begonnen werden, Makroschleifen (%DO …) und Makrobedingungen (%IF … %THEN …) hinzuzufügen. Ab jetzt lässt sich das Makro beliebig erweitern. Für die Datumsersetzung wäre es beispielsweise machbar ein Makroparameter hinzufügen, der eine optionale Verwendung der Sekundenangabe ermöglicht.

3. Schritt: Programmcode zum Überprüfen der Eingangsparameter hinzufügen. Weitere Checks und/oder Erweiterungen unter ständiges Testen des Makros hinzufügen.

# 5 Zusammenfassung

Dieser Beitrag hat sich auf die Verwendung des ISO 8601 formatierten Datums mit Uhrzeitangabe im klinischen Umfeld konzentriert. Der Schwerpunkt lag dabei darauf, eine möglichst wiederverwendbare Lösung zu finden, um ISO 8601 formatierte Daten auszulesen und fehlende Angaben zu ersetzen. Dabei mussten mehrere Stolperfallen mit Hilfe eines Programmcodes gelöst werden, der die größtmögliche Sicherheit verspricht. Das Auslesen der einzelnen Zeitkomponenten wurde dabei über einen regulären Ausdruck realisiert, der wiederum den Input syntaktisch beschreibt. Dadurch wird sichergestellt, dass nur Daten bearbeitet werden, die der syntaktischen Beschreibung genügen. Die Sicherheit liegt also in dem gewählten regulären Ausdruck. Eine weitere Schwierigkeit ist das bestimmen des letzten Tages eines Monats. Wer hierfür die interne SAS Funktion INTNX verwendet, ist damit auf der sicheren Seite. Diese Funktion berücksichtigt automatisch den Monat und die Schaltjahre. Die Umwandlung aus den einzelnen Zeitkomponenten zu einem numerischen Datum mit Uhrzeit wird mit den SAS Funktionen DHMS und MDY abgedeckt.

Das Thema Makroentwicklung wurde im Zusammenhang mit der Lösungsfindung einleitend beschrieben. Es gibt sicherlich verschiedene Methoden ein Makro zu entwickeln, aber die vorgestellte Methode bietet vor allem Anfängern eine hohe Erfolgsquote. Ein wesentlicher Punkt der Makroentwicklung ist die Vorbereitung des Makros durch eine schriftliche Spezifikation. Der Programmierstart beginnt mit der Programmierung eines ausgewählten Szenarios, über Makrovariablen unter Verwendung eines ausführbaren SAS Code. Später wird dieser Code in einem Makro umgewandelt. Ab diesem Zeitpunkt können weiterführende Parameterchecks und Erweiterungsmöglichkeiten hinzugefügt und ausgiebig getestet werden.

## Literatur

[1] ISO, „Date and time format - ISO 8601“, https://www.iso.org/iso-8601-date-and-time-format.html

[2] FDA, „STUDY DATA STANDARDS: WHAT YOU NEED TO KNOW”, https://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM511237.pdf

[3] SAS, “SAS 9 - Perl Regular Expressions Tip Sheet”, https://support.sas.com/rnd/base/datastep/perl_regexp/regexp-tip-sheet.pdf