

Consort-Diagramm der ICH E3 mit Hilfe von SAS Annotate

Murat Ipek
PAREXEL International
Spandauer Damm 130
14050 Berlin
Murat.Ipek@PAREXEL.com

Zusammenfassung

Die International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use (ICH) erarbeitet in einem mehrstufigen Verfahren einheitliche, empfehlende Guidelines für die Bewertung der Qualität, der Wirksamkeit und Unbedenklichkeit von Arzneimitteln. Von großer Bedeutung ist das vereinheitlichte Format für die Einreichung von Zulassungsunterlagen, insbesondere von Studienberichten, die in der ICH-Guideline E3 „Structure and Content of Clinical Study Reports“ verankert ist. Im Anhang IVa und IVb der ICH-E3-Guideline finden sich unter dem Titel „Disposition of Patients“ sogenannte Consort-Diagramme. Diese Diagramme bestehen in der Regel aus einer grafischen Übersicht, die die Verteilung der Studienteilnehmer übersichtlich darstellt. Dieser Beitrag zeigt, wie Sie mit Hilfe von SAS und SAS/GRAPH Annotate ein Consort-Diagramm als Grafik produzieren können.

Schlüsselwörter: SAS/GRAPH Annotate, Annotate-Datei, %ANNOMAC, PROC GSLIDE, Consort-Diagramm, Patienten-Population, Analyse-Population

1 Einführung

Ziel dieses Beitrags ist es, mit Hilfe der SAS/GRAPH Annotate Facility ein Consort-Diagramm zu erstellen, das die Verteilung der Patienten in die Analyse-Populationen sowie in die Behandlungsgruppen grafisch darstellt. SAS/GRAPH Annotate erlaubt die freie Grafikprogrammierung, um entweder die Ausgabe der Grafik-Prozeduren mit Elementen wie Text oder Symbolen zu ergänzen oder vollständig eigene Grafiken zu erstellen [1].

1.1 Problemstellung

Eine klinische Studie wird mit Patienten oder gesunden Probanden durchgeführt, mit dem Ziel, wissenschaftliche Fragestellungen zu beantworten und die medizinische Behandlung zu verbessern. Sie sind, bedingt durch die problematische Vergangenheit, sehr stark reglementiert. Um beispielsweise eine klinische Studie in Deutschland durchführen zu können muss der pharmazeutische Hersteller bei der Europäischen Arzneimittelagentur (EMA) diese beantragen. Des Weiteren muss die Bundesoberbehörde, in Deutschland ist es das Bundesinstitut für Arzneimittel und Medizinprodukte (BfArM) oder das Paul-Ehrlich-Institut (PEI), die Durchführung der Prüfung genehmigen. Paral-

lel muss ein Antrag zur Durchführung bei der zuständigen Ethik-Kommission gestellt werden. In länderübergreifenden, multizentrischen klinischen Studien nimmt somit die Anzahl der Einreichungen von Studienunterlagen linear zu.

Zusätzlich müssen die Zulassungsunterlagen bestimmten Vorgaben genügen. Diese werden von der *International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use*, kurz ICH, herausgegeben die die Aufgabe hat, Beurteilungskriterien von Arzneimitteln für Menschen als Basis der Arzneimittelzulassung in Europa, den USA und Japan zu harmonisieren. Neben den Zulassungsunterlagen erarbeitet die ICH auch empfehlende Guidelines für die Bewertung der Qualität, Wirksamkeit und Unbedenklichkeit von Arzneimitteln.

Unter dem Titel „Efficacy“ finden sich auf der Internetpräsenz der ICH (www.ICH.org) zahlreiche Guidelines die sich mit dem Design, der Durchführung, der Patientensicherheit und der Berichterstattung von klinischen Studien beschäftigen. Die ICH-Guidelines werden in der Europäischen Union vom zuständigen Ausschuss für Humanarzneimittel (CHMP) bei der EMA nahezu vollständig übernommen. Die Guidelines sind somit Leitlinien, von denen die Pharmaunternehmen nur in begründeten Fällen abweichen können [2].

Eine dieser Guidelines „*Structure and Content of Clinical Study Reports*“, kurz E3, das unter Medical Writern, den Personen die die Studienergebnisse schriftlich in einem Studienbericht festhalten als „Bibel“ umschrieben wird, enthält die Gliederung des Studienberichts. Es finden sich zahlreiche Beispiele für die Darstellung der erhobenen Patienteninformationen und deren Studiendaten. Beispielsweise findet sich im Anhang IVb der ICH E3 eine Übersicht über die Verteilung der Patienten in die verschiedenen Analyse-Populationen sowie den unterschiedlichen Behandlungsgruppen. Eine Kopie ist dem Anhang A dieses Artikels beigelegt.

In der Regel wird diese Übersicht, auch Consort-Diagramm genannt, durch den Medical Writer manuell angefertigt, in dem die erforderlichen Daten aus verschiedenen Tabellen herangezogen werden. Dem Automatismus und der Optimierung willen hatten wir uns das Ziel gesetzt, die Erstellung des Consort-Diagramms in SAS zu programmieren. Umgesetzt wurde die Programmierung mit Hilfe der SAS/GRAPH Annotate Facility und der GSLIDE-Prozedur. Die einzige Voraussetzung ist das Vorhandensein eines Datensatzes auf Individueller-Ebene mit den Merkmalsausprägungen „Y“ oder „N“ für die verschiedenen Analyse-Populationen. Weiterhin wird eine Variable für die Ausschlussgründe von Patienten aus Analyse-Populationen und einer weiteren Variable für die Zuordnung der Patienten in die untersuchten Behandlungsgruppen benötigt.

1.2 SAS/GRAPH Annotate Facility

Die SAS/GRAPH Annotate Facility ist eine eigenständige Grapfiksprache und erlaubt die freie Grafikprogrammierung. Sie kann benutzt werden, um die Ausgabe grafischer

Prozeduren durch Hinzufügen einfacher Elemente, wie Symbole oder Text, zu überarbeiten oder zu erweitern. Für die Verarbeitung wird ein Datensatz benötigt, der in die relevanten SAS/GRAPH Prozeduren integriert werden kann.

Des Weiteren stehen mit PROC GSLIDE und PROC GANNO zwei Annotate-Prozeduren zur Verfügung. Die Prozedur GANNO führt lediglich Annotate-Anweisungen aus und ignoriert definierte Titel, Notes und Fußnoten und weitere Grafikelemente wie Rahmen usw. Um Texte aus TITLE-, FOOTNOTE- und NOTE-Statements mit auszugeben, kann auf die Prozedur GSLIDE zurückgegriffen werden. Mit ihrer Hilfe können sehr einfach Plakate und Aushänge gestaltet und Annotate-Grafiken übersichtlicher gestaltet werden.

Ein Annotate-Datensatz enthält in jeder Observation eine (Grafik-) Aktion und muss drei Arten von Variablen enthalten, die genau regeln „Was“ (Funktionsvariable), „Wo“ (Positionsvariable) und „Wie“ (Attributvariable[n]) getan werden soll. Die Variablen haben feste, vorgegebene Namen und können durch explizite Angabe der Variablen und Werte erzeugt oder implizit durch Benutzung von Annotate-Makros erstellt werden. Annotate-Makros müssen vor der Verwendung kompiliert werden, dass durch den Aufruf des Makros %ANNOMAC erfolgt. Die „wichtigsten“ Funktionen, die im Zusammenhang dieser Arbeit auch Anwendung fanden, wären [3]:

- %MOVE(*x, y*): Bewegt den Grafikkursor an Position (X,Y)
- %DRAW(*x, y, color, line, size*): Zeichnet eine Linie von der letzten Position (XLAST,YLAST) nach (X,Y)
- %RECT(*x1, y1, x2, y2, color, line, size*): Zeichnet ein Rechteck über die Diagonale von der letzten Position (x1=XLAST,x2=YLAST) nach (X=x2,Y=y2)
- %LABEL(*x, y, text-string, color, angle, rotate, size, style, position*): Schreibt „Text“ an Position (X,Y)
- %ARROW(*x1, y1, x2, y2, color, line, size, angle, style*): Zeichnet ein Pfeil zwischen den Koordinaten (x1=XLAST,x2=YLAST) und (X=x2,Y=y2)

Es ist zwingend notwendig alle Parameter der Annotate-Makros in der richtigen Reihenfolge für eine erfolgreiche Ausführung der Anweisungen aufzurufen. Falls ein Standardwert benutzt werden soll, so muss dies explizit durch ein „.“ Angegeben werden. XLAST und YLAST sind interne Variablen und enthalten die Koordinaten der vorherigen Annotate-Aktion [3].

Auch muss zu Beginn definiert werden, in welchem Koordinatensystem die Annotate-Anweisungen durchgeführt werden sollen, da sowohl x- und y-Koordinaten als auch Textelemente von diesem abhängen. Das Koordinatensystem wird durch die Variablen XSYS und YSYS definiert. Sie haben einen Einfluss auf die Werte der Achsen X und Y und wie sie durch die Annotate-Funktionen interpretiert werden. Für diese Arbeit wurde die Definition des Koordinatensystems abhängig von den Werten gemacht, also absolute Achsenbestimmungen benutzt, wobei der Grafikbereich zwischen der horizontalen

und vertikalen Achse unterschiedlich gehandhabt wird. Die horizontale Achse benutzt den gesamten Grafikbereich der horizontal zur Verfügung steht (XSYS=2), wogegen die vertikale Achse hinsichtlich des Grafikbereichs verkleinert wurde, um Platz für eventuelle Titel und Fußnoten freizuhalten (YSYS=5).

Des Weiteren wurde die „SIZE“-Variable HSYS mit HSYS=2 eingeführt mit der die Darstellung beziehungsweise die Größe von Grafikelementen beeinflusst werden kann wie beispielsweise die Texthöhe bei Benutzung von %LABEL(). Sei hier auf die SAS-Dokumentation für weitere Details verwiesen [3].

2 Die Umsetzung in SAS

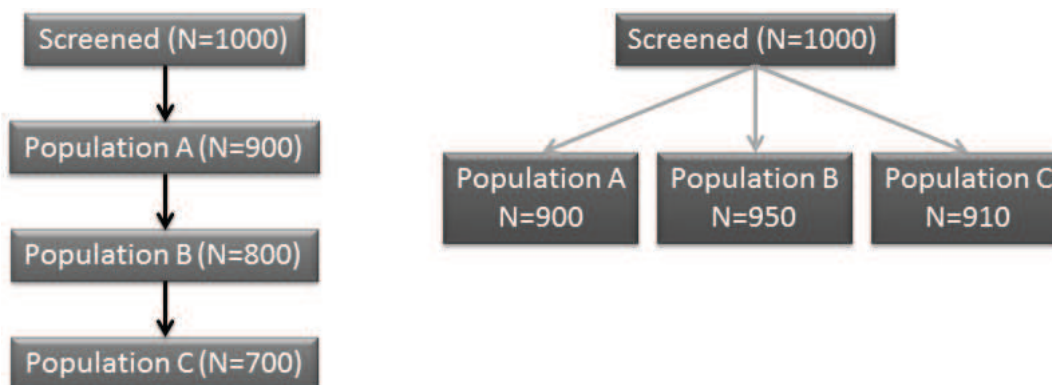
2.1 Häufigkeiten berechnen

In diesem Abschnitt werden wir Schritt für Schritt die Anzahl der Patienten in den untersuchten Analyse-Populationen, die Anzahl der Patienten in den Behandlungsgruppen sowie die Anzahl der Patienten in Abhängigkeit der Ausschlussgründe berechnen. Der Ausgangsdatensatz im folgenden Beispiel ist das Analysis Data Subject Level (ADSL) der CDISC ADaM Spezifikation.

Zu Beginn muss eine Differenzierung zwischen einer „hierarchischen“ und einer „parallelen“ Struktur der Analyse-Populationen getroffen werden. Hierarchisch in dem Sinne, dass die Population Y eine Teilmenge der Population X bildet und parallel in dem Sinne, dass die Populationsgröße der Population Y unabhängig von der Zugehörigkeit der Patienten zur Population X bestimmt werden kann. Zur visuellen Veranschaulichung sei auf die folgende Abbildung verwiesen.

Abbildung 1: Unterschied zwischen der „hierarchischen“ (links) und der „parallelen“ (rechts) Struktur der Analyse-Populationen.

Im Rahmen dieser Arbeit wird die „hierarchische“ Struktur der Analyse-Populationen



implementiert.

Berechne Anzahl der Patienten

Die Berechnung der Anzahl der Patienten und die Erstellung des Consort-Diagramms sind in ein SAS Makro integriert. Zum besseren Verständnis seien im Folgenden einige Makrovariablen erläutert, die bei der Beschreibung der Vorgehensweise benutzt werden:

- `&_fcActPopulation.`: Beschreibt die untersuchte Population; Beispiele: „Screened“, „Randomized“, „FAS“, ...
- `&fcPopCategoryYes.`: Beschreibt die Merkmalsausprägung für einen Einschluss in eine Analyse-Population. In der Regel ist es ein „Y“.
- `&fcIndataset.`: Beschreibt den Eingabedatensatz. In der Regel ist es ADSL.
- `&_fcNumOfPopulations.`: Beschreibt die Anzahl der untersuchten Analyse-Populationen.
- `&_fci.`: Ist ein Schleifenindex im Makro damit mehrere Analyse-Populationen automatisch gezählt und ausgewertet werden können. Für die Populationen in der Abbildung 1 würde `&_fci.=1` bis `&_fcNumOfPopulations.=4` laufen.
- `&_fcActPopulationBefore.`: Beschreibt die Population die eine Obermenge bildet, damit die „hierarchische“ Struktur berücksichtigt wird.
- `&fcTrtVar.`: Beschreibt die Variable die die Behandlungsinformation enthält.
- `&fcReasonExclVar.`: Beschreibt die Variable die die Gründe zum Ausschluss aus einer Analyse-Population enthält.

Im ersten Schritt wird mit der SQL-Prozedur die Anzahl der Patienten pro Analyse-Population ermittelt. Anschließend wird mit Hilfe der FREQ-Prozedur die Populationsgröße und die Verteilung der Patienten in die untersuchten Behandlungsgruppen ermittelt und in Datensätzen festgehalten:

```
PROC SQL;
  CREATE TABLE &_fcActPopulation._&fcPopCategoryYes. AS
    SELECT *
      FROM &fcIndataset.
        WHERE &_fcActPopulation.="&fcPopCategoryYes."
    %IF &_fci.>1 %THEN %DO;
      AND &_fcActPopulationBefore.="&fcPopCategoryYes."
    %END;
;
QUIT;
%LET _sqllobs01 = &SQLOBS.;

PROC FREQ DATA=&_fcActPopulation._&fcPopCategoryYes. NOPRINT;
  TABLE &_fcActPopulation. / OUT=fc50_freq&fcPopCategoryYes.&_fci.;
  TABLE &fcTrtVar. / OUT=fc40_freq&fcPopCategoryYes._trt_&_fci.;
RUN;
```

Die Anzahl der Beobachtungen wird über die automatische Makrovariable SQLOBS festgehalten. Diese wird im weiteren Verlauf des Makros verwendet um die Stabilität des Makros zu gewährleisten, da nur Analyse-Populationen mit mindestens einem Patienten als sinnvoll erachtet werden.

Das Consort-Diagramm enthält neben der Anzahl der eingeschlossenen Patienten auch die Gründe, die zum Ausschluss eines Patienten aus einer Analyse-Population führten. Diese wird ebenfalls mit Hilfe der `FREQ`-Prozedur in Abhängigkeit der Patientenanzahl ermittelt, wie im folgenden SAS-Code dargestellt:

```
PROC FREQ DATA=&_fcActPopulation._&fcPopCategoryNo. NOPRINT;  
  TABLE &_fcActPopulation.*&fcReasonExclVar. /  
    OUT=fc50_freqReasonExcl&_fci.;  
RUN;
```

Anschließend wird jeder Ausschlussgrund durch eine einfache Textmanipulation mit der Anzahl der ausgeschlossenen Patienten zusammengefügt:

```
fcReasonExclusion=STRIP(&fcReasonExclVar.) || "(N=" || STRIP(  
  PUT(count,BEST.)) || ")" ;
```

Eine der Herausforderungen war die grafische Darstellung der Ausschlussgründe in einer dafür vorgesehenen Box passend darzustellen. Ausschlussgründe können eine unterschiedliche Länge haben und z.T. sehr lang sein. Da der Platz für die Darstellung begrenzt ist müssen Ausschlussgründe über eine definierte maximale Länge hinaus umgebrochen werden, so dass die Darstellung „längerer“ Ausschlussgründe über zwei oder mehr Zeilen erfolgt. Hinzu kommt die Bedingung, dass nur ganze Wörter umgebrochen werden, um die Lesbarkeit aufrechtzuerhalten. Die folgende Tabelle 1 enthält Beispiele für Ausschlussgründe, die bei einer Zeichenlänge von >70 mit einem „#“ markiert werden. An genau diesen Markierungen wird in der grafischen Darstellung der Text umgebrochen und in einer neuen „Zeile“ dargestellt.

Die „passende“ Stelle zum Setzen der Markierung wird über die `FIND()`-Funktion realisiert. Hierbei wird die Suche nach einem Leerzeichen rückwärts durchgeführt, um erstens nach einem Wort umzubrechen und zweitens die zur Verfügung stehende maximale Länge kleiner 70 zu ermitteln.

```
_splitpos= FIND(&var., " ", (-1)*(_splitpos+&lineLength.-1));
```

Die Makrovariable `&var.` enthält die untersuchte Zeichenkette (Ausschlussgrund) wogegen die zur Verfügung stehende Zeichenlänge in der Makrovariablen `&lineLength.` festgehalten wird (hier `lineLength=70`). Die Variable `_splitpos` enthält die Position in der die Markierung gesetzt werden soll.

Tabelle 1: Darstellung der Zeichenmodifizierung bei gegebener maximaler Zeichenlänge

Ausschlussgründe	Länge	Anzahl Marker+	Modifizierte Ausschlussgründe mit Trennzeichen*
Missing: Reason for Exclusion is Missing (N=1)	46	0	Missing: Reason for Exclusion is Missing (N=1)
No measurement at Baseline (N=200)	34	0	No measurement at Baseline (N=200)
Patient is positive tested at baseline. Patient did not complete the study (N=10)	81	1	Patient is positive tested at baseline. Patient did not complete the#study (N=10)
Patient is positive tested at baseline. Patient did not complete the study. Patient had additional assessments not planned per protocol. Subject took prohibited medication (N=6)	177	2	Patient is positive tested at baseline. Patient did not complete the#study. Patient had additional assessments not planned per protocol. Subject#took prohibited medication (N=6)
Patient did not complete the study (N=8)	40	0	Patient did not complete the study (N=8)

+Die Anzahl der Trennungen ergibt sich aus der maximal zur Verfügung stehenden Zeichenlänge oder Breite des Feldes. In diesem Beispiel wurde eine maximale Zeichenlänge von 70 angenommen.

*Ausschlussgründe, die größer als 70 Zeichen, sind werden durch ein sogenanntes „Splitchar“ markiert. Die Darstellung „längerer“ Ausschlussgründe erfolgt auf mehreren Zeilen bedingt durch die Anzahl der Splitchars.

Das Setzen der Markierung(en) „#“ wurde mit Hilfe von Perl Regular Expressions realisiert. Die Funktion `PRXCHANGE()` führt, für ein vorgegebenes Muster, eine Textersetzung durch, das für unseren Fall wie folgt umgesetzt wurde:

```
&var.=PRXCHANGE("s/({",_splitpos-1,"})(.+)/\1&splitChar.\2/"
,-1,&var.);
```

Dabei wird die Makrovariable `&var.` für eine Zeichenlänge von `n=_splitpos-1` durchsucht, festgehalten, und mit der Markierung zusammengeführt (`\1&splitChar.`). Die Option `\1` (`\2`) entspricht dabei der Zeichenkette, die in der ersten (zweiten) runden Klammer festgehalten wurde „(...)“. Die „-1“ als dritter Parameter gibt an, dass die Zeichenkette bis zum Ende durchsucht und eine Ersetzung bei jedem Treffer durchgeführt wird. Siehe auch Tabelle 1 für einige Beispiele. Eine gute Einführung in die Regular Expressions findet sich in dem Beitrag von Deckert und Salzmann [4] sowie zahlreichen SUGI-Artikeln, unter anderem in [5].

Um eine möglichst hohe Automatisierungsrate zu erreichen, wird das Label der Analyse-Population aus der Metatabelle SASHELP.VCOLUMN ausgelesen. Dieser Datensatz beinhaltet alle Variablen-Metainformation aus allen Tabellen der aktuellen SAS-Sitzung. Mit dem folgenden SAS-Code wird das Label der Analyse-Population in der Makrovariablen `_fcPopVarlabel` festgehalten.

```
PROC SQL NOPRINT;
  SELECT STRIP(label) INTO :_fcPopVarlabel SEPARATED BY " "
  FROM sashelp.vcolumn
  WHERE UPCASE(libname) = "%UPCASE(%SCAN(&fcindataset.,1,.))"
  AND UPCASE(memname) = "%UPCASE(%SCAN(&fcindataset.,2,.))"
  AND UPCASE(name) = "%UPCASE(&_fcActPopulation.)"
  ;
QUIT;
```

In Kombination mit einer Schleife können auf diese Art und Weise mehrere Analyse-Populationen in Kombination mit den jeweiligen Ausschlussgründen manipuliert, ausgezählt und zusammen mit der Anzahl der ausgeschlossenen Patienten festgehalten werden. Die Abbildung 2 stellt ein Beispiel für den Datensatz nach erfolgter Zählung dar.

fcPopLabel	fcPopCount	fcTrtCount	fcReasonCombined
Intent-To-Treat#Analysis#Population	1405	Active (N=703)#Placebo (N=702)	
Safety Analysis#Population	1403	Active (N=702)#Placebo (N=701)	-Patient did not receive any study medication (N=2)
Per-Protocol#Analysis#Population	1027	Active (N=506)#Placebo (N=521)	-Missing: Reason for Exclusion is Missing (N=10)#-No measurement at Baseline (N=200)#-Patient is positive tested at baseline. (N=69)#-Patient is positive tested at baseline. Patient did not complete the#study (N=33)#-Patient is positive tested at baseline. Patient did not complete the#study. Patient had additional assessments not planned per#protocol.Patient took prohibited medication (N=17)#-Patient did not complete the study (N=47)

Abbildung 2: Zähldatensatz nach Manipulation langer Zeichenketten

Der Zähldatensatz wie in Abbildung 2 dargestellt liefert die Basis für SAS/GRAPH Annotate.

2.2 Consort-Diagramm zeichnen

Zunächst wird der Datensatz, wie in Abbildung 2 dargestellt, so aufgeschlüsselt, dass jede Beobachtung mit einer Markierung „#“ in einer eigenen „Zeile“ dargestellt wird.

Beispielsweise wird die Analyse-Population

```
Intent-To-Treat#Analysis#Population
```

in die Form

```
Intent-To-Treat
  Analysis
  Population
```

überführt und über drei Observationen verteilt. Dieser Schritt wird für die Ausschlussgründe und die Behandlungsgruppen wiederholt. Der resultierende Datensatz enthält keinen Text mehr, der über mehrere Zeilen dargestellt werden muss. Umgesetzt wurde die Manipulation mit einer Schleife, die über die Anzahl der Ausschlussgründe, der Analyse-Populationen und der Behandlungen läuft und nach der Markierung (&fcSplitchar.) sucht, den Text extrahiert und mit dem `OUTPUT`-Statement eine neue Observation einfügt:

```
DO i=1 TO numRowsRsn;
  rsnSplit=STRIP(SCAN(fcReasonCombined,i,"&fcSplitchar."));
  IF SUBSTR(rsnSplit,1,1) NE "-" THEN rsnSplit=" "||rsnSplit;
  reasonExists=1;
  OUTPUT;
END;
```

Die binäre Variable `reasonExists` entspricht dem Wahrheitswert, dass ein Ausschlussgrund existiert und findet Anwendung bei der Erstellung der Grafik. Nachdem die Analyse-Populationen, die Behandlungen und die Ausschlussgründe überarbeitet wurden, stellt der folgende Datensatz die Grundlage zur Erstellung des SAS/Graph Annotate-Datensatzes dar.

Im vorletzten Schritt wird der Annotate-Datensatz erzeugt, der für jedes Element, eine eigene Vorschrift á la Annotate-Syntax enthalten muss. Wie in Abschnitt 1.2 erläutert werden mit dem Aufruf des Makros `%ANNOMAC` eine Vielzahl von Annotate-Makros kompiliert.

Als erstes werden die Anweisungen zum Zeichnen der Analyse-Populationen mit Hilfe der Annotate-Funktion `%LABEL()` definiert (siehe Abschnitt 1.2 für Parameter der Annotate-Funktion). Die Variablen `fcMitte` und `fcry` dienen der dynamischen Darstellung von Elementen um die bestmögliche Darstellung der Texte, Boxen und Verbindungslinien zu gewährleisten.

```
/* PRINT POPULATION INFORMATION PER OBSERVATION */
%LABEL(fcMitte,fcry2-1,popSplitted,black,0,0,1.5,'Courier New',5);
```

trtSplitted	popSplitted	reasonSplitted	reasonExists
Active (N=703)	Intent-To-Treat		.
Placebo (N=702)	Analysis		.
	Population		.
	N=1405		.
Active (N=702)	Safety Analysis	-Patient did not receive any study medication (N=2)	1
Placebo (N=701)	Population		1
	N=1403		1
Active (N=506)	Per-Protocol	-Missing: Reason for Exclusion is Missing (N=10)	1
Placebo (N=521)	Analysis	-No measurement at Baseline (N=200)	1
	Population	-Patient is positive tested at baseline. (N=69)	1
	N=1027	-Patient is positive tested at baseline. Patient did not complete the	1
		study (N=33)	1
		-Patient is positive tested at baseline. Patient did not complete the	1
		study. Patient had additional assessments not planned per	1
		protocol. Patient took prohibited medication (N=17)	1
		-Patient did not complete the study (N=47)	1

Abbildung 3: Darstellung der Analyse-Population und Ausschlussgründe pro Observation

Für die Analyse-Populationen und die Behandlungen sieht die Vorschrift zum Zeichnen folgendermaßen aus:

color	text	X	Y	ANGLE	ROTATE	SIZE	STYLE	POSITION	FUNCTION
black	Randomized	27.5	94	0	0	1.5	'Courier New'	5	LABEL
black	A (N=703)	6	94	0	0	1.5	'Courier New'	6	LABEL
black	Analysis	27.5	92	0	0	1.5	'Courier New'	5	LABEL
black	B (N=702)	6	92	0	0	1.5	'Courier New'	6	LABEL
black	Population	27.5	90	0	0	1.5	'Courier New'	5	LABEL
black	N=1405	27.5	88	0	0	1.5	'Courier New'	5	LABEL

Abbildung 4: SAS/GRAPH Annotate Vorschrift zum Zeichnen der Analyse-Population und der Behandlungen

Die Anweisung für das Zeichnen eines Rahmens wird mit der Annotate-Funktion %RECT() definiert. Die Anweisung für das Zeichnen der Verbindungslinien zwischen den einzelnen Rechtecken erfolgt mit der Annotate-Funktion %ARROW().

```
%RECT(fcLinks,fcry2-1,fcRechts,rechtEckRO,black,1,1);
%ARROW(fcMitte,fcry2-1,fcMitte,fcry2-fcArrowLength,black,1,1,0
, 'Courier New');
```

Hier sei erwähnt, dass Bedingungen existieren, die erstens die Dynamik des Zusammenhangs aufrechterhalten und zugleich dafür sorgen, dass mit dem Zeichnen der letzten Analyse-Population, Behandlungen oder Ausschlussgründe, keine weitere Verbindungslinie gezeichnet wird.

Zu guter Letzt wird der erzeugte Annotate-Datensatz der Prozedur GSLIDE übergeben, damit die Anweisungen ausgeführt und das Resultat in einer Grafik dargestellt wird.

```
PROC GSLIDE ANNOTATE=fcAnnotate;
RUN;
QUIT;
```

Hat man alles richtig gemacht erhält man schließlich die folgende grafische Darstellung (Abbildung 5). Diese enthält zu jeder Analyse-Population die Anzahl der Patienten und die Verteilung der Patienten in die untersuchten Behandlungsgruppen. Des Weiteren werden die Gründe zusammen mit der Anzahl der Patienten, die aus einer oder mehreren Analyse-Populationen ausgeschlossen wurden mit ausgegeben.

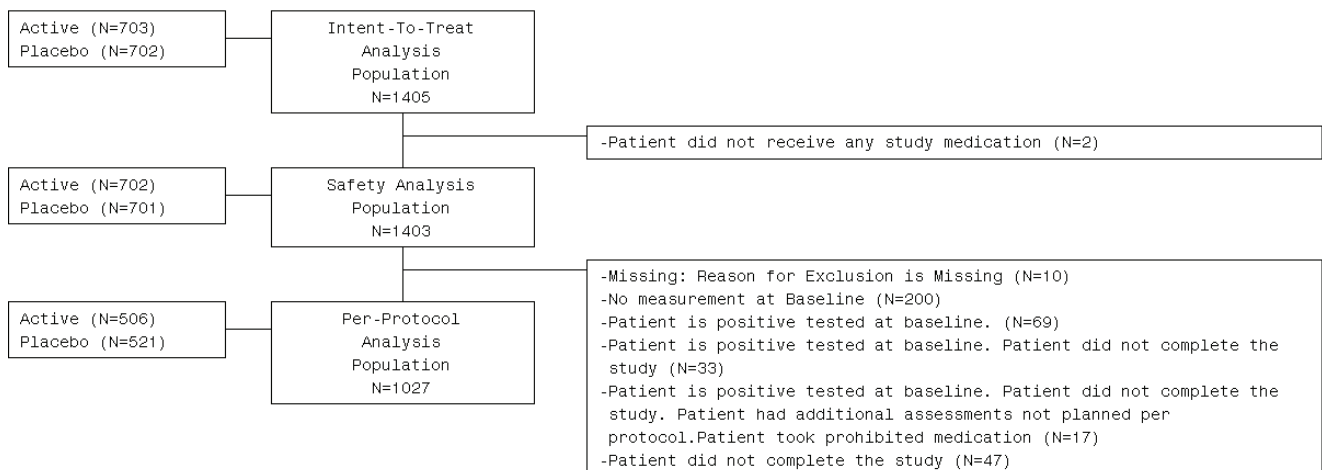


Abbildung 5: Verteilung der Patienten auf die untersuchten Analyse-Populationen, der Behandlungsgruppen und der Gründe die zum Ausschluss führten.

Literatur

- [1] Carina Ortseifen, Grischa Pfister, Heribert Ramroth, Marianne Weires: Tipps und Tricks für den leichteren Umgang mit der SAS Software, KSFE 2010.
- [2] In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 26. September 2013, 06:49 UTC. URL: http://de.wikipedia.org/w/index.php?title=International_Conference_on_Harmonisation_of_Technical_Requirements_for_Registration_of_Pharmaceuticals_for_Human_Use&oldid=122884573 (Abgerufen: 15. Januar 2015, 14:13 UTC)

- [3] In: SAS/Graph 9.4: Reference, Third Edition. URL: [http://support.sas.com/ documentation/](http://support.sas.com/documentation/) (Abgerufen: 19. Januar 2015, 11:56 UTC).
- [4] Andreas Deckert und Heiko Zimmermann: Tipps und Tricks für den leichteren Umgang mit der SAS Software, KSFE 2011.
- [5] Ron Cody: An Introduction to Perl Regular Expressions in SAS 9, SUGI29, Paper 265-29, Tutorial.

Anhang A Disposition of Patients – Annex IVb

