SESUG Paper HP-260-2019

Who, When, and Where – A Step-by-Step Approach to Creating a Color-Coded Tracker for Clinical Trial Subjects

Venita DePuy, PhD, Bowden Analytics

ABSTRACT

In a perfect world, all subjects participating in a clinical trial would attend each and every scheduled visit on the planned date. Realistically, the project management team may spend a significant amount of time trying to keep track of who is where, and when. Busy medical offices may not get subjects scheduled at the right times, participants may be too ill to attend visits, or any number of other issues may arise that lead to missed or out-of-window visits during the course of the trial. As subject participation in the trial draws to a close, there is increased pressure to schedule monitoring visits and site close-out visits as soon as possible, to allow final data cleaning prior to a timely database lock and the final study tables, listings, and figures.

We will present a step-by-step approach to creating a traffic-lighted subject tracker, intended for use in clinical trials (but applicable to other fields as well). Based on each subject's initial treatment date, all future study visits are populated in the spreadsheet, using cell color and font characteristics to indicate a predicted future date. Overdue visits (when the tracker is run and the predicted date is prior to the program run date) are then highlighted to indicate a potential late visit. Completed visits are populated with the actual visit date (with out-of-window indicators if necessary) and color-coded to reflect a past visit. Subjects who have completed the trial are automatically moved to a separate tab of the sheet, with all information retained. This program can be set to run automatically each morning, after a daily data refresh, to have a handy, visual reference for subject visits and associated data entry.

INTRODUCTION

This paper is intended for all levels of programmers. As such, it starts out with a description of the simplest way to create a basic subject tracker. Then, sequential modifications are introduced, with code snippets and a limited output table to show the cumulative impact of those modifications. The entire code to create the complete tracker, which includes sample data, is in Appendix 2. The output created by the program is in Appendix 1.

THE BASIC SUBJECT TRACKER

The first steps in this process are to gather the data, create the desired data layout, and export it to an XML spreadsheet.

GATHERING DATA

There are two basic approaches to gathering the data – either use raw data from the case report forms (CRFs), or wait until CDISC data is available. There is no 'best' option. If you use raw data straight from the database, as collected on the CRFs, the data will be available from the very beginning of the data collection portion of the study. That can allow you to have the tracker available as soon as possible, and provide the greatest amount of insight to the project management team during the early part of subject enrollment. On the other hand, the vast majority of clinical trials utilize CDISC datasets. If the tracker is created to use CDISC data structure, it can be more easily adapted to other trials. I notice that more companies are making an effort to have SDTM datasets (if not ADaM datasets and draft tables, listings, and figuress) programmed much earlier in the course of the study, so there may not be much time difference between the two, or the subject tracker may not be as helpful early in the study when the focus is more on enrollment than monitoring study visit patterns. Clinical research organizations typically have standard CRFs and database templates, so even if variable names are being used that are not industry standard, it is likely that those variable names are used across studies and the program can be applied to other trials. If there is not a pressing need to choose one over the other, I would encourage basing the tracker on SDTM datasets due to the ability to better adapt the tracker to other studies.

Although study-specific, the following data sources would typically be included in the tracker:

- Demographics (DM): general subject characteristics such as age, date of birth, sex, race, first dose date and date of informed consent
- Disposition(DS): disposition information including subject completion or early termination, termination reason, and date of completion or termination
- Study visits (SV): subject visits and dates
- Exposure (EX) or Drug Administration (DA): exposure to study product or amount dispensed, depending on the specifics of the clinical trial
- Screen Failure (SF): Screen failure information may be of interest

In general, I transpose the datasets that are 1 record per subject per visit (EX, DA, SV) and merge those with DM, DS, and SF to have a composite dataset with 1 record per subject.

For the purpose of this paper, I have created some sample data for illustrative purposes. Representative data – limited demographics, and visits for the first 10 weeks and Week 36 visit – is included herein. I would expect to have a much larger number of subject attributes and visit dates included in a tracker used in a clinical trial. Excerpts of the output are included in Tables 1- 5, with the full output from the program in Appendix 1.

OVERALL LAYOUT OF THE SUBJECT TRACKER

My subject tracker has two sections: Enrollment – Dose Tracking and Completed or Early Termination (ET) subjects.

Enrollment – Dose Tracking

This tab is the main part of the tracker. As shown in **Table 1**, it has subject number and characteristics, followed by informed consent and randomization dates, then the dates and dose levels for all the planned (or potential) visits.

The original tracker that I created was for use in a study with regular dose escalations (and occasional de-escalations); therefore, this is designed to have both visit date and dose level at each visit displayed. Other trials may have fixed dose levels throughout the study, and only present visit dates.

Completed or ET

This (hopefully short!) listing displays information on both screen failures (if entered in the database) and subjects who withdrew from the study. It can be a good reference to determine if screen failures are more prevalent at a certain site, during a certain part of the study, and so forth. Disposition information can be incorporated into the composite dataset, and displayed here. Or, the tracker can be simplified by presenting information from all subjects on one sheet, instead of splitting by subject status.

CREATING THE INITIAL TRACKER

The tracker is literally a couple of Proc Report statements, nested within some ODS TAGSETS.EXCELXP statements. In this simplest form (without traffic lighting or color-coded text), it could even be created using Proc Print statements in lieu of Proc Report.

The format is as follows:

```
ODS TAGSETS.ExcelXP
      path="X:\xxxx\xxxx\xxxxxx"
      file='SESUG_HP_260_2019_Example.xml' STYLE=XLsansprinter;

 ODS TAGSETS.ExcelXP options( sheet_name="Enrollment-Dose Tracking");
      PROC REPORT DATA=xxx NOWINDOWS;
      . . . .
      Run;

 ODS TAGSETS.ExcelXP CLOSE;
```

Table 1. Basic Subject Tracker (excerpt)

SUBJECT INFORMATION					VISIT 1		UPDOSING				VISIT 3	
PI	Subject ID	Age (yrs)	Sex	Date of Informed Consent	First Dose Date	Dose	Week 2 date	Wk 2 dose	Week 4 date	Wk 4 dose	Week 36 date	Wk 36 dose
Jones	101_001	41	M	24MAR2019	09MAY2019	6	29MAY2019	6	12JUN2019	12		
Jones	101_002	25	M	12MAR2019	02APR2019	6	17APR2019	12	29APR2019	20		
Smith	102_001	32	F	07APR2019	02JUL2019	6	16JUL2019	12				
Smith	102_002	35	M	06MAY2019	19JUN2019	6	01JUL2019	12	16JUL2019	6		
Smith	102_003	27	M	09MAY2019	10JUN2019	6	23JUN2019	40	11JUL2019	20		
Smith	102_004	24	F	18MAR2019	15APR2019	6	30APR2019	12	12MAY2019	20		
Jackson	103_001	34	M	15APR2019	20MAY2019	6	03JUN2019	12	03JUN2019	20		

That approach gives you a tracker similar to that shown in **Table 1**. Note that this table shows a subset of the information in the tracker, due to space considerations. While it does show the information as recorded in the database, this can be expanded to provide more information at your fingertips.

MODIFICATION #1: PREDICTED VISIT DATES

A main goal for this tracker is to predict future visit dates, in order to be able to better schedule site visits and estimate subject completion timelines. Therefore, in addition to the visit dates present in the database, we will populate future visit dates, based on the first dose date. Visits are at 2 week intervals after the first dose, through Week 36. It is a fairly straightforward matter of creating planned visit dates, for Week 2 through Week 36 in this example, based on that first dose date. Future visits can be identified by a missing dose level associated with that date, so an additional flag is not necessary to differentiate planned future visits from visits that have already occurred.

This approach is very useful for predicting when key visits will occur, if sites are to be monitored after the completion of the Week 12 visits, for example. It also allows the project management team to predict the last visit within a site (since all of the 101_xxx subjects are at Site 101), in order to schedule site close-out visits.

Once predicted visit dates have been added, it may be difficult to determine if a particular visit has occurred on a certain date, or was scheduled to occur and either did not occur or has not been entered. Color-coding the text makes it easier to identify the status of a visit.

MODIFICATION #2: IDENTIFYING PAST VISITS, FUTURE VISITS, OVERDUE VISITS, AND INCORRECT DATE PATTERNS USING DIFFERENT TEXT COLORS

For clarity, we will assign the following color coding of dates:

- Future dates will be blue
- Problematic dates (overdue or obviously incorrect) will be red
- Visits which have occurred and been entered in the database (and are not problematic) will remain **black**.

Overall, this is done by creating an additional variable for every date field, which indicates which color should be used for the text. The values of that variable are assigned using a Proc Format statement, as follows:

```
PROC FORMAT;
      value DateFmt      0='#000000'
                         1='red'
                         2='blue';
      run;
```

In the case of the Week 4 visit, we want to compare the Week 4 date (DT1_04) to the prior dates (Visit 1 [DT1] and Week 2 [DT1_02]). We create format variable F1_04 that will have a value of 0 (past visit, no problems), 1 (problematic visit), or 2 (future visit). This can be derived as:

```
if    DT1_04 <= max(D1, DT1_02) or
      (DT1_04 <= ("&SYSDATE"d - 10) and D1_04 = .) then F1_04=1;
  else if  DT1_04 > ("&SYSDATE"d - 10) then F1_04=2;
  else F1_04=0;
```

which flags the date in red text if it is the same date or earlier than a previous visit date (indicating an erroneous date entered somewhere). It is also flagged in red text if the SAS program is submitted more than 10 days after an expected visit date, which has not been entered in the database, indicating an overdue visit or a site that needs prompted to complete data entry. The date is flagged in blue if it is a future date or within the 10-day window allowed for data entry. Finally, the text remains black for visits which have occurred and been entered in the database, and are not flagged as problematic.

The following code shows how the text color is altered for the Week 4 date, using the foreground style option In Proc Report to assign the color for D1_04 based on the value of F1_04. The full code, reflecting all the variables, is in Appendix 2:

```
PROC REPORT data= xxx nowindows;
     column ('SUBJECT INFORMATION'  subjid  );
     column ('UPDOSING'  DT1_04  );
     column  F1_04;
     column dummy; *  Dummy column to perform traffic lighting;

     define subjid     / display order order=internal "Subject ID";
     define DT1_04     / display "Week 4 date" ;
     define F1_04      / display  noprint;
     define dummy      / computed noprint;

     compute dummy;
       call define('DT1_04', 'STYLE',
           "STYLE=[foreground=" || put(F1_04, datefmt.) ||']');
     endcomp;
     run; quit;
```

Table 2. Subject Tracker with Date Text Color Changes (excerpt)

SUBJECT INFORMATION					**VISIT 1**		**UPDOSING**				**VISIT 03/EXIT**	
PI	**Subject ID**	**Age (yrs)**	**Sex**	**Date of Informed Consent**	**First Dose Date**	**Dose**	**Week 2 date**	**Wk 2 dose**	**Week 4 date**	**Wk 4 dose**	**Week 36 date**	**Wk 36 dose**
Jones	101_001	41	M	24MAR2019	09MAY2019	6	29MAY2019	6	12JUN2019	12	16JAN2020	
Jones	101_002	25	M	12MAR2019	02APR2019	6	17APR2019	12	29APR2019	20	10DEC2019	
Smith	102_001	32	F	07APR2019	02JUL2019	6	16JUL2019	12	30JUL2019		10MAR2020	
Smith	102_002	35	M	06MAY2019	19JUN2019	6	01JUL2019	12	16JUL2019	6	26FEB2020	
Smith	102_003	27	M	09MAY2019	10JUN2019	6	23JUN2019	40	11JUL2019	20	17FEB2020	
Smith	102_004	24	F	18MAR2019	15APR2019	6	30APR2019	12	12MAY2019	20	23DEC2019	
Jackson	103_001	34	M	15APR2019	20MAY2019	6	03JUN2019	12	03JUN2019	20	27JAN2020	

Table 2 displays partial results from the program as submitted on 10Oct2019, so the visit dates were compared to that date (as stored in &sysdate). Therefore, 102_001's 30Jul2019 projected visit date for Week 4 is flagged as it has not been completed, as indicated by the missing Week 4 dose level. 103_001's 03Jun2019 date for Week 4 is flagged as it is the same as the Week 2 date for that subject. Data entry errors happen; this is a quick way to alert the project management team, so that they can work with the data management team to have a query issued to correct that date. The Week 36 dates are all in the future, as indicated by the blue text.

MODIFICATION #3: ADDING TRAFFIC LIGHTING OF DOSE LEVEL, BASED ON CHANGE FROM LAST VISIT

In a study of this type that has a great deal of dose titration, it is helpful to color-code the dose increases, decreases, and constant dose level periods. To make it more eye-catching (and to help differentiate from the color-coded text of visit dates), we instead color-code the background of the cell, i.e. traffic lighting.

We will assign the following traffic lighting to dose levels:

- Increases will be green
- No change in dose level will be yellow
- Decreases will be orange

This is very similar to the color coding of date text; we create an additional variable for each post-baseline dose level, which indicates which color should be used for the background of the cell. The values of that variable are assigned using a Proc Format statement, as follows:

```
PROC FORMAT;
      value ChgFmt  -1 = '#FFCC00'
                     0 = '#FFFF99'
                     1 = '#AAFD8E'
                     other='white';
      run;
```

We again use Week 4 as a programming example; see the full code in Appendix 2 for comprehensive programming code. We compare the Week 4 dose level, D1_04, to the Week 2 dose level, D1_02. The change from prior dose level at Week 4 will be stored in in C1_04. This can be derived as follows:

```
If D1_04 > D1_02 > . then C1_04 = 1;
  else if D1_04 = D1_02 > .  then C1_04 = 0;
  else if . < D1_04 < D1_02 then C1_04 = -1;
```

which shades the cell green in the event of a dose increase (C1_04 = 1), yellow if the dose remains the same (C1_04 = 0), and orange if the dose decreases (C1_04 = -1). These colors were chosen as, in the original trial, doses were expected in increase consistently until reaching the maximum planned dose, and stable dose levels (or especially decreases) were potentially indicative of adverse events. The project management team was interested in being able to see if there was a prevalence of either of the latter two categories, from a subject safety perspective.

The following excerpt of code assigns that variable to the background cell color for the Week 4 dose level, D1_04. As before, Appendix 2 includes the full code. For simplicity, the code excerpt includes the subject identifier, Week 4 date, and Week 2 and Week 4 dose levels, without the modifications introduced in the earlier step:

```
PROC REPORT data= xxx nowindows;
      column ('SUBJECT INFORMATION'  subjid );
      column ('UPDOSING' D1_02 DT1_04     D1_04 );
      column  C1_04           ;
      column dummy; *  Dummy column to perform traffic lighting;

      define subjid     / display order order=internal "Subject ID";
      define D1_02 / display        "Wk 2 dose";
      define D1_04 / display        "Wk 4 dose";
      define C1_04 / display noprint;

      define DT1_02 / display       "Week 2 date";
      define dummy / computed noprint;

       compute dummy;
         call define('D1_04', 'STYLE', 'STYLE=[background= '
                  ||put(C1_04, chgfmt.) || ']');
       endcomp;
       run; quit;
```

Table 3. Subject Tracker with Traffic-Lighted Dose Levels and Date Text Color Changes (excerpt)

SUBJECT INFORMATION					**VISIT 1**		**UPDOSING**				**VISIT 03/EXIT**	
PI	**Subject ID**	**Age (yrs)**	**Sex**	**Date of Informed Consent**	**First Dose Date**	**Dose**	**Week 2 date**	**Wk 2 dose**	**Week 4 date**	**Wk 4 dose**	**Week 36 date**	**Wk 36 dose**
Jones	101_001	41	M	24MAR2019	09MAY2019	6	29MAY2019	6	12JUN2019	12	16JAN2020	
Jones	101_002	25	M	12MAR2019	02APR2019	6	17APR2019	12	29APR2019	20	10DEC2019	
Smith	102_001	32	F	07APR2019	02JUL2019	6	16JUL2019	12	30JUL2019		10MAR2020	
Smith	102_002	35	M	06MAY2019	19JUN2019	6	01JUL2019	12	16JUL2019	6	26FEB2020	
Smith	102_003	27	M	09MAY2019	10JUN2019	6	23JUN2019	40	11JUL2019	20	17FEB2020	
Smith	102_004	24	F	18MAR2019	15APR2019	6	30APR2019	12	12MAY2019	20	23DEC2019	
Jackson	103_001	34	M	15APR2019	20MAY2019	6	03JUN2019	12	03JUN2019	20	27JAN2020	

Table 3 again displays partial results due to space constraints, with full results in Appendix 1. As you can see:

- 101_001's dose remains constant (yellow) at Week 2, and increases at Week 4
- 102_001 and 102_003 have dose increases at Week 2 and decreases at Week 4

MODIFICATION #4: TRAFFIC LIGHTING OF OVERDUE VISITS

To make apparent overdue visits (whether actual missed visits, visits scheduled later than expected, or slow data entry of on-time visits) more readily apparent, we’re next going to add traffic lighting to mark all of the overdue visits with light blue (cyan) cell background. That will make these dates, which are already in blue text, stand out even more.

As before, we could simply create another format (in this case, for a white background or a cyan background):

```
PROC FORMAT;
  value Overdue    9 ='#00FFFF'
                   other='white';
   run;
```

but for efficiency, we are instead simply going to create 2 new levels for an existing format, and incorporate those:

```
PROC FORMAT;
 value ChgFmt     -1 = '#FFCC00'
                   0 = '#FFFF99'
                   1 = '#AAFD8E'
                   9 = '#00FFFF'
                   other= 'white';
   run;
```

The overdue variable is also simple to create, as shown using OV1_02 for an overdue Week 2 visit:

```
If DT1_02 <= “&sysdate”d – 10 and missing (D1_02) then OV1_02 = 9;
```

Which is then incorporated into the output table by assigning the background for the Week 2 visit date (DT1_02) based on the value of OV1_02, as follows:

```
proc report data= FINAL split='*' nowindows;
      column ('SUBJECT INFORMATION'  subjid   );
      column ('UPDOSING' DT1_02  );
      column  OV1_02          ;
      column dummy; *  Dummy column to perform traffic lighting;

      define Subjid     / display order order=internal      "Subject ID";
      define DT1_02 / display       "Week 2 date"   ;
      define OV1_02 / display noprint;
      define dummy / computed noprint;

      compute dummy;
       call define('DT1_02', 'STYLE', "STYLE=[ foreground="
                  ||put(F1_02, datefmt.)
                  || "background=" ||put(OV1_02, chgfmt.) || ']');
       endcomp;
 run; quit;
```

Table 4. Subject Tracker with Traffic-Lighted Overdue Visits and Dose Levels, and Date Text Color Changes (excerpt)

SUBJECT INFORMATION					**VISIT 1**		**UPDOSING**				**VISIT 03/EXIT**	
PI	**Subject ID**	**Age (yrs)**	**Sex**	**Date of Informed Consent**	**First Dose Date**	**Dose**	**Week 2 date**	**Wk 2 dose**	**Week 4 date**	**Wk 4 dose**	**Week 36 date**	**Wk 36 dose**
Jones	101_001	41	M	24MAR2019	09MAY2019	6	29MAY2019	6	12JUN2019	12	16JAN2020	
Jones	101_002	25	M	12MAR2019	02APR2019	6	17APR2019	12	29APR2019	20	10DEC2019	
Smith	102_001	32	F	07APR2019	02JUL2019	6	16JUL2019	12	30JUL2019		10MAR2020	
Smith	102_002	35	M	06MAY2019	19JUN2019	6	01JUL2019	12	16JUL2019	6	26FEB2020	
Smith	102_003	27	M	09MAY2019	10JUN2019	6	23JUN2019	40	11JUL2019	20	17FEB2020	
Smith	102_004	24	F	18MAR2019	15APR2019	6	30APR2019	12	12MAY2019	20	23DEC2019	
Jackson	103_001	34	M	15APR2019	20MAY2019	6	03JUN2019	12	03JUN2019	20	27JAN2020	

Table 4 further modifies the results shown in Tables 1 – 3 by adding the traffic lighting for 102_001's overdue Week 4 visit, to draw attention to the fact that the project management team likely needs to contact the site for a status update on that subject.

MODIFICATION #5: TRAFFIC LIGHTING OF COLUMN HEADERS

While a relatively minor change, for large spreadsheets, it can be helpful to modify the background of the column headers for increased legibility or to call attention to a certain column. In this case, I wanted to visually differentiate the different weeks by alternating from a light red to light blue background, depending on the date. I also wanted to change the background of the age column header to purple, to call attention to that field.

Since these changes are limited to just the column header, the alteration is made in Proc Report, and additional variables are not created (unlike earlier steps). The below excerpts from Proc Report show that the age column header is assigned a background color of purple (CX8A5AA6), while the dose levels alternate between light blue (CX90B9D9) and light red (CXE5A099). The dates are assigned similar alternating colors in the style(header) portion of the define statement.

```
define age / display   style(header)=[backgroundcolor=CX8A5AA6 ];

define D1 / display    style(header)=[backgroundcolor=CX90B0D9 ];
define D1_02/display   style(header)=[backgroundcolor=CXE5A099 ];
define D1_04/display   style(header)=[backgroundcolor=CX90B0D9 ];
define D1_06/display   style(header)=[backgroundcolor=CXE5A099 ];
```

Also, due to the importance of the subject identifier, I will put that entire column in boldface, using the style(column) option in Proc Report:

```
define Subjid/  display   "Subject ID" STYLE(Column)=data_bold;
```

Table 5. Subject Tracker with Traffic-Lighted Column Headers, Overdue Visits, and Dose Levels, and Date Text Color Changes (excerpt)

SUBJECT INFORMATION					**VISIT 1**		**UPDOSING**				**VISIT 03/EXIT**	
PI	**Subject ID**	**Age (yrs)**	**Sex**	**Date of Informed Consent**	**First Dose Date**	**Dose**	**Week 2 date**	**Wk 2 dose**	**Week 4 date**	**Wk 4 dose**	**Week 36 date**	**Wk 36 dose**
Jones	101_001	41	M	24MAR2019	09MAY2019	6	29MAY2019	6	12JUN2019	12	16JAN2020	
Jones	101_002	25	M	12MAR2019	02APR2019	6	17APR2019	12	29APR2019	20	10DEC2019	
Smith	102_001	32	F	07APR2019	02JUL2019	6	16JUL2019	12	30JUL2019		10MAR2020	
Smith	102_002	35	M	06MAY2019	19JUN2019	6	01JUL2019	12	16JUL2019	6	26FEB2020	
Smith	102_003	27	M	09MAY2019	10JUN2019	6	23JUN2019	40	11JUL2019	20	17FEB2020	
Smith	102_004	24	F	18MAR2019	15APR2019	6	30APR2019	12	12MAY2019	20	23DEC2019	
Jackson	103_001	34	M	15APR2019	20MAY2019	6	03JUN2019	12	03JUN2019	20	27JAN2020	

As you can see in both Table 5 and Appendix 1, having the visit date and dose level visually coupled using the traffic-lighting of the column header makes it much easier to tell at a glance which date goes with which dose level.

MODIFICATION #6: SEPARATING ENROLLED SUBJECTS FROM COMPLETED / TERMINATED SUBJECTS

The last modification is to split the active subjects from those who are no longer on the study, to output them to two separate tabs in the XML spreadsheet. It is a simple matter of adding an additional ODS TAGSETS statement to define the sheet name, and a WHERE statement to subset the output.

The format is as follows:

```
ODS TAGSETS.ExcelXP path=”X:” file=’xxx.xml’ STYLE=XLsansprinter;

 ODS TAGSETS.ExcelXP options( sheet_name="Enrollment-Dose Tracking");
      PROC REPORT DATA=xxx (where=(OFFSTUDY=0)) NOWINDOWS;
      . . . .
      Run;
ODS TAGSETS.ExcelXP options(  sheet_name="Completed and ET");
      PROC REPORT DATA=xxx (where=(OFFSTUDY=1)) NOWINDOWS;
      . . . .
      Run;
 ODS TAGSETS.ExcelXP CLOSE;
```

CONCLUSION

Overall, this paper walks through the steps from creating a basic subject tracker, to having a tracker with multiple types of traffic lighting and colored text. Snippets of SAS code were provided for each step, while the complete SAS code is in Appendix 2.

That complete code also includes additional modifications to create a custom style in Proc Template (based on Vince DelGobbo’s work) and expands the text snippets to provide calculations for all variables of a given type.

REFERENCES

DelGobbo, Vincent. 2010. “Traffic Lighting Your Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS®” *Proceedings of the SAS Global Forum 2010*, Seattle, WA. Available at http://support.sas.com/resources/papers/proceedings10/153-2010.pdf.

ACKNOWLEDGMENTS

My deepest thanks go to Vince DelGobbo, whose SAS Global Forum and regional Users Group presentations are always enlightening.

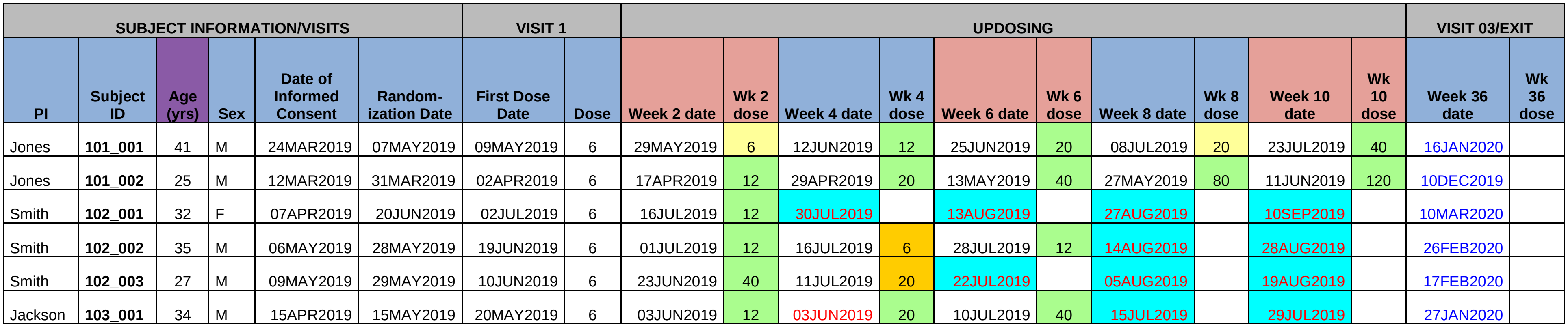
CONTACT INFORMATION

You may reach the author, Venita DePuy, with questions or comments. She is the owner of Bowden Analytics, based in Raleigh, NC, which provides statistical and SAS programming services. Her email is Bowden.analytics@gmail.com, and you may speak with her directly at (919) 578-7828.

APPENDIX I

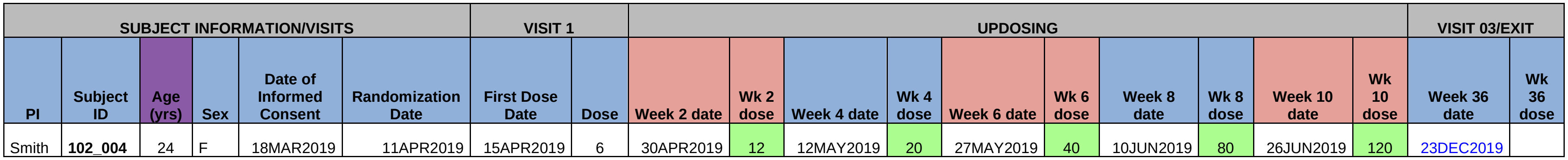
The SAS program in Appendix 2 creates an XML file with two tabs:

The Enrollment – Dose Tracking tab, as shown below:



SUBJECT INFORMATION/VISITS						VISIT 1		UPDOSING										VISIT 03/EXIT	
PI	**Subject ID**	**Age (yrs)**	**Sex**	**Date of Informed Consent**	**Random-ization Date**	**First Dose Date**	**Dose**	**Week 2 date**	**Wk 2 dose**	**Week 4 date**	**Wk 4 dose**	**Week 6 date**	**Wk 6 dose**	**Week 8 date**	**Wk 8 dose**	**Week 10 date**	**Wk 10 dose**	**Week 36 date**	**Wk 36 dose**
Jones	**101_001**	41	M	24MAR2019	07MAY2019	09MAY2019	6	29MAY2019	6	12JUN2019	12	25JUN2019	20	08JUL2019	20	23JUL2019	40	16JAN2020	
Jones	**101_002**	25	M	12MAR2019	31MAR2019	02APR2019	6	17APR2019	12	29APR2019	20	13MAY2019	40	27MAY2019	80	11JUN2019	120	10DEC2019	
Smith	**102_001**	32	F	07APR2019	20JUN2019	02JUL2019	6	16JUL2019	12	30JUL2019		13AUG2019		27AUG2019		10SEP2019		10MAR2020	
Smith	**102_002**	35	M	06MAY2019	28MAY2019	19JUN2019	6	01JUL2019	12	16JUL2019	6	28JUL2019	12	14AUG2019		28AUG2019		26FEB2020	
Smith	**102_003**	27	M	09MAY2019	29MAY2019	10JUN2019	6	23JUN2019	40	11JUL2019	20	22JUL2019		05AUG2019		19AUG2019		17FEB2020	
Jackson	**103_001**	34	M	15APR2019	15MAY2019	20MAY2019	6	03JUN2019	12	03JUN2019	20	10JUL2019	40	15JUL2019		29JUL2019		27JAN2020	

And the Completed or ET tab, as shown below:



SUBJECT INFORMATION/VISITS						VISIT 1		UPDOSING										VISIT 03/EXIT	
PI	**Subject ID**	**Age (yrs)**	**Sex**	**Date of Informed Consent**	**Randomization Date**	**First Dose Date**	**Dose**	**Week 2 date**	**Wk 2 dose**	**Week 4 date**	**Wk 4 dose**	**Week 6 date**	**Wk 6 dose**	**Week 8 date**	**Wk 8 dose**	**Week 10 date**	**Wk 10 dose**	**Week 36 date**	**Wk 36 dose**
Smith	**102_004**	24	F	18MAR2019	11APR2019	15APR2019	6	30APR2019	12	12MAY2019	20	27MAY2019	40	10JUN2019	80	26JUN2019	120	23DEC2019	

Subjects are assigned to the relevant tab based upon the OFFSTUDY variable, indicating whether the subject has withdrawn from the study or not. This allows the project management team to focus efforts on currently enrolled subjects, while still having information on other subjects easily available.

APPENDIX II

The following SAS code will create the entire tracker in XML, as described in the earlier steps. Sample data is included for ease of programming. Be sure to update the path output location.

```
*------------------------------------------------------------------------------------------------
--------------------;
*  Who, When and Where - A Step-by-Step Approach to Creating a Color-Coded Tracker for Clinical
Trial Subjects;
*  a presentation for the Southeast SAS Users Group (SESUG) conference, October 2019
*
*  Copyrighted by Venita DePuy, PhD
*  Owner, Bowden Analytics, Raleigh NC
*
*  This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
*
*  Contact Venita at bowden.analytics@gmail.com for relevant requests
*------------------------------------------------------------------------------------------------
-------------------;

data dset;
infile DATALINES dsd dlm=',';
format DT1 DT1_02 DT1_04 DT1_06 DT1_08 DT1_1 DT2_36 ICDT RANDODT date9.;
input PI $     SUBJID $       AGE     Sex $   ICDT :date11.  RANDODT :date11.
               DT1 :date11.     D1
               DT1_02   :date11. D1_02
               DT1_04  :date11.  D1_04
               DT1_06  :date11.  D1_06
               DT1_08  :date11.  D1_08
               DT1_1   :date11.  D1_1
               DT2_36  :date11.  D2_36       OFFSTUDY  ;
CARDS;
Jones,101_001,41,M,24-Mar-2019,07-May-2019,09-May-2019,6,29-May-2019,6,12-Jun-2019,12,25-Jun-
2019,20,08-Jul-2019,20,23-Jul-2019,40,.,.,0
Jones,101_002,25,M,12-Mar-2019,31-Mar-2019,02-Apr-2019,6,17-Apr-2019,12,29-Apr-2019,20,13-May-
2019,40,27-May-2019,80,11-Jun-2019,120,.,.,0
Smith,102_001,32,F,07-Apr-2019,20-Jun-2019,02-Jul-2019,6,16-Jul-2019,12,.,.,.,.,.,.,.,.,.,.,0
Smith,102_002,35,M,06-May-2019,28-May-2019,19-Jun-2019,6,01-Jul-2019,12,16-Jul-2019,6,28-Jul-
2019,12,.,.,.,.,.,.,0
Smith,102_003,27,M,09-May-2019,29-May-2019,10-Jun-2019,6,23-Jun-2019,40,11-Jul-
2019,20,.,.,.,.,.,.,.,.,0
Smith,102_004,24,F,18-Mar-2019,11-Apr-2019,15-Apr-2019,6,30-Apr-2019,12,12-May-2019,20,27-May-
2019,40,10-Jun-2019,80,26-Jun-2019,120,.,.,1
Jackson,103_001,34,M,15-Apr-2019,15-May-2019,20-May-2019,6,03-Jun-2019,12,03-Jun-2019,20,10-Jul-
2019,40,.,.,.,.,.,.,0
;
run;
*Formats--------------------------------------------------------------------;
*DateFMT is used to code text to red (problem), blue (future), black (#000000);
*ChgFMT is used to set background to orange (FFCC00), yellow (FFFF99) or green (AAFD8E) for dose
changes;
*Overdue dates have background set to cyan (00FFFF) - incorporated into ChgFmt format;
PROC FORMAT;
        value ChgFmt -1 = '#FFCC00'
                  0 = '#FFFF99'
                  1 = '#AAFD8E'
                                9 = '#00FFFF'
                  other= 'white';

        value DateFmt  0='#000000'
                                 1='red'
                                 2='blue'    ;
        run;

*Creating future visit dates for missing visits--------------------------------------------------
----------------;
data dset1;
        set dset;
        if DT1 ne . then do;
                if DT1_02 = . then DT1_02 = DT1+2*7;
                if DT1_04 = . then DT1_04 = DT1+4*7;
                if DT1_06 = . then DT1_06 = DT1+6*7;
                if DT1_08 = . then DT1_08 = DT1+8*7;
                if DT1_1  = . then DT1_1  = DT1+10*7;
                if DT2_36  = . then DT2_36  = DT1+36*7;
                end;
        run;

*Creating flags for completed, overdue/erroneous, or future visits-------------------------------
-----------------------------------;
data dset2;
        set dset1;
        sysdate="&sysdate.";
        if DT1  <= RANDODT then F1a=1;
        if  (DT1  <= ("&SYSDATE"d - 10) and D1  ne .) then F1b=1;
        if  (DT1  <= ("&SYSDATE"d - 10)  ) then F1b=1;
        if  (DT1  <= ("&SYSDATE"d - 10) and D1  ne .) then F1b=1;

        if DT1  <= RANDODT or (DT1  <= ("&SYSDATE"d - 10) and D1   = .) then F1 =1;
                else if  DT1  > ("&SYSDATE"d - 10) then F1 =2;
                else F1 =0;
        if DT1_02 <= DT1 or (DT1_02 <= ("&SYSDATE"d - 10) and D1_02 = .) then F1_02=1;
                else if  DT1_02 > ("&SYSDATE"d - 10) then F1_02=2;
                else F1_02=0;
        if DT1_04 <= max(D1, DT1_02) or (DT1_04 <= ("&SYSDATE"d - 10) and D1_04 = .) then
F1_04=1;
                else if  DT1_04 > ("&SYSDATE"d - 10) then F1_04=2;
                else F1_04=0;
        if DT1_06 <= max(D1, DT1_02, DT1_04) or (DT1_06 <= ("&SYSDATE"d - 10) and D1_06 = .) then
F1_06=1;
                else if  DT1_06 > ("&SYSDATE"d - 10) then F1_06=2;
                else F1_06=0;
        if DT1_08 <= max(D1, DT1_02, DT1_04, DT1_06) or (DT1_08 <= ("&SYSDATE"d - 10) and D1_08 =
.) then F1_08=1;
                else if  DT1_08 > ("&SYSDATE"d - 10) then F1_08=2;
                else F1_08=0;
        if DT1_1 <= max(D1, DT1_02, DT1_04, DT1_06, DT1_08) or (DT1_1 <= ("&SYSDATE"d - 10) and
D1_1 = .) then F1_1=1;
                else if  DT1_1 > ("&SYSDATE"d - 10) then F1_1=2;
                else F1_1=0;
        if DT2_36 <= max(D1, DT1_02, DT1_04, DT1_06, DT1_08, DT1_1) or (DT2_36 <= ("&SYSDATE"d -
10) and D2_36 = .) then F2_36=1;
                else if  DT2_36 > ("&SYSDATE"d - 10) then F2_36=2;
                else F2_36=0;
        run;
 proc print data=dset2 width=min;run;
*Creating flags for change in dose level---------------------------------------------------------
---------;
data dset3;
        set dset2;
        If D1_02 > D1  > . then C1_02 = 1;
                else if D1_02 = D1  > .  then C1_02 = 0;
                else if . < D1_02 < D1  then C1_02 = -1;
        If D1_04 > D1_02 > . then C1_04 = 1;
                else if D1_04 = D1_02 > .  then C1_04 = 0;
                else if . < D1_04 < D1_02 then C1_04 = -1;
        If D1_06 > D1_04 > . then C1_06 = 1;
                else if D1_06 = D1_04 > .  then C1_06 = 0;
                else if . < D1_06 < D1_04 then C1_06 = -1;
        If D1_08 > D1_06 > . then C1_08 = 1;
                else if D1_08 = D1_06 > .  then C1_08 = 0;
                else if . < D1_08 < D1_06 then C1_08 = -1;
        If D1_1 > D1_08 > . then C1_1 = 1;
                else if D1_1 = D1_08 > .  then C1_1 = 0;
                else if . < D1_1 < D1_08 then C1_1 = -1;
        If D2_36 > D1_1 > . then C2_36 = 1;
                else if D2_36 = D1_1 > .  then C2_36 = 0;
                else if . < D2_36 < D1_1 then C2_36 = -1;
        run;
```

```
*Creating flags for overdue visits---------------------------------------------------------------
---;
data dset4;
        set dset3;
        If DT1    <= "&sysdate"d - 10 and missing (D1 )   then OV1  = 9;
        If DT1_02 <= "&sysdate"d - 10 and missing (D1_02) then OV1_02 = 9;
        If DT1_04 <= "&sysdate"d - 10 and missing (D1_04) then OV1_04 = 9;
        If DT1_06 <= "&sysdate"d - 10 and missing (D1_06) then OV1_06 = 9;
        If DT1_08 <= "&sysdate"d - 10 and missing (D1_08) then OV1_08 = 9;
        If DT1_1  <= "&sysdate"d - 10 and missing (D1_1 ) then OV1_1  = 9;
        If DT2_36  <= "&sysdate"d - 10 and missing (D2_36) then OV2_36  = 9;
        run;
proc sort data=dset4 out=FINAL;
        by SUBJID;
        run;

*Creating output template------------------------------------------------------------------;
*  Create a custom STYLE based on the stock "sansPrinter" STYLE;
 proc template;
   define STYLE STYLEs.XLsansPrinter;
   parent=STYLEs.sansprinter;
   STYLE header from header /  font_size = 10pt  just= center  vjust= bottom;             /*
Change attributes of the column headings  */
   STYLE notecontent from notecontent /
       bordertopwidth    = 1     borderbottomwidth = 1
       borderleftwidth   = 1     borderrightwidth  = 1;
               /*  Change attributes of the blank lines  */
    STYLE data_bold from data /    font_weight = bold;
        /*  Used to create bold text  */
    STYLE data_center from data /   just = center;
               /*  Used to center text  */
    STYLE data_date9 from data /  tagattr='format:m/d/yyyy type:DateTime';              /*
Used to convert SAS date values to Excel date values  */
   end;
 run; quit;

/* Need to update the fonts. Excel does not support the sans-serif fomt. */
  proc template;
    define style styles.xlsansprinter;
      parent=styles.sansprinter;
      replace fonts /
        'TitleFont2' = ("Arial",10pt,Bold)
        'TitleFont' = ("Arial",10pt,Bold)
        'StrongFont' = ("Arial",10pt,Bold)
        'EmphasisFont' = ("Arial",10pt, Bold)
        'FixedEmphasisFont' = ("Arial",10pt)
        'FixedStrongFont' = ("Arial",10pt,Bold)
        'FixedHeadingFont' = ("Arial",10pt,Bold)
        'BatchFixedFont' = ("Arial",10pt)
        'FixedFont' = ("Arial",10pt)
        'headingEmphasisFont' = ("Arial",10pt,Bold )
        'headingFont' = ("Arial",10pt,Bold)
        'docFont' = ("Arial",10pt);
      STYLE header from header /  font_size = 10pt  just= center  vjust= bottom;          /*
Change attributes of the column headings  */
      STYLE notecontent from notecontent /
            bordertopwidth    = 1     borderbottomwidth = 1
            borderleftwidth   = 1     borderrightwidth  = 1;
               /*  Change attributes of the blank lines  */
      STYLE data_bold from data /    font_weight = bold;
               /*  Used to create bold text  */
      STYLE data_center from data /   just = center;
                      /*  Used to center text  */
      STYLE data_date9 from data /  tagattr='format:m/d/yyyy type:DateTime';
        /*  Used to convert SAS date values to Excel date values  */
      end;
    run;
  quit;
 OPTIONS MISSING=' ';
 *Create Proc Report output (in macro loop to easily split into completed/ET and ongoing
subjects)--------------------------------------------;


%macro ToReport(completerclass);
 proc report data= FINAL split='*' nowindows;
       WHERE OFFSTUDY=&completerclass;
       column ('SUBJECT INFORMATION/VISITS' PI subjid age sex  icdt randodt );
       column ('VISIT 1'                     DT1     D1);
       column ('UPDOSING'
                         DT1_02  D1_02
                         DT1_04      D1_04
                         DT1_06      D1_06
                         DT1_08      D1_08
                         DT1_1       D1_1  );
       column ('VISIT 03/EXIT'    DT2_36     D2_36 );
       column  F1                            OV1
                      F1_02  C1_02   OV1_02
                      F1_04  C1_04   OV1_04
                      F1_06  C1_06   OV1_06
                      F1_08  C1_08   OV1_08
                      F1_1   C1_1            OV1_1
                      F2_36  C2_36   OV2_36                 ;
        column dummy; *  Dummy column to perform traffic lighting;

 *  'ID' columns;
       define PI              / display                            "PI"

       style(header)=[backgroundcolor=CX90B0D9 ];
       define Subjid          / display order order=internal       "Subject ID"
       STYLE(Column)=data_bold       style(header)=[backgroundcolor=CX90B0D9 ];
       define age             / display                     "Age (yrs)"
       STYLE(Column)=data_center     style(header)=[backgroundcolor= CX8A5AA6 ];
       define sex             / display                     "Sex"

       style(header)=[backgroundcolor=CX90B0D9 ];
       define icdt            / display                     "Date of Informed Consent"
                                     style(header)=[backgroundcolor=CX90B0D9 ];
       define randodt         / display                     "Randomization Date"
                                             style(header)=[backgroundcolor=CX90B0D9 ];

 *  Data columns: doses;
       define D1 / display           "Dose" style(header)=[backgroundcolor=CX90B0D9 ]
style(column)=[just=center] ;
       define D1_02 / display        "Wk 2 dose"     style(header)=[backgroundcolor=CXE5A099 ]
style(column)=[just=center];
       define D1_04 / display        "Wk 4 dose"  style(header)=[backgroundcolor=CX90B0D9 ]
style(column)=[just=center];
       define D1_06   / display      "Wk 6 dose"     style(header)=[backgroundcolor=CXE5A099 ]
style(column)=[just=center];
       define D1_08   / display      "Wk 8 dose"  style(header)=[backgroundcolor=CX90B0D9 ]
style(column)=[just=center] ;
       define D1_1   / display       "Wk 10 dose"    style(header)=[backgroundcolor=CXE5A099 ]
style(column)=[just=center];
       define D2_36   / display      "Wk 36 dose" style(header)=[backgroundcolor=CX90B0D9 ]
style(column)=[just=center];

 *  Hidden columns containing the range flags: flags for changes in dose;
       define C1_02 / display noprint;
       define C1_04 / display noprint;
       define C1_06   / display noprint;
       define C1_08   / display noprint;
       define C1_1   / display noprint;
       define C2_36   / display noprint;

 *  Data columns: dates ;* STYLE(Column)=data_date9;
       define DT1 / display                  "First Dose Date"
style(header)=[backgroundcolor=CX90B0D9 ];
       define DT1_02 / display       "Week 2 date"    style(header)=[backgroundcolor=CXE5A099 ];
       define DT1_04 / display       "Week 4 date"  style(header)=[backgroundcolor=CX90B0D9 ];
       define DT1_06   / display   "Week 6 date"    style(header)=[backgroundcolor=CXE5A099 ];
       define DT1_08   / display   "Week 8 date"  style(header)=[backgroundcolor=CX90B0D9 ];
       define DT1_1   / display      "Week 10 date"    style(header)=[backgroundcolor=CXE5A099
];
       define DT2_36   / display   "Week 36 date" style(header)=[backgroundcolor=CX90B0D9 ];

 *  Hidden columns containing the range flags: flags for dates;
       define F1 / display noprint;
       define F1_02 / display noprint;
       define F1_04 / display noprint;
       define F1_06   / display noprint;
       define F1_08   / display noprint;
       define F1_1   / display noprint;
       define F2_36   / display noprint;

       define OV1 / display noprint;
       define OV1_02 / display noprint;
       define OV1_04 / display noprint;
       define OV1_06   / display noprint;
       define OV1_08   / display noprint;
       define OV1_1   / display noprint;
       define OV2_36   / display noprint;

 *  Dummy column to perform traffic lighting;
       define dummy / computed noprint;

 *  Traffic light the data columns based on the hidden columns;
 compute dummy;
 call define('D1_02', 'STYLE', 'STYLE=[background= '||put(C1_02, chgfmt.) || ']');
 call define('D1_04', 'STYLE', 'STYLE=[background= '||put(C1_04, chgfmt.) || ']');
 call define('D1_06', 'STYLE', 'STYLE=[background= '||put(C1_06, chgfmt.) || ']');
 call define('D1_08', 'STYLE', 'STYLE=[background= '||put(C1_08, chgfmt.) || ']');
 call define('D1_1' , 'STYLE', 'STYLE=[background= '||put(C1_1 , chgfmt.) || ']');
 call define('D2_36', 'STYLE', 'STYLE=[background= '||put(C2_36, chgfmt.) || ']');

 call define('DT1',    'STYLE', "STYLE=[ foreground=" ||put(F1, datefmt.)    || "background="
||put(OV1, chgfmt.) || ']');
 call define('DT1_02', 'STYLE', "STYLE=[ foreground=" ||put(F1_02, datefmt.) || "background="
||put(OV1_02, chgfmt.) || ']');
 call define('DT1_04', 'STYLE', "STYLE=[ foreground=" ||put(F1_04, datefmt.) || "background="
||put(OV1_04, chgfmt.) || ']');
 call define('DT1_06', 'STYLE', "STYLE=[ foreground=" ||put(F1_06, datefmt.) || "background="
||put(OV1_06, chgfmt.) || ']');
 call define('DT1_08', 'STYLE', "STYLE=[ foreground=" ||put (F1_08, datefmt.) || "background="
||put(OV1_08, chgfmt.) || ']');
 call define('DT1_1' , 'STYLE', "STYLE=[ foreground=" ||put (F1_1,  datefmt.) || "background="
||put(OV1_1,  chgfmt.) || ']');
 call define('DT2_36', 'STYLE', "STYLE=[ foreground= "||put( F2_36, datefmt.) || "background="
||put(OV2_36, chgfmt.) || ']');

 endcomp;
 run; quit;
 %mend ToReport;
options mprint mlogic symbolgen;
*Producing output ------------------------------------------------------------------------------
-------;
 **** output location *******;
 ods tagsets.ExcelXP path="\\SERVER-W7-1\Server Documents\Bowden Analytics\SESUG\_My talks\"
                                                    file='SESUG_HP_260_2019_Example.xml'
STYLE=XLsansprinter;

  *  Create the tracker page for on-study subjects  ;
 ods tagsets.ExcelXP options(sheet_name="Enrollment-Dose Tracking "
                             suppress_bylines='yes'
                             absolute_column_width='6,6,4,4,8,12, 8,4,8,4,8,4,8,4,8,4,8,4,8,4'
                             autofit_height='yes'
                             frozen_headers='yes' );

%ToReport(0);  *call Proc Report for on-study subjects;

  *  Create the Early Term/ Completed Subjects page  ;
 ods tagsets.ExcelXP options(sheet_name='Completed or ET'
                             suppress_bylines='yes'
                             absolute_column_width='6,6,4,4,8,12, 8,4,8,4,8,4,8,4,8,4,8,4,8,4'
                             autofit_height='yes'
                             frozen_headers='yes' );

%ToReport(1);  *call Proc Report for off-study subjects;

ods tagsets.ExcelXP close;
```