PharmaSUG 2023 - Paper QT-234

Using P21 Checks to Help Your DM Out

Chad Fewell, Deciphera Pharmaceuticals;
Jesse Beck

ABSTRACT

There is a lot of work that goes into the cleaning of data. Data Management (DM) issues queries on an ongoing basis and Statistical Programming (SP) escalates data issues they see within their programs. DM will create edit checks within the database to help with cleaning. In addition, they manually review the data on an ongoing basis looking for data that appears incorrect. Statistical Programming works to program raw data into SDTMs checking for data issues along the way. There appears to be a "gap" in data issues that are looked at from a continuous basis to issues that are discovered prior to submission. This is where Pinnacle 21 comes into play (P21). P21 uses SDTMs to help clean the data, but sometimes due to resourcing SDTMs are not completed until close to study termination. This causes a delay in issues that are difficult to find manually by DM but are essential to be cleaned prior to Database Lock (DBL). "How do we help avoid this delay in checking more complex data issues?" one may ask. P21 provides an extensive list of all the checks they are looking for along with the severity of the data issues. The purpose of this paper will detail how to execute P21 checks to supplement the database edit checks as well as all the ongoing data cleaning processes to provide a detailed user-friendly report to DM of data issues that can have a wide range in complexity typically not seen on an ongoing basis.

INTRODUCTION

This paper will serve as a brief glimpse into how statistical programmers can utilize Pinnacle 21 (P21) in a simple manner without having to develop SDTMs to help data management clean the database data more effectively. P21 is a validation report, which utilizes SDTMs, that all data points must go through to determine the integrity of the data being submitted for publication. Instead of waiting until the programmers submit the datasets into P21, it would benefit all parties if the data was proactively and routinely checked for P21 issues without waiting to run the P21 report.

The purpose of this paper is not to include every P21 check on the raw data. That would not be an efficient use of time especially because P21 already does this for us. The purpose of this paper is to include checks that can be quickly found without running the entire P21 report. In addition, there may be some critical data points we want to keep an eye on without having to run the entire P21 report. Then finally we want to present the issues in a user-friendly form that allows DM to digest the issues without having to be fluent in what the P21 report is or looks at.

DETERMINE WHAT DATA TO CHECK

There are several factors that can help determine what kind of checks to make. Some could be based off primary and secondary end points of the trial. Depending on how much of a priority the trial is, whether its Phase I/II/III, will dictate how many and the complexity of the checks needed to be routinely checked. We also do not want to duplicate already existing database checks unless one finds it necessary to do so. Re-checking database checks may not an effective use of a person’s time or an effective use of the point of this paper. However, there are some P21 checks that databases simply cannot do that need to be created that are programmatically possible prior to SDTMs. These checks can be utilized to help clean the data, or if necessary, make sure the database checks are catching the issues they need to.

RUN CHECKS

Let us illustrate this approach on the AE CRF. There are typical checks built into the AE CRF such as making sure AE end date occurs after AE start date, if end date is present then AE outcome has a status of resolved or something similar, or AE start/end dates overlap with one another. However, there are some P21 checks we can add on top of these database checks.

For example, let us look at P21 check SD1254 if an AE has an outcome of ‘Fatal’ we would expect for there to be a corresponding death information within the database, which in the SDTM DM domain derives the DTHFL. Typically, there are no cross-reference checks for this check within the database. Programmatically we can add these to create a report of possible data issues. First, we pull in the raw AE data. Then we pull in the Death data. We cross check that the information and create an issue to be followed up on, if true:

```
*************************************************************************;
*** Rule ID# SD1254 DTHFL does not equal 'Y' when AE.AEOUT = 'FATAL'
*************************************************************************;

data ae;
  set raw.ae;
  where AEOUT = ‘Fatal’;
run;

data death;
    set raw.death;
    DDORRES = ‘Adverse Event’;
run;

proc sort data=ae; by subjid; run;
proc sort data=death; by subjid; run;

data ae_chk;
    merge ae(in=a) death(in=b);
    by subjid;
    if a but not b then Issue = ‘Death Must be present when an Adverse
    Event results in death’;
run;
```

This is a very bare bone example of a simple check to make. Another cross-reference check that can be made that isn’t too complicated is checking P21 rule SD0080 “AE start date is after the latest disposition date.” This one is dependent on where the end point is expected to be captured within the database, and could look like the following:

```
**********************************************************************;
*** Rule ID# SD0080 AE start date is after the latest Disposition date;
**********************************************************************;

data eos;
   set raw.eos;
   keep subjid DSDAT;
run;

proc sort data=ae; by subjid; run;
proc sort data=eos; by subjid; run;

data ae_2;
   merge ae eos;
   by subjid;
run;

data ae_2;
   set ae_2;
   if DSDAT ne . then do;
```

```
     if input(strip(AESTDAT),yymmdd10.) > input(strip(DSDAT),yymmdd10.)
     then do;
        Issue = 'Start Date/Time of Adverse Event (AESTDTC) should be
        less than or equal to the Start Date/Time of the latest
        Disposition Event (DSSTDTC).';
        TYPE = 'WARNING';
        output;
     end;
  end;
run;
```

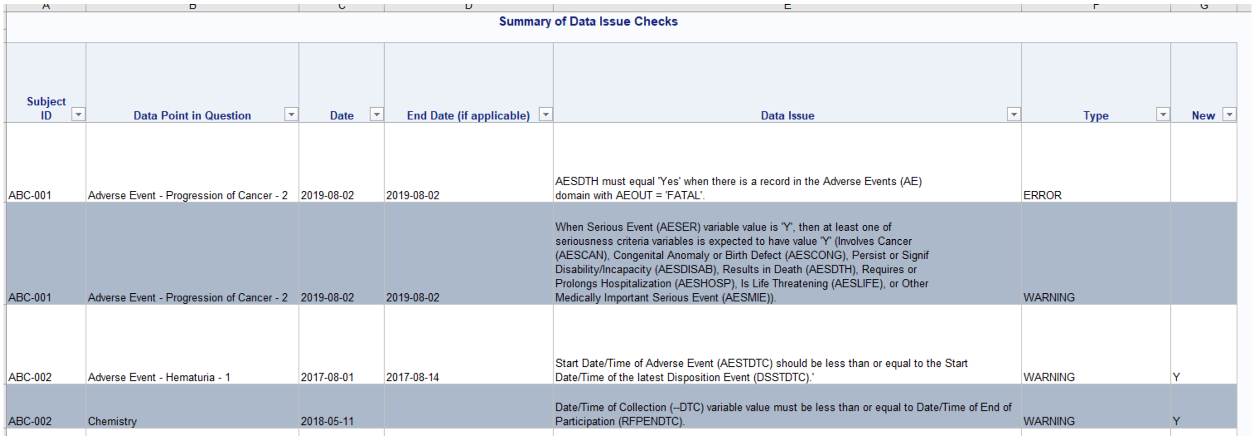
In this example we added some things to help with the reports readability. We include the variable TYPE to provide the type of warning P21 would issue. In this instance it would just be a “Warning” about the data issue. In addition, to help DM locate the issue we keep the AE, AE#, and the start date it occurred for the patient.

CREATE REPORT

After all the checks have been determined and code as has been created, we need to present the information in a way that can be easily digested by DM. To do this, we create an excel output that details this information. In addition, a variable ‘NEW’ is previously derived and has been added to the final output. Since the report is routinely ran it would be helpful if the reviewer could look and determine which findings are new and which findings are outstanding since last run.

```
ods excel file=<location>\P21_Data_Checks.xlsx' style=HTMLBlue
options(embedded_titles='yes' suppress_bylines='yes');

title 'Summary of Data Issue Checks';
footnote;
ods excel options(sheet_name='P21' autofilter='1-7');

proc report data=check nowindows split='*';

  columns SUBJID TERM STDAT ENDAT ISSUE TYPE NEW;
  define SUBNJID / display 'Subject ID';
  define TERM / display 'Data Point in Question';
  define STDAT / display 'Start Date';
  define ENDAT / display 'End Date';
  define ISSUE / display 'Data Issue';
  define TYPE / display 'FDA Severity';
  define NEW / display 'NEW';

  compute SUBJID;
  * Change background color for alternating rows;
      RowNum+1;
      if (mod(RowNum, 2) eq 0) then call define(_row_, 'style',
      'style=[background=#acb9ca]');
   endcomp;
quit;

ods excel close;
```

Display 1



Summary of Data Issue Checks

Subject ID	Data Point in Question	Date	End Date (if applicable)	Data Issue	Type	New
ABC-001	Adverse Event - Progression of Cancer - 2	2019-08-02	2019-08-02	AESDTH must equal 'Yes' when there is a record in the Adverse Events (AE) domain with AEOUT = 'FATAL'.	ERROR	
ABC-001	Adverse Event - Progression of Cancer - 2	2019-08-02	2019-08-02	When Serious Event (AESER) variable value is 'Y', then at least one of seriousness criteria variables is expected to have value 'Y' (Involves Cancer (AESCAN), Congenital Anomaly or Birth Defect (AESCONG), Persist or Signif Disability/Incapacity (AESDISAB), Results in Death (AESDTH), Requires or Prolongs Hospitalization (AESHOSP), Is Life Threatening (AESLIFE), or Other Medically Important Serious Event (AESMIE)).	WARNING	
ABC-002	Adverse Event - Hematuria - 1	2017-08-01	2017-08-14	Start Date/Time of Adverse Event (AESTDTC) should be less than or equal to the Start Date/Time of the latest Disposition Event (DSSTDTC).'	WARNING	Y
ABC-002	Chemistry	2018-05-11		Date/Time of Collection (--DTC) variable value must be less than or equal to Date/Time of End of Participation (RFPENDTC).	WARNING	Y

Here we see a formatted table that can easily be used and sorted by any reviewer to see issues within the database. This allows the viewer(s) to proactively see all current P21 issues that can be queried and fixed, and the viewer(s) can sort based on the "New" column to see what newer datapoints need to be queried on. The issues text currently listed with the report are verbatim the P21 text. If one wanted the report to be even more readable, they could amend the "Data Issue" text to use more common terms of what the data issue is about. For example, instead of "Date/Time of Collection (--DTC) variable value must be less than or equal to Date/Time of End of Participation (RFPENDTC)" it could be updated to "Chemistry test occurs after patient ended study participation."

CONCLUSION

This illustrates a quick and easy way to programmatically generate a report that points out P21 data issues using raw data without having to create the SDTMs to submit into P21. The amount of detail and work a user wants to put into such a report is completely up to them. This type of report will be greatly beneficial to DM to lower manual review time. The report also has a benefit to statistical programmers as they can see upcoming P21 issues prior to SDTM creation and would lower the findings produced within the P21 report. In addition, the idea is not limited to just P21 checks. Any data checks that DM would like to have automated could be done as well, with the approval of the report creator. The primary goal of a report like this is to clean the data on an ongoing basis in a quick and efficient manner. This ultimately will help produce data with the upmost integrity heading into data submissions or publications.

REFERENCES

Pinnacle 21 (https://www.pinnacle21.com/validation-rules/sdtm).

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Chad Fewell
Deciphera Pharmaceuticals
(615)887-3001
Chadkfewell@gmail.com

Jesse Beck
(615)631-1984
Jesse.Beck711@gmail.com