PharmaSUG China 2021 - Paper AD-075

Automated Process from aCRF to ADRG for SAS programming with Python and VBA

Weineng Zhou, Parexel International, Shanghai, China

ABSTRACT

According to the FDA and PMDA data submission requirements or NMPA suggestions, the following documents should be submitted: aCRF, SDTM datasets, ADaM dataset, SDTM Define.xml, ADaM Define.xml, Study Data reviewer's guide (SDRG), Analysis Data Reviewer's Guide (ADRG). We know that SDTM datasets and ADaM dataset are created by SAS programming. aCRF, define.xml, SDRG, ADRG are always implemented manually. As we can see, SAS programming is a time-consuming and labor-intensive project since it takes a lot of time to complete all the works above for electronic submission. Now, we could try to use Python and VBA to automatically annotate unique blank CRF, generate complex algorithm documents, identify the page number where the SDTM variable was located in CRF, and automatically fill the identified page number in the specification if the origin is CRF. Also, we need to write sdrg and adrg documents after the dataset and define.xml were completed, most of the tables in the document can be automatically generated by Python.And we have developed related tools for graphical user interaction.

INTRODUCTION

We know that the following documents need to be delivered to the FDA/PMDA/NMPA for clinical trials data submission, including annotations CRF, sdtm datasets, sdtm define.xml, ADaM data sets, ADaM define.xml, study data review guide (sdrg), analysis data review guide (adrg), tables, listings and graphs (TFL), As can be seen from Fig.1, There is a lot of work to do for SAS programmers, involving programming and documentations. Some are quite time-consuming. For example, it always took lots of time to annotate CRF to map all the SDTM variables . Alos, when preparing the sdtm spec or define.xml, corresponding page numbers of the variables need to be listed when origins of the variables are from CRF. For most programmers, they actually prefer to write code instead of documentation, such as annotating CRFs, filling out spec, writing sdrg, adrg, and so on. To reduce the time consumed at documentation, I optimized the workflow which could improve efficiency and reduce the workload of SAS programmers,so that they can be proactive in learning new technologies and be more productive, then they have more time toenjoy coffee.

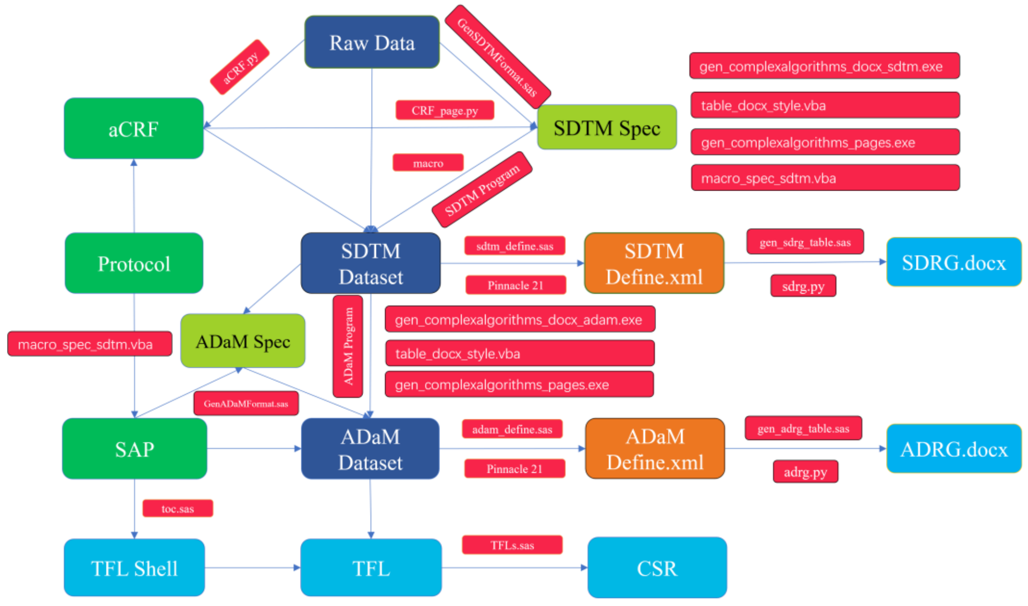


Fig.1 Task Flow Chart

MATERIAL AND METHOD

1. AUTOMATICALLY ANNOTATE CRF

Clinical data are always collected by CRF. The database designer will assign some field name to collect the informationneeded for data analysis. The field name is always used to indicate how to fill in the result in the EDC system. The filed name would be a variable name in raw data after data extraction from EDC. In general, the raw data exported from EDC system does not always meet the CDISC standard, and we need to map the raw data to SDTM following SDTM Implementation Guide. Firstly, we need to annotate blank CRF with SDTM variables. We assume that draft sdtm specification with the standard SDTM variables and the SUPP variable is ready here. It's enough if origins just indicate 'CRF', 'eDT', 'Assign' and so on. Obviously it will not affect our CRF annotation if only the CRF pages are left blank. It would take a lot of time to fill it out manually, so we developed a tool using python with the interface shown below. All we need to do is to put in a blank CRF and the draft spec.xlsx. The tool will help usautomatically generate aCRF with SDTM variables. The accuracy and completion rate is about 80%-90% after testing in my project. The procedure is following: First, dataset with SDTM or SUPP variable and label will be generated from sepc with python. Second, Loop through each variable label and each field information to calculate text similaritybetween field information on the CRF and the SDTM or SUPP variable label, If the similarity between field label and variable label Reach 70% more or less, then Python program will get the coordinates of the field label. Finally annotation text filled in variable name will be added in blank CRF based on coordinates with python module pdf_annotate.

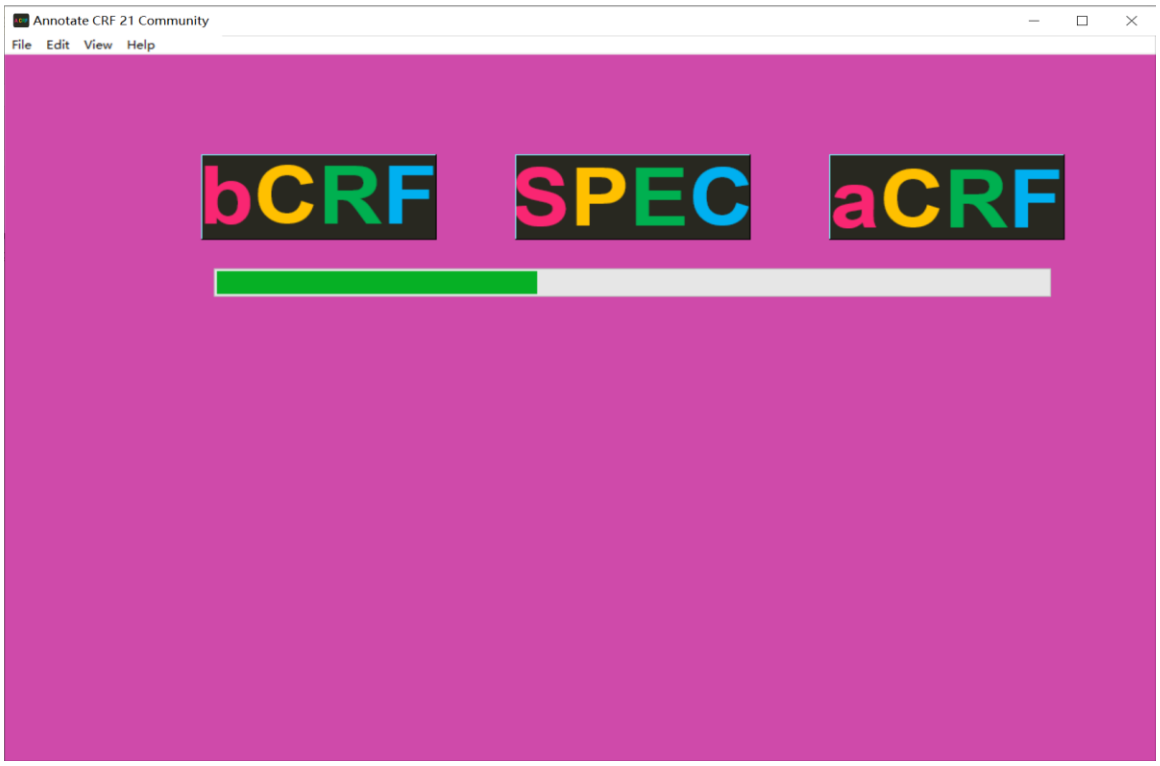
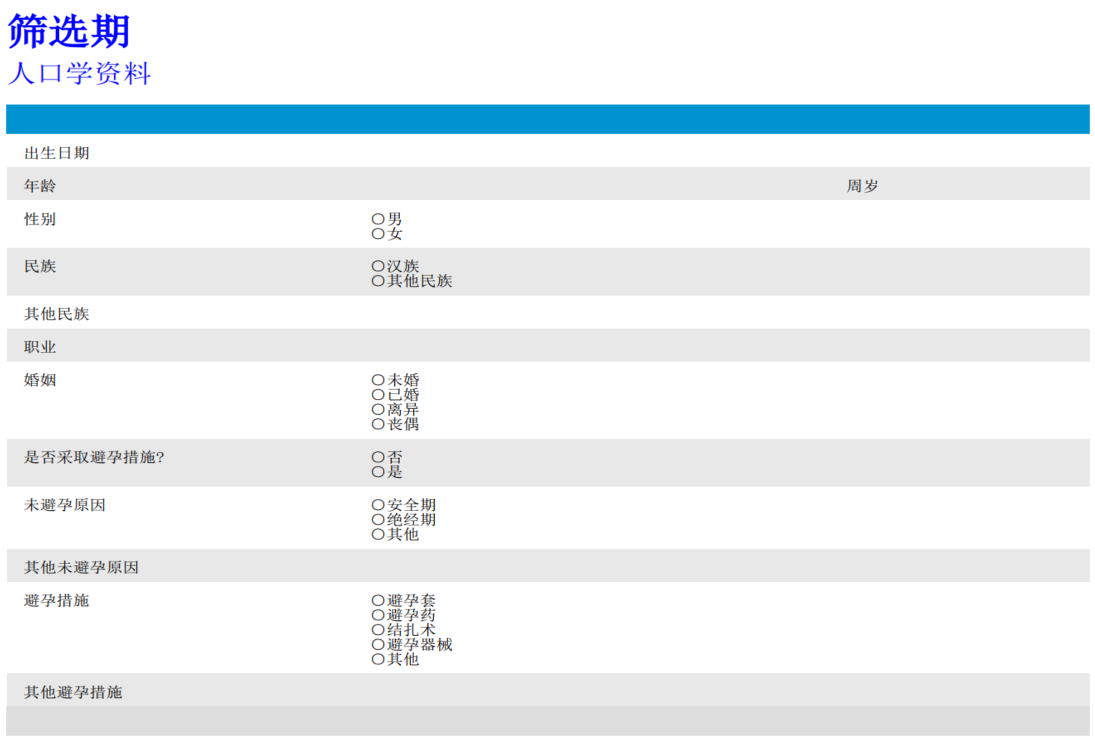


Fig.2 Automatically annotate CRF software

2. FROM BLANKCRF.PDF TO ACRF.PDF

It will take about 1-2 minutes to annotate a 64-page Unique Blank CRF. The result is showed as below, Fig.3 was the original Unique Blank CRF, Fig.4 was the CRF annotated automatically by the tool. Almost everything is accurate expect ethnic field. An extra commenting was on other ethnic field but it should not be there. We could make some update manually if needed.As to the style of the text box, such as borders and fill colors, it could be edited later by the PDF editor. As a conclusion, the use of automated annotation CRF tool can greatly improve the efficiency of SAS programmers.

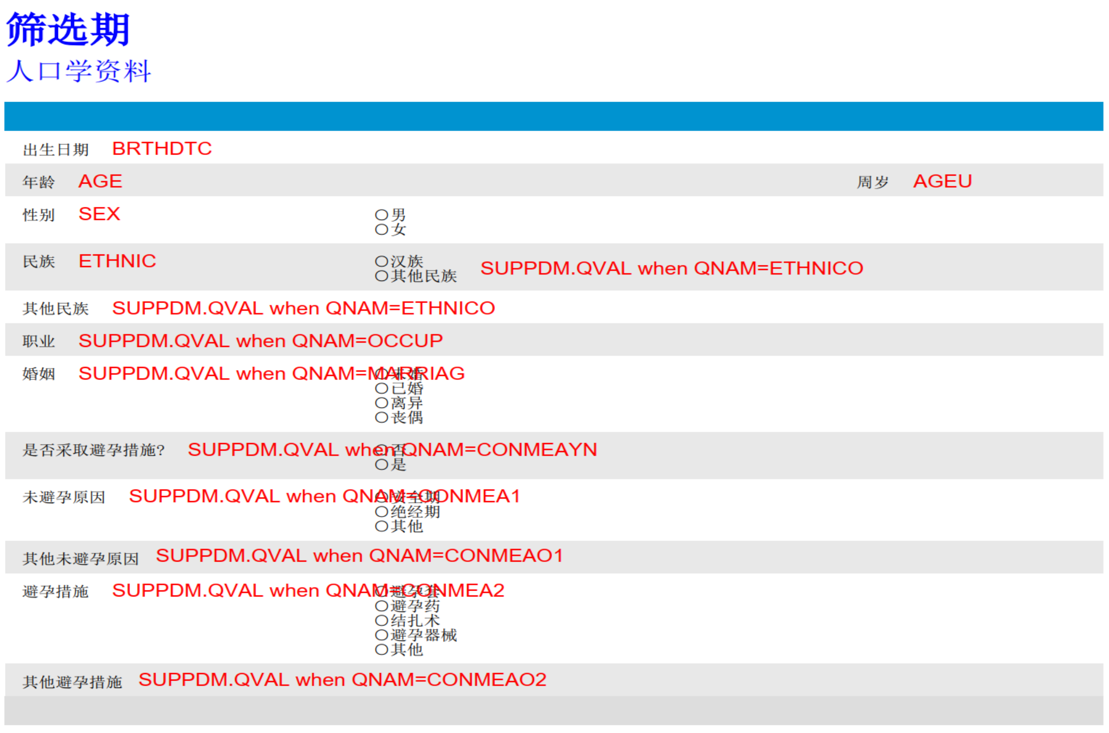


筛选期

人口学资料

出生日期		
年龄		周岁
性别	○男 ○女	
民族	○汉族 ○其他民族	
其他民族		
职业		
婚姻	○未婚 ○已婚 ○离异 ○丧偶	
是否采取避孕措施?	○否 ○是	
未避孕原因	○安全期 ○绝经期 ○其他	
其他未避孕原因		
避孕措施	○避孕套 ○避孕药 ○结扎术 ○避孕器械 ○其他	
其他避孕措施		

Fig.3 Unique Blank CRF



筛选期

人口学资料

出生日期 BRTHDTC		
年龄 AGE		周岁 AGEU
性别 SEX	○男 ○女	
民族 ETHNIC	○汉族 ○其他民族	SUPPDM.QVAL when QNAM=ETHNICO
其他民族 SUPPDM.QVAL when QNAM=ETHNICO		
职业 SUPPDM.QVAL when QNAM=OCCUP		
婚姻 SUPPDM.QVAL when QNAM=MARRIAG	○未婚 ○已婚 ○离异 ○丧偶	
是否采取避孕措施? SUPPDM.QVAL when QNAM=CONMEAYN	○否 ○是	
未避孕原因 SUPPDM.QVAL when QNAM=CONMEA1	○安全期 ○绝经期 ○其他	
其他未避孕原因 SUPPDM.QVAL when QNAM=CONMEAO1		
避孕措施 SUPPDM.QVAL when QNAM=CONMEA2	○避孕套 ○避孕药 ○结扎术 ○避孕器械 ○其他	
其他避孕措施 SUPPDM.QVAL when QNAM=CONMEAO2		

Fig.4 annotated CRF by software

3. AUTOMATICALLY ANNOTATE CRF BY MERGE XFDF OR XML

For similar studies, we can create annotated CRFs for a new project by referring to previously annotated CRFs. The comment text in PDF can be exported as an xfdf or xml file. We can use the Python standard module PDFNetPython3.PDFNetPython to realize it. And then we use the PDFMerge xfdf method to copy the comments in previously similar studies to the new Unique blank CRF. We can even copy the comments completely if the page numbers of the two PDF files and the structure are the same ideally. The comments for most pages can be copied to the new Unique blank CRF corresponding page in this way if most pages are the same. As shown in Fig.5 , we have developed tool which applies only to very similar studies. We just need to import an old comment CRF and a similar blank CRF then we can get a new annotated CRF. Also, we need to make minor update if needed.

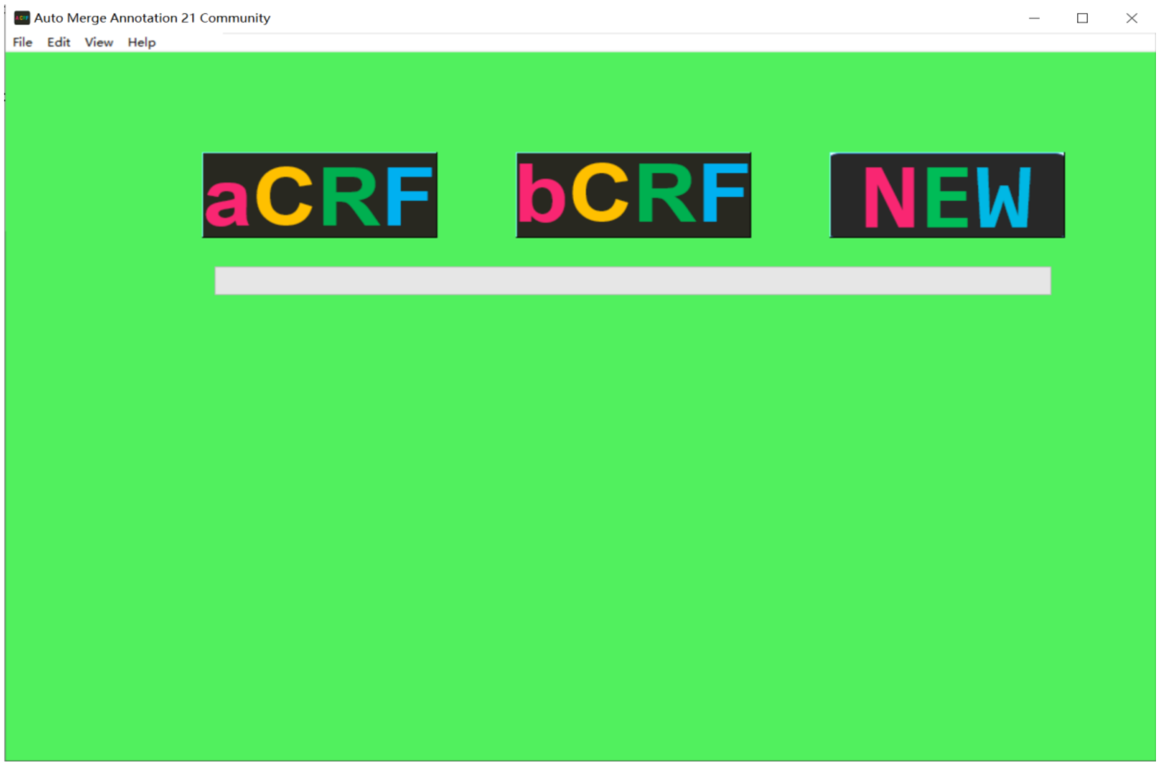
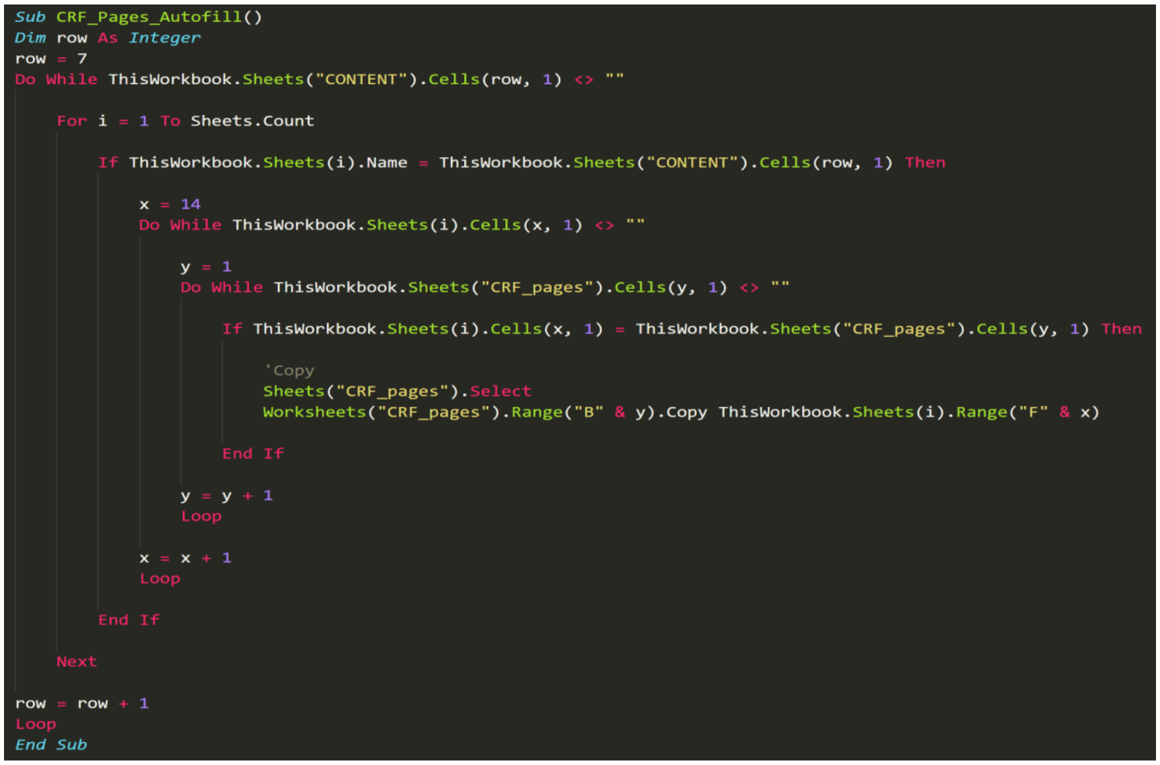


Fig.5 Automatically annotate CRF software by merge XFDF

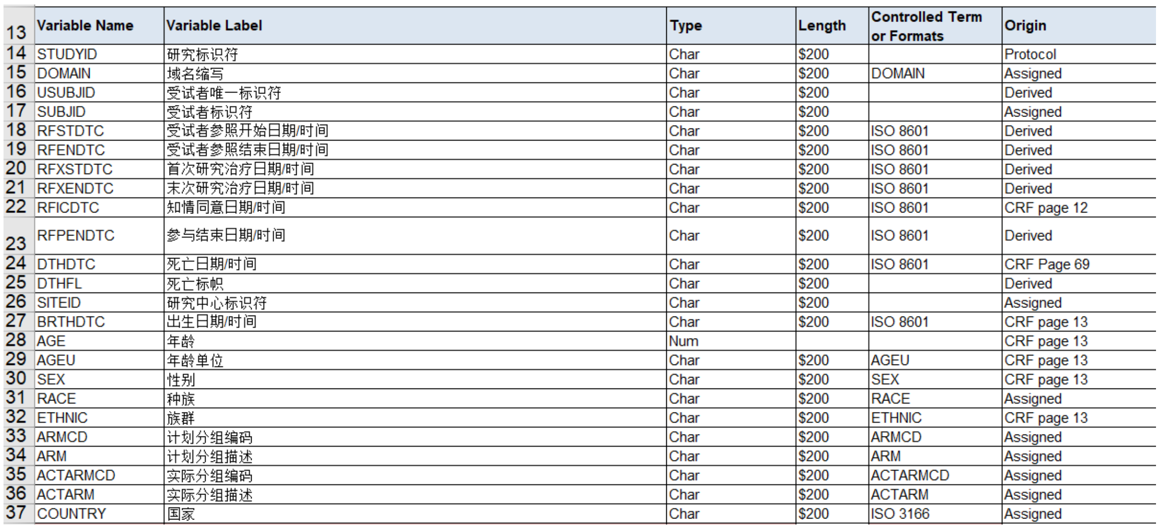
4. AUTOMATICALLY GENERATE SDTM VARIABLE PAGES

After the CRF annotation completed, we need to find and fill in the page number of these variables to origin column in spec. assuming that there are 100 variables which have origins from CRF, and the CRF has 100 pages,obviously it is quite a time-consuming work if we need to search page numbers from the blank CRF for each of the variables. If we use the tool to search automatically, it'll take only dozens of minutes even though it needs to be searched 10,000 times for the program, Fig 3 is an automated tool to search for variable page numbers developed by Python, we only need to enter the annotated CRF, spec with variables and label filled out, it will automatically insert a sheet on spec after the program runs, with variables from CRF and its corresponding page number. We just need to run a VBA code to fill in the page number of the variable into the spec column automatically.

Fig.6 Automatically generate CRF Variable pages software



```
Sub CRF_Pages_Autofill()
Dim row As Integer
row = 7
Do While ThisWorkbook.Sheets("CONTENT").Cells(row, 1) <> ""

    For i = 1 To Sheets.Count

        If ThisWorkbook.Sheets(i).Name = ThisWorkbook.Sheets("CONTENT").Cells(row, 1) Then

            x = 14
            Do While ThisWorkbook.Sheets(i).Cells(x, 1) <> ""

                y = 1
                Do While ThisWorkbook.Sheets("CRF_pages").Cells(y, 1) <> ""

                    If ThisWorkbook.Sheets(i).Cells(x, 1) = ThisWorkbook.Sheets("CRF_pages").Cells(y, 1) Then

                        'Copy
                        Sheets("CRF_pages").Select
                        Worksheets("CRF_pages").Range("B" & y).Copy ThisWorkbook.Sheets(i).Range("F" & x)

                    End If

                y = y + 1
                Loop

            x = x + 1
            Loop

        End If

    Next

row = row + 1
Loop
End Sub
```

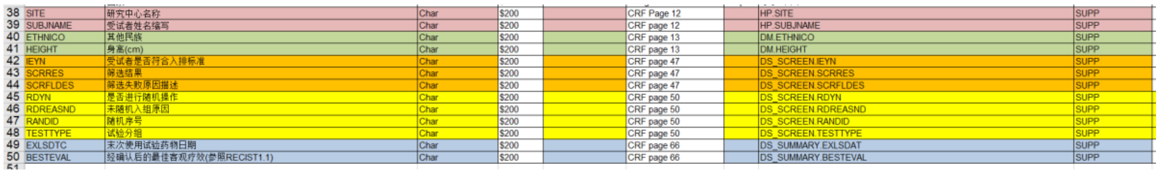


13	Variable Name	Variable Label	Type	Length	Controlled Term or Formats	Origin
14	STUDYID	研究标识符	Char	$200		Protocol
15	DOMAIN	域名缩写	Char	$200	DOMAIN	Assigned
16	USUBJID	受试者唯一标识符	Char	$200		Derived
17	SUBJID	受试者标识符	Char	$200		Assigned
18	RFSTDTC	受试者参照开始日期/时间	Char	$200	ISO 8601	Derived
19	RFENDTC	受试者参照结束日期/时间	Char	$200	ISO 8601	Derived
20	RFXSTDTC	首次研究治疗日期/时间	Char	$200	ISO 8601	Derived
21	RFXENDTC	末次研究治疗日期/时间	Char	$200	ISO 8601	Derived
22	RFICDTC	知情同意日期/时间	Char	$200	ISO 8601	CRF page 12
23	RFPENDTC	参与结束日期/时间	Char	$200	ISO 8601	Derived
24	DTHDTC	死亡日期/时间	Char	$200	ISO 8601	CRF Page 69
25	DTHFL	死亡标帜	Char	$200		Derived
26	SITEID	研究中心标识符	Char	$200		Assigned
27	BRTHDTC	出生日期/时间	Char	$200	ISO 8601	CRF page 13
28	AGE	年龄	Num			CRF page 13
29	AGEU	年龄单位	Char	$200	AGEU	CRF page 13
30	SEX	性别	Char	$200	SEX	CRF page 13
31	RACE	种族	Char	$200	RACE	Assigned
32	ETHNIC	族群	Char	$200	ETHNIC	CRF page 13
33	ARMCD	计划分组编码	Char	$200	ARMCD	Assigned
34	ARM	计划分组描述	Char	$200	ARM	Assigned
35	ACTARMCD	实际分组编码	Char	$200	ACTARMCD	Assigned
36	ACTARM	实际分组描述	Char	$200	ACTARM	Assigned
37	COUNTRY	国家	Char	$200	ISO 3166	Assigned

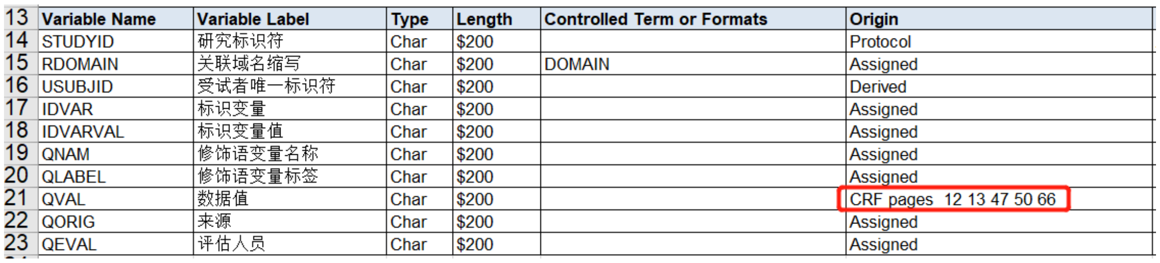
Fig.7 Automatically fill CRF pages with VBA

5. AUTOMATICALLY GENERATE VARIABLE QVAL PAGES

For supp domain, QVAL is often collected from multiple pages of CRF, so we need to fill in all page numbers of the SUPP variables into the origin column of the QVAL. As shown in the figure below, we need to fill in the page numbers of these SUPP variables. This is actually a more complex work. We refer to the following VBA code,hoping to give you inspiration specifically. The main function is to assign CRF pages to the SUPP domain variable - QVAL.



38	SITE	研究中心名称	Char	$200		CRF Page 12		HP.SITE	SUPP
39	SUBJNAME	受试者姓名缩写	Char	$200		CRF page 12		HP.SUBJNAME	SUPP
40	ETHNICO	其他民族	Char	$200		CRF page 13		DM.ETHNICO	SUPP
41	HEIGHT	身高(cm)	Char	$200		CRF page 13		DM.HEIGHT	SUPP
42	IEYN	受试者是否符合入排标准	Char	$200		CRF page 47		DS_SCREEN.IEYN	SUPP
43	SCRRES	筛选结果	Char	$200		CRF page 47		DS_SCREEN.SCRRES	SUPP
44	SCRFLDES	筛选失败原因描述	Char	$200		CRF page 47		DS_SCREEN.SCRFLDES	SUPP
45	RDYN	是否进行随机操作	Char	$200		CRF page 50		DS_SCREEN.RDYN	SUPP
46	RDREASND	未随机入组原因	Char	$200		CRF page 50		DS_SCREEN.RDREASND	SUPP
47	RANDID	随机序号	Char	$200		CRF page 50		DS_SCREEN.RANDID	SUPP
48	TESTTYPE	试验分组	Char	$200		CRF page 50		DS_SCREEN.TESTTYPE	SUPP
49	EXLSDTC	末次使用试验药物日期	Char	$200		CRF page 66		DS_SUMMARY.EXLSDAT	SUPP
50	BESTEVAL	经确认后的最佳客观疗效(参照RECIST1.1)	Char	$200		CRF page 66		DS_SUMMARY.BESTEVAL	SUPP



13	Variable Name	Variable Label	Type	Length	Controlled Term or Formats	Origin
14	STUDYID	研究标识符	Char	$200		Protocol
15	RDOMAIN	关联域名缩写	Char	$200	DOMAIN	Assigned
16	USUBJID	受试者唯一标识符	Char	$200		Derived
17	IDVAR	标识变量	Char	$200		Assigned
18	IDVARVAL	标识变量值	Char	$200		Assigned
19	QNAM	修饰语变量名称	Char	$200		Assigned
20	QLABEL	修饰语变量标签	Char	$200		Assigned
21	QVAL	数据值	Char	$200		CRF pages 12 13 47 50 66
22	QORIG	来源	Char	$200		Assigned
23	QEVAL	评估人员	Char	$200		Assigned

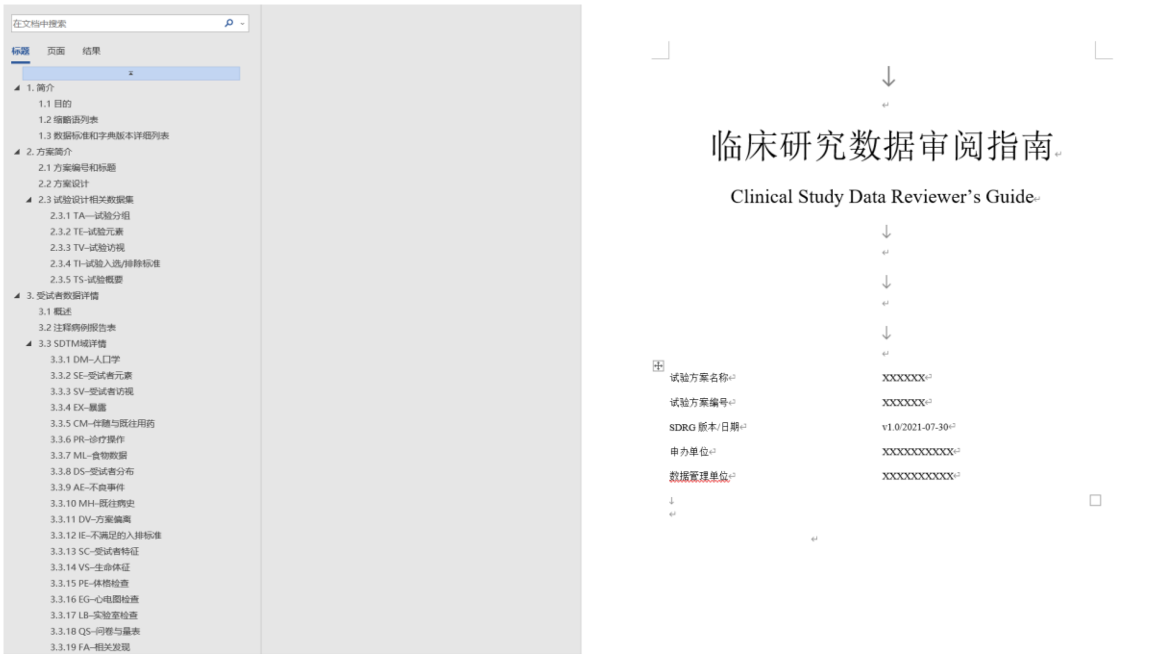
Following is the Refference VBA code

```
'Determine whether the table exists
Function WorksheetExists(WorksheetName As String, Optional wb As Workbook) As Boolean
    If wb Is Nothing Then Set wb = ThisWorkbook
    With wb
        On Error Resume Next
        WorksheetExists = (.Sheets(WorksheetName).Name = WorksheetName)
        On Error GoTo 0
    End With
End Function

Function RemoveDupesColl(myArray As Variant) As Variant
'DESCRIPTION: Use the collection method to remove duplicates from the array.
'NOTES: Returns the only element in the array, but converts the array element to a string.
    Dim i As Long
    Dim arrColl As New Collection
    Dim arrDummy() As Variant
    Dim arrDummy1() As Variant
    Dim item As Variant
    ReDim arrDummy1(LBound(myArray) To UBound(myArray))
    For i = LBound(myArray) To UBound(myArray) 'convert to string
        arrDummy1(i) = CStr(myArray(i))
    Next i
    On Error Resume Next
    For Each item In arrDummy1
       arrColl.Add item, item
    Next item
    Err.Clear
    ReDim arrDummy(LBound(myArray) To arrColl.Count + LBound(myArray) - 1)
    i = LBound(myArray)
    For Each item In arrColl
       arrDummy(i) = item
       i = i + 1
    Next item
    RemoveDupesColl = arrDummy
End Function


Function SortArrayAtoZ(myArray As Variant)

Dim i As Long
Dim j As Long
Dim Temp

'Sort the Array A-Z
For i = LBound(myArray) To UBound(myArray) - 1
    For j = i + 1 To UBound(myArray)
        If UCase(myArray(i)) > UCase(myArray(j)) Then
            Temp = myArray(j)
            myArray(j) = myArray(i)
            myArray(i) = Temp
        End If
    Next j
Next i

SortArrayAtoZ = myArray

End Function


Function SortArrayZtoA(myArray As Variant)

Dim i As Long
Dim j As Long
Dim Temp

'Sort the Array Z-A
For i = LBound(myArray) To UBound(myArray) - 1
    For j = i + 1 To UBound(myArray)
        If UCase(myArray(i)) < UCase(myArray(j)) Then
            Temp = myArray(j)
            myArray(j) = myArray(i)
            myArray(i) = Temp
        End If
    Next j
Next i

SortArrayZtoA = myArray

End Function



Sub QVAL_CRF_pages()

Dim row As Integer
Dim x As Integer
```

```vba
Dim k As Integer
Dim re As Object
Dim allMatches As Object

Set re = CreateObject("VBScript.RegExp")
    re.Pattern = "[a-zA-Z]+"
    re.IgnoreCase = True
    re.Global = True

'row = 7

'Do While ThisWorkbook.Sheets("CONTENT").Cells(row, 1) <> ""

    For i = 1 To Sheets.Count

        If Len(ThisWorkbook.Sheets(i).Name) = 2 Then

            x = 14
            Dim col As New Collection

            Do While ThisWorkbook.Sheets(i).Cells(x, 6) <> ""

                If ThisWorkbook.Sheets(i).Cells(x, 9) = "SUPP" Then

                    Set Value = ThisWorkbook.Sheets(i).Cells(x, 6)

                    Set allMatches = re.Execute(Value)

                    If allMatches.Count > 0 Then
                        Page = re.Replace(Value, "")
                    Else
                        Page = "(Not matched)"
                    End If

                    col.Add Page

                    Debug.Print Page

                    'MsgBox (Page)

                End If

            x = x + 1
            Loop

            'MsgBox ThisWorkbook.Sheets(i).Name & " SUPP Variable Collection contains " & CStr(col.Count) & " items"

            'for Collection with at leat one element
            If col.Count > 0 Then

                'MsgBox "The first item is:  " & col.item(1)

                'SUPP page list
                page_string = ""
                For Each num In col
                    page_string = page_string & num
                Next

                'String to array by space
                arr1 = Split(page_string, " ")
                'length of array
                Length = UBound(arr1) - LBound(arr1) + 1
                'MsgBox ("Length:" & Length)

                'for array with 0 element
                If Length = 0 Then
                    If WorksheetExists("SUPP" & ThisWorkbook.Sheets(i).Name) Then
                        Sheets("SUPP" & ThisWorkbook.Sheets(i).Name).Select
                        ThisWorkbook.Sheets("SUPP" & ThisWorkbook.Sheets(i).Name).Range("F21") = ""
                    Else
                        'MsgBox "SUPP" & ThisWorkbook.Sheets(i).Name & " Sheet does not exist"
                    End If
                End If

                'for array with at leat one element
                If Length >= 1 Then

                    'MsgBox (ThisWorkbook.Sheets(i).Name & " domain SUPP Variable CRF Pages Source:" & page_string)

                    Dim arr2() As Variant

                    'duplicate
                    arr2 = RemoveDupesColl(arr1)

                    'sort
                    arr3 = SortArrayAtoZ(arr2)

                    'count of element
                    count_ele = UBound(arr3) - LBound(arr3)

                    'MsgBox ("SUPP" & ThisWorkbook.Sheets(i).Name & " has " & count_ele & " unique page")

                    If count_ele = 1 Then
                        page_unique_list = "CRF page"
                        For Each Page In arr3
                            'MsgBox ("Page:" & Page)
                            page_unique_list = page_unique_list & " " & Page
                        Next
                    End If

                    If count_ele >= 2 Then
                        page_unique_list = "CRF pages"
                        For Each Page In arr3
                            'MsgBox ("Page:" & Page)
                            page_unique_list = page_unique_list & " " & Page
                        Next
                    End If

                    If WorksheetExists("SUPP" & ThisWorkbook.Sheets(i).Name) Then
                        Sheets("SUPP" & ThisWorkbook.Sheets(i).Name).Select
                        ThisWorkbook.Sheets("SUPP" & ThisWorkbook.Sheets(i).Name).Range("F21") = page_unique_list
                        'MsgBox (page_unique_list)
                    Else
                        'MsgBox "SUPP" & ThisWorkbook.Sheets(i).Name & " Sheet does not exist"
                    End If

                End If

            End If ' If col.Count > 0 Then

            Set col = Nothing

        End If  'If Len(ThisWorkbook.Sheets(i).Name) = 2 Then

    Next  'For i = 1 To Sheets.Count

'row = row + 1
'Loop

End Sub
```

6. AUTOMATICALLY GENERATE SDRG/ADRG DOCUMENT

It is believed that most pharmaceutical companies or CROs will have different ways to generate define.xml. This section will be ignored here. One of the various methods is to use Pinnacle 21 to generate the define.xml. When the sdtm/adam spec is created, we can generate a standard define spec excel file based on the SDTM spec and datasets for generating define.xml. Similarly, we can use SAS programming to generate an excel file with different sheets which support sdrg or adrg writing, such as table of issue summary. Then we use the python program to import the excel sheet and automatically write the table into the sdrg document. For some descriptive text in sdrg, we can write it in a program so that we can write both text and tables to the word document with Python. This way is applicable for ADRG similarly. For those who like programming, it's a pleasure to write text in an edit editor such as Pycharm. It saves a lot of time since some of the tables in the article are inserted automatically through programs. We can get some inspiration that it should be done with the program as far as possible, especially for repetitive work. It will improve the motivation for work and make us more like writing documents.



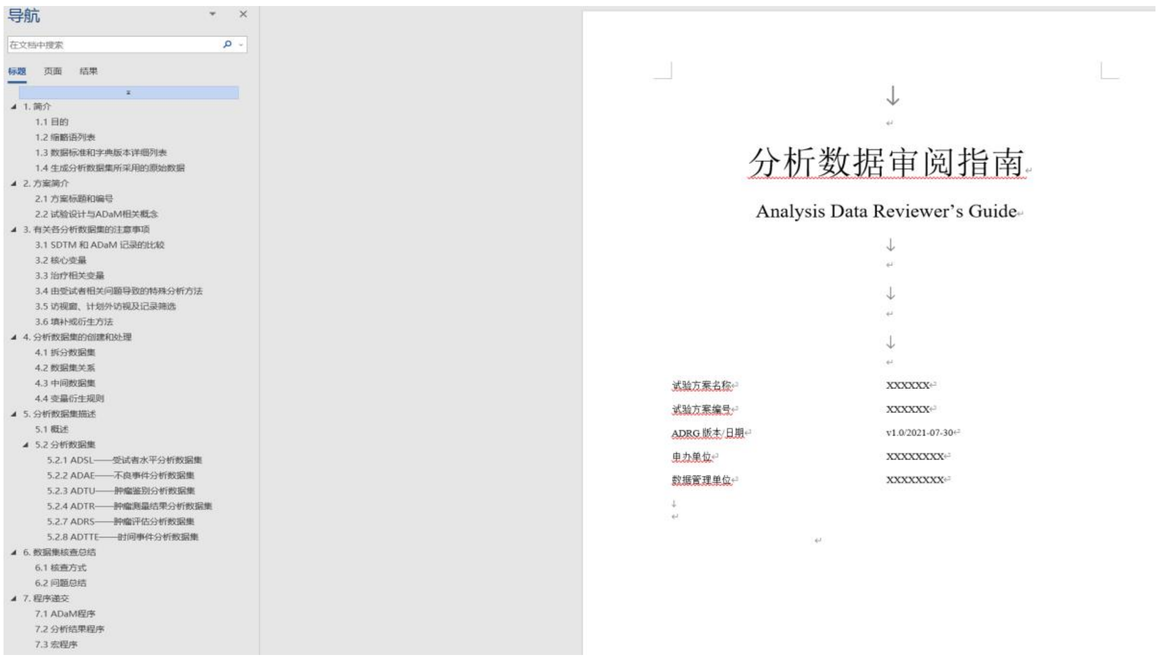
1. 简介
1.1 目的
1.2 缩略语列表
1.3 数据标准和字典版本详细列表
2. 方案简介
2.1 方案编号和标题
2.2 方案设计
2.3 试验设计相关数据集
2.3.1 TA—试验分组
2.3.2 TE–试验元素
2.3.3 TV–试验访视
2.3.4 TI–试验入选/排除标准
2.3.5 TS-试验概要
3. 受试者数据详情
3.1 概述
3.2 注释病例报告表
3.3 SDTM域详情
3.3.1 DM–人口学
3.3.2 SE–受试者元素
3.3.3 SV–受试者访视
3.3.4 EX–暴露
3.3.5 CM–伴随与既往用药
3.3.6 PR–诊疗操作
3.3.7 ML–食物数据
3.3.8 DS–受试者分布
3.3.9 AE–不良事件
3.3.10 MH–既往病史
3.3.11 DV–方案偏离
3.3.12 IE–不满足的入排标准
3.3.13 SC–受试者特征
3.3.14 VS–生命体征
3.3.15 PE–体格检查
3.3.16 EG–心电图检查
3.3.17 LB–实验室检查
3.3.18 QS–问卷与量表
3.3.19 FA–相关发现

临床研究数据审阅指南

Clinical Study Data Reviewer's Guide

试验方案名称	XXXXXX
试验方案编号	XXXXXX
SDRG 版本/日期	v1.0/2021-07-30
申办单位	XXXXXXXXXX
数据管理单位	XXXXXXXXXX

Fig.8 Automatically generated SDRG Document



1. 简介
1.1 目的
1.2 缩略语列表
1.3 数据标准和字典版本详细列表
1.4 生成分析数据集所采用的原始数据
2. 方案简介
2.1 方案标题和编号
2.2 试验设计与ADaM相关概念
3. 有关各分析数据集的注意事项
3.1 SDTM 和 ADaM 记录的比较
3.2 核心变量
3.3 治疗相关变量
3.4 由受试者相关问题导致的特殊分析方法
3.5 访视窗、计划外访视及记录筛选
3.6 填补或衍生方法
4. 分析数据集的创建和处理
4.1 拆分数据集
4.2 数据集关系
4.3 中间数据集
4.4 变量衍生规则
5. 分析数据集描述
5.1 概述
5.2 分析数据集
5.2.1 ADSL——受试者水平分析数据集
5.2.2 ADAE——不良事件分析数据集
5.2.3 ADTU——肿瘤鉴别分析数据集
5.2.4 ADTR——肿瘤测量结果分析数据集
5.2.7 ADRS——肿瘤评估分析数据集
5.2.8 ADTTE——时间事件分析数据集
6. 数据集核查总结
6.1 核查方式
6.2 问题总结
7. 程序递交
7.1 ADaM程序
7.2 分析结果程序
7.3 宏程序

分析数据审阅指南

Analysis Data Reviewer's Guide

试验方案名称	XXXXXX
试验方案编号	XXXXXX
ADRG 版本/日期	v1.0/2021-07-30
申办单位	XXXXXXXX
数据管理单位	XXXXXXXX

Fig.9 Automatically generated ADRG Document

CONCLUSION

It is complex to process clinical data in practice. SAS programmers not only do programming based on clinical data, but also annotate blank CRF, write some specification files, sdrg, adrg and other documents. In the case that SAS programmers are in short supply, it’s such a heavy task to complete these programming and document writing work. Therefore, we need to develop some programs or tools that can improve productivity. This not only saves labor costs, but also makes more SAS programmers like the job. I believe that study more technology will bring you joy and achievement during work. For junior SAS programmers, it will bring them great encouragement by learning a few macro programs, using regular expressions. It will also stimulate their own enthusiasm to learn. it's a very bad situation if you get bored with simple, repetitive work. This paper aims to develop some tools to improve work efficiency and enthusiasm. The author not only uses SAS, but also uses Python and VBA to do some automated work, I believe this article will give you inspiration that, as long as we are willing to learn more technology, the work is still full of infinite possibilities.

ACKNOWLEDGMENTS

I would like to acknowledge my previous line manager Yadong Miao and my line manager Fu Wang for their encouragement.

CONTACT INFORMATION

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

Name: Alan Zhou

Enterprise: PAREXEL International

Address: 9F & Unit A/B/C 10F, No.506, Shangcheng Road, Pudong District, Shanghai, China, 200120

City, State ZIP: Shanghai, 200120

Work Phone: +86 2120505341

E-mail: Alan.Zhou@parexel.com

Web: http://www.parexel.com/