PharmaSUG China 2023 - Paper AA-160

Using VBA for Clinical Data Collection and Analysis in Small-scale Studies

Weiping Zhang, Guangan Cao, BeiGene (Beijing) Co., Ltd.

ABSTRACT

Clinical Data Analysis Tool (CDAT) is a tool based on Visual Basic for Applications (VBA) that integrates data collection, data derivation and analysis for small-scale studies which sponsored by physician. CDAT can provide professional support to study investigator and automatize the data analysis. It aims to help standardize the statistical analysis in these studies, enhancing the consistency and reliability of data analysis.

CDAT has a simple EDC system created by EXCEL that has standardized page names and field names. After data has been entered into corresponding fields, the embedded VBA macro can create analysis results automatically included different study endpoints and summary tables, depending on the indication. For example, overall response based on RECIST 1.1, PFS and OS will be derived in a solid tumor study. It is a platform for clinical data collection and analysis. It can also do data edit checks and highlight the potential data issue.

VBA is a programming language developed by Microsoft. It enables users to automate tasks and extend the functionality of Office applications by writing macros. It is built in Microsoft Excel and can use Excel as UI directly. Benefited from the advantage of VBA, CDAT can be built within a small Excel file which is very familiar to physicians. They can collect study data in Excel as they usually do but get the analysis results at the same time.

This paper will demonstrate how CDAT uses VBA to simplify and optimize the clinical data analysis for small-scale studies. We hope that this paper will provide some insights and inspirations for other researchers who are interested in using VBA for clinical data analysis.

In this paper, Microsoft Excel for Microsoft 365 MSO (16.0.14326.20936) is being used.

INTRODUCTION

Small-scale Studies are a type of research that involves a small sample size and a limited scope which sponsored by physician. These studies often lack specialized databases and data analysts, so automating the data collection and statistical analysis can save time and increase accuracy.

When it comes to clinical data collection and analysis, you may think of SAS. But many physicians are probably not very good at using SAS. For clinical data that is not subject to drug submission, we can take other approaches. As a programming language built in Microsoft Excel, VBA has its own unique advantages. VBA can automate nearly every operation that we can perform with a mouse, keyboard, or a dialog box in Excel. It allows users to create and run small programs in Excel, which is a familiar and user-friendly platform for most people.

We have developed a tool based on VBA called CDAT (Clinical Data Analysis Tool) for physicians. This tool standardizes and simplifies the data collection and statistical analysis for small-scale studies. It has a simple EDC system and can create analysis results automatically included different study endpoints and summary table. It can also do data edit checks and flag the potential data issues. Therefore, physicians don't need to manually do these tasks. CDAT can help standardize the data collection and automate data analysis, saving time and enhancing the consistency and reliability of clinical data and analysis results.

This paper will illustrate how VBA can be applied in pharmaceutical development using CDAT as an example.

CDAT TOOL

The CDAT tool consists of five components: a simple EDC system, an overall response calculator, a summary table generator, a statistical analysis system, and a data edit check system. The following sections will explain the roles and characteristics of these components.

PART 1 USER INTERFACE DESIGN

The CDAT tool has a main interface that allows users to enter the basic information of the study and select the indication (Figure 1). The tool automatically adjusts the sheets and macro codes based on the selected indication, as different indications have different designs.

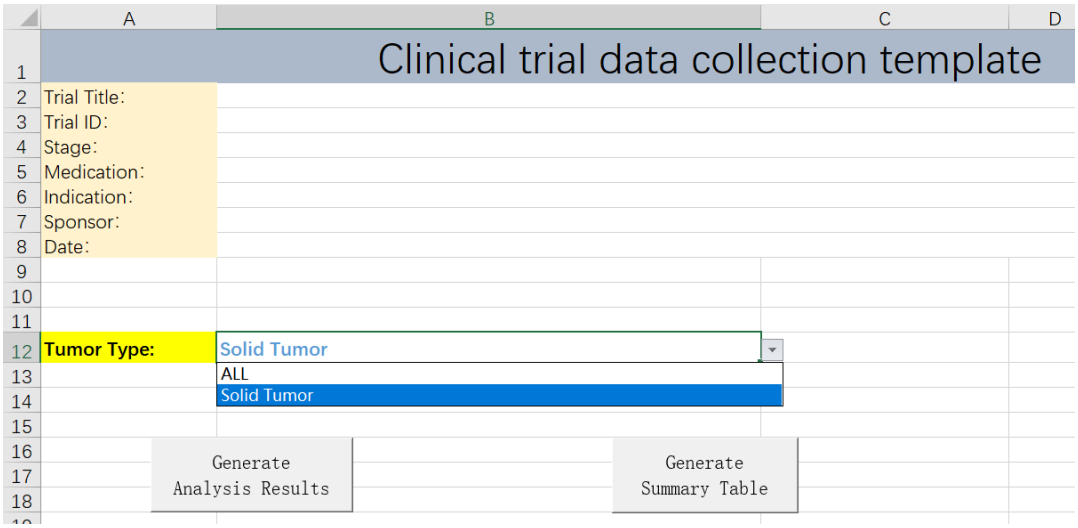

Figure 1. Main Interface

Figure 2 and Figure 3 show an example code for changing the sheets and macro codes by selecting the indication. Figure 3 also shows how to limit the usage date of the tool.

```
Private Sub Worksheet_Change(ByVal Target As Range)

    Application.ScreenUpdating = False
    If Target.Column <> 2 Or Target.Row <> 13 Then
        Exit Sub
    End If
    If Target.Cells.Count > 1 Then
        Exit Sub
    End If

    Dim rb As Workbook, ws As Worksheet, num As Integer, name As String, sht As Worksheet
    Set rb = ThisWorkbook
    rb.Worksheets("Tool").Select
    If Target.Value = "Solid Tumor" Then

        If rb.Sheets("Tool").UsedRange.Rows.Count > 22 Then rb.Sheets("Tool").Rows("23:" & rb.Sheets("Tool").UsedRange.Rows.Count).ClearContents
        rb.Sheets("Logic").Range("a24:c46").Copy
        rb.Sheets("Tool").Select
        Range("A23").Select
        ActiveSheet.Paste
        Range("A18").Select

        For Each sht In ThisWorkbook.Worksheets
            name = sht.name
            If name = "Tool" Or name = "Demography" Or name = "Treatment cycle" Or name = "Overall response" Or name = "Overall response logic" Or _
            name = "Post-treat Anti-cancer Therapy" Or name = "End of treatment" Or name = "Survival Follow-up" Or name = "Adverse event" Then
                sht.Visible = True
                Else: sht.Visible = False
            End If
        Next
    End If

End Sub
```

Figure 2. Code of Changing the Sheets

```
Sub ana_all()

    Dim datee As Date
    datee = #4/11/2024#
    If Date > datee Then
        MsgBox "The current version of this tool has expired. Please contact us to obtain the latest version! "
        Exit Sub
    End If
    If Sheets("Tool").Range("B13").Value = "Solid Tumor" Then Call onco_calcu
    If Sheets("Tool").Range("B13").Value = "ALL" Then Call all_calcu

End Sub
```

Figure 3. Code of Changing the Codes and Limiting the Usage Date

PART 2 EDC SYSTEM

CDAT has a simple EDC system created by EXCEL which standardized page names and field names. Users can also customize the study-specify variables. The EDC system includes the most common pages such as demography (Figure 4), treatment cycle (Figure 5), overall response (Figure 6), survival follow-up (Figure 7), and adverse event (Figure 8), etc.

	A	B	C	D	E	F	G	H	I	J
1	Suject ID	Sex(Cat)	Age(Con): <=65;>65;<=75;>75	ECOG Score(Cat)	Smoke(Cat)	Diease Stage(Cat)	Safety Analysis Set	Intent Analysis Set	Subgroups(Cat)	BMI(Con): <18.5;18.5 - <=21.8;21.8 - <23.9;>=23.9
2	PT1	Male	56	1	Ever	II	Yes	Yes	EGFR mutation	24.45
3	PT3	Female	48	0	Never	IIIA	Yes	Yes	EGFR TKI	18.5
4	PT2	Male	67	2	Current	IVB	Yes	Yes	anti-angiogenesis	21.8
5	PT4	Female	73	1	Ever	IVA	Yes	Yes	EGFR TKI	23.67
6	086-001	Male	56	1	Ever	II	Yes	Yes	anti-angiogenesis	23.45
7	086-002	Female	48	0	Never	IIIA	Yes	Yes	anti-angiogenesis	18.9
8	086-003	Male	67	2	Current	IVB	Yes	Yes	EGFR TKI	20.21
9	086-004	Female	73	1	Ever	IVA	Yes	Yes	EGFR mutation	21.67
10	086-005	Male	66	2	Current	0is	Yes	Yes	EGFR TKI	22.3
11	086-006	Female	59	1	Never	0a	No	No	EGFR mutation	18.23
12	086-007	Female	68	1	Current	UN	Yes	Yes	EGFR mutation	17.98

Figure 4. Demography Page

	A	B	C
1	Suject ID	Treatment Cycle	Cycle Start Date
2	086-001	1	2020/6/18
3	086-002	1	2020/12/3
4	086-003	1	2021/4/1
5	086-004	1	2021/4/1
6	086-005	4	2020/4/29
7	086-006	5	2020/5/26
8	086-007	6	2020/6/20
9	PT1	1	2020/3/1
10	PT1	2	2020/3/29
11	PT1	4	2020/4/5

Figure 5. Treatment Cycle

	A	B	C	D	E	F
1	Suject ID	Tumor Assessment Date	Target Lesion	Non-Target Lesion	New Lesion	Overall Response
2	PT1	2020/4/26	PR	CR	No	PR
3	PT1	2020/6/18	NA	Non-CR/Non-PD	No	非CR/非PD
4	PT2	2020/5/28	CR	Non-CR/Non-PD	No	PR
5	PT2	2020/7/18	PR	CR	No	PR
6	PT2	2020/10/10	SD	Non-CR/Non-PD	No	SD

Figure 6. Overall Response Page

	A	B	C
1	Suject ID	Post-treat Anti-cancer Therapy No.	Post-treat Anti-cancer Therapy Start Date
2	PT1	1	2020/6/23
3	PT1	2	2020/8/1
4	PT1	3	2020/9/5
5	PT2	1	2020/9/20
6	PT2	2	2020/9/30
7	PT2	3	2020/10/30
8	PT4	1	2020/12/27

Figure 7. Survival Follow-up Page

	A	B	C	D
1	Suject ID	Adverse Event	CTCAE Grade: Any; <3; >=3	Study Drug Related?
2	086-002	Alopecia	1	Yes
3	PT2	Rash	2	Yes
4	PT3	Anemia	1	Yes
5	PT4	Hyperuricemia	2	Yes
6	086-001	Nausea	2	Yes
7	086-002	Anemia	1	Yes
8	086-003	Rash	1	Yes
9	PT3	Anemia	2	Yes
10	086-003	Rash	1	Yes
11	086-007	Vomiting	2	Yes

Figure 8. Adverse Event Page

PART 3 OVERALL RESPONSE CALCULATOR

In the overall response page, users can enter subject ID, tumor assessment date, target lesion response, non-target lesion response and new lesion status. The CDAT tool will then calculate the overall response automatically. Users can also input overall response manually. If the user-entered overall response differs from the tool-calculated one, the corresponding cell will turn yellow (Figure 9). Figure 10 gives an example code for calculating the overall response automatically.

	A	B	C	D	E	F
1	Suject ID	Tumor Assessment Date	Target Lesion	Non-Target Lesion	New Lesion	Overall Response
2	PT1	2020/4/26	PR	CR	No	PD
3	PT1	2020/6/18	NA	Non-CR/Non-PD	No	Non-CR/Non-PD
4	PT2	2020/5/28	CR	Non-CR/Non-PD	No	PR
5	PT2	2020/7/18	PR	CR	No	PR
6	PT2	2020/10/10	SD	Non-CR/Non-PD	No	SD

Figure 9. Flag the Overall Response that may be Incorrect

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Application.EnableEvents = False
    Application.ScreenUpdating = False
    If (Target.Column = 1 Or Target.Column = 2 Or Target.Column = 3 Or Target.Column = 4 Or Target.Column = 5 Or _
    Target.Column = 6 Or Target.Column = 7 Or Target.Column = 8) And Target.Row > 1 Then
        Dim i%, j%, cell_fl As Object
        Set cell_fl = CreateObject("scripting.dictionary")
        Set RangeOfInterest = Range("A:H")
        Set Intersection = Intersect(Target, RangeOfInterest)
        For Each cell In Intersection
            i = cell.Row
            j = cell.Column
            If Not cell_fl.exists(i) Then Call cal_res_all(i, j)
            cell_fl(i) = True
        Next
    End If
    Application.ScreenUpdating = True
    Application.EnableEvents = True
End Sub
```

Figure 10. Code of Computing Overall Response

PART 4 STATISTICAL ANALYSIS SYSTEM

Users can generate all the analysis results automatically by clicking the “Generate analysis results” command button after entering the data into corresponding fields (Figure 11). The tool will show a message box if an endpoint is not computed in any subject (Figure 12).

	A	B	C	D	E	K	L	M	N	O	P	Q
1	Subject ID	Treatment Start Date	Best of Reponse(Cat)	Best of Reponse Date	First PD Date	PFS （day)	PFS censor	OS(day)	OS censor	DOR(day)	DOT(day)	TTR(day)
2	PT1	2020/3/1	PR	2020/4/26		110	Yes	760	Yes	54	663	57
3	PT3	2019/12/3	CR	2020/4/28	2020/4/29	149	No	149	Yes	103	21	47
4	PT2	2020/4/29	PR	2020/5/28		135	Yes	155	No	106	358	30
5	PT4	2019/12/3	CR	2020/12/26		390	Yes	734	Yes	1	21	390
6	086-001	2020/6/18	PR	2020/8/18		62	Yes	62	Yes	1	21	62
7	086-002	2020/12/3	SD	2021/12/3	2021/12/3	366	No	366	Yes		21	
8	086-003	2021/4/1	PD	2021/12/8	2021/12/8	252	No	253	Yes		21	
9	086-004	2021/4/1	PD	2021/6/1	2021/6/1	62	No	154	Yes		21	
10	086-005	2020/4/29	SD	2020/4/30		2	Yes	124	Yes		21	
11	086-006	2020/5/26	CR	2020/9/26	2020/9/26	124	No	128	Yes		21	
12	086-007	2020/6/20				1	Yes	1	Yes		21	

Figure 11. Statistical Analysis Results

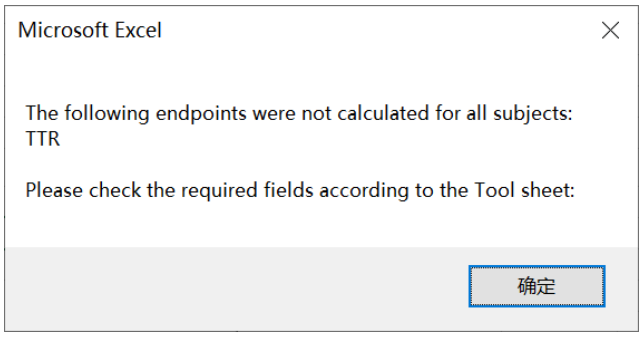

Figure 12. Message Box

PART 5 SUMMARY TABLE GENERATOR

The CDAT can also create different summary tables for the variables to be analyzed based on their variable types defined by users. The ‘Summary Table’ tab will display all the tables when users click the “Generate Summary Table” command button.

To get summary tables for continuous variables, users just need to add a variable type identifier 'Con' in the variable header (Figure 13). The tool will generate all the summary tables for continuous variables automatically (Figure 14). Figure 15 gives an example code.

	A	B	C
1	Suject ID	Age(Con):	BMI(Con):
2	PT1	56	24.45
3	PT3	48	18.5
4	PT2	67	21.8
5	PT4	73	23.67
6	086-001	56	23.45
7	086-002	48	18.9
8	086-003	67	20.21
9	086-004	73	21.67
10	086-005	66	22.3
11	086-006	59	18.23
12	086-007	68	17.98

Figure 13. Variable Head of Continuous Variables

	A	B	C	D	E	F
1	Results:					
2						
3	Variable (continuous)	Mean	Standard Deviation	Medium	Quartile(Q1,Q3)	Range(Min-Max)
4	Age	62.20	9.47	66.5	56, 67.75	48-73
5	BMI	21.29	2.29	21.735	19.23, 23.16	17.98-24.45

Figure 14. Results of Continuous Variables

```
If InStr(sht.Cells(1, i).Value, "Con") > 0 Or InStr(sht.Cells(1, i).Value, "Total of Treatment Cycles") > 0 Then
    num_add = 0
    varname = Split(Replace(sht.Cells(1, i).Value, " (", "("), "(")(0)
    calcuyn = False

    ReDim arr(num - 1)
    For h = 2 To t_row
        If IsNumeric(sht.Cells(h, 1).Value) = True Then patient = Str(sht.Cells(h, 1).Value)
        If IsNumeric(sht.Cells(h, 1).Value) = False Then patient = Trim(sht.Cells(h, 1).Value)
        If dic_eff(patient) = "Yes" And sht.Cells(h, i).Value <> "" Then
           num_add = num_add + 1
           calcuyn = True
           arr(num_add - 1) = sht.Cells(h, i).Value
        End If
    Next h

  If calcuyn = True Then
    lastrow = rb.Sheets("Summary Table").Range("a65536").End(xlUp).Row
    rb.Sheets("Summary Table").Range("a" & lastrow + 1).Value = varname      'Derive the variable name

    rb.Sheets("Summary Table").Range("b" & lastrow + 1).Value = Round(Application.WorksheetFunction.Average(arr), 2)     'Mean
    rb.Sheets("Summary Table").Range("c" & lastrow + 1).Value = Round(Application.WorksheetFunction.StDev(arr), 2)       'Std
    rb.Sheets("Summary Table").Range("d" & lastrow + 1).Value = Application.WorksheetFunction.Median(arr)               'Medium
    rb.Sheets("Summary Table").Range("e" & lastrow + 1).Value = Str(Round(Application.WorksheetFunction.Quartile(arr, 1), 2)) _
    & ", " & Str(Round(Application.WorksheetFunction.Quartile(arr, 3), 2))                'Quartile
    rb.Sheets("Summary Table").Range("f" & lastrow + 1).Value = Trim(Str(Application.WorksheetFunction.Min(arr))) & "-" _
    & Trim(Str(Application.WorksheetFunction.Max(arr)))                                   'Range
  End If

End If
```

Figure 15. Code of Continuous Variables

To get summary tables for categorical variables, users just need to add a variable type identifier 'Cat' in the variable header (Figure 16). The tool will generate all the summary tables for categorical variables automatically (Figure 17). Figure 18 gives an example code.

	A	B	D	E	I
1	Suject ID	Sex(Cat)	ECOG Score(Cat)	Smoke(Cat)	Subgroups(Cat)
2	PT1	Male	1	Ever	EGFR mutation
3	PT3	Female	0	Never	EGFR TKI
4	PT2	Male	2	Current	anti-angiogenesis
5	PT4	Female	1	Ever	EGFR TKI
6	086-001	Male	1	Ever	anti-angiogenesis
7	086-002	Female	0	Never	anti-angiogenesis
8	086-003	Male	2	Current	EGFR TKI
9	086-004	Female	1	Ever	EGFR mutation
10	086-005	Male	2	Current	EGFR TKI
11	086-006	Female	1	Never	EGFR mutation
12	086-007	Female	1	Current	EGFR mutation

Figure 16. Variable Head of Categorical Variables

	I	J	K	L
1	Variable (categorical)	Category	Frequency (n)	Percent (%)
2	Sex	Male	5	50
3		Female	5	50
4	ECOG Score	1	5	50
5		0	2	20
6		2	3	30
7	Smoke	Ever	4	40
8		Never	2	20
9		Current	4	40
10	Subgroups	EGFR mutation	3	30
11		EGFR TKI	4	40
12		anti-angiogenesis	3	30

Figure 17. Results of Categorical Variables

```
    For h = 0 To num_add - 1
        If IsNumeric(arr(h)) = True Then keyvalue = Str(arr(h))
        If IsNumeric(arr(h)) = False Then keyvalue = arr(h)

        If keyvalue <> "" Then dic(keyvalue) = dic(keyvalue) + 1

    Next h

    keynum = dic.Count

  If keynum <> 0 Then

    lastrow = rb.Sheets("Summary Table").Range("j65536").End(xlUp).Row
    rb.Sheets("Summary Table").Range("i" & lastrow + 1).Value = varname

    rb.Sheets("Summary Table").Columns("J:J").NumberFormatLocal = "@"

    rb.Sheets("Summary Table").Cells(lastrow + 1, "J").Resize(keynum, 1) = Application.WorksheetFunction.Transpose(dic.Keys)

    For h = lastrow + 1 To keynum + lastrow
        pre_value = rb.Sheets("Summary Table").Range("j" & h).Value
        rb.Sheets("Summary Table").Range("k" & h).Value = dic(pre_value)
        If exist_label = True Then rb.Sheets("Summary Table").Range("j" & h).Value = dic_label(pre_value)
        rb.Sheets("Summary Table").Range("L" & h).Value = Round(Val(rb.Sheets("Summary Table").Range("k" & h).Value) * 100 / deno, 2)
    Next h

   End If

End If
```

Figure 18. Code of Categorical Variables

The CDAT also designs and provides some additional identifier formats for users, making the tool more flexible and powerful in supporting various summary tables. Some of these identifier formats are as follows:

To do frequency table on continuous variables using custom classification criteria, users only need to specify the classification criteria in the variable header (Figure 19). The tool will generate the frequency table automatically (Figure 20).

	A	B	C
1	Suject ID	Age(Con): <=65;>65;<=75;>75	BMI(Con): <18.5;18.5 - <=21.8;21.8 - <23.9;>=23.9
2	PT1	56	24.45
3	PT3	48	18.5
4	PT2	67	21.8
5	PT4	73	23.67
6	086-001	56	23.45
7	086-002	48	18.9
8	086-003	67	20.21
9	086-004	73	21.67
10	086-005	66	22.3
11	086-006	59	18.23
12	086-007	68	17.98

Figure 19. Variable Head of Continuous Variables with Classification Criteria

	I	J	K	L
1	Variable (categorical)	Category	Frequency (n)	Percent (%)
2	Age	<=65	4	40
3		>65	6	60
4		<=75	10	100
5		>75	0	0
6	BMI	<18.5	1	10
7		18.5 - <=21.8	5	50
8		21.8 - <23.9	3	30
9		>=23.9	1	10

Figure 20. Frequency Table of Continuous Variables with Classification Criteria

To generate cross-tabulation frequency tables on two categorical variables, users only need to specify the two variables in the “Crosstab” column as worksheet name: column index * worksheet name: column index (Figure 21). The tool will generate the cross-tabulation tables automatically (Figure 22). If the categorical variable is in the current sheet, the worksheet name can be omitted.

	A	B	E	I	K
1	Suject ID	Sex(Cat)	Smoke(Cat)	Subgroups(Cat)	Crosstab
2	PT1	Male	Ever	EGFR mutation	B*E
3	PT3	Female	Never	EGFR TKI	I*Preliminary analysis result:C
4	PT2	Male	Current	anti-angiogenesis	
5	PT4	Female	Ever	EGFR TKI	
6	086-001	Male	Ever	anti-angiogenesis	
7	086-002	Female	Never	anti-angiogenesis	
8	086-003	Male	Current	EGFR TKI	
9	086-004	Female	Ever	EGFR mutation	
10	086-005	Male	Current	EGFR TKI	
11	086-006	Female	Never	EGFR mutation	
12	086-007	Female	Current	EGFR mutation	

Figure 21. Specify the Two Variables in the Crosstab Column

	U	V	W	X	Y
1	Crosstab Results:				
2					
3	R * C				
4	Sex \ Smoke	Ever	Never	Current	
5		4	2	4	
6	Male	2 (50%)	0	3 (75%)	
7	Female	2 (50%)	2 (100%)	1 (25%)	
8					
9	Subgroups \ 最好疗效	PR	CR	SD	PD
10		3	2	3	2
11	EGFR mutation	1 (33.3%)	0	1 (33.3%)	1 (50%)
12	EGFR TKI	0	2 (100%)	1 (33.3%)	1 (50%)
13	anti-angiogenesis	2 (66.7%)	0	1 (33.3%)	0

Figure 22. Cross-Tabulation Results

To get the summary results of adverse event incidence rate, users just need to enter the adverse events in the “Adverse event” worksheet. The tool will generate the summary table for adverse events automatically (Figure 23).

	O	P	Q	R
1	Safety Results:			
2				
3	Characteristics	Any grade	< Grade 3	≥ Grade 3
4		9 (90%)	9 (90%)	3 (30%)
5	Alopecia	2 (20%)	1 (10%)	1 (10%)
6	Rash	2 (20%)	2 (20%)	0
7	Anemia	4 (40%)	4 (40%)	0
8	Hyperuricemia	3 (30%)	1 (10%)	2 (20%)
9	Nausea	1 (10%)	1 (10%)	0
10	Vomiting	2 (20%)	2 (20%)	0

Figure 23. Adverse Event Summary Table

PART 6 DATA EDIT CHECK SYSTEM

Users can also perform the edit check automatically by clicking the “Check” command button. The tool will highlight the cells with potential issues in yellow and add issue messages in comments (Figure 24 Figure 25 shows an example code.

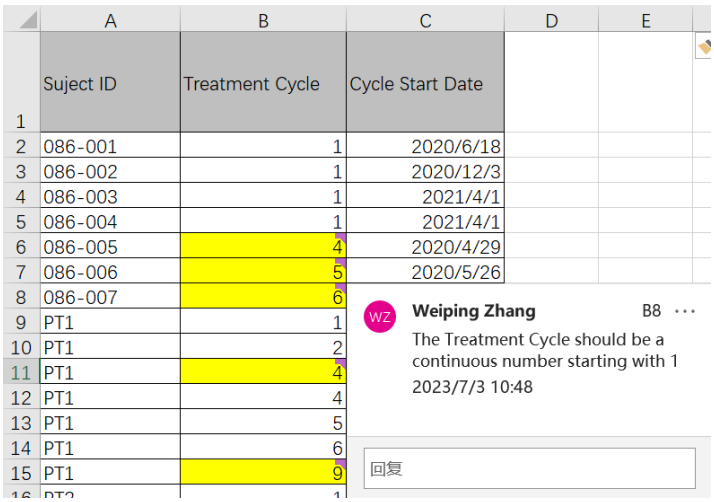

Figure 24. Edit Check Results

```
If Range(ltc_cy_c & ii) <> cycle Then
    Range(ltc_cy_c & ii).Interior.Color = 65535
    Range(ltc_cy_c & ii).AddCommentThreaded ("The Treatment Cycle should be a continuous number starting with 1")
End If
```

Figure 25. Code of Changing Cells Color and Add Comments

CONCLUSION

In conclusion, CDAT is a tool that leverages VBA to provide a simple and efficient way of collecting and analyzing clinical data for small-scale studies. It can help physicians to conduct studies with standardized and reliable statistical methods, and to obtain analysis results automatically. CDAT demonstrates the potential and usefulness of VBA for clinical data analysis, and we hope that it can inspire more researchers to explore this programming language.

REFERENCES

Microsoft. 2022. "Getting started with VBA in Office"

https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office

CONTACT INFORMATION

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

Weiping Zhang
BeiGene (Beijing) Co.,Ltd.
weiping.zhang@beigene.com

Guangan Cao
BeiGene (Beijing) Co.,Ltd.
guangan.cao@beigene.com