PharmaSUG 2019 – Paper AP-116

Excel with MS Excel and X Commands:
SAS® Programmers' Quick Tips Guide to Useful Advanced MS Excel Functions and X Commands, for Efficient Project Management

Shefalica Chand, Seattle Genetics, Inc., Bothell, WA

ABSTRACT

SAS® is the bread and butter for clinical/statistical programmers in the pharmaceutical and biotech industry. Nevertheless, there are many functionalities, tools, and resources that can be used in conjunction with SAS® to facilitate a better user experience for biostatistics and other cross-functional partners.

In this paper, we will share some useful advanced MS Excel functions like VLOOKUP, HYPERLINK, CONCATENATION, etc., to improve efficiency, remove ambiguity and redundancies, and enhance accuracy by eliminating manual intervention (for example, creating specification, tracking sheet, managing review comments, and tracking resolutions and implementations).

Here is a sneak-peek into these Excel functions:

VLOOKUP: This function helps search values based on a particular column (index column) from one Excel sheet to another by matching common values in the index columns.

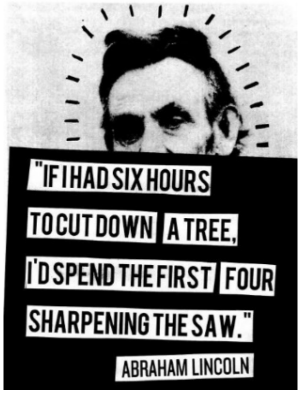
HYPERLINK: This function can help create hyperlinks to external files.

CONCATENATE: This function combines values from multiple cells and/or text strings. Can be separated by comma or range of Excel cells can be selected.

We will also discuss the **SAS® X Commands** that can help make some of the manual processes more accurate and simpler, for example, using X Commands to **copy files** from one location to another, automatically **create folders/directories** at a given location, quickly **convert SAS® program files to TXT** files as a bulk action, and **remove files or folders**, and a few other similar helpful actions.

In addition, we will explore practical implementation of these tools and resources in our industry, environment, and day-to-day activities.

INTRODUCTION



Hard work and smart work are two sides of the same coin, thus they need to go hand in hand. It goes without saying that we need to first work hard to create a strong foundational base, and with smart work we can do the same amount of work faster and more efficiently. There is a thin line between working relentlessly hard and enabling ourselves to learn smart working techniques.

The core job of clinical/statistical SAS® programmers comprises coding and project management activities (status tracking, timelines tracking, project updates, work allocation, TLF outputs review, comments adjudication, etc.). Some or most of these project management activities could be manual and time consuming.

This paper will highlight some functionalities and techniques of Advanced MS Excel and SAS® X commands/statements that may help improve efficiencies and save a tremendous amount of time.

USEFUL MICROSOFT EXCEL TIPS FOR SAS® PROGRAMMERS

At the first look, Microsoft Excel seems like a simple grid of columns and rows. It is also a popular and convenient program used for calculations, making charts and recording/organizing data.

Here are a few simple MS Excel features to help enhance efficiencies, and overall user experience, for clinical/statistical SAS® programmers and the projects that they handle.

CONCATENATE Function

This function combines values from multiple cells and/or text strings. Can be separated by commas or a range of Excel cells can be selected. The Excel CONCATENATE function can join up to 30 items and returns the result as text.

Syntax: **=CONCATENATE (text1, text2, [text3], ...)**

The text items can be text strings, numbers, or cell references that refer to one cell. Alternatively, the ampersand character (&) can give the same result.

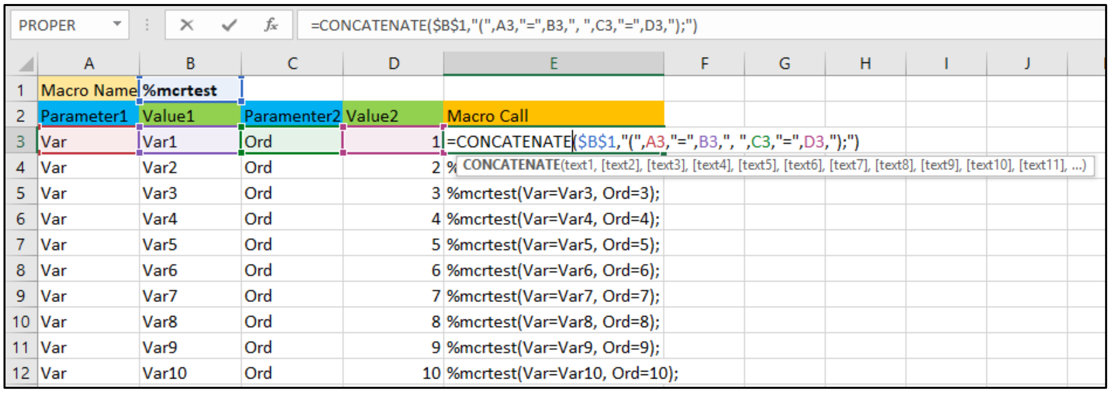
Practical Usage: Writing multiple macro calls, to avoid typing each of them separately.

Example:

The following simple example shows how we can easily create a long list of SAS macro calls quickly, instead of having to manually type them one by one. In the illustrated example, cell values and text connectors like ( ) = ; were used in the appropriate order to write a complete macro call (see Figure 1).

The only effort needed is to write the first formula correctly. To copy the formula down the length, the mouse can either be held and dragged all the way down, or the tiny box at the bottom right-hand corner of the cell can just be double clicked.

The next step is to just copy the set of cells with the formula and paste it directly into the SAS® program.



=CONCATENATE(B1,"(",A3,"=",B3,", ",C3,"=",D3,");")

	A	B	C	D	E	F	G	H	I	J
1	Macro Name	%mcrtest								
2	Parameter1	Value1	Paramenter2	Value2	Macro Call					
3	Var	Var1	Ord	1	=CONCATENATE(B1,"(",A3,"=",B3,", ",C3,"=",D3,");")					
4	Var	Var2	Ord	2	%					
5	Var	Var3	Ord	3	%mcrtest(Var=Var3, Ord=3);					
6	Var	Var4	Ord	4	%mcrtest(Var=Var4, Ord=4);					
7	Var	Var5	Ord	5	%mcrtest(Var=Var5, Ord=5);					
8	Var	Var6	Ord	6	%mcrtest(Var=Var6, Ord=6);					
9	Var	Var7	Ord	7	%mcrtest(Var=Var7, Ord=7);					
10	Var	Var8	Ord	8	%mcrtest(Var=Var8, Ord=8);					
11	Var	Var9	Ord	9	%mcrtest(Var=Var9, Ord=9);					
12	Var	Var10	Ord	10	%mcrtest(Var=Var10, Ord=10);					

CONCATENATE(text1, [text2], [text3], [text4], [text5], [text6], [text7], [text8], [text9], [text10], [text11], ...)

Figure 1. CONCATENATE function to write macro call

HYPERLINK Function

The HYPERLINK function creates quick hyperlinks to any location within the current workbook, or opens a document stored at a particular location. The location could be a folder on a network server or an Internet site.

Syntax: **=HYPERLINK (link_location, [friendly_name])**

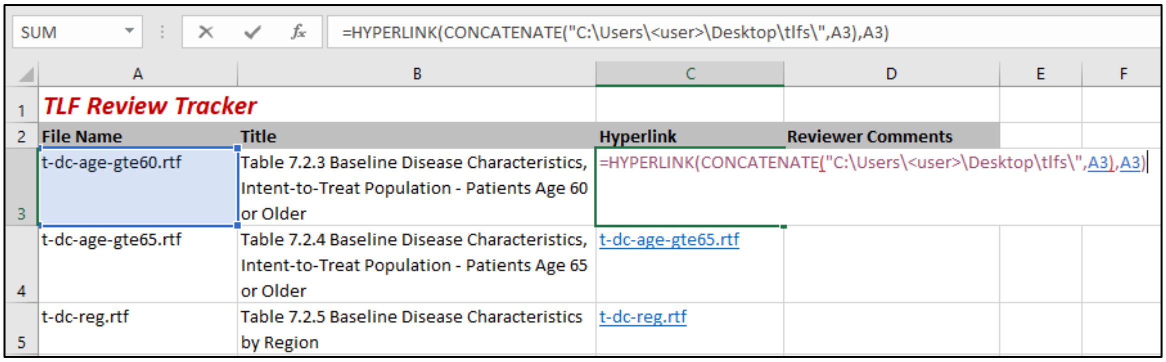
link_location (required): Can be anything, from a path to a file, web address (URL), text string enclosed in quotation marks, workbook cell, bookmark or file name of the document to be hyperlinked.

friendly_name (optional): The value that is displayed in the cell. By default, friendly_name is displayed in blue and is underlined. It can be a value, a text string, a name, or a cell that contains a value. If friendly_name is not specified, the cell displays the link_location.

Practical Usage: Creating hyperlinks within a tracker spreadsheet containing list of outputs, for easy referencing of TLF outputs.

Example 1:

The hyperlink in row 3 of Figure 2 will access the table **t-dc-age-gte60.rtf** from the location **C:\Users\<user>\Desktop\tlfs**.



SUM

```
=HYPERLINK(CONCATENATE("C:\Users\<user>\Desktop\tlfs\",A3),A3)
```

	A	B	C	D	E	F
1	*TLF Review Tracker*					
2	File Name	Title	Hyperlink	Reviewer Comments		
3	t-dc-age-gte60.rtf	Table 7.2.3 Baseline Disease Characteristics, Intent-to-Treat Population - Patients Age 60 or Older	=HYPERLINK(CONCATENATE("C:\Users\<user>\Desktop\tlfs\",A3),A3)			
4	t-dc-age-gte65.rtf	Table 7.2.4 Baseline Disease Characteristics, Intent-to-Treat Population - Patients Age 65 or Older	t-dc-age-gte65.rtf			
5	t-dc-reg.rtf	Table 7.2.5 Baseline Disease Characteristics by Region	t-dc-reg.rtf			

Figure 2. HYPERLINK function to access TLF outputs

The links will work even if the tracker file is emailed or saved at a different location, as long as the user of the tracker has access to the path mentioned in the **link_location**, where the TLF outputs are located.

The hyperlink will break if the TLF output is moved to a different location, other than that specified in the **link_location**.

Example 2 (More Dynamic Approach):

If the output and the tracker Excel spreadsheet are located at the same location, for example, in a single folder on the network drive, then the **link_location** and **friendly_name** can have the same value, to be entered within double quotes. There is no need to specify the complete path of the location.

=HYPERLINK("t-dc-age-gte60.rtf","t-dc-age-gte60.rtf")

The hyperlinks will work even if the whole folder is moved to another location. However, the links will break if any of the files (output or Excel tracker) are emailed or saved at a different location.

VLOOKUP Function

Performs a vertical lookup by searching for a value in the first column of a table and returning the value in the same row in the index number position.

Syntax: **=VLOOKUP (lookup_value, table_array, col_index_num, [range_lookup])**

lookup_value: The value or range you want to look up.

table_array: The range containing the lookup value.

col_index_num: The number of the column in the range that contains the return value. For example, if you specify C:F as the range and want to return the values from column F, you should count C as the first column and col_index_num will be 4.

range_lookup (optional): Specify TRUE for approximate match or FALSE for exact match of the return value. If nothing is specified, the default value is TRUE.

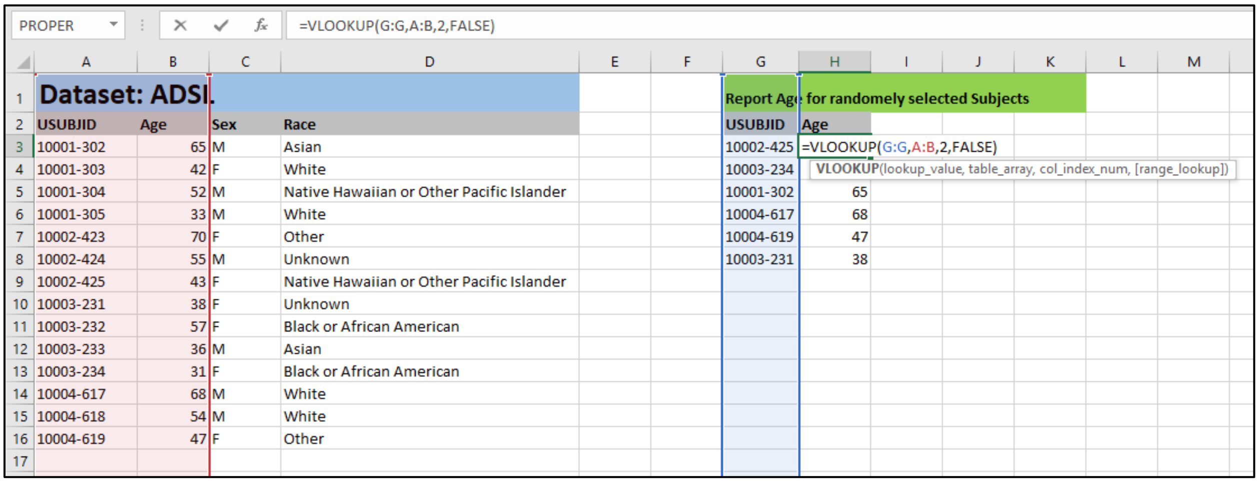
Practical Usage: It works quite similar to the **Merge** statement in SAS. This is useful if we have two Excel spreadsheets and we would like to look up a range of values in another range of values and return the desired return value.

Example:

The simple example below shows how to report "Age" for randomly selected subjects.

Here we see that we wish to look up the value of subjects in range G:G, from the A:B range, and return the value from B, which is the 2nd column from A.

The true usage of this function is seen with large amounts of data, where this function can help save a tremendous amount of time and effort.



=VLOOKUP(G:G,A:B,2,FALSE)

	A	B	C	D	E	F	G	H
1	Dataset: ADSL						Report Age for randomely selected Subjects	
2	USUBJID	Age	Sex	Race			USUBJID	Age
3	10001-302	65	M	Asian			10002-425	=VLOOKUP(G:G,A:B,2,FALSE)
4	10001-303	42	F	White			10003-234	
5	10001-304	52	M	Native Hawaiian or Other Pacific Islander			10001-302	65
6	10001-305	33	M	White			10004-617	68
7	10002-423	70	F	Other			10004-619	47
8	10002-424	55	M	Unknown			10003-231	38
9	10002-425	43	F	Native Hawaiian or Other Pacific Islander				
10	10003-231	38	F	Unknown				
11	10003-232	57	F	Black or African American				
12	10003-233	36	M	Asian				
13	10003-234	31	F	Black or African American				
14	10004-617	68	M	White				
15	10004-618	54	M	White				
16	10004-619	47	F	Other				
17								

VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])

Figure 3: VLOOKUP to look up values

Another practical use of this function is to merge reviewers' comments residing in multiple files to one consolidated file, for ease of management and maintenance.

SHORTCUTS

Before we talk about shortcuts, it is important to emphasize a general good practice. Reducing our reliability on the mouse and using the keyboard to navigate within and across various files and applications may result in an exponential increase in efficiency (due to reduced switching time).

Some general navigation tricks will be shared at the end of this section.

MS Excel Quick Tips

Here are some of the functions and shortcuts to help manage Excel files like a pro.

- **Move from one worksheet tab to the next worksheet tab**: Ctrl + PgDn and Ctrl + PgUp
- **Quickly move to the top or end of a data range**: Ctrl + Arrow
- **Using Shift key to select data**: Ctrl + Shift +Arrow keys will extend the current selection to the last nonblank cell in that direction
- **Double click to copy down**: To copy a formula or value down the length of your data set, no need to hold and drag the mouse all the way down, just double click the tiny box at the bottom right-hand corner of the cell
- **Copy content from the cell above**: Ctrl + D
- Table 1 presents some **Excel functions** that work **similarly to their SAS® counterparts**

Desired Action	Function
Return specified number of characters from the start of a text string	=LEFT (text string, number of characters)
Return specified number of characters from the end of a text string	=RIGHT (text string, number of characters)
Return specified number of characters from the "start number" position	=MID (text string, start number, number of characters)
Remove leading and trailing spaces	=TRIM (text string)
Return the length of the specified text string	=LEN (text string)
Convert text to all upper-case	=UPPER (text string)
Convert text to all lower-case	=LOWER (text string)
Convert text to proper case (Capitalize the first letter in each word)	=PROPER (text string)

Table 1. Useful Excel text functions

- **Removing formula from cell**: Ctrl + C to copy content, press the keys in the order Alt, E, S, V
- **Transposing a range of cells**: Ctrl + C to copy content, then go to the cell position where transpose needs to be applied and press the keys in the order Alt, E, S, E
- **Removing duplicate values**:

  Select the data range, then go to Data → Remove Duplicates (see Figure 3).

Shortcut: Select the data range, then press the keys in the order Alt + A + M

Values in the data range do not need to be sorted to use this functionality.

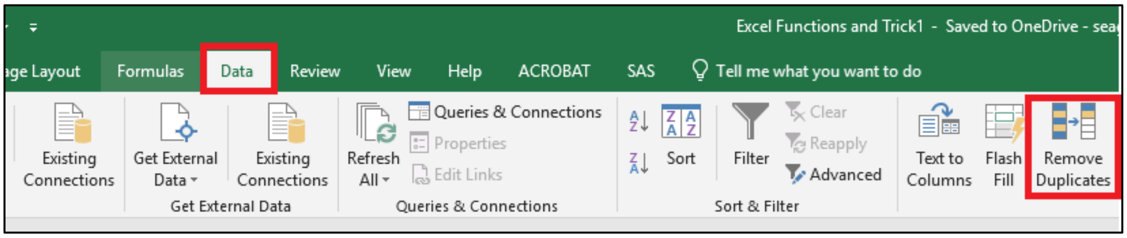


Figure 3. Removing duplicate values

General MS Windows Navigation Keyboard Shortcuts

Here are some of the general navigation shortcuts for MS Windows that may improve productivity.

- Switch between open windows: Alt + Tab (or Alt + Shift + Tab to go back)
- Close an open window: Alt + F4
- Get back the accidentally closed browser tab: Ctrl + Shift + T
- Show/hide Desktop: Win + D
- Open the start Menu: Ctrl + Esc
- Open the hidden Start menu (Quick Link): Win + X

With a little practice and familiarity, our fingers will be fluttering across the keyboard like a pro and we will see noticeable efficiency gains.

SAS® X COMMANDS/STATEMENTS

Windows or MS-DOS commands can be executed within SAS® by using the X statement or the X command. This can help reduce many manual processes, to achieve more accurate and efficient results.

The X statement and X command are quite similar in their functionalities in the SAS® windowing environment. The main difference being, the X statement is submitted like any SAS statement, whereas, the X command is issued as a windowing environment command.

Syntax:
```
X '<command>';
```

Some practical use cases:

1. **Copy files from one location to another (COPY)**: With the help of the X command/statement, we can copy files from one location to another. The files can be in any format: .sas, .txt, .docx, .xlsx, etc.

   The command below can copy all SAS® programs from an input location over to the specified output location and rename the file extensions to .txt.

   This functionality is tremendously helpful if we want to convert several SAS® programs (TLF and ADaM programs) to.txt formats.

   ```
   X 'copy <input-path>\*.sas <output-path>\*.txt';
   ```

   Doing this process manually can be quite inefficient, time consuming and error prone.

2. **Copy entire directory trees (XCOPY)**: XCOPY performs a similar function to COPY, except it can move files as well as directories from one location to another.

   XCOPY command should be used in conjunction with these options:

   /s - Copy all files from a directory and in any subdirectory within it.

   /e – Copy subdirectories, even if they are empty. This option must be used with /s.

   ```
   X 'xcopy <input-directory-path> <output-directory-path> /s/e';
   ```

3. **Automatically create folders/directories (MD or MKDIR)**: X command/statement can help efficiently create directories/folder at a desired location.
   Example: Below is another simple example to help efficiently manage manual work during the time of a regulatory submission.

   ```
   X 'md <new-directory-path>';
   ```

   The above commands can be used in the right order to automatically create a folder and copy the required files/folders to the newly created folder.

4. **Move the content from one location to another (MOVE):** Similar to COPY command, a **MOVE** command can help move files to a desired location.

```
X 'move <input-path>\*.sas <output-path>';
```

To move all the content from a folder, use *.* with input path.

5. **Remove folders/directories (RD or RMDIR)**: This X command/statement can help remove directories/folder, however, it is a high-risk task and should be used with caution.

   RD(RMDIR) command cannot delete a directory if the directory is not empty.

   Example: The command below will immediately delete the directory/folder mentioned, if it is empty.

   ```
   X 'rd <directory path to be removed/deleted>';
   ```

   This command can be used to delete an unnecessary folder.

6. **Delete one or more files (DEL or ERASE)**: This command can delete files residing at the specified locations.

   ```
   X 'del <path of file(s) to be deleted>/*.sas';
   ```

   Another risky command that should be used judiciously, as the files will be deleted immediately, without any alerts.

Carefully thought-through use of X commands/statements enables users to automate tasks and can help reduce an enormous amount of manual work to improve efficiencies and accuracy.

CONCLUSION

Quality and accuracy are paramount in our work in the drug development industry, as it directly affects patients. At the same time, speed is critical due to the strong need to get new drugs out to patients as soon as possible. Practicing smart working techniques may bring a substantial improvement in our projects, by reducing time and enhancing efficiencies.

This paper is an effort in that direction. Implementing these learnings into our day-to-day activities can generate innovative solutions to improve management of repetitive, time-consuming tasks.

ACKNOWLEDGMENTS

I would like to thank my team for their support and valuable suggestions and comments.

RESOURCES

Microsoft Office Support: support.office.com

SAS Support: support.sas.com

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the authors at:

Shefalica Chand
Seattle Genetics, Inc.
21823 - 30th Drive S.E.
Bothell, WA 98021
425-527-4866
schand@seagen.com