PharmaSUG 2024 - Paper AD- 218

Potentials and Caveats When Using ChatGPT for Enhanced SAS Macro Writing

Xinran (Flora) Luo, Weijie Yang, Everest Clinical Research

ABSTRACT

AI language like ChatGPT has impressed and even intimidated programmers. There are discussions of ChatGPT with examples of simple SAS steps and there are descriptions of various usages of ChatGPT without examples, but few papers discuss the use of ChatGPT in SAS macro development with examples.

This paper explores the utility of ChatGPT in enhancing the process of writing SAS macros from scratch, using an example of checking SAS log in batch on Windows and comparing the procedure of using conventional search engines. The focus is not only on utilizing ChatGPT's capabilities to provide programmers with initial ideas of program structure when they encounter unusual work request but also on demonstrating its application in developing a robust macro by showing key steps of the conversations between programmers and ChatGPT.

Although ChatGPT proves invaluable in offering insights and suggestions, it's imperative to acknowledge certain caveats. Not all responses provided by ChatGPT are infallible, especially in the context of technical domains like SAS programming. Emphasizing the importance of independent verification, this paper underscores the need for users, especially new learners of SAS, to scrutinize and validate the suggestions before implementation.

This paper aims to empower SAS practitioners by showcasing how ChatGPT can complement their macro-writing endeavors. By highlighting both the potentials and limitations of leveraging AI language models like ChatGPT, this paper contributes to fostering a balanced and discerning approach towards utilizing AI-driven assistance in SAS programming and macro development.

INTRODUCTION

Writing SAS macros can present several challenges, requiring careful attention to syntax, logic, and efficiency. One common challenge is ensuring that the macro generates correct code for different scenarios and handles unexpected inputs gracefully. Additionally, debugging macros can be challenging due to limited error messages and the need to understand macro processing in SAS. Furthermore, documenting macros effectively is crucial for ensuring their usability and maintainability over time. Overall, mastering SAS macro writing involves navigating these challenges while balancing flexibility, efficiency, and reliability in macro design and implementation.

ChatGPT is an advanced natural language processing model developed by OpenAI. It's capable of understanding and generating human-like text based on the input it receives. With its vast knowledge and language understanding capabilities, ChatGPT can be a valuable tool for various applications, including SAS programming. ChatGPT can assist SAS programmers by providing suggestions, explanations, and examples related to SAS syntax, procedures, and functions. In terms of SAS macro development, ChatGPT can help generate template code, suggesting macro variables, and offering ideas for macro logic. This can be particularly helpful for beginners who are learning to write macros or for experienced programmers seeking inspiration for new macro solutions.

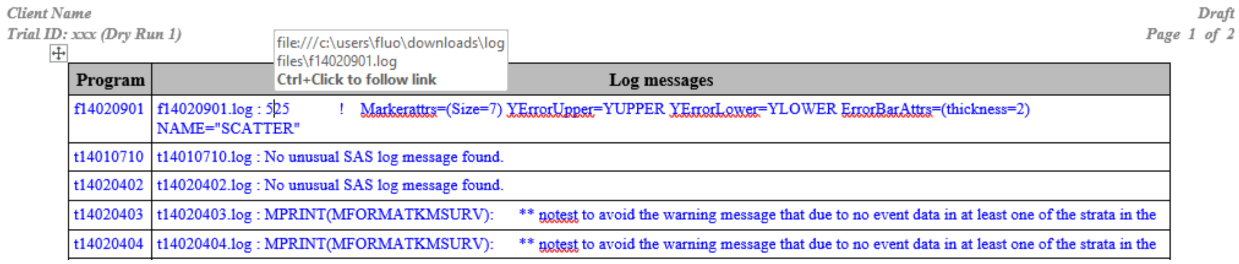
The existing papers only demonstrate how to use ChatGPT to search for simple, single questions or commands in SAS. Writing a macro involves multiple small questions, and the continuous conversational search approach of ChatGPT is well-suited for brainstorming and divergent thinking. Additionally, breaking down the versatile functionalities that the macro needs to achieve into clear and concise questions is beneficial, even without ChatGPT. This approach of breaking down the big goal into smaller objectives, clearly expressing our requirements/questions, and tackling them one by one is very useful. Again, ChatGPT's conversational approach is more inspiring than regular searches because it provides tailored responses that are custom-made to address our questions, rather than searching through SAS SUPPORT. Furthermore, the code is replicable, and there are comments explaining each step of the process. These comments generated by ChatGPT help with the maintenance of SAS macros and provide users with a better understanding.

In this paper, a specific SAS macro will be used as an example to illustrate the entire process of using ChatGPT to assist in writing the macro, and to introduce key steps to the audience. The paper will also compare conventional search engines with ChatGPT, highlighting the latter's characteristics. However, ChatGPT also has its limitations, and users need to carefully discern and test the code before formally adopting it.

TASK DESCRIPTION

One way to check code or perform quality control (QC) is to examine the log files for errors or warnings. The task is to develop a SAS macro from scratch to check SAS log in batch on Windows by searching multiple log files for specific messages or keywords of interest, such as "overwritten," "uninitialized," or "values have been converted." This eliminates the need to manually open each log file for inspection. When faced with such an unusual request and lacking direction, consulting ChatGPT becomes necessary.

Figure 1 shows the desired output of log summary. There will be 2 columns for program names and log messages respectively. Additionally, each message has a hyperlink to the corresponding log file and the program name in the other column has a hyperlink to the corresponding program, so that users can check codes in detail.



Client Name *Draft*
Trial ID: xxx (Dry Run 1) *Page 1 of 2*

file:///c:\users\fluo\downloads\log files\f14020901.log
Ctrl+Click to follow link

Program	Log messages
f14020901	f14020901.log : 525 ! Markerattrs=(Size=7) YErrorUpper=YUPPER YErrorLower=YLOWER ErrorBarAttrs=(thickness=2) NAME="SCATTER"
t14010710	t14010710.log : No unusual SAS log message found.
t14020402	t14020402.log : No unusual SAS log message found.
t14020403	t14020403.log : MPRINT(MFORMATKMSURV): ** notest to avoid the warning message that due to no event data in at least one of the strata in the
t14020404	t14020404.log : MPRINT(MFORMATKMSURV): ** notest to avoid the warning message that due to no event data in at least one of the strata in the

Figure 1. Desired Outputs by PC SAS

Our goal is to use PC SAS to extract the log messages we care about and store them in a log summary rtf file by simply inputting the path to the program and log files as parameters to the SAS macro (Figure 2).

```
%macro_name(inpath=C:\Users\fluo\Downloads\log files\,
            pgmpath=C:\Users\fluo\Downloads\programs\,
            outfile=log_summary.rtf);
```

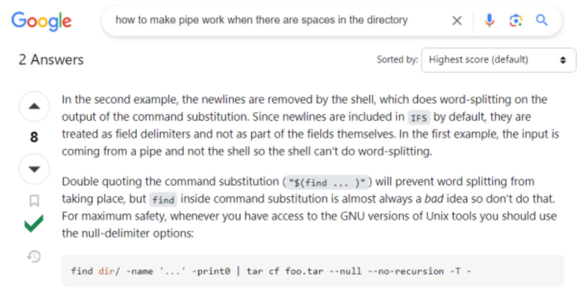
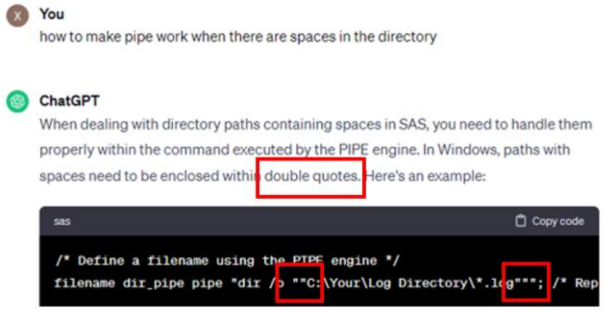
Figure 2. Desired Macro Parameters

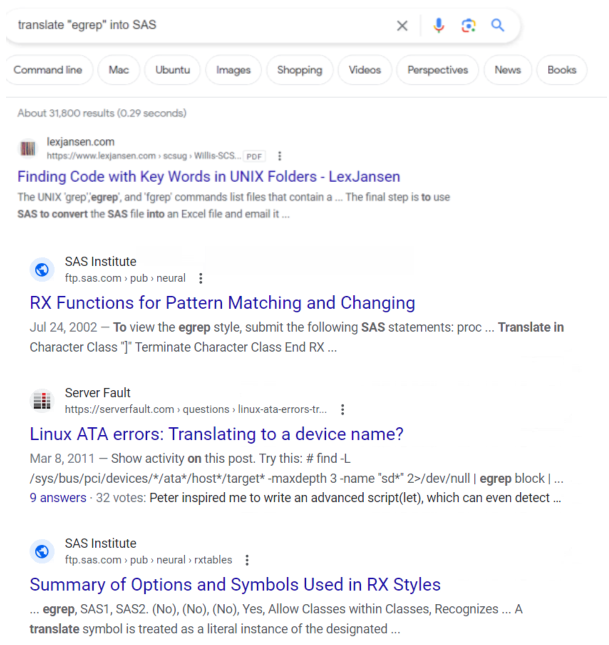
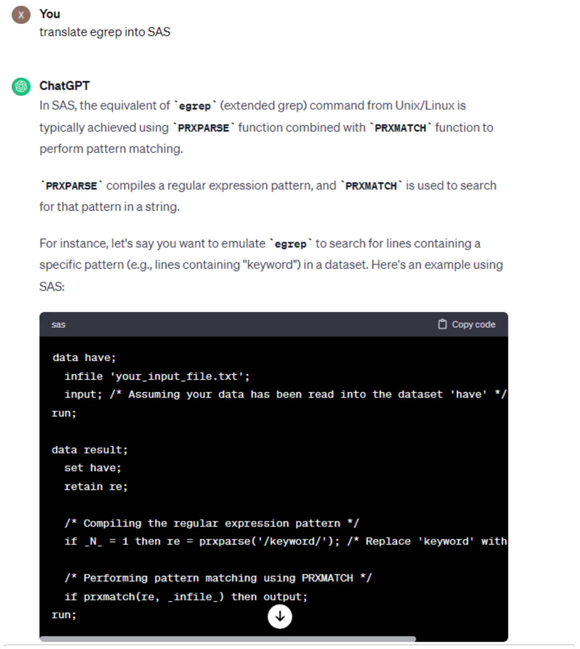
COMPARISON BETWEEN GOOGLE AND CHATGPT

When working on the SAS macro, questions were asked using Google and ChatGPT (Question 1, Question 2). ChatGPT gave the correct answer the first time it was asked. Google can also find answers, but it requires opening several web pages, reading other people's questions to confirm if they match one's own, and then searching between the lines of potentially lengthy responses because the answers from a Google search might contain more than just the solution to one specific question.

Please see the comparison of search results between Google and ChatGPT (Table 1).

Google	ChatGPT
Question 1. How to make \`pipe\` work when there are spaces in the directory?	

 Google
how to make pipe work when there are spaces in the directory
2 Answers
Sorted by: Highest score (default)
8
In the second example, the newlines are removed by the shell, which does word-splitting on the output of the command substitution. Since newlines are included in `IFS` by default, they are treated as field delimiters and not as part of the fields themselves. In the first example, the input is coming from a pipe and not the shell so the shell can't do word-splitting.
Double quoting the command substitution (`"$(find ... )"`) will prevent word splitting from taking place, but `find` inside command substitution is almost always a *bad* idea so don't do that. For maximum safety, whenever you have access to the GNU versions of Unix tools you should use the null-delimiter options:
`find dir/ -name '...' -print0 | tar cf foo.tar --null --no-recursion -T -`
	 **You**
	how to make pipe work when there are spaces in the directory
	ChatGPT
	When dealing with directory paths containing spaces in SAS, you need to handle them properly within the command executed by the PIPE engine. In Windows, paths with spaces need to be enclosed within double quotes. Here's an example:
	sas Copy code
	`/* Define a filename using the PIPE engine */`
	`filename dir_pipe pipe "dir /b ""C:\Your\Log Directory\*.log"""; /* Rep`

Question 2. How to extract keywords from log files?	
 translate "egrep" into SAS Command line Mac Ubuntu Images Shopping Videos Perspectives News Books About 31,800 results (0.29 seconds) lexjansen.com https://www.lexjansen.com › scsug › Willis-SCS... PDF Finding Code with Key Words in UNIX Folders - LexJansen The UNIX 'grep','**egrep**', and 'fgrep' commands list files that contain a ... The final step is **to** use **SAS to convert** the **SAS** file **into** an Excel file and email it ... SAS Institute ftp.sas.com › pub › neural RX Functions for Pattern Matching and Changing Jul 24, 2002 — **To** view the **egrep** style, submit the following **SAS** statements: proc ... **Translate in** Character Class "]" Terminate Character Class End RX ... Server Fault https://serverfault.com › questions › linux-ata-errors-tr... Linux ATA errors: Translating to a device name? Mar 8, 2011 — Show activity **on** this post. Try this: # find -L /sys/bus/pci/devices/*/ata*/host*/target* -maxdepth 3 -name "sd*" 2>/dev/null | **egrep** block | ... 9 answers · 32 votes: Peter inspired me to write an advanced script(let), which can even detect ... SAS Institute ftp.sas.com › pub › neural › rxtables Summary of Options and Symbols Used in RX Styles ... **egrep**, SAS1, SAS2. (No), (No), (No), Yes, Allow Classes within Classes, Recognizes ... A **translate** symbol is treated as a literal instance of the designated ...	 **You** translate egrep into SAS **ChatGPT** In SAS, the equivalent of \``egrep`\` (extended grep) command from Unix/Linux is typically achieved using \``PRXPARSE`\` function combined with \``PRXMATCH`\` function to perform pattern matching. \``PRXPARSE`\` compiles a regular expression pattern, and \``PRXMATCH`\` is used to search for that pattern in a string. For instance, let's say you want to emulate \``egrep`\` to search for lines containing a specific pattern (e.g., lines containing "keyword") in a dataset. Here's an example using SAS: sas Copy code

```
data have;
  infile 'your_input_file.txt';
  input; /* Assuming your data has been read into the dataset 'have' */
run;

data result;
  set have;
  retain re;

  /* Compiling the regular expression pattern */
  if _N_ = 1 then re = prxparse('/keyword/'); /* Replace 'keyword' with

  /* Performing pattern matching using PRXMATCH */
  if prxmatch(re, _infile_) then output;
run;
```

Table 1. Comparison between Google and ChatGPT

CONVERSATION WITH CHATGPT

In the process of conversing with ChatGPT to complete this SAS macro, several tips were summarized on how to have an effective conversation with ChatGPT. These tips will be interspersed with the steps of this SAS macro as examples.

BREAK DOWN THE TASK INTO STEPS

When you have no clear direction, you ask several core questions regarding the main functionalities that the SAS macro needs to achieve, in this example, including how to extract keywords and how to read multiple files. Once you have an overall idea, you can gradually break down the goals into smaller objectives and tackle them one by one. In this example, these are the steps you follow to develop this macro, and also the questions you ask ChatGPT (Table 2). The details or follow-up questions are shown in black on the right (Table 2), while the errors generated by the code provided by ChatGPT are highlighted in red.

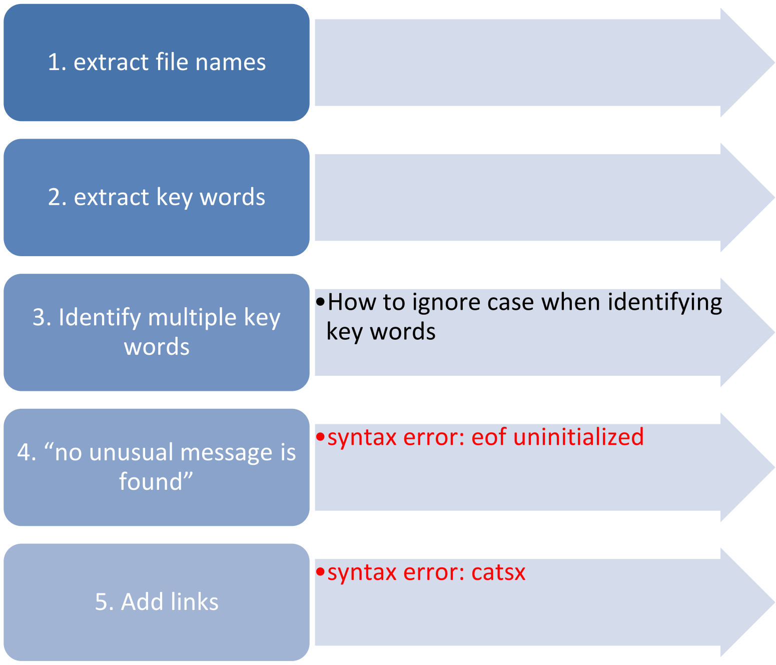
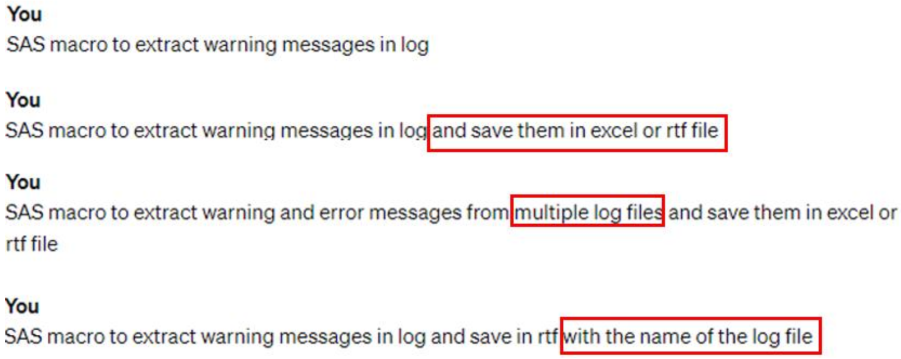

1. extract file names
2. extract key words
3. Identify multiple key words
•How to ignore case when identifying key words
4. “no unusual message is found”
•syntax error: eof uninitialized
5. Add links
•syntax error: catsx

Table 2. Break Down the Task into Steps

TRANSFER TASKS TO QUESTIONS WITH CONTRAINTS AND LIMITS

You start with asking ChatGPT a rather general question. After seeing the answers given, you start to revise and flesh out your questions by adding constraints and limitations, such as the characteristics of input data, the file type of the output, and the specific layout of the output. The below 4 sequential questions are an example. At first, ChatGPT only provided an \`index\` statement to identify a single log file. However, as the questions became more specific, the code it provided became more complex, including \`pipe\`, \`prxparse\` and ODS statements for reading multiple files, identifying multiple keywords, and displaying the output, respectively.



You
SAS macro to extract warning messages in log

You
SAS macro to extract warning messages in log and save them in excel or rtf file

You
SAS macro to extract warning and error messages from multiple log files and save them in excel or rtf file

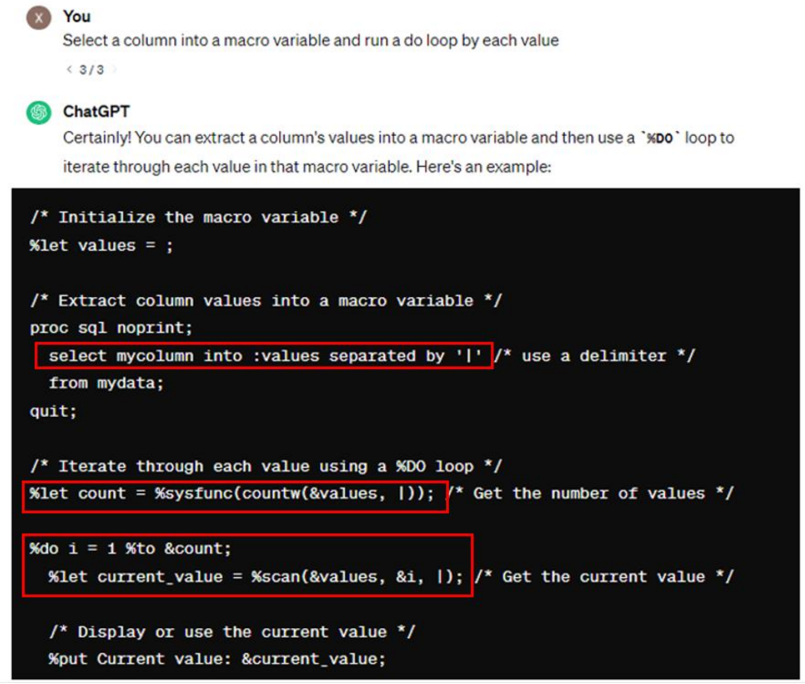
You
SAS macro to extract warning messages in log and save in rtf with the name of the log file

Figure 3. Set Limitations and Constraints

After using \`prxparse\` to extract key words, there are additional conditions that need to be included in the macro. For instance, ignore case sensitivity and identify multiple keywords instead of one. Unlike Google searches, search results from ChatGPT will link to previously asked questions so you just need to ask the single question and don’t have to explain the background again.

ASK CLEAR QUESTIONS

Sometimes, even if you have an overall idea, you may still be unsure about how to write the code specifically or where to start. It could be just a vague outline at this point. In such cases, you can seek guidance from ChatGPT. Your task is to clearly express your questions, including what you aim to achieve and the specific steps you want to implement. These steps are best described as clear, small, and manageable tasks rather than broad, general tasks. This way, ChatGPT can provide you with the answers you are looking for. In this example, after extracting multiple file names, you need to store them in a macro variable and use a do loop for each file name for subsequent steps (Figure 4).



You

Select a column into a macro variable and run a do loop by each value

ChatGPT

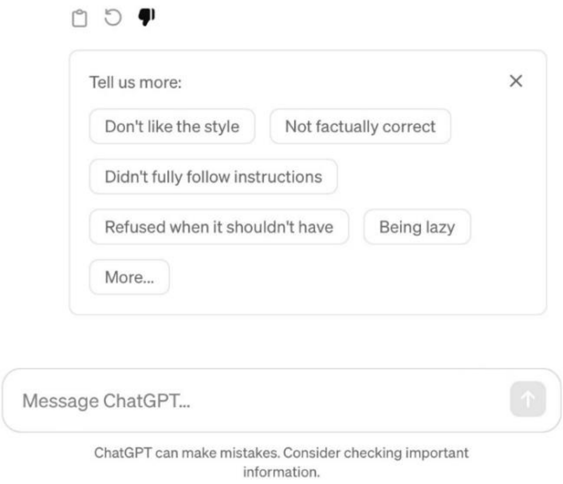
Certainly! You can extract a column's values into a macro variable and then use a `%DO` loop to iterate through each value in that macro variable. Here's an example:

```
/* Initialize the macro variable */
%let values = ;

/* Extract column values into a macro variable */
proc sql noprint;
  select mycolumn into :values separated by '|' /* use a delimiter */
  from mydata;
quit;

/* Iterate through each value using a %DO loop */
%let count = %sysfunc(countw(&values, |)); /* Get the number of values */

%do i = 1 %to &count;
  %let current_value = %scan(&values, &i, |); /* Get the current value */

  /* Display or use the current value */
  %put Current value: &current_value;
```

Figure 4. Do Loop

ALWAYS VERIFY

While ChatGPT can provide seemingly correct and well-written answers with explanatory comments, always remember to verify each answer. You can think of ChatGPT as a highly intelligent child who can respond fluently to many questions, but it can also provide inaccurate answers. In such situations, it needs feedback, such as corrections, providing the correct answer, or giving a thumbs down (Figure 5) to its response. It helps us learn and improve work efficiency, and it also needs our help to gradually refine and improve. In general, if we already know the correct answer and intentionally ask ChatGPT, we can directly correct it when ChatGPT provides an inaccurate answer. However, when developing SAS macros, the answers are exploratory and there may be multiple correct solutions. Fortunately, SAS itself is a tool that can directly validate ChatGPT code. This requires users to have a certain level of SAS knowledge, and for beginners, at least having a functional SAS environment to run the code and validate the answers provided by ChatGPT is necessary. The most direct way is to input the first error or warning generated by the ChatGPT code into the dialog box for validation. ChatGPT will then apologize and try to give another answer. During the development of this SAS macro, two ways of modifying answers were encountered: one is to switch to a completely different answer, and the other is to make corrections based on the existing answer. Please see the below examples (Figure 6, Figure 7, Figure 8, Figure 9)



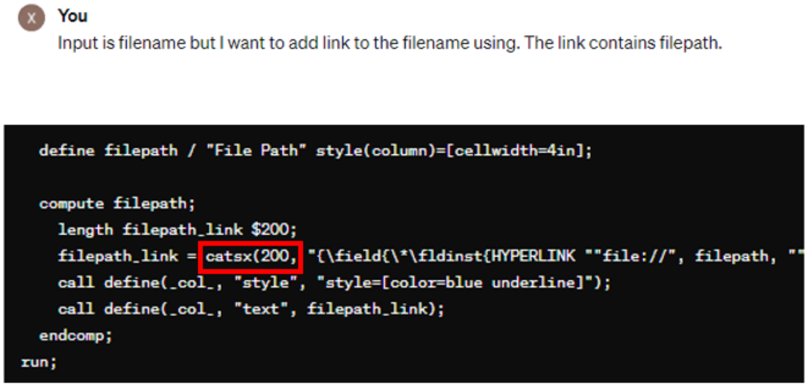
Tell us more:

ChatGPT can make mistakes. Consider checking important information.

Figure 5. Feedback for ChatGPT

Example of an Alternative Answer

In this SAS macro example, the last step was about adding hyperlinks that could connect to the SAS program and log file corresponding to each message. ChatGPT provided an answer containing a code snippet that included a somewhat unfamiliar function called `catsx` (Figure 6).



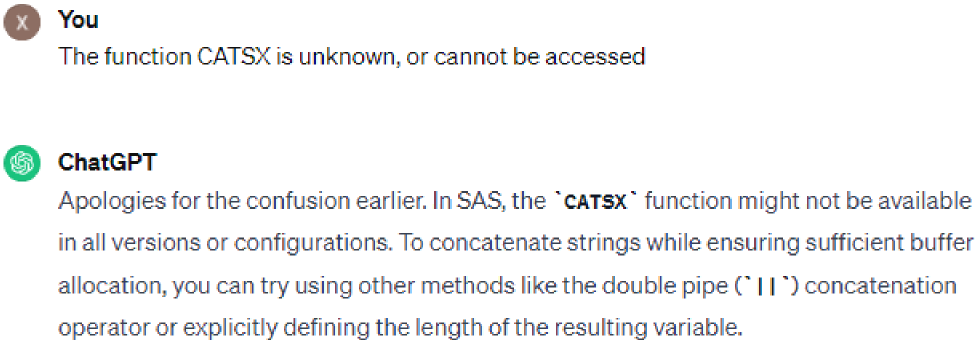
You

Input is filename but I want to add link to the filename using. The link contains filepath.

```
  define filepath / "File Path" style(column)=[cellwidth=4in];

  compute filepath;
    length filepath_link $200;
    filepath_link = catsx(200, "{\field{\*\fldinst{HYPERLINK ""file://", filepath, ""
    call define(_col_, "style", "style=[color=blue underline]");
    call define(_col_, "text", filepath_link);
  endcomp;
run;
```

Figure 6. Add Link Version 1

After copying the code into SAS and running it, errors occurred. ChatGPT provided an alternative answer of using `||` after the first error was asked (Figure 7).



You

The function CATSX is unknown, or cannot be accessed

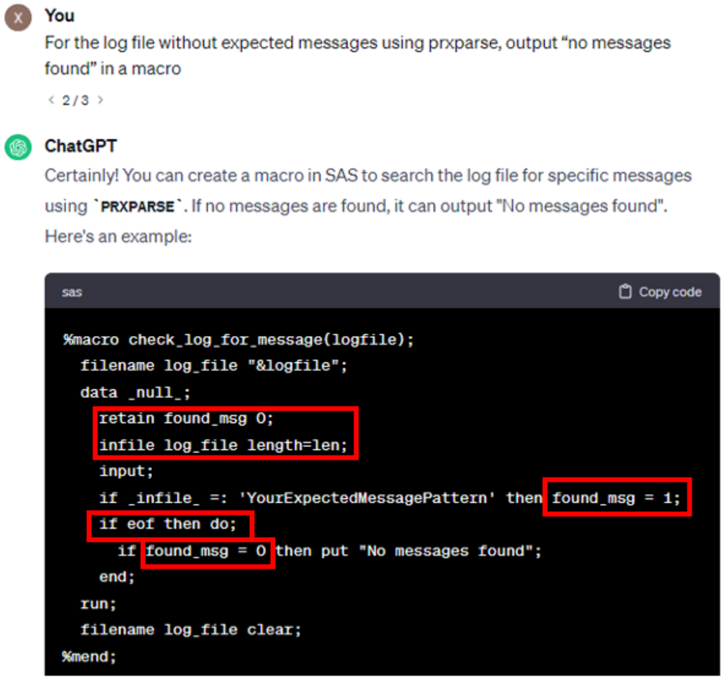
ChatGPT

Apologies for the confusion earlier. In SAS, the `CATSX` function might not be available in all versions or configurations. To concatenate strings while ensuring sufficient buffer allocation, you can try using other methods like the double pipe (`||`) concatenation operator or explicitly defining the length of the resulting variable.

Figure 7. ChatGPT's Alternative Answer

Example of a Revised Answer

In addition to identifying and displaying the messages we're monitoring, if no messages are detected, it should generate the sentence in the log summary output: "No unusual message is found." When ChatGPT was asked about this requirement, it provided what looks like the correct answer (Figure 8).



You

For the log file without expected messages using prxparse, output “no messages found” in a macro

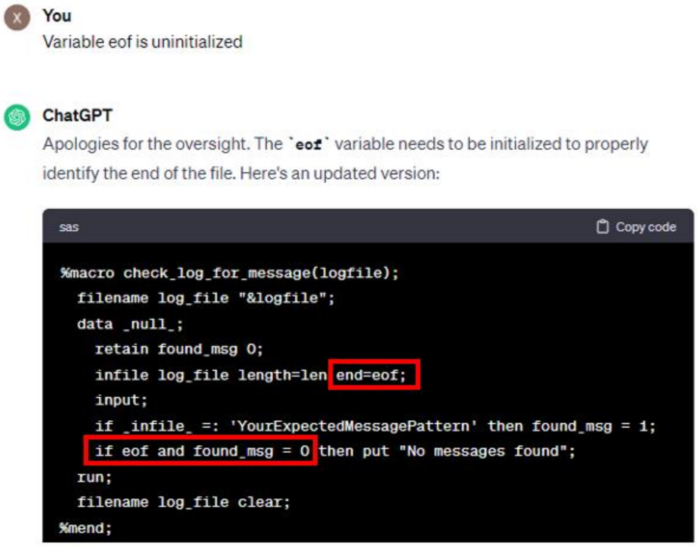
ChatGPT

Certainly! You can create a macro in SAS to search the log file for specific messages using `PRXPARSE`. If no messages are found, it can output "No messages found". Here's an example:

```
%macro check_log_for_message(logfile);
  filename log_file "&logfile";
  data _null_;
    retain found_msg 0;
    infile log_file length=len;
    input;
    if _infile_ =: 'YourExpectedMessagePattern' then found_msg = 1;
    if eof then do;
      if found_msg = 0 then put "No messages found";
    end;
  run;
  filename log_file clear;
%mend;
```

Figure 8. “No messages found” Version 1

However, errors occurred. But there's no need to panic. Similarly, input the first error to debug. In fact, ChatGPT's initial response is already very close to the correct answer. At this point, it has slightly refined the answer based on the previous response (Figure 9).



You

Variable eof is uninitialized

ChatGPT

Apologies for the oversight. The `eof` variable needs to be initialized to properly identify the end of the file. Here's an updated version:

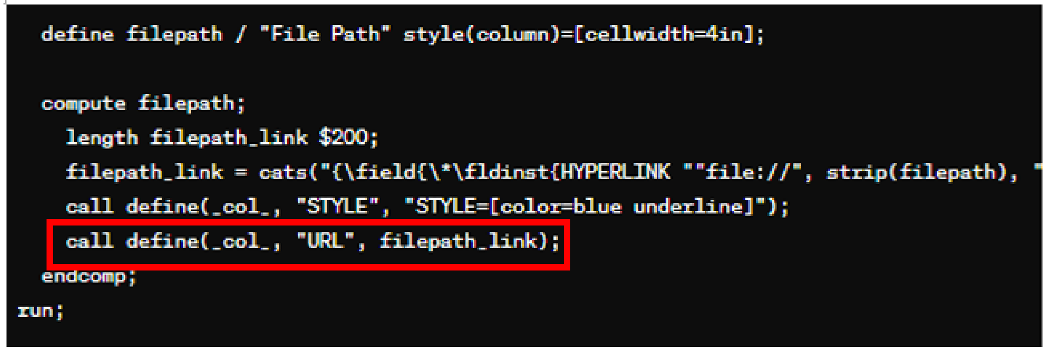
```
%macro check_log_for_message(logfile);
  filename log_file "&logfile";
  data _null_;
    retain found_msg 0;
    infile log_file length=len end=eof;
    input;
    if _infile_ =: 'YourExpectedMessagePattern' then found_msg = 1;
    if eof and found_msg = 0 then put "No messages found";
  run;
  filename log_file clear;
%mend;
```

Figure 9. “No messages found” Version 2

LEARN MORE ABOUT THE ANSWERS BY ASKING FOLLOW-UP QUESTIONS

Besides solving specific problems, ChatGPT can also serve as a learning tool. Explanations of basic concepts or code snippets can give beginners a preliminary understanding. Learning through ChatGPT is a convenient way because of the conversational pattern, although since ChatGPT is still in development, for accuracy, it is recommended to use it in conjunction with the official SAS documentation. Again, when dealing with specific code, it's important to verify extensively.

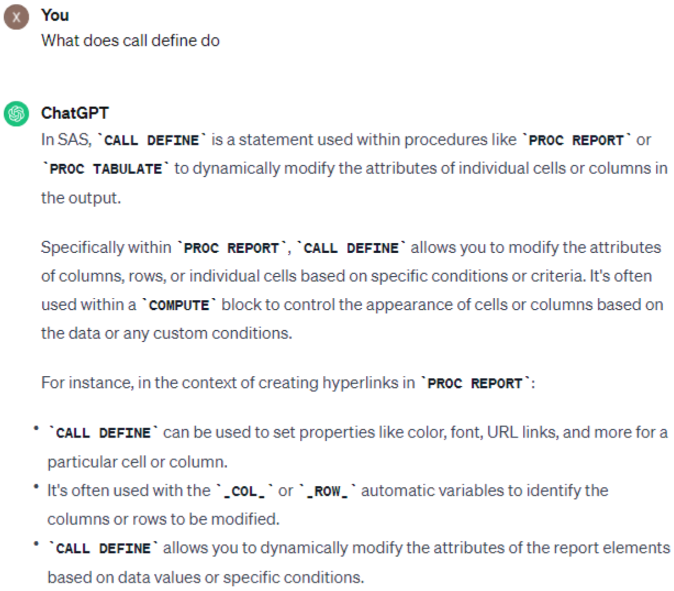
After a few rounds of debugging by inputting the errors into ChatGPT, finally the codes worked in the macro (Figure 10).



```
  define filepath / "File Path" style(column)=[cellwidth=4in];

  compute filepath;
    length filepath_link $200;
    filepath_link = cats("{\field{\*\fldinst{HYPERLINK ""file://", strip(filepath), "
    call define(_col_, "STYLE", "STYLE=[color=blue underline]");
    call define(_col_, "URL", filepath_link);
  endcomp;
run;
```

Figure 10. Add Link Final Version

Follow-up questions can also be asked when you want to learn more about why this code works. example, if you want to know more about why “call define” or “URL” should be used, you can ask the below question (Figure 11).



You

What does call define do

ChatGPT

In SAS, `CALL DEFINE` is a statement used within procedures like `PROC REPORT` or `PROC TABULATE` to dynamically modify the attributes of individual cells or columns in the output.

Specifically within `PROC REPORT`, `CALL DEFINE` allows you to modify the attributes of columns, rows, or individual cells based on specific conditions or criteria. It's often used within a `COMPUTE` block to control the appearance of cells or columns based on the data or any custom conditions.

For instance, in the context of creating hyperlinks in `PROC REPORT`:

- `CALL DEFINE` can be used to set properties like color, font, URL links, and more for a particular cell or column.
- It's often used with the `_COL_` or `_ROW_` automatic variables to identify the columns or rows to be modified.
- `CALL DEFINE` allows you to dynamically modify the attributes of the report elements based on data values or specific conditions.

Figure 11. Follow-up Question about `CALL DEFINE`

CONCLUSION

Through the opportunity of writing the macro, I had conversations with ChatGPT and summarized the below takeaway insights on effective questioning and considerations when using ChatGPT for SAS macro writing:

1. Break down functionalities of the SAS macro into a step-by-step process.
2. Include specific keywords and constraints to make the responses from ChatGPT more tailored to our needs.
3. Clearly express the problems by addressing one clear question or requirement at a time.
4. Be cautious when using the answers. It's best to test or run the SAS code to verify its accuracy since ChatGPT's responses may not always be reliable.
5. If encountering errors, there's no need to panic. Input the first error message into ChatGPT, and it can potentially suggest corrections based on its previous response.

When we're unsure about a task or have some ideas but don't know how to implement them in code, ChatGPT can be consulted. Compared to regular search engines, ChatGPT's advantage lies in providing tailored answers with abundant practical insights. For instance, it offers relatively complete code that can be directly used, with comments to help understanding. It serves as a valuable tool for learning and improving work efficiency. Our role is to decompose tasks and clearly express the requirements for each step, enabling effective questioning. However, at the same time, we shouldn't overly rely on or blindly follow ChatGPT, especially for SAS beginners. It's advisable to test and validate the provided code before adopting it.

REFERENCES

Nyakundi, Hillary. “How to Communicate with ChatGPT – A Guide to Prompt Engineering”. Accessed April 20, 2023. https://www.freecodecamp.org/news/how-to-communicate-with-ai-tools-prompt-engineering/.

Whitney, Lance. “That's Not Right: How to Tell ChatGPT When It's Wrong”. Accessed May 15, 2023. That's Not Right: How to Tell ChatGPT When It's Wrong | PCMag.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my manager Weijie for her support, trust, and encouragement all along.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Also, if you want to require the complete codes of the SAS macro, contact the author at:

Xinran (Flora) Luo
flora.luo@ecrscorp.com