CompareConditionalFormatting VBA Script Documentation

Overview

This VBA script is designed to compare and evaluate conditional formatting rules between two worksheets in an Excel workbook. It calculates a student's score based on the accuracy of their conditional formatting implementation when compared to a reference worksheet.

Prerequisites

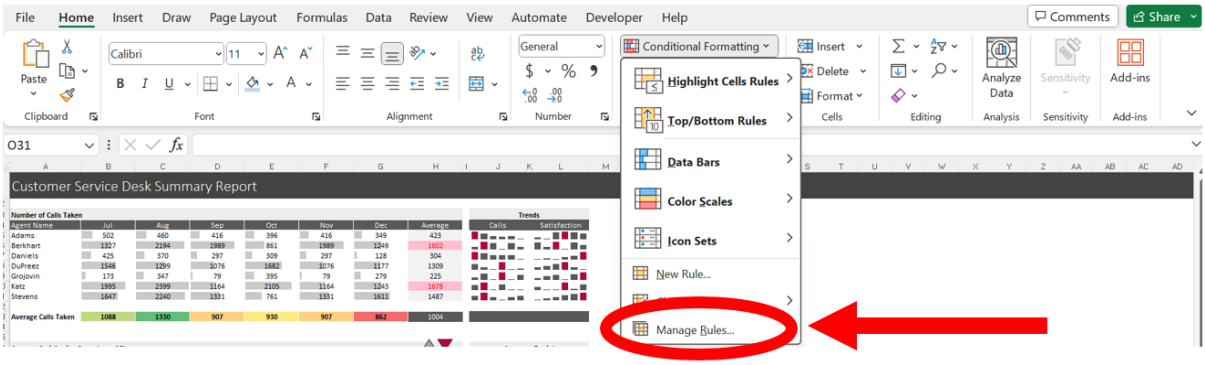
Before using this script, ensure the following:

- You have an Excel workbook open.

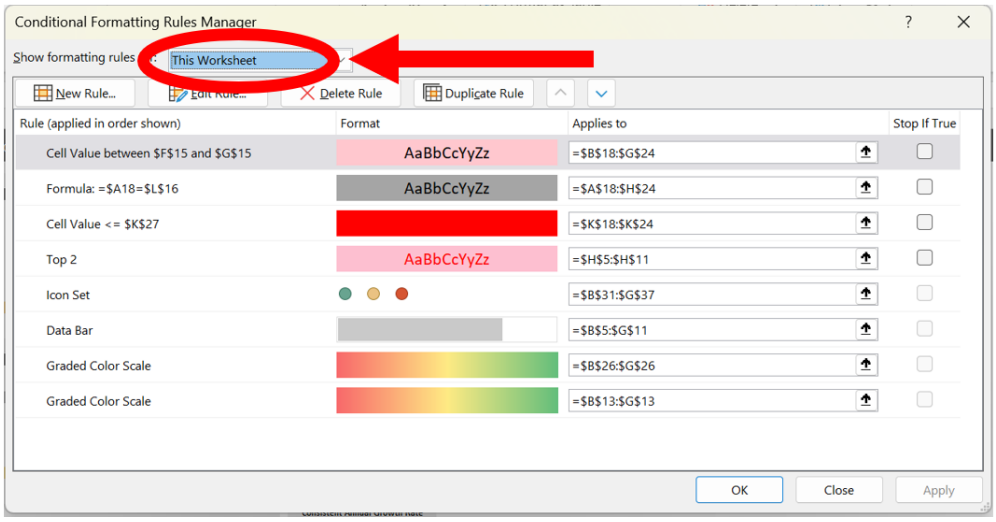
- The workbook contains three worksheets: "OfficialAnswer", "StudentAnswer", and "Feedback".

Checking the OfficialAnswer sheet is correct

Before using this script, you should look at the OfficialAnswer worksheet and select on the menu “Home -> Conditional Formatting -> Manage Rules”.



Then you can choose to show the list of conditional formatting rules for “This Worksheet” (see image below). When executed, the vba code will look specifically at these rules that you see on this menu item, and checking them one-by-one from top to bottom to see if they are on the studentAnswer sheet. The student does not need to have their formatting listed in the same order as the answer sheet.



You should reorder the formatting if you would like them marked in a different order, or remove any unnecessary formatting that you may have forgotten was there. The vba code will award up to 1 mark for each correct format from top to bottom down this list. And 1 mark for the correct number of conditions.

Usage

1. Open your Excel workbook.
2. In the VBA editor, insert a new module and paste the `CompareConditionalFormatting` code into it.
3. Execute the script by running it from the VBA editor.
4. The script will perform the following tasks:
    a. Check if the correct number of formatting conditions is applied to the "StudentAnswer" worksheet.
    b. Compare each conditional formatting rule in the "OfficialAnswer" worksheet with the corresponding rule in the "StudentAnswer" worksheet.
    c. On the “Feedback” sheet, write each format type in order and assign marks and feedback based on the accuracy of the student's formatting implementation.
    d. Calculate the total marks achieved by the student.
5. The script will generate a report in the "Feedback" worksheet, providing format type, feedback and marks for each formatting rule and the total score.

Example

See the worksheet for an example. If you delete the questions, feedback and marks on the Feedback worksheet before running the code, then you will be able to observe how the code will rewrite this feedback. If you change any formatting on the StudentAnswer worksheet, then you can rerun the code and observe how the feedback changes.

Modifications

1. **To change the formatting conditions checked for:**

The code is general and does not require modifications simply to check a completely different set of formatting of types 1-6. Simply create a new excel workbook with the formatting you wish to check, rename the worksheets OfficialAnswer, StudentAnswer, Feedback, and copy the code into the vba editor and it will work.

However, if you wish to check specific properties not currently checked by the code (and deduct marks if they are incorrect), then you will need to add these in the section of the vba code marked advanced grading.

You can ask ChatGPT for the code if you wish. The best way to do this is a prompt similar to the following, changing the 3 to the type of formatting that you wish to check.

Prompt:

Given this code for format conditions of type 1, can you duplicate with necessary changes for format conditions of type 3. Don't check types match or ranges match. The code I want you to duplicate is as follows:

Case 1, 2, xlExpression If rule1.Formula1 <> rule2.Formula1 Or rule1.Font.Bold <> rule2.Font.Bold Or ColorDifference(rule1.Interior.Color, rule2.Interior.Color) > 50 Or rule1.StopIfTrue <> rule2.StopIfTrue Then

ws3.Range("B" & ruleIndex).Value = 0.5

ws3.Range("C" & ruleIndex).Value = "Correct type and range, but the formula or color or similar used was incorrect"

totalMarks = totalMarks - 0.5 End If 'Debug.Print "Expression", rule1.Formula1, rule1.Font.Bold, rule1.Interior.Color, rule1.StopIfTrue

This prompt takes code that does work, and asks ChatGPT to modify it for another case. This is exactly how the types 3-6 was written in the current vba code.

But be careful. ChatGPT will often include code to look for formatting properties that do not exist and hence lead to compilation errors. In such a case tell ChatGPT there is an error with the properties and to try again. Repeat until the code works. If there is a very specific property you want to check, make sure you add a description of that property to the prompt.

2. **To modify the marks or feedback**

If you want to modify the marks awarded or feedback for certain cases, then search the code for:

ws3.Range("B" & ruleIndex).Value

Wherever this occurs is a location where a mark is awarded (or not) for a certain task. Below that code is always the line:

ws3.Range("C" & ruleIndex).Value

Where feedback for that mark is awarded. The default mark is (for each piece of formatting):

- 0 for missing formatting (including incorrect location or type),
- 0.5 for formatting of correct location and type, but some other error, and,
- 1 for correct formatting.

3. **To modify the names of the sheets**

The names of the worksheets checked are written once in the first few lines of code. You can modify these to be whatever you wish.

```
Set ws1 = ThisWorkbook.Sheets("OfficialAnswer")

Set ws2 = ThisWorkbook.Sheets("StudentAnswer")

Set ws3 = ThisWorkbook.Sheets("Feedback")
```