FileMaker Pro™

Determining Mode Values(s)

Paper prepared by R J Cologon PhD

August 2002

Mode - the most frequently occurring value (or values if there are more than one of equal frequency) is a useful indicator among a numeric sequence. In its current versions (ie up to v6) FileMaker pro provides a set of number, aggregate and summary functions, but these do not include some common statistical functions such as **mode**.

Unlike most other aggregate and summary functions, Mode may be null (eg when all values have the same frequency, or when all values are the same) – or being able to have multiple values (ie when more than one value is of equal highest frequency).

Requirements for mode values generally centre on a need to determine Mode for the found set of records within a file. The procedure described below provides a method for determining the value (or value) of Mode for the current found set, via a script, working in conjunction with several calculation and summary fields.

For the purposes of this explanation, it is assumed that the value for which the mode is to be calculated is in a pre-existing number field called ‘Amount’. The procedure is as follows:

1. Create a new number field called 'FoundAmount'.

2. Create a self-join relationship called "AmountCalc" which matches the 'FoundAmount' field to itself.

3. Create an unstored calculating field called 'AmountFrequency' (result type Number) with the formula: Count(AmountCalc:FoundAmount).

4. Create four summary fields called 'AmountMax' (defined as Maximum of 'FoundAmount'), 'AmountMean' (defined as Average of 'FoundAmount'), 'FrequencyMax' (defined as Maximum of 'AmountFrequency') and 'FrequencyMin' (defined as Minimum of 'AmountFrequency').

5. Create a text field called 'ModeAmount', and a global text field called 'gCurrentMode{s}'.

6. Create a value list called 'MultipleModes' which uses values from the 'ModeAmount' field.

7. Create a script called “Get Current Mode{s}” which runs along the lines of:

Code:

```
Allow User Abort [Off]
Go to Record/Request/Page [First]
Loop
  Set Field ["FoundAmount", "Amount"]
  Go to Record/Request/Page [Exit after last, Next]
End Loop
Go to Record/Request/Page [First]
Loop
  Set Field ["ModeAmount", "Case(AmountMean = AmountMax or
                  FrequencyMax = FrequencyMin, "[no mode]",
                  AmountFrequency = FrequencyMax, FoundAmount, ""))"]
  Go to Record/Request/Page [Exit after last, Next]
End Loop
Set Field ["gCurrentMode{s}", "Substitute(ValueListItems("YourDBname.fp5",
               "MultipleModes"), "¶", " ~ ")"]
Go to Record/Request/Page [First]
Set Field ["gFoundCount", "Status(CurrentFoundCount)"]
Set Field ["gFoundAmountMean", "AmountMean"]
Loop
  Set Field ["FoundAmount", ""]
  Set Field ["ModeAmount", ""]
  Go to Record/Request/Page [Exit after last, Next]
End Loop
Go to Record/Request/Page [First]
```

Now you can place the gCurrentMode{s} field on your layout (probably within the footer). The other fields mentioned - viz 'FoundAmount', 'AmountFrequency', 'AmountMax', 'AmountMean', 'FrequencyMax', 'FrequencyMin' and 'ModeAmount' - need not be shown on any layouts; they will operate 'behind the scenes'.

When the script is run, if there is no mode for the found set, the global field will display '[no mode]'. If there are one or more values for mode, each mode value will be shown in the global field, (multiple mode values will be separated by a tilde (~) character).

Refining the presentation

The chief limitation of this approach (apart from being a little tedious to set up) is that when a further 'Find' or an 'Omit' or 'Show All Records' command is performed, the global field value will no longer reflect the new found set, yet will remain on display until the script is run again or the field is cleared.

There are several methods to address this. One is to attempt to ensure that users will perform any subsequent actions using scripts you have provided, and add the step Set Field ["gCurrentMode{s}", ""] to each script. Another is to create a further four fields, as follows:

A. 'gFoundCount' - a global number field which is set by an additional initial step in the above script (SetField ["gFoundCount", "Status(CurrentFoundCount)"]).

B. 'gFoundAmountMean' - a global number field which is set by a second additional initial step in the above script (SetField ["gFoundAmountMean", "AmountMean"]).

C. 'CurrentMode{s}' - a calculating field of result type Text with the formula: Case(gFoundCount = Status(CurrentFoundCount) and gFoundAmountMean = AmountMean, gCurrentMode{s})

D. “ModeBarMessage” - a calculating field of result type Text with the formula: Case(gFoundCount = Status(CurrentFoundCount) and gFoundAmountMean =AmountMean, "Mode" & If(PatternCount(gCurrentMode{s}, "~"), "s", "") & " for the current 'found set' " & If(PatternCount(gCurrentMode{s}, "~"), "are", "is") & ":", "Click here to calculate MODE for the current found set")

Finally, place a wide button on your screen in the header or footer part, set the button text to:

<<ModeBarMessage>> <<CurrentMode{s}>>

…then link the button to the “Get Current Mode{s}” script.

The button will normally display the message “Click here to calculate MODE for the current found set” and, when clicked, will run the script and display a message along the lines of one of the following:

Mode for the current found set is: [no mode]
Mode for the current found set is: 37
Modes for the current found set are: 37 ~ 52 ~ 91

This ensures that in most circumstances, the Mode(s) will cease to display (ie the button text will revert to its default message) if the found set is changed, until/unless the script is run again to calculate a mode for the new found set or the found set for which mode was calculated is reinstated (eg via a ‘re-find’ procedure). This last procedure can be made increasingly robust by adding other "summary to stored global" comparisons to the formulae at steps C and D above, making it increasingly unlikely that the protection mechanism will fail. The risk (of a superseded Mode value being displayed) can thus be made extremely remote, though, theoretically, not entirely eliminated.

An unlocked demonstration file in .fp5 format which provides a working model of the technique described above is available for download from the NightWing web site. The file is available free of charge (as is this technique description) as a service to clients and fellow developers.

Prepared by:
R J Cologon, PhD
NightWing Enterprises
Melbourne, Australia
CobaltSky@nightwing.com.au
http://www.nightwing.com.au/FileMaker/