ExFunCollocation加载宏使用方法

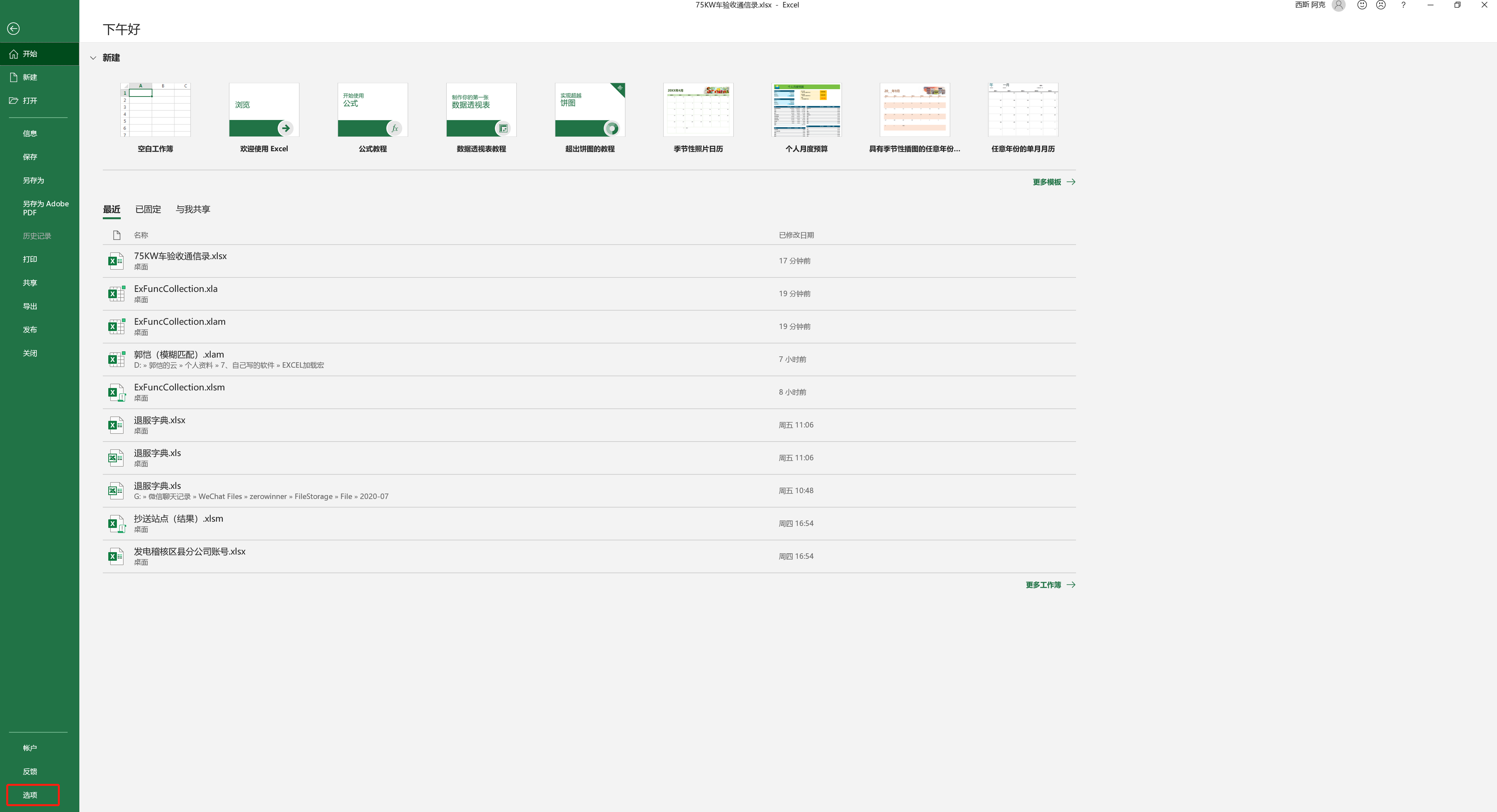
（开发者：郭恺13907203605）

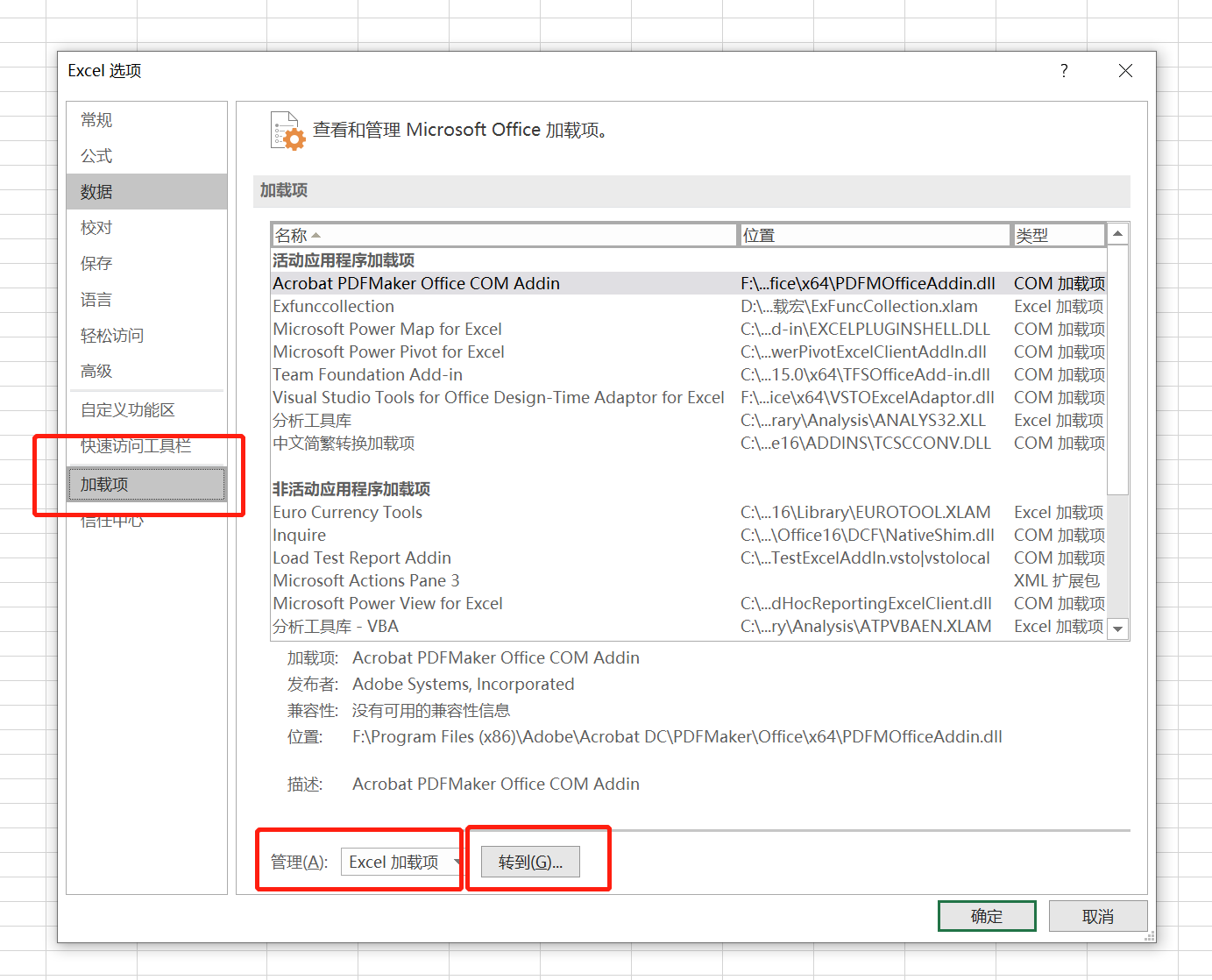
1. 什么是ExFunCollocation

ExFunCollocation是一个包含一些自定义函数的Excel加载宏，加载后能够使用一些额外的函数。

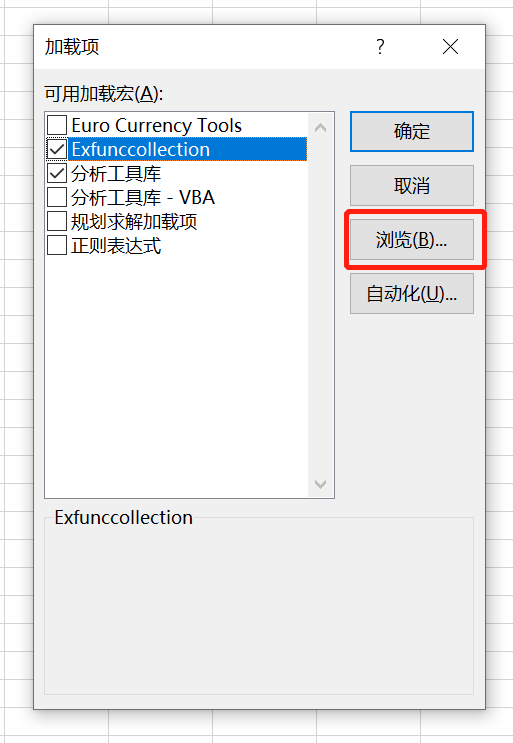
目前ExFunCollocation包含的函数有：

1. 模糊匹配（FuzzyMatching）
2. 使用正则表达式匹配（RegExp\_Match）
3. 多条件vlookup函数（MultiVlookup及SuperVlookup）
4. 多条件选择函数，等效于多个IF函数的嵌套（SelectCase）
5. 多条件选择函数（SelectIf）
6. 如何加载
7. 打开任意一个excel，点击文件、点击左下角的选项

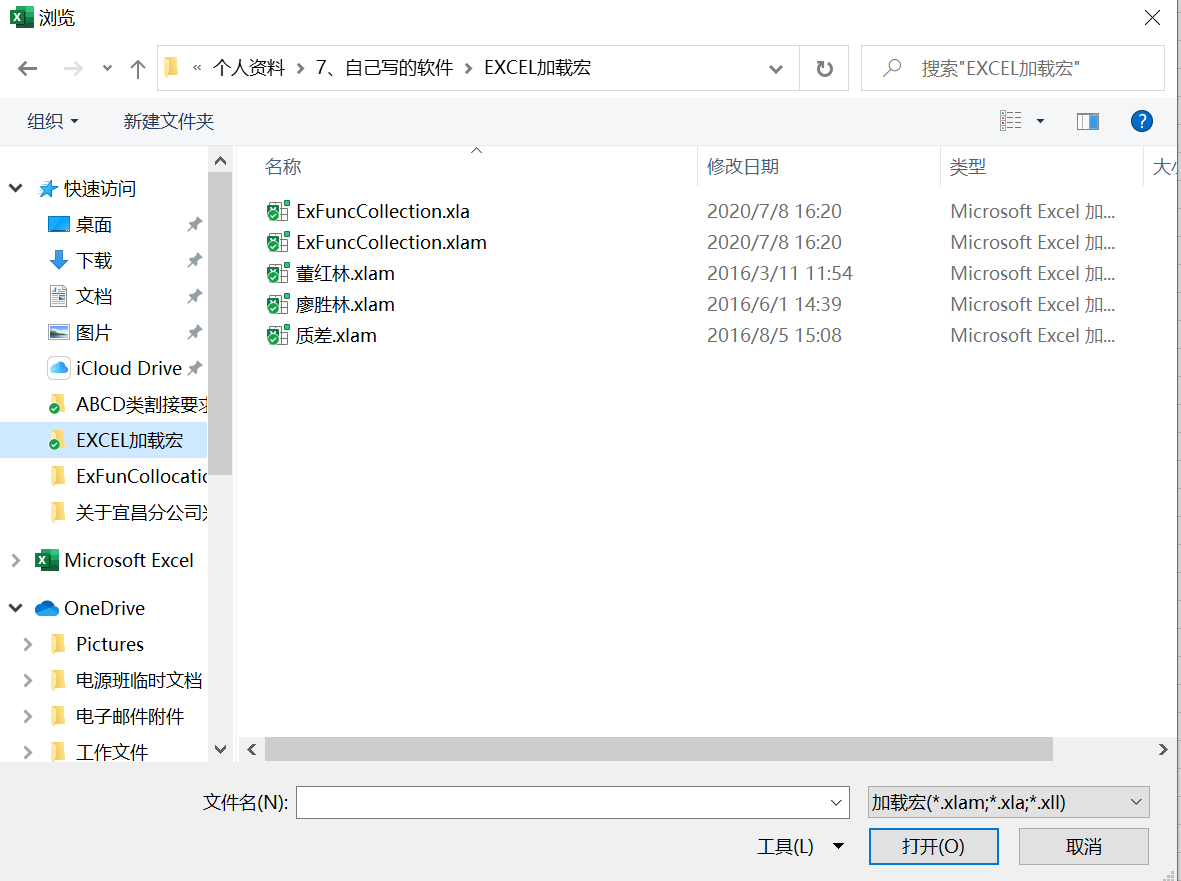


（二）点击左侧“加载项”、将“管理”选择为“Excel加载项”、点击“转到...” 

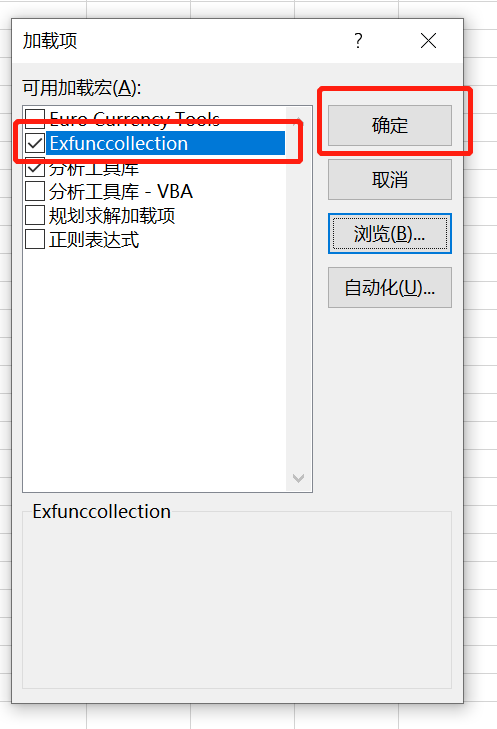
（三）点击“浏览...”



（四）选中“ExFuncCollection.xlam或ExFuncCollection.xla”文件，点击确定（excel2013及其以上使用.xlam，excel97-2007使用.xla）



（五）此时会回到步骤（三）的窗口，然后勾选“ExFuncCollection”，再点击确定。至此加载完毕。



（六）一台电脑只用加载一次，加载完后，在这个电脑打开的任意EXCEL文件都可以使用其中的函数，使用方法和excel内置函数一致。

1. 如何使用
   1. 模糊匹配（FuzzyMatching）
2. 概述

在给定丢失比例（相似度）的情况下，对内容进行模糊匹配，并返回最近似的结果。丢失比例设定太高的情况下，匹配结果并不理想，请谨慎使用。

1. 调用方法“FuzzyMatching.FuzzyMatching(MuBiao,QuYu,LossRate[,DisplayCount])”
2. 参数说明
   1. MuBiao：待查找的单元格，只能为单个单元格
   2. QuYu：用MuBiao单元格的内容在QuYu内查找可能的匹配，QuYu为一列
   3. LossRate：丢失比例，取值范围0-1，，相似度=1- LossRate，限定模糊匹配时的相识度，相似度不得低于1- LossRate，或者说，丢失比例不得高于LossRate
   4. DisplayCount：可选参数，类型为布尔型，默认值为False。值为False时，函数结果为在丢失比例小于给定值（LossRate）情况下，匹配到内容的个数。值为True时，函数结果为在丢失比例小于给定值（LossRate）情况下，对应丢失比例最小的内容。
   5. 使用正则表达式匹配（RegExp\_Match）
3. 概述

使用正则表达式对内容进行匹配，并返回匹配到的结果

1. 调用方法

“RegExp\_Match.RegExp\_Match(Sc,M[,Result])”

1. 参数说明
   1. Sc：待匹配的单元格或一列
   2. M：待匹配的文本的模式（使用正则表达式通配符构成的字符串）
   3. Result：可选参数，结果显示方式，取值为1或者2或者3，默认值为1

取值为1时，结果为匹配到的单元格的行的行号，如果有多个能够匹配的结果，行号之间用“、”连接

取值为2时，结果为匹配到的单元格数量

取值为3时，结果为匹配到的单元格的内容，如果有多个能够匹配的结果，行号之间用“、”连接

* 1. 多条件vlookup函数（MultiVlookup及SuperVlookup）

1. 概述

作用类似于vlookup函数，但可以指定任意多个比对条件，用指定的多组条件对指定区域内的每行进行比对，比对成功后返回该行指定的列的值

1. 调用方法

“SuperVLookup.SuperVLookup(ToFindArray,GoalCol, DisplayValue,Co1p,Co1M,Co1 [,Co2p,Co2M,Co2]...)”

1. 参数说明
   1. ToFindArray：区域，用于查找数据的区域，相当于Vlookup函数的第二个参数
   2. GoalCol：数值型，用ToFindArray的第GoalCol列作用返回结果的列，相当于Vlookup函数的第三个参数
   3. DisplayValue，字符串型，

当为“vaf”时，返回第一个匹配到的结果

当为“val”时，返回匹配到的所有结果的列表

当为“n”时，返回第所有能够匹配的行的数量

当为“rnf”时，返回第一个匹配到的结果的行号

当为“rnl”时，返回匹配到的所有结果行号的列表

* 1. Co1p,Co1M ,Co1为1组参数，包括后面的Co2p,Co2M,Co2,或者Co3p,Co3M,Co3等，必须成组出现或不出现，可以有任意多组，每组代表一个匹配条件
  2. Co1p：数值型，代表用ToFindArray中的第Co1p列与Co1单元格内的数据进行比较
  3. Co1M：字符串类型，比较模式，可以为“<”、“<=”、“=”、“>”、“>=”、“大于”、“大于等于”、“等于”、“小于”、“小于等于”、“包含”、“被包含”、“不包含”、“正则”、“RE”。其中“正则”、“RE”是同义词，标识比较模式为使用正则表达式比较，“<” 、“小于”为同义词，其他类似。
  4. Co1：单元格或一个字面量，用于和ToFindArray中的第Co1p列进行比较
  5. 举例：SuperVLookup.SuperVLookup(A1:D10,4, True,1,“=”,F1 ,3,“包含”,G1)
  6. ToFindArray被指定为A1:D10，为10行4列的区域
  7. GoalCol为4，比对成功后函数将返回对应行的D列的值
  8. DisplayValue为True，所以函数返回，对比成功的行的D列的值，而不是能够匹配上的行的数量
  9. Co1p为1，Co1M为“=”，Co1为单元格F1，可以理解为，第一组匹配条件为ToFindArray区域中第1列的值等于F1单元格内值
  10. Co2p为3，Co2M为“包含”，Co2为单元格G1，可以理解为，第二组匹配条件为ToFindArray区域中第3列的值（文本）包含G1单元格内值（文本）
  11. 等ToFindArray某行同时满足第一组及第二组条件时，返回该行的第4列（GoalCol为4）的单元格内的值。
  12. 对ToFindArray的同一列可以指定多组条件，也就是说Co1p、Co2p、Co3p...并不要求不能相同
  13. MutiVLookup是SuperVLookup一个简化版本，省去了参数Co1M、Co2M、Co3M...,所有比对操作锁定为“=”
  14. MutiVLookup调用方法为“MutiVLookup.MutiVLookup(ToFindArray,GoalCol, DisplayValue,Co1p,Co1[,Co2p,Co2]...)”，其他参数与SuperVLookup一致
  15. 多条件选择函数，等效于多个IF函数的嵌套（SelectCase）

1. 概述

多分支判断

1. 调用方法“SelectCase.SelectCase(Co1,T1[,Co2,T2]...)”
2. 参数说明
   1. Co1、T1：Co1为条件1，Co1为True时，返回T1
   2. Co2、T2：当Co1为False时，对Co2进行判断，当Co2为True时，返回T2
   3. Co3、T3等以此类推，可以有任意多分支
   4. 多条件选择函数（SelectIf）
3. 概述

多分支判断

1. 调用方法“SelectCase.SelectIf(R,Co1,T1[,Co2,T2]...)”
2. 参数说明
   * + - 1. R：单个单元格
         2. Co1、T1：Co1为条件1，Co1等于R时，返回T1
         3. Co2、T2：当Co1不等于R时，对Co2进行判断，当Co2等于R时，返回T2
         4. Co3、T3等以此类推，可以有任意多分支