教程

1.计算表说明

1.1 计算表功能简介

计算表是一款表格式数据的计算软件，可以对表格中的数据进行各种分析与计算：

- ✧ 电子表格的制作
- ✧ 表格式数据的分析与结构化计算
- ✧ 分析计算各类衍生数据，如同期对比、余额计算、份额分配等
- ✧ 对表格中的数据做排序、过滤、分组、多层合并、多层连接等各种变换

1.2 计算表安装与运行

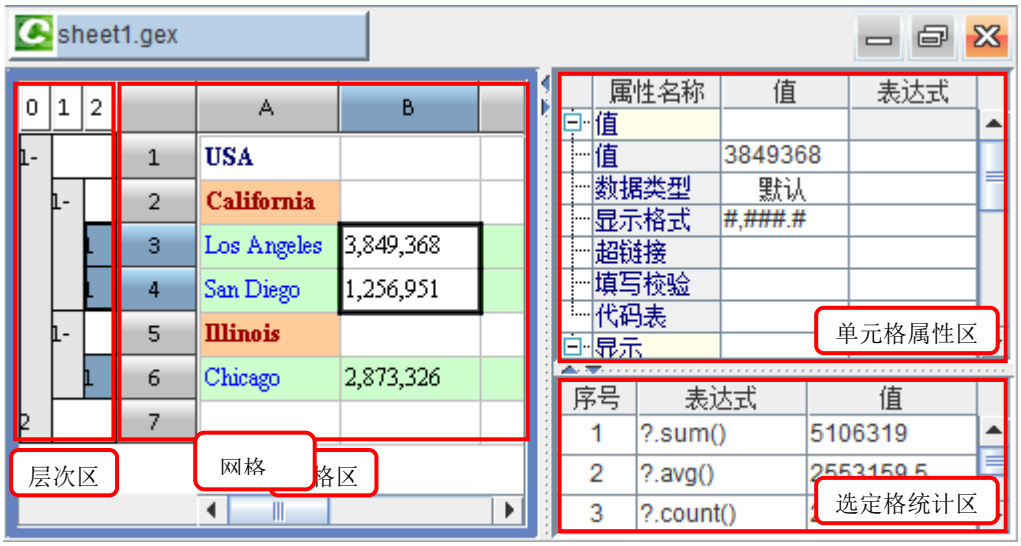
拿到安装包后执行安装程序，按提示逐步进行：

1) 运行安装程序
2) 按照提示，依次点击下一步，接受许可证协议
3) 选择安装路径，点击安装
4) 完成安装。

如果对 Java 的运行环境配置比较了解，而且本地已经安装了 JDK1.5 以上版本，也可以选择不自动安装 JDK 的计算表安装包来安装，在安装中需要填写本机 JDK 所在目录。

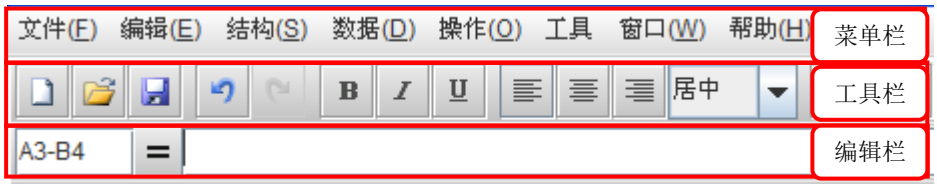
1.3 计算表界面简介，计算表文件的创建

运行计算表主程序，点击新建按钮可以**创建计算表文件**。创建计算表文件，还可以在菜单栏的栏中选择新建选项或者使用快捷键【Ctrl+N】。



左侧窗口中，右边的区域是**网格区**，网格区中是当前打开的计算表文件，左边的区域是**层次区**，显示计算表文件的层次结构；右侧窗口中，上边的区域是**单元格属性区**，可以查看及设定单元格的各种属性，下边的区域是**选定格统计区**，可以自动对选定格进行统计，包括合计，平均值，计数，也可以在选定格统计区中添加自定义统计表达式。

在计算表界面的最上部，也就是网格区的上方，是菜单栏、工具栏和编辑栏。



在菜单栏中，可以在各个菜单组中选择相应的菜单项来执行各种操作。在工具栏中可以直接点击按钮来执行一些最常用的操作。在编辑栏中，可以查看或者修改计算表中选中单元格的格值或者表达式。

2. 基础操作与序列

2.1 计算表中的数据类型

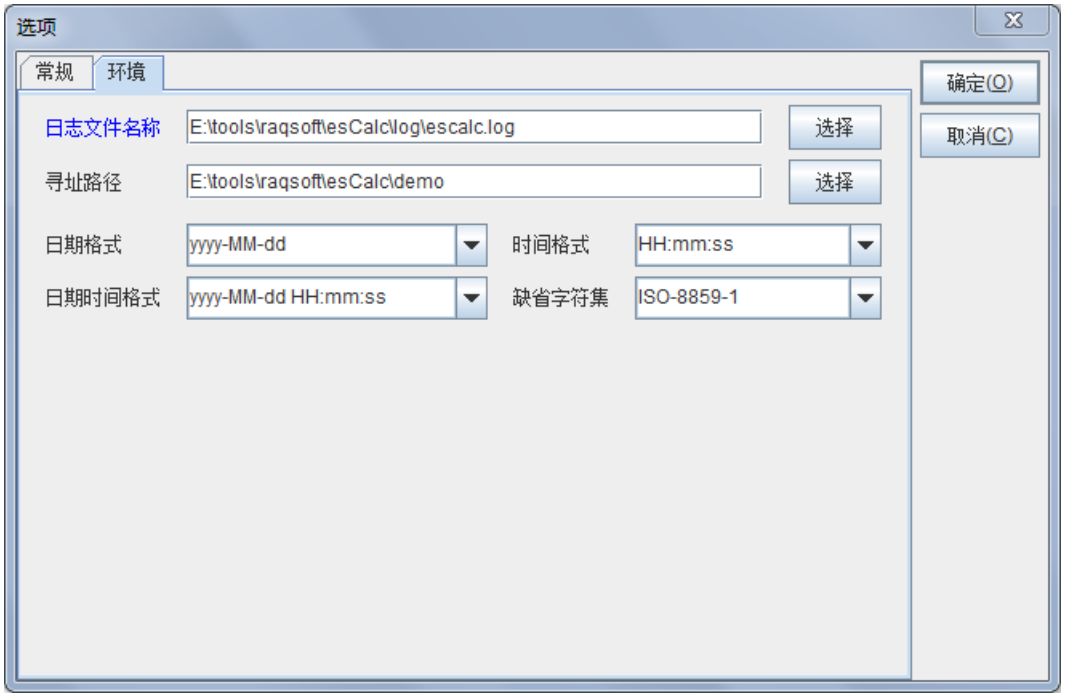
在计算表中，有以下几种数据类型：

- **数** 包括整数、实数、科学计数法表示的实数
  - **54、43.31、-4.45E13、3%**
- **布尔型** 包括 **true/false**
  - **true、false**
- **字符串** 在使用表达式时，用双引号括起来，其中转义字符用\，转义码采用 JAVA 的规定。而在直接定义字符串常量时，不用引号。
  - **abcd**
  - **="Georgia\tDelaware"**

使用表达式定义字符串时，转义规则与 JAVA 相同，可参考手册。

- **日期/时间** 写成 **yyyy-mm-dd** 和 **hh:mm:ss** 形式
  - **2010-1-3、23:04:23**

可以在菜单栏的工具选项中，点击选项按钮，在环境选项卡中，设定时间/日期数据的格式，以及字符编码等。



2.2 直接定义单元格值

在计算表中，格值可以直接在单元格中定义。这样的单元格称为**常数格**。

当格子的数据类型定义为**默认**时，根据格子中的数据，会解析为各种数据类型，如果无法理解，会把单元格中的格串解释为字符串。

0	1		A	B	C	D	E
1-		1	54	43.31	-4.45E3	3%	
	1	2	abcd	raqsoft			
2		3	2010-1-3	23:04:23			

在上面的格子中，A1,B1,C1 与 D1 格子中的格值都会被解释为实数，其中 C1 为使用科学计数法的格式的实数，D1 为百分数。A2 与 B2 中的格值会被解释为字符串。A3 中的格值会被解释为日期，B3 中的格值会被解释为时间。

- **注意：3%的写法在直接定义格值时是有效的，但是在输入后会被修改为 0.03。"%"作为百分号使用，只有在常数格中才是允许的，而不能使用在表达式中。**

在单元格中，还可以使用一些常数保留字，注意**大小写是敏感的！**

- **null** 空值，空格子的格值也是 null
- **true** 真
- **false** 假

在单元格处理格子中的数据或者表达式时，与计算表的层次设定是无关的。在本章剩余的示例中，将**不再展示计算表的层次**。

字符串常数格

格串是以英文单引号'开头的单元格，表示是格值为字符串的常数格，格值即为单引号后字串所构成的字符串。此时，单引号后面的所有字符都会被解析为字符串，不必再添加引号、转义符等字

符。

- **'abc\d 字符串 abc\d。**
- **'1234.5 字符串 1234.5。**

	A	B	C	D	E
1	'C:\files\gex	'=,>,<,+...	'1234.5	1234.5	

当字符串常数格中的字串输入后，格值会自动计算出实际的格值，并显示在格子中。也就是说，字符串前面的'会被去除，但是，在上方的编辑栏以及右方的单元格属性区中，显示的格值仍然是实际设定的以'开头的格串。

上例中，A1、B1 和 C1 均为字符串常数格，使用字符串常数格可以很方便地定义带有特殊字符、或者以等号开头的字符串，以及由数字所构成的字符串。如 D1 与 C1 的格值显示相同，但是 C1 中是字符串，D1 中是实数。

2.3 计算表中的表达式

如果单元格中的格串是用"="开头的，则等号后面的字串会被解析为表达式，而单元格的格值就是这个表达式的计算结果。这样的单元格称为**计算格**，单元格中的格值表达式又称为**计算式**。

- **="Hello"+" world"**
- **=1+4*3-6**

	A	B	C	D	E
1	="A"+" dog"	=1+4*3-6			

计算格会在右下方用青色的小三角标示。当计算式输入之后，可以计算出格值，并显示在格子中。A1 和 B1 在表达式输入后格值会显示为表达式的计算结果：

	A	B	C	D	E
1	A dog	7			

当选中计算格时，网格区的单元格中，会显示格值，即计算结果；而在上方的编辑区中会显示计算格的计算式。

计算格的格值和计算式都可以进行修改。选中单元格后，在单元格内点击，或者在编辑区中点击，可以分别编辑单元格值和计算式，也可以按下 F2 键在两者间切换。格子的计算式和格值也可以在右侧单元格属性区的单元格表达式属性中查看或者修改。

2.4 单元格的表示

计算表中的单元格定义为常数格或者计算格后，单元格中就有了格值。和使用参数一样，直接在表达式中使用单元格名称即可调用其它单元格的格值。

- **=A1*5**
- **=A1/B1+3*B1**

	A	B	C	D	E
1	4	2			
2	=A1*5	=A1/B1+3*B1			

A2 与 B2 中的表达式在输入后会可以进行计算，结果如下：

	A	B	C	D	E
1	4	2			
2	20	8.0			

2.5 立即计算格与联动计算格

在 **2.3 计算表中的表达式**中，我们讲述了在计算格中使用表达式的情况。在这里我们来了解一下计算表中计算格的详细使用方法。

2.5.1 立即计算格与联动计算格

当计算表处于自动计算模式时，当计算式写入单元格后，单元格会立即计算出格值，因此，这种计算格称为**立即计算格**。立即计算格只会在计算式输入时计算一次获得格值，在此之后，除非修改计算式，否则不会进行再次计算。如果计算式中涉及的单元格值被修改，立即计算格的格值也不会变化。

与立即计算格相对应，还有一种计算格称为**联动计算格**，它的不同之处仅在于计算式是以两个等号==作为开头的。也就是说，联动计算格中的计算式的开头比立即计算格多一个=。联动计算格除了在计算式输入或修改后计算，还会在调用的格值被修改时联动处理，重新计算。

	A	B	C	D	E
1	4	2			
2	=A1+3*B1	==A1+3*B1			

A2 为立即计算格，B2 为联动计算格，当单元格为联动计算格时，单元格的右下角会出现一个蓝色的小三角，不同单元格中小三角的颜色有可能不同。当 A2 与 B2 的计算式设定后，两个单元格中的格值都会计算出来，并显示在格子中：

	A	B	C	D	E
1	4	2			
2	10	10			

此时，如果 A1 或 B1 中的格值被改变，A2 与 B2 的格值会生成不同的结果：

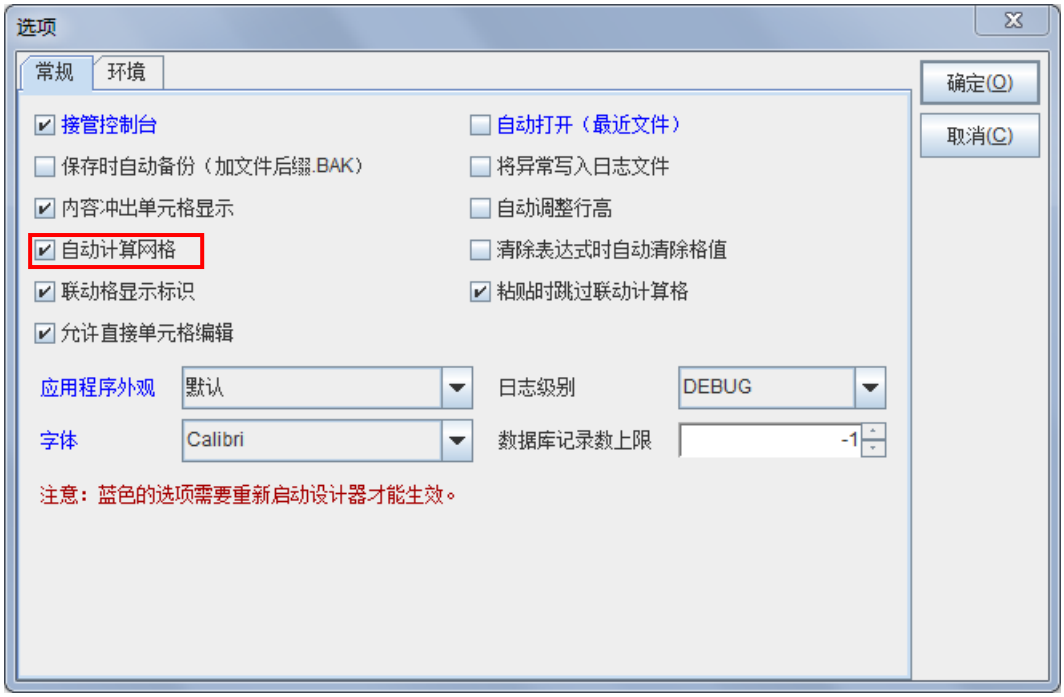
	A	B	C	D	E
1	5	4			
2	10	17			

只有联动计算格 B2 中的值会随着引用单元格的格值改变，立即计算格 A2 中的格值仍然保持原值。

联动计算格的格值是自动计算的，通常情况下是不需要修改的，一旦单元格的格值被修改，那么这个单元格会自动变为立即计算格。

2.5.2 是否自动计算网格

计算表中，在菜单栏中点击**工具→选项**，在选项设置窗口中可以设定是否自动计算网格。



当自动计算网格属性被选中时， 在计算表中编辑计算表时，立即计算格在计算式设定或修改完成后将会自动完成计算，联动计算格在计算式设定或修改，移进相关单元格中的数据修改时都会自动计算。

当自动计算网格属性未被勾选时，在计算表中编辑计算表时，立即计算格和联动计算格不会自动进行计算，都将在指定计算时进行计算。指定计算，可以在菜单栏中点击**数据→重算网格**，或者按快捷键 F9 进行。除非计算式被修改，否则立即计算格只会计算一次，而联动计算格在每次指定计算时都会重新计算。

2.6 计算表中的单元格属性

计算表中的单元格拥有很多属性，可以在右侧的单元格属性区中查看及设定。

	属性名称	值	表达式
⊟	值		
┆	值	17	=A1+3*B1
┆	数据类型	默认	
┆	显示格式		
┆	超链接		
┆	填写校验		
└	代码表		
⊟	显示		
┆	前景色		
┆	背景色		
┆	字体	Calibri	
┆	字大小	12	
┆	加粗	☐	
┆	斜体	☐	
└	下划线	☐	
⊟	段落		
┆	自动换行	☑	
┆	水平对齐	左对齐	
┆	纵向对齐	居中	
└	缩进	3.0	

单元格的属性主要分为三类：单元格值设定，单元格外观，段落设定。单元格值设定中可以查看及修改单元格的表达式、值、数据类型、显示格式、超链接、校验表达式等属性；单元格外观中可以查看及修改单元格的前景色、背景色、字体、字号、字型等属性；段落设定中可以查看及修改单元格中的文字是否自动换行、横向对齐、纵向对齐、缩进等属性。

2.7 认识序列

2.7.1 什么是序列

一些数据构成的有序集合即称为序列，构成序列的数据称为其成员。序列相当于高级语言中的数组，但其成员的数据类型不要求一致。

序号	1	2	3	4	5	...
成员	1 数数	true 布尔型	"abc" 字符型	1.1234 浮点数	2010-4-12 日期型	...

序列中的成员是**有序的**，根据序号，就可以找到某个成员。注意，序列的**序号是从 1 开始的**。

2.7.2 创建序列

➢ 常数序列

将成员用**[]**括起来即可表示常数序列。直接定义常数序列时，当序列中存在字符串时，不写双引号，计算表将会在输入完成后处理。

- **[1,3,4]**
- **[a,b,c]**
- **[s,2011-3-14,54]**

➢ 用表达式定义序列

用表达式直接定义序列，格式与常数序列类似，表达式中可以引用单元格，并可用冒号表示一片单元格区域，取区域时单元格值先行后列排列。

- **=[1,A4,B1:B4,C1:C3]**

0	1		A	B	C	D	E
1-		1	**1**	**2**	**3**		
	1	2	**4**	**5**	**6**		
	2	3	**7**	**8**	**9**		
	3	4	**[a,b,c]**	**[s,2011-03-14,54]**			
2		5	**=[1,A4,B1:B4,C1:C3]**				

A5 格中的表达式计算后，格值为：[1,[a,b,c],2,5,8,[s,2011-03-14,54],3,6,9]。

在用单元格名定义序列时，与计算表的层次设定是无关的。在序列这一章剩余的示例中，将**不再展示计算表的层次**，同时也**不对表达式单元格进行标示**。

- **空列**

没有成员的序列称作**空列**，可以直接用**[]**定义。

	A	B	C	D	E
1	**[]**				

2.8 序列的常用聚合运算

在计算表中，对于 *n* 序列 *A*，经常使用下面几个聚合运算函数来计算总数，合计，平均值等。

- ***A*.count()**

计算 *A* 中非空成员计数，和 *A*.len()不同。

- ***A*.sum()**

对 *A* 中所有成员求和。

- ***A*.avg()**

计算 *A* 中成员的平均值，计算时不计空值成员。

- ***A*.min()**

求最小值。

- ***A*.max()**

求最大值。

	A	B	C	D	E
1	**[null,4,6,,2,4,,5]**				
2	**=A1.count()**	**=A1.sum()**	**=A1.avg()**	**=A1.min()**	**=A1.max()**

A2 中表达式计算序列 A1 的非空成员个数，计算后格值为 5。

B2 中表达式返回序列 A1 中成员的和，计算后格值为 21。

C2 中表达式计算序列 A1 中中成员的平均值，计算后格值为 4.2，从结果中可见，avg 函数在计算序列平均值时，是不计序列中的空值的。

D2 中表达式计算序列 A1 中成员的最小值，计算后格值为 2，从结果中可见，min 函数在计算序列最小值时，是不计序列中的空值的。

E2 中表达式计算序列 A1 中成员的最大值，计算后格值为 6，max 函数在计算序列平均值时，同样是不计序列中的空值的。

序列的详细使用，请参看 **10 序列的高级应用**。

3. 计算表的结构

计算表的计算表中存储的数据，并不是一个简单的二维表，而是用层次结构相互关联的。

3.1 行的序号与层次

计算表的计算表文件中，每一行都拥有序号与层次，这些属性，可以在层次区中查看。

打开计算表 esCalc03_1.gex，如下：

0	1	2		A	B	C	D	E
1-			1	Employee				
	1		2	Department	Finance			
	2-		3	Female				
		1	4	Ashley Wilson	NY	11000		
	3-		5	Male				
		1	6	Daniel Davis	FL	10000		
	4		7	Count	2	Sum	21000	
	1		8	Department	R&D			
	2		9	Female				
	3-		10	Male				
		1	11	Justin Smith	TX	7000		
		1	12	Jacob Davis	TX	16000		
	4		13	Count	2	Sum	23000	
2			14	Count	4	Sum	44000	

在这个计算表中，可以看到，每一行在层次区中都会有一个编号，这个编号是该行的**序号**，如第 2 行和第 6 行的序号是 1，第 9 行的序号是 2，等等。

网格中，每一行的序号在层次区中所在的列，就是这一行的**层次**。在层次区顶端的第 1 行中，标记了每一层的层次号。上面的例子中，第 2 行和第 9 行的序号在层次区的第 2 列中，层次是 1，第 6 行的序号在层次区的第 3 列中，层次是 2。在计算表中，第 1 行的层次永远是 0，序号永远是 1。

3.2 主行、区段与续行

在计算表中，序号为 1 的行，称为**主行**。

从一个 *k* 层主行开始，直到遇到第一个同为 *k* 层的主行，或者层次小于 *k* 的任意行之前，这样的区域称为**区段**。区段第一行的 *k* 层主行也称为**区段主行**，而区段主行的层次 *k* 就是这个**区段的层次**。显然，在区段中，只会有一个 *k* 层主行。

在层次为 *k* 的区段中，有的时候会有多个层次为 *k* 的行，区段中层次为 *k* 的所有行均称为区段中的**续行**，区段中的主行也是续行之一。

重新打开计算表 esCalc03_1.gex，我们来了解一下主行、区段和续行。

0	1	2		A	B	C	D	E
1-			1	Employee				
	1		2	Department	Finance			
	2-		3	Female				
		1	4	Ashley Wilson	NY	11000		
	3-		5	Male				
		1	6	Daniel Davis	FL	10000		
	4		7	Count	2	Sum	21000	
	1		8	Department	R&D			
	2		9	Female				
	3-		10	Male				
		1	11	Justin Smith	TX	7000		
		1	12	Jacob Davis	TX	16000		
	4		13	Count	2	Sum	23000	
2			14	Count	4	Sum	44000	

在这个计算表中，序号为 1 的主行是第 1,2,4,6,8,11,12 行。其中，第 1 行的层次为 0，第 2,8 行的层次为 1，第 4,6,11,12 行的层次为 2。

再来看一下区段的概念：第 2 行是层次为 1 的主行，从它开始向下寻找，直到第 8 行，也是层次为 1 的主行，在它之前，从第 2 行到第 7 行这个区域就是一个区段，即红色方框标记的区域，这个区段的层次是 1。第 8 行也是层次为 1 的主行，从它开始向下寻找，第 14 行是层次为 0 的行，注意第 14 行并非主行，在它之前，从第 8 行到第 13 行这个区域也是一个层次为 1 的区段，用蓝色方框标出。另外，第 4,6,11,12 行，它们本身也可以看成是只有 1 行的，层次为 2 的区段。实际上，任何一个计算表，整体都可以视为是一个层次为 0 的区段。

在红色方框的区段内，第 2,3,5,7 行就是这个区段中的续行；同样在蓝色方框的区段中，第 8,9,10,13 行就是区段中的续行。

3.3 父行、子行与属行

在计算表中，*k* 层行上方的第一个 *k*-1 层行，称为 *k* 层行的**父行**，显然，0 层行是没有父行的，而其它行均有父行。与之类似，在 *k* 层行下方与之相邻的 *k*+1 层行，均为 *k* 层行的**子行**，计算表中的行并不一定有子行。在 *k* 层行下方，连续的层次大于 *k* 的行，称为 *k* 层行的**属行**。

重新打开计算表 esCalc03_1.gex，我们来了解一下父行、子行和属行。

0	1	2		A	B	C	D	E
1-			1	Employee				
	1		2	Department	Finance			
	2-		3	Female				
		1	4	Ashley Wilson	NY	11000		
	3-		5	Male				
		1	6	Daniel Davis	FL	10000		
	4		7	Count	2	Sum	21000	
	1		8	Department	R&D			
	2		9	Female				
	3-		10	Male				
		1	11	Justin Smith	TX	7000		
		1	12	Jacob Davis	TX	16000		
	4		13	Count	2	Sum	23000	
2			14	Count	4	Sum	44000	

第 4 行的父行是第 3 行，第 6 行的父行是第 5 行，第 11,12 行的父行是第 10 行，而第 2,3,5,7,8,9,10,13 行的父行都是第 1 行。注意，父行不一定是主行，也可能是续行。

子行其实是与父行相对应的，第 3 行的子行是第 4 行，第 5 行的子行是第 6 行，第 10 行的子行是第 11,12 行，第 1 行的子行是第 2,3,5,7,8,9,10,13 行。可见，子行也可能是主行或者续行。计算表中的行不一定有子行，比如第 9 行没有任何子行。

再来看一下属行。第 3 行的属行是第 4 行，第 5 行的属行是第 6 行，第 10 行的属行是第 11,12 行。第 1 行的属行是第 2 到 13 行，可见，计算表中，某一行的属行不仅仅包含它的所有子行，还包括所有子行的属行。

3.4 同构与同位

在计算表中，两行的子行的续行数量相同，且这些子行也对应满足这样的同构条件，则称这两行**同构**，或者称它们为**同构行**。

在计算表中，两行的层次和序号相同，且它们的父行也满足这样的同位条件，则称这两行**同位**，或者称它们为**同位行**。

打开计算表 esCalc03_2.gex，我们通过这个计算表来了解一下同构和同位的概念：

0	1	2	3		A	B	C	D	E
1-				1	Movie				
2-				2	2011 Grosses				
	1			3	PARAMOUNT			Total Gross	863,421,774
	2-			4	Ranking	PG-13		Ranking Total	739,944,167
		1-		5	Paramount				
			1	6	Transformers: Dark of the Moon				352,390,543
		2		7				Total	352,390,543
		1-		8	Paramount & Dreamworks				
			1	9	Mission: Impossible - Ghost Protocol				206,523,000
			1	10	Thor				181,030,624
		2		11				Total	387,553,624
	3-			12	Ranking	PG		Ranking Total	123,477,607
		1		13	Rango				123,477,607
	1			14	SONY / COLUMBIA			Total Gross	222,975,001
	2			15	Ranking	PG-13		Ranking Total	
	3-			16	Ranking	PG		Ranking Total	222,975,001
		1		17	The Smurfs				142,614,158
		1		18	Zookeeper				80,360,843
3-				19	2012 Grosses				
	1			20	Beauty and the Beast (3D)				
	2			21	Distributor	BUENA VISTA			
	3			22	Ranking	G			
	1			23	The Devil Inside				
	2			24	Distributor	PARAMOUNT			
	3			25	Ranking	R			

来了解一下什么是同构，本例中，第 12 行和第 16 行，它们的子行续行数量都是 1，所以它们是同构的。两个同构行，要求它们的子行的续行数量相同，但是并不要求它们的子行数量相同。

同构行并不一定层次相同，比如，第 5 行和第 12 行，虽然它们的层次分别是 2 和 1，但是它们是同构的。

第 4 行和第 12 行，它们子行的续行数不等，因此不是同构的。

再来看看第 2 行和第 19 行，它们子行的续行数都是 3，但是由于第 19 行的所有子行都没有子行，子行的续行数量是 0，因此它们和第 2 行的子行不满足同构条件，因此第 2 行和第 19 行不同构。

再来看一看同位的概念。第 7 行和第 11 行，它们的层次都是 2，序号都是 2，而它们的父行是相同的，都是第 4 行，因此第 7 行和第 11 行同位。

第 13,17,18 行，它们的层次都是 2，序号都是 1；它们的父行分别是第 12 行和第 16 行，而这两行的层次都是 1，序号都是 3，且父行同为第 2 行，满足同位。因此，第 12 行和第 16 行是同位行，第 13,17,18 行是同位的。我们同样也可以确定，第 6,9,10 行也是同位的。

再来看第 12 行和第 22 行，它们的层次都是 1，序号都是 3；但是它们的父行分别是第 2 行和第 19 行，第 2 行和第 19 行虽然层次同为 0，但是序号不同，分别为 2 和 3。因此第 2 行和第 19 行不

同位，而第 12 行和第 22 行也不同位。

0 层行中没有重复的序号，因此 0 层行是不可能有同位行的。

同位行必须是同构的，在执行增删子行或者增删续行等操作时，所有同位行都会同步执行。而即使同位行中某个行的子行全部删除，子行的结构也会保存。

3.5 同位区段

同位主行的区段，称为同位区段。

0	1	2	3		A	B	C	D	E
1-				1	Movie				
2-				2	2011 Grosses				
	1			3	PARAMOUNT			Total Gross	863,421,774
	2-			4	Ranking	PG-13		Ranking Total	739,944,167
		1-		5	Paramount				
			1	6	Transformers: Dark of the Moon				352,390,543
		2		7				Total	352,390,543
		1-		8	Paramount & Dreamworks				
			1	9	Mission: Impossible - Ghost Protocol				206,523,000
			1	10	Thor				181,030,624
		2		11				Total	387,553,624
	3-			12	Ranking	PG		Ranking Total	123,477,607
		1		13	Rango				123,477,607
	1			14	SONY / COLUMBIA			Total Gross	222,975,001
	2			15	Ranking	PG-13		Ranking Total	
	3-			16	Ranking	PG		Ranking Total	222,975,001
		1		17	The Smurfs				142,614,158
		1		18	Zookeeper				80,360,843
3-				19	2012 Grosses				
	1			20	Beauty and the Beast (3D)				
	2			21	Distributor	BUENA VISTA			
	3			22	Ranking	G			
	1			23	The Devil Inside				
	2			24	Distributor	PARAMOUNT			
	3			25	Ranking	R			

我们使用 **3.4 同构与同位**中看到的计算表实例来了解一下同位区段。同位主行，既要构成同位行，又必须是序号为 1 的主行。本例中的同位主行有以下几组：第 3,14 行，第 20,23 行，第 5,8 行，第 13,17,18 行，第 6,9,10 行。因此，分别以 3,14 行为区段主行的区段是层次为 1 的同位区段（用红色方框标出）；分别以 20,23 行为区段主行的区段也是层次为 1 的同位区段（用蓝色方框标出）；分别以 5,8 行为区段主行的区段是层次为 2 的同位区段（用橙色框标出）。

第 13,17,18 行都没有子行，它们可以视为层次为 2 的单行的同位区段；与之类似，第 6,9,10 行可以视为层次为 3 的单行的同位区段。

3.6 单元格所在行与层次

单元格所在行，就是指单元格所占用的行；如果单元格是合并格，且跨越了多行，则以第一行为准。

单元格的层次，就是单元格所在行的层次。

打开计算表 esCalc03_3.gex，我们通过这个计算表实例来了解一下单元格所在行与单元格的层次：

0	1	2		A	B	C	D	E
1-			1	Tennis Semifinals Result				
	1		2	Robert Foster vs. Ryan Norris				
	2-		3		Foster	Norris		1:3
		1	4	1	7	6	7:5	
		1	5	2	2	6		
		1	6	3	6	7	5:7	
		1	7	4	4	6		
	1		8	Nicholas Davis vs. Andy Moore				
	2-		9		Davis	Moore		3:2
		1	10	1	6	3		
		1	11	2	3	6		
		1	12	3	6	7	4:7	
		1	13	4	6	1		
		1	14	5	7	5		

如果单元格不是合并格，则单元格所在行与层次都很容易确定。如 B5 格，它所在的行就是第 5 行，而 B5 的层次就是第 5 行的层次：第 2 层。

如果单元格是合并格，但是处于同一行内，单元格所在行与层次同样很容易确定。如 A8:D8 这 4 个单元格构成的合并格，这个合并格所在行就是第 8 行，而这个合并格的层次就是第 8 行的层次：第 1 层。

如果单元格是跨越了多行的合并格，那么单元格所在行以合并格的第一行为准，合并格的层次也由这一行决定。如 E3:E7 这 5 个单元格构成的合并格，这个合并格所在行就是所在的首行：第 3 行，而它的层次就是它所在行的层次：第 1 层。

3.7 合并格

在计算表中允许使用合并格，合并格所在行以第一行为准。

在合并格中，位于它第一行第一列处的单元格，称为合并格的首格。一个合并格也可以用它的首格来表示。

设置合并格时，只要选中一片单元格，然后点击工具栏中的 按钮，或者在菜单栏中点击**编辑→合并**，即可设定合并格，再次点击，就会拆解合并格。

在 **3.6 单元格所在行与层次**中，我们看到了在计算表中使用合并格的情形，下面，仍然用上一节中的实例，来了解一下计算表中合并格的使用以及对合并格的要求。

0	1	2		A	B	C	D	E
1-			1	Tennis Semifinals Result				
	1		2	Robert Foster vs. Ryan Norris				
	2-		3		Foster	Norris		1:3
		1	4	1	7	6	7:5	
		1	5	2	2	6		
		1	6	3	6	7	5:7	
		1	7	4	4	6		
	1		8	Nicholas Davis vs. Andy Moore				
	2-		9		Davis	Moore		3:2
		1	10	1	6	3		
		1	11	2	3	6		
		1	12	3	6	7	4:7	
		1	13	4	6	1		
		1	14	5	7	5		

在计算表中，并不能随意把相邻的一片单元格合并为一个合并格。计算表中的合并格受结构限制，必须满足两个条件：第一，**合并格必须包含于所在行的区段中**；第二，**合并格中任一行在这个区段中的所有同位格也必须在合并格中**。

这样的要求，对于完全处于同一行内的合并格没有任何影响，如 A1,A2,A8 这些合并格。

对于处于多行中的合并格，我们用合并格 E3 为例来理解一下对合并格的要求。

先来看第一个要求。合并格 E3 所在的行是第 3 行，第 3 行所在的区段，就是以第 2 行为主行的区段，即第 2 至第 7 行，所以，E3 合并格必须包含在第 2 至第 7 行这个区域中。因此，如果想尝试着把 E3 至 E8 合并在一起的话，由于 E8 超出了第 3 行所在的区段，因此无法进行合并。如果想尝试着把 E2 至 E7 合并在一起的话，由于 E2 所在的行是第 2 行，第 2 行所在的区段也是第 2 行至第 7 行，因此这样的合并是允许的。

再来看第二个要求。在合并格 E3 所在的区段，也就是第 2 至第 7 行中，第 4,5,6,7 这四行是同位行，在这 4 行之中，只要有 1 行包含在合并格中，全部的 4 个同位行就都要包含在合并格中，也就是说，4,5,6,7 这 4 行同位行，或者全不在合并格中，或者全部在合并格中。如例子中的合并格 E3，包含了 E3:E7，也就是同时包含了 4,5,6,7 这 4 个同位行。与之类似，如果尝试把 A3 至 A7 合并，也是可以的。如果尝试把 E1 和 E2 这两个单元格进行合并，由于 4,5,6,7 这 4 个同位行都不在合并格中，因此也是允许的。如果尝试把 E3,E4,E5 这三个单元格进行合并，则由于第 4,5 行包含在合并格中，它们的同位行第 6,7 行并未包含在合并格中，因此无法进行合并。

如果尝试把 E4,E5,E6,E7 这四个单元格进行合并，又会如何呢？由于此时的合并格所在的行不是第 3 行或者第 2 行，而是第 4 行，而第 4 行所在的区段就只是由第 4 行单行构成的第 2 层区段，因此，E5,E6,E7 这三个单元格全都超出了区段的范围，因此是无法进行合并的。

计算表中，在同位区段中，合并格的设定是相同的，当在某个区段中设定合并格时，这个区段的所有同位区段中，也会自动同步设定合并格。

3.8 同位格

所在行为同位行，且位于同一列的单元格，称为**同位格**。同位格缺省使用相同的计算式和属性，在计算表中设定单元格的属性及计算式时，所有同位格中的属性及表达式都会同时被修改。

同位区段中，与区段层次相同的单元格必有同位格。

打开计算表 esCalc03_4.gex，我们通过这个计算表实例来了解一下同位格的概念：

0	1	2		A	B	C	D	E	F
1			1	Gymnastics					
2-			2	Name	Vault	UnevenBars	BalanceBeam	Floor	Total
	1-		3	Group A					
		1	4	Silva	14.175	14.175	14.175	14.35	56.875
		1	5	Pavlova	15.275	14.525	15.975	15.05	60.825
		1	6	Kaslin	15.35	14.275	14.425	13.95	58.0
	2		7		44.8	42.975	44.575	43.35	175.7
	1-		8	Group B					
		1	9	Downie	15.025	15.625	14.7	14.1	59.45
		1	10	Bonora	14.85	14.625	15.1	14.375	58.95
	2		11		29.875	30.25	29.8	28.475	118.4
3			12	Sum	74.675	73.225	74.375	71.825	294.1

在这个计算表实例中，第 3,8 行为同位行，第 4,5,6,9,10 行为同位行，第 7,11 行为同位行。在同位行中，位于同一列上的单元格，就是同位格。如 B4,B5,B6,B9,B10 这些单元格，它们所在的行是同位行，且都位于第 2 列，因此它们是同位格。再比如 A3 和 A8 这两个合并格，它们所在的行也是同位行，且都位于第 1 列，因此它们也是同位格。

又如 **3.7 合并格**的例子中，合并格 E3 和 E9，它们所在的行第 3 行和第 9 行是同位行，且它们都在第 5 列，因此也是同位格。

在计算表中，同位格默认情况下是具有相同的单元格属性的，如前景色、背景色、字体、字号、字型、字色、显示格式等等。当某个单元格中的这些属性被修改，所有的单元格的相应属性会同时被修改。

同位格的格值是可以各不相同的，但是，如果在同位格中使用计算式，所有同位格中的计算式也会类似属性一样会同时被修改。同位格中的计算式并不是完全相同，而是会根据所在的位置自动变迁。如上例中，F4 中计算总分的计算式是=round(B4+C4+D4+E4,3)，当 F4 中的计算式输入后，它的所有同位格，即 F5,F6,F9,F10 中的表达式也会被设定，并根据所处的位置自动变迁。在 F4 中计算式中，引用的本行的单元格，会被变迁为相应同位行中的单元格。如 F6 中的计算式为 =round(B6+C6+D6+E6,3)，F9 中的计算式为=round(B9+C9+D9+E9,3)。

需要注意的是，只有同位格中的计算式会在设置的时候自动设置并自动变迁，如 B7 和 C7，它们并不是同位格，因此并不会同步设置，也不会自动变迁。

此时，如果修改某个同位格中的属性，如 A6 的背景色，此时，A6 的同位格 A4,A5,A9,A10 中的背景色也会一同改变。与之类似，如果要把 F10 中的计算式修改为=int(B10+C10+D10+E10)，则 F10 的同位格 F4,F5,F6,F9 中的计算式也会随之改变，同时自动迁移。修改之后，计算表如下所示：

0	1	2		A	B	C	D	E	F
1			1	Gymnastics					
2-			2	Name	Vault	UnevenBars	BalanceBeam	Floor	Total
	1-		3	Group A					
		1	4	Silva	14.175	14.175	14.175	14.35	56
		1	5	Pavlova	15.275	14.525	15.975	15.05	60
		1	6	Kaslin	15.35	14.275	14.425	13.95	58
	2		7		44.8	42.975	44.575	43.35	175.7
	1-		8	Group B					
		1	9	Downie	15.025	15.625	14.7	14.1	59
		1	10	Bonora	14.85	14.625	15.1	14.375	58
	2		11		29.875	30.25	29.8	28.475	118.4
3			12	Sum	74.675	73.225	74.375	71.825	294.1

3.9 主格

区段中，续行上的某个单元格可以被设定为**主格**。主格在区段中唯一，同位区段中的主格必须是同位的。在计算表中，如果在某个续行上设定了主格，则这个单元格的所有同位格也会被自动设为主格。

如果需要将某个单元格设定为主格，或者需要将某个主格设定取消，在单元格上按下鼠标右键，在右键菜单中，的**单元格**选项下，单击**设置/取消主格**按钮即可。

我们使用 **3.8 同位格**中，修改过同位格的计算表实例，或者直接打开计算表 esCalc03_5.gex，然后将 A3,F6 两个单元格设为主格。设定后，结果如下：

0	1	2		A	B	C	D	E	F
1			1	Gymnastics					
2-			2	Name	Vault	UnevenBars	BalanceBeam	Floor	Total
	1-		3	Group A					
		1	4	Silva	14.175	14.175	14.175	14.35	56
		1	5	Pavlova	15.275	14.525	15.975	15.05	60
		1	6	Kaslin	15.35	14.275	14.425	13.95	58
	2		7		44.8	42.975	44.575	43.35	175.7
	1-		8	Group B					
		1	9	Downie	15.025	15.625	14.7	14.1	59
		1	10	Bonora	14.85	14.625	15.1	14.375	58
	2		11		29.875	30.25	29.8	28.475	118.4
3			12	Sum	74.675	73.225	74.375	71.825	294.1

当某个单元格被设为主格时，这个单元格的左上角会出现一个小三角做出标示，小三角的颜色会根据单元格底色的不同而有所不同。可以发现，当某个单元格被设为主格时，它的所有同位格也会被设为主格，如 A8 也随 A3 一同被设为主格，F4,F5,F9,F10 都随着 F6 一同被设为主格。

主格在续行所在的区段内是唯一的，也就是说，同一行中只会有一个主格。

如果尝试着把 A5 设为主格，设定后的结果如下：

0	1	2		A	B	C	D	E	F
1			1	Gymnastics					
2-			2	Name	Vault	UnevenBars	BalanceBeam	Floor	Total
	1-		3	Group A					
		1	4	Silva	14.175	14.175	14.175	14.35	56
		1	5	Pavlova	15.275	14.525	15.975	15.05	60
		1	6	Kaslin	15.35	14.275	14.425	13.95	58
	2		7		44.8	42.975	44.575	43.35	175.7
	1-		8	Group B					
		1	9	Downie	15.025	15.625	14.7	14.1	59
		1	10	Bonora	14.85	14.625	15.1	14.375	58
	2		11		29.875	30.25	29.8	28.475	118.4
3			12	Sum	74.675	73.225	74.375	71.825	294.1

可以看到，在同位行第 4,5,6,9,10 这些行中，同位格 A4,A5,A6,A9,A10 被设定为了新的主格，新设的主格会取代旧的主格。

当网格中进行对位、分组、连接、合并等数据关联操作时，会使用到主格，将在 **8.数据操作**的相应小节中进行详细的说明。

4. 与结构相关的函数

4.1 表达式的自动变迁

在计算表中，如果在单元格中输入计算式，那么这个单元格中的所有同位格中也会被同时设入计算式，并自动变迁。如果在单元格的某个属性中使用表达式，也会统一处理。

下面我们打开计算表 esCalc04_1.gex，来了解一下计算表是如何对表达式进行自动变迁的。

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6	28	Sat.	29	20	
		1-		7	week 5				
			1	8	29	Sun.	30	18	
			1	9	30	Mon.	24	16	
			1	10	31	Tues.	25	12	
	1-			11	February				
		1-		12	week 1				
			1	13	1	Wed.	35	11	
			1	14	2	Thur.	38	16	
3				15					

在上面的计算表中，统计了某地区一段时间内每一天的最高气温和最低气温，并按照不同的月份、周分层展示。

现在如果在 E5 格中，写入计算式=C5-D5，计算每天的温差，那么 E5 的同位格 E6,E8,E9,E10,E13,E14 这些单元格中，也会自动设入计算式，并自动变迁，计算结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	7
			1	6	28	Sat.	29	20	9
		1-		7	week 5				
			1	8	29	Sun.	30	18	12
			1	9	30	Mon.	24	16	8
			1	10	31	Tues.	25	12	13
	1-			11	February				
		1-		12	week 1				
			1	13	1	Wed.	35	11	24
			1	14	2	Thur.	38	16	22
3				15					

点击拥有计算式的这些单元格，在右侧的单元格属性列表中，可以看到每个单元格中的计算式变迁后的结果。如 E6 中计算式为=C6-D6，E9 中计算式为=C9-D9，E13 中计算式为=C13-D13 等等。可以看到，如果计算式中，引用了本行的单元格，在它的同位格的计算式中，也会相应地变迁为各自行内的单元格。

如果在 E6 中写入的计算式修改为=C6-C5，计算最高气温与前一天的差值，又会如何呢？E6 同位格中的表达式也会自动调整变迁，计算后，结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	28
			1	6	28	Sat.	29	20	1
		1-		7	week 5				
			1	8	29	Sun.	30	18	1
			1	9	30	Mon.	24	16	-6
			1	10	31	Tues.	25	12	1
	1-			11	February				
		1-		12	week 1				
			1	13	1	Wed.	35	11	10
			1	14	2	Thur.	38	16	3
3				15					

点击进入每个拥有计算式的单元格，可以查看它们的计算式。可以发现，由于 E6 的计算式中，引用了它前一个同位行中的单元格 C5，所以它的同位格中，这个引用的单元格也相应变为各自的前

一个同位行中的单元格。如，E9 中的计算式为=C9-**C8**，E13 中计算式为=C13-**C10**，等等。需要注意的是，E5 中的计算式为=C5-#REF!，这是由于第 5 行是它所有同位行中的第 1 行，是无法对应找到“前一个同位行”的，因此相应的单元格也无法找到。单元格无法找到时，单元格值按照空值或 0 处理。

如果在 E5 中写入的计算式修改为=left(A1,4)+right(A4,2)，又会如何呢？计算后，结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	Temp 4
			1	6	28	Sat.	29	20	Temp 4
		1-		7	week 5				
			1	8	29	Sun.	30	18	Temp 5
			1	9	30	Mon.	24	16	Temp 5
			1	10	31	Tues.	25	12	Temp 5
	1-			11	February				
		1-		12	week 1				
			1	13	1	Wed.	35	11	Temp 1
			1	14	2	Thur.	38	16	Temp 1
3				15					

查看 E5 同位格的计算式，可以发现，在计算式中，引用的各层主行中的单元格，如 A4，会自动变迁为各个同位格中相应层次各自主行中的相应单元格；如果引用的单元格与当前格没有层次联系，如 A1，则不会变化。例如，A6 中计算式为=left(A1,4)+right(A4,2)，A9 中计算式为=left(A1,4)+right(A7,2)，A13 中计算式为=left(A1,4)+right(A12,2)，等等。

4.2 [*a*:*b*]与{*a*:*b*}

[*a*:*b*]与{*a*:*b*}都是生成单元格序列的常用函数，其中参数 *a* 和 *b* 都是单元格名称。

- **[*a*:*b*]**

  从 *a* 到 *b* 的格值序列。

- **{*a*:*b*}**

  从 *a* 到 *b* 且与 *a* 同位的格值序列。

使用[*a*:*b*]生成单元格序列时，与网格的层次结构完全无关。使用{*a*:*b*}生成单元格序列则不同，只会返回与 *a* 同位的格值序列，相当于从[*a*:*b*]生成的单元格序列中，选出与 a 同位的单元格构成格值序列。

打开计算表 esCalc04_2.gex，通过这个计算表实例，我们来看一下这两个函数：

0	1	2		A	B	C	D
1-			1	Turnover			
	1-		2	Quarter 1			
		1	3	January	13400.50		
		1	4	February	13425.00		
		1	5	March	12800.00		
	1-		6	Quarter 2			
		1	7	April	15333.40		
		1	8	May	11800.80		
2			9	Count			
3			10	=[A5:B7]			
4			11	={A5:B7}			

计算后，A10 中单元格值为序列["March",12800,"Quarter 2",null,"April",15333.4]。通过计算结果可以看到，在返回的单元格值序列中，包含 A5 至 B7 这个区域内的所有的单元格，按照先行后列的顺序构成序列，合并格的格值只存在于首格中，如 A6 中的格值为"Quarter 2"，而 B6 不是合并格首格，格值为空值 null。

A11 中单元格值为序列["March","April"]，可见，在指定的区域内，只有 A5 及同位格 A5,A7 的格值被选出构成序列。

4.3 行号与层序号

- **row()**

返回当前行的行号。

我们使用 **4.2[*a*:*b*]与{*a*:*b*}**中的计算表实例，在 D2 和 D3 中添加计算式=row()，输入后，D2 和 D3 的所有同位格中也会被设入同样的计算式，同时，把 A10 和 A11 中的计算式同样修改为=row()，结果如下：

0	1	2		A	B	C	D
1-			1	Turnover			
	1-		2	Quarter 1			2
		1	3	January	13400.50		3
		1	4	February	13425.00		4
		1	5	March	12800.00		5
	1-		6	Quarter 2			6
		1	7	April	15333.40		7
		1	8	May	11800.80		8
2			9	Count			
3			10	10			
4			11	11			

从各个计算格中的结果中可以看到，使用 row()函数时，能够返回单元格所在的行号，与单元格的层次，单元格所在的列等都无关。

➢ **ord(*A*,*L*)**

层 *L* 范围内 *A* 的同位格序号，*L* 仅是层。

为了了解这个函数，我们修改上面的计算表，删除 D 列中单元格的计算式，添加一个空行，并将 A10 中计算式设为=ord(A7,A2)，A11 中计算式设为=ord(A7,A1)，A12 中计算式设为=ord(A7)，再在 D3 中添加计算式=ord(D3)，结果如下：

0	1	2		A	B	C	D
1-			1	Turnover			
	1-		2	Quarter 1			
		1	3	January	13400.50		1
		1	4	February	13425.00		2
		1	5	March	12800.00		3
	1-		6	Quarter 2			
		1	7	April	15333.40		1
		1	8	May	11800.80		2
2			9	Count			
3			10	1			
4			11	4			
5			12	4			

来分析一下 A10 中计算式=ord(A7,A2)的计算过程，A2 格的层次是 1，计算时首先找到单元格 A7 所在的层次为 1 的区段主行，即第 6 行，然后计算出 A7 所在的行在这个区段中的同位格序号，在第 6 行为主行的区段内，A7 的同位格是 A7 和 A8，A7 在其中的序号为 1，因此 A10 中的计算结果是 1。

再看 A11 中的计算式=ord(A7,A1)，A1 格的层次是 0，A7 所在的层次为 0 的区段主行是第 1 行，在以第 1 行为主行的区段中，A7 的同位格是 A3,A4,A5,A7,A8，A7 在其中的序号为 4，因此 A11 中的计算结果是 4。

再看 A12 中的计算式=ord(A7)，当函数参数中的 *L* 省略时，会按照当前格的层次计算，A12 的层次和 A1 的层次相同，都是 0，因此 A12 中的计算结果与 A11 中相同，也是 4。函数中，*A* 不能省略。

最后再来看一下 D3 及其同位格中的情况，D3 中的计算式为=ord(D3)，当 D3 中的计算式设定后，它的同位格 D4,D5,D7,D8 中的计算式也自动设定，并自动变迁，如 D4 中计算式为=ord(D4)，D8 中计算式为 ord(D8)…当 ord(*A*,*L*)函数中的 *A* 为当前格，且 *L* 省略的时候，*L* 不会设定为当前格，而是设定为当前格所在行的父行，这与 A12 中的情况是不同的。也就是说，*A* 为当前格的时候，ord(*A*)为当前格在父行区段中的同位格序号。D3,D4,D5 的父行是第 2 行，它们的同位格序号分别为 1,2,3；而 D7,D8 的父行是第 6 行，它们的同位格序号为 1,2。

需要注意的是，在计算时，*L* 不一定在 *A* 所在的区段中（如 A12 中的表达式），计算 ord(*A*,*L*)函数时只根据 *L* 的层次来计算。

在计算 ord(*A*,*L*)函数时，*A* 必须有与 *L* 的层次相等的父行，当 *L* 与 *A* 层次相同时，是无法计算出结果的，会产生错误。

➢ **num(*A*,*L*)**

层 *L* 范围内 *A* 的同位区段数量，*L* 仅是层。

为了了解这个函数，我们修改上面的计算表，将 A10 中计算式设为=num(A7,A2)，A11 中计算

式设为= num(A7,A1)，A12 中计算式设为= num(A6)，再在 D3 中添加计算式=num(D3)，结果如下：

0	1	2		A	B	C	D
1-			1	Turnover			
	1-		2	Quarter 1			
		1	3	January	13400.50		3
		1	4	February	13425.00		3
		1	5	March	12800.00		3
	1-		6	Quarter 2			
		1	7	April	15333.40		2
		1	8	May	11800.80		2
2			9	Count			
3			10	2			
4			11	5			
5			12	2			

来分析一下 A10 中计算式=num(A7,A2)的计算过程，A2 格的层次是 1，计算时首先找到单元格 A7 所在的层次为 1 的区段主行，即第 6 行，然后计算出 A7 所在的行在这个区段中的同位区段数量，也就是 A7 的同位格的数量，在第 6 行为主行的区段内，A7 的同位格是 A7 和 A8，总数为 2，因此 A10 中的计算结果是 2。

再看 A11 中的计算式=num(A7,A1)，A1 格的层次是 0，A7 所在的层次为 0 的区段主行是第 1 行，在以第 1 行为主行的区段中，A7 的同位格是 A3,A4,A5,A7,A8，总数为 5，因此 A11 中的计算结果是 5。

再看 A12 中的计算式=num(A6)，当函数参数中的 *L* 省略时，会按照当前格的层次计算，A12 的层次是 0，A6 的 0 层主行是第 1 行，在第 1 行的区段中，A6 的同位格是 A2 和 A6，共 2 个，因此 A12 中计算结果是 2。函数中，*A* 不能省略。

与 ord(*A*,*L*)函数中的情况类似，当 num(*A*,*L*)中，*A* 为当前格，且 *L* 省略的时候，*L* 不会设定为当前格，而是设定为当前格所在行的父行。也就是说，*A* 为当前格的时候，num(*A*)为当前格在父行区段中的同位格总数。D3,D4,D5 的父行是第 2 行，它们的同位格总数为 3；而 D7,D8 的父行是第 6 行，它们的同位格总数为 2。

在计算时，同样不要求 *L* 在 *A* 所在的区段中，计算 num(*A*,*L*)函数时只根据 *L* 的层次来计算。在计算 num(*A*,*L*)函数时，*A* 必须有与 *L* 的层次相等的父行，当 *L* 与 *A* 层次相同时，是无法计算出结果的，会产生错误。

4.4 层序号的简写

- #

  相当于 ord(当前格)，即当前格在父行区段中的同位格序号。

- ##

  相当于 num(当前格)，即当前格的父行区段中的同位格数量。

我们使用 **4.3 行号与层序号**中的计算表实例，在 D2,D3 中设定计算式=#，在 E2,E3 中设定计算式=##，或者直接打开计算表 esCalc04_3.gex，来了解一下这两个函数。计算结果如下：

0	1	2		A	B	C	D	E
1-			1	Turnover				
	1-		2	Quarter 1			1	2
		1	3	January	13400.50		1	3
		1	4	February	13425.00		2	3
		1	5	March	12800.00		3	3
	1-		6	Quarter 2			2	2
		1	7	April	15333.40		1	2
		1	8	May	11800.80		2	2
2			9	Count				
3			10					
4			11					
5			12					

D2,D3,E2,E3 中设定计算式后，它们的同位格中也会自动设定计算式。D2:D8 区域中，所有单元格的计算式都是=#，返回它们在各自父行的区段中的同位格序号；E2:E8 区域中，所有单元格的计算式都是=##，返回它们各自父行区段中的同位格数量。

- ***L#***

相当于 ord(当前格,*L*)，即当前格在层 *L* 范围内的同位格序号。

- ***L##***

相当于 num(当前格,*L*)，即当前格在层 *L* 范围内的同位区段数量。

我们仍然使用上面的计算表实例，来了解一下这两个函数。

在 D2 和 D3 中设定计算式=A1#，在 E2 和 E3 中设定计算式=A1##，计算结果如下：

0	1	2		A	B	C	D	E
1-			1	Turnover				
	1-		2	Quarter 1			1	2
		1	3	January	13400.50		1	5
		1	4	February	13425.00		2	5
		1	5	March	12800.00		3	5
	1-		6	Quarter 2			2	2
		1	7	April	15333.40		4	5
		1	8	May	11800.80		5	5
2			9	Count				
3			10					
4			11					
5			12					

D2,D3,E2,E3 中设定计算式后，它们的同位格中也会自动设定计算式。A1 是 0 层行，D2:D8 区域中，所有单元格的计算式都是=A1#，返回它们在所在的 0 层区段中的同位格序号；E2:E8 区域中，所有单元格的计算式都是=A1##，返回它们所在的 0 层区段中的同位格数量。

4.5 用层次定位单元格

- ***A*[*L*]**

  当前格的 *L* 范围内的 *A*，越界返回空，*L* 仅是层。

- ***A***

  相当于 *A*[全网]，这就是通常的单元格调用，直接返回单元格值。

使用 *A* 调用单元格，一定会返回调用的单元格值；而用层次定位单元格，使用 *A*[*L*]的方式调用，可以对 *A* 的范围加以限定，如果 *A* 并不在当前格的 *L* 层范围内，那么将会越界，返回空。

打开计算表 esCalc04_4.gex，通过这个计算表实例，我们来了解一下通常的单元格调用和用层次定位单元格的区别。

0	1	2		A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
	1-		3	California				
		1	4		1990	29760021		
		1	5		2000	33871648		
		1	6		2010	37253956		
	1-		7	Texas				
		1	8		1990	16986510		
		1	9		2000	20851820		
		1	10		2010	25145561		
	1-		11	New York				
		1	12		1990	17990455		
		1	13		2000	18976457		
		1	14		2010	19378102		

计算表中，列出了最近 1990,2000,2010 年三次人口普查各州人口，现在想计算各州人口与前一次数据相比的增长数量。为了比较，在 D14 中设入计算式=C14-C13，在 E14 中设入计算式=C14-C13[A11]，计算表计算后，结果如下：

0	1	2		A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
	1-		3	California				
		1	4		1990	29760021	29760021	29760021
		1	5		2000	33871648	4111627	4111627
		1	6		2010	37253956	3382308	3382308
	1-		7	Texas				
		1	8		1990	16986510	-20267446	16986510
		1	9		2000	20851820	3865310	3865310
		1	10		2010	25145561	4293741	4293741
	1-		11	New York				
		1	12		1990	17990455	-7155106	17990455
		1	13		2000	18976457	986002	986002
		1	14		2010	19378102	401645	401645

由于第 4,5,6,8,9,10,12,13,14 这几行都是同位行，D14 和 E14 设入计算式后，它们的同位格中也会自动设入计算式并自动变迁。如 D13 中计算式为=C13-C12，E13 中计算式为= C13-C12[A11]；D12 中计算式为=C12-C10，E12 中计算式为=C12-C10[A11]；D8 中计算式为=C8-C6，E8 中计算式为=C8-C6[A7]；D4 中计算式为=C4-#REF!，E4 中计算式为=C4-#REF![A3]，等等。

我们来看第 8 行，本来准备计算 Texas 州 1990 年的人口增量。但是，D8 中的计算式是=C8-C6，可见表达式变迁时，会单纯地与“上一个人口数据”做差值，即上个同位行中的数据 C6，即使这个数据不属于 Texas，这样，就变成了与 California 州 2010 年的人口数据作比较了，产生了错误。而在 E8 中，计算式是=C8-C6[A7]，通过层次来定位单元格，限定了数据的范围要在 A7 所在行的区段之内，这样就限定了计算式中的数据必须是 Texas 的数据，避免了变迁时出现的错误。

通过用层次定位单元格，我们在 E 列中可以获得正确的结果。

使用 *A*[*L*]，用层次定位单元格，可以用来处理与上期比较等计算。

4.6 同位格集

由指定单元格在某个区域内的所有同位格组成的集合，称为**同位格集**。

- ***L*{*A*}**

格 *L* 的范围内，*A* 的同位格集。

打开计算表 esCalc04_5.gex，这个计算表中，列出了一些州和城市的信息，我们通过这个实例来了解一下同位格集：

0	1	2	3		A	B	C	D	E
1-				1	States and Cities				
	1-			2	A				
		1-		3		Alaska			
			1	4			Anchorage		
		1-		5		Arizona			
			1	6			Phoenix		
			1	7			Tucson		
	1-			8	C				
		1-		9		California			
			1	10			Los Angeles		
			1	11			San Diego		
			1	12			San Jose		
2				13	=B5{C4}			=B9{C4}	
3				14	=A2{C10}				

计算后，A13 单元格值为["Phoenix","Tucson"]，D13 单元格值为["Los Angeles","San Diego","San Jose"]，A14 单元格值为["Anchorage","Phoenix","Tucson"]。以 A14 中的计算为例，A2 所在的区段是第 2 行到第 7 行的部分，在这个范围内，C10 的同位格是 C4,C6,C7，所以 A14 中的计算结果就是这三个格的格值组成的序列。在计算同位格集 *L*{*A*}时，*A* 可以指定任何一个同位格，不一定要在 *L* 的范围内。

函数 *L*{*A*}中，*L* 可以省略，当 *L* 省略时，自动取为当前格的父行与 *A* 的父行中层次较高者。如果在上面的网格中，D4 中设入计算式={C4}，E4 中设入计算式={B3}，计算后结果如下：

0	1	2	3		A	B	C	D	E
1-				1	States and Cities				
	1-			2	A				
		1-		3		Alaska			
			1	4			Anchorage	["Anchorage"]	["Alaska","Arizona"]
		1-		5		Arizona			
			1	6			Phoenix	["Phoenix","Tucson"]	["Alaska","Arizona"]
			1	7			Tucson	["Phoenix","Tucson"]	["Alaska","Arizona"]
	1-			8	C				
		1-		9		California			
			1	10			Los Angeles	["Anchorage","Phoenix","Tucson"]	["California"]
			1	11			San Diego	["Anchorage","Phoenix","Tucson"]	["California"]
			1	12			San Jose	["Anchorage","Phoenix","Tucson"]	["California"]
2				13	["Phoenix","Tucson"]			["Los Angeles","San Diego","San Jose"]	
3				14	["Anchorage","Phoenix","Tucson"]				

D4 及其同位格中的计算式中，调用的单元格与其层次相等，如={C4}，所以 *L* 自动取为当前格的父行，所以 D4,D6,D7,D10,D11,D12 中的计算结果均为“本州的所有城市”。E4 及其同位格中的计算式，调用的单元格在其父行中，如={B3}，所以 *L* 取为计算式中单元格 *A* 的父行，所以 E4 及其同

位格中的计算结果均为“当前字母列表下的所有州”。

在计算表中，还可以从同位格集中，根据序号偏移量，获取某个同位格。

➢ ***L*[*A*;*x*]**

格 *L* 的范围内，在 *A* 的同位格集中，选出与当前格对应的同位格序号相差 *x* 的 *A* 的同位格。

打开计算表 esCalc04_6.gex，我们通过这个计算表来了解一下 *L*[*A*;*x*]的使用。网格中记录了 2010 年和 2011 年的季度收益，在 C3 中设入计算式=A2[B3;-1]，D3 中设入计算式=B3-A2[B3;-1]，在 E3 中设入计算式=A1[A2;-1][B3;0]，计算后，结果如下：

0	1	2		A	B	C	D	E
1-			1	Quarterly Income				
	1-		2	2010				
		1	3	Quarter 1	50		50	
		1	4	Quarter 2	200	50	150	
		1	5	Quarter 3	400	200	200	
		1	6	Quarter 4	700	400	300	
	1-		7	2011				
		1	8	Quarter 1	800		800	50
		1	9	Quarter 2	950	800	150	200
		1	10	Quarter 3	1000	950	50	400
		1	11	Quarter 4	1250	1000	250	700

C3 中的计算式=A2[B3:-1]，表示同一个 A2 的区段中，B3 的前一个同位格的值，在本例中表示本年上个季度的收益。D3 中的计算式计算的是本季度收益比同年上个季度的增量。再来看 E3 中的计算式=A1[A2:-1][B3:0]，前半部分 A1[A2:-1]，获得的是当前 A2 的上一个同位格，即前一年，后半部分，再从前一年的区段中，获取和当前 B3 同位格序号相同的同位格，在本例中 D3 中的计算式即表示上一年本季度的收益。通过类似 D3 中计算式的写法，可以按照多层的同位格序号来定位单元格。

在计算表中，还可以根据序号的偏移量，在同位格集中获取某个范围内的子集。

➢ ***L*{*A*;*a*:*b*}**

格 *L* 的范围内，在 *A* 的同位格集中，从 *L*[*A*;*a*]起，到 *L*[*A*;*b*]为止的所有 *A* 的同位格组成的序列。如果 *a* 省略则从 *L* 范围内第一个 *A* 的同位格开始，如果 *b* 省略则至 *L* 范围内最后一个 *A* 的同位格为止。

我们修改刚刚使用的计算表，在 C3 中设入计算式=A2{B3;:0}.sum()，在 D3 中设入计算式=A2{B3;:-1}.sum()，在 E3 中设入计算式=A1{B3;1}.count()，计算后，结果如下：

0	1	2		A	B	C	D	E
1-			1	Quarterly Income				
	1-		2	2010				
		1	3	Quarter 1	50	50		7
		1	4	Quarter 2	200	250	50	6
		1	5	Quarter 3	400	650	250	5
		1	6	Quarter 4	700	1350	650	4
	1-		7	2011				
		1	8	Quarter 1	800	800		3
		1	9	Quarter 2	950	1750	800	2
		1	10	Quarter 3	1000	2750	1750	1
		1	11	Quarter 4	1250	4000	2750	0

C3 中计算本年内的累加收益，D3 中计算本年度截至上个季度的累计收益，E3 中计算剩余同位格总数。

使用 *L*[*A*;*x*],*L*{*A*;*a*:*b*}这两个函数，我们可以方便地计算同期比，累加之类的计算。

通过对同位格集使用序列的各种函数进行运算，可以完成各类统计计算，如 *A*.sum()，*A*.avg()，*A*.max()等等

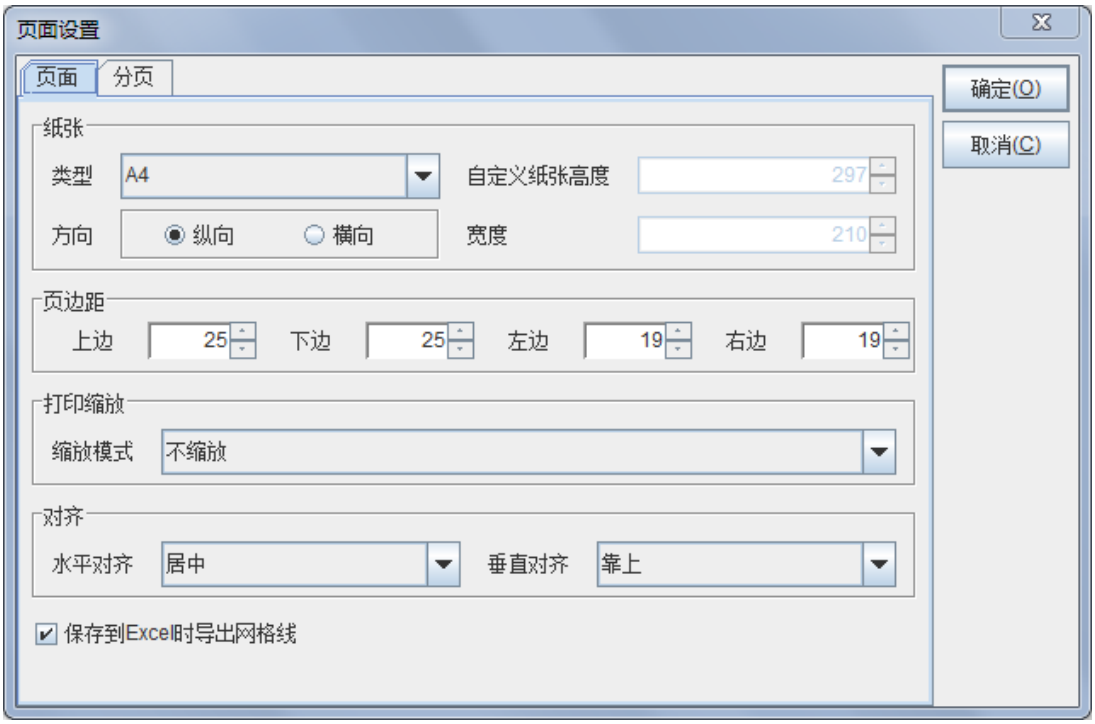
4.7 页面设定与分页后计算

在计算表中，计算表文件计算后，可以进行分页操作，以便于打印输出。

4.7.1 打印与分页

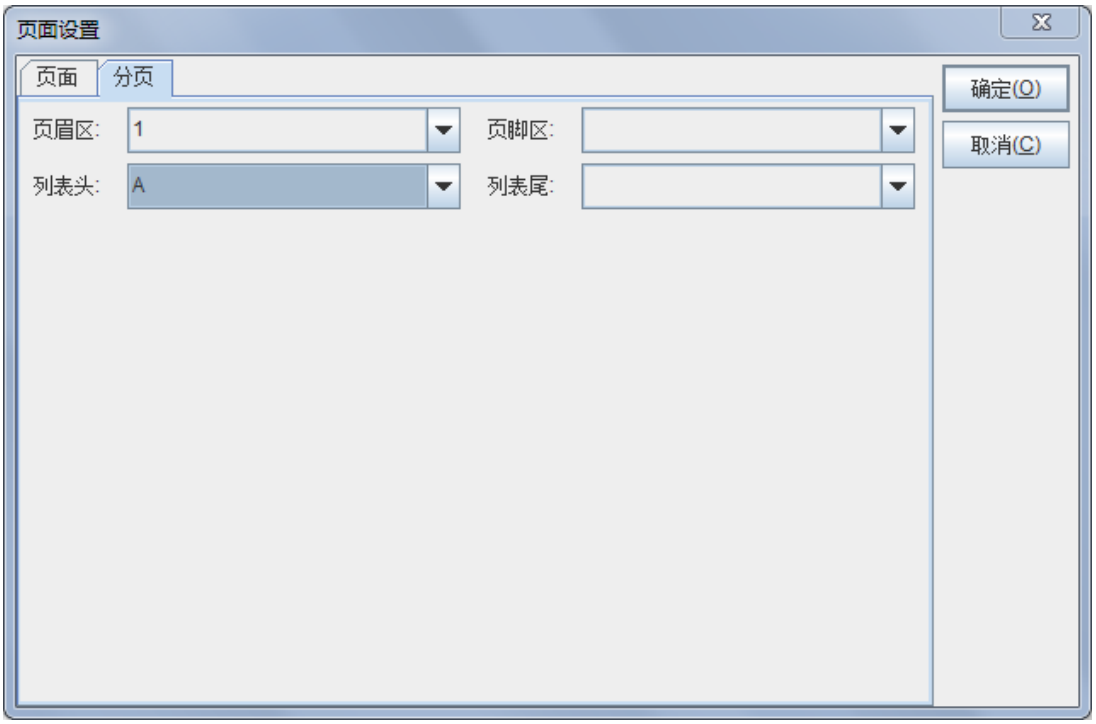
计算表中的计算表文件，可以进行打印输出。在菜单栏中点击**文件→打印**，或者使用快捷键 Ctrl+P，就可以弹出打印预览界面。计算表在打印时，会进行分页。

在菜单栏中点击**文件→页面设置**会弹出页面设置窗口：



在**页面**页，可以进行纸张设定，页边设定，打印缩放设定以及打印位置设定等。

在纸张设定中，可以选择纸张型号，或者将型号选为自定义自行设定纸张宽度和高度（单位是毫米）；还可以选择纸张方向是横向还是纵向。



在分页设定页面中，可以选择分页时进行重复的表头行/列和表尾行/列。表头行只能是从第一行起连续的 0 层行，表尾行只能是从最后一行起连续的 0 层行。

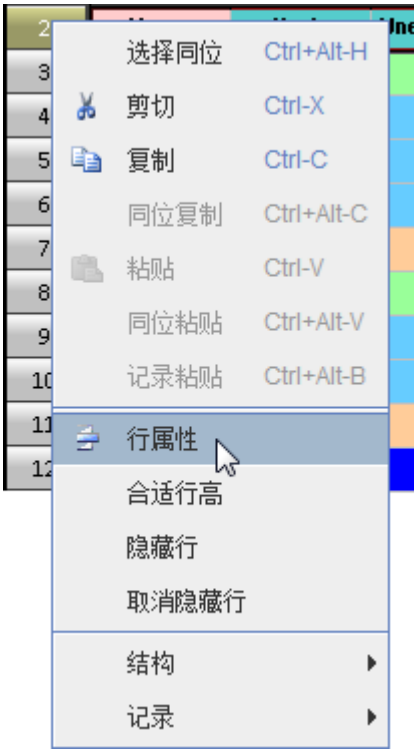
计算表在分页时，会根据纸张设定，自动计算每一页能容纳的行列数，进行分页。在分页时，表头区域设定中的行和列会在每页重复。分页时先纵向分页，再横向分页。

4.7.2 行后分页与列后分页

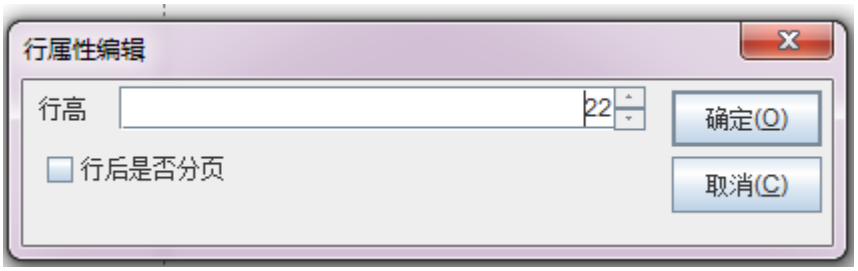
在计算表中，除了按照纸张设定自动进行分页之外，还可以设置行后分页或者列后分页。

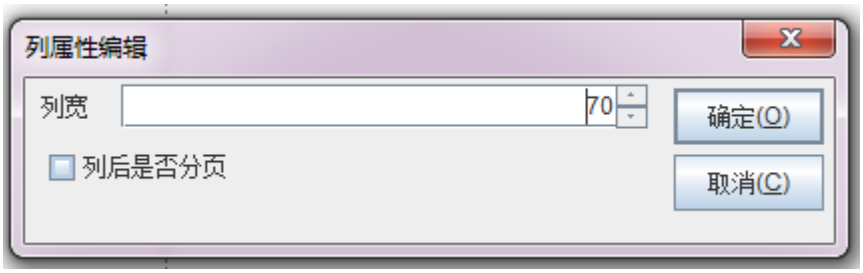
如果网格中的某一行设定了行后分页，那么将从这一行的后面强制进行纵向分页；如果网格中的某一列设定了列后分页，那么将从这一列的后面强制进行横向分页。

设置行后分页或者列后分页时，在行号所在的行首格或者列号所在的列首格中，按下鼠标右键，在弹出的菜单中选择行属性/列属性，编辑行信息或者列信息：



在弹出的行属性或者列属性的设定面板中，勾选行后分页或者列后分页即可：





4.7.3 分页后计算

为了便于分页时处理，计算表中也提供了几个分页后进行计算的函数以供使用。

- **pgno()**

当前页号。

- **pgall()**

  总页数。

- **pgcell(*C*)**

  本页所有与 *C* 同位的单元格值序列。

在 **2.5 立即计算格与联动计算格**中提到过，在自动计算模式下，立即计算格将在表达式输入时自动进行计算，之后就不会再重新计算了。对于上面这些需要分页后计算的函数。就会使得计算结果出现错误。因此，**分页后计算函数通常会使用在联动计算格中**。

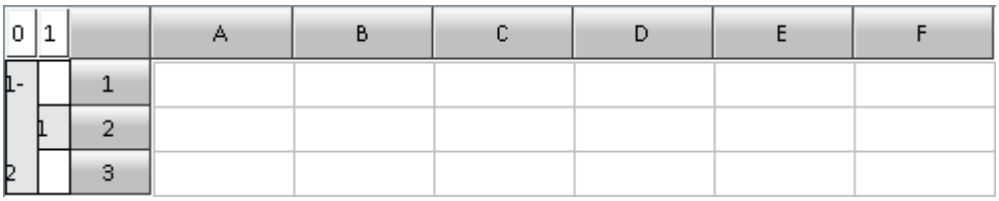
5. 编辑区段的结构

在计算表中，区段是网格的基本结构，对区段的编辑操作，也是计算表中的基本操作。对区段的编辑，首先要进行的就是确定一个区段的结构。

5.1 区段的结构

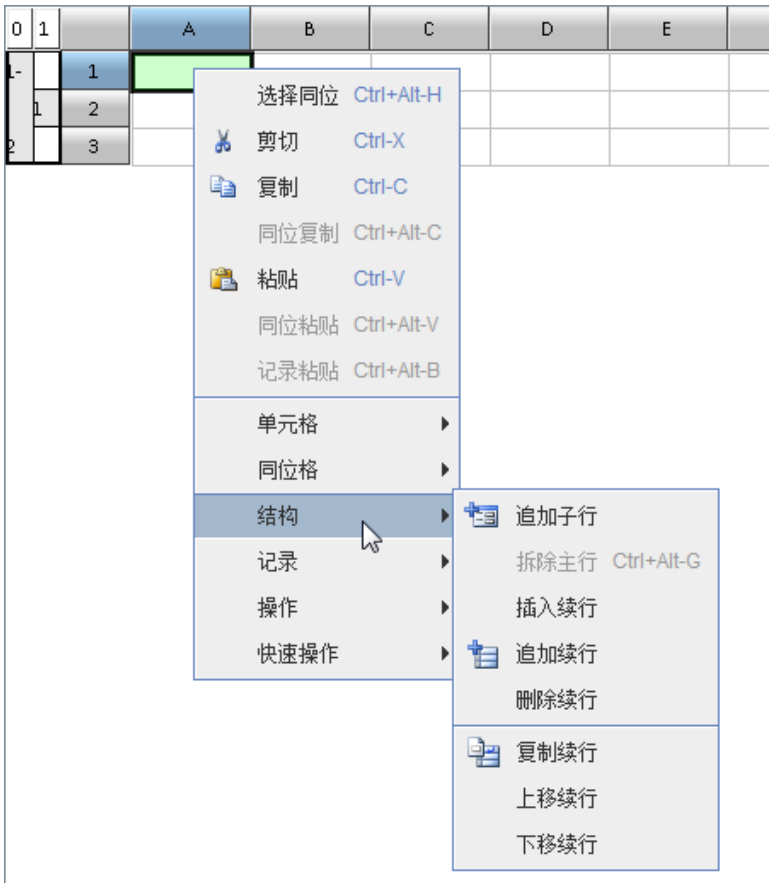
区段是由主行，续行以及它们的子行所构成，编辑它的结构，就是对主行，续行以及子行的各种操作。

在计算表中，一个新建的计算表文件如下图所示：



新的计算表文件中，第 1 行和第 3 行是 0 层行，第 2 行是 1 层行，是第 1 行的子行。

如果需要编辑计算表中，某一行区段的结构，在这一行的任何一个单元格或者行首格上，单击右键，在右键菜单中的**结构**项目中，可以选择对区段结构的各种操作。



也可以在选中单元格后，在菜单栏中点击**结构**选项，选择需要的操作。

由于同位行必须是同构的，因此，在改变某一行的结构时，如果这一行拥有同位行，那么它所有的同位行会一同执行同样的改变。

5.2 追加子行

为当前行添加一个子行，如果当前行已经拥有子行，那么原来的子行会下降一层，变为新加子行的子行。

在右键菜单中点击**结构→追加子行**，可以添加子行。

打开计算表 esCalc05_1.gex，网格如下：

0	1	2		A	B	C	D	E
1-			1	Temperature in F (2012)				
2-			2	Date	Day of Week	Max Temp.	Min Temp.	
	1-		3	January				
		1	4	week 4				
		1	5	week 5				
	1-		6	February				
		1	7	week 1				
3			8					

如果在选择第 4 行，执行追加子行，那么除了会在第 4 行之后添加一个子行之外，它的子行第 5 行和第 7 行也会添加一个新的子行，执行后结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5					
		1-		6	week 5				
			1	7					
	1-			8	February				
		1-		9	week 1				
			1	10					
3				11					

从结果中可以看到，追加的子行中，所有单元格都是初始的空白单元格。

如果在本节最初的网格中，选择第 3 行，执行追加子行，结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4					
			1	5	week 4				
			1	6	week 5				
	1-			7	February				
		1-		8					
			1	9	week 1				
3				10					

从结果中可以看到，由于第 3 行和它的同位行第 6 行本来拥有子行，执行追加子行后，它们原本的子行层次降低了一层，变为了新增子行的子行。

5.3 拆除主行

删除没有续行但有子行的主行，当前行的子行将上升成其父行的子行。

在右键菜单中点击**结构→拆除主行**，可以拆除主行。

再次打开计算表 esCalc05_1.gex，网格如下：

0	1	2		A	B	C	D	E
1-			1	Temperature in F (2012)				
2-			2	Date	Day of Week	Max Temp.	Min Temp.	
	1-		3	January				
		1	4	week 4				
		1	5	week 5				
	1-		6	February				
		1	7	week 1				
3			8					

只有某一行能够执行拆除主行的操作，在右键菜单中才会允许选择拆除主行的选项。能够拆除主行，就要求这一行是主行，而且拥有子行和父行，如上面的网格中，第 1,3,4,5,6,7 这几行是主行，而第 1 行没有父行，第 4,5,7 行没有子行，因此只有第 3 行和第 6 行可以执行拆除子行的操作。

如选中第 3 行，执行拆除子行，结果如下：

0	1		A	B	C	D	E
1		1	Temperature in F (2012)				
2-		2	Date	Day of Week	Max Temp.	Min Temp.	
	1	3	week 4				
	1	4	week 5				
	1	5	week 1				
3		6					

可以看到，原网格中的第 3 行和它的同位行第 6 行都被移除，它们的子行层次上升一层，变成了它们主行第 2 行的子行。

5.4 续行的添加和删除

5.4.1 插入续行

在当前行之前添加当前行的续行。

在右键菜单中点击**结构→插入续行**，可以插入续行。

打开计算表 esCalc05_2.gex，网格如下：

0	1	2		A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
	1-		3	California				
		1	4		1990	29760021		
		1	5		2000	33871648		
		1	6		2010	37253956		
	1-		7	Texas				
		1	8		1990	16986510		
		1	9		2000	20851820		
		1	10		2010	25145561		

如果选择第 7 行，插入续行，结果如下：

0	1	2		A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
	1		3					
	2-		4	California				
		1	5		1990	29760021		
		1	6		2000	33871648		
		1	7		2010	37253956		
	1		8					
	2-		9	Texas				
		1	10		1990	16986510		
		1	11		2000	20851820		
		1	12		2010	25145561		

可以看到，原网格中的第 7 行和它的同位行第 3 行之前，都被插入了一个新的续行，续行与原来的第 3 行和第 7 行，具有相同的层次，而且同一列中的单元格具有相同的前景色背景色等属性。

当同时选中多个连续的同层续行（主行也是一种续行）时，再执行插入续行，会根据选中的行数插入多个续行。

5.4.2 追加续行

在当前行之后，当前行的所有属行之前添加当前行的续行，当前行的所有属行都会转变为新续行的属行。

在右键菜单中点击**结构→追加续行**，可以添加续行。

重新打开计算表 esCalc05_2.gex，选择第 7 行，追加续行，执行后结果如下：

0	1	2		A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
	1		3	California				
	2-		4					
		1	5		1990	29760021		
		1	6		2000	33871648		
		1	7		2010	37253956		
	1		8	Texas				
	2-		9					
		1	10		1990	16986510		
		1	11		2000	20851820		
		1	12		2010	25145561		

可以看到，原网格中的第 7 行和它的同位行第 3 行之后，都被插入了一个新的续行，续行与原来的第 3 行和第 7 行，具有相同的层次，而且同一列中的单元格具有相同的前景色背景色等属性。与执行插入续行时不同，原来第 3 行和第 6 行各自的所有子行，都被移到了新的续行中。

当同时选中多个连续的同层续行（主行也是一种续行）时，再执行追加续行，会根据选中的行数添加多个续行。

5.4.3 删除续行

将续行删除，包括其属行；主行删除后下一个续行接替为主行

在右键菜单中点击**结构→删除续行**，可以删除续行。

使用 **5.4.2 追加续行**中执行追加续行后的实例，或者直接打开计算表 esCalc05_3.gex，我们选择第 3 行，执行删除续行，执行后结果如下：

0	1	2		A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
	1-		3					
		1	4		1990	29760021		
		1	5		2000	33871648		
		1	6		2010	37253956		
	1-		7					
		1	8		1990	16986510		
		1	9		2000	20851820		
		1	10		2010	25145561		

可以看到，第 3 行和它的同位行第 8 行都被删除，由于它们是主行，在它们之后的续行将接替为主行。实际上，如果某一个续行被删除，那么在它所在的与它同层的区段中，排在它后面的续行的序号都会依次上升。如本例中，原本的第 4 行和第 9 行，它们的序号就由 2 上升为了 1.

重新打开计算表 esCalc05_3.gex，如果不是选择第 3 行，而是选择第 4 行，执行删除续行，执行后结果如下：

0	1			A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
		1	3	California				
		1	4	Texas				

可以看到，当第 4 行和它的同位行第 9 行被删除时，由于它们是父行，因此它们的子行也被删除了。

重新打开计算表 esCalc05_3.gex，如果我们不是选择第 4 行或者第 3 行，而是选择了第 6 行执行删除续行，执行后结果如下：

0	1			A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
		1	3	California				
		2	4					
		1	5	Texas				
		2	6					

从结果中可以看到，在原来的网格中，第 6 行和它的同位行第 5,7,10,11,12 行，全都被删除了。由于这些行都是主行，但是没有续行，因此它们的父行的所有子行都被删除。

执行删除续行时，如果在同一个区段中，选择了多个续行（可能包含主行或者它们的属行），那么会同时删除多个续行。

5.4.4 复制续行

复制当前行及其属行，并添加到当前行之前，作为当前行的续行。

在右键菜单中点击**结构→复制续行**，可以复制续行。

我们打开计算表 esCalc05_2.gex，网格如下：

0	1	2		A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
	1-		3	California				
		1	4		1990	29760021		
		1	5		2000	33871648		
		1	6		2010	37253956		
	1-		7	Texas				
		1	8		1990	16986510		
		1	9		2000	20851820		
		1	10		2010	25145561		

如果选择第 7 行，复制续行，结果如下：

0	1	2		A	B	C	D	E
1			1	population statistics				
2-			2	State	Year	Population		
	1-		3	California				
		1	4		1990	29760021		
		1	5		2000	33871648		
		1	6		2010	37253956		
	2-		7	California				
		1	8		1990	29760021		
		1	9		2000	33871648		
		1	10		2010	37253956		
	1-		11	Texas				
		1	12		1990	16986510		
		1	13		2000	20851820		
		1	14		2010	25145561		
	2-		15	Texas				
		1	16		1990	16986510		
		1	17		2000	20851820		
		1	18		2010	25145561		

可以看到，原网格中的第 7 行和它的同位行第 3 行，都被复制为了新的续行，而在复制时它们的子行也被一同复制。

5.5 移动续行

5.5.1 上移续行

上移续行和它的属行，但是不得跨越它父行的区段。

在右键菜单中点击**结构→上移续行**，可以上移续行。

打开计算表 esCalc03_1.gex，网格如下：

0	1	2		A	B	C	D	E
1-			1	Employee				
	1		2	Department	Finance			
	2-		3	Female				
		1	4	Ashley Wilson	NY	11000		
	3-		5	Male				
		1	6	Daniel Davis	FL	10000		
	4		7	Count	2	Sum	21000	
	1		8	Department	R&D			
	2		9	Female				
	3-		10	Male				
		1	11	Justin Smith	TX	7000		
		1	12	Jacob Davis	TX	16000		
	4		13	Count	2	Sum	23000	
2			14	Count	4	Sum	44000	

如果选择第 5 行，执行上移续行，结果如下：

0	1	2		A	B	C	D	E
1-			1	Employee				
	1		2	Department	Finance			
	2-		3	Male				
		1	4	Daniel Davis	FL	10000		
	3-		5	Female				
		1	6	Ashley Wilson	NY	11000		
	4		7	Count	2	Sum	21000	
	1		8	Department	R&D			
	2-		9	Male				
		1	10	Justin Smith	TX	7000		
		1	11	Jacob Davis	TX	16000		
	3		12	Female				
	4		13	Count	2	Sum	23000	
2			14	Count	4	Sum	44000	

从结果中可以看到，第 5 行和它的子行，在区段中上移了一位，序号由 3 变为了 2。同时它的同位行第 10 行及其子行，也在区段中上移了一位。执行上移续行后，续行的序号会减小，因此主行是无法执行上移续行操作的。

5.5.2 下移续行

下移续行和它的属行，不得跨越它父行的区段。

在右键菜单中点击**结构→下移续行**，可以下移续行。

我们重新打开计算表 esCalc03_1.gex，选择第 2 行，下移续行，执行后结果如下：

0	1	2		A	B	C	D	E
1-			1	Employee				
	1-		2	Female				
		1	3	Ashley Wilson	NY	11000		
	2		4	Department	Finance			
	3-		5	Male				
		1	6	Daniel Davis	FL	10000		
	4		7	Count	2	Sum	21000	
	1		8	Female				
	2		9	Department	R&D			
	3-		10	Male				
		1	11	Justin Smith	TX	7000		
		1	12	Jacob Davis	TX	16000		
	4		13	Count	2	Sum	23000	
2			14	Count	4	Sum	44000	

从结果中可以看到，第 2 行和它的同位行，在各自的区段中下移了一位，序号由 1 变为了 2。执行下移续行后，续行的序号会增大，因此区段中最后一个续行是无法执行下移续行操作的。

如果不是选择第 2 行，而是选择第 3 行下移续行的话，则会得到和 **5.5.1 上移续行**中，对第 5 行执行上移续行之后完全相同的结果。

5.6 行的操作

在网格中点击行首格，可以选择整行。如果在行首格中点击右键，可以在右键菜单中，执行各种对行的操作，如下图：

编辑行属性，还可以在选定行之后，在菜单栏中点击**编辑→行高列宽**，在其中选择所需操作。

5.6.1 隐藏行

隐藏某些续行。

在行首格中，点击右键，在右键菜单中选择隐藏行，即可隐藏所选择的续行。

打开计算表 esCalc05_4.gex，如下：

0	1	2		A	B	C	D	E
1			1	Sales Information				
2-			2	Quarter	Month	Replenishment	sell	Stock
	1-		3	Quarter 1				
		1	4		January	4000	3400	600
		1	5		February	4000	5270	-670
		1	6		March	4000	3250	80
	2		7	Quarterly total		12000	11920	
	1-		8	Quarter 2				
		1	9		April	5000	5050	30
		1	10		May	5000	4880	150
	2		11	Quarterly total		10000	9930	
3			12	Total		22000	21850	

在这个计算表中，记录了前 5 个月某产品的进货量和销售量。在 C7 中的计算式为==\{C4\}.sum()，C7 和 C11 的作用是统计本季度的总进货量；在 D7 中的计算式为==\{D4\}.sum()，D7 和 D11 用来计算本季度的总销售量；C12 中的计算式为==\{C4\}.sum()，计算总进货量，D12 中的计算式为==\{D4\}.sum()，计算总销售量。在 E10 中的计算式为==E9+C10-D10，它和它的同位格用来计算每月的库存剩余量。

在这个网格中，选择第 7 行，在右键菜单中点击隐藏行，执行隐藏续行，结果如下：

0	1	2		A	B	C	D	E
1			1	Sales Information				
2-			2	Quarter	Month	Replenishment	sell	Stock
	1-		3	Quarter 1				
		1	4		January	4000	3400	600
		1	5		February	4000	5270	-670
		1	6		March	4000	3250	80
	1-		8	Quarter 2				
		1	9		April	5000	5050	30
		1	10		May	5000	4880	150
3			12	Total		22000	21850	

从结果中可以发现，第 7 行和第 11 行被隐藏了。隐藏续行时，所选中的行的所有同位行会同时被隐藏，续行本身并没有消失，各行的行号不变。

如果此时再选择第 8 行，隐藏续行，那么结果如下：

0	1	2		A	B	C	D	E
1			1	Sales Information				
2-			2	Quarter	Month	Replenishment	sell	Stock
		1	4		January	4000	3400	600
		1	5		February	4000	5270	-670
		1	6		March	4000	3250	80
		1	9		April	5000	5050	30
		1	10		May	5000	4880	150
3			12	Total		22000	21850	

可以发现，第 3 行和它的同位行第 8 行被隐藏，但是它们的所有属行都保持原有状态，未被隐藏。

再选中第 4 行，隐藏续行，则结果如下：

0	1	2		A	B	C	D	E
1			1	Sales Information				
2-			2	Quarter	Month	Replenishment	sell	Stock
3			12	Total		22000	21850	

可以发现，第 4 行和它的同位行第 5,6,9,10 行被隐藏。续行被隐藏之后，在展示、打印或者导出时都会被隐藏，但是其中的数据仍然存在，可以在表达式中被调用。

5.6.2 取消隐藏行

显示隐藏的续行。

选中连续的若干行，点击右键，在右键菜单中选择取消隐藏行，即可在选中行号范围内，显示所有被隐藏的续行。

我们使用 **5.6.1 隐藏续行**中设置了隐藏续行后的最终网格，或者直接打开计算表 esCalc05_5.gex，同时选择第 2 行和第 12 行，执行取消隐藏行，结果如下：

0	1	2		A	B	C	D	E
1			1	Sales Information				
2-			2	Quarter	Month	Replenishment	sell	Stock
	1-		3	Quarter 1				
		1	4		January	4000	3400	600
		1	5		February	4000	5270	-670
		1	6		March	4000	3250	80
	2		7	Quarterly total		12000	11920	
	1-		8	Quarter 2				
		1	9		April	5000	5050	30
		1	10		May	5000	4880	150
	2		11	Quarterly total		10000	9930	
3			12	Total		22000	21850	

可以看到，第 2 行到第 12 行之间的所有行都被显示了出来。执行取消隐藏行时，如果某一行被取消了隐藏，那么它的同位行也将被取消隐藏。

5.6.3 行属性

设置行属性，包括行高和是否行后分页。

正如在 **4.7.2 行后分页与列后分页**中提到的，设置行属性时，首先选中行，在右键菜单中选择行属性，即可在弹出的行属性面板中查看或修改行属性。

当某一行的行属性被修改时，它的所有同位行的行属性会被一同修改。

5.6.4 合适行高

自动设置所选择行的行高，使得这一行的行高刚好容纳本行中的文字。

选中行首格，点击右键，在右键菜单中选择合适行高，即可将当前行的行高设为最适合值。

当所选行的行高改变时，它的所有同位行也会同时被修改，行高设为当前行的最适合行高。如果所选行中的单元格都是空格，那么执行合适行高操作后，行高会保持原值不变，而不会被设为 0。

5.7 单元格的复制与粘贴

选择同区段内同层连续单元格，可以复制表达式及属性粘贴到目标格。

在粘贴时，单元格的属性，如前景色，背景色，对齐属性，字体等，也会被一同粘贴。

如果单元格中存在表达式，在复制单元格时，将按照表达式中，单元格的位置及意义，智能变迁。例如，打开计算表 esCalc05_6.gex，如下：

	1	2		A	B	C	D	E
1			1	Sales Information				
2-			2	Quarter	Month	Replenishment	sell	Stock
	1-		3	Quarter 1				
		1	4		January	4000	3400	600
		1	5		February	4000	5270	-670
		1	6		March	4000	3250	80
	2		7	Quarterly total		12000		
	1-		8	Quarter 2				
		1	9		April	5000	5050	30
		1	10		May	5000	4880	150
	2		11	Quarterly total		10000		
3			12	Total				

其中，C7 中的计算式为==\{C4\}.sum()，即计算位于同一列的 C4 及其所有同位格的和，在这里计算第 1 季度的总进货量。它的同位格 C11 中表达式字段迁移为==\{C9\}.sum()，计算第 2 季度的总进货量。

如果选择 C7，按 Ctrl+C 复制，然后选择 D11，按 Ctrl+V 粘贴，则网格如下：

	1	2		A	B	C	D	E
1			1	Sales Information				
2-			2	Quarter	Month	Replenishment	sell	Stock
	1-		3	Quarter 1				
		1	4		January	4000	3400	600
		1	5		February	4000	5270	-670
		1	6		March	4000	3250	80
	2		7	Quarterly total		12000	11920	
	1-		8	Quarter 2				
		1	9		April	5000	5050	30
		1	10		May	5000	4880	150
	2		11	Quarterly total		10000	9930	
3			12	Total				

首先，可以看到，C7 的外观属性被复制到了 D11 中，查看 D11 中的表达式，是==\{D9\}.sum()，原来表达式中的 C4，粘贴后被修改为与目标格在同一列，且位于同一区段中的单元格 D9。可见，在复制粘贴表达式时，表达式中引用的单元格会迁移处理：在列的方向按照源格和目标格的位置变化相应改变；在行的方向上的改变则与同位格中表达式的自动迁移类似。

如果复制 C7 格，将其粘贴到 C12 及 D12，则网格如下：

	1	2		A	B	C	D	E
1			1	Sales Information				
2-			2	Quarter	Month	Replenishment	sell	Stock
	1-		3	Quarter 1				
		1	4		January	4000	3400	600
		1	5		February	4000	5270	-670
		1	6		March	4000	3250	80
	2		7	Quarterly total		12000	11920	
	1-		8	Quarter 2				
		1	9		April	5000	5050	30
		1	10		May	5000	4880	150
	2		11	Quarterly total		10000	9930	
3			12	Total		22000	21850	

C12 中的表达式为==\{C4\}.sum()，D12 中的表达式为==\{D4\}.sum()。

5.8 列的操作

在网格中点击列首格，可以选择整列。如果在列首格中点击右键，可以在右键菜单中，选择执行各种对列的操作，如下图：

5.8.1 列的插入与删除

➢ **列的删除**

在计算表中删除列，只需要点击列首格选择整列，然后在鼠标的右键菜单中选择删除列，或者在菜单栏中点击**结构→删除列**，即可删除选中列。同时选中多列时，会同时删除。

➢ **列的插入**

在计算表中插入列，只需要点击列首格选择整列，然后在鼠标的右键菜单中选择插入列，或者在菜单栏中点击**结构→插入列**，即可在选中列之前插入一列，新列中各行单元格的属性与选中列相同。同时选中多列时，会同时插入多列。

➢ **列的追加**

在计算表中追加列，只需要点击列首格选择整列，然后在鼠标的右键菜单中选择追加列，或者在菜单栏中点击**结构→追加列**，即可在所有列之后新加入一列，新列中各行单元格的属性与选中列相同。同时选中多列时，会同时追加多列。

5.8.2 隐藏/显示列

➢ **隐藏/显示列**

在计算表中，可以选择将某些列隐藏。点击列首格选择整列，然后在鼠标的右键菜单中，选择隐藏。

与隐藏的续行类似，隐藏的列在展示，打印或者导出时都不会显示，但是在计算中可以被调用。

5.8.3 列属性

设置列属性，包括列宽和是否列后分页。

在 **4.7.2 行后分页与列后分页**中已经提到，设置列属性时，首先选中列，在右键菜单中选择列属性，即可在弹出的列属性面板中查看或修改列属性。

5.8.4 合适列宽

自动设置所选择列的列宽，使得这一列的列宽刚好容纳本列中最宽的文字。

选中列首格，点击右键，在右键菜单中选择合适列宽，即可将当前列的列宽设为最适合值。

如果所选列中的单元格都是空格，那么执行合适列宽操作后，列宽会保持原值不变，而不会被设为 0。

5.9 导入与导出

5.9.1 导入常见格式文件

计算表中，可以使用文本文件、Excel 文件中的数据。在菜单栏中点击**文件→打开其它格式**，即可选择多种格式文件进行导入。

下面以文本文件来简单说明：

```
ID      Name      Gender   Age
1       Emily     F        17
2       Elizabeth F        16
3       Sean      M        17
4       Lauren    F        15
```

如果选择带标题 Txt，文本中的第一行将作为标题至于 0 层行中，作为其它同位行的主行，打开后的网格如下：

0	1		A	B	C	D
1-		1	ID	Name	Gender	Age
	1	2	1	Emily	F	17
	1	3	2	Elizabeth	F	16
	1	4	3	Sean	M	17
	1	5	4	Lauren	F	15

如果选择不带标题 Txt，文本中的所有行都会被置于同位行中，打开后的网格如下：

0	1		A	B	C	D
1-		1				
	1	2	ID	Name	Gender	Age
	1	3	1	Emily	F	17
	1	4	2	Elizabeth	F	16
	1	5	3	Sean	M	17
	1	6	4	Lauren	F	15

5.9.2 导出为常见格式文件

计算表中的网格数据，可以输出为其它格式的文件，如文本文件，Excel 文件，Html 文件等。

在菜单栏中点击**文件→另存为其它格式**，即可选择将网格数据导出。

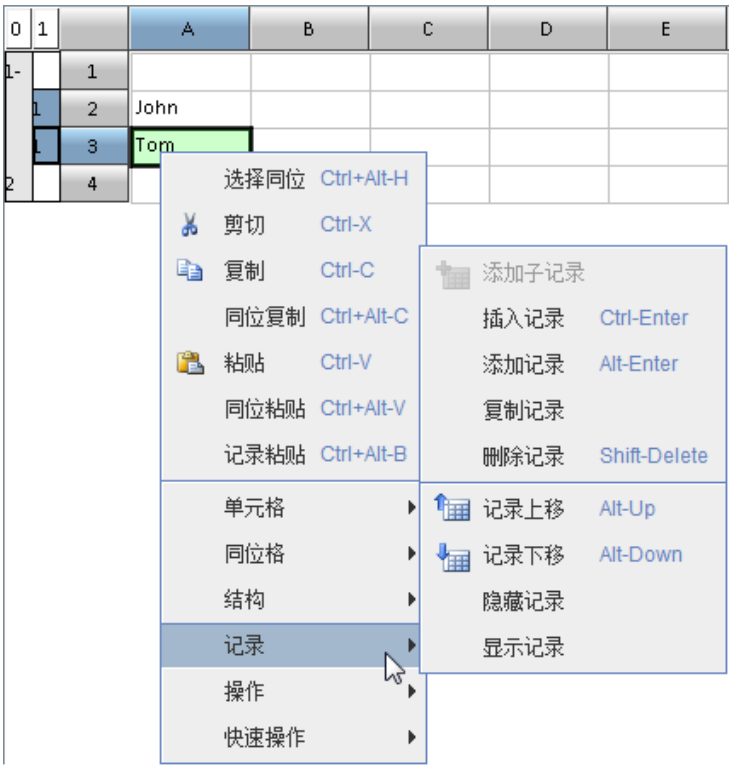
在将网格文件导出时，网格中所有的层次信息都会丢失，只会按照单元格的位置进行导出，文本文件中每列数据之间用 Tab 分隔，每行数据之间用回车分隔。

6. 编辑记录

6.1 记录和记录区段

在计算表中，区段的结构决定了之后，为了将数据填入区段，需要在区段中对记录进行编辑。

如果需要在某个区段中来对记录进行添加，删除等操作，可以在区段的主行或者区段中某条记录行的任何一个单元格或者行首格上，单击右键，在右键菜单中的记录项目，可以选择对区段中记录的各种操作。



也可以在选中相应的单元格后，在菜单栏中点击数据选项，选择需要的操作。

操作区段中的记录时，不会改变区段的结构。如果某行没有在结构中添加子行，是无法直接添加记录的；而即使把某个主行的所有记录属行全部删除，这个主行的子行的结构仍然存在。

6.2 添加子记录

为某个续行添加一个空子记录。

在右键菜单中点击**记录→添加子记录**，可以添加子记录。

打开计算表 esCalc06_1.gex，网格如下：

0	1	2	3		A	B	C	D	E
1-				1	States and Cities				
	1-			2	A				
		1-		3		Alaska			
			1	4			Anchorage		
		1		5		Arizona			
	1-			6	C				
		1-		7		California			
			1	8			Los Angeles		
			1	9			San Diego		
			1	10			San Jose		
	1			11	D				
2				12					

如果某行的结构中没有子行，那么是无法添加子记录的，如第 4,8,9,10 行，以及第 12 行。如果想为这些行添加子记录，首先要编辑这些行的结构，进行添加子行操作。

如果选择第 5 行和第 3 行，分别执行添加子记录操作，添加子记录后，结果如下：

0	1	2	3		A	B	C	D	E
1-				1	States and Cities				
	1-			2	A				
		1-		3		Alaska			
			1	4					
			1	5			Anchorage		
		1-		6		Arizona			
			1	7					
	1-			8	C				
		1-		9		California			
			1	10			Los Angeles		
			1	11			San Diego		
			1	12			San Jose		
	1			13	D				
2				14					

原网格中第 5 行是 Arizona 州，第 3 行是 Alaska 州。从结果可以看到，执行添加子记录后，分别为这两个州增加了空白的城市记录，Arizona 州中原本没有城市记录，会添加一个子行记录；Alaska 州已经拥有了 1 条城市记录，会在已有记录之前增加一个子行记录。新增的记录行中，每个单元格的前景色背景色等等属性由区段已经设定的结构决定。

需要注意的是，虽然 Arizona 州下并没有城市记录，但是在它的区段结构中存在子记录的结构。

如果计算表 esCalc06_1.gex 中，选择第 11 行，执行添加子记录，又会如何呢？结果如下：

0	1	2	3		A	B	C	D	E
1-				1	States and Cities				
	1-			2	A				
		1-		3		Alaska			
			1	4			Anchorage		
		1		5		Arizona			
	1-				C				
		1-				California			
			1	8			Los Angeles		
			1	9			San Diego		
			1	10			San Jose		
	1-			11	D				
		1-		12					
			1	13					
2				14					

从结果可以看到，第 11 行是首字母 D，在它的下面原本没有州的记录，执行添加子记录后，会添加一个新的州记录；由于区段结构中，州记录中还存在着城市的子行，因此在添加州记录的同时，也会为新的州记录添加一个新的城市记录。

6.3 记录的添加与删除

6.3.1 插入记录

在当前区段之前添加当前区段的空白同位区段。

在右键菜单中点击**记录→插入记录**，可以插入记录。

打开计算表 esCalc06_2.gex，网格如下：

0	1	2		A	B	C	D	E
1			1	Employees				
2-			2	Eld	Name	State	Gender	
	1-		3	Finance				
		1	4	1	Rebecca Moore	Arizona	Female	
		1	5	6	Matthew Johnson	Washington	Male	
	1-		6	R&D				
		1	7	7	Alexis Smith	California	Female	
		1	8	8	Megan Wilson	New York	Female	
		1	9	9	Victoria Davis	Tennessee	Female	
	1-		10	Sales				
		1	11	19	Samantha Williams	Alabama	Female	
3			12					

如果分别选择第 7 行和第 5 行，执行插入记录操作，插入记录后，结果如下：

0	1	2		A	B	C	D	E
1			1	Employees				
2-			2	EId	Name	State	Gender	
	1-		3	Finance				
		1	4	1	Rebecca Moore	Arizona	Female	
		1	5					
		1	6	6	Matthew Johnson	Washington	Male	
	1-		7	R&D				
		1	8					
		1	9	7	Alexis Smith	California	Female	
		1	10	8	Megan Wilson	New York	Female	
		1	11	9	Victoria Davis	Tennessee	Female	
	1-		12	Sales				
		1	13	19	Samantha Williams	Alabama	Female	
3			14					

可以看到，原网格中的第 7 行和第 5 行之前，都被插入了一个新的同位区段。

重新打开 esCalc06_2.gex，如果选择第 6 行，执行插入记录，又会如何呢？结果如下：

0	1	2		A	B	C	D	E
1			1	Employees				
2-			2	EId	Name	State	Gender	
	1-		3	Finance				
		1	4	1	Rebecca Moore	Arizona	Female	
		1	5	6	Matthew Johnson	Washington	Male	
	1-		6					
		1	7					
	1-		8	R&D				
		1	9	7	Alexis Smith	California	Female	
		1	10	8	Megan Wilson	New York	Female	
		1	11	9	Victoria Davis	Tennessee	Female	
	1-		12	Sales				
		1	13	19	Samantha Williams	Alabama	Female	
3			14					

从结果可以看到，原网格中的第 7 行是部门区段，执行插入记录后，会在它之前插入一个新的部门记录；由于区段结构中，部门记录中还存在着员工记录的子行，因此在添加部门记录的同时，也会按照区段结构为新的部门记录添加一个新的员工记录。

当同时选中多个连续的同层区段时，再执行插入记录，会根据选中的区段总数插入多个空白记录。

插入记录可以使用快捷键 Ctrl+Enter 执行。

6.3.2 添加记录

在当前区段之后添加当前区段的空白同位区段。

在右键菜单中点击**记录→添加记录**，可以添加记录。

重新打开计算表 esCalc06_2.gex，分别选择第 9 行和第 3 行，添加记录，执行后结果如下：

0	1	2		A	B	C	D	E
1			1	Employees				
2-			2	EId	Name	State	Gender	
	1-		3	Finance				
		1	4	1	Rebecca Moore	Arizona	Female	
		1	5	6	Matthew Johnson	Washington	Male	
	1-		6					
		1	7					
	1-		8	R&D				
		1	9	7	Alexis Smith	California	Female	
		1	10	8	Megan Wilson	New York	Female	
		1	11	9	Victoria Davis	Tennessee	Female	
		1	12					
	1-		13	Sales				
		1	14	19	Samantha Williams	Alabama	Female	
3			15					

从结果可以看到，在原计算表的第 9 行之后添加了一条空白的员工记录；在原计算表的第 3 行所在区段之后添加了一个新的部门记录，同时按照部门区段的结构在新部门下添加了一条空白的员工记录。添加记录的效果和插入记录类似，只是添加记录时，新加入的子行或者子区段位于当前记录或者区段的后面。

当同时选中多个连续的同层区段时，再执行添加记录，会根据选中的区段总数添加多个空白记录。

添加记录可以使用快捷键 Alt+Enter 执行。

6.3.3 复制记录

复制选择区段并插入。

在右键菜单中点击**记录→复制记录**，可以复制记录。

重新打开计算表 esCalc06_2.gex，分别选择第 9 行和第 3 行，复制记录，执行后结果如下：

0	1	2		A	B	C	D	E
1			1	Employees				
2-			2	EId	Name	State	Gender	
	1-		3	Finance				
		1	4	1	Rebecca Moore	Arizona	Female	
		1	5	6	Matthew Johnson	Washington	Male	
	1-		6	Finance				
		1	7	1	Rebecca Moore	Arizona	Female	
			8	6	Matthew Johnson	Washington	Male	
	1-		9	R&D				
		1	10	7	Alexis Smith	California	Female	
		1	11	8	Megan Wilson	New York	Female	
		1	12	9	Victoria Davis	Tennessee	Female	
		1	13	9	Victoria Davis	Tennessee	Female	
	1-		14	Sales				
		1	15	19	Samantha Williams	Alabama	Female	
3			16					

从结果可以看到，在原计算表的第 9 行的员工记录被复制了一遍，添加到所在的区段中；在原计算表的第 3 行的部门记录整个区段都被复制一遍，并添加到所在区段中。复制记录时，不仅仅添加了记录或者记录区段，而且会把当前的记录或者记录区段完全复制，如果当前记录中有多条子记录，也会被一起复制。

如果在同一个区段中，选择了多个连续的记录时，再执行复制记录，会复制所选中的所有记录区段。

6.3.4 删除记录

删除选择的记录或者选择的记录区段。

在右键菜单中点击**记录→删除记录**，可以删除记录。

我们再来看看计算表 esCalc06_2.gex：

0	1	2		A	B	C	D	E
1			1	Employees				
2-			2	EId	Name	State	Gender	
	1-		3	Finance				
		1	4	1	Rebecca Moore	Arizona	Female	
		1	5	6	Matthew Johnson	Washington	Male	
	1-		6	R&D				
		1	7	7	Alexis Smith	California	Female	
		1	8	8	Megan Wilson	New York	Female	
		1	9	9	Victoria Davis	Tennessee	Female	
	1-		10	Sales				
		1	11	19	Samantha Williams	Alabama	Female	
3			12					

在网格中，选择第 6 行，执行删除记录，结果如下：

0	1	2		A	B	C	D	E
1			1	Employees				
2-			2	EId	Name	State	Gender	
	1-		3	Finance				
		1	4	1	Rebecca Moore	Arizona	Female	
		1	5	6	Matthew Johnson	Washington	Male	
	1-		6	Sales				
		1	7	19	Samantha Williams	Alabama	Female	
3			8					

从结果中可以看到，原计算表中，第 6 行 R&D 的部门记录，以及其中的所有员工子记录行，都被一起删除。

此时，再选择第 7 行，执行删除记录，结果如下：

0	1	2		A	B	C	D	E
1			1	Employees				
2-			2	EId	Name	State	Gender	
	1-		3	Finance				
		1	4	1	Rebecca Moore	Arizona	Female	
		1	5	6	Matthew Johnson	Washington	Male	
	1		6	Sales				
3			7					

从结果中可以看到，选中的员工记录被删除。此时，Sales 部门中，一条记录都不存在，但是，区段的结构仍然和之前相同，并不会受删除记录的影响。

如果在同一个区段中，选择了多个连续的记录时，再执行删除记录，会删除所有选中的记录区段。

删除记录可以使用快捷键 **Shift+Delete** 来执行。

6.4 移动记录

6.4.1 记录上移

上移选择的区段，允许跨越它父行的区段。

在右键菜单中点击**记录→记录上移**，可以上移记录。

打开计算表 esCalc06_3.gex，网格如下：

0	1	2		A	B	C	D	E
1			1	Athletes				
2-			2	ID	Name	Country	Weight	
	1-		3	Heavyweight				
		1	4	111	Andrew Helm	AUS	80.5	
		1	5	124	Larsen Johnson	USA	89.7	
	1-		6	Middleweight				
		1	7	356	Wu Ye	CHN	78.9	
	1-		8	Lightweight				
		1	9	86	Alessio Pellegrini	ITA	60.0	
3			10					

分别选择第 5 行和第 9 行，执行记录上移，结果如下：

0	1	2		A	B	C	D	E
1			1	Athletes				
2-			2	ID	Name	Country	Weight	
	1-		3	Heavyweight				
		1	4	124	Larsen Johnson	USA	89.7	
		1	5	111	Andrew Helm	AUS	80.5	
	1-		6	Middleweight				
		1	7	356	Wu Ye	CHN	78.9	
		1	8	86	Alessio Pellegrini	ITA	60.0	
	1		9	Lightweight				
3			10					

从结果中可以看到，原网格中的第 5 行记录上移了一位；而由于第 9 行的运动员记录是 Lightweight 下的第 1 条记录，因此上移记录的时候，跨越了父行的区段，移到了 Middleweight 中。上移记录时，记录只会在它的同位行中移动，如果记录已经位于同位行中的最上方，如原网格中的第 4 行记录，则无法再进行上移记录操作。

此时，再选择第 6 行，执行记录上移，结果如下：

0	1	2		A	B	C	D	E
1			1	Athletes				
2-			2	ID	Name	Country	Weight	
	1-		3	Middleweight				
		1	4	356	Wu Ye	CHN	78.9	
		1	5	86	Alessio Pellegrini	ITA	60.0	
	1-		6	Heavyweight				
		1	7	124	Larsen Johnson	USA	89.7	
		1	8	111	Andrew Helm	AUS	80.5	
	1		9	Lightweight				
3			10					

可以看到，第 6 行的 Middleweight 记录，与其区段中的属行一起，上移了一位，移到了 Heavyweight 的记录区段之前。

如果在同一个区段中，选择了多个连续的记录时，再执行记录上移，会上移所选中的所有记录区段。

记录上移可以使用快捷键 Alt+Up 来执行。

6.4.2 记录下移

下移选择的区段，允许跨越它父行的区段。

在右键菜单中点击**记录→记录下移**，可以下移记录。

重新打开计算表 esCalc06_3.gex，选择第 4 行和第 7 行，记录下移，执行后结果如下：

0	1	2		A	B	C	D	E
1			1	Athletes				
2-			2	ID	Name	Country	Weight	
	1-		3	Heavyweight				
		1	4	124	Larsen Johnson	USA	89.7	
		1	5	111	Andrew Helm	AUS	80.5	
	1		6	Middleweight				
	1-		7	Lightweight				
		1	8	356	Wu Ye	CHN	78.9	
		1	9	86	Alessio Pellegrini	ITA	60.0	
3			10					

可以看到，下移记录与上移记录相反，选中记录会下移一位，如果选中记录是父行区段的最后一条记录，下移记录时会跨过父行的区段。下移记录时，记录只会在它的同位行中移动，如果记录已经位于同位行中的最下方，如原网格中的第 4 行记录，则无法再进行上移记录操作。

此时，再选择第 3 行，执行记录下移，结果如下：

0	1	2		A	B	C	D	E
1			1	Athletes				
2-			2	ID	Name	Country	Weight	
	1		3	Middleweight				
	1-		4	Heavyweight				
		1	5	124	Larsen Johnson	USA	89.7	
		1	6	111	Andrew Helm	AUS	80.5	
	1-		7	Lightweight				
		1	8	356	Wu Ye	CHN	78.9	
		1	9	86	Alessio Pellegrini	ITA	60.0	
3			10					

可以看到，第 3 行的 Heavyweight 记录，与其区段中的属行一起，下移了一位，移到了 Middleweight 的记录区段之后。

如果在同一个区段中，选择了多个连续的记录时，再执行记录下移，会下移所选中的所有记录区段。

记录上移可以使用快捷键 Alt+Down 来执行。

6.5 合并记录

将多个连续区段的子区段同位合并，主区段只留第一个。

在菜单栏中点击**数据→合并记录**，可以合并区段。

重新打开计算表 esCalc06_2.gex，网格如下：

0	1	2		A	B	C	D	E
1			1	Employees				
2-			2	EId	Name	State	Gender	
	1-		3	Finance				
		1	4	1	Rebecca Moore	Arizona	Female	
		1	5	6	Matthew Johnson	Washington	Male	
	1-		6	R&D				
		1	7	7	Alexis Smith	California	Female	
		1	8	8	Megan Wilson	New York	Female	
		1	9	9	Victoria Davis	Tennessee	Female	
	1-		10	Sales				
		1	11	19	Samantha Williams	Alabama	Female	
3			12					

选择第 3 行到第 11 行，在菜单栏中点击**数据→合并记录**，进行合并区段操作，执行后结果如下：

0	1	2			A	B	C	D	E
1				1	Employees				
2-				2	EId	Name	State	Gender	
	1-			3	Finance				
		1		4	1	Rebecca Moore	Arizona	Female	
		1		5	6	Matthew Johnson	Washington	Male	
		1		6	7	Alexis Smith	California	Female	
		1		7	8	Megan Wilson	New York	Female	
		1		8	9	Victoria Davis	Tennessee	Female	
		1		9	19	Samantha Williams	Alabama	Female	
3				10					

可以看到，选中区段的子记录被合并到了第一个主区段中，后面的主区段被删除。

再打开计算表 esCalc06_1.gex，网格如下：

0	1	2	3		A	B	C	D	E
1-				1	States and Cities				
	1-			2	A				
		1-		3		Alaska			
			1	4			Anchorage		
		1		5		Arizona			
	1-			6	C				
		1-		7		California			
			1	8			Los Angeles		
			1	9			San Diego		
			1	10			San Jose		
	1			11	D				
2				12					

选择第 2 行至第 6 行，执行合并记录，结果如下：

0	1	2	3		A	B	C	D	E
1-				1	States and Cities				
	1-			2	A				
		1-		3		Alaska			
			1	4			Anchorage		
		1		5		Arizona			
		1-		6		California			
			1	7			Los Angeles		
			1	8			San Diego		
			1	9			San Jose		
	1			10	D				
2				11					

可以看到，字母 A 和 C 列表下的所有州记录被合并，归属在了字母 A 的主区段中，字母 C 主区段被删除，而未被选中的字母 D 区段未受影响。而在州记录中各自的子记录城市记录仍然保留在各自的州记录区段中。

6.6 隐藏记录

6.6.1 隐藏记录

隐藏选定的记录区段。

在右键菜单中点击**记录→隐藏记录**，可以隐藏记录。

打开计算表 esCalc06_4.gex，网格如下：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6	28	Sat.	29	20	
		2		7	Max & Min		29	20	
		1-		8	week 5				
			1	9	29	Sun.	30	18	
			1	10	30	Mon.	24	16	
			1	11	31	Tues.	25	12	
		2		12	Max & Min		30	12	

在这个计算表中，C7,D7 和它们的同位格都是联动计算格。C7 中的计算式为==\{C5\}.max()，它和它的同位格 C12 用来统计各自周内的最高温度；D7 中的计算式为==\{D5\}.min()，它和它的同位格 D12 用来统计各自周内的最低温度。

现在分别选择第 6 行和第 9 行，执行隐藏记录操作，结果如下：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
		2		7	Max & Min		29	20	
		1-		8	week 5				
			1	10	30	Mon.	24	16	
			1	11	31	Tues.	25	12	
		2		12	Max & Min		30	12	

从结果中可以发现，第 6 行和第 9 行被隐藏了。隐藏记录和删除记录不同，隐藏记录时记录并没有消失，因此 C7,D7,C12,D12 中的计算结果未受影响。

再选择第 4 行，隐藏记录，结果如下：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		8	week 5				
			1	10	30	Mon.	24	16	
			1	11	31	Tues.	25	12	
		2		12	Max & Min		30	12	

可以发现，第 4 行所在的区段全部被隐藏了，包括它的属行和续行。

如果在同一个区段中，选择了多个连续的记录，可以同时执行隐藏记录。

6.6.2 显示记录

显示选定行所在的区段中被隐藏的记录区段。

在右键菜单中点击**记录→显示记录**，可以显示记录。

再来看 **6.6.1 隐藏记录**中设置了隐藏记录后的最终网格，或者直接打开计算表 esCalc06_5.gex，选择第 8 行，执行显示记录，结果如下：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		8	week 5				
			1	9	29	Sun.	30	18	
			1	10	30	Mon.	24	16	
			1	11	31	Tues.	25	12	
		2		12	Max & Min		30	12	

从结果中可以看到，第 8 行所在的区段中，被隐藏的记录显示了出来。

再选择第 3 行，执行显示记录，结果如下：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6	28	Sat.	29	20	
		2		7	Max & Min		29	20	
		1-		8	week 5				
			1	9	29	Sun.	30	18	
			1	10	30	Mon.	24	16	
			1	11	31	Tues.	25	12	
		2		12	Max & Min		30	12	

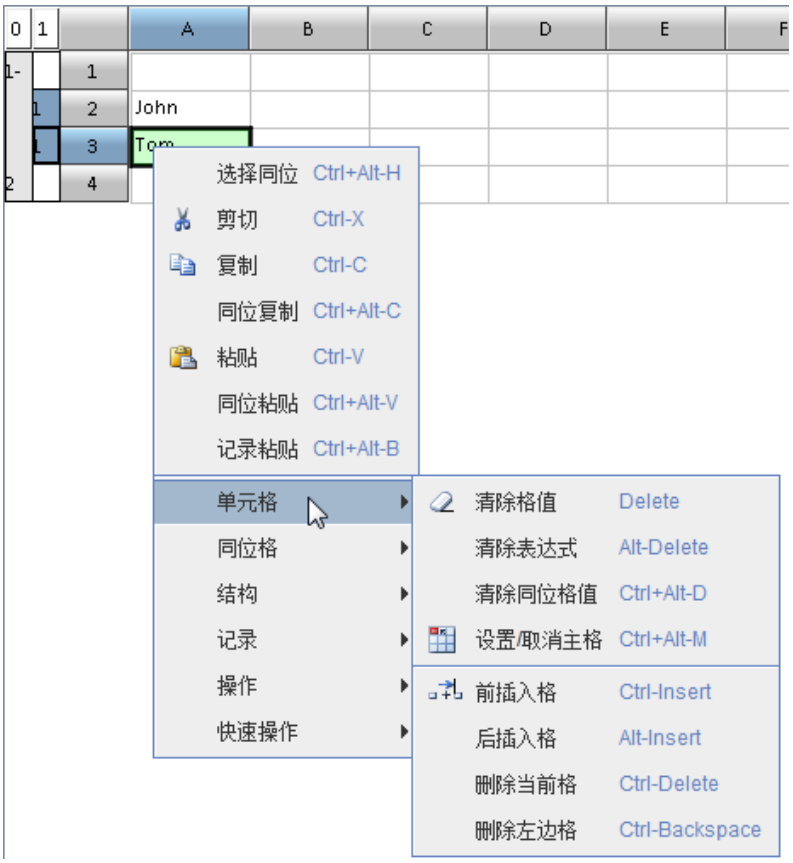
可以看到，第 3 行的所有记录，以及记录所在的整个区段，都被显示了出来。

如果在网格中选择了多行，执行显示记录时，会显示其中所有行及其所在区段的记录。

7.单元格编辑

7.1 单元格编辑

在计算表中，选择单元格后，可以在右键菜单中的单元格项目中选择各种操作：



7.1.1 单元格的清除操作

➢ **清除格值**

选择单元格，然后在右键菜单中，点击**单元格→清除格值**，可以将选定单元格的格值设为空值null。

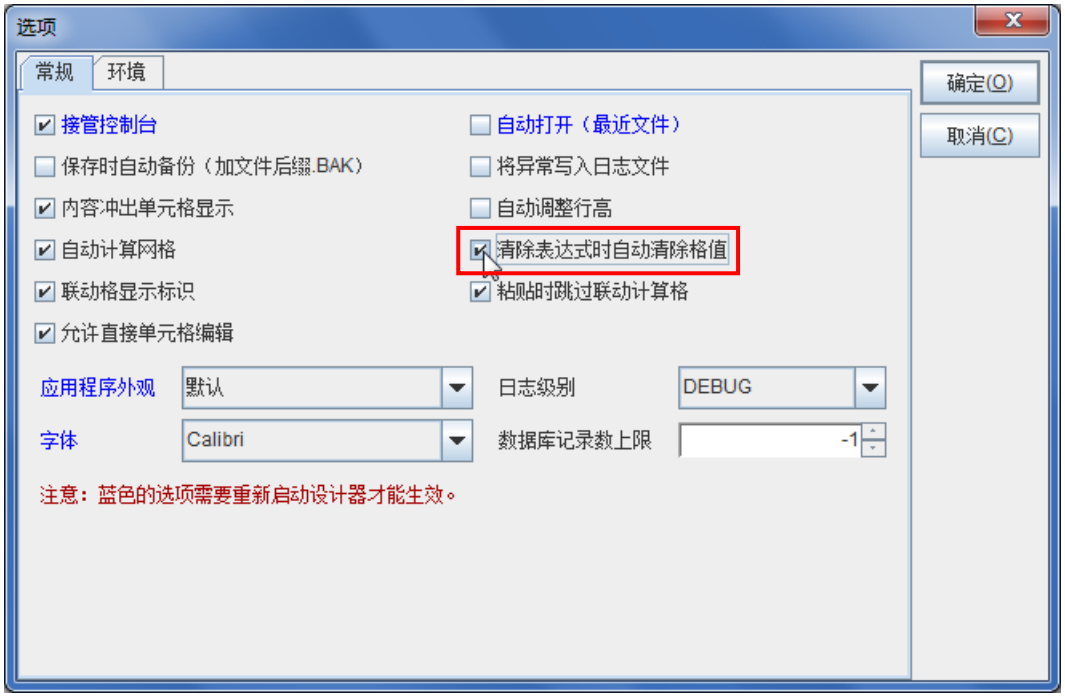
清除格值时，常数格中的格值会被清空；立即计算格的格值会被设为 null，除非修改计算式，否则不会再进行计算；联动计算格会被转为立即计算格，同时格值也被设为 null，除非修改计算式，否则也不会再进行计算。

清除格值也可以在选择单元格后按下 **Delete** 键执行。

➢ **清除表达式**

选择单元格，然后在右键菜单中，点击**单元格→清除表达式**，可以清除选定单元格的表达式。

清除表达式时，根据设置的不同会有不同的操作效果。可以在菜单栏中点击**工具→选项**，在选项设置窗口中，设定清除表达式时是否清除格值。如果勾选了这个选项，在清除计算式时，会同时执行清除格值操作；否则会保留当前格值，仅仅清除计算式。



清除表达式也可以在选择单元格后同时按下 **Alt+Delete** 键执行。

➢ **清除同位格值**

选择单元格，然后在右键菜单中，点击**单元格→清除同位格值**，可以清除选定单元格所有同位格的值。

清除同位格值也可以在选择单元格后同时按下 Ctrl+**Alt+D** 键执行。

7.1.2 设定主格

➢ **设定主格**

选择单元格，然后在右键菜单中，点击**单元格→设置/取消主格**，可以将单元格设为主格，或者将主格设定去除。

设立主格，在 **3.9 主格**中进行了详细的说明。

7.1.3 删除/插入单元格

➢ **前插入格**

在选中的单元格处插入一个空白格，并从它开始将本行中右侧的单元格依次右移一格。当前行的所有同位行中也将进行同样的操作，如果这些同位行中的某一行最后一个单元格中有值，那么将会先在网格的最后添加一个空白列，然后再右移各单元格。选择单元格后，在右键菜单中，点击**单元格→前插入格**，可以执行前插入格。

前插入格也可以在选择单元格后同时按下 **Ctrl-Insert** 键执行。

➢ **后插入格**

在选中的单元格右边插入一个空白格，并从它右侧的单元格开始将本行中右侧的单元格依次右移一格。当前行的所有同位行中也将进行同样的操作，如果这些同位行中的某一行最后一个单元格中有值，那么将会先在网格的最后添加一个空白列，然后再右移各单元格。选择单元格后，在右键菜单中，点击**单元格→后插入格**，可以执行后插入格。

后插入格也可以在选择单元格后同时按下 **Alt-Insert** 键执行。

➢ **删除当前格**

选定单元格后，点击**单元格→删除当前格**，则选中的单元格及其同位格都会被移除，右侧的单元格会左移过来补齐，本行及其同位行的最后会补加一个空白单元格；而非同位行中的单元格不会受到影响。执行删除当前格后，计算表中的列数不受影响。

删除当前格也可以在选择单元格后同时按下 **Ctrl-Delete** 键执行。

➢ **删除左边格**

选定单元格后，点击**单元格→删除当前格**，则当前位置左侧的单元格及其同位格都会被移除，从当前单元格起右侧的单元格会左移过来补齐，本行及其同位行的最后会补加一个空白单元格；而非同位行中的单元格不会受到影响。执行删除左边格后，计算表中的列数不受影响。

删除左边格也可以在选择单元格后同时按下 **Ctrl-Backspace** 键执行。

再来看看计算表 esCalc06_4.gex：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6	28	Sat.	29	20	
		2		7	Max & Min		29	20	
		1-		8	week 5				
			1	9	29	Sun.	30	18	
			1	10	30	Mon.	24	16	
			1	11	31	Tues.	25	12	
		2		12	Max & Min		30	12	

选择 C5，执行前插入格，结果如下：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.		28	21
			1	6	28	Sat.		29	20
		2		7	Max & Min		29	20	
		1-		8	week 5				
			1	9	29	Sun.		30	18
			1	10	30	Mon.		24	16
			1	11	31	Tues.		25	12
		2		12	Max & Min		30	12	

可以看到，C5 格和它右侧的单元格右移，它原来的位置插入了一个空白格，第 5 行的所有同位行中也进行了同样的操作。此时再选择 D6，执行删除当前格，结果如下：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.		21	
			1	6	28	Sat.		20	
		2		7	Max & Min		29	20	
		1-		8	week 5				
			1	9	29	Sun.		18	
			1	10	30	Mon.		16	
			1	11	31	Tues.		12	
		2		12	Max & Min		30	12	

从结果中看到，D6 格被删除，它右侧的单元格依次左移，本行的最后补充了一个空白格，第 6 行的所有同位行中也进行了同样的操作。需要注意的是 C7 和 C12 单元格右下角的标记变成了红色的小三角，这个标记提示这两个单元格的计算式中引用的单元格被删除，出现了错误。

此时再选择 D7，执行删除左边格，结果如下：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.		21	
			1	6	28	Sat.		20	
		2		7	Max & Min		20		
		1-		8	week 5				
			1	9	29	Sun.		18	
			1	10	30	Mon.		16	
			1	11	31	Tues.		12	
		2		12	Max & Min		12		

可以看到，D7 和它的同位格 D12 的左侧单元格被删除，同时这两个同位格及其右侧的单元格依次左移，在行后补充空白格。

再选择 A5，执行后插入格，结果如下：

0	1	2	3		A	B	C	D	E
1				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27		Fri.		21
			1	6	28		Sat.		20
		2		7	Max & Min		20		
		1-		8	week 5				
			1	9	29		Sun.		18
			1	10	30		Mon.		16
			1	11	31		Tues.		12
		2		12	Max & Min		12		

可以看到，在 A5 及其同位格的后面插入了一个空白格，同时这些同位格后面的单元格依次右移。

执行删除/插入单元格时，可以选择多个连续的单元格，但是如果选择了多列，只有最左侧一列中的单元格会被有效选中，此时删除/插入单元格的操作将在所有选中行及它们的同位行中执行。

7.2 同位格复制与粘贴

在计算表中，为了便于在网格中复制粘贴数据，提供了一些专门在同位格中使用的操作。

➢ **选择同位**

在同一个记录区段中的同层续行中选择几个单元格，然后在右键菜单中，点击选择同位，按照指定的选取方向,选出指定单元格在当前操作区内的所有同位格，可以同时对选中的单元格执行操作，如清除格值等。

在选择同位格时，在弹出选项窗口中有三种模式可以选择，**全部选择**、**向上选择**以及**向下选择**，可以根据需要设定。

➢ **同位复制**

在同一个记录区段中的同层续行中选择几个单元格，然后在右键菜单中，点击同位复制，可以复制指定单元格在当前操作区内的所有同位格，可以将同位格中的数据进行拷贝出来，粘贴到网格的其它位置，或者其它计算表中，也可以粘贴到文本文件中。

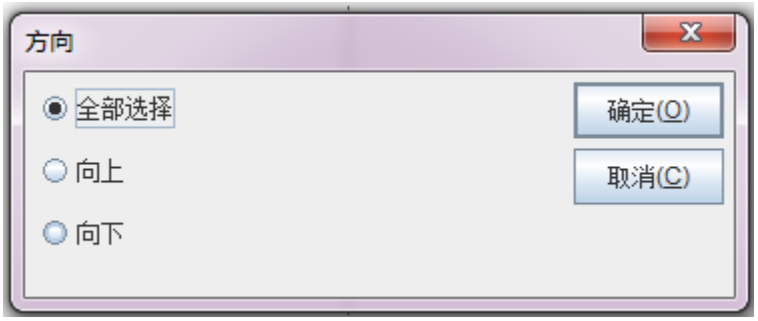
➢ **同位粘贴**

选择一个单元格，然后在右键菜单中，点击同位粘贴，可以从指定单元格开始，将数据依次粘贴到单元格中；如果有多行数据，将顺次粘贴到当前行排在后面的所有同位行中；如果同位行的数量不足，将按照数据的行数在最后一个父层区段中自动添加记录。同位格粘贴时，与操作区的设定无关。如果需要粘贴多列数据，但是网格区的列数不足，将会弹出窗口以供选择是否添加列。

打开计算表 esCalc07_1.gex，如下：

0	1	2		A	B	C	D	E
1-			1					
	1-		2	Week 1				
		1	3	Sunday		a		
		1	4	Monday		b		
		1	5	Tuesday		c		
		1	6	Wednesday		d		
	1-		7	Week 2				
		1	8	Sunday		e		
	1-		9	Week 3				
		1	10	Sunday		f		

在网格中，选择 A4 格，执行选择同位格，弹出操作选项窗口如下：



如果设定为全选择，确认后，A3、A4、A5、A6、A8、A10 几个单元格变色，即选中了 A4 的所有同位格。如果设定为向上选择，则会选中 A3 和 A4 两个单元格，这两个单元格变色。如果设定为向下选择，则会选中 A4、A5、A6、A8 和 A10 这 5 个单元格。

现在按住 Ctrl 键，同时选择单元格 A4 和 C4，执行同位复制，执行后可以看到 A3 至 A6、A8、A10 以及 C3 至 C6、C8、C10 这些单元格全会变色，说明这些单元格中的格值被复制。点击 D4，执行同位粘贴，执行后结果如下：

0	1	2		A	B	C	D	E
1-			1					
	1-		2	Week 1				
		1	3	Sunday		a		
		1	4	Monday		b	Sunday	a
		1	5	Tuesday		c	Monday	b
		1	6	Wednesday		d	Tuesday	c
	1-		7	Week 2				
		1	8	Sunday		e	Wednesday	d
	1-		9	Week 3				
		1	10	Sunday		f	Sunday	e
			11				Sunday	f

从结果中可以看出，同位格粘贴时，只粘贴了单元格的格值，而没有改变单元格的属性。由于复制了 6 条数据，而从第 4 行起只有 5 条同位记录，因此自动在最后一个父层记录 Week3 中添加了一条同位记录。

此时，同时选中 C5 和 E5，在右键菜单中执行选择同位，在弹出的菜单中选择选择全部，可以看到同位格 C3 至 C6、C8、C10，以及 E3 至 E6、E8、E10 都被选中，按 **Delete** 键清除格值，结果如下：

0	1	2		A	B	C	D	E
1-			1					
	1-		2	Week 1				
		1	3	Sunday				
		1	4	Monday			Sunday	
		1	5	Tuesday			Monday	
		1	6	Wednesday			Tuesday	
	1-		7	Week 2				
		1	8	Sunday			Wednesday	
	1-		9	Week 3				
		1	10	Sunday			Sunday	
			11				Sunday	

可以看到选中单元格中的格值被清除。

7.3 记录粘贴

与同位格粘贴类似，在计算表中还有一种操作是记录粘贴，可以在右键菜单中选择记录粘贴来执行。执行记录粘贴，将在当前选中的记录之前添加新记录，并将拷贝的数据逐条设入。

我们继续使用 **7.2 同位格复制与粘贴**中的计算表，或者直接打开计算表 esCalc07_2.gex，如下：

0	1	2		A	B	C	D	E
1-			1					
	1-		2	Week 1				
		1	3	Sunday				
		1	4	Monday			Sunday	
		1	5	Tuesday			Monday	
		1	6	Wednesday			Tuesday	
	1-		7	Week 2				
		1	8	Sunday			Wednesday	
	1-		9	Week 3				
		1	10	Sunday			Sunday	
		1	11				Sunday	

同时选择 A11 和 D11，执行同位复制，可以看到， A11 和 D11 的所有同位格中的数据被复制，然后点击 C4 格，执行记录粘贴，结果如下：

	1	2		A	B	C	D	E
1-			1					
	1-		2	Week 1				
		1	3	Sunday				
		1	4			Sunday		
		1	5			Monday	Sunday	
		1	6			Tuesday	Monday	
		1	7			Wednesday	Tuesday	
		1	8			Sunday	Wednesday	
		1	9			Sunday	Sunday	
		1	10				Sunday	
		1	11	Monday			Sunday	
		1	12	Tuesday			Monday	
		1	13	Wednesday			Tuesday	
	1-		14	Week 2				
		1	15	Sunday			Wednesday	
	1-		16	Week 3				
		1	17	Sunday			Sunday	
		1	18				Sunday	

从结果中可以看到，根据被复制的数据，在原网格的 C4 格之前插入了 7 条记录，并从选择列开始将数据依次粘贴。

在计算表中，进行记录粘贴操作前，所复制的数据，既可以从网格中进行复制，也可以从外部文件中拷贝，如 Txt 文件、Excel 文件，数据库浏览器中等等。

7.4 同位格操作

在计算表中，还提供了另外一些同位格专用的操作，可以在右键菜单中的同位格项目中进行选择：

➢ **同位格插入**

在选中的单元格处插入空白格，并从它开始将位于操作区中的同位格原有的格值依次下移，操作区后面将插入一条新同位记录，在其同位格中置入操作区中最后一个同位格的原值。执行操作时选择的单元格必须位于同一区段中，且是连续的同层格，在右键菜单中，点击**同位格→同位插入**，可以进行同位格插入。

➢ **同位格删除**

删除选中单元格的值，并将它后面位于操作区中的同位格的原有格值依次上移。执行操作时选择的单元格必须位于同一区段中，且时连续的同层格，在右键菜单中，点击**同位格→同位删除**，可以执行同位格删除。

➢ **复制到同位格**

将选中单元格的值，复制到操作区中指定位置的同位格中。执行操作时选择的单元格必须位于同一区段中，且时连续的同层格，在右键菜单中，点击**同位格→复制到同位格**，可以执行复制到同位格操作。

打开计算表 esCalc07_3.gex：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6	28	Sat.	29	20	
		1-		7	week 5				
			1	8	29	Sun.	30	18	
			1	9	30	Mon.	24	16	
			1	10	31	Tues.	25	12	
	1-			11	February				
		1-		12	week 1				
			1	13	1	Wed.	35	11	
			1	14	2	Thur.	38	16	
3				15					

同时选择 A6 和 B6，执行同位格插入，结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6			29	20	
		1-		7	week 5				
			1	8	28	Sat.	30	18	
			1	9	29	Sun.	24	16	
			1	10	30	Mon.	25	12	
	1-			11	February				
		1-		12	week 1				
			1	13	31	Tues.	35	11	
			1	14	1	Wed.	38	16	
			1	15	2	Thur.			
3				16					

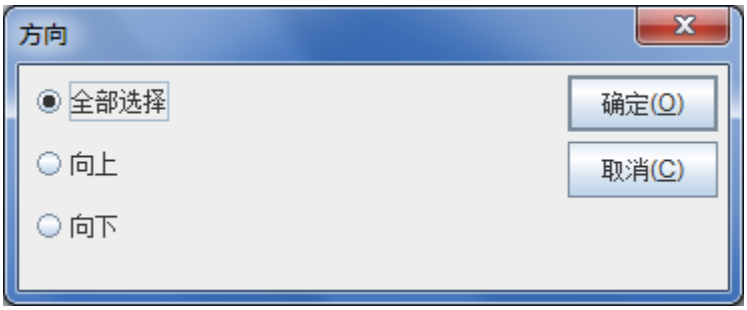
可以看到，从第 6 行起 A6 的同位行中，AB 两列中的数据都下移了一位，其它列的单元格值不变。在最后一个同位行之后添加了一个同位行。

在这个网格中，同时选择 A8 和 B8，执行同位格删除，结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6			29	20	
		1-		7	week 5				
			1	8	29	Sun.	30	18	
			1	9	30	Mon.	24	16	
			1	10	31	Tues.	25	12	
	1-			11	February				
		1-		12	week 1				
			1	13	1	Wed.	35	11	
			1	14	2	Thur.	38	16	
			1	15					
3				16					

可以看到，在第 8 行之后的同位行第 9、10、13、14 和 15 行，它们的 AB 两列的值上移了一位，其它列中的单元格值不变。

选择 B8 格，执行复制到同位格，弹出选项面板如下：



在这个面板中，可以设定复制同位格的执行方向。向上执行，会将选中单元格的格值，复制到操作区中排在当前格之前的同位格中。向下执行，会将选中单元格值，复制到操作区中排在当前格之后的同位格中。全部选择，将复制到所有的同位格。

选择**向下**执行，执行复制到同位格后，结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6			29	20	
		1-		7	week 5				
			1	8	29	Sun.	30	18	
			1	9	30	Sun.	24	16	
			1	10	31	Sun.	25	12	
	1-			11	February				
		1-		12	week 1				
			1	13	1	Sun.	35	11	
			1	14	2	Sun.	38	16	
			1	15		Sun.			
3				16					

B8 中的单元格值，被复制到了它下面的各个同位格中。

在计算表中，进行同位格粘贴操作前，所复制的数据，既可以从网格中进行复制，也可以从外部文件中拷贝，如 Txt 文件、Excel 文件，数据库浏览器中等等。

➢ **左移单元格**

将选中的单元格及它的所有同位格，与它们左侧相邻的格子交换。执行左移单元格，可以在右键菜单中，点击**同位格→左移单元格**。

打开计算表 esCalc07_4.gex，网格如下：

	1	2		A	B	C	D	E
1			1	Athletes				
2-			2	ID	Name	Country	Weight	
	1-		3	Heavyweight				
		1	4	111	Andrew Helm	AUS	80.5	
		1	5	124	Larsen Johnson	USA	89.7	
	1-		6	Middleweight				
		1	7	356	Wu Ye	CHN	78.9	
	1-		8	Lightweight				
		1	9	86	Alessio Pellegrini	ITA	60.0	
3			10					

选择 D5，执行左移单元格，结果如下：

0	1	2		A	B	C	D	E
1			1	Athletes				
2-			2	ID	Name	Country	Weight	
	1-		3	Heavyweight				
		1	4	111	Andrew Helm	80.5	AUS	
		1	5	124	Larsen Johnson	89.7	USA	
	1-		6	Middleweight				
		1	7	356	Wu Ye	78.9	AUS	
	1-		8	Lightweight				
		1	9	86	Alessio Pellegrini	60.0	AUS	
3			10					

可以看到，左移单元格时，单元格的外观属性随之移动，但单元格的列宽不受影响。如果选中 A 列的单元格，左移单元格没有任何效果。

左移单元格可以使用快捷键 **Alt+Left**。

➢ **右移单元格**

将选中的单元格及它的所有同位格，与它们右侧相邻的格子交换。执行右移单元格，可以在右键菜单中，点击**同位格→右移单元格**。如果在上面的网格中，选择 C9，执行右移单元格，则会得到最初的网格。

右移单元格可以使用快捷键 **Alt+Right**。

➢ **上移单元格**

将选中的单元格及它的所有同位格，与它们上侧相邻的同一区段的格子交换。执行上移单元格，可以在右键菜单中，点击**同位格→上移单元格**。

如，重新打开计算表 esCalc07_4.gex，同时选择单元格 A2~E2，然后执行上移单元格，结果如下：

	1	2		A	B	C	D	E
1			1	ID	Name	Country	Weight	
2-			2	Athletes				
	1-		3	Heavyweight				
		1	4	111	Andrew Helm	80.5	AUS	
		1	5	124	Larsen Johnson	89.7	USA	
	1-		6	Middleweight				
		1	7	356	Wu Ye	78.9	AUS	
	1-		8	Lightweight				
		1	9	86	Alessio Pellegrini	60.0	AUS	
3			10					

与左移单元格类似，上移后单元格的外观属性同时移动，但行高等属性不会变化。上移单元格时，只能从区段中的 1 个续行，移动到与之相邻的另一个续行。

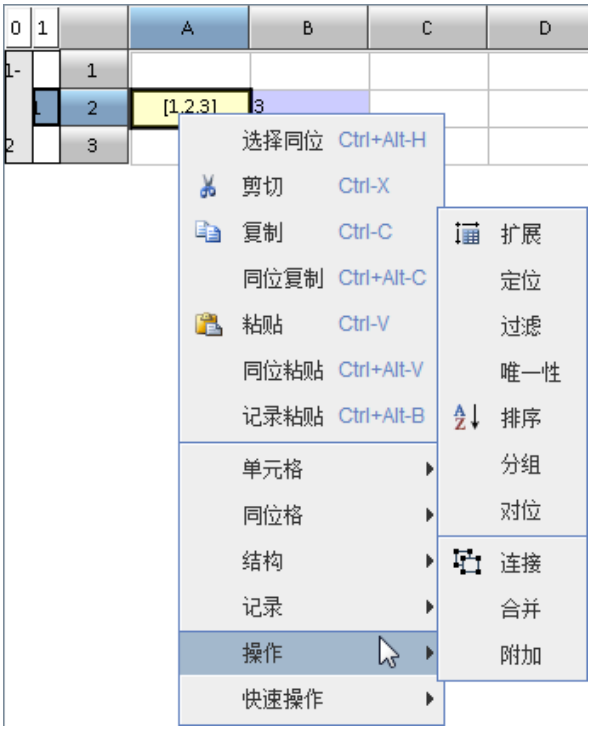
➢ **下移单元格**

将选中的单元格及它的所有同位格，与它们下侧相邻的同一区段的格子交换。执行下移单元格，可以在右键菜单中，点击**同位格→下移单元格**。

8. 数据操作

在统计数据时，经常需要对原始数据进行各种整理。在计算表中，可以对数据进行扩展，过滤，分组等各种操作。

如果需要对计算表中选定的行或者单元格中的数据进行操作，可以单击右键，在右键菜单中的操作项目，可以选择对数据进行各种操作。



也可以在选中相应的单元格后，在菜单栏中点击操作选项，选择需要的操作。

8.1 扩展

选定若干取值为集合的格，据其格值复制多遍区段，并将集合格值拆到各区段中；同位格同步扩展。

在右键菜单中点击**操作→扩展**，可以执行扩展操作。

打开计算表 esCalc08_1.gex，如下面的网格：

0	1		A	B	C	D	E
1-		1					
	1	2	**[1,2,3]**	**3**			
2		3					

选中 A2 单元格，执行扩展操作，结果如下：

0	1			A	B	C	D	E
1-			1					
	1		2	1	3			
	1		3	2	3			
	1		4	3	3			
2			5					

从结果中可以看到，由于原网格中 A2 的值是集合，所以可以进行扩展。A2 中的值是序列[1,2,3]，共有 3 个成员，在扩展后，A2 所在的行被复制扩展为 3 行，每一行中 A2 所对应的单元格的格值依次设为集合中的一个成员。扩展中，A2 所在行中各个单元格的前景色，背景色等属性，以及 A2 之外的格值等等属性都会被复制。

如果需要扩展的单元格所在的行存在子行或者续行时又如何呢？我们打开计算表 esCalc08_2.gex，来看下面的计算表：

0	1	2		A	B	C	D	E
1-			1	Game Record				
	1-		2	[G1,G2]				
		1	3		1st Half			
		1	4		2nd Half			
2-			5	Students				
	1		6	Name	[Jack,Tom]			
	2		7	Gender				

在网格中，分别选中 B6 和 A2 单元格，执行扩展操作，结果如下：

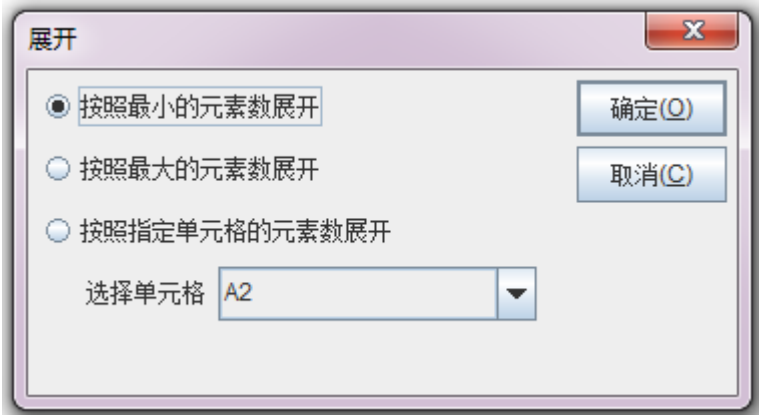
0	1	2		A	B	C	D	E
1-			1	Game Record				
	1-		2	G1				
		1	3		1st Half			
		1	4		2nd Half			
	1-		5	G2				
		1	6		1st Half			
		1	7		2nd Half			
2-			8	Students				
	1		9	Name	Jack			
	2		10	Gender				
	1		11	Name	Tom			
	2		12	Gender				

从结果中可以看到，当扩展格所在的行拥有主行或者同级的续行的时候，那么它所在的整个区段都会在扩展时被复制。

如果同时选中了多个单元格时，又会如何呢？打开 esCalc08_3.gex，看看下面的计算表：

0	1		A	B	C	D	E
1-		1					
	1	2	[1,2,3]	[a,b]			
2		3					

同时选中 A2 和 B2 两个单元格，执行扩展操作，此时会弹出对话框：



在这个对话框中，可以选择 3 种扩展模式：按照最小的元素数展开，按照最大的元素数展开，按照指定单元格的元素数展开。

选择**按照最小的元素数展开**时，结果如下：

0	1		A	B	C	D	E
1-		1					
	1	2	1	a			
	1	3	2	b			
2		4					

在这个例子中，B2 中的成员数最多，是 2 个，所以按最小成员数扩展，会扩展为 2 行，扩展行中各个单元格的值依次设置。

选择按最大成员数扩展时，结果如下：

0	1		A	B	C	D	E
1-		1					
	1	2	1	a			
	1	3	2	b			
	1	4	3				
2		5					

在这个例子中，A2 中的成员数最多，是 3 个，所以按最大成员数扩展，会扩展为 3 行，B2 中的成员不足的在扩展行中格值会设为 null。

如果选择单元格扩展，那么会按照所选单元格中的成员数进行扩展。

选择多个单元格扩展时，多个单元格可以在同一行的多列中，或者在同一个区段连续的同层行中。

打开计算表 esCalc08_4.gex，网格如下：

0	1		A	B	C	D	E
1-		1					
	1	2	[1,2,3]	a			
	1	3	4	b			
	1	4	[5,6]	c			
2		5					

选择 A2，执行扩展操作，结果如下：

0	1		A	B	C	D	E
1-		1					
	1	2	1	a			
	1	3	2	a			
	1	4	3	a			
	1	5	4	b			
	1	6	5	c			
	1	7	6	c			
2		8					

从结果可以看到，如果某个单元格执行扩展操作时，如果它有同位格，那么它的同位格也会同时执行扩展操作。

8.2 定位

找到满足条件的同位格，将光标移过去。

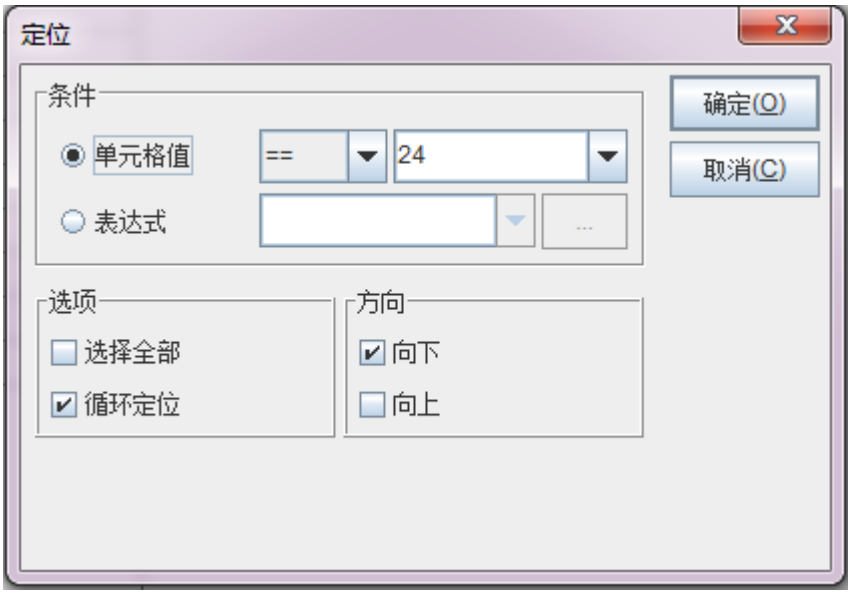
在右键菜单中点击**操作→定位**，可以执行定位操作。

定位操作会对所有同位格中的数据进行判断，将光标移到第一个满足设定条件的同位格中。打开计算表 esCalc08_5.gex，网格如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6	28	Sat.	29		
		1-		7	week 5				
			1	8	29	Sun.	30	18	
			1	9	30	Mon.	24	16	
			1	10	31	Tues.	25	0	
	1-			11	February				
		1-		12	week 1				
			1	13	1	Wed.	35	0	
			1	14	2	Thur.	38	16	
3				15					

计算表中记录了某地一段时间内的气温数据。

选择 C5，执行定位操作，弹出定位操作面板如下：



定位操作面板中，在上面部分可以设置定位条件。定位条件中，可以设定单元格值等于、不等于、大于、不大于、小于、不小于某个设定值；也可以设定单元格值查找条件为最大值或最小值。还可以直接填写定位条件表达式，在使用定位条件表达式时，用@表示格值，如@>25 表示查找条件为单元格值大于 25。

定位操作面板的下面部分，可以设置定位操作选项：是否选择全部，是否循环定位；还可以设置定位操作的方向是向下或向上。在查找满足条件的同位格时，会按照设定的查找方向进行。如果设定了选择全部，则会选出所有满足条件的同位格，否则只会选择第一个，而光标都会移动到第一个满足条件的同位格中。如果未设定循环定位，则按照查找方向到达最上方或者最下方的同位格后，会停止查找，否则会循环进行定位。

如果在定位操作面板中选择单元格值，设置==24，确认执行定位后，光标会移到 C9，如果在定位表达式中填写@==24，定位执行的结果是一样的。

如果在定位操作面板中选择单元格值，将条件选为最大值，确认执行后，光标会移到 C14。

8.3 过滤

指定某格值在其父层内过滤。

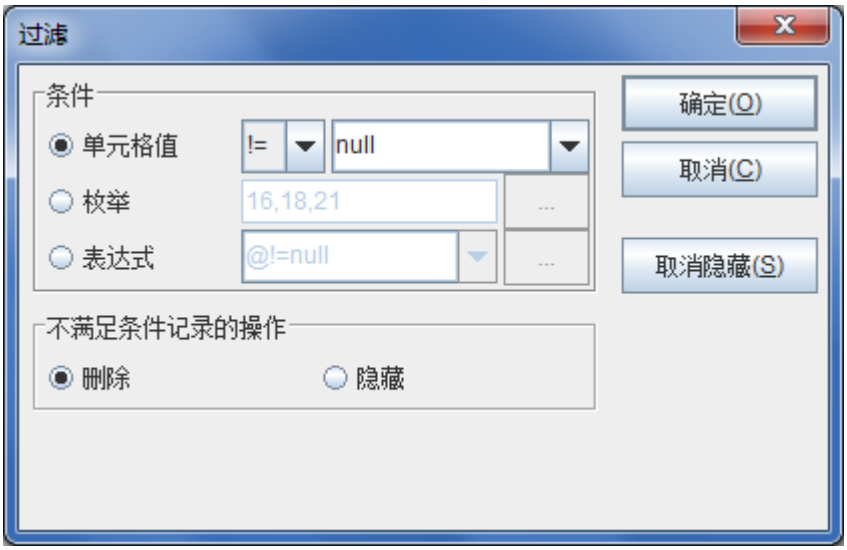
在右键菜单中点击**操作→过滤**，可以执行过滤操作。

过滤操作会对所有同位格中的数据进行判断，将不满足条件的同位格所在的记录区段删除或隐藏。我们重新打开计算表 esCalc08_5.gex：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
			1	6	28	Sat.	29		
		1-		7	week 5				
			1	8	29	Sun.	30	18	
			1	9	30	Mon.	24	16	
			1	10	31	Tues.	25	0	
	1-			11	February				
		1-		12	week 1				
			1	13	1	Wed.	35	0	
			1	14	2	Thur.	38	16	
3				15					

我们注意到，D 列的最低气温数据里，有的数据为 0，有的数据缺失。我们认为最低气温为 0 或者 null 的数据都是非法数据，需要进行过滤。

为此，选择 D5 格，执行过滤，会弹出过滤操作面板：



在过滤面板中，可以设置过滤条件，过滤条件的设置和定位条件类似，只不过多出一个**枚举**的过滤条件，当在过滤条件中选择**枚举**时，可以一次选择多个需要保留的格值。对于不满足条件的记录区段，可以选择删除或者隐藏。

选择过滤条件单元格值，设置过滤条件为!=null，选择删除将不满足条件的结果删除，执行过滤后结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
		1-		6	week 5				
			1	7	29	Sun.	30	18	
			1	8	30	Mon.	24	16	
			1	9	31	Tues.	25	0	
	1-			10	February				
		1-		11	week 1				
			1	12	1	Wed.	35	0	
			1	13	2	Thur.	38	16	
3				14					

可以看到，最低温度为空的记录被删除了。实际上，过滤时，选择的单元格在哪一个同位行中都是可以的。

再选择 D13 格，这一次过滤条件仍然为单元格值，设置过滤条件为!=0 进行过滤，选择删除将不满足条件的记录删除，执行过滤后结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
			1	5	27	Fri.	28	21	
		1-		6	week 5				
			1	7	29	Sun.	30	18	
			1	8	30	Mon.	24	16	
	1-			9	February				
		1-		10	week 1				
			1	11	2	Thur.	38	16	
3				12					

可以看到，最低温度为 0 的记录也被删除了。

再选择 C5，执行过滤，直接填写过滤条件表达式为@>28，选择隐藏将不满足条件的记录隐藏，执行过滤后结果如下：

0	1	2	3		A	B	C	D	E
1-				1	Temperature in F (2012)				
2-				2	Date	Day of Week	Max Temp.	Min Temp.	
	1-			3	January				
		1-		4	week 4				
		1-		6	week 5				
			1	7	29	Sun.	30	18	
	1-			9	February				
		1-		10	week 1				
			1	11	2	Thur.	38	16	
3				12					

可以看到，通过过滤，将最高气温大于 28℉ 的记录保留，其它记录被隐藏。此时，如果选择 C7 进行过滤，然后点击过滤操作面板中的取消隐藏按钮，可以将过滤范围内，所有被隐藏的同位行都显示出来。

打开计算表 esCalc08_6.gex，来看下面的网格：

0	1	2		A	B	C
1-			1	2011 Grosses		
2-			2	Title	MPAA Rating	Total Gross
	1-		3	Universal		
		1	4	Fast Five	PG-13	209837675
		1	5	Bridesmaids	R	169106725
	2		6		Total	378944400
	1-		7	Warner Bros.		
		1	8	Harry Potter and the Deathly Hallows Part 2	PG-13	381011219
		1	9	The Hangover Part II	R	254464305
		1	10	Sherlock Holmes: A Game of Shadows	PG-13	186271351
	2		11		Total	821746875
	1-		12	Buena Vista		
		1	13	Pirates of the Caribbean: On Stranger Tides	PG-13	241071802
		1	14	Cars 2	G	191452396
	2		15		Total	432524198

这个网格中的数据是 2011 年一些电影的票房情况，其中 C6 中的计算式是=={C4}.sum()，它和它的同位格都是联动计算格，计算同一个公司发行电影的总票房。现在要根据总票房做出过滤，保留总票房小于 500,000,000 的电影公司的票房信息。为此，我们选择 C6 格，执行过滤操作，直接设定过滤表达式为@<500000000，选择删除将不满足条件的记录删除，执行过滤后结果如下：

0	1	2		A	B	C
1-			1	2011 Grosses		
2-			2	Title	MPAA Rating	Total Gross
	1-		3	Universal		
		1	4	Fast Five	PG-13	209837675
		1	5	Bridesmaids	R	169106725
	2		6		Total	378944400
	1-		7	Buena Vista		
		1	8	Pirates of the Caribbean: On Stranger Tides	PG-13	241071802
		1	9	Cars 2	G	191452396
	2		10		Total	432524198

可以看到，通过对 C6 和它的同位格进行过滤，将总票房小于 500,000,000 的电影公司的票房信息保留，其它记录被删除。被删除的记录所在的整个区段会被一起删除，它的子记录以及与它同处在一个区段中的续行，都会被一同删除。

8.4 唯一性

针对某个格值在父层内的唯一性处理。

在右键菜单中点击**操作→唯一性**，可以执行唯一性操作。

打开计算表 esCalc08_7.gex，网格如下：

0	1		A	B	C
1		1	Gymnastics Artistic Women's Individual All-Around Results		
2-		2	Name	Event	Score
	1	3	Grace Miller	Vault	14.000
	1	4	Kayla Rodriguez	Vault	14.575
	1	5	Kayla Rodriguez	Uneven bars	14.675
	1	6	Kayla Rodriguez	Balance beam	13.975
	1	7	Kayla Rodriguez	Floor	14.400
	1	8	Lauren Davis	Vault	14.275
	1	9	Grace Miller	Floor	13.975
	1	10	Grace Miller	Uneven bars	12.875
	1	11	Lauren Davis	Floor	13.000
	1	12	Grace Miller	Balance beam	13.100
	1	13	Daniel Smith	Parallel bars	14.375
	1	14	Lauren Davis	Balance beam	15.225
	1	15	Lauren Davis	Uneven bars	14.275

这个计算表中，记录了某次女子全能体操比赛的成绩，下面我们通过对姓名进行唯一性操作来整理成绩记录。

选择 A3 单元格，执行唯一性操作，会弹出唯一性操作面板如下：

在唯一性面板中，可以选择对同值记录的 4 种操作方式：合并，去重，删除，无动作。还可以设定在判断同值时是否只处理相邻，也可以选择是否在做唯一性操作时，删除无同值的记录。

先来选择**去重**进行唯一性操作，结果如下：

0	1		A	B	C
1		1	Gymnastics Artistic Women's Individual All-Around Results		
2-		2	Name	Event	Score
	1	3	Grace Miller	Vault	14.000
	1	4	Kayla Rodriguez	Vault	14.575
	1	5	Lauren Davis	Vault	14.275
	1	6	Daniel Smith	Parallel bars	14.375

通过和原计算表对比，可以看到，按照姓名进行去重的唯一性操作后，每个姓名只留下了第一条记录，其它记录均被删除。

在 esCalc08_7.gex 的计算表中，选择 A3，设定**去重**，同时选定**同值判断只处理相邻**，执行唯一性操作，结果如下：

0	1		A	B	C
1		1	Gymnastics Artistic Women's Individual All-Around Results		
2-		2	Name	Event	Score
	1	3	Grace Miller	Vault	14.000
	1	4	Kayla Rodriguez	Vault	14.575
	1	5	Lauren Davis	Vault	14.275
	1	6	Grace Miller	Floor	13.975
	1	7	Lauren Davis	Floor	13.000
	1	8	Grace Miller	Balance beam	13.100
	1	9	Daniel Smith	Parallel bars	14.375
	1	10	Lauren Davis	Balance beam	15.225

将结果与刚刚的对比，可以发现，选定了同值判断只处理相邻之后，只有相邻的同值记录会保留第一个，如果不相邻，则不会进行唯一性操作。

在 esCalc08_7.gex 的计算表中，选择 A3，设定**删除**，执行唯一性操作，结果如下：

0	1		A	B	C
1-		1	Gymnastics Artistic Women's Individual All-Around Results		
2-		2	Name	Event	Score
	1	3	Daniel Smith	Parallel bars	14.375

可以看到，只有无重复值的记录被保留，有重复姓名的记录全部被删除。

重新打开 esCalc08_7.gex 计算表，选择 A3，设定**无动作**，同时选定**删除无同值**选项，执行唯一性操作后，结果如下：

0	1		A	B	C
1		1	Gymnastics Artistic Women's Individual All-Around Results		
2-		2	Name	Event	Score
	1	3	Grace Miller	Vault	14.000
	1	4	Kayla Rodriguez	Vault	14.575
	1	5	Kayla Rodriguez	Uneven bars	14.675
	1	6	Kayla Rodriguez	Balance beam	13.975
	1	7	Kayla Rodriguez	Floor	14.400
	1	8	Lauren Davis	Vault	14.275
	1	9	Grace Miller	Floor	13.975
	1	10	Grace Miller	Uneven bars	12.875
	1	11	Lauren Davis	Floor	13.000
	1	12	Grace Miller	Balance beam	13.100
	1	13	Lauren Davis	Balance beam	15.225
	1	14	Lauren Davis	Uneven bars	14.275

可以发现，唯一性操作的结果和刚刚的正好相反，只删除了无重复值的记录。选定了删除无同值后，在进行唯一性操作时，会删除无重复值的记录，也可以在设定合并或者去重的唯一性操作时使用。

重新打开计算表 esCalc08_7.gex，选择 A3，设定**无动作**，选定**删除无同值**选项，也选定**同值判断只处理相邻**，执行唯一性操作后，结果如下：

0	1		A	B	C
1		1	Gymnastics Artistic Women's Individual All-Around Results		
2-		2	Name	Event	Score
	1	3	Kayla Rodriguez	Vault	14.575
	1	4	Kayla Rodriguez	Uneven bars	14.675
	1	5	Kayla Rodriguez	Balance beam	13.975
	1	6	Kayla Rodriguez	Floor	14.4
	1	7	Grace Miller	Floor	13.975
	1	8	Grace Miller	Uneven bars	12.875
	1	9	Lauren Davis	Balance beam	15.225
	1	10	Lauren Davis	Uneven bars	14.275

从结果中可以看到，当选定了同值判断只处理相邻时，删除无同值选项在执行时会删除所有没有相邻同值记录的数据。

再打开计算表 esCalc08_8.gex，来看下面的网格：

0	1	2		A	B	C	D	E
1			1	States and Cities				
2-			2	State	Abbr.	City	Population	
	1-		3	California	CA			a
		1	4			Los Angeles	3849368	1
	1-		5	Texas	TX			b
		1	6			Houston	2144491	2
		1	7			San Antonio	1296682	3
	1-		8	California	CA			c
		1	9			San Diego	1256951	4
		1	10			San Jose	929936	5
	1-		11	Florida	FL			d
		1	12			Jacksonville	794555	6
	1-		13	Texas	TX			e
		1	14			Austin	709893	7

这个计算表中，存储了一些州和城市城市的信息，其中 E 列中对对每条记录做了标记以备区分。下面根据州的简称来进行唯一性操作。选择 B3，设定**合并**来进行唯一性操作，结果如下：

0	1	2		A	B	C	D	E
1			1	States and Cities				
2-			2	State	Abbr.	City	Population	
	1-		3	California	CA			a
		1	4			Los Angeles	3849368	1
		1	5			San Diego	1256951	4
		1	6			San Jose	929936	5
	1-		7	Texas	TX			b
		1	8			Houston	2144491	2
		1	9			San Antonio	1296682	3
		1	10			Austin	709893	7
	1-		11	Florida	FL			d
		1	12			Jacksonville	794555	6

从结果中可以看到。选择合并进行唯一性操作时，选定的记录和去重的操作类似，只会保留第一个，但是如果记录拥有子记录时，会将后面的子记录并入第一个记录中。

8.5 排序

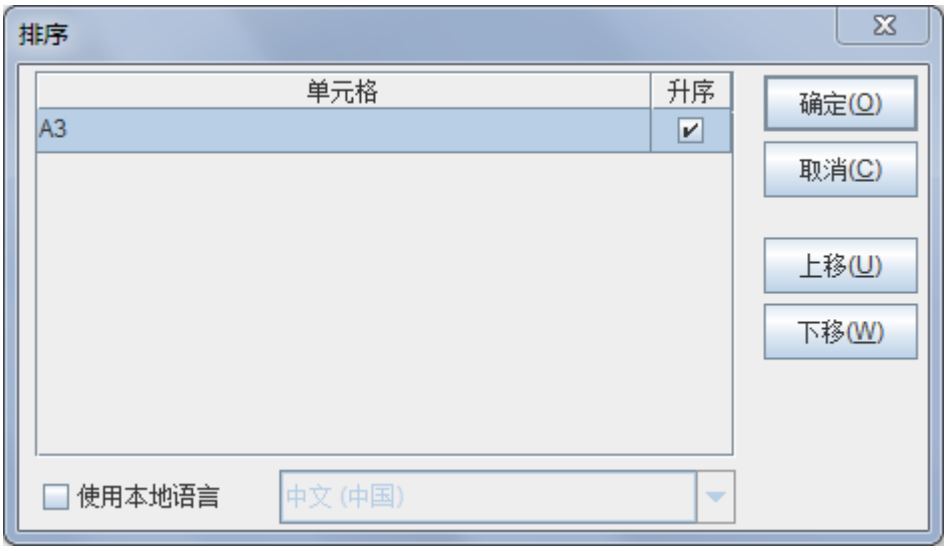
针对某些格值在父层内排序。

在右键菜单中点击**操作→排序**，可以执行排序操作。

我们再打开计算表 esCalc08_7.gex，使用体操成绩计算表：

0	1		A	B	C
1		1	Gymnastics Artistic Women's Individual All-Around Results		
2-		2	Name	Event	Score
	1	3	Grace Miller	Vault	14.000
	1	4	Kayla Rodriguez	Vault	14.575
	1	5	Kayla Rodriguez	Uneven bars	14.675
	1	6	Kayla Rodriguez	Balance beam	13.975
	1	7	Kayla Rodriguez	Floor	14.400
	1	8	Lauren Davis	Vault	14.275
	1	9	Grace Miller	Floor	13.975
	1	10	Grace Miller	Uneven bars	12.875
	1	11	Lauren Davis	Floor	13.000
	1	12	Grace Miller	Balance beam	13.100
	1	13	Daniel Smith	Parallel bars	14.375
	1	14	Lauren Davis	Balance beam	15.225
	1	15	Lauren Davis	Uneven bars	14.275

选择 A3，执行排序操作，会弹出排序操作面板如下：



在排序操作面板中，可以看到选定的排序单元格和是否升序。如果有多个排序条件时，可以通过上移和下移来调整排序条件的次序。

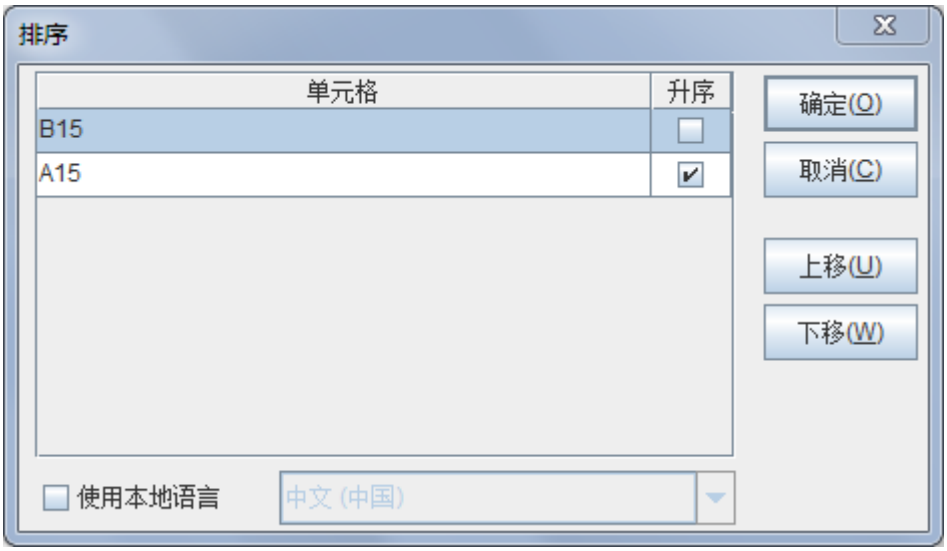
勾选使用本地语言，排序将按选定的本地语言排序。

按 A3 升序，执行排序操作，结果如下：

0	1		A	B	C
1		1	Gymnastics Artistic Women's Individual All-Around Results		
2-		2	Name	Event	Score
	1	3	Daniel Smith	Parallel bars	14.375
	1	4	Grace Miller	Vault	14.000
	1	5	Grace Miller	Floor	13.975
	1	6	Grace Miller	Uneven bars	12.875
	1	7	Grace Miller	Balance beam	13.100
	1	8	Kayla Rodriguez	Vault	14.575
	1	9	Kayla Rodriguez	Uneven bars	14.675
	1	10	Kayla Rodriguez	Balance beam	13.975
	1	11	Kayla Rodriguez	Floor	14.400
	1	12	Lauren Davis	Vault	14.275
	1	13	Lauren Davis	Floor	13.000
	1	14	Lauren Davis	Balance beam	15.225
	1	15	Lauren Davis	Uneven bars	14.275

可以看到，第 3 行的所有同位行，按照姓名字母顺序从小到大进行了排序，姓名相同的记录保持原来的顺序。排序时，选择任何一个同位行中相应的单元格，结果都是相同的。

排序操作可以选择同一区段中的多个同层单元格，并允许设定单元格排序的先后次序。在最初的计算表 esCalc08_7.gex 中，同时选择 A15 和 B15 进行排序，并通过上移或下移调整排序次序为先按 B15 降序，再按 A15 升序，调整后的排序操作面板如下：



排序完成后，结果如下：

0	1		A	B	C
1		1	Gymnastics Artistic Women's Individual All-Around Results		
2-		2	Name	Event	Score
	1	3	Grace Miller	Vault	14.000
	1	4	Kayla Rodriguez	Vault	14.575
	1	5	Lauren Davis	Vault	14.275
	1	6	Grace Miller	Uneven bars	12.875
	1	7	Kayla Rodriguez	Uneven bars	14.675
	1	8	Lauren Davis	Uneven bars	14.275
	1	9	Daniel Smith	Parallel bars	14.375
	1	10	Grace Miller	Floor	13.975
	1	11	Kayla Rodriguez	Floor	14.400
	1	12	Lauren Davis	Floor	13.000
	1	13	Grace Miller	Balance beam	13.100
	1	14	Kayla Rodriguez	Balance beam	13.975
	1	15	Lauren Davis	Balance beam	15.225

可以看到，第 3 行的所有同位行，先按照项目的字母顺序从大到小进行了排序，项目相同的记录，按照姓名的字母顺序从小到大进行了排序。

如果需要选择的多个单元格不是连续的单元格，可以按住 Ctrl 多选。

再打开计算表 esCalc08_8.gex：

0	1	2		A	B	C	D	E
1			1	States and Cities				
2-			2	State	Abbr.	City	Population	
	1-		3	California	CA			a
		1	4			Los Angeles	3849368	1
	1-		5	Texas	TX			b
		1	6			Houston	2144491	2
		1	7			San Antonio	1296682	3
	1-		8	California	CA			c
		1	9			San Diego	1256951	4
		1	10			San Jose	929936	5
	1-		11	Florida	FL			d
		1	12			Jacksonville	794555	6
	1-		13	Texas	TX			e
		1	14			Austin	709893	7

选择 B3，按照升序执行排序操作，结果如下：

0	1	2		A	B	C	D	E
1			1	States and Cities				
2-			2	State	Abbr.	City	Population	
	1-		3	California	CA			a
		1	4			Los Angeles	3849368	1
	1-		5	California	CA			c
		1	6			San Diego	1256951	4
		1	7			San Jose	929936	5
	1-		8	Florida	FL			d
		1	9			Jacksonville	794555	6
	1-		10	Texas	TX			b
		1	11			Houston	2144491	2
		1	12			San Antonio	1296682	3
	1-		13	Texas	TX			e
		1	14			Austin	709893	7

可以看到，当按照州简写的字母顺序从小到大排序时，它们各自的子记录会随着区段主行移动。

打开计算表 esCalc08_9.gex，再看下面的网格：

0	1	2		A	B	C	D	E
1			1	States and Cities				
2-			2	State	Abbr.	City	Population	
	1-		3	California	CA			a
		1	4			Los Angeles	3849368	1
		1	5			San Diego	1256951	4
		1	6			San Jose	929936	5
	1-		7	Texas	TX			b
		1	8			Houston	2144491	2
		1	9			San Antonio	1296682	3
		1	10			Austin	709893	7
	1-		11	Florida	FL			d
		1	12			Jacksonville	794555	6

这是将州记录进行合并唯一性操作后的网格，在这个网格中，选择 C4，设定降序，执行排序操作，结果如下：

0	1	2		A	B	C	D	E
1			1	States and Cities				
2-			2	State	Abbr.	City	Population	
	1-		3	California	CA			a
		1	4			San Jose	929936	5
		1	5			San Diego	1256951	4
		1	6			Los Angeles	3849368	1
	1-		7	Texas	TX			b
		1	8			San Antonio	1296682	3
		1	9			Houston	2144491	2
		1	10			Austin	709893	7
	1-		11	Florida	FL			d
		1	12			Jacksonville	794555	6

可以看到，在执行排序操作时，所有的同位行会同时执行，但是只会在原有的父层内进行排序，而不会改变父层。

8.6 分组

指定某格等值分组，新增分组段主格用该格值，操作区为 0 层。

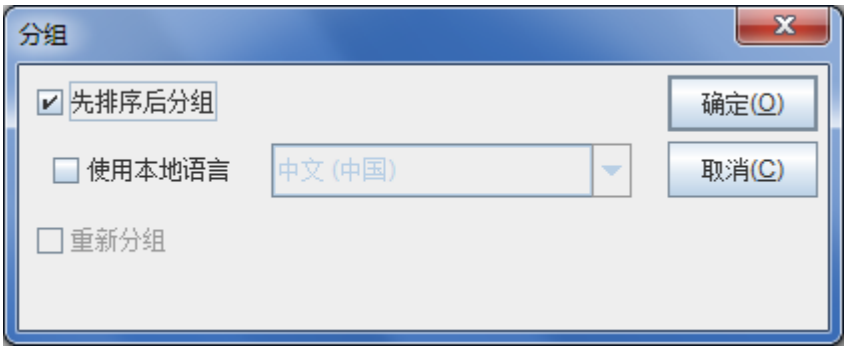
在右键菜单中点击**操作→分组**，可以执行分组操作。

打开计算表 esCalc08_10.gex，网格如下：

0	1		A	B	C	D
1		1	Liquors			
2-		2	Liquor	Type	Origin	
	1	3	Appleton Estate Reserve	Rum	Jamaica	
	1	4	Coruba Royal	Rum	Jamaica	
	1	5	Malibu Coconut	Rum	England	
	1	6	Bombay Sapphire	Gin	England	
	1	7	Seagers	Gin	England	
	1	8	Kahlua	Cordials	Mexico	
	1	9	Chatelle Brandy	Brandy	France	
	1	10	Conjure Cognac	Brandy	France	
	1	11	Grand Marnier Cordon Rouge	Cordials	France	
	1	12	Baileys Irish Cream	Cordials	Ireland	

网格中是一些酒类的信息，包括酒的名称，类型和产地。现在，准备根据类型将这些信息进行分组。

选择 B3，执行分组操作，弹出分组操作面板如下：



在分组操作面板中，可以选择是否**先排序后分组**，和是否**重新分组**。在选择先排序后分组时，可以设定在排序时**使用本地语言**。

先把先排序后分组勾掉，不选择，然后进行分组操作，结果如下：

0	1	2		A	B	C	D
1			1	Liquors			
2-			2	Liquor	Type	Origin	
	1-		3		Rum		
		1	4	Appleton Estate Reserve	Rum	Jamaica	
		1	5	Coruba Royal	Rum	Jamaica	
		1	6	Malibu Coconut	Rum	England	
	1-		7		Gin		
		1	8	Bombay Sapphire	Gin	England	
		1	9	Seagers	Gin	England	
	1-		10		Cordials		
		1	11	Kahlua	Cordials	Mexico	
	1-		12		Brandy		
		1	13	Chatelle Brandy	Brandy	France	
		1	14	Conjure Cognac	Brandy	France	
	1-		15		Cordials		
		1	16	Grand Marnier Cordon Rouge	Cordials	France	
		1	17	Baileys Irish Cream	Cordials	Ireland	

可以看到，各条同位记录按照指定单元格值进行了分组，把相邻指定单元格值相等的记录分到同一组中，并添加了一个分组层。分组层中的行都为新增加的行，各单元各种的属性均为初始属性。在分组行中与选定单元格同列的格子中，将依次设入分组值，同时这个单元格还会被设为主格。

选择任一个同位行中相应的单元格进行分组操作，结果是相同的。

如果在 esCalc08_10.gex 的计算表中，选择 B3，把**先排序后分组**选定，然后进行分组操作，结果如下：

0	1	2		A	B	C	D
1			1	Liquors			
2-			2	Liquor	Type	Origin	
	1-		3		Brandy		
		1	4	Chatelle Brandy	Brandy	France	
		1	5	Conjure Cognac	Brandy	France	
	1-		6		Cordials		
		1	7	Kahlua	Cordials	Mexico	
		1	8	Grand Marnier Cordon Rouge	Cordials	France	
		1	9	Baileys Irish Cream	Cordials	Ireland	
	1-		10		Gin		
		1	11	Bombay Sapphire	Gin	England	
		1	12	Seagers	Gin	England	
	1-		13		Rum		
		1	14	Appleton Estate Reserve	Rum	Jamaica	
		1	15	Coruba Royal	Rum	Jamaica	
		1	16	Malibu Coconut	Rum	England	

可以看到，选择了分组排序后，在分组操作前，会根据选定的单元格值，先对各个记录区段进行排序，然后再进行分组。需要注意的是，这次分组是在最初的计算表中进行的。

我们修改一下上面的网格中的分组行，如下：

0	1	2		A	B	C	D
1			1	Liquors			
2-			2	Liquor	Type	Origin	
	1-		3	Brandy	Count	2	
		1	4	Chatelle Brandy	Brandy	France	
		1	5	Conjure Cognac	Brandy	France	
	1-		6	Cordials	Count	3	
		1	7	Kahlua	Cordials	Mexico	
		1	8	Grand Marnier Cordon Rouge	Cordials	France	
		1	9	Baileys Irish Cream	Cordials	Ireland	
	1-		10	Gin	Count	2	
		1	11	Bombay Sapphire	Gin	England	
		1	12	Seagers	Gin	England	
	1-		13	Rum	Count	3	
		1	14	Appleton Estate Reserve	Rum	Jamaica	
		1	15	Coruba Royal	Rum	Jamaica	
		1	16	Malibu Coconut	Rum	England	

将个分组行中的主格改为了 A3 及它的同位格，并在 C3 中添加了计算式==\{B4\}.count()，来计算每个分组中的记录数，在 B3 和它的同位格中填入了常数字符串 Count。

现在选择 C4，准备按照产地重新进行分组，为此，在分组操作面板中，同时选中先排序后分组

和重新分组，分组操作后，结果如下：

0	1	2		A	B	C	D
1			1	Liquors			
2-			2	Liquor	Type	Origin	
	1-		3			3	
		1	4	Bombay Sapphire	Gin	England	
		1	5	Seagers	Gin	England	
		1	6	Malibu Coconut	Rum	England	
	1-		7			3	
		1	8	Chatelle Brandy	Brandy	France	
		1	9	Conjure Cognac	Brandy	France	
		1	10	Grand Marnier Cordon Rouge	Cordials	France	
	1-		11			1	
		1	12	Baileys Irish Cream	Cordials	Ireland	
	1-		13			2	
		1	14	Appleton Estate Reserve	Rum	Jamaica	
		1	15	Coruba Royal	Rum	Jamaica	
	1-		16			1	
		1	17	Kahlua	Cordials	Mexico	

从结果中可以看到，选中单元格所在的记录的所有同位记录区段，都进行了重新整理，即使它们的父层并不是同一行。所有的记录排序后，按照产地进行了重新分组。值得注意的是，在结果网格新的父层行中，各个单元格的属性和原来的父层是相同的，如前景色，背景色等等，而且主格的设定也没有改变。在新的父层行中，C3 中的计算式仍然保留，它和它的同位格仍然计算每个分组中的记录数。另外，新的父层行中，所有单元格中的格值都被清空，联动计算格或者计算格中的格值被重新计算。

值得注意的是，与前面的分组操作不同，重新分组时，每一组的分组值没有被设入父层行中。如果重新分组时，也类似直接分组时设定分组值，那么在这个例子中，C3 中的计算式就会被分组值覆盖，原父层行中的属性就被破坏了。为了避免改变原父层行中的属性和计算式等被改变，重新分组时将不会自动设定父层行中的分组值，也不会改变原来父层行中的主格设定，需要手动修改。直接分组时，由于父层行都是新加入的行，自动设定主格和分组值就不会产生同样的问题。

来看下面一个问题：某个公司需要在一周中安排员工轮流休假一天，每位员工的休假日记录在 esCalc08_11.gex 的计算表中：

0	1			A	B	C	D
1			1	Off Day			
2-			2	Name	Gender	Off Day	
	1		3	Harry Potter	Male	Monday	
	1		4	Ronald Weasley	Male	Tuesday	
	1		5	Hermione Granger	Female	Thursday	
	1		6	Lord Voldemort	Male	Wednesday	
	1		7	Albus Dumbledore	Male	Wednesday	
	1		8	Ginevra Weasley	Female	Monday	
	1		9	Rubeus Hagrid	Male	Tuesday	
	1		10	Luna Lovegood	Female	Monday	
	1		11	Severus Snape	Male	Wednesday	
	1		12	Draco Malfoy	Male	Tuesday	

现在按照休息日将员工信息分组，我们选择 C3，设定先排序后分组，进行分组操作，结果如下：

0	1	2		A	B	C	D
1			1	Off Day			
2-			2	Name	Gender	Off Day	
	1-		3			Monday	
		1	4	Harry Potter	Male	Monday	
		1	5	Ginevra Weasley	Female	Monday	
		1	6	Luna Lovegood	Female	Monday	
	1-		7			Thursday	
		1	8	Hermione Granger	Female	Thursday	
	1-		9			Tuesday	
		1	10	Ronald Weasley	Male	Tuesday	
		1	11	Rubeus Hagrid	Male	Tuesday	
		1	12	Draco Malfoy	Male	Tuesday	
	1-		13			Wednesday	
		1	14	Lord Voldemort	Male	Wednesday	
		1	15	Albus Dumbledore	Male	Wednesday	
		1	16	Severus Snape	Male	Wednesday	

如果再按照性别分组，又会如何呢？选择 B4，设定先排序后分组，进行分组操作，结果如下：

0	1	2	3		A	B	C	D
1				1	Off Day			
2-				2	Name	Gender	Off Day	
	1-			3			Monday	
		1-		4		Female		
			1	5	Ginevra Weasley	Female	Monday	
			1	6	Luna Lovegood	Female	Monday	
		1-		7		Male		
			1	8	Harry Potter	Male	Monday	
	1-			9			Thursday	
		1-		10		Female		
			1	11	Hermione Granger	Female	Thursday	
	1-			12			Tuesday	
		1-		13		Male		
			1	14	Ronald Weasley	Male	Tuesday	
			1	15	Rubeus Hagrid	Male	Tuesday	
			1	16	Draco Malfoy	Male	Tuesday	
	1-			17			Wednesday	
		1-		18		Male		
			1	19	Lord Voldemort	Male	Wednesday	
			1	20	Albus Dumbledore	Male	Wednesday	
			1	21	Severus Snape	Male	Wednesday	

从结果中可以看到，如果执行分组的同位记录区段已经进行了分组，处于多个父层行下，那么在执行分组操作时，会在每个组中再次进行分组。

8.7 对位

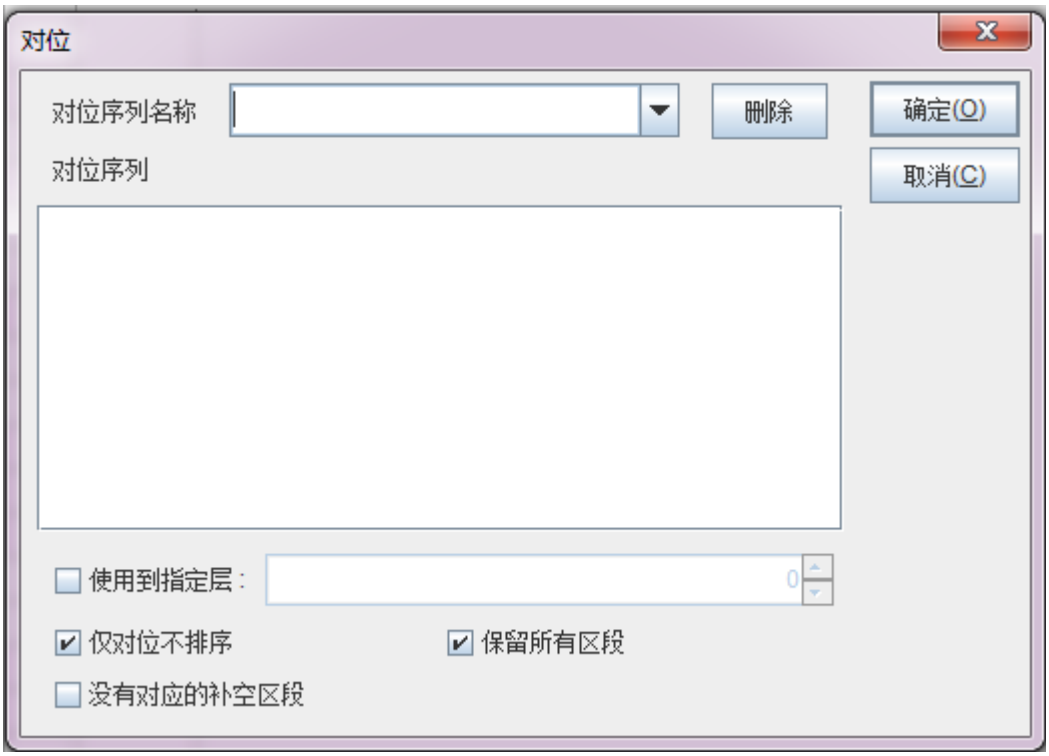
针对某个格值按指定常数序列对位，多层序列时用主格值分类。

在右键菜单中点击**操作→对位**，可以执行对位操作，大致步骤如下：

1) 选中某一个排序单元格，右键→**操作**，选择**对位**
2) 选择已有的对位序列，或者新建对位序列
3) 确定，即可完成对位操作

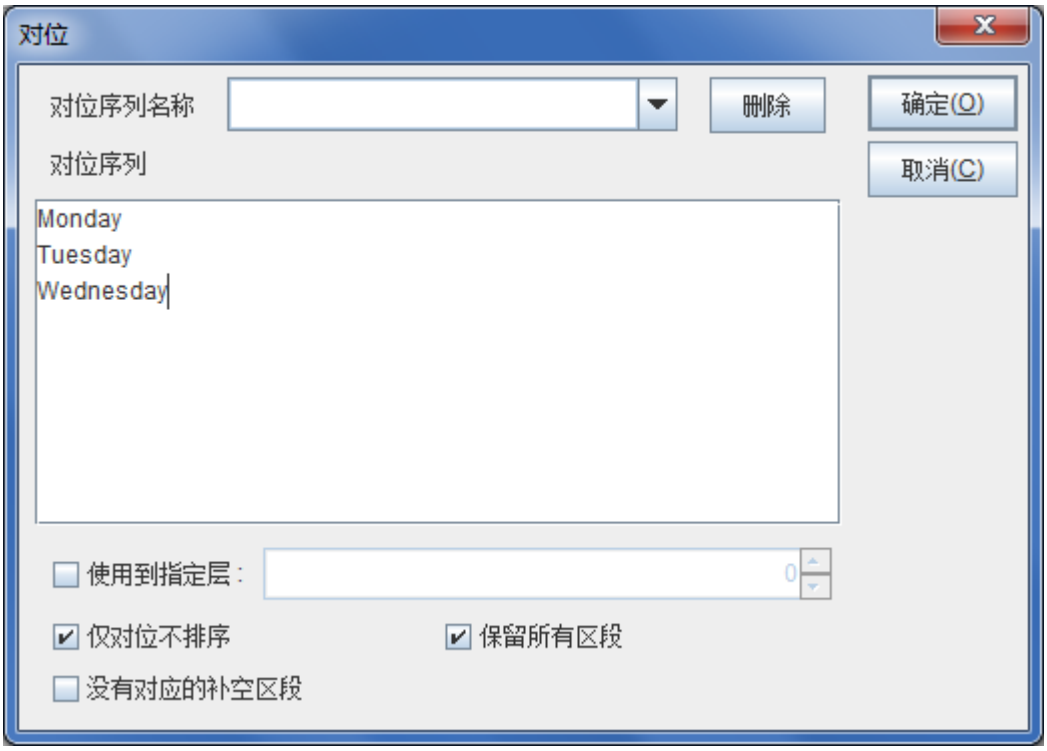
从上一节中，根据休息日对员工进行分组时，我们可以注意到一个问题，在通常的分组排序中，由于只能对分组字段进行字母排序，这样得到的分组顺序是 Monday,Thursday,Tuesday,Wednesday，这样的顺序往往并不是所需要的，为此在计算表中，提供了对位操作，可以按照指定顺序对记录区段进行排序。

打开在 **8.6 分组**中，最后完成了两层分组的计算表，或者直接打开计算表 esCalc08_12.gex，选择 C3，执行对位操作，弹出对位操作面板如下：



对位操作面板中，可以设定对位序列，并可以为对位序列设定名称，或者按名称获取使用过的对位序列。在对位操作中，还可以设定 3 个操作选项：是否按对位序列排序，是否保留所有记录，是否用对位序列补空。在设定多层的对位序列时，还可以选择在对位操作时使用对位序列的最低层次。

我们先按照默认设置，选定按对位序列名称和保留所有区段，并将对位序列设定为[Monday,Tuesday,Wednesday]，如下：

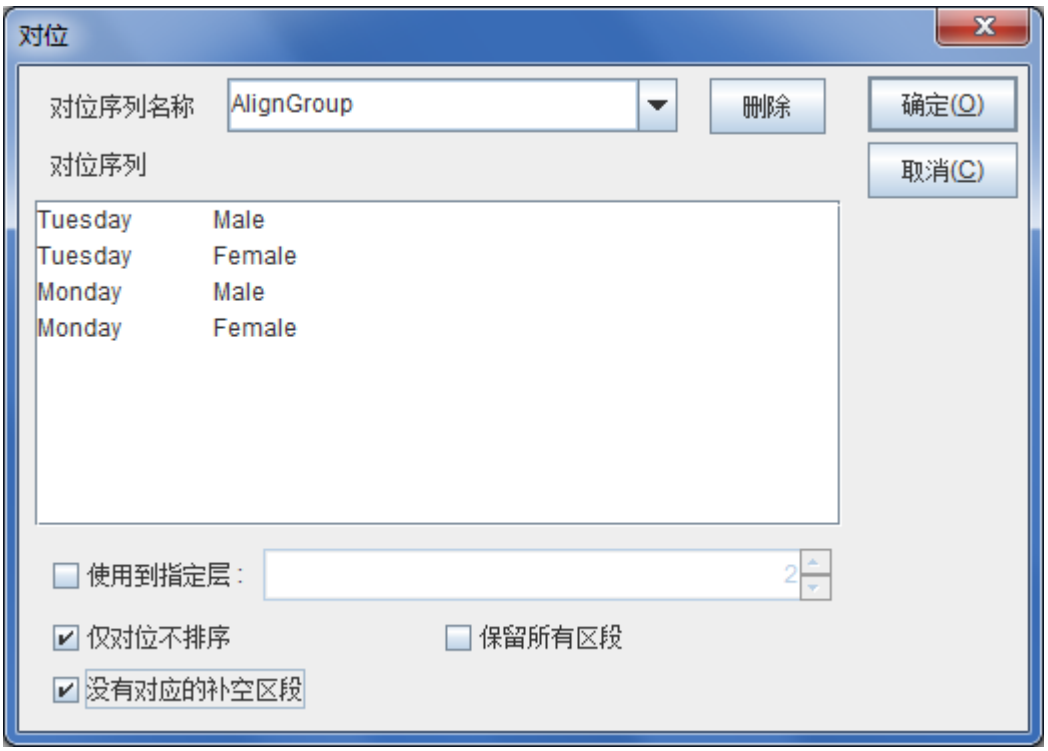


定义对位序列时，每个成员占一行。执行排序操作后，结果如下：

0	1	2	3		A	B	C	D
1				1	Off Day			
2-				2	Name	Gender	Off Day	
	1-			3			Monday	
		1-		4		Female		
			1	5	Ginevra Weasley	Female	Monday	
			1	6	Luna Lovegood	Female	Monday	
		1-		7		Male		
			1	8	Harry Potter	Male	Monday	
	1-			9			Tuesday	
		1-		10		Male		
			1	11	Ronald Weasley	Male	Tuesday	
			1	12	Rubeus Hagrid	Male	Tuesday	
			1	13	Draco Malfoy	Male	Tuesday	
	1-			14			Wednesday	
		1-		15		Male		
			1	16	Lord Voldemort	Male	Wednesday	
			1	17	Albus Dumbledore	Male	Wednesday	
			1	18	Severus Snape	Male	Wednesday	
	1-			19			Thursday	
		1-		20		Female		
			1	21	Hermione Granger	Female	Thursday	

设定了按照对位序列排序后，选定层的记录区段将按照对位序列排序，如果记录区段中包含子记录，将随之移动。当选定保留所有记录时，选定单元格值不在对位序列中的记录区段，会置于最后，如果不选定**保留所有记录**，这些记录区段会被删除。

我们再看看使用多层对位序列的情况，选择 B4 执行对位操作，将排位序列定义为[[Tuesday,Male],[Tuesday,Female],[Monday,Male],[Monday,Female]]，选定按对位序列名称和没有对应的补空区段，并将对位序列设定为[Monday,Tuesday,Wednesday]，如下：



定义多层对位序列时，两个层次的设定值之间用 Tab 分隔，每个成员占一行，我们给这个对位序列命名为 AlignGroup。执行排序操作后，结果如下：

0	1	2	3		A	B	C	D
1				1	Off Day			
2-				2	Name	Gender	Off Day	
	1-			3			Tuesday	
		1-		4		Male		
			1	5	Ronald Weasley	Male	Tuesday	
			1	6	Rubeus Hagrid	Male	Tuesday	
				7	Draco Malfoy	Male	Tuesday	
		1-		8		Female		
			1	9				
	1-			10			Monday	
		1-		11		Male		
			1	12	Harry Potter	Male	Monday	
		1-		13		Female		
				14	Ginevra Weasley	Female	Monday	
				15	Luna Lovegood	Female	Monday	

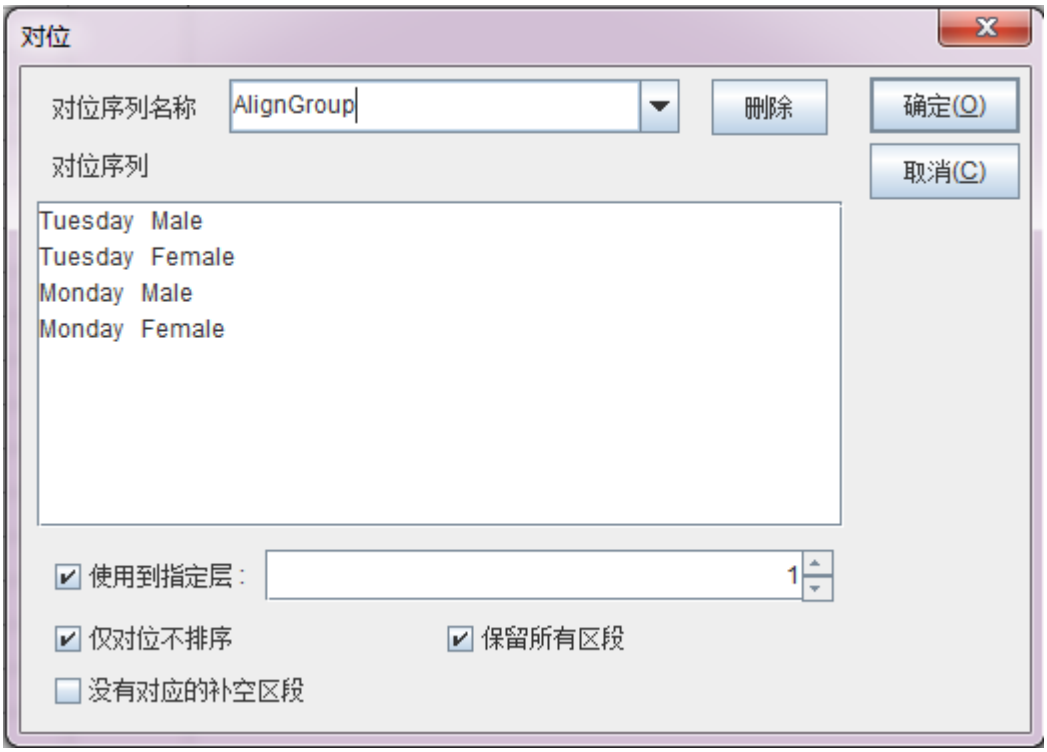
从结果中可以看出，记录区段按照对位序列中的设定进行了多层对位排序，在多层对位时，选定的单元格对应对位序列中最后的层次，它的父层会依次对应对位序列中之前的层次。父层中进行对位操作所依据的单元格，是各级父层的主格。

在未选择保留所有记录的情况下，父层和当前单元格的值不在记录区段的设定中的记录区段都被删除。由于选择了用对位序列补空选项，因此，在结果序表中填补了 Tuesday→Female 的空分组，并在其中建立了一条空记录。

重新打开计算表 esCalc08_11.gex，网格如下：

0	1		A	B	C	D
1		1	Off Day			
2-		2	Name	Gender	Off Day	
	1	3	Harry Potter	Male	Monday	
	1	4	Ronald Weasley	Male	Tuesday	
	1	5	Hermione Granger	Female	Thursday	
	1	6	Lord Voldemort	Male	Wednesday	
	1	7	Albus Dumbledore	Male	Wednesday	
	1	8	Ginevra Weasley	Female	Monday	
	1	9	Rubeus Hagrid	Male	Tuesday	
	1	10	Luna Lovegood	Female	Monday	
	1	11	Severus Snape	Male	Wednesday	
	1	12	Draco Malfoy	Male	Tuesday	

选择 C3，执行对位操作，在对位序列名称的下拉列表中，可以选择 AlignGroup，使用存储好的序列。由于这个是个两层的序列，现在只想用序列中第一层的星期几的值，那么可以将对位序列最低层次中设为 1，选定按对位序列名称，如下：



执行对位操作后，结果如下：

	1		A	B	C	D
1		1	Off Day			
2-		2	Name	Gender	Off Day	
	1	3	Ronald Weasley	Male	Tuesday	
	1	4	Rubeus Hagrid	Male	Tuesday	
	1	5	Draco Malfoy	Male	Tuesday	
	1	6	Harry Potter	Male	Monday	
	1	7	Ginevra Weasley	Female	Monday	
	1	8	Luna Lovegood	Female	Monday	
	1	9	Hermione Granger	Female	Thursday	
	1	10	Lord Voldemort	Male	Wednesday	
	1	11	Albus Dumbledore	Male	Wednesday	
	1	12	Severus Snape	Male	Wednesday	

对位操作经常和分组操作一起使用，将记录按指定顺序排序后分组，或者将分组后的父层区段按照指定顺序排序。

8.8 连接

包括主格在内的若干同层格的同位格值，与目标记录的主格对应连接，填入目标记录的指定格区域；用双方父层的主格值分类。

在右键菜单中点击**操作→连接**，可以执行连接操作。大致步骤如下：

1） 在源网格中，选定操作区，选择同区段内同层连续单元格，复制
2) 在目标网格中，选中与其连接的主格所在行的任意单元格，执行连接

如此，源网格中选中单元格对应的主格与目标网格中选中单元格的主格匹配连接，从源网格复制的数据从选定的目标格依次粘贴。

进行连接操作时，选中格的主格进行连接匹配的同时，其父格的主格也会同步进行匹配。系统会自动在源网格的操作范围内，从选中格所在行开始，取得其主格，以及以上所有层的主格（除 0 层外），然后与目标网格的各层主格匹配连接。因此除 0 层外，连接时，不管是源网格还是目标网格，各层都需要设置主格， 目标网格参与连接的区域以源网格操作区的层数为准，假设源网格操作区从选中格开始向上一共有 n 层，则目标网格就在选中目标格开始数向上第 n 层的父层行区域范围内参与连接。

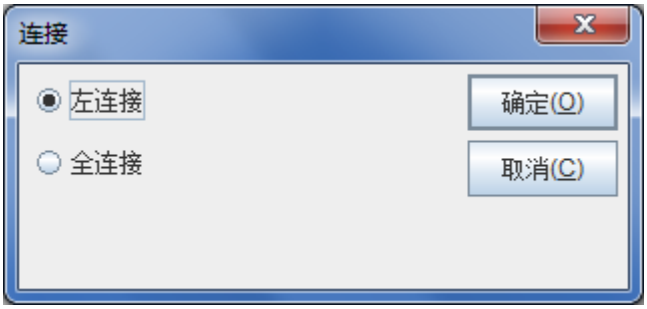
下面，我们来看两个网格。计算表 esCalc08_13.gex 中记录了州和城市的一些信息，其中州的简称和城市名称为各层主格，如下

0	1	2		A	B	C	D	E
1-			1	States and Cities				
	1-		2	CA	California	37253956	Sacramento	
		1	3	Los Angeles	3849368			
		1	4	San Diego	1256951			
		1	5	San Jose	929936			
	1-		6	TX	Texas	25145561	Austin	
		1	7	Houston	2144491			
		1	8	San Antonio	1296682			
		1	9	Austin	709893			
	1-		10	FL	Florida	18801310	Tallahassee	
		1	11	Jacksonville	794555			

计算表 esCalc08_14.gex 中是一些州中旅游景点的信息，按照所在州进行了分组，州的简称为主格。如下所示：

0	1	2		A	B	C	D
1-			1	Tourist Attractions			
	1-		2	CA			
		1	3		Hollywood		
		1	4		Disneyland		
		1	5		Seaworld		
	1-		6	HI			
		1	7		Waikiki Beach		
		1	8		Hanauma Bay		
	1-		9	TX			
		1	10		Hermann Park		
		1	11		Mountain Bonnell		

现在看看连接是如何执行的，我们准备将 esCalc08_13.gex 中的州信息，根据州的简称，对应粘贴到计算表 esCalc08_14.gex 中。首先在 esCalc08_13.gex 中选择所要复制的信息，同时选择 B2,C2 两个单元格，按下 Ctrl+C 或者在右键菜单中选择复制，将州的名称，人口信息进行复制。然后在 esCalc08_14.gex 中，选择 C2 格，执行连接操作，弹出连接操作面板如下：



在连接操作面板中，可以选择连接方式是左连接或者是全连接。

在这里选择默认的左连接，执行连接操作，结果如下：

0	1	2		A	B	C	D
1-			1	Tourist Attractions			
	1-		2	CA		California	37253956
		1	3		Hollywood		
		1	4		Disneyland		
		1	5		Seaworld		
	1-		6	HI			
		1	7		Waikiki Beach		
		1	8		Hanauma Bay		
	1-		9	TX		Texas	25145561
		1	10		Hermann Park		
		1	11		Mountain Bonnell		

从结果中可以看到，通过将两个计算表中的州简称进行等值连接，将第一个网格中的州信息复制到了第二个网格中，复制时从选中执行连接操作的单元格开始，将复制的单元格值依次粘贴。粘贴时，第二个网格中，各个单元格的前景色、背景色等属性都没有改变。选择左连接时，第一个网格中的佛罗里达州在第二个网格中没有对应数据，因此未被粘贴；而第二个网格中的夏威夷州无法获得相应州信息。

下面再重新进行一次连接操作，在计算表 esCalc08_13.gex 中，同时选择 B2,C2,D2 三个单元格，按下 Ctrl+C 或者在右键菜单中选择复制，将州的名称，人口，首府信息进行复制。然后在计算表 esCalc08_14.gex 中，选择 A2 格，执行连接操作，在连接操作面板中，选择全连接，执行连接操作后，结果如下：

0	1	2		A	B	C	D
1-			1	Tourist Attractions			
	1-		2	California	37253956	Sacramento	
		1	3		Hollywood		
		1	4		Disneyland		
		1	5		Seaworld		
	1-		6	HI			
		1	7		Waikiki Beach		
		1	8		Hanauma Bay		
	1-		9	Texas	25145561	Austin	
		1	10		Hermann Park		
		1	11		Mountain Bonnell		
	1-		12	Florida	18801310	Tallahassee	
		1	13				

从结果中可以看到，在执行连接操作时，在将复制的信息连接复制入目标网格中时，从选定格开始会依次粘贴，即使是主格中的原有数据也会被覆盖。全连接时，第一个网格中的佛罗里达州在第二个网格中没有对应数据，也会在相应区段最后增加一个空区段将州信息进行粘贴；而第二个网格中的夏威夷州仍然无法获得相应州信息。

再来看一下多层连接时的情况，计算表 esCalc08_15.gex 中，将旅游景点名称按照所在城市和州

进行了两层分组，州简称和城市名为各层主格，如下所示：

0	1	2	3		A	B	C	D
1-				1	Tourist Attractions			
	1-			2	CA			
		1-		3	Los Angeles			
			1	4		Hollywood		
			1	5		Disneyland		
		1-		6	San Jose			
			1	7		Seaworld		
	1-			8	HI			
		1-		9	Honolulu			
			1	10		Waikiki Beach		
			1	11		Hanauma Bay		
	1-			12	TX			
		1-		13	Houston			
			1	14		Hermann Park		
		1-		15	Austin			
			1	16		Mountain Bonnell		

由于这个网格中，旅游景点拥有城市和州两层主格，因此连接操作时，所复制的单元格也要有对应的父层。为此，在计算表 esCalc08_13.gex 中，同时选择 A3 和 B3，复制城市名称和人口数据，然后在计算表 esCalc08_15.gex 中，选择 C3，执行连接操作，如果选择左连接，结果如下：

0	1	2	3		A	B	C	D
1-				1	Tourist Attractions			
	1-			2	CA			
		1-		3	Los Angeles		Los Angeles	3849368
			1	4		Hollywood		
			1	5		Disneyland		
		1-		6	San Jose		San Jose	929936
			1	7		Seaworld		
	1-			8	HI			
		1-		9	Honolulu			
			1	10		Waikiki Beach		
			1	11		Hanauma Bay		
	1-			12	TX			
		1-		13	Houston		Houston	2144491
			1	14		Hermann Park		
		1-		15	Austin		Austin	709893
			1	16		Mountain Bonnell		

可以看到，复制的城市信息，按照相应的州和城市层次，粘贴到了旅游信息网格中，各州记录

中指定的位置。

如果在刚刚的连接操作时，选择的是全连接，则结果如下：

0	1	2	3		A	B	C	D
1-				1	Tourist Attractions			
	1-			2	CA			
		1-		3	Los Angeles		Los Angeles	3849368
			1	4		Hollywood		
			1	5		Disneyland		
		1-		6	San Jose		San Jose	929936
			1	7		Seaworld		
		1-		8	San Diego		San Diego	1256951
			1	9				
	1-			10	HI			
		1-		11	Honolulu			
			1	12		Waikiki Beach		
			1	13		Hanauma Bay		
	1-			14	TX			
		1-		15	Houston		Houston	2144491
			1	16		Hermann Park		
		1-		17	Austin		Austin	709893
			1	18		Mountain Bonnell		
		1-		19	San Antonio		San Antonio	1296682
			1	20				
	1-			21	FL			
		1-		22	Jacksonville		Jacksonville	794555
			1	23				

和单层连接时的情况雷同，全连接时，第一个网格中的所有城市和所有的州的信息都会被粘贴，如果在目标网格中没有对应数据，会在相应父层行区段的最后增加一个空记录区段将信息粘贴，如果没有相应父行，会连同父层区段一同创建。在新建区段的各层主格中，将设入相应的主格值。

在一个网格的不同分片中也可以进行连接，和两个网格中的执行操作类似。

8.9 合并

选择记录，复制到同构（包括子区段）目标记录区域，用主格值对应合并，按双方父层的主格值分组。

在右键菜单中点击**操作→合并**，可以执行合并操作。

任选一单元格，复制。在目标网格中选择同构单元格右键**操作→合并**。

大致步骤如下：

1) 在源网格中，选择单元格，复制。系统将用最上层的主格值与目标网格对应合并。
2) 在目标网格中，选中与其合并的主格所在行的任意单元格，执行合并

选中格的主格进行合并匹配时，其父格的主格也会同步进行匹配。系统会自动在源网格复制区

域最上层所在行开始，取得其主格，以及以上所有层的主格（除 0 层外），然后与目标网格的各层主格匹配连接。因此除 0 层外，合并时，不管是源网格还是目标网格，各层都需要设置主格， 目标网格参与合并的区域以源网格操作区的层数为准，假设源网格操作区从复制格开始向上一共有 n 层，则目标网格就在选中目标格开始数向上第 n 层的父层行区域范围内参与合并。如下面的两个州和城市信息网格：

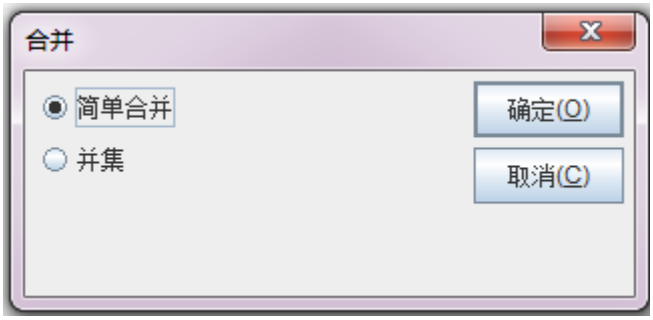
州和城市信息 1 存储在计算表 esCalc08_16.gex 中，网格如下，其中州简称和城市名为各层主格：

0	1	2		A	B	C	D	E
1-			1	States and Cities 1				
	1-		2	CA	California	37253956	Sacramento	
		1	3	Los Angeles	3849368			
		1	4	San Diego	1256951			
		1	5	San Jose	929936			
	1-		6	TX	Texas	25145561	Austin	
		1	7	Houston	2144491			
		1	8	San Antonio	1296682			
		1	9	Austin	709893			
	1-		10	FL	Florida	18801310	Tallahassee	
		1	11	Jacksonville	794555			

州和城市信息 2 存储在计算表 esCalc08_17.gex 中，网格如下，其中州简称和城市名为各层主格：

0	1	2		A	B	C	D	E
1			1	States and Cities 2				
2-			2	State	Abbr.	Population	Area	Capital
	1-		3	Texas	TX	25145561	268820	Austin
		1	4		El Paso	609415		
		1	5		Dallas	1232940		
	1-		6	Illinois	IL	12830632	54826	Springfield
		1	7		Chicago	2873326		
	1-		8	California	CA	37253956	163700	Sacramento
		1	9		San Jose	929936		
		1	10		San Francisco	744041		
		1	11		San Diego	1256951		

现在准备将第一个网格中的州信息合并到第二个网格中。为此，在州和城市信息 1 这个计算表中，在州记录行中任意选择一个单元格，如 B6，执行复制；在州和城市信息 2 这个计算表中，同样在州记录行中任意选择一个单元格，如 E8，进行合并操作，弹出合并操作面板如下：



在合并操作面板中，可以选择方式是简单合并或者并集。

选择**并集**方式，执行合并操作，在这种模式下，重复的州信息不会再次出现，操作结果如下：

0	1	2		A	B	C	D	E
1			1	States and Cities 2				
2-			2	State	Abbr.	Population	Area	Capital
	1-		3	Texas	TX	25145561	268820	Austin
		1	4		El Paso	609415		
		1	5		Dallas	1232940		
	1-		6	Illinois	IL	12830632	54826	Springfield
		1	7		Chicago	2873326		
	1-		8	California	CA	37253956	163700	Sacramento
		1	9		San Jose	929936		
		1	10		San Francisco	744041		
		1	11		San Diego	1256951		
	1-		12	FL	Florida	18801310	Tallahassee	
		1	13	Jacksonville	794555			

从结果可以看到，合并模式下的并集操作，会将网格 1 中的州信息合并到网格 2 中，对于网格 2 中已存在的州记录没有任何影响，如 Texas 州中的城市记录未受影响；而网格 2 中不存在的州记录将会在目标区段最后，连同整个区段一起粘贴进目标网格中，如 Florida 州的信息。而在粘贴时，会按照位置依次粘贴，每个单元格中的属性并不会复制过来，而是使用目标网格中原有设定。在判断记录是否已存在时，是根据主格中的数据进行的，即使网格 1 和网格 2 中的主格州缩写并不在同一列。

如果在合并操作时，我们在第 1 个网格中复制了 B3，并选择了**简单合并**选项，那么在网格 1 中所选单元格之后的州的记录都会复制到网格 2 中目标区段的最后面，结果如下所示：

0	1	2		A	B	C	D	E
1			1	States and Cities 2				
2-			2	State	Abbr.	Population	Area	Capital
	1-		3	Texas	TX	25145561	268820	Austin
		1	4		El Paso	609415		
		1	5		Dallas	1232940		
	1-		6	Illinois	IL	12830632	54826	Springfield
		1	7		Chicago	2873326		
	1-		8	California	CA	37253956	163700	Sacramento
		1	9		San Jose	929936		
		1	10		San Francisco	744041		
		1	11		San Diego	1256951		
	1-		12	CA	California	37253956	Sacramento	
		1	13	Los Angeles	3849368			
		1	14	San Diego	1256951			
		1	15	San Jose	929936			
	1-		16	TX	Texas	25145561	Austin	
		1	17	Houston	2144491			
		1	18	San Antonio	1296682			
		1	19	Austin	709893			
	1-		20	FL	Florida	18801310	Tallahassee	
		1	21	Jacksonville	794555			

从结果可以看到，简单合并模式下的合并操作，会将网格 1 中的州信息复制，直接在目标区段最后，连同整个区段一起粘贴进目标网格中。在简单合并模式下，复制的最底层记录中可以不设定主格。

如果将网格 1 中的城市信息，合并到最初的网格 2 中，又如何呢？为此，在 esCalc08_16.gex 网格中，选择城市记录中的任一个单元格，如 B7，执行复制，然后在最初的 esCalc08_17.gex 网格中，同样选择城市记录中的任一个单元格，如 A4，执行合并操作，选择并集模式，执行结果如下：

0	1	2		A	B	C	D	E
1			1	States and Cities 2				
2-			2	State	Abbr.	Population	Area	Capital
	1-		3	Texas	TX	25145561	268820	Austin
		1	4		El Paso	609415		
		1	5		Dallas	1232940		
		1	6	Houston	2144491			
		1	7	San Antonio	1296682			
		1	8	Austin	709893			
	1-		9	Illinois	IL	12830632	54826	Springfield
		1	10		Chicago	2873326		
	1-		11	California	CA	37253956	163700	Sacramento
		1	12		San Jose	929936		
		1	13		San Francisco	744041		
		1	14		San Diego	1256951		
		1	15	Los Angeles	3849368			
	1-		16		FL			
		1	17	Jacksonville	794555			

从结果可以看到，合并时，会根据多层主格中的数据来进行，而父层行中的数据仅设入了主格的数据。

8.10 附加

将某记录的连续续行及其属行粘贴到网中最后成为新的 0 层行。

在右键菜单中点击**操作→附加**，可以执行附加操作。

重新打开计算表 esCalc03_1.gex，我们再来看看下面的网格：

0	1	2		A	B	C	D	E
1-			1	Employee				
	1		2	Department	Finance			
	2-		3	Female				
		1	4	Ashley Wilson	NY	11000		
	3-		5	Male				
		1	6	Daniel Davis	FL	10000		
	4		7	Count	2	Sum	21000	
	1		8	Department	R&D			
	2		9	Female				
	3-		10	Male				
		1	11	Justin Smith	TX	7000		
		1	12	Jacob Davis	TX	16000		
	4		13	Count	2	Sum	23000	
2			14	Count	4	Sum	44000	

可以在某个区段中选择 1 个续行或者多个连续的续行所在的单元格，执行附加操作，将把选中的连续续行及其属行中的数据，附加到报表 0 层行的最后。特别的，如果报表的最后一行是 0 层行，将保留最后一个 0 层行，而把附加操作加入的各行放在最后一行之前。

比如，选择 B3 到 B5 这 3 个单元格执行附加操作。此时选中的连续续行是第 3 行和第 5 行，它们及其续行的数据就是第 3,4,5,6 行这 4 行数据，执行附加操作后，结果如下：

0	1	2		A	B	C	D	E
1-			1	Employee				
	1		2	Department	Finance			
	2-		3	Female				
		1	4	Ashley Wilson	NY	11000		
	3-		5	Male				
		1	6	Daniel Davis	FL	10000		
	4		7	Count	2	Sum	21000	
	1		8	Department	R&D			
	2		9	Female				
	3-		10	Male				
		1	11	Justin Smith	TX	7000		
		1	12	Jacob Davis	TX	16000		
	4		13	Count	2	Sum	23000	
2-			14	Female				
	1		15	Ashley Wilson	NY	11000		
3-			16	Male				
	1		17	Daniel Davis	FL	10000		
4			18	Count	4	Sum	44000	

可以看到，由于原网格中最后一行是 0 层行，因此附加的各行插入到了最后一行之前，并加入

到了 0 层区段中，选中的最上层续行被复制为了 0 层的新续行。

8.11 快速数据操作

计算表中，为了便于进行类似的数据操作，在右键菜单中提供了一组快速数据操作：

在快捷数据操作中，可以选择**过滤**，**隐藏**，**取消隐藏**，**升序**，**降序**，**分组**，**分组不排序**进行相应的操作。

当选择快捷数据操作时，将使用上一次相似操作时的设定：过滤、隐藏、取消隐藏将使用上一次过滤操作时的设定，分组将使用上一次分组操作时的设定。在快捷数据操作中选择升序排序或者降序排序时，将针对当前格进行升序或降序的排序操作。

9. 注册与加密

9.1 注册计算表

默认的安装程序是计算表免费版本，如果需要运行计算表其他版本，需要设置注册码。各版本之间的区别及如何获得注册码，请参阅官方网站。

获取注册码后，则可以在计算表中进行注册，操作步骤如下：

1) 执行帮助菜单中的注册按钮，打开程序选项对象框。

2) 输入注册码之后，点击**查看**按钮检查注册码是否正确，如果注册码无误，则可以点**注册**进行注册。

当注册码为空时，同样可以扫描注册信息并进行注册，此时将注册为免费版。

9.2 网格描述

在计算表中，可以在网格文件中设置网格描述信息。

在菜单栏的**工具**选项中，点击**网格描述**，可以弹出网格描述窗口，在窗口中查看或者设置网格描述。

9.3 文件加密

9.3.1 权限

为防止未经许可查看或访问网格文件，计算表开发了网格文件加密，加密有如下 4 级权限：

查看权：这是计算表中的最低权限，可以查看网格描述，但不能修改这些信息。在拥有查看权限时，可以计算表格，查看各个单元格中的结果，以及各个单元格的展现样式；但是无法看到单元格中的任何属性。

填写权：这是计算表中的较低权限，除了拥有查看权限之外，还可以修改或删除单元格值。

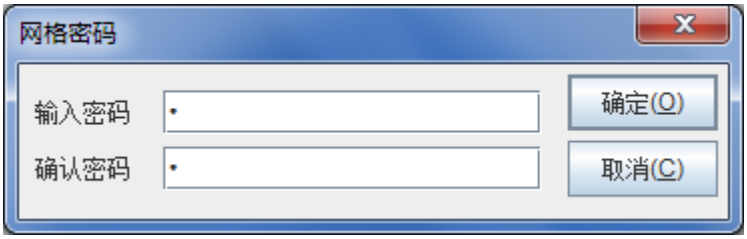
编辑权：这是计算表中的较高权限，除了拥有填写权限之外，还可以对网格中的记录进行各种操作，但是不能改变区段的结构，也不能修改单元格的属性。

完全控制权：这是计算表中的最高权限，可以对网格文件进行任何操作。

9.3.2 设置密码

在菜单栏中的工具选项中，点击网格密码按钮，可以弹出网格密码设定窗口：

在拥有某种权限时，只允许修改比当前权限更低的权限密码。点击相应权限下的按钮，可以在密码设定窗口中设定该权限的密码：



修改某个权限的密码时，需要先取消密码，再重新设定密码。取消密码时，必须要正确输入相应权限已设定的密码。

9.3.3 填写校验

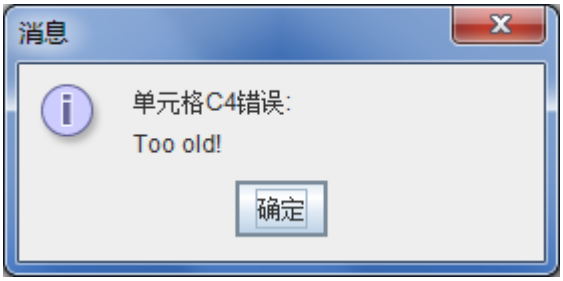
当用户拥有填写权或者修改权时，都可以编辑单元格值。为了防止在这种情况下，输入不符合要求的数据，可以在单元格中设定填写校验表达式，如果输入了不符合要求的单元格值，会弹出相应的文字提示。

打开计算表 esCalc09_1.gex，我们来如下面的网格：

0	1		A	B	C	D	E
1		1	Students				
2-		2	Name	Gender	Age		
	1	3	Tom	Male	12		
	1	4	Jerry	Male	14		

在 C3 中填入校验属性中填入 if(@>20,"Too old!",@<6,"Too young!")，这样，当 C3 及其同位格中的格值大于 20 或者小于 6 时，会分别弹出错误信息。

重新用编辑权限打开计算表文件，然后在 C3 中将年龄改为 26，则会弹出下面的错误信息：



点击确定后，会返回 C3 格要求重新编辑格值。

需要注意的是，在拥有完全控制权时，填写校验是不起作用的。

9.4 用户权限说明

- **免费版**

免费版的用户，计算表启动计算表时需要连接到官方网站，且无法打印网格中的数据。

- **标准版**

永久有效，标准版的用户拥有正常操作的权限。

10. 序列的高级应用

10.1 序列与成员

- **ifa(*x*)**

x 是否是序列。

- ***A*.len()**

序列 *A* 中成员的个数，称为 *A* 的**序列长度**。

- **空列、*n* 序列**

长度为 0 的序列，就是空列；长度为 *n* 的序列，称为 ***n* 序列**。

- **单列、纯序列**

没有重复成员的序列，称为单列；所有成员的数据类型相同的序列称为**纯序列**。

- ***A*(*i*)**

取出序列 *A* 中的第 *i* 个成员。注意，在计算表中，获取序列成员用 *A*(*i*)而**不是** *A*[*i*]。

	A	B	C	D	E
1	[1,5,3]				
2	=A1.len()	=A1(2)			

单元格中的表达式计算后，A2 格值为 3，B2 格值为 2。

- ***A*(*i*)=*x***

对序列 *A* 中的第 *i* 个成员赋值。

	A	B	C	D	E
1	[1,2,3]				
2	=A1(2)=5				

计算后，A2 格值为 5，同时 A1 的格值会变为[1,5,3]。

- ***A*.p(*i*)**

i>0 时，返回 *A* 中的第 *i* 个成员的序号；*i*<0 时，返回倒数第 *i* 个成员的序号。如果越界，返回的序号为 0。

	A	B	C	D	E
1	[a,b,c,d,e]				
2	=A1.p(3)	=A1.p(-2)	=A1.p(10)		

计算后，A2 格值为 3，B2 格值为 4，C2 格值为 0。

- ***A*.m(*i*)**

i>0 时，返回 *A* 中的第 *i* 个成员；*i*<0 时，返回倒数第 *i* 个成员。如果越界，返回 null。

	A	B	C	D	E
1	[a,b,c,d,e]				
2	=A1.m(3)	=A1.m(-2)	=A1.m(10)		

计算后，A2 格值为 c，B2 格值为 d，C2 格值为空值 null。

10.2 数列

成员**都是整数**的序列，称为**数列**。在计算表中，经常用 **to** 函数定义一个连续递增或递减的数列。

10.2.1 数列相关的概念

- ***n* 置换**

如果{n}数列同时又是 n 序列和单列，即是由 1,2,…,*n* 这 *n* 个数所构成，顺序不定的数列，称为 *n* 置换。

- **[1,4,3,2,5]、[1,2,3,4,5]和[5,4,3,2,1]，这些数列都是 5 置换。**

10.2.2 to 函数

- **to(*a*,*b*)**

产生从整数 *a* 到整数 *b* 的所有整数构成的数列，包括 *a* 和 *b* 本身，*a*>*b* 或者 *a*<*b* 均可。

- **to(*n*)**

to(1,*n*)的简写形式。

	A	B	C	D	E
1	=to(3,5)	=to(5,2)	=to(4)		

计算后，A1 格值为序列[3,4,5]，B1 格值为序列[5,4,3,2]，C1 格值为序列[1,2,3,4]。

- ***A*.to(*a*,*b*)**

序列 *A* 中，从第 *a* 个起到第 *b* 个成员所构成的序列，即 ***A*(to(*a*, *b*))**的简写形式。当 *a* 省略时，默认为 1；当 *b* 省略时，默认为 ***A*.len()**，注意此时逗号不能省略。

- ***A*.to(*a*)**

序列 *A* 中，从 1 个起到第 *a* 个成员所构成的序列，相当于 ***A*.to(1, *a*)**。

	A	B	C	D	E
1	=to(10).to(3)				
2	[a,b,c,d,e]				
3	=A2.to(3,5)				

计算后，A1 格值为序列[1,2,3]，A3 格值为序列[c,d,e]，A3 的表达式也可写为=A2.to(3,)。

10.3 产生列

由序列 *A* 的成员构成的序列称为 *A* 的**产生列**。

10.3.1 使用函数生成产生列

- ***A*(*p*)**

  [*A*(*p*(1)),*A*(*p*(2)),…]，用位置数列 *p* 获得 *A* 的产生列。

- ***A*.p(*p*)**

  [*A*(*p*(1)),*A*(*p*(2)),…]，获得位置数列 *p* 在 *A* 中对应的实际位置，*p* 中可以有负数。

- ***A*.m(*p*)**

  A(*A*.p(*p*))，用位置数列 *p* 获得 *A* 的产生列，*p* 中可以有负数。

- ***A*.dup()**

  A(to(*A*.len()))，即复制序列 *A*。

- ***A*.rvs()**

  A(to(*A*.len(),1))，即反转序列 *A*。

- ***A*.step(*m*,*k*)**

  A([*k*,*k*+*m*,*k*+2**m*,…])，即返回 *A* 的产生列，其位置数列从 *k* 开始，步进值为 *m*。

	A	B	C	D	E
1	[a,b,c,d,e]	[1,2,3,3]	[1,2,-2,8]		
2	=A1(B1)	=A1.p(C1)	=A1.m(C1)	=A1.dup()	=A1.rvs()

A2 中表达式用数列 B1 为序号生成序列 A1 的产生列，计算后格值为["a","b","c","c"]；B2 中表达式用数列 C1 获取在序列 A1 中的实际序号，超限为 0，计算后格值为[1,2,4,0]；C2 中表达式用数列 C1 获取排列 A1 的产生列，超限序号的对应成员为空，计算后格值为["a","b","d",null]；D2 中表达式复制序列 A1，计算后格值与 A1 相同；E2 中表达式获得 A1 成员反转的序列，计算后格值为["e","d","c","b","a"]。

10.3.2 在 *A*.p(*p*)与 *A*.m(*p*)中使用选项

在 *A*.p(*p*)与 *A*.m(*p*)中，可以使用选项**@r** 和**@0**。在使用@r 选项时，对于数列中越界的序号，在获取成员或者实际序号时，将进行回转操作，循环读取。在使用@0 选项时，数列中越界的序号将被忽略，0 或者空值不会出现在结果序列中。

	A	B	C	D	E
1	[a,b,c,d,e]	[1,2,-2,8]			
2	=A1.p@r(B1)	=A1.m@r(B1)	=A1.p@0(B1)	=A1.m@0(B1)	

A2 中的表达式用数列 B1 获取在序列 A1 中的实际序号，使用@r 选项，越界序号循环，计算后格值为[1,2,4,3]；B2 中的表达式用数列 B1 获取序列 A1 的产生列，使用@r 选项，越界序号循环，计算后格值为["a","b","d","c"]；C2 中的表达式用数列 B1 获取在序列 A1 中的实际序号，使用@0 选项，越界序号不出现，计算后格值为[1,2,4]，8 越界，结果序列里不出现 0；D2 中的表达式用数列 B1 获取序列 A1 的产生列，使用@0 选项，序号越界则 null 不出现，计算后格值为["a","b","d"]，8 作为序号越界，结果序列中不出现 null。

10.4 序列与字串

通过 *s*.array()和 *A*.string()两个函数，序列和字串可以很方便地相互转化。

- ***s*.array(*d*)**

将字串 *s* 以分隔符 *d* 拆分成序列，自动识别数据类型；*d* 缺省为逗号。

- ***A*.string(*d*)**

将序列 *A* 用分隔符 *d* 连接，拼接成字串，自动处理数据类型；*d* 缺省为逗号。

	A	B	C	D	E
1	a,1,c,2011-8-11,false				
2	=A1.array()	=A2.string()			

A2 中的表达式，将字串以逗号作为分隔符，拆分成序列，序列中每个成员自动识别为相应的数据类型，计算后格值为序列["a",1,"c",2011-8-11,false]；B2 中的表达式将 A2 中的序列拼为字符串，序列中的字符串成员将会添加双引号，计算后格值为"a",1,"c",2011-8-11,false。

10.5 序列的基本运算

10.5.1 序列的双目运算

- ***A*|*B***

和列，简单将两个序列连接起来，其中 *B* 的成员添加在 *A* 成员的后面。当 *A*、*B* 其中之一或者两者都是单值，而不是序列时，作为单值的序列来处理。

- ***A*&*B***

并列，获得新序列，在 *A* 的成员之后再加入 *B* 中成员，加入时去除 *B* 中已在 *A* 中出现的成员。当 *A*、*B* 其中之一或者两者都是单值，而不是序列时，作为单值的序列来处理。

- ***A*^*B***

交列，*A* 与 *B* 的交集，从 *A* 中依次获得同时出现在 *B* 中的成员所组成的序列。

- ***A**B***

差列，在 *A* 中但未在 *B* 中出现的成员

- ***k**A***

A|*A*|…|*A*，*k* 个 *A* 做和列，即将 *A* 复制 *k* 遍，表达式中 *k* 和 *A* 的位置可以互换。

	A	B	C	D	E
1	[a,b,1,2,3,4]	[d,b,10,12,4,3]			
2	=A1|B1	=A1&B1	=A1^B1	=A1\B1	
3	=2*A1	=[1]*8			

A2 中表达式计算序列 A1 与序列 B1 的和列，计算后格值为序列["a","b",1,2,3,4,"d","b",10,12,4,3]，可以看到，结果将 A1 和 B1 中的成员拼在了一起。

B2 中表达式计算序列 A1 与序列 B1 的并列，计算后格值为序列["a","b",1,2,3,4,"d",10,12]，可以看到，结果中不会出现重复成员。

C2 中表达式计算序列 A1 与序列 B1 的交列，计算后格值为序列["b",3,4]。

D2 中表达式计算序列 A1 与序列 B1 的差列，计算后格值为序列["a",1,2]。

A3 中表达式计算序列 A1 与整数 2 相乘，即将 A2 复制两遍，计算后格值为["a","b",1,2,3,4,"a","b",1,2,3,4]。

B3 中表达式计算序列[1]与整数 8 相乘，计算后格值为 8 个 1 组成的序列：[1,1,1,1,1,1,1,1]。

10.5.2 序列的对位运算

长度相同两个数值构成的序列可按成员进行对位计算，返回序列。

- ***A++B***

  [*A*(1)+*B*(1),*A*(2)+*B*(2),…]

- ***A--B***

  [*A*(1)-*B*(1),*A*(2)-*B*(2),…]

- ***A**B***

  [*A*(1)**B*(1),*A*(2)**B*(2),…]

- ***A//B***

  [*A*(1)/*B*(1),*A*(2)/*B*(2),…]

- ***A%%B***

  [A(1)%B(1),A(2)%B(2),…]，这里%是求余运算。

- ***A\\B***

  [A(1)\B(1),A(2)\B(2),…]，这里\是求整运算。

	A	B	C	D	E	F
1	[1,2,3]	[4,1,2]				
2	=A1++B1	=A1--B1	=A1**B1	=A1//B1	=A1%%B1	=A1\\B1

A2 中表达式为数列 A1 与 B1 对位相加，计算后格值为数列[5,3,5]。

B2 中表达式为数列 A1 与 B1 对位相减，计算后格值为数列[-3,1,1]。

C2 中表达式为数列 A1 与 B1 对位相乘，计算后格值为数列[4,2,6]。

D2 中表达式为数列 A1 与 B1 对位相除，计算后格值为序列[0.25,2.0,1.5]，注意，所有成员均为实数的数列并不是数列，数列是指所有成员均为整数的序列。

E2 中表达式为数列 A1 与 B1 对位计算求余，计算后格值为数列[1,0,1]。

F2 中表达式为数列 A1 与 B1 对位计算求整，计算后格值为数列[0,2,1]。

10.5.3 序列的比较

用函数 cmp(*x*,*y*)可以比较两个表达式 *x* 和 *y* 计算结果的大小，用函数 cmp(*A*,*B*)可以比较两个序列 *A* 与 *B* 的大小。

- **cmp(*x*,*y*)**

  比较表达式 *x* 和 *y* 计算值的大小，*x*>*y* 时返回 1，*x*<*y* 时返回-1，*x*=*y* 则返回 0。如果 *x* 与 *y* 无法比较，则报错。

	A	B	C	D	E
1	=cmp("Alex","Allen")		=cmp(0.4*3,1)	=cmp(2+4,6)	
2	/=cmp("abc",15)		/=cmp(15,"abc")		

A1 中表达式比较两个字符串的大小，根据字母的 ASCII 码的大小依次比较，由于"Alex"<"Allen"，因此计算后格值为-1。

C1 中表达式比较两个数的大小，由于 1.2>1，因此计算后格值为 1。

D1 中表达式比较两个数的大小，由于 6==6，因此计算后格值为 0。

最后再来看 A2 和 C2 中的表达式，由于字符串与整数不能比较，因此计算时会报错。

- **cmp(*A*,*B*)**

  比较序列大小时，对位比较每个成员的值，遇到第一个不等成员时根据大小分别返回 1 或-1，*A* 与 *B* 全等则返回 0。

	A	B	C	D	E
1	=cmp(["a","b","c"],["a","b","c"])				
2	=cmp([1,3,5,7],[1,3,7,5])				
3	=cmp([7,6,5,4],[7,6,4,10,11])				

在计算后，A1 格值为 0，即两序列相等；A2 格值为-1，即第 1 个序列较小；A3 格值为 1，即第 1 个序列较大。

与普通数值的比较相类似，对于两个序列 *A* 和 *B*，根据 cmp(*A*,*B*)的结果，可以简写为以下形式：

A*>*B	cmp(*A*,*B*)>0
A*<*B	cmp(*A*,*B*)<0
A*==*B	cmp(*A*,*B*)==0
A*!=*B	cmp(*A*,*B*)!=0
A*>=*B	cmp(*A*,*B*)>=0
A*<=*B	cmp(*A*,*B*)<=0

10.5.4 序列的聚合运算

在计算表中，对于 *n* 序列 *A*，可以使用函数进行各种聚合运算。

- ***A*.count()**

计算 *A* 中非空成员计数，和 *A*.len()不同。

- ***A*.ifn()**

获取 *A* 中的第一个非空成员。

- ***A*.sum()**

对 *A* 中所有成员求和。

- ***A*.avg()**

计算 *A* 中成员的平均值，计算时不计空值成员。

	A	B	C	D	E
1	[null,4,6,,2,4,,5]				
2	=A1.count()	=A1.len()	=A1. ifn()	=A1.sum()	=A1.avg()

A2 中表达式计算序列 A1 的非空成员个数，计算后格值为 5，注意这和 len()函数是不同的。

B2 中表达式计算序列 A1 成员个数，计算后格值为 8。

C2 中表达式返回序列 A1 的首个非空成员，计算后格值为 4。

D2 中表达式计算序列 A1 中成员的和，计算后格值为 21。

E2 中表达式计算序列 A1 中成员的平均值，计算后格值为 4.2，可见，avg 函数在计算序列平均值时，是不计序列中的空值的。

- ***A*.min()**

求最小值。

- ***A*.max()**

求最大值。

- ***A*.variance()**

求方差。计算方差时，不计空值成员。

- ***A*.ranki(*y*)**

求 *y* 在序列中值的排名，不要求 *y* 是序列中成员的值。

- ***A*.rank()**

求序列中每个成员的排名。

	A	B	C	D	E
1	[null,4,6,,2,4,,5]				
2	=A1.min()	=A1.max()	=A1.variance()	=A1.ranki(4.5)	=A1.rank()

A2 中表达式计算序列 A1 中的最小值，计算后格值为 2。

B2 中表达式计算序列 A1 中的最大值，计算后格值为 6。

C2 中表达式计算序列 A1 的方差，计算后格值为 1.76。

D2 中表达式计算在序列 A1 中，4.5 的排名，4.5 不一定存在于序列中，计算后格值为 3。

E2 中表达式计算序列 A1 的排名序列，计算后格值为序列[6,3,1,6,5,3,6,2]。

- ***A*.conj()**

求序列 *A* 中所有成员的和列，如果 *A* 的某个成员不是序列，则作为 1 序列处理。

- ***A*.union()**

求序列 *A* 中所有成员的并列，如果 *A* 的某个成员不是序列，则作为 1 序列处理。(类似 SQL 中的 UNION 运算符的效果，结果序列中不会出现重复成员)

- ***A*.diff()**

依次将序列 *A* 中的成员做差列计算，如果 *A* 的某个成员不是序列，则作为 1 序列处理。

- ***A*.isect()**

求序列 *A* 中所有成员的交列，如果 *A* 的某个成员不是序列，则作为 1 序列处理。

	A	B	C	D	E
1	[[1,2,3],3,[3,4],[5,6,3]]				
2	=A1.conj()	=A1.union()	=A1.diff()	=A1.isect()	

A2 中表达式计算序列 A1 的连接，计算后格值为序列[1,2,3,3,3,4,5,6,3]。

B2 中表达式计算序列 A1 的联合，计算后格值为序列[1,2,3,4,5,6]，联合的计算结果等价于计算连接后再去除相同的成员。

C2 中表达式计算序列 A1 中成员的差集，计算后格值为序列[1,2]。

D2 中表达式计算序列 A1 中成员的交集，计算后格值为序列[3]。

10.6 循环

10.6.1 序列循环中表达式的约定

循环函数参数中可能引用序列成员，约定规则：

- **~**

当前序列成员

- ***A*.#**

当前成员的序号

- ***A*.sum(~*~)** 计算 *A* 中成员的平方和。
- ***A*.max(if(*A*.#%2==0,~,0))** 找出偶数位置成员的最大值。

10.6.2 序列的循环函数

针对序列的每个成员做某种计算的函数称为**循环函数**，一般形式为 ***A.f*(…)**，如 **10.5.4 序列的聚合运算**中的函数如 ***A*.sum()**等，实际上，序列的聚会运算也是循环函数。

循环函数的计算行为由函数名决定，如 **sum** 表示求和、**avg** 表示平均、…。

➢ ***A*.count(*x*)**

返回 *A* 中使 *x* 满足的成员个数

- ***A*.count(~>5)** 比 5 大的成员个数。

	A	B	C	D	E
1	[1,3,5,null,7]				
2	=A1.count()	=A1.sum(~*~)	=A1.count(~>2)		

A2 中表达式计算序列 A1 中非空成员的个数，计算后格值为 4。

B2 中表达式计算序列 A1 中成员的平方和，计算后格值为 84。

C2 中表达式计算序列 A1 中大于 2 的成员个数，计算后格值为序列 3。

返回相关序列：

➢ ***A*.(*x*)**

返回针对 *A* 中每个成员计算表达式 *x* 后构成的序列

	A	B	C	D	E
1	[1,3,5,null,7]				
2	=A1.(A1.#)	=A1.(~*~)	=A1.(A1.##)		

A2 中表达式计算序列 A1 中每个成员的序号构成的序列，计算后格值为序列[1,2,3,4,5]。

B2 中表达式计算序列 A1 中每个成员的平方构成的序列，计算后格值为序列[1,9,25,null,49]。

C2 中表达式对序列 A1 中每个成员计算 A1 的序列长度并组成序列，计算后格值为序列[5,5,5,5,5]，从结果可以看出，*A*.##表示的是 *A* 的序列长度，包括其中为 null 的成员。

整数循环：

➢ ***n.f*(*x*)**

to(*n*).*f*(*x*)的简写形式。

	A	B	C	D	E
1	=10.sum()	=10.avg()	=5.(2*~-1)		

A1 中表达式计算 1 加到 10 的和，计算后格值为 55。

B1 中表达式计算 1 到 10 的平均值，计算后格值为 4.5。

C1 中表达式计算前 5 个奇数构成的数列，计算后格值为序列[1,3,5,7,9]。