

o++o 和 EXCEL 的比较

Klaus Benecke, Eicke Redweik, Stephan Schenkl

(状态: 2022 年 1 月 5 日)

本文以一个典型的结构化表格问题为基础, 比较了 o++o 和 EXCEL 在编程方面的努力。此外, 还考虑了对结构化表的模式稍作修改所需的努力。事实证明, 在 o++o 程序中, 只有最初的表格需要稍作扩展, 而 EXCEL 应用程序则需要进行重大修改。甚至可以说, 这些是不同的 EXCEL 应用程序。从用户的角度来看, 这个问题或一类问题的解决被 o++o 大大简化了, 因为许多 EXCEL 应用可以用 o++o 以类似但更简单的方式实现。如果将典型的 EXCEL 用户与 o++o 用户进行比较, 就会发现, 无论是 O++o 解决方案的制定还是随后的调整, 都比 EXCEL 解决方案花费的时间要少很多。这项工作的结果既适用于非结构化表, 也适用于结构化表, 只要它们不包含空集, 从而可以很容易地被扁平化而不损失信息。最后, 根据一般方面对两种方法进行了比较。

内容

1.初始问题 1	
1.1. 与 o++o2 的溶液	
1.2 用 EXCEL 解决 (手动)	5
1.3 用 EXCEL 解决 (自动化)	6
2. 扩展问题 10	
2.1 使用 o++o11 的解决方案	
2.2 使用 Excel 的解决方案(手动)	12
2.3 使用 Excel 的解决方案(自动化)	14
3.评价 15	
4 一般性比较 17	

1. 最初的问题

对于以下表格, BMI (身体质量指数) 的数值应根据公式确定 $BMI = GEWICHT : (LAENGE \cdot LAENGE) = GEWICHT : LAENGE : LAENGE$ 公式。每个人的数值和该人列出的每个年龄段的数值都要计算, 还要按年龄分组, 计算该年龄段的人的平均 BMI 值, 以及所有 BMI 值的平均值。只有 ALTER 值超过 20 的人才能被包括在内。我们一会儿就会看到, 用 o++o 的问题的措辞比前面的德语描述更容易掌握。

名称	辽宁省	ALTER	重量
克劳斯	1.68	18	61
		30	65
		56	80
罗尔夫	1.78	40	72
卡蒂	1.70	18	55
		40	70
华莱士	1.00	3	16
Viktoria	1.61	13	51

伯特	1.72	18	66
		30	70

1.1. 用 o++o 解决

给定的表格可以保存为文件或输入到程序部分（上部浅绿色区域），如下图 TAB 表示。o++o 方案由所需的 选译(avec 线)和 带有聚合的重组(gib 线) 组成。通过

第 2 轮

然后所有的数字都被四舍五入到小数点后 2 位。TAB 格式的结果可以在下面的深绿色区域找到, 表头在中间区域。

The screenshot shows the o++o web interface. The top section is a light green text area for input. The bottom section is a dark green text area for output. Between them is a control bar with buttons for width, meta, output, run, autoclear, load, save, new window, and help.

Input (TAB format):

```
<TAB!
NAME,      LAENGE,  (ALTER,  GEWICHT  l)m
Klaus      1.68      18      61
              30      65
              61      81
              68      76
Rolf       1.78      40      72
Kathi      1.70      18      55
              40      70
Walleri    1.00      3       16
Viktoria   1.61      13      51
Bert       1.72      18      66
              30      70

!TAB>
avec NAME! ALTER>20
BMI:=GEWICHT : LAENGE : LAENGE
gib DUR,(ALTER,DUR,(NAME, DUR m) m) DUR:=BMI! ++:
rnd 2
```

Control Bar:

width: 500 meta: normal output: tab run autoclear:
☒ load bmi.otto save new window help

Output (TAB format):

```
DUR, (ALTER, DUR2, (NAME, DUR3 m) m)

23.58 18      20.98 Bert 22.31
              Kathi 19.03
              Klaus 21.61
      30      23.35 Bert 23.66
              Klaus 23.03
      40      23.47 Kathi 24.22
              Rolf  22.72
      61      28.7  Klaus 28.7
      68      26.93 Klaus 26.93
```

o++o:输入数据和处理代码（顶部，用绿色突出显示），输出标签模式（中间，用很深的颜色突出显示）和输出标签内容（底部，用深绿色突出显示）。

如果如前所述，你只是将源数据保存为格式为 "tab "的文本文件，文件名为 `personen.tab`。

姓名，身長，（年龄，体重 m）m

克劳斯 1.68 18 61

30 65

61 80

罗尔夫 1.78 40 72

卡蒂 1.70 18 55

40 70

Walleri 1.00 3 16

维多利亚州 1.61 13 51

伯特 1.72 18 66

30 70

上述方案采取以下形式。

People.tab

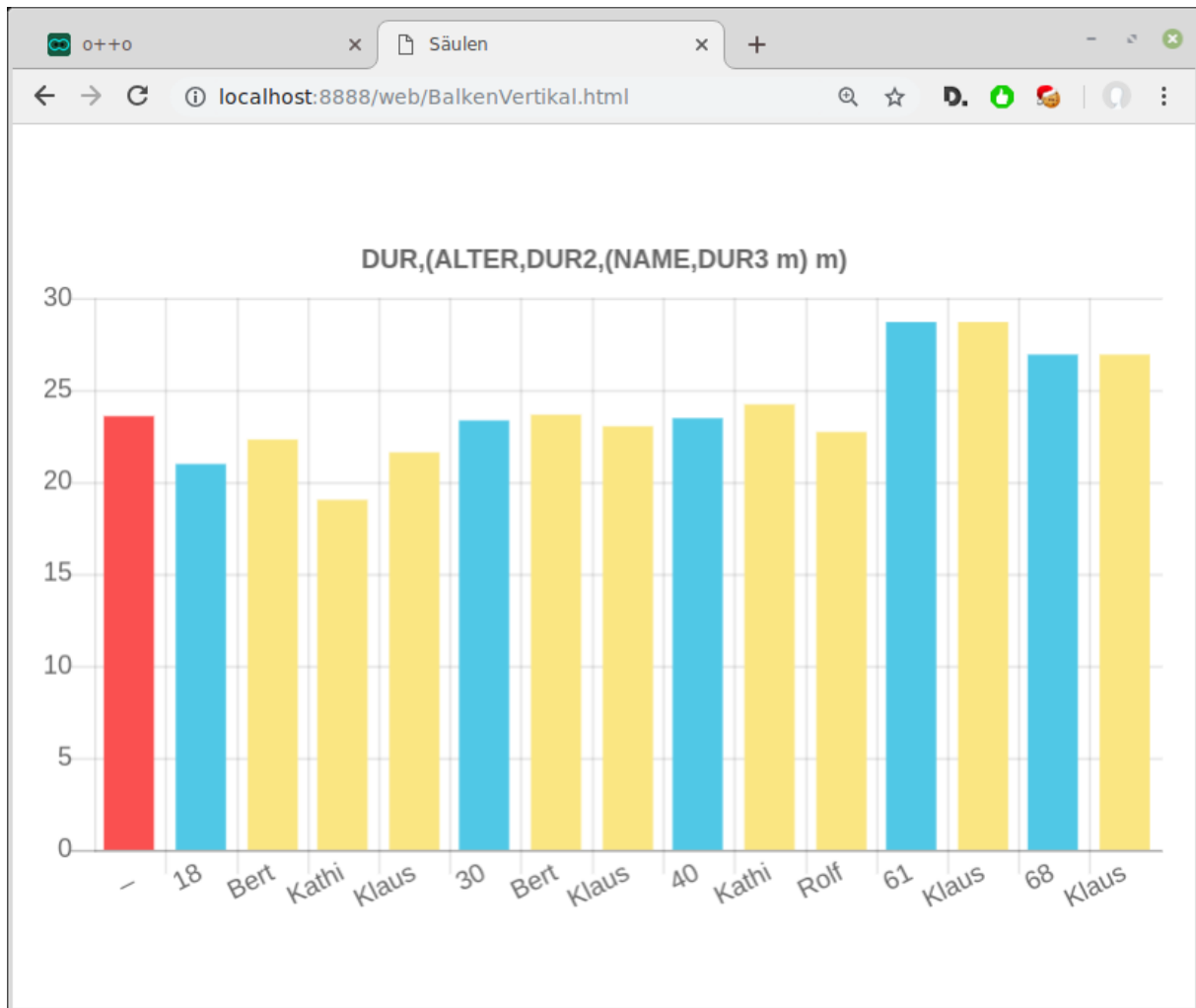
avec NAME!ALTER>20

给予 BMI, (AGE, BMI, (NAME, BMI m) m) BMI:=WEIGHT:LENGTH:LENGTH!++:

第 2 轮

关于++o 的详细描述可以在 www.ottops.de。

通过点击图表按钮，会有一个所需结构图的选择列表提示。再点击一下，就会显示所选的图。这里，上级栏目的数值以蓝色突出显示。零点水平的柱子是红色的。



o++o- 图：结构化输出表的条形图

在++o 图中，数字值被解释为列长，非数字值被解释为签名。由于 ALTER 在上述程序中是数字性的，为了生成上述图表，程序行

ALTER::=ALTER 字

被附上。

1.2. 用 EXCEL 解决（手动）。

在使用 EXCEL 时，必须首先根据 "年龄大于 20 岁的人" 这一标准进行手动过滤，因为天真的应用过滤条件 $ALTER > 20$ 会在 "平铺" 表格后消除相关数据行。

BMI 的公式输入本表的 E2 并复制到 E9。

E2						
	A	B	C	D	E	F
1	NAME	LAENGE	ALTER	GEWICHT	BMI3	
2	Klaus	1,68	18	61	21,6128118	
3		1,68	30	65	23,0300454	
4		1,68	56	80	28,3446712	
5	Rolf	1,78	40	72	22,7244035	
6	Kathi	1,7	18	55	19,0311419	
7		1,7	40	70	24,2214533	
8	Bert	1,72	18	66	22,3093564	
9		1,72	30	70	23,6614386	
10						

在 EXCEL 中计算 BMI:手动过滤相关数据行,复制 LAENGE,通过公式确定每个数据行的 BMI 并复制。

随后,按年龄进行分类和分组。名字也被复制到空单元格中,以创建一个平面的表格结构。每个年龄组的 BMI 平均值的计算在 F 栏进行,所有数据的 BMI 平均值的计算在 G 栏进行。

G2							
	A	B	C	D	E	F	G
1	NAME	LAENGE	ALTER	GEWICHT	BMI3	BMI2	BMI
2	Klaus	1,68	18	61	21,6128118	20,9844367	23,1169153
3	Kathi	1,7	18	55	19,0311419		
4	Bert	1,72	18	66	22,3093564		
5							
6							
7	Klaus	1,68	30	65	23,0300454	23,345742	
8	Bert	1,72	30	70	23,6614386		
9							
10							
11							
12							
13	Rolf	1,78	40	72	22,7244035	23,4729284	
14	Kathi	1,7	40	70	24,2214533		
15							
16							
17							
18							
19	Klaus	1,68	56	80	28,3446712	28,3446712	
20							

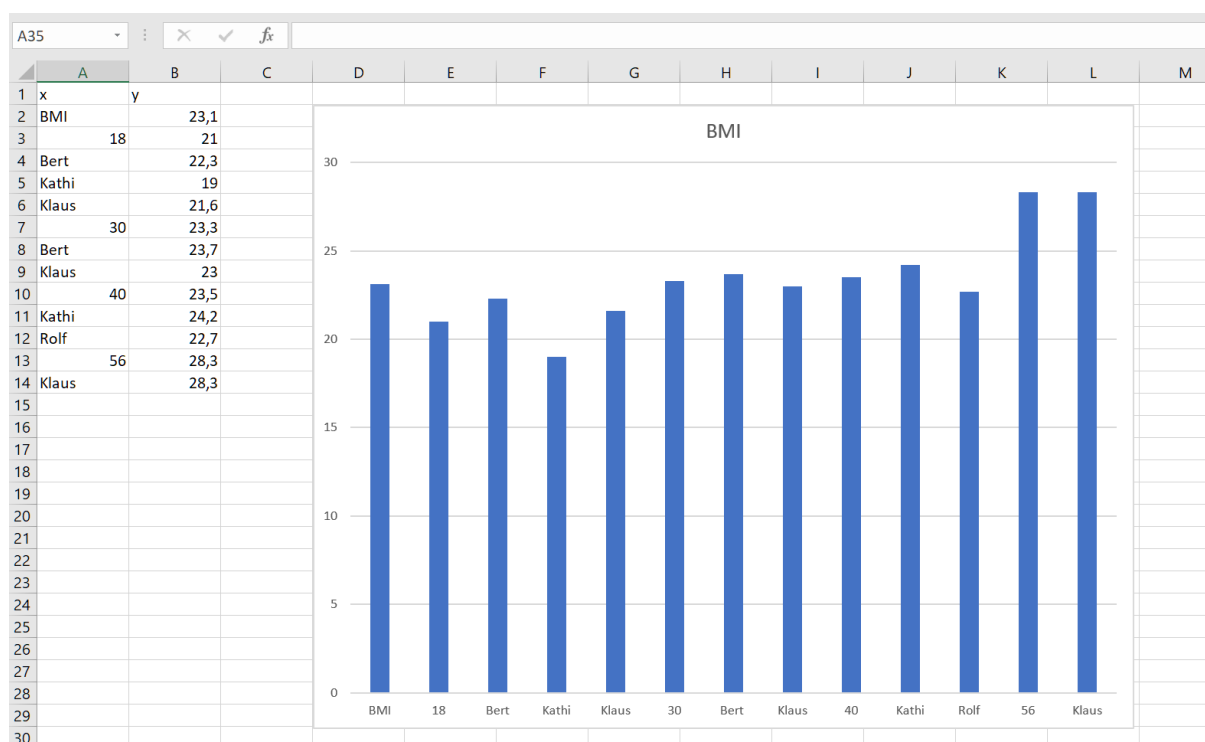
在 EXCEL 中对分组的数据进行 BMI 计算。通过 ALTER 对平面表的数据行进行手工分组,并确定 BMI 的平均值。

如果行或列被添加或删除,计算单元格的公式区域必须在手动编辑时进行修正。根据所需的安排,必须移动列,并将结果表中不需要的列隐藏。

C2						
	A	B	C	D	E	H
1	BMI	ALTER	BMI2	NAME	BMI3	
2	23,1169153	18	20,9844367	Klaus	21,6128118	
3		18		Kathi	19,0311419	
4		18		Bert	22,3093564	
5		30	23,345742	Klaus	23,0300454	
6		30		Bert	23,6614386	
7		40	23,4729284	Rolf	22,7244035	
8		40		Kathi	24,2214533	
9		56	28,3446712	Klaus	28,3446712	
10						

EXCEL 中的手动选列：手动安排和隐藏列。

除了四舍五入和 ALTER 列中的冗余，这个结果表与++o 的计算结果一致。图形输出需要进一步手工准备要显示的数据，这也随着数据量的增加而增长。



在 EXCEL 中的图形表示：数据被安排在一个平面表中，以创建一个条形图

1.3. 使用 EXCEL 的解决方案（自动）。

初步说明：EXCEL 倾向于将空单元格解释为零值或 0 值。为了防止出现不需要的 0 值，经常使用 IF(<cell>="";"";<formula>)形式的查询，每次都不明确重复。

此外，在可变的表格大小上工作的步骤，例如行选择或列选择，如果传递空值，就会在输出表格以外的单元格中产生错误。在这种情况下，这些单元格被 IFERROR(<formula>; "")转换为空单元格。这也不是在每个点上都明确地重复。

处理管道是通过在一个新的工作表上输出每个步骤来实现的。该模式在许多处理步骤中没有被触及，而是被原封不动地转移到下面的工作表中。模式的单元被称为模式单元，其他相关单元被称为数据单元。

处理整个工作表的 EXCEL 公式可能需要很长的时间和大量的内存。在这个例子中，使用的公式在每个工作表(包括模式)上占用了 A1:J50 的最大范围。

当使用 EXCEL 时，数据可以手动输入或以 "csv" 格式的文件形式存储。通过使用自由单元格，一个结构化的表格也可以用于 EXCEL，如例子中所示。在这种情况下，第一个预处理步骤是将输入表 "扁平化"，因为在 EXCEL 中进一步处理需要一个非结构化的表。

这一步是通过公式 `IF(input.A2="";A1;input.A2)` 实现的。由于第一行被完全占用，这个应用程序用上面的单元格的内容来填充每个空单元格。

	A	B	C	D
1	NAME	LAENGE	ALTER	GEWICHT
2	Klaus	1,68	18	61
3			30	65
4			56	80
5	Rolf	1,78	40	72
6	Kathi	1,7	18	55
7			40	70
8	Walleri	1	3	16
9	Viktoria	1,61	13	51
10	Bert	1,72	18	66
11			30	70

	A	B	C	D
1	NAME	LAENGE	ALTER	GEWICHT
2	Klaus	1,68	18	61
3	Klaus	1,68	30	65
4	Klaus	1,68	56	80
5	Rolf	1,78	40	72
6	Kathi	1,7	18	55
7	Kathi	1,7	40	70
8	Walleri	1	3	16
9	Viktoria	1,61	13	51
10	Bert	1,72	18	66
11	Bert	1,72	30	70

工作表输入：电子表格中的结构化表格

工作表输入-平面：电子表格中的平面表

现在可以根据 "年龄大于 20 岁的人" 的标准进行自动过滤。选择最好分三个步骤进行。

- 人们用公式 `'input-flat'.C2>=20` 来确定要选择的行，其中 C 列包含 ALTER 值（工作表'vec-bool'）。
- 用公式 `IF(COUNTIF('avec-bool'.B2:B50;'input-flat'.$A2);'input-flat'.$A2;"")`（工作表'vec-holes'）把要选择的行转移到一个全新的工作表中，留下空白行。
- 空白线分两步消除。
 - 首先用 `NOT('avec-holes'.A2="")` 检测空行，并在该工作表中用 `COUNTIF(A1:A2;1)` 创建（工作表'hole-help'）的（非空）数据行索引。
 - 用公式 `LOOKUP('hole-help'.$C2;'hole-help'.$B1:'hole-help'.$B50;'avec-holes'.A2:A50)` 只读入非空的数据行（工作表'vec'）。

在这个处理步骤之后，将得到一个与用++o 创建的表相当的表。

	A	B	C	D
1				
2	FALSCH			
3	WAHR Klaus			
4	WAHR Klaus			
5	WAHR Rolf			
6	FALSCH			
7	WAHR Kathi			
8	FALSCH			
9	FALSCH			
10	FALSCH			
11	WAHR Bert			

	A	B	C	D
1	NAME	LAENGE	ALTER	GEWICHT
2	Klaus	1,68	18	61
3	Klaus	1,68	30	65
4	Klaus	1,68	56	80
5	Rolf	1,78	40	72
6	Kathi	1,7	18	55
7	Kathi	1,7	40	70
8				
9				
10	Bert	1,72	18	66
11	Bert	1,72	30	70

工作表 Avec-Bool: 用于选择相关键的真值

工作表: 选择后电子表格中的平面表有空白行

	A	B	C	D
1		0		
2	WAHR	1	0	
3	WAHR	2	1	
4	WAHR	3	2	
5	WAHR	4	3	
6	WAHR	5	4	
7	WAHR	6	5	
8	FALSCH	6	6	
9	FALSCH	6	7	
10	WAHR	7	8	
11	WAHR	8	9	

	A	B	C	D
1	NAME	LAENGE	ALTER	GEWICHT
2	Klaus	1,68	18	61
3	Klaus	1,68	30	65
4	Klaus	1,68	56	80
5	Rolf	1,78	40	72
6	Kathi	1,7	18	55
7	Kathi	1,7	40	70
8	Bert	1,72	18	66
9	Bert	1,72	30	70
10				
11				

工作表孔-帮助。选择空行的真值

工作表 Avec: 选择后没有空白行的平面表

现在有必要根据年龄对数据进行分类。为了简单起见，首先按照 ALTER, NAME, WEIGHT, LENGTH 的方案用 LOOKUP(工作表'gib-help') 对列进行排序。

排序只针对个别列，所以首先按 ALTER: SMALL(OFFSET('gib-help'. \$A\$2:\$A\$50;0;0;49-COUNTIF('gib-help'. \$A\$2:\$A\$50;""));ROW(F2)-1) 排序。) 请特别注意，行数的确定取决于支持的最大数据行数 (49=50-1, 因为保留了一个模式行)。

对于这个排序的列，现在使用 INDEX('\$gib-help'.B:B;'gib-help-sort-help'. \$F2+1;1)(工作表'gib-help-sort') 逐行添加缺少的数据单元。

	A	B	C	D
1	ALTER	NAME	GEWICHT	LAENGE
2		18 Klaus	61	1,68
3		30 Klaus	65	1,68
4		56 Klaus	80	1,68
5		40 Rolf	72	1,78
6		18 Kathi	55	1,7
7		40 Bert	70	1,7
8		18 Bert	66	1,72
9		30 Bert	70	1,72
10				
11				

Gib-Help 工作表。自动重新排序表列

	A	B	C	D
1	ALTER	NAME	GEWICHT	LAENGE
2		18 Klaus	61	1,68
3		18 Kathi	55	1,7
4		18 Bert	66	1,72
5		30 Klaus	65	1,68
6		30 Bert	70	1,72
7		40 Rolf	72	1,78
8		40 Kathi	70	1,7
9		56 Klaus	80	1,68
10				
11				

Gib-Help-Sort 工作表:按第一个表栏 (年龄) 排序

这是包含体重和身长信息的最后一张工作表，所以最晚现在必须用 '\$gib-help-sort'.C2/'\$gib-help-sort'.D2/'\$gib-help-sort'.D2(工作表'BMI') 计算 BMI。

	A	B	C	D	E
1	ALTER	NAME	GEWICHT	LAENGE	BMIh
2		18 Klaus	61	1,68	21,61281179
3		18 Kathi	55	1,7	19,03114187
4		18 Bert	66	1,72	22,30935641
5		30 Klaus	65	1,68	23,03004535
6		30 Bert	70	1,72	23,66143862
7		40 Rolf	72	1,78	22,72440348
8		40 Kathi	70	1,7	24,22145329
9		56 Klaus	80	1,68	28,3446712
10					
11					

BMI 工作表:一个新的栏目构成了结构化输出表中 BMI 栏目的基础。

gib 的处理是通过首先接管 ALTER 和 NAME 列来完成的 (公式同列重排序: HLOOKUP(B\$1;'gib-help-sort'. \$A\$1:\$J\$50;ROW(\$K2))). 根据保存的 BMI 值计算出三个新栏目。

- bmi=average(\$bmi.e2:e50)
- bmi2=if(b2=b1;"";averageif(bmi.\$a\$2:\$a\$50; b2;bmi.\$e\$2:\$e\$50))
- BMI3=\$BMI.E2

然后按年龄进行分类和分组。为此，名字也必须被复制到空单元格中。每个年龄组的 BMI 平均值的计算在 F 栏进行，所有数据的 BMI 平均值的计算在 G 栏进行。

如果增加或删除了行或列，必须对要计算的单元格的公式区域进行修正。根据所需的安排，必须移动列，结果表中不需要的列必须被隐藏 (工作表'gib')。

	A	B	C	D	E
1	BMI	ALTER	BMI	NAME	BMI
2	23,11691525	18	20,98443669	Klaus	21,61281179
3				Kathi	19,03114187
4				Bert	22,30935641
5		30	23,34574198	Klaus	23,03004535
6				Bert	23,66143862
7		40	23,47292839	Rolf	22,72440348
8				Kathi	24,22145329
9		56	28,3446712	Klaus	28,3446712
10					
11					

工作表 **Gib**:结构化输出表, 包括 BMI 的所有分组值

除了四舍五入, 这个结果表与 ++o 的计算结果一致。四舍五入是用 IFERROR(ROUND(\$gib.A1;2);\$gib.A1)完成的 (注意: 这里用 IFERROR 是为了忽略文本单元格)。

	A	B	C	D	E
1	BMI	ALTER	BMI	NAME	BMI
2	23,12	18	20,98	Klaus	21,61
3				Kathi	19,03
4				Bert	22,31
5		30	23,35	Klaus	23,03
6				Bert	23,66
7		40	23,47	Rolf	22,72
8				Kathi	24,22
9		56	28,34	Klaus	28,34
10					
11					

输出工作表。最终的输出表有四舍五入的数值

2. 扩展问题

通过对每个人的体重和年龄的几个数值来扩展这个例子, 得出以下计算过程。在计算每个年龄段的 BMI 平均值和 BMI 的总平均值时, 要包括所有体重的数值。如果首先形成每人体重和年龄的平均值, 就会出现偏差的结果。

名称	辽宁省	ALTER	重量
克劳斯	1.68	18	61 60 62
		30	65 63 67
		56	80 82 79
罗尔夫	1.78	40	72 70 74
卡蒂	1.70	18	55 52
		40	70 71
华莱士	1.00	3	16 15
Viktoria	1.61	13	51 50
伯特	1.72	18	66 65
		30	70 71

2.1. 用 o++o 解决

以 tabment 和 o++o 方案的形式输入表格。我们看到, 除了输入额外的数据外, 只需要在 WEIGHT 后面插入一个 l 表示列表, 在 TAB 处插入一个 H。H 代表水平, 因为现在重量数据已经被紧凑地水平显示。

<TABH!

NAME, LAENGE, (ALTER, GEWICHTl m)m

```
!TABH>
avec NAME! ALTER>20
gib BMI,(ALTER,BMI,(NAME,BMI m) m) BMI:=GEWICHT:LAENGE:LAENGE!++:
rnd 2
```

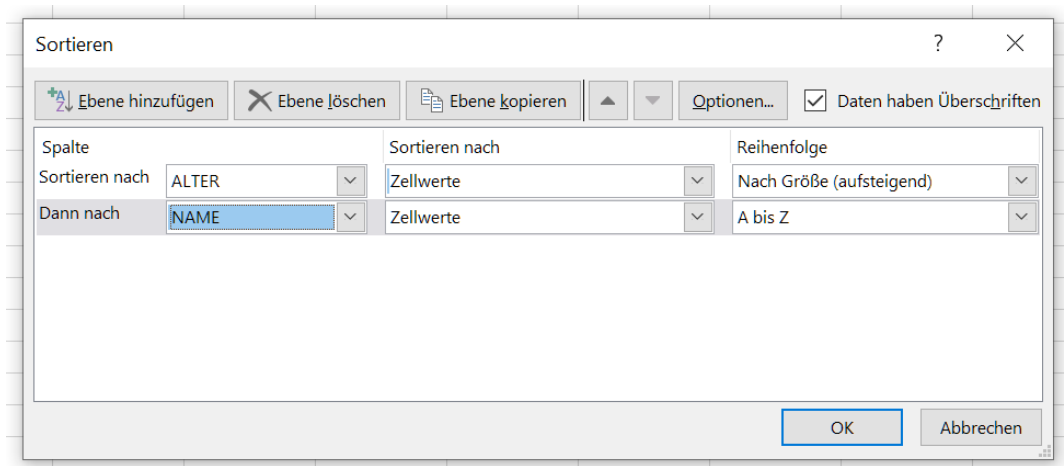
10

OttoPSClient 中的扩展问题：只有输入表被修改，而具有相应模式的输出表被同一代码创建。如果事先将类型从年龄转换为 WORD，所需的图表又可以通过两次点击生成。由于它与上图相似，我们在此不做第二次说明。

2.2. 使用 Excel 的解决方案（手动）。

在创建所需的表列后，输入 NAME、LENGTH、AGE 和 WEIGHT 的值。这就造成了冗余，因为要包括的每个权重都必须与 NAME 和 AGE 的相应值在一行。根据 "年龄大于 20 岁的人" 的标准进行的过滤必须再次手动完成。

用户定义的排序被选择为按 ALTER 和 NAME 对表的数值进行排序。这是半自动的，但在每次修改输入数据后必须再次手动触发。



EXCEL 中的半自动排序：通过对话可以手动选择要排序的列。

	A	B	C	D	E
1	NAME	LAENGE	ALTER	GEWICHT	
2	Bert	1,72	18	66	
3	Bert	1,72	18	65	
4	Kathi	1,7	18	55	
5	Kathi	1,7	18	52	
6	Klaus	1,68	18	61	
7	Klaus	1,68	18	60	
8	Klaus	1,68	18	62	
9	Bert	1,72	30	70	
10	Bert	1,72	30	71	
11	Klaus	1,68	30	65	
12	Klaus	1,68	30	63	
13	Klaus	1,68	30	67	
14	Kathi	1,7	40	70	
15	Kathi	1,7	40	71	
16	Rolf	1,78	40	72	
17	Rolf	1,78	40	70	
18	Rolf	1,78	40	74	
19	Klaus	1,68	61	80	
20	Klaus	1,68	61	82	
21	Klaus	1,68	61	79	
22					
23					

EXCEL 中的扁平排序表:中间结果在解决了权重列表后包括更多的数据行。

在这个表格中插入了一个新的列 BMI_x，其单元格中的 BMI 值根据公式=D2/B2^2 计算。字段 E2 的公式必须被复制到 E 列的所有单元格。这样做，公式会根据选定的行自动调整。

BMI3 列的计算公式为=AVERAGE(E2:E3)。必须注意正确选择用于计算平均值的范围。

同样，BMI2 的值也是根据有关地区和 BMI 计算的。

H2								
	A	B	C	D	E	F	G	H
1	NAME	LAENGE	ALTER	GEWICHT	BMI _x	BMI3	BMI2	BMI
2	Bert	1,72	18	66	22,3093564	22,1403461	20,8776213	23,2622421
3	Bert	1,72	18	65	21,9713359			
4	Kathi	1,7	18	55	19,0311419	18,5121107		
5	Kathi	1,7	18	52	17,9930796			
6	Klaus	1,68	18	61	21,6128118	21,6128118		
7	Klaus	1,68	18	60	21,2585034			
8	Klaus	1,68	18	62	21,9671202			
9	Bert	1,72	30	70	23,6614386	23,8304489	23,3502068	
10	Bert	1,72	30	71	23,9994592			
11	Klaus	1,68	30	65	23,0300454	23,0300454		
12	Klaus	1,68	30	63	22,3214286			
13	Klaus	1,68	30	67	23,7386621			
14	Kathi	1,7	40	70	24,2214533	24,3944637	23,3924276	
15	Kathi	1,7	40	71	24,567474			
16	Rolf	1,78	40	72	22,7244035	22,7244035		
17	Rolf	1,78	40	70	22,0931701			
18	Rolf	1,78	40	74	23,3556369			
19	Klaus	1,68	61	80	28,3446712	28,462774	28,462774	
20	Klaus	1,68	61	82	29,053288			
21	Klaus	1,68	61	79	27,9903628			
22								

在 EXCEL 中含有 BMI 值的表格: BMI_x、BMI3、BMI2 和 BMI 这几列被添加并按此顺序确定现在可以用 "格式化单元格"来定义小数点后的位数，不再需要的列可以在结果显示中隐藏。然后, 计算结果在 EXCEL 中显示如下。

	A	C	F	G	H	
1	NAME	ALTER	BMI3	BMI2	BMI	
2	Bert	18	22,14	20,88	23,26	
4	Kathi		18,51			
6	Klaus		21,61			
9	Bert	30	23,83	23,35		
11	Klaus		23,03			
14	Kathi	40	24,39	23,39		
16	Rolf		22,72			
19	Klaus	61	28,46	28,46		
22						

在 EXCEL 中的最终输出: 计算结果是一个格式化的表格

2.3. 使用 Excel 的解决方案 (自动)。

该解决方案只是在从 'gib-help-sort' 过渡到 'gib' 时有所不同。另一个工作表 'gib-avg-help' 用于标记包含 ALTER 和 NAME 信息的行，例如 CONCATENATE(TEXT(\$BMI.A2; "##, ##"); \$BMI.B2)。

这导致了 BMI 值的新公式。

- `bmi=average($bmi.e2:e50)`
- `bmi2=if(b2=b1;"";averageif(bmi.a2:a50; b2;bmi.e2:e50))`
- `BMI3 =IFERROR(IF(AND(B2=B1;D2=D1);"";AVERAGEIF('gib-avg-help'. B2:B50;CONCATENATE(TEXT(B2; "##,##");D2);BMI.E2:E50);""`

工作表仍然包含没有 BMI 值的冗余行（工作表'gib-holes'）。

这些可以用与 "avec-holes"（工作表 "gib"）中的空白线相同的方式进行标记和选择。之后，按照惯例进行四舍五入（工作表'rnd'）。

	A	B	C	D	E
1	BMI	ALTER	BMI2	NAME	BMI3
2	23,26	18	20,88	Klaus	21,61
3				Kathi	18,51
4				Bert	22,14
5		30	23,35	Klaus	23,03
6				Bert	23,83
7		40	23,39	Rolf	22,72
8				Kathi	24,39
9		56	28,46	Klaus	28,46
10					
11					

工作表 BMI2 输出。最终的输出表有四舍五入的数值

3. 评价

与使用电子表格工作相比,如上例所示,需要一连串精心格式化的工作表和中间结果以及大量的用户输入,而++o 可以用一个由 4 行组成的程序来解决这一任务,并且可以修改。

电子表格很受欢迎,因为内容可以被交互式地修改,而且结果可以立即看到。即使是像上述例子中的简单应用,也需要大量的工作和对软件工作方式的精确了解。o++o 以更少的时间和精力,生成一个可以交互修改的程序,并提供立即可见的结果。此外,可能的修改意义更为深远。

o++o 使计算具有高度的灵活性,而适应性的努力很少。例如,在行中

给予 BMI, (AGE, BMI, (NAME, BMI m) m) BMI:=WEIGHT:LENGTH:LENGTH!++:

只有 ALTER 和 NAME 被交换,以实现一个完全不同的请求。

用 EXCEL 计算 BMI 值的扩展例子的手动解决方案需要以下步骤。

1. 在工作表中创建表列, 输入 NAME、LENGTH、AGE 和 WEIGHT 的值。
2. 手动过滤, 因为菜单功能 "过滤" 没有足够的表现力
3. 填充空单元格, 以便可以应用排序菜单功能, 进行后续排序
4. 在 BMIx 列的顶部单元格中输入 BMI 计算公式
5. 将此公式复制到这一列的所有单元格中 (在给定的例子中, 有 20 个单元格)。
6. 输入计算均值 BMI3 的公式, 并将此公式复制到另外 7 个单元格中。
7. 调整这些细胞中的区域
8. 输入 BMI2 和 BMI 的公式, 并将 BMI2 的公式复制到另外 3 个单元格, 另外调整范围
9. 使用 "格式化单元格" 菜单功能
10. 隐藏结果显示中不需要的 3 列

因此, 首次创建这个工作表需要使用 3 个菜单函数, 并在要计算的单元格中输入 4 个公式。这些公式必须被复制到其他 29 个单元格。此外, 公式的有效范围必须在 12 个单元格中调整。

如果在这个例子中加入更多的数值, 必须在这个工作表中调整公式的有效范围。

使用 EXCEL 的自动解决方案需要在几个工作表中进行类似的步骤, 但不需要在输入数据改变时对公式进行后续调整。

另一方面, 用 ++o 的解决方案, 只需要。

1. 表的输入
2. 输入一个由 2 或 3 行组成的 ++o 程序。
3. 结果可以以各种显示格式调用, 包括图形输出, 在每种情况下只需点击一次 (或两次)。

我们用 ++o 解决这个任务的努力还不到 EXCEL 所需时间的 20%。对于这个假设, 我们假定了对 EXCEL 和 O++o 系统的了解。然而, 为了更准确和普遍地估计工作量的差异, 需要进行更广泛的研究。

4. 一般性比较

在所选的例子上的比较没有触及许多其他方面。我们注意到以下独立于上述例子的 EXCEL 和 ++o 的相关区别。

1. o++o 表格总是有一个表头 (用于水平维度), 通常用其列进行计算。水平维度由图元 (一对泛化) 描述, 而垂直维度由集合 (列表、集合、多集合.....) 指定。
2. 在 EXCEL 中输入数据可能比在 O++o 的 Tab 文件中输入数据要快, 因为后者需要输入适当的空格。使用 hsq 文件可以避免这种情况。然而, 对于初学者来说, 这些内容有些难以阅读。
3. 你必须在 EXCEL 中把每个数字写在一个单独的单元格中。这很快就填满了 (智能手机的) 屏幕。在 ++o 中, 以 tab 格式输入时有一个类似的问题。适合初学者的显示器往往直接显示表格格式, 并占用相应的空间。
4. 由于 EXCEL 不知道任何相应的模式, 所以不能直接处理 XML 数据。

5. 你不能用 EXCEL 的概念来对数据库、XML 进行查询，.....与 EXCEL 的概念。你必须学习完全不同的语言（SQL，XQuery，...）。
6. 理解 EXCEL 工作表是非常困难的，因为一个 ++o 公式通常对应着几个 EXCEL 公式，而且这些 EXCEL 公式分布在许多单元格中，并可以用引用来连接。由于缺乏清晰度，EXCEL 工作表很容易出现错误，而且现有的错误很难纠正。
7. ++o 是基于数学概念和集合操作，如集合、列表、元组、理货操作，.....。对 EXCEL 的处理知道的概念不多，但需要很多详细的知识（何时以及如何调整复制的公式；在插入或删除行或列时是否也要进行调整，等等）。
8. 由于 ++o 程序包含的公式很少，所以更容易阅读和修改。
9. 数据和程序通常以 O++O 分开。因此，数据可以被几个程序使用而没有任何问题。
10. ++o 条件，例如，只选择数据而不是公式。例如，对于一个简单的条件 "avec LAENGE>1.60 "应用于我们的输出表，在 EXCEL 中没有相应的过滤条件，因为整个人的记录（每条几行）包括子（ALTER,WEIGHT）对被删除。一个条件 "avec NAME!WEIGHT>70 "即使在 "扁平化 "表之后也不能被映射，因为它 "包含 "一个存在量词（我们正在寻找所有包含一个体重大于 70 的人）。尽管 ++o 条件比 EXCEL 过滤条件更有表现力，但它们更容易跟踪，因为不会发生与单元格公式的碰撞。
11. 对于每个值的聚合（总和、平均数、最大值.....），必须在 EXCEL 中进行预排序或分组，而不是在 O++O 中。
12. ++o 可以直接处理结构化表格的内容方面。这些通常比平面表格更接近于所需的印刷图像。为了创建更具体的打印图像，人们可以通过 XML 输出数据，并使用样式表语言。而在 EXCEL 中，内容问题和形式问题并没有分开。
13. ++o 是基于一个抽象的数据表格概念。一个 tabment(=TABelle+dokuMENT) 可以用多种方式(tab、xml、graph、diagram...) 和紧凑的方式(hsq) 具体表示。EXCEL 是基于一个工作表的二维具体视图。该视图包括数据和公式。
14. EXCEL 公式比 ++o 操作更难理解。一个单一的公式可能比一个完整的 ++o 程序需要更多的分析。