

软件使用指南

 **ROV Modeler**

 **ROV Optimizer**

 **ROV Valuator**

**R I S K**  
**R I S K**

# 目录

软件介绍 .....	3
系统要求.....	3
版权和联系信息.....	3
<b>Risk 模型 Modeler .....</b>	<b>4</b>
<b>变量映射 .....</b>	<b>6</b>
数据连接.....	6
技巧:保存为一个 CSV 文件格式.....	6
手动输入.....	7
技巧:手动输入数据.....	8
数据计算.....	8
设置模拟假设.....	9
数据拟合.....	10
技巧:变量管理.....	10
技巧:使用 ROV Portfolio 来运行多个模型.....	11
技巧:保存 Profiles,数据分离器,输出数据库和语言.....	12
技巧:自定义用户界面和功能模型列表.....	13
技巧:样例模型和高级功能.....	14
<b>Risk Valuator.....</b>	<b>16</b>
<b>Risk Optimizer.....</b>	<b>19</b>
连接到其他数据库.....	23
案例一:连接到 Oracle .....	23
案例二:连接到用户 DSN 中.....	23
<b>ROV Scheduler, ROV Portfolio, ROV Charter .....</b>	<b>24</b>
无用户界面运行 XML.....	26
与其他系统相互集成.....	29
<b>附录: SQL 使用案例 .....</b>	<b>34</b>
用户案例 1: 通过值来选择一排数据.....	36
用户案例 2: Use of '和'.....	37
用户案例 3: Use of 'OR'.....	38
用户案例 4: '和' 和 'OR' 共同使用.....	39
用户案例 5: 'IN' 的使用.....	40
用户案例 6: 'BETWEEN' 的使用.....	41
用户案例 7: 'LIKE' 的使用.....	42
用户案例 8: Simple Math Functions.....	43
用户案例 9: 数学公式的多种使用.....	44
用户案例 10: 'Union' 的使用来连接命令行.....	45
用户案例 11: Filtering Different Value Types.....	46
用户案例 12: 选择最上面的 N 排数据.....	47
用户案例 13: 'NOT IN' 的使用.....	48

用户案例 14: 'EXISTS' 的使用.....	49
用户案例 15: 多个表格的使用.....	50
用户案例 16: 简单的范例.....	51
用户案例 17: 使用 And 的范例.....	52
用户案例 18: 使用 Union 和排序的案例.....	53
用户案例 19: 使用数学的案例.....	54
用户案例 20: 重复使用 And 和 OR 的, 数学的案例.....	55
用户案例 21: Use of 'UNION ALL'.....	56
用户案例 22: SQL 功能的用法.....	57
用户案例 23: 'GROUP BY' 的用法.....	58
用户案例 24: 'DISTINCT' 的使用.....	59
用户案例 25: 'ORDER BY' 的使用.....	60
用户案例 26: 通过日期'BETWEEN' 进行选择.....	61

## 软件介绍

本帮助文件将介绍由 Real Options Valuation 研发设计的 ROV Risk 模型 Modeler. 软件将 Excel 中的模型转化到数据库环境中,允许终端用户能够直接连接到数据库和大数据文件,清理数据后高速处理分析数据.ROV Risk 模型 er 包括以下基本模块:

- ROV 模型 Modeler 是一个可定制化的模型分析组件,用来进行多种类型的模型分析,包括计算不同行业的分析模型,预测和仿真模型,历史数据拟合,时间序列预测(ARIMA, 自回归,回归,随机过程等),波动率计算(VARCH)以及其他应用.组件同时拥有从不同数据库中连接和下载数据的功能(例如 Oracle OFDM ,SQL server,Excel,CSV,文本文件和其他 ODBC 相关数据库),能够清洗整合数据(运用 SQL 命令行和数据清理规则),基于现有数据计算新的变量,运行蒙特卡罗模拟,运行数据分布式拟合等.组件具有可定制性.用户可以修改功能列表,描述,模型功能的界面显示,允许用户定制工具来满足自身的实际需求.
- ROV Basel 模型 Modeler 是一个高级分析组件,用来处理多种类型的模型,包括行业模型(例如银行,保险公司,金融服务公司),违约率 PD,违约损失率 LGD,违约风险暴露 EAD,风险价值 VaR 等关键因素.功能上,与 ROV 模型 er 基本保持一致.
- ROV Risk Optimizer 能够运用仿真和随机优化技术进行项目的选择投资组合的非线性优化.在这过程中可以选择整型,二次型和连续变量来进行多种限制.
- ROV Risk Valuator 拥有 600 多个模型来进行商品,期货,资产组合风险回报的简单期权和奇异期权的评估.请参见附录中所有模型的详细列表.
- ROV Risk Charter 运行模型 er 和 Optimizer 中生成的不同 profiles,并产生 XML 文件.这些 XML 文件被 ROV Dashboard 运行之后,能够生成动态图表,表格,导航图和报告.ROV Dashboard 是 Real Options Valuation 公司开发的另一种软件.
- ROV Scheduler 运行模型 er 和 Optimizer 中生成的不同的 profiles,并返回文本文件结果.结果能够保存并上传到 Excel 或者是其他数据库中.
- ROV Portfolio 同时运行模型 Modeler 和 Optimizer 的 profiles 和模型.这类似于 Risk Scheduler,不同的模型可以从不同的 profile 中取出同时运行.区别在于分析可以在既定后的一段时间内运行

## 系统要求

软件能够在任何 Windows 或 MAC 的环境下运行(MAC 系统要求支持并行处理或者虚拟机来模拟一个 Windows 操作系统),需要能够兼容 Microsoft Excel 和 ODBC 数据库及数据文件.软件要求 100M 的硬盘空间,推荐至少 1G 的内存.建议用户拥有系统管理员权限(一般系统默认),但也能够运行在受限制权限的用户上(一般安装软件在非保护的文件夹内,以便正常运行软件)

## 版权和联系信息

ROV Risk 模型 er 软件由 Real Options Valuation 公司研发设计. Copyright 2008-2009 由 Dr. Johnathan Mun 先生拥有. 软件受美国版权法律及国际协议保护.任何未授权的复制,盗版,软件发布将追求严肃的法律责任.

联系软件开发者:  
Real Options Valuation, Inc.  
4101F Dublin Boulevard, Suite 425  
Dublin, California 94568 USA  
[admin@realoptionsvaluation.com](mailto:admin@realoptionsvaluation.com)  
电话: +1.925.271.4438 传真: +1.925.369.045  
[www.realoptionsvaluation.com](http://www.realoptionsvaluation.com)

## Risk 模型 Modeler

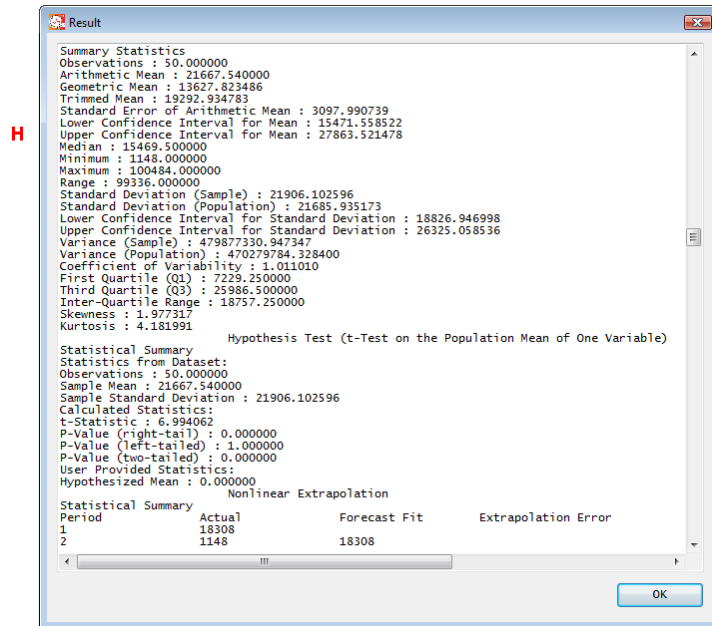
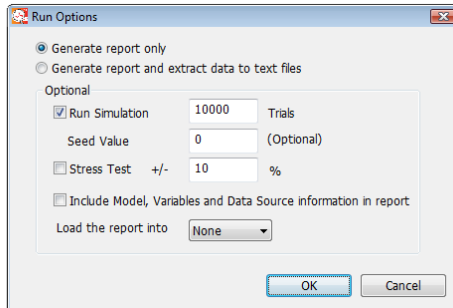
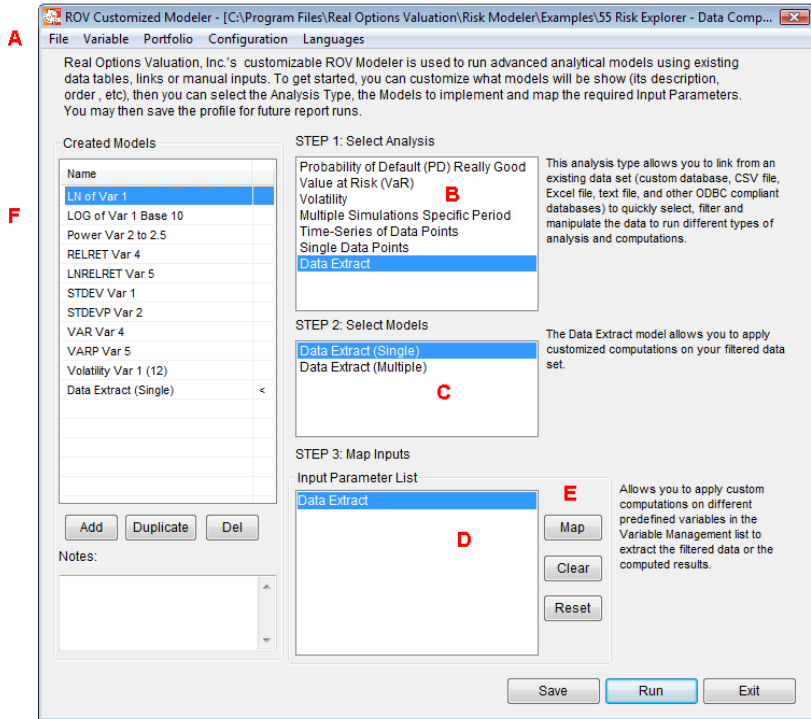
ROV 模型 Modeler(以下简称 ROV 模型 er)是一个风险模拟和高级分析工具,它提供模型来模拟,拟合,计算并报告结果给用户.软件与不同的 ODBC 数据库相互兼容,允许用户连接数据库,手动输入数据,设置模拟假设来的到运行分析所必须的数据输入.此外,软件具有可定制性,模型,描述和运用的分析都是可定制的(用户能够改变方法的名字,描述或者删除整个模型).软件可以脱离 Excel 或者其他数据库来运行,计算速度非常快.比如,运行一个存储有几百万个数据的 Excel 数据表,打开这个文件可能需要 10 分钟到 30 分钟的时间,在 Excel 中改变一个计算需要同样多的时间.如果需要重复这个过程(诸如每个 Excel 文件需要一个月或者一周,需要将分析计算每个月或者每周复制一次,这个过程在 Excel 中是无法进行的).除此之外,大多数数据库软件都只能存储数据而不能进行数据分析或者数据操纵.Risk 模型 er 是数据库的中间形态表现在它能够进行数据的内部计算.而这些计算是能够重复进行和预先计划的.

当您打开 ROV 模型 er 之后,你会看到如下图所示的用户界面.简而言之,软件界面拥有一个菜单栏 [A], 一系列分析类型和模型组[B], 分析类型下的模型列表[C], 选定模型下的输入[D], 映射或连接模型中变量[E], 在单个 profile 中创建不同的模型[F]. 举例来说,分析的类型可以是 Volatility [B]. 拥有不同的方法来计算一系列时间序列数据的 volatility,包括使用 GARCH 模型,对数模型,指数权重平移平均模型等等,这些模型都是在模型列表之中的[C]. 基于模型的选择,要求的数据输入是不同的[D].对于每个变量数据输入,可以进行数据映射[E]. 你可以选择需要进行计算的模型,这些模型可以通过 Add 按钮来添加新的模型[F].每一个列表中新建立的模型可以独立运行(选择模型后单击 RUN 按钮),仿真,压力测试和报告也能够相应生成[G].分析运行得非常快,结果会立即显示出来[H].

以下总结了运行 ROV Risk 模型 Modeler 的基本步骤:

1. 选择分析类型 [B]
2. 选择模型类型 [C]
3. 每次选择一个需要输入的变量[D]映射已有数据或者手动输入数据[E]
4. 选择需要运行的模型 [F] 运行分析 [G]

ROV 模型 Modeler 一次可以运行一个模型. 如果希望从一个或多个 profile 中运行不同的模型,可以使用 ROV Portfolio.如果希望模型在特点的时间或特定次数来运行,可以使用 ROV Schedule.



## 变量映射

当你选择 **MAP** 按钮[E], 将会出现一些选择来将数据映射到输入变量中[I]. 具体功能包括连接数据到已有数据文件或数据库中, 手动输入数据, 使用数据之前进行变量修改和计算, 设置假设来进行仿真模拟, 或者对一个大容量数据进行分布拟合寻找最优模型. 以下将详细列举变量映射的类型以及如何进行操作.

## 数据连接

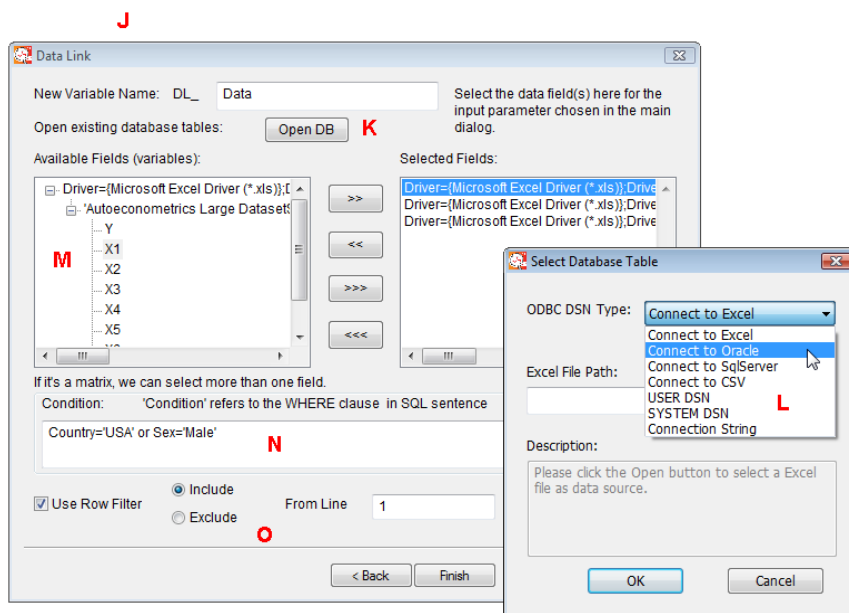
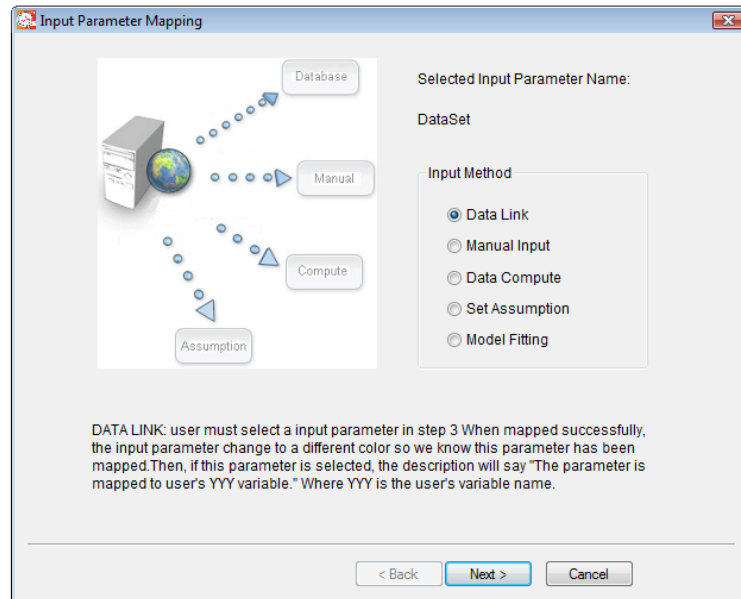
当选择数据连接选项后, 会显示出几种 ODBC 方式来进行与已有数据的连接[J]. 每个变量拥有一个变量名(默认情况下每个变量名拥有前缀“DL”, 这样的话可以识辨一个变量是否已经连接数据了), 然后选择“数据连接”选 [K]. 可以看见七种 ROV 模型 er 支持的数据库或数据文件类型, 包括 Excel 文件, Oracle 数据库, SQL Server 文件, 普通的 CSV 数据文件和其他符合 ODBC 规则的文件(DSN 格式). 例如, 选择一个“连接到 Excel”选项[L], 单击“浏览”按钮选择需要连接的 Excel 文件, 单击 OK. 在跳出的可选择窗口中[M], 可以看到工作表, 在每个工作表中, 显示了变量名. 一般来说, 当使用 Excel 作为数据存储介质的时候, 请将变量名存储在工作表的第一排. 接下来便可以通过单击>>选择希望使用的变量名, 也可以单击>>>将所有变量一次性选中(相反, 你可以从已经选中的变量中通过单击<<来移除变量, 或者单击<<<来移除所有变量).

软件还有其他数据连接来源的例子. 对于 Oracle 数据连接选项, 需要输入对应的用户名和密码来连接到数据库中. 当单击 OK 按钮, 软件会调用数据库连接方法实现连接到数据库中. 对于不同的 ODBC 应用, 软件的能够实现常用数据库操作, 包括 CONNECT, QUERY 等等. 你也可以使用 SQL 基本命令行来进行数据清理[N]. 关于 SQL 命令行的详细介绍, 可以参见附录中 SQL 的范例. 最后, 软件还有一个简单的数据筛选工具, 能够快速将某些数据导入导出.

## 技巧:保存为一个 CSV 文件格式

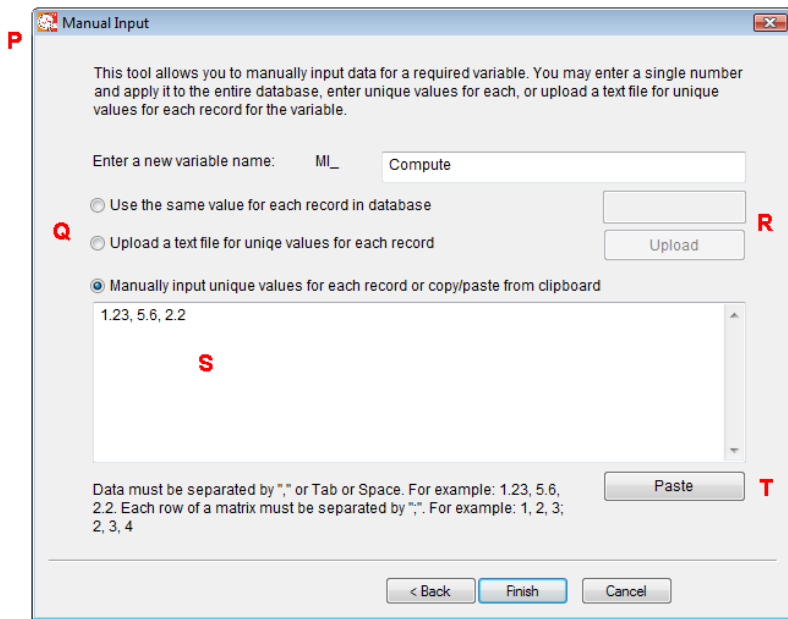
CSV 文件格式是最经常使用的文件格式(作为上传下载数据的文件, 与许多数据库相互兼容). 以下是创建 CSV 文件格式的向导.

- 我们建议您始终将一个普通文件格式转化为 CSV 格式, 因为 CSV 格式有更多的特性而且数据可以快速被浏览. 只需要将这个普通格式文件打开, 另存为 CSV 格式就可以了.
- 对 CSV 文件进行操作时, 请确保不要增加任何的排, 数值或者在底端输入数据. 这是因为在 CSV 文件底端的任何操作都会被保存, 即使是删除任何数值. 如果您在底端做了任何计算, 请将整排选中, 运行 DELETE ROW(s)来取消所有被 CSV 文件保存的项目(这是因为被删除的数值依然包含空值). 删除 ROW 的操作非常关键否则 SQL 将包含空值, 可能会得出错误的计算结果.
- 数据文件的第一排存储变量名. 变量名可能有空格和其他特殊符号. 然而, 在一些数据库中, 特殊符号和空格可能不能被识别. 因此, 我们不建议将空格和特殊符号加入变量名中(比如避免使用 @, %, #, &, /等符号).
- 如果数据的第一排是一个整型数值(0, 1, 2, 等等), 请确保它是一个十进制数值. 有时候在一些使用 MySQL 和 SQL 脚本的数据库中, 这些数值可能会被识别为一个字符串而不是一个数值. 最好能确认一下类型. 通过在 EXCEL 中编辑 CSV 文件的十进制数来增加一些十进制数.



## 手动输入

第二种方法是手动输入.有时候除了连接到数据库,我们需要手动输入一些数据.在这种情况下,一般先选择手动输入选项[I].手动输入的默认前缀是“MI”[P].输入变量的名字,并选择是否愿意重复使用相同的数值[Q].(比方说,假设输入的模型需要 5 个输入变量,X1,X2,X3,X4,X5,每个变量有 100,000 个数据点,而且已经将变量 X5 映射好了,这时希望为所有的 100,000 个数据点赋予相同的数值 0.10 而不必每一次地单个输入),从文本文件中上传数据[R],或者手动输入数据 [S] (同时也可以从另外一个文件或者电脑剪切板中粘贴数据[T]).当一切完成之后,点击“完成”



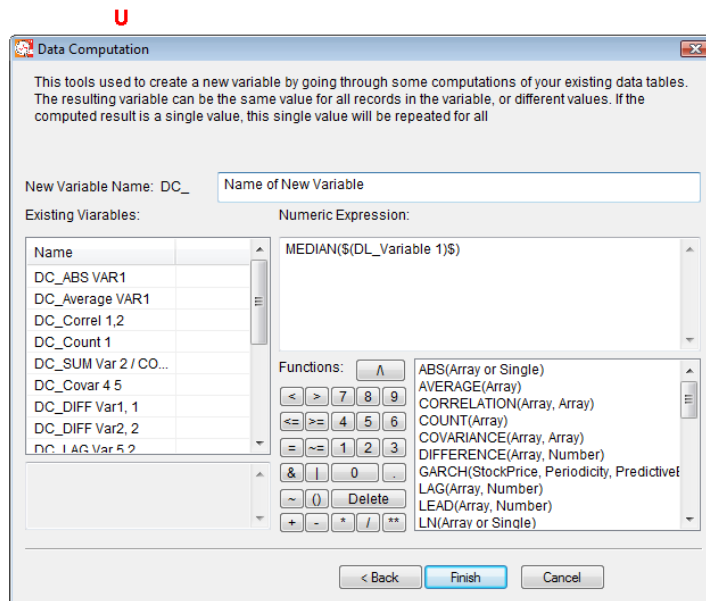
### 技巧:手动输入数据

在需要手动输入数据的时候,理解数据格式是非常重要的。

- 如果希望给单个变量输入数据,可以使用逗号作为数据之间的分割号.比方说,希望输入 4 个月的销售数据,他们是 54256, 222930.23, 111202.3, 132334,由逗号进行分离(您可以使用尽可能多的精确度)因此,不要将逗号作为千分位分割号.这样的话,一千点五应该表示为 1000.5 或者 1000.50 而不是 1,000.50,而且请不要使用任何货币符号,比如“\$”
- 如果变量要求输入不同的行时,分号是行与行之间的区分符号.比如说,假设需要输入一个 2x2 的矩阵(2 行 2 列),输入的数据为 1,0.1;0.1,1 逗号作为同一行中数据的分割,分号作为展开新一列的标识.您可以在 ROV Risk Valuator 软件中的 Value at Risk or Portfolio Risk 的模型计算中看到这样的例子.
- 系统中有默认的分隔符,您可以通过菜单栏中选择 Configuration 及选择 Data Separator 来改变分隔符.

### 数据计算

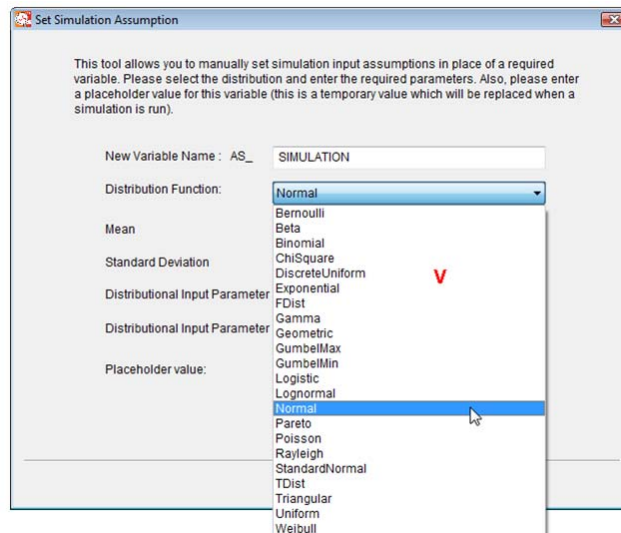
第三种方法是运用数据计算的方法到先前映射的数据上[U]. 比如,您可以映射没有用在模型中的其他变量,这些变量可以作为中间变量.(参见“变量管理”部分了解详细信息).举个例子,在截图中[U],我们可以看到数据连接变量“DL\_Variable 1”,一个和已有数据库连接的变量,作为一个中间变量.其数据是一个中间数据.先前已经映射过的变量显示在左边,称为已有变量.函数和数学公式在底端.您可以用数据计算方法来进行任何类型的计算。



## 设置模拟假设

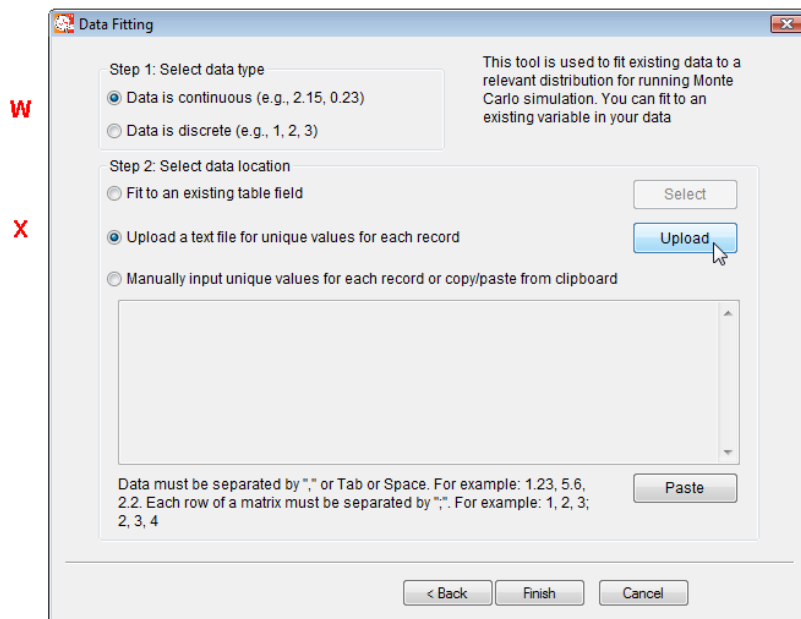
第四种方法对变量设定一种模拟分布假设 [I]。您需要首先选择分布,输入相关参数来设置适当的模拟假设 [V]。假设的默认前缀是“AS”。输入一个新的变量名,选择分布。软件共有 22 个分布可供选择。对于选择的分布,输入变量参数(比如说针对“普通分布”,输入平均值和标准差)。您同时需要输入一个临时存放数值(在您不运行模拟的情况下,这是计算模型时候的一个临时数值)。

请注意:限于篇幅,本用户文档无法列举出每个分布的详细细节。请参考 *Johnathan Mun* 博士撰写的“*模型 ing Risk: Applying Monte Carlo Simulation, Real Options Analysis, Stochastic Forecasting, 和 Portfolio Optimization*”一书 (Wiley Finance 2006 年出版), 来详细了解每个分布的含义,如何运用正确运用每一个分布。同时,您也通过网站[www.realloptionsvaluation.com](http://www.realloptionsvaluation.com) 中的“下载”一栏,下载 Risk Simulator 软件的用户手册。



## 数据拟合

第四种方法是数据拟合 [U]。有时候,您可能不能明确哪种分布是您希望运行模拟的分布.如果您已经拥有了数据,您可以使用数据拟合方法来决定最优的分布. 首先选择数据是连续的(比如 1.235, -12.23,)还是离散的(比如 -1, 20, 300) [W] 再选择数据存储的位置(选择您希望你拟合到一个存在的数据表位置,还是对每条记录上传一个文本文件作为单一值,还是对每条记录手动输入单一值或者复制/粘贴数据) [X].单击”完成”。

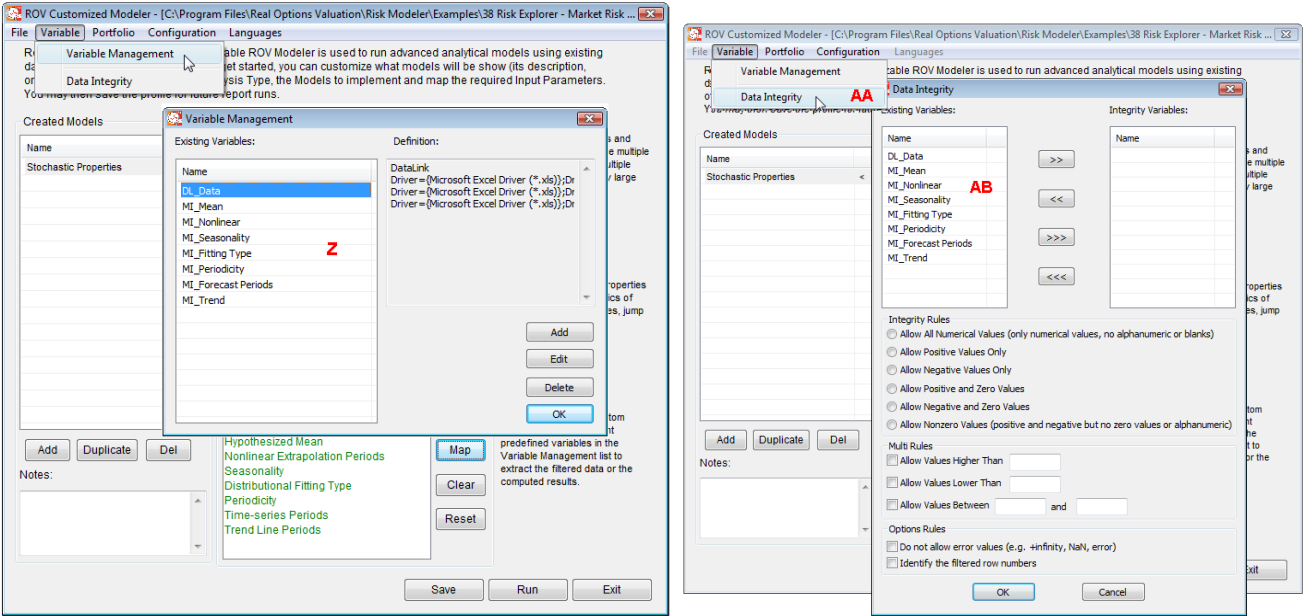


## 技巧:变量管理

变量管理工具是 ROV 模型 er 软件不可或缺的重要模块. 您可以单击菜单栏”变量”并选择”变量管理” [Y] 软件将显示已经被映射的变量列表 [Z]. 您可以选择”添加”,”编辑”,”删除”现有的变量.变量管理的功能在数据计算中是非常显著的,您可以从数据库中连接尽可能多的变量,进而进行对数据的操作.通过使用”数据连接”,”数据变量管理”和”数据计算”,您能够在数据使用前控制事件发生的顺序和操控数据.

除此之外,还有一个数据集成工具[AA], 可以进行数据的筛选.比如只筛选出正值,只筛选出负值,比一个值大或小.您也可以从之前映射过的变量中进行选择 [AB], 加到集成变量列表中,选择希望的规则

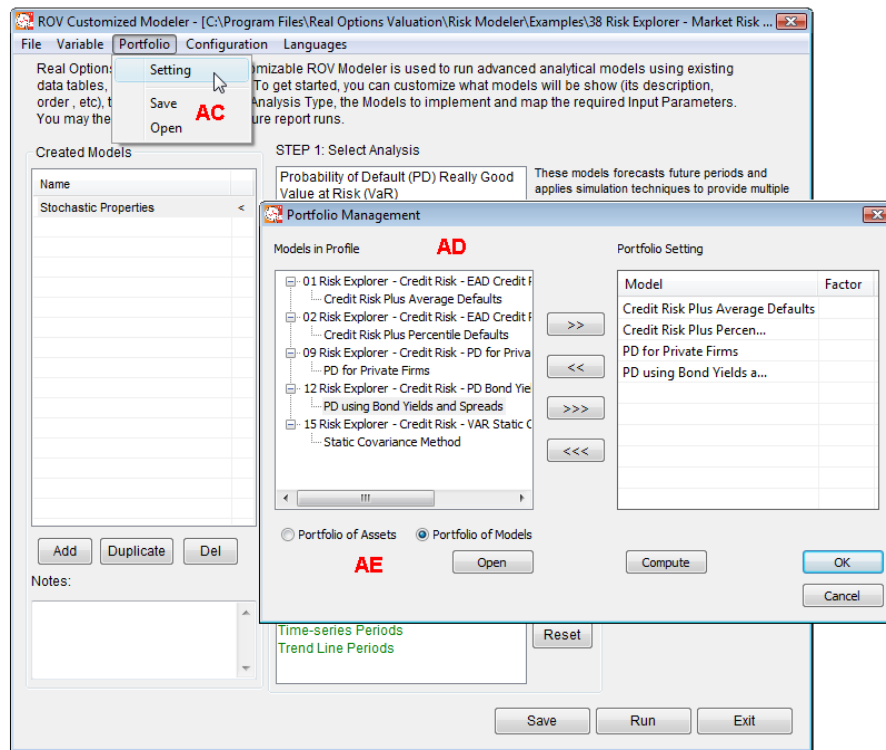
最后,请别忘记您还可以使用 SQL 命令行的方式进行数据筛选.使用 SQL 命令行的详细内容,您可以参见附录中的一些示例.



## 技巧:使用 ROV Portfolio 来运行多个模型

在 ROV 模型 er 软件中, 您可以建立多个模型并将其保存到单个文件中(我们称之为文件 profile). 当您需要运行一个模型的时候, 您需要从创建模型列表中选择模型, 一次运行一个模型. 有时候, 您需要在不同 profile 中创建不同的模型并同时运行. 这可以通过使用 ROV Portfolio 工具来实现 [AC]. 您可以通过开始, 程序, Real Options Valuation, ROV 模型 er 快捷方式或者在 ROV 模型 er 菜单中选择“文件夹”单击“设置”来打开软件[AC] 文件夹管理用户界面将会出现, 您可以通过单击“Open”来打开任何保存过的 Profile 文件. 所有 profile 中的模型都将列出来[AD]. 您可以选择希望运行的模型, 单击>>来增加模型或者>>>来添加所有模型, 反之亦然.

接下来您可以选择模型文件夹, 如果模型独立运行, 或者选择资产文件夹, 如果给予不同资产一定的权重 [AE]. 举个例子, 如果有 5 个模型需要计算, 需要 5 个独立的结果, 但是所有 5 个模型都是一次运行, 选择模型文件夹方法. 相反地, 如果您有 5 个资产结果, 希望计算所有资产的回报, 每个资产有一个权重(设置为 100%或者 1.0), 选择资产文件夹方法, 输入每项资产的权重.



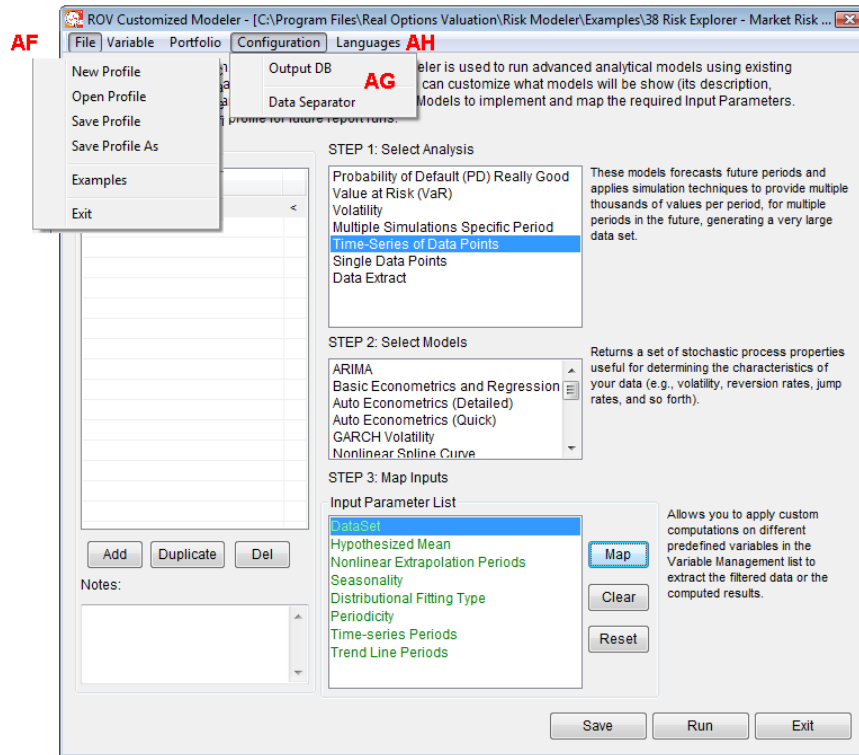
### 技巧: 保存 Profiles,数据分离器, 输出数据库和语言

在 ROV 模型 er 中, 您可以单击 File 菜单按钮[AF]来保存或者打开一个 profile. 请记住一个 profile 是一个 ROV 模型 er 中存储所有数据变量, 连接, 手动数据计算, 数据源地址, 使用的模型, 参数, 变量列表的文件.

软件中有 100 个 profile 例子, 您可以用来作为向导学习如何使用软件. 某些模型适用于不同的行业, 比如金融服务, 银行, 投资, 公司财务, 预测, 项目选择, 组合优化. 由于篇幅所限不能无法介绍所有模型的详细细节. 请参考 Dr. Johnathan Mun 博士的"Advanced Analytical 模型 s: Over 800 模型 s 和 300 Applications from Basel II to Wall Street 和 Beyond"一书来获取模型的详细信息.

除此之外, "配置"菜单[AG]允许您自定义输出的数据库, 比如可以直接将结果上传到数据库中. 除此之外, 您可以直接通过"数据分离器"菜单来定义分离.

最后, 还有一个"语言"菜单, 您可以转化不同的语言如英语, 中文, 日语, 西班牙语, 法语, 德语, 俄罗斯语, 意大利语. 请注意大多数的文字都可以翻译, 而一些技术术语无法翻译, 这些内容将依旧保持英语.



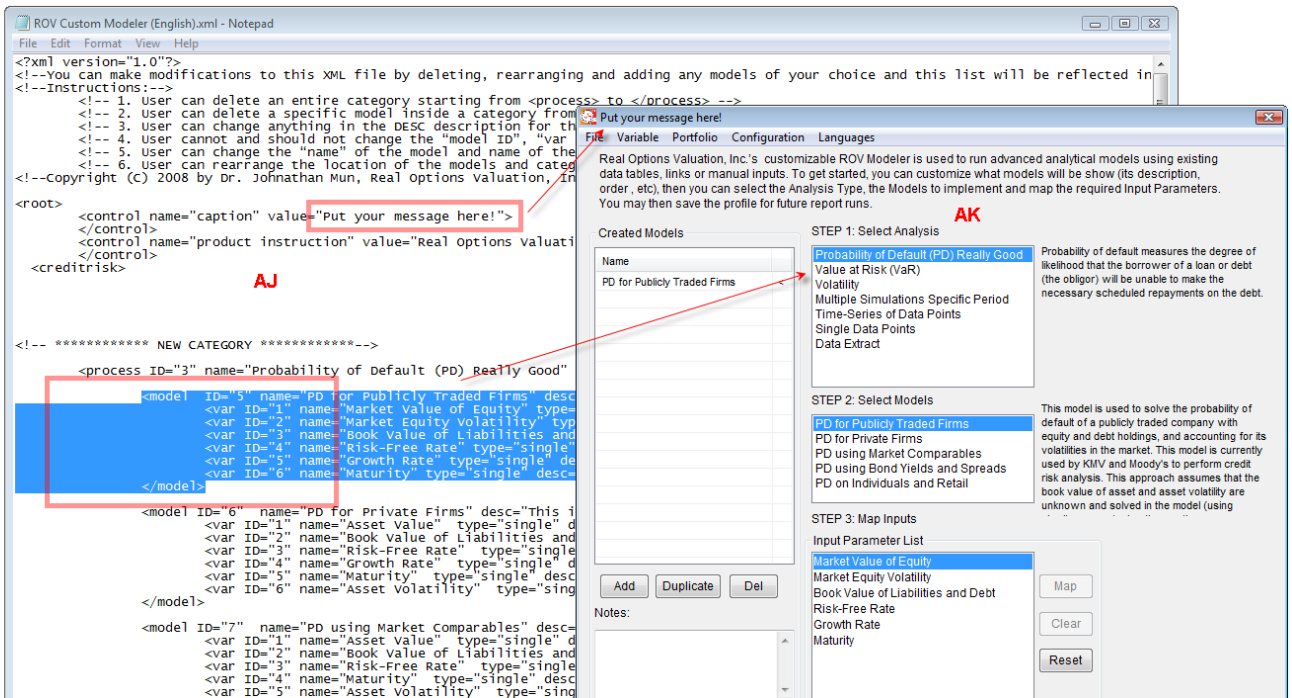
## 技巧:自定义用户界面和功能模型列表

在软件的安装过程中，会产生一个文件“ROV Custom 模型 er.xml” [AI].不同的语言会有不同的 XML 文件，英文版本命名为“ROV Custom 模型 er.xml” [AI].XML 文件控制着用户界面的名字和描述。您可以直接使用 XML 编辑器来编辑文件，或者使用记事本(打开记事本，拖动 XML 文件将其导入记事本中编辑) [AJ].在 XML 文件中，您可以做以下事情：

- 用户可以删除整个类型，从<process>到</process>.
- 用户可以删除类中的某一个模型，从<模型>到</模型>.
- 用户可以在 DESC 中改变任何描述性语言.
- 用户不能也不建议改变“模型 ID”，“var ID”，“type” 和 “param\_style” values.
- 只要 ID 值不改变，用户可以改变模型名称，参数名称.
- 用户可以将模型和类的位置改变到最上方.

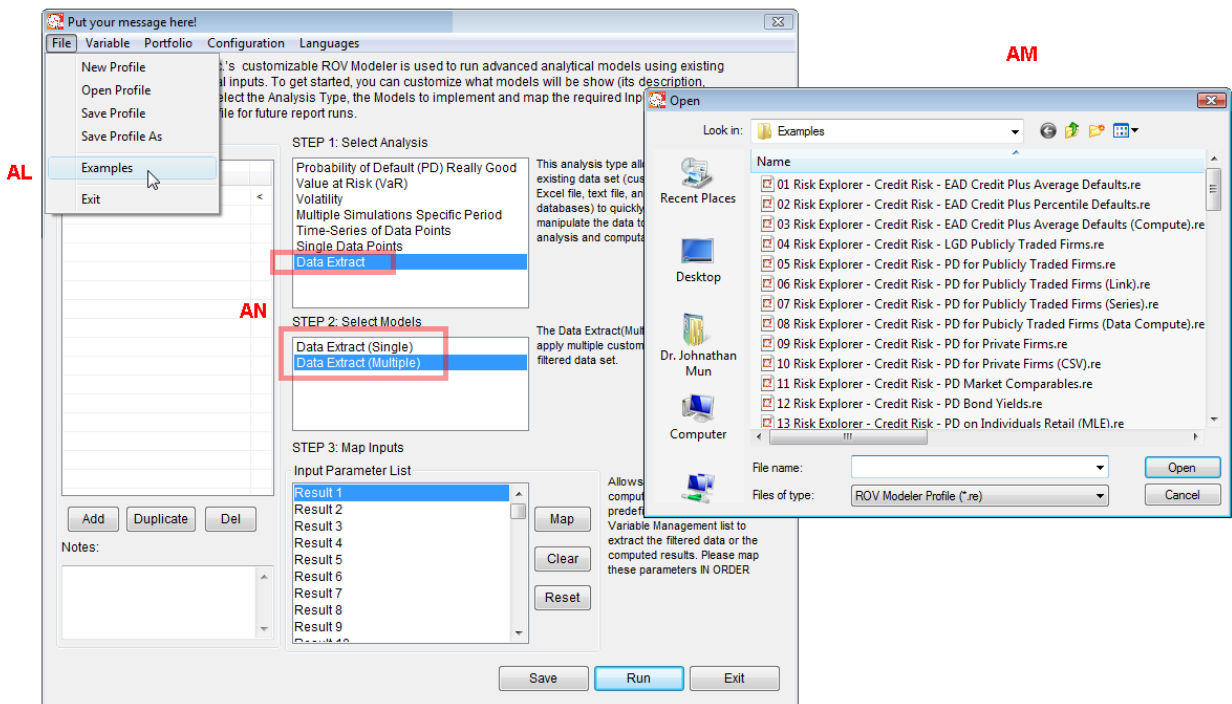
当您改变 XML 文件后保存[AJ], 然后开始 ROV 模型 er 程序, 您会看到变化结果[AK].

AI



### 技巧: 样例模型和高级功能

学习软件的很好的方式是了解样例 profile [AM]. 通过单击“文件”菜单，选择“风险分析样例”，您可以看到不同的[AL]. 常使用的关键功能有：数据抽取模型[AN](用来直接从数据库或者数据文件中抽取数据，进行运算并且返回分析结果).自回归模型[AO](在时间序列数据模型类型下，用来寻找数据的最优拟合模型),用户案例[AP]模型来了解如何使用 SQL 命令行[AQ](请参考附录范例来了解如何使用 SQL 功能调用).



Put your message here! - [C:\Program Files\Real Options Valuation\Risk Modeler\Examples\56 Autoeconometrics (Small Da...]

File Variable Portfolio Configuration Languages

Real Options Valuation, Inc.'s customizable ROV Modeler is used to run advanced analytical models using existing data tables, links or manual inputs. To get started, you can customize what models will be show (its description, order, etc), then you can select the Analysis Type, the Models to implement and map the required Input Parameters. You may then save the profile for future report runs.

**Created Models**

Name
Auto Econometrics 6 VAR
Auto Econometrics 5 VAR
Auto Econometrics 4 VAR
Auto Econometrics 3 VAR
Auto Econometrics 2 VAR

**STEP 1: Select Analysis**

Probability of Default (PD) Really Good  
Value at Risk (VaR)  
Volatility  
Multiple Simulations Specific Period  
**Time-Series of Data Points**  
Single Data Points  
Data Extract

These models forecasts future periods and applies simulation techniques to provide multiple thousands of values per period, for multiple periods in the future, generating a very large data set.

**STEP 2: Select Models**

ARIMA  
Basic Econometrics and Regression  
**Auto Econometrics (Detailed)**  
Auto Econometrics (Quick)  
GARCH Volatility  
Nonlinear Spline Curve

**STEP 3: Map Inputs**

**Input Parameter List**

Dependent Variable (Y)  
Independent Variables (X)  
P-Value Threshold  
Time-Series Lags

Allows you to apply custom computations on different predefined variables in the Variable Management list to extract the filtered data or the computed results. Please map these parameters IN ORDER

Map Clear Reset

Notes:

Save Run Exit

Put your message here! - [C:\Program Files\Real Options Valuation\Risk Modeler\Examples\79 Conditional Use Case 21-25.re]

File Variable Portfolio Configuration Languages

Real Options Valuation, Inc.'s customizable ROV Modeler is used to run advanced analytical models using existing data tables, links or manual inputs. To get started, you can customize what models will be show (its description, order, etc), then you can select the Analysis Type, the Models to implement and map the required Input Parameters. You may then save the profile for future report runs.

**Created Models**

Name
Use Case 21
Use Case 22
Use Case 23
Use Case 24
Use Case 25

**AP**

**Data Link**

New Variable Name: DL\_ **Data Extract**

Select the data field(s) here for the input parameter chosen in the main dialog.

Open existing database tables: **Open DB**

Available Fields (variables):

Selected Fields:

Driver=(Microsoft Excel Driver (\*.xls));Drive

If it's a matrix, we can select more than one field.

Condition: 'Condition' refers to the WHERE clause in SQL sentence

1 = 0 UNION ALL (SELECT SUM([Store\_Information\$].[Number]) FROM [Store\_Information\$] GROUP BY Store\_Name)

**AQ**

Use Row Filter  Include  Exclude From Line to

Notes:

Back Finish Cancel

Save Run Exit

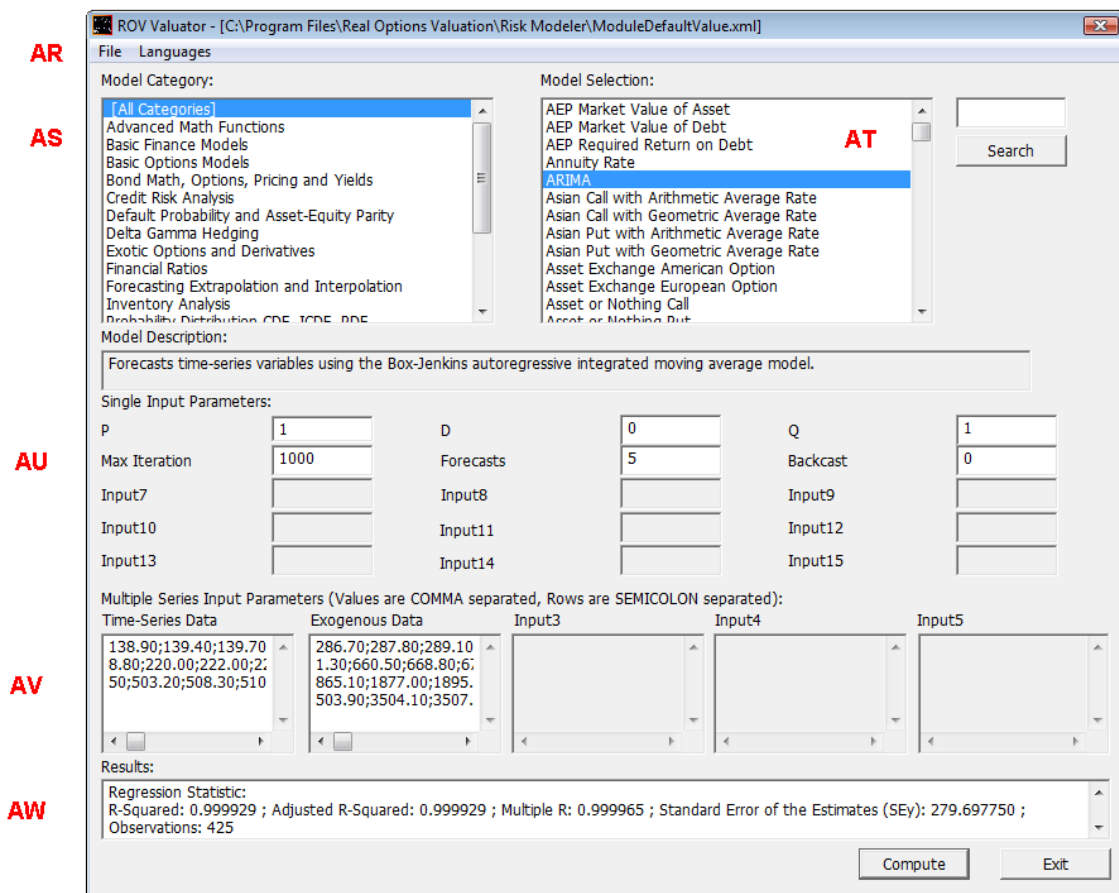
## Risk Valuator

Risk Valuator 拥有超过 600 个模型和函数，能够评估简单期权，奇异期权，商品，期货，资产组合风险回报等金融产品。

用户可以在所选择的模型中输入要求数据的同时便得到返回的结果。Risk Explorer 模块能够用来评估衍生产品，债务证券，奇异期权，期权嵌入金融工具和多种不同类型的金融模型。

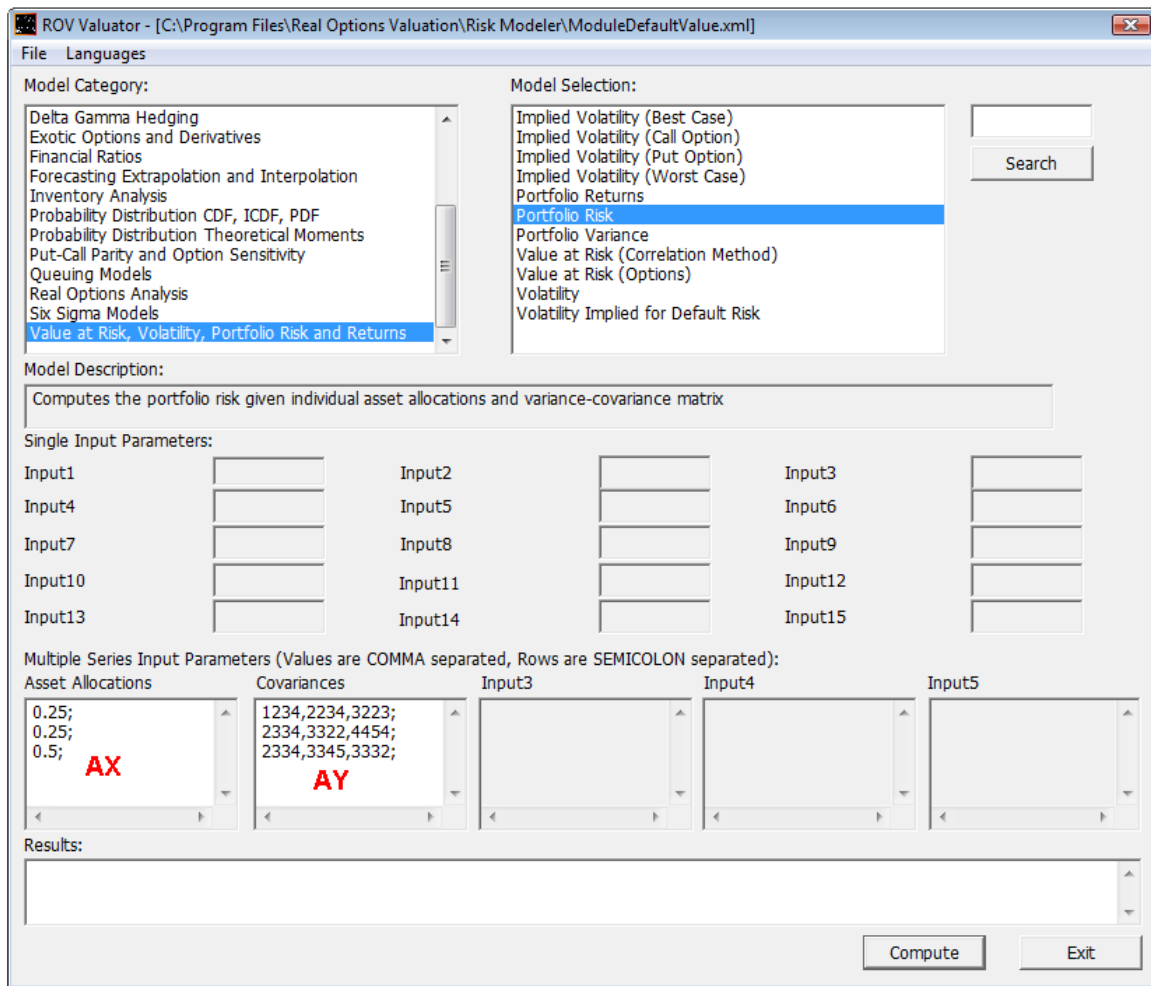
超过 600 种高级金融模型可以按照以下应用分类:

- 高级数学函数
- 基本金融模型
- 基本期权模型
- 债券数学，期权，产品定价，收益率
- 信用风险分析
- Delta Gamma 度量标准
- 奇异期权和衍生品
- 金融比率
- 预测, 外推法，内推法
- 可行性拟合
- 买入卖出校验和期权敏感性分析
- 实物期权分析
- 风险股价，波动率，资产组合风险回报



Risk Valuator[AR]是用于快速运算的工具，运算范围涵盖从简单基础模型到高级分析模型，能够处理单一数据或者系列数组。安装软件后，选择开始 Risk Valuator.在对话框中选择模型类别[AS]并在模型选择[AT]列表中选择感兴趣的模型。软件将会列出要求输入值的参数。单一数值的输入(诸如 10 或者 10.4532)将会显示为单一值的输入框[AU]，而多数值的输入将会显示为序列值的输入框[AV]。当输入序列值时，使用逗号或者空格来将数值分开(举个例子，一个 6 个月利率的时间序列可以输入为 0.12, 0.124, 0.112, 0.1, 0.09, 0.16 或者 0.12 0.124 0.112 0.1 0.09 0.16) [AW].

有些时候，诸如 Value at Risk 这样的模型使用使用一种标准的相关性方法，要求数据之间有相关性矩阵，诸如 Value at Risk.举个例子，有三种资产，目标是算出这种资产组合的 VaR，其中每种资产拥有各自的数量，某一时刻的波动率和资产之间和相关性矩阵。在这样一种情形下，数量和波动率将会是一串单行数据（在输入一个数据后单击回车，创建新的一行，设计一个新的资产类或者使用分行作为行分割器 [AX]），而相关性矩阵中单一列不同行将被逗号断开，而不同列将被分号断开 [AY]。Risk Valuator 模块不允许用户连接到不同数据库中，也不可以设置仿真。如果要做这些话，需要使用 Risk Explorer.有时候同一模型在两个软件中同时存在。Risk Explorer 模块用于不用连接到数据库快速得到数据。



您在使用软件的过程中，可以点击“文件”菜单，选择“上传范例输入”。然后选择一个模型类别并选择一个模型。你会发现输入值载入到软件之中。然后单击“计算”来得到结果。您可以使用范例输入作为指引。

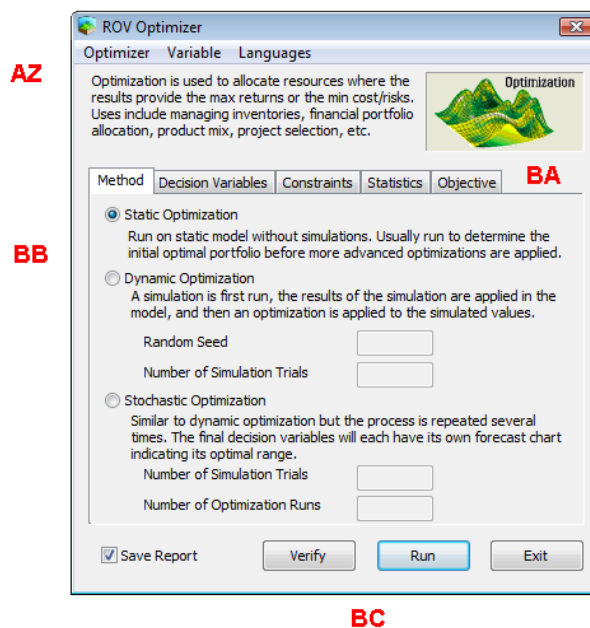
与 ROV 模型 er 软件一样，您可以定制 ROV Valuator 的模型列表与模型描述。通过路径 c:\program files\real options valuation\risk 模型 er 来打开软件安装路径，寻找到文件“ROV Custom Valuator (English).xml”，选择正确的文件 XML.文件控制着用户界面的名字和描述。您可以直接使用 XML 编辑器来编辑文件，或者使用记事本(打开记事本，拖动 XML 文件将其导入记事本中编辑)

1. 用户可以删除整个类型，从<category>到</category>.
2. 用户可以删除类中的某一个模型，从<function>到</function>.
3. 用户可以在“category name”, “displayname” 和“desc”中改变任何描述性语言.
4. 用户不能也不建议改变“function name”, “type” 和“param\_type”值
5. 用户可以但是不建议改变模型的“var name”值(可能存在范例数值被载入但是数据无效的风险)
6. 用户可以将模型和类的位置改变到最上方
7. 用户不必删除模型，建议您在重新需要模型的时候模型依旧存在您也可以通过使用模型中的范例自己创建模型类别

## Risk Optimizer

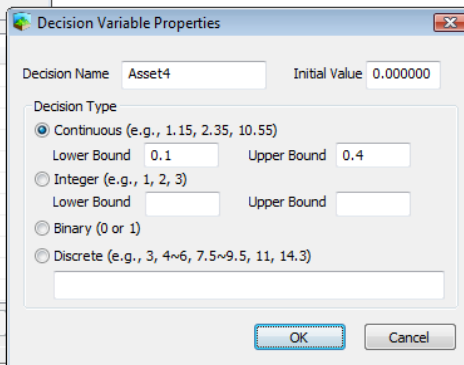
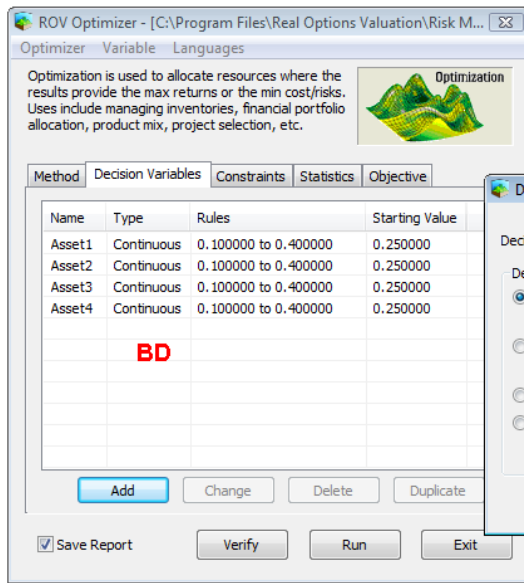
ROV Risk Optimizer 能够运用仿真和随机优化技术进行项目的选择投资组合的非线性优化。为银行，投资公司，制造型企业，研发机构提供服务。在这过程中可以选择整型,二次型和连续变量来进行多种限制，目标可以是线性或者非线性的。除此之外，Risk Optimizer 支持用户连接到已有数据库，运行仿真来实现最佳拟合模型，同时将这些功能融入到优化技术之中。详细细节请参考 Johnathan Mun 博士的“*Modeling Risk: Applying Monte Carlo Simulation, Real Options Analysis, Stochastic Forecasting, and Portfolio Optimization*”一书(Wiley Finance 2006)。

以下是一个如何使用 ROV Optimizer 的范例[AZ](建议您单击“文件”菜单，选择“模型”来上载先前定义的模型，学习模型的建立方式)。当您安装 Risk Optimizer 之后，您可以打开或者看见软件的用户界面。“方法”，“决策变量”和“限制”。选择“方法” [BA]并选择“静态优化” [BB]。对于静态优化，动态优化，随机优化的区别，请咨询我们的技术支持部门，或者浏览以上 Mun 博士的书，或者参加 Real Options Valuation 公司的培训研讨班。

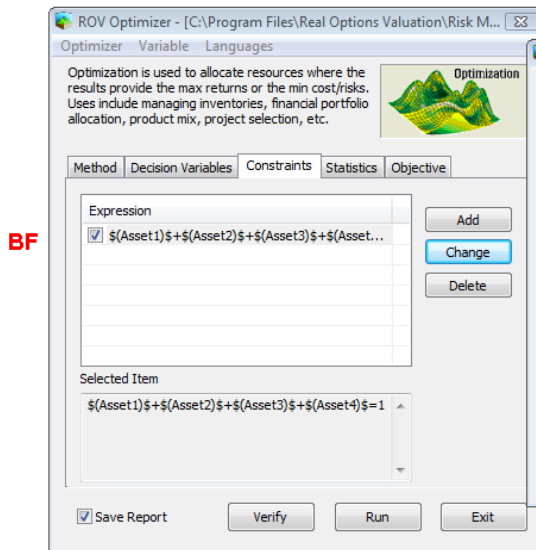


接下来，点击决策变量 [BA]. 然后按下 ADD 来增加一些变量。比如在如下的图表中我们有四个不同的变量[BD] (Asset1 to Asset4)，每一个变量可以设置成连续的，整型的，二项的或者连续的[BE]。方便简单的展示，我们将所有的变量设置成在 0.10 到 0.40 之间连续数值(即所有资产的分配只允许比例在 10%到 40%之间)。将四个变量都这样设置。

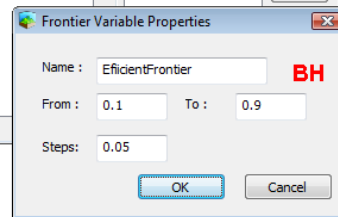
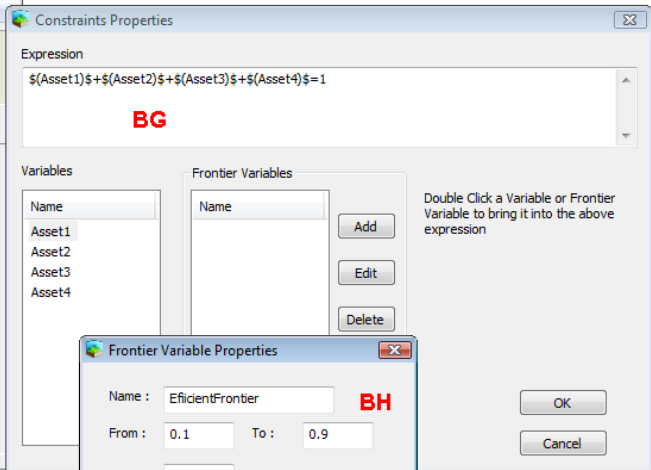
接下来点击“限制”标签并选择“添加” [BF]。在表达式输入框中，输入限制条件(您可以在图示的例子之中,决策变量的值加起来必须为 1.0，就是说再一个投资组合中，所有的资产比例加起来必须为 100%[BG]。您也可以通过添加一个“边界变量”来创建一个有效边界 [BH]。关于有效边界的细节，您可以参考 Mun 博士风险模型的相关书籍。



BE



BF

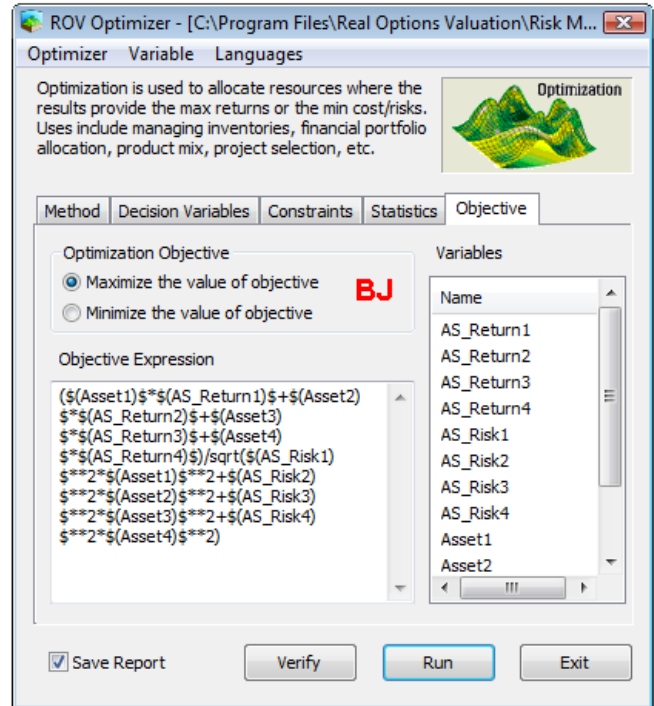
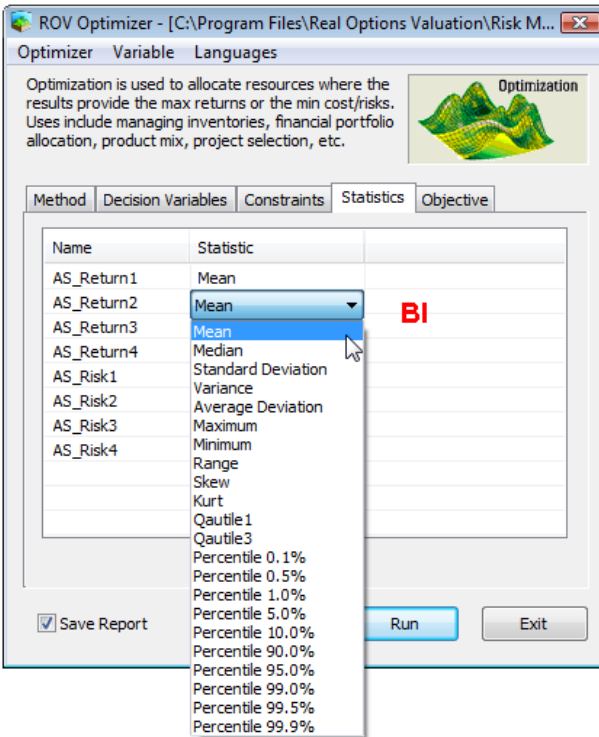


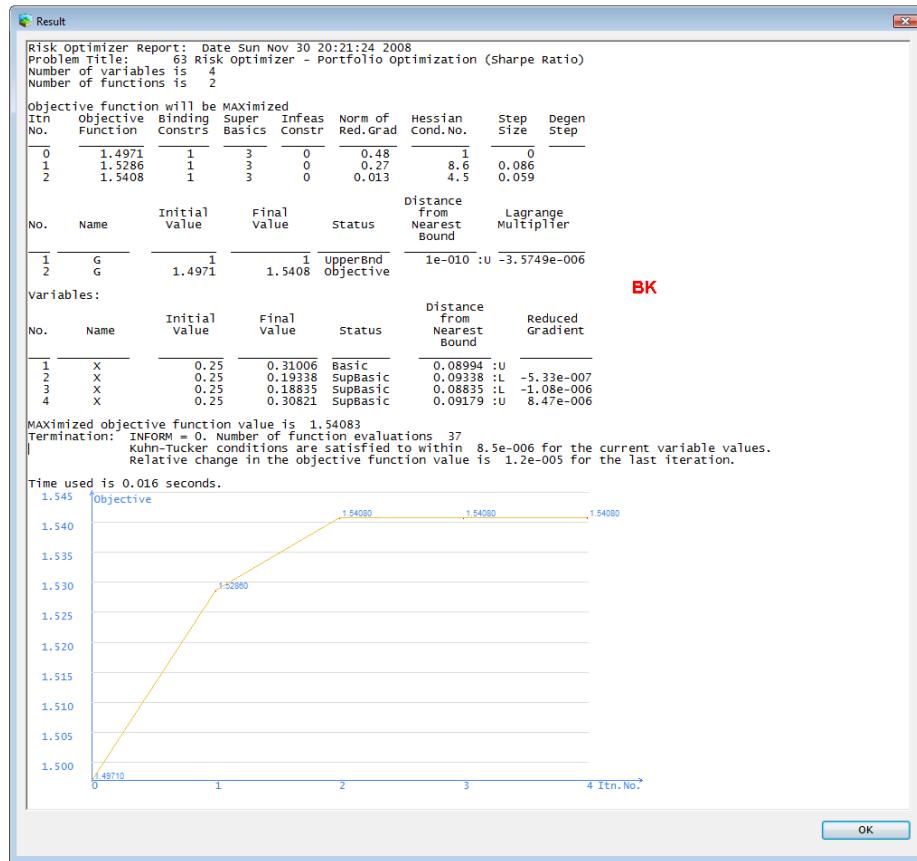
BH

除此之外，如果您使用了静态优化，您可以跳过“统计”选项。对于动态优化和随机优化，当变量映射到拟合分布时，“统计”选项是非常有用的[B1]。

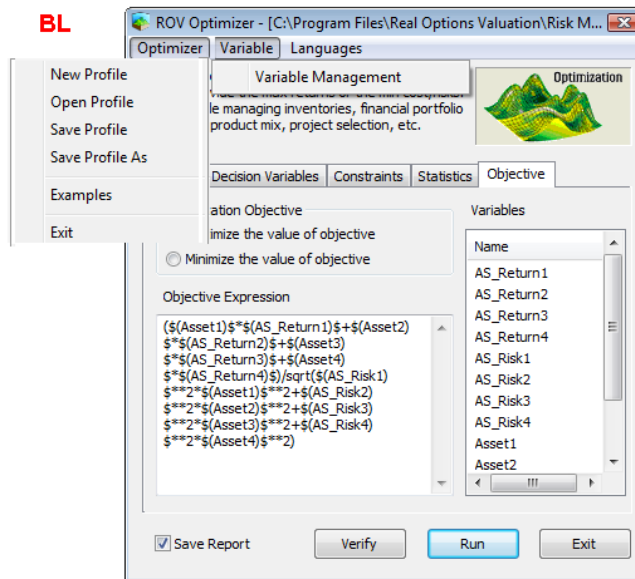
下一步，选择“目标” [B2]，同时还可以选择对目标选择最大值或者最小值。此外，可以对目标设定相关的表达式。你可以双击变量列表，为目标的表达式输入变量名字符串。

优化模型正确建立后，优化结果将会显示出来 [B3]。结果会显示循环的次数，模型的配置，参数，决策变量的初始和优化值，技术分析以及一个优化图表。





在 ROV Optimizer 文件菜单中有两个重要的功能，包括“范例”和“变量管理” [BL]. 变量管理工具允许您添加，编辑，删除变量. 比如说，通过单击添加按钮，输入参数映射工具将允许您连接，计算，粘贴，模拟，或者拟合现有数据，来进行优化过程. 最后，如果选择动态或随机优化，变量设置了风险仿真假设，您可以使用优化的统计性质来运行优化。

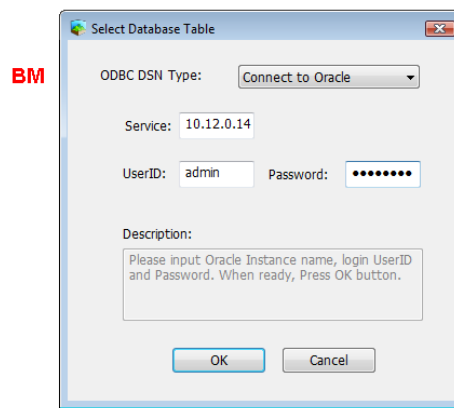


## 连接到其他数据库

ROV Risk 模型 er 通过使用 ODBC 标准能够连接到不同类型的数据.当映射到数据库时, 您可以选择数据连接方法. 单击下一步, 在新变量名中输入一个名字. 然后选择打开数据库来选择数据库类型, 同时您可以选择 ROV Risk 模型 er 提供的不同数据类型, 包括 CSV, Excel, SQL Server, Oracle, 用户 DSN, 系统 DSN, 连接串, ODBC 数据源标准.

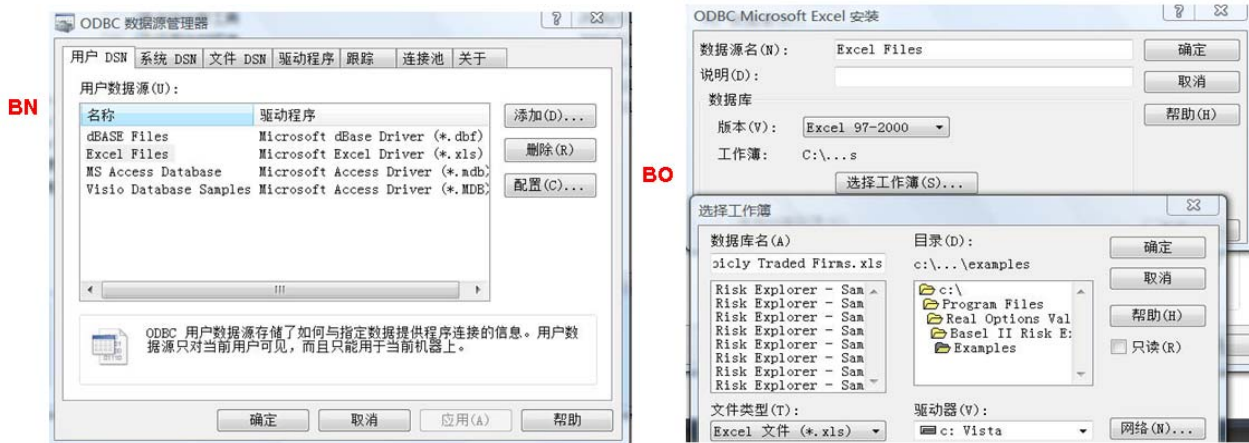
### 案例一: 连接到 Oracle

当您选择 ODBC DSN 为连接到 Oracle 时, 输入数据库服务器的本地 IP 地址和对应的用户名 ID 和密码来登录到服务器上 **[BM]**. 您可以发现选择的变量. 同时您也可以条件框中写入 SQL 语句, 直到正确的变量和数值连接到 ROV Risk 模型 er 中. 非常重要是数据库组件必须是 Oracle7.3 以上的版本.



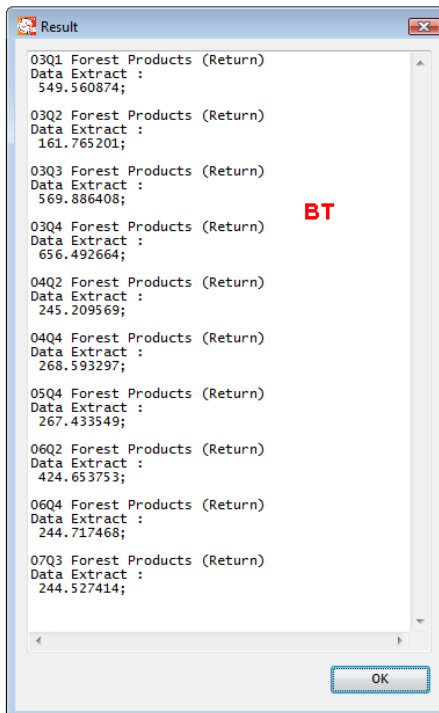
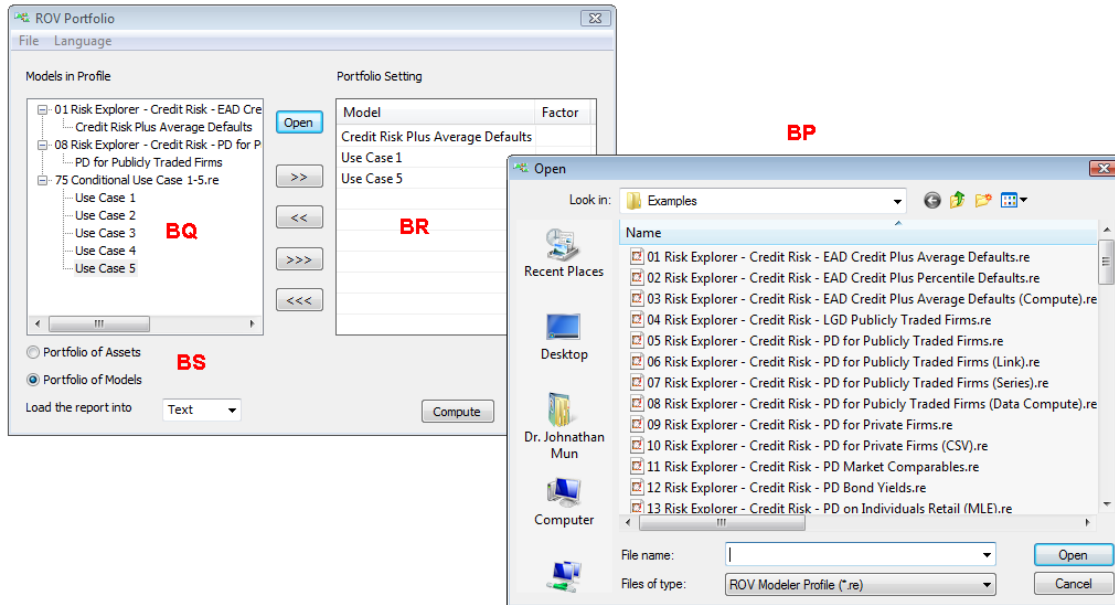
### 案例二: 连接到用户 DSN 中

在您选择 ODBC DSN 为用户 DSN 时, 您必须在第一步设置 DSN 到某一个文件中. 单击开始, 选择控制面板工具, 您可以看见数据源(ODBC) **[BN]** 选择用户 DSN 并且单击 Excel 文件, 单击配置. 在新窗口中, 单击选择工作组, 找到已有的一个 Excel 文件, 单击 OK **[BO]**. 您可以返回到 ROV Risk 模型 er 软件, 通过用户连接映射一个变量, 单击打开 DB, 用户 DSN. 选择 Excel 文件, 将会出现一个列表. 您可以连接已有数据表到选择域.



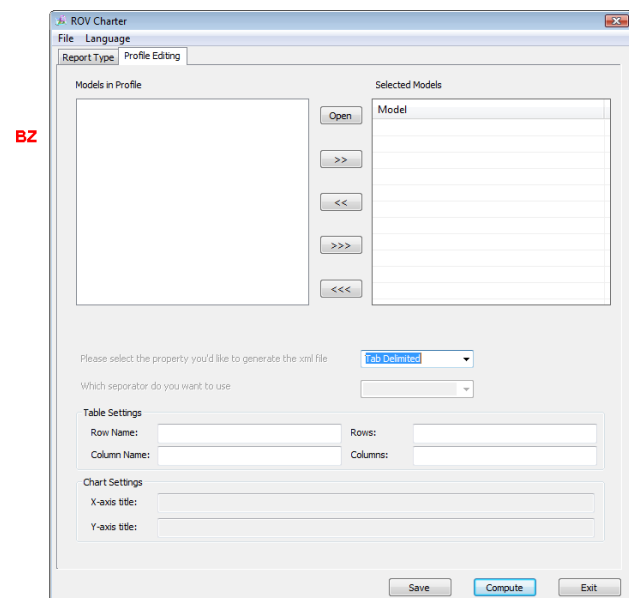
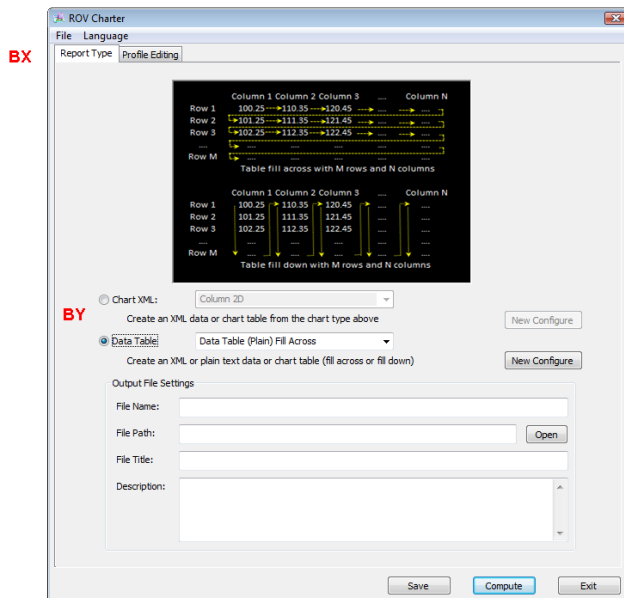
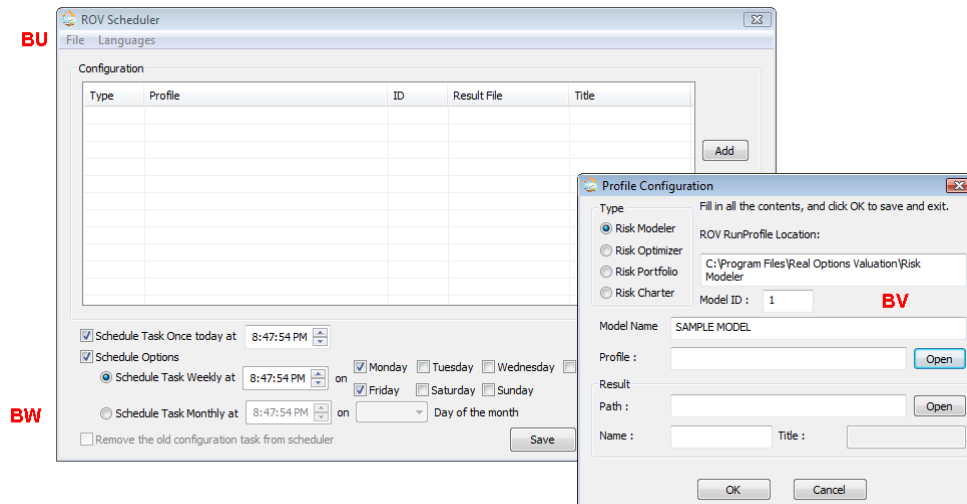
## ROV Scheduler, ROV Portfolio, ROV Charter

本说明文档开始我们列举了 ROV Portfolio, ROV Scheduler 和 ROV Charter 的使用。ROV Portfolio 用来一次从单个 profile 或者多个 profile 中运行多个模型[BP] 您可以单击 Open 来打开不同的 profile [BQ] . 您可以选择希望以单个独立变量运行模型[BR]还是在一个组合中运行相互依存的模型(资产文件夹) [BS] 当模型文件夹运行时, 结果将是一系列的数值[BT] . 这些文件夹是可以被保存的。这意味着一个单个文件夹可以包含多个模型和多个 profile.



类似的, 您可以建立 ROV Scheduler [BU]来运行多个模型, 从不同 profile 中提取出来的文件夹[BV], 就和 ROV Portfolio 一样。区别在于您可以为何时运行模型设置时间[BW] 最后,

您可以通过 ROV Charter[BX]工具来从不同的 profile 来运行不同的模型 [BZ] ,并且将结果设置为 XML 文件[BY] , 这个 XML 文件可以运行在 ROV Dashboard 中，以图表或者表格的方式在网页服务器环境中显示。



## 无用户界面运行 XML

有时候您希望在没有软件界面的情况下运行 ROV 模型 er 的 XML Profile.您可以将这种方法等同于 SDK,并且将这个 SDK 集成到您现有的系统之中。这种方法非常简单因为所有模型都与模型 Lgc.dll 文件相互连接, 适用于一个类似于 C 的界面:

```
int RunProfile (std::wstring& pszWorkDir, std::wstring& pszProfile, std::wstring& pszTitle, int nModID, int nType, std::wstring& pszFileName)
```

目前为止, 它支持三种 profile 类型:

*Risk 模型 profile, nType=0*

*Optimization profile nType=1*

*Portfolio profile nType=2*

模型 Lgc.dll 文件所要求的头文件包括三个 ROV 头文件: 模型 Lgc.h, 模型 H 和 le.h 和 结果 Set.h; 和 the related library headers, includes STL, GLib, LibXML. 以下是调用模型 Lgc.dll 及生成报告的的原代码.

```
// sample.cpp: Defines the entry point for the console application.
//Step1 adjust project setting, adding include directories 和 dependencies:
//Below is additional include directories:
//G:\RSCPP\Codes\ROV Risk 模型 er\LibExtra\libxml++-2.18.2";
//G:\RSCPP\Codes\ROV Risk 模型 er\LibExtra\libxml++-2.18.2/MSVC_Net2003/libxml++";
//G:\RSCPP\Codes\ROV Risk 模型 er\LibExtra\glibmm-2.14.2\glib";
//G:\RSCPP\Codes\ROV Risk 模型 er\LibExtra\glibmm-2.14.2\glib\glibmm";
//G:\RSCPP\Codes\ROV Risk 模型 er\LibExtra\glibmm-2.14.2/MSVC_NET2003\glibmm";
//G:\RSCPP\Codes\ROV Risk 模型 er\LibExtra\glib-2.14.3";
//G:\RSCPP\Codes\Include
//Below is additional dependencies:
//模型 Lgc.lib
//Step2 include the 模型 Lgc.h header

#include "stdafx.h"
#include <string>
#include <iostream>
#include "G://RSCPP//Codes//ROV Risk 模型 er//模型 Lgc//模型 lgc.h"

using std::wstring;

//Step3 call RunProfile method to compute profile
int callRisk 模型 Profile()
{
//Location of configuration files where contains the sepcfg.xml file
    wstring s1(L"G://RSCPP//Profiles");

//Absolute path of profile
    wstring s2(L"G://RSCPP//Profiles//Examples//01 ROV Risk 模型 er - Credit Risk - EAD Credit Plus Average Defaults.re");
```

```

//No use
wstring s3(L "");
//Location of output file
wstring s4(L"G://output.txt");
//The 模型's ID in the profile
int nMode = 1;
//Run Risk 模型 type
int nType = 0;
int i = RunProfile(s1, s2, s3, nMode, nType, s4);
//Return 0 means success, else failed
std::cout<<i<<std::endl;
return 0;
}

```

```

int callOptimizationProfile()
{
//Location of configuration files
wstring s1(L"G://RSCPP//Profiles");
//Absolute path of profile
wstring s2(L"G://RSCPP//Profiles//ROSample//ROSample//dis9.ro");
//Title of report
wstring s3(L"Example report!");
//Location of output file
wstring s4(L"G://output_ro.txt");
//No use
int nMode = 0;
//Run Optimization type
int nType = 1;
int i = RunProfile(s1,s2,s3, nMode, nType,s4);
std::cout<<i<<std::endl;
return 0;
}

```

```

int callPortfolioProfile()
{
//Location of configuration files
wstring s1(L"G://RSCPP//Profiles");
//Absolute path of profile
wstring s2(L"G://RSCPP//Profiles//ROSample//ROSample//dis9.ro");
//No use
wstring s3(L "");
//Location of output file
wstring s4(L"G://output_ro.txt");
//No use
int nMode = 0;

//Run Portfolio type
int nType = 2;

```

```
int i = RunProfile(s1,s2,s3, nMode, nType,s4);
std::cout<<i<<std::endl;
return 0;
}

int main()
{
    callRisk 模型 Profile();
    callOptimizationProfile();
}
```

## 与其他系统相互集成

ROV Risk 模型 er 同样也是一个 DLL SDK, 您可以将 SDK 集成到您自己的系统中运行. Risk 模型 er DLL 由一些模型和算法组成, XML 文件中的数据可以加载到每个模型中. 您可以手动输入数据或者使用现有数据或者连接到数据库中. 当用户打开 Risk 模型 er 时候, 算法会导入 XML 文件中的数据并返回计算结果. 您可以将结果和输出导入到一个文件和路径中. 以下是 DLL 调用的一个例子:

- Load DLL: 在 Windows 操作系统中, 您可以通过这个函数来加载 Risk 模型 er DLL:  
`LoadLibrary(_T("模型 Lgc.dll"))`
- 请保证在使用模型中输入了正确的参数. 所有的输入参数都是基于如下的 XML 文件结构. 输入和参数类型的数目在不同的模型和算法中是相互独立的. 数据值可以是单个值, 变量, 来自 Excel 或者数据库中的一系列值, 决定于用户数据的位置.

这是一个简单的存储模型参数的 XML 结构:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <risktype type="credit"/>
  <模型 name="PD for Publicly Traded Firms" ID="5" process="Probability of Default (PD)"
note="">
    <var name="MarketValueEquity" ID="1" datatype="">
      <map matype="Manual" name="MI_1">
        <data name="MI_1" uniquevalue="2" text="3000"/>
      </map>
    </var>
    <var name="MarketEquityVolatility" ID="2" datatype="">
      <map matype="Manual" name="MI_2">
        <data name="MI_2" uniquevalue="2" text="0.45"/>
      </map>
    </var>
    <var name="BookValueLiabilities 和 Debt" ID="3" datatype="">
      <map matype="Manual" name="MI_3">
        <data name="MI_3" uniquevalue="2" text="10000"/>
      </map>
    </var>
    <var name="RiskFree" ID="4" datatype="">
      <map matype="Manual" name="MI_4">
        <data name="MI_4" uniquevalue="2" text="0.05"/>
      </map>
    </var>
    <var name="GrowthRate" ID="5" datatype="">
      <map matype="Manual" name="MI_5">
        <data name="MI_5" uniquevalue="2" text="0.07"/>
      </map>
    </var>
    <var name="Maturity" ID="6" datatype="">
      <map matype="Manual" name="MI_6">
        <data name="MI_6" uniquevalue="2" text="1.00"/>
      </map>
    </var>
  </模型>
</root>
```

```

    </map>
  </var>
</模型>
<simulation runsim="1" trials="500" seed="0" runstress="0" percent="0"/>
<output_option radio="0"/>

<variable>
</variable>
</root>

```

模型 *s* 是算法而 *VAR* 是模型所需要的变量。Simulation 是模拟功能的信息。请注意 XML 文件的规则和节点是不能够改变的，因为所有的 Risk 模型 er 模型是基于 XML 文件中的数据来计算的。否则，Risk 模型 er 算法无法在没有正确数量的参数输入下运行。

- 模型返回的结构如下：

```

std::vector<结果 sSet*> *p 结果;
template<class T>
struct C 结果 Set
{
    std::vector<C 结果 Set<T>*>* child;
    int type; // double, string
    int setType; // single, vector, vector<vector>
    int identity;
    bool bChild;
    std::vector<std::vector<T>*>* thisValue;
};

```

所有的模型在从 XML 计算输入参数后，都会回到 *p* 结果指针。您可以将这些值输出到指定文件和路径中。

- 调用模型的方法：

以下代码将展示在 Windows 环境下调用 Risk 模型 er 算法的数据接口和功能。参数 *pszProfile* 是存储 XML 文件的文件名，*nModID* 是每个模型的 ID 号。

```

//define a function pointer。
typedef void* (*Compute)(wchar_t* pszProfile, int nModID);
//Cteate a function pointer case 和 point to Compute 模型。
Compute pFunc = (Compute)GetProcAddress(m_h 模型 LgcDll, "Compute 模型");
if(pFunc != NULL)
    //Call Compute 模型 Method : This will locate the needed 模型 in terms of the nModID 该方法会根据 nModID
    p 结果 = pFunc(pszProfile, uID);
//Define a function pionter
typedef void (*Free 结果)(void* p 结果);

```

//Create a function pointer case 和 point to the Free 结果 : Release the memory space used by the return value

Free 结果 pFunc = (Free 结果)GetProcAddress(m\_h 模型 LgcDll, "Free 结果");

if(pFunc != NULL)

pFunc((void\*)p 结果);

## 案例

### 1. 步骤一:

a. 新建 xml 文件 *PD.xml*, 复制上面的 xml 文件内容到该 xml 文件中并保存

### 2. 步骤二:

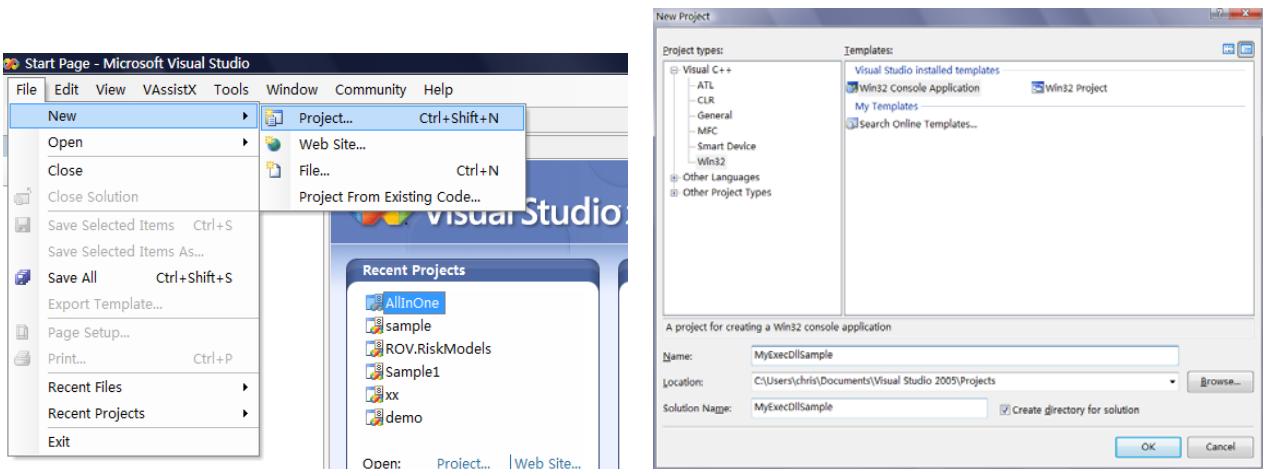
a. 打开 Visual Studio 2005, 单击 *File*, *New*, *Project* 新建一个项目

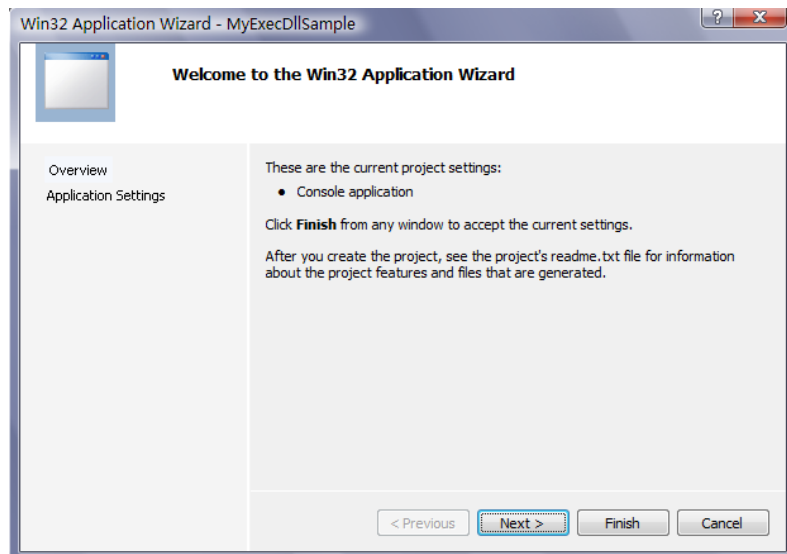
b. 选择“*Win32 Console Application*”并命名该项目为 *MyExecDllSample*

c. 单击 *OK* 创建 *Win32 Application Wizard*. 在 *Overview* 对话框中, 单击 *Next*

d. 在 *Win32 Application Wizard* 的 *Application Settings* 界面中, 选择 *Application Type* 和 *Console Application*

e. 在 *Win32 Application Wizard*'s *Application Settings* 界面中, 在 *Additional Options* 选项中去掉 *Precompiled Header*





3. 步骤三：使用以下代码替换掉 CPP 文件中的代码：

```
#include "stdafx.h"
#include "结果 Set.h"
#include <iostream>
#include <vector>
#include <string>
#include <stdio.h>
int _tmain(int argc, _TCHAR* argv[])
{
    //load the library 和 use the 模型 s to compute
    HINSTANCE hDll = LoadLibrary(_T("模型 Lgc.dll"));
    if(hDll == NULL)
        ::MessageBox(NULL, _T("Load calculation dll failed!"), _T("Error"), MB_OK);
    std::vector<RCBasell::结果 sSet*> *p 结果 = NULL;
    typedef void* (*Compute)(wchar_t* pzProfile, int nModID);
    Compute pFunc = (Compute)GetProcAddress(hDll, "Compute 模型");
    if(pFunc != NULL)
        p 结果 = (std::vector<RCBasell::结果 sSet*> *)pFunc(_T("PD.xml"), 5);

    //Save the structure which the p 结果 point:Output the 结果
    string strOut;
    typedef std::vector<double> columns;
    for(int i=0; i<p 结果->size();i++)
    {
        RCBasell::结果 sSet* pRet = p 结果->at(i);
        strOut.append(pRet->name);
        strOut.append("\n");
        char szData[32] = {0};
        std::vector<columns> *pValueTmp = (std::vector<columns>*)pRet->pValue;
```

```

        _snprintf(szData,sizeof(szData)/sizeof(szData[0])-1, "%lf", pValueTmp->at(0).at(0));
        strOut.append(szData);
        strOut.append("\r\n");
    }
    cout << strOut;

    //Release the return memory space
    typedef void (*FreeRes)(void* p 结果);
    FreeRes pFunction = (FreeRes)GetProcAddress(hDll, "Free 结果");
    if(pFunction != NULL)
        pFunction((void*)p 结果);
    FreeLibrary(hDll);

    return 0;
}
Build>Build Solution, generate the EXE file

```

#### 4. 步骤四：运行生成的应用程序

- 复制 *PD.xml* 到生成的应用程序同一个目录。开启命令行窗口并到该目录，键入应用程序名字并运行。
- 如果不能正常编译或运行，首先查看 SDK 路径是否包含在编译选项目录中；其次查看 xml 文件是否和应用程序在同一路径下。

```

C:\Windows\system32\cmd.exe
/root/model/var[6]
Attribute ID vaule: 6
maptype is: Manual
/root/model/var[6]/map
Attribute maptype vaule: Manual
Children are: text
Children are: data
1.00
1.00
MarketValueEquity: 3000 MarketEquityVolatility: 0.45 BookValueLiabilitiesDebt: 10000 RiskFree: 0.05 GrowthRate: 0.07 Maturity: 1
AssetValue: 12510.000614
VolatilityOfAsset: 0.109052
ProbabilityOfDefault: 0.004134
DistanceToDefault: 2.640904
ExpectedRecoveryRate: 0.941678
MarketValueOfDebt: 9510.000614

```

## 附录：SQL 使用案例

## SQL 条件语句使用案例

以下展示了 Risk 模型 er 中常用的清洗，筛选数据的方法.每个用户案例展示了对问题的描述，XLS 范例数据文件，Risk 模型 er RE profile 和模型名称，整个方法的一些截图.当然，想要加快学习，这些案例数据还是过于小，但是用户案例文档中同样的方法和技术应用于所有大小容量的数据。以下是一些案例的总结：

- **Variable > Value** 得到大于标杆值的数据
- **Variable > 80 AND Variable < 100** 得到一个区间数据值
- **Variable < 80 OR Variable > 100** 通过 OR 条件语句来选择数据
- **(Variable > 80 AND Variable < 90) OR (Variable > 100)** 同时使用 AND 和 OR
- **Variable IN ('aaa', 'ccc')** 匹配数据中的一些字符串
- **Variable BETWEEN 80 AND 100** 两个值中间的一些数据
- **Variable LIKE '%AN%'** 匹配%长字符串和字符，包括空格
- **Variable LIKE '\_AN'** 允许匹配单个字符(下划线)
- **Variable1 / Variable2, Variable1 \* Variable2, Variable1 + Variable2...** 运行计算
- **(Y/100 + Z/10)/ 3 > X OR (Z - Y/100) > X** 允许 OR 和计算的组合
- **X < 4 UNION SELECT X FROM [first\$] WHERE X > 10** 允许多个查询语句的连接
- **ISNUMERIC(Variable)** 允许数值数量的选择
- **1 = 2 UNION SELECT TOP 5 [first\$].X,[first\$].Y,[first\$].Z FROM [first\$]** 通过 union 和 top 功能来选择最上面的一排数据
- **NOT X IN (SELECT TOP 5 [first\$].[X] FROM [first\$])** 选择非最上一排数据
- **EXISTS (SELECT [first\$].Z FROM [first\$] WHERE Z>75)** 查看是否返回任何值，如果没有，返回一个空值
- **Variable1 IN (SELECT [second\$].[A] FROM [second\$])** 连接多个数据表
- 注意: **`Long Variable Names`**支持长变量名和数值的标记(例如 **`Country of Origin`** = **`United States`**)
- 注意: **Union** 总是以升序的方式排列第一列结果

## 用户案例 1: 通过值来选择一排数据

情景: 在大容量数据中, 我们可以使用条件语句来选择特定数据(比如大于一个标杆值).

SQL 描述: Variable > Value

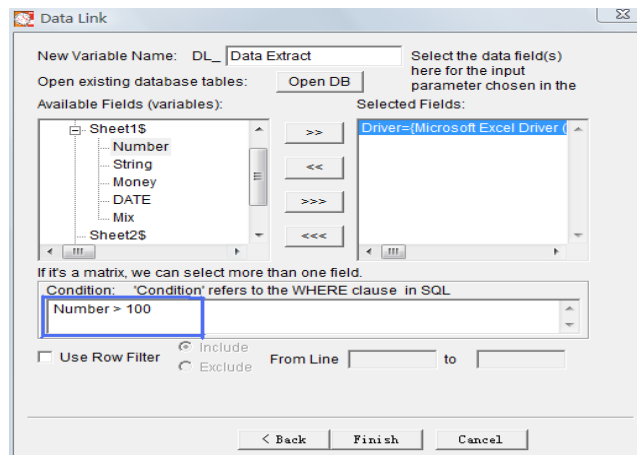
范例: Number > 100

范例 Profile 和模型: Use Case 1-5.re 和 Use Case 1 模型

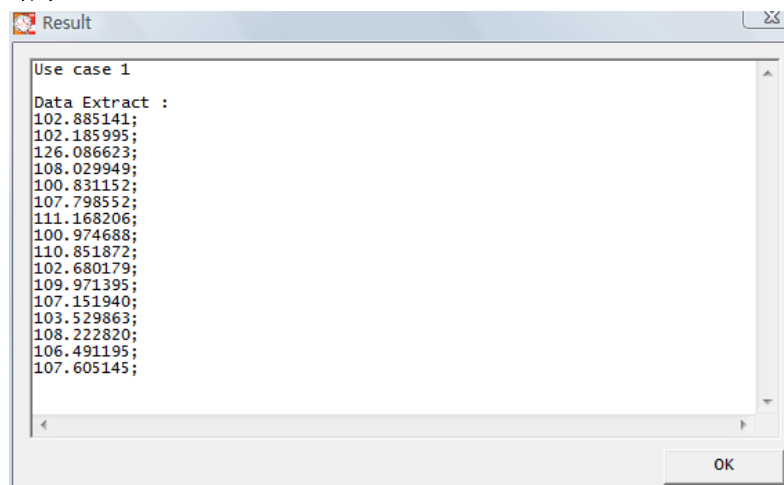
范例数据文件: Sample Data 1.xls

注意: 您可以选择 >=, <=, >, <

	A	B	C	D	E		A	B	C	D	E
1	Number	String	Money	DATE	Mix	16	110.851872	ooo	¥83.37	4/13/1900	103.05
2	102.885141	aaa	¥108.16	4/13/1900	102.88	17	102.680179	ppp	¥82.69	4/5/1900	102.87
3	84.705038	bbb	¥100.89	4/13/1900	aaa	18	69.971395	qqq	¥92.11	4/21/1900	*
4	92.160695	ccc	¥108.16	4/8/1900	122.26	19	93.250333	rrr	¥98.85	4/6/1900	89.48
5	102.185995	ddd	¥104.63	3/31/1900	108.46	20	96.570041	sss	¥104.34	3/26/1900	&&
6	91.309775	eee	¥111.91	4/28/1900	100.64	21	97.653884	ttt	¥101.73	4/4/1900	108.71
7	126.086623	fff	¥99.98	3/27/1900	99.74	22	75.886155	uuu	¥115.73	3/29/1900	108.09
8	108.029949	ggg	¥91.16	4/3/1900	110.64	23	107.151940	vvv	¥86.08	4/5/1900	95.85
9	88.869916	hhh	¥100.39	4/27/1900	83.52	24	103.529863	www	¥114.24	3/28/1900	95.75
10	95.844675	iii	¥108.17	4/4/1900	106.53	25	108.222820	xxx	¥100.62	4/20/1900	()
11	100.831152	jjj	¥115.29	4/2/1900	#!@#&#	26	106.491195	yyy	¥106.88	4/16/1900	108.62
12	107.798552	kkk	¥98.99	4/1/1900	107.64	27	80.822858	zzz	¥117.50	4/16/1900	100.64
13	111.168206	lll	¥78.03	4/19/1900	96.88	28	91.103886	abc	¥103.53	4/8/1900	95.56
14	93.873964	mmm	¥109.57	4/7/1900	101.14	29	92.807726	def	¥85.72	4/14/1900	103.20
15	100.974688	nnn	¥104.37	4/16/1900	113.02	30	107.605145	ghi	¥84.81	4/4/1900	%%%%
						31	94.677514	jkm	¥101.05	4/13/1900	FCCC



结果:



## 用户案例 2: Use of 'AND'

情景: Use '和' to connect two or more conditions together, if all conditions are "TRUE", then the data is selected.

SQL 描述: **condition AND condition AND condition AND...**

Example: **Number > 80 AND Number < 100**

范例 Profile 和模型: 用户案例 1-5.re 和 用户案例 2 模型

范例数据文件: **Sample Data 1.xls**

	A	B	C	D	E		A	B	C	D	E
1	Number	String	Money	DATE	Mix	16	110.851872	ooo	¥83.37	4/13/1900	103.05
2	102.885141	aaa	¥108.16	4/13/1900	102.88	17	102.680179	ppp	¥82.69	4/5/1900	102.87
3	84.705038	bbb	¥100.89	4/13/1900	aaa	18	69.971395	qqq	¥92.11	4/21/1900	*
4	92.160695	ccc	¥108.16	4/8/1900	122.26	19	93.250333	rrr	¥98.85	4/6/1900	89.48
5	102.185995	ddd	¥104.63	3/31/1900	108.46	20	96.570041	sss	¥104.34	3/26/1900	&&
6	91.309775	eee	¥111.91	4/28/1900	100.64	21	97.653884	ttt	¥101.73	4/4/1900	108.71
7	126.086623	fff	¥99.98	3/27/1900	99.74	22	75.886155	uuu	¥115.73	3/29/1900	108.09
8	108.029949	ggg	¥91.16	4/3/1900	110.64	23	107.151940	vvv	¥86.08	4/5/1900	95.85
9	88.869916	hhh	¥100.39	4/27/1900	83.52	24	103.529863	www	¥114.24	3/28/1900	95.75
10	95.844675	iii	¥108.17	4/4/1900	106.53	25	108.222820	xxx	¥100.62	4/20/1900	()
11	100.831152	jjj	¥115.29	4/2/1900	#0#0#	26	106.491195	yyy	¥106.88	4/16/1900	108.62
12	107.798552	kkk	¥98.99	4/1/1900	107.64	27	80.822858	zzz	¥117.50	4/16/1900	100.64
13	111.168206	lll	¥78.03	4/19/1900	96.88	28	91.103886	abc	¥103.53	4/8/1900	95.56
14	93.873964	mmm	¥109.57	4/7/1900	101.14	29	92.807726	def	¥85.72	4/14/1900	103.20
15	100.974688	nnn	¥104.37	4/16/1900	113.02	30	107.605145	ghi	¥84.81	4/4/1900	%%%%
						31	94.677514	jkm	¥101.05	4/13/1900	FCCC

New Variable Name: DL\_ Data Extract

Open existing database tables: Open DB

Available Fields (variables): Sheet1\$ Number, String, Money, DATE, Mix; Sheet2\$

Selected Fields: Drivers=(Microsoft Excel Driver...)

Condition: 'Condition' refers to the WHERE clause in SQL  
Number > 80 AND Number < 100

Use Row Filter  Include  Exclude From Line to

< Back Finish Cancel

结果:

Use case 2

Data Extract :

84.705038;  
92.160695;  
91.309775;  
88.869916;  
95.844675;  
93.873964;  
93.250333;  
96.570041;  
97.653884;  
85.886155;  
80.822858;  
91.103886;  
92.807726;  
94.677514;

OK

### 用户案例 3: Use of 'OR'

情景: 使用'OR'来连接两个或者更多的条件语句, 一旦任何条件语句是"TRUE", 数据将被选中, 即使其他条件语句是"FAUSE".

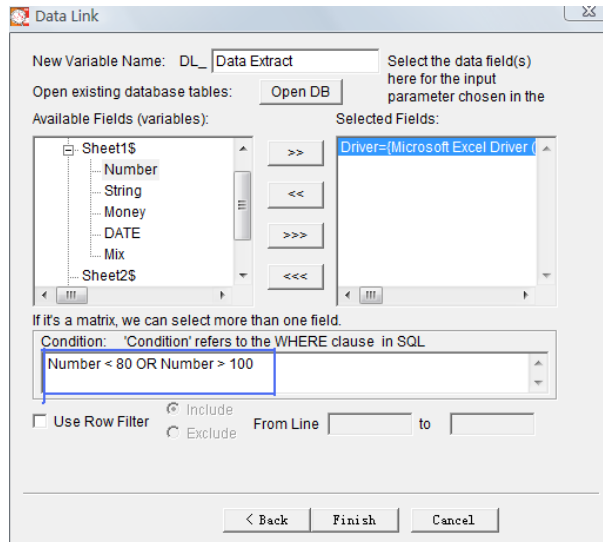
SQL 描述: **condition OR condition OR condition OR...**

Example: **Number < 80 OR Number > 100**

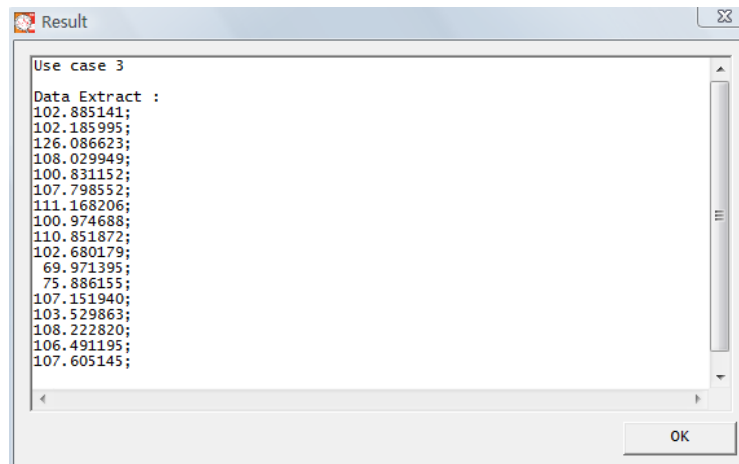
范例 Profile 和模型: 用户案例 1-5.re 和 用户案例 3 模型

范例数据文件: **Sample Data 1.xls**

	A	B	C	D	E		A	B	C	D	E
1	Number	String	Money	DATE	Mix	16	110.851872	ooo	¥83.37	4/13/1900	103.05
2	102.885141	aaa	¥108.16	4/13/1900	102.88	17	102.680179	ppp	¥82.69	4/5/1900	102.87
3	84.705038	bbb	¥100.89	4/13/1900	aaa	18	69.971395	qqq	¥92.11	4/21/1900	*
4	92.160695	ccc	¥108.16	4/8/1900	122.26	19	93.250333	rrr	¥98.85	4/6/1900	89.48
5	102.185995	ddd	¥104.63	3/31/1900	108.46	20	96.570041	sss	¥104.34	3/26/1900	&&
6	91.309775	eee	¥111.91	4/28/1900	100.64	21	97.653884	ttt	¥101.73	4/4/1900	108.71
7	126.086623	fff	¥99.98	3/27/1900	99.74	22	75.886155	uuu	¥115.73	3/29/1900	108.09
8	108.029949	ggg	¥91.16	4/3/1900	110.64	23	107.151940	vvv	¥86.08	4/5/1900	95.85
9	88.869916	hhh	¥100.39	4/27/1900	83.52	24	103.529863	www	¥114.24	3/28/1900	95.75
10	95.844675	iii	¥108.17	4/4/1900	106.53	25	108.222820	xxx	¥100.62	4/20/1900	()
11	100.831152	jjj	¥115.29	4/2/1900	¥\$@#0#	26	106.491195	yyy	¥106.88	4/16/1900	108.62
12	107.798552	kkk	¥98.99	4/1/1900	107.64	27	80.822858	zzz	¥117.50	4/16/1900	100.64
13	111.168206	lll	¥78.03	4/19/1900	96.88	28	91.103886	abc	¥103.53	4/8/1900	95.56
14	93.873964	mmm	¥109.57	4/7/1900	101.14	29	92.807726	def	¥85.72	4/14/1900	103.20
15	100.974688	nnn	¥104.37	4/16/1900	113.02	30	107.605145	ghi	¥84.81	4/4/1900	%%%%
						31	94.677514	jkm	¥101.05	4/13/1900	FCCC



结果:



#### 用户案例 4: 'AND' 和 'OR' 共同使用

情景: 我们可以同时使用'和' 和'OR'.

SQL 描述: **condition AND condition OR condition...**

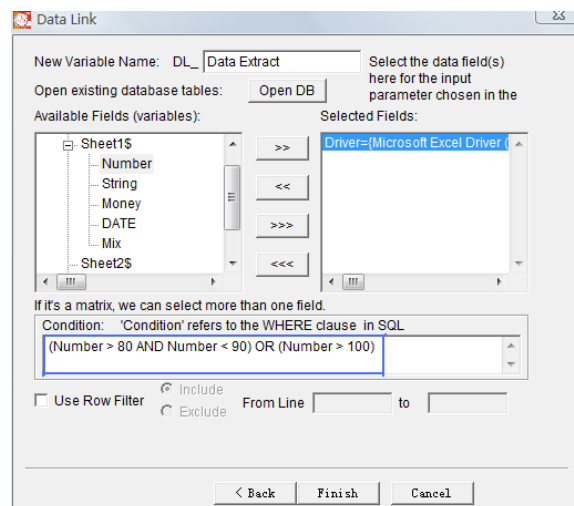
范例: **(Number > 80 AND Number < 90) OR (Number > 100)**

范例 Profile 和模型: 用户案例 1-5.re 和 用户案例 4 模型

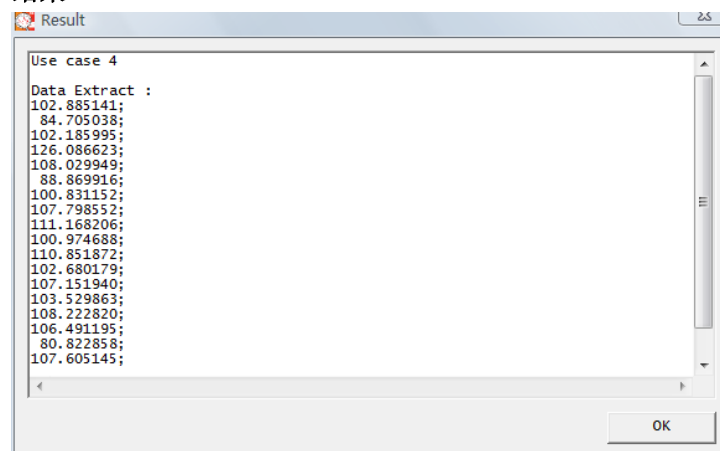
范例数据文件: **Sample Data 1.xls**

注意: 您可以使用插入语来集成命令行

	A	B	C	D	E		A	B	C	D	E
1	Number	String	Money	DATE	Mix	16	110.851872	ooo	¥83.37	4/13/1900	103.05
2	102.885141	aaa	¥108.16	4/13/1900	102.88	17	102.680179	ppp	¥82.69	4/5/1900	102.87
3	84.705038	bbb	¥100.89	4/13/1900	aaa	18	69.971395	qqq	¥92.11	4/21/1900	*
4	92.160695	ccc	¥108.16	4/8/1900	122.26	19	93.250333	rrr	¥98.85	4/6/1900	89.48
5	102.185995	ddd	¥104.63	3/31/1900	108.46	20	96.570041	sss	¥104.34	3/26/1900	&&
6	91.309775	eee	¥111.91	4/28/1900	100.64	21	97.653884	ttt	¥101.73	4/4/1900	108.71
7	126.086623	fff	¥99.98	3/27/1900	99.74	22	75.886155	uuu	¥115.73	3/29/1900	108.09
8	108.029949	ggg	¥91.16	4/3/1900	110.64	23	107.151940	vvv	¥86.08	4/5/1900	95.85
9	88.869916	hhh	¥100.39	4/27/1900	83.52	24	103.529863	www	¥114.24	3/28/1900	95.75
10	95.844675	iii	¥108.17	4/4/1900	106.53	25	108.222820	xxx	¥100.62	4/20/1900	()
11	100.831152	jjj	¥115.29	4/2/1900	#0#0#	26	106.491195	yyy	¥106.88	4/16/1900	108.62
12	107.798552	kkk	¥98.99	4/1/1900	107.64	27	80.822858	zzz	¥117.50	4/16/1900	100.64
13	111.168206	lll	¥78.03	4/19/1900	96.88	28	91.103886	abc	¥103.53	4/8/1900	95.56
14	93.873964	mmm	¥109.57	4/7/1900	101.14	29	92.807726	def	¥85.72	4/14/1900	103.20
15	100.974688	nnn	¥104.37	4/16/1900	113.02	30	107.605145	ghi	¥84.81	4/4/1900	%%%%
						31	94.677514	jkm	¥101.05	4/13/1900	FCCC



结果:



## 用户案例 5: 'IN'的使用

情景: 通过使用 'IN' 命令来指定单个数据或者多个数据.

SQL 描述: **Variable IN ('value1', 'value2'...)**

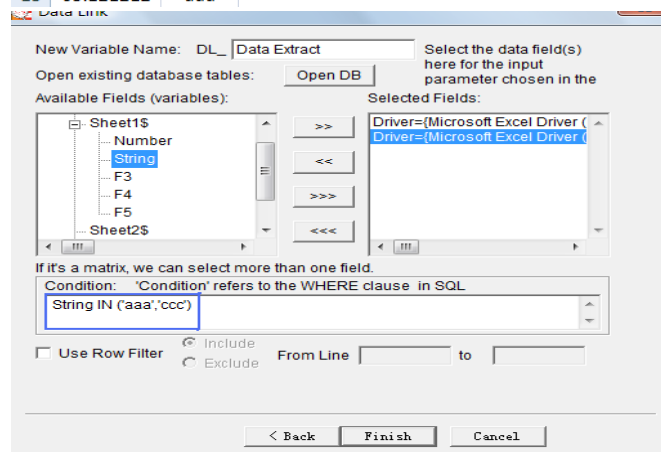
Example: **String IN ('aaa', 'ccc')**

范例 Profile 和模型: 用户案例 1-5.re 和 用户案例 5 模型

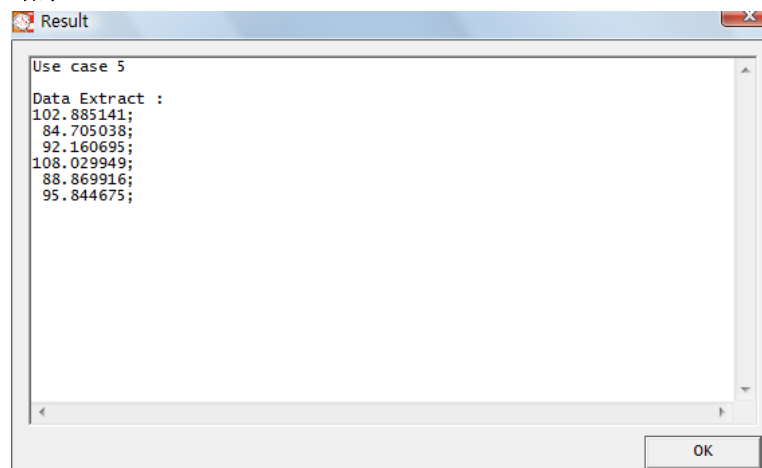
范例数据文件: **Sample Data 2.xls**

注意: 如果筛选的数值是字符串, 使用'quotes'

	A	B
1	<b>Number</b>	<b>String</b>
2	102.885141	aaa
3	84.705038	aaa
4	92.160695	aaa
5	102.185995	bbb
6	91.309775	bbb
7	126.086623	bbb
8	108.029949	ccc
9	88.869916	ccc
10	95.844675	ccc
11	100.831152	ddd
12	70.121340	ddd
13	65.121212	ddd



结果:



## 用户案例 6: 'BETWEEN'的使用

情景: 使用 'BETWEEN' 来选择一个区域中的数据.

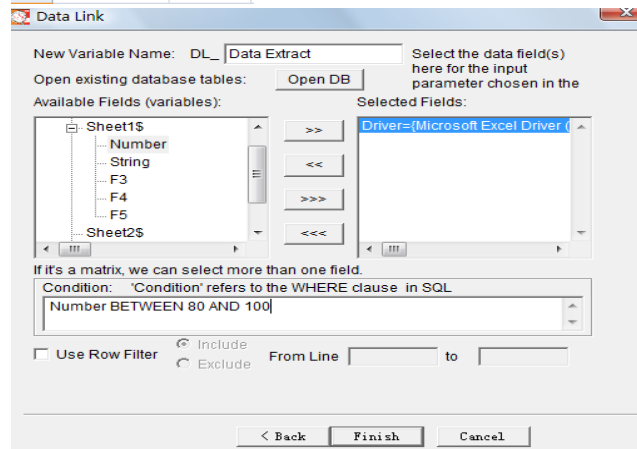
SQL 描述: Variable BETWEEN 'value1' AND 'value2'

范例: Number BETWEEN 80 AND 100

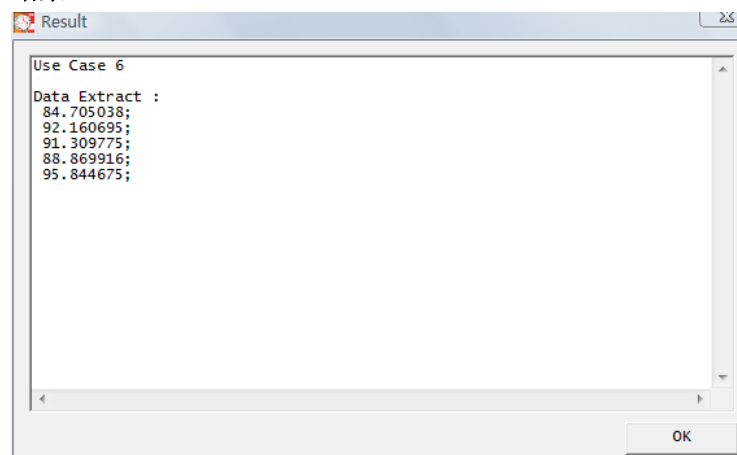
范例 Profile 和模型: 用户案例 5-10.re 和 用户案例 6 模型

范例数据文件: Sample Data 2.xls

	A	B
1	Number	String
2	102.885141	aaa
3	84.705038	aaa
4	92.160695	aaa
5	102.185995	bbb
6	91.309775	bbb
7	126.086623	bbb
8	108.029949	ccc
9	88.869916	ccc
10	95.844675	ccc
11	100.831152	ddd
12	70.121340	ddd
13	65.121212	ddd



结果:



## 用户案例 7: 'LIKE'的使用

情景: 'LIKE'条件语句使得您在 Where 语句中进行类型的匹配.

### SQL 描述:

您可以选择的类型:

% 允许您匹配任何长度的任何字符(包括零长度)

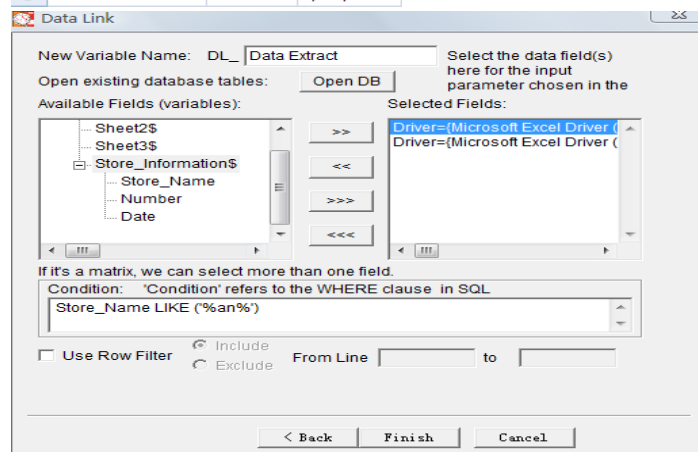
\_ 允许您匹配单个字符

范例: `store_name LIKE '%AN%'`

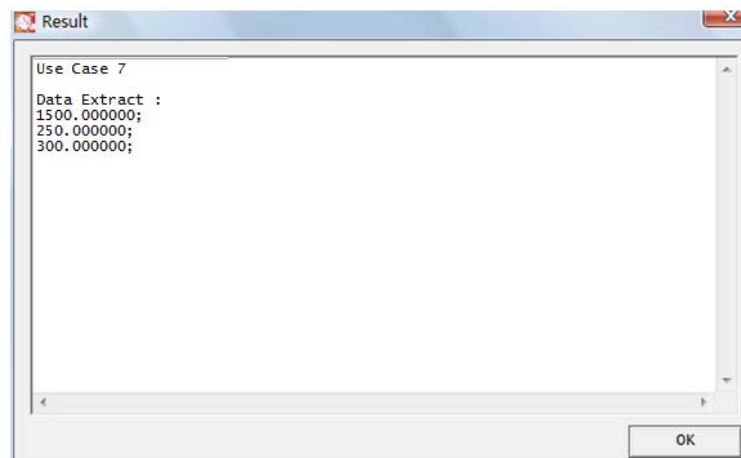
范例 Profile 和模型: 用户案例 5-10.re 和 用户案例 7 模型

范例数据文件: Sample Data 3.xls

	A	B	C
1	<b>Store_Name</b>	<b>Number</b>	<b>Date</b>
2	Los Angeles	1500.00	8/1/2008
3	San Diego	250.00	5/1/2008
4	San Francisco	300.00	2008/6/31
5	Boston	700.00	4/23/2008



结果:



## 用户案例 8: Simple Math Functions

情景: 应用到变量的基本数学公式.

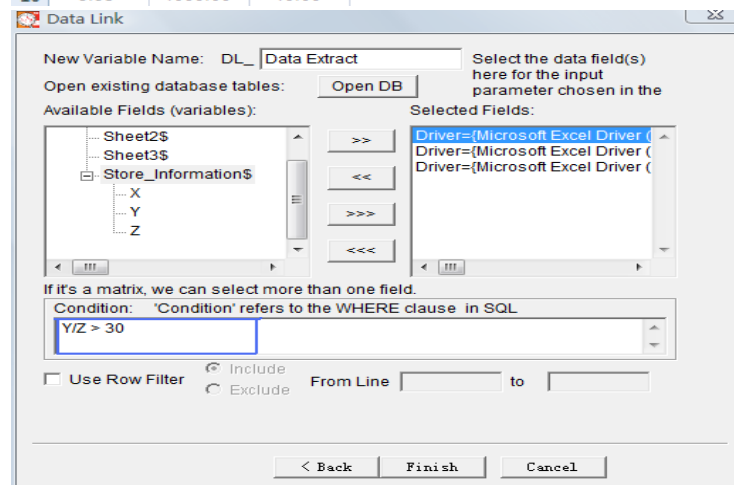
SQL 描述:  $Variable1 / Variable2$ ,  $Variable1 * Variable2$ ,  $Variable1 + Variable2 \dots$

范例:  $Y/Z > 30$

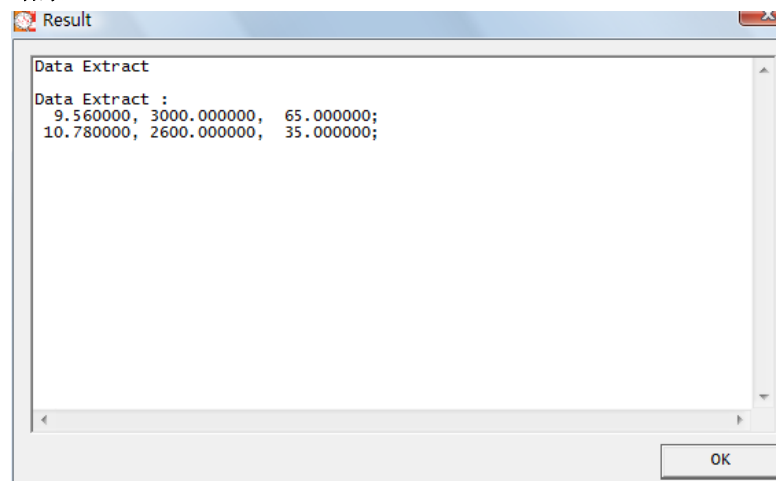
范例 Profile 和模型: 用户案例 5-10.re 和 用户案例 8 模型

范例数据文件: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



结果:



## 用户案例 9: 数学公式的多种使用

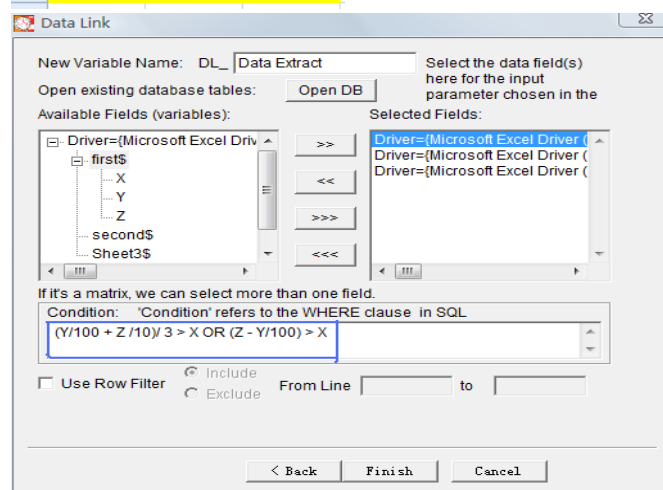
情景: 数学公式可以非常复杂.

Example:  $(Y/100 + Z/10)/3 > X$  OR  $(Z - Y/100) > X$

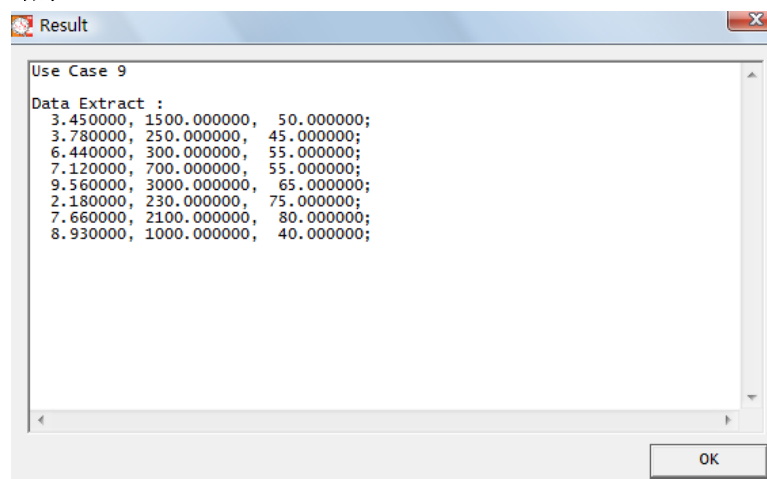
范例 Profile 和模型: 用户案例 5-10.re 和 用户案例 9 模型

范例数据文件: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



结果:



## 用户案例 10: 'Union' 的使用来连接命令行

**情景:** Union 是连接两个查询语句非常重要的命令. 当创建复杂的命令行时, 将整个命令行分割开并使用'Union'

**SQL 描述:** `CONDITION1 UNION SELECT COLUMN FROM TABLENAME WHERE CONDITION2`

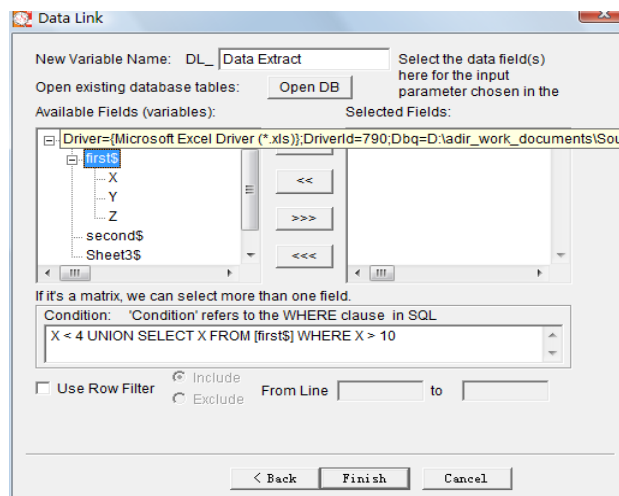
**Example:** `X < 4 UNION SELECT X FROM [first$] WHERE X > 10`

**范例 Profile 和模型:** 用户案例 5-10.re 和 用户案例 10 模型

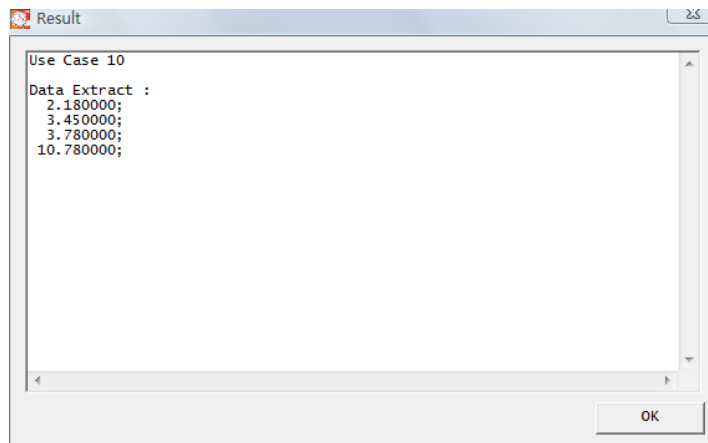
**范例数据文件:** Sample Data 4.xls

**Note:** 使用 Union 某些时候可以排序数据

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



**结果:**



## 用户案例 11: Filtering Different Value Types

**情景:** 如果数据同时拥有数字，字符串和其他数据类型，我们可以应用 'ISNUMERIC' 命令行来筛选数字型数据。

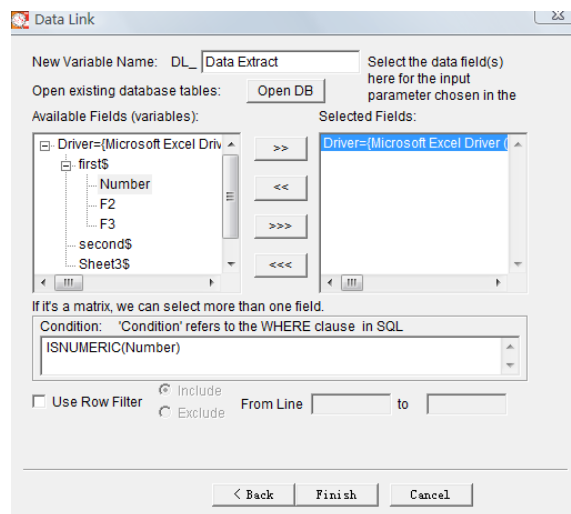
**SQL 描述:** ISNUMERIC(Variable)

**范例:** ISNUMERIC(Number)

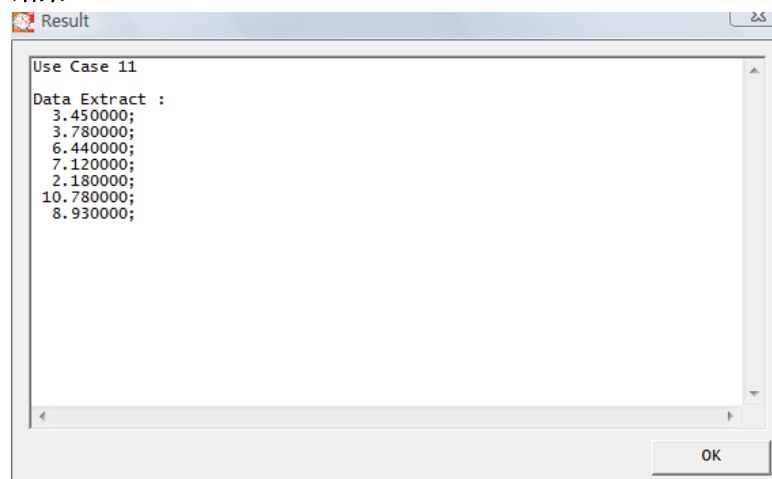
**范例 Profile 和模型:** 用户案例 11-15.re 和 用户案例 11 模型

**范例数据文件:** Sample Data 5.xls

	A
1	<b>Number</b>
2	3.45
3	3.78
4	6.44
5	7.12
6	AaA
7	2.18
8	BBB
9	10.78
10	8.93



**结果:**



## 用户案例 12: 选择最上面的 N 排数据

情景: 同时使用‘UNION’和‘TOP’来选择表格最上面的 N 排数据.

SQL 描述: **TOP N \* FROM TABLE\_NAME**

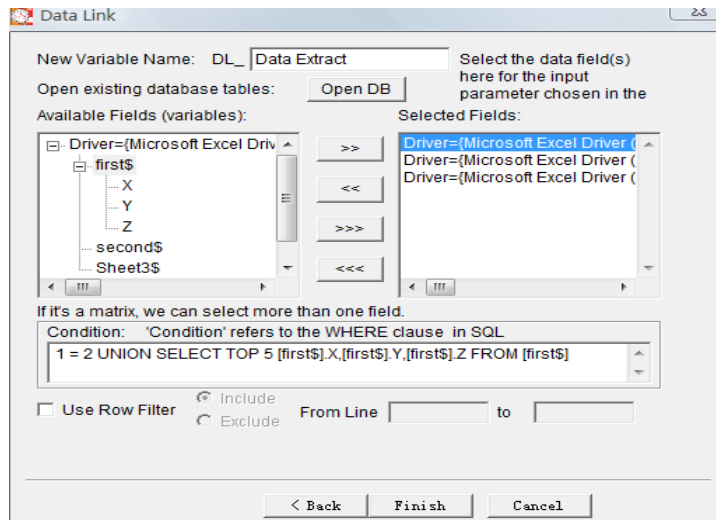
范例: **1 = 2 UNION SELECT TOP 5 [first\$].X,[first\$].Y,[first\$].Z FROM [first\$]**

注意: 第二个 **SELECT** 命令筛选的行必须和列表中的一致.‘1=2’意味着永远 **FALSE** 所以让第一个选择条件语句没有结果.

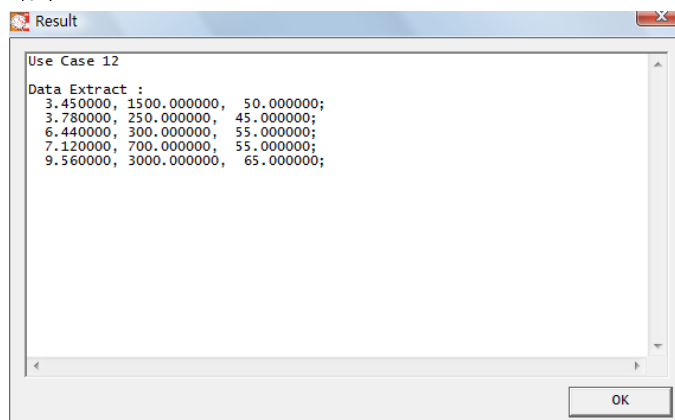
范例 Profile 和模型: 用户案例 11-15.re 和 用户案例 12 模型

范例数据文件: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



结果:



### 用户案例 13: 'NOT IN'的使用

情景: 'NOT IN' 用来选出从下一个条件命令的值.如果 column 数值是唯一的, 可以用来从一系列排中得到数据.

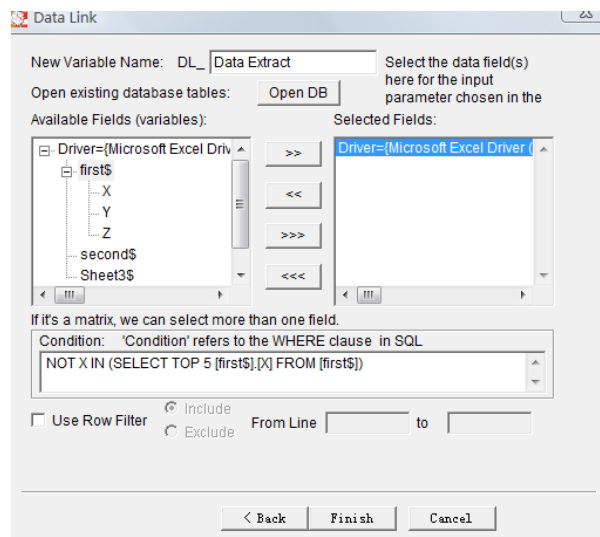
SQL 描述: **NOT Variable IN (command)**

范例: **NOT X IN (SELECT TOP 5 [first\$].[X] FROM [first\$])**

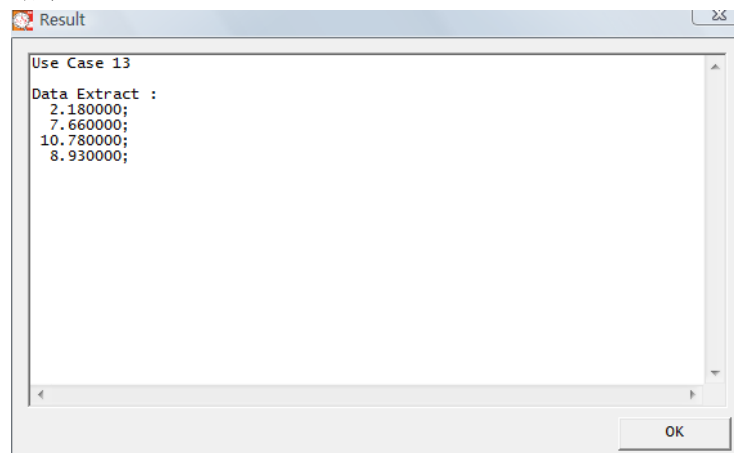
范例 Profile 和模型: 用户案例 11-15.re 和 用户案例 13 模型

范例数据文件: **Sample Data 4.xls**

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



结果:



## 用户案例 14: 'EXISTS'的使用

**情景:** 'EXISTS'测试内部查询语句是否返回任何排数据。如果返回,外部查询继续进行。如果不返回数据,外部查询不再进行,整个 SQL 语句不返回任何变量。

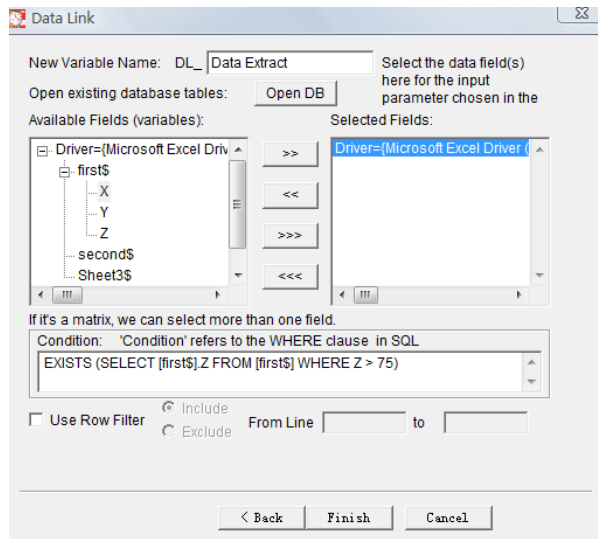
**SQL 描述:** `EXISTS (SELECT * FROM "table_name2" WHERE [Condition])`

**Example:** `EXISTS (SELECT [first$].Z FROM [first$] WHERE Z>75)`

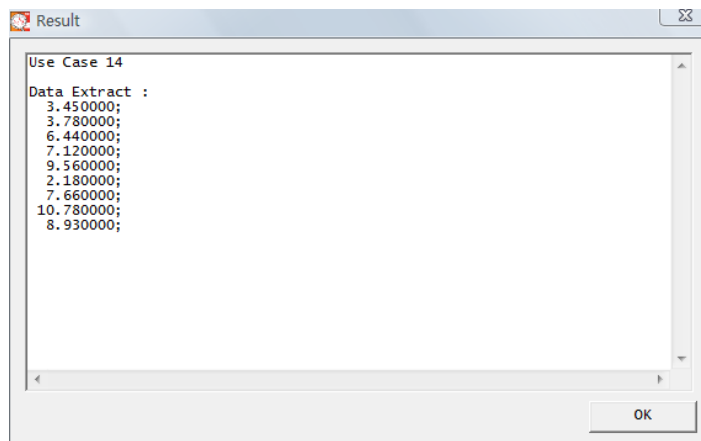
**范例 Profile 和模型:** 用户案例 11-15.re 和 用户案例 14 模型

**范例数据文件:** Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



**结果:**



## 用户案例 15: 多个表格的使用

情景: 使用'SELECT'命令行来连接多个表格.

SQL 描述: `Variable1 IN (SELECT Variable2 FROM Table_Name2 WHERE Condition2)`

Example: `X IN (SELECT [second$].[A] FROM [second$])`

范例 Profile 和模型: 用户案例 11-15.re 和 用户案例 15 模型

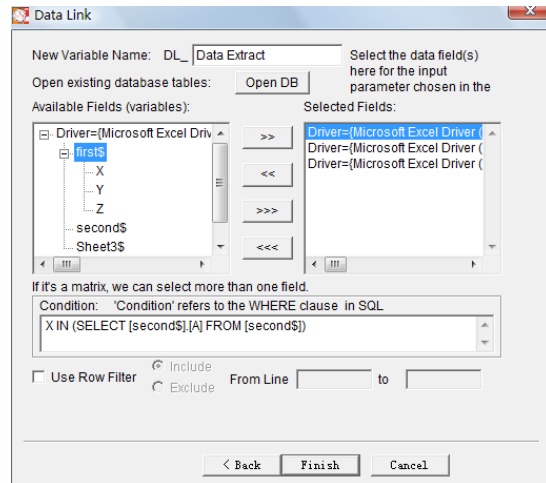
范例数据文件: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00

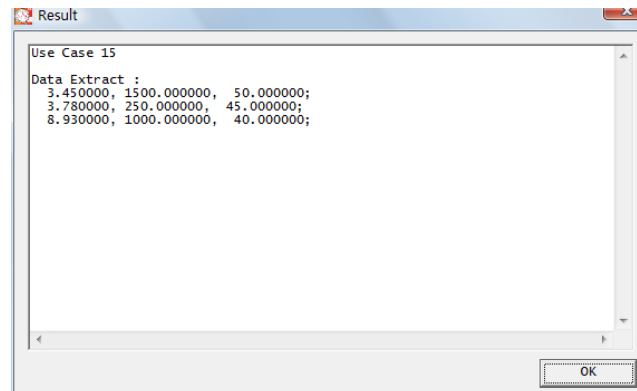
	A
1	A
2	3.45
3	3.78
4	8.93
5	6.66

第一个表格

第二个表格



结果:



## 用户案例 16: 简单的范例

情景: 选择通过所有考试的学生.

范例 Profile 和模型: 用户案例 16-20.re 和 用户案例 16

范例数据文件: Sample Data 6.xls

	A	B	C	D	E	F	G
1	Students No	Name	Age	Math	English	Biology	Geography
2	1	John	16	95	66	83	76
3	2	Tom	15	67	78	55	89
4	3	Jerry	16	93	67	92	87
5	4	Bob	17	88	88	97	92
6	5	Alexandra	16	77	98	89	68
7	6	William	18	78	100	100	70
8	7	Lily	15	96	79	87	89
9	8	Rose	16	91	84	79	90
10	9	Jack	14	99	57	92	93
11	10	Vivi	18	94	77	86	96
12	11	Vicky	15	87	65	95	75
13	12	Babala	15	99	97	95	96
14	13	Chris	17	76	57	87	98
15	14	Amanda	16	56	78	95	90
16	15	Alice	16	89	77	98	100
17	16	Amy	15	83	67	66	91
18	17	Annie	17	96	87	92	91
19	18	Cindy	16	78	89	92	85
20	19	Cora	17	67	82	83	89
21	20	Ella	18	67	65	86	56

Data Link

New Variable Name: DL\_ Data Extract

Select the data field(s) here for the input parameter chosen in the

Open existing database tables: Open DB

Available Fields (variables):

- Name
- Age
- Math
- English
- Biology
- Geography
- F8

Selected Fields:

- Driver=(Microsoft Excel Driver (
- Driver=(Microsoft Excel Driver (
- Driver=(Microsoft Excel Driver (
- Driver=(Microsoft Excel Driver (
- Driver=(Microsoft Excel Driver (

If it's a matrix, we can select more than one field.

Condition: 'Condition' refers to the WHERE clause in SQL

Math > 60 AND English > 60 AND Biology > 60 AND Geography > 60

Use Row Filter  Include  Exclude From Line  to

< Back Finish Cancel

结果:

Result

Use Case 16

Data Extract :

```
1.000000, 95.000000, 66.000000, 83.000000, 76.000000;
3.000000, 93.069592, 67.000000, 92.029799, 87.485345;
4.000000, 88.337018, 88.000000, 96.537220, 92.185350;
5.000000, 77.000000, 98.000000, 88.936537, 68.000000;
6.000000, 78.000000, 99.742818, 99.599507, 70.188680;
7.000000, 96.000000, 79.000000, 87.000000, 89.425400;
8.000000, 90.552683, 83.519584, 78.541530, 89.604188;
10.000000, 94.416071, 77.000000, 85.791101, 96.048082;
11.000000, 86.913575, 65.000000, 95.215249, 75.000000;
12.000000, 98.984351, 96.877771, 94.525668, 96.000000;
15.000000, 89.129202, 77.000000, 98.451890, 99.578803;
16.000000, 83.000000, 67.000000, 66.000000, 91.170687;
17.000000, 96.018234, 87.000000, 91.958423, 91.370947;
18.000000, 78.000000, 89.480930, 92.197027, 85.000000;
19.000000, 67.000000, 82.456611, 83.170222, 89.131081;
```

OK

## 用户案例 17: 使用 AND 的范例

情景: 选择姓名以'A'或者'J'开头, 年龄大于 16 岁的学生

范例 Profile 和模型: 用户案例 16-20.re 和 用户案例 17 模型

范例数据文件: Sample Data 6.xls

	A	B	C	D	E	F	G
1	Students No	Name	Age	Math	English	Biology	Geography
2	1	John	16	95	66	83	76
3	2	Tom	15	67	78	55	89
4	3	Jerry	16	93	67	92	87
5	4	Bob	17	88	88	97	92
6	5	Alexandra	16	77	98	89	68
7	6	William	18	78	100	100	70
8	7	Lily	15	96	79	87	89
9	8	Rose	16	91	84	79	90
10	9	Jack	14	99	57	92	93
11	10	Vivi	18	94	77	86	96
12	11	Vicky	15	87	65	95	75
13	12	Babala	15	99	97	95	96
14	13	Chris	17	76	57	87	98
15	14	Amanda	16	56	78	95	90
16	15	Alice	16	89	77	98	100
17	16	Amy	15	83	67	66	91
18	17	Annie	17	96	87	92	91
19	18	Cindy	16	78	89	92	85
20	19	Cora	17	67	82	83	89
21	20	Ella	18	67	65	86	56

Data Link

New Variable Name: DL\_ Data Extract

Open existing database tables: Open DB

Select the data field(s) here for the input parameter chosen in the

Available Fields (variables):

- first\$
- Students No
- Name
- Age
- Math
- English
- Biology

Selected Fields:

- Driver={Microsoft Excel Driver (
- Driver={Microsoft Excel Driver (
- Driver={Microsoft Excel Driver (

If it's a matrix, we can select more than one field.

Condition: 'Condition' refers to the WHERE clause in SQL

(Name LIKE 'J%' OR Name LIKE 'A%') AND (Age > 15)

Use Row Filter  Include  Exclude From Line  to

< Back Finish Cancel

结果:

Result

Use Case 17

Data Extract :

```
1.000000, 16.000000, 66.000000;
3.000000, 16.000000, 67.000000;
5.000000, 16.000000, 98.000000;
14.000000, 16.000000, 78.000000;
15.000000, 16.000000, 77.000000;
17.000000, 17.000000, 87.000000;
```

OK

## 用户案例 18: 使用 Union 和排序的案例

情景: 选择地理成绩最高的 5 个分数.

范例 Profile 和模型: 用户案例 16-20.re 和 用户案例 18 模型

范例数据文件: Sample Data 6.xls

	A	B	C	D	E	F	G
1	Students No	Name	Age	Math	English	Biology	Geography
2	1	John	16	95	66	83	76.000000
3	2	Tom	15	67	78	55	88.601113
4	3	Jerry	16	93	67	92	87.485345
5	4	Bob	17	88	88	97	92.185350
6	5	Alexandra	16	77	98	89	68.000000
7	6	William	18	78	100	100	70.188680
8	7	Lily	15	96	79	87	89.425400
9	8	Rose	16	91	84	79	89.604188
10	9	Jack	14	99	57	92	92.732209
11	10	Vivi	18	94	77	86	96.048082
12	11	Vicky	15	87	65	95	75.000000
13	12	Babala	15	99	97	95	96.000000
14	13	Chris	17	76	57	87	98.260693
15	14	Amanda	16	56	78	95	89.711824
16	15	Alice	16	89	77	98	99.578803
17	16	Amy	15	83	67	66	91.170687
18	17	Annie	17	96	87	92	91.370947
19	18	Cindy	16	78	89	92	85.000000
20	19	Cora	17	67	82	83	89.131081
21	20	Ella	18	67	65	86	56.000000

Data Link

New Variable Name: DL\_ Data Extract

Open existing database tables: Open DB

Available Fields (variables): first\$, Students No, Name, Age, Math, English, Biology

Selected Fields: Driver={Microsoft Excel Driver (

If it's a matrix, we can select more than one field.

Condition: 'Condition' refers to the WHERE clause in SQL

```
1 = 0 UNION (SELECT TOP 5 [Students No], Geography FROM [first$] ORDER BY Geography DESC)
```

Use Row Filter  Include  Exclude From Line to

< Back Finish Cancel

结果:

Result

Use Case 18

Data Extract :

```
9.000000, 92.732209;
10.000000, 96.048082;
12.000000, 96.000000;
13.000000, 98.260693;
15.000000, 99.578803;
```

OK

### 用户案例 19: 使用数学的案例

情景: 选择名字包含 'A', 平均成绩高于 85 分的学生.

范例 Profile 和模型: 用户案例 16-20.re 和 用户案例 19 模型

范例数据文件: Sample Data 6.xls

	A	B	C	D	E	F	G	H
1	Students No	Name	Age	Math	English	Biology	Geography	Average
2	1	John	16	95	66	83	76	80.00
3	2	Tom	15	67	78	55	89	72.15
4	3	Jerry	16	93	67	92	87	84.90
5	4	Bob	17	88	88	97	92	91.26
6	5	Alexandra	16	77	98	89	68	82.98
7	6	William	18	78	100	100	70	86.88
8	7	Lily	15	96	79	87	89	87.86
9	8	Rose	16	91	84	79	90	85.55
10	9	Jack	14	99	57	92	93	85.13
11	10	Vivi	18	94	77	86	96	88.31
12	11	Vicky	15	87	65	95	75	80.53
13	12	Babala	15	99	97	95	96	96.60
14	13	Chris	17	76	57	87	98	79.57
15	14	Amanda	16	56	78	95	90	79.79
16	15	Alice	16	89	77	98	100	91.04
17	16	Amy	15	83	67	66	91	76.79
18	17	Annie	17	96	87	92	91	91.59
19	18	Cindy	16	78	89	92	85	86.17
20	19	Cora	17	67	82	83	89	80.44
21	20	Ella	18	67	65	86	56	68.50

Data Link

New Variable Name: DL\_ Data Extract

Select the data field(s) here for the input parameter chosen in the

Open existing database tables: Open DB

Available Fields (variables):

- first\$
- Students No
- Name
- Age
- Math
- English
- Biology

Selected Fields:

Driver=(Microsoft Excel Driver (

If it's a matrix, we can select more than one field.

Condition: 'Condition' refers to the WHERE clause in SQL

Name LIKE '%a%' AND (Math + English + Biology + Geography)/4 > 85

Use Row Filter  Include  Exclude From Line \_\_\_\_\_ to \_\_\_\_\_

< Back Finish Cancel

结果:

Result

Use Case 19

Data Extract :

```
6.000000;  
9.000000;  
12.000000;  
15.000000;  
17.000000;
```

OK

## 用户案例 20: 重复使用 AND 和 OR 的, 数学的案例

情景: 当学生平均年龄大于 16 岁, 选择平均成绩在 85 到 95; 或者学生平均年龄小于 16 岁, 平均成绩高于 80.

范例 Profile 和模型: 用户案例 16-20.re 和 用户案例 20 模型

范例数据文件: Sample Data 6.xls

	A	B	C	D	E	F	G	H
1	Students No	Name	Age	Math	English	Biology	Geography	Average
2	1	John	16	95	66	83	76	80.00
3	2	Tom	15	67	78	55	89	72.15
4	3	Jerry	16	93	67	92	87	84.90
5	4	Bob	17	88	88	97	92	91.26
6	5	Alexandra	16	77	98	89	68	82.98
7	6	William	18	78	100	100	70	86.88
8	7	Lily	15	96	79	87	89	87.86
9	8	Rose	16	91	84	79	90	85.55
10	9	Jack	14	99	57	92	93	85.13
11	10	Vivi	18	94	77	86	96	88.31
12	11	Vicky	15	87	65	95	75	80.53
13	12	Babala	15	99	97	95	96	96.60
14	13	Chris	17	76	57	87	98	79.57
15	14	Amanda	16	56	78	95	90	79.79
16	15	Alice	16	89	77	98	100	91.04
17	16	Amy	15	83	67	66	91	76.79
18	17	Annie	17	96	87	92	91	91.59
19	18	Cindy	16	78	89	92	85	86.17
20	19	Cora	17	67	82	83	89	80.44
21	20	Ella	18	67	65	86	56	68.50

(Age >= 16 和 ((Math + English + Biology + Geography)/4 BETWEEN 85 和 95)) OR (Age < 16 和 ((Math + English + Biology + Geography)/4 > 80))

Data Link

New Variable Name: DL\_Data Extract

Select the data field(s) here for the input parameter chosen in the

Open existing database tables: Open DB

Available Fields (variables):

- Students No
- Name
- Age
- Math
- English
- Biology

Selected Fields:

- Drivers:(Microsoft Excel Driver...

If it's a matrix, we can select more than one field.

Condition: 'Condition' refers to the WHERE clause in SQL

(Age >= 16 AND ((Math + English + Biology + Geography)/4 BETWEEN 85 AND 95)) OR (Age < 16 AND ((Math + English + Biology + Geography)/4 > 80)

Use Row Filter  Include  Exclude From Line to

< Back Finish Cancel

结果:

Result

Use Case 20

Data Extract :

4.000000;  
6.000000;  
7.000000;  
8.000000;  
9.000000;  
10.000000;  
11.000000;  
12.000000;  
15.000000;  
17.000000;  
18.000000;

OK

## 用户案例 21: 'UNION ALL'

**情景:** 'UNION ALL' 命令的目的是结合两个查询语句.'UNION ALL' 和 'UNION'的区别在于, 当 'UNION'选择不同数据时, 'UNION ALL'选择所有数据.

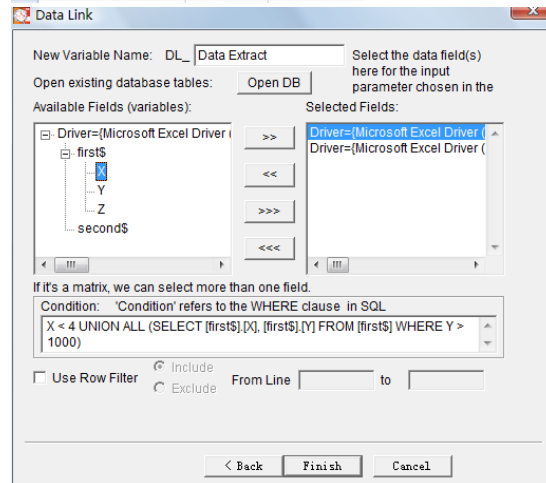
**SQL 描述:** [SQL 描述 1] UNION ALL [SQL 描述 2]

**范例:** `X < 4 UNION ALL (SELECT [first$].[X], [first$].[Y] FROM [first$] WHERE Y > 1000)`

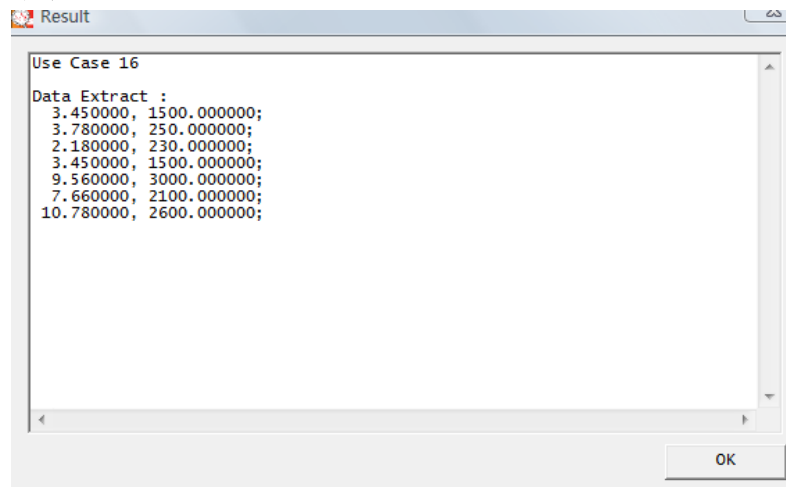
**范例 Profile 和模型:** 用户案例 21-25.re 和 用户案例 21 模型

**范例数据文件:** Sample Data 4.xls

1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



**结果:**



## 用户案例 22: SQL 功能的用法

**情景:** SQL 有一些算法功能, 'AVG', 'COUNT', 'MAX', 'MIN', 'SUM'. 在对结果进行一些操作时这个功能非常有用.

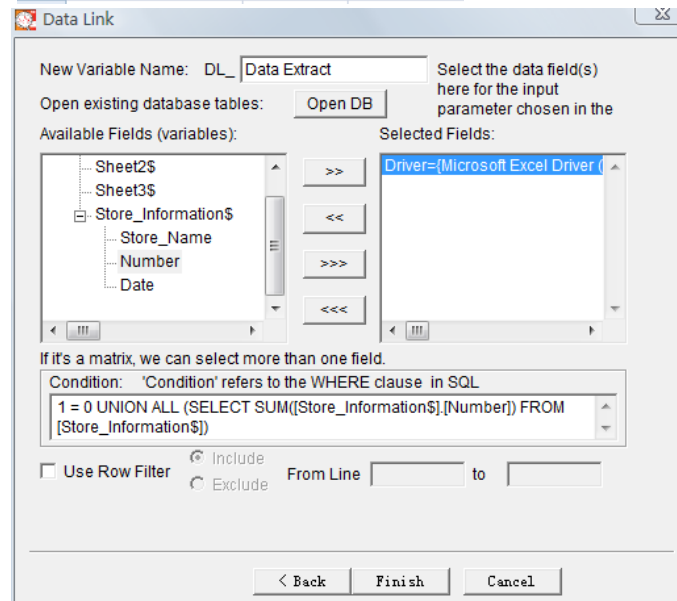
**SQL 描述:** `SELECT "function type"("column_name") FROM "table_name"`

**范例: 1 = 0 UNION ALL (SELECT SUM([Store\_Information\$].[Number]) FROM [Store\_Information\$])**

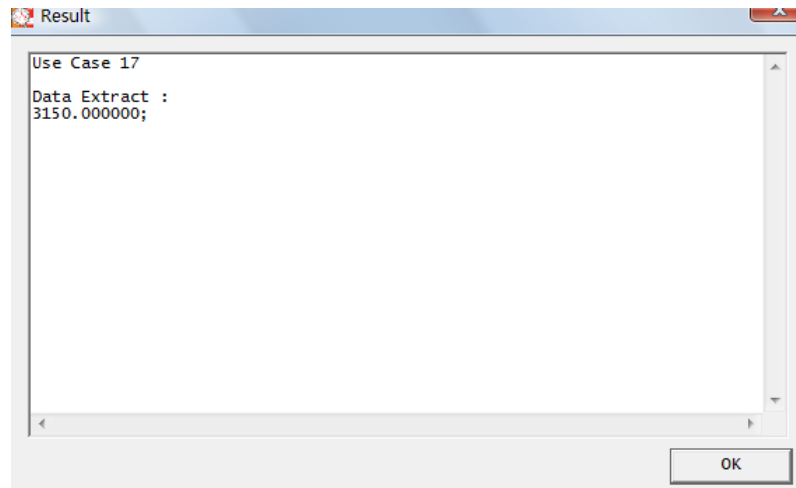
**范例 Profile 和模型:** 用户案例 20-25.re 和 用户案例 22 模型

**范例数据文件:** Sample Data 7.xls

1	Store_Name	Number	Date
2	Los Angeles	1500.00	2008/8/1
3	San Diego	250.00	2008/5/1
4	San Francisco	300.00	2008/6/31
5	Boston	700.00	2008/4/23
6	Los Angeles	400.00	2008/6/1



**结果:**



### 用户案例 23: 'GROUP BY'的用法

**情景:** 在用户案例 22 中, 我们可以通过 'sum'来计算所有数据, 但如果想计算每一个分值的值该如何操作呢? 我们可以运用'GROUP BY'.

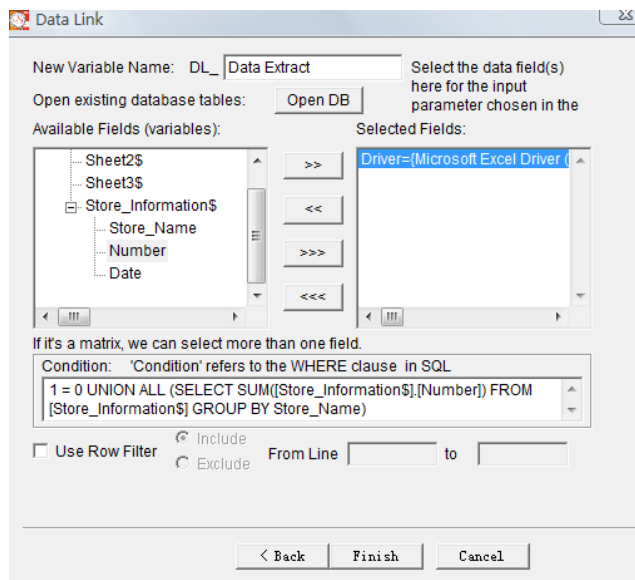
**SQL 描述:** `SELECT "column_name1", SUM("column_name2") FROM "table_name" GROUP BY "column_name1"`

**范例:** `1 = 0 UNION ALL (SELECT SUM([Store_Information$].[Number]) FROM [Store_Information$] GROUP BY Store_Name)`

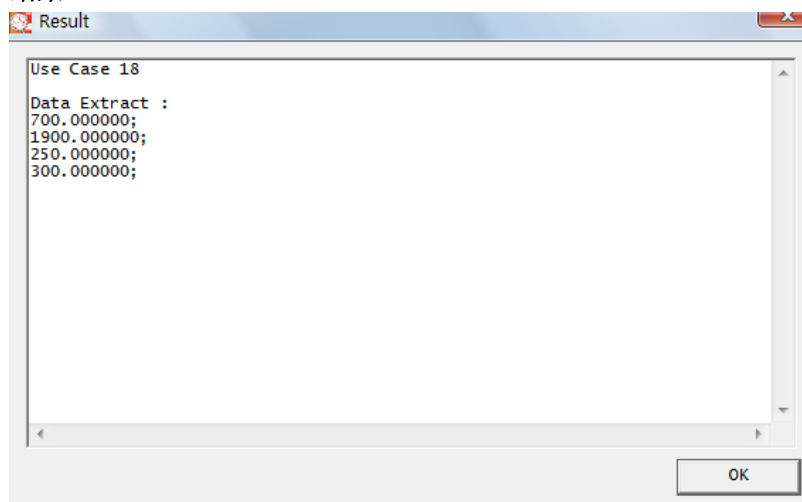
**范例 Profile 和模型:** 用户案例 20-25.re 和 用户案例 23 模型

**范例数据文件:** Sample Data 7.xls

1	Store_Name	Number	Date
2	Los Angeles	1500.00	2008/8/1
3	San Diego	250.00	2008/5/1
4	San Francisco	300.00	2008/6/31
5	Boston	700.00	2008/4/23
6	Los Angeles	400.00	2008/6/1



**结果:**



## 用户案例 24: 'DISTINCT'的使用

**情景:** 当一列数据中有一些数值相近, 您不希望显示他们, 可以使用'DISTINCT'命令来显示唯一值.

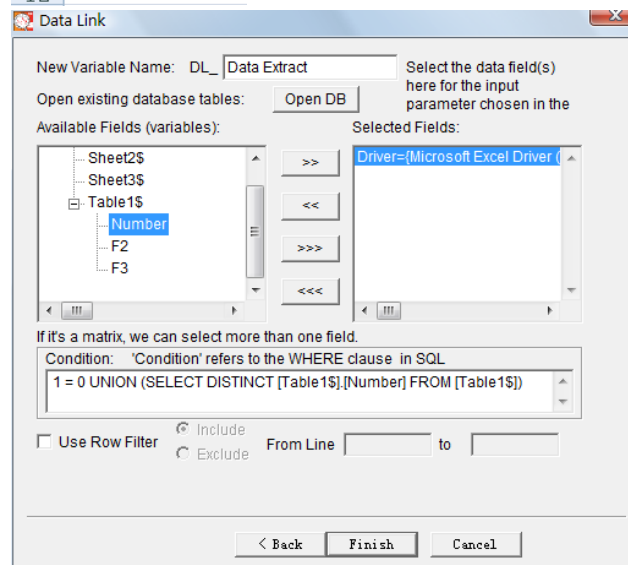
**SQL 描述:** `SELECT DISTINCT Variable FROM Table_name`

**范例:** `1 = 0 UNION (SELECT DISTINCT [Table1$].[Number] FROM [Table1$])`

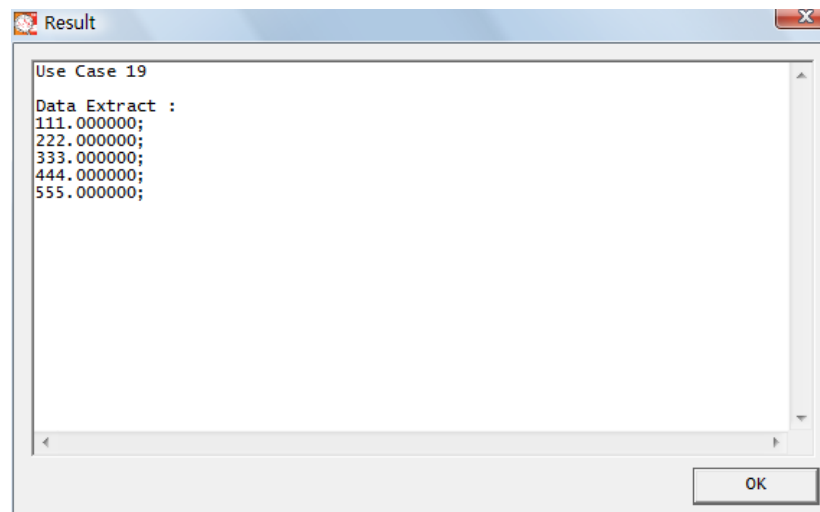
**范例 Profile 和模型:** 用户案例 20-25.re 和 用户案例 24 模型

**范例数据文件:** Sample Data 8.xls

	Number
1	
2	111
3	111
4	111
5	222
6	222
7	222
8	333
9	444
10	555
11	444
12	222



**结果:**



## 用户案例 25: 'ORDER BY'的使用

**情景:** 当您希望以特定的排序排列数据, 请使用'ORDER BY'命令.

**SQL 描述:** `SELECT "column_name" FROM "table_name" [WHERE "condition"]`

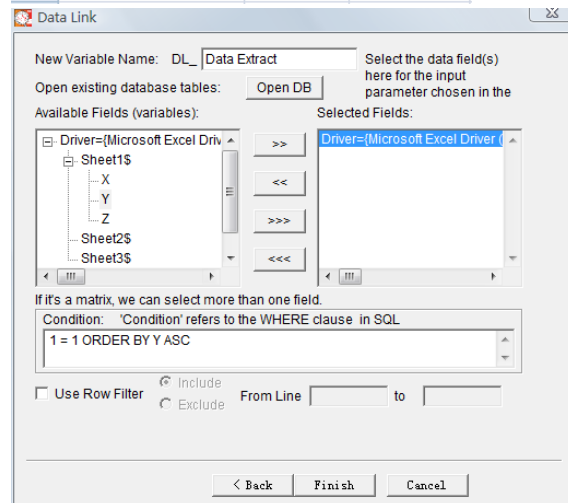
**ORDER BY** "column\_name" [ASC, DESC]

**范例:** `Number > 80` 和 `Number < 100`

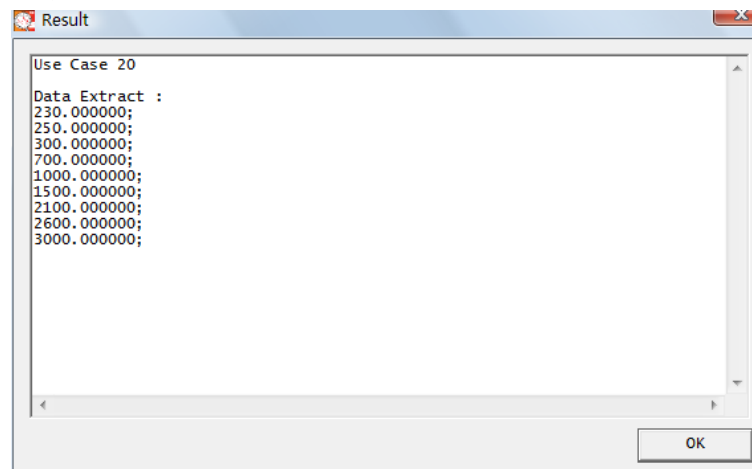
**范例 Profile 和模型:** 用户案例 20-25.re 和 用户案例 25 模型

**范例数据文件:** Sample Data 4.xls

1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



**结果:**



## 用户案例 26: 通过日期'BETWEEN'进行选择

情景: 'Between'可以用于 Date 变量但是需要特殊的格式.

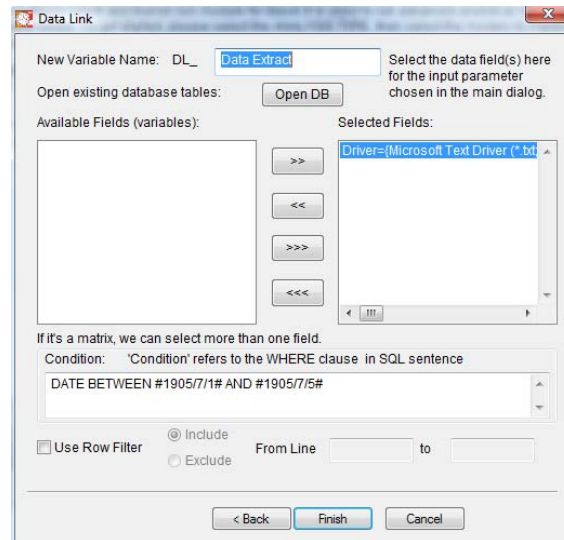
SQL 描述: **BETWEEN #date1# 和 #date2#**

范例: **DATE BETWEEN #1905/7/1# 和 #1905/7/5#**

范例 Profile 和模型: 用户案例 DATES.re 和 profile 中的两个模型

范例数据文件: **Sample Data 9.xls** 和 **Sample Data 10.csv**

	A	B	C	D
1	Normal (Multi)	Uniform	Binomial	DATE
2	87.53	45.29	6	7/1/1905
3	abc	45.29	6	7/2/1905
4	99.66	46.94	6	7/3/1905
5	108.75	45.96	6	7/4/1905
6	108.75	#\$45.96	6	7/5/1905



结果:

