第一章
KNIME简介与安装



目录

01 KNIME简介

02 KNIME安装及设置

03 扩展安装及删除

04 经典界面与现代界面

05 本章小结



01

KNIME简介

KNIME简介

01

KNIME的主要功能

KNIME是一款开源的数据分析、报告和集成平台，它支持数据挖掘、机器学习、统计分析等多种数据处理任务，广泛应用于科研和商业领域。

02

KNIME的产品体系

KNIME的产品体系包括KNIME Analytics Platform、KNIME Server和KNIME Hub等，这些产品共同构成了一个强大的数据处理生态系统，满足不同用户的需求。

03

KNIME的应用场景

KNIME在金融分析、机器学习、生物信息学、化学信息学等多个领域都有广泛的应用，通过其强大的数据处理能力，帮助用户解决复杂的数据分析问题。

主要功能

01 自动化工作流

KNIME的工作流可以完全自动化，支持定时执行和批量处理，大大提高了数据处理的效率和可重复性，适合大规模数据分析任务。

02 可视化设计

通过直观的拖放界面，KNIME允许用户轻松创建丰富的数据可视化，包括图表、图形和地图，使数据分析结果一目了然。

03 丰富扩展、低代码平台

KNIME提供丰富的扩展，使用节点构建数据处理的流程，同时还支持多种编程语言。

04 数据挖掘与分析等

KNIME提供强大的数据挖掘和分析工具，支持从简单的统计分析到复杂的机器学习算法，帮助用户深入探索数据背后的模式和趋势。



02

KNIME安装及设置

KNIME安装

01

下载KNIME安装包

访问KNIME官方网站，选择适合您操作系统的安装包进行下载。确保网络连接稳定，以便顺利完成下载过程。

02

运行安装程序

找到下载好的安装包，双击运行。根据提示完成安装向导

1.选择合适的安装路径和组件

2.设置可用内存量（也可事后再knime.ini文件内修改）

03

首次启动设置存储路径

安装完成后，首次打开KNIME软件时需要设置默认的文件保存路径（建议保存在非系统盘位置）

界面切换设置

界面切换设置的重要性

界面切换设置是KNIME软件使用中的关键步骤，它直接影响到用户的操作体验和工作效率。通过合理的界面设置，用户可以更快速地找到所需功能，提高工作效率。

经典界面与现代界面的切换方法

KNIME提供了两种主要的用户界面：经典界面和现代界面。用户可以根据个人喜好和工作需求，在软件设置中轻松切换这两种界面，以获得最佳的使用体验。

界面切换设置的个性化调整

除了基本的界面切换外，KNIME还允许用户进行个性化的界面设置，如调整工具栏布局、视图大小、关闭或显示界面，以满足不同用户的使用习惯，提升操作舒适度。



03

扩展安装、卸载

KNIME两种界面的比较

经典界面特点

01

经典界面以其直观的操作和熟悉的布局受到许多用户的喜爱，它保留了传统软件的菜单栏、工具栏等元素，使得用户可以快速上手。

现代界面优势

02

现代界面采用了更加简洁的设计语言，通过隐藏不必要的元素来减少视觉干扰，同时提供了更灵活的工作空间布局，适合追求高效工作的用户。

界面选择建议

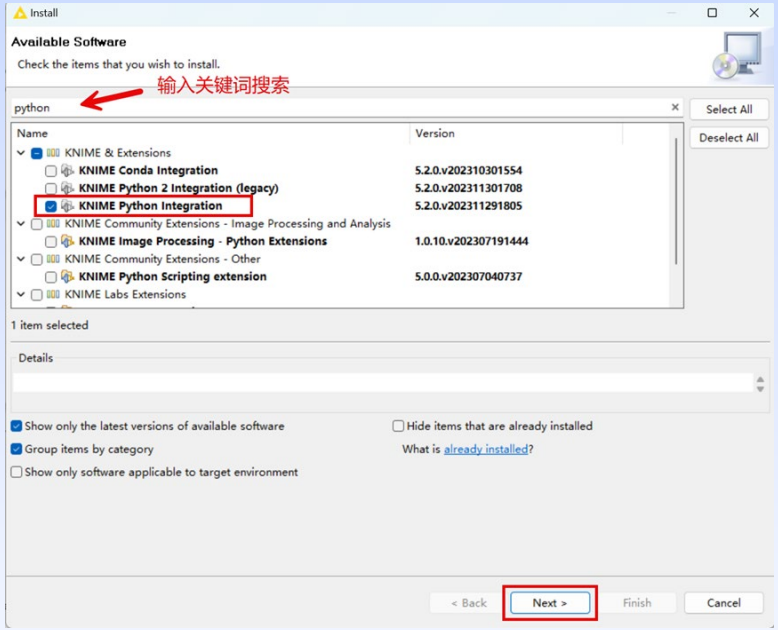
03

根据个人习惯和工作需求选择合适的界面，对于初学者或喜欢传统操作方式的用户，经典界面可能更为合适；而对于追求效率和现代化体验的用户，则推荐使用现代界面。

KNIME扩展安装及卸载

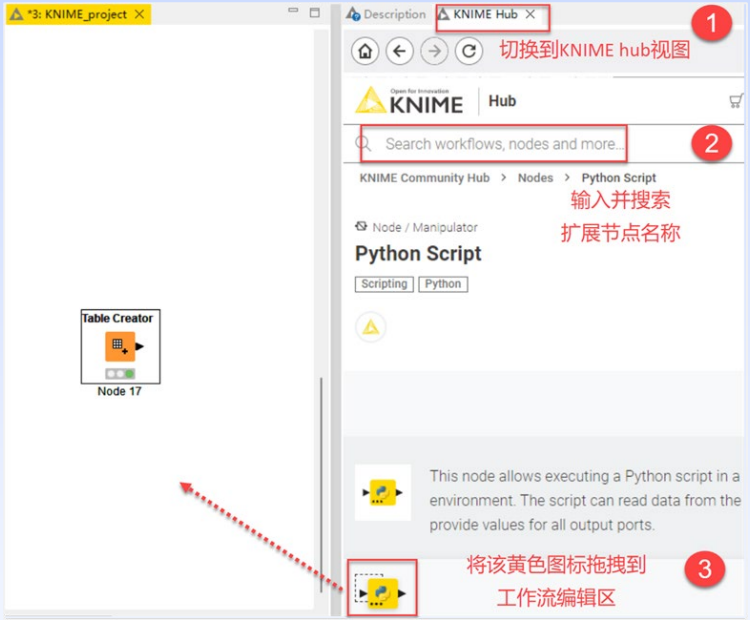
01 软件菜单中安装

通过KNIME的软件菜单直接安装扩展，用户可以选择所需的扩展包进行一键安装，这种方式简单快捷，适合初学者快速上手。



02 KNIME Hub拖拽安装

利用KNIME Hub网站提供的拖拽功能，用户可以将所需的扩展直接拖入KNIME工作区进行安装，这种方法直观且方便，增强了用户体验。

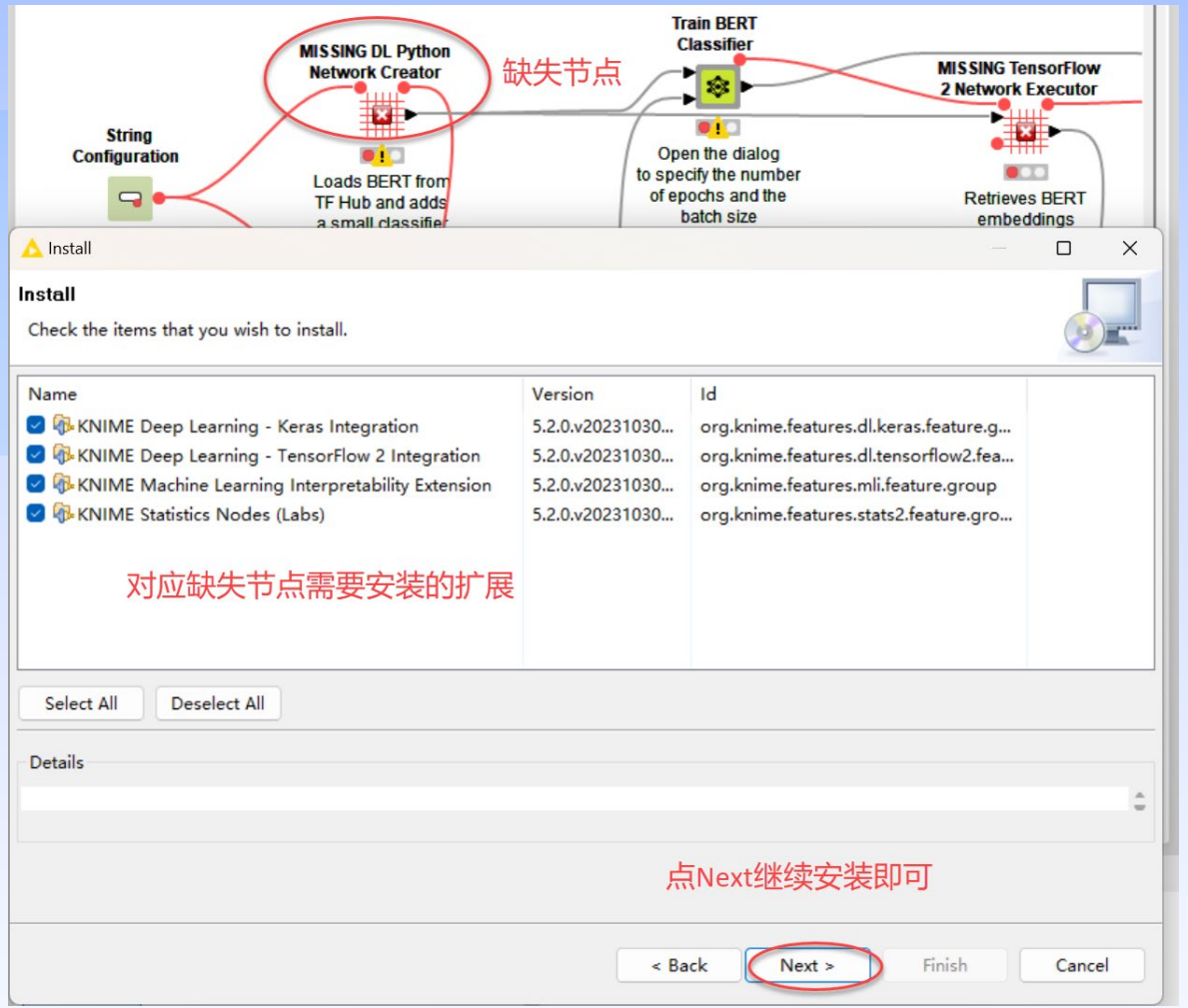


导入工作流时安装

03

安装缺失的节点

当导入的工作流包含尚未安装在本地KNIME环境中的节点时，系统会自动提示用户安装这些缺失的节点，确保工作流能够运行。



利用NodePit安装

01

NodePit简介

NodePit是一个在线平台，提供了大量的KNIME扩展节点，用户可以通过这个平台轻松找到并安装所需的扩展，以增强KNIME的功能。

02

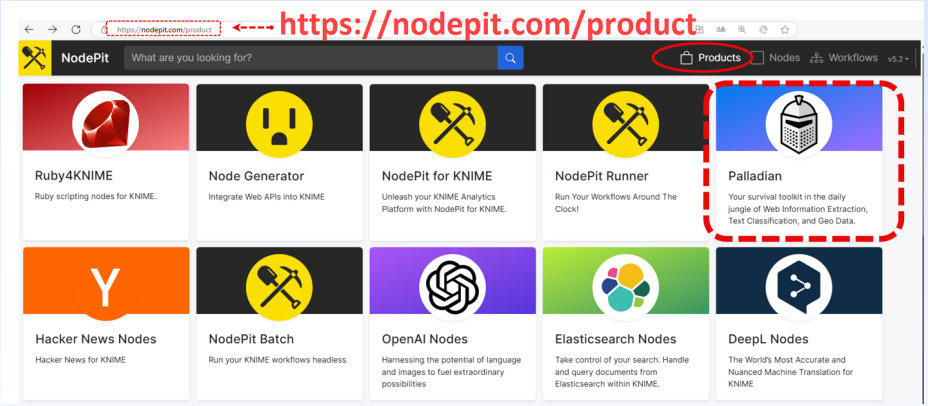
搜索与选择扩展

在NodePit上，用户可以根据关键词搜索相关的KNIME扩展，通过阅读扩展的描述和评价，选择最适合自己需求的扩展进行安装。

03

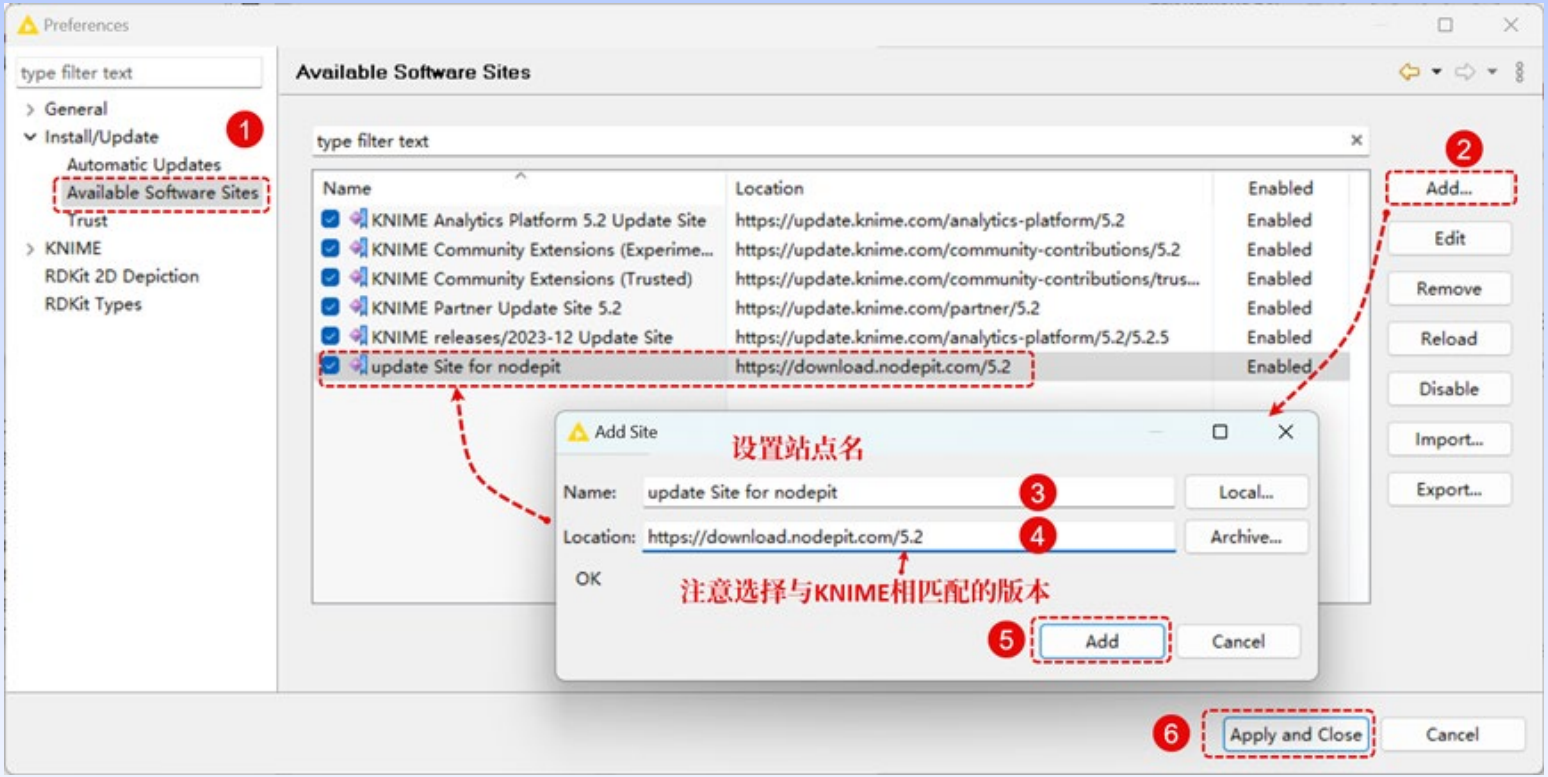
安装过程

安装扩展的过程简单快捷，用户只需点击“安装”按钮，NodePit会自动处理下载和安装过程，无需用户进行复杂的配置。



利用NodePit安装—步骤1： Preferences中设置可用更新站点

04

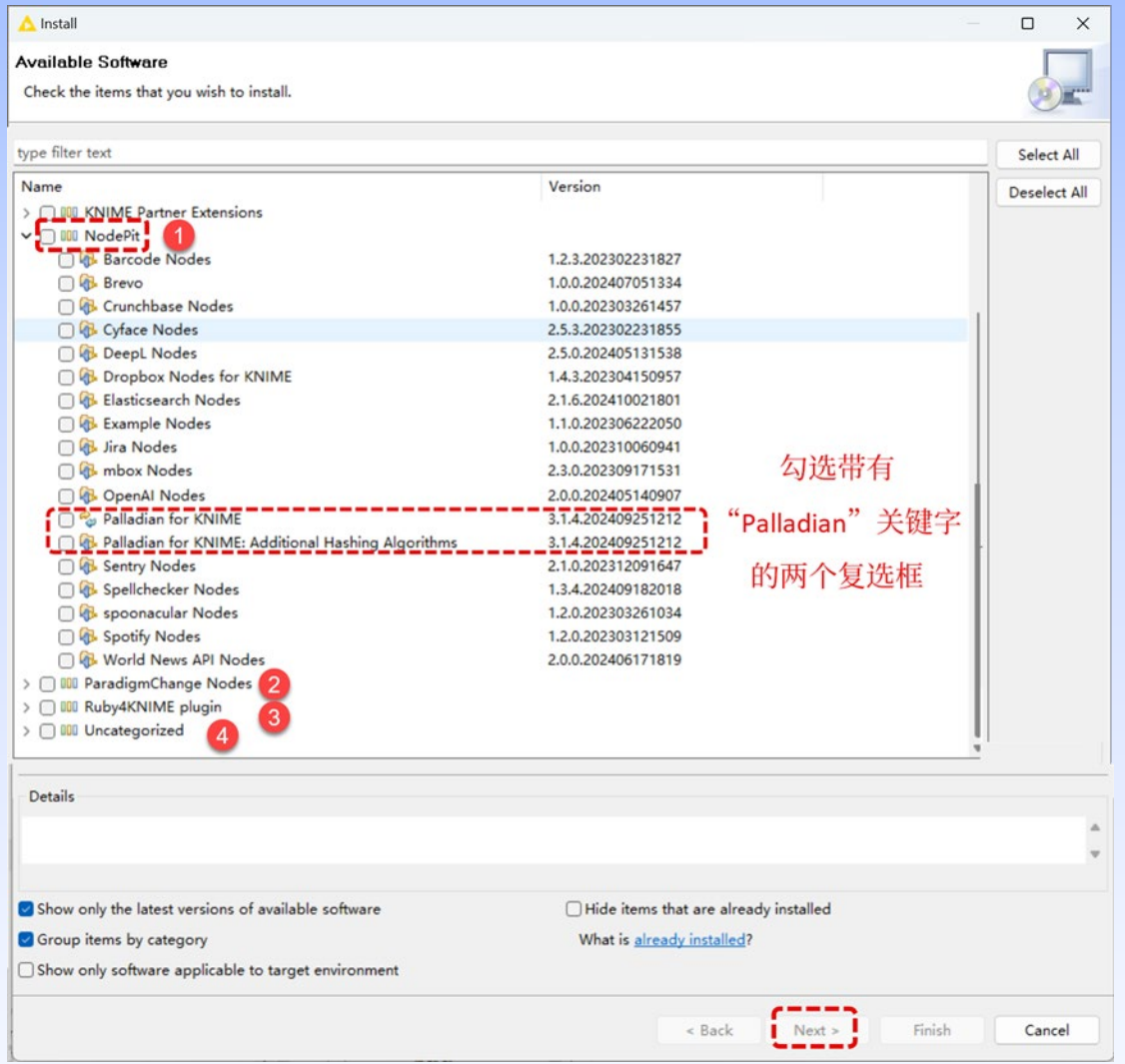


利用NodePit安装

步骤2：勾选安装

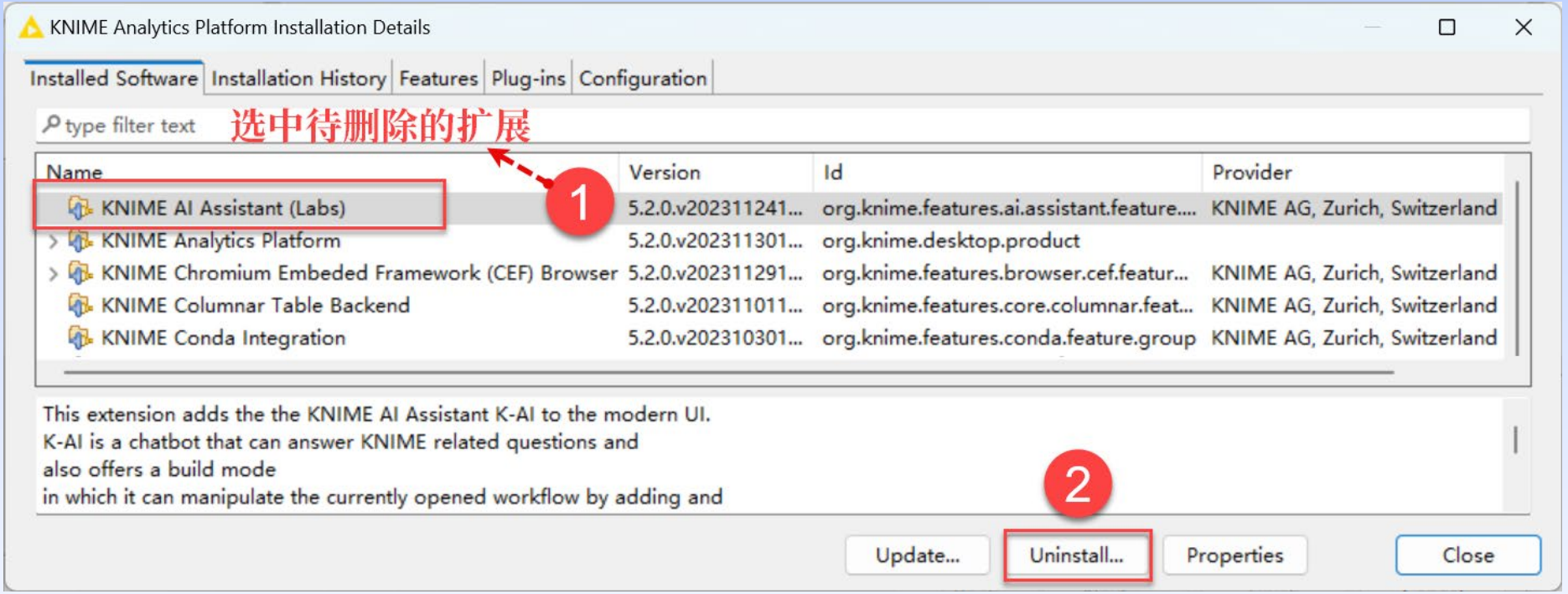
04

操作过程，详见本书附带视频教程：
补充知识-NodePit扩展安装



删除扩展

当不再需要某个扩展时，可以通过KNIME的扩展管理界面轻松卸载，保持软件的整洁和高效运行，避免不必要的资源占用。





04

经典界面与现代界面

经典界面

经典界面布局

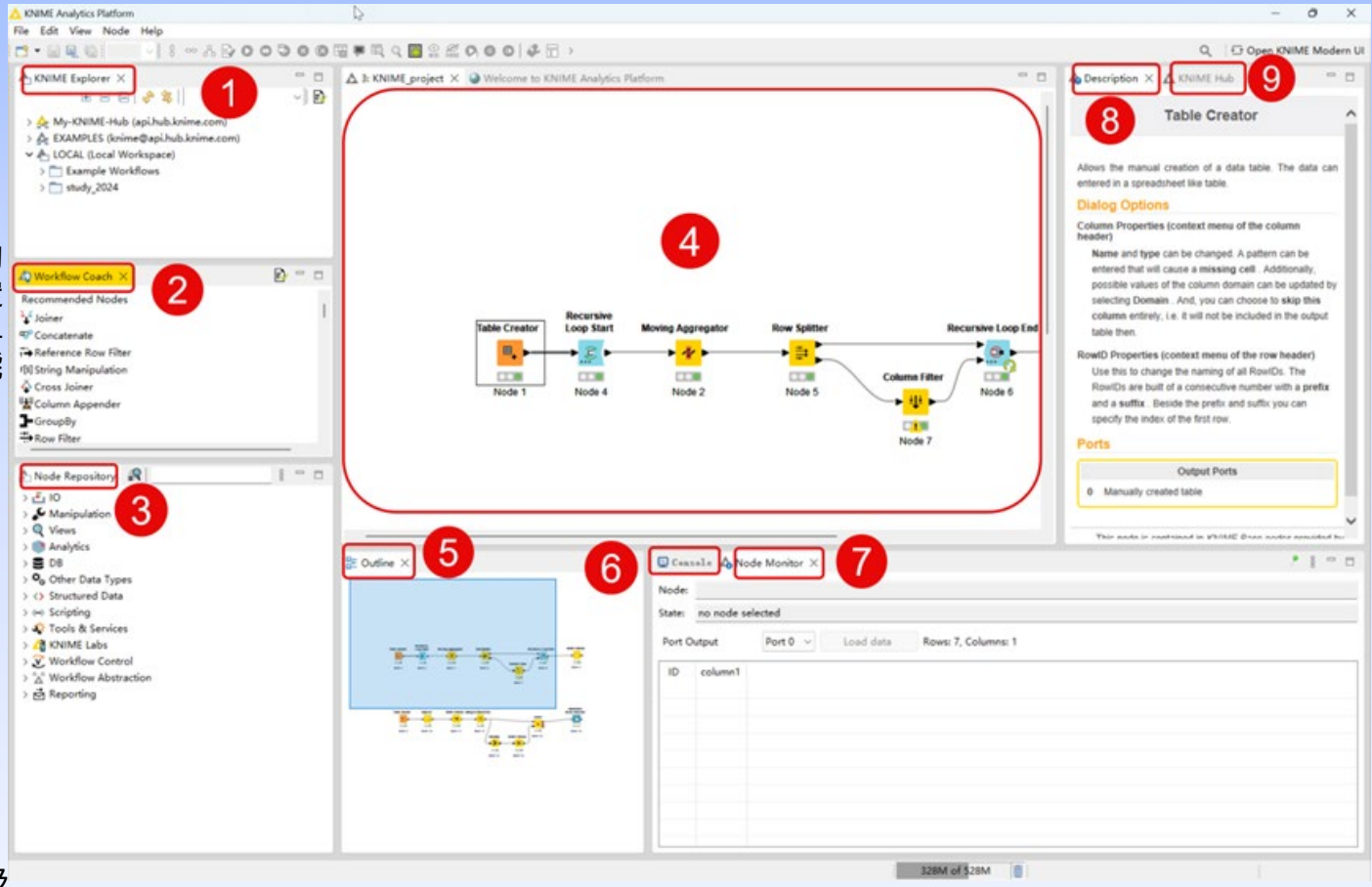
经典界面以其直观的布局和熟悉的操作方式，为用户提供了一个稳定的工作环境。它保留了传统的菜单栏、工具栏和工作区，使得用户能够快速找到所需功能。

功能模块组织

在经典界面中，功能模块被清晰地组织在不同的区域，如数据预处理、建模和分析等。这种模块化的设计有助于用户专注于特定任务，提高工作效率。

经典界面优势

经典界面比较成熟，管理工作流及节点更方便，部分功能现代界面下相比经典界面仍有缺失。



现代界面

现代界面概述

现代界面是KNIME软件中的一种用户界面模式，它采用了现代化的设计理念，提供了更加直观和便捷的操作体验。

现代界面特点

现代界面具有简洁明了的布局、丰富的可视化组件以及灵活的工作流程管理功能，使得数据分析过程更加高效和易于理解。

现代界面优势

相比经典界面，现代界面在用户体验上有了显著提升，如查看数据更方便，支持更多注释层级编排，节点查找及操作更方便，适合经验老手。



两种界面优缺点比较

01

经典界面的优点

经典界面以其直观的操作和熟悉的布局，为用户提供了稳定的工作环境。它适合那些习惯于传统软件操作方式的用户，减少了学习新工具的时间成本。--推荐新手选择该界面

02

两种界面的选择依据

选择经典界面还是现代界面，主要取决于用户的个人偏好和具体工作需求。对于追求稳定性和熟悉感的用户，经典界面是更好的选择；而对于追求高效和创新体验的用户，则可能更倾向于现代界面。

03

现代界面的优势

现代界面采用了更加现代化的设计理念，提供了更为丰富的视觉体验和交互功能。它通过优化用户界面和增强数据处理能力，提高了工作效率和用户体验。



05

经典界面与现代界面

本章小结

01 **KNIME简介及安装回顾**

本章介绍了KNIME的基本概念、主要功能和产品体系，以及如何进行软件的安装和基本设置，为后续的学习打下坚实的基础。

02 **KNIME扩展管理**

探讨了KNIME中扩展的安装与卸载方法，包括通过软件菜单、KNIME Hub拖拽、导入工作流时自动安装以及利用NodePit等多种方式，强调了扩展管理的重要性。

03 **KNIME界面比较分析**

对比分析了KNIME的经典界面与现代界面的特点，讨论了两种界面各自的优缺点，帮助用户根据自己的需求选择合适的操作界面。

感谢观看

THANKS



第二章
KNIME 必备基础

快速掌握 KNIME 核心概念与操作



目录

01 Node节点

02 工作流Workflow及工作流组 Workflow Group

03 工作流注释

04 数据类型及转换

05 核心概念

06 本章小结

本章概览


KNIME Explorer
My-KNIME-Hub (api.hub.knime.com)
EXAMPLES (knime@api.hub.knime.com)
LOCAL (Local Workspace)
Example Workflows
KNIME_study
chapter_2_basic_concept
01-概览
study_2024
8 工作流组
4: chapter2
*5: 01-概览
7 工作流
1 节点
4 连线
5 端口类型
6 新增端口
Read Excel Sheet Names
Table Row to Variable Loop Start
Excel Reader
Loop End
Excel Writer
读入一个文件的多个sheet名
生成参数逐个传递至reader节点
每次读取一个sheet表
合并结果
合并结果保存至本地
2 节点配置
3 节点状态



01

Node节点

Node节点状态

Node节点状态概述

Node节点的状态通过交通信号灯进行指示，包括未连接、未配置、缺乏输入数据时的红灯，准备执行的黄灯以及执行完毕的绿灯。这种可视化状态有助于用户快速了解每个节点的工作状态。

节点状态变化

节点的状态会随着操作的不同而变化。例如，当节点未连接到其他节点或缺乏输入数据时显示为红灯；完成配置并准备好执行时显示为黄灯；成功执行后显示为绿灯。

状态变化原因

节点状态的变化通常是由于端口连接、参数配置和执行操作的结果。只有当所有输入端口连接到先前节点且所有必需参数配置完毕后，节点才能从红灯状态切换到可执行状态。

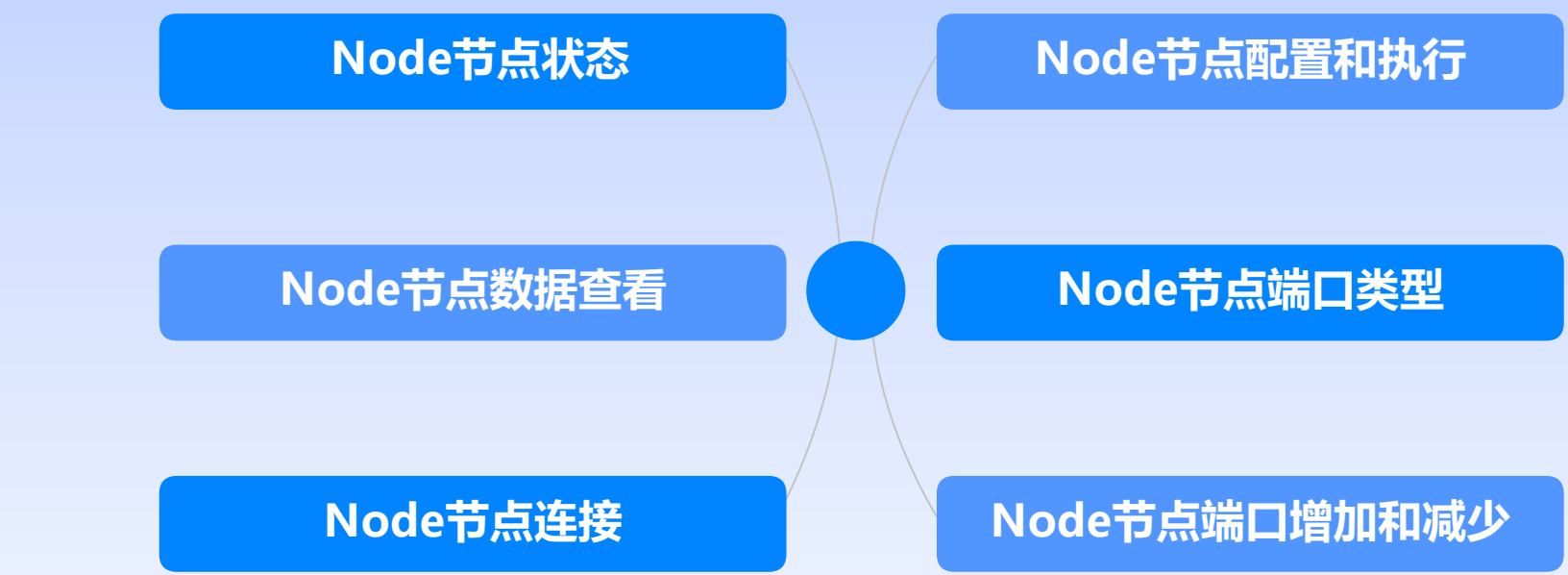
状态可视化

KNIME通过图形化界面展示每个节点的状态。用户可以通过节点顶部的信号灯颜色直观地了解其当前状态，从而更好地监控和管理工作流程中的各个节点。

状态与操作关联

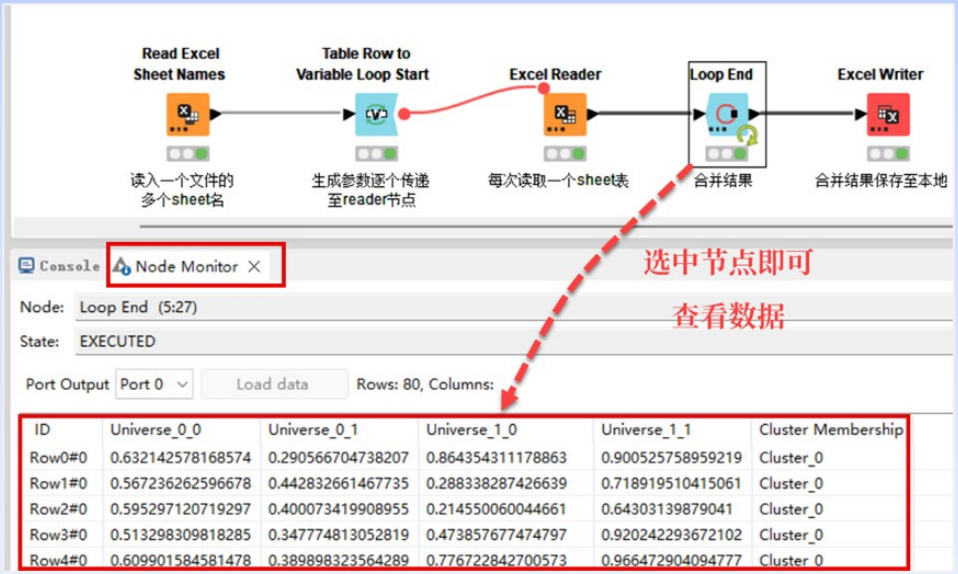
不同的操作会改变节点的状态。例如，启动节点将使其从红灯状态变为绿灯，表示已成功执行。停止节点则将其状态重置为未连接状态，以便于用户根据需求进行灵活控制。

Node节点配置和执行



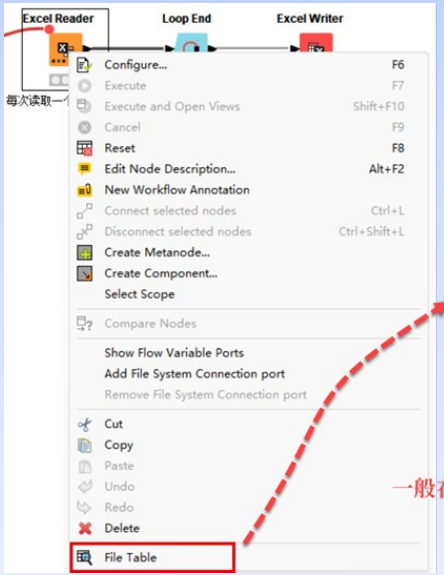
Node节点数据查看

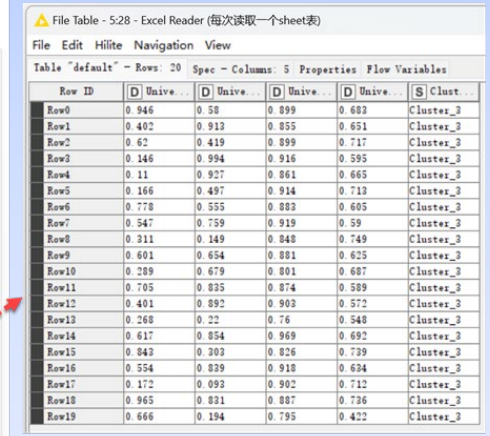
方法1



ID	Universe_0_0	Universe_0_1	Universe_1_0	Universe_1_1	Cluster Membership
Row0#0	0.632142578168574	0.290566704738207	0.864354311178863	0.900525758959219	Cluster_0
Row1#0	0.567236262596678	0.442832661467735	0.288338287426639	0.718919510415061	Cluster_0
Row2#0	0.595297120719297	0.400073419908955	0.214550060044661	0.64303139879041	Cluster_0
Row3#0	0.513298309818285	0.347774813052819	0.473857677474797	0.920242293672102	Cluster_0
Row4#0	0.609901584581478	0.389898323564289	0.776722842700573	0.966472904094777	Cluster_0

方法2





Row ID	D Unive...	D Unive...	D Unive...	D Unive...	S Clust...
Row0	0.946	0.58	0.899	0.683	Cluster_3
Row1	0.402	0.913	0.855	0.651	Cluster_3
Row2	0.62	0.419	0.899	0.717	Cluster_3
Row3	0.146	0.994	0.916	0.595	Cluster_3
Row4	0.11	0.927	0.861	0.665	Cluster_3
Row5	0.166	0.497	0.914	0.713	Cluster_3
Row6	0.778	0.555	0.883	0.605	Cluster_3
Row7	0.547	0.759	0.919	0.59	Cluster_3
Row8	0.311	0.149	0.848	0.749	Cluster_3
Row9	0.601	0.654	0.881	0.625	Cluster_3
Row10	0.289	0.679	0.801	0.687	Cluster_3
Row11	0.705	0.835	0.874	0.589	Cluster_3
Row12	0.401	0.892	0.903	0.572	Cluster_3
Row13	0.268	0.22	0.76	0.548	Cluster_3
Row14	0.617	0.854	0.969	0.692	Cluster_3
Row15	0.843	0.303	0.826	0.739	Cluster_3
Row16	0.554	0.839	0.918	0.634	Cluster_3
Row17	0.172	0.093	0.902	0.712	Cluster_3
Row18	0.965	0.831	0.887	0.736	Cluster_3
Row19	0.666	0.194	0.795	0.422	Cluster_3

一般在右键菜单底部，选中后查看该节点的全部数据

备注:经典界面下查看数据的两种方法

Node节点端口

01 输入端口

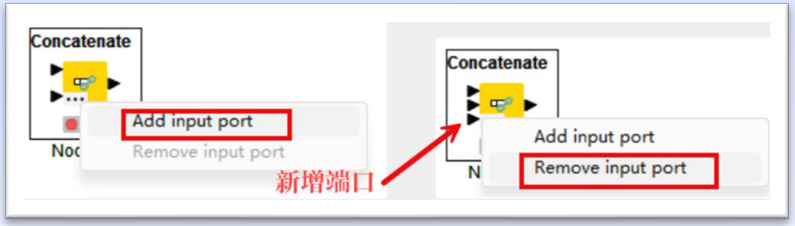
输入端口用于接收来自前一个节点的输出数据。它通常显示为蓝色，并带有“输入”标签，表示该端口用于接收外部数据流。

02 输出端口

输出端口用于将处理后的数据发送到后续节点或文件、数据库等。它通常显示为橙色，并带有“输出”标签，表示该端口用于发送处理结果。

03 添加、删除端口

控制端口用于传输控制信息，如配置参数和命令行指令。它不传输实际数据，但用于节点的启动、配置和状态监控。



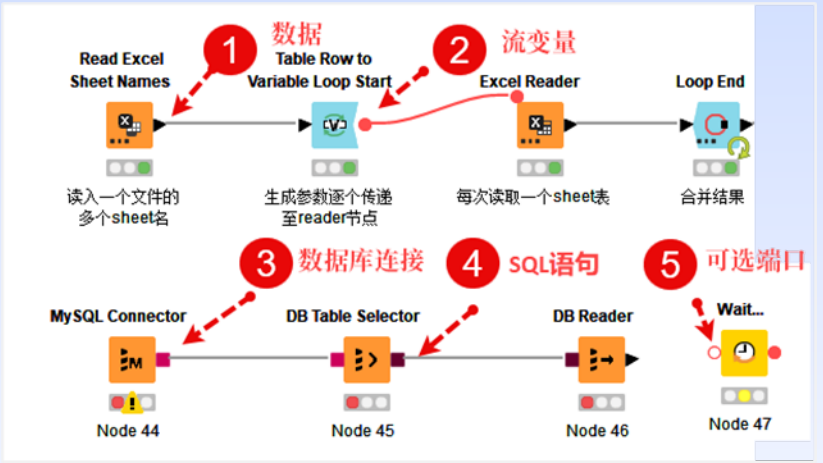


04 数据端口

数据端口用于传输数据表，使用黑色三角表示，最为常用

05 流变量端口

另外一个最常见的端口类型，红色先连接。



Node节点端口增加和减少

端口定义

在KNIME中，每一个Node节点都拥有输入端口和输出端口。输入端口用于接收外部数据或命令，而输出端口则用于发送处理后的结果或状态信息。

增加端口

要增加新的端口，首先需要选中目标Node节点。接着，通过右键点击并选择“节点”菜单中的“打开”，进入节点配置界面，最后添加新的输入或输出端口。

减少端口

要减少端口，需要先确认当前Node节点所有使用的端口。在节点配置界面中，可以移除不需要的端口。移除端口时需注意保留必要的输入和输出端口以确保流程的正确性。

端口状态指示

KNIME通过不同颜色的交通信号灯来指示节点的状态。分支中也存在端口状态，激活分支端口正常，未激活分支红色叉号表示

Node节点连接

Node节点连接概述

在KNIME中，Node节点是数据操作的基本处理单元。为了实现节点的成功执行，需要将其连接到工作流中，以获得数据流并进行相应的操作。

输出端口与输入端口连接

通过点击Node节点的输出端口并拖动到适当的输入端口来连接它们。确保所有必要的端口已正确配对，以确保数据能顺畅地在节点间流动。

配置节点连接状态

配置节点时，需将节点调整为红色状态图标，表示其已准备好进行配置和执行。完成配置后节点将显示绿色状态，表明其已成功连接到工作流中。

管理多个Node节点连接

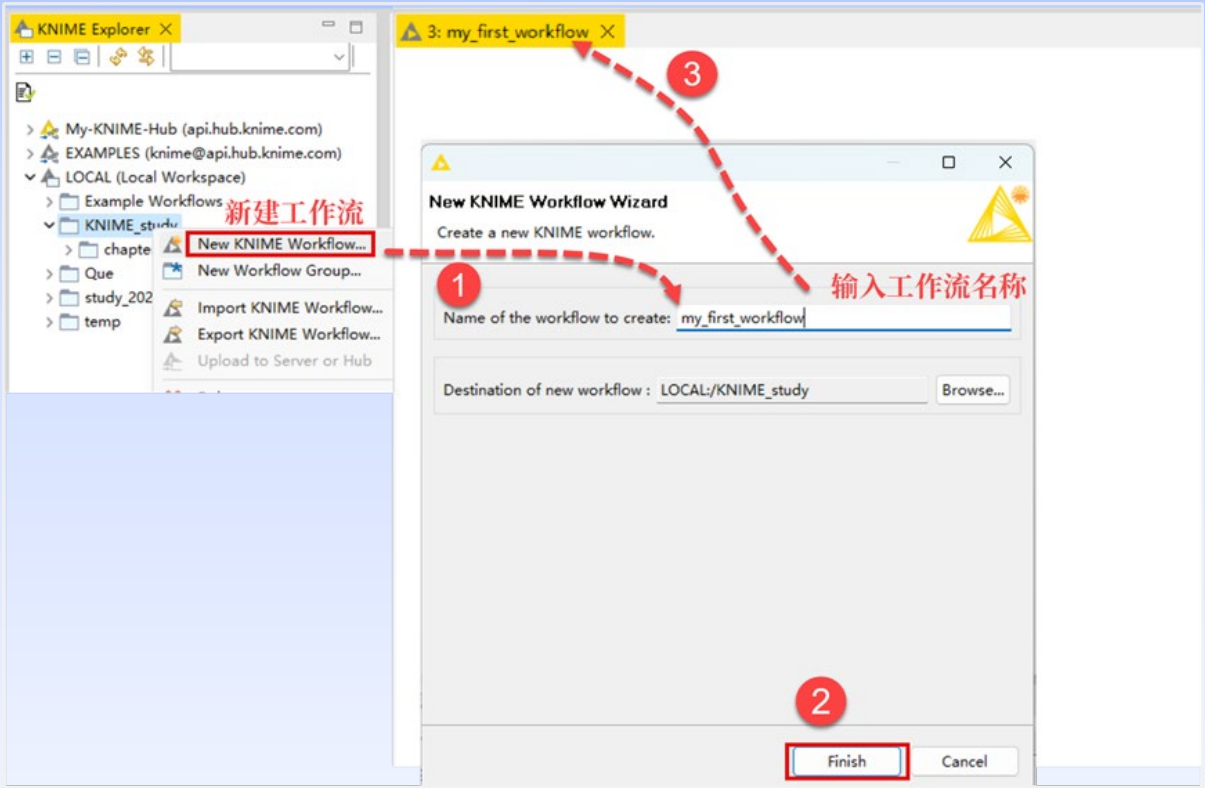
可以同时管理多个Node节点的连接，通过KNIME Explorer中的挂载点进行管理。挂载点允许用户方便地查看、添加和移除与特定节点的连接。



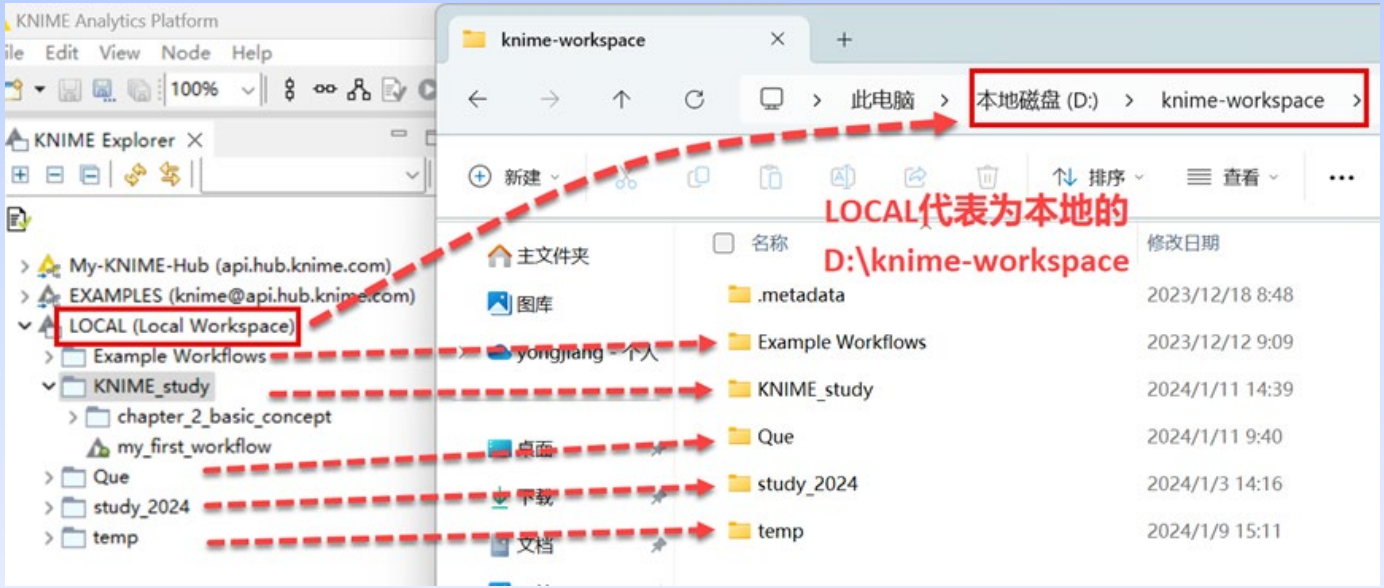
02

工作流Workflow及工作流组Workflow Group

新建工作流


KNIME Explorer
My-KNIME-Hub (api.hub.knime.com)
EXAMPLES (knime@api.hub.knime.com)
LOCAL (Local Workspace)
Example Workflows
新建工作流
chapte
Que
study_202
temp
New KNIME Workflow...
New Workflow Group...
Import KNIME Workflow...
Export KNIME Workflow...
Upload to Server or Hub
3: my_first_workflow
3
New KNIME Workflow Wizard
Create a new KNIME workflow.
1
输入工作流名称
Name of the workflow to create: my_first_workflow
Destination of new workflow : LOCAL:/KNIME_study
Browse...
2
Finish
Cancel

新建工作流组



同新建工作流操作

注意，创建一个新的工作流或工作流组，本地将新增一个文件夹

导出与导入工作流

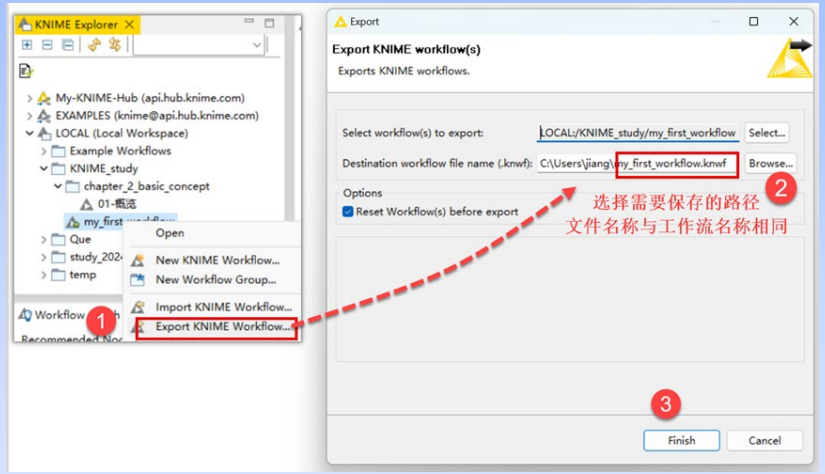
01 导出工作流步骤

方法1

“File”菜单，选择“Export”。接着选择保存路径并命名文件，最后点击“Save”即可。

方法2:

见右侧图标



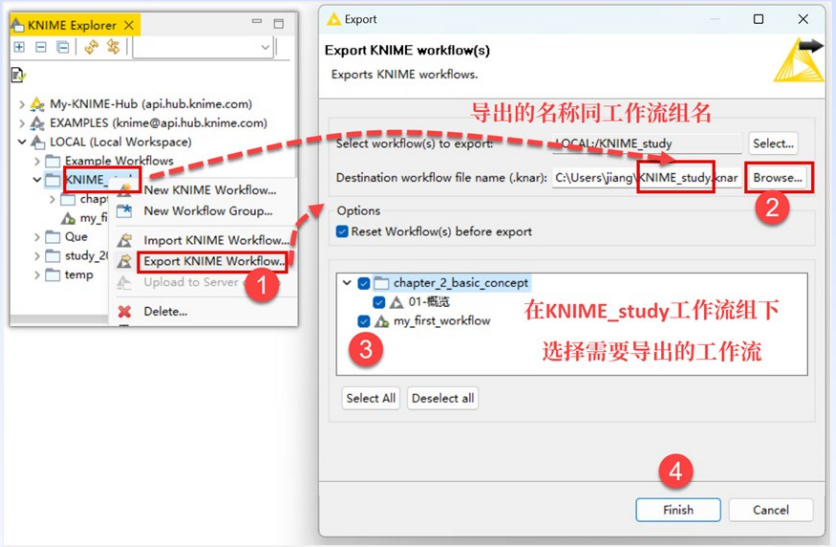
02 导入工作流步骤

方法1

导入工作流时，需要从菜单中找到“File”并选择“Import”，选择需要导入的工作流文件。

方法2

见右侧图标



工作流其他操作

01 工作流属性编辑

02 删除工作流

03 工作流重命名

04 移动工作流

05 查看工作流位置等



03

工作流注释

添加编辑注释（注意两种界面操作差异）

01 添加注释

在KNIME中，可以通过工具栏中的“添加注释”按钮为工作流添加注释。注释用于提供额外的说明或信息，帮助理解和解释工作流程的各个步骤。

02 编辑注释内容

选中需要编辑的注释后，可以在右侧的注释面板中直接修改内容。确保使用简洁明了的语言描述注释的信息，以便其他用户能够快速理解注释的意图。

03 删除注释

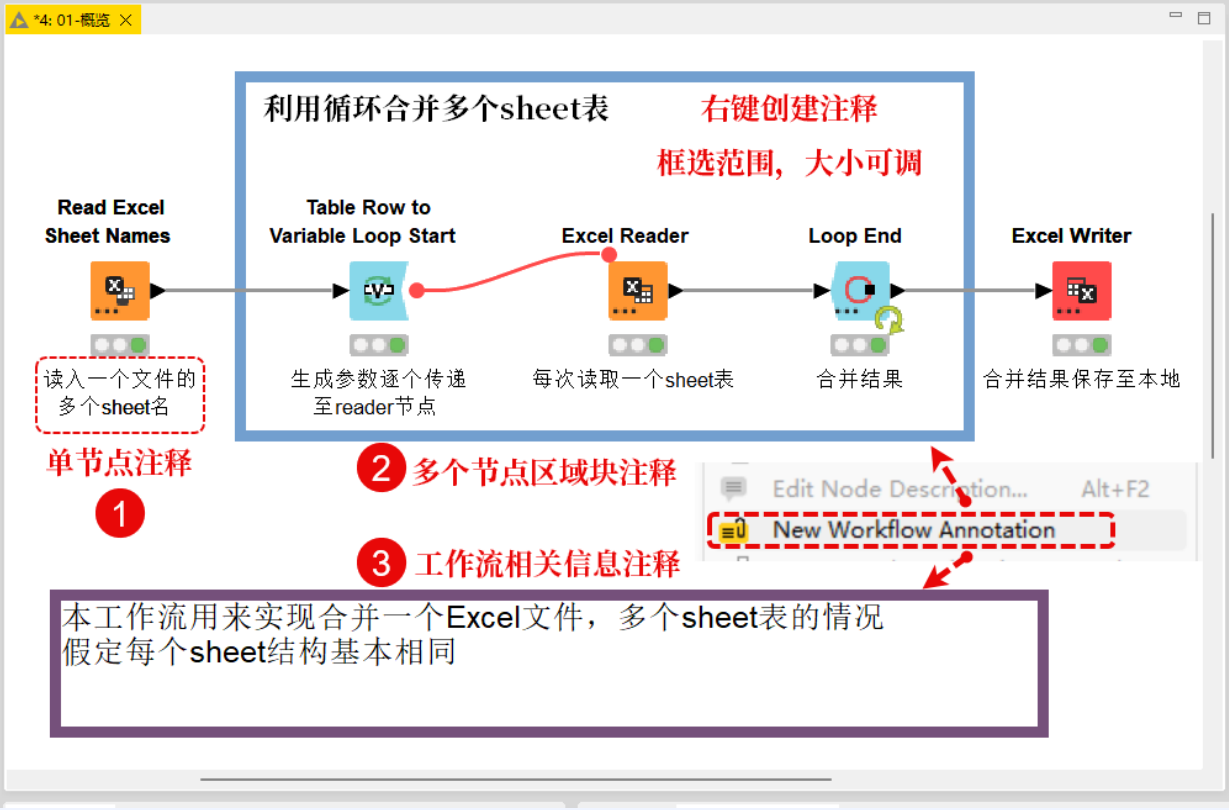
选中不需要的注释后，可以使用工具栏中的“删除”按钮将其移除。确保在删除注释前已经备份重要信息，以免误操作导致不必要的数据丢失。

04 排序和组织注释

通过工具栏中的选项，可以对注释进行排序和组织。按照创建日期、标题或其他标准对注释进行排序，有助于提高工作流程的可读性和逻辑性。

--现代界面下编辑注释更方便和美观

注释的不同类型





1）经典界面下，点击工作流任意空白处，才能出现编辑按钮。

2）现代界面下，该界面出现在左上角。



04

数据类型及转换

数据类型

数据类型概述

KNIME支持多种数据类型，包括数值型、字符串型、布尔型等。这些数据类型适用于不同类型的数据分析需求，用户可以根据实际业务选择适合的数据类型。



数据类型转换

在KNIME中，用户可以轻松地进行数据类型转换。通过使用相应的节点和工具，可以实现不同数据类型之间的转换，如将字符串数据转换为数值型，以便于后续的分析和处理。

数据类型验证

KNIME提供了数据类型验证功能，确保数据在进入下一步处理前符合预期的类型。这有助于预防因数据类型错误而导致的分析结果不准确问题，提高数据预处理的效率和准确性。

数据类型相关节点

KNIME中包含多种与数据类型相关的节点，如数值转换节点、字符串操作节点和类型校验节点。这些节点方便用户根据需要对数据进行类型转换和验证，增强数据处理能力。

数据类型转换

数据类型基本概念

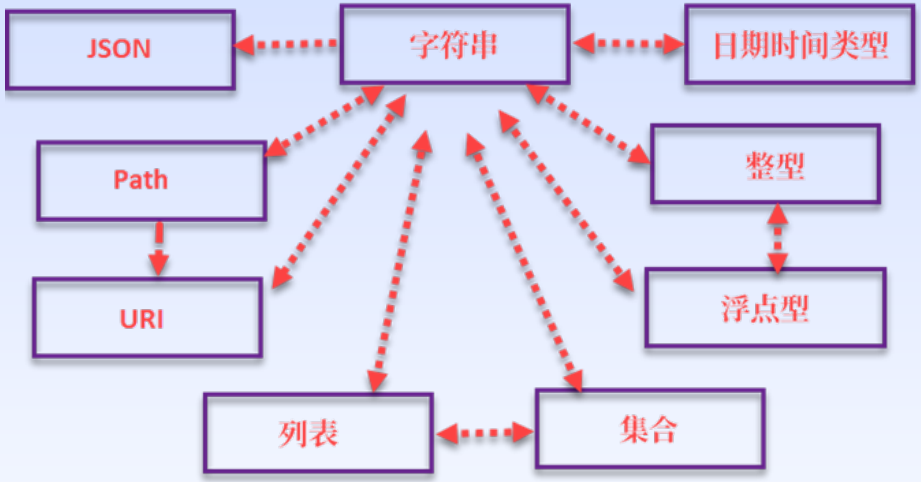
数据类型是数据在数据库或程序中的具体格式，它定义了数据如何被存储和操作。在KNIME中，常见的数据类型包括数值型、字符串型、日期型等，每种类型都有其特定的用途和转换规则。

整型　日期类型　浮点型　字符串　列表　集合

编号	日期	金额	产品	产品_SplitResultList	产品_SplitResultSet
1	2023-01-05	396	A, A, C	[A, A, C]	[A, C]
2	2023-01-06	455	C, D, A	[C, D, A]	[C, D, A]
3	2023-01-07	575	A, C, C	[A, C, C]	[A, C]
4	2023-01-08	773	A, B, B	[A, B, B]	[A, B]
5	2023-01-09	501	A, D, E	[A, D, E]	[A, D, E]
6	2023-01-10	337	A, C	[A, C]	[A, C]

字符串数据转换

字符串型数据用于存储文本信息，如姓名、地址等。在KNIME中，字符串数据可以通过String Manipulation节点进行转换





05

核心概念

KNIME工作流程

01

工作流程定义

工作流程是一系列相互连接的节点组成的序列，用于自动化数据处理和分析任务。每个节点执行特定的操作，整个流程按照预设的顺序依次执行，实现数据分析的自动化。

02

工作流创建与管理

在KNIME中，可以通过新建工作流来定义数据处理任务。工作流由多个节点组成，用户可以根据需求添加、删除或调整节点顺序，以满足不同的数据分析需求。

03

工作流配置与执行

配置工作流涉及为每个节点分配输入数据、设置执行参数和选择输出路径。配置完成后，通过KNIME Workbench可以启动工作流的执行，系统将自动按顺序调用各个节点完成数据处理。

04

工作流监控与调试

在工作流执行过程中，可以通过KNIME Workbench监控各节点的执行情况和中间结果。如果出现错误或需要调整，可以暂停并进入调试模式，定位问题并进行修正。

05

工作流分享与协作

KNIME Hub是一个公共存储库，用户可以将自己的工作流和节点上传到KNIME Hub，与其他用户共享。此外，还可以通过社区资源获取示例工作流和预打包组件，提高数据分析效率。

KNIME核心概念

编号	核心概念	说明
1	Extension	KNIME扩展，通过安装扩展，可以在节点库中添加更多节点。Extension类似于编程语言中的Library（库）、包（Package）、模块（Module）
2	Node	实现某一功能的节点，可类比编程中的函数。一般需要在节点配置选项内设置相关参数。
3	Workflow	工作流，KNIME中，在工作流编辑区内将一个个节点有机连接起来，实现特定功能或解决某个问题，是各个节点的有机组合和搭配。
4	Workflow Group	工作流组，存储各个工作流的地方。类似文件夹，通过工作组更好的分门别类的管理工作流。每个工作流组实质对应本地一个文件夹。
5	Metadata	元数据，描述数据的数据，数据表的类列，数据类型，列序索引、值域范围等，查看数据表时的Spec提供此类信息，或者从节点Extract Table Spec获取数据表的元数据。
6	Metanode	元节点，将多个连续区域内的节点打包为一个节点，简化工作流，使之看起来更整洁，隐藏不必要的细节。
7	Component	将多个连续区域内的节点打包为一个节点，除了简化工作流外目的外，组件有多种不同的用途。本书第十三章至十六章详细展开了组件的不同应用。目的不同，选择的节点也不同，例如组件内部是选择Configuration类节点还是Widget类节点。组件属于进阶学习的核心概念之一。
8	Flow Variable	工作流变量，简称流变量。类似编程语言中的变量概念。KNIME中利用流变量可以实现不同数据间的联动，参数化、自动化，提高效率，Component中也必不可少，绝对的核心基础概念。掌握流变量才能理解和学好循环，流变量也是进阶的必备基础，第九章单独对此展开。



06

本章小结

总结本章节内容

本章概述

KNIME是一个开源的数据挖掘和可视化平台，适用于教学、研究和实际应用。它通过节点的协同工作，实现数据处理、分析和可视化，具有高度的灵活性和扩展性。

Node节点详解

Node节点是KNIME的核心组件，用于执行各种数据处理任务。每个Node节点代表一个特定的功能，如数据转换、清洗、聚合等，可以灵活组合以完成复杂的数据分析流程。

工作流与工作流组

工作流是KNIME中用于组织和执行任务的框架，工作流组则进一步细分了工作流内的任务分组。通过创建和管理工作流与工作流组，用户能够有效地管理和执行复杂的数据分析任务。

数据类型及转换

数据类型是KNIME中的基本概念，不同的数据类型支持不同的数据处理操作。数据类型转换则是将一种数据类型转换为另一种，以满足特定的分析需求，是数据分析的重要步骤。

提供实践建议

实践重要性

将KNIME应用于实际工作中，通过解决具体问题来提升工具的使用技巧和业务理解。实践是检验学习成果的最佳方式，能够有效促进技能的巩固和提升。

工作流设计最佳实践

创建高效的工作流需要遵循一系列最佳实践，如模块化设计、避免不必要的复杂性和确保可重用性。这些原则有助于提高工作流的效率、安全性和可维护性。



多看帮助文档

每个节点都提供了详细的参考文档，遇到不懂的参数选项时，及时查询文档

持续学习和改进

持续关注KNIME的最新更新和社区动态，不断学习新功能和改进方法。多参与社区交流，使用KNIME解决问题，不断提升自身技能和工作流开发水平。

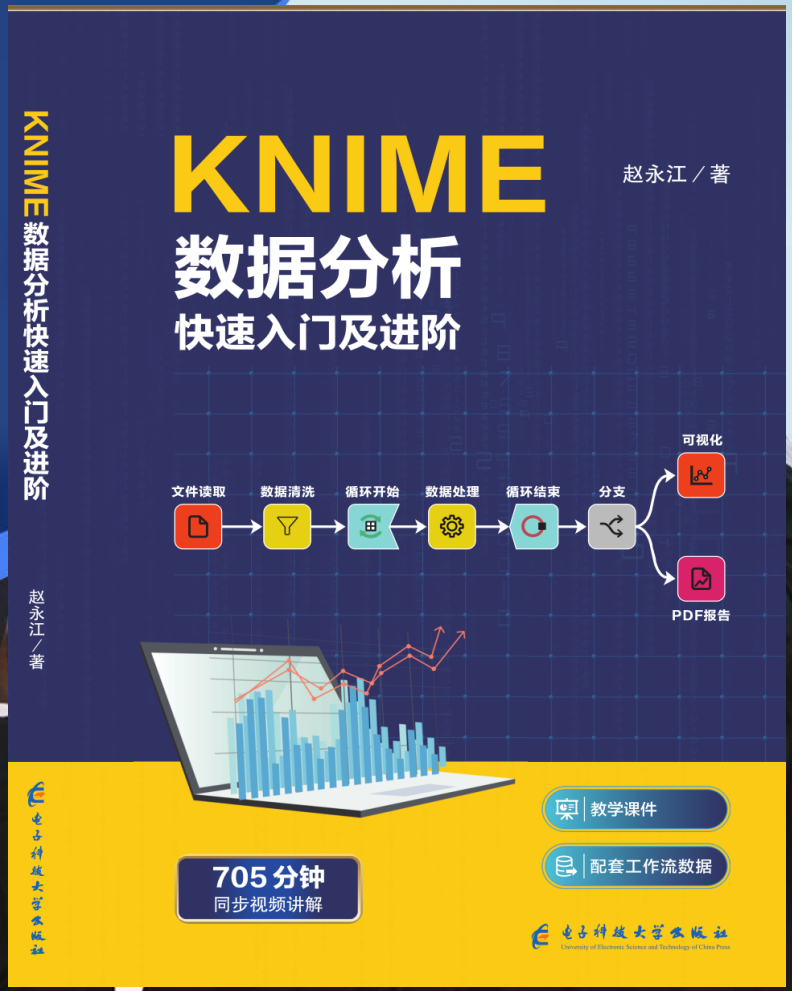
感谢观看

THANKS



第三章
数据分析概览及数据读写

从数据分析方法到数据读写实践



目录

01 数据分析流程简介

02 数据结构和数据来源

03 常用数据读入节点

04 常用数据写入类节点

05 文件路径

06 本章小结



01

数据分析流程简介

常见数据分析方法

描述性分析法

用于总结数据的主要特征，如平均数、中位数、众数、标准差等

诊断性分析

识别和探究数据中的模式和原因

实验性分析

通过实验设计验证假设

预测性分析法

利用历史数据预测未来趋势或结果，常结合机器学习、深度学习等模型来预测

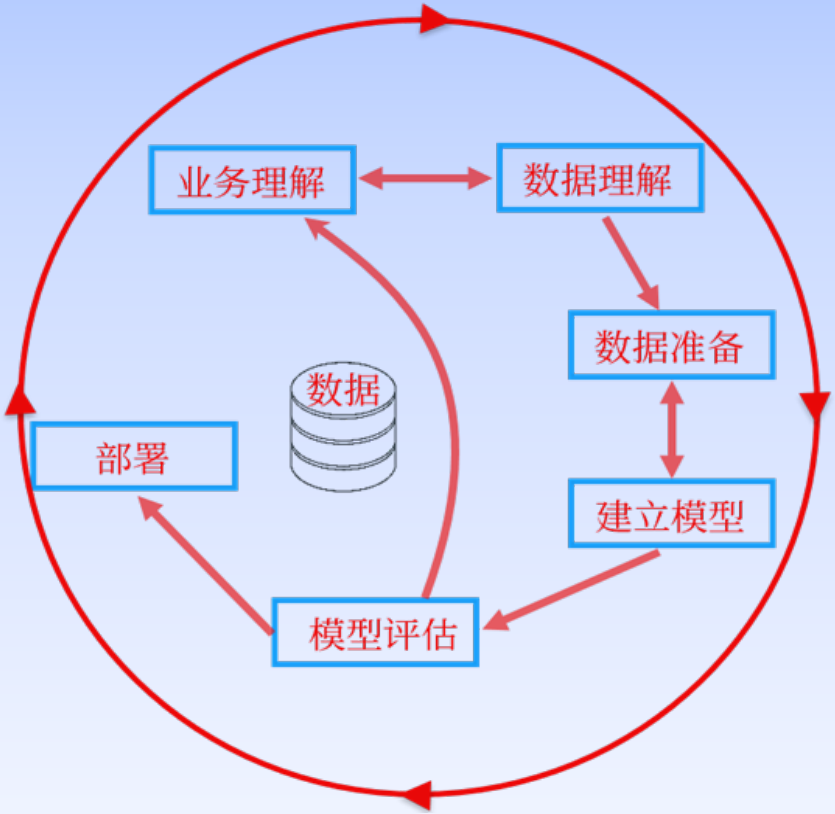
数据清洗与预处理

旨在提高数据质量，为后续分析做好准备

大数据分析等

利用分布式技术对海量数据进行处理

数据挖掘标准流程



数据挖掘标准流程

商业理解

商业理解阶段旨在明确数据挖掘项目的目标和业务需求。通过分析商业背景、确定成功的衡量标准，评估成本与收益，识别潜在风险，初步了解行业术语，为后续的数据挖掘工作提供明确方向。

数据理解

数据理解阶段涉及对数据的初步探索，包括了解其来源、结构和质量。此阶段需要检测数据质量、描述数据特征、探测数据意义，并基于业务需求提出假设以验证其潜在价值和知识。

数据准备

数据准备阶段对原始数据进行处理，包括清洗、转换、集成和建模等步骤。这一过程确保数据满足分析需求，提高数据质量，并为建立有效的模型打下坚实基础。

建立模型

建立模型阶段利用预处理后的数据建立预测或分类模型。选择合适的算法和参数，训练模型，并通过交叉验证等方法优化性能，以提高模型的准确性和可靠性。

模型评估与部署

模型评估阶段通过多种评估方法检验模型的性能，包括准确率、召回率和F1分数等指标。评估结果用于调整模型参数，改进模型效果。最终将模型部署到生产环境，实现数据挖掘目标。

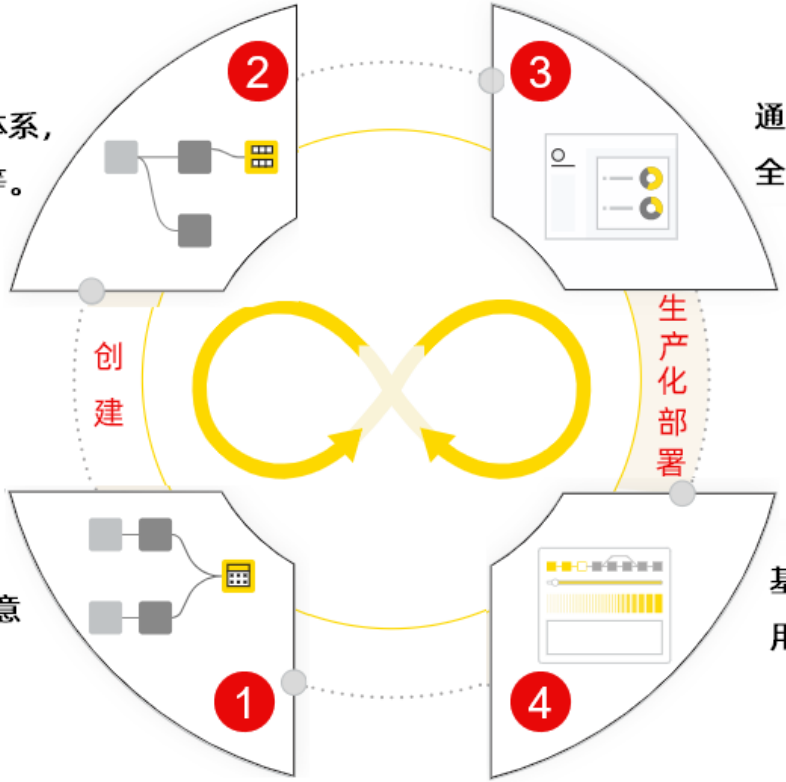
KNIME数据科学分析流程

建模与可视化

支持完整的数据分析与建模技术体系，如统计分析、机器学习、可视化等。

部署与监控

通过标准化流程，将数据科学解决方案安全部署到生产环境，并持续监控及管理。



数据整合与转换

通过 300 多种数据连接器，从任意数据源接入各类数据。

应用与交互

基于云原生架构，实现企业级规模化应用，如报表与API调用，交互式分析。



02

数据结构和数据来源

数据结构

结构化数据

常见的行列结构的数据表、Excel表、数据库表等

半结构化数据

如网页数据，通过Xpath解析，也可以获取标准化的数据表

非结构化数据

如文本、图片、Word文档、PDF文档等

数据来源

数据获取方式

数据的获取可以通过多种途径，包括使用API接口、数据库连接以及直接从文件中读取。不同的数据源可能需要采用不同的方法，以确保获取的数据具有完整性和准确性。



数据源分类

数据源可以根据其性质进行分类，如开放型数据库、专业行业数据、社交媒体数据等。每种类型的数据源都有其特定的优势和局限性，需要根据研究目标进行选择。

数据质量评估

在选择数据源时，评估数据的质量和可靠性至关重要。需要考虑的因素包括数据的更新频率、样本的代表性以及错误率等，以确保数据分析的准确性和有效性。

数据管理

在使用公共数据源时，需要遵守相关的访问协议和规定。使用私有数据时，注意数据脱敏及加密处理



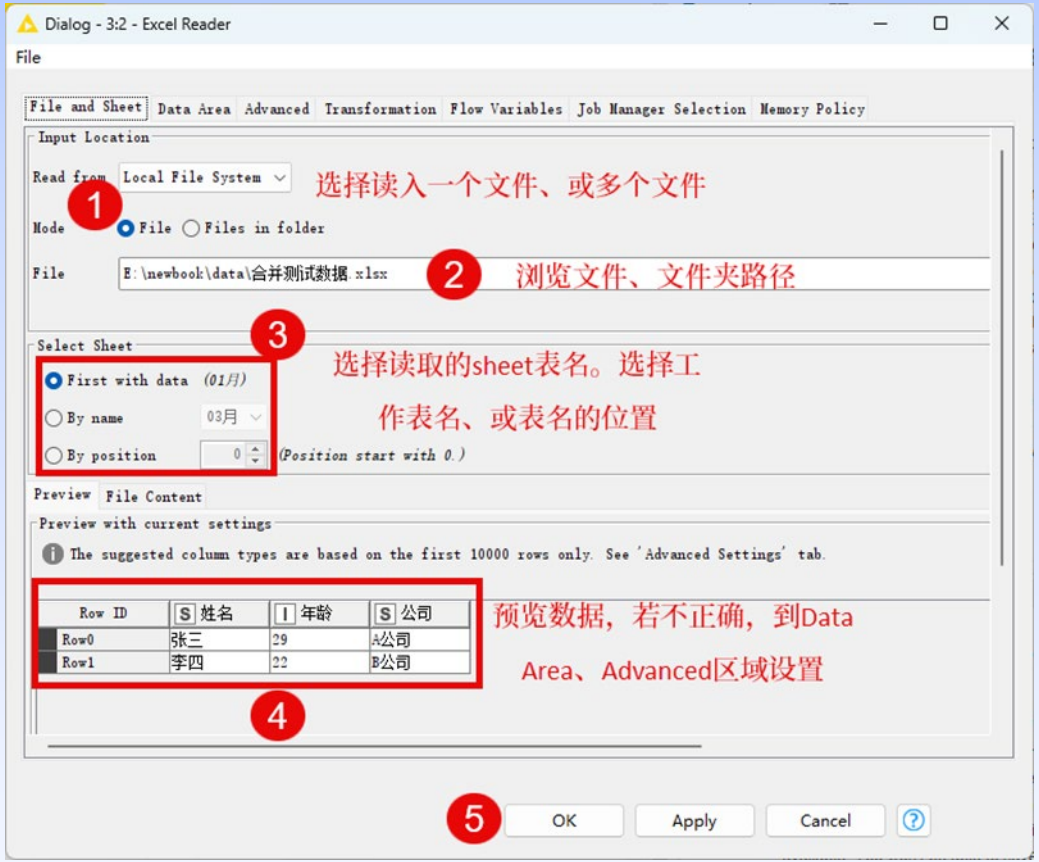
03

常用数据读入节点

Excel Reader 读取单个文件

Excel Reader概述

Excel Reader是KNIME中用于读取Excel文件的节点，支持多种数据格式和选项，包括读取特定工作表、设置标题行等高级功能。



Excel Reader 读取多个文件

1、读入多个文件的同一个工作表名（基于相同的工作表名）



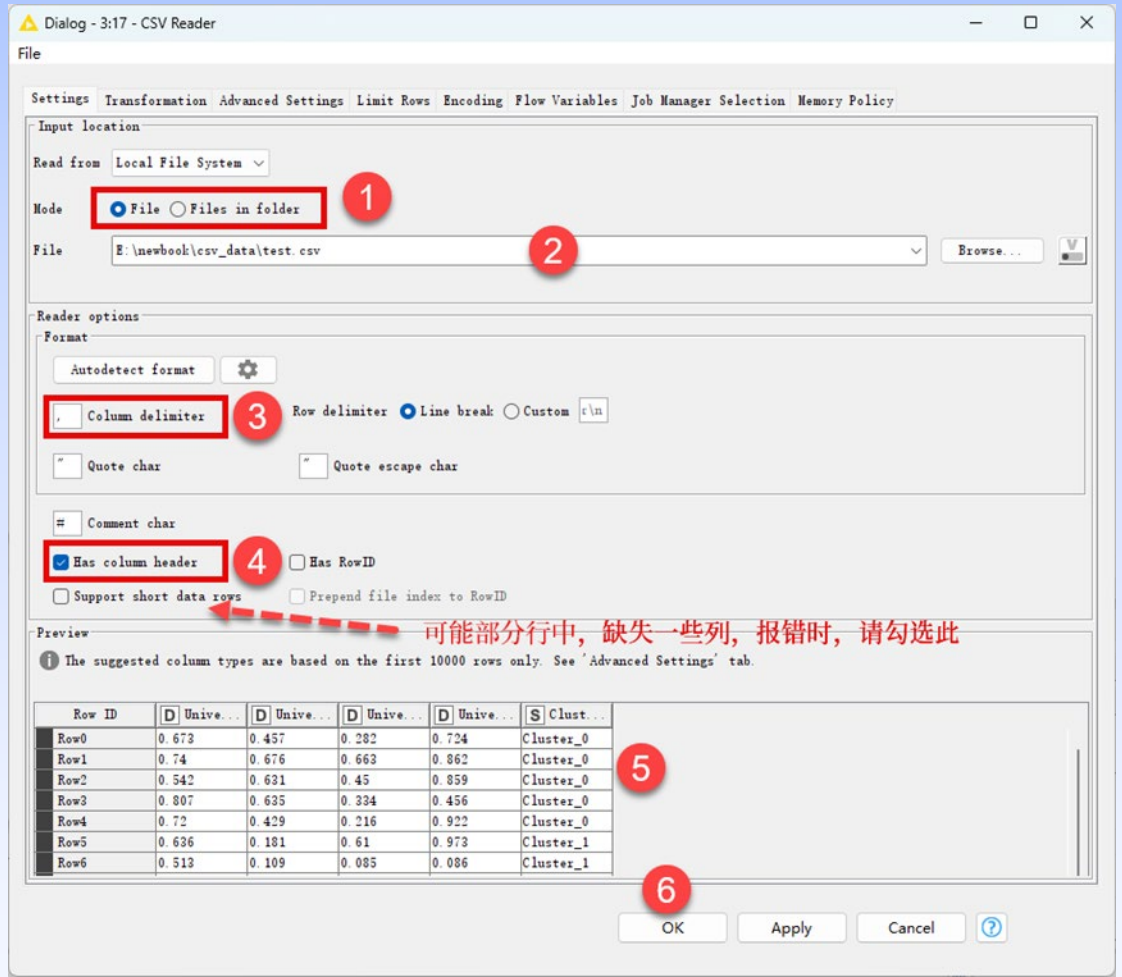
3、单个Excel文件N个sheet合并

2、多个Excel文件的合并（基于相同的索引位置）

4、多个Excel文件N个sheet合并

详见3.3.1各个示例，复杂场景，利用循环更方便

CSV Reader


Dialog - 3:17 - CSV Reader
File
Settings
Transformation
Advanced Settings
Limit Rows
Encoding
Flow Variables
Job Manager Selection
Memory Policy
Input location
Read from
Local File System
Mode
File
Files in folder
1
File
E:\newbook\csv_data\test.csv
2
Browse...
Reader options
Format
Autodetect format
,
Column delimiter
3
Row delimiter
Line break
Custom
\r\n
"
Quote char
"
Quote escape char
#
Comment char
Has column header
4
Has RowID
Support short data rows
Prepend file index to RowID
可能部分行中，缺失一些列，报错时，请勾选此
Preview
The suggested column types are based on the first 10000 rows only. See 'Advanced Settings' tab.
Row ID
D Unive...
D Unive...
D Unive...
D Unive...
S Clust...
Row0 0.673 0.457 0.282 0.724 Cluster_0
Row1 0.74 0.676 0.663 0.862 Cluster_0
Row2 0.542 0.631 0.45 0.859 Cluster_0
Row3 0.807 0.635 0.334 0.456 Cluster_0
Row4 0.72 0.429 0.216 0.922 Cluster_0
Row5 0.636 0.181 0.61 0.973 Cluster_1
Row6 0.513 0.109 0.085 0.086 Cluster_1
5
6
OK
Apply
Cancel

File Reader

1、File Reader比较常用的读取文本数据文件的节点，可以处理csv，txt等文本文件。

2、File Reader（complex Format），用于读取复杂的文件格式。当File Reader不能很好处理文本文件时再考虑此节点。

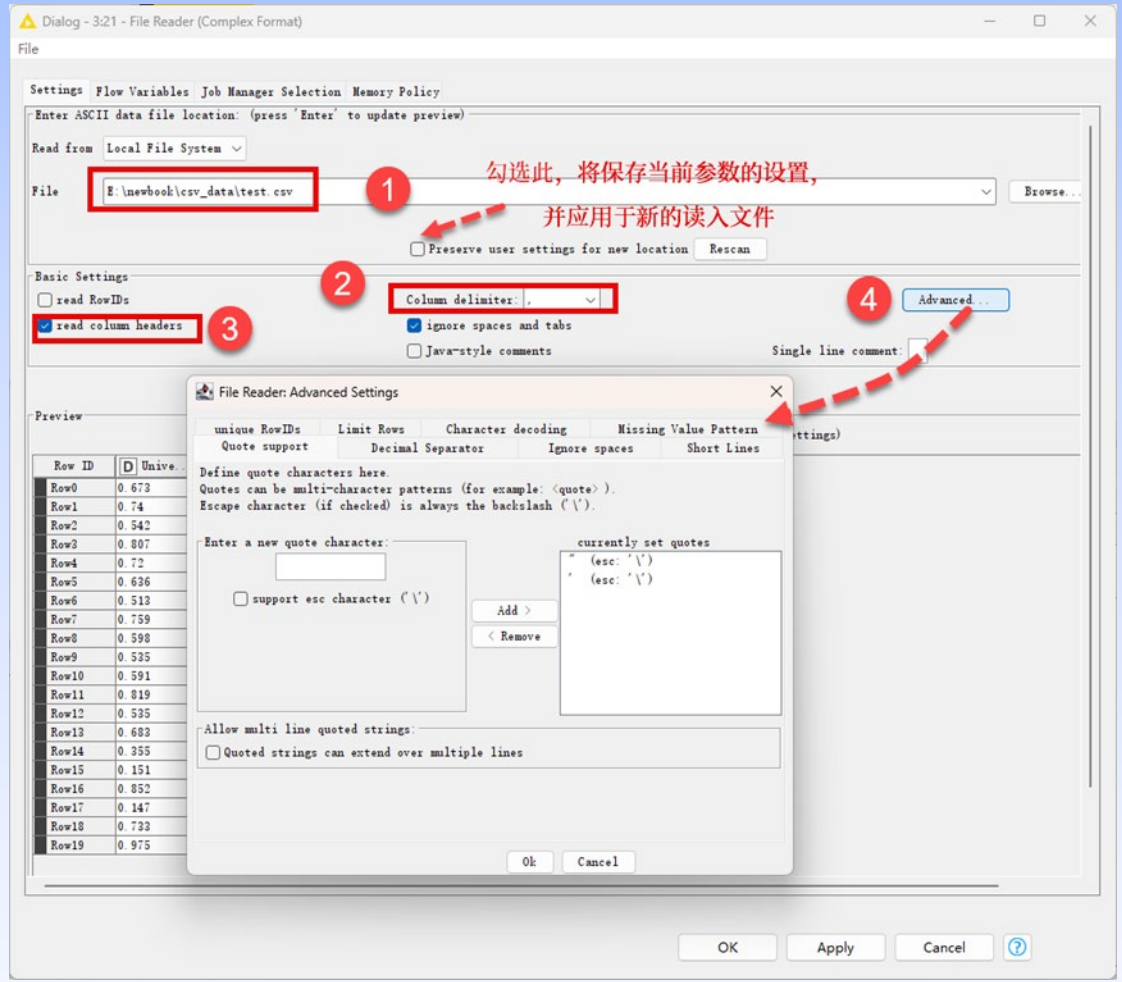


Table Reader

Table Reader概述

只能读入Table Writer输出的数据文件，文件后缀为table。该节点配置选项与CSV Reader类似

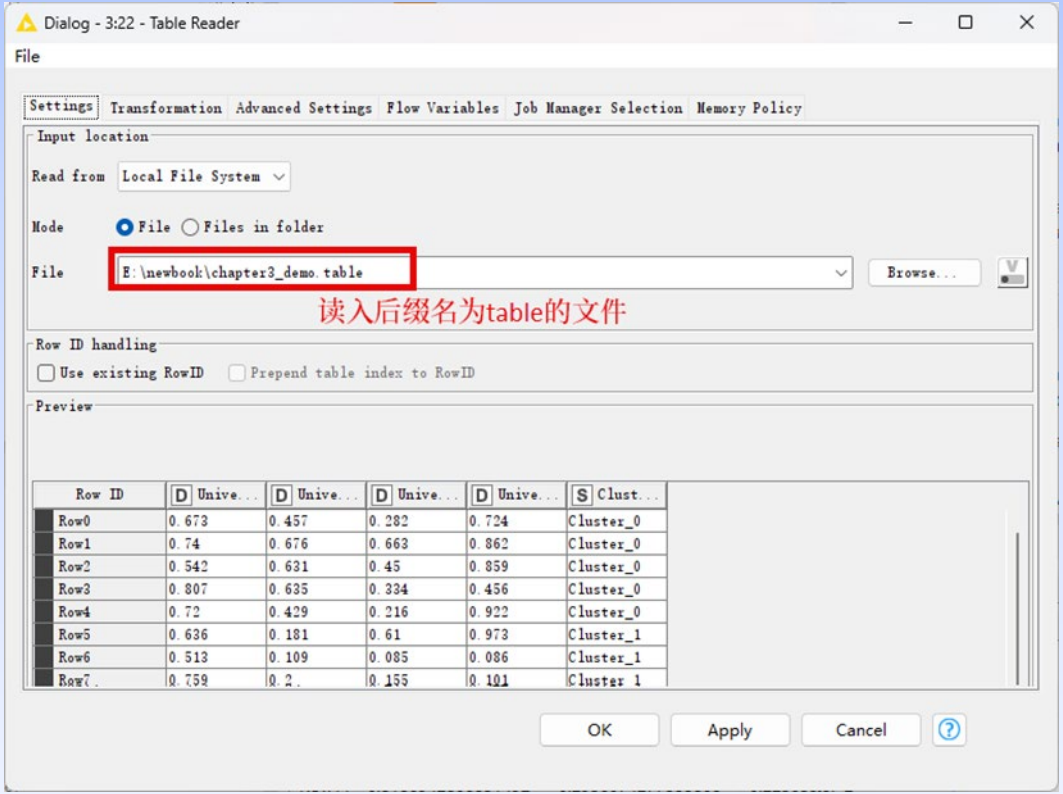
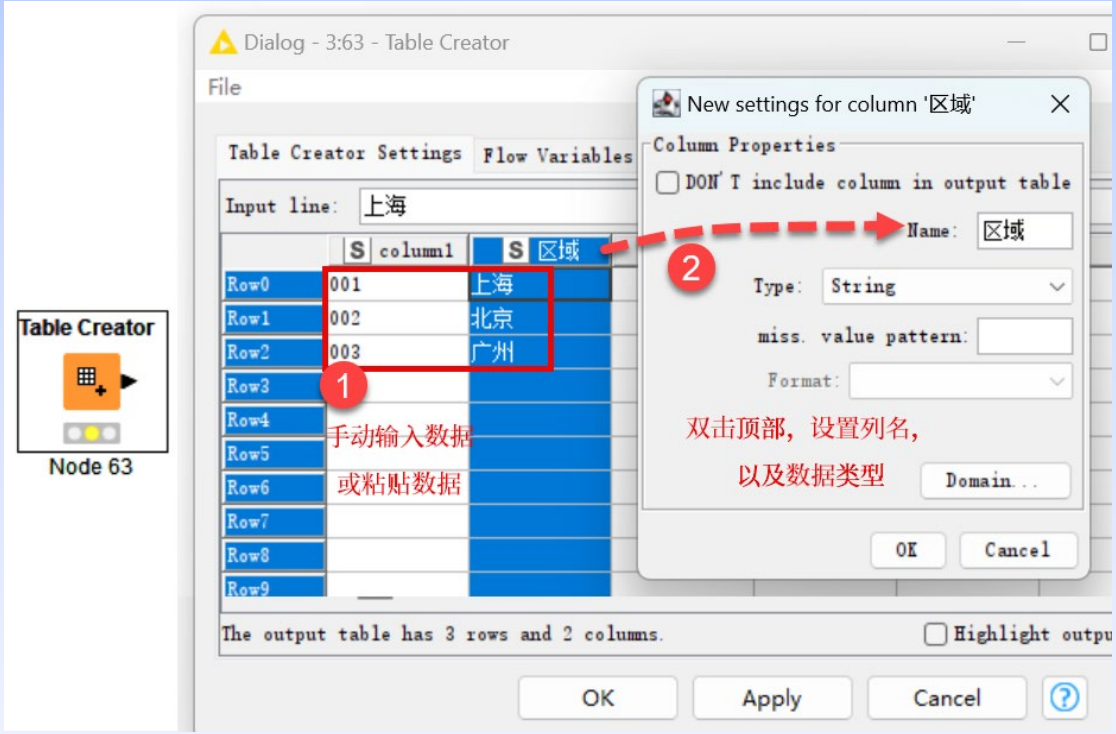


Table Creator

Table Creator概述

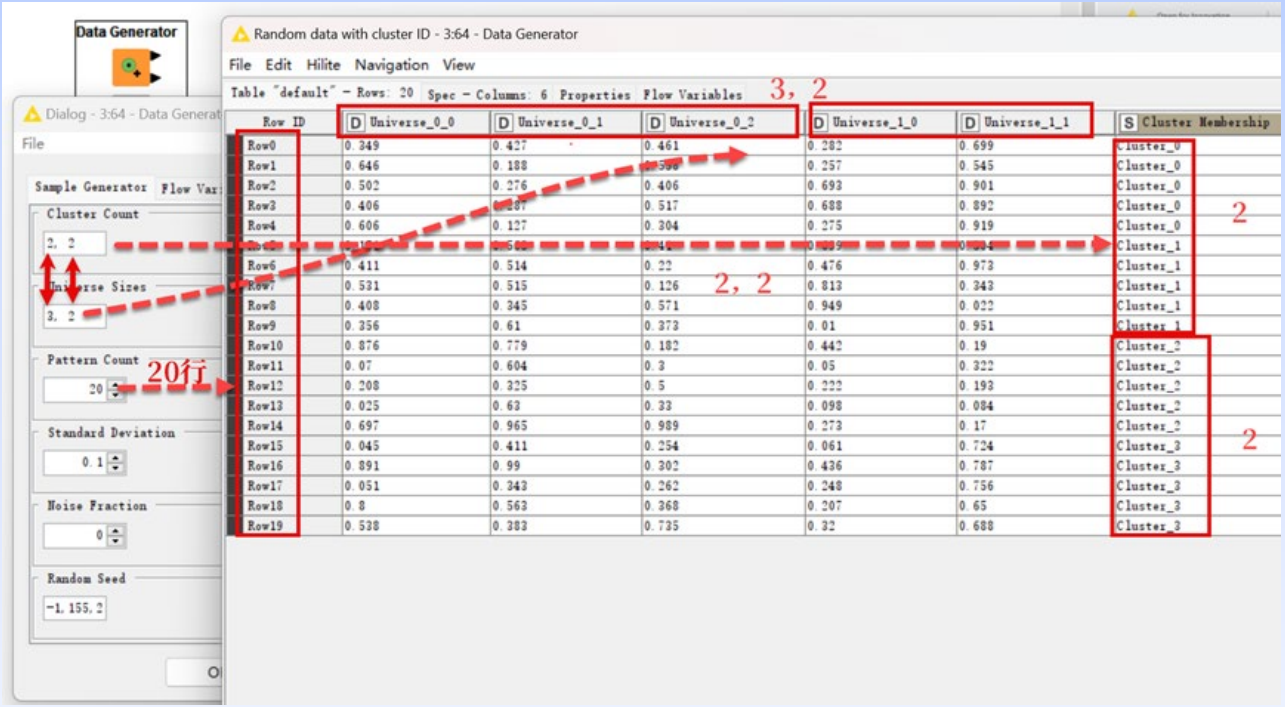
此节点主要是手动输入数据，或者将其他复制的数据粘贴至表格中。一般数据量较小，或者可能作为一些参数时使用，好处是方便调试，可以随时修改数据，且无需考虑文件路径问题。



Data Generator

数据生成器概述

Data Generator是一种工具，用于生成大量数据以支持数据分析和模型训练。它通过定义数据生成的参数，如样本大小、数据类型等，生成符合需求的数据。

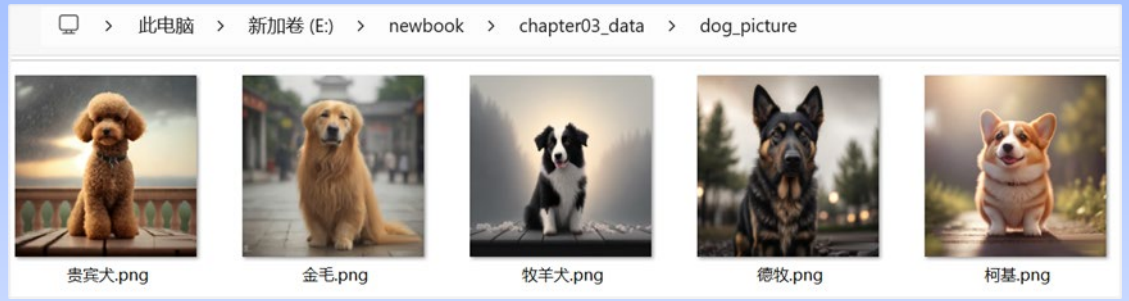


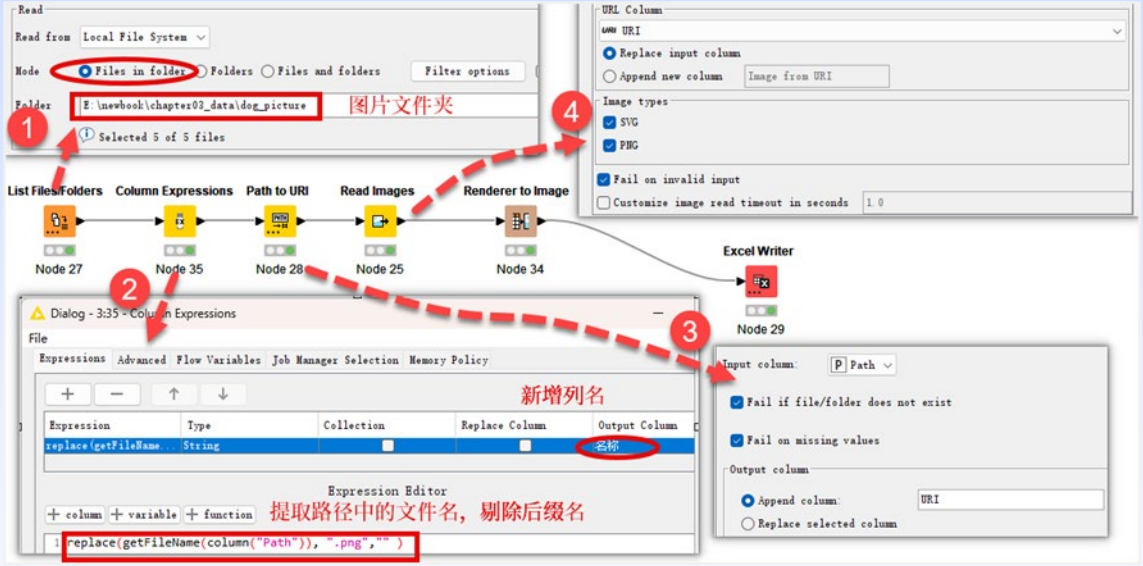
Row ID	Universe_0_0	Universe_0_1	Universe_0_2	Universe_1_0	Universe_1_1	Cluster Membership
Row0	0.349	0.427	0.461	0.282	0.699	Cluster_0
Row1	0.646	0.188	0.596	0.257	0.545	Cluster_0
Row2	0.502	0.276	0.406	0.693	0.901	Cluster_0
Row3	0.406	0.287	0.517	0.688	0.892	Cluster_0
Row4	0.606	0.127	0.304	0.275	0.919	Cluster_0
Row5	0.196	0.502	0.41	0.009	0.004	Cluster_1
Row6	0.411	0.514	0.22	0.476	0.973	Cluster_1
Row7	0.531	0.515	0.126	0.813	0.343	Cluster_1
Row8	0.408	0.345	0.571	0.949	0.022	Cluster_1
Row9	0.356	0.61	0.373	0.01	0.951	Cluster_1
Row10	0.876	0.779	0.182	0.442	0.19	Cluster_2
Row11	0.07	0.604	0.3	0.05	0.322	Cluster_2
Row12	0.208	0.325	0.5	0.222	0.193	Cluster_2
Row13	0.025	0.63	0.33	0.098	0.084	Cluster_2
Row14	0.697	0.965	0.989	0.273	0.17	Cluster_2
Row15	0.045	0.411	0.254	0.061	0.724	Cluster_3
Row16	0.891	0.99	0.302	0.436	0.787	Cluster_3
Row17	0.051	0.343	0.262	0.248	0.756	Cluster_3
Row18	0.8	0.563	0.368	0.207	0.65	Cluster_3
Row19	0.538	0.383	0.735	0.32	0.688	Cluster_3

Read Images

Read Images概述

此节点批量读入本地后缀名为png或SVG的图片文件，或者读入KNIME中View类下各节点生成的矢量图SVG文件。





A	B	C
Path	名称	URI
E:\newbook\chapter03_data\dog_picture\德牧.png	德牧	
E:\newbook\chapter03_data\dog_picture\柯基.png	柯基	
E:\newbook\chapter03_data\dog_picture\牧羊犬.png	牧羊犬	
E:\newbook\chapter03_data\dog_picture\贵宾犬.png	贵宾犬	
E:\newbook\chapter03_data\dog_picture\金毛.png	金毛	

读入常用数据文件便捷操作

Excel Reader

CSV Reader

File Reader

Table Reader



针对个别类型数据文件支持直接拖入，KNIME将自动选择节点，并设置文件路径，跳出该节点配置窗口



04

常用数据写入类节点

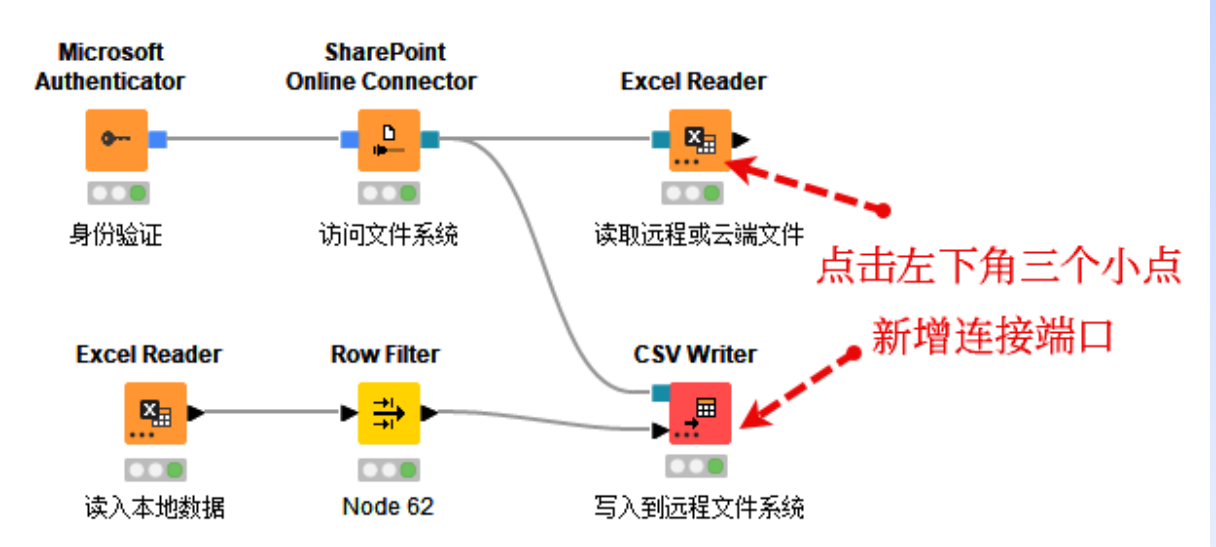
常用写入类节点

编号	节点名称	节点功能	对应节点
1	Excel Writer	将数据写入Excel文件，默认第一个sheet表名为default，可手动指定sheet名。既有工作簿、工作表都存在追加和覆盖写入模式。	Excel Reader
2	CSV Writer	写入CSV文件，指定列名分隔符，设置是否包括表头、文件编码格式等。文件后缀可为csv、txt等常用文本后缀名。	CSV Reader
3			File Reader
4			File Reader（Complex Format）
5	Table Writer	KNIME专用的数据文件，文件以table为后缀。KNIME用户之间使用此格式共享数据时，读入的数据类型将保持一致	Table Reader
6	Images Writer（table）	可将指定列中的图片，逐个保存图片至指定路径中	Read Images
7	PMML Writer	训练建模后得到的PMML Writer节点导出的模型文件，利用此模型，对新数据进行预测。该模型格式较为通用，如Java、Python均可读入该模型文件	PMML Reader
8	Model Writer	类似PMML文件，也是经过训练后的Model Writer节点导出的模型文件，该文件为KNIME平台中使用的模型文件，不支持其他平台或程序读取，只能使用Model Reader节点读取该节点导出的模型文件。	Model Reader

远程文件读取与写入

远程文件读取概述

远程文件读取通常涉及通过网络协议访问位于不同计算机上的数据文件。这种方法在大数据处理和分布式系统中尤为常见，允许用户从远程服务器或云端存储中获取所需数据。





05

文件路径

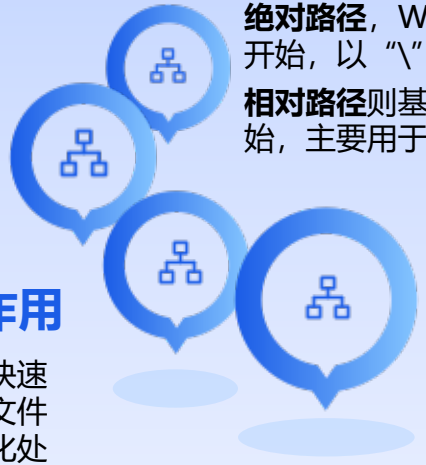
文件路径简介

文件路径定义

文件路径是指示文件在计算机系统中存放位置的字符串，用于帮助用户在磁盘上找到特定文件。它是文件系统结构的重要组成部分，允许用户通过路径名访问文件。

文件路径作用

文件路径的主要作用是确保用户能够准确、快速地访问所需文件。它不仅帮助用户在复杂的文件系统中导航，还支持文件的批量操作和自动化处理，提高数据管理效率。



绝对路径与相对路径

文件路径主要分为**绝对路径和相对路径。**

绝对路径，Windows系统以盘符开始、Linux或Mac从根文件夹开始，以“\”作为分隔符(如/home/demo.xlsx)。

相对路径则基于当前工作目录，路径开始处不以盘符或“/”开始，主要用于在同一文件夹内的文件定位。

相对路径的优点

方便工作流的跨平台、跨设备移植，也方便不同用户对工作流的执行，避免因本地路径找不到而报错

绝对路径与相对路径

01

绝对路径定义

绝对路径是从文件系统的根目录开始的完整路径，可以唯一地标识一个文件或目录。使用绝对路径时，不需要考虑当前工作目录的位置，因为路径已经包含了所有必要的信息。

02

相对路径定义

相对路径是以当前文件所在目录为参照基础，链接到目标文件或文件夹的路径。相对路径通过目录层次结构中的相对位置来定位文件，适用于需要动态更新路径的场景。

03

绝对路径优缺点

绝对路径的优点包括路径固定、不会随环境变化而改变，便于备份和共享。然而，其缺点是不够灵活，不适用于频繁移动或重命名文件的情况，且在多用户环境中管理起来较为复杂。

04

相对路径优缺点

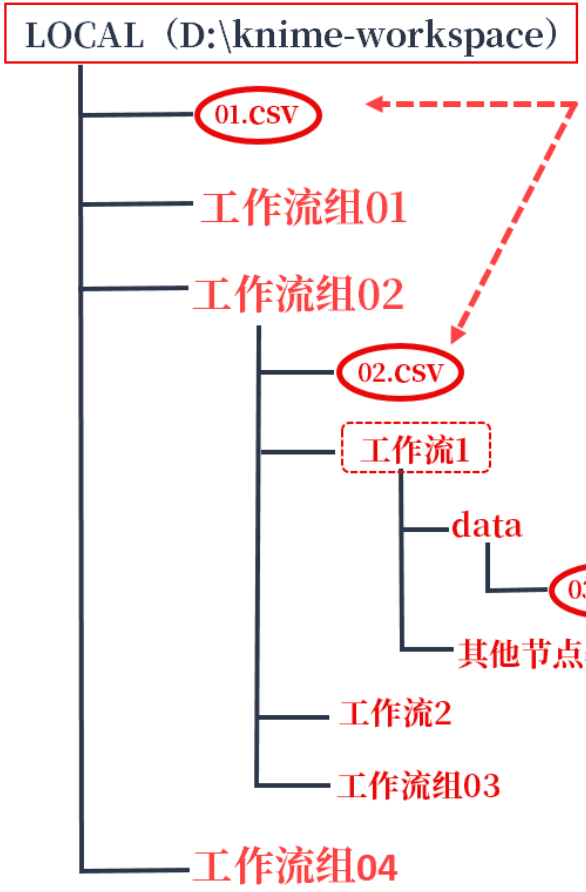
相对路径的优点是灵活性高，不受文件系统结构变化的影响，特别是在文件移动后仍能正确定位。但缺点是不够直观，容易发生错误，特别是在大型项目中，路径关系可能变得复杂。

05

绝对路径与相对路径实际应用

在实际开若工作流仅本人使用，两者路径方式都可以。若工作流引用了本地数据文件，建议使用相对路径将数据内嵌值工作流中

相对路径与数据文件位置关系



1

Relative to → Current Mountpoint

当前挂载点，文件初始浏览位置为工作流空间的根目录

2

Relative to → Current workflow

假定当前工作流为“工作流1”，此时，可浏览与工作流1所在同级工作流组（包括子工作流组）、或者向上浏览至LOCAL处。总之，只能选择工作流组目录下文件

3

Relative to → Current workflow data area

假定当前工作流为“工作流1”，此时，只能浏览该工作流下data文件夹中的“03.csv”



06

本章小结

回顾重要概念和实践技巧

数据分析基本概念

数据分析是通过适当的统计分析方法对大量数据进行处理、汇总、理解和消化的过程，旨在最大化地开发数据的功能和作用。它涉及数据收集、整理、解释和可视化，是现代商业决策和科研活动的重要支撑。

数据分析技术

数据分析技术包括描述性统计、回归分析、分类和聚类等。这些技术帮助从数据中提取有价值的信息，揭示变量之间的关系和模式，支持决策制定和业务优化。

数据可视化与沟通

数据可视化通过图表、仪表盘等形式展示分析结果，使复杂的数据变得直观易懂。有效的数据可视化技巧可以帮助更好地向管理层和利益相关者传达分析结论，提升决策效率。

数据收集与准备

数据收集是数据分析的第一步，通过设计问卷、实验或从外部数据库获取数据。数据准备包括数据清洗、缺失值处理和数据转换，确保数据质量和一致性，为后续分析打下坚实基础。

实践技巧与工具使用

掌握数据分析需要学习和使用多种工具和编程语言，如Python、R和SQL。同时，建立高效的数据流程和团队协作机制，以及应用高级分析技术和模型评估方法，也是成功完成数据分析项目的关键。

提供进一步学习建议和资源

推荐学习资源

- KNIME官网
- KNIME在线L1-L4课程
- KNIME论坛等
- 《KNIME视觉化数据分析》雒玉玺等著

实践项目与案例分析

通过参与实际项目和分析真实世界的案例，可以加深对数据分析方法的理解。这些实践经验有助于将理论知识应用于解决实际问题。



参与到专业社区

加入数据分析的专业社区，与同行交流经验、分享技巧并获取最新的行业动态。

持续更新与跟进

数据分析领域不断发展，新的工具和技术不断涌现。定期关注相关博客、新闻和研讨会，保持知识更新，确保掌握最新的行业趋势。

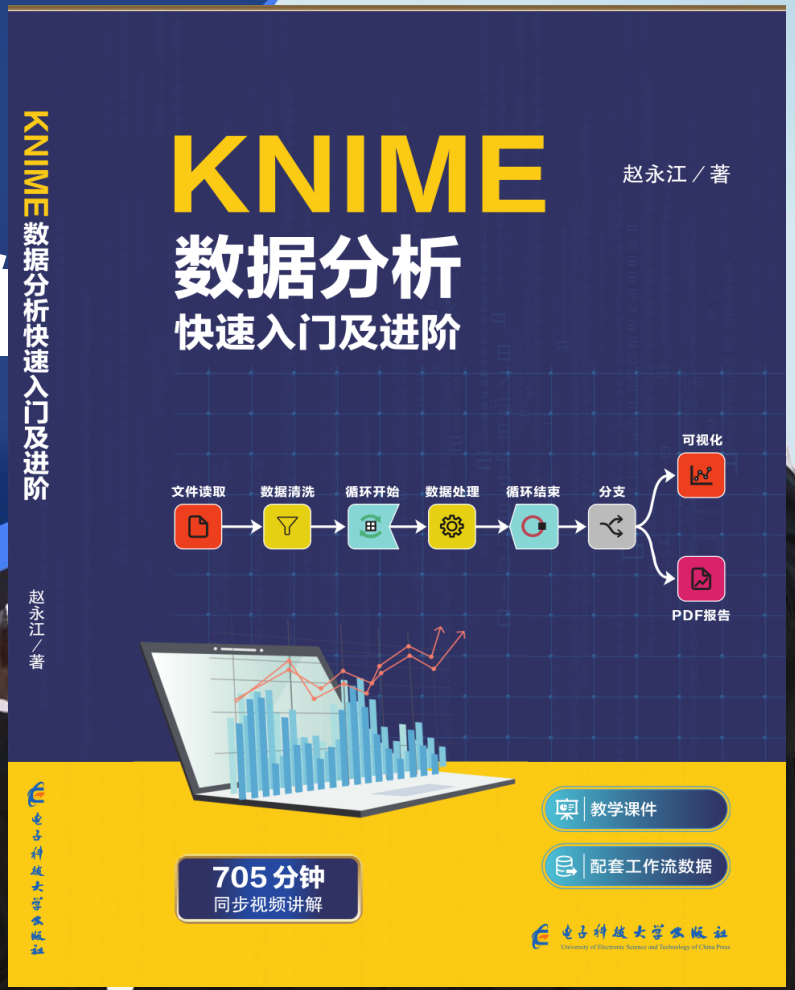
感谢观看

THANKS



第4章 基础数据操作

掌握数据处理基础操作



目录

01 过滤

02 排序

03 列重命名

04 数据类型转换

05 常见数据类型操作

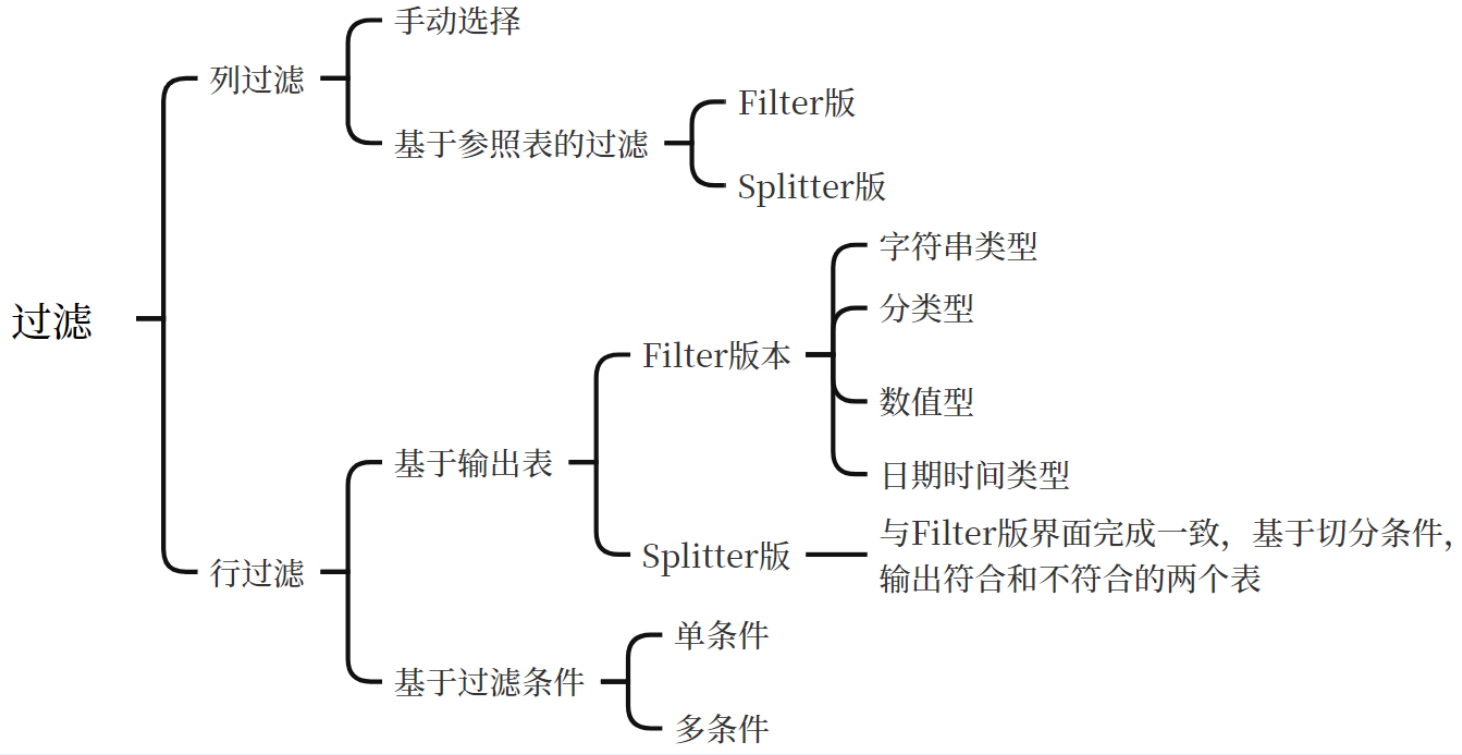
06 本章小结



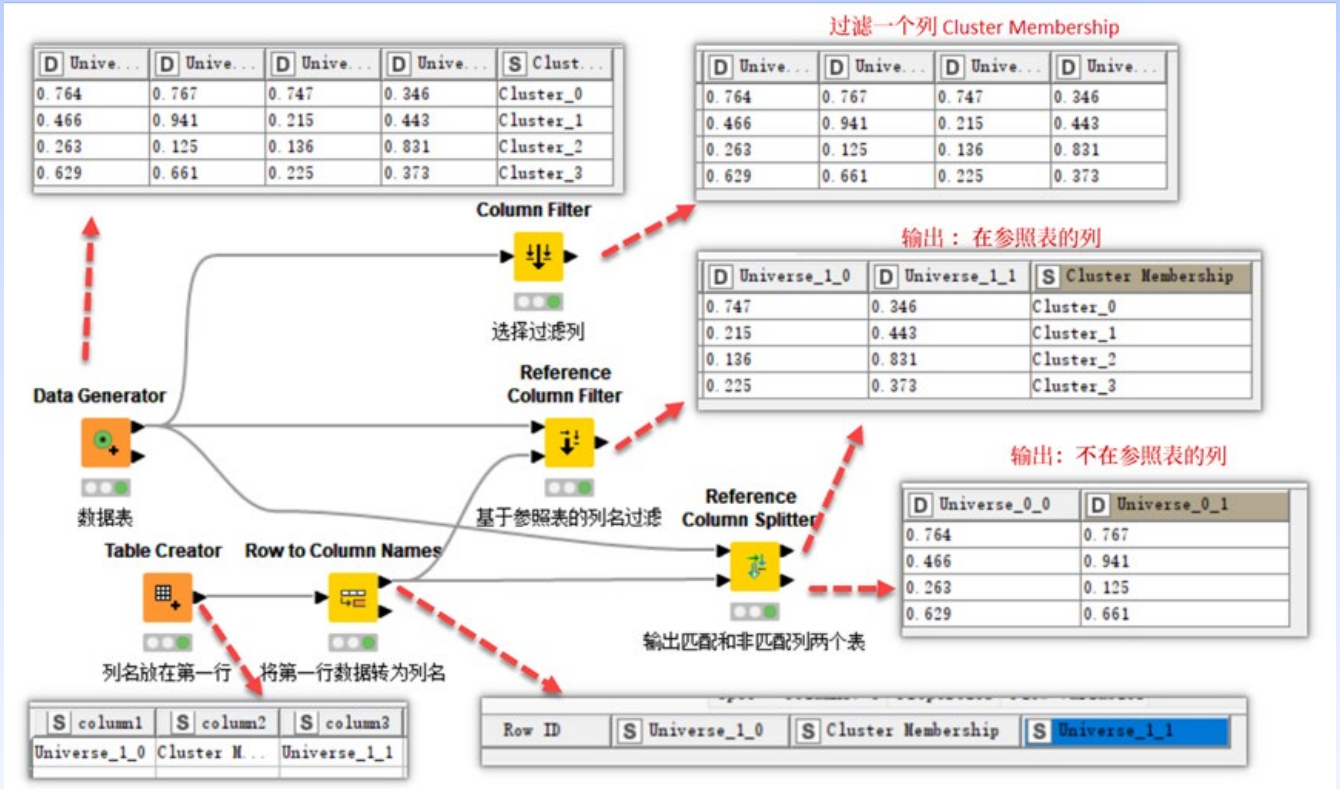
01

过滤

过滤类型概览


过滤
列过滤
手动选择
基于参照表的过滤
Filter版
Splitter版
行过滤
基于输出表
Filter版本
字符串类型
分类型
数值型
日期时间类型
Splitter版
与Filter版界面完成一致，基于切分条件，输出符合和不符合的两个表
基于过滤条件
单条件
多条件

列过滤



节点窗口内选择列名四种方式

1. 手动选择，
2. 基于关键字和通配符，
3. 正则匹配，
4. 基于数据类型

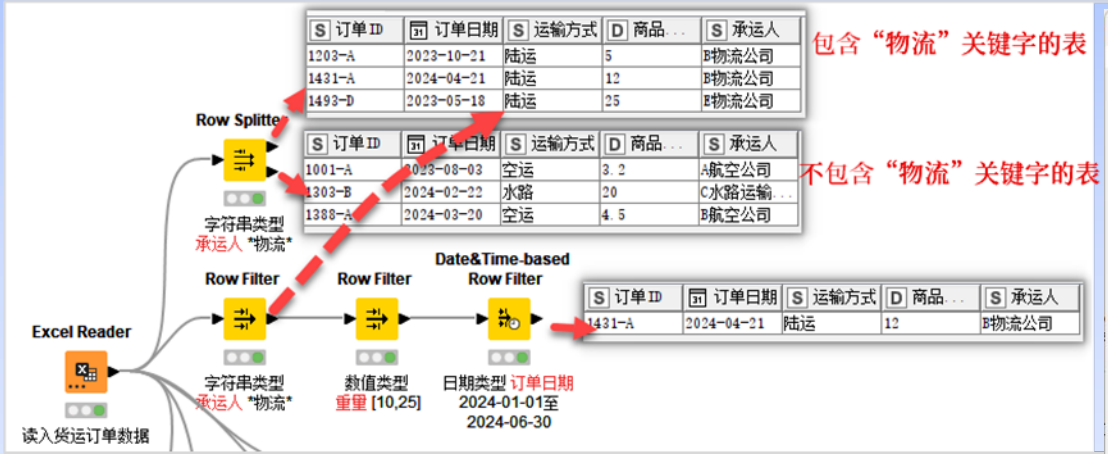
行过滤概览

	Filter版	Splitter版	说明
通用型	Row Filter	Row Splitter Numeric Row Splitter	基于数据类
分类型	Nominal Value Row Filter Reference Row Filter	Nominal Value Row Splitter Reference Row Splitter	
日期时间型	Date&Time-based Row Filter		
多条件逻辑判断	Rule-based Row Filter Rule-based Row Filter (Dictionary)	Rule-based Row Splitter Rule-based Row Splitter (Dictionary)	基于条件判断



5.4版本新增节点，
灵活便捷，功能强大

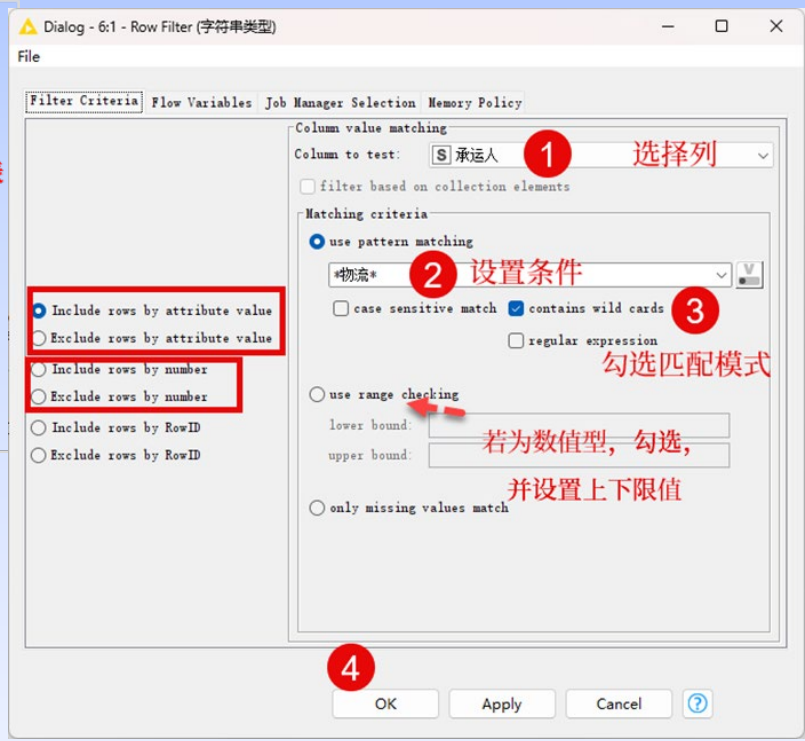
行过滤



订单ID	订单日期	运输方式	商品	承运人
1203-A	2023-10-21	陆运	5	B物流公司
1431-A	2024-04-21	陆运	12	B物流公司
1493-D	2023-05-18	陆运	25	B物流公司

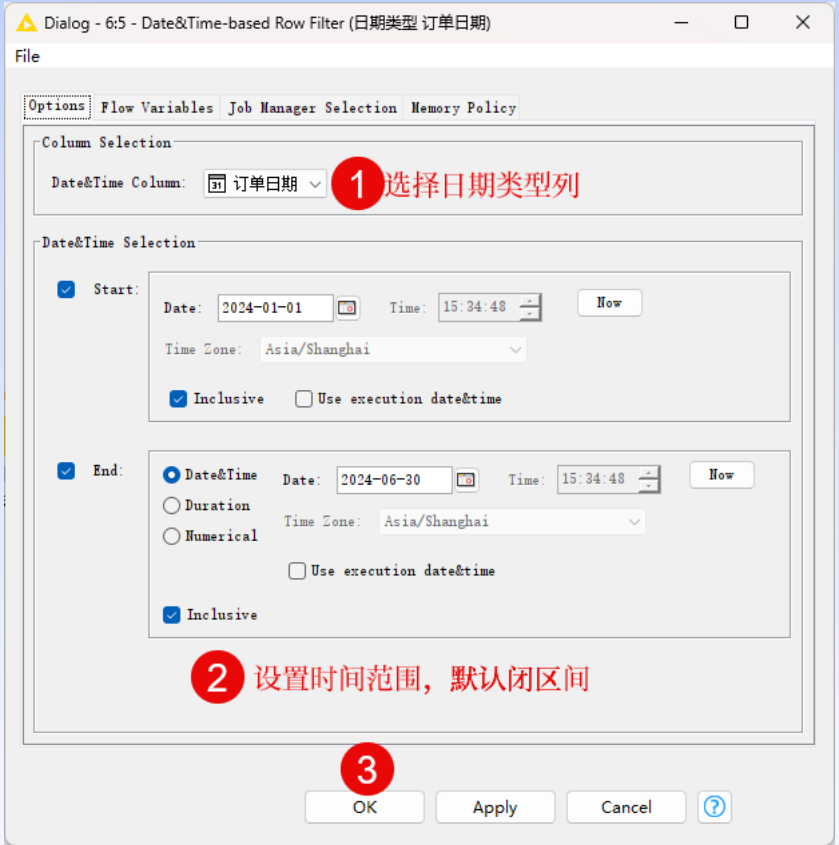
订单ID	订单日期	运输方式	商品	承运人
1001-A	2023-08-03	空运	3.2	A航空公司
1303-B	2024-02-22	水路	20	C水路运输
1388-A	2024-03-20	空运	4.5	B航空公司

订单ID	订单日期	运输方式	商品	承运人
1431-A	2024-04-21	陆运	12	B物流公司

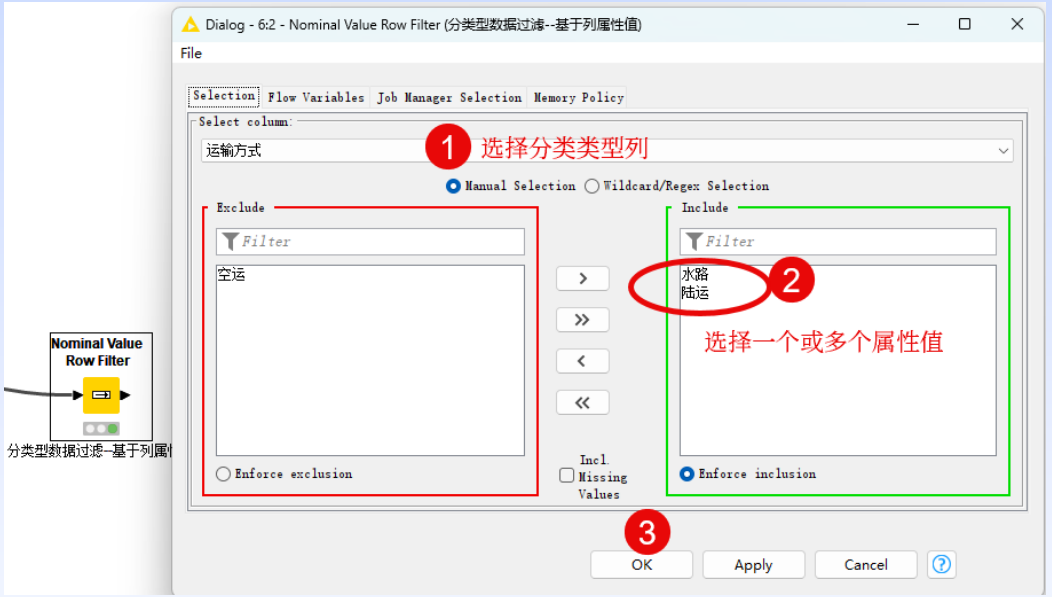


行过滤

日期类型

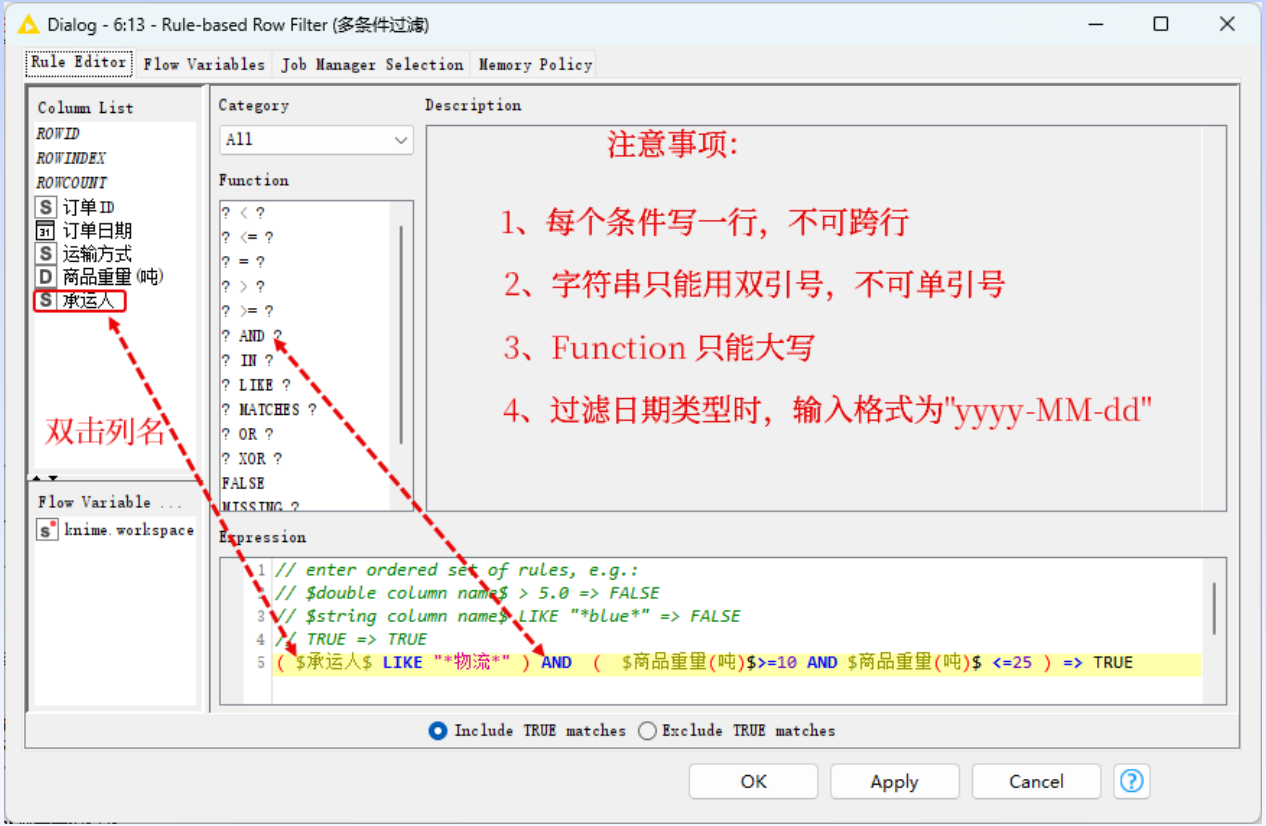


分类型（字符串）



行过滤

单条件 或 多条件 判断



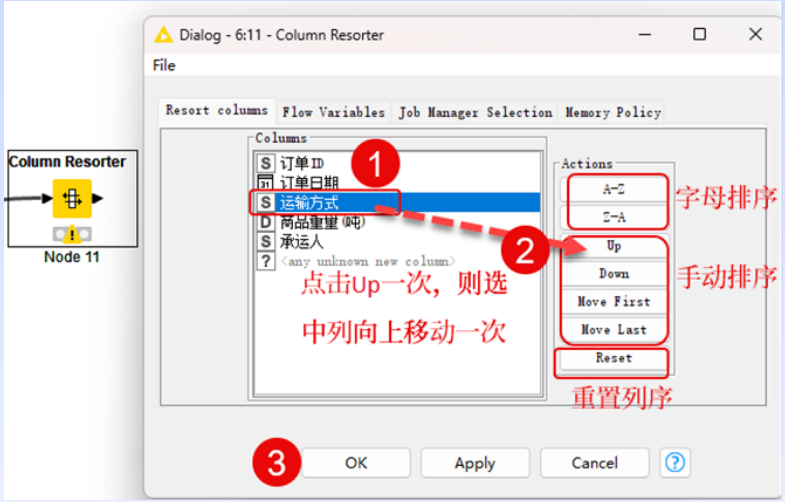


02

排序

列排序

01 手动排序



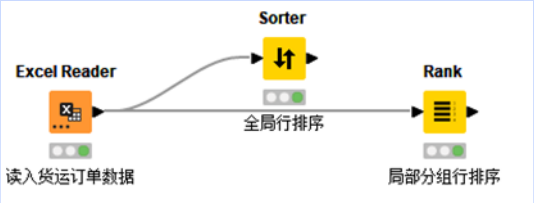
02 基于参照表自动排序



行排序

01

Sorter全局排序





行排序

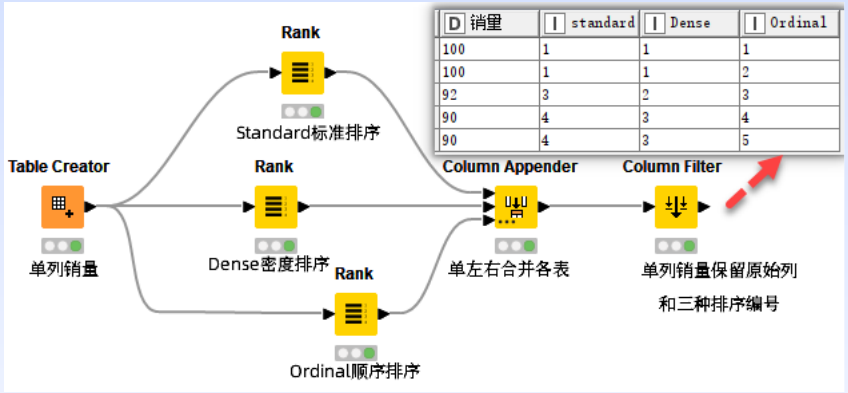
02

Rank 局部分组排序

（1）Standard： 标准排序，相同值同序，序号不连续。

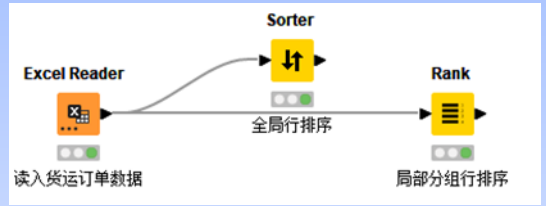
（2）Dense：密度排序，相同值同序，序号连续，不会跳序。

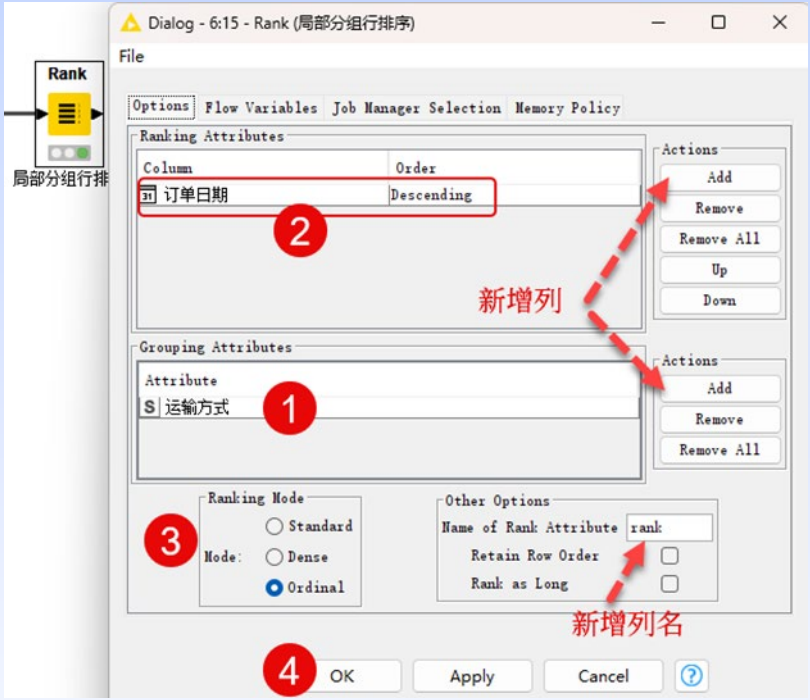
（3）Ordinal：顺序排序，不考虑值是否相同，以排序的绝对顺序自然数编号，序号唯一。



销量	standard	Dense	Ordinal
100	1	1	1
100	1	1	2
92	3	2	3
90	4	3	4
90	4	3	5

示例 1





示例 2

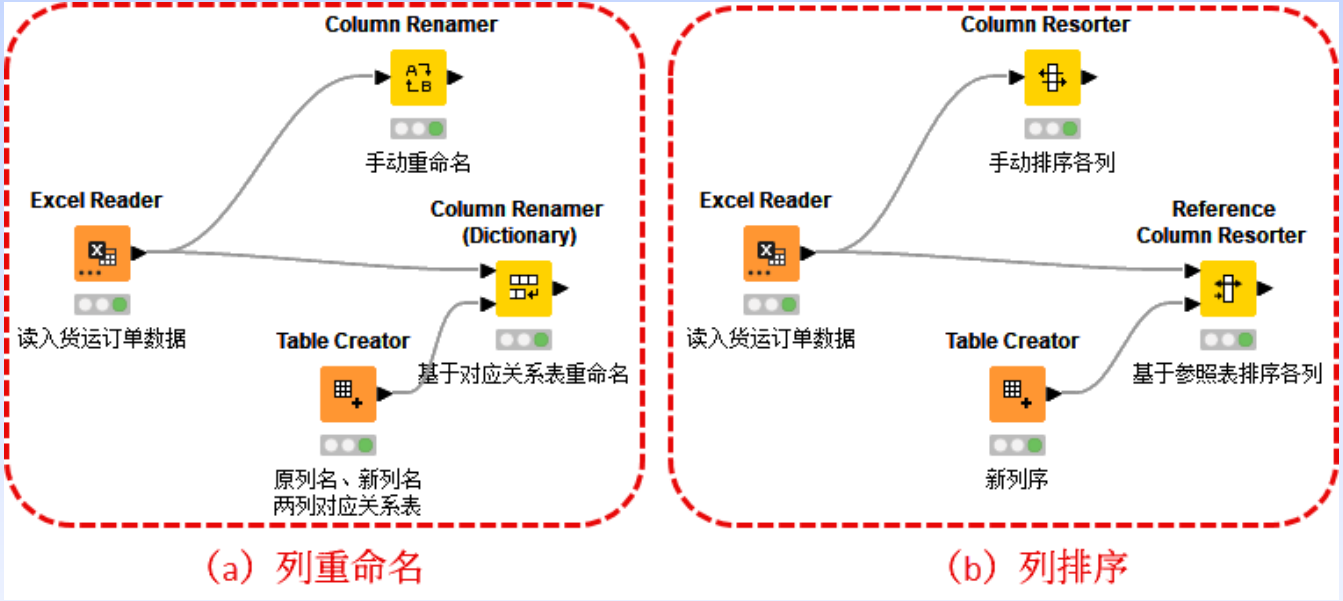


03

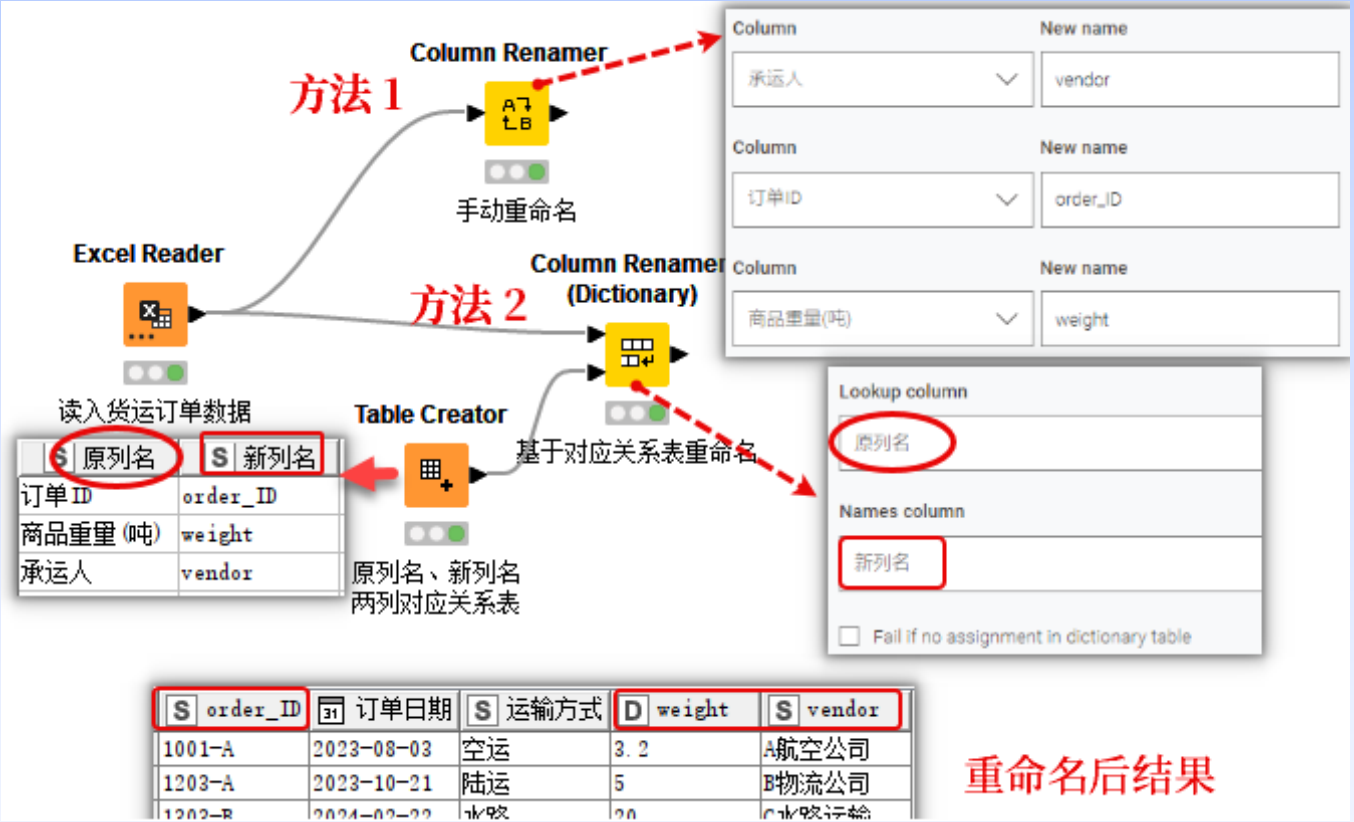
列重命名

列重命名示例

注意对比学习



列重命名的两种方法


Column Renamer
方法 1
手动重命名
Excel Reader
Column Renamer (Dictionary)
方法 2
读入货运订单数据
Table Creator
基于对应关系表重命名
原列名
新列名
订单 ID
order_ID
商品重量 (吨)
weight
承运人
vendor
原列名、新列名
两列对应关系表
Column
New name
承运人
vendor
Column
New name
订单ID
order_ID
Column
New name
商品重量(吨)
weight
Lookup column
原列名
Names column
新列名
Fail if no assignment in dictionary table
order_ID
订单日期
运输方式
weight
vendor
1001-A
2023-08-03
空运
3.2
A航空公司
1203-A
2023-10-21
陆运
5
B物流公司
重命名后结果



04

数据类型转换

常用数据类型总结

01 数字类型

数字类型包括整数和浮点数。整数用于表示没有小数部分的数值，如-5、0、10等。浮点数则用于表示带小数部分的数值，如2.3、-0.5等，常用于科学计算和财务计算中。

02 字符串类型

字符串类型用于存储文本数据，由一系列字符组成。常见操作包括字符串的拼接、分割、替换和长度计算。

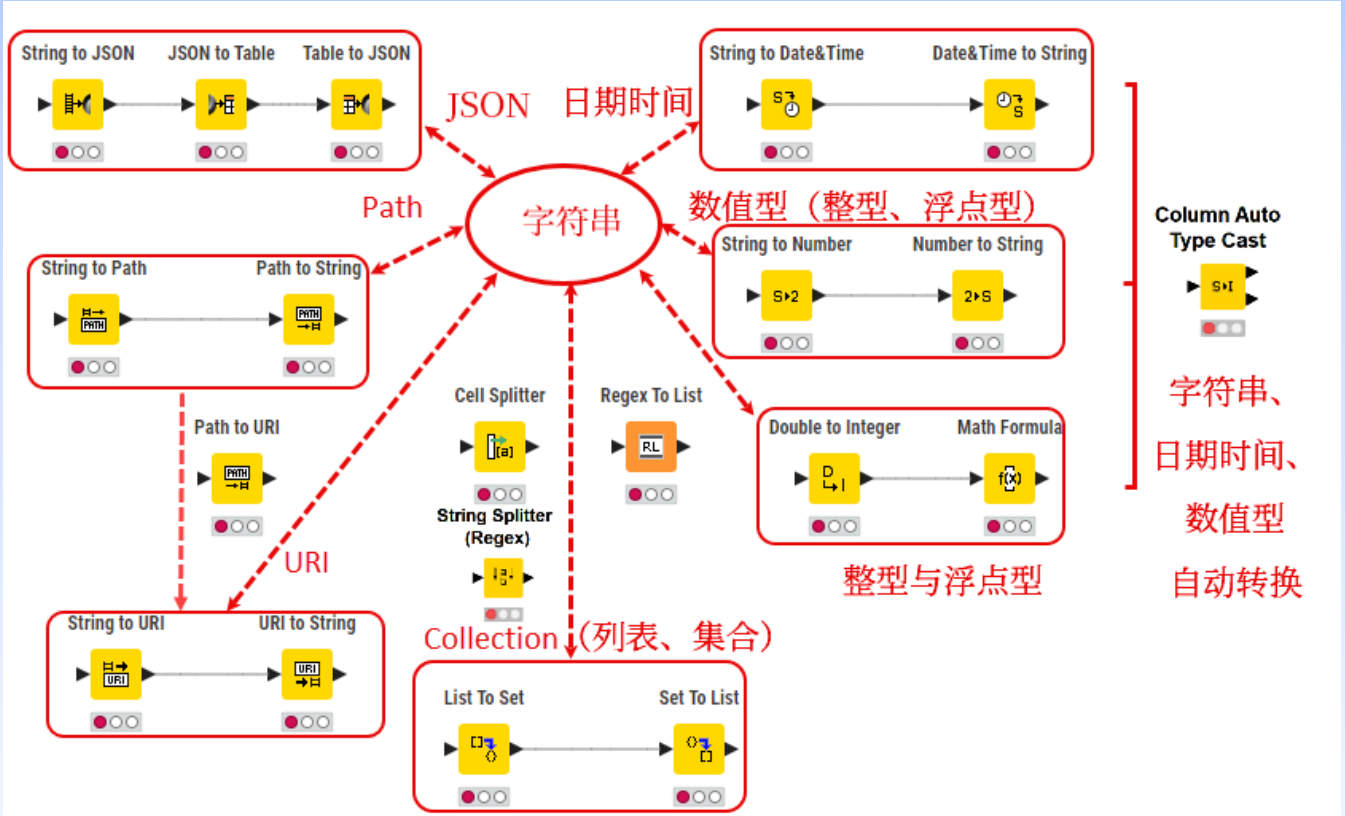
03 日期时间类型

日期时间类型用于表示时间和日期，包括年、月、日、时、分、秒等信息。常见操作包括日期的加减、时间差的计算及格式化输出。

04 列表与集合类型

列表是有序的、可以重复的一组数据，常用操作包括添加、删除、查找和排序元素。集合则是无序的、唯一元素的集合，常用操作有添加、删除元素及判断元素是否存在。

常用数据类型转换节点


String to JSON
JSON to Table
Table to JSON
JSON
日期时间
String to Date&Time
Date&Time to String
Path
字符串
数值型（整型、浮点型）
String to Path
Path to String
String to Number
Number to String
Column Auto Type Cast
Path to URI
Cell Splitter
Regex To List
Double to Integer
Math Formula
String Splitter (Regex)
URI
字符串、日期时间、数值型自动转换
整型与浮点型
String to URI
URI to String
Collection（列表、集合）
List To Set
Set To List

非兼容数据类型转换及导出问题

01

数据类型转换冲突

在数据处理过程中，经常需要将一个数据类型转换为另一个数据类型。然而，不同数据类型之间可能存在兼容性问题，如将日期型字符串数据转换为转换为日期数据类型。

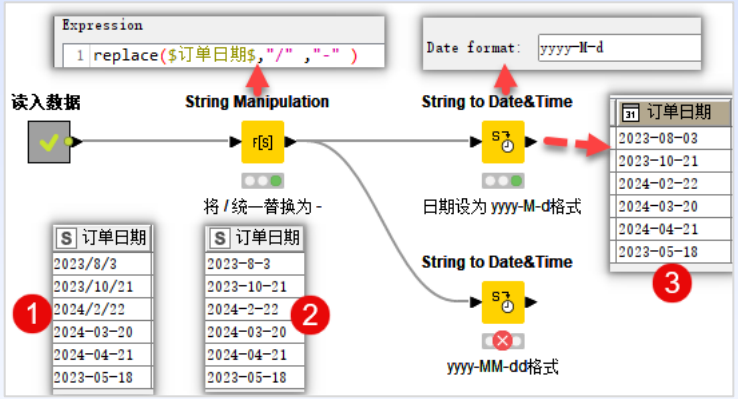
02

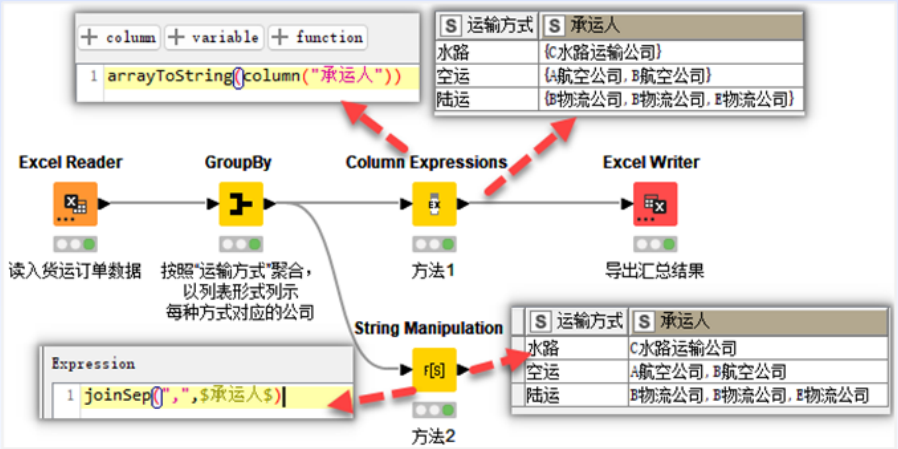
数据类型不匹配

比如在使用Excel Writer写入Excel文件时，若KNIME中列的数据类型为List或Set，则不能写入到Excel文件中

03

列表转字符串的方法







05

常见数据类型操作

字符串类型

字符串定义与表示

字符串是一系列字符的序列，通常用双引号或单引号进行标识。在编程中，字符串常用于表示文本数据或作为数据的容器。理解字符串的定义和表示方式对于后续操作非常重要。



字符串访问与操作

字符串提供了多种方法来访问和操作其内部的数据。常用的方法包括索引运算符、切片操作以及字符串函数，这些方法可以灵活地处理字符串中的不同部分。

字符串大小写转换

字符串的大小写转换是常见的字符串操作之一，可以通过特定的函数或方法来实现。例如在String Manipulation节点中可以使用函数将字符转换为小写或大写。

字符串拼接与替换

字符串拼接是将两个或多个字符串合并成一个新字符串的操作，而字符串替换则是将原字符串中的部分内容替换为新的文本。这些操作在编程中经常使用，以实现数据的有效管理和操作。

数值型

数值型数据简介

01

数值型数据是一类具有明确大小和顺序的数据类型，通常用于表示数量、金额等具体信息。这类数据在数据库中以二进制形式存储，可以参与数学运算和逻辑操作。

数值型数据基本操作

02

对数值型数据的基本操作包括加法、减法、乘法和除法。这些操作在数据处理和分析中非常重要，能够实现数值的快速增减和比例调整。

数值型数据排序与过滤

03

可以通过排序和过滤操作对数值型数据进行管理。排序操作可以按照指定条件对数值进行升序或降序排列，而过滤操作则用于筛选出符合特定条件的数值记录。

日期时间类型

日期时间类型概述

日期时间类型是用于存储和操作日期及时间值的数据类型。常见的日期时间类型包括DATETIME、DATE、TIMESTAMP等，每种类型有其特定的使用场景和功能。



DATETIME类型

DATETIME类型结合了日期和时间信息，可精确到年月日、时分秒。它适用于需要详细时间信息的场景，如记录事件发生的具体时间。

TIMESTAMP类型

TIMESTAMP类型专门用于时间戳的存储，能够自动记录插入时的当前时间，适合需要时间记录功能的应用，但注意不同时区的时间显示可能有差异。

日期时间类型的使用注意事项

在使用日期时间类型时，需要注意不同类型的特点和使用场景。例如，TIMESTAMP类型适合记录时间戳，而DATETIME则适合存储具体的日期和时间。此外，确保在合适的情况下使用正确的类型以避免数据错误。

列表和集合类型

列表定义与特点

列表是一种有序的、可变的容器，可以包含重复元素。通过索引访问和修改列表中的元素，索引从0开始。列表适用于存储不同类型的数据和进行有序集合操作。

集合定义与特点

集合是无序的、不可变的容器，只能包含唯一的元素。不能通过索引访问和修改集合，因为其元素唯一且无序。集合用于去重操作和集合运算，如交集、并集等。

列表与集合区别

列表是有序的、可变的容器，可以修改；集合是无序的、不可变的容器，不能修改。列表适用于有重复元素的有序集合操作，而集合适用于去重和集合运算。

列表操作与应用

列表可以进行添加、删除、查找、替换等操作。常见的列表操作包括遍历、排序、分割等。列表在数据处理中广泛应用，如实现堆排序算法、动态规划等。

集合基本操作

集合支持的基本操作包括添加（add）、删除（remove）、交集（intersection）、并集（union）和差集（difference）等。这些操作有助于实现数据去重、集合运算等功能。



06

本章小结

回顾与总结

01 数据类型转换重要性

数据类型转换是数据处理中的基本操作，用于在不同数据类型之间传递和处理信息。它满足了不同算法和业务逻辑的需求，确保数据能被正确识别和使用，从而提高了程序的健壮性和可读性。

02 常见数据类型转换技巧

基本数据类型之间的转换包括整型、浮点型和字符型的转换，通常通过隐式或显式方式进行。隐式转换由编译器自动执行，适用于数值类型间转换；显式转换需要手动指定类型，如使用函数int()、float()等来实现精确转换。

03 字符串与数据类型转换方法

字符串与基本数据类型的转换通过函数如str()实现，将字符串转换为整数、浮点数等。这种转换在处理日期时间字符串时尤为重要，确保数据可以被正确解析和操作。

04 类型转换注意事项

在进行数据类型转换时，需要注意避免因类型不兼容而导致的错误。例如，将字符串转换为数值型时可能遇到无法转换的情况。

实践建议

01

数据清洗实践

数据清洗是数据处理的关键步骤，通过检测和修正错误、处理缺失值和异常值等手段，确保数据的准确性和一致性。常见方法包括使用统计检测手段、规则引擎、或基于业务逻辑进行数据质量监控。

02

数据转换策略

数据转换将不同类型的数据转换为适合分析和存储的格式，如将文本数据转为数值型或日期时间格式。KNIME提供诸多常见的数据类型转换函数或节点。

感谢观看

THANKS



第五章 条件判断

基础必备，掌握条件判断技能



目录

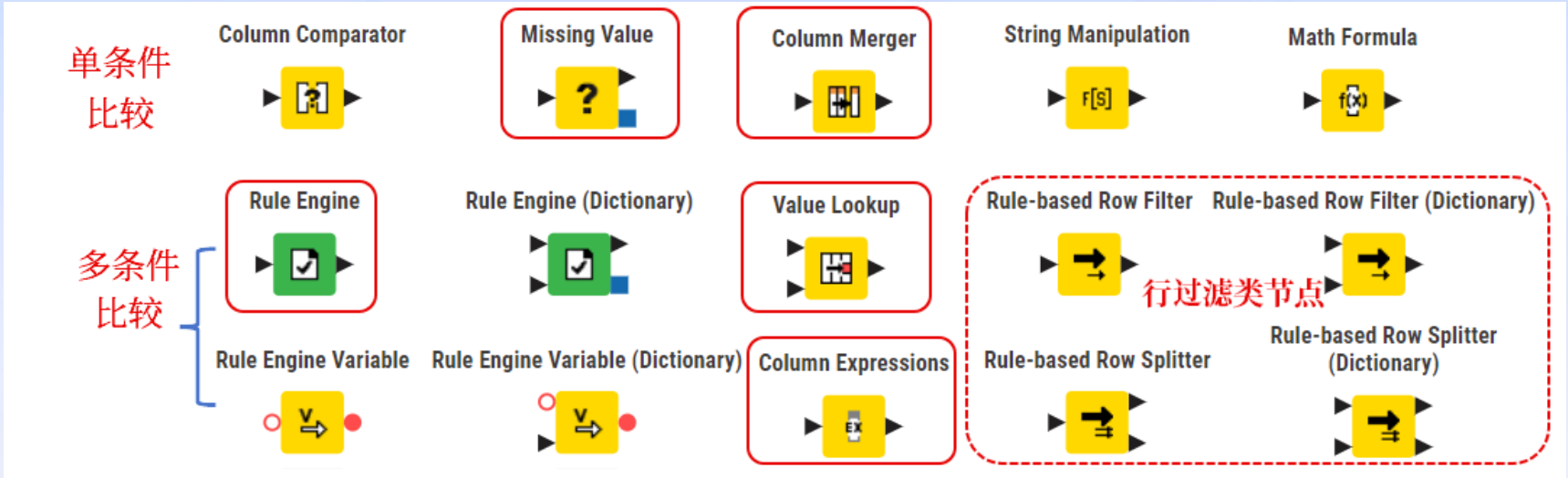
01 单条件判断

02 多条件判断

03 复杂条件编程类节点

04 本章小结

常用节点


单条件
比较
Column Comparator
Missing Value
Column Merger
String Manipulation
Math Formula
多条件
比较
Rule Engine
Rule Engine (Dictionary)
Value Lookup
Rule-based Row Filter
Rule-based Row Filter (Dictionary)
行过滤类节点
Rule Engine Variable
Rule Engine Variable (Dictionary)
Column Expressions
Rule-based Row Splitter
Rule-based Row Splitter
(Dictionary)



01

单条件判断

条件判断

基本条件判断语句

使用if-else语句是最基本的条件判断方法。通过if-else块，根据条件是否成立执行不同代码块。这是最常用的条件判断方式，适用于简单逻辑判断。

三元运算符应用

三元运算符是一种简短的条件计算方式，例如：condition ? value_if_true : value_if_false。它能够将表达式的结果直接赋值给变量，简化了条件判断的代码。

Column Expression节点支持该判断条件



多条件判断实现

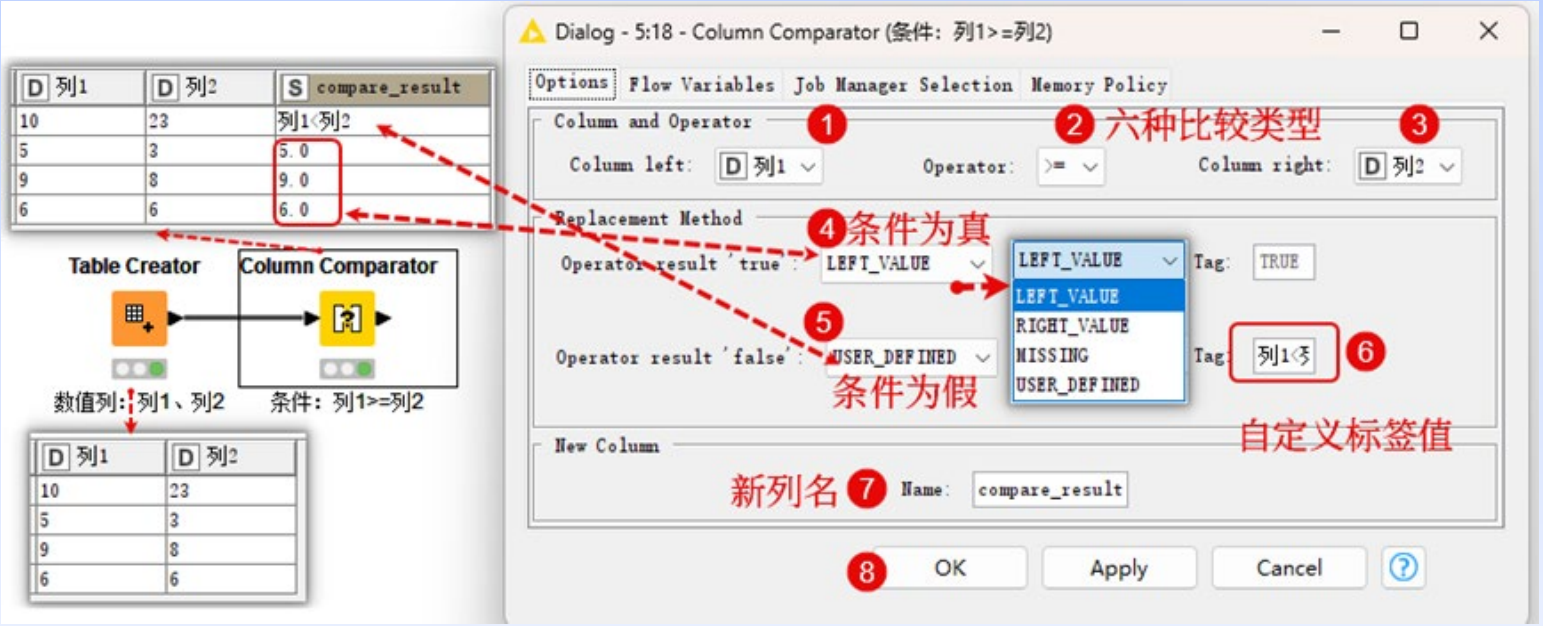
Rule类节点或Expression Row Filter节点中，可结合AND、OR逻辑运算符进行多个条件的判断。

多个并列判断条件，要注意做到不重复、不遗漏

部分节点隐含单个条件判断

如Column Comparator、Column Merger等

两列比较取值技巧


D 列1
D 列2
S compare_result
10
23
列1<列2
5
3
5.0
9
8
9.0
6
6
6.0
Table Creator
Column Comparator
数值列：列1、列2
条件：列1>=列2
D 列1
D 列2
10
23
5
3
9
8
6
6
Dialog - 5:18 - Column Comparator (条件：列1>=列2)
Options
Flow Variables
Job Manager Selection
Memory Policy
Column and Operator
1
2 六种比较类型
3
Column left:
D 列1
Operator:
>=
Column right:
D 列2
Replacement Method
4 条件为真
Operator result 'true':
LEFT_VALUE
LEFT_VALUE
RIGHT_VALUE
MISSING
USER_DEFINED
Tag:
TRUE
5
Operator result 'false':
USER_DEFINED
条件为假
Tag:
列1<列
6
自定义标签值
New Column
新列名
7
Name:
compare_result
8
OK
Apply
Cancel

缺失值条件判断取值

缺失值定义

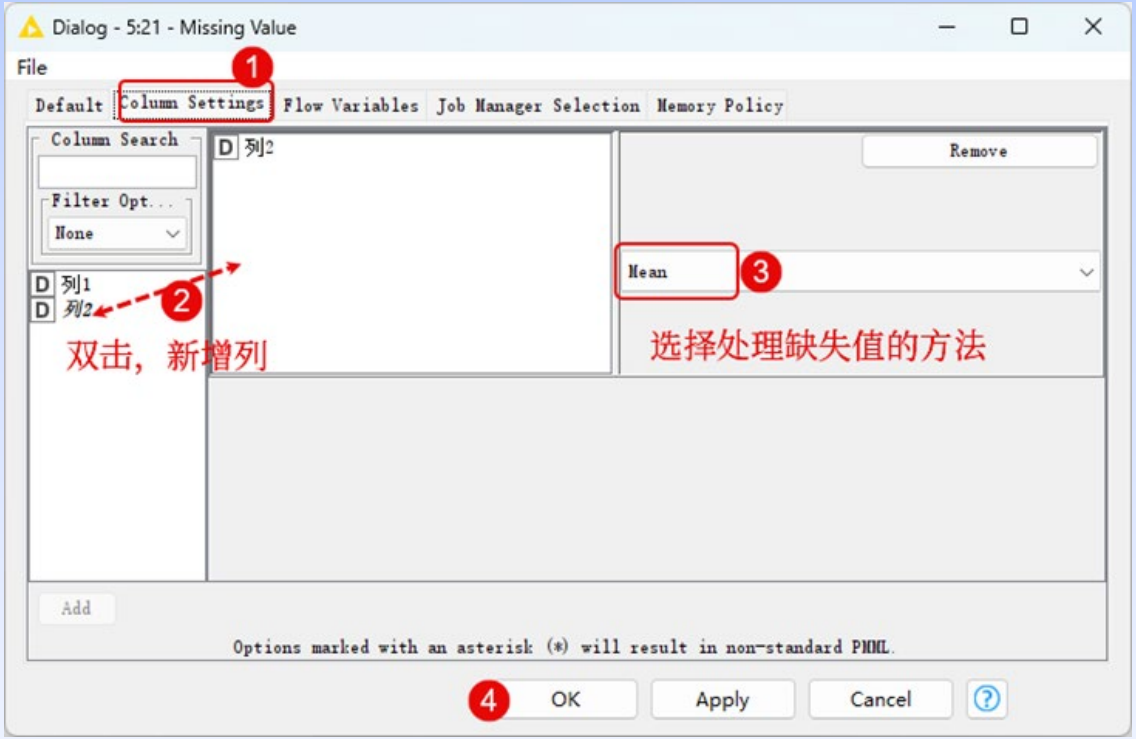
缺失值是指数据集中某些观测值的缺失情况，KNIME中显示红色问号。

缺失值删除

删除包含缺失值的行是常见的处理方法之一，如Row Filter节点

缺失值填充

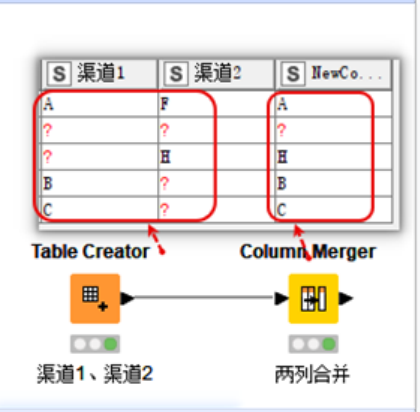
当删除缺失值会导致数据不完整时，可以使用多种方法对缺失值进行填充，如使用均值、中位数、插值法等。这些方法可以根据数据的特性选择最合适的填充策略，以恢复数据的完整性。 常 用Missing Value节点

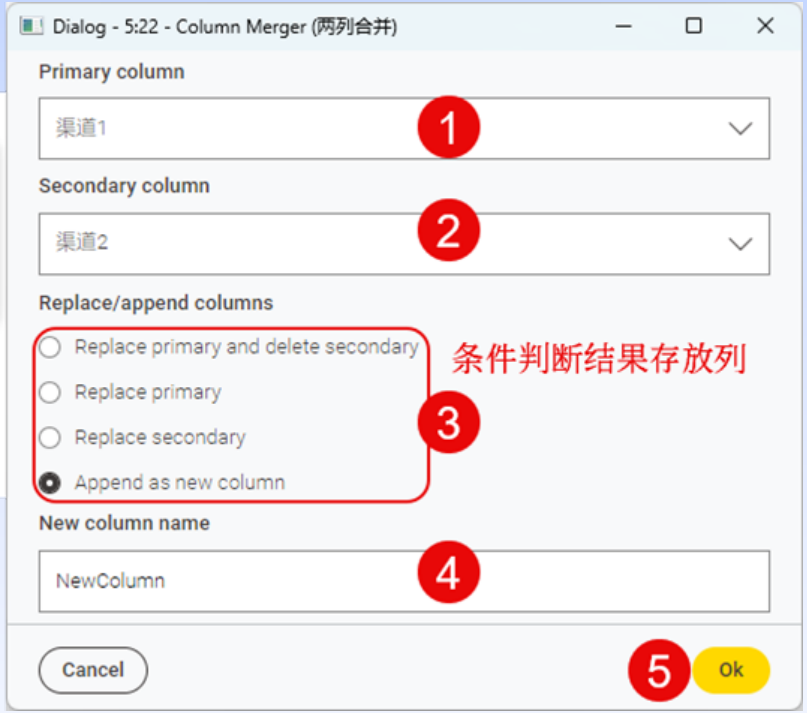


缺失值列填充（使用其他列值）

Column Merger

利用其他非缺失值列填充缺失值列





注意：Column Merge是以缺失值为判断条件，因此，切勿将空值当做缺失值

空值与缺失值区别及相互转换

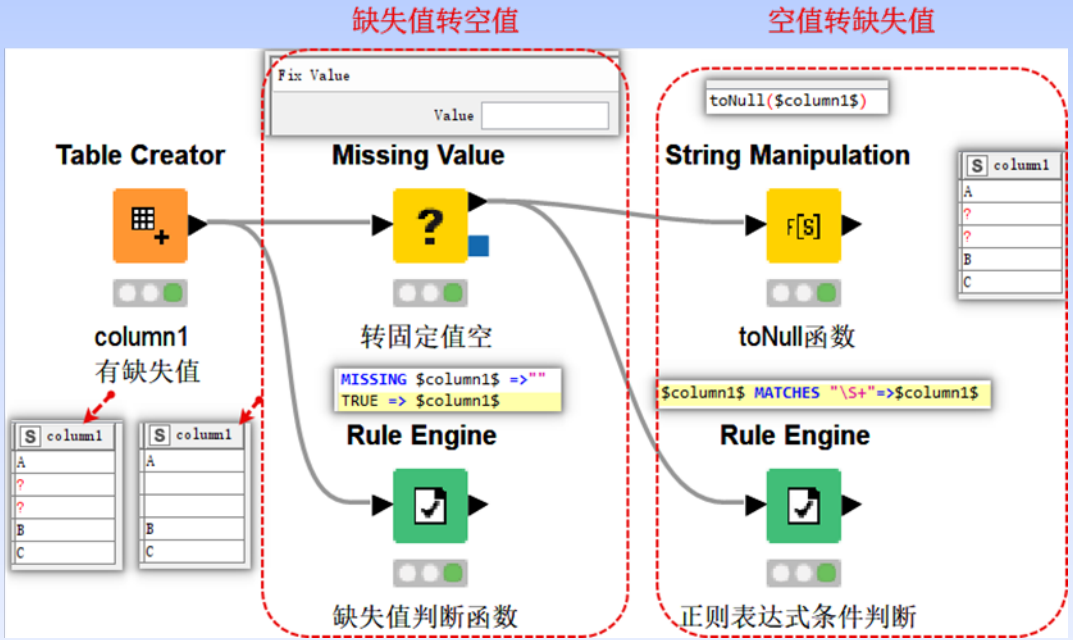
空值与缺失值区别

空值和缺失值都指向数据中缺少实际值的情况，但空值更侧重于变量或字段没有被赋予具体值，而缺失值则更广泛地指代数据中任何缺少实际值的情况，包括空值在内。

KNIME中如何区别：

缺失值：数据表单元格显示红色问题

空值：不显示任何字符，字符长度为0

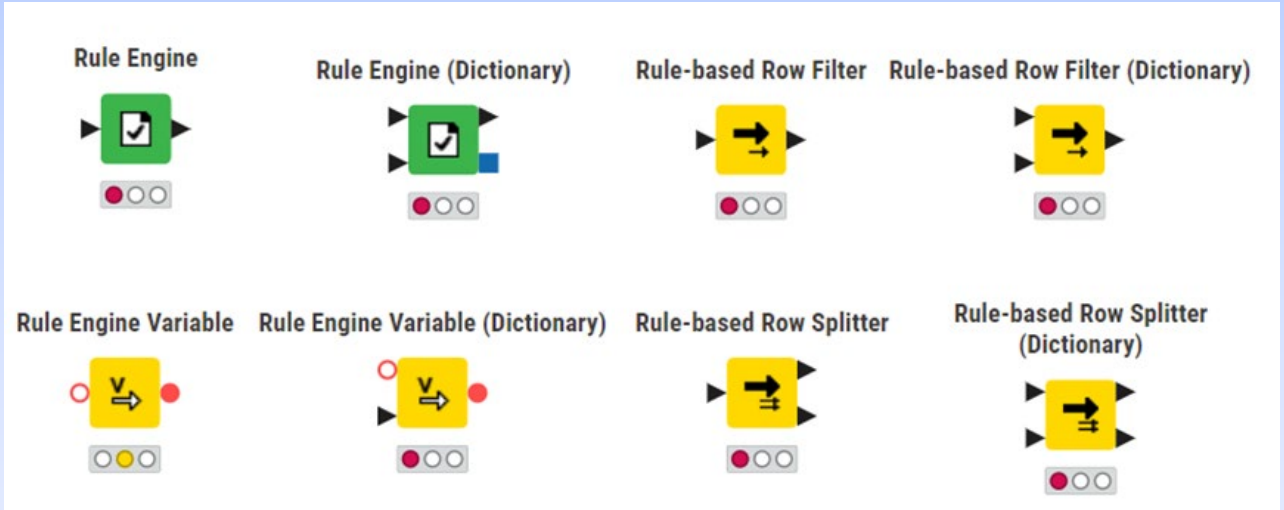




02

多条件判断

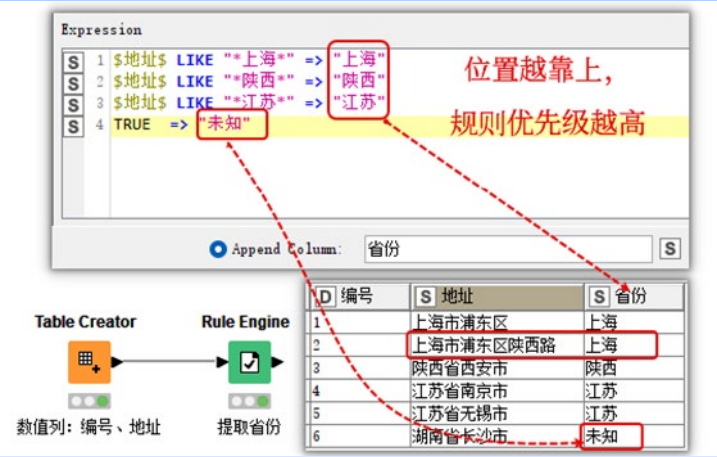
多条件判断Rule Engine类节点

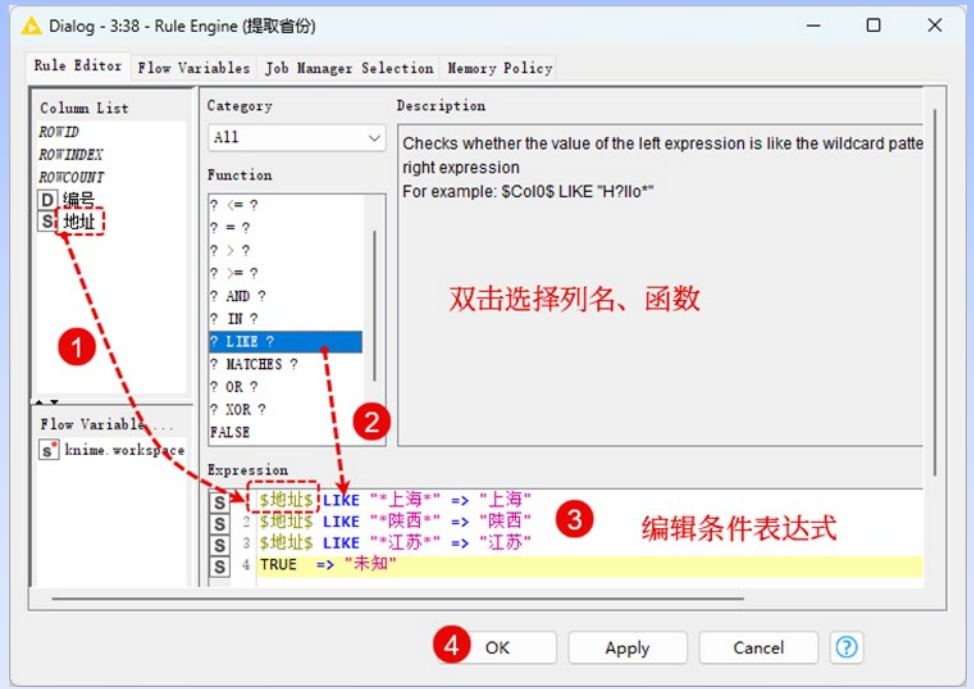


Rule Engine是一种基于规则判断的节点，学会一个节点，不难掌握其他同类型的节点

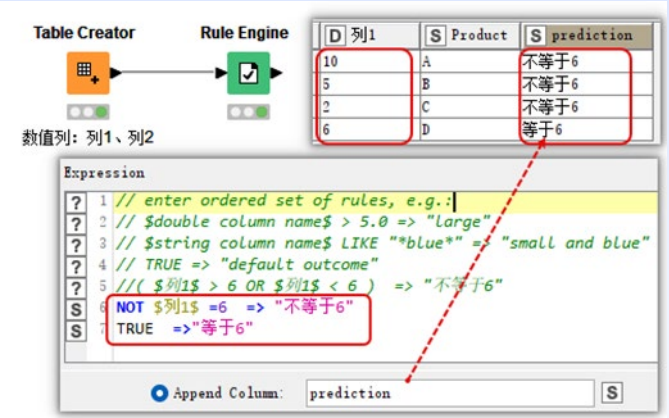
Rule Engine节点示例

示例1

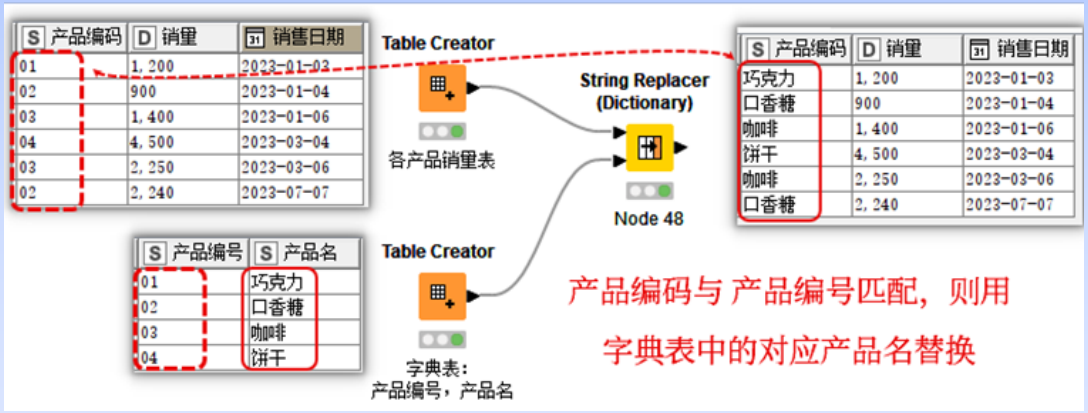




示例2



String Replacer（Dictionary）



产品编码	销量	销售日期
01	1, 200	2023-01-03
02	900	2023-01-04
03	1, 400	2023-01-06
04	4, 500	2023-03-04
03	2, 250	2023-03-06
02	2, 240	2023-07-07

产品编号	产品名
01	巧克力
02	口香糖
03	咖啡
04	饼干

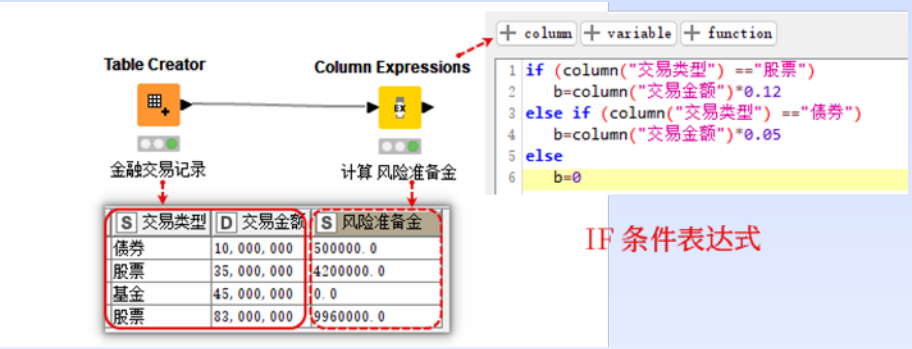
产品编码	销量	销售日期
巧克力	1, 200	2023-01-03
口香糖	900	2023-01-04
咖啡	1, 400	2023-01-06
饼干	4, 500	2023-03-04
咖啡	2, 250	2023-03-06
口香糖	2, 240	2023-07-07

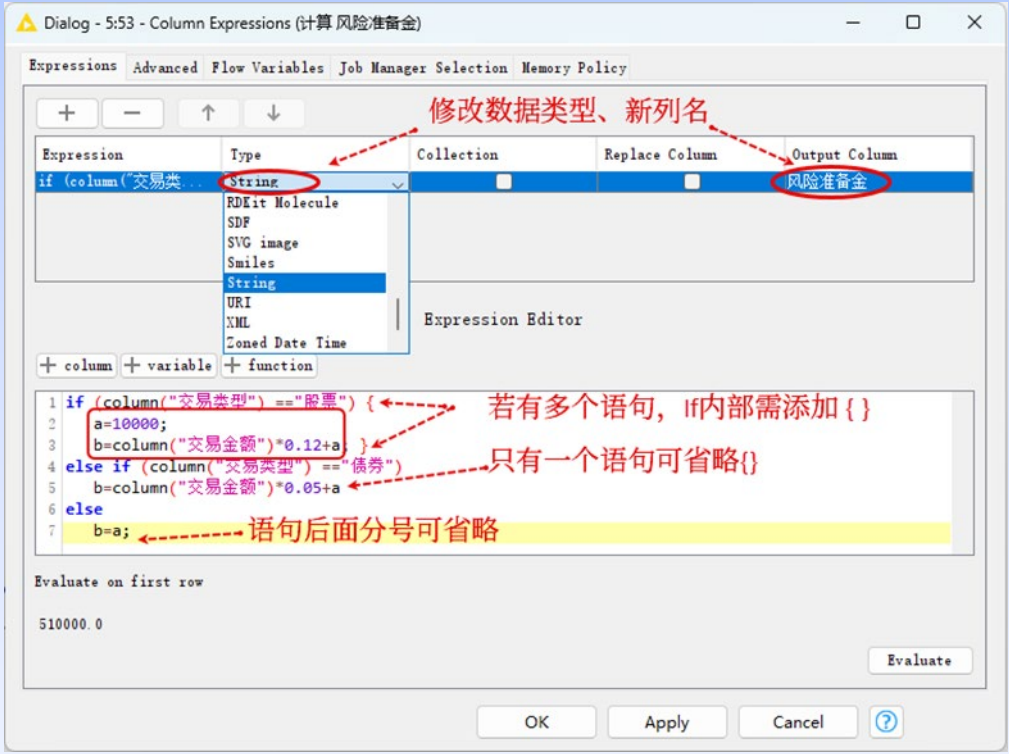


Column Expressions使用

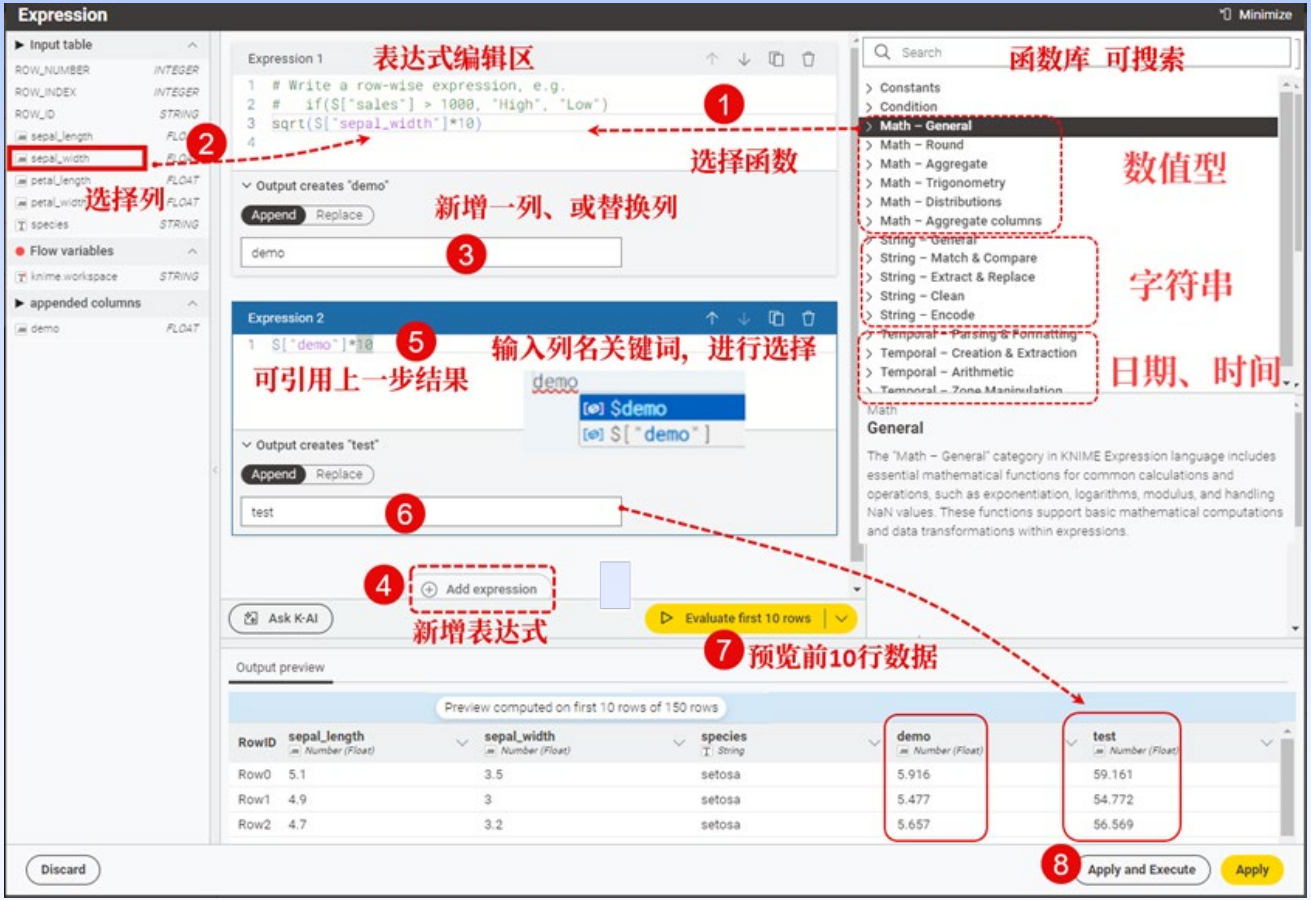
Column Expressions简介

Column Expressions拥有众多内置函数，可以写判断条件，也可以做循环、自定义函数，务必重点掌握





Expression节点使用(Expression 5.3版本新增)



Expression拥有众多内置函数，通过函数嵌套，可以实现更加灵活的条件判断

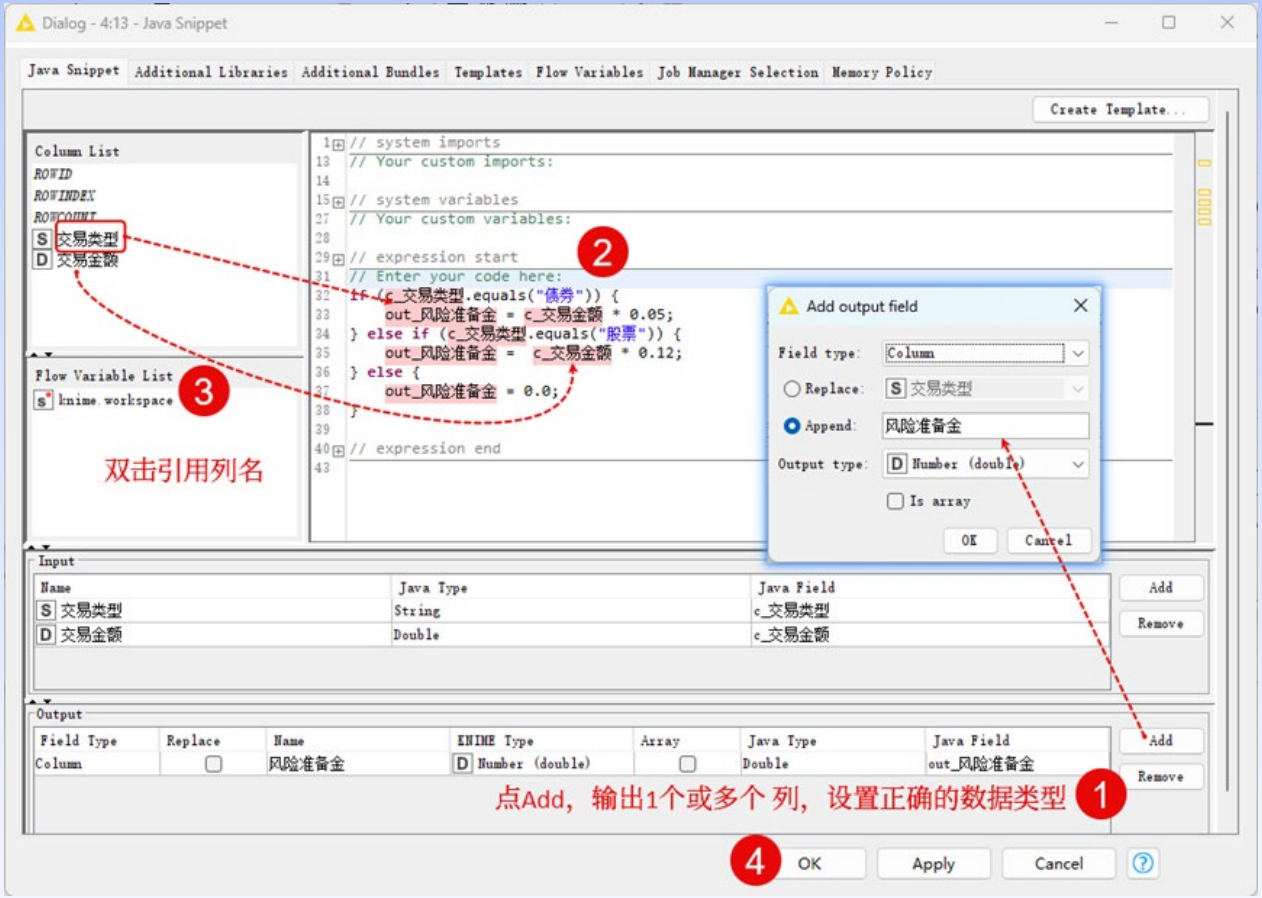
该节点提供的多个条件判断的**If和switch**函数，无需多重嵌套，非常方便



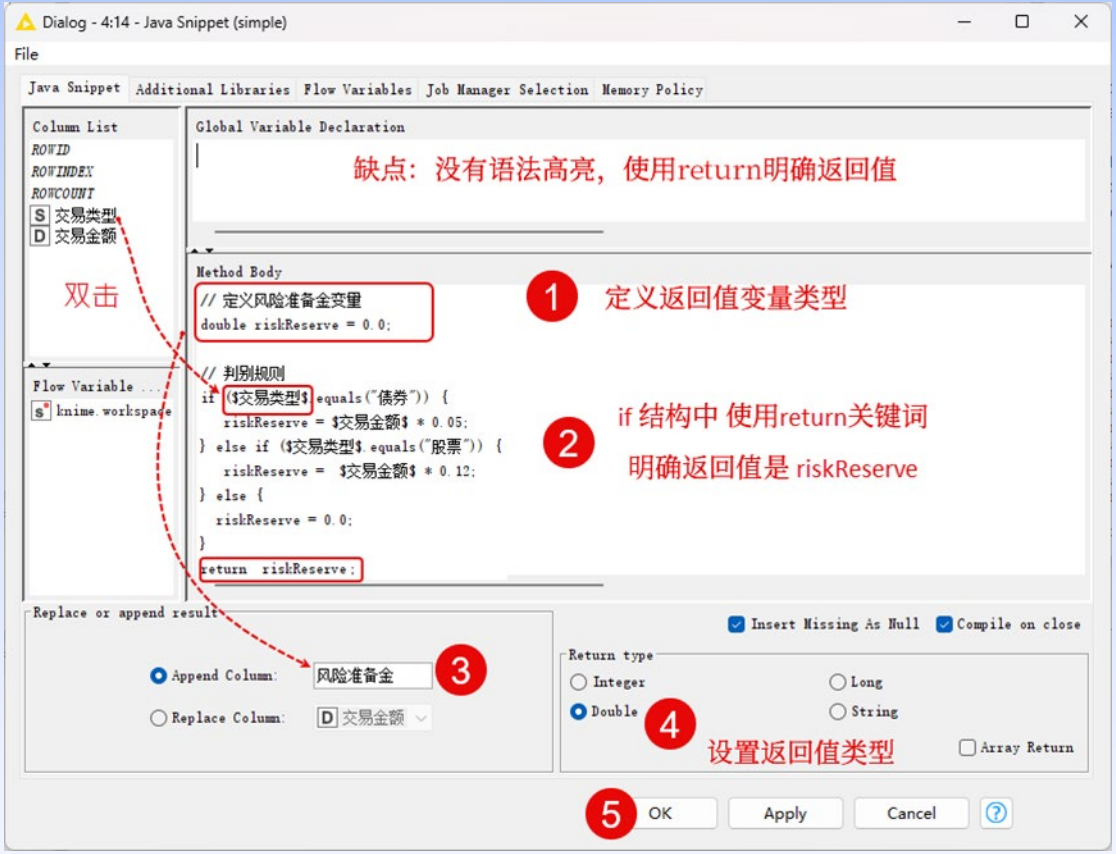
03

复杂条件编程类节点

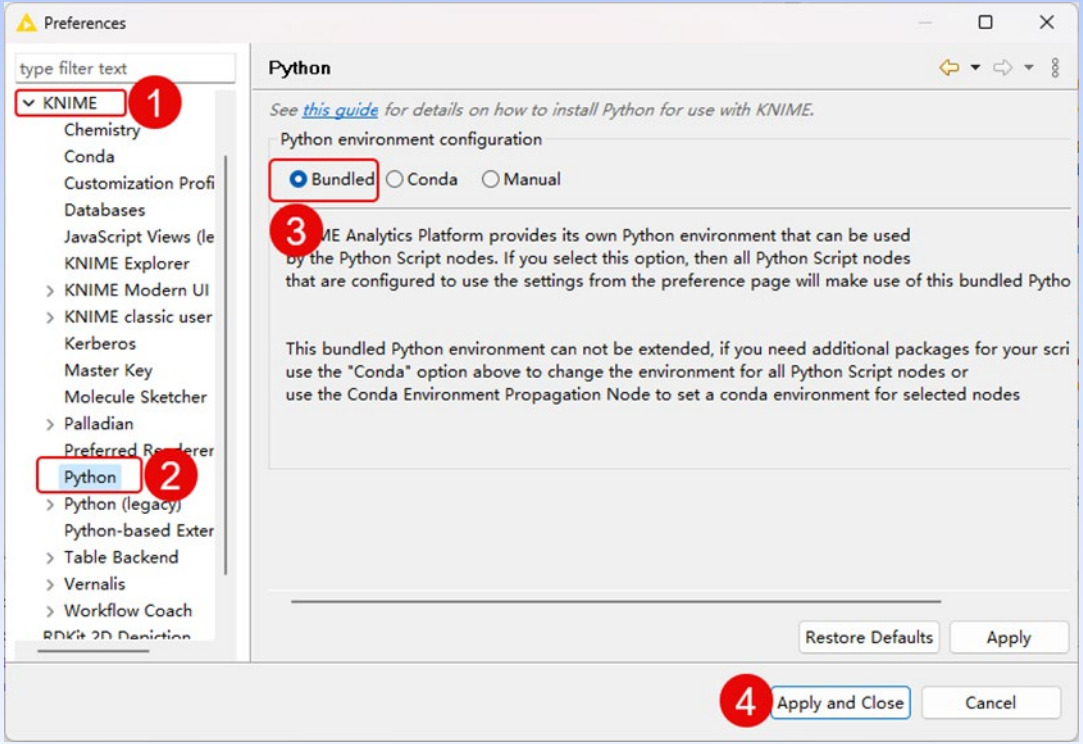
Java编程节点详解—Java Snippet



Java编程节点详解—Java Snippet（simple）



Python编程节点（利用集成环境）



Python编程节点（Python Script)





04

本章小结

回顾主要内容

条件判断基本概念

条件判断是编程中的基本概念，用于根据特定条件执行不同代码块。常见的条件判断语句包括if、else和switch，通过这些语句可以实现对不同情况的处理。



单条件判断方法

单条件判断涉及评估一个条件是否为真，并根据结果执行相应的代码块。常用的方法包括使用if语句进行布尔值判断，或者使用其暗含单个条件判断的节点，如Column Merger、Missing Value、Row Filter等。

多条件判断策略

多条件判断通常需要使用更复杂的逻辑结构，如使用结合条件运算符和逻辑表达式实现更复杂的多条件判断功能。

常用带有“Rule”关键字节点、及Column Expressions、Expression节点

复杂条件编程技术

对于更为复杂的条件判断需求，Java和Python中的多分支语句和字典索引，以实现更加灵活和强大的条件判断功能。

总结学习心得

条件判断重要性

掌握条件判断是编程中的基本技能，它允许我们根据不同情况执行不同操作。无论是单条件还是多条件判断，都是提高程序灵活性和逻辑性的关键。

学习方法与技巧

学习条件判断时，应重点理解各种操作符的用法，并通过实际编写代码进行实践。同时，多阅读他人的代码示例，可以帮助更快地理解和掌握复杂条件的判断方法。

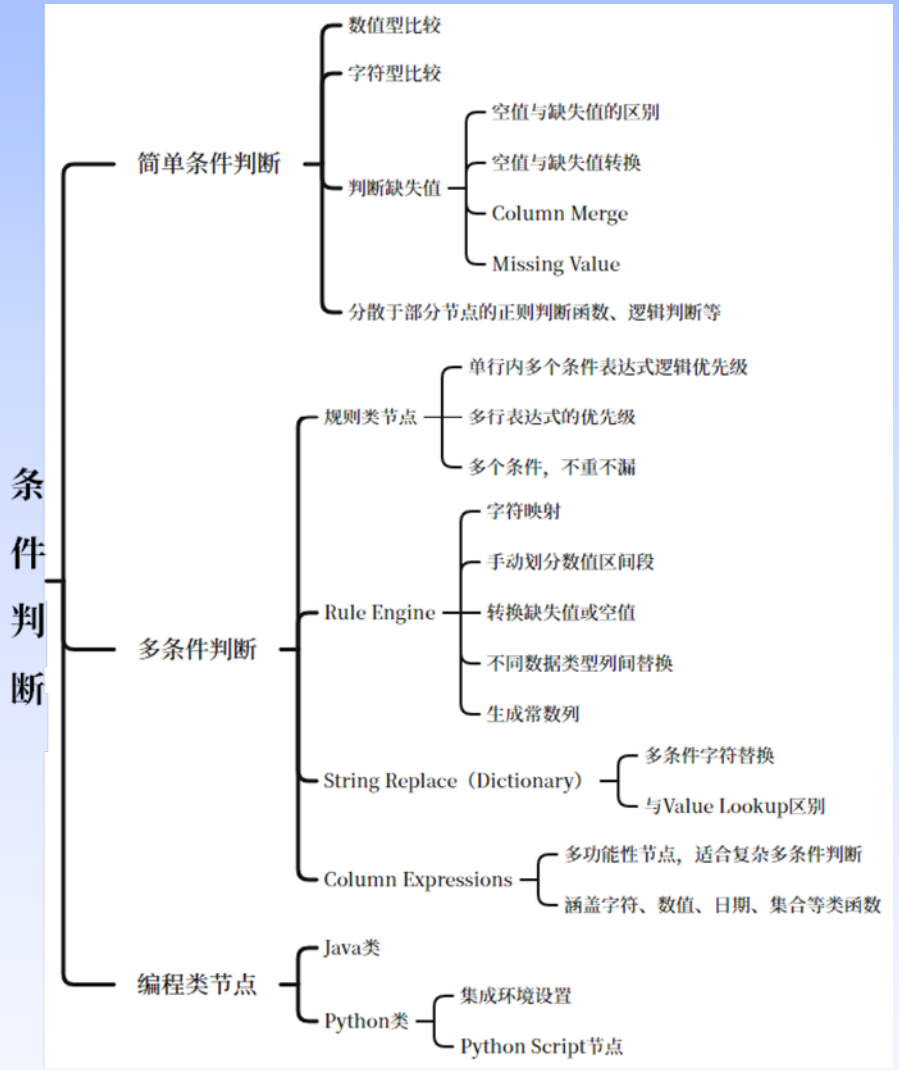
常见错误及解决方案

在学习条件判断过程中，可能会遇到一些常见的错误，如条件表达式不匹配、误用操作符等。通过总结这些错误及其解决方案，可以提升问题解决能力并避免重复犯错。

条件判断实际应用案例

在实际项目中，条件判断常用于数据筛选、业务逻辑处理等方面。通过分析具体案例，可以更好地理解条件判断在实际应用中的重要性和多样化应用。

本章总结


条件判断
简单条件判断
数值型比较
字符型比较
判断缺失值
空值与缺失值的区别
空值与缺失值转换
Column Merge
Missing Value
分散于部分节点的正则判断函数、逻辑判断等
多条件判断
规则类节点
单行内多个条件表达式逻辑优先级
多行表达式的优先级
多个条件，不重不漏
Rule Engine
字符映射
手动划分数值区间段
转换缺失值或空值
不同数据类型列间替换
生成常数列
String Replace（Dictionary）
多条件字符替换
与Value Lookup区别
Column Expressions
多功能性节点，适合复杂多条件判断
涵盖字符、数值、日期、集合等类函数
编程类节点
Java类
Python类
集成环境设置
Python Script节点

感谢观看

THANKS



第六章
表级别操作与属性管理

深入解析表级别操作，熟练掌握各种变换



目录

01 表格左右合并

02 表格左右切分

03 表格上下合并

04 表格上下切分

05 表格裁切

06 表格结构重塑

07 获取表格属性

08 修改表格属性

09 验证表格属性

10 比较表格属性或数据差异

11 本章小结



01

表格左右合并

Joiner

Joiner概述

Joiner是一种用于表格数据合并的工具，通过将两个或多个表格的数据行进行连接，创建一个新的数据集。它常用于整合来自不同来源的数据，提供更全面的分析。

操作步骤

使用Joiner合并表格数据时，首先需要选择要合并的表数据，然后指定连接字段，最后执行合并操作。

注意事项

在使用Joiner进行数据合并时，应确保所选的连接字段具有唯一性，以避免数据重复和错误。此外，还需要检查数据类型和格式是否兼容，以确保合并操作顺利进行。

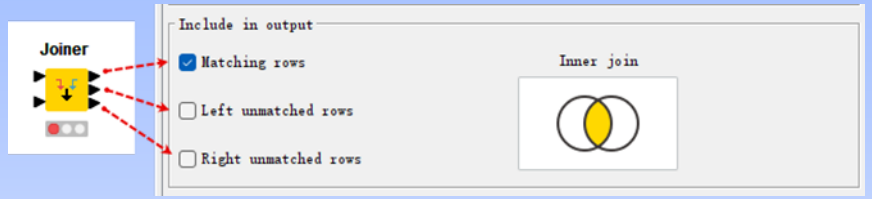
常见问题及解决方案

常见的问题包括数据不匹配、重复记录和连接字段选择不当等。为解决这些问题，可以预先进行数据清洗和验证，选择合适的连接字段，并使用高级选项如去重和错误处理功能来提高数据合并的准确性。

Join在手，多表不愁

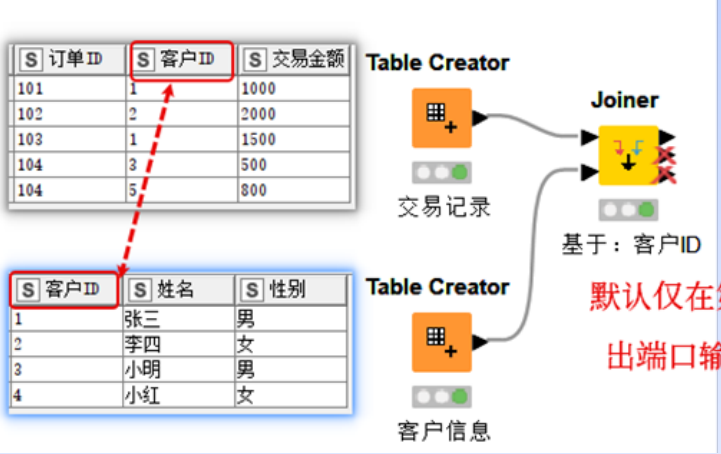
- 匹配模式
- 一对一 、 一对多

Joiner



类型	Matching row 第一输出端口	Left unmatched rows 第二输出端口	Right unmatched row 第三输出端口	join 类型图示
Inner join 内连接	√			
Left antijoin 左反连接		√		
Right antijoin 右反连接			√	
Left outer join 左外连接	√	√		
Right outer join 右外连接	√		√	
Full antijoin 全反连接		√	√	
Full outer join 全连接	√	√	√	

Joiner



S 订单ID	S 客户ID	S 交易金额
101	1	1000
102	2	2000
103	1	1500
104	3	500
104	5	800

S 客户ID	S 姓名	S 性别
1	张三	男
2	李四	女
3	小明	男
4	小红	女

1 内连接

S 订单ID	S 客户ID	S 交易金额	S 客户ID (right)	S 姓名	S 性别
101	1	1000	1	张三	男
102	2	2000	2	李四	女
103	1	1500	1	张三	男
104	3	500	3	小明	男

2 左连接

S 订单ID	S 客户ID	S 交易金额	S 客户ID (right)	S 姓名	S 性别
101	1	1000	1	张三	男
102	2	2000	2	李四	女
103	1	1500	1	张三	男
104	3	500	3	小明	男
104	5	800	?	?	?

3 右连接

S 订单ID	S 客户ID	S 交易金额	S 客户ID (right)	S 姓名	S 性别
101	1	1000	1	张三	男
102	2	2000	2	李四	女
103	1	1500	1	张三	男
104	3	500	3	小明	男
?	?	?	4	小红	女

4 全连接

S 订单ID	S 客户ID	S 交易金额	S 客户ID (right)	S 姓名	S 性别
101	1	1000	1	张三	男
102	2	2000	2	李四	女
103	1	1500	1	张三	男
104	3	500	3	小明	男
104	5	800	?	?	?
?	?	?	4	小红	女

核心节点，务必熟练掌握

Cross Joiner

Cross Joiner定义

Cross Joiner是用于在两个或多个表中获取所有可能组合的连接方式，不依赖于任何关联条件。它返回两个表中的每一行与另一个表中的每一行的所有组合，形成笛卡尔积。

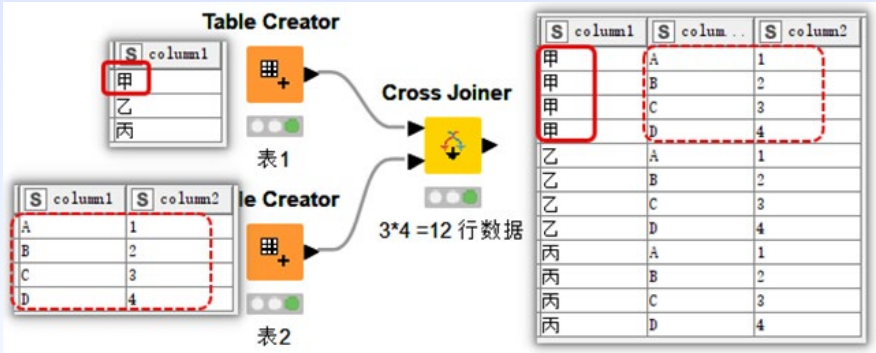
Cross Joiner应用场景

Cross Joiner常用于生成多列组合值，例如学校课程选修情况或汉字全拼。它适用于需要从多个不相关表中提取数据的场景，有助于快速生成复杂的组合信息。



Cross Joiner限制

Cross　Joiner返回的结果包含所有表的总行数的乘积，这可能导致结果集过大。因此，在使用时应谨慎，以避免处理不必要的大量数据，特别是在数据量大的情况下。



Value Lookup

Value Lookup节点

类似Joiner节点，但只能指定查找匹配的1列，不能是多列，其匹配结果类别Joiner中的左连接。

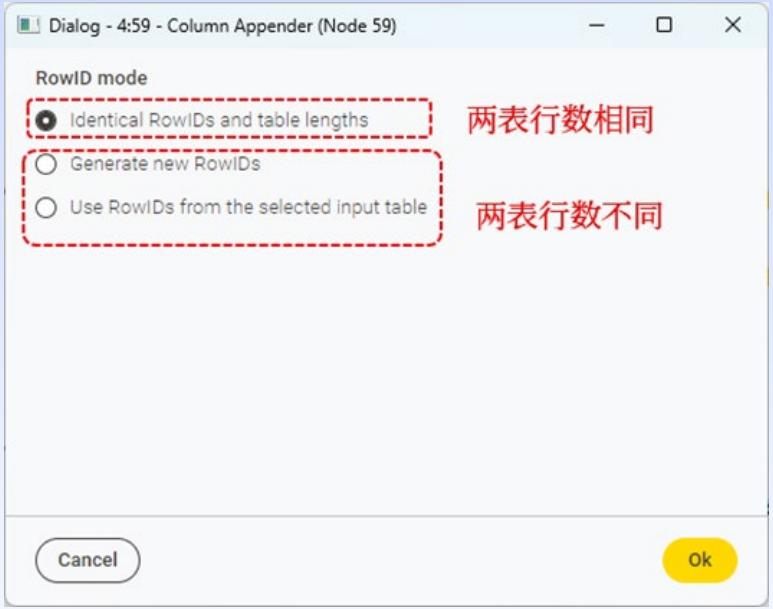


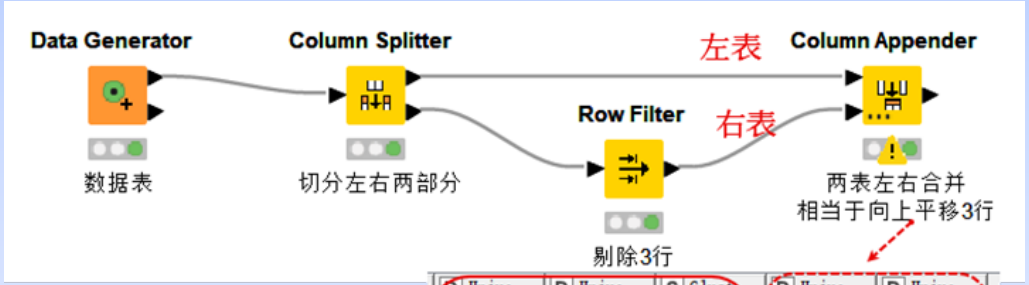
Value Lookup节点只能处理一对一的关系，若出现一对多，比如说01编码对应字典表中的多个产品，此时节点中可以选择匹配第一个值或最后一个值，甚至还能在不能匹配时，选择匹配距离下一个最近的较大值或较小值等，这两点与Joiner节点有着很大的不同。

Column Appender

Column Appender 概述

Column Appender 方法用于将新的列对象添加到表格的列集合中。





多列向上（下）平移

D Unive...	D Unive...	S Clust...	D Unive...	D Unive...
0.571	0.862	Cluster_0	0.619	0.394
0.598	0.011	Cluster_0	0.789	0.216
0.589	0.977	Cluster_0	0.866	0.608
0.329	0.672	Cluster_1	0.338	0.733
0.441	0.144	Cluster_1	0.484	0.635
0.311	0.768	Cluster_1	0.234	0.518
0.002	0.735	Cluster_2	0.895	0.315
0.141	0.768	Cluster_2	0.479	0.414
0.651	0.874	Cluster_2	0.381	0.396
0.82	0.395	Cluster_3	?	?
0.092	0.212	Cluster_3	?	?
0.133	0.408	Cluster_3	?	?



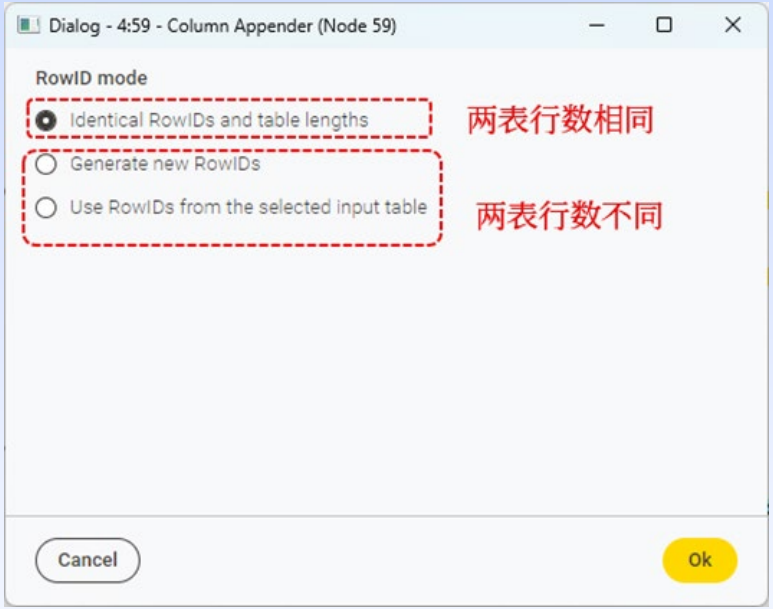
02

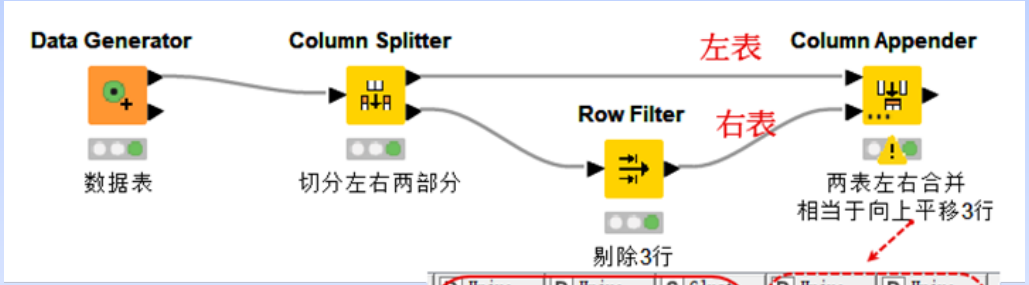
表格左右切分

Column Splitter

Column Splitter 概述

根据选择的列，将表格切分为两部分





多列向上（下）平移

D Unive...	D Unive...	S Clust...	D Unive...	D Unive...
0.571	0.862	Cluster_0	0.619	0.394
0.598	0.011	Cluster_0	0.789	0.216
0.589	0.977	Cluster_0	0.866	0.608
0.329	0.672	Cluster_1	0.338	0.733
0.441	0.144	Cluster_1	0.484	0.635
0.311	0.768	Cluster_1	0.234	0.518
0.002	0.735	Cluster_2	0.895	0.315
0.141	0.768	Cluster_2	0.479	0.414
0.651	0.874	Cluster_2	0.381	0.396
0.82	0.395	Cluster_3	?	?
0.092	0.212	Cluster_3	?	?
0.133	0.408	Cluster_3	?	?

Column Splitter与Column Appender为互逆操作节点



03

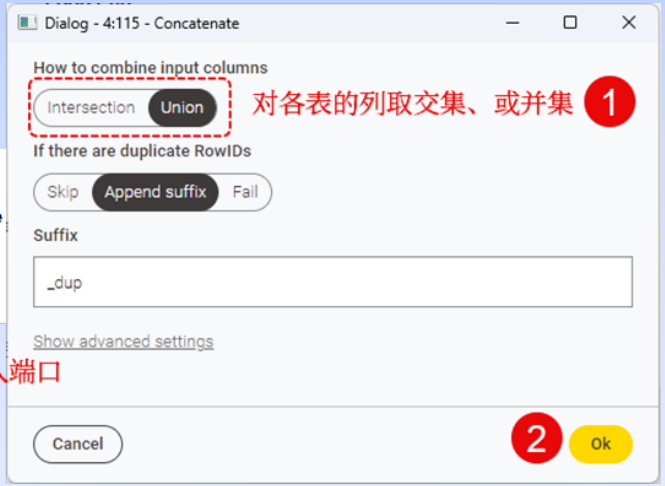
表格上下合并

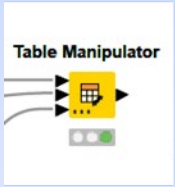


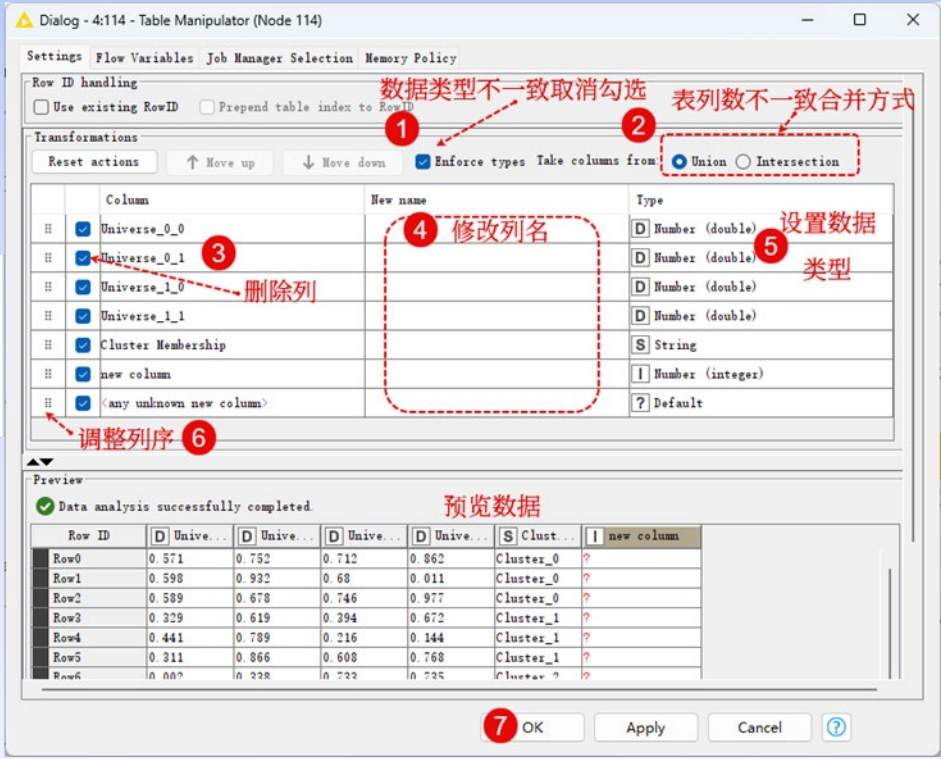
手动合并


Concatenate

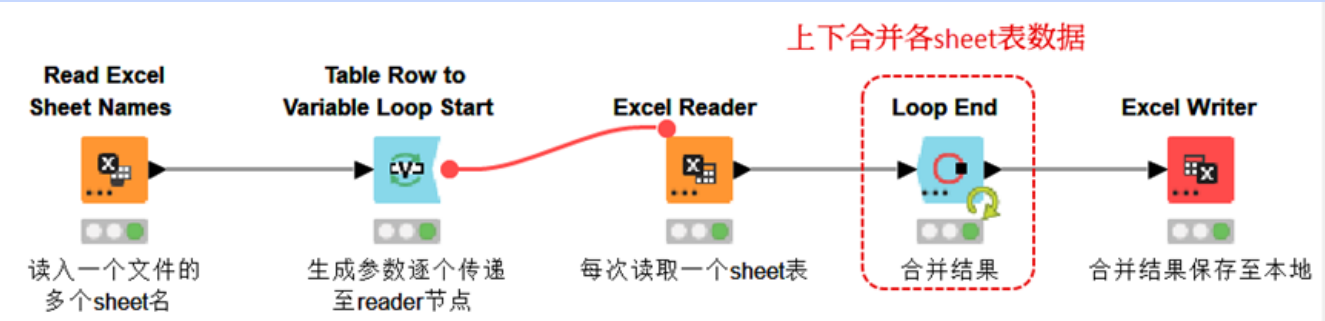
增加输入端口


Dialog - 4:115 - Concatenate
How to combine input columns
Intersection
Union
对各表的列取交集、或并集
1
If there are duplicate RowIDs
Skip
Append suffix
Fail
Suffix
_dup
Show advanced settings
Cancel
2
Ok


Table Manipulator


Dialog - 4:114 - Table Manipulator (Node 114)
Settings
Flow Variables
Job Manager Selection
Memory Policy
Row ID handling
Use existing RowID
Prepend table index to RowID
数据类型不一致取消勾选
1
表列数不一致合并方式
2
Transformations
Reset actions
Move up
Move down
Enforce types
Take columns from:
Union
Intersection
Column
New name
Type
Universe_0_0
Universe_0_1
Universe_1_0
Universe_1_1
Cluster Membership
new column
<any unknown new column>
Number (double)
String
Number (integer)
Default
3
删除列
4
修改列名
5
设置数据类型
调整列序
6
Preview
Data analysis successfully completed.
预览数据
Row ID
Row0
Row1
Row2
Row3
Row4
Row5
Cluster_0
Cluster_1
7
OK
Apply
Cancel

自动合并



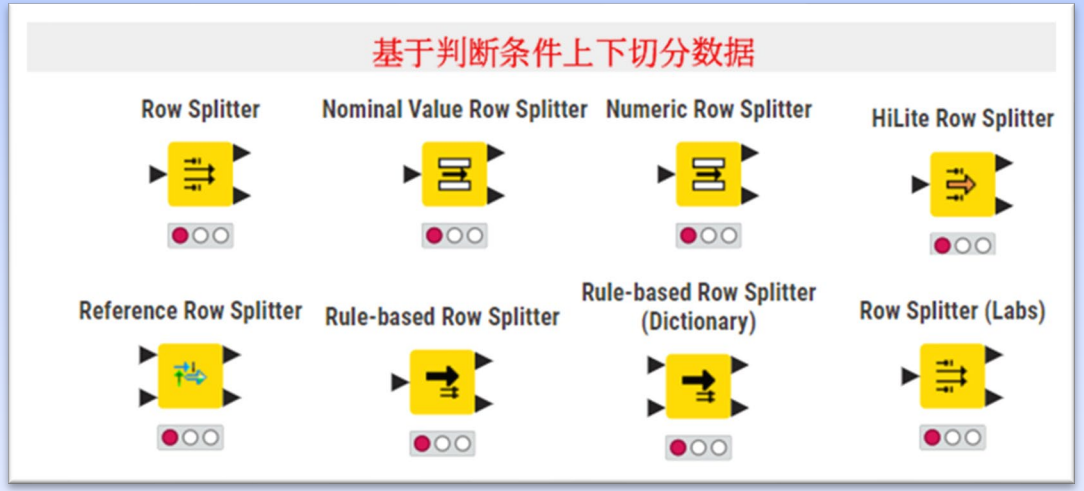
第10章循环再展开



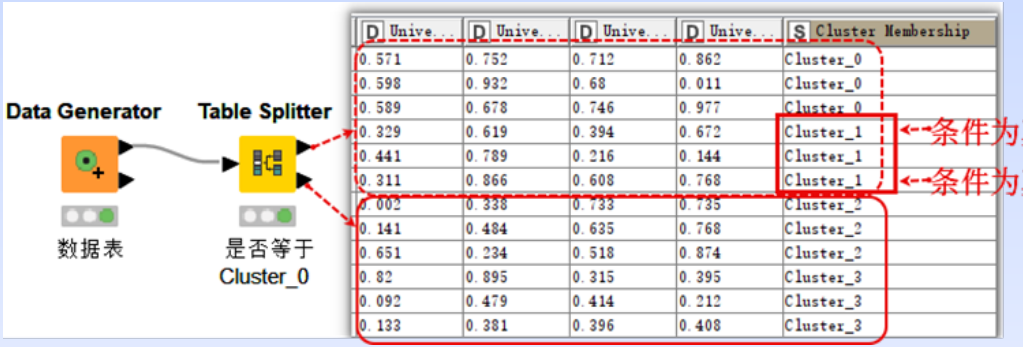
04

表格上下切分

基于判断条件的真假

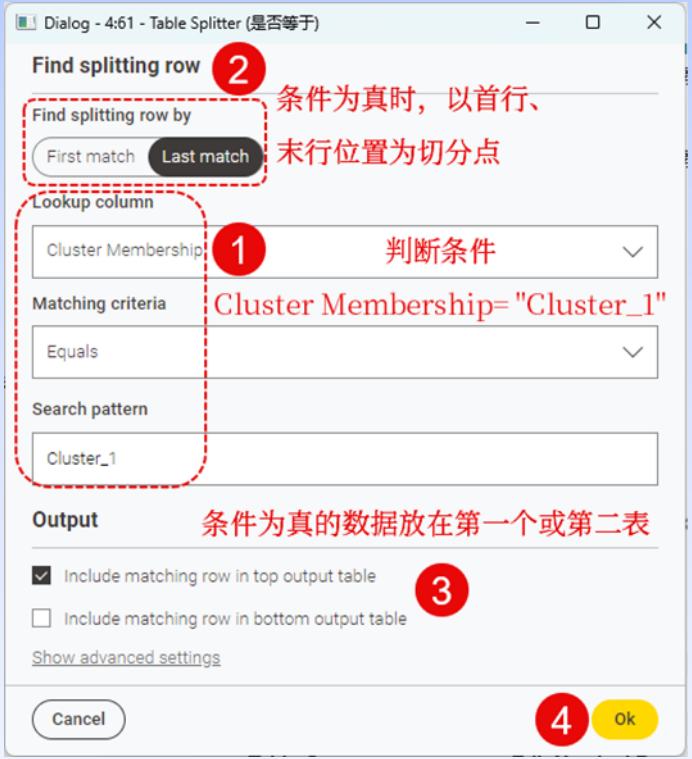

基于判断条件上下切分数据
Row Splitter
Nominal Value Row Splitter
Numeric Row Splitter
HiLite Row Splitter
Reference Row Splitter
Rule-based Row Splitter
Rule-based Row Splitter (Dictionary)
Row Splitter (Labs)

基于判断条件为真的行位置

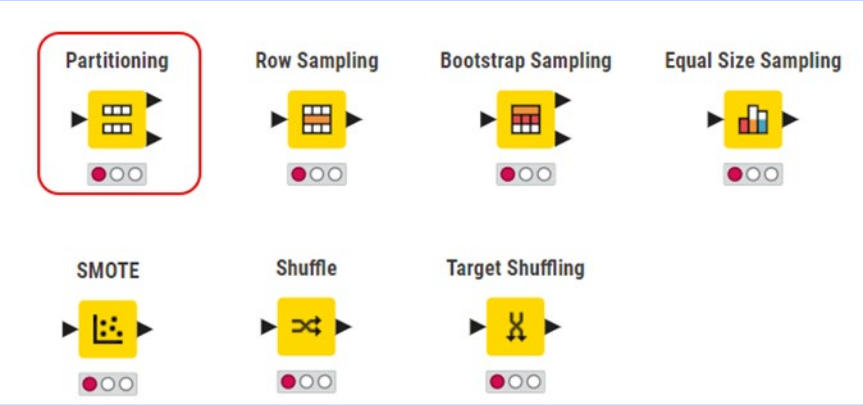


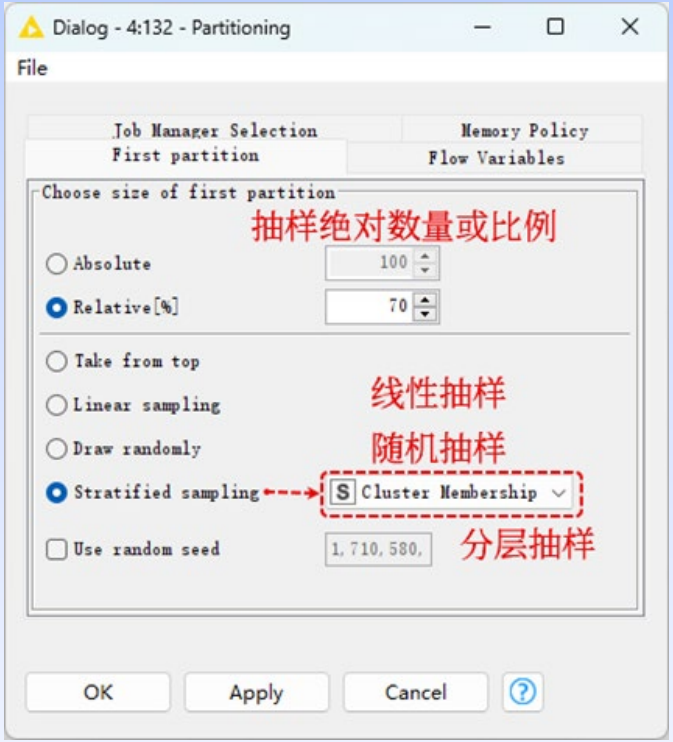
Unive...	Unive...	Unive...	Unive...	Cluster Membership
0.571	0.752	0.712	0.862	Cluster_0
0.598	0.932	0.68	0.011	Cluster_0
0.589	0.678	0.746	0.977	Cluster_0
0.329	0.619	0.394	0.672	Cluster_1
0.441	0.789	0.216	0.144	Cluster_1
0.311	0.866	0.608	0.768	Cluster_1
0.002	0.338	0.733	0.735	Cluster_2
0.141	0.484	0.635	0.768	Cluster_2
0.651	0.234	0.518	0.874	Cluster_2
0.82	0.895	0.315	0.395	Cluster_3
0.092	0.479	0.414	0.212	Cluster_3
0.133	0.381	0.396	0.408	Cluster_3

←条件为真的首行
←条件为真的末行



基于模型训练和测试目的



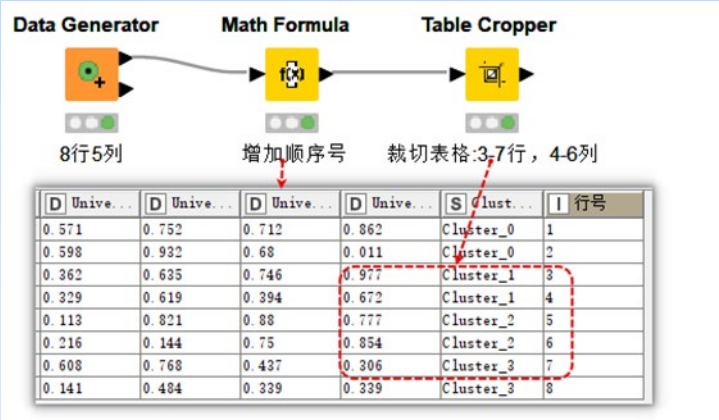




05

表格裁切

表格裁切

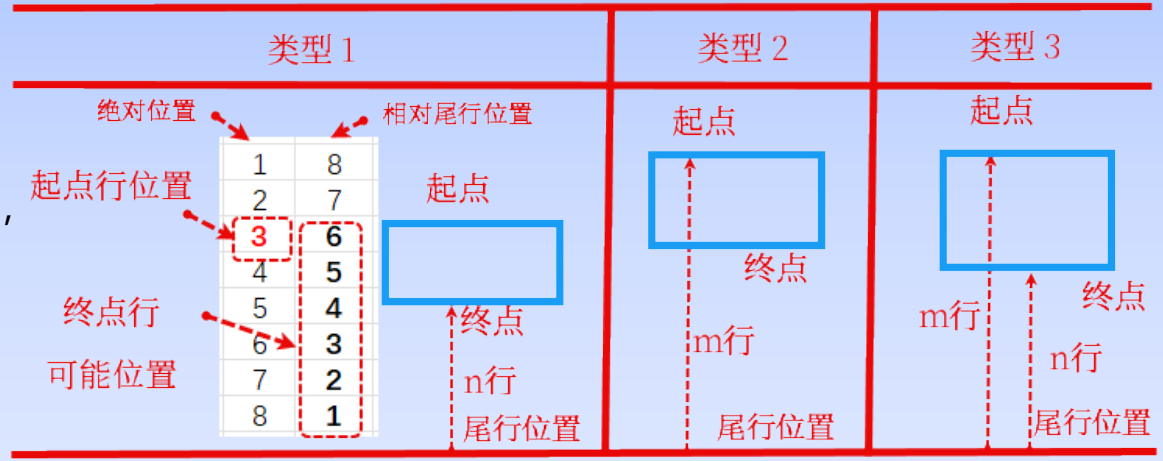


表格的裁切主要是对连续的行与列选择一定的范围，可类比裁切图片那样，允许用户仅提取出感兴趣的部分连续区域数据。



表格裁切

Table Cropper节点功能上理解很简单，但界面设置中的相对位置很容易让人困惑，不知如何设置。其实该节点的设置要从行与列的绝对位置和相对位置两个方面来理解，就很容易掌握。



与Table Cropper节点类似的节点还有 Cell Updater节点

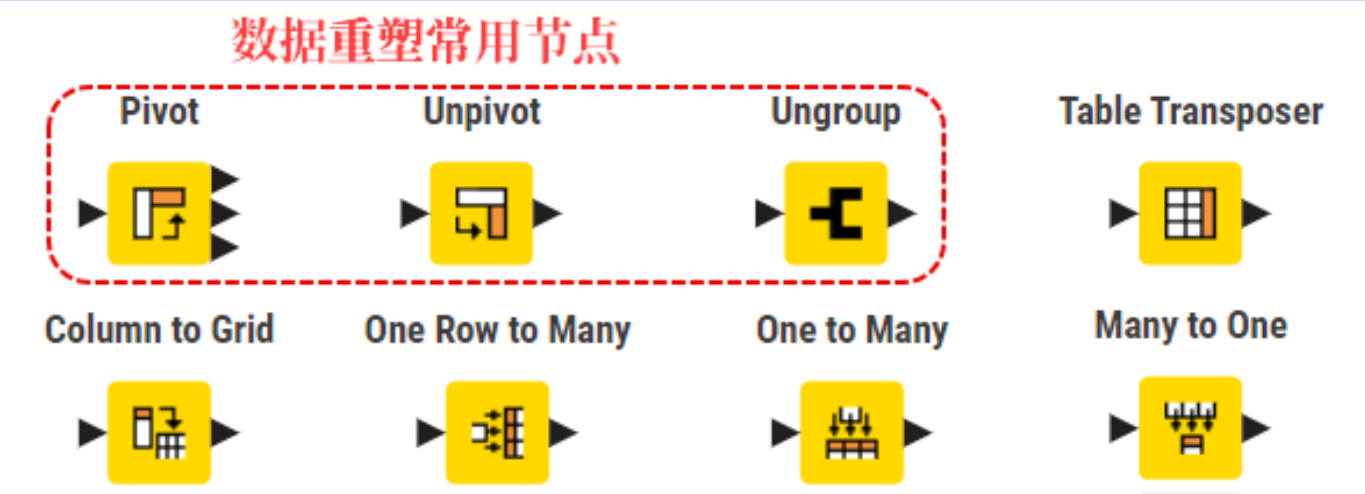
Cell Updater节点也有行的绝对位置和相对于尾部的相对位置选项，该节点主要利用流变量来更新数据表中选择列的指定行位置数据。掌握了Table Cropper节点，再来学习Cell Updater节点应该更容易。



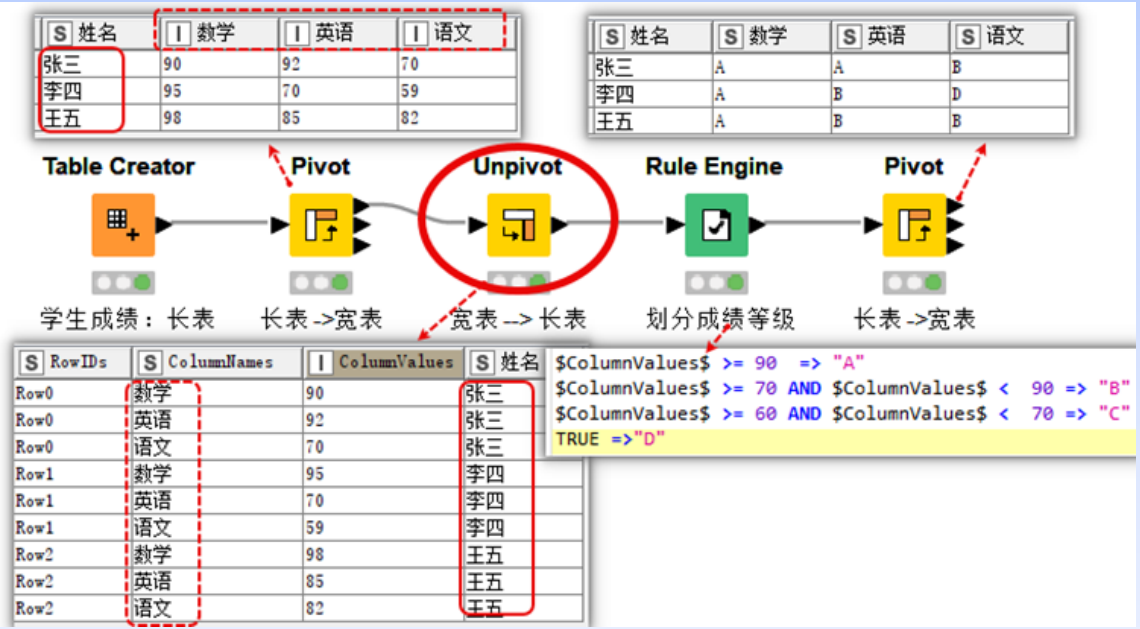
06

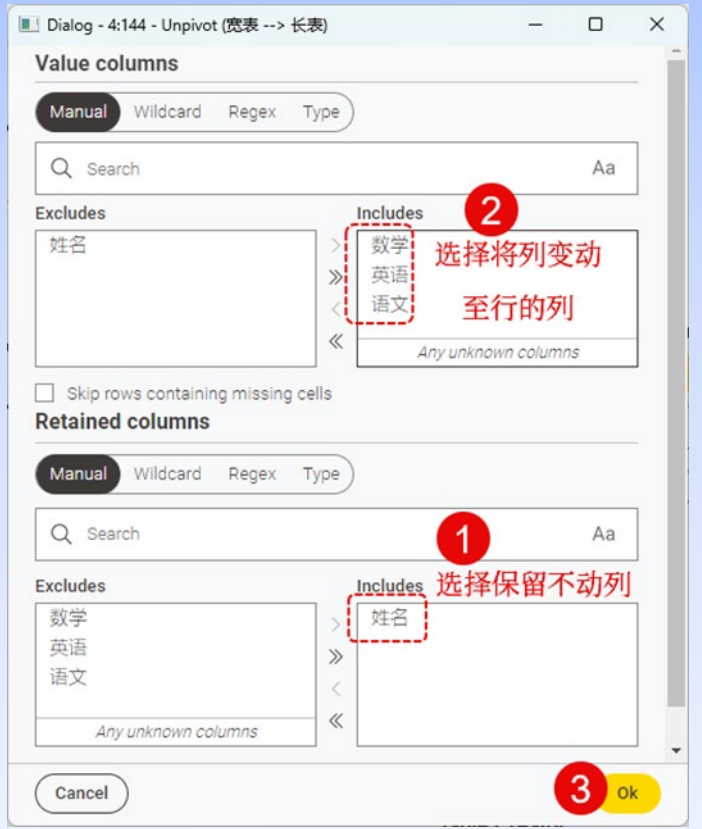
表格结构重塑

常用节点概览

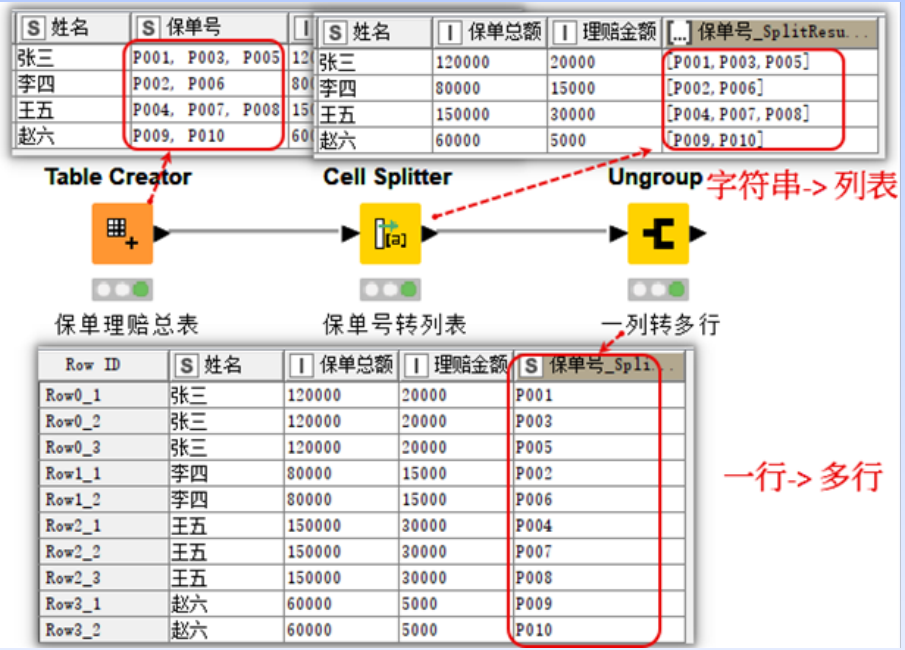

数据重塑常用节点
Pivot
Unpivot
Ungroup
Table Transposer
Column to Grid
One Row to Many
One to Many
Many to One

长表与宽表互转


S 姓名 | I 数学 | I 英语 | I 语文
张三 90 92 70
李四 95 70 59
王五 98 85 82
S 姓名 | S 数学 | S 英语 | S 语文
张三 A A B
李四 A B D
王五 A B B
Table Creator
Pivot
Unpivot
Rule Engine
Pivot
学生成绩：长表
长表->宽表
宽表->长表
划分成绩等级
长表->宽表
S RowIDs | S ColumnNames | I ColumnValues | S 姓名
Row0 数学 90 张三
Row0 英语 92 张三
Row0 语文 70 张三
Row1 数学 95 李四
Row1 英语 70 李四
Row1 语文 59 李四
Row2 数学 98 王五
Row2 英语 85 王五
Row2 语文 82 王五
$ColumnValues$ >= 90 => "A"
$ColumnValues$ >= 70 AND $ColumnValues$ < 90 => "B"
$ColumnValues$ >= 60 AND $ColumnValues$ < 70 => "C"
TRUE =>"D"


Dialog - 4:144 - Unpivot (宽表 --> 长表)
Value columns
Manual
Wildcard
Regex
Type
Search
Aa
Excludes
姓名
Includes
数学
英语
语文
2
选择将列变动至行的列
Any unknown columns
Skip rows containing missing cells
Retained columns
Manual
Wildcard
Regex
Type
Search
Aa
1
选择保留不动列
Excludes
数学
英语
语文
Includes
姓名
Any unknown columns
Cancel
3
Ok

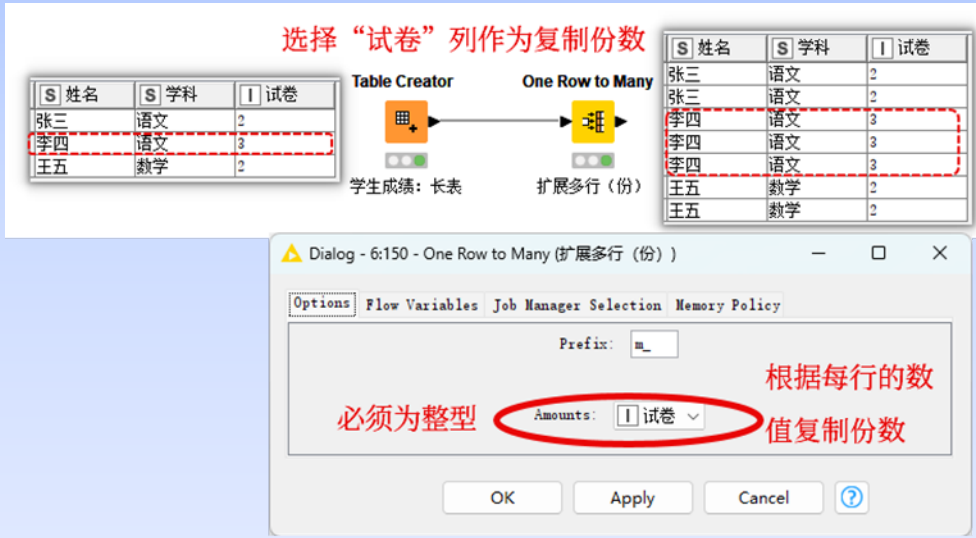
一行与多行互转



姓名	保单号
张三	P001, P003, P005
李四	P002, P006
王五	P004, P007, P008
赵六	P009, P010

姓名	保单总额	理赔金额	保单号_SplitResu...
张三	120000	20000	[P001, P003, P005]
李四	80000	15000	[P002, P006]
王五	150000	30000	[P004, P007, P008]
赵六	60000	5000	[P009, P010]

Row ID	姓名	保单总额	理赔金额	保单号_Spli...
Row0_1	张三	120000	20000	P001
Row0_2	张三	120000	20000	P003
Row0_3	张三	120000	20000	P005
Row1_1	李四	80000	15000	P002
Row1_2	李四	80000	15000	P006
Row2_1	王五	150000	30000	P004
Row2_2	王五	150000	30000	P007
Row2_3	王五	150000	30000	P008
Row3_1	赵六	60000	5000	P009
Row3_2	赵六	60000	5000	P010



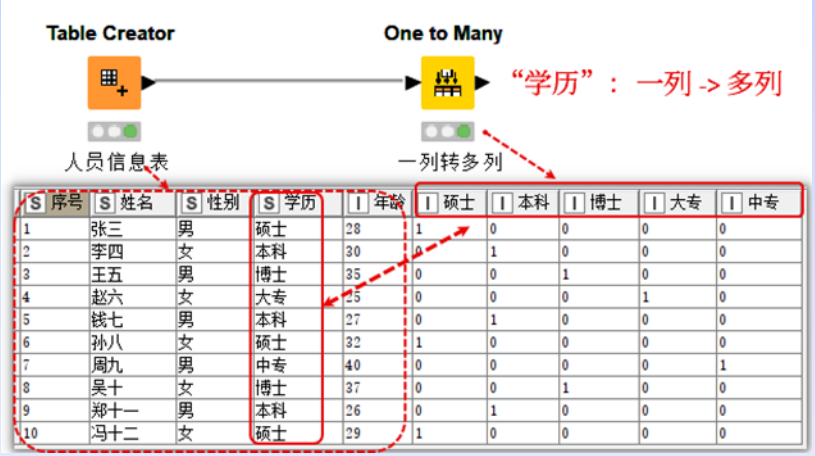
姓名	学科	试卷
张三	语文	2
李四	语文	3
王五	数学	2

姓名	学科	试卷
张三	语文	2
张三	语文	2
李四	语文	3
李四	语文	3
李四	语文	3
王五	数学	2
王五	数学	2

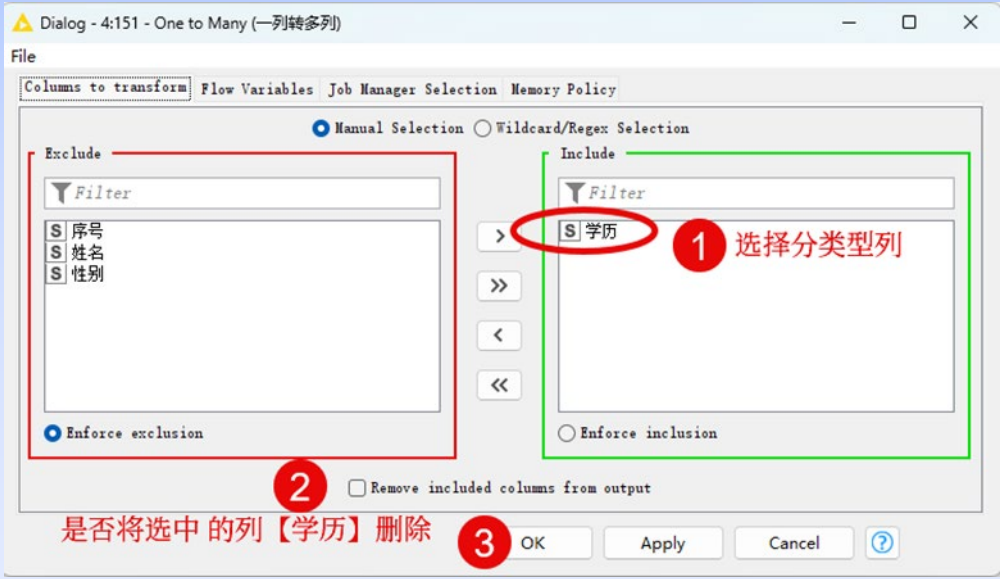
Regex Extractor正则节点也可以实现同样的操作，第12章展开。

一列与多列互转

一列转多列方法



S 序号	S 姓名	S 性别	S 学历	I 年龄	I 硕士	I 本科	I 博士	I 大专	I 中专
1	张三	男	硕士	28	1	0	0	0	0
2	李四	女	本科	30	0	1	0	0	0
3	王五	男	博士	35	0	0	1	0	0
4	赵六	女	大专	25	0	0	0	1	0
5	钱七	男	本科	27	0	1	0	0	0
6	孙八	女	硕士	32	1	0	0	0	0
7	周九	男	中专	40	0	0	0	0	1
8	吴十	女	博士	37	0	0	1	0	0
9	郑十一	男	本科	26	0	1	0	0	0
10	冯十二	女	硕士	29	1	0	0	0	0



多列转一列方法：使用Many to One还原（略）

一列转多列

Cell Splitter节点

同Excel中的分列功能，基于分隔符，将字符串拆分为多列

Cell Splitter By Position

一般用于等宽字符串，内部不存在分隔符时，对字符串进行拆分

Regex Split

利用正则表达式捕获组分割字符串，得到多列

Regex Extractor

利用正则表达式捕获组将匹配结果拆分为多列

Split Collection Column

将列表或集合类型数据拆分为多列

多列转一列

Column Combiner节点

同Excel中的Concatenate函数（不要与KNIME中的Concatenate节点混淆），将多列合并为一列字符串类型结果

String Manipulation和Column Expressions节点中的join函数

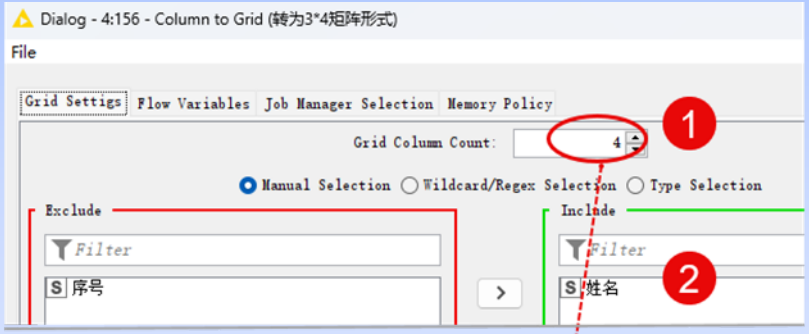
若选择的列非字符串数据类型，需要使用string()函数转为字符串类型，否则无法合并

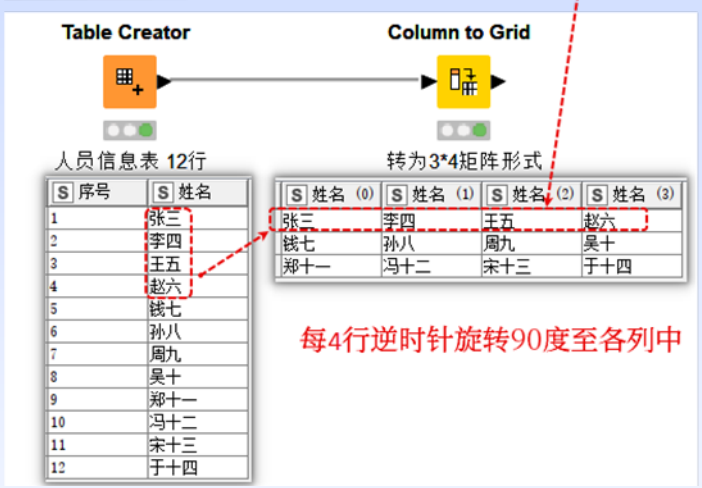
Create Collection Column

将多列合并为单独的一列，该列数据类型为列表类型。

其他Java、Python类合并函数

多行转多列

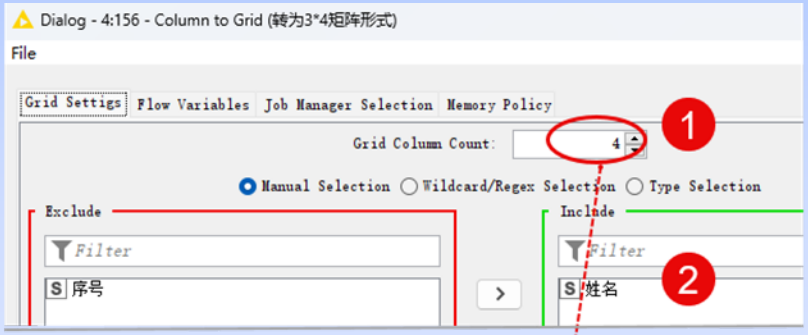


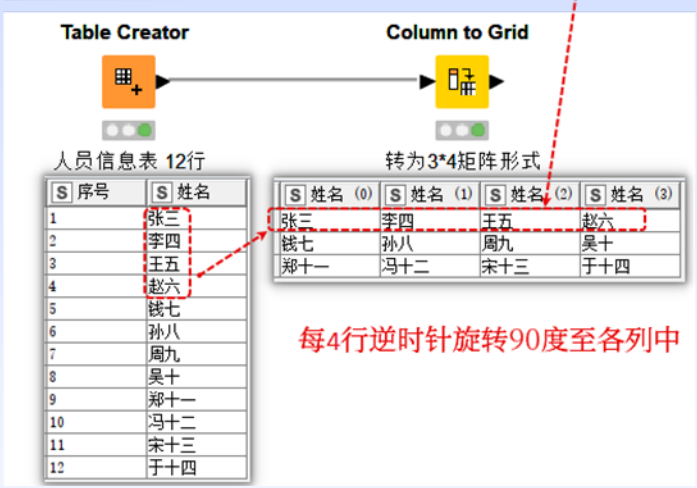


S 序号	S 姓名
1	张三
2	李四
3	王五
4	赵六
5	钱七
6	孙八
7	周九
8	吴十
9	郑十一
10	冯十二
11	宋十三
12	于十四

S 姓名 (0)	S 姓名 (1)	S 姓名 (2)	S 姓名 (3)
张三	李四	王五	赵六
钱七	孙八	周九	吴十
郑十一	冯十二	宋十三	于十四

多列转多行



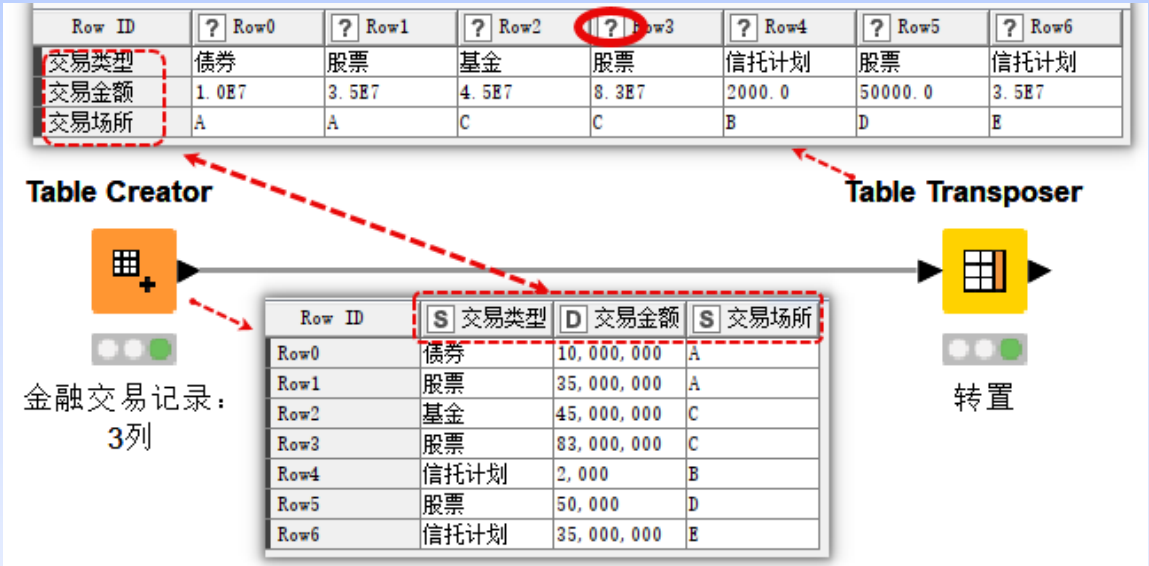


练习：

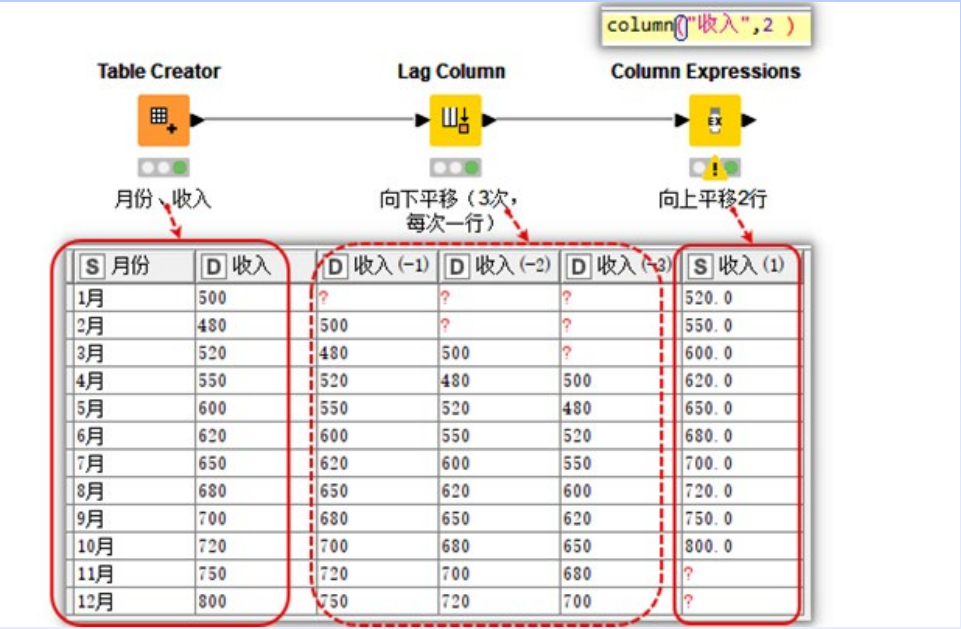
将左侧Column to Grid结果还原至原始数据

(详见6.6.6节提示：使用Unpivot节点)

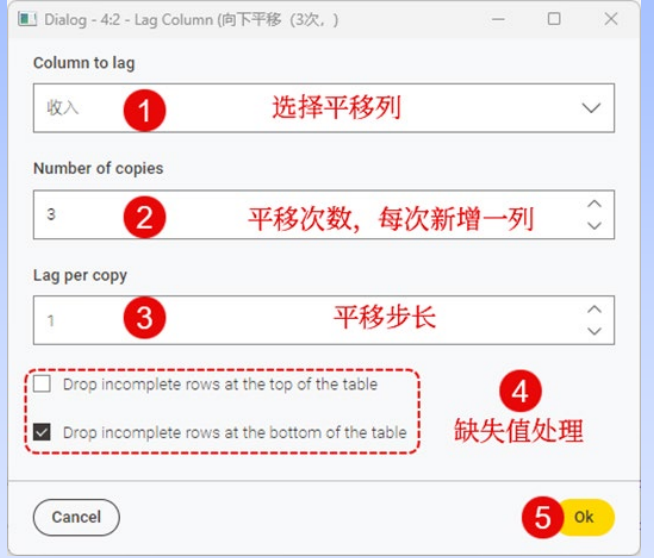
行列转置

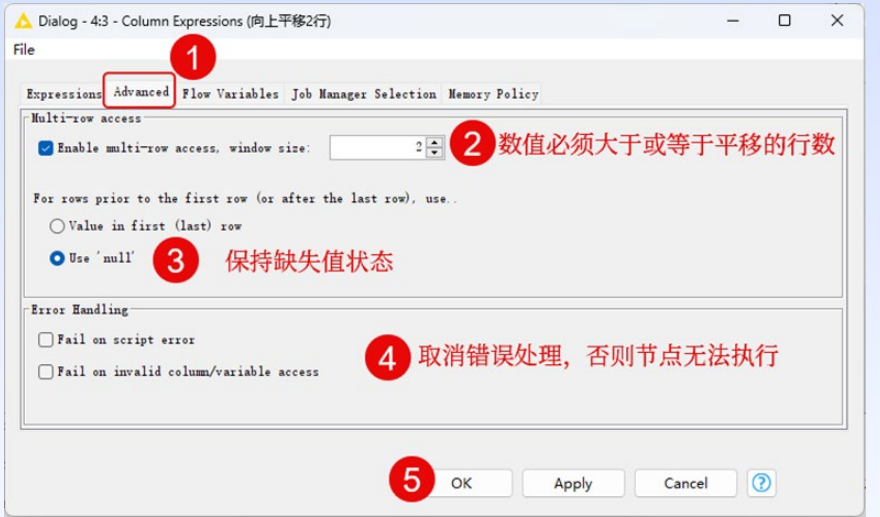

Row ID
Row0
Row1
Row2
Row3
Row4
Row5
Row6
交易类型
债券
股票
基金
股票
信托计划
股票
信托计划
交易金额
1.0E7
3.5E7
4.5E7
8.3E7
2000.0
50000.0
3.5E7
交易场所
A
A
C
C
B
D
E
Table Creator
Table Transposer
金融交易记录：
3列
转置
Row ID
交易类型
交易金额
交易场所
Row0
债券
10,000,000
A
Row1
股票
35,000,000
A
Row2
基金
45,000,000
C
Row3
股票
83,000,000
C
Row4
信托计划
2,000
B
Row5
股票
50,000
D
Row6
信托计划
35,000,000
E

列平移



S 月份	D 收入	D 收入 (-1)	D 收入 (-2)	D 收入 (-3)	S 收入 (1)
1月	500	?	?	?	520.0
2月	480	500	?	?	550.0
3月	520	480	500	?	600.0
4月	550	520	480	500	620.0
5月	600	550	520	480	650.0
6月	620	600	550	520	680.0
7月	650	620	600	550	700.0
8月	680	650	620	600	720.0
9月	700	680	650	620	750.0
10月	720	700	680	650	800.0
11月	750	720	700	680	?
12月	800	750	720	700	?





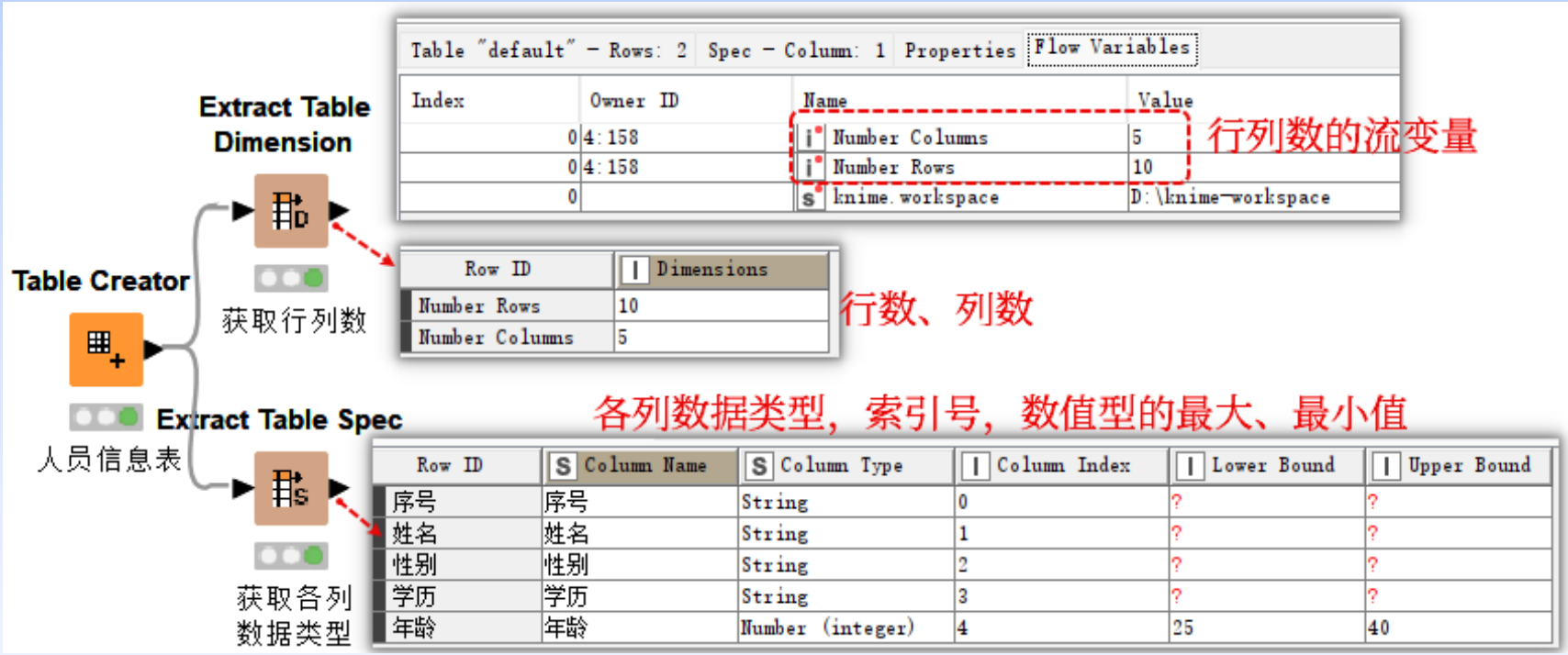
5.3版本新增Expression节点，上下平移列的操作也很方便



07

获取表格属性

获取表格属性


Extract Table Dimension
Table Creator
获取行列数
人员信息表
Extract Table Spec
获取各列数据类型
Table "default" - Rows: 2 Spec - Column: 1 Properties Flow Variables
Index
Owner ID
Name
Value
0
4:158
Number Columns
5
Number Rows
10
knime.workspace
D:\knime-workspace
行列数的流变量
Row ID
Dimensions
Number Rows
10
Number Columns
5
行数、列数
各列数据类型，索引号，数值型的最大、最小值
Row ID
Column Name
Column Type
Column Index
Lower Bound
Upper Bound
序号
String
姓名
性别
学历
年龄
Number (integer)
25
40



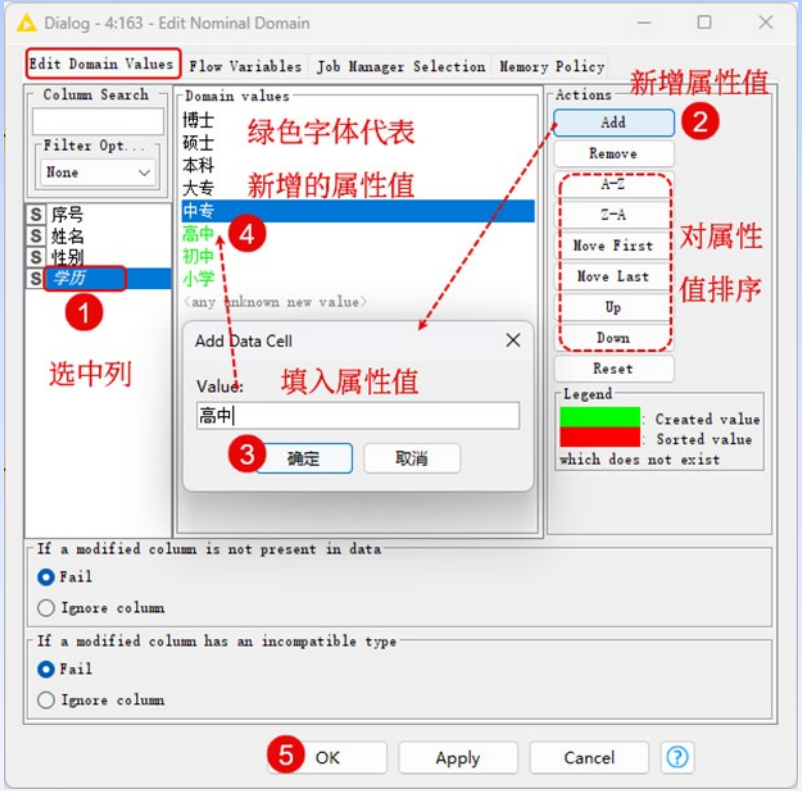
08

修改表格属性

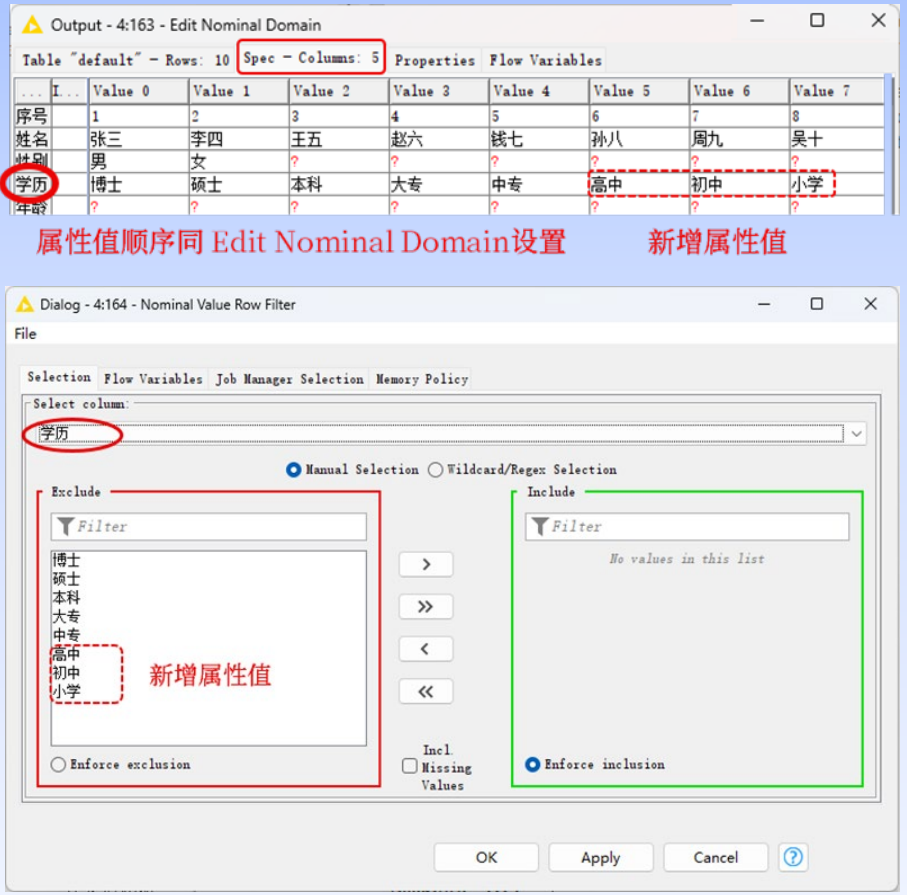
修改属性值常用节点


Edit Nominal Domain
Edit Nominal Domain (Dictionary)
Domain Calculator
Edit Numeric Domain
（a）分类性（默认列的属性值个数不超过60）
（b）数值型

分类型属性值缺失



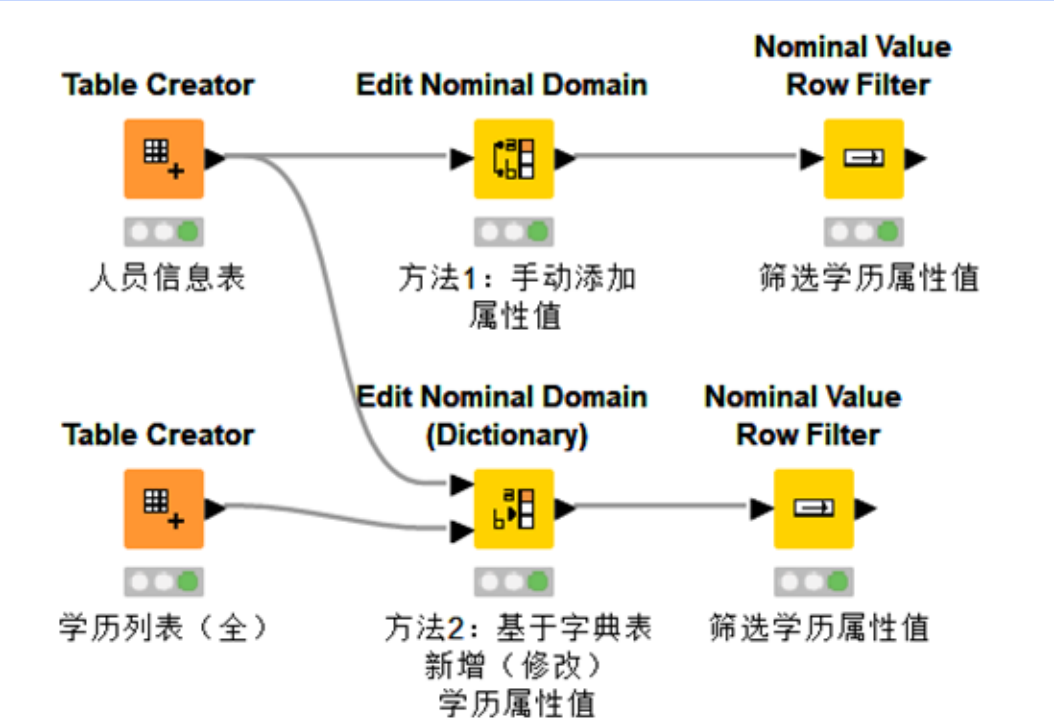
增加属性值



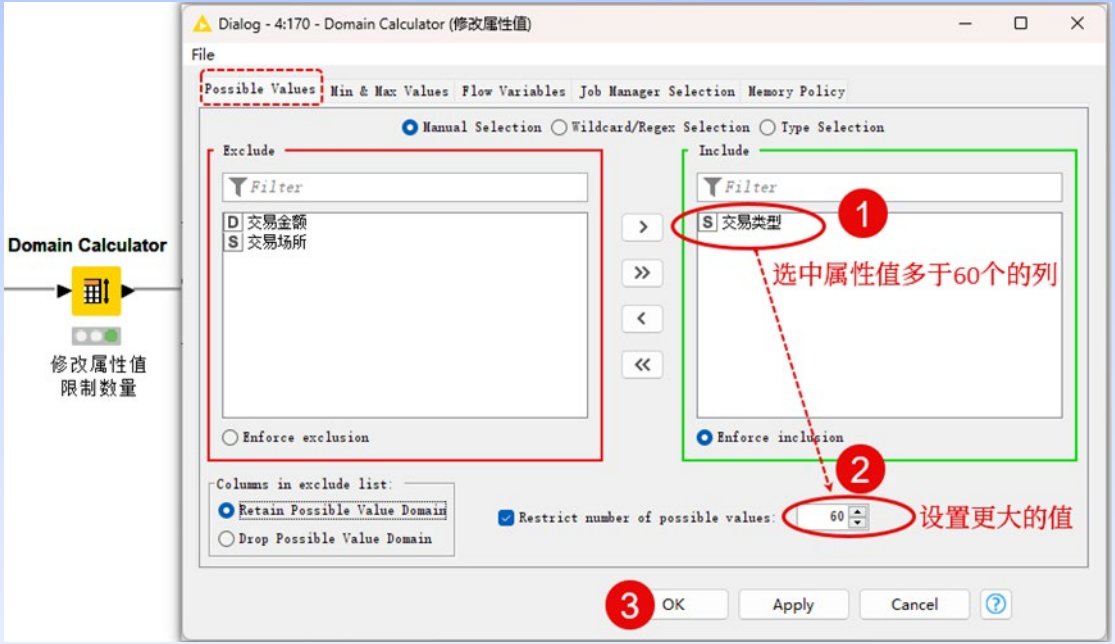
增加属性值后，结果变化、及对其他节点的影响

分类型属性值缺失

增加属性值 工作流

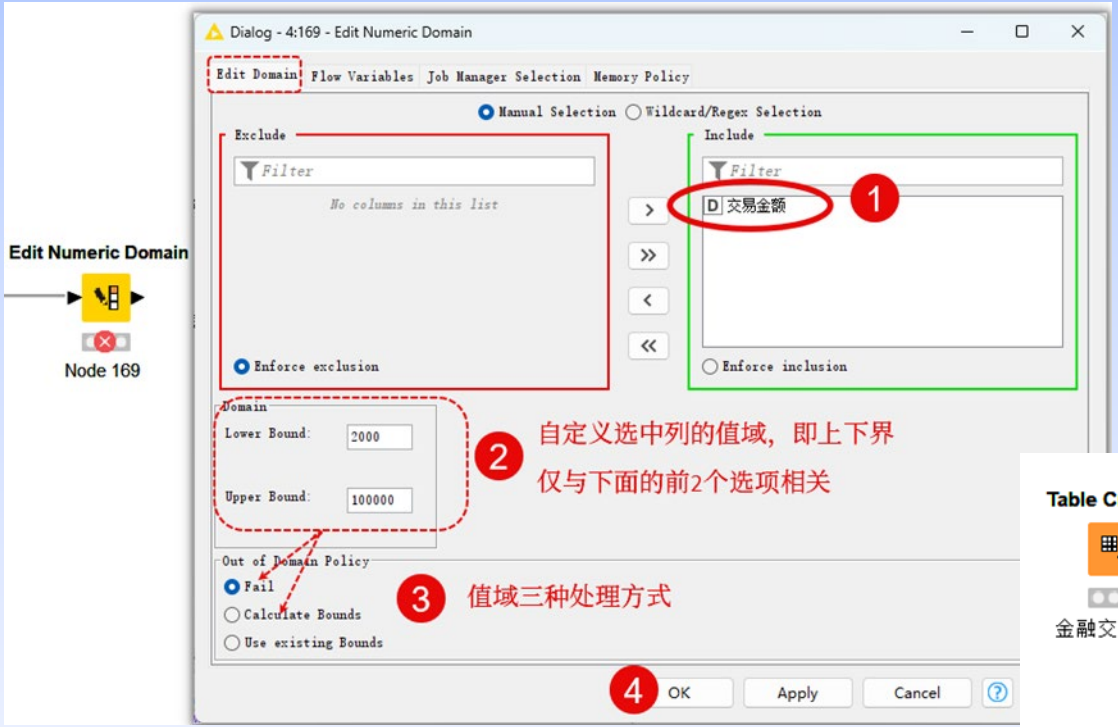


分类型属性值过多

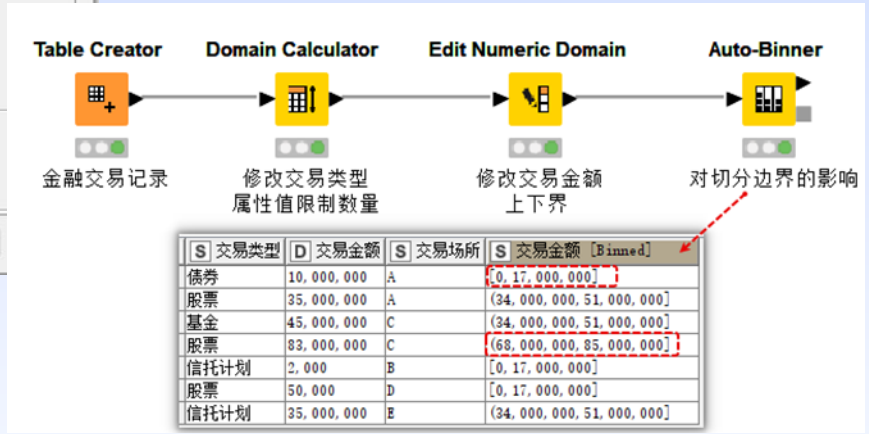


对于分类型属性过多的数据集，该列属性值列表下拉菜单或列表中不显示。

修改数值型上下界



对其他节点的影响



S 交易类型	D 交易金额	S 交易场所	S 交易金额 [Binned]
债券	10,000,000	A	[0,17,000,000]
股票	35,000,000	A	(34,000,000,51,000,000]
基金	45,000,000	C	(34,000,000,51,000,000]
股票	83,000,000	C	(68,000,000,85,000,000]
信托计划	2,000	B	[0,17,000,000]
股票	50,000	D	[0,17,000,000]
信托计划	35,000,000	E	(34,000,000,51,000,000]



09

验证表格属性

验证表格属性

常用节点

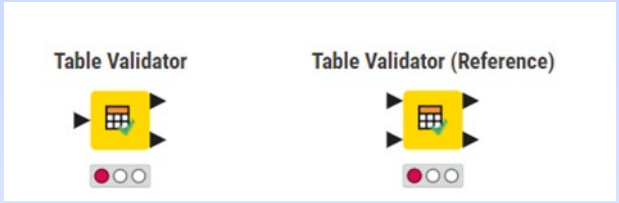
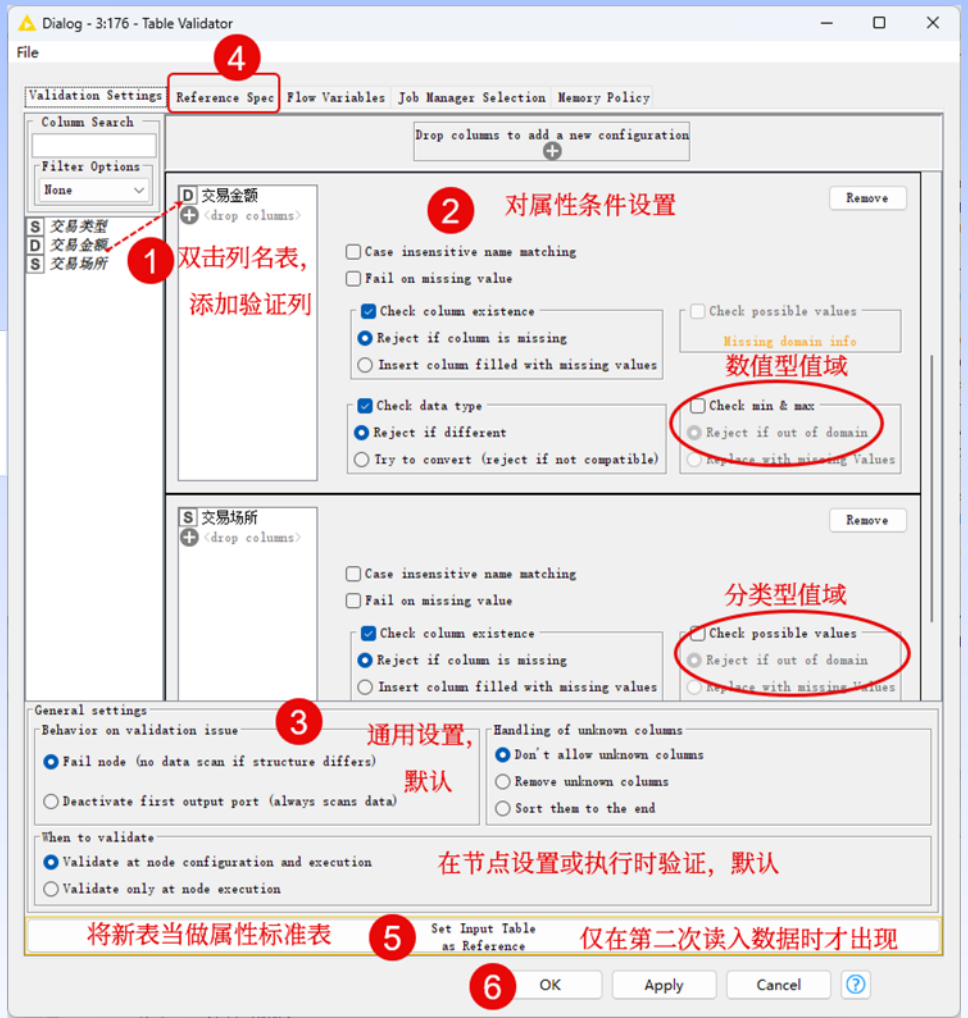
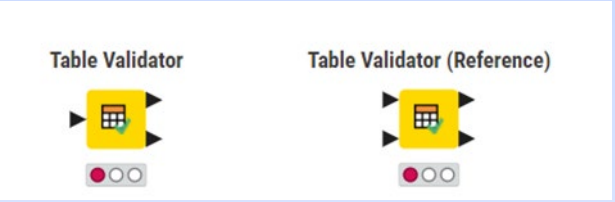

Table Validator
Table Validator (Reference)

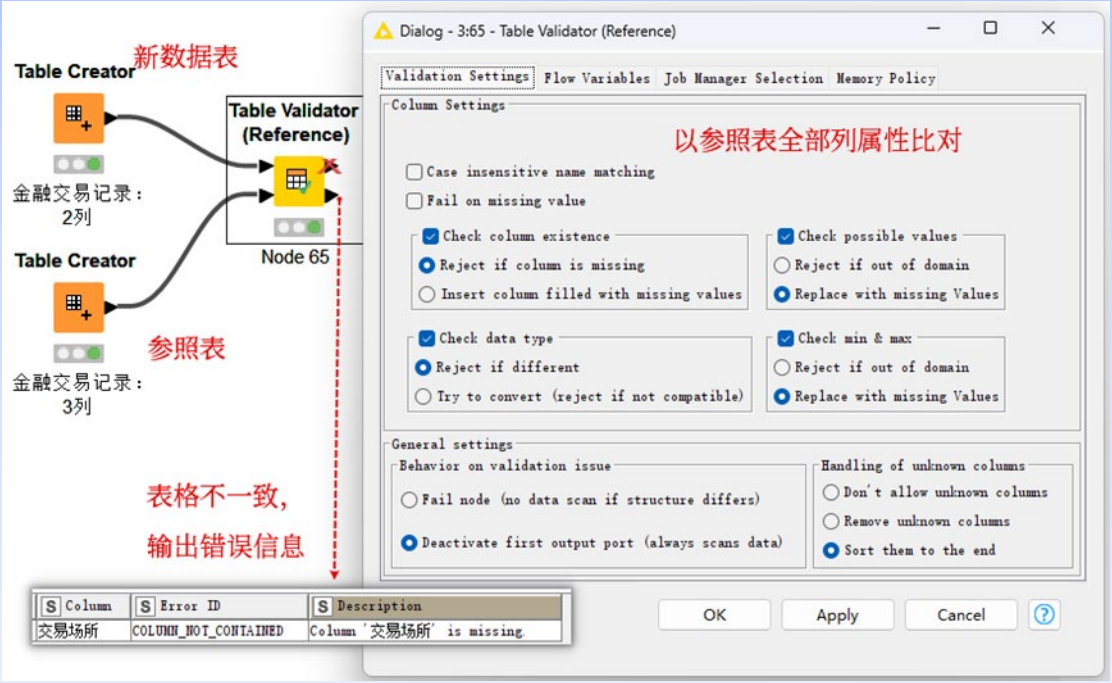

Table Validator
数据属性验证


Dialog - 3:176 - Table Validator
File
4
Validation Settings
Reference Spec
Flow Variables
Job Manager Selection
Memory Policy
Column Search
Filter Options
None
Drop columns to add a new configuration
D 交易金额
S 交易类型
D 交易金额
S 交易场所
1 双击列名表，添加验证列
2 对属性条件设置
Remove
Case insensitive name matching
Fail on missing value
Check column existence
Reject if column is missing
Insert column filled with missing values
Check possible values
Missing domain info
数值型值域
Check data type
Reject if different
Try to convert (reject if not compatible)
Check min & max
Reject if out of domain
S 交易场所
分类型值域
3 通用设置，默认
General settings
Behavior on validation issue
Fail node (no data scan if structure differs)
Deactivate first output port (always scans data)
Handling of unknown columns
Don't allow unknown columns
Remove unknown columns
Sort them to the end
When to validate
Validate at node configuration and execution
Validate only at node execution
在节点设置或执行时验证，默认
将新表当做属性标准表
5
Set Input Table as Reference
仅在第二次读入数据时才出现
6
OK
Apply
Cancel

验证表格属性

常用节点


Table Validator
Table Validator (Reference)

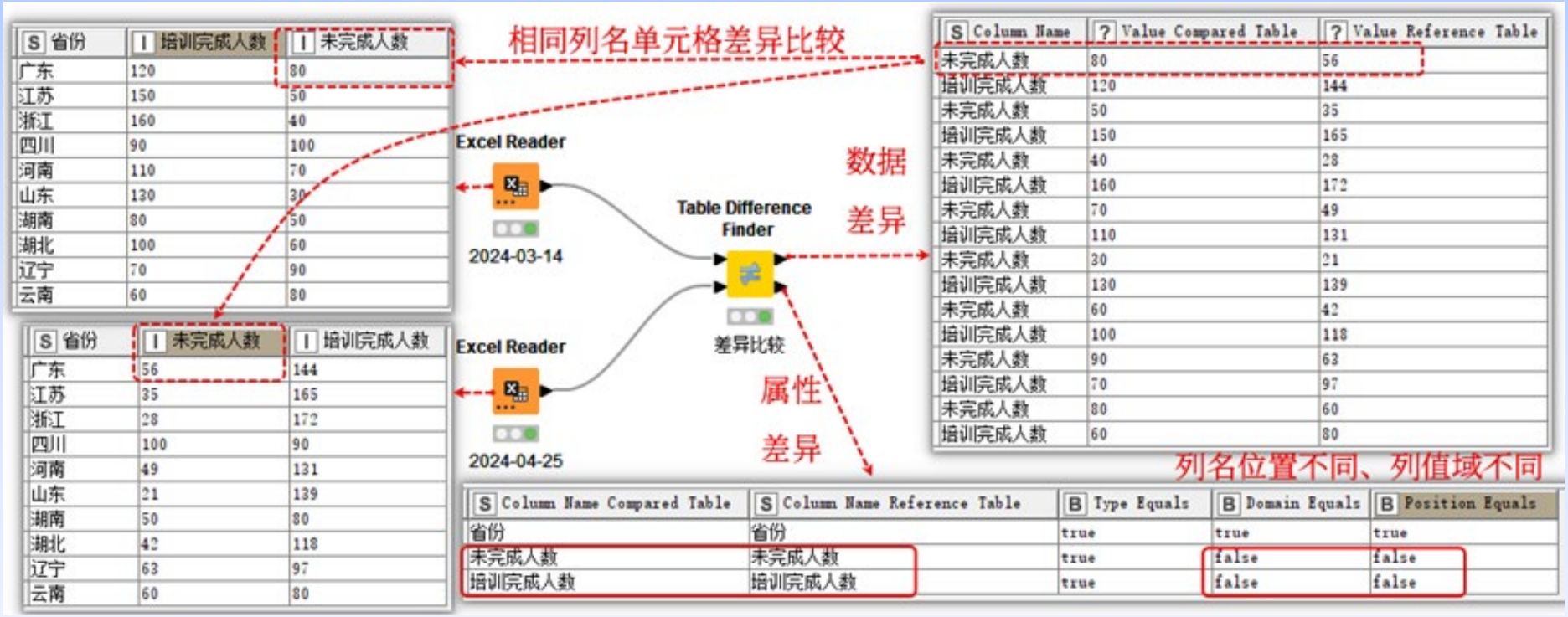

Table Creator
新数据表
金融交易记录：
2列
Table Creator
参照表
金融交易记录：
3列
Table Validator
(Reference)
Node 65
表格不一致，
输出错误信息
Column
Error ID
Description
交易场所
COLUMN_NOT_CONTAINED
Column '交易场所' is missing.
Dialog - 3:65 - Table Validator (Reference)
Validation Settings
Flow Variables
Job Manager Selection
Memory Policy
Column Settings
以参照表全部列属性比对
Case insensitive name matching
Fail on missing value
Check column existence
Reject if column is missing
Insert column filled with missing values
Check possible values
Reject if out of domain
Replace with missing Values
Check data type
Reject if different
Try to convert (reject if not compatible)
Check min & max
Reject if out of domain
Replace with missing Values
General settings
Behavior on validation issue
Fail node (no data scan if structure differs)
Deactivate first output port (always scans data)
Handling of unknown columns
Don't allow unknown columns
Remove unknown columns
Sort them to the end
OK
Apply
Cancel



10

比较表格属性或数据差异

比较表格差异（单元格数据差异，属性差异等）



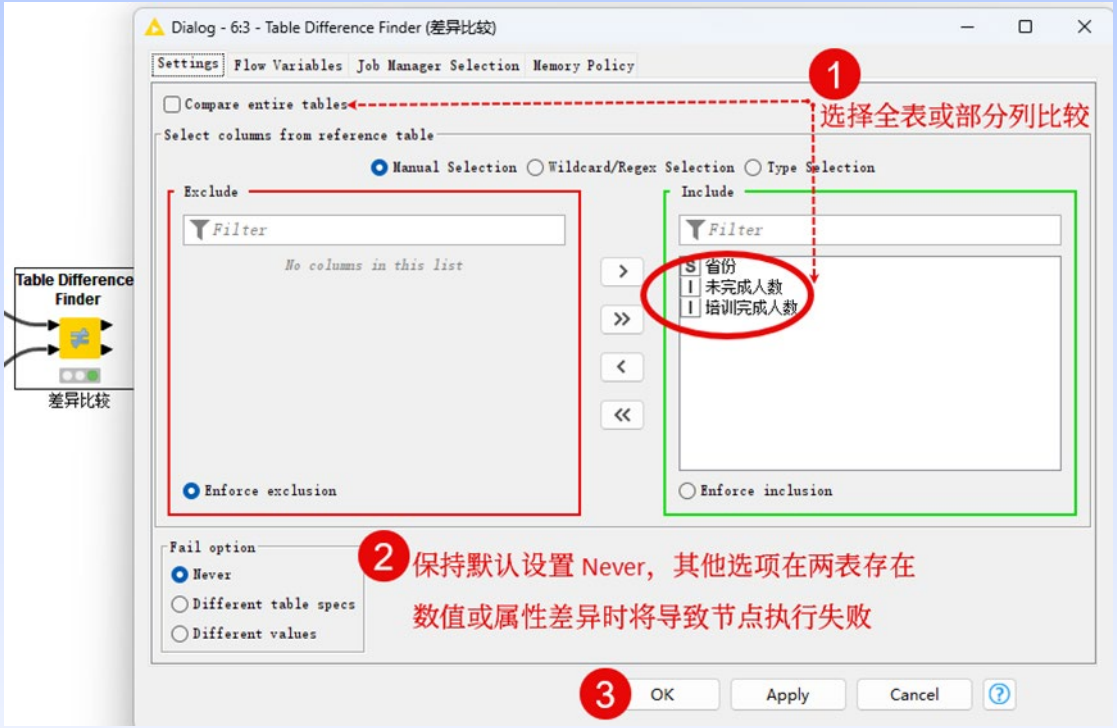
省份	培训完成人数	未完成人数
广东	120	80
江苏	150	50
浙江	160	40
四川	90	100
河南	110	70
山东	130	30
湖南	80	50
湖北	100	60
辽宁	70	90
云南	60	80

省份	未完成人数	培训完成人数
广东	56	144
江苏	35	165
浙江	28	172
四川	100	90
河南	49	131
山东	21	139
湖南	50	80
湖北	42	118
辽宁	63	97
云南	60	80

Column Name	Value Compared Table	Value Reference Table
未完成人数	80	56
培训完成人数	120	144
未完成人数	50	35
培训完成人数	150	165
未完成人数	40	28
培训完成人数	160	172
未完成人数	70	49
培训完成人数	110	131
未完成人数	30	21
培训完成人数	130	139
未完成人数	60	42
培训完成人数	100	118
未完成人数	90	63
培训完成人数	70	97
未完成人数	80	60
培训完成人数	60	80

Column Name Compared Table	Column Name Reference Table	Type Equals	Domain Equals	Position Equals
省份	省份	true	true	true
未完成人数	未完成人数	true	false	false
培训完成人数	培训完成人数	true	false	false

比较表格差异（单元格数据差异，属性差异等）

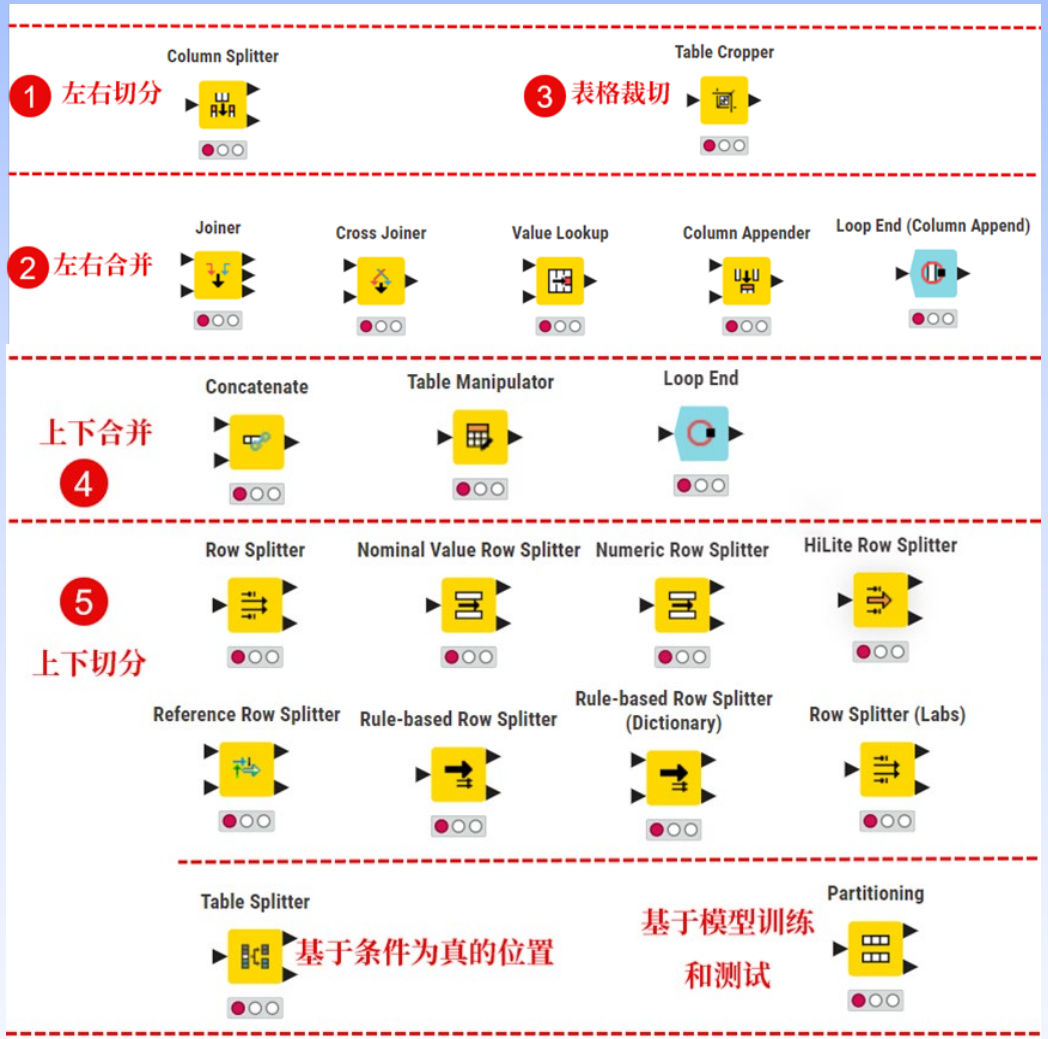




11

本章小结

本章节点总结


1 左右切分
Column Splitter
3 表格裁切
Table Cropper
2 左右合并
Joiner
Cross Joiner
Value Lookup
Column Appender
Loop End (Column Append)
上下合并
4
Concatenate
Table Manipulator
Loop End
5
上下切分
Row Splitter
Nominal Value Row Splitter
Numeric Row Splitter
HiLite Row Splitter
Reference Row Splitter
Rule-based Row Splitter
Rule-based Row Splitter (Dictionary)
Row Splitter (Labs)
Table Splitter
基于条件为真的位置
Partitioning
基于模型训练和测试

本章节点总结


6
表格结构
重塑
Pivot
Unpivot
Ungroup
Table Transposer
Column to Grid
One Row to Many
One to Many
Many to One
7
获取表格属性
Extract Table Dimension
Extract Table Spec
8
创建表格属性
Table Structure Creator
9
修改表格属性
Edit Nominal Domain
Edit Nominal Domain (Dictionary)
Domain Calculator
Edit Numeric Domain
Table Updater
10
验证表格属性
Table Validator
Table Validator (Reference)
11
比较表格属性
及数据差异
Table Difference Finder

感谢观看

THANKS



第七章 数据汇总

数据汇总节点全面总结



目录

01 汇总三要素

02 GroupBy节点应用

03 Pivot汇总节点应用

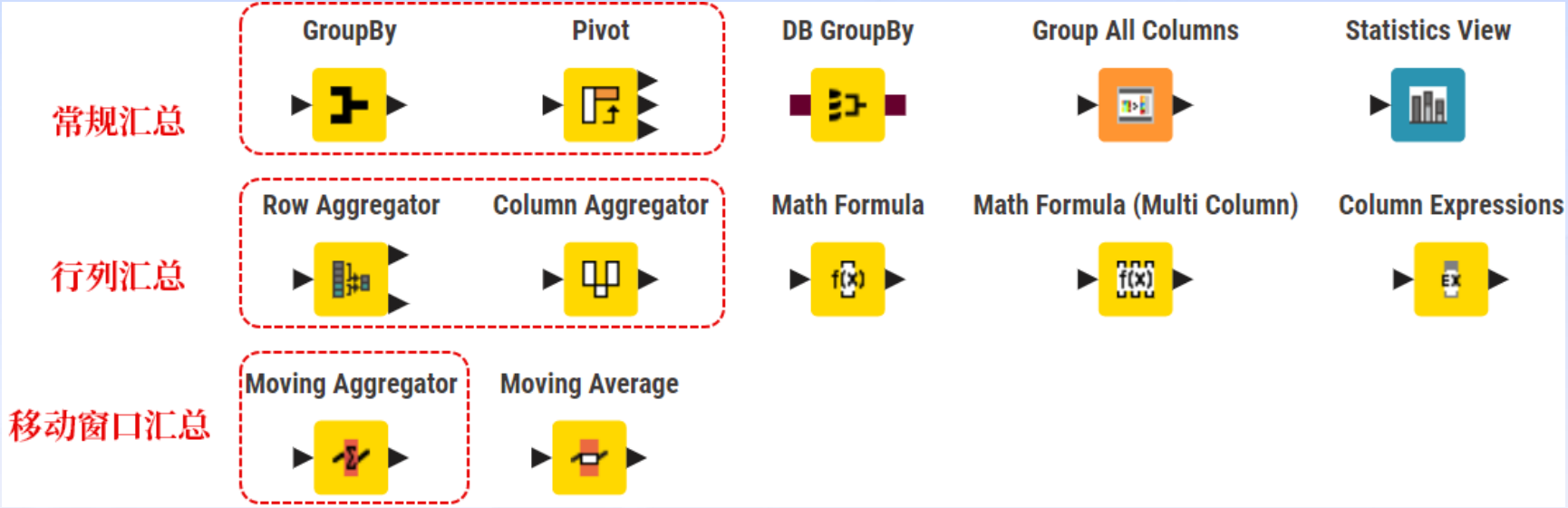
04 行列汇总方法

05 移动窗口汇总技术

06 集合类型数据聚合方法

07 本章小结

常用节点


常规汇总
GroupBy
Pivot
DB GroupBy
Group All Columns
Statistics View
行列汇总
Row Aggregator
Column Aggregator
Math Formula
Math Formula (Multi Column)
Column Expressions
移动窗口汇总
Moving Aggregator
Moving Average



01

汇总三要素

汇总三要素

01 分组维度

按什么维度进行汇总

数据汇总中的分类字段是用来对数据进行排序和分组的重要依据。通过设定合理的分类字段，可以确保数据在汇总过程中能够按照预期的类别进行整理，从而便于后续的分析与操作。

02 汇总列的选择

要对什么列进行汇总

KNIME中一般是以列为主要对象做统计，如以销量或金额的列进行汇总或统计。当然，个别节点以行为对象进行统计

03 汇总方法

对汇总列使用什么汇总类方法

汇总方式决定了数据在汇总过程中采用的计算方法，包括求和、计数、平均值等。选择合适的汇总方式能够有效地展示数据集中的核心趋势和特征，为决策提供有力支持。

注意事项

1、要素1可单独使用，相当于对选择的单列去重（单列）、或对多列的唯一组合去重

2、要素2和要素3需要一起使用



02

GroupBy节点应用

GroupBy基本用法

GroupBy核心概念

GroupBy节点用于对数据进行分组，通过指定的键将数据集分割成多个子集。每个分组独立处理，常用于统计、聚合和分析数据，提升数据处理效率。

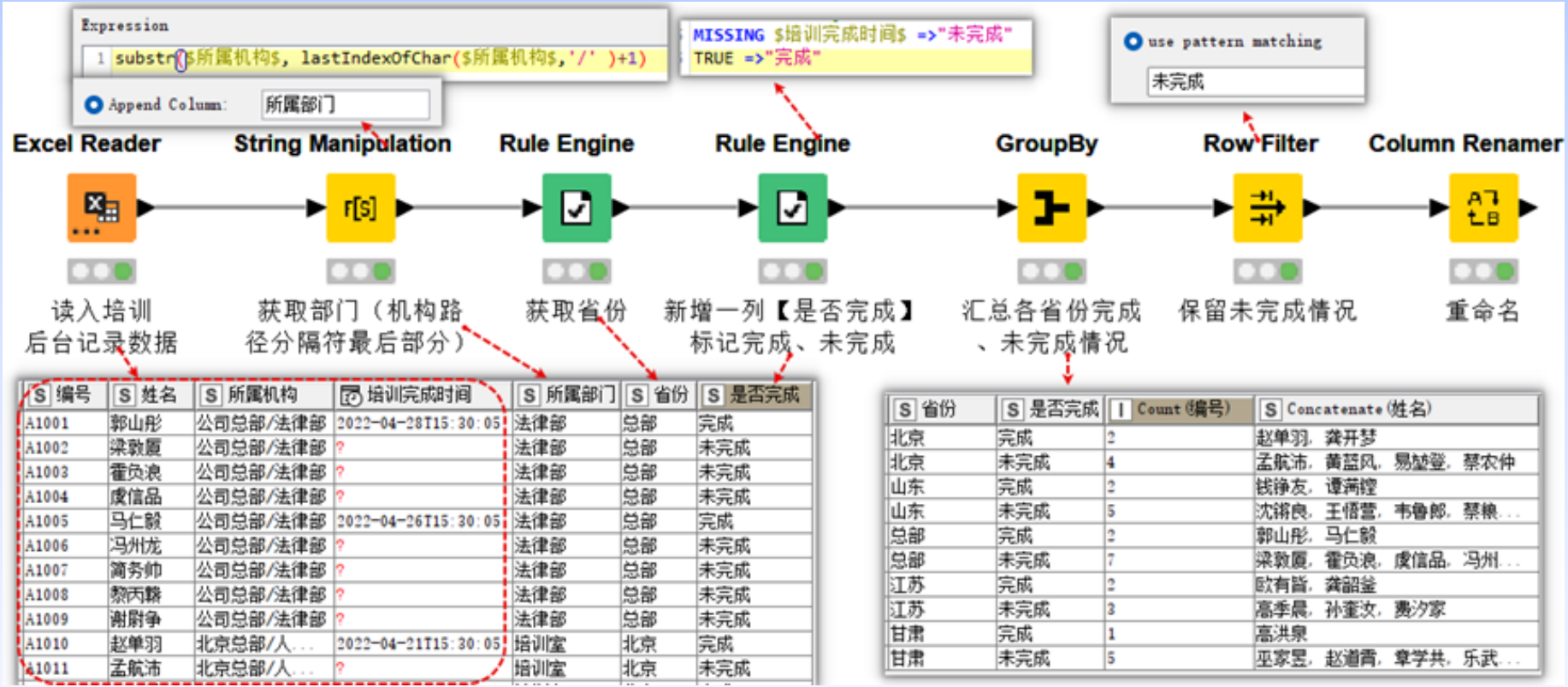
GroupBy与聚合函数结合使用

GroupBy常与聚合函数如sum、mean、count等结合使用，以实现数据的汇总统计。通过定义聚合函数，可以针对每个分组计算特定统计值，提供详细的分组数据汇总。

GroupBy应用场景

GroupBy广泛应用于数据统计、数据分析和数据可视化中。例如，可以按日期、地区或产品类别分组，分别计算各组的总和、平均值或其他统计指标。

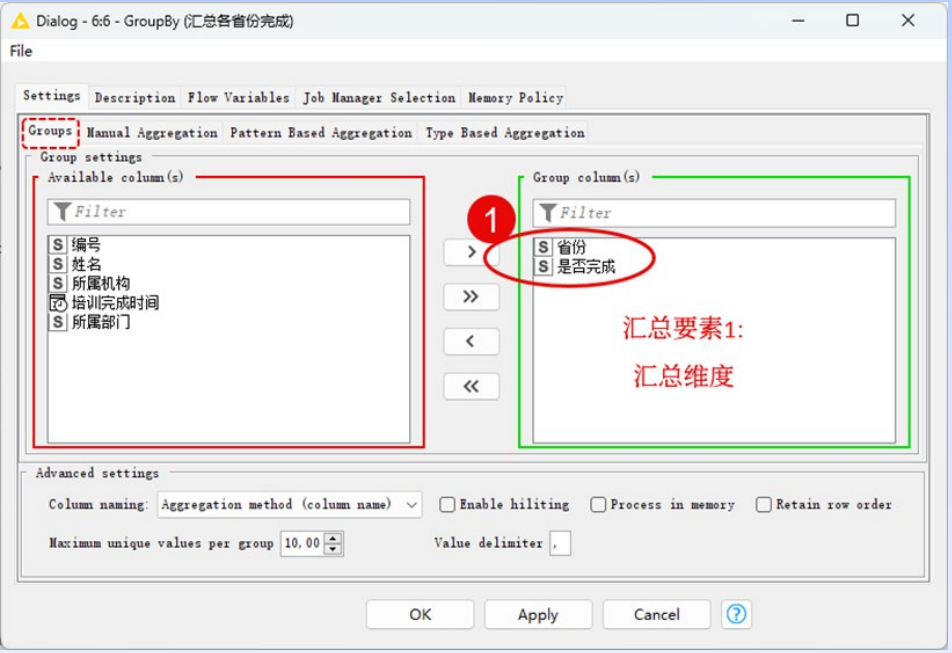
GroupBy节点应用案例1：员工在线培训完成情况

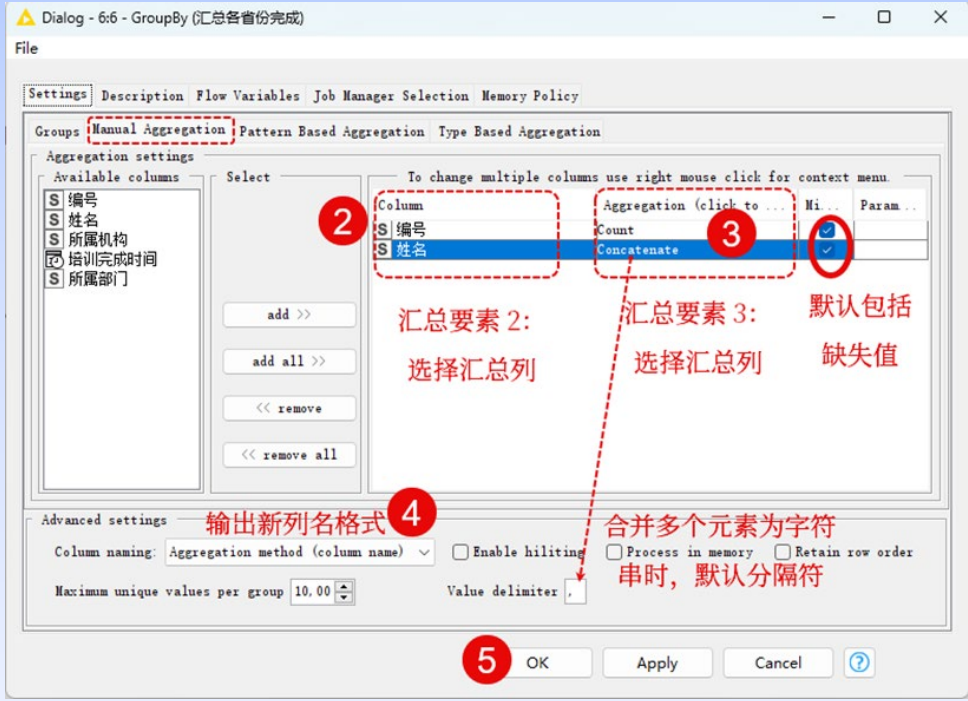


编号	姓名	所属机构	培训完成时间	所属部门	省份	是否完成
A1001	郭山彤	公司总部/法律部	2022-04-28T15:30:05	法律部	总部	完成
A1002	梁敦厦	公司总部/法律部	?	法律部	总部	未完成
A1003	霍负浪	公司总部/法律部	?	法律部	总部	未完成
A1004	虞信品	公司总部/法律部	?	法律部	总部	未完成
A1005	马仁毅	公司总部/法律部	2022-04-26T15:30:05	法律部	总部	完成
A1006	冯州龙	公司总部/法律部	?	法律部	总部	未完成
A1007	简务帅	公司总部/法律部	?	法律部	总部	未完成
A1008	黎丙辣	公司总部/法律部	?	法律部	总部	未完成
A1009	谢尉争	公司总部/法律部	?	法律部	总部	未完成
A1010	赵单羽	北京总部/人...	2022-04-21T15:30:05	培训室	北京	完成
A1011	孟航沛	北京总部/人...	?	培训室	北京	未完成

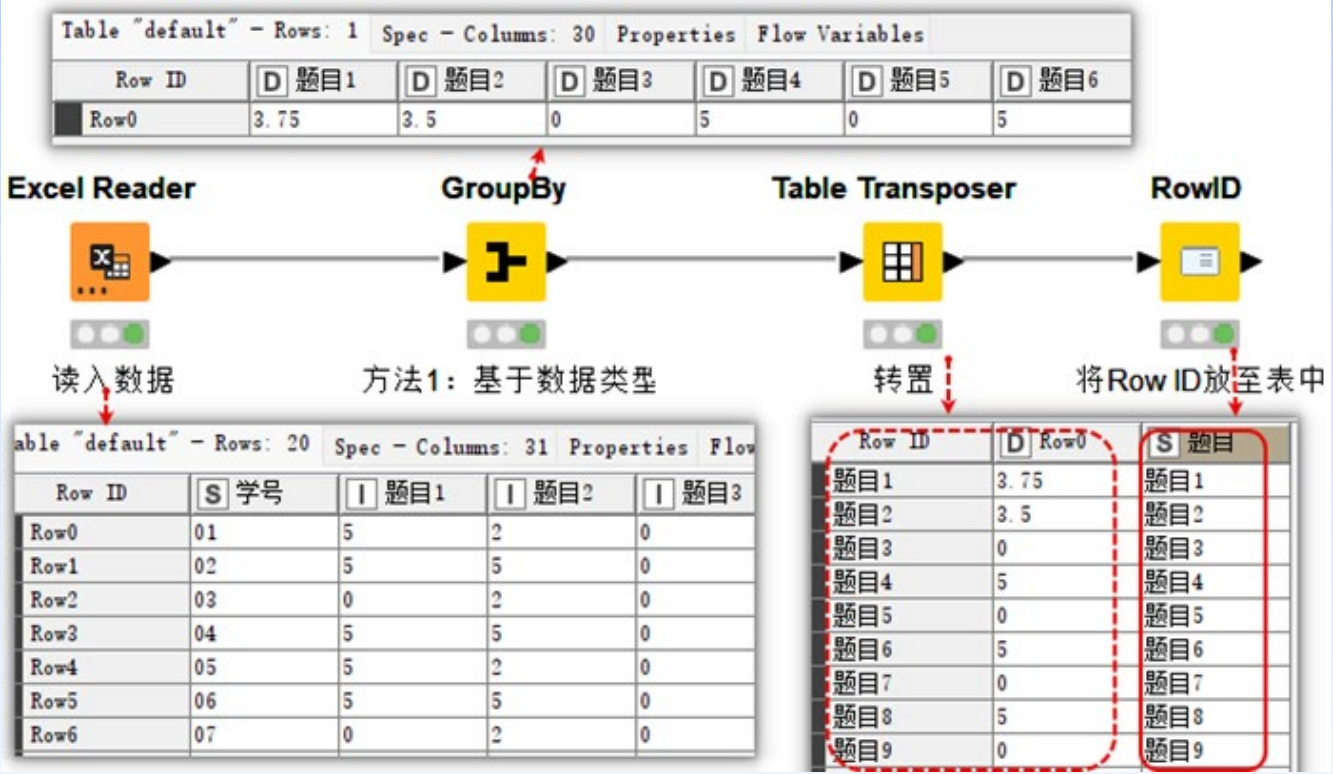
省份	是否完成	Count(编号)	Concatenate(姓名)
北京	完成	2	赵单羽，龚开梦
北京	未完成	4	孟航沛，黄蓝风，易楚登，蔡农仲
山东	完成	2	钱铮友，谭满瞪
山东	未完成	5	沈锵良，王悟营，韦鲁郎，蔡粮...
总部	完成	2	郭山彤，马仁毅
总部	未完成	7	梁敦厦，霍负浪，虞信品，冯州...
江苏	完成	2	欧有皆，龚韶釜
江苏	未完成	3	高季晨，孙奎汝，费汐家
甘肃	完成	1	高洪泉
甘肃	未完成	5	巫家昱，赵道霄，章学共，乐武...

GroupBy节点设置





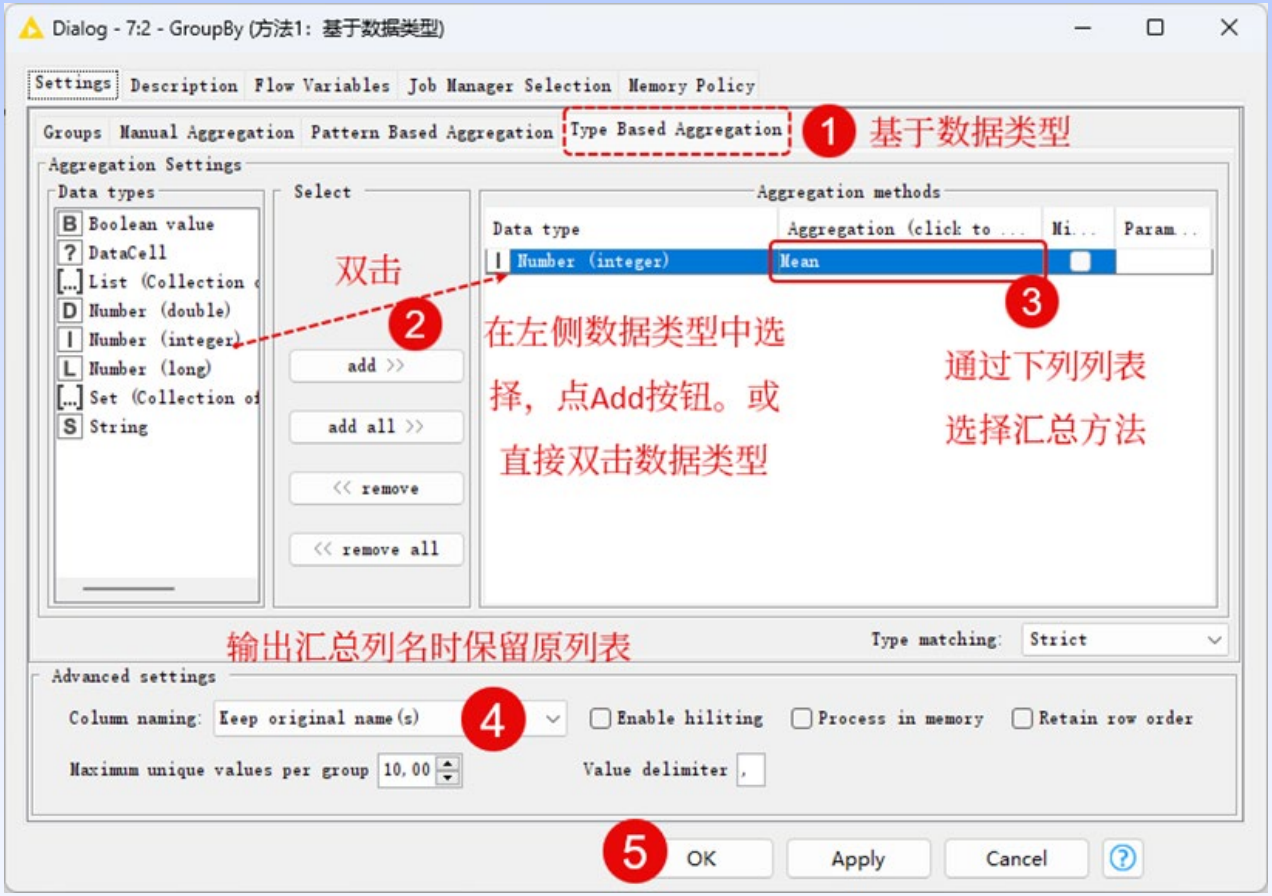
GroupBy节点应用案例2：计算各题平均得分



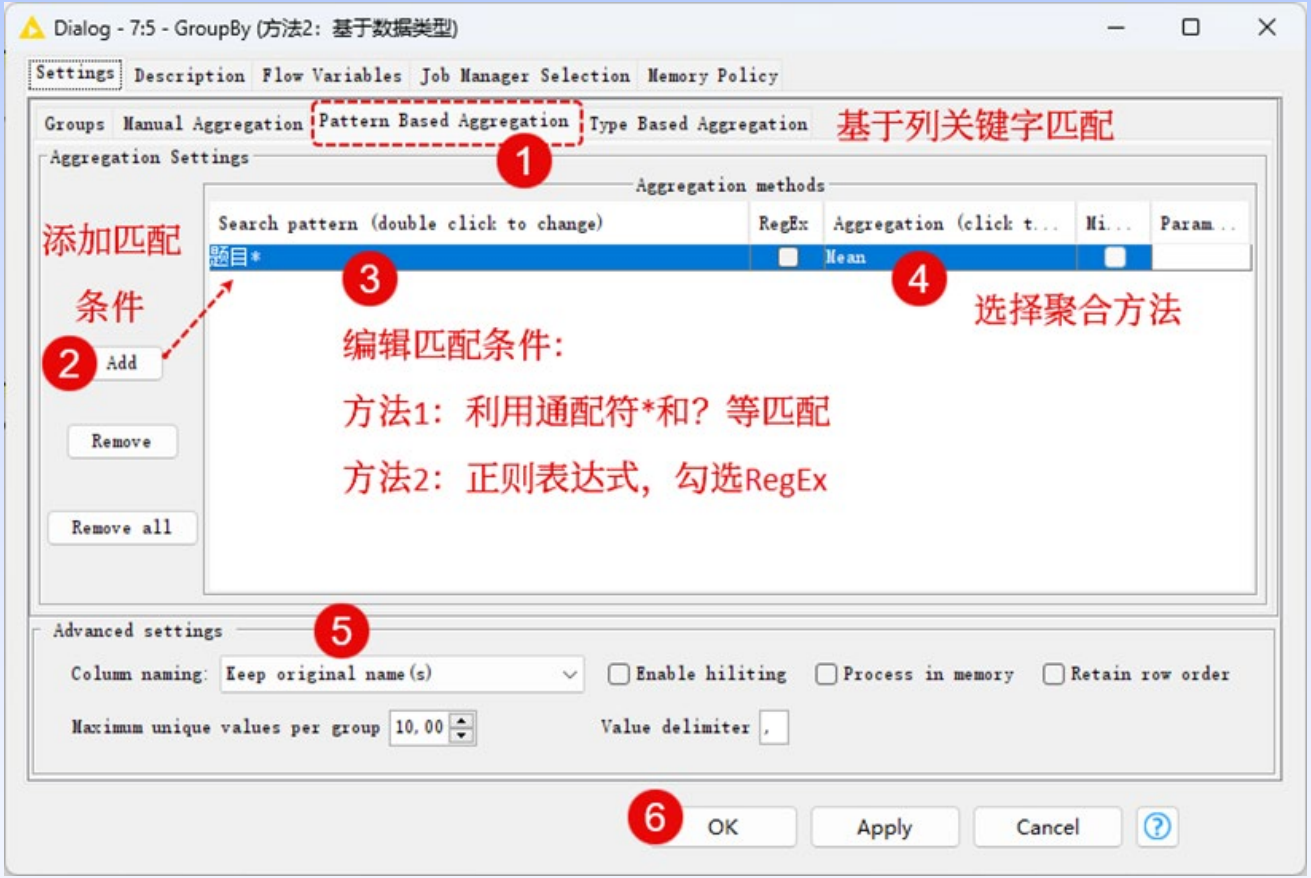
仅使用汇总要素2和3

选择列时，使用通配符或正则选择
详见随后2页PPT的节点设置界面截图

Group节点的设置—基于数据类型



Group节点的设置---基于关键字匹配





03

Pivot汇总节点应用

Pivot表基本概念

Pivot节点

数据透视节点用于汇总、分析、探索和展示数据。它能够帮助用户快速从大量数据中提取出有意义的信息，并以简洁明了的方式展示，有时也称之为二维交叉表。



Pivot节点与GroupBy节点区别

节点设置、及节点呈现结果存在差异，内在逻辑基本一致

Pivot表结构

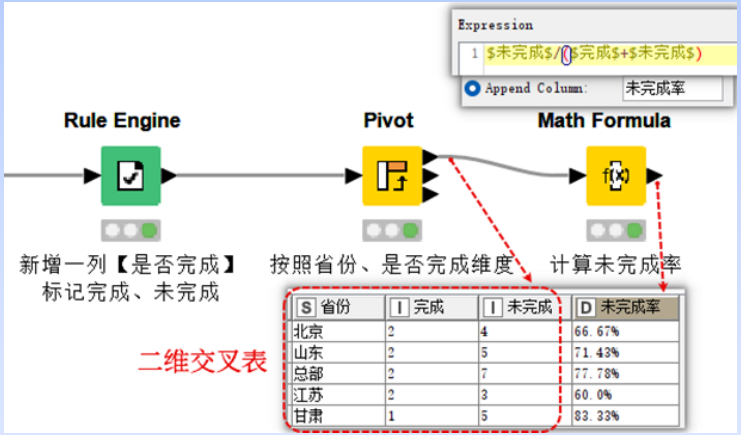
数据透视表允许用户动态改变版面布置，以便按照不同方式分析数据。

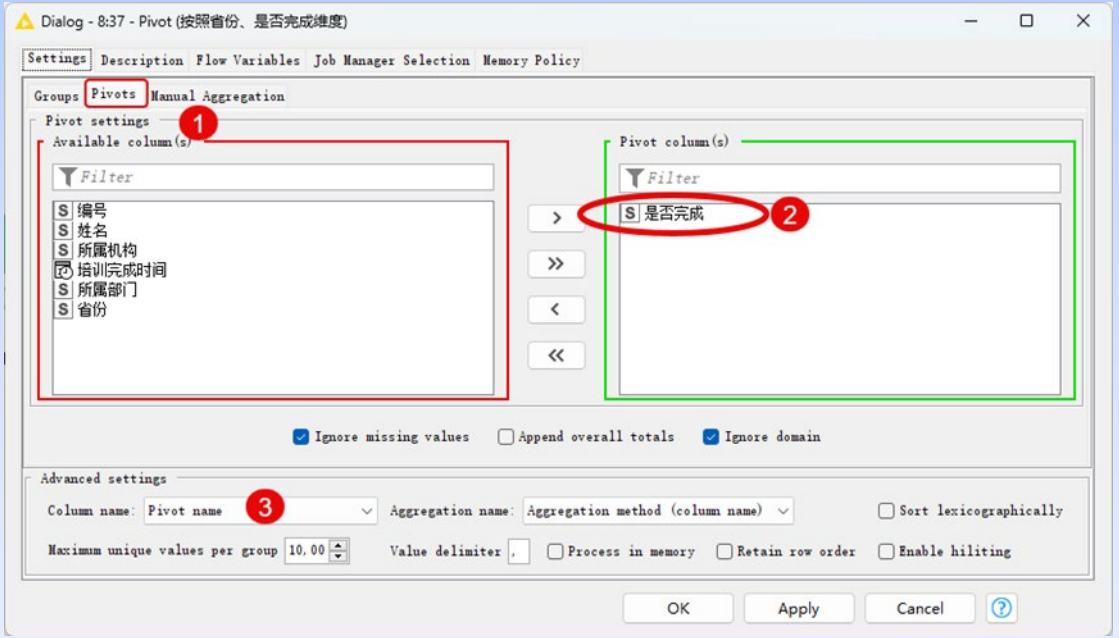
注意：

GroupBy节点纵向排列汇总结果（长表）

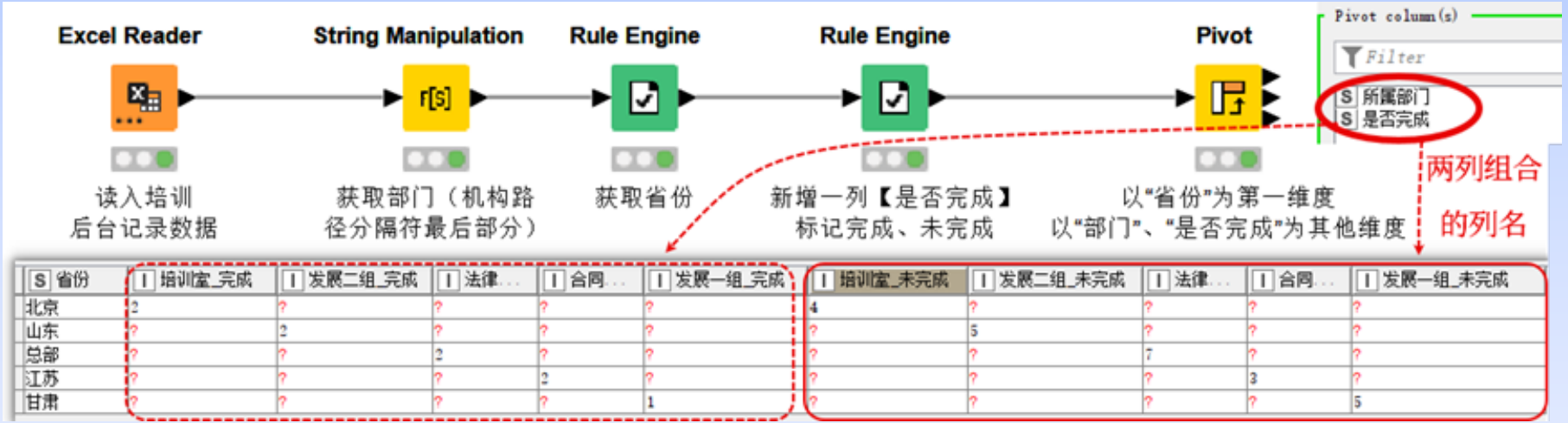
Pivot节点横向排列汇总结果（宽表）

案例1：二维交叉表形式汇总





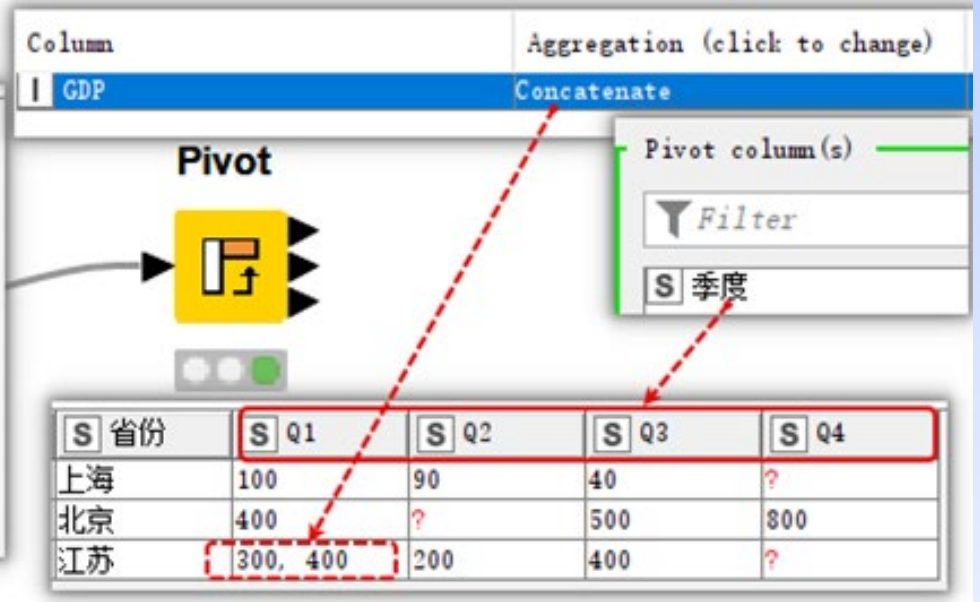
案例2：多维交叉表



省份	培训室_完成	发展二组_完成	法律	合同	发展一组_完成	培训室_未完成	发展二组_未完成	法律	合同	发展一组_未完成
北京	2	?	?	?	?	4	?	?	?	?
山东	?	2	?	?	?	?	5	?	?	?
总部	?	?	2	?	?	?	?	7	?	?
江苏	?	?	?	2	?	?	?	?	3	?
甘肃	?	?	?	?	1	?	?	?	?	5

案例3：数据格式转换（长表转宽表）

S 省份	S 季度	I GDP	I 人口
江苏	Q1	300	11
江苏	Q1	400	11
江苏	Q2	200	8
江苏	Q3	400	12
上海	Q1	100	3
北京	Q1	400	25
上海	Q2	90	5
上海	Q3	40	3
北京	Q3	500	75
北京	Q4	800	55

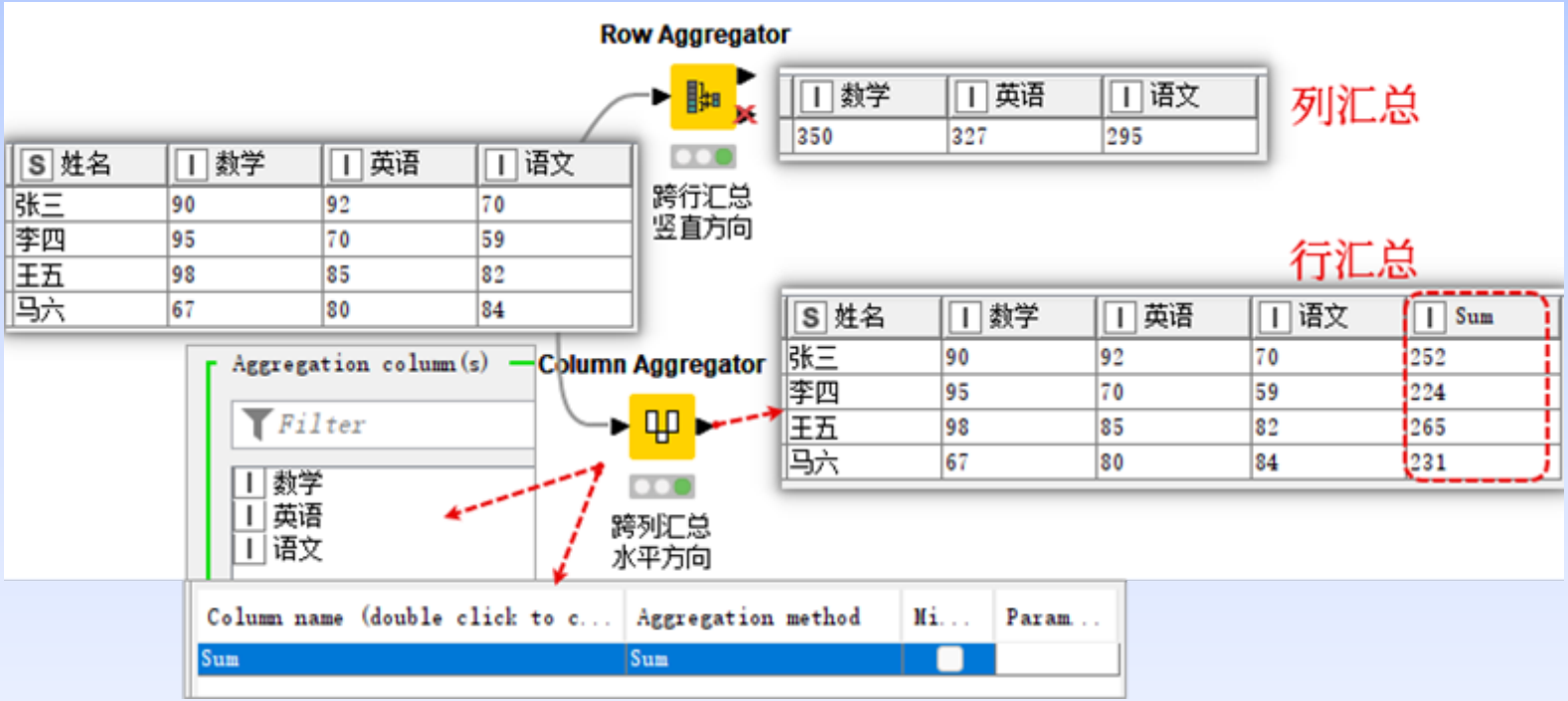




04

行列汇总方法

行列汇总节（GroupBy节点的衍生简化版）



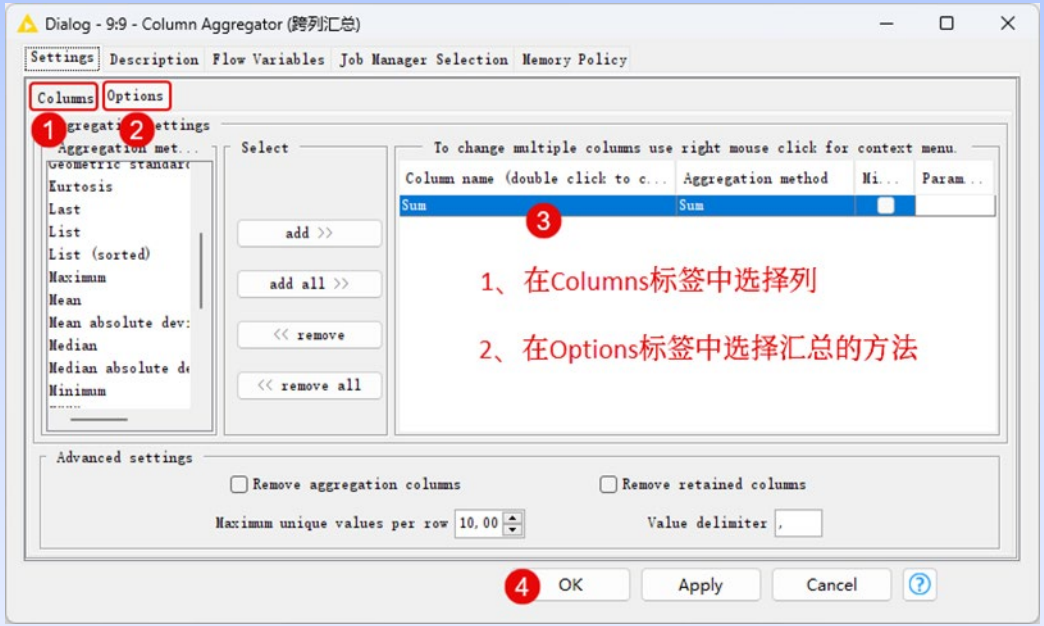
姓名	数学	英语	语文
张三	90	92	70
李四	95	70	59
王五	98	85	82
马六	67	80	84

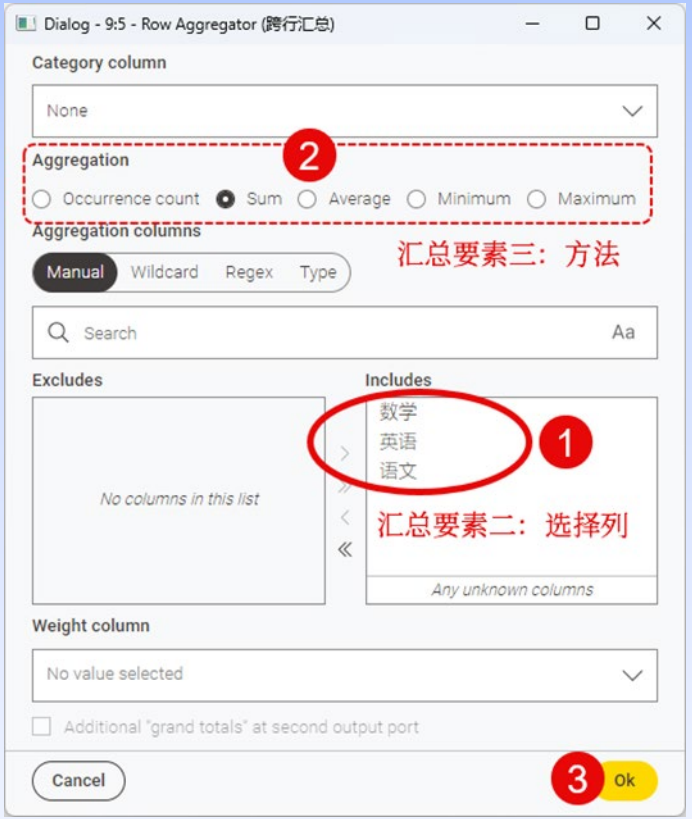
数学	英语	语文
350	327	295

姓名	数学	英语	语文	Sum
张三	90	92	70	252
李四	95	70	59	224
王五	98	85	82	265
马六	67	80	84	231

行列汇总顾名思义，针对行或列进行汇总，也即是在数据的水平或竖直方向上做汇总。这种类型的汇总，不需要汇总第一要素。注意有时行或列的理解和表达存在歧义，请结合汇总数据的方向去理解。

行列汇总节设置（GroupBy节点的衍生简化版）

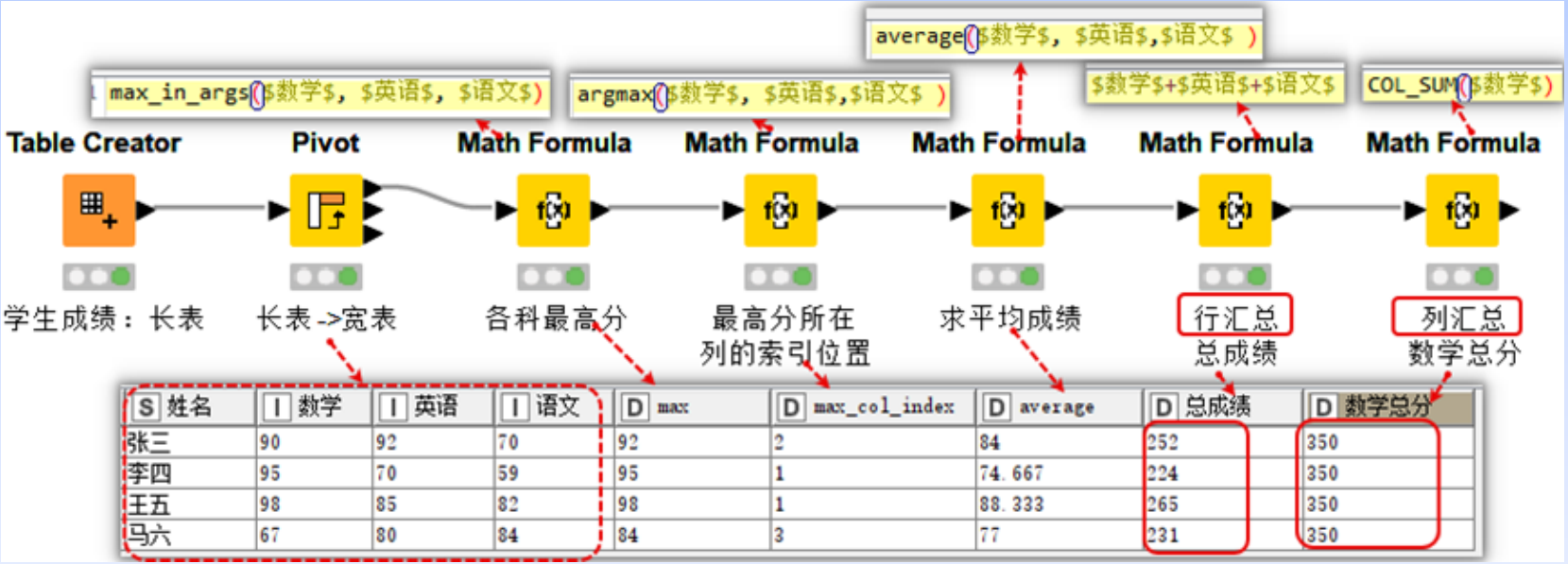




注意：两个节点的汇总方法，以及都不需要汇总第一要素

此外，Expression节点也内置了多个针对行或列的汇总（聚合）函数。

列汇总节（Math Formula）



S 姓名	I 数学	I 英语	I 语文	D max	D max_col_index	D average	D 总成绩	D 数学总分
张三	90	92	70	92	2	84	252	350
李四	95	70	59	95	1	74.667	224	350
王五	98	85	82	98	1	88.333	265	350
马六	67	80	84	84	3	77	231	350

Math Formula节点汇总类函数分布在aggregation function和constant子类中，相对而言汇总的方法少些。有些汇总方法的名称也存在差异，比如求均值，一般使用mean，而该节点函数名为average，两者意思等同。



05

移动窗口汇总

固定窗口与变动窗口汇总对比

固定窗口汇总定义与原理

固定窗口汇总是在处理数据时，预先设定一个固定的窗口大小。此方法适用于对短时间内的数据进行总结和分析，通过将固定数量的数据集中在一起，可以快速获取到整体趋势和特征。

变动窗口汇总定义与原理

变动窗口汇总则是指在处理数据时，窗口的大小是动态调整的。此方法适用于需要根据数据流的变化实时更新分析结果的场景，能够更灵活地捕捉数据的变化并做出相应的汇总。



固定窗口适用场景

固定窗口汇总适用于稳定的数据集，能够提供准确的整体趋势和特征，但无法适应数据流的变化。在实际应用中，固定窗口汇总常用于数据分析、机器学习模型训练等需要稳定数据的场景。

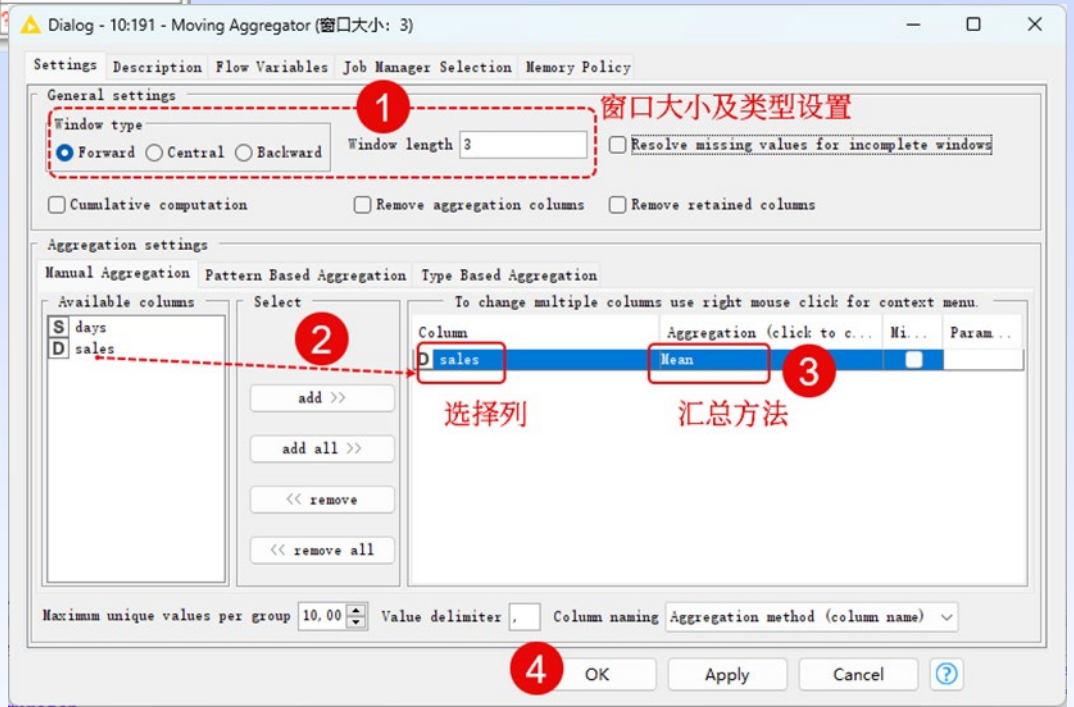
变动窗口适用场景

变动窗口汇总则多用于实时数据处理、流式分析等需要动态调整汇总结果的场景，例如累加求和。

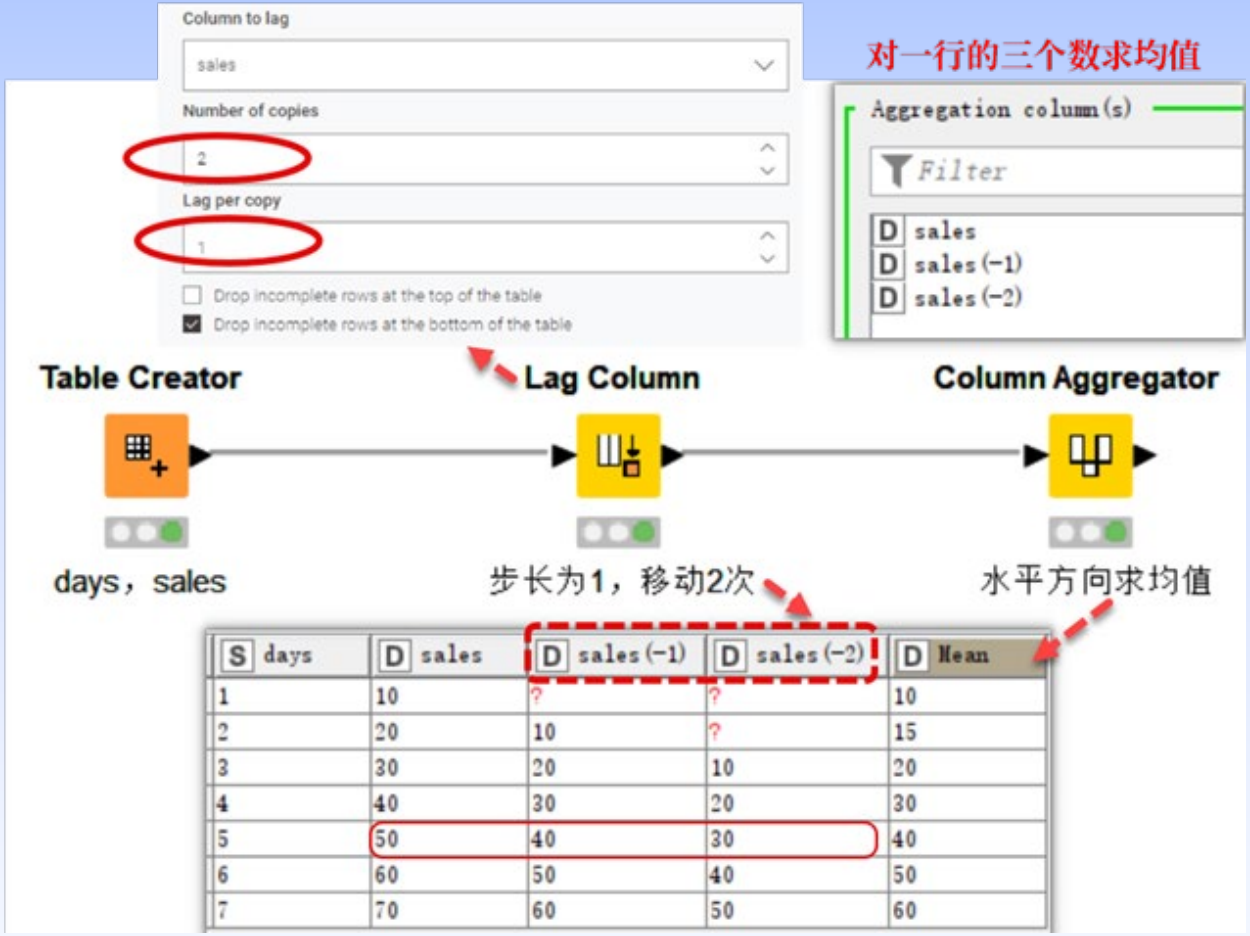
注意：窗口覆盖范围的数据集，即为汇总要素1（待汇总的组）

固定窗口汇总


S days
D sales
1
10
2
20
3
30
4
40
5
50
6
60
7
70
窗口1
Moving Aggregator
窗口大小：3
移动方式：forward
窗口5
S days
D sales
D Mean(sales)
1
10
20
2
20
30
3
30
40
4
40
50
5
50
60
6
60
?
7
70
?


Dialog - 10:191 - Moving Aggregator (窗口大小：3)
Settings
Description
Flow Variables
Job Manager Selection
Memory Policy
General settings
1
窗口大小及类型设置
Window type
Forward
Central
Backward
Window length
3
Resolve missing values for incomplete windows
Cumulative computation
Remove aggregation columns
Remove retained columns
Aggregation settings
Manual Aggregation
Pattern Based Aggregation
Type Based Aggregation
Available columns
S days
D sales
Select
2
add >>
add all >>
<< remove
<< remove all
To change multiple columns use right mouse click for context menu.
Column
Aggregation (click to c...
Mi...
Param...
D sales
Mean
3
选择列
汇总方法
Maximum unique values per group
10,00
Value delimiter
,
Column naming
Aggregation method (column name)
4
OK
Apply
Cancel

固定窗口汇总（手动计算）

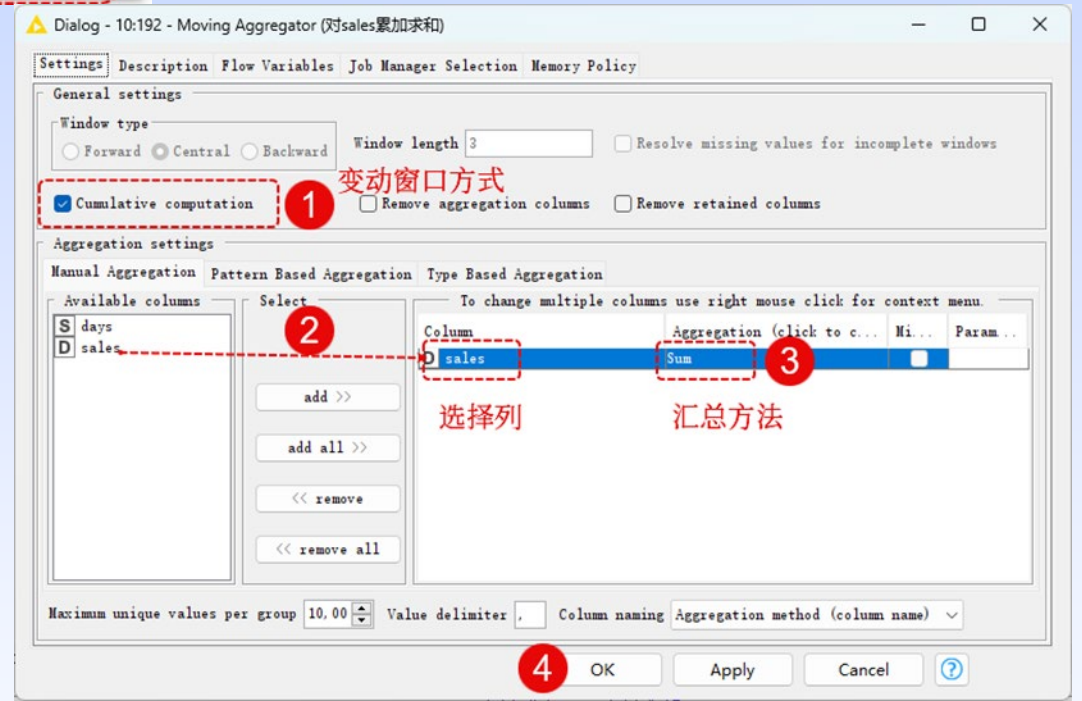


S days	D sales	D sales(-1)	D sales(-2)	D Mean
1	10	?	?	10
2	20	10	?	15
3	30	20	10	20
4	40	30	20	30
5	50	40	30	40
6	60	50	40	50
7	70	60	50	60

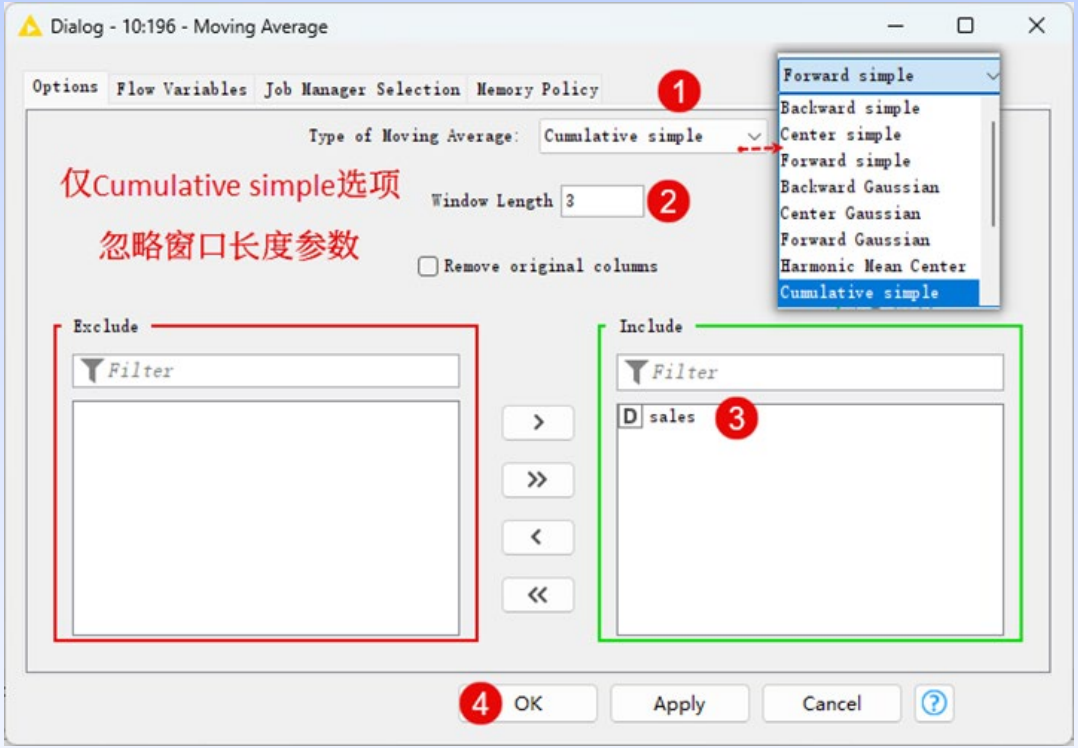
变动窗口汇总（累加求和）



S days	D sales	D Sum(sales)
1	10	10
2	20	30
3	30	60
4	40	100
5	50	150
6	60	210
7	70	280



变动窗口汇总（求均值）



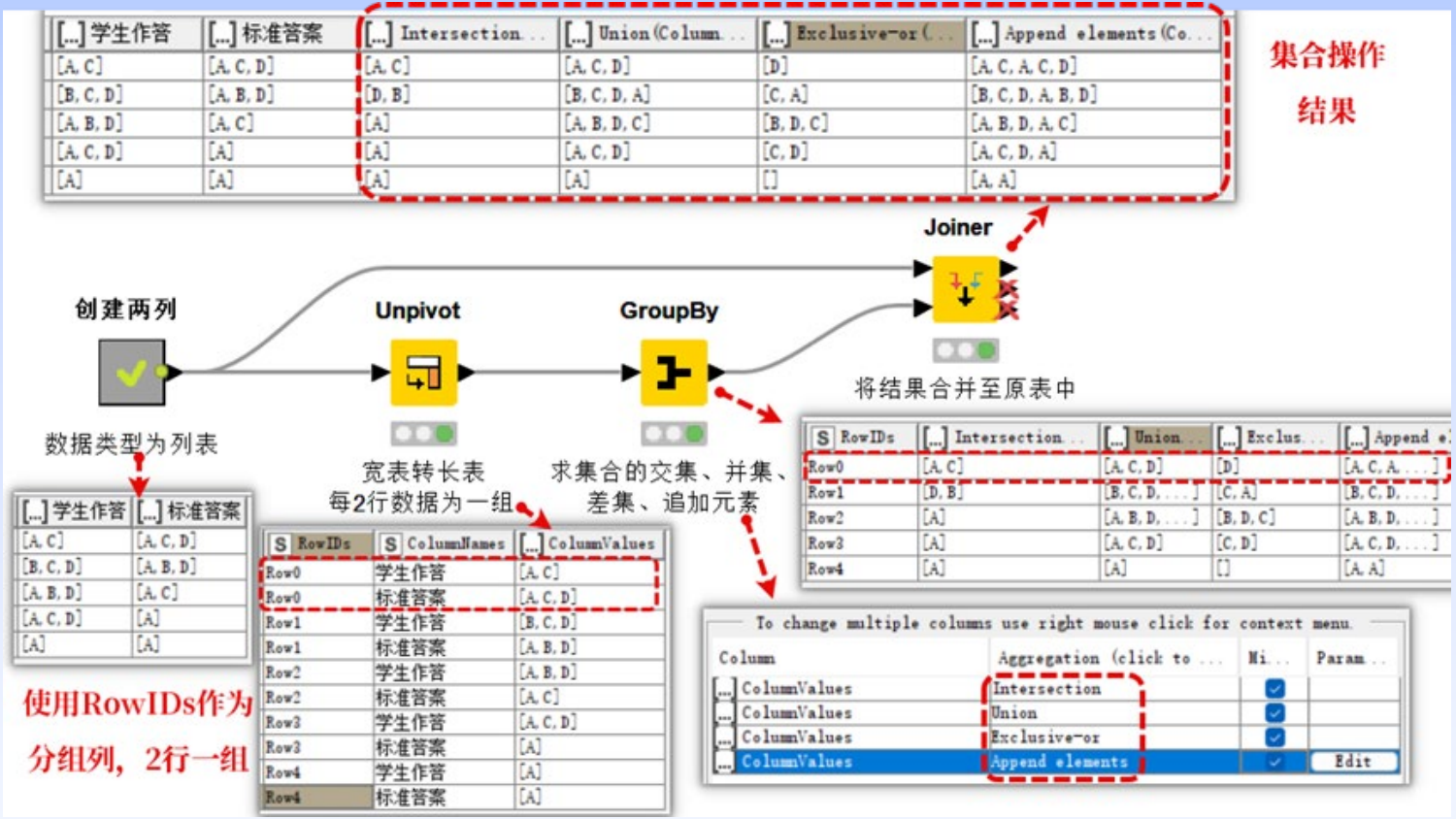
除了Moving Aggregator节点外，还有一个比较特别的节点Moving Average节点。该节点既有固定尺寸的移动窗口均值，也有变动窗口均值。相比Moving Aggregator节点多了几种不同的均值方法。



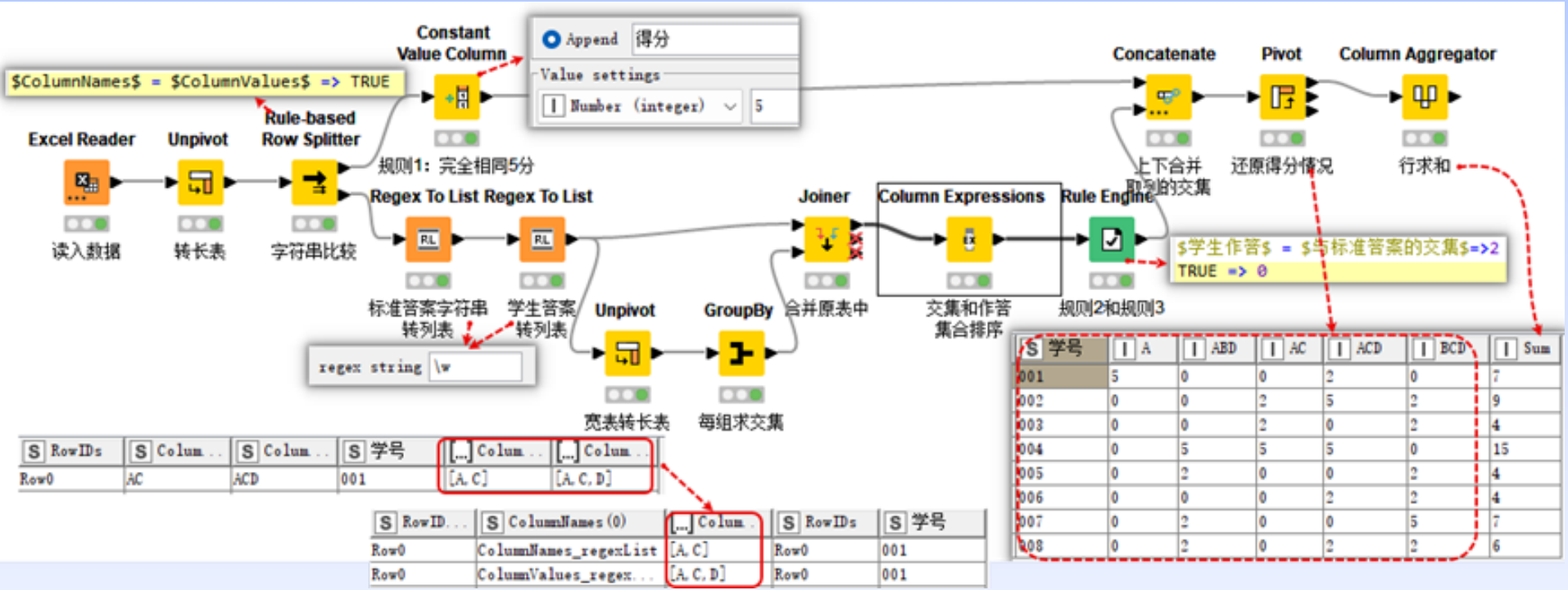
06

集合类型数据聚合方法

集合类型数据聚合—案例关键步骤



集合类型数据聚合—案例



RowIDs	Colum...	Colum...	学号	Colum...	Colum...
Row0	AC	ACD	001	[A, C]	[A, C, D]

RowID...	ColumnNames (0)	Colum...	RowIDs	学号
Row0	ColumnNames_regexList	[A, C]	Row0	001
Row0	ColumnValues_regex...	[A, C, D]	Row0	001

学号	A	ABD	AC	ACD	BCD	Sum
001	5	0	0	2	0	7
002	0	0	2	5	2	9
003	0	0	2	0	2	4
004	0	5	5	5	0	15
005	0	2	0	0	2	4
006	0	0	0	2	2	4
007	0	2	0	0	5	7
008	0	2	0	2	2	6



07

本章小结

主要技术回顾

01

数据汇总基本概念

数据汇总是将原始数据进行分类、计算和整理的过程，以便更好地理解和分析数据。通过数据汇总，可以将分散、无序的数据转化为集中、有序的形式，从而更易于分析和解释。

03

数据汇总技术重要性

在大数据时代，掌握数据汇总技术对于提高数据处理和分析能力至关重要。通过有效的数据汇总，可以提取出大量原始数据中的关键信息，为后续的数据分析提供坚实的基础。

02

数据汇总目标与方法选择

在进行数据汇总时，首先需要明确汇总目标，选择合适的方法和技术。关注数据的分布和异常值，使用合适的方法，可以提高数据汇总的效率和准确性。

实践应用建议

01 确定数据需求

在实践应用中，首先需要明确数据汇总的目标和需求。确定要解决的问题或业务需求，有助于选择合适的数据汇总技术和策略，确保最终结果满足实际需求。

02 理解数据结构

了解数据的基本结构是实践应用的关键。包括数据的维度、字段及其关系，这有助于选择合适的汇总方法，如行列汇总或透视汇总，以获得所需的汇总结果。

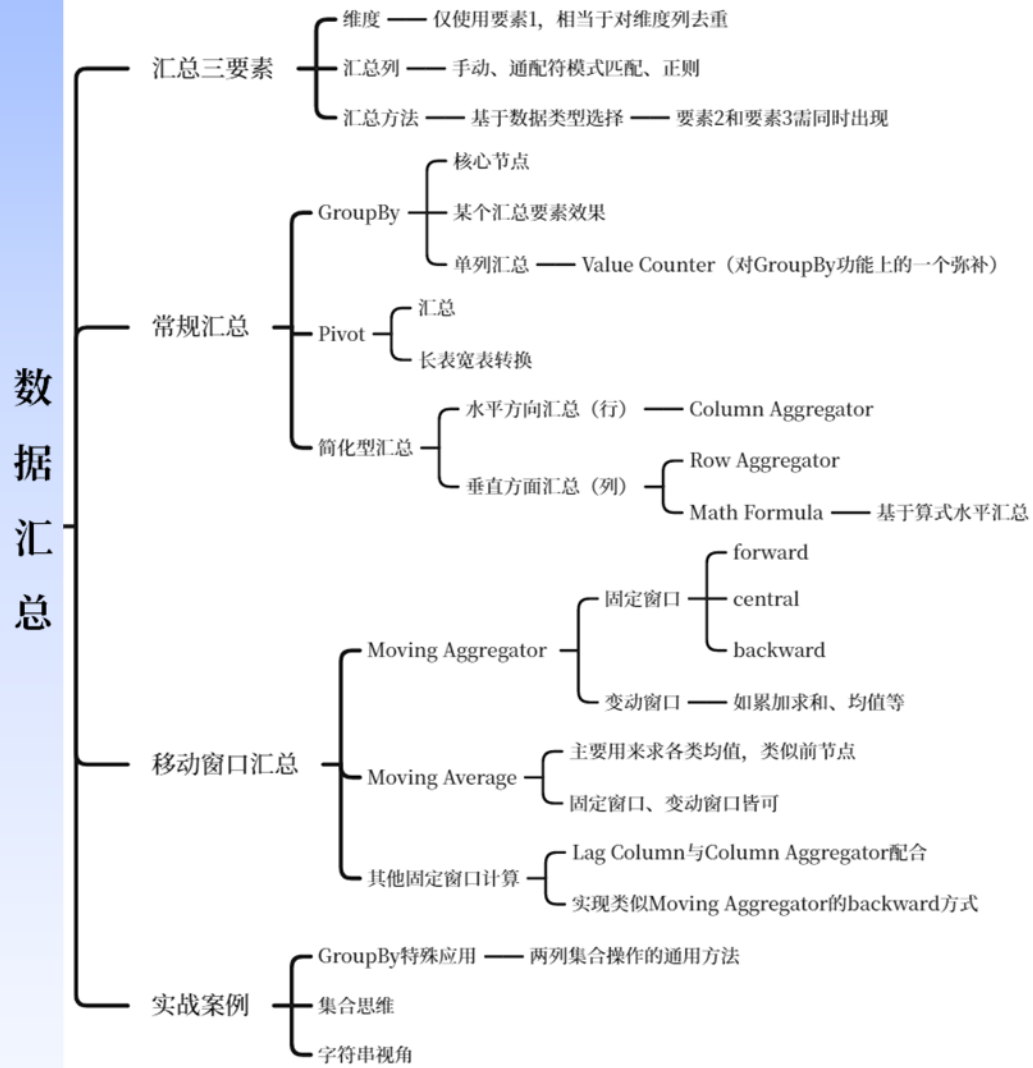
03 选择合适工具

根据数据规模和复杂性，选择适合的数据汇总工具。KNIME具有强大的数据处理能力，可以应对多种数据格式和复杂的数据结构，是实践应用的良好选择。

04 结果验证与调整

数据汇总完成后，需要对结果进行验证，确保其准确性和完整性。根据验证结果，及时调整汇总方案，解决可能存在的问题，确保数据汇总结果的可靠性和适用性。

本章总结

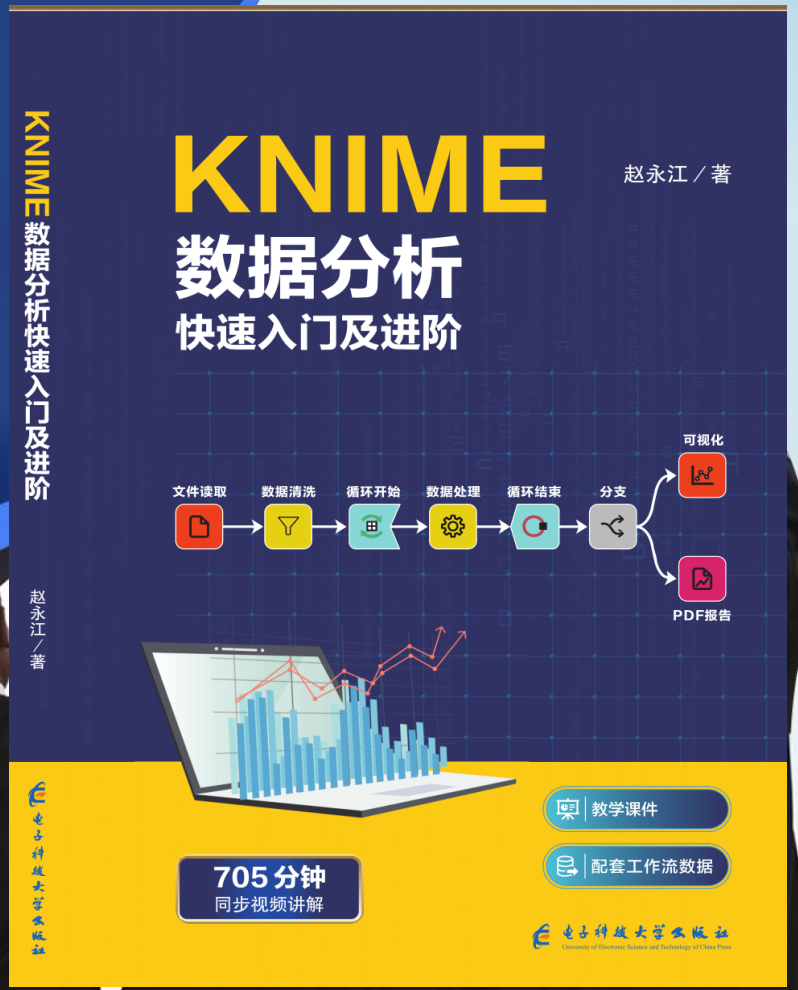

数据汇总
汇总三要素
维度
仅使用要素1，相当于对维度列去重
汇总列
手动、通配符模式匹配、正则
汇总方法
基于数据类型选择
要素2和要素3需同时出现
常规汇总
GroupBy
核心节点
某个汇总要素效果
单列汇总
Value Counter（对GroupBy功能上的一个弥补）
Pivot
汇总
长表宽表转换
简化型汇总
水平方向汇总（行）
Column Aggregator
垂直方面汇总（列）
Row Aggregator
Math Formula
基于算式水平汇总
移动窗口汇总
Moving Aggregator
固定窗口
forward
central
backward
变动窗口
如累加求和、均值等
Moving Average
主要用来求各类均值，类似前节点
固定窗口、变动窗口皆可
其他固定窗口计算
Lag Column与Column Aggregator配合
实现类似Moving Aggregator的backward方式
实战案例
GroupBy特殊应用
两列集合操作的通用方法
集合思维
字符串视角

感谢观看

THANKS



第8章 数据可视化


KNIME
数据分析
快速入门及进阶
赵永江／著
文件读取
数据清洗
循环开始
数据处理
循环结束
分支
可视化
PDF报告
KNIME数据分析快速入门及进阶
赵永江／著
705分钟
同步视频讲解
教学课件
配套工作流数据

目录

01 可视化节点概述

02 数据准备

03 基于维度的可视化节点分类

04 可视化节点的设置

05 绘图结果的应用

06 本章小结



01

可视化节点概述

可视化重要性

提高数据理解力

数据可视化通过将复杂数据转化为直观的图形和图表，帮助人们更容易理解和分析数据。这种转化使得原本难以直接解读的数据变得清晰易懂，为决策提供有力支持。

增强数据传递效果

数据可视化能够显著提升数据的传递效率。相比纯文本报告，可视化工具以图像和图形的形式展示数据，使信息传递更加迅速，确保观众在短时间内获取关键信息。



多维数据展示

数据可视化允许在单一视图中展示数据的多个维度，如时间、地域等。这使得用户可以从多个角度分析数据，发现潜在模式和趋势，为复杂的数据分析提供全面的视角。

辅助记忆与回顾

可视化数据有助于加强记忆，因为大脑对视觉信息的处理效率远高于文字信息。此外，可视化数据便于回顾和总结，有助于长期记忆和知识积累。

可视化节点定义

01

可视化节点概述

可视化节点是数据可视化工具中的核心组成部分，它们用于将数据转换为图形表示。这些节点提供了丰富的功能和灵活性，以支持不同类型的数据分析和展示需求。

02

可视化节点分类

可视化节点按照各节点在KNIME节点库的位置通常分为四类（详见下一页）。每种类型的节点都有其特定的用途和功能，能够有效地展示不同类型数据的特点和趋势。

03

辅助类图形节点

辅助类图形节点主要用于增强其他节点的功能或提供额外的信息。例如，标签、图例和注释节点可以提供数据的详细说明，帮助用户更好地理解可视化结果。

可视化节点分类（基于节点库位置）

KNIME 视图节点

- Views
  - Visualization Column Appender
  - Bar Chart
  - Box Plot
  - Density Plot
  - Generic ECharts View
  - Image View
  - Text View
  - Heatmap
  - Histogram
  - Line Plot
  - Parallel Coordinates Plot
  - Pie Chart
  - ROC Curve
  - Scatter Plot
  - Scatter Plot Matrix
  - Stacked Area Chart
  - Statistics View
  - Sunburst Chart
  - Table View

JavaScript类节点

- Views
  - JavaScript
    - Generic JavaScript View (JavaScript)
    - Bar Chart (JavaScript)
    - Box Plot (JavaScript)
    - Conditional Box Plot (JavaScript)
    - Decision Tree View (JavaScript)
    - Heatmap (JavaScript)
    - Histogram (JavaScript)
    - Lift Chart (JavaScript)
    - Line Plot (JavaScript)
    - Parallel Coordinates Plot (JavaScript)
    - Pie/Donut Chart (JavaScript)
    - ROC Curve (JavaScript)
    - Scatter Plot (JavaScript)
    - Stacked Area Chart (JavaScript)
    - Sunburst Chart (JavaScript)
    - Table Editor (JavaScript)
    - Table View (JavaScript)
    - Tag Cloud (JavaScript)
    - Tile View (JavaScript)

JavaScript试验类节点

- KNIME Labs
  - JavaScript Views (Labs)
    - Plotly
      - Bubble Chart (Plotly)
      - Continuous Error Plot (Plotly)
      - Contour Plot (Plotly)
      - 2D Density Plot (Plotly)
      - Error Bars Plot (Plotly)
      - Line Plot (Plotly)
      - Radar Plot (Plotly)
      - Scatter Plot (Plotly)
      - 3D Scatter Plot (Plotly)
      - Stacked Area Chart (Plotly)
      - Surface Plot (Plotly)
      - Violin Plot (Plotly)
    - Data Explorer
    - Tagged Document Viewer

不推荐图形节点

- Views
  - Local (Swing)
    - JFreeChart
      - Bar Chart (JFreeChart)
      - Bubble Chart (JFreeChart)
      - GroupBy Bar Chart (JFreeChart)
      - HeatMap (JFreeChart)
      - Histogram Chart (JFreeChart)
      - Interval Chart (JFreeChart)
      - Line Chart (JFreeChart)
      - Pie Chart (JFreeChart)
      - Scatter Plot (JFreeChart)
    - Box Plot (legacy)
    - Conditional Box Plot (legacy)
    - HiLite Table (legacy)
    - Histogram (legacy)
    - Interactive Histogram (legacy)
    - Interactive Pie chart (legacy)
    - Interactive Table (legacy)
    - Lift Chart (legacy)
    - Line Plot (legacy)
    - Parallel Coordinates (legacy)
    - Pie chart (legacy)

辅助类节点

- Views
  - Property
    - Color Manager
    - Size Manager
    - Shape Manager
    - Color Appender
    - Size Appender
    - Shape Appender
    - Extract Color
    - Number Format Manager
    - String Format Manager
  - Utility
    - CSS Editor
    - Image to Table
    - Table to Image
    - Renderer to Image
    - String to SVG

注意：5.5版本将该类标记为legacy

可视化节点分类（其他分类）

展示节点

展示节点用于生成可视化图表和报告，使数据分析结果直观易懂。这些节点包括柱状图节点、折线图节点、散点图节点等，支持多种图表类型的生成和配置。

交互节点

交互节点允许用户与可视化图表进行实时互动，如选择数据、调整参数和触发事件。这些节点增强了用户体验，使数据分析更加灵活和深入。该类节点可应用于动态仪表盘。此外，KNIME视图类节点在设置节点参数时也存在交互性。

自定义节点

自定义节点允许用户根据具体需求创建和配置节点，实现个性化的数据分析流程。用户可以使用JavaScript编写自定义节点，增强KNIME的功能和适用性。

此外，也可使用Python View节点，利用Python代码生成各种图形

5.7版本支持将Python View节点生成的图像插入到报告中

图形节点功能

01

统计图形节点

统计图形节点用于展示数据的分布和趋势，包括柱状图、折线图、饼图等。这些节点能够清晰地展示数据的比较、变化和比例关系，便于用户快速理解数据的核心信息。

02

矩阵图形节点

矩阵图形节点专门用于展示二维数据矩阵，如散点图、热力图等。通过矩阵图形，用户可以直观地观察不同变量之间的关系及其密度，适用于探索性数据分析和模式识别。

03

地图图形节点

地图图形节点用于将地理数据进行可视化展示，包括地图、热力图、地点分布图等。此类节点能够帮助用户从地理角度分析数据，发现地理位置上的规律和异常。

04

辅助类图形节点等

辅助类图形节点提供附加功能，如标签、图例、工具提示等。这些节点虽然不直接处理数据，但能显著提升图表的可读性和用户体验，使数据解释更为直观和准确。

图形节点应用场景

数据科学项目

KNIME在数据科学项目中广泛应用，涵盖数据探索、预处理、特征工程、模型训练和评估等各个阶段。其图形化工作流设计简化了数据处理流程，提高了数据分析的直观性和效率。

跨平台应用

KNIME具备跨平台特性，支持Windows、macOS和Linux操作系统。这使得用户能够在不同平台上使用相同的工具进行数据分析，提高了工作效率和协作性，满足了多样化的需求。

01

02

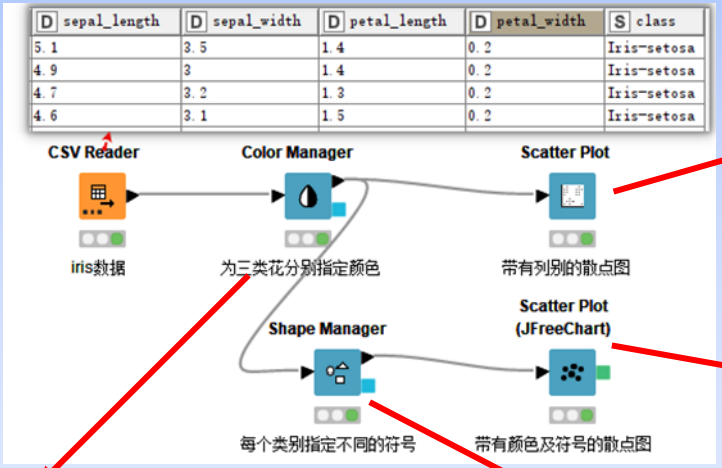
03

报告和仪表盘集成

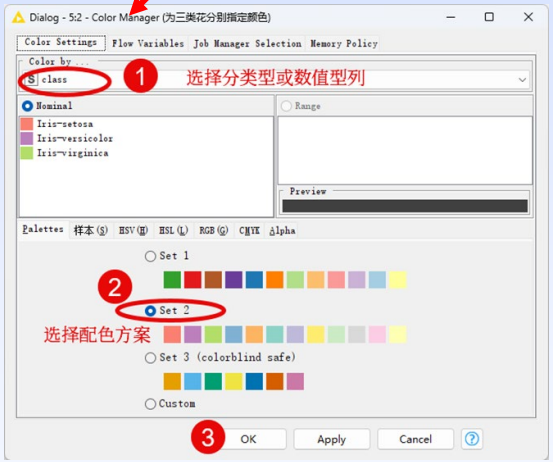
在制作报告或仪表盘时，KNIME的图形节点能够将复杂的数据可视化结果集成到文档或仪表盘中，提供直观的视觉支持。这有助于观众更好地理解数据背后的信息，增强沟通效果。

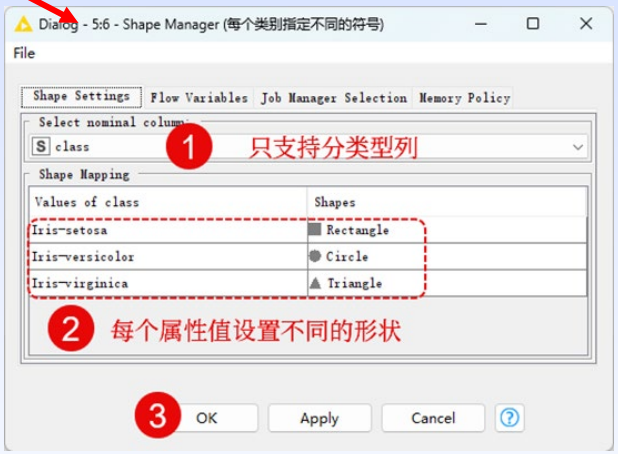
可视化节点示例

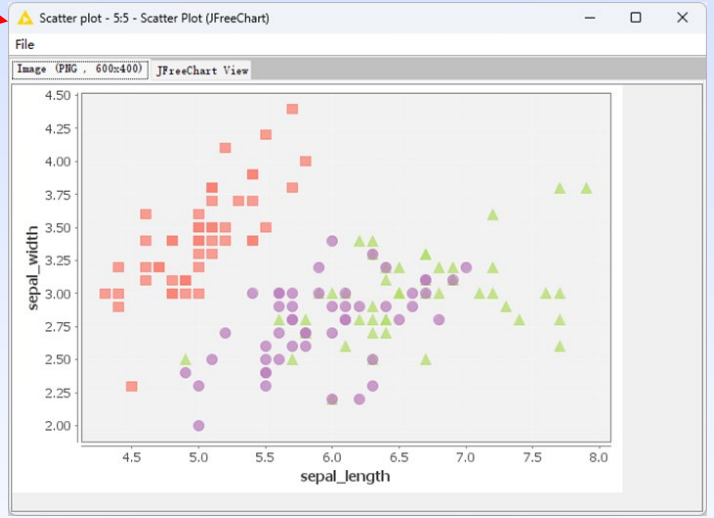
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa













02

数据准备

数据清洗与转换

数据清洗步骤

数据清洗包括初步的数据预处理、详细清洗操作以及异常值和缺失值的处理。这一过程有助于消除数据中的噪声和错误，提高数据质量，为后续分析和可视化打下基础。



数据类型转换

数据类型转换是数据清洗的重要环节，通过将不同类型的数据转换为适合分析和可视化的格式，如将字符串数据转换为数值型或时间格式。这有助于提升数据的一致性和可用性。

数据标准化处理

数据标准化处理用于使不同来源或格式的数据具有一致的度量单位和范围，如将日期格式统一或把温度数据转换为摄氏度。这有助于确保数据分析结果的可靠性和可比性。

数据去重与合并

数据去重与合并操作用于消除重复数据并整合多个数据源，确保分析数据的完整性和准确性。通过此过程，可以避免数据冗余，提高数据清洗的效率和效果。

可视化节点在工作流中的相对位置

01

数据导入节点

数据导入节点用于读取和处理多种数据格式，包括Excel、CSV、MySQL连接器和数据库表。这些节点支持多种数据源，能将excel/csv文件以及数据库中的数据直接导入，为后续的数据处理提供便利。

02

数据处理节点

数据处理节点用于清洗、转换和预处理数据，如去除重复值、处理缺失值等。这些节点在数据分析流程中扮演着重要角色，确保数据的准确性和一致性，提高分析结果的质量。

03

数据分析节点

数据分析节点包括统计分析和机器学习算法，用于挖掘数据中的潜在模式和关系。这些节点提供了多种统计方法及算法，如回归分析、聚类分析和分类模型，帮助用户进行深入的数据分析。

04

数据可视化节点

数据可视化节点用于创建图表、报告和仪表盘，以直观展示数据分析结果。这些节点支持多种可视化方式，如柱状图、折线图、散点图等，帮助用户更好地理解和解释数据。

扩展包安装

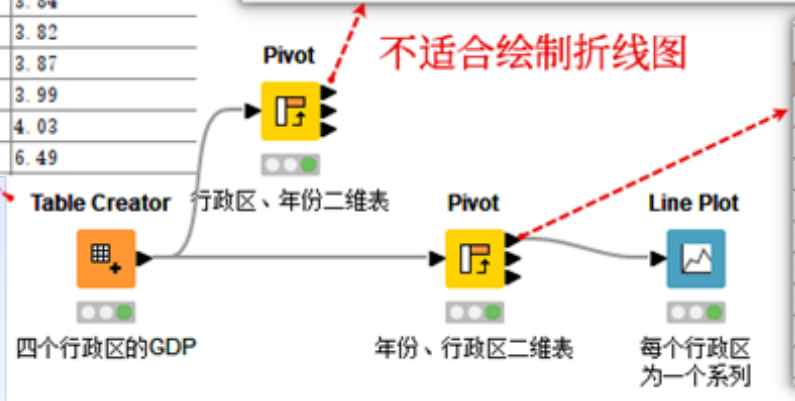
KNIME通过安装扩展包来扩展其节点库，例如网络图Network Viewer节点。通过安装扩展包，增强平台的功能和灵活性，提升整体的数据分析能力。

数据准备

01 数据导入 → 02 数据准备 → 03 数据可视化

S 行政区	S 年份	D GDP(单位...
上海	2011年	2.48
上海	2012年	2.87
上海	2013年	3.27
上海	2014年	3.76
上海	2015年	3.87
上海	2016年	3.84
上海	2017年	3.82
上海	2018年	3.87
上海	2019年	3.99
上海	2020年	4.03
江苏	2011年	6.49

S 行政区	D 2011年	D 2012年	D 2...	D 2...	D 2...	D 2...	D 2...	D 2...	D 2019年	D 2020年
上海	2.48	2.87	3.27	3.76	3.87	3.84	3.82	3.87	3.99	4.03
北京	1.64	1.93	2.13	2.28	2.36	2.49	2.68	2.86	3.04	3.17
广东	5.73	6.42	7.09	7.95	8.59	8.99	9.73	10.27	10.78	11.19
江苏	6.49	7.27	8.02	8.94	9.61	9.99	10.49	11.07	11.52	12.02



S 年份	D 上海	D 北京	D 广东	D 江苏
2011年	2.48	1.64	5.73	6.49
2012年	2.87	1.93	6.42	7.27
2013年	3.27	2.13	7.09	8.02
2014年	3.76	2.28	7.95	8.94
2015年	3.87	2.36	8.59	9.61
2016年	3.84	2.49	8.99	9.99
2017年	3.82	2.68	9.73	10.49
2018年	3.87	2.86	10.27	11.07
2019年	3.99	3.04	10.78	11.52
2020年	4.03	3.17	11.19	12.02



03

基于维度的可视化节点分类

一维可视化节点

01

一维可视化节点概述

一维可视化节点用于展示和管理一维数据，如时间序列数据、计数数据等。通过将数据转化为时间轴上的点或条形，能够直观地展示数据的分布和趋势。

02

一维时间序列分析

一维时间序列分析节点可以对时间序列数据进行各种处理和分析，包括数据平滑、滤波、回归等。这些操作帮助提取和展示时间序列中的重要特征和模式。

03

一维计数数据展示

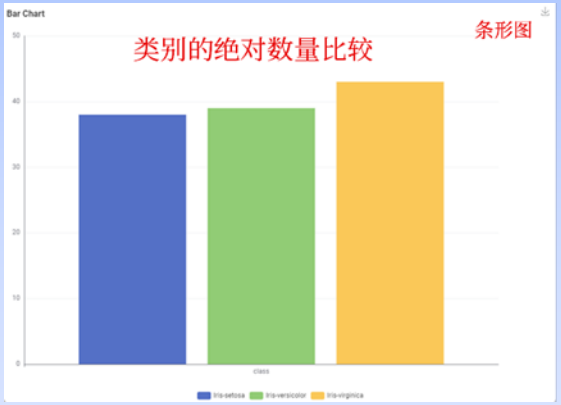
一维可视化节点也适用于展示计数数据，如各项目的频数统计。利用柱状图或饼图等可视化方式，可以清晰地显示不同类别的数据比例和分布情况。

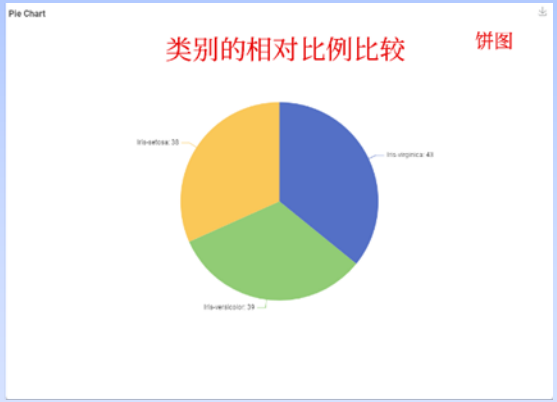
04

一维节点应用场景

一维可视化节点广泛应用于各种数据分析场景，如销售趋势分析、人口统计、质量控制等。通过一维数据展示，可以帮助用户快速理解和解释复杂数据。

一维可视化节点

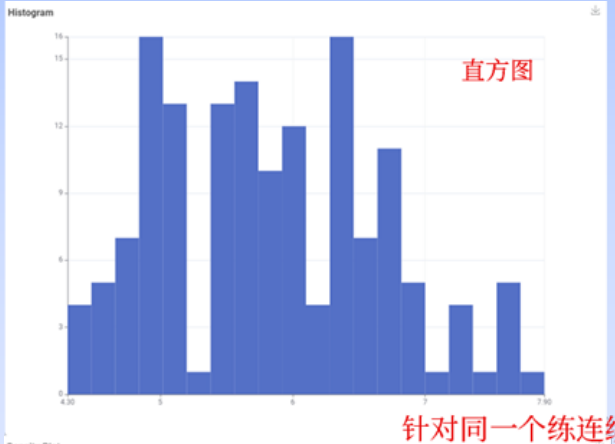


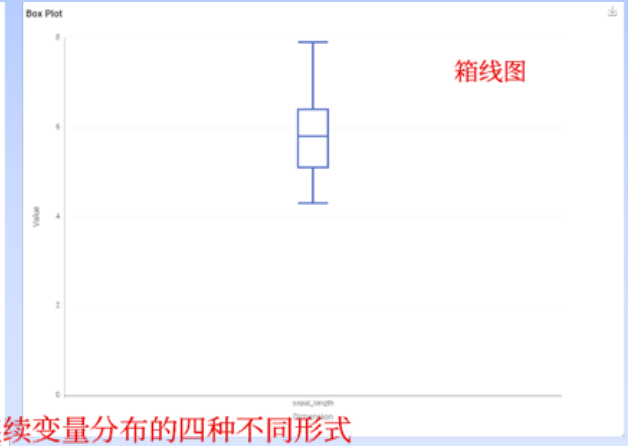




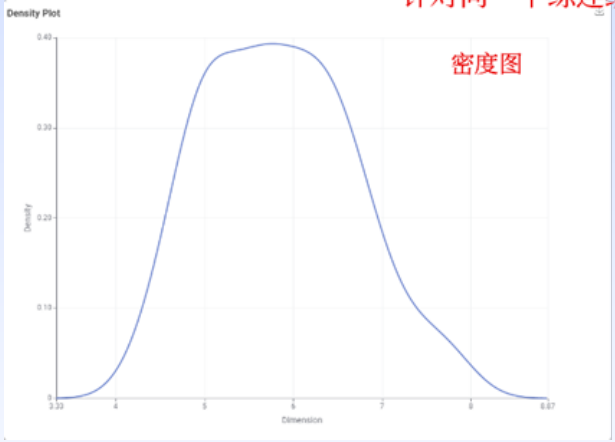
一维绘图的目的是更好地理解单列数据的变化趋势和数量关系，进行数据的比较，以便更好地理解不同类别之间的差异或相似性。由于维度单一，相对来说，分析维度不够精细，可辅助进行数据理解，进行数据探索。

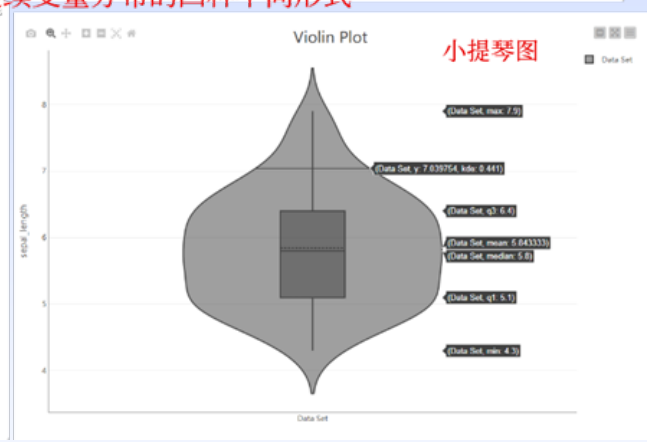
一维可视化节点





针对同一个练连续变量分布的四种不同形式





二维可视化节点

二维可视化节点功能

二维可视化节点用于展示数据的二维分布，如散点图、柱状图等。这些节点能够清晰地显示变量之间的关系和分布情况，帮助用户直观理解数据特征。

常见二维可视化节点类型

常见的二维可视化节点包括小提琴图、箱线图和密度图等节点。这些节点支持多种数据格式，能够灵活展示不同类型的数据关系和趋势。



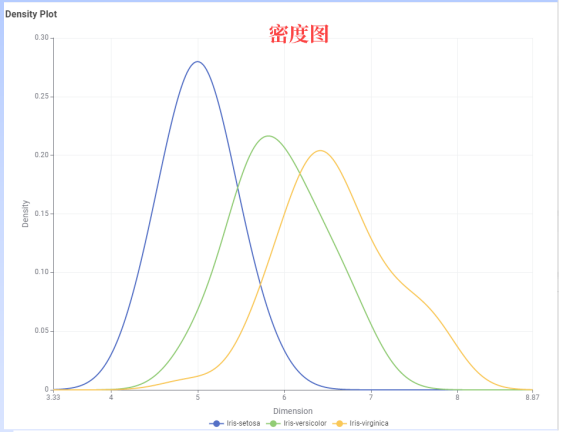
使用二维可视化节点进行数据分析

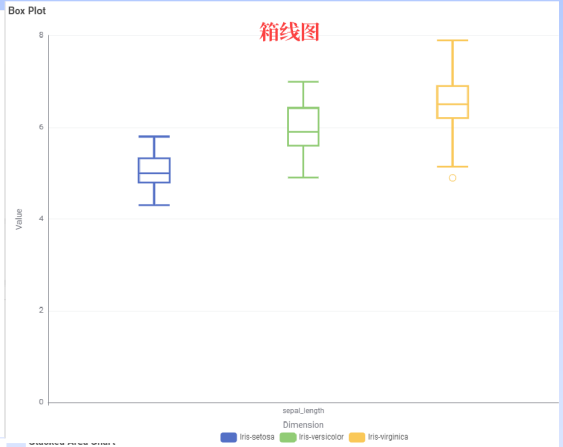
利用二维可视化节点进行数据分析，可以快速识别数据中的关键模式和异常值。例如，通过散点图发现变量之间的相关性，或利用柱状图比较不同分组的数据差异。

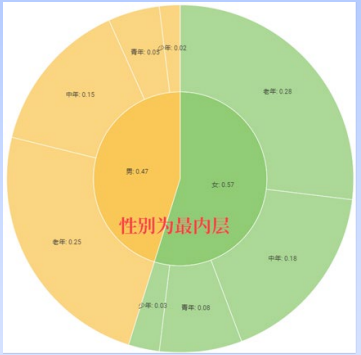
配置与优化二维可视化节点

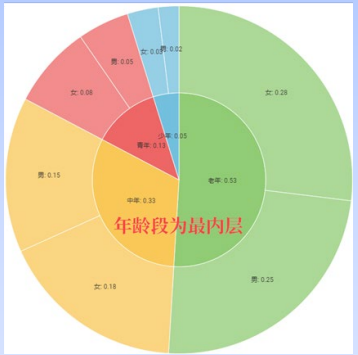
在使用二维可视化节点时，可以根据需求调整图表的样式、标签和颜色等配置选项。优化配置有助于提升图表的可读性和信息传递效率。

二维可视化节点

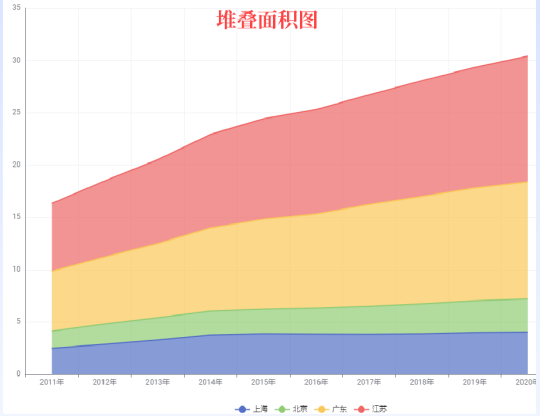

Density Plot
密度图
Density
Dimension
Iris-setosa
Iris-versicolor
Iris-virginica


Box Plot
箱线图
Value
sepal_length
Dimension
Iris-setosa
Iris-versicolor
Iris-virginica

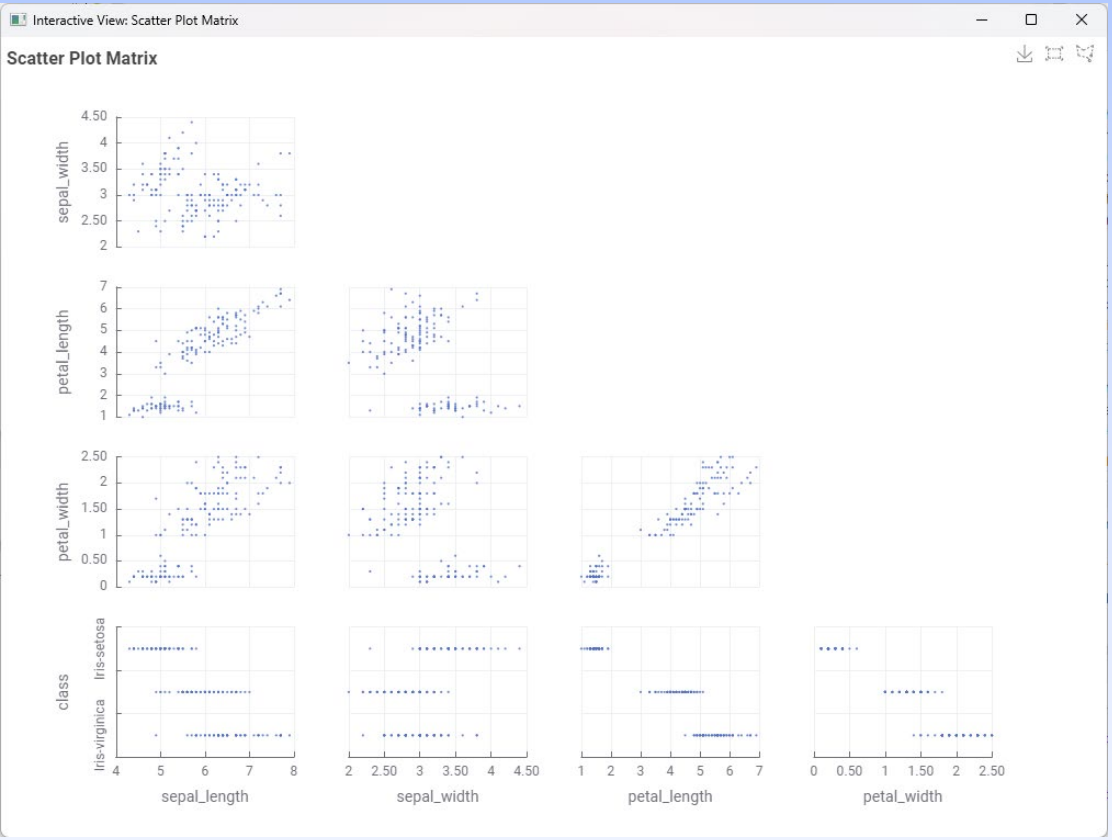

性别为最内层

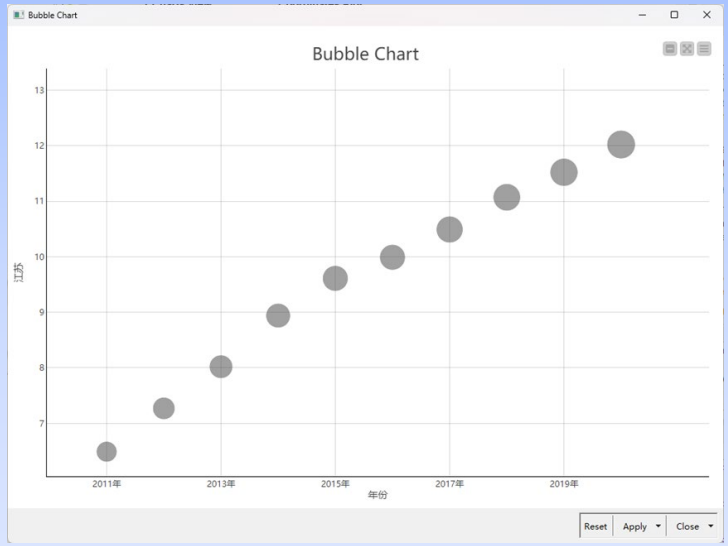

年龄段为最内层


Violin Plot
小提琴图
sepal_length
Iris-setosa
Iris-versicolor
Iris-virginica
class


堆叠面积图
上海
北京
广东
江苏

二维可视化节点


Interactive View: Scatter Plot Matrix
Scatter Plot Matrix
sepal_width
petal_length
petal_width
class
Iris-setosa
Iris-virginica
sepal_length
sepal_width
petal_length
petal_width

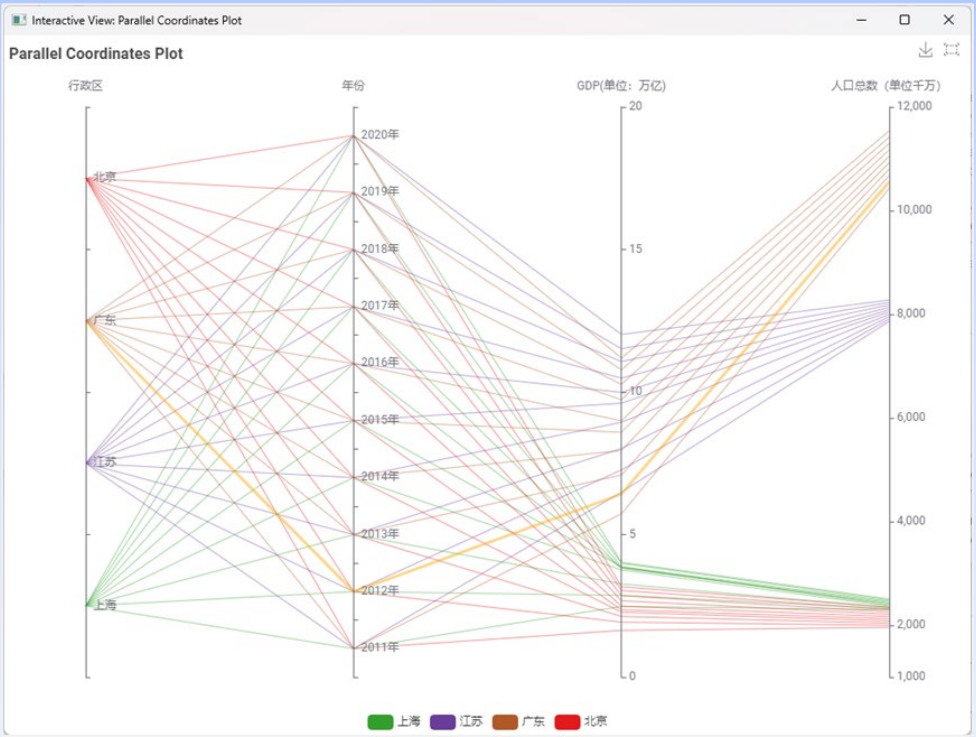

Bubble Chart
Bubble Chart
江苏
年份
2011年
2013年
2015年
2017年
2019年
Reset
Apply
Close

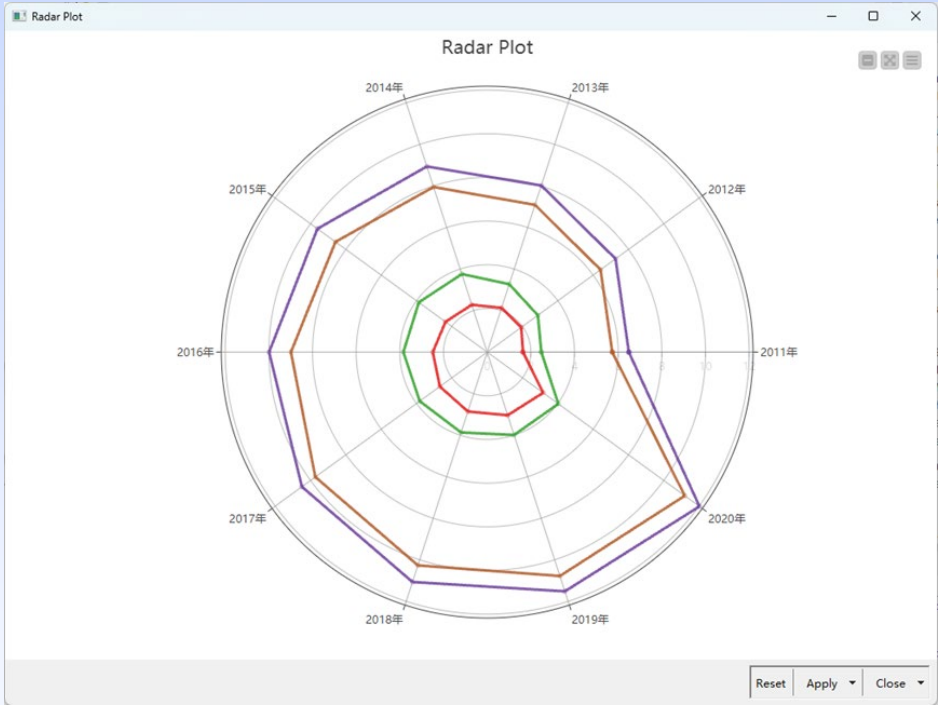

Interactive View: Heatmap
Heatmap
上海
江苏
广东
北京
2011年
2012年
2013年
2014年
2015年
2016年
2017年
2018年
2019年
2020年

多维可视化节点

图形	节点名关键字	图形说明
平行坐标图	Parallel Coordinates Plot	展示多个变量之间的关系，每个列视为一个维度，每一行的数据将在各个列维度连成一条线，通过多条平行的轴来展示多变量间的关系。可以使用Color Manager节点指定某分类型的颜色，以更好的区别不同类别。
雷达图	Radar Plot、Radar Plot Appender	展示多维属性在一个对象上的得分或等级，多用于评价系统或性能评估。Radar Plot Appender是将各个图直接插入表格内的一列中，而不是输出一个图。前者生成一个图，后者生成一列图（每个图插入到列的各个单元格中）。
火花线图	Spark Line Plot Appender	Spark line是一种迷你图表，用于简洁地展示数据趋势和模式，如平均温度、股市交易活跃度等。这种图表的设计主要是为了在有限的空间内高效地传达信息。
三维散点图	3D Scatter Plot	包括3D散点图、3D折线图、3D柱状图等，可以在三维空间中展示三个连续变量之间的关系。

多维可视化节点


Interactive View: Parallel Coordinates Plot
Parallel Coordinates Plot
行政区
年份
GDP(单位：万亿)
人口总数（单位千万）
北京
广东
江苏
上海
2020年
2019年
2018年
2017年
2016年
2015年
2014年
2013年
2012年
2011年
20
15
10
5
0
12,000
10,000
8,000
6,000
4,000
2,000
1,000
上海
江苏
广东
北京


Radar Plot
Radar Plot
2014年
2013年
2015年
2012年
2016年
2011年
2017年
2020年
2018年
2019年
Reset
Apply
Close

其他可视化节点

数据探索性节点

主要有三个节点，Data Explorer、Statistics、Statistics View。这些节点自动计算出多个统计指标，对于理解数据非常方便。Data Explorer节点，以直方图形式显示每列的统计值及数值分布

与模型有关的可视化节点

展现模型结果或评估模型性能相关图，如Decision Tree View、Lift Chart、ROC Curve

其他通用性图节点

交互式按钮节点允许用户创建交互式的图形，如点击按钮显示不同数据。该节点广泛应用于动态报告和仪表盘，使用户可以通过交互方式探索和分析数据。如Generic Echarts View、Generic JavaScript View 以及Python View节点。



与仪表盘或报告相关节点

文字（Text View）、数据表（Table View）、平铺图（Tile View）虽是文字或数据表，也视为一类特殊的图。此外，Table Editor节点的视图结果中，可以对表格进行编辑

辅助图形节点应用技巧

数据辅助功能

辅助图形节点提供多种数据辅助功能，包括数据标签、数据点提示和动态追踪。这些功能帮助用户更直观地理解数据，提高数据分析的准确性与效率。



交互式数据探索

通过辅助图形节点，用户可以进行交互式数据探索，如缩放、旋转和过滤图表。交互功能使得用户能够从不同角度观察数据，发现潜在的模式和异常值。

多维度数据可视化

辅助图形节点支持多维度数据可视化，例如时间序列分析、地理空间数据展示等。通过丰富的图形类型和定制选项，用户可以创建复杂的数据视觉作品。

自动化报告生成

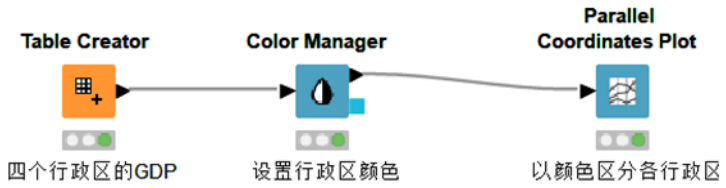
辅助图形节点可以轻松集成到自动化报告中，实现数据的自动展示和解释。这有助于快速生成详细的分析报告，提高沟通效率和信息的透明度。



04

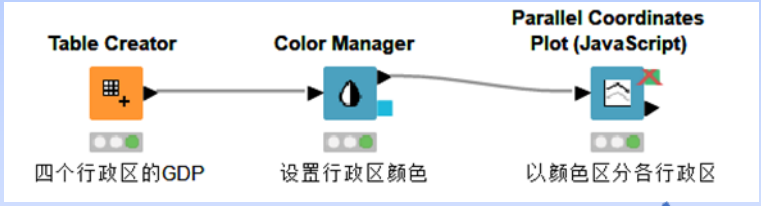
可视化节点设置

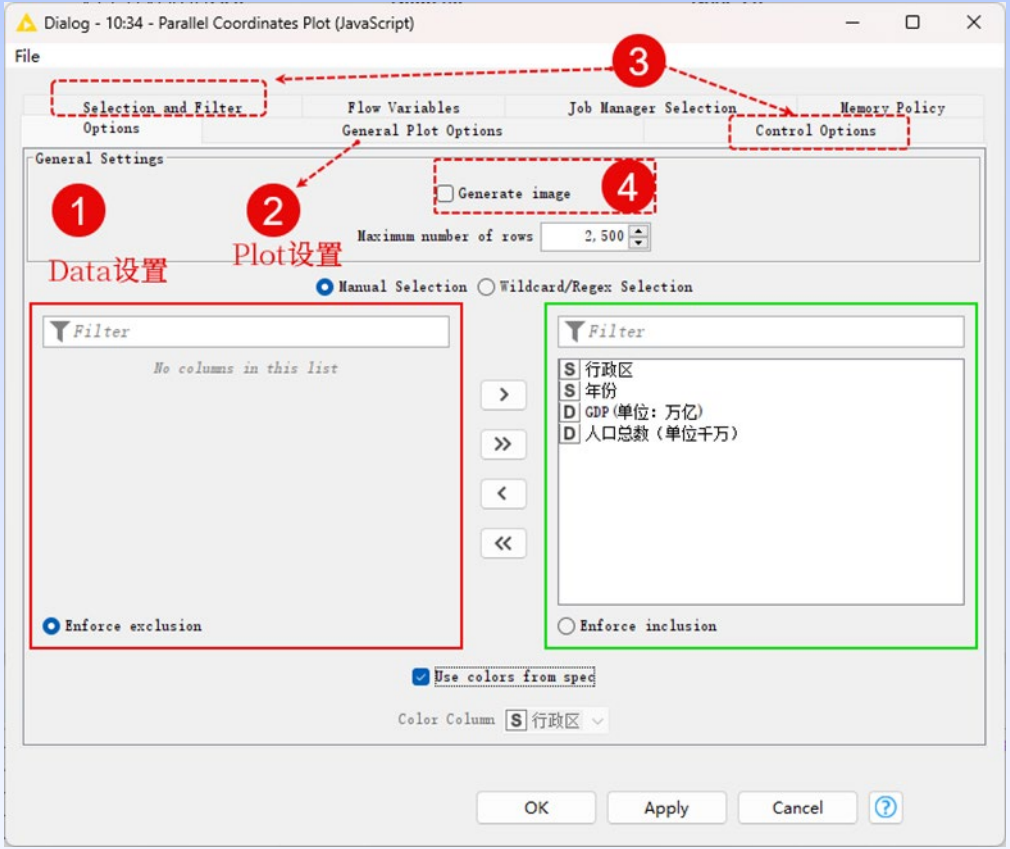
KNIME视图类节点示例





JavaScript类视图节点示例





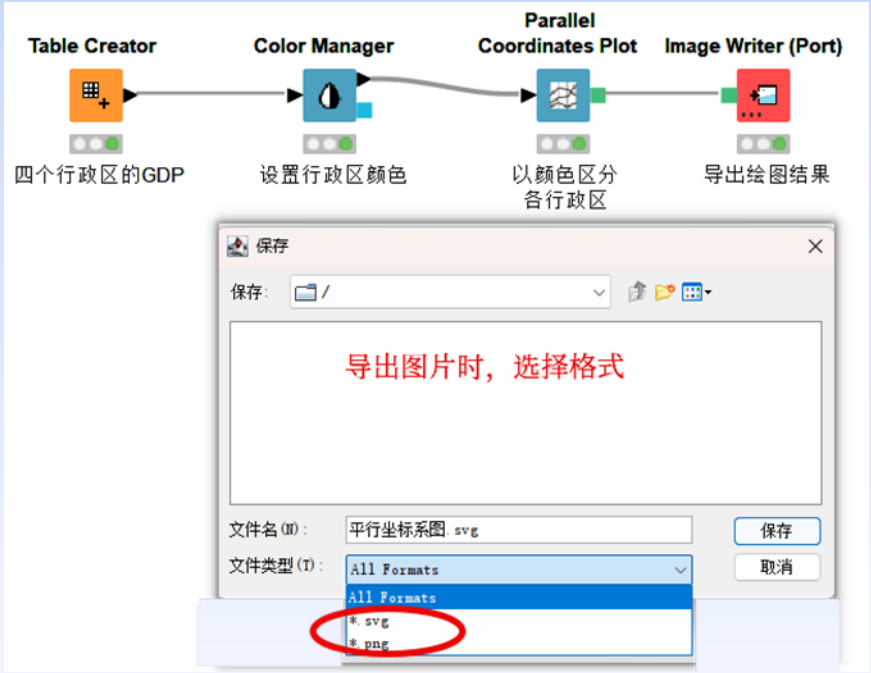


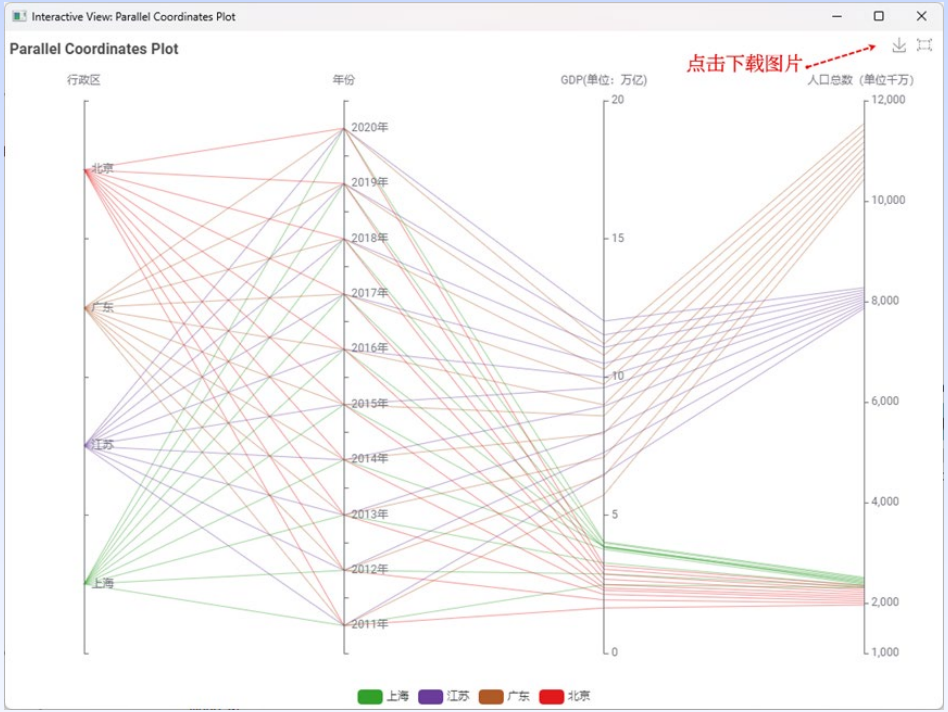
05

绘图结果的应用

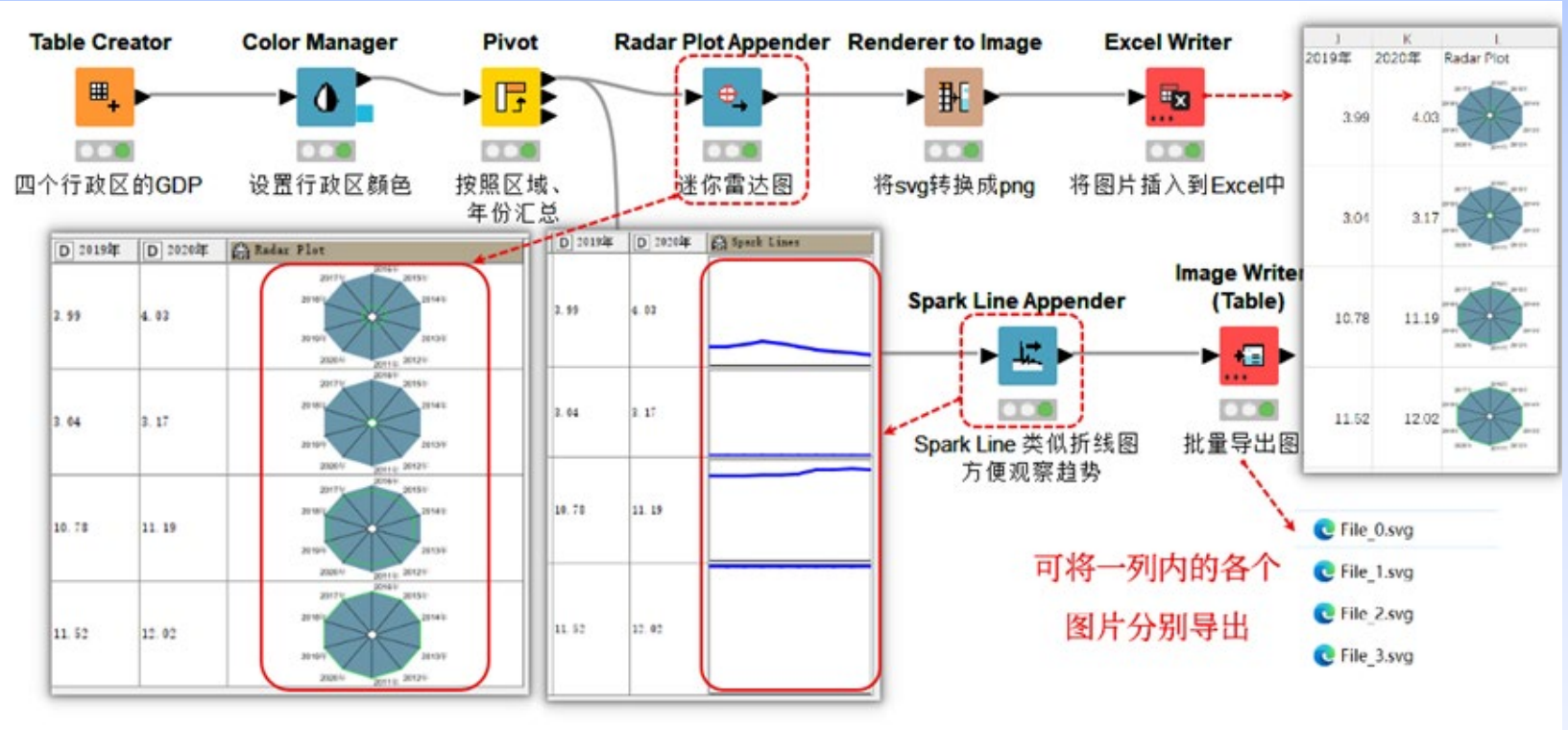
1、导出图片

完成可视化后，用户可以将KNIME视图类节点生成的图片或图表导出为常见格式（如SVG、PNG），方便在报告中引用或进一步传播。常见两种导出方式，具体如下：



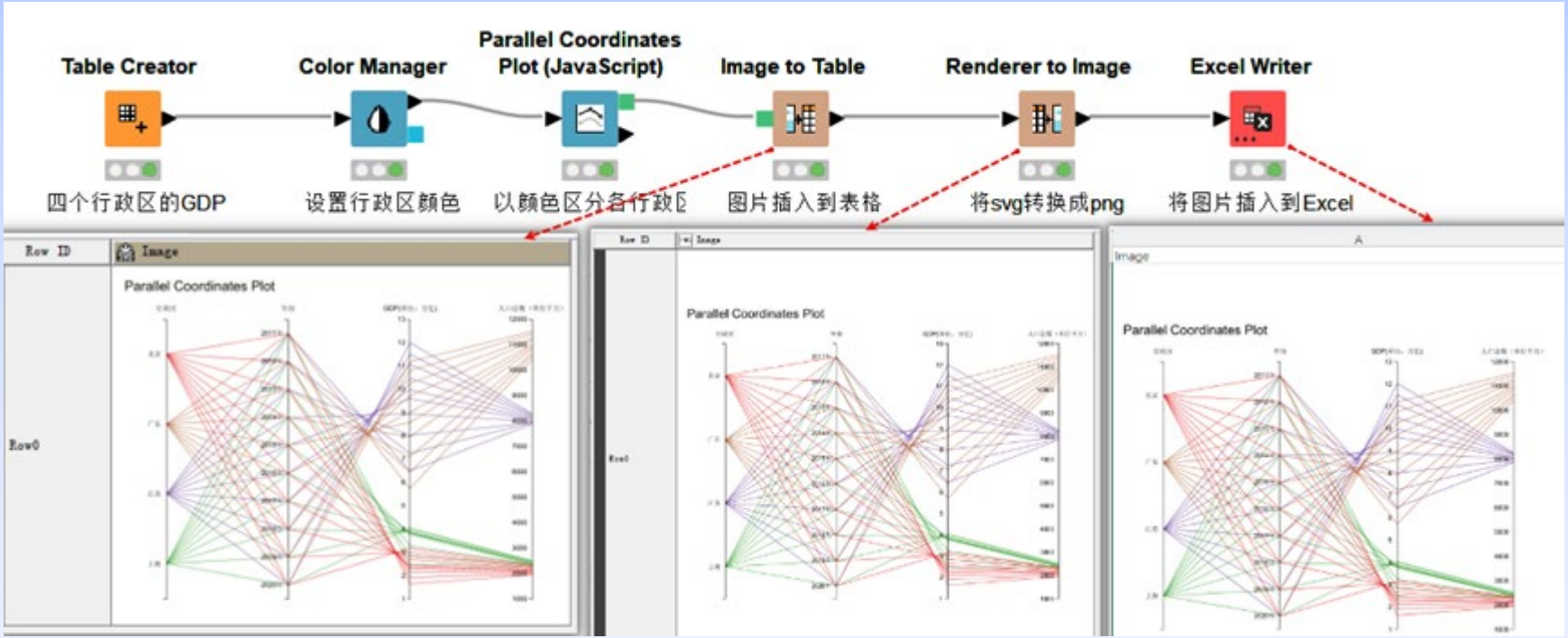


2、图片插入到单元格



使用Image Writer（Table）节点可将单元格中各个图片批量导出

2、图片插入到单元格



将任意可视化结果插入到Excel单元格中

3、图片集成到仪表盘或报告中

详见第13、14章相关内容

这些内容往往涉及多表联动，条件设置，流变量处理，甚至循环，等这部分内容学完，再回头看节点配置的部分选项就可能更清楚，因为有些选项是为了与其他节点交互和关联而存在。



06

本章小结

关键知识点回顾

01

数据可视化定义与意义

数据可视化通过图表、图形等形式展示数据，增强理解和解释。它提供了一种直观、易于沟通的方式，帮助人们探索数据、发现规律和趋势，从而提高数据分析的效果和效率。

02

常见可视化技术与应用

折线图常用于显示时间变化趋势，饼图展示类别占比关系，散点图显示变量之间的关系。这些图表类型在数据分析中广泛应用，有助于直观展现复杂的数据结构。

03

可视化设计基本原则

合理选择图表类型、遵循设计原则如颜色使用、标签和标题设置等，是成功数据可视化的关键。正确的设计能够有效传达信息，提升数据的易读性和说服力。

关键知识点回顾


数据可视化
图形分类
绘图类节点
KNIME视图节点 —— 常用于报告中
JavaScript类 —— 常用于动态仪表盘
JavaScript类试验性节点 —— Plotly交互图
JFreeChart（不推荐使用）
辅助绘图类节点 —— 设置图形颜色、类别等
数据准备 —— 有些节点必须事前汇总加工，或调整数据结构
基于维度
一维可视化节点
绝对数量对比
条形图
词云图
相对数量比 —— 饼图
单维度数据分布
直方图、密度图
箱线图、小提琴图
二维可视化节点
数值与数值 —— 散点图、散点图矩阵
数值与类别 —— 折线图、堆叠图、气泡图、热力图等
多维可视化节点
平行坐标图
雷达图、迷你雷达图
火花线图（迷你小图）
三维散点图
其他类可视化节点
数据探索类 —— Data Explore、Statistics、Statistics View
模型评估相关图 —— Lift Chart、ROC Curve
通用类绘图节点 —— 使用JavaScript代码、或Python代码
与仪表盘或报告相关节点 —— Text View、Table View、Tile View等
可视化节点的设置 —— KNIME视图类节点、JavaScript类节点
绘图结果应用
导出单个图片
图片插入到单元格中
多个批量插入 —— Appender类节点
单个插入 —— 图形类型转换等节点配合操作
构建多图表联动的仪表盘
文字与图表结合的格式化报告

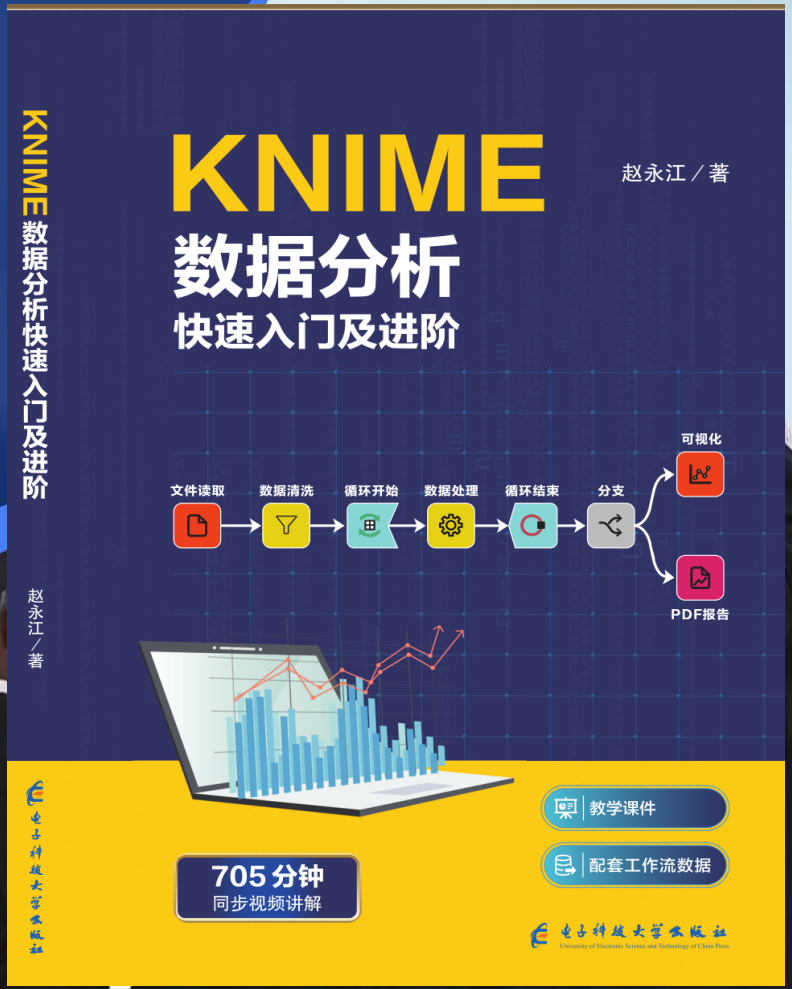
感谢观看

THANKS



第九章 流变量

深入理解流变量类型及应用，进阶必备基础



目录

01 流变量的类型及端口

02 创建流变量

03 查看流变量

04 编辑和使用流变量

05 流变量与Component

06 本章小结



01

流变量的类型及端口

流变量类型

1、基于数据类型

（一）按照常用数据类型划分，流变量分类如下：

（1）数值型。如整型（Integer，Long）、浮点型（Double）。

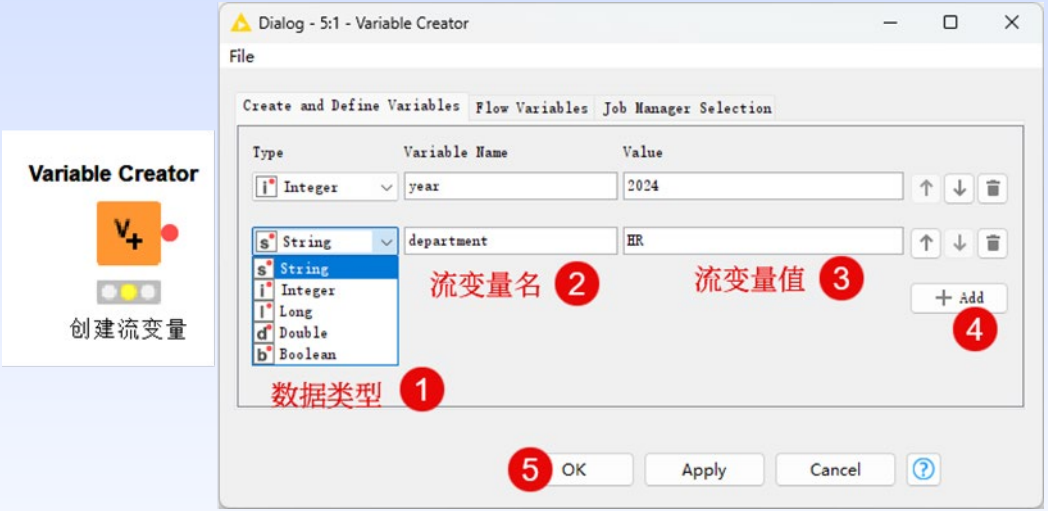
（2）布尔型。流变量取值true或false。

（3）字符串型。常规的字符串类型，如部门名称等。

（4）日期时间类型，如节点Date&Time Configuration。

（5）集合类型。该类型流变量中具有多个元素，如节点Column Selection Configuration。

（6）路径类流变量。如常见的目录路径、文件路径等，如节点Local File Browser Configuration。

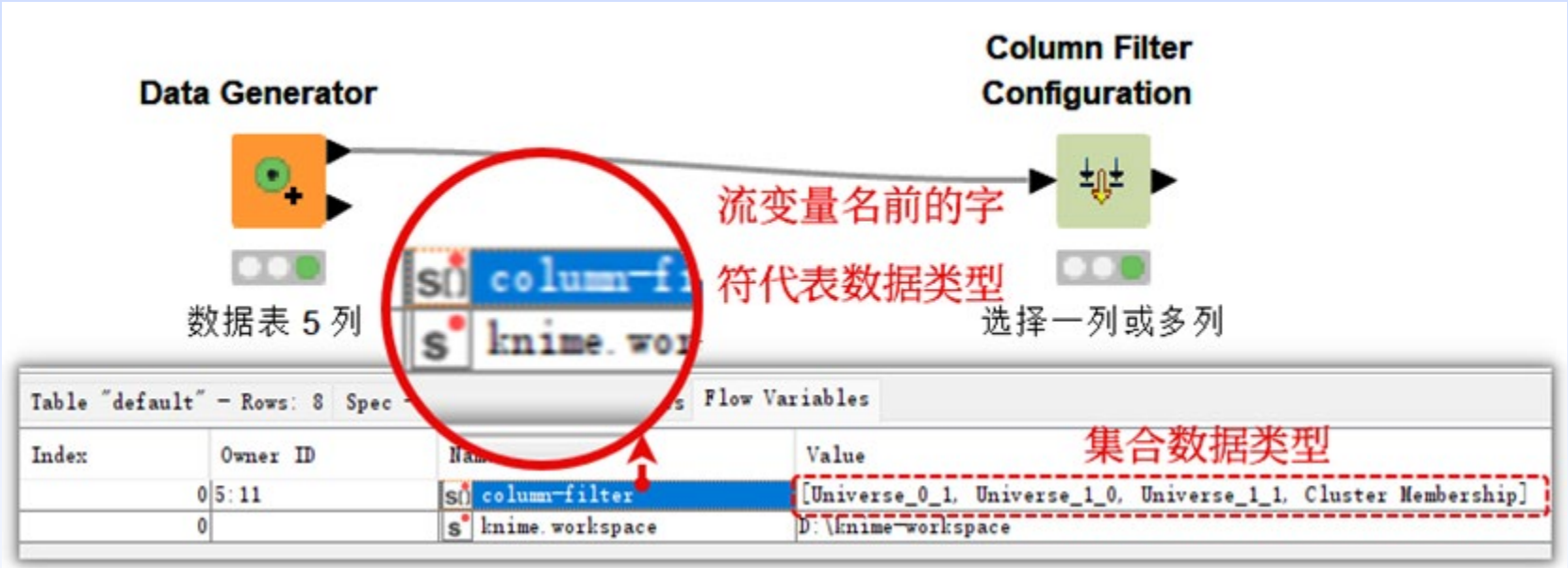


流变量类型

2、基于流变量对应值多少

（1）单值流变量。如利用节点Variable Creator创建一个名为“department”的流变量，其对应的值为“HR”。

（2）多值流变量，也可以称为集合类型流变量。主要指集合类型的流变量，为列表（List）或集合（Set）类型，可统称为Collection或Array类型

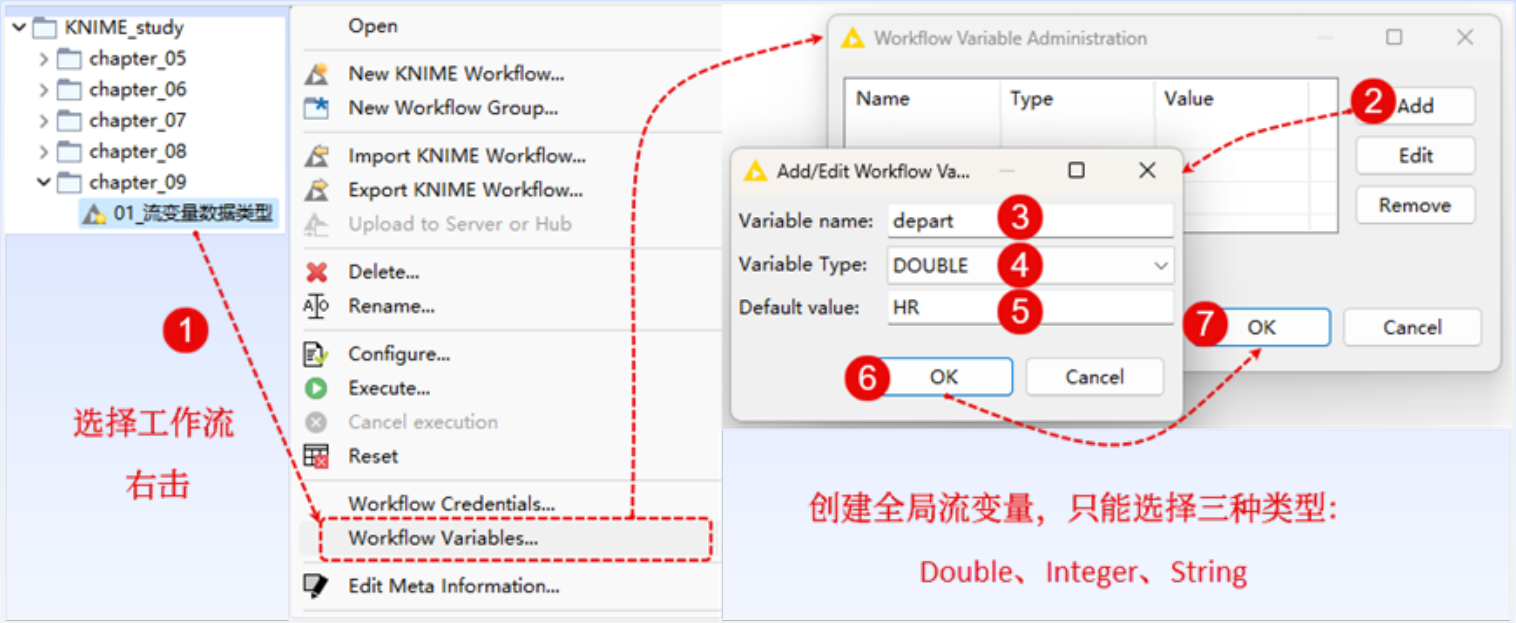


流变量类型

3、流变量作用域

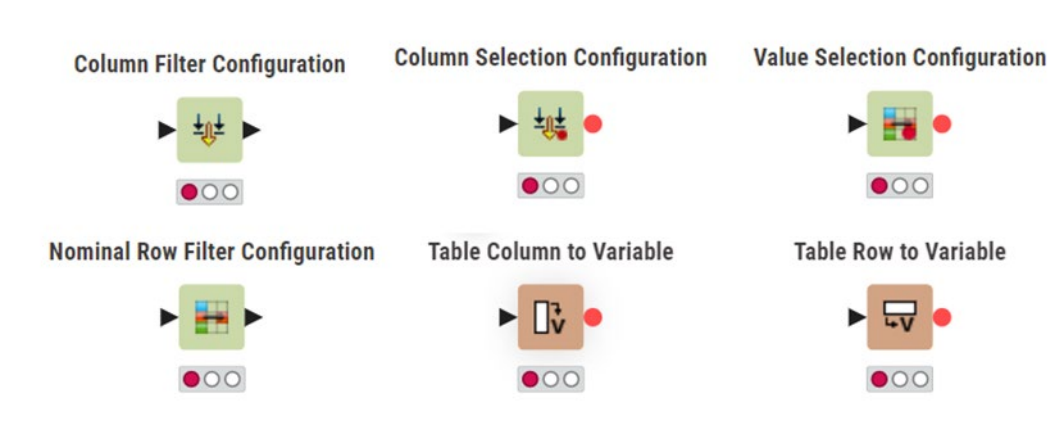
（1）局部流变量。最为常见的类型，如循环内部的流变量，其起作用的范围一般仅在循环体内。

（2）全局流变量。主要指该流变量在整个工作流中任何一个节点都可以查看或使用。最典型的流变量为knime.workspace，每个节点流变量标签页内都存在。（下图操作仅限 KNIME**经典界面**）



流变量类型

4、流变量的数据来源



（1）来源于数据表的流变量。

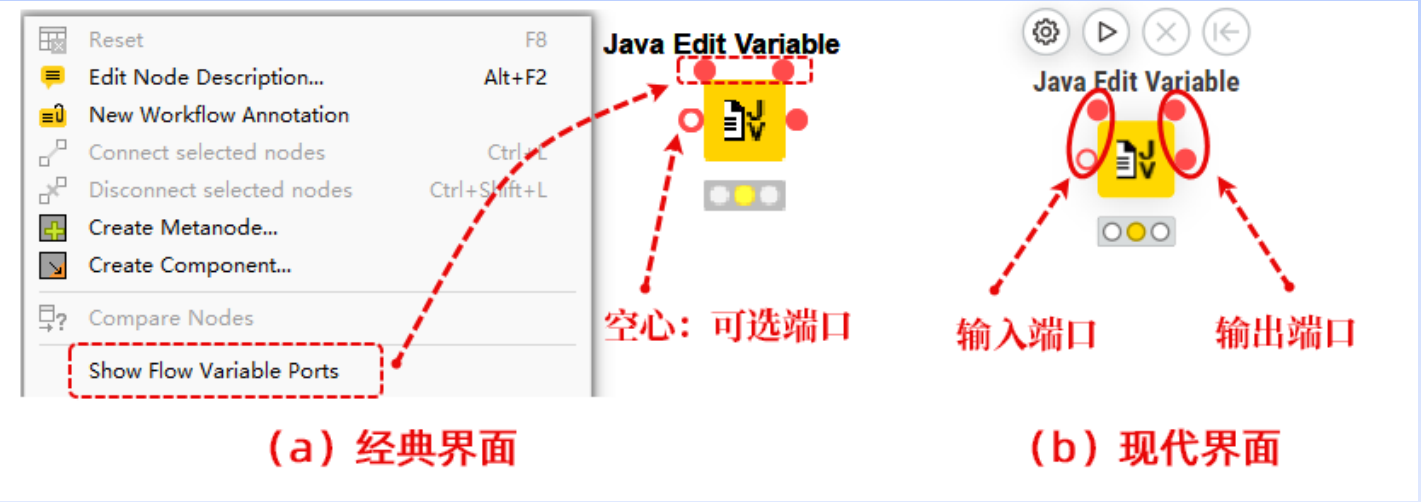
（2）节点配置选项内参数转为流变量



（3）手动自定义流变量

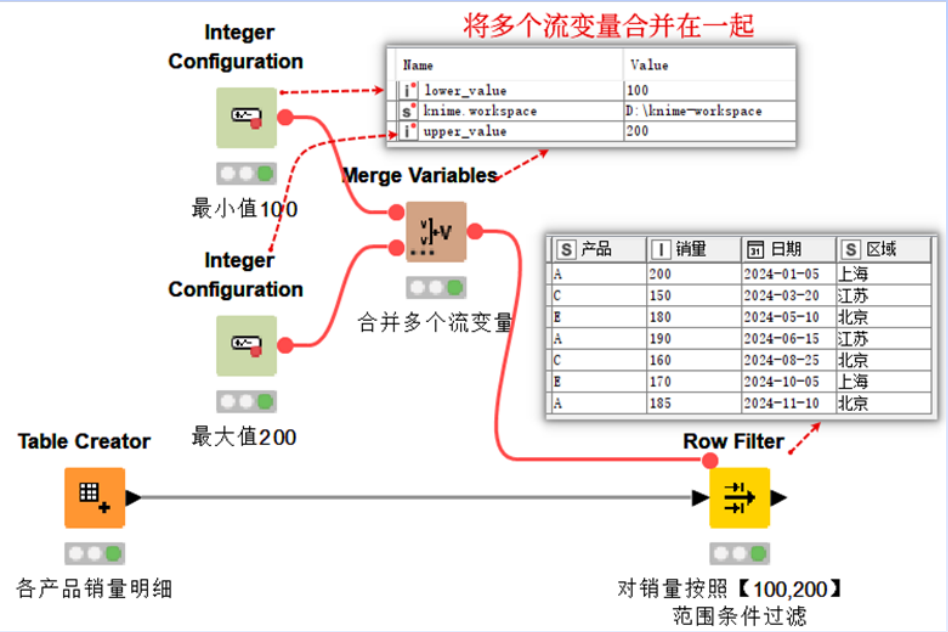
Configuration类和Widgets类中的很多节点，即可以手动输入，也可以通过来源于数据表中的数据创建流变量

流变量端口

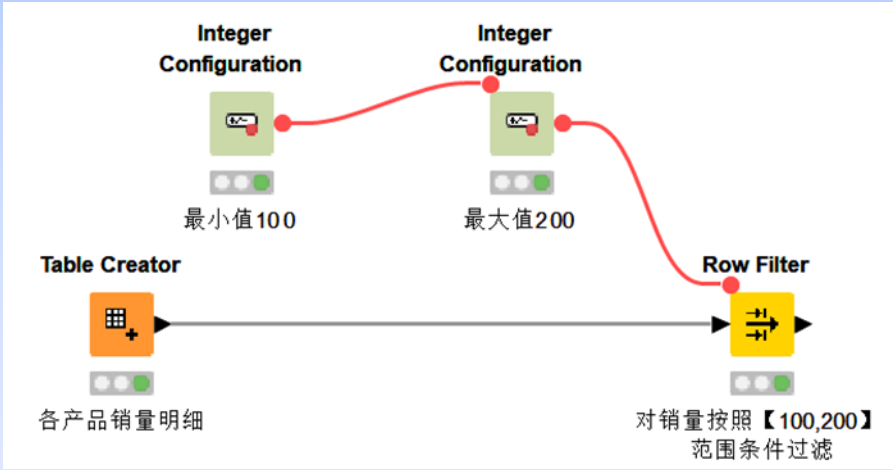


- 经典界面下，默认不显示流变量端口
- 现代界面下，鼠标悬停节点处即显示端口
- 左侧及左上角为输入端口，右侧及右上角位输出端口

流变量之间的并联、串联连接形式

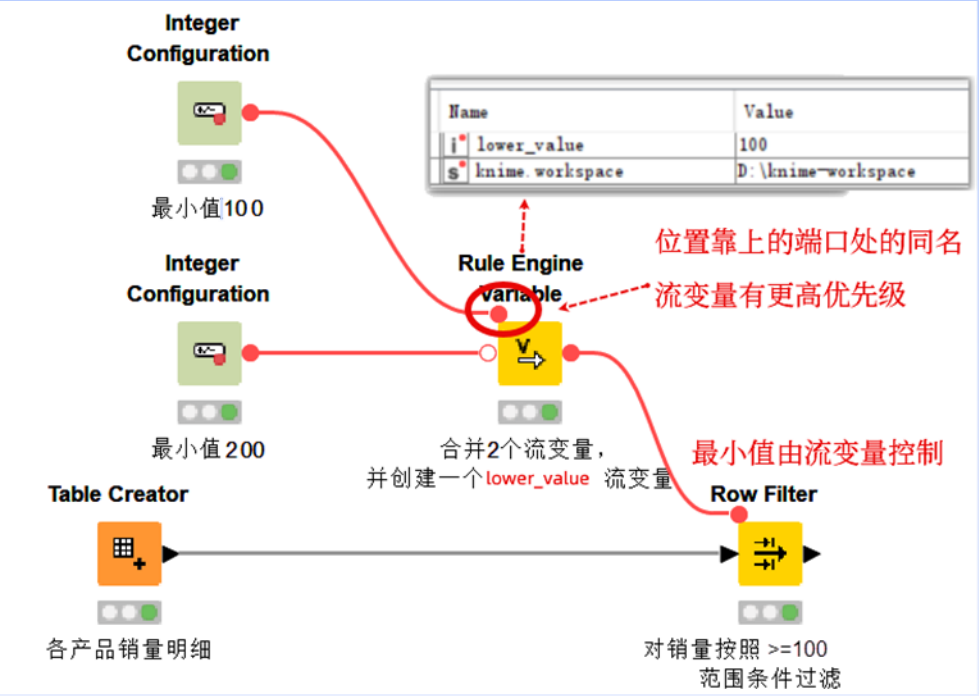


并联形式

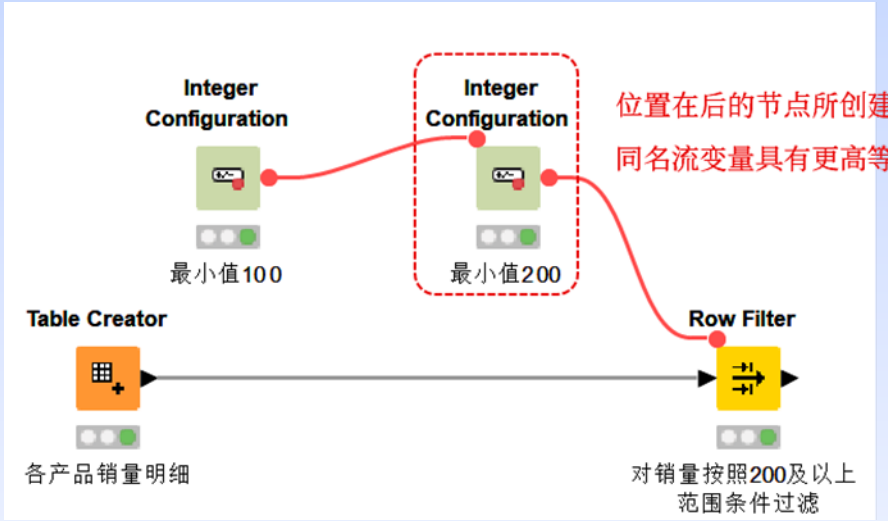


串联形式

同名流变量之间的并联、串联优先级问题



并联形式



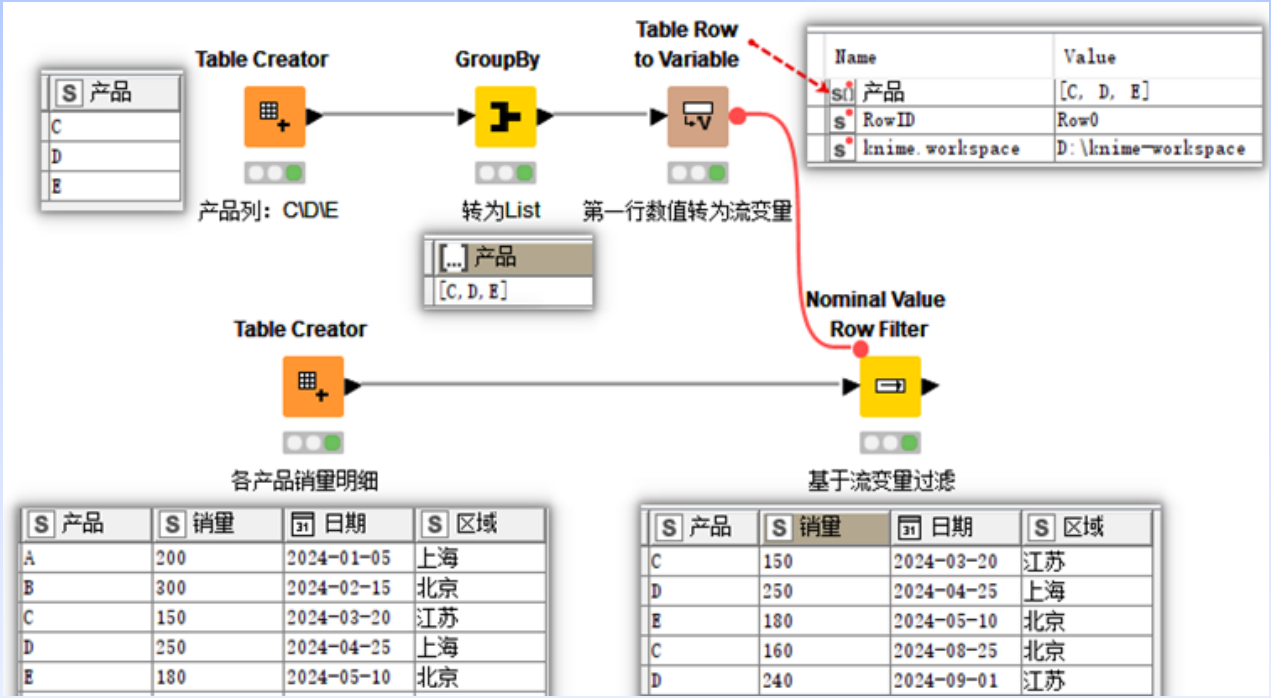
串联形式

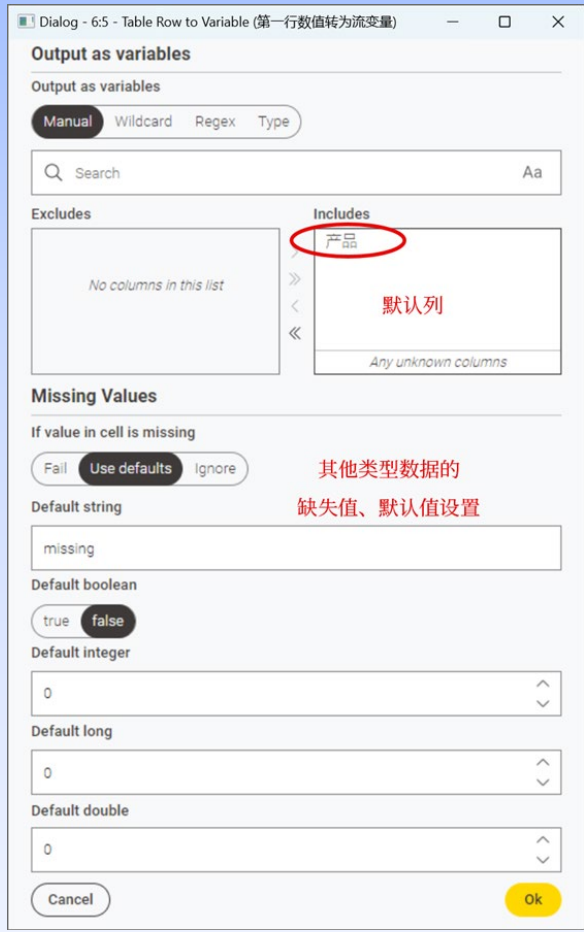


02

创建流变量

创建集合类型流变量—示例

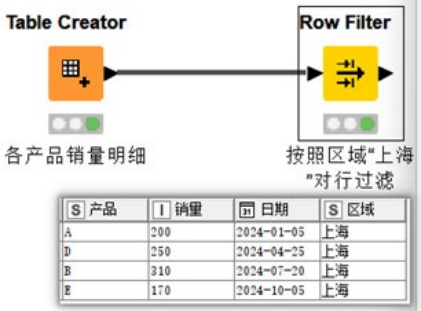




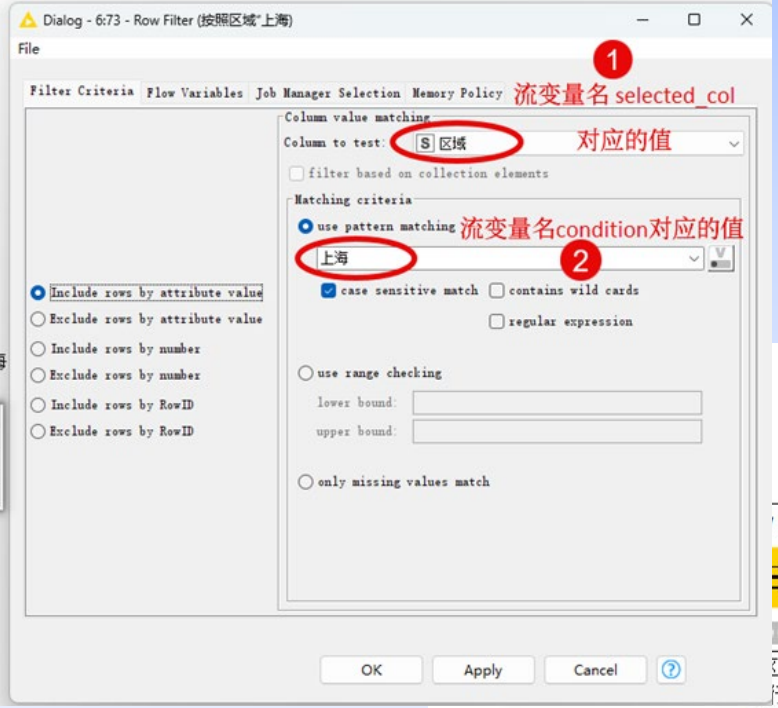
使用流变量实现对Nominal Row Filter节点参数的控制

1. 首先利用GroupBy节点中的List或Set函数创建包含多个元素的List或Set类型的数据
2. 然后通过Table Row to Variable节点将第一行的数据转为流变量，流变量的名称与列名相同
3. 最后传递至Nominal Row Filter

基于节点配置选项参数创建流变量—示例



产品	销量	日期	区域
A	200	2024-01-05	上海
D	250	2024-04-25	上海
B	310	2024-07-20	上海
E	170	2024-10-05	上海



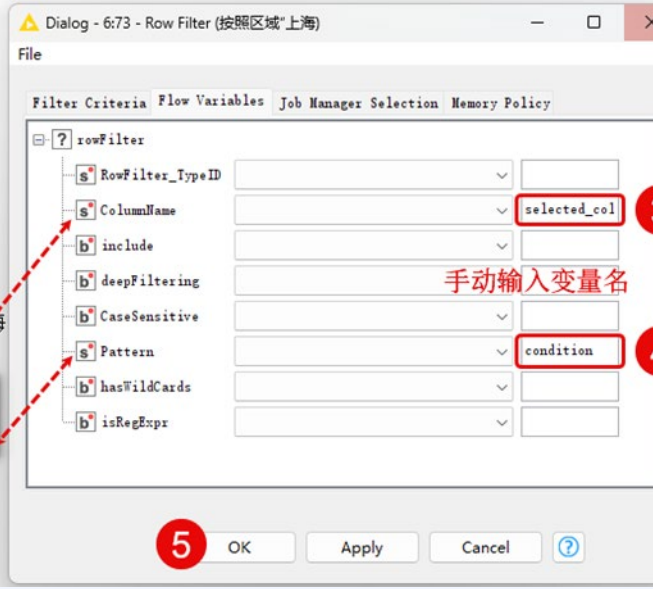
使用流变量实现对Nominal Row Filter节点参数的控制

1. 首先利用GroupBy节点中的List或Set函数创建包含多个元素的List或Set类型的数据
2. 然后通过Table Row to Variable节点将第一行的数据转为流变量，流变量的名称与列名相同
3. 最后传递至Nominal Row Filter

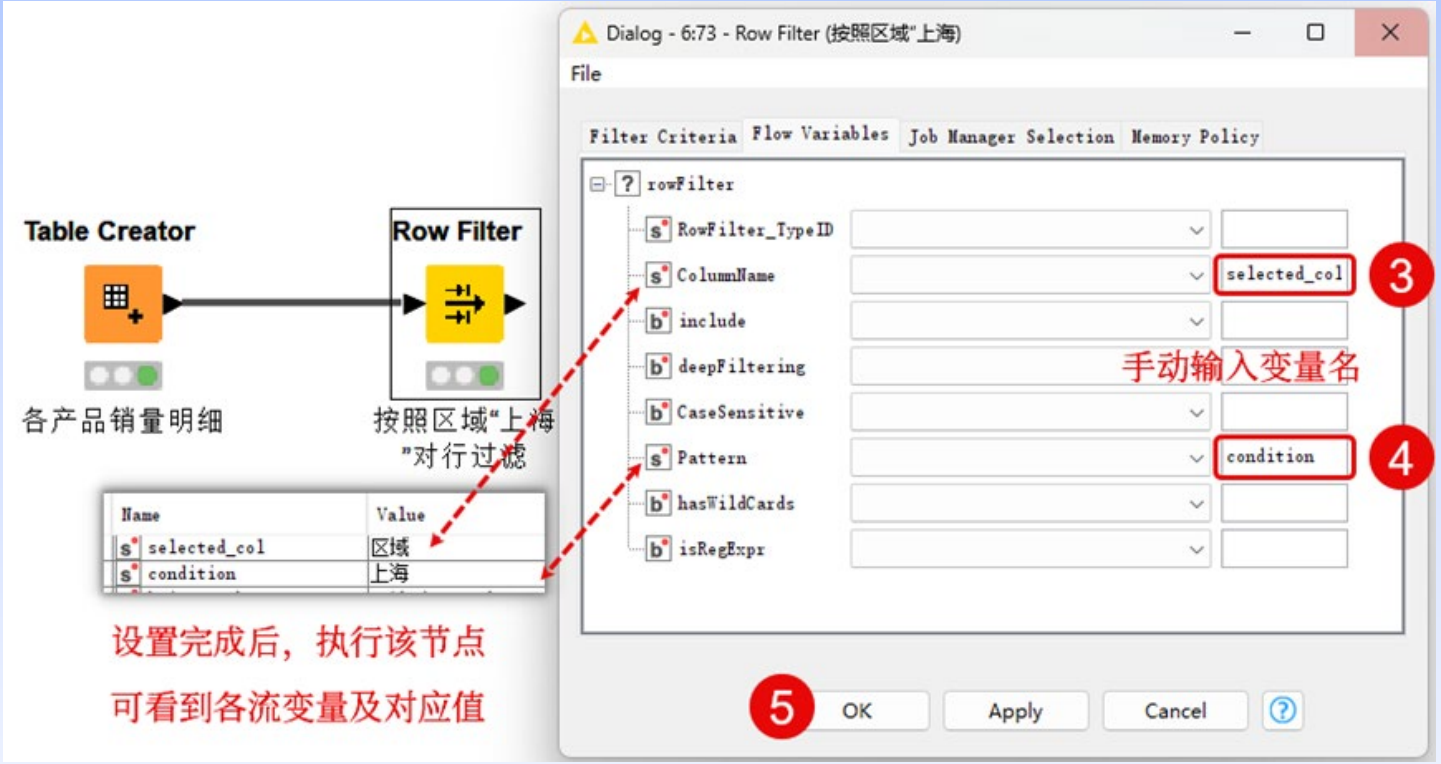
Name	Value
selected_col	区域
condition	上海

设置完成后，执行该节点

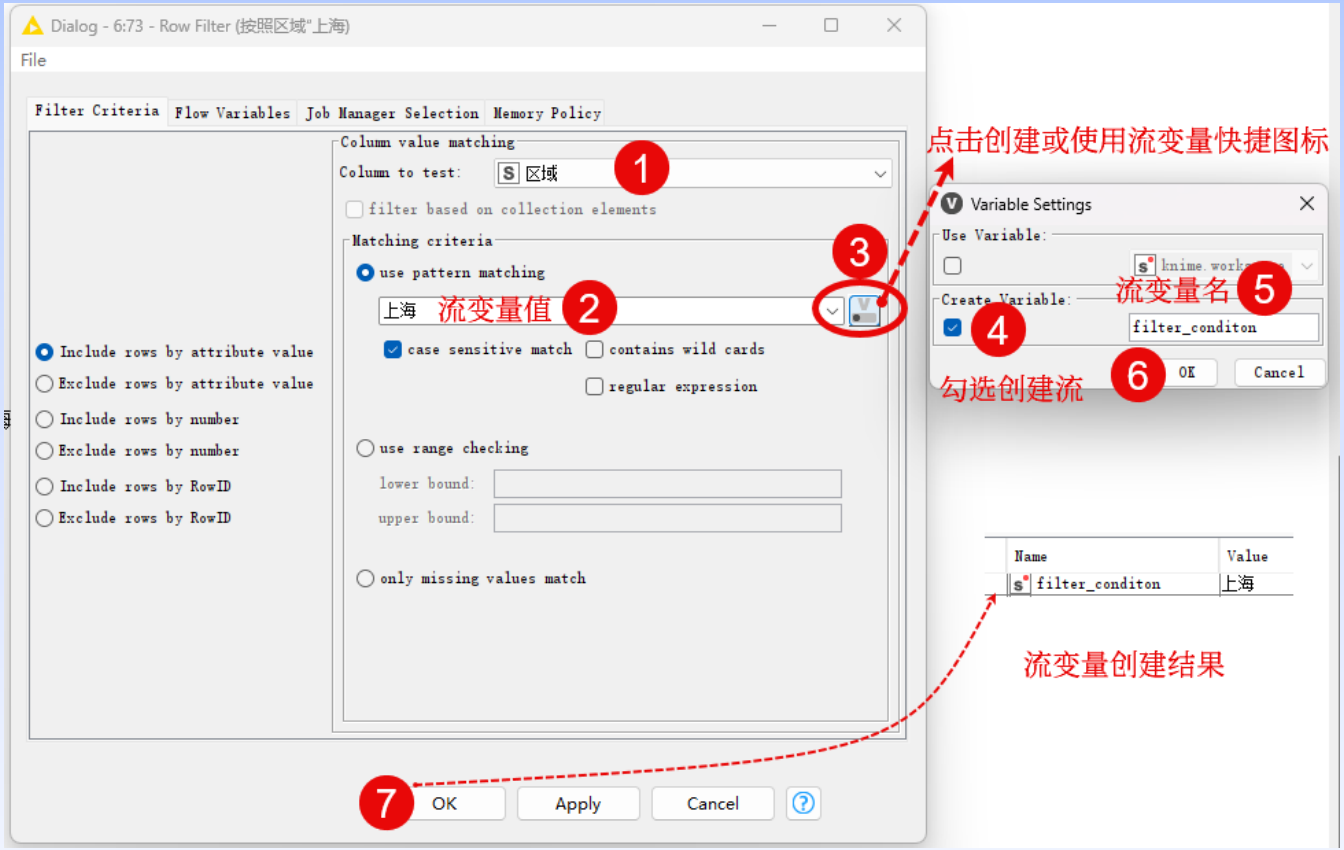
可看到各流变量及对应值



基于节点配置选项参数创建流变量—示例



基于节点配置选项参数创建流变量—示例



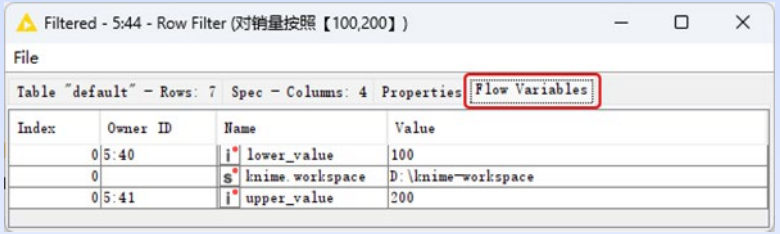
创建流变量的快捷方式。注意：并非所有参数都有对应的快捷图标



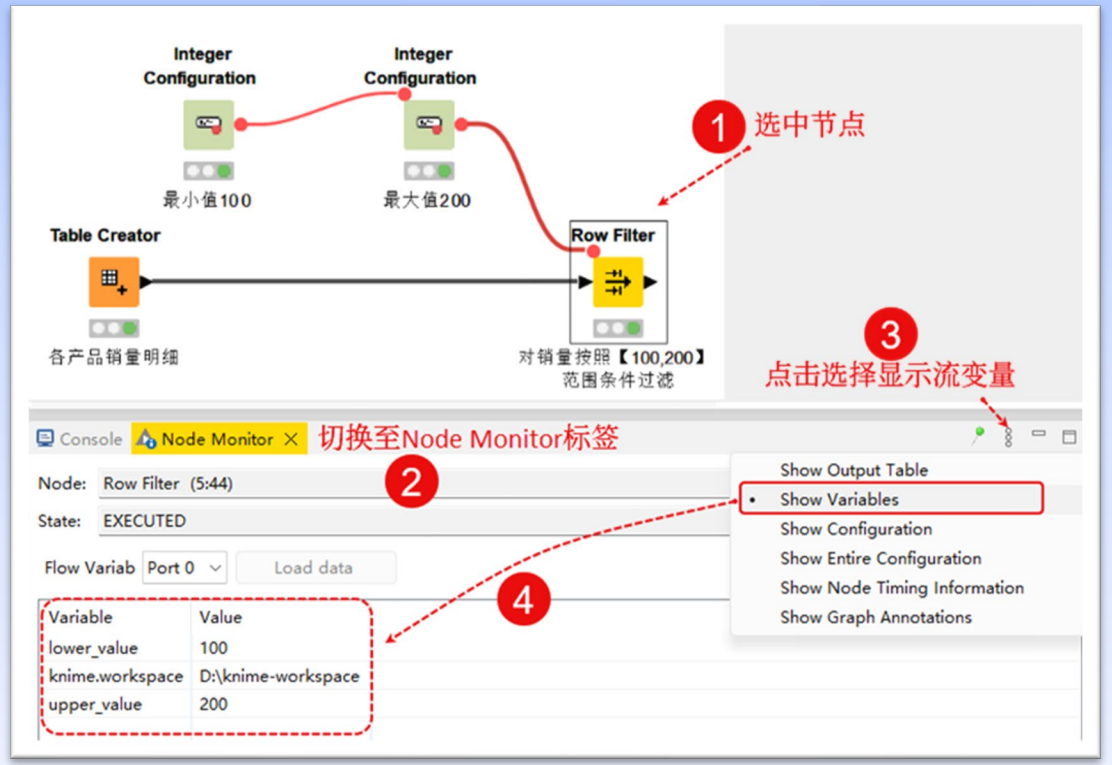
03

查看流变量

经典界面下：查看流变量

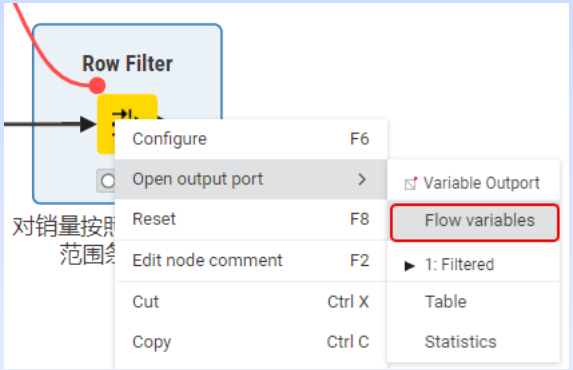


方法1

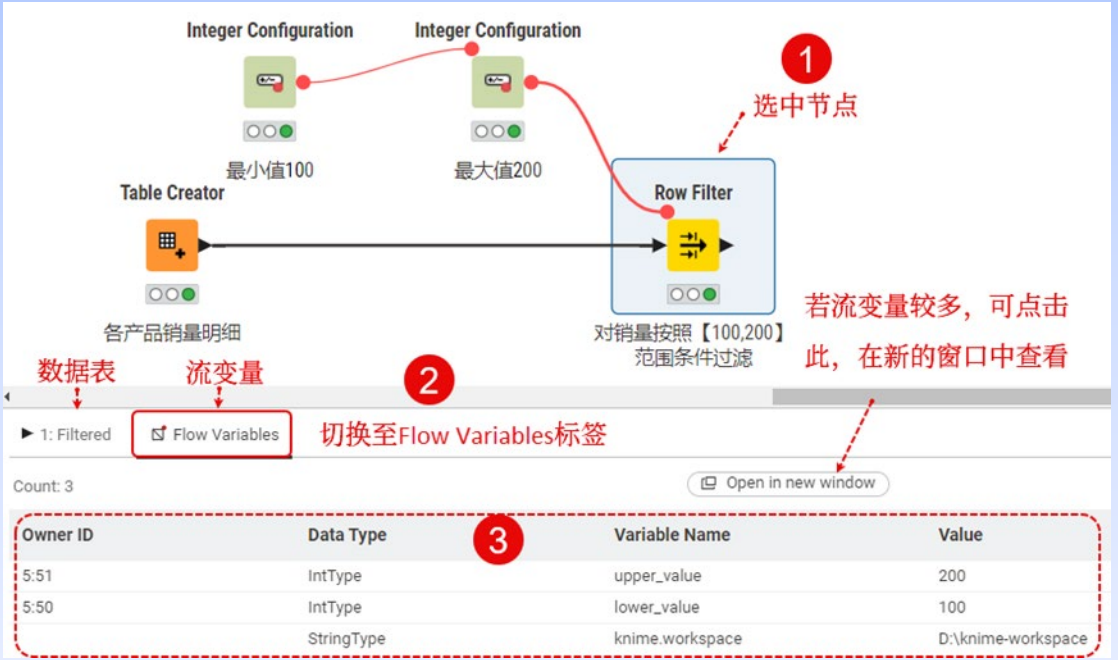


方法2

现代界面下：查看流变量



方法1



方法2

现代界面下查看流变量更加方便，并且可以随时查看数据和流变量，而经典节点下，Node Monitor需要额外切换数据表或流变量标签页

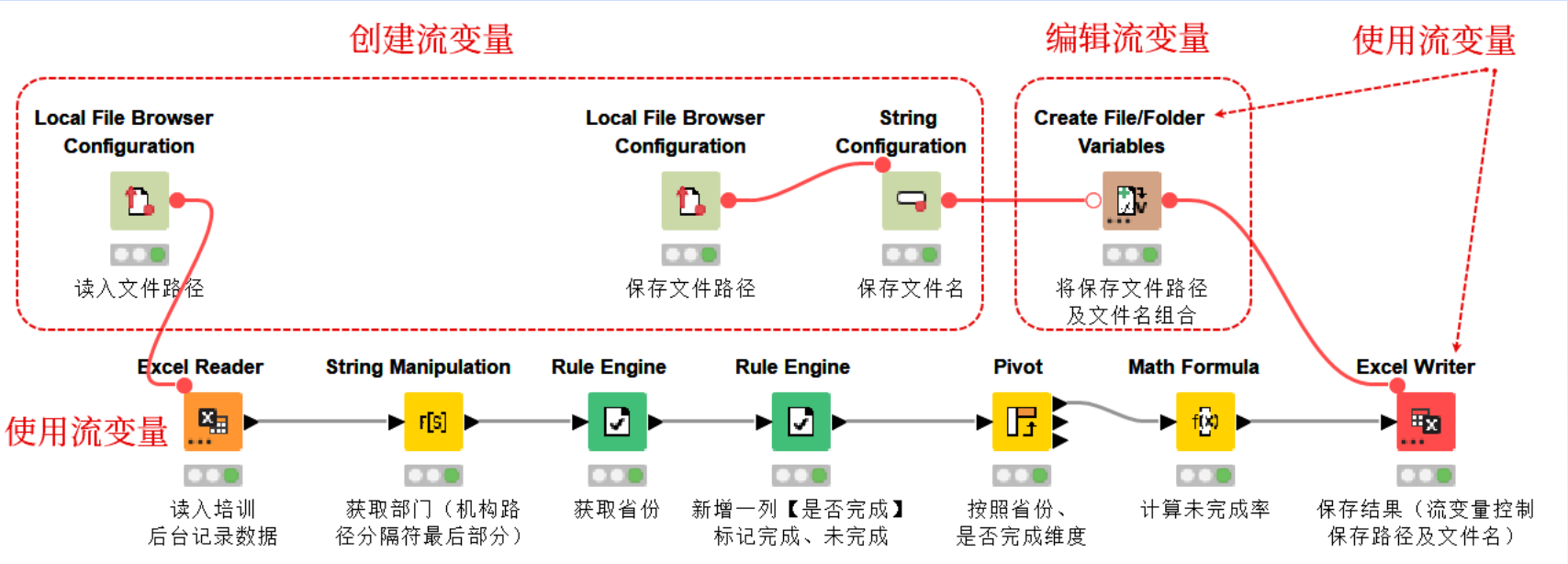


04

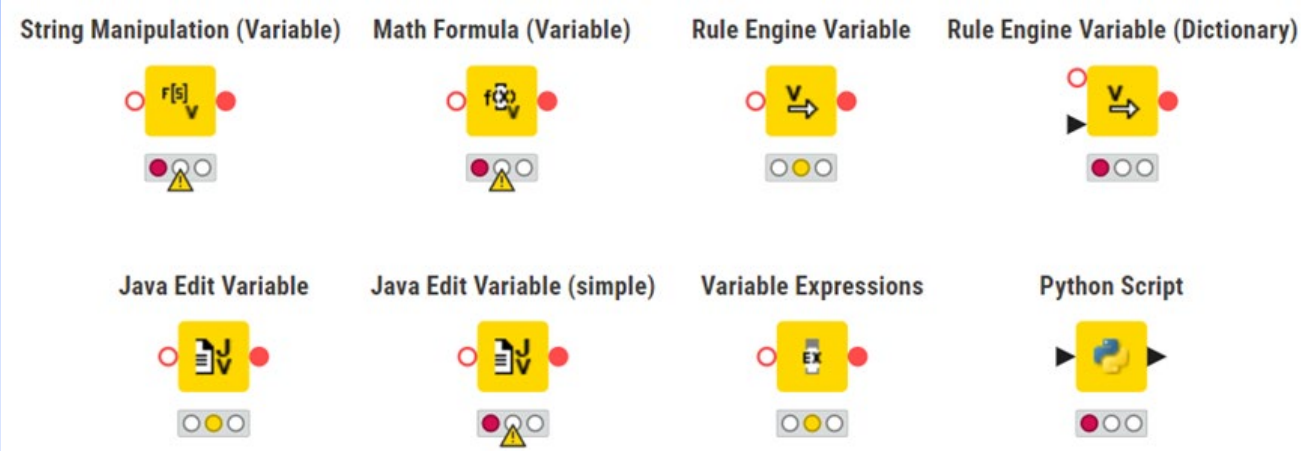
编辑和使用流变量

编辑流变量—示例

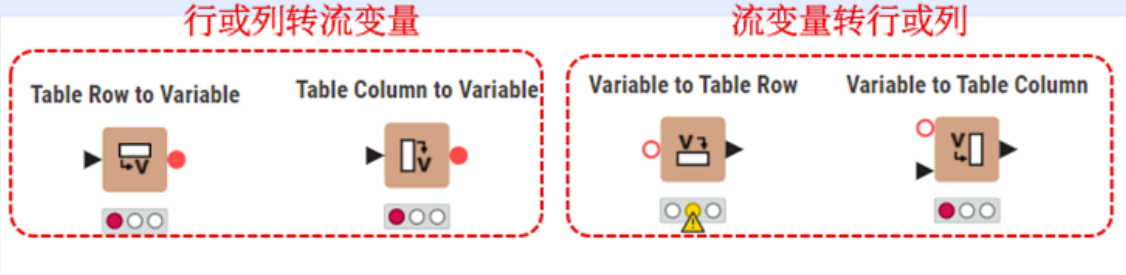
以第七章统计培训完成情况的数据为案例，增加了流变量控制读入文件路径，写入文件目录，新建文件名，并将后两个组合在一起构成待写入文件的路径和文件名，控制Excel Writer节点的写入路径和文件名。



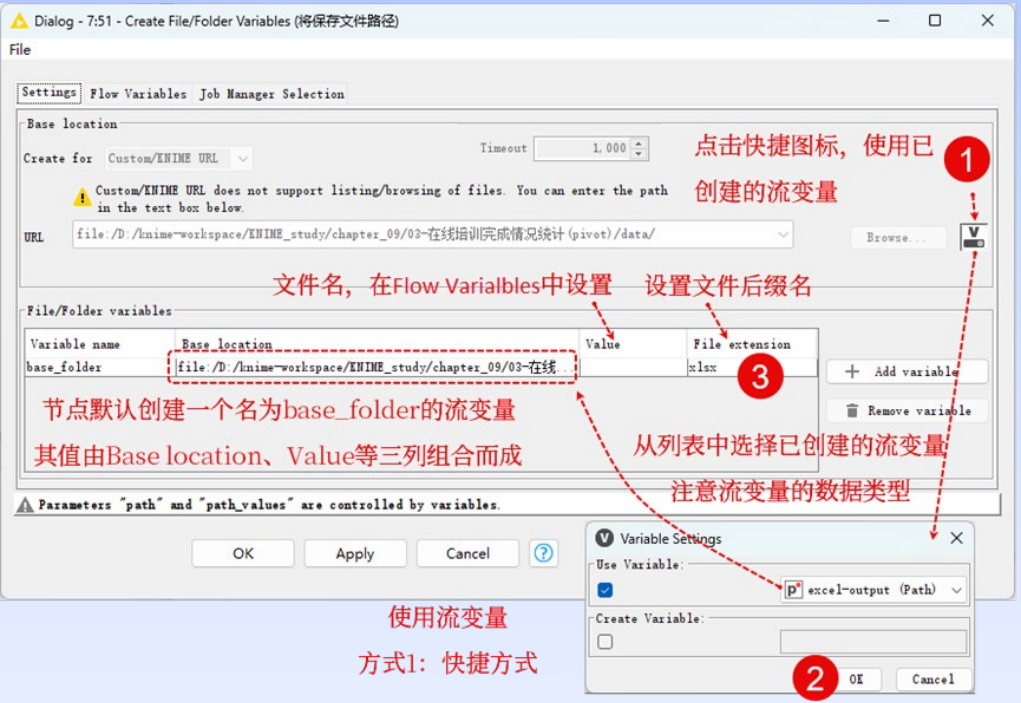
编辑流变量—常用节点

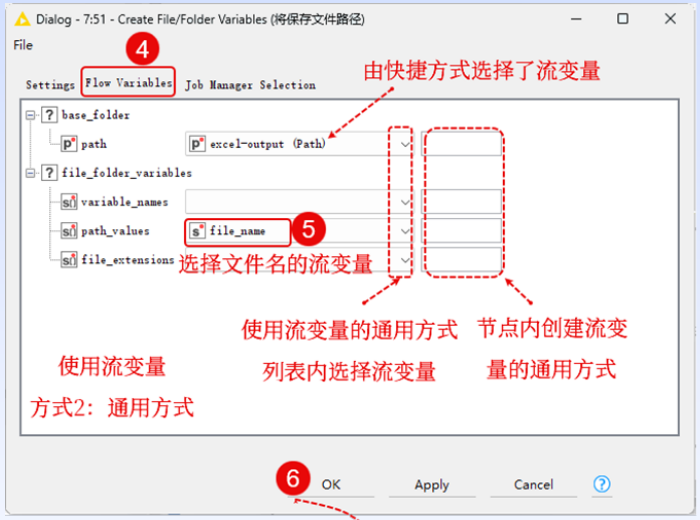


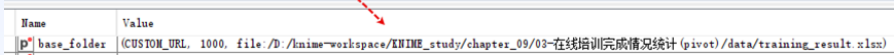
注意流变量可与数据表进行互相转换。这意味着有些操作可先在数据表中操作，得到期望结果后再转为流变量



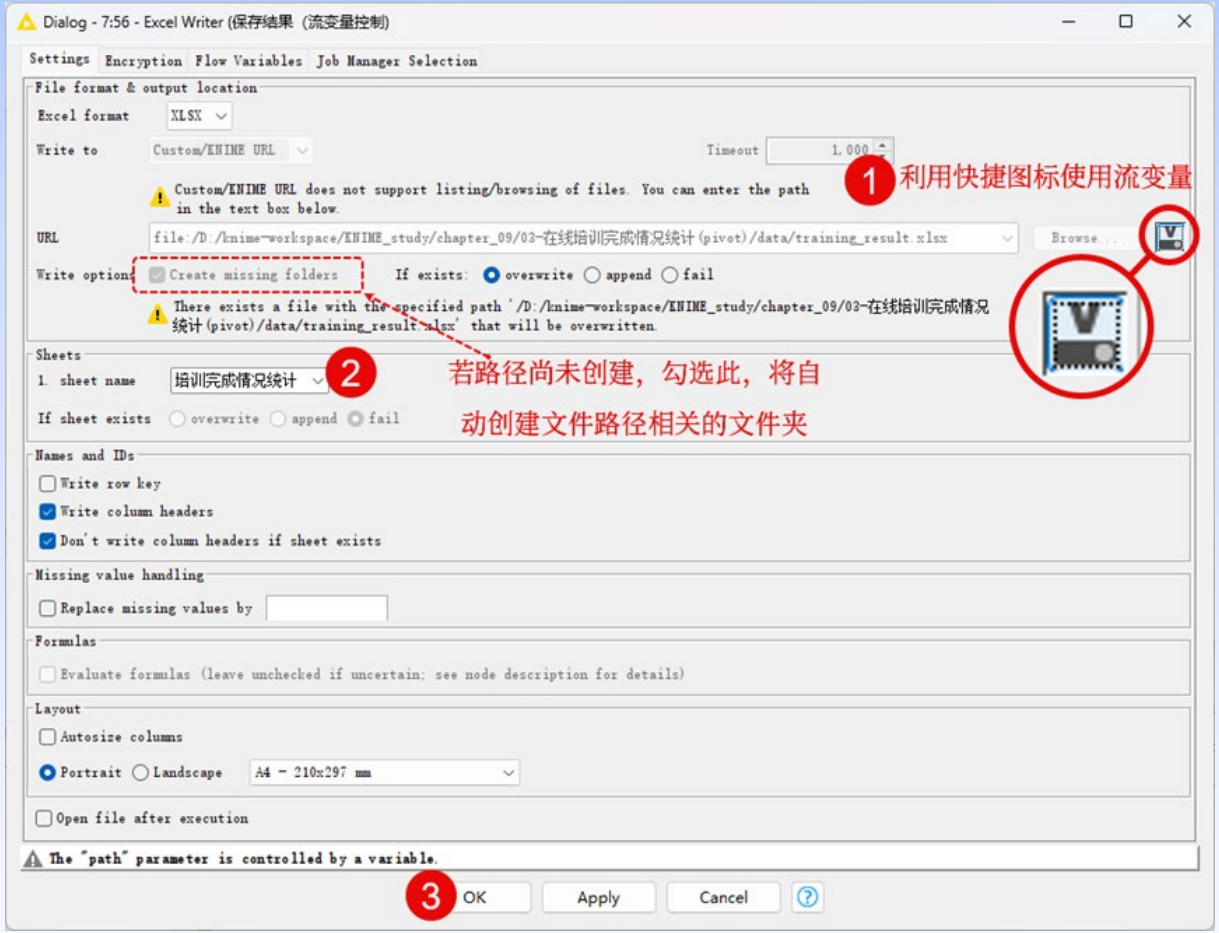
使用流变量（组合文件路径流变量）



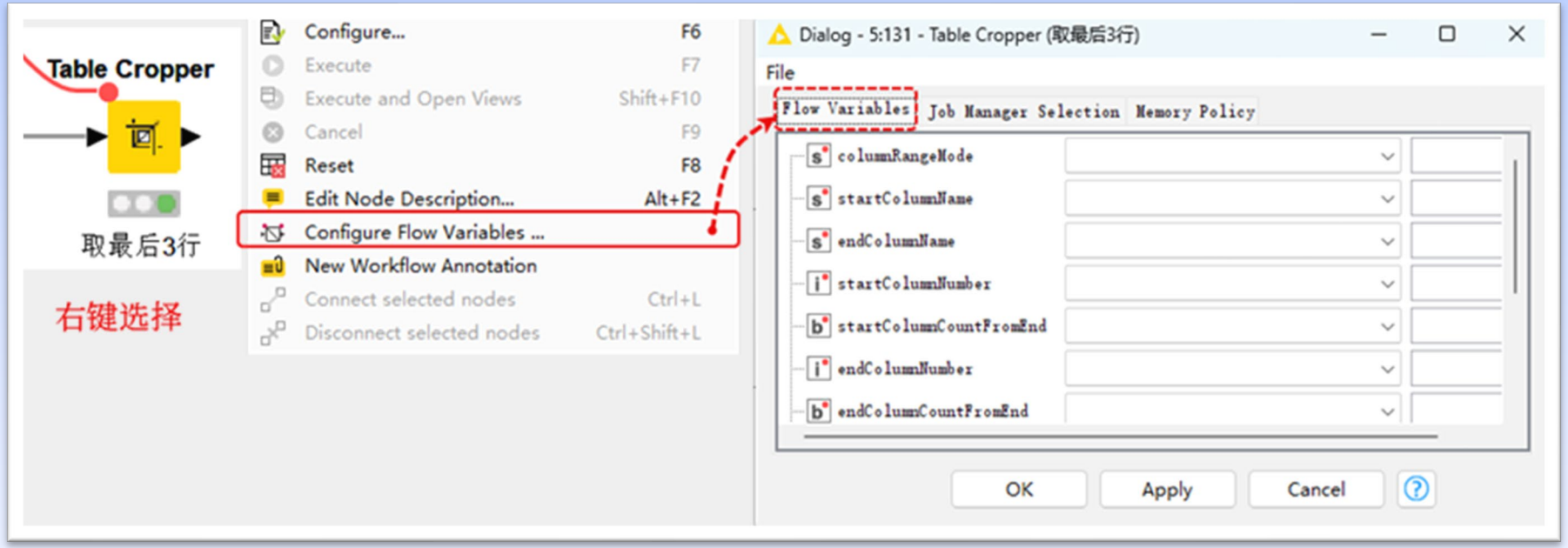




使用流变量（使用文件路径流变量）



现代界面下 使用流变量（与经典界面稍有不同）

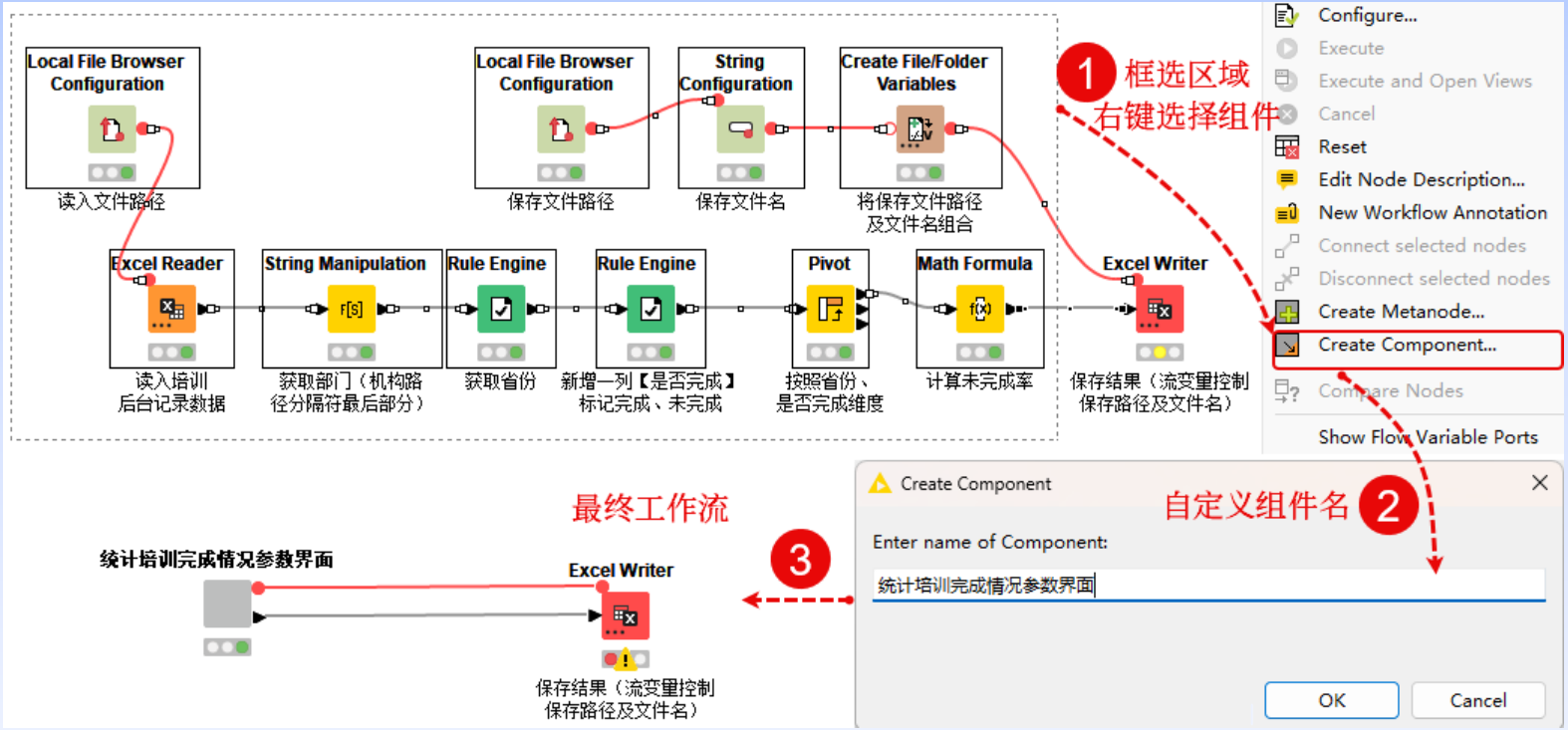




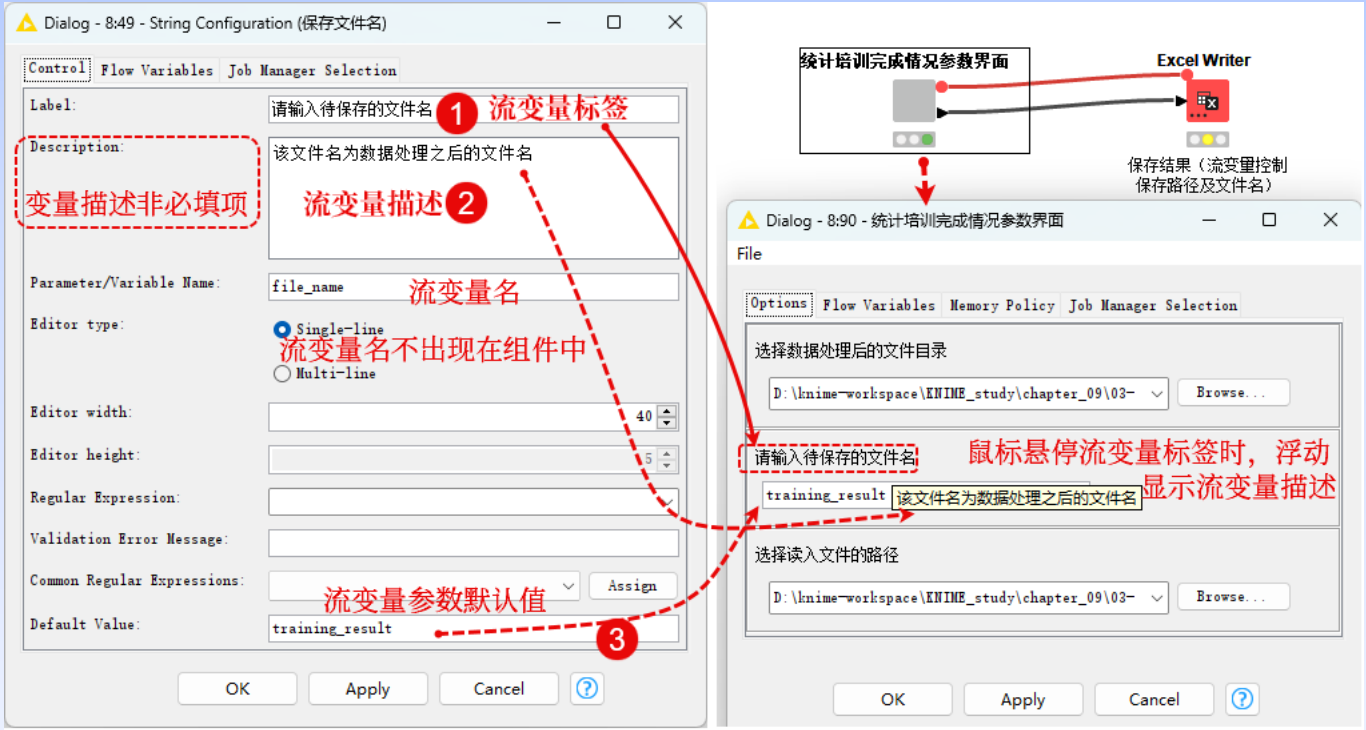
05

流变量与 Component

创建组件Component

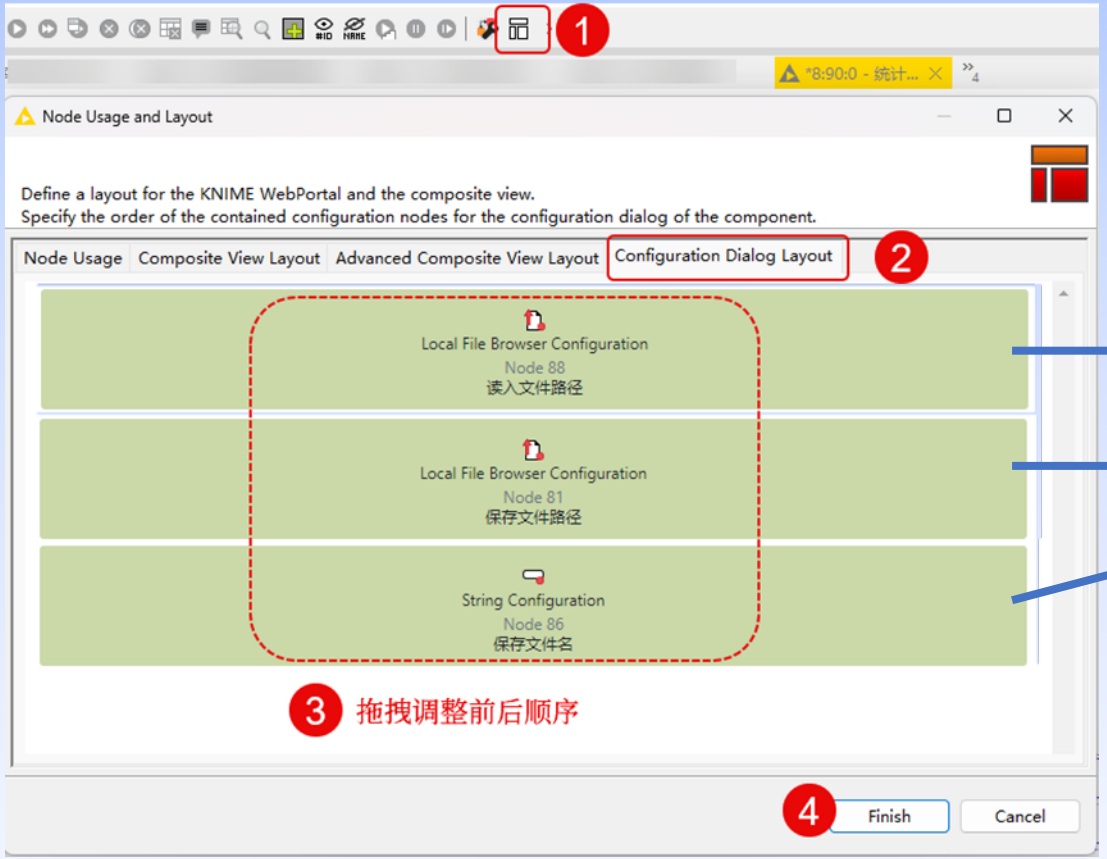


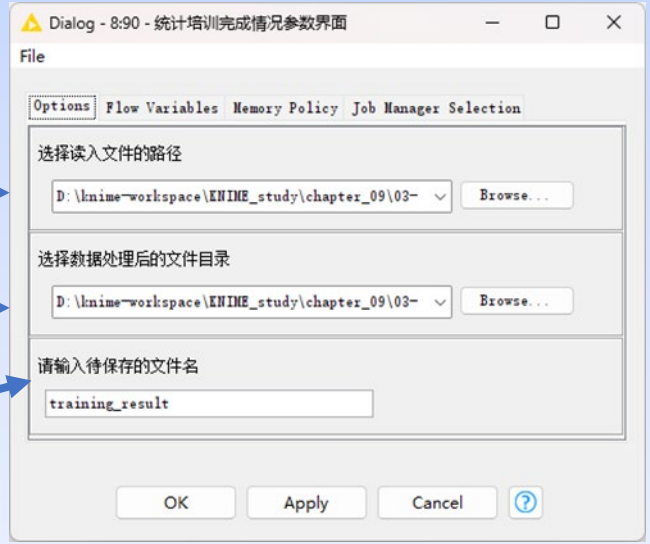
组件内参数与Configuration类节点流变量对应关系



Component参数由Configuration类节点创建，组件窗口内只显示流变的Label内容，不显示流变量名，这样更方便提示用户进行参数设置

调整Component节点参数布局





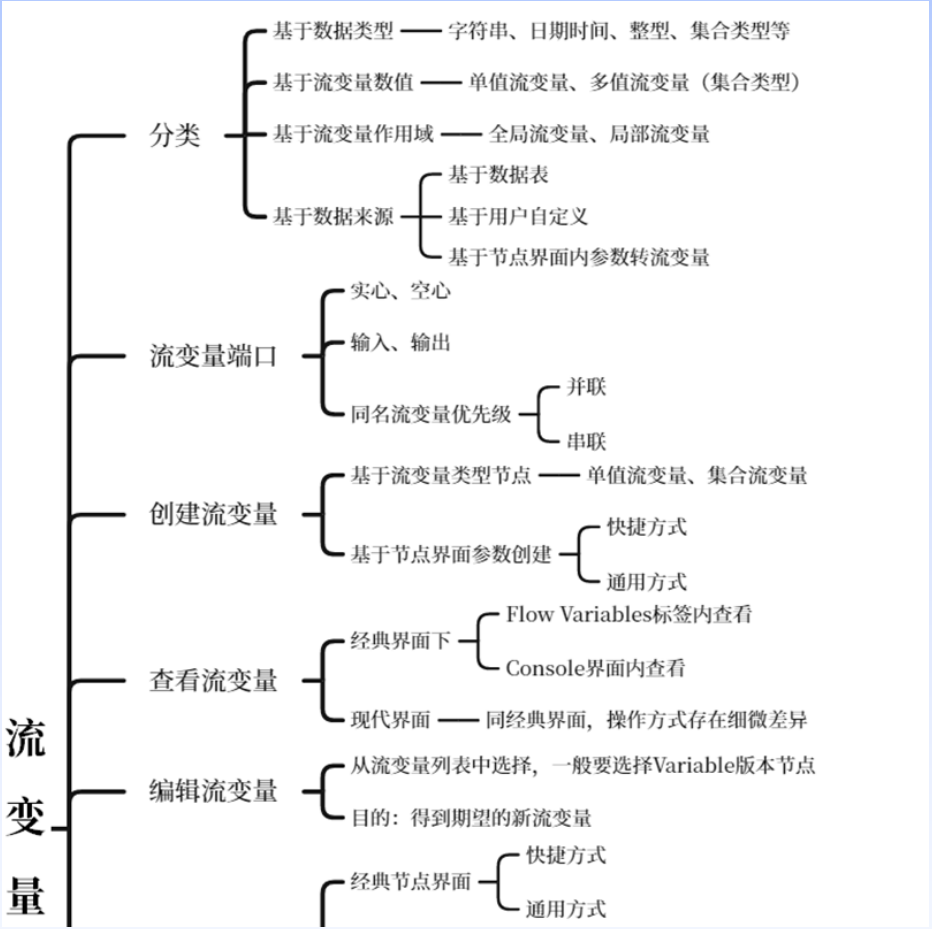
Configuration类节点注释很有必要，否则不易区分，如本截图中的两个Local File Browser Configuration节点



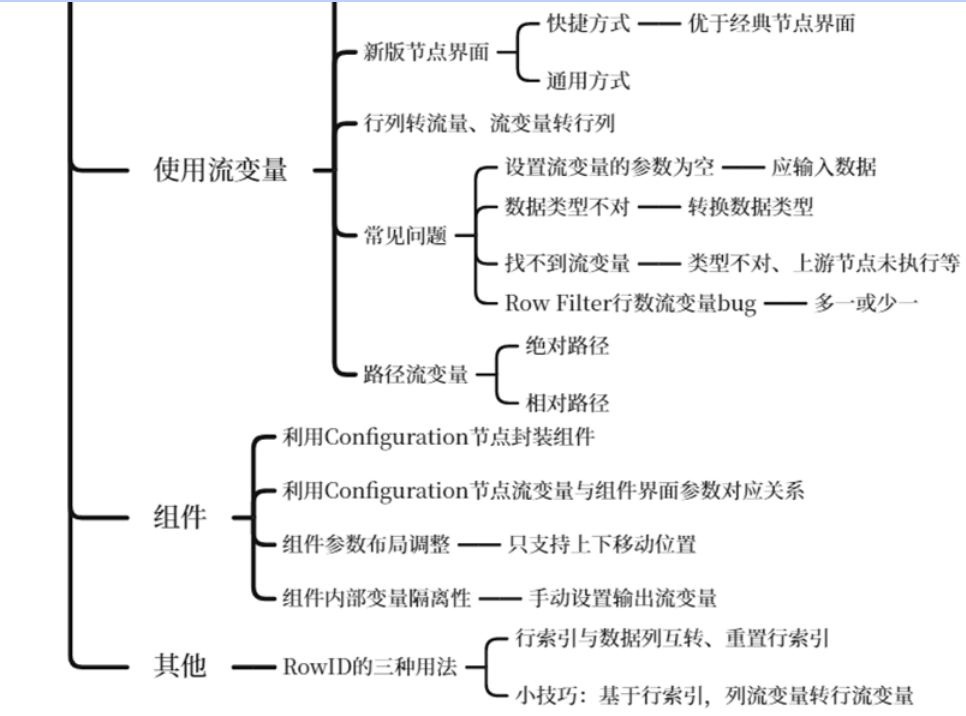
06

本章小结

强调编辑和使用流变量注意事项



强调编辑和使用流变量注意事项



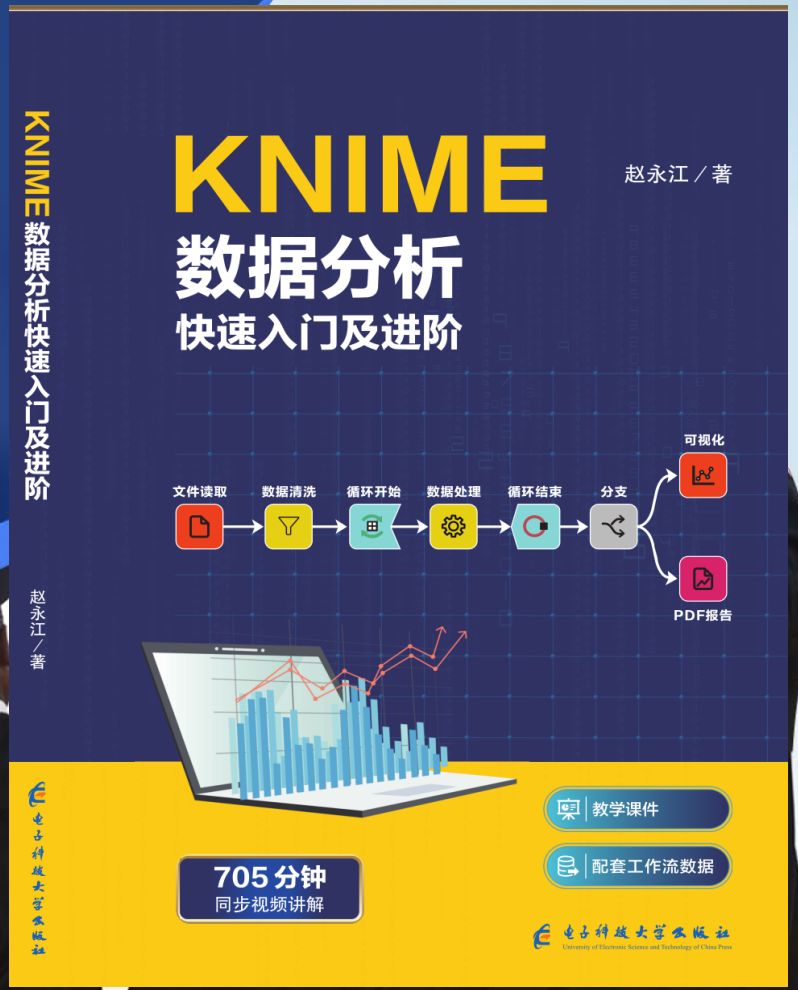
感谢观看

THANKS



第10章 循环

循环概念与应用，进阶必备



目录

01 循环概述

02 计数循环

03 基于行分组循环

04 基于列分组循环

05 条件循环

06 递归循环

07 多个循环间关系

08 循环流程控制

09 常见循环问题

10 带有循环功能其他节点

11 本章小结



01

循环概述

循环基本概念

01

循环定义

循环是编程中反复执行某个动作或某段代码块的一种机制，通过设定初始状态、终止条件、循环体和变量变化等要素，实现对代码的高效复用。它能够简化程序，节约内存并提高运行效率。

02

循环四要素

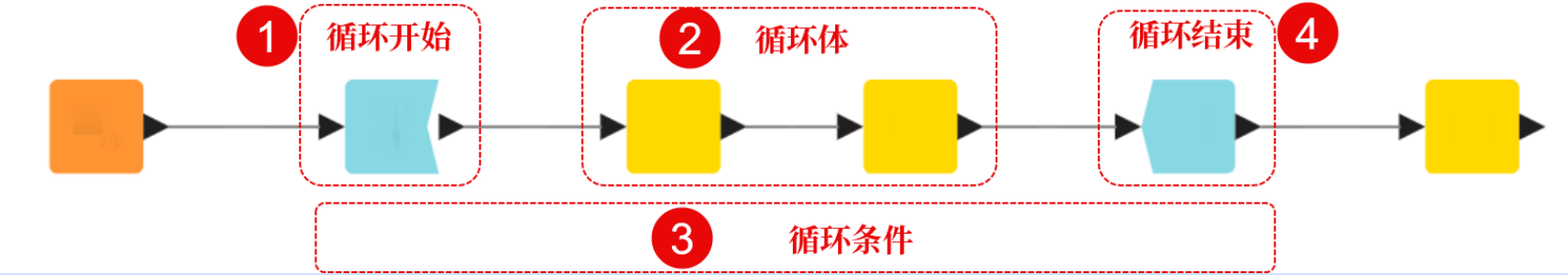
一个完整的循环包括初始状态、终止条件、循环体和变量变化。这些要素共同作用，确保循环能够按照预期重复执行，直到满足终止条件为止。

03

循环分类

循环可以分为计数循环、条件循环、基于分组的循环等多种类型。每种循环根据其不同的控制条件和应用场景，具有各自独特的优势和适用场景。

循环四要素



循环开始

在循环结构中，首先需要定义一个用于计数或标识每次循环开始的变量。这个变量通常从某个初始值开始，并在每次循环中按照特定的增量变化。

循环条件

循环条件是决定循环是否继续执行的关键。常见的循环条件包括基于时间、事件触发或特定数值的判断，通过条件判断确保循环的有目的性和效率。



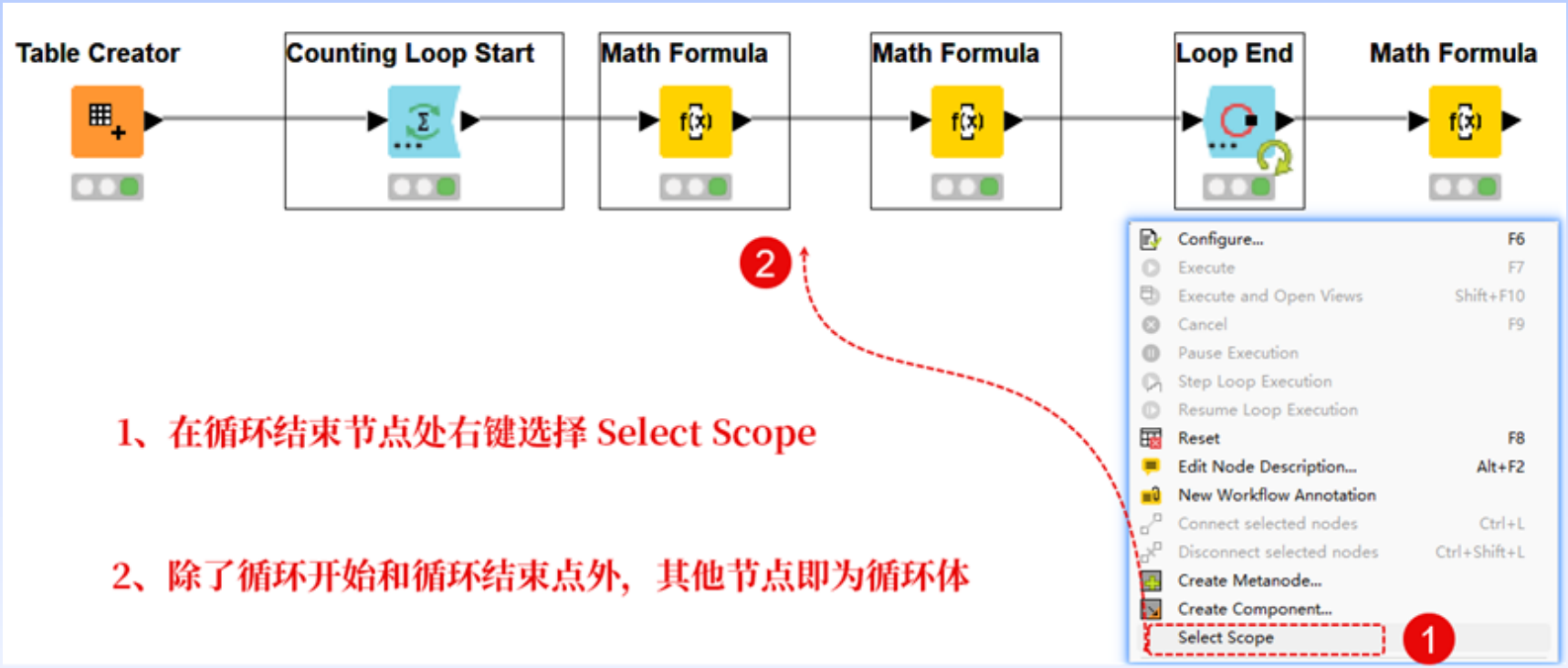
循环体

循环体是循环结构中的核心部分，包含了重复执行的操作或指令。这部分代码决定了循环的主要功能和目的，通常根据具体的业务逻辑编写。

循环结束

为了保持循环的终止，需要在每次循环中应修改循环中的流变量。这可以通过简单的加法、乘法或其他运算逻辑判断来实现。

查看循环体


Table Creator
Counting Loop Start
Math Formula
Math Formula
Loop End
Math Formula
2
Configure... F6
Execute F7
Execute and Open Views Shift+F10
Cancel F9
Pause Execution
Step Loop Execution
Resume Loop Execution
Reset F8
Edit Node Description... Alt+F2
New Workflow Annotation
Connect selected nodes Ctrl+L
Disconnect selected nodes Ctrl+Shift+L
Create Metanode...
Create Component...
Select Scope
1
1、在循环结束节点处右键选择 Select Scope
2、除了循环开始和循环结束点外，其他节点即为循环体

循环分类

1	计数循环	Counting Loop Start 	Loop End  Loop End (Column Append) 	基于指定次数
2	区间段循环	Interval Loop Start 		
3	分块循环	Chunk Loop Start 	Loop End  Variable Loop End 	基于不同的条件 对数据集分组
4	列表循环	Table Row to Variable Loop Start 		
5	分组循环	Group Loop Start 		
6	列循环	Column List Loop Start 	Loop End (Column Append) 	基于选定的各列
7	条件循环	Generic Loop Start 	Variable Condition Loop End 	固定搭配 基于条件判断
8	递归循环	Recursive Loop Start 	Recursive Loop End 	固定搭配 当前循环依懒上一次循环结果



02

计数循环

Counting Loop Start

计数循环基本概念

计数循环是编程中常用的一种循环结构，通过循环次数的递增或递减来控制代码块的执行。编程中通常使用for循环来实现，可以简洁地遍历数组、列表等数据结构。



初始化计数循环变量

在计数循环开始时，需要初始化循环控制变量，KNIME中自动生成关于循环次数的流变量CurrentIteration。

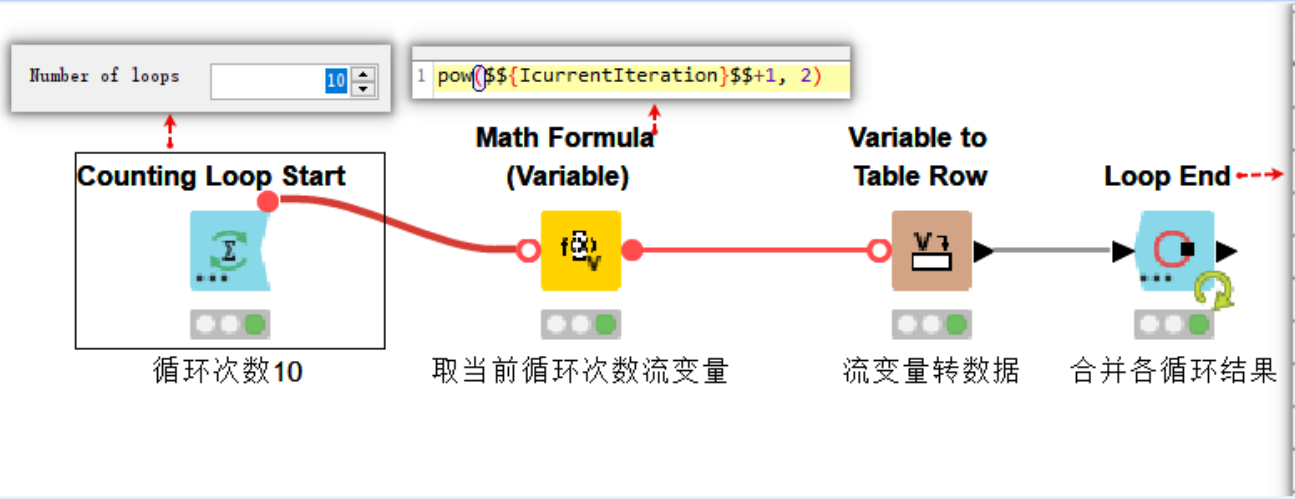
计数循环条件判断

计数循环通过条件判断语句来确定是否继续执行循环体，根据条件的不同，循环体可以被执行零次、一次或多次。KNIME中循环条件已默认设置为达到最大循环次数为结束条件，当然，也可根据情况另外增加循环结束条件。

计数循环结束方法

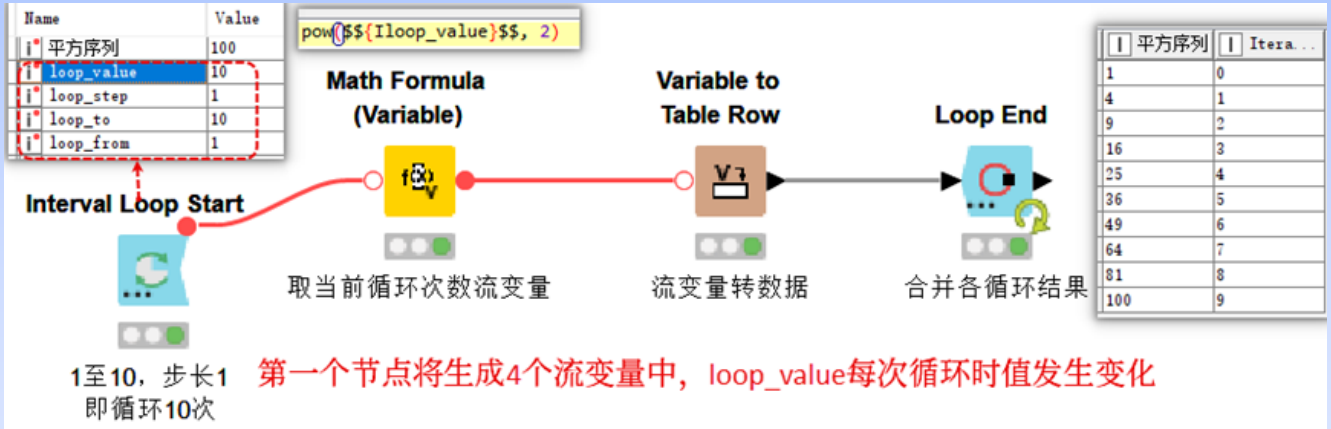
计数循环可以通过设定一个条件或达到特定的循环次数来结束，KNIME中无需设置，达到最大次数即结束循环。

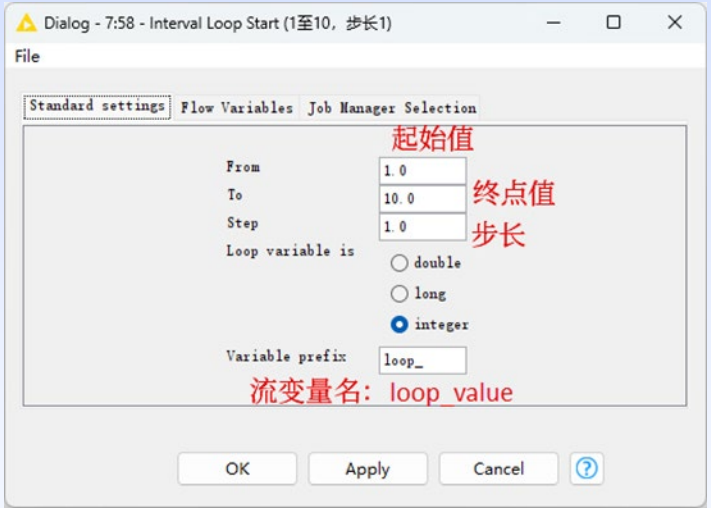
Counting Loop Start



平方序列	Iteration
1	0
4	1
9	2
16	3
25	4
36	5
49	6
64	7
81	8
100	9

Interval Loop Start







03

基于行分组循环

Chunk Loop Start

Chunk Loop Start概述

Chunk Loop Start是一种基于**连续数据行分组**的循环，用于逐组处理数据。它通过循环访问数据集中的每行或每组数据，执行特定的操作或逻辑，适用于对表格数据进行逐行加工的场景。

Chunk Loop Start应用场景

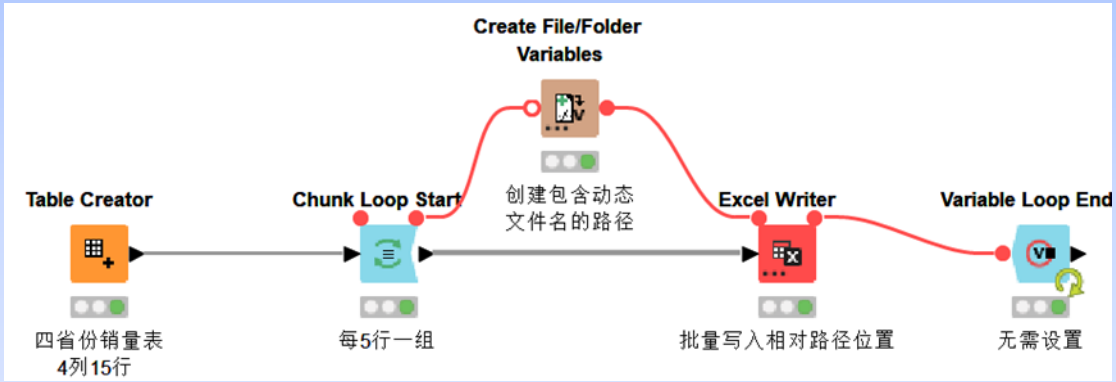
Chunk Loop Start广泛应用于数据处理、数据分析和机器学习等领域。它可以用于数据清洗、转换、聚合等操作，提升数据处理的效率和准确性，简化复杂的数据处理任务。

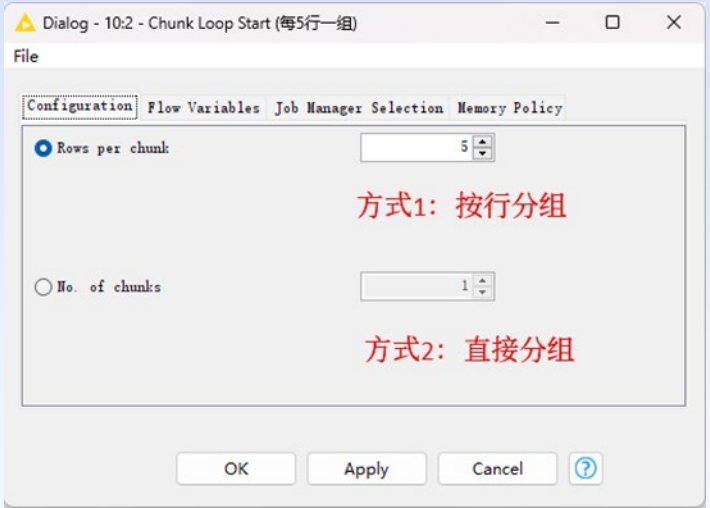
Chunk Loop Start节点配置

配置Chunk Loop Start节点时，需要定义输入和输出端口，并设置合适的循环参数。输入端口连接数据源，输出端口用于输出处理后的数据。根据数据量和处理需求设置循环次数和分组大小。

Chunk Loop Start

示例：分块切分数据并写入各文件中





方式2：基于分组数，自动计算每组应为多少行数据

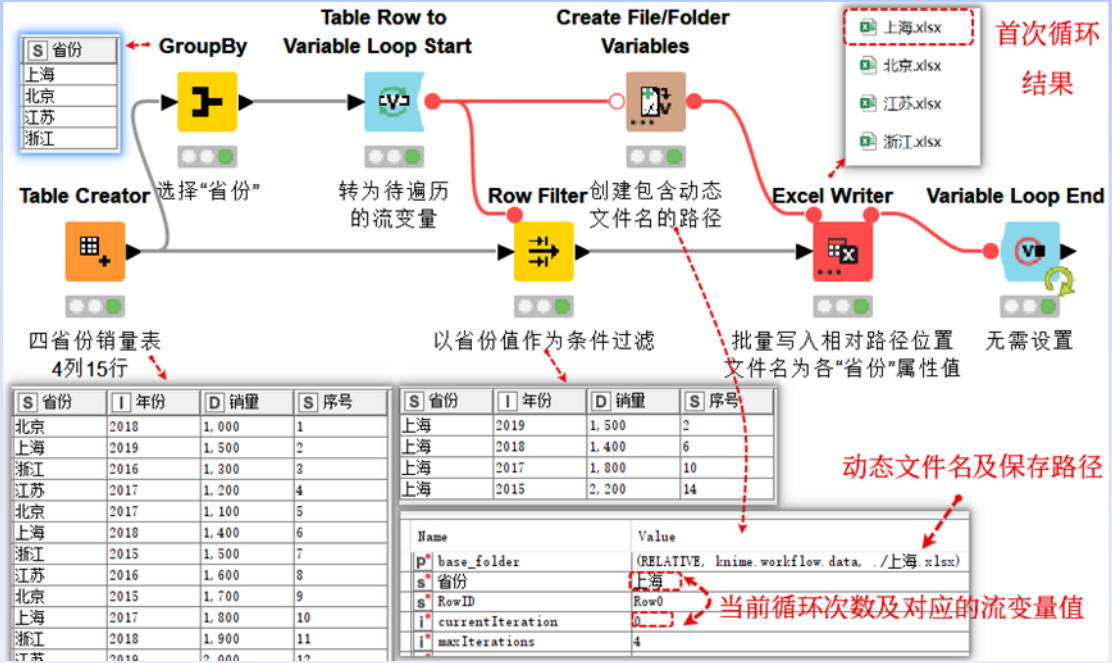
Table Row to Variable Loop Start

行到变量循环

“Table Row to Variable Loop Start”是一种特殊的循环，它将数据表中的指定列中的每一行转换为一个流变量。该概念更类似于Python中的列表循环。

实际应用示例

在KNIME工具中，“Table Row to Variable Loop Start”常用于按省份对原始数据进行分组和过滤。通过这个循环，可以计算每个分组的数据，并写入到单个文件中。



Group Loop Start

Group Loop Start概述

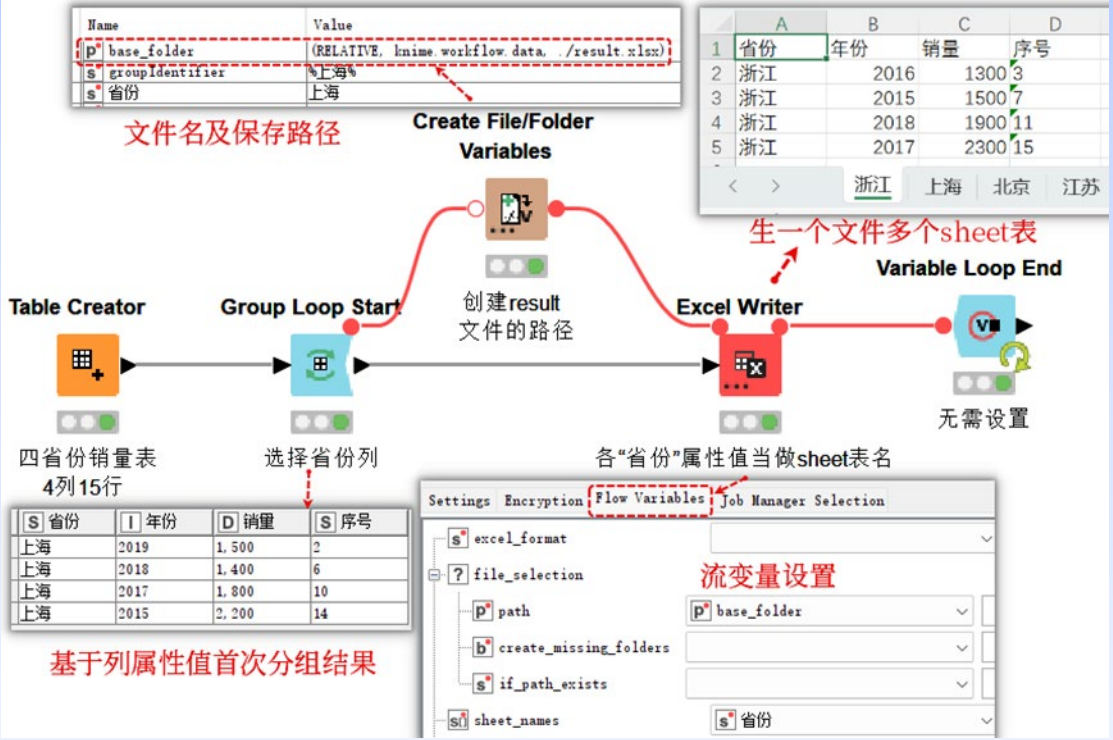
Group Loop Start是一种用于基于列分组的循环，通过将数据按照特定的列进行分组，并在循环体中逐一处理每一组数据。这种循环结构在数据处理和批量操作中非常实用。

Group Loop Start应用场景

Group Loop Start广泛应用于需要对数据集进行批量处理的场景，如数据分析、机器学习训练和文件处理。它可以显著提高数据处理的效率。

注意：

该节点与Table Row to Variable Loop Start基本上可以互换使用，注意用法的细微区别





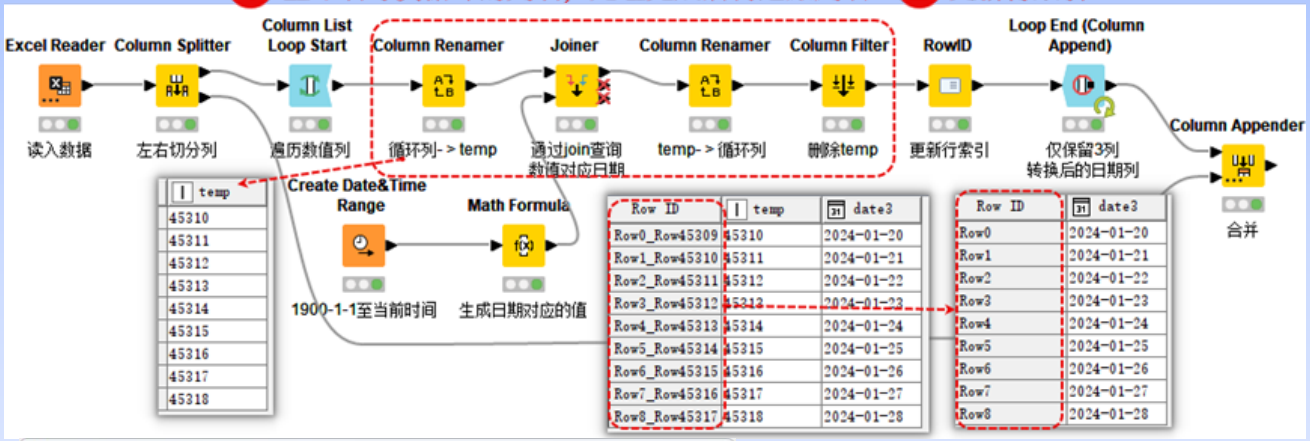
04

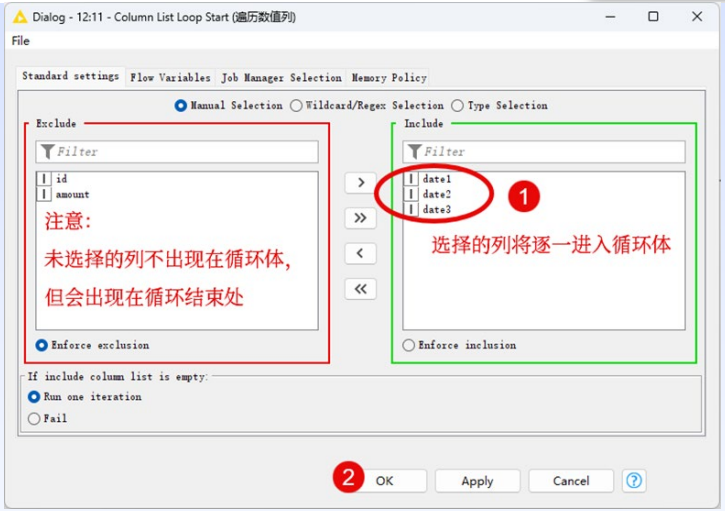
基于列分组循环

Column List Loop Start

Column List Loop Start示例

1 重命名每次循环的列名，处理完成后再还原列名　2 更新行索引





注意：

1、Column List Loop Start节点相比前面的循环需要一定的技巧，其难点在于循环体的设计和数据处理。

2、该节点的分组是指每一列作为一个循环的对象，基于循环四要素的循环条件也无需设置，取值到选定列范围的最后一列即结束循环。

3、此外该节点的循环结束节点通常搭配Loop End （Column Append），即循环结束时收集数据，将左右数据表合，并在一起。



05

条件循环

Generic Loop Start

通用循环定义

通用循环（Generic Loop Start）是一种在数据处理过程中反复执行特定任务的节点，直至满足特定条件或达到指定次数。类似于编程语言中的While循环，它可以处理复杂的数据流程和条件判断。

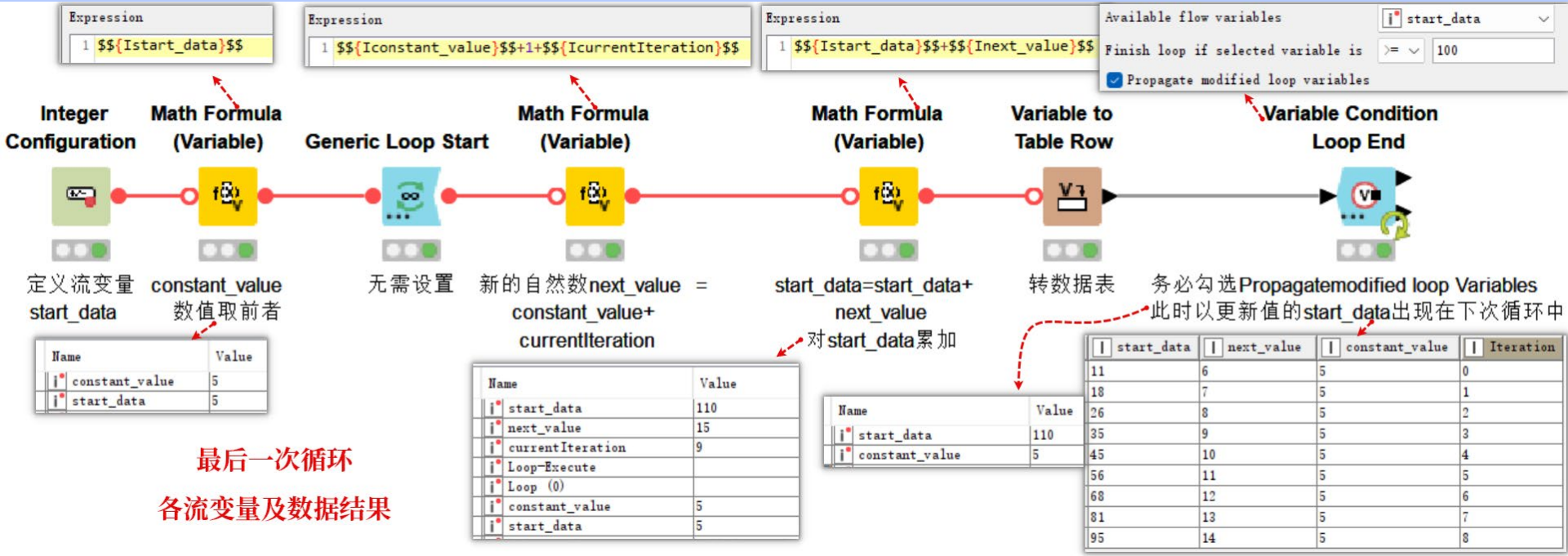
通用循环结构

通用循环通常由开始节点（Generic Loop Start）、和结束节点（Variable Condition Loop End）组成。开始节点启动循环，结束节点根据设定的条件决定是否继续循环，达到设置的终止条件，则结束节点终止循环。

通用循环注意事项

在使用通用循环时，需要确保循环条件的设置合理且准确，以避免无限循环或过早结束。同时，应合理安排循环体内的任务执行顺序，以优化整体性能和资源利用。

Generic Loop Start



最后一次循环

各流变量及数据结果

示例：条件循中的反向传递流变量示例

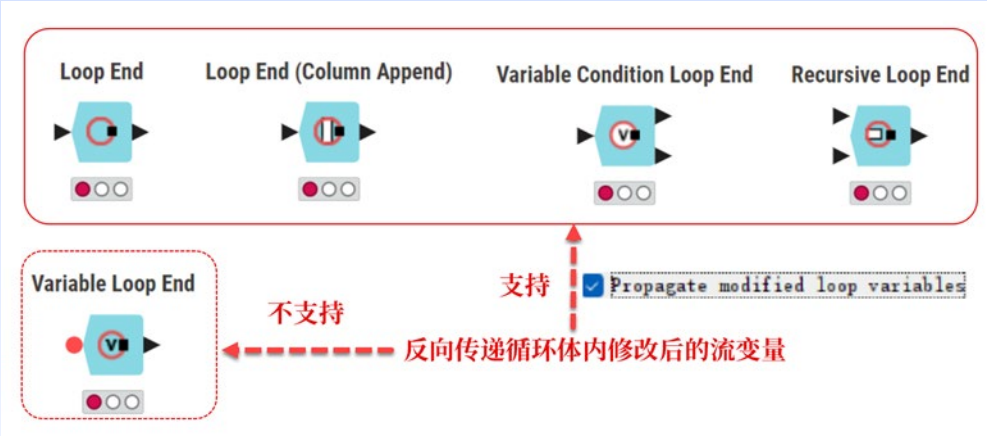
基于反向传递流变量控制循环

反向传递流变量定义

反向传递流变量是一种在循环中使用的特殊变量，用于在每次迭代中传递前一次的结果。这种变量可以在循环的每次迭代中更新和存储重要信息。

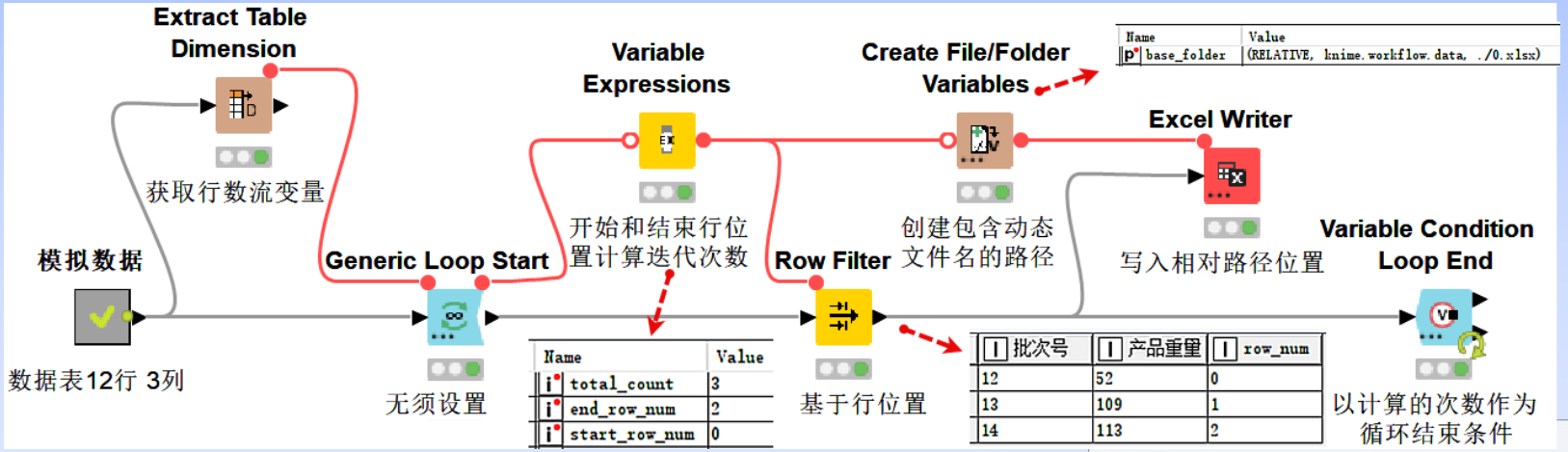
控制循环流程结构

基于反向传递流变量的控制循环主要通过设置初始值、更新步骤和结束条件来控制循环流程。初始值用于第一次迭代，更新步骤确保每次迭代都使用最新数据，而结束条件则决定何时停止循环。

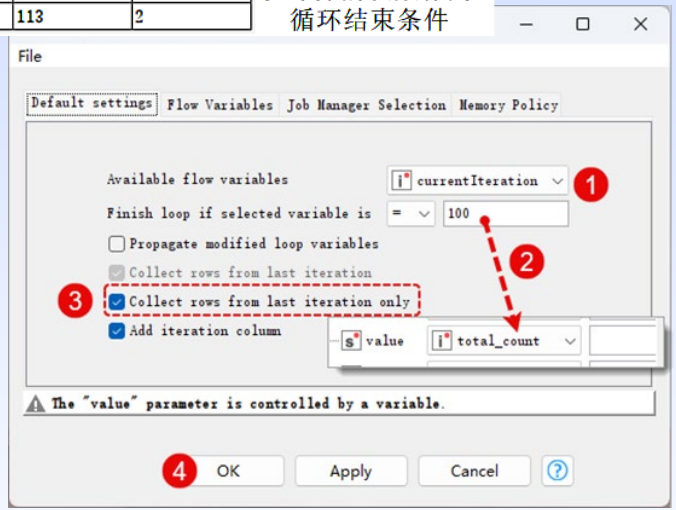


支持反向传播循环结束节点

其他控制循环情形



使用Generic Loop Start实现类似Chunk Loop Start的操作。比如说希望每3行为一组，导出为每个文件。此时，使用当前迭代次数的流变量，再乘以行数，分别构造一个开始行和结束行





06

递归循环

Recursive Loop Start

递归循环定义

递归循环是一种通过函数或方法自我调用来实现循环的技术。KNIME 中若本次循环需要用到上一次循环结果，那么就可以考虑使用递归循环。

--递归循环是各循环中的难点。

基于行过滤递归循环

基于行过滤的递归循环通过筛选数据中的特定行来实现递归。这种方法常用于处理表格数据，可以根据指定条件递归访问每一行或若干行为一组的数据，适用于数据分析和处理。（详见本书示例）

递归循环应用实例

递归循环在许多领域有广泛应用，例如计算阶乘、树遍历、文件管理系统等。通过将问题分解为子问题并递归处理，可以简化复杂问题的求解过程，提高程序的效率和可维护性。

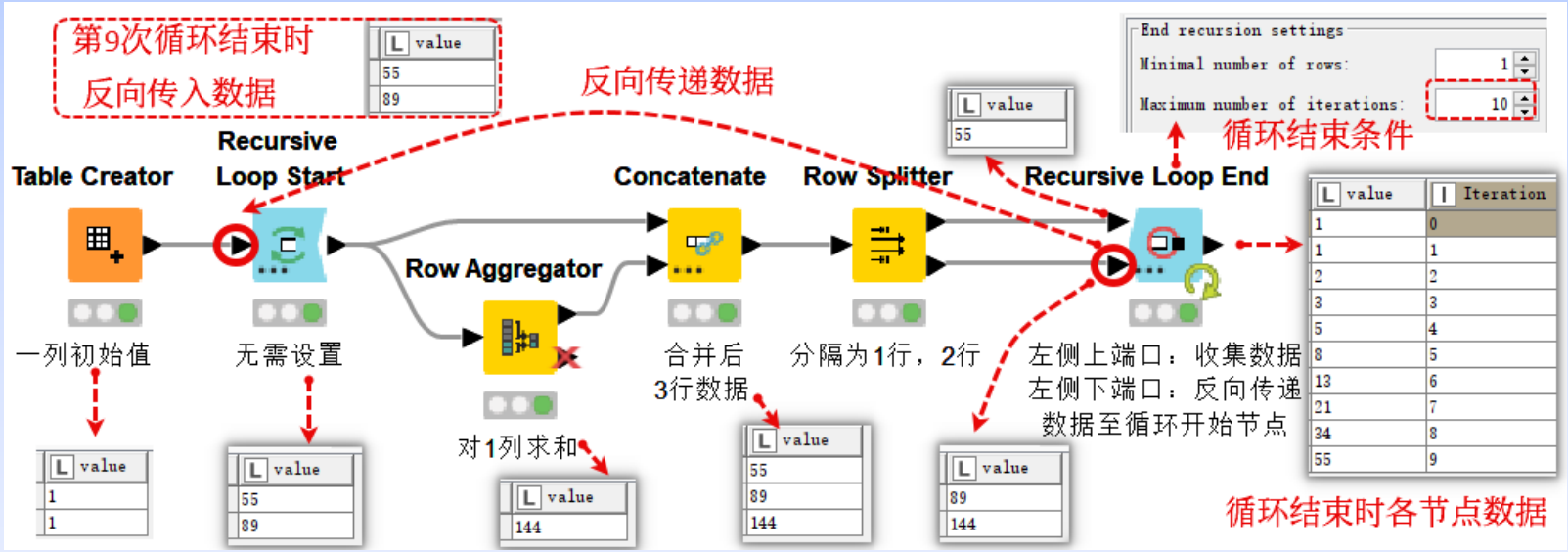
基于列过滤递归循环

基于列过滤的递归循环通过逐列筛选数据来实现递归。它根据指定条件递归访问每一列的数据，常用于处理表格数据中需要按列进行操作的场景，如数据汇总和分析。（详见本书示例）

递归循环优势与劣势

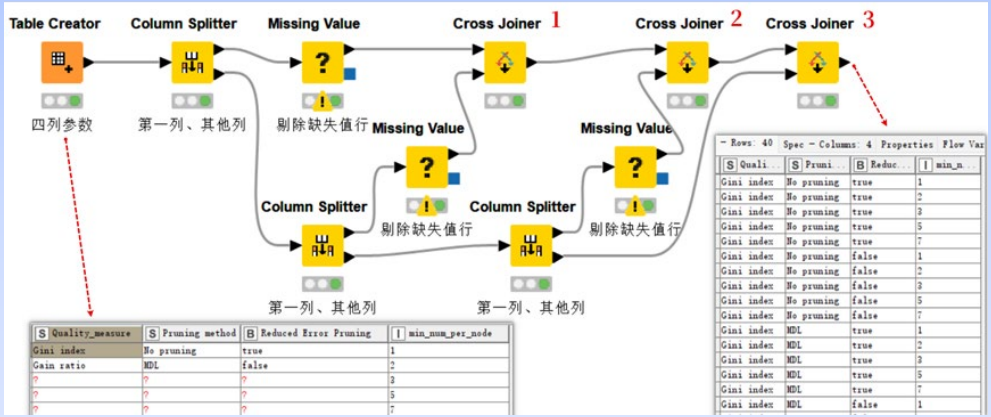
递归循环的优势包括工作流更简洁、易于维护，特别适用于处理可以明确分解为子问题的场合。然而，递归可能需要基于逻辑设计工作流，是KNIME循环中的难点，对用户要求较高。

基于行过滤递归循环



递归生成斐波那契数列

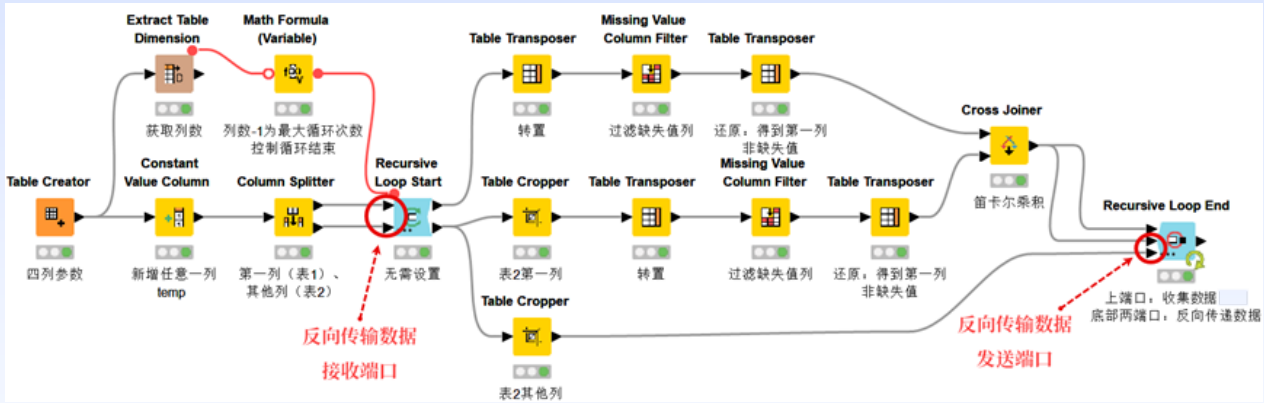
基于列过滤递归循环



无论是4列还是20列，递归循环都能处理，而左侧工作流则需要19次使用Cross Joiner节点手动实现组合，显然不方便。

求四列属性值的所有组合（不使用递归循环）

求四列属性值的所有组合（使用递归循环）



递归循环小结

递归与循环区别

KNIME中，递归循环中本次循环需要用到上一次循环结果，而其他类型的循环则本次循环与上一次循环没有多大关系，更不需要用到上一次循环结果。前置可以处理一些复杂的问题，甚至是其他循环不能替代的。

递归函数定义与应用

递归函数根据基本情况和递归步骤进行定义。在基本情况下，直接计算输入结果；在递归步骤中，通过多次调用自身来解决复杂问题，这种方法常用于解决分治问题。

递归优缺点分析

递归的优势在于能够将复杂问题层层转化为规模较小的子问题，提高问题解决的效率。然而，递归函数存在无限递归的风险，易 出现死循环情况，需谨慎使用。

01 02 03 04 05

递归策略应用场景

递归策略常用于解决分治问题，如排序、搜索、图形处理等领域。它能够有效地将复杂问题分解为多个规模较小的子问题，简化问题的解决过程。

递归与循环在实际编程中选择

在实际编程中，递归和循环各有适用场景。递归适用于需要明确分层解决问题的场景，而循环则适用于频繁重复相同操作的场合。选择合适的控制结构可以提高代码的效率和可读性。



07

多个循环间关系

串联与并联循环

串联循环定义

串联循环是指多个循环步骤依次执行，每个步骤的输出作为下一个步骤的输入，直到最后一个步骤完成。这种模式确保了数据和结果的连续性，便于整体分析和处理。

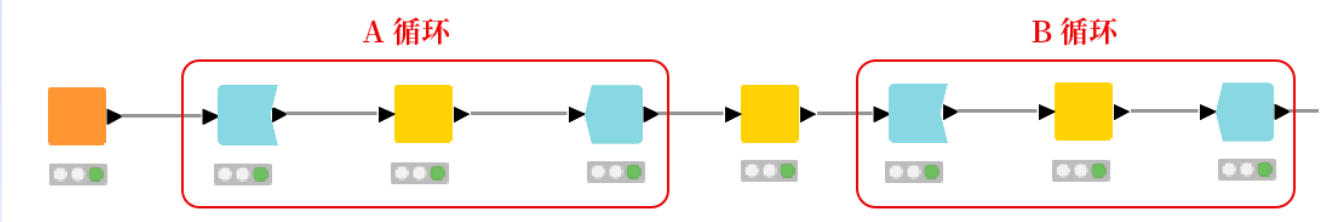
并联循环定义

并联循环指的是多个循环同时独立运行，每个循环的输出相互独立，互不影响。这种模式适用于需要高效并行处理的场景，提高了整体的运算能力。

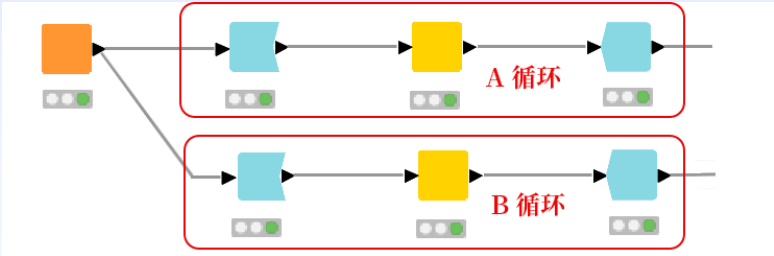
串联与并联循环区别

串联循环强调步骤间的连续性，适合处理有明确先后顺序的数据；而并联循环则强调各步骤的独立性，适用于需要多任务并行处理的场景。两者在应用上各有优势。

串联



并联



嵌套循环

嵌套循环定义

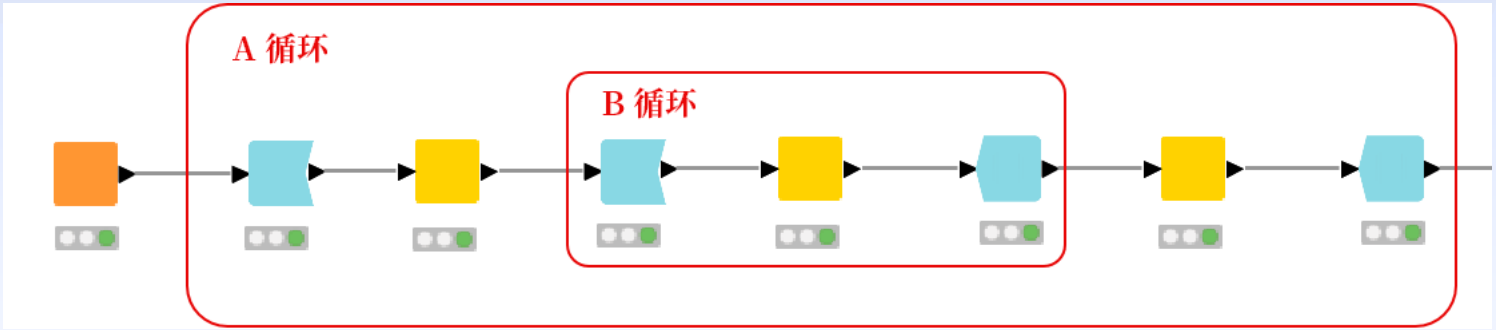
嵌套循环指的是在一个循环内部再包含另一个或多个循环，形成层次化的控制结构。这种循环可以用于实现复杂的数据操作和处理，提升代码的灵活性与功能性。

循环嵌套优势

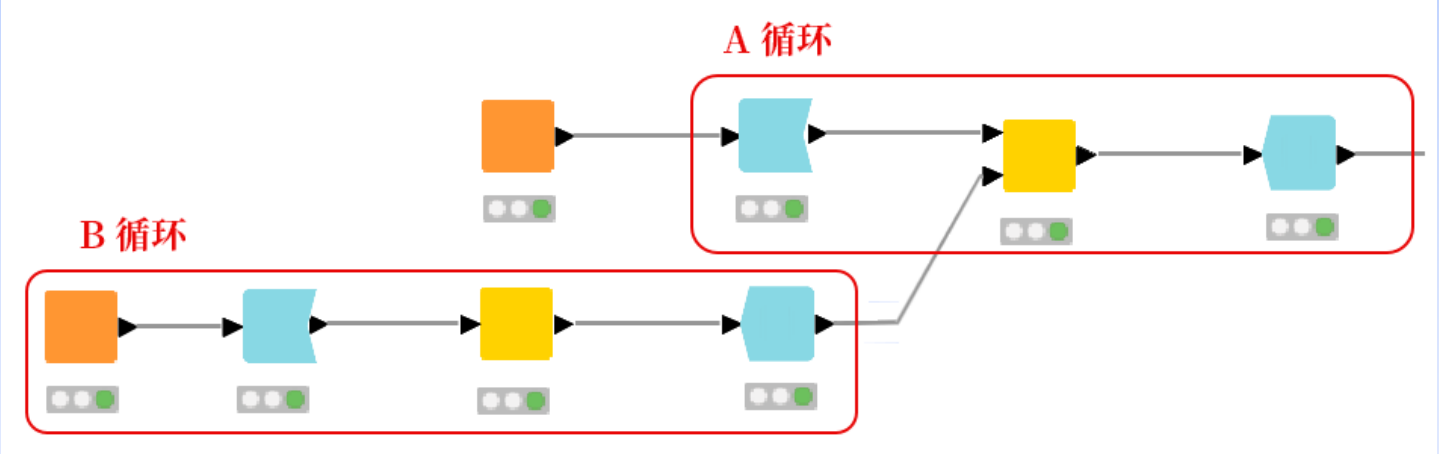
嵌套循环可以有效地处理多维数据和复杂模式，通过逐层遍历数据，能够实现更加精细的控制。它增强了程序的功能性和可维护性，特别是在需要对数据进行复杂操作时表现尤为出色。

常见嵌套循环示例

常见的嵌套循环示例包括打印九九乘法表等。这些示例展示了如何通过循环嵌套来简化计算过程并提高代码的可读性和效率。

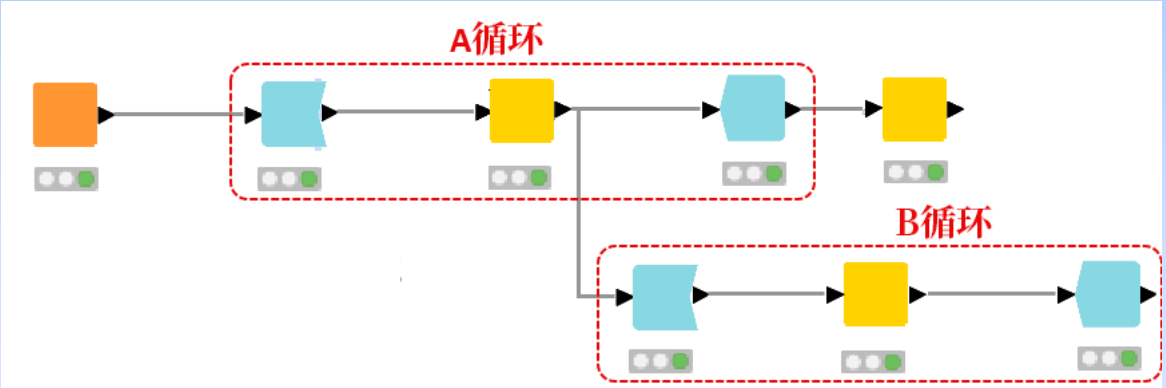


分支型循环



分支型循环定义

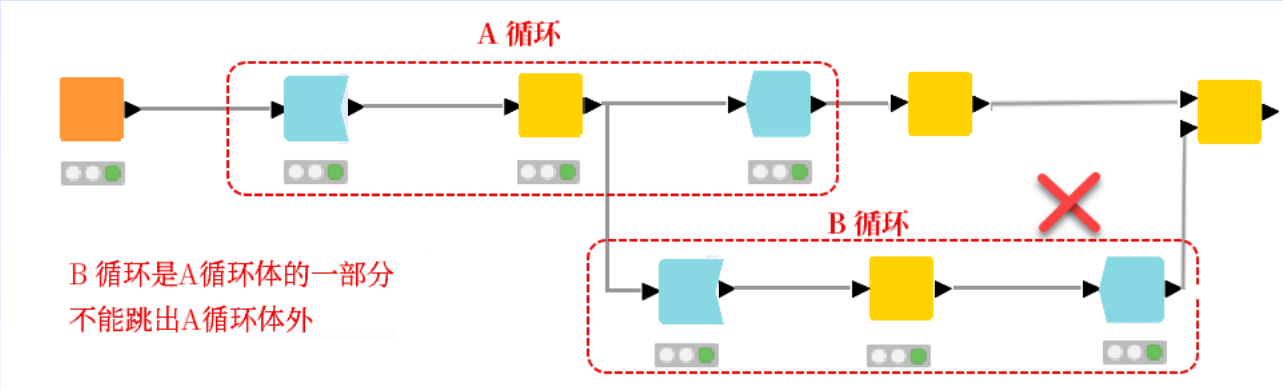
非典型嵌套循环



小技巧：

记住“兀”字，就能区分

分支型循环和非典型嵌套循环





08

循环流程控制

循环流程控制

01 立即跳出循环

在某些情况下，需要立即结束循环，而不是等待循环体完成，如Breakpoint节点。

02 等待

通过设置等待时间或等待条件，触发循环的执行或终止等，如Wait节点。

03 条件判断与分支

在循环过程中进行条件判断和分支是常见的编程策略。通过Switch类节点可以在每次迭代中检查条件，根据不同的结果执行不同的操作，从而实现更复杂的逻辑处理。

04 异常处理与错误恢复

在循环过程中，可能会遇到各种错误和异常情况。使用Try、Catch Errors节点可以在发生错误时捕获异常，并采取相应的措施来保证程序的正常运行，避免因错误而导致的程序崩溃。

多循环结束条件

串联与并联循环

在编程中，串联和并联是两种常见的循环结构。串联循环按顺序依次执行每个循环体，而并联循环则同时执行多个循环体。理解它们的特性有助于优化代码逻辑。



嵌套循环

嵌套循环是指在一个循环内部再包含另一个循环。这种结构通常用于实现更复杂的控制流，如多层筛选或分步操作。嵌套循环需要特别注意循环的结束条件，以避免无限循环。

分支型循环

分支型循环根据不同条件选择不同的执行路径。这种循环结构常用于根据不同情况执行不同任务，提高代码的灵活性和可维护性。

非典型嵌套循环

非典型嵌套循环指的是那些不符合传统嵌套模式的循环结构。其本质仍为嵌套循环



09

常见循环问题

常见循环错误



循环条件错误

检查循环条件是否设置正确，确保循环能够正常结束。如果循环条件永远为真，循环将无法结束，导致程序逻辑错误。正确的循环条件是保证循环正常退出的基础。

循环体内部逻辑错误

检查循环体内部的逻辑是否正确。可能由于错误的条件判断或操作导致循环无法正常结束。确保循环体内部的每一步骤都能正确执行，避免因逻辑错误导致的异常。

提前退出循环

在循环体中加入提前退出循环的条件，如满足特定条件时使用Breakpoint节点。这样可以在需要时及时终止循环，避免无效的迭代和资源浪费。

数据结构或数据类型不一致

在循环结束节点针对此类问题有专用的选项，通过勾选解决此类不一致问题。

循环体内错误导致无法保存数据

循环体内错误类型

循环体内可能出现多种错误类型，包括逻辑错误、语法错误和流变量赋值错误。这些错误可能导致程序提前返回或停止执行，影响数据的正确处理和保存。

错误导致无法保存数据原因

在循环体中的错误可能由于错误的循环条件、不正确的流变量更新或对数据的无效操作引起。这些错误会直接影响数据的保存和处理，因此可以在循环体内逐次追加写入文件。

避免循环体内错误的策略

为避免循环体内错误导致无法保存数据，应确保循环条件正确无误，变量赋值要精确，并减少不必要的IO操作。通过优化循环内部的代码逻辑，可以提高数据保存的准确性。

循环结构混乱

循环条件错误

循环结构混乱的一个常见原因是循环条件设置不正确。如果条件判断不准确，程序将无法正常进入或退出循环，导致逻辑错误和程序崩溃。

变量初始化与更新问题

在循环结构中，未对变量进行正确初始化或更新也会导致混乱。例如，未定义初始值的变量可能在循环中被误用，造成不可预知的行为。



控制流程错误

控制流程错误是引起循环结构混乱的另一个因素。这会使得程序的控制流程变得复杂且难以追踪。

嵌套循环层级过深

过度使用嵌套循环会导致逻辑层次过深，增加理解和维护的难度。缺乏清晰的注释和规范的命名会使代码更加混乱，难以阅读和调试。



10

带有循环功能其他节点

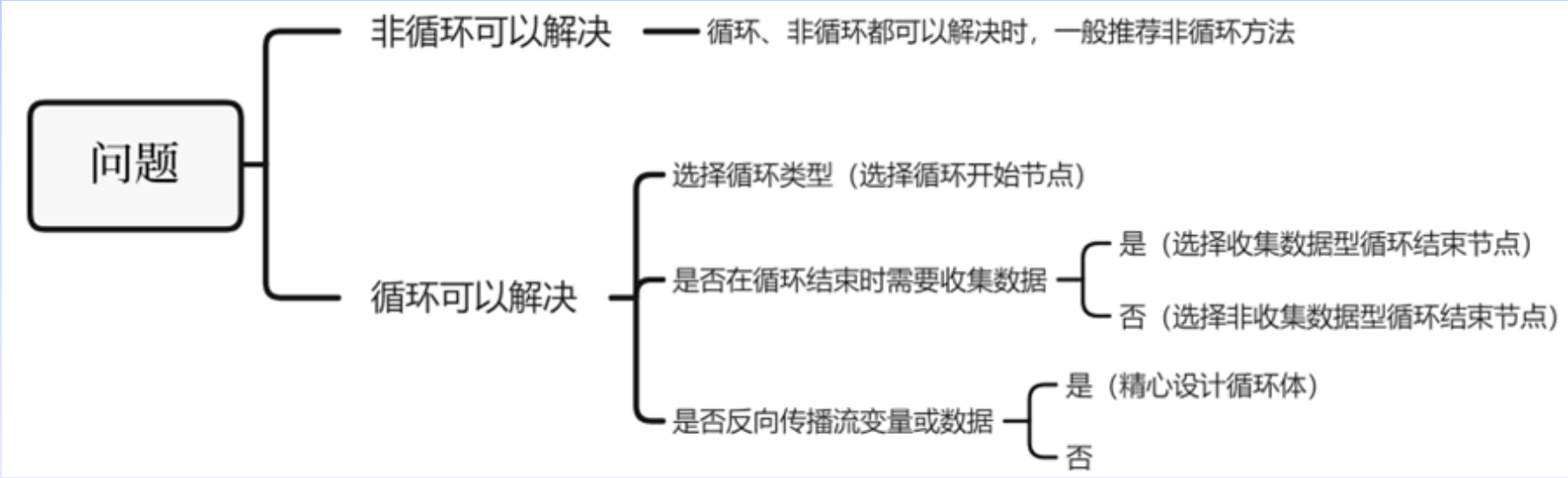
常用避开循环节点

当循环次数过多时，可能造成执行时间长问题。有些情况下，也可以借助一些技巧，转换思路，避开循环。

此外，也有些节点虽然名称上看不出与循环有任何关系，但其实暗含循环功能，或通过不同节点的配合操作，实现与循环一样的效果。

常用节点：详见10.10章节，共计13个节点。

选择循环与否流程图



1、一般来说，利用循环可以简化思考，更灵活的处理数据，但有时可能存在效率问题。



11

本章小结

回顾关键知识点

01 循环四要素定义

02 循环的常见分类

03 多个循环之间的关系

04 常见循环问题及对策

05 控制循环的其他情形

06 避开循环的其他节点

总结常见问题及解决方案

循环次数问题

在设计工作流时，未正确设置循环终止条件或逻辑错误可能导致循环次数异常。解决方案包括检查循环条件和循环计数是否正确，确保在每次迭代后都更新条件或计数，以避免无限循环。

死循环问题

程序可能因逻辑错误、资源竞争或等待特定事件而陷入死循环。解决方法包括检查代码中的逻辑错误，设置最大循环次数，并引入一些终止条件，防止无限执行。



业务处理逻辑问题

循环体内要遵循业务的处理逻辑，此类逻辑错误虽能正常执行，但非预期结果，排查起来更困难。

数据问题

数据的一致性和完整性，要根据数据格式选择适合的方法。

感谢观看

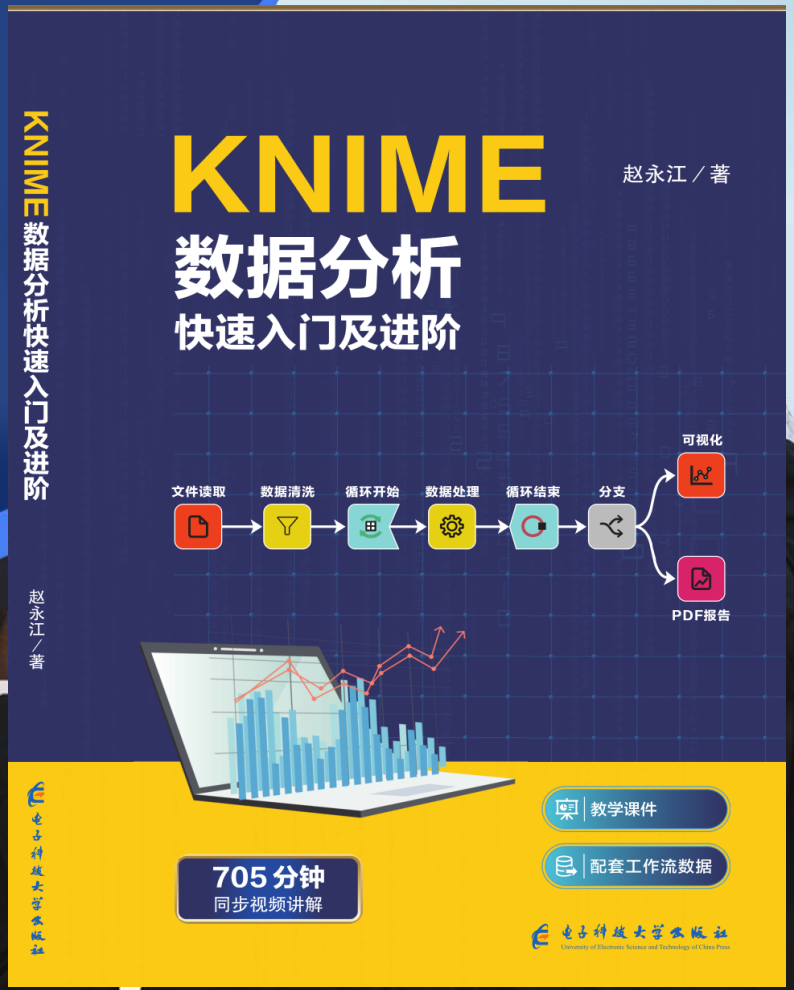
THANKS



第11章

分支与错误处理

掌握分支结构与错误处理技巧



目录

01 分支四要素

02 两分支

03 多分支

04 分支与循环

05 其他扩展分支

06 错误处理Try和Catch Errors

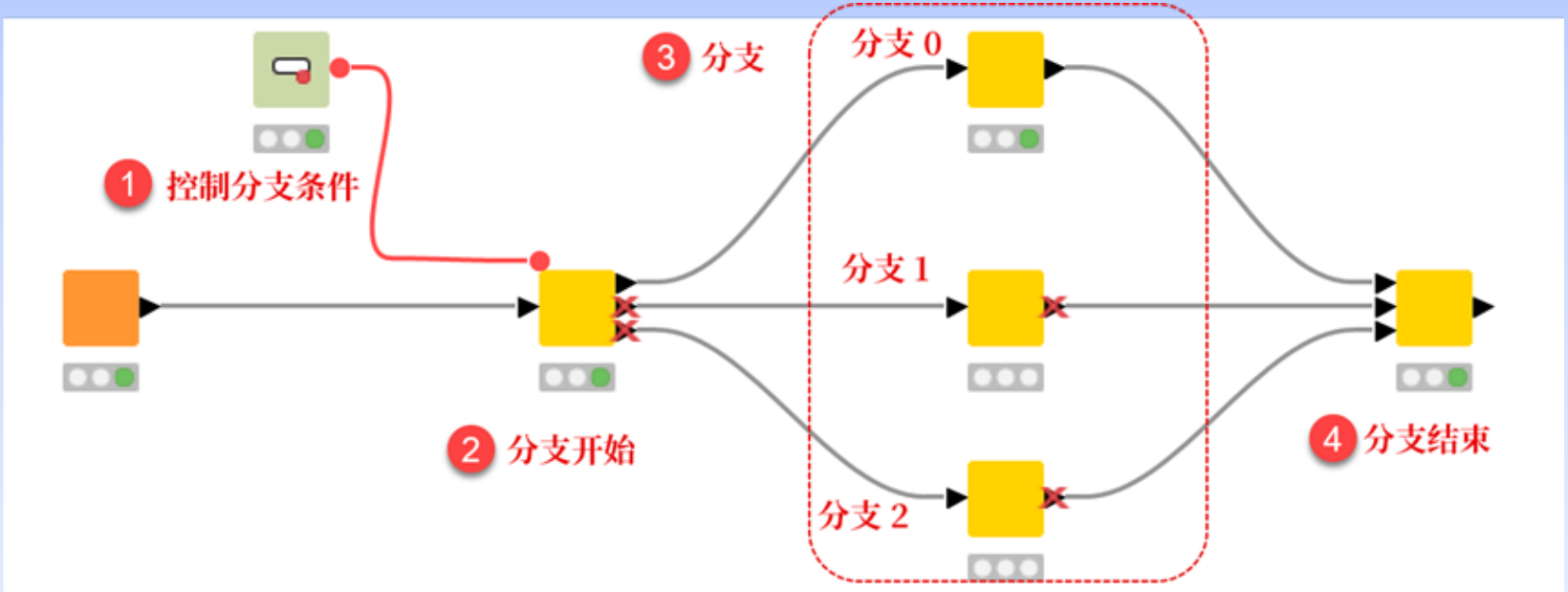
07 本章小结



01

分支四要素

分支四要素



- 分支开始与分支结束节点往往成对配合使用，大部分分支结构遵循分支四要素结构
- Empty Table Switch节点可以不使用分支结束节点

定义与分类

分支定义

分支工作流是KNIME工作流中用于根据条件判断结果选择不同执行路径的工作流。常见的分支节点包括if、switch等类，它们通过条件表达式来决定代码的执行顺序和方向。



条件判断语句

条件判断语句也称为分支语句，主要包括if节点和switch类节点。判断语句根据条件表达式的值决定是否执行特定分支工作流，而switch语句则根据表达式的值激活多个分支工作流中的某一个。

循环执行中的分支工作流

循环执行语句用于控制代码块的反复执行。，它们根据条件表达式的值决定是否继续执行循环内的代码块。

应用场景与实例

IF Switch应用实例

在Java编程中，IF Switch语句常用于根据不同条件执行不同操作。例如，可以根据用户输入的账户类型决定显示不同的操作界面。分支及组件的配合，可以实现更灵活的用户交互体验。

CASE Switch Star应用实例

CASE Switch Star语句适用于需要从多个选项中选择一个的场景。例如，本章中基本不同采购类型，发送不同邮件。

多分支联合应用

当面临复杂业务逻辑时，多分支联合应用能够提供更强的逻辑处理能力。例如，在电商平台的订单处理流程中，结合IF、CASE和循环结构，可以实现自动化的订单分类与状态更新，提高整体运营效率。

Java IF（Table）应用实例

Java IF（Table）节点中可以使用Java语句来处理多分支逻辑。

案例应用：利用分支发送不同邮件

在电子邮件服务系统中，可以通过条件表达式，根据收件人的不同需求选择不同的邮件模板。例如，对于重要客户发送定制邮件，而对于普通用户则使用标准模板，提升个性化服务水平。

注意事项与最佳实践

避免多层分支嵌套

多层分支嵌套是编程新手常犯的错误之一。过多的嵌套会增加复杂度，降低可读性和可维护性。应尽量简化分支层次，使工作流结构清晰。

使用分支或IF条件判断类函数

在if语句中合理使用else关键字可以增强代码的健壮性。当条件不满足时，else分支能够确保程序执行其他必要的代码块，避免流程中断或错误。



注意分支语句语法

在使用switch类节点时，需要确保每个分支的表达式是唯一且互斥的，做到不重复不遗漏，避免匹配失败，确保流程正确跳转。

适当运用多分支结构

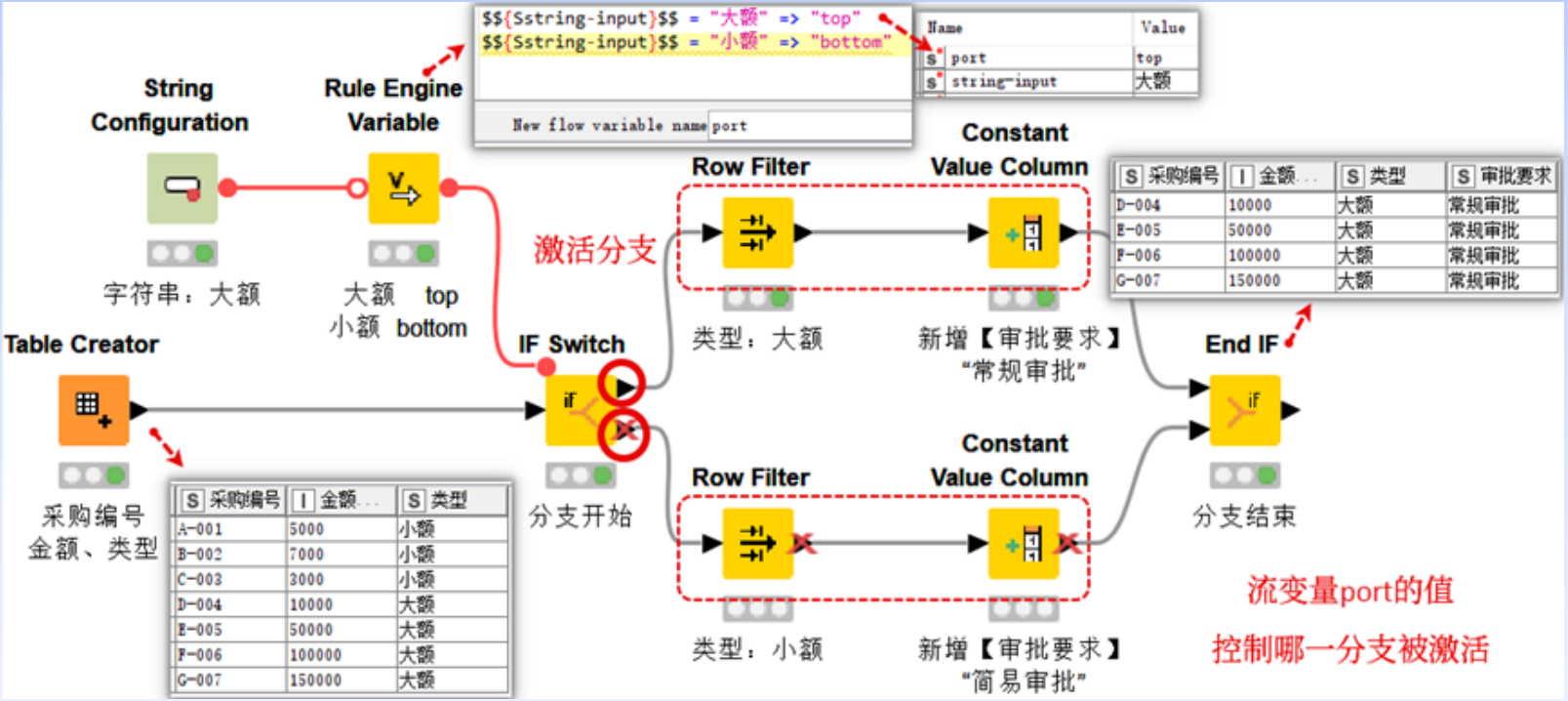
可以处理多种情况，提高代码的灵活性和扩展性。但在使用时需注意避免过度分支，以免增加阅读难度和维护成本。



02

两分支

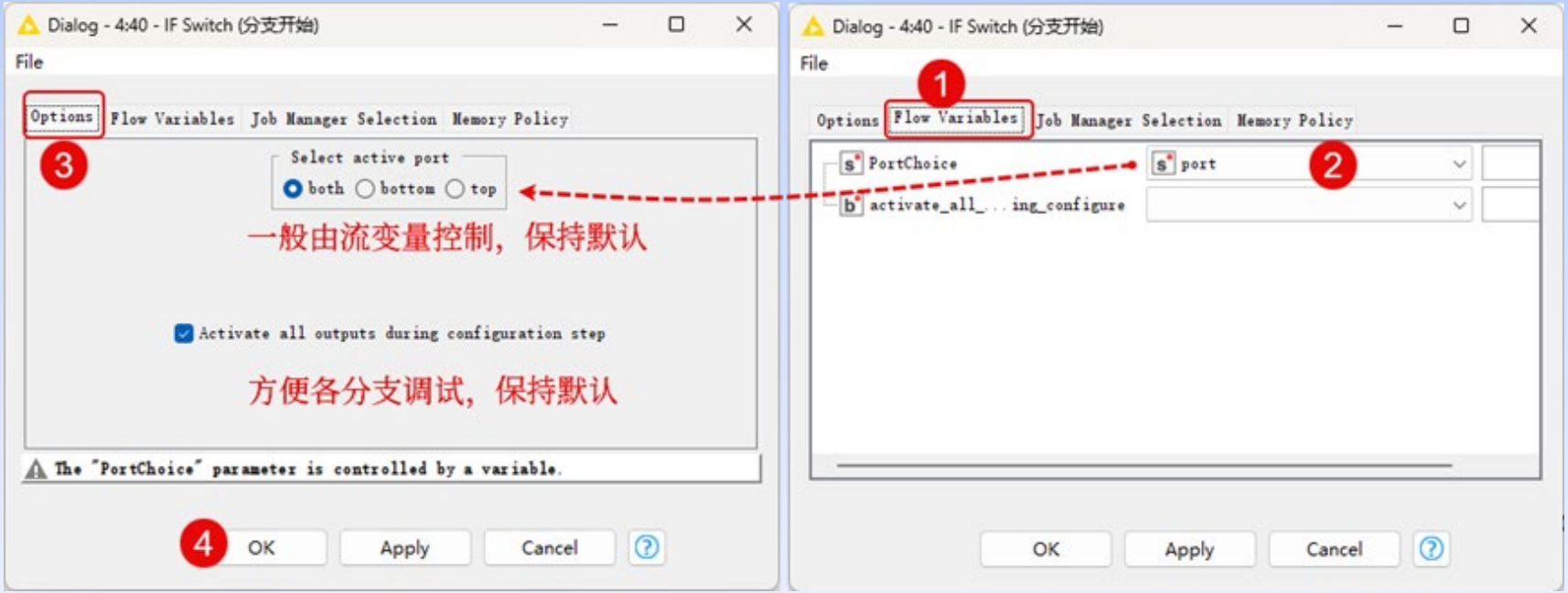
IF Switch



采购编号	金额	类型	审批要求
D-004	10000	大额	常规审批
E-005	50000	大额	常规审批
F-006	100000	大额	常规审批
G-007	150000	大额	常规审批

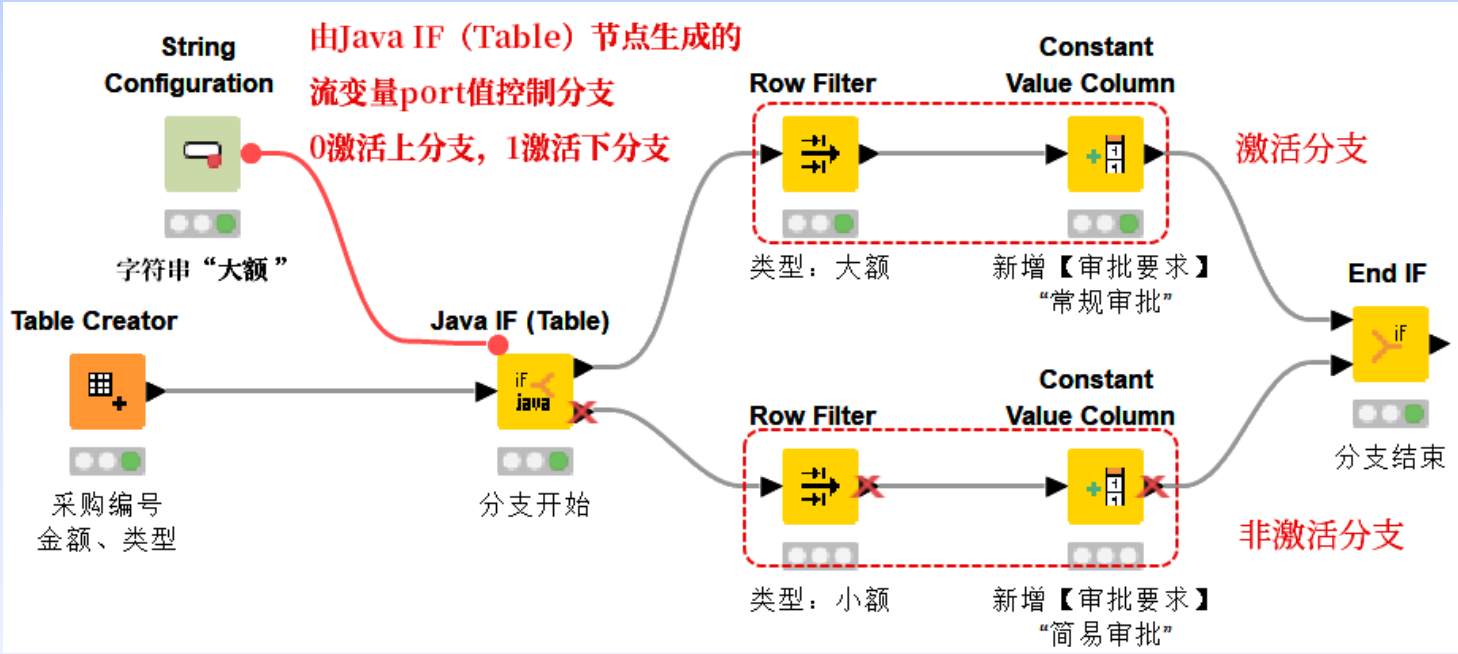
采购编号	金额	类型
A-001	5000	小额
B-002	7000	小额
C-003	3000	小额
D-004	10000	大额
E-005	50000	大额
F-006	100000	大额
G-007	150000	大额

IF Switch设置

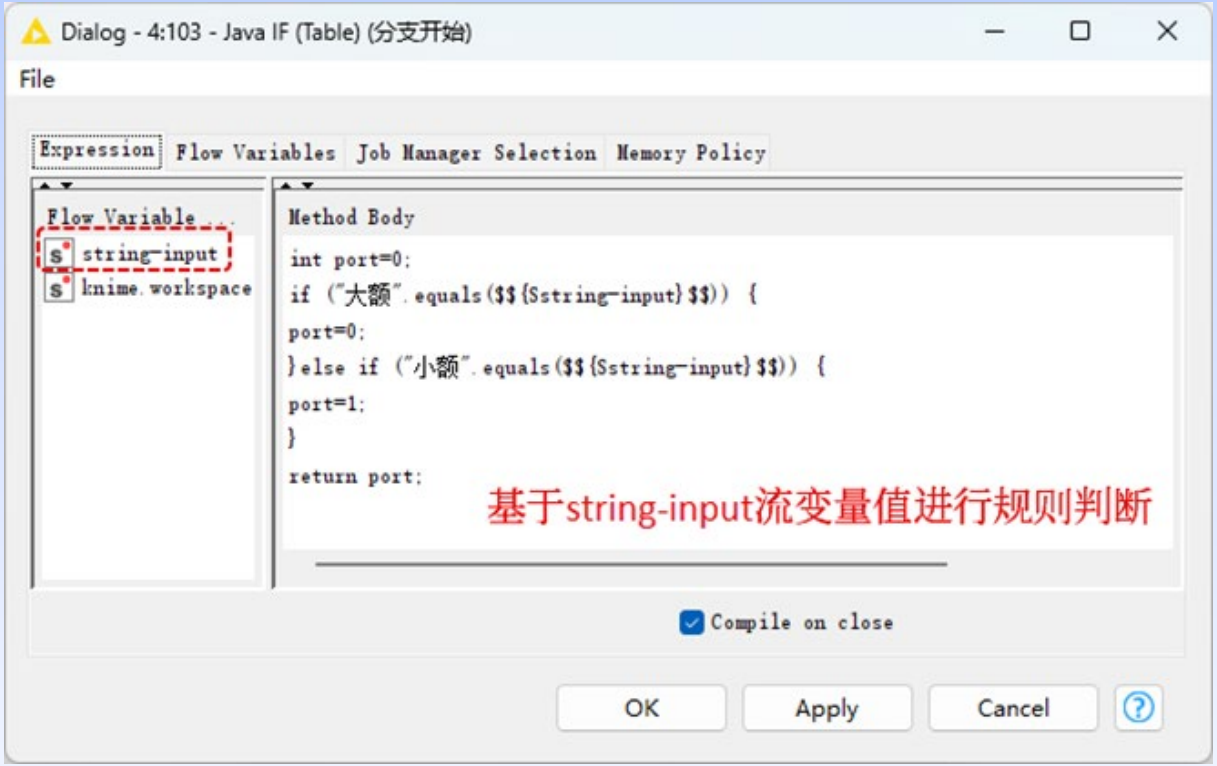


使用流变量控制激活分支

Java IF（Table）



Java IF（Table）设置

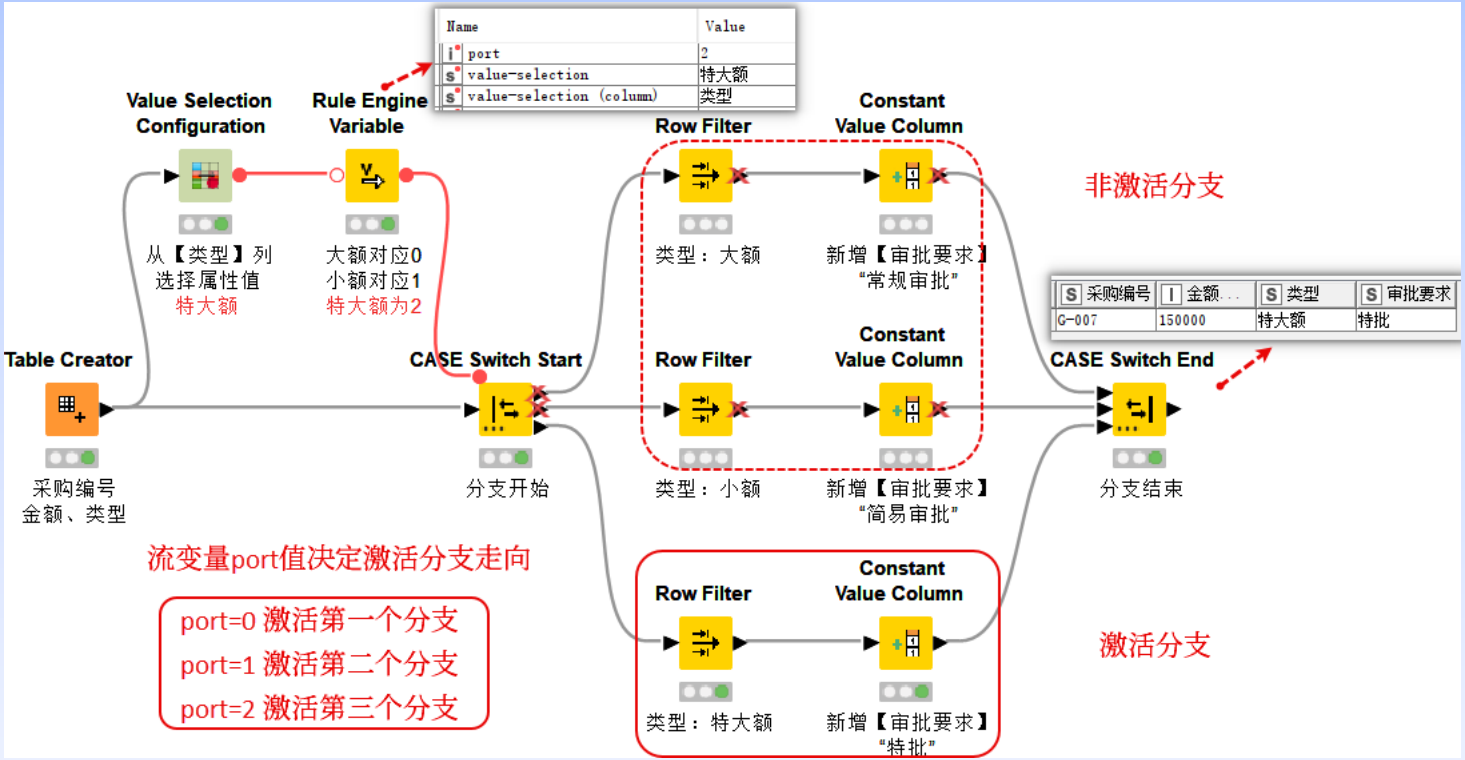




03

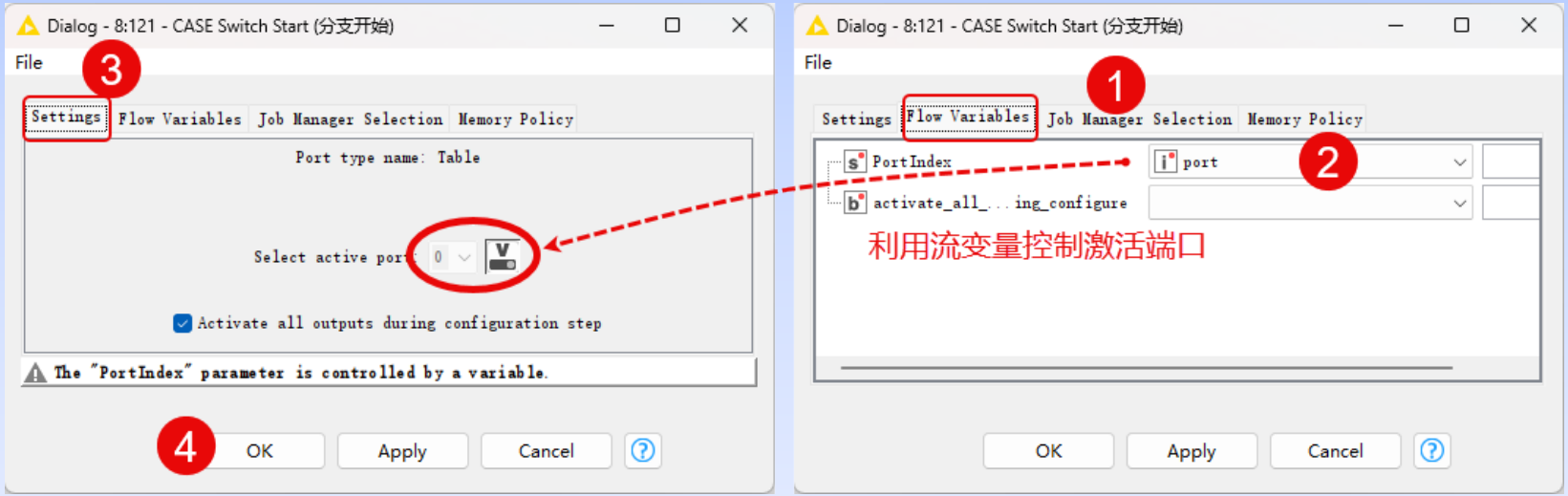
多分支

CASE Switch类



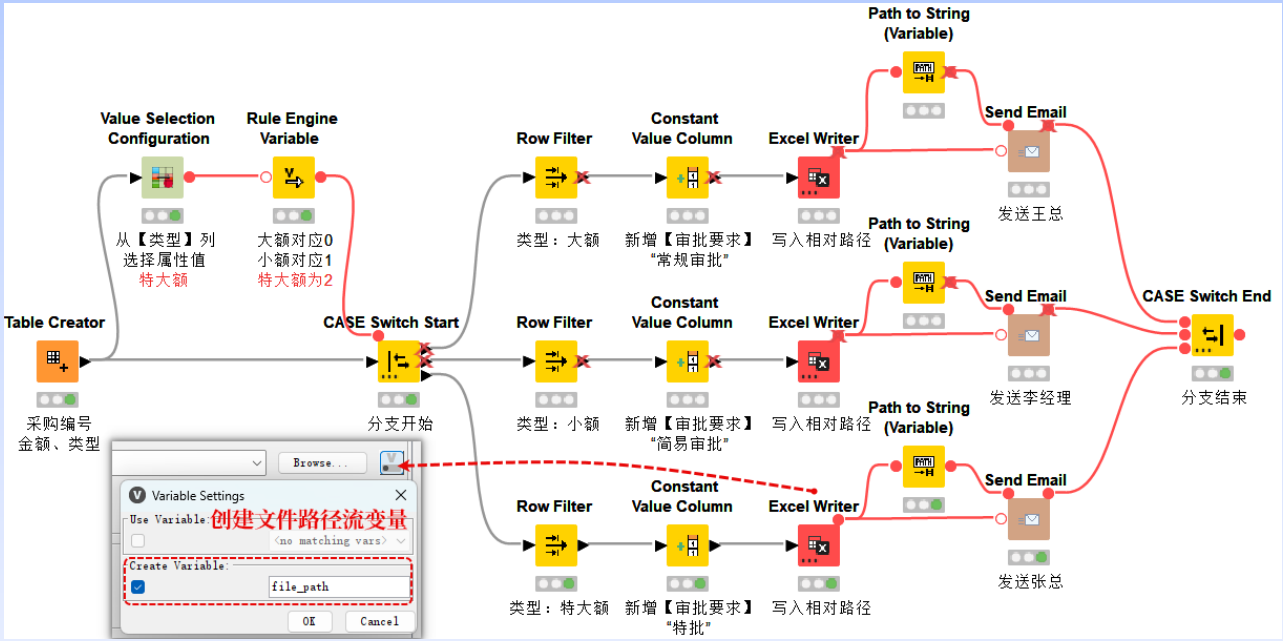
注意：只有IF Switch 节点使用字符串top或bottom代表分支上下，
其他分支开始类节点大多使用数字0，1，2等来代表分支及顺序

CASE Switch类节点设置



注意：只有IF Switch 节点使用字符串top或bottom代表分支上下，
其他分支开始类节点大多使用数字0，1，2等来代表分支及顺序

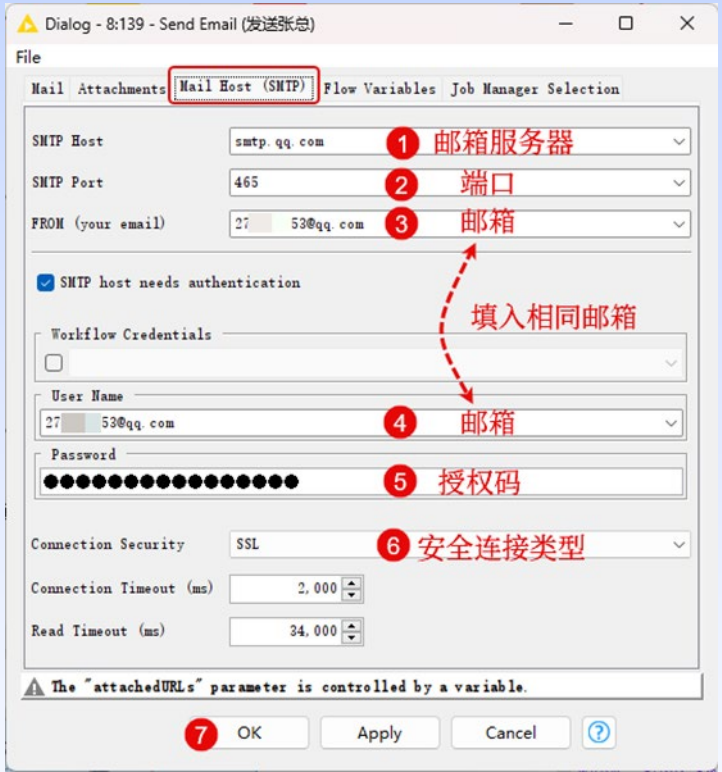
案例应用：利用分支发送不同模版邮件





附：利用QQ邮箱为发件人的相关设置

QQ邮箱页面顶部的“设置”→“账号”→“ POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV服务”→“开启服务”，然后根据要求进行发送短信验证、手机号码验证，最后将得到授权码。将此授权码填入到Send Email节点中的Password输入框中。



附：QQ邮箱获取授权码


QQ邮箱 mail.qq.com
邮箱首页 | 升级为邮箱会员 设置 换肤
1 点击“设置”
反馈建议 | 帮助中心 | 账号与安全 | 切换账号 | 退出
邮件搜索...
写信
收信
通讯录
收件箱(14)
星标邮件
群邮件(1)
草稿箱(11)
已发送
已删除
垃圾箱
我的文件夹
其他邮箱
日历 | 记事本
简历 NEW
发票助手
在线文档
附件收藏
文件中转站
贺卡
邮箱设置
2 点击“账号”
常规 账号 换肤 收信规则 反垃圾 文件夹和标签 其他邮箱 信纸 体验室 会员中心
POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV服务
服务状态：
服务已关闭 开启服务
3 点击“开启服务”
(POP3/IMAP/SMTP/CardDAV/CalDAV服务均支持SSL连接。如何设置？)
温馨提示：在第三方登录QQ邮箱，可能存在邮件泄露风险，甚至危害Apple ID安全，建议使用QQ邮箱手机版登录。
继续开启服务以获取授权码登录第三方客户端邮箱。
收取选项：
最近30天 的邮件
收取“我的文件夹”
收取“QQ邮件订阅”
SMTP发信后保存到服务器
(以上收取选项对POP3/IMAP/SMTP/Exchange均生效。了解更多)
收取垃圾邮件隔离提醒
(该收取选项只对POP3生效。我使用了IMAP/Exchange协议，怎么办？)
同步选项：
禁止收信软件删信（为什么会有收信软件删信？）
(该收取选项只对POP3生效。)
注销邮箱
前往安全中心注销邮箱
(注销后QQ邮箱将不再收取任何来信，注销等待期后将删除邮箱中全部邮件及邮箱应用数据，删除不可恢复。)

附：QQ邮箱获取授权码

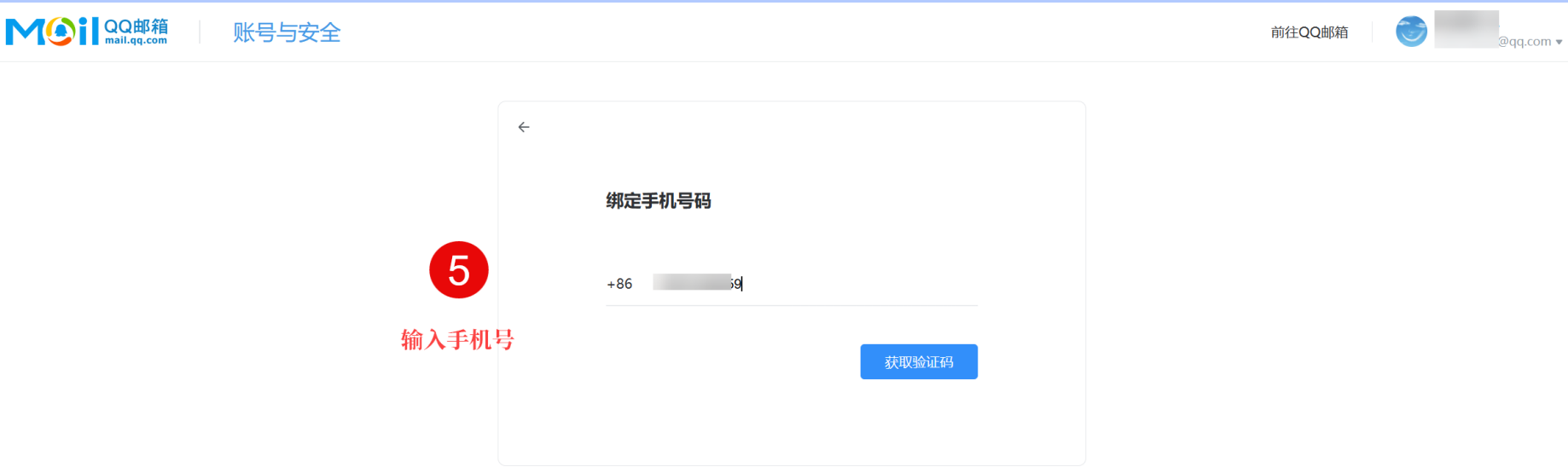
QQ邮箱 mail.qq.com | 账号与安全

前往QQ邮箱 [redacted]@qq.com

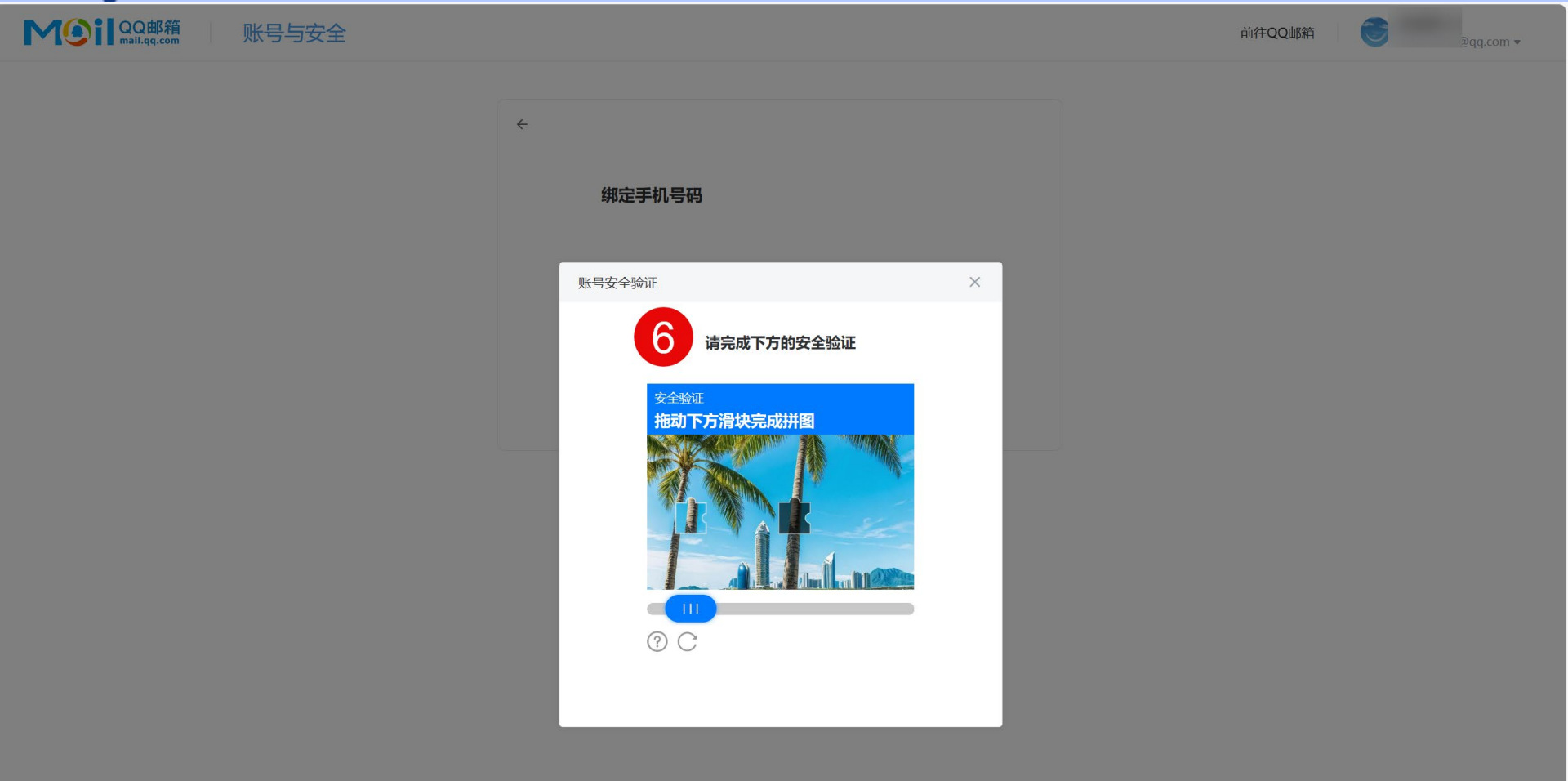
4



附：QQ邮箱获取授权码



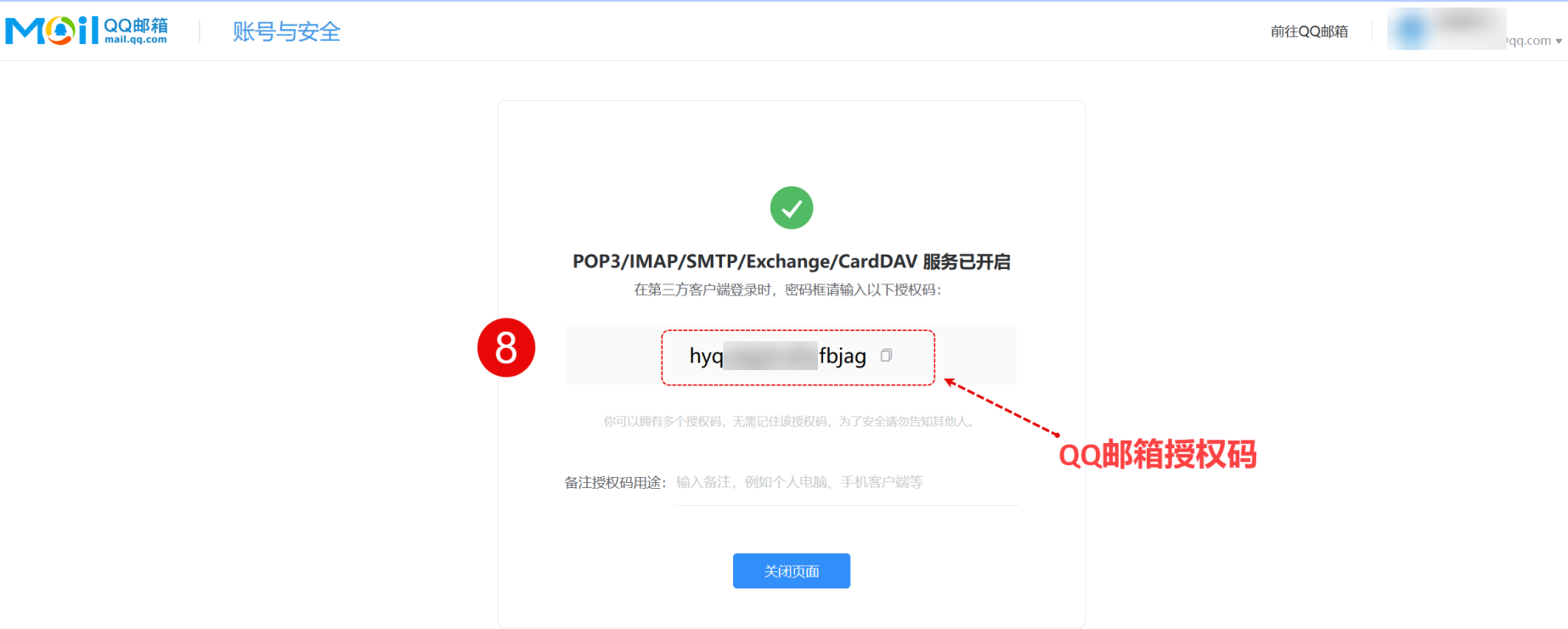
附：QQ邮箱获取授权码


QQ邮箱
mail.qq.com
账号与安全
前往QQ邮箱
绑定手机号码
账号安全验证
6
请完成下方的安全验证
安全验证
拖动下方滑块完成拼图

附：QQ邮箱获取授权码


QQ邮箱
mail.qq.com
账号与安全
前往QQ邮箱
@qq.com
7
输入验证码
短信验证码已发送至 +86 9
347771
42秒后重试
输入手机收到的验证码
提交验证

附：QQ邮箱获取授权码


QQ邮箱 mail.qq.com
账号与安全
前往QQ邮箱
POP3/IMAP/SMTP/Exchange/CardDAV 服务已开启
在第三方客户端登录时，密码框请输入以下授权码：
8
hyq fbjag
你可以拥有多个授权码，无需记住该授权码，为了安全请勿告知其他人。
备注授权码用途：输入备注，例如个人电脑、手机客户端等
关闭页面
QQ邮箱授权码

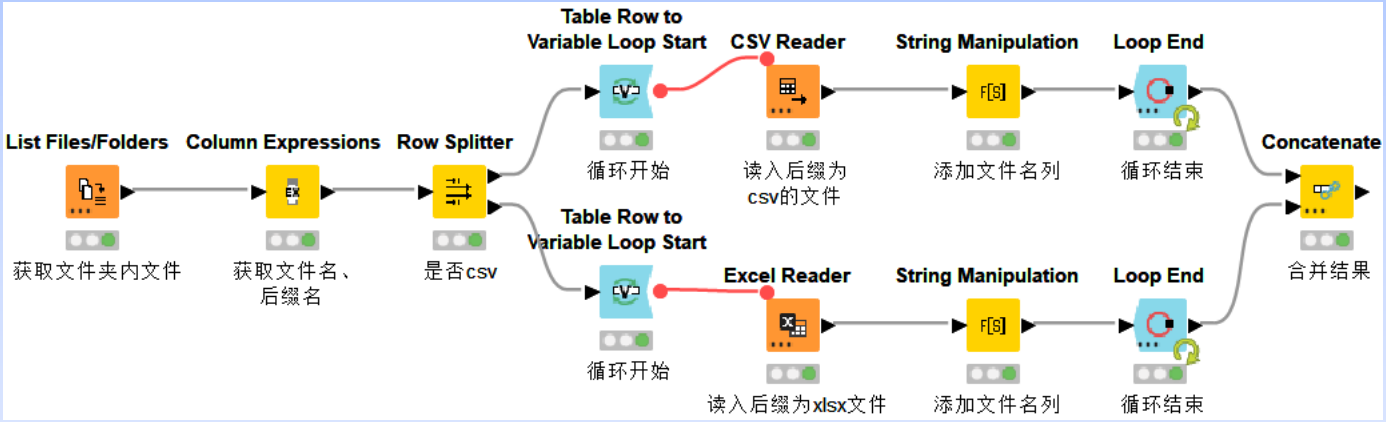


04

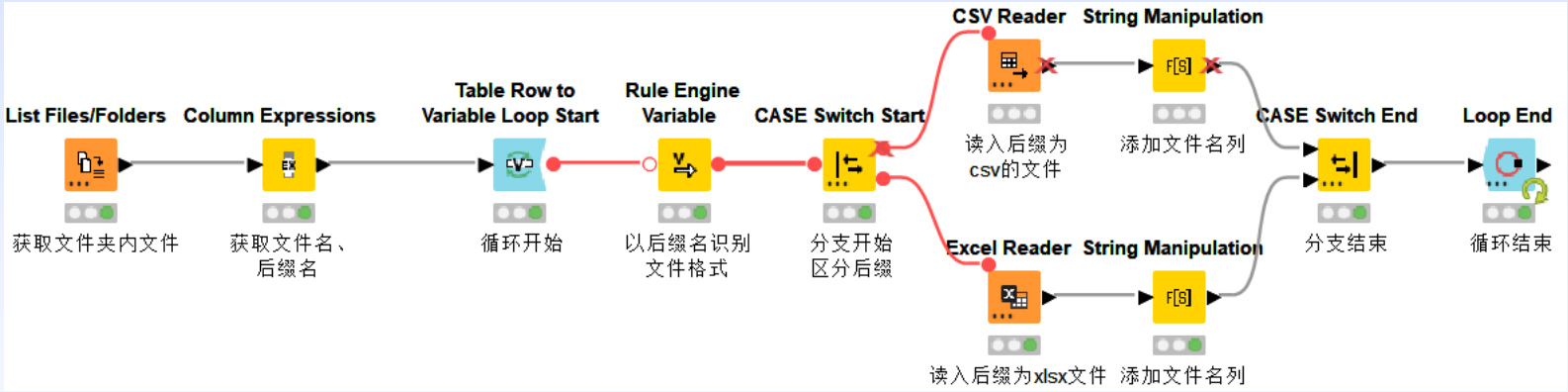
分支与循环

基本概念

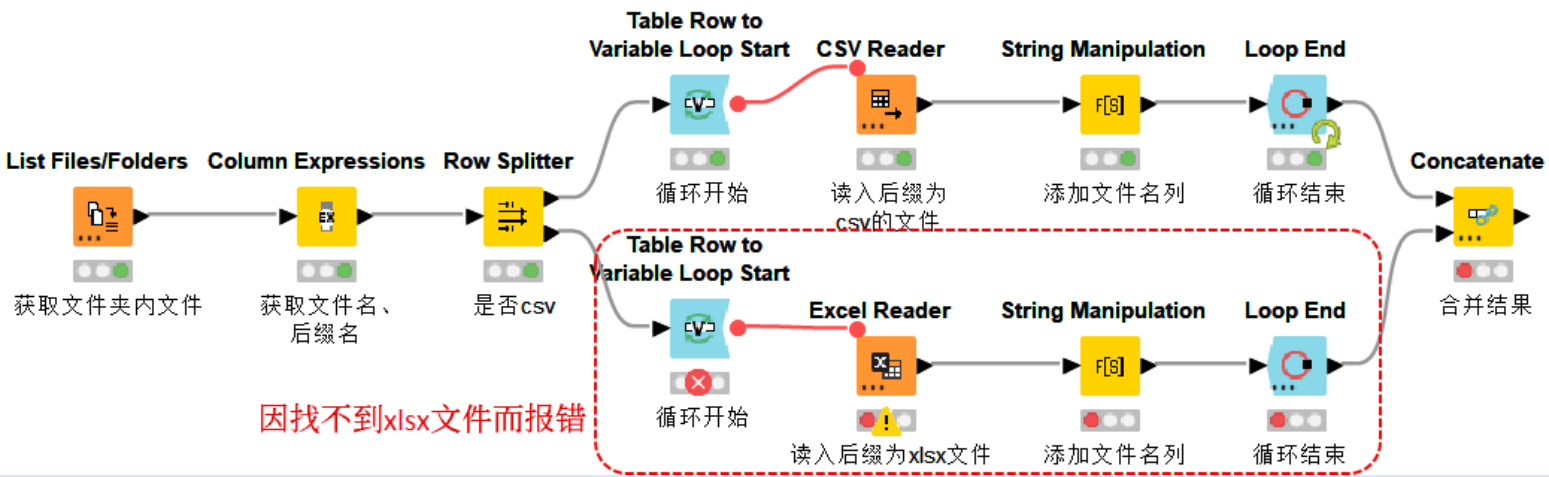
不使用分支节点



使用分支节点



实现方法



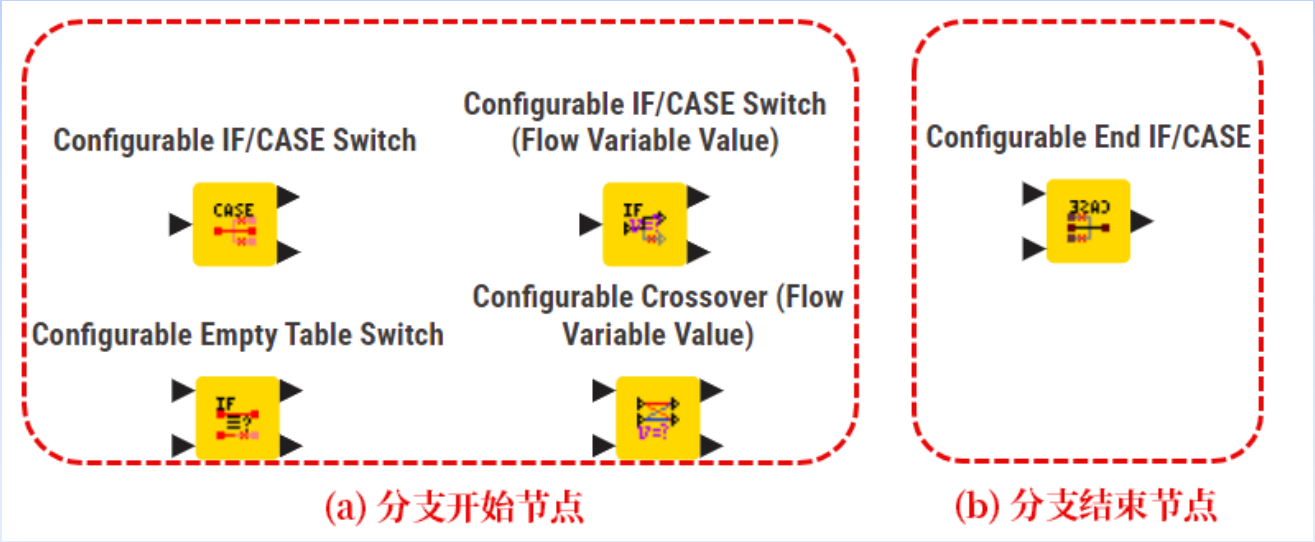
- 未使用分支的工作流之所以报错，是因为底部工作流需要读入xlsx后缀格式的Excel文件，但没有此格式文件。
- 而分支工作流中，即使移除Excel类型文件，另外一个分支还可以正常执行。对比两者，分支处理的更方便，容错能力更强。
- 对于两个不同版本工作流，前者要求两个支线工作流必须都能执行，后者分支版工作流中，能执行其一分支即可，请对比体会两者差异。



05

其他扩展分支

可配置交叉分支介绍


Configurable IF/CASE Switch
Configurable IF/CASE Switch (Flow Variable Value)
Configurable Empty Table Switch
Configurable Crossover (Flow Variable Value)
Configurable End IF/CASE
(a) 分支开始节点
(b) 分支结束节点



06

错误处理Try和Catch Errors

错误处理概述

01

错误处理基本概念

错误处理是设计工作流的重要环节，用于识别和纠正程序中的错误。通过有效的错误处理机制，可以提高工作流运行的稳定性和可靠性，减少运行时错误对用户体验的影响。

02

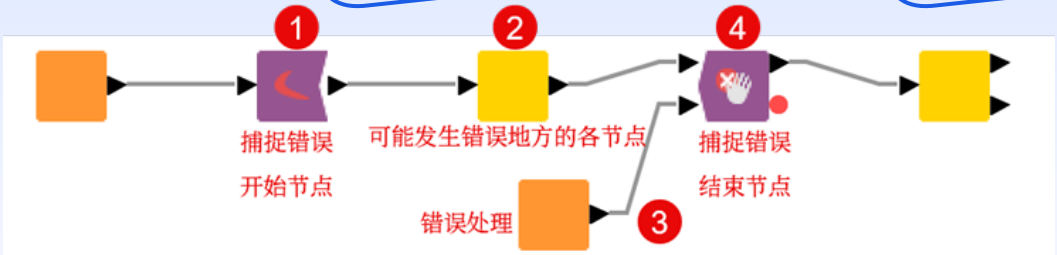
错误处理机制

这些机制可以帮助开发者在错误发生时进行有效的错误捕获和处理，从而避免错误的扩散和影响。最主要的是避免错误导致工作流异常终止执行。

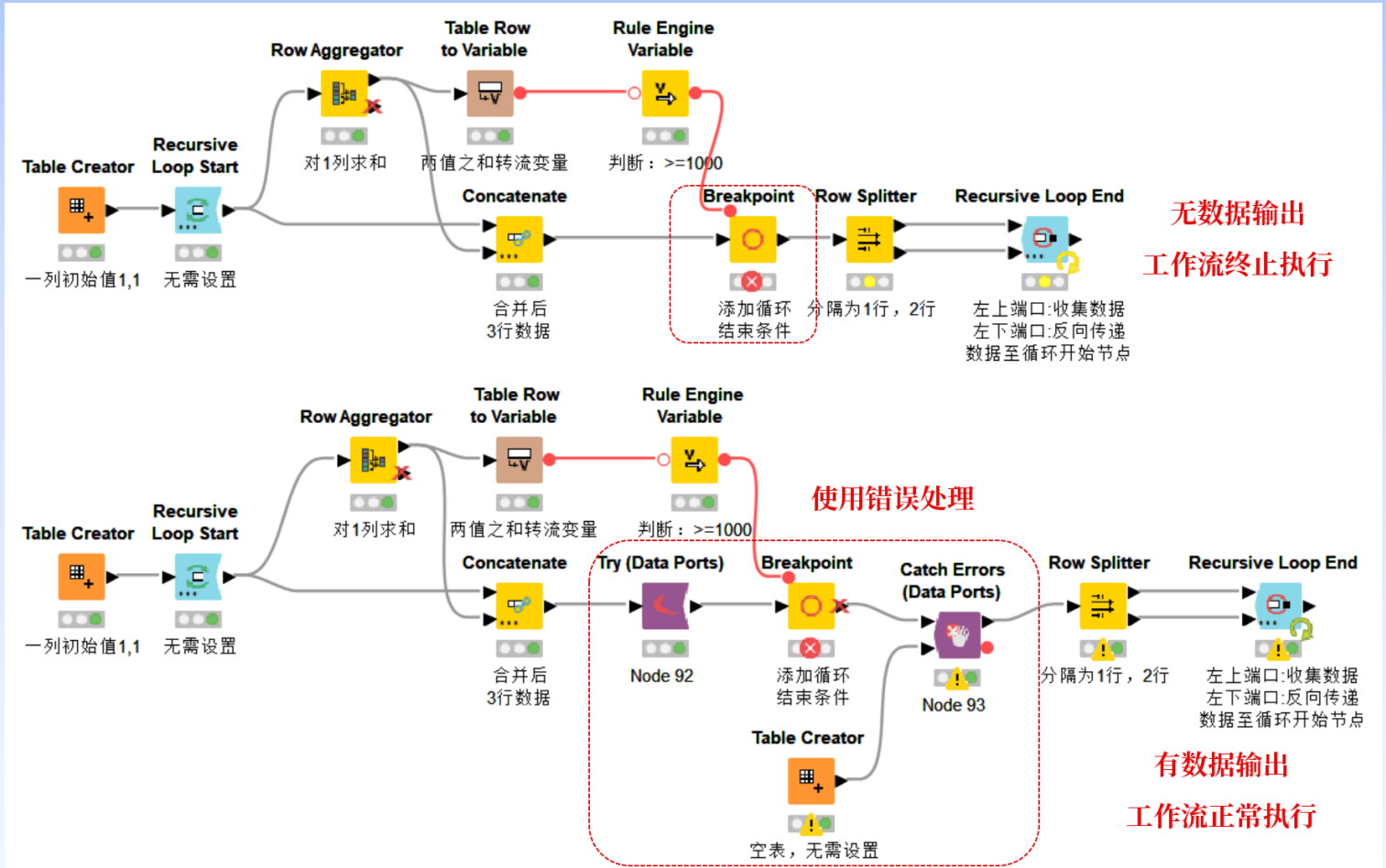
03

常见错误类型

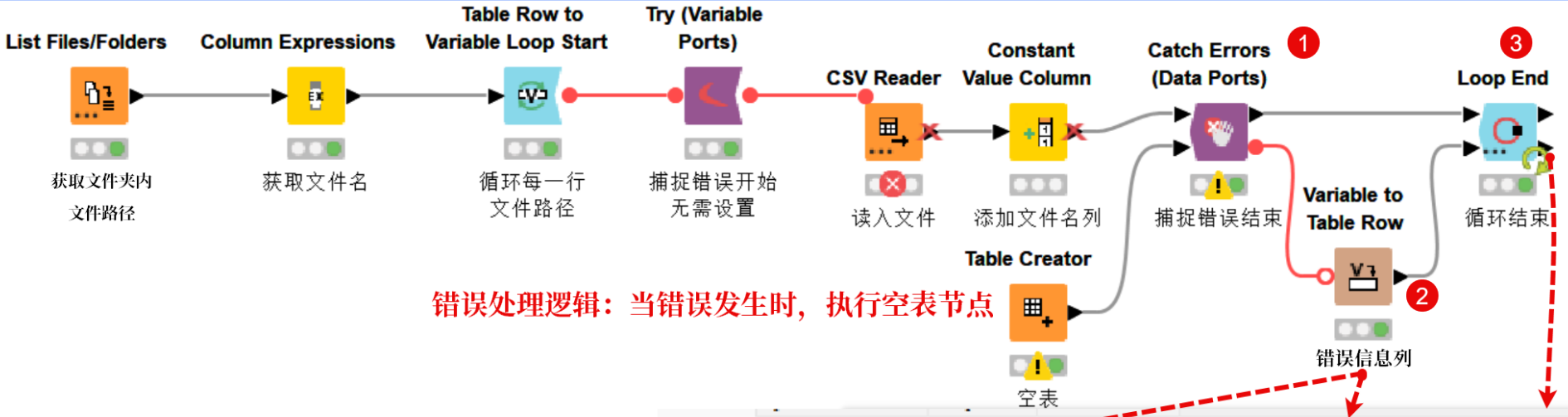
程序在运行过程中可能遇到各种类型的错误，如文件找不到、路径不存在、内存不足、输入错误等。不同类型的错误需要不同的处理方式，因此了解错误的种类有助于更有效地应对和修复问题。



错误处理节点示例


Table Creator
Recursive Loop Start
Row Aggregator
Table Row to Variable
Rule Engine Variable
一列初始值1,1
无需设置
对1列求和
两值之和转流变量
判断：>=1000
Concatenate
合并后 3行数据
Breakpoint
添加循环 结束条件
Row Splitter
分隔为1行，2行
Recursive Loop End
左上端口:收集数据
左下端口:反向传递
数据至循环开始节点
无数据输出
工作流终止执行
使用错误处理
Try (Data Ports)
Node 92
Catch Errors (Data Ports)
Node 93
Table Creator
空表，无需设置
有数据输出
工作流正常执行

错误处理节点应用—合并多个数据文件、跳过读入错误类节点



列出出错的文件名和执行状态，方便事后核对。
错误信息使用默认信息，也可以在Catch Errors节点中自定义错误信息

文件名	Itera...	FailingNodeMessage
01.csv	0	?
02.csv	1	?
03.csv	2	?
04_空格及逗号分隔符.csv	3	The data row has too few data elements.
05_多一列（表头列与实际数据列不一致）.csv	4	The data row has too many data elements.



07

本章小结

主要知识点回顾

分支结构基本概念

分支结构是工作流中用于控制程序流程的基础结构，根据条件的不同选择不同的执行路径。常见节点包括if和switch类节点，它们根据流变量条件的真假执行相应的分支工作流。

分支节点分类与应用

分支节点主要分为二分支和多分支。单分支结构根据条件直接执行相应语句；二分支结构通过if条件语句判断并选择不同路径；多分支结构则通过发送不同模版邮件进行了详细介绍。



错误处理机制概述

错误处理机制是确保程序在遇到异常情况时能够正确响应的重要部分。常见的错误处理为Try、Catch Errors节点，它允许两个节点范围之间的节点在出现错误时继续执行另外的预定操作，从而保证工作流运行的可靠性和稳定性。

分支与循环结合使用

分支与循环的结合使用可以有效控制程序的流程。在分支语句中引入循环语句，可以实现更复杂的控制逻辑，如迭代处理数据或条件检查。

总结

掌握分支结构基本要素

学习了分支结构的四要素：分支开始、分支工作流、分支控制条件和分支结束。理解这些要素有助于编写高效的分支代码，确保程序在不同条件下正确执行相应操作。

注意：**四个要素不是同时必须都具备，例如分支结束节点并不是必须。**

错误处理机制重要性

学习了使用try-catch错误处理机制来管理异常情况。掌握了如何在工作流中合理地设计错误处理流程，以提高工作流的稳定性。

错误处理机制，本质上也是一种特殊的二分支结构



多分支策略实际应用

通过案例应用与优化，学习了如何使用CASE Start和IF节点来处理复杂业务逻辑。掌握了多分支策略在解决实际问题中的应用方法，提高了工作流的可读性和逻辑性。

分支与循环综合应用

探讨了分支与循环结构的综合运用，如IF/CASE类节点与循环结合使用。通过实际案例，理解了如何利用这些结构实现复杂的业务逻辑和数据处理。

感谢观看

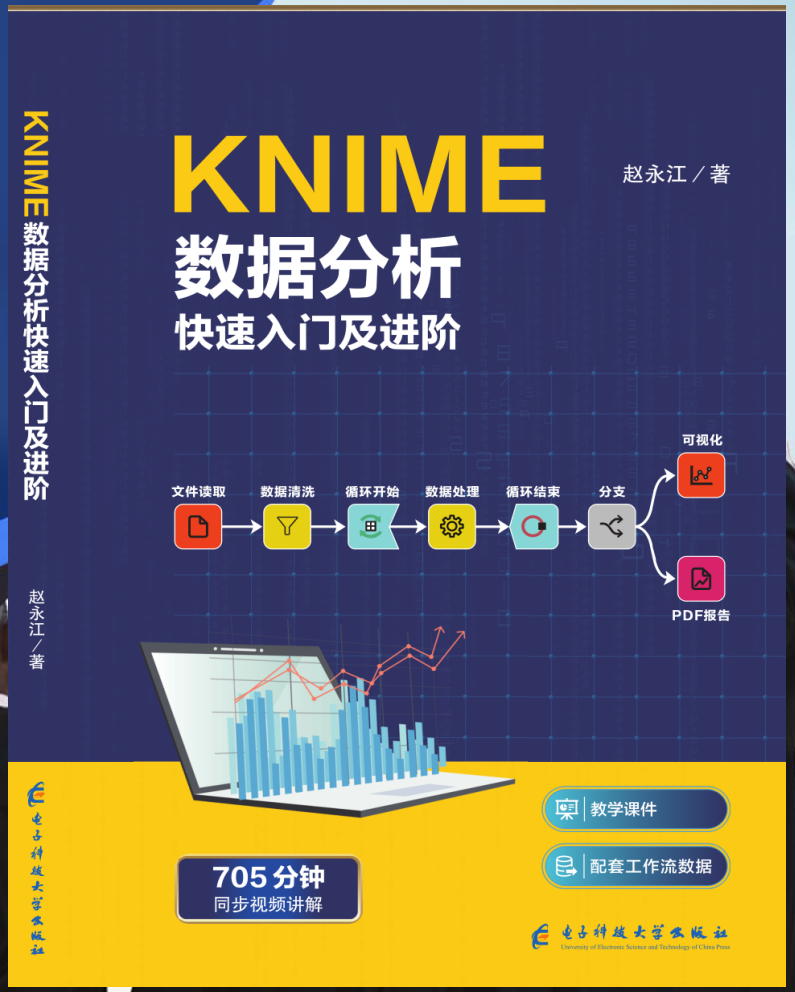
THANKS



第12章

字符串高级操作

掌握正则表达式与高阶数据处理技巧



目录

01 概述

02 正则表达式基础

03 KNIME正则节点及应用

04 易错易混正则知识点

05 其他高阶操作

06 本章小结



01

概述

正则表达式基础简介

基本匹配

正则表达式由字母和数字组合而成，用于执行搜索时的格式。它是通过点运算符、字符集、重复次数等元字符进行定义，能够实现对字符串的精确匹配和查找。

元字符

正则表达式使用特殊字符（元字符）来增强其匹配能力。例如，点运算符可以指定匹配位置，字符集定义了可匹配的字符范围，而重复次数则决定了匹配的次数。



位置锚定

通过使用位置锚定符号，如^、$，可以确定匹配字符出现的位置。这些符号帮助定位字符串中特定部分，从而更精准地进行模式匹配和提取数据。

分组与引用

通过括号将字符捆绑在一起，形成捕获组。这不仅可以简化正则表达式，还能用于后续的引用和操作，如后向引用和前向引用，提高表达式的灵活性和效率。

字符串高级操作概览

01

字符串高级操作概述

字符串高级操作包括字符串的分割、连接、替换、搜索等操作，这些操作可以有效地增强字符串的处理能力，提高 数 据处理和分析的灵活性和效率。

02

字符串基本操作

字符串的基本操作包括字符串的创建、赋值、拼接、长度计算等，这些操作是字符串处理的基础，也是进一步进行高级操作的前提。

03

字符串高级操作技巧

字符串高级操作的技巧包括使用正则表达式进行模式匹配、通过切片操作实现字符串的提取与修改，以及应用字符串操作方法解决实际问题。

高阶操作技术要点

字符串相似度比较

字符串相似度比较可以通过编辑距离、Jaccard指数等方法实现。编辑距离用于衡量两个字符串通过插入、删除和替换操作转换为另一个字符串所需的最小操作次数。Jaccard指数则通过计算两个字符串对应位置字符的相似度，来评估它们之间的相似性。

数据标准化

数据标准化是将不同格式的数据转换为统一格式的过程，以确保数据的一致性和可比性。常见的数据标准化方法包括使用标准编码、统一度量单位和数据清洗等技术。这些方法有助于提高数据的质量和分析的准确性，为后续的数据处理和挖掘打下坚实基础。



数据脱敏与加密

数据脱敏与加密是保护敏感信息的重要手段。数据脱敏通过对原始数据进行处理，去除或替换特定字段中的敏感信息，如身份证号、电话号码等，以确保隐私数据不被泄露。数据加密则是通过算法将数据转化为不可读的密文，只有持有密钥的用户才能解密查看，有效防止数据在传输过程中被窃取或篡改。

数据验证

数据验证是通过一系列规则和算法确保数据的准确性、完整性和一致性的过程。常见的数据验证方法包括基于规则的校验、基于模型的校验和基于统计的校验等。有效的数据验证能够降低错误数据带来的风险，提升数据分析和应用的质量，为决策提供可靠依据。



02

正则表达式基础

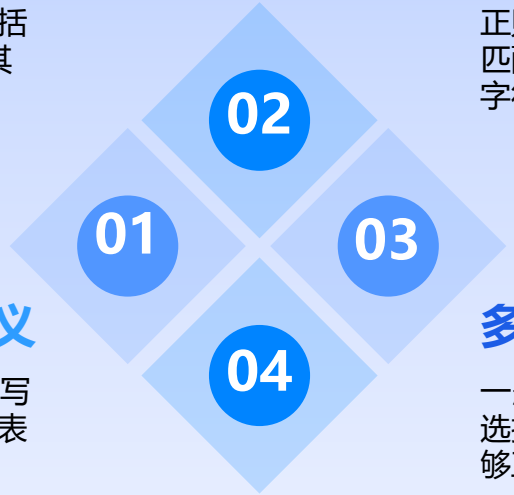
字符及字符集定义

基本字符集

正则表达式中的"."字符代表任意单一字符，包括空格。"+"表示前一个字符的前面有至少一个其他字符，"*"可以匹配零次或多次前一个字符。

特殊字符集

正则表达式支持多种特殊字符，如"^"用于匹配输入字符串的开始位置，"$"匹配输入字符串的结束位置，"|"用于表示"或"操作。



字符类定义

通过方括号[]定义字符类，如[a-z]匹配小写字母，[0-9]匹配任意数字。使用连字符-表示范围，例如[a-c]匹配a、b、c。

多行文本处理

一般在KNIME节点中，通过勾选界面选项选择是否处理多行文本，确保正则表达式能够正确识别和匹配目标字符串。

次数、位置与匹配模式

01

次数

正则表达式中的"*"代表零次或多次，"+"代表一次或多次，"{n}"代表至少n次，"{n,}"代表n次到无穷大。这些运算符用于控制字符串中特定模式出现的次数，提高匹配的灵活性。

02

位置

正则表达式支持使用特殊字符如^、$来定位字符串的开始和结束，以及使用方括号[]来指定模式出现的位置。例如，`^[a-z]+$`匹配任意小写字母组成的字符串，并确保其出现在字符串的开始和结束位置。正则断言也常用来匹配指定条件的某个位置字符串。

03

匹配模式

正则表达式提供了多种匹配模式， 如 贪婪与非贪婪模式、四类断言模式，其他还 有 全局匹配、词边界(b/B)、多行等。全局匹配使模式在多行文本中也有效，词边界匹配确保模式仅在字符串的词边界处匹配。

断言、正则修饰符介绍

01 断言基本概念

正则表达式的断言是一种模式匹配技术，用于在匹配时对字符串进行条件性的预测。它不会消耗输入字符串，仅仅是在匹配的位置上进行条件判断，常用于限定匹配位置。

02 正向断言与反向断言

正向断言用于确认匹配字符串的开始，如正向肯断言定“Office(?=97|2000|2003|2007)”可以匹配字符“Office2007”中的“Office”，但不能从字符串“Office2016”中匹配出字符“Office”。

正向与反向是相对于目标字符串Office而言，辅助定位条件在右侧为正向，在左侧为反向。

03 正则修饰符介绍

正则修饰符为正则表达式提供了额外的控制和优化选项，这些修饰符可以增强正则表达式的功能，提高匹配效率和准确性。KNIME中一般通过正则函数中的参数、或正则节点界面选项设置。

运算符优先级说明

括号运算符优先级

括号运算符在正则表达式中具有最高优先级，用于定义子表达式。圆括号 () 的优先级高于方括号 [] 和其他运算符，确保表达式从内向外进行计算。

量词运算符优先级

量词运算符指定字符串元素的重复次数，如 * 匹配零个或多个元素，+ 匹配一个或多个元素。量词运算符的优先级次于字符类和连接运算符。

特殊字符优先级

转义字符优先级最高，用于改变紧跟其后的字符含义，如 \d代表一个数字

管道运算符优先级

管道符或竖线 |，正则表达式中的“或”操作符，匹配两个表达式中的一个，优先级一般是最低的。

断言运算符优先级

断言运算符用于检查字符串中特定条件的元素，如 ^ 匹配行开始，$ 匹配行结束。断言运算符的优先级低于其他算术运算符，但高于大多数正则表达式运算符。



03

KNIME正则节点及应用

基于正则式匹配一个或多个



正则表达式

[^,]+公司

前三个相同

Regex Extractor

每个匹配结果为一行

流水号	融资人	担保人	融资金额	Full Match
1001	张三	李志强,A有限公司,B股份有限公司,张晓蕾	20000	A有限公司
1001	张三	李志强,A有限公司,B股份有限公司,张晓蕾	20000	B股份有限公司
1002	李四	赵文博,C有限公司,周瑞阳,D有限公司,...	80000	C有限公司
1002	李四	赵文博,C有限公司,周瑞阳,D有限公司,...	80000	D有限公司
1002	李四	赵文博,C有限公司,周瑞阳,D有限公司,...	80000	E有限公司

Excel Reader

融资担保人信息

Regex Substring

返回字符串类型

Regex To List

返回列表类型

流水号	融资人	担保人	融资金额	担保人_regexSubstring	担保人_regexList
1001	张三	李志强,A有限公司,B股份有限公司,张晓蕾	20000	A有限公司,B股份有限公司	[A有限公司,B股份有限公司]
1002	李四	赵文博,C有限公司,周瑞阳,D有限公司,...	80000	C有限公司,D有限公司,E有限公司	[C有限公司,D有限公司,E有限公司]

Regex Split

提取：第一个公司

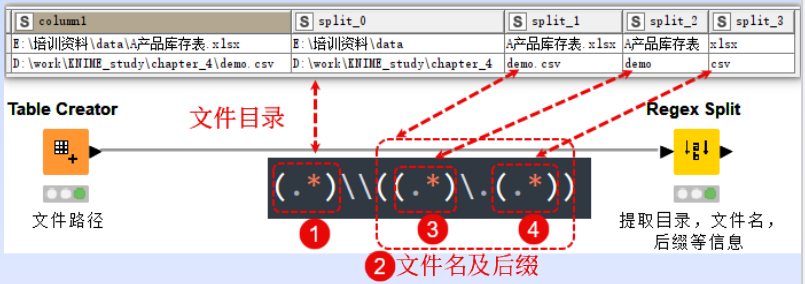
Regex Split

提取：最后一个公司

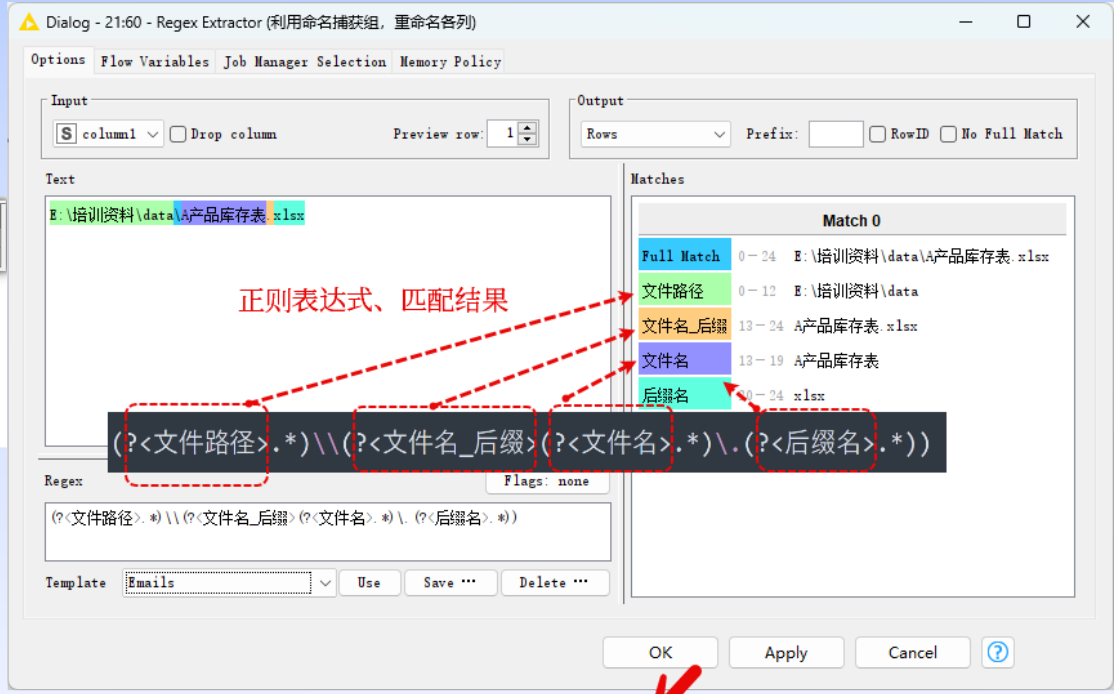
流水号	融资人	担保人	融资金额	split_0	split_0 (#1)
1001	张三	李志强,A有限公...	20000	A有限公司	B股份有限公司
1002	李四	赵文博,C有限公...	80000	C有限公司	E有限公司

基于捕获组匹配一个或多个

方法1



方法2：使用捕获组命名



输出结果

column1	Full Match	文件路径	文件名_后缀	文件名	后缀名
E:\培训资料\data\A产品...	E:\培训资料\data\A产品库存表.xlsx	E:\培训资料\data	A产品库存表.xlsx	A产品库存表	xlsx
D:\work\KNIME_study\cha...	D:\work\KNIME_study\chapter_4\demo.csv	D:\work\KNIME_study\chapter_4	demo.csv	demo	csv

注意：

- Regex Split节点不支持捕获组命名。
- 5.3及之后版本中，Regex Split节点更名为String Splitter（Regex），更名后支持捕获组命名，但暂不支持中文命名，英文命名是可以的。

捕获组与捕获组命名

捕获组定义与作用

正则表达式中的捕获组使用圆括号()定义，用于保存匹配的子字符串。它允许我们将多个子模式组合在一起，并捕获这些子模式匹配的文本，方便后续处理或引用。

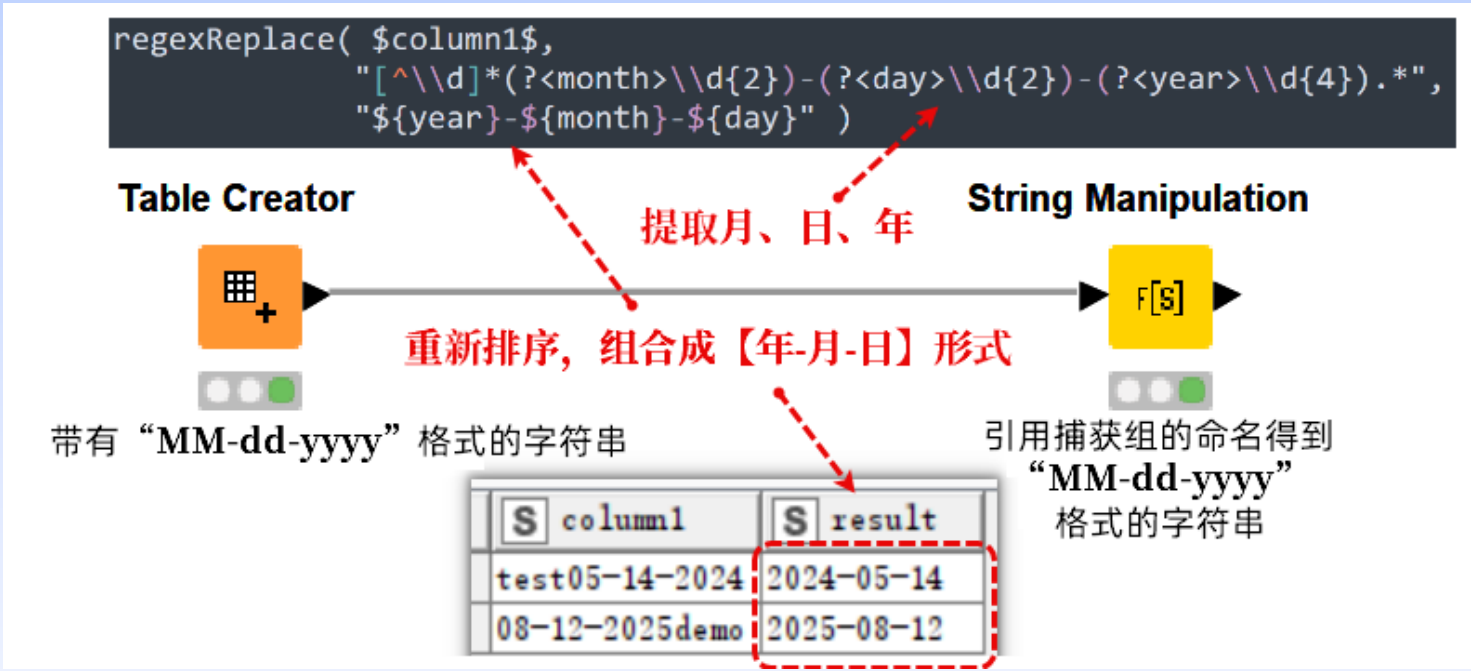
普通捕获组与命名捕获组

捕获组主要分为普通捕获组和命名捕获组。普通捕获组使用默认编号进行标识（如String Manipulation节点正则函数两种都可以使用，本书皆有示例），而命名捕获组通过显式命名进行标识。命名捕获组目前只有部分语言支持，如PHP、Python、Java。

捕获组工作原理

当正则表达式引擎匹配到捕获组时，它会记录与每个捕获组匹配的文本片段。这些捕获的文本可以用于后续的处理或替换，帮助实现更复杂的字符串操作。

命名捕获组赋值与引用—示例



字符串匹配替换或编辑

正则表达式基本概念

正则表达式是一种文本模式，用于描述字符串中重复出现的模式。它由普通字符和特殊字符组成，可以匹配、替换或验证字符串中的特定模式。

字符串替换操作

使用正则表达式进行字符串的替换操作。



多步替换和捕获组

通过使用正则表达式中的捕获组，可以实现多步替换。捕获组可以保存匹配到的特定子串，并在后续替换操作中使用这些捕获的数据，从而实现复杂的字符串处理任务。

基于正则表达式的字符串操作实用案例

正则表达式在处理日志文件、清洗数据、网站爬虫等方面有广泛应用。例如，可以用于提取日期、IP地址、电子邮件等特定信息，并替换或删除不需要的部分。

行列过滤

列过滤基本概念

01

列过滤是数据库查询中的一种重要技术，通过正则表达式在Filter类节点中选择列，可以只选择表中的部分列进行展示或操作，从而提高查询效率和数据安全性。

行过滤应用场景

02

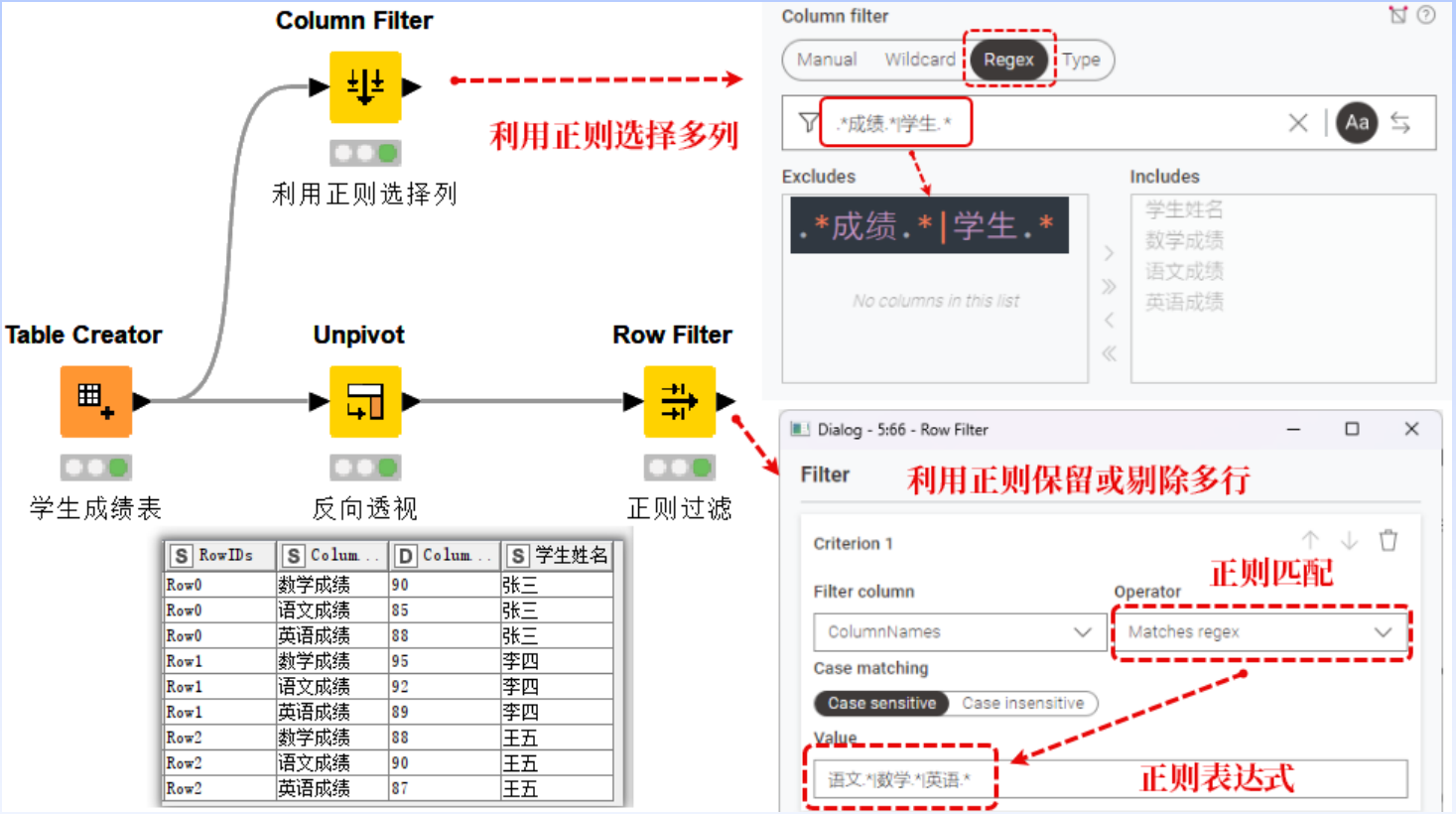
行过滤通常应用于需要对数据集进行过滤或剪裁的操作场景，如Row Filter、Table Cropper节点，确保只有满足特定条件的行才会被返回，这有助于减少不必要的数据传输和存储成本。

列过滤、行过滤均存在正则选择选项

03

通过通配符或正则表达式快速选择相关列或行。

行列过滤—示例



RowIDs	Colum...	Colum...	学生姓名
Row0	数学成绩	90	张三
Row0	语文成绩	85	张三
Row0	英语成绩	88	张三
Row1	数学成绩	95	李四
Row1	语文成绩	92	李四
Row1	英语成绩	89	李四
Row2	数学成绩	88	王五
Row2	语文成绩	90	王五
Row2	英语成绩	87	王五



04

易错易混正则知识点

各正则节点用法比较

节点名	部分正则式or全部正则式	捕获组	双重转义	返回结果
Cell Replacer	部分	不需要	不需要	字符串。
Regex Extractor	部分	不需要	不需要	一行或多行。获取多个结果竖直方向排列。
Regex Extractor	部分	需要	不需要	一列或多列（使用捕获组，支持捕获组命名）。获取多个结果水平方向展开。
Regex Substring	部分	不需要	不需要	一个或多个拼接的字符串。
Regex To List	部分	不需要	不需要	列表。
Regex Split	全部	需要	不需要	一列或多列（每个捕获组为一列）。
Column Rename(Regex)	部分	需要	不需要	字符串，可对捕获组结果引用，编辑。
String Manipulation的regexReplace函数	全部	需要	需要	字符串，可对捕获组结果引用，编辑（支持捕获组命名）。
String Manipulation 的 regexMatcher 函数	全部	需要	需要	对捕获组正则式判断，返回True或False。
Rule Engine，Rule-based类节点的MATCHES函数	全部	不需要	不需要	对整个字符串正则式判断，返回判断为真的各行。

5.3及之后版本中，Regex Split节点更名为String Splitter（Regex），更名后的节点做了升级，支持部分正则式

问号多重含义解析

非贪婪匹配

问号可以用于表示非贪婪匹配，即在正则表达式中，它使前面的字符或子表达式成为可选的。例如，"a+?" 将匹配 "aab" 中的 "ab"，而非整个字符串 "aab"。



零次或一次出现

问号可以将前面的字符或子表达式标记为可选的，即出现零次或一次。例如，"colou?" 可以匹配 "color" 或 "colour"，体现了其灵活的匹配能力。

非贪婪分组

在捕获分组中，问号也用来表示非贪婪匹配。如 (a|b)? 表示 a 或 b 是可选的，这样的分组能够更精准地匹配目标字符串的一部分。

特殊字符转义等

需要注意的是，在正则表达式中，问号通常需要转义为 \?，以正确匹配 "?" 字符本身，避免与普通字符混淆。

详见本书：表12-8

扩号多重含义分析

序号	类别	格式
1	非捕获组	(?:pattern)
2	断言	(?=pattern) (?!pattern) (?<=pattern) (?<!pattern)
3	原子性分组	(?>pattern)
4	捕获组	(pattern)
5	捕获组嵌套	(pattern1(pattern2))
6	限定性重复	(abc)+
7	反向引用	$1， $2， ...
8	捕获组命名	(?<name>pattern)



05

其他高阶操作

字符串相似度比较方法

01

编辑距离（Edit Distance）

编辑距离用于衡量两个字符串之间需要进行多少次操作才能使一个字符串变成另一个字符串。常见的编辑操作包括插入、删除和替换，通过计算最少的编辑次数来衡量字符串间的相似度。

02

余弦相似性（Cosine Similarity）

余弦相似性是一种通过计算两个向量间夹角余弦值来判断其相似程度的方法。在字符串相似度比较中，余弦相似性通过计算两个字符串对应位置字符的加权相关性，得到一个介于-1到1之间的相似度评分。

03

Jaccard 相似性（Jaccard Similarity）

Jaccard 相似性通过计算两个集合交集的大小与并集大小的比值来反映相似度。对于字符串，可以通过计算两个字符串相同字符的数量除以各自字符总数的平方根来得到 Jaccard 相似度，范围从0到1。

04

Levenshtein Distance

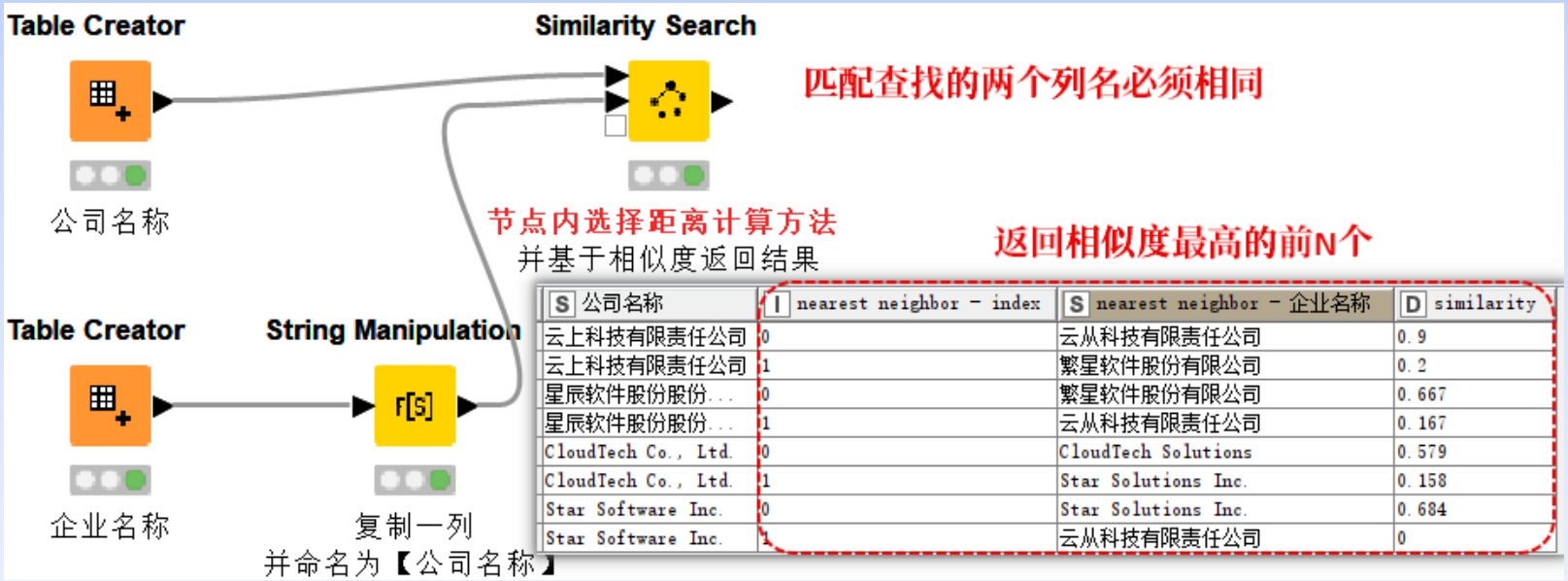
Levenshtein Distance（莱文斯坦距离）是一种衡量字符串差异的方法。它通过计算两个字符串需要经过多少次替换、插入或删除操作才能从其中一个字符串变成另一个字符串。这种方法适合量化不同拼写变体之间的相似度。

05

汉明距离（Hamming Distance）

汉明距离用于衡量两个等长字符串之间对应位置上不同字符的个数，是另一种常用的字符串相似度度量方法。汉明距离越小，表示两个字符串越相似。常用于文本纠错和数据清洗等场景。

字符串相似度比较—示例



S 公司名称	I nearest neighbor - index	S nearest neighbor - 企业名称	D similarity
云上科技有限责任公司	0	云从科技有限责任公司	0.9
云上科技有限责任公司	1	繁星软件股份有限公司	0.2
星辰软件股份股份...	0	繁星软件股份有限公司	0.667
星辰软件股份股份...	1	云从科技有限责任公司	0.167
CloudTech Co., Ltd.	0	CloudTech Solutions	0.579
CloudTech Co., Ltd.	1	Star Solutions Inc.	0.158
Star Software Inc.	0	Star Solutions Inc.	0.684
Star Software Inc.	1	云从科技有限责任公司	0

数据脱敏、加密及解密技术

数据脱敏概述

数据脱敏是一种保护敏感信息和隐私数据的技术，通过策略性地修改或替换原始数据，创建不含真正敏感信息的副本。常见应用场景包括开发、测试、分析和培训等，确保数据在非生产环境中的安全使用。

静态数据与动态数据脱敏

数据脱敏主要分为静态数据脱敏和动态数据脱敏。静态数据脱敏针对数据库中的静态数据进行一次性处理，而动态数据脱敏则在数据访问过程中实时处理，确保每次查询结果都不包含敏感信息。

数据脱敏基本原则

数据脱敏需遵循五项基本原则：防逆向破解原则、表征原始数据原则、引用完整性原则、防数据推理原则及防重复原则。这些原则确保了脱敏后的数据既安全又保持了一定的实用性。

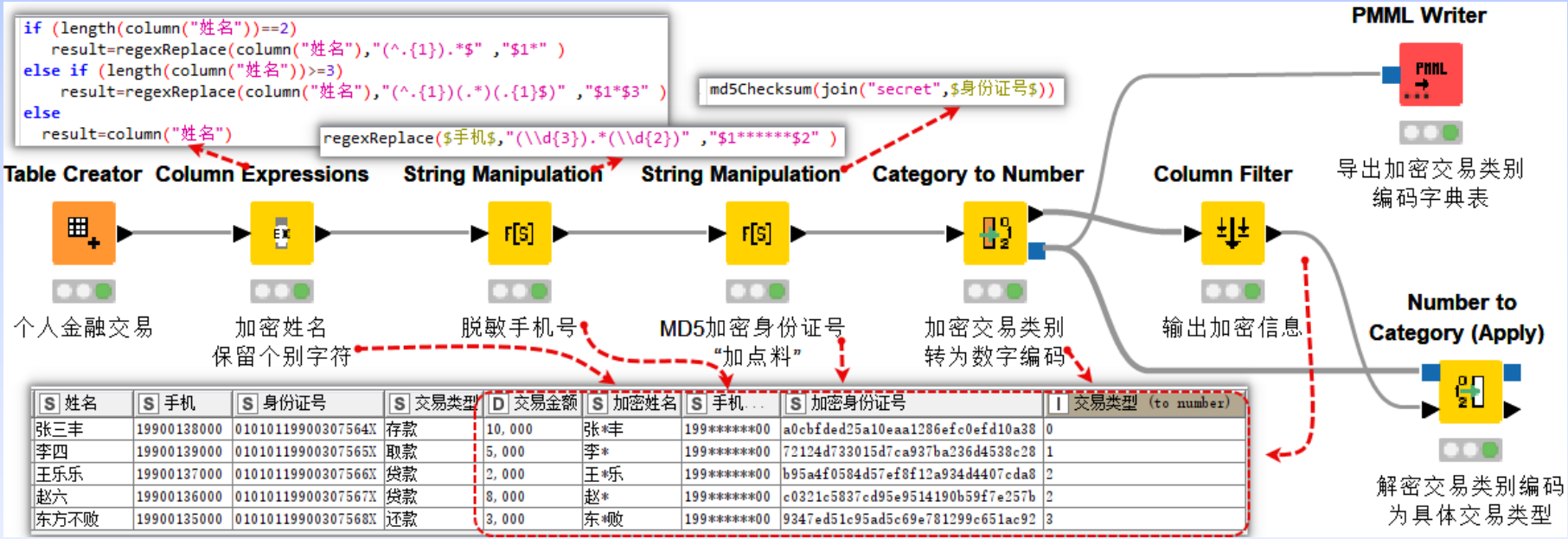
数据脱敏实施步骤

数据脱敏的实施通常分为四个步骤：敏感数据发现、敏感数据梳理、脱敏方案制定和脱敏任务执行。每个步骤结合特定的算法和规则，确保数据在应用环境中既能实现安全保护，又能符合实际需求。

数据脱敏技术应用场景

数据脱敏广泛应用于金融、运营商和企业等领域。其高效且灵活的处理方式，使得组织能够在满足合规要求的同时，安全共享和测试敏感数据，有效降低了数据泄露的风险。

数据脱敏、加密及解密—示例



姓名	手机	身份证号	交易类型	交易金额	加密姓名	手机	加密身份证号	交易类型 (to number)
张三丰	19900138000	01010119900307564X	存款	10,000	张*丰	199*******00	a0cbfded25a10eaa1286efc0efd10a38	0
李四	19900139000	01010119900307565X	取款	5,000	李*	199*******00	72124d733015d7ca937ba236d4538c28	1
王乐乐	19900137000	01010119900307566X	贷款	2,000	王*乐	199*******00	b95a4f0584d57ef8f12a934d4407cda8	2
赵六	19900136000	01010119900307567X	贷款	8,000	赵*	199*******00	c0321c5837cd95e9514190b59f7e257b	2
东方不败	19900135000	01010119900307568X	还款	3,000	东*败	199*******00	9347ed51c95ad5c69e781299c651ac92	3

数据标准化策略讨论

01 数据标准化定义和目的

数据标准化是通过数学变换将数据集统一到相同的尺度上，确保不同量纲的数据可以在同一标准下进行比较和分析。其目的是消除数据差异，提升数据的一致性和可比性。

02 常用数据标准化方法

常用的数据标准化方法包括最小-最大法、Z分数法、小数定标法和归一化法，如Normalizer节点。每种方法适用于不同的数据特征和场景，具有各自的优缺点，需要根据具体需求选择合适的方法。

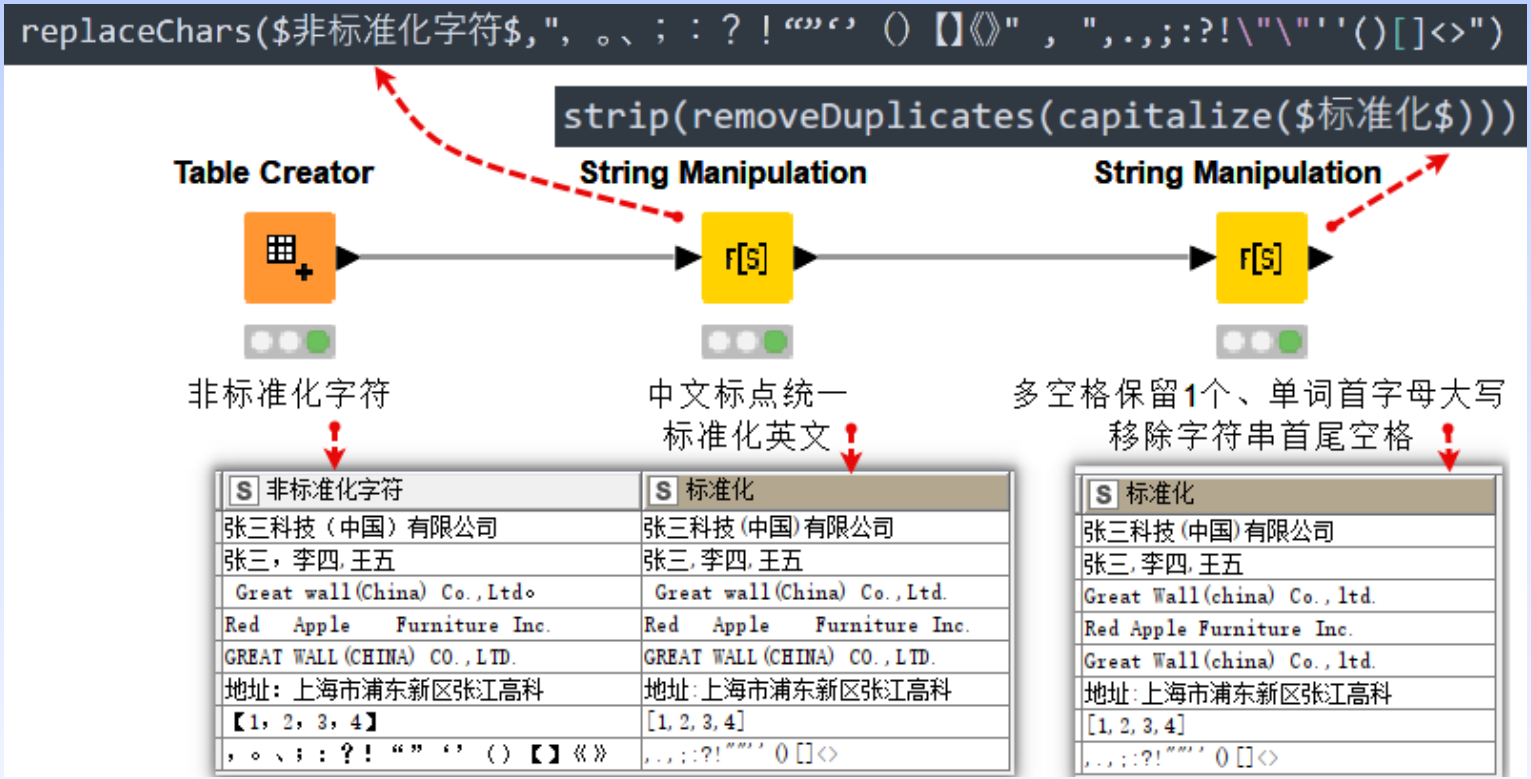
03 数据标准化在机器学习中应用

在机器学习中，数据标准化是重要的预处理步骤。常见方法有均值归一化、向量归一化和指数转换等，这些方法通过缩放数据使它们落入特定的区间，便于模型的学习和训练。

04 数据标准化方法选择与影响

数据标准化方法的选择会影响数据分析和评价的结果。不同的标准化方法适用于不同的数据特征和应用场景，没有通用法则可循，需结合具体需求进行选择。

数据标准化—示例


replaceChars($非标准化字符$,", 。、；：？！“”‘’ () 【】《》" , ",.,;:?!\"\"''()[]<>")
strip(removeDuplicates(capitalize($标准化$)))
Table Creator
String Manipulation
String Manipulation
r[s]
r[s]
非标准化字符
中文标点统一
标准化英文
多空格保留1个、单词首字母大写
移除字符串首尾空格
非标准化字符
张三科技（中国）有限公司
张三，李四，王五
Great wall(China) Co.,Ltd。
Red Apple Furniture Inc.
GREAT WALL(CHINA) CO.,LTD.
地址：上海市浦东新区张江高科
【1，2，3，4】
，。、；：？！“”‘’ () 【】《》
标准化
张三科技(中国)有限公司
张三,李四,王五
Great wall(China) Co.,Ltd.
Red Apple Furniture Inc.
GREAT WALL(CHINA) CO.,LTD.
地址:上海市浦东新区张江高科
[1,2,3,4]
,.,;:?!""' 0 []<>
标准化
张三科技(中国)有限公司
张三,李四,王五
Great Wall(china) Co.,ltd.
Red Apple Furniture Inc.
Great Wall(china) Co.,ltd.
地址:上海市浦东新区张江高科
[1,2,3,4]
,.,;:?!""' 0 []<>

数据验证方法总结

数据类型验证

数据类型验证用于确保数据符合预期的类型，如整数、浮点数或字符串等。通过定义数据类型的范围和格式规则，可以有效防止非法数据输入，提高数据准确性。如KNIME中Integer Configuration节点可对整数可以设置取值范围。

唯一性验证

唯一性验证用于确保数据集中没有重复的记录。通过定义字段的唯一约束条件，可以确保每个记录都具有唯一的值，从而避免数据冗余和错误。

综合验证策略

综合验证策略结合多种验证方法，如数据类型、长度、日期格式、唯一性和非空验证，可以更全面地保障数据质量。这种策略适用于复杂系统的数据管理，确保数据的准确性和可靠性。

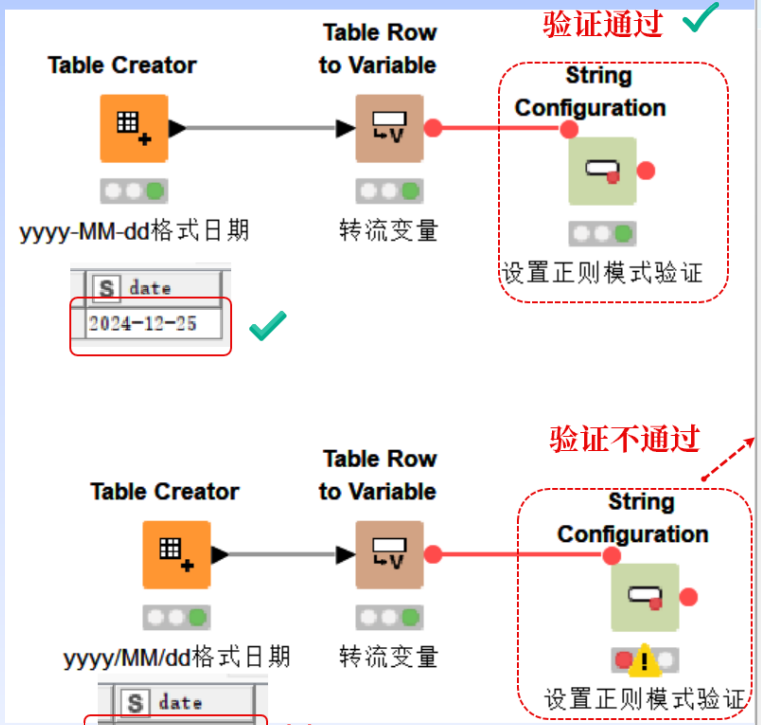
长度限制验证

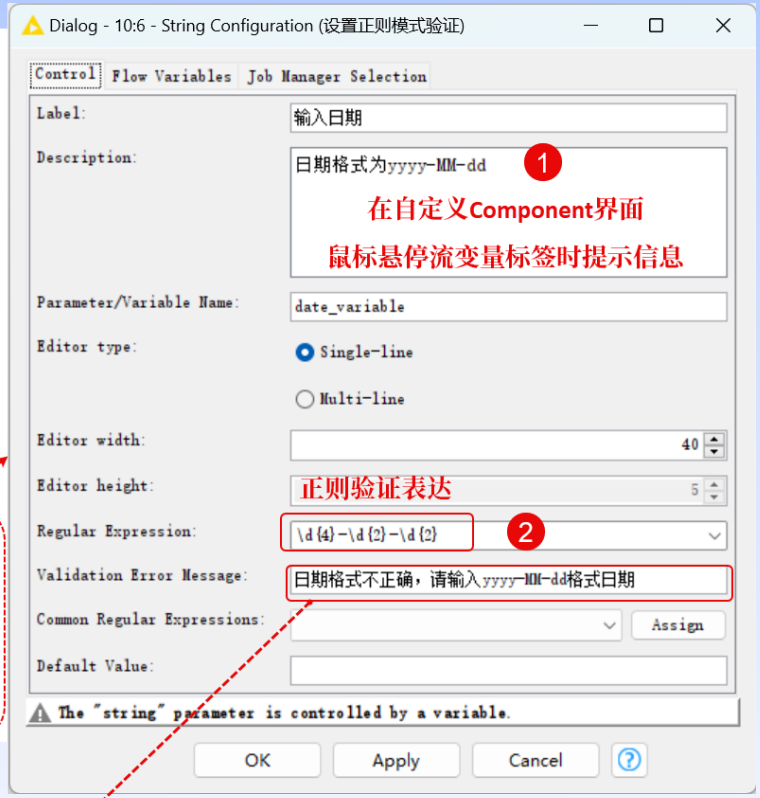
长度限制验证用于确保字段值的长度在指定范围内。这通常通过设置最大最小长度来实现，以防止过长或过短的输入，维护数据一致性。

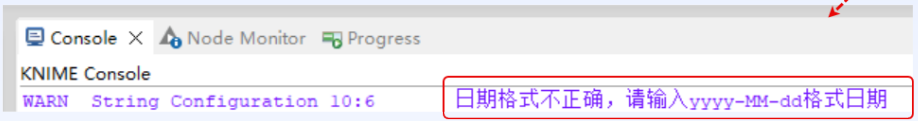
非空验证

非空验证用于确保字段值不能为空。通过设置自定义验证规则，可以强制用户必须填写必要的信息，保证数据的完整性。

数据验证—示例







验证不通过时，显示自定义错误信息



06

本章小结

回顾正则表达式核心概念

01

正则表达式定义

正则表达式是一种用于描述和匹配字符串模式的语言。它由普通字符和特殊字符组成，可以应用于各种编程语言中，KNIME 也有诸多正则相关节点，常用于复杂文本处理和数据验证等任务。

02

正则表达式语言特点

正则表达式具有高度的抽象性和精简性，使用少量的字符就能表达复杂的匹配规则。它能够处理多种文本情况，包括匹配指定字符、捕获特定子串、验证输入格式等。

03

正则捕获组概念

正则表达式中的捕获组使用圆括号()定义，用于保存匹配的子字符串。它允许我们将多个子模式组合在一起，并捕获这些子模式匹配的文本

梳理高阶数据处理技巧

字符串相似度比较

字符串相似度比较是高阶数据处理中的关键技巧，通过使用编辑距离算法和Jaccard索引等方法，可以高效地衡量不同字符串之间的相似程度。这些技术在文本匹配、信息检索等领域有广泛应用。



数据脱敏与加密技术

数据脱敏与加密技术是保护敏感数据的重要手段。常见的方法包括替换敏感字段、应用同义词替换和生成合成数据等，而加密技术则通过算法如AES和MD5确保数据在传输和存储中的安全。

数据标准化处理

数据标准化处理是确保数据一致性的重要步骤。通过标准化方法如Z-score，将不同量级的数据转换为统一的尺度，有助于提升数据分析的准确性和可靠性。

数据验证与校验

数据验证与校验是确保数据质量的重要环节。利用正则表达式、数据校验方法和统计检测手段，可以有效识别和纠正错误数据，提高数据的准确性和可用性，为后续分析打下坚实基础。

强调实践与建议方向

实践重要性

正则表达式的学习和应用需要大量的实践。通过编写和测试各种正则表达式，可以更好地理解其工作机制并提高匹配效率，从而在实际项目中发挥最大效用。

建议方向

初学者应从基础概念和简单应用开始，逐步过渡到复杂场景。需多注意KNIME中各正则节点用法的差异，部分节点需要双重转义符，如String Manipulation节点。Regex Extractor提供了更好的交互界面和实时结果。



资源利用

网上有不少网站提供在线正则表达式的测试，可以很方便的观察到正则匹配结果。或者利用AI工具辅助生成正则表达式。

问题解决策略

面对复杂的匹配问题，建议先分析目标字符串的结构，再尝试构建相应的正则表达式。反复测试和优化是找到最佳匹配方案的关键步骤。或者清晰表述问题，借助AI生成正则表达式。

感谢观看

THANKS



第十三章 报告

从流程到技巧，掌握报告制作每一环节



目录

01 创建报告流程概述

02 创建报告四步骤

03 报告七要素

04 批量创建报告方法

05 报告中使用非兼容可视化节点

06 本章总结



01

创建报告流程概述

确定报告目标与受众

确定报告目标

明确报告的主要目的和预期成果。这有助于指导报告的结构和内容，确保信息传递的有效性和针对性。

识别受众需求

分析目标受众的需求、兴趣和偏好。通过问卷调查、用户数据分析等方法，了解受众的特点和期望，以便更好地满足他们的需求。

评估受众反馈

收集并分析受众对报告的反馈，包括阅读量、下载量和分享量等行为数据。通过反馈，可以评估报告的影响力和效果，及时进行优化调整。

调整报告风格与内容

根据受众特点和需求，调整报告的语言、术语和内容风格。使用简洁明了的语言和适当的图表，可以提高报告的可理解性和接受度。

收集和整理数据

01

确定数据收集需求

在开始数据收集前，明确报告所需的关键数据和信息。这有助于指导数据收集的方向，确保收集到的数据对报告的编制有实际帮助。

02

选择数据收集方法

根据数据类型和可用资源，选择合适的数据收集方法，如问卷调查、访谈、数据库查询等。不同的方法适用于不同类型的数据，确保数据的全面性和准确性。

03

制定数据整理计划

制定详细的数据整理计划，包括数据清洗、分类、汇总等步骤。通过系统化的数据整理过程，提高数据的准确性和可使用性，为报告编写打下坚实基础。

04

使用数据管理工具

利用专业的数据管理工具，如数据库软件或数据分析平台，进行数据的存储、处理和分析。这些工具能提供强大的数据处理能力，提高工作效率。

05

审核与验证数据

在报告编制前，对收集和整理的数据进行审核和验证。检查数据的逻辑一致性、完整性和准确性，确保报告中的数据可靠且具有说服力。

设计和构建报告布局

01

页面布局设计

报告的页面布局设计包括划分不同章节和段落，使用标题和子标题来突出主要内容和结构。合理利用留白并保持一致的间距和边距，使报告视觉效果整洁，提升整体阅读体验。

02

文本与视觉元素配置

在报告中合理配置文本和视觉元素，如图表、表格和图片。确保视觉元素清晰可读，并与文本内容相辅相成，增强信息的传递效果。同时避免过度装饰，以免分散读者注意力。

03

分栏与插图应用

报告设计中应合理运用分栏和插图，以增强内容的层次感和可读性。分栏有助于区分不同模块，而插图则能直观展示复杂数据或流程，但需注意不要过多使用，以免影响整体美观。

04

颜色主题与布局优化

选择适合的颜色主题，可以有效提升报告的专业性和吸引力。颜色应与品牌形象和内容主题相匹配，并在布局上进行优化，确保视觉上的舒适度和美观度，提高报告的整体质量。

监控和更新报告频率

01 定义报告频率

确定报告的发布周期是监控和更新报告的首要任务。根据业务需求和数据变化的频率，设定定期报告、实时报告或两者结合，以确保信息及时传递。

02 使用自动化工具

利用自动化工具进行报告生成和维护，可以显著提高效率和准确性。这些工具可以自动从数据源（如数据库）中提取最新数据，确保报告内容始终与实际情况保持一致。

03 用户反馈机制

建立用户反馈机制，收集用户对报告的意见和建议。了解用户需求的变化，有助于优化报告内容和形式，提高报告的实用性和用户满意度。

04 数据安全与权限管理

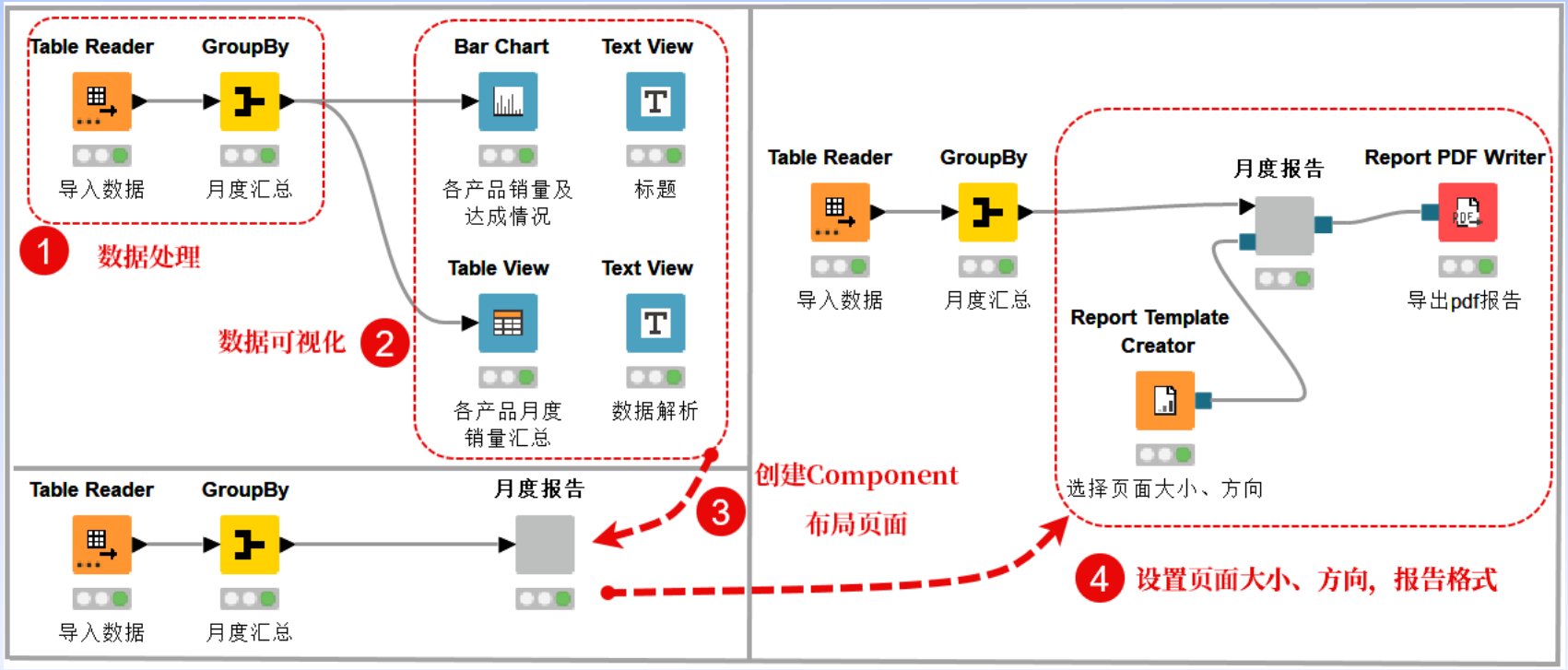
在监控和更新报告的过程中，确保数据的安全和正确分发至关重要。实施严格的权限管理策略，防止未经授权的数据访问和篡改，确保报告的准确性和可靠性。



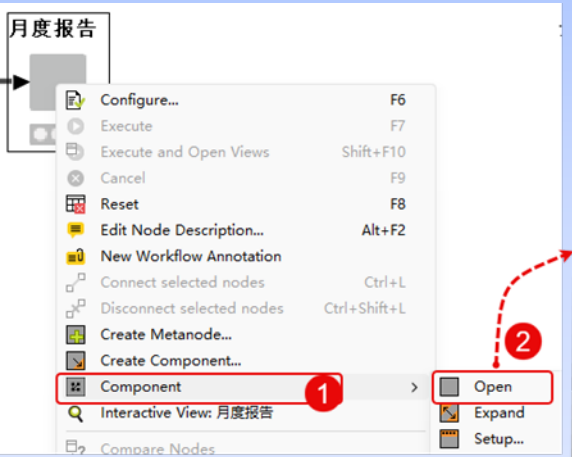
02

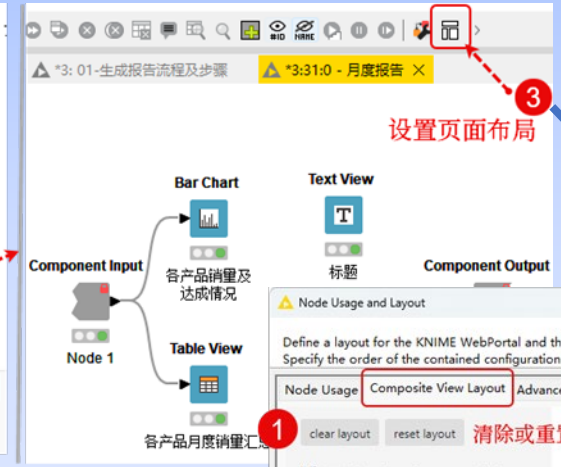
创建报告四步骤

创建报告的四个步骤


Table Reader
GroupBy
Bar Chart
Text View
导入数据
月度汇总
各产品销量及达成情况
标题
1
数据处理
Table View
Text View
数据可视化
2
各产品月度销量汇总
数据解析
Table Reader
GroupBy
月度报告
导入数据
月度汇总
3
创建Component
布局页面
Table Reader
GroupBy
月度报告
Report PDF Writer
导入数据
月度汇总
导出pdf报告
Report Template Creator
选择页面大小、方向
4
设置页面大小、方向，报告格式

创建报告（组件）及调整布局





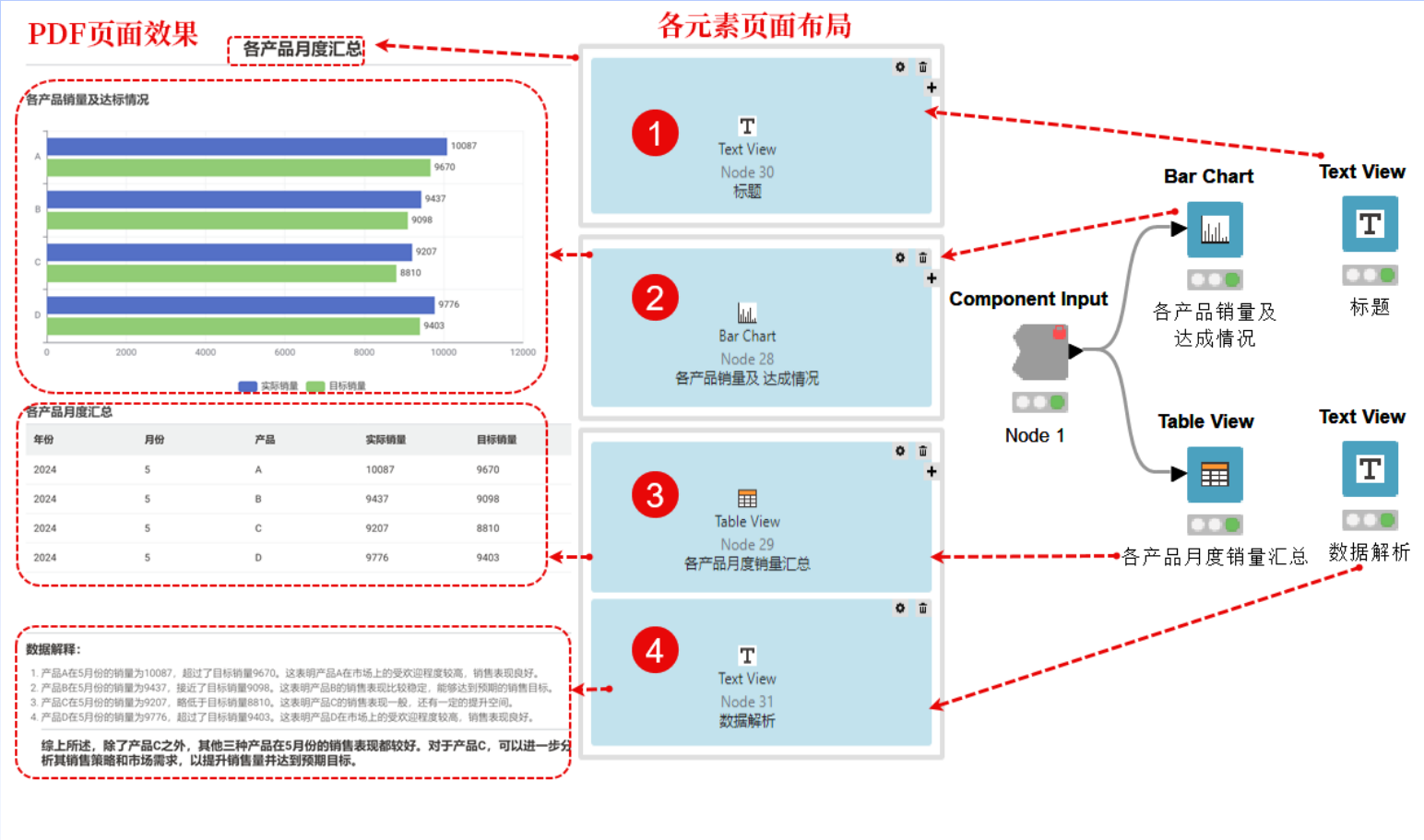
Step1：创建组件

Step:2：调整各视图类节点页面布局



务必勾选

各节点与PDF报告对应关系

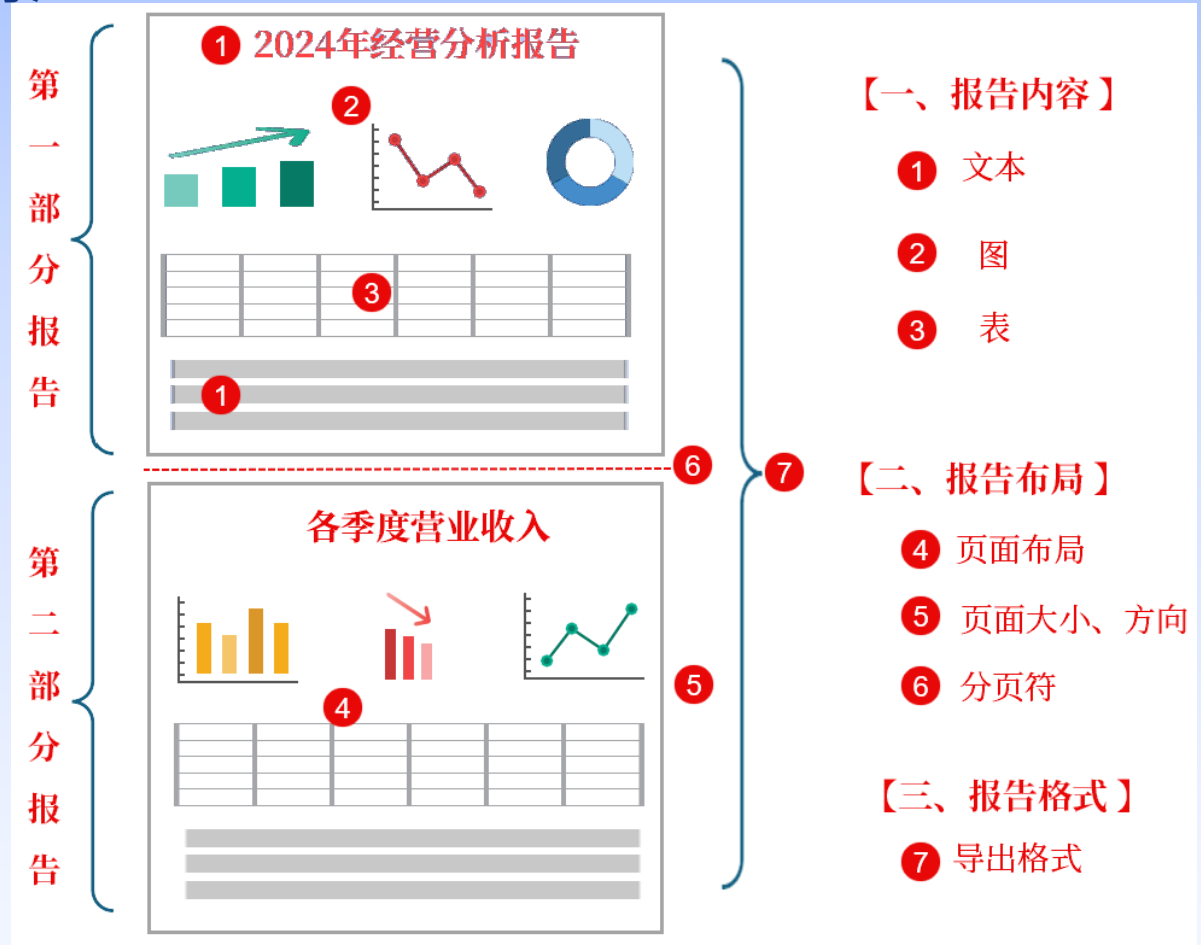




03

报告七要素

报告七元素


第一部分报告
1 2024年经营分析报告
2
3
1
6
第二部分报告
各季度营业收入
4
5
7
【一、报告内容】
1 文本
2 图
3 表
【二、报告布局】
4 页面布局
5 页面大小、方向
6 分页符
【三、报告格式】
7 导出格式

报告七元素

重点

01 文本

02 图

03 表

04 页面布局

05 页面大小与方向

06 分页符

07 报告格式

文本

01 文本选择与编辑

在报告创建过程中，文本的选择和编辑至关重要。应确保使用正式、准确的语言。同时，确保文本无语法错误和拼写错误，提升报告的专业性和可读性。**可使用流变量实现动态文本，或使用HTML语法格式显示不同颜色。**

02 内容结构与逻辑

报告的内容结构与逻辑直接影响信息的传递效果。建议采用“三段式”结构：引言、正文和结论，每个部分要明确主题，逻辑清晰，层次分明，使读者能够轻松理解和跟随报告的思路。



图

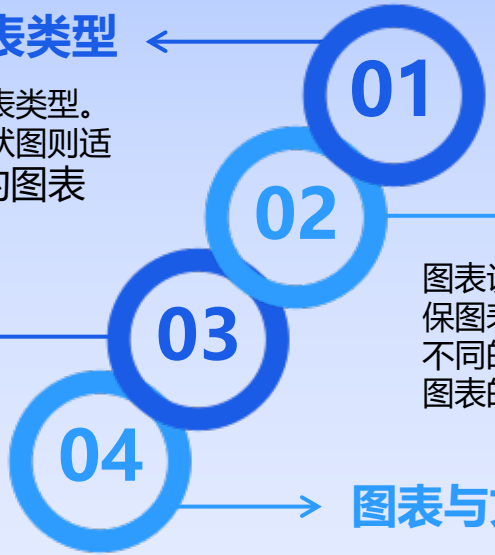
选择合适图表类型

根据数据类型和展示需求，选择合适的图表类型。例如，折线图适用于展示趋势和变化，柱状图则适合对比不同项目的数据大小。选择正确的图表类型有助于清晰、准确地传达信息。

可视化图形节点的选择

1、报告只支持KNIME视图类节点（Views根目录下各节点，第8章分类）

2、其他非兼容类可视化节点，通过导出图片至Image View，这样也可以插入到报告中。（详见本章示例）



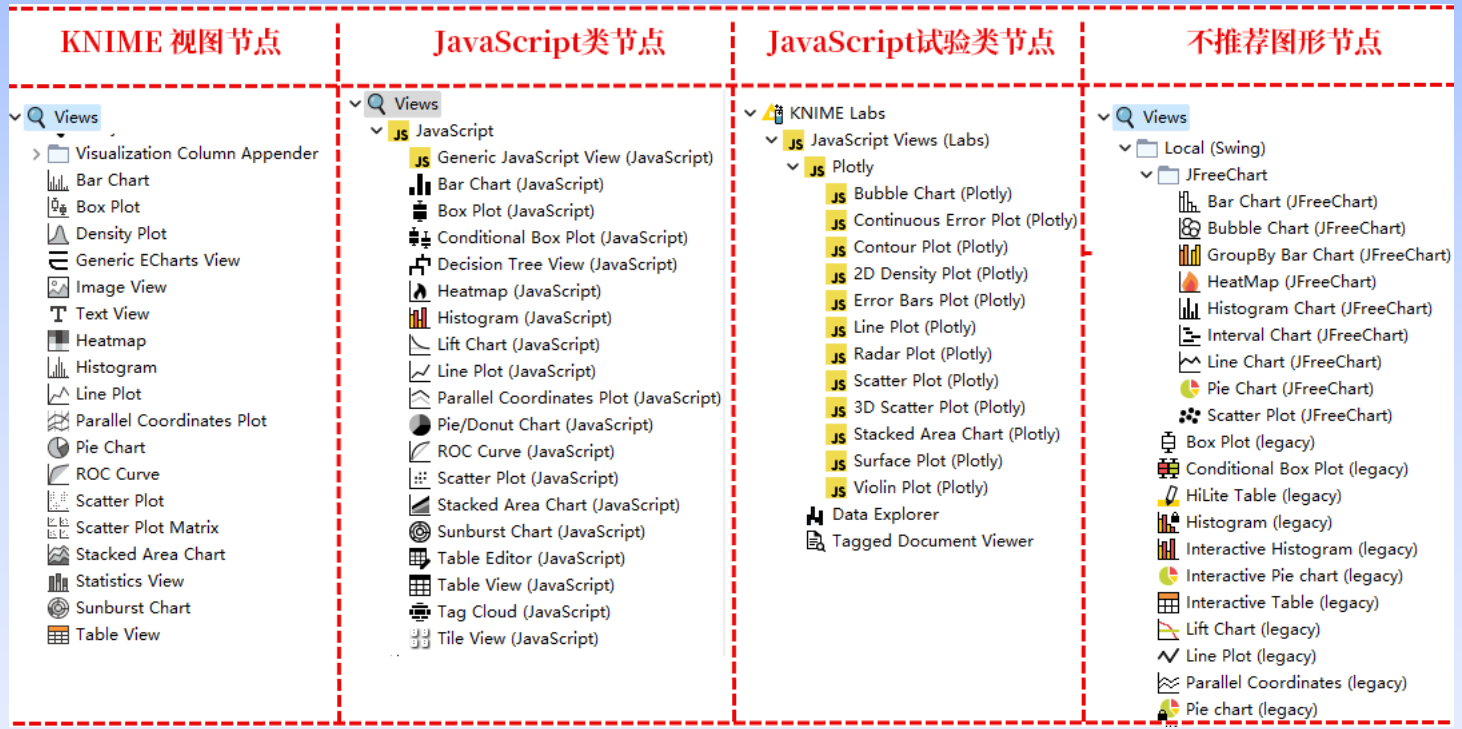
图表设计原则

图表设计应简洁明了，避免过于复杂的布局。确保图表标题明确、坐标轴标签清晰可读，并使用不同的颜色或图案区分不同的数据系列，以增强图表的可读性和理解度。

图表与文本结合

在报告中合理搭配图表和文本，有助于全面解读数据。图表应与相关解释性文本紧密配合，提供数据背景和分析结论，使观众能够更好地理解和消化所展示的信息。

图



报告只支持KNIME视图类节点

其他非兼容类可视化节点，通过导出图片方式也可以插入到报告中。

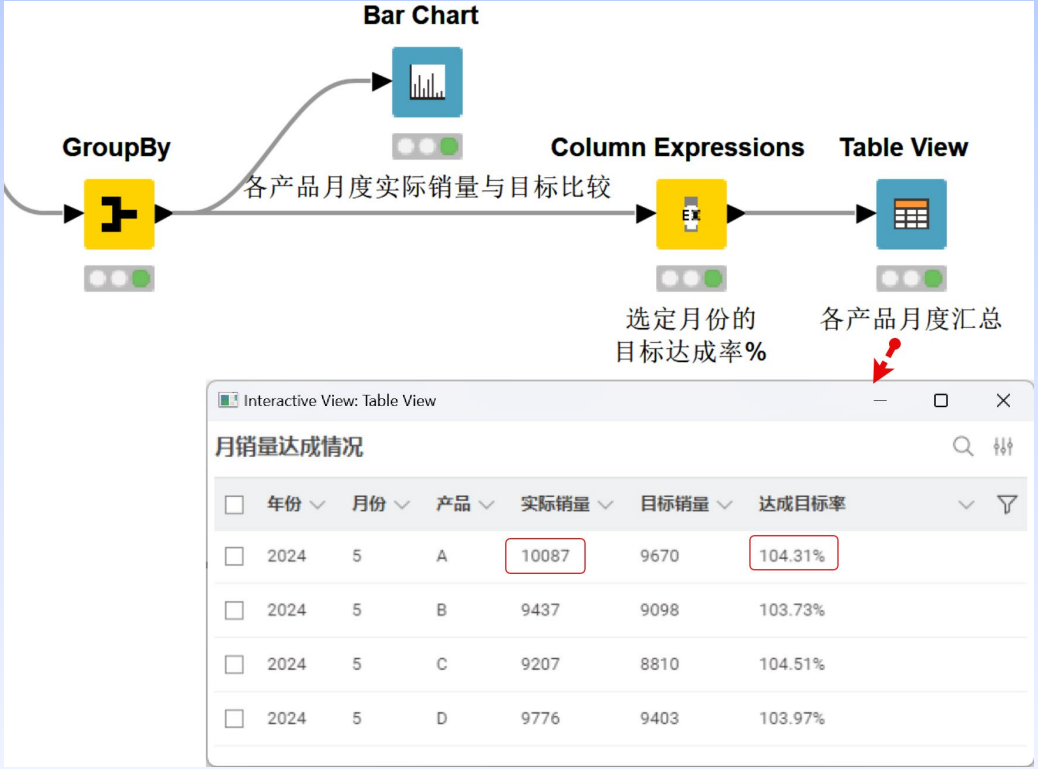
表

插入表格

在报告中，使用表格可以清晰地展示数据和信息。选择适当的表格类型（如三线表），确保其结构简洁明了，有助于读者快速理解和记忆所展示的数据。

图表位置安排

图表应放在恰当的位置，与相关文本内容紧密关联，避免重复和无关的信息。每个图表应有清晰的标题和注释，确保读者能够理解图表所表达的信息。

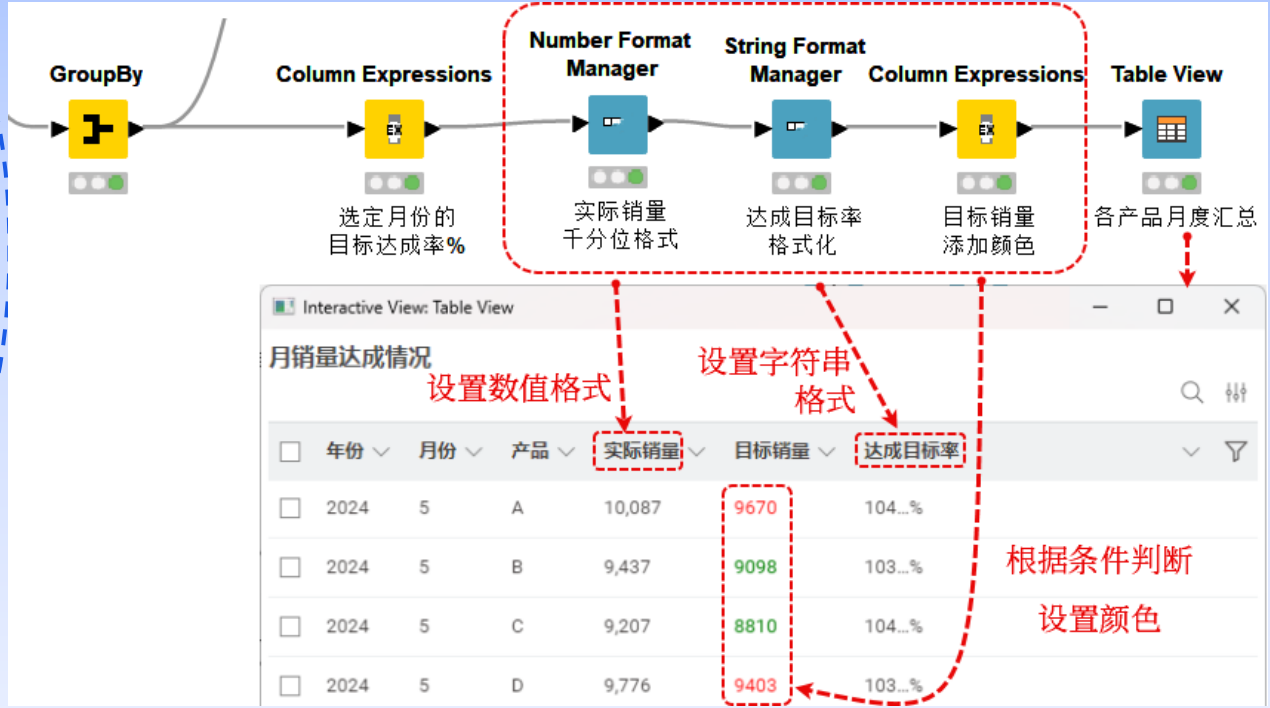


年份	月份	产品	实际销量	目标销量	达成目标率
2024	5	A	10087	9670	104.31%
2024	5	B	9437	9098	103.73%
2024	5	C	9207	8810	104.51%
2024	5	D	9776	9403	103.97%

表

设置表格显示内容

在报告中，可使用HTML标签对格式语法实现更多的格式、颜色、背景等设置，详见书中各个示例



页面布局

页面布局类型

报告的页面布局可分为横向和纵向两种基本类型。横向布局适用于宽度较大的内容展示，而纵向布局则适合高度较高的信息整合，根据实际内容选择合适的布局类型有助于提升报告的易读性和美观度。

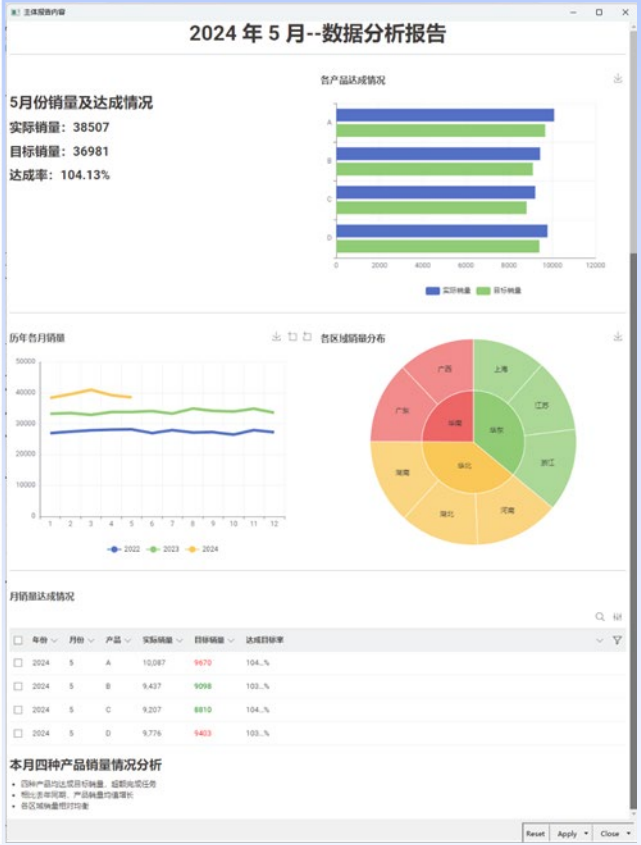
图表与文字布局

报告中图表和文字的布局需要精心规划。图表应与相关文字说明紧密结合，确保读者能够容易理解数据背后的含义。推荐采用左右或上下布局，将图表和文字分区展示，以提高报告的整体阅读体验。

设置页面布局



报告显示效果



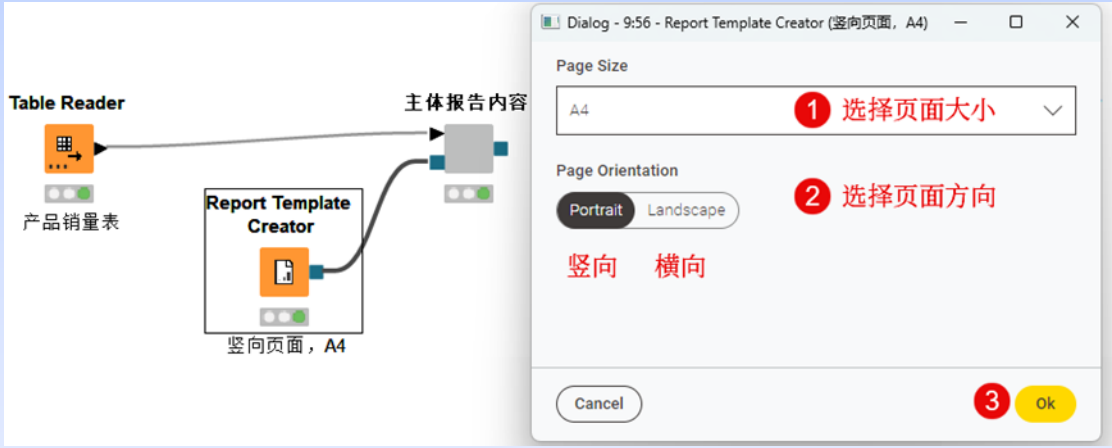
页面大小、方向

页面大小设置

页面大小的选择应根据报告内容和用途来确定。常见的页面大小有A4、A3及信纸等，其中A4是标准尺寸，适用于多数报告。较大的页面可以显示更多的内容，适合宣传海报和大幅面文件；而较小的页面则节省纸张和打印时间。

页面方向设定

页面的方向可以选择纵向或横向。一般来说，报告和书籍多采用纵向布局，以便于阅读和装订；而表格和图表更适合使用横向布局，有助于清晰展示数据和信息。



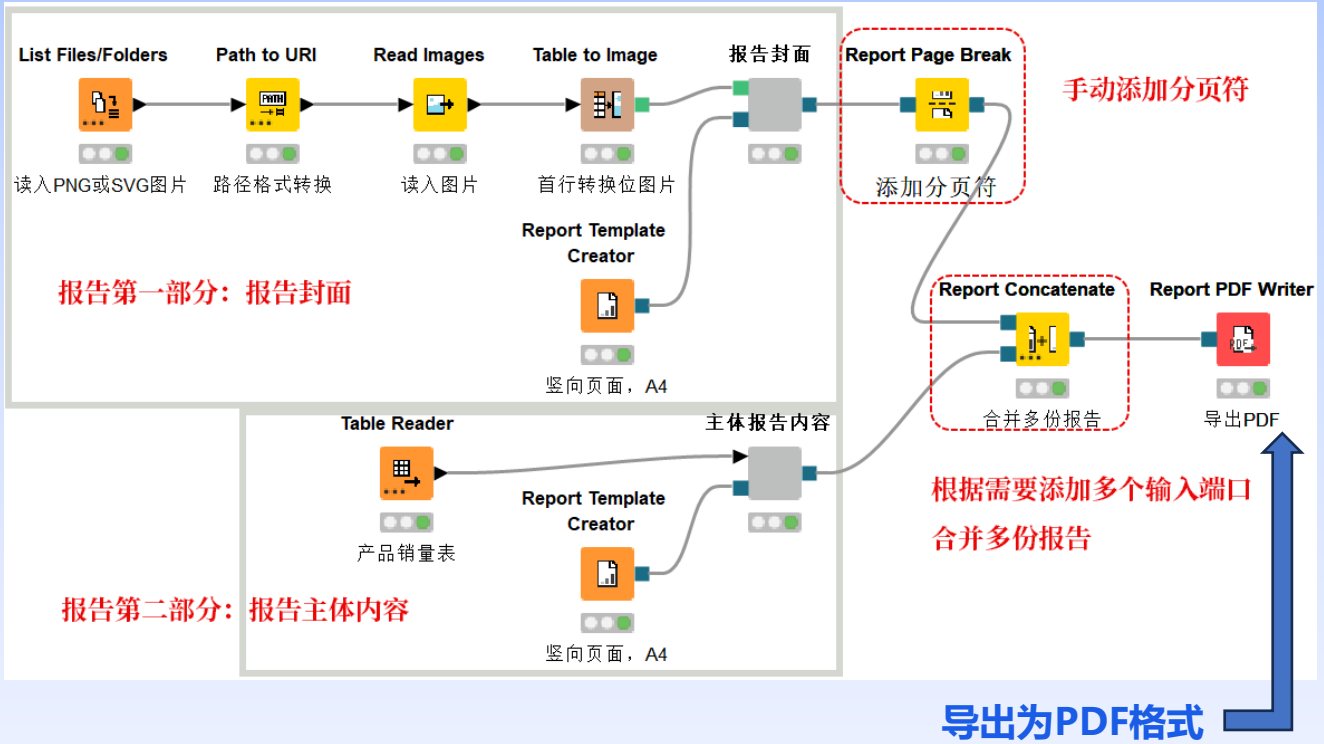
分页符、报告格式

插入分页符方法

- 类似于在Word文档中插入分页符，分隔页面，确保下一部分内容不会出现在本节中。
- 有些节点界面带有分页符选项，无需Report Page Break节点，如Report Concatenate节点。

报告格式

- 报告输出格式推荐使用 Report PDF Writer节点创建PDF格式报告。
- 也可选择是Report HTML Writer节点创建一个HTML文件

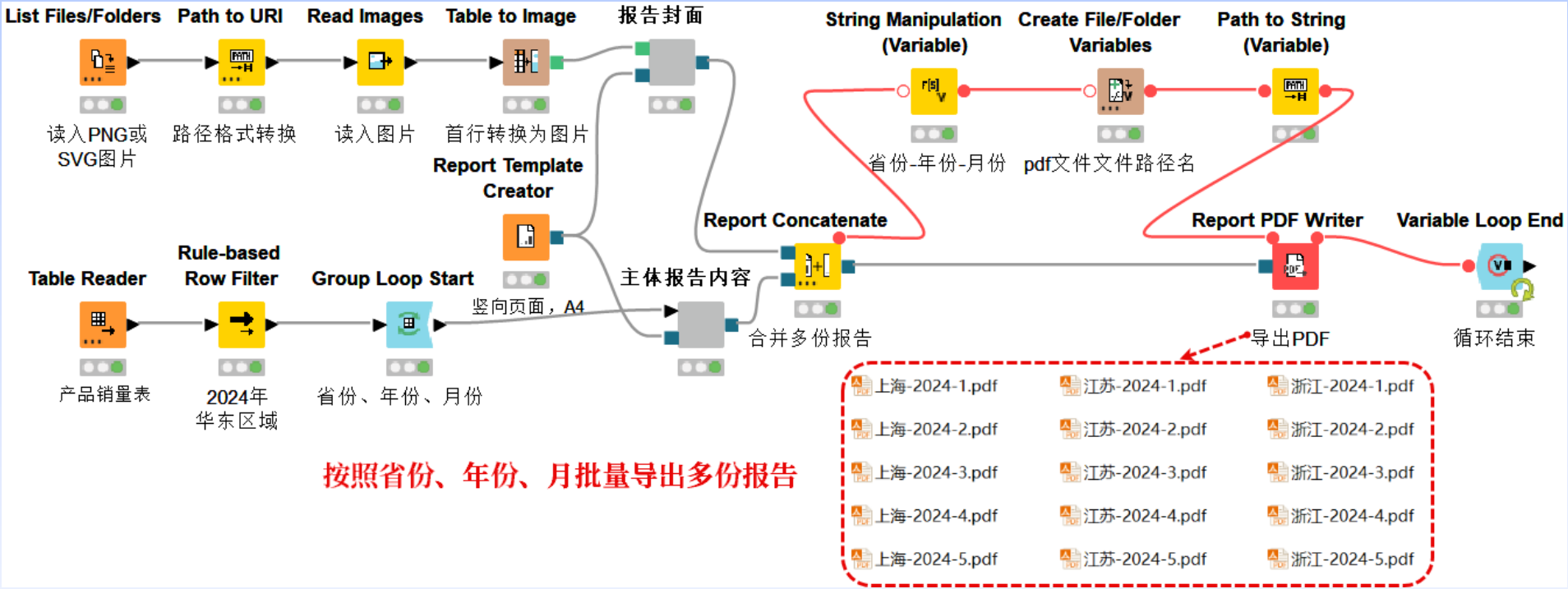




04

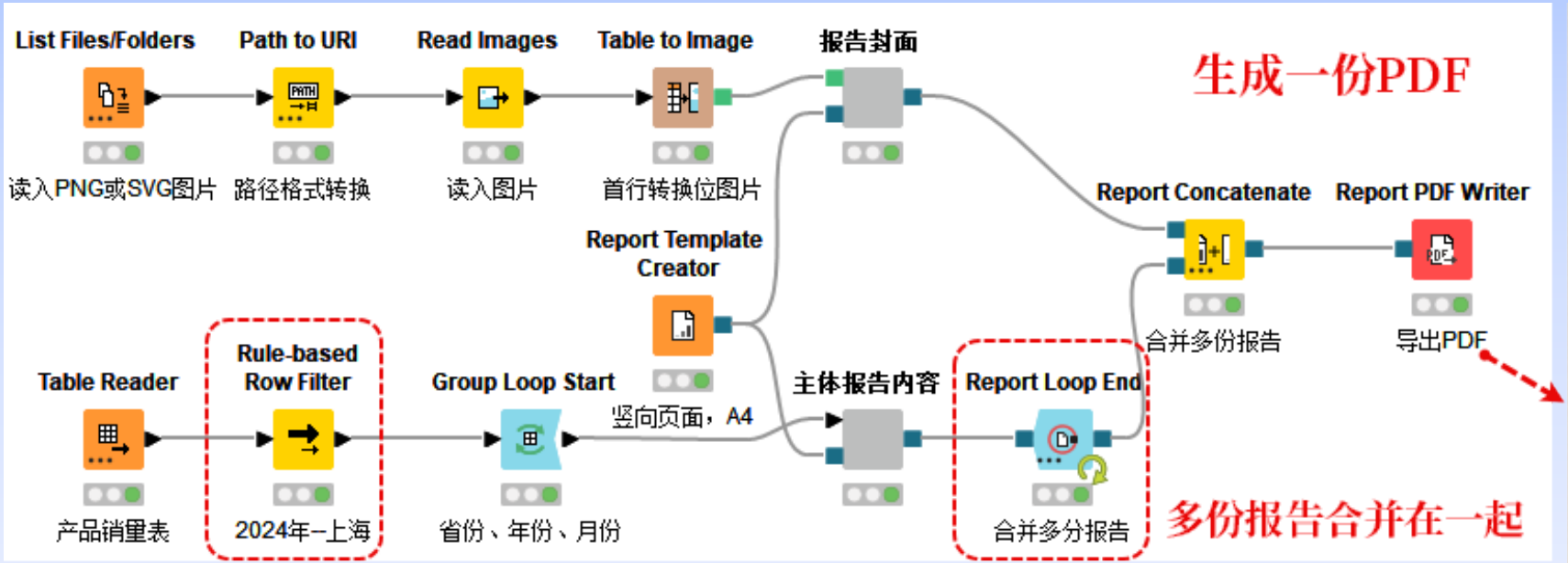
批量创建报告方法

创建多份独立报告



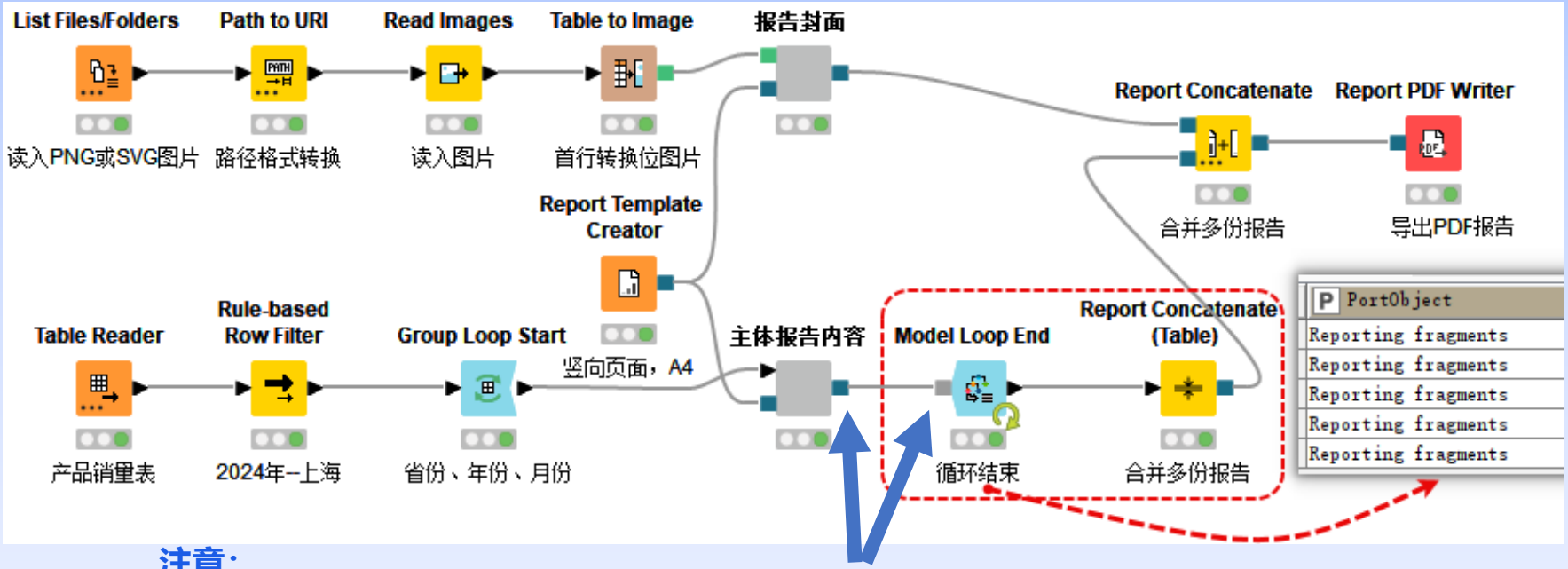
按照省份、年份、月批量导出多份报告

创建一份多个内容报告（方法1）





创建一份多个内容报告（方法2）



注意：

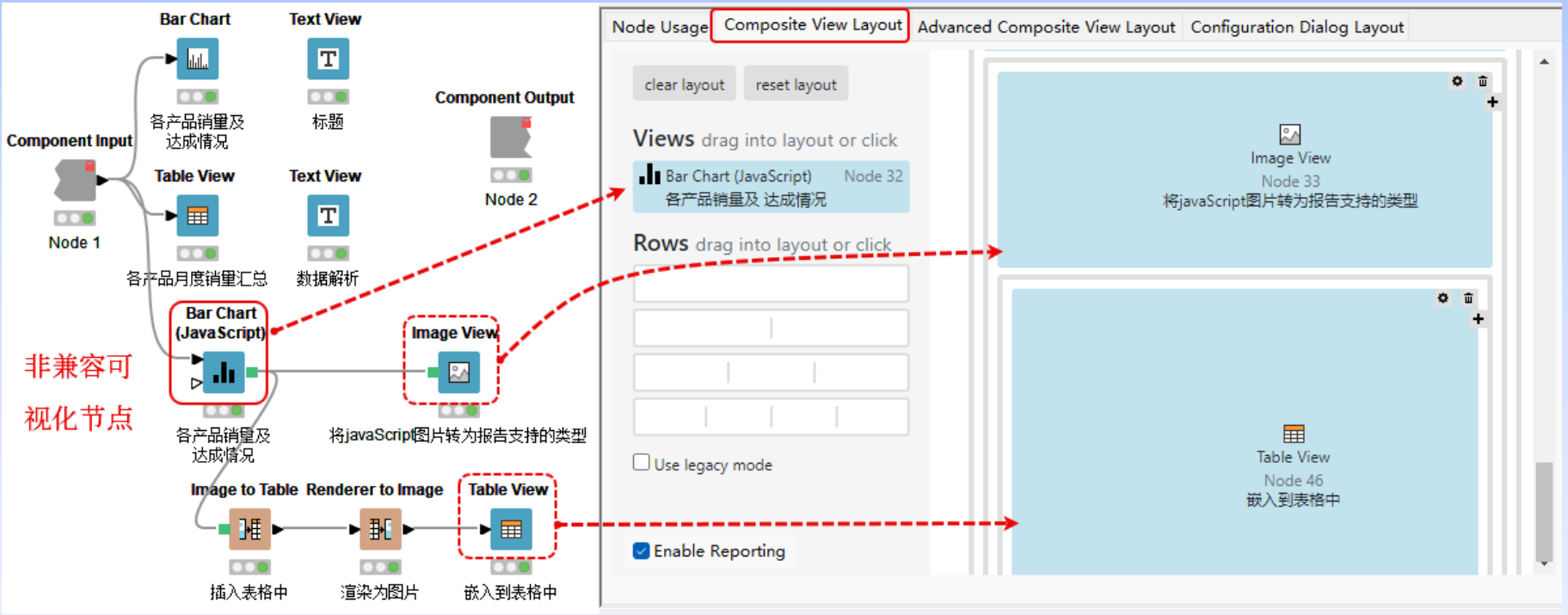
“主体报告内容”节点的**输出端口颜色**与“Model Loop End”节点**输入端口颜色并不一样**，两者可以连接，意味着后者可兼容前者输出的对象，这属于第二章提到的节点端口连接的特殊情况，绝大部分节点之间连接要求端口类似及颜色需一致。



05

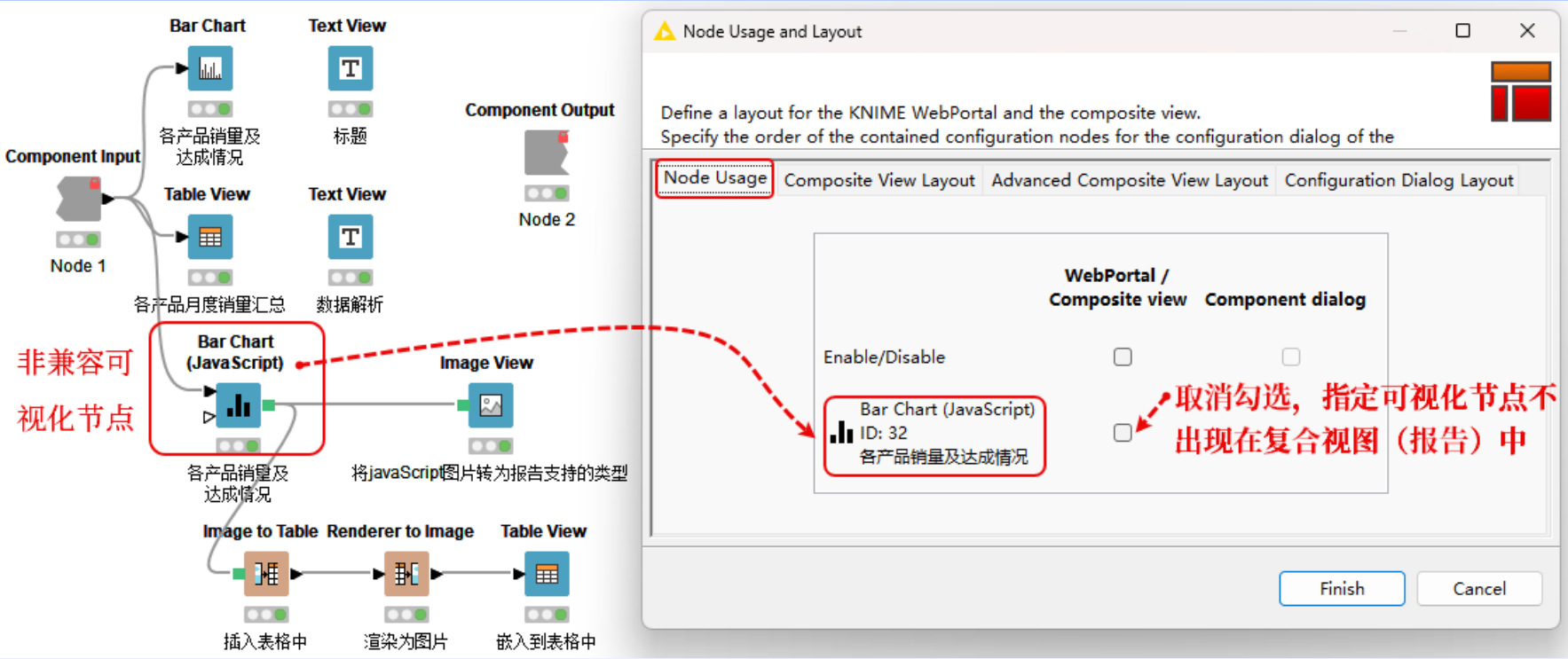
报告中使用非兼容可视化节点

报告中使用非兼容可视化节点



- 新版报告仅支持节点库View根目录下各可视化节点，也即是第八章图形节点分类中的KNIME视图类节点，不支持JavaScript类和Plotly类节点。
- 将JavaScript类和Plotly类节点的图像导出至Image View节点，可间接的使用该类节点。

报告中使用非兼容可视化节点---页面布局设置



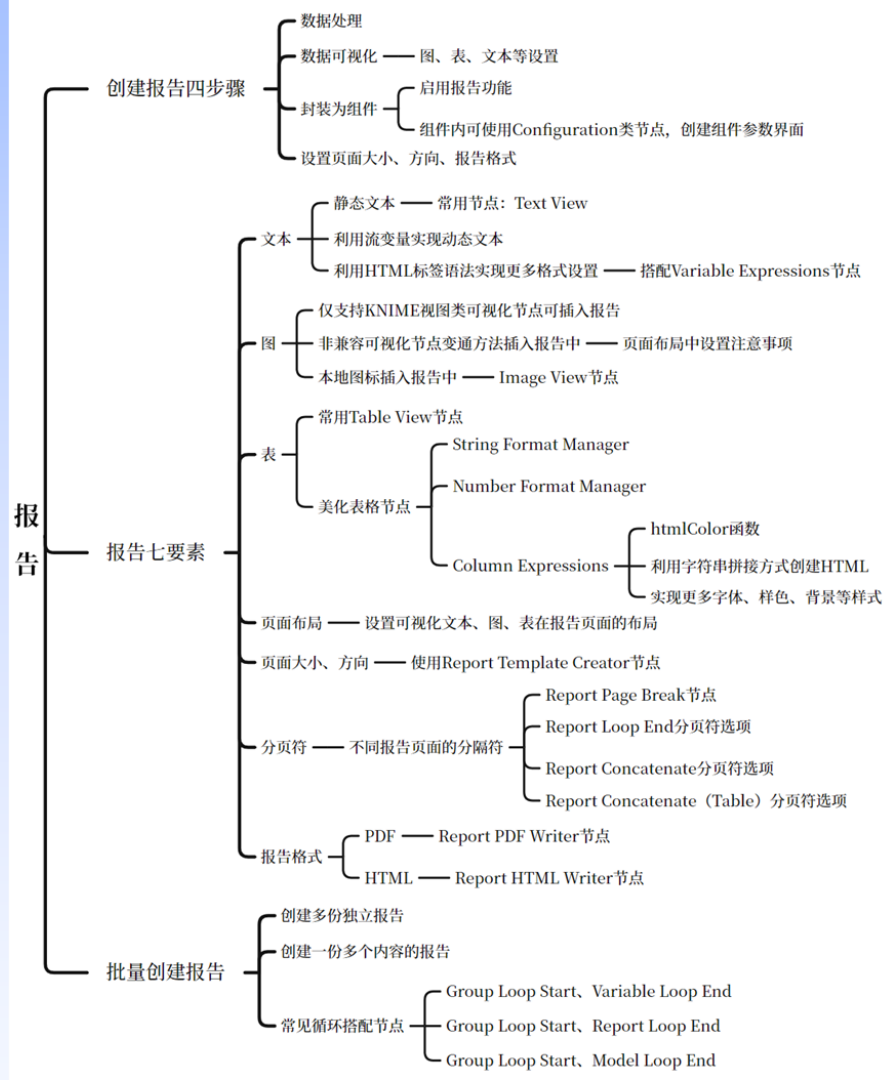
- Bar Chart（JavaScript）节点不能使用在页面布局中（因为不兼容），在页面布局时，留置在组件页面布局Views侧而不用，注意图中已经Image View节点已经实现了该柱状图的展示。



06

总结与建议

本章总结


报告
创建报告四步骤
数据处理
数据可视化 —— 图、表、文本等设置
封装为组件
启用报告功能
组件内可使用Configuration类节点，创建组件参数界面
设置页面大小、方向、报告格式
报告七要素
文本
静态文本 —— 常用节点：Text View
利用流变量实现动态文本
利用HTML标签语法实现更多格式设置 —— 搭配Variable Expressions节点
图
仅支持KNIME视图类可视化节点可插入报告
非兼容可视化节点变通方法插入报告中 —— 页面布局中设置注意事项
本地图标插入报告中 —— Image View节点
表
常用Table View节点
美化表格节点
String Format Manager
Number Format Manager
Column Expressions
htmlColor函数
利用字符串拼接方式创建HTML
实现更多字体、样色、背景等样式
页面布局 —— 设置可视化文本、图、表在报告页面的布局
页面大小、方向 —— 使用Report Template Creator节点
分页符 —— 不同报告页面的分隔符
Report Page Break节点
Report Loop End分页符选项
Report Concatenate分页符选项
Report Concatenate（Table）分页符选项
报告格式
PDF —— Report PDF Writer节点
HTML —— Report HTML Writer节点
批量创建报告
创建多份独立报告
创建一份多个内容的报告
常见循环搭配节点
Group Loop Start、Variable Loop End
Group Loop Start、Report Loop End
Group Loop Start、Model Loop End

感谢观看

THANKS



第十四章
动态数据应用：
交互式仪表盘设计

掌握创建步骤与控件配置技巧



目录

01 创建交互式仪表盘步骤

02 Component参数及视图布局

03 特殊交互式控件

04 Widget类和Configuration类节点比较

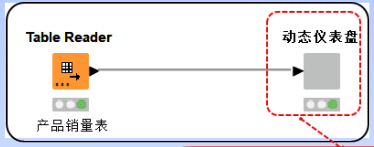
05 本章小结



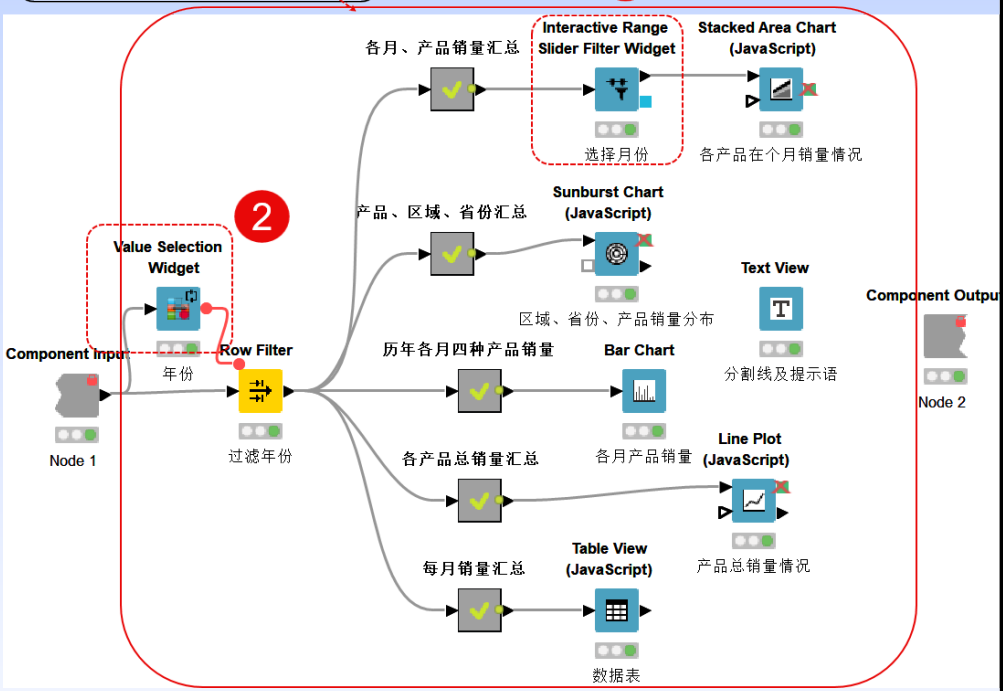
01

创建交互式仪表盘步骤

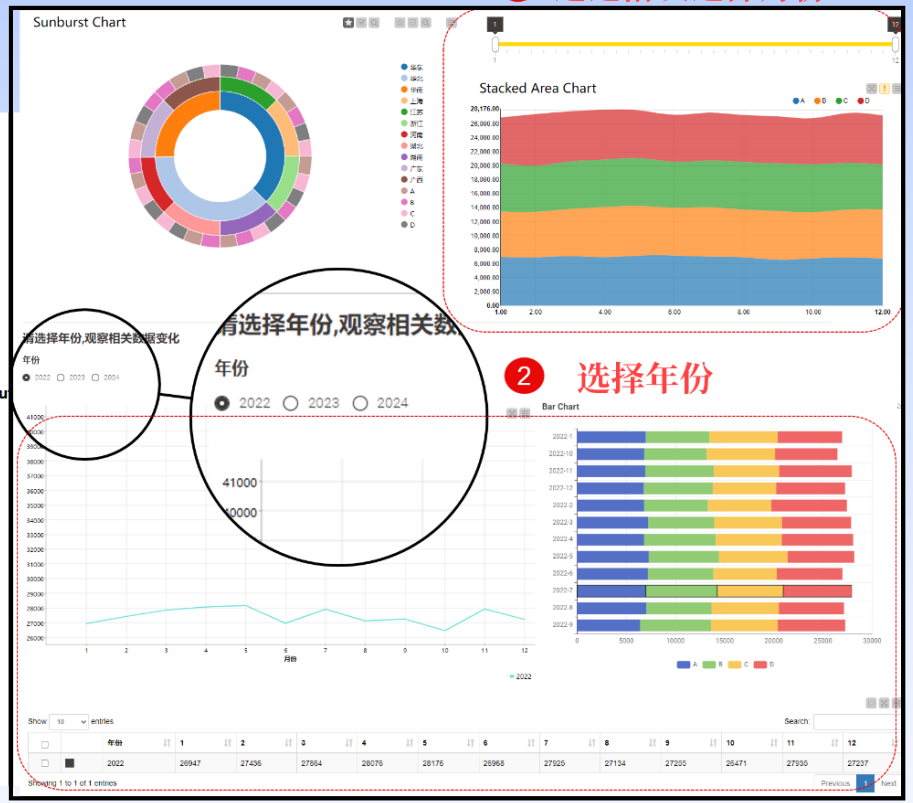
创建动态仪表盘步骤--与创建报告过程类似



动态仪表盘工作流



动态仪表盘效果图

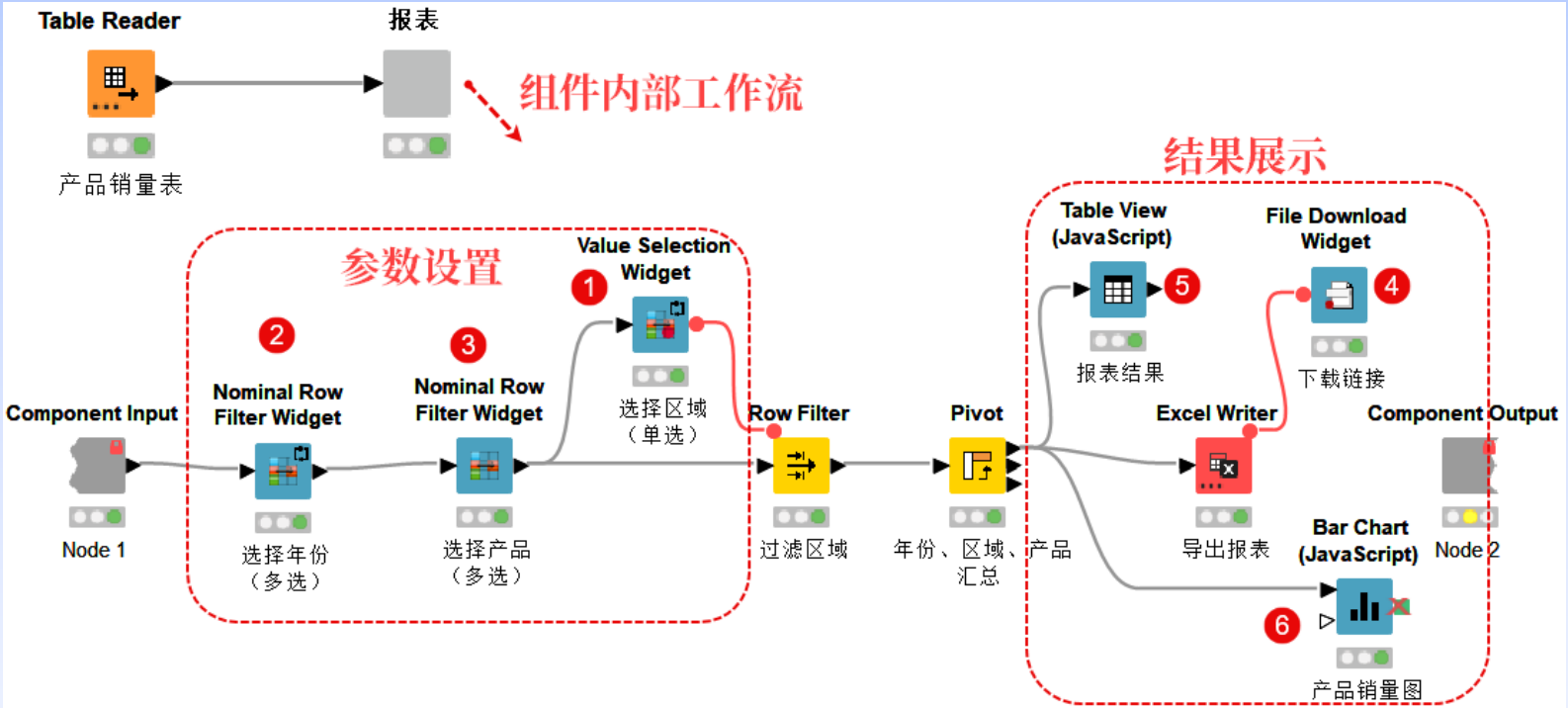




02

Component参数及视图布局

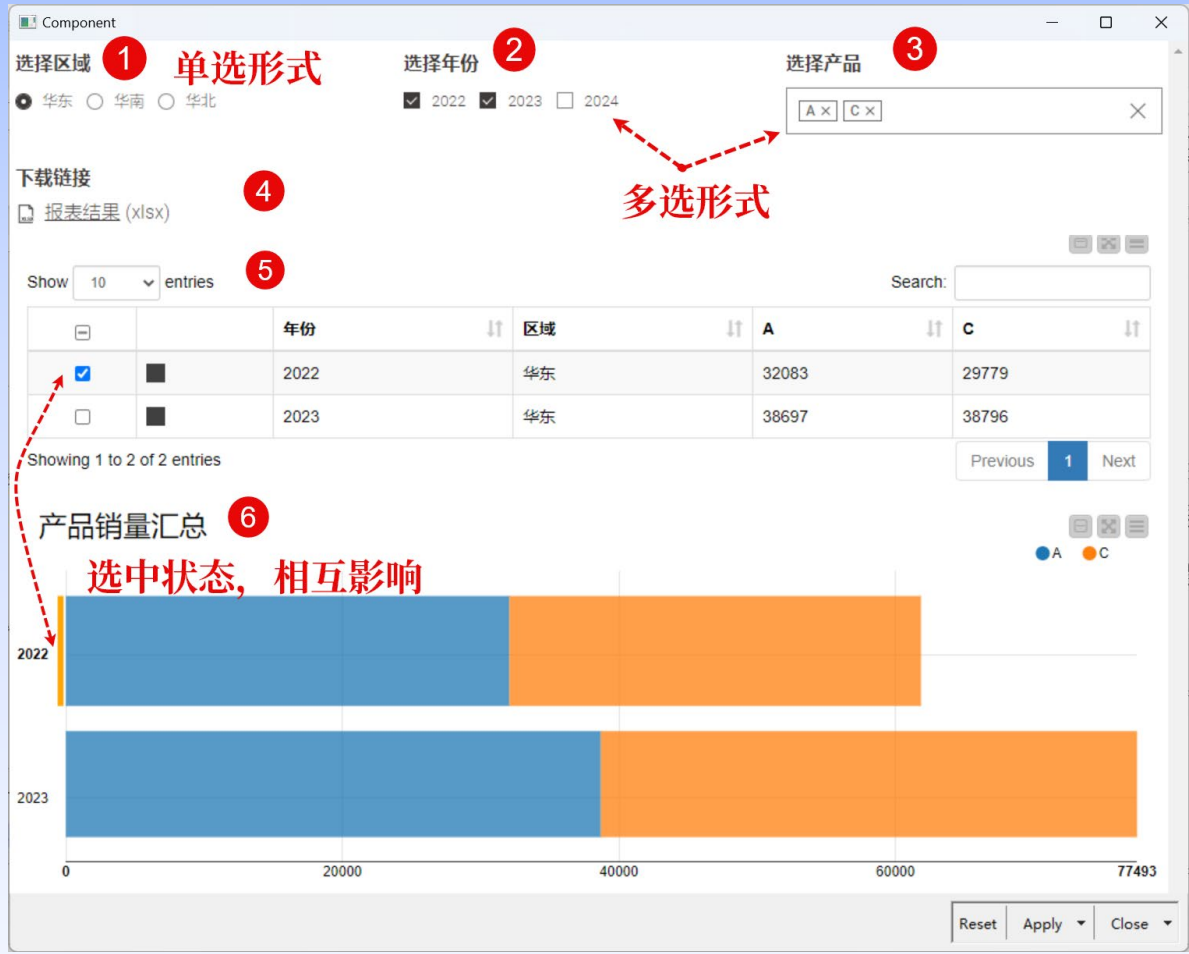
一、相互独立参数设置



各节点参数按照序号分别对应组件界面内各参数，见下一页

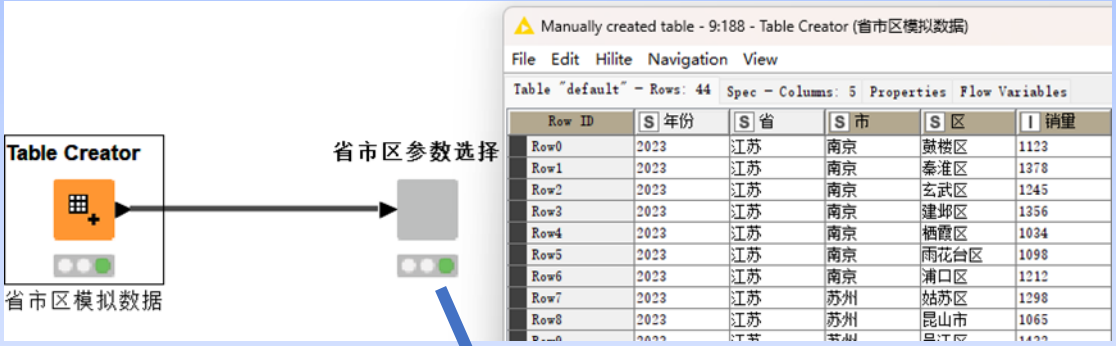
一、相互独立参数设置

参数：区域、年份、产品相互独立



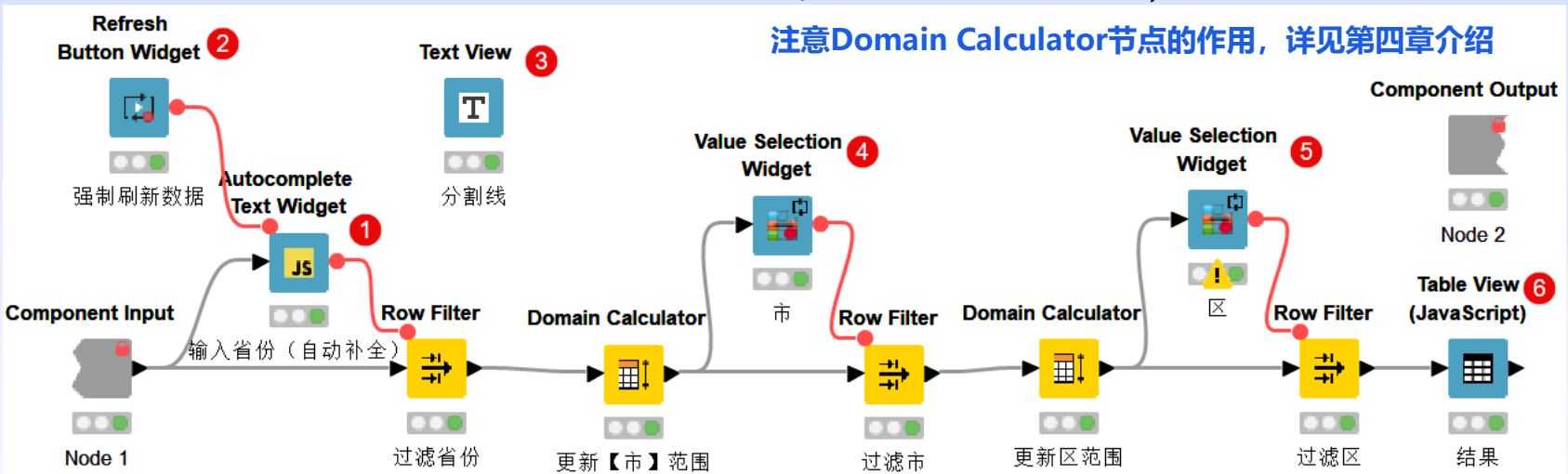
二、隶属层级参数设置

参数：省、市、区存在隶属层级关系

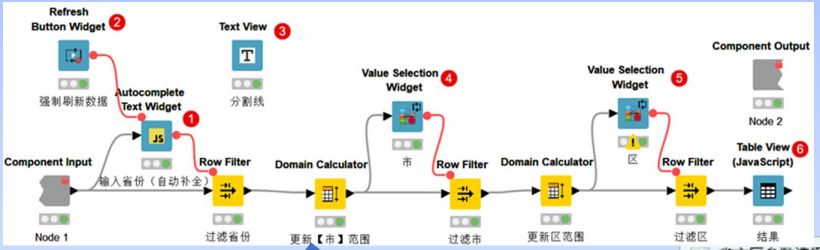


Row ID	年份	省	市	区	销量
Row0	2023	江苏	南京	鼓楼区	1123
Row1	2023	江苏	南京	秦淮区	1378
Row2	2023	江苏	南京	玄武区	1245
Row3	2023	江苏	南京	建邺区	1356
Row4	2023	江苏	南京	栖霞区	1034
Row5	2023	江苏	南京	雨花台区	1098
Row6	2023	江苏	南京	浦口区	1212
Row7	2023	江苏	苏州	姑苏区	1298
Row8	2023	江苏	苏州	昆山市	1065

5.4版本重构Row Filter节点界面，若在Column domains中选择“Compute”选项，效果等同本示例中使用Domain Calculator节点重新计算过滤后数据表的值域。若使用新版Row Filter节点，可不用使用Domain Calculator节点。



二、隶属层级参数设置

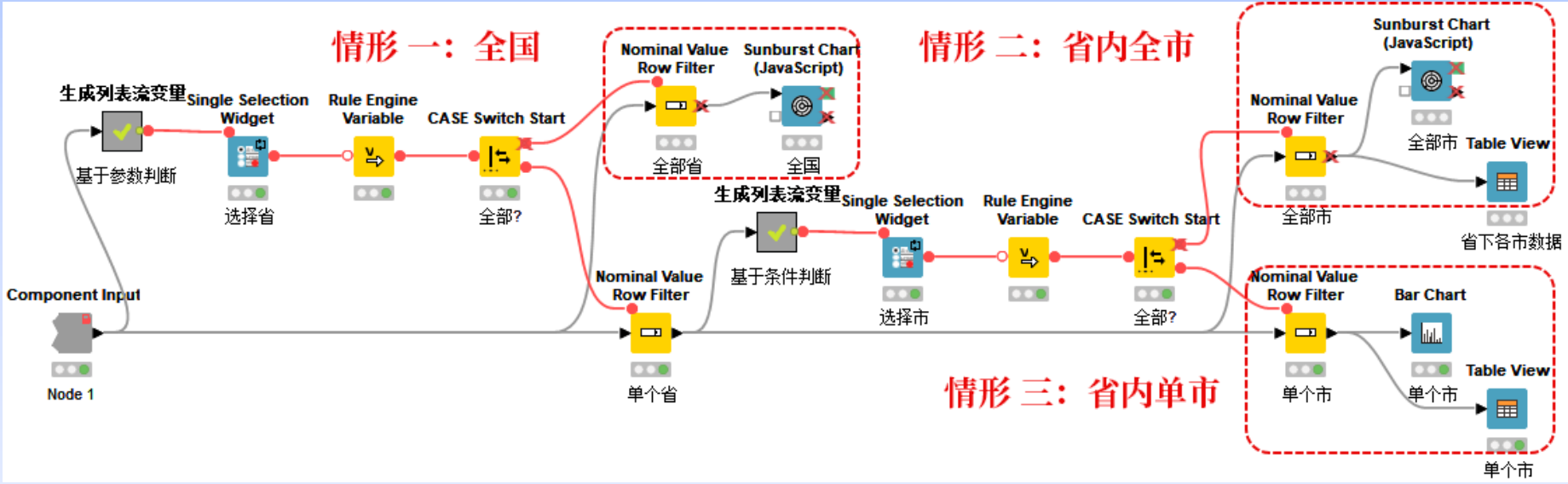


参数：省、市、区存在隶属层级关系



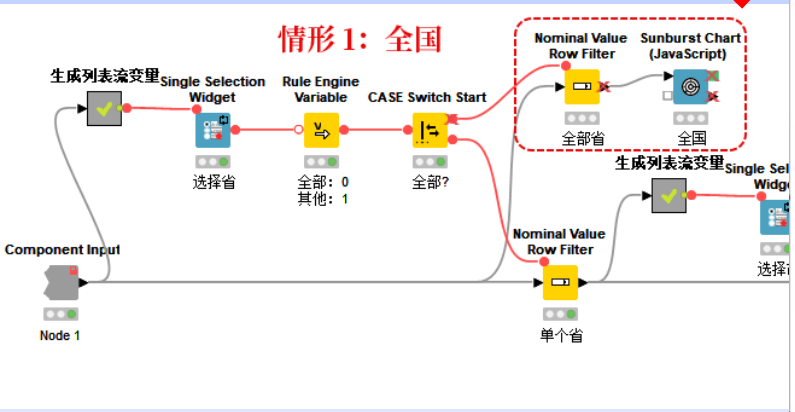
在交互式仪表盘设计中，首先需要明确各参数之间的层级关系。通过设置上级参数和下级参数的关联，可以确保数据在不同层级间正确传递，实现动态更新和交互功能。

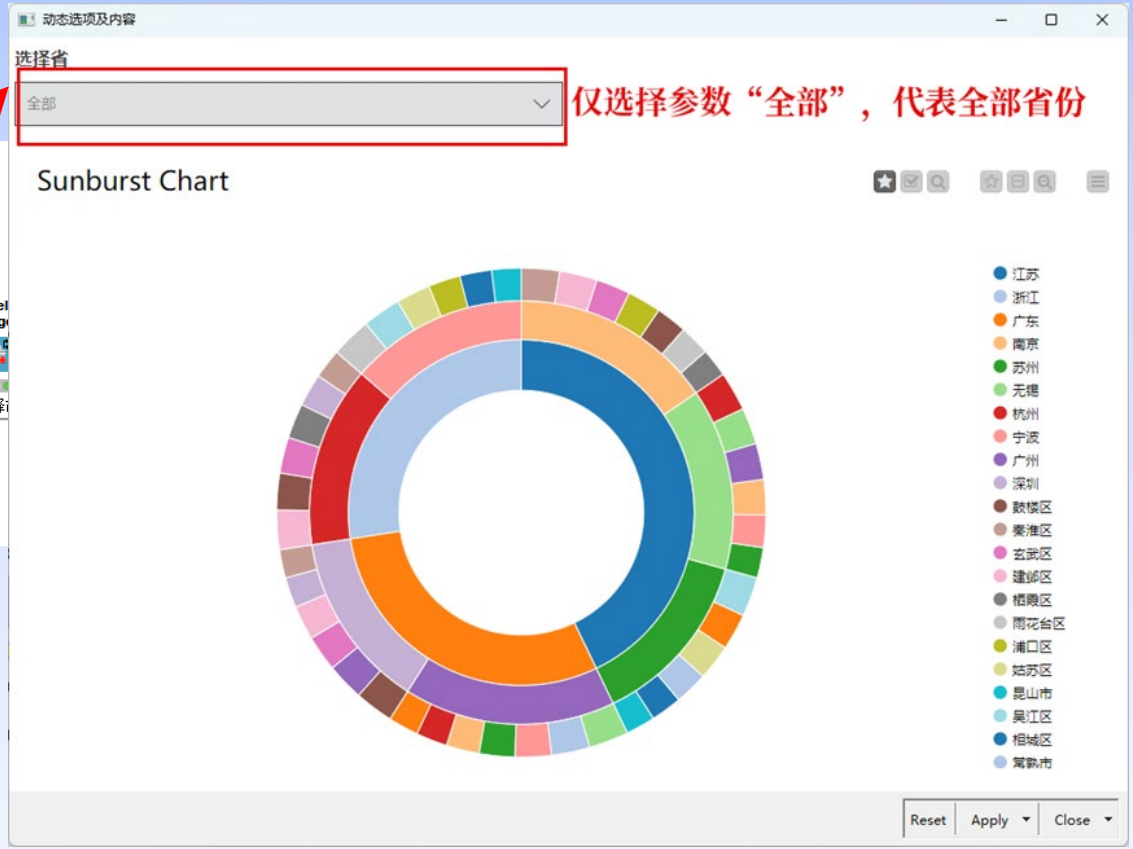
三、应用分支设置动态参数及结果（特殊视图布局）



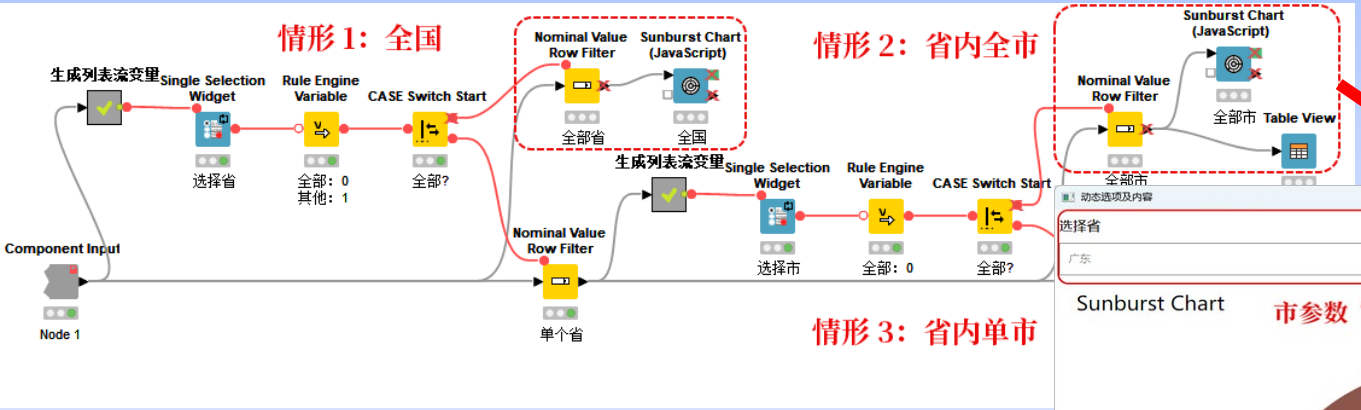
组件内部工作流

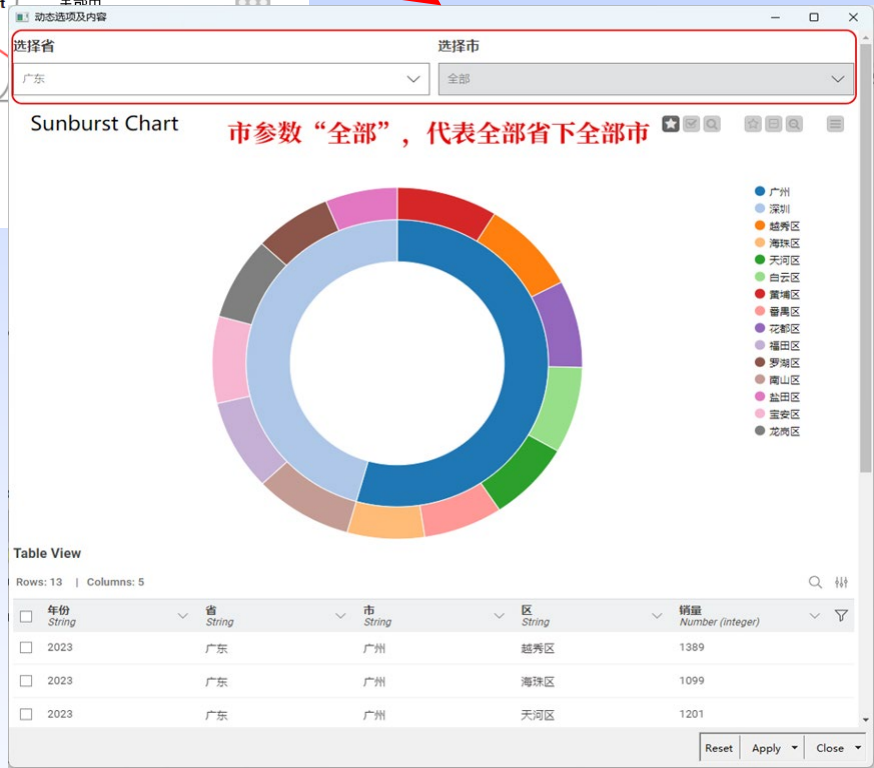
三、应用分支设置动态参数及结果（特殊视图布局）



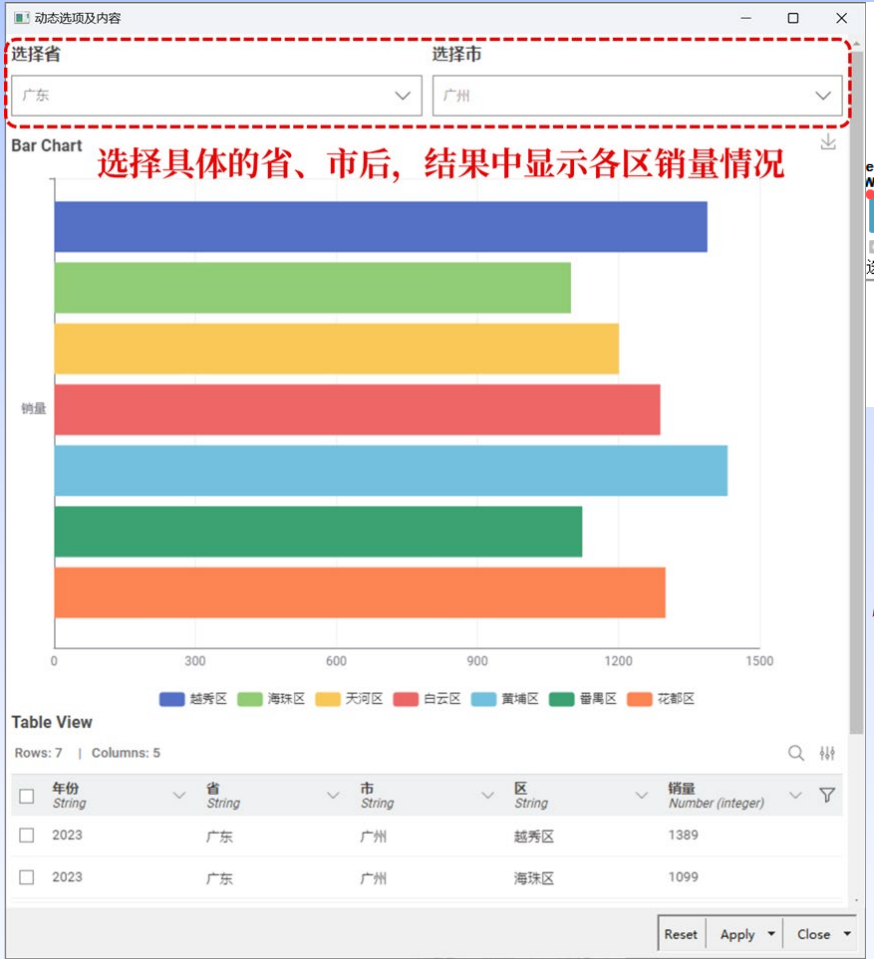


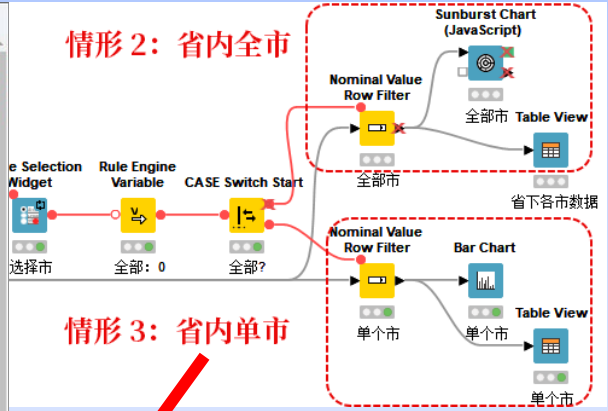
三、应用分支设置动态参数及结果（特殊视图布局）





三、应用分支设置动态参数及结果（特殊视图布局）





三、应用分支设置动态参数及结果（特殊视图布局）

- 共有三个分支，选择的不同的参数，则执行不同的分支，因此显示各分支对应的内容。
- 各分支的可视化节点应放置在同一个单元格中（在组件的Composite View Layout标签页设置布局）

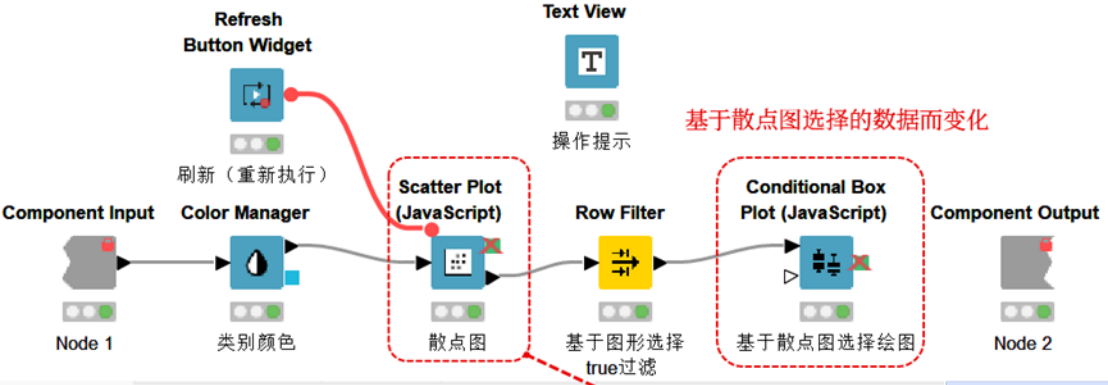




03

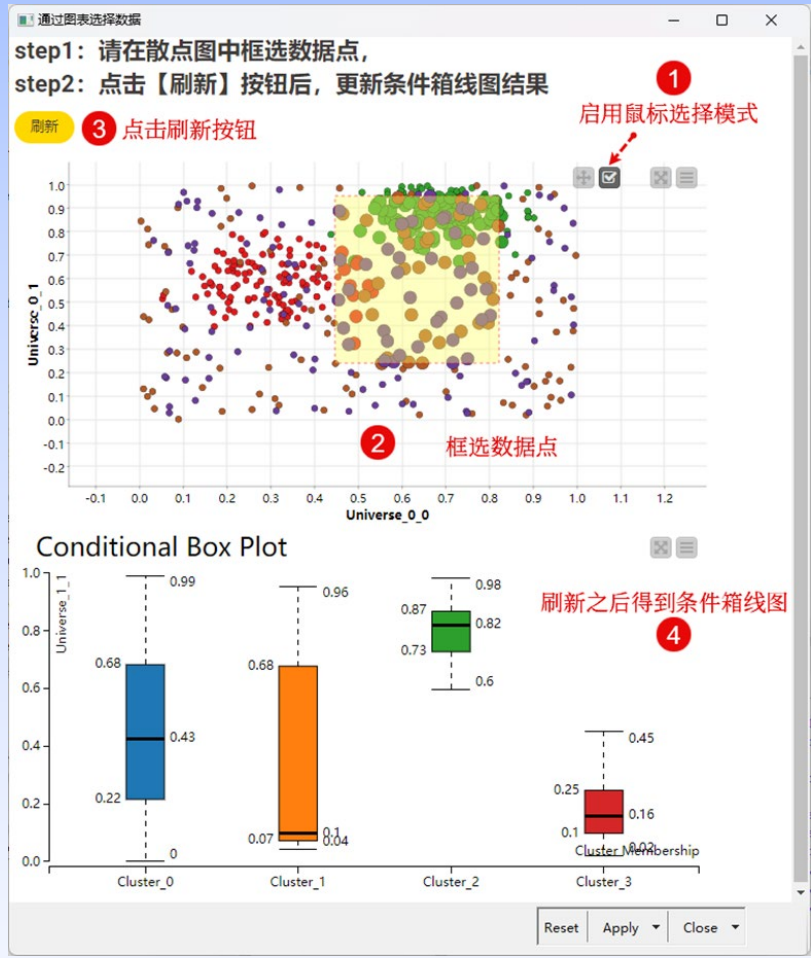
特殊交互式控件

图中选择数据点



D Unive...	D Unive...	D Unive...	D Unive...	S Clust...	B Selected (Scatter Plot)
0.814	0.929	0.908	0.149	Cluster_0	true
0.658	0.994	0.575	0.396	Cluster_0	false
0.632	0.826	0.26	0.462	Cluster_0	true
0.649	0.751	0.614	0.033	Cluster_0	true
0.793	0.834	0.676	0.629	Cluster_0	true
0.859	0.702	0.189	0.027	Cluster_0	false
0.809	0.898	0.594	0.228	Cluster_0	true

散点图中被选中数据标记为true



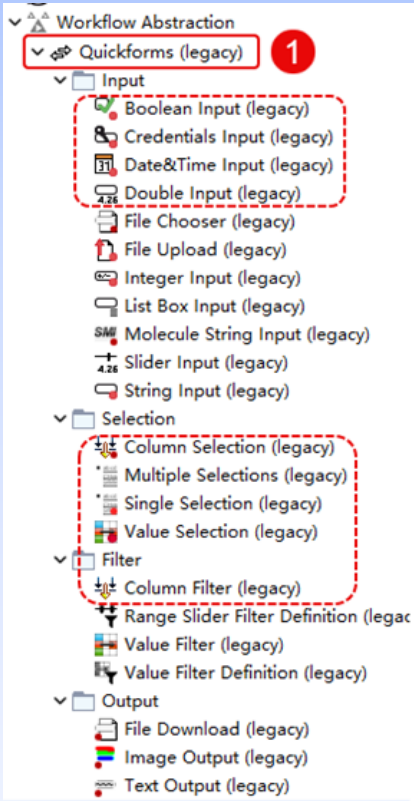
在散点图中选中的数据，将影响下游条件箱线图的结果。

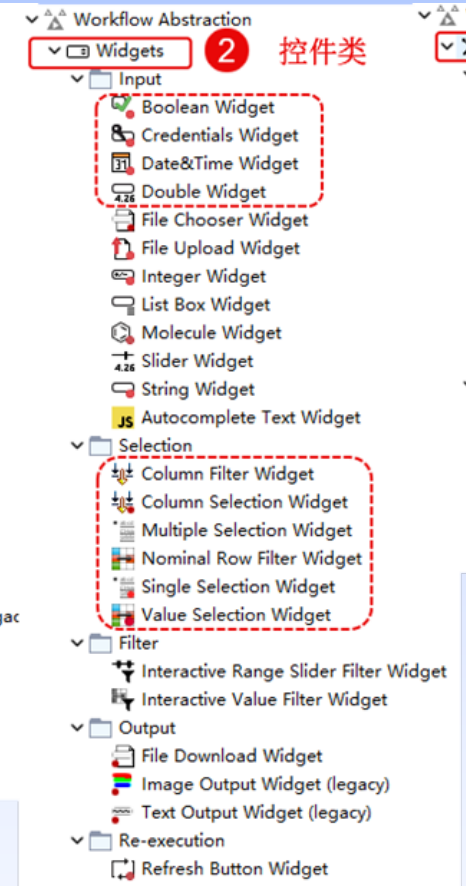


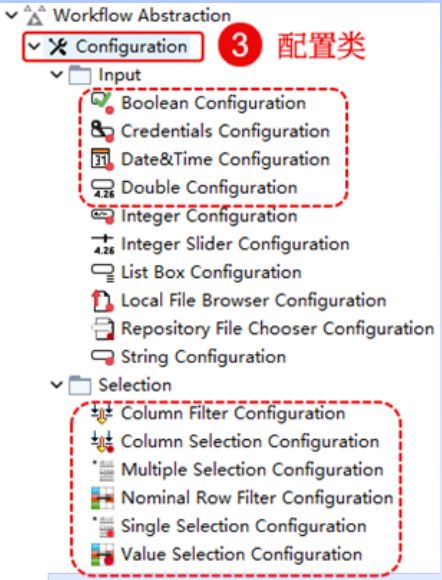
04

Widget类和Configuration类节点比较

Widget类与Configuration类节点







Widget节点与Configuration节点比较

历史演进

Widget类节点的历史演进始于Web应用程序的早期开发，随着技术的进步，从简单的图形界面逐渐演变为复杂的交互式仪表盘。这一过程伴随着对用户体验和数据可视化需求的不断上升。

定义与功能比较

Widget类节点通常用于在应用程序中创建基本的图形用户界面元素，如按钮、标签等。这些节点提供了一种快速且简便的方式来构建用户界面。而Configuration类节点则用于配置整个应用程序的参数和设置，提供更高的灵活性和控制度。



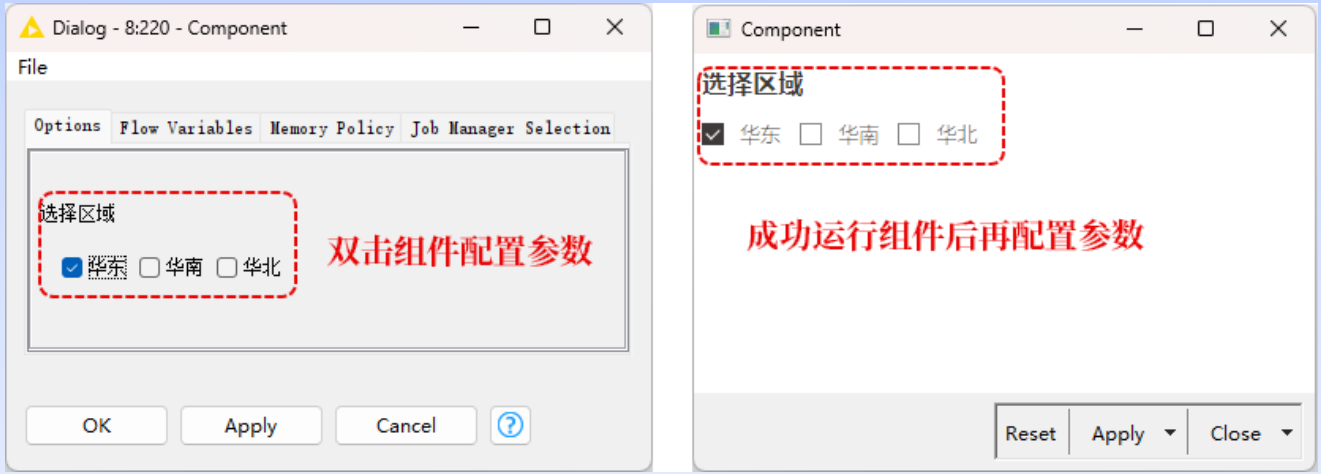
应用场景对比

Widget类节点广泛应用于需要频繁更新和动态交互的用户界面场景，例如实时数据监控和仪表显示。Configuration类节点则主要用于配置应用程序的整体参数。

执行方式的差异

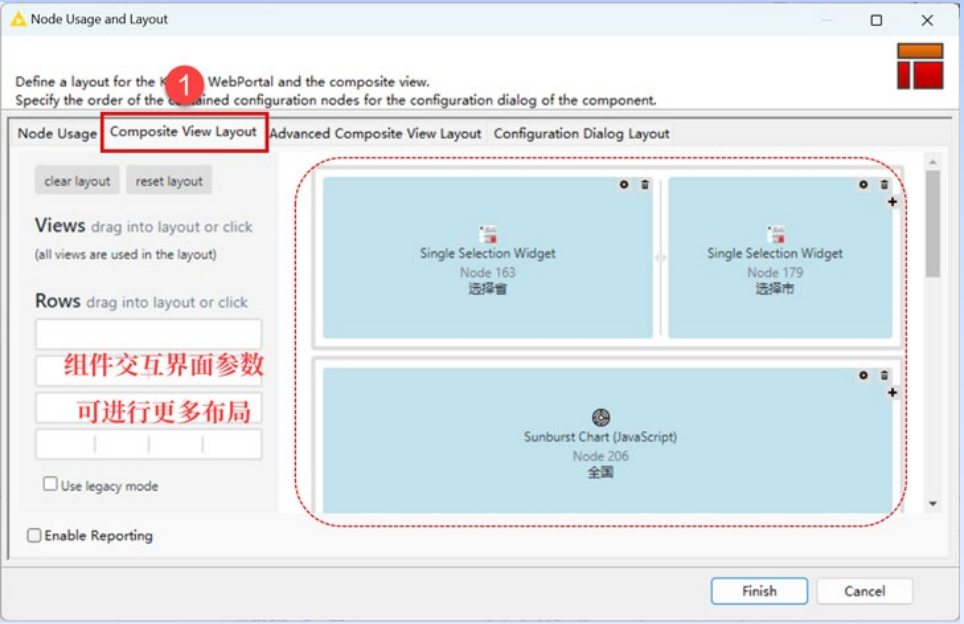
Widget类节点需要先执行一次工作流后，根据需要直接在结果界面内调整参数，而Configuration类节点，在执行组件前需要先设置参数，然后再执行工作流

Widget节点与Configuration节点比较



界面差异

Widget节点与Configuration节点比较



Widget类和View类节点在复合视图布局标签页调整显示位置

调整布局差异



Configuration类节点在配置对话框视图布局标签页调整位置，**只能上下调整**，布局形式较为单一。



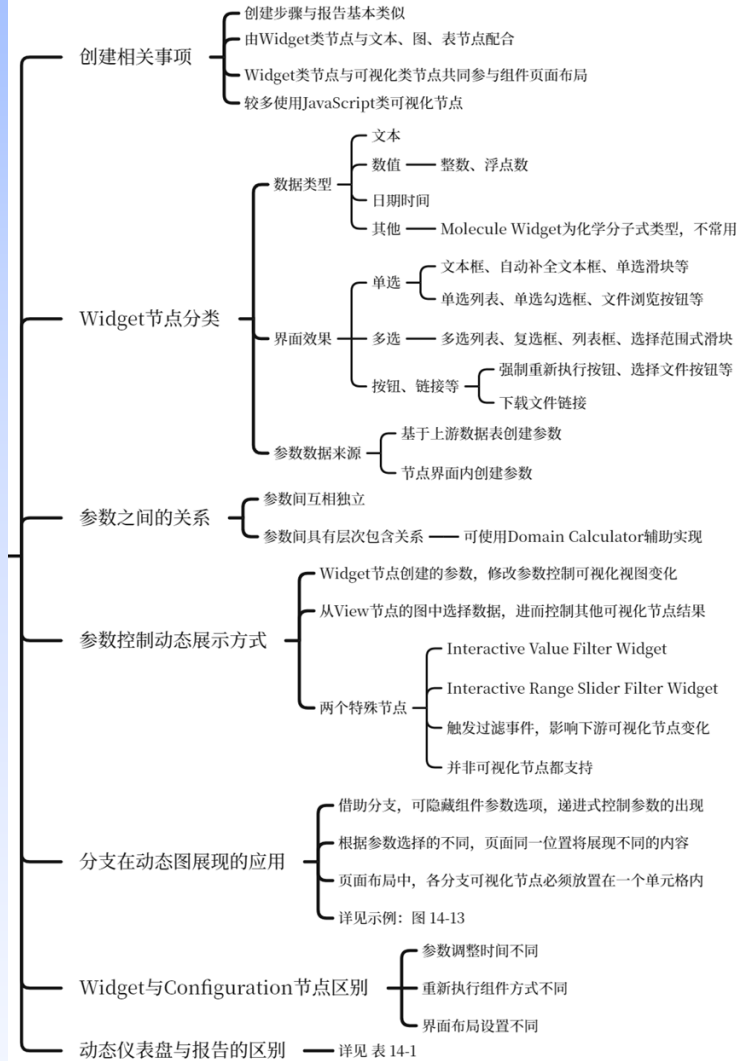
05

本章小结

动态仪表盘与报告的区别

区别点	动态仪表盘	报告
Composite View Layout中Enable Reporting选项	不勾选	勾选
可视化节点	支持View类各节点，以及JavaScript类、Plotly类可视化节点。	View类根目录下各节点，即KNIME视图类节点，不支持JavaScript类节点，需要进行迂回变通才能支持。
Widget类节点	必用，组件交互结果界面参数由该类节点控制。	一般不使用
Configuration类节点	可选节点，执行组件前设置。若考虑交互结果的便捷性，皆可由Widget类节点替代	可选节点，若使用，在执行组件前设置组件界面内参数。
结果	在本地端以窗口页面形式展示。若将工作流部署服务器端，授权用户可以通过网页查看组件界面交互结果。更改参数后可以看到新的展示结果，具有动态性。	导出为PDF或HTML文件，属于静态化报告。若需要更多格式化报告，可借助循环批量自动生成。

本章小结


创建相关事项
创建步骤与报告基本类似
由Widget类节点与文本、图、表节点配合
Widget类节点与可视化类节点共同参与组件页面布局
较多使用JavaScript类可视化节点
Widget节点分类
数据类型
文本
数值
整数、浮点数
日期时间
其他
Molecule Widget为化学分子式类型，不常用
界面效果
单选
文本框、自动补全文本框、单选滑块等
单选列表、单选勾选框、文件浏览按钮等
多选
多选列表、复选框、列表框、选择范围式滑块
按钮、链接等
强制重新执行按钮、选择文件按钮等
下载文件链接
参数数据来源
基于上游数据表创建参数
节点界面内创建参数
参数之间的关系
参数间互相独立
参数间具有层次包含关系
可使用Domain Calculator辅助实现
参数控制动态展示方式
Widget节点创建的参数，修改参数控制可视化视图变化
从View节点的图中选择数据，进而控制其他可视化节点结果
两个特殊节点
Interactive Value Filter Widget
Interactive Range Slider Filter Widget
触发过滤事件，影响下游可视化节点变化
并非可视化节点都支持
分支在动态图展现的应用
借助分支，可隐藏组件参数选项，递进式控制参数的出现
根据参数选择的不同，页面同一位置将展现不同的内容
页面布局中，各分支可视化节点必须放置在一个单元格内
详见示例：图 14-13
Widget与Configuration节点区别
参数调整时间不同
重新执行组件方式不同
界面布局设置不同
动态仪表盘与报告的区别
详见 表 14-1

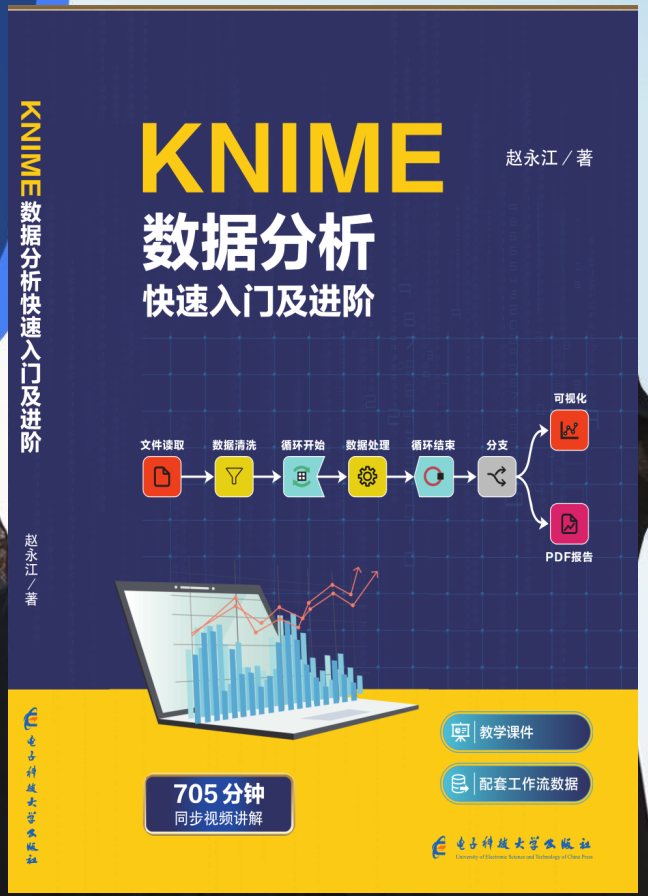
感谢观看

THANKS



第十五章
自定义图形化用户界面

打造自定义节点、图形化界面工作流



目录

01 创建自定义节点三步骤

02 管理自定义节点

03 创建图形化用户界面

04 组件用途总结

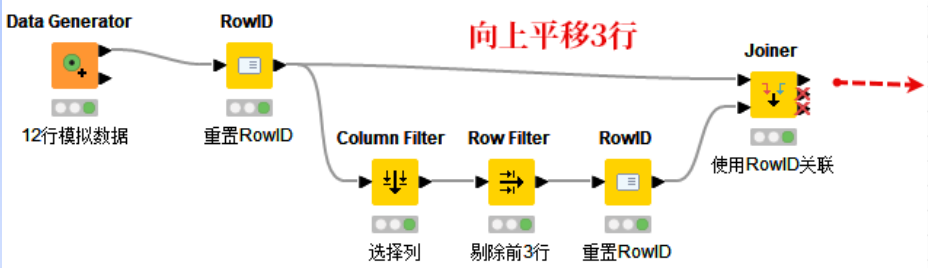
05 本章小结



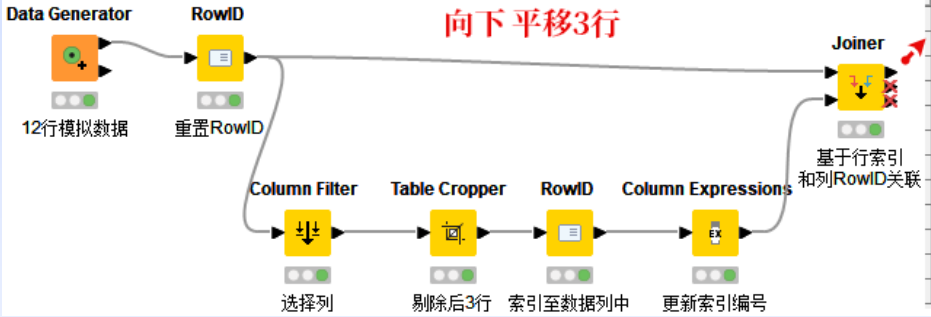
01

创建自定义节点三步骤

步骤1：实现基本功能



Unive...	Unive...	Unive...	Unive...	Clust...	Unive...	Clust...
0.814	0.929	0.908	0.149	Cluster_0	0.033	Cluster_1
0.658	0.994	0.575	0.396	Cluster_0	0.947	Cluster_1
0.632	0.826	0.26	0.462	Cluster_0	0.629	Cluster_1
0.273	0.445	0.614	0.033	Cluster_1	0.806	Cluster_2
0.417	0.699	0.075	0.947	Cluster_1	0.908	Cluster_2
0.244	0.767	0.676	0.629	Cluster_1	0.71	Cluster_2
0.575	0.388	0.875	0.806	Cluster_2	0.239	Cluster_3
0.594	0.228	0.636	0.908	Cluster_2	0.219	Cluster_3
0.428	0.965	0.868	0.71	Cluster_2	0.068	Cluster_3
0.958	0.485	0.567	0.239	Cluster_3	?	?
0.219	0.111	0.949	0.219	Cluster_3	?	?
0.155	0.552	0.576	0.068	Cluster_3	?	?

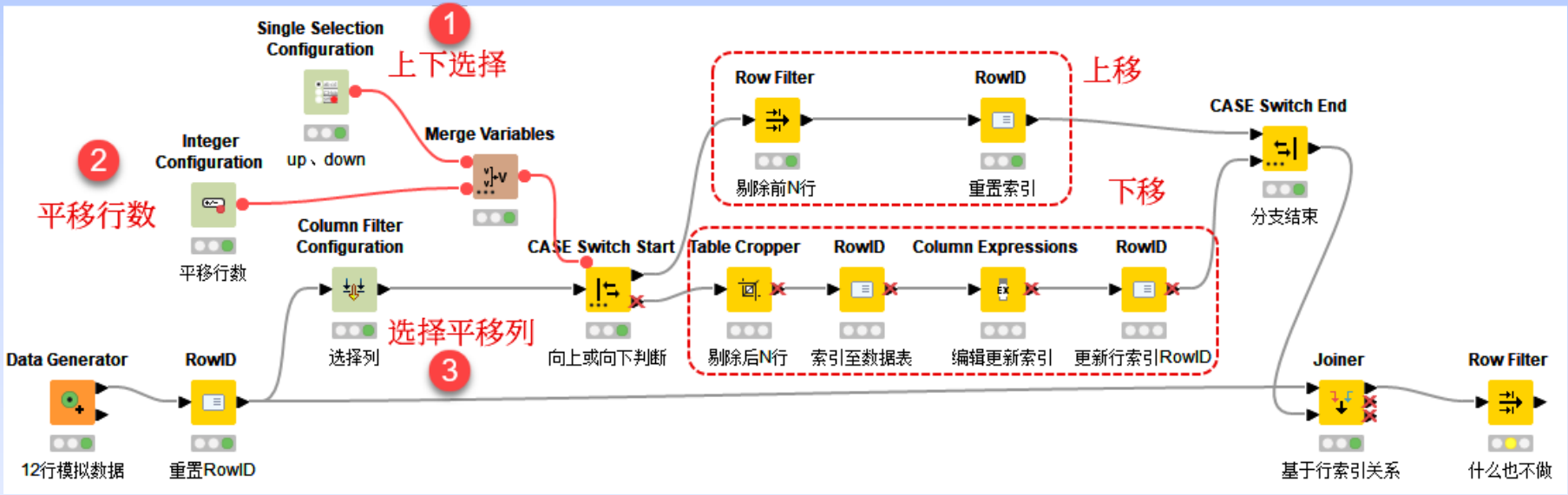


Unive...	Unive...	Unive...	Unive...	Clust...	Unive...	Clust...
0.814	0.929	0.908	0.149	Cluster_0	?	?
0.658	0.994	0.575	0.396	Cluster_0	?	?
0.632	0.826	0.26	0.462	Cluster_0	?	?
0.273	0.445	0.614	0.033	Cluster_1	0.149	Cluster_0
0.417	0.699	0.075	0.947	Cluster_1	0.396	Cluster_0
0.244	0.767	0.676	0.629	Cluster_1	0.462	Cluster_0
0.575	0.388	0.875	0.806	Cluster_2	0.033	Cluster_1
0.594	0.228	0.636	0.908	Cluster_2	0.947	Cluster_1
0.428	0.965	0.868	0.71	Cluster_2	0.629	Cluster_1
0.958	0.485	0.567	0.239	Cluster_3	0.806	Cluster_2
0.219	0.111	0.949	0.219	Cluster_3	0.908	Cluster_2
0.155	0.552	0.576	0.068	Cluster_3	0.71	Cluster_2

目标：

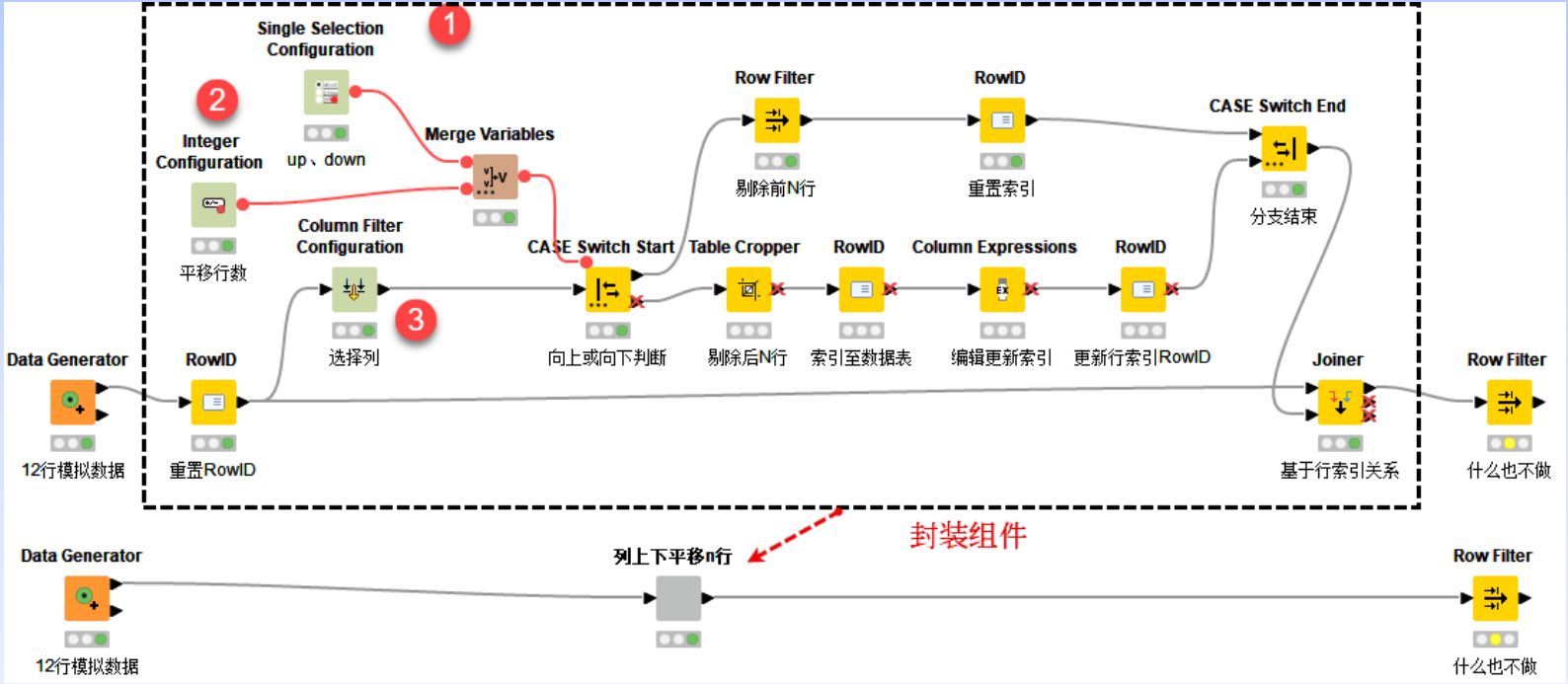
实现列向上、向下平移功能，最后封装为组件（自定义节点），通过组件参数界面设置参数，方便多次使用

步骤2：参数化关键节点



当对关键界面进行参数化时，需要根据情况，改造原有工作流

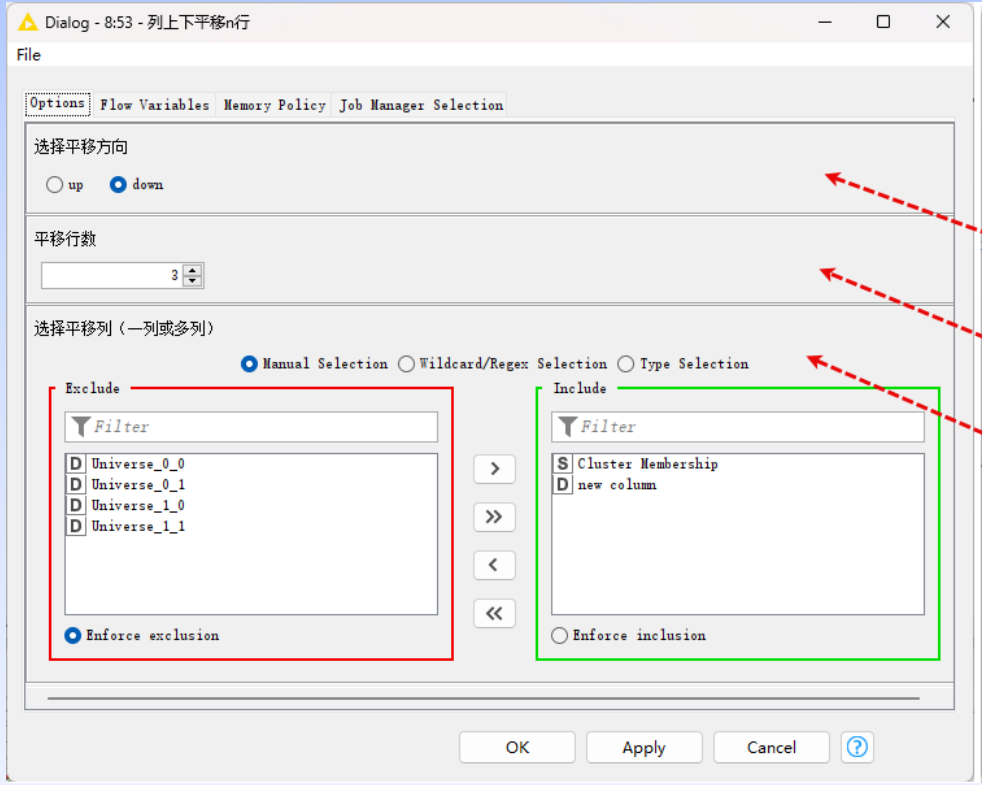
步骤3：封装组件并设置参数布局



目标：

实现列向上、向下平移功能，最后封装为组件（自定义节点），通过组件参数界面设置参数，方便多次使用

步骤3：封装组件并设置参数布局





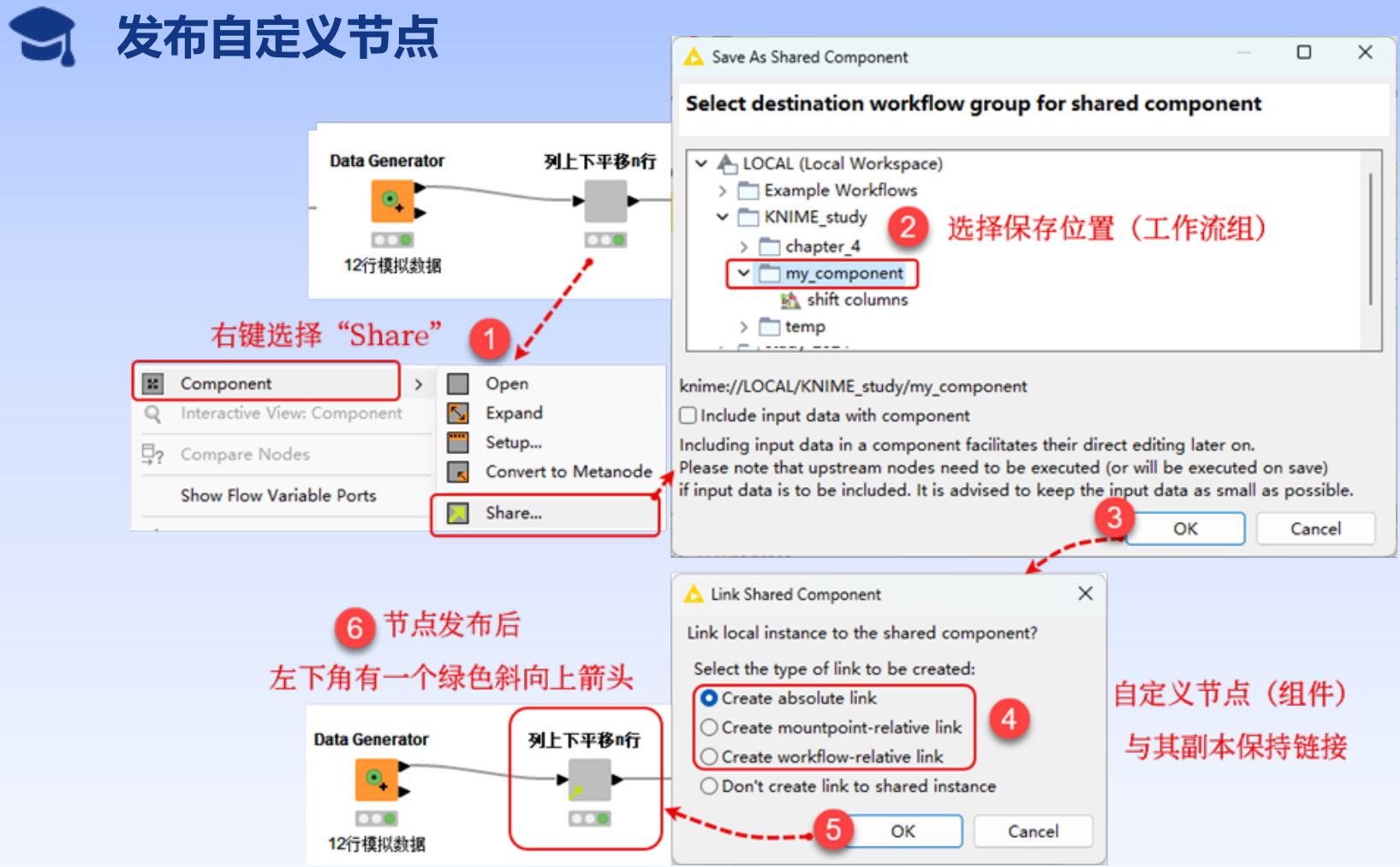
在组件的配置对话框选项卡中（Configuration Dialog Layout）设置各Configuration类节点的上下位置。



02

管理自定义节点

发布自定义节点


Data Generator
列上下平移n行
12行模拟数据
右键选择“Share”
1
Component
Interactive View: Component
Compare Nodes
Show Flow Variable Ports
Open
Expand
Setup...
Convert to Metanode
Share...
Save As Shared Component
Select destination workflow group for shared component
LOCAL (Local Workspace)
Example Workflows
KNIME_study
2
选择保存位置（工作流组）
chapter_4
my_component
shift columns
temp
knime://LOCAL/KNIME_study/my_component
Include input data with component
Including input data in a component facilitates their direct editing later on.
Please note that upstream nodes need to be executed (or will be executed on save)
if input data is to be included. It is advised to keep the input data as small as possible.
3
OK
Cancel
Link Shared Component
Link local instance to the shared component?
Select the type of link to be created:
Create absolute link
Create mountpoint-relative link
Create workflow-relative link
Don't create link to shared instance
4
5
OK
Cancel
自定义节点（组件）
与其副本保持链接
6
节点发布后
左下角有一个绿色斜向上箭头
Data Generator
列上下平移n行
12行模拟数据

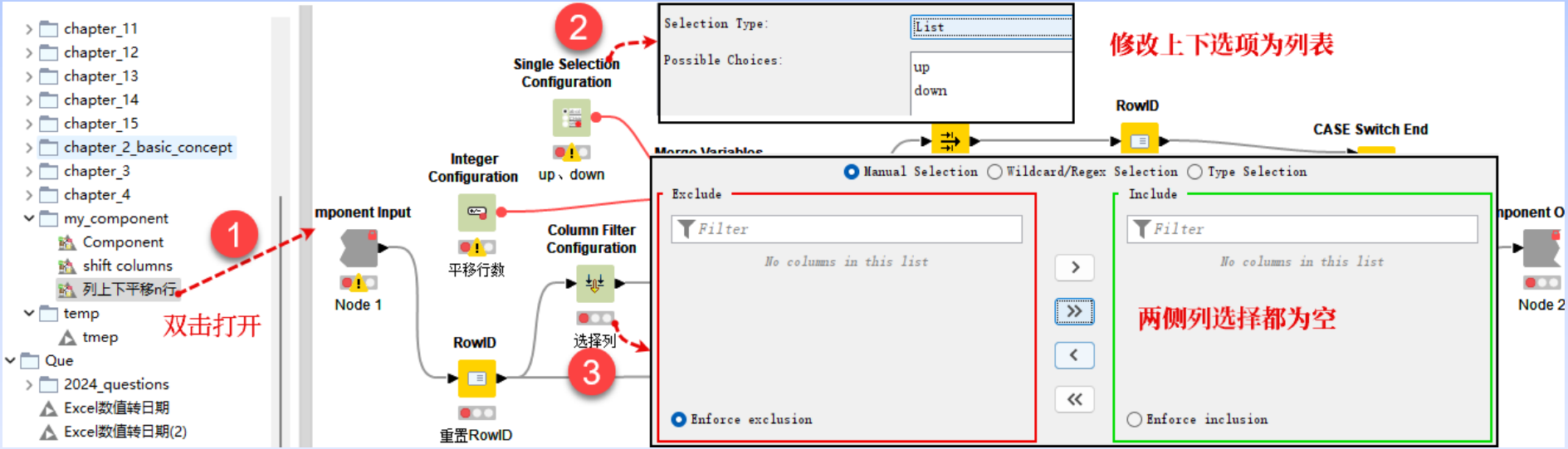
使用自定义节点



- 自定义节点为原始版本
- 此处自定义节点的修改会影响使用该自定义节点的其他工作流
- 该自定义节点可保存本地、或KNIME Community Hub中

使用自定义节点为原始自定义节点的副本

修改自定义节点



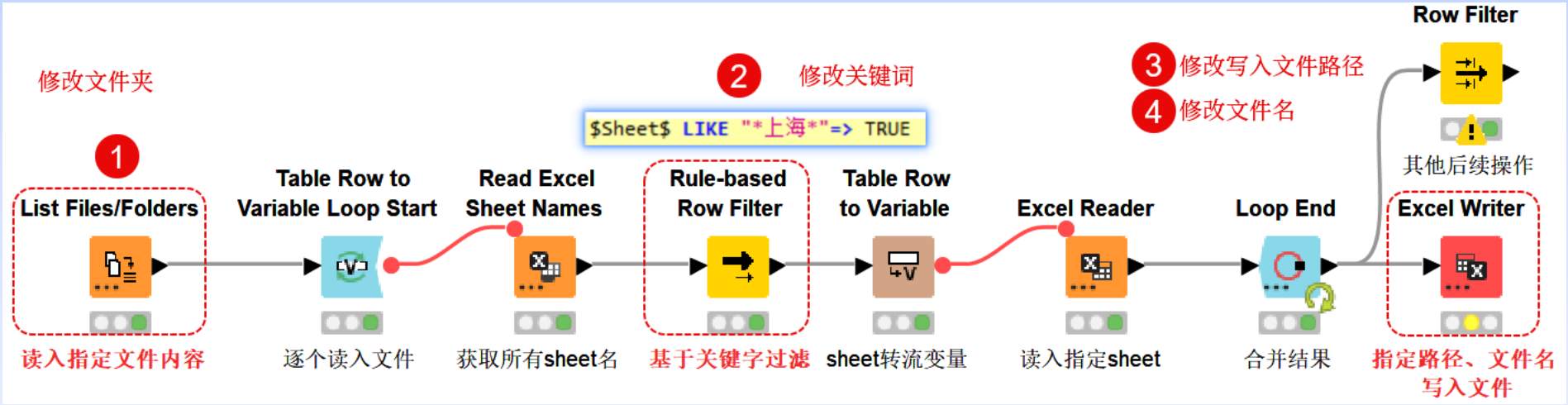
移除选中的列，避免组件“记忆”了创建组件时默认选择的列。详见本章案例解释）



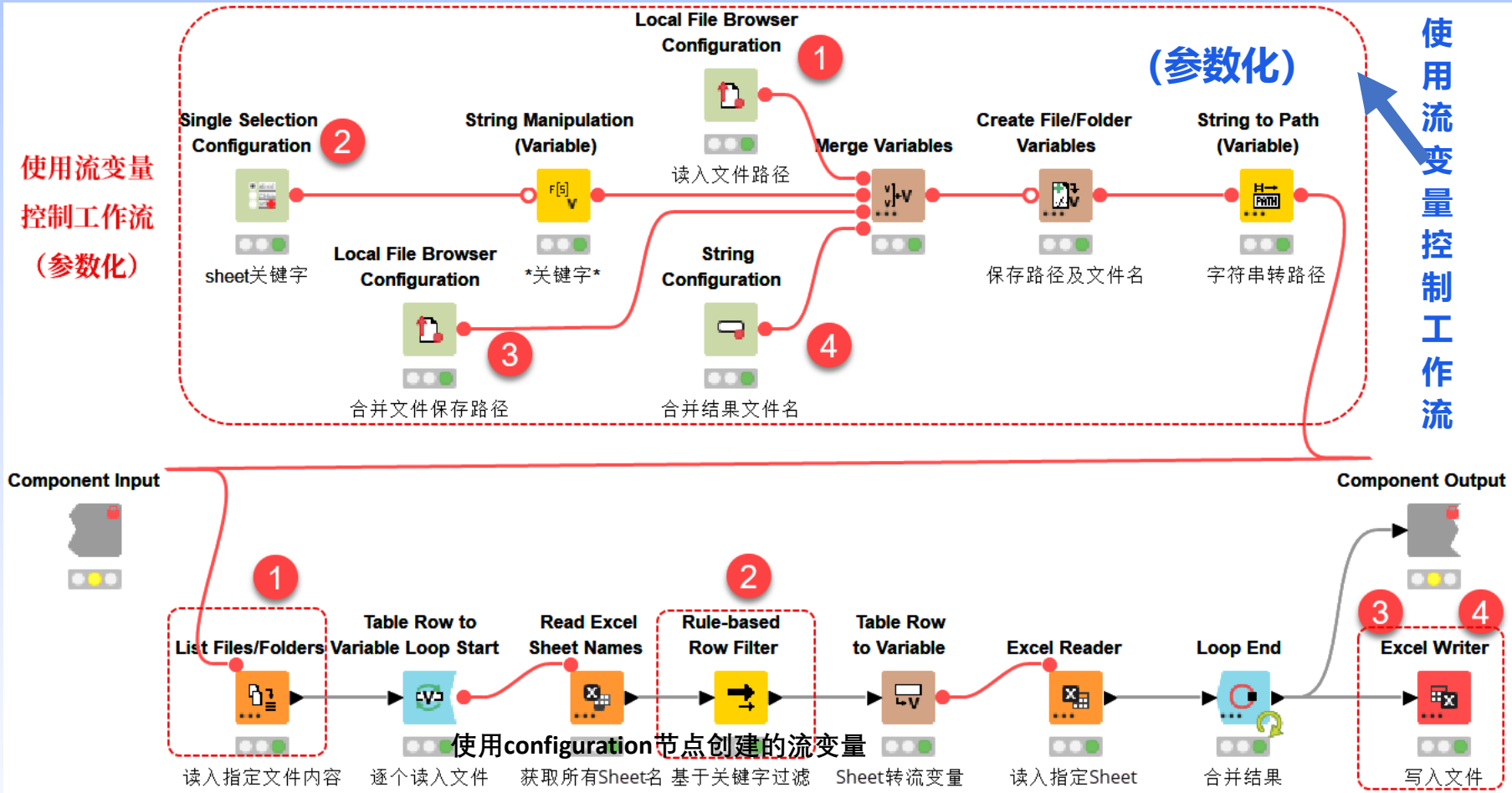
03

创建图形化用户界面

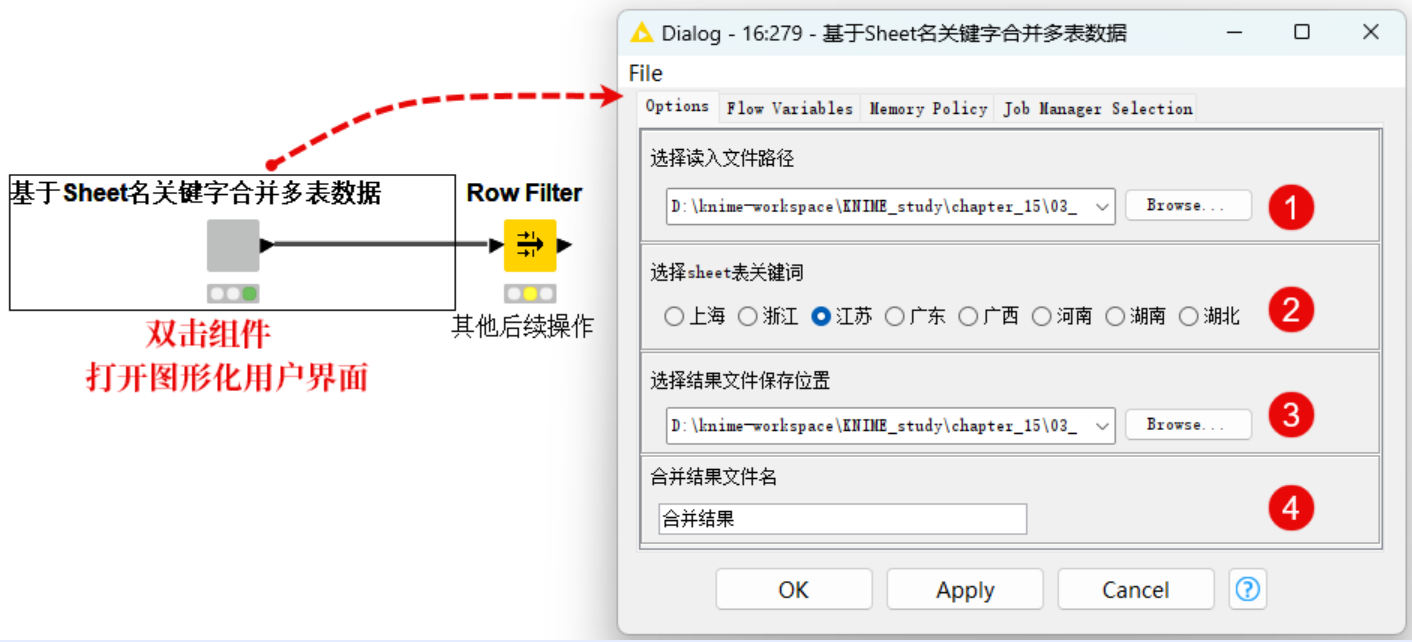
案例：合并多个Excel文件中指定关键字的sheet表



案例：合并多个Excel文件中指定关键字的sheet表



交互式控件（对话框）



- 该图形化用户界面可以像一个小工具一样，在一个界面内修改各个节点的设置，隐藏了技术细节，方便任意用户使用。
- 由于该界面主要主要针对特定结构的数据文件，在使用时仅对某一类数据文件有效。



04

组件用途总结

组件的五种应用场景

Configuration与Widget节点各自对应组件的不同应用场景

组件布局类别	应用场景	节点	Component功能
复合视图（Composite View）	报告（场景1）	View根目录下节点（其他可视化节点不直接支持）、Configuration类、Reporting类节点，不使用Widget类节点。	利用组件封装和布局功能，将图、文字、数据表有机合理的组织在一起，输出静态化报告文件。
复合视图（Composite View）	数据应用：动态仪表板盘（场景2）	所有可视化节点、Widget类节点、可选Configuration类节点。	利用组件封装和布局功能，将图、文字、数据表有机合理的组织在一起，并利用Widget类节点实现对参数的调整，并重新运行，交互式查看调整参数后的新输出。
参数界面（Configuration Dialog）	创建图形化参数界面的自定义节点，重复使用（场景3）	Configuration类节点、可选Views节点。	实现自定义节点的图形化界面，方便重复利用及分享。通过多个节点组合，实现一个特定功能（可类比编程中实现某个功能的函数），节点一般需定义输入和输出。
参数界面（Configuration Dialog）	为工作流创建图形化参数界面（场景4）	Configuration类节点、可选Views节点。	实现特定工作流的图形化用户界面，隐藏细节，简化工作流，便捷化调整工作流参数。
无需界面	读取超大文件（场景5）	节点库中支持streamable方式的节点。	流式执行（Streaming execution）是一种数据处理模式，它允许批处理数据，无需等待所有数据都读入完毕才开始处理。常用与实时处理数据，或处理超大文件（如电脑内存16G，需要处理32G的大文件）。

组件Component与元节点Metanode的区别

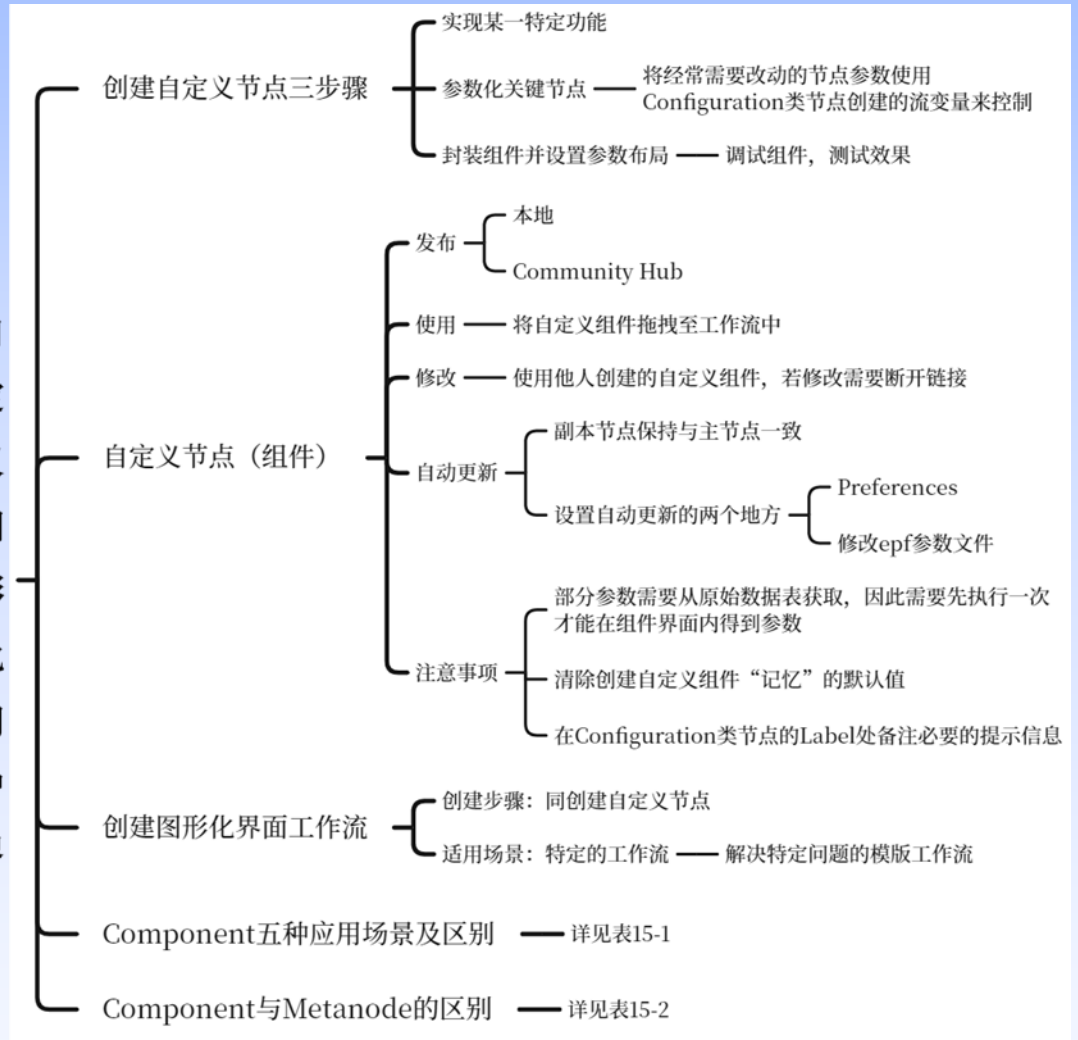
	Component	Metanode
复合视图	√	×
参数界面（对话框）	√	×
隔离流变量	√	×
流式处理	√	×
发布、分享自定义节点	√	×
隐藏细节，简化工作流	√	√
节点状态（底部红黄绿）	√	×



05

本章小结

本章小结


自定义图形化用户界
创建自定义节点三步骤
实现某一特定功能
参数化关键节点
将经常需要改动的节点参数使用Configuration类节点创建的流变量来控制
封装组件并设置参数布局
调试组件，测试效果
自定义节点（组件）
发布
本地
Community Hub
使用
将自定义组件拖拽至工作流中
修改
使用他人创建的自定义组件，若修改需要断开链接
自动更新
副本节点保持与主节点一致
设置自动更新的两个地方
Preferences
修改epf参数文件
注意事项
部分参数需要从原始数据表获取，因此需要先执行一次才能在组件界面内得到参数
清除创建自定义组件“记忆”的默认值
在Configuration类节点的Label处备注必要的提示信息
创建图形化界面工作流
创建步骤：同创建自定义节点
适用场景：特定的工作流
解决特定问题的模版工作流
Component五种应用场景及区别
详见表15-1
Component与Metanode的区别
详见表15-2

感谢观看

THANKS



第16章
组件的流式执行

探索流式执行机制与应用



目录

01 流式执行概述

02 流式执行两步骤

03 应用：处理超大单个文件

04 本章小结



01

流式执行概述

流式执行与顺序执行区别

任务

拉来一车萝卜原材料（类比数据），假定生产腌制萝卜条有四步工序，依次为洗萝卜、切萝卜、腌萝卜、称重封装（各步骤视为一个个节点）

顺序执行

- 先洗完一车所有萝卜，再执行切萝卜，每道工序等需要上步工序完成再开始本道工序。
- 对应KNIME各节点，可理解为逐个节点执行，有时也叫做逐表执行

流式执行

- 先把萝卜等分为多份，比如50筐，每洗完一筐就流入下步工序，第二道工序每处理完一筐就传入下道工序，并接收上一步工序传来新的一筐萝卜。这样，每道工序都开始动起来。这样拆解为不同批次，流水线作业方式效率更高。
- 对KNIME来说，将大数据文件切分为不同的数据块，每块分别传递和执行，减少了输入、输出和内存占用量，充分利用各个节点同时处理数据，加快了处理速度

流式执行与顺序执行区别

执行方式差异

流式执行和顺序执行的核心区别在于执行方式。流式执行通过连续、高效的数据处理模式，以数据流的形式处理任务，而顺序执行则按依次顺序逐条处理任务。

资源利用效率

流式执行通过并行处理多个任务，提高了资源利用效率。它能够同时处理多个任务，显著减少等待时间，提高整体系统的响应速度和性能。



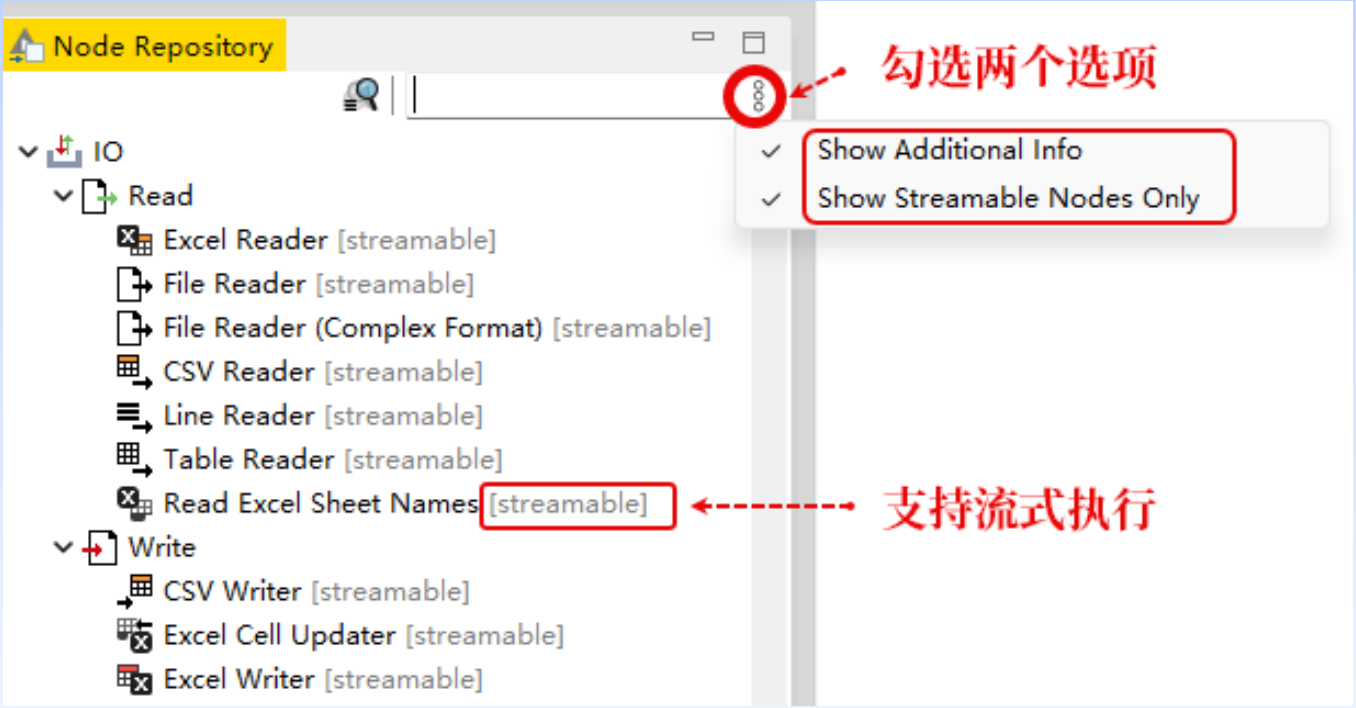
内存占用情况

在处理大规模数据集时，流式执行通常具有较低的内存占用率。其高效的数据管理和优化的内存使用策略，使得系统能够在处理大量数据时保持良好的性能。

存储中间结果方面

流式执行不存储各流式节点中间结果。顺序执行（逐表执行、逐节点执行）各个节点均保存中间执行结果。

查看支持流式执行节点（仅限经典界面）





02

流式执行两步骤

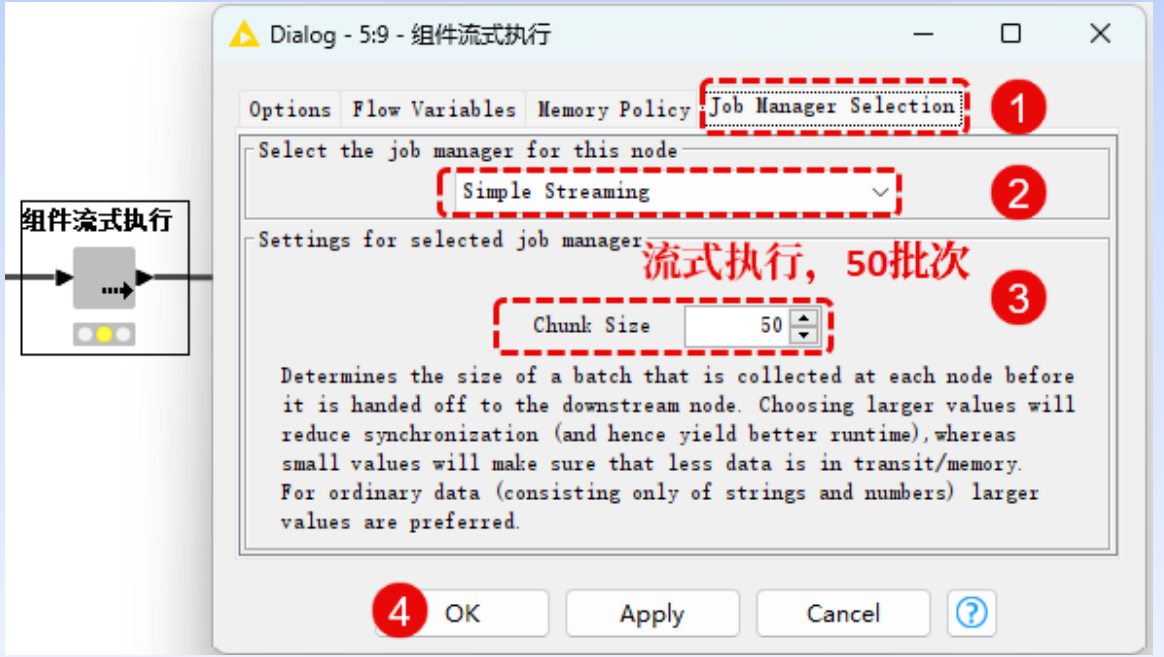
流式执行步骤

Step1：

使用支持流式处理的节点创建工作流

Step2：

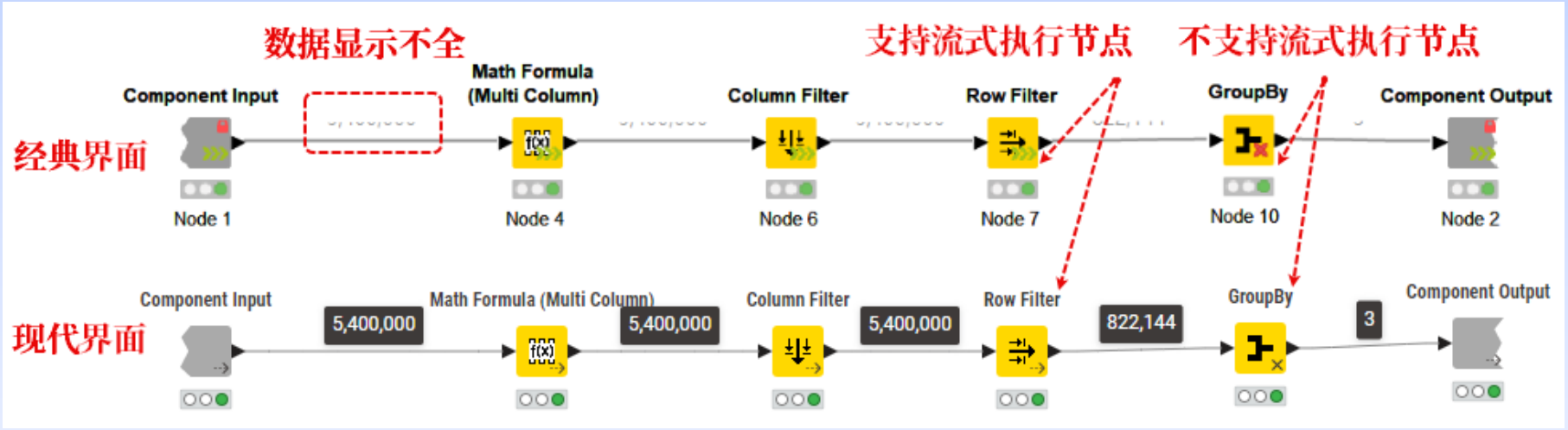
将支持流式处理的节点选中，创建组件，并设置流式执行



注意：

- 早期版本若使用流式处理需要额外安装扩展KNIME Streaming Execution（Beta）扩展（该扩展下无任何节点）。
- 5.2及之后版本不需要额外安装即可使用流式处理。

组件内流式节点与非流式节点区别





03

应用：处理超大单个文件

处理超大文件方法

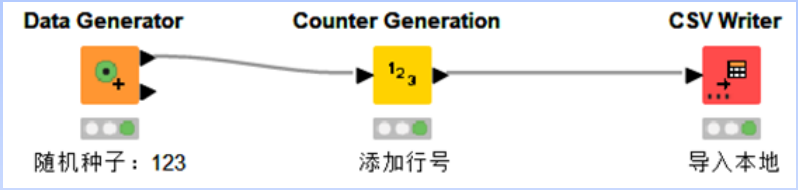
流式执行

流式执行通过逐行读取文件内容，避免将整个文件加载到内存中，从而有效管理内存使用。这种方法可以处理超大单个文件，防止因内存溢出或性能问题导致的错误。

利用循环处理超大单个文件

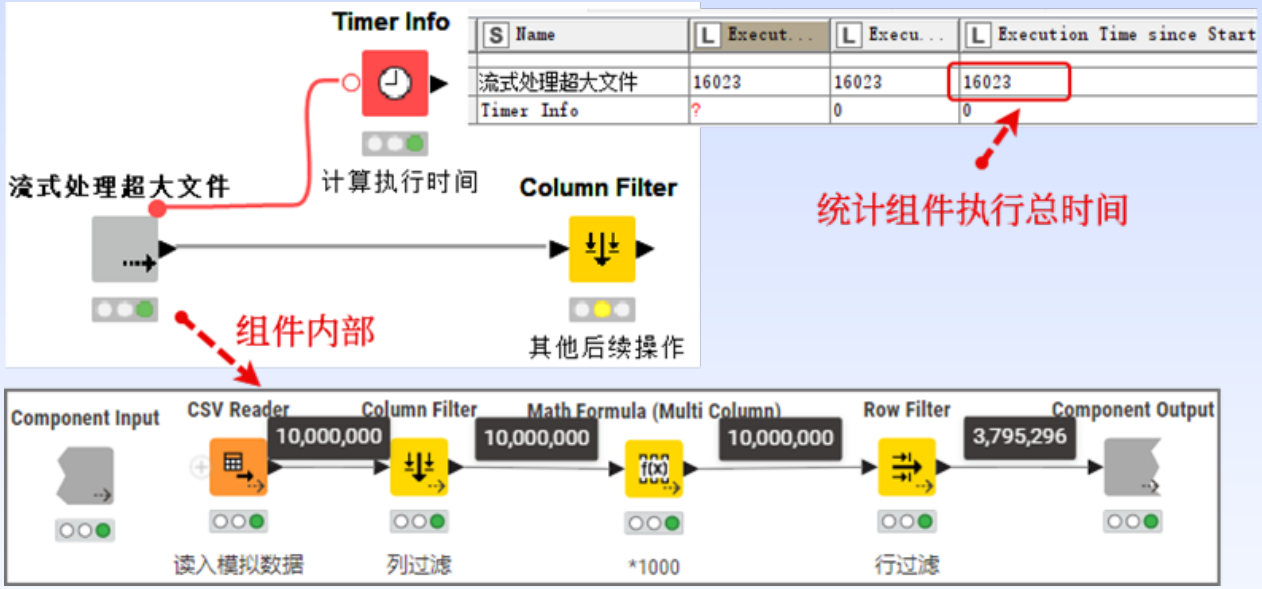
可以利用循环来处理超大单个文件。通过循环，可以实现对文件内容的分块读取和处理，从而提高处理效率并减少内存占用，适用于超大文件的流式处理。

示例1：流式处理大数据文件

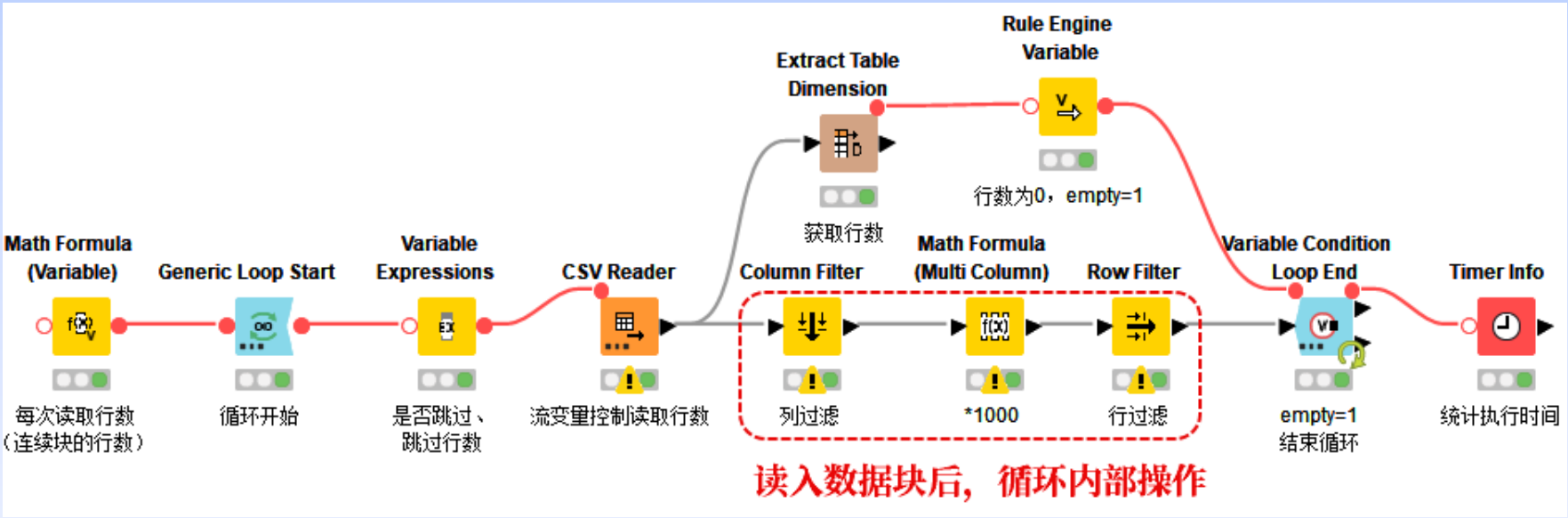


生成一千万条模拟数据

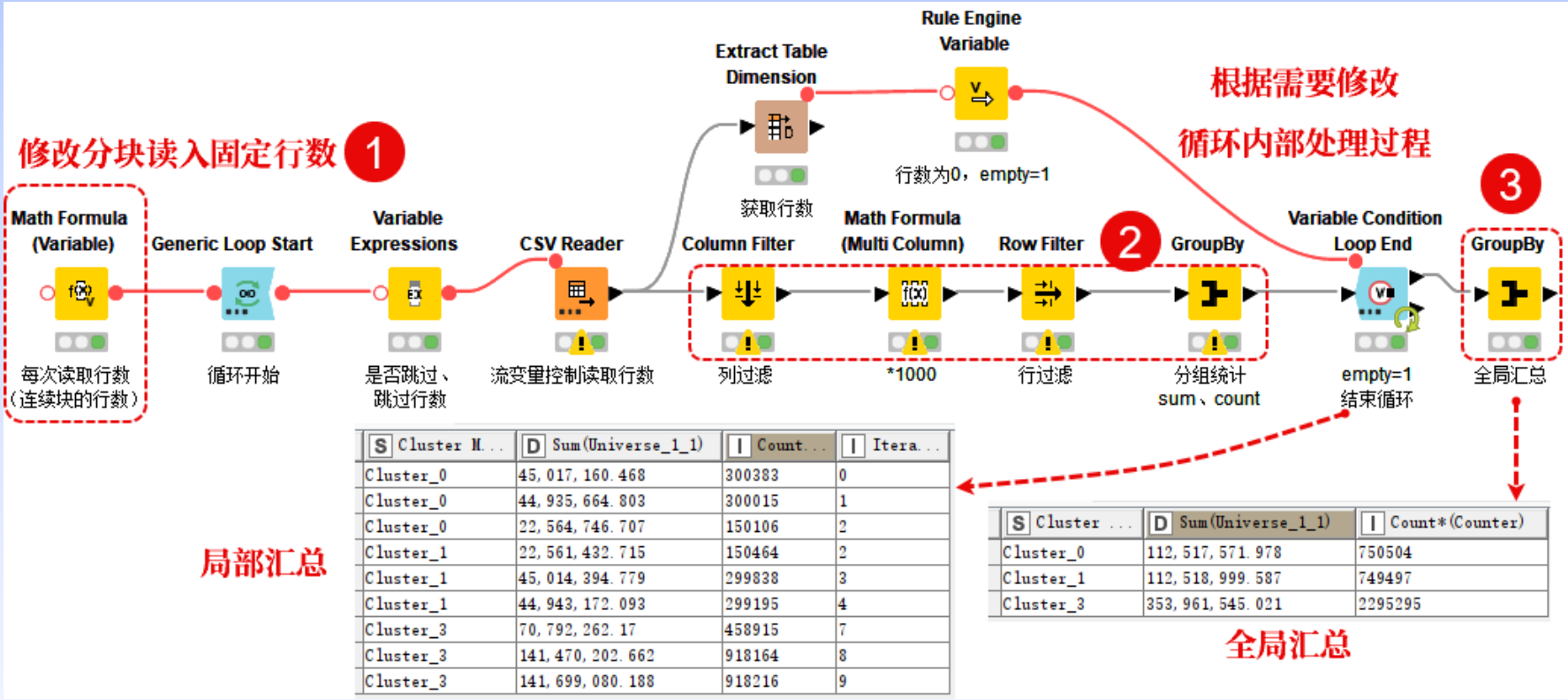
使用流式方式读入及处理模拟数据



示例2：利用循环分块读入超大文件数据



示例2：局部汇总与全局汇总



局部汇总

Cluster N...	Sum(Universe_1_1)	Count...	Itera...
Cluster_0	45,017,160.468	300383	0
Cluster_0	44,935,664.803	300015	1
Cluster_0	22,564,746.707	150106	2
Cluster_1	22,561,432.715	150464	2
Cluster_1	45,014,394.779	299838	3
Cluster_1	44,943,172.093	299195	4
Cluster_3	70,792,262.17	458915	7
Cluster_3	141,470,202.662	918164	8
Cluster_3	141,699,080.188	918216	9

全局汇总

Cluster ...	Sum(Universe_1_1)	Count*(Counter)
Cluster_0	112,517,571.978	750504
Cluster_1	112,518,999.587	749497
Cluster_3	353,961,545.021	2295295



04

本章小结

流式执行关键要点总结

流式执行定义

流式执行是一种通过数据流进行计算和处理的方法，其核心在于数据的连续输入和即时处理，无需等待整个数据处理完毕即可返回部分结果。

流式执行优势

流式执行具有高效性和实时性的特点，能够显著提升数据处理速度，减少系统资源的浪费，并提高整体系统的可靠性和稳定性。

流式执行应用场景

在流式执行过程中，面对超大单个文件可能导致系统资源耗尽。为应对这一问题，可以采用分块读取和并行处理的方法，将大文件分割为多个小部分，并利用多线程或多进程并行处理，提高处理效率。

流式执行局限性

流式执行虽然在性能上具有显著优势，但支持流式执行的节点有限，不适用于所有类型的数据处理任务。

感谢观看

THANKS



第十七章

利用Python赋能KNIME



目录

01 Python Integration集成环境

02 自定义Python虚拟环境

03 Excel数据逐行填充至Word模版

04 Pixi创建和应用虚拟环境

05 本章小结



01

Python Integration集成环境

安装及设置Python Integration

01 **Python Integration概述**

Python Integration是KNIME中用于集成Python的模块，用户无需另外安装Python，通过Python Integration，用户可以在KNIME中使用Python代码执行复杂的数据分析任务。

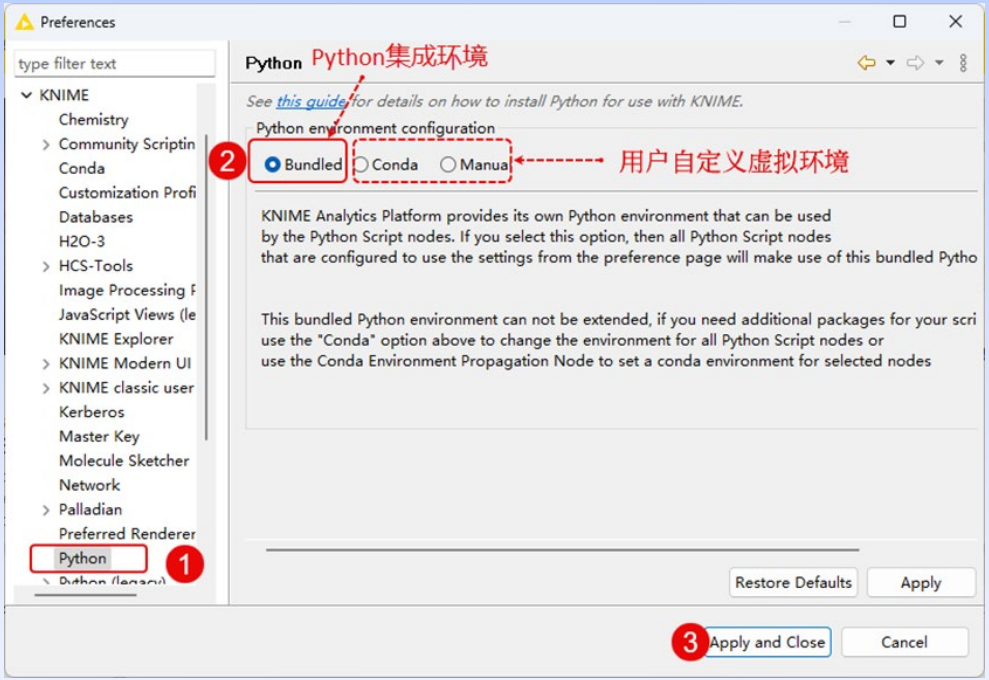
02 **安装Python Integration**

安装Python Integration扩展。该集成环境包含了常用的Numpy、Pandas等常用模块，用户不能删除或升级各模块版本。

03 **配置Python环境**

在Python Script或Python View节点中选择Python集成环境选项，该环境即为Python Integration扩展所安装的环境。

设置集成环境



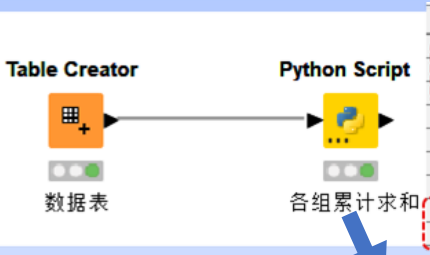


在偏好中设置集成环境

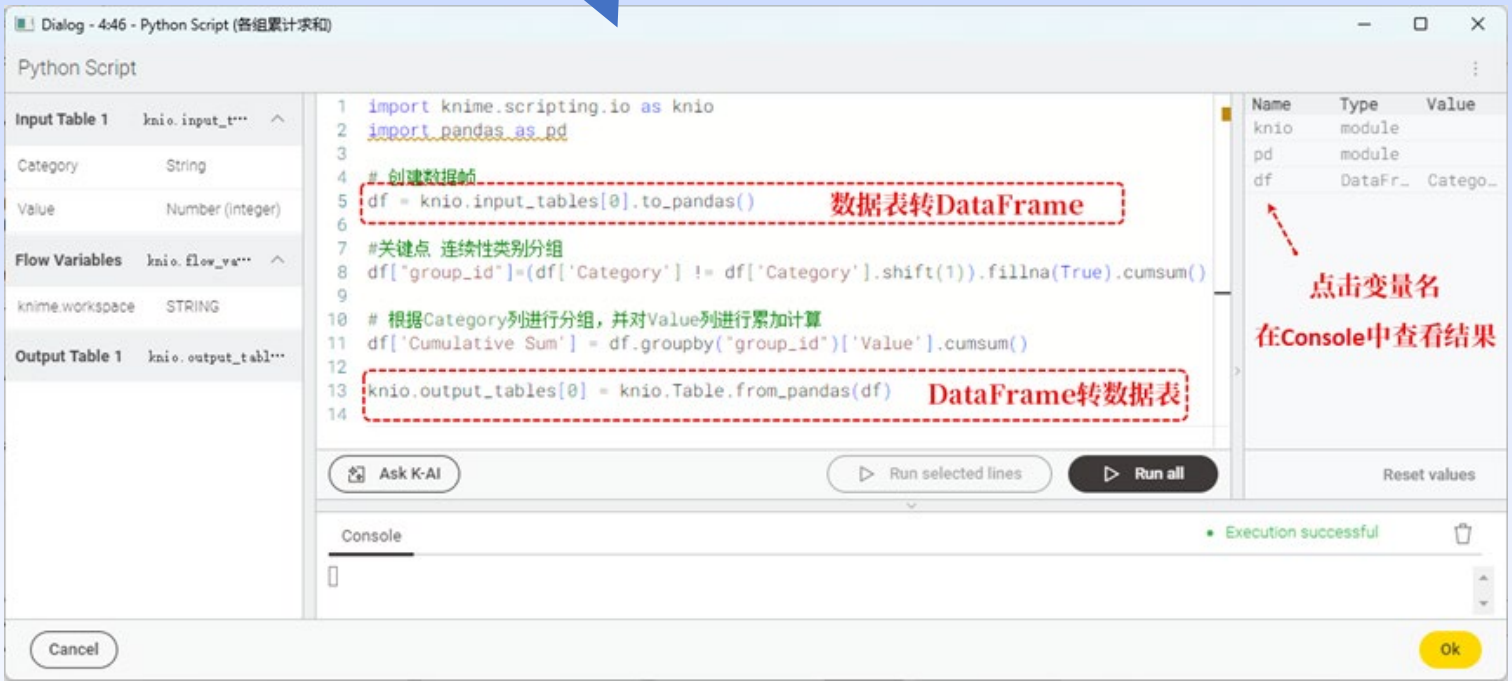
在Python Script节点中使用集成环境

使用Python类节点：案例--连续类别组累加（方法1）

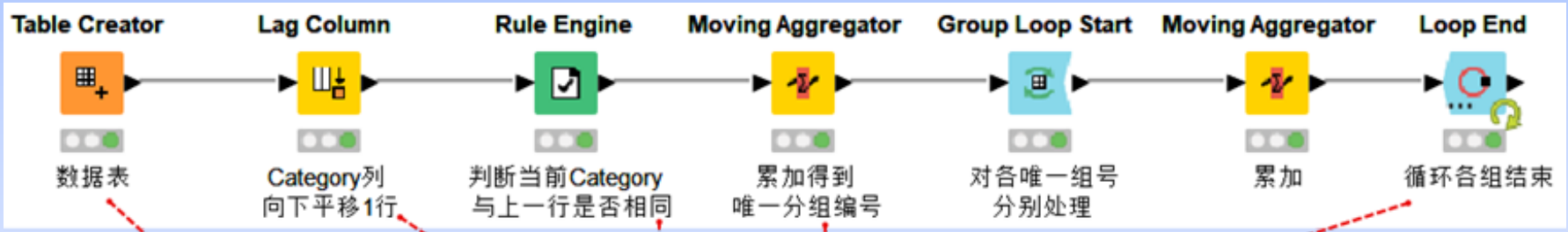
Category	Value
A	1
A	2
A	3
B	5
B	1
C	2
C	1
A	8
A	2



Category	Value	group_id	Cumulative Sum
A	1	1	1
A	2	1	3
A	3	1	6
B	5	2	5
B	1	2	6
C	2	3	2
C	1	3	3
A	8	4	8
A	2	4	10



案例--连续类别组累加（方法2）



Category	Value	Category(-1)	is_same	Sum(is_same)	Sum(Value)
A	1	?	1	1	1
A	2	A	0	1	3
A	3	A	0	1	6
B	5	A	1	2	5
B	1	B	0	2	6
C	2	B	1	3	2
C	1	C	0	3	3
A	8	C	1	4	8
A	2	A	0	4	10

- 若不使用Python Script或其他编程节点，则工作流将需要使用多个节点。对比方法1和方法2发现，Python Script节点化简了各个计算步骤，但需要一定的编程基础。
- 方法2重现了方法1的各个计算过程，两者处理逻辑完全相同。

查看Python集成环境各模块及python版本




```python
import knime.scripting.io as knio
import pkg_resources
import pandas as pd

# 获取所有已安装的包信息
packages = [(package.key, package.version) for package in pkg_resources.working_set]

# 将包信息转换为DataFrame
df_packages = pd.DataFrame(packages, columns=['Package Name', 'Version'])

knio.output_tables[0] = knio.Table.from_pandas(df_packages)
```

Package Name	Version
asttokens	2.4.1
colorama	0.4.6
comm	0.2.0
debugpy	1.8.0
decorator	5.1.1
executing	2.0.1
ipykernel	6.27.0
ipython	8.17.2
jedi	0.19.1
jupyter-client	8.6.0
jupyter-core	5.5.0

查看集成环境中各模块版本

```python
import knime.scripting.io as knio
import sys
import pandas as pd

knio.output_tables[0] = knio.Table.from_pandas(
    pd.DataFrame( {'Python_version':[sys.version]}) )
```

查看Python版本

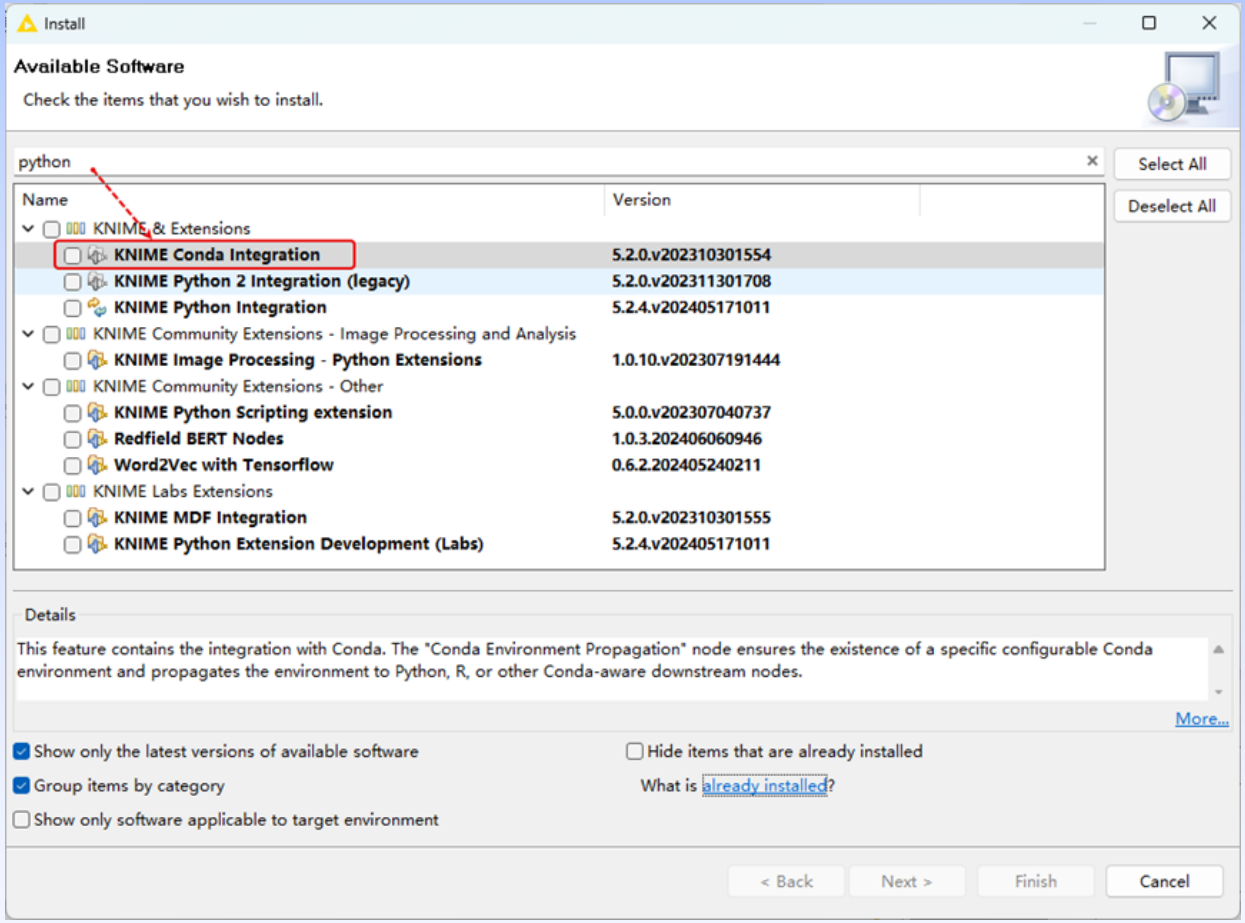
集成环境中，开箱即用Python及常用的包，不能升级Python版本或已安装的各模块。



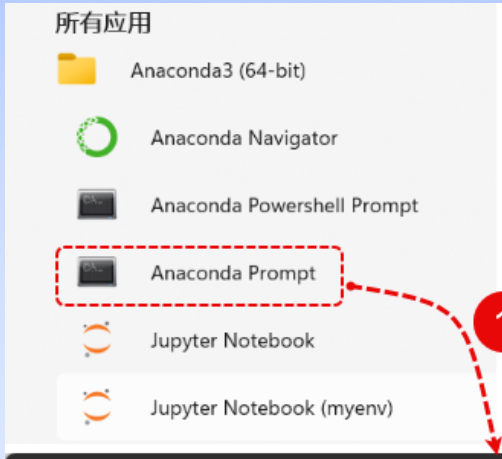
02

自定义Python虚拟环境

安装扩展


Install
Available Software
Check the items that you wish to install.
python
Select All
Deselect All
Name
Version
KNIME & Extensions
KNIME Conda Integration
5.2.0.v202310301554
KNIME Python 2 Integration (legacy)
5.2.0.v202311301708
KNIME Python Integration
5.2.4.v202405171011
KNIME Community Extensions - Image Processing and Analysis
KNIME Image Processing - Python Extensions
1.0.10.v202307191444
KNIME Community Extensions - Other
KNIME Python Scripting extension
5.0.0.v202307040737
Redfield BERT Nodes
1.0.3.202406060946
Word2Vec with Tensorflow
0.6.2.202405240211
KNIME Labs Extensions
KNIME MDF Integration
5.2.0.v202310301555
KNIME Python Extension Development (Labs)
5.2.4.v202405171011
Details
This feature contains the integration with Conda. The "Conda Environment Propagation" node ensures the existence of a specific configurable Conda environment and propagates the environment to Python, R, or other Conda-aware downstream nodes.
More...
Show only the latest versions of available software
Group items by category
Show only software applicable to target environment
Hide items that are already installed
What is already installed?
< Back
Next >
Finish
Cancel

创建虚拟环境

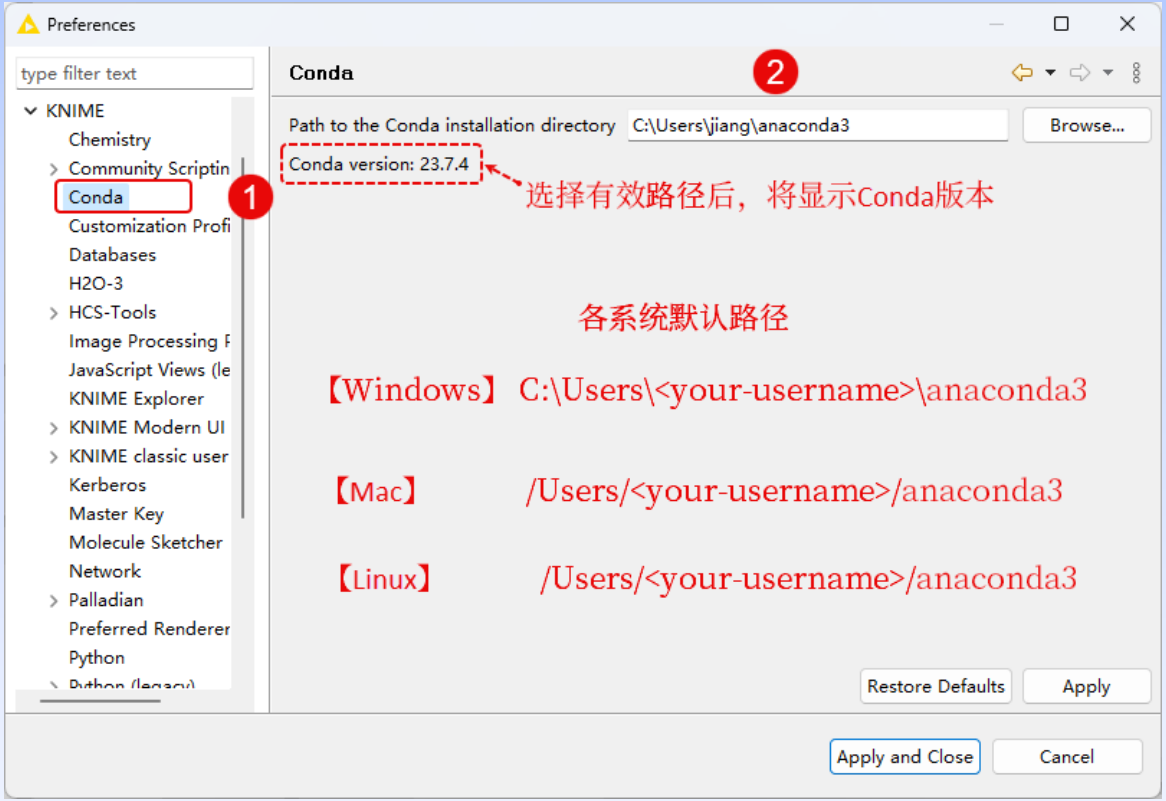


1.Anaconda默认“base”环境，该环境不能被KNIME所使用，
需要另外创建一个新环境，如 my_python_env

2.创建虚拟环境时，安装必备包，并指定Python版本

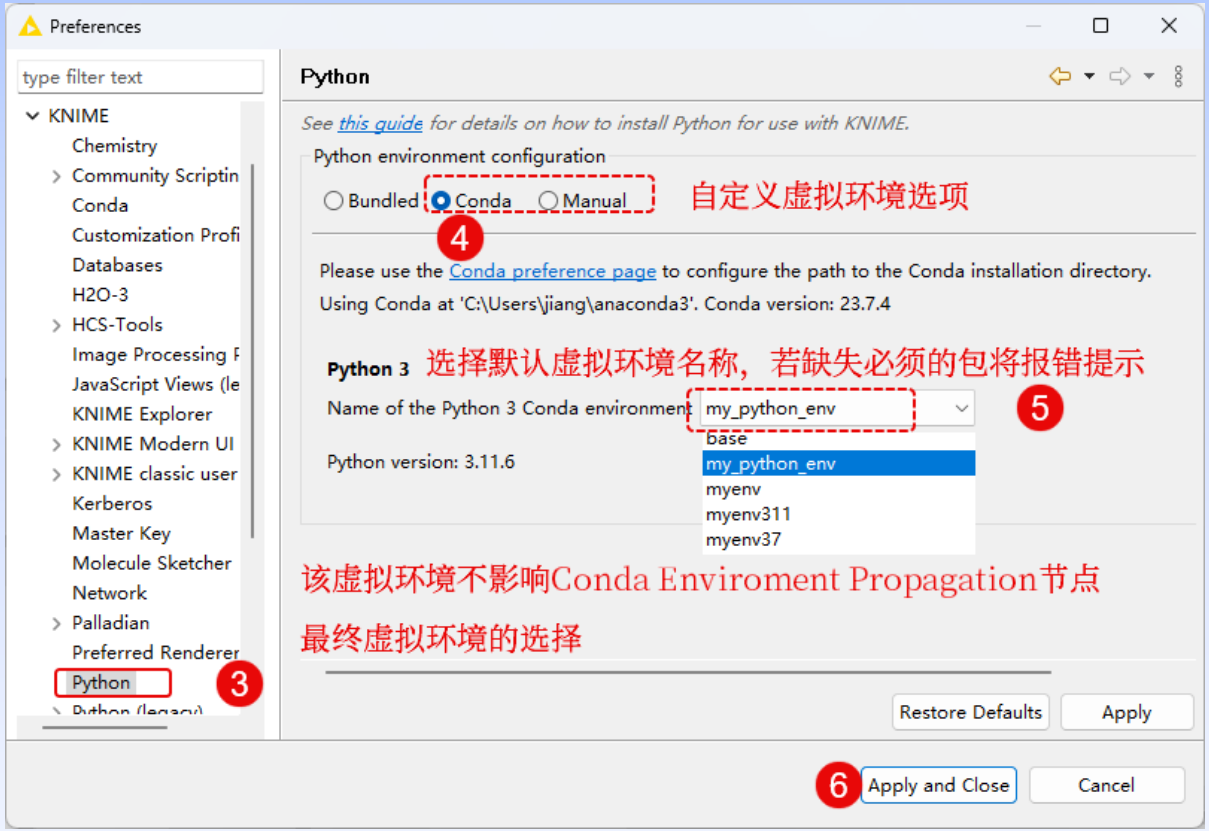


KNIME中设置虚拟环境



获取Python虚拟环境管理工具Conda，Anaconda软件自带Conda工具，也可只安装管理工具Conda

KNIME中设置虚拟环境

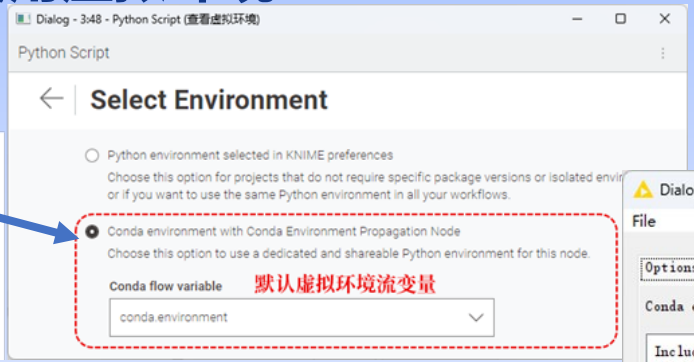


选择默认虚拟环境

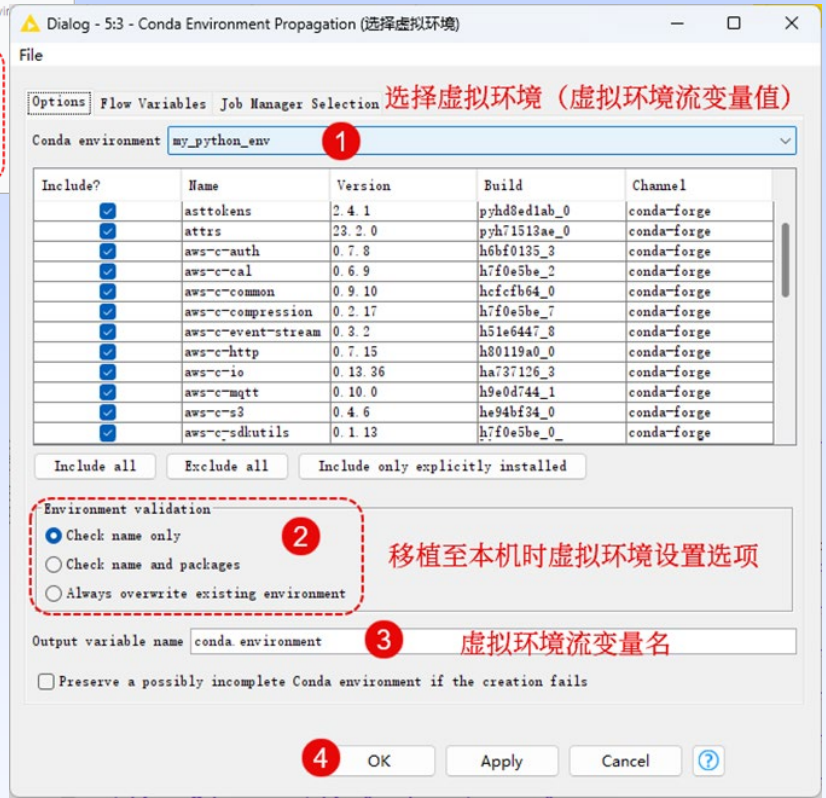
KNIME中使用虚拟环境

向下游Python节点传递虚拟环境变量
conda.enviroment

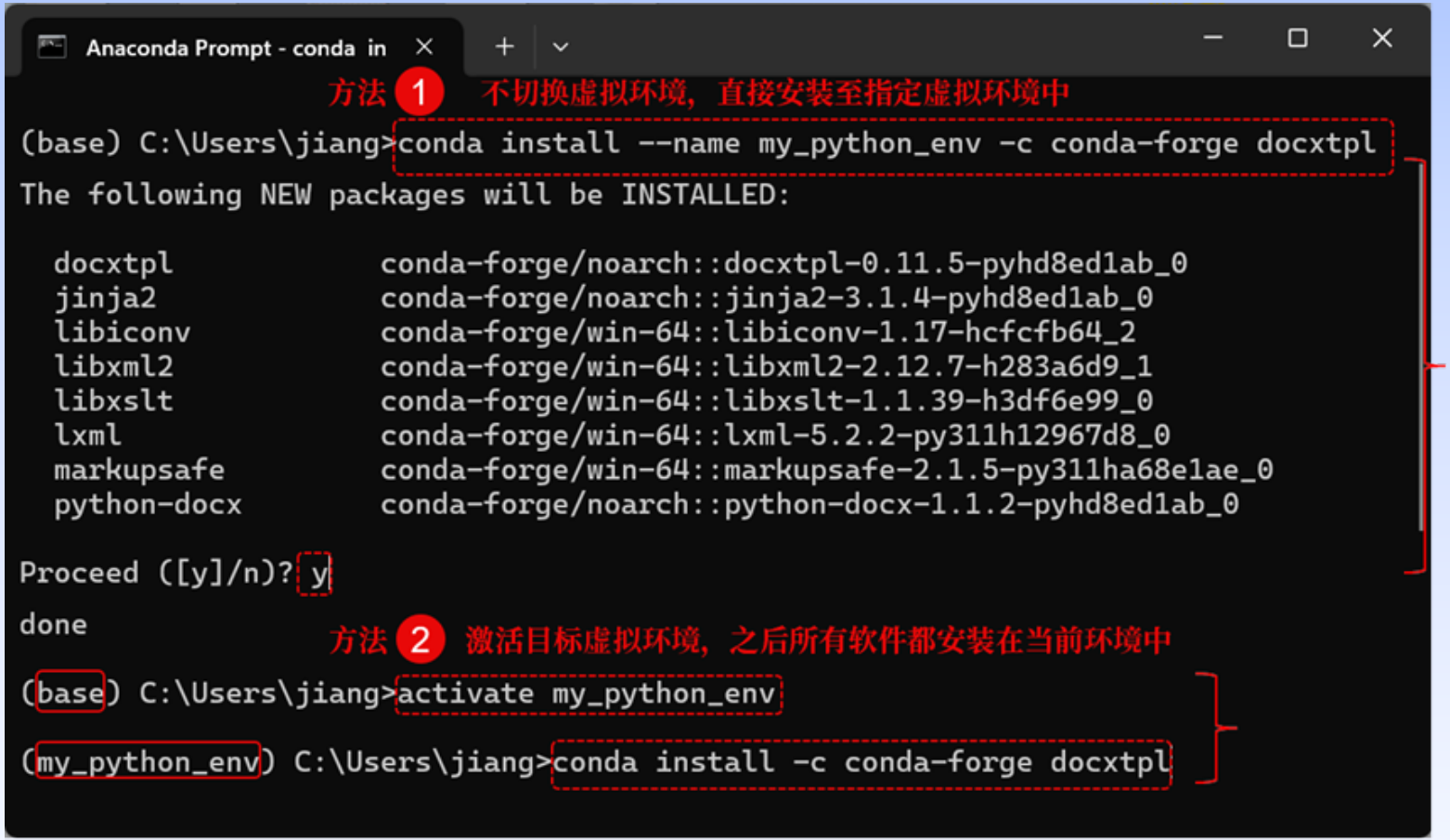




Python Script节点将使用上游节点传递而来的虚拟环境



在自定义虚拟环境中安装软件包



管理自定义虚拟环境

安装扩展

安装扩展KNIME Conda Integration

创建虚拟环境

利用业界广泛使用的Conda工具创建虚拟环境

设置虚拟环境

KNIME中进行必要的设置，如Conda工具路径，默认虚拟环境名

管理虚拟环境

虚拟环境可以方便地进行更新和删除。例如使用安装命令更新项目中的所有依赖包到最新版本，也可以删除不需要的安装包。或者使用命令删除不再需要的虚拟环境。

查看Python版本及各安装包

使用pip命令查看安装包

可以通过在终端或命令提示符中运行pip list命令来获取已安装的Python包及其版本信息。该命令将输出一个表格，列出所有已安装的软件包名称和对应的版本，便于进行依赖管理和版本对比。

查看特定包的版本

可以使用pip show命令查看特定包的版本号。例如，输入`pip show numpy`，可以获取numpy包的具体版本信息，包括版本号、日期和编译器信息，有助于确保开发环境的一致性。



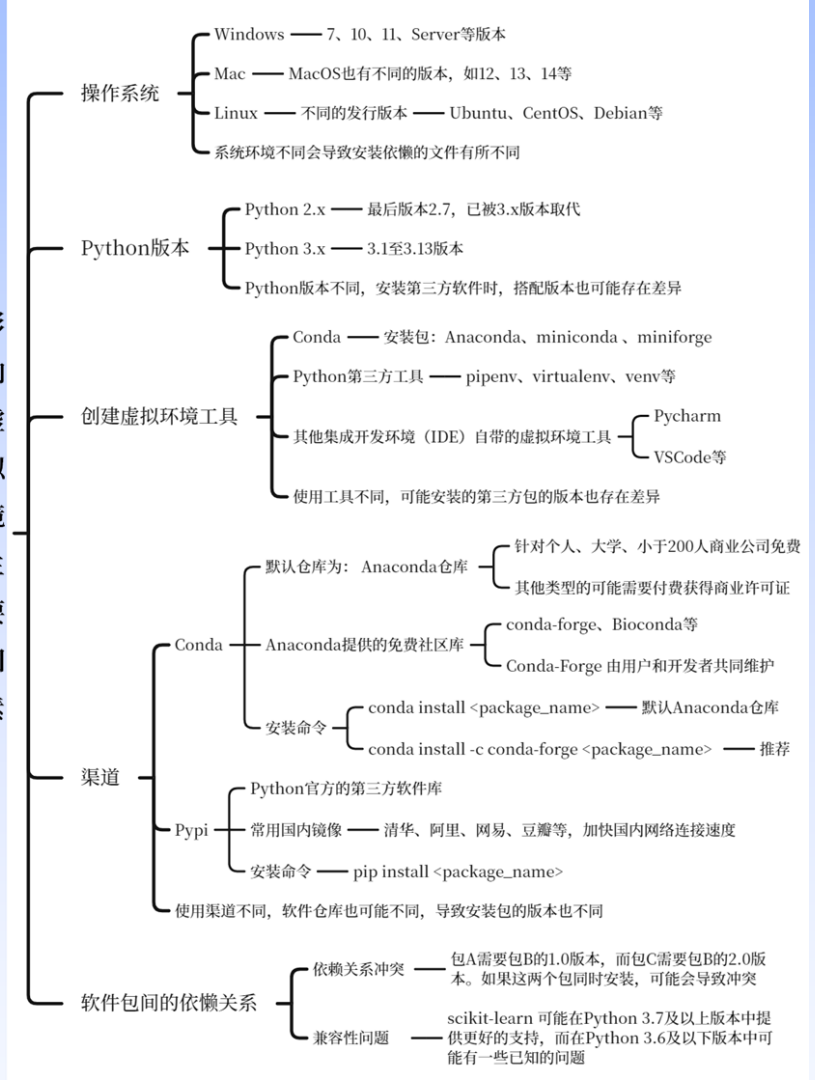
利用虚拟环境管理工具

虚拟环境管理工具如conda和 virtualenv等可以帮助用户隔离和管理不同项目的依赖环境。通过创建和管理虚拟环境，可以确保每个项目依赖的版本一致，便于调试和发布。

检查系统级安装的Python环境

可以通过在终端输入`python -V`或`python --version`来检查当前系统级安装的Python环境和版本。这些命令会输出Python的详细版本信息，有助于确认开发环境的兼容性。

影响虚拟环境诸多因素


影响虚拟环境主要因素
操作系统
Windows —— 7、10、11、Server等版本
Mac —— MacOS也有不同的版本，如12、13、14等
Linux —— 不同的发行版本 —— Ubuntu、CentOS、Debian等
系统环境不同会导致安装依懒的文件有所不同
Python版本
Python 2.x —— 最后版本2.7，已被3.x版本取代
Python 3.x —— 3.1至3.13版本
Python版本不同，安装第三方软件时，搭配版本也可能存在差异
创建虚拟环境工具
Conda —— 安装包：Anaconda、miniconda 、miniforge
Python第三方工具 —— pipenv、virtualenv、venv等
其他集成开发环境（IDE）自带的虚拟环境工具
Pycharm
VSCode等
使用工具不同，可能安装的第三方包的版本也存在差异
渠道
Conda
默认仓库为：Anaconda仓库
针对个人、大学、小于200人商业公司免费
其他类型的可能需要付费获得商业许可证
Anaconda提供的免费社区库
conda-forge、Bioconda等
Conda-Forge 由用户和开发者共同维护
安装命令
conda install <package_name> —— 默认Anaconda仓库
conda install -c conda-forge <package_name> —— 推荐
Pypi
Python官方的第三方软件库
常用国内镜像 —— 清华、阿里、网易、豆瓣等，加快国内网络连接速度
安装命令 —— pip install <package_name>
使用渠道不同，软件仓库也可能不同，导致安装包的版本也不同
软件包间的依懒关系
依赖关系冲突
包A需要包B的1.0版本，而包C需要包B的2.0版本。如果这两个包同时安装，可能会导致冲突
兼容性问题
scikit-learn 可能在Python 3.7及以上版本中提供更好的支持，而在Python 3.6及以下版本中可能有一些已知的问题



03

案例：
Excel数据逐行填充至Word模版

业务需求

Excel数据文件

待填充Word模版文件

A	B	C	D	E	F
主体	事项	年	月	日	律师
张三科技有限公司	侵害商标权	2021	10	1	赵六
李四医药有限公司	不正当竞争	2021	10	2	赵六
王五医药连锁有限公司	侵害商标权及不正当竞争	2021	10	3	周七

1

Excel逐行填充至Word模版

2

授 权 委 托 书（法人）

委托人名称：浙江 XX 制药有限公司

受托人名称：XXX 律师事务所

受托人指派律师：{{律师 }}　　联系电话：123456789

委托人 浙江 XX 制药有限公司 根据《中华人民共和国民事诉讼法》等相关法律法规的规定，特聘请 XXX 律师事务所为本委托人与 {{主体}}{{事项}} 纠纷一案中提供法律服务。XXX 律师事务所接受委托，现指派 {{律师}} 律师作为本案的具体经办人参与本案诉讼以及非诉讼阶段的所有活动。代理人及经办律师的权限为第 2 项：

1、一般代理

2、特别代理

☐代为调查、提供证据　☐代为出庭　☐申请财产保全或证据保全

委托人（盖章）：

{{年}}年 {{月}}月 {{日}}日

Excel数据逐行填充至Word模版

01

准备数据

首先，从Excel文件中读取数据。逐行遍历每一行数据，并将必要信息提取为字典格式，以便于后续操作。

自定义虚拟环境中需要安装软件包docxtpl

02

03

读取Word模板

通过python-docx库读取Word模板文件。分析模板中的变量和内容结构，确保数据填充时能够准确对应各个字段。

04

数据填充

将提取的Excel数据填充到Word模板中对应的变量位置。利用Python脚本控制填充过程，确保每行数据按顺序正确插入到Word文档中。

05

保存并生成文档

填充完成后，保存修改后的Word文档。通过批量生成的方式，将每行记录单独保存为一个Word文件，并使用“编号+公司名+文件类型”的方式作为文件名。

优化与调试

在执行过程中，对代码进行优化以提高运行效率。针对可能出现的错误进行调试，确保数据填充的准确性和文档生成的完整性。

Excel数据逐行填充至Word模版




```python
import pandas as pd
from docxtpl import DocxTemplate
import os

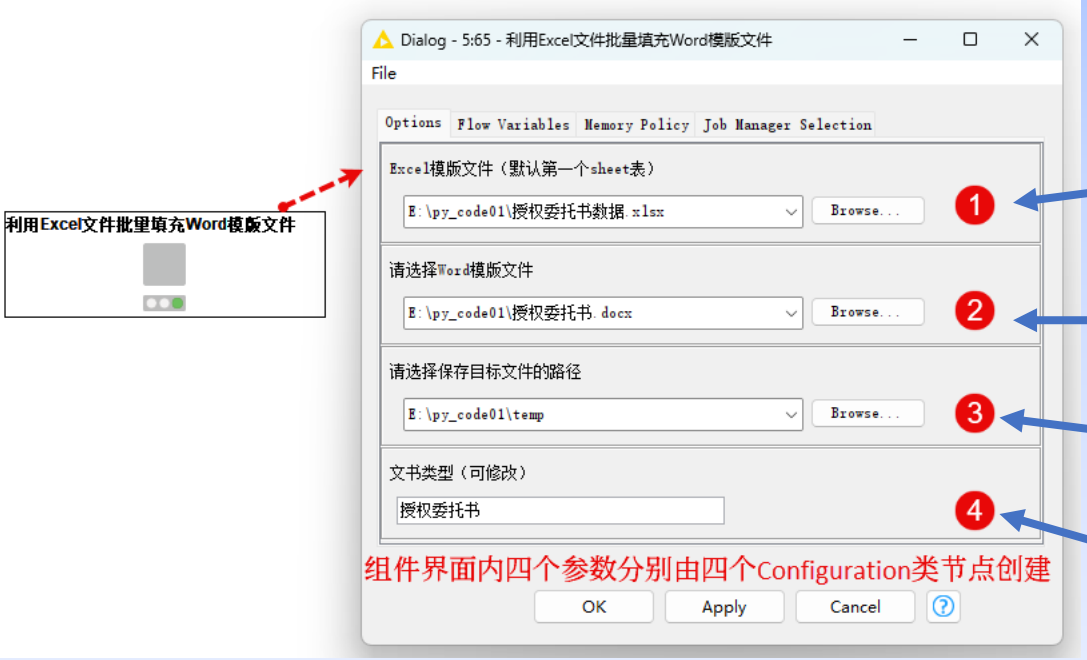
excel_path=r"F:\py_code01\授权委托书数据.xlsx"    # 默认excel文件第一行为表头，第二行为数据
save_path=r"F:\py_code01\save_files"    #保存文件件夹位置
template_path=r"F:\py_code01\授权委托书.docx"   #模板文件，使用占位符{{列名}},

df=pd.read_excel(excel_path)  #默认第一个sheet表
contexts=df.to_dict('records')  # 将数据表转换为字典格式

num=0
for context in contexts:
    num=num+1
    bianhao=str(num).zfill(3)  #文件名最前编号
    tpl=DocxTemplate(template_path)
    tpl.render(context)  #替换模板文件指定处内容
    file="{}_{}_授权委托书.docx"
    file_name=os.path.join(save_path,file)
    tpl.save(file_name.format(bianhao,context["主体"]))     #"主体"为文件名的列
```

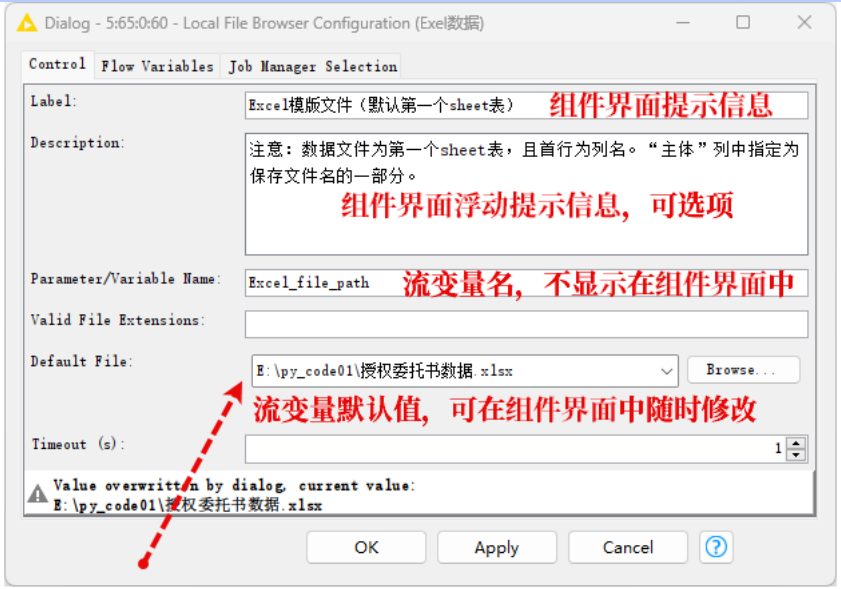
文件名规则：编号_主体 _授权委托书

Excel数据逐行填充至Word模版（优化）

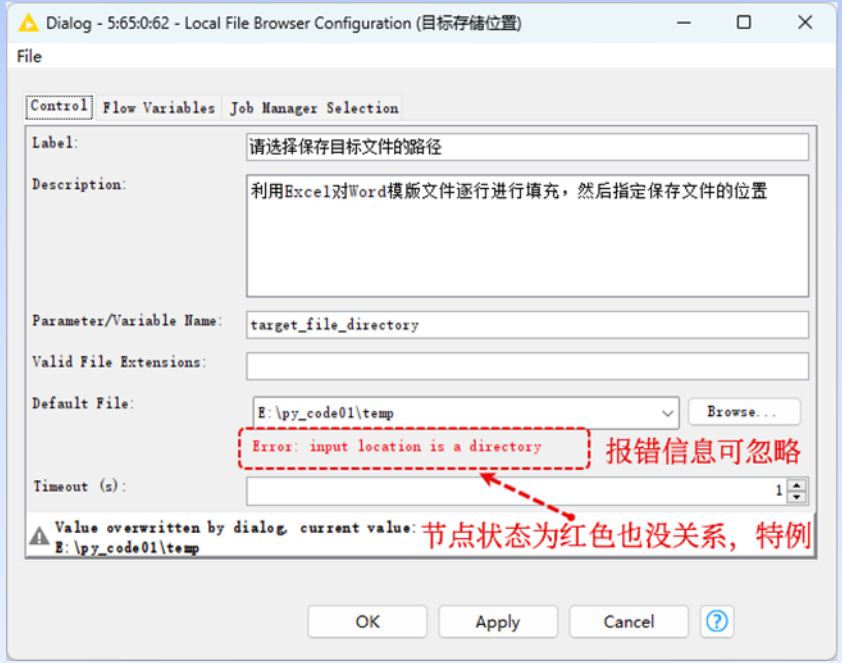


构建图形化界面（详见本书第15章内容）

Excel数据逐行填充至Word模版（优化版）部分节点界面设置



此处只是流变量默认值，封装为组件后，后续只在组件界面中修改参数，而无需再进入本节点修改，用户在对话框内输入的参数将覆盖默认值



Excel数据逐行填充至Word模版（优化版）部分节点界面设置

Dialog - 5:65:0:59 - Python Script (Exce数据逐行填充至)

Python Script

Flow Variables	knio.flow_variables
Excel_file_path (file name)	STRING
Excel_file_path (URL)	STRING
Excel_file_path	STRING
file_name_part	STRING
target_file_directory (file name)	STRING
target_file_directory (URL)	STRING
target_file_directory	STRING
word_file_path (file name)	STRING
word_file_path (URL)	STRING
word_file_path	STRING
knime.workspace	STRING

拖拽流变量至代码中（无需手动输入）

由流变量控制4个参数

微调文件名（修改文件类型）

```
1  import knime.scripting.io as knio
2  import pandas as pd
3  from docxtpl import DocxTemplate
4  import os
5
6  excel_path=knio.flow_variables["Excel_file_path"] # 默认excel文件第一行为表头，第二行为数据
7  template_path=knio.flow_variables["word_file_path"]    #模板文件，使用占位符{{列名}},
8  save_path= knio.flow_variables["target_file_directory"]  #保存文件件夹位置
9  file_name_part=knio.flow_variables["file_name_part"]
10
11
12 df=pd.read_excel(excel_path)  #默认第一个sheet表
13 contexts=df.to_dict('records')  # 将数据表转换为字典格式
14
15 num=0
16 for context in contexts:
17     num=num+1
18     bianhao=str(num).zfill(3)  #文件名最前编号
19     tpl=DocxTemplate(template_path)
20     tpl.render(context)  #替换模板文件指定处内容
21     file="{}_{}_{}.docx"  # 各占位符分别对应：编号-主体（列名）-文件类型
22     file_name=os.path.join(save_path,file)
23     tpl.save(file_name.format(bianhao,context["主体"],file_name_part))  #"主体"为文件名的列
```



04

Pixi创建虚拟环境

创建Pixi虚拟环境

详见本书视频操作

1、安装扩展

安装扩展KNIME Conda Integration

2、安装Pixi

- 在线安装、或下载安装包安装。
- 将安装路径添加到环境变量Path中。
- 关闭当前PowerShell窗口，并重新打开。

初始化项目

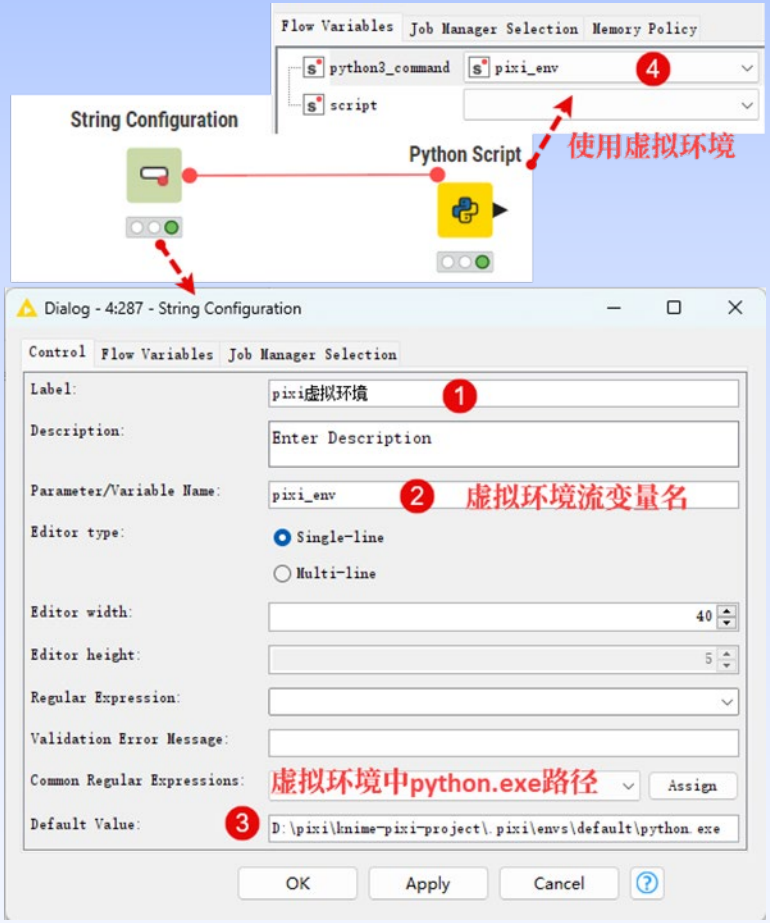
- 创建一个文件夹，使用cd命令切换到该文件夹
- 命令行初始化项目
- 修改配置文件

管理Pixi虚拟环境

若需删除该虚拟环境，直接删除文件夹即可。

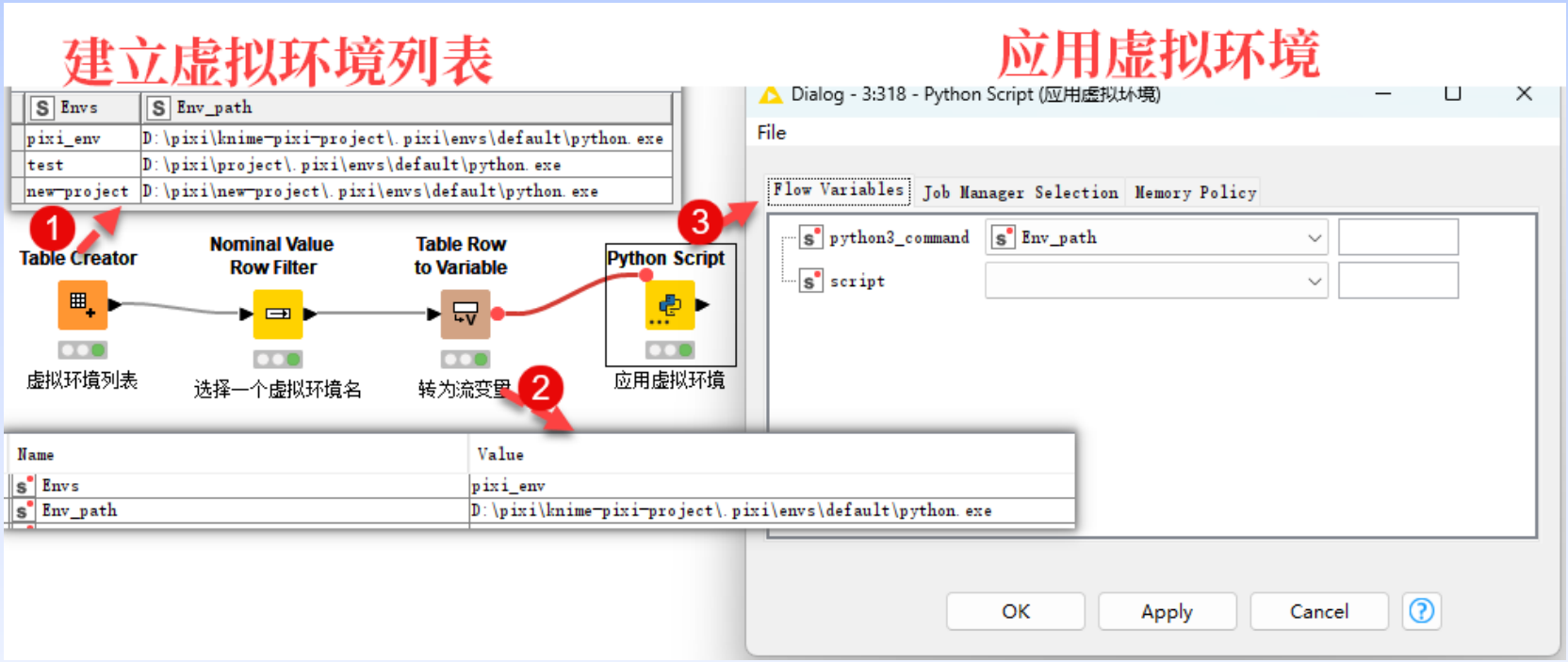
1. 安装包：pixi add scikit-learn matplotlib
2. 删除包：pixi remove matplotlib
3. 查看安装包：pixi list
4. pixi add --pypi <package-name> 从PyPi安装Python包

应用Pixi虚拟环境（方法1：管理单个虚拟环境）



将python.exe路径作为流变量，使用流变量控制Python Script节点所使用的虚拟环境。

应用Pixi虚拟环境（方法2：管理多个虚拟环境）

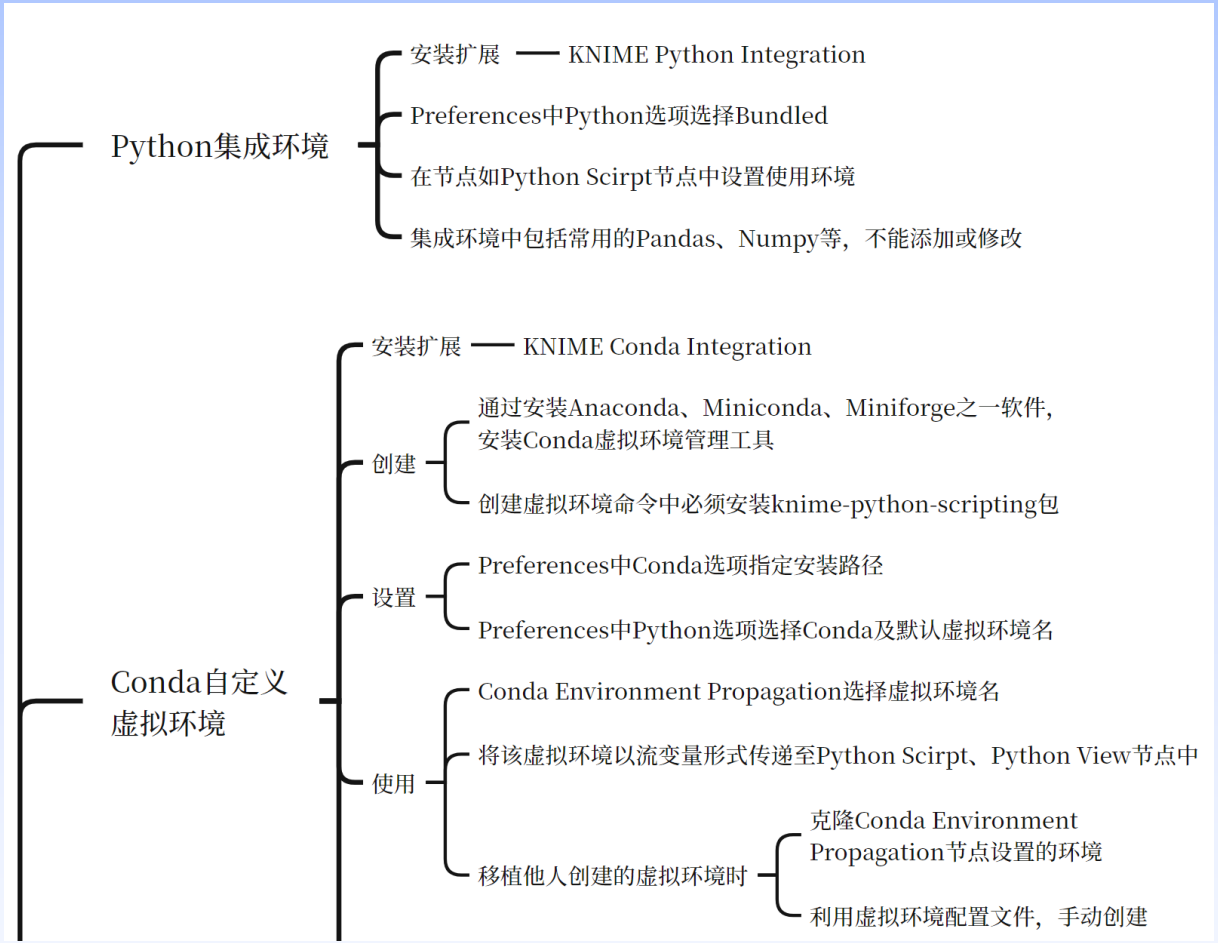




05

本章小结

本章小结


Python集成环境
安装扩展 — KNIME Python Integration
Preferences中Python选项选择Bundled
在节点如Python Scirpt节点中设置使用环境
集成环境中包括常用的Pandas、Numpy等，不能添加或修改
Conda自定义虚拟环境
安装扩展 — KNIME Conda Integration
创建
通过安装Anaconda、Miniconda、Miniforge之一软件，安装Conda虚拟环境管理工具
创建虚拟环境命令中必须安装knime-python-scripting包
设置
Preferences中Conda选项指定安装路径
Preferences中Python选项选择Conda及默认虚拟环境名
使用
Conda Environment Propagation选择虚拟环境名
将该虚拟环境以流变量形式传递至Python Scirpt、Python View节点中
移植他人创建的虚拟环境时
克隆Conda Environment Propagation节点设置的环境
利用虚拟环境配置文件，手动创建

本章小结

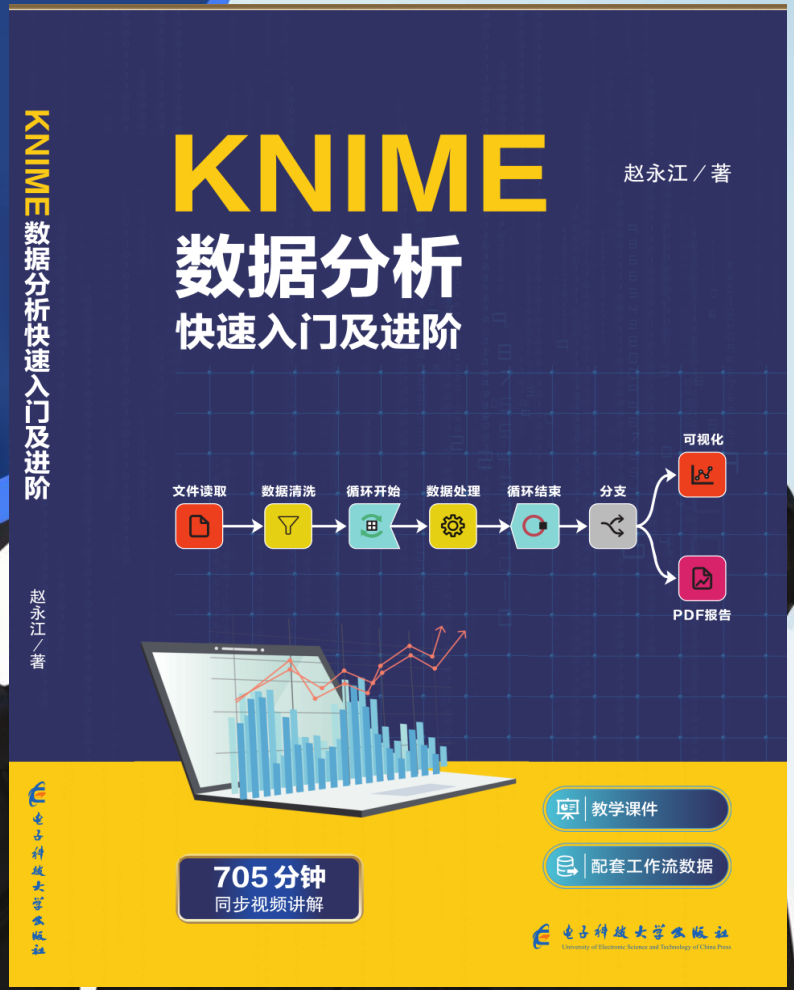
- Pixi自定义虚拟环境
  - 准备 —— 安装Pixi工具、KNIME Python Integration扩展
  - 创建虚拟环境
    - 创建目录，并在此目录执行初始化pixi init命令
    - 修改配置文件pixi.toml中的channels参数
    - 安装KNIME必备包并创建虚拟环境
  - 应用虚拟环境
    - 偏好设置Python环境配置选择Manual，设置路径
    - 将pixi项目路径作为流变量传递至Python Script节点（推荐）
  - 管理虚拟环境 —— 查看、安装、删除包，删除虚拟环境，克隆虚拟环境
- Python环境最佳实践（偏好设置）
  - 默认选择Bundled —— 方便使用Python集成环境
  - Conda —— 不选，可通过Conda Enviroment Propagation节点选择
  - Manual —— 不选，利用pixi项目路径流变量，选择pixi创建的任意虚拟环境
- 影响虚拟环境因素
  - 操作系统、Python版本、虚拟环境管理工具、渠道、软件包间的关系
  - 五个方面的影响因素详见图17-17总结
  - 注意conda工具选择Anaconda仓库时可能涉及的法律合规问题
- 案例应用
  - 设置自定义虚拟环境
  - 虚拟环境中安装第三方软件 —— conda install -c conda-forge docxtpl
  - 利用Excel数据逐行填充Word模版文件 —— 批量生成授权委托书、合同等

感谢观看

THANKS



第十八章
实战案例与技巧分享



目录

01 18.1案例：计算连续工作日

02 18.2案例：机房基站全部故障识别及持续时间统计

03 18.3案例：串标围标风险识别

04 18.4案例：债券信用评级起始时间段计算

05 18.5案例：领料单数据汇总

06 总结及思考

案例学习

01 业务需求

业务需求是业务分析的起点和目标。若经过业务理解步骤后，发现数据不支持该业务需求，此时应重新收集数据，或改变业务需求目标。

02 数据处理及分析

基于业务逻辑，对数据进行必要的清洗和数据预处理，加工成所需要的结果。

常用数据处理过程中大多使用Manipulation类节点，务必熟练掌握。必要时使用循环、分支、或少量的Python代码简化处理过程，提高效率。

02 数据理解

基于业务目标，对获得的数据进行理解，如进行探索性分析，数据表是原始明细表，还是加工汇总表等等。这些往往需要花费较多的时间，一旦数据理解清楚了，且数据能够支持业务目标，那就可以进行正式的数据预处和分析了。

03 解题思路

通过工作流完成一个业务需求的分析，再思考下有无更好的方法，或者有无可以优化的步骤等。学习常见问题的处理思路，积累的小技巧越多，思路也会越开阔。除了关注节点的细节，还应宏观上学习工作流的解题思路。

本章六个案例，可从上述四个方面思考，具体过程请参见各案例对应的工作流。



01

18.1案例：计算连续工作日

18.1案例：计算连续工作日

表18-1 订单时效表

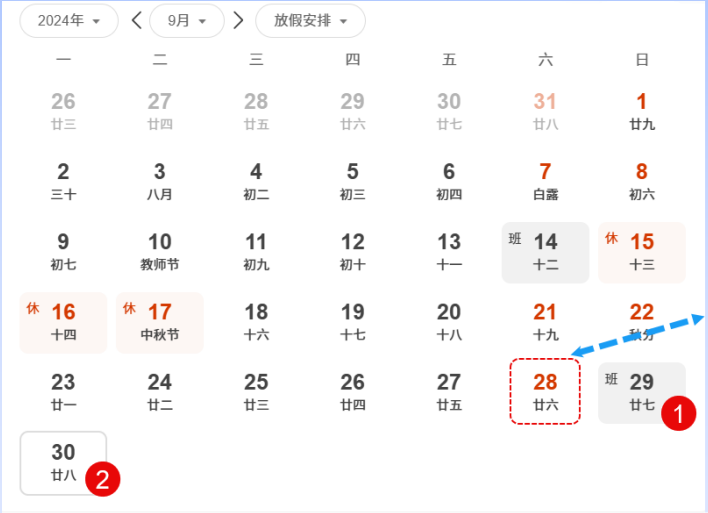
订单日	时效工作日_天	最后发货日
2024-03-25	15	
2024-08-26	12	
2024-04-27	20	
2024-09-28	6	
2024-10-29	8	
2024-05-30	21	
2024-08-21	9	
2024-11-01	9	
2024-04-02	10	
2024-07-03	13	



如何求最后发货日?

区分：自然日、工作日

18.1案例：计算连续工作日—关键思路（将日期编码）



188+6=194

2024-09-24	3	非周末	工作日	185
2024-09-25	4	非周末	工作日	186
2024-09-26	5	非周末	工作日	187
2024-09-27	6	非周末	工作日	188
2024-09-28	7	周末	非工作日	188
2024-09-29	1	国庆节调...	工作日	189
2024-09-30	2	非周末	工作日	190
2024-10-01	3	国庆节放假	非工作日	190
2024-10-02	4	国庆节放假	非工作日	190
2024-10-03	5	国庆节放假	非工作日	190
2024-10-04	6	国庆节放假	非工作日	190
2024-10-05	7	国庆节放假	非工作日	190
2024-10-06	1	国庆节放假	非工作日	190
2024-10-07	2	国庆节放假	非工作日	190
2024-10-08	3	非周末	工作日	191
2024-10-09	4	非周末	工作日	192
2024-10-10	5	非周末	工作日	193
2024-10-11	6	非周末	工作日	194
2024-10-12	7	国庆节调...	工作日	195
2024-10-13	1	周末	非工作日	195
2024-10-14	2	非周末	工作日	196



02

18.2案例：机房基站全部故障识别及持续时间统计

18.2案例：机房基站全部故障识别及持续时间统计

地市	机房名称	所属设备名称	基站名称	发生时间	恢复时间	告警主题
亳州	亳州A机房	COZ-设备B	COZ-基站2	2023/4/6 11:42	2023/4/6 11:43	连接中断
亳州	亳州A机房	COZ-设备C	COZ-基站3	2023/4/6 11:41	2023/4/6 11:43	连接中断
亳州	亳州A机房	COZ-设备A	COZ-基站1	2023/4/6 10:53	2023/4/6 12:08	连接中断
亳州	亳州A机房	COZ-设备C	COZ-基站3	2023/4/19 12:37	2023/4/19 12:39	连接中断
宿州	宿州B机房	BUZ-设备2	BUZ基站2	2023/4/6 11:37	2023/4/6 11:41	连接中断
宿州	宿州B机房	BUZ-设备2	BUZ基站2	2023/4/6 20:20	2023/4/6 20:24	连接中断
宿州	宿州B机房	BUZ-设备3	BUZ基站3	2023/4/21 10:59	2023/4/21 11:01	连接中断
宿州	宿州B机房	BUZ-设备3	BUZ基站3	2023/4/21 11:02	2023/4/21 11:03	连接中断
宿州	宿州B机房	BUZ-设备1	BUZ基站1	2023/4/21 10:58	2023/4/21 11:05	连接中断
宿州	宿州B机房	BUZ-设备1	BUZ基站1	2023/4/21 11:05	2023/4/21 11:07	连接中断
滁州	滁州C机房	ABC-设备1	ABC基站1	2023/4/28 7:07	2023/4/28 9:47	连接中断
滁州	滁州C机房	ABC-设备5	ABC基站5	2023/4/28 7:09	2023/4/28 9:52	连接中断
滁州	滁州C机房	ABC-设备3	ABC基站3	2023/4/28 7:08	2023/4/28 9:52	连接中断
滁州	滁州C机房	ABC-设备2	ABC基站2	2023/4/28 7:08	2023/4/28 9:52	连接中断
滁州	滁州C机房	ABC-设备4	ABC基站4	2023/4/28 7:09	2023/4/28 9:53	连接中断
淮北	淮北D机房	HBS-设备1	HBS基站1	2023/4/27 0:51	2023/4/27 0:53	连接中断
淮北	淮北D机房	HBS-设备2	HBS基站2	2023/4/27 0:49	2023/4/27 0:51	连接中断
淮北	淮北D机房	HBS-设备3	HBS基站3	2023/4/27 0:51	2023/4/27 0:54	连接中断
淮北	淮北D机房	HBS-设备4	HBS基站4	2023/4/27 0:51	2023/4/27 0:53	连接中断
淮北	淮北F机房	HB-设备1	HB-基站1	2023/4/27 0:43	2023/4/27 8:46	连接中断
淮北	淮北F机房	HB-设备2	HB-基站2	2023/4/27 0:44	2023/4/27 8:46	连接中断
淮北	淮北F机房	HB-设备3	HB-基站3	2023/4/27 0:45	2023/4/27 8:46	连接中断
南京	南京E机房	ABC-设备1	基站1	2024/6/11 0:01	2024/6/11 5:01	连接中断
南京	南京E机房	ABC-设备2	基站2	2024/6/11 0:03	2024/6/11 3:01	连接中断
南京	南京E机房	ABC-设备3	基站3	2024/6/11 0:04	2024/6/11 4:31	连接中断
南京	南京E机房	ABC-设备4	基站4	2024/6/11 0:06	2024/6/11 5:11	连接中断
南京	南京E机房	ABC-设备5	基站5	2024/6/11 1:01	2024/6/11 4:21	连接中断

各基站故障明细表

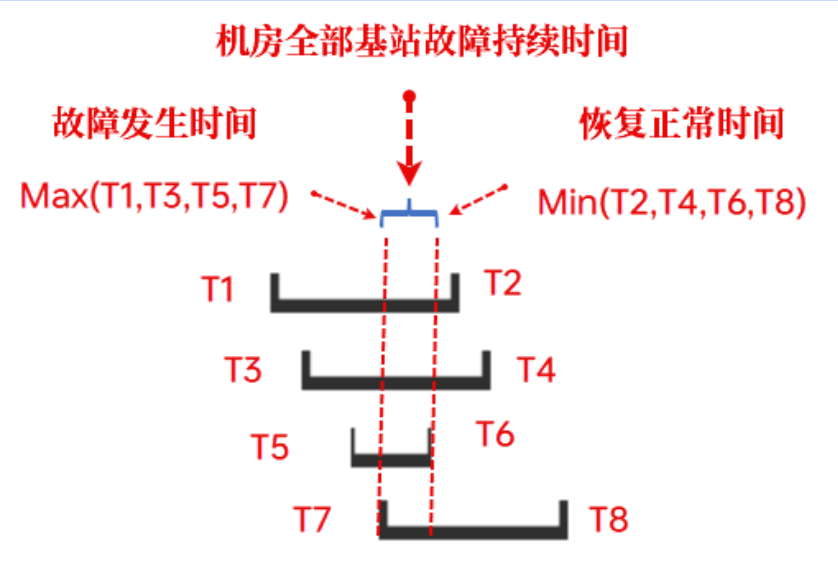
18.2案例：机房基站全部故障识别及持续时间统计

各市机房对应基站统计表

地市	机房名称	基站个数
亳州	亳州A机房	3
亳州	亳州B机房	2
亳州	亳州C机房	5
宿州	宿州A机房	4
宿州	宿州B机房	3
南京	南京C机房	2
南京	南京D机房	3
南京	南京E机房	5
淮北	淮北D机房	4
淮北	淮北F机房	3
滁州	滁州C机房	5

18.2案例：机房基站全部故障识别及持续时间统计

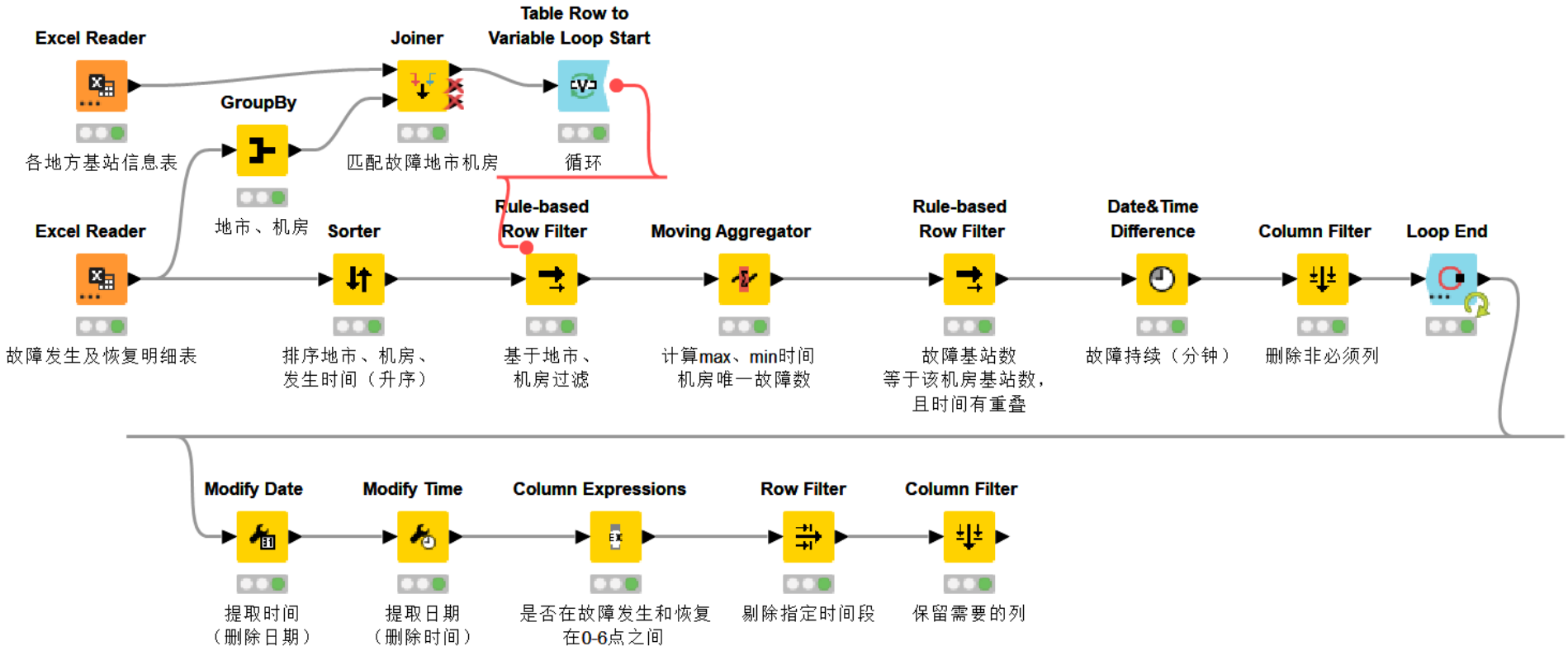
计算四个基站全部故障的持续时间



故障发生时间 < 恢复正常时间

解题思路：找所有基站存在共同故障的时间段。

18.2案例：机房基站全部故障识别及持续时间统计





03

18.3案例：串标围标风险识别

18.3案例：串标围标风险识别

目标：

审查竞标厂商可能存在一起竞标，轮流中标的情况。

手段：

若在多个项目中，存在某固定组合的两个或三个厂商频繁竞标多个项目，且其中该组合的某个厂商中标或轮流中标，则可能存在较大的串标和围标风险。对于这些可疑情况，可作为审计线索进行重点审查，如各厂商之间有无关联关系，如公司相互持股、存在共同的股东、公司地址或联系方式相同等。

数据中共有13个厂商，任意两个和三个厂商组合共计C(13,2) +C(13,3) =364，KNIME无直接可用的排列组合节点，因此使用Python组合函数。

采购编号	竞标厂商	中标人
1	A	A
1	B	
1	C	
1	D	
2	D	D
2	E	
2	F	
3	A	B
3	B	
3	C	
4	A	A
4	B	
4	C	
5	M	
5	N	N
5	O	
6	P	P
6	Q	
6	B	
7	P	
7	A	
7	B	C
7	C	
8	A	B
8	B	
8	C	
9	Y	
9	Z	Z
9	A	
10	A	B
10	B	
10	C	

18.3案例：串标围标风险识别

```python
import knime.scripting.io as knio
import itertools
import pandas as pd

df=knio.input_tables[0].to_pandas()  #knime数据表转df
# 计算竞标厂商中2个和3个组合
two_combinations = list(itertools.combinations(df["竞标厂商"], 2))
three_combinations = list(itertools.combinations(df["竞标厂商"], 3))
result= two_combinations+three_combinations
df_new=pd.DataFrame({"all_combinations":result}) # 字典表转df
df_new["num"]=[len(i) for i in df_new["all_combinations"]]  # 厂商组合个数
df_new["num"] = df_new["num"].astype("int32")          # 转为短整型
# 厂商组合字符串形式，为中标人是否属于该厂商组合做准备
df_new["supplier_str"]=[(",".join(i)) for i in df_new["all_combinations"]]
knio.output_tables[0] =knio.Table.from_pandas(df_new)  # df转knime数据表
```

Table - 5:3 - Python Script (所有竞标厂商的)

File Edit Hilite Navigation View

Table "default" – Rows: 364 Spec – Columns: 3 Properties Flow Variables

Row ID	all_combinations	num	supplier_str
Row68	[O, P]	2	O, P
Row69	[O, Q]	2	O, Q
Row70	[O, Y]	2	O, Y
Row71	[O, Z]	2	O, Z
Row72	[P, Q]	2	P, Q
Row73	[P, Y]	2	P, Y
Row74	[P, Z]	2	P, Z
Row75	[Q, Y]	2	Q, Y
Row76	[Q, Z]	2	Q, Z
Row77	[Y, Z]	2	Y, Z
Row78	[A, B, C]	3	A, B, C
Row79	[A, B, D]	3	A, B, D
Row80	[A, B, E]	3	A, B, E
Row81	[A, B, F]	3	A, B, F
Row82	[A, B, M]	3	A, B, M
Row83	[A, B, N]	3	A, B, N
Row84	[A, B, O]	3	A, B, O
Row85	[A, B, P]	3	A, B, P
Row86	[A, B, Q]	3	A, B, Q
Row87	[A, B, Y]	3	A, B, Y
Row88	[A, B, Z]	3	A, B, Z
Row89	[A, C, D]	3	A, C, D
Row90	[A, C, E]	3	A, C, E
Row91	[A, C, F]	3	A, C, F
Row92	[A, C, M]	3	A, C, M
Row93	[A, C, N]	3	A, C, N
Row94	[A, C, O]	3	A, C, O
Row95	[A, C, P]	3	A, C, P
Row96	[A, C, Q]	3	A, C, Q
Row97	[A, C, Y]	3	A, C, Y
Row98	[A, C, Z]	3	A, C, Z
Row99	[A, D, E]	3	A, D, E

目标：

- 利用Python组合函数，求出13个供应商中2个、3个的厂商的所有组合。
- 找出这样的组合，且属于其中之一中标厂商共参与的采购项目数，参与的越多，潜在串标围标风险越大

18.3案例：串标围标风险识别



all_c...	num	suppl...
[A, B]	2	A, B
[A, C]	2	A, C
[A, D]	2	A, D
[A, E]	2	A, E

采购编号	Uniqu...	First...	Uniqu...	Sorte...	is_exist
1	A, B	A	2	[A, B]	0
10	A, B	B	2	[A, B]	3
3	A, B	B	2	[A, B]	3
4	A, B	A	2	[A, B]	0
8	A, B	B	2	[A, B]	3

采购编号	竞标厂商	中标人
1	A	A
1	B	?
1	C	?
1	D	?
2	D	D
2	E	?
2	F	?

采购编号	竞标厂商	中标人
1	A	A
1	B	A
3	A	B
3	B	B
4	A	A
4	B	A
6	B	P

采购编号	Uniqu...	First...	Uniqu...	Sorte...
1	A, B	A	2	[A, B]
10	A, B	B	2	[A, B]
3	A, B	B	2	[A, B]
4	A, B	A	2	[A, B]
6	B	P	1	[B]
7	B, A	C	2	[A, B]
8	A, B	B	2	[A, B]
9	A	C	1	[A]

Sorte...	Uniqu...	Concate...	Concat...
[A, B]	5	1, 10, 3, 4, 8	A, B, B, A, B



04

18.4案例：债券信用评级起始时间段计算

18.4案例：债券信用评级起始时间段计算

公募债券评级跟踪表

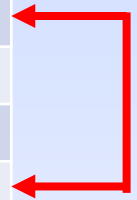
债券号	债券名称	债券评级	评级机构	评级时间
001	23公募债券1	AAA	信评机构A	2021-05-01
001	23公募债券1	AAA	信评机构A	2021-10-02
001	23公募债券1	AA	信评机构B	2022-02-03
001	23公募债券1	AA	信评机构B	2022-06-04
001	23公募债券1	AAA	信评机构C	2023-05-05
002	24公募债券2	BBB	信评机构A	2021-03-17
002	24公募债券2	A	信评机构A	2021-09-08
002	24公募债券2	A	信评机构D	2022-04-09
002	24公募债券2	AA	信评机构D	2023-05-10

金融市场中，不同信用评级机构的表示方式可能差异。以投资市场较为通用的评级，从高到底，依次为AAA、AA、A、BBB、BB、B、CCC、CC、C、D。有时评级也会使用BB+或BB-的形式进行微调

18.4案例：债券信用评级起始时间段计算

债券评级结果表

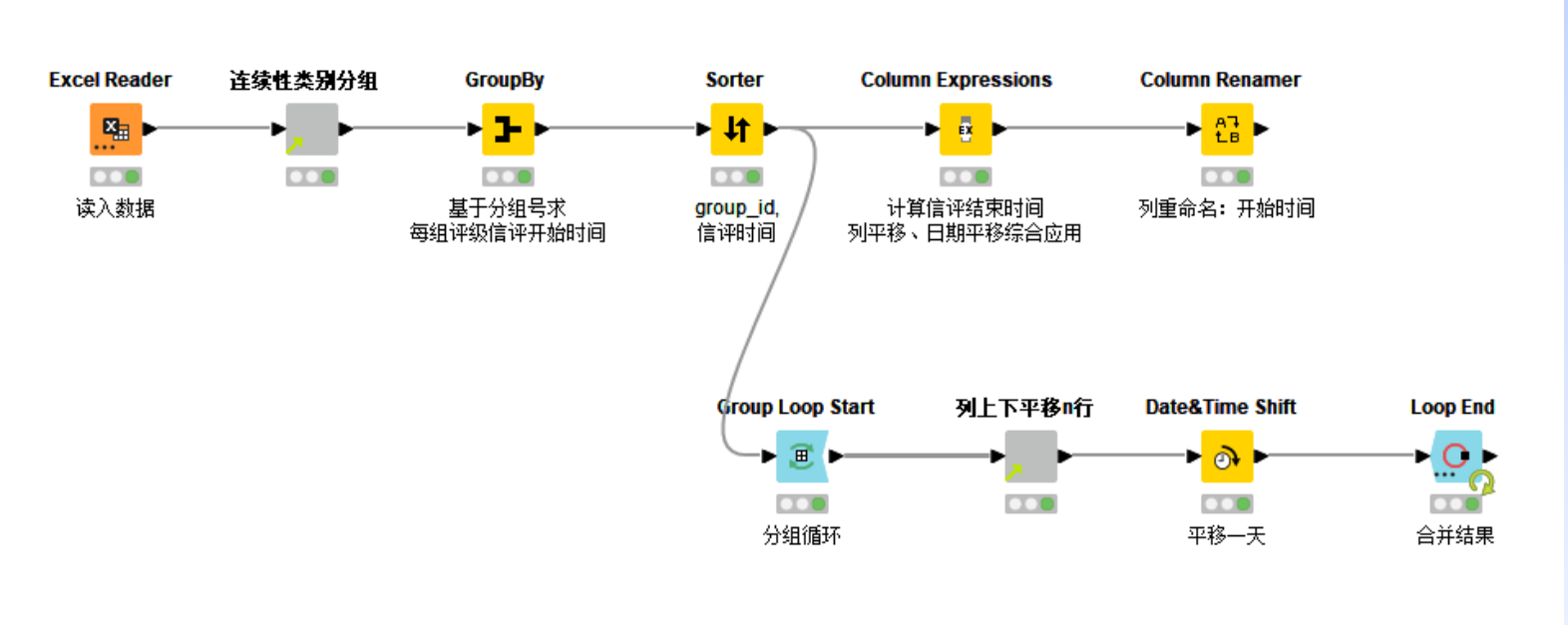
债券号	债券名称	债券评级	开始日期	结束日期
001	23公募债券1	AAA	2021-05-01	2022-02-02
001	23公募债券1	AA	2022-02-03	2023-05-04
001	23公募债券1	AAA	2023-05-05	2024-06-14
002	24公募债券2	BBB	2021-03-17	2021-09-07
002	24公募债券2	A	2021-09-08	2023-05-09
002	24公募债券2	AA	2023-05-10	2024-06-14



此处为缺失值，使用工作流运行时的日期2024-06-14进行填充

18.4案例：债券信用评级起始时间段计算

两种不同的方法





05

18.5案例：领料单数据汇总

18.5案例：领料单数据汇总

原始数据

需求结果

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	材料编号	材料名称	规格	计量单位	数量请领	数量实发	金额单价	金额金额	领料部门	领料日期	流水号	主管	仓管主管	领料	发料
2	100010	材料A	50*40	吨	10	10	2000	20000	生产一部	2024 年06月 01 日	202406-0001	张三	李四	王五	李四
3	310002	材料B	400/pcs	箱	20	20	1500	30000	生产一部	2024 年06月 01 日	202406-0001	张三	李四	王五	李四
4	410003	材料C	35*35	个	100	100	30	3000	生产一部	2024 年06月 01 日	202406-0001	张三	李四	王五	李四
5	100010	材料D	55*45	吨	15	12	2000	24000	生产一部	2024 年06月 02日	202406-0002	张三	李四	王五	李四
6	310002	材料B	400/pcs	箱	10	10	1500	15000	生产一部	2024 年06月 02日	202406-0002	张三	李四	王五	李四
7	100010	材料A	50*40	吨	10	10	2000	2000	生产一部	2024 年04月5日	202406-0003	张三	李四	王五	李四
8	310002	材料B	400/pcs	箱	15	15	1500	22500	生产一部	2024 年04月5日	202406-0003	张三	李四	王五	李四
9	410003	材料C	35*35	个	100	100	30	3000	生产一部	2024 年04月5日	202406-0003	张三	李四	王五	李四
10	230035	材料E	500/pcs	卷	5	4	1000	4000	生产一部	2024 年04月5日	202406-0003	张三	李四	王五	李四

领料单

领料部门：生产一部　　2024 年06月 01 日

流水号：202406-0001

材料编号	材料名称	规格	计量单位	数量		金额	
				请领	实发	单价	金额
100010	材料A	50*40	吨	10	10	2000.00	20000.00
310002	材料B	400/pcs	箱	20	20	1500.00	30000.00
410003	材料C	35*35	个	100	100	30.00	3000.00
合　计							53000

主管：张三　　仓管主管：李四　　领料：王五　　发料：李四

领料单

领料部门：生产一部　　2024 年06月 02日

流水号：202406-0002

材料编号	材料名称	规格	计量单位	数量		金额	
				请领	实发	单价	金额
100010	材料D	55*45	吨	15	12	2000.00	24000.00
310002	材料B	400/pcs	箱	10	10	1500.00	15000.00
合　计							39000

主管：张三　　仓管主管：李四　　领料：王五　　发料：李四

领料单

领料部门：生产一部　　2024 年04月5日

流水号：202406-0003

材料编号	材料名称	规格	计量单位	数量		金额	
				请领	实发	单价	金额
100010	材料A	50*40	吨	10	10	2000.00	2000.00
310002	材料B	400/pcs	箱	15	15	1500.00	22500.00
410003	材料C	35*35	个	100	100	30.00	3000.00
230035	材料E	500/pcs	卷	5	4	1000.00	4000.00
合　计							31500

主管：张三　　仓管主管：李四　　领料：王五　　发料：李四

18.5案例：领料单数据汇总---找规律，各个击破

领料单数据表结构解析

领料单

领料部门：生产一部　　2024 年06月 01 日　← 需添加“领料日期”作为列名　　① ← 第一行数据

流水号：　202406-0001　　② ← 第二行数据

材料编号	材料名称	规格	计量单位	数量		金额	
				请领	实发	单价	金额
100010	材料A	50*40	吨	10	10	2000.00	20000.00
310002	材料B	400/pcs	箱	20	20	1500.00	30000.00
410003	材料C	35*35	个	100	100	30.00	3000.00
合　　计							53000

主管：张三　　仓管主管：李四　　领料：王五　　发料：李四　　③ ← 最后一行数据

利用关键词“领料部门”定位读取数据开始位置

利用关键词“主管”定位读取数据结束位置

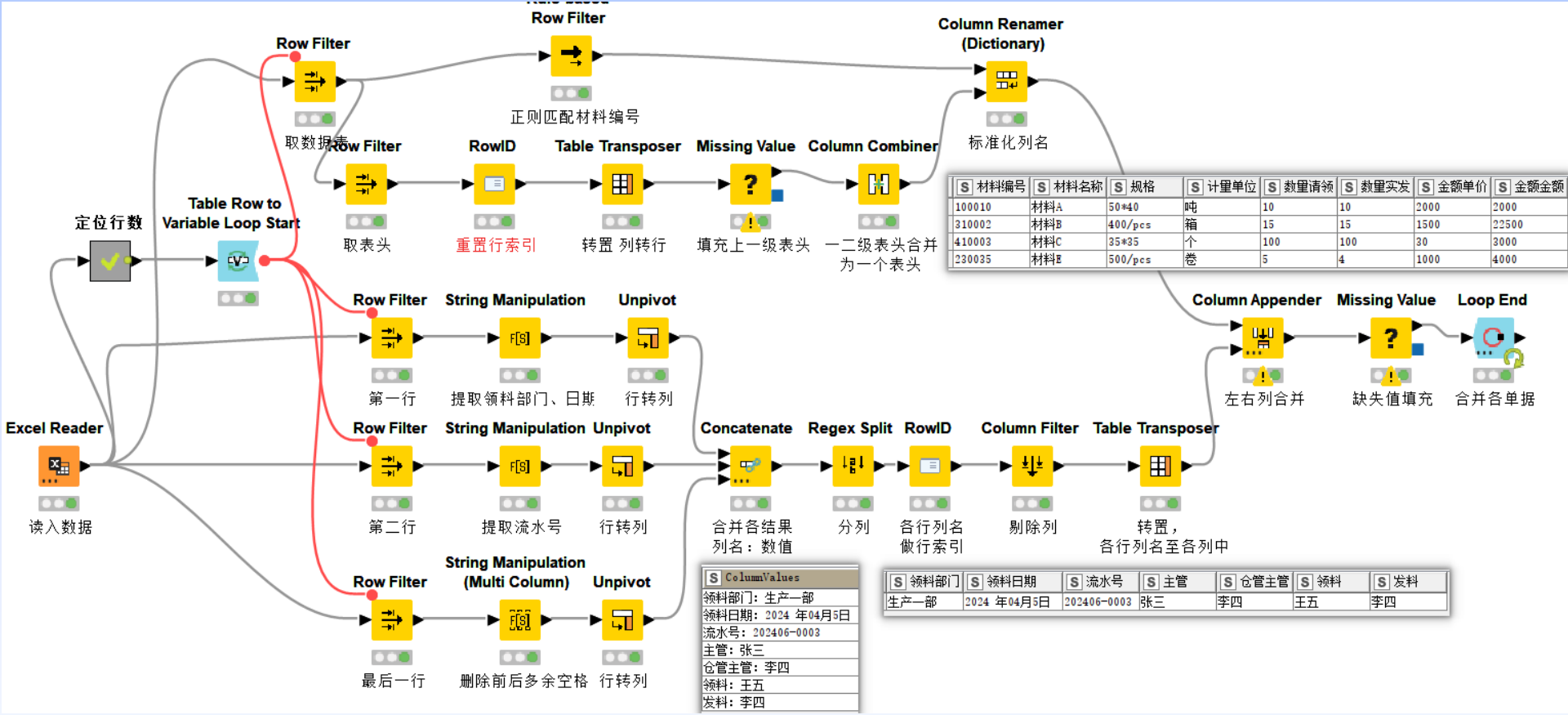
多层级表头 ④

主要数据 ⑤

无效数据

- 共计5个部分，各个击破
- 变与不变：只有编号5中行数多少不确定，其他位置都是有规律，相对确定的

18.5案例：领料单数据汇总---工作流



材料编号	材料名称	规格	计量单位	数量请领	数量实发	金额单价	金额金额
100010	材料A	50*40	吨	10	10	2000	2000
310002	材料B	400/pcs	箱	15	15	1500	22500
410003	材料C	35*35	个	100	100	30	3000
230035	材料E	500/pcs	卷	5	4	1000	4000

ColumnValues
领料部门：生产一部
领料日期：2024 年04月5日
流水号：202406-0003
主管：张三
仓管主管：李四
领料：王五
发料：李四

领料部门	领料日期	流水号	主管	仓管主管	领料	发料
生产一部	2024 年04月5日	202406-0003	张三	李四	王五	李四



06

总结及思考

节点学习总结

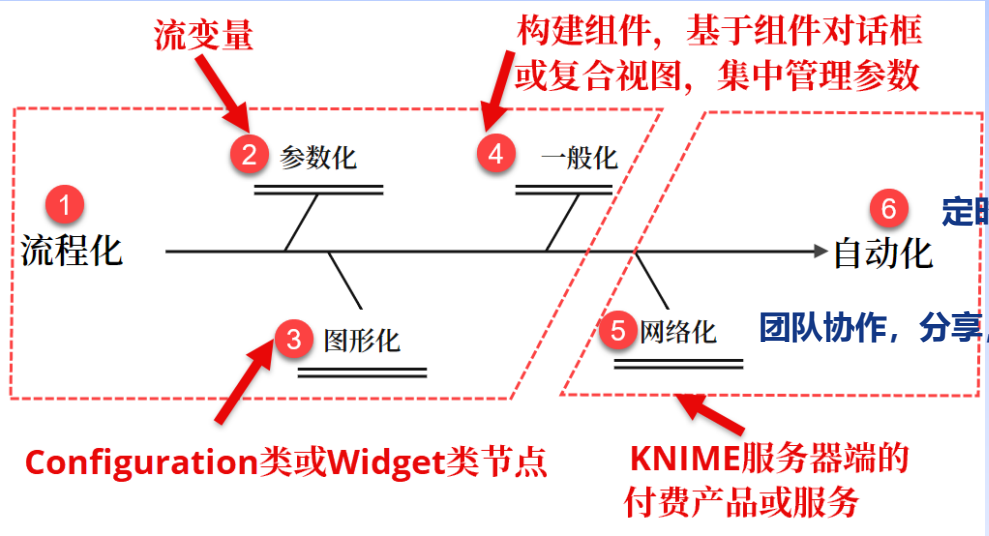


多总结归纳

KNIME工作流框架


数据接入
文件/数据库
接口等
数据融合
多源对齐
合并关联
数据准备
预处理与清洗
转换与计算等
分析与建模
可视化/报告
建模等
交付输出
发布/导出
部署等
静态交付
表
图
图文报告
动态交付
仪表盘
应用
API
模型

KNIME功能的思考与总结



- 相对来说，流程化、参数化、图形化、一般化是基础和重点。
- 网络化和自动化在前者的基础上，更好地提高了效率。由于后者往往涉及商业付费类服务，不在本书范围内。
- 本章18.6案例中，首先构建工作流解决一列数值转为日期的问题（**流程化**），再借助循环和流变量解决多列转换的问题（**参数化**），然后借助Configuration类节点，利用组件集中管理参数（**图形化**），最后实现一个通用的自定义节点（或工作流），解决了这类数值转日期的问题（**一般化**）。

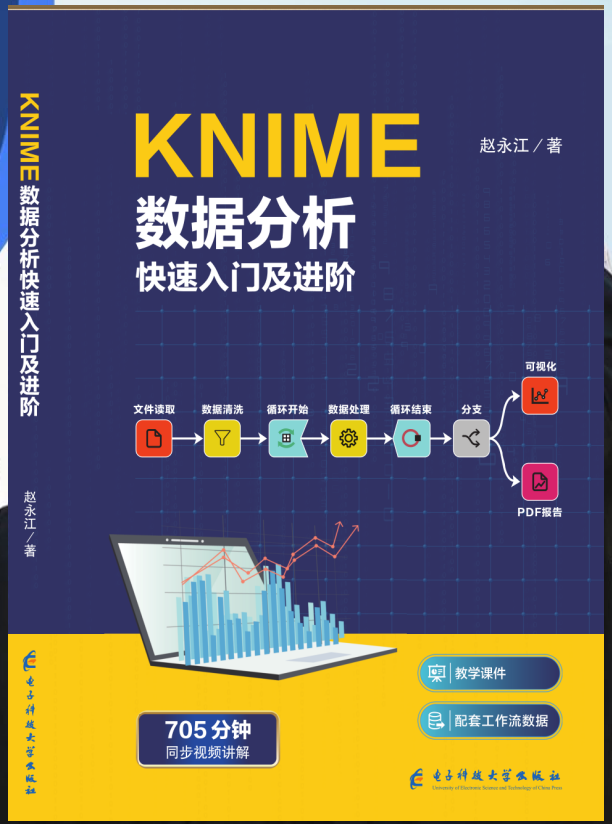
感谢观看

THANKS



结束篇 Bonus

各版本新增节点及比较等



目录

01 5.3、5.4新增常用节点

02 5.3、5.4 更名节点

03 5.3及之前版本与5.4版本差异

04 5.5版本日期时间类函数变化

05 KNIME学习杂谈



01

5.3、5.4新增常用节点

5.3、5.4新增常用节点

新增节点	说明	版本
Example Data Reader	四个数据集，方便学习，测试	5.3
Variable Filter	过滤流变量，避免过多流变量传递下游	5.3
Expression	新构建一个统一的界面，针对字符、数值进行处理，支持IF、Switch逻辑判断	5.3
Variable Expression		5.4
Expression Row Filter		5.4



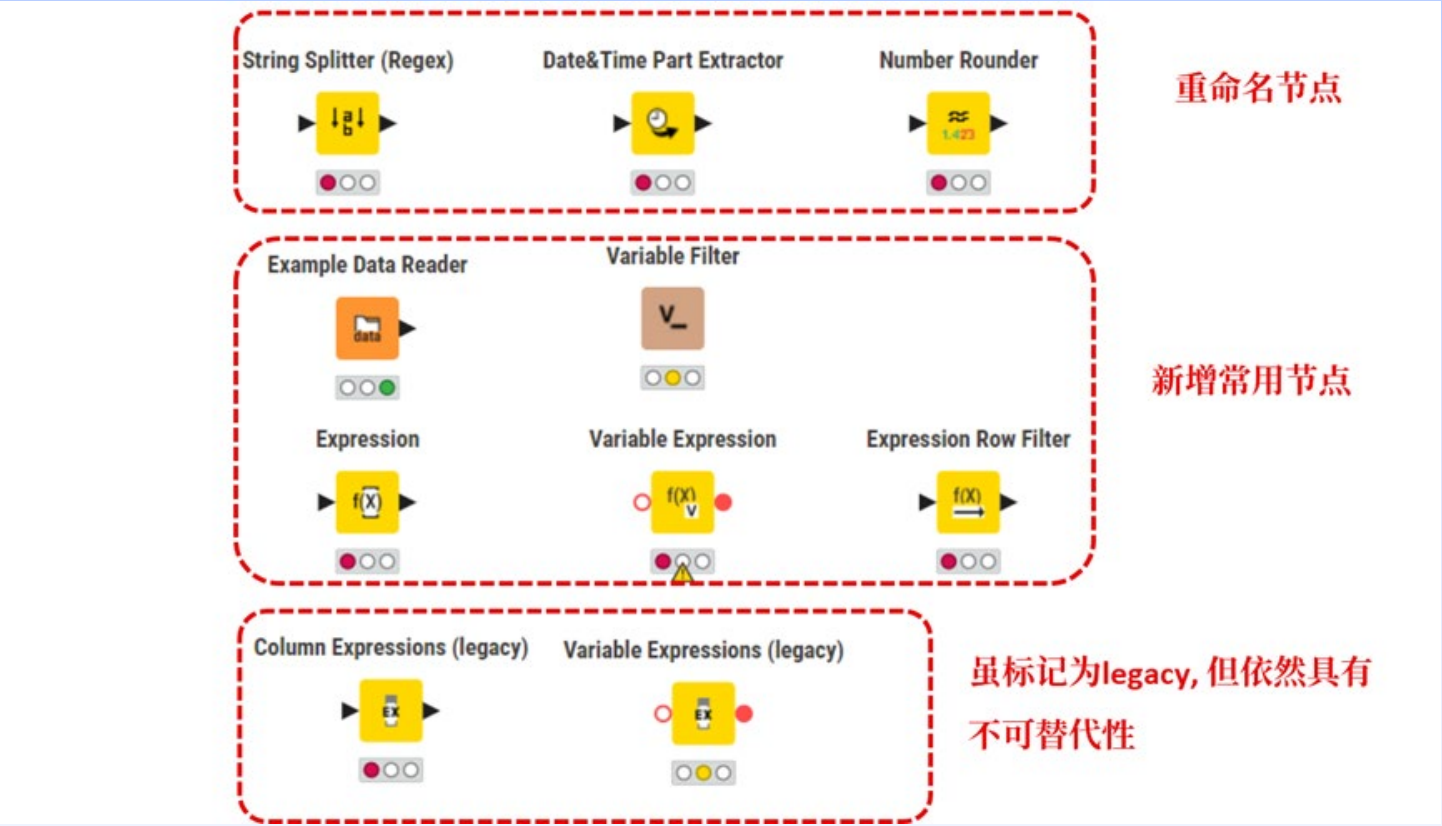
02

5.3、5.4 更名节点

5.3、5.4 更名节点

原节点名	新节点名	版本	说明
Regex Split	String Splitter (Regex)	5.3	增加支持捕获组命名
Round Double	Number Rounder	5.3	调整界面WebUI风格
Extract Date&Time Fields	Date&Time Part Extractor	5.3	调整界面WebUI风格
Column Expressions	Column Expressions (legacy)	5.4	暂无其他节点可完全替代
Variable Expressions	Variable Expressions (legacy)	5.4	暂无其他节点可完全替代

新增节点及重命名节点


String Splitter (Regex)
Date&Time Part Extractor
Number Rounder
重命名节点
Example Data Reader
Variable Filter
Expression
Variable Expression
Expression Row Filter
新增常用节点
Column Expressions (legacy)
Variable Expressions (legacy)
虽标记为legacy, 但依然具有
不可替代性



03

5.3及之前版本与5.4版本差异

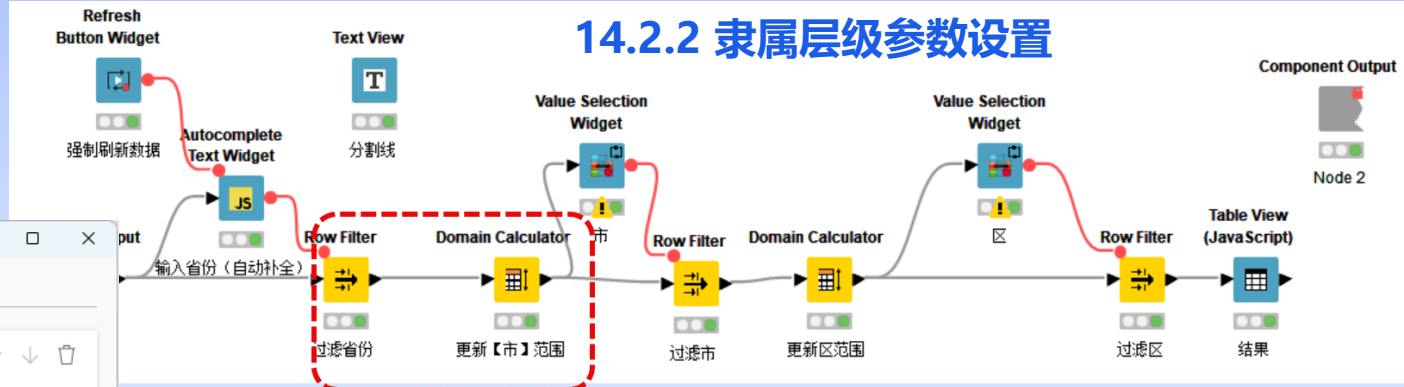
版本差异比较

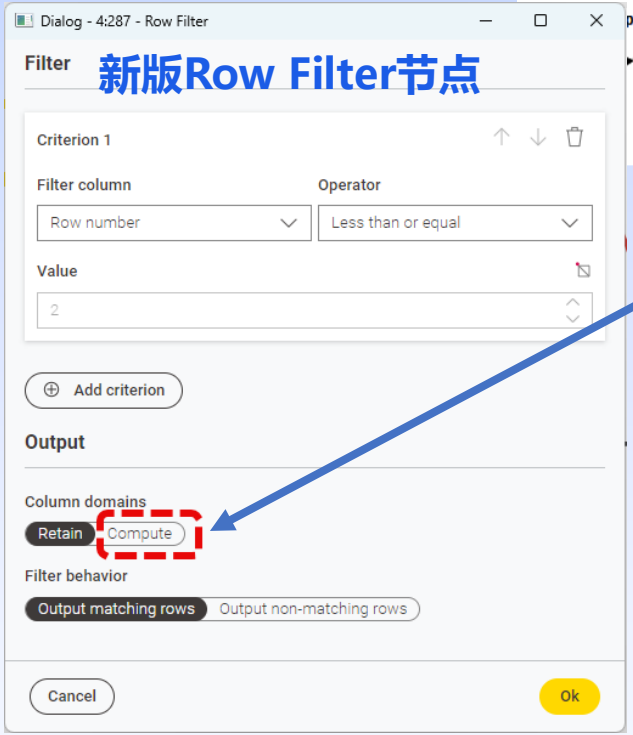
差异点	5.3及之前版本	5.4版本
AI Assistant	需要安装才能使用K-AI	默认已安装中包含 K-AI
现代 UI 和经典 UI 之间切换	无需关闭和重新打开工作流	切换时需关闭所有工作流

KNIME中各节点的界面也在不断的改进和优化，将会有更多的节点重构为Web UI界面（即本书提及的新版风格节点界面）

- 如最常用的Joiner、Row Filter、Row Splitter节点已更改为Web UI界面，其中最后两个节点修正了早先Row Filter的bug。
- 该风格界面的优点是创建和调用流变量更加方便，功能方面，总体上保持不变，因此不必纠结版本的差异。

Row Filter 亮点功能





- 5.3及之后版本Row Filter支持自动更新值域，可无需14.2.2那样再使用Domain Calculator节点



04

5.5版本日期时间类函数变化

5.5版本 日期时间类节点名称及功能变化节选

序号	所属模块	原名称	新名称	变更类型	版本	备注
37	**Time Series**	-	Date Rounder	新增	5.5	将日期圆整至月初/末，季度初/末等，方便聚合操作
38	**Time Series**	-	Time Rounder	新增	5.5	将时间圆整至最接近的分钟、小时等，方便聚合操作
39	**Time Series**	Extract Date&Time Fields	Date&Time Part Extractor	重命名	5.3	命名规范化，统一为"Extractor"
40	**Time Series**	Date&Time Shift	Date Shifter / Time Shifter	拆分、重命名	5.5	原节点被拆分为两个独立的节点，分别处理日期和时间
41	**Time Series**	Modify Date	Date Modifier	重命名	5.5	命名规范化，统一为"Modifier"
42	**Time Series**	Modify Time	Time Modifier	重命名	5.5	
43	**Time Series**	Modify Time Zone	Time Zone Modifier	重命名	5.5	
44	**Time Series**	Create Date&Time Range	Date&Time Range Creator	重命名	5.5	命名规范化，统一为"Creator"
45	**Time Series**	Duration to Number	Duration (Time-based) to Number	重命名	5.5	名称更精确
46	**Time Series**	Extract Duration Fields	Duration Part Extractor	重命名	5.5	命名规范化，统一为"Extractor"
47	**Time Series**	Date&Time Difference	Date&Time Difference	升级功能	5.5	增加选项，明确是列2减列1，或反之
48	**Time Series**	-	Date&Time Format Manager	新增	5.5	补充日期格式管理。类似String Format Manager、Number Format Manager

- 5.5版本中在Expression节点中增加了日期和时间类函数功能。



05

KNIME学习杂谈

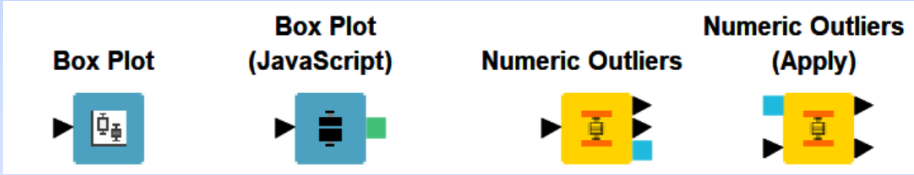
KNIME学习杂谈

新版风格节点、与经典节点界面之流变量

如Column Filter节点中的流变量名称为何那么多？但节点界面内除了选择列没有看到其他选项？
---个别节点界面内参数与流变量并不完全一一对应。

定性分析与定量分析

如Box Plot与Numeric outlier



不要一味追求工作流少用节点

逻辑清晰、简单易用就好。能少用代码尽量少用。**完成比完美更重要**，有时间再优化。

思维转变

如一个表多列之间可能存在相同的列，如何找出？假定20个列，对表转置，合并各列为一个字符串，再使用Groupby节点分组计数即可找到重复行的个数，由此得到重复列的个人

多总结、勿贪多、抓重点

注意KNIME各版本（4.3-5.8）节点名称（或功能）的变化—详见本书附录1

感谢观看

THANKS

作者：赵永江

E-Mail：yongjiang2312225@163.com

