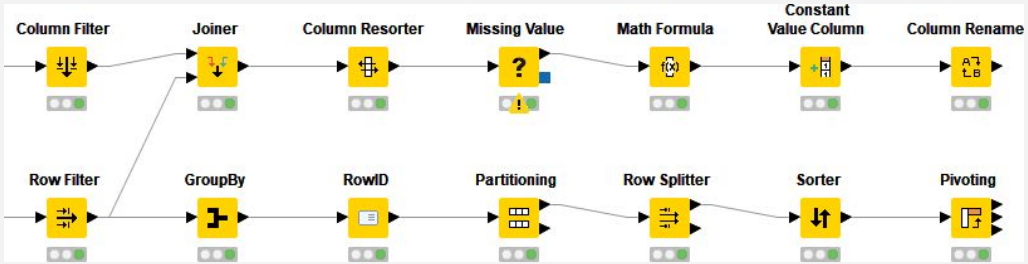
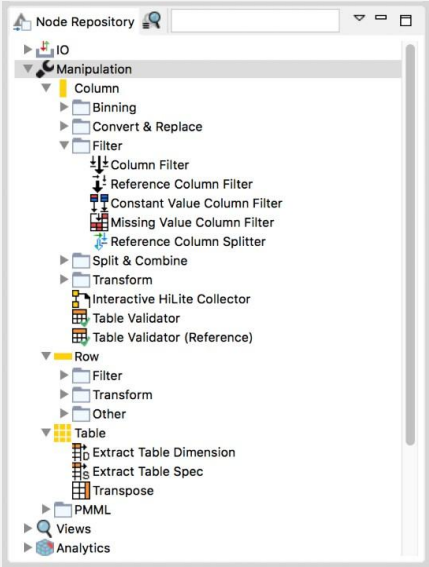
Data Manipulation

Data Manipulation Clean, Join, Aggregate

Data Manipulation Nodes

- Yellow color with a variety of input and output ports
- Apply a transformation to input data
- Many, many nodes!

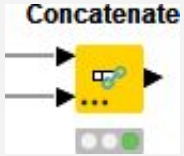


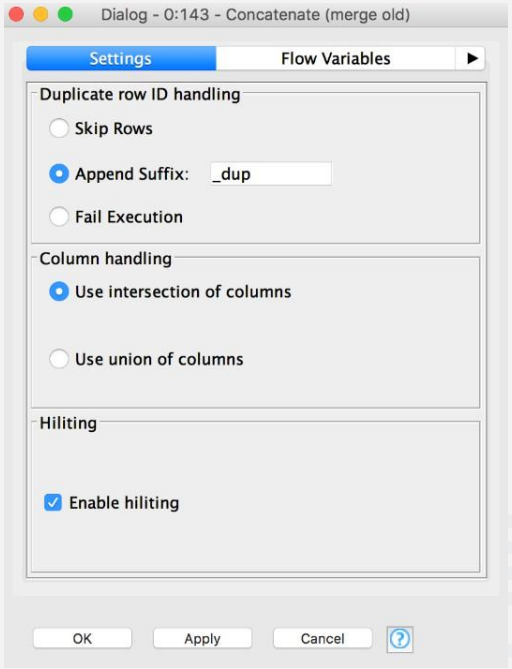


New Node: Concatenate

Combine rows from 2 or more tables with shared columns

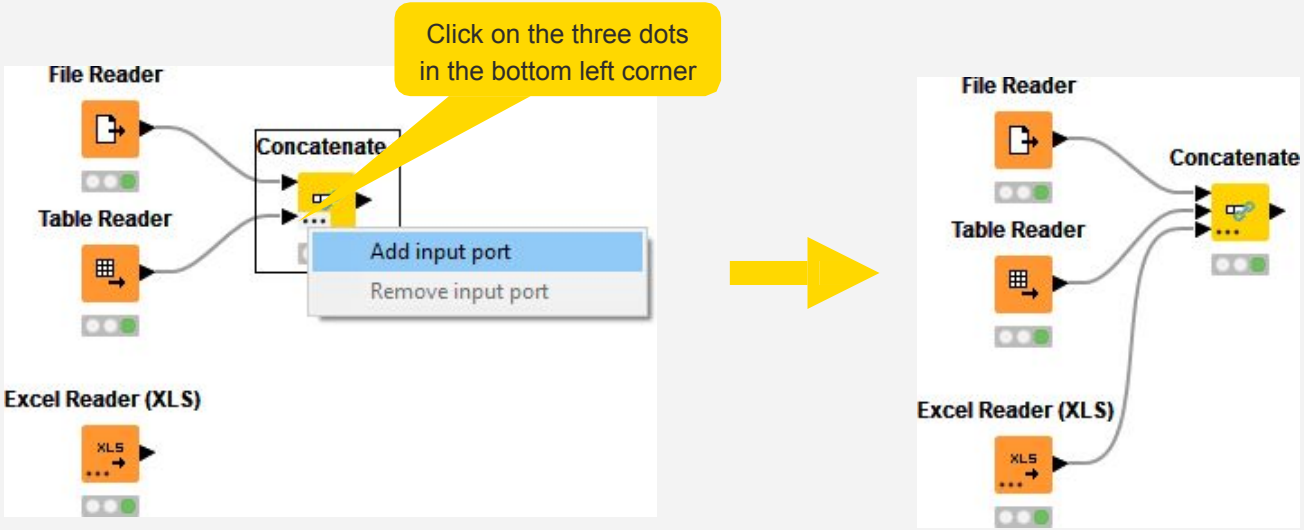
- Handles duplicate row keys gracefully
- Take the union or intersection of columns





Dynamic Ports

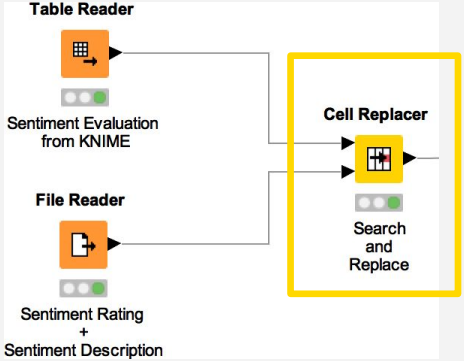
Add and remove node ports based on your needs, e.g. in order to concatenate three or more tables

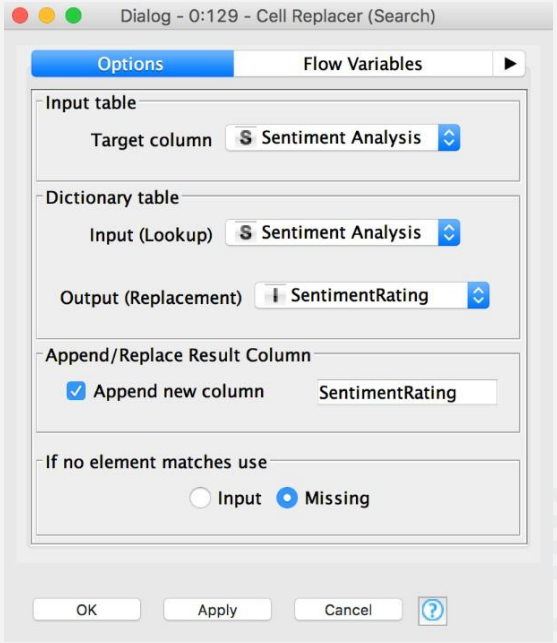


New Node: Cell Replacer

Replaces the content of a column based on a lookup

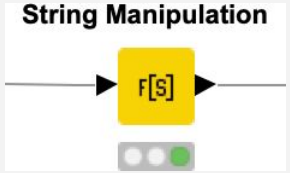
- Top port references the table to be searched
- Bottom port holds the lookup table (search keys and replacement values)

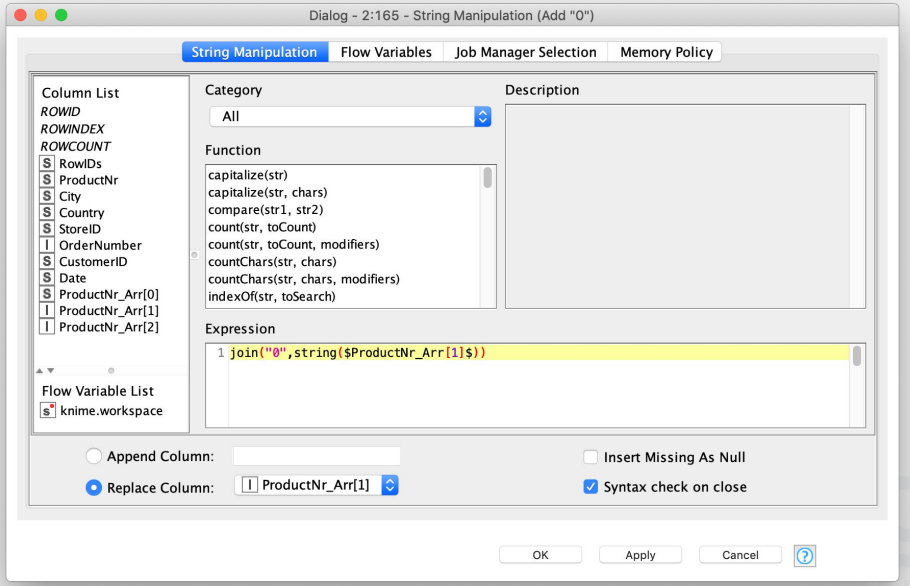




New Node: String Manipulation

- Create and edit values in a String Column
  - Cleans up capitalization
  - Joins string values
  - Pads strings, e.g. padLeft
  - Replaces string values





Data Manipulation Exercise, Activity I

Start with exercise: *Data Manipulation, Activity I*

- Concatenate web activity data from the old and new systems
- Replace the written sentiment values with the numeric sentiment scores
- Make sure that all product names in the product data spreadsheet are written in lower case letters

Joining Columns of Data

Left Table

CustomerKey	OrderDate	OrderID
22	2019-09-23	#23444
24	2019-09-30	#23457
15	2019-10-07	#28985
10	2091-10-13	#29999

Join by CustomerKey

Right Table

CustomerKey	DoB	City	Gender
17	1974-02-23	Berlin	F
65	2001-05-25	Stuttgart	F
35	1988-08-05	Cologne	M
15	1983-07-20	Hamburg	M
10	1993-01-13	Berlin	M

Inner Join

CustomerKey	OrderDate	OrderID	DoB	City	Gender
15	2019-10-07	#28985	1983-07-20	Hamburg	M
10	2091-10-13	#29999	1993-01-13	Berlin	M

Left Outer Join

CustomerKey	OrderDate	OrderID	DoB	Citv	Gender
22	2019-09-23	#23444	?	?	?
24	2019-09-30	#23457	?	?	?
15	2019-10-07	#28985	1983-07-20	Hamburg	M
10	2091-10-13	#29999	1993-01-13	Berlin	M

Right Outer Join

CustomerKey	OrderDate	OrderID	DoB	City	Gender
17	?	?	1974-02-23	Berlin	F
65	?	?	2001-05-25	Stuttgart	F
35	?	?	1988-08-05	Cologne	M
15	2019-10-07	#28985	1983-07-20	Hamburg	M
10	2091-10-13	#29999	1993-01-13	Berlin	M

Joining Columns of Data

Left Table

CustomerKey	OrderDate	OrderID
22	2019-09-23	#23444
24	2019-09-30	#23457
15	2019-10-07	#28985
10	2091-10-13	#29999

Join by CustomerKey

Full Outer Join

Right Table

CustomerKey	DoB	City	Gender
17	1974-02-23	Berlin	F
65	2001-05-25	Stuttgart	F
35	1988-08-05	Cologne	M
15	1983-07-20	Hamburg	M
10	1993-01-13	Berlin	M

Missing values in the left table

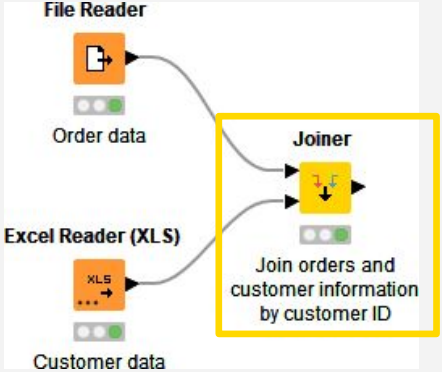
CustomerKey	OrderDate	OrderID	DoB	City	Gender
17	?	?	1974-02-23	Berlin	F
65	?	?	2001-05-25	Stuttgart	F
35	?	?	1988-08-05	Cologne	M
15	2019-10-07	#28985	1983-07-20	Hamburg	M
10	2091-10-13	#29999	1993-01-13	Berlin	M
22	2019-09-23	#23444	?	?	?
24	2019-09-30	#23457	?	?	?

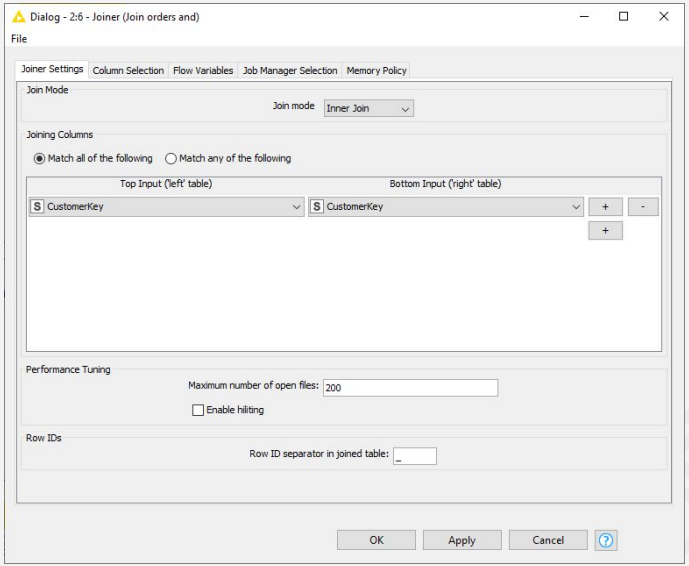
Missing values in the right table

New Node: Joiner

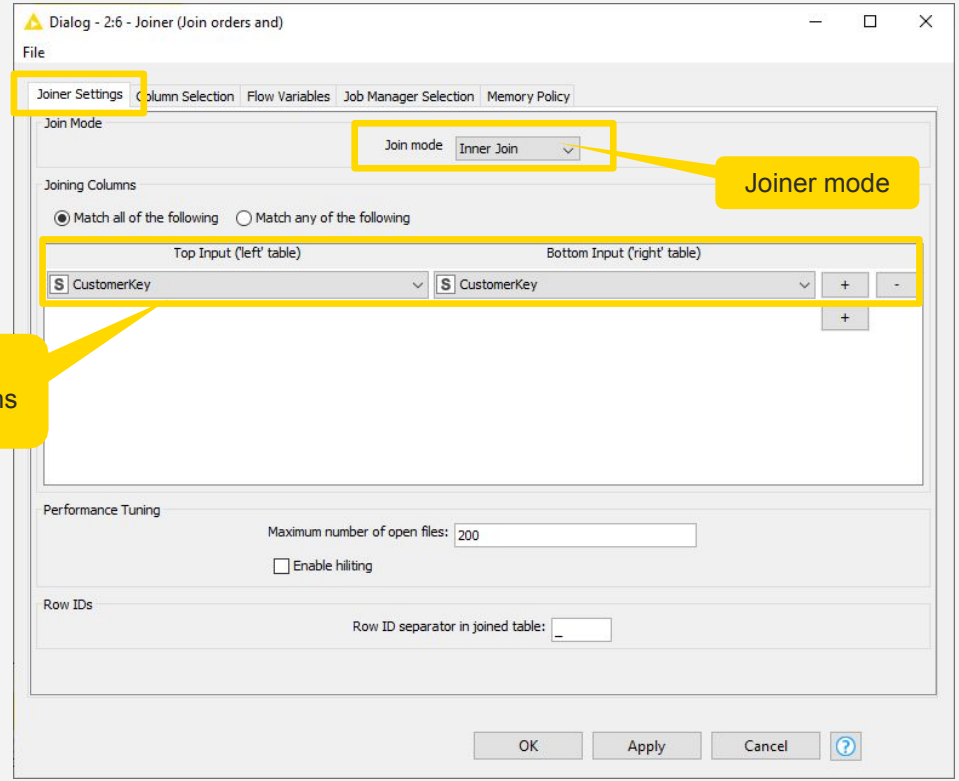
Combines columns from 2 different tables

- Top port contains “Left” data table
- Bottom port contains “Right” data table

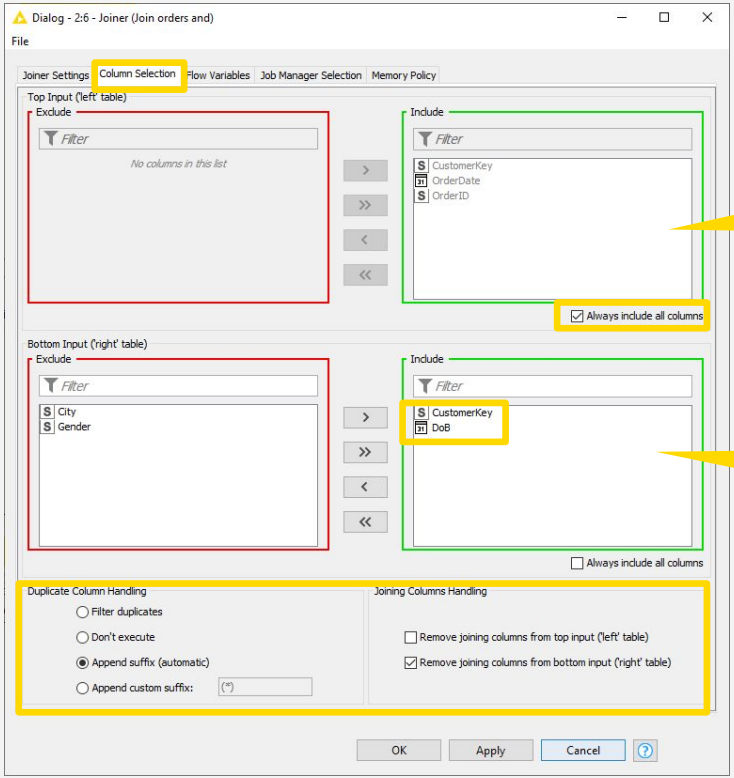




Joiner Configuration – Linking Rows



Joiner Configuration – Column Selection



Columns from left table to output table

Columns from right table to output table

Data Aggregation

Product ID	Category	# Ordered Items
P 1	Clothing	2
P 2	Home	3
P 3	Clothing	1
P 4	Clothing	5
P 5	Electronics	7
P 6	Electronics	5

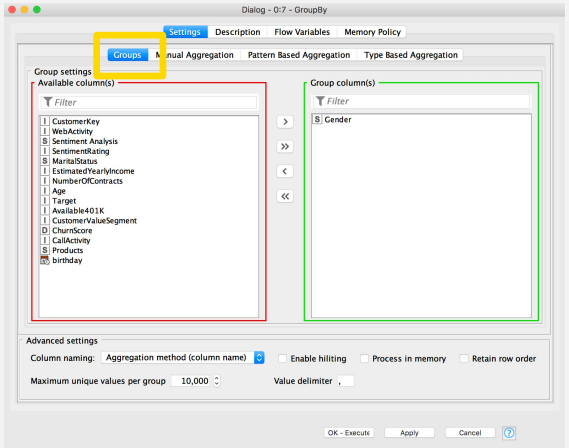
Group	Sum(# Ordered Items)
Clothing	8
Home	3
Electronics	12

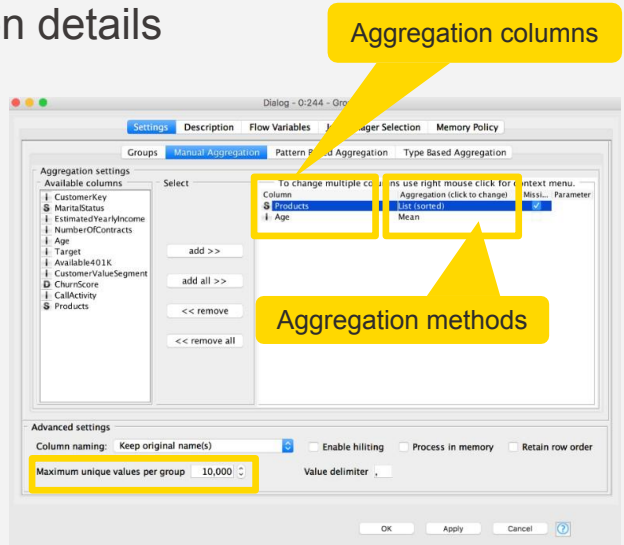
Aggregated on Category (group) by Sum (aggregation method)

New Node: GroupBy

Aggregate rows to summarize data

- First tab provides grouping options
- Second tab provides control over aggregation details



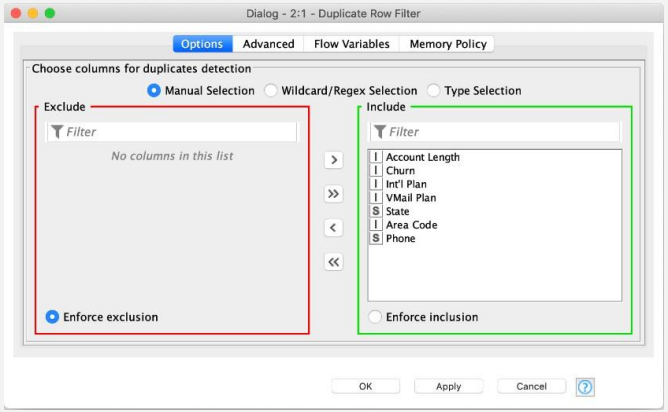


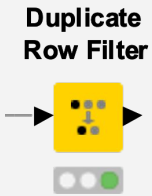
YouTube KNIME TV video:
https://youtu.be/bDwF-TOMtWw

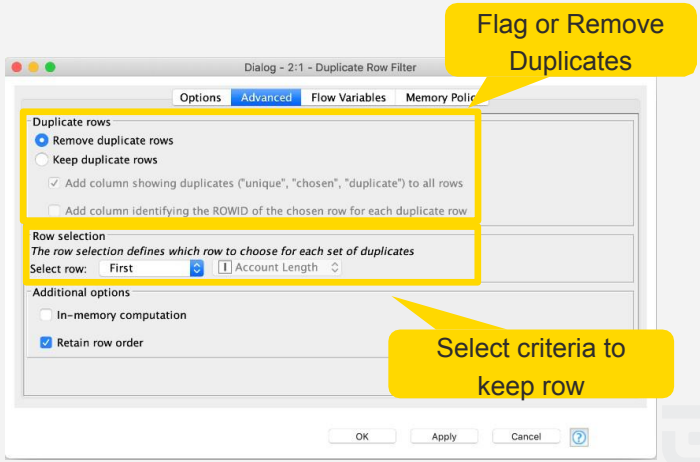
New Node: Duplicate Row Filter

Detect duplicate rows and apply a selected treatment

- First tab provides the option to select columns
- Second tab provides options for treating duplicated values

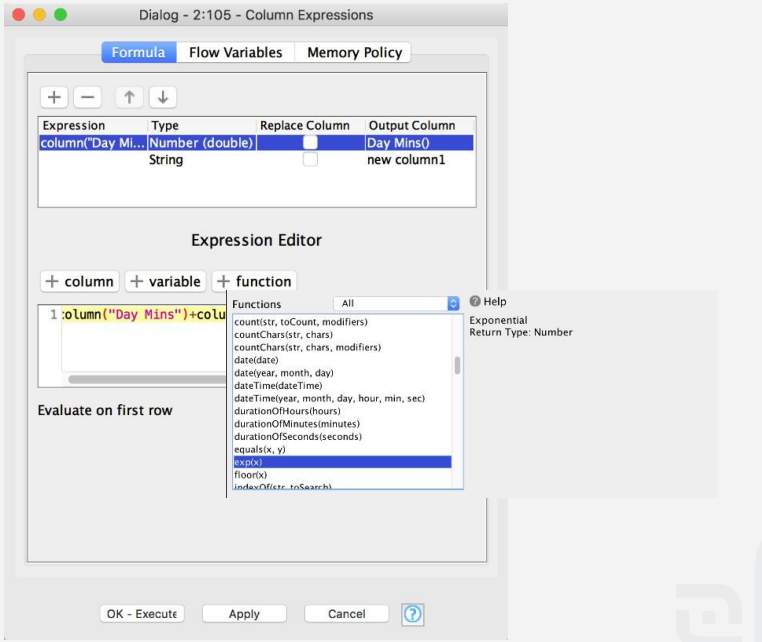






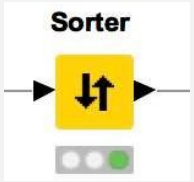
New Node: Column Expression

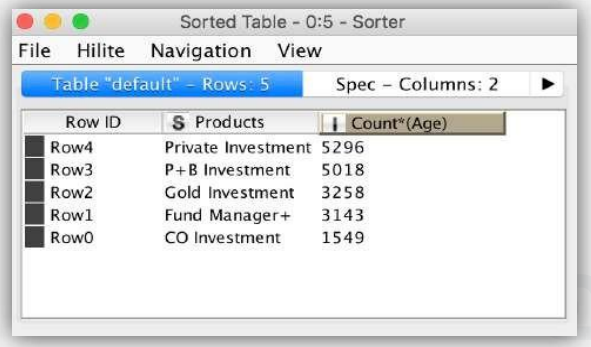
- Append or modify an arbitrary number of columns using expressions
- Many different functions are available
- No restriction on number of lines per expression allow to write complex expressions
- Part of the KNIME Labs extension



Sorter

- Sorts a table!
- Choice of ascending or descending
- Sort by multiple columns

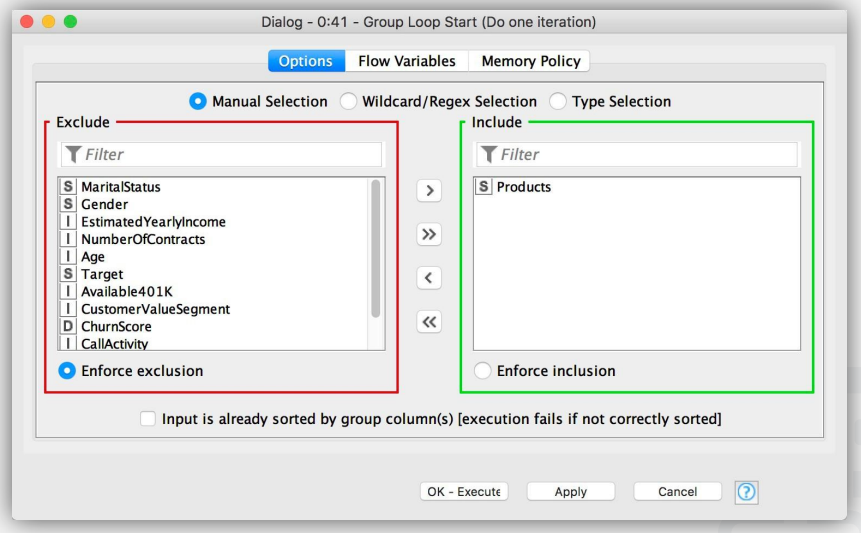


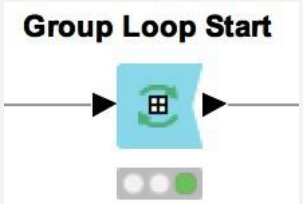


Flow Variables

Group Loop Start

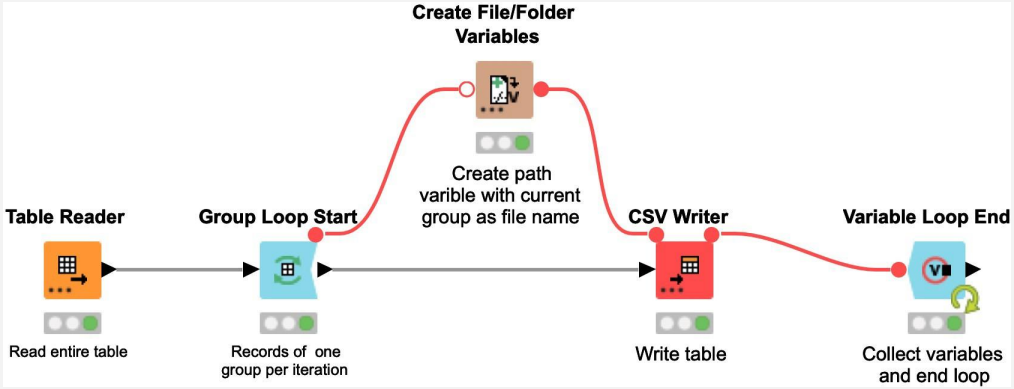
- Similar to GroupBy except without aggregation tab.
- Each iteration of the loop passes the next group of rows.
- You can implement an aggregation task. It can be anything from a complex calculation to updating a database.





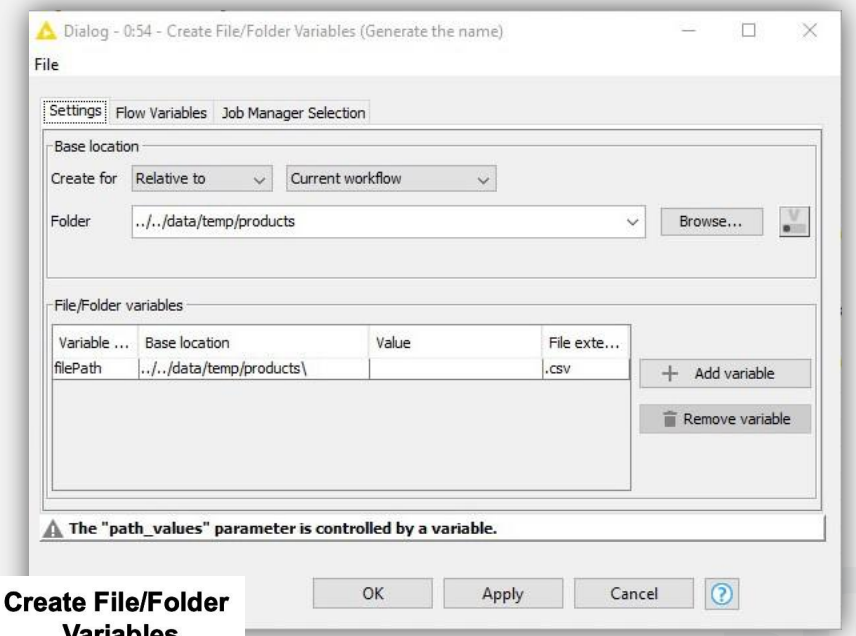
Example: Writing Aggregated Files

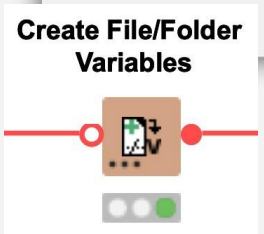
- Group Loop Start ➔ Variable Loop End
- Group data by specific column values
- Iterate over all groups of data
- Create an appropriate file name
- Write grouped data to tables with new file name



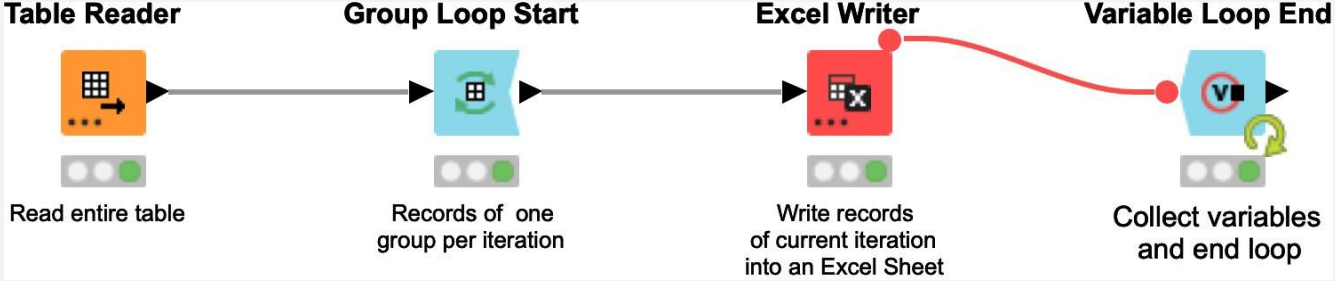
Create File/Folder Variables

- Creates one or multiple path flow variable(s) pointing to files / folders
- Inputs:
  - Base location
  - Flow variable name(s)
  - Value (file name or path relative to base location)
  - File extension (optional)
- Output variables can be used to control the output location in writer nodes.





Example: Writing Multiple Excel Sheets



Example: Reading Many Excel Sheets

- List all sheet names of an Excel file
- Convert sheet name into a flow variable (1 sheet name per iteration)
- In each iteration, read the spreadsheet with the current sheet name
- Close the loop and collect the results

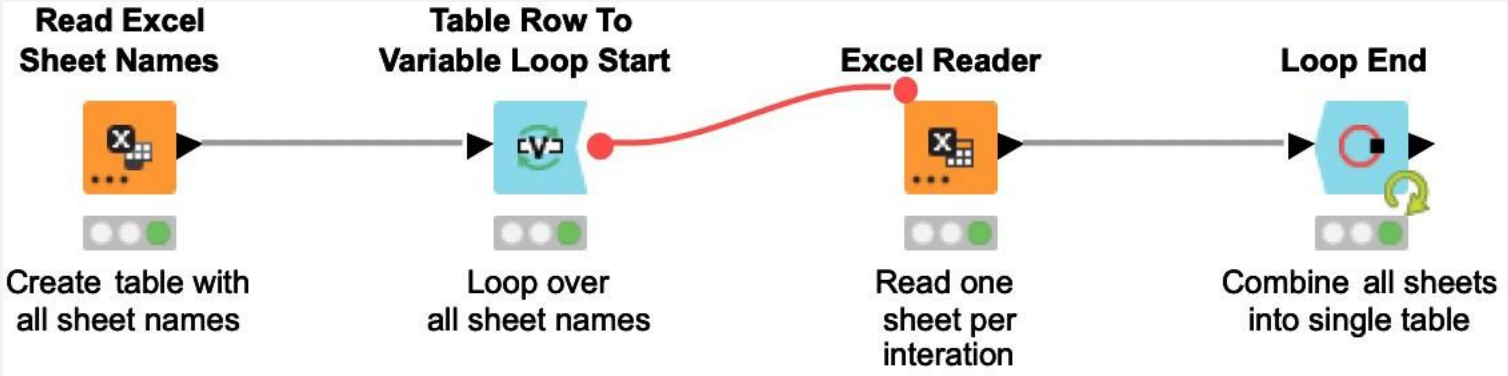
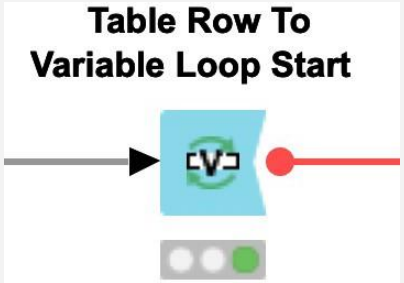
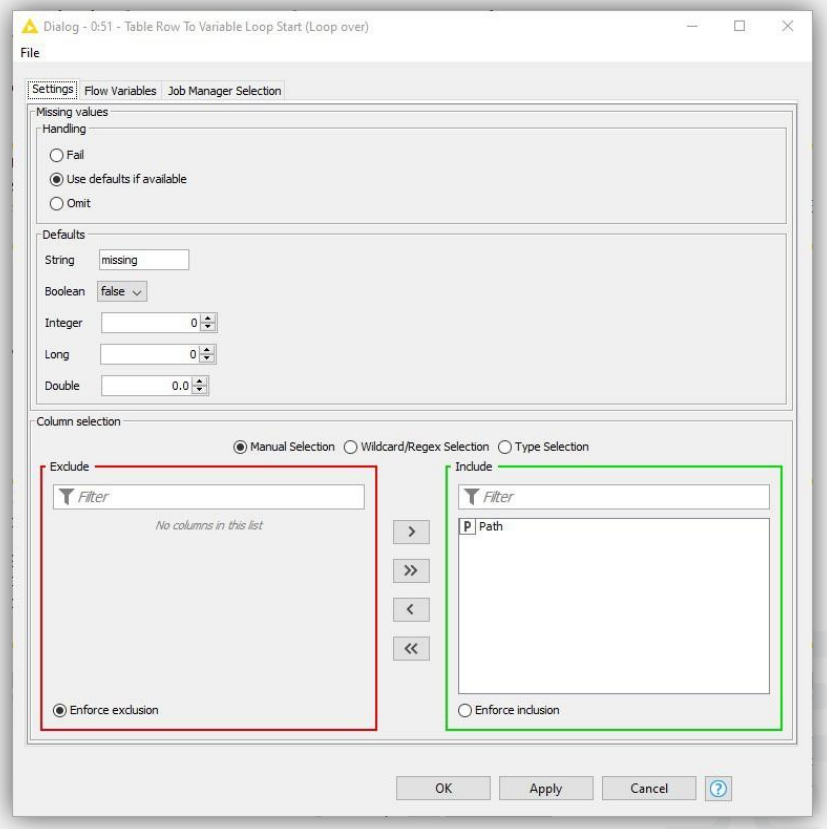


Table Row to Variable Loop Start

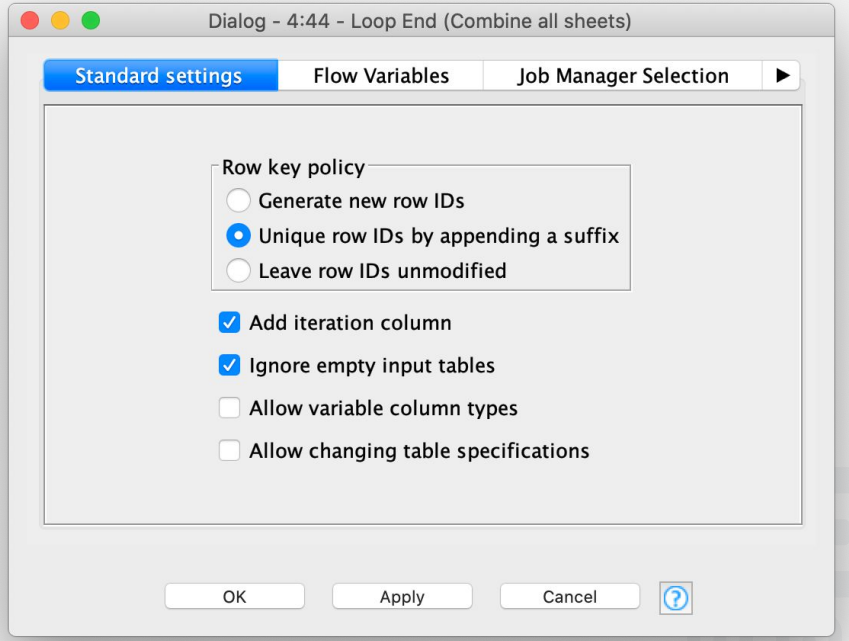
- Similar to the
  Table Row to Variable node
- Each iteration of the loop converts the next row of the input table into Flow Variables
- Injects variables into other nodes to re-execute subflows with a progression of settings

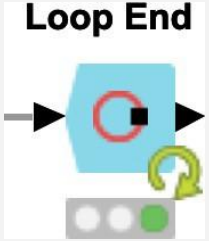




Loop End

- Can be used to end of a loop
- Collects the results of the different iterations by row-wise concatenation of the incoming tables
- Provides options to:
  - Add a column with the iteration number
  - Allow variable column types
  - Allow changing table specifications



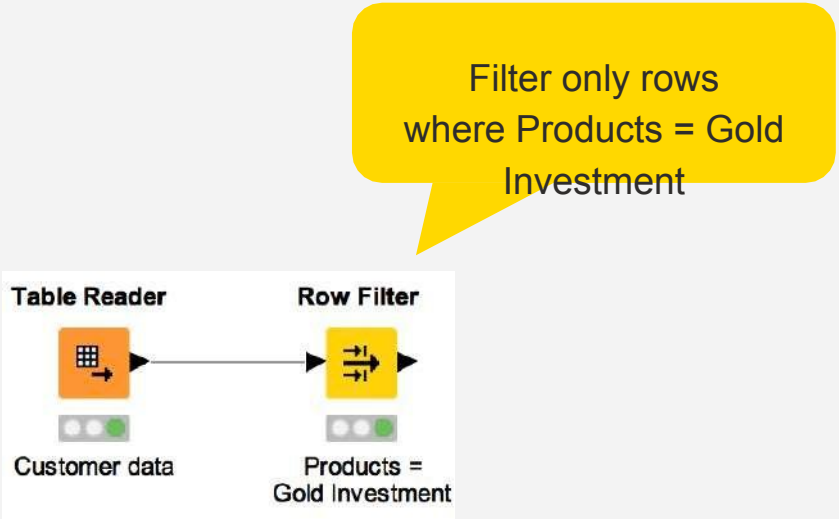


Goal of this Session

- What is a Flow Variable?
- Create a Flow Variable
- Use a Flow Variable as a parameter in the node settings
- Use a Configuration node to parameterize a Component
- Use a Widget node to enable interaction on a WebPortal page

Flow Variables: Usage Example

- Each month you need to produce a sales report for the most popular product



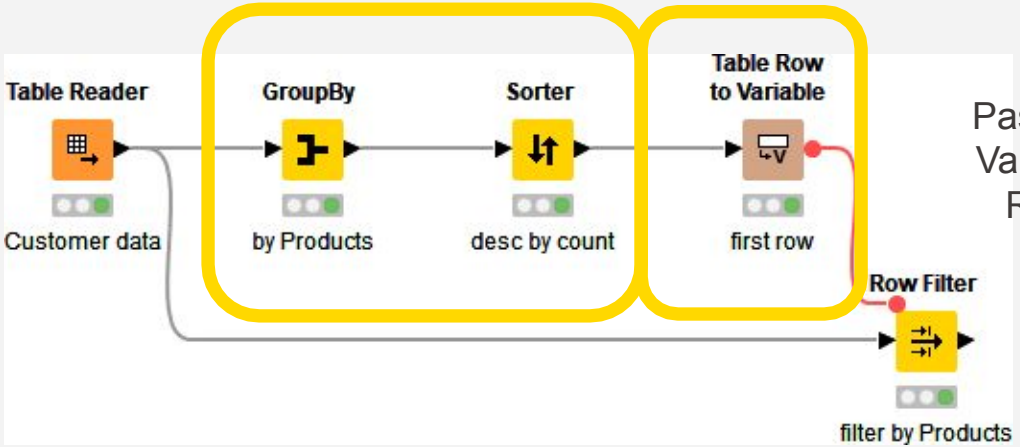
Flow Variables: Usage Example

- Each month I need to launch the Analytics Platform, aggregate the data to identify the most popular product, and update the Row Filter accordingly
- Or do I? Perhaps Flow Variables can help…

Automatically Filter by Most Popular Product

Count products, and put most popular at the top of the list

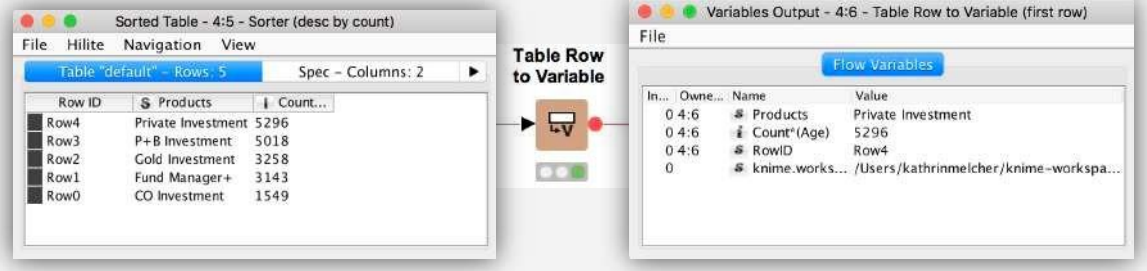
Create a Flow Variable containing the name and count of the most popular product

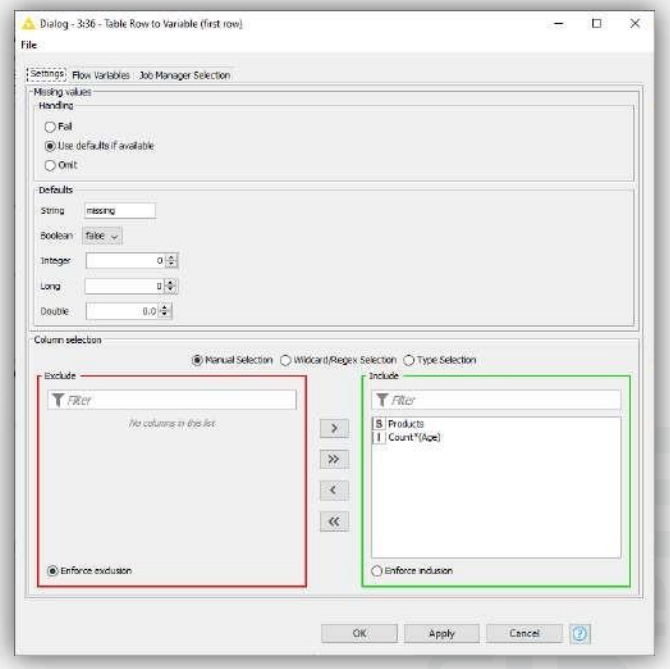


Pass the Flow Variable to the Row Filter

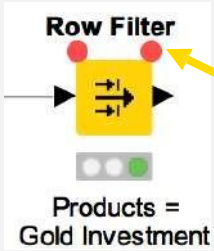
Table Row to Variable

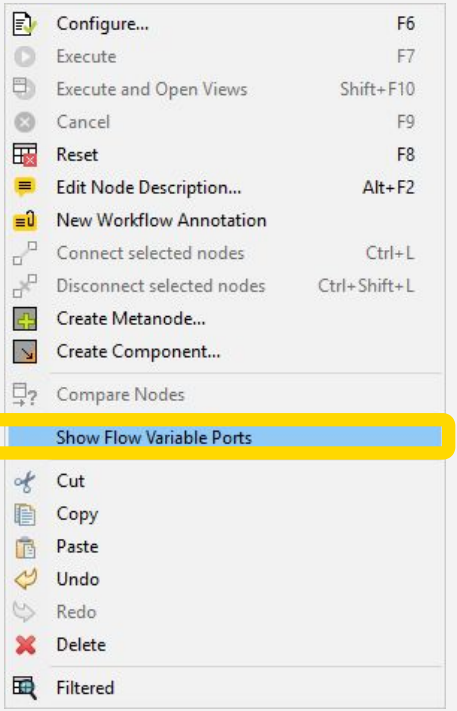
- Takes a table as input and converts the first row to Flow Variables
  - Column names -> Flow Variable names
  - Column values -> Flow Variable values
- Only the first row is transformed, additional rows are discarded



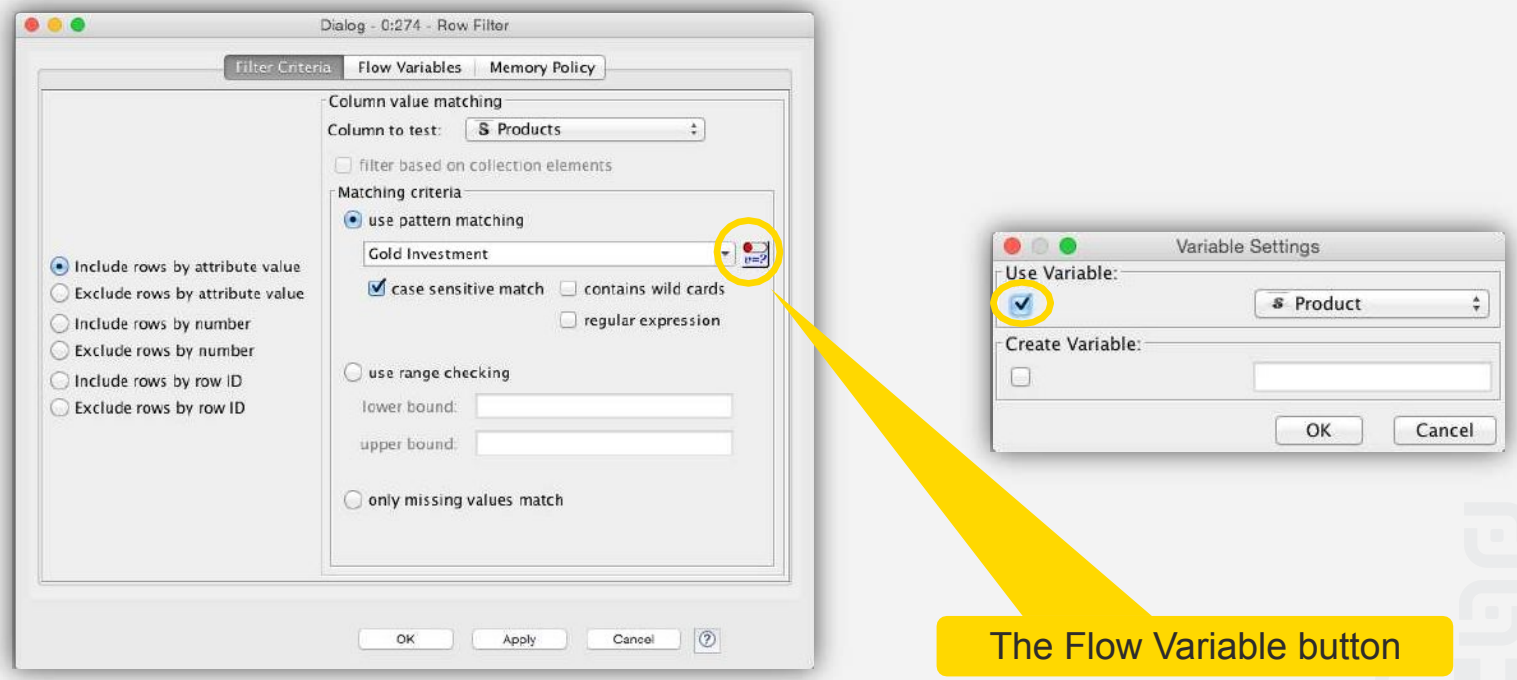


Flow Variable Ports

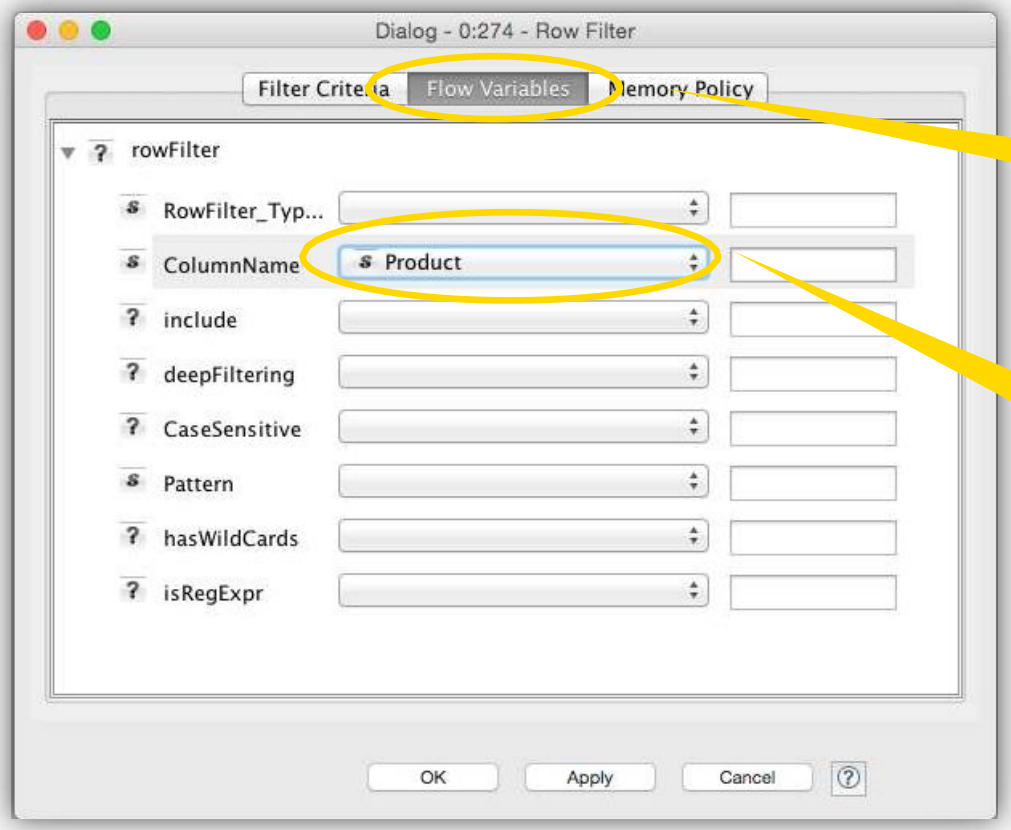




Apply a Flow Variable (Button)



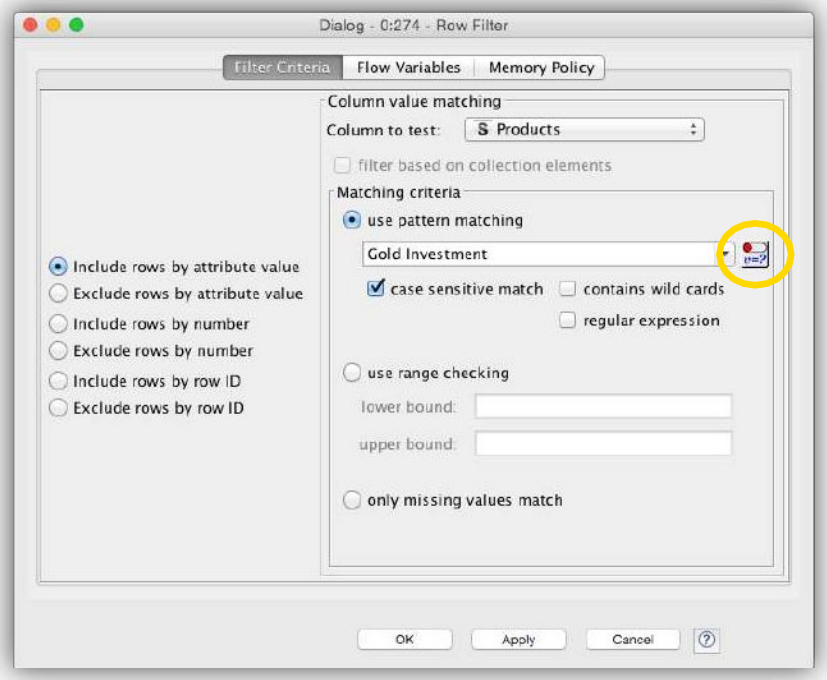
Apply a Flow Variable (Advanced)

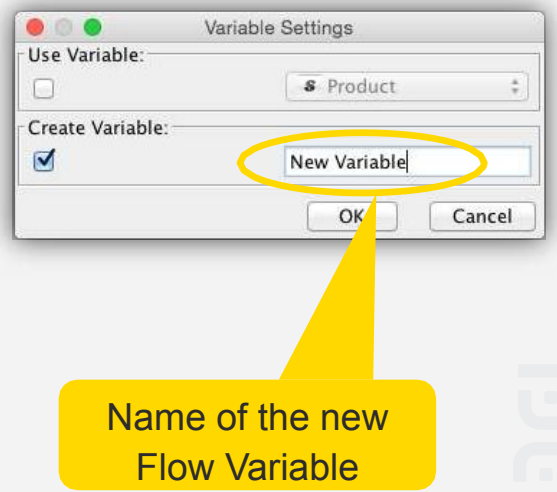


The Flow Variables tab

List of available Flow Variables

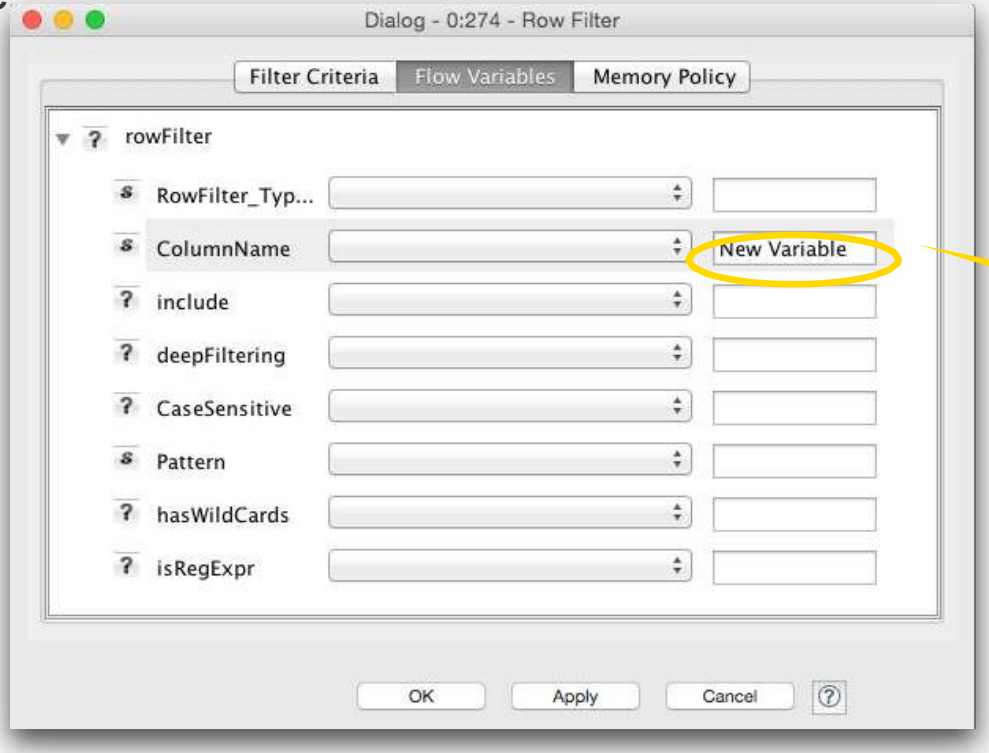
Create a Flow Variable (Button)





Create a Flow Variable (Advanced)

Converting a setting value into a Flow Variable



Name of the new Flow Variable

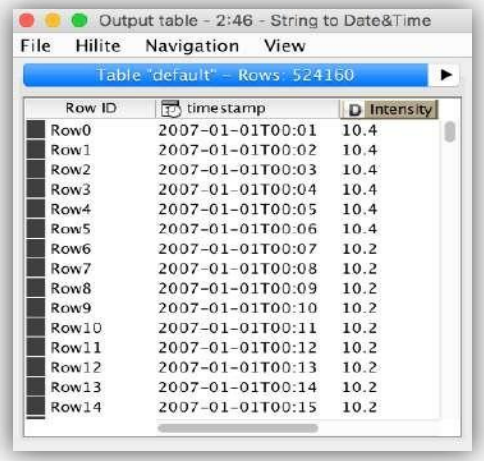
Key Features: Flow Variables

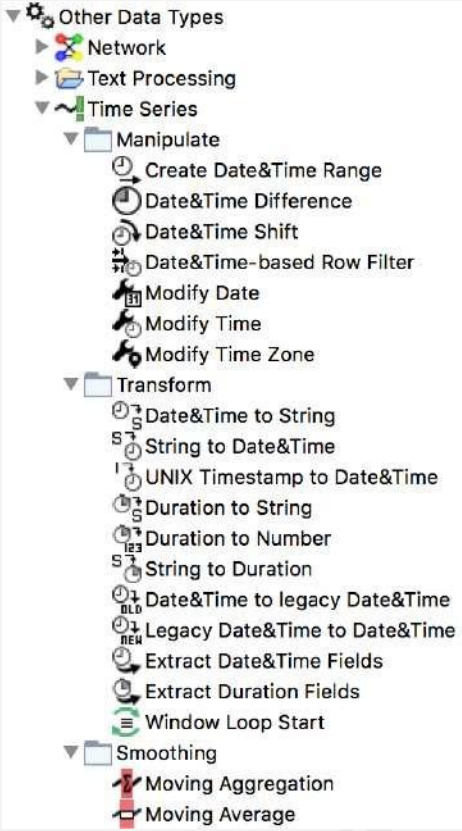
- Flow Variables are workflow parameters used to overwrite existing node settings
- A Flow Variable is carried along workflow branches (parallel branches don’t share local Flow Variables)
- Flow Variables can be of type String, Integer, Double, Boolean, Long and Array, Path
- Flow Variables can be created
    1. in the “Flow Variables” tab of any node
    2. using the “Table Row to Variable” node
    3. using Configuration and Widget nodes

Date/Time Data

Date & Time Overview

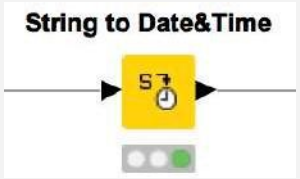
- Dedicated data type for date and time data
- Supported in Date&Time nodes
  - (and others: GroupBy, Pivot, Line Plot)
- Complete re-write in KNIME 3.4

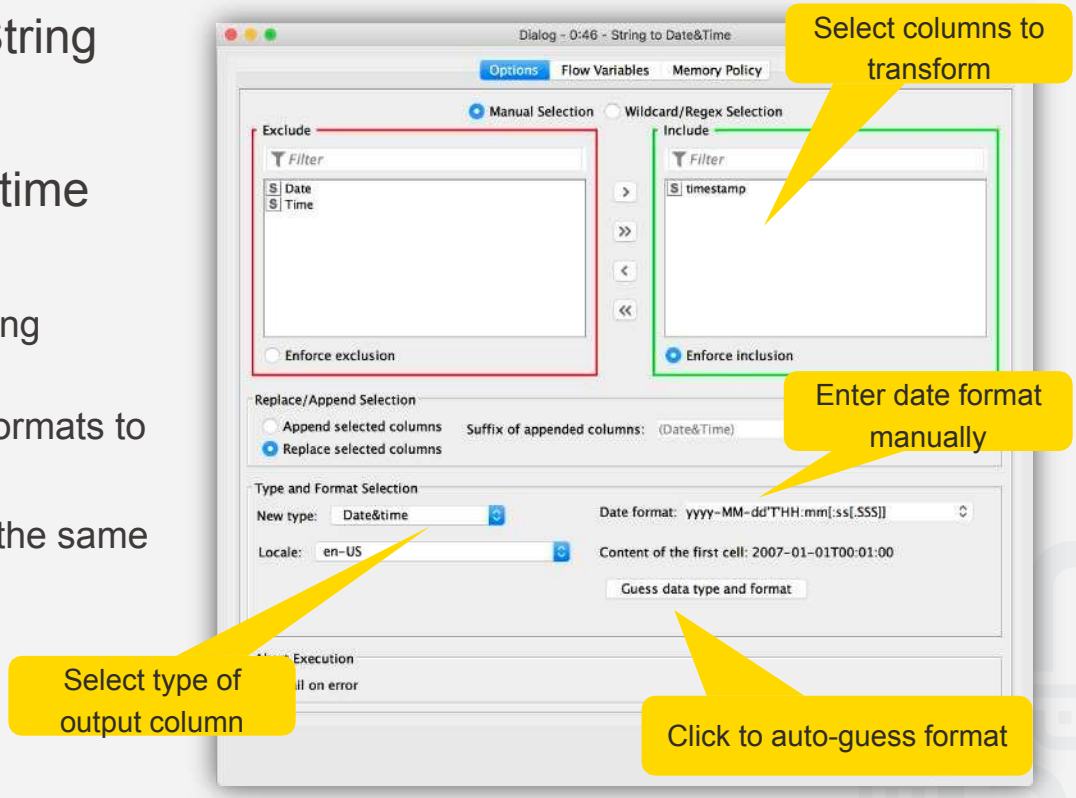




String to Date&Time

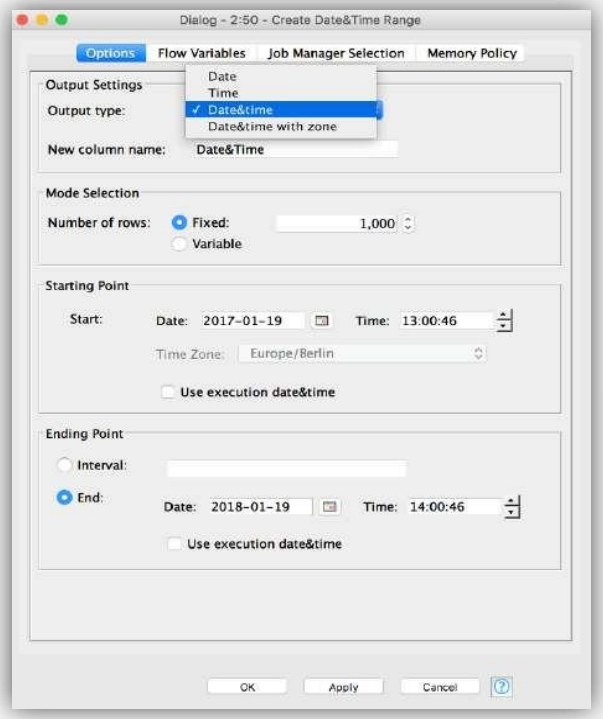
- Convert date/time data from String into a native Date&time cell
- Guesses correctly many date/time formats in String columns
  - Enter format manually if auto-guessing doesn‘t work
  - KNIME automatically adds custom formats to auto-guess list
  - Convert multiple columns that have the same date/time format by one node

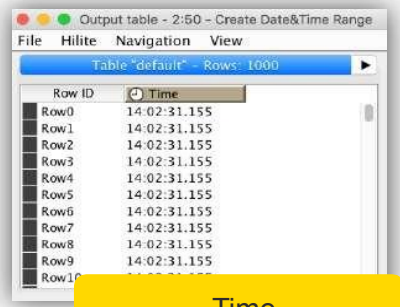




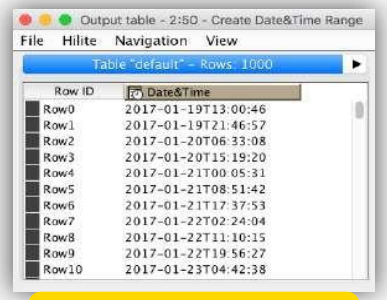
Date&Time – Data Types

Date

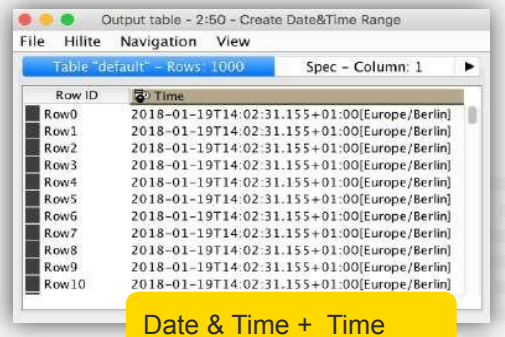




Time



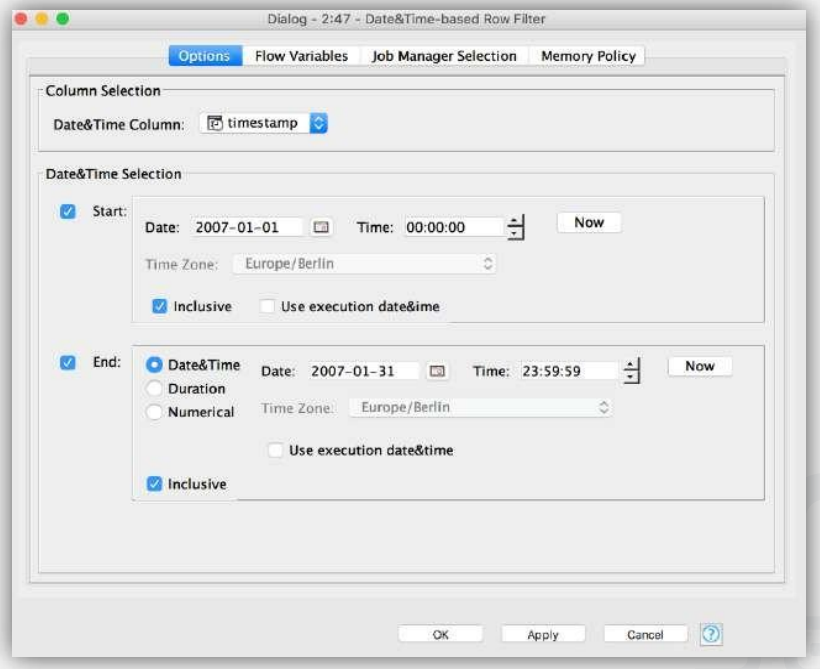
Date & Time

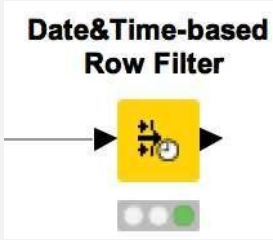


Date & Time + Time zone

Date&Time-based Row Filter

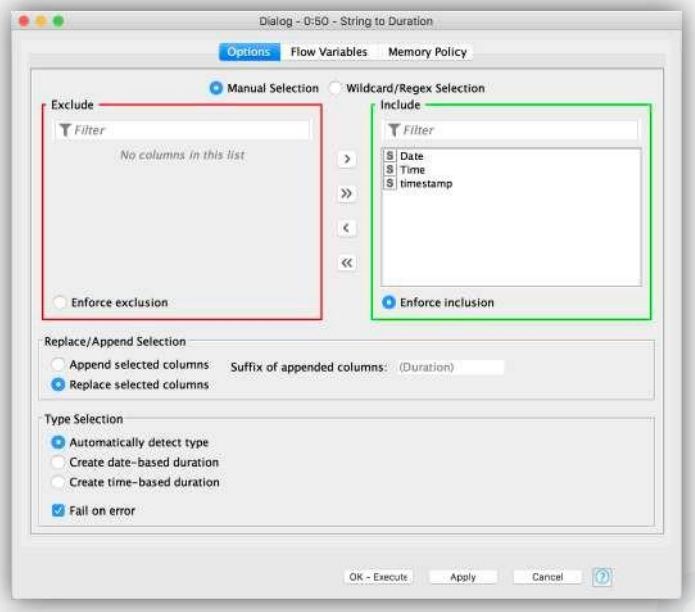
- Filter rows from a specified time period
- Range can be limited on upper bound, lower bound, or both
- Options for end point:
  - Date&Time: Fixed date and time
  - Duration: Duration string (e.g. 2y 3M)
  - Numerical: Select granularity from a dropdown menu and enter a number

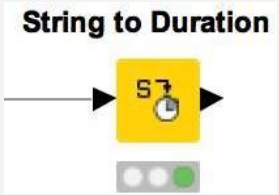


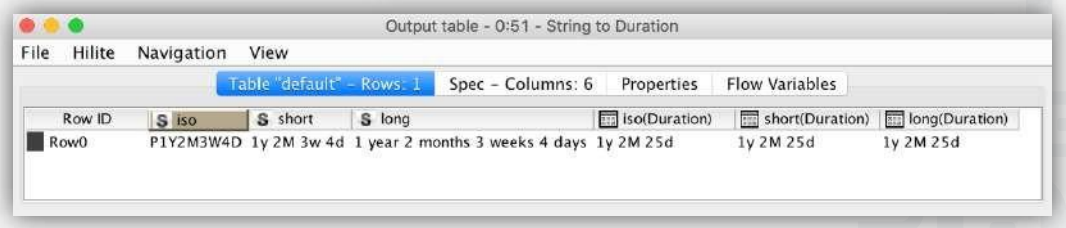


String to Duration

- Takes a String and converts it to a duration cell
- Three different options to format input Strings
- Example: Convert 1 year, 2 months, 3 weeks, and 4 days to duration cell
  - ISO-8601: “P1Y2M3W4D“
  - Short letter: “1y 2M 3w 4d”
  - Long word: “1 year 2 months 3 weeks 4 days”

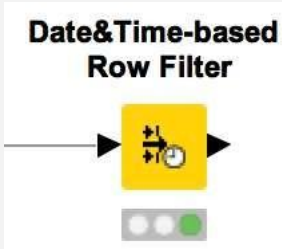


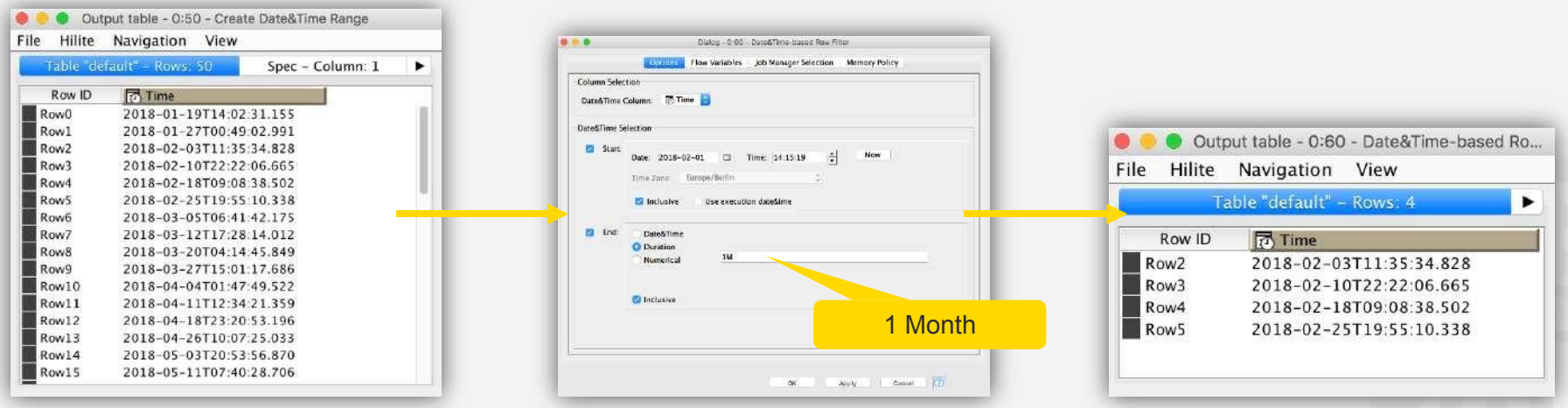




Duration-based Filtering

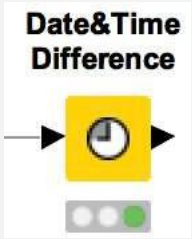
- Date&Time-based Row Filter allows to extract time periods
- From the start date, select all rows within the defined period
- Use one of the three options to define the duration, e.g.
  - ISO-8601: “P1Y2M3W4D“
  - Short letter: “1y 2M 3w 4d”
  - Long word: “1 year 2 months 3 weeks 4 days”

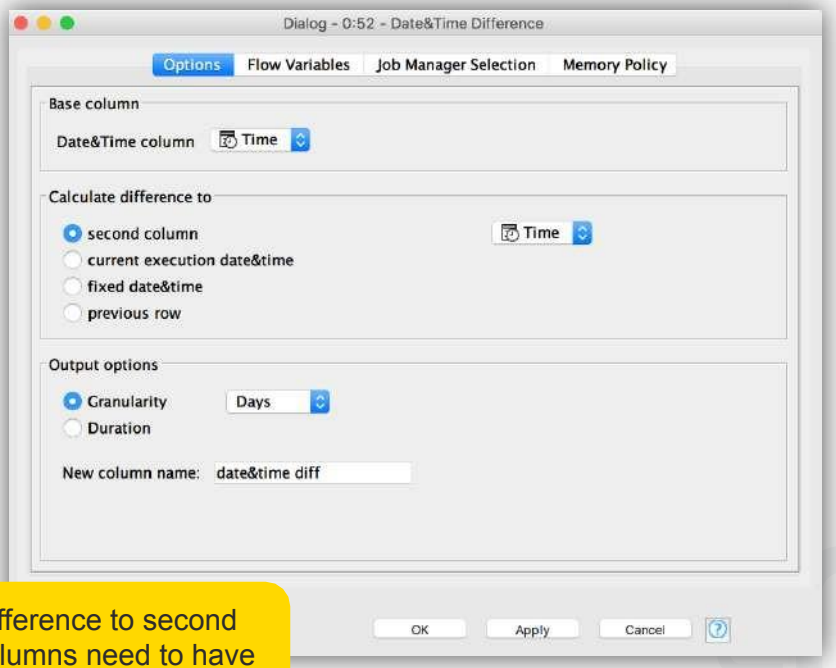




Date&Time Difference

- Choose desired granularity (days, hours, minutes, etc.)
- Check the difference between a time column and…
  - Another time column
  - Execution time
  - User-defined time
  - Previous row

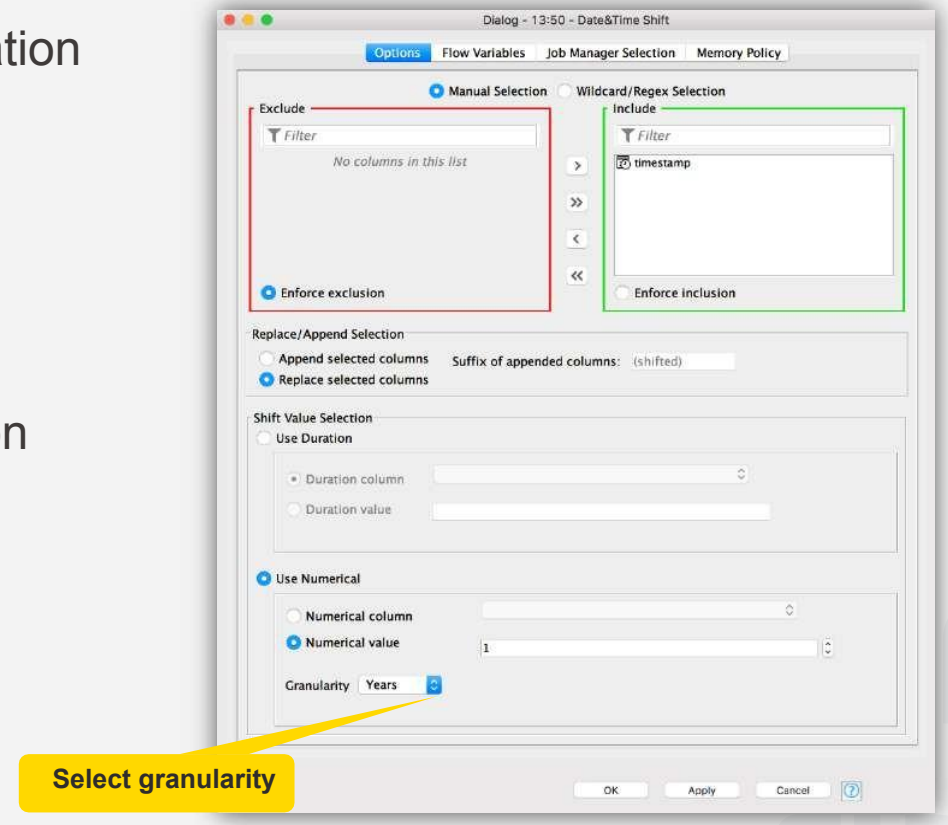




To calculate difference to second column, both columns need to have the same type!

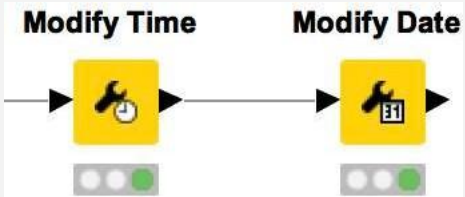
Date&Time Shift

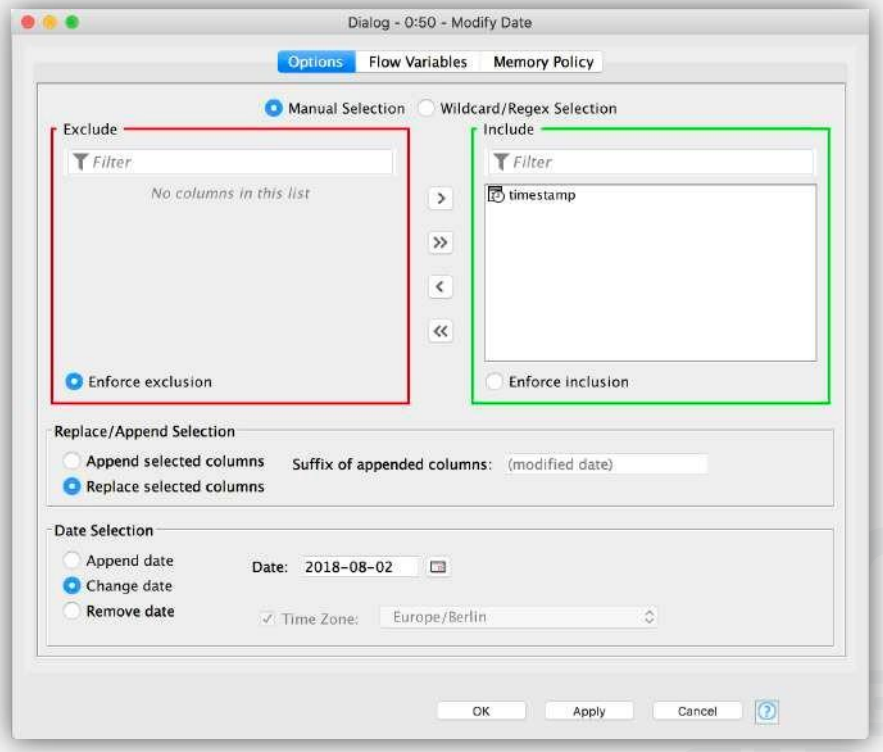
- Shifts date or time by either a duration or a numerical value
- Use duration:
  - Use duration column
  - Or shift by user defined value
    - E.g. 1y, 2M, 5h, etc.
- Use numerical value in combination with selected granularity
  - Use numerical column
  - Or shift by user defined value



Modify Time / Modify Date

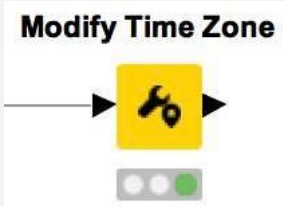
- Modify Date&Time columns
- Three options:
  - Append time (date) to a date (time) column
  - Change time (date) to a fixed value
  - Remove time (date) from a Date&Time column
- Column selection shows only columns that can be modified by the current configuration

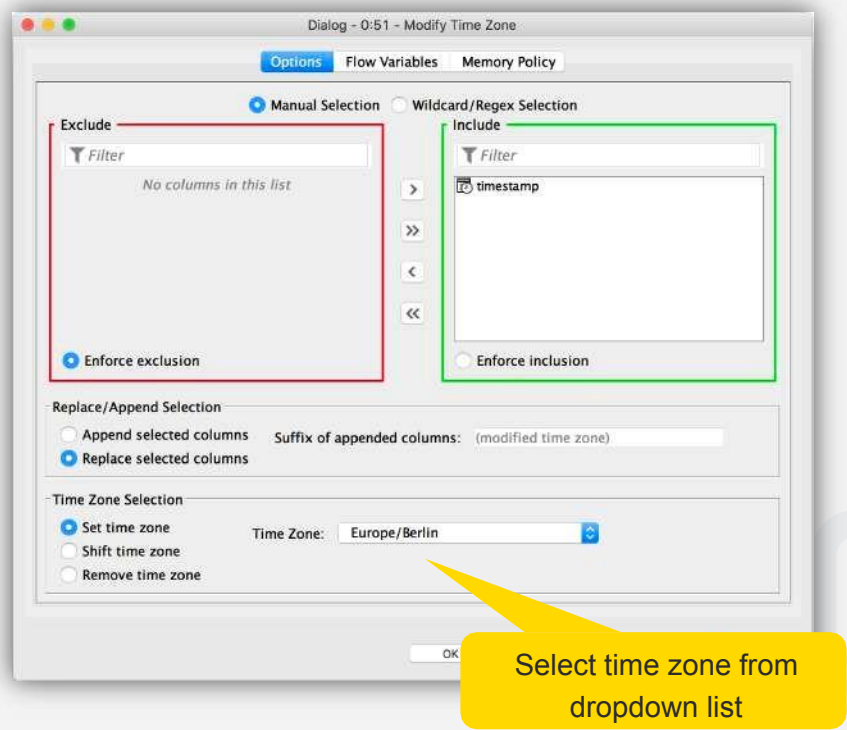




Modify Time Zone

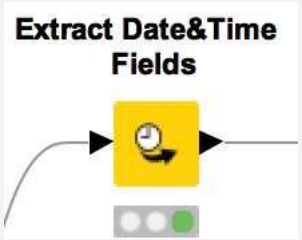
- Similar to Modify Time/Modify Date
- Input: Date&Time
  - Set time zone
- Input: Date&Time (Time zone)
  - Set time zone
  - Shift time zone
  - Remove time zone

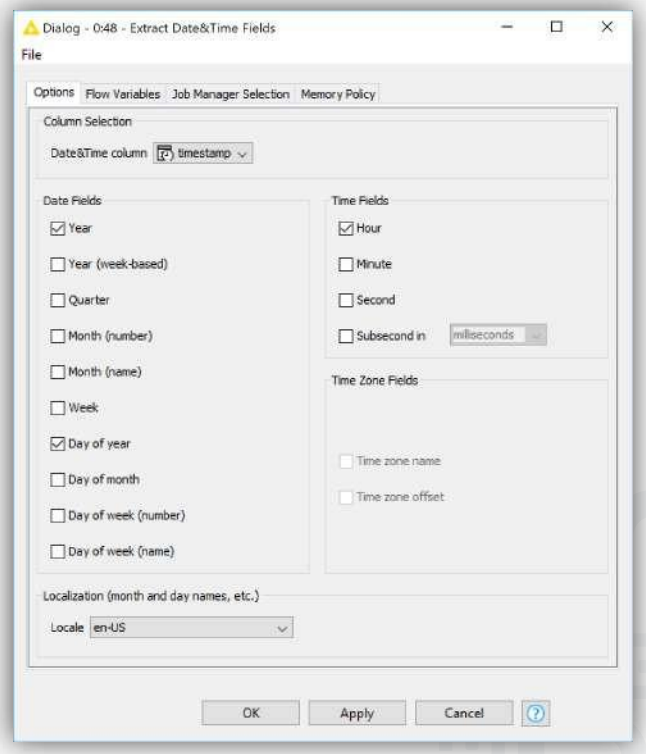




Extract Date&Time Fields

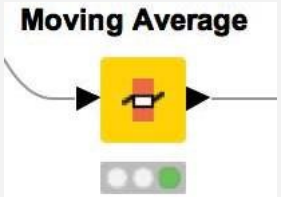
- Extract date fields (year, day, month,...) or time fields (hour, minute, second,...) from a Date&Time cell
- Pick and choose which fields to include
- Useful when used in combination with data aggregation nodes (GroupBy, Pivoting, etc.)

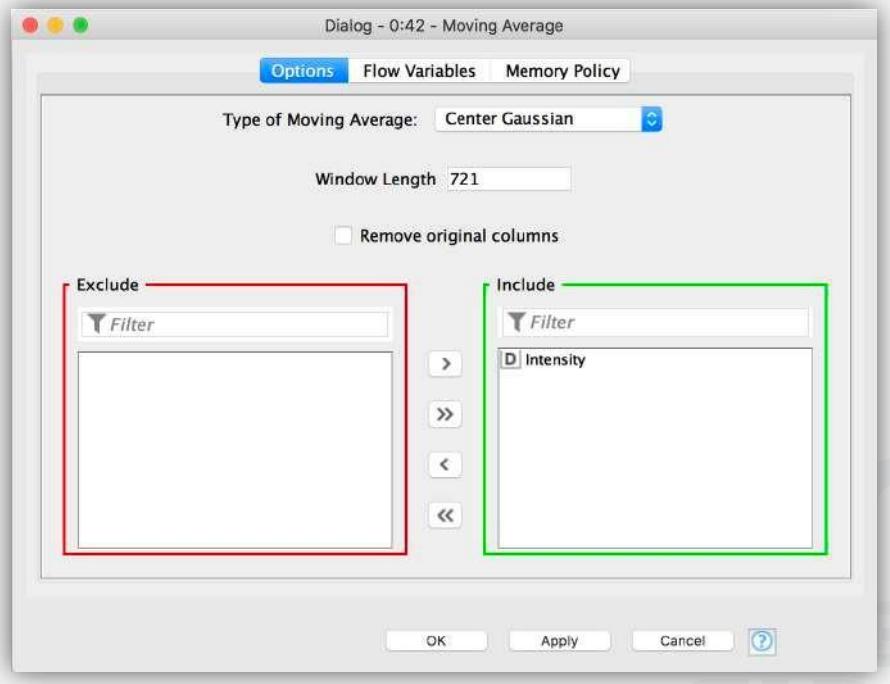




Moving Average

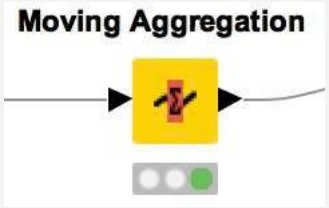
- Effective “smoothing” node
- Smoothing defined by
  - window type (centered, forward or backward)
  - window length
  - weighted or not
- Useful when plotting aggregated time series data to more easily see trends

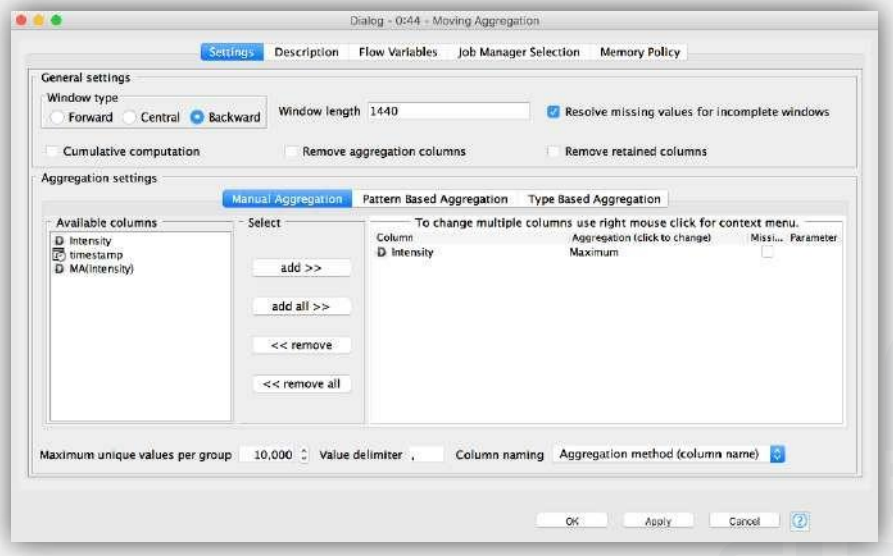




Moving Aggregation

- Blend of GroupBy + Moving Average Functionality
- Group by moving window
- Aggregate using standard KNIME methods

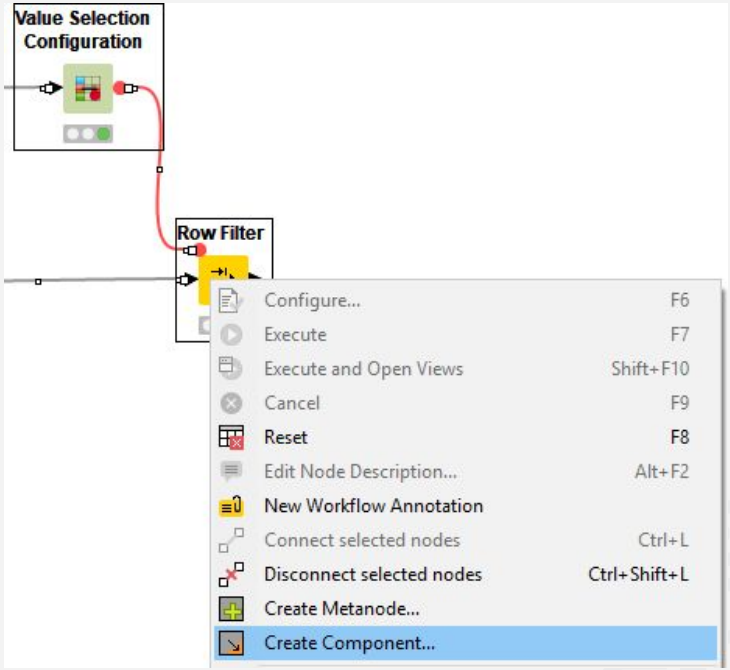




Workflow Organization and Documentation

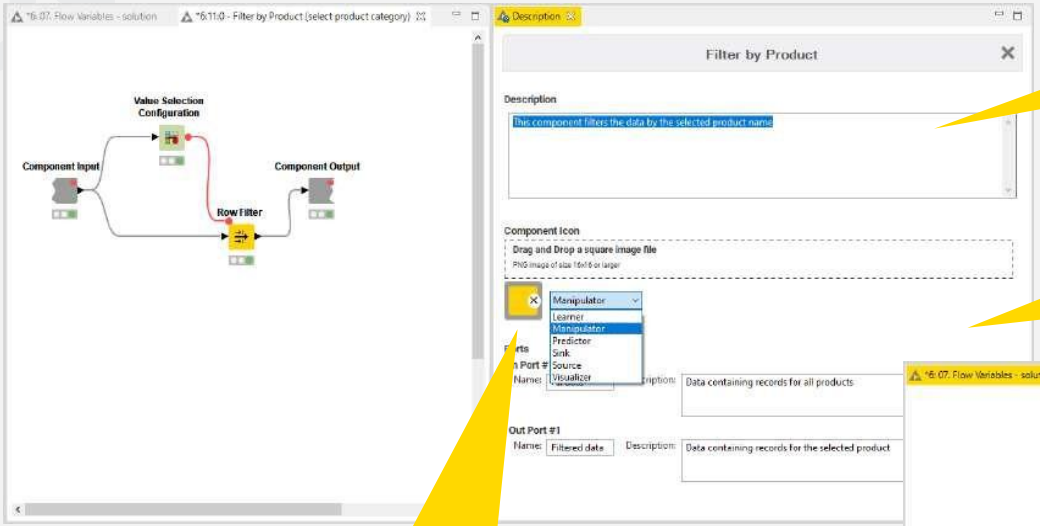
Create a Component

- Select nodes to encapsulate into a Component
- Right click a node
- Select “Create Component…”



Component Description

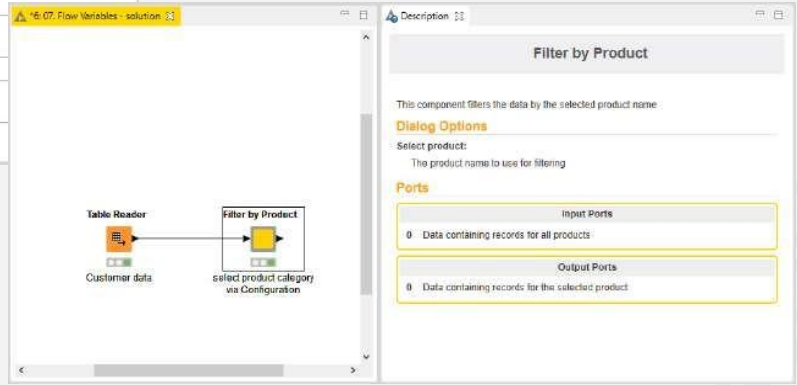
Make your component look like a KNIME node



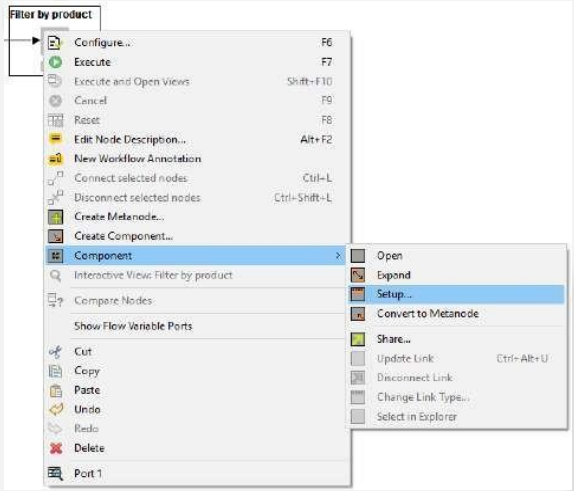
Add description of the component

Add description of the input and output ports

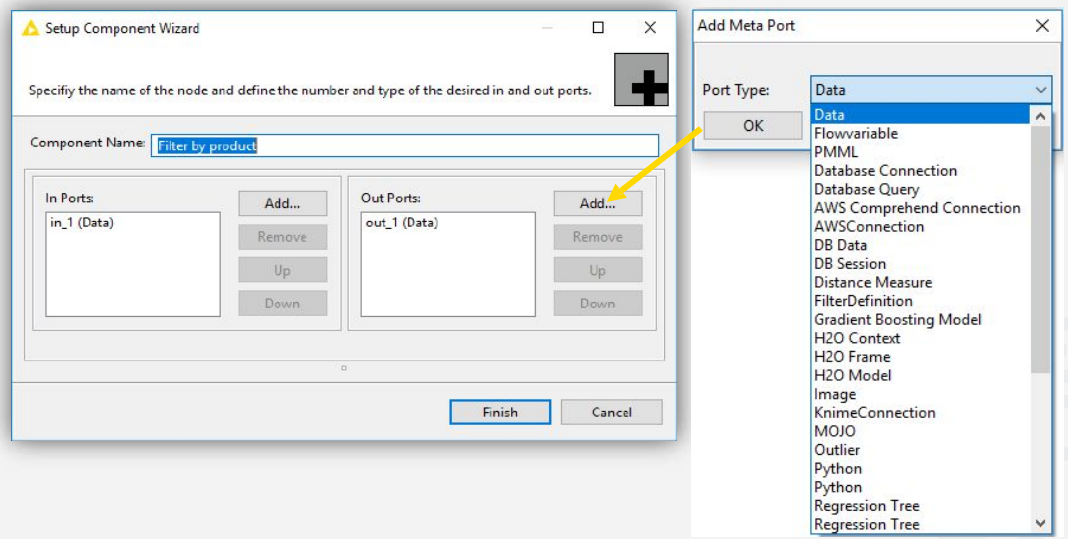
Add background color or icon



Configure Component Ports

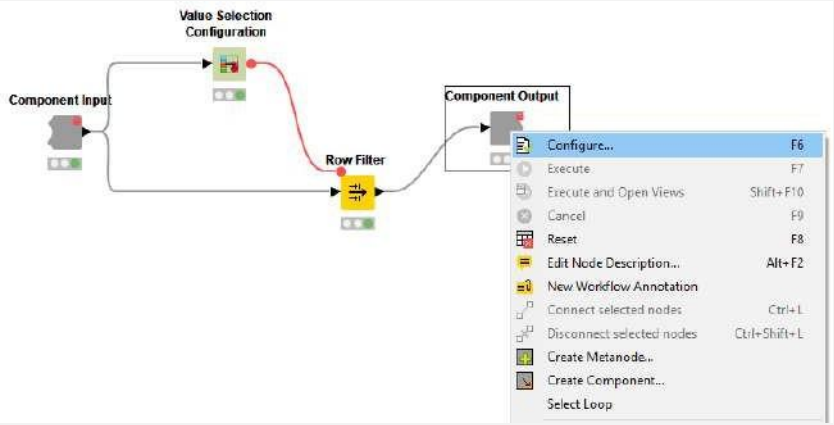


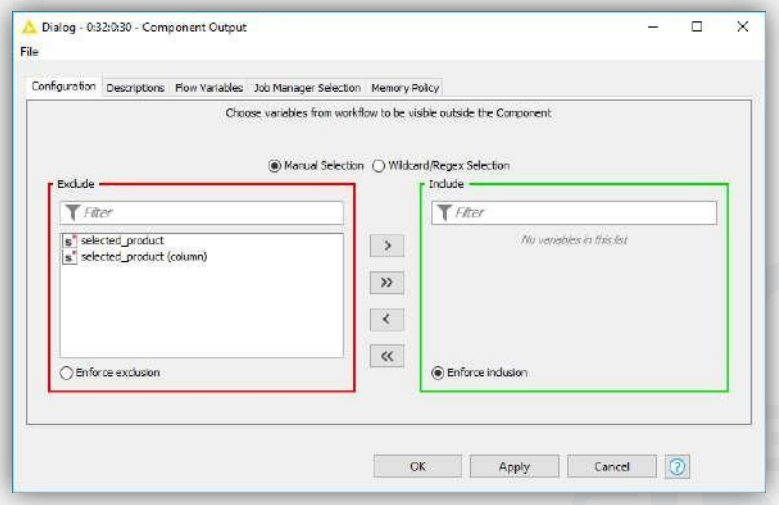
- Add input and output ports to Metanodes/Components
- Remove ports to adapt to changes after creation of Metanode/Component



Passing Variables from Components

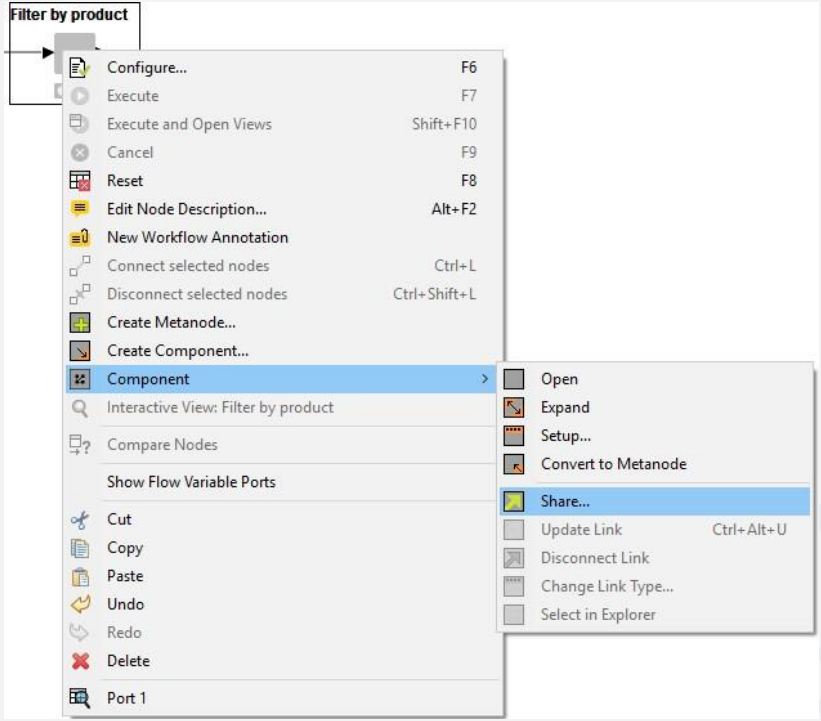
- Flow Variables by default only available locally inside Component
- Configure Component Input/Output to pass Flow Variables from/to outside Component





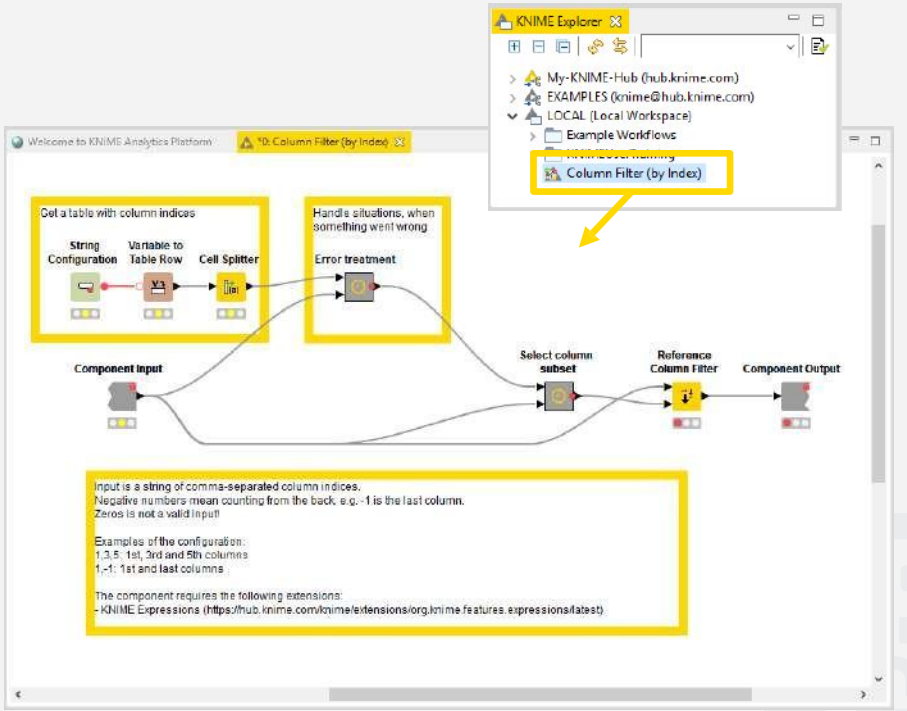
What is a Shared Component?

- Components can be saved in your KNIME workspace for later reuse
- To do this, simply right-click any Component and select “Share…”
- Shared Components are read-only instances of a Component
- Public Shared Components are available on EXAMPLES Server and on KNIME Hub



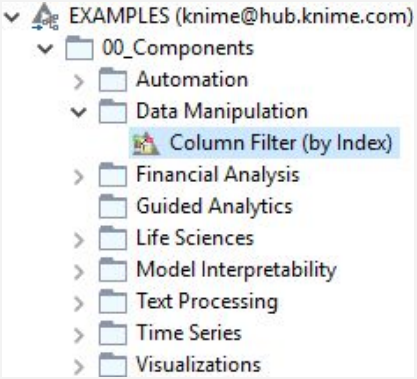
How can you Edit a Shared Component?

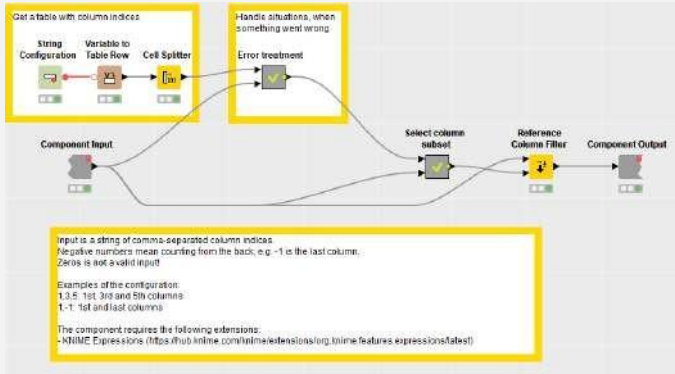
- Components can be edited using the Component Editor, similar to workflows
- To edit a Component using the Component Editor, double-click the Component in its location in the KNIME Explorer
- To ensure components are executable when opened in the Component Editor, chose the option to “Include input data with component” when sharing it



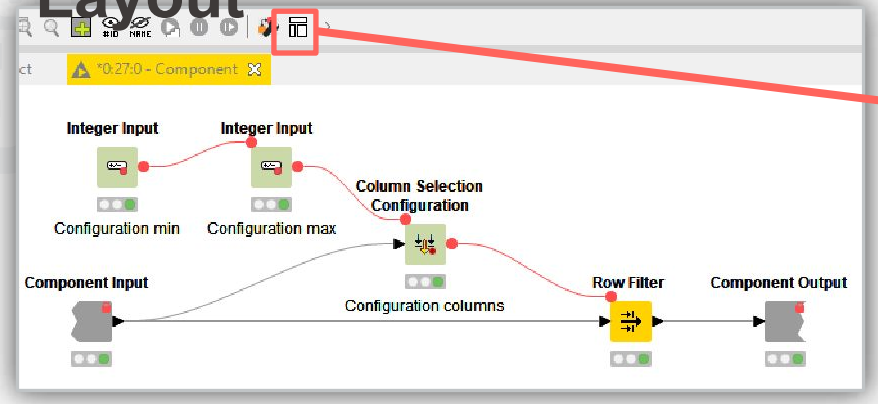
How can you Use a Shared Component?

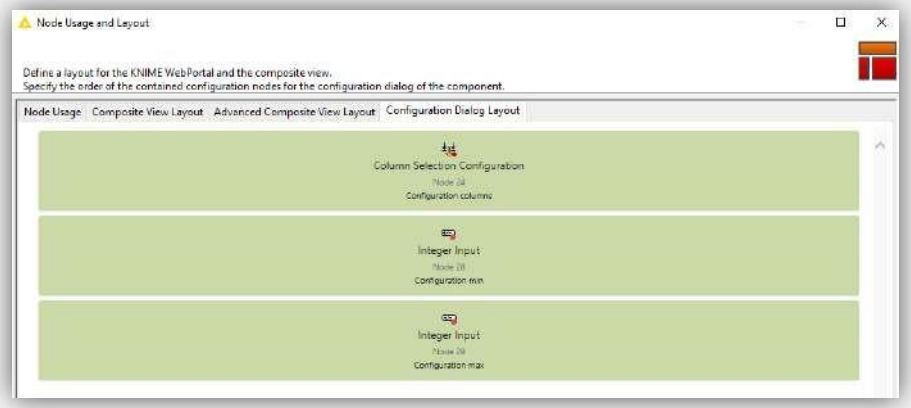
- To use a Shared Component, drag and drop it to the workflow editor
- Instances of Shared Components can be updated either manually or when workflow is opened
- Shared Component can also be unlinked from its original location, which makes it editable in the workflow directly
- Update Shared Components by overwriting them



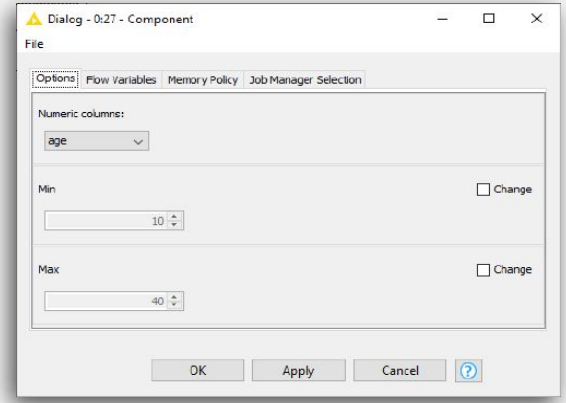


Configuration Dialog Layout



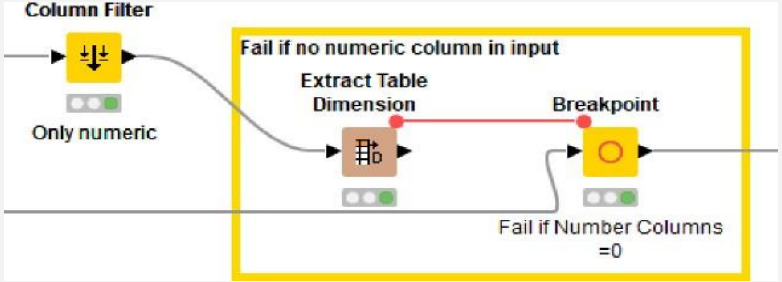


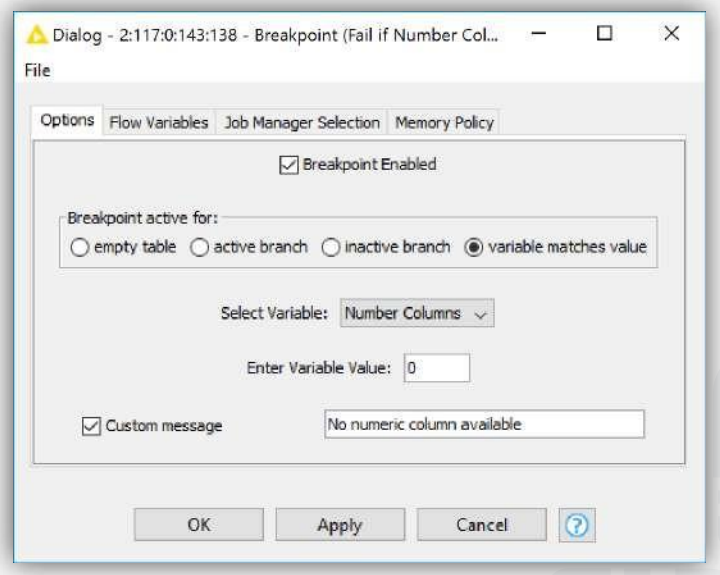
- Click layout button when inside component to modify the order of the setting options in configuration window of the component



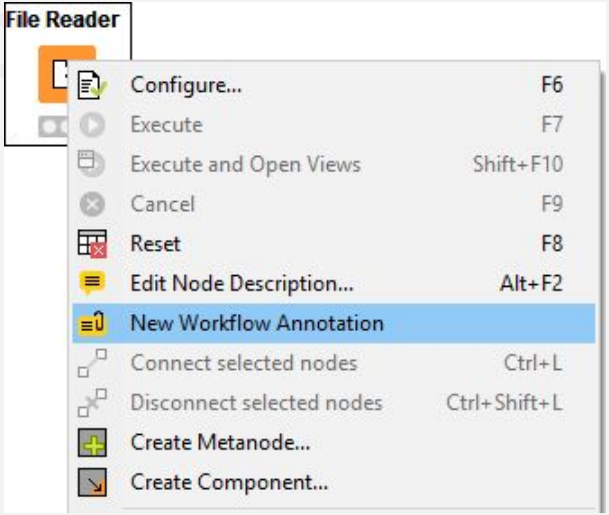
Breakpoint

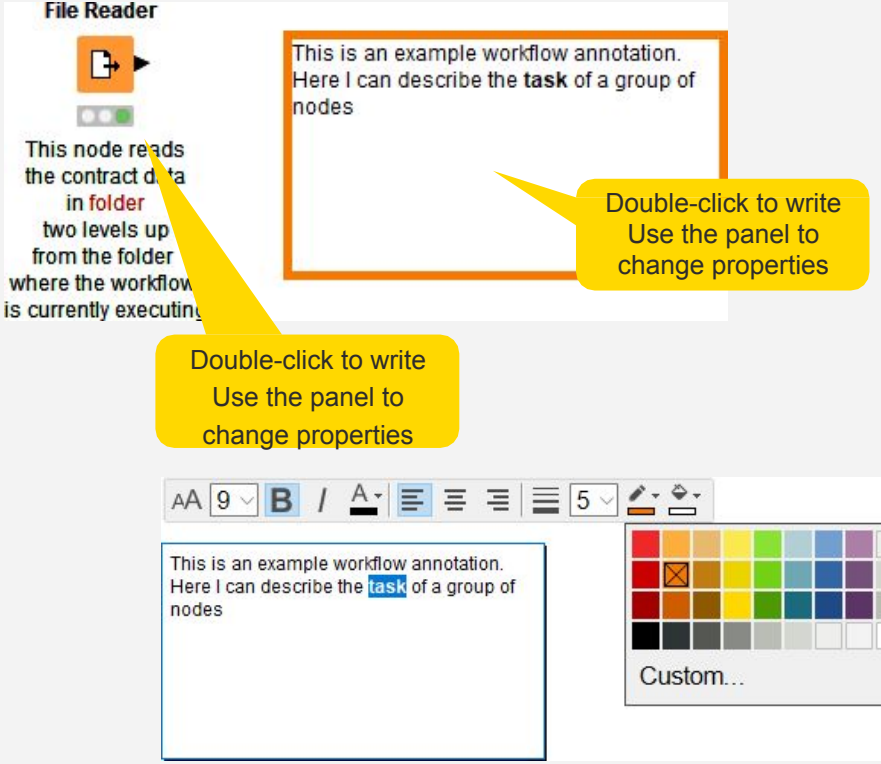
- Stops execution of a workflow branch
- Useful to stop the execution of a component and provide a custom error message
- Execution stops based on the selected condition:
  - Empty table
  - Active/Inactive branch
  - Flow Variable value





Comments & Annotations



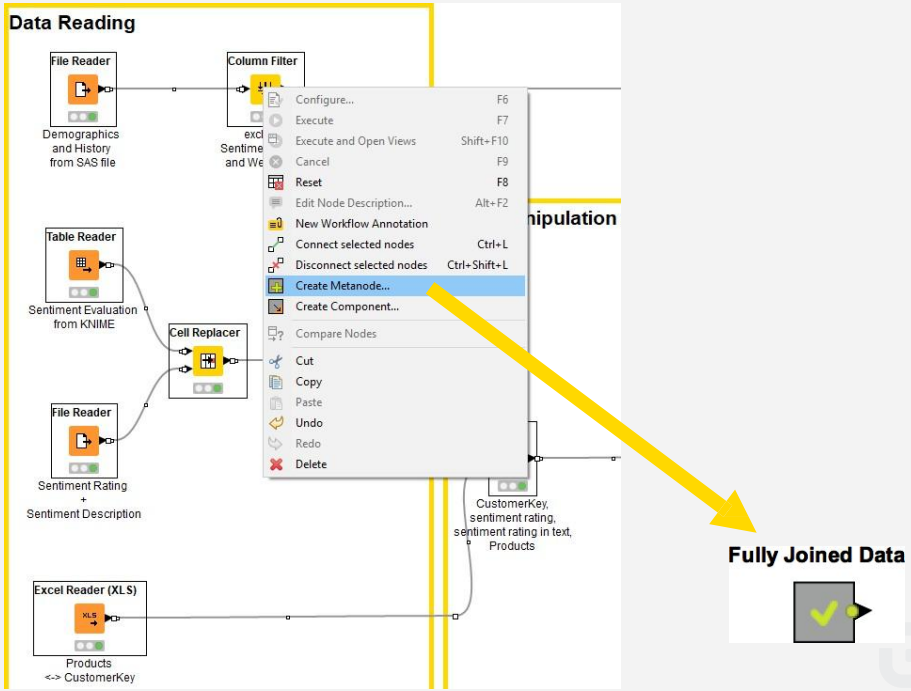


YouTube KNIME TV Channel:

https://youtu.be/AHURYB_O8sA

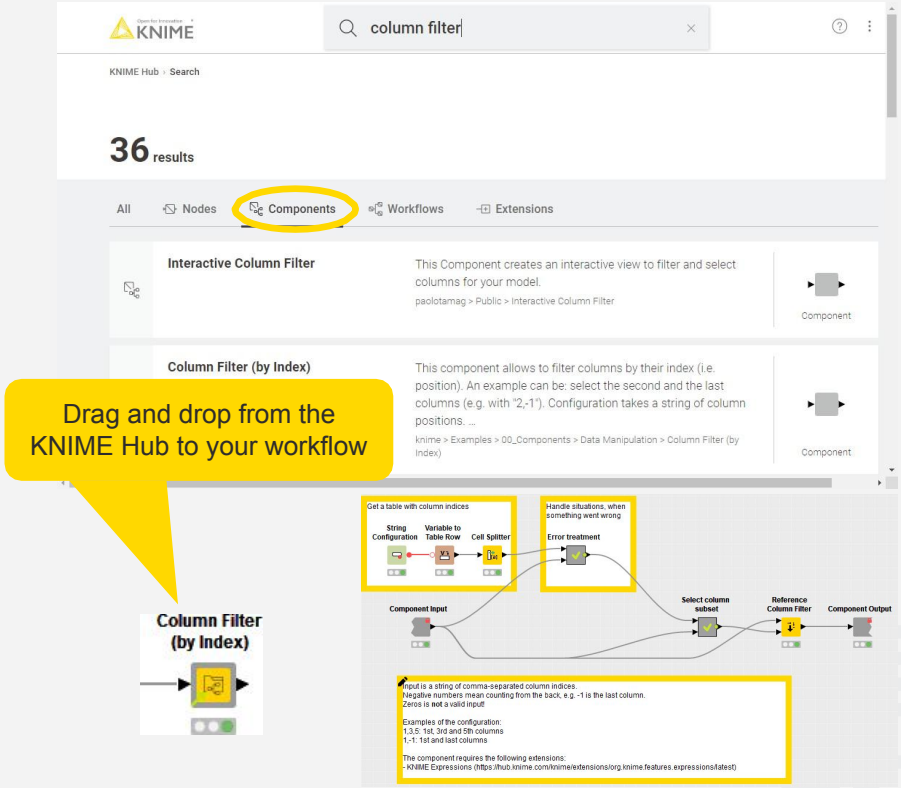
Workflow Organisation – Good Practices

- Workflow annotations
- Node labels
- Metanodes
  - Right click -> Create Metanode...
  - Organize workflow by task
  - Hide complexity & improve readability



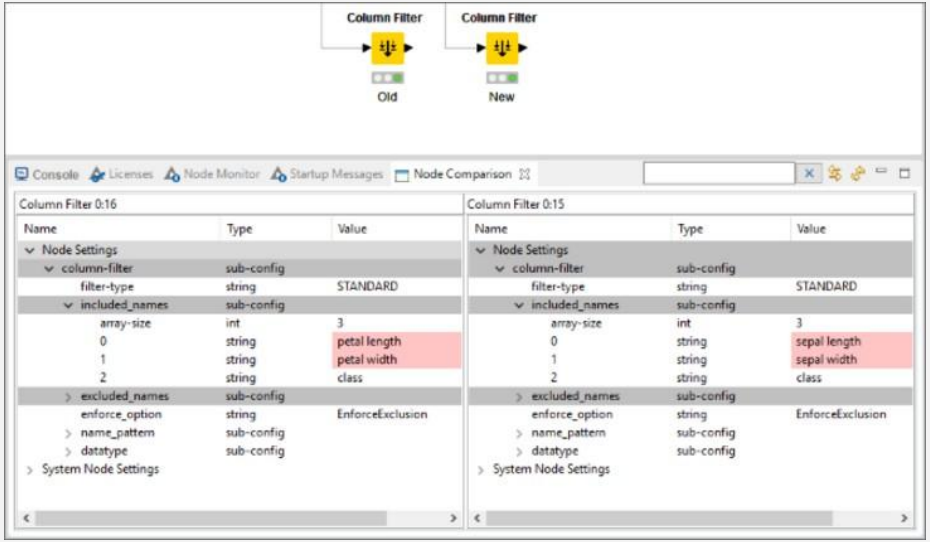
Workflow Organisation – Components

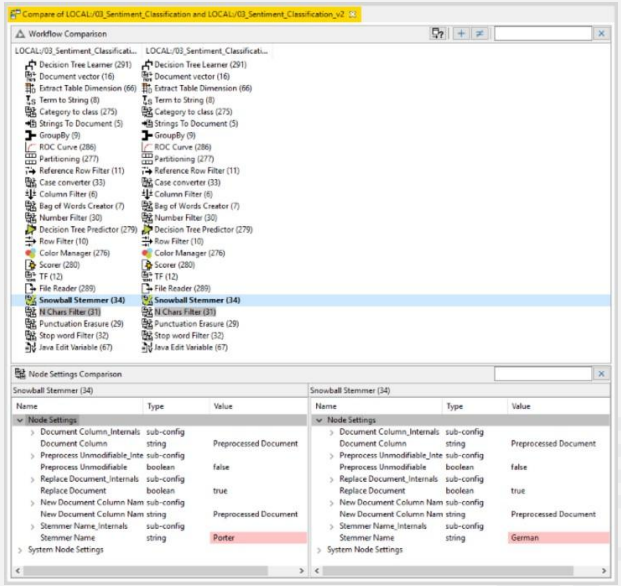
- Component encapsulates a reusable functionality as a KNIME workflow
- Components can be configured as any KNIME nodes
- Access and share components on the KNIME Hub



KNIME Workflow Diff

- Automates identification and comparison of nodes in a workflow, metanodes, and two different workflows
- Identifies insertions, deletions, substitutions, and parameter changes



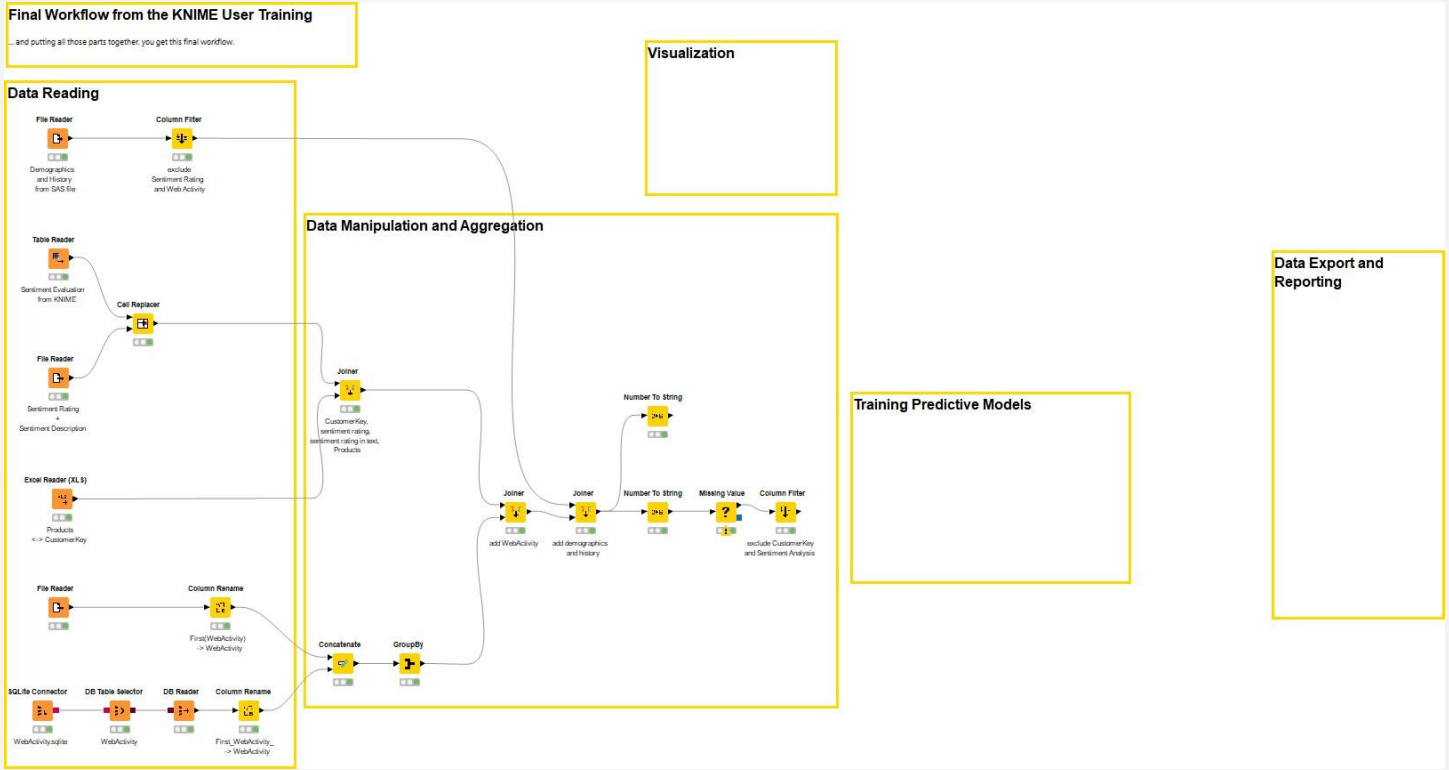


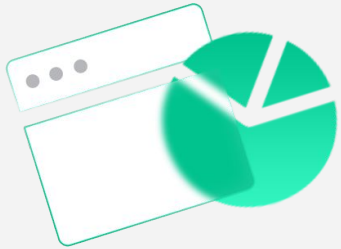
Data Manipulation Exercise

Start with exercise *Data Manipulation*

- Join all data into one table using a series of joiner nodes (use "Customer Key" as the joining column)
- Filter out duplicate rows
- Clean up and document your workflow using annotations, node labels, and metanodes

Today's Example





Terima Kasih