Open for Innovation
KNIME

[L2-DS] KNIME Analytics Platform for Data Scientists: Advanced

KNIME AG

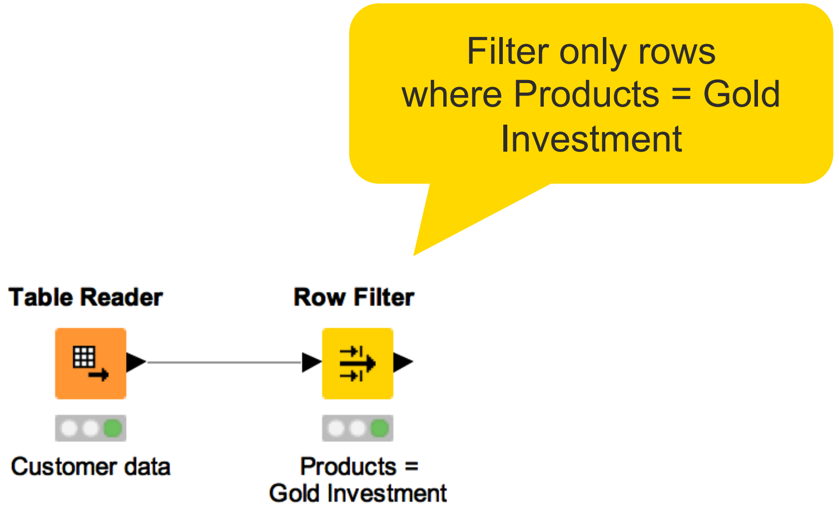
Flow Variables

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


Filter only rows where Products = Gold Investment
Table Reader
Row Filter
Customer data
Products = Gold Investment

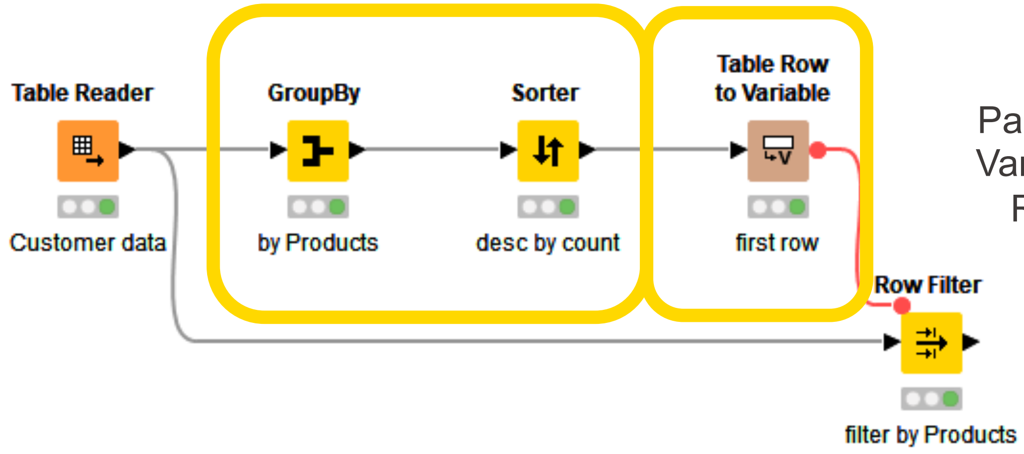
Flow Variables: Usage Example

- 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

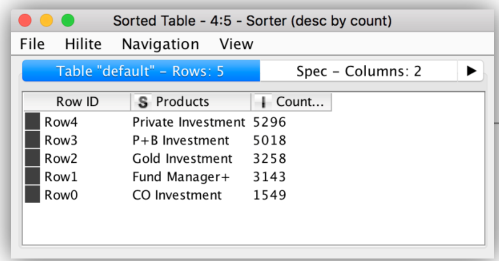
Create a Flow Variable containing the name and count of the most popular product



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

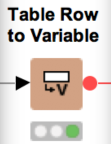


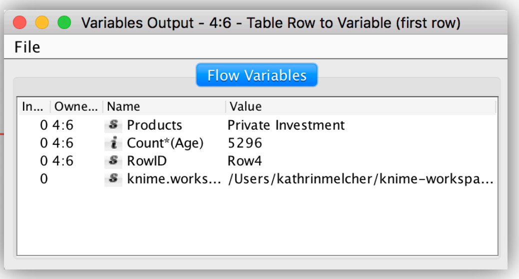
Sorted Table - 4:5 - Sorter (desc by count)

File Hilite Navigation View

Table "default" – Rows: 5 | Spec – Columns: 2

Row ID	S Products	I Count...
Row4	Private Investment	5296
Row3	P+B Investment	5018
Row2	Gold Investment	3258
Row1	Fund Manager+	3143
Row0	CO Investment	1549



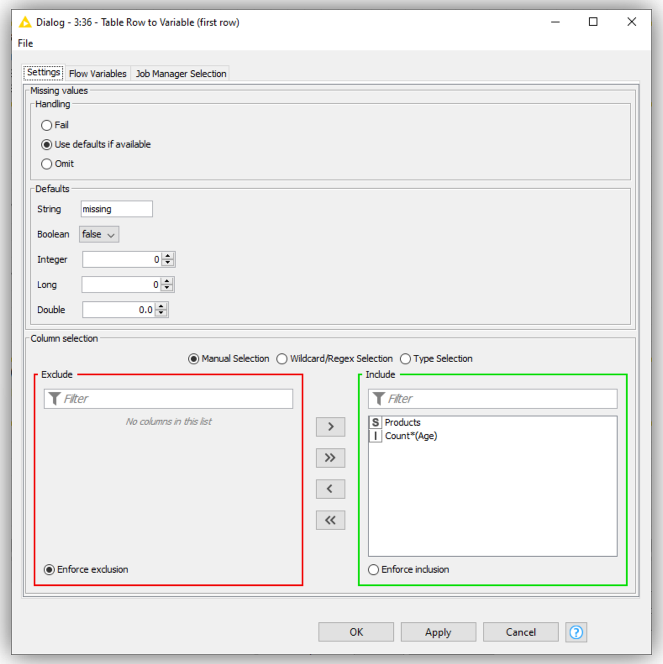


Variables Output - 4:6 - Table Row to Variable (first row)

File

Flow Variables

In...	Owne...	Name	Value
0	4:6	Products	Private Investment
0	4:6	Count*(Age)	5296
0	4:6	RowID	Row4
0		knime.works...	/Users/kathrinmelcher/knime-workspa...



Dialog - 3:36 - Table Row to Variable (first row)

File

Settings | Flow Variables | Job Manager Selection

Missing values

Handling: Fail [✓ Use defaults if available] Omit

Defaults

String: missing
Boolean: false
Integer: 0
Long: 0
Double: 0.0

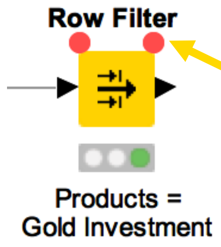
Column selection

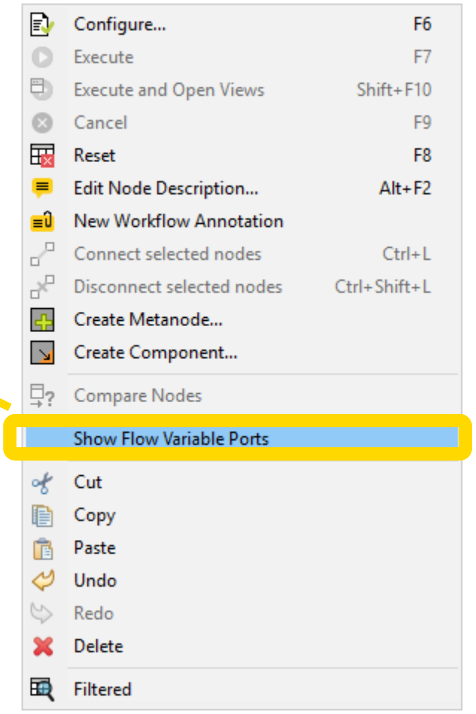
[✓ Manual Selection] Wildcard/Regex Selection Type Selection

Exclude: No columns in this list — [✓ Enforce exclusion]

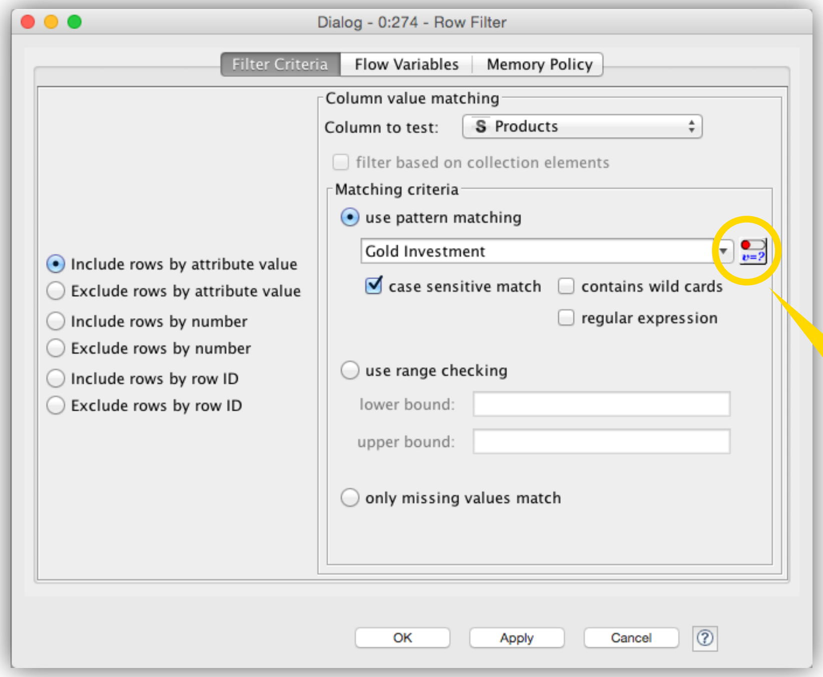
Include: S Products, I Count*(Age) — Enforce inclusion

Flow Variable Ports





Apply a Flow Variable (Button)



Dialog - 0:274 - Row Filter

Filter Criteria | Flow Variables | Memory Policy

- [✓ Include rows by attribute value]
- Exclude rows by attribute value
- Include rows by number
- Exclude rows by number
- Include rows by row ID
- Exclude rows by row ID

Column value matching

Column to test: Products

☐ filter based on collection elements

Matching criteria

[✓ use pattern matching]

Gold Investment

☑ case sensitive match ☐ contains wild cards

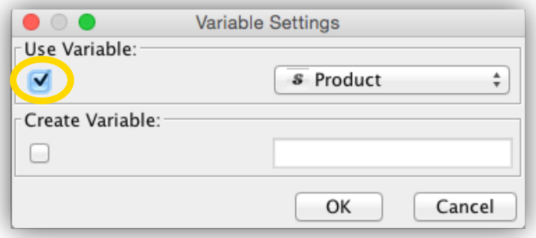
☐ regular expression

use range checking

lower bound:

upper bound:

only missing values match



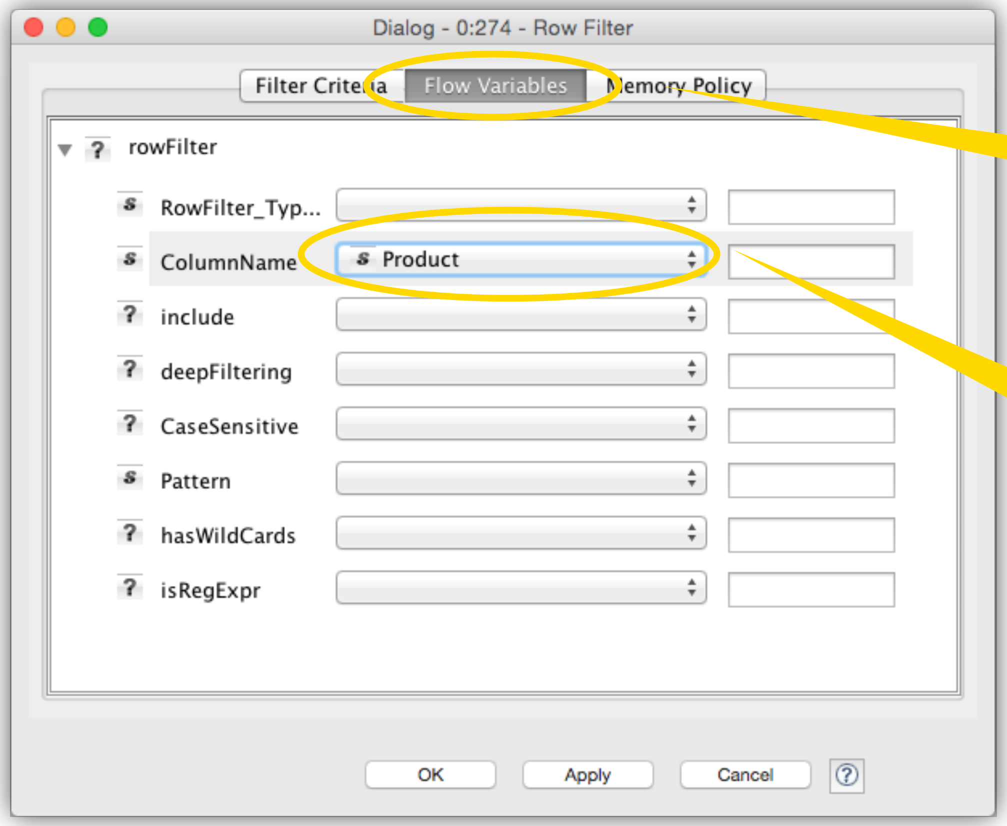
Variable Settings

Use Variable: ☑ Product

Create Variable: ☐

The Flow Variable button

Apply a Flow Variable (Advanced)



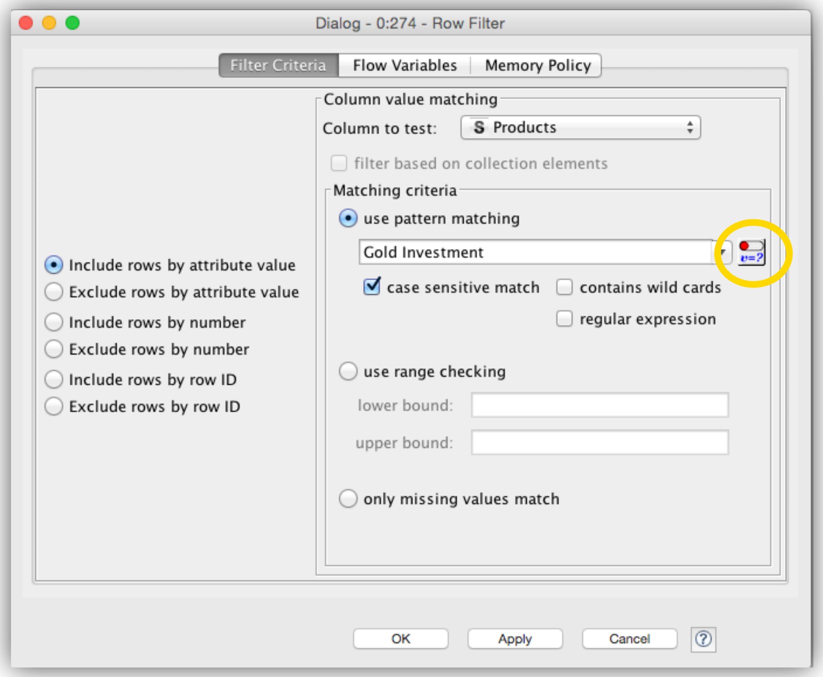
rowFilter

RowFilter_Typ...		
ColumnName	Product	
include		
deepFiltering		
CaseSensitive		
Pattern		
hasWildCards		
isRegExpr		

The Flow Variables tab

List of available Flow Variables

Create a Flow Variable (Button)


Dialog - 0:274 - Row Filter

Filter Criteria | Flow Variables | Memory Policy

- [✓ Include rows by attribute value]
- Exclude rows by attribute value
- Include rows by number
- Exclude rows by number
- Include rows by row ID
- Exclude rows by row ID

Column value matching

Column to test: Products

☐ filter based on collection elements

Matching criteria

[✓ use pattern matching]

Gold Investment

☑ case sensitive match ☐ contains wild cards

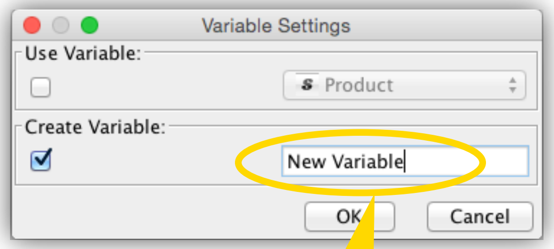
☐ regular expression

use range checking

lower bound:

upper bound:

only missing values match


Variable Settings

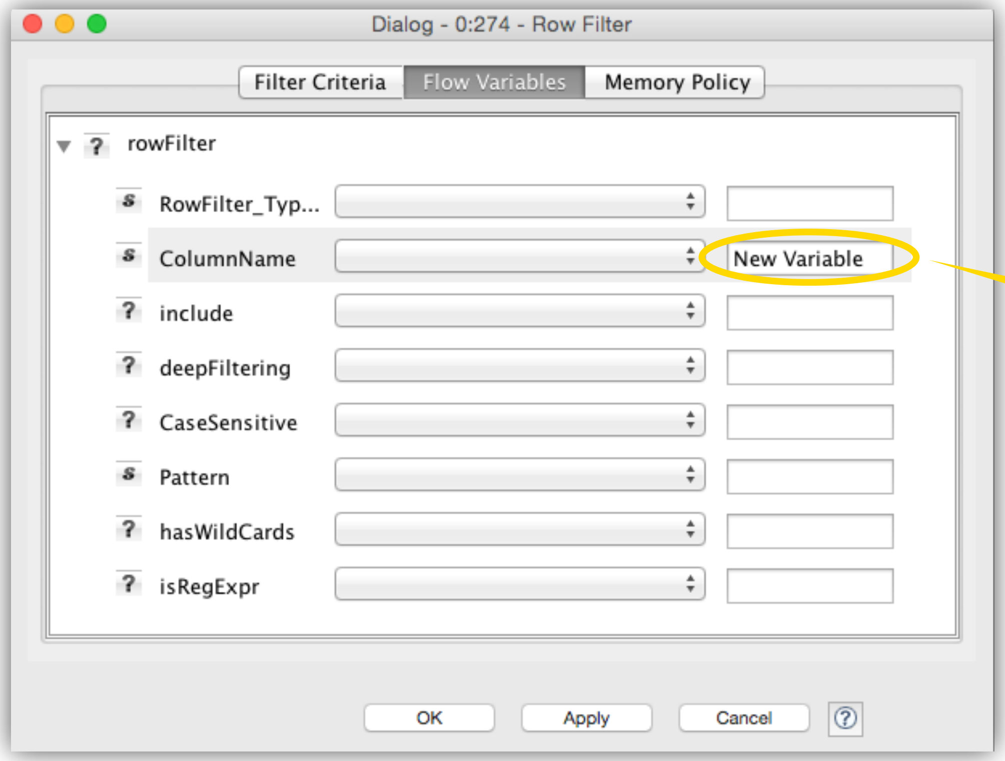
Use Variable: ☐ Product

Create Variable: ☑ New Variable

Name of the new Flow Variable

Create a Flow Variable (Advanced)

Converting a setting value into a Flow Variable



Dialog - 0:274 - Row Filter

Filter Criteria | Flow Variables | Memory Policy

rowFilter

RowFilter_Typ...		
ColumnName		New Variable
include		
deepFiltering		
CaseSensitive		
Pattern		
hasWildCards		
isRegExpr		

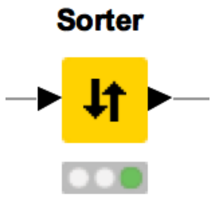
Name of the new Flow Variable

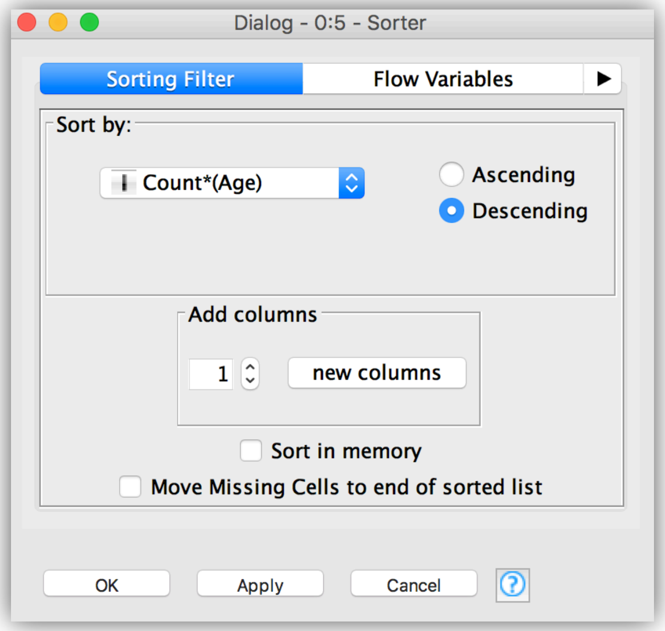
Key Features: Flow Variables

- 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

Sorter

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



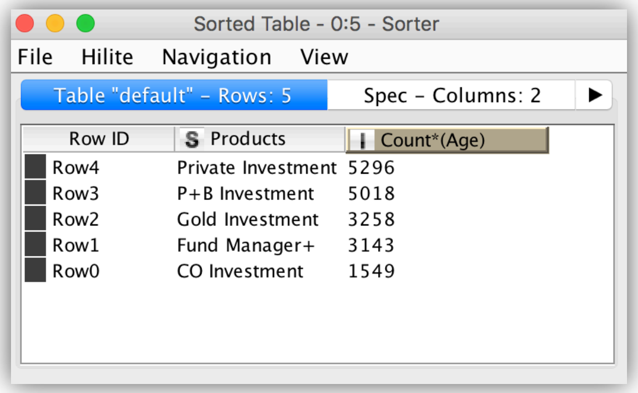


Sort by: Count*(Age) [ ] Ascending [✓] Descending

Add columns: 1 new columns

[ ] Sort in memory

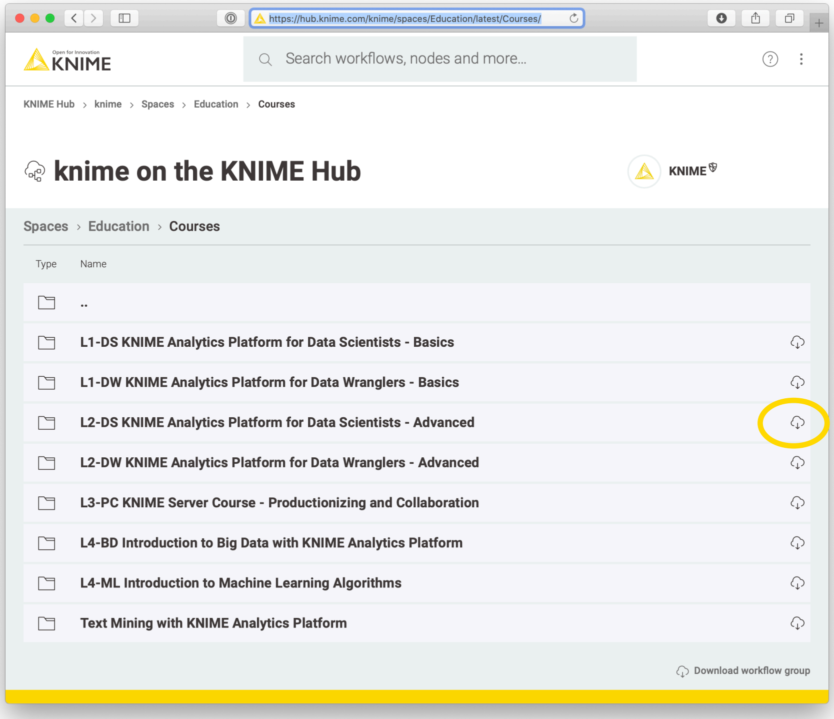
[ ] Move Missing Cells to end of sorted list



Row ID	Products	Count*(Age)
Row4	Private Investment	5296
Row3	P+B Investment	5018
Row2	Gold Investment	3258
Row1	Fund Manager+	3143
Row0	CO Investment	1549

Exercise Session 1:

- Download the course material from the KNIME Hub https://hub.knime.com/knime/spaces/Education/latest/Courses/


https://hub.knime.com/knime/spaces/Education/latest/Courses/

Search workflows, nodes and more...

KNIME Hub › knime › Spaces › Education › Courses

knime on the KNIME Hub

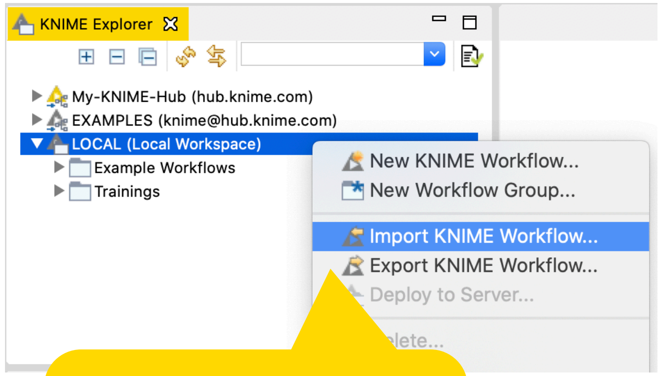
Spaces › Education › Courses

Type	Name
	..
	L1-DS KNIME Analytics Platform for Data Scientists - Basics
	L1-DW KNIME Analytics Platform for Data Wranglers - Basics
	L2-DS KNIME Analytics Platform for Data Scientists - Advanced
	L2-DW KNIME Analytics Platform for Data Wranglers - Advanced
	L3-PC KNIME Server Course - Productionizing and Collaboration
	L4-BD Introduction to Big Data with KNIME Analytics Platform
	L4-ML Introduction to Machine Learning Algorithms
	Text Mining with KNIME Analytics Platform

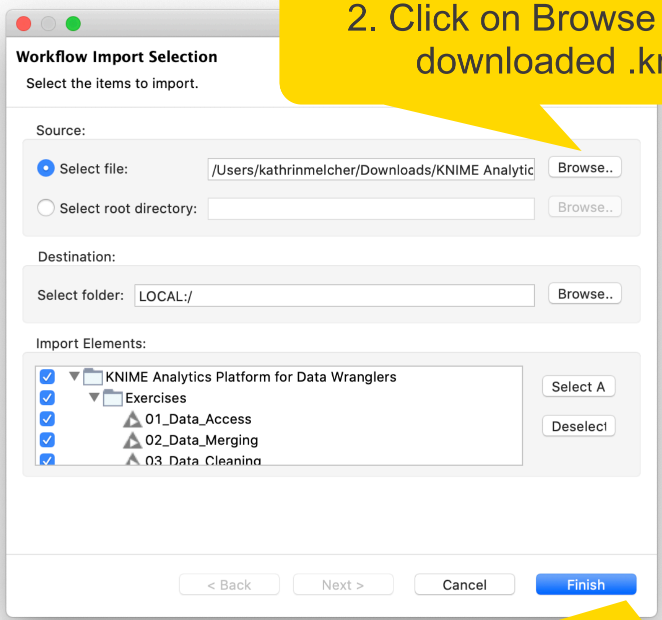
Download workflow group

Exercise Session 1

- Import the course material to KNIME Analytics Platform



1. Right click on LOCAL and select Import KNIME Workflow….



Workflow Import Selection

Select the items to import.

Source:

Select file: /Users/kathrinmelcher/Downloads/KNIME Analytic

Select root directory:

Destination:

Select folder: LOCAL:/

Import Elements:

- KNIME Analytics Platform for Data Wranglers
  - Exercises
    - 01_Data_Access
    - 02_Data_Merging
    - 03_Data_Cleaning

2. Click on Browse and select downloaded .knar file

3. Click on Finish

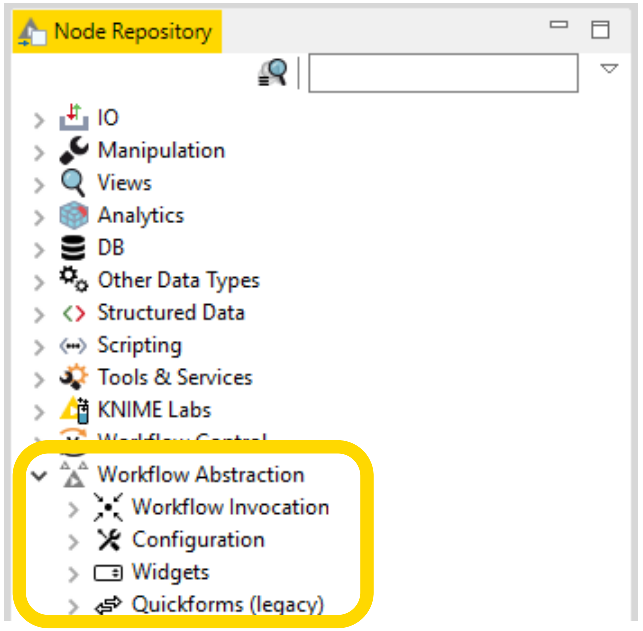
Flow Variables Exercise: Activity I

Start with exercise: *Flow Variables, Activity I*

Filter the customer data to

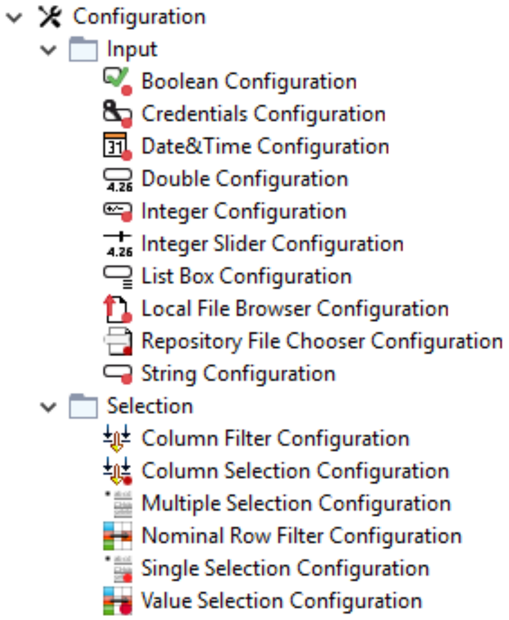
1. customers of the "Gold Investment" product
2. customers of the most common product in the data

Configuration Nodes for Variable Creation and Output



Node Repository

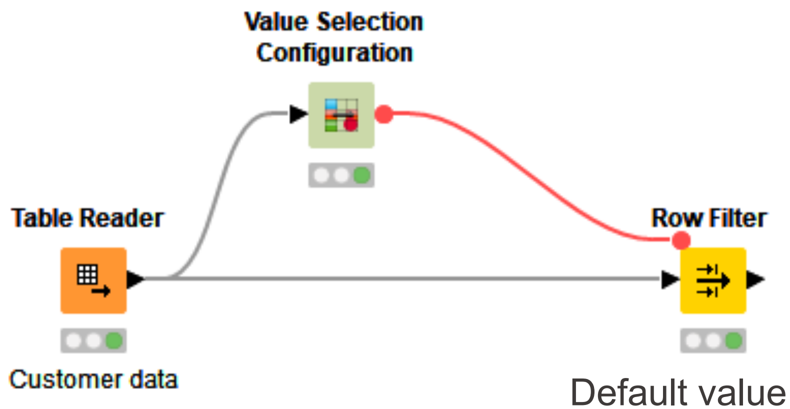
- IO
- Manipulation
- Views
- Analytics
- DB
- Other Data Types
- Structured Data
- Scripting
- Tools & Services
- KNIME Labs
- Workflow Control
- Workflow Abstraction
  - Workflow Invocation
  - Configuration
  - Widgets
  - Quickforms (legacy)



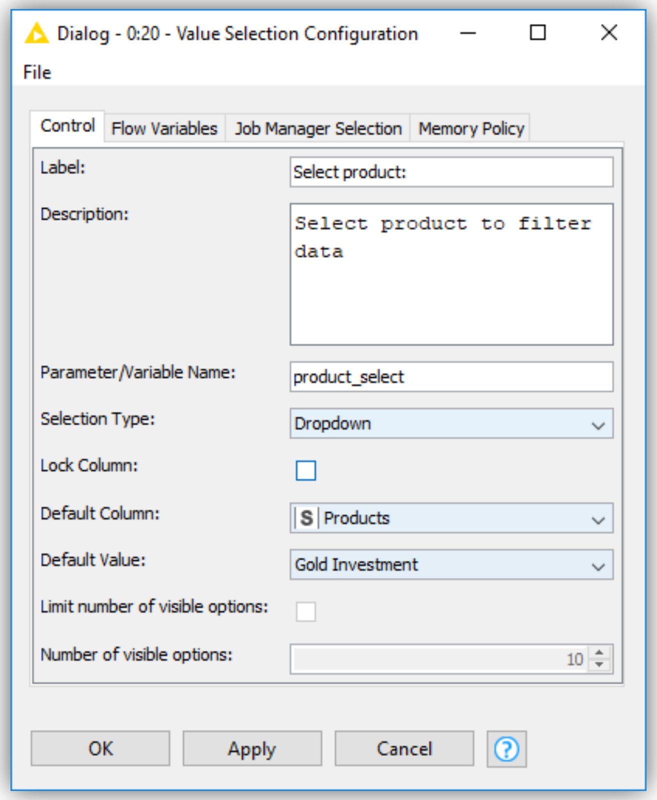
- Configuration
  - Input
    - Boolean Configuration
    - Credentials Configuration
    - Date&Time Configuration
    - Double Configuration
    - Integer Configuration
    - Integer Slider Configuration
    - List Box Configuration
    - Local File Browser Configuration
    - Repository File Chooser Configuration
    - String Configuration
  - Selection
    - Column Filter Configuration
    - Column Selection Configuration
    - Multiple Selection Configuration
    - Nominal Row Filter Configuration
    - Single Selection Configuration
    - Value Selection Configuration

Configuration Node Configuration

Use Configurations to create Flow Variables



Default value is selected (here P+B Investment)



Label:	Select product:
Description:	Select product to filter data
Parameter/Variable Name:	product_select
Selection Type:	Dropdown
Lock Column:	☐
Default Column:	S Products
Default Value:	Gold Investment
Limit number of visible options:	☐
Number of visible options:	10

Simple Configuration of Component

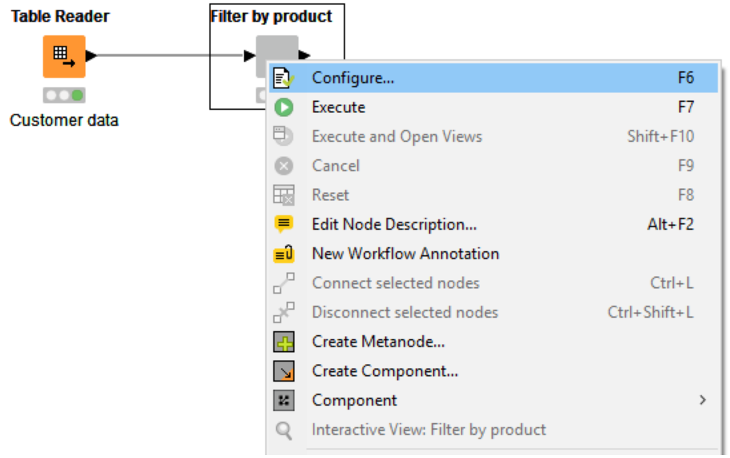
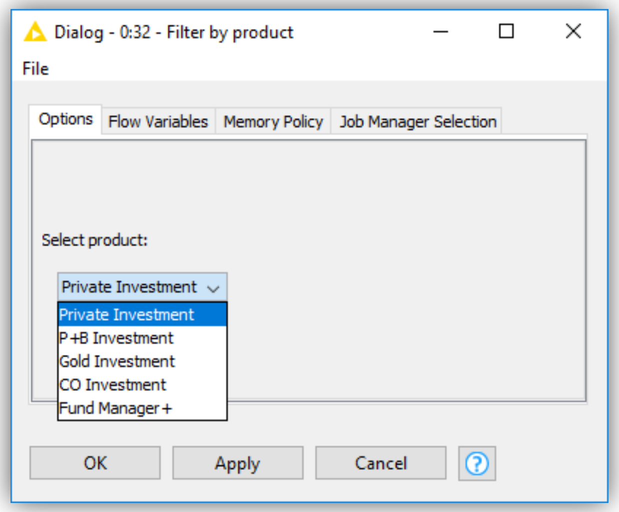


Table Reader
Customer data
Filter by product
Configure... F6
Execute F7
Execute and Open Views Shift+F10
Cancel F9
Reset F8
Edit Node Description... Alt+F2
New Workflow Annotation
Connect selected nodes Ctrl+L
Disconnect selected nodes Ctrl+Shift+L
Create Metanode...
Create Component...
Component
Interactive View: Filter by product



Select product:

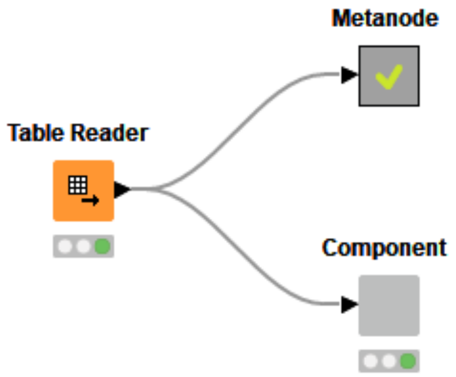
Private Investment

- Private Investment
- P+B Investment
- Gold Investment
- CO Investment
- Fund Manager+

- Double click a Component to configure
- For use in WebPortal, replace Configuration nodes by Widget nodes

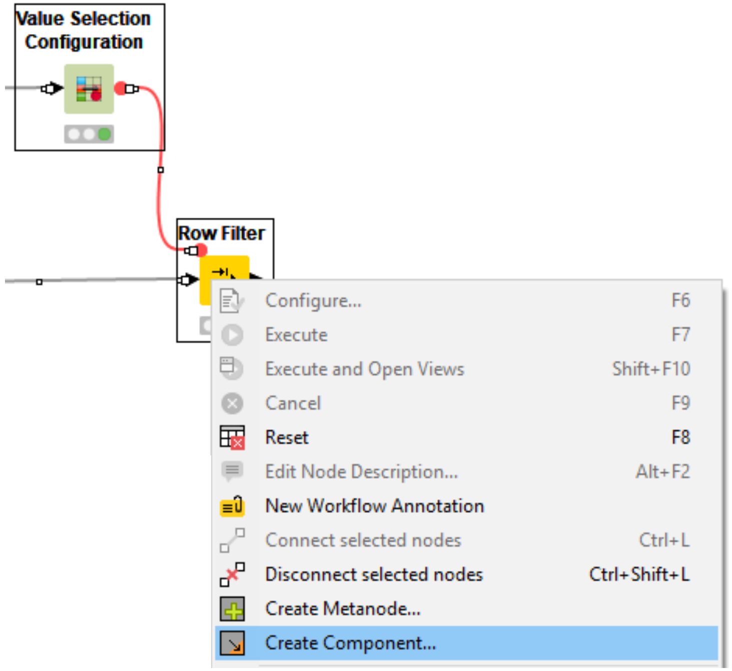
Components

- Encapsulates a functionality for reuse and sharing
- Components main features:
  - Local Flow Variable scope
  - Configurable via Configuration nodes
- Key to advanced functionality in KNIME products:
  - Component corresponds to a KNIME WebPortal page
  - Configurations on a WebPortal page are defined using Widget nodes
  - Possibility to be shared via KNIME Hub



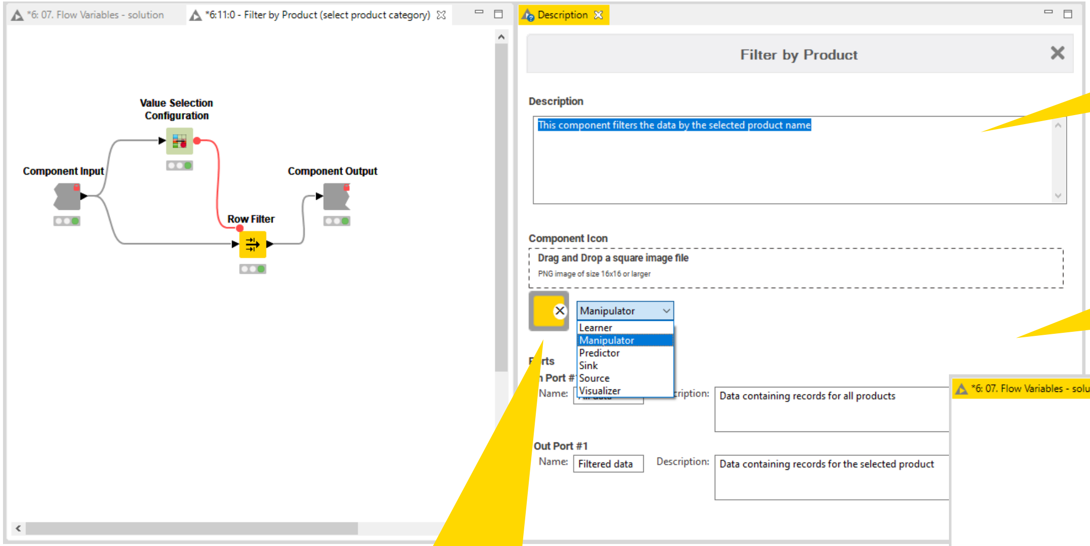
Create a Component

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


Value Selection Configuration
Row Filter
Configure... F6
Execute F7
Execute and Open Views Shift+F10
Cancel F9
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...

Component Description

Make your component look like a KNIME node


*6: 07. Flow Variables - solution
*6:11:0 - Filter by Product (select product category)
Value Selection Configuration
Component Input
Component Output
Row Filter
Description

Filter by Product

Description

This component filters the data by the selected product name

Component Icon

Drag and Drop a square image file

PNG image of size 16x16 or larger

Manipulator

- Learner
- Manipulator
- Predictor
- Sink
- Source
- Visualizer

Ports

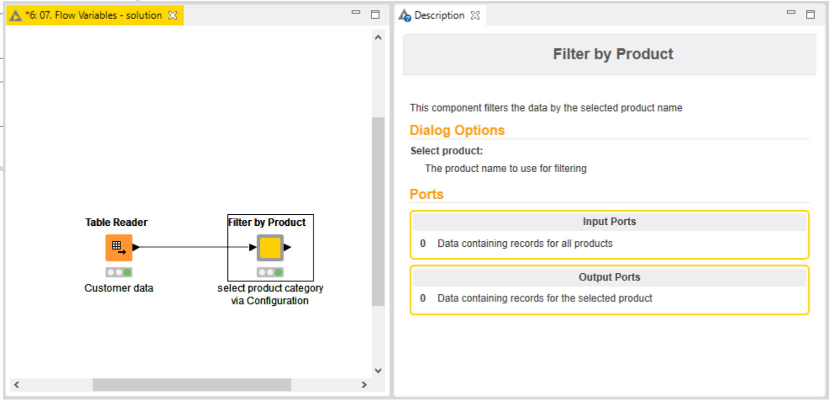
In Port #1 Name: All data Description: Data containing records for all products

Out Port #1 Name: Filtered data Description: Data containing records for the selected product

Add description of the component

Add description of the input and output ports

Add background color or icon


*6: 07. Flow Variables - solution
Description
Table Reader
Customer data
Filter by Product
select product category via Configuration

Filter by Product

This component filters the data by the selected product name

Dialog Options

Select product:

The product name to use for filtering

Ports

Input Ports

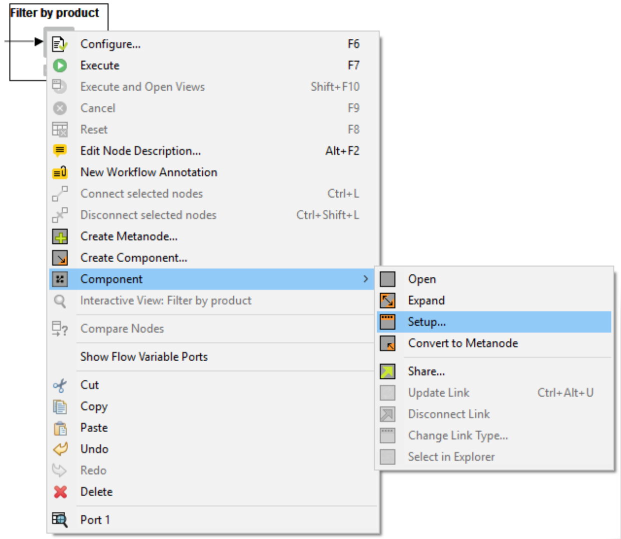
0 Data containing records for all products

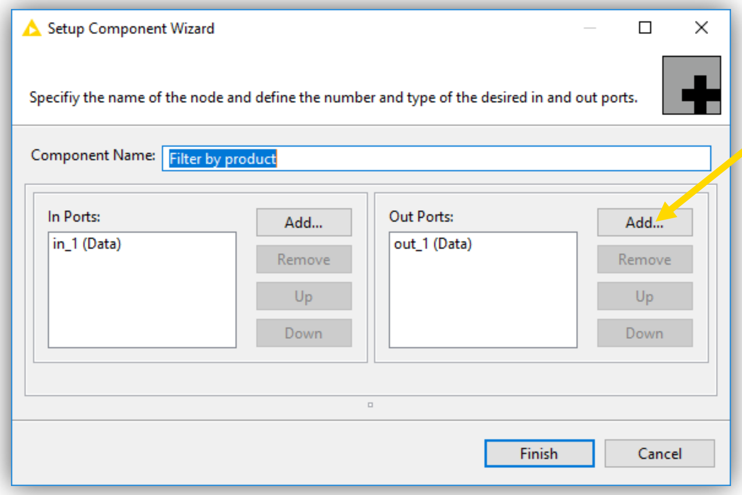
Output Ports

0 Data containing records for the selected product

Configure Component Ports

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



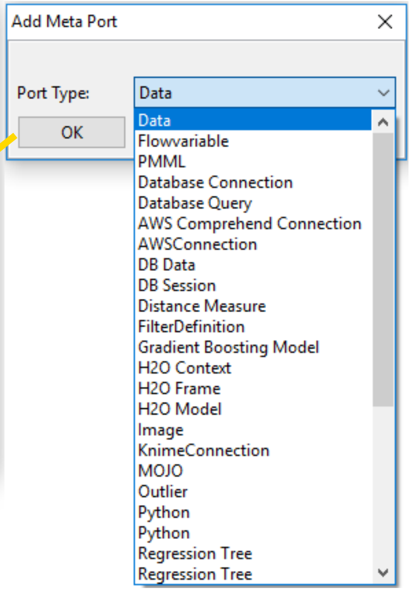

Setup Component Wizard

Specifiy the name of the node and define the number and type of the desired in and out ports.

Component Name: Filter by product

In Ports: in_1 (Data)

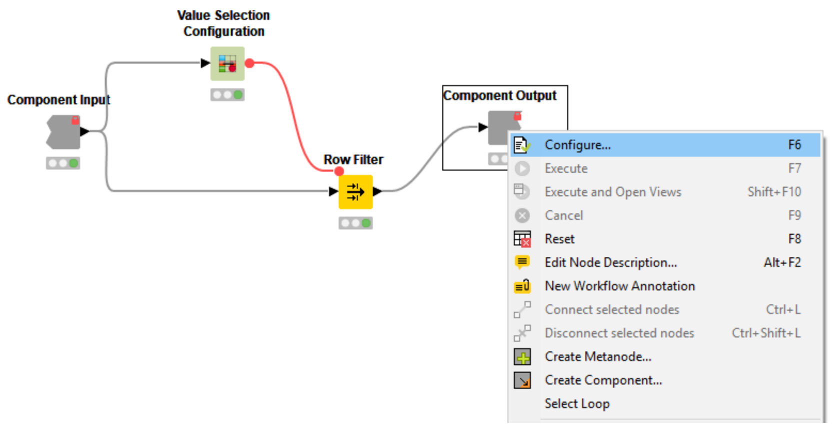
Out Ports: out_1 (Data)


Add Meta Port

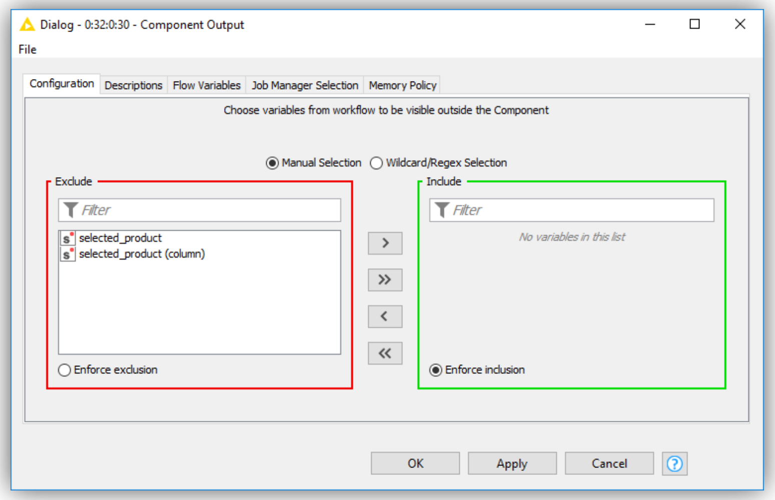
Port Type: Data

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



Value Selection Configuration
Component Input
Row Filter
Component Output
Configure... F6
Execute F7
Execute and Open Views Shift+F10
Cancel F9
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 Loop


Dialog - 0:32:0:30 - Component Output

File

Configuration | Descriptions | Flow Variables | Job Manager Selection | Memory Policy

Choose variables from workflow to be visible outside the Component

[✓ Manual Selection] Wildcard/Regex Selection

Exclude

Filter

selected_product
selected_product (column)

Enforce exclusion

Include

Filter

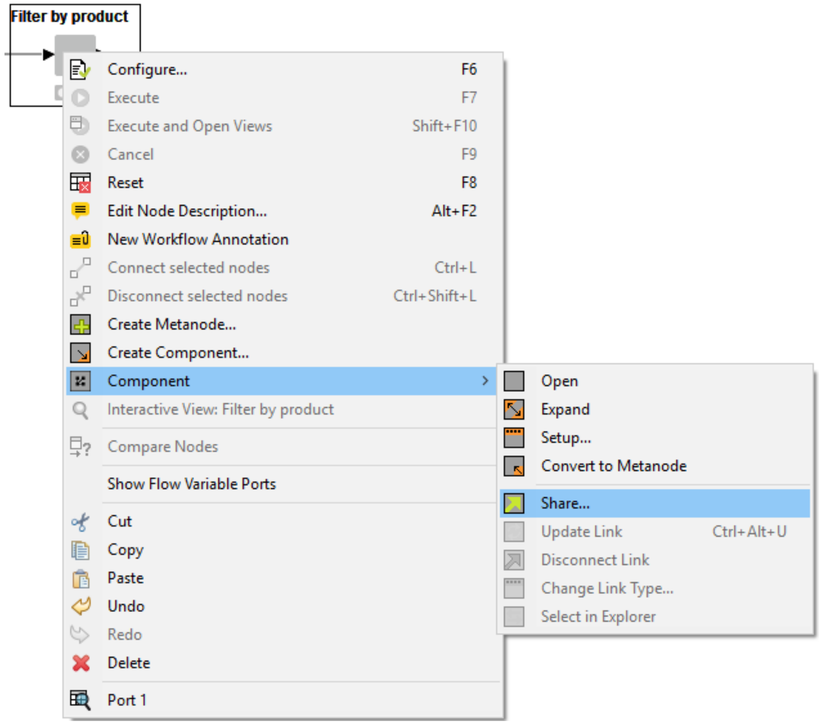
No variables in this list

[✓ Enforce inclusion]

OK Apply Cancel

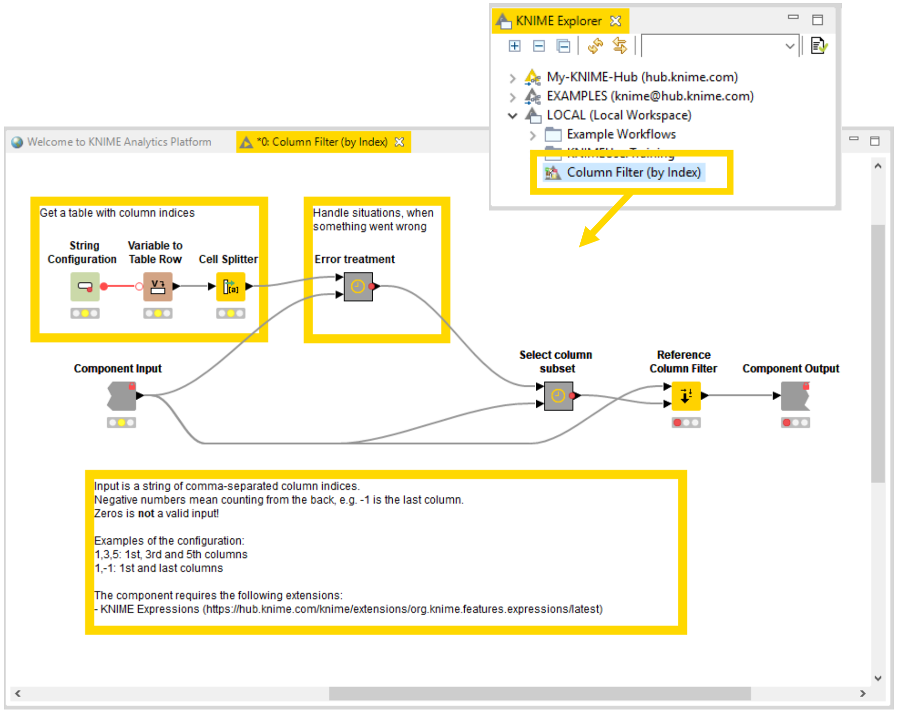
What is a Shared Component?

- 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



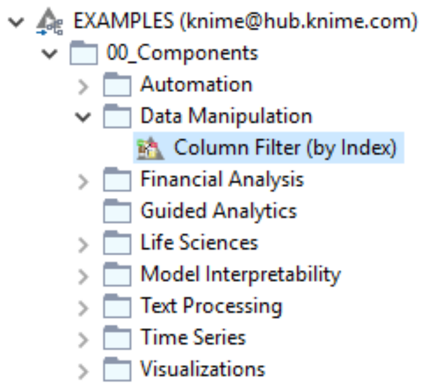
Input is a string of comma-separated column indices.
Negative numbers mean counting from the back, e.g. -1 is the last column.
Zeros is **not** a valid input!

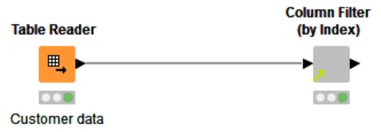
Examples of the configuration:
1,3,5: 1st, 3rd and 5th columns
1,-1: 1st and last columns

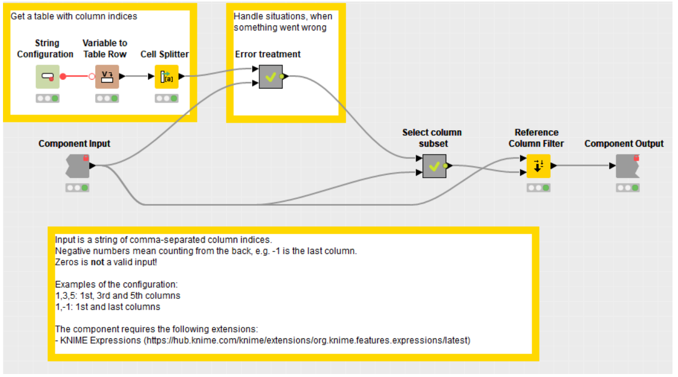
The component requires the following extensions:
- KNIME Expressions (https://hub.knime.com/knime/extensions/org.knime.features.expressions/latest)

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





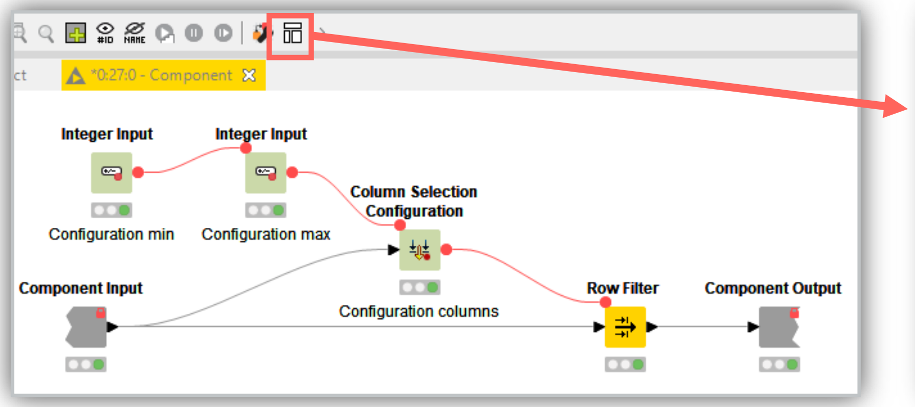


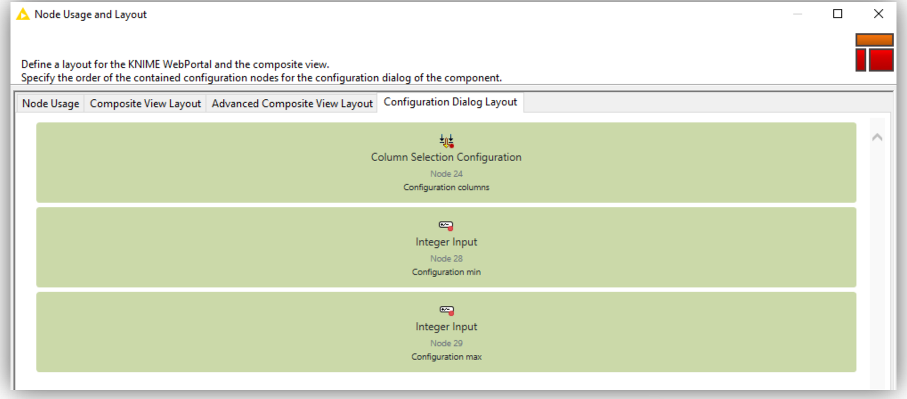
Input is a string of comma-separated column indices.
Negative numbers mean counting from the back, e.g. -1 is the last column.
Zeros is **not** a valid input!

Examples of the configuration:
1,3,5: 1st, 3rd and 5th columns
1,-1: 1st and last columns

The component requires the following extensions:
- KNIME Expressions (https://hub.knime.com/knime/extensions/org.knime.features.expressions/latest)

Configuration Dialog Layout




Node Usage and Layout

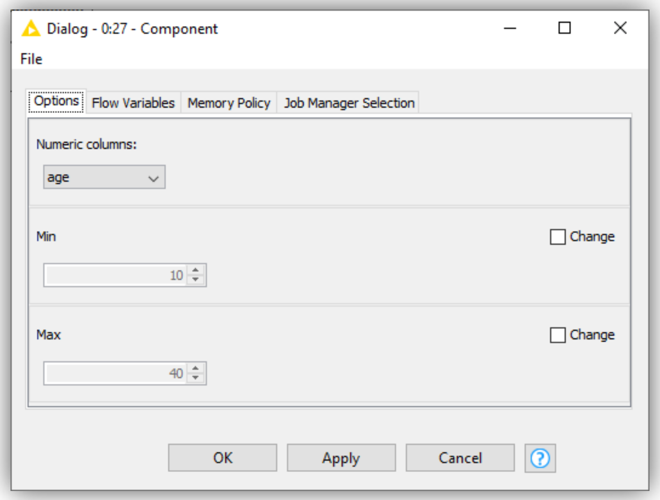
Define a layout for the KNIME WebPortal and the composite view.
Specify the order of the contained configuration nodes for the configuration dialog of the component.

Column Selection Configuration
Node 24
Configuration columns

Integer Input
Node 28
Configuration min

Integer Input
Node 29
Configuration max

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


Dialog - 0:27 - Component

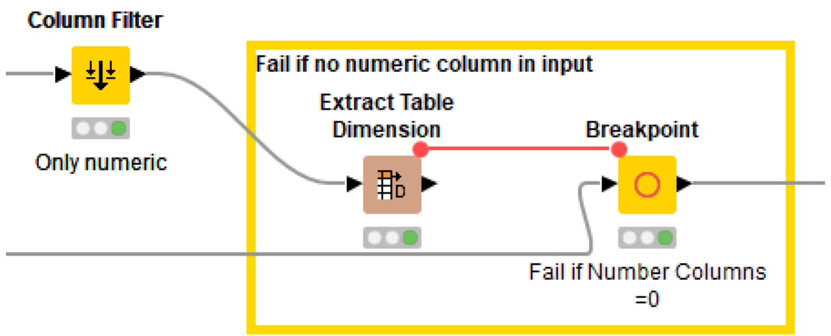
Numeric columns: age

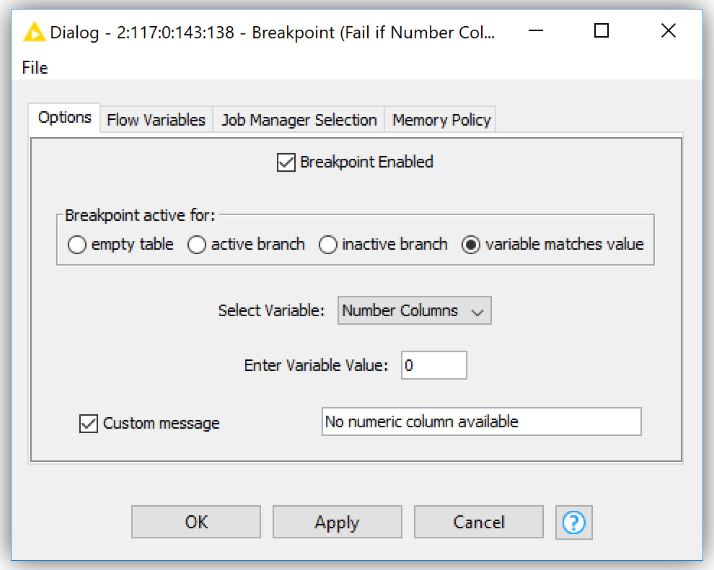
Min ☐ Change
10

Max ☐ Change
40

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





Breakpoint Enabled [✓]

Breakpoint active for: empty table / active branch / inactive branch / [✓ variable matches value]

Select Variable: Number Columns

Enter Variable Value: 0

Custom message [✓]: No numeric column available

Flow Variables Exercise: Activity II

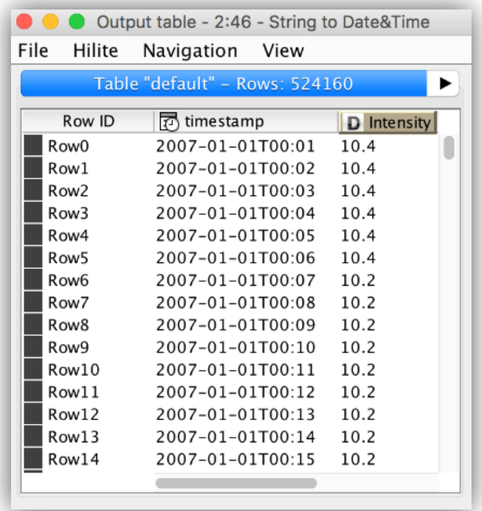
Start with exercise: *Flow Variables, Activity II*

- Create a component that allows a user to choose an investment product and filter the data by that product

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

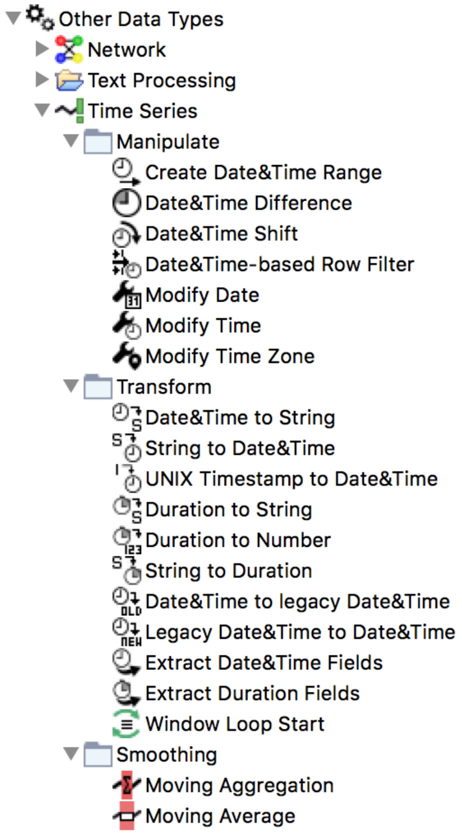


Output table - 2:46 - String to Date&Time

File Hilite Navigation View

Table "default" – Rows: 524160

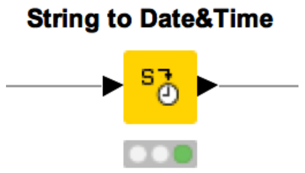
Row ID	timestamp	Intensity
Row0	2007-01-01T00:01	10.4
Row1	2007-01-01T00:02	10.4
Row2	2007-01-01T00:03	10.4
Row3	2007-01-01T00:04	10.4
Row4	2007-01-01T00:05	10.4
Row5	2007-01-01T00:06	10.4
Row6	2007-01-01T00:07	10.2
Row7	2007-01-01T00:08	10.2
Row8	2007-01-01T00:09	10.2
Row9	2007-01-01T00:10	10.2
Row10	2007-01-01T00:11	10.2
Row11	2007-01-01T00:12	10.2
Row12	2007-01-01T00:13	10.2
Row13	2007-01-01T00:14	10.2
Row14	2007-01-01T00:15	10.2

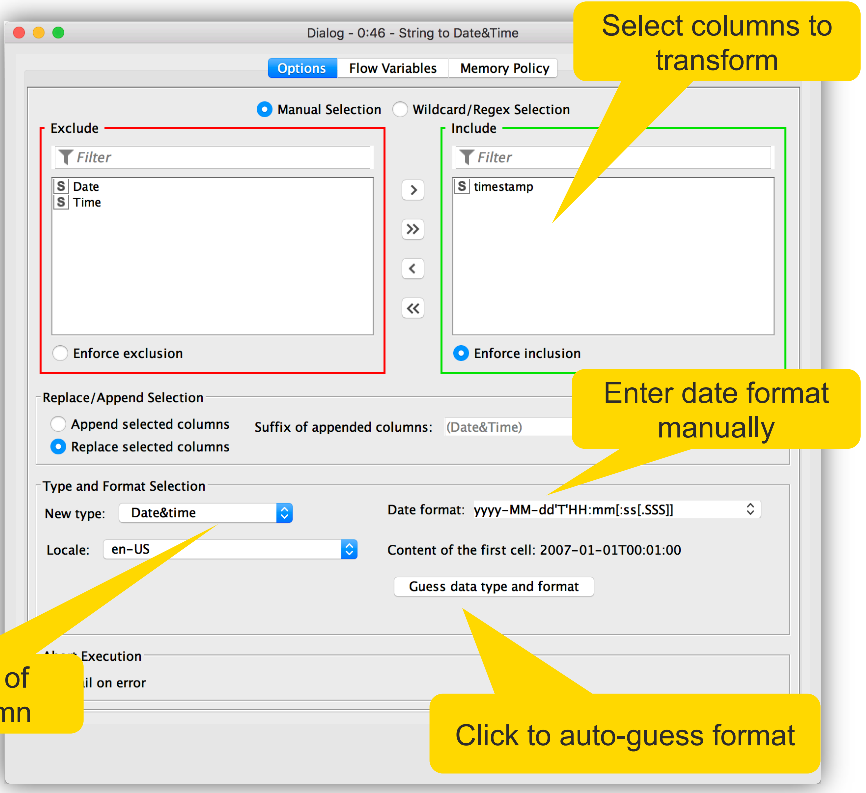


- Other Data Types
  - Network
  - Text Processing
  - Time Series
    - Manipulate
      - Create Date&Time Range
      - Date&Time Difference
      - Date&Time Shift
      - Date&Time-based Row Filter
      - Modify Date
      - Modify Time
      - Modify Time Zone
    - Transform
      - Date&Time to String
      - String to Date&Time
      - UNIX Timestamp to Date&Time
      - Duration to String
      - Duration to Number
      - String to Duration
      - Date&Time to legacy Date&Time
      - Legacy Date&Time to Date&Time
      - Extract Date&Time Fields
      - Extract Duration Fields
      - Window Loop Start
    - Smoothing
      - Moving Aggregation
      - Moving Average

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




Dialog - 0:46 - String to Date&Time
Options
Flow Variables
Memory Policy
Filter
Filter
Guess data type and format
Select columns to transform
Enter date format manually
Select type of output column
Click to auto-guess format

[✓ Manual Selection] Wildcard/Regex Selection

Exclude: Date, Time

Include: timestamp

Enforce exclusion [✓ Enforce inclusion]

Replace/Append Selection

Append selected columns [✓ Replace selected columns]

Suffix of appended columns: (Date&Time)

Type and Format Selection

New type: Date&time

Date format: yyyy-MM-dd'T'HH:mm[:ss[.SSS]]

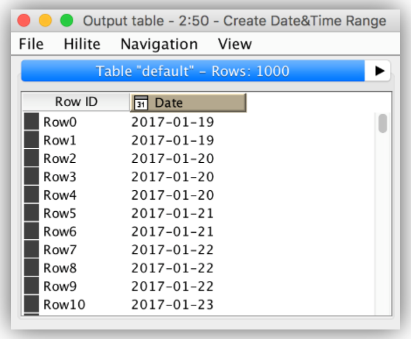
Locale: en-US

Content of the first cell: 2007-01-01T00:01:00

Abort Execution

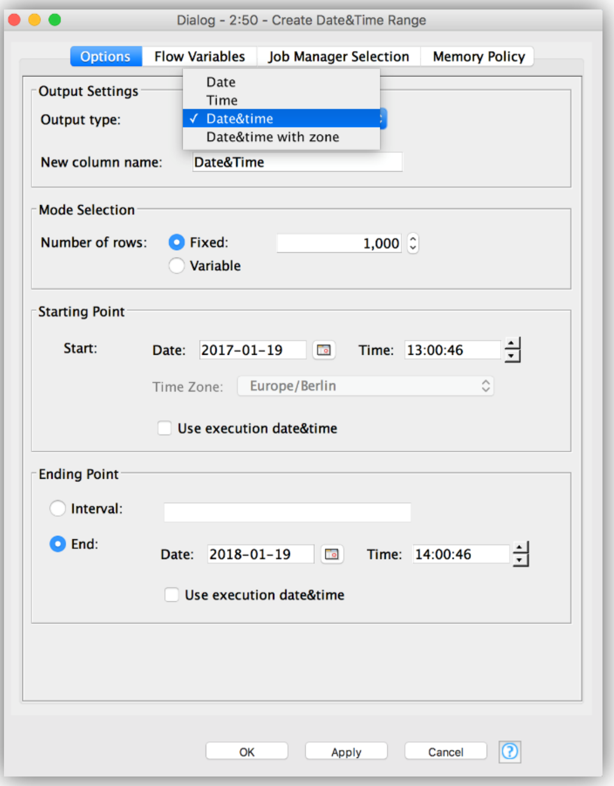
Fail on error

Date&Time – Data Types


Output table - 2:50 - Create Date&Time Range
File Hilite Navigation View
Table "default" – Rows: 1000

Row ID	Date
Row0	2017-01-19
Row1	2017-01-19
Row2	2017-01-20
Row3	2017-01-20
Row4	2017-01-20
Row5	2017-01-21
Row6	2017-01-21
Row7	2017-01-22
Row8	2017-01-22
Row9	2017-01-22
Row10	2017-01-23

Date



Output Settings

Output type: ✓ Date&time

New column name: Date&Time

Mode Selection

Number of rows: [● Fixed:] 1,000 ○ Variable

Starting Point

Start: Date: 2017-01-19 Time: 13:00:46

Time Zone: Europe/Berlin

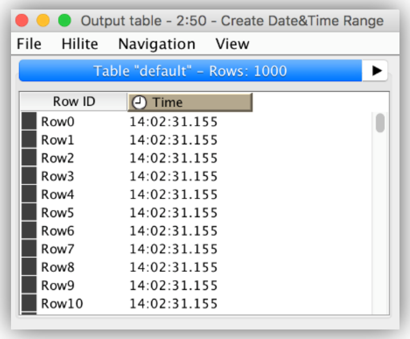
☐ Use execution date&time

Ending Point

○ Interval:

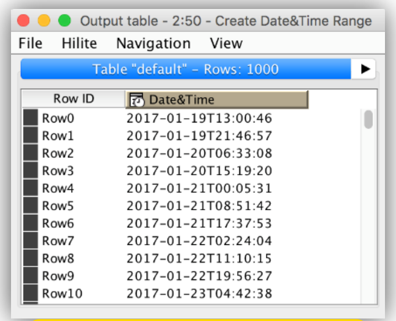
[● End:] Date: 2018-01-19 Time: 14:00:46

☐ Use execution date&time


Output table - 2:50 - Create Date&Time Range
File Hilite Navigation View
Table "default" – Rows: 1000

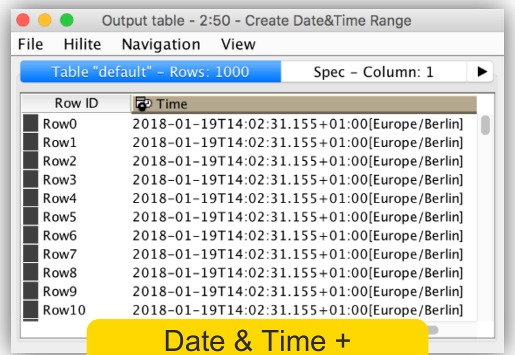
Row ID	Time
Row0	14:02:31.155
Row1	14:02:31.155
Row2	14:02:31.155
Row3	14:02:31.155
Row4	14:02:31.155
Row5	14:02:31.155
Row6	14:02:31.155
Row7	14:02:31.155
Row8	14:02:31.155
Row9	14:02:31.155
Row10	14:02:31.155

Time


Output table - 2:50 - Create Date&Time Range
File Hilite Navigation View
Table "default" – Rows: 1000

Row ID	Date&Time
Row0	2017-01-19T13:00:46
Row1	2017-01-19T21:46:57
Row2	2017-01-20T06:33:08
Row3	2017-01-20T15:19:20
Row4	2017-01-21T00:05:31
Row5	2017-01-21T08:51:42
Row6	2017-01-21T17:37:53
Row7	2017-01-22T02:24:04
Row8	2017-01-22T11:10:15
Row9	2017-01-22T19:56:27
Row10	2017-01-23T04:42:38

Date & Time

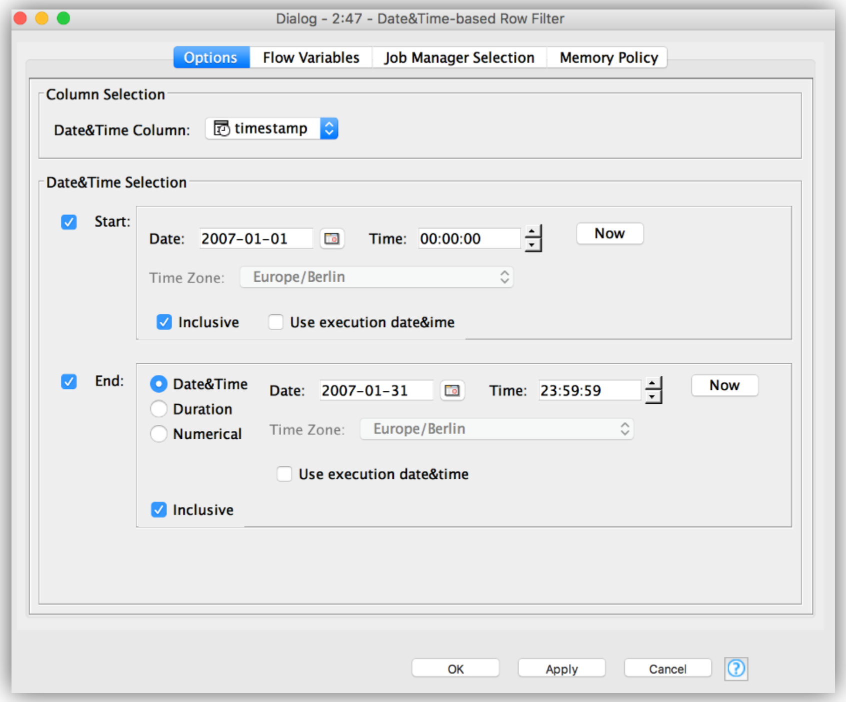

Output table - 2:50 - Create Date&Time Range
File Hilite Navigation View
Table "default" – Rows: 1000
Spec – Column: 1

Row ID	Time
Row0	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row1	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row2	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row3	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row4	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row5	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row6	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row7	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row8	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row9	2018-01-19T14:02:31.155+01:00[Europe/Berlin]
Row10	2018-01-19T14:02:31.155+01:00[Europe/Berlin]

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



Dialog - 2:47 - Date&Time-based Row Filter

Options | Flow Variables | Job Manager Selection | Memory Policy

Column Selection

Date&Time Column: timestamp

Date&Time Selection

[✓] Start: Date: 2007-01-01 Time: 00:00:00 Now

Time Zone: Europe/Berlin

[✓] Inclusive [ ] Use execution date&ime

[✓] End: [✓ Date&Time] Duration Numerical Date: 2007-01-31 Time: 23:59:59 Now

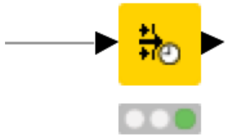
Time Zone: Europe/Berlin

[ ] Use execution date&time

[✓] Inclusive

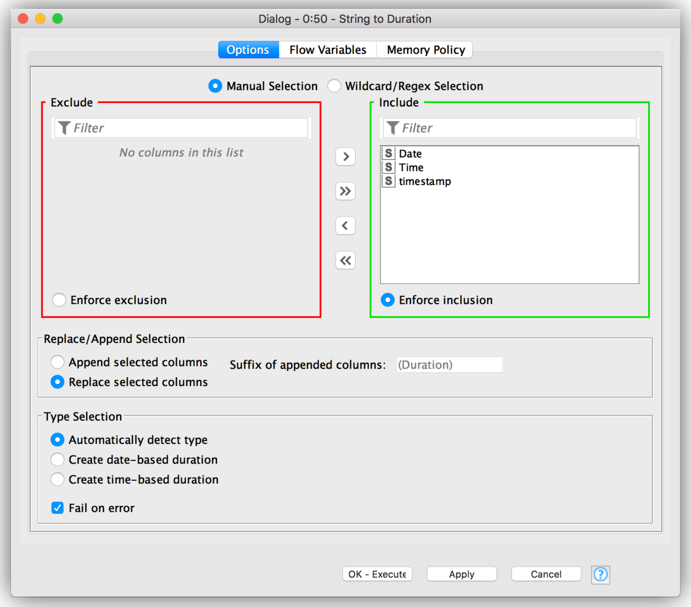
OK Apply Cancel

Date&Time-based Row Filter



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”



Dialog - 0:50 - String to Duration

[✓ Manual Selection] Wildcard/Regex Selection

Exclude: No columns in this list
Enforce exclusion

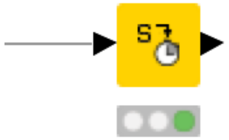
Include: Date, Time, timestamp
[✓ Enforce inclusion]

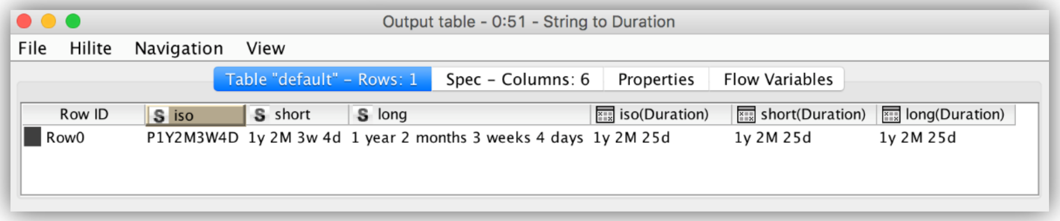
Replace/Append Selection
Append selected columns
[✓ Replace selected columns]
Suffix of appended columns: (Duration)

Type Selection
[✓ Automatically detect type]
Create date-based duration
Create time-based duration

[✓ Fail on error]

String to Duration



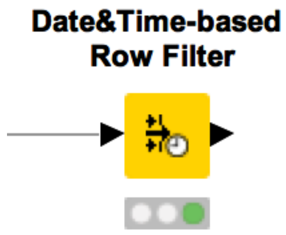


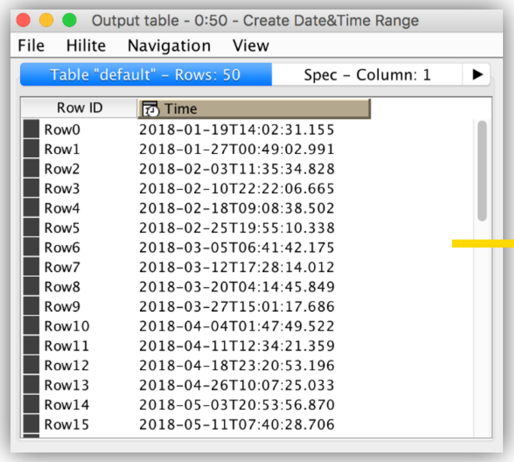
Output table - 0:51 - String to Duration

Row ID	iso	short	long	iso(Duration)	short(Duration)	long(Duration)
Row0	P1Y2M3W4D	1y 2M 3w 4d	1 year 2 months 3 weeks 4 days	1y 2M 25d	1y 2M 25d	1y 2M 25d

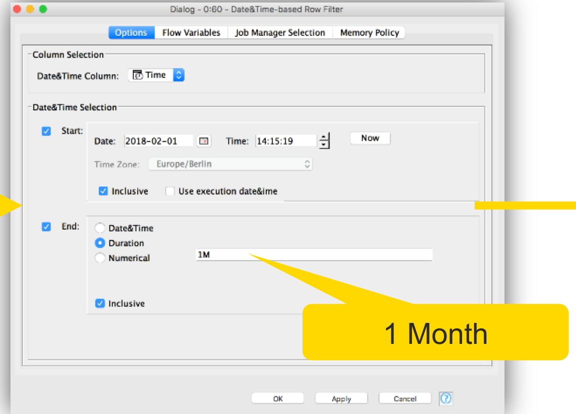
Duration-based Filtering

- 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”




Output table - 0:50 - Create Date&Time Range
File Hilite Navigation View
Table "default" – Rows: 50
Spec – Column: 1

Row ID	Time
Row0	2018-01-19T14:02:31.155
Row1	2018-01-27T00:49:02.991
Row2	2018-02-03T11:35:34.828
Row3	2018-02-10T22:22:06.665
Row4	2018-02-18T09:08:38.502
Row5	2018-02-25T19:55:10.338
Row6	2018-03-05T06:41:42.175
Row7	2018-03-12T17:28:14.012
Row8	2018-03-20T04:14:45.849
Row9	2018-03-27T15:01:17.686
Row10	2018-04-04T01:47:49.522
Row11	2018-04-11T12:34:21.359
Row12	2018-04-18T23:20:53.196
Row13	2018-04-26T10:07:25.033
Row14	2018-05-03T20:53:56.870
Row15	2018-05-11T07:40:28.706



Column Selection

Date&Time Column: Time

Date&Time Selection

[✓] Start: Date: 2018-02-01 Time: 14:15:19 Now

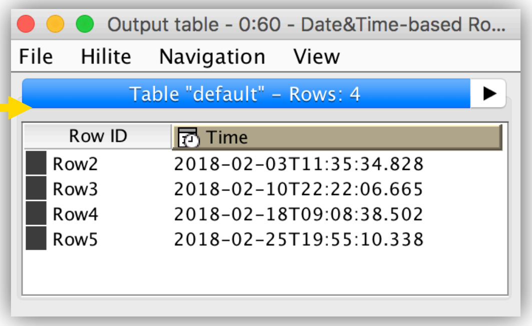
Time Zone: Europe/Berlin

[✓] Inclusive [ ] Use execution date&ime

[✓] End: ( ) Date&Time [● Duration] ( ) Numerical 1M

[✓] Inclusive

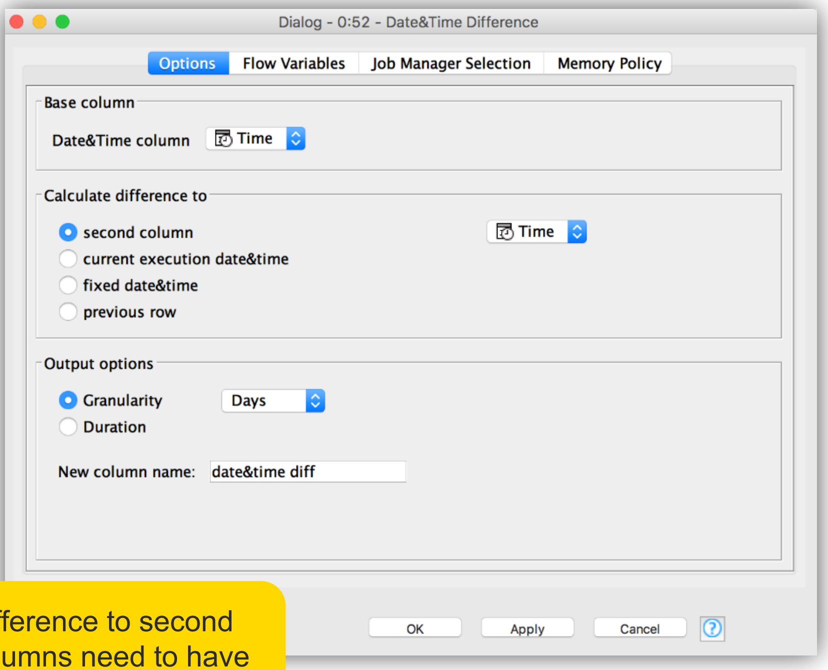
1 Month


Output table - 0:60 - Date&Time-based Ro...
File Hilite Navigation View
Table "default" – Rows: 4

Row ID	Time
Row2	2018-02-03T11:35:34.828
Row3	2018-02-10T22:22:06.665
Row4	2018-02-18T09:08:38.502
Row5	2018-02-25T19:55:10.338

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



Dialog - 0:52 - Date&Time Difference

Base column

Date&Time column: Time

Calculate difference to

- [✓ second column] Time
- current execution date&time
- fixed date&time
- previous row

Output options

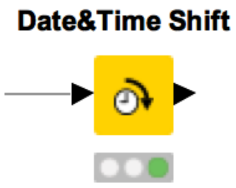
- [✓ Granularity] Days
- Duration

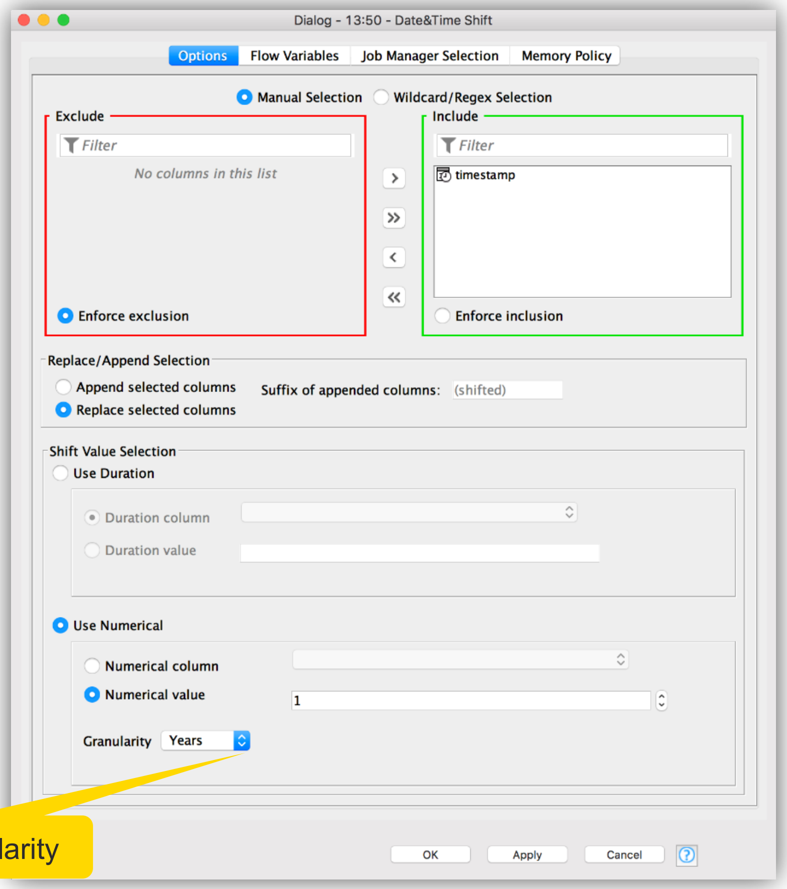
New column name: date&time diff

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





Manual Selection [✓] Wildcard/Regex Selection

Exclude: Filter — No columns in this list — Enforce exclusion [✓]

Include: Filter — timestamp — Enforce inclusion

Replace/Append Selection
- Append selected columns
- [✓ Replace selected columns]

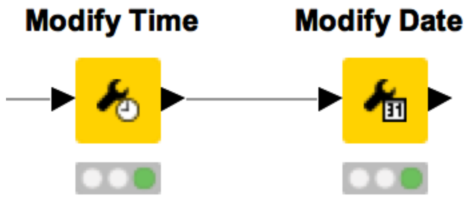
Suffix of appended columns: (shifted)

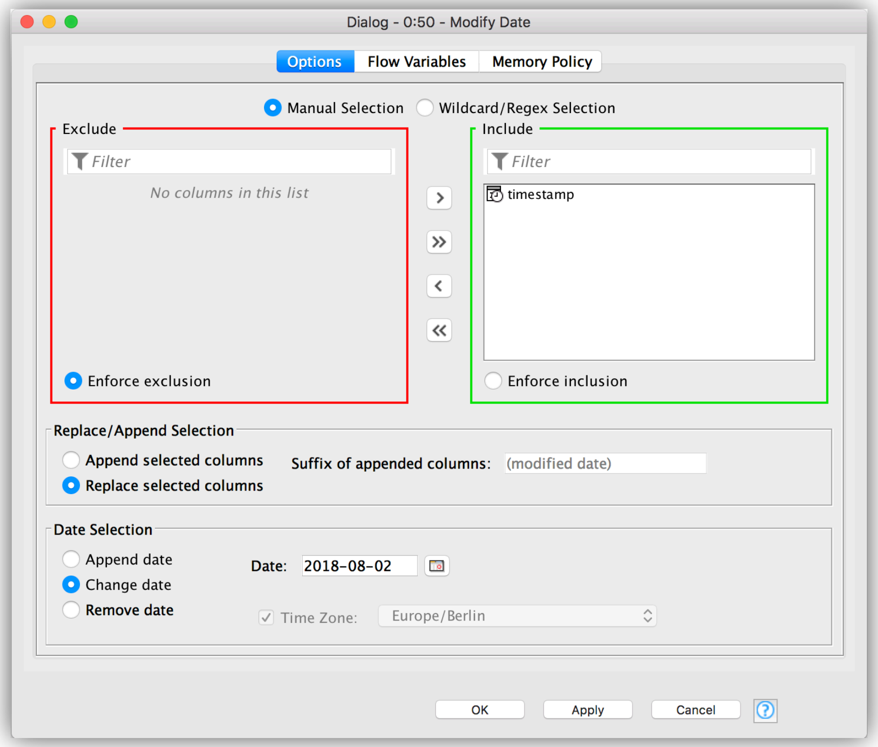
Shift Value Selection
- Use Duration
  - Duration column
  - Duration value
- [✓ Use Numerical]
  - Numerical column
  - [✓ Numerical value]: 1
  - Granularity: Years

Select granularity

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




Dialog - 0:50 - Modify Date

Options | Flow Variables | Memory Policy

[✓ Manual Selection] Wildcard/Regex Selection

Exclude: Filter — No columns in this list — [✓ Enforce exclusion]

Include: Filter — timestamp — Enforce inclusion

Replace/Append Selection
- Append selected columns
- [✓ Replace selected columns]

Suffix of appended columns: (modified date)

Date Selection
- Append date
- [✓ Change date]
- Remove date

Date: 2018-08-02

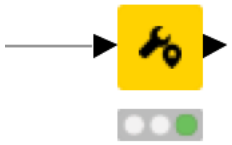
✓ Time Zone: Europe/Berlin

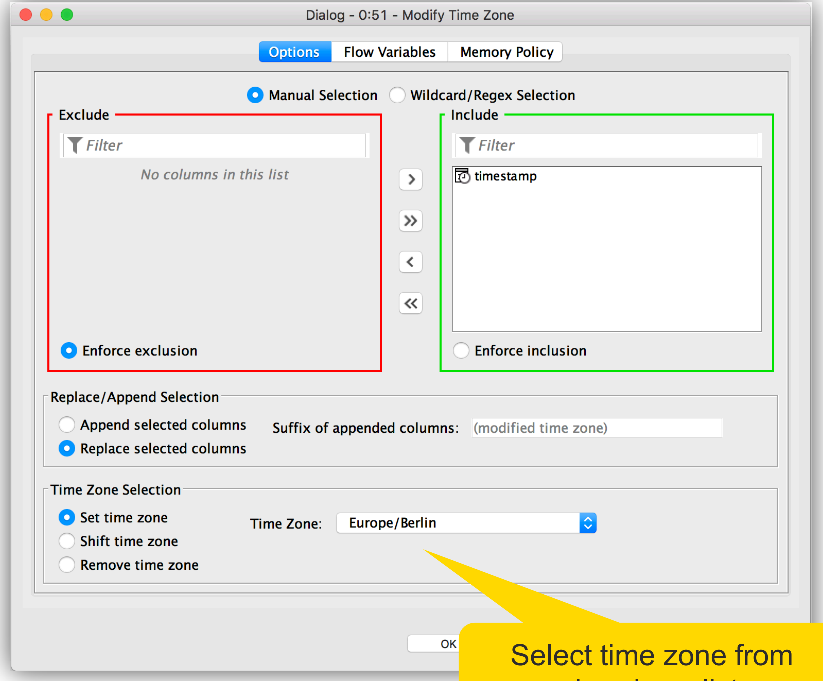
OK | Apply | Cancel

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

Modify Time Zone





Dialog - 0:51 - Modify Time Zone

[✓ Manual Selection] Wildcard/Regex Selection

Exclude: Filter — No columns in this list — [✓ Enforce exclusion]

Include: Filter — timestamp — Enforce inclusion

Replace/Append Selection
Append selected columns
[✓ Replace selected columns]
Suffix of appended columns: (modified time zone)

Time Zone Selection
[✓ Set time zone]
Shift time zone
Remove time zone
Time Zone: Europe/Berlin

Select time zone from dropdown list

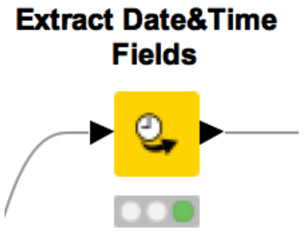
Date and Time Analysis Exercise, Activity I

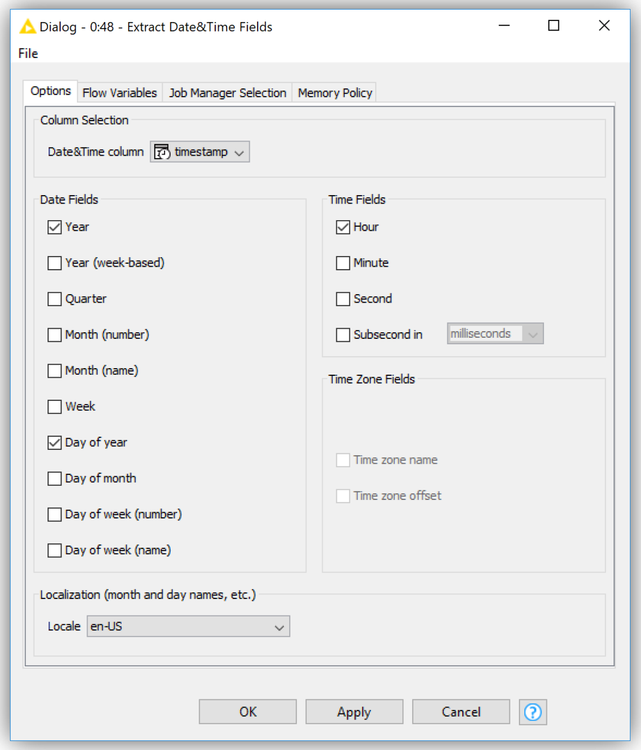
Start with exercise: *Date and Time Analysis, Activity I*

- Read *meter_data.csv* data
- Combine the individual date and time values into a timestamp with the String Manipulation node
- Convert the timestamp from String to Date&Time
- Extract the records for January 2007 with the Date&Time-based Row Filter node

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.)




Dialog - 0:48 - Extract Date&Time Fields

File

Options | Flow Variables | Job Manager Selection | Memory Policy

Column Selection

Date&Time column: timestamp

Date Fields
- [✓] Year
- [ ] Year (week-based)
- [ ] Quarter
- [ ] Month (number)
- [ ] Month (name)
- [ ] Week
- [✓] Day of year
- [ ] Day of month
- [ ] Day of week (number)
- [ ] Day of week (name)

Time Fields
- [✓] Hour
- [ ] Minute
- [ ] Second
- [ ] Subsecond in milliseconds

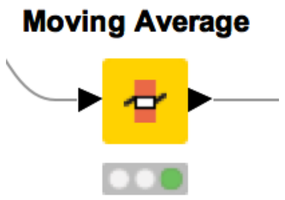
Time Zone Fields
- [ ] Time zone name
- [ ] Time zone offset

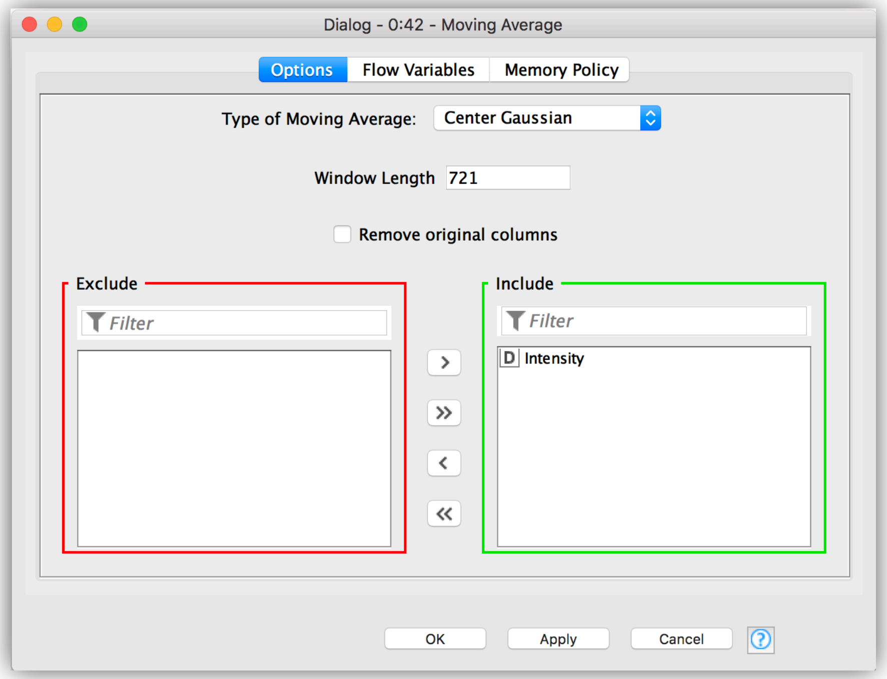
Localization (month and day names, etc.)

Locale: en-US

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





Type of Moving Average: Center Gaussian

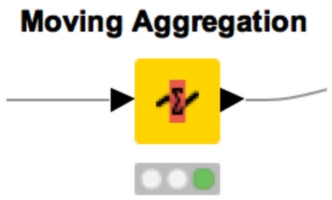
Window Length 721

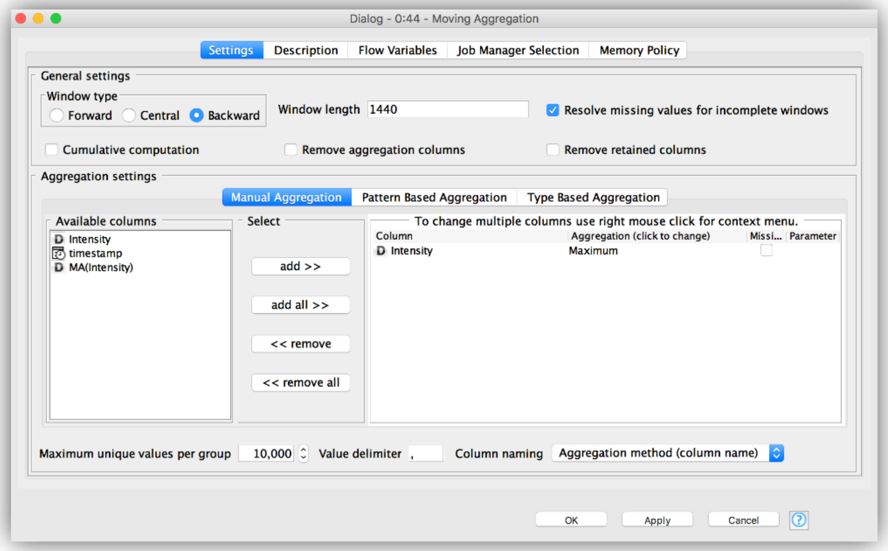
☐ Remove original columns

Include: D Intensity

Moving Aggregation

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




Dialog - 0:44 - Moving Aggregation

Settings | Description | Flow Variables | Job Manager Selection | Memory Policy

General settings

Window type: Forward [✓ Backward] Central — Window length 1440 — [✓ 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: Intensity, timestamp, MA(Intensity)

Select: add >>, add all >>, << remove, << remove all

To change multiple columns use right mouse click for context menu.

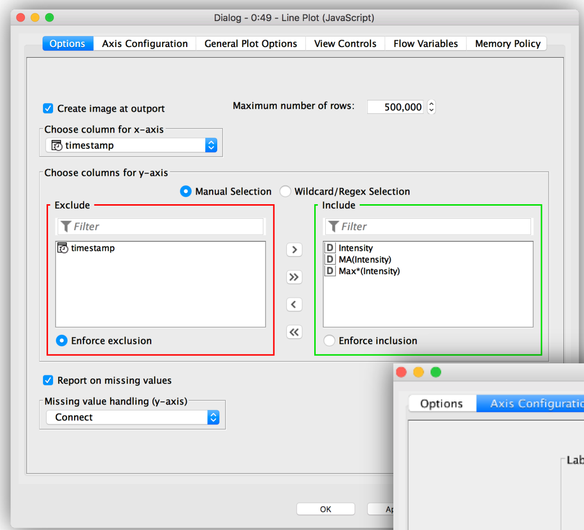
Column	Aggregation (click to change)	Missi...	Parameter
Intensity	Maximum		

Maximum unique values per group 10,000 — Value delimiter , — Column naming Aggregation method (column name)

OK | Apply | Cancel

Line Plot

- Line plot with support for Date columns



Dialog - 0:49 - Line Plot (JavaScript)

Options | Axis Configuration | General Plot Options | View Controls | Flow Variables | Memory Policy

[✓] Create image at outport   Maximum number of rows: 500,000

Choose column for x-axis: timestamp

Choose columns for y-axis

[✓ Manual Selection] Wildcard/Regex Selection

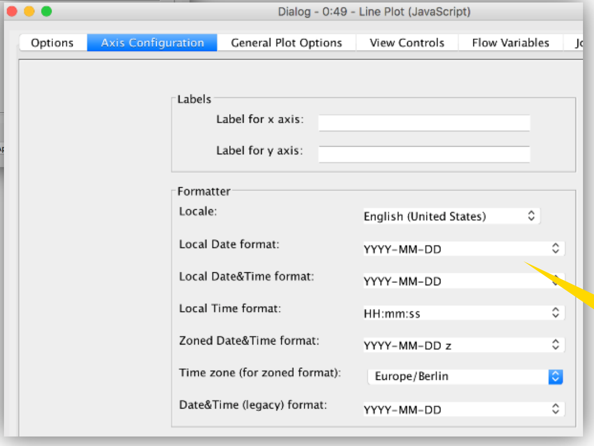
Exclude: timestamp

Include: Intensity, MA(Intensity), Max*(Intensity)

[✓ Enforce exclusion] Enforce inclusion

[✓] Report on missing values

Missing value handling (y-axis): Connect



Dialog - 0:49 - Line Plot (JavaScript)

Options | Axis Configuration | General Plot Options | View Controls | Flow Variables

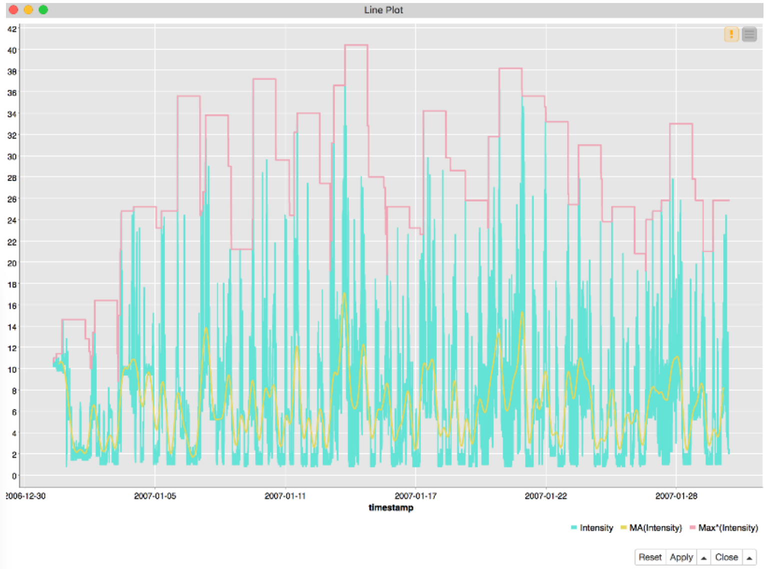
Labels

Label for x axis:

Label for y axis:

Formatter

Locale:	English (United States)
Local Date format:	YYYY-MM-DD
Local Date&Time format:	YYYY-MM-DD
Local Time format:	HH:mm:ss
Zoned Date&Time format:	YYYY-MM-DD z
Time zone (for zoned format):	Europe/Berlin
Date&Time (legacy) format:	YYYY-MM-DD



Line Plot

Format axis

Date and Time Analysis Exercise, Activity II

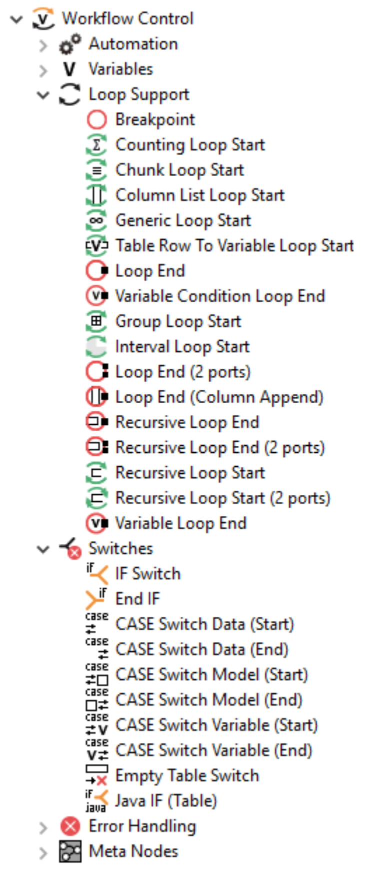
Start with exercise: *Date and Time Analysis, Activity II*

- Read *sampled_meter_data.table* data
- Extract Year, Day of Year, and Hour values from the timestamp into separate columns
- Calculate the average timestamp and average intensity by year, day, and hour
- Start a new workflow branch and calculate gaussian centered moving average of the intensity
- Calculate the maximum of the intensity column for the preceding day (1440 previous records)
- Plot the original, average, and maximum intensity in a line plot

Workflow Control
Loops, Switches, Try-Catch

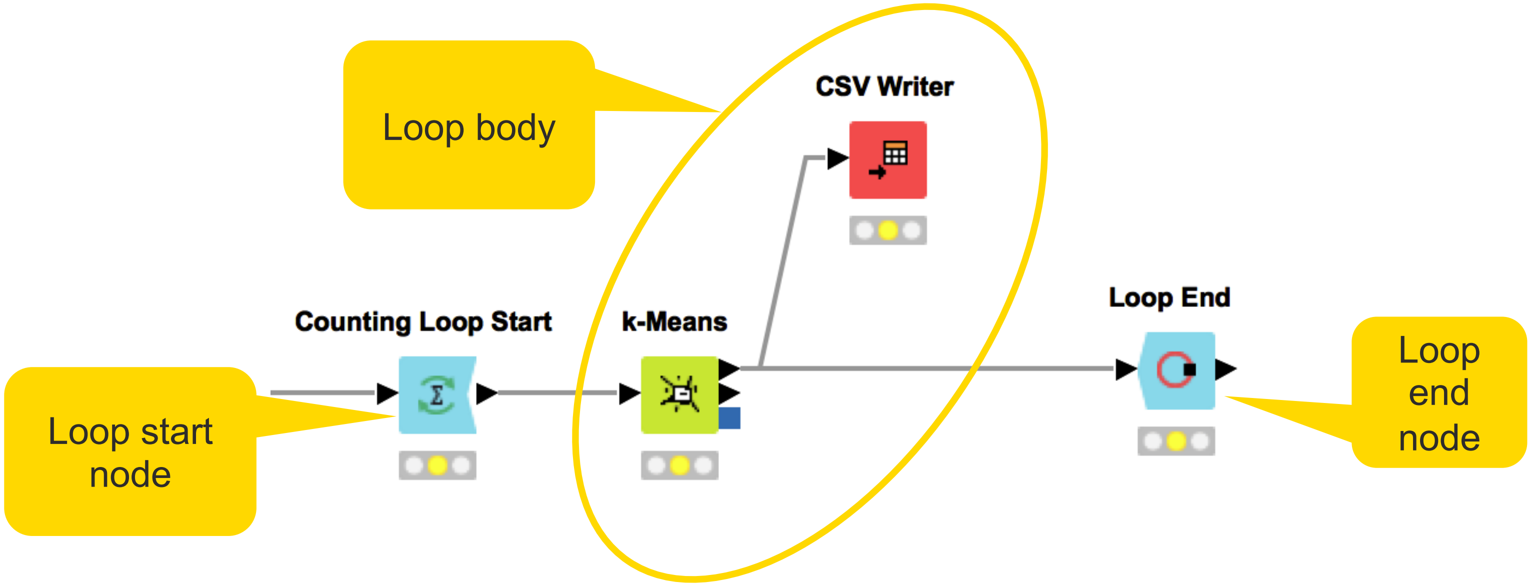
Workflow Control Structures

- Loops
  - Iterate over a workflow snippet with variable inputs.
- Switches
  - Direct the path of a workflow by selectively executing one or more workflow branches.
- Try-Catch
  - Handle workflow branches that may fail in execution and you don‘t know before execution


Workflow Control
Automation
Variables
Loop Support
Breakpoint
Counting Loop Start
Chunk Loop Start
Column List Loop Start
Generic Loop Start
Table Row To Variable Loop Start
Loop End
Variable Condition Loop End
Group Loop Start
Interval Loop Start
Loop End (2 ports)
Loop End (Column Append)
Recursive Loop End
Recursive Loop End (2 ports)
Recursive Loop Start
Recursive Loop Start (2 ports)
Variable Loop End
Switches
IF Switch
End IF
CASE Switch Data (Start)
CASE Switch Data (End)
CASE Switch Model (Start)
CASE Switch Model (End)
CASE Switch Variable (Start)
CASE Switch Variable (End)
Empty Table Switch
Java IF (Table)
Error Handling
Meta Nodes

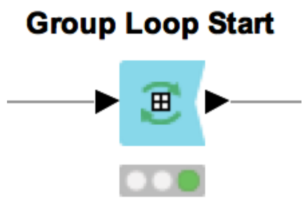
The Loop Block

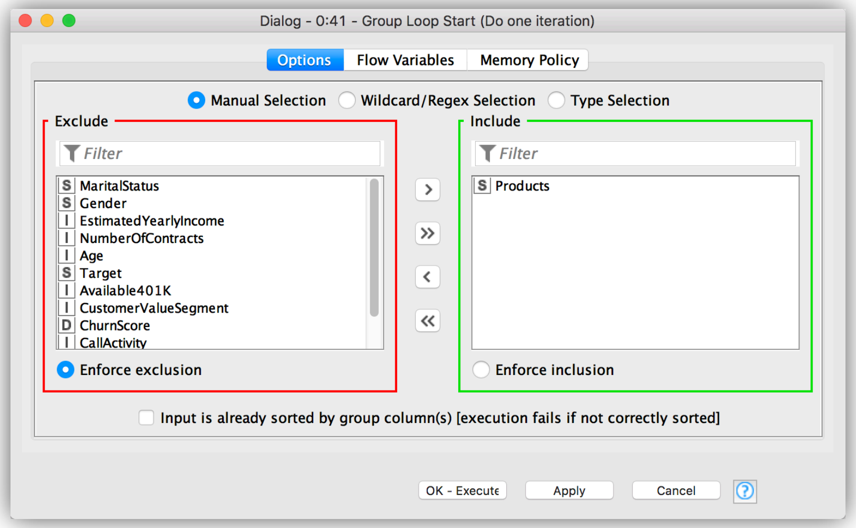
- A loop block is defined by appropriate loop start and loop end nodes.
- Loop body = Nodes in between (including side branches).



Group Loop Start

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




Dialog - 0:41 - Group Loop Start (Do one iteration)

Options | Flow Variables | Memory Policy

[✓ Manual Selection] Wildcard/Regex Selection Type Selection

Exclude
- S MaritalStatus
- S Gender
- I EstimatedYearlyIncome
- I NumberOfContracts
- I Age
- S Target
- I Available401K
- I CustomerValueSegment
- D ChurnScore
- I CallActivity

[✓ Enforce exclusion]

Include
- S Products

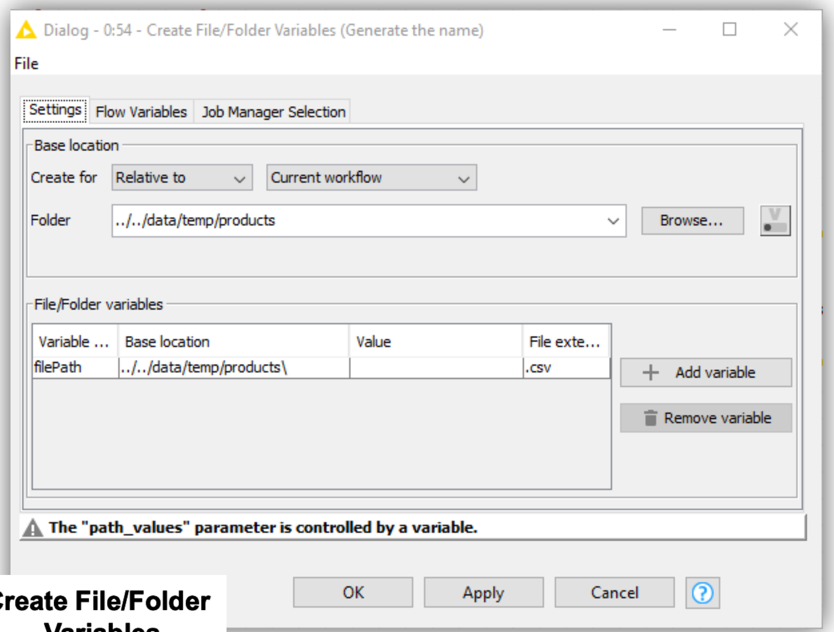
Enforce inclusion

☐ Input is already sorted by group column(s) [execution fails if not correctly sorted]

OK - Execute | Apply | Cancel

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.


Dialog - 0:54 - Create File/Folder Variables (Generate the name)

File

Base location

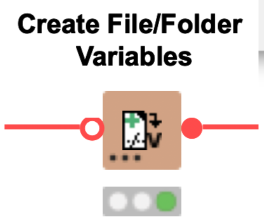
Create for: Relative to — Current workflow

Folder: ../../data/temp/products

File/Folder variables

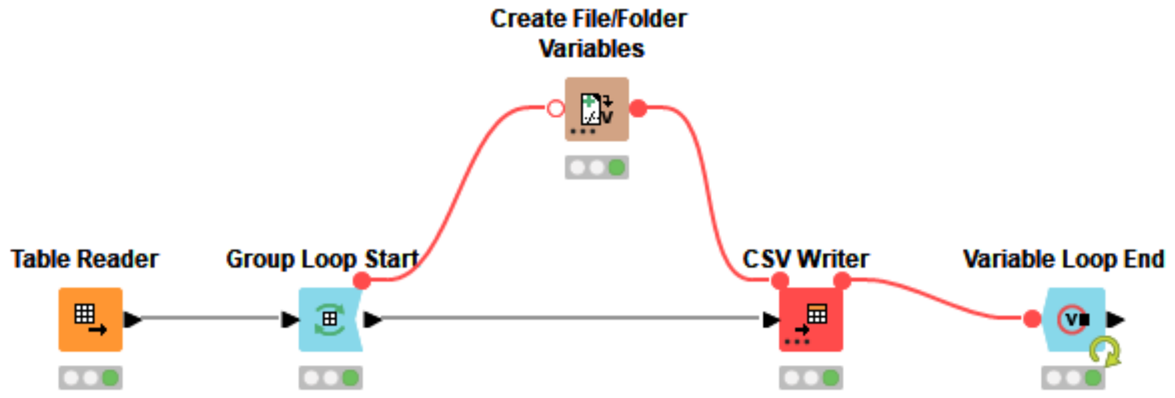
Variable ...	Base location	Value	File exte...
filePath	../../data/temp/products\		.csv

The "path_values" parameter is controlled by a variable.



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 path variable
- Write grouped data to tables with new file name



Example: Writing Multiple Excel Sheets

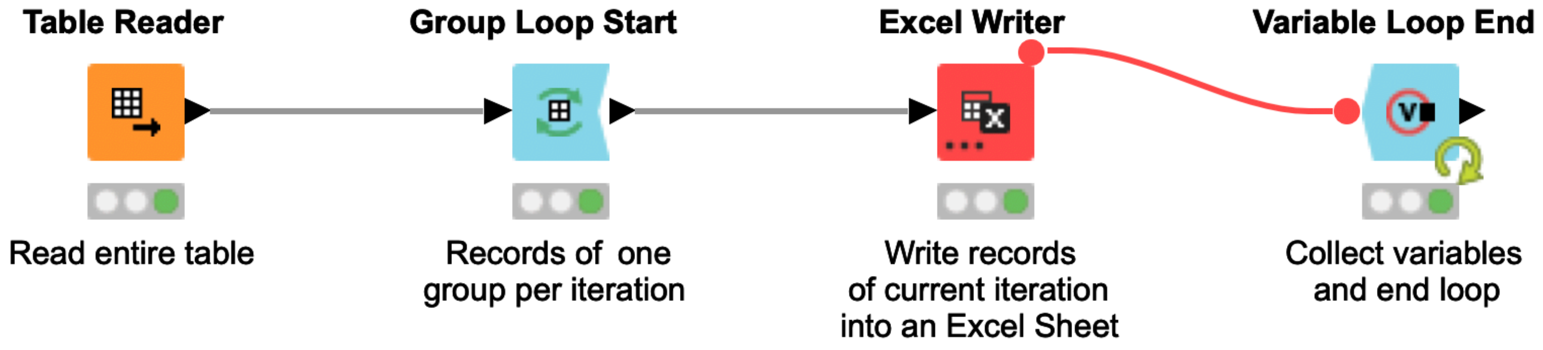


Table Reader
Read entire table
Group Loop Start
Records of one group per iteration
Excel Writer
Write records of current iteration into an Excel Sheet
Variable Loop End
Collect variables and end loop

Workflow Control Exercise, Activity I

Goal: Build a loop that will create an Excel file with separate Excel sheets for the records of different products.

- Read the table CurrentDetailData.table (Table Reader node)
- Start a loop that handles the records for the different products in separate iterations (Group Loop Start node)
- For each product write one Excel sheet into a single Excel file (Excel Writer node)
- Close and execute the loop (Variable Loop End node)

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

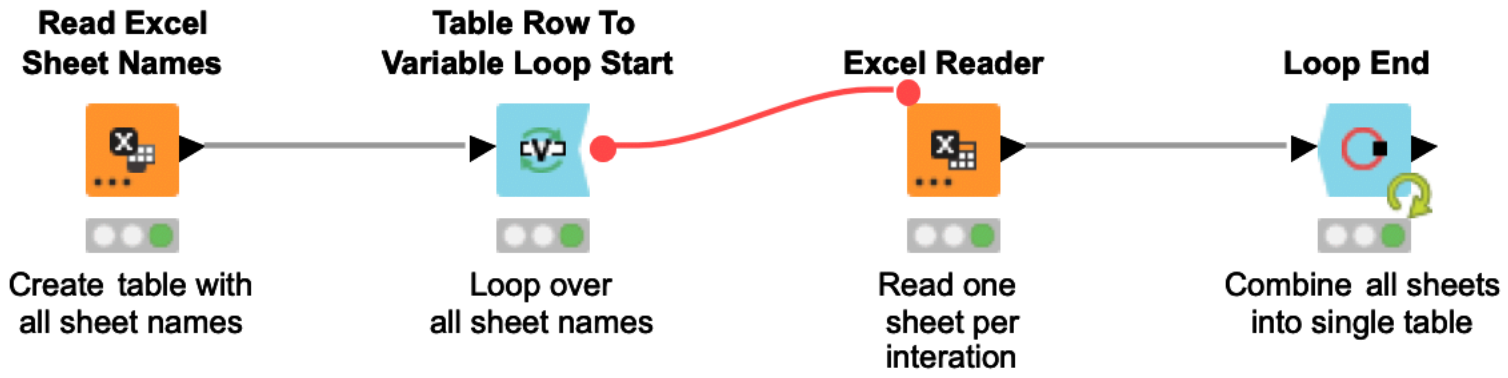
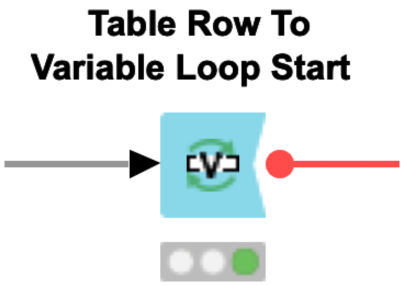
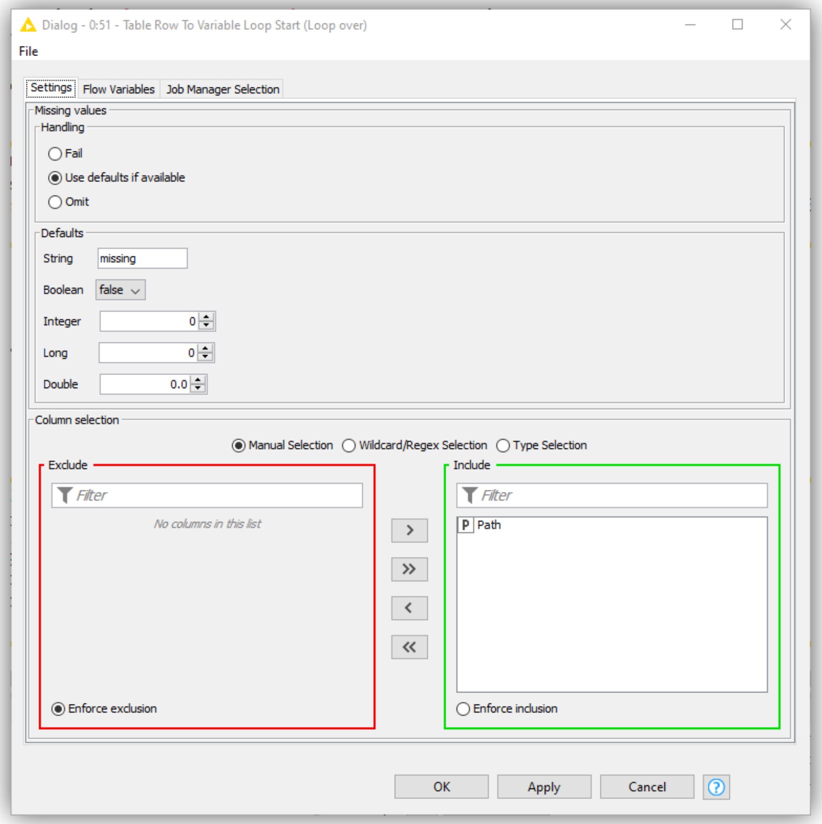

Read Excel Sheet Names
Table Row To Variable Loop Start
Excel Reader
Loop End
Create table with all sheet names
Loop over all sheet names
Read one sheet per interation
Combine all sheets into single table

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




Dialog - 0:51 - Table Row To Variable Loop Start (Loop over)

File

Settings | Flow Variables | Job Manager Selection

Missing values

Handling

( ) Fail
(●) Use defaults if available
( ) Omit

Defaults

String	missing
Boolean	false
Integer	0
Long	0
Double	0.0

Column selection

(●) Manual Selection ( ) Wildcard/Regex Selection ( ) Type Selection

Exclude: Filter — No columns in this list — (●) Enforce exclusion

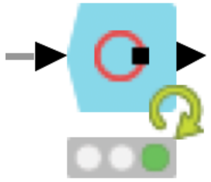
Include: Filter — P Path — ( ) Enforce inclusion

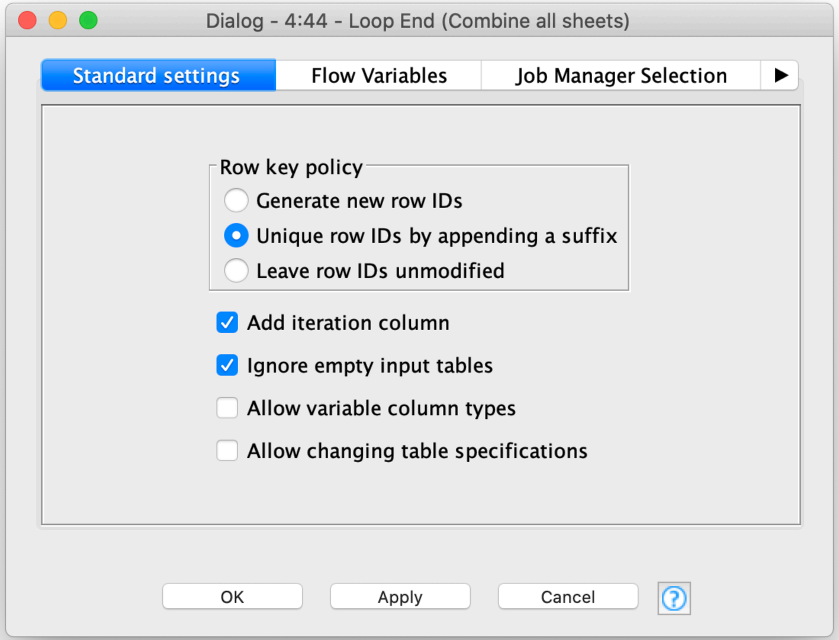
OK | Apply | Cancel

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

Loop End





Dialog - 4:44 - Loop End (Combine all sheets)

Row key policy
- Generate new row IDs
- [✓ Unique row IDs by appending a suffix]
- Leave row IDs unmodified

- [✓] Add iteration column
- [✓] Ignore empty input tables
- [ ] Allow variable column types
- [ ] Allow changing table specifications

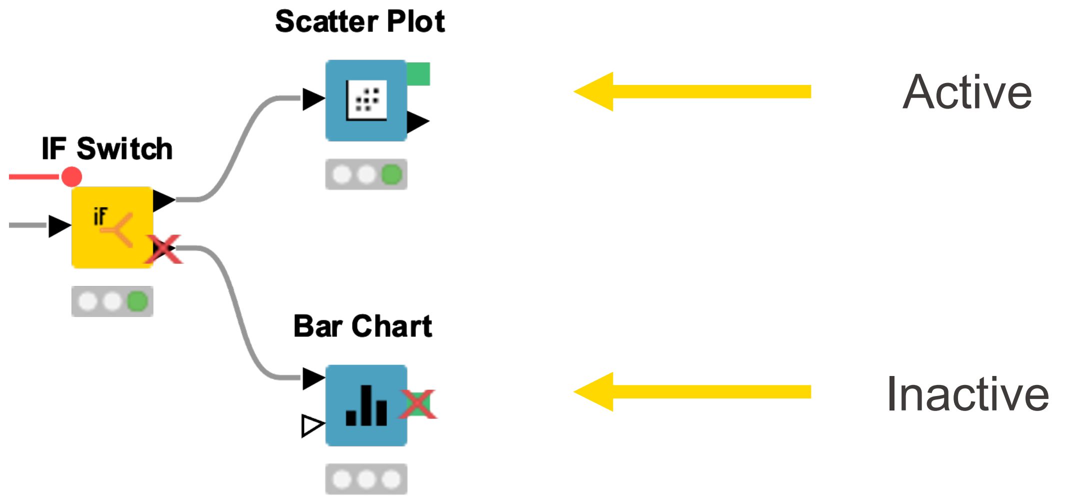
Workflow Control Exercise, Activity II

Goal: Create a loop that reads and concatenates all the sheets in an Excel file.

- Create a table that contains all sheet names of the Excel file created in Activity I (Read Excel Sheet Names node)
- Start a loop that iterates over the sheet names (Table Row to Variable Loop Start node)
- Read the Excel sheet with the sheet name in the current iteration (Excel Reader node)
- Close the loop and concatenate the tables from the different iterations (Loop End node)

Switches

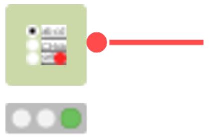
- A switch allows you to selectively activate branches of a workflow
- Inactive branches are marked with a red x on their output ports. Inactive nodes propagate down stream.

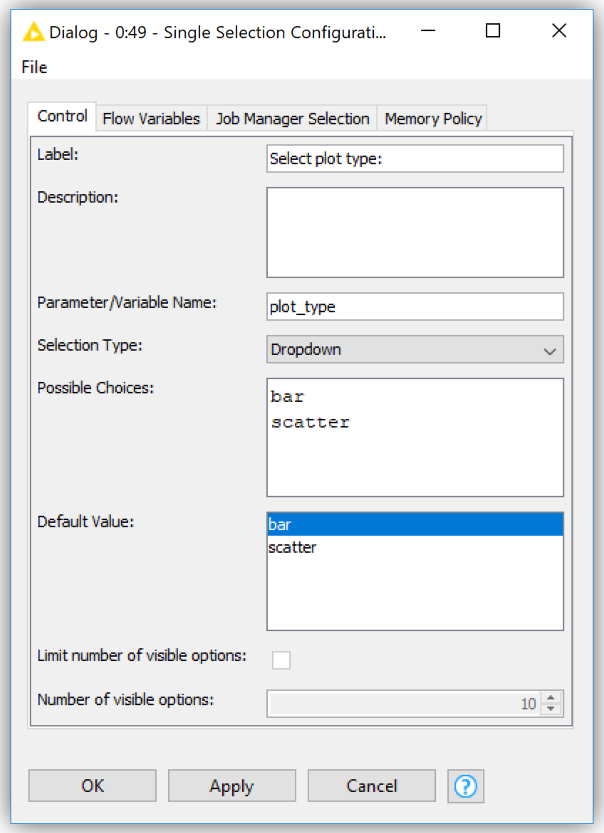


Single Selection Configuration

- Configuration: Select single value from list of Strings
- Returns selection as string type Flow Variable
- Choose between different layout options (dropdown, radio buttons...)

Single Selection Configuration

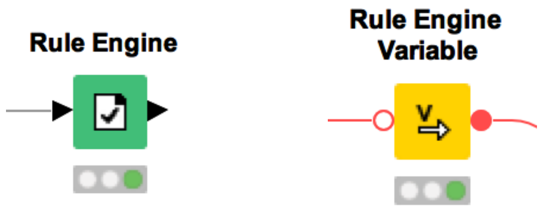


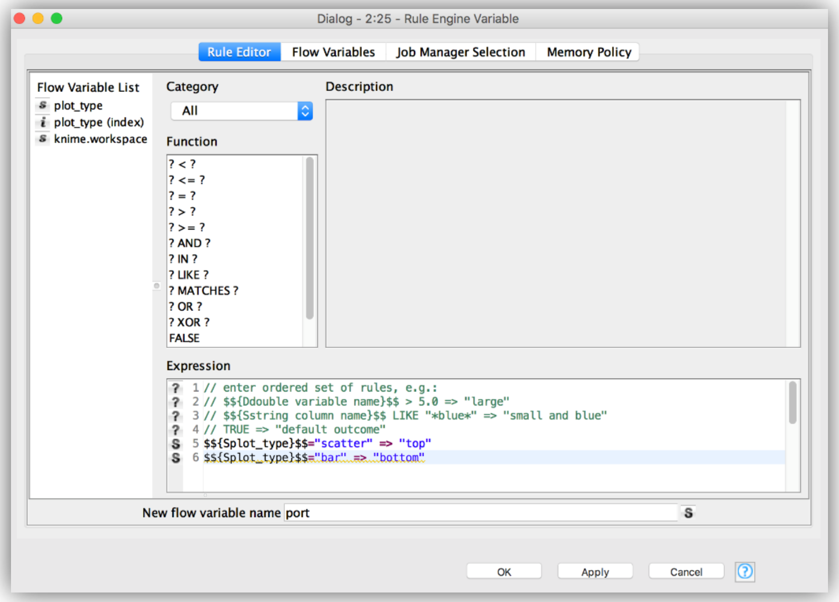


Label:	Select plot type:
Description:	
Parameter/Variable Name:	plot_type
Selection Type:	Dropdown
Possible Choices:	bar scatter
Default Value:	bar (selected) scatter
Limit number of visible options:	☐
Number of visible options:	10

Rule Engine/Rule Engine Variable

- Define custom logic using simple rules.
- Rules like: <Antecedent> => <Consequence>, e.g 1=1 => “true”
- May be used in Flow Variables or tables
- Easiest way to encode logic for switches





Flow Variable List
- plot_type
- plot_type (index)
- knime.workspace

Category: All

Function
```
? < ?
? <= ?
? = ?
? > ?
? >= ?
? AND ?
? IN ?
? LIKE ?
? MATCHES ?
? OR ?
? XOR ?
FALSE
```

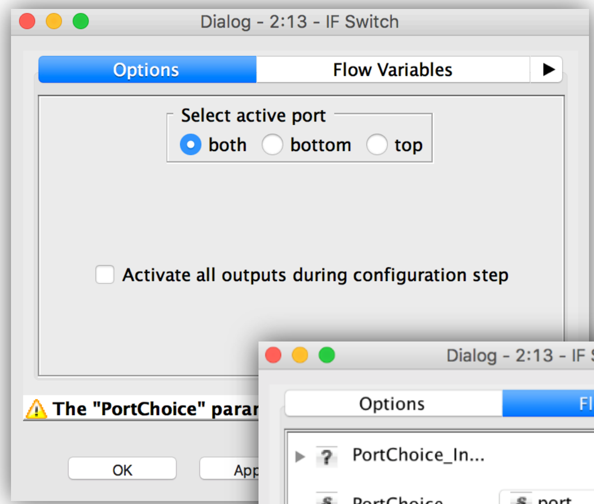
Description

Expression
```
// enter ordered set of rules, e.g.:
// $${Ddouble variable name}$$ > 5.0 => "large"
// $${Sstring column name}$$ LIKE "*blue*" => "small and blue"
// TRUE => "default outcome"
$${Splot_type}$$="scatter" => "top"
$${Splot_type}$$="bar" => "bottom"
```

New flow variable name: port

If Switch

- Controls which branches of your workflow are active programmatically
- Controlled with a Flow Variable, setting the value to the literal Strings: “top”, “bottom”, “both”
- May be used in Flow Variables or tables (different nodes)

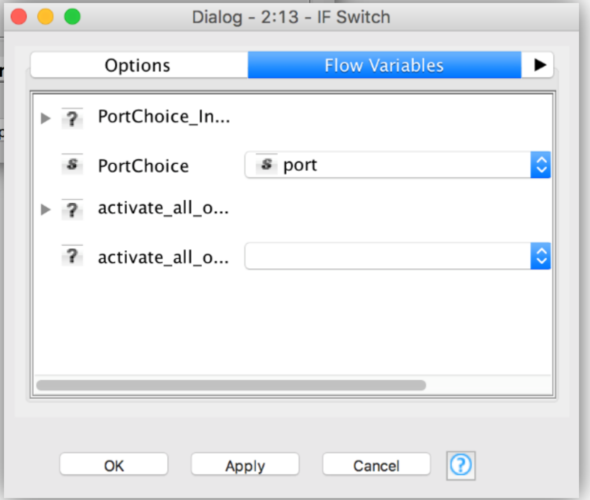


Dialog - 2:13 - IF Switch

Select active port: [✓ both] bottom top

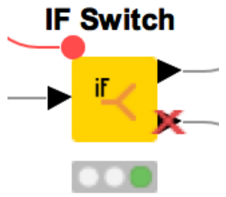
☐ Activate all outputs during configuration step

⚠ The "PortChoice" parar



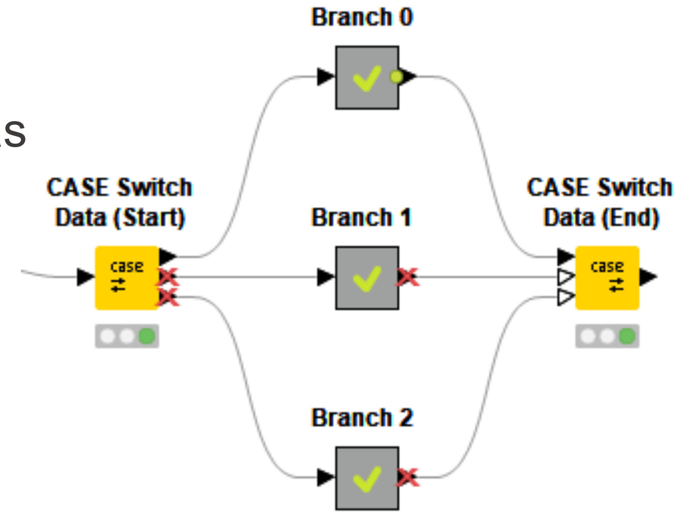
Dialog - 2:13 - IF Switch

PortChoice_In...	
PortChoice	port
activate_all_o...	
activate_all_o...	



Case Switch Data

- Similar to If-Switch: Takes data from single input port and passes it to the active output port
- Nodes connected to inactive branches are not executed
- Configure via node dialog, or pass port index as Flow Variable
  - 0, 1, 2 for top, middle, and bottom port
- Case switches also available for Flow Variable and model ports



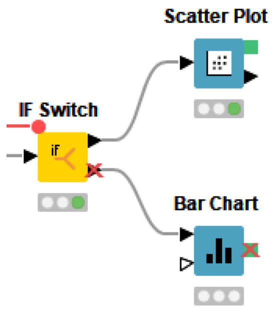
The difference between Loops and Switches

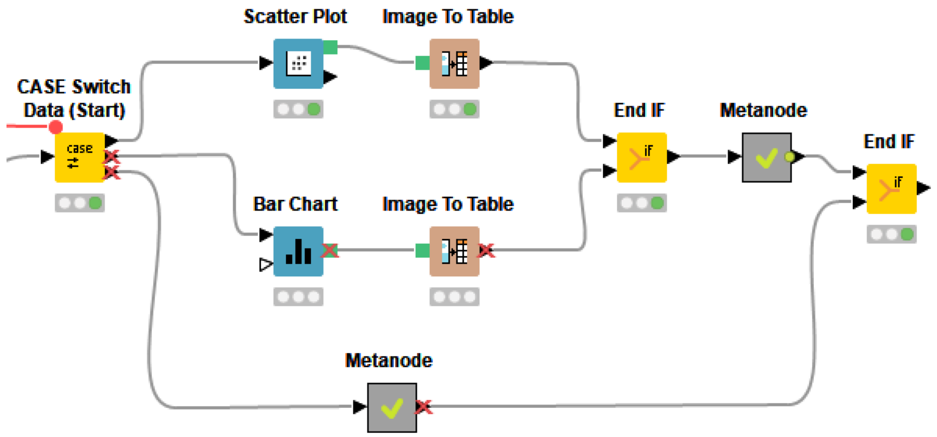
Loops

- The Loop Start is connected to the Loop End node, they form a pair.
- A loop iterates over a workflow part.

Switches

- A Switch Start can be used without a corresponding Switch End. They can also be combined.





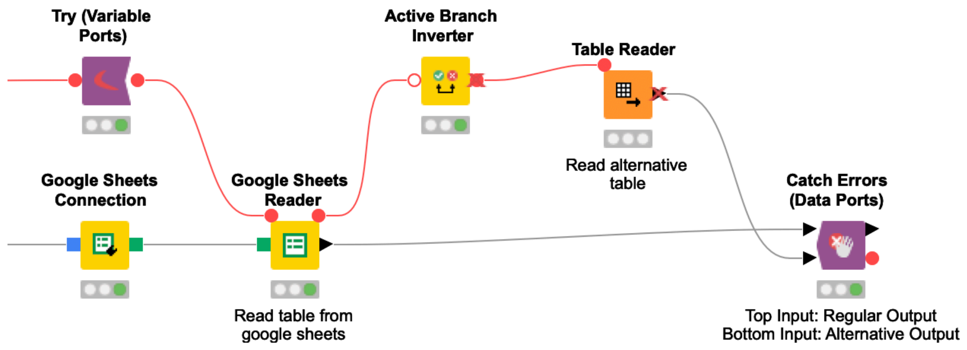
Workflow Control Exercise, Activity III

- Extend the workflow below with a switch that only creates one type of visualization
  - Create a Single Selection Configuration node with the possible values "scatter" and "bar"
  - Use the CASE Switch Data (Start) that activates the top or the middle branch depending on the selection scatter/bar (Use the "...(index)" flow variable to define the active port)
  - Combine the outputs of the two branches with the CASE Switch Data (End) node

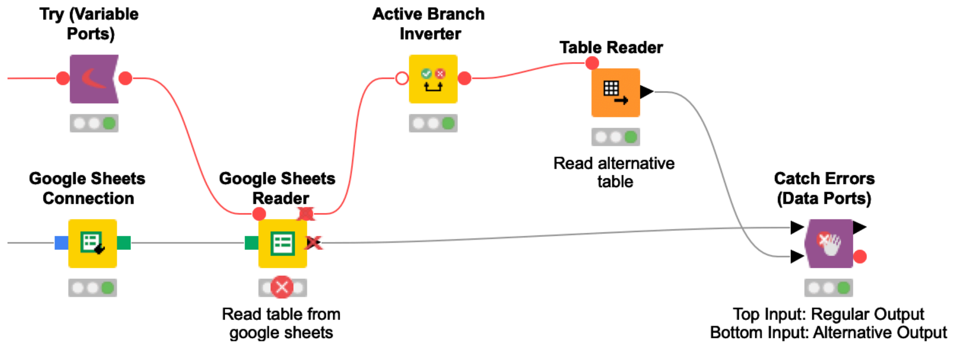
Try-Catch

- A way to catch errors in workflows
- Useful when it is hard to know if a node will execute (for example,
  when reading from a Google Sheet)
- KNIME tries to execute the nodes, but if it fails will fall back to
  an alternative branch

Regular Execution



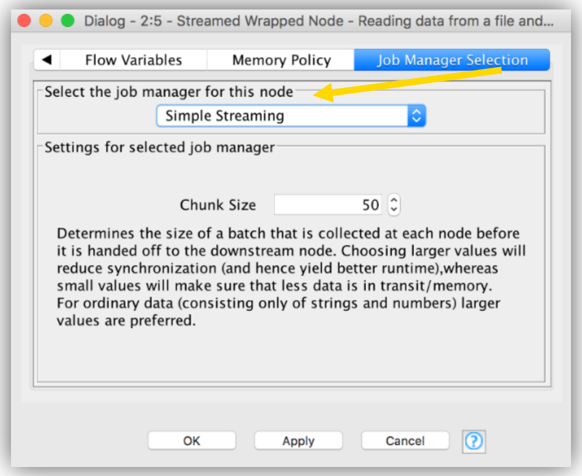
Alternative Execution



Streaming

- Standard execution: Node by node. Node processes all data, finishes, then passes data to next node, etc.
- Streaming: Nodes executed concurrently, each nodes passes data to the next as soon as it is available, i.e. before node is fully executed
  - Faster execution, esp. for reading/preprocessing data
- Install KNIME Streaming Execution (Beta) extension
- Create Component -> Configure -> Job Manager Selection -> Simple Streaming
  - Not available for all nodes (show in node repository)
  - Can only execute entire metanode, not individual nodes
  - Intermediate results not available since nothing is cached

Streaming



Dialog - 2:5 - Streamed Wrapped Node - Reading data from a file and...

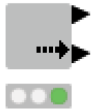
Select the job manager for this node: Simple Streaming

Settings for selected job manager

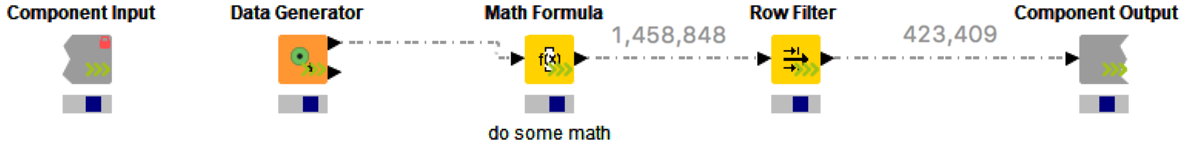
Chunk Size: 50

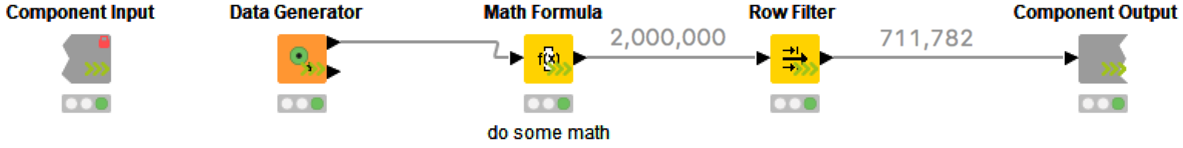
Determines the size of a batch that is collected at each node before it is handed off to the downstream node. Choosing larger values will reduce synchronization (and hence yield better runtime),whereas small values will make sure that less data is in transit/memory. For ordinary data (consisting only of strings and numbers) larger values are preferred.

Streamed Component - Reading data from a file and process



Sub Workflow as Component.
To open it:
- right click > Component > Open
- Ctrl + Double Click





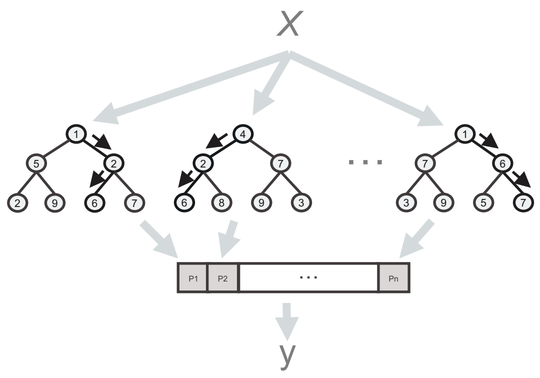
Advanced Data Mining

Overview

- Ensemble models
  - Random Forest / Tree Ensembles
  - Gradient Boosted Trees
- Parameter optimization
- Cross validation
- H2O and Keras integration in KNIME

KNIME's Tree Ensemble Models

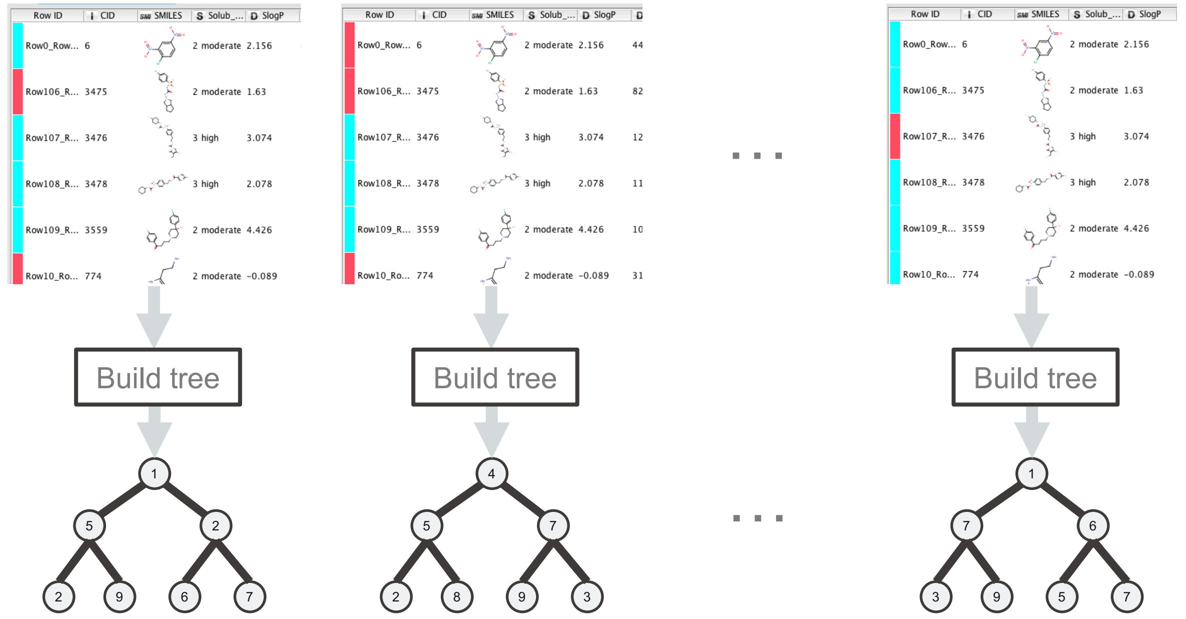
- The general idea is to take advantage of the "wisdom of the crowd"
- Ensemble models: Combining predictions from a large number of weak predictors, e.g. decision trees
- Leads to a more accurate and robust model
- This is called "bagging"



Typically: for classification the individual models vote and the majority wins; for regression, the individual predictions are averaged

How Does Bagging Work?

- Pick a different random subset of the training data for each model in the ensemble (bag)



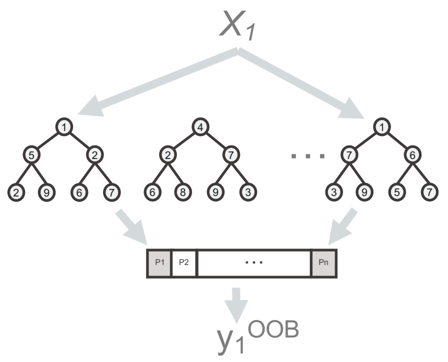
	Row ID	CID	SMILES	Solub_...	SlogP
🟦	Row0_Row...	6		2 moderate	2.156
🟥	Row106_R...	3475		2 moderate	1.63
🟦	Row107_R...	3476		3 high	3.074
🟦	Row108_R...	3478		3 high	2.078
🟦	Row109_R...	3559		2 moderate	4.426
🟥	Row10_Ro...	774		2 moderate	-0.089

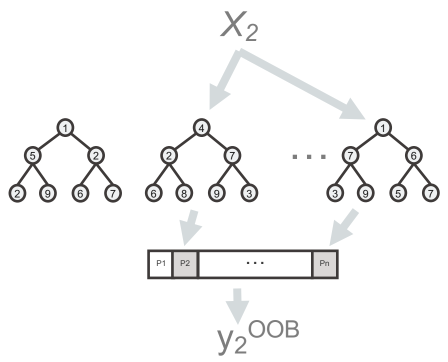
	Row ID	CID	SMILES	Solub_...	SlogP	D
🟥	Row0_Row...	6		2 moderate	2.156	44
🟥	Row106_R...	3475		2 moderate	1.63	82
🟦	Row107_R...	3476		3 high	3.074	12
🟦	Row108_R...	3478		3 high	2.078	11
🟦	Row109_R...	3559		2 moderate	4.426	10
🟥	Row10_Ro...	774		2 moderate	-0.089	31

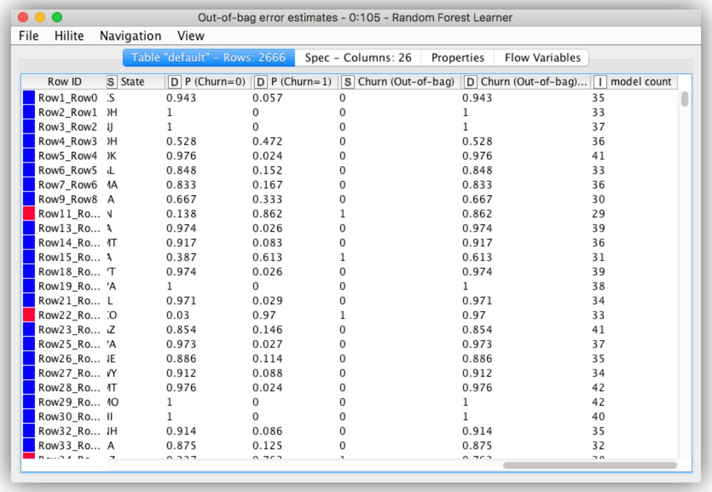
	Row ID	CID	SMILES	Solub_...	SlogP
🟦	Row0_Row...	6		2 moderate	2.156
🟦	Row106_R...	3475		2 moderate	1.63
🟥	Row107_R...	3476		3 high	3.074
🟦	Row108_R...	3478		3 high	2.078
🟦	Row109_R...	3559		2 moderate	4.426
🟦	Row10_Ro...	774		2 moderate	-0.089

An Extra Benefit of Bagging: Out of Bag Estimation

- Allows testing the model using the training data: when validating, each model should only vote on data points that were not used to train it





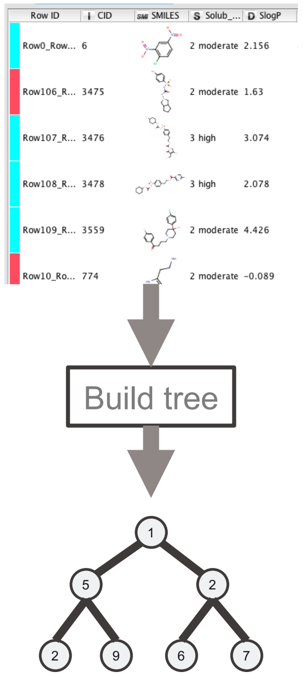


Out-of-bag error estimates - 0:105 - Random Forest Learner
File
Hilite
Navigation
View
Table "default" – Rows: 2666
Spec – Columns: 26
Properties
Flow Variables

	Row ID	S State	D P (Churn=0)	D P (Churn=1)	S Churn (Out-of-bag)	D Churn (Out-of-bag)...	I model count
🟦	Row1_Row0	:S	0.943	0.057	0	0.943	35
🟦	Row2_Row1	)H	1	0	0	1	33
🟦	Row3_Row2	IJ	1	0	0	1	37
🟦	Row4_Row3	)H	0.528	0.472	0	0.528	36
🟦	Row5_Row4	)K	0.976	0.024	0	0.976	41
🟦	Row6_Row5	\L	0.848	0.152	0	0.848	33
🟦	Row7_Row6	IA	0.833	0.167	0	0.833	36
🟦	Row9_Row8	A	0.667	0.333	0	0.667	30
🟥	Row11_Ro...	|	0.138	0.862	1	0.862	29
🟦	Row13_Ro...	\	0.974	0.026	0	0.974	39
🟦	Row14_Ro...	IT	0.917	0.083	0	0.917	36
🟦	Row15_Ro...	\	0.387	0.613	1	0.613	31
🟦	Row18_Ro...	'T	0.974	0.026	0	0.974	39
🟦	Row19_Ro...	'A	1	0	0	1	38
🟦	Row21_Ro...	L	0.971	0.029	0	0.971	34
🟥	Row22_Ro...	:O	0.03	0.97	1	0.97	33
🟦	Row23_Ro...	\Z	0.854	0.146	0	0.854	41
🟦	Row25_Ro...	'A	0.973	0.027	0	0.973	37
🟦	Row26_Ro...	IE	0.886	0.114	0	0.886	35
🟦	Row27_Ro...	VY	0.912	0.088	0	0.912	34
🟦	Row28_Ro...	IT	0.976	0.024	0	0.976	42
🟦	Row29_Ro...	IO	1	0	0	1	42
🟦	Row30_Ro...	II	1	0	0	1	40
🟦	Row32_Ro...	IH	0.914	0.086	0	0.914	35
🟦	Row33_Ro...	A	0.875	0.125	0	0.875	32
🟥	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Random Forest

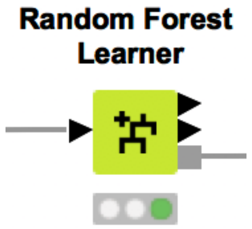
- Train a bag of decision trees
- For each tree / model a training set is generated by sampling uniformly with replacement from the standard training set
- An extra element of randomization is used when building the trees : **each node** in the decision tree only “sees” **a subset of the input columns**, typically $\sqrt{N}$
- Random forests tend to be very robust w.r.t. overfitting (though the individual trees are almost certainly overfit)
- Extra benefit: training tends to be much faster

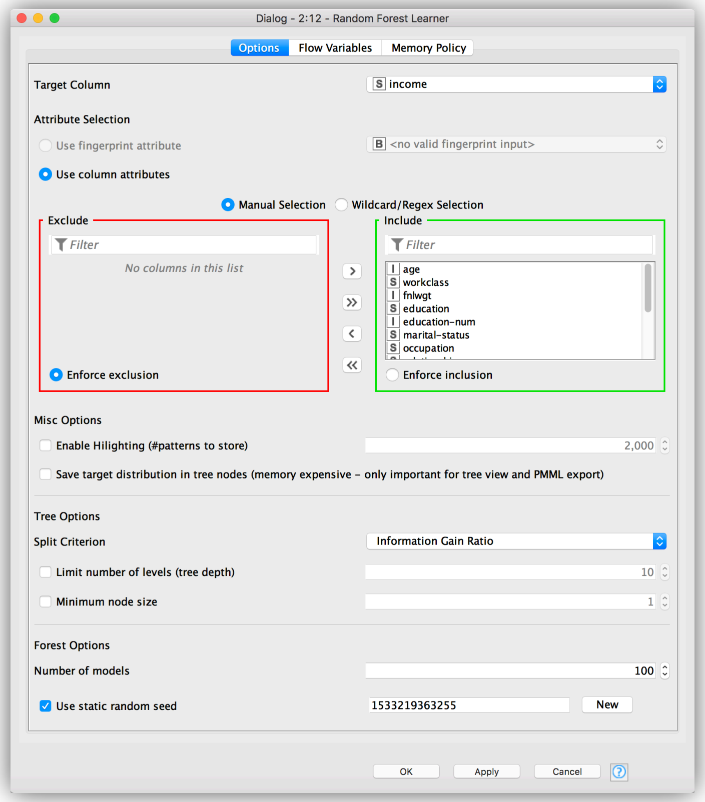


Row ID	CID	SMILES	Solub_...	SlogP
Row0_Row...	6		2 moderate	2.156
Row106_R...	3475		2 moderate	1.63
Row107_R...	3476		3 high	3.074
Row108_R...	3478		3 high	2.078
Row109_R...	3559		2 moderate	4.426
Row10_Ro...	774		2 moderate	-0.089

Random Forest Learner

- The output model describes a random forest and is applied in the corresponding predictor node using a simple majority vote
- The statistics table on the attributes tells how often each attribute…
  - … is used in the first three splits
  - … was a possible candidate in the first three splits





Target Column: S income

Attribute Selection

Use fingerprint attribute: B <no valid fingerprint input>

[✓ Use column attributes]

[✓ Manual Selection] Wildcard/Regex Selection

Exclude
Filter
No columns in this list
[✓ Enforce exclusion]

Include
Filter
I age
S workclass
I fnlwgt
S education
I education-num
S marital-status
S occupation
Enforce inclusion

Misc Options

Enable Hiliting (#patterns to store): 2,000

Save target distribution in tree nodes (memory expensive – only important for tree view and PMML export)

Tree Options

Split Criterion: Information Gain Ratio

Limit number of levels (tree depth): 10

Minimum node size: 1

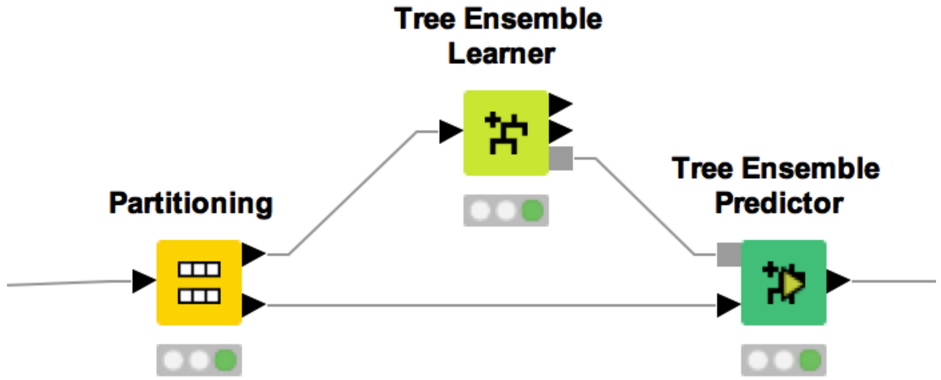
Forest Options

Number of models: 100

[✓ Use static random seed]: 1533219363255

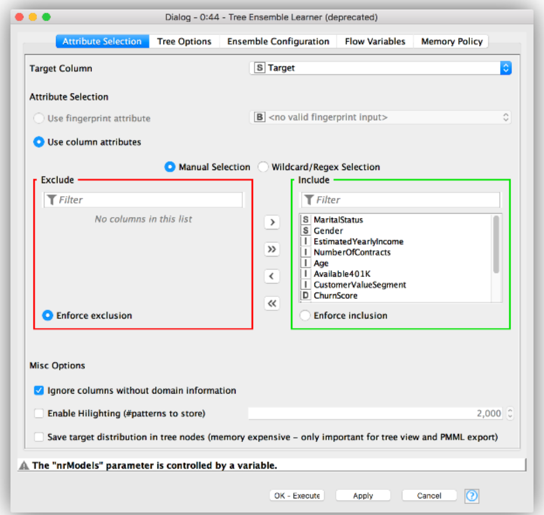
Tree Ensembles

- Random Forest is a specific tree ensemble with predefined ensemble parameters
- The Tree Ensemble Learner node allows to train different tree ensembles
  - E.g. different row and column sampling options
- Optimization of a tree ensemble is complex due to a surplus of configuration options
  - Number of models
  - Number of columns
  - Number of rows
  - Tree depth
  - ...



Tree Ensemble Learner

- Choose which columns to include
- Configure a prototype tree (depth, split criteria etc.)
- Setup ensemble parameters (model count, row/column subsampling)



Target Column: S Target

Attribute Selection

○ Use fingerprint attribute B <no valid fingerprint input>

● Use column attributes

● Manual Selection ○ Wildcard/Regex Selection

Exclude: No columns in this list ● Enforce exclusion

Include: S MaritalStatus, S Gender, I EstimatedYearlyIncome, I NumberOfContracts, I Age, I Available401K, I CustomerValueSegment, D ChurnScore ○ Enforce inclusion

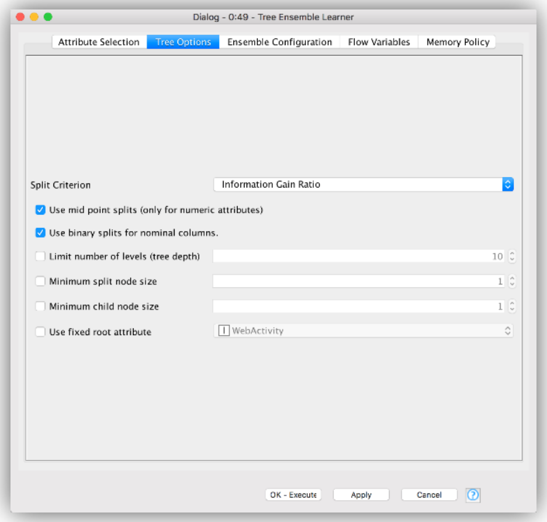
Misc Options

✓ Ignore columns without domain information

☐ Enable Hilighting (#patterns to store) 2,000

☐ Save target distribution in tree nodes (memory expensive – only important for tree view and PMML export)

⚠ The "nrModels" parameter is controlled by a variable.



Split Criterion: Information Gain Ratio

✓ Use mid point splits (only for numeric attributes)

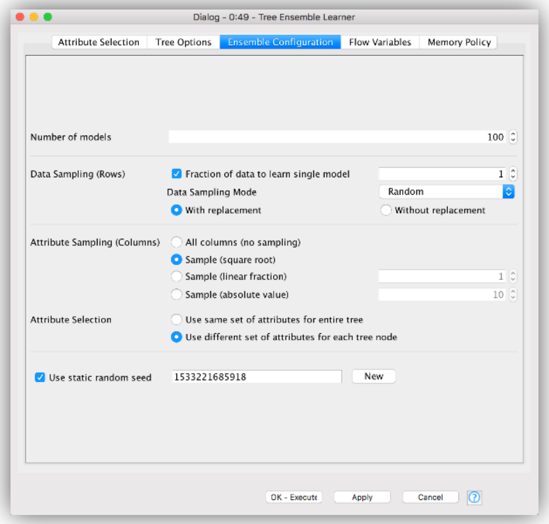
✓ Use binary splits for nominal columns.

☐ Limit number of levels (tree depth) 10

☐ Minimum split node size 1

☐ Minimum child node size 1

☐ Use fixed root attribute I WebActivity



Number of models: 100

Data Sampling (Rows): ✓ Fraction of data to learn single model 1

Data Sampling Mode: Random

● With replacement ○ Without replacement

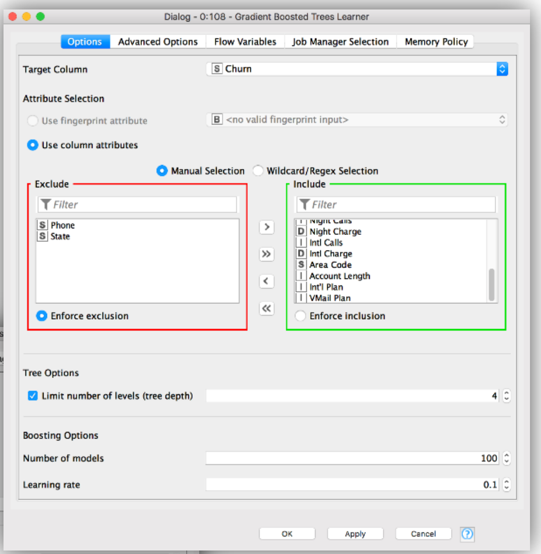
Attribute Sampling (Columns): ○ All columns (no sampling) ● Sample (square root) ○ Sample (linear fraction) 1 ○ Sample (absolute value) 10

Attribute Selection: ○ Use same set of attributes for entire tree ● Use different set of attributes for each tree node

✓ Use static random seed 1533221685918

Gradient Boosted Trees Learner

- Another algorithm for creating ensembles of decision trees
- Starts with a shallow tree
- Builds additional trees to fit the residual errors
- Can introduce randomness in choice of data subsets (“stochastic gradient boosting”) and in variable choice (Advanced Options)



Dialog - 0:108 - Gradient Boosted Trees Learner

Target Column: Churn

Attribute Selection

- Use fingerprint attribute: <no valid fingerprint input>
- [✓ Use column attributes]

[✓ Manual Selection] Wildcard/Regex Selection

Exclude: Phone, State — [✓ Enforce exclusion]

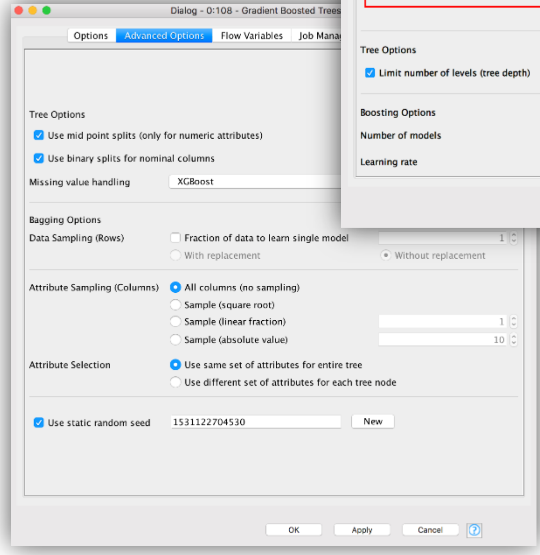
Include: Night Calls, Night Charge, Intl Calls, Intl Charge, Area Code, Account Length, Int'l Plan, VMail Plan — Enforce inclusion

Tree Options

- ✓ Limit number of levels (tree depth): 4

Boosting Options

- Number of models: 100
- Learning rate: 0.1



Dialog - 0:108 - Gradient Boosted Trees

Tree Options

- ✓ Use mid point splits (only for numeric attributes)
- ✓ Use binary splits for nominal columns

Missing value handling: XGBoost

Bagging Options

Data Sampling (Rows): ☐ Fraction of data to learn single model: 1 — With replacement [✓ Without replacement]

Attribute Sampling (Columns): [✓ All columns (no sampling)] Sample (square root) Sample (linear fraction): 1 Sample (absolute value): 10

Attribute Selection: [✓ Use same set of attributes for entire tree] Use different set of attributes for each tree node

✓ Use static random seed: 1531122704530

Gradient Boosted Trees Learner

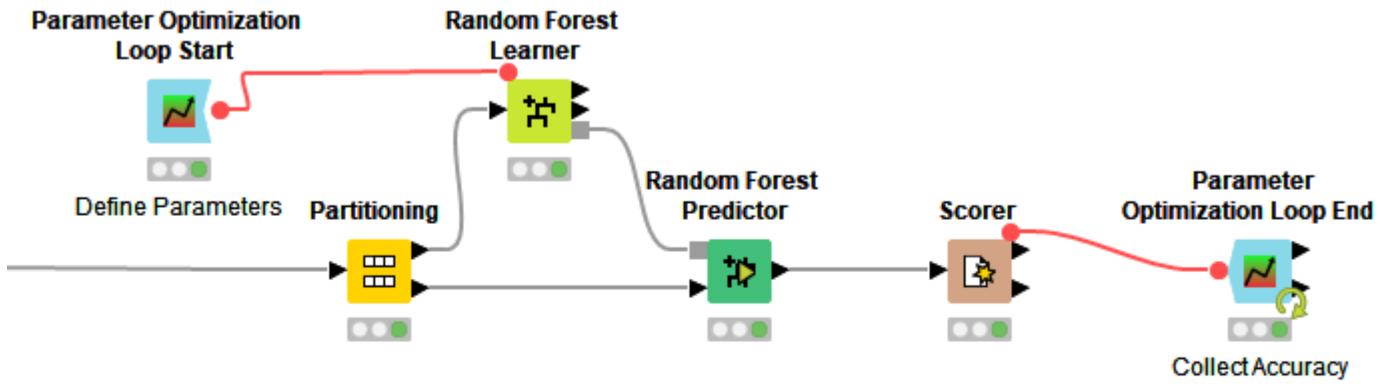
Advanced Data Mining Exercise, Activity I

Start with exercise: *Advanced Data Mining, Activity I*

- Read *CurrentDetailData.table* data
- Partition the data 50/50 using stratified sampling on the "Target" column
- Train and apply a Random Forest model to predict the "Target" column
- Use a tree depth of 5 and 50 models

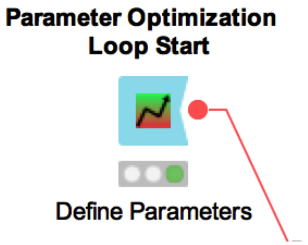
Parameter Optimization

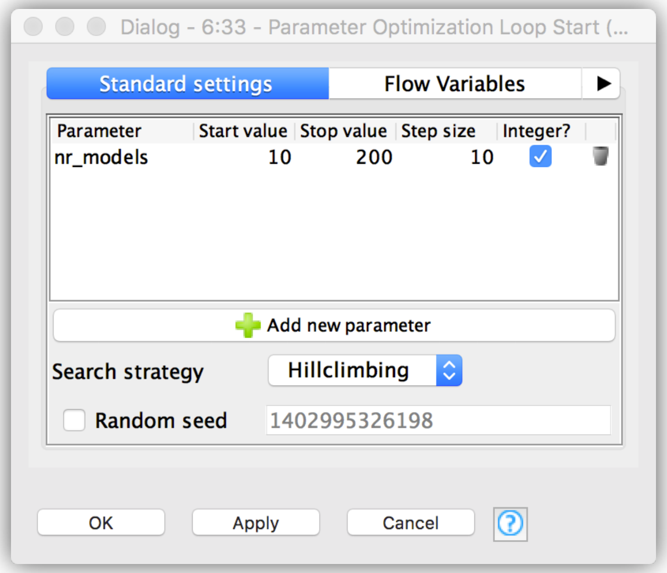
- Some modeling approaches are very sensitive to their configuration.
- Calculating optimum settings is not always possible.
- Parameter Optimization loops may help find a good configuration.



Parameter Optimization Loop Start

- Define parameters to optimize
- Set upper/lower bounds and step sizes (and flag integers)
- Choose an optimization method
  - Brute force for maximum accuracy, but slower computation
  - Hillclimbing for better faster runtimes, but may get stuck in local optimum settings
  - Random search to randomly search for parameter values within a given range
  - Bayesian Optimization (TPE)





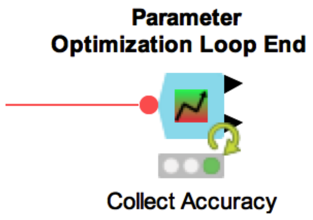
Parameter	Start value	Stop value	Step size	Integer?
nr_models	10	200	10	✓

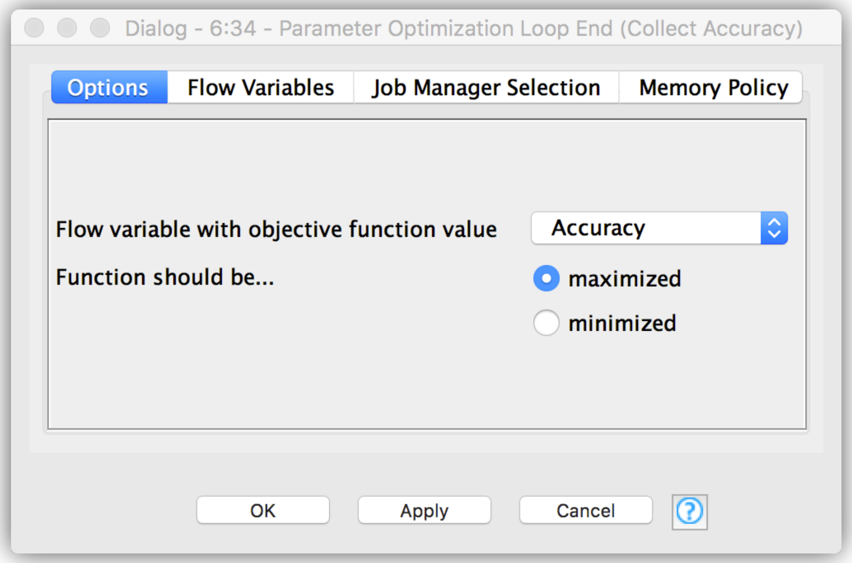
Search strategy: Hillclimbing

☐ Random seed 1402995326198

Parameter Optimization Loop End

- Collects a value to optimize as Flow Variable.
- Value may be maximized (accuracy) or minimized (error)





Flow variable with objective function value: Accuracy

Function should be... [✓ maximized] minimized

Advanced Data Mining Exercise, Activity II

Start with exercise: *Advanced Data Mining, Activity II*

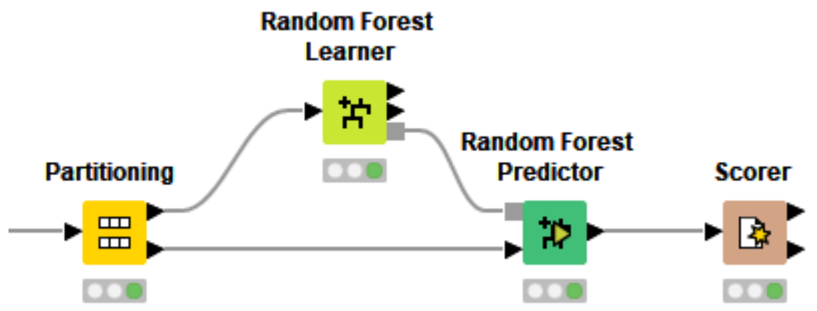
- Add a parameter optimization loop to your model training process
- Use Hillclimbing to determine the optimum number of models (min=10, max=200, step=10, int = yes)
- Use maximum accuracy as the objective value
- What is the best number of models?

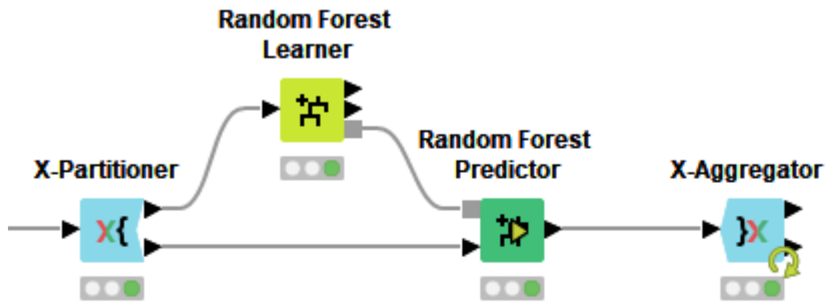
(Hint: don't forget to use the flow variable in the Random Forest Learner node)

- (Optional): Train and save a model with the best parameter set (using a Table Row to Variable, Random Forest Learner, and Model Writer node)

Cross Validation

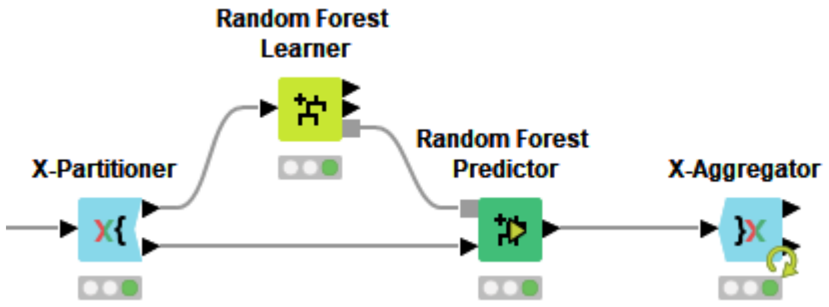
- Used to evaluate model stability
- Re-execute the modeling process many times using different data partitions
- Collect aggregated statistics on model accuracy

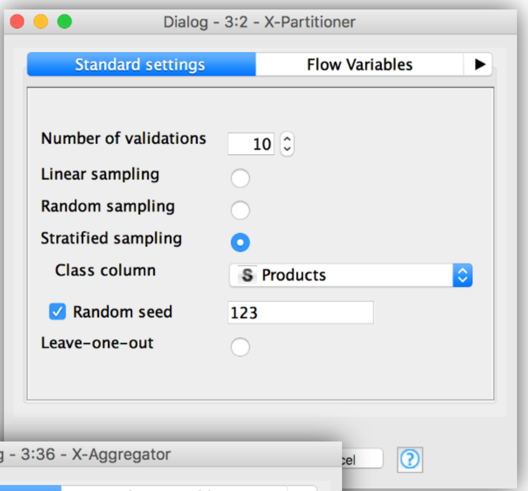




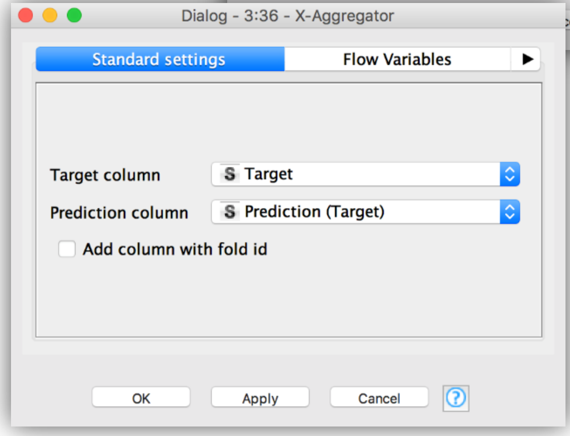
Example: Cross Validation

- X-Partitioner → X-Aggregator
- X-Partitioner replaces Partition
- X-Aggregator replaces Scorer
- Can be used with any learner/predictor




Dialog - 3:2 - X-Partitioner

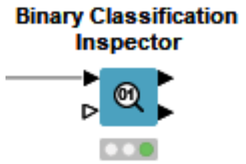
Number of validations: 10
Linear sampling
Random sampling
[✓ Stratified sampling]
Class column: S Products
[✓] Random seed: 123
Leave-one-out

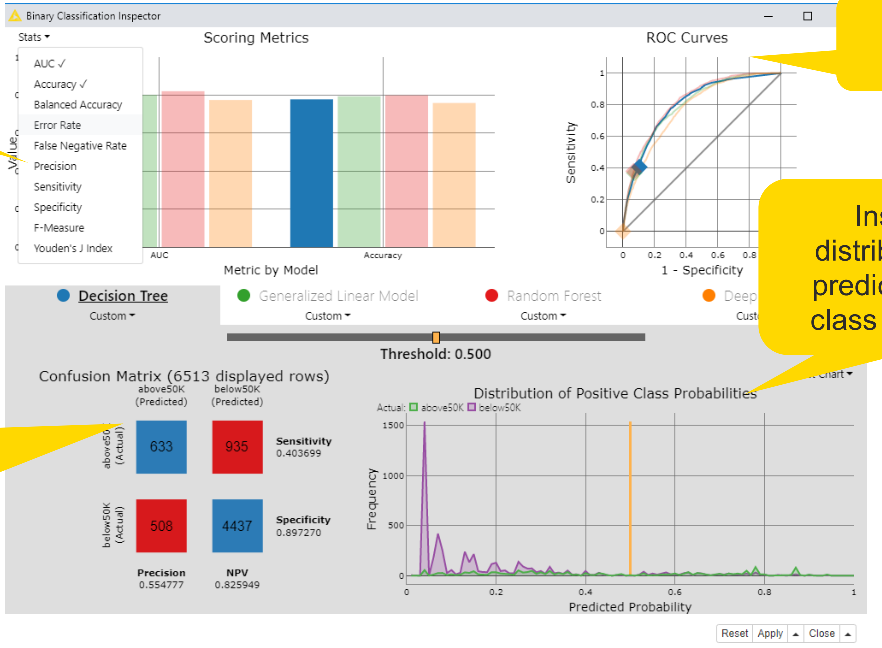

Dialog - 3:36 - X-Aggregator

Target column: S Target
Prediction column: S Prediction (Target)
[ ] Add column with fold id

Binary Classification Inspector

- Inspect and compare the performances of classification models
- Adjust the classification threshold according to the goal of your model





Confusion Matrix (6513 displayed rows)

	above50K (Predicted)	below50K (Predicted)	
above50K (Actual)	633	935	Sensitivity 0.403699
below50K (Actual)	508	4437	Specificity 0.897270
	Precision 0.554777	NPV 0.825949	

Select the scoring metrics to compare

Compare the ROC curves

Inspect the distribution of the predicted positive class probabilities

Inspect the class assignments for different classification thresholds

Advanced Data Mining Exercise, Activity III

Start with exercise: *Advanced Data Mining, Activity III*

- Create a 10-fold cross validation for your model
- Take a look at the error rates produced by the different iterations. Does the model seem stable?

Advanced Data Mining Exercise, Activity IV (Optional)

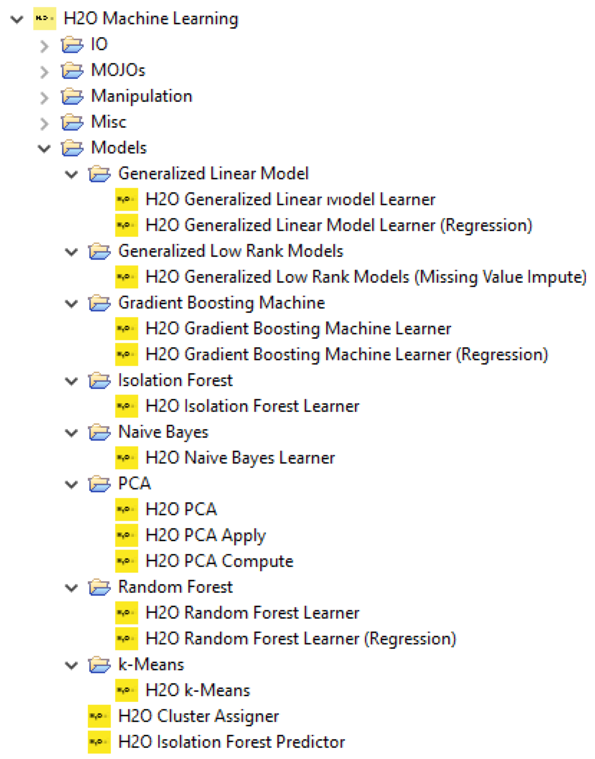
Start with exercise: *Advanced Data Mining, Activity IV (Optional)*

Goal: Train a decision tree and a random forest model and compare their performance

- Partition the data 50/50 using stratified sampling on the "Target" column
- Train and apply a Random Forest model to predict the "Target" column
- Train and apply a Decision Tree model to predict the "Target" column
- Combine the performances of both models (Column Appender node)
- Evaluate the performances of the models (Binary Classification Inspector node) Which model performs better?

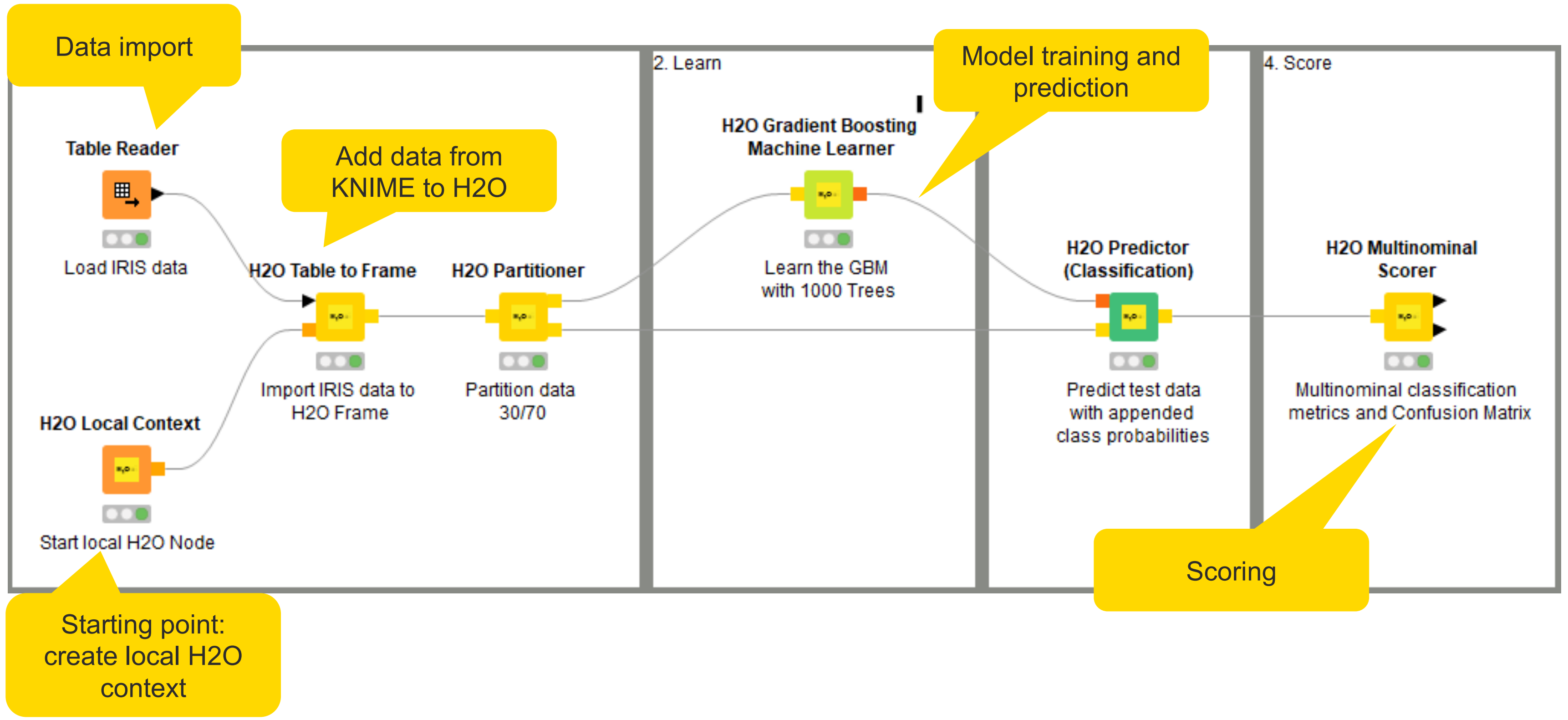
H2O Integration

- KNIME integrates the H2O machine learning library
- H2O: Open source, focus on scalability and performance
- Supports many different models
  - Generalized Linear Model
  - Gradient Boosting Machine
  - Random Forest
  - k-Means, PCA, Naive Bayes, Isolation Forest, etc. and more to come!
- Includes support for MOJO model objects for deployment



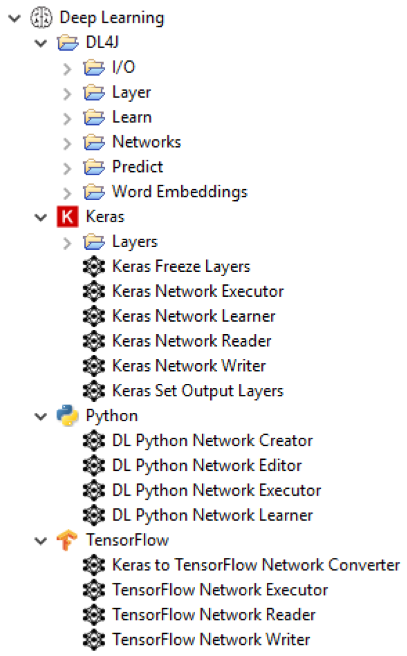
```
H2O Machine Learning
  IO
  MOJOs
  Manipulation
  Misc
  Models
    Generalized Linear Model
      H2O Generalized Linear Model Learner
      H2O Generalized Linear Model Learner (Regression)
    Generalized Low Rank Models
      H2O Generalized Low Rank Models (Missing Value Impute)
    Gradient Boosting Machine
      H2O Gradient Boosting Machine Learner
      H2O Gradient Boosting Machine Learner (Regression)
    Isolation Forest
      H2O Isolation Forest Learner
    Naive Bayes
      H2O Naive Bayes Learner
    PCA
      H2O PCA
      H2O PCA Apply
      H2O PCA Compute
    Random Forest
      H2O Random Forest Learner
      H2O Random Forest Learner (Regression)
    k-Means
      H2O k-Means
    H2O Cluster Assigner
    H2O Isolation Forest Predictor
```

H2O Integration - Example

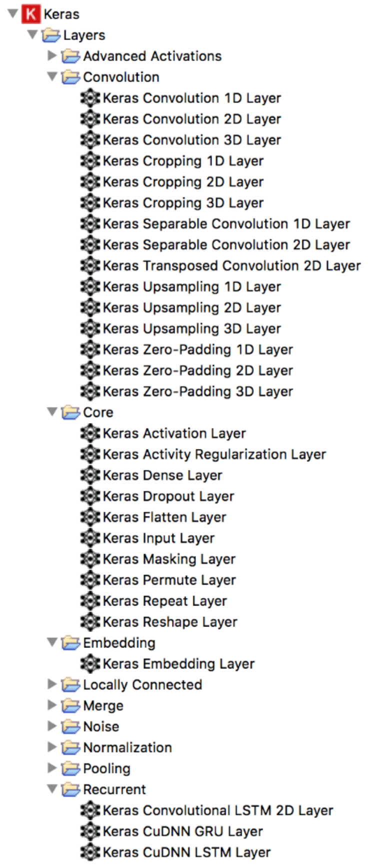

Data import
Add data from KNIME to H2O
Model training and prediction
Scoring
Starting point: create local H2O context
2. Learn
4. Score
Table Reader
Load IRIS data
H2O Table to Frame
Import IRIS data to H2O Frame
H2O Partitioner
Partition data 30/70
H2O Local Context
Start local H2O Node
H2O Gradient Boosting Machine Learner
Learn the GBM with 1000 Trees
H2O Predictor (Classification)
Predict test data with appended class probabilities
H2O Multinominal Scorer
Multinominal classification metrics and Confusion Matrix

Deep Learning Integration

- Keras integration:
  - Many different layer nodes.
  - Define your network, train and apply a network without a single line of code.
- DL Python integration
- TensorFlow integration

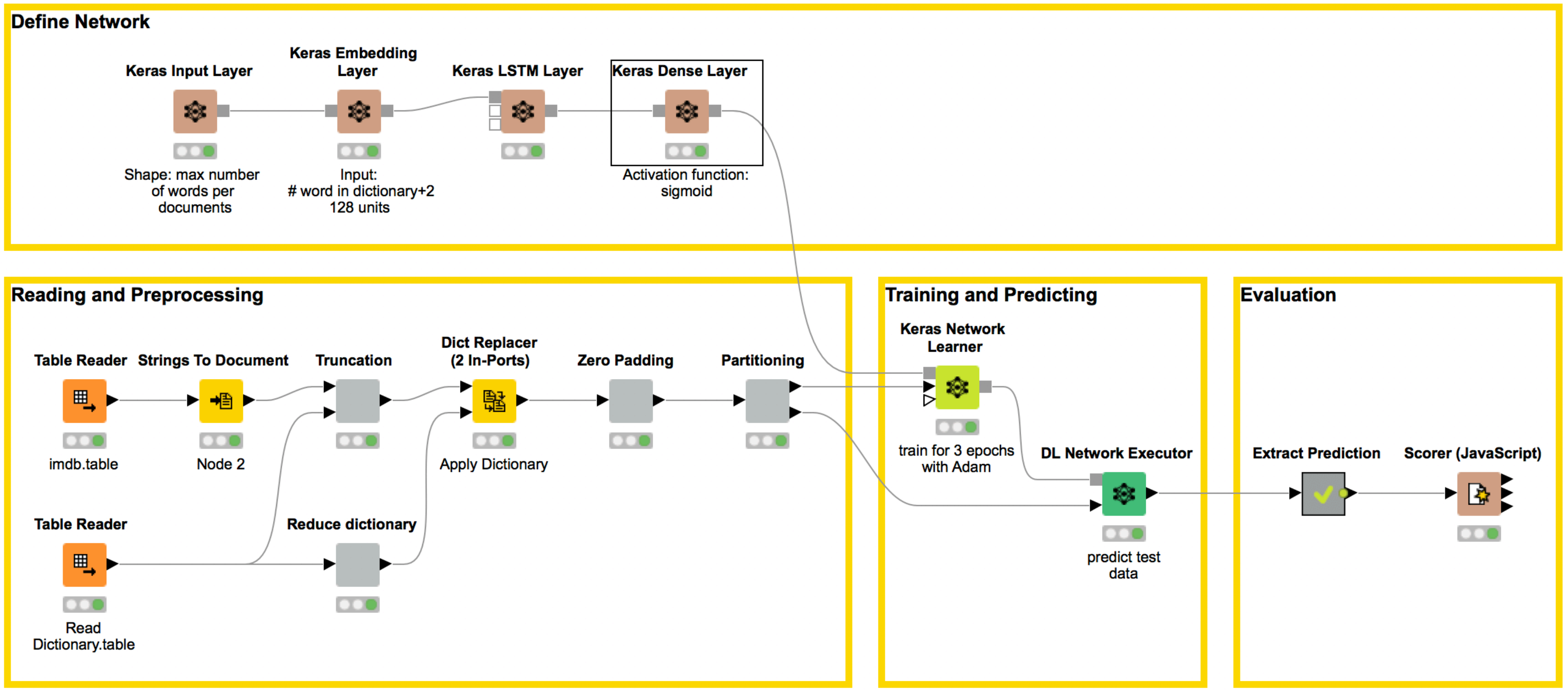


Deep Learning
- DL4J
  - I/O
  - Layer
  - Learn
  - Networks
  - Predict
  - Word Embeddings
- Keras
  - Layers
  - Keras Freeze Layers
  - Keras Network Executor
  - Keras Network Learner
  - Keras Network Reader
  - Keras Network Writer
  - Keras Set Output Layers
- Python
  - DL Python Network Creator
  - DL Python Network Editor
  - DL Python Network Executor
  - DL Python Network Learner
- TensorFlow
  - Keras to TensorFlow Network Converter
  - TensorFlow Network Executor
  - TensorFlow Network Reader
  - TensorFlow Network Writer



Keras
- Layers
  - Advanced Activations
  - Convolution
    - Keras Convolution 1D Layer
    - Keras Convolution 2D Layer
    - Keras Convolution 3D Layer
    - Keras Cropping 1D Layer
    - Keras Cropping 2D Layer
    - Keras Cropping 3D Layer
    - Keras Separable Convolution 1D Layer
    - Keras Separable Convolution 2D Layer
    - Keras Transposed Convolution 2D Layer
    - Keras Upsampling 1D Layer
    - Keras Upsampling 2D Layer
    - Keras Upsampling 3D Layer
    - Keras Zero-Padding 1D Layer
    - Keras Zero-Padding 2D Layer
    - Keras Zero-Padding 3D Layer
  - Core
    - Keras Activation Layer
    - Keras Activity Regularization Layer
    - Keras Dense Layer
    - Keras Dropout Layer
    - Keras Flatten Layer
    - Keras Input Layer
    - Keras Masking Layer
    - Keras Permute Layer
    - Keras Repeat Layer
    - Keras Reshape Layer
  - Embedding
    - Keras Embedding Layer
  - Locally Connected
  - Merge
  - Noise
  - Normalization
  - Pooling
  - Recurrent
    - Keras Convolutional LSTM 2D Layer
    - Keras CuDNN GRU Layer
    - Keras CuDNN LSTM Layer

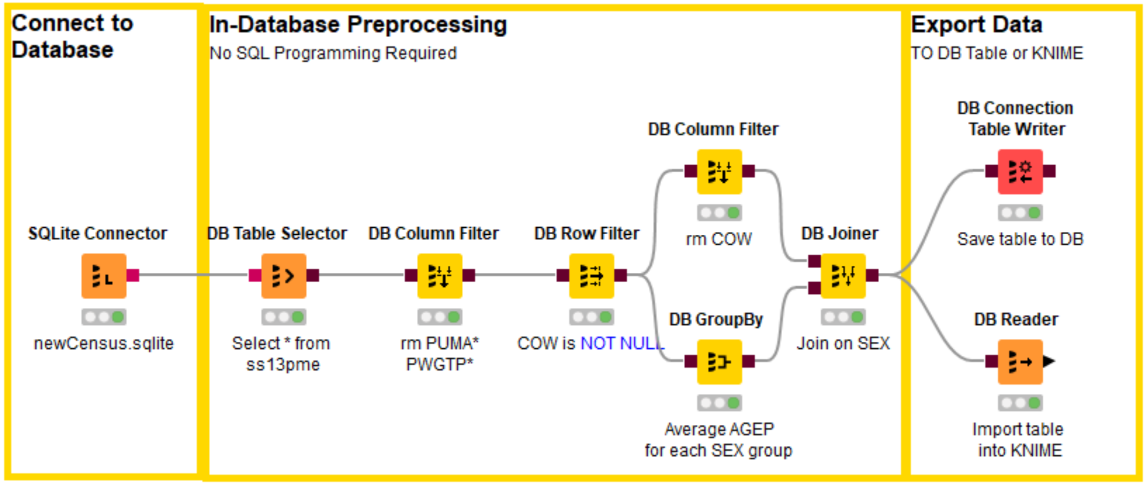
Sentiment Analysis Using Keras



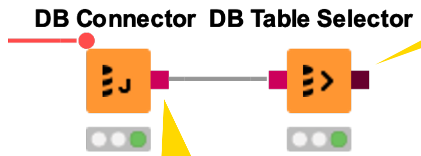
Databases

Database Extension

- Visually assemble complex SQL statements (no SQL coding needed)
- Connect to all JDBC-compliant databases
- Harness the power of your database within KNIME
- Complete rewrite in KNIME Analytics Platform 4.0



Database Port Types

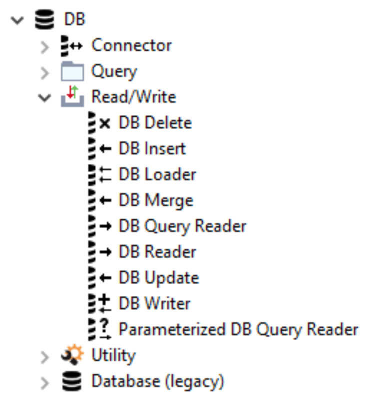


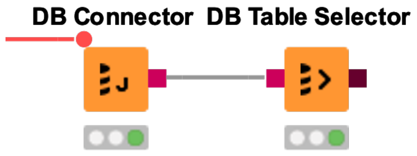
DB Data Port
- Connection information
- SQL statement

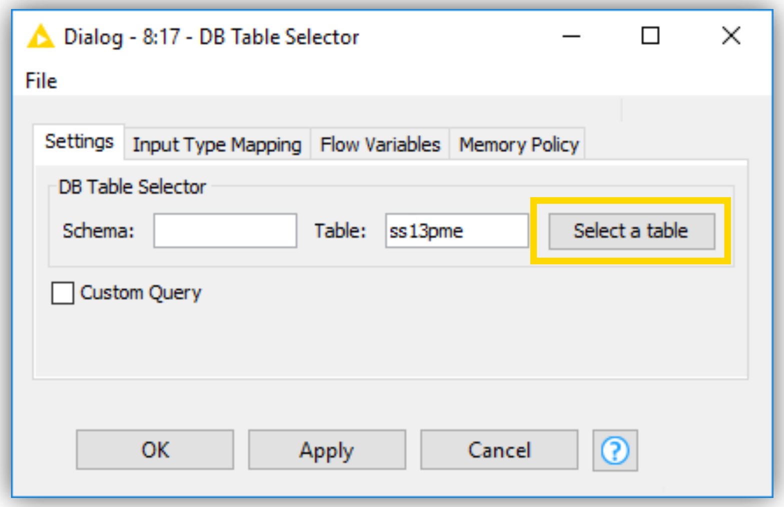
DB Connection Port
Connection information

DB Table Selector

- Takes connection information and constructs a query
- Explore DB metadata
- Outputs a SQL query







DB Table Selector

Schema:

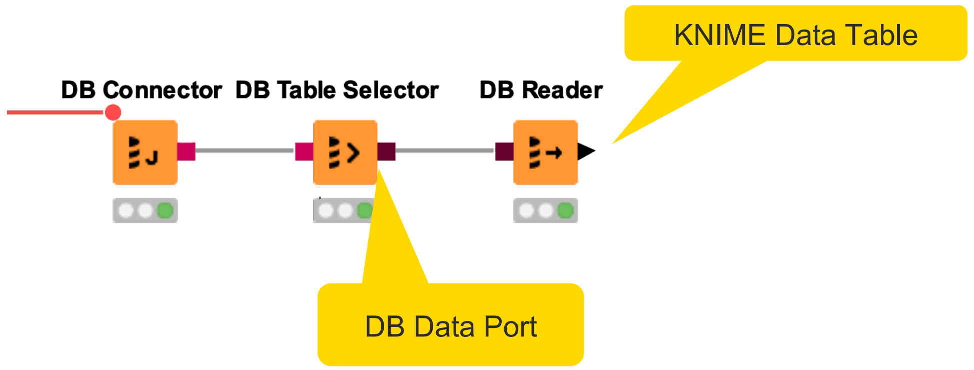
Table: ss13pme

Select a table

☐ Custom Query

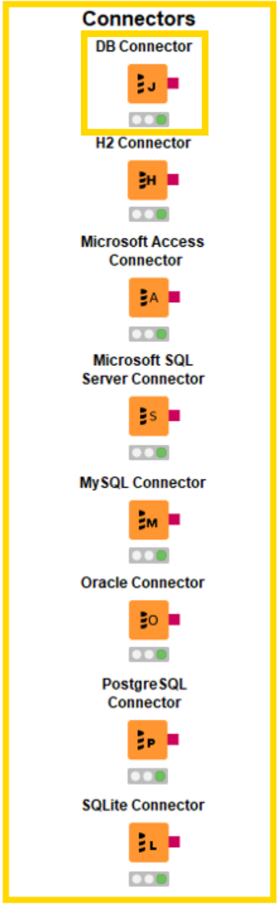
DB Reader

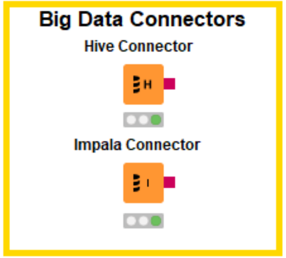
- Executes incoming SQL Query on Database
- Reads results into a KNIME data table



Database Connectors

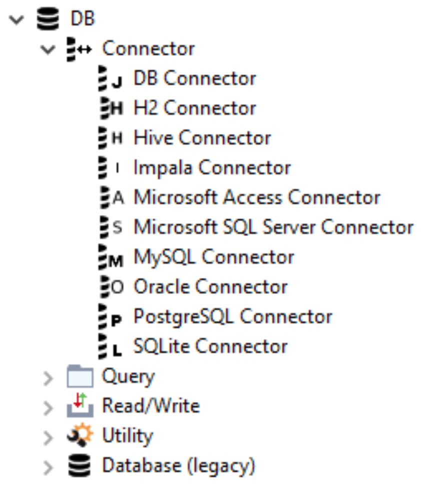
- Dedicated nodes to connect to specific Databases
  - Necessary JDBC driver included
  - Easy to use
  - Import DB specific behavior/capability
- Hive, Impala connectors part of the KNIME Big Data Connectors extension
- General DB Connector
  - Can connect to any JDBC source
  - Register new JDBC driver via File -> Preferences -> KNIME -> Databases



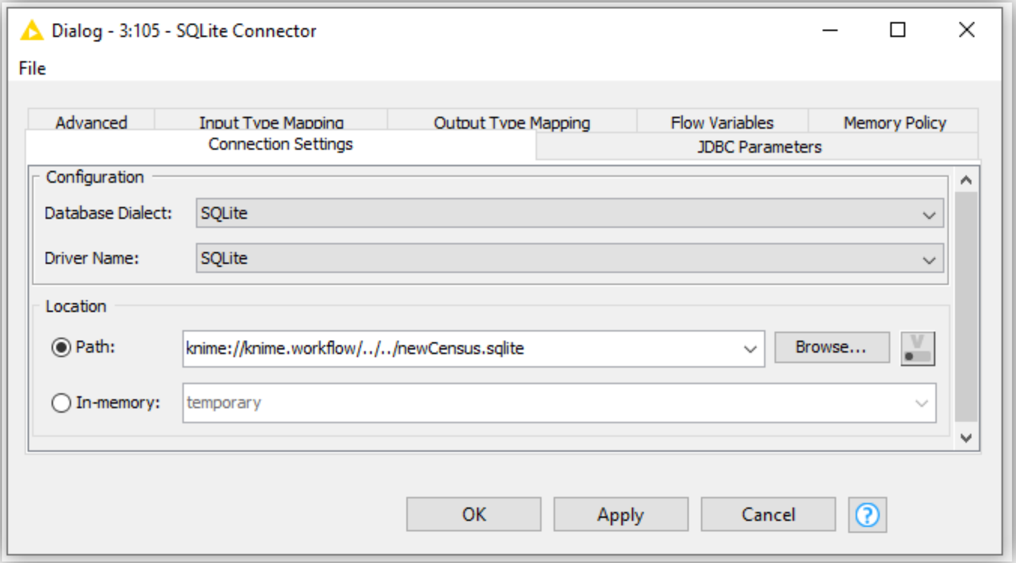


Dedicated Database Connectors

- MySQL, MS SQL Server, Postgres, SQLite, Amazon Redshift, etc.
- Propagate connection information to other DB nodes



- DB
  - Connector
    - DB Connector
    - H2 Connector
    - Hive Connector
    - Impala Connector
    - Microsoft Access Connector
    - Microsoft SQL Server Connector
    - MySQL Connector
    - Oracle Connector
    - PostgreSQL Connector
    - SQLite Connector
  - Query
  - Read/Write
  - Utility
  - Database (legacy)



Dialog - 3:105 - SQLite Connector

File

Tabs: Advanced | Input Type Mapping | Output Type Mapping | Flow Variables | Memory Policy | Connection Settings | JDBC Parameters

Configuration

Database Dialect: SQLite

Driver Name: SQLite

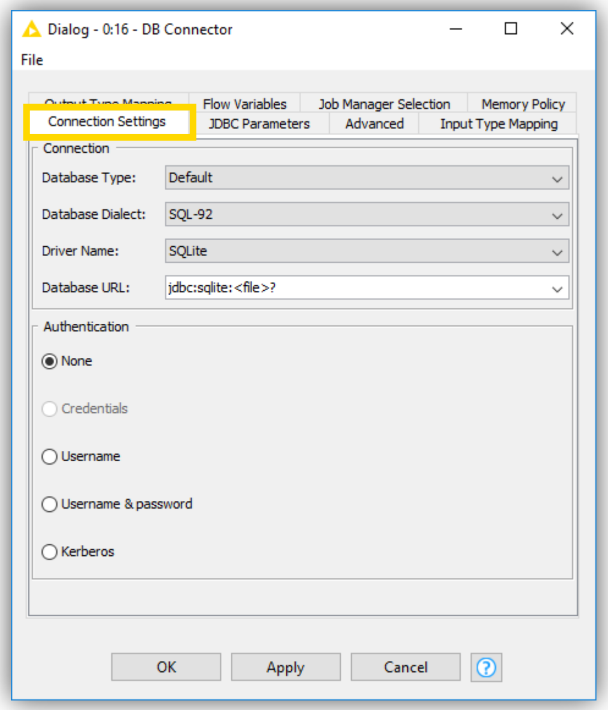
Location

(●) Path: knime://knime.workflow/../../newCensus.sqlite Browse...

( ) In-memory: temporary

OK | Apply | Cancel

«General» DB Connector Node



Dialog - 0:16 - DB Connector

File

Output Type Mapping | Flow Variables | Job Manager Selection | Memory Policy | Connection Settings | JDBC Parameters | Advanced | Input Type Mapping

Connection

Database Type: Default

Database Dialect: SQL-92

Driver Name: SQLite

Database URL: jdbc:sqlite:<file>?

Authentication

[✓ None]

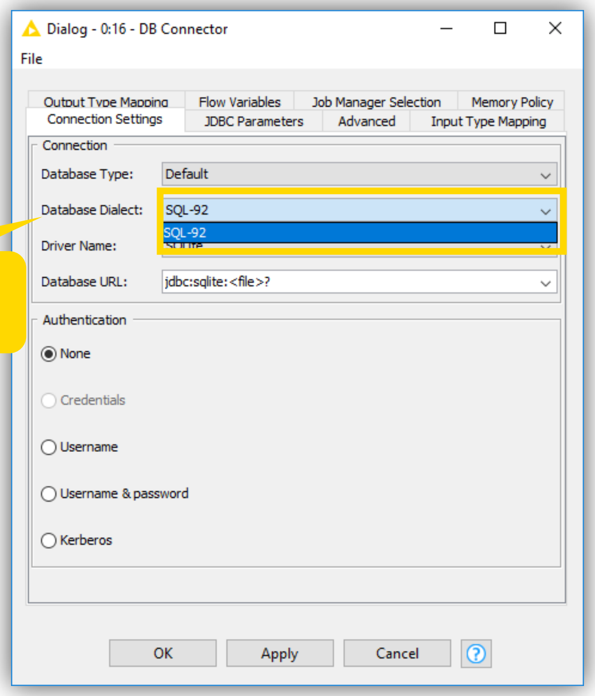
Credentials

Username

Username & password

Kerberos

OK | Apply | Cancel



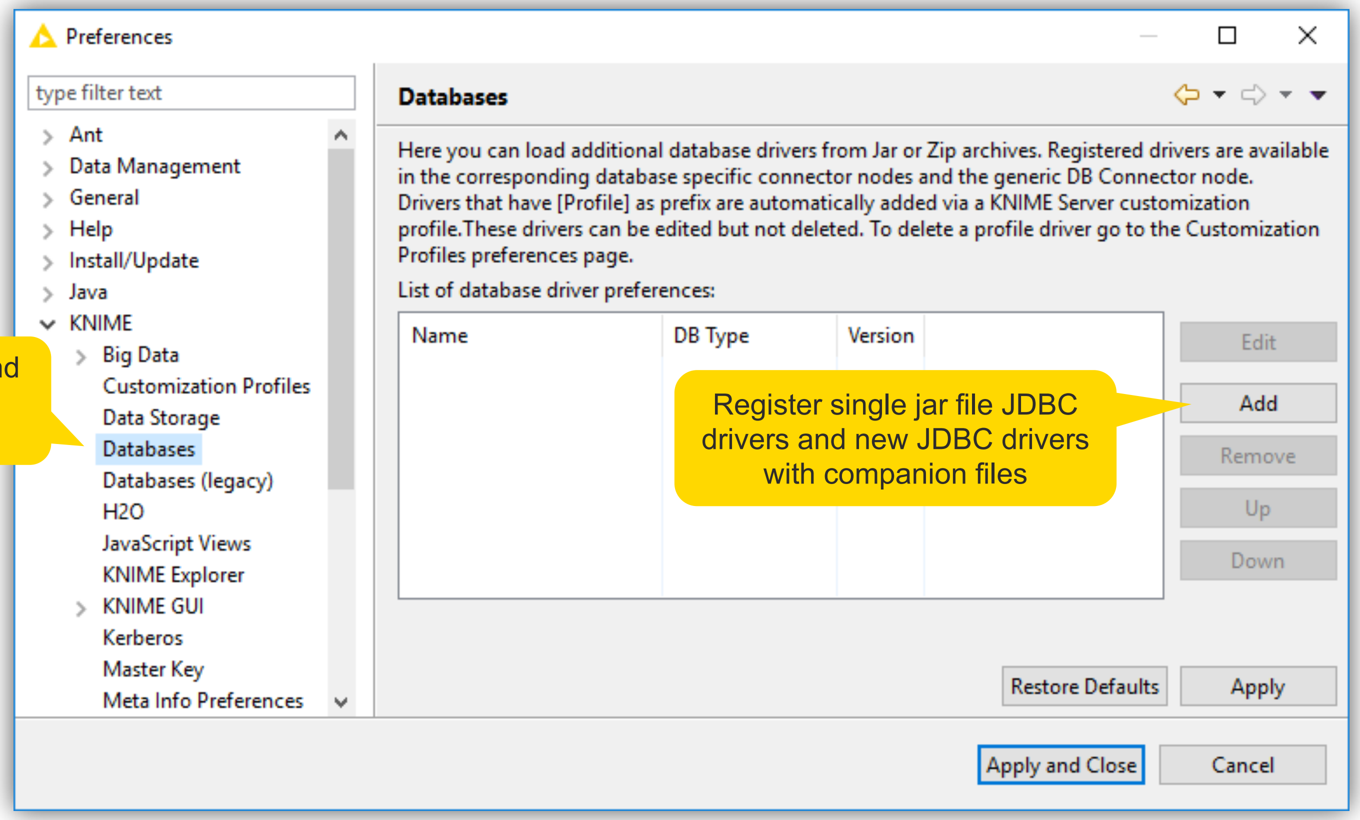
Dialog - 0:16 - DB Connector

Database Dialect: SQL-92

Database type defines SQL dialect

Register JDBC Driver

Open KNIME and go to File -> Preferences



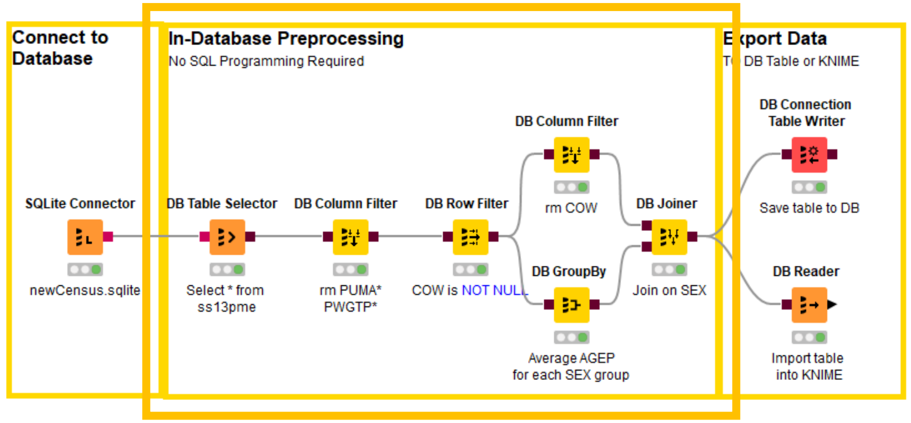
Here you can load additional database drivers from Jar or Zip archives. Registered drivers are available in the corresponding database specific connector nodes and the generic DB Connector node. Drivers that have [Profile] as prefix are automatically added via a KNIME Server customization profile.These drivers can be edited but not deleted. To delete a profile driver go to the Customization Profiles preferences page.

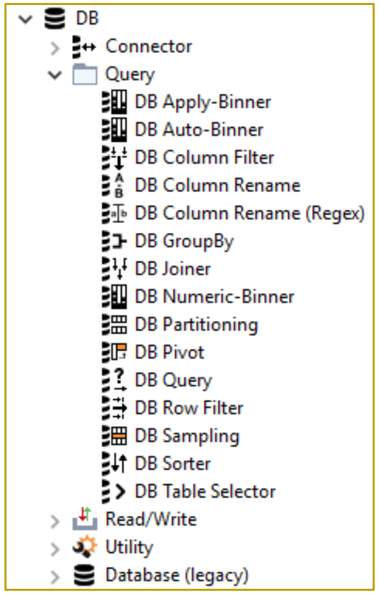
List of database driver preferences:

Register single jar file JDBC drivers and new JDBC drivers with companion files

In-Database Processing

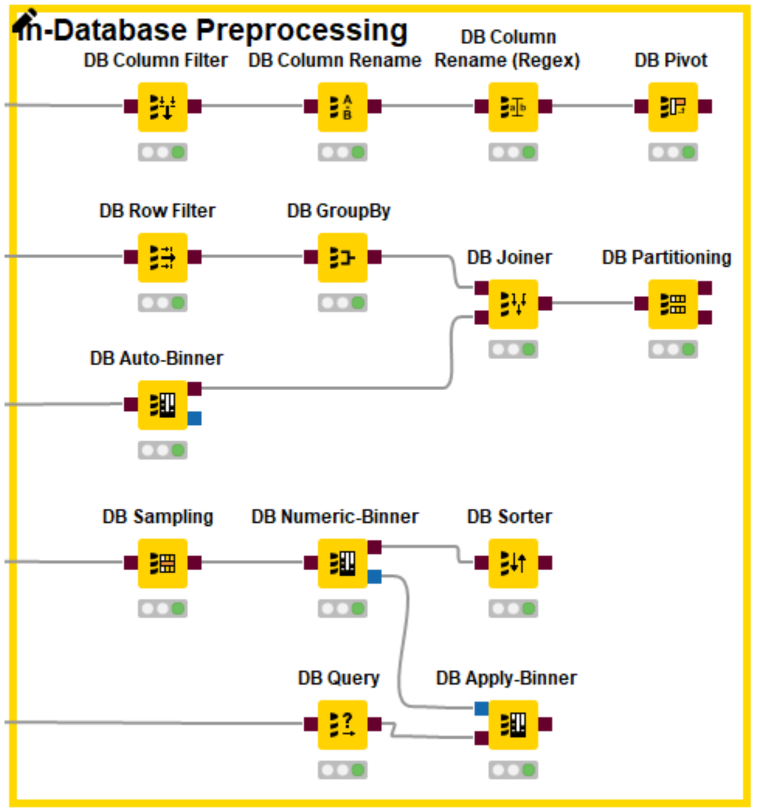
- Database Manipulation node generates a SQL query on top of the input SQL query (brown square port)





Query Nodes

- Filter rows and columns
- Join tables/queries
- Extract samples
- Bin numeric columns
- Sort your data
- Write your own query
- Aggregate your data
- Partition your data



Data Aggregation

RowID	Group	Value
r1	m	2
r2	f	3
r3	m	1
r4	f	5
r5	f	7
r6	m	5

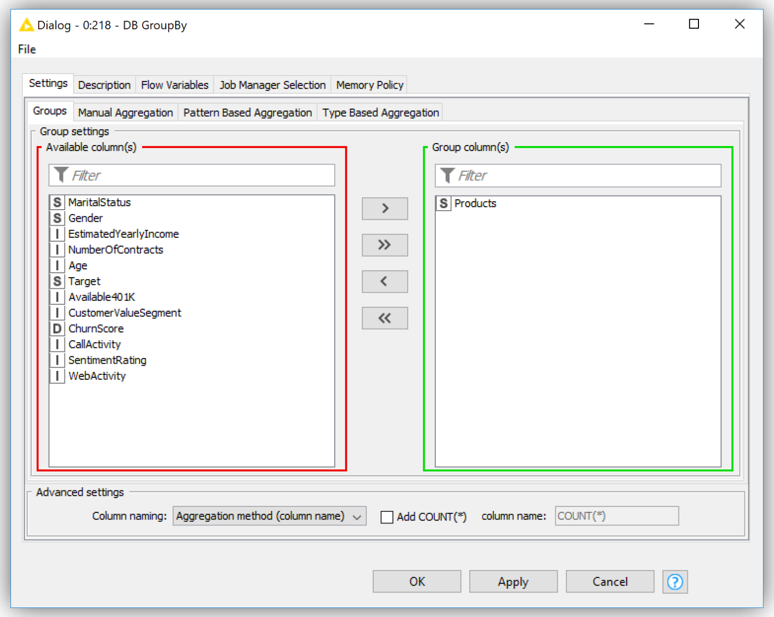
RowID	Group	Sum(Value)
r1+r3+r6	m	8
r2+r4+r5	f	15

Aggregated on “Group” by method:
sum(“Value”)

DB GroupBy

Aggregate rows to summarize data

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



Dialog - 0:218 - DB GroupBy

File

Settings | Description | Flow Variables | Job Manager Selection | Memory Policy

Groups | Manual Aggregation | Pattern Based Aggregation | Type Based Aggregation

Group settings

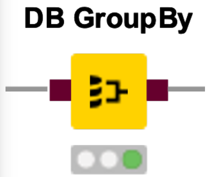
Available column(s): MaritalStatus, Gender, EstimatedYearlyIncome, NumberOfContracts, Age, Target, Available401K, CustomerValueSegment, ChurnScore, CallActivity, SentimentRating, WebActivity

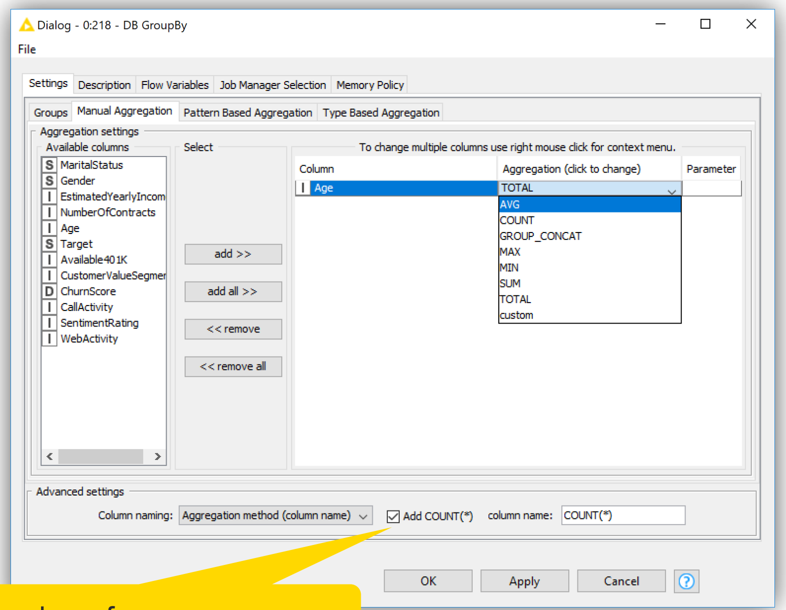
Group column(s): Products

Advanced settings

Column naming: Aggregation method (column name) ☐ Add COUNT(*) column name: COUNT(*)

OK | Apply | Cancel





Dialog - 0:218 - DB GroupBy

File

Settings | Description | Flow Variables | Job Manager Selection | Memory Policy

Groups | Manual Aggregation | Pattern Based Aggregation | Type Based Aggregation

Aggregation settings

Available columns: MaritalStatus, Gender, EstimatedYearlyIncom, NumberOfContracts, Age, Target, Available401K, CustomerValueSegmer, ChurnScore, CallActivity, SentimentRating, WebActivity

Select: add >> | add all >> | << remove | << remove all

To change multiple columns use right mouse click for context menu.

Column	Aggregation (click to change)	Parameter
Age	TOTAL	

AVG, COUNT, GROUP_CONCAT, MAX, MIN, SUM, TOTAL, custom

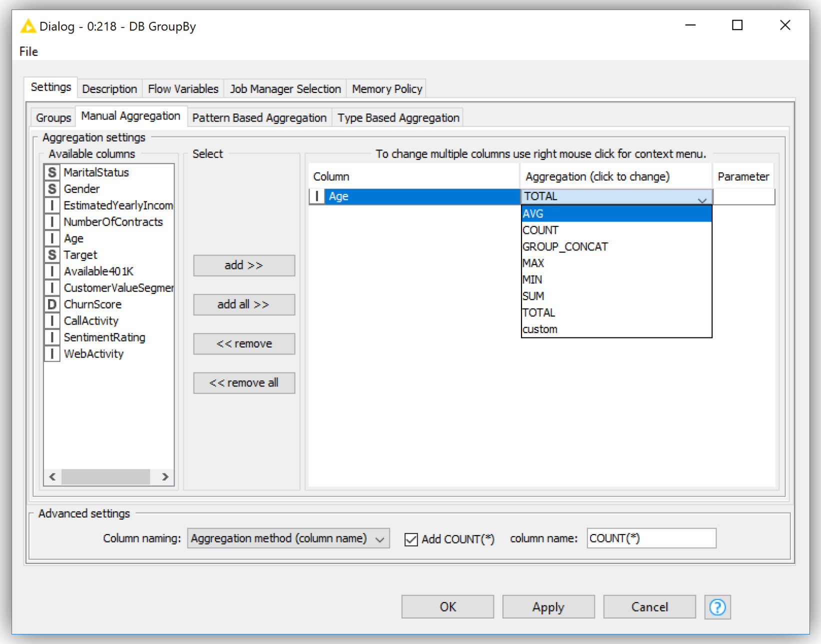
Advanced settings

Column naming: Aggregation method (column name) ☑ Add COUNT(*) column name: COUNT(*)

OK | Apply | Cancel

Returns number of rows per group

DB GroupBy – DB Specific Aggregation Methods



Aggregation settings

Available columns: MaritalStatus, Gender, EstimatedYearlyIncom, NumberOfContracts, Age, Target, Available401K, CustomerValueSegmer, ChurnScore, CallActivity, SentimentRating, WebActivity

Select — To change multiple columns use right mouse click for context menu.

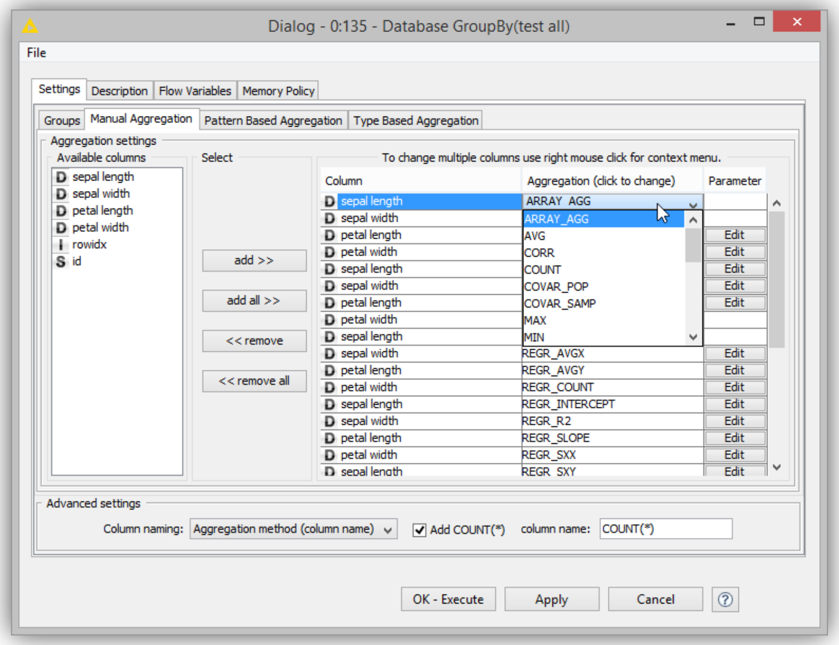
Column	Aggregation (click to change)	Parameter
Age	TOTAL	

AVG
COUNT
GROUP_CONCAT
MAX
MIN
SUM
TOTAL
custom

Advanced settings

Column naming: Aggregation method (column name) [✓] Add COUNT(*) column name: COUNT(*)

SQLite: 7 aggregation functions



Aggregation settings

Available columns: sepal length, sepal width, petal length, petal width, rowidx, id

Select — To change multiple columns use right mouse click for context menu.

Column	Aggregation (click to change)	Parameter
sepal length	ARRAY_AGG	
sepal width		
petal length		Edit
petal width		Edit
sepal length		Edit
sepal width		Edit
petal length		Edit
petal width		
sepal length		
sepal width	REGR_AVGX	Edit
petal length	REGR_AVGY	Edit
petal width	REGR_COUNT	Edit
sepal length	REGR_INTERCEPT	Edit
sepal width	REGR_R2	Edit
petal length	REGR_SLOPE	Edit
petal width	REGR_SXX	Edit
sepal length	REGR_SXY	Edit

ARRAY_AGG
AVG
CORR
COUNT
COVAR_POP
COVAR_SAMP
MAX
MIN

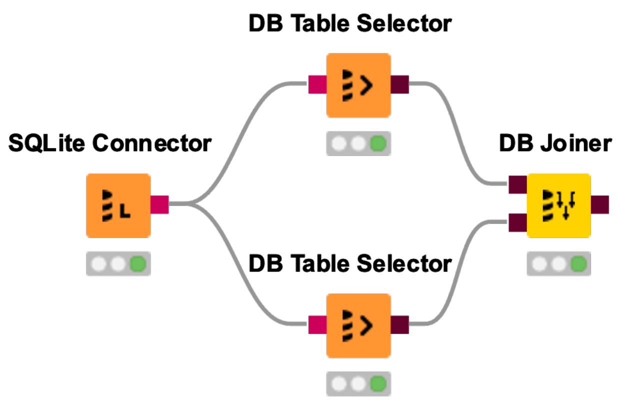
Advanced settings

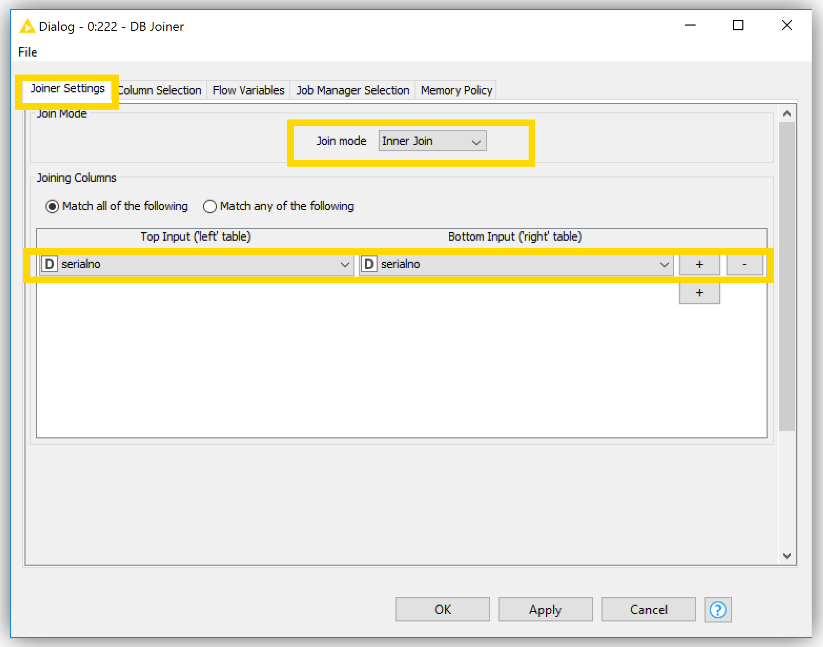
Column naming: Aggregation method (column name) [✓] Add COUNT(*) column name: COUNT(*)

PostgreSQL: 25 aggregation functions

DB Joiner

- Combines columns from 2 different tables
- Top port contains “Left” data table
- Bottom port contains the “Right” data table




Dialog - 0:222 - DB Joiner

File

Joiner Settings | Column Selection | Flow Variables | Job Manager Selection | Memory Policy

Join Mode

Join mode: Inner Join

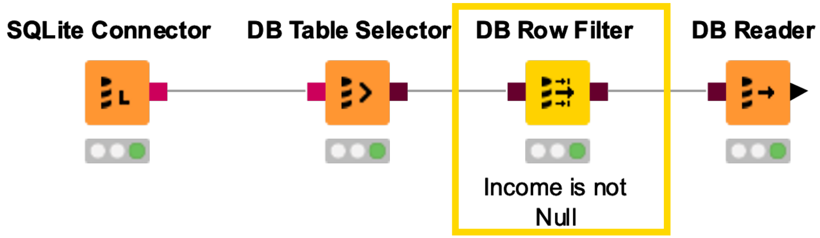
Joining Columns

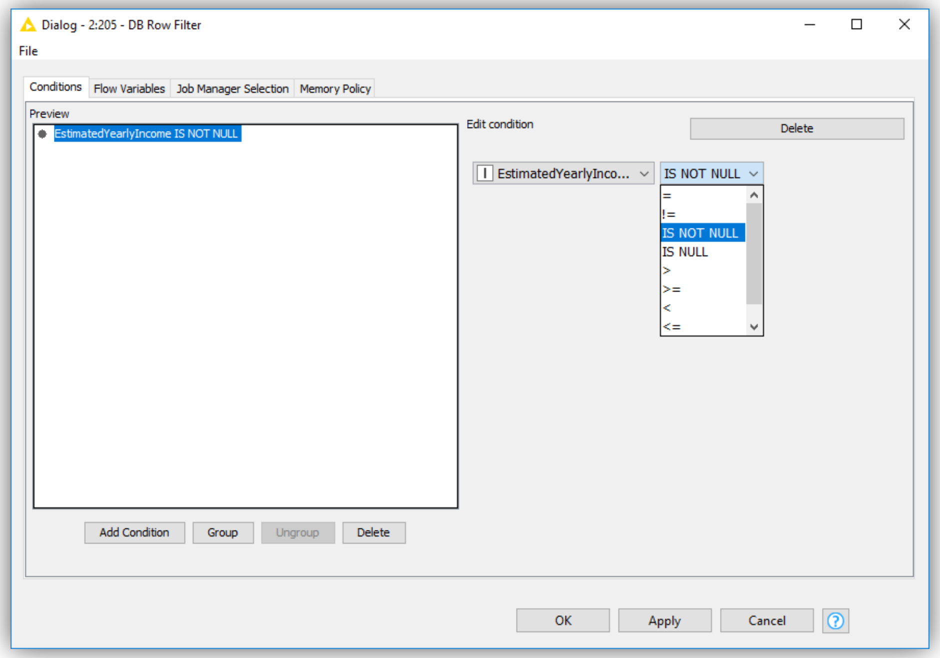
(● Match all of the following) ○ Match any of the following

Top Input ('left' table)	Bottom Input ('right' table)
D serialno	D serialno

DB Row Filter

- Filters rows that do not match the filter criteria
- Use the *IS NULL* or *IS NOT NULL* operator to filter missing values




Dialog - 2:205 - DB Row Filter

File

Conditions | Flow Variables | Job Manager Selection | Memory Policy

Preview

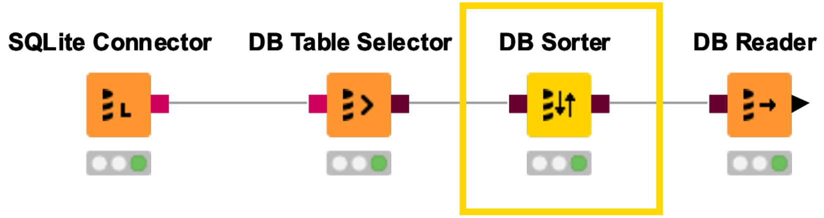
EstimatedYearlyIncome IS NOT NULL

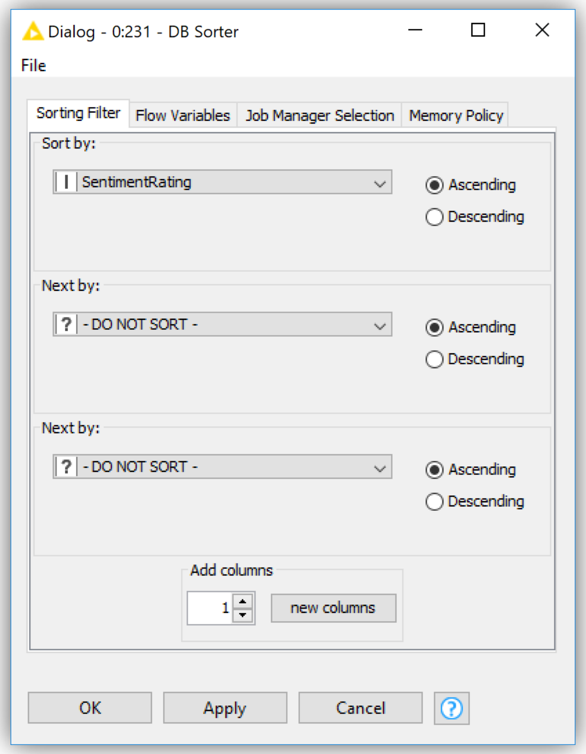
Edit condition

EstimatedYearlyInco... IS NOT NULL

DB Sorter

- Sorts the input data by one or multiple columns





Sort by: SentimentRating [✓ Ascending] Descending

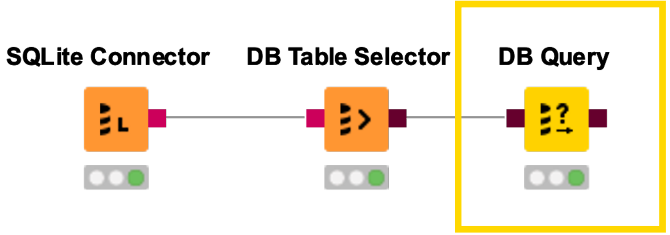
Next by: - DO NOT SORT - [✓ Ascending] Descending

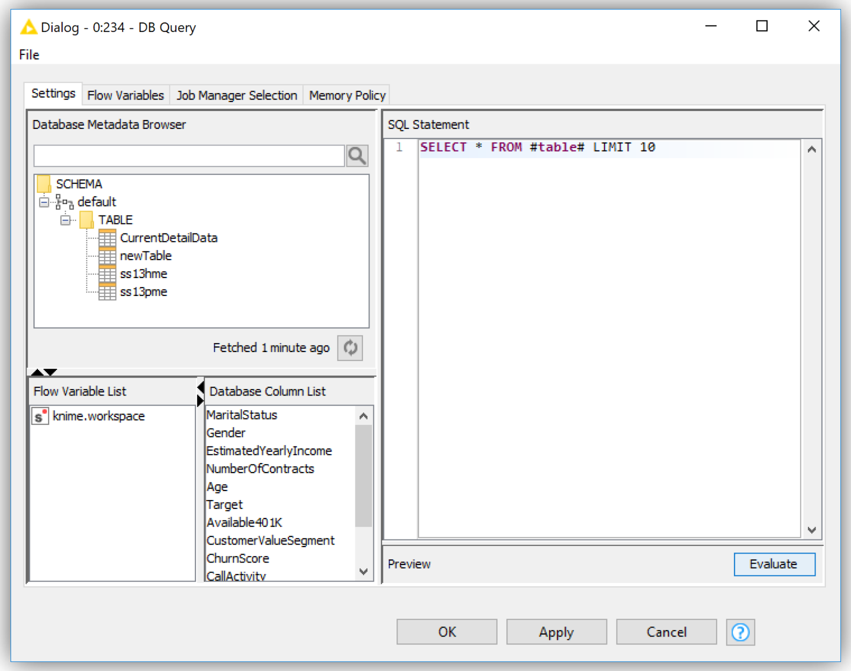
Next by: - DO NOT SORT - [✓ Ascending] Descending

Add columns: 1 new columns

DB Query

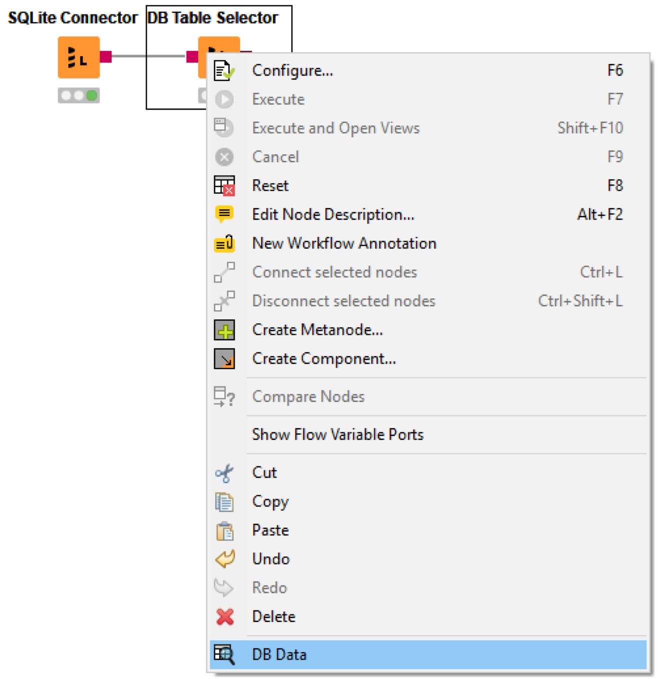
- Executes arbitrary SQL queries
- #table# is replaced with input query

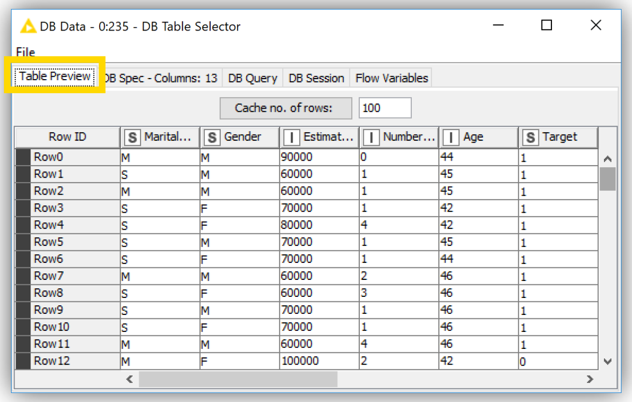




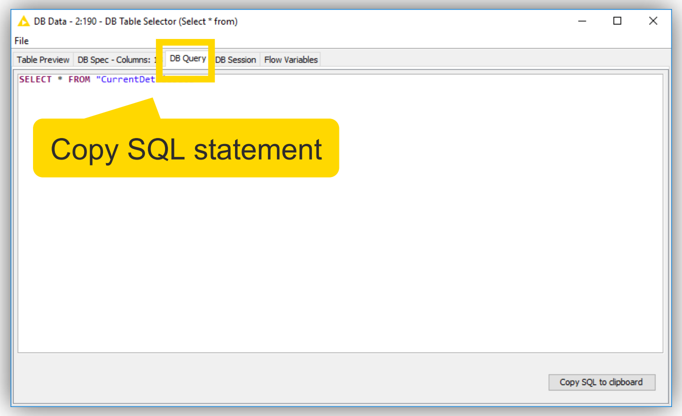
```
SELECT * FROM #table# LIMIT 10
```

Database Connection Port View





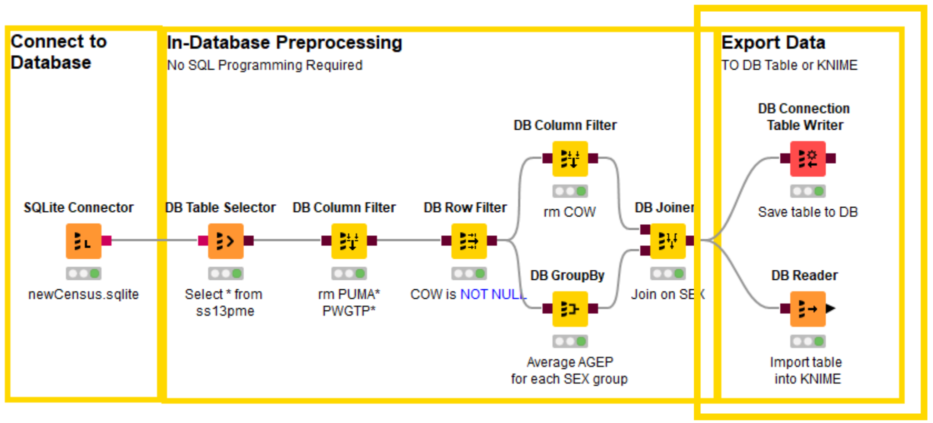
Row ID	S Marital...	S Gender	I Estimat...	I Number...	I Age	S Target
Row0	M	M	90000	0	44	1
Row1	S	M	60000	1	45	1
Row2	M	M	60000	1	45	1
Row3	S	F	70000	1	42	1
Row4	S	F	80000	4	42	1
Row5	S	M	70000	1	45	1
Row6	S	F	70000	1	44	1
Row7	M	M	60000	2	46	1
Row8	S	F	60000	3	46	1
Row9	S	M	70000	1	46	1
Row10	S	F	70000	1	46	1
Row11	M	M	60000	4	46	1
Row12	M	F	100000	2	42	0

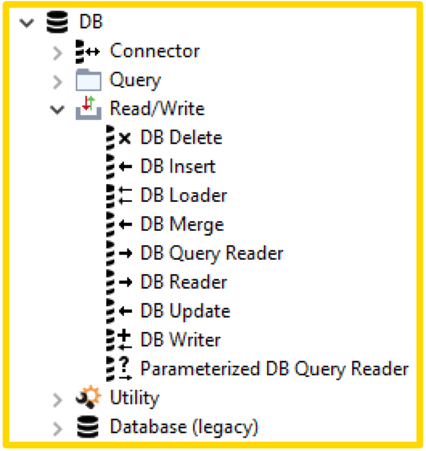


```
SELECT * FROM "CurrentDet
```

Export Data

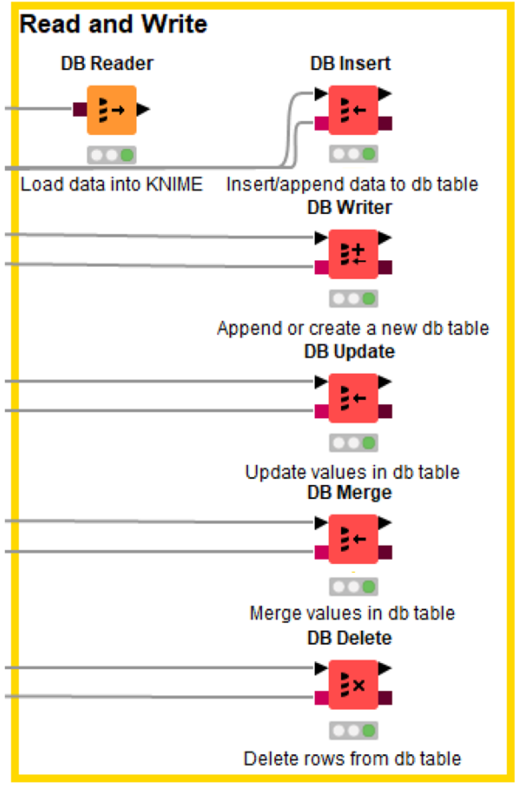
- Writing data back into database
- Exporting data into KNIME
- SQL operations are **executed on the database!**





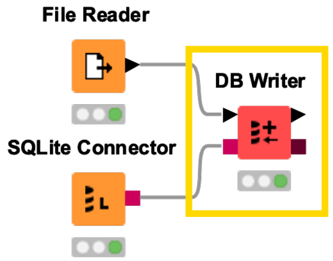
Database Writing Nodes

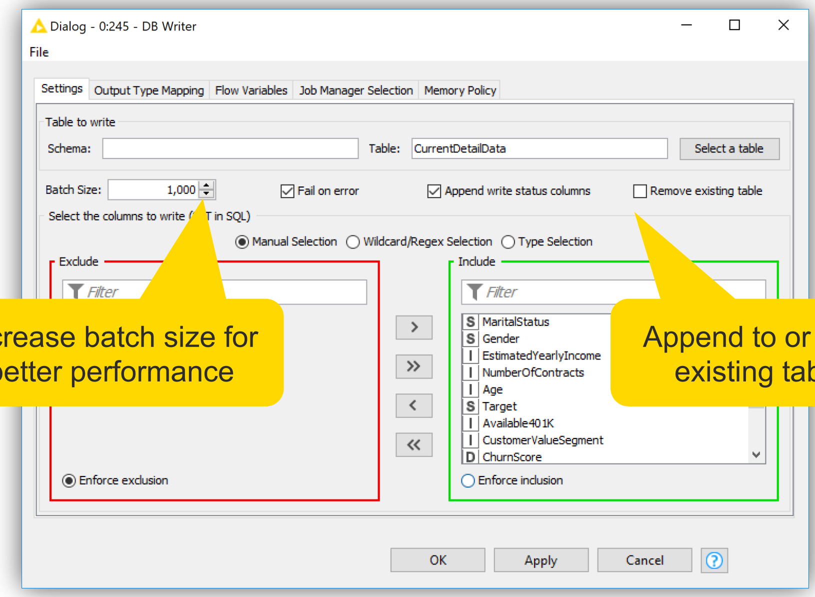
- Create table as select
- Insert/append/merge data
- Update values in table
- Delete rows from table


Read and Write
DB Reader
Load data into KNIME
DB Insert
Insert/append data to db table
DB Writer
Append or create a new db table
DB Update
Update values in db table
DB Merge
Merge values in db table
DB Delete
Delete rows from db table

DB Writer

- Writes data from a KNIME data table **directly** into a database table




Dialog - 0:245 - DB Writer

File

Settings | Output Type Mapping | Flow Variables | Job Manager Selection | Memory Policy

Table to write

Schema: | Table: CurrentDetailData | Select a table

Batch Size: 1,000 | [✓] Fail on error | [✓] Append write status columns | [ ] Remove existing table

Select the columns to write (…T in SQL)

(●) Manual Selection ( ) Wildcard/Regex Selection ( ) Type Selection

Exclude: Filter — (●) Enforce exclusion

Include: Filter

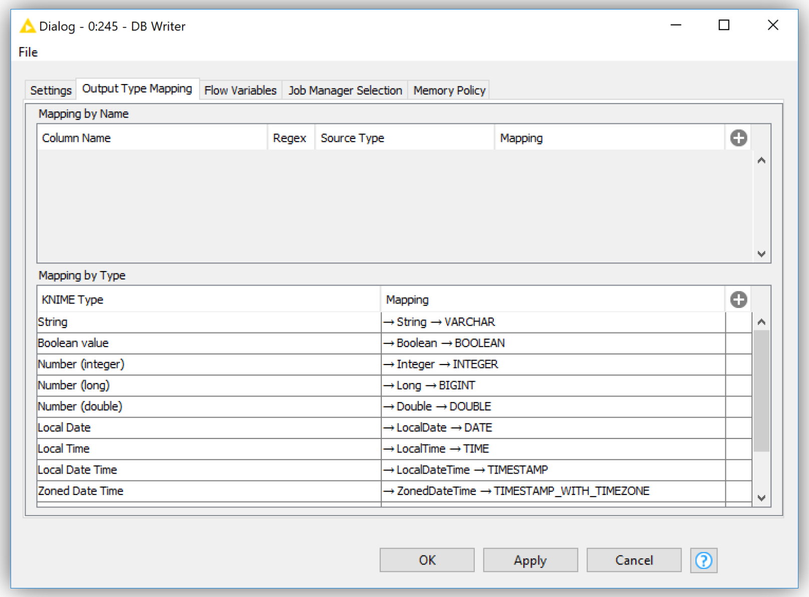
- S MaritalStatus
- S Gender
- I EstimatedYearlyIncome
- I NumberOfContracts
- I Age
- S Target
- I Available401K
- I CustomerValueSegment
- D ChurnScore

( ) Enforce inclusion

OK | Apply | Cancel

Increase batch size for better performance

Append to or drop existing table


Dialog - 0:245 - DB Writer

File

Settings | Output Type Mapping | Flow Variables | Job Manager Selection | Memory Policy

Mapping by Name

Column Name	Regex	Source Type	Mapping

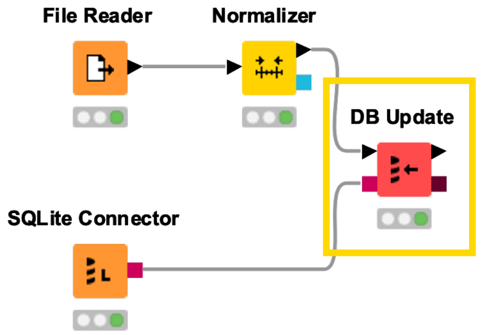
Mapping by Type

KNIME Type	Mapping
String	→ String → VARCHAR
Boolean value	→ Boolean → BOOLEAN
Number (integer)	→ Integer → INTEGER
Number (long)	→ Long → BIGINT
Number (double)	→ Double → DOUBLE
Local Date	→ LocalDate → DATE
Local Time	→ LocalTime → TIME
Local Date Time	→ LocalDateTime → TIMESTAMP
Zoned Date Time	→ ZonedDateTime → TIMESTAMP_WITH_TIMEZONE

OK | Apply | Cancel

DB Update

- Updates all database records that match the update criteria



Increase batch size for better performance

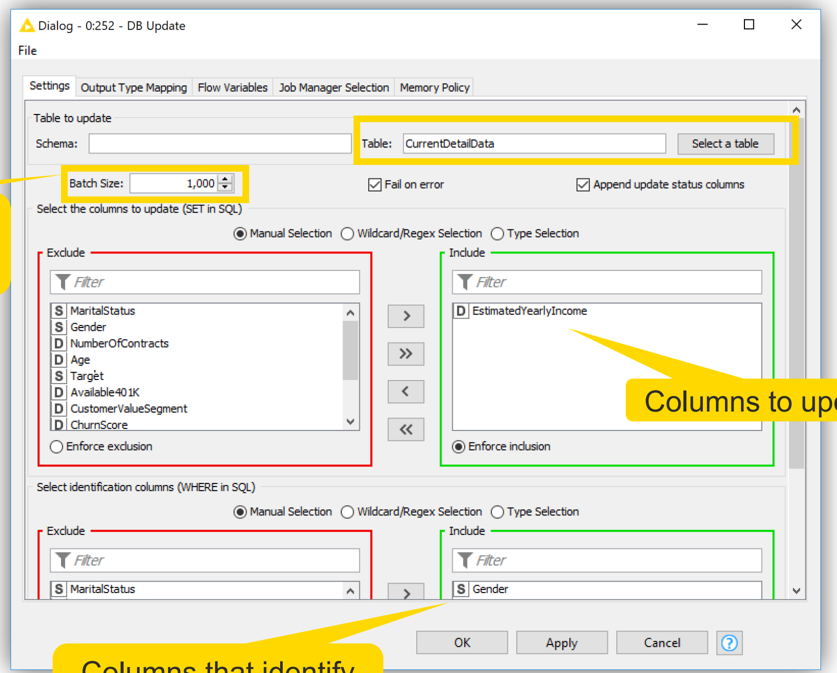


Table to update

Schema:

Table: CurrentDetailData

Batch Size: 1,000

[✓] Fail on error

[✓] Append update status columns

Select the columns to update (SET in SQL)

[✓ Manual Selection] Wildcard/Regex Selection Type Selection

Exclude: S MaritalStatus, S Gender, D NumberOfContracts, D Age, S Target, D Available401K, D CustomerValueSegment, D ChurnScore

Enforce exclusion

Include: D EstimatedYearlyIncome

[✓ Enforce inclusion]

Select identification columns (WHERE in SQL)

[✓ Manual Selection] Wildcard/Regex Selection Type Selection

Exclude: S MaritalStatus

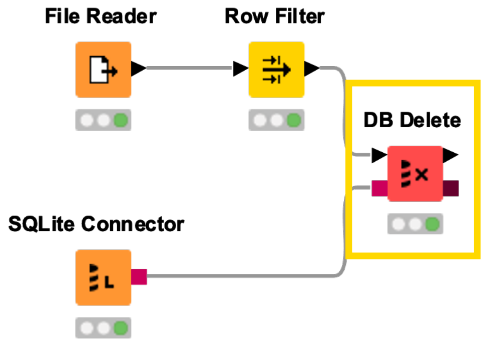
Include: S Gender

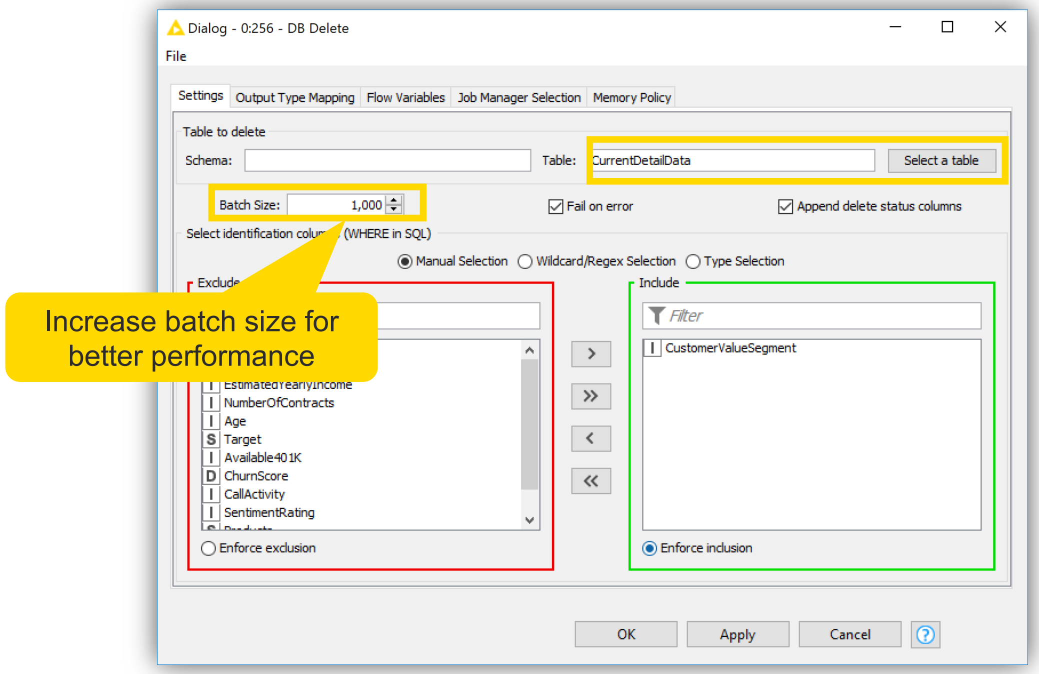
Columns to update

Columns that identify the records to update

DB Delete

- Deletes all database records that match the values of the selected columns





Dialog - 0:256 - DB Delete

File

Settings | Output Type Mapping | Flow Variables | Job Manager Selection | Memory Policy

Table to delete

Schema:

Table: CurrentDetailData

Select a table

Batch Size: 1,000

[✓] Fail on error

[✓] Append delete status columns

Select identification columns (WHERE in SQL)

[✓ Manual Selection] Wildcard/Regex Selection Type Selection

Exclude: EstimatedYearlyIncome, NumberOfContracts, Age, Target, Available401K, ChurnScore, CallActivity, SentimentRating, Products

Enforce exclusion

Include: CustomerValueSegment

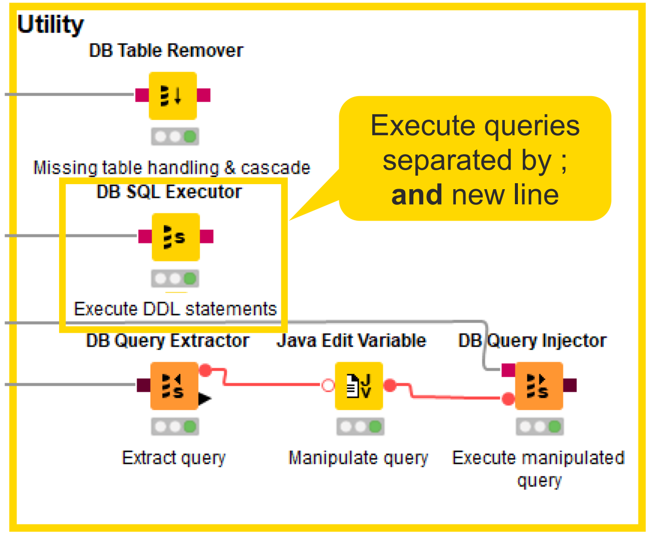
[✓ Enforce inclusion]

OK | Apply | Cancel

Increase batch size for better performance

Utility

- Drop table
  - missing table handling
  - cascade option
- Execute any SQL statement e.g. DDL
- Manipulate existing queries



Database Exercise

- Connect to the *database.mv.db* database with the H2 Connector node
- Write the Fully Joined Data into the database as a new table called "adult"
- Select the "adult" table in the database
- Count the number of records per product
- Filter out products that occur less than 1000 times by joining the aggregated and the original database table
- Read the filtered database table into a KNIME data table

Thank You!

education@knime.com