Lab 6 – Data Cleaning using KNIME

Table of Contents

Learning Outcome

At the end of this practical, you will be able to:

- Perform data cleaning to resolve common data quality issues such as duplicate data, missing data, inconsistency formatting, etc prior to data analytics.

Introduction

This practice covers the steps on how to carry out an ETL process using KNIME by extracting data from an excel file, transforming and preparing it for the next step (Modeling) in the CRISP-DM model.

A summary of the steps involved in the following exercises:

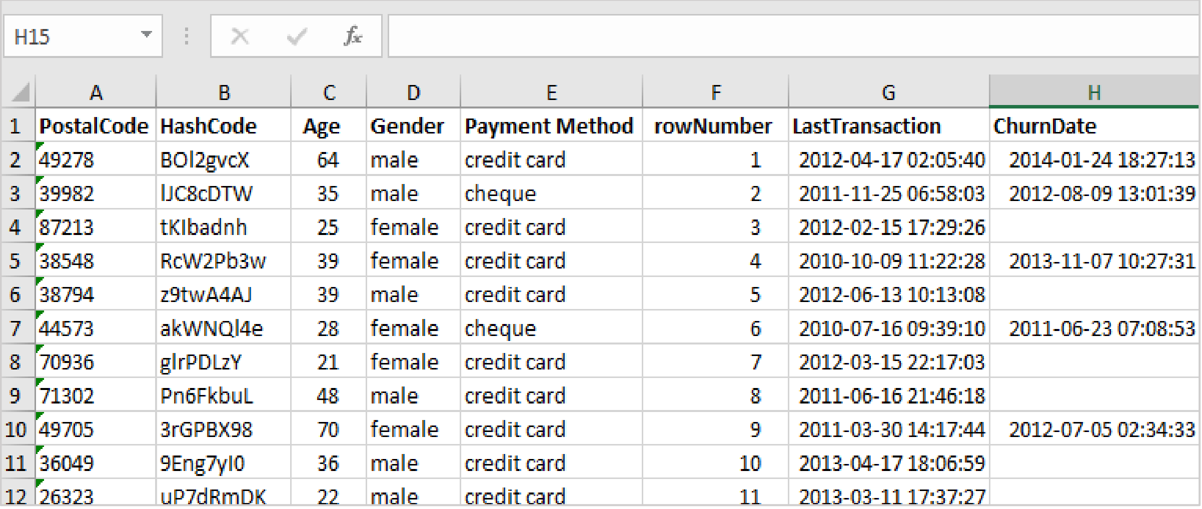
1. Import data from files/sources to KNIME Analytics Platform.
2. Identify data quality issues.
3. Carry out data preparation (transformation and cleaning).

Learning Resources: https://www.KNIME.com/learning-hub

Understanding the Data

Given the business scenario a company that sells mobile apps and they wish to retain existing customers, let's carry out Data Understanding which is the step before Data Preparation in the CRISP-DM model.

Download a copy of the **Customer Data.xlsx** from NYP LMS. Open in Microsoft Excel.



H15

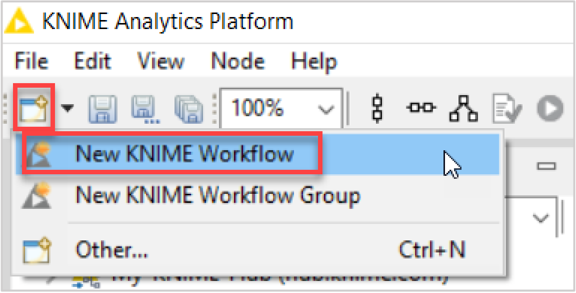
	A	B	C	D	E	F	G	H
1	PostalCode	HashCode	Age	Gender	Payment Method	rowNumber	LastTransaction	ChurnDate
2	49278	BOl2gvcX	64	male	credit card	1	2012-04-17 02:05:40	2014-01-24 18:27:13
3	39982	lJC8cDTW	35	male	cheque	2	2011-11-25 06:58:03	2012-08-09 13:01:39
4	87213	tKIbadnh	25	female	credit card	3	2012-02-15 17:29:26	
5	38548	RcW2Pb3w	39	female	credit card	4	2010-10-09 11:22:28	2013-11-07 10:27:31
6	38794	z9twA4AJ	39	male	credit card	5	2012-06-13 10:13:08	
7	44573	akWNQl4e	28	female	cheque	6	2010-07-16 09:39:10	2011-06-23 07:08:53
8	70936	glrPDLzY	21	female	credit card	7	2012-03-15 22:17:03	
9	71302	Pn6FkbuL	48	male	credit card	8	2011-06-16 21:46:18	
10	49705	3rGPBX98	70	female	credit card	9	2011-03-30 14:17:44	2012-07-05 02:34:33
11	36049	9Eng7yl0	36	male	credit card	10	2013-04-17 18:06:59	
12	26323	uP7dRmDK	22	male	credit card	11	2013-03-11 17:37:27	

The Customer Data.xlsx file has 8 columns and 1000 lines of data. Each row contains information about a single customer. There is an Age, a Gender, and a PostalCode column. These three columns contain basic mainly unchangeable information on each customer. The rest of the data contains information about the buying behaviour of the customers and other information.

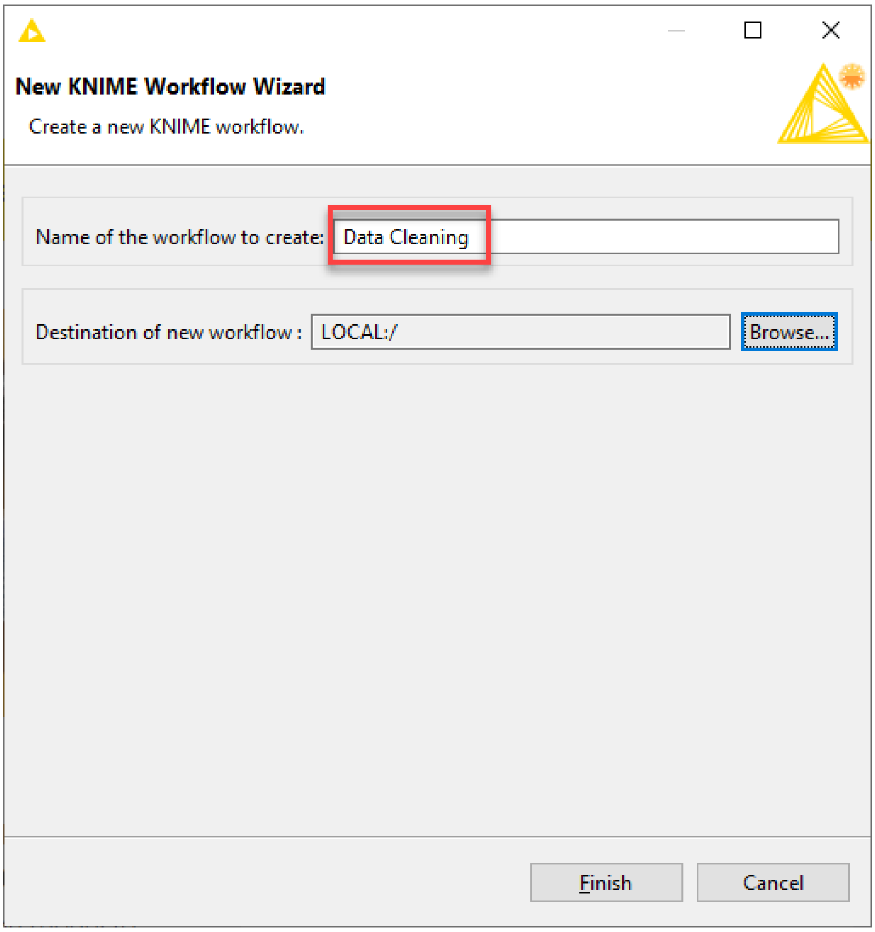
It is easy to understand data within Excel by opening and viewing it. But we want more than to just looking at the data. We need to analyse it in KNIME to get a detailed understanding in order to identify any data quality issues. So first we need to import the data into KNIME.

Task 1: Import Data

1. Let's start by creating a new workflow. Close the Welcome Page. Click on **New** in the toolbar panel at the top of the KNIME Workbench. Select **New KNIME Workflow**.



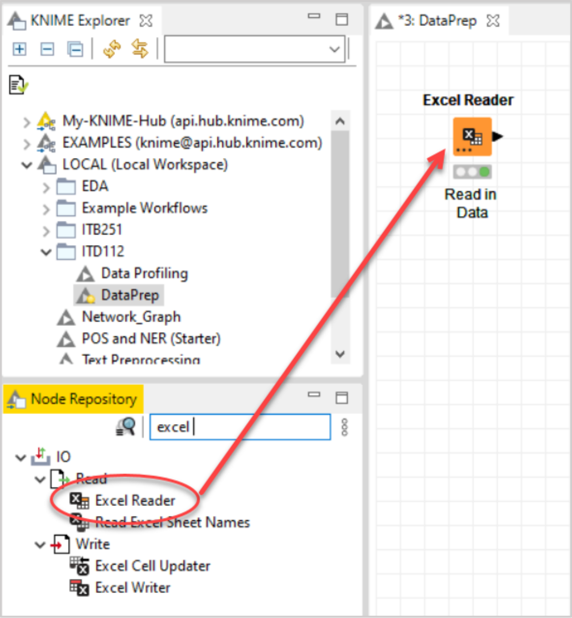
2. Name the workflow as **Data Cleaning** and click **Finish**.



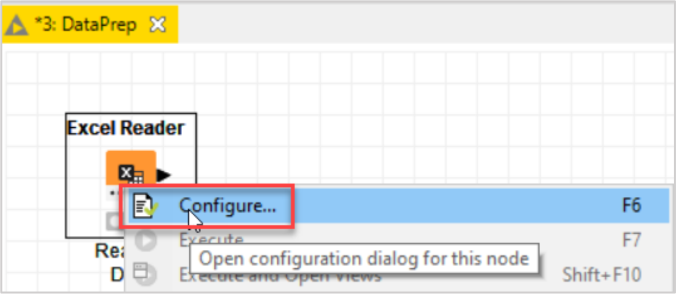
Name of the workflow to create: Data Cleaning

Destination of new workflow : LOCAL:/

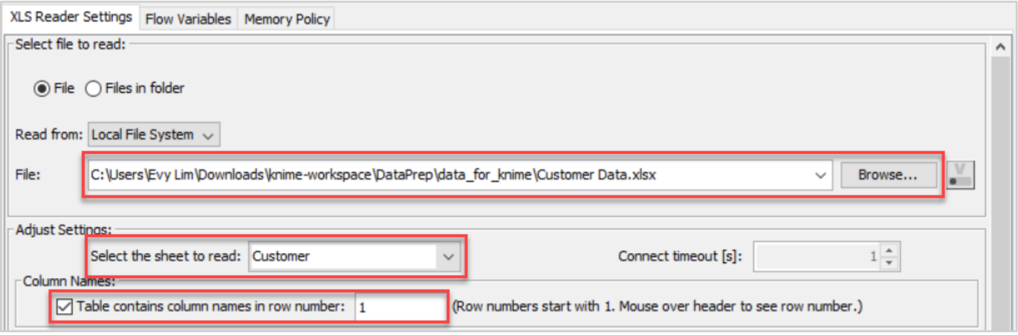
3. In the **Node Repository**, search for the **Excel Reader** node. Drag and drop it to the Workflow Editor. Or double click on the Excel Reader node in the node repository, it automatically appears in the workflow editor.



4. We need to do some configuration to read in the data files. Double click on the **Excel Reader** node or right click it and select **Configure**.


*3: DataPrep
Excel Reader
Configure... F6
Execute F7
Open configuration dialog for this node
Execute and Open Views Shift+F10

5. In the Configuration dialog, define the file path by clicking the **Browse** button. Select the **Customer Data.xlsx** excel file you have downloaded. Select the **Customer** worksheet to read and check the **Table contains column name in row number: 1**.



Select file to read:

● File ○ Files in folder

Read from: Local File System

File: C:\Users\Evy Lim\Downloads\knime-workspace\DataPrep\data_for_knime\Customer Data.xlsx

Adjust Settings:

Select the sheet to read: Customer

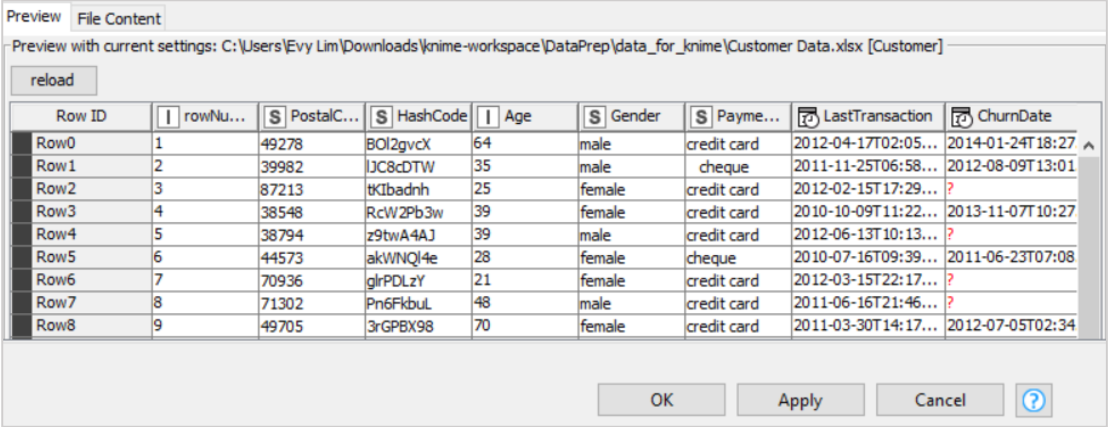
Connect timeout [s]: 1

Column Names:

☑ Table contains column names in row number: 1 (Row numbers start with 1. Mouse over header to see row number.)

Note: If there are no headers in the data, the columns will then get generic names like Col1, Col2, etc. This is automatically assigned by KNIME.

6. In the Preview tab, refresh to check the data is correct and click **OK**.



Preview with current settings: C:\Users\Evy Lim\Downloads\knime-workspace\DataPrep\data_for_knime\Customer Data.xlsx [Customer]

Row ID	I rowNu...	S PostalC...	S HashCode	I Age	S Gender	S Payme...	LastTransaction	ChurnDate
Row0	1	49278	BOl2gvcX	64	male	credit card	2012-04-17T02:05...	2014-01-24T18:27
Row1	2	39982	lJC8cDTW	35	male	cheque	2011-11-25T06:58...	2012-08-09T13:01
Row2	3	87213	tKIbadnh	25	female	credit card	2012-02-15T17:29...	?
Row3	4	38548	RcW2Pb3w	39	female	credit card	2010-10-09T11:22...	2013-11-07T10:27
Row4	5	38794	z9twA4AJ	39	male	credit card	2012-06-13T10:13...	?
Row5	6	44573	akWNQl4e	28	female	cheque	2010-07-16T09:39...	2011-06-23T07:08
Row6	7	70936	glrPDLzY	21	female	credit card	2012-03-15T22:17...	?
Row7	8	71302	Pn6FkbuL	48	male	credit card	2011-06-16T21:46...	?
Row8	9	49705	3rGPBX98	70	female	credit card	2011-03-30T14:17...	2012-07-05T02:34

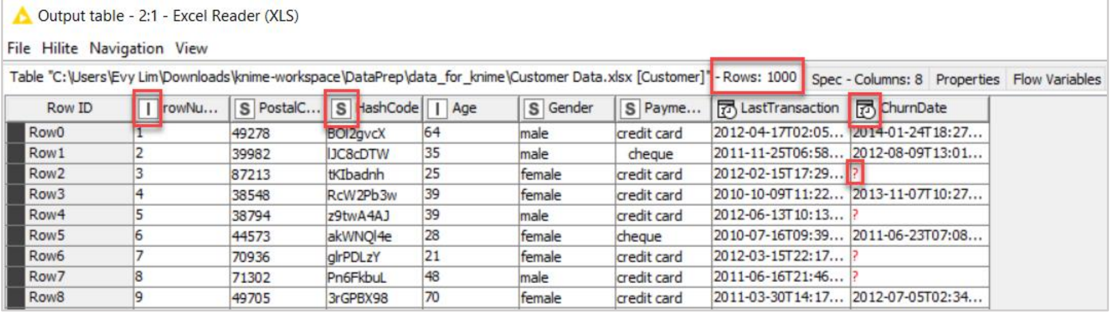
7. Save your workflow.

Task 2: Data Exploration

1. You may want to inspect the output table to see if the data file was read as intended. Right click on the **Excel Reader** node and select **Execute**.

2. After node status changes to executed ○○●, open the output table by right click on the **Excel Reader** node and select **Output table**.

   The output table of the Excel Reader node only shows the same data that we saw in excel but with the following additional information reported by KNIME.



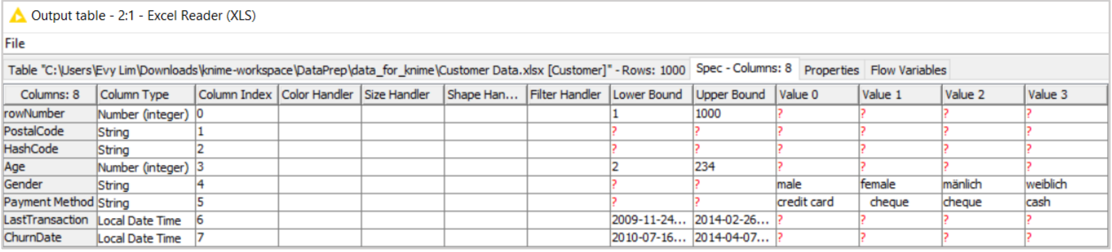
Output table - 2:1 - Excel Reader (XLS)

Table "C:\Users\Evy Lim\Downloads\knime-workspace\DataPrep\data_for_knime\Customer Data.xlsx [Customer]" - Rows: 1000

Row ID	I rowNu...	S PostalC...	S HashCode	I Age	S Gender	S Payme...	LastTransaction	ChurnDate
Row0	1	49278	BOl2gvcX	64	male	credit card	2012-04-17T02:05...	2014-01-24T18:27...
Row1	2	39982	lJC8cDTW	35	male	cheque	2011-11-25T06:58...	2012-08-09T13:01...
Row2	3	87213	tKIbadnh	25	female	credit card	2012-02-15T17:29...	?
Row3	4	38548	RcW2Pb3w	39	female	credit card	2010-10-09T11:22...	2013-11-07T10:27...
Row4	5	38794	z9twA4AJ	39	male	credit card	2012-06-13T10:13...	?
Row5	6	44573	akWNQl4e	28	female	cheque	2010-07-16T09:39...	2011-06-23T07:08...
Row6	7	70936	glrPDLzY	21	female	credit card	2012-03-15T22:17...	?
Row7	8	71302	Pn6FkbuL	48	male	credit card	2011-06-16T21:46...	?
Row8	9	49705	3rGPBX98	70	female	credit card	2011-03-30T14:17...	2012-07-05T02:34...

   a) The number of rows and attributes (columns).
   b) Default types of the various attributes.
   c) There are a lot of question marks (?) in the DateChurn attribute. A question mark indicates that KNIME does not have a value for this cell, it is missing. In the case of DateChurn that is because there are no values for this attribute in the input data.

3. Click on the **Spec - Columns** tab.



Output table - 2:1 - Excel Reader (XLS)

File

Table "C:\Users\Evy Lim\Downloads\knime-workspace\DataPrep\data_for_knime\Customer Data.xlsx [Customer]" - Rows: 1000 | Spec - Columns: 8 | Properties | Flow Variables

Columns: 8	Column Type	Column Index	Color Handler	Size Handler	Shape Han...	Filter Handler	Lower Bound	Upper Bound	Value 0	Value 1	Value 2	Value 3
rowNumber	Number (integer)	0					1	1000	?	?	?	?
PostalCode	String	1					?	?	?	?	?	?
HashCode	String	2					?	?	?	?	?	?
Age	Number (integer)	3					2	234	?	?	?	?
Gender	String	4					?	?	male	female	mänlich	weiblich
Payment Method	String	5					?	?	credit card	cheque	cheque	cash
LastTransaction	Local Date Time	6					2009-11-24...	2014-02-26...	?	?	?	?
ChurnDate	Local Date Time	7					2010-07-16...	2014-04-07...	?	?	?	?

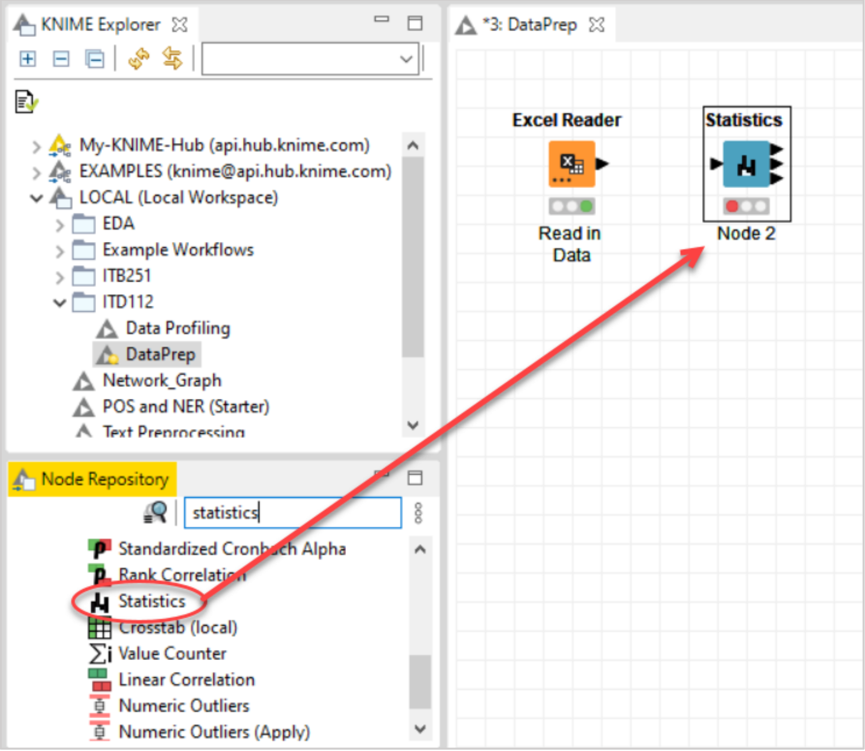
In the **Spec - Columns** tab, the list all attributes - every column from the Table view is now summarized in its own row. The left-most column displays the attribute name, the next column lists the data type

The right half of the view displays various statistics (depending on the data type):

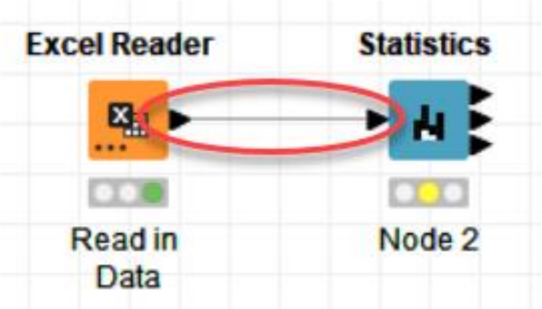
- The lower and upper bound of the attribute
- The list of all the value in the attribute (eg. Gender has 4 different values)

You may have questions like "What is the range of a numeric attribute?", "Which values does a nominal attribute have?" etc. Such information is easily accessible via the **Statistics** node in KNIME to explore the distribution of the data.

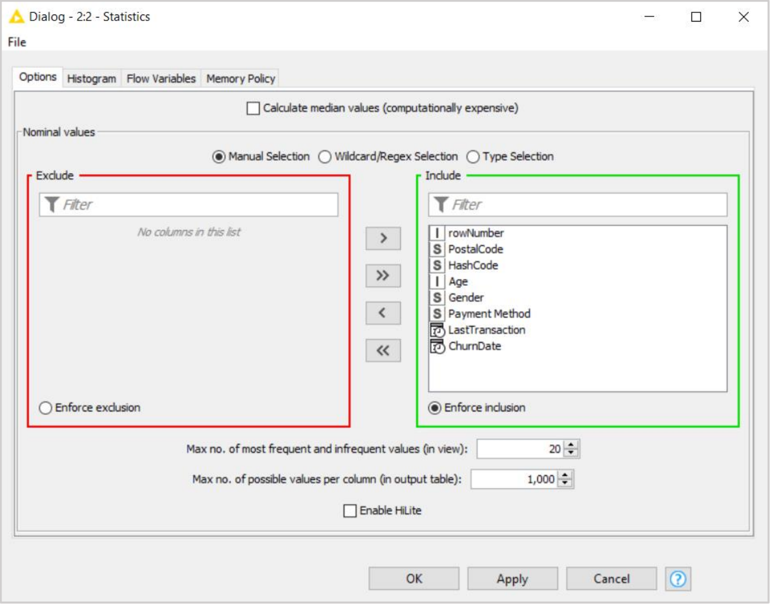
4. In the Node Repository, search for the **Statistics** node. Drag and place it beside the Excel Reader node.



5. Connect the output of the **Excel Reader** node to the **Statistics** node.



6. Right click on the **Statistics** node, select **Configure**. Check that all the attributes are included and click **OK**.



Dialog - 2:2 - Statistics

File

Options | Histogram | Flow Variables | Memory Policy

☐ Calculate median values (computationally expensive)

Nominal values

● Manual Selection ○ Wildcard/Regex Selection ○ Type Selection

Exclude

Filter

No columns in this list

○ Enforce exclusion

Include

Filter

- I rowNumber
- S PostalCode
- S HashCode
- I Age
- S Gender
- S Payment Method
- LastTransaction
- ChurnDate

● Enforce inclusion

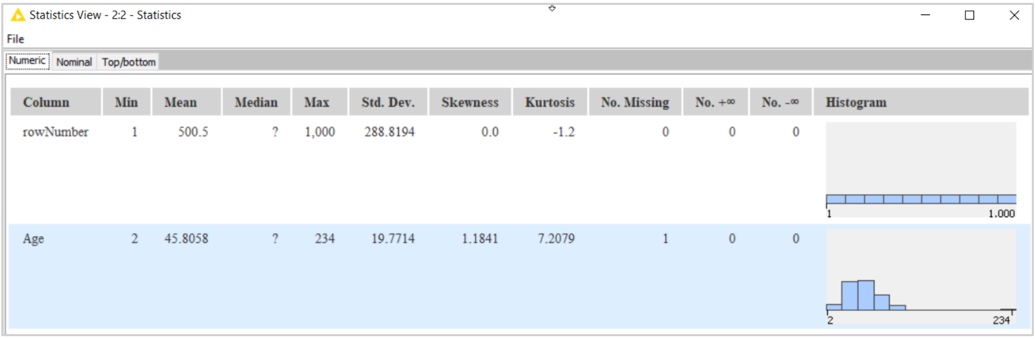
Max no. of most frequent and infrequent values (in view): 20

Max no. of possible values per column (in output table): 1,000

☐ Enable HiLite

7. Right click on the **Statistics** node, select **Execute and Open Views**.

8. Under Statistics View, you can click through the different tabs (Numeric, Nominal and Top/bottom) to view the different statistics.



Statistics View - 2:2 - Statistics

File

Numeric | Nominal | Top/bottom

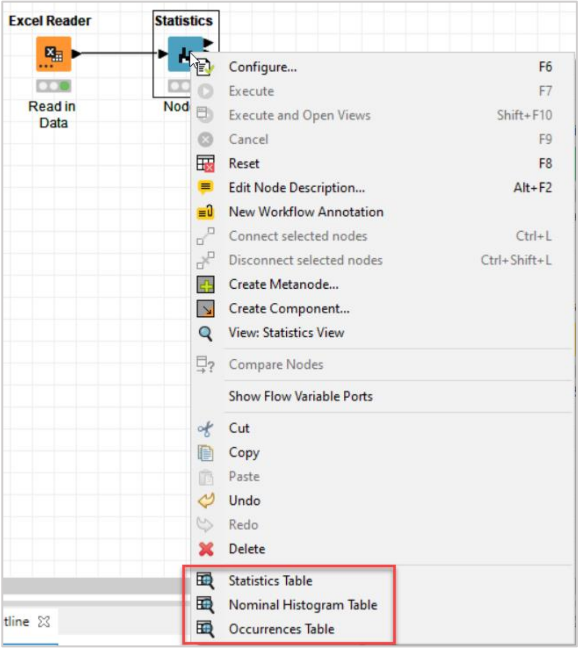
Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
rowNumber	1	500.5	?	1,000	288.8194	0.0	-1.2	0	0	0	1 1.000
Age	2	45.8058	?	234	19.7714	1.1841	7.2079	1	0	0	2 234

It shows the number of missing values, minimum, maximum, average, and standard deviation for numeric.

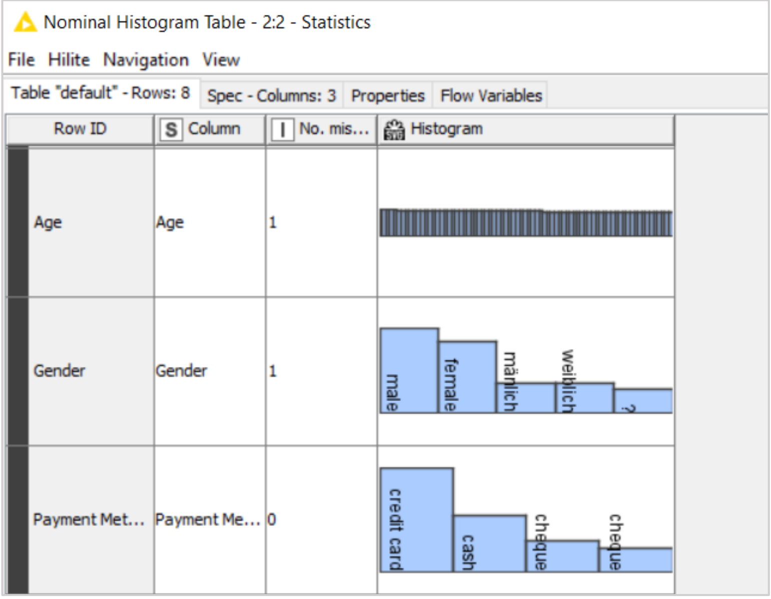
The Nominal tab displays the histogram for all the categorical columns.

The Top/bottom tab shows the most frequent/infrequent values from the categorical columns.

9. Right click on the Statistics node, you can view more statistical information using the **Statistics Table**, **Nominal Histogram Table** and **Occurrences Table**.



Take some time to analyse the distribution and note down any details of interest. See if you can come out with the same, or even more, data issues as given in the next section.



Nominal Histogram Table - 2:2 - Statistics

File Hilite Navigation View

Table "default" - Rows: 8 | Spec - Columns: 3 | Properties | Flow Variables

Row ID	Column	No. mis...	Histogram
Age	Age	1	
Gender	Gender	1	
Payment Met...	Payment Me...	0	

Save your workflow.

Task 3: Data Cleaning

Remember that in our business scenario, our customers are using an app to buy products. Based on their actions we want to predict who is likely to churn in the near future such that we can prevent that by giving special offers and other marketing actions tailored to this group of customers. If a customer has churned in the past then the ChurnDate tells you the date when he uninstalled the app. If he did not uninstall the app, it means that he is still an existing customer, and hence the churn date is missing. Using the appropriate algorithms, you can find relationships between attributes and the label. But before this can happen, we need to ensure that the input data is as clean as possible.

Data Inspection

Let’s leverage the statistics node and identify any data abnormality or quality issues.

a) Missing values

- Age and Gender contain one missing value each. For this case it means that we don’t have this information and one of the reasons could be the customer probably did not provide during sign-up.
- ChurnDate contains a lot of missing values. In this case, a missing value in ChurnDate actually has a meaning, and an important one which is, the respective customer did not churn.

Note: This shows the power of the Statistics, it is virtually impossible to spot one or two missing values in a data set with thousands of rows.

Basically, there are a few ways of dealing with missing values:

1. **Remove the attribute** if it contains too many missing values
2. **Introduce a new category value** for missing values in a nominal attribute e.g., unknown. While this may not add any value to us, it gives KNIME a chance to create rules for examples having an unknown category. For the computer, unknown means something very different from missing.
3. **Remove the data rows** with missing values of the attribute if there are only a few missing values because we would not be losing much information by removing them.

b) Data formatting issue

- There is formatting issue for payment method attribute which result in 4 different payment methods instead of three.

c) Data range

- Customer age should be somewhere between 16 and 110 years of age. Here we have customers as young as 2 years and as old as 234 years. Obviously, we have some wrong data here.

d) Gender

- There should be exactly two different genders, but there are actually four. In addition to male and female, there are gender values of mänlich and weiblich (genders in german). This may be due to one of our employees who comes from Germany. A consistent naming convention should be set in this case.

e) Irrelevant attribute

- HashCode attribute does not contain any valuable and/or interpretable information.

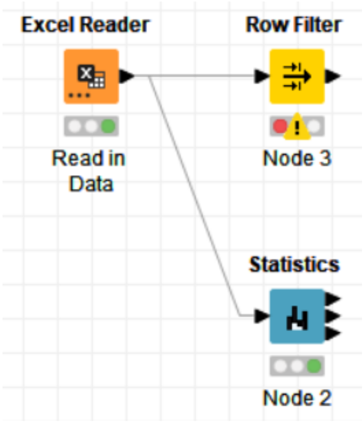
f) Unique identifer

- RowNumber uniquely identifies each customer which can confuse the modeling algorithms. We would need to ignore this during predictive analysis and that KNIME should treat this attribute as an ID.

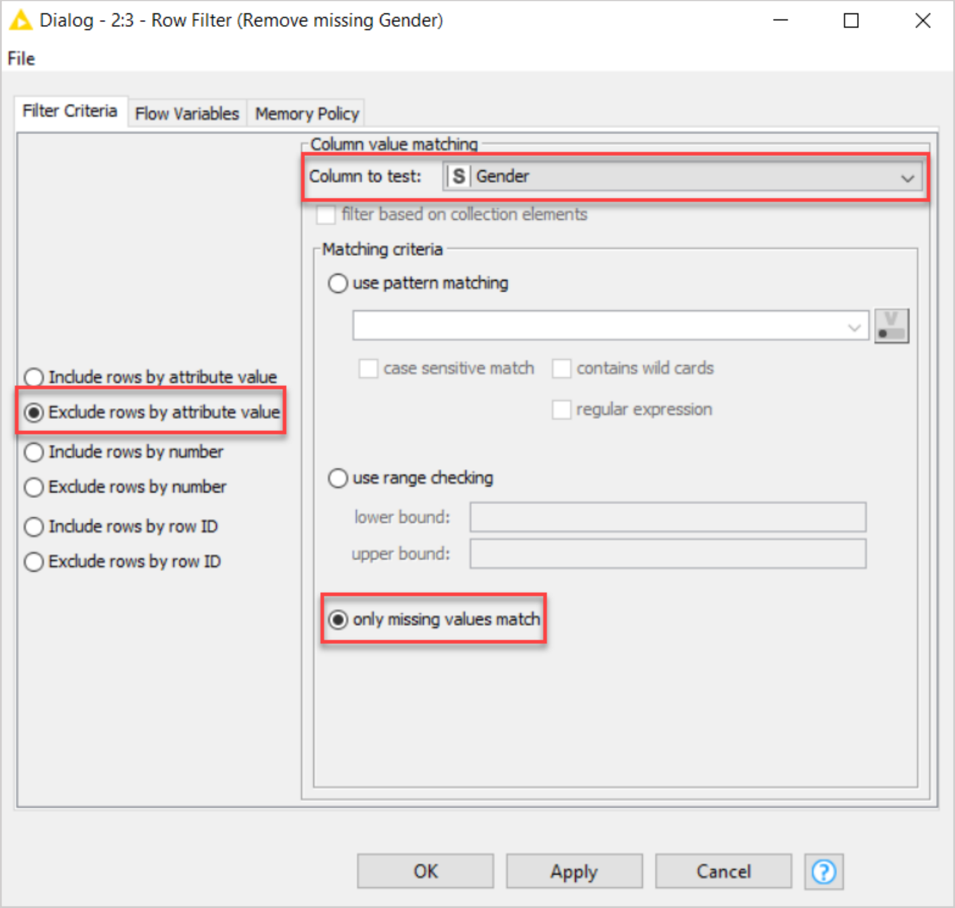
Filter Missing Values

In the following steps we will walk through the process of dealing with each of the data quality issues that we have just identified. Let's begin by creating a new process that will handle all the data preparation steps.

1. In the node repository, search for **Row Filter** node. Add the **Row Filter** node to the output of the **Excel Reader** node.



2. Open the configuration dialog of the Row Filter node and exclude rows from the input table where **Gender** has missing value.



Dialog - 2:3 - Row Filter (Remove missing Gender)

File

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: S Gender

□ filter based on collection elements

Matching criteria

○ use pattern matching

□ case sensitive match □ contains wild cards □ regular expression

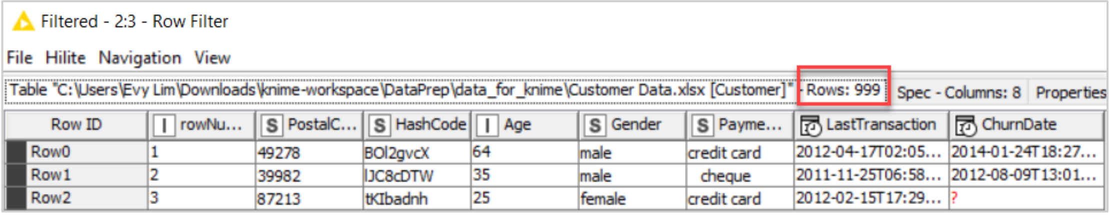
○ use range checking

lower bound:

upper bound:

● only missing values match

3. Right click on the **Row Filter** node and select **Execute**. Open the Filtered table by right click on the Row Filter node and select **Filtered**.



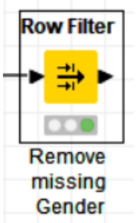
Filtered - 2:3 - Row Filter

File Hilite Navigation View

Table "C:\Users\Evy Lim\Downloads\knime-workspace\DataPrep\data_for_knime\Customer Data.xlsx [Customer]" - Rows: 999 | Spec - Columns: 8 | Properties

Row ID	I rowNu...	S PostalC...	S HashCode	I Age	S Gender	S Payme...	LastTransaction	ChurnDate
Row0	1	49278	BOl2gvcX	64	male	credit card	2012-04-17T02:05...	2014-01-24T18:27...
Row1	2	39982	IJC8cDTW	35	male	cheque	2011-11-25T06:58...	2012-08-09T13:01...
Row2	3	87213	tKIbadnh	25	female	credit card	2012-02-15T17:29...	?

The filtered table of the Row Filter node shows that there are now 999 rows instead of 1000 rows originally.

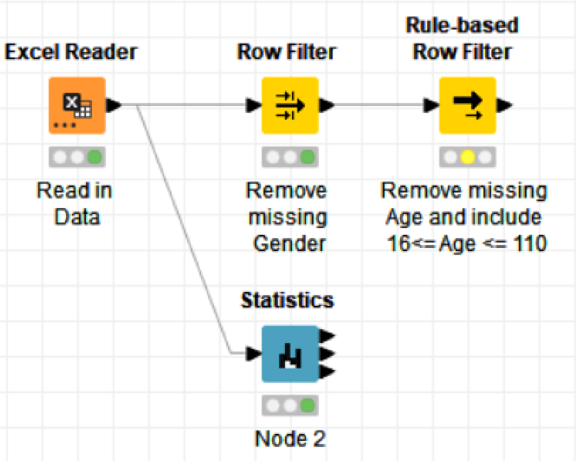


4. You can rename the node on the workflow so that it is easier for you to remember the node's function without checking its configuration.

Rule-based Row Filter for Age

There are some values of Age attribute that are out of range. A customer is 234 years old probably is due to a typo error. And there are other customers as young as 2 years old. In this case, they are invalid values (assuming our business terms and conditions only allow customers older than 16). We must get rid of those examples with an invalid Age attribute. We are going to limit the valid age range from 16 to 110 years old, and we also need to take care of a row with missing Age data. We can use multiple Row Filter nodes to resolve other data issues, but we can also use a single Rule-based Row Filter node to perform multiple operations at the same time.

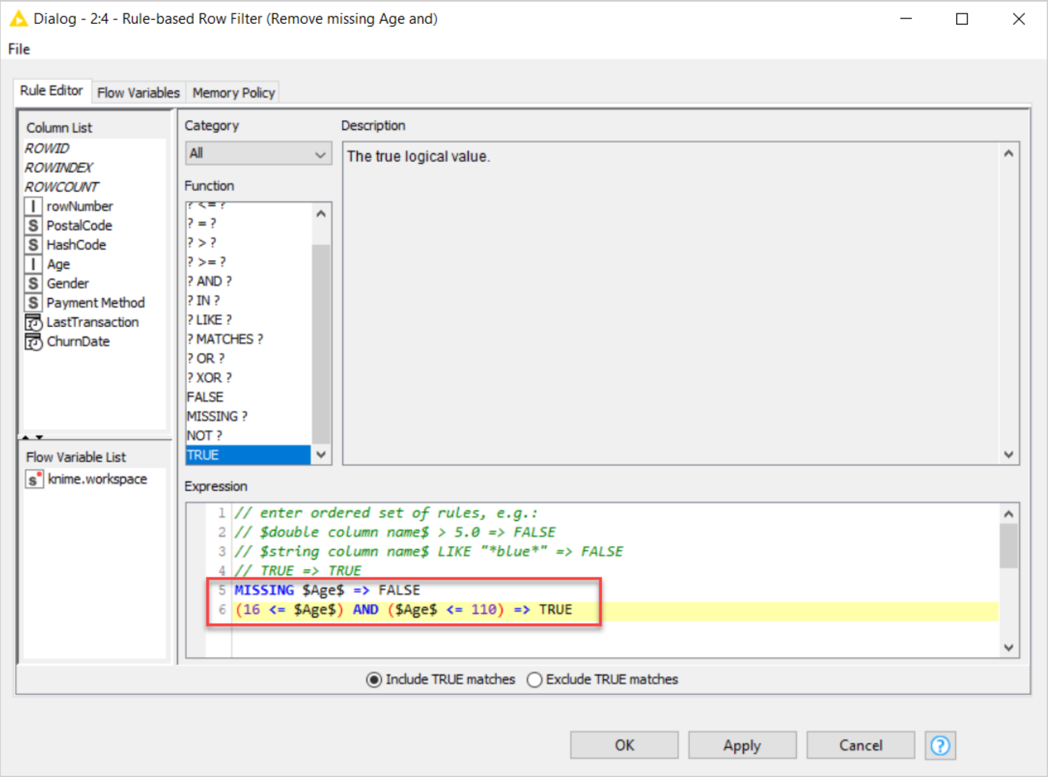
1. In the node repository, search for **Rule-based Row Filter** node. Add the **Rule-based Row Filter** node to the output of the **Row Filter** node and rename it as shown.



2. Double click the Rule-based Row Filter node to open the Rule Editor. To filter the missing Age column,
   a. Double click **MISSING ?** in Function
   b. Double click **Age** in the Column List
   c. Type **=>**
   d. Double click **FALSE** in Function

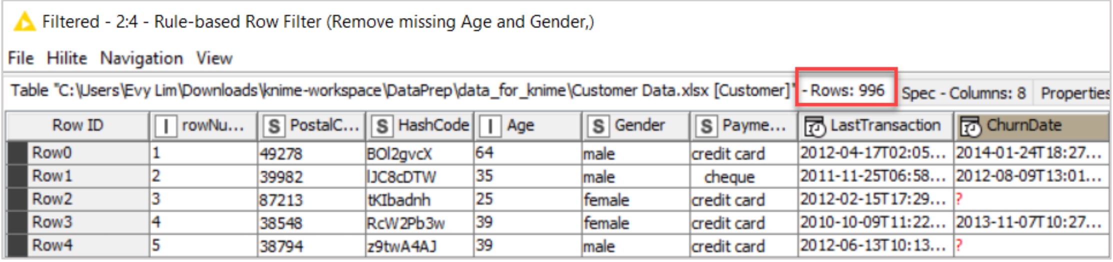
   To create this formula: **MISSING Age => FALSE**

   Complete the following rule to apply a reasonable range to the age column:
   (16 <= Age) AND (Age <=110) => TRUE



```
// enter ordered set of rules, e.g.:
// $double column name$ > 5.0 => FALSE
// $string column name$ LIKE "*blue*" => FALSE
// TRUE => TRUE
MISSING $Age$ => FALSE
(16 <= $Age$) AND ($Age$ <= 110) => TRUE
```

3. Execute and check the output. Your data set should now contain 996 examples and should have any missing values other than ChurnDate. You can use the Statistics node to do a check.



Filtered - 2:4 - Rule-based Row Filter (Remove missing Age and Gender,)

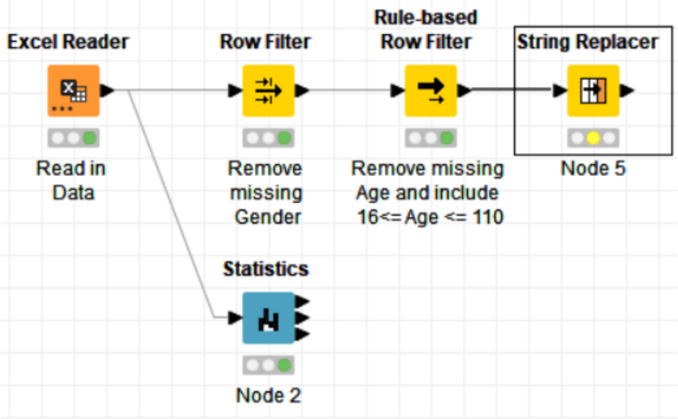
Table "C:\Users\Evy Lim\Downloads\knime-workspace\DataPrep\data_for_knime\Customer Data.xlsx [Customer]" - Rows: 996 | Spec - Columns: 8 | Properties

Row ID	I rowNu...	S PostalC...	S HashCode	I Age	S Gender	S Payme...	LastTransaction	ChurnDate
Row0	1	49278	BOl2gvcX	64	male	credit card	2012-04-17T02:05...	2014-01-24T18:27...
Row1	2	39982	lJC8cDTW	35	male	cheque	2011-11-25T06:58...	2012-08-09T13:01...
Row2	3	87213	tKIbadnh	25	female	credit card	2012-02-15T17:29...	?
Row3	4	38548	RcW2Pb3w	39	female	credit card	2010-10-09T11:22...	2013-11-07T10:27...
Row4	5	38794	z9twA4AJ	39	male	credit card	2012-06-13T10:13...	?

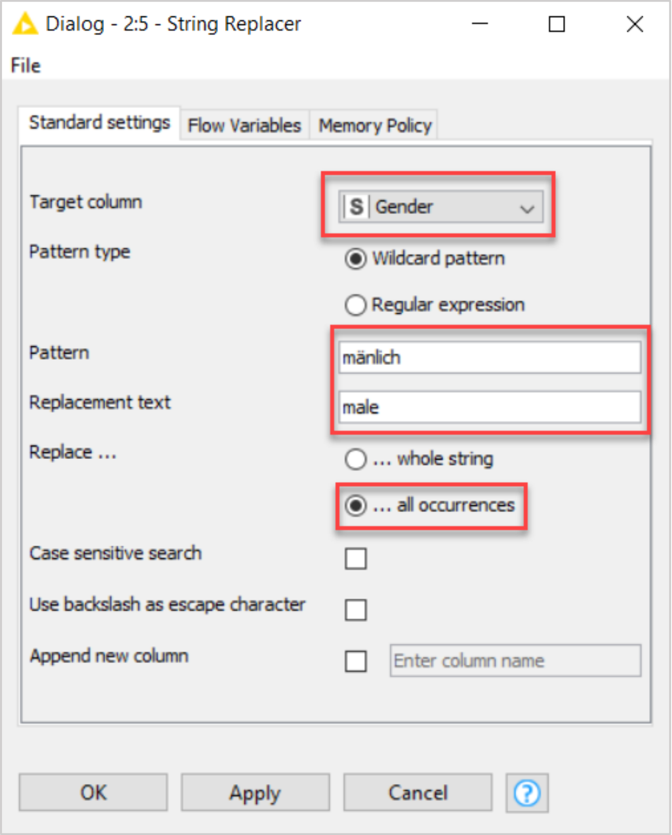
Replacing Invalid Genders

The Gender attribute also contains the German terms mänlich (for male) and weiblich (for female). This can seriously confuse the modeling algorithms as it will be assumed that there are four different terms for genders attribute. So, we have to correct this by replacing the German words with English gender words.

1. We can use **String Replacer** node to perform the replacement one-by-one. In the node repository, search for **String Replacer** node. Add the **String Replacer** node to the output of the Rule-based Row Filter node.

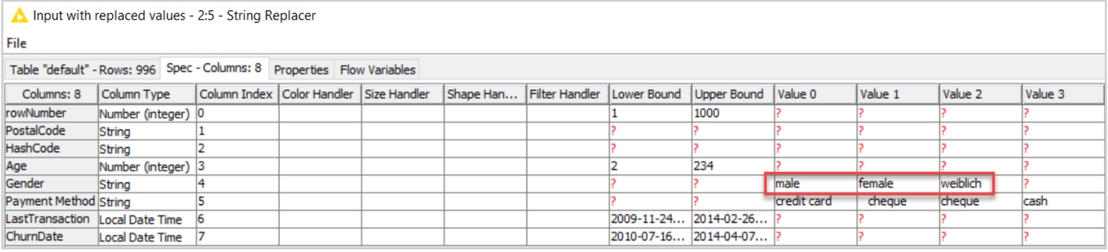


2. Open the configuration dialog of the **String Replacer** node and configure as such to replace the German word for male with the English counterpart.



Target column: S Gender
Pattern type: ● Wildcard pattern ○ Regular expression
Pattern: mänlich
Replacement text: male
Replace ...: ○ ... whole string ● ... all occurrences
Case sensitive search □
Use backslash as escape character □
Append new column □ Enter column name

3. Execute and check the output. You can see that mänlich has been replaced with male.



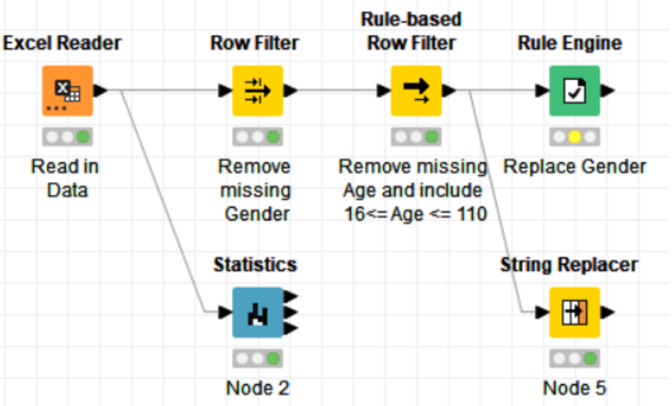
Input with replaced values - 2:5 - String Replacer

File

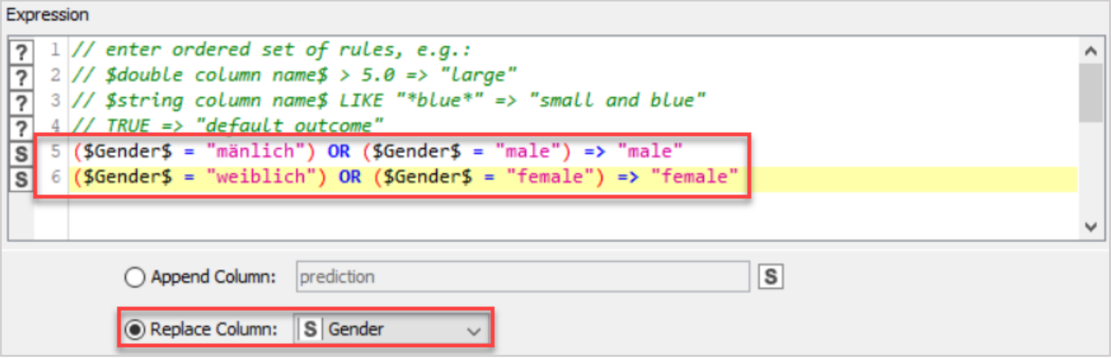
Table "default" - Rows: 996 | Spec - Columns: 8 | Properties | Flow Variables

Columns: 8	Column Type	Column Index	Color Handler	Size Handler	Shape Han...	Filter Handler	Lower Bound	Upper Bound	Value 0	Value 1	Value 2	Value 3
rowNumber	Number (integer)	0					1	1000	?	?	?	?
PostalCode	String	1					?	?	?	?	?	?
HashCode	String	2					?	?	?	?	?	?
Age	Number (integer)	3					2	234	?	?	?	?
Gender	String	4					?	?	male	female	weiblich	?
Payment Method	String	5					?	?	credit card	cheque	cheque	cash
LastTransaction	Local Date Time	6					2009-11-24...	2014-02-26...	?	?	?	?
ChurnDate	Local Date Time	7					2010-07-16...	2014-04-07...	?	?	?	?

4. You can add another String Replacer to replace the German word for female. If not, we can use the versatile **Rule Engine** node to perform all the replacement at one go.



5. Open the Rule Editor and enter the following rules:
($Gender$ = "mänlich") OR ($Gender$ = "male") => "male"
($Gender$ = "weiblich") OR ($Gender$ = "female") => "female"



Expression

```
// enter ordered set of rules, e.g.:
// $double column name$ > 5.0 => "large"
// $string column name$ LIKE "*blue*" => "small and blue"
// TRUE => "default outcome"
($Gender$ = "mänlich") OR ($Gender$ = "male") => "male"
($Gender$ = "weiblich") OR ($Gender$ = "female") => "female"
```

○ Append Column: prediction

● Replace Column: S Gender

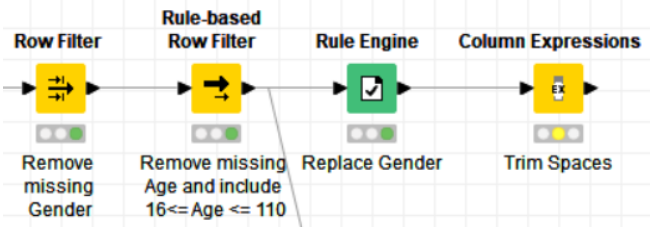
Remember to set the Replace Column to Gender.

6. Rename the node as **Replace Gender**. Execute and check the output to confirm the replacement are indeed applied.

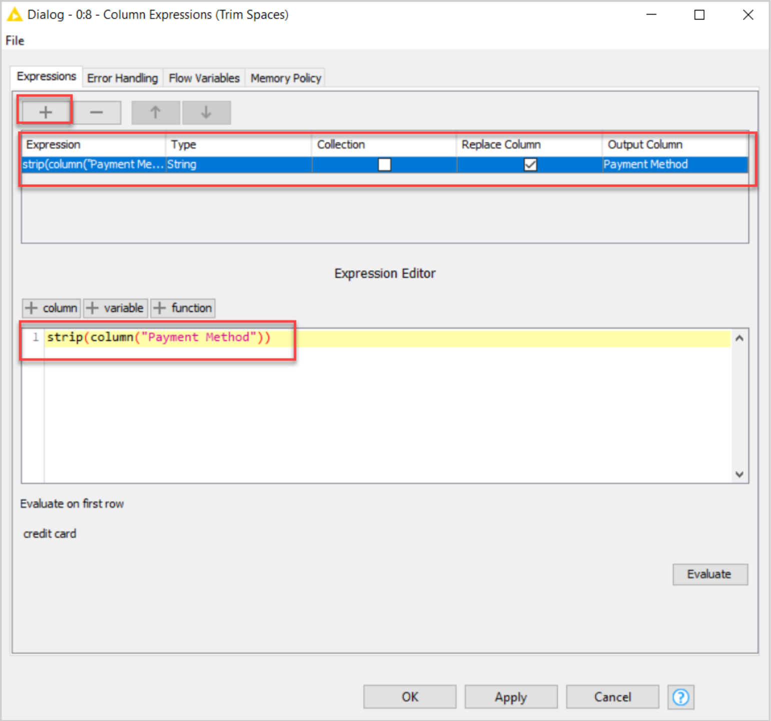
Remove Leading Spaces for Payment Method Attribute

We had observed that the entry “cheque” seems to appear twice when considering unique values in the Payment Method attribute. The reason is there are empty spaces in front of some the entries i.e. “ cheque”.

1. We can use a **Column Expression** node (*you may need to install the KNIME Expressions extension if you have not done so*) to strip nominal data of leading and trailing spaces. Add the **Column Expression** node to the output of the Replace Gender Rule Engine node. Rename it as **Trim Spaces**.



2. Open the configuration dialog. Click on the **+** to create a new Expression. Configure as follows:
   a. Set the Type to **String**
   b. Check the Replace Column
   c. Select **Payment Method** for the Output Column
   d. Enter the following Expression:
   strip(column("Payment Method"))



Expression	Type	Collection	Replace Column	Output Column
strip(column("Payment Me...	String	□	☑	Payment Method

```
strip(column("Payment Method"))
```

credit card

3. Execute and check the output. The leading spaces in front of “cheque” is removed.

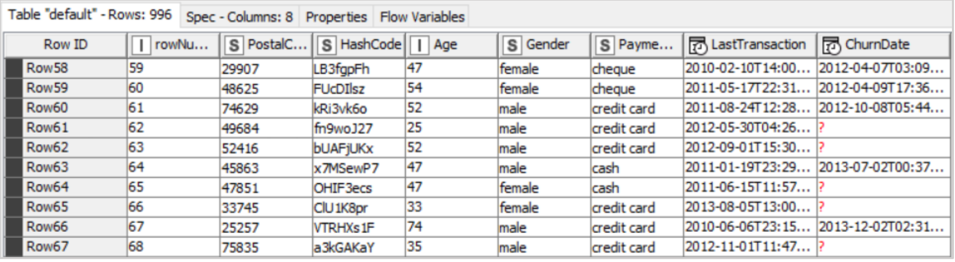


Table "default" - Rows: 996 | Spec - Columns: 8 | Properties | Flow Variables

Row ID	I rowNu...	S PostalC...	S HashCode	I Age	S Gender	S Payme...	LastTransaction	ChurnDate
Row58	59	29907	LB3fgpFh	47	female	cheque	2010-02-10T14:00...	2012-04-07T03:09...
Row59	60	48625	FUcDIlsz	54	female	cheque	2011-05-17T22:31...	2012-04-09T17:36...
Row60	61	74629	kRi3vk6o	52	male	credit card	2011-08-24T12:28...	2012-10-08T05:44...
Row61	62	49684	fn9woJ27	25	male	credit card	2012-05-30T04:26...	?
Row62	63	52416	bUAFjUKx	52	male	credit card	2012-09-01T15:30...	?
Row63	64	45863	x7MSewP7	47	male	cash	2011-01-19T23:29...	2013-07-02T00:37...
Row64	65	47851	OHIF3ecs	47	female	cash	2011-06-15T11:57...	?
Row65	66	33745	ClU1K8pr	33	female	credit card	2013-08-05T13:00...	?
Row66	67	25257	VTRHXs1F	74	male	credit card	2010-06-06T23:15...	2013-12-02T02:31...
Row67	68	75835	a3kGAKaY	35	male	credit card	2012-11-01T11:47...	?

Generate Attributes

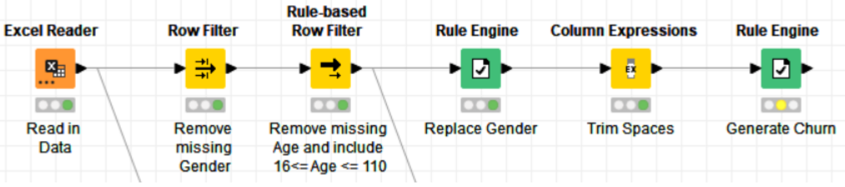
Since a missing churn date means that the customer is still active, the entries that do not have churn data should not be removed.

We would want to create a model that simply predicts whether a customer will churn in the near future or remain loyal. To train this model we need a new Churn attribute with the following rule:

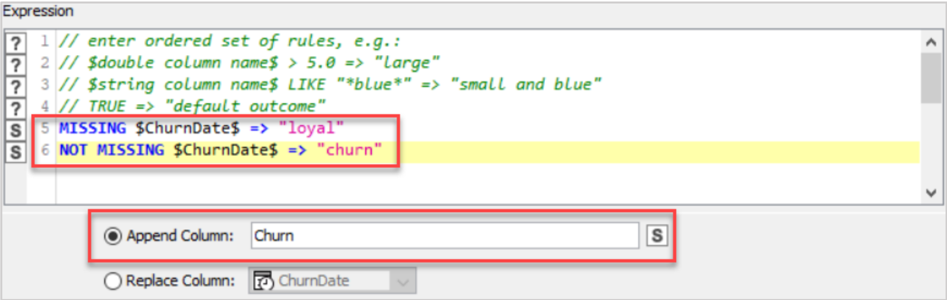
If the churn date is missing, then loyal otherwise churn.

The above rule will result in a nice and neat binominal attribute that contains exactly the information that we want to predict later on - the perfect label for our supervised learning problem. In KNIME, new attributes can be generated together with the Rule Engine node.

1. Add another **Rule Engine** node to the output of the Replace Gender Rule Engine node. Rename it as **Generate Churn**.



2. Open the Rule Editor and enter the following rules:
MISSING $ChurnDate$ => "loyal"
NOT MISSING $ChurnDate$ => "churn"



Expression

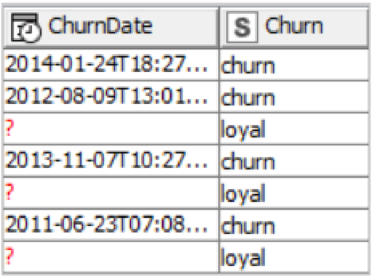
```
// enter ordered set of rules, e.g.:
// $double column name$ > 5.0 => "large"
// $string column name$ LIKE "*blue*" => "small and blue"
// TRUE => "default outcome"
MISSING $ChurnDate$ => "loyal"
NOT MISSING $ChurnDate$ => "churn"
```

● Append Column: Churn

○ Replace Column: ChurnDate

Select Append Column and name the column as **Churn**.

3. Execute and check the output. You can see an additional column created with the value based on your rules.

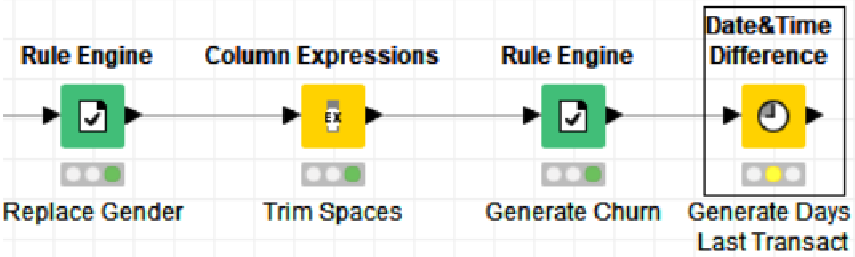


ChurnDate	Churn
2014-01-24T18:27...	churn
2012-08-09T13:01...	churn
?	loyal
2013-11-07T10:27...	churn
?	loyal
2011-06-23T07:08...	churn
?	loyal

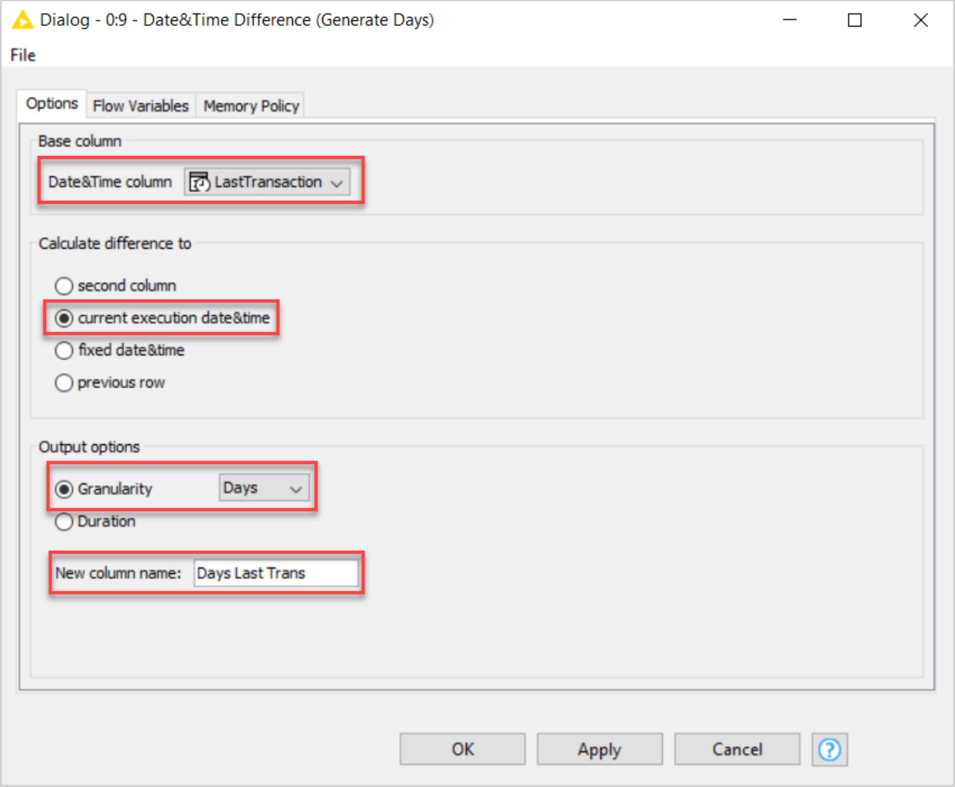
Data Type Conversion

To allow learning models handle date timestamps better, we need to change the data type of date time to a numerical type. One of the ways to do this is to extract the days since last transaction.

1. We can use the **Date&Time Difference** node to calculate the number of days since last transaction. Add the **Date&Time Difference** node to the output of the Generate Churn Rule Engine node. Rename it as **Generate Days Last Transact**.



2. Open the configuration dialog. Configure as follows:
    a. Select **LastTransaction** for the Date&Time column
    b. Select **current execution date&time** for the Calculate difference to
    c. Select **Days** for Granularity
    d. Name the new column as **Days Last Trans**



Dialog - 0:9 - Date&Time Difference (Generate Days)

File

Options | Flow Variables | Memory Policy

Base column

Date&Time column: LastTransaction

Calculate difference to

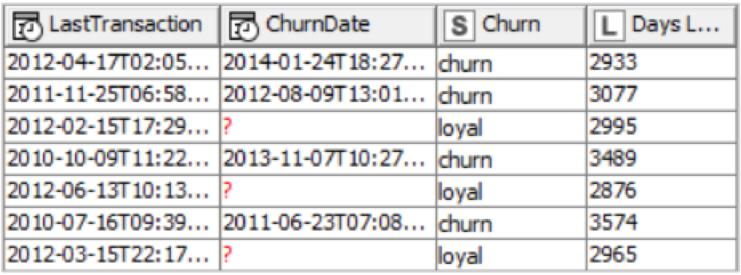
○ second column
● current execution date&time
○ fixed date&time
○ previous row

Output options

● Granularity: Days
○ Duration

New column name: Days Last Trans

3. Execute the node and check the output. You can see a new column created with the days calculated based on the LastTransaction column.



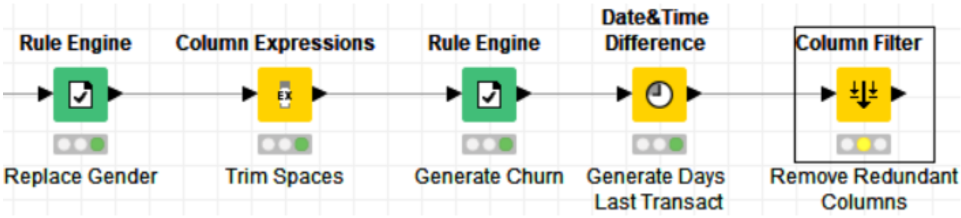
LastTransaction	ChurnDate	Churn	Days L...
2012-04-17T02:05...	2014-01-24T18:27...	churn	2933
2011-11-25T06:58...	2012-08-09T13:01...	churn	3077
2012-02-15T17:29...	?	loyal	2995
2010-10-09T11:22...	2013-11-07T10:27...	churn	3489
2012-06-13T10:13...	?	loyal	2876
2010-07-16T09:39...	2011-06-23T07:08...	churn	3574
2012-03-15T22:17...	?	loyal	2965

Remove Redundant Attributes

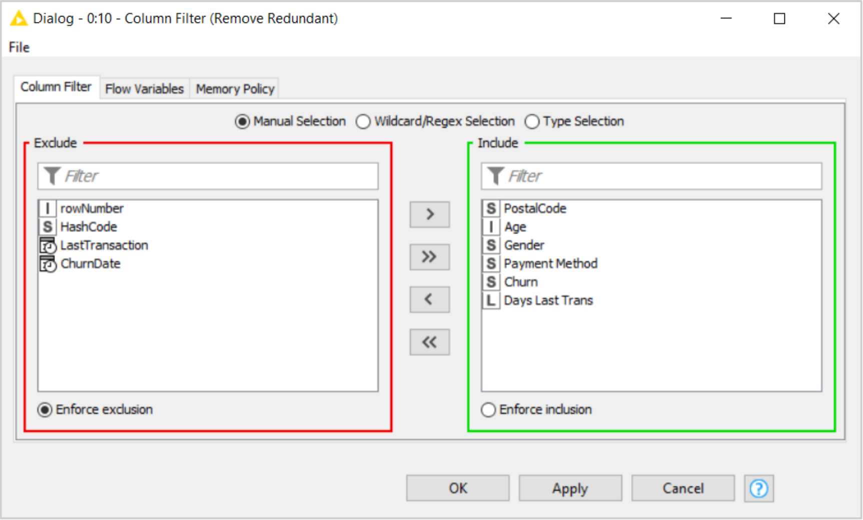
During supervised learning an algorithm tries to find relations between the input attributes and the label attribute (which we will create at a later step). HashCode may affect the running of the algorithm as it contains a seemingly random collection of numbers and letters. It is derived from the other attributes via a so-called hash function and is useful for fast database searching or equality comparisons between several roles, but for supervised learning this does not have any use. Therefore, it should be removed.

We need to remove the ChurnDate and LastTransaction attribute from the data since we have created the new column for analysis. And also, rowNumber was the original index column, but it is redundant now since KNIME has created an index column. By doing so, we allow the learning algorithm to find real relations between properties such as age, gender, etc., and the churn behavior.

1. A **Column Filter** node can remove these columns easlity. Add a **Column Filter** node to the output of the Date&Time Difference node. Rename it as **Remove Redundant Columns**.



2. Open the Configuration Editor. Select **rowNumber**, **HashCode**, **LastTransaction** and **ChurnDate** to be excluded.


Dialog - 0:10 - Column Filter (Remove Redundant)

File

Column Filter | Flow Variables | Memory Policy

● Manual Selection ○ Wildcard/Regex Selection ○ Type Selection

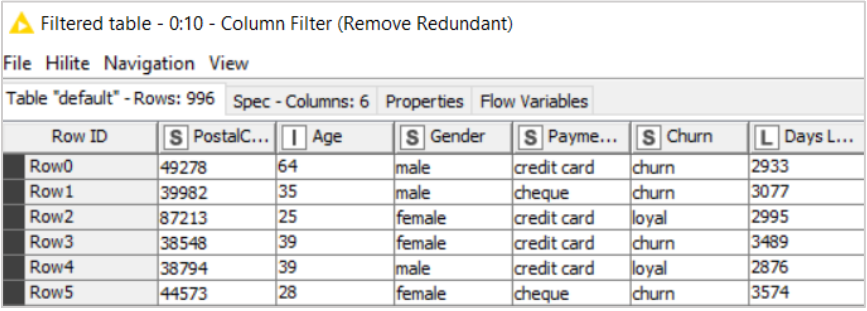
Exclude
- I rowNumber
- S HashCode
- LastTransaction
- ChurnDate

● Enforce exclusion

Include
- S PostalCode
- I Age
- S Gender
- S Payment Method
- S Churn
- L Days Last Trans

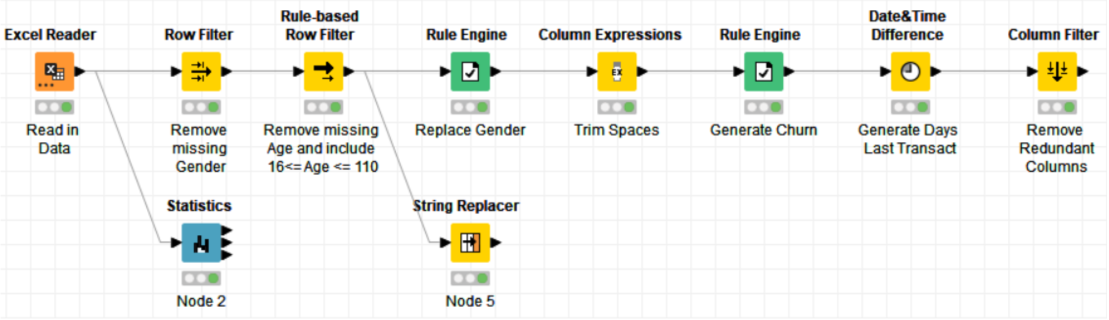
○ Enforce inclusion

3. Execute the node and check the output.


Filtered table - 0:10 - Column Filter (Remove Redundant)

Row ID	S PostalC...	I Age	S Gender	S Payme...	S Churn	L Days L...
Row0	49278	64	male	credit card	churn	2933
Row1	39982	35	male	cheque	churn	3077
Row2	87213	25	female	credit card	loyal	2995
Row3	38548	39	female	credit card	churn	3489
Row4	38794	39	male	credit card	loyal	2876
Row5	44573	28	female	cheque	churn	3574

4. Save your workflow. Your completed workflow should look like this:



Task 4: Export Cleaned Data (Try-it-yourself)

Search for the node that can export your cleaned data. Try this step on your own.

~The End~