

Edited by
ELISABETH RICHTER

KNIME, Automation, and AI

The KNIME for Finance Collection

KNIME v5.3



Copyright © 2024 by KNIME Press

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording or likewise.

For information regarding permissions and sales, write to:

KNIME Press

Talacker 50

8001 Zurich

Switzerland

knimepress@knime.com

www.knime.com

It's Time for Finance Automation

Finance departments are at the heart of each company. Finance activities include planning, forecasting, budgeting, and analysis to support major business decisions and analyze the overall financial health of the company, as well as accounting and treasury tasks. All those activities require precision, correctness, repeatability, and speed. In order to obtain repeatable, reliable, correct, and fast procedures, many companies have already started digital transformation projects for all finance processes.

You don't need a full lab of software developers to implement the automated procedures that a finance department needs. Low code tools, like KNIME Analytics Platform, provide the same level of reliability as programming languages but can be adopted also by non-programmers. Next to being an AI enabling and data science tool, KNIME Analytics Platform is often used to automate processes and analysis. It also offers a large variety of connectors, all standardized, all presenting the same UI, making it very easy to connect to all types of data sources: databases, ERP systems, CRM software, web resources, files, cloud repositories, and more.

KNIME Business Hub, the commercial counterpart of KNIME Analytics Platform, allows for collaboration, productionization, and scheduling of all implemented operations, which means it allows for automation. This is very practical in finance departments, as you can generate the report you need on Monday morning, regularly, using the latest data, on time, and at a low cost, without any manual work.

One last note is about governance. In finance use cases, governance is always critical. To make sure you remain within the legal boundaries, you need to know at any point which data is being used, for which task, where it is saved, and in general you need to understand what each workflow does. In KNIME Analytics Platform, due to its step-wise execution and its visual interface, you can always inspect the results of each step individually. In this way all data produced by an application can be traced back to its original data, understood, and modified if needed.

In this booklet, we collected a set of jump-start workflows for common operations in finance departments. All those blueprint workflows have been stored in a dedicated repository on the KNIME Community Hub, named KNIME for Finance, and are available for free download and usage. So far, the [KNIME for Finance](#) space includes the following topics covered in a generic finance department: Accounting, Audit & Compliance, FP&A, KPIs, Tax and Financial Services.

Table of Contents

ACCOUNTING	1
BANK RECONCILIATION	2
INVOICE AND DUNNING PROCESS MANAGEMENT	13
EVALUATE PROFITABILITY WITH FAST TIMESHEET AGGREGATION IN KNIME	19
AUDIT & COMPLIANCE	24
FRAUD DETECTION USING A SUPERVISED MACHINE LEARNING MODEL	25
FRAUD DETECTION USING QUANTILES	31
INTRODUCING AI IN FINANCE DEPARTMENTS	37
FINANCIAL PLANNING & ANALYSIS	47
BUDGET MONITORING	48
5 FINANCIAL METRICS EVERY FP&A ANALYST MUST KNOW ABOUT	53
MONTHLY AND YEAR-TO-DATE REVENUE AGGREGATIONS	57
FINANCIAL SERVICES	63
CREDIT SCORING FOR LOAN APPROVAL	64
TAX	72
SALES TAX REPORTING	73
TRANSFER PRICING RECHARGE	91
KPI	103
CUSTOMER CHURN KPI	104
EMPLOYEE TURNOVER KPI	110
REVENUE GROWTH KPI	117

NODE & COMPONENT INDEX

124

Accounting

With KNIME, accounting professionals can automate processes such as account reconciliation, journal entry, and month-end closing. KNIME connects to different accounting systems, frees up time for meaningful tasks, and enables better transparency through process documentation of visual workflows. In this chapter, we introduce some use cases for recording, summarizing, analyzing, and reporting financial transactions in an organization.

This chapter includes the articles:

- Bank Reconciliation, p. 2
- Invoice and Dunning Process Management, p. 13
- Evaluate Profitability with fast Timesheet Aggregation in KNIME, p. 19

Bank Reconciliation

Author: Elisabeth Richter

Workflows on KNIME Community Hub: [Bank Reconciliation](#) & [Data App for Bank Reconciliation](#)

Rule number one in accounting is ensuring you always have a grasp on your company's financial performance. You basically need to understand where money is coming from and going to. The process is called account reconciliation and it's crucial.

You only want correct values to be reported in the general ledger. By comparing financial records, you can confirm they are consistent and complete. Discrepancies will uncover mistakes from simple errors through to serious fraudulent activities.

There are many different types of account reconciliation, e.g., vendor reconciliation, business-specific reconciliation, intercompany reconciliation, or customer reconciliation.

This article focuses on bank reconciliation. It's one of the most popular types of account reconciliation and compares a company's internally recorded transactions, e.g., the cash book, with externally recorded transactions, e.g., bank statements.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Bank Reconciliation](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Understand Financial Performance with Account Reconciliation

We set ourselves the task to build a KNIME solution that handles account reconciliation in order to understand financial performance. This involves:

- Understanding where your money is coming from and going to
- Ensuring transactions are properly documented
- Aligning internal financial ledgers and statements with external financial statements.

Bank reconciliation process involves following these four steps:

Step 1: Collect the data and bring it into a comparable format.

Before being able to start your reconciliation process, first of all make sure to collect all available data, ensure that the data is complete, and bring it into a comparable format. We need to collect data from the **company cash book** and **bank statements**.

A company's **cash book** is a subsidiary of the general ledger in which all cash-related transactions are chronologically recorded, and it contains an opening balance (i.e., the amount brought forward from the previous accounting period) and a closing balance (i.e., the amount carried forward to the next period).

A **bank statement**, provided by your bank, contains all account activities on your bank account, also chronologically ordered. A bank statement is usually provided once a month and states the opening and closing balances on the first and last day of the period, as well as each transaction amount debited or credited, and the total account balance.

Due to their nature, cash book data and bank statements are likely provided in different formats and file types. For example, bank statements are often provided as OFX, CSV, or even PDF, whereas cash books might be managed through accounting tools or the data is stored in a database and needs to be exported first. Collecting the data from these different sources can be challenging.

Whereas bank statements are usually structured as straight-forward tables (see the bank statement shown below), cash books tend to be in a more complex structure. In general, a cash book is divided into two parts, a **debit side** and a **credit side**. Transactions recorded on the debit side refer to all sorts of receipts, so everything that increases our assets, and the transactions recorded on the credit side refer to all sorts of payments, hence everything that decreases our assets.

As a result, one data row might store multiple transactions, one on the Debit side and one on the Credit side (see the company cash book shown below).

We need to do some data wrangling to declutter the two-side structure into a simpler table structure.

Step 2: Compare entries of cash book and bank statement

Once your data is in a comparable format, you can systematically compare each and every line of your bank statement and your cash book and check whether they each appear in both records.

Mark all matching transactions and flag those that are either missing or where the transaction amount is not matching to get a better overview.

Note. It is important to not only check whether a transaction appears in both records but also that their transaction amounts are equal.

Now make an assessment of the situation. Calculate the difference between the cash book's closing balance and the bank statement's closing balance to get the **unreconciled amount**.

The overall aim of reconciliation is to stepwise reduce the unreconciled amount to zero which in turn means that the closing balances equal each other.

Step 3: Calculate adjusted closing balance for both the cash book and the bank statement

In this step, we account for the differences detected in Step 2 and adjust the cash book and bank statement by calculating the adjusted closing balance for both records.

There are in general three causes for such discrepancies:

- **Omissions.** These are activities on the bank account that are not yet included in the cash book. This comes from transactions of which the company does not know about until they appear on the bank statement, for example, bank fees or in case of missing receipts or non-sufficient funds. Such cases must be detected and the books must be adjusted accordingly.
- **Timing Differences.** Sometimes, transactions are recorded in different time periods in the cash book and the bank statement, for example, when a payment has already been recorded in the cash book but has not yet been processed by the bank. Such timing differences usually adjust themselves in the future.
- **Errors by the Accountant or the Bank.** Those are erroneous transactions, for example, the recording of an incorrect transaction amount. Errors can happen on both ends but note that errors by the bank are rather rare. Hence, before considering this option make sure you have thoroughly checked every transaction and eliminated every other discrepancy before.

Step 4: Final check: Compare adjusted closing balances

After accounting for each discrepancy, we now have two adjusted closing balances which equal each other. The unreconciled amount should be zero. The adjusted closing balance corresponds to the **True Cash Balance**.

If your adjusted closing balances still don't match, you need to go back each step and trace down the source of the error. If you haven't closed your books properly in the previous period, you might find the error there.

Accounting
Bank Reconciliation

Once the adjusted closing balances are equal, and the unreconciled amount is zero, you can take it from here and prepare your journal entries. Then, the next time you do a bank reconciliation you won't face the same issues again.

The Workflow: Generate a Bank Reconciliation Statement from Cash Book & Bank Statements

For this blog post, I used sample data available on [Kaggle](#) which I added some additional synthetic entries to match the desired use case. Both records are provided as Excel files and are for the month of June 2023 (see figures below). The cash book is a single column cash book, the easiest type. Other types are double column or triple column.

Bank Statement				
Date	Particulars	Debit	Credit	Balance
1-Jun-23	Opening Balance			186,200
5-Jun-23	Electricity Bill Cheque # 0007864	24,300		161,900
5-Jun-23	Dividend Income - Online Transfer		2,600	164,500
5-Jun-23	Receipt Cheque # 0007981		21,200	185,700
8-Jun-23	Payment Cheque # 0007866	17,400		168,300
10-Jun-23	Outward Payment Cheque # 0007867	1,700		166,600
13-Jun-23	Receipt Cheque # 07982		18,500	185,100
14-Jun-23	Bank Charges	3,200		181,900
20-Jun-23	Payment Cheque # 0007865	30,700		151,200
20-Jun-23	Receipt Cheque # 07983		11,800	163,000
21-Jun-23	Outward Payment Cheque # 0007868	9,500		153,500
21-Jun-23	Rental Paid Cheque # 7870	16,100		137,400
24-Jun-23	Bank Charges	1,800		135,600
27-Jun-23	Receipt Ch. # 0007984		4,700	140,300
28-Jun-23	Direct Debit	8,800		131,500
29-Jun-23	Payment Cheque # 0007873	1,200		130,300
29-Jun-23	Receipt Ch. # 0007985		27,900	158,200
30-Jun-23	Payment Cheque # 0007871	2,500		155,700
30-Jun-23	Closing Balance			155,700

The fictional bank statement I used in this blog post, available on Kaggle. A bank statement is where the bank records all transactions issues to your bank account and is usually sent out once a month.

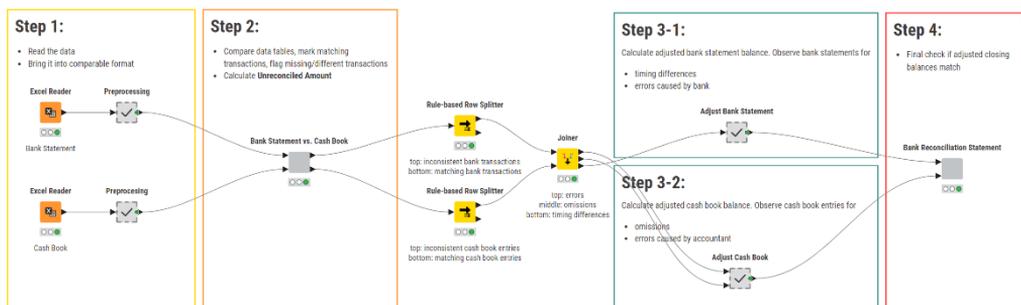
Accounting Bank Reconciliation

Company Cash Book					
Debit - Dr			Credit - Dr		
Date	Details	Amount (\$)	Date	Details	Amount (\$)
01-Jun-23	Balance b/d	186,200	01-Jun-23	Electricity - 7864	24,300
04-Jun-23	Mr. Ali - 7981	21,200	02-Jun-23	Mr. Usman - 7865	30,700
09-Jun-23	Mr. Younis - 7982	18,500	05-Jun-23	Mr. Yaseen - 7866	17,400
19-Jun-23	Mrs. Samia - 7983	11,800	06-Jun-23	Mrs. Noor - 7867	1,700
24-Jun-23	Mr. Arif - 7984	4,700	10-Jun-23	Mr. John - 7868	9,500
27-Jun-23	Mrs. Ayesha - 7985	27,900	14-Jun-23	Mr. John - 7850	7,100
29-Jun-23	Mrs. Husna - 7986	9,800	16-Jun-23	Rental - 7870	16,100
30-Jun-23	Mr. Arshad - 7987	13,400	20-Jun-23	Mr. Kaleem - 7871	2,500
			21-Jun-23	Mr. Zubair AUS - 7952	3,700
			22-Jun-23	Mr. Naseer - 7873	1,020
			30-Jun-23	Balance c/d	179,480
	Total	293,500		Total	293,500

The fictional cash book we used in this blog post, available on Kaggle. A cash book is where a company records all cash-related transactions.

The solution workflow "[Bank Reconciliation](#)" is available for download from the KNIME Community Hub and performs the following steps:

- Read the two records into KNIME Analytics Platform using one *Excel Reader* node each, and bring them into a comparable format such that both tables are of the following structure: *Date, Particulars, Debit, Credit, Balance, CF, Description, and Transaction ID*.
- Compare each transaction in the two records. Transactions that are equal in both records are flagged green, transactions that are missing or show unequal transaction amounts are flagged red. The view created by this component also states the closing balances of both the bank statement and the cash book as well as the unreconciled amount.
- Calculate the adjusted closing balances for the bank statement and the cash book.
- Generate the **Bank Reconciliation Statement** and check whether the adjusted closing balances match after adjusting the records.

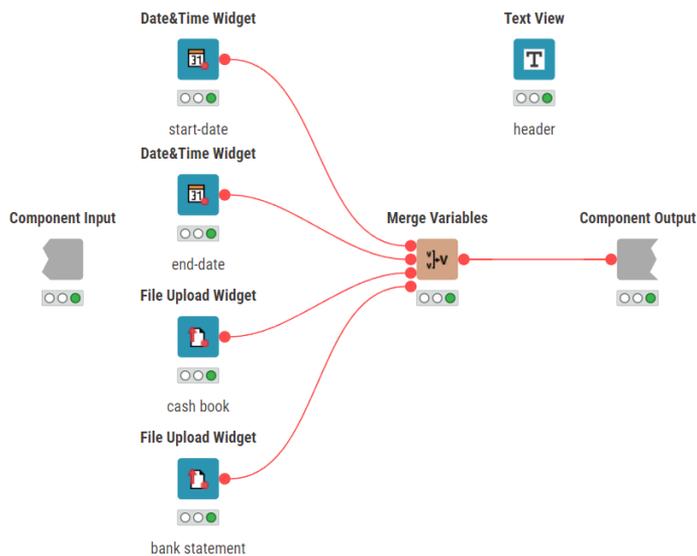


The example workflow "[Bank Reconciliation](#)" is available for download from the KNIME Community Hub.

A shareable Data App: Select custom Time Periods for Reconciliation

In order to create a slightly more sophisticated solution, I decided to turn my workflow into a browser-based data app that lets the user interact easily with the underlying workflow. For example, upload their own bank statement and cash book data, or define a specific time period for the bank reconciliation to be performed. A workflow being deployed as a data app can also be shared with colleagues without them having the need to interact with KNIME Analytics Platform at all.

To do so, I simply added two *Date&Time Widget* nodes as well as two *File Upload Widget* nodes to the workflow and encapsulated them into the “Select file & time period” component.



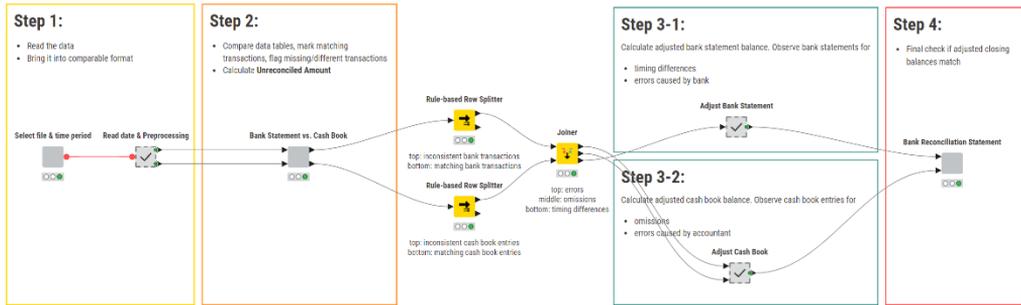
The inside of the “Select file & time period” component. It lets you upload your own data files (File Upload Widget node) and allows you to select an individual time period (Date&Time Widget node).

I’ve then connected the flow variable output of the component to the two *Excel Reader* nodes to parse the *Path* variable of the selected file to the respective reader nodes. To account for the individually defined time period, I’ve added a *Date&Time-based Row Filter* node after each of the reader nodes and used the *start-date* and *end-date* propagated by the component to filter both records respectively. This all is part of Step 1.

Besides the changes made to Step 1, the data reading, no other changes were required. You can see the final Data App workflow below.

Accounting

Bank Reconciliation



The example workflow "Data App for Bank Reconciliation" available for download from the KNIME Community Hub.

The Results: Semi-automated Bank Reconciliation

In this blog post, we have demonstrated how you can semi-automate bank reconciliation on some sample data with a KNIME workflow and then interact with the data conveniently with a data app to read in your own data and define individual time periods for reconciliation. It then carries out bank reconciliation by comparing each and every transaction recorded in the bank statement to the transactions recorded in the company-owned cash book. As a result, the workflow produces a bank reconciliation statement that tells you at a glance how your books need to be corrected.

The screenshots below show the three pages of the data app.

Accounting
Bank Reconciliation

Page 1:

The first page of the data app is the result of the “Select file & time period” component. The two *Date&Time Widget* nodes allows the end user to specify the individual accounting period and the two *File Upload Widget* nodes allow the end user to upload their own bank statement and cash book files.

The screenshot displays the 'Bank Reconciliation: Comparing Bank Statement and Cash Book' interface. At the top, it instructs the user to 'Select the individual time period for the bank reconciliation to be performed & upload your bank statement and cash book data'. Below this, there are three main sections: 'First day:' with a date input field showing '2023-06-01' and a 'Today' button; 'Last day:' with a date input field showing '2023-06-30' and a 'Today' button; and two file upload sections: 'Bank Statement data:' with a 'Select file' button and 'Bank Statement.xlsx' text, and 'Cash Book data:' with a 'Select file' button and 'Cash Book.xlsx' text. At the bottom right, there are 'Cancel' and 'Next' buttons.

Page 1 of the Bank Reconciliation data app. It allows the end user to specify the accounting period and to upload a bank statement und cash book file.

Accounting Bank Reconciliation

Page 2:

Page 2 displays the dashboard which is the result of the “Bank Statement vs. Cash Book” component (Step 2). The dashboard highlights each transaction that is identical in both records and flags those that are either missing in one record or report different transaction amounts. This allows us to obtain an initial assessment of the situation.

Bank Reconciliation: Comparing Bank Statement and Cash Book

2023-06-01 to 2023-06-30

Bank Statement					Cash Book						
Date	Particulars	Debit	Credit	Balance	-	Date	Details	Debit	Credit	Balance	-
2023-06-01	Opening Balance			186200		2023-06-01	Balance b/d			186200	
2023-06-05	Electricity Bill Ch...	24300		161900	✓	2023-06-01	Electricity - 7864	24300		161900	✓
2023-06-05	Dividend Income...		2600	164500	✗	2023-06-02	Mr. Usman - 7865		30700	131200	✓
2023-06-05	Receipt Cheque ...		21200	185700	✓	2023-06-04	Mr. Ali - 7981	21200		152400	✓
2023-06-08	Payment Cheque...	17400		168300	✓	2023-06-05	Mr. Yaseen - 7866	17400		135000	✓
2023-06-10	Outward Payme...	1700		166600	✓	2023-06-06	Mrs. Noor - 7867	1700		133300	✓
2023-06-13	Receipt Cheque ...		18500	185100	✓	2023-06-09	Mr. Younis - 7982	18500		151800	✓
2023-06-14	Bank Charges	3200		181900	✗	2023-06-10	Mr. John - 7868	9500		142300	✓
2023-06-20	Payment Cheque...	30700		151200	✓	2023-06-14	Mr. John - 7850	7100		135200	✗
2023-06-20	Receipt Cheque ...		11800	163000	✓	2023-06-16	Rental - 7870		16100	119100	✓
2023-06-21	Outward Payme...	9500		153500	✓	2023-06-19	Mrs. Samia - 7983	11800		130900	✓

The closing balance provided by the bank (\$155700) does not match the closing balance recorded in the cash book (\$179480). The unreconciled amount is \$23780.

Evaluating the available data shows the following discrepancies:

- Timing Differences
- Omissions
- Errors (wrong transaction amount)

← Back
Cancel
Next

Page 2 of the Bank Reconciliation data app. The dashboard highlights each transaction that is identical in both records and flags those that are either missing on one record or report different transaction amounts.

Accounting
Bank Reconciliation

Page 3:

The final page of the data app shows the Bank Reconciliation Statement. It's a summary of the detected discrepancies in our books and tells us how we need to prepare our journals in order to reflect the true financial situation of our company.

Remember: Timing Differences correct themselves in the future, however, omissions and other errors need to be accounted for.

In addition, it shows the unadjusted and the adjusted closing balances for both records and states the unreconciled amount. Ideally, this is equal to zero, but it could be the case that even after the bank reconciliation we still have discrepancies left in our data. If this is the case, you need to go back each and every step and detect where you went wrong. It might be an issue carried forward from the previous period or a more severe error caused by an accountant or the bank.

Bank Reconciliation: Comparing Bank Statement and Cash Book			
2023-06-01 to 2023-06-30			
Bank Statement		Cash Book	
Statement	\$	Statement	\$
Unadjusted Closing Balance:	155700	Unadjusted Closing Balance:	179480
Deduct: Outstanding cheque	(7100)	Correct error: Payment Cheque # 0007873	(180)
Deduct: Outstanding cheque	(3700)	Add: Dividend Income - Online Transfer	2600
Add: Deposit in transit	9800	Deduct: Bank Charges	(3200)
Add: Deposit in transit	13400	Deduct: Bank Charges	(1800)
Adjusted Closing Balance:	168100	Deduct: Direct Debit	(8800)
		Adjusted Closing Balance:	168100

Accounting for all discrepancies results in an adjusted bank statement closing balance of \$168100 and an adjusted cash book closing balance of \$168100. Since the adjusted closing balances are identical, the unreconciled amount equals \$0. You can now take it from here and prepare your journal entries respectively.

← Back

Page 3 of the Bank Reconciliation data app. It displays the Bank Reconciliation Statement, summarizing how to prepare our journals in order to reflect the true financial situation of our company.

You could even enhance this blueprint workflow, for example, by adding the KNIME Reporting nodes which lets you file the composite view as a PDF report, or adding a *Excel Writer* node to export the bank reconciliation results back to an Excel file.

In this example, I've used data from an Excel file. In reality, bank statements are often provided in OFX, CSV, or even PDF format, and also cash books come in various types.

If you're using your own data with this workflow example, you might need to tweak the data reading step (Step 1) a little, so that your file structure can be properly processed. For example, to read in PDF files you can use the *PDF Parser* node.

As soon as you've adjusted the data access to fit your data structure, you can let the workflow do the rest and automate your monthly bank reconciliation.

KNIME for Finance

Human error is one of the biggest pain points in the repetitive tasks that are part of account reconciliation. It's precisely these repetitive finance tasks that present the greatest opportunity for automation. With low-code/no-code data science you can create automated processes and reduce the risk of error. Reuse the KNIME solution and schedule it to run automatically whenever you need.

Invoice and Dunning Process Management

Author: Ryan Rudd

Workflow on KNIME Community Hub: [Dunning Management Process](#)

Yearly renewals are often at the heart of a business' responsibility to maintain a healthy cash flow and sustain business growth. Despite its importance, it's often an area that lacks efficiency or procedure.

The goal of a dunning/renewal management application is to track accounts that have paid on time and, more importantly, to implement a procedure to address accounts that are past their allotted due date. In this article we show how you can improve invoice and dunning process management by using a data science tool.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Dunning Process Management](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Improve financial Efficiency with streamlined Dunning Processes

We set ourselves the task to build an invoice and dunning process management solution that will both improve the efficiency of the finance team and provide the best service to customers.

In discussion with finance experts, we identified three key objectives to ensure a streamlined dunning process.

1. Track communications with late accounts through follow-ups and reminders to stay organized and keep customers aware of their current status.
2. Create more granular metrics around each customer to plan better in the future and ensure a more seamless renewal process.
3. Centralize data to give renewal account experts a single location to consume all the data and improve efficiency.

Let's look at how we put the solution together.

Enable easy Access to the relevant Data

One of the most important aspects of the new solution is getting access to the locations where the data related to your accounts is stored. This step is critical to centralizing this procedure away from siloed locations.

Typically, you'll be accessing data through a CRM, internal databases, invoice-related applications, or any combination of the three. This type of more complex data access can be handled easily by data science tools, as they offer a wider range of data operations. A low-code tool gives you a visual, intuitive interface so you don't need programming expertise to use it. KNIME Analytics Platform is also open source, so free for you to download and use.

In our dunning management task, the most important information is the renewal date, followed by unique customer identifiers, associated account personnel, and additional data fields that would be beneficial to the analyst working with this application.

We also want our solution to keep track of communication around the accounts. For example, let's say there was a message to be sent out to an account via our KNIME solution. Keeping track of communications could help the account team strategize next outreaches or use this information as historical data to plan for future renewals.

Our centralized data will now be the "single source of truth" and make sure we have all our critical data at our fingertips to use easily, at any time.

Enable easy Interaction with the Analysis

The data does not provide much use unless it's put in a more user-friendly and consumable manner. Shareable data apps are great for providing an easy way to include advanced data science into business processes without needing to train business users in data science. They give business users access to just the right touch points and can be easily extended or adapted.

We therefore want to build data app that gives the renewal account experts easy access to all the required information to manage their dunning process, such as:

- Listing overdue invoices
- Tracking overdue account communications
- Sending new reminders to accounts

While there is always more that can be built into the application, this should provide a good base. If needed, it's possible to incorporate additional customizations or functionality.

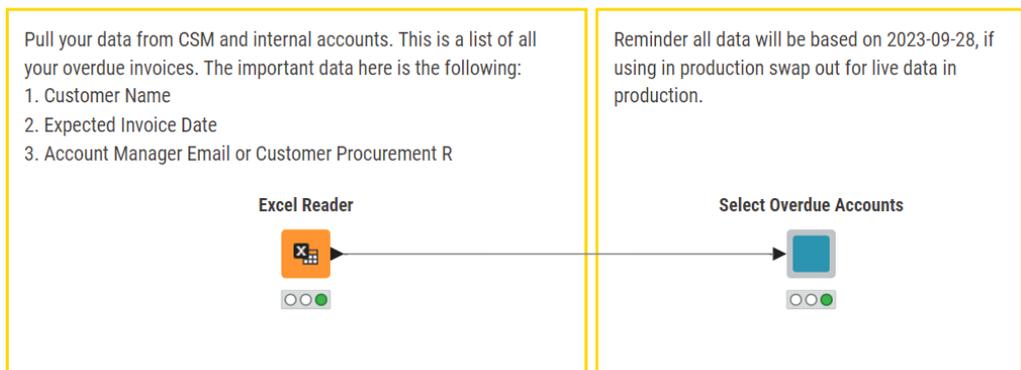
Enable easy Communication between Data & Business Experts

Now that we're feeding the data into the application and displaying it, the final step is to understand how people are going to be interacting with the application and provide the renewal account experts with an intuitive way to work with the data.

The data expert can build the workflow that the renewal accounts expert might not have been able to write themselves, but can understand because of the intuitive, visual, low-code environment. The immediate clarity of the approach enables the data expert and the renewal account expert to easily discuss improvements to the workflow.

The Workflow: Dunning Management

Let's walk through how I built the workflow. You can download the Dunning Management Process workflow [here](#) and try it out on your own dunning processes.



The workflow to streamline dunning management processes.

For the purposes of our example, I first generated a table with dummy data.

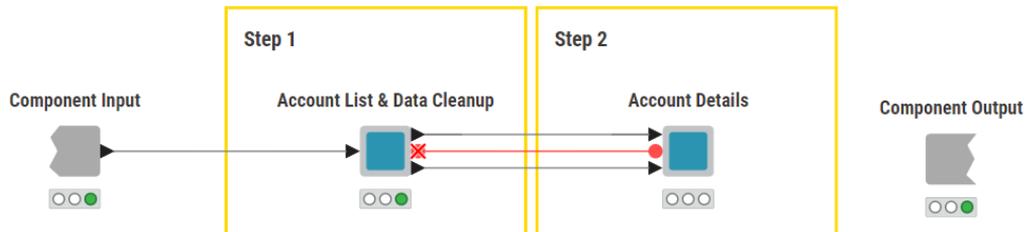
RowID	Partner String	Due Date Local Date	Payment Terms String	Total Number (double)	Invoice lines/Produ... String	today date Local Date	Account Manager String
Row0	Stellar Tech Solutions	2023-09-05	Salesforce Net 30	23,000	Product 2	2023-09-28	VZ
Row1	GreenWave Innovations	2023-07-31	Salesforce Net 30	23,452	Product 3	2023-09-28	AQ
Row2	QuantumSystems	2023-09-20	Salesforce Net 30	10,062.5	Product 1	2023-09-28	ND
Row3	Harmony Health Services	2023-08-09	Salesforce Net 30	41,446	Product 1	2023-09-28	ZD
Row4	NovaTech Industries	2023-09-23	Salesforce Net 60	28,750	Product 2	2023-09-28	ND
Row5	Silver Star Technologies	2023-04-30	Salesforce Net 30	58,905	Product 3	2023-09-28	ND
Row6	EcoGrowth Ventures	2023-08-31	Salesforce Net 30	54,600	Product 3	2023-09-28	WK
Row7	Zenith Labs	2023-07-31	Salesforce Net 30	33,245.33	Product 2	2023-09-28	ND
Row8	Infinity Design Studio	2023-08-29	Salesforce Net 60	104,120.67	Product 3 - Enterprise	2023-09-28	HH
Row9	BlueSky Energy	2023-07-07	Salesforce Net 30	14,500	Product 2	2023-09-28	MK
Row10	BlueSky Energy	2023-08-30	Salesforce Net 60	374,208	Product 3	2023-09-28	DJ
Row11	SwiftStream Logistics	2023-07-01	Salesforce Net 30	56,250	Services	2023-09-28	MP
Row12	Terrafusion Labs	2023-08-03	Salesforce Net 30	23,000	Product 2	2023-09-28	ND
Row13	SunFlare Solar Solutions	2023-06-14	Salesforce Net 30	23,000	Product 1	2023-09-28	VT

A table of dummy data.

The *Column Expressions* node is left as a reminder that the "Today's Date" column by default has a static value, and when ready for production, it should be fed through the

expressions node to replace the values with the actual date of when the workflow is running.

Moving on to the *Select Overdue Accounts* component – this is where a majority of the application’s functionality lives. The following screenshot is the internal content of the component and will be broken down into its separate sections.



Inside the component *Select Overdue Accounts*

What the workflow does:

It reads in a data table that tracks all comments and notes about every overdue customer. The Excel sheet is tightly tied to the outputs of the application. You can go into the *Account Info* component (inside the *Account Details* component) and see what is getting written to the Excel sheet. The *Refresh Button Widget* node allows for users to select an account, and once the button is selected, it will then pull additional info regarding that specific account.

The data is merged, and the output will designate a green or red color to each late account. If the color assigned to the account is green, this means that our account reps have made contact in the past seven days. If it has been longer than seven days, the color will be red, signifying that it is time to try and contact the account once again.

The *Table View* node displays the list of accounts with its respective color code to help guide the application’s consumer.

The data is formatted so that the renewal account expert can easily interact with the application and draw insights from the data.

The workflow provides additional account information and input fields for the renewal account expert to add comments or new information. When the renewal account expert selects an account, they will be able to expand past renewal comments made in current or past renewal cycles. After updating and refreshing, the workflow updates the red colored account to give it the green status.

Note. The outreach part of the application is not in this workflow. The decision is up to the developer to decide whether they want to reach out internally with updated information or to send emails to the account directly through the application.

A shareable Data App for Dunning Management

The renewal account experts will use the data app to manage their dunning processes. You can see what this looks like in the screenshots of the interface below.

Current Overdue Invoices
Show 10 entries
Search

Partner	Due Date	Payment Terms	Total	Invoice lines/Product	today date	Account Manager Full Name	Partner (#1)	Invoice/Bill Date	Due Date (#1)	date&time diff	to-do	cause-of-action	last-updated-note
Stellar Tech Solutions	2023-09-05	Salesforce Net 30	23000	Product 2	2023-09-28	VZ	Stellar Tech Solutions	2023-08-04	2023-09-05	21	Reach out to customer	Late Payment	2023-09-28
GreenWave Innovations	2023-07-31	Salesforce Net 30	23452	Product 3	2023-09-28	AQ	GreenWave Innovations	2023-07-01	2023-07-31	57	Reach out to customer	Late Payment	2023-09-28
QuantumSystems	2023-09-20	Salesforce Net 30	10062.5	Product 1	2023-09-28	ND	QuantumSystems	2023-08-03	2023-09-20	6	Reach out to customer third time	No Response	2023-09-21
Harmony Health Services	2023-08-09	Salesforce Net 30	41446	Product 1	2023-09-28	ZD	?	Invalid date	Invalid date	?	?	?	Invalid date
NovaTech Industries	2023-09-23	Salesforce Net 60	28750	Product 2	2023-09-28	ND	?	Invalid date	Invalid date	?	?	?	Invalid date

Showing 1 to 10 of 24 entries
Previous **1** 2 3 Next

Account Invoice History

[Invoice Data](#)

Account Manager

ND

Customer Name

Silver Star Technologies

Amount Due

58905.0

Product Type

Product 3

Additional Notes

Cause of Action

To Do

Update

[Send updated notes](#)

The data app for renewal account experts to manage dunning processes.

In the video on KNIMETV (linked above) you can see how a renewal account expert can use the data app. They are prompted to select an account, upload custom notes, and finally update the account with the inputted comments.

Note. Find more information about building data apps in the [Data Apps Beginners Guide](#).

KNIME for Finance

The use of data science tools is already spreading into the finance department! Spreadsheets have served us all well, but when data outgrows the capabilities of conventional tools, a data science tool can serve us with more sophisticated and robust techniques.

Download KNIME Analytics Platform and try out the workflow in this article for your own use case. It's free to use and adapt to your own invoice and dunning processes.

Evaluate Profitability with fast Timesheet Aggregation in KNIME

Authors: Elisabeth Richter & Heather Fyson

Workflow on KNIME Community Hub: [Timesheet Aggregation & Analysis](#)

It's 5 PM and your boss wants to know just how profitable the latest projects are for the business. Oh, and they need this information the next day by 8:30 AM, latest.

You know the drill and it's messy. Preparing for this kind of project profitability evaluation means you have to gather all the relevant employee timesheet data, salary data, as well as information about the projects, their timelines, and the clients. All this data is stored in multiple Excel files. Bringing it all into one place is a time-consuming and tedious task.

In this article we want to outline a low-code data science solution that will enable you to access all the data you need and make the calculations quickly and accurately.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Timesheet Aggregation](#) video to get an overview of this solution or browse the playlist for other solutions.

Key Takeaways of the KNIME Solution

- Access data easily even if it's scattered on different systems
- Automate for different data or time periods
- Visualize data plus option to automatically generate reports

The Task: Evaluate Project Profitability with Timesheet Aggregation

Companies account for the time workers spend on tasks and projects in timesheets. This data can be used for all kinds of purposes such as invoicing, payroll processing,

human resource management, and more. Here, we want to use it to check on the profitability of your company's projects by comparing your project revenues with costs.

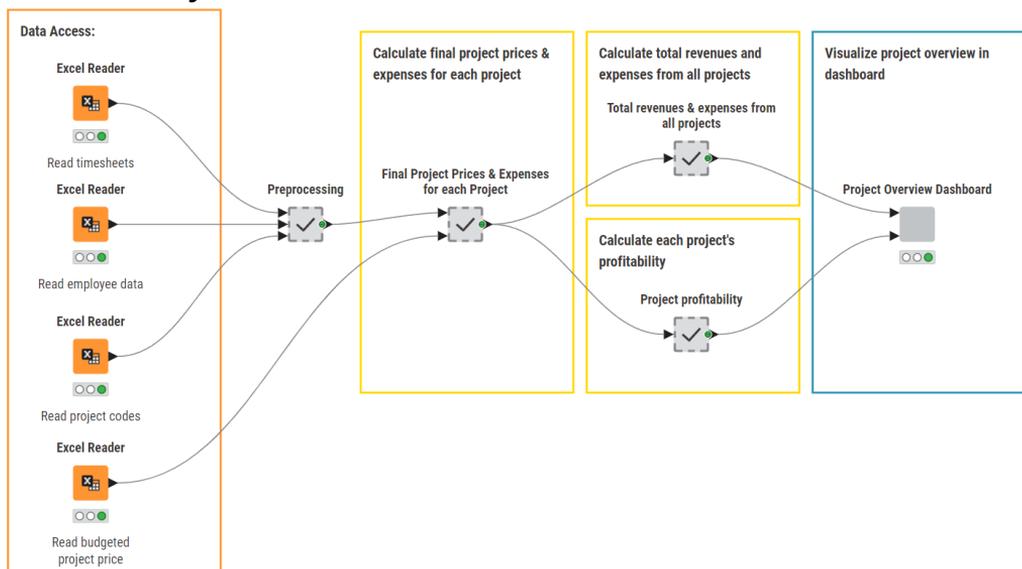
You'll use your employee timesheet and salary data to calculate the labor costs for each project and compare these with the final project costs and the price you agreed on with the client.

The comparison helps you understand just how profitable the project is as well as reveal any discrepancies between the actual expenses and the budgeted price.

You now need to get this data from timesheet files, salary files, project-related files that indicate whether the project was for an in-house or external client, and client-related files which provide information on the client's project budget.

Let's see how this is done by the Timesheet Aggregation & Analysis workflow in KNIME.

The Workflow: Access, Calculate, and Visualize Project Profitability



The [Timesheet Aggregation & Analysis](#) workflow to calculate and analyze project profitability.

The KNIME workflow is set up to:

- Access the four different Excel files to collect the data
- Preprocess the data to be ready for calculations
- Calculate the actual prices and expenses for each project
- Calculate the total revenues and expenses related to all projects

- Calculate the profitability of each project
- Visualize the results in a Projects Overview dashboard

Let's look at the calculation steps of this workflow in a bit more detail.

Calculate final Project Prices and the Expenses for each Project

The final project price is calculated as the duration an employee worked on the project times the hourly rate charged to the client.

The total expenses consist of the salary costs which are calculated as the duration an employee worked on the project times the employee's hourly salary rate.

Calculate total Revenues and Expenses from all Projects

The workflow calculates the total profit from all projects by summing up the revenue and expenses of all projects.

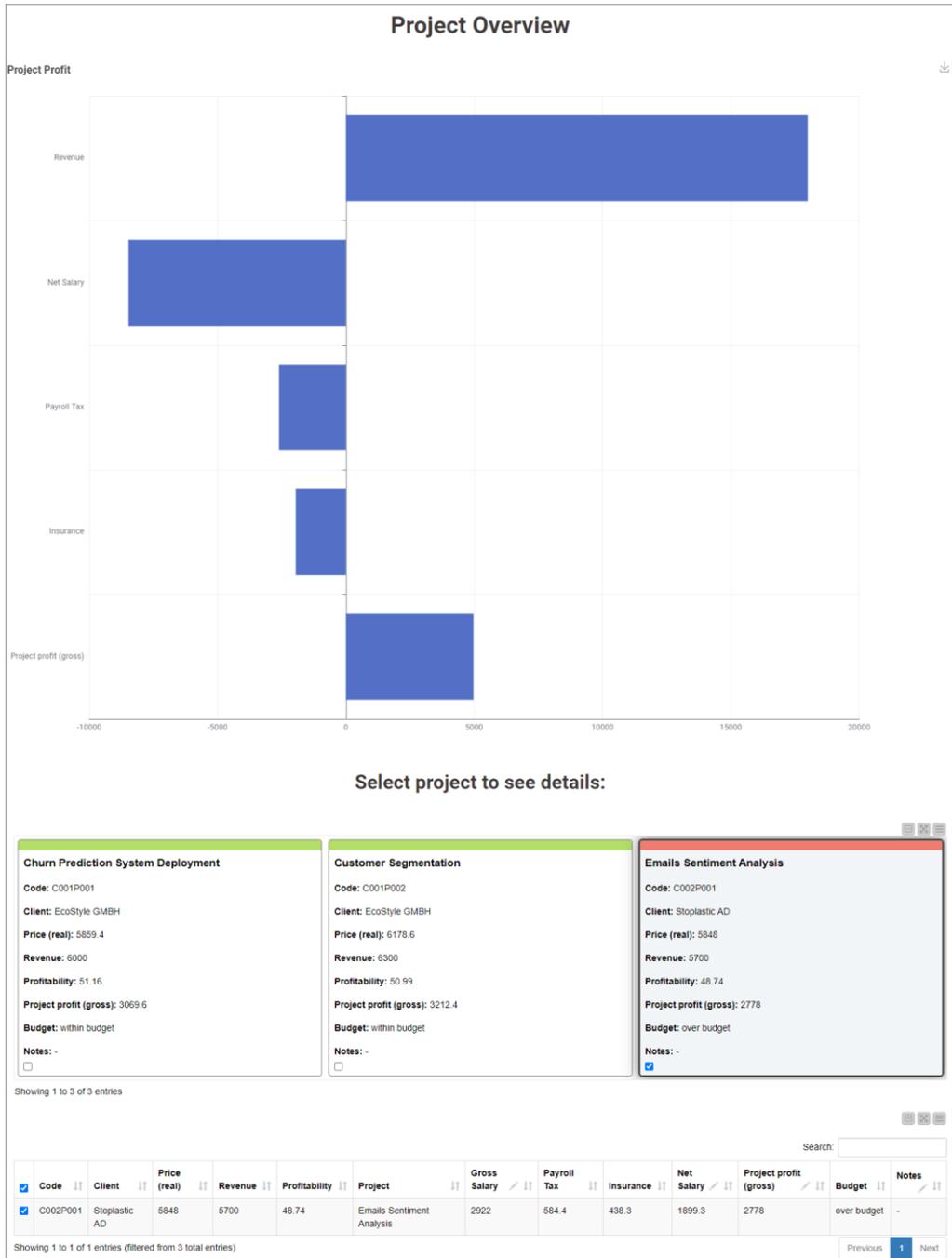
Calculate each Project's Profitability

Our solution now calculates each project's profitability as the ratio of its profit to its revenue. It's a performance indicator of the project that allows easy comparison.

Visualize Project Overview in Dashboard

The workflow outputs all the calculations to a Project Overview dashboard, which you could send to your boss. You can now identify your well- and not-so-well performing projects. Based on this information your company can develop strategies to address potential discrepancies.

Accounting
Evaluate Profitability with fast Timesheet Aggregation in KNIME

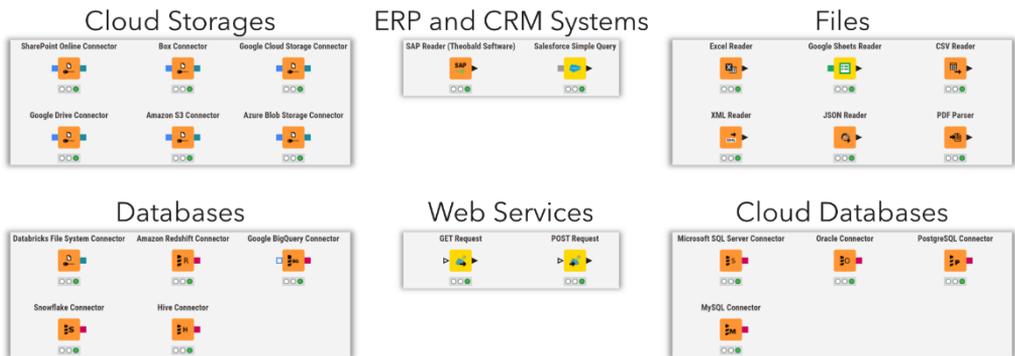


The interactive project overview dashboard.

The Results: A re-usable Timesheet Aggregation Solution to assess Project Profitability

The KNIME solution means you have the information your boss needs at your fingertips. Once set up, the KNIME workflow can be re-run at any time.

If you need to access additional files in different locations, KNIME offers many different [connectors](#) that you can add to your workflow and connect to the new source, for example, if your company has files in the cloud or in databases.



A collection of common KNIME connector nodes to connect to various data sources like cloud storages or databases.

You can also add more functionality to the workflow to generate and distribute reports faster, e.g., to automatically generate a PDF file of your dashboard, which you can send to your boss.

Why KNIME for Finance

KNIME Analytics Platform is an open-source free low code platform, offering a large variety of data operations. Thanks to its visual and intuitive user interface, implementing solutions does not require any programming expertise.

Download KNIME and explore more [KNIME solutions for finance](#). Use these example workflows as starting points for your own work or to get an idea about the capabilities KNIME provides for finance.

Audit & Compliance

With KNIME, audit and compliance professionals can perform end-to-end process testing and control validations. These professionals can significantly lower audit costs by automating tasks such as data collection, preparation, and analysis. In addition, they can identify irregular patterns and anomalies earlier in the process. In this chapter, we introduce some workflow examples for reviewing financial data, identifying areas of concern, and ensuring compliance with regulations, policies, and procedures.

This chapter includes the articles:

- Fraud Detection using a Supervised Machine Learning Model, p. 25
- Fraud Detection using Quantiles, p. 31
- Introducing AI in Finance Departments, p. 37

Fraud Detection using a Supervised Machine Learning Model

Author: Thor Landstrom

Workflows on KNIME Community Hub: [01_Training Random Forest for Fraud Detection](#) & [02_Deployment Random Forest for Fraud Detection](#)

Credit card fraud detection stands out as an ongoing challenge to accurately identify all new fraud patterns. Datasets containing fraud examples are rare, and when they do exist, they often include a limited number of outdated cases. This scarcity makes fraud detection particularly challenging, as it must continuously adapt to the evolving tactics of fraudsters.

There are two approaches to fraud detection:

1. **Classic machine learning based predictions**, when your dataset contains enough fraud examples
2. **Outlier detection based techniques**, when your dataset does not contain a sufficient number of fraud examples

The dataset that we will use contains a small percent of fraudulent transactions. Based on these examples, we will implement the classic machine-learning based approach for fraud detection for this article.

To see how to implement fraud detection algorithms using outlier detection-based techniques, refer to the next subchapter.

Whatever your data situation is, this series will show you how KNIME Analytics Platform offers a low-code solution for this problem. It can enable financial teams to automate data intake from various sources and leverage advanced analytics to detect fraudulent transactions, without the need for a coding background.

In this article on fraud detection, you'll learn how to use the Random Forest supervised learning algorithm to help identify fraudulent transactions.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Fraud Detection: Random Forest](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Identify fraudulent Transactions

Credit card transactions can essentially be divided into two categories: legitimate and fraudulent. The task at hand is to accurately identify and flag fraudulent transactions to ensure that a small minority of flagged transactions are legitimate.

The process of fraud detection often involves several manual and automated steps to analyze transaction patterns, customer behavior, and other relevant factors. For our purposes, we will only focus on the automation part of detection by training a model on a labeled dataset and applying it to a new transaction to simulate incoming data from an outside data source.

We use a popular dataset available from Kaggle called [Credit Card Fraud Detection](#). This dataset is composed of real, anonymized transactions made by credit cards in September 2013 by European cardholders. It includes 284,807 transactions over two days, containing 492 fraudulent transactions. The dataset represents a severe class imbalance between the “good” (0) and “fraud” (1), where “fraud” account for only 0.172% of the data.

The dataset contains 31 columns:

- V1 – V28: numerical input variables from a [PCA \(Principal Component Analysis\) transformation](#)
- Time: seconds elapsed from current transaction to first transaction
- Amount: transaction amount
- Class: “1” means fraud, “0” means good/other

A key feature needed for our training is “Class” as we need labeled data for a supervised training algorithm.

The process for creating our classification model follows the steps below. Even if there is data coming from multiple sources, the overall process does not change:

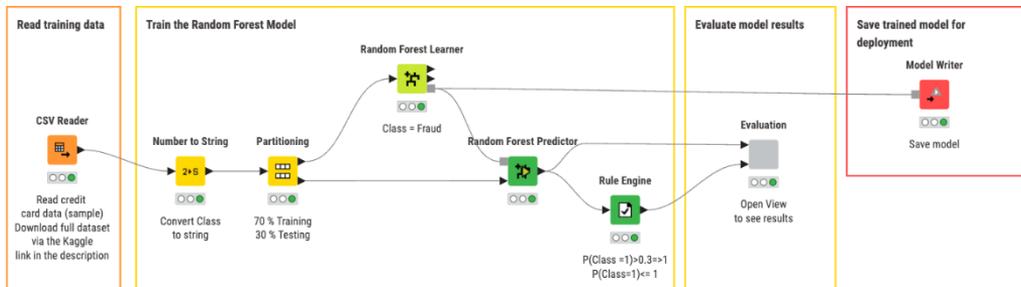
- Create/import a labeled training dataset
- Partition the data
- Train the model
- Evaluate model performance
- Import the new, unseen transactions
- Deploy the model and feed the new transactions in

Notify if any fraudulent transactions are classified

The Workflows: Training a Machine Learning Model to identify fraudulent Transactions

Note. All workflows used in this article are available publicly and free to download on the KNIME Community Hub. You can find the workflows on the KNIME for Finance space under [Fraud Detection](#) in the [Random Forest section](#).

The first workflow covers training our model. You can view and download the training workflow [Random Forest Model Training](#) from the KNIME Community Hub.



KNIME workflow to train the random forest model for Fraud Detection.

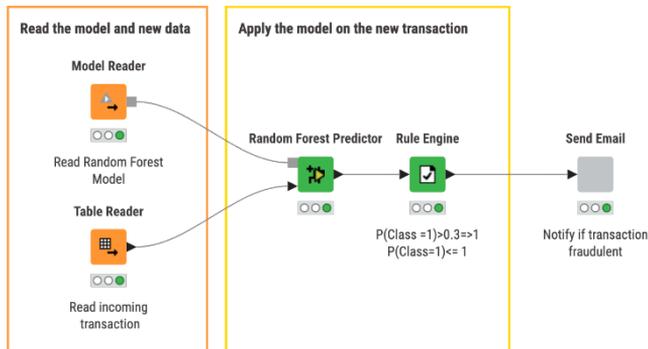
With this workflow you can:

- **Read training data** from a specified data source. In our case, we use data from the Kaggle dataset previously mentioned.
- **Partition the Data** by dividing the training data into a 70:30 split, where 70% of our data goes to training the model, and the other 30% for testing the model.
- **Train the Random Forest Model** by using the *Random Forest Learner* and *Random Forest Predictor* nodes, using appropriate settings.
- **Evaluate model results** by opening the view of the component “Evaluation” to check overall accuracy of the Random Forest Model.
- **Save the trained model for deployment** in the next workflow if you are satisfied with the performance of the model

An alternative method could be used if you have two separate datasets for training and testing. You can remove the partition and connect the *Number to String* node directly to the *Random Forest Learner* or *Random Forest Predictor* nodes depending on which dataset you are using (training goes top, testing goes to bottom).

As an additional metric, we have incorporated a *Rule Engine* node to adjust the threshold for decision-making. Typically, this threshold is set at 0.5, but we've modified it to 0.3. This adjustment aims to explore whether setting a lower threshold can increase the overall accuracy of the predictions by allowing a more sensitive classification criteria, albeit with a potential increase in false positives.

The second workflow [Random Forest Deployment](#) deploys the random forest model.



KNIME workflow to deploy a random forest model for Fraud Detection.

With this workflow you can:

- **Read both the previously trained model and new data** for classification
- **Apply the model on the new transaction** and optionally change the threshold value based on performance
- **Send an Email to notify if a transaction is fraudulent**

Inside the “Send Email” component, we check whether the transaction is fraudulent or not. If it is, an email is sent to the specified person for follow up.

The Results: A Model to classify Transactions

Let’s go back to the first workflow for a moment. From the training workflow, the results we get by opening the view of the “Evaluation” component are shown below.

Threshold = 0.5						Threshold = 0.3					
Confusion Matrix						Confusion Matrix					
		0 (Predicted)		1 (Predicted)				0 (Predicted)		1 (Predicted)	
0 (Actual)		85289		6	99.99%	0 (Actual)		85283		12	99.99%
1 (Actual)		40		108	72.97%	1 (Actual)		35		113	76.35%
		99.95%		94.74%				99.96%		90.40%	
Class Statistics						Class Statistics					
Class	Recall	Precision	Sensitivity	Specificity	F-measure	Class	Recall	Precision	Sensitivity	Specificity	F-measure
0	99.99%	99.95%	99.99%	72.97%	99.97%	0	99.99%	99.96%	99.99%	76.35%	99.97%
1	72.97%	94.74%	72.97%	99.99%	82.44%	1	76.35%	90.40%	76.35%	99.99%	82.78%
Overall Statistics						Overall Statistics					
Overall Accuracy		Cohen's kappa (κ)				Overall Accuracy		Cohen's kappa (κ)			
99.95%		0.824				99.94%		0.828			

Evaluation of the credit card Fraud Detection model.

We have two confusion matrices and several tables that show us overall results of the model using two different threshold values. As we can see, they are essentially the same, albeit the threshold value of 0.5 is slightly more accurate.

We can later use this knowledge to update our threshold in the deployment workflow if we see a more significant difference in accuracy, but for our case, the differences are negligible.

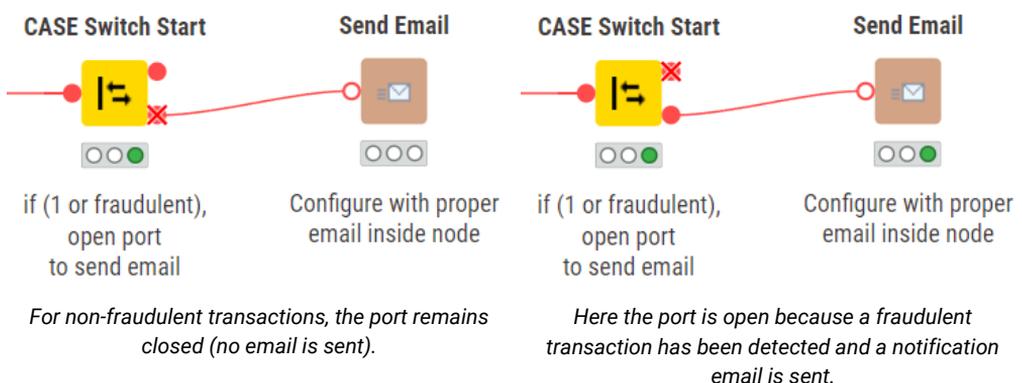
The new transaction being read in is passed into the trained model and a threshold is assigned to the transaction. For our case, we use a threshold of 0.3 to classify whether or not the transaction is fraudulent or not.

As shown in the tables above, we did not see a significant difference between the two thresholds.

- A **higher threshold** means the model needs more confidence before labeling a transaction as fraudulent, making it more cautious about classifying transactions.
- A **lower threshold** makes the system more likely to identify a transaction as fraudulent, increasing the number of transactions flagged as suspicious.

This adjustment in the threshold helps us find a balance between catching frauds and reducing false positives. The new transaction is classified as “not fraudulent” or “good”, so our switch statement closes the port to the email.

Below are two snippets of what can be expected to occur depending on the classification of the transaction. On the left, we have a non-fraudulent transaction. The port remains closed so that no notification email is sent. And on the right, the port opens up if the transaction is fraudulent or “1”.



KNIME for Finance: Early Fraud Detection

In helping to detect credit card fraud, KNIME Analytics Platform provides a solution using a low-code, visual, and intuitive user interface. We could schedule the deployment workflow to run on intake of any new transactions at a specific time of day – or if we wanted to check only large transactions, we could run the deployment workflow as is and have it check on any new transaction passed in for classification.

[Download KNIME](#) and explore more finance solutions in [KNIME for Finance](#).

Fraud Detection using Quantiles

Author: Thor Landstrom

Workflows on KNIME Community Hub: [01_Training Quantile Method for Fraud Detection](#) & [02_Deployment Quantile Method for Fraud Detection](#)

Fraud patterns are continually evolving, making it increasingly challenging to keep pace with the sheer volume of transactions. In this article we want to look at quantiles as a simple and quick solution for identifying suspicious transactions and give you a solution in KNIME that is able to analyze large volumes of transaction data quickly.

The previous subchapter as well as this subchapter both look at two types of fraud detection: classic machine learning based predictions – when your dataset contains enough fraud examples (e.g., Random Forest) – and outlier detection-based techniques – when your dataset doesn't contain enough fraud examples. Here we're going to use quantiles to detect the outliers or the suspicious transactions in our dataset of credit card transactions.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Fraud Detection: Quantiles](#) video to get an overview of this solution or browse the playlist for other solutions.

What are Quantiles?

Quantiles are values that divide a dataset into equal-sized intervals, helping to understand the distribution of the data. By examining quantiles, we can identify what constitutes "normal" data. In the context of credit card transactions, quantiles help establish the typical range of legitimate transactions. Transactions that deviate significantly from this range can be flagged as suspicious or potentially fraudulent, as they are considered outliers. While quantiles are typically aimed at linearly distributed datasets, we will demonstrate that they can also perform well on highly skewed datasets, such as the one in our use case.

Why KNIME for Quantile Calculations?

While quantile calculations can be performed in Excel, the advantage of using KNIME Analytics Platform is that as a data science tool, KNIME is designed to handle large datasets efficiently. You can also use it to automate data intake from various sources, reducing the need for manual intervention, and its visual, low-code environment makes it accessible for financial teams without expertise in coding.

Let's get started!

The Task: Identify fraudulent Transactions using Quantiles

Credit card transactions fall into two primary categories: legitimate and fraudulent. The challenge lies in accurately identifying and flagging fraudulent transactions to ensure minimal false positives.

Fraud detection typically involves a mix of manual and automated steps for transaction patterns, customer behavior, and other pertinent factors. For our focus, we will concentrate on the automation aspect by training a model on a labeled dataset and applying it to new transactions to simulate incoming data from an external source.

We utilize a well-known dataset from Kaggle named "Credit Card Fraud Detection". This dataset consists of real, anonymized credit card transactions made in September 2013 by European cardholders. It contains 284,807 transactions over two days, including 492 fraudulent transactions. This dataset is characterized by a significant class imbalance, with fraudulent transactions ("fraud") making up only 0.172% of the data, compared to the vast majority of legitimate transactions ("good").

The dataset contains 31 columns:

- V1 – V28: numerical input variables from a [PCA \(Principal Component Analysis\) transformation](#)
- Time: seconds elapsed from current transaction to first transaction
- Amount: transaction amount
- Class: "1" means fraud, "0" means good/other

A key feature needed for our training is "Class" as we can use it to score the performance of the quantile-based classification method.

The process for creating our classification model follows the steps below. Even if there is data coming from multiple sources, the overall process does not change:

- Create/import a labeled training dataset
- Z-score normalize the data

- Remove non outliers based on defined quantile scalar
- Mark outliers on input data
- Evaluate model performance
- Import the new, unseen transactions
- Deploy the model and feed the new transactions in
- Notify if any fraudulent transactions are classified

The Workflows: Training a Outlier Detection Model to identify fraudulent Transactions with Quantiles

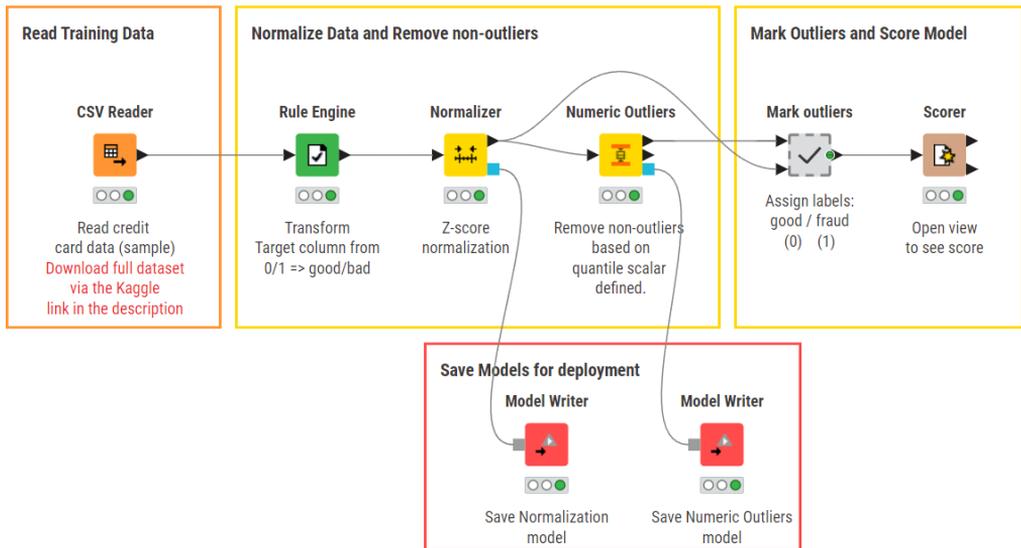
Note. All workflows used in this article are available publicly and free to download on the KNIME Community Hub. You can find the workflows in the KNIME for Finance space under [Fraud Detection](#) in the [Quantile](#) section.

The first workflow covers training our model. You can view and download the training workflow [Quantile Method Training](#) from the KNIME Community Hub. With this workflow you can:

- **Read training data** from a specified data source. In our case, we use data from the Kaggle dataset previously mentioned.
- **Normalize the data** by applying a z-score normalization on all numeric data columns.
- **Remove non-outliers** by using the *Numeric Outliers* node based on the defined quantile scalar.
- **Mark outliers** based on the result from the *Numeric Outliers* node
- **Evaluate model results** by opening the view of the *Scorer* node to check overall accuracy of the numeric outlier model.
- **Save the models for deployment** in the next workflow if you are satisfied with the performance.

Audit & Compliance

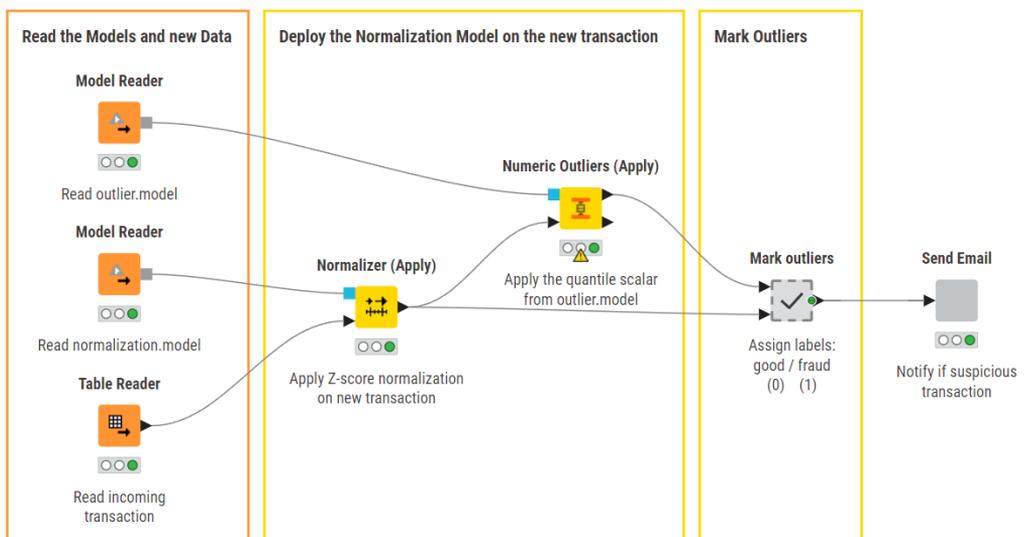
Fraud Detection using Quantiles



KNIME workflow to train an outlier detection model to identify fraudulent transactions.

In our second workflow, [Quantile Method Deployment](#), you can:

- **Read the previously saved models and new data** for classification
- **Apply the models on the new transaction** after normalization of the new transaction
- **Mark outliers**
- **Send an Email** to notify if a transaction is fraudulent

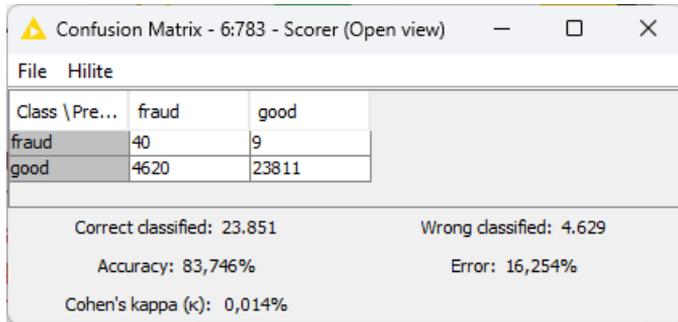


KNIME workflow to deploy an outlier detection model to identify fraudulent transactions.

Inside the “Send Email” component, we check whether the transaction is fraudulent or not. If it is, an email is sent to the specified person for follow up.

The Results: A Model for classifying Transactions with Quantiles

Let's go back to the first workflow for a moment. From the training workflow, the results we get by opening the view of the Scorer node is shown below.



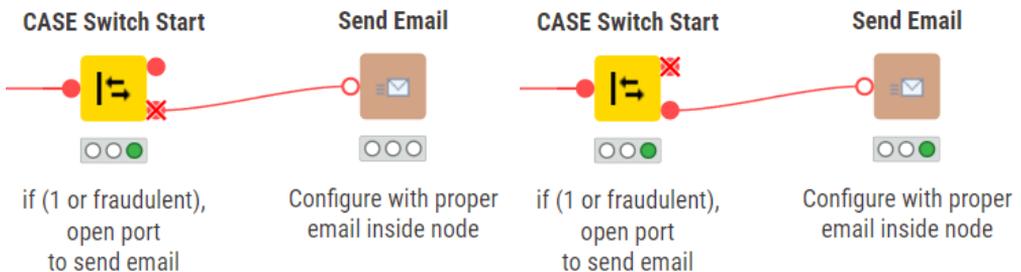
Class \ Pre...	fraud	good
fraud	40	9
good	4620	23811

Correct classified: 23.851 Wrong classified: 4.629
Accuracy: 83,746% Error: 16,254%
Cohen's kappa (κ): 0,014%

Evaluation of the quantile-based Fraud Detection model.

We have a confusion matrix that outlines statistics of the quantile-based method of classification. We can see that the overall accuracy came out to 83.7 %. In the context of the alternative methods available, this method does not perform up to par. For example, the [Random Forest](#) method yielded a much better accuracy. Nevertheless, this type of performance is to be expected as quantile-based classification is one of the simple implementations of anomaly detection that works well for linear or evenly distributed data. As the method we use is sensitive to distributions in the data, our dataset used from Kaggle is highly skewed with a very small percentage of fraudulent transactions which is one of the main reasons we see this relatively low performance. Although not the best, the quantile method offers a quick and easy way of classifying data which can potentially be used for quick preprocessing or for linear data use cases.

The new transaction being read in is normalized and applied into the saved models from the training workflow. The new transaction is classified as “not fraudulent” or “good”, so our switch statement closes the port to the email.



For non-fraudulent transactions, the port remains closed (no email is sent).

Here the port is open because a fraudulent transaction has been detected and a notification email is sent.

Above we have two snippets of what can be expected to occur depending on the classification of the transaction. On the left, we have a non-fraudulent transaction, and on the right the port will open up if the transaction is fraudulent or “1”.

Why KNIME for Finance

KNIME provides finance teams with easy access to advanced data science techniques tailored for managing large data volumes efficiently. Utilizing KNIME’s intuitive interface, teams can collect, clean, and analyze data from both new and legacy sources, automate tedious data processing tasks, and develop advanced models for fraud detection. Explore more finance solutions on the KNIME for Finance space on KNIME Community Hub.

Introducing AI in Finance Departments

Author: Roberto Cadili

Workflows on KNIME Community Hub: [01_DataViz, IQR & GenAI for Fraud Detection](#) & [02_Deployment DataViz, IQR & GenAI for Fraud Detection](#)

Contract frauds involve deceitful practices in the creation, execution, or enforcement of agreements, aiming to gain unfair advantages or financial benefits. Possible fraud types in contracts include misrepresentation, where false or misleading information is provided; forgery, involving falsified signatures, investment sums or documents; and Ponzi schemes, where returns are paid from new investors' money rather than profits.

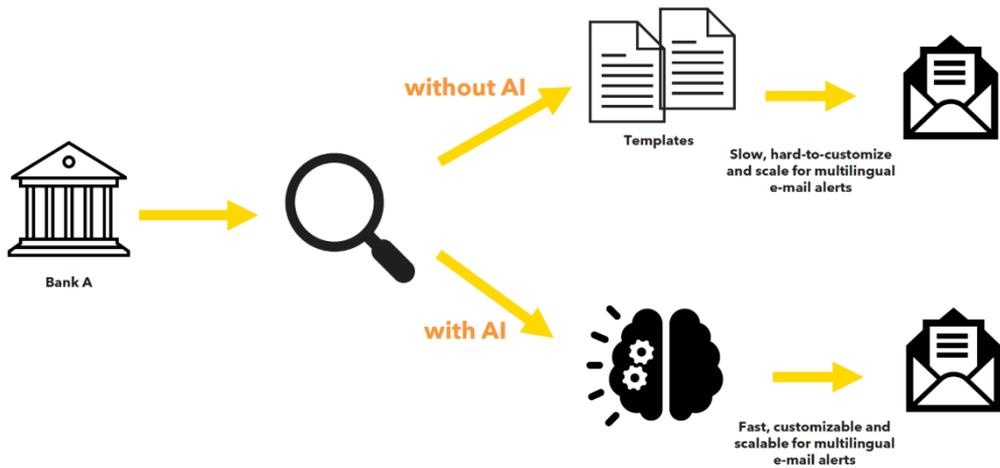
In this article, we'll explain how GenAI can help finance departments automate creating custom multilingual alerts, after fraudulent contracts are identified using visualizations and statistical measures.

How can GenAI help Finance Departments?

Large Language Models (LLMs), with their ability to understand human languages and generate coherent responses, can help boost productivity and enable new levels of automation and personalization.

In the context of contract frauds for auditing purposes, GenAI can help automate the creation of custom fraud alerts in different languages to effectively address a multilingual audience. Traditionally, before AI, a financial institution with a multilingual customer base would rely on human-translated templates. This process works but is slow, time-consuming, hard to customize for individual customers and does not scale well every time a new language is requested.

With the help of GenAI, these limitations are significantly reduced. Once suspicious contracts are identified, a financial institution can parameterize a prompt with customer details (e.g., name, surname, language, etc.) and query an LLM to generate alerts in seconds. Not only is this process fast and scales well for many different languages, but it can be further customized to include contract details and definitions using *Retrieval Augmented Generation*.



Creation of fraud alters with vs. without AI.

Finance departments can easily implement GenAI-driven solutions in KNIME Analytics Platform using the nodes of the [KNIME AI Extension](#).

However, before we can leverage GenAI effectively to signal suspicious activities, contract frauds need to be identified. One of the major challenges here lies in the availability of datasets containing fraud examples. When the dataset contains enough fraud examples, a [machine learning-based approach](#), for example, enables us to conduct a multidimensional analysis to identify fraudulent contracts, leading to better accuracy.

However, in reality, datasets containing many fraud examples are rare. This forces the adoption of different strategies, such as identifying fraudulent contracts using [outlier detection-based strategies](#) with visual or statistical techniques (e.g., [quantiles](#) or IQR).

Visual and Statistical Techniques for Fraud Detection

Data visualization simplifies identifying fraudulent contracts by transforming complex datasets into intuitive formats, making it easier to spot anomalies from what constitutes "normal" data. Visualizations are also very useful when fraud examples are scarce and we need to present our findings in a way that is aesthetically pleasing and easy to interpret. Common visualizations include pie/bar charts for comparing categories, scatter plots/bubble charts for relationships among numeric columns, histograms/violin plots for data distribution, and box plots for central tendency and variability. To facilitate the task, the creation of a [comparative dashboard using KNIME components](#) can provide an interactive and customizable overview of complex data relationships.

However, using visualizations requires the finance department to manually check every plot, hindering automation efforts. **Statistical techniques**, such as the

interquartile range (IQR) can come to the rescue. This is a simple, nonparametric outlier detection method in a one-dimensional feature space. To calculate the IQR, the data set is divided into quartiles, i.e., Q1 (lower quartile), Q2 (median), and Q3 (upper quartile), where $IQR = Q3 - Q1$. In addition to that, we define k , the interquartile range multiplier. This parameter is usually set to 1.5, and defines how sensitive the outlier detection will be.

An outlier is then a data point x that lies outside the interquartile range.

$$x > Q3 + k * IQR \text{ or } x < Q1 - k * IQR,$$

This outlier treatment can easily be implemented in KNIME Analytics Platform using the *Numeric Outliers* node.

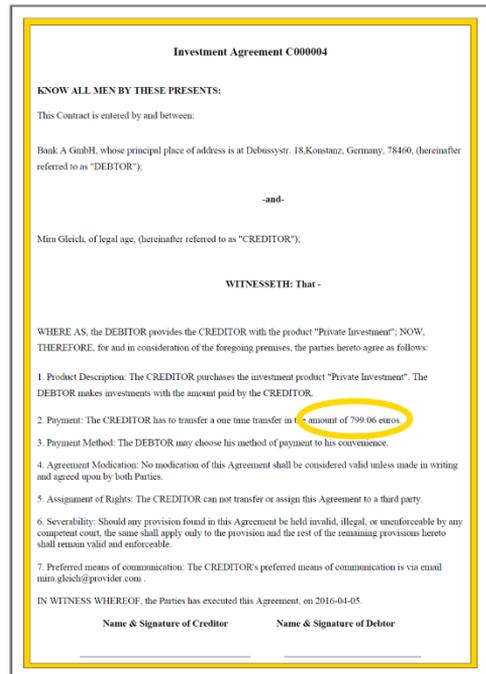
Note. We experimented using GenAI also for the detection of fraudulent contracts. However, the experiment created concerns over the correctness of results, the interpretability of the detection process, and the cost in human labor for prompting, and data pre- and post-processing.

The Task: Identify fraudulent Investment Sums and generate Alerts

Among contract types, investment agreements are particularly prone to fraud. These contracts often involve complex terms, significant financial stakes, and promises of high returns, making them attractive targets for fraudulent activities. Typically, fraudsters try to forge signatures and personal details, alter contract terms, or fake transaction sums, exploiting investors' lack of expertise and trust.

In today's task, we'll act as a financial institution and concentrate on the identification of fraudulent investment agreements, focusing on the investment sum stated in the documents. The challenge lies in accurately flagging fraudulent amounts using data visualization and the IQR method, and leveraging GenAI to create customized, multilingual alerts. Lastly, the solution should be deployed as an on-demand web-based application.

The dataset contains 100 investment agreements in PDF format. Each agreement has a unique ID and reports the terms and object of the investment, including the product name, the investor's name, email address, date and investment sum.



A sample investment agreement.

A second dataset, stored in an SQLite database, contains investors' information, such as name, email address, and language, as well as details about the type of investment agreement.

The process involves five steps:

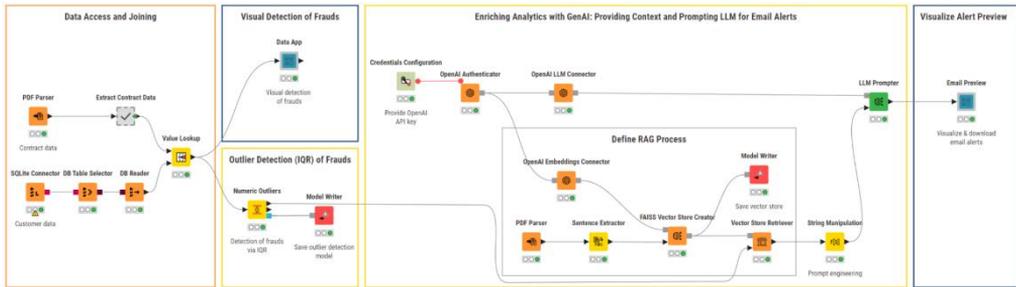
- Access, parse and join data sources
- Identify outliers using visualizations and the IQR
- Create custom alerts in different languages with GenAI
- Deploy the solution as a web-based application

The Workflows: Use Visualizations and IQR to identify Frauds and create Alerts with GenAI

All workflows used in this blog post are available publicly and free to download on the KNIME Community Hub. You can find the workflows on the [KNIME for Finance](#) space under Fraud Detection in the "[Visualizations and IQR](#)" section.

The first workflow, "[01_DataViz, IQR & GenAI for Fraud Detection](#)", covers the detection of frauds and the generation of alerts. You can view and download the workflow from the KNIME Community Hub.

Audit & Compliance Introducing AI in Finance Departments



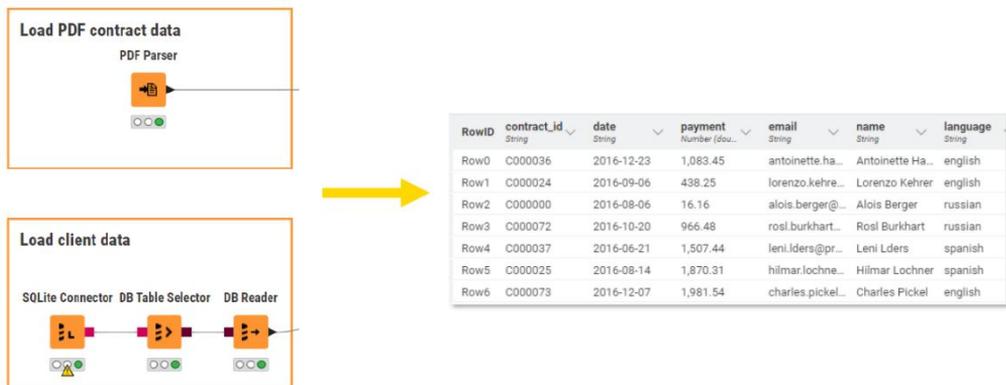
KNIME workflow to detect frauds visually and statistically, and to create alerts with GenAI.

Access, Parse and Join Data Sources

We start off by importing investment agreements using the *PDF Parser* node. Next, we parse the documents to extract key information, such as agreement IDs, the investors' name, email, date, and invested sum.

In the lower workflow branch, we connect to an SQLite database, select the *customer_table* and import the data containing investors' personal details, including language, email and agreement type.

Using a *Value Lookup* node on the email column, we join the two data sources.



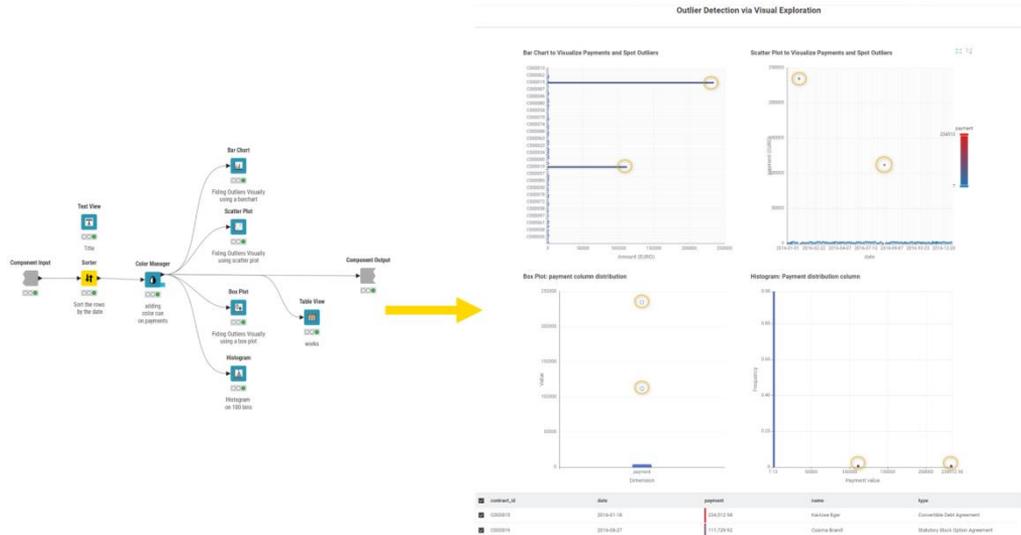
Accessing and joining data sources.

Identify Outliers using Visualizations and the IQR

To detect fraudulent contracts with data visualizations, we rely on the creation of a comparative dashboard using KNIME components.

Components in KNIME are custom nodes that bundle specific functionalities, can have their own configuration dialog and composite views. The latter feature facilitates the creation of customizable, interactive dashboards with charts, tables, and widgets. Users can also design visual layouts, edit HTML content, and generate reports.

We assign color cues to investment sums, and visualize them using a bar chart, a scatter plot, a box plot and a histogram. When wrapped in a component, plots can propagate the selection of one data point across all plots. In this way, we can clearly see the data points that strongly deviate from the rest (marked in yellow), and inspect the details.



Inside the “Visual Detection of Frauds” component (left), and the resulting dashboard where outliers are marked in yellow (right).

The second strategy to detect contract frauds uses the IQR method. The KNIME implementation is straightforward and requires only one node: the *Numeric Outliers*.

In the node configurations, we select the payment column, define k, and remove all data points that are not outliers. This choice of treatment is useful to isolate those investors that need to be alerted. Lastly, we save the model for deployment.

Create custom Alerts in different Languages with GenAI

Once fraudulent agreements are identified, we can leverage GenAI to streamline the creation of multilingual alerts without relying on time-consuming templates. Additionally, we want our alerts to include agreement details, as well as explanations of the agreement type as defined by the knowledge base of the financial institution.

To do that, we’ll rely on the nodes of [KNIME AI Extension](#). The overall approach can be summarized in four steps: **Authenticate – Connect – Customize – Prompt**.

After authenticating to the AI provider, for example OpenAI, we connect to the LLM of choice. In our case, gpt-3.5-turbo.

In the lower workflow branch, we use the *PDF Parser* to access the knowledge base of the financial institution, where agreement types are described. With this document, we

define a RAG process, aimed at customizing the model responses and enriching the content of the alerts with context. To do that, we split the document into sentences, embed them and store the embeddings in a FAISS Vector Store, which we save for deployment.

Next, we use the *Vector Store Retriever* node to perform a similarity search between the agreement types in the investor database and the institution’s knowledge base containing the descriptions of each type of agreement. The goal is to retrieve the most similar descriptions from the vector store.

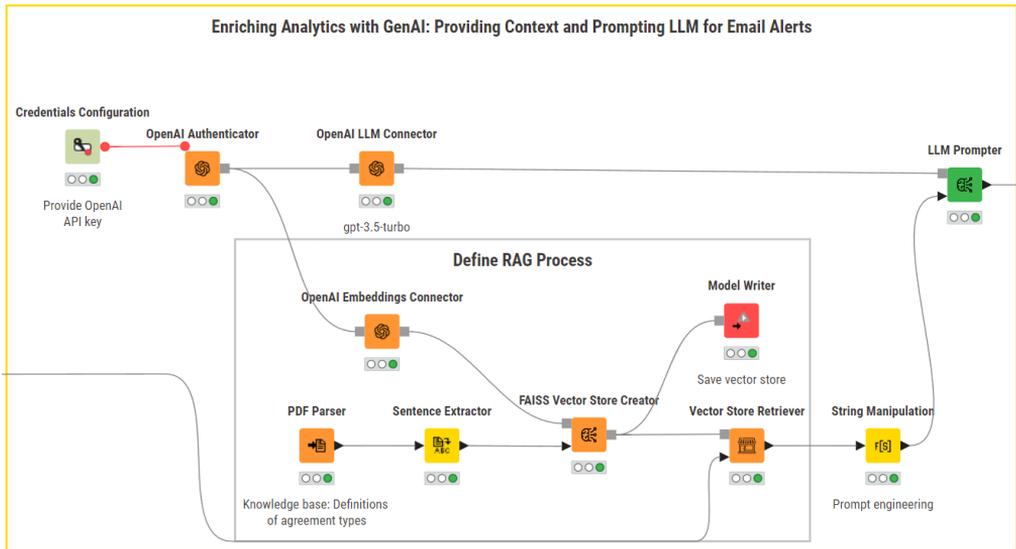
Once similar documents are retrieved, we engineer a prompt using the String Manipulation node. In the interest of automation, we parameterize the prompt using values of the detected outliers (i.e., language, investor name, agreement type, date and sum) and augment it with details about the type of agreement signed by the investors.

#	RowID	prompt
1	Row...	Write an email in english to this customer Cosima Brandl with the following requirements: The email object must alert that suspicious activity was detected on the contract Statutory Stock Option Agreement (C000019) dated or 2016-08-27 with payment amounting to 111729.92 in the local currency. Describe then the type of contract 'Statutory Stock Option Agreement' by using the following information : {Statutory Stock Option Agreement Statutory stock options are often referred to as Incentive Stock Options or Qualified Stock Options. Nonstatutory Stock Option Agreement A nonstatutory stock option is also referred to as Nonqualified Stock Options. Statutory stock options have tax benefits but some with strict requirements that cost the company more up front.} Investigation is required as soon as possible. Require the customer to reply to this email to take action. Sign the email by 'Bank A GmbH', and include the bank's principal address below the signature: Debussysr. 18, Konstanz, Germany, 78460

Prompt to generate alerts: yellow-marked elements are the result of prompt parameterization using values of retained outliers. In red, the retrieved agreement descriptions form the vector store.

Lastly, we feed the prompt to the LLM Prompter node to generate personalized multilingual alerts. We can inspect and download the generated text in the composite view of the “Email Preview” component.

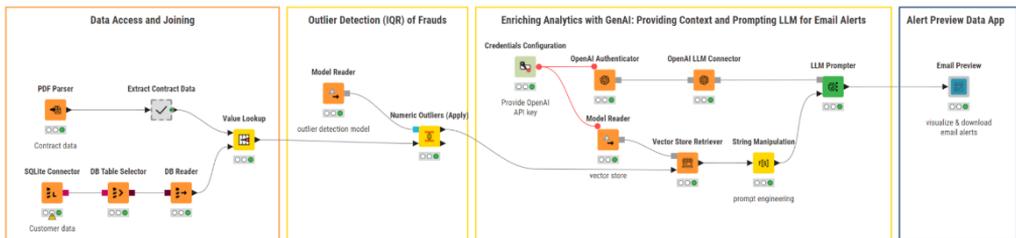
Audit & Compliance
Introducing AI in Finance Departments



Workflow segment to create multilingual alerts enriched with custom context via RAG.

Deploy the Solution as a web-based Application

The second workflow, "[02_Deployment DataViz, IQR & GenAI for Fraud Detection](#)" deploys the application as a data app that can be consumed on-demand on a web browser.



Deployment workflow to detect frauds statistically and create alerts with GenAI.

The deployment workflow follows a very similar design.

The key difference in this workflow lies in removing the identification of outliers with visualizations, as this method requires manual inspection and cannot be automated in production. Additionally, we rely on the Model Reader nodes to re-use the outlier detection model and the vector store that were created in the previous workflow.

Lastly, we exploit another feature of KNIME components. The workflow and the "Email Preview" component can be deployed as an on-demand [data app](#) on the [KNIME Business Hub](#) and define interactive pages in web applications for ease of consumption and interaction.

Audit & Compliance

Introducing AI in Finance Departments

Deployment as a Data App on the KNIME Business Hub.

The Results: Detected Frauds and multilingual Alert Preview as a Data App

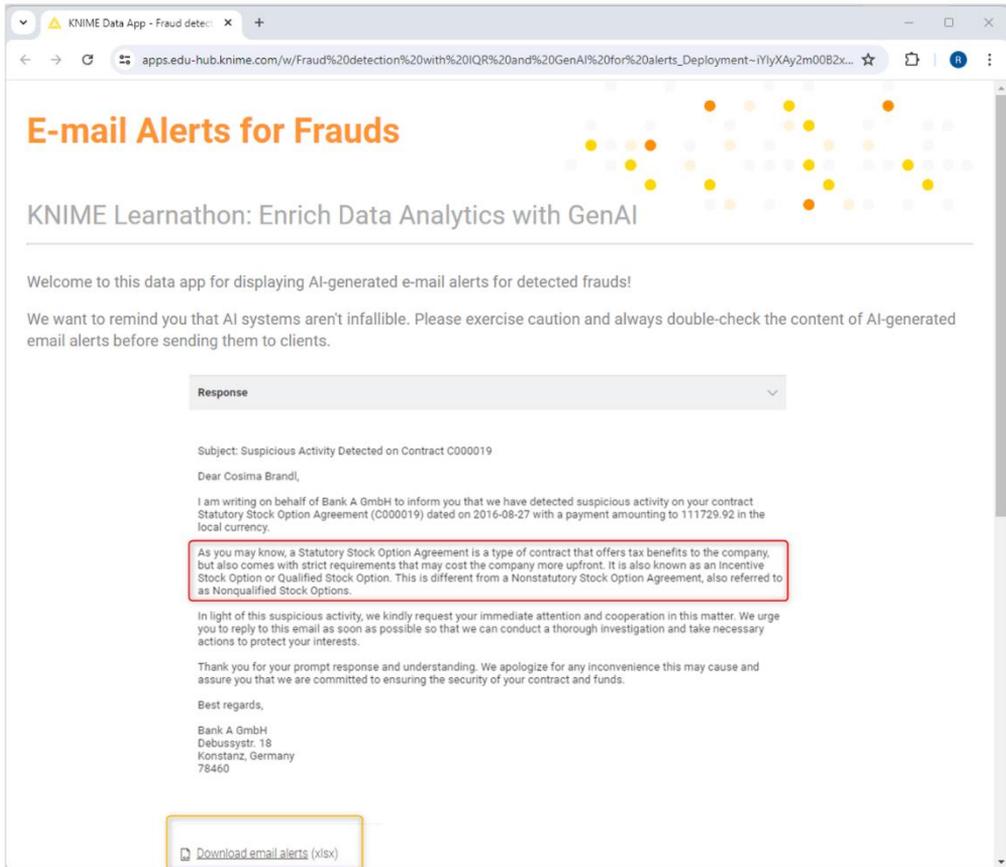
The techniques illustrated above consistently identify two outliers, whose investment sums are considerably higher than the rest.

<input checked="" type="checkbox"/>	contract_id	date	payment	name	type
<input checked="" type="checkbox"/>	C000015	2016-01-18	234,512.98	Kai-Uwe Eger	Convertible Debt Agreement
<input checked="" type="checkbox"/>	C000019	2016-08-27	111,729.92	Cosima Brandl	Statutory Stock Option Agreement

Investors at risk of fraud.

For these investors, custom email alerts in different languages are generated and displayed as a Data App. Here, we display only the English version. The alert warns the recipient of suspicious activities and requests her to immediately contact the financial institution. With the help of GenAI, the generated response is enriched to include also an explanation of the agreement type (circled in red). The content of the alert can be downloaded and further edited before notifying the investors.

Audit & Compliance Introducing AI in Finance Departments



Alert preview on the KNIME Business Hub with the option to download it.

KNIME for Finance: Scale Fraud Detection with Stats and GenAI

In helping detect contract frauds with different techniques, KNIME Analytics Platform provides a flexible and automated solution using a low-code, visual, and intuitive user interface.

The introduction of GenAI to enrich the analytical process helps scale the creation of custom alerts in multiple languages, ensuring timely, reliable and cost-effective notifications.

[Download KNIME](#) and explore more finance solutions in [KNIME for Finance](#).

Financial Planning & Analysis

With KNIME, financial planning professionals can make more accurate forecasts and budgets, automate repetitive tasks, reduce human error, and expedite reporting. This enables them to focus on analysis that matters, e.g., understanding revenue, spotting cost reduction opportunities, and identifying profitability drivers. In this chapter, we introduce some use cases for planning, forecasting, scenario modeling, budgeting, and performance reporting.

This chapter includes the articles:

- Budget Monitoring, p. 48
- 5 Financial Metrics every FP&A Analyst must know about, p. 53
- Monthly and Year-to-Date Revenue Aggregations, p. 57

Budget Monitoring

Author: Rosaria Silipo

Workflow on KNIME Community Hub: [Budget Monitoring Data App](#)

One of the main tasks of the finance department is to prepare the budget overview. At the beginning of the year budgets are created for the various products, divisions, departments, or projects to last the whole year. Money is assigned according to budgets at the beginning of the year, spent during the year, and throughout the year the finance department must make sure that money spent does not exceed money assigned.

Let's look at how to calculate it in KNIME and have a reusable solution that you can use next year, too.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Budget Overview](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Budget Monitoring

The calculation is easy and deals with:

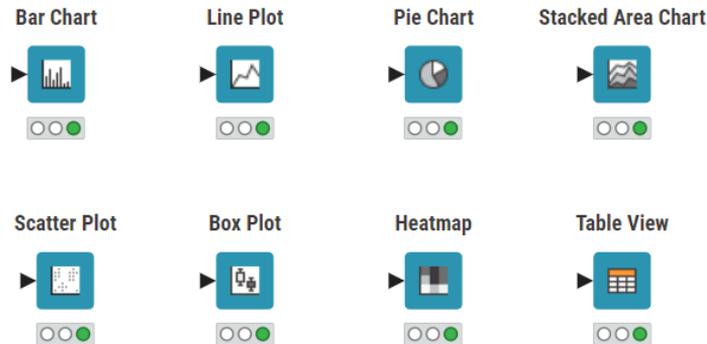
- Money assigned: A lump sum of money established for a project/department/etc. at the beginning of the year.
- Money used: The sum of the expenses throughout the year.
- Money remaining: The difference between the money assigned and the money spent at each point in time.

It's important to visualize budget monitoring to see identify where things are going well (or not). Here are ideas for three possible visualizations.

- Total budget overview (total money assigned, total money used, and total money remaining) across all projects.
- Single budget overview for each project. If there was a minus in the budget management, which project/department was responsible for it? In general, even in more favorable conditions, it is useful to know how the budget management went for the different projects.

- Another interesting overview refers to the trends over time. Has the total assigned budget been reduced across the years? Has budget management improved over time, i.e. years, months, quarters?

KNIME offers the ability to create many different kinds of visualizations.



An excerpt from the KNIME Data Visualization cheat sheet showing common visualization techniques and their KNIME nodes.

Let's take an example with 11 projects, identified by desert names and running across three years, from 2021 to 2023. For each project, we will import from our accounting system the money initially assigned and the total money used throughout each year. See an example in the screenshot below, where money is expressed in 1000 units.

The screenshot shows a KNIME Table node with 132 rows and 7 columns. The data is as follows:

#	RowID	name	color	ranking	money assigned (k)	money used (k)	reference year	Quarter
1	Row0	Sahara	green	1	365	420	2023	Q1
2	Row1	Mojave	black	19	520	510	2023	Q1
3	Row2	Kalahari	pink	7	260	252	2023	Q1
4	Row3	Blue	blue	2	400	410	2023	Q1
5	Row4	White	white	6	300	279	2023	Q1
6	Row5	Gobi	yellow	5	435	435	2023	Q1

An example of a budget balance spreadsheet.

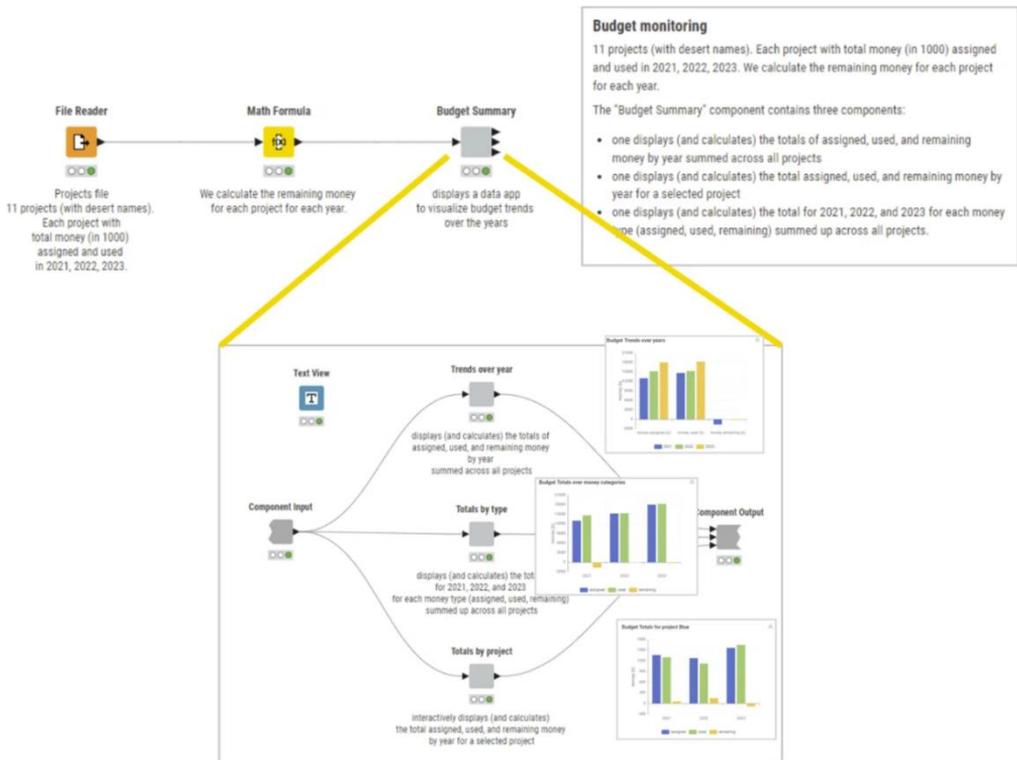
The Workflow: Budget Monitoring Report

The workflow [Budget Monitoring Report](#) for this use case is available and free to download from the KNIME Community Hub and includes the following steps:

- **Read budget data** – *money assigned* and *money used* for each project – from an ERP system, an Excel file, or a CSV file with the appropriate reader node. In our example, we used a CSV file and a *File Reader* node.
- **Calculate the remaining money** as (*money assigned* – *money used*) for each project.

Financial Planning & Analysis Budget Monitoring

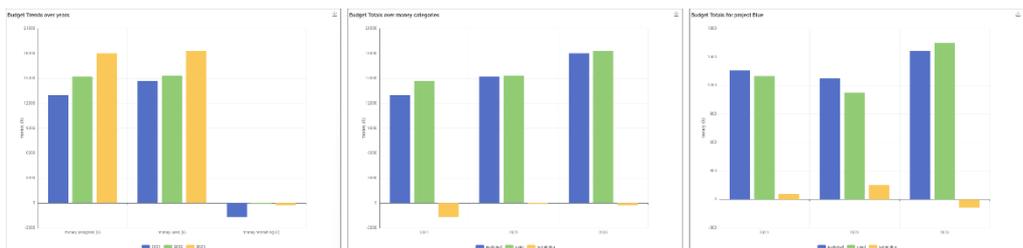
- **Visualize the data**



The workflow, Budget Monitoring Report, to visualize budget amounts.

To visualize the data, we chose the following three options:

- The trends across years for money assigned, money used, and remaining money (left)
- The total amounts for all projects year after year (center)
- The total amounts for a single project year after year (right). For this chart we added one control item for project selection.



The bar charts from the "Budget Monitoring Report" workflow display project budgets from different perspectives.

The Results: Budget Monitoring in a live Data App

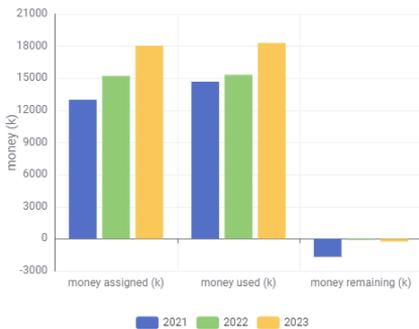
Let's have a look at the results on the data app (see the figure below). In the first bar chart (top left), we see that the budget amount assigned to projects in total has grown considerably year after year, and so has grown the amount used.

In the second chart (top right), we see that in the first year (2021), there has been a miscalculation about the money effectively needed for the various projects and therefore there was a negative amount in the difference between assigned money and used money. The finance department has probably learned from that experience and assigned more money to some projects in the upcoming years.

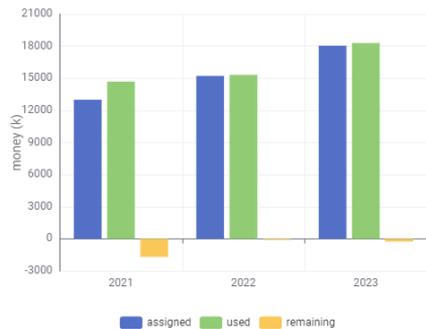
Finally in the lower part of the data app, we can observe the budget balance of each single project across the years. For example, which project was responsible for the negative amount of remaining money in 2021? By selecting a different project in the box on the left, we can see the balance situation in the bar chart on the right. While project *Blue*, *White*, *Kara Kum* have been quite virtuous in spending the budget money in 2021, not the same can be said for the other projects. Most of them, however, have learned over the next two years and asked for and used a proper budget.

Project Budget Summary

Budget Trends over years



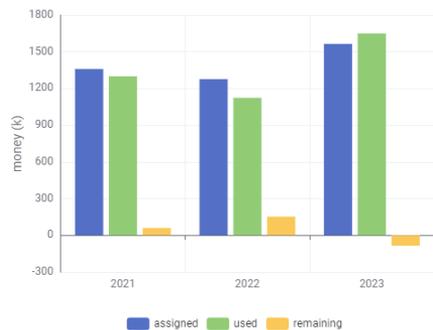
Budget Totals over money categories



Select Project:

Blue
Sahara
Mojave
Kalahari
Blue
White
Gobi
Kara Kum

Budget Totals for project Blue



The data app displays the bar charts for the budget monitoring from different perspectives.

KNIME for Finance

The intuitive low-code environment of KNIME Analytics Platform enables non-technical users to get started easily with data science and benefit from techniques that enable you to automate repetitive tasks, reduce human error, and expedite reporting. Explore more use cases for [KNIME for Finance](#) on KNIME Community Hub.

5 Financial Metrics every FP&A Analyst must know about

Author: Cristian Rastsanu, Mydral

Workflows on KNIME Community Hub: [Financial Metrics - IRR, MIRR & XIRR](#) & [Financial Metrics - NPV & XNPV](#)

Timely access to accurate financial indicators is paramount for informed decision-making. However, the manual calculation of these indicators often proves to be a bottleneck, introducing inefficiencies, inconsistencies, and delays. FP&A analysts are tasked with bridging the gap between raw data and actionable insights and need efficient solutions to streamline the process.

The ability to use key financial metrics is a crucial skill that enables FP&A analysts to make informed decisions quickly and adapt with agility to market fluctuations. Automating financial calculations not only saves time but also enhances accuracy, providing a strategic edge to businesses.

[Mydral](#), an organization with expertise in consulting and specialized in data visualization and understanding, has developed models using the low-code KNIME Analytics Platform to automate the financial calculation of the five essential metrics every FP&A analyst needs:

- [Internal Rate of Return \(IRR\)](#)
- [Modified Internal Rate of Return \(MIRR\)](#)
- [Extended Internal Rate of Return \(XIRR\)](#)
- [Net Present Value \(NPV\)](#)
- [Extended Net Present Value \(XNPV\)](#)

Let's delve into these five essential financial models, which are all available to download from the Financial Analysis space on KNIME Community Hub, and see how KNIME simplifies the process.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the videos [Financial Metrics: IRR, XIRR & MIRR](#) and [Financial Metrics: NPV & XNPV](#) to get an overview of this solution or browse the playlist for other solutions.

The Task: Evaluate Investment Feasibility with KNIME

KNIME Analytics Platform is a powerful tool that can be used to automate and optimize financial calculations. The task at hand involves automating the evaluation of investment feasibility, a crucial aspect of financial analysis. Traditional methods of assessing investment opportunities involve cumbersome manual calculations of metrics such as *IRR*, *MIRR*, *XIRR*, *NPV*, and *XNPV*. These calculations not only consume valuable time but are also prone to errors, potentially leading to flawed decision-making.

By leveraging KNIME's intuitive interface and comprehensive suite of tools, FP&A analysts can simplify and accelerate the process of investment feasibility analysis. The platform's drag-and-drop functionality allows analysts to build customized workflows tailored to their specific requirements. With nodes dedicated to each financial model, including *IRR*, *MIRR*, *XIRR*, *NPV*, and *XNPV*, KNIME facilitates the automation of these calculations with ease.

Furthermore, KNIME's integration capabilities enable seamless connectivity with various data sources, ensuring that the analysis is based on up-to-date and reliable information. Whether it's incorporating data from databases, spreadsheets, or external APIs, KNIME provides the flexibility to adapt to diverse data environments.

The Workflow: 5 Components to assess the Profitability and Viability of Investment Projects

There are different types of investments we can make:

- Personal investments on individual projects
- Regular financial investments, where we invest a regular sum of money e.g. once a month, and
- Random financial investments, which are usually riskier, non-regular investments

Let's look at the five components developed by *Mydral* to evaluate investment feasibility.

Use Net Present Value (NPV) and Extended Net Present Value (XNPV) in KNIME

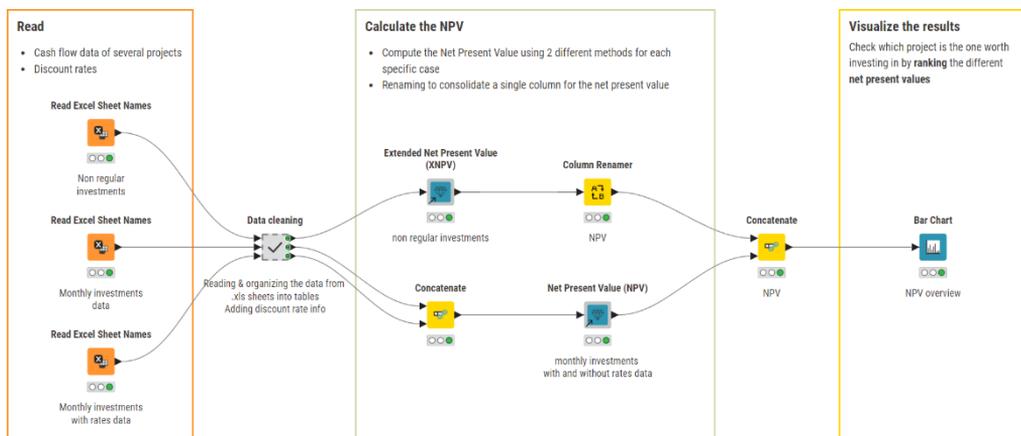
The [Net Present Value \(NPV\)](#) component helps you to evaluate an investment based on the highest value at a given rate. This calculation gives you the value of the expected

future income stream, determined as of the date of valuation. The *NPV* is typically used to evaluate personal investments and regular financial investments.

The [Extended Net Present Value \(XNPV\)](#) component is more suitable for evaluating random financial investments. This also calculates the *NPV*, but the calculation is based on specific dates.

You can use the *NPV* and *XNPV* components in your workflows. First read the cash flow data for the proposed investments, apply cleaning and standardization for each cash flow, then use the *NPV* component to calculate the metric. Finally, visualize the results.

Explore the example workflow, [Financial Metrics - NPV & XNPV](#), available for free on KNIME Community Hub, which demonstrates how to use the *NPV* and *XNPV* components.



An example workflow you can download from KNIME Community Hub to calculate Net Present Value and Extended Net Present Value metrics and visualize the results.

Use Internal Rate of Return (IRR), Modified Internal Rate of Return (MIRR), and Extended Internal Rate of Return (XIRR) in KNIME

Let's have a look at evaluating investment feasibility with the metrics *IRR*, *MIRR*, and *XIRR*.

The component for [Internal Rate of Return \(IRR\)](#) on an investment or project calculates the rate of return when the net present value of all cash flows (both positive and negative) from the investment is assumed to be equal to zero. This is useful for evaluating monthly investments where we don't need to specify any date.

The [Modified Internal Rate of Return \(MIRR\)](#) component calculates the IRR taking into account the finance rate and the reinvestment rate. This is useful for evaluating monthly investments with rates data.

The [Extended Internal Rate of Return \(XIRR\)](#) calculates the IRR at a specific date. This metric is typically used for evaluating random investments.

You can plug these components into your KNIME workflow, read the cash flow data for the investment, apply cleaning and standardization for each cash flow, and then apply the components to calculate the metric.

Explore the example workflow, [Financial Metrics – IRR, MRR & XIRR](#), available for free on KNIME Community Hub, which demonstrates how to use the IRR, MIRR and XIRR components.



An example workflow you can download from KNIME Community Hub to calculate Investment Return metrics and visualize the results.

The Results: Visualize your Investment Feasibility Evaluation

Now we have all return rates from all these investments. We put all of them together, and then we visualize them using a bar chart. And from the bar chart, it is immediately visible which investment project has the highest return rate. This makes the work of a financial analyst faster and easier.

Why KNIME for Finance

KNIME's low-code environment ensures that even users with limited programming experience can leverage its capabilities effectively. This democratization of financial analysis empowers FP&A teams to focus their efforts on interpreting results and deriving actionable insights rather than getting bogged down by manual calculations.

Monthly and Year-to-Date Revenue Aggregations

Author: Rosaria Silipo

Workflow on KNIME Community Hub: [Monthly and YTD Revenues](#)

Regular monthly and year-to-date (YTD) aggregations provide essential information to monitor a company's financial performance across time periods, evaluate the impact of specific events, and communicate the financial status.

In fast-paced business environments quick monthly or YTD aggregations provide up-to-date information and enable informed decisions in a timely manner. But it involves pulling together the data from multiple sources, which is complex and time-consuming. Delayed calculations can result in outdated or inaccurate information. Easy integration of multi-source data is pivotal to quick data aggregation. In this subchapter we show how you can quickly calculate monthly and year-to-date aggregations in KNIME.

The open source and free KNIME Analytics Platform lets you build visual workflows to automate and augment spreadsheet work. That means you can more efficiently combine and process data from numerous sources in a way that's repeatable. You spend less time on data aggregation.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Monthly and YTD Aggregations](#) video to get an overview of this solution or browse the playlist for other solutions.

Before we start, let's recap on what monthly revenue and year-to-date aggregation mean.

What is Monthly Revenue Aggregation?

Monthly revenue aggregation is the sum of all revenues for each month. It will show total sales revenue for each month and enable analysts to identify sales patterns, peak periods, and trends.

What is Year-to-Date (YTD) Revenue Aggregation?

Year-to-date (YTD) revenue aggregation is the sum of the monthly revenues from the beginning of the year till the current month. For example, if it is April, the YTD revenue is given by the sum of the monthly revenues of January, February, March, and April.

The Task: Calculate & Visualize Monthly and YTD Aggregates

Today's task is to display monthly and YTD revenue aggregations over months for two different years.

Let's imagine we're dealing with a restaurant business. Each evening, all dinners served and paid for are recorded. Up to 2022, these transactions were recorded in an Excel file. Starting from 2023, the recording system moved to a Google spreadsheet, so we'll need to access our data from both sources.

Let's break down the calculation into the processes we have to go through:

- Import restaurant transactions from Excel files and Google spreadsheets for 2022 and 2023
- Aggregate amounts by month to create monthly revenues
- Calculate year-to-date revenues for each month
- Visualize monthly revenues over months for year 2022 and 2023 in a bar chart, side by side
- Visualize YTD revenues over months for both years 2022 and 2023 in a line plot to provide an easy way to compare over years, if required.

The Workflow: Monthly and YTD Revenues

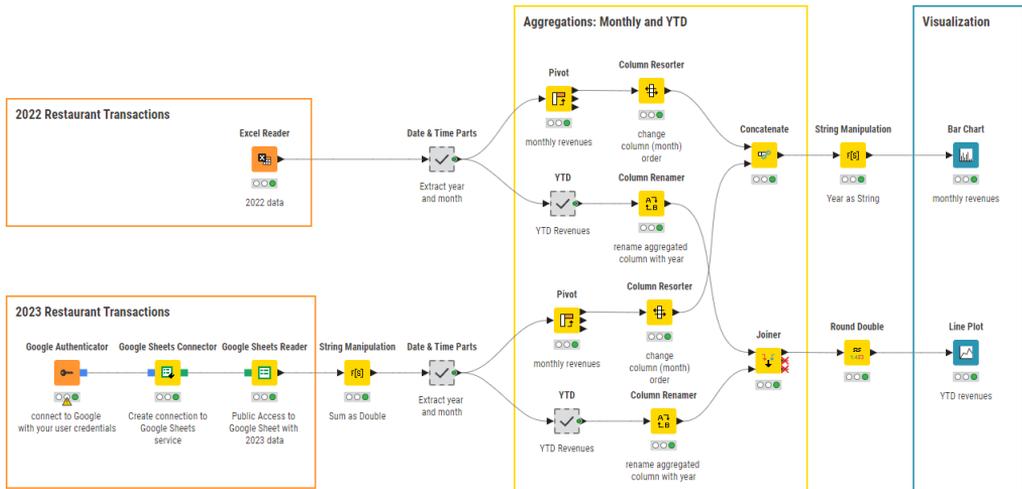
The workflow "[Monthly and YTD Revenues](#)" is available and free to download from the KNIME Community Hub.

Let's see how our above processes translate into our workflow, below. You can see that we have three main steps:

1. Retrieve the restaurant transactions for 2022 and 2023
2. Aggregate the monthly and year-to-date (YTD) revenues
3. Visualize the calculations in a bar chart and a line plot

Financial Planning & Analysis

Monthly and Year-to-Date Revenue Aggregations

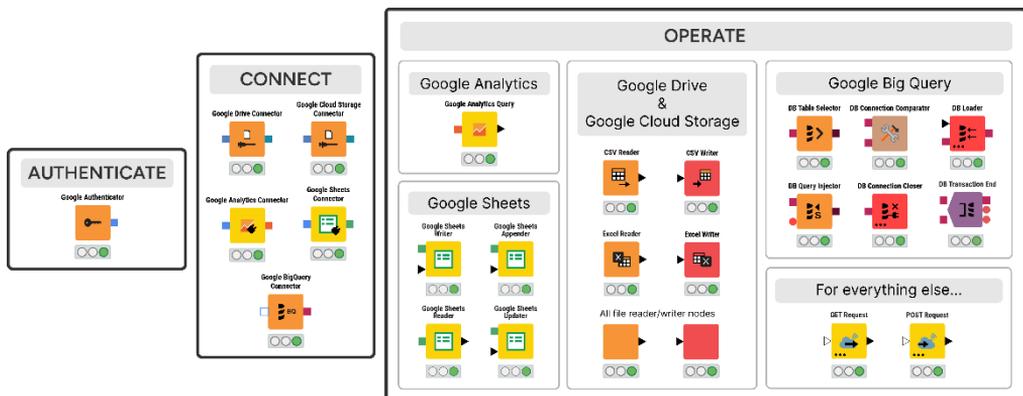


The KNIME workflow to calculate & visualize monthly and year-to-date (YTD) revenues.

Step 1: Retrieve Restaurant Transactions

- Import transactions for 2022 from an Excel spreadsheet with the *Excel Reader* node.
- Import transactions for 2023 from a Google spreadsheet with the sequence of nodes: *Google Authenticator* – *Google Sheets Connector* – *Google Sheets Reader*

Note. When you read in data from a Google spreadsheet you'll need a few additional steps, mainly for authentication. KNIME Analytics Platform offers a large number of nodes to integrate with Google resources. It all starts with the *Google Authenticator* node. This node gets you authenticated on Google and then allows you to access the Google services you have signed up for.



KNIME nodes to work with Google resources.

Financial Planning & Analysis

Monthly and Year-to-Date Revenue Aggregations

After authentication, KNIME Analytics Platform offers several connector nodes: to Google Analytics, to Google Sheets, to Google Big Query, to Google Drive, and to Google Cloud Storage. After connection, other nodes allow you to implement specific operations compatible with the selected service (see figure above).

Step 2: Aggregate monthly revenues and year-to-date revenues

Let's aggregate the **monthly revenues** first:

- **Aggregate** transactions by month to obtain monthly revenues. We could have used a *GroupBy* node here. However, in order to display the monthly values for different years side by side, the *Bar Chart* node requires a column for each month and a row for each year. This is the aggregated output format produced by the *Pivot* node.
- **Concatenate** monthly revenues for the years 2022 and 2023.

Table with 13 columns: RowID, Year, January, February, March, April, May, June, July, August, September, October, November, December. The table shows monthly revenue values for two rows representing the years 2022 and 2023.

#	RowID	Year	January	February	March	April	May	June	July	August	September	October	November	December
1	Row0	2022	111,634.24	96,943.45	111,076.17	109,456.38	109,154.04	106,484.99	110,131.7	112,901.89	107,812.15	111,576.75	112,592.49	108,985.45
2	Row...	2023	119,612.14	102,430.71	116,289.76	115,571.75	116,738.25	113,145.28	118,409.95	117,852.98	115,682.52	119,466.2	116,970.94	117,259.78

Data table with monthly revenues formatted to input the Bar Chart node.

And now we'll aggregate the **year-to-end revenues** for each month:

- **Aggregate** transactions as YTD. The output format required by the *Line Plot* node consists of one column for the months and one column for the corresponding YTD value. So, here, we first aggregate the transactions for each month with a *GroupBy* node and then we calculate the YTD value with the *Moving Aggregator* node, all inside the YTD metanode.
- **Join** YTD revenues for the years 2022 and 2023.

Table with 4 columns: #, RowID, 2022, 2023. The table shows YTD revenue values for 12 rows representing the months of the year for two years, 2022 and 2023.

#	RowID	2022	2023
1	January	111,634.24	119,612.14
2	February	208,577.69	222,042.85
3	March	319,653.86	338,332.61
4	April	429,110.24	453,904.36
5	May	538,264.28	570,642.61
6	June	644,749.27	683,787.89
7	July	754,880.97	802,197.84
8	August	867,782.86	920,050.82
9	September	975,595.01	1,035,733.34
10	October	1,087,171.76	1,155,199.54
11	November	1,199,764.25	1,272,170.48
12	December	1,308,749.7	1,389,430.26

Data table with YTD revenues formatted to input the Line Plot node.

Step 3: Visualize monthly and YTD revenues

- **Visualize monthly revenues** over months for 2022 and 2023 with a *Bar Chart* node, grouping by year
- **Visualize YTD revenues** over months for 2022 and 2023 with a *Line Plot* node. The months on are the x-axis and the YTD values are on the y-axis

Tips to improve Performance

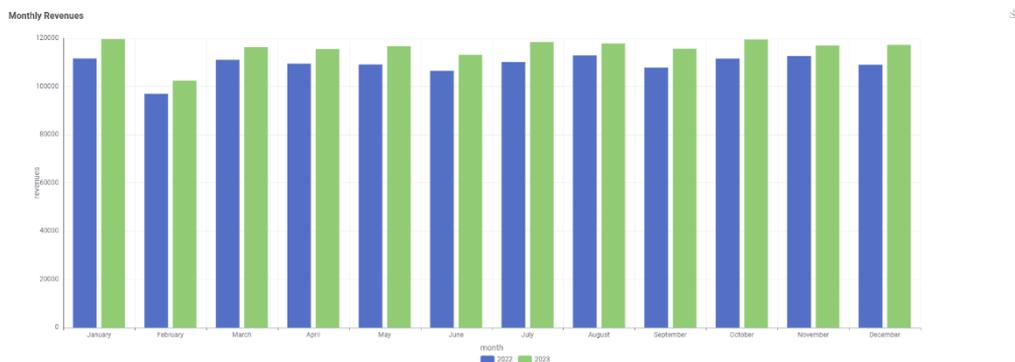
We could have performed all aggregations within the *Bar Chart* node. In this case, all totals are recalculated every time we open the bar chart, which can lead to very long loading time, especially for large amounts of data. This waiting time might be sub-optimal for the end user, even more so if the *Bar Chart* node is part of a data app. We opted for pre-calculated values before the visualization in the bar chart.

In the YTD values, after all the aggregations, there might be a long sequence of decimal digits. They do not carry much meaning beyond the second or third digit. So, we rounded the numbers up to the first two decimal digits.

In the *Bar Chart* node, group values can only be of type string. So, we transform the year values from number to string to perform the grouping in the node.

The Results: Identify Trends in Visualizations

Here's the bar chart showing the monthly revenues for the year 2022 and the year 2023 side by side.

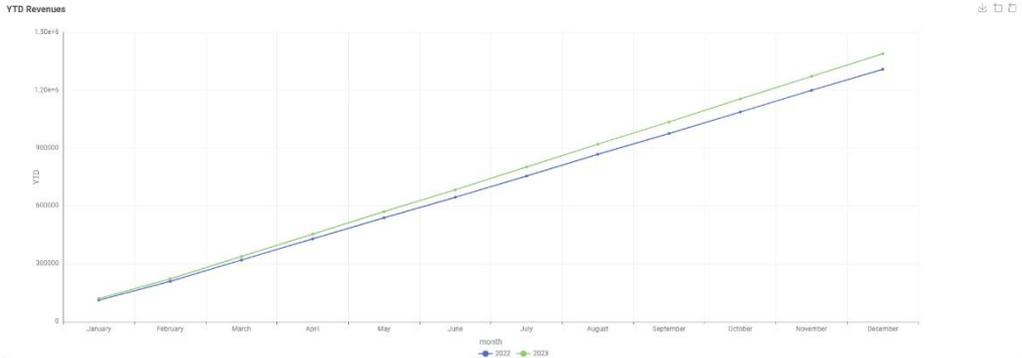


A Bar Chart visualization of monthly and YTD revenues.

You can see that the pattern is similar for both years, February being a lower revenue month, and that 2023 has produced a bit more revenues than 2022.

Financial Planning & Analysis
Monthly and Year-to-Date Revenue Aggregations

This last observation is confirmed in the view of the YTD revenues in the line plot. In the YTD plot (below) the line for 2023 is clearly above the line for 2022.



A Line Plot visualization of the monthly and YTD revenues.

KNIME for Finance

Data experts spend 70-80% of their time wrangling and preparing data before they can start performing aggregations and analyzing it. This becomes even more complex when the data is unstructured, large in size, and has to be pulled from various sources and types.

KNIME, as a low-code data analytics platform, offers a solution. With versatile functionality offered by its nodes like the *Joiner*, *Concatenate*, *GroupBy*, *Value Lookup*, and more, finance departments can merge diverse datasets with precision and quickly perform complex data aggregation and joining tasks.

You can download KNIME and try it out for yourself. It's open source and free to use.

Financial Services

In this chapter we have a look at the Financial Services industry, which mainly focuses on money management, including the provision of savings, the checking of accounts and loans, the offering of insurance policies, and more. With the help of KNIME, Financial Services professionals are able to automate tedious data processing tasks, make data analysis accessible to both business and data experts, or build sophisticated models to calculate accurate credit scores, predict cash flow, anticipate customer needs, and more. In this chapter, we introduce the use case of approving loans.

This chapter includes the article:

- Credit Scoring for Loan Approval, p. 64

Credit Scoring for Loan Approval

Author: Michele Bassanelli

Workflows on KNIME Community Hub: [01_Training Loan Request](#) & [02_Deployment Loan Request](#)

Increasing numbers of banks are adopting machine learning to compute credit scores to deliver better performance than traditional methods. In such cases, the bank clerk trains a prediction model on data about loan applicants, evaluates the model's performance, and then uses the model on new incoming loan applications to assign applicants a credit score. This would function perfectly if we assume that any mistakes made by the prediction model are the same. But they're not. Granting a loan to a customer who will never repay it is worse than denying a loan to a creditworthy customer.

In this subchapter, learn how you can train a machine learning model to predict credit scores and additionally how you can determine when you need to optimize the model by assessing the cost of a mistake. We use the low-code KNIME Analytics Platform to train and optimize the model. The intuitive, user-friendly interface of the tool enables anyone to start using modern data science methods without having to learn to code first.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Credit Scoring](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Train and Optimize a Machine Learning Model for Credit Scoring

When making predictions with a machine learning model, it's unlikely you'll achieve 100% accuracy. The model will never be perfect. It will surely make mistakes. Mistakes can be costly, especially in particular domains, like medicine. Then, what is the cost of a mistake? Is it worth it to optimize my model even more and gain that 1% of accuracy?

In the article "[How to optimize the classification threshold](#)", you can learn how to optimize the classification threshold of a classifier predicting whether or not to grant loans to the customers of a bank. Let's revisit that use case.

It's classic credit scoring. Of the many applicants asking the bank for loans, which ones should we grant the loans to? Applicants can easily be separated in two groups: the ones who repay the loans and the ones who do not. The goal is to assign the loan to the first group of customers and deny it to the second group.

The loan granting process usually requires some manual steps to evaluate the applicant's assets, age, credit history, and so on. During this manual process, mistakes will be made for sure. Every now and then, loans will be granted to applicants who will not repay them, and loans will be refused to applicants who would have otherwise repaid them. A good clerk produces only a minimum number of such mistakes. But how much does a mistake cost to the bank? Are all mistakes the same?

A commonly used dataset to infer credit worthiness of applicants is the [German Credit Data Set](#), available on the site of the [University of California Archive for Machine Learning and Intelligent Systems](#).

The dataset contains 1000 loan applicants. Each applicant is described via 20 features, grouped in:

- socio-demographic features, like age and married/unmarried status.
- financial features, like income and other wealth indicators.
- personal features, like job, housing, and properties.
- Loan request data, such as loan amount and purpose.

The target variable (column 20) describes the applicant's creditworthiness as evaluated by the bank clerk (2 = risky and 1 = creditworthy). 700 applicants (70%) are labeled as creditworthy in the dataset and 300 applicants (30%) as risky.

The Workflows: Train a Machine Learning Model to approve or disapprove Loan Requests

Let's replace our human clerk with a machine learning algorithm, like a random forest, and let's train it to distinguish the applicants that are worth a loan from those who are not worth it. The workflow below reads the data from the "German Credit Dataset" file, splits it in two datasets – one for training and one for testing – prepares the data, trains the machine learning model to predict whether the applicant is trustworthy or risky, (i.e. whether to grant or reject a loan request), and then evaluates how well the model performs on the test set.

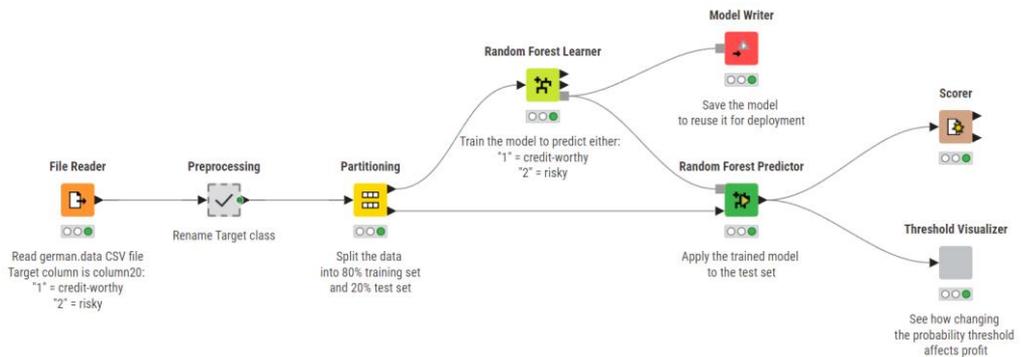
You can download the workflow [01_Training_Loan_Request](#) for free from KNIME Community Hub.

Financial Services
Credit Scoring for Loan Approval

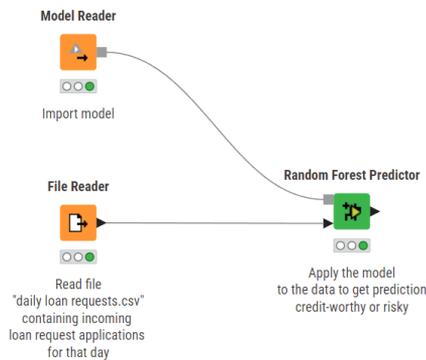
A separate workflow deploys the trained model on new data. That is, it reads the trained model and applies it to evaluate the new incoming loan applications day after day, in everyday bank operations.

The deployment workflow [02_Deployment Loan Request](#) is also publicly available to download from KNIME Community Hub.

Notice that the trained model in the end produces a probability of the applicant being risky. Decisions about granting loans depend on how conservative we are at evaluating this probability. The lower the threshold applied to the probability, the more conservative the decision process.



This workflow trains a Random Forest to predict whether or not a loan applicant is creditworthy.



This workflow deploys a previously trained model and applies it to new data to predict whether or not the new applicants are creditworthy.

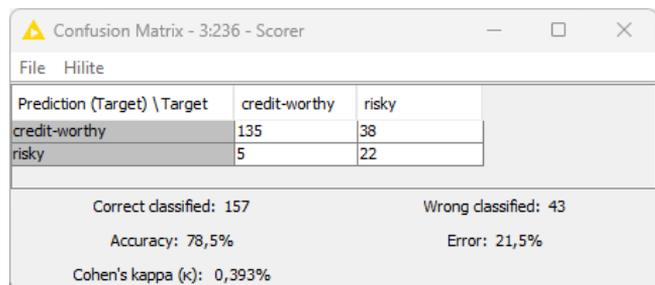
In production the model operates on real data, and we do not have the luxury of knowing whether the model makes a mistake, time will be the ultimate judge. We can only evaluate how correct the model is on the test set, where the real evolution of the applicant is fully known.

For the model performance evaluation, usually, a confusion matrix is calculated on the data of the test set. The confusion matrix is a way of counting the number of correct predictions and the number of mistakes. You can see in each cell:

- The number of applicants who were correctly labeled as trustworthy. In this case, the bank would correctly grant their loan request.
- The number of applicants who were incorrectly labeled as risky, but they were trustworthy. The bank would deny their loan request, but if granted they would have repaid it.
- The number of risky applicants erroneously labeled as trustworthy. This is probably the biggest mistake the bank could make, because it would grant a loan that would never be repaid, thus losing the whole loan amount.
- The number of applicants who were correctly judged as risky, and their loan request was rejected.

As it is now, for that random choice of data for the training and test set, our model is correct on 78.5% of the applicants and wrong on the remaining 21.5%.

The total percentage of mistakes on the test set applicants is 21.5%. However, not all mistakes are the same. Are we sure that refusing a loan to a potential good customer has the same value as granting a loan to somebody who will not pay it back? Every mistake has its cost. Also, every correct decision has its profit. Rejecting a risky loan request produces zero profit, while granting a trustworthy loan request will produce in the long term some profit for the bank. While we want to grant loans to the worthy applicants, we want even more to refuse loans to the unworthy applicants. Let's delve a bit deeper on that.



Prediction (Target) \ Target	credit-worthy	risky
credit-worthy	135	38
risky	5	22

Correct classified: 157 Wrong classified: 43
Accuracy: 78,5% Error: 21,5%
Cohen's kappa (κ): 0,393%

A confusion matrix. Column headers indicate the applicant class in the test set and row IDs the model prediction. Each cell contains the number of applicants from a specific original class and receiving the indicated prediction by the model.

How to quantify the Cost of a Decision

It is not easy to quantify the cost of decisions. The article "An analysis of profit and customer satisfaction in consumer finance case studies" in *Business, Industry And Government Statistics*, 2, pages 147-156, 2014, by C. Wang & M. Zhuravlev assigns the following cost/profit factor w to each one of the possible decisions in credit scoring.

In general, refusing a loan does not bring any loss. The loan is not granted, the money cannot be lost. In this case, the cost/profit $w = 0$.

- $w = 0$ for correctly refusing a loan to risky customers. This means that rejecting a risky customer bears no cost and no profit.
- $w = 0$ also for incorrectly refusing a loan to a creditworthy customer. This decision, though incorrect, bears no cost to the bank. Obviously, it is a mistake, but not a serious one. If we must make mistakes, these are the easiest ones to make. The loan will be refused, no money is lost.

Granting a loan is the decision that bears consequences. If we grant the loan to a creditworthy customer, the average profit depends on the loan interest. If we grant a loan to a risky customer, we will lose it all. In this case, the cost factor is $w = -1$. In the above referenced article (C. Wang & M. Zhuravlev) it was estimated:

Prediction Target	risky	creditworthy
risky	0	0
creditworthy	-1	0.35

Profit/Cost Matrix. Each cell contains the profit/cost for each euro invested in a loan.

- $w = 0.35$ for correctly granting a loan to creditworthy customers. On average a customer repaying a 1 € debt brings a 0.35 € profit to the bank.
- $w = -1$ for incorrectly granting a loan to risky customers. Assigning a loan to a risky customer has a cost of -1 € for each 1 € of the loan.

The Results: A Threshold Visualizer for accurate Credit Scoring

Total profit/cost on the test set: Let's consider 200 loan requests coming in every day, with applicants distributed 70% (140) – 30% (60) across the two classes, and an average loan amount of 10 000 €.

Let's also suppose that our machine learning model is perfectly correct, then every day the bank invests for a full profit/cost of:

$$profit/cost = 140 * 0.35 + 60 * 0 * 10\,000 \text{ €} = 490\,000 \text{ €}$$

Let's now suppose that our model accepts all loan applications. In this case the daily total investment is:

$$profit/cost = 140 * 0.35 + 60 * (-1) * 10\,000 \text{ €} = -110\,000 \text{ €}$$

Let's now suppose that our model denies all loan applications. In this case the daily investment would be:

$$profit/cost = 140 * 0 + 60 * 0 * 10\,000 \text{ €} = 0 \text{ €}$$

The machine learning model will not work perfectly. It will perform in between these numbers and will lose or make money in between these amounts. For the confusion matrix in the figure above, for example, the investment would be:

$$profit/cost = 135 * 0.35 + 38 * -1 + 22 * 0 + 5 * 0 * 10\,000 \text{ €} = 92\,500 \text{ €}$$

Is it possible to steer the model decisions in the direction that makes more money?

Choosing the threshold appropriately: After training, the machine learning model produces the probability for the input applicant to belong to one of the two classes: creditworthy or risky. By default, if the probability of the customer being risky p is above 0.5, the loan request is rejected. On the opposite, if the probability of the customer being trustworthy $1 - p$ exceeds 0.5 then the loan request is granted. The default threshold is 0.5, which means we consider both decisions equally costly. What about changing the threshold value to favor one or the other class?

For example, instead of a threshold of 0.5 we could use a threshold of 0 for probability p . This means that we reject all loan applications, ending up with the $0 \text{ profit} - 0 \text{ loss}$ situation. If we use a threshold of 1, we accept all loan requests, leading to a loss of circa $-110k \text{ €}$. We can vary the threshold till we find the most rewarding value given the cost matrix that we presented above.

This task was implemented in the “Threshold Visualizer” component. The composite view of this component shows the new confusion matrix, including the cost matrix, and how it varies when selecting a different decision threshold.

Financial Services Credit Scoring for Loan Approval

Confusion Matrix

Prediction Label	Loan-worthy	Risky
credit-worthy	135	38
risky	5	22

Showing 1 to 2 of 2 entries

Cost Matrix

Prediction Target	Loan-worthy	Risky
credit-worthy	1.35	-1
risky	0	0

Showing 1 to 2 of 2 entries

Profit Matrix

Gain	Loss
182.25	-38
0	0

Showing 1 to 2 of 2 entries

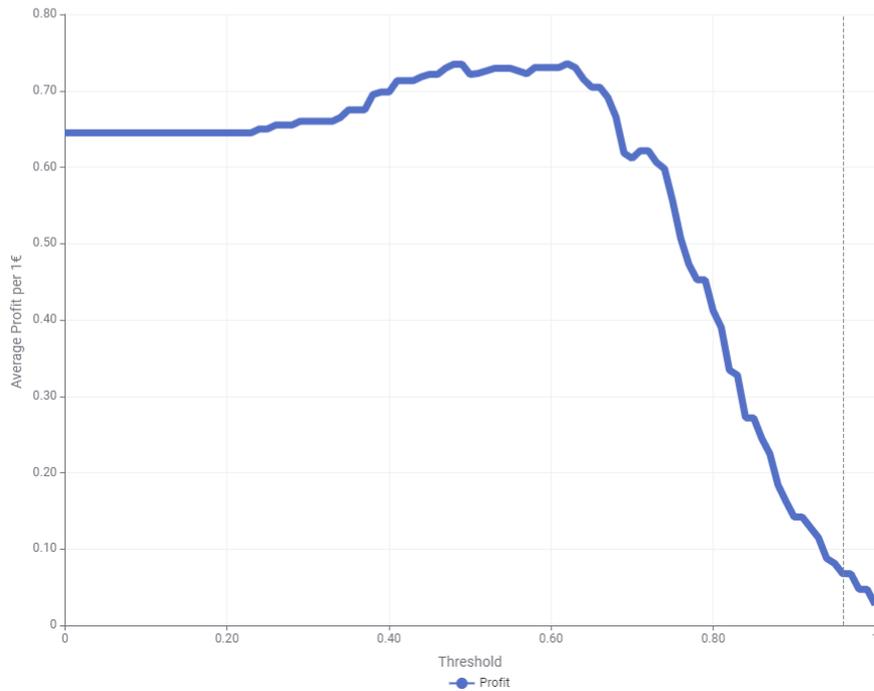
Threshold



Update the confusion matrix

refresh

Profit vs. Probability Threshold



View of the Threshold Visualizer component.

An Important Addition to the classic Model Training Cycle

We wanted to show the process of training a machine learning model to predict successful loan applications. A common approach is to read the data, split it in training and test set, prepare the data, train the model, and evaluate the model performance on the test set. Then a deployment workflow is created, consuming the trained model on new incoming loan applications.

This is all considering that all mistakes are the same. Granting or not granting a loan to a risky or to a creditworthy customer does not make a difference. This is not true. Granting a loan to a customer who will never repay it is worse than denying a loan to a creditworthy customer. We assigned a cost / profit to each decision, which translated into a total cost / profit when applied to all decisions on the whole data set. Suddenly a 1% more in accuracy means an increase in profit or cost depending on the class affected by the accuracy improvement.

Varying the threshold of the model decision and optimizing it in terms of total cost on the test set becomes an important step to add to the classic model training cycle.

Download KNIME and try out the [workflows in this section](#) for your own credit scoring use cases.

Tax

With KNIME, tax professionals can ensure compliance with effective tax data management, analytics, and visualization. KNIME allows tax professionals to be confident of data integrity and gain better understanding of anomalies. In addition, KNIME reduces time spent gathering data from multiple sources (e.g., ERP and billing systems), cleaning data, and reporting. In this chapter, we introduce some workflow examples for tax apportionment, calculations, validations, and reporting.

This chapter includes the articles:

- Sales Tax Reporting, p. 73
- Transfer Pricing Recharge, p. 91

Sales Tax Reporting

Authors: Daria Liakh & Tejus Naik

Workflows on KNIME Community Hub: [Nexus Threshold Analysis](#) & [Sales Apportionment](#)

Sales tax is a statutory requirement and analyzing sales tax liability accurately is important for all organizations. Ever-changing local tax rules add to the complexity and put organizations at risk of a tax audit, which can lead to back taxes and penalties.

Many sales tax reporting processes are repetitive and quickly become labor-intensive if teams have to perform them all manually.

Automating sales tax calculations, analysis and reporting can help finance teams quickly reconcile and verify the sales tax they pay in a given year. There are proprietary tax software tools in the market that can be used to automate these tax calculations, however, these “black box” solutions may not provide the clarity and audit trail required to explain tax calculations in case of an audit.

In this article we’d like to show how you can use KNIME to automate sales tax calculations and provide you with an in-built audit trail. The intuitive drag-and-drop nature of KNIME Analytics Platform makes it simple to adjust the solution to incorporate new local tax rules as necessary.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Sales Tax Reporting](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Automate Sales Tax Reporting

We’re going to demonstrate **two different aspects** of sales tax reporting based on the example of a company that sells subscriptions. The two types of sales tax reporting are:

- **Apportionment of sales and sales tax calculations:** Total sales are calculated by multiplying the usage duration within the taxation period by the subscription fee. For accurate sales tax calculations though, we need to make sure that all the sales where the billing zip code is unknown get apportioned to the appropriate states.

Tax
Sales Tax Reporting

- **Economic nexus threshold analysis:** [Economic nexus](#) requires sellers to collect sales tax in states where the seller’s sales exceed the state’s monetary threshold. We will demonstrate how we can create an application that informs an organization about the states where it needs to collect and remit sales tax.

We’ll also walk through a [calculator](#) we built for this task. It calculates the duration of an event/subscription in frames of a floating 12-month period. This calculator is available as a so-called [shared component](#) for you to plug into your own solutions.

The sales tax reporting process for each type is essentially the same. Below, we’ve highlighted the differences in bold.

Sales Apportionment/Sales Tax Calculations	Economic Nexus Threshold Analysis
Extract and clean data from multiple sources (e.g., ERP systems, Excel sheets, etc.): <ul style="list-style-type: none"> • Transaction data, in our case generated subscription data; • Data with ZIP code and Tax Rate information for each state name. 	Extract and clean data from multiple sources (e.g., ERP systems, Excel sheets, etc.): <ul style="list-style-type: none"> • Transaction data, in our case generated subscription data; • Data with ZIP code information for each state name. • Data Table with nexus threshold information per state.
Calculate Total Sales per customer based on the membership’s duration falling into a floating 12-months period.	Calculate Total Sales per customer based on the membership’s duration falling into a floating 12-month period.
Apportion sales to appropriate states where billing ZIP code is unknown. Calculate sales tax by state.	Identify the states where the total state sales have breached the economic nexus threshold assigned statuses; above, below, approaching.
Visualize the results and allow download of the final tables.	Visualize the results and allow download of the final table with statuses assigned.



A collection of common KNIME connector nodes to connect to various data sources like cloud storages or databases.

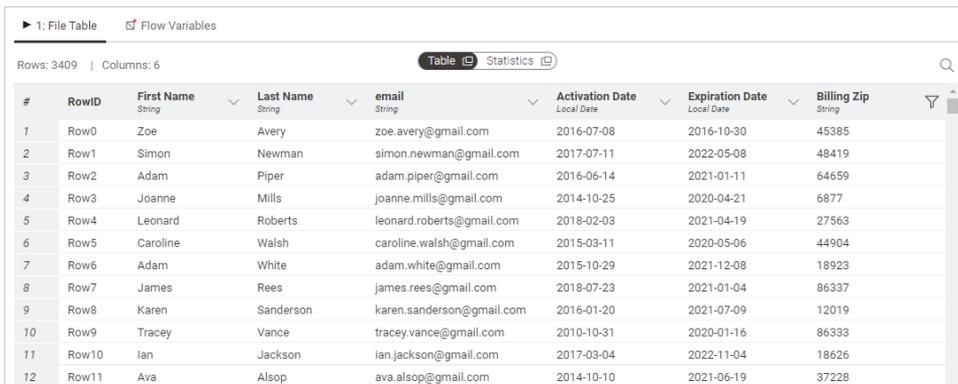
Tax

Sales Tax Reporting

The billing data can be pulled using any of the KNIME connectors. Collect data from ERP systems such as SAP using the [SAP Reader](#) node, out of Odoos using the [GET Request](#) node, out of a database using the DB nodes, or simply out of an Excel file using the [Excel Reader](#) node. For the purposes of this article, we've built a workflow that takes data from an Excel file. This is so that you can download the workflow and try it out immediately.

In order to make this example work on everybody's laptop, without the need of accounts and credentials, we used sample Excel files as our source data for:

- Subscriber billing information with, in this case, Expiration and Activate Dates of the contract (generated):



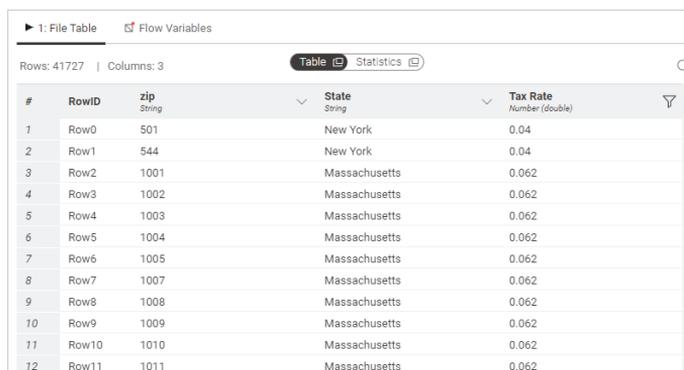
► 1: File Table | Flow Variables

Rows: 3409 | Columns: 6

#	RowID	First Name	Last Name	email	Activation Date	Expiration Date	Billing Zip
1	Row0	Zoe	Avery	zoe.avery@gmail.com	2016-07-08	2016-10-30	45385
2	Row1	Simon	Newman	simon.newman@gmail.com	2017-07-11	2022-05-08	48419
3	Row2	Adam	Piper	adam.piper@gmail.com	2016-06-14	2021-01-11	64659
4	Row3	Joanne	Mills	joanne.mills@gmail.com	2014-10-25	2020-04-21	6877
5	Row4	Leonard	Roberts	leonard.roberts@gmail.com	2018-02-03	2021-04-19	27563
6	Row5	Caroline	Walsh	caroline.walsh@gmail.com	2015-03-11	2020-05-06	44904
7	Row6	Adam	White	adam.white@gmail.com	2015-10-29	2021-12-08	18923
8	Row7	James	Rees	james.rees@gmail.com	2018-07-23	2021-01-04	86337
9	Row8	Karen	Sanderson	karen.sanderson@gmail.com	2016-01-20	2021-07-09	12019
10	Row9	Tracey	Vance	tracey.vance@gmail.com	2010-10-31	2020-01-16	86333
11	Row10	Ian	Jackson	ian.jackson@gmail.com	2017-03-04	2022-11-04	18626
12	Row11	Ava	Alsop	ava.alsop@gmail.com	2014-10-10	2021-06-19	37228

Subscription data.

- Reference data to map billing zip code to state names and tax rates. The sources used for assembling the table can be found here on [opendatasoft](#) and [tax foundation](#).



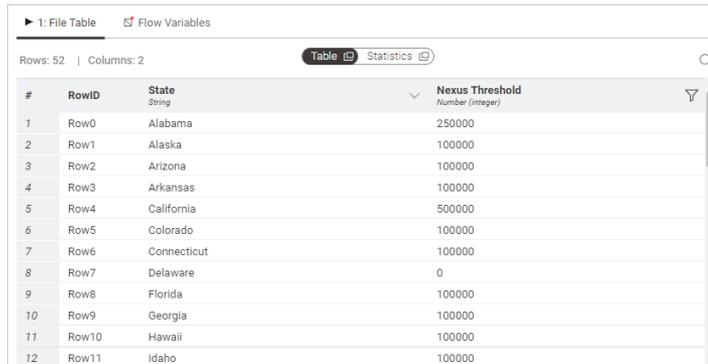
► 1: File Table | Flow Variables

Rows: 41727 | Columns: 3

#	RowID	zip	State	Tax Rate
1	Row0	501	New York	0.04
2	Row1	544	New York	0.04
3	Row2	1001	Massachusetts	0.062
4	Row3	1002	Massachusetts	0.062
5	Row4	1003	Massachusetts	0.062
6	Row5	1004	Massachusetts	0.062
7	Row6	1005	Massachusetts	0.062
8	Row7	1007	Massachusetts	0.062
9	Row8	1008	Massachusetts	0.062
10	Row9	1009	Massachusetts	0.062
11	Row10	1010	Massachusetts	0.062
12	Row11	1011	Massachusetts	0.062

State-specific ZIP and tax rate data.

Tax Sales Tax Reporting



#	RowID	State String	Nexus Threshold Number (integer)
1	Row0	Alabama	250000
2	Row1	Alaska	100000
3	Row2	Arizona	100000
4	Row3	Arkansas	100000
5	Row4	California	500000
6	Row5	Colorado	100000
7	Row6	Connecticut	100000
8	Row7	Delaware	0
9	Row8	Florida	100000
10	Row9	Georgia	100000
11	Row10	Hawaii	100000
12	Row11	Idaho	100000

Economic nexus threshold data.

The Workflows: Sales Tax Reporting

The workflows for [Sales Tax Reporting](#) are available and free to download from the KNIME Community Hub. Let's walk through the workflows for both use cases and then we will show you how to turn a workflow into a data app.

Sales Apportionment – Static Workflow

You can download the Sales Apportionment – Static workflow [here](#). It consists of three main parts, with each part handled by a workflow collected into components, which you can see below.



Overview of the Sales Apportionment – Static workflow.

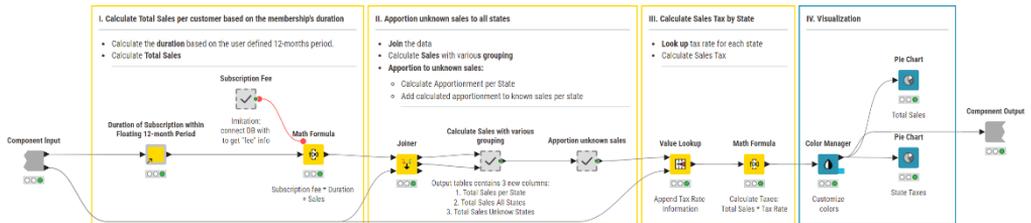
The components take care of the following tasks:

1. Read and clean the data, described above.
2. Calculate sales taxes per state with apportionment of sales where billing zip code is unknown for the taxation period specified by a user and have a first visual check with pie charts.
3. Sort states by its tax liability and prepare the resulting tables to view and download the results.

Let's look in more detail in how the sales tax reporting is actually performed in case of unknown sales in the data, i.e., how we apportion them to appropriate states. By holding *CTRL* + double clicking the component "Sales Tax Reporting" we see the following:

Tax

Sales Tax Reporting



Overview of the inside of the component "Sales Tax Reporting".

First, we calculate total sales per customer based on the membership's duration. For that, use the [component](#) to automatically calculate the duration of any active event/subscription (with Activation and Expiration dates' information provided) in frames of a floating 12-month period. This is useful for yearly reports, when one needs, for instance, to calculate the duration of the active subscription falling into the reporting period.

Note. To recap what components are and how they encapsulate functionality check out the video "[What is a Component in KNIME Analytics Platform?](#)" on KNIMETV.

All you have to do is specify the start date of the 12-month period and the component automatically determines the end of the period (+12 month) and calculates the duration of each contract/event/subscription which was active in this selected period, considering also the point in time of activation and expiration dates.

Next, the *Math Formula* node calculates the sales for each customer by multiplying the duration of his/her active subscription by the subscription fee. The latter is a fixed number here just for simplified reasons, but you can prompt the user to specify it by using *Widget* or *Configuration* nodes, or connect to any source with this information – depending on your specific situation/set-up.

Now we want to apportion sales with unknown billing zip to all states. We use the *Joiner* node to blend the subscriber information with the state name and tax rate using the billing zip code as the matching column.

In the "Calculate Sales with various grouping" metanode, we use the *GroupBy* node to calculate three totals:

- Total Sales per state;
- Total Sales for all states;
- and Total Unknown Sales.

Note. Metanodes are different from components. Components really are KNIME nodes that you create which bundle functionality, have their own configuration dialog and their own composite views. Metanodes on the other hand are containers of a part of your workflow, that help to build cleaner and structured nested workflows.

Tax
Sales Tax Reporting

Once the above totals are calculated, we move to the metanode “Apportion unknown sales”, which essentially consists of 2 *Math Formula* nodes:

- The first calculates the total unknown sales to be apportioned to a state using the formula:

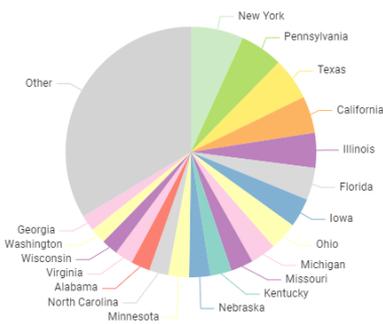
$$\begin{aligned} & \text{Total Unknown Sales to be apportioned to a state} \\ & = \left(\frac{\text{Total Sales per State}}{\text{Total Sales}} \right) * \text{Total Sales Unknown} \end{aligned}$$

- The second adds the apportioned unknown sales to the known sales for the state to calculate the total sales per state.

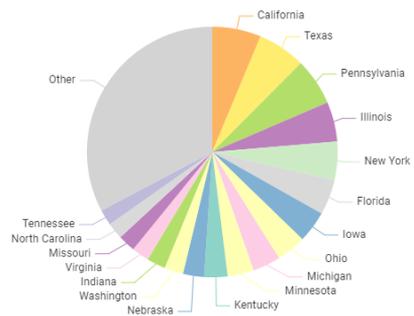
Now let’s calculate sales tax for each state. For that we look the state-specific tax rate up with the *Value Lookup* node and multiply total sales by it for each state with the *Math Formula* node.

We want to visualize the Total Sales and Sales Tax by State with the *Pie Chart* nodes.

Total Sales by State



Sales Tax By State



Preview of the resulting Pie Charts – stay tuned for the Data App enhancement!

Finally, we want to prepare the resulting tables for download. We do that with the “View and Download Results” component, where we display the table in the composite view with the *Table View* node and save two output data tables with *Excel Writer* nodes: one of each contains only 10 states sorted by the highest tax liability with the *Top k Row Filter* node.

Tax Sales Tax Reporting

10 States with Highest Sales Tax Liability

Rows: 10 | Columns: 3

State <i>String</i>	Total Sales <i>Number (double)</i>	Sales Tax <i>Number (double)</i>
California	228,146.475	16,540.619
Texas	260,426.192	16,276.637
Pennsylvania	265,897.331	15,953.84
Illinois	216,109.97	13,506.873
New York	326,079.854	13,043.194
Florida	199,696.555	11,981.793
Iowa	179,453.342	10,767.201
Ohio	174,529.318	10,035.436
Michigan	157,568.789	9,454.127
Minnesota	129,665.982	8,914.536

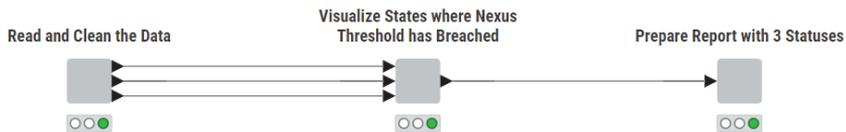
Resulting table with 10 states sorted by how high their tax liability is.

Here we have it, taxes were calculated with apportionment of the sales with unknown billing zip.

Now, let us move to the second use case, where we want to show you how one could analyze the economic nexus threshold with KNIME.

Nexus Threshold – Static Workflow

You can download the Nexus Threshold – Static workflow [here](#). In this case the membership data has no sales with missing billing ZIP. Hence, no apportionment is required.

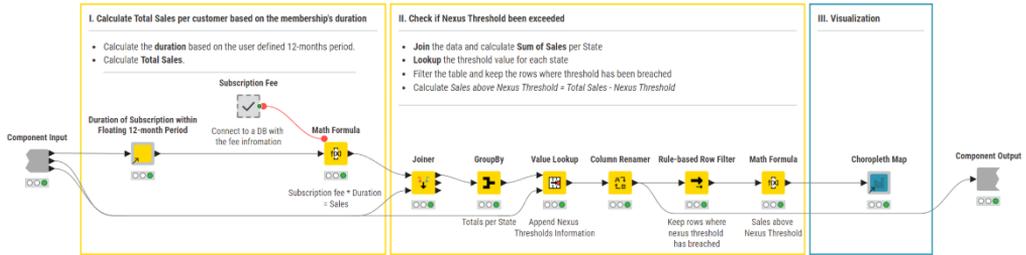


Overview of the Nexus Threshold – Static workflow.

You see one extra output port for the “Read and Clean Data” component - this is the 3d table we mentioned with nexus threshold information for each state.

Let’s look in more detail about how we analyze the nexus threshold situation for each state within “Visualize States where Nexus Threshold has Breached” component.

Tax Sales Tax Reporting



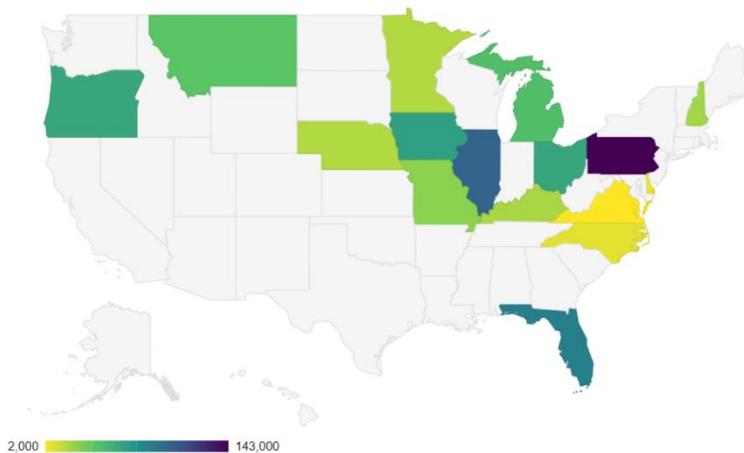
Overview of workflow to visualize states where the Nexus Threshold was breached.

The first step is identical to the previous workflow. In the second step, we want to look into the threshold analysis. We calculate total sales per state, add to the table the state-specific economic nexus threshold values and filter the data by keeping only the rows, in this case States, for which the nexus threshold has been breached.

Additionally, we calculate the difference between the total sales and threshold value and display those with the *Choropleth Map* component. It creates a choropleth map that colors states for which threshold breached. The color scheme is created by using the numerical values from the column we calculated one step before - the difference between the total sales and threshold value, per state.

States where Nexus Threshold has Breached

The color scale below represents degree of separation between the nexus threshold and total sales in that state.



An interactive visualization of the states that breached the Nexus threshold.

Now let's have a look at the component "Prepare Report with 2 Statuses". Here the Rule Engine node assigns one of the three nexus threshold (NT) related statuses to each state:

Tax
Sales Tax Reporting

- Above NT means that the nexus threshold for that state has been breached.
- Approaching NT means within 10000 \$ from reaching the nexus threshold for the selected state.
- Below NT means total sales for that state are more than 10000 \$ below the nexus threshold.

The *Color Manager* node assigns red, yellow, and green colors, respectively, which are added to the table.

Nexus Threshold(NT) Status Report

Rows: 51 | Columns: 3

State <i>String</i>	Total Sales <i>Number (double)</i>	Nexus Threshold (NT) Status <i>String</i>
Alabama	107,500	Below NT
Alaska	12,500	Below NT
Arizona	67,000	Below NT
Arkansas	55,000	Below NT
California	208,500	Below NT
Colorado	73,000	Below NT
Connecticut	37,500	Below NT
Delaware	6,000	Above NT
District of Columbia	5,500	Below NT
Florida	182,500	Above NT
Georgia	89,500	Below NT
Hawaii	7,500	Below NT
Idaho	44,000	Below NT
Illinois	197,500	Above NT

Resulting table with statuses for Nexus threshold per state.

Share Sales Tax Workflows as Interactive, Browser-based Data Apps

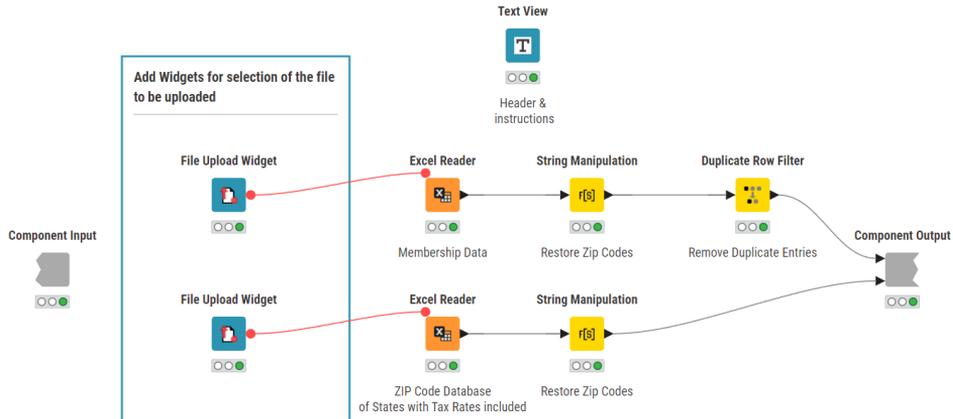
A [data app](#) is a web application that lets the user access the sales tax reporting workflows through a simple user interface.

Each of the 3 components in each workflow represents a web page in our data app. What is missing is interactivity in each of the components.

We want to enable the user to select the file they want to upload, select the taxation period, and download the report. We'll use *Widget* nodes:

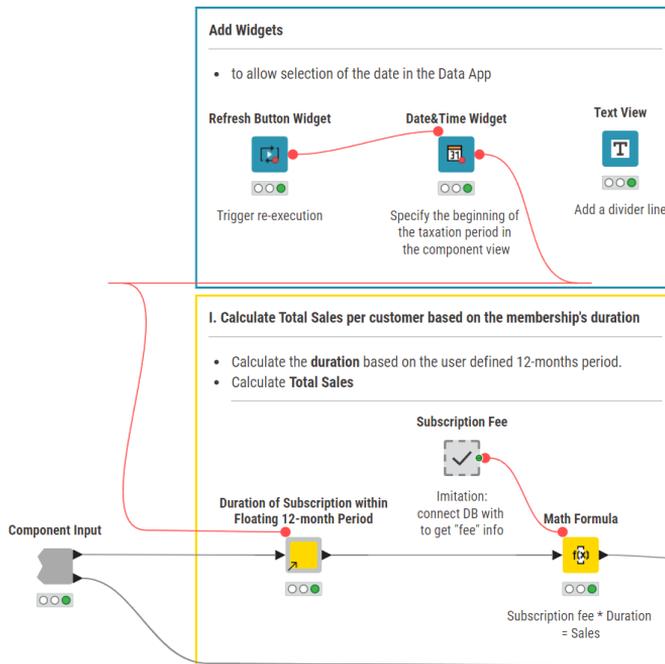
- To enable the user to select the file they want to upload we have to make adjustments to both workflows in the "Read and Clean the Data" component. We add the *Text View* node to display the instructions in the interactive view. The *File Upload Widget* nodes enable the user of the data app to upload their files.

Tax Sales Tax Reporting



Example of using Widget nodes: "Read and Clean the Data" component with added Widgets in the "Sales Apportionment – Data App" workflow.

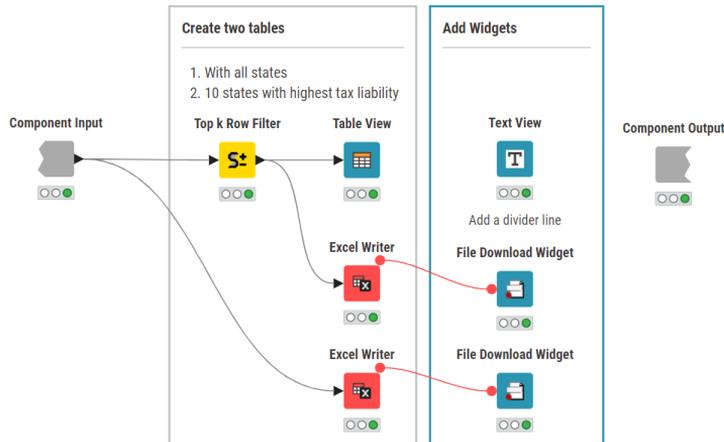
- To allow the user to select the start of the taxation period, we have to adjust both workflows in the second component. We add the *Date&Time Widget*, which creates a flow variable with the selected date. This is then used to overwrite the configuration settings of the shared component used to calculate the duration of an active subscription in this newly defined 12-months period.



Example of using Widget nodes: "Sales Tax Reporting" component with added Widgets in the "Sales Apportionment – Data App" workflow.

- To enable the user to download the final report, we add the File Download Widget nodes.

Tax Sales Tax Reporting



Example of using Widget nodes: “View and Download the Results” component with added Widgets in the “Sales Apportionment – Data App” workflow.

We give our data app a final touch by adding a flowchart header. This is handled by the [Data App Flowchart](#) component.

The data app workflows are available and free to download from the KNIME Community Hub:

- [Sales Apportionment - Data App](#)
- [Nexus Threshold Analysis - Data App](#)

These workflows have the same structure as their static versions. However, here, we control the beginning of the sales tax period from the interactive dashboard which changes the 12-months period for consideration for the tax reporting, add instructions and the flowchart, and allow more interactivity with file upload and download.

The Results: Deploy your Data App to KNIME Business Hub

Now, let’s deploy these two data apps on KNIME Business Hub and see how they will look for an end user.

Note. Learn how to deploy a data app to KNIME Business Hub in the video “[Deploy a data app on KNIME Business Hub](#)” on KNIMETV.

You can see screenshots below: the first three screenshots show the three pages of the **Sales Apportionment** data app, and the other three screenshots show the three pages of the **Nexus Threshold Analysis** data app. The flow chart on the top which we created tracks your progress of using the data app. The three pages you get in the data app correspond to the three components we built in KNIME Analytics Platform.

Sales Apportionment – Data App:

Page 1

The first page of the data app is the result of the “Read and Clean the Data” component. The two *File Upload Widget* nodes allow the end user to upload their own subscription data and a table containing the state name, ZIP code, and tax rate.

Open for Innovation
KNIME

Upload Data Select the Taxation Period and Execute Form the Report

State Sales Tax Data App

- Data App automates the calculation steps for sales tax report for the subscription data with dynamic selection of the 12-months period.
- Based on the user-specified start date of the taxation period, the Data App will automatically take the active membership falling into this period and compute the total sales per state based on the subscription fee.

Sales Apportionment Logic:

- Sales where billing zip code is unknown are being apportioned to all participating states.
- For the sake of simplicity, we have assumed that all states determine apportionment solely based on sales (i.e. not considering payroll and property apportionment factors).

Select subscription data (with activation&expiration dates):

Select file Membership Data.xlsx

Select table containing: State Name, Zip and Tax Rate

Select file States_zip_taxrate.xlsx

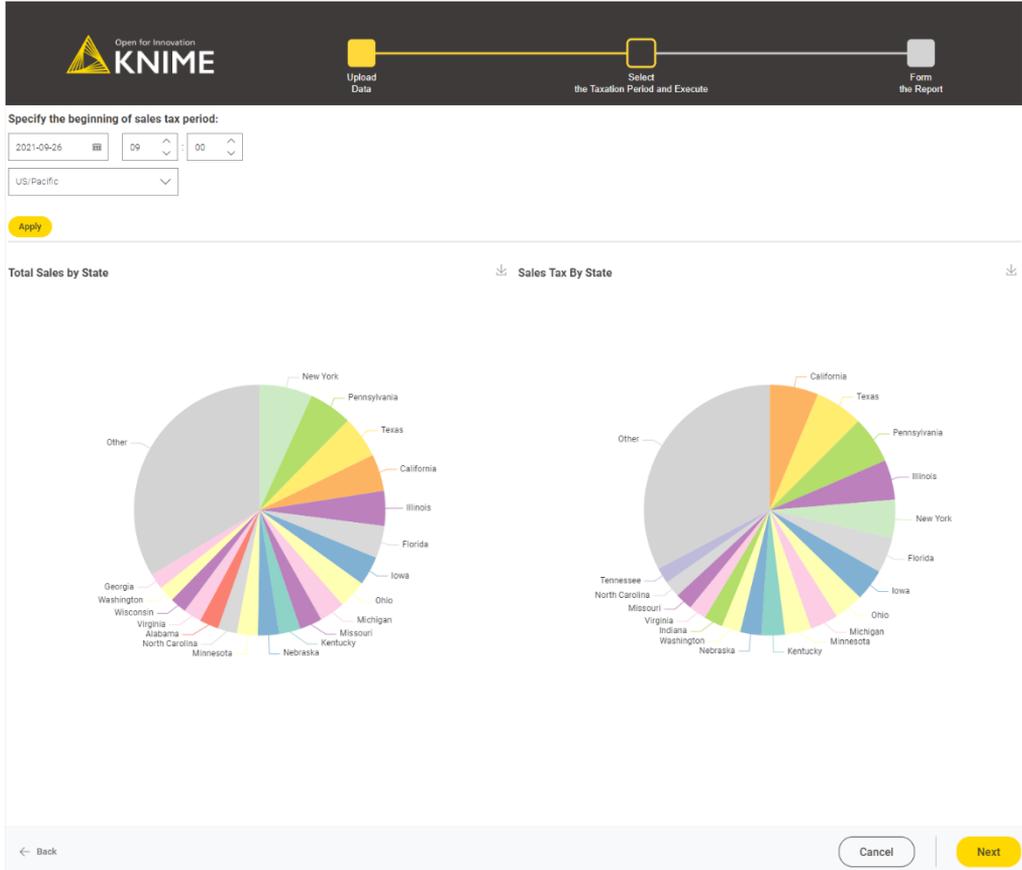
Cancel Next

Page 1 of the Sales Apportionment data app. It allows the end user to upload their own subscription data and a table containing the state name, ZIP code, and tax rate.

Tax
Sales Tax Reporting

Page 2

Page 2 displays the dashboard which is the result of the “Sales Tax Reporting” component. The dashboard allows the end user to specify the beginning of the sales tax period and displays the Total Sales and Sales Tax by State in two pie charts.



Page 2 of the Sales Apportionment data app. It allows the end user to specify the beginning of the sales tax period and displays the Total Sales and Sales Tax by State.

Tax

Sales Tax Reporting

Page 3:

The final page of the data app shows the top 10 states with the highest sales tax liability and allows the end user to download the result as a table (Table 1). It also creates a table with the tax information for all states which can also be downloaded (Table 2). It's the result of the "View and Download the Results" component.

The screenshot shows the KNIME interface for Sales Tax Reporting. At the top, there is a navigation bar with the KNIME logo and three steps: 'Upload Data', 'Select the Taxation Period and Execute', and 'Form the Report'. Below the navigation bar, there are two table options, each with a 'Download (xlsx)' button. The first table, 'Table 1: 10 states with highest sales tax liability', displays the following data:

State	Total Sales	Sales Tax
California	228,146,475	16,540,619
Texas	260,426,192	16,276,687
Pennsylvania	265,897,331	15,953,84
Illinois	216,109,97	13,506,873
New York	326,079,854	13,043,194
Florida	199,696,555	11,981,793
Iowa	179,453,342	10,767,201
Ohio	174,529,318	10,035,436
Michigan	157,568,789	9,454,127
Minnesota	129,665,982	8,914,536

The second table, 'Table 2: All states with tax information', is not fully visible in the screenshot. At the bottom of the interface, there is a 'Back' button.

Page 3 of the Sales Apportionment data app displays two tables. Table 1 showing the top 10 states with the highest sales tax liability, table 2 showing the tax information for all states. Both tables can be downloaded.

Nexus Threshold Analysis – Data App:

Page 1

The first page of the data app is the result of the “Read and Clean the Data” component. It allows to end user to update their respective data files.

The screenshot shows the 'Nexus Threshold Status Analysis' data app interface. At the top, there is a dark header with the KNIME logo (Open for Innovation) and a progress bar with three steps: 'Upload Data', 'Select the Taxation Period and Execute', and 'Form the Report'. The main title is 'Nexus Threshold Status Analysis'. Below the title, there are two bullet points explaining the app's functionality. A section titled 'Nexus Threshold Logic:' lists three steps: summing sales by state, comparing total sales to the nexus threshold, and highlighting states on a map. The interface features three file selection sections: 'Select subscription data (with activation&expiration dates):' with a file named 'Membership Data.xlsx'; 'Select table containing State Name and Zip' with a file named 'Sates_zip_taxrate.xlsx'; and 'Select table containing Nexus Threshold per State' with a file named 'Nexus Threshold.xlsx'. At the bottom right, there are 'Cancel' and 'Next' buttons.

Open for Innovation
KNIME

Upload Data

Select the Taxation Period and Execute

Form the Report

Nexus Threshold Status Analysis

- Data App automates the calculation steps for sales tax report for the subscription data with *dynamic* selection of the 12-months period.
- Based on the user-specified start date of the taxation period, the Data App will automatically take the active membership falling into this period and compute the total sales per state based on the subscription fee.

Nexus Threshold Logic:

- Sum up sales by state
- Compare total sales by state with nexus threshold for that state
- Highlight the states where nexus threshold on a US map for visualization

Select subscription data (with activation&expiration dates):

Select file Membership Data.xlsx

Select table containing State Name and Zip

Select file Sates_zip_taxrate.xlsx

Select table containing Nexus Threshold per State

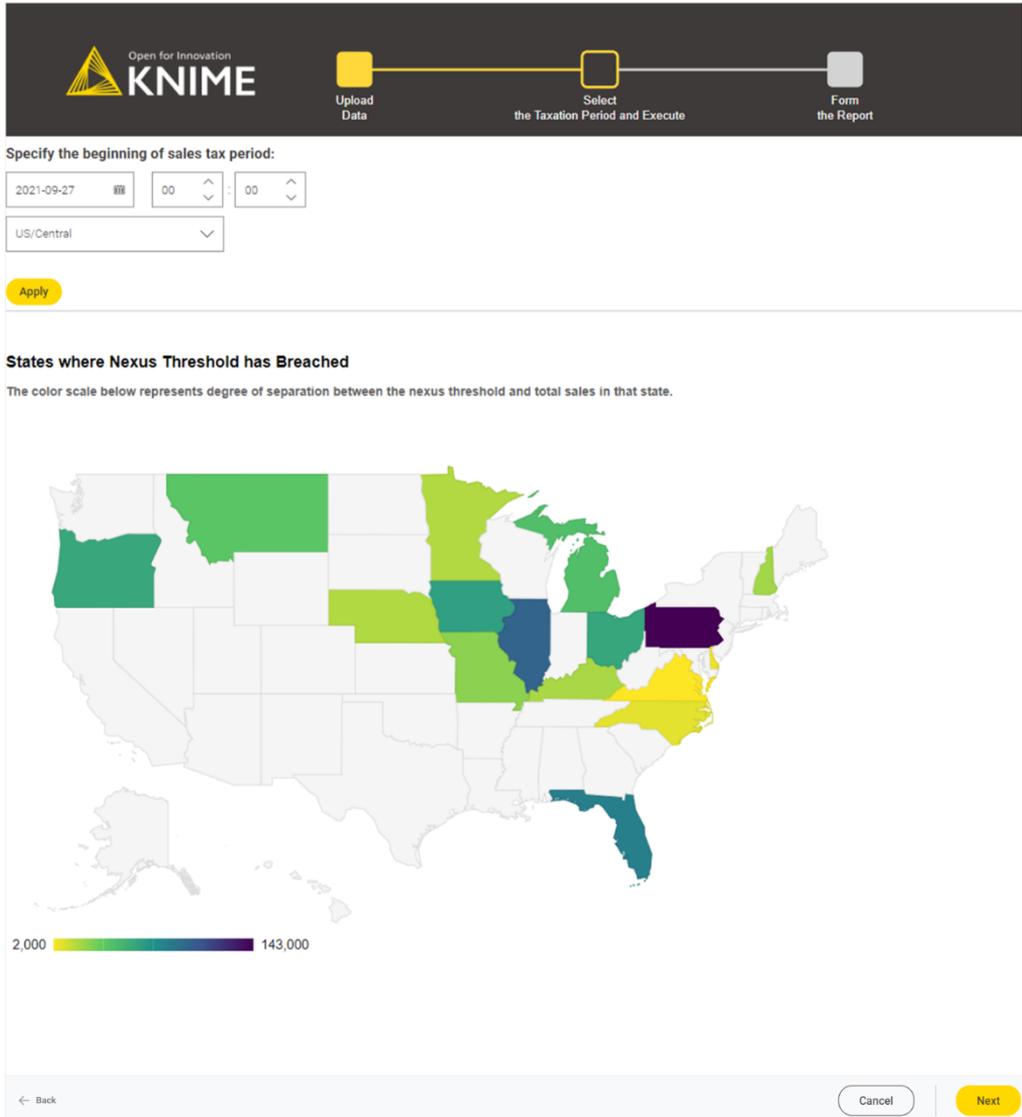
Select file Nexus Threshold.xlsx

Cancel | Next

Page 1 of the Nexus Threshold data app allows the end user to upload their respective data.

Page 2

The second page of the data app is the result of the “Visualize States where Nexus Threshold has a Breach” component. It lets the user interactively define the sales tax period through a *Date&Time Widget* node and then visualizes the states where the Nexus Threshold has breached on a map.



Page 2 of the Nexus Threshold data app lets the user define the sales tax period and then visualizes the states where the Nexus Threshold has breached on a map.

Tax
Sales Tax Reporting

Page 3

The final page of the data app displays the Nexus Threshold status report in a table view and also allows to download the report as an Excel file.

Nexus Threshold Status Report

[Download \(xlsx\)](#)

The statuses are defined:

- **Above NT** means that nexus threshold for that state has been breached.
- **Approaching NT** means within 10000 \$ from reaching the nexus threshold for the selected state.
- **Below NT** means total sales for that state are more than 10000 \$ below the nexus threshold.

Nexus Threshold(NT) Status Report

Rows: 51 | Columns: 3

State String	Total Sales Number (double)	Nexus Threshold (NT) Status String
Alabama	107,500	Below NT
Alaska	12,500	Below NT
Arizona	67,000	Below NT
Arkansas	55,000	Below NT
California	208,500	Below NT
Colorado	73,000	Below NT
Connecticut	37,500	Below NT
Delaware	6,000	Above NT
District of Columbia	5,500	Below NT
Florida	182,500	Above NT
Georgia	89,500	Below NT
Hawaii	7,500	Below NT
Idaho	44,000	Below NT
Illinois	197,500	Above NT
Indiana	83,000	Below NT
Iowa	164,000	Above NT
Kansas	56,500	Below NT
Kentucky	120,500	Above NT
Louisiana	40,500	Below NT
Maine	61,500	Below NT
Maryland	49,500	Below NT
Massachusetts	70,000	Below NT
Michigan	144,000	Above NT
Minnesota	118,500	Above NT
Mississippi	47,500	Below NT
Missouri	127,000	Above NT
Montana	39,500	Above NT
Nebraska	119,000	Above NT
Nevada	44,500	Below NT
New Hampshire	22,000	Above NT

← Back

Page 3 of the Nexus Threshold data app displays the Nexus Threshold status report in a table view and allows to download the report as an Excel file.

3 Key Benefits of the KNIME Solution

- A dynamic data app allows the user to perform operations from a user interface; more plots and charts could also be added to provide an even more robust and comprehensive view of an organization's sales tax liability and economic nexus status.
- You can easily connect to external sources via REST API (GET Request node) or to database (DB nodes) or to specific ERP systems (SAP Reader node) - just change the Excel Reader nodes to the dedicated connectors.
- If your input data changes, the taxation period can be shifted, and the workflow automatically takes care of the calculations correctly without you having to make manual adjustments.

KNIME for Finance

KNIME Analytics Platform is an open-source free low code platform, offering a large variety of data operations. Thanks to its visual and intuitive user interface, implementing solutions does not require any programming expertise. Finance experts can explore the Finance space on KNIME Community Hub for more low-code/no-code examples.

Follow us and send us your ideas for the next finance challenge at blog@knime.com.

Transfer Pricing Recharge

Authors: Daria Liakh & Max Menne

Workflows on KNIME Community Hub: [Transfer Pricing Recharge – Static](#) & [Transfer Pricing Recharge – Data App](#)

Transfer pricing recharge refers to the internal process within a multi-entity corporation where one subsidiary charges another subsidiary for goods, services, or intellectual property transferred between them.

It helps multinational corporations determine fair prices for transferred assets and services. It also helps these corporations maintain transparency and compliance with tax regulations by preventing price manipulation for tax avoidance purposes. Transfer pricing recharge is a crucial aspect of managing intercompany transactions, ensuring a balanced distribution of resources, and fostering accountability within the complex framework of a global business structure.

Companies take care of cost allocation for transfer pricing by initially identifying and categorizing costs associated with specific activities or projects. This process often involves meticulous spreadsheet analysis, where direct costs like materials and labor, along with indirect costs such as overhead, are allocated based on the resources consumed by each division or project.

The drawback of manual spreadsheet analysis lies in the potential for errors, time-consuming data entry, and the complexity of managing large datasets.

With the KNIME Analytics Platform such repetitive spreadsheet tasks can be automated, the accuracy of the analysis improved, and it allows for collaboration with ease. Insights can be obtained faster with the low-code/no-code interface.

Let's see how KNIME helps.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Transfer Pricing Recharge](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Transfer Pricing Recharge with the Cost-plus Method

We will demonstrate a KNIME solution for transfer pricing recharge that uses the cost-plus method. Support functions, e.g. in the headquarters (HQ) incur cost based on work that benefits other entities. Based on the arm's length principle of transfer pricing, services that are performed in favor of the other entities need to be recharged to the entities with a market price.

In this example, they are to be recharged to the entities based on cost drivers which are **Headcount** and **Revenue**.

On a high level, the transfer pricing and cost allocation process we showcase includes the following steps:

1. Extract accounting data from multiple sources (ERP and CRM systems, Files, Databases, etc.):
 - a. Data of Costs, which should be recharged;
 - b. Headcount and Revenue Figures
2. Calculate the charge with the specified markup* value
3. Calculate how much of the different costs should be assigned to each entity in %
4. Calculate recharges by entity
5. Save all bookings into a sheet or write to a database
6. Visualize recharges per country for each selected account
7. Create Accounting Bookings to be uploaded to ERP

**Note: In the domain of transfer pricing, a markup refers to the additional percentage or amount added to the cost of a product or service, needed to determine its transfer price.*

The data can be pulled using any of the KNIME connectors. You can collect data from ERP systems such as SAP using the [SAP Reader](#) node, out of Odoon using the [GET Request](#) node, out of a database using the DB nodes, or simply out of an Excel file using the [Excel Reader](#) node. For the purposes of this article, we've built a workflow that takes data from an Excel file. This is so that you can download the workflow and try it out immediately.

Tax Transfer Pricing Recharge



A collection of common KNIME connector nodes to connect to various data sources like cloud storages or databases.

In order to make this example work on everybody's laptop, without the need of accounts and credentials, we use sample Excel files or node as our source of the data:

Company	Journal	Account	Cost Center	Debit	Credit	Currency	Total
US HQ	OPEX	Personnel Costs	HR Group	25,000		USD	25,000.00
US HQ	OPEX	Hiring Costs International	HR Group	3,000		USD	3,000.00
US HQ	OPEX	Travel Costs	HR Group	12,000		USD	12,000.00
US HQ	OPEX	Software Costs International	IT Group	7,000		USD	7,000.00
US HQ	OPEX	Tax Advisory International	Tax Group	70,000		USD	70,000.00
US HQ	OPEX	Personnel Costs	HR Group		-100.00	USD	100.00

An example of journal entries.

A dummy file, created with *Table Creator* node, containing Headcount and Revenue Figures per entity.

Charge String	GER Number (double)	CH Number (double)	France Number (double)	Belgium Number (double)	Italy Number (double)	US non HQ Number (double)	US HQ Number (double)
HC	20	30	50	10	90	200	50
Revenue in kUSD	10	11	7	3	10	20	0

An example of Headcount and Revenue figures.

The Workflow: Transfer Pricing Recharge

The workflows for [Transfer Pricing Recharge](#) are available and free to download from the KNIME Community Hub.

Let's walk through the static workflow first and then we will show you how to turn it into a dynamic data app. You can download the Transfer Pricing Recharge – Static workflow [here](#).

The workflow for the described use case can be observed below. On a high level it consists of 3 steps, two of which are handled by workflows collected into components.

Tax
Transfer Pricing Recharge

Tip. Learn what a component is and how to create and reuse them in these videos:

- [What is a Component in KNIME Analytics Platform?](#)
- [Create, Modify, and Configure a Component in KNIME Analytics Platform](#)

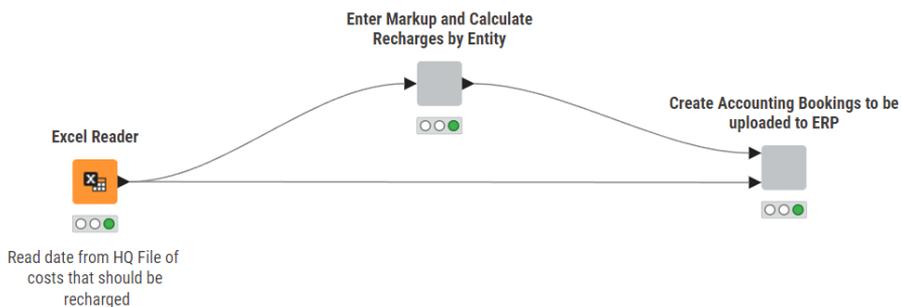
The idea of this workflow is to charge certain costs from the headquarters (HQ) entity to entities that are using its services. These include a markup based on transfer pricing.

For instance, the HQ entity supports all entities in the hiring process and then distributes its hiring costs to the local entities including a markup, based on a transfer pricing setup (e.g. comparable analysis with help of transfer pricing consultants).

There is a driver of the cost, that is Headcount or Revenue by which the costs are distributed to the entities. An output is generated that shows how much of the cost each entity needs to bear.

Further, the accounting entries for the HQ entity are created.

- To interact with Dashboards: **Right Click -> (Execute and) Open View**
- To see the calculation steps and more explanation -> **Right Click -> Component -> Open**



Transfer Pricing Recharge – Static workflow.

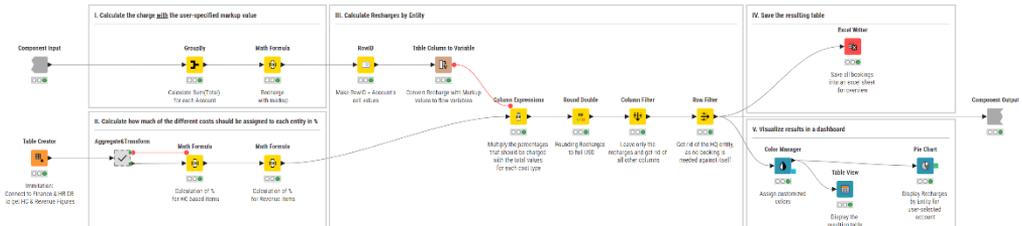
The steps of the workflow:

1. First, we read the cost data from HQ.
2. Second, inside the component “Enter Markup and Calculate Recharges by Entity” various nodes are used to calculate recharges by entity based on the static markup value. The bookings information is then being saved locally and displayed in a form of table and pie chart.
3. Third, within the component “Create Accounting Bookings to be uploaded to ERP” we create a final table to be uploaded to ERP and save it.

Let’s look inside of the components in more detail.

A Component to enter Markup and Calculate Recharges by Entity

We've documented the component structuring it into different blocks, for easier understanding. You can see the overview below. It consists of five blocks of action.



A KNIME component: "Enter Markup and Calculate Recharges by Entity".

1. Calculate the charge with the fixed markup value.

Here, we use the *GroupBy* node to calculate total costs for each account, using the Costs Data. Then, we recharge it with the static markup value, specified in the *Math Formula* node (15%):

$$\text{Recharge} = \text{Total Costs} * \left(1 + \frac{\text{Markup in \%}}{100}\right)$$

Note. Stay tuned to see how we can make markup value specification dynamic!

2. Calculate how much of the different costs should be assigned to each entity in %

Here, within the metanode "Aggregate & Transform ", we calculate the sum across all entities for HC and Revenue in kUSD, separately and convert those values to flow variables *Headcount (HC)* and *Revenue* for further calculations.

Next, we use the *Math Formula* nodes to divide *HC* by *Total HC* and, consequently, *Revenue* by *Total Revenue*, for each entity.

3. Calculate Recharges by Entity

In a nutshell in this block, we combine two branches from above and generate the data table where we can see the recharge by entity.

How do we do that?

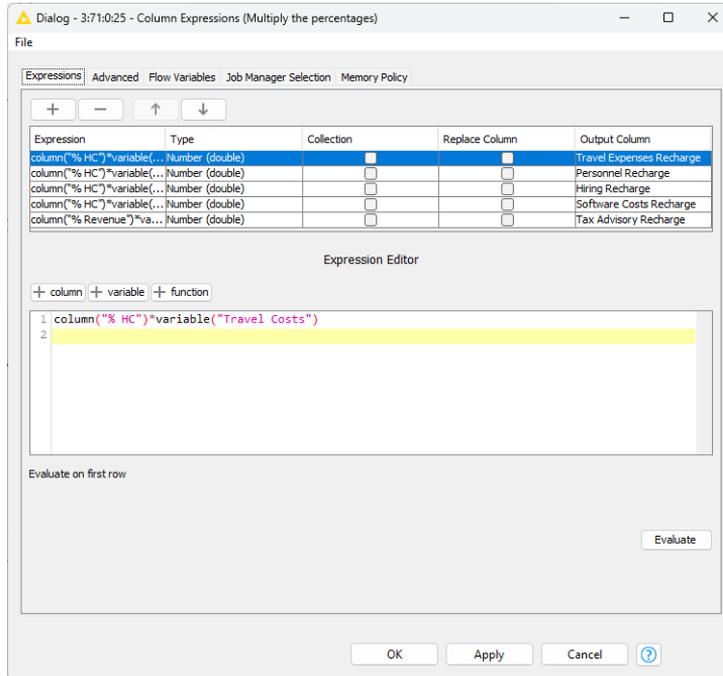
To allow the input data to have various numbers of accounts, i.e., and not just the 5 in this example, we use the *Table Column To Variable* node to convert all values in the "Recharge with Markup" column to flow variables for further calculation. Since this node will call the flow variables with the RowID column values, for the sake of understanding, we use a *RowID* node before, which replaces RowID values with the ones from the "Account" column.

Tax

Transfer Pricing Recharge

Then with a very powerful KNIME node – *Column Expression* – we create 5 expressions, which you can see in the figure below.

The reasoning behind those calculations is the following: We have various cost drivers and we also have various activities. Based on economic cost allocation logic we identify which cost driver drives which activity.



How we configured the Column Expressions node.

Then, we round the calculated Recharges to the full USD values, keep only relevant columns and filter out the US HQ row, as no booking is needed against itself.

4. Save the resulting Table

We write the data into an excel sheet, but that could be any other data source.

5. Visualize Results in a Dashboard

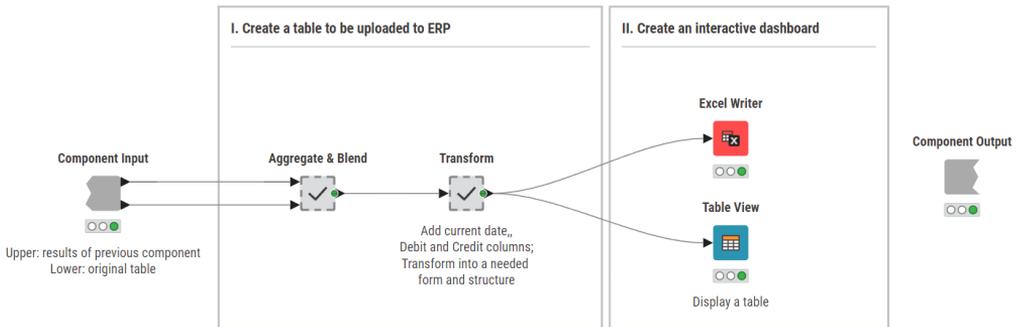
We create visualizations to be displayed in the component's view: final table and the pie chart with Recharges per Entity. You can view the results of calculations in the table below.

Entity	Travel Expenses Rechar...	Personnel Recharge	Hiring Recharge	Software Costs Recharge	Tax Advisory Recharge
GER	613	1,283	153	358	13,197
CH	920	1,924	230	537	14,516
France	1,533	3,207	383	894	9,238
Belgium	307	641	77	179	3,959
Italy	2,760	5,773	690	1,610	13,197
US non HQ	6,133	12,829	1,533	3,578	26,393

Recharges by entity.

A Component to create the Accounting Booking and Upload it to the ERP system

The second component is dedicated to creating a correctly formatted accounting entry (see below). It has two input ports: one comes from the previous component and contains the data table with Recharges by Entity, the second connects to the first imported Costs Data.



Component "Create Accounting Booking to be Uploaded to ERP".

In the *Aggregate & Blend* metanode we remove repeated columns, transpose the table and calculate the sum by cost type, i.e. for each Account's Recharge. We then rename the "Account" values for documentary purposes and then we join this resulting table with the original table with totals calculated.

Inside the second metanode "Transform" we add:

- the current date to each entry - to mark the date of booking;
- "Debit" and "Credit" columns to specify from which accounts we credit and to which we debit;
- and the column "Booking Text" with the constant values "Intercompany Recharge", for documentary purposes.

And here we have it, the final table:

Rows: 10 | Columns: 8

Date String	Journal String	Account String	Cost Center String	Credit Number (double)	Debit Number (double)	Currency String	Booking Text String
2024-04-22	OPEX	Travel Costs	HR Group	12,266	0	USD	Intercompany Recharge
2024-04-22	OPEX	Personnel Costs	HR Group	25,657	0	USD	Intercompany Recharge
2024-04-22	OPEX	Hiring Costs International	HR Group	3,066	0	USD	Intercompany Recharge
2024-04-22	OPEX	Software Costs International	IT Group	7,156	0	USD	Intercompany Recharge
2024-04-22	OPEX	Tax Advisory International	Tax Group	80,500	0	USD	Intercompany Recharge
2024-04-22	OPEX	Intercompany	HR Group	0	12,266	USD	Intercompany Recharge
2024-04-22	OPEX	Intercompany	HR Group	0	25,657	USD	Intercompany Recharge
2024-04-22	OPEX	Intercompany	HR Group	0	3,066	USD	Intercompany Recharge
2024-04-22	OPEX	Intercompany	IT Group	0	7,156	USD	Intercompany Recharge
2024-04-22	OPEX	Intercompany	Tax Group	0	80,500	USD	Intercompany Recharge

Accounting Bookings to be uploaded to ERP.

Tax

Transfer Pricing Recharge

As a last step of this component, we save this resulting table as an .xlsx file, imitating the upload to ERP.

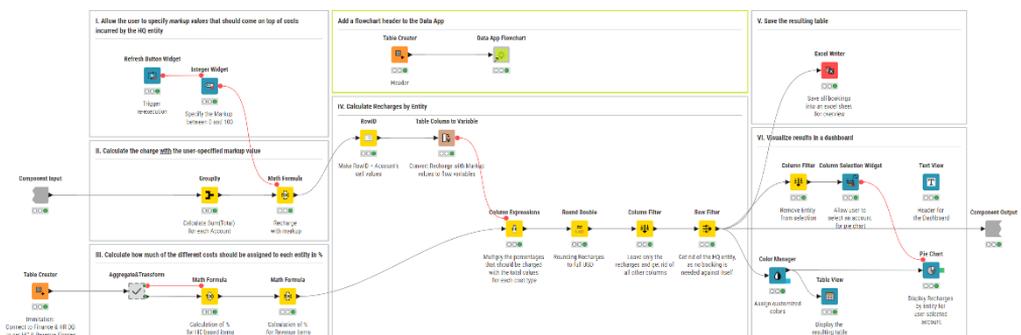
A Data App for Transfer Pricing Recharge

Wouldn't it be nice to have this workflow change dynamically, depending on a non-technical user input? For instance, if an accountant needs to specify new markup values, they would be able to do this via a simple user interface in a browser-based [data app](#), without having to access the underlying workflow.

We want to enable the user to specify the markup value, control for which account the Pie Chart displays the recharges, and download the final report. We'll use *Widget* nodes:

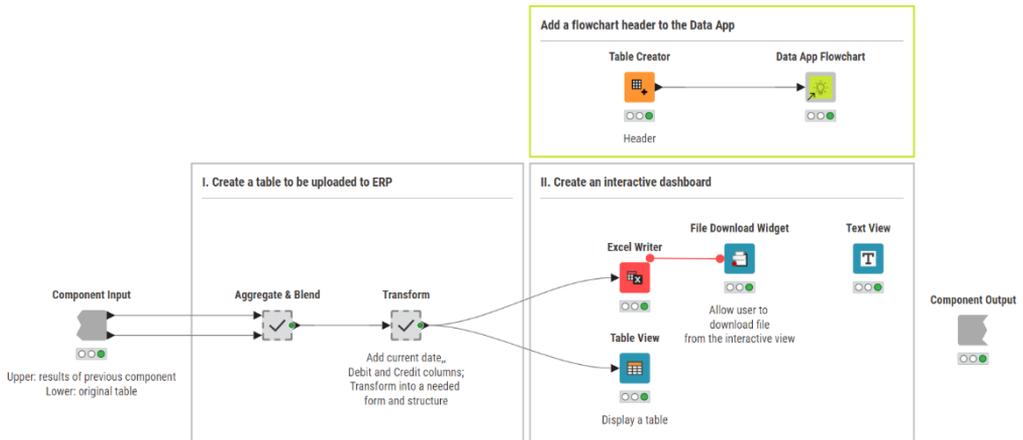
- To allow the user to specify the markup value (in %) that should come on top of costs incurred by the HQ entry, we have to adjust the first component by adding the *Integer Widget* node and the *Refresh Button Widget* node, to trigger re-execution. The value specified by a user would then as a flow variable overwrite the markup value used for calculations in the *Math Formula* node.
- To allow the user to select for which account to display recharges per entity in the Pie Chart node, we add the *Column Selection Widget* node to the first component.
- To allow the user to download the table prepared for ERP, in the second component we add the *File Download Widget* node.

We add headers to both components with *Text View* nodes. We also give our data app a final touch by adding a flowchart header. This is handled by the [Data App Flowchart](#) component. Below you can see both components with *Widgets* added.



Component "Enter Markup and Calculate Recharge by Entity" with Widgets added.

Tax Transfer Pricing Recharge



Component “Create Accounting Booking to be Uploaded to ERP” with Widgets added.

If you then deploy this Data App on the KNIME Business Hub, these two components, interactive dashboards, will be two pages of the Data App, allowing users to first preview the results based on the input markup and, then, switch to the next page which will create the table of the Accounting Bookings prepared for upload to ERP and allow them to directly download the resulting table.

The data app workflow is available and free to download from the KNIME Community Hub [here](#). This workflow has the same structure as its static versions. However, here, allow more interactivity.

The Results: Shareable Data App for Price Transfer Recharge

Now, let’s deploy these two data apps on [KNIME Business Hub](#) and see how they will look for an end user.

Note. Learn how to deploy a data app to KNIME Business Hub with the “[Deploy a data app on KNIME Business Hub](#)” video on KNIMETV.

In the gif below, we show how the data app works. The two pages correspond to the two components we created in KNIME Analytics Platform.

Tax
Transfer Pricing Recharge

Page 1

The first page of the data app is the result of the “Enter Markup and Calculate Recharges by Entity” component. It lets the end user define the markup value and based on that calculates the recharges by entity, shown in the table view and visualized by the pie chart



□
Calculate Recharges

Form the Report

Recharges by Entity

Specify Markup

15

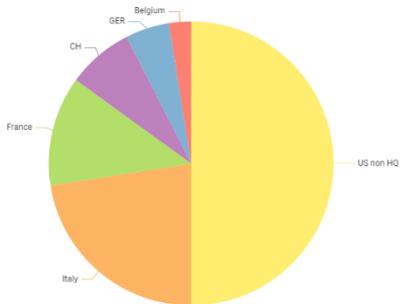
Recalculate Recharges

Entity	Travel Expenses Recharge	Personnel Recharge	Hiring Recharge	Software Costs Recharge	Tax Advisory Recharge
GER	613	1,283	153	358	13,197
CH	920	1,924	230	537	14,516
France	1,533	3,207	383	894	9,238
Belgium	307	641	77	179	3,959
Italy	2,760	5,773	690	1,610	13,197
US non HQ	6,133	12,829	1,533	3,578	26,393

Select the Account to display Recharges by Entity

Travel Expenses Recharge

↓

CancelNext

Page 1 of the data app calculates and displays the recharge by entity, calculated based on the user defined markup value.

Tax
Transfer Pricing Recharge

Page 2:

The second page of the data app is the result of the “Create Accounting Bookings to be uploaded to ERP” component. It is dedicated to creating the correctly formatted accounting entry to be uploaded to the respective ERP system.

Open for Innovation
KNIME

Calculate Recharges

Form the Report

Accounting Bookings to be uploaded to ERP

Click here to download the document
[Download \(xlsx\)](#)

Rows: 10 | Columns: 8

Date String	Journal String	Account String	Cost Center String	Credit Num.	Debit Num.	Currency String	Booking Text String
2024-06-27	OPEX	Travel Costs	HR Group	12,266	0	USD	Intercompany Recharge
2024-06-27	OPEX	Personnel Costs	HR Group	25,657	0	USD	Intercompany Recharge
2024-06-27	OPEX	Hiring Costs International	HR Group	3,066	0	USD	Intercompany Recharge
2024-06-27	OPEX	Software Costs International	IT Group	7,156	0	USD	Intercompany Recharge
2024-06-27	OPEX	Tax Advisory International	Tax Group	80,500	0	USD	Intercompany Recharge
2024-06-27	OPEX	Intercompany	HR Group	0	12,266	USD	Intercompany Recharge
2024-06-27	OPEX	Intercompany	HR Group	0	25,657	USD	Intercompany Recharge
2024-06-27	OPEX	Intercompany	HR Group	0	3,066	USD	Intercompany Recharge
2024-06-27	OPEX	Intercompany	IT Group	0	7,156	USD	Intercompany Recharge
2024-06-27	OPEX	Intercompany	Tax Group	0	80,500	USD	Intercompany Recharge

← Back

Page 2 of the data app creates the correctly formatted accounting entry to be uploaded to the respective ERP system.

3 Key Benefits of the KNIME solution

- The intuitive low-code environment enables non-tech savvy users to easily update calculation variables, recalculate, and receive the bookings to be performed in the ERP.
- The dynamic data app allows the user to perform operations via a user-friendly interface; more plots and charts could also be added to provide an even more robust and comprehensive view.
- The solution is easily connected to a REST API or DB by replacing the Excel reader node.

KNIME for Finance

KNIME Analytics Platform is an open-source free low code platform that opens up access to advanced analytics techniques to anyone who wants to make sense of data. Thanks to its visual and intuitive user interface, implementing solutions does not require coding expertise.

Explore the [KNIME for Finance](#) space on KNIME Community Hub for more low-code/no-code examples.

Key Performance Indicators

With KNIME, financial analysts can report and visualize key performance indicators (KPIs). These KPIs provide a holistic view, align the entire organization, and inform strategic business decisions. In this chapter, we introduce some workflow examples for some key performance indicators (KPIs) that the finance team might need to report and visualize on a regular basis.

This chapter includes the articles:

- Customer Churn KPI, p. 104
- Employee Turnover KPI, p. 110
- Revenue Growth KPI, p. 117

Customer Churn KPI

Author: Luca Porcelli

Workflow on KNIME Community Hub: [Customer Churn KPI Monthly](#)

Customer churn is one of the most important KPIs because it monitors the health of the relationship between the company and its customers.

When a customer cancels, or does not renew, the contract is said to churn. Customer churn measures the percentage of cancelled or not renewed contracts on a given period.

- If **customer churn is low**, it means that the company's product has gained the trust of the customers. This implies positive customer satisfaction and a solid customer base of loyal customers.
- If **customer churn is high**, it may indicate customer dissatisfaction, problems with the product or the service, or difficulty in maintaining customer loyalty.

In this use case we show how you can calculate the Customer Churn KPI with KNIME Analytics Platform, an open source, low code data science tool.

With the KNIME solution you'll have a shareable solution that calculates the KPI automatically. If the input data for your calculation changes, the KNIME solution automatically recalculates the KPI values without any manual editing. And you can share the solution with colleagues on your finance team via a link to a browser-based data app.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Customer Churn KPI](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Calculate Customer Churn KPI with KNIME

The customer churn KPI is defined as:

$$\text{Customer Churn KPI} = \frac{\text{no. customer who left in period}}{\text{total customers at the start in period}} * 100$$

Since a common timeframe is "month", we'll concentrate on the monthly Customer Churn KPI for the rest of this post.

Key Performance Indicators Customer Churn KPI

The data to calculate the churn on is a long list of contracts (see figure below), with a start date and an end date and some additional information about the bought products and the amounts paid. When the contract from customer X ends and is not renewed, this counts as a churn. This data usually comes from a CRM system, like for example Salesforce.

Customer	Customer ID	Contract ID	Contract Start Date	Contract End Date	Product	Subsidiary	currency	Contract Value local currency	Contract Value EUR
Customer 1	1	11	12/1/2020	12/1/2023	Large	Company 1	USD	75,000.00	
Customer 1	1	12	1/1/2018	1/1/2024	Large	Company 2	CHF	75,000.00	
Customer 1	1	13	12/1/2016	12/1/2020	Small	Company 2	CHF	40,000.00	
Customer 2	2	21	12/1/2019	12/1/2024	Small	Company 1	USD	40,000.00	
Customer 3	3	31	10/1/2019	10/7/2022	Small	Company 2	CHF	40,000.00	
Customer 4	4	41	1/1/2020	1/1/2024	Large	Company 3	EUR	75,000.00	
Customer 4	4	42	8/1/2018	8/1/2024	Large	Company 3	EUR	75,000.00	
Customer 4	4	43	5/1/2023	5/1/2024	Large	Company 3	EUR	75,000.00	
Customer 5	5	51	5/1/2020	5/1/2024	Small	Company 1	USD	40,000.00	
Customer 6	6	61	3/1/2017	3/1/2023	Small	Company 2	CHF	40,000.00	
Customer 7	7	71	2/1/2017	2/1/2024	Small	Company 2	CHF	40,000.00	
Customer 7	7	72	5/1/2023	5/1/2024	Small	Company 1	USD	40,000.00	
Customer 8	8	81	12/1/2016	12/1/2023	Small	Company 1	USD	40,000.00	
Customer 9	9	91	7/1/2015	1/1/2020	Large	Company 3	EUR	75,000.00	
Customer 10	10	101	1/1/2018	1/1/2024	Large	Company 2	CHF	75,000.00	

An example of customer data from an HR system.

Let's look at the steps to calculate customer churn KPI:

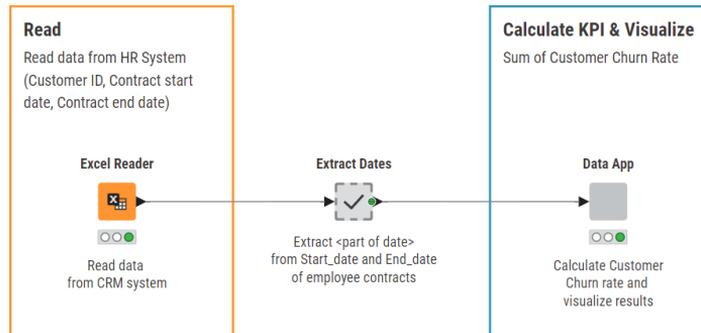
1. Collect the customer data and their contracts from the CRM system
2. Define the start date and the end date for each customer, also considering overlapping contracts
3. Count customers with start date and end date during the selected month
4. Count the total number of customers in each month
5. Compute the monthly Customer Churn KPI, according to the formula
6. Visualize the KPI in a web-based data app

The Workflows: Calculate and Visualize monthly Customer Churn with a Data App

The [Customer Churn KPI Monthly](#) workflow calculates the Customer Churn KPI for a company. It's available for free download from the KNIME Community Hub. The workflow follows these steps:

- **Read** data from the CRM system with the appropriate reader node, like the *Salesforce Simple Query* node or a *GET Request* node. Here, we exported the data to an Excel file and used an *Excel Reader* node to read it
- **Extract** <Part of Date>, Year, Quarter, and Month from columns Start_Date and End_Date from customer contracts
- **Calculate and visualize** the KPI with a browser-based data app

Key Performance Indicators Customer Churn KPI



The workflow "Customer Churn KPI Monthly" to calculate the monthly Customer Churn KPI for a fixed year.

The data app component (on the right):

- Enables the user to select the year for the calculation of the KPI values via the menu generated by a [Single Selection Widget](#) node.
- Extracts the data for the selected year.
- Calculates the start and end date for each customer, also considering duplicated and overlapping contracts (In the "Duplicate Row Filter" metanode).
- Counts the number of customers who churned in each month (In the "counts" metanode).
- Calculates the total number of customers in each month (In the "Customer" metanode).
- Calculates the monthly customer churn KPI (in the "Customer Churn" metanode).
- Prepares the data for visualization.
- Visualizes the KPI values in a web-based dashboard

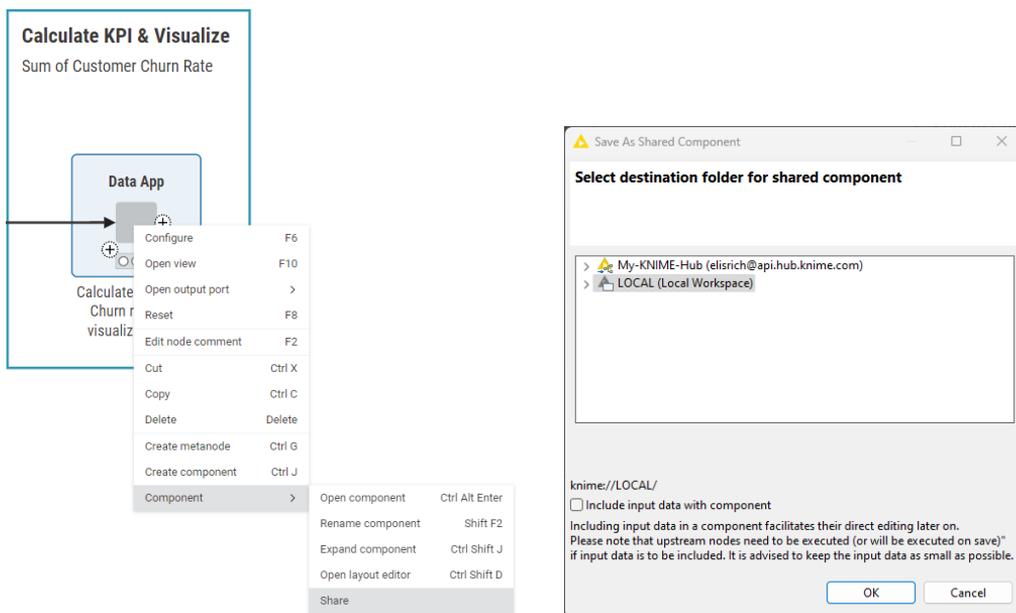
If you've seen more of our "KNIME for Finance" articles, the formula displayed above for Customer Churn KPI will remind you of the formula for the [Employee Turnover KPI](#). They are very similar, though not identical. The visualization however could be exactly the same. To avoid reinventing the wheel, we decided to adopt the "Data App" component from the previous employee turnover KPI workflow.

Reuse Functionality with Shared Components

In KNIME Analytics Platform, it is possible to share functionality with other workflows/users by creating a component. This is how you do it:

- Open the workflow and select the component
- Right-click the component and select "Component" → "Share"

Key Performance Indicators Customer Churn KPI



To share a component with others, right-click the component and select "Component" > "Share".

Select the location where to store the component, either locally or on your KNIME Hub space.

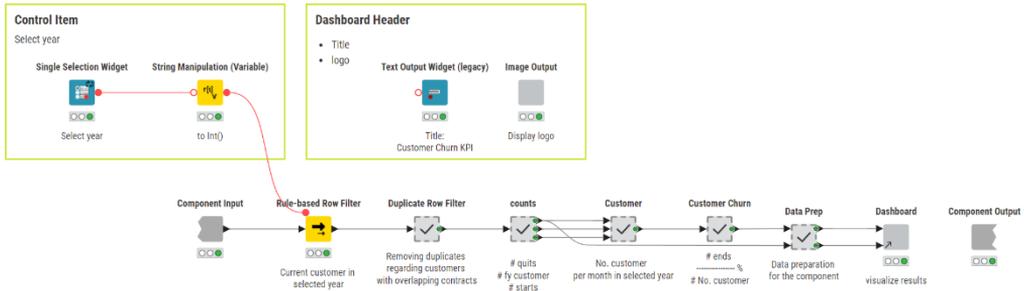
- You will be prompted to select the location, on your local workspace or on your KNIME Hub space, where to store the component template

To use a linked version of the component template:

- Navigate to the location where the component template has been saved
- Drag & drop the component template in your workflow.
- A linked version of the component template will be created in your workflow.
- The linked version of the component is write-protected. If you want to change it, you need to disconnect the linked component from its template and create a local independent copy. To do that, right-click on the linked component and select "Disconnect link".

Following this procedure, we created the component template "Data App" in [Employee Turnover KPI Monthly](#), which is a public space on the KNIME Community Hub, and then we have created the linked component in our workflow. You can recognize that the "Data App" component in the figure below is a linked component from the arrow in the right lower corner.

Key Performance Indicators Customer Churn KPI



The workflow inside the data app component.

The Results: Explore KPI Year-on-Year via Interactive Data App

We've shown how you can calculate the Customer Churn KPI and how to reuse a shared component from an existing workflow.

The visualization is indeed the same as for our KNIME solution for Employee Turnover KPI: because we reused the visualization component as a linked component. The shared component view includes a line plot to monitor the customer churn KPI trend and a pie chart to describe the customer churn for different office locations, exactly as it showed the trend and the pie chart for employee turnover KPI.

Looking at the view from the data app, you will notice that the Customer Churn KPI at the end of the year 2023 does not stay within the boundaries of acceptable values. This trend must be investigated and actions must be taken.

Customer Churn KPI

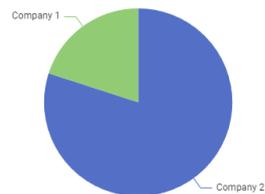
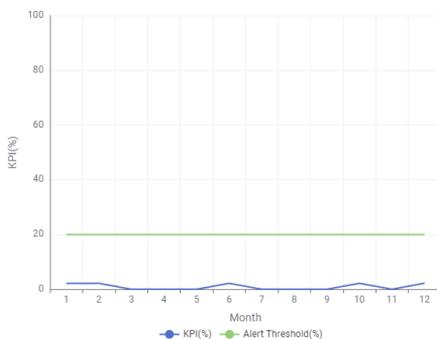


Select year:

- 2021
- 2022
- 2023

KPI over months

Pie Chart Visualization



The data app shows the monthly plot and company chart for the Customer Churn KPI.

KNIME for Finance: Automated & Reusable Solutions

KNIME Analytics Platform gives you an intuitive user interface to access advanced data science techniques. Here we showed you a simple example of calculating and visualizing customer churn with this [KNIME workflow](#).

The KNIME solution has three key benefits:

- It's shareable and user-friendly: You can share the link to the browser-based data app with other users
- It's automated: In scenarios where the input data changes, the KNIME workflow automatically recalculates the KPI values without the need for any manual editing
- It's connectable: With access to 300+ data sources and web APIs, you could connect to your data, independently of its location.

Try out the [workflow](#) yourself. If you're new to KNIME, here's the link to [download KNIME](#) (it's open source and free to use).

Employee Turnover KPI

Author: Luca Porcelli

Workflow on KNIME Community Hub: [Employee Turnover KPI Monthly](#) & [Employee Turnover KPI PDF](#)

Company KPIs are mostly associated with financial information. However, there is a set of non-financial KPIs that are used to monitor a company's health. One of those is employee turnover.

Shifts in the number of employees are what we call Employee Turnover. If Employee Turnover is low, the company has been successful in retaining its workforce throughout the year. This indicates a positive work environment, satisfied employees, and a stable team. A high employee turnover can indicate dissatisfaction among employees, issues with work culture, or challenges in retaining talent.

While some turnover is inevitable within the normal functioning of a company, it can become worrisome if the rate grows too fast and passes an alarming value, or if the trend is negative. The employee turnover KPI can be used to initiate action to prevent negative trends and revert to healthier values.

In this use case we want to show you how you can calculate Employee Turnover KPI with KNIME.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Employee Turnover KPI](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Calculate Employee Turnover KPI

We want to build a KNIME solution that will automatically calculate a monthly employee turnover KPI.

Employee turnover KPI is defined as:

$$\text{Employee Turnover KPI} = \frac{\text{no. employees who left in period}}{\text{average no. of employees in period}} * 100$$

Let's look at the steps to calculate the Employee Turnover KPI.

Key Performance Indicators

Employee Turnover KPI

- Retrieve employee data from the HR System
- Count all employees who joined and left in the selected year
- Calculate the average number of employees per month, taking into consideration new hires and quits
- Compute the monthly Employee Turnover KPI, according to the formula
- Visualize the KPI in a dashboard that can be shared via a browser-based data app

You're typically using internal data to calculate employee turnover KPI e.g., human resources records, time monitoring systems, supervisor feedback, and workforce management platforms.

Below is a screenshot of a typical employee dataset coming from an HR system. Notice the columns *Start_Date* and *End_Date*, showing respectively the date of start and termination of the employment contract. Other columns in the dataset include salary amount, salary currency, and employee ID.

Employee	Employee ID	Start Date	End Date	Location	Cost Location	currency	Base Salary local currency	Bonus local currency	Base Salary EUR	Bonus EUR
Employee 1	1	2020-12-01	2023-02-01	New York	Sales	USD	50000	35000		
Employee 2	2	2019-12-01		New York	R&D	USD	65000	12000		
Employee 3	3	2019-10-01		Zurich	Marketing	CHF	45000	8000		
Employee 4	4	2020-01-01		Rome	Marketing	EUR	49000	10000		
Employee 5	5	2020-05-01		New York	Finance	USD	51000	13000		
Employee 6	6	2017-03-01	2023-02-01	Zurich	Sales	CHF	78000	40000		
Employee 7	7	2017-02-01		Zurich	R&D	CHF	54000	15000		
Employee 8	8	2016-12-01		New York	HR	USD	52500	11000		
Employee 9	9	2015-07-01	2020-01-01	Rome	R&D	EUR	61000	23000		
Employee 10	10	2018-01-01		Zurich	Sales	CHF	77000	46000		
Employee 11	11	2017-12-01		Rome	R&D	USD	80000	24000		
Employee 12	12	2017-04-01	2023-02-01	Rome	Sales	USD	73000	43000		
Employee 13	13	2016-09-01		Zurich	IT	CHF	55000	20000		
Employee 14	14	2016-11-01	2020-01-01	Zurich	Sales	CHF	82000	37000		
Employee 15	15	2015-04-01	2022-12-01	New York	R&D	USD	66000	15000		
Employee 16	16	2018-08-01		New York	Support	USD	40000	8000		
Employee 17	17	2017-04-01	2023-05-01	Rome	Support	EUR	38000	5000		

An example of employee data from an HR system.

The Workflows: Calculate & Visualize monthly Employee Turnover KPI with a Data App

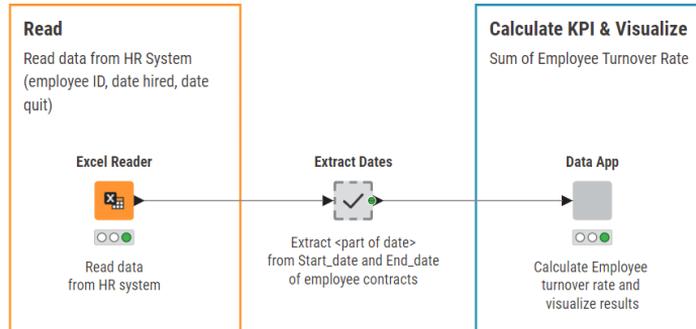
The KNIME workflow [Employee Turnover KPI Monthly](#) calculates the Employee Turnover KPI for a company and is available and free to download from the KNIME Community Hub.

The workflow includes the following steps to:

- **Read data** from the HR system with the appropriate reader node. In our example, we exported the data to an Excel file and therefore we used an *Excel Reader* node. The wizard node here to access an HR system live is the *GET Request* node, that implements a GET Request for a REST service.

Key Performance Indicators
Employee Turnover KPI

- **Extract <Part of Date>**, Year, Quarter, and Month from columns *Start_Date* and *End_Date* from employee contracts.
- **Calculate and visualize the KPI within a Data App.** This is the last component in the workflow.



The Employee Turnover KPI Monthly workflow to calculate the monthly Employee Turnover KPI for a fixed year.

The data app component:

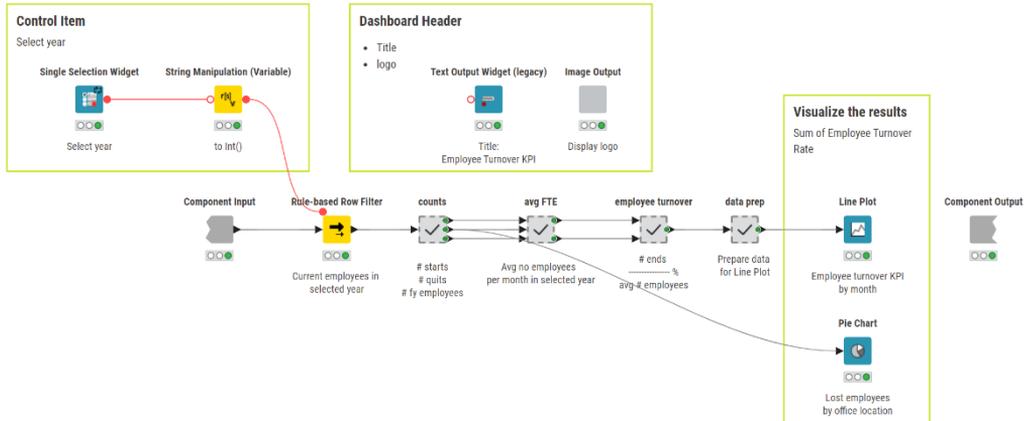
- Enables the user to select the year for the calculation of the KPI values via a *Single Selection Widget* node, which generates a dropdown menu showing the years you can select from.
- Extracts data for the selected year.
- Identifies all employees who joined and left in the selected year (In the "counts" metanode) and counts the total number.
- Calculates the monthly average number of employees (In the "avg FTE" metanode) using this formula:

$$\text{average no employees in period} = \frac{\text{no empl beginning period} + \text{no empl. end period}}{2}$$

- Calculates the monthly employee turnover KPI (in the "employee turnover" metanode).
- Prepares the KPI values for visualization, by adding the month names.
- Visualizes the data in a line plot and a pie chart. The line plot displays the trend of the KPI values over months. The pie chart displays the total number of employee departures across different office locations. This is sometimes useful to understand if an office has a worse dynamic than others. Notice that an alarm line has been added to the line plot: if the KPI trend passes this line (which could for example be the external benchmark for that KPI), then the high value of the KPI denotes a worrisome situation within the company.

Key Performance Indicators Employee Turnover KPI

Adds a title to the dashboard (using the [Text Output Widget](#) node) and a company logo. Here we used the "Display Logo" component to display the KNIME logo.



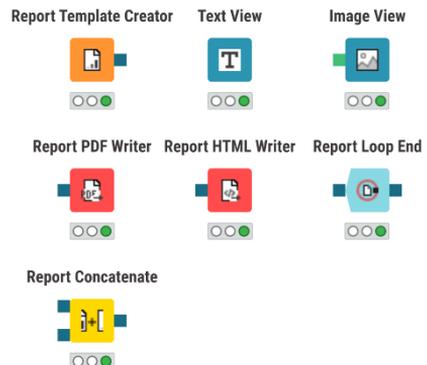
The workflow in the data app component.

Export KPI to a PDF document

While the data app is shareable and lets any user select the year and check out the employee turnover KPI on the fly you might also want to share the KPI as a PDF.

New functionality in the latest release KNIME Analytics Platform 5.2 lets you export a view of the data app into a PDF formatted document.

- The [Report Template Creator](#) node defines the settings for the report, such as page and orientation.
- The [Text View](#) node creates a text display by also changing the format.
- The [Image View](#) node displays an image.
- The [Report PDF Writer](#) writes the data app to PDF format.
- The [Report HTML Writer](#) writes the data app to HTML format.
- The [Report Loop End](#) collects all reports from a loop, showing a summary at the end.
- The [Report Concatenate](#) merges several data apps together into a report.



The PDF nodes allow to format a data app view into a PDF document to create a professional report.

Key Performance Indicators Employee Turnover KPI

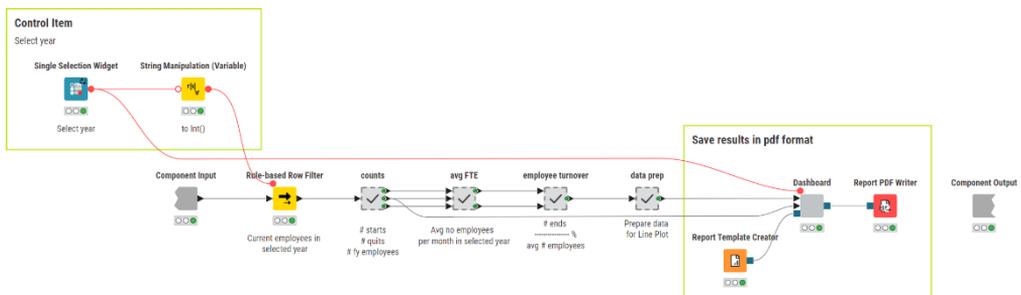
Let's walk through how to create a PDF document from the view of the "Data App" component that will show a line plot and pie chart in a dashboard.

First, create a new component within our "Data App" component with just the *Line Plot* and the *Pie Chart* nodes. We'll call this component "Dashboard" (the workflow in this component is shown below).

Add the *Text View* and *Image View* nodes to make the report more pleasing. Titles and images have been included to ensure a more engaging and easily understandable experience for readers.

Create a report template that we'll use to contain the view from the "Dashboard" component. Do this by adding two nodes: one before (*Report Template Creator*) and one after (*Report PDF Writer*) the "Dashboard" component.

This workflow, [Employee Turnover KPI PDF](#) now performs the same KPI calculations as the previous workflow, [Employee Turnover KPI Monthly](#), but it exports the view of the "Dashboard" component to a PDF report. These two workflows are freely accessible for you to download from KNIME Community Hub.



The workflow in the Dashboard component to create a PDF report for your Employee Turnover KPI.

The Results: Explore the KPI Year by Year with an Interactive Data App and/or PDF Report

In this article, we've shown how to build an interactive data app that calculates the monthly Employee Turnover KPI, based on the year selected by the user, and transform the data app view for a selected year into a static PDF report. The screenshot below shows the interactive data app.

You can see that while the Employee Turnover KPI remains within acceptable boundaries on average, there are problematic office changes from year to year.

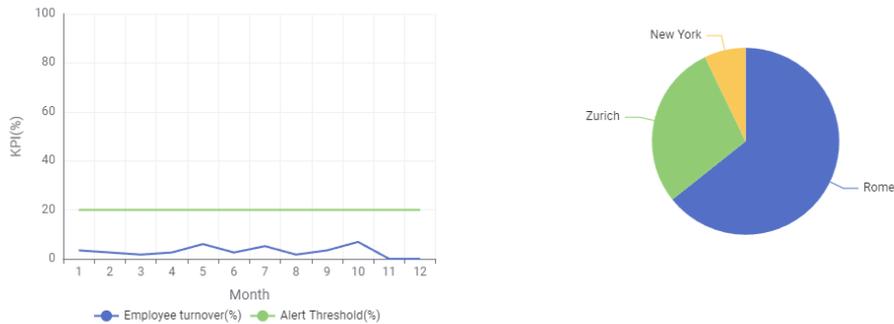
Employee Turnover KPI

Select year:

- 2021
- 2022
- 2023

Employee Turnover KPI over months

N° of terminations by office location



The data app displays the monthly plot and the location office chart for the Employee Turnover KPI.

4 Key Benefits of the Solution at a Glance

- It's **shareable and interactive**: You can easily share the browser-based data app with colleagues and they can dive into the analysis and get the KPI for the year they're interested in.
- It's **easy to integrate** new charts & diagrams: KNIME offers a wide range of visualization options, which makes it easy to extend the current data app and build out your dashboard to provide a more comprehensive view of the business trends.
- It's **automated**: When your input data varies, the KNIME workflow recalculates the KPI without you needing to do any manual adjustment, typical when working with spreadsheets and macros.
- It's got **universal connectivity**: With access to 300+ data sources plus community-driven APIs, you'll be able to connect up to your data wherever it is.

KNIME for Finance

KNIME is a low-code/no-code data science tool. The visual interface is user-friendly and lowers the entry barrier to using data science techniques. It enables you to quickly develop automated solutions as you don't need any programming experience to start using it.

[Download KNIME](#) (it's open source and free to use) and explore the [workflows](#) described in the article.

Revenue Growth KPI

Authors: Rosaria Silipo & Ralf Gruesshaber

Workflows on KNIME Community Hub: [Revenue Growth KPI Monthly](#) & [Revenue Growth KPI Monthly Static](#)

Companies rely on a set of standard Key Performance Indicators (KPI) metrics to measure the health of their business. KPIs measure revenues, employee retention, customer satisfaction, brand awareness, auditing, and more.

Many of these KPIs are already included in many commercial tools, like CRM for customers, ERP systems for revenues, or HR platforms for employees. Finance teams could gain a truly comprehensive view of the company's status if they were able to calculate all their KPIs from different sources and display them all in the same report, dashboard, or shareable Data App.

In this article, we want to learn what the revenue growth KPI is and how to calculate it automatically with KNIME. Using KNIME, as a data science tool, gives us access to techniques to collect all our data from disparate sources and build that comprehensive view of the company's status.

Note. The KNIME for Finance YouTube playlist on [KNIMETV](#) contains videos about using KNIME to tackle common tasks in finance departments. Each video introduces a specific topic and demonstrates a solution with KNIME Analytics Platform. Watch the [Revenue Growth KPI](#) video to get an overview of this solution or browse the playlist for other solutions.

The Task: Calculate Revenue Growth KPI

Revenue Growth KPI measures the increase in revenues from one period to the next. It's the most common KPI used by CEOs to explain the status of the business.

If we consider a particular business year, the period we want to measure could be month, quarter, or even day. Month is the most commonly adopted period for Revenue Growth KPI calculation.

Revenue growth rate is defined as:

$$\text{Revenue Growth KPI} = \frac{\text{Current period revenue} - \text{Previous period revenue}}{\text{Previous period revenue}}$$

Key Performance Indicators Revenue Growth KPI

If we take a typical journal in a finance department, we see a list of sales and expenses, accounted for respectively in the “credit” and the “debit” column. The revenues are the amounts in the “credit” column.

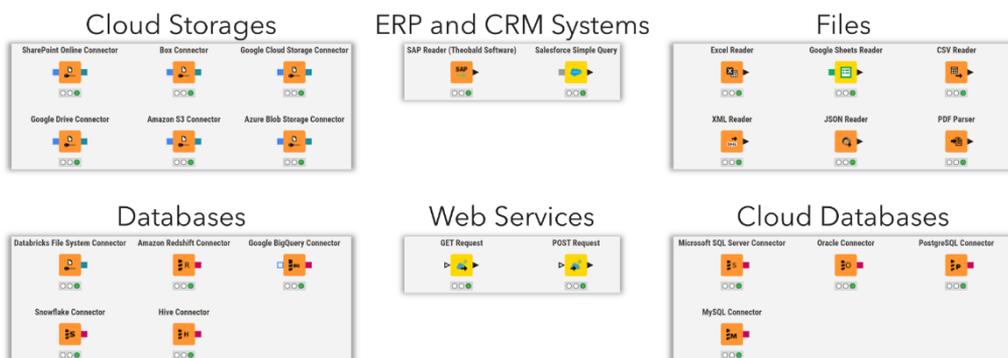
Both amounts, for debit and for credit, correspond to the original amount reported in the invoice or in the sale contract converted to the company adopted currency. Our data then comes from at least two sources: the company’s entry journal on an ERP system, e.g., SAP, Odoo, Excel files, etc., and an external service for the daily exchange rate.

If invoice documents and sale contracts reside on different platforms, the number of data sources might increase. The process, however, does not change. That is:

- Pull the journal entries from the ERP system.
- Collect the exchange rates from an external service
- Multiply the original amounts by the date’s exchange rate and populate the “credit” / “debit” columns.
- Aggregate the credit entries (i.e. revenues) by the selected period. Let’s build a monthly KPI here, with period = month.
- Calculate Revenue Growth KPI.
- Visualize the KPI.

Thanks to the many connector nodes available within KNIME Analytics Platform, native as well as developed by the KNIME Community, the journal entry data could be pulled out of SAP using the *SAP Reader* node, out of Odoo using the *GET Request* node, out of a database using the DB nodes, or simply out of an Excel file using the *Excel Reader* node.

In order to make this example work on everybody’s laptop, without the need of accounts and credentials, we used an Excel file as our accounting journal in the finance department.



A collection of common KNIME connector nodes to connect to various data sources like cloud storages or databases.

Key Performance Indicators Revenue Growth KPI

Below is a screenshot of the simulated journal entry data that we used for this use case.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Partner	Ref_ID	Product	Unit	Price	Amount (original currency)	Currency	Amount in Currency	CHF Date	Subsidiary	Journal	Journal Entry Account	Debit	Credit	Cost Center	
2	Company 1	ID_123	Piano	5	10000	50000	CHF		6/12/2022	Enterprise A		1234		50000	AB	
3	Company 2	ID_124	Guitar	2	200	400	CHF		1/27/2022	Enterprise B		1225		400	AB	
4	Company 3	ID_125	Saxophone	3	1000	3000	CHF		9/14/2022	Enterprise C		1322		3000	AB	
5	Company 4	ID_126	Pens	50	0.2	10	EUR		3/15/2022	Enterprise C		2496	10		CC	
6	Company 5	ID_127	Notes	100	1	100	EUR		4/16/2022	Enterprise B		2756	100		CE	
7	Company 3	ID_128	Piano	1	10000	10000	CHF		11/17/2022	Enterprise A		1234		10000	AB	

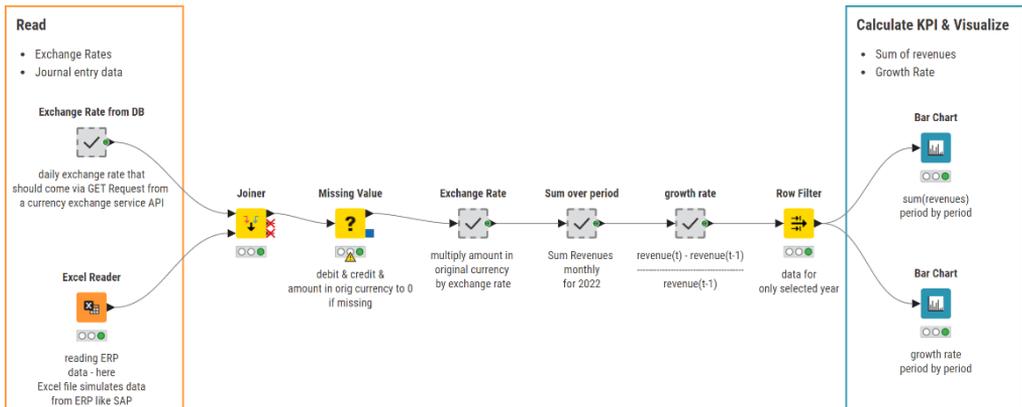
An example of journal entries pulled from an ERP system.

The Workflow: Calculate Monthly Revenue Growth KPI for a fixed Year

The simplest solution workflow [Revenue Growth KPI Monthly Static](#) is available and free to download from the KNIME Community Hub. With this workflow you can:

- **Read journal entry data** from SAP, Salesforce, or an Excel file with the appropriate reader node. In our example, we used an Excel file.
- **Collect the exchange rates** via REST API to an external service. Since many exchange rate API services are for payment, we simulated the service in this workflow with an SQLite database.
- **Join** the two data tables – journal entries and exchange rates – by date to obtain one single global data table.
- **Fill the missing values** in “credit” and “debit” column with 0s via the Missing Value node (we assume that no payment or no sale was made if the item is not present in the column).
- **Adjust the amount** in “credit”/“debit” according to the exchange rate, where needed.
- **Aggregate Revenues** monthly.
- **Calculate Revenue Growth KPI**
- **Visualize the KPI via bar chart** for a specific year

Key Performance Indicators Revenue Growth KPI



The workflow, [Revenue Growth KPI Monthly Static](#), to calculate monthly the Revenue Growth KPI for a given year and visualize the results in a Bar Chart.

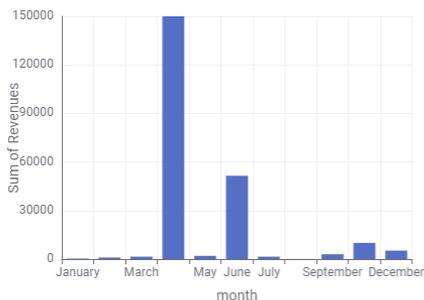
The Data App: Generate dynamic Visualization of KPI by Year

The previous solution is quite simple and already does what is expected. Let's now create a dashboard from the two bar charts and add a title and logo. We do this by wrapping the relevant nodes into a component, which we'll call our "data app".

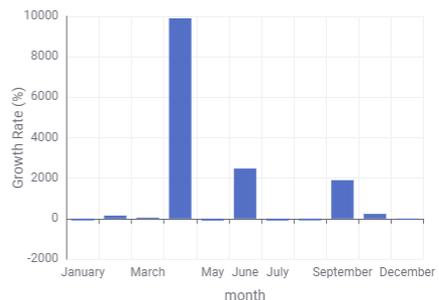
The nodes we'll need for our data app are:

- The two bar chart nodes and the *Row Filter* node that selects the year
- A *Text Output Widget* node to give the dashboard a title
- The "Display Logo" component, to show a logo image in the dashboard

Sum of Revenues



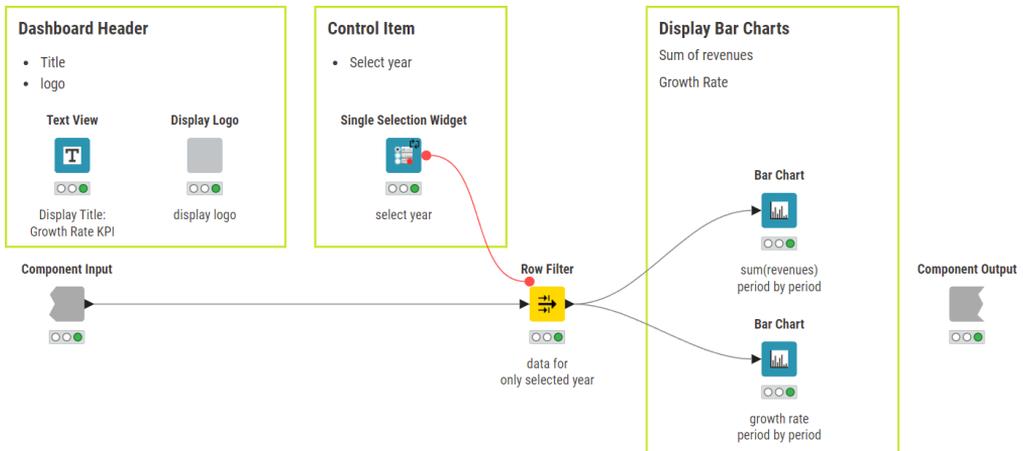
Growth Rate KPI (%)



The two Bar Charts created by the data app component.

Key Performance Indicators Revenue Growth KPI

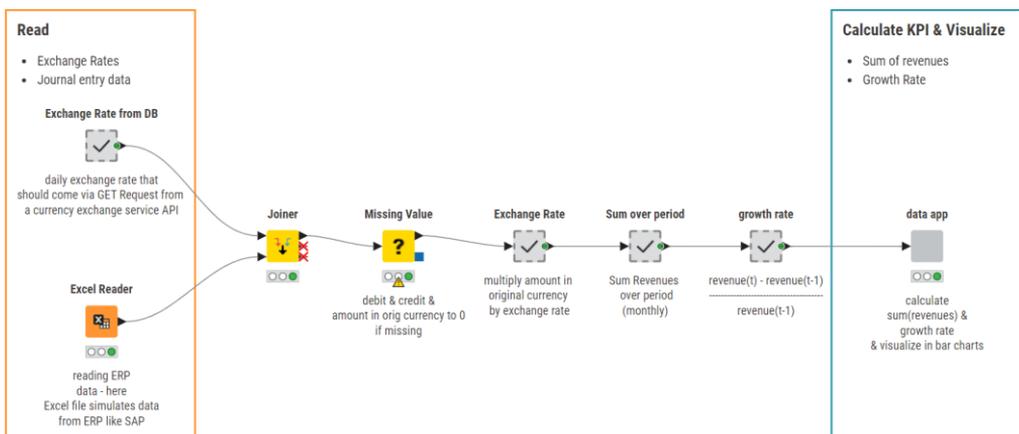
Note. [Components](#) are a powerful feature of KNIME Analytics Platform. While in theory they are just nodes that contain other nodes, in practice, by inheriting the views of the graphical nodes within, they produce powerful dynamic interactive dashboards.



The nodes you need to build your data app that visualizes Revenue Growth Rate as a Bar Chart

Now, let's make the selected year for the visualization a dynamic parameter of the data app. We introduce a *Single Selection Widget* node to include a drop-down menu for the year selection in the data app view. We can then use this parameter, i.e., the selected year, to control the *Row Filter* node before the KPI visualization.

The final workflow [Revenue Growth KPI Monthly](#) (shown below) is available and free to download from the KNIME Community Hub. This workflow has the same core as our simpler workflow, [Revenue Growth KPI Monthly Static](#). However, here the *Row Filter* node, and therefore the KPI visualization, is controlled by the *Widget* node for the year selection (pictured above).

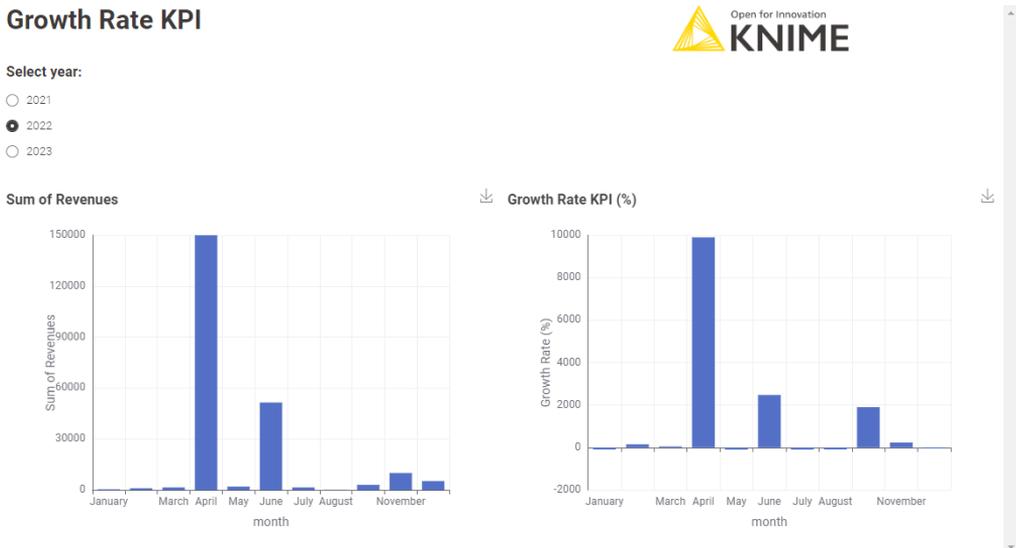


The workflow, [Revenue Growth KPI Monthly](#), to calculate monthly the Revenue Growth KPI for a fixed year and visualizes the results in an interactive data app.

The Results: Explore Revenue Growth KPI with Interactive Data App

In this workflow we have assembled a data app, not only to calculate the monthly Revenue Growth KPI, but also to display it dynamically based on the user selection of the year. The figure below shows the data app and its control options.

Growth Rate KPI



The data app showing the Bar Charts to visualize revenue growth and total revenues. The user can interact with the data app by selecting the year.

Could we have performed the same task with Excel? If we limited ourselves to a static representation of the KPI, surely, we could have. However, this implementation offers a few added benefits:

- A **dynamic data app** allows the user to interact with the dashboard, more plots and charts from other KPIs could also be added to make the business overview more comprehensive, and it can be shared easily with colleagues.
- The possibility to **connect to external sources** via REST API (*GET Request* node) or to database (*DB Reader* nodes) or to specific ERP systems (*SAP Reader* node or other similar nodes)
- And, of course, **automation**. When input data changes the workflow continues to calculate the KPI correctly without needing to adjust macros and rows in the spreadsheet.

This content is also available as a YouTube video [KNIME for Finance – Revenue Growth KPI](#).

KNIME for Finance

KNIME Analytics Platform is an open-source free low code platform. Thanks to its visual and intuitive user interface, you can implement solutions without having to learn to program first. KNIME's user-friendly environment makes it easier to access sophisticated data science techniques which you can use to help with automation, blending data, and building parametric dashboards. This analytic depth and breadth also sets the stage for you to try more complex projects such as building predictive models, or [integrating AI](#).

Node & Component Index

B		I	
Bar Chart	60, 61	Image View	113, 114
C		Integer Widget	98
Choropleth Map	80	Internal Rate of Return (IRR)	53
Color Manager	81	J	
Column Expression	96	Joiner	62, 77
Column Expressions	15	L	
Column Selection Widget	98	Line Plot	60, 61, 114
Concatenate	62	M	
D		Math Formula	77, 78, 95, 98
Date&Time Widget	7, 9, 82	Modified Internal Rate of Return (MIRR)	53
Date&Time-based Row Filter	7	Moving Aggregator	60
DB Reader	122	N	
E		Net Present Value (NPV)	53
Excel Reader ..	6, 7, 59, 75, 90, 92, 105, 111, 118	Number to String	27
Excel Writer	12, 78	Numeric Outliers	33, 39, 42
Extended Internal Rate of Return (XIRR)	53	P	
Extended Net Present Value (XNPV)	53	PDF Parser	12, 41, 42
F		Pie Chart	78, 114
File Download Widget	98	Pivot	60
File Reader	49	R	
File Upload Widget	7, 9, 81, 84	Random Forest Learner	27
G		Random Forest Predictor	27
GET Request	75, 92, 105, 111, 122	Refresh Button Widget	16, 98
Google Authenticator	59	Report Concatenate	113
Google Sheets Connector	59	Report HTML Writer	113
Google Sheets Reader	59	Report Loop End	113
GroupBy	60, 62, 77, 95	Report PDF Writer	113

Report Template Creator..... 113
Row Filter 120, 121
Rule Engine 27

S

Salesforce Simple Query 105
SAP Reader 75, 92, 122
Scorer 33, 35
Single Selection Widget..... 106, 112, 121

T

Table Column To Variable 95
Table Creator 93
Table View 16, 78
Text Output Widget 113, 120
Text View 81, 98, 113, 114
Top k Row Filter..... 78

V

Value Lookup..... 41, 62, 78
Vector Store Retriever 43

KNIME, Automation, and AI

The KNIME for Finance Collection

Most operations in finance departments require precision, correctness, repeatability, and speed. In order to obtain repeatable, reliable, correct, and fast procedures, many companies have already started digital transformation projects for all finance processes. This booklet collects a set of jump-start workflows for the most common finance tasks.

Elisabeth Richer holds a master's degree in Social and Economic Data Science. During her studies, she developed a keen interest in Machine Learning, Deep Learning, and various NLP-related techniques. Her research focused on understanding media bias and examining user behavior on social media. She is part of the Evangelism team at KNIME and works as a Data Science Publisher with a particular focus on the books published under KNIME Press.