Data Science without Programming? The Potential of KNIME

Annika Hamachers
German Police University, Münster
annika.hamachers@dhpol.de

Abstract

This document introduces the free open-source software KNIME® that enables sophisticated data analyses without having to write a single line of code. It will be depicted what KNIME is and how it differs from standard tools from the 'data scientist's toolbox' such as R or Python. After presenting the general workaround in KNIME, I will provide a deeper analysis of KNIME's features and the advantages and disadvantages of this alternative approach to data analyses. In the conclusion, I will share my personal opinion on whom this tool is for and who might not find it convenient.

1 What is KNIME®

KNIME is a highly scalable open-source platform that can easily be used for multiple data analysis tasks. What makes it special is its very intuitive modular setup: In KNIME you build *visual data science workflows* consisting of several separate units (so called 'nodes') that perform the single steps in which you want your data to be processed. In a very comprehensive drag'n'drop-style graphical interface you simply select the nodes you need and chain them together in the right order. And what makes it even more sleek: You can configure those nodes and adjust them to the specific requirements of your analysis without having to write a single line of code.

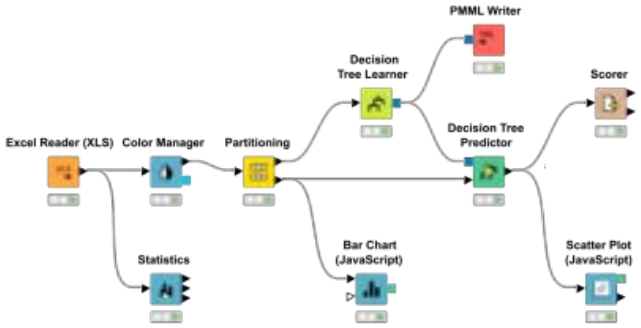

Figure 1: An example workflow in KNIME

The first version of KNIME was already released in 2006 and has steadily grown in its functionality. To date, the tool is (according to the developers' website) used by quite a large community of data scientists in over 60 counties and features almost 4000 nodes and more than 200 downloadable extensions for all kinds of different data processing tasks (from data cleaning and aggregation over various statistical analyses to modules for neat visualization and reporting).

KNIME currently comes in two versions: the free *KNIME Analytics Platform* which lets you run a variety of analyses on your Mac, Windows, or Linux computer and *KNIME Server* which is the enterprise version of it that offers additional features for team-based collaboration, automation of processes, and the creation of data-based applications from workflows. The server version, however, is quite pricy.

If this tutorial gets you curious about KNIME and you want to explore it, too, you can download the free Analytics Prlatform from here.

2 Working with KNIME - The Basics

Working with KNIME is really straight forward and intuitive – though the user interface might look quite opac and intimidating at first. The good thing is, a user will very likly not need to engage with all of the default panels, anyways.

Figure 2: The KNIME GUI

In order to showcase how KNIME works, it's best to demonstrate how to built a very basic example workflow in this environment.

The initial step to get started to create a new KNIME workflow from scratch, is to either select the 'Create new workflow' option from the welcome panel or click 'File → New → New KNIME Workflow'. This opens a window where we can select a name for this workflow and a location where we want to store it:

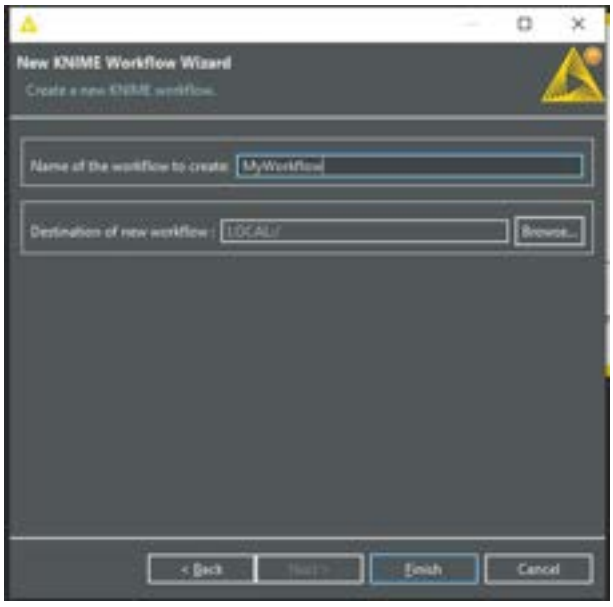

Figure 3: Creating a new workflow

The first thing, I recommend doing on starting a new project in KNIME, is to coarsly pre-structure the workflow a little bit – even before we actually start

building it: Therefore, a user would create so called **workflow annotations** by right-clicking anywhere in the main panel.

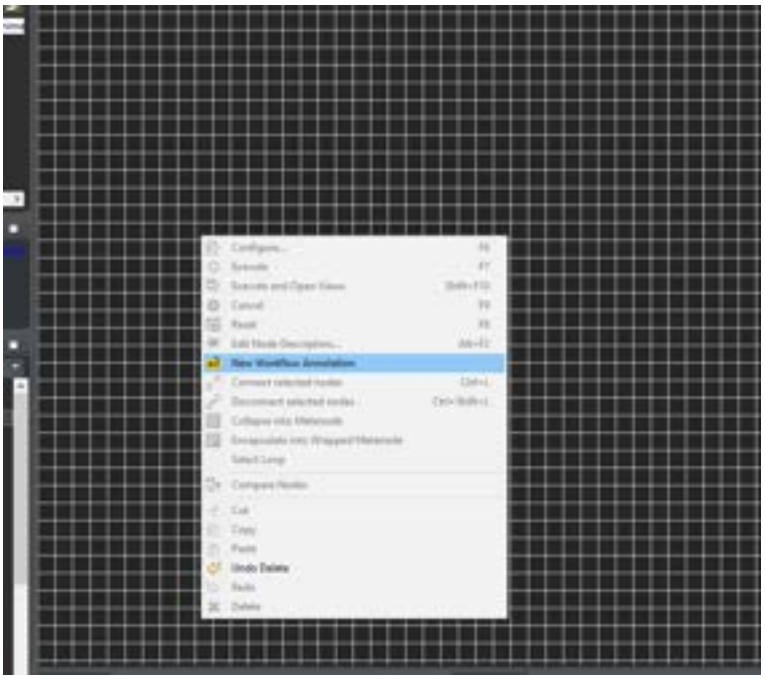

Figure 4: Creating workflow annotations

This creates rectangular shapes that can be adjusted in size and style and to which text can be added that documents and guides the analysis.

To do something useful with KNIME, every workflow will have to consist of at least three parts: loading data into KNIME, do something with those data, and saving the results, so that you can access them later. Accordingly, I like to start with annotations for those three steps. Depending on the complexity of my workflows, those might however be more fine-tuned.

Figure 5: Exemplar annotation blocks

To fill those empty canvases with content and to actually start building a KNIME workflow, the user then has to head over to the **Node Repository** panel.

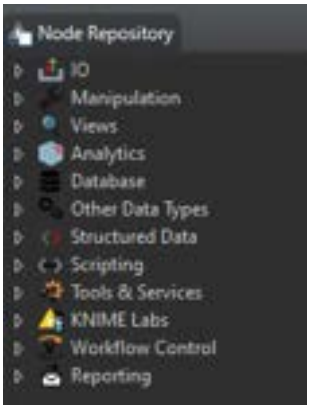

Figure 6: The node repository

Here, you find all the nodes, KNIME currently has to offer ordered by categories and subcategories. The most important categories to start with are certainly **'IO'** which stands for 'input and output' and contains nodes that manage reading in data and saving KNIME output to disk, **'Manipulation'** which contains nodes useful for data preparation tasks such as aggregating, grouping etc. , **'Analytics'** which yields classical statistical analysis but also nodes for more advanced algorithms such as machine learning tasks, and **'Views'** that contains all the nodes that help you create neat graphics from your data.

Navigating this dropdown menu can be quite challenging at first because there are so many nodes, most of which you are certainly not sure what exactly they do. However, KNIME offers help just across your screen: at the other side of your GUI, you'll find the **Node Description** panel which provides you with detailed information on any node you select in the Node Repository (if not, adjust the 'View' settings of your KNIME GUI).

Figure 7: Accessing node descriptions

Moreover, right above the Node Repository, the user can find the **Workflow Coach**. Ifhe or she selects a node in an open workflow, this panel makes suggestions which other nodes would work well with this node.

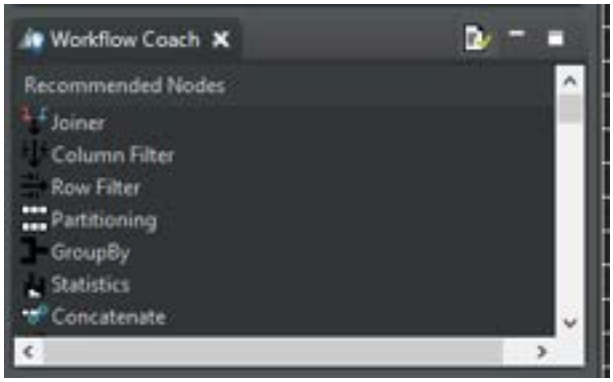

Figure 8: The workflow coach

Once you know, which node you need, just click the name, and drag and drop it onto your canvas.

Figure 9: Draging a node to the canvas

If you like, you can also give your node a name (this is particularly useful for producing reports from your data). For this super basic example workflow, I selected three relatively simple nodes - one for every data processing step: a csv reader that makes comma seperated data available in KNIME, a node performing correlation analyses on these data and an Excel writer node that saves the results of this analysis into a spreadsheet.

Figure 10: A selection of unconnected nodes

Next, you'll need to connect the nodes. Therefore, you'll have to click the output port of a node (the small black triangle on the right of a node) and drag a line to the input port (small black triangle on the left) of the node that is next in line.

Figure 11: Two connected nodes

When all nodes are there and wired in the right order, it is likely that those nodes still do not 'know' what to do, yet and, thus need to be configured.

Nodes that still need configuration indicate this with a red traffic light icon underneath them. Nodes that are set up correctly and wait to be executed turn yellow. Nodes which have been executed turn green. And finally, nodes that encounter a problem in the way they have been set up indicate this with a warning sign (triangle with an exclamation mark). When you hover over the warning sign, KNIME will tell you what the problem is.

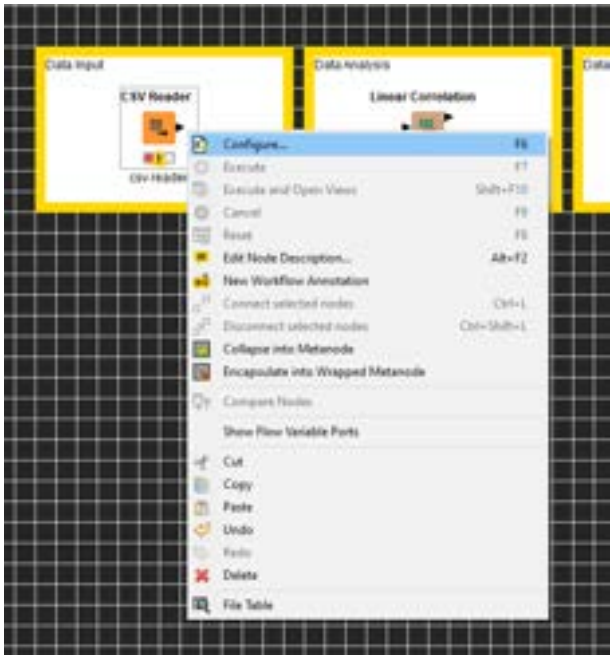

Figure 12: Configuring nodes

In the case of the csv reader, we are for instance told that it does not point to an input file, yet. So, we open the **node context menu** by right-clicking. This menu is the place where we as the analysist actually interact with a node. Here, we could e.g., delete it, copy it, view its output etc. Above all, this is, however the place, where we can access the node's settings by choosing **'configure'**.

The appearance of a node configuration menu, of course, varies from node to node. These are for example the selections KNIME lets you make on the CSV Reader:

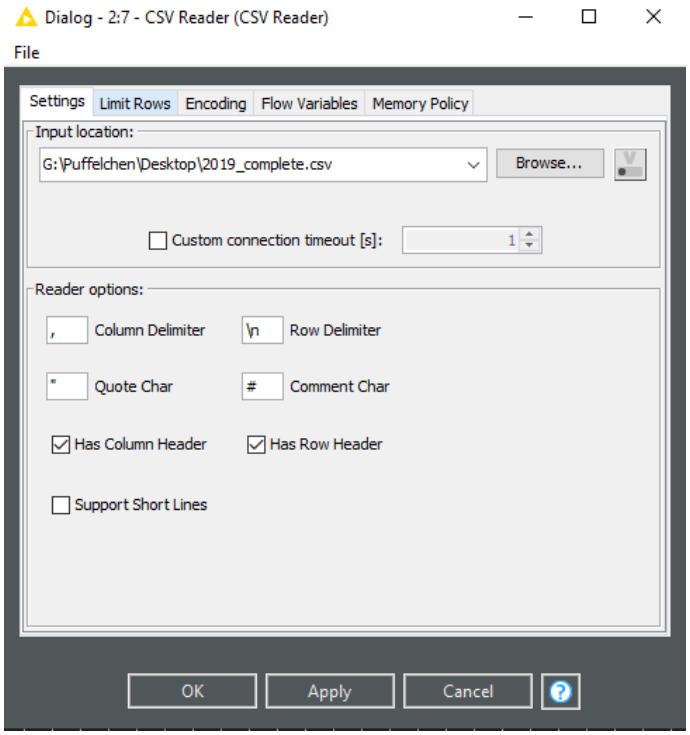

Figure 13: Exemplar node configuration menu

Once all nodes are configured, the entire analysis can be performed. We therefor simply switch to the last node in the chain, right-click it, and select 'Execute' from the context menu. This will run this node and all its predecessors.

Figure 14: Executing nodes

3 Why would I want to use KNIME (and why maybe not)?

KNIME with its Lego-like building blocks approach to data analysis is for sure pretty cool. But is it actually worth trying out or even switching to it entirely? In order to answer this question, I want to add a pros & cons section here, where I'd like to explicitly dive deeper into the functionalities the free version of KNIME (the Analytics Platform) and compare the tool to more established, coding-based items from a data scientist's toolkit, like R or Python.

And just one disclaimer beforehand: As so often in life, advantages and disadvantages are a matter of perspective. Accordingly, if the features of KNIME suit you or not depends a lot on the kind of analysis you want to perform, what you

intend to do with the output of this analysis, and also to a great deal on your personal statistics and/or programming skills. Therefore, I'll be speaking of *potential* pros and cons and you'll find several characteristics of KNIME mentioned in both lists.

3.1 (Potential) Pros

KIME is extremely visual

Probably the biggest advantage of KNIME is its highly visually appealing character: The workflows usually tell a spectator immediately what has been or will be done with the data. In this way, KNIME fuses the normally separate steps of data analysis and **documentation** into one single procedure. This makes KNIME extremely handy for business reports or academic publications because you do not have to waste entire paragraphs on telling you clients or colleagues what you did but simply show it to them. This is particularly practical for situations in which you want to share your insights with **audiences who have little or no background**, at all, in the type of analyses you performed – because even though they might not understand the technical details, they will very likely still grasp the main concepts. Moreover, this is very likely to enhance the **reproducibility** of your analyses: The more comprehensive the steps of your analysis are, the easier it is for someone else to replicate them. Similarly, it is super easy for yourself to replicate different results and **test alternative workflows** against each other. Just place multiple different nodes (or differently configured nodes) on your canvas and keep track and keep rewiring your workflow.

Moreover, the visual cues in KNIME also make **'debugging'** super easy: If your analysis comes to a halt, you will not have to try and search the entire workflow and step by step narrow down, where the error might be hidden, instead KNIME marks the spot for you and flags the node, where the error occurs with a warning sign.

Finally, since KNIME is so super visual by nature, it (of course) also features a plethora of nodes that let you create **beautiful graphics** from your data and customize them to your needs – classic ones (bar chart, scatter plot) as well as advanced charts (parallel coordinates, sunburst, network graph, heat map, interactive graphics). The options here are admittedly much prettier than e.g., with Excel or SPSS and usually much faster produced than e.g., with R or Python.

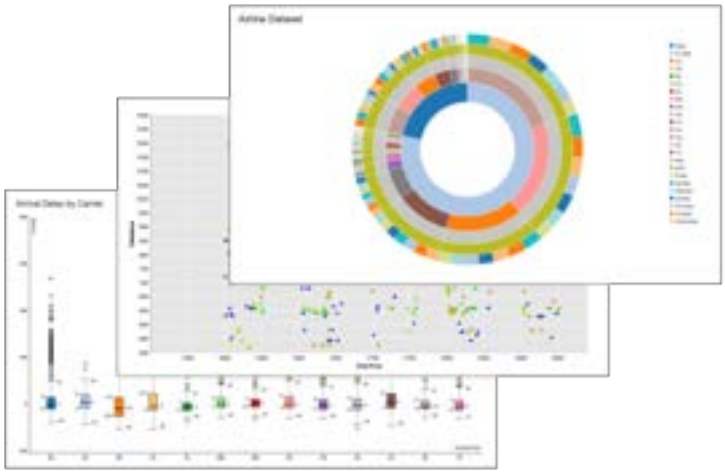

Figure 15: Exemplar plots created in KNIME

KNIME is good at Machine Learning, NLP & AI

Especially in recent years, one of the primary foci of the KNIME developers has been on topics surrounding machine learning (e.g., automated big data analyses, neural networks, Artificial Intelligence, or Natural Language Processing). Usually, those are tasks that call for a great deal of expert knowledge and much computational power (and therefore often a specific setup of your computer). The KNIME team, however, managed to build interfaces for deep learning algorithms (NNs, RNNs, LSTMs etc.) that guide you even though the creation of such elaborate models within a few minutes. And you can do pretty much everything you would want to do to evaluate those models: perform cross validation to guarantee model stability, validate your models by applying performance metrics (including accuracy, R2, AUC, ROC...), and tune your model with hyperparameter optimization, boosting, bagging, or stacking.

And even for people with a profound background in machine learning, KNIME, especially the NLP features, can come as a blessing: If you chose or were forced to work on Windows, working with texts that contain non-ASCII characters, can be quite troublesome. KNIME is usually an excellent way to escape this 'encoding hell'.

Moreover, KNIME provides you with functionalities that let you build upon several of the most powerful, well-established deep learning frameworks (such *Keras* or *TensorFlow*).

No coding is needed

In order to work with KNIME, you do not have to learn the (sometimes quite challenging and overwhelming) fundamentals of a programming language (e.g., how you initialize variables, what data objects you could use, what naming conventions there are etc.). Instead, you set up your nodes mostly via drop-down menus and even have a workflow assistant to guide you through this. So, basically, you can start right away, even on extremely elaborate topics even experienced data science

cracks usually tend to struggle with (KNIME for instance offers a "Friendly Introduction to Codeless Deep Learning" as a regularly reoccurring free webinar). And KNIME takes the no-hard-skills-needed idea even to a meta-level: It lets you create data analysis applications from your workflows, where your customers (or other third parties who need not have a background in data analysis, at all) are provided with an interactive, so called "guided analysis". This is basically a web portal, where you let them play around with your data and also provide them with the basic information they need therefore.

Figure 16: Guided Analytics in KNIME

Coding skills are, yet, not wasted

If you, however, happen to know how to code, you can do even more with KNIME: It offers several nodes that let you integrate code snippets into your workflow. KNIME for instance features interfaces to Python, R, and Java. If you know how to code in the latter, you could even create your very own KNIME nodes: KNIME is written in Java and is open-source (the entire code is publicly available on GtHub under the terms of GNU General Public License, Version 3). The KNIME creators even actively encourage other developers to alter the base code and promote quite an easy workaround to do so (explaining how to extend the three node base classes 'NodeModel', 'NodeDialog', and/or 'NodeView').

Many established tools can be blended with KNIME

KNIME offers you indeed very, very many options to almost seamlessly blend tools from different domains with its native nodes in your workflows. We already mentioned the scripting nodes for R, Python, and Java, but there is much more: Among others, KNIME features

- excellent extensions for **performance enhancement** (*KNIME Big Data Connectors* for executing *Hive* queries on *Hadoop*, the *KNIME Extension for Apache Spark* for training models on Hadoop, or the *KNIME Cloud Analytics Platform* for *Azure*),

- a variety of **database parsing** options (it enables you to integrate data from *Oracle*, *Microsoft SQL*, *Apache Hive*, and lets you load *JSON*, *Avro*, *Parquet*, *ORC* files from *HDFS*, *S3*, or *Azure* etc.), and
- many connectors to specific **data sources** (such as *Salesforce*, *SharePoint*, *SAP Reader*, *Twitter*, *AWS S3*, *Google Sheets*, *Azure*, and more).

There is a lot of help out there

One final, very big selling point of KNIME is that you can almost jump-start your analysis because of the tons of assisting resources that are being offered to you:

- Inside of KNIME itself, you have an integrated **workflow coach** that helps you make the right node selections and assists you on their configuration.
- There are a lot of written as well as video **tutorials** freely available (e.g., the starter course 'A KNIME Introduction in 90 Minutes' that really tells you pretty much everything you'll need to know at first.
- On KNIME's website you can access a whole pool of additional, more **in-depth documents** via the so called **KNIME Academic Alliance** – KNIME's initiative for 'supporting a community of educators and researchers to empower a new generation of data wranglers'. Here you can browse through academic whitepapers covering specific analyses with the tool, entire free books, and even teaching materials, that would guide you if you decided to give courses on KNIME yourself.
- Generally, KNIME embraces the **community** idea a lot – but not only virtually on their website, where you can create an account to engage in the forum or become part of the before mentioned Academic Alliance. KNIME also hosts **in-person events and workshops** in different countries for which you can sign up.

Figure 17: An exemplar webinar

- But probably the most widely used helping KNIME resource is the so-called **KNIME Hub**. This is a database that holds hundreds of KNIME extensions and components available for you, and even thousands of nodes and entire workflows. You just have to type in a feature you are looking for or an entire type of analysis you would like to perform and can browse through the results KNIME recommends. The single posts each contain a detailed description of the workflow/node/extension/component as well as a direct download option. This is super sleek because if you find that somebody already built a workflow for the same task you are up to, you can directly import it into your local KNIME installation and probably have your analysis done within a matter of minutes.

Figure 18: Searching the KNIME Hub

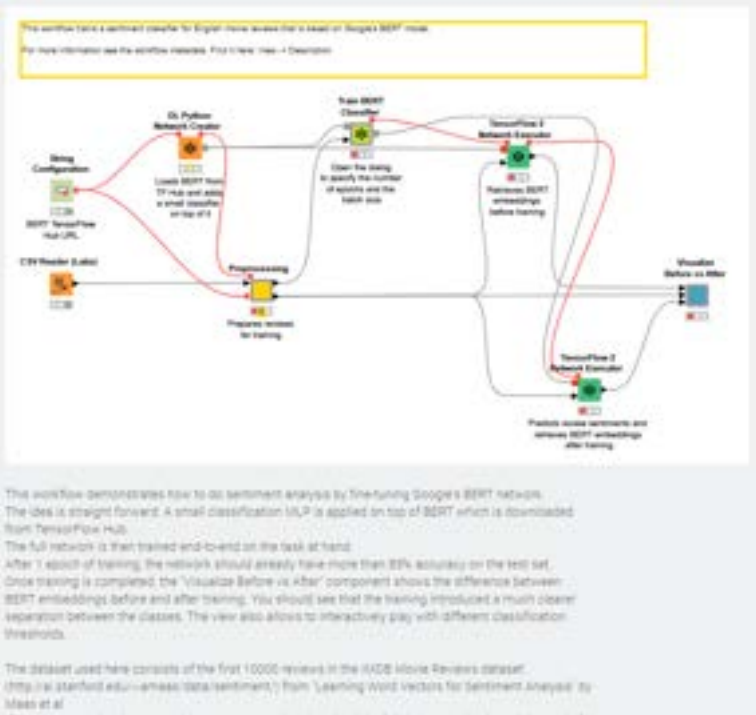

Figure 19: An exemplar workflow description on the KNIME Hub

3.2 (Potential) Cons

KNIME is not an organization

The terms 'open-source' and 'free' tend to get mixed up quite often when speaking of software. So, it might strike some people that KNIME, though it is open-source, is not as 'free' as they might be used to from other open-source solutions in the realm of data science: The fact that it is hosted at a .com address instead of a .org (like e.g., R or Python), already tells us that KNIME is the product of a company with a view to profit and not of a non-profit organization or foundation. At the moment, KNIME seems to make profit primarily from selling its server version, developing tailor-made solutions for business customers, and training (when you browse the events catalogue, you'll find some of them to be quite pricy). Yet, also the free Analytics Platform comes at a small cost: You will be prompted to create an account and share your personal data (name, e-mail address and company), when clicking the download link. You can, however, skip this by clicking on "Download KNIME" in the header timeline of this page. Moreover, the current terms of service are none out of the ordinary – as to now, your data seem to be quite safe with the company. Still, this login comes with a taste and it is not guaranteed that the terms of service won't change, and that all features will stay for free.

No coding

KNIME is being marketed with the catchy claim "You don't need an advanced degree in coding any more to run sophisticated analyses" on its website. This statement sounds tempting, has, however, several flaws: First, also for applying R, Python, Julia and the like to your data, you do not need to have a degree. Programming is no rocket science (well, admittedly, some rocket science involves programming... but), it might nowadays indeed even be the most prototypic thing to learn as an autodidact because of the millions of free resources out there. Second, not needing to have profound knowledge in an area does not necessarily have to be something desirable. KNIME has the tendency to oversimplify things so much that the tool is prone to misuse. Remember: not every result that a computer spits out does have to be a meaningful result. At the latest when you, want to discuss your findings with experts in this field (e.g., in a paper for a renowned scientific journal) you might find that a "Friendly Introduction to Codeless" whatever did not prepare you enough. Third, every graphical drag'n'drop or point'n'click interface comes at the cost of flexibility because you are limited to the preselection of options the developers incorporated for you. Since there is a sheer plethora of such options in KNIME, this does not seem to be an issue at first sight. However, no preselection – no matter how large – is equivalent to true independence from the developer. This is especially crucial if you are a researcher since research aims at progress within the scientific community and thus, you'll very likely sooner or later come to the point where you want to try things out no one has done before and might feel rather trapped within the limits of a tool like KNIME. Forth, the #1 platform for exchange between professional data scientists is without a doubt Stack Overflow. No matter what your question is or what you want to do with your data, chances

are good that a solution (including code that you could just copy from there) has already been posted to this forum. Since KNIME, however, does not involve actual coding, it never really took off as a topic here: While only 263 discussion in sum are tagged with this tool, for Python it were 1357 new ones only for the day I checked.

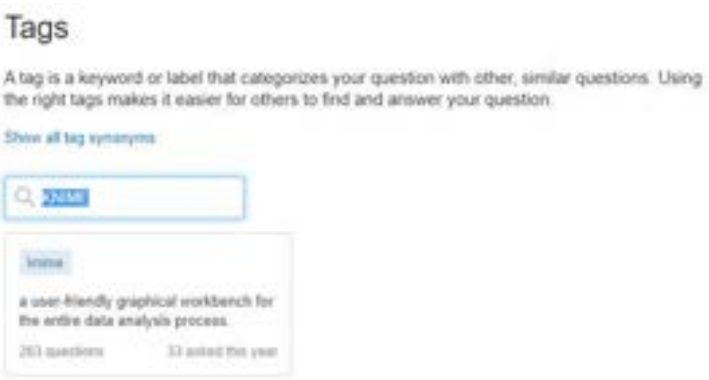

Figure 20: Search results for the tag 'Knime' on Stackoverflow

Figure 21: Search results for the tag 'Python' on Stackoverflow

Things are not as easy as they seem

Even though you do not have to learn a programming language in order to work with KNIME and everything looks really intuitive from the far, the learning curve for KNIME is not necessarily as narrow as you'd expect it to be: At first, it can be quite intimidating to face a catalogue of several thousands of nodes and having to find and combine the right ones for your purposes. What adds to this problem is that it can be quite hard to figure out what KNIME actually can do with its own genuine nodes and where you will have to use an interface to another tool or will not be able to carry out the intended task, at all (see e.g., next section). Also, for some tasks, configuring nodes might be even more complicated than writing code: Particularly data preprocessing (cleaning, aggregating etc.) can be quite a hassle with KNIME and can often be accomplished by a single short line of code in R or Python. Accordingly, if you are already relatively familiar with a programming language but confronted with a task you never had to perform before (e.g., training

convolutional neural networks or something alike), you might be better off, investing the time to research a code-based solution.

Some important features are (still) missing

Though KNIME can do much, it cannot do everything a data scientist might want to do. Especially in the last years, the development focus of the KNIME core team (but also of the community) seems to have been on powerful deep learning modules: they created nodes for RNNs, LSTMs, reinforcement learning etc. This or course is nice, if you want to perform deep learning. If you are, however, looking for something more 'classical' to do with your data, KNIME might probably disappoint you. Especially important analyses, we social scientists like to perform for experimental studies or surveys are missing: As of now, there are for instance no nodes for **structural equation modelling** or **factor analysis** (besides PCA). Also, though KNIME offers interfaces to a lot of other tools/languages, the list is still incomplete. You might for instance miss possibilities to integrate **C** or **Julia** snippets or use scripts created in **SPSS** (.sav files can however be read). Also, the reporting options might feel deficient to a data scientist: Though you can use KNIME workflows in Jupyter notebooks and can create PDFs or entire webpages from so called BIRT reports, there is no straight forward way to team KNIME up with **LaTeX** (like e.g., with *kniR* or *sweave* for R). Finally, many data scientists who are used to pushing their scripts etc. to *GitHub* or *Bitbucket* might be struck by the fact that the free KNIME Analytics Platform does not support **versioning**. It is only available with KNIME Server that currently comes at a steep price of at least **14,500 USD** per year.

3.3 Conclusion

So, having all that said and balancing off all the pros and cons, when and to whom would I actually recommend using KNIME?

Admittedly, if you already are a decent data scientist with lots of experience in actual programming languages, chances are good, you are better off sticking to your usual toolkit for most scenarios:

For truly innovative analysis you'll much likely still be faster using R, Python or Julia because researching some new code snippets will still take less time than navigating through all the different nodes KNIME has to offer and, yet you are much more flexible. Moreover, this flexibility does not only apply to the analyses themselves but also to your options for reporting them and for collaboration. This makes KNIME also less convenient for scientific publications aiming at an audience of fellow data scientists since there you would want to make your data easily accessible and reproducible with established state-of-the-art tools. However, KNIME definitely has its raison d'être and comes in handy in a bunch of other situations:

The greatest advantage of KNIME is probably that it lets you do many not so simple tasks in an easy manner and, thus, sets you up quickly on a topic you might not be too familiar with, yet. This is ideal if you want to produce neat output for smaller projects you do not want to or cannot invest tons of time into (such as short

presentations or business reports). Additionally, KNIME might come in practical even if you are a profound expert on a topic but your audience might not be. It lets you engage easily with people with no to little background in data science because the visual workflows are much more comprehensive than some lines of code. This might KNIME probably still applicable for some scientific publications, namely for those addressed to audiences from the social sciences of humanities who have relatively little code literacy. Finally, this might also make KNIME an excellent tool for teaching: You can focus on the basic principles of a data science topic (such as neuronal nets, LDA or the like) with your students without having to fear to lose them over technical details such as coding obstacles.