**Unit-1**

**Data Scientist’s Tool Box: Turning data into actionable knowledge, introduction to the tools that will be used in building data analysis software: version control, markdown, git, GitHub, R, and RStudio.**

# What is Data Science: Lifecycle, Applications, Prerequisites and Tools



**What Is Data Science?**

Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. Data science uses complex machine learning algorithms to build predictive models.

The data used for analysis can come from many different sources and presented in various formats.

**The Data Science Lifecycle**

Data science’s lifecycle consists of five distinct stages, each with its own tasks:

1. Capture: Data Acquisition, Data Entry, Signal Reception, Data Extraction. This stage involves gathering raw structured and unstructured data.
2. Maintain: Data Warehousing, Data Cleansing, Data Staging, Data Processing, Data Architecture. This stage covers taking the raw data and putting it in a form that can be used.
3. Process: Data Mining, Clustering/Classification, Data Modeling, Data Summarization. Data scientists take the prepared data and examine its patterns, ranges, and biases to determine how useful it will be in predictive analysis.
4. Analyze: Exploratory/Confirmatory, Predictive Analysis, Regression, Text Mining, Qualitative Analysis. Here is the real meat of the lifecycle. This stage involves performing the various analyses on the data.
5. Communicate: Data Reporting, Data Visualization, Business Intelligence, Decision Making. In this final step, analysts prepare the analyses in easily readable forms such as charts, graphs, and reports.

## Prerequisites for Data Science

Here are some of the technical concepts you should know about before starting to learn what is data science.

### 1. Machine Learning

Machine learning is the backbone of data science. Data Scientists need to have a solid grasp of ML in addition to basic knowledge of statistics.

### 2. Modeling

Mathematical models enable you to make quick calculations and predictions based on what you already know about the data. Modeling is also a part of[Machine Learning](https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-machine-learning) and involves identifying which algorithm is the most suitable to solve a given problem and how to train these models.

### 3. Statistics

Statistics are at the core of data science. A sturdy handle on statistics can help you extract more intelligence and obtain more meaningful results.

### 4. Programming

Some level of programming is required to execute a successful[data science project](https://www.simplilearn.com/data-science-projects-article). The most common programming languages are Python, and R. Python is especially popular because it’s easy to learn, and it supports multiple[libraries for data science and ML](https://www.simplilearn.com/top-python-libraries-for-data-science-article).

### 5. Databases

A capable data scientist needs to understand how databases work, how to manage them, and how to extract data from them.

## What Does a Data Scientist Do?

A [data scientist](https://www.simplilearn.com/tutorials/data-science-tutorial/how-to-become-a-data-scientist) analyzes business data to extract meaningful insights. In other words, a data scientist solves business problems through a series of steps, including:

* Before tackling the data collection and analysis, the data scientist determines the problem by asking the right questions and gaining understanding.
* The data scientist then determines the correct set of variables and data sets.
* The data scientist gathers structured and unstructured data from many disparate sources—enterprise data, public data, etc.
* Once the data is collected, the data scientist processes the raw data and converts it into a format suitable for analysis. This involves cleaning and validating the data to guarantee uniformity, completeness, and accuracy.
* After the data has been rendered into a usable form, it’s fed into the analytic system—ML algorithm or a statistical model. This is where the data scientists analyze and identify patterns and trends.
* When the data has been completely rendered, the data scientist interprets the data to find opportunities and solutions.
* The data scientists finish the task by preparing the results and insights to share with the appropriate stakeholders and communicating the results.

**Why Become a Data Scientist?**

According to Glassdoor and Forbes, demand for data scientists will increase by 28 percent by 2026, which speaks of the profession’s durability and longevity, so if you want a [secure career](https://www.simplilearn.com/tutorials/data-science-tutorial/data-science-career), data science offers you that chance.

Furthermore, the profession of data scientist came in[second place in the Best Jobs in America for 2021](https://www.glassdoor.com/List/Best-Jobs-in-America-LST_KQ0,20.htm) survey, with an average base salary of USD 127,500.

So, if you’re looking for an exciting career that offers stability and [generous compensation](https://www.simplilearn.com/tutorials/data-science-tutorial/data-scientist-salary), then look no further!

## Where Do You Fit in Data Science?

Data science offers you the opportunity to focus on and specialize in one aspect of the field. Here’s a sample of different ways you can fit into this exciting, fast-growing field.

### Data Scientist

* Job role: Determine what the problem is, what questions need answers, and where to find the data. Also, they mine, clean, and present the relevant data.
* Skills needed: Programming skills (SAS, R, Python), storytelling and data visualization, statistical and mathematical skills, knowledge of Hadoop, SQL, and Machine Learning.

### Data Analyst

* Job role: Analysts bridge the gap between the data scientists and the business analysts, organizing and analyzing data to answer the questions the organization poses. They take the technical analyses and turn them into qualitative action items.
* Skills needed: Statistical and mathematical skills, programming skills (SAS, R, Python), plus experience in data wrangling and data visualization.

### Data Engineer

* Job role: [Data engineers](https://www.simplilearn.com/tutorials/data-science-tutorial/data-scientist-vs-data-analyst-vs-data-engineer) focus on developing, deploying, managing, and optimizing the organization’s data infrastructure and data pipelines. Engineers support data scientists by helping to transfer and transform data for queries.
* Skills needed: NoSQL databases (e.g., MongoDB, Cassandra DB), programming languages such as Java and Scala, and frameworks (Apache Hadoop).

## Data Science Tools

The data science profession is challenging, but fortunately, there are plenty of tools available to help the data scientist succeed at their job.

* Data Analysis: SAS, Jupyter, R Studio, MATLAB, Excel, RapidMiner
* Data Warehousing: Informatica/ Talend, AWS Redshift
* Data Visualization: Jupyter, Tableau, Cognos, RAW
* Machine Learning: Spark MLib, Mahout, Azure ML studio

## What is Markdown?

**Markdown** is a human readable syntax (also referred to as a markup language) for formatting text documents. Markdown can be used to produce nicely formatted documents.

Markdown syntax allows you to format text in many ways, such as making headings, bolding and italicizing words, creating bulleted lists, adding links, formatting mathematical symbols and making tables. These options allow you to format text in visually appealing and organized ways to present your ideas.

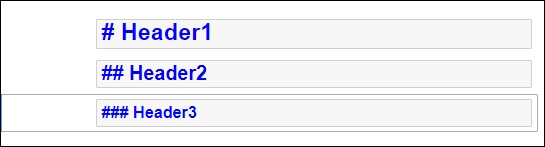
You can use Markdown to format text in many different tools including [**GitHub.com**](https://guides.github.com/features/mastering-markdown/), R using **[RMarkdown](https://rmarkdown.rstudio.com/lesson-1.html" \t "_blank)**, and **Jupyter** Notebook, which you will learn more about this page.

Markdown cell displays text which can be formatted using markdown language. In order to enter a text which should not be treated as code by Notebook server, it must be first converted as markdown cell either from cell menu or by using keyboard shortcut M while in command mode. The In[] prompt before cell disappears.

Header cell

A markdown cell can display header text of 6 sizes, similar to HTML headers. Start the text in markdown cell by # symbol. Use as many # symbols corresponding to level of header you want. It means single # will render biggest header line, and six # symbols renders header of smallest font size. The rendering will take place when you run the cell either from cell menu or run button of toolbar.

Following screenshot shows markdown cells in edit mode with headers of three different levels.



When cells are run, the output is as follows −



Note that Jupyter notebook markdown doesn’t support WYSWYG feature. The effect of formatting will be rendered only after the markdown cell is run.

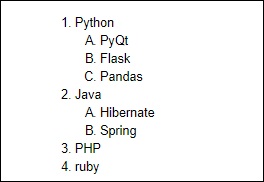
Ordered Lists

To render a numbered list as is done by <ol> tag of HTML, the First item in the list should be numbered as 1. Subsequent items may be given any number. It will be rendered serially when the markdown cell is run. To show an indented list, press tab key and start first item in each sublist with 1.

If you give the following data for markdown −

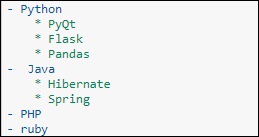


It will display the following list −

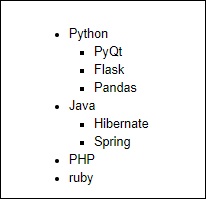


Bullet lists

Each item in the list will display a solid circle if it starts with – symbol where as solid square symbol will be displayed if list starts with \* symbol. The following example explains this feature −



The rendered markdown shows up as below −



Hyperlinks

Markdown text starting with http or https automatically renders hyperlink. To attach link to text, place text in square brackets [] and link in parentheses () optionally including hovering text. Following screenshot will explain this.

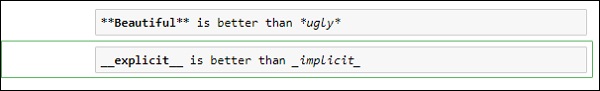


The rendered markdown appears as shown below −

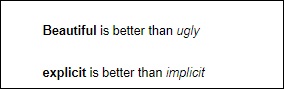


Bold and Italics

To show a text in bold face, put it in between double underscores or two asterisks. To show in italics, put it between single underscores or single asterisks.



The result is as shown below −



Images

To display image in a markdown cell, choose ‘Insert image’ option from Edit menu and browse to desired image file. The markdown cell shows its syntax as follows −

Logo Attachment

Image will be rendered on the notebook as shown below −



Table

In a markdown cell, a table can be constructed using | (pipe symbol) and – (dash) to mark columns and rows. Note that the symbols need not be exactly aligned while typing. It should only take respective place of column borders and row border. Notebook will automatically resize according to content. A table is constructed as shown below −



The output table will be rendered as shown below −



### Benefits of Markdown for Earth Data Science

Being able to include both Markdown and code (e.g. Python) cells in a Jupyter Notebook file supports reproducible science by allowing you to:

* Document your workflow: You can add text to the document that describes the steps of your processing workflow (e.g. how data is being processed and what results are produced).
* Describe your data: You can describe the data that you are using (e.g. source, pre-processing, metadata).
* Interpret code outputs: You can add some text that interprets or discusses the outputs.

all in one document!

### Markdown Cells in Jupyter Notebook

| **Function** | **Keyboard Shortcut** | **Menu Tools** |
| --- | --- | --- |
| Create new cell | Esc + a (above), Esc + b (below) | Insert→ Insert Cell Above OR Insert → Insert Cell Below |
| Copy Cell | c | Copy Key |
| Paste Cell | v | Paste Key |

## Highlight Code

If you want to highlight a function or some code within a plain text paragraph, you can use one backtick on each side of the text like this:

`Here is some code!`

which renders like this:

Here is some code!

The symbol used is the backquote, or grave; not an apostrophe (on most US keyboards, it is on the same key as the tilde (~)).

## Horizontal Lines (Rules)

You can also create a horizontal line or rule to highlight a block of Markdown syntax (similar to the highlighting a block of code using the backticks):

\*\*\*

Here is some important text!

\*\*\*

which renders like this:

Here is some important text!

## Practice Your Markdown Skills

1. Open or create a new Jupyter Notebook file.
2. Add a new Markdown cell and include:
   * A title for the notebook (e.g. Intro to Earth Analytics - Chapter Four)
   * A **bullet list** with:
     + A bold word for Author: and then add text for your name.
     + A bold word for Date: and then add text for today’s date.
3. Add another Markdown cell and include:
   * A list of your top three favorite foods (e.g. blueberries, chocolate bars, avocados).
     + Italicize the first item in your list.
     + Add a hyperlink (i.e. webpages) for the second item in your list (include the name of the food in the title of the hyperlink).
     + Add an image for the last item in your list (include the name in the alt text of the image).

# 1. Get familiar with version control, Git and GitHub

## What is version control?

Version control allows you to keep track of your work and helps you to easily explore the changes you have made, be it data, coding scripts, notes, etc. You are probably already doing some type of version control, if you save multiple files, such as Dissertation\_script\_25thFeb.R, Dissertation\_script\_26thFeb.R, etc. This approach will leave you with tens or hundreds of similar files, making it rather cumbersome to directly compare different versions, and is not easy to share among collaborators. **With version control software such as**[**Git**](https://git-scm.com/)**, version control is much smoother and easier to implement. Using an online platform like [Github](https://github.com/) to store your** files means that you have an online back up of your work, which is beneficial for both you and your collaborators.

## What are the benefits of using version control?

Having a GitHub repo makes it easy for you to keep track of collaborative and personal projects - all files necessary for certain analyses can be held together and people can add in their code, graphs, etc. as the projects develop. Each file on GitHub has a history, making it easy to explore the changes that occurred to it at different time points. You can review other people’s code, add comments to certain lines or the overall document, and suggest changes. For collaborative projects, GitHub allows you to assign tasks to different users, making it clear who is responsible for which part of the analysis. You can also ask certain users to review your code. For personal projects, version control allows you to keep track of your work and easily navigate among the many versions of the files you create, whilst also maintaining an online backup.

## How to get started

**Please register on the [Github website](https://github.com/).**

On your computer, you need to install Git first. The process will depend on your operating system: please follow the instructions below by clicking the relevant button.

Linux

Windows

macOS

The files you put on GitHub will be public (i.e. everyone can see them & suggest changes, but only the people with access to the repository can directly edit and add/remove files). You can also have private repositories on GitHub, which means that only you can see the files. GitHub now offers [free private repositories as standard](https://blog.github.com/2019-01-07-new-year-new-github/) with up to three collaborators per repository. They also offer a free education package, with access to software and other perks, you can apply for one using [this link](https://education.github.com/discount_requests/new).

## How does version control work?

### What is a repository?

You can think of a repository (aka a repo) as a “main folder”, everything associated with a specific project should be kept in a repo for that project. Repos can have folders within them, or just be separate files.

You will have a local copy (on your computer) and an online copy (on GitHub) of all the files in the repository.

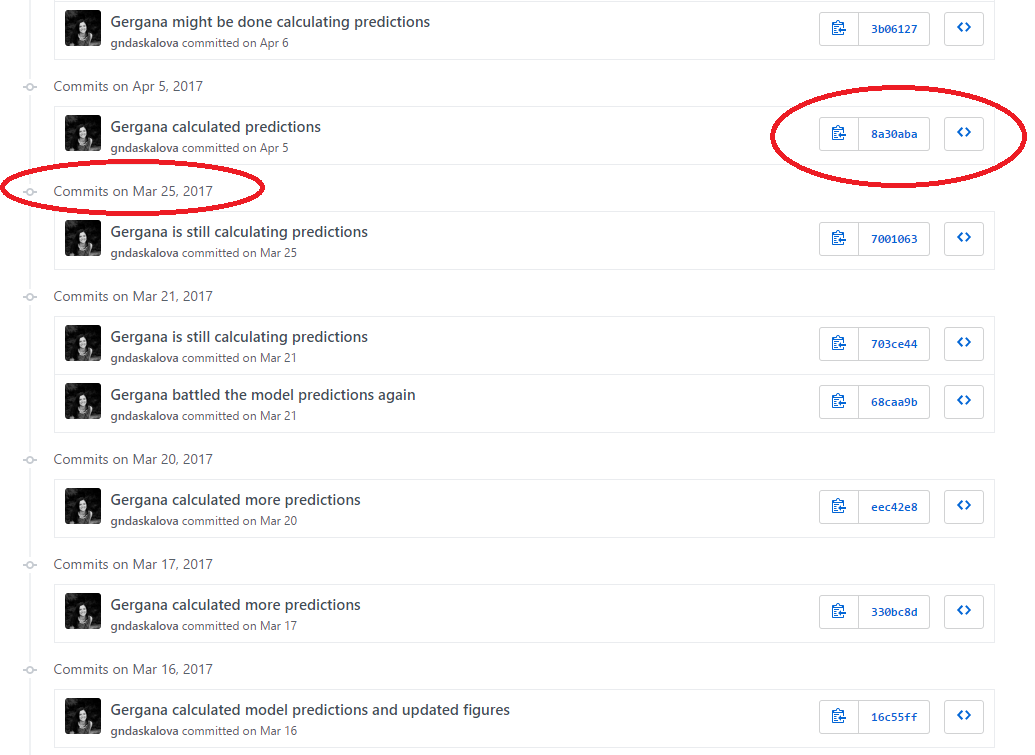
### The workflow

The GitHub workflow can be summarised by the “commit-pull-push” mantra.

1. Commit
   * Once you’ve saved your files, you need to commit them - this means the changes you have made to files in your repo will be saved as a version of the repo, and your changes are now ready to go up on GitHub (the online copy of the repository).
2. Pull
   * Now, before you send your changes to Github, you need to pull, i.e. make sure you are completely up to date with the latest version of the online version of the files - other people could have been working on them even if you haven’t. You should always pull before you start editing and before you push.
3. Push
   * Once you are up to date, you can push your changes - at this point in time your local copy and the online copy of the files will be the same.

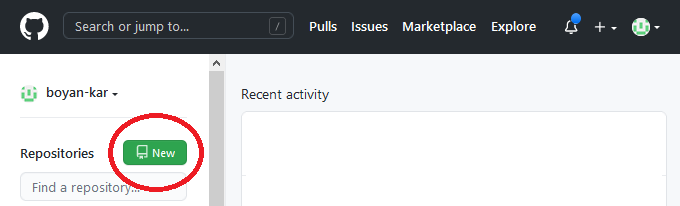
Each file on GitHub has a history, so instead of having many files like Dissertation\_1st\_May.R, Dissertation\_2nd\_May.R, you can have only one and by exploring its history, you can see what it looked at different points in time.

For example, here is the history for a repo with an R script inside it, as viewed on Github. Obviously it took me a while to calculate those model predictions!

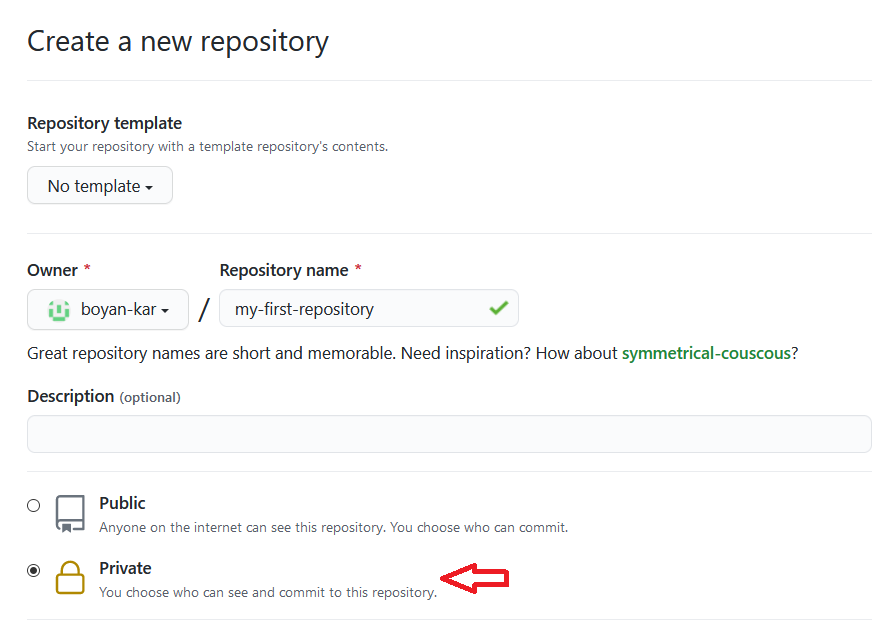


# 2. Create your own repository and project folder structure

To make a repository, go to Repositories/New repository - choose a concise and informative name that has no spaces or funky characters in it. This can be your master repo that holds together past and ongoing research, data, scripts, manuscripts. Later on you might want to have more repositories - e.g. a repository associated with a particular project that you want to make public or a project where you are actively seeking feedback from a wider audience. For now, we will focus on organising and using your main repository that holds the files for all your work. With a free GitHub account, you can use public or private respositories.



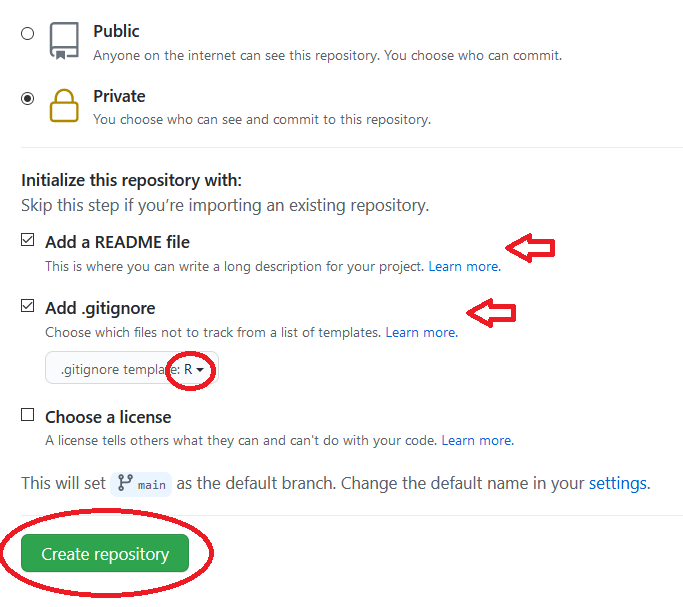
Let’s create a new private repository. You can call it whatever you like if the name is available.



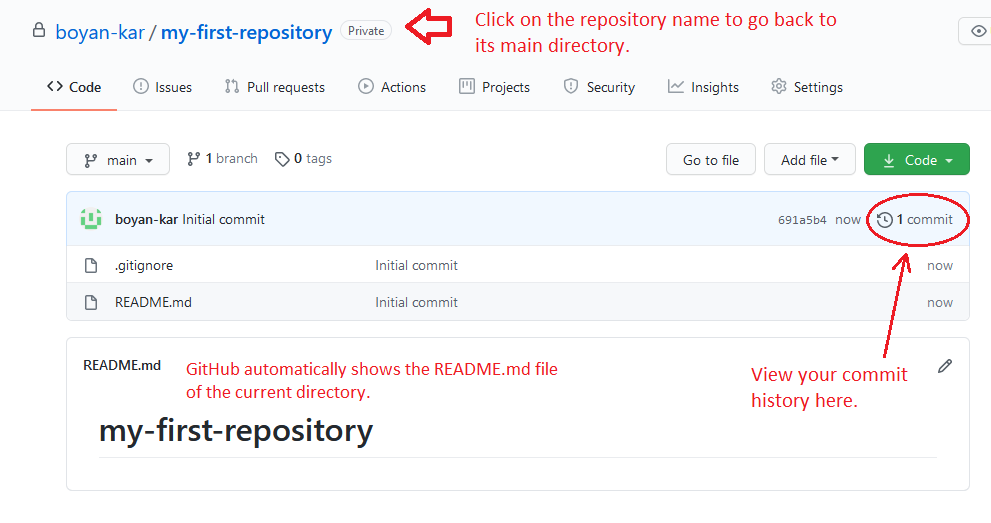
**Click on Initialise repo with a README.md file**. It’s common practice for each repository to have a README.md file, which contains information about the project, the purpose of the repository, as well as any comments on licensing and data sources. Github understands several text formats, including .txt and .md. .md stands for a file written in [Markdown](https://en.wikipedia.org/wiki/Markdown). You might have used Markdown before from within RStudio to create neatly organised reports of your code and its outputs (you can also check out our [Markdown tutorial](https://ourcodingclub.github.io/tutorials/rmarkdown/index.html). You can also use Markdown to write plain text files, for example the file you are reading now was written in Markdown.

**We will also create a .gitignore file**. This file lets Git know what kind of files should not be included in the repository. We’ll have a look at that file in a bit. Tick the box, then search for **R** in the drop-down template (or whatever programming language you will be using for the project).

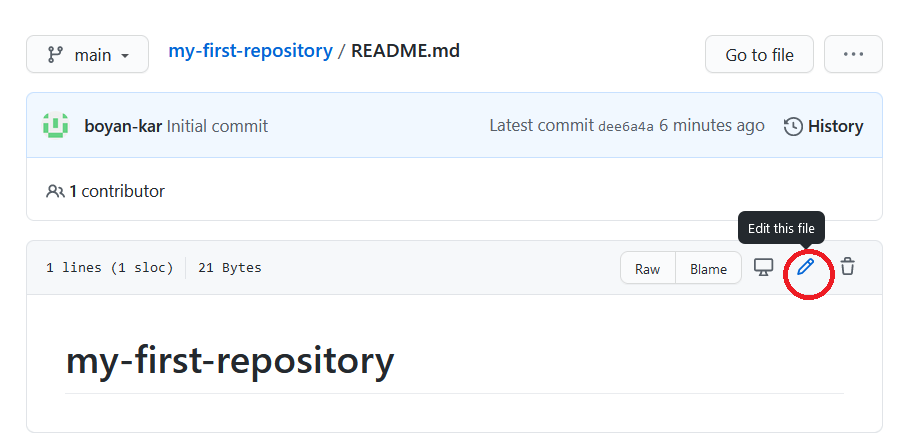
Once you are ready, click on **Create repository**.



Here is how the repository should look:



You can directly edit your README.md file on Github by clicking on the file and then selecting Edit this file.



## Exercise 1: Write an informative README.md file

You can now write the README.md file for your repository. To make headings and subheadings, put hashtags before a line of text - the more hashtags, the smaller the heading will appear. You can make lists using - and numbers 1, 2, 3, etc.. **When working on a shared project, discuss what you may want to include with your collaborators - here are some things you might want to consider:**

Copy contents

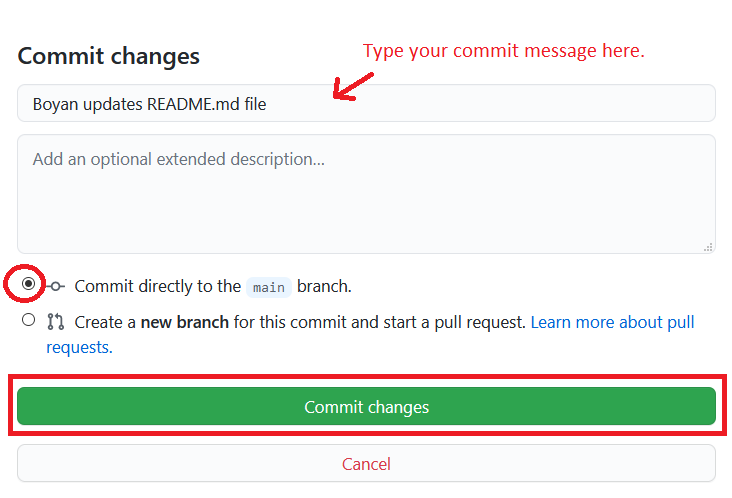
- Your name

- Project title

- Links to website & social media

- Contact details

Once you have written your README.md file, scroll to the bottom of the page. You can now **commit** the file to the repository. To do so, specify a **commit message** which briefly describes the changes. **Commit messages should concise, but descriptive.** Select **Commit directly to the main branch** and then click on **Commit changes**.



## Exercise 2: Edit the .gitignore file

Repositories often have a file called .gitignore and we are about to make one shortly. In this file you specify which files you want Git to ignore when users make changes and add files. Examples include temporary Word, Excel and Powerpoint files, .Rproj files, .Rhist files, etc. Some files you might want to only have on your local repository (i.e. on your computer), but not online as they might be too big to store online.

Select the .gitignore and click ‘Edit’. As you will see, the template GitHub provides for R already includes many file types usually found in R projects that should not be included in shared repositories. You can add more files by specifying each file type on a separate line. **Scroll to the bottom of the document and paste the following additions, without overwriting the rest**. Comments in the file are designated by a # sign. Then, commit the file to the main branch.

Copy contents

# Prevent users to commit their own .RProj

\*.Rproj

# Temporary files

\*~

~$\*.doc\*

~$\*.xls\*

\*.xlk

~$\*.ppt\*

# Prevent mac users to commit .DS\_Store files

\*.DS\_Store

# Prevent users to commit the README files created by RStudio

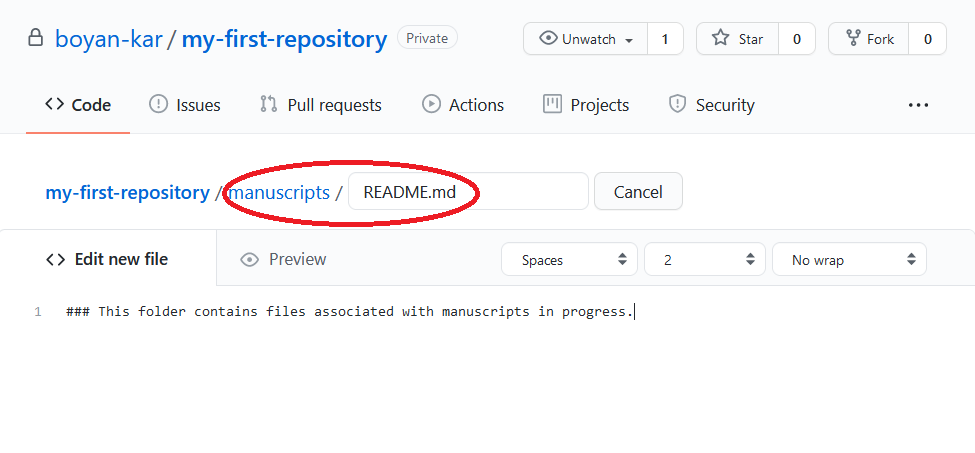
\*README.html

\*README\_cache/

#\*README\_files/

## Exercise 3: Create folders

Think of different folders you may want to include in your repository. If working on a shared repo, discuss with your collaborators. For the repository for a lab group, some examples include: manuscripts, data, figures, scripts, scripts/users/personal\_folder\_your\_name. To make a new folder, click on Create new file and add in the name of your new folder, e.g. manuscripts/ before the file name, in this case a quick README.md file. When creating folders within your repo through GitHub’s website, you always need to make at least one file associated with them, you can’t just create an empty folder. You can then write and commit the file.



## GitHub etiquette

If you’ll be sharing the repository with collaborators and even for your own benefit, it’s a good idea to define some rules on how to use the repository before we start working within it - for example what GitHub and coding etiquette should people be following? Is there a prefered folder structure, file naming system?

We can make a new github-etiquette.md file that outlines the rules that people with access to your repository should follow.

## Exercise 4: Write a github-etiquette.md file

Go to your lab’s main repository, click on Create new file and add github-etiquette.md as a file name. Remember to include the file extension .md - otherwise GitHub won’t know what’s the file format.

## A few GitHub rules:

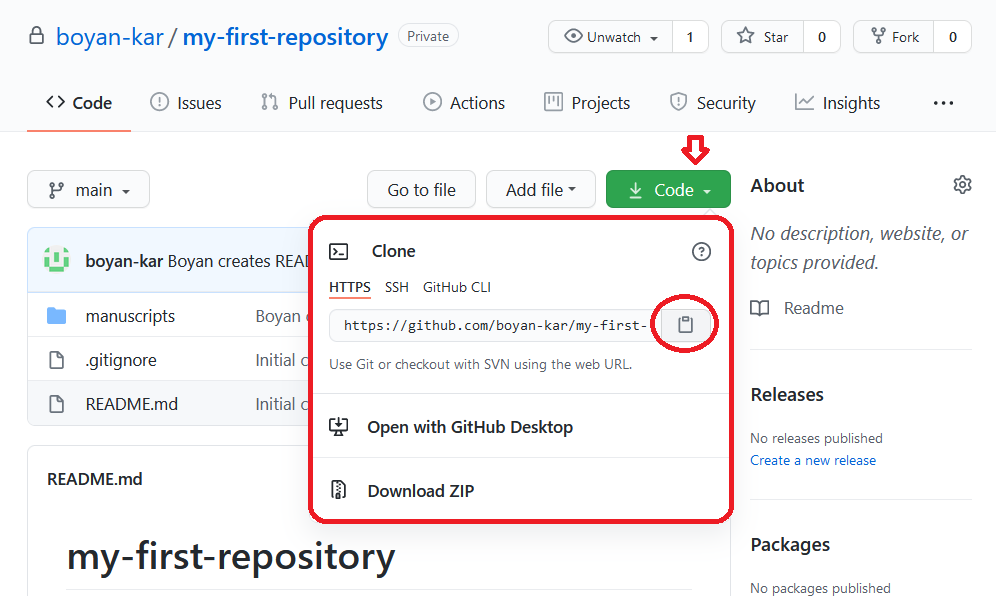
* Keep file paths short and sensible.
* Don’t use funky characters and spaces in your file names, these cause trouble because of differences in Mac/Windows systems.
* Always **pull** before you start working on your project and **before you push** in case someone has done any work since the last time you pulled - you wouldn’t want anyone’s work to get lost or to have to resolve many coding conflicts.

# 3. Sync and interact with your repository through RStudio

The “commit-pull-push” workflow can be embedded within RStudio using “Projects” and enabling version control for them - we will be doing that shortly in the tutorial.

**Log into your Github account and navigate to the repository you created earlier**

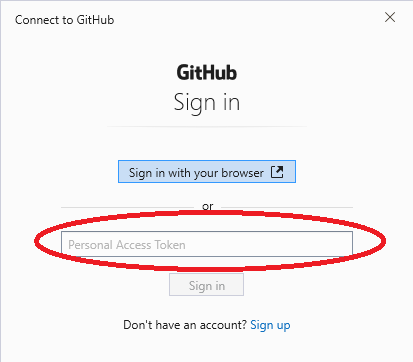
Click Code and copy the HTTPS link.



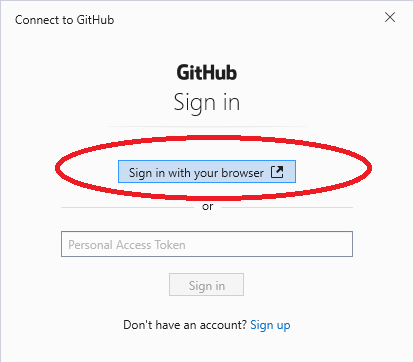
Now open RStudio, click File/ New Project/ Version control/ Git and paste the HTTPS link from the Github repository into the Repository URL: field. Select a folder on your computer - that is where the “local” copy of your repository will be (the online one being on Github).

Once you attempt the project, you will be asked to authenticate. **You only need to do this once on your computer.** There are multiple possible ways to authenticate your GitHub account on your computer and make it work with RStudio. Here are two recommended approaches:

Sign in via Personal Access Token (All platforms, recommended)



Sign in via internet browser (Windows, fast)

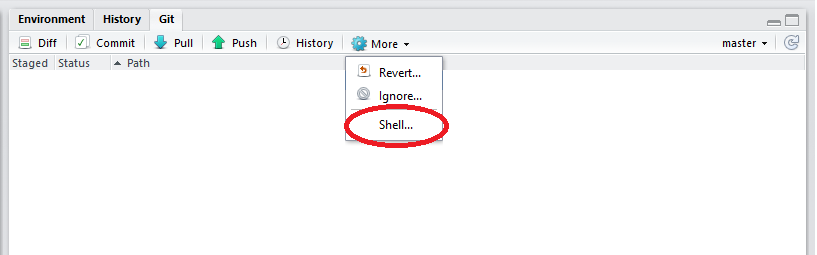


Troubleshooting (GitHub in RStudio)

**Before your first commit from your computer, you will need to configure your username and email. This is easily done, and you only need to do it once, afterwards you can commit-pull-push at your convenience!**

In the top right corner of the RStudio screen (in the Git tab), click on More/Shell.

**NOTE: If using a Windows PC, the Shell option should launch Git Bash. If it doesn’t open Git Bash, please find Git Bash on your computer instead. You can usually search for it in the Start menu, or right-click on the empty space in any folder in the File Explorer and click “Git Bash Here”.**



### Copy the following code:

Copy contents

git config --global user.email your\_email@example.com

# Add the email with which you registered on GitHub and click Enter

git config --global user.name "Your GitHub Username"

# Add your GitHub username and click Enter

If it worked fine, there will be no messages, you can close the shell window and do your commit again, this time it will work!

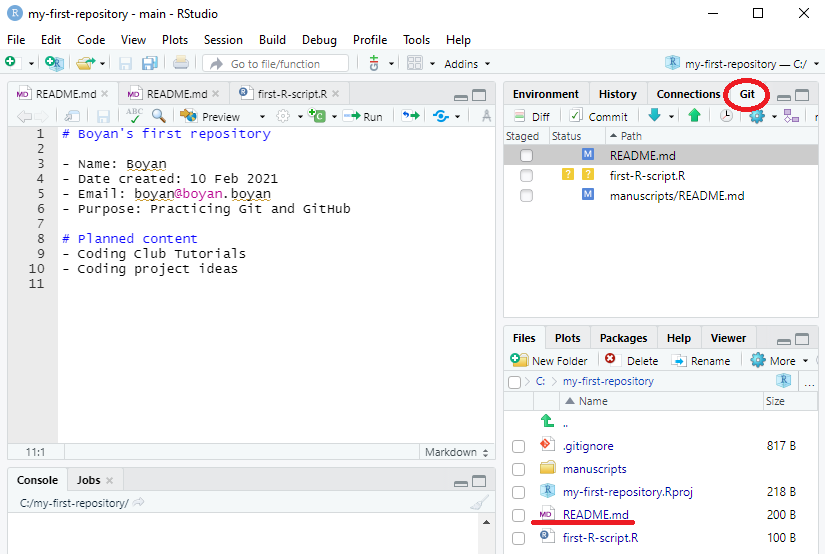
## Hiccups?

We know that there might be problems with the newest updates of the Mac software and installing git and linking it with RStudio. The solutions appear to be very specific to the Mac version you have, so if the above steps didn’t work, a good starting point is googling "rstudio can’t find git mac **your version**" and trying out the suggested solutions.

Once the files have finished copying across (this may take a while depending on the size of the repo you’re joining), you will notice that a few things about your RStudio session have changed: there is a Git tab in the top right corner of RStudio, and all the files that are in the repo are now on your computer as well.

You are now ready to start making changes and documenting them through Github! **Note that you can’t push empty folders.**

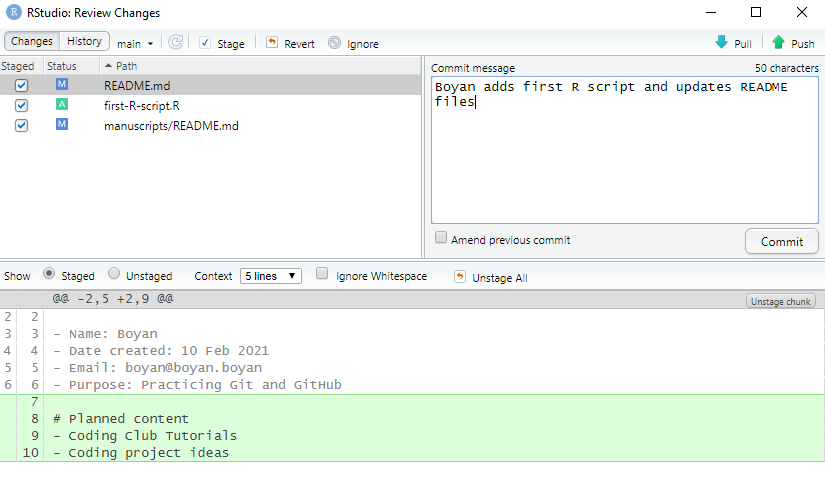
You can open some of the files you made online earlier - for example if you click on your README.md file, it will open in RStudio and you can make changes. Add some more text just for the sake of exemplifying how version control works. Save the file in the same location (i.e., your repository).



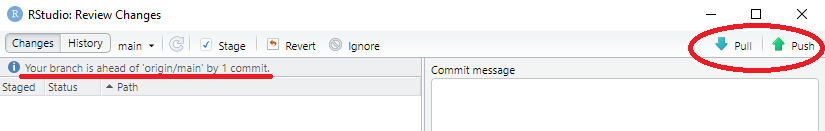
If you click on the Git tab you will see that now your README.md file is listed there. Add a tick next to it. Now it has an M - this means you have modified the file. If there’s an A, that’s an added file, and a D is a deleted file.

If you select the README.md file and click on Diff, you will see the changes you have made. Once the file is selected, it is staged, ready to be commited to Github.

Click on Commit and add in your commit message - aim to be concise and informative - what did you do? Once you have clicked on Commit, you will get a message about what changes you have made.

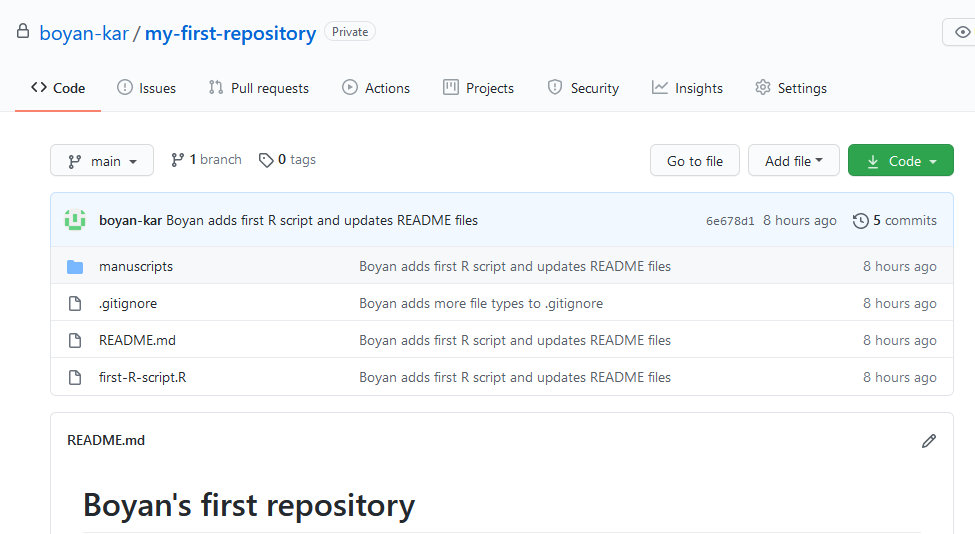


You will see a message saying that your branch is now one commit ahead of the origin/main branch - that is the branch that is on Github - we now need to let Github know about the changes we have made.

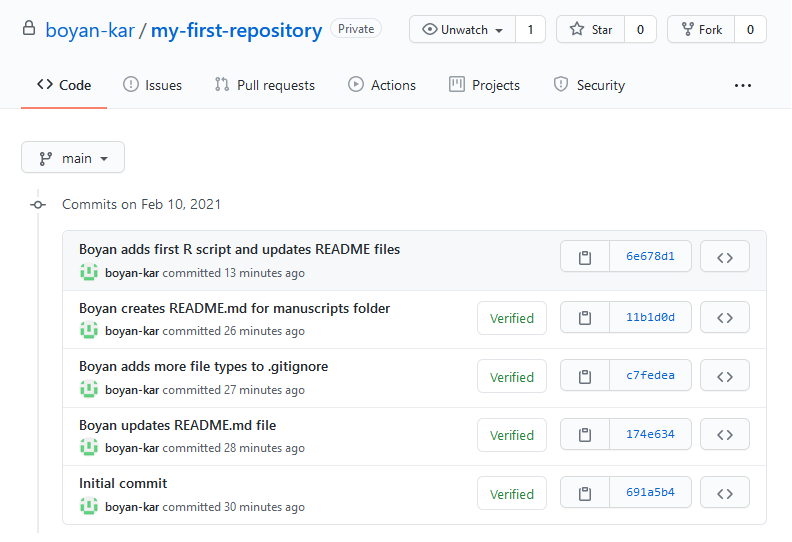


We can’t repeat it enough: **always Pull before you Push.** Pull means that you are retrieving the most recent version of the Github repository onto your local branch - this command is especially useful if several people are working within the same repository - imagine there was a second script examining soil pH along this elevation gradient, and your collaborator was working on it the same time as you - you wouldn’t want to “overwrite” their work and cause trouble. In this case, you are the only one working on these files, but it’s still good to develop the practice of pulling before you push. Once you’ve pulled, you’ll see a message that you are already up to date, you can now push! Click on Push, wait for the loading to be over and then click on Close - that was it, you have successfully pushed your work to Github!

Go back to your repository on Github, where you can now see all of your updated files online.



Click on your script file and then on History - this is where you can see the different versions of your script - obviously in real life situations you will make many changes as your work progresses - here we just have two. Thanks to Github and version control, you don’t need to save hundreds of almost identical files (e.g. Dissertation\_script\_25thFeb.R, Dissertation\_script\_26thFeb.R) - you have one file and by clicking on the different commits, you can see what it looked like at different points in time.



**You are now ready to add your scripts, plots, data files, etc. to your new project directory and follow the same workflow as outlined above - stage your files, commit, pull, push.**

## Potential problems

Sometimes you will see error messages as you try to commit-pull-push. Usually the error message identifies the problem and which file it’s associated with, if the message is more obscure, googling it is a good step towards solving the problem. Here are some potential problems that might arise:

### Code conflicts

While you were working on a certain part of a script, someone else was working on it, too. When you go through commit-pull-push, GitHub will make you decide which version you want to keep. This is called a code conflict, and you can’t proceed until you’ve resolved it. You will see arrows looking like >>>>>>>>> around the two versions of the code - delete the version of the code you don’t want to keep, as well as the arrows, and your conflict should disappear.

### Pushing the wrong files

If you accidentally push what you didn’t intend to, deleted many things (or everything!) and then pushed empty folders, you can revert your commit. You can keep reverting until you reach the point in time when everything was okay. This is an easy way out if you’re the only person working in the repository - **be aware that if there are other people that have committed to the repository, reverting will also undo all of their work, as reverting refers to the repository as a whole, not just your own work in it.**

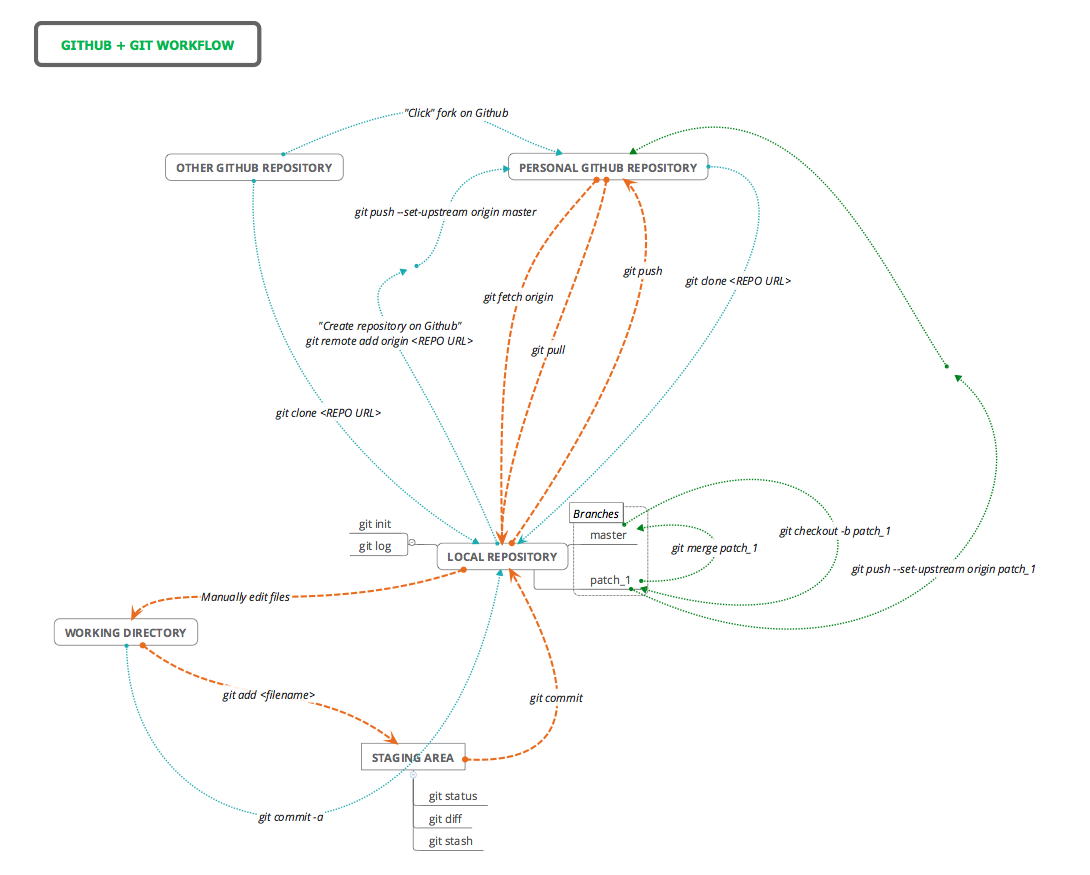
Using these “undo” commands can be daunting, so make sure you read up on the different commands before you attempt anything that may delete work permanently: [here’s a starter](https://www.atlassian.com/git/tutorials/undoing-changes/git-revert). It’s a good idea to regularly back up your repository to an external hard drive juuuust in case!

### Verified commits

When you browse your commit history through the GitHub website, you may notice that commits made through the website are listed as “Verified”, while commits pushed from your computer are not. This is generally not a big deal, but in large collaborative projects you may want to verify your locally made commits - [here is a guide how](https://docs.github.com/en/github/authenticating-to-github/about-commit-signature-verification).

# 4. Sync and interact with your repository through the command line

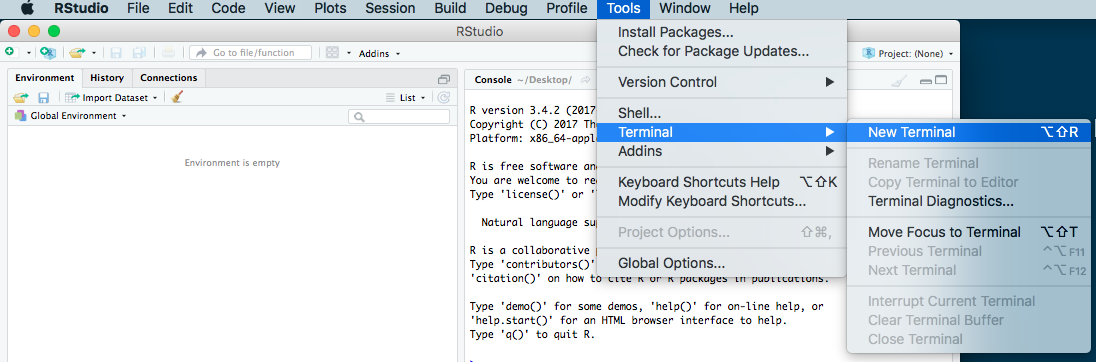
Traditionally, Git uses the command line to perform actions on local Git repositories. In this tutorial we ignored the command line but it is necessary if you want more control over Git. There are several excellent introductory guides on version control using Git, e.g. [Prof Simon Mudd’s Numeracy, Modelling and Data management guide](http://simon-m-mudd.github.io/NMDM_book/#_version_control_with_git), [The Software Carpentry guide](https://swcarpentry.github.io/git-novice/), and this [guide from the British Ecological Society Version Control workshop](https://github.com/BES2016Workshop/version-control). For more generic command line tools, look at this [general cheat sheet](https://www.git-tower.com/blog/command-line-cheat-sheet) and this [cheat sheet for mac users](https://github.com/0nn0/terminal-mac-cheatsheet). We have also created a table and flow diagram with some basic Git commands and how they fit into the Git/Github workflow. Orange lines refer to the core workflow, the blue lines describe extra functions and the green lines deal with branches:



|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Origin** | **Destination** | **Description** |
| git clone REPO\_URL | Personal Github | Local | Creates a local copy of a Github repo. The URL can be copied from Github.com by clicking the `Clone or Download` button. |
| git add README.md | Working Dir | Staging Area | Add "README.md" to staging area. |
| git commit | Staging Area | Local | Commits changes to files to the local repo. |
| git commit -a | Working Dir | Local | adds and commits all file changes to the local repo. |
| git pull | Personal Github | Local | Retrieve any changes from a Github repo. |
| git push | Local | Personal Github | Sends commited file changes to Github repo. |
| git merge | Other branch | Current branch | Merge any changes in the named branch with the current branch. |
| git checkout -b patch1 | NA | NA | Create a branch called "patch1" from the current branch and switch to it. |
| git init | NA | NA | Initialise a directory as a Git repo. |
| git log | NA | NA | Display the commit history for the current repo |
| git status | NA | NA | See which files are staged/unstaged/changed |
| git diff | NA | NA | See the difference between staged uncomitted changes and the most recent commit |
| git stash | NA | NA | Save uncommitted changes in a temporary version and revert to the most recent commit |

Below is a quick exercise so you can familiarise yourself with these command line tools. There are a few ways to use interact with Git using the terminal:

1. If you are already in RStudio on a Mac or Linux machine, you can open a terminal within RStudio by going to Tools -> Terminal -> New Terminal in the menu.
2. If you are on a Mac or Linux machine you could just open a terminal program and run Git from there. Most Mac and Linux machines will have Git installed by default. On Mac you can go open a terminal by going to: Applications/Utilities/Terminal.app.
3. If you are on a personal Windows machine, you can run Git using Git Bash, which can be installed when you installed Git. You should be able to launch it from More -> Shell in RStudio. If that doesn’t work, look up the programme under your Start Menu.



Once you have opened a terminal using one of the above methods, start by creating a folder somewhere on your local system called git\_test, using the mkdir (make directory) command by typing the following into the terminal and hitting “Enter”. For example, to create the directory in the Documents folder:

Copy contents

mkdir ~/Documents/git\_test

Then enter that folder using cd (change directory):

Copy contents

cd ~/Documents/git\_test

To make the folder into a Git repository:

Copy contents

git init

Now the folder has been made into a Git repository, allowing you to track changes to files. Now, lets create a README.md file inside the repository and put some text in it, using whatever text editor you are comfortable with. Make sure to place this README.md file into the repository folder on your device so it can be found!

You can create empty text files via a single command in the shell:

Copy contents

touch README.md

touch .gitignore

touch test.R

Now, to add a file to be tracked by the Git repository:

Copy contents

git add README.md

To check what files have been staged and unstaged changes:

Copy contents

git status

The README.md file has now been added to the staging area, but has not yet been committed to a version of the repository. To commit a version:

Copy contents

git commit **-m** "Your commit message here"

Currently, the Git repository is still only on our local computer. Versions are being committed, but they are not being backed up to a remote version of the repository on Github. Go to Github and create a repository called git\_test, like you did earlier on in the workshop, but this time don’t create a README.md because we have just made one on the local computer. Now, copy the HTTPS link for that repository. In the terminal, link the local Git repository with the remote repository using the following code, replacing <HTTPS\_LINK> with the link you copied:

Copy contents

git remote add origin <HTTPS\_LINK>

Then make the first push to that newly linked remote repository:

Copy contents

git push **-u** origin main

Now you can continue editing files, adding changes (git add <FILE>), committing changes (git commit), pulling (git pull) and pushing (git push) changes, similar to the process you did with clicking buttons in RStudio. Feel free to explore some of the more advanced commands laid out in the table and flow diagram above. You can also check out a more advanced command line tutorial written by [Prof Simon Mudd for Numeracy, Modelling and Data management guide](http://simon-m-mudd.github.io/NMDM_book/#_version_control_with_git).

## Difference Between Business Intelligence and Data Science

Business intelligence is a combination of the strategies and technologies used for the analysis of business data/information. Like data science, it can provide historical, current, and predictive views of business operations. However, there are some key differences.

|  |  |
| --- | --- |
| Business Intelligence | Data Science |
| Uses structured data | Uses both structured and unstructured data |
| Analytical in nature - provides a historical report of the data | Scientific in nature - perform an in-depth statistical analysis on the data |
| Use of basic statistics with emphasis on visualization (dashboards, reports) | Leverages more sophisticated statistical and predictive analysis and machine learning (ML) |
| Compares historical data to current data to identify trends | Combines historical and current data to predict future performance and outcomes |

## Applications of Data Science



Data science has found its applications in almost every industry.

#### 1. Healthcare

Healthcare companies are using data science to build sophisticated medical instruments to detect and cure diseases.

#### 2. Gaming

Video and computer games are now being created with the help of data science and that has taken the gaming experience to the next level.

#### 3. Image Recognition

Identifying patterns in images and detecting objects in an image is one of the most popular [data science applications](https://www.simplilearn.com/applications-of-data-science-deep-learning-ai-artificial-intelligence-article).

#### 4. Recommendation Systems

Netflix and Amazon give movie and product recommendations based on what you like to watch, purchase, or browse on their platforms.

#### 5. Logistics

Data Science is used by logistics companies to optimize routes to ensure faster delivery of products and increase operational efficiency.

#### 6. Fraud Detection

Banking and financial institutions use data science and related algorithms to detect fraudulent transactions.

## Data Science Use Cases

Here are some brief overviews of a couple of use cases, showing data science’s versatility.

* [Law Enforcement:](https://www.ibm.com/case-studies/police-charleroi) In this scenario, data science is used to help police in Belgium to better understand where and when to deploy personnel to prevent crime. With only limited resources and a large area to cover data science used dashboards and reports to increase the officers’ situational awareness, allowing a police force that’s spread thin to maintain order and anticipate criminal activity.
* [Pandemic Fighting:](https://www.ibm.com/case-studies/state-of-rhode-island/) The state of Rhode Island wanted to reopen schools, but was naturally cautious, considering the ongoing COVID-19 pandemic. The state used data science to expedite case investigations and contact tracing, enabling a small staff to handle an overwhelming number of concerned calls from citizens. This information helped the state set up a call center and coordinate preventative measures.
* [Driverless Vehicles:](https://www.ibm.com/case-studies/lunewave-inc-watson-ai-building-sensors-for-autonomous-vehicles) Lunewave, a sensor manufacturing company, was looking for a way to make sensor technology more cost-effective and accurate. They turned to data science and machine learning to train their sensors to be safer and more reliable, as well as using data to improve their 3D-printed sensor manufacturing process.

## FAQs

### What’s the difference between data science, artificial intelligence, and machine learning?

Artificial Intelligence makes a computer act/think like a human. Data science is an AI subset that deals with data methods, scientific analysis, and statistics, all used to gain insight and meaning from data. Machine learning is a subset of AI that teaches computers to learn things from provided data.

### Difference Between Data Analysis, Data Mining & Data Modeling

Data analysis is done with the purpose of finding answers to specific questions. Data analytics techniques are similar to business analytics and business intelligence.

Data Mining is about finding the different patterns in data. For this, various mathematical and computational algorithms are applied to data and new data will get generated.

Data Modeling is about how companies organize or manage the data. Here, various methodologies and techniques are applied to data. Data analysis is required for data modeling.

1. **What is Data Science in simple words?**

Data science is an AI subset that deals with data methods, scientific analysis, and statistics, all used to gain insight and meaning from data.

### What does a Data Scientist do?

A data scientist analyzes business data to extract meaningful insights.

### What is Data Science with an example?

Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. For example, finance companies can use a customer’s banking and bill-paying history to assess creditworthiness and loan risk.

### What kinds of problems do data scientists solve?

Data scientists solve issues like:

1. Loan risk mitigation
2. Pandemic trajectories and contagion patterns
3. Effectiveness of various types of online advertisement
4. Resource allocation
5. **Do data scientists code?**

Sometimes they may be called upon to do so.

### Can I learn Data Science on my own?

Data science is a complex field with many difficult technical requirements. It’s not advisable to try learning data science without the help of a structured learning program.

Turning data into actionable Information

Analyzing data, in general, assumes that the data has already been presented, or “reported” on–in the strict definition of the word. Analyzing literally means “taking apart,” i.e. sifting through something, breaking it down in its components to better understand it. Analysis in business intelligence is therefore the art of understanding data by “taking it apart” and asking it relevant questions.

Or, put an even better way, reporting presents data; analysis turns data into information. Information that, to be useful, can be then acted upon in the interest of the company’s strategy.

We can look at analysis as the simple act of asking your data questions. Take a table of data, for example, showing you a column of sales reps’ names and another column displaying total orders taken. The data is neutral. You can’t immediately make business sense of this simple table, unless you ask it questions. Now, ask the table “who has taken the highest amount of orders?” by sorting the second column, descending. Now the data has turned into information. A simple sort has been your way to ask your data a question, and you are therefore armed with the piece of information that (say), Jones is your top-performing sales rep.

Naturally, analysis can be much more complex than this. It can involve looking at your data from multiple dimensions (OLAP), spotting trends and exceptions, and even predicting future patterns. Regardless, what all these techniques have in common is that they turn neutral data into meaningful information.

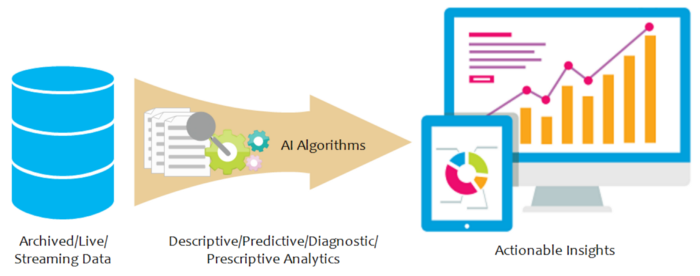
**The Goal of Analysis**

As we have said, analysis turns data into information. In business intelligence, this means asking relevant questions of your data so that you draw the necessary knowledge to make business decisions and take actions that further the company’s strategy.

**The Benefits of Analysis**

* Analysis is the step that lets users understand their data, turning it into information; without analysis, data loses its context and much of its meaning.
* Analysis empowers users to ask questions of their data; this is the main way in which users are said to “interact” with the data. In this sense, the more the analysis interface allows users to obtain meaningful questions of their data, the more it is interactive.
* Analysis lends the necessary answers that guide business end-users to making the correct decisions and taking appropriate action.
* Analysis highlights the critical factors and points the end-user towards them. By doing so, it facilitates prioritization and makes the business process more efficient.

**Turning Data into Actionable Insights**



**Data and Insights**

Insights are the new gold not the data as data is worth very little unless this data is turned into critical actionable insights. These insights can be used to support decision making and can help in refining the design and manufacturing processes. Machine learning algorithms can be used to accomplish different applied data mining tasks. These tasks can be descriptive analytics, predictive analytics, diagnostic analytics, and prescriptive analytics. Descriptive data analytics provides insight into the past and the present while predictive analytics forecasts the future. Diagnostic analytics provides root-cause analysis and prescriptive analytics advises on possible outcomes and their anticipated impacts.

**Data Analytics**

Descriptive data analytics provides a better understanding of the data and its nature and identifies patterns or relationships in the data. These descriptive models answer the following questions:

* What has happened?
* What happens now?
* What is the trend of a certain variable?
* What is the relationship between variables?
* How an item is performing w.r.t other item or a benchmark item?

Predictive models make prediction about future values of data and forecasts new proprieties instead of just exploring data properties like in case of descriptive analytics. These models answer the following questions:

* What would happen?
* When would it happen?
* Where would it happen?

Diagnostic analytics provides root-cause analysis to answer questions like:

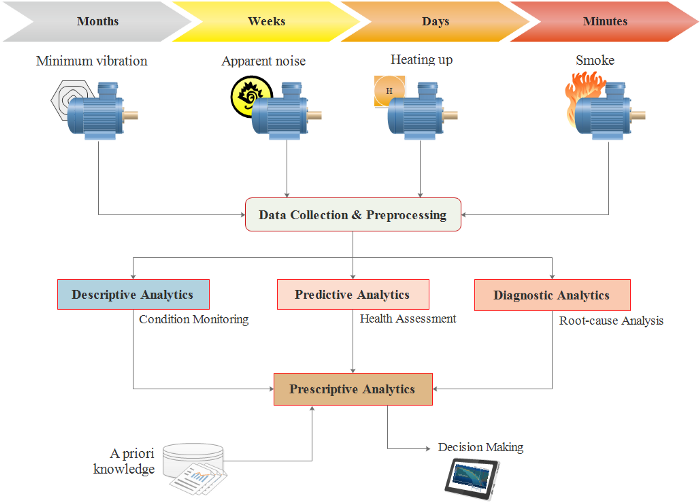
* Why did it happen?
* Why would it happen?

Last but not least, prescriptive models focus on decision support. This decision support or recommendation engine answers the following questions.

* How would the predictions obtained from predictive models impact everything else?
* What are the proactive decision/actions to be made?
* How we benefit from predictions/recommendations?
* What are the best actions to make?
* What is the best time to take this action?
* What would be the impact of this action?

**Predictive maintenance as an example**

Predictive maintenance is a preventive maintenance approach that relies on regular monitoring and analysis of actual machine condition, operating efficiency, other indicators of the operating condition in order to detect incipient problems, minimize the number and cost of unscheduled outages and prevent catastrophic failure of critical equipment. According to McKinsey, predictive maintenance will help companies save $630 Billion by 2025. Fortunately, electro-mechanical equipment does not break without warning. Months before the faults occur, minimum vibration can be found. Weeks before the fault, apparent noise begins to develop. Days before the machine heats up and minutes before the break down, it starts to smoke.



In this context, descriptive analytics can generate insight about the current conditions of the machine. While a predictive model can perform health assessment and anticipates any incipient problem or possible failure or anomalies. Diagnostic analytics provides root-cause analysis and finally, prescriptive analytics model can process information about the current status of the machine generated by descriptive analytics module and the health assessment from predictive analytics modules and any other a priori knowledge that may be available. It then produces timely recommendations/ decisions about proactive actions or maintenance plan to mitigate any possible risk. Using these recommended decisions and anticipated impact information generated by the prescriptive analytics model, the user can proactively plan maintenance considering the highest priority maintenance needs or can create work orders and messages the assigned technicians about the needed maintenance in order to keep the productivity and reduce down time and maintenance expenditures.

Data summarization, visualization, clustering, data association and sequence discovery are common tasks in descriptive analytics. While predictive analytics uses classification, regression and time-series analysis to provide information like anomaly prediction, failure risk score, time to failure or remaining useful life estimate and/or degrading trends. Diagnostic analytics used techniques like events and causal factor analysis, change analysis, barrier analysis, fault tree analysis and natural language processing to provide explanation for the causes of possible breakdown. Prescriptive analytics employs different AI techniques from machine learning, algorithmic inference, optimization and natural language processing to provide recommendations to mitigate any possible risk in the machine.

In the current weak/narrow AI wave, classification and regression are the most commonly matured and used ML techniques. Strong/general AI wave will mature more sophisticated ML techniques to process the data relying more on cognitive data collection and preparation, clustering, data association and time-series analysis including sequence discovery, trend/cyclicity/seasonality, similarity analysis and anomaly detection in archived, steaming and live data.

## TOP 14 SOFTWARE & TOOLS FOR DATA ANALYSTS (2022)

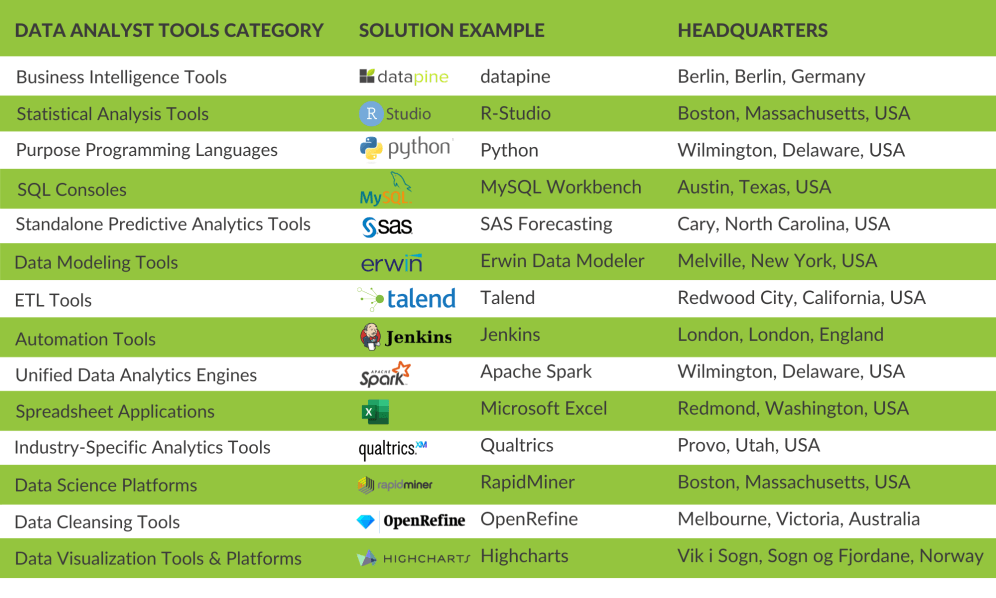
To be able to perform data analysis at the highest level possible, analysts and data professionals will use tools and software that will ensure the best results in several tasks from executing algorithms, preparing data, generate predictions, automate processes, to standard tasks such as visualizing and reporting on the data. Although there are many of these solutions on the market, data analysts must choose wisely in order to benefit their analytical efforts. That said, in this article, we will cover the best data analyst tools and name the key features of each based on various types of analysis processes. But first, we will start with a basic definition and a brief introduction.

## 1) WHAT ARE DATA ANALYST TOOLS?

**Data analyst tools** is a term used to describe software and applications that data analysts use in order to develop and perform analytical processes that help companies to make better, informed business decisions while decreasing costs and increasing profits.

In order to make the best possible decision on which software you need to choose as an analyst, a list of the top data analyst tools that have various focus and features, organized in software categories and represented with an example of each.

## 2) WHAT TOOLS DO DATA ANALYSTS USE?

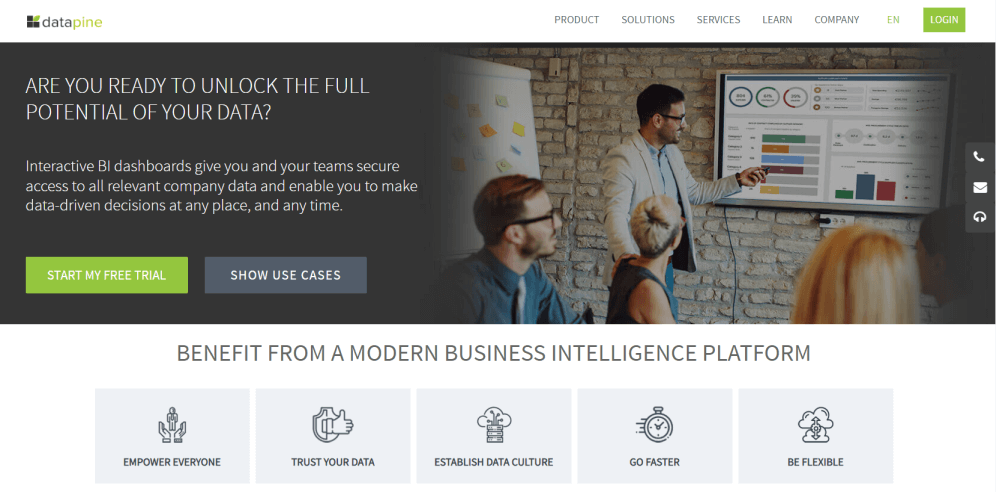


To make the most out of the infinite number of software that is currently offered on the market, we will focus on the most prominent tools needed to be an expert data analyst. These data analysis tools are mostly focused on making analyst’s life’s easier by providing them with solutions that make complex analytical tasks more efficient.

### 1. BUSINESS INTELLIGENCE TOOLS

[BI tools](https://www.datapine.com/articles/best-bi-tools-software-review-list) are one of the most represented means of performing data analysis. Specializing in business analytics, these tools will prove to be beneficial for every data analyst that needs to analyze, monitor, and report on important findings. Features such as self-service, predictive analytics, and advanced SQL modes make these solutions easily adjustable to every level of knowledge, without the need for heavy IT involvement. By providing a set of useful features, analysts can understand trends and make tactical decisions. Our data analytics tools article wouldn’t be complete without business intelligence, and datapine is one example that covers most of the requirements both for beginner and advanced users. This all-in-one tool aims to facilitate the entire analysis process from data integration and discovery to reporting.

### DATAPINE



**KEY FEATURES:**

Visual drag-and-drop interface to build SQL queries automatically, with the option to switch to, advanced (manual) SQL mode

Powerful predictive analytics features, interactive charts and dashboards, and automated reporting

AI-powered alarms that are triggered as soon as an anomaly occurs or a goal is met

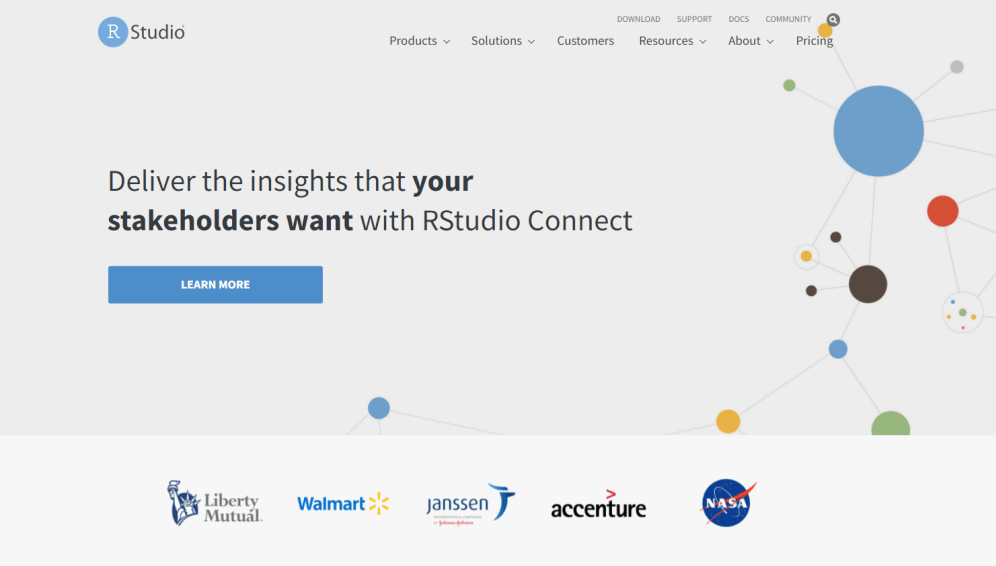
[datapine](https://www.datapine.com/) is a popular business intelligence software, that is focused on delivering simple, yet powerful analysis features into the hands of beginners and advanced users that need a fast and reliable [online data analysis](https://www.datapine.com/data-analysis-tools) solution for all analysis stages. An intuitive user interface will enable you to simply drag-and-drop your desired values into datapine’s Analyzer and create numerous charts and graphs that can be united into an interactive dashboard. If you’re an experienced analyst, you might want to consider the SQL mode where you can build your own queries or run existing codes or scripts. Another crucial feature is the predictive analytics forecast engine that can analyze data from multiple sources which can be previously integrated with their various data connectors. While there are numerous predictive tools out there, datapine provides simplicity and speed at its finest. By simply defining the input and output of the forecast based on specified data points and desired model quality, a complete chart will unfold together with predictions.

We should also mention robust artificial intelligence that is becoming an invaluable assistant in today’s analysis processes. Neural networks, pattern recognition, and threshold alerts will alarm you as soon as a business anomaly occurs or a previously set goal is met so you don’t have to manually analyze large volumes of data – the data analytics software does it for you. Access your data from any device with an internet connection, and share your findings easily and securely via dashboards or customized reports for anyone that needs quick answers to any type of business question.

### 2. STATISTICAL ANALYSIS TOOLS

Next in our list of data analytics tools comes a more technical are related to statistical analysis. Referring to computation techniques that often contain a variety of statistical techniques to manipulate, explore, and generate insights, there exist multiple programming languages to make (data) scientists’ work easier and more effective. With the expansion of various languages that are today present on the market, science has its own set of rules and scenarios that need special attention when it comes to statistical data analysis and modeling. Here we will present one of the most popular tools for a data analyst – R programming. Although there are other languages that focus on (scientific) data analysis, R is particularly popular in the community.

### R PROGRAMMING / R-STUDIO



**KEY FEATURES:**

An ecosystem of more than 10 000 packages and extensions for distinct types of data analysis

Statistical analysis, modeling, and hypothesis testing (e.g. analysis of variance, t test, etc.)

Active and communicative community of researchers, statisticians, and scientists

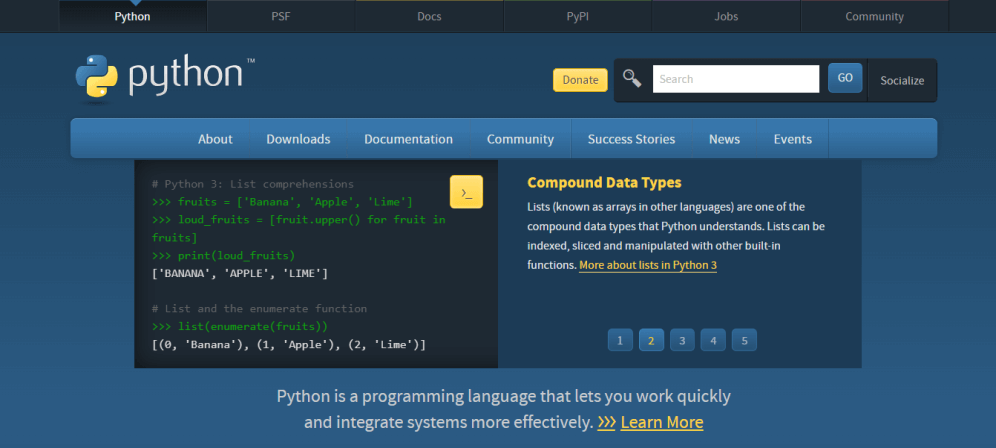
R is one of the top data analyst tools that is usually referred to as a language designed by statisticians. It’s development dates back to 1995 and it’s one of the most used tools for statistical analysis and data science, keeping an open- source policy and running on a variety of platforms, including Windows and macOS. [RStudio](https://rstudio.com/" \t "_blank) is by far the most popular integrated development environment. R’s capabilities for data cleaning, data reduction, and data analysis report output with R markdown, makes this tool an invaluable analytical assistant that covers both general and academic data analysis. It is compiled of an ecosystem of more than 10 000 packages and extensions that you can explore by categories, and perform any kind of statistical analysis such as regression, conjoint, factor cluster analysis, etc. Easy to understand for those that don’t have a high-level of programming skills, R can perform complex mathematical operations by using a single command. A number of graphical libraries such as ggplot and plotly make this language different than others in the statistical community since it has efficient capabilities to create quality visualizations.

R was mostly used in the academic area in the past, today it has applications across industries and large companies such as Google, Facebook, Twitter, and Airbnb, among others. Due to an enormous number of researchers, scientists, and statisticians using it, R has an extensive and active community where innovative technologies and ideas are presented and communicated regularly.

### 3. GENERAL-PURPOSE PROGRAMMING LANGUAGES

Programming languages are used to solve a variety of data problems. We have explained R and statistical programming, now we will focus on general ones that use letters, numbers, and symbols to create programs and require formal syntax used by programmers. Often, they’re also called text-based programs because you need to write software that will ultimately solve a problem. Examples include C#, Java, PHP, Ruby, Julia, and Python, among many others on the market. Here we will present Python as one of the best tools for data analysts that have coding knowledge as well.

### PYTHON



**KEY FEATURES:**

An open-source solution that has simple coding processes and syntax so it’s fairly easy to learn

Integration with other languages such as C/C++, Java, PHP, C#, etc.

Advanced analysis processes through machine learning and text mining

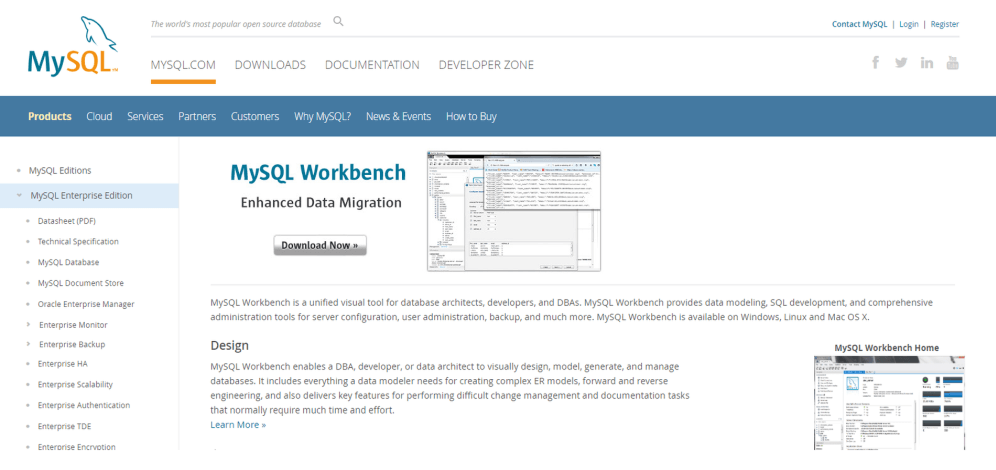
[Python](https://www.python.org/) is extremely accessible to code in comparison to other popular languages such as Java, and its syntax is relatively easy to learn making this tool popular among users that look for an open-source solution and simple coding processes. In data analysis, Python is used for data crawling, cleaning, modeling, and constructing analysis algorithms based on business scenarios. One of the best features is actually its user-friendliness: programmers don’t need to remember the architecture of the system nor handle the memory – Python is considered a high-level language that is not subject to the computer’s local processor.

Another noticeable feature of Python is its portability. Users can simply run the code on several operating systems without making any changes to it so it’s not necessary to write completely new code. This makes Python a highly portable language since programmers can run it both on Windows and macOS. An extensive number of modules, packages and libraries make Python a respected and usable language across industries with companies such as Spotify, Netflix, Dropbox and Reddit as the most popular ones that use this language in their operations. With features such as text mining and machine learning, Python is becoming a respected authority for advanced analysis processes.

### 4. SQL CONSOLES

Our data analyst tools list wouldn’t be complete without SQL consoles. Essentially, SQL is a programming language that is used to manage/query data held in relational databases, particularly effective in handling structured data as a database tool for analysts. It’s highly popular in the data science community and one of the analyst tools used in various business cases and data scenarios. The reason is simple: as most of the data is stored in relational databases and you need to access and unlock its value, SQL is a highly critical component of succeeding in business, and by learning it, analysts can offer a competitive advantage to their skillset. There are different relational (SQL-based) database management systems such as MySQL, PostgreSQL, MS SQL, and Oracle, for example, and by learning these data analysts’ tools would prove to be extremely beneficial to any serious analyst. Here we will focus on MySQL Workbench as the most popular one.

### MYSQL WORKBENCH



**KEY FEATURES:**

A unified visual tool for data modeling, SQL development, administration, backup, etc.

Instant access to database schema and objects via the Object Browser

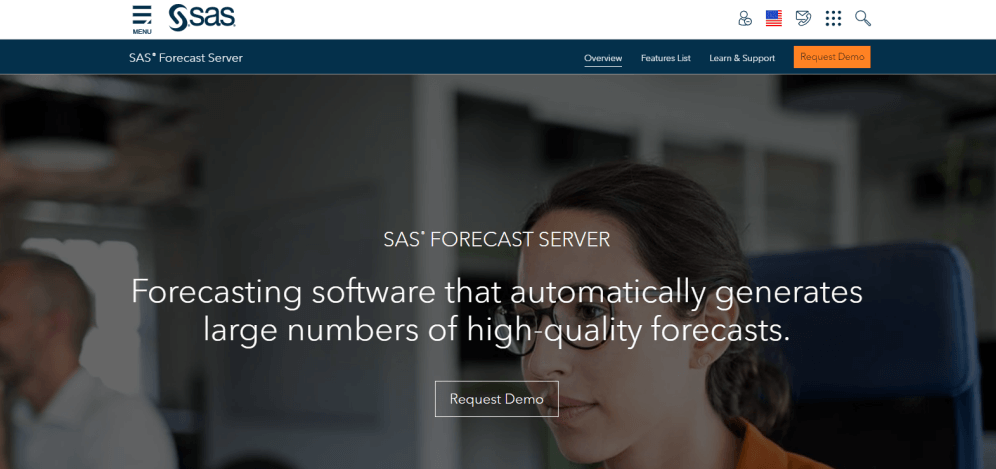
SQL Editor that offers color syntax highlighting, reuse of SQL snippets, and execution history

[MySQL Workbench](https://www.mysql.com/products/workbench/) is used by analysts to visually design, model, and manage databases, optimize SQL queries, administer MySQL environments, and utilize a suite of tools to improve the performance of MySQL applications. It will allow you to perform tasks such as creating and viewing databases and objects (triggers or stored procedures, e.g.), configuring servers, and much more. You can easily perform backup and recovery as well as inspect audit data. MySQL Workbench will also help in database migration and is a complete solution for analysts working in relational database management and companies that need to keep their databases clean and effective.

### 5. STANDALONE PREDICTIVE ANALYTICS TOOLS

Predictive analytics is one of the advanced techniques, used by analysts that combine data mining, machine learning, predictive modeling, and artificial intelligence to predict future events, and it deserves a special place in our list of data analysis tools as its popularity increases in recent years with the introduction of smart solutions that enabled analysts to simplify their predictive analytics processes. You should keep in mind that some BI tools we already discussed in this list offer easy to use, built-in predictive analytics solutions but, in this section, we focus on standalone, advanced predictive analytics that companies use for various reasons, from detecting fraud with the help of pattern detection to optimizing marketing campaigns by analyzing consumers’ behavior and purchases. Here we will list a data analysis software that is helpful for predictive analytics processes and helps analysts to predict future scenarios.

### SAS FORECASTING



**KEY FEATURES:**

Automatic forecasting for a large number of entities or products, including hierarchical forecasting

Scalability and modeling by combining 2 or more models and creating an ensemble

An unlimited model repository that includes time series and casual methods such as ARIMA and ARIMAX

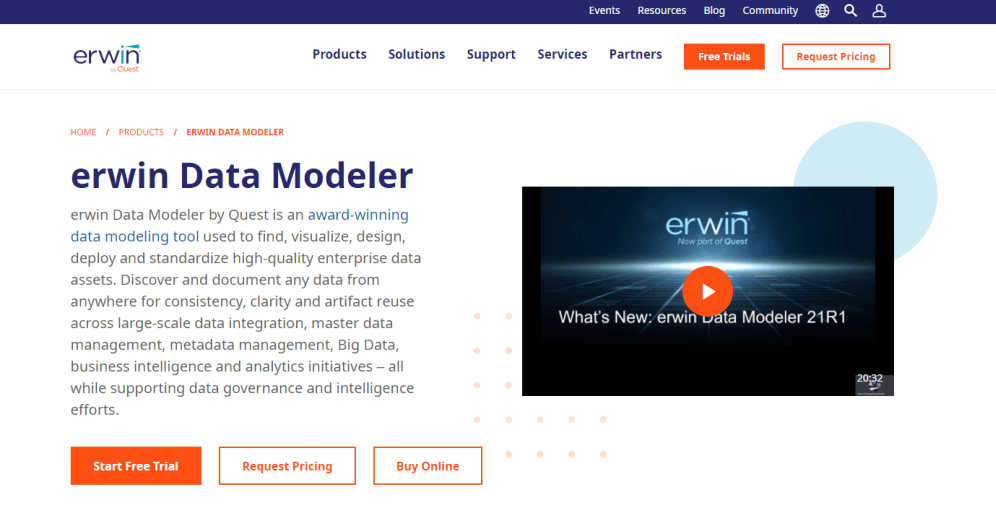
[SAS Forecasting for Desktop](https://www.sas.com/en_us/software/forecasting-for-desktop.html) has established itself as one of the most prominent advanced data analysis software that offers a wide range of forecasting methods, including hierarchical reconciliation, event modeling, what-if analysis, and scenario planning. Their features comprise 7 core areas of forecasting processes, some of them we already mentioned: automatic forecasting, scalability and modeling, unlimited model repository, easy-to-use GUI, event-modeling console, what-if analysis, and data preparation. Based on the variables that you enter in the modeling process, SAS will automatically select variables to generate forecasts to unravel what happens in your business.

Moreover, with a bundle of the SAS Forecast Server, and Visual Forecasting solutions, this data software enables users to produce a large number of forecasts, and automate their processes. Since the company is on the market for decades, they have established themselves as an authority figure in predictive analytics, and it certainly makes sense to give them a try.

### 6. DATA MODELING TOOLS

Our list of data analysis tools for analysts wouldn’t be complete without data modeling. Creating models to structure the database and design business systems by utilizing diagrams, symbols, and text, ultimately represent how the data flows and is connected in between. Businesses use data modeling tools to determine the exact nature of the information they control and the relationship between datasets, and analysts are critical in this process. If you need to discover, analyze, and specify changes on information that is stored in a software system, database or other application, chances are your skills are critical for the overall business. Here we will show one of the most popular data analyst software used to create models and design your data assets.

### ERWIN DATA MODELER (DM)



**KEY FEATURES:**

Automated data model generation to increase productivity in analytical processes

Single interface no matter the location or the type of the data

7 different versions of the solution you can choose from and adjust based on your business needs

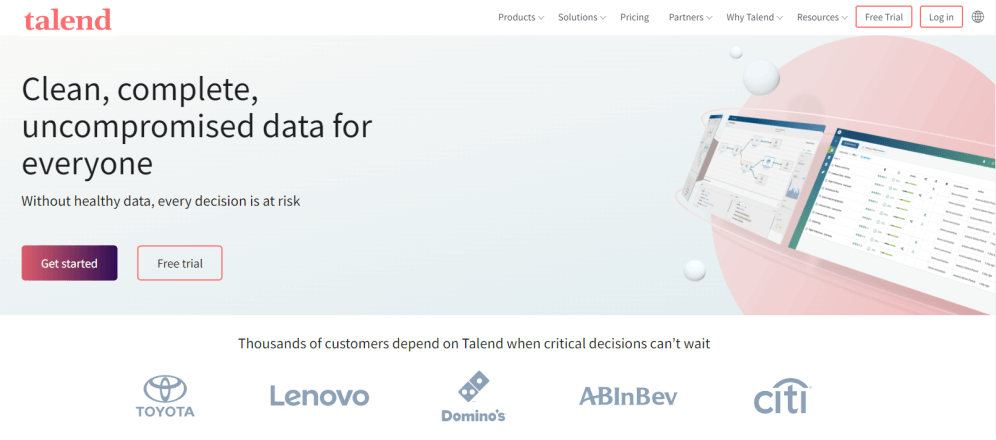
[erwin DM](https://erwin.com/products/erwin-data-modeler/) works both with structured and unstructured data in a data warehouse and in the cloud. It’s used to “find, visualize, design, deploy and standardize high-quality enterprise data assets,” as stated on their official website. erwin can help you reduce complexities and understand data sources to meet your business goals and needs. They also offer automated processes where you can automatically generate models and designs to reduce errors and increase productivity. This is one of the tools for analysts that focus on the architecture of the data and enable you to create logical, conceptual, and physical data models.

Additional features such as a single interface for any data you might possess, no matter if it’s structured or unstructured, in a data warehouse or the cloud makes this solution highly adjustable for your analytical needs. With 7 versions of the erwin data modeler, their solution is highly adjustable for companies and analysts that need various data modeling features.

### 7. ETL TOOLS

ETL is a process used by companies, no matter the size, across the world, and if a business grows, chances are you will need to extract, load and transform data into another database to be able to analyze it and build queries. There are some core types of ETL tools such as batch ETL, real-time ETL, and cloud based ETL, each with its own specifications and features that adjust to different business needs. These are the tools used by analysts that take part in more technical processes of data management within a company, and one of the best examples is Talend.

### TALEND



**KEY FEATURES:**

Collecting and transforming data through data preparation, integration, cloud pipeline designer

Data governance feature to build a data hub and resolve any issues in data quality

Sharing data through comprehensive deliveries via APIs

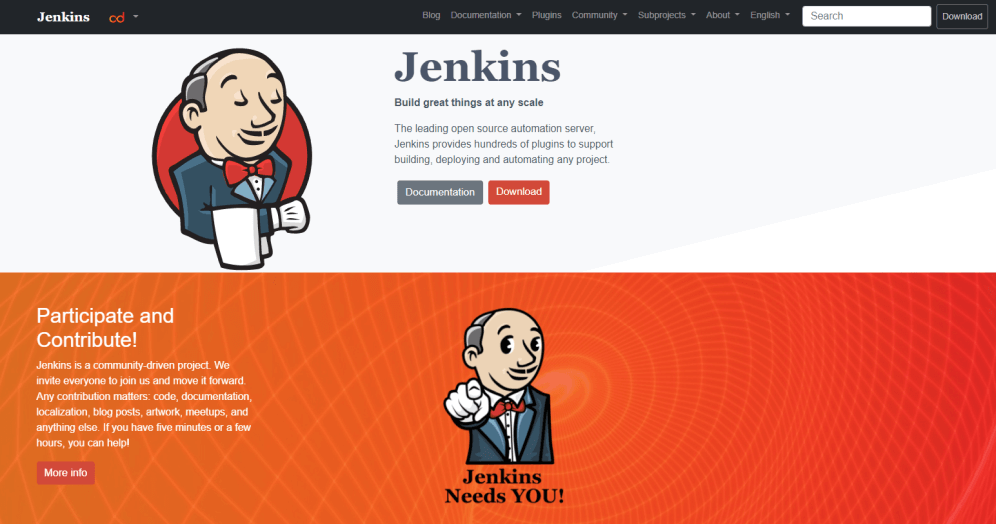
[Talend](https://www.talend.com/) is a data integration platform used by experts across the globe for data management processes, cloud storage, enterprise application integration, and data quality. It’s a Java-based ETL tool that is used by analysts in order to easily process millions of data records, and offers comprehensive solutions for any data project you might have. Talend’s features include (big) data integration, data preparation, cloud pipeline designer, and stitch data loader to cover multiple data management requirements of an organization. This is an analyst software extremely important if you need to work on ETL processes in your analytical department.

Apart from collecting and transforming data, Talend also offers a data governance solution to build a data hub and deliver it through self-service access through a unified cloud platform. You can utilize their data catalog, inventory and produce clean data through their data quality feature. Sharing is also part of their data portfolio; Talend’s data fabric solution will enable you to deliver your information to every stakeholder through a comprehensive API delivery platform. If you need a data analyst tool to cover ETL processes, Talend might be worth considering.

### 8. AUTOMATION TOOLS

As mentioned, the goal of all the solutions present on this list is to make data analysts life’s easier and more efficient. Taking that into account, automation tools could not be left out of this list. In simple words, data analytics automation is the practice of using systems and processes to perform analytical tasks with almost no human interaction. In the past years, automation solutions have impacted the way analysts perform their jobs as these tools assist them in a variety of tasks such as data discovery, preparation, data replication, and more simple ones like report automation or writing scripts. That said, automating analytical processes significantly increases productivity, leaving more time to perform more important tasks. We will see this more in detail through Jenkins one of the leaders in open-source automation tools.

### JENKINS



**KEY FEATURES:**

popular continuous integration (CI) solution with advanced automation features such as running code in multiple platforms

Job automations to set up customized tasks can be scheduled or based on a specific event

Several job automation plugins for different purposes such as Jenkins Job Builder, Jenkins Job DLS or Jenkins Pipeline DLS

Developed in 2004 under the name Hudson, [Jenkins](https://www.jenkins.io/) is an open-source CI automation server that can be integrated with several DevOps tools via plugins. By default, Jenkins assists developers to automate parts of their software development process like building, testing, and deploying. However, it is also highly used by data analysts as a solution to automate jobs such as running codes and scripts daily or when a specific event happened. For example, run a specific command when new data is available.

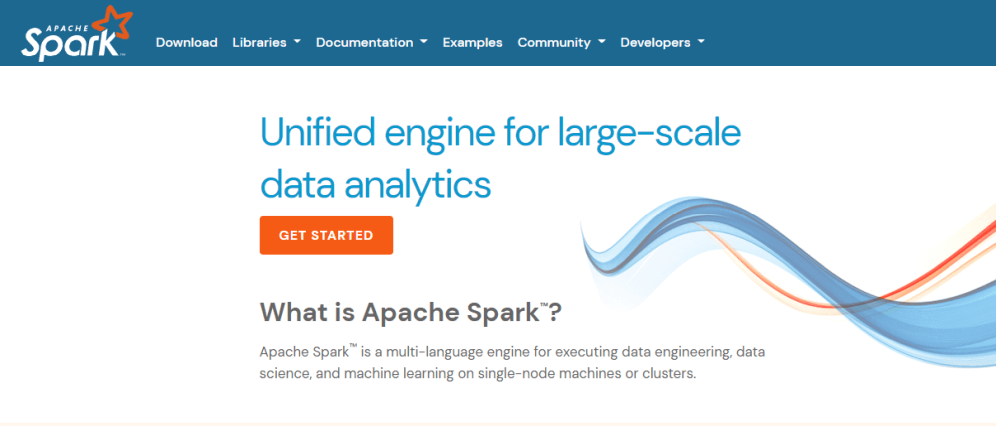
There are several Jenkins’s plugins to generate jobs automatically. For example, the Jenkins Job Builder plugin takes simple descriptions of jobs in YAML or JSON format and turns them into runnable jobs in Jenkins’s format. On the other side, the Jenkins Job DLS plugin provides users with the capabilities to easily generate jobs from other jobs and edit the XML configuration to supplement or fix any existing elements in the DLS. Lastly, the Pipeline plugin is mostly used to generate complex automated processes.

For Jenkins, automation is not useful if it’s not tight to integration. For this reason, they provide hundreds of plugins and extensions to integrate Jenkins with your existing tools. This way, the entire process of code generation and execution can be automated at every stage and in different platforms - leaving analysts enough time to perform other relevant tasks. All the plugins and extensions from Jenkins are developed in Java meaning the tool can also be installed in any other operator that runs on Java.

### 9. UNIFIED DATA ANALYTICS ENGINES

If you work for a company that produces massive datasets and needs a big data management solution, then unified data analytics engines might be the best resolution for your analytical processes. To be able to make quality decisions in a big data environment, analysts need tools that will enable them to take full control of their company’s robust data environment. That’s where machine learning and AI play a significant role. That said, Apache Spark is one of the data analysis tools on our list that supports big-scale data processing with the help of an extensive ecosystem.

### APACHE SPARK



**KEY FEATURES:**

High performance: Spark owns the record in the large-scale data processing

A large ecosystem of data frames, streaming, machine learning, and graph computation

A collection of over 100 operators for transforming and operating on large scale data

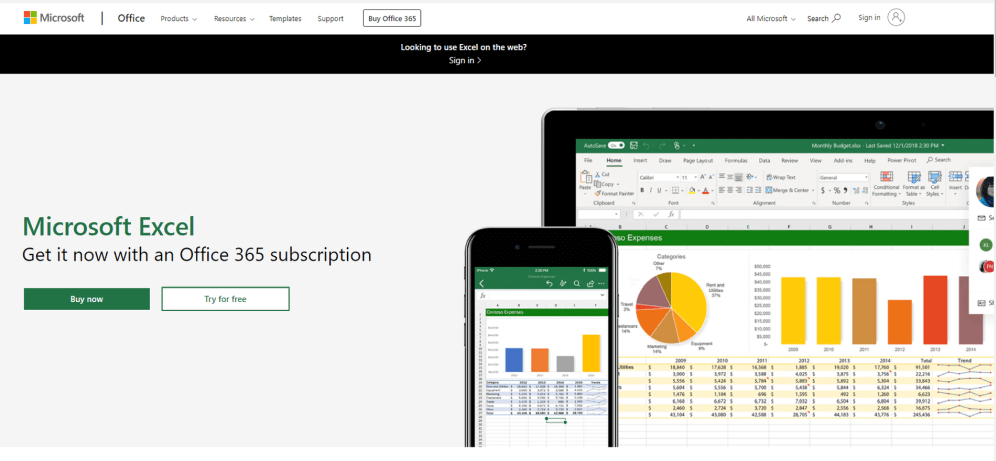
[Apache Spark](https://spark.apache.org/) is originally developed by UC Berkeley in 2009 and since then, it has expanded across industries and companies such as Netflix, Yahoo, and eBay that have deployed Spark, processed petabytes of data and proved that Apache is the go-to solution for big data management. Their ecosystem consists of Spark SQL, streaming, machine learning, graph computation, and core Java, Scala, and Python APIs to ease the development. Already in 2014, Spark has officially [set a record](https://databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html) in large-scale sorting. Actually, the engine can be 100x faster than Hadoop and this is one of the features that is extremely crucial for massive volumes of data processing.

You can easily run applications in Java, Python, Scala, R, and SQL while more than 80 high-level operators that Spark offers will make your data transformation easy and effective. As a unified engine, Spark comes with support for SQL queries, MLlib for machine learning and GraphX for streaming data that can be combined to create additional, complex analytical workflows. Additionally, it runs on Hadoop, Kubernetes, Apache Mesos, standalone or in the cloud and can access diverse data sources. Spark is truly a powerful engine for analysts that need support in their big data environment.

### 10. SPREADSHEET APPLICATIONS

Spreadsheets are one of the most traditional forms of data analysis. Quite popular in any industry, business or organization, there is a slim chance that you haven’t created at least one spreadsheet to analyze your data. Often used by people that don’t have high technical abilities to code themselves, spreadsheets can be used for fairly easy analysis that doesn’t require considerable training, complex and large volumes of data and databases to manage. To look at spreadsheets in more detail, we have chosen Excel as one of the most popular in business.

### EXCEL



**KEY FEATURES:**

Part of the Microsoft Office family, hence, it’s compatible with other Microsoft applications

Pivot tables and building complex equations through designated rows and columns

Perfect for smaller analysis processes through workbooks and quick sharing

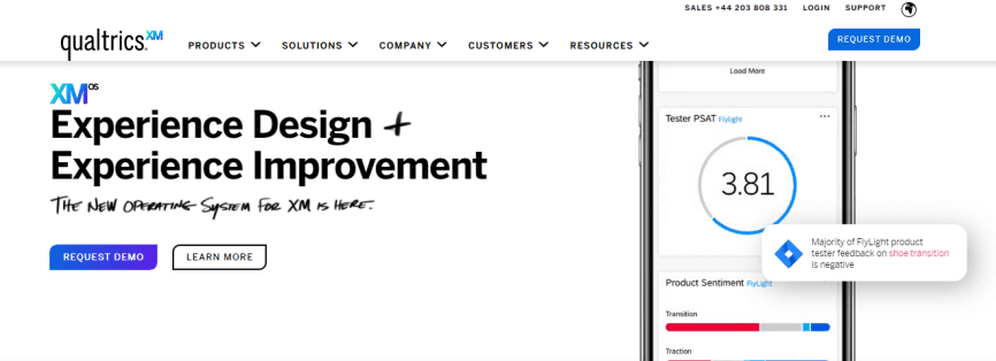
[Excel](https://products.office.com/en/excel) needs a category on its own since this powerful tool has been in the hands of analysts for a very long time. Often considered as a traditional form of analysis, Excel is still widely used across the globe. The reasons are fairly simple: there aren’t many people who have never used it or came across it at least once in their career. It’s a fairly versatile data analyst tool where you simply manipulate rows and columns to create your analysis. Once this part is finished, you can export your data and send it to the desired recipients, hence, you can use Excel as a [report tool](https://www.datapine.com/online-reporting) as well. You do need to update the data on your own, Excel doesn’t have an automation feature similar to other tools on our list. Creating pivot tables, managing smaller amounts of data and tinkering with the tabular form of analysis, Excel has developed as an electronic version of the accounting worksheet to one of the most spread tools for data analysts.

A wide range of functionalities accompany Excel, from arranging to manipulating, calculating and evaluating quantitative data to building complex equations and using pivot tables, conditional formatting, adding multiple rows and creating charts and graphs – Excel has definitely earned its place in traditional data management.

### 11. INDUSTRY-SPECIFIC ANALYTICS TOOLS

While there are many data analysis tools on this list that are used in various industries and are applied daily in analysts’ workflow, there are solutions that are specifically developed to accommodate a single industry and cannot be used in another. For that reason, we have decided to include of one these solutions on our list, although there are many others, industry-specific data analysis programs and software. Here we focus on Qualtrics as one of the leading research software that is used by over 11000 world’s brands and has over 2M users across the globe as well as many industry-specific features focused on market research.

### QUALTRICS



**KEY FEATURES:**

4 main experience features: customer, brand, employee, and product

Additional research services by their in-house experts

Advanced statistical analysis with their Stats iQ analysis tool

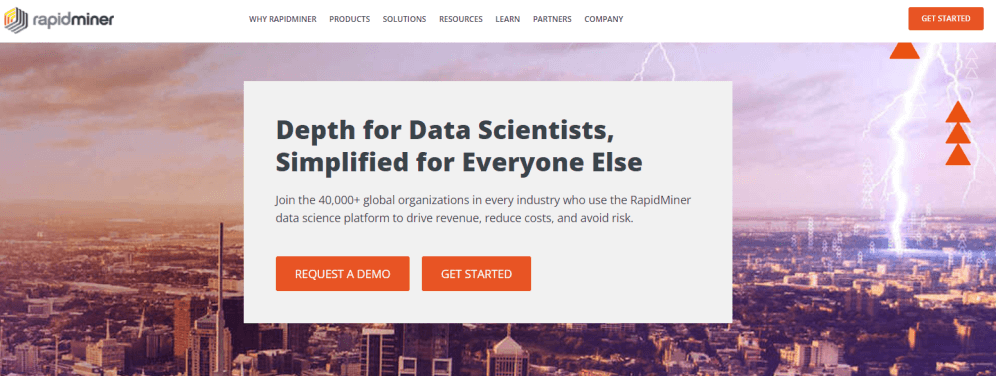
[Qualtrics](https://www.qualtrics.com/) is a software for data analysis that is focused on experience management and is used for market research by companies across the globe. They offer 4 product pillars: the customer experience, brand, employee, and product experience, and additional research services performed by their own experts. Their XM platform consists of a directory, automated actions, Qualtrics iQ tool, and platform security features that combine automated and integrated workflows into a single point of access. That way, users can refine each stakeholder’s experience and use their tool as an “ultimate listening system.”

Since automation is becoming increasingly important in our data-driven age, Qualtrics has also developed drag-and-drop integrations into the systems that companies already use such as CRM, ticketing, or messaging, while enabling users to deliver automatic notifications to the right people. This feature works across brand tracking and product feedback as well as customer and employee experience. Other critical features such as the directory where users can connect data from 130 channels (including web, SMS, voice, video, or social), and Qualtrics iQ to analyze unstructured data will enable users to utilize their predictive analytics engine and build detailed customer journeys. If you’re looking for a data analytic software that needs to take care of market research of your company, Qualtrics is worth the try.

### 12. DATA SCIENCE PLATFORMS

Data science can be used for most software solutions on our list, but it does deserve a special category since it has developed into one of the most sought-after skills of the decade. No matter if you need to utilize preparation, integration or data analyst reporting tools, data science platforms will probably be high on your list for simplifying analytical processes and utilizing advanced analytics models to generate in-depth data science insights. To put this into perspective, we will present RapidMiner as one of the top data analyst software that combines deep but simplified analysis.

### RAPIDMINER



**KEY FEATURES:**

A comprehensive data science and machine learning platform with more than 1500 algorithms

Possible to integrate with Python and R as well as support for database connections (e.g. Oracle)

Advanced analytics features for descriptive and prescriptive analytics

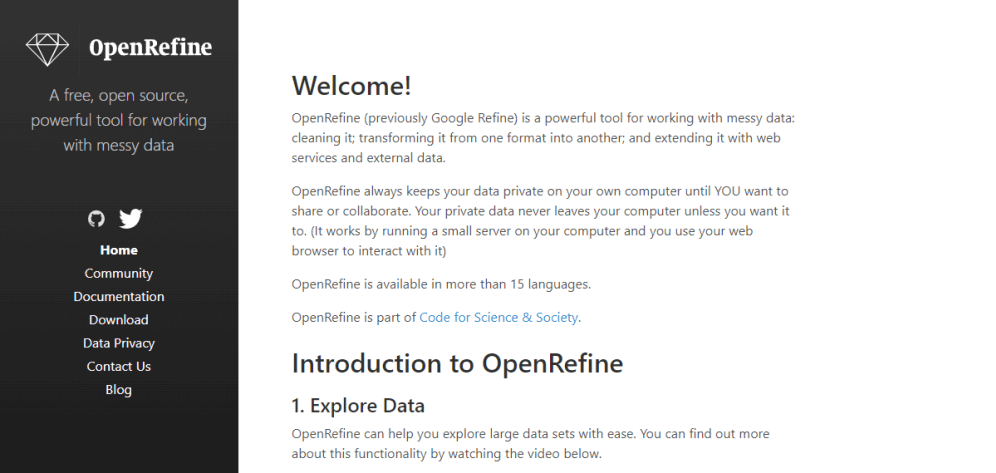
[RapidMiner](https://rapidminer.com/) is a tool used by data scientists across the world to prepare data, utilize machine learning and model operations in more than 40 000 organizations that heavily rely on analytics in their operations. By unifying the entire data science cycle, RapidMiner is built on 5 core platforms and 3 automated data science products that help in the design and deploy analytics processes. Their data exploration features such as visualizations and descriptive statistics will enable you to get the information you need while predictive analytics will help you in cases such as churn prevention, risk modeling, text mining, and customer segmentation.

With more than 1500 algorithms and data functions, support for 3rd party machine learning libraries, integration with Python or R, and advanced analytics, RapidMiner has developed into a data science platform for deep analytical purposes. Additionally, comprehensive tutorials and full automation, where needed, will ensure simplified processes if your company requires them, so you don’t need to perform manual analysis. If you’re looking for analyst tools and software focused on deep data science management and machine learning, then RapidMiner should be high on your list.

### 13. DATA CLEANSING PLATFORMS

The amount of data being produced is only getting bigger, hence, the possibility of it involving errors. To help analysts avoid these errors that can damage the entire analysis process is that data cleansing solutions were developed. These tools help analysts prepare their data by eliminating errors, inconsistencies, and duplications enabling them to extract accurate conclusions from the data. Before cleansing platforms were a thing, analysts would manually clean the data, this is also a dangerous practice since the human eye is prompt to error. That said, powerful cleansing solutions have proved to boost efficiency and productivity while providing a competitive advantage as data becomes reliable. The cleansing software we picked for this section is a popular solution named OpenRefine.

### OPENREFINE



**KEY FEATURES:**

Data explorer to clean “messy” data using transformations, facets, clustering, among others

Transform data to the format you desire, for example, turn a list into a table by importing the file into OpenRefine

Includes a large list of extensions and plugins to link and extend datasets with various web services

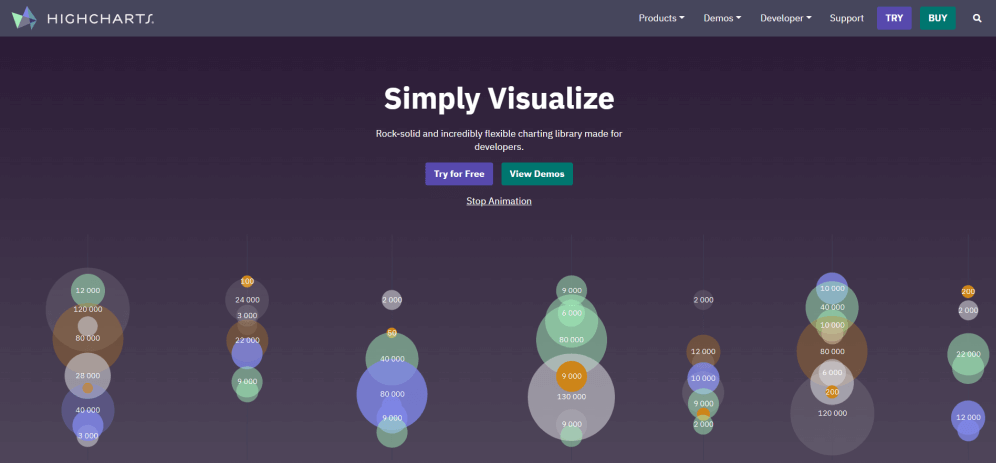
Previously known as Google Refine, [OpenRefine](https://openrefine.org/" \t "_blank) is a Java-based open-source desktop application for working with large sets of data that needs to be cleaned. The tool also enables users to transform their data from one format to another and extend it with web services and external data. OpenRefine has a similar interface to the one of spreadsheet applications and can handle CSV file formats, but all in all, it behaves more as a database. Upload your datasets into the tool and use their multiple cleaning features that will let you spot anything from extra spaces to duplicated fields.

Available in more than 15 languages, one of the main principles of OpenRefine is privacy. The tool works by running a small server on your computer and your data will never leave that server unless you decide to share it with someone else.

### 14. DATA VISUALIZATION TOOLS & PLATFORMS

Data visualization has become one of the most indispensable elements of data analytics tools. If you’re an analyst, there is probably a strong chance you had to develop a visual representation of your analysis or utilize some form of data visualization. Here we need to make clear that there are differences between professional data visualization tools often integrated through already mentioned BI tools, free available solutions as well as paid charting libraries. They’re simply not the same. Also, if you look at data visualization in a broad sense, Excel and PowerPoint also have it on offer, but they simply cannot meet the advanced requirements of a data analyst who usually chooses professional BI or data viz tools as well as modern charting libraries, as mentioned. We will take a closer look at Highcharts as one of the most popular charting libraries on the market.

### HIGHCHARTS



**KEY FEATURES:**

Interactive JavaScript engine for charts used in web and mobile projects

Designed mostly for a technical-based audience (developers)

WebGL-powered boost module to render millions of datapoints directly in the browser

[Highcharts](https://www.highcharts.com/) is a multi-platform library that is designed for developers looking to add interactive charts into web and mobile projects. This charting library works with any back-end database and data can be given in CSV, JSON or updated live. They also feature intelligent responsiveness that fits the desired chart into the dimensions of the specific container but also places non-graph elements in the optimal location automatically.

Highcharts supports line, spline, area, column, bar, pie, scatter charts and many others that help developers in their online-based projects. Additionally, their WebGL-powered boost module enables you to render millions of datapoints in the browser. As far as the source code is concerned, they allow you to download and make your own edits, no matter if you use their free or commercial license. In essence, Basically, Highcharts is designed mostly for the technical target group so you should familiarize yourself with developers’ workflow and their JavaScript charting engine. If you’re looking for a more easy to use but still powerful solution, you might want to consider an [online data visualization tool](https://www.datapine.com/data-visualization-tools) like datapine.

## 3) KEY TAKEAWAYS & GUIDANCE

We have explained what are data analyst tools and gave a brief description of each to provide you with insights needed to choose the one (or several) that would fit your analytical processes the best. We focused on diversity in presenting tools that would fit technically skilled analysts such as R Studio, Python, or MySQL Workbench. On the other hand, data analysis software like [datapine](https://www.datapine.com/) cover needs both for data analysts and business users alike so we tried to cover multiple perspectives and skill levels.