# Automate the boring data science stuff

Subtitle: Focus on what you can’t automate



**Tags: EDA - import - pandas profiling - sweetviz - dtale - pandera - great expectations - modin**

**You start your learning path in data analysis or data science in python and you get used to data wrangling with pandas and data visualization with seaborn/matplotlib. You get accustomed to these defacto libraries as they simply can do almost anything.**

* **You write lots of code for basic visualizations and when you start to get sense of the data, you are just too tired to dig deeper into further insights**
* **You spend ages trying to load a large file with read\_csv.**
* **You write code from scratch for data validation or scan columns manually. Please don’t tell me you skip data validation altogether as you are just too bored. Well - we must be related :)**

**What if there are solutions somewhere?**

**What if you can automate your exploratory data analysis, and load large datasets much faster? What if you can automate your data validation steps?**

**But I can’t leave Pandas and seaborn. They are essential.**

**You won’t. You will build on them. You can spare time and effort for more advanced insights.**

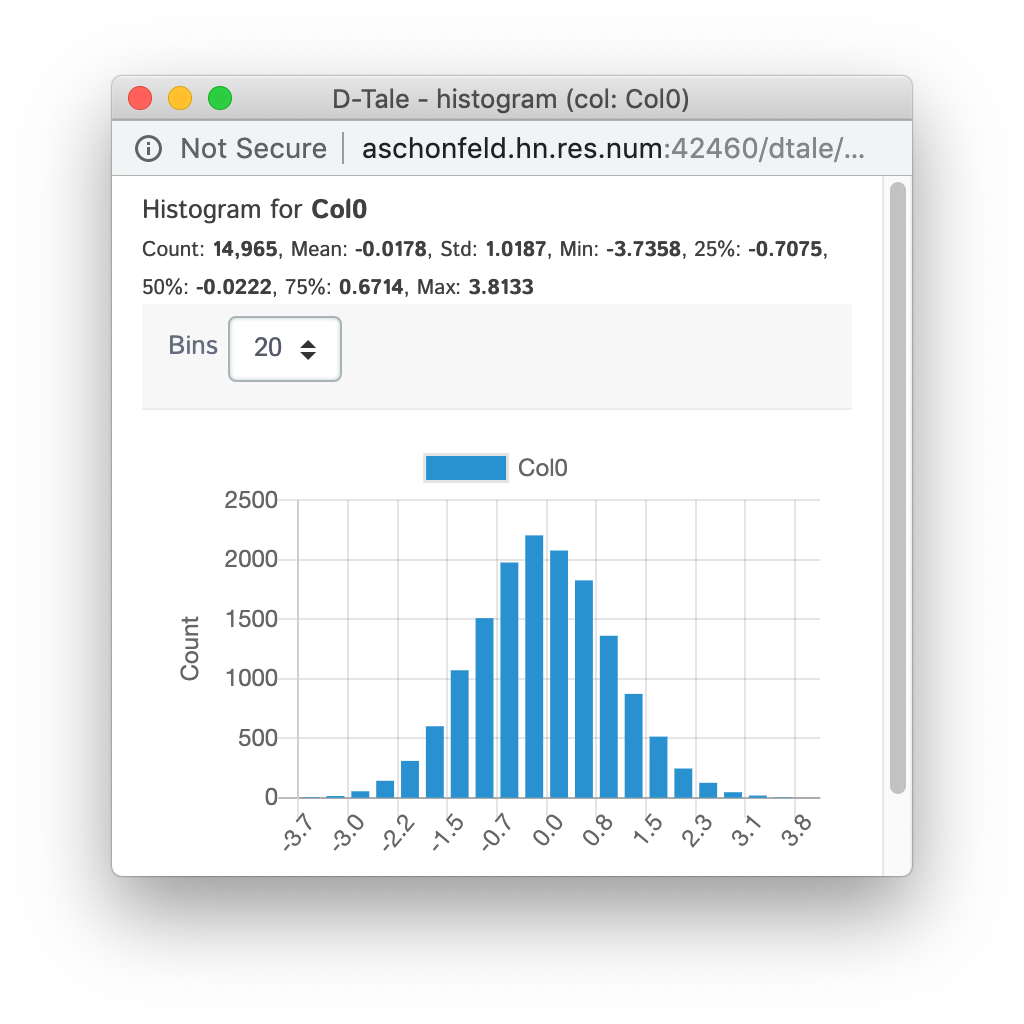
**🎯** It is not about being lazy. It's about saving time by automating the donkey work. These ***Seven*** libraries can help focus on what deserves your attention.

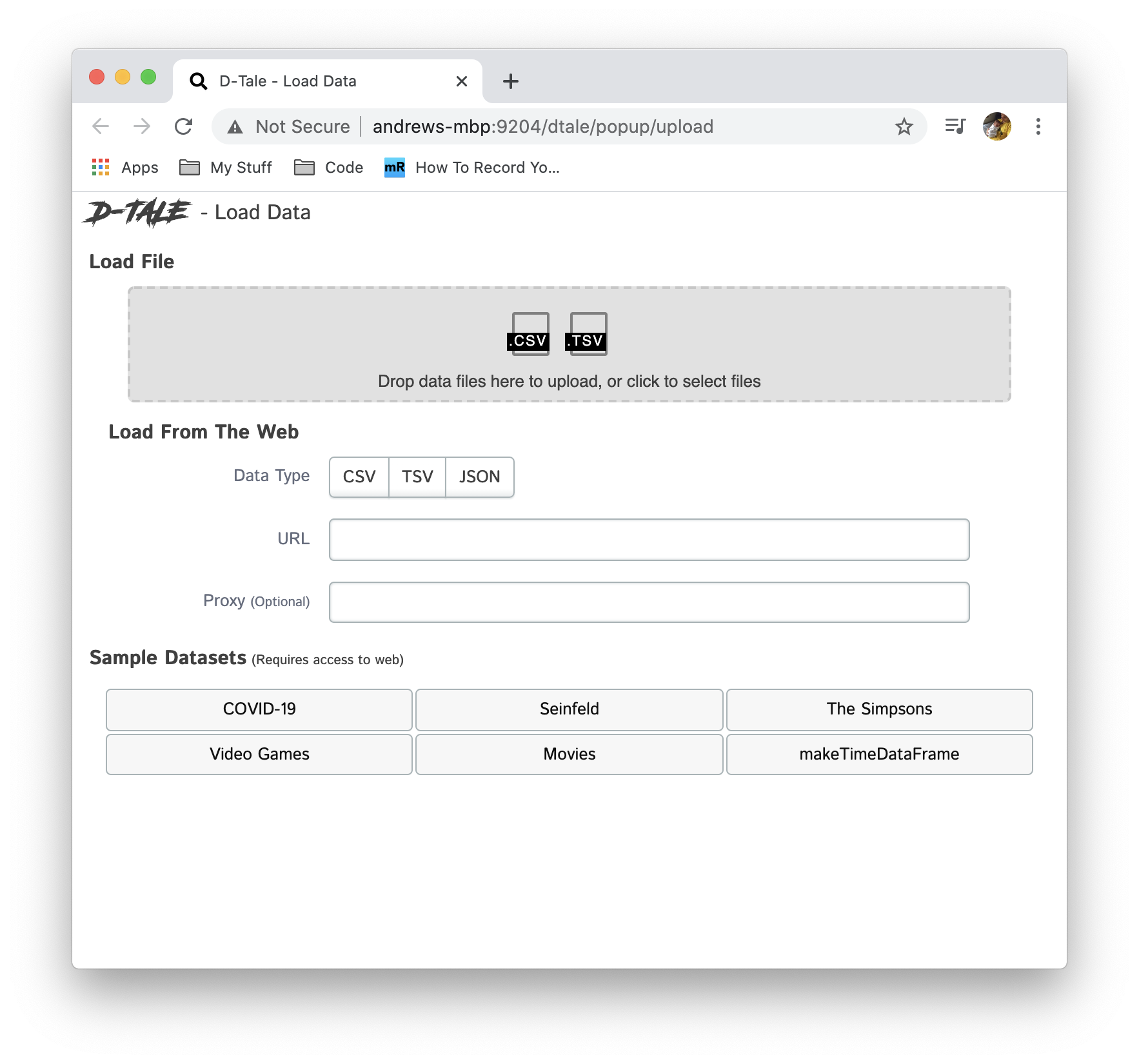
## 

[*4 Libraries that can perform EDA in one line of python code | by Satyam Kumar | Towards Data Science*](https://hemingwayapp.com/4%20Libraries%20that%20can%20perform%20EDA%20in%20one%20line%20of%20python%20code%20%7C%20by%20Satyam%20Kumar%20%7C%20Towards%20Data%20Science)

## 1- For Excel Users

The [*d-tale library*](https://pypi.org/project/dtale/)may be the perfect solution for Excel users. With a simple drag-and-drop interface, you can organize and visualize your data as if it was in an Excel sheet. Finally, you can press a button to generate the code that helped you do all this. That's a killer feature when we talk about reproducibility.





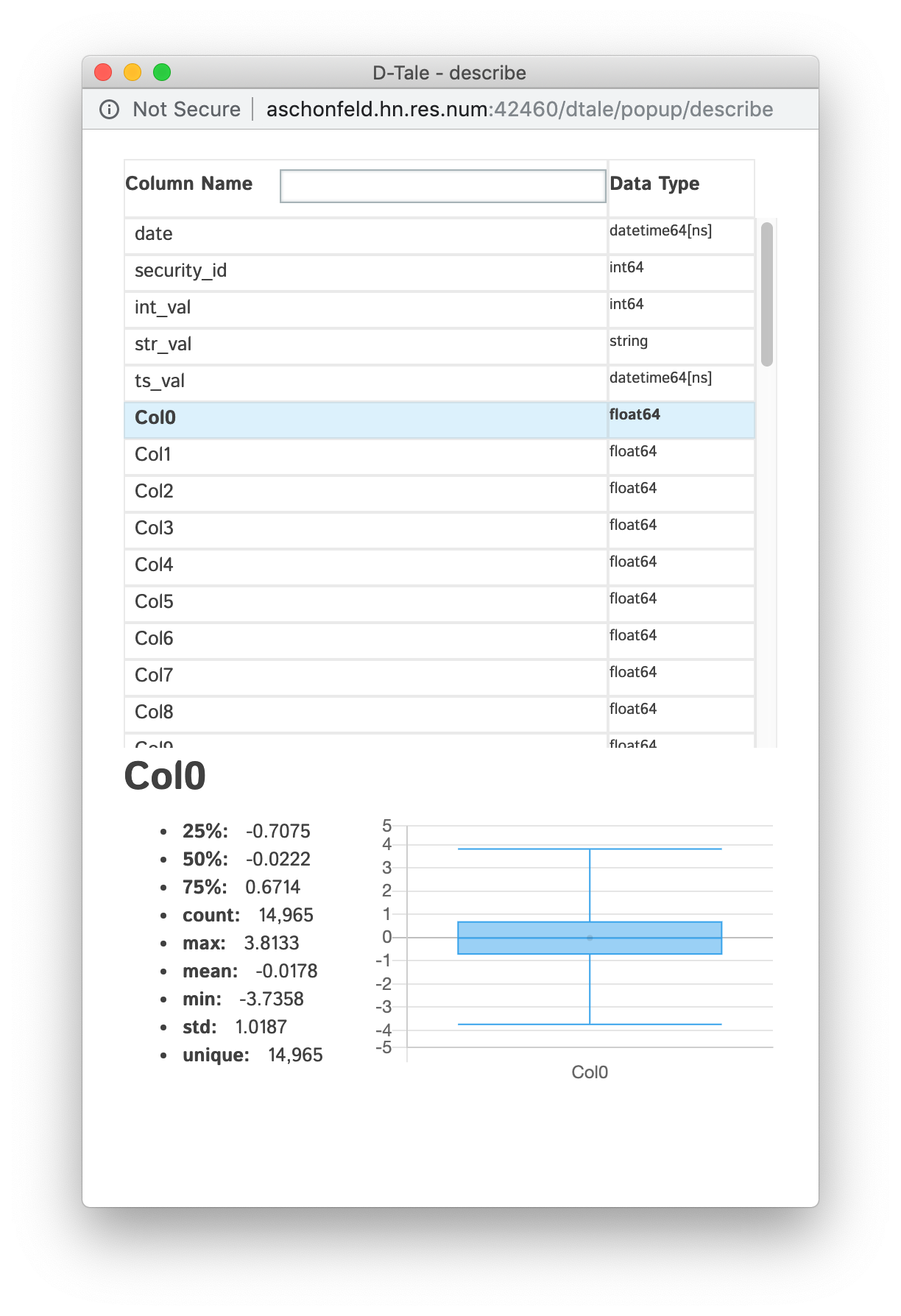


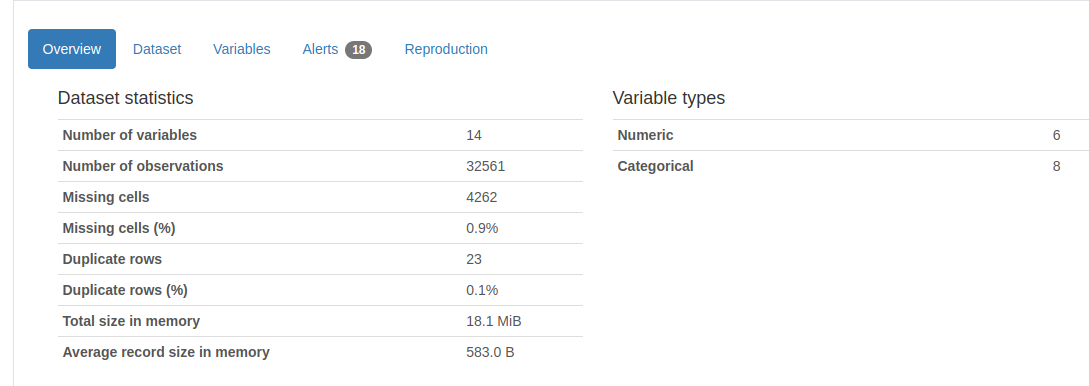
Image source <https://pypi.org/project/dtale/>

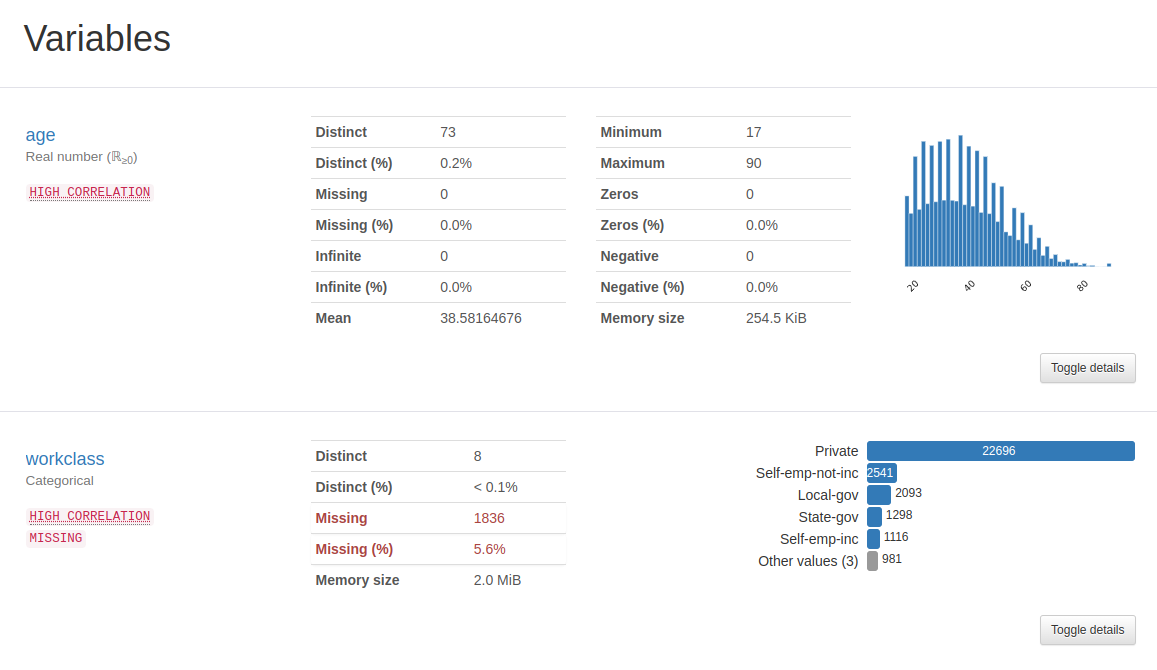
## 2- Full Report 📝

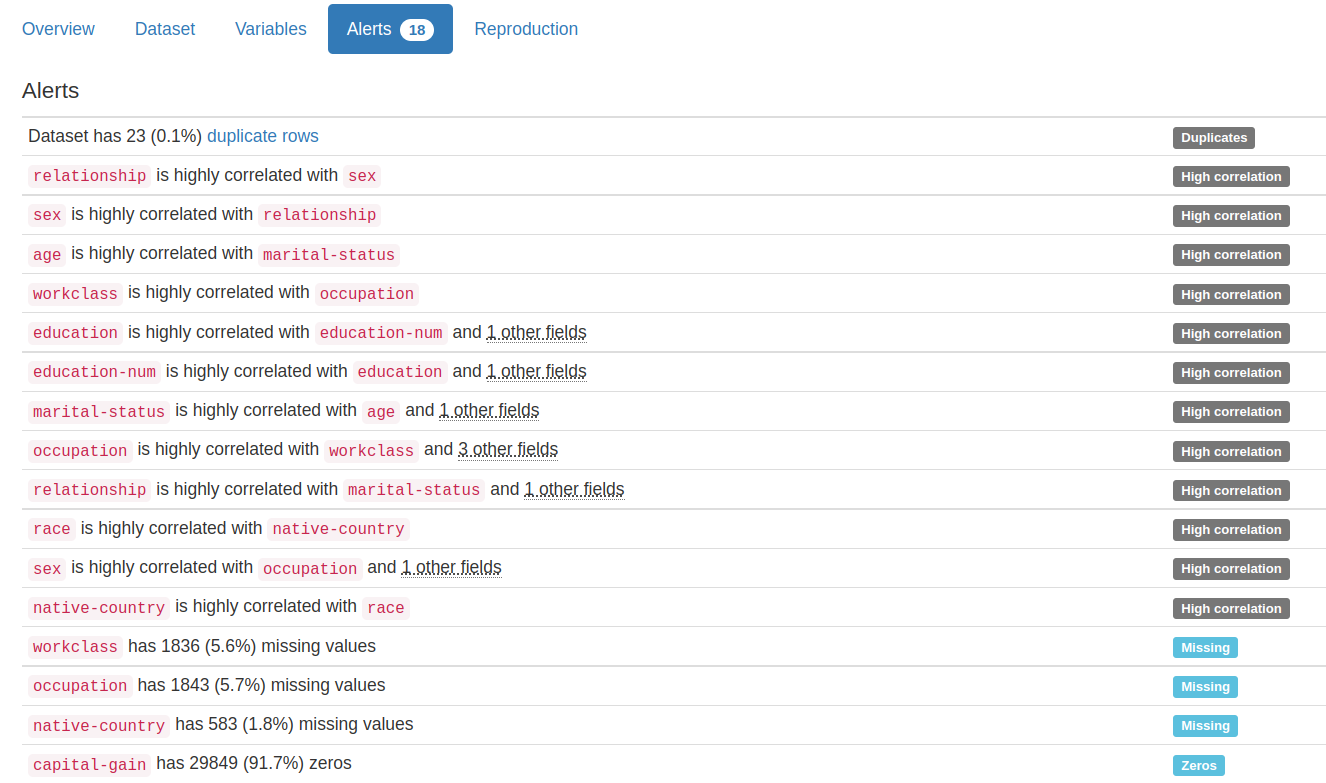
What if you require a comprehensive report with a few lines of code? A report for missing values, cardinalities, abnormal values detection, and correlations. An effortless way to confirm all your datasets in one shot. Pandas profiling may be your first stop.

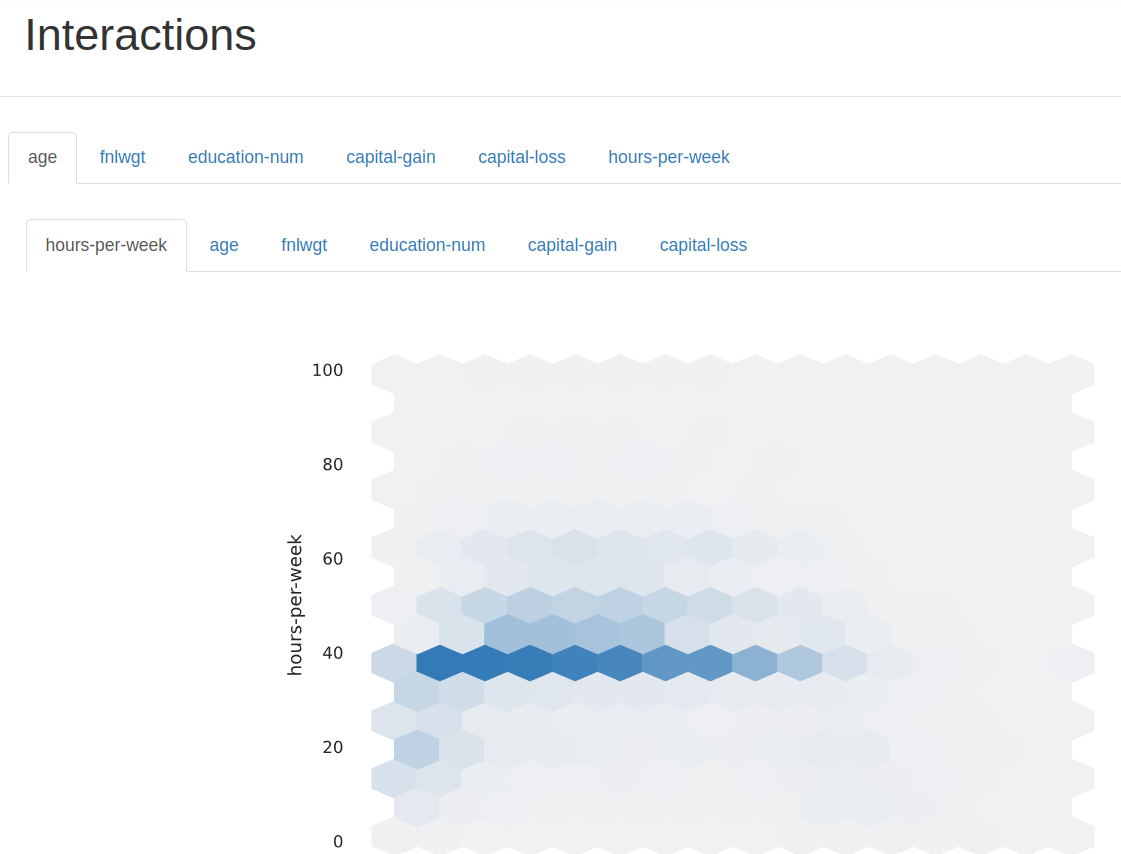


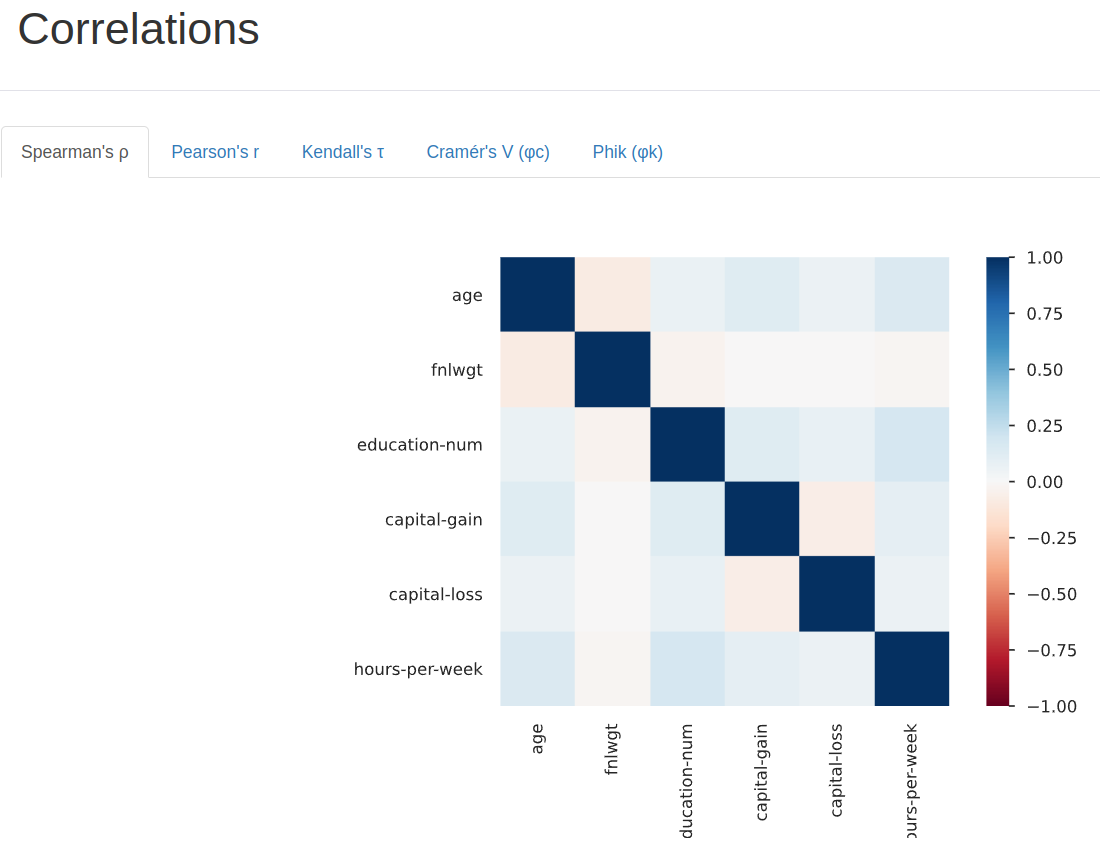


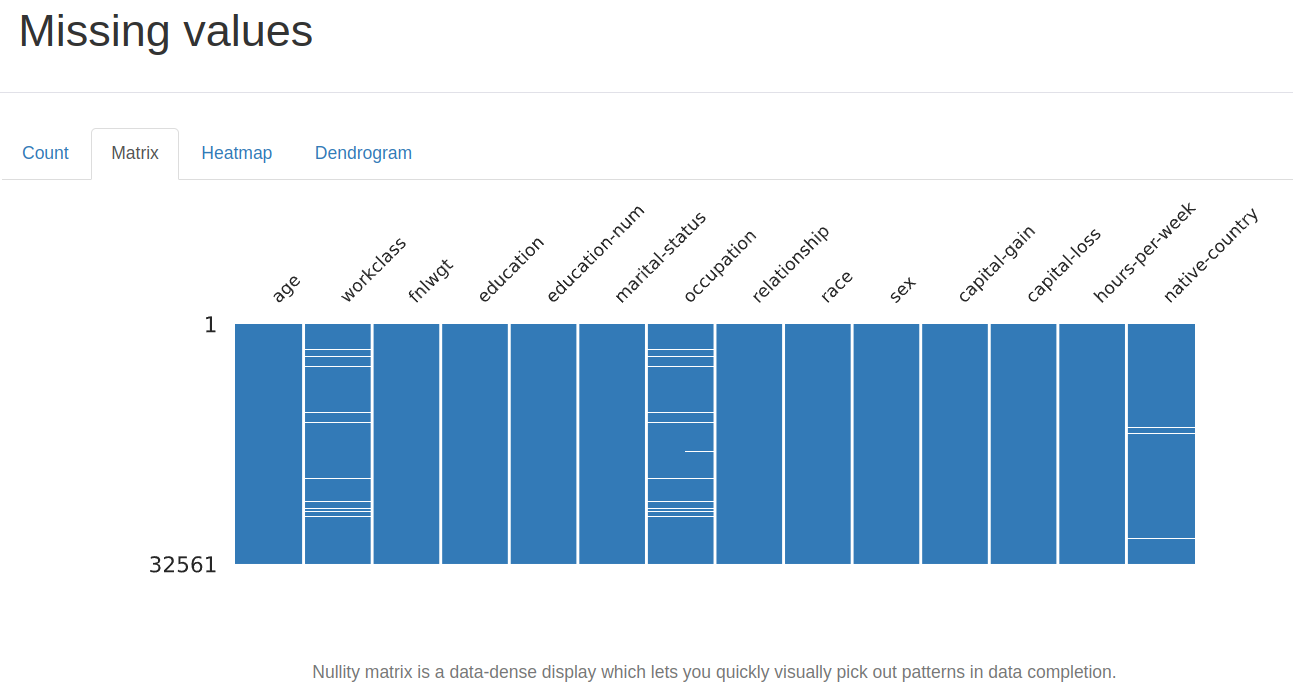












[*Exploratory Data Analysis with Pandas Profiling | by Albert Sanchez Lafuente | Towards Data Science*](https://towardsdatascience.com/exploratory-data-analysis-with-pandas-profiling-de3aae2ddff3)

## 

## 

## 

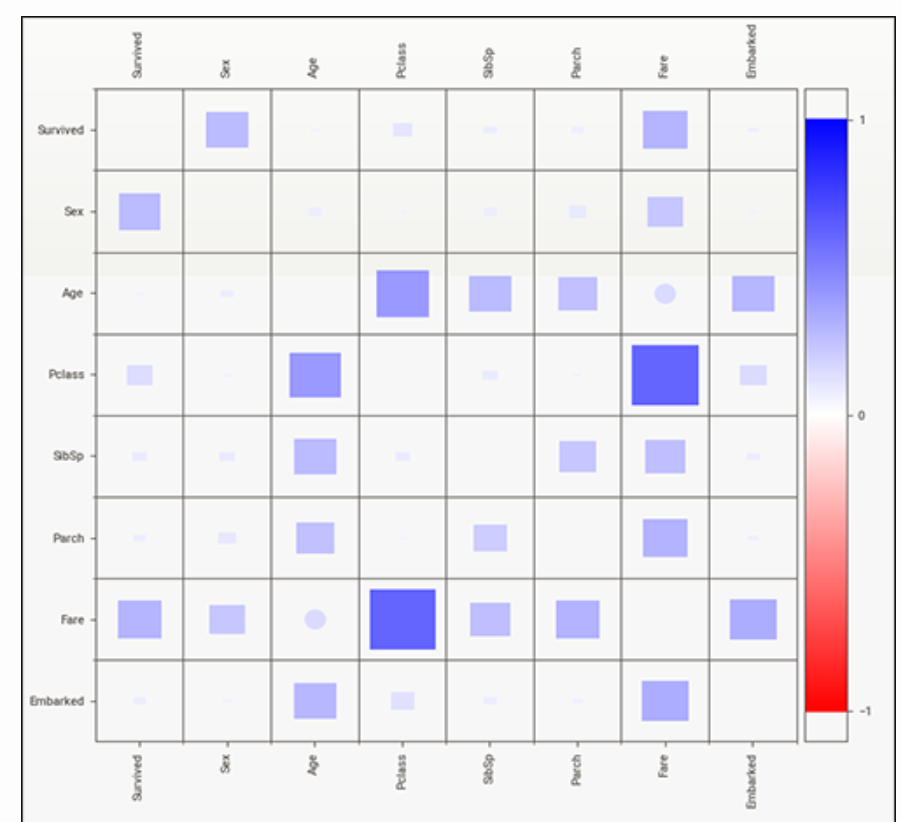
## 

## 

## 

## 3- Co-Relations and Distribution

But, what if you want to observe familiar relations of your categorical variables as well? That’s when Sweetviz comes to the rescue.



[*Powerful EDA (Exploratory Data Analysis) in just two lines of code using Sweetviz | by Francois Bertrand | Towards Data Science*](https://towardsdatascience.com/powerful-eda-exploratory-data-analysis-in-just-two-lines-of-code-using-sweetviz-6c943d32f34)

Also, the distributions of variables in your train and test sets can be radically different. Instead of the manual checks, Sweetviz can liberate you again. Over here is a notebook on Kaggle competition data that can assist you to detect these differences.



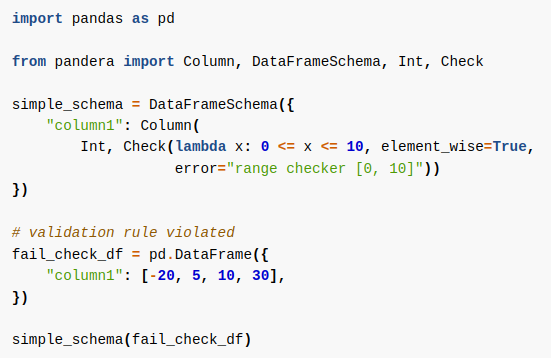
[*EDA in 2 lines of code [Sweetviz] | Kaggle*](https://www.kaggle.com/code/mahmoudhamza/eda-in-2-lines-of-code-sweetviz)

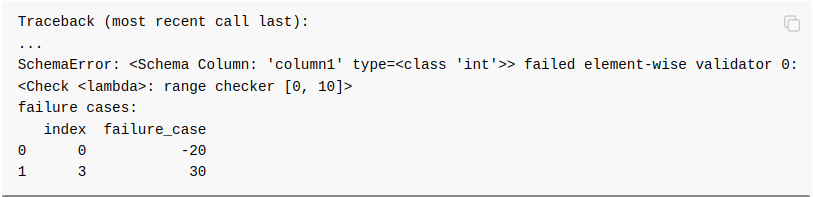
In conclusion, you can relate your variables to the outcome variable in all your visuals. Once more, Sweetviz comes to the rescue. Check [*here*](https://www.kaggle.com/code/mahmoudhamza/eda-in-2-lines-of-code-sweetviz)for an example.

## 4- Validation ✅

What if I want to confirm the dataset? For example, I am working in the medical field. Most of the time, I do maintain some expectations. For example, lab values need to be within a certain range. For diagnoses, they need to belong to a certain group; i.e., Leukemia, solid tumors, and brain cancer. All items should belong to this list. Any solution?

[*Validate Your pandas DataFrame with Pandera | by Khuyen Tran | Towards Data Science*](https://towardsdatascience.com/validate-your-pandas-dataframe-with-pandera-2995910e564)



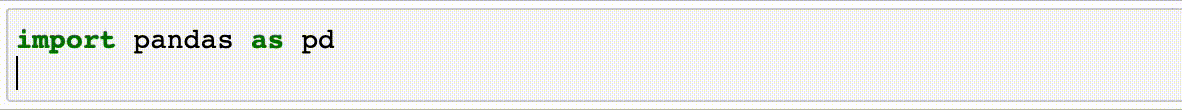


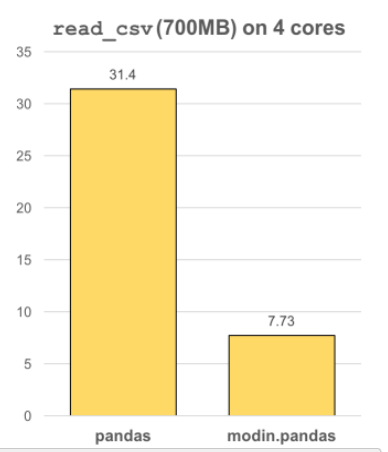
You won't solely apply checks to your DataFrame. You can indeed apply checks employing decorators. Which can preserve a few lines of code to examine the validity of inputs and outputs of functions.

*Side note: you can find many valuable tricks and libraries if you checked this GitHub page:*

[*khuyentran1401/Data-science: Collection of useful data science topics along with code and articles*](https://github.com/khuyentran1401/Data-science)

But wait for a second; we are not merely talking about pandas DataFrame. Have you ever been stuck for ages trying to import a DataFrame using pandas? What about a very significantly fast alternative? Let's all welcome Modin. 👇🏼





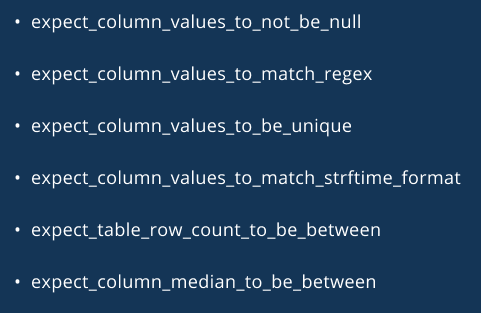
Modin uses Ray or Dask to provide an effortless way to speed up your pandas notebooks, scripts, and libraries. Unlike other distributed DataFrame libraries. Modin offers seamless integration and compatibility with existing pandas code. Even using the DataFrame constructor is identical.

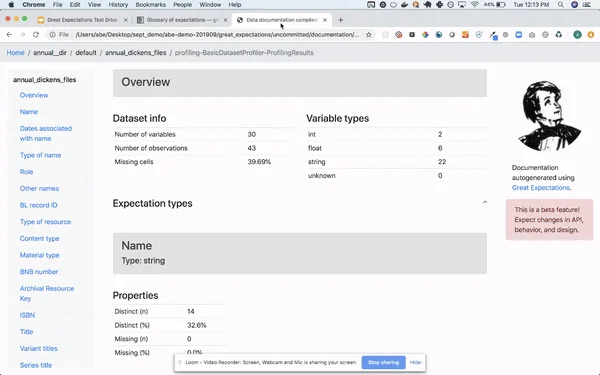
Modin is a lightweight, robust DataFrame. Because it is so lightweight, Modin provides speed-ups of up to 4x on a laptop with four physical cores.

[*Scale your pandas workflow by changing a single line of code*](https://modin.readthedocs.io/en/stable/)

What if the functionalities that are present in pandera are not enough? I lack an extensive system to confirm my data. I maintain great expectations of my data, and I need to develop such expectations.

It's time for the [*Great Expectations Library.*](https://greatexpectations.io/) 🔥





### As you can see; Batteries-included data validation