

Generate Reports Using Pandas Profiling, Deploy Using Streamlit



Kaustubh Gupta

Last Updated : 23 Oct, 2024



Pandas library offers a wide range of functions, making it an indispensable tool for data manipulation that caters to almost every task. One convenient feature, often employed for gaining quick insights into a dataset, is the pandas describe function. This function gives users a descriptive statistical summary of all the features, helping them understand the data's overall characteristics. However, for a more comprehensive analysis, the pandas profiling Package is an additional valuable tool in the Pandas ecosystem.

Pandas profiling is the solution to this problem. It offers **report generation** for the dataset with lots of features and customizations for the report generated. In this article, we will explore this library, look at all the features provided, and some advanced use cases and integrations that can be useful to create stunning reports out of the data frames!

This article was published as a part of the [Data Science Blogathon](#).

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details

Accept all cookies

Use necessary cookies

6. Integrations

- Widget in Jupyter notebook

7. How to Make it a Part of Streamlit App?

- Step 1: Install the streamlit_pandas_profiling
- Step 2: Create a Python file

8. Conclusion

Installation

Like every other Python package, pandas [profiling](#) can be easily installed via the pip package manager:

```
pip install pandas-profiling
```

Copy Code

It can also be installed via Conda package manager too:

```
conda env create -n pandas-profiling
conda activate pandas-profiling
conda install -c conda-forge pandas-profiling
```

Copy Code

Dataset and Setup

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy & Cookies Policy](#).

[Show details](#)

Before I discuss the Python pandas profiling, have a look at the **pandas describe function output** for the dataframe:

```
df.describe(include='all')
```

[Copy Code](#)

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
count	246091	246091	246091.000000	246091	246091	2.460910e+05	246091
unique	33	646	NaN	6	124	NaN	51628
top	Uttar Pradesh	BIJAPUR	NaN	Kharif	Rice	NaN	1
freq	33306	945	NaN	95951	15104	NaN	4028
mean	NaN	NaN	2005.643018	NaN	NaN	1.200282e+04	NaN
std	NaN	NaN	4.952164	NaN	NaN	5.052340e+04	NaN
min	NaN	NaN	1997.000000	NaN	NaN	4.000000e-02	NaN
25%	NaN	NaN	2002.000000	NaN	NaN	8.000000e+01	NaN
50%	NaN	NaN	2006.000000	NaN	NaN	5.820000e+02	NaN
75%	NaN	NaN	2010.000000	NaN	NaN	4.392000e+03	NaN
max	NaN	NaN	2015.000000	NaN	NaN	8.580100e+06	NaN

(Notice that I have used the `include` parameter of the `describe` function set to “all” which forces pandas to include all the data types of the dataset to be included in the summary. The string type values are accompanied by options such as `unique`, `top`, and `frequency`)

Let’s import the Python pandas profiling library:

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy](#) & [Cookies Policy](#).

[Show details](#)

not part of the package. But as soon as you import the profiling library, it adds this function to dataframe objects.


- You can pass the dataframe object to the profiling function and then call the function object created to start the generation of the profile.

You will get the same output report in either of the methods. I am using the second method to generate the report for the imported agriculture dataset.

```
profile = ProfileReport(df)
profile
```

[Copy Code](#)

```
In [*]: profile = ProfileReport(df)
profile
```

Summarize dataset: 55%  11/20 [00:12<00:03, 2.44it/s, Calculate phi_k correlation]

```
In [ ]:
```

Animation Showing report generation

Sections of the Report

Now that the report is generated, let's explore all the sections of the report one by one.

Overview

This section consists of the 3 tabs: **Overview**, **Warnings**, and **Reproduction**.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy](#) & [Cookies Policy](#).

[Show details](#)

right on data quality by revealing insights into missing cells and their corresponding percentage, offering a quick assessment of the dataset's completeness. The Duplicate rows section provides information on the presence of identical rows, including the percentage of duplicate rows. As a holistic touch, the overview concludes with the total memory size, encapsulating the overall footprint of the dataset. Integrating pandas profiling seamlessly facilitates a profound understanding of these essential aspects, enhancing the efficiency of exploratory data analysis.

Overview

Overview		Warnings 12	Reproduction
Dataset statistics		Variable types	
Number of variables	7	Categorical	4
Number of observations	246091	Numeric	3
Missing cells	3727		
Missing cells (%)	0.2%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	13.1 MiB		
Average record size in memory	56.0 B		

The **warnings** tab contains any warnings related to cardinality, correlation with other variables, missing values, zeroes, skewness of the variables, and many others.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details

Overview

Warnings 12

Reproduction

Warnings

District_Name has a high cardinality: 646 distinct values	High cardinality
Crop has a high cardinality: 124 distinct values	High cardinality
Area is highly correlated with Production	High correlation
Production is highly correlated with Area	High correlation
Area is highly correlated with Production	High correlation
Production is highly correlated with Area	High correlation
State_Name is highly correlated with Season	High correlation
Season is highly correlated with State_Name	High correlation
Production has 3727 (1.5%) missing values	Missing
Area is highly skewed ($\gamma_1 = 48.0012777$)	Skewed
Production is highly skewed ($\gamma_1 = 40.74179229$)	Skewed
Production has 3526 (1.4%) zeros	Zeros

The **reproduction** tab displays information related to the report generation. It shows the start and end times of the analysis, the time taken to generate the report, the software version of pandas profiling, and a configuration download option.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details

Overview	Warnings 6	Reproduction
Reproduction		
Analysis started	2021-06-23 16:20:59.071801	
Analysis finished	2021-06-23 16:21:26.806938	
Duration	27.74 seconds	
Software version	pandas-profiling v3.0.0	
Download configuration	config.json	

We will discuss the configuration file in this article's advanced use case section.

Variables

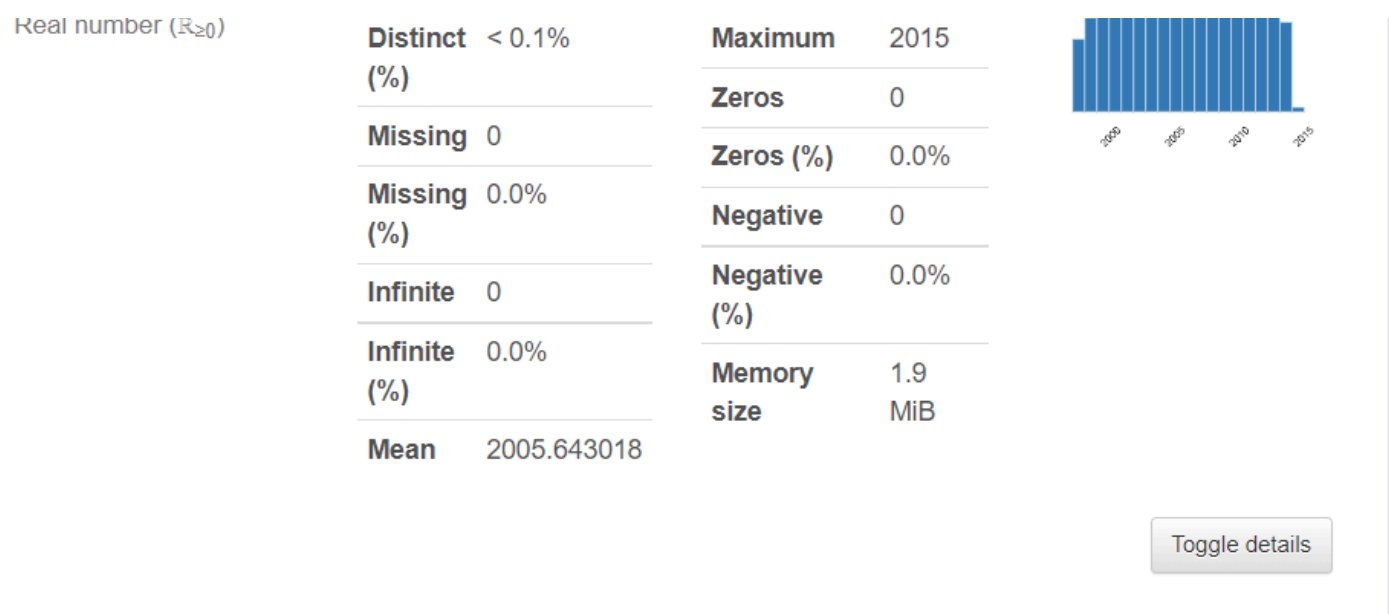
This section of the report gives a detailed [analysis](#) of all the variables/columns/features of the dataset. The information presented varies depending upon the data type of variable. Let's break it down.

Numeric Variables

You get information about the distinct values, missing values, min-max, mean, and negative values count for numeric data type features. You also get small representation values in the form of a Histogram.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy & Cookies Policy](#).

[Show details](#)



The toggle button expands to the Statistics, Histogram, Common Values, and Extreme Values tabs.

The statistics tab includes:

- 1. Quantile statistics: Min-Max, percentiles, median, range, and IQR (Inter Quartile range)
- 2. Descriptive statistics: Standard Deviation, Coefficient of variance, Kurtosis, mean, skewness, variance, and monotonicity.

Quantile statistics

Minimum	1997
5-th percentile	1998
Q1	2002
median	2006
Q3	2010
95-th percentile	2013
Maximum	2015
Range	18
Interquartile range (IQR)	8

Descriptive statistics

Standard deviation	4.952163853
Coefficient of variation (CV)	0.002469115295
Kurtosis	-1.135213503
Mean	2005.643018
Median Absolute Deviation (MAD)	4
Skewness	0.00958070166
Sum	493570696
Variance	24.52392682
Monotonicity	Not monotonic

The histogram tab displays the frequency of variables or distribution of numeric data. The common values tab is basically value_counts of the variables presented as both counts and percentage frequency.

value	Count	Frequency (%)
2003	17287	<div></div> 7.0%
2002	16671	<div></div> 6.8%
2008	14550	<div></div> 5.9%
2007	14526	<div></div> 5.9%
2006	14328	<div></div> 5.8%
2004	14117	<div></div> 5.7%
2009	14116	<div></div> 5.7%
2011	14071	<div></div> 5.7%
2010	14065	<div></div> 5.7%
2005	13799	<div></div> 5.6%
Other values (9)	98561	<div></div> 40.1%

String Variables

For string-type variables, you get Distinct (unique) values, distinct percentages, missing missing percentages, memory size, and a horizontal bar presentation of all the unique values with count presentation.

Season

Categorical

HIGH...CORRELATION

Distinct

6

Distinct (%)

< 0.1%

Missing

0

Kharif

95951

Rabi

66987

Whole Year

57305

Summer

1484

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details

The toggle button expands to the **Overview, Categories, Words, and Characters** tab.

The Overview tab displays the max-min median mean length, total characters, distinct characters, distinct categories, unique characters, and sample from the dataset for string type values.

Overview

Categories

Words

Characters

Length

Max length25

Median length7

Mean length8.515614143

Min length3

Characters and Unicode

Total characters2095616

Distinct characters52

Distinct categories7?

Distinct scripts2?

Distinct blocks1?

Unique

Unique1?

Unique (%)< 0.1%

Sample

1st rowArecanut

2nd rowOther Kharif pulses

3rd rowRice

4th rowBanana

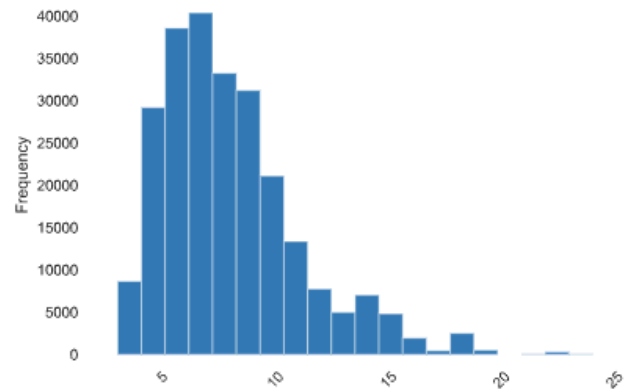
5th rowCashewnut

The categories tab displays a histogram and sometimes a pie chart of the feature's value counts. The table contains the value, count, and percentage frequency.

Common values

Value	Count	Frequency (%)
BIJAPUR	945	0.4%
TUMKUR	936	0.4%
BELGAUM	925	0.4%
HASSAN	895	0.4%
BELLARY	887	0.4%
DAVANGERE	886	0.4%
AURANGABAD	879	0.4%
HAVERI	870	0.4%
CHAMARAJAN...	844	0.3%
CHITRADURGA	840	0.3%
Other values (6...	237184	96.4%

Length



Histogram of lengths of the category

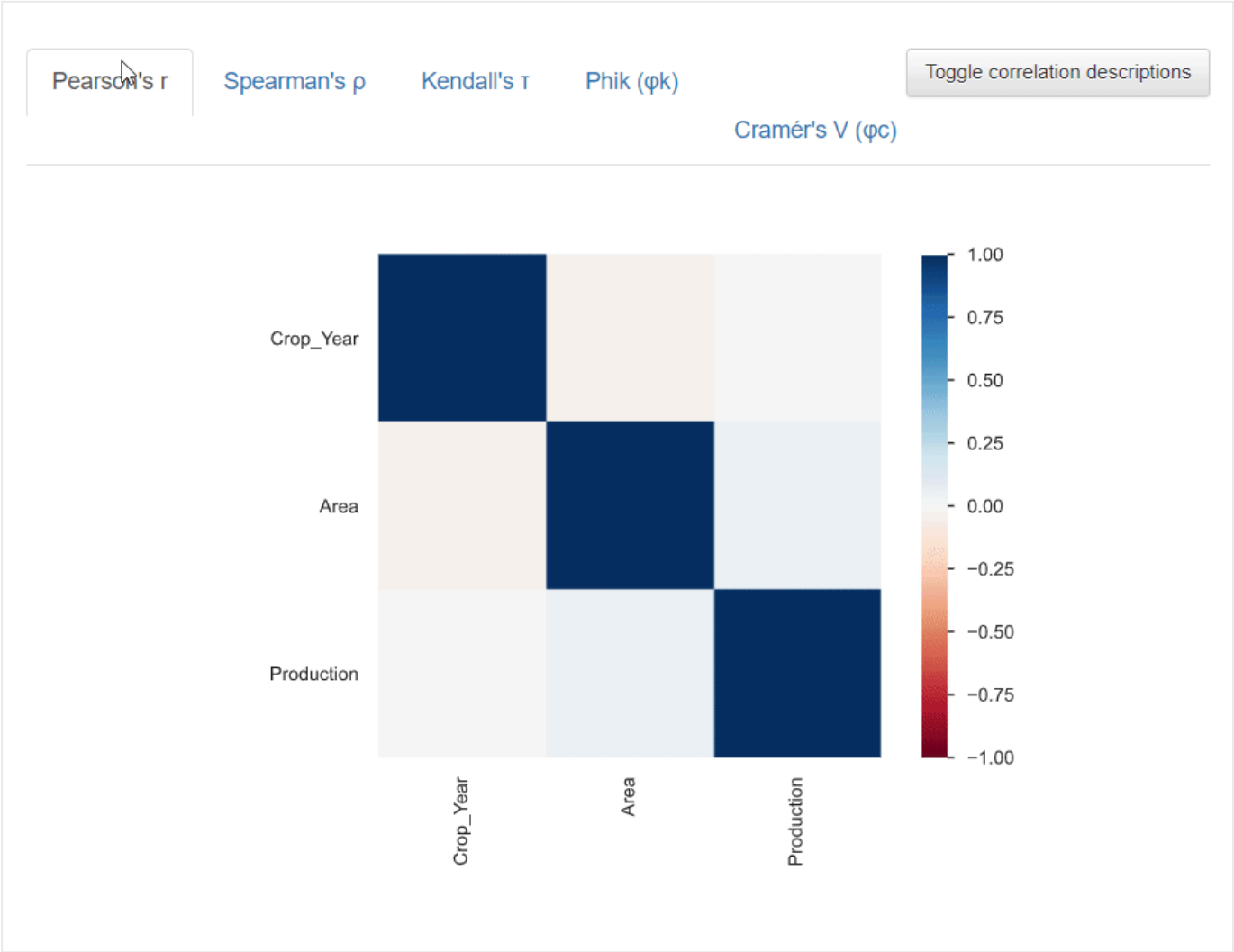
The words and the characters tab does the same job as the categories tab in terms of presenting the data in tabular and histogram format. Still, it can go much deeper into the lower case, upper case, punctuation, special characters categories count too!

Correlations

Correlation describes the degree to which two variables move in coordination with one another. The pandas profiling python report provides five types of correlation coefficients: Pearson's r , Spearman's ρ , Kendall's τ , Phik (ϕ_k), and Cramér's V (ϕ_c).

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details



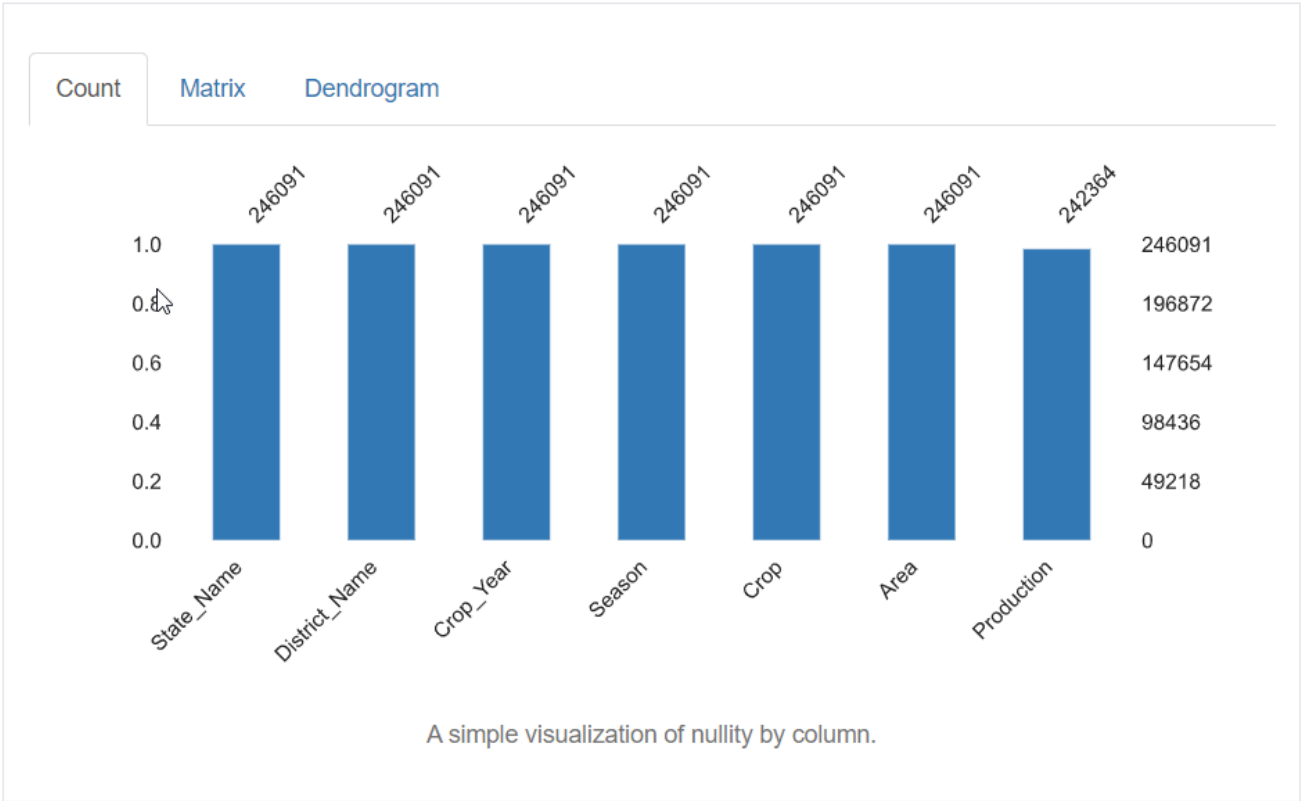
You can also click the toggle button for details about the correlation coefficients.

Missing values

The report generated also contains visualizations for the missing values in the dataset. You get three types of plots: count, matrix, and dendrogram. The count plot is a basic bar plot with an x-axis as column names, and the length of the bar represents the

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy & Cookies Policy](#).

[Show details](#)



Sample

This section displays the first and last 10 rows of the dataset.

How to Save the Report?

So far, you’ve learned how to generate dataframe reports with a single line of code or function and explored the report’s included features. You may want to export this analysis to an external file for integration with other applications or web publishing

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details

While saving. To save the report, call the `to_file()` function on the profile object.

```
profile.to_file("Analysis.html")
profile.to_file("Analysis.json")
```

[Copy Code](#)

```
profile.to_file("Analysis.html")
```

Export report to file: 100%  1/1 [00:00<00:00, 27.03it/s]

```
profile.to_file("Analysis.json")
```

Render JSON: 100%  1/1 [00:04<00:00, 4.78s/it]

Export report to file: 100%  1/1 [00:00<00:00, 1.56it/s]

Advanced Usage

The report generated by Pandas profiling Python is a complete analysis without any input from the user except the dataframe object. All the report elements are chosen automatically, and default values are preferred.

There might be some elements in the report that you don't want to include, or you need to add your metadata for the final report. There comes the advanced usage of this library. You can control every aspect of your report by changing the default configurations.

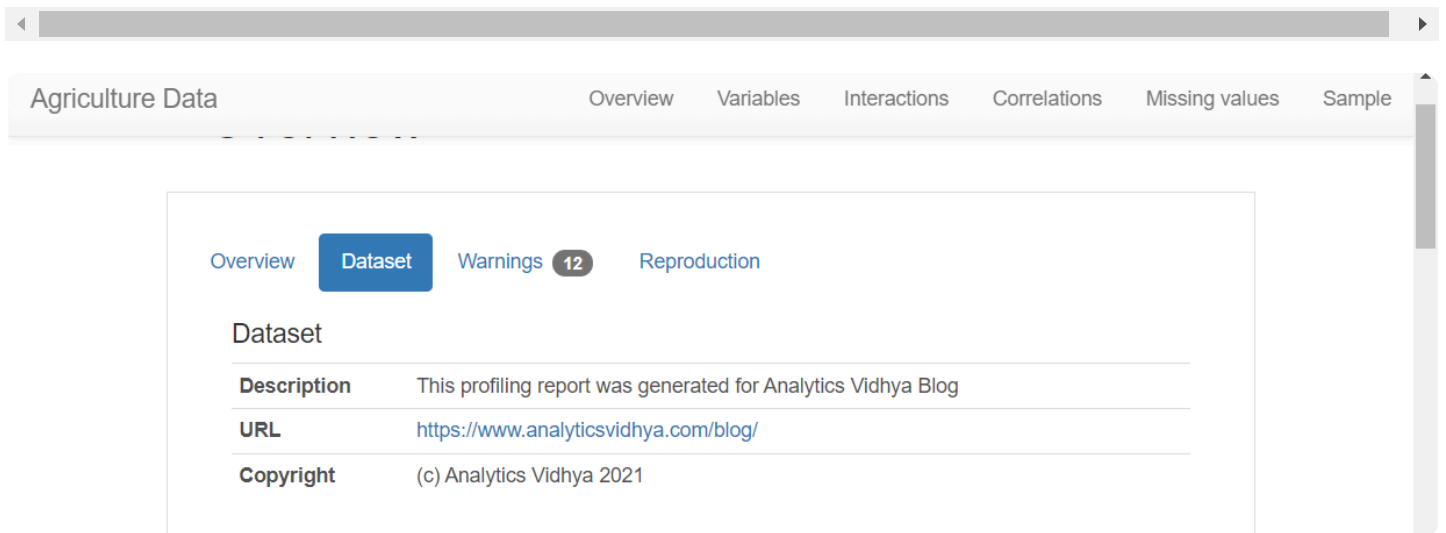
Let's see some of how you can customize your reports.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy & Cookies Policy](#).

[Show details](#)

"copyright_year", and "copyright_holder". This information will appear in the dataset overview section. A new tab called "dataset" will be created for this metadata. To add this data to the report, use the dataset parameter in the ProfileReport function and pass this data as a dictionary:

```
profile = ProfileReport(df,
                        title="Agriculture Data",
                        dataset={
                            "description": "This profiling report was generated for Analytics Vidhya Blog",
                            "copyright_holder": "Analytics Vidhya",
                            "copyright_year": "2021",
                            "url": "https://www.analyticsvidhya.com/blog/",
                        },)
profile
```

[Copy Code](#)


The screenshot shows the 'Agriculture Data' report interface. The top navigation bar includes 'Overview', 'Variables', 'Interactions', 'Correlations', 'Missing values', and 'Sample'. Below this, the 'Dataset' tab is selected, showing a table with the following information:

Dataset	
Description	This profiling report was generated for Analytics Vidhya Blog
URL	https://www.analyticsvidhya.com/blog/
Copyright	(c) Analytics Vidhya 2021

You can also add information about the variables used in the [dataset](#) using the variables parameter. This takes in the dictionary with descriptions as the key and value as another dictionary with a key-value pair, where the key is the variable name and the value is the description of the variable.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy & Cookies Policy](#).

[Show details](#)


```
    "Production": "How much production?",  
  }  
}
```

When you add this to your ProfileReport function, a separate tab will be created named “Variables” under the overview section:

[Overview](#) [Dataset](#) [Variables](#) [Warnings](#) **12** [Reproduction](#)

Variable descriptions

State_Name	Name of the state
District_Name	Name of district
Crop_Year	Year when it was seeded
Season	Crop year
Crop	Which crop was seeded?
Area	How much area was allocated to the crop?
Production	How much production?

Controlling parameters of the Report

Suppose you don’t want to display all types of correlation coefficients. You can disable other coefficients by using the configuration for correlations. This is also a dictionary object and can be passed to the ProfileReport function:

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy & Cookies Policy](#).

[Show details](#)

Similarly, you can customize every report section, the HTML format, plots, and everything.



Check out [this page of the documentation](#) for details.

Integrations

After making your reports stunning by configuring every aspect of it, you might want to publish it anyhow. You can export it to HTML format and upload it to the web. But there are some other methods to make your report stand out.

Widget in Jupyter notebook

While running the panda profiling in your Jupyter notebooks, you will get the HTML rendered in the code cell only. This disturbs the experience of the user. You can make it act like a widget that is easily accessible and offers a compact view. To do this, simply call “**.to_widgets()**” on your profile object:

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details

Overview	Dataset	Variables	Warnings (12)	Reproduction
Number of variables	7		Categorical	4
Number of observations	246091		Numeric	3
Missing cells	3727			
Missing cells (%)	0.2%			
Duplicate rows	0			
Duplicate rows (%)	0.0%			
Total size in memory	13.1 MiB			
Average record size in memory	56.0 B			

Report generated with [pandas-profiling](#).

How to Make it a Part of Streamlit App?

Yes! You can make this report as a part of a streamlit app, too. Streamlit is a powerful package that enables GUI web app building with minimal code. The applications are interactive and compatible with almost every device. You can make your reports as a part of the streamlit app by following this code:

Step 1: Install the streamlit_pandas_profiling

```
pip install streamlit-pandas-profiling
```

[Copy Code](#)

Step 2: Create a Python file

Create a python file and write code in this format:

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy](#) & [Cookies Policy](#).

[Show details](#)

```
df = pd.read_csv("crops_data.csv", na_values=['='])

profile = ProfileReport(df,

                        title="Agriculture Data",

                        dataset={

                            "description": "This profiling report was generated for Analytics Vidhya Blog",

                            "copyright_holder": "Analytics Vidhya",

                            "copyright_year": "2021",

                            "url": "https://www.analyticsvidhya.com/blog/",

                        },

                        variables={

                            "descriptions": {

                                "State_Name": "Name of the state",

                                "District_Name": "Name of district",

                                "Crop_Year": "Year when it was seeded",

                                "Season": "Crop year",
```

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy](#) & [Cookies Policy](#).

Show details

```
    }

}

)

st.title("Pandas Profiling in Streamlit!")

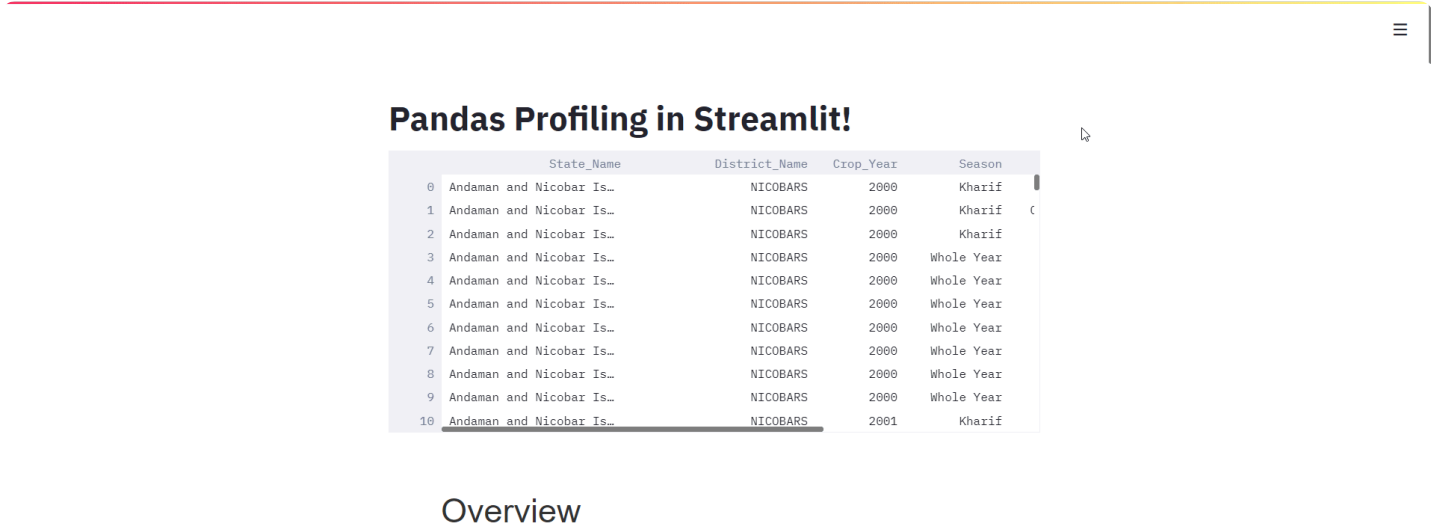
st.write(df)

st_profile_report(profile)
```

Step 3: Run your streamlit app

In the terminal, type:

```
streamlit run .py
```



Overview

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details

Exploratory Data Analysis (EDA) is like exploring a new place. You start by looking around to understand what's there. Similarly, in EDA, you look at a [dataset](#) to see what's in it and what it can tell you. It's essentially the initial data exploration stage in data science, where you delve into the dataset statistics and examine its intricacies.

Here's what you do during EDA:

- **Look at the Numbers:** You start by checking basic things like averages, ranges, and the spread of the numbers.
- **Make Pictures:** Instead of just staring at numbers, you make charts and graphs to show the data visually. It's like drawing a map of your exploration.
- **Clean Up:** Sometimes, data can be messy with missing pieces or weird values. So, you clean it up by filling in missing parts or removing the weird stuff.
- **Create New Ideas:** You might develop new ideas or ways to look at the data, like combining different parts or changing how you measure things.
- **Find Connections:** You try to see if different parts of the data are related. For example, if one thing goes up, does another also go up?
- **Make Things Simple:** If the data is too complicated, you might simplify it to see the big picture more clearly.
- **Look at Time:** If your data changes over time, you'll examine how it changes and whether there are any repeating patterns.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy & Cookies Policy](#).

[Show details](#)

making big considerations. It's like exploring a map before going on a big adventure.

Conclusion

In this article, you are introduced to a new tool, "Pandas Profiling," a one-stop solution for generating reports out of the Pandas dataframe. We explore all the features of this tool, different sections, and their content. Then, we move on to saving the report generated. Later, we look at some of the advanced use cases of this library and finally integrate the Streamlit app to make the reports more promising and interactive.

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.



[Kaustubh Gupta](#)

Kaustubh Gupta is a skilled engineer with a B.Tech in Information Technology from Maharaja Agrasen Institute of Technology. With experience as a CS Analyst and Analyst Intern at Prodigal Technologies, Kaustubh excels in Python, SQL, Libraries, and various engineering tools. He has developed core components of product intent engines, created gold tables in Databricks, and built internal tools and dashboards using Streamlit and Tableau. Recognized as India's Top 5 Community Contributor 2023 by Analytics Vidhya, Kaustubh is also a prolific writer and mentor, contributing significantly to the tech community through speaking sessions and workshops.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy & Cookies Policy](#).

[Show details](#)

Listicle

Machine Learning

Project

Python

Python

Streamlit

Structured Data

Technique

Free Courses



Generative AI - A Way of Life

Explore Generative AI for beginners: create text and images, use top AI tools, learn practical skills, and ethics.



Getting Started with Large Language Models

Master Large Language Models (LLMs) with this course, offering clear guidance in NLP and model training made simple.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy](#) & [Cookies Policy](#).

[Show details](#)

chatbots with enterprise data.



Improving Real World RAG Systems: Key Challenges & Practical Solutions

Explore practical solutions, advanced retrieval strategies, and agentic RAG systems to improve context, relevance, and accuracy in AI-driven applications.



Microsoft Excel: Formulas & Functions

Master MS Excel for data analysis with key formulas, functions, and LookUp tools in this comprehensive course.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy](#) & [Cookies Policy](#).

[Show details](#)

Responses From Readers

What are your thoughts?...

Submit reply



Santosh Kesava

Hi , This is really a informative post thank you for posting.
My scenario is same but the only missing part is how to

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy](#) & [Cookies Policy](#).

Show details

Nice article. I'm trying to run pandas profiling in my python 3.8.x version. But I'm getting error message saying PydanticImportError: BaseSettings has been moved to thepydantic-settings package while I'm running from pandas_profiling import ProfileReport. Can you please guide me to resolve the issue?



We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy & Cookies Policy](#).

[Show details](#)

• ✓

How to use pandas-profiling?

A. To use pandas-profiling, you should first install it using pip. Then, import it into your Python script or Jupyter Notebook. Load your dataset with Pandas, create a ProfileReport object, and call its `to_file()` or `to_widgets()` methods to obtain a detailed analysis and visualization of your data.

What is Pandas profiling?

Why use pandas profiling?

Q4. How to pip install pandas-profiling?

Write for us →

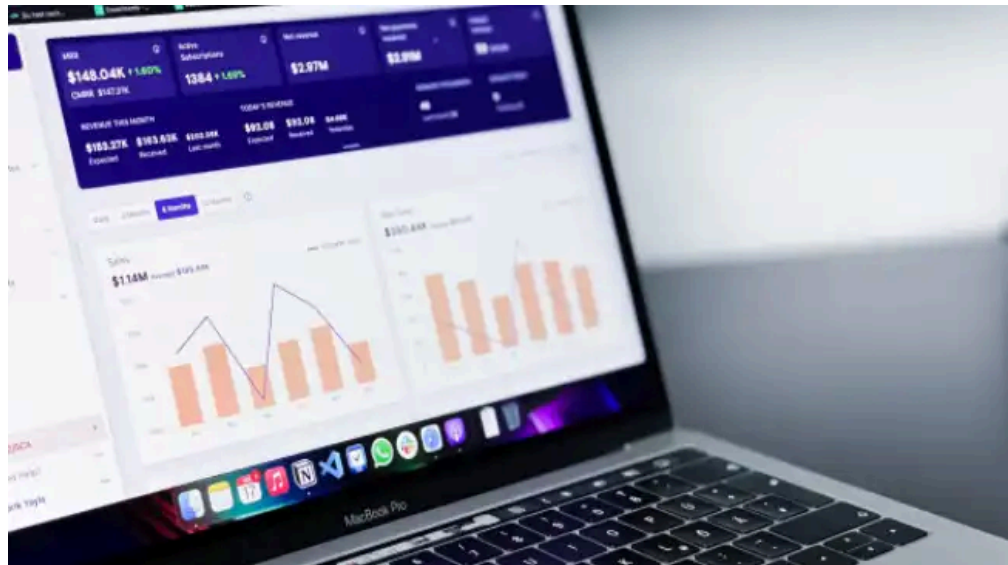
Write, captivate, and earn accolades and rewards for your work

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details

Personalized GenAI Learning Path 2025 ✨ Crafted Just for YOU!

- Level Up Your Data Science Game



Flagship Courses

GenAI Pinnacle Program | AI/ML BlackBelt Courses

Free Courses

Generative AI | Large Language Models | Building LLM Applications using Prompt Engineering | Building Your first RAG System using LlamaIndex | Stability.AI | MidJourney | Building Production Ready RAG systems using LlamaIndex | Building LLMs for Code | Deep Learning | Python |

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy](#) & [Cookies Policy](#).

[Show details](#)

Generative AI Tools and Techniques

GANs | VAEs | Transformers | StyleGAN | Pix2Pix | Autoencoders | GPT | BERT | Word2Vec | LSTM | Attention Mechanisms | Diffusion Models | LLMs | SLMs | StyleGAN | Encoder Decoder Models | Prompt Engineering | LangChain | LlamaIndex | RAG | Fine-tuning | LangChain AI Agent | Multimodal Models | RNNs | DCGAN | ProGAN | Text-to-Image Models | DDPM | Document Question Answering | Imagen | T5 (Text-to-Text Transfer Transformer) | Seq2seq Models | WaveNet | Attention Is All You Need (Transformer Architecture)

Popular GenAI Models

Llama 3.1 | Llama 3 | Llama 2 | GPT 4o Mini | GPT 4o | GPT 3 | Claude 3 Haiku | Claude 3.5 Sonnet | Phi 3.5 | Phi 3 | Mistral Large 2 | Mistral NeMo | Mistral-7b | Gemini 1.5 Pro | Gemini Flash 1.5 | Bedrock | Vertex AI | DALL.E | Midjourney | Stable Diffusion

Data Science Tools and Techniques

Python | R | SQL | Jupyter Notebooks | TensorFlow | Scikit-learn | PyTorch | Tableau | Apache Spark | Matplotlib | Seaborn | Pandas | Hadoop | Docker | Git | Keras | Apache Kafka | AWS | NLP | Random Forest | Computer Vision | Data Visualization | Data Exploration | Big Data | Common Machine Learning Algorithms | Machine Learning

Company

Discover

About Us

Blogs

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our Privacy Policy & Cookies Policy.

Show details

GenAI Program

Events

Agentic AI Program

Podcasts

Contribute

Enterprise

Become an Author

Our Offerings

Become a Speaker

Trainings

Become a Mentor

Data Culture

Become an Instructor

AI Newsletter

[Terms & conditions](#) • [Refund Policy](#) • [Privacy Policy](#) • [Cookies Policy](#) © Analytics
Vidhya 2025.All rights reserved.

We use cookies essential for this site to function well. Please click to help us improve its usefulness with additional cookies. Learn about our use of cookies in our [Privacy Policy](#) & [Cookies Policy](#).

[Show details](#)