Exploratory Data Analysis on Autopilot: Python's Automatic Solutions

I.V. Dwaraka Srihith[1], A. David Donald[2], T. Aditya Sai Srinivas[3], G. Thippanna[4], P. Vijaya Lakshmi[5]

[1]Student, Alliance University, Bangalore

[2]Assistant Professor, [3]Associate Professor, [4]Professor, [5]Student, Ashoka Women's Engineering College, Kurnool

****Corresponding Author***

E-mail Id: - *dwarakanani525@gmail.com*

ABSTRACT

Python has gained immense popularity in the fields of data science and machine learning due to its extensive libraries and efficient coding capabilities, enabling time-saving solutions. This article presents a comprehensive tutorial on Automatic Exploratory Data Analysis (EDA) using Python. By leveraging Python libraries, we can swiftly extract valuable insights and statistical information from datasets, reducing the manual effort involved in data exploration. The article aims to equip readers with the knowledge and tools to efficiently analyze data, revealing hidden patterns and trends, all accomplished through just a few lines of code. By the end of this article, readers will have a clear understanding of how Python's automated EDA techniques can revolutionize the data analysis process, maximizing efficiency and productivity.

Keywords: *Python libraries, time-saving, data science, machine learning, Automatic EDA, information, statistics, data, code.*

INTRODUCTION

Python has emerged as a popular programming language for data science and machine learning, primarily due to its extensive libraries that enable efficient and time-saving solutions. In this article, we will delve into the realm of Automatic Exploratory Data Analysis (EDA) using Python. EDA plays a crucial role in understanding datasets, extracting valuable insights, and making informed decisions.

By harnessing the power of Python libraries, we can streamline the process of data exploration and uncover key patterns and statistical information with minimal effort. This article aims to provide readers with a comprehensive understanding of how Python's automated EDA techniques can revolutionize the data analysis process, allowing us to extract valuable knowledge from our data efficiently.

RELATED WORK

"AutoViz: Automatic Visualization of Data" by Ram Seshadri et al. This paper introduces AutoViz, a Python library that automates the visualization process for EDA by selecting appropriate visualizations based on the data type.

"Pandas Profiling: Automatic Exploratory Data Analysis" by Jos Polfliet et al. Pandas Profiling is a Python library that generates detailed EDA reports with descriptive statistics, visualizations, and correlations, enabling quick data insights.

"AutoEDA: Automatic Exploratory Data Analysis with a Focus on Statistical Testing" by Alexander Watson. AutoEDA

is a Python package that combines statistical testing with automated EDA, allowing users to explore data distributions and relationships while performing hypothesis testing.

"Sweetviz: Automated EDA and Visualization" by Francois Bertrand. Sweetviz is a Python library that generates comprehensive EDA reports, including summary statistics, visualizations, and comparisons between target variables, making it easy to gain insights from the data.

"DataPrep: A Python Library for Data Preprocessing" by SfuStatistics/DataPrep. The DataPrep library offers various automated data preprocessing and EDA functions, simplifying the tasks of cleaning and exploring datasets.

AUTOMATIC EDA

Automatic EDA, also known as Automatic Exploratory Data Analysis, serves as the initial and vital step in the data science journey for a Data Scientist. It involves analyzing and comprehending various aspects of the data, including identifying missing values and outliers. Python offers a range of powerful libraries such as Pandas, Matplotlib, Seaborn, and Plotly, which are commonly employed by machine learning practitioners for conducting exploratory data analysis.

In addition to these well-known libraries, there exists another valuable library called "dataprep" that can be utilized for EDA purposes. This library facilitates the exploration of data by providing essential statistics and information through interactive visualizations and summary statistics. By leveraging dataprep, Data Scientists gain a comprehensive understanding of the data, allowing them to extract meaningful insights efficiently.

AUTOMATIC EDA USING PYTHON

In this section, we will embark on a Data Science tutorial that explores the realm of Automatic Exploratory Data Analysis (EDA) using Python. Specifically, we will employ the powerful dataprep library in Python to accomplish our EDA tasks. If you haven't utilized this library previously, fret not, as it can be easily installed by executing the pip command: pip install dataprep.

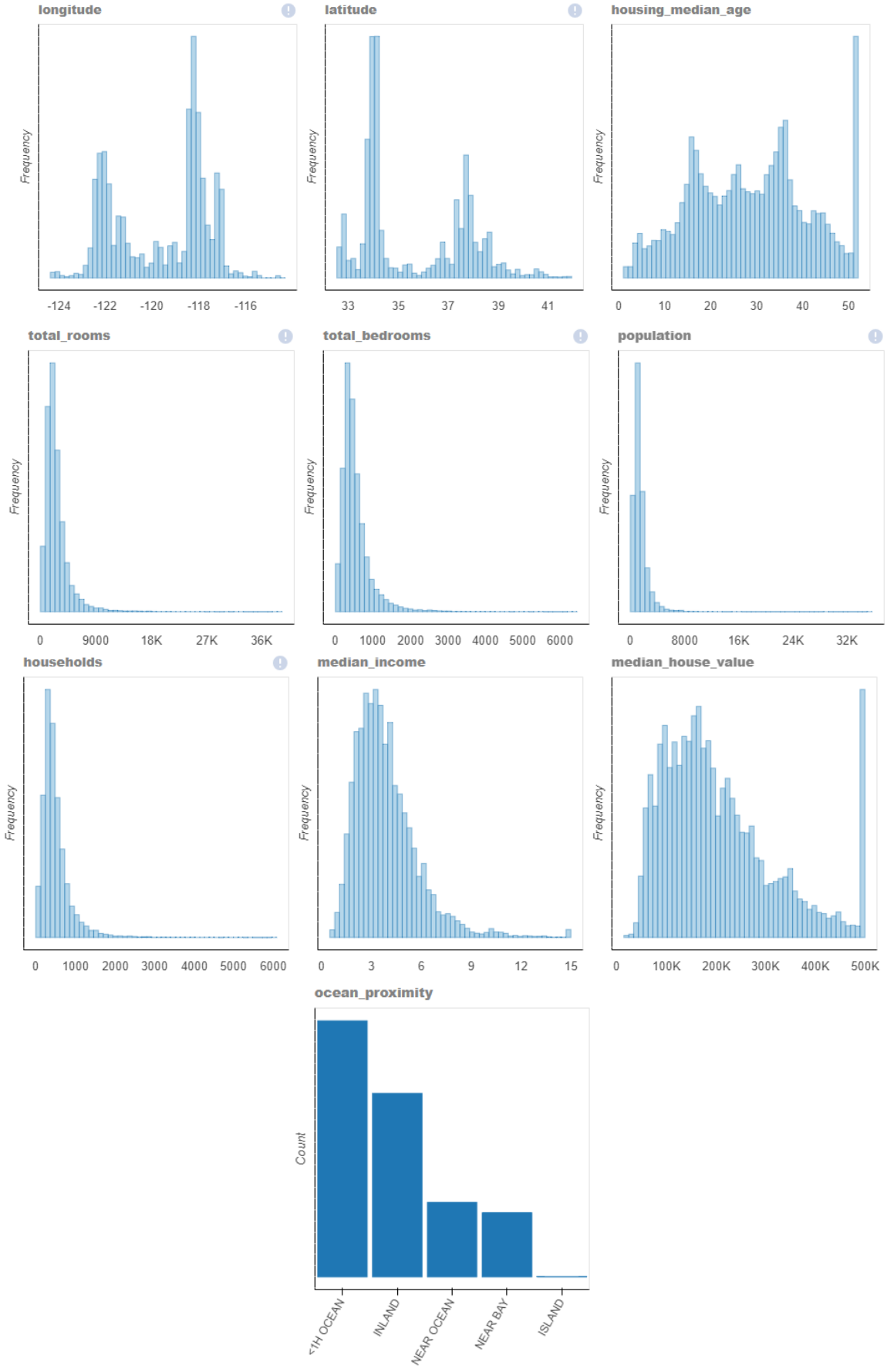
Without further ado, let's dive into the tutorial by importing the essential Python libraries and loading the dataset, laying the groundwork for our Automatic EDA journey.

```
from dataprep.eda import plot, plot_correlation, create_report, plot_missing
from google.colab import files
uploaded = files.upload()
import pandas as pd
data = pd.read_csv('housing.csv')
```

```
data.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

```
plot(data)
```


longitude
Frequency
-124
-122
-120
-118
-116
latitude
Frequency
33
35
37
39
41
housing_median_age
Frequency
0
10
20
30
40
50
total_rooms
Frequency
0
9000
18K
27K
36K
total_bedrooms
Frequency
0
1000
2000
3000
4000
5000
6000
population
Frequency
0
8000
16K
24K
32K
households
Frequency
0
1000
2000
3000
4000
5000
6000
median_income
Frequency
0
3
6
9
12
15
median_house_value
Frequency
0
100K
200K
300K
400K
500K
ocean_proximity
Count
INLAND
NEAR OCEAN
NEAR BAY
ISLAND

As depicted in the previous section, we conducted an extensive exploratory data analysis (EDA) on the entire dataset. However, if you are interested in examining the EDA specifically for a particular column, follow the steps outlined below, where I will showcase the EDA process for the 'median_house_value' column within the dataset.

```
plot(data, 'median_house_value')
```

Stats | Histogram | KDE Plot | Normal Q-Q Plot | Box Plot

Overview

Distinct Count	3842
Unique (%)	18.6%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Memory Size	322.5 KB
Mean	206855.8169
Minimum	14999
Maximum	500001
Zeros	0
Zeros (%)	0.0%
Negatives	0
Negatives (%)	0.0%

Descriptive Statistics

Mean	206855.8169
Standard Deviation	115395.6159
Variance	1.3316×10^{10}
Sum	4.2695×10^{09}
Skewness	0.9777
Kurtosis	0.3275
Coefficient of Variation	0.5579

Quantile Statistics

Minimum	14999
5-th Percentile	66200
Q1	119600
Median	179700
Q3	264725
95-th Percentile	489810
Maximum	500001
Range	485002
IQR	145125

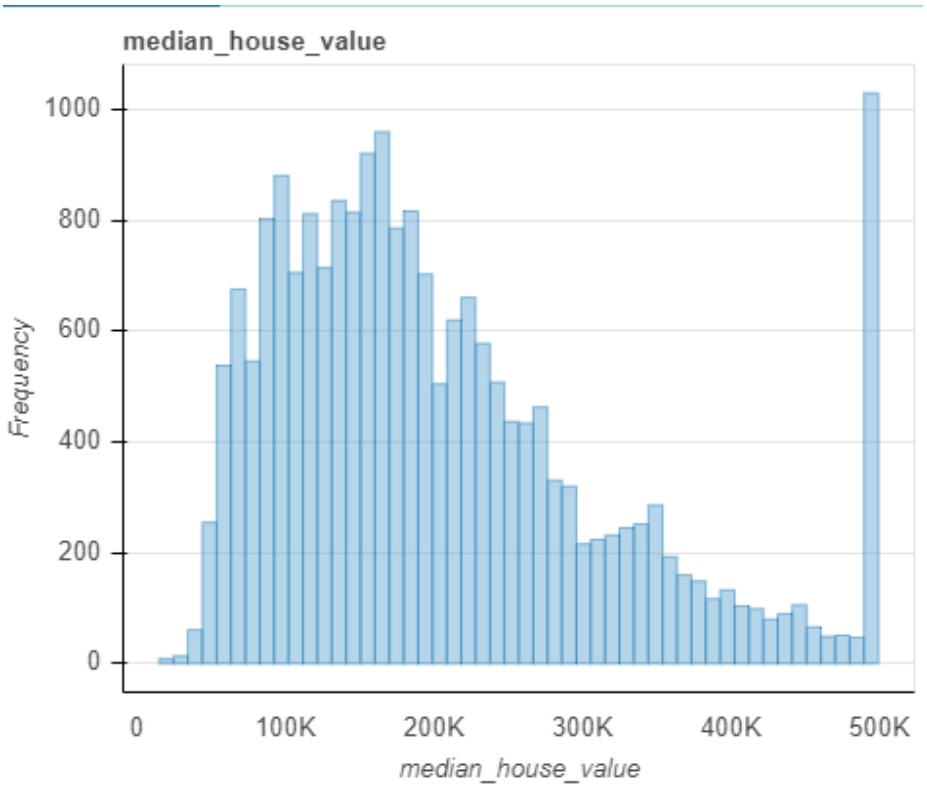


Fig.1:-*Histogram*

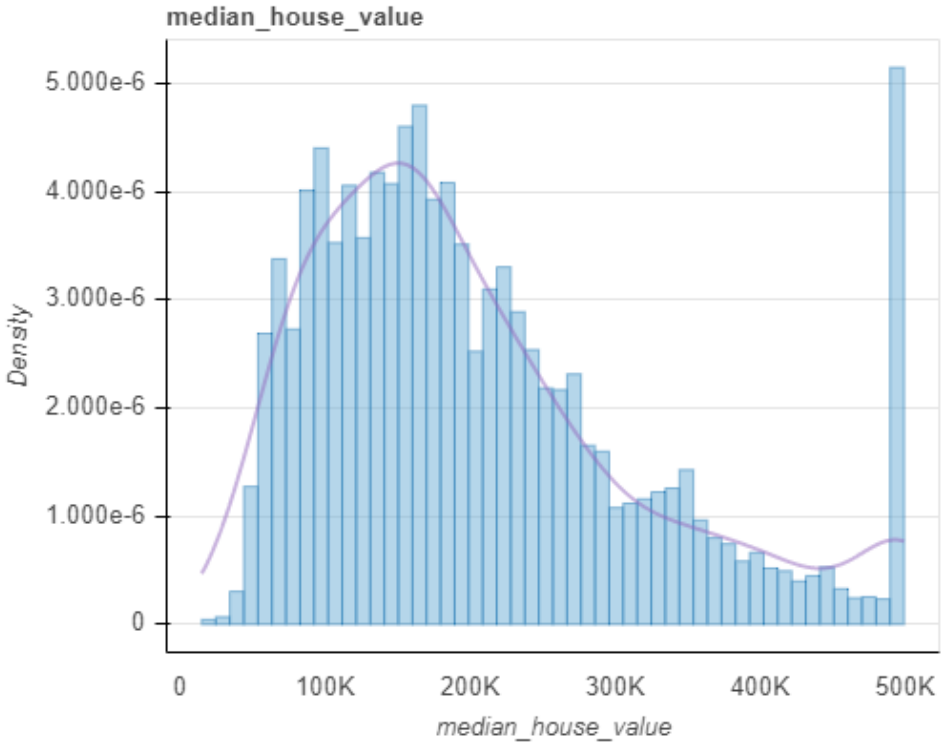


Fig.2:-*KDE Plot*

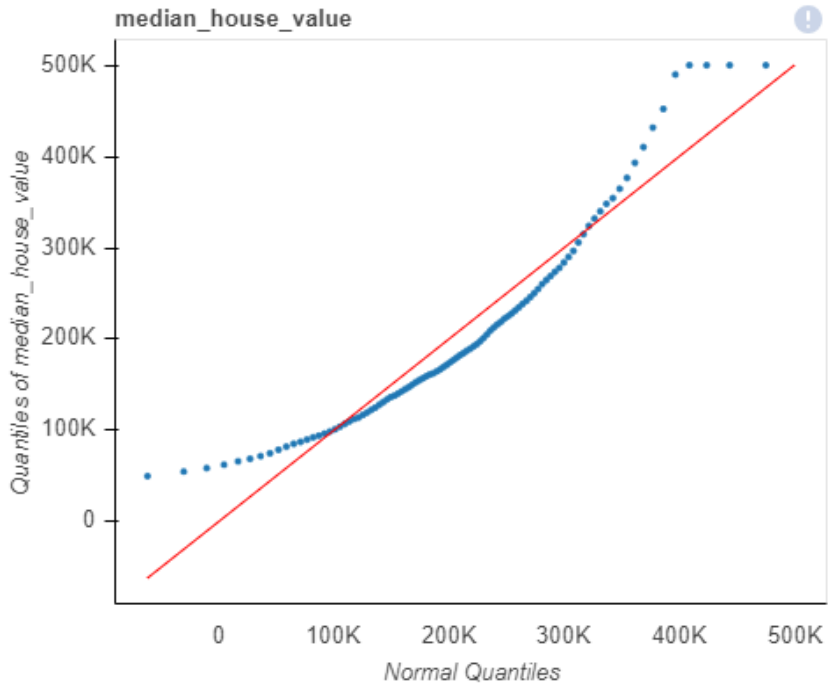


Fig.3:-*Normal Q-Q Plot*

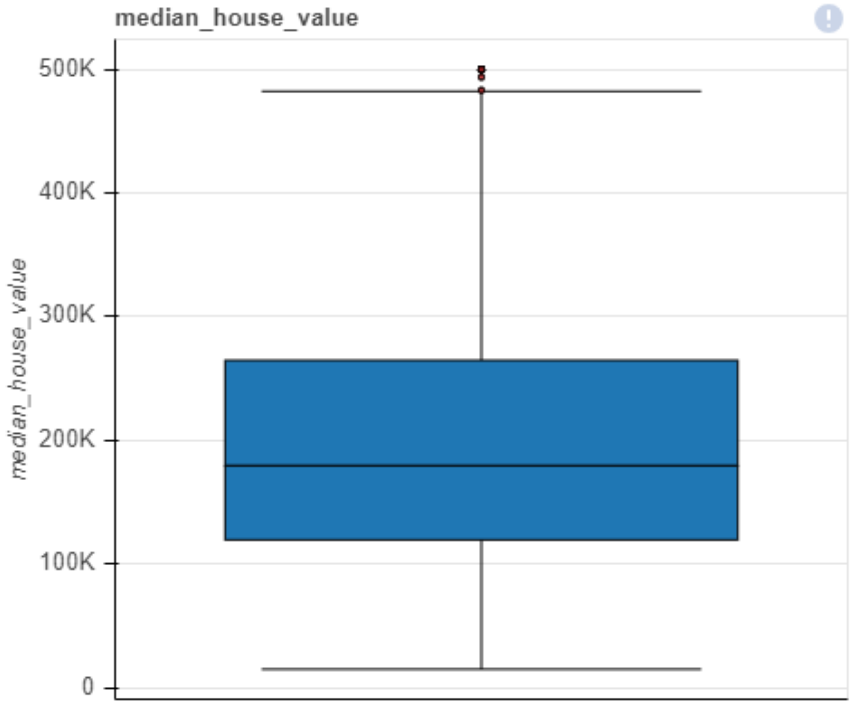


Fig.4:-*Box plot*

CONCLUSION

By leveraging Python's robust ecosystem, data scientists can effortlessly understand and analyze complex datasets, identifying key patterns, outliers, and missing values. The tutorial provided a comprehensive overview of how to conduct EDA efficiently, showcasing the power of Python libraries such as Pandas, Matplotlib, Seaborn, and Plotly. By

embracing automated EDA techniques, data scientists can streamline their workflow, gaining deeper insights and making informed decisions. This tutorial serves as a valuable resource for both beginners and experienced practitioners in the field of data science.

REFERENCES

1. Seshadri, R., Chandakkar, P. S., & Mathur, A. (2019). AutoViz: Automatic Visualization of Data. IEEE Transactions on Visualization and Computer Graphics, 26(1), 1018-1028.
2. Polfliet, J., Van den Bossche, J., & De Wachter, S. (2020). Pandas Profiling: Automatic Exploratory Data Analysis. Journal of Open Source Software, 5(54), 2591.
3. Watson, A. (2021). AutoEDA: Automatic Exploratory Data Analysis with a Focus on Statistical Testing. Journal of Open Source Software, 6(57), 2892.
4. Bertrand, F. (2020). Sweetviz: Automated EDA and Visualization. Journal of Open Source Software, 5(55), 2763.
5. DataPrep - A Python Library for Data Preprocessing. Retrieved from: https://github.com/sfu-db/DataPrep