Introduction to Data Science
CA0 - Report
1402/12/21

Amirreza Akbari | 810899045
Reza Baghestani | 810899046
Hananeh Jamali | 810899053

Purpose

Web Scraping and Introductory Data Analysis

Summary

Briefly, we plan to scrape data of the last 10 transaction blocks from Etherscan.io, perform various data cleaning methods on the data, conduct basic statistical calculations and analysis, and visualize the data with different plots. Then we are going to generate two samples from the population using Simple Random Sampling (SRS) and Stratified Sampling methods, conduct all previous calculations, analysis and visualizations on the samples, and compare them with the population.

Dataset

The collected dataset for this assignment is the transaction data from the last 10 blocks on the Etherscan website. Each block consists of approximately 150 to 200 unique transactions on average. Every transaction has this recorded information: Txn Hash (ID code), Method, Block (block number), Age (register time), From (source), To (destination), Value, Txn Fee. The dataset is later modified and cleaned for better use and analysis. A copy of the original cleaned dataset is also generated that excludes the transactions with a Value of 0.

All the dataframes (datasets) used in this project are only used for analyzing and conducting different statistical calculations.

1. Environment Setup

Jupyter Notebook is used as the Python environment for this project. Necessary libraries are installed and imported such as pandas, numpy, matplotlib.pyplot, seaborn, selenium, and scipy.

2. Data Collection

Data Scraping

In this section, a Python script uses the Selenium library to navigate to the Etherscan website and extract transaction data with the Chrome WebDriver.

The script waits for the dynamic content (transaction table) to load, using WebDriverWait. The latest transaction's block number is then extracted from the table.

The script proceeds to extract data from multiple blocks by navigating to specific URL of each block and iterating through all of their existing pages (handling pagination) until a certain preset target block count is met. In this case the target block count is set to 10. The data of each transaction is located and scraped by their xpath attribute. A random delay between 1 and 3 seconds is also added to consider rate limiting issues.

Finally, the extracted raw data is stored in a CSV file using a pandas DataFrame created from the list of dictionaries.

○ Data Cleaning

This section focuses on cleaning the raw Ethereum transaction data with various methods ensuring preparation of the data for subsequent analysis.

The raw data is loaded from the 'raw_ethereum_transactions.csv' file into a pandas DataFrame. These cleaning methods have been applied to the dataset:

- **Duplicate Removal:** Duplicate rows are removed from the DataFrame to ensure data integrity.
- **Column Removal:** Columns such as 'Txn_Hash', 'Block', 'Age', 'From', and 'To' are dropped as they are unnecessary for future analysis.
- **Missing Values:** Missing values in the 'Value', 'Txn_Fee', and 'Method' columns are filled with specific values. 'Value' and 'Txn_Fee' are filled with 0, while 'Method' is filled with 'other'.
- **Categorizing Transaction Methods:** Transaction methods are divided into 5 general categories: Transfer, Unidentified, Approve, Execute, and Swap.
- **Converting Wei to Ether:** Some of the Value data is recorded in the Ethereum cryptocurrency Wei. This data is converted to Ether (ETH).
- **Numeric Conversion:** The 'Value' and 'Txn_Fee' columns are converted to numeric types. The 'ETH' string is removed from the 'Value' column, and both columns are transformed to numeric, handling errors as needed.

The clean DataFrame is saved to a CSV file named 'clean_ethereum_transactions.csv.' Additionally, a copy of the DataFrame excluding transactions with 'Value' equal to 0 is saved as 'clean_ethereum_transactions_without_zeros.csv.'

• 3. Data Analysis

○ Statistical Analysis:

This section focuses on conduction of simple and introductory statistical calculations on the clean dataset for a better understanding of the distribution of transaction “Value”s and “Fee”s. The mean, median, and standard deviation of Value and Fee columns are evaluated for this purpose.

All of the aforementioned calculations are conducted on the Value column of both clean datasets (with and without zero values) and the Fee Column of the full clean dataset in two conditions. First without any special conditions on grouping and then with grouping the data based on their method.

The results indicate that the presence of many zero values in the 'Value' column significantly affects the mean and median in the full dataset. Excluding these zero values leads to a higher mean and median. The standard deviation is relatively high in both 'Value' rows, indicating a wide distribution of values.

The 'Txn_Fee' column has a lower mean and median compared to the 'Value' column, suggesting lower variability in transaction fees.

○ Visualization:

The code in this section generates different plots for the same data that the statistical calculations and analysis were conducted on. The following plots are generated for the Value column of both clean datasets (with and without zero values) and the Fee Column of the full clean dataset

- **Histogram Plots:** These plots show that a great fraction of the data is focused very close to zero and consequently the histogram plots are not a good visualization choice.
- **Logarithmic Histogram Plots:** Since the data is very close to zero, using a logarithmic scale transforms the data into a more understandable visualization.
- **Normal Distribution Plots:** A normal distribution plot is fitted alongside the logarithmic histogram plots to compare the empirical distribution of the data with the theoretical normal distribution. These plots indicate that both Value plots are very similar to the adjusted fitted normal distribution plots. Also, the Fee plot usually shows a skewness to the left side.
- **Box and Violin Plots:** These plots are also generated to identify potential outliers and provide a comprehensive view of the data's distribution.

• 4. Data Sampling

In this section, we will delve into the process of data sampling and perform an initial analysis on the transaction data we have collected. Our objective is to understand the distribution of transaction values by sampling the data and comparing the sample statistics with the population statistics.

Simple Random Sampling and Stratified Sampling methods are used to create two different samples to work with in this part of the project.

○ Sampling

In the SRS step, a sample is drawn from the clean dataset with 10% of the total transactions. The use of a random state (42) ensures reproducibility. This random sample is crucial for making inferences about the entire dataset, providing insights without the need to analyze the complete data.

Stratified sampling is performed to ensure representation across different 'Method' categories. The sample sizes are determined proportionally based on the distribution of 'Method' in the original dataset. This method is beneficial when there are distinct groups within the data, and it helps to capture the characteristics of each stratum more effectively. The random state (42) is set for reproducibility, ensuring consistent results in subsequent runs.

○ Visualization

The code in this section generates different plots for the same data that we conducted statistical calculations and analysis on in the previous section. The following plots are generated for the Value and Fee columns of both generated samples and the clean database without zero values (the population).

- **Histogram Plots:** These plots show that a great fraction of the data is focused very close to zero same as the plots in the previous section (but slightly less focused in comparison) and consequently the histogram plots are not a good visualization choice.
- **Logarithmic Histogram Plots:** Again, since the data is very close to zero, using a logarithmic scale transforms the data into a more understandable visualization. These plots indicate that both Value plots are slightly similar to the normal distribution plots but in a bimodal distribution. The Fee plot still does not exhibit a good visualization.

All in all, the statistical and plot analysis of the generated samples show a good resemblance to the original dataset (the population). But one statistical analysis (Value or Fee) of one or two of the samples show a significant difference with the statistical analysis on the original dataset, where the plots seem to be always similar.

• 5. Conclusion

In conclusion, this comprehensive analysis of Ethereum transaction data has provided valuable insights into the distribution and characteristics of transactions within the last 10 blocks. The data collection process involved web scraping from Etherscan.io, followed by detailed cleaning to ensure the accuracy and reliability of subsequent analyses. By conducting statistical calculations, visualizations, and sampling methods, we explored the distribution of transaction values and fees. The study revealed interesting patterns, particularly the concentration of transactions around zero values. Logarithmic scale plots and normal distribution fittings revealed the data's nature better. The exploration of simple random and stratified sampling techniques offered a glimpse into the dataset's diversity and helped establish representative samples for further investigation. Overall, this analysis provides a robust foundation for understanding Ethereum transactions and sets the stage for more in-depth studies and hypothesis testing.

- ## 6. References

Ethereum

Selenium Documentation

Pandas Documentation

NumPy Documentation

Matplotlib Documentation

Seaborn Documentation

SciPy Documentation

Medium

ChatGPT 3.5 & Claude 3 Sonnet

Stack Overflow