Proceedings of EKONBIZ 2026
ISBN: 978-99955-45-52-9
UDK: 005.133:004.43PYTHON

Submitted: 31.03.2026.
Accepted: 29.05.2026.
Original Article

STATISTICAL DATA PROCESSING IN ECONOMICS USING THE PYTHON PROGRAMMING LANGUAGE

Srđan Damjanović
Faculty of Business Economics Bijeljina, University of East Sarajevo, Republic of Srpska, BiH
srdjan.damjanovic@fpe.ues.rs.ba
ORCID: 0000-0003-4807-5311

Predrag Katanić
Faculty of Business Economics Bijeljina, University of East Sarajevo, Republic of Srpska, BiH
predrag.katanic@fpe.ues.rs.ba
ORCID: 0000-0002-6476-809X

Luis Silva Rodrigues
CEOS.PP, ISCAP, Polytechnic of Porto, Portugal
lsr@iscap.ipp.pt
ORCID: 0000-0001-5471-8083

Juliana Smirnova
Vilnius Gediminas Technical University, Vilnius, Lithuania
juliana.smirnova@vilniustech.lt
ORCID: 0000-0001-5182-4139

Abstract: *This paper examines the use of the Python programming language for statistical processing of economic data, with a particular focus on its role in financial analysis and education. In the context of increasing data availability and the growing importance of data-driven decision-making, Python has emerged as a practical and flexible environment for performing statistical analysis. The study highlights the use of key Python libraries — including NumPy, Pandas, SciPy, and Matplotlib — for data collection, preprocessing, analysis, and visualization.*

The empirical part of the paper illustrates how Python can be used to retrieve real-world financial data from online sources and perform both descriptive and inferential statistical analyses. Using daily closing stock price data for selected companies over the period 2019–2025, the paper demonstrates the application of basic statistical measures, hypothesis testing methods (paired t-test and chi-square test), and correlation analysis (Pearson correlation matrix). The results indicate the practical utility of Python in handling economic datasets and supporting the interpretation of quantitative findings.

The paper also emphasizes the importance of integrating programming skills into economics education in order to bridge the gap between theoretical statistical knowledge and practical data analysis. By enabling economists to conduct the full analytical workflow independently, Python can support more efficient and timely data-driven decision-making..

Key words*: Python, statistical analysis, economic data, data analysis, data visualization*

JEL classification*: O33*

1. INTRODUCTION

In the era of big data, the importance of data analysis technologies in finance has increased substantially. Among contemporary programming languages, Python has emerged as one of the most widely used tools in the fintech sector due to its flexibility, readability, and rich ecosystem of libraries (Qiao, 2025; Yi, 2025). Python is particularly attractive for financial analysis because of its concise syntax and extensive collection of open-source libraries and customizable development packages. Financial professionals can use Python to retrieve data from enterprise ERP systems, CSV files, Excel spreadsheets, relational databases, and financial software APIs. Libraries such as Pandas enable efficient data cleaning and preprocessing, including handling missing values, correcting data formats, standardizing dates and numerical representations, and identifying or removing outliers. These steps are essential for ensuring data quality prior to further analysis, reporting, and modeling. In addition to traditional visualizations such as bar charts, line graphs, and pie charts, Python also supports the development of advanced and interactive visual analytics that improve the interpretability of financial data and support timely managerial decision-making (Wei, 2025).

Statistical analysis is the process of collecting and examining datasets to identify patterns and derive meaningful conclusions. In modern data-driven environments, statistical analysis involves collecting raw data and identifying relationships among variables to reveal patterns and trends relevant to decision-making. As data science continues to grow, statistics has become a fundamental component of analytical processes and is widely taught at economics faculties around the world (Holman & Hacherl, 2023). Practical statistical knowledge and analytical skills play a crucial role in transforming raw data into valuable economic insights.

Statistical analysis generally includes two main approaches: descriptive and inferential statistics. Descriptive statistical analysis summarizes and organizes data without drawing broader conclusions, while inferential statistical analysis allows researchers to make generalizations or predictions based on the analyzed data. Professionals working in statistical analysis contribute to scientific research, improve social and economic conditions, and support informed business decision-making.

Despite the recognized importance of statistical analysis, the traditional teaching of statistics at many economics faculties often lacks practical data science applications relevant to real business environments (Riyantoko et al., 2024). In economic research practice, data collection and statistical processing are frequently performed by different specialists. Universities often rely on commercial statistical software packages, such as SPSS, for statistical data analysis. However, many economists lack sufficient expertise to use such software tools independently and therefore rely on external experts to perform statistical analyses. As a consequence, economists can begin analyzing their data only after statistical processing is complete, which often creates a significant time gap between data collection and the interpretation of results.

In an environment where rapid decision-making is increasingly important, there is a growing need to shorten the time required to process and analyze data. This paper proposes that economics students should acquire basic Python programming skills to independently collect, process, and analyze large datasets from various online sources and databases (Hashimzade et al., 2025). Using Python, both data acquisition and statistical processing can be performed within a single programming environment. As a result, analytical processes that previously took weeks or months can now be completed in minutes. This approach allows economists to dedicate more time to interpreting results and drawing meaningful conclusions about future economic developments.

Python programming has therefore become an essential tool for implementing data science methods in economics. Its libraries for numerical computation and statistical analysis — such as NumPy, Pandas, and SciPy — enable efficient data manipulation, modeling, and analysis (Bilina & Lawford, 2012). Moreover, Python's simple and readable syntax facilitates learning and helps students understand the computational logic behind statistical procedures. The integration of statistical analysis with Python programming allows students to practice analytical methods through practical examples and computational experimentation (Riyantoko et al., 2025).

The growing demand for programming competencies in economics underscores the need to integrate programming into undergraduate economics curricula systematically (Hashimzade et al., 2025). Laboratory-based and case-oriented teaching methods allow students to work with realistic financial data scenarios and complete the full analytical workflow — from data acquisition and cleaning to statistical analysis and visualization — using Python tools. By simulating real-world financial data processing environments, students gain practical insight into financial data management and develop the ability to identify and solve analytical problems.

Nevertheless, a noticeable gap still exists between the programming competencies expected at the graduate level and those typically acquired during undergraduate studies, particularly in areas such as numerical methods, optimization, and dynamic modeling. This paper seeks to contribute to the broader discussion on integrating programming into undergraduate economics education and to encourage the exchange of experiences and best practices in this field. The contribution of the paper is primarily educational and applied: it illustrates how Python can be used to perform a complete analytical workflow, from data acquisition and preprocessing to descriptive statistics, inferential testing, correlation analysis, and visualization of financial data. The increasing use of artificial intelligence (AI) in financial analytics and the automation of analytical workflows further emphasize the importance of aligning economics education with modern technological developments (Schwarz, 2025). Although AI-based tools may support programming and statistical analysis, the primary focus of this paper is the use of Python as a practical environment for collecting, processing, analyzing, and visualizing economic data.

Overall, the growing body of research on the application of Python in economics illustrates its strong potential for the statistical processing of large datasets and for improving analytical efficiency in economic research (Bilina & Lawford, 2012; Qiao, 2025; Wei, 2025). Although significant progress has been made, many economists remain reluctant to engage in programming, underscoring the importance of further promoting programming literacy in economics education (Riyantoko et al., 2024; Riyantoko et al., 2025).

2. LITERATURE REVIEW

Numerous researchers have investigated the use of Python in economics and data analysis. A significant part of this research has focused on the statistical processing of data using Python, as well as on reviewing studies that analyze the role of Python in statistical and data-driven applications.

One of the earlier studies in this area was conducted by Bilina and colleagues in 2012. Their research examined speech signal processing, including automatic speech recognition, synthetic speech, and natural language processing. The authors emphasized the growing importance of these technologies for business, industry, and personal computing. They also analyzed the evolution of speech recognition systems in industrial applications and highlighted how such technological developments enable next-generation voice-enabled services. The study provided a comprehensive overview of speech recognition technologies and summarized findings from prior research, while also presenting their potential applications in healthcare, robotics, forensic science, defense, and aviation.

More recent studies emphasize the role of Python in teaching statistical computing methods. In 2023, Holman and Hacherl examined the increasing importance of statistical computing skills for future business professionals in the context of the growing relevance of data science in organizational decision-making processes. Their research focused on the Monte Carlo simulation, which is widely use in many academic and professional fields. The authors proposed a pedagogical approach for teaching Monte Carlo simulation using Python. In their study, students first completed simulation exercises in spreadsheets and then repeated the same tasks using Python. The results indicated that such a teaching strategy can significantly enhance students' understanding of statistical computing, particularly among those familiar with spreadsheet tools but with limited programming experience.

In 2024, Riyantoko and colleagues explored the role of statistical learning through programming in improving statistical literacy and supporting educational innovation in technology-enhanced learning environments. Their research addressed the difficulties that many novice students face when learning statistics through programming due to the conceptual and technical complexity involved. The authors proposed a set of Python programming exercises based on three problem types: Element Fill-in-Blank Problems (EFP), Grammar-Concept Understanding Problems (GUP), and Value Trace Problems (VTP). These exercises were designed to support self-study in introductory statistics courses. The evaluation was conducted on 67 first-year undergraduate students at UPN “Veteran” Jawa Timur University in Indonesia. The results demonstrated that the proposed approach can effectively assist students in learning statistics using Python.

Also in 2024, Ho conducted a study as part of the Statistical Analysis for Data Science course within the MSc in IT and Data Science program at the European International University in Paris. The project used several statistical techniques in Python, combined with MySQL, to analyze company payroll and employee data. The study illustrated how Python can be used for practical statistical analysis in real business datasets and demonstrated its potential in applied data science education.

Further research conducted in 2025 by Riyantoko and colleagues expanded on their earlier work by developing a self-learning method for fundamental statistics in data science education using Python. Their approach integrates interactive learning problems into the Programming Learning Assistant System (PLAS), combining coding exercises, conceptual understanding tasks, and output-tracing activities. The study involved 40 first-year undergraduate students and used statistical methods to evaluate learning outcomes. The results showed a statistically significant correlation ($\rho < 0.05$) between students who used the proposed learning method and those who did not, indicating that the approach can effectively improve student knowledge retention and practical statistical skills when learning statistics through Python programming.

Finally, in 2025, Schwarz examined the application of generative artificial intelligence in statistical data analysis, focusing on the use of the tool ChatGPT in statistics education at universities of applied sciences. The study analyzed how generative AI can support statistical analysis by generating appropriate programming code and assisting users with limited statistical knowledge. By using artificially generated datasets, the author demonstrated both the capabilities and limitations of AI-assisted statistical analysis. The findings indicate that while generative AI tools can facilitate data analysis, human supervision and a solid understanding of statistical concepts remain essential. The study concludes that statistics education should place greater emphasis on conceptual understanding rather than solely on software-specific technical skills, thereby enabling students to use AI-based analytical tools effectively while maintaining methodological rigor.

3. METHODOLOGY AND DATA

3.1. RESEARCH PURPOSE AND DATA SOURCE

The empirical part of the paper is designed as an applied demonstration of Python-based statistical analysis using real-world financial data. The objective is not to provide investment recommendations, but to illustrate how Python can support data acquisition, descriptive analysis, statistical inference, correlation analysis, and visualization in an educational economics context.

The financial data were obtained from Yahoo Finance using the Python yfinance library. The analysis uses daily closing stock prices for selected companies. The code defines the observation window from January 2019 to December 2025, while the actual number of observations depends on data availability at the time of retrieval. The complete Python code used for data acquisition and all analyses is provided in Appendix A.

3.2. SELECTION OF COMPANIES

Two sets of companies were selected for the empirical analyses, chosen to illustrate different patterns of stock price behaviour during the observed period.

For the exploratory data analysis (Section 4.1), the following companies were selected:

- **Apple Inc. (AAPL)** — a large technology company with a generally upward price trend over the observed period, representing growth-oriented stocks.
- **PepsiCo (PEP)** — a consumer goods company exhibiting a more stable price trajectory with a mild upward trend, representing lower-volatility stocks.
- **Tesla Inc. (TSLA)** — a technology and electric vehicle company with pronounced price fluctuations during the observed period, representing high-volatility stocks.

For the inferential statistical analysis (Section 4.2), the following companies were selected:

- **Apple Inc. (AAPL)** — included for continuity with the previous section and as a benchmark technology stock.
- **The Coca-Cola Company (KO)** — a consumer goods company with a more stable price trajectory, selected to contrast with the dynamics of technology stocks.
- **Beyond Meat (BYND)** — a company with pronounced price fluctuations during the first part of the observed period followed by a sustained price decline, selected to illustrate divergent market behaviour relative to the other two companies.

The selection of these companies is intended for illustrative and educational purposes. The companies were chosen to represent different patterns of volatility and trend behaviour rather than to provide a representative sample of any market segment.

3.3. STATISTICAL METHODS

The analysis proceeds in two stages. The first stage (Section 4.1) applies exploratory data analysis using descriptive statistics and visualizations. The following measures are computed: mean, median, standard deviation, minimum, maximum, and frequency distribution. Visualizations include a line chart of stock price trends and histograms of price distributions.

The second stage (Section 4.2) applies inferential statistical methods. Before applying parametric tests, the normality of the data should be assessed; the Shapiro–Wilk test is included in Appendix A for this purpose. The empirical analysis then applies: (1) a paired t-test to compare the closing prices of Apple and Coca-Cola; (2) a chi-square test of independence to examine the relationship between the directions of daily price changes for Apple and Beyond Meat; and (3) Pearson correlation analysis to assess the linear relationships among all three companies. The complete code for all methods is provided in Appendix A.

3.4. LIMITATIONS

The analysis is limited to selected publicly traded companies and to historical stock market data. The results should therefore be interpreted as an educational and methodological demonstration rather than as a general conclusion about all financial markets or as an investment recommendation. In addition, correlation analysis does not imply causality, and statistical significance should be interpreted in conjunction with economic context and domain knowledge.

4. PYTHON-BASED STATISTICAL ANALYSIS IN ECONOMICS

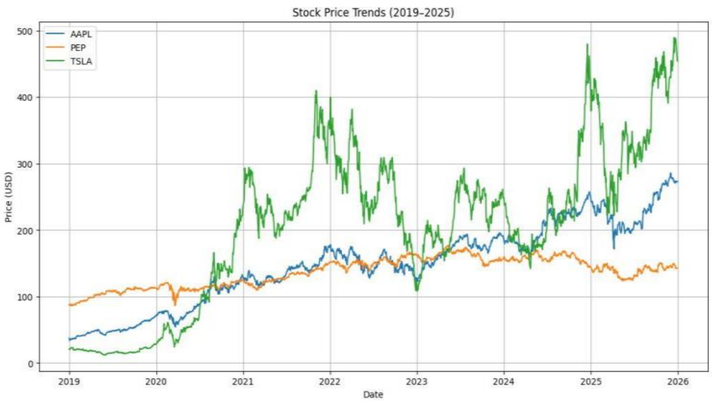
Statistics provides a methodological framework for collecting, organizing, analyzing, and interpreting economic data. In economics, statistical methods enable researchers to identify patterns, test hypotheses, estimate relationships among variables, and support evidence-based decision-making. With the support of libraries such as NumPy, SciPy, and Pandas, the Python programming language enables efficient processing of large datasets, advanced statistical analysis, and the creation of visualizations that facilitate a clearer interpretation of results.

Using the NumPy library, economists can efficiently manipulate numerical data and perform fundamental statistical calculations, such as computing means, standard deviations, and correlations. The SciPy library provides access to advanced statistical techniques, including hypothesis testing and regression analysis. At the same time, Pandas enables intuitive manipulation and analysis of tabular data that closely align with the needs of economic research. Together, these libraries enable fast, reliable, and scalable data analysis, even for large datasets that would be difficult to process with traditional methods.

4.1. EXPLORATORY DATA ANALYSIS WITH PYTHON

The first step in any statistical analysis is the collection of data to be analyzed. A dataset represents an organized collection of values that describe specific phenomena, processes, or variables. The quality of the analysis largely depends on how accurately the data are collected, categorized, and processed. In economics, data originate from various sources — such as surveys, market research, official government statistics, or financial reports — and they often serve as the basis for business and policy decision-making. In this section, daily closing stock price data for Apple, PepsiCo, and Tesla were retrieved from Yahoo Finance using the yfinance library, covering the period from 2019 to 2025. Figure 1 presents a comparative overview of stock price movements for the three selected companies over the observed period.

Figure 1. Stock Price Trend for Apple, PepsiCo, and Tesla



Source: Authors

The descriptive analysis was conducted in Python using yfinance for data acquisition, pandas and NumPy for data processing, and Matplotlib for visualization. An efficient approach to computing the main descriptive statistics is to use the describe() method available in pandas, which provides the count, mean, standard deviation, minimum, quartiles, and maximum in a single operation. The complete code is provided in Appendix A. Table 1 summarizes the key descriptive statistics obtained for the three companies.

Table 1. Descriptive statistics of daily closing stock prices (2019–2025)

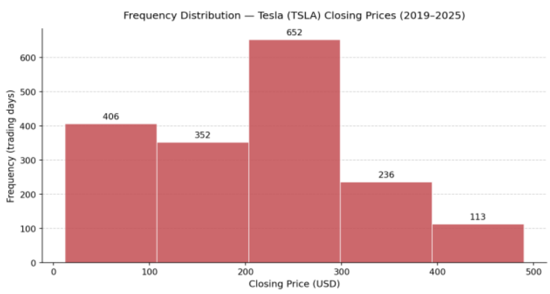
Company	Mean (USD)	Median (USD)	Std. Dev. (USD)	Min (USD)	Max (USD)
Apple (AAPL)	148.27	150.08	61.14	33.77	285.92
PepsiCo (PEP)	136.83	141.73	21.83	86.04	177.25
Tesla (TSLA)	205.73	220.17	118.22	11.93	489.88

Source: Authors

The results in Table 1 illustrate substantial differences in the price dynamics of the three companies. Tesla shows the highest standard deviation (118.22 USD) and the widest price range (approximately 478 USD), indicating the greatest variability among the selected stocks during the observed period. PepsiCo exhibits the lowest standard deviation (21.83 USD) and the narrowest price range (approximately 91 USD), reflecting more stable price movements consistent with a consumer goods company. Apple occupies an intermediate position in terms of variability, with a generally increasing price trend visible in Figure 1.

Figure 2 presents the frequency distribution (histogram) of Tesla's closing prices, generated using Python. The distribution shows a concentration of observations in the middle price range, with fewer observations at the extremes, consistent with the relatively large standard deviation and wide price range noted in Table 1.

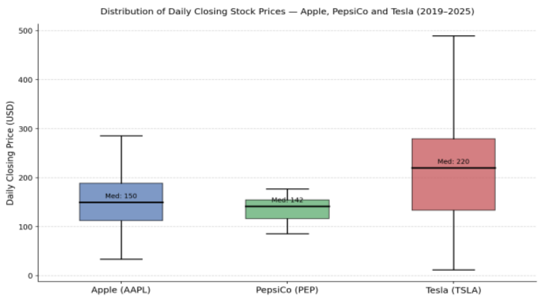
Figure 2. Frequency Distribution for Tesla Frequency Distribution for Tesla



Source: Authors

The histogram illustrates the distribution of Tesla's daily closing prices across five intervals. The results show that the largest number of observations falls in the interval from approximately 203 to 299 USD (652 observations), while the fewest observations appear in the highest price range (394–490 USD, 113 observations). This pattern is consistent with the pronounced price fluctuations and the periods of rapid increase and subsequent decline characteristic of Tesla's stock during the observed period.

Figure 3. Box Plot of Daily Closing Stock Prices for Apple, PepsiCo, and Tesla (2019–2025)



Source: Authors

While histograms summarize the overall distribution of values, box plots provide a more detailed view by simultaneously displaying the median, interquartile range (IQR), and potential outliers for each company. Figure 3 presents the box plot for all three selected companies, generated in Python using the plot (kind='box') method from the pandas library.

The box plot reinforces and extends the findings from Table 1 in several important ways. The box for PepsiCo is the narrowest and is positioned in the lowest price range among the three companies, with a compact interquartile range that confirms the stability of its price movements throughout the observed period. The relatively small number of values falling outside the whiskers indicates that extreme price observations were rare for PepsiCo. The box for Apple is wider and positioned higher, reflecting the general upward trend in its price over the observed period; the presence of lower outliers corresponds to the period of lower prices in the early part of the observation window (2019–2020). Tesla's box is the widest of the three, with the highest median and the most pronounced spread between the first and third quartile, consistent with its high standard deviation reported in Table 1. The long whiskers and the presence of outliers at both ends of Tesla's distribution indicate periods of unusually high and unusually low prices, which reflect the strong market fluctuations characteristic of this stock during the observed period.

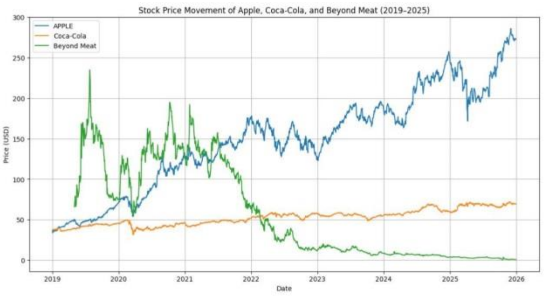
Taken together, the line chart (Figure 1), the histogram (Figure 2), and the box plot (Figure 3) illustrate how different types of visualization complement one another in exploratory data analysis: the line chart reveals temporal dynamics and trends, the histogram summarizes the overall frequency distribution, and the box plot compares distributional properties — including central tendency, dispersion, and the presence of extreme values — across multiple companies in a single visual output.

4.2. STATISTICAL INFERENCE WITH PYTHON

Statistical inference is a set of methods that enable drawing conclusions about a population from a sample. While descriptive statistics focuses on describing data, inferential statistics allows for hypothesis testing, parameter estimation, and forecasting. The main purpose of inferential statistics is to derive reliable conclusions about an entire population based on a limited sample, a particularly useful approach in economics, where it is often impossible to analyze all available data.

In this section, daily closing price data for Apple, Coca-Cola, and Beyond Meat are used to demonstrate three inferential methods: a paired t-test, a chi-square test of independence, and Pearson correlation analysis. Figure 4 provides a comparative overview of stock price movements for these three companies over the observed period.

Figure 4. Stock Price Trend for Apple, Coca-Cola, and Beyond Meat



Source: Authors

Normality testing. Before applying parametric tests, it is advisable to assess whether the data satisfy the normality assumption. For the paired t-test, this involves testing the normality of paired differences.

The Shapiro–Wilk test is an appropriate method for this purpose. The code for performing the Shapiro–Wilk test on the paired differences between Apple and Coca-Cola closing prices is provided in Appendix A. If the test indicates a significant departure from normality ($p \leq 0.05$), the Wilcoxon signed-rank test — a nonparametric alternative to the paired t-test — should be considered. Code for both tests is included in Appendix A.

Paired t-test: Apple vs. Coca-Cola. A paired t-test was applied to compare the daily closing prices of Apple and Coca-Cola over the shared observation period, using only dates for which data were available for both companies. The results are as follows:

T-statistic: 76.1215

P-value: 0.000000

The paired t-test indicates a statistically significant difference between the daily closing prices of Apple and Coca-Cola during the observed period. However, this result should be interpreted cautiously.

Stock prices are measured on different absolute scales and are influenced by company-specific factors, sectoral dynamics, and broader market conditions. The large t-statistic primarily reflects the difference in price levels between the two companies rather than a meaningful economic relationship. The test illustrates the use of Python for parametric statistical inference; for a more informative comparison of co-movement between the two stocks, an analysis based on daily returns rather than price levels would be more appropriate.

Chi-square test: Apple vs. Beyond Meat. The chi-square test of independence was applied to examine whether the direction of daily price changes (upward or downward) for Apple and Beyond Meat is statistically independent. Daily movements were categorized as 1 (price increase) or 0 (price decrease). The contingency table is shown below.

Table 2. Chi-square test: Apple vs. Beyond Meat

	BYND: 0	BYND: 1
AAPL: 0	491	291
AAPL: 1	428	466

Source: Authors

Chi2 Statistic: 36.8576

P-value: 0.000000

Degrees of freedom: 1

The statistically significant result suggests that the direction of daily price changes for Apple and Beyond Meat is not independent. Specifically, the contingency table indicates an asymmetry: on days when Beyond Meat's price declined (BYND: 0), Apple's price was more evenly split between increases and decreases, while on days when Beyond Meat's price increased (BYND: 1),

Apple's price was more likely to have increased as well. However, this association does not imply a causal relationship between the two stocks. The result may reflect shared exposure to broader market movements rather than any direct economic link between the two companies.

Pearson correlation analysis. Pearson correlation analysis was used to assess the strength and direction of linear relationships among the closing prices of the three companies. The resulting correlation matrix is in Table 3.

Table 3. Pearson correlation analysis

	AAPL	KO	BYND
AAPL	1.000	0.884	−0.757
KO	0.884	1.000	−0.819
BYND	−0.757	−0.819	1.000

Source: Authors

The strong positive correlation between Apple and Coca-Cola ($r = 0.884$) indicates that their price series moved in a broadly similar direction during the observed period. This may partly reflect a shared upward trend common to both stocks over the 2019–2025 window rather than a direct economic relationship between the two companies. The negative correlations involving Beyond Meat ($r = -0.757$ with Apple and $r = -0.819$ with Coca-Cola) are consistent with the price decline of Beyond Meat during a period when both Apple and Coca-Cola generally increased in value. As noted in the methodology section, correlation does not imply causality, and price-level correlations may be influenced by trends that are independent of any true co-movement. For a more rigorous analysis of stock market interdependencies, future research could compute correlations based on daily returns rather than price levels.

It should also be noted that Pearson correlation is a parametric method that assumes linear relationships and approximately normally distributed variables. Given that stock price series often exhibit non-normality and trend-related properties, the interpretation of Pearson correlation coefficients should be considered in the context of these limitations.

DISCUSSION

The empirical examples presented in this paper demonstrate that Python can support the full analytical workflow - from data acquisition to descriptive statistics, inferential testing, and visualization - within a single programming environment. Several observations merit further discussion.

From a descriptive perspective, the comparison of Tesla, Apple, and PepsiCo illustrates how Python can efficiently identify differences in price dispersion and trend behaviour across companies with very different market profiles. The combination of summary statistics (Table 1) and visual output (Figures 1, 2, and 3) provides a more complete picture than either approach alone: the table quantifies the differences precisely, while the line chart, histogram, and box plot convey the temporal dynamics, distributional shape, and spread of values respectively.

From an inferential perspective, the results of the three statistical tests illustrate different aspects of the relationships among the selected stocks. The paired t-test demonstrates a significant difference in price levels, though the result is primarily driven by scale differences rather than reflecting an economically meaningful co-movement. The chi-square test reveals a statistically significant association in the direction of daily price changes, which could reflect common exposure to broad market conditions. The Pearson correlation matrix shows strong positive co-movement between Apple and Coca-Cola and negative co-movement with Beyond Meat, consistent with the divergent price trends visible in Figure 4. In all cases, the statistical results should be read as descriptive summaries of the observed data rather than as causal claims.

From an educational perspective, these examples illustrate an important pedagogical value of Python: students can directly observe how choices of statistical method, data transformation, and interpretation affect the conclusions that can be drawn from the same dataset.

At the same time, the analysis has several limitations that should be acknowledged. The selected companies represent a small and non-random sample. The use of price levels rather than returns limits the interpretability of correlation and t-test results in a financial sense. The analysis does not control for macroeconomic conditions, sector effects, or other confounding factors. These limitations are inherent to a short educational demonstration and do not detract from the methodological illustration, but they reinforce the point that the results should not be interpreted as investment guidance or generalized market conclusions.

CONCLUSION

This paper illustrates the practical value of Python as an environment for statistical data processing in economics. Through selected examples based on real-world financial data, Python was used to retrieve data from an online source, compute descriptive statistics, perform inferential tests, and visualize results - all within a single programming environment. The exploratory data analysis demonstrated how Python can efficiently summarize and compare the distributional properties of stock price series with different volatility profiles. The inferential analysis illustrated the application of a paired t-test, a chi-square test, and Pearson correlation analysis, while also highlighting the importance of interpreting statistical results in their economic and methodological context rather than as isolated numerical outputs. The findings suggest that Python can help students and economists connect statistical theory with real-world data, supporting a more transparent and reproducible analytical workflow. Integrating such applied programming exercises into economics curricula may contribute to narrowing the gap between statistical theory and practical data analysis skills. However, the analysis presented in this paper should be interpreted as an applied educational demonstration rather than as a comprehensive financial market study, and the results do not constitute investment recommendations. Future research could extend the approach by using daily returns rather than price levels, applying larger and more diverse datasets, incorporating additional financial indicators, and exploring more advanced statistical or machine learning methods. Further work could also assess the effectiveness of Python-based analytical exercises as a pedagogical tool in economics education through controlled empirical studies.

REFERENCES

[1] Bilina, R., & Lawford, S. (2012). "Python for unified research in econometrics and statistics," *Econometric Reviews*, vol. 31 (5), 558–591.

[2] Damjanović, S., & Katanić, P. (2025). "Comparison of Four Programming Languages Used for Communication with Measuring Instruments," *Regionalni razvoj i demografski tokovi zemalja jugoistočne evrope*, vol. 30, 461–473.

[3] Hashimzade, N., Kirsanov, O., & Kirsanova, T. (2025). "Integrating Programming into the Modern Undergraduate Economics Curriculum," *International Review of Economics Education*, vol. 49 (3), 1–12.

[4] Ho, M. (2024). "Application of statistical techniques using Python and MYSQL," *Applied and Computational Engineering*, vol. 178 (1), 226–235.

[5] Holmana, J. O., & Hacherl, A. (2023). "Teaching Monte Carlo Simulation with Python," *Journal of Statistics and Data Science Education*, vol. 31 (1), 33–44.

[6] Katanic, P., & Damjanovic, S. (2025). *Python za ekonomiste*, (1st ed., Ch. 6, pp. 281–360) Fakultet poslovne ekonomije Bijeljina.

[7] Ponagantiadarsh, & Vada, V. P. (2025). "Real-Time stock market analytics using Python and Big Data Tools," *International Journal of Research Publication and Reviews*, vol. 6 (8), 5113–5120.

[8] Qiao, Y. (2025). "Research on the Application of Python Big Data in Financial Analysis," *Financial Economics Research*, vol. 2 (2), 7–19.

[9] Riyantoko, P., Funabiki, N., Brata, K. C., Mentari, M., Damaliana, A. T., & Prasetya, D. A. (2025). "A Fundamental Statistics Self-Learning Method with PythonProgramming for Data Science Implementations," *Information MDPI*, vol. 16(7), 1–23.

[10] Riyantoko, P., Funabiki, N., Wai, K., Aung, S., Muhaimin, A., & Trimono (2024). "A proposal of Python programming exercise problems for basic statistics learning," *In Proceedings of the 2024 Seventh International Conference on Vocational Education and Electrical Engineering (ICVEE)*, 289–295.

[11] Schwarz, J. (2025). "The use of generative AI in statistical data analysis and its impact on teaching statistics at universities of applied sciences," *Teaching statistics*, vol. 47, 118–128.

[12] Tang, S. (2026). "Creating a Python platform for students to learn full-stack development of financial applications," *Financial Innovation*, vol. 12 (1), 1–29.

[13] Wei, R. (2025). "Data Analysis with Python: Methods, Tools and Practices-Data Processing, Visualization and Prediction Based on Python," *Applied and Computational Engineering*, vol. 178 (1), 226–235.

[14] Yi, J. (2025). "The Application of Python in Financial Big Data Management and Teaching," *Management*, vol. 8 (3), 52–60.