International Multidisciplinary Journal of Emerging Technologies and Applications (IMJETA)
Vol. 1, No. 3, pp. 79-94, June 2026
Received June 02, 2026; Revised June 09, 2026; Accepted June 23, 2026
Published June 30, 2026

Automating Data Envelopment Analysis in Python: Functional Comparison with XlDEA/XIDEA and Methodological Assessment of Second-Stage Inference

Marlon Stalin Taco Arias[0009-0006-9590-5014]

RS ROTH, Quito, Ecuador
eemsta@hotmail.com

Abstract. This study evaluated whether a reproducible Python workflow can strengthen Data Envelopment Analysis in industrial efficiency studies when compared with spreadsheet-based tools such as XlDEA/XIDEA. A methodological, documentary, and computational comparative design was applied. The study examined two implementation environments: a reproducible Python workflow and spreadsheet-based analysis tools. Data were collected through a structured comparison matrix that assessed methodological coverage, automation and scalability, reproducibility and auditability, and second-stage inferential robustness. The analytical procedure reviewed input-oriented CCR estimation, bootstrap inference, Tobit modeling on inefficiency, truncated regression with double bootstrap, and automated report generation. The main result indicates that Python provides a more scalable and auditable architecture for repeated analysis, especially when monthly data, multiple decision-making units, and standardized outputs are required. However, spreadsheet tools remain useful for exploratory applications because they offer greater initial accessibility for non-programming users. The study concludes that Python is preferable for production-grade efficiency analysis, while truncated regression with double bootstrap should guide future second-stage inference when contextual determinants of efficiency are analyzed.

Keywords: Data Envelopment Analysis; Python Workflow; Technical Efficiency; Bootstrap Inference; Truncated Regression.

1. Introduction

Data Envelopment Analysis (DEA) is a non-parametric frontier method used to estimate the relative efficiency of decision-making units (DMUs) that transform multiple inputs into multiple outputs. Its appeal in industrial, energy, and asset-management contexts is that it does not require a predefined production function and can compare heterogeneous operating units as long as the inputs and outputs are conceptually comparable. In the case of monthly industrial datasets, DEA also

allows managers to identify efficient units, quantify efficiency gaps, and organize improvement priorities from a benchmarking perspective.
The practical problem addressed in this article is not limited to DEA estimation. Classical DEA is deterministic and sensitive to sampling variation, outliers, model orientation, and the selection of inputs and outputs. Moreover, applied studies often move from efficiency estimation to second-stage explanation, where contextual variables are related to efficiency or inefficiency scores. This second step creates a methodological challenge because DEA scores are generated estimates rather than directly observed dependent variables. Consequently, conventional regressions may provide misleading standard errors or overstate inferential strength when the first-stage estimation process is ignored.

The relevance of this topic is reinforced by the need for reproducible analytical workflows in industrial efficiency studies. Spreadsheet-based DEA tools, including XlDEA/XIDEA-type environments, are accessible and useful for exploratory analysis, but they may leave part of the analytical trail embedded in manual operations, worksheet structures, or user decisions. A Python workflow, by contrast, can encode data cleaning, model selection, bootstrap parameters, econometric specifications, and reporting rules in a single reproducible script. This is relevant to Sustainable Development Goal 9 because it supports industrial innovation and resilient infrastructure through data-based performance management; it is also linked to Sustainable Development Goal 12 because efficiency analysis contributes to responsible resource use.

In addition to its contribution to Sustainable Development Goal 9, the study emphasizes the practical value of reproducibility for organizations that operate assets under cost, reliability, and availability constraints. A DEA model that is executed manually in separate files can provide useful isolated results, but it is more difficult to verify whether the same assumptions, exclusions, and input-output definitions were preserved across months. A coded workflow, in contrast, can document the analytical rules before the model is executed. This distinction is relevant because efficiency scores may influence maintenance prioritization, replacement decisions, benchmarking exercises, and internal reporting. Therefore, the contribution of the article is located at the intersection of industrial analytics, computational reproducibility, and methodological rigor.

The objective of this study was to adjust and formalize a methodological article in English that evaluates the functional contribution of a reproducible Python workflow for DEA, compares it with spreadsheet-based DEA tools, and assesses the methodological implications of using Tobit regression versus truncated regression with double bootstrap in the second stage. The central argument is that Python improves traceability, scalability, and auditability, while the strongest inferential route for contextual-variable analysis remains the Simar-Wilson truncated regression framework with double bootstrap.

2. Theoretical Framework

DEA was established as a frontier-based method for measuring the relative efficiency of comparable units under multiple inputs and outputs (Charnes et al., 1978). The CCR model assumes constant returns to scale, whereas the BCC model incorporates variable returns to scale and separates pure technical efficiency from scale efficiency (Banker et al., 1984). Later reviews and methodological syntheses show that DEA has evolved from an initial mathematical programming framework into a broad family of models, including radial, non-radial, super-efficiency, cross-efficiency, network, dynamic, and environmental extensions (Seiford & Thrall, 1990; Andersen & Petersen, 1993; Doyle & Green, 1994; Seiford, 1996; Tone, 2001; Cook & Seiford, 2009; Cook et al., 2014).

The design of a DEA study requires explicit decisions regarding the DMU set, the input-output structure, returns to scale, model orientation, weight flexibility, and treatment of non-discretionary variables. The literature warns that poor variable selection, insufficient DMU count, inappropriate aggregation, and uncontrolled outliers can distort the frontier and weaken the managerial interpretation of efficiency scores (Ruggiero, 1998; Dyson et al., 2001; Allen et al., 1997; Podinovski & Thanassoulis, 2007). In this sense, the variables analyzed in this article are not empirical production variables alone; they also include methodological dimensions that affect the credibility of the workflow: coverage, automation, reproducibility, auditability, and inferential robustness.

In an input-oriented CCR model, the score indicates the proportional input reduction that would be required for a unit to reach the efficient frontier while maintaining the observed level of outputs. This orientation is appropriate when the analyst assumes that managers exercise greater control over resources consumed than over outputs generated. In industrial settings, this assumption is common when fuel consumption, maintenance cost, labor hours, or downtime are treated as controllable inputs, whereas energy production or service availability may be constrained by demand, dispatch, or operating schedules. Consequently, the interpretation of efficiency depends on the managerial control boundary defined before the model is solved.

Model specification also requires attention to discrimination power. DEA is sensitive to the ratio between the number of DMUs and the total number of inputs and outputs. When the model contains too many variables relative to the number of observations, many units may appear efficient by construction, reducing the usefulness of the frontier for benchmarking. The theoretical literature therefore recommends parsimonious input-output structures, careful treatment of non-discretionary factors, and sensitivity analysis when the model is used for decision support. These recommendations are especially relevant for monthly industrial panels because the same asset may be observed across periods, but the analyst must

avoid treating repeated observations as independent if operating conditions are structurally different.
Additional foundational sources further support this specification. Cooper et al. (2007) systematize multiplier and envelopment forms, scale assumptions, non-Archimedean elements, slack treatment, and software-based implementations; Cooper et al. (2011) consolidate the state of the art across radial, non-radial, network, dynamic, and application-oriented DEA models; Thanassoulis (2001) emphasizes practical modeling decisions for comparable organizational units and integrated software use; Ray (2004) connects DEA with neoclassical production theory, distance functions, and returns-to-scale interpretation; and Bogetoft and Otto (2011) link DEA with benchmarking practice, statistical analysis, and stochastic frontier alternatives. These contributions reinforce the article's distinction between operational implementation, mathematical programming structure, and the statistical interpretation required when efficiency scores are later used in econometric analysis.

Bootstrap inference provides an additional layer of methodological control because it recognizes that the DEA frontier is estimated from a finite sample. Although deterministic efficiency scores are often interpreted as exact rankings, the frontier can shift when the sample changes or when influential units are removed. Bootstrap procedures approximate the sampling distribution of the estimator and help distinguish robust performance differences from differences that may be small relative to statistical uncertainty. This is important in repeated reports because operational decisions should not be based only on point estimates when confidence intervals overlap. A reproducible Python workflow makes the number of replications, random seed, bias correction rule, and percentile interval calculation explicit, which strengthens the audit trail of the analysis.

The debate on second-stage analysis is also central to the theoretical framework. Tobit regression became popular because DEA scores are bounded, but later studies argued that conventional regression on estimated efficiency scores may be inconsistent when the dependence structure generated by the first stage is ignored. The truncated regression with double bootstrap responds to that concern by incorporating a resampling design that better reflects the statistical properties of DEA estimators. The practical implication is that the Tobit model can be retained for exploratory or managerial diagnosis, but stronger inferential claims about contextual determinants require a procedure specifically designed for two-stage DEA. This article therefore distinguishes between operational usefulness and statistical validity instead of treating all second-stage models as equivalent alternatives.

The growth of DEA has been extensively documented. Citation and application surveys identify DEA as one of the most active branches of efficiency and productivity research, with applications in banking, health care, transportation, education, energy, environment, and industrial operations (Liu et al., 2013a, 2013b;

Emrouznejad & Yang, 2018). In energy and environmental studies, DEA has been used to evaluate resource productivity, undesirable outputs, and environmental performance, which makes the method relevant for organizations seeking to improve operational efficiency while reducing waste and emissions (Zhou et al., 2008; Mardani et al., 2018; Sueyoshi & Goto, 2012).

A critical limitation of deterministic DEA is that efficiency scores are calculated from an estimated frontier and are therefore sensitive to the observed sample. Bootstrap methods were introduced to approximate sampling variation, bias, and confidence intervals in non-parametric frontier models (Simar & Wilson, 1998, 2000). These methods are especially important when small differences among DMUs are interpreted as managerial priorities, or when efficiency scores are compared across months, assets, or operating groups. The bootstrap perspective supports a more cautious reading of efficiency results because a point estimate close to the frontier may not be statistically distinguishable from adjacent units.

Second-stage modeling has generated extensive debate. Tobit regression has been commonly used because DEA scores are bounded, but its appropriateness is not automatic. Hoff (2007) compared alternative second-stage approaches; McDonald (2009) argued that efficiency scores behave more like fractional data than censored observations; and Ramalho et al. (2010) proposed fractional regression as a coherent econometric alternative. Papke and Wooldridge (1996) provide a foundational treatment of fractional response models that is relevant to the bounded-score issue. Banker and Natarajan (2008) established conditions under which two-stage DEA followed by regression can be meaningful, while Simar and Wilson (2007, 2011) warned that naive second-stage regressions may be invalid unless the data-generating process and first-stage estimation are handled explicitly. Daraio and Simar (2005) further emphasized the role of environmental variables in non-parametric frontier analysis.

The truncated regression with double bootstrap is therefore a reference method when the research objective is strict inference about contextual determinants of efficiency. The first bootstrap corrects the bias of DEA efficiency estimates; the second supports inference in the truncated regression stage. This design recognizes that DEA scores are estimated and serially dependent on the sample frontier. By contrast, Tobit on inefficiency can be useful for exploratory management analysis because it is simpler to program and interpret, but it does not fully reproduce the dependence structure generated by the first-stage DEA estimator (Kneip et al., 2008; Simar & Wilson, 2007, 2011).

The computational environment is also part of the theoretical problem because reproducibility depends on the visibility of data transformations, parameters, and analytical choices. Scientific-computing literature highlights the need for transparent code, version control, reproducible workflows, and documented computational environments (Peng, 2011; Sandve et al., 2013; Wilson et al., 2014).

Python supports this requirement through a mature ecosystem for numerical arrays, tabular data, optimization, visualization, and report generation. NumPy, pandas, SciPy, and Matplotlib provide the functional base for matrix construction, data cleaning, linear programming, likelihood optimization, bootstrap simulation, and graphical output (Hunter, 2007; McKinney, 2010; van der Walt et al., 2011; Virtanen et al., 2020). In frontier analysis, Wilson (2008) also demonstrated the value of specialized software for transparent efficiency estimation, reinforcing the broader principle that analytical reproducibility is not merely a technical preference but a methodological requirement.

3. Methodology

This section establishes the methodological, documentary, and computational comparative framework utilized to evaluate Data Envelopment Analysis (DEA) environments. Rather than estimating empirical production frontiers from restricted records, the primary objective is to systematically analyze and validate an automated Python workflow, contrasting its programmable logic against the traditional accessibility of spreadsheet-based tools.

To provide an immediate structural overview of this comparative approach, the complete sequential architecture of the study—spanning from the initial research design to the final quality control protocols—is visually mapped out in the following diagram (see Figure 1).

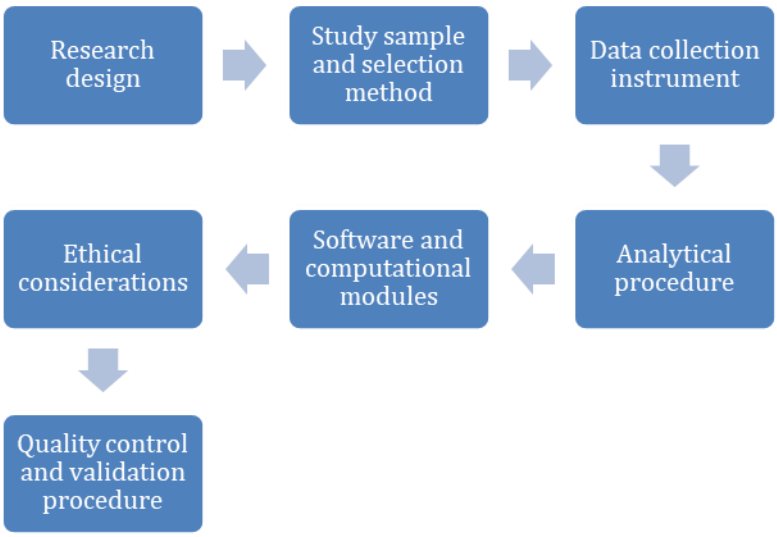

Figure 1. Methodological workflow and structural architecture of the DEA evaluation study.

- Research design: The study used a methodological, documentary, and computational comparative design. It did not estimate a new production

frontier from confidential industrial records; instead, it systematized and evaluated an analytical workflow for DEA implementation in Python and compared it with spreadsheet-oriented DEA environments. The design is methodological because it assesses analytical procedures; documentary because it is grounded in DEA and second-stage econometric literature; and computational because it specifies how the workflow can be operationalized through Python modules.

- Study sample and selection method: The intentional analytical sample comprised two implementation environments: a reproducible Python workflow and spreadsheet-based DEA tools represented by XlDEA/XIDEA-type applications. The unit of analysis was the analytical workflow applied to monthly DMU matrices, where the number of DMUs, inputs, and outputs may vary according to the industrial case. The selection was intentional because the objective was not to generalize from a random software sample but to compare two common implementation logics: programmable reproducibility and spreadsheet accessibility.

- Data collection instrument: A structured comparison matrix was used as the instrument. The matrix included four dimensions: methodological coverage, automation and scalability, reproducibility and auditability, and inferential robustness of the second stage. Its criteria and rating structure are included in Appendix 1, and the instrument was used to organize the comparison between Python, XlDEA/XIDEA-type tools, Tobit modeling, and truncated regression with double bootstrap.

- Analytical procedure: The proposed Python workflow follows six sequential stages. First, input files are loaded and standardized by period. Second, data validation checks numeric types, missing values, inconsistent records, and the comparability of DMUs. Third, input and output matrices are constructed as X and Y. Fourth, for each DMU, the input-oriented CCR linear program minimizes theta_i subject to Ylambda >= y_i, theta_i x_i - Xlambda >= 0, and lambda >= 0. Fifth, bootstrap resampling estimates bias and confidence intervals for DEA scores. Sixth, the second stage models inefficiency as u_i = 1 - theta_i through Tobit as an operational approximation and evaluates truncated regression with double bootstrap as the recommended inferential extension.

- Software and computational modules: The workflow can be implemented using pathlib or os for paths, NumPy for numerical arrays, pandas for tabular processing, SciPy for linear programming and likelihood optimization, Matplotlib for visualization, openpyxl for Excel reporting, patsy for formula-based design matrices, and datetime for temporal labels. Optional interactive execution may use tkinter for folder selection. These modules permit both batch automation and manual review of outputs.

- Ethical considerations: The study does not involve human subjects, personal data, or intervention with participants. When applied to industrial datasets, the workflow should anonymize asset identifiers when required, restrict access to confidential operational records, document every transformation, and preserve reproducible logs. The method also requires transparent reporting of exclusions, such as DMUs without valid operation or records that do not meet comparability criteria.

- Quality control and validation procedure: The computational workflow proposed in the article also contains validation controls: verification of numeric variables before DEA estimation, exclusion logs for invalid DMUs, solver-status review after each linear-programming run, consistency checks between exported tables and graphs, and traceable naming of period folders. These controls are necessary because automated analysis can reproduce errors as efficiently as it reproduces valid procedures. For that reason, automation must be accompanied by documented validation rules, especially when outputs are intended for managerial review or academic dissemination.

4. Results

The comparative analysis indicates that Python and spreadsheet-based DEA tools serve different but complementary analytical purposes. Python is stronger when the organization requires repeated execution, auditable code, parameter control, batch processing, and a path toward robust second-stage inference. XlDEA/XIDEA-type tools remain useful when the priority is accessibility, rapid exploration, and a graphical interface for users who do not program.

Regarding methodological coverage, the Python workflow integrates DEA estimation, bootstrap inference, second-stage modeling, and standardized reports in one controlled environment. Spreadsheet-based tools usually perform the main DEA estimation efficiently, but their extension toward nested bootstrap procedures or econometric second-stage analysis tends to require additional files, manual steps, macros, or external software. Therefore, Python offers a more continuous route from estimation to inference.

Regarding automation and scalability, Python reduces the probability of inconsistent execution across monthly files. It can iterate over periods, validate inputs, execute linear programs, export tables, and consolidate annual results using the same rules. In spreadsheet tools, scalability depends more strongly on user discipline, worksheet structure, and template stability. A single modified range, renamed column, or overwritten formula may alter the reproducibility of the analysis.

Regarding reproducibility and auditability, Python stores assumptions in code: model orientation, returns to scale, tolerance values, number of bootstrap replications, exclusion rules, and output structure. This increases the possibility of audit, peer review, and future replication. In graphical spreadsheet environments, some assumptions may remain implicit in manual operations, creating a risk when the analysis must be repeated by another analyst or defended in a formal review.

Regarding second-stage inference, Tobit on inefficiency offers a practical first approximation when the objective is managerial diagnosis. However, truncated regression with double bootstrap provides a more defensible inferential framework when the research objective is to evaluate determinants of efficiency. The result is not that Tobit is useless, but that its role should be limited to exploratory or operational contexts unless the underlying assumptions are explicitly justified.

The comparative synthesis is presented in Table 1 and Table 2.

Table 1. Functional comparison between the reproducible Python workflow and XlDEA/XIDEA-type tools

Criterion	Reproducible Python workflow	XlDEA/XIDEA-type tools	Analytical implication
Logic of use	Programmable, parameterized, and versionable workflow	Graphical interface centered on spreadsheet operations	Python favors standardization; spreadsheet tools favor initial accessibility
Methodological coverage	Integrates CCR DEA, bootstrap, Tobit, reporting, and future double-bootstrap truncated regression	Facilitates DEA estimation, but second-stage inference normally requires external procedures or additional manual work	Python provides better continuity between estimation and inference
Automation	Processes multiple periods and generates standardized outputs automatically	Repetition depends on templates, macros, and user discipline	Python reduces manual intervention and execution variability
Scalability	Suitable for monthly series, many DMUs, and intensive resampling	May become sensitive to changes in worksheets, ranges, or file structure	Python is preferable for recurrent industrial studies
Reproducibility and audit	Assumptions, exclusion rules, parameters, and	Part of the procedure may remain implicit	Python improves auditability and

	outputs are explicit in code	in user actions	replication
Entry barrier	Requires programming and environment management	More accessible to non-programmers	Spreadsheet tools remain useful for exploratory analysis

Table 2. Methodological comparison between Tobit and truncated regression with double bootstrap

Criterion	Tobit on inefficiency	Truncated regression with double bootstrap	Methodological implication
Dependent variable	Uses u_i = 1 - theta_i and treats inefficiency as censored at zero.	Models DEA scores or transformations on a truncated support.	The truncated alternative aligns better with the generated-score nature of DEA.
Treatment of uncertainty	May use parameter bootstrap but does not fully reproduce the first-stage dependence.	Combines DEA bias correction and second-stage inference through nested bootstrap logic.	Double bootstrap provides stronger inference.
Computational burden	Moderate and appropriate for operational routines.	High, requiring many replications and careful numerical control.	There is a trade-off between rigor and computational cost.
Implementation	Relatively simple with standard likelihood optimization.	More complex and methodologically demanding.	Tobit facilitates initial adoption; truncated regression requires specialization.
Preferred use	Exploratory analysis, management diagnosis, and complementary second-stage interpretation.	Research-grade inference on contextual variables and determinants of efficiency.	The choice depends on the analytical objective.

5. Discussion

The findings show that the distinction between operational usefulness and inferential rigor is central. Spreadsheet-based tools are valuable because they lower the entry barrier and allow quick DEA exploration. This matters in organizations where efficiency analysis is new or where users need immediate benchmarking outputs. Nevertheless, when the analytical process is repeated monthly, incorporated into reports, or used to support research conclusions, manual spreadsheet operations may become a source of methodological risk.

The Python workflow strengthens the chain of evidence because it links data preparation, DEA estimation, bootstrap inference, second-stage modeling, and report generation. This integration is particularly important in industrial applications where multiple assets or operating units are evaluated over time. In such cases, the main requirement is not only to calculate an efficiency score, but also to know how the score was produced, which data were excluded, which assumptions were applied, and how uncertainty was quantified.

The comparison between Tobit and truncated regression with double bootstrap also clarifies the methodological boundary of the article. Tobit is easier to implement and can be useful for preliminary diagnosis; however, its censored-data logic does not fully match the nature of DEA scores. Truncated regression with double bootstrap is computationally heavier, but it better reflects the fact that DEA scores are generated by a first-stage frontier estimator. For this reason, the adjusted article recommends Tobit for operational exploration and the double-bootstrap truncated model for research-grade inference.

The main limitation is that the article is methodological and comparative rather than a full empirical replication across software environments. It does not claim universal superiority of Python, nor does it prove numerical equivalence between Python and XlDEA/XIDEA using the same dataset. A future empirical extension should process an identical dataset in both environments, compare point estimates, quantify differences due to numerical tolerance and model configuration, and implement the full Simar-Wilson double-bootstrap algorithm.

Future research should extend the workflow toward a full empirical application with monthly industrial data, multiple assets, and contextual variables such as operating hours, load factor, ambient conditions, maintenance category, and fuel quality. Such an application would allow the analyst to compare point estimates, bootstrap confidence intervals, bias-corrected scores, and second-stage coefficients under alternative specifications. It would also make it possible to evaluate the computational cost of nested resampling and to define practical thresholds for the number of bootstrap replications. In addition, a future version should include a reproducibility package with anonymized data, source code, and execution instructions so that reviewers can replicate the main tables and verify the analytical decisions made by the researcher.

6. Conclusion

The adjusted manuscript concludes that a reproducible Python workflow constitutes a robust alternative for DEA studies that require repeated execution, monthly processing, traceable assumptions, and standardized reporting. The first contribution is operational: Python reduces manual manipulation, codifies cleaning rules, preserves the analytical sequence, and supports batch processing across periods and decision-making units. The second contribution is methodological: the

same environment can integrate input-oriented CCR DEA, bootstrap inference, Tobit modeling, and future migration toward truncated regression with double bootstrap.

This last approach is recommended when the study seeks stronger inference about contextual variables because DEA scores are estimated rather than directly observed. The comparison also identifies limitations. Python requires programming skills, dependency management, numerical validation, and version control discipline. XlDEA/XIDEA remains valuable for teaching, exploratory diagnosis, and users who prioritize accessibility over automation. Therefore, the choice of tool should not be framed as universal software superiority but as alignment between analytical purpose and methodological demand. For industrial efficiency studies associated with energy productivity, asset management, and sustainable innovation, the preferred route is a staged implementation: spreadsheet tools for initial familiarization, Python for production-grade analysis, and double-bootstrap truncated regression for rigorous second-stage research.

7. References

Allen, R., Athanassopoulos, A., Dyson, R. G., & Thanassoulis, E. (1997). Weights restrictions and value judgements in data envelopment analysis: Evolution, development and future directions. Annals of Operations Research, 73, 13-34. https://doi.org/10.1023/A:1018968909638

Andersen, P., & Petersen, N. C. (1993). A procedure for ranking efficient units in data envelopment analysis. Management Science, 39(10), 1261-1264. https://doi.org/10.1287/mnsc.39.10.1261

Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. Management Science, 30(9), 1078-1092. https://doi.org/10.1287/mnsc.30.9.1078

Banker, R. D., & Natarajan, R. (2008). Evaluating contextual variables affecting productivity using data envelopment analysis. Operations Research, 56(1), 48-58. https://doi.org/10.1287/opre.1070.0460

Bogetoft, P., & Otto, L. (2011). Benchmarking with DEA, SFA, and R. Springer. https://doi.org/10.1007/978-1-4419-7961-2

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. European Journal of Operational Research, 2(6), 429-444. https://doi.org/10.1016/0377-2217(78)90138-8

Cook, W. D., & Seiford, L. M. (2009). Data envelopment analysis (DEA) - Thirty years on. European Journal of Operational Research, 192(1), 1-17. https://doi.org/10.1016/j.ejor.2008.01.032

Cook, W. D., Tone, K., & Zhu, J. (2014). Data envelopment analysis: Prior to choosing a model. Omega, 44, 1-4. https://doi.org/10.1016/j.omega.2013.09.004

Cooper, W. W., Seiford, L. M., & Tone, K. (2007). Data envelopment analysis: A comprehensive text with models, applications, references and DEA-solver software (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-45283-8

Cooper, W. W., Seiford, L. M., & Zhu, J. (Eds.). (2011). Handbook on data envelopment analysis (2nd ed.). Springer. https://doi.org/10.1007/978-1-4419-6151-8

Daraio, C., & Simar, L. (2005). Introducing environmental variables in nonparametric frontier models: A probabilistic approach. Journal of Productivity Analysis, 24(1), 93-121. https://doi.org/10.1007/s11123-005-3042-8

Doyle, J., & Green, R. (1994). Efficiency and cross-efficiency in DEA: Derivations, meanings and uses. Journal of the Operational Research Society, 45(5), 567-578. https://doi.org/10.1057/jors.1994.84

Dyson, R. G., Allen, R., Camanho, A. S., Podinovski, V. V., Sarrico, C. S., & Shale, E. A. (2001). Pitfalls and protocols in DEA. European Journal of Operational Research, 132(2), 245-259. https://doi.org/10.1016/S0377-2217(00)00149-1

Emrouznejad, A., & Yang, G. L. (2018). A survey and analysis of the first 40 years of scholarly literature in DEA: 1978-2016. Socio-Economic Planning Sciences, 61, 4-8. https://doi.org/10.1016/j.seps.2017.01.008

Gattoufi, S., Oral, M., & Reisman, A. (2004). A taxonomy for data envelopment analysis. Socio-Economic Planning Sciences, 38(2-3), 141-158. https://doi.org/10.1016/S0038-0121(03)00022-3

Hoff, A. (2007). Second stage DEA: Comparison of approaches for modelling the DEA score. European Journal of Operational Research, 181(1), 425-435. https://doi.org/10.1016/j.ejor.2006.05.019

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95. https://doi.org/10.1109/MCSE.2007.55

Kneip, A., Simar, L., & Wilson, P. W. (2008). Asymptotics and consistent bootstraps for DEA estimators in nonparametric frontier models. Econometric Theory, 24(6), 1663-1697. https://doi.org/10.1017/S026646660808065X

Liu, J. S., Lu, L. Y. Y., Lu, W. M., & Lin, B. J. Y. (2013a). Data envelopment analysis 1978-2010: A citation-based literature survey. Omega, 41(1), 3-15. https://doi.org/10.1016/j.omega.2010.12.006

Liu, J. S., Lu, L. Y. Y., Lu, W. M., & Lin, B. J. Y. (2013b). A survey of DEA applications. Omega, 41(5), 893-902. https://doi.org/10.1016/j.omega.2012.11.004

Mardani, A., Streimikiene, D., Balezentis, T., Saman, M. Z. M., Nor, K. M., & Khoshnava, S. M. (2018). Data envelopment analysis in energy and environmental economics: An overview of the state-of-the-art and recent development trends. Energies, 11(8), Article 2002. https://doi.org/10.3390/en11082002

McDonald, J. (2009). Using least squares and Tobit in second stage DEA efficiency analyses. European Journal of Operational Research, 197(2), 792-798. https://doi.org/10.1016/j.ejor.2008.07.039

McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, 56-61. https://conference.scipy.org/proceedings/scipy2010/mckinney.html

Papke, L. E., & Wooldridge, J. M. (1996). Econometric methods for fractional response variables with an application to 401(k) plan participation rates. Journal of Applied Econometrics, 11(6), 619-632. https://doi.org/10.1002/(SICI)1099-1255(199611)11:6<619::AID-JAE418>3.0.CO;2-1

Peng, R. D. (2011). Reproducible research in computational science. Science, 334(6060), 1226-1227. https://doi.org/10.1126/science.1213847

Podinovski, V. V., & Thanassoulis, E. (2007). Improving discrimination in data envelopment analysis: Some practical suggestions. Journal of Productivity Analysis, 28(1-2), 117-126. https://doi.org/10.1007/s11123-007-0045-x

Ramalho, E. A., Ramalho, J. J. S., & Henriques, P. D. (2010). Fractional regression models for second stage DEA efficiency analyses. Journal of Productivity Analysis, 34(3), 239-255. https://doi.org/10.1007/s11123-010-0184-0

Ray, S. C. (2004). Data envelopment analysis: Theory and techniques for economics and operations research. Cambridge University Press. https://doi.org/10.1017/CBO9780511606731

Ruggiero, J. (1998). Non-discretionary inputs in data envelopment analysis. European Journal of Operational Research, 111(3), 461-469. https://doi.org/10.1016/S0377-2217(97)00306-8

Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. PLoS Computational Biology, 9(10), e1003285. https://doi.org/10.1371/journal.pcbi.1003285

Seiford, L. M., & Thrall, R. M. (1990). Recent developments in DEA: The mathematical programming approach to frontier analysis. Journal of Econometrics, 46(1-2), 7-38. https://doi.org/10.1016/0304-4076(90)90045-U

Seiford, L. M. (1996). Data envelopment analysis: The evolution of the state of the art (1978-1995). Journal of Productivity Analysis, 7(2-3), 99-137. https://doi.org/10.1007/BF00157037

Simar, L., & Wilson, P. W. (1998). Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models. Management Science, 44(1), 49-61. https://doi.org/10.1287/mnsc.44.1.49

Simar, L., & Wilson, P. W. (2000). A general methodology for bootstrapping in non-parametric frontier models. Journal of Applied Statistics, 27(6), 779-802. https://doi.org/10.1080/02664760050081951

Simar, L., & Wilson, P. W. (2007). Estimation and inference in two-stage, semi-parametric models of production processes. Journal of Econometrics, 136(1), 31-64. https://doi.org/10.1016/j.jeconom.2005.07.009

Simar, L., & Wilson, P. W. (2011). Two-stage DEA: Caveat emptor. Journal of Productivity Analysis, 36(2), 205-218. https://doi.org/10.1007/s11123-011-0230-6

Sueyoshi, T., & Goto, M. (2012). DEA environmental assessment: Comparison between public and private ownership in petroleum industry. European Journal of Operational Research, 216(3), 668-678. https://doi.org/10.1016/j.ejor.2011.07.046

Thanassoulis, E. (2001). Introduction to the theory and application of data envelopment analysis: A foundation text with integrated software. Springer. https://doi.org/10.1007/978-1-4615-1407-7

Tobin, J. (1958). Estimation of relationships for limited dependent variables. Econometrica, 26(1), 24-36. https://doi.org/10.2307/1907382

Tone, K. (2001). A slacks-based measure of efficiency in data envelopment analysis. European Journal of Operational Research, 130(3), 498-509. https://doi.org/10.1016/S0377-2217(99)00407-5

United Nations. (2015). Transforming our world: The 2030 agenda for sustainable development. United Nations General Assembly. https://sdgs.un.org/2030agenda

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. Computing in Science & Engineering, 13(2), 22-30. https://doi.org/10.1109/MCSE.2011.37

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods, 17, 261-272. https://doi.org/10.1038/s41592-019-0686-2

Wilson, P. W. (2008). FEAR: A software package for frontier efficiency analysis with R. Socio-Economic Planning Sciences, 42(4), 247-254. https://doi.org/10.1016/j.seps.2007.02.001

Wilson, G., Aruliah, D. A., Brown, C. T., Hong, N. P. C., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P., & Wilson, P. (2014). Best practices for scientific computing. PLoS Biology, 12(1), e1001745. https://doi.org/10.1371/journal.pbio.1001745

Zhou, P., Ang, B. W., & Poh, K. L. (2008). A survey of data envelopment analysis in energy and environmental studies. European Journal of Operational Research, 189(1), 1-18. https://doi.org/10.1016/j.ejor.2007.04.042

Appendix 1

Structured Comparison Instrument

The following instrument was used to organize the documentary and computational comparison reported in the methodology section. Each criterion was assessed qualitatively as high, medium, or low compliance according to the evidence available in the reviewed workflow and implementation environment.

Table A1: Structured comparison instrument

Dimension	Criterion	Evidence reviewed	Assessment scale
Methodological coverage	Ability to integrate DEA estimation, bootstrap inference, second-stage	Workflow steps, supported models, analytical outputs, and reporting	High / Medium / Low

	modeling, and report generation.	structure.	
Automation and scalability	Ability to process monthly files, multiple decision-making units, and repeated analytical cycles.	Batch execution capacity, file handling logic, and repeatability across periods.	High / Medium / Low
Reproducibility and auditability	Visibility of assumptions, model orientation, parameters, exclusion rules, and output generation logic.	Code structure, documented parameters, versionability, and traceability of transformations.	High / Medium / Low
Inferential robustness	Adequacy of Tobit, bootstrap, and truncated regression with double bootstrap for second-stage interpretation.	Econometric assumptions, dependence of estimated scores, bias correction, and confidence-interval logic.	High / Medium / Low
Operational accessibility	Ease of use for analysts with and without programming experience.	Interface requirements, programming burden, template maintenance, and user training needs.	High / Medium / Low