2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #50294

A Conceptual Framework for Empowering Engineering Students Worldwide Through a Python-Based Statistics Course

Dr. Edris Ebrahimzadeh, University of South Alabama

I am an Assistant Professor of Instruction in the Chemical and Biomolecular Engineering department of the University of South Alabama, and a Registered Professional Engineer (P.E.). My primary research interest is Engineering Education.

©American Society for Engineering Education, 2026

A Conceptual Framework for Empowering Engineering Students Worldwide Through a Python-Based Statistics Course

Abstract
Engineering students play a pivotal role in the creation and enhancement of innovative products, manufacturing systems, and processes, necessitating a strong foundation in statistical methods. These methods offer engineers both descriptive and analytical tools to address variability in observed data. Despite the significance of statistics in engineering, it is often challenging for students to incorporate a comprehensive statistics course into their curricula due to other academic requirements. Consequently, many engineering programs lack dedicated courses in engineering statistics [1].

In response to this gap, our paper presents a unique approach that can be effectively delivered as a freshman engineering course. While the statistical techniques presented are foundational across various disciplines such as business, management, life sciences, and social sciences, our focus remains on catering to an engineering-oriented audience. This targeted approach aims to empower engineering students to harness statistics for a myriad of applications within their field.

One distinctive aspect of our proposed course is the utilization of Python, a freely available programming language widely embraced in all engineering disciplines. By employing Python, we not only provide an accessible and cost-effective alternative to licensed statistical packages but also extend a valuable advantage to students in developing countries who may face challenges accessing proprietary software.

This paper is conceptual in nature and focuses on course design, curricular integration, and instructional examples rather than reporting empirical learning outcomes.

Keywords: engineering statistics, Python, first-year engineering, statistical analysis

1. Introduction

Engineering is a global discipline requiring professionals to analyze, design, and optimize systems while managing uncertainty. A strong foundation in statistical methods enables engineers to handle variability in observed data, improving decision-making and problem-solving capabilities. However, many engineering programs worldwide do not offer standalone courses in engineering statistics, leaving students with limited exposure to essential statistical concepts [2].

Recognizing this gap, Hoerl and Snee [3] proposed "Statistical Engineering" as a discipline bridging statistical thinking and applied engineering tools. Numerous studies [4-12] have explored innovative ways to enhance engineering statistics education, including gamification, software integration, and active learning strategies.

This paper advocates for a practical, application-driven approach to teaching introductory engineering statistics using Python. Unlike traditional probability-heavy courses, our curriculum emphasizes applied statistics for real-world engineering problem-solving. By incorporating Python, students gain hands-on experience in statistical computing, fostering skills applicable across international educational settings. The course is designed to be adaptable for diverse institutions, including those in developing regions where access to proprietary software may be limited.

2. Scope and Contribution of This Work

This paper presents a conceptual framework and instructional model for an applied Python-based engineering statistics course intended for first-year engineering students. Rather than reporting empirical assessment data, the paper focuses on curricular motivation, course structure, and representative instructional examples drawn from multiple engineering disciplines. The goal is to provide a transferable course model that institutions can adapt to their local curricular constraints, particularly in settings where access to proprietary software or additional credit hours is limited.

3. Conceptual and Design-Based Research Approach

This work follows a conceptual, design-based approach to curriculum development commonly used in engineering education scholarships. Rather than reporting empirical learning outcomes, the paper focuses on articulating the motivation, structure, and instructional design of a Python-based engineering statistics course. Representative examples are used to illustrate how core statistical concepts can be taught through applied, discipline-relevant problems. This approach is intended to support transferability across institutions and to inform future implementations that may include formal assessment and evaluation.

4. Course Description

This course is designed as an applied statistics course rather than a theoretical probability course. The structure is as follows:

- **Week 1:** Introduction to statistical applications in engineering, empirical modeling, experimental design, and process monitoring.
- **Weeks 2-4:** Probability concepts with an emphasis on the normal distribution.
- **Weeks 5-11:** Confidence intervals, hypothesis testing, and their applications in engineering.
- **Weeks 12-13:** Regression models and predictive analytics.

- **Weeks 14-15:** Statistical process control, including Shewhart control charts.

Each topic is reinforced through practical engineering problems drawn from different disciplines. Python is used as the primary tool for statistical analysis, ensuring that students across international institutions can access and apply statistical methods without requiring expensive software.

The primary instructional goals of the proposed course are to:

- Develop foundational statistical reasoning skills relevant to engineering applications.
- Enable students to analyze and visualize engineering data using Python.
- Introducing uncertainty, variability, and inference as essential components of engineering decision-making.
- Prepare students for reinforcement of statistical concepts in later laboratory, design, and discipline-specific courses.

5. Curricular Integration Considerations

Many engineering programs already require a freshman-level programming or problem-solving course, commonly using Python or MATLAB. The proposed course is intended to function as either a replacement for, or a co-requisite alongside, such introductory courses rather than as an additional standalone requirement. In this model, statistical reasoning is introduced concurrently with programming skills, allowing students to immediately apply computational tools to data-driven engineering problems.

Because the course is positioned early in the curriculum, it is not intended to provide complete statistical mastery. Instead, it establishes foundational concepts that can be reinforced in later laboratory, design, and discipline-specific courses. This approach aligns with common curricular practice in which skills introduced in the first year are revisited and expanded throughout the program.

By positioning the course as an early foundational experience rather than a terminal statistics course, the model supports long-term skill development through reinforcement in subsequent engineering courses.

6. Sample Course Exercises

Unlike traditional service-based probability and statistics courses that emphasize mathematical derivations and generalized examples, the proposed course prioritizes engineering-context problems, computational implementation, and interpretation of results. While core topics such as probability distributions, confidence intervals, hypothesis testing, and regression overlap with traditional courses, the emphasis shifts toward application, data analysis, and decision-making rather than analytical proof.

6.1. Counting Techniques and Binomial Random Variables

In statistics, counting techniques play a crucial role in solving various problems. A fundamental concept related to counting techniques is the Bernoulli trial, a trial with only two possible outcomes. The random variable X, representing the number of trials resulting in success, follows a binomial distribution with parameters $0 < p < 1$ and $n = 1, 2, \ldots$ The probability mass function of X is given by:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x} \qquad \text{Eq. 1}$$

Python's `comb` function can be employed to solve problems involving binomial random variables. For instance, consider a scenario where each sample of water has a 10% chance of containing a particular organic pollutant. To determine the probability that at least four samples contain the pollutant, the following sum needs to be calculated.

$$P(X \geq 4) = \sum_{x=4}^{18} \binom{18}{x} (0.1)^x (0.9)^{18-x} \qquad \text{Eq. 2}$$

This problem can be efficiently solved using a simple iteration structure in Python. Binomial random variables are versatile tools for modeling various physical systems, and the binomial probability mass function allows us to compute probabilities for these models.

6.2. Probability Density Function

Density functions are frequently employed in engineering to characterize physical systems. Consider the density of a loading on a long, thin beam, as depicted in Figure 1. For any point x along the beam, the density is described by a function (in grams/cm). Large loadings correspond to large values for the function. The total loading between points a and b is determined by the integral of the density function from a to b, representing the area under the density function over this interval.

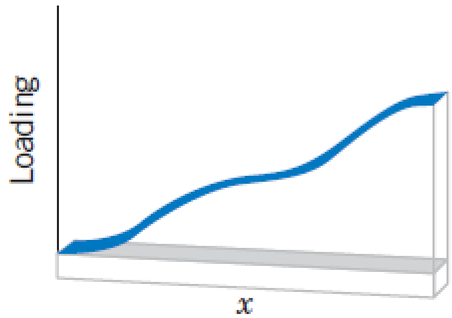

Figure 1. Density function of a loading on a long, thin beam.

Now, let the continuous random variable X represent the diameter of a hole drilled in a sheet metal component. The target diameter is 12.5 millimeters, and historical data suggest that the distribution of X can be modeled by a probability density function:

$$f(x) = 20e^{-20(x-12.5)}, x \geq 12.5 \qquad \text{Eq. 3}$$

Suppose a part with a diameter larger than 12.60 millimeters is considered scrapped. What proportion of parts is scrapped? The density function can be easily plotted using Python.

$$P(X > 12.60) = \int_{12.6}^{\infty} f(x)dx = \int_{12.6}^{\infty} 20e^{-20(x-12.5)}\, dx \qquad \text{Eq. 4}$$

While evaluating this integral may be challenging without calculus knowledge, Python can handle it efficiently.

Furthermore, to find the mean and variance of X:

$$E(X) = \int_{12.5}^{\infty} xf(x)dx = \int_{12.5}^{\infty} x \cdot 20e^{-20(x-12.5)}\, dx \qquad \text{Eq. 5}$$

$$V(X) = \int_{12.5}^{\infty} \left(x - E(X)\right)^2 f(x)dx \qquad \text{Eq. 6}$$

Evaluation of these integrals typically requires knowledge of integration by parts. However, Python can perform these calculations without the need for explicit integration techniques.

6.3. Normal Probability Density Function

Undoubtedly, the most widely used model for the distribution of a random variable is the normal distribution. The normal probability distribution function can be expressed as:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad -\infty < x < \infty \qquad \text{Eq. 9}$$

To calculate probabilities for a normal random variable, one must find the area under the normal probability distribution function. Python codes can be utilized to plot the normal probability density function, as illustrated in Figure 2.

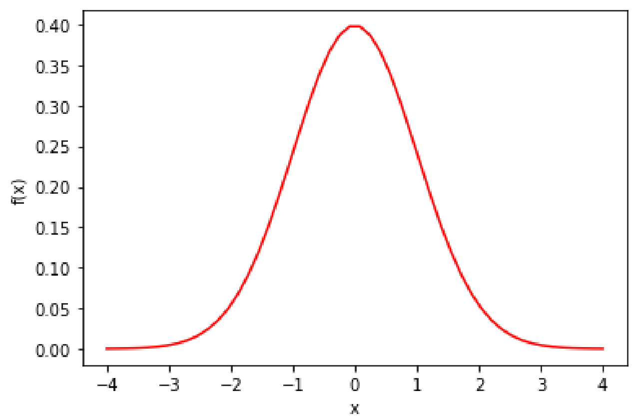

Figure 2. Standard Normal Probability Function

The graph provides insights into various investigations, such as:

a) The impact of standard deviation on the shape of the curve.
b) Exploring the 68-95-99.7% rule and employing the cumulative integration technique discussed in binomial distribution.

While tables offer cumulative probabilities for a standard normal random variable, calculating the area between two random variables traditionally involves extensive calculations. However, Python codes can efficiently evaluate these areas without manual calculations or reliance on statistical tables.

For example, consider the diameter of a shaft in an optical storage drive, which follows a normal distribution with a mean of 0.2508 inches and a standard deviation of 0.0005 inches. The specifications on the shaft are 0.2500 ± 0.0015 inches. The proportion of shafts conforming to specifications, denoted by X can be calculated as

$$P(0.2485 < X < 0.2515) = \cdots = 0.91924$$

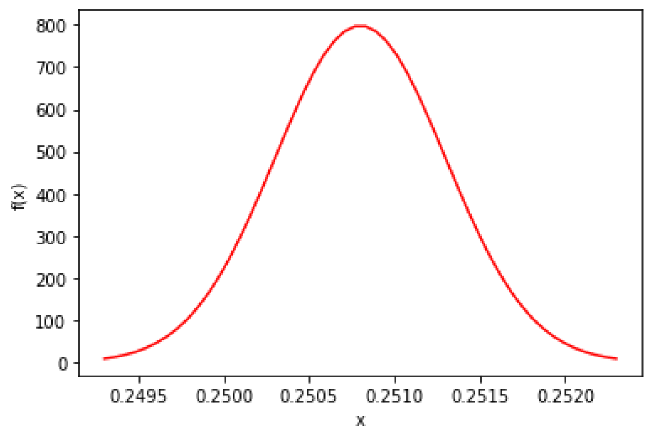

Figure 3. Normal Probability Function for the shaft case study.

It's noteworthy that most of the nonconforming shafts are too large, given that the process mean is located very near the upper specification limit. Students can further explore the impact of recentering the process by placing the process mean at 0.25 and observing its effect on the process yield.

6.4. Box Plots

A box plot visually represents the three quartiles, the minimum, and the maximum of a dataset within a rectangular box, either horizontally or vertically. The box spans the interquartile range, with the left edge at the first quartile q_1 and the right edge at the third quartile q_3.

In Table 1, the compressive strengths (in psi) of 80 specimens of a new aluminum-lithium alloy are recorded in the order of testing. The raw data format may not provide clear insights into the compressive strength distribution. Questions like "What percent of the specimens fail below 120 psi?" are challenging to answer without visualization. Constructing a boxplot using Python can effectively convey this information.

Table 1. Compressive Strength (in psi) of 80 Aluminum-Lithium Alloy Specimens

105	221	183	186	121	181	180	143
97	154	153	174	120	168	167	141
245	228	174	199	181	158	176	110
163	131	154	115	160	208	158	133
207	180	190	193	194	133	156	123
134	178	76	167	184	135	229	146
218	157	101	171	165	172	158	169
199	151	142	163	145	171	148	158
160	175	149	87	160	237	150	135
196	201	200	176	150	170	118	149

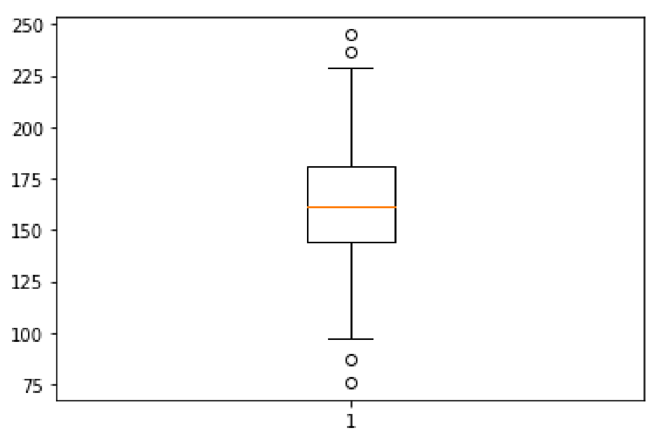

Figure 4. Boxplot of the aluminum-lithium alloy case study

This box plot reveals that the distribution of compressive strengths is fairly symmetric around the central value. The lengths of the left and right boxes around the median, along with the whiskers, are approximately equal. Additionally, there are two mild outliers at lower strength and two at higher strength.

6.5. Parameter Estimation for the Mean

Engineers often engage in parameter estimation practices, such as the Charpy V-notch method outlined in ASTM Standard E23 for notched bar impact testing of metallic materials. This method, utilizing impact energy measurements, assesses whether a material undergoes a ductile-to-brittle transition with decreasing temperature.

Consider a scenario where tests on a sample of 10 specimens are conducted using this procedure. While the sample average $\bar{X}$ serves as an estimate for the true mean impact energy (symbolized as μ), it is acknowledged that the true mean impact energy is unlikely to precisely match this estimate. Conveying the test results as a singular value lacks appeal, as it fails to provide any indication of the proximity to μ. The estimate could be in close alignment with the true mean or significantly deviate from it. To address this, it is prudent to present the estimate in the form of a confidence interval-a range of plausible values that offers a more comprehensive depiction of the uncertainty associated with the estimate.

When constructing confidence intervals on the mean μ of a population, it is important to ensure that the population is normally distributed. In practical conditions where the variation of the population (σ^2) is unknown, and based on the Central Limit Theorem, if the sample size is sufficiently large ($n > 40$), substituting the variation of the sample (s^2) for the variation of the population (σ^2) results in little error in the test procedure, regardless of the shape of the distribution of the parameter.

However, for smaller sample sizes ($n < 40$), replacing the variation of the population with the variation of the sample could result in large errors if the probability distribution of the population is not normal. In such cases, plotting the Normal probability plot based on the observations and performing linear regression can help assess the normality of the population.

For example, an article in the journal of Materials Engineering describes the results of tensile adhesion tests on 22 U-700 alloy specimens (Table 2). The load at specimen failure is recorded. Figure 5 and Figure 6 show a normal probability plot and a box plot of the tensile adhesion test data, respectively. These displays provide good support for the assumption that the population is normally distributed.

Table 2. Results of tensile adhesion tests on 22 U-700 alloy specimen.

19.8	10.1	14.9	7.5	15.4	15.4
15.4	18.5	7.9	12.7	11.9	11.4
11.4	14.1	17.6	16.7	15.8	
19.5	8.8	13.6	11.9	11.4	

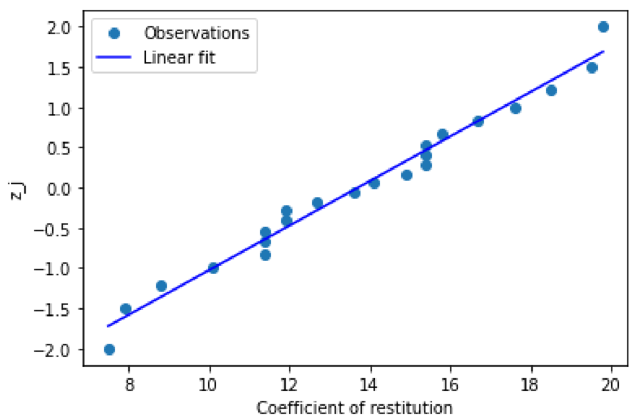

Figure 5. Normal probability plot of tensile adhesion test case study

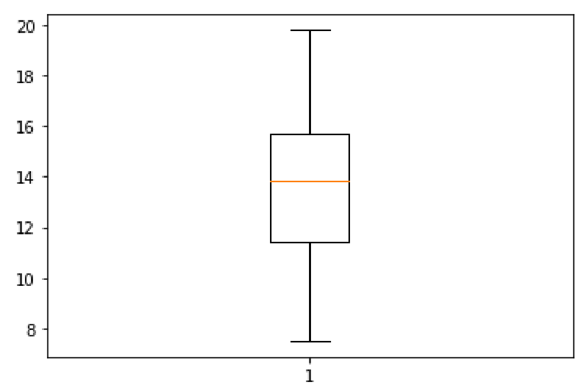

Figure 6. Boxplot of tensile adhesion test case study

Using Python, the 95% confidence interval for the population mean μ is easily calculated as:

$$12.14 \leq \mu \leq 15.29$$

The confidence interval is relatively wide due to the considerable variability in the tensile adhesion test measurements. A larger sample size would have led to a shorter interval.

6.6. Hypothesis Testing of Two Population Means

Engineers frequently encounter situations where decisions must be made between conflicting statements regarding the numerical value of a parameter. Take, for instance, an engineer designing an air crew escape system with an ejection seat powered by a rocket motor. The burning rate of the rocket motor's propellant is critical, ideally at a mean of 50 cm/sec for proper functionality. Deviations from this value pose risks-too low may result in malfunction, while too high can lead to instability and potential harm to the pilot. To address such questions, engineers employ a statistical technique known as hypothesis testing.

In a practical example, two catalysts are analyzed to determine their impact on the mean yield of a chemical process. Catalyst 1 is currently in use, and catalyst 2 is being considered for its cost-effectiveness. The goal is to adopt catalyst 2 if it does not significantly change the process yield. The test results, shown in Table 3, prompt the question: Is there any difference between the mean yields?

Table 3. Catalyst Yield Data

Observation Number	**Catalyst 1**	**Catalyst 2**
1	91.50	89.19
2	94.18	90.95
3	92.18	90.46
4	95.39	93.21
5	91.79	97.19
6	89.07	97.04
7	94.72	91.07
8	89.21	92.75

The parameters of interest are μ_1 and μ_2 representing the mean process yield using catalysts 1 and 2, respectively. The hypothesis setup is:

- Null hypothesis: $H_0: \mu_1 = \mu_2$
- Alternate hypothesis: $H_A: \mu_1 \neq \mu_2$

Using Python, P −value is 0.729. Since the P −value exceeds $\alpha = 0.05$, the null hypothesis cannot be rejected. Therefore, there is no strong evidence to conclude that catalyst 2 results in a mean yield different from the mean yield when catalyst 1 is used. Figure 7 and Figure 8 display the normal probability plot of the two samples of yield data and comparative box plots, indicating no issues with the normality assumption.

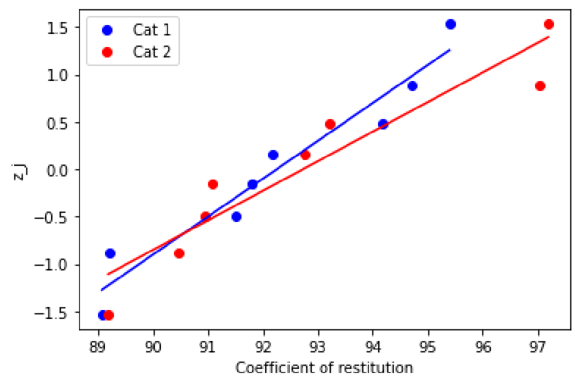

Figure 7. Normal probability plot of the catalyst yield data case study

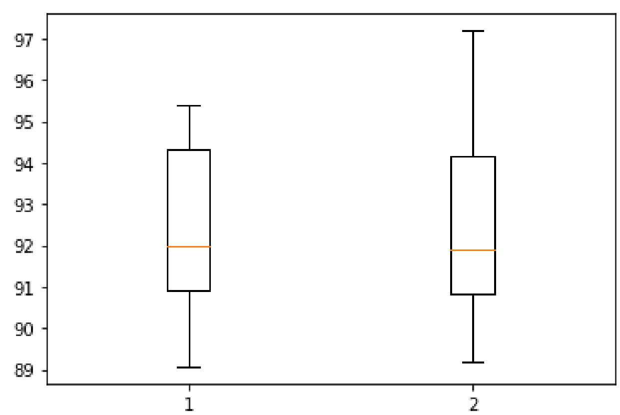

Figure 8. Comparative box plot of the catalyst yield data case study

6.7. Regression Analysis

Many problems in engineering and the sciences involve studying or analyzing the relationship between two or more variables. For instance, the pressure of gas in a container is related to temperature, the velocity of water in an open channel is related to the width of the channel, and the displacement of a particle at a certain time is related to its velocity.

Regression analysis, a collection of statistical tools, is used to model and explore relationships between variables related in a nondeterministic manner. Given its frequent application in various branches of engineering and science, regression analysis stands as one of the most widely used statistical tools. Assessing the adequacy of a linear regression model involves testing statistical hypotheses about the model parameters and constructing certain confidence intervals.

As an illustration, consider the data in Table 4, where y represents the purity of oxygen produced in a chemical distillation process, and x is the percentage of hydrocarbons in the main condenser of the distillation unit. The scatter diagram produced by Python (Figure 9) indicates that while no simple curve passes exactly through all the points, there is a strong indication that the points scatter randomly around a straight line (Figure 10).

Table 4. Oxygen and Hydrogen Levels

Observation Number	**Hydrogen Level, x (%)**	**Purity, y (%)**
1	0.99	90.01
2	1.02	89.05
3	1.15	91.43
4	1.29	93.74
5	1.46	96.73
6	1.36	94.45
7	0.87	87.59
8	1.23	91.77
9	1.55	99.42
10	1.40	93.56
11	1.19	93.54
12	1.15	92.52
13	0.98	90.56
14	1.01	89.54
15	1.11	89.85
16	1.20	90.39
17	1.26	93.25
18	1.32	93.41
19	1.43	94.98
20	0.95	87.33

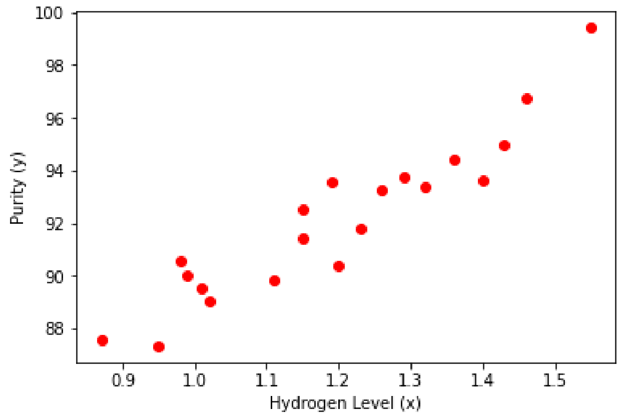

Figure 9. Scatter Diagram of Oxygen and Hydrogen level case study

The regression model describing the relationship between oxygen purity and hydrocarbon level is as follows:

$$\hat{y} = 14.947x + 74.283 \quad \text{Eq. 10}$$

with a correlation coefficient $R = 0.937$.

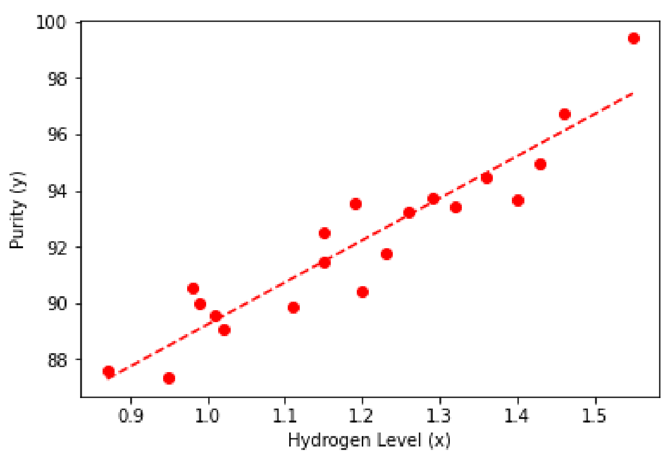

Figure 10. Regression line of the Hydrogen and Oxygen level case study

Using the regression model, predicting an oxygen purity of $\hat{y} = 90.73\%$ when the hydrocarbon level is $x = 1.1\%$ is feasible. However, these predictions are subject to error, and confidence intervals are computed to describe the estimation error from a regression model.

The confidence interval for the slope is $[12.181, 17.714]$, which does not include zero, providing strong evidence (at $\alpha = 0.05$) that the slope is not zero. The reasonably narrow confidence interval (± 2.766) is due to the fairly small error variance $\sigma^2 = 1.18$.

7. Recommendation

Given that many engineering programs worldwide already include a freshman programming course in Python or MATLAB, we propose a conceptual model in which a two-credit co-requisite focused on applied statistical engineering is integrated into the first-year curriculum. This addition would:

- Enhance students' ability to analyze engineering data effectively.
- Equip students with essential skills for international research and industry applications.
- Bridge the gap between programming skills and statistical reasoning.
- Make statistical education accessible to institutions with limited resources by using Python as a free and open-source tool.

By embedding statistical engineering into the freshman curriculum, students gain critical skills that enhance their academic and professional careers, aligning with the global emphasis on data literacy and computational proficiency.

8. Limitations and Scope

As a conceptual course design paper, this work does not report empirical evidence of student learning outcomes. Additionally, long-term retention of statistical skills depends on reinforcement in subsequent courses, and the effectiveness of the online modality may vary across institutions. Combining programming and statistics in a single course also introduces potential cognitive load for first-year students. These limitations motivate future work involving pilot implementations and formal assessment; however, they do not diminish the value of the proposed framework as a scalable and adaptable model for early engineering statistics education.

9. Conclusion

The primary outcome of this work is a transferable course framework and set of instructional examples rather than measured student performance data. This paper highlights the need for integrating statistical engineering into early engineering education, particularly with an international perspective. By shifting the focus from purely theoretical probability to applied statistical analysis, students develop practical problem-solving skills applicable in global engineering contexts. Python serves as an accessible and powerful tool for implementing statistical methods, making engineering statistics education more inclusive and adaptable.

We advocate for incorporating statistical engineering as a co-requisite to freshman programming courses, ensuring that students develop computational and analytical proficiency simultaneously.

This approach fosters a new generation of engineers who can effectively apply programming and statistical techniques in international research and industry settings. By emphasizing accessibility and adaptability, this course model supports global engineering education initiatives and prepares students for the challenges of a data-driven world.

References

[1] J. Burns, "Work in Progress: Redesigning a Multidisciplinary Engineering Statistics Course," in *Proc. ASEE Annu. Conf. Expo.*, 2019.

[2] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 5th ed. Hoboken, NJ, USA: Wiley, 2011.

[3] R. W. Hoerl and R. D. Snee, "Closing the gap: Statistical engineering can bridge statistical thinking with methods and tools," *Qual. Progress*, vol. 43, no. 5, pp. 52–53, 2010.

[4] J. Burns, E. Aref, and M. Majd, "Assessing Instructional Effectiveness and Understanding Factors that Contribute to Student Performance in an Engineering Statistics Course: An Exploratory Study," in *Proc. ASEE Annu. Conf. Expo.*, 2020.

[5] R. Shainin, "Statistical Engineering: Six Decades of Improved Process and Systems Performance," *Qual. Eng.*, vol. 25, no. 4, 2012.

[6] N. O. Erdil, "Fostering Entrepreneurial Mindset in an Engineering Statistics Course," in *Proc. ASEE Annu. Conf. Expo.*, 2021.

[7] D. Michalaka, D. S. Greenburg, and N. H. Shetty, "Incorporating Gamification at an Engineering Statistics course to improve student learning and engagement," in *Proc. ASEE Annu. Conf. Expo.*, 2023.

[8] S. Chitikeshi, J. Hildebrant, O. Popescu, O. M. Ayala, and V. M. Jovanovic, "Integrating Statistical Methods in Engineering Technology Courses," in *Proc. ASEE Annu. Conf. Expo.*, 2018.

[9] M. Cooper, "Loose Change and Dishwasher Optimization: Creative Applications of Engineering Statistics," in *Proc. ASEE Annu. Conf. Expo.*, 2013.

[10] A. Ahmad, "Use of Minitab Statistical Analysis Software in Engineering Technology," in *Proc. ASEE Annu. Conf. Expo.*, 2019.

[11] R. G. Batson, "How to Make Engineering Statistics More Appealing to Millennial Students," in *Proc. ASEE Annu. Conf. Expo.*, 2018.

[12] R. J. Marino, C. M. Weyant, and B. B. Terranova, "Probability and Statistics – Early Exposure in the Engineering Curriculum," in *Proc. ASEE Annu. Conf. Expo.*, 2019.