

Linux Essentials in Higher Education: Building Digital Literacy for Students

Björn Bäcker, Manfred Rössle, Christian Koot

Abstract— This paper evaluates "Linux Essentials," a prototypical workshop designed to address the "operational literacy" gap among non-Computer Science undergraduates using RStudio and Jupyter Notebooks. Employing a qualitative case study methodology at a German University of Applied Sciences, the three-hour intervention utilized instructional scaffolding and an iterative "theory-to-practice" cycle to teach high-leverage competencies, including shell navigation, package management, and CRUD operations within a virtualized Debian environment.

Findings from instructor observations and semi-structured interviews suggest that active experimentation and synchronous feedback significantly increased student self-efficacy and reduced command-line anxiety. However, the study also identified challenges related to cognitive load and student heterogeneity, necessitating tiered tasks and recommending an extension of the instructional timeframe to four hours. Ultimately, the framework demonstrates that structured technical interventions can successfully establish foundational literacy in non-CS cohorts, though further quantitative research is required to substantiate long-term scalability and pedagogical impact.

Keywords— Instructional Scaffolding, Linux Literacy, Business Analytics Education, Active Learning

I. INTRODUCTION

A growing number of statistics and analytics courses introduce students directly to tools such as RStudio or Jupyter Notebook, often assuming only minimal prior programming experience. However, we observed over recent years at our institution that students faced significant challenges not only with programming concepts, but also with basic operational literacy in the underlying operating system. This observation is supported by a recent survey, in which only 51% of students reported feeling adequately prepared by their university studies for the digital challenges of their future careers. [1]

This paper presents a pedagogical case study of a short, intensive "Linux Essentials" workshop designed as a companion to a statistics course using RStudio at a German university of applied sciences. The workshop targets undergraduates who typically have no formal background in computer science and exhibit heterogeneous levels of digital literacy. Our central design challenge was to support these students in acquiring just enough Linux knowledge to become independent, without overwhelming them with the full conceptual depth of operating systems. To address this, the course focuses on a reduced but high-leverage subset of competencies: installation and configuration within a controlled

virtualized environment, basic shell navigation, core file operations, and selected utility tools that directly support later work with data analytics software.

The contribution of this paper is twofold: firstly, we document the design and implementation of the Linux Essentials workshop. Secondly, we reflect on its affordances and limitations for building foundational operational literacy in non-CS students.

II. METHOD / CONTEXT

The Linux Essentials workshop was conducted at Hochschule Aalen with undergraduate Business Analytics students enrolled in a statistics course that employed RStudio as its primary software environment. The workshop was scheduled as a three-hour block during the semester and positioned explicitly as a support offering to reduce technical friction in subsequent statistics sessions. Participation was voluntary but strongly recommended by the course instructor, who observed recurring difficulties with installation, file management, and environment configuration in previous cohorts.

The teaching context is characterized by high heterogeneity in students' prior experience with Unix-like systems and command-line interfaces. Some participants had limited exposure through personal use to terminal usage, while others reported exclusively using graphical user interfaces on Windows. To accommodate this range, the curriculum was deliberately scaffolded to assume zero prior knowledge while still allowing more advanced students to remain engaged through optional extension tasks. This heterogeneity is not treated as a problem that needs to be solved, but as a core contextual constraint that shapes both the pacing and the choice of pedagogical strategies.

Methodologically, we approach the workshop as a qualitative pedagogical case study. Our analysis is grounded in the systematic description of course design decisions, the structure of learning activities, and the observed error patterns and interactions that emerged during the session. Data sources include instructor field notes taken during and immediately after the workshop and informal verbal feedback collected during the closing discussion as well as two months after the course was held. We do not claim generalizability in a statistical sense; instead, we aim to articulate design rationales and practical insights that may inform similar initiatives in other non-CS higher education settings.

Manuscript received Jan. 29, 2026.

B. Bäcker, M. Rössle, and C. Koot are with Aalen University, D-73430 Aalen, Germany

III. LINUX ESSENTIALS

A. Educational Context and Constraints

The course was designed for undergraduate students within the Faculty of Business and Health to address the critical need of bridging theoretical mathematical statistics with their practical application in data science. It is offered as an accompaniment for lectures, where this need arises. Specifically, the course aimed to provide the foundational knowledge required to manage applications like Jupyter Notebook or RStudio within a Linux Desktop environment. A critical design variable was the heterogeneity of the student body. Given that non-CS majors often exhibit high variance in digital literacy, we anticipated different levels of prior exposure to Unix-based operating systems. Consequently, the curriculum was scaffolded to assume zero prior knowledge while scaling rapidly to intermediate concepts relevant to Linux environments, enabling students to pursue further specialization in complex Data Science and DevOps infrastructures.

Acknowledging the cognitive constraints of a single-session workshop, we accepted that students could not develop a deep mental model of the Linux operating system through this three-hour intervention alone. Thus, the learning objective shifted from creating a deep understanding to conceptual understanding and exposing the students to the material for the first time in a positive way. To optimize the limited contact time, we focused on cultivating Self-Efficacy: providing an initial understanding, ensuring positive first experiences to reduce CLI anxiety, and equipping students with strategies for future learning. Therefore, the instructor repeatedly encouraged students to utilize search engines and Large Language Models (LLMs) as support tools. This approach aims to foster self-sufficiency, enabling students to continue learning through guided discovery long after the workshop concludes. By normalizing external help-seeking, we aim to transition students from passive consumers to active investigators and researchers.

B. Pedagogical Architecture

The lecture components were not limited to abstract definitions. They rather served as a bridge to technical competency. As illustrated in Fig. 1 - Iterative and cyclical course approach, this architecture follows an iterative and cyclical approach, mirroring the feedback loops found in agile management frameworks and instructional scaffolding [2].

Each module started by introducing a core concept, immediately deconstructed its specific implementation within the Debian Linux architecture, followed by demonstrating its operational usage. After the demonstration, students engaged in immediate application via situated practice to reinforce retention.

For example, after explaining the theoretical concept of package management and repositories, students were tasked with the practical execution of installing software (e.g., the cowsay package) via the command line interface (CLI). This immediate application ensured that abstract concepts like "repositories" are cemented through the practical usage of apt and dpkg. Once the students completed the exercise, the

instructor verified the results before proceeding to the next module.

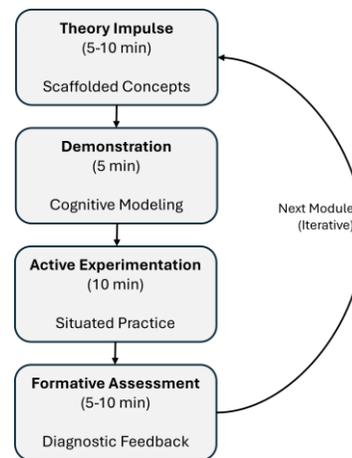


Fig. 1 - Iterative and cyclical course approach

This cyclical approach allowed for a regular assessment of the students' progress, provided many opportunities for practice during the course and limited the duration of high-cognitive-load phases (Theory impulses) to manageable intervals.

1) Active Experimentation and Formative Assessment

To ensure all participants reached a minimum understanding of the presented content, the results of the "Active Experimentation" phase were verified for each individual student before proceeding. This immediate verification creates observable learning artifacts (e.g., a working `cowsay` command), enabling synchronous feedback—a core mechanism of active learning that meta-analyses show improves exam performance by 6% and reduces failure rates by 55% compared to traditional lecturing. [3]

2) Error Analysis and Ad-Hoc Scaffolding

The opportunity for synchronous feedback proved valuable. Novice uncertainty in working with the CLI manifested through unexpected behavioral errors, such as hitting "Enter" after every word or failing to consider the current working directory. These individual interactions provided a crucial window for ad-hoc scaffolding, allowing the instructor to debug student misconceptions in real-time.

3) Heterogeneity Management

A critical challenge in technical workshops is maintaining engagement across a heterogeneous cohort. Students with prior knowledge are not cognitively stimulated by trivial tasks like installing a single package. To prevent boredom and disruption, bonus exercises were prepared for advanced students, increasing the complexity of the task without advancing the curriculum for the rest of the class. For example, while the class installed the cowsay package, advanced students were tasked with installing Visual Studio Code with dpkg. Simultaneously, individual scaffolding during active experimentation supported novices, aligning with evidence that

active learning differentially benefits lower-performing students in STEM. [4]

4) Scalability and Cooperative Learning

While the verification of results for each individual student is optimal for small cohorts (<10), a “divide and conquer” approach must be adopted for larger classes. By implementing Cooperative Learning groups, students are given the opportunity for peer-to-peer information exchange. Advanced students can reinforce their learning by assisting their peers, while the teacher is able to monitor the learning progress of the group.

C. Technical Setup & Environment

The course was hosted via Zoom, facilitating real-time screen sharing and remote troubleshooting.

The students operated within a prepared Debian virtual machine hosted on VirtualBox. This virtualization layer served as a hardware-agnostic abstraction, ensuring that all students regardless of their host system had the same prerequisites. The virtual machine and utilities like shared folders and bidirectional clipboards were set up during a previous lecture, ensuring that the three-hour window was dedicated exclusively to learning rather than installation troubleshooting.

The virtual machine was provided as a “golden image” pre-configured with data science tooling (R, Python, Jupyter Notebook) for the lecture. However, the Linux Essentials course is set up deliberately to focus on the underlying operating system functions.

D. Curricular Prioritization

The vastness of the Linux ecosystem necessitated a ruthless prioritization of topics. We focused on “high-leverage” skills - tools that provide the maximum utility for future use.

The selection process followed a practitioner-led approach, curated directly from the professional workflows of the authors.

The course began by contextualizing the technology. Students were introduced to the historical background and philosophy of Linux (e.g., the Open-Source definition) to clarify its dominance in modern infrastructure. Core ecosystem concepts, such as the distinction between the Kernel, Distributions, and Window Managers, were structured using popular examples to create mental anchors for the students. Following that, the students were taught the different competencies needed when working with the Linux operating system. The core competencies and their corresponding curricular mapping are detailed in Table 1 - Competencies And Curriculum

TABLE 1 - COMPETENCIES AND CURRICULUM

Module	Competency	Curriculum
1	Ecosystem fluency	Linux Philosophy, Open-Source, Kernel Architecture, Security, Transparency, Distributions and examples, Enterprise vs. Community
	Package Management	Repositories, Synaptic, apt/apt-get, dpkg, Flatpak
2	Shell Navigation	Terminal Window, pwd, ls, cd, absolute vs. relative paths, man, --help
	Filesystem Hierarchy	/, /home, /root, /bin, /etc, /var
3	File Operations (CRUD) & Nano Text Editor	touch, cp, mv, rm, cat, less, more, head, tail, nano
4	Archiving & Data Integrity	tar, zip, unzip, sha256sum, md5sum

For each module, the in TABLE II - EXERCISES PER MODULE described exercises were prepared, to provide the active experimentation step of the cycle:

TABLE II - EXERCISES PER MODULE

Module	Active Experimentation
1	1) Updating Repositories 2) Installing Updates 3) Installation of the package cowsay 4) (Extra) VS Code Installation with dpkg
2	1) Navigation in directories 2) Creating a directory 3) List all information for the directory 4) What else do you see in the output?
3	1) Test out CRUD Operations 2) Edit a file with nano 3) How do you save your changes? 4) (Extra) Sort the yearly data of an enterprise into separate directories per year
4	1) Computing a hash 2) Unpack a tar archive 3) Unpack a zip archive 4) (Extra) Unpack an archive and compute hashes for every file. Match these to a given hash to find the solution.

IV. EVALUATION

The instructional materials were pilot-tested with a cohort of four male students (aged 20–24) currently enrolled in their third semester of the Bachelor’s program in Business Analytics at Aalen University. Following the session, semi-structured interviews were conducted to assess the pedagogical efficacy of the intervention. The qualitative feedback highlighted several key themes:

- Active Learning and Iteration: Participants spoke highly of the “hands-on” instructional design, specifically noting that the opportunity for trial and error facilitated a deeper engagement with the material.
- Synchronous Technical Support: The provision of real-time tutor intervention via screen-sharing was identified as a critical factor in maintaining student progress.

- **Technical Literacy:** Students reported significant gains in understanding directory structures and the configuration of shared folders between host and guest environments within VirtualBox.
- **Command-Line Proficiency:** The guided introduction to terminal operations was cited as a particularly valuable component of the technical training.
- **Cognitive Load and Complexity:** While the majority of the feedback was positive, one participant reported challenges regarding the escalating complexity of the tasks, leading to a temporary loss of orientation within the workflow.

- [3] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the national academy of sciences*, 111(23), 8410-8415. <https://doi.org/10.1073/pnas.1319030111>
- [4] E.J. Theobald, M.J. Hill, E. Tran, S. Agrawal, E.N. Arroyo, S. Behling, N. Chambwe, D.L. Cintrón, J.D. Cooper, G. Dunster, J.A. Grummer, K. Hennessey, J. Hsiao, N. Iranon, L. Jones, H. Jordt, M. Keller, M.E. Lacey, C.E. Littlefield, A. Lowe, S. Newman, V. Okolo, S. Olroyd, B.R. Peacock, S.B. Pickett, D.L. Slager, I.W. Caviedes-Solis, K.E. Stanchak, V. Sundaravardan, C. Valdebenito, C.R. Williams, K. Zinsli, & S. Freeman, Active learning narrows achievement gaps for underrepresented students in undergraduate science, technology, engineering, and math, *Proc. Natl. Acad. Sci. U.S.A.* 117 (12) 6476-6483, <https://doi.org/10.1073/pnas.1916903117> (2020).

V. DISCUSSION

Despite efficient pacing, the strict three-hour constraint required the omission of the following advanced modules:

- Identity & Privilege Management
- Network & Utility tools
- System Monitoring & Process Management
- I/O Redirection & Editors

This was due to longer than expected discussions during the Diagnostic Feedback phase, which was permitted for the interactive approach that was chosen. These prepared topics were deferred to self-study materials to preserve the overall quality of the session. Nonetheless this shows, that the chosen approach for the course can require the teacher to sacrifice lecture time.

These observations suggest that future iterations of this instructional format should incorporate a significantly extended timeframe (e.g., 4 hours) to accommodate the identified complexities.

Furthermore, while this qualitative pilot study yielded encouraging preliminary insights, generalizable conclusions (particularly concerning measurable learning outcomes) necessitate rigorous quantitative evaluation. Subsequent research should employ standardized assessment metrics to substantiate the pedagogical efficacy of the course design.

VI. CONCLUSION

The prototypical instructional framework for Linux Essentials presented in this study demonstrates significant potential for the specified target demographic. While these initial findings are promising, a robust empirical validation of the curriculum's long-term efficacy remains a primary objective for subsequent research. Future longitudinal studies will be essential to substantiate the pedagogical impact and scalability of this instructional design.

REFERENCES

- [1] Hartmann, S., & Marx, M. (2022). *Der digitale Wandel aus dem Blickwinkel von Studierenden: Gern gesehener Helfer? Oder Konkurrent am Arbeitsplatz?* (Sonderpublikation Studienreihe Fachkraft 2030). jobvalley (Studitemps GmbH) & Maastricht University. https://jobvalley.com/pdf/221017_Digitalisierung.pdf
- [2] Wood, D., Bruner, J. S., & Ross, G. (1976). *The role of tutoring in problem solving*. *Journal of child psychology and psychiatry*, 17(2), 89-100. <https://doi.org/10.1111/j.1469-7610.1976.tb00381.x>