From Notebooks to Immersion: Integrating Augmented Reality into Data Science Workflows

Derek Willis*   J. Edward Swan II†

Mississippi State University

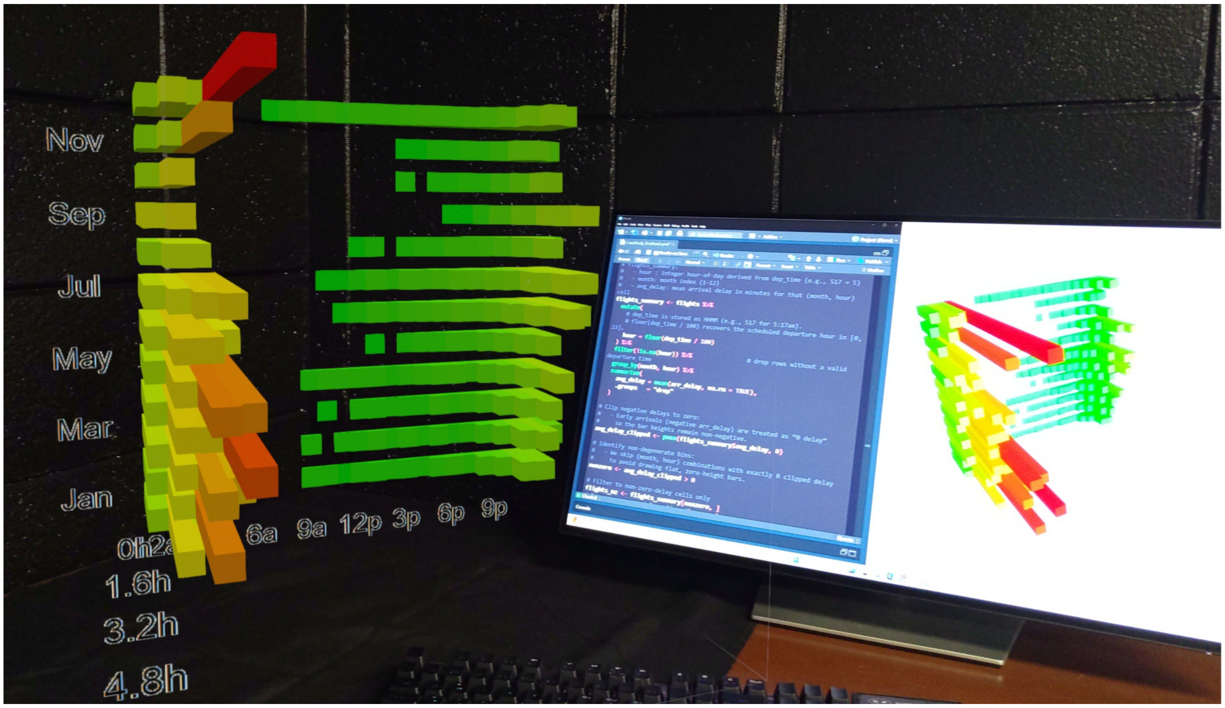

Figure 1: Notebook-based analysis alongside immersive AR inspection of a 3D visualization. XRGL enables desktop-based construction and modification while supporting stereoscopic inspection in an AR head-mounted display.

ABSTRACT

Data-science notebooks support iterative analysis but are typically limited to monoscopic 2D displays. We present XRGL, a notebook-native workflow integrating augmented reality (AR) to enable immersive inspection of 3D visualizations while preserving established notebook practices. Unlike prior work that emphasizes immersive environments as the primary analytic workspace [2], XRGL separates interaction modes: analysis and authoring remain on the desktop, while spatial inspection is performed in AR. An exploratory user evaluation suggests the workflow is usable and supports smooth transitions between desktop and immersive views. This work demonstrates the feasibility of integrating notebook-based analysis with AR without disrupting existing workflows.

Index Terms: Augmented reality, immersive analytics, data science notebooks, 3D visualization, workflow integration, exploratory analysis.

*e-mail: dpw9@msstate.edu
†e-mail:swan@cse.msstate.edu

This is an author version preprint. The final version is available as: Derek Willis, J. Edward Swan II, "From Notebooks to Immersion: Integrating Augmented Reality into Data Science Workflows", In *IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops* (VRW 2026), Daegu, Korea, March 21–25, 2026, pages 1173–1174, DOI: 10.1109/VRW70859.2026.00261.

1 INTRODUCTION

Interactive data-science notebook environments, such as Jupyter notebooks, VS Code, RStudio, and emerging IDEs like Positron, are widely used for exploratory data analysis, integrating code, narrative, and output within a single environment. These tools increasingly support interactive 3D graphics [5]. However, notebook-based visualization remains constrained to monoscopic 2D displays, requiring analysts to infer depth and spatial relationships from indirect visual cues rather than stereoscopic views of 3D data.

AR head-mounted displays provide spatially anchored, stereoscopic views of 3D data that can support reasoning about shape, scale, and spatial relationships [1]. Prior immersive analytics systems often position immersive environments as the primary analytic workspace [2]. In practice, such systems typically require external tools, custom applications, or game-engine-based pipelines outside established notebook workflows, introducing friction that limits routine use. This work explores whether immersive inspection can be integrated into notebook-based workflows in a lightweight, workflow-native manner by deliberately separating 2D desktop authoring from 3D immersive inspection, recognizing that conventional monitors with keyboard-and-mouse input remain more efficient for code-driven analysis, while AR is reserved for stereoscopic spatial exploration.

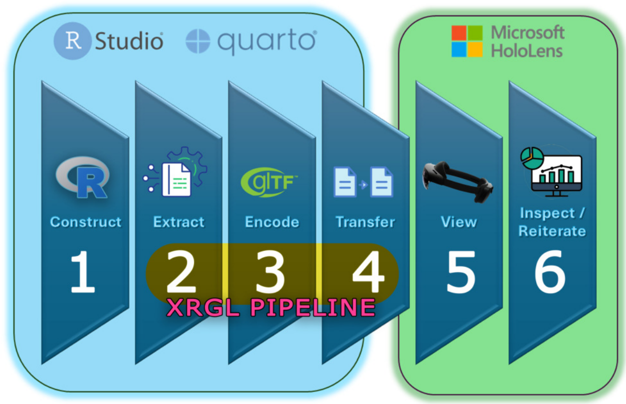

Figure 2: The XRGL pipeline separates notebook-based 2D analysis from 3D immersive inspection, enabling iterative movement between desktop and AR views.

2 XRGL: A Notebook-Native AR Workflow

This project introduces XRGL, a lightweight, notebook-native R package for immersive inspection of 3D visualizations. To our knowledge, XRGL is the first notebook-native workflow to separate 2D analysis from 3D immersive inspection while remaining entirely within a data-science notebook. AR is treated as an extension of the visualization loop, as illustrated in Figure 2.

The workflow begins with scene construction using the R Graphics Library (rgl), commonly used for interactive 3D visualization within notebooks [5]. When immersive inspection is desired, XRGL extracts the active scene, converts its components into a standardized representation, and writes a self-contained model file. The model is transferred to an optical see-through AR device and opened using the native 3D viewer.

The XRGL pipeline can be invoked from a single notebook cell, enabling iterative movement between desktop and immersive views with minimal overhead. By relying on standard file formats and native viewing applications [3, 4], XRGL avoids custom rendering engines, preserving the reproducibility of notebook-based workflows.

3 Exploratory Evaluation

We conducted an exploratory evaluation to assess the feasibility and usability of the XRGL workflow within a notebook-based analysis context. Four participants were guided through a small set of representative 3D scenes, including geometric primitives, structured grids, and data-driven visualizations generated within notebooks. Participants were informed that they could remove the head-mounted display or flip the visor upward at any time for comfort while interacting with the notebook using a standard monitor, keyboard, and mouse. In practice, all four participants elected to keep the headset on with the visor down throughout the session and reported no discomfort. For each task, participants executed notebook cells to generate a scene, invoked XRGL to export the visualization, and inspected the resulting model using a Microsoft HoloLens 2, an optical see-through AR head-mounted display. Figure 3 shows a representative XRGL session during a case-study task, illustrating how notebook-based code execution and desktop visualization coexist with immersive AR inspection of the exported 3D model.

Feedback was collected using a brief post-session questionnaire and open-ended prompts. The questionnaire consisted of seven Likert-scale items and three free-response items assessing usability, comfort, and workflow integration. Participants reported that the workflow was intuitive and that transitions between desktop and immersive views were smooth. Several participants noted that immersive inspection supported reasoning about spatial structure less

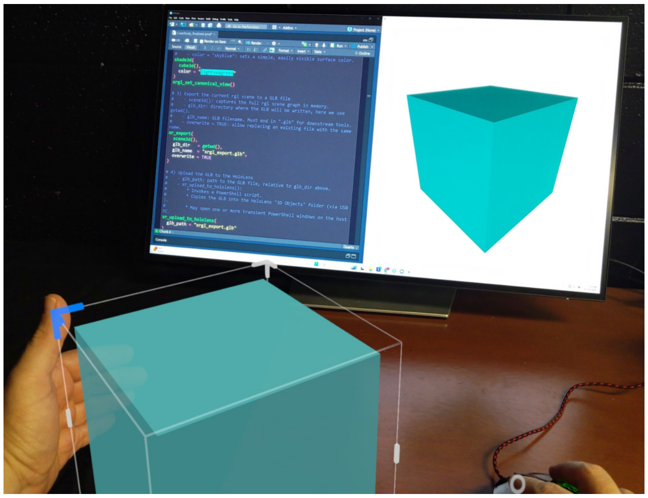

Figure 3: Representative XRGL session during a case-study task. A participant inspects an exported 3D visualization using an optical see-through AR head-mounted display while continuing notebook-based code execution and desktop interaction on a conventional monitor.

apparent in monoscopic desktop views. The evaluation was qualitative in nature and intended to characterize usability, workflow continuity, and perceived integration with existing notebook practices rather than to measure task performance or comprehension outcomes.

4 Significance

By preserving established notebook practices and avoiding specialized AR development environments, XRGL lowers barriers to experimentation with immersive visualization during routine data analysis. Rather than repositioning immersion as a replacement for existing analytic workflows, the approach illustrates how augmented reality can function as an incremental extension that supports stereoscopic inspection of 3D data when needed.

More broadly, this design highlights an alternative integration strategy for immersive analytics that prioritizes continuity, reproducibility, and analyst control. By decoupling 2D authoring from 3D immersive inspection, XRGL suggests a pathway for incorporating AR into existing data-science ecosystems without requiring wholesale changes to tools, practices, or expertise.

Supplemental Materials

The supplemental materials include (1) a short video demonstrating XRGL applied to a representative case-study task, highlighting transitions between notebook-based analysis and immersive AR inspection, and (2) a PDF of the submitted poster.

References

[1] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997. doi: 10.1162/pres.1997.6.4.355 1

[2] M. Chen et al. Immersive analytics. *IEEE Computer Graphics and Applications*, 38(3):125–140, 2018. doi: 10.1109/MCG.2018.032421654 1

[3] Khronos Group. gltf 2.0 specification. `https://www.khronos.org/gltf/`, 2017. Accessed 2025. 2

[4] Microsoft. 3d viewer and 3d content on hololens. `https://learn.microsoft.com/`, 2021. Accessed 2025. 2

[5] D. Murdoch. rgl: 3d visualization using opengl. *The R Journal*, 3(2):29–33, 2011. 1, 2