2025-09-17

fozziejoin: High-Performance String Distance Joins in R

Jon Downs
Goose Data Science and Engineering, LLC

Executive Summary

Approximate string matching is essential for record linkage in messy datasets, but existing tools like `fuzzyjoin` can be **slow and memory-intensive**.

This proposal supports the continued development of **`fozziejoin`**, a high-performance R package that delivers **over 100× speedups** for certain algorithms by replacing the `stringdist` backend with optimized Rust code. Speedups are most pronounced on Linux systems.

`fozziejoin` already implements **9 of the 10 string distance algorithms** supported by `fuzzyjoin`, with **consistent improvements across platforms**.

To complete its core functionality and prepare for CRAN submission, this project will:

- **Add** the `soundex` algorithm and a `semi` join type
- **Align** function signatures and outputs with `fuzzyjoin` for easy migration
- **Finalize** documentation, including vignettes
- **Achieve** CRAN publication

If funded, `fozziejoin` will offer **scalable, open-source tools** for **public health**, **social science**, and **government analytics**, with broad utility for **large administrative datasets**.

The source code and development history are available at:
https://github.com/JonDDowns/fozziejoin/tree/main

Signatories

Project Team

The project is led by Jon Downs. Jon brings extensive experience developing proprietary tools for public sector data science teams. Work will be conducted through a personal LLC. A Contributor Covenant–based Code of Conduct and other improvements will be developed to facilitate future collaboration and community contributions.

Consulted

During proposal preparation, Jon consulted Hadley Wickham, who responded that the approach "sounds like a good plan." The proposal and code repository have also been shared with the extendR Discord community, where it received positive feedback and interest.

The Problem

Approximate string matching is essential for data integration and entity resolution in administrative datasets. In R, the `fuzzyjoin` package (Robinson 2020) is widely used for this purpose, with **9,103 CRAN downloads in August 2025** (Csárdi 2022).

However, `fuzzyjoin` relies on the `stringdist` (Mark van der Loo 2014) package, which is not optimized for this use case. It returns **all pairwise string distances** as an intermediate result—most of which are discarded by a user-defined threshold. This creates substantial and avoidable memory allocation.

Other packages, such as `zoomerjoin` (Green 2025), offer performant probabilistic methods via **Locality Sensitive Hashing (LSH)**. But LSH limits algorithm choice and result accuracy, so `zoomerjoin` does **not fully replace** `fuzzyjoin`.

The proposal

Overview

To address performance limitations in `fuzzyjoin`'s string distance joins, we propose a new R package: `fozziejoin`. `fozziejoin` currently implements **9 of 10 string distance algorithms** from `fuzzyjoin`, with select benchmarks showing 100× speedups on Windows and Linux.

These improvements enable **scalable approximate joins in R**, with applications in **public health**, **social science**, and **government analytics**.

The codebase is designed for **CRAN compliance**. This grant will:

- Improve alignment with `fuzzyjoin`'s interface and functionality
- Develop documentation and supporting materials
- Submit to CRAN

Timeline: December 2025 to July 2026

Detail

Minimum Viable Product

The minimum viable product is a **CRAN-published version of `fozziejoin`**, including:

- The `stringdist_join` family of functions
- Comprehensive documentation and usage vignette
- An example dataset
- A testing suite

Interface and output will broadly align with `fuzzyjoin`, though exact replication is not a goal. Differences and planned adjustments are tracked in the following GitHub issue, and will evolve based on community feedback and practical use:

https://github.com/JonDDowns/fozziejoin/issues/5

Architecture

fozziejoin uses Rust via rextendr to accelerate string matching in R.

Assumptions

This project assumes:

- fozziejoin's performance gains will justify its installation requirements
- Alternatives like zoomerjoin are not direct substitutes for fuzzyjoin's flexibility

If either assumption proves false, adoption may be limited or redundant.

External dependencies

Requires R 4.2+, the base stats package, and optionally tibble (Müller and Wickham 2023). Development tools include cargo, rustc, and xz. After CRAN distribution, CRAN binaries will reduce the install burden for Windows and macOS users.

Project plan

Start-up phase

The project uses the **MIT license** and is hosted on **GitHub**. The **Contributor Covenant** will guide community standards, and **GitHub Discussions** will support collaboration and feedback. **Monthly updates** and **quarterly summaries** will be provided for ISC reviewers.

Technical delivery

Development will span **8–9 months**, beginning **Nov/Dec 2025**, and organized into four milestones:

Date	**Milestone**
February 15, 2026	Add the soundex algorithm and semi join type
April 15, 2026	Align function signatures and output structures with fuzzyjoin
May 15, 2026	Complete documentation, vignettes, and source install guidance
July 31, 2026	Submit the package to CRAN and respond to feedback as needed

Other aspects

To publicize the work, the proposer will:

- Share updates on the **R Consortium blog** (at least quarterly)
- Promote via **LinkedIn** and the **extendR Discord** community
- Submit a talk proposal to the next **UseR!** conference
- Attend **ISC meetings** as requested to provide updates and receive feedback

Budget & funding plan

The budget covers labor costs for development, documentation, and CRAN prep at **$100/hour** — a standard rate for specialized R/Rust work. The scope is defined for high-impact delivery.

Milestone	Target Date	Hours	Funding
Finalize core functionality	Feb 15, 2026	40 hrs	$4,000
Match `fuzzyjoin` signatures	Apr 15, 2026	25 hrs	$2,500
Documentation & vignettes	May 15, 2026	25 hrs	$2,500
CRAN acceptance	Jul 31, 2026	20 hrs	$2,000
Total		**110 hrs**	**$11,000**

Success

Definition of done

Success is defined as release on CRAN with all milestones completed.

Measuring success

Progress will be tracked through milestone completion and public deliverables. Monthly updates and quarterly summaries will be shared with ISC reviewers. Community feedback via GitHub Discussions and social media will serve as informal indicators of adoption and interest.

Future work

`fozziejoin` could be extended in several directions:

- Additional join types from `fuzzyjoin`, such as numeric distance, regex, or geographic joins
- Supporting `arrow` and other output modalities
- Utilize `fuzzyjoin` to create a `RecordLinkage` alternative

The project is modular and extensible, enabling future community-centered development.

Csárdi, Gábor. 2022. "Cranlogs: Download Logs from the RStudio CRAN Mirror." https://cran.r-project.org/package=cranlogs.

Green, Beniamino. 2025. "Zoomerjoin: Superlatively Fast Fuzzy Joins." https://CRAN.R-project.org/package=zoomerjoin.

Mark van der Loo. 2014. "Stringdist: Approximate String Matching, Fuzzy Text Search, and String Distance Functions." https://cran.r-project.org/package=stringdist.

Müller, Kirill, and Hadley Wickham. 2023. "Tibble: Simple Data Frames." https://CRAN.R-project.org/package=tibble.

Robinson, David. 2020. "Fuzzyjoin: Join Tables Together on Inexact Matching." https://cran.r-project.org/package=fuzzyjoin.