Data Management and Sharing Plan

## Types of Data and Materials

PolicyEngine will produce the following types of data and materials through this project:

* Source code for tax-benefit microsimulation models
* Synthetic microdata representing US households
* Policy rules encoded as computer-readable parameters
* Documentation of models, data, and methods
* User guides and tutorials
* Survey data on user needs and feedback
* Metrics on model usage and performance

## Data and Metadata Standards

PolicyEngine will use the following standards:

* Python code following PEP 8 style guide
* YAML for policy parameter specifications
* CSV and HDF5 for tabular and hierarchical data
* Markdown for documentation

Where existing standards are inadequate, we will document our approach and rationale.

## Access, Sharing and Privacy Policies

PolicyEngine is committed to open access. We will make all code, data and documentation publicly available on GitHub under an MIT license, with the following exceptions:

* Sensitive survey responses will be anonymized before sharing
* Synthetic tax microdata will be checked to ensure no re-identification is possible before release (most data comes from the publicly available Current Population Survey)

Access to development branches will be restricted to authorized contributors. All code changes will go through peer review before merging.

## Policies for Re-use and Distribution

Users may freely re-use and redistribute PolicyEngine materials under the terms of the MIT license. We encourage derivative works and only ask for attribution.

## Archiving and Preservation

GitHub will serve as the primary archive for code and documentation.

## Security Plan

PolicyEngine takes a multilayered approach to security:

**Access Controls:**

* Two-factor authentication required for GitHub access
* Role-based access controls on development branches
* API access secured via OAuth 2.0

**Quality Assurance:**

* Automated test suite run on all code changes
* Manual code review required before merging
* Continuous integration to catch errors early

**Secure Development:**

* Dependencies automatically updated and scanned for vulnerabilities
* Static code analysis to identify potential security issues
* Regular security training for developers

**Release Management:**

* Versioned releases with changelogs
* Automated deployment pipeline with approvals
* Ability to roll back releases if issues found

We do not anticipate handling sensitive personal data. If this changes, we will implement appropriate safeguards and seek IRB approval as needed.

This data management approach will ensure PolicyEngine's outputs are FAIR (Findable, Accessible, Interoperable, Reusable) while maintaining security and quality. We are committed to open science principles and maximizing the impact of NSF-funded research.