

The Resilience and Disaster Recovery (RDR) Tool Suite

User Guide Version 2024.1

Volpe Project: OS70D123

May 22, 2024

VNTSC-OST-24-07 Rev. 1

DOI: 10.21949/1530959 Rev. 1

Prepared for:

**United States Department of Transportation
Office of Research, Development, and Technology
Washington, D.C.**



**The Federal Highway Administration
Office of Natural Environment
Washington, D.C.**



Recommended citation:

Jonathan Badgley, Andrew Breck, Juwon Drake, Daniel Flynn, Olivia Gillham, Michelle Gilmore, Kristin C. Lewis, Alexander Oberg, Gretchen Reese, Scott Smith, Kevin Zhang. The Resilience and Disaster Recovery (RDR) Tool Suite: User Guide Version 2024.1. Volpe National Transportation Systems Center, Cambridge, MA, 2024. DOI: 10.21949/1530959 Rev. 1.

Notice

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report.

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE May 22, 2024		3. REPORT TYPE AND DATES COVERED Report (6/2021-5/2024)
4. TITLE AND SUBTITLE The Resilience and Disaster Recovery (RDR) Tool Suite: User Guide Version 2024.1			5a. FUNDING NUMBERS OS20A121, OS70D123	
6. AUTHOR(S) Jonathan Badgley, Andrew Breck, Juwon Drake, Daniel Flynn, Olivia Gillham, Michelle Gilmore, Kristin C. Lewis (Project Manager), Alexander Oberg, Gretchen Reese, Scott Smith, Kevin Zhang			5b. CONTRACT NUMBER 693JK421NT800008 693JK423NT800029	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Department of Transportation John A Volpe National Transportation Systems Center 220 Binney, Cambridge, MA 02142-1093			8. PERFORMING ORGANIZATION REPORT NUMBER VNTSC-OST-24-07 Rev. 1	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Dept. of Transportation Office of Research, Development and Technology 1200 New Jersey Ave, SE, Washington, DC 20590			10. SPONSORING/MONITORING AGENCY REPORT NUMBER DOI: 10.21949/1530979 Rev. 1	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT This document is available to the public at https://volpeusdot.github.io/RDR-Public .			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Volpe developed the Resilience and Disaster Recovery (RDR) Tool Suite in support of the USDOT Office of Research, Development and Technology in collaboration with the Federal Highway Administration's Office of Natural Environment. The RDR Tool Suite enables transportation practitioners to assess the return-on-investment of resilient infrastructure across a range of potential hazard conditions to help prioritize resilience investments. This User Guide provides users with information on how to run the RDR Tool Suite and its components. It is complemented by the RDR Tool Suite Technical Documentation, Quick Start Guide, Reference Scenario Library, and Run Checklist.				
14. SUBJECT TERMS			15. NUMBER OF PAGES 115	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	

Contents

Executive Summary.....	9
1 Introduction	10
2 RDR Setup Instructions	12
2.1 Software Requirements	12
2.2 Software Installation	12
2.3 Software Configuration.....	14
3 Preparing Data for RDR	15
3.1 Base Routable Network Preparation	15
3.1.1 Road Network Components.....	18
3.1.2 Road Network Data Sources	23
3.1.3 Road Network Validation	25
3.2 Trip Table Preparation	26
3.2.1 Production-Attraction versus Origin-Destination	27
3.2.2 Trip Table Data Sources	27
3.3 Trip Table Validation	29
4 Running the RDR Tool Suite	31
4.1 Executing Additional Runs	33
5 RDR Exposure Analysis Tool	35
5.1 Configuring the Exposure Analysis GIS Network.....	35
5.2 Configuring the RDR Exposure Analysis Tool	37
5.3 Running the Exposure Analysis Tool	40
5.4 Considerations and Limitations	42
6 RDR Return on Investment Analysis Tool	44
6.1 Data Inputs.....	44
6.1.1 Configuring Scenario Definition Inputs	45
6.1.2 Configuring Exposure Scenarios.....	52
6.1.3 Configuring the Core Model Network.....	53
6.1.4 Configuring the RDR Metamodel Parameterization Submodule	54
6.1.5 Configuring the RDR Scenario Expansion Submodule	57
6.1.6 Configuring Resilience Projects.....	62
6.1.7 Configuring the ROI Analysis.....	65
6.2 Reporting and Visualization	67
6.2.1 Opening the Tableau Workbook.....	67

6.2.2	General guidelines for using the Tableau Visualization	67
6.2.3	General guidelines for using results in decision-making	68
6.2.4	General guidelines for comparing RDR to other analyses, e.g., REMI	68
7	RDR Equity and Benefits Analysis Tool.....	70
7.1	Running the Equity Overlay Script	70
7.2	Running the TAZ Metrics Script.....	73
8	RDR User Interface	76
8.1	Running the User Interface	76
8.1.1	Basic Commands and Navigation	77
8.1.2	Setting Parameters.....	79
8.1.3	Building a Batch File	81
8.2	Help and Documentation	82
8.3	Saving and Loading	82
9	Troubleshooting.....	83
9.1	Installation	83
9.2	Input Data Preparation	83
9.3	RDR Scenario Run Execution	84
9.3.1	Executing a Run for the First Time	85
9.4	RDR Output Analysis	86
10	Conclusion.....	87
	Appendix A: RDR ROI Analysis Tool Input Files	88
	Appendix B: RDR ROI Analysis Tool Module Outputs.....	91
	Appendix C: Glossary of Terms Relevant to RDR Tool Suite	94
	Appendix D: Defining Exposure Data	96
	Appendix E: Transportation Asset and Hazard Scenario Data Summary.....	100
	Appendix F: Baseline Network Run Helper Tool	103
	Appendix G: Base Year Run Helper Tool.....	106
	Appendix H: Format Demand Helper Tools.....	108
	Appendix I: Format Network Helper Tools.....	110

Tables

Table 3-1 Road and transit links for a hypothetical RDR network	17
Table 3-2 Road link toll and travel time by link type and traveler type.....	17
Table 3-3 Default facility types in the RDR Tool Suite	19
Table 3-4 GTFS facility types not supported by RDR.....	19
Table 3-5 Link fields for AequilibraE compared to GMNS	21
Table 3-6 Example of link_types_table.csv (with notional values)	23
Table 4-1: RDR Metamodel and ROI Analysis Tool Modules	31
Table 6-1: Default depth-damage relationship (from Simonovic et al. 2011)	60
Table 6-2: Default depth-damage look-up table used by the RDR Tool Suite for flood-based hazards.....	60
Table 7-1 Example TAZ categorization CSV file	70
Table 8-1: Universal commands for the RDR UI.....	77
Table 8-2: Shortcut commands to access parameter entry pages in the RDR UI	77
Table A-0-1: RDR Tool Suite Input Files	88
Table E-0-1: Transportation asset data resources for scenario development.	100
Table I-0-1 Selected transit link fields	111
Table I-0-2 Link costs and travel times.....	114

Figures

Figure 1-1: Structure of the RDR Tool Suite. The scenario space increases as more potential hazard severities, durations, recovery periods, and resilience investments are assessed.	10
Figure 2-1: RDR Directory Structure	13
Figure 3-1 Data preparation flow for setting up a routable road network.....	15
Figure 3-2 Road and transit links for a hypothetical RDR network.....	16
Figure 3-3: Sample network file for Reference Scenario 4	20
Figure 3-4 Sample routable road network with nodes, links, and shape points	21
Figure 3-5 Sample download from Overture (Binney Street, Cambridge, MA)	25
Figure 3-6 Data preparation flow for setting up trip tables.....	26
Figure 3-7 Sample snapshot of home-work flows from LEHD	29
Figure 4-1: Environment Variables in Batch File	33
Figure 4-2: End of Batch File	33
Figure 5-1: Virginia Beach TDM Network Alignment with ArcGIS Basemap.....	36
Figure 5-2: TDM Misalignment with ArcGIS Basemap	36
Figure 5-3: Example output from a simple GIS Exposure Analysis overlaying depth grid data onto the TDM transportation network to identify disrupted links, assuming all exposure leads to disruption.....	37
Figure 5-1: An example of a manual link availability CSV file	41
Figure 6-1: Input Data Folder Structure	44
Figure 6-2: AEMaster Subfolder Structure.....	45
Figure 6-3: Sample Configuration File.....	47
Figure 6-4: UncertaintyParameters tab of model parameters file for Quick Start 1	48
Figure 6-5: ProjectGroups tab of model parameters file for Quick Start 1	49
Figure 6-6: Hazards tab of model parameters file for Quick Start 1.....	50
Figure 6-7: User inputs file for Quick Start 1	51
Figure 6-8: Sample exposure analysis file for Quick Start 1.....	53
Figure 6-9: Configuration file showing parameters used to define hazard event duration uncertainties .	58

Figure 6-10: Sample project info table for Quick Start 1	64
Figure 6-11: Sample project table file for Quick Start 1	65
Figure 6-12: Sample file of base year core model runs for Quick Start 1	66
Figure 7-1 View of USDOT Transportation Disadvantaged Census Tracts for an area of Virginia. Orange indicates disadvantaged census tracts, blue shows TAZ for the study area, and nodes of the travel model are shown as points. Map shown for illustrative purposes; the helper tool does not produce maps.....	72
Figure 7-2 Zoomed in view of the overlay of transportation disadvantaged census tracts and TAZ used for RDR modeling. Map shown for illustrative purposes; the helper tool does not produce maps.....	72
Figure 8-1: Main menu of RDR UI	76
Figure 8-2: Example of parameter entry page in RDR UI	80
Figure 8-3: Example of RDR UI data entry page for an object with multiple associated parameters	81
Figure I-0-1 Example transit links from GTFS mapped on a streets basemap	111
Figure I-0-2 Buffer around transit stops	113

Acronyms

Acronym	Meaning
ABC	Accelerated Bridge Construction
ARTBA	American Road Transportation and Builders Association
BCA	Benefit-Cost Analysis
BCA-U/Regret	Benefit-Cost Analysis under Uncertainty/Regret Analysis
BPR	Bureau of Public Roads
CMIP	Coupled Model Intercomparison Project
DOT	Department of Transportation
EMAT	Exploratory Modeling and Analysis Tool
FEMA	Federal Emergency Management Agency
FHWA	Federal Highway Administration
GIS	Geographic Information System
GMNS	General Modeling Network Specification
HERS	Highway Economic Requirements System
HRTPO	Hampton Roads Transportation Planning Organization
IRI	International Roughness Index
MPO	Metropolitan Planning Organization
NCHRP	National Cooperative Highway Research Program
OMX	Open Matrix
OSM	OpenStreetMap
OST	Office of the Secretary of Transportation
PHT	Person Hours Traveled
PMT	Person Miles Traveled
RDRM	Resilience and Disaster Recovery Metamodel
ROI	Return on Investment
STRAHNET	Strategic Highway Network
TAZ	Traffic (or Travel) Analysis Zones
TDM	Travel Demand Model(ing)
TMIP	Travel Model Improvement Program
TNC	Transportation Network Company
USDOT	United States Department of Transportation
VAST	Vulnerability Assessment Scoring Tool
VDOT	Virginia Department of Transportation
VHT	Vehicle Hours Traveled
VMT	Vehicle Miles Traveled
XLRM	EXternal factors, policy Levers, Relationships, and Metrics

Executive Summary

The **Resilience and Disaster Recovery (RDR) Tool Suite** was developed to help transportation agencies explore a large scenario space for transportation disruption and mitigation and evaluate the performance of resilience investments in the context of long-range transportation planning. The tool suite utilizes established Robust Decision-Making concepts^{1,2} that build on current travel demand modeling (TDM) analyses and address deeply uncertain future scenarios. Robust Decision-Making has been used under a similar modeling context by the FHWA Travel Model Improvement Program – Exploratory Analysis and Modeling Tool (TMIP-EMAT), a scenario-based decision-making tool that can be integrated with existing travel demand forecasting models. The RDR Tool Suite enables transportation agencies to assess transportation resilience return on investment (ROI) for specific transportation assets over a range of potential future conditions and hazard scenarios, which can then be used as a consideration in existing project prioritization processes.

The user guide provides step-by-step instructions for running the RDR Tool Suite, from installing software to interpreting the results of the analysis.

¹ RAND. Robust Decision Making. *RAND*. [Online] [Cited: July 20, 2022.] <https://www.rand.org/pubs/tools/TL320/tool/robust-decision-making.html>.

² Lempert, R. (2019). Robust Decision Making (RDM). Decision Making under Deep Uncertainty: From Theory to Practice. V. A. W. J. Marchau, W. E. Walker, P. J. T. M. Bloemen and S. W. E. Popper, Springer: 329.

1 Introduction

The RDR Tool Suite allows agencies to estimate performance under a broad array of hazard conditions and recovery patterns. There are three main components to the tool suite structure as shown in Figure 1-1. The RDR Exposure Analysis Tool is a standalone GIS-based tool for automating the process of assessing disruption to a transportation network based on exposure to a hazard. The RDR Return on Investment (ROI) Analysis Tool calculates the net benefits of resilience investments across a range of future scenarios by monetizing system performance metrics to estimate the economic impacts of disruption and the resilience investment. The RDR Equity and Benefits Analysis Tool, first released with the 2023.1 release, is a standalone tool that calculates changes in travel performance metrics due to a resilience investment during a particular hazard disaggregated at the level of traffic analysis zone (TAZ) categories.

The ROI Analysis Tool is built on the RDR Metamodel (RDRM), which takes the performance metrics results from a TDM, including number of trips, vehicle miles traveled (VMT), and vehicle hours traveled (VHT), and performs complementary disruption analyses (taking hazard exposure data that can be generated by the RDR Exposure Analysis Tool) using an open-source routing model called AequilibraE as an alternative core model. The RDRM then performs regressions and iterative scenario expansion to estimate the change in trips, VHT, and VMT associated with a range of hazard conditions and investment scenarios. The RDRM models travel behavior response to disruptions in link capacity and availability by inferring how many trips will be made, their mileage, and the time of those trips in an equilibrium state from sampled core model runs. The RDRM calculates the impact of a resilience investment, measured by changes in system performance when the investment is deployed versus a baseline disruption scenario in which the project and project alternatives are not deployed.

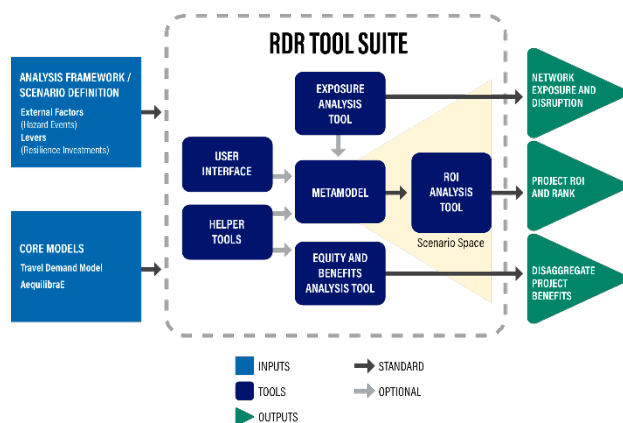


Figure 1-1: Structure of the RDR Tool Suite. The scenario space increases as more potential hazard severities, durations, recovery periods, and resilience investments are assessed.

The ROI Analysis Tool then uses these performance metrics to calculate annual costs associated with a range of recovery patterns. The ROI Analysis Tool also uses default or user-provided values for road, bridge, and transit cost of repair to estimate the cost of repair and cleanup associated with the exposure to the hazard from the RDRM. The ROI Analysis Tool uses the RDRM disruption and network performance outputs to estimate economic net benefits of avoiding disruption over the economic analysis period for a range of resilience investments and recovery patterns. It then collates the performance of different resilience investments into a visualization dashboard and ranks the projects based on their performance across the full range of uncertainties analyzed.

The RDR Tool Suite allows the user to explore a variety of problems using one or more of its components. A full run of the ROI Analysis Tool takes the user through the entire resilience return on investment analysis, from building out the analysis framework of all possible future scenarios to

calculating return on investment across a range of resilience projects. The user can also focus on more specific problems, either by running individual modules or adjusting a subset of input files. Below is a (non-comprehensive) list of potential use cases the RDR Tool Suite can help the user analyze:

- To assess which network assets are vulnerable under a given hazard condition, run the RDR Exposure Analysis Tool. Detailed documentation is provided in Section 5.
- To assess network effects of a hazard event, generate network-level performance metrics using the core model, with inputs detailed in Section 6.1.4.3.
- To estimate network-level performance metrics across a full range of possible futures from a set of core model runs, configure and fit the RDR regression model, as detailed in Section 6.1.5.1.
- To analyze a wider range of hazard recovery times but otherwise use the same scenario space, adjust the recovery module parameters in the configuration file and consider running an additional run (Section 4.1).
- To build the RDR regression model on a larger set of core model runs, adjust the metamodel parameters in the configuration file and run again with the same run_id (Section 4.1).
- To adjust repair costs with updated numbers, create user-defined repair cost look-up tables, as detailed in Section 6.1.5.3.
- To switch from running a regret analysis, or a breakeven analysis, to a benefit-cost analysis, associate future hazard events with probabilities (see the model parameters file in Section 6.1.1.2) and provide resilience project costs (see the resilience project files in Section 6.1.6.2).
- To understand how to communicate resilience costs and benefits to an audience, refer to Section 6.2.
- To compare the impact of a resilience project during a hazard event on different TAZ categories, run the RDR Equity and Benefits Analysis Tool. Detailed documentation is provided in Section 7.

This User Guide for the Resilience and Disaster Recovery (RDR) Tool Suite enables a user to install the tool suite and execute analyses. A companion document, the RDR Technical Documentation, provides details on the structure and functions of the RDR Tool Suite. The RDR Quick Start Guide provides step-by-step instructions for installing RDR and running a first scenario on a small, hypothetical transportation network. The RDR Reference Scenario Library provides test analyses in the RDR Tool Suite on a series of hypothetical and real-world transportation networks. The RDR Run Checklist provides a short reference on the inputs to check before executing an analysis. The RDR Tool Suite and documentation, along with test data for executing the Quick Start and Reference Scenario exercises, are available on GitHub at <https://volpeusdot.github.io/RDR-Public>.

2 RDR Setup Instructions

2.1 Software Requirements

Required software to run the RDR Tool Suite:

1. Conda dependency management system³ (see footnote for installation link)
2. Tableau Reader v. 2020.3 or later⁴ (see footnote for installation link) or Tableau Desktop full version (any license level)
3. (Optional) ArcGIS v10.x or ArcGIS Pro with Spatial Analyst extension (if using RDR Exposure Analysis Tool or certain other helper tools) (see Section 5, Section 7, and [Appendix I: Format Network Helper Tools](#))
4. The RDR Tool Suite

The RDR Tool Suite is written in Python 3.11⁵ and R 4.1.3,⁶ and requires an installation of the dependency management system conda, which will handle the installation of both Python and R. For this installation process, we recommend Anaconda or Miniconda,⁷ which is a miniature version of Anaconda that only includes conda, its dependencies, and Python. Alternatively, if the user has already installed other software (such as ArcGIS Pro) which uses its own package dependency management system, the user can leverage that rather than install a standalone version of conda. Note that Python will be automatically installed with conda if using Anaconda or Miniconda and R will be installed when setting up the RDR conda environment (Section 2.3). Neither needs to be installed separately by the user.

2.2 Software Installation

The RDR Tool Suite can be downloaded from the RDR repository on GitHub (<https://volpeusdot.github.io/RDR-Public>). To download, follow the instructions in the “Installation and Usage” section; download the ZIP file for the latest release. If downloading directly from the GitHub repository, click the Code dropdown and select “Download ZIP”. Once downloaded, it is highly recommended that the user unzip the file contents into a directory on their machine located at “C:\GitHub\RDR”, though the tool suite can work regardless of directory location. The user should keep track of their specific directory file path, as it is needed to edit the batch files used to execute the RDR Tool Suite. A video demo for how to install the RDR Tool Suite is also available on the [Volpe Center YouTube channel](#).

³ Conda. [Online]. User Guide: Installation. Accessed 26 July 2022 from <https://docs.conda.io/projects/conda/en/latest/user-guide/install/>.

⁴ Tableau Reader. [Online]. Accessed 26 July 2022 from <https://www.tableau.com/products/reader>

⁵ Python. [Online]. Download. Accessed 26 July 2022 from <https://www.python.org/downloads/>

⁶ R. [Online]. The R Project for Statistical Computing. Accessed 26 July 2022 from <https://www.r-project.org/>

⁷ Anaconda. [Online]. Documentation: Installation. Accessed 26 July 2022 from <https://docs.anaconda.com/anaconda/install/>

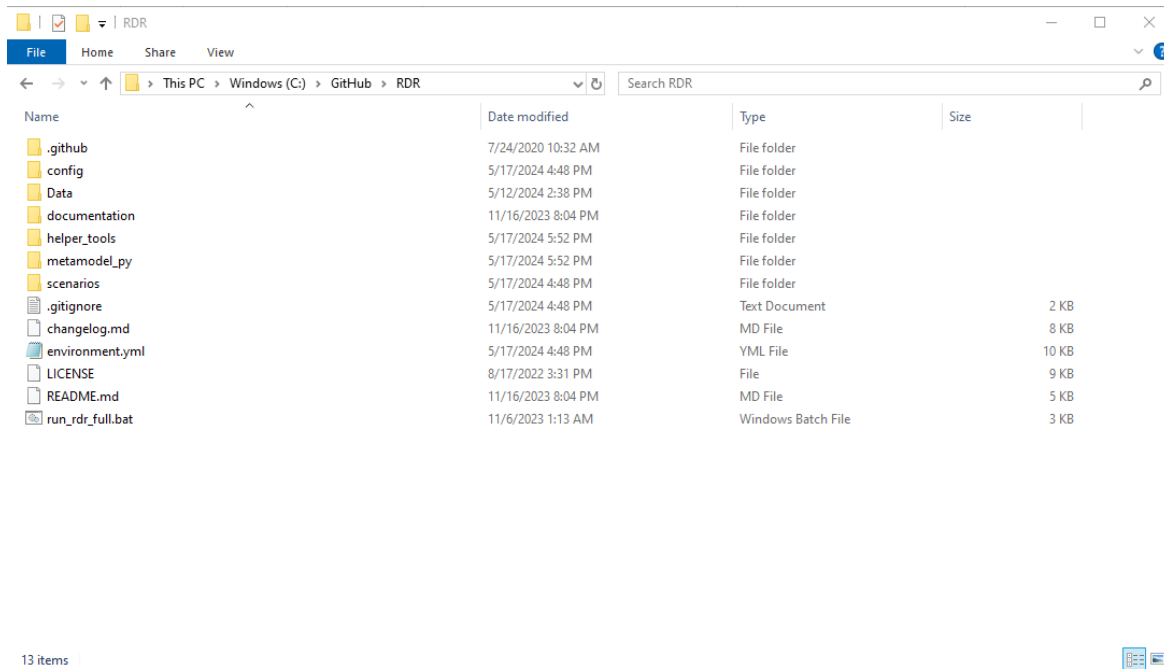


Figure 2-1: RDR Directory Structure

Figure 2-1 shows the directory structure for the RDR Tool Suite, including:

1. 'README.md': provides background on how to use the tool suite.
2. 'LICENSE': provides public license for the software and user agreement terms.
3. 'changelog.md': details code changes for each public release.
4. 'environment.yml': specifies the necessary code and package dependencies installed in the steps given in Section 2.3.
5. 'run_rdr_full.bat': template of a batch file that is used to run the RDR ROI Analysis Tool, as described in Section 4.
6. "config" subfolder: contains a sample version of the configuration file (detailed in Section 6.1.1.1) used to set run parameters for components of the tool suite, as well as a set of default and sample input files.
7. "Data" subfolder: serves as the recommended central location for the user to place input files and for the RDR Tool Suite components to generate output files.
8. "documentation" subfolder: contains the Technical Documentation, User Guide, Quick Start Guide, Reference Scenario Library, and Run Checklist for the RDR Tool Suite.
9. "helper_tools" subfolder: contains a set of code files for preparing and validating inputs and running tools in the RDR Tool Suite, including the RDR Exposure Analysis Tool, the RDR Equity and Benefits Analysis Tool, the Baseline Network Run Helper Tool, the Base Year Run Helper Tool, and helper tools for creating demand and transit input files.
10. "metamodel_py" subfolder: contains the main Python and R code files that comprise the RDR Tool Suite.
11. "scenarios" subfolder: contains all of the sample input files and network data for the Quick Start example and Reference Scenarios.

2.3 Software Configuration

The RDR Tool Suite is run from a custom conda environment, detailed in the ‘environment.yml’ file. The ‘environment.yml’ file lists the specific Python and R dependencies and versions used by the tool suite. See Section 9.1 for installation troubleshooting tips. In order to run the RDR Tool Suite, the user first needs to create the conda environment from the ‘environment.yml’ file:

1. Open an Anaconda Prompt terminal window. Searching for “Anaconda Prompt” in the Start menu should locate the application.
2. In the Anaconda Prompt terminal window, navigate to the location of the RDR directory containing the ‘environment.yml’ file using the “cd” command:

```
o cd C:\GitHub\RDR
```

In the above command, replace “C:\GitHub\RDR” with the full file path location of the user’s RDR directory if it is not C:\GitHub\RDR.

3. Run the following commands in the terminal window:

```
o conda env create -f environment.yml
o conda info --envs
```

Note that there is one hyphen in “-f” in the first command and two hyphens in “--envs” in the second command.

4. The second command in step 3 should output a list of available conda environments in the terminal window. Check that an environment named “RDREnv” shows up as an available environment.
5. (Optional) If for some reason the “RDREnv” conda environment is not functioning as expected, remove the environment using the following command, then start again at step 3. Refer to the conda documentation⁸ for details.

```
o conda env remove --name RDREnv
```

Note that there are two hyphens in “--name”.

Once the conda environment has been created following the steps laid out in this section, the user can use the Quick Start Guide (located in the “documentation” subfolder of the user’s RDR directory) to execute the Quick Start example. The Quick Start example will confirm that the user correctly installed the RDR Tool Suite on their machine and provide a walkthrough of key components of the tool. To conduct their own custom analyses, the user will need to create all required input files detailed in the Run Checklist for their specific application/network before they can run the tool, following the steps laid out in Section 4.

⁸ Conda. [Online]. User Guide: Managing environments. Accessed 26 July 2022 from <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>

3 Preparing Data for RDR

This section describes how to set up the data inputs for the RDR Tool Suite, specifically the base network and trip tables prior to performing RDR scenario runs.

3.1 Base Routable Network Preparation

RDR uses a routable road network (which can include local streets) and, optionally, a transit network.

For RDR, it is preferable that the network links include true shapes information, which means that the links on the network follow the physical roads on the ground and are not simply straight-line connections between nodes. A true shapes network is necessary for running the Exposure Analysis Tool and aids in the display of the RDR network in GIS software. See Section 3.1.2 for resources to acquire a routable network for use in RDR.

Figure 3-1 shows the data flow for creating, validating, and using the routable network.

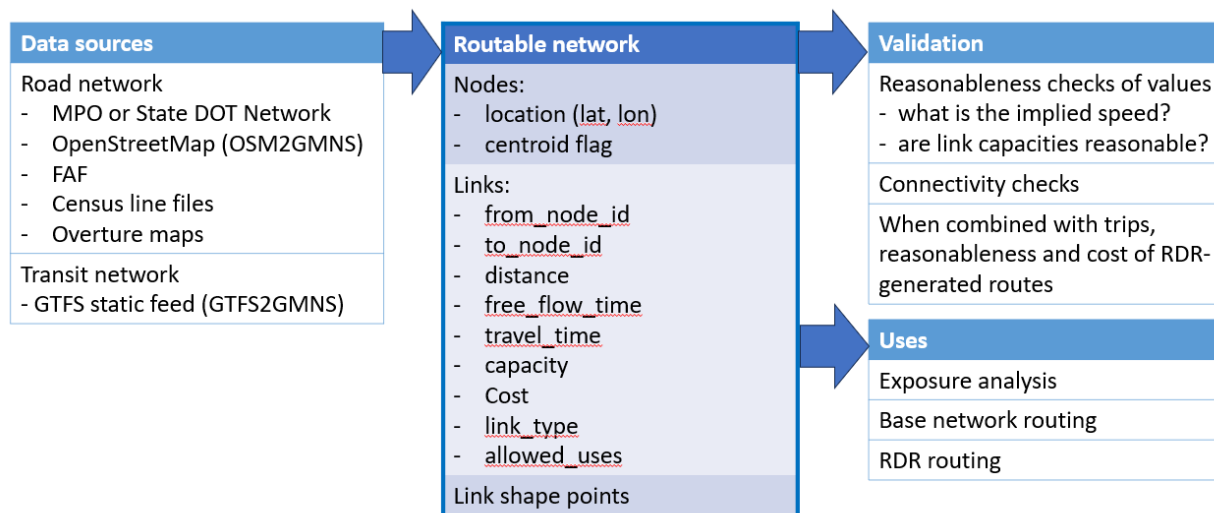


Figure 3-1 Data preparation flow for setting up a routable road network

The user can generate a routable network from a TDM network; these are accurate from a routing and mileage perspective but may not be geospatially accurate. A routable network can also be built from existing GIS networks that have been created by the city, metropolitan planning organization, or state. The user might also use OpenStreetMap as a source, using tools such as `osmnx`⁹ or `osm2gmns`,¹⁰ to synthesize the routable network and convert it to the Generalized Modeling Network Specification (GMNS) format used by RDR. A transit network may also be synthesized from General Transit Feed Specification (GTFS) files using tools such as `gtfs2gmns`.¹¹ Note that synthesizing a routable network from a GIS network often requires considerable effort to ensure that the logical connections between links (e.g., the nodes) are correct. The RDR Tool Suite provides a series of helper tools to aid the user in preparing their network files, particularly for RDR scenarios involving transit. See [Appendix I: Format Network Helper Tools](#) for more details.

⁹ Github: OSMNX repository: <https://github.com/gboeing/osmnx>

¹⁰ Lu, Jiawei, Xuesong (Simon) Zhou. 2022. [Online]. `Osm2gmns`. Accessed 26 July 2022 from <https://osm2gmns.readthedocs.io/en/latest/index.html>

¹¹ <https://libraries.io/pypi/gtfs2gmns>

Nodes and links in the network files have specific type designations that define how the core model uses them to route trip demand on the network. Some nodes in the network are labeled as centroids (origins and destinations of flow). Other nodes connect physical links. **The user should ensure that the number of nodes labeled as centroids is the same as the number of zones in the trip table.**

There are two link types for road networks:

- Road travel links, representing roads (typically include freeways, arterials, and some collectors).
- Centroid connectors, representing artificial links that connect the centroids to the travel links.

Transit nodes are classified as either transit stops, where passengers can board, deboard, and transfer between transit routes, or transit service nodes. If the RDR scenario includes transit service, the user can designate several additional link types for transit networks:

- Service links, representing line haul service on a transit route (e.g., bus, tram/light rail, subway/heavy rail).
- Transfer links, representing transfers between transit routes.
- Boarding links, representing boarding and de-boarding transit vehicles.
- Centroid connectors, representing artificial links that connect the centroids to the boarding links. (Currently, the RDR Tool Suite does not allow drive access to transit; therefore, all transit centroid connector links should be treated as walk access.)

Figure 3-2 illustrates examples of road links (in blue) and transit links (in red). The origin centroid is labeled O, and the destination centroid is labeled D. For the transit network, nodes 4, 7, 8, and 11 are “stop” nodes, representing the bus stops or transit stations. Nodes 5, 6, 9, and 10 are called service nodes, either “bus_service_nodes” or “tram_service_nodes”. They represent the transit routes that serve a particular stop. For example, if a stop is served by three transit routes, it will have one stop node, and three service nodes, one for each route.

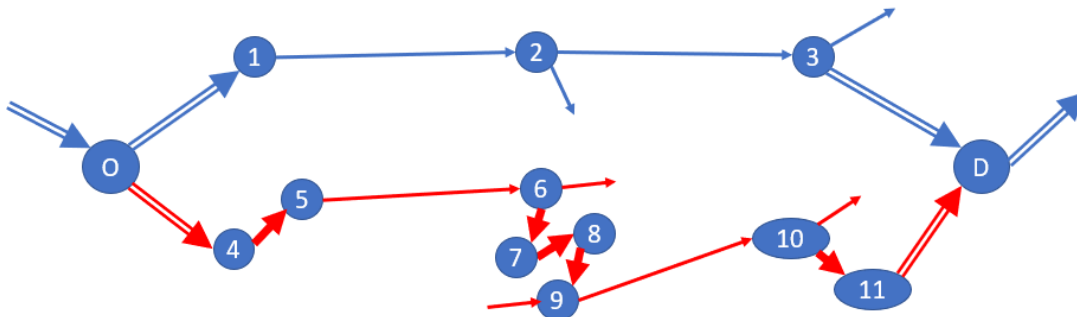


Figure 3-2 Road and transit links for a hypothetical RDR network

Each link has a direction of flow, and other measures, such as number of lanes (in one direction), capacity, free flow speed, travel time, distance, and (optionally) toll. Centroid connectors are generally assigned a high capacity and a low speed, so that they are only used for accessing the centroids. Table 3-1 lists the links shown in the example network in Figure 3-2.

Table 3-1 Road and transit links for a hypothetical RDR network

From	To	Description
0	1	Road centroid connector.
1	2	Road link.
2	3	Road link.
3	D	Road centroid connector.
0	4	Transit centroid connector (walk access). Free flow speed is set to walking speed (e.g., 3 mph).
4	5	Transit boarding link. The travel time is the wait time, and the “toll” is the transit fare.
5	6	Transit service link. The travel time is the in-vehicle time on the transit vehicle.
6	7	Transit deboarding link with very low travel time and zero toll.
7	8	Transfer link: The travel time is transfer walking time.
8	9	Boarding link after transfer: The travel time is the wait time, and the “toll” is the transit fare.
9	10	Transit service link.
10	11	Transit deboarding link with very low travel time and zero toll.
11	D	Transit centroid connector. Free flow speed is set to walking speed (e.g., 3 mph).

For links that involve driving passenger vehicles (e.g., road links), tolls and travel times will depend on whether the traveler has access to an automobile. For travelers without access to an automobile (represented by the no car trip table in RDR), the cost of using the road network is assumed to be the cost of using a taxi or transportation network company (TNC) (e.g., Uber, Lyft) broken down as a TNC initial charge and a cost per mile. The user can either manually add these costs to the link table or use the `format_network` helper tools ([Appendix I: Format Network Helper Tools](#)) to configure them. Values for the helper tool parameters, including `transit_fare`, `bus_wait_time`, `subway_wait_time`, `tnc_initial_cost`, `tnc_cost_per_mile`, and `tnc_wait_time` are found in `format_network.config`, in the `format_network` helper tool folder. Times are in minutes and costs are in cents.

Table 3-2 Road link toll and travel time by link type and traveler type

Description	Toll (car) cents	Toll (no car) cents	Travel time (car) minutes	Travel time (no car) minutes
Road centroid connector from centroid to road link	0	TNC initial charge plus cost per mile	Travel time ¹²	TNC wait time plus travel time
Road link	Toll (if any)	Toll (if any) plus TNC cost per mile	Travel time	Travel time
Road centroid connector from road link to centroid	Parking fee (if any)	TNC cost per mile	Travel time	Travel time

¹² To discourage flows via the centroid connectors, speeds on centroid connector links should be set to a low value, 10 mph or less.

3.1.1 Road Network Components

The road network includes three comma-separated (CSV) files: nodes, links, and (optional) shape points.

Nodes are points that represent travel analysis zone (TAZ) centroids or road intersections. The “Networks” subfolder must contain a CSV file named ‘node.csv’ with a full list of network nodes aggregated across network link files, following the General Modeling Network Specification (GMNS).¹³

This file is used by the ‘aeq_run’ module and has required fields:

1. ‘node_id’ – Unique ID of network node.
2. ‘x_coord’ – The x coordinate of network node.
3. ‘y_coord’ – The y coordinate of network node.
4. ‘node_type’ – Text string indicating type of node (e.g., whether node is a centroid).

In RDR, centroid nodes have a node_type of ‘centroid’. Non-centroid nodes can be designated by any other text string. Centroid nodes should be given lower node IDs than non-centroid nodes. The ‘node_id’ of the highest numbered centroid node must be designated by the “highest_zone_number” parameter in the configuration file.

Links are lines that represent either (1) connectors to the centroids, or (2) road and/or transit segments. Field names are based on those in the GMNS. Links are directional, meaning that a two-way road segment is represented as a pair of links, one going in each direction. The required fields are:

1. link_id – ID of the network link.
2. from_node_id – ID of predecessor node of network link.
3. to_node_id – ID of successor node of network link.
4. directed – Must be set to 1. All links are required to be directed; two-way roads should be entered as two separate directed links.
5. length – Numeric length of network link in miles.
6. facility_type – Numeric category defining the facility type of the network link. The user can provide their own custom types as long as they match the Facility Type fields in the repair cost and repair time tables (and the optional link types table) used. To use the default repair cost and repair time tables provided with the tool suite, the facility type for network links should follow the values in Table 3-3. This follows the FHWA Highway Functional Classification¹⁴ for road network links with values 1-7.¹⁵ The appropriate bridge facility types in Table 3-3 should be used for bridge links. Similarly, to use the default repair cost and repair time tables with transit, the facility type for transit links should be set to those in Table 3-3. The codes for transit are drawn from GTFS route_type definitions.
7. capacity – Numeric capacity of network link in vehicles per day per lane.
8. free_speed – Numeric free-flow speed of network link in miles per hour.
9. lanes – Number of lanes in the direction of travel of network link.
10. allowed_uses – Must be set to c for all links, even for transit links.
11. toll – Toll (if any) in cents, including transit fares and TNC costs (and parking fares if modeled).

¹³ Github: GMNS repository: https://github.com/zephyr-data-specs/GMNS/tree/main/Specification_md. Although a GMNS network can be very detailed, RDR only makes use of nodes and links.

¹⁴ FHWA. 2013. Highway Functional Classification Concepts, Criteria, and Procedures.

https://www.fhwa.dot.gov/planning/processes/statewide/related/highway_functional_classifications/fcauab.pdf

¹⁵ 1 – interstates, 2 – other freeways and expressways, 3 – other principal arterials, 4 – minor arterials, 5 – major collectors, 6 – minor collectors, 7 – local roads

12. travel_time – Travel time in minutes, including transit and TNC wait times.

Facility types, for both highways and transit, are detailed below in Table 3-3.

Table 3-3 Default facility types in the RDR Tool Suite

Code	Description	Notes
1	Interstate highways	Access controlled
2	Other freeways and expressways	Access controlled
3	Other principal arterials	N/A
4	Minor arterials	N/A
5	Major collectors	N/A
6	Minor collectors	N/A
7	Local roads	N/A
11	National Highway System bridges	Repair cost calculated per square foot
12	Other bridges (non-NHS)	Repair cost calculated per square foot
100	Tram, streetcar, light rail	N/A
101	Subway, metro	Underground rail system
102	Rail	Intercity or long-distance, including commuter rail
103	Bus	Short- and long-distance bus routes, including transit
600	Tram boarding link	N/A
601	Subway boarding link	N/A
602	Rail boarding link	N/A
603	Bus boarding link	N/A
700	Tram deboarding link	N/A
701	Subway deboarding link	N/A
702	Rail deboarding link	N/A
703	Bus deboarding link	N/A
801	Transit transfer link	N/A
901	Road centroid connector	Abbreviation: road CC
902	Road external connector	N/A
903	Transit centroid connector	Abbreviation: transit CC

Table 3-4 lists facility types defined in GTFS that are currently not supported by RDR. These network assets should be removed from the network files or coded under a different facility type with a manually-adjusted and appropriate travel time (and toll, if specified).

Table 3-4 GTFS facility types not supported by RDR

Code	Description	Notes
4	Ferry	Used for short- and long-distance boat service
5	Cable tram	Used for street-level rail cars where the cable runs beneath the vehicle (e.g., cable car in San Francisco)
6	Aerial lift	Cable transport where cabins, cars, gondolas, or open chairs are suspended by means of one or more cables
7	Funicular	Any rail system designed for steep inclines
11	Trolleybus	Electric buses that draw power from overhead wires using poles
12	Monorail	N/A

If the user provides an additional trip table for travelers without access to an automobile in their RDR run, corresponding 'toll_nocar' and 'travel_time_nocar' fields are required in the links files. All fields are used by the 'aeq_run' module; fields 'link_id', 'length', 'lanes', and 'facility_type' are used by the 'recov_init' module. The 'facility_type' field is used to calculate repair cost and repair time in the RDR ROI Analysis Tool. A sample network attribute file from Reference Scenario 4, which includes both road and transit links, is shown in Figure 3-3.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	link_id	from_node_id	to_node_id	directed	length	facility_ty	facility_name	capacity	free_speed	lanes	allowed_uses	toll	toll_nocar	travel_time	travel_time_nocar
2	1	11	12	1	0.9	3	arterial	1000	25	2 c		0	225	2.16	2.16
3	2	12	11	1	0.9	3	arterial	1000	25	2 c		0	225	2.16	2.16
4	3	11	10	1	0.45	6	collector	1000	25	1 c		0	112.5	1.08	1.08
5	4	10	11	1	0.45	6	collector	1000	25	1 c		0	112.5	1.08	1.08
14	100	1	11	1	0.2	901	road centroid connector	10000	10	1 c		0	350	1.2	11.2
15	101	2	12	1	0.1	901	road centroid connector	10000	10	1 c		0	325	0.6	10.6
16	102	15	3	1	0.2	901	road centroid connector	10000	10	1 c		0	50	1.2	1.2
17	20	23	27	1	0.9	101	subway	10000	40	1 c		0	0	1.35	1.35
18	21	27	23	1	0.9	101	subway	10000	40	1 c		0	0	1.35	1.35
19	22	22	26	1	0.9	103	bus	1000	20	1 c		0	0	2.7	2.7
20	23	26	22	1	0.9	103	bus	1000	20	1 c		0	0	2.7	2.7
21	24	24	25	1	0.5	103	bus	1000	20	1 c		0	0	1.5	1.5
22	25	25	24	1	0.5	103	bus	1000	20	1 c		0	0	1.5	1.5
23	120	1	20	1	0.8	903	transit CC walk	10000	3	1 c		0	0	16	16
24	121	2	20	1	0.1	903	transit CC walk	10000	3	1 c		0	0	2	2
25	122	21	3	1	0.1	903	transit CC walk	10000	3	1 c		0	0	2	2
26	130	20	23	1	0.01	601	boarding	10000	3	1 c		200	200	4	4
27	131	20	22	1	0.01	603	boarding	10000	3	1 c		200	200	7	7
28	132	26	29	1	0.01	703	deboarding	10000	3	1 c		0	0	0.2	0.2
29	133	27	29	1	0.02	701	deboarding	10000	3	1 c		0	0	0.4	0.4
30	134	25	21	1	0.01	703	deboarding	10000	3	1 c		0	0	0.2	0.2
31	135	28	24	1	0.01	603	boarding	10000	3	1 c		200	200	7	7
32	136	24	28	1	0.01	703	deboarding	10000	3	1 c		0	0	0.2	0.2
33	137	29	28	1	0.01	801	transfer	10000	3	1 c		0	0	0.2	0.2

Figure 3-3: Sample network file for Reference Scenario 4

Shape points are provided in an optional true shapes file named 'TrueShape.csv' and should be located in the "LookupTables" subfolder of the input data folder. It provides geospatially explicit shape information for links in the network through the well-known text (WKT) field, which can be used to visualize the impact of hazard events. The CSV file contains the following fields, which are used by the 'aeq_run' module:

1. link_id: ID of the link, matching the ID in the links table.
2. WKT: a well-known-text line string of the shape points.

The template SQLite database 'project_database.sqlite' is provided with the RDR Tool Suite in the "config" subfolder of the RDR directory. The database is used to input network node and link attributes to AequilibraE. The template SQLite database should be copied into the "AEMaster" subfolder of the input data folder prior to any RDR run.

Figure 3-4 illustrates a portion of a road network. The red lines are road links, with intersections as small black circles. The larger purple circles are the centroids, while the dotted purple lines are centroid connectors. The centroids labeled with numbers in the 300s represent internal zones, while those labeled with numbers in the 400s represent external zones, on the edges of the network.

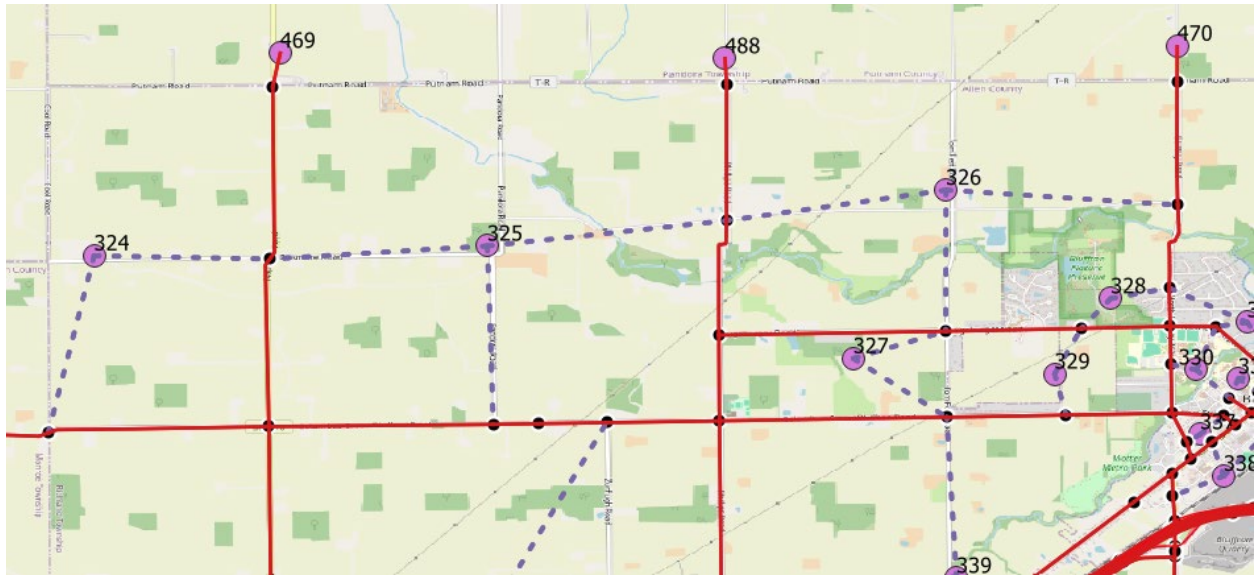


Figure 3-4 Sample routable road network with nodes, links, and shape points

3.1.1.1 Converting Links to the AequilibraE Format

The attributes of the links used by AequilibraE are slightly different from those used by GMNS. Table 3-5 shows how RDR makes the translation.

Table 3-5 Link fields for AequilibraE compared to GMNS

GMNS field	AequilibraE field	How it is translated
link_id (unique identifier)	link_id	Simple copy
from_node_id	from_node_id	Simple copy
to_node_id	to_node_id	Simple copy
facility_type (freeway, arterial, etc.)	facility_type	Simple copy
directed (direction of flow, A->B = 1)	directed	Simple copy
length (length of the link in miles)	length	Simple copy
capacity (capacity in vehicles per day per lane)	capacity_ab (capacity in vehicles per day per link)	See Note 1, below
free_speed (in mph)	free_speed	Simple copy
lanes	lanes	Simple copy
allowed_uses (e.g., AUTO, WALK, BIKE)	allowed_uses	Simple copy
toll (toll in cents)	toll	See Note 2, below
travel_time (travel time in minutes)	free_speed	See Note 3, below
geometry (geometry of the link in well-known-text, used for display in a GIS)	<i>not needed</i>	N/A
-	link_available	See Note 4, below
-	alpha	See Note 5, below
-	beta	See Note 5, below

The RDR Tool Suite includes a series of helper tools to assist the user in converting their network to one compatible with RDR and AequilibraE. Details on and direct links to the helper tools summarized below can be found in [Appendix I: Format Network Helper Tools](#).

1) Per hour to per day conversions: To convert from vehicles / hour / lane to vehicles / day / lane as is the required input units for the RDR Tool Suite:

- $capacity \left(\frac{veh}{day} \right) = peak_hr_to_day_cap_conversion * capacity$
- Peak_hr_to_day_cap_conversion is a parameter (typically between 10 and 15) that converts a peak hour capacity to a daily capacity.

Note that while AequilibraE requires capacity in vehicles per day per **link**, the RDR Tool Suite requires capacity data to be input in vehicles per day per **lane**; RDR will automatically use the “lanes” field to get a per-link value.

2) Tolls: The “toll” field for AequilibraE input should incorporate any roadway tolls (in cents), in addition to travel costs related to use of transit (transit fare on transit boarding links) and TNCs (TNC initial charge on road centroid connectors, TNC cost per mile on roadway links). The helper tools in the “format_network” subfolder can help a user apply these costs to their network.

3) Transit travel time costs: The “travel_time” field for AequilibraE input should incorporate free speed travel time (calculated from the “length” and “free_speed” GMNS fields), in addition to wait times related to use of transit (transit wait times on transit boarding and transfer links) and TNCs (TNC wait time on road centroid connectors). The helper tools in the “format_network” subfolder can help a user apply these costs to their network.

4) Link availability: The “link_available” field is a number, between 0 and 1, which indicates the availability of the link. A value of 0 indicates a link is not available, 1 indicates a link is fully available. Numbers between 0 and 1 represent a reduction in capacity. For each scenario that is run, RDR calculates “link_available” from the exposure field in a CSV file in the “Hazards” folder of the input data directory.

5) Volume delay parameters: The alpha and beta fields are the volume delay parameters. The volume-delay parameters describe how travel time increases as flow reaches capacity. AequilibraE supports several volume delay functions,¹⁶ including Bureau of Public Roads (BPR) and Speiss conical. The BPR volume/delay function has been around for many years,¹⁷ and is represented as:

$$congested_time = free_flow_time \times (1 + \alpha \times (flow/capacity)^\beta)$$

Recommended BPR parameters were historically $\alpha = 0.15$ and $\beta = 4$. Current practice is to use different values, depending on the functional class of the road, with larger values of β used for freeways and lower values used for other types of roads. In older versions of AequilibraE (e.g., version 0.6.x), the α parameter is labeled “b”, and the β parameter is labeled “power”.

Some TDMs use other options such as conical functions. The BPR functions in AequilibraE can be adjusted to closely match those of the TDM core model as needed.

There is an option in the RDR Tool Suite of using ‘link_types_table.csv’ (to be placed in the “LookupTables” subfolder of the input data folder) to fill in alpha and beta on each link. Below is an

¹⁶ <https://aequilibrae.com/python/V.0.9.2/generated/aequilibrae.paths.VDF.html>

¹⁷ Martin, William A., Nancy A. McGuckin. 1998. NCHRP Report 365: Travel Estimation Techniques for Urban Planning. Accessed 26 July 2022 from: https://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_rpt_365.pdf. Chapter 9 of NCHRP Report 365 (1998), describes the BPR function, and provides recommended parameters.

example, with notional values (Table 3-6). The default values of $\alpha = 0.15$ and $\beta = 4$ are used for those links where the facility_type is blank or does not match a facility type in the link types look-up table.

Table 3-6 Example of link_types_table.csv (with notional values)

facility_type	alpha	beta
1	0.1	5
2	0.15	4

6) GMNS link conversion: An Excel-based helper tool, 'GMNS_link_conversion.xlsx', is available to help users convert a GMNS network file into a network file compatible with the tool suite. The helper tool includes the following:

- Peak_hr_to_day_capy_conversion parameter
- Lookup table for volume delay (α and β) parameters by functional class
- Lookup table for links whose availability is less than 1

The following parameters are used by the network routing model:

- vot_per_hour: Value of time in dollars / hour. This will be converted to cents / minute and is used to factor tolls into the link impedance. This parameter is specified in the configuration file.
- allow_centroid_flows: User can select 1 to allow flows to be routed through centroids/centroid connectors (default) or 0 to block these types of paths. To allow multimodal routing of trips (e.g., route comprised of both car and transit travel), this must be set to 1. This parameter is specified in the configuration file.

The helper tool can be found here: https://github.com/VolpeUSDOT/RDR-Public/blob/master/helper_tools/GMNS_link_conversion.xlsx.

3.1.2 Road Network Data Sources

The user must have their own network data source to use the RDR Tool Suite. Sources of routable network data include (1) the Metropolitan Planning Organization (MPO) or State DOT, (2) OpenStreetMap, (3) Freight Analysis Framework, (4) Census line files, (5) Overture maps. Other sources may also be available.

3.1.2.1 MPO and State DOT data

Larger MPOs and most state DOTs have a travel demand model (TDM) that uses a routable network. Users can obtain a road network from them.

Advantages of this approach:

- The TDM network has already been cleaned up and accepted for use in regional planning.

Disadvantages of this approach:

- A network at the appropriate level of detail might not be available. For example, in a rural area, there may be no MPO network and the State DOT network might only include major roads.

- The network might not have true shapes information. A TDM does not require a true shapes network, so the MPO / State DOT network might be what is known as a stick network, with straight lines connecting each node.

3.1.2.2 *OpenStreetMap*

OpenStreetMap¹⁸ (OSM) is a freely available map of the world, developed by users. It is owned by the non-profit OpenStreetMap Foundation and is licensed under the Open Data Commons Open Database License, which states that “You are free to copy, distribute, transmit and adapt our data, as long as you credit OpenStreetMap and its contributors. If you alter or build upon our data, you may distribute the result only under the same license.”¹⁹

Arizona State University has developed osm2gmns,²⁰ a tool for extracting routable networks from OSM and generating GMNS networks. The GMNS network can then be imported into RDR.

Advantages of this approach:

- Provides an all-roads network of the United States (and also, of much of the world).
- Tools exist for extracting OSM data and generating routable networks.

Disadvantages of this approach:

- Some basic Python knowledge is required to run osm2gmns.
- The conversion is not perfect. In a test of OSM2GMNS, differentiation between one-way and two-way streets would sometimes be incorrect, leading to poor connectivity. Some manual clean-up will still be required.

3.1.2.3 *Freight Analysis Framework*

The FHWA-sponsored Freight Analysis Framework²¹ (FAF) includes a high-level FAF Model Highway Network of the United States. It is downloadable as a zip file and includes networks in TransCAD and Geodatabase formats.

Advantages of this approach:

- The network is a national highway network that has already been used for routing.

Disadvantages of this approach:

- The network only includes highways and major roads, and likely does not include enough detail for routing in a metropolitan area.

3.1.2.4 *Census Line Files*

Census TIGER line files provide another source for transportation network data. They are downloadable by state and county.

Advantages of this approach:

- Readily available and includes all roads at the County level.

¹⁸ <https://www.openstreetmap.org/>

¹⁹ <https://www.openstreetmap.org/copyright>

²⁰ <https://osm2gmns.readthedocs.io/en/latest/>

²¹ https://ops.fhwa.dot.gov/freight/freight_analysis/faf/

- Includes true shapes information.

Disadvantages of this approach:

- Not a node-link routable network.

3.1.2.5 Overture Maps

Overture Maps Foundation²² was founded in 2022. It is dedicated to producing open map datasets and released its first set of data in 2023. It appears to have a similar level of detail to maps from OpenStreetMap with proper connections for routing. Figure 3-5 shows a sample snapshot of data from Overture for Cambridge, Massachusetts.

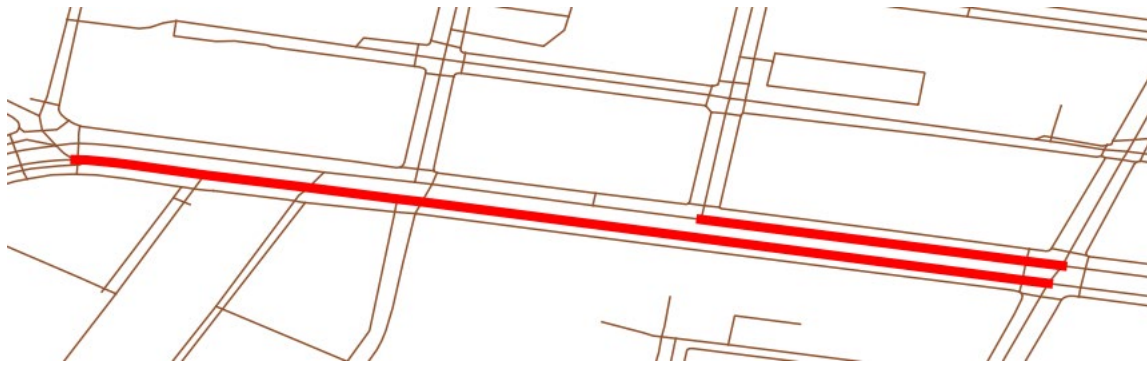


Figure 3-5 Sample download from Overture (Binney Street, Cambridge, MA)

Advantages of this approach:

- May provide an all-roads network for large regions.
- Has a node (connector) – link (segment) structure for routability.

Disadvantages of this approach:

- Very new.
- Basic SQL knowledge (and access to Amazon Web Services or Azure) is required for downloading.
- Does not seem to have a representation of one-way streets.

3.1.3 Road Network Validation

Once the user has acquired their network data, the user can perform validation checks on their network without running any models by evaluating the following:

- Do the number of nodes with node_type = centroid match the number of zones in the trip table?
- In the links table, does every from_node_id and to_node_id exist in the nodes table?
- Are the lengths reasonable (between 0 and some finite number of miles)?
- Are capacities reasonable? Since these are daily directional capacities per lane, the user would expect to see values of approximately 5,000 on a minor road, 10,000 on a major road, and

²² <https://overturemaps.org/>

20,000 on a freeway. Centroid connector capacities should be high so that they do not constrain flows to and from centroids.

- Is the free_speed greater than 0, and is it reasonable? Are free speeds on centroid connectors low (typically, 3 to 20 mph), so that flows are not routed via the centroid connectors? It may be helpful to think of the centroid connectors as representing local streets and/or parking lots that connect the origins and destinations to the main network.
- Is the number of lanes on each link at least 1?
- Are tolls (in cents) correct (will be zero on most links)?
- Are the travel times (in minutes) reasonable? They can be computed based on free_speed and distance.
- If shape points are used, do the shape points have the same coordinate system as the nodes?

3.2 Trip Table Preparation

RDR relies on daily origin-destination person flows to load the network (Figure 3-6). Data sources for these flows may include origin-destination trip tables, or production-attraction data from various sources. The National Cooperative Highway Research Program²³ (NCHRP) contains a discussion of data needs and sources for trip tables. RDR uses person trip tables, typically generated at the trip distribution step of the modeling process.

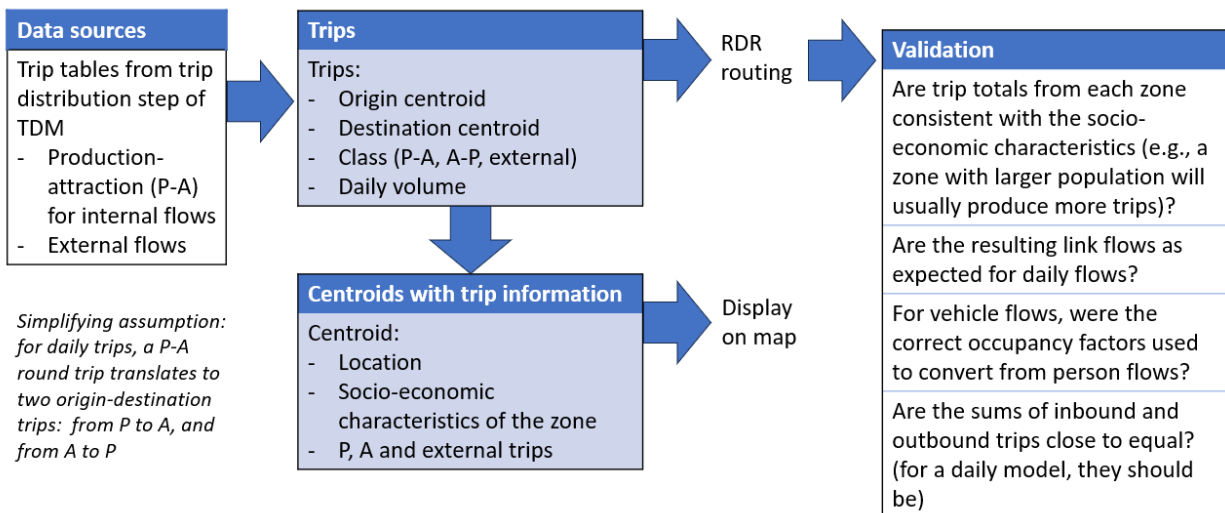


Figure 3-6 Data preparation flow for setting up trip tables

RDR expects trip tables to be in an open matrix (OMX) format. OMX is an open binary format for efficiently storing large matrices. Daily trip tables used as input by RDR can often be constructed from TDM data with model specifications in the documentation. Parameters may include appropriate conversions (if necessary) from production-attraction (home – non-home destination) trips to the actual origin and destination trips. If trip tables are not available from an existing TDM, there are several ways to synthesize them:

²³ NCHRP. 2012. Travel Demand Forecasting: Parameters and Techniques. Vol. 716. Washington, D.C: Transportation Research Board.

1. Conduct surveys of travelers, asking (at a minimum) their origins and destinations. For automobile travel, it may also be possible to survey license plates, later matching them to the locations where the vehicles are registered.
2. Estimate trips produced or attracted to various land uses, either via observations or using tools such as the ITE trip generation manual (<https://itetripgen.org>). It is then necessary to distribute the trips from each source to the various destinations.
3. Buy O-D data from a commercial provider (who typically prepares these data based on cellphone observations).

Many travel demand models can export trip tables in OMX format. RDR also provides a Format Demand Helper Tool (see [Appendix H: Format Demand Helper Tools](#) in this document), which the user can use to convert CSV and tabular trip tables (e.g., from an existing TDM) to OMX format.

3.2.1 Production-Attraction versus Origin-Destination

Trip tables typically start as production-attraction tables. According to Travel Forecasting Resource (an online compilation of information on travel forecasting), “[a] trip is produced where the need (or desire) for travel is located; and a trip is attracted where the need (or desire) is satisfied. A trip production can be either an origin or a destination, and a trip attraction can be either an origin or a destination.”²⁴

The productions are typically home locations, while the attractions are often work locations. Consider, for example, 100 people who live in suburb A, traveling to work in city B. The production-attraction table would have:

- 100 round trips, produced in suburb A, and attracted to city B.

The equivalent origin-destination table would include two entries:

- 100 one-way trips from suburb A to city B, and
- 100 one-way trips from city B to suburb A.

If the available data are in production-attraction form, it is necessary to convert it to origin-destination form. The user can do the following to make this conversion:

- Determine if the units in the production-attraction (P-A) table are single trips or round trips.
- Make a copy of the P-A table and swap the productions and attractions. These will represent the return trips (e.g., from work to home) in an origin-destination table.
- Concatenate the two tables. The original table has origin-destination trips from the production to the attraction. The added table has the return trips.
- If the original table was in single trips, divide the number of trips by 2, to avoid double counting.

3.2.2 Trip Table Data Sources

Data sources include MPO or State DOT trip table data, Census data, and synthesis from multiple traffic counts.

3.2.2.1 MPO and State DOT Data

Larger MPOs and most state DOTs have a travel demand model (TDM) that either uses or can generate trip tables. Users can obtain the trip tables from them.

²⁴ https://tfresource.org/topics/Trip_Generation.html

Advantages of this approach:

- Trip tables have already been cleaned up and accepted for use in regional planning.

Disadvantages of this approach:

- Trip tables at the appropriate level of detail might not be available. For example, in a rural area, there may be no regional model and the State DOT model might not be detailed enough.
- Need to be careful to distinguish between production / attraction and origin / destination tables.

3.2.2.2 Census Transportation Planning Products

The American Association of State Highway and Transportation Officials (AASHTO) periodically purchases a tabulation of U.S. Census data that is useful for transportation planning, including trip table development. It is available at <https://transportation.org/ctpp/ctpp-data/>.

The census also provides Longitudinal Employer-Household Dynamics (LEHD) data, which may aid with some home-work flows (Figure 3-7). It provides home and employment information, down to the block level.

See <https://onthemap.ces.census.gov/> and <https://lehd.ces.census.gov/data/#lodes> for more information.

Advantages of this approach:

- National, freely available data source.

Disadvantages of this approach:

- Only covers home-to-work trips.
- May not cover some employment types.

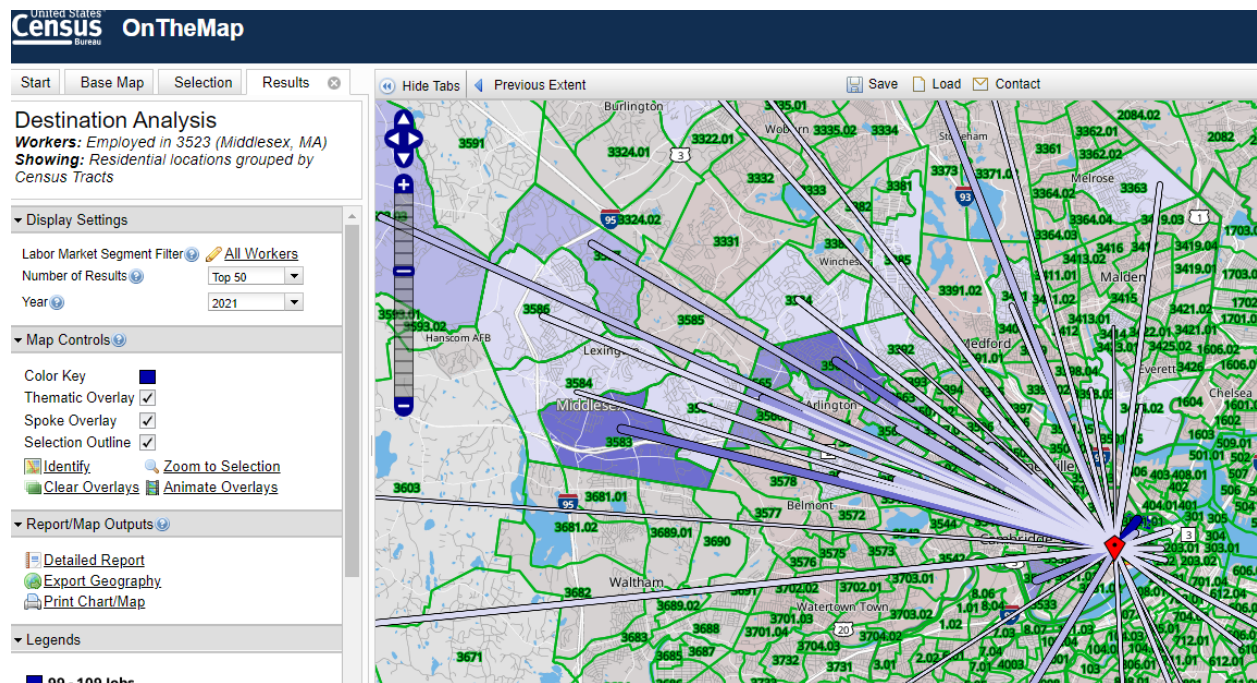


Figure 3-7 Sample snapshot of home-work flows from LEHD

3.2.2.3 Synthesis from Traffic Data

Trip tables may be derived based on traffic and travel observations. These derivations may be purchased commercially from firms that make use of cell phone data and other observations.

3.3 Trip Table Validation

The baseline network helper tool (See [Appendix F: Baseline Network Run Helper Tool](#)) routes the base network and trip tables, without any disruption scenarios. Before proceeding to running RDR scenarios, the reasonableness of the base network and trip table routing should be checked using this tool.

The output file `link_flow_full.csv` provides links, link flows, and volume-capacity ratios. Validation checks the user can apply include:

- Are trip totals (sums of flows on the centroid connectors) from each zone consistent with the socio-economic characteristics of the zone (e.g., a zone with larger population will usually produce more trips)?
- Are the sums of daily inbound and outbound trips from each zone close to equal? For a daily model, they are expected to be similar.
- Are the resulting link flows as expected for daily flows?
- Are volume-capacity ratios reasonable? One may expect to see a few greater than 1 (indicating significant congestion), but a large number greater than 1 usually indicates a network coding error (e.g., using hourly capacities with daily flows, or failing to account for vehicle occupancy in converting from person flows to vehicle flows).

The user can execute further checks by examining the skims output by the helper tool (`sp_baseyear.omx` and `rt_baseyear.omx`). The skims provide origin-destination distances and travel times. The function `review_skim`, located in [Appendix H: Format Demand Helper Tools](#), aids in reviewing skims that are in OMX format.

In addition, the baseline network helper tool produces two GeoJSON files for visualization of the network, its attributes, and the link flows in the baseline no disruption, no resilience project scenario. Both files are created in the same output folder as the `link_flow_full.csv` file, one containing node information (`node.json`) and one containing link information (`link_flow_full.json`). Both of these files can be easily imported to most GIS software for visual validation of the base network and trip table inputs.

4 Running the RDR Tool Suite

Batch files are collections of executable commands that are used to run each tool in the RDR Tool Suite; a template batch file used for RDR runs is in the main RDR directory. This ‘run_rdr_full.bat’ batch file executes the RDR Metamodel and the RDR ROI Analysis Tool (i.e., fits a metamodel and runs an analysis). Similar batch files are used to run the Exposure Analysis Tool, the Equity and Benefits Analysis Tool, and other helper tools. A batch file is also provided separately to run each Quick Start and Reference Scenario example in the corresponding “scenarios” subfolder; see the Quick Start Guide or Reference Scenario Library for next steps.

Table 4-1 describes the different modules of the RDR Metamodel and RDR ROI Analysis Tool. All of these steps are run in order in the ‘run_rdr_full.bat’ batch file.

Table 4-1: RDR Metamodel and ROI Analysis Tool Modules

RDR Tool Suite Module	Function
‘lhs’ (Metamodel Parameterization – Latin Hypercube Sampling)	Select core model runs to conduct with AequilibraE
‘aeq_run’ (Transportation Disruption, Core Model Runs)	Calculate core model results using AequilibraE for selected runs
‘aeq_compile’ (Core Model Runs)	Compile core model results
‘rr’ (Scenario Expansion – Metamodel Regression)	Construct regression model for full set of scenarios
‘recov_init’ (Scenario Expansion – Exposure Recovery, Damage and Repair Recovery)	Build out hazard exposure, damage, and repair recovery for set of scenarios
‘recov_calc’ (ROI Analysis, Reporting and Visualization)	Consolidate metamodel outputs and calculate metrics across period of analysis
‘o’ (Reporting and Visualization)	Write report for RDR Tool Suite run

To use a batch file to execute a ROI analysis run, the user should follow these steps (similar steps should be followed to use the other tools in the suite):

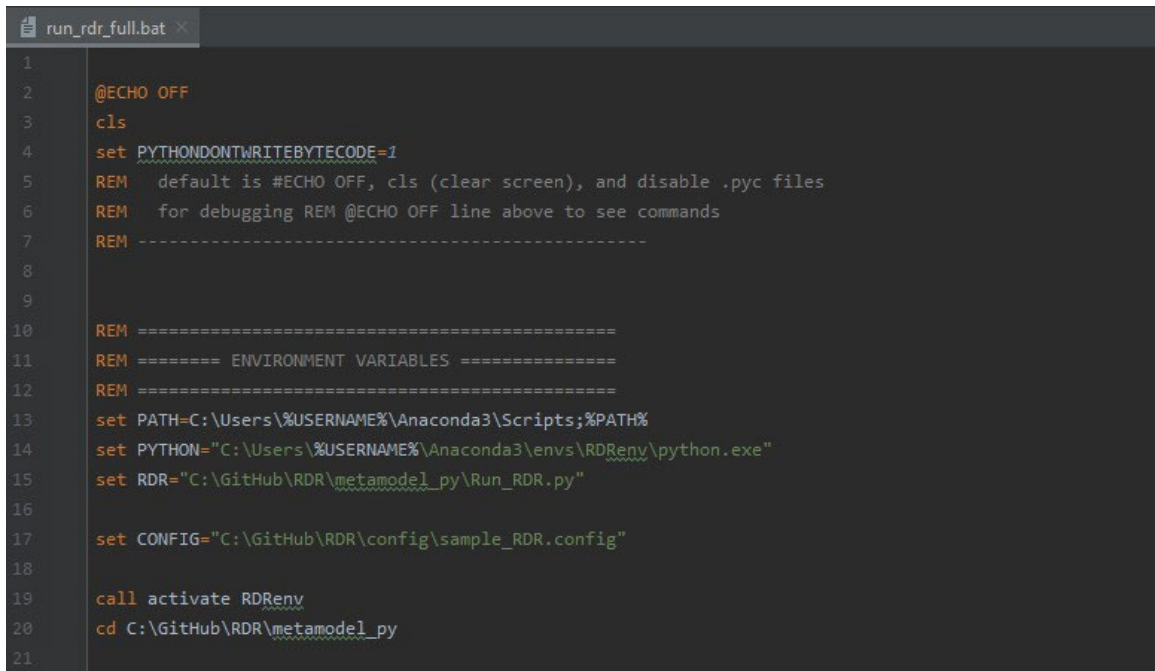
1. Open the batch file in a text editor (e.g., Notepad, Notepad++, PyCharm, etc.).
2. Edit the environment variables and “change directory” command found in lines 13 to 20 of the batch file (see Figure 4-1):
 - a. Edit the ‘PATH’ variable at line 13 of the batch file to include the full path of the “Scripts” subfolder of the user’s conda installation. (See note below for tips to find the location of the user’s conda installation.) This is required for the batch file to activate the RDR environment. Make sure the command appends to the existing ‘PATH’ variable, i.e., line 13 should always end with “; %PATH%”.
 - b. Set the ‘PYTHON’ variable at line 14 of the batch file to the full path of the user’s local Python executable for the RDR environment. (Note: For the batch files of some tools, e.g., the Exposure Analysis Tool, this should be set to the user’s ArcGIS Python executable instead.) The Python executable can be found in the user’s conda installation, within the RDR environment created in Section 2.3. (Note: To find the location of the user’s local Python executable, open an Anaconda Prompt terminal window. Run the commands:

- activate RDRenv
- where python

The second command should return the full path of the user's local Python executable for the "RDRenv" conda environment, ending in `RDRenv\python.exe`).

- c. Set the 'RDR' variable at line 15 of the batch file to the full path of the main Python script 'Run_RDR.py' used to run the RDR Tool Suite. The Python script can be found in the "metamodel_py" subfolder of the user's RDR directory.
 - d. Set the 'CONFIG' variable at line 17 of the batch file to the full path of the user configuration file for the run. Make sure the file location and name entered match the configuration file corresponding to the analysis being run by the batch file.
 - e. Set the directory in the "cd" command in line 20 of the batch file to the "metamodel_py" subfolder of the user's RDR directory, which contains the Python and R code files that comprise the RDR Tool Suite.
3. (Optional) There is a pause in the batch file at the end of executing a run (e.g., for debugging or confirming the run was executed as expected). To remove the pause, comment out line 60 of the file by adding "REM " in front of the command "pause" (see Figure 4-2). Note that there is a space at the end of "REM ". The pause command at line 68 is only reached if an error occurs during the run, though the user may wish to comment that "pause" command out as well.
 4. Save and close the batch file.
 5. Execute the batch file using one of the following two methods (Note: The user's initial run of the tool suite may take longer than usual as it installs necessary software packages):
 - a. Double-click on the batch file. A terminal window should open automatically.
 - b. Manually execute the batch file:
 - Open any terminal window (e.g., Command Prompt, Terminal, Anaconda Prompt).
 - Drag and drop the batch file into the open terminal window.
 - Click into the terminal window and press Enter.

For either method of executing the batch file, a set of informational logging statements will be output in the terminal window, indicating progress of the run as well as any error messages. If the user would like to redirect the terminal window statements to a text file, they should use the second method laid out in 5(b) and append " > myoutput.txt 2>&1" to the path to the batch file (e.g., "run_rdr_full.bat > myoutput.txt 2>&1"). Instead of printing to the terminal window, logging statements will be stored in the designated text file. Detailed logging is also provided in the "logs" subfolder of the output data folder specified in the configuration file.

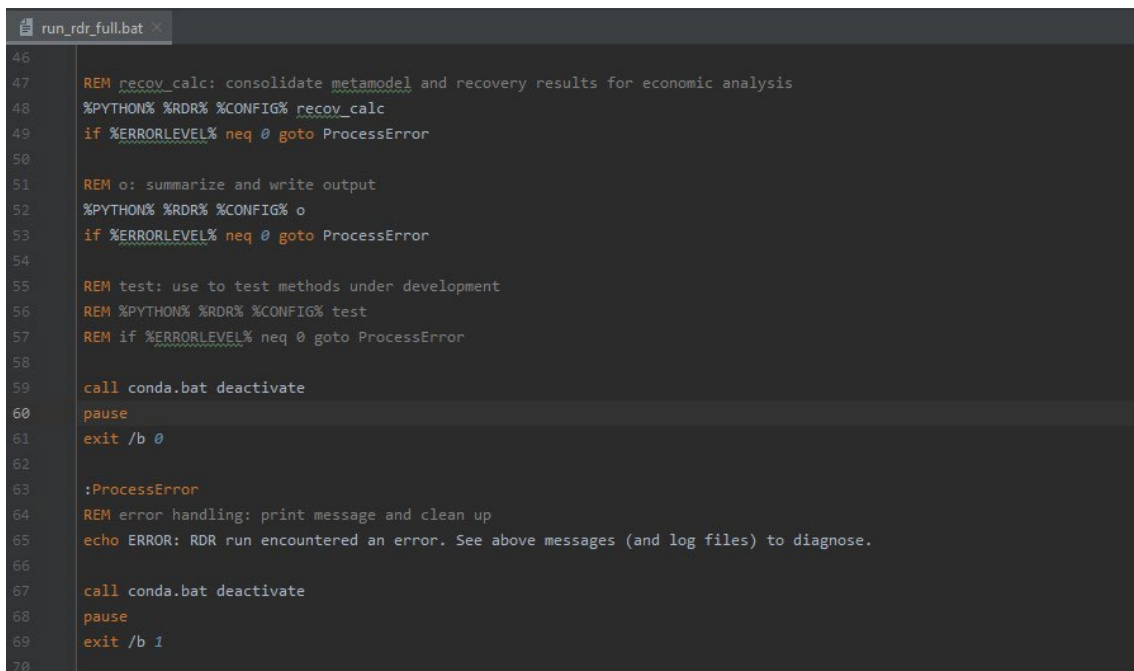


```

1
2  @ECHO OFF
3  cls
4  set PYTHONDONTWRITEBYTECODE=1
5  REM default is #ECHO OFF, cls (clear screen), and disable .pyc files
6  REM for debugging REM @ECHO OFF line above to see commands
7  REM -----
8
9
10 REM =====
11 REM ===== ENVIRONMENT VARIABLES =====
12 REM =====
13 set PATH=C:\Users\%USERNAME%\Anaconda3\Scripts;%PATH%
14 set PYTHON="C:\Users\%USERNAME%\Anaconda3\envs\RDRevy\python.exe"
15 set RDR="C:\Github\RDR\metamodel_py\Run_RDR.py"
16
17 set CONFIG="C:\Github\RDR\config\sample_RDR.config"
18
19 call activate RDRevy
20 cd C:\Github\RDR\metamodel_py
21

```

Figure 4-1: Environment Variables in Batch File



```

46
47 REM recov_calc: consolidate metamodel and recovery results for economic analysis
48 %PYTHON% %RDR% %CONFIG% recov_calc
49 if %ERRORLEVEL% neq 0 goto ProcessError
50
51 REM o: summarize and write output
52 %PYTHON% %RDR% %CONFIG% o
53 if %ERRORLEVEL% neq 0 goto ProcessError
54
55 REM test: use to test methods under development
56 REM %PYTHON% %RDR% %CONFIG% test
57 REM if %ERRORLEVEL% neq 0 goto ProcessError
58
59 call conda.bat deactivate
60 pause
61 exit /b 0
62
63 :ProcessError
64 REM error handling: print message and clean up
65 echo ERROR: RDR run encountered an error. See above messages (and log files) to diagnose.
66
67 call conda.bat deactivate
68 pause
69 exit /b 1
70

```

Figure 4-2: End of Batch File

4.1 Executing Additional Runs

To use existing core model runs from a previous RDR analysis along with new AequilibraE core model runs, either due to a change in the scenario space or to provide more sampled points to the regression model, the user can re-run the tool suite with the same 'run_id' parameter in the configuration file and the same output directory specified. The 'lhs' module will use all scenarios that have already been run

with that run_id and will select more scenarios to run with the core model, and the 'aeq_run' module will create more AequilibraE outputs in the same output directory labeled by the 'run_id' parameter. The 'aeq_compile' module references this shared output directory to compile core model runs used to fit the regression model. New output files will overwrite any files of the same name that were generated by previously-run analyses. If the user wishes to save the output files from a previously-run analysis, they should make renamed copies or move them out of the output folder.

When the user runs the tool suite with the same run_id as a prior run, the tool will present a warning message, as follows, and the user will need to provide input on whether to proceed or cancel, depending on whether any changes are allowable:

"The run_id specified was already used for a prior run. Core model runs with the same run_id will be combined in this RDR run. There are only certain changes in the inputs that are allowable, including the following

- *Configuration file changes:*
 - *Start year, end year, base year, future year*
 - *Metamodel type*
 - *LHS sample target*
 - *AequilibraE run type*
 - *[recovery] section parameters*
 - *[analysis] section parameters except for Value of Travel Time*
- *Addition of new scenario dimensions or removal of prior scenario dimensions (e.g. hazards, economic scenarios, trip loss elasticities, resiliency projects), and corresponding edits in the following input files to capture addition/subtraction of scenario dimensions:*
 - *UserInputs.xlsx*
 - *Model_Parameters.xlsx*
 - *Base year core model runs file*
 - *Resilience projects files*
- *Addition of new input files to correspond with new scenario dimensions identified above:*
 - *Exposure analysis file ([Filename].csv) for each new hazard event*
 - *Link file ([econ][projgroup].csv) for each new combination of economic scenario and project group*
- *Edits to exposure-damage table file (optional input)*
- *Edits to repair cost and time lookup table files (optional inputs)*

Consult the RDR checklist or other materials in the RDR 'documentation' folder for help. If you can confirm that you have not made any changes aside from what is listed above, you may press 'r' to continue with this run. This will add new runs to the existing core model runs and overwrite any files of the same name that were generated by previously-run analyses. While you can add or subtract scenario dimensions, you CANNOT change the details for scenario dimensions that existed in the prior run with the same name. If you do not want to combine with previous core model runs, press 's' to stop this run and try again with a different run_id.

Press either 'r' (to continue run) or 's' (to stop run) followed by the 'Enter' key after reviewing the above."

5 RDR Exposure Analysis Tool

An exposure analysis applies geospatial information regarding hazard severity onto the transportation network to assess hazard exposure severity on the network links specifically; therefore, an exposure analysis should be performed on a geospatially accurate network dataset. The goal of the RDR Exposure Analysis Tool (`exposure_grid_overlay.py`) is to automate the process of assessing disruption to a road or transit network based on exposure to a hazard, using Geographic Information Systems (GIS). If the network is a GIS export of a Travel Demand Model (TDM) or a transit network derived using the Format Network Helper Tool, the output from this tool can be leveraged by the Resilience and Disaster Recovery Metamodel (RDRM). (Note that if the user's network involves both road and transit subnetworks, we recommend running the tool separately on each subnetwork and then merging the output files into one exposure file for use by the RDRM.) Additionally, this tool can also be used as a standalone script; it can be useful for creating, analyzing, and visualizing specific exposure and disruption scenarios without running the full RDRM. The script is written to work on a Windows computer in conjunction with an installation of ArcGIS (v 10.x) or ArcGIS Pro, with the Spatial Analyst extension. The script is designed for an intermediate or advanced GIS user who is already familiar with running custom Python scripts. The script must be run using Windows Command Prompt or PowerShell. The user may also wish to use an Integrated Development Environment (IDE) such as IDLE (which is provided with ArcGIS) or PyCharm to make custom, optional modifications to the code.

The script code is also available on GitHub at https://github.com/VolpeUSDOT/RDR-Public/tree/main/helper_tools/exposure_grid_overlay. The script can be modified by the user to suit specific situations. As an added benefit, the RDR Exposure Analysis Tool allows the user to experiment with various methodologies for converting exposure to disruption.

5.1 Configuring the Exposure Analysis GIS Network

If using a TDM network to conduct an exposure analysis, it is important to assess the source of the network and its geographic accuracy. Some TDM roadway networks, sometimes called stick-networks, do not accurately reflect the geographic location of the roadways. Figure 5-1 below shows a screenshot of a section of the TDM network (red lines) overlaid on top of a road map in ArcGIS in the Virginia Beach area of Virginia. While some of the segments match the ArcGIS road map fairly well, there are some components of the TDM network that are much coarser than the real-world network (Figure 5-2 below is zoomed in on one such area). If this network were used in a GIS-based hazard exposure analysis (e.g., flooding), it would be inaccurate.

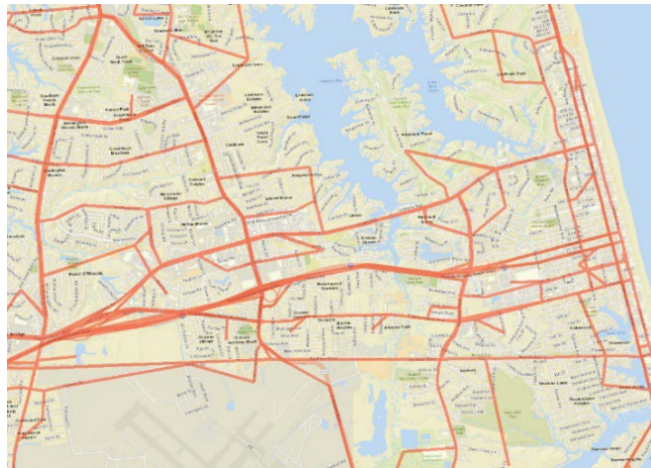


Figure 5-1: Virginia Beach TDM Network Alignment with ArcGIS Basemap

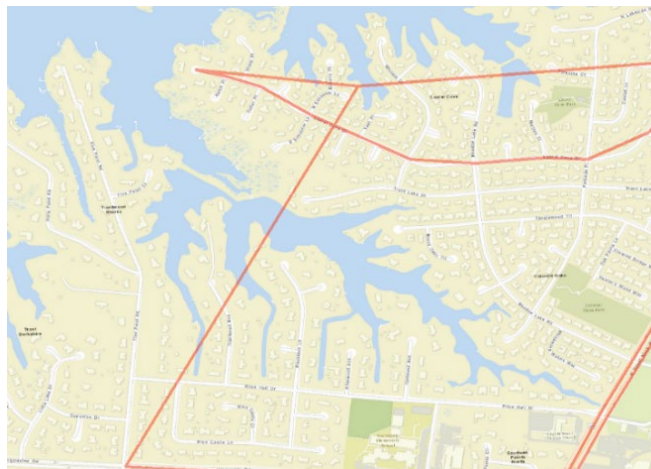


Figure 5-2: TDM Misalignment with ArcGIS Basemap

If the network is schematic and geographically inaccurate, the user should first try to export a more geographically accurate network from the TDM. If this is not possible, the user will need to source a more geospatially precise network. The user will then need to translate the geographically accurate network to the TDM network, either manually or via a crosswalk linking individual GIS-based assets to TDM-based assets. The user must then manually transfer over the changes in capacity and speed to the TDM network for analysis.

A user may want to integrate other attributes from separate GIS-based transportation networks into a TDM network. Commonly used GIS software such as Esri's ArcGIS Pro (and the older ArcGIS) contain geoprocessing tools that can help facilitate this integration. For example, the **Transfer Attributes** tool can be used to spatially match a distinct roads dataset with the GIS export of the TDM. The network dataset might contain additional attributes that are not native to the TDM network, but which could be helpful in modeling different hazard scenarios (e.g., roadway segments might include attributes on road pavement, surface type, international roughness index (IRI), and soil type).

Additional information on defining exposure data and potential geospatial data sources can be found in [Appendix D: Defining Exposure Data](#) and [Appendix E: Transportation Asset and Hazard Scenario Data Summary](#).

5.2 Configuring the RDR Exposure Analysis Tool

After the hazard data and transportation network data have been identified, the final step is to bring the data together in order to determine the level of exposure associated with each asset in the network. Figure 5-3 below represents the kind of data produced in the exposure analysis in graphical form.

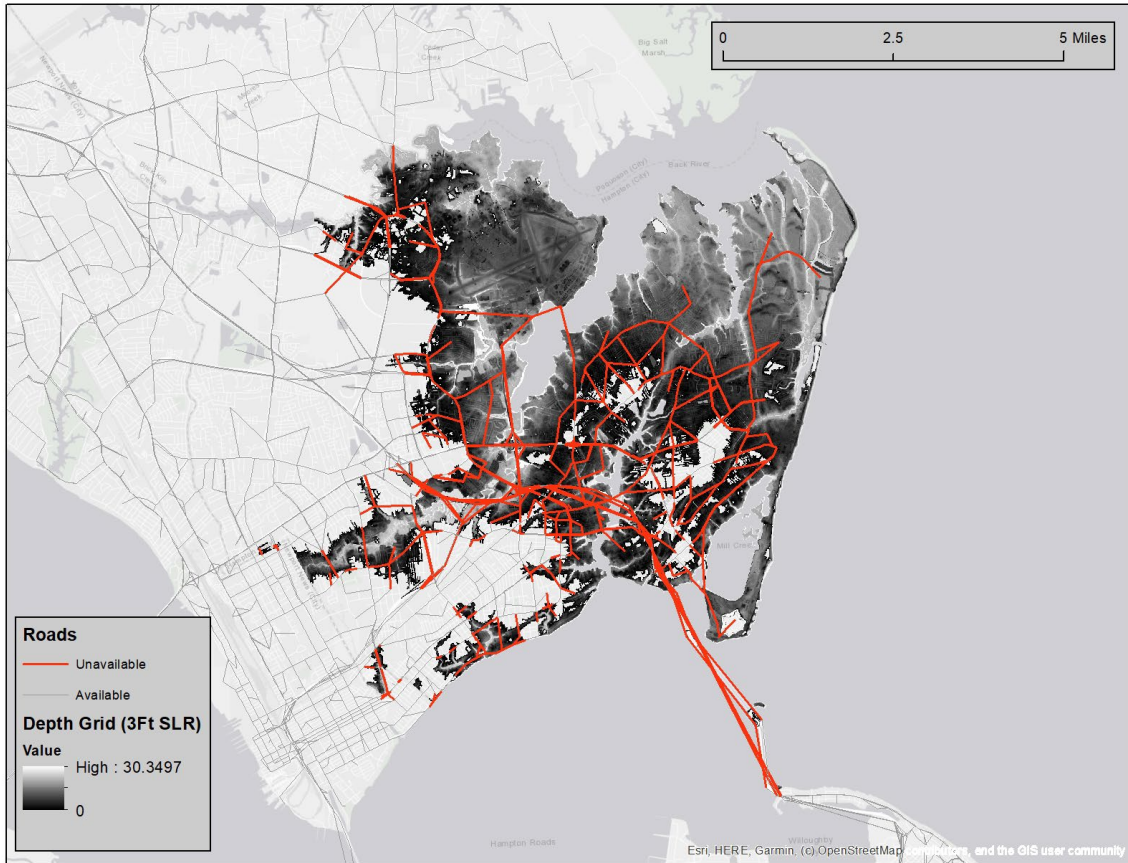


Figure 5-3: Example output from a simple GIS Exposure Analysis overlaying depth grid data onto the TDM transportation network to identify disrupted links, assuming all exposure leads to disruption.

The input that the RDRM needs in order to leverage exposure data is a comma-delimited table that provides the unique asset identifier and specific exposure level for each asset of interest. The asset ID must match the relevant asset ID in the TDM so that these data can be appropriately linked together.

If a user has an existing GIS network that includes exposure analysis data linking assets to hazard data, these data could be integrated into the TDM network using Esri's ArcGIS Pro to create the exposure analysis files. The Exposure Analysis Tool provides an automated, user-configurable method for integrating these data.

The user will need to modify certain configuration parameters within a standalone configuration file in order to get the tool up and running. A template configuration file ("exposure_grid_sample.config") accompanies the tool, but the file can be edited and renamed as appropriate for the specific region and hazard scenario. These customizable parameters include the following:

- **Input Exposure Grid** (input_exposure_grid)—the full path to the raster formatted input exposure grid associated with the hazard scenario of interest. One hazard for which these sorts of datasets are commonly available is flooding, but any GIS-formatted gridded dataset representing exposure to a hazard (in which higher numeric values represent higher levels of exposure to a hazard) can be utilized. In the case of flooding, depth grids may be published by government entities, such as NOAA, FEMA, states, and local Metropolitan Planning Organizations (MPOs) or created as an output of modeling tools such as FEMA’s HAZUS tool.
- **Input Network** (input_network)—path to the input road or transit network GIS-based feature class (shapefile, file geodatabase feature class, etc.).
- **Output Directory** (output_dir)—the folder where outputs will be stored. The output directory will be created if it does not exist. If the directory does exist, any existing outputs in this directory will be overwritten.
- **Run Name** (run_name)—scenario name with no spaces used to name all file outputs. Choose a descriptive run name that distinguishes it from other hazard scenarios (i.e., “100_yr_flood_hazard_scenario”).
- **Link Availability Approach** (link_availability_approach)—the approach used to convert exposure into a corresponding level of disruption for each individual segment in the network. The current options are:
 - ‘Binary’ where any value above 0 is considered full exposure and link will not be available. Other links will remain fully available. This method is used when detailed information about the relationship between the level of hazard exposure and disruption is unavailable or the user wants a “worst case” estimate of disruption.
 - ‘Default_Flood_Exposure_Function’—utilizes depth-disruption function adapted from an existing function (Pregolato et al. 2017), where the availability of a roadway decreases from 100% to 0% at a linear rate between 0 and 300 millimeters of flood depth.²⁵ If this option is chosen, the exposure_unit must be defined so that units can be properly converted to millimeters. This function is only applicable to flooding depth grids, not other exposure grids. This method can be used when a simple linear function converting flood depth to disruption is desired, without the complexity of a manual approach or custom exposure-disruption function.
 - ‘Manual’—user-defined bins representing the conversion of exposure into link availability. If this option is chosen, the link_availability_csv path must be defined. This method is used when the user does not have a specific exposure-disruption function but can still estimate categories of disruption based on exposure.
 - ‘Facility_Type_Manual’ —user-defined bins representing the conversion of exposure into link availability by facility type. If this option is chosen, the link_availability_csv path must be defined. This method is used when the analyst does not have a specific exposure-disruption function but can still estimate categories of disruption based on the combination of exposure bins and facility type. For example, if a hazard will disrupt bridges and other network assets differently, this approach can be used to distinguish these link availability impacts. Facility types not represented in the CSV file are assumed not disrupted.

²⁵ Pregolato, M., Ford, A., Wilkinson, S., & Dawson, R. (2017). “The impact of flooding on road transport: A depth-disruption function.” *Transportation Research Part D: Transport and Environment*, Vol. 55, pp.67-

81. <https://www.sciencedirect.com/science/article/pii/S1361920916308367>

- 'Beta_Distribution_Function'-- allows the user to define a custom exposure-damage function for any hazard for which severity grids exist. The goal is to give the user flexibility in determining how link availability could change in a non-linear fashion due to varying levels of exposure. Due to the complexity of this approach compared to the others, there are several additional configuration parameters that must be defined if this approach is selected. These fields include 'alpha', 'beta', 'lower_bound', 'upper_bound' and 'beta_method' (described in further detail below). The [Keisan Online Calculator](#) can be used to help the user test different alpha and beta parameters which define the shape of the customized beta distribution. The user must define the bounds of values where link availability is expected to be partially impacted by a particular hazard (values outside these bounds will default to 0% availability or 100% availability depending on the beta method chosen).
- **Exposure Field** (exposure_field)—this is the field name associated with the exposure level for each grid cell in the exposure grid. The most common field name associated with gridded GIS-data is 'Value' but depending on the specific exposure grid, it may differ. Confirm by examining the details of the exposure grid within ArcGIS.
- **Search Distance** (search_distance). Search distance for determining the maximum exposure along a network segment (include units—e.g., feet, meters, or yards)—the user is strongly encouraged to define a search distance that is half of the raster's grid size, to ensure that all overlapping exposure levels are taking into account.
- **Network Fields to Keep** (fields_to_keep). The fields/column names that the user would like to keep in the final csv output. If using the results of this tool as an input for the RDR Metamodel, ensure that this includes a unique ID field called 'link_id'. If this does not exist already, create it before running this tool.
- **Comment Text** (comment_text)—text to describe the hazard scenario. If not needed, this can be left blank.
- **Evacuation** (evacuation)—indicates whether or not the user wishes to flag evacuation routes as part of the analysis. Needs to be set to True or False.
- **Evacuation Input** (evacuation_input)—if evacuation is set to True, then this parameter defines the path to the GIS feature class representing evacuation routes.
- **Evacuation Route Search Distance** (evacuation_route_search_distance)—a distance in which to search around evacuation routes in order to identify evacuation routes in the input network. Any network segments existing completely within the search distance of the evacuation input will be flagged as evacuation routes.
- **Emergency** (emergency)—indicates whether or not the analysis represents emergency vehicle and/or high clearance exposure/disruption scenario. Needs to be set to True or False. If True, this simply modifies the format of the output to make it clear that the link availability calculated is for emergency vehicles.

The following parameters are located in the configuration file but only need to be defined if the relevant link availability approach is selected.

- **Specific Configuration Parameters for 'Manual' Link Availability Approach**
 - **Link Availability CSV** (link_availability_csv)—must be defined if manual or facility_type_manual link availability approach is being utilized. The user should base

this off of the corresponding template csv (“exposure_grid_manual.csv” or “exposure_grid_facility_type_manual.csv”) that accompanies the script.

- **Specific Configuration Parameters for ‘Default_Flood_Exposure’ Link Availability Approach**
 - **Exposure Unit** (exposure_unit)—must be defined if default flood exposure link function is being utilized. Compatible units are feet, yards, and meters.
- **Specific Configuration Parameters for ‘Beta_Distribution_Function’ Link Availability Approach**
 - **Alpha** (alpha)—a number greater than 0 that helps define the shape of the beta distribution.
 - **Beta** (beta)—a number greater than 0 that helps define the shape of the beta distribution.
 - **Lower Bound** (lower_bound)—the exposure value where link availability reaches 0% (if lower cumulative beta distribution function is utilized) or 100% (if upper cumulative beta distribution function is utilized).
 - **Upper Bound** (upper_bound)—the exposure value where link availability reaches 100% (if lower cumulative beta distribution function is utilized) or 0% (if upper cumulative beta distribution function is utilized).
 - **Beta Method** (beta_method)—user can select ‘lower cumulative’ or ‘upper cumulative’. If ‘lower cumulative’, link availability reaches 0% at lower bound and 100% at upper bound. If upper cumulative, link availability reaches 100% at lower bound and 0% at upper bound. The shape of the curve defining link availability between the lower and upper bounds is determined by the alpha and beta parameters. For either method, link availability will always be 100% for links with no hazard exposure.

In order for the script to run most efficiently, the user should define local paths for all inputs and outputs, rather than remote network paths.

5.3 Running the Exposure Analysis Tool

In order to run the tool once the configuration file is prepared, the “exposure_grid_overlay_run.bat” file must be edited to point to the following three files:

1. The Python executable associated with the user’s installation of ArcGIS or ArcGIS Pro (e.g., “C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\python.exe” in the case of ArcGIS Pro 3)
2. The location of the script on the user’s machine
3. The location of the specific configuration file on the user’s machine

The user can then open the Windows Command Prompt interface or Windows PowerShell. For instance, for the Command Prompt, go to the Windows Start Menu, type Command Prompt in the search bar, click on the Command Prompt icon and a new Command Prompt will open. Then drag the batch file into the shell, and press enter to run (the user may need to press enter more than once).

Once the tool begins to run it will leverage a series of geoprocessing tools built into ArcGIS. The script will print several brief log messages to document its status. The full process is summarized below:

1. Prepares the output directory and geodatabase, deleting any existing outputs.
2. Extracts the exposure grid raster values that overlap the network.
3. Converts these raster cells into vector-based point data.

4. Associates the maximum exposure grid point within the user-specified search distance of each network segment.
5. Depending on the link availability approach set by the user, the code will convert exposure levels to a value which determines how much of each link's capacity is available given the hazard.
 - a. If the binary approach is selected, any network segment with an exposure above 0 is assigned a link availability of 0. Any network segment without any hazard exposure is assigned a link availability of 1 (fully available).
 - b. If the default flood exposure function is selected, then the exposure values will be computed using the function as defined earlier in this documentation.
 - c. If the manual approach is selected, the link availability CSV is read and processed to convert values that fall between each pair of min_inclusive and max_exclusive values into their corresponding link_availability value. For example, in the example CSV provided below, exposure values greater than or equal to 0 and less than 0.1 will be converted to a link availability value of 1 (no disruption). Subsequent ranges of values will be converted to lower values of link availability until exposure values greater than or equal to 1 are converted to a link availability of 0 (fully disrupted).

	A	B	C
1	min_inclusive	max_exclusive	link_availability
2	0	0.1	1
3	0.1	0.2	0.8
4	0.2	0.5	0.5
5	0.5	1	0.1
6	1	999999	0

Figure 5-4: An example of a manual link availability CSV file

- d. If the facility type manual approach is selected, the link availability CSV is read and processed to convert values that fall between each pair of min_inclusive and max_exclusive values into their corresponding link_availability value, for every facility_type listed. If the facility_type is not indicated in the CSV file, the link is assumed fully available.
- e. If the beta distribution function is selected, the specific beta distribution parameters are processed to convert values that fall within the lower and upper bounds to their corresponding lower or upper cumulative distribution function of the beta distribution. Values that fall outside of the bounds will convert to 0 or 1 depending on whether the lower or upper cumulative distribution function is selected. For either method, link availability will always be 100% for links with no hazard exposure.
6. The tool outputs a GIS feature class of each network segment with its corresponding link availability and exposure value.
7. If evacuation is set to True, an additional column of data (evacuation_route) is output which indicates for each network segment whether or not the segment is part of an evacuation route. (1 indicates that it is part of evacuation route, 0 indicates that it is not). If evacuation is set to True, then an evacuation input and evacuation route search distance must also be defined. Alternatively, if the input network already has information as to whether or not each segment is

part of an evacuation route, the user can feel free to calculate the `evacuation_route` column on their own (ensuring each segment is represented by a 1 or 0 indicating whether or not it is part of an evacuation route), and simply carrying through `evacuation_route` field as one of the “fields_to_keep” defined in a separate parameter. In this case, the user can set `evacuation` to `False`, as there is no need for the tool to make the evacuation route determination again.

8. If `emergency` is set to `True`, then the “link_availability” field in the output is modified slightly to read “link_availability_emergency”, indicating that this is an emergency vehicle/high clearance vehicle scenario.
9. A simple comma-separated values (CSV) file is also generated which provides the exposure level and link availability for each segment in the network. The user can join this back to the original network for more comprehensive analysis or feed this CSV into the RDR Metamodel. Note that if running analyses on both road and transit networks, these outputs should be merged before feeding back into the RDR Metamodel (one exposure data CSV per network).

Depending on the size of the inputs, the final run time may vary from a few minutes to more than an hour.

5.4 Considerations and Limitations

- This script has been tested on several combinations of GIS datasets (including depth grids published by Hampton Roads Transportation Planning Organization (HRTPO) and the Federal Emergency Management Agency (FEMA), and road networks provided by both HRTPO and the State of North Carolina). However, the script is not guaranteed to work on every combination of road or transit network and exposure grid. An advanced GIS user with coding experience may need to make adjustments to the script in order to get the script working with their specific datasets.
- The script currently models hazard exposure effects on network links (e.g., not other infrastructure like signaling or stations). It is not intended to provide exposure information on artificial links not representing a realistic network path, such as centroid connectors.
- The output generated by the script is limited by the precision of the input datasets. For example, if the exposure grid is not high-resolution (e.g., 30-meter resolution) than the exposure values for each road or transit segment are likely to be less reliable.
- The user may wish to manually edit the output to adjust the link availability in areas where the exposure grid is not accurately determining the availability of a link (e.g., bridges may be elevated above the inundation).
- Depending on the size of the road or transit network and exposure grid, this process may take several minutes or hours to run. It is recommended that exposure grids and networks are clipped beforehand to the area of interest to minimize run time. For large regional networks and detailed exposure datasets, run times could be upwards of an hour, but the script will run much faster with smaller networks.
- The script is currently designed to be run in its entirety with one command. Users with some basic Python experience may wish to only run portions of the script and are welcome to comment out or otherwise separate or modify portions of the code in order to—for example—run the first part (exposure piece) or second part (the link availability/disruption piece) in isolation.
- Ensure that all relevant GIS data used as inputs in this tool is not actively open within ArcGIS or ArcGIS Pro before running.

The RDR Exposure Analysis Tool can be found at https://github.com/VolpeUSDOT/RDR-Public/tree/master/helper_tools/exposure_grid_overlay.

6 RDR Return on Investment Analysis Tool

6.1 Data Inputs

The RDR Tool Suite relies on several required inputs, in addition to a few optional inputs the user can provide for a more case-specific analysis. This section provides an in-depth description of these inputs. Several samples of data inputs are provided in the Quick Start example and Reference Scenarios detailed in the Quick Start Guide and Reference Scenario Library. The user is recommended to start with a copy of a sample scenario as a template for their own scenario. The user should copy a scenario folder into their RDR “Data” directory, rename the folder to provide a descriptive name, and replace each input file with their own data to execute a run that reflects their transportation scenario and hazards of interest. The RDR Tool Suite also contains an input validation helper tool to aid the user in checking that they have all required input files with the necessary columns and data types; more information can be found in Section 6.1.1.4.

All input files should be placed in the data input folder specified in the user configuration file. The user needs to ensure the following subfolders are within the input data folder:

1. “AEMaster” – Used as a template for the directory structure created for each AequilibraE core model run. Contains demand files and the SQLite database used by AequilibraE.
 - a. Within the “AEMaster” folder, the user should make sure a “matrices” subfolder exists containing demand OMX files.
2. “Hazards” – Contains exposure analysis files.
3. “LookupTables” – Contains resilience project files, in addition to being the recommended location for all user-created look-up tables used by the RDR Tool Suite.
4. “Networks” – Contains network attribute files.

An example of the structure of the input data folder is shown in Figure 6-1.

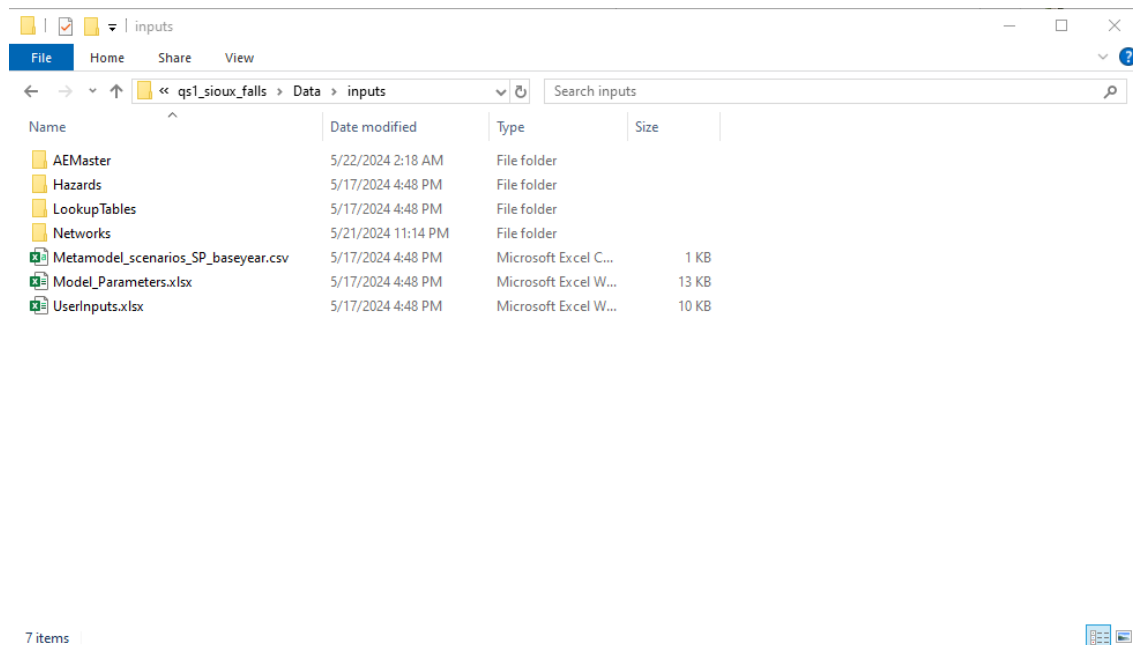


Figure 6-1: Input Data Folder Structure

Figure 6-2 shows the substructure of the “AEMaster” subfolder.

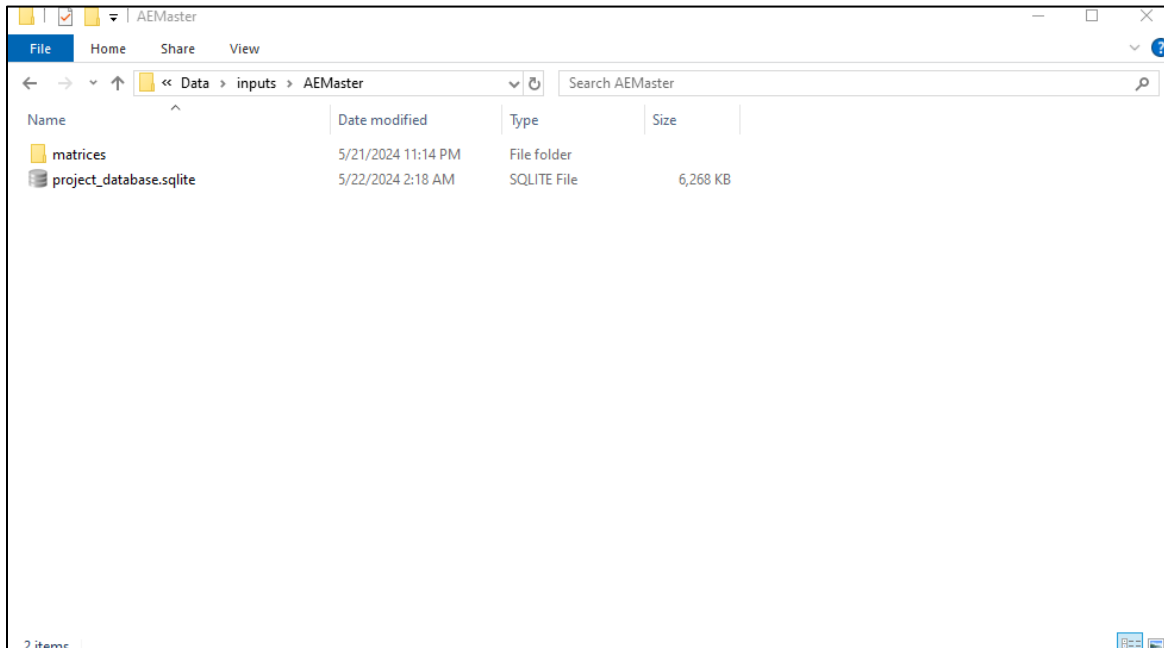


Figure 6-2: AEMaster Subfolder Structure

[Appendix A: RDR Tool Suite Input Files](#) provides a comprehensive list of all of the types of input files, where they should be located, what they should be named, required fields, and which RDR ROI Analysis Tool modules use them. Depending on the set of modules the user wishes to run in their analysis, they will need to provide the corresponding input files.

6.1.1 Configuring Scenario Definition Inputs

The input parameters defining the uncertainties modeled in an RDR analysis are specified by the user using three distinct input files, each detailed in one of the following subsections.

The specific dimensions used by the RDR Tool Suite to build out the scenario space are:

- Hazard events
- Hazard recovery stages
- Hazard event frequency factors
- Economic scenarios
- Trip loss elasticities
- Project groups
- Resilience projects

The model parameters file (Section 6.1.1.2) defines these dimensions for the RDR Metamodel; the user inputs file (Section 6.1.1.3) defines these dimensions for the RDR ROI Analysis Tool. The configuration file (Section 6.1.1.1) further parameterizes the scenario space by building out the list of possible hazard recovery periods. The model parameters file also provides relationships between project groups and resilience projects, and between hazard-related dimensions.

The look-up table given in the ProjectGroups tab of the model parameters file defines the mapping between the project group categories, which represent sets of potential resilience projects that have been grouped together, and the individual resilience projects. The project group categories are also used to specify and group future network attributes that encompass any new roads and assets built for planned infrastructure projects in the project group. If the user does not have pre-defined groups of resilience projects, they can create a single project group using an arbitrary project group name that encompasses all potential resilience projects.

The table defined in the Hazards tab of the model parameters file provides specifications for each hazard event. The RDR Tool Suite automatically builds out hazard recovery stages during the hazard period based on user parameters provided in the configuration file and the number of recovery stages specified in the model parameters file; the recovery module selects potential recovery stages for each hazard event based on these parameters. The “Event Probability in Start Year” field is used in the benefit-cost analysis (BCA) to evaluate the performance of resilience projects across different hazard events.

6.1.1.1 Configuration File

The configuration file is the primary input file for running the RDR Tool Suite, specifying the parameters that define a run of the tool suite. The user should modify the parameters for their specific analysis.

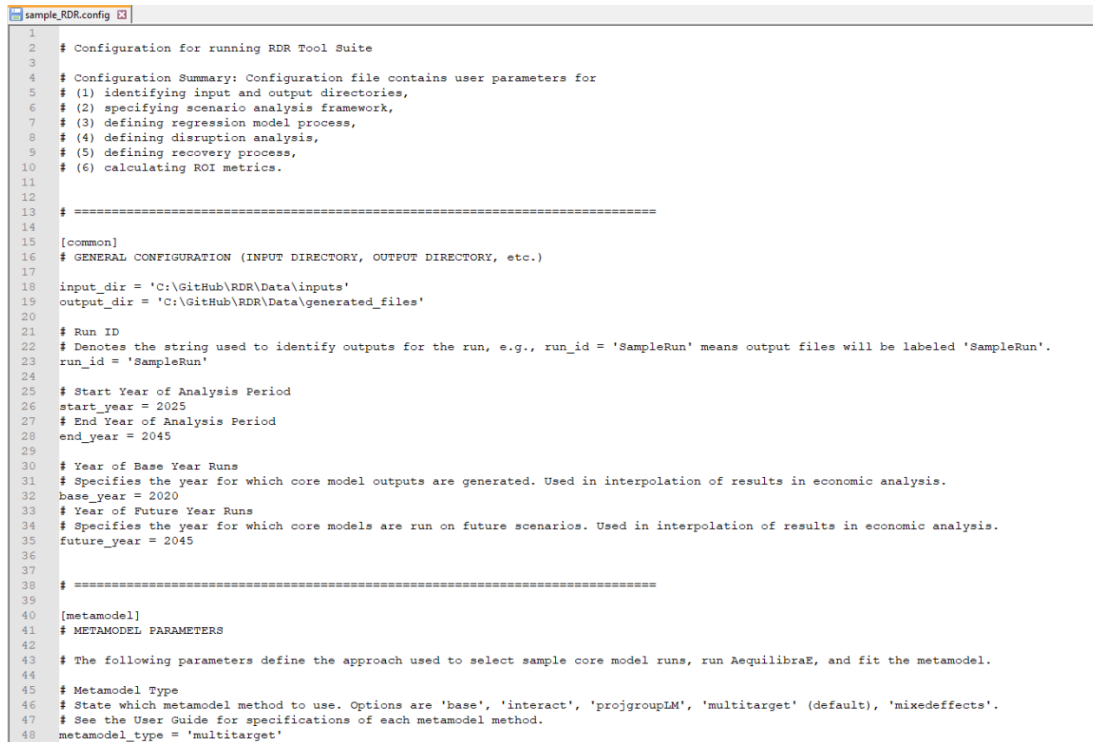
A sample version of the configuration file named ‘sample_RDR.config’ (seen in Figure 6-3 and located at https://github.com/VolpeUSDOT/RDR-Public/blob/master/config/sample_RDR.config) is provided in the “config” subfolder of the main RDR directory. The sample configuration file provides definitions of the parameters, values each parameter can take, and specifications of where in the RDR Tool Suite each parameter is used. Default values for most parameters are provided, which the user can replace if they have case-specific information for their analysis. In addition, several parameters are only required for specific settings (e.g., the ‘link_availability_csv’ parameter is only required if the ‘link_availability_approach’ parameter is set to ‘Manual’); these are identified within the sample configuration file. The user should create a copy of the sample file for their own run and modify the contents as needed.

Note that if the user runs the RDR User Interface, the configuration file is saved as a JSON file instead of a text file; the save file contains the same configuration information, along with scenario definition inputs found in the model parameters file. See Section 8 for more details.

The configuration file is split into five sections:

1. [common] – Identifies ‘input_dir’, the directory where all input files except the configuration file are located, ‘output_dir’, the directory where output files of the tool suite are created, ‘run_id’, a text string identifier for the RDR run, and years for the analysis period and core model runs.
2. [metamodel] – Specifies parameter values for how the AequilibraE core model is run, which output metrics are recorded, and how AequilibraE runs are sampled and used to fit the regression model in the RDR Metamodel; see Section 6.1.4 for more details.
3. [disruption] – Specifies parameter values for disruption analysis related to hazard events and resilience investments. Primarily used by the RDR Exposure Analysis Tool and the RDR Metamodel; see Sections 5 and 6.1.4 for more details.

4. [recovery] – Specifies parameter values for the hazard recovery and damage repair processes. Primarily used by the RDR Metamodel and the RDR ROI Analysis Tool; see Sections 6.1.4 and 6.1.5 for more details.
5. [analysis] – Specifies parameter values for model analysis, reporting, and benefit-cost analysis. Primarily used by the RDR ROI Analysis Tool; see Section 6.1.7 for more details.



```

1  # Configuration for running RDR Tool Suite
2
3  # Configuration Summary: Configuration file contains user parameters for
4  # (1) identifying input and output directories,
5  # (2) specifying scenario analysis framework,
6  # (3) defining regression model process,
7  # (4) defining disruption analysis,
8  # (5) defining recovery process,
9  # (6) calculating ROI metrics.
10
11
12
13  # =====
14
15  [common]
16  # GENERAL CONFIGURATION (INPUT DIRECTORY, OUTPUT DIRECTORY, etc.)
17
18  input_dir = 'C:\GitHub\RDR\Data\inputs'
19  output_dir = 'C:\GitHub\RDR\Data\generated_files'
20
21  # Run ID
22  # Denotes the string used to identify outputs for the run, e.g., run_id = 'SampleRun' means output files will be labeled 'SampleRun'.
23  run_id = 'SampleRun'
24
25  # Start Year of Analysis Period
26  start_year = 2025
27  # End Year of Analysis Period
28  end_year = 2045
29
30  # Year of Base Year Runs
31  # Specifies the year for which core model outputs are generated. Used in interpolation of results in economic analysis.
32  base_year = 2020
33  # Year of Future Year Runs
34  # Specifies the year for which core models are run on future scenarios. Used in interpolation of results in economic analysis.
35  future_year = 2045
36
37
38  # =====
39
40  [metamodel]
41  # METAMODEL PARAMETERS
42
43  # The following parameters define the approach used to select sample core model runs, run AequilibraE, and fit the metamodel.
44
45  # Metamodel Type
46  # State which metamodel method to use. Options are 'base', 'interact', 'projgroupLM', 'multitarget' (default), 'mixedeffects'.
47  # See the User Guide for specifications of each metamodel method.
48  metamodel_type = 'multitarget'

```

Figure 6-3: Sample Configuration File

6.1.1.2 Model Parameters File

The model parameters file is the key user input file for defining the analysis framework. It specifies the complete scenario space for the RDR Metamodel and defines the list of hazard events and recovery stages. The model parameters file is named ‘Model_Parameters.xlsx’ and is located in the input data folder specified by ‘input_dir’ in the configuration file. Note that if using the RDR User Interface, this file is not created and instead the data are stored in the JSON configuration file.

The model parameters file contains three sheets, which jointly define the analysis framework. The “UncertaintyParameters” tab (Figure 6-4) lists all possible values for scenarios dimensions; the “ProjectGroups” tab (Figure 6-5) supplies a mapping of project groups to resilience projects; the “Hazards” tab (Figure 6-6) defines all hazard events.

The “UncertaintyParameters” tab has five required columns:

1. ‘Hazard Events’ are text strings describing the possible hazard events and match the “Hazards” tab;
2. ‘Recovery Stages’ are non-negative integers starting from 0 (indicating initial hazard event severity) that enumerate the potential hazard recovery stages. The stages represent levels of receding exposure a hazard event may pass through during the hazard exposure period from

initial hazard severity to end of the hazard event (e.g., larger integers represent further recession);

3. 'Economic Scenarios' are text strings naming the possible future economic scenarios and correspond to the filenames for demand files and network attribute files;
4. 'Trip Loss Elasticities' are numeric values (less than or equal to zero) quantifying the change in trip demand due to increased travel time used by the core model;
5. 'Project Groups' are text strings naming the groupings of resilience projects and match the "ProjectGroups" tab and correspond to the filenames for network attribute files.

Note that data in this tab are entered column by column; depending on how many values each uncertainty parameter can take, the number of entries per column may vary. For example, in Figure 6-4, there are two recovery stages but only one economic scenario, so the number of rows in each of those columns differs. The uncertainty scenarios are built exclusively from the uncertainty parameter dimensions listed here, and the tool will not recognize additional parameters. A user may enter any number of parameter values for each uncertainty parameter; users can also choose to ignore an uncertainty by entering exactly one value. For example, a user may only have one economic scenario, reflecting a single best guess at future trip demand tables.

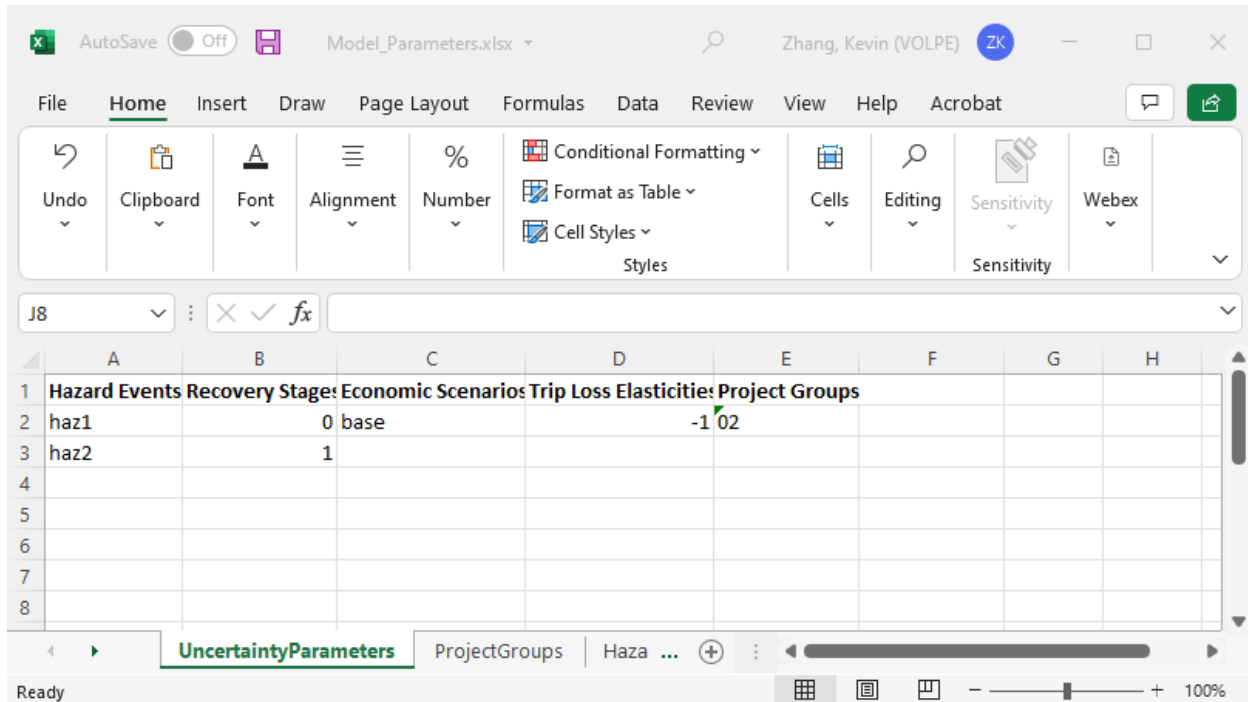


Figure 6-4: UncertaintyParameters tab of model parameters file for Quick Start 1

The "ProjectGroups" tab has two required fields:

1. 'Project Groups' are text strings naming the groupings of resilience projects and match the "UncertaintyParameters" tab and correspond to the filenames for network attribute files;
2. 'Resiliency Projects' are text strings identifying resilience projects and match the 'project_table.csv' and 'project_info.csv' input files.

In mapping the relationship between project groups and resilience projects in the "ProjectGroups" tab, the user does not need to include the baseline scenario of no resilience investment for each project

group; the RDR Tool Suite creates that baseline automatically for the ROI analysis of each resilience investment.

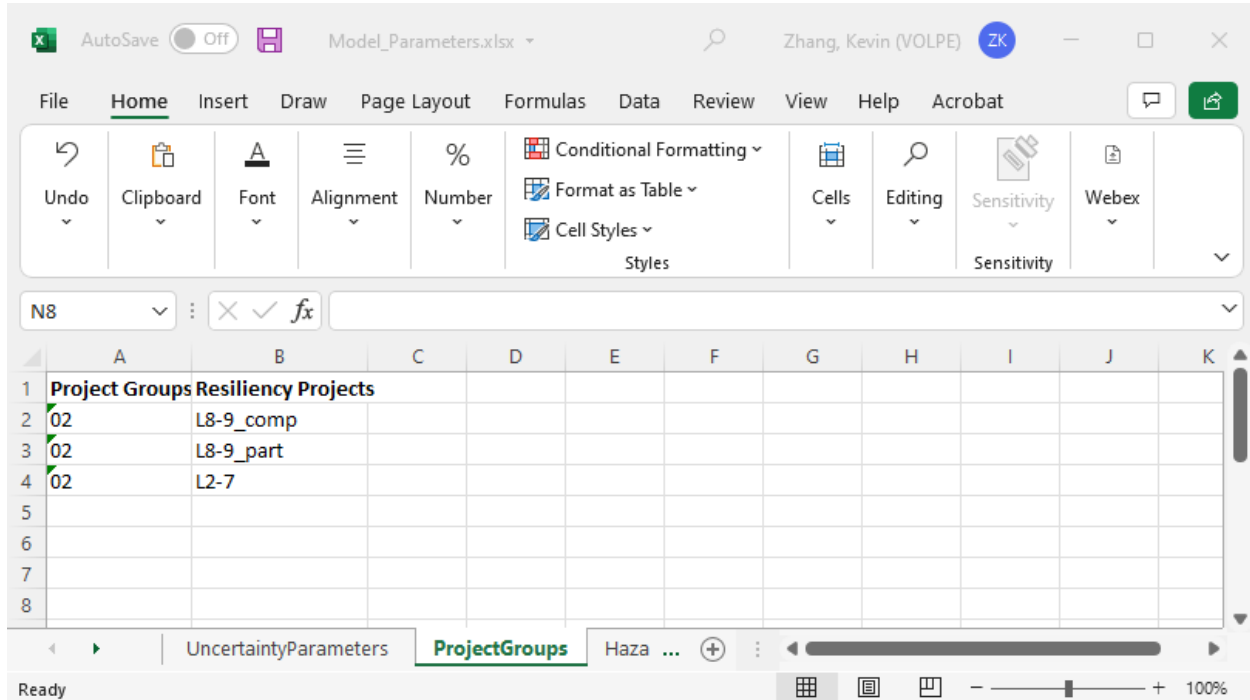


Figure 6-5: ProjectGroups tab of model parameters file for Quick Start 1

The “Hazards” tab has five required fields:

1. ‘Hazard Event’ are text strings describing possible hazard events and match the “UncertaintyParameters” tab;
2. ‘Filename’ are text strings connecting hazard events to exposure analysis files and correspond to the filenames of exposure analysis files (without the “.csv” file type extension);
3. ‘HazardDim1’ are text strings describing the short-term dimension of the hazard event used for filtering results (e.g., ‘100’ for 100-year storm surge for flooding event);
4. ‘HazardDim2’ are text strings describing the long-term dimension of the hazard event used for filtering results (e.g., ‘3’ for 3-foot sea-level rise for flooding event);
5. ‘Event Probability in Start Year’ are numeric annual probabilities of the hazard event occurring. Note that the hazard event probabilities are specified for the start year of the analysis period, not the base year. For scenarios running Benefit-Cost Analysis under uncertainty (BCA-U), all probabilities should be set to 1.

The ‘Hazard Event’ and ‘Filename’ fields are used in the ‘recov_init’ module, while the ‘recov_calc’ module uses the ‘Hazard Event’, ‘HazardDim1’, ‘HazardDim2’, and ‘Event Probability in Start Year’ fields.

	A	B	C	D	E	F	G	H	I
1	Hazard Event	Filename	HazardDim1	HazardDim2	Description	Event Probability in Start Year			
2	haz1	haz1	10	0	river flood, 10-year flood with no river level rise	0.1			
3	haz2	haz2	1	0	river flood, 1-year flood with no river level rise	0.9			
4									
5									
6									
7									
8									
9									
10									

Figure 6-6: Hazards tab of model parameters file for Quick Start 1

6.1.1.3 User Inputs File

The UserInputs.xlsx Excel workbook is used to generate the benefit-cost analysis and corresponding visualizations in the Tableau dashboard and is located in the input data folder specified by 'input_dir' in the configuration file. This user inputs file is the primary input for the RDR ROI Analysis Tool. It specifies the set of scenarios for which the ROI analysis is run. The user inputs file should specify a subset of the scenario space as defined in the model parameters file for the RDRM, and values within each column of the user inputs file must be drawn from the values contained in the model parameters file. The structure of the input file is similar to the "UncertaintyParameters" tab of the Model_Parameters.xlsx input file (shown in Figure 6-4), but there is an additional field for the Event Frequency Factor (Figure 6-7 below). As with the "UncertaintyParameters" tab of the model parameters file, note that data in this input file are entered column by column; depending on how many values each uncertainty parameter can take, the number of entries per column may vary. Note that if using the RDR User Interface, this file is not created and instead the data are stored in the JSON configuration file.

The "UserInputs" tab (Figure 6-7) has five required fields:

1. 'Hazard Events' are text strings and should be a subset of the list in the model parameters file;
2. 'Economic Scenarios' are text strings and should be a subset of the list in the model parameters file;
3. 'Trip Loss Elasticities' are numeric and should be a subset of the list in the model parameters file;
4. 'Resiliency Projects' are text strings and should be a subset of the list in the model parameters file;
5. 'Event Frequency Factors' are numeric and define how the probability of each hazard event will increase or decrease year-on-year during the analysis period. A value of 1 means no change across years in hazard event probability values; a value above 1 means an increase in frequency; a value below 1 means a decrease in frequency.

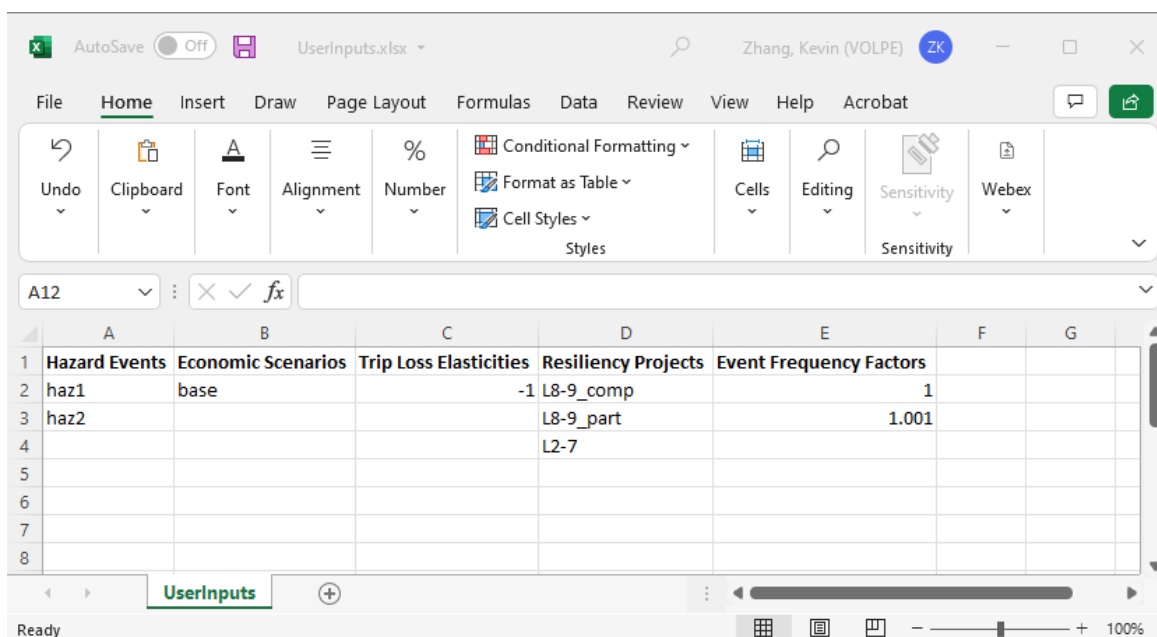


Figure 6-7: User inputs file for Quick Start 1

6.1.1.4 RDR Input Validation Helper Tool

The RDR Tool Suite comes with a Python-based tool to help users validate their input files before a run of the RDR Metamodel and ROI Analysis Tool. The helper script checks that the user has included all required input files, each with all of the required fields, as specified by the user in the model parameters input file (which defines the scenario space). The helper script also checks that particular data columns are able to be converted as needed to numeric values. All input validation errors are consolidated into one log file generated in the output folder.

The input validation helper tool leverages the same configuration file used for a full RDR run. The configuration parameters determine the input files that are required and their required structure. The user should run the input validation helper tool as follows:

1. Navigate to the “helper_tools” subfolder of the RDR directory (e.g., “C:\GitHub\RDR\helper_tools”).
2. Navigate to the “input_validation” subfolder and open the “run_input_validation.bat” batch file in a text editor (e.g., right-click on the file and select Edit).
3. Confirm that lines 13 through 15 point to the following three files:
 - a. The location of the user’s Anaconda scripts folder,
 - b. The Python executable associated with the user’s RDR Python environment (RDREnv),
 - c. The location of the “rdr_input_validation.py” script on the user’s machine.
4. Edit line 17 of the batch file to specify the full file path of the configuration file corresponding with the RDR scenario the user plans to run.
5. Open the Windows Command Prompt interface or Windows PowerShell (e.g., go to the Windows Start Menu, type Command Prompt in the search bar, click on the Command Prompt icon).
6. Drag the batch file into the terminal window, click into the terminal window, and press enter to run.

7. The output of the input validation helper tool is a log file listing all of the errors identified with the user's input files. The log file is located in the "logs" subfolder of the output directory specified by the user in the configuration file.

Note that the input validation helper tool is not a comprehensive check of all possible user input errors. In particular, the script does not check for errors in the batch file, the configuration file, or optional input files. For more information on input file specifications, refer to [Appendix A: RDR Tool Suite Input Files](#) and the RDR Run Checklist.

6.1.2 Configuring Exposure Scenarios

6.1.2.1 *Creating Exposure Analysis Files from the Network Map and Exposure Data*

If the user needs to create a new exposure map, there are several ways to accomplish this. First, the user could create the map manually if the number of assets exposed is small or can be assessed based on expert knowledge. The user can create a simple table which links the asset ID from the TDM to an exposure level (e.g., flood depth). Again, the asset ID must match the relevant asset ID in the TDM to appropriately link these data. This manual approach, or manual verification of outputs at least, is recommended for application of exposure to transit assets (particularly dedicated guideway assets), as the RDR Exposure Analysis Tool does not provide specific relationships for hazard impact to transit links.

Second, the user can use a GIS tool to overlay hazard data on top of the transportation asset data to determine where the two datasets intersect or overlap and to associate exposure levels with each asset. To aid agencies in executing a simple GIS-based exposure analysis, the RDR Exposure Analysis Tool provides a simplified analysis script for use in ArcGIS or ArcGIS Pro (see Section 5). While the script is not required in order to run the core models or RDRM, it can be useful for creating, analyzing, and visualizing specific exposure and disruption scenarios without running the full tool.

Data from the RDR Exposure Analysis Tool or an existing GIS-based data capturing hazard exposure levels can easily be visualized using GIS software such as QGIS or ArcGIS Pro. Since the exact structure of exposure data may vary greatly from scenario to scenario, and access to specific GIS software varies from agency to agency, this part of the process is not automated by the Exposure Analysis Tool. However, with the GIS output from the RDR Exposure Analysis Tool (or existing agency data), a user with basic GIS knowledge should be able to easily create simple maps, such as the above in Figure 5-3, that visualize a hazard scenario with the resulting disruption output.

6.1.2.2 *Exposure Analysis Files*

The exposure analysis CSV files (name format '{Filename}.csv') provide link-level hazard exposure data for each hazard event. The "Filename" name attribute is designated by the user in the "Hazards" tab of the model parameters file (Section 6.1.1.2). The exposure analysis files are located in the "Hazards" subfolder of the input data folder and are used as inputs to the AequilibraE core model and the damage and repair components of the tool suite. Each row in the file denotes a link in the transportation network. The required fields are:

1. 'link_id' – ID of network link.
2. 'from_node_id' – ID of predecessor node of network link.
3. 'to_node_id' – ID of successor node of network link.
4. 'Value' (or corresponding text string specified by the 'exposure_field' parameter in the configuration file) – Numeric exposure level of hazard event on link. Units of the exposure level

must be specified by the 'exposure_unit' parameter in the configuration file if using the default flood exposure function.

All fields are used by the 'aeq_run' and 'recov_init' modules. The exposure analysis CSV files can be created by the RDR Exposure Analysis Tool described in Section 5 (recommended) or created manually by the user. If created manually, only network links that are impacted by the hazard event need to be included with the required fields listed above. A sample exposure analysis file from Quick Start 1 is shown in Figure 6-8.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	link_id	from_node_id	to_node_id	Value	evacuation_route	link_availability	comments						
2	1	1	2	0	0	1	river flood						
3	2	1	3	0	0	1	river flood						
4	3	2	1	0	0	1	river flood						
5	4	2	6	0	0	1	river flood						
6	5	3	1	0	0	1	river flood						
7	6	3	4	0	0	1	river flood						
8	7	3	12	0	0	1	river flood						
9	8	4	3	0	0	1	river flood						
10	9	4	5	0	0	1	river flood						
11	10	4	11	0	0	1	river flood						
12	11	5	4	0	0	1	river flood						
13	12	5	6	0	0	1	river flood						
14	13	5	9	0	0	1	river flood						

Figure 6-8: Sample exposure analysis file for Quick Start 1

6.1.3 Configuring the Core Model Network

6.1.3.1 Configuring the Routable Network

In order to run the core models and populate the RDR Metamodel, the user needs a routable network file associated with each scenario of interest (i.e., future economic scenario, project group). For evaluating travel demand, a routable network of nodes and links is necessary to route trips over the transportation network to calculate VHT and VMT metrics. Section 3.1 describes the details of setting up a network.

6.1.3.2 Network Files

In the RDR Tool Suite, link files will often be different for different scenarios (e.g., additional links may be added or link capacities may change based on planned construction in response to future economic scenarios), while the same nodes file is used for all scenarios. These files are located in the "Networks" subfolder of the input data folder and are used as inputs to the AequilibrE core model and the RDR ROI Analysis Tool.

The links file (name format ' {econ}{projgroup}.csv ') lists the full set of network links for each combination of future economic scenario and project group (each replacing the respective bracketed

text in the filename) found in the model parameters file (e.g., two future economic scenario options and three project groups lead to six links files).

6.1.3.3 *Configuring Transportation Disruption Submodule Based on Hazard Exposure*

The RDRM code contains a sample configuration file (`'sample_RDR.config'`) which allows the user—among other configuration options—to set up the disruption analysis. The user can set the parameters called **Link Availability Approach** (`link_availability_approach`) and **Exposure Field** (`exposure_field`).

Additional parameters depending on the link availability approach selected (**'Manual,'** **'Facility_Type_Manual,'** **'Default_Flood_Exposure,'** and **'Beta_Distribution_Function'**) are located in the configuration file but only need to be defined if the relevant link availability approach is selected. For more information on these parameters and how to set them, see Section 5.2.

In addition, two other parameters in the configuration file specify how disruption is applied to centroid connector links and resilience project links, respectively:

- `'highest_zone_number'` – Node ID designating the highest node ID in the network for a centroid node. Links connecting one or more centroid nodes are not impacted by hazard events.
- `'resil_mitigation_approach'` – Method used to calculate mitigation for resilience projects. If `'manual'` is used, then the user must provide link-level exposure mitigation data for each project link in the `'project_table.csv'` input file (see Section 6.1.6.2).

Links not found in the exposure analysis input files are assigned a link availability of 99.9% by default.

6.1.3.4 *Output and Validation*

The output of the exposure and disruption module is a CSV file containing the entries for each transportation network asset and its availability under each unique hazard scenario. This output file is an input for the core models. This file enables the analysis of the different hazard scenarios with resilience investments (mitigations).

The user should validate the results before proceeding to the next section regarding damage disruption analyses. For example, the user should review the outputs to check for misidentification to ensure disruption is only occurring on the appropriate network links given the hazard severity, e.g., check to see whether bridges that cannot be inundated are disrupted.

6.1.4 *Configuring the RDR Metamodel Parameterization Submodule*

This section describes how to use the core models to generate initial data for input to the RDRM. TDM analysis is designed to produce a steady state that describes the travel patterns for an average weekday under a given set of specified conditions. These steady states are produced for a base year and an out year (e.g., 50 years into the future). The choice of base year depends on what data are available to the user and the agency decision making goals. The out year is a future year for which the user has reasonable estimates of trip demand and is within the period of analysis.

6.1.4.1 *RDR Metamodel Components*

The analytical steps executed by the RDRM are as follows:²⁶

- Latin Hypercube Sampling selects which scenarios will be run in the core models.

²⁶ This process is reordered from the technical document to reflect the order of the mechanical steps rather than the conceptual steps of the process.

- Run core model for selected scenarios and match core model outputs with their scenario factors.
 - Scenario factors can include hazard attributes (e.g., sea level rise, storm surge), assumed elasticity of trip demand, resilience project, economic scenario, or other user-defined factors.
- Metamodel regression
 - Parameterize model with core model and/or supplemental model runs.
 - Response variables of the regression are total trips, PMT, and PHT aggregated across the transportation network. If the user chooses to calculate metrics by mode, these are also included as response variables of the regression.
 - Run regression to produce the estimated performance metrics for all combinations of scenario factors not implemented in the core model or supplemental model runs.
 - Validate regression model outputs and reconsider as needed the specification of the regression model(s) and/or number of core model runs needed.
- Scenario expansion and annualization
 - Extend daily hazard snapshots to produce logical space of scenario hazard recovery events using user-defined recovery parameters.
 - Interpolate regression model outputs to produce estimated performance metrics for all hazard events.
- Pass results to RDR ROI Analysis Tool

6.1.4.2 Configuring the Latin Hypercube Sampling

The Latin hypercube sampling (LHS) module is used to select the set of core model runs used to fit the regression model in the RDRM. The module chooses uncertainty scenarios randomly and runs a series of coverage tests to make sure the selected set provides adequate data to fully fit a regression model for the entire scenario space.

The user specifies the total number of samples selected to fit the regression using the 'lhs_sample_target' parameter in the configuration file. If the parameter value is too small and the sample is unable to provide coverage of the full scenario space, then the user will encounter an error during the run and be prompted to enter a larger number. The larger the LHS sample target, the longer the runtime, as each core model run can take several hours depending on the size of the network. As a rule of thumb, sampling at least 20% of the scenario space using the core models is recommended and can be further adjusted based on the results of the regression if needed (see Section 6.1.5.1 below). While there is no fixed number for scenarios sampled that should be used to parameterize the metamodel, for smaller numbers the resulting metamodel may have high uncertainty around the estimated number of trips and hours of travel and thus lead to highly variable results in the ROI analysis. In addition to checking model fit in the regression module, it is recommended that the user conduct some sensitivity analysis of resulting metamodel outputs around the number of sampled scenarios.

6.1.4.3 Configuring the AequilibraE Core Model

The RDR Tool Suite includes a link to AequilibraE (version 1.0.0), an externally provided open-source shortest path and routing model that may be used to compute network flows and summary statistics for baseline and disrupted networks. In RDR, AequilibraE is used to quickly determine daily link flows and skims (times and distances between each origin and destination) for both base and disrupted networks.

AequilibraE requires the following inputs:

- An origin-destination trip table file stored as a square matrix in open matrix (OMX) format (see below for details).
- A routable network consisting of node and link tables (see Section 6.1.3), which are stored in an SQLite database.

Several parameters in the configuration file under the [metamodel] section are used to configure the core model. The 'aeq_run_type' parameter specifies whether a shortest path or routing methodology should be used by AequilibraE. The 'run_minieq' parameter specifies if the routing code should be run multiple times in order for vehicles to adjust route choice and settle more into an equilibrium state. A configurable parameter 'allow_centroid_flows' indicates whether routing is permitted via the centroid connectors within AequilibraE.

In addition, there are several input files required to run the AequilibraE core model. They include demand files, the (optional) true shapes file, a template SQLite database, and a set of (optional) AequilibraE look-up tables. Refer to Section 3 for details on setting up the network and trip tables for the core model. Technical details of how the input files are used by the core model can be found in the main technical documentation.

The demand OMX files (name format '{econ}_demand_summed.omx') provide origin-destination trip table data for each future economic scenario specified in the model parameters file for AequilibraE in the 'aeq_run' module. These files are located in the "AEMaster/matrices" subfolder of the input data folder. These demand files must be provided by the user in the open matrix (OMX) data file format. The demand matrix (or matrices) in the OMX file should be a square matrix. The RDR Tool Suite provides a set of Python helper tools and documentation for converting trip table data into the OMX format (see [Appendix H: Format Demand Helper Tools](#) for more details). A single OMX file may contain several trip tables. In RDR, the OMX demand file should always contain a table that is named 'matrix'. These are trips for households that have access to an automobile. If RDR is to be used for transit and equity analysis and zero-car household data are available, there is functionality to include a second trip table, named 'nocar', that contains trips for zero-car households.

To convert network and exposure data into inputs for the AequilibraE core model, a set of optional look-up tables can be used as input files. These files should be placed in the "LookupTables" subfolder of the input data folder. Currently there is only one optional look-up table used by RDR:

- 'link_types_table.csv', discussed in Section 3.1.1.1, specifies custom volume-delay function parameters for each link type, which are provided as input to AequilibraE. The input file has required fields 'facility_type', 'alpha', and 'beta', all used by the 'aeq_run' module. The 'facility_type' field should match the 'facility_type' field in the network attribute files.

Once the core model input files have been prepared, the user should run the AequilibraE core model with a scenario setting comparable to an existing TDM run in order to validate the AequilibraE model and confirm outputs are reasonable. For more information on what to consider when running AequilibraE for the first time, refer to Section 9.3.1.

6.1.4.4 Trip Table Files

The RDR implementation of AequilibraE expects a single trip table file, in the shape of a square matrix in OMX format,²⁷ representing person trips for a generic day of travel. This file may optionally contain two trip tables, one for 1+ car households (named 'matrix') and one for zero car households (named 'nocar'). The documentation for the existing MPO TDM will have information on the parameters needed to synthesize such a trip table.

The reason for having two trip tables is that 1+ car households and 0-car households experience different costs and travel times for using the road network (see Table 3-2, earlier in this document). Zero-car households have the added wait time and out-of-pocket cost of taxi/TNC wait time and taxi/TNC fare. The effect is to push the 0-car households to the transit links (whose costs and travel times are generally the same for both types of households).

6.1.5 Configuring the RDR Scenario Expansion Submodule

6.1.5.1 Configuring the Regression

A number of different possible regression models can be applied by the user. The simplest is a linear regression, where each input is a predictor and each target variable has a different regression model. A more complex model available is a multitarget Gaussian process regression, where each of the target variables (overall trips, hours, and miles as well as mode-specific metrics if specified by user) are fit simultaneously. This multitarget approach has shown the best performance in model fit across regression models in testing and is the default model set by RDR. The full set of models that can be run by the user using the RDRM code (specifically, in `rdr_Metamodel_Regression.Rmd`) is:

- 'multitarget' (default) – Implements a simplified version of TMIP-EMAT's multitarget Gaussian process regression.²⁸
- 'base' - Constructs separate linear regression models for each response variable.
- 'interact' – Extends the base model to include interaction terms (e.g., the statistical interaction between hazard event and recovery stage, and between project group and resilience project).
- 'projgroupLM' – Extends the base model with separate regression sub-models for each set of resilience projects within each project group.
- 'mixedeffects' – Extends the base model with a mixed-effects approach, using random effects for each resilience project within a project group.

The user can change the regression model they use in the RDRM by adjusting the 'metamodel_type' parameter in the configuration file. If the user chooses to modify the regression model used in the run after an initial analysis, additional sampling of the scenario space using AequilibraE may be required to provide adequate sample coverage for the new regression model (i.e., it may require a higher value for the `lhs_sample_target` parameter).

As a rule of thumb, if the coefficients of determination (r^2) of the resulting regression models for trips, travel time, or travel distance are below 0.5, the user should consider sampling more scenarios with core model runs (again increasing the `lhs_sample_target` parameter). On the other hand, a very high r^2 (>0.9) may indicate overfitting of the model, particularly if the number of scenario factors used as

²⁷ See [Open Matrix Format | TF Resource](#) and [GitHub - osPlanning/omx: Open Matrix \(OMX\)](#). RDR also provides a helper tool for converting comma-separated origin-destination-trip lists to an OMX file.

²⁸ TMIP-EMAT. [Online]. Meta-Model Regression. Accessed 26 July 2022 from <https://tmip-emat.github.io/source/emat.metamodels/regression.html>.

independent variables in the regression is large. The RDR Technical Documentation contains more information and guidance on the pros and cons of each regression model option.

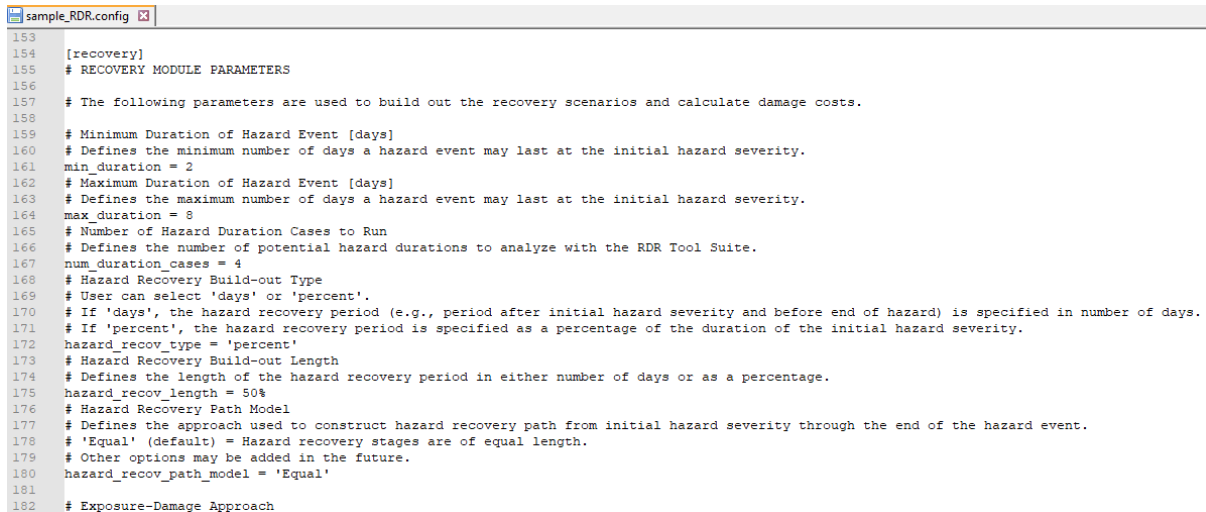
The multitarget regression is the default regression model, but if insufficient data are available to fit the model, the base model approach will be used. A log message will appear to alert users that the regression model has changed to the base approach in this case.

6.1.5.2 Configuring Metamodel Exposure Recovery

The recovery process requires the following inputs:

- The configuration file containing parameters specifying how exposure recovery paths and repair recovery paths are built;
- “Model_Parameters.xlsx” and “UserInputs.xlsx” input files defining the parameters and their values that make up the scenarios;
- Exposure analysis files for each hazard scenario to be analyzed;
- The “project_table.csv” look-up table mapping resilience investments to their associated network links and network files with link attributes like distance and functional class;
- (Optional) Look-up tables for exposure-damage, repair cost, and repair time that the user provides. Default look-up tables are also provided.

The configuration file contains parameters used by the RDRM to build out complete hazard recovery paths (series of network states from maximum disruption to full recovery) from the hazard scenarios described in this section. The recovery section of the RDR configuration file (Figure 6-9) is used to specify the hazard duration parameters which include: minimum duration of hazard event, maximum duration of hazard event, number of hazard duration cases to run, hazard recovery build-out type and length, and hazard recovery path model. Default values for these parameters are provided. Using these parameters, the RDRM builds out hazard scenarios with varying hazard event duration and hazard recovery paths.



```

153
154 [recovery]
155 # RECOVERY MODULE PARAMETERS
156
157 # The following parameters are used to build out the recovery scenarios and calculate damage costs.
158
159 # Minimum Duration of Hazard Event [days]
160 # Defines the minimum number of days a hazard event may last at the initial hazard severity.
161 min_duration = 2
162 # Maximum Duration of Hazard Event [days]
163 # Defines the maximum number of days a hazard event may last at the initial hazard severity.
164 max_duration = 8
165 # Number of Hazard Duration Cases to Run
166 # Defines the number of potential hazard durations to analyze with the RDR Tool Suite.
167 num_duration_cases = 4
168 # Hazard Recovery Build-out Type
169 # User can select 'days' or 'percent'.
170 # If 'days', the hazard recovery period (e.g., period after initial hazard severity and before end of hazard) is specified in number of days.
171 # If 'percent', the hazard recovery period is specified as a percentage of the duration of the initial hazard severity.
172 hazard_recov_type = 'percent'
173 # Hazard Recovery Build-out Length
174 # Defines the length of the hazard recovery period in either number of days or as a percentage.
175 hazard_recov_length = 50%
176 # Hazard Recovery Path Model
177 # Defines the approach used to construct hazard recovery path from initial hazard severity through the end of the hazard event.
178 # 'Equal' (default) = Hazard recovery stages are of equal length.
179 # Other options may be added in the future.
180 hazard_recov_path_model = 'Equal'
181
182 # Exposure-Damage Approach

```

Figure 6-9: Configuration file showing parameters used to define hazard event duration uncertainties

The possible exposure recovery paths for a given scenario are constructed from a set of parameters that the user can define in the configuration file or leave to the default values. The ‘min_duration’ and ‘max_duration’ parameters, specified in days, define the shortest and longest time the hazard event lasts at the initial exposure severity. The ‘num_duration_cases’ parameter specifies the number of

exposure recovery paths to generate for each scenario. The ‘hazard_recov_type’, ‘hazard_recov_length’, and ‘hazard_recov_path_model’ parameters define the subsequent network states of the exposure recovery path. The recession length, which is specified by ‘hazard_recov_length’ in either days or as a percentage of the initial hazard duration (depending on the value of ‘hazard_recov_type’), defines the duration of the hazard event after the initial exposure state. The recession path model parameter specifies the relative durations of the subsequent network states. In particular, the ‘Equal’ recession path model option currently implemented specifies that the exposure recovery path network states have equal duration, after the initial exposure state until the hazard has completely receded.

As an example, consider the following default (adjustable) parameters settings in the configuration file:

- min_duration = 2,
- max_duration = 8,
- num_duration_cases = 4,
- hazard_recov_type = ‘percent’,
- hazard_recov_length = 50%,
- hazard_recov_path_model = ‘Equal’.

The RDRM Recovery process uses these parameters to construct the possible exposure recovery paths for a given scenario and hazard event. The duration of the hazard at the initial exposure severity takes 4 possible values (‘num_duration_cases’), ranging from 2 days (‘min_duration’) at the shortest to 8 days (‘max_duration’) at the longest. From this, the RDRM creates four possible exposure recovery paths in the scenario space with initial exposure severity durations of 2 days, 4 days, 6 days, and 8 days. The recovery module then defines the subsequent network states of the exposure recovery path from the initial exposure severity duration. The recession lengths are defined to be 50% of the initial hazard durations; for the 8-day initial hazard duration, this means a recession length of 4 days, etc. To assign a recovery network state to each of the 4 days of the hazard recession period, the RDRM uses the Hazards table in “Model_Parameters.xlsx” input file to determine equally spaced network states starting at the initial exposure state to the final no-exposure state (as per the ‘Equal’ recession path model). Within the RDRM, the progression of network states is notated using a Hazard Level field constructed by the RDRM from the Hazards table—every possible exposure recovery path is written as a text string of non-increasing Hazard Levels, with one for each day of the hazard duration. For example, the exposure recovery path ‘4,4,4,4,3,1’ defines a hazard that lasts for six days before reaching the no-exposure state. The initial exposure severity of Hazard Level 4 lasts for four days, and the recession period is two days with the first day at Hazard Level 3 and the second day at Hazard Level 1. After the sixth day, the hazard has completely receded.

6.1.5.3 Configuring Metamodel Damage and Repair Recovery

There are four main input files that the user can provide to calculate damage and repair metrics. Alternate methods or default input files are also provided for these functions; sample files demonstrate how the user can create their own inputs and are located in the “config” subfolder of the RDR directory. Technical details of how the RDR Metamodel uses each of these input files can be found in the RDR Tool Suite Technical Document.

The RDRM provides the user with a few options for calculating asset damage. The method used is selected by the user in the configuration file with the parameter ‘depth_damage_approach’:

- The 'Binary' approach provides the coarsest damage calculation. Any level of exposure above 0 feet is considered to incur full damage to the network segment in the road network. The link is considered fully damaged and incurs the full damage cost and time to repair. Network segments with no exposure incur no damage.
- The 'Default_Damage_Table' uses a depth-damage table adapted from London data (see Table 6-1) for road links.²⁹ Note that the table is specific to water hazards on roadways and provides a mapping of exposure depth to asset damage percentage. The version of the table used by the RDR Tool Suite converts exposure depth to intervals to provide full coverage of all possible exposure depth values. The adapted default table is also provided below (see Table 6-2), along with asset damage percentages for exposure intervals for bridge assets, also sourced from London data, and for transit assets, taken from an engineering study on saltwater flood exposure for rail rapid transit.

Table 6-1: Default depth-damage relationship (from Simonovic et al. 2011)

Asphalt Concrete		
Depth (m)*	Damage (%)	Explanation
-0.5	0	Slight damage may occur to subgrade and substructure due to seepage
0	0.05	Presume there is no damage to the surface layer until water level is above paved elevation
1	0.2	Including modest damage due to water on asphalt surface
2	0.5	Higher degree due to floodwaters inundating paved surface
5	1	Upper boundary of road damage

Table 6-2: Default depth-damage look-up table used by the RDR Tool Suite for flood-based hazards

Asset Type	min_exposure	max_exposure	Damage (%)
Highway	-999999	-0.82	0
Highway	-0.82	1.64	0.05
Highway	1.64	4.92	0.2
Highway	4.92	11.48	0.5
Highway	11.48	999999	1
Transit	-999999	0	0
Transit	0	0.5	0.17
Transit	0.5	3	0.48
Transit	3	7	0.63
Transit	7	15	0.78
Transit	15	999999	1
Bridge	-999999	-3.28	0
Bridge	-3.28	0	0.8
Bridge	0	999999	1

²⁹ Simonovic, S P; Burn, D; Sandink, D; Eum, H; Sredojevic, D; Peck, A; Bowering, E. 2011. The City of London: Vulnerability of Infrastructure to Climate Change. University of Western Ontario Department of Civil and Environmental Engineering. London, England.

- The 'Manual' approach requires the user to provide their own exposure-damage table. The structure of the table must match the structure of the default depth-damage table with fields named `min_exposure` and `max_exposure` providing exposure depth intervals that map to a particular asset damage percentage. The Asset Type field can be used to differentiate the exposure-damage function for different assets (e.g., roadway, bridge). The user should base this off of a template CSV ("`sample_manual_exposure-damage_table.csv`") that accompanies the code in the "config" subfolder of the main RDR directory.

The user-provided exposure-damage table CSV file calculates link damage percentage based on user-defined bins for each asset type. The required fields are:

1. 'Asset Type' – Text string categorizing the network link. These should match the 'Category' field in the resilience project input files.
2. 'min_exposure' – Numeric lower bound for exposure level for each bin. This is compared to the 'Value' field of the exposure analysis input files.
3. 'max_exposure' – Numeric upper bound for exposure level for each bin. This is compared to the 'Value' field of the exposure analysis input files.
4. 'Damage (%)' – Numeric damage percentage value assigned for each bin. This should be a number between 0 and 1.

All fields are used by the 'recov_init' module.

The user-defined repair cost table CSV file is used to calculate link repair cost based on user-defined asset and facility type categories. The user should base this off of a template CSV ("`sample_repair-cost_table.csv`") that accompanies the code in the "config" subfolder of the main RDR directory. The required fields are:

1. 'Asset Type' – Text string categorizing the network link. These should match the 'Category' field in the resilience project input files.
2. 'Facility Type' – Numeric value categorizing the facility type of the network link. These should match the 'facility_type' field in the network attribute files.
3. 'Damage Repair Cost' – Monetary value of repair cost for a network link of a particular asset type and facility type (specified in dollar units given by 'dollar_year' parameter in configuration file). Repair costs must be defined per lane-mile for all asset types except 'Bridge', which is defined per square foot. These look-up values are scaled by damage percentage to calculate link-level repair costs in the tool.
4. 'Total Repair Cost' – Monetary value of total repair cost (includes debris clean up, etc.) for a network link of a particular asset type and facility type (specified in dollar units given by 'dollar_year' parameter in configuration file). Repair costs must be defined per lane-mile for all asset types except 'Bridge', which is defined per square foot.

The default repair cost table also contains the field 'Network Type', which must be specified in the configuration file. If the user wishes to use the default repair cost table provided with the code, they should confirm the following:

1. 'Asset Type' – The 'Category' field in the resilience project input files should take only values: "Highway", "Bridge", or "Transit".

2. 'Facility Type' – The “facility_type” field in the network link input files should: (1) specify functional class and/or GTFS classification of the road, bridge, or transit link, (2) correspond to the default facility types used by RDR as listed in Table 3-3.

All fields are used by the 'recov_init' module.

The user-defined repair time table CSV file is used to calculate average repair time based on user-defined asset type categories. The user should base this off of a template CSV (“sample_repair-time_table.csv”) that accompanies the code in the “config” subfolder of the main RDR directory. The required fields are:

1. 'Asset Type' – Text string categorizing the network link. These should match the 'Category' field in the resilience project input files.
2. 'min_inclusive' – Numeric lower bound for repair category for each bin. This is compared to the 'facility_type' field of the network attribute files.
3. 'max_exclusive' – Numeric upper bound for repair category for each bin. This is compared to the 'facility_type' field of the network attribute files.
4. 'repair_time' – Numeric repair time for a network link of a particular asset type in units of days. Repair times are average look-up values and are scaled by damage percentage to calculate link-level repair times in the tool.

All fields are used by the 'recov_init' module. The 'min_inclusive' and 'max_exclusive' fields create bins for determining repair time. If the default repair time is used, repair category values are created from 'facility_type' field for road and transit links (see Table 3-3) and summed 'length' field across links (in feet) for bridge links. In addition, as with the repair cost table, the 'Category' field in the resilience project input files should only take values “Highway”, “Bridge”, or “Transit”. If a user-defined repair time look-up table is used, the 'facility_type' field is used for all links. The default repair time values were provided by Virginia DOT and sourced from a Hurricane Sandy transit study and may not be applicable to all agencies; the user should use agency-specific estimates whenever possible.

6.1.6 Configuring Resilience Projects

6.1.6.1 *Configuring Transportation Disruption Mitigation Based on Resilience Investment*

A user can represent the mitigation of a resilience investment in three distinct ways. First, a resilience investment that adds new network links can be modeled by adding a new link to the network. Second, a resilience investment can be modeled as a reduction in the exposure analyses (e.g., reducing flooding depth by a given amount). The modified exposure is then carried through the disruption and damage analyses. Third, agencies can directly adjust the extent of disruption and damage associated with a resilience investment for each given hazard (e.g., to assume complete or partial mitigation on the asset). The RDR Tool Suite provides agencies with a straightforward method for the exposure modification approach for incorporating resilience alternatives.

The current RDRM disruption analysis offers two approaches for modeling mitigation of a resilience investment: complete mitigation and link-level partial mitigation. The approach followed is set by the user in the configuration file using the 'resil_mitigation_approach' parameter. The first approach models resilience investments as a complete mitigation of the impact of a hazard event on link availability and damage costs. Investment in a resilience project leads to full link availability on the associated network links and zero asset damage, which translates to no repair cost or recovery time on the associated network links, for all hazard events analyzed.

The second approach of partial mitigation allows the user to specify an exposure mitigation value for each network link of the asset associated with the resilience investment. This reduction in exposure is applied across all hazard events, the calculated impact on disruption and damage of the mitigation is attributed to the resilience investment, and associated cost savings compared to the baseline of no mitigation are aggregated for the ROI analysis. This modeling functionality is provided by the 'Exposure Reduction' field in the project table input file described in Section 6.1.6.2. Modeling varying mitigation effects of a resilience investment by hazard event requires manual adjustment outside the RDRM and is not within the scope of the current disruption analysis.

6.1.6.2 Resilience Project Files

Two input files provide data on resilience investments: the project info file and the project table file.

The project information file (Figure 6-10) lists all resilience projects ('Project ID'), their costs ('Project Cost'), and their lifespans ('Project Lifespan'), and is used by the RDR ROI Analysis Tool for economic analysis. Each resilience project is also given a project name ('Project Name') and associated with a specific asset of the transportation network ('Asset'). The project information file is named "project_info.csv" and located in the "LookupTables" subfolder of the input data folder. All fields are used by the 'recov_calc' module. There are five required fields in the table:

1. 'Project ID' – ID of the resilience project. These should match the resilience projects identified in the model parameters file. Resilience projects are not allowed to be given the ID 'no' as this is reserved for the no-action baseline.
2. 'Project Name' – Text string describing the asset and resilience investment associated with the resilience project.
3. 'Asset' – Text string categorizing the asset associated with the resilience project.
4. 'Project Cost' – Numeric values of the resilience investment costs for each resilience project. The costs provided in the project info file should be for the project lifespan and should reflect the dollar year units specified by the 'dollar_year' parameter in the configuration file.
5. 'Project Lifespan' – Numeric values of the resilience investment lifespans (in years) for each resilience project.

An additional optional field for 'Annual Maintenance Cost' can be provided; this cost will be applied every year of the period of analysis. A similar additional optional field for 'Reployment Cost' can be provided and will be applied every project lifespan *after* the initial project construction in the start year. Both additional optional fields need to be toggled on to be used in the benefits calculations. This toggle can be found in the Recovery section of the configuration file, denoted by variable 'maintenance' and 'redployment'.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Project ID	Project Name	Asset	Project Cost								
2	L8-9_comp	L8-9 Complete Mitigation	Link8-9	\$750,000.00								
3	L8-9_part	L8-9 Partial Mitigation	Link8-9	\$250,000.00								
4	L2-7	L2-7 Complete Mitigation	Link2-7	\$2,000,000.00								
5	L6-8	L6-8 Complete Mitigation	Link6-8	\$500,000.00								
6	L10-16	L10-16 Complete Mitigation	Link10-16	\$1,500,000.00								
7	L10-17	L10-17 Complete Mitigation	Link10-17	\$2,000,000.00								
8	L15-19	L15-19 Complete Mitigation	Link15-19	\$2,500,000.00								
9	L20-21	L20-21 Complete Mitigation	Link20-21	\$3,000,000.00								
10	L20-22	L20-22 Complete Mitigation	Link20-22	\$3,500,000.00								
11												
12												
13												
14												

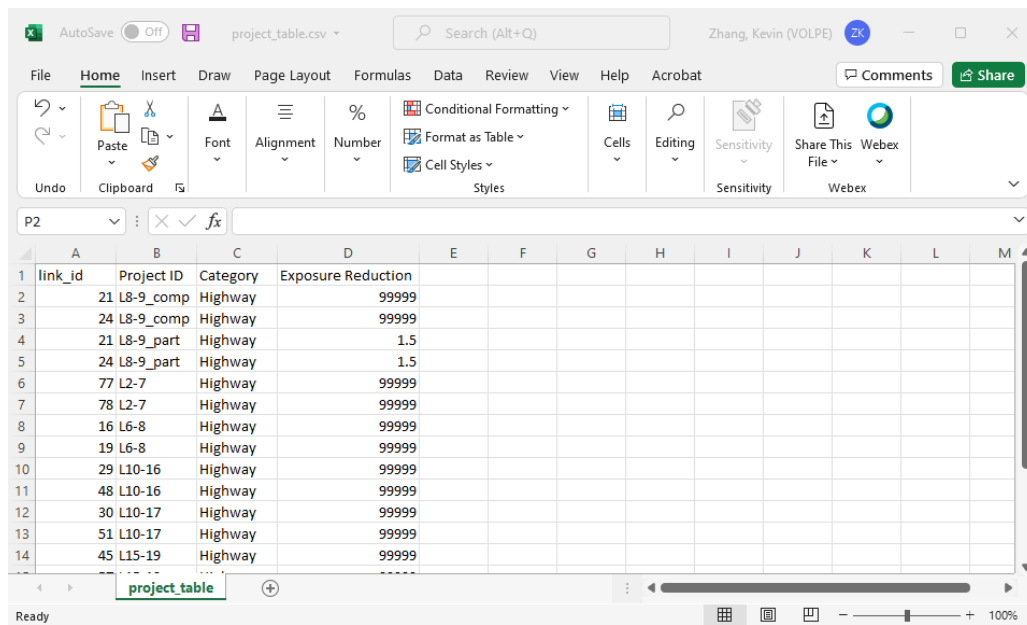
Figure 6-10: Sample project info table for Quick Start 1

The project table file (Figure 6-11) lists all resilience projects and the network links they are associated with. It is named “project_table.csv” and located in the “LookupTables” subfolder of input data folder. Each row of the CSV file represents a link associated with the specified resilience project. There are three required fields in the table:

1. ‘link_id’ – ID of network link.
2. ‘Project ID’ – ID of the resilience project associated with the link. These should match the resilience projects identified in the model parameters file.
3. ‘Category’ – Text string categorizing the resilience project links. The ‘Category’ field values must correspond to the ‘Asset Type’ values in both the repair cost and repair time tables. To use the default repair cost and repair time tables provided with the tool suite, the ‘Category’ field values must be either ‘Highway’, ‘Bridge’, or ‘Transit’.

An additional optional field for partial mitigation functionality:

4. ‘Exposure Reduction’ – Numeric value specifying link-level mitigation associated with the resilience project. This field is required if the ‘resil_mitigation_approach’ parameter in the configuration file is set to ‘Manual’. Blank values in this field imply no mitigation on the project link; a value of 99999 is used to denote complete mitigation on the project link regardless of hazard event.



link_id	Project ID	Category	Exposure Reduction
21	L8-9_comp	Highway	99999
24	L8-9_comp	Highway	99999
21	L8-9_part	Highway	1.5
24	L8-9_part	Highway	1.5
77	L2-7	Highway	99999
78	L2-7	Highway	99999
16	L6-8	Highway	99999
19	L6-8	Highway	99999
29	L10-16	Highway	99999
48	L10-16	Highway	99999
30	L10-17	Highway	99999
51	L10-17	Highway	99999
45	L15-19	Highway	99999

Figure 6-11: Sample project table file for Quick Start 1

6.1.7 Configuring the ROI Analysis

The RDR ROI Analysis Tool takes the regression model built by the RDRM and monetizes the benefits and costs for each resilience project across all uncertainty scenarios specified in the user inputs file (Section 6.1.1.3) for the user-specified period of analysis. Once the regression model has been constructed by the RDRM for the analysis framework specified by the user, the majority of ROI analysis inputs are contained within the configuration file (Section 6.1.1.1).

6.1.7.1 Economic Parameters

The user parameters in the configuration file pertaining to ROI analysis are contained in the [analysis] section. Most parameters have default values provided in the sample configuration file; these are primarily referenced from USDOT BCA guidance. These include values for: year-on-year discounting factors, vehicle occupancy rates, vehicle operating costs, value of time parameters, and monetization factors for safety, noise, and emissions benefits. If the user has more specific values for their region of analysis, they can set these in the configuration file.

Three parameters in the configuration file should be noted in particular. The 'roi_analysis_type' parameter defines the type of ROI analysis the user wants to perform. It can be set to 'BCA', 'Regret', or 'Breakeven'. The 'base_year' and 'future_year' values, found in the [common] section of the configuration file, correspond to the trip tables provided for the core model (see Section 6.1.4.4). Note that 'base_year' and 'future_year' cannot be the same year; the shortest period for the ROI analysis is two years. In addition, a required input file is a full set of base year core model runs.

The base year core model runs must be provided in a file named "Metamodel_scenario_{SP/RT}_baseyear.csv" where the bracketed text is either 'SP' or 'RT' as specified in the configuration file. The input file should be located in the input data folder, with required fields 'hazard', 'recovery', 'trips', 'miles', and 'hours', all used by the 'recov_calc' module. If the user chooses to calculate car and transit-specific metrics, the input file also has required fields 'lr_trips' (light rail trips), 'hr_trips' (heavy rail trips), 'bus_trips', 'car_trips', 'lr_miles', 'hr_miles', 'bus_miles',

'car_miles', 'lr_hours_wait' (wait time for light rail), 'hr_hours_wait' (wait time for heavy rail), 'bus_hours_wait', 'lr_hours_enroute' (time in transit for light rail), 'hr_hours_enroute', 'bus_hours_enroute', and 'car_hours'. The base year core model runs should be run for the specified base year with the corresponding base year hazard event and recovery stage (e.g., no sea-level rise if that is only a future year condition) and with the base year demand. A sample base year core model runs file from Quick Start 1 is shown in Figure 6-12.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	socio	projgroup	resili	elasticity	hazard	recovery	trips	miles	hours	lost_trips	extra_miles	extra_hours	circuitous_trips_removed	
1	baseyear		no	-1	haz1	0	176226.98	739242.7	19571.067	112253.02	-1702.8455	28.539303	93.018415	
2	baseyear		no	-1	haz1	1	220918.81	1283842.1	35437.2	67561.19	256264.03	8044.1505	67561.19	
3	baseyear		no	-1	haz2	0	284986.76	1330465.4	35437.067	3493.2399	10243.506	316.27725	3493.2399	
4	baseyear		no	-1	haz2	1	288480	1329720	35437.067	0	0	0	0	

Figure 6-12: Sample file of base year core model runs for Quick Start 1

The RDR Tool Suite contains a helper tool to create the base year core model runs input file, see [Appendix G: Base Year Run Helper Tool](#). The helper tool uses the core model input files to iteratively run the base year trip table demand through AequilibrE for each hazard event and recovery stage combination.

6.1.7.2 Selection of economic performance metrics

There are three methods for evaluating project performance in the RDR Tool Suite: Benefit-Cost Analysis (BCA), Benefit-Cost Analysis under uncertainty (BCA-U, also referred to as BCA-Regret analysis), and breakeven analysis. The user must specify which method they are choosing for the RDR scenario using the 'roi_analysis_type' parameter in the configuration file. Particular user inputs are required to run each method:

- **BCA** – To run a BCA, the user must provide resilience project costs in the resilience project input file "project_info.csv" (Section 6.1.6.2) and hazard event probabilities in the model parameters file (Section 6.1.1.2).
- **BCA-U** – To run a BCA-U, the user must provide resilience project costs, but uncertainty scenario probabilities do not need to be specified and should be set to 1; the user should refer to the "Regret" metrics as reported in the Tableau visualization output (see Section 6.2).
- **Breakeven** – To run a breakeven analysis, the user should set all resilience project costs to zero. Hazard event probabilities should be provided in the model parameters file.

6.1.7.3 *Selection of ROI performance period*

The performance period is specified by the user in the [common] section of the configuration file. The two parameters, 'start_year' and 'end_year', define the beginning and ending years of the performance period for which benefits and costs are calculated by the RDR Tool Suite. Note that these two parameters are distinct from 'base_year' and 'future_year', as the ROI performance period does not necessarily need to coincide with the years for which trip tables are provided to the core model. Core model outputs for the base year and future year are adjusted during the annualization process for the ROI performance period specified by the start year and end year parameters.

6.2 Reporting and Visualization

This section describes how to use the RDR Tool Suite's Reporting and Visualization Module in decision making. The purpose of the RDR Reporting and Visualization Module is to provide the user with a streamlined approach for reviewing the ROI analysis within the analysis team as well as with decision-makers external to the analysis team. For details on how the RDR Tool Suite Reporting and Visualization Module is structured and the data it contains, please see Section 7 in the Technical Documentation.

6.2.1 Opening the Tableau Workbook

The primary outputs of an RDR run are the Tableau visualizations contained in a Tableau workbook (.twbx file extension). The workbook is automatically generated after running the RDR Tool Suite (Section 3) in the "Reports" folder of the output directory specified by the user in the configuration file. The "Reports" folder contains a timestamped Tableau report folder for each run completed. The only contents of this folder are the Tableau visualizations. To open this file, Tableau must be installed. Double-click on the file to open. Note that the Tableau workbook may take some time to open initially as it loads all of the data and filters; subsequent uses of the workbook will be faster.

6.2.2 General guidelines for using the Tableau Visualization

The RDR Tool Suite Tableau visualizations are designed to allow users to quickly assess analysis results for high-level conclusions such as a simple rank-ordering of the projects by performance under the BCA framework as well as to explore in detail how projects perform across the range of scenarios. Each dashboard in the Tableau file is designed to help address particular questions.

- **Benefit-Cost Analysis:** This table shows the results within the BCA analysis or the breakeven analysis. The dashboard provides a simple listing of the projects by their BCA ranking (BCR) or their breakeven value. It also provides a helpful graphic that shows the contribution to the total benefits of the weighted benefits from each independent hazard event included in the analysis which allows comparison of how the projects achieved their BCR and net benefits relative to projects experiencing the same hazards.
- **Regret Dashboard:** This dashboard shows the overall results of the Regret analysis. useful for presenting results to decision makers or external stakeholders. This dashboard provides a simple listing of the projects by their regret ranking with average net benefits and average benefit cost ratio. The dashboard also shows a simple measure of the variability of the net benefits and BCR for each project called the coefficient of variation, which is a unitless measure of the variability of the project performance across all of the hazards analyzed. Lower values mean the project performs more consistently across the scenario space while higher values mean the project performs inconsistently across the scenario space. The table also includes helpful graphical visualization of how a project ranks across all of the different scenarios. This dashboard should

be used if the user wants to see what the spread of a project's rankings are across the entire scenario space and wants to be able to quickly assess how that spread relates to other projects.

- **Asset-Project Dashboard:** This dashboard should be used if the user would like to get a more detailed picture of the performance of the project across the various scenarios.
- **Asset Dashboard:** This dashboard should be used when considering more than one project for a given asset, e.g., one project raises the bridge, while another adds scour protection. The dashboard allows the user to compare the projects for the same asset directly with three scatterplots, a scenario count by ranking bar graph, and a box-and-whisker plot graph of the main benefit categories for each project.
- **Exploratory Dashboard:** This dashboard is useful to explore the outcomes at the most fine-grained level. This dashboard is most useful for looking at the outcome data to see what the relationships are between the scenario parameters and the outcomes.
- **Parameters Dashboard:** This dashboard shows all of the model parameter selections that were made for the analysis.

6.2.3 General guidelines for using results in decision-making

The RDR Tool Suite results can be incorporated directly into many cost-effectiveness decision-making processes. The analytical approach relies fundamentally on BCA. Users should be careful to note that the tool only estimates the resilience benefits of a project. If a project is expected to provide more benefits than those that accrue from mitigating the impact of a hazard, such as expanding capacity of an asset under normal operating conditions by adding an additional lane, the RDR Tool Suite tool does not estimate the benefits of those non-resilience aspects of the projects. The user will need to compute those non-resilience benefits using some other means. The RDR Tool Suite can be used to estimate the benefits of investments in asset resilience. Users should be very careful not to double count the capital costs and maintenance costs of the resilience project. If another BCA has been performed for an investment, and the user is using the RDR Tool Suite to calculate only those resilience benefits of the project (those produced by mitigating a hazard rather than those that accrue under normal operation), then the Breakeven analysis approach should be used as it provides only the estimate of benefits rather than net benefits.

6.2.4 General guidelines for comparing RDR to other analyses, e.g., REMI

The RDR Tool Suite provides BCA, BCA under Uncertainty, or breakeven analysis results that can be incorporated into decision-making processes that involves cost effectiveness. Some users may wish to incorporate the RDR Tool Suite results into other decision-making processes and frameworks. The RDR Tool Suite may or may not be appropriate for these approaches, and users should carefully consider whether the RDR Tool Suite results are compatible with other frameworks or analyses. In particular, users may wish to incorporate the RDR Tool Suite results into an Economic Impact Assessment (EIA) or compare the RDR Tool Suite results with those of an EIA. Users should avoid comparing the RDR Tool Suite results, which are based on the BCA framework, with those of EIA because the two frameworks are different. An EIA considers how the composition of the economy is changing as a result of a decision focusing on macro-economic factors such as sectoral output, wages, and tax revenue, etc. BCA does not consider these compositional changes but rather focuses on the direct real loss or gain of resources (e.g., gasoline used, time traveled, and crash costs, etc.) due to the project. BCA compares the project performance against a baseline which EIA does not do. BCA considers costs and benefits to all of society, rather than just a single region as is typical with an EIA. BCA also ignores transfers between parties that do result in a real resource whereas transfers between parties is an essential component of an EIA (e.g.,

EIA considers changes in tax revenue which BCA does not consider because it is a transfer). These crucial differences between EIA and BCA make it necessary for users to avoid comparing the results of the RDR Tool Suite with those of an EIA.

Some EIA tools estimate impacts that are appropriate for BCA in addition to those appropriate for the EIA. These tools calculate direct non-transfer costs and benefits such as the impact of the project on travel such as travel time savings, vehicle operating cost savings, and safety benefits from reduced crashes, etc.³⁰ The estimates of the BCA benefit categories using these models in principle can be compared to those calculated by RDR Tool Suite. However, users should also be careful to determine whether the EIA tools are estimating benefits under the same hazard scenarios that are modeled under the RDR Tool Suite. The typical EIA model does not estimate the impact of hazardous events on network performance in the way that RDR does.

³⁰ REMI's Transight is one such model. <https://www.remi.com/model/trans-sight/>

7 RDR Equity and Benefits Analysis Tool

The RDR Equity and Benefits Analysis Tool allows users to disaggregate and summarize the results of an RDR scenario in relation to a user-supplied variable. The tool was originally developed to assess social equity, but it could be used to explore how impacts and benefits are related to any user-supplied variable that can be associated to TAZs. There are two components:

- **Equity Overlay (run_equity_overlay.bat)** – If the user does not yet have any data for the variable of interest, or has data but it is at some other geospatial aggregation other than TAZ (e.g. Census blocks, block groups, or tracts), the user can use this tool to associate the data with TAZ. Regardless of whether the user uses this Equity Overlay method, the user must have data at the TAZ level to run the TAZ Metrics component of the tool successfully.
- **TAZ Metrics (run_TAZ_metrics.bat)** – Once the user has data at the TAZ level (either through use of the Equity Overlay described above or derived through some other means), the user can run this tool to generate CSV files and HTML files that convey metrics of interest by category.

The tool then summarizes the results of an RDR scenario in terms of that variable of interest, as described below.

7.1 Running the Equity Overlay Script

This part of the analysis tool is optional. If users already have an assignment of TAZ used in the RDR runs to variable values, they can provide that as a CSV file which will be used in the TAZ Metrics component of the tool (described in the following section). The format of the required CSV file is as follows:

Table 7-1 Example TAZ categorization CSV file

TAZ	<User-Supplied Variable>
653	0
654	0
656	0
657	0
680	0
681	1
686	1
687	0
688	0

In the table, “TAZ” is the identifier for TAZ used in RDR, and <User-Supplied Variable> is the variable of interest. The name of the CSV file is specified in the ‘output_name’ variable in the equity_metrics.config file, while the TAZ identifier is specified in the ‘TAZ_col_name’ variable and the <User-Supplied Variable> in ‘equity_feature’.

If a user does not already have such a file prepared, they can use the RDR Equity Overlay method to generate this file.

To use the Equity Overlay method, begin by confirming that the user has the required inputs.

Inputs

- Shapefile of MPO zones to join with the layer containing the variable of interest. These are assumed to be the TAZ which are used in the AequilibraE runs but can be census tracts or other geographies. Specify the location of this using the TAZ_source parameter in the configuration file, described in the last bullet of this list.
- (Optional) Shapefile with data that the user wishes to overlay and associate with the TAZ. If the user does not have such a file, the user can still run the tool by identifying an online feature server and feature as a data source. By default, the tool uses the [USDOT Transportation Disadvantaged Census Tracts](#) feature server, and the 'OverallDis' feature, which is a binary classification that combines six disadvantage theme indicators (Transportation, Health, Economy, Equity, Resilience, and Environmental). This equity categorization is at the Census tract level. Regardless of whether the user chooses to use a local file or a feature server, the user will specify the data with the equity_source and equity_feature parameters in the configuration file, described in the last bullet of this list.
- Configuration file indicating which equity layer to use and other parameters (equity_metrics.config). The user should review the parameters in this configuration file and change them, as needed. The Equity Overlay method uses parameters from the first two sections of the configuration file ("common") and ("equity overlay"). Before doing this, the user may wish to save a new version of the configuration file and rename it with a descriptive name (perhaps matching the run_id specified in this version of the configuration file) as a record of the parameters used for later reference. If using this, before attempting to double-click and run the run_equity_overlay.bat file the user will first need to "right-click" to open and edit the run_equity_overlay.bat file (using a text editor or code editor application) to point to the renamed configuration file.

Outputs

- Output CSV file with TAZ identifier and user-supplied variable columns. Specify the file name for this output using the output_name parameter in the configuration file, described above. The next component of the equity and benefits analysis tool, TAZ metrics, will look for and use a CSV file of this name.

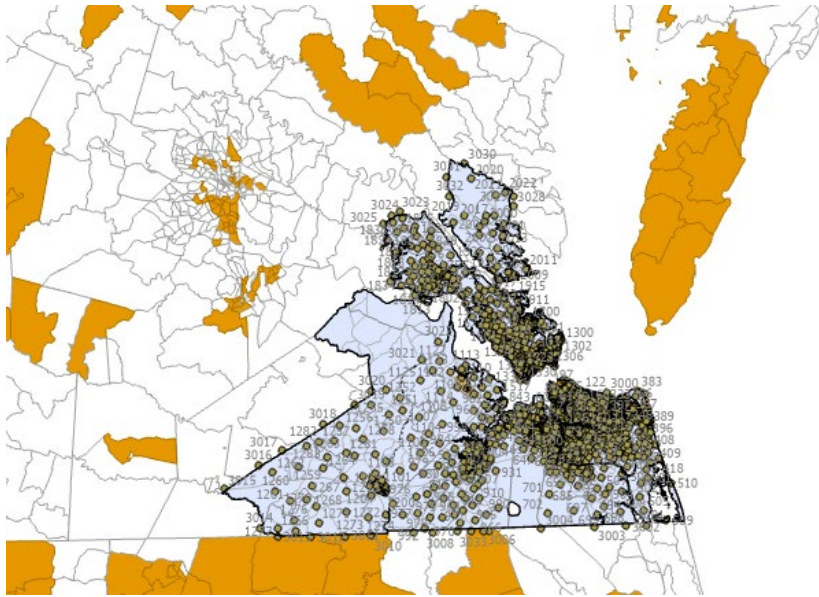


Figure 7-1 View of USDOT Transportation Disadvantaged Census Tracts for an area of Virginia. Orange indicates disadvantaged census tracts, blue shows TAZ for the study area, and nodes of the travel model are shown as points. Map shown for illustrative purposes; the helper tool does not produce maps.

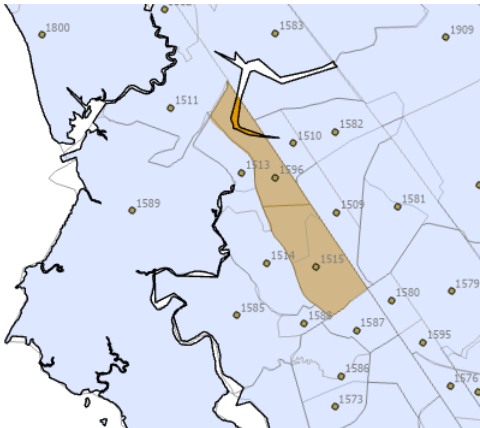


Figure 7-2 Zoomed in view of the overlay of transportation disadvantaged census tracts and TAZ used for RDR modeling. Map shown for illustrative purposes; the helper tool does not produce maps.

ArcGIS Dependency

The equity overlay method depends on ArcGIS Pro being installed on the machine where the method is being run. The assumption is that ArcGIS Pro was installed at the system level, however a user may have ArcGIS installed at the user level. Confirm the location of the python.exe for ArcGIS Pro by navigating in Windows file explorer. If necessary, “right-click” to open and edit the run_equity_overlay.bat file to point to the correct location of the python.exe for ArcGIS Pro, if it is different from the below assumption).

Assumed location of python.exe for ArcGIS Pro installed at a system level:

- C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3

Possible location of python.exe for ArcGIS Pro installed at a user level:

- C:\Users\<Your User Name> \AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\arcgispro-py3

7.2 Running the TAZ Metrics Script

This component of the tool uses a Jupyter notebook to analyze AequilibraE runs (with and without resilience investment) and output an HTML file that reports changes in metrics in relation to the user-supplied variable of interest.

The default assumption is that the user will run the Equity Overlay method (above) as a first step, and then use the output from that as an input to this TAZ metrics analysis. However, the user may also directly provide data assigning a variable value to each TAZ from another source, rather than running the equity overlay analysis. If providing other data, the data must be numeric. The tool will run one of two comparable analyses, depending on the nature of the data:

- If there are fewer than 20 unique values in the data, the tool will run the “MetricsByTAZ_categorical.ipynb” notebook, which treats the variable as a categorical variable.
- If there are 20 or more unique values in the data, the tool will run the “MetricsByTAZ_continuous.ipynb” notebook, which treats the variable as a continuous variable.

Both analyses follow a similar and parallel structure.

The purpose of the outputs is to help the user examine and understand differential impacts of (1) a disruption and (2) resilience investments intended to mitigate the disruption, in relation to the user-supplied variable of interest. The analysis displays variables to help illuminate the following questions from various angles. Questions driving this analysis include:

- How does the baseline magnitude of trips/minutes per trip/miles per trip vary by variable category or continuous value?
- How does the relevance of the disruption vary by category or continuous value?
- How do the projected benefits of the resilience investment vary by category or continuous value? i.e., is there an apparent relationship?

To use this TAZ Metrics method, begin by confirming the required inputs are ready.

Inputs

- CSV file with TAZ identifier and user-supplied variable columns (this can be generated using the Equity Overlay tool, described above, or can be supplied separately). The beginning of the Equity Overlay section, above, shows the required specifications for this file. Identify the file name for this using the output_name parameter in the configuration file, described below, and save it in the location that the user specifies in the equity_analysis_dir parameter of the configuration file.
- equity_metrics.config configuration file - Review the parameters in this configuration file and change them, as needed. Before doing this, save a new version of the configuration file and

rename it with a descriptive name (perhaps matching the `run_id` specified in this version of the configuration file) as a record of the parameters set for later reference. This may already be done if using the Equity Overlay method. If doing this, before attempting to double-click and run the `run_TAZ_metrics.bat` file the user first will need to “right-click” to open and edit the `run_TAZ_metrics.bat` file (using a text editor or code editor application) to point to the renamed configuration file. The configuration file allows the user to specify the following:

- `output_name` is the file name for the CSV that contains the data by TAZ.
- `equity_analysis_dir` is the location where the tool will look for the file mentioned above, and it is also the location where the tool will save all outputs.
- ‘`path_to_RDR_config_file`’ – This should identify the location of the configuration file pertinent to the existing RDR metamodel run and corresponding AequilibraE runs that will be used for this equity and benefits analysis tool run. The analysis will use this configuration file to identify where to access the OMX files from those runs. **Note:** the user will need to prepare the appropriate input files needed for the RDR run as specified in the RDR configuration file, and in the rest of this user guide. The equity analysis tool will use the `input_dir` specified in the RDR configuration file to locate these required inputs. However, it will not use the `output_dir` specified in the RDR config file. Instead, it will save the output files to the location specified in the `equity_analysis_dir` parameter of the equity configuration file.
- ‘`resil`’ - resilience project
- ‘`hazard`’, ‘`recovery`’, ‘`socio`’, ‘`project group`’, ‘`elasticity`’, ‘`run type`’ - AequilibraE scenario dimensions

Outputs

- HTML file showing graphs and some statistical analysis
- CSV files with the underlying data

Recap of the Most Important Instructions Above - Running the Equity Analysis Tool Components in a Nutshell

Prior to running either component (`run_equity_overlay.bat` or `run_TAZ_metrics.bat`), “right-click” to open (using a text editor or code editor application) and then check the corresponding batch (.bat) file and edit, if needed, to set the location of the equity configuration file, if the user changes the name or location from the default, which points to the `equity_metrics.config` file in the equity and benefits analysis tool folder (C:\GitHub\RDR\helper_tools\equity_analysis). If running the equity overlay component, the user may also need to change the path to `python.exe` for ArcGIS Pro in the batch file, if the default location does not match the user’s system.

The user also must prepare the necessary input files, detailed above, and edit the equity configuration file to assign the desired parameters.

After completing the above steps, run each component of the tool by double-clicking on the corresponding batch file (`run_equity_overlay.bat` first, if using, and then `run_TAZ_metrics.bat`) in the equity analysis and benefits tool folder.

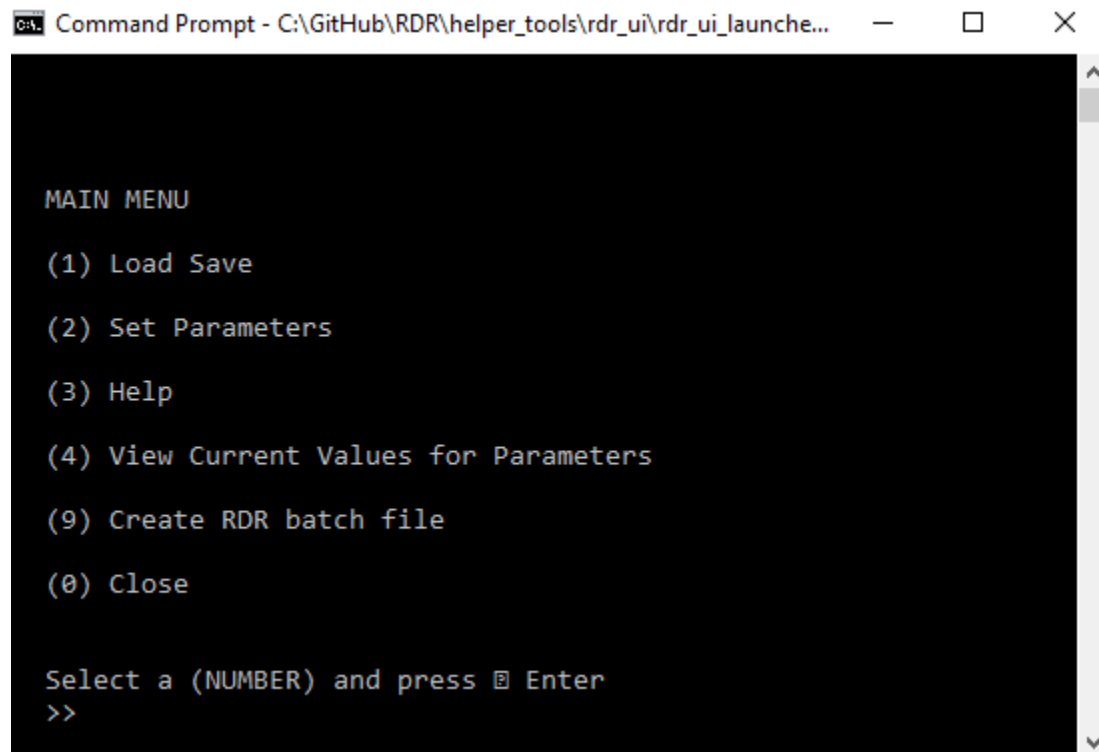
The outputs from each component (except for the intermediary AequilibraE outputs) will be located in the folder specified in the equity configuration file at `equity_analysis_dir`. The outputs from the `run_TAZ_metrics.bat` component will have the equity “run_id” from the equity configuration file in the file name. If the user runs the tool multiple times with the same equity “run_id,” the most recent output will overwrite the prior one of the same name.

8 RDR User Interface

The RDR User Interface (RDR UI) is a command-line interface designed to guide the user through setting up an RDR run. The RDR UI is designed to make RDR more user-friendly and make configuring and running multiple, similar runs simpler. Users can use the RDR UI to set all RDR variables, referred to in the RDR UI as parameters, and to generate an RDR batch file which is associated with the set of parameters that the user specifies. The parameters are saved to a JSON file which RDR can read directly, supplanting the need for a config file and parameter spreadsheets.

8.1 Running the User Interface

To start the RDR UI, navigate to the RDR install location. Enter the `helper_tools` folder, then the `rdr_ui` folder, then right-click to open the `rdr_ui_launcher.bat` file in a text editor (e.g., Notepad, Notepad++, PyCharm, etc.) and ensure that the `rdr_ui_launcher.bat` points to the correct PATH and PYTHON locations. Once correct, double-click `rdr_ui_launcher.bat` to start the RDR UI. A command prompt window should open and the RDR UI splash screen should appear. Press the 'Enter' key to enter the Main Menu, which will appear as in Figure 8-1 below. To select an item from the Main Menu, on the keyboard press the corresponding number to the left of the item and then press the 'Enter' key. Do not enter any other characters other than the number, including space. If the wrong number is entered, press the 'Backspace' (or 'Back' on some keyboards) key before pressing the 'Enter' key to erase the number, then enter the correct number and press the 'Enter' key.

A screenshot of a Windows Command Prompt window titled "Command Prompt - C:\GitHub\RDR\helper_tools\rdr_ui\rdr_ui_launcher...". The window has a black background with white text. It displays the "MAIN MENU" with the following options: (1) Load Save, (2) Set Parameters, (3) Help, (4) View Current Values for Parameters, (9) Create RDR batch file, and (0) Close. At the bottom, it prompts the user to "Select a (NUMBER) and press [Enter]" followed by a double greater-than sign ">>".

```
Command Prompt - C:\GitHub\RDR\helper_tools\rdr_ui\rdr_ui_launcher...

MAIN MENU

(1) Load Save
(2) Set Parameters
(3) Help
(4) View Current Values for Parameters
(9) Create RDR batch file
(0) Close

Select a (NUMBER) and press [Enter]
>>
```

Figure 8-1: Main menu of RDR UI

RDR UI is a command-line interface, so mouse interactions are not needed. All interactions with RDR UI consist of keyboard inputs followed by pressing the 'Enter' key. The only exceptions are that the user can optionally drag and drop files or directories into the command prompt window to provide a path

instead of typing it out or copying and pasting it, and the user may also use the mouse to select and then copy information from some other source and paste it into the command prompt window. No input will be passed to the RDR UI before the 'Enter' key is pressed. In general, the RDR UI will parse all inputs exactly as entered. Note that this means that the RDR UI will not drop any spaces found anywhere in an input, including leading or trailing spaces, nor will it remove spaces in the middle of an input. In addition, an input consisting solely of spaces will be parsed as such.

RDR UI expects commands or assignments when prompted. Commands are inputs that cause the RDR UI to execute an action (see Table 8-1 and Table 8-2), while assignments are inputs that cause the RDR UI to write data. Commands begin with '-', the hyphen or dash. As such, assignments cannot begin with '-', as they will be parsed as commands. This restriction means that no parameter can have a value beginning with '-'.

8.1.1 Basic Commands and Navigation

Table 8-1 shows the universal commands for the RDR UI. Enter them at any time to enact the corresponding actions. While entering parameters, the RDR UI will move the user along sequentially through each parameter entry page. After pressing "Enter," the next parameter entry section will appear. To go directly to a particular parameter entry page rather than following this sequence, use the corresponding shortcut command in Table 8-2. Enter these at any time to visit the pages where you can enter information for the corresponding parameters.

Table 8-1: Universal commands for the RDR UI

Action	Universal Command Options
Return to main menu	['-mm', '-main menu']
Quit program	['-qp', '-exit']
Save as	['-sa', '-save as']
Save	['-s', '-save']
Load save	['-l', '-load']
Create RDR batch file	['-cb', '-batch']
Return to most recently visited parameter page	['-cc', '-current']
Return to most recently completed parameter page	['-bb', '-back']
Return to main help page	['-h', '-help']

Table 8-2: Shortcut commands to access parameter entry pages in the RDR UI

Parameter	Shortcut	Explanation
input_dir	-in	Input directory
output_dir	-ou	Output directory
run_id	-id	Run ID (string used to identify outputs of the run)
start_year	-sy	Start year of analysis period
end_year	-ey	End year of analysis period
base_year	-by	Year of base year core model runs
future_year	-fy	Year of future year core model runs

metamodel_type	-met	Metamodel type, see Table 4-1
lhs_sample_target	-lhs	Latin hypercube sampling sample size
aeq_run_type	-art	AequilibraE model run type
run_minieq	-rme	Mini-equilibrium run
allow_centroid_flows	-acf	AequilibraE paths allowed through centroids
calc_transit_metrics	-ctm	Calculate transit-specific metrics
link_availability_approach	-laa	Link availability approach, see Section 5.2
exposure_field	-exf	Exposure level field name in the exposure input files
link_availability_csv	-lap	Link availability CSV file path
alpha	-alp	Alpha in the beta distribution disruption function
beta	-bet	Beta in the beta distribution disruption function
lower_bound	-lob	Lower bound of the beta distribution disruption function
upper_bound	-upb	Upper bound of the beta distribution disruption function
beta_method	-bem	Direction of beta distribution disruption function
zone_conn	-zoc	Highest node ID designating a centroid
resil_mitigation_approach	-rma	Approach used to convert resilience investment to mitigation
num_recovery_stages	-nrs	Number of hazard recovery stages
min_duration	-mid	Minimum number of days the initial hazard may last
max_duration	-mad	Maximum number of days the initial hazard may last
num_duration_cases	-ndc	Number of initial hazard duration cases
hazard_recov_type	-hrt	Hazard recovery build-out type
hazard_recov_length	-hrl	Hazard recovery build-out length
hazard_recov_path_model	-hrp	Hazard recovery path model
exposure_damage_approach	-eda	Approach used to convert exposure to damage level
exposure_unit	-exu	Units of exposure level in the exposure input files
exposure_damage_csv	-edp	Exposure-damage CSV file path
repair_cost_approach	-rca	Approach used to convert damage to repair cost
repair_network_type	-rnt	Default repair cost network type
repair_cost_csv	-rcp	Repair cost CSV file path
repair_time_approach	-rta	Approach used to convert damage to repair time
repair_time_csv	-rtp	Repair time CSV file path
roi_analysis_type	-rat	Return on investment (ROI) analysis type
dollar_year	-dyr	Year for dollar units
discount_factor	-dfa	Year-on-year discounting factor
co2_discount_factor	-cfa	Year-on-year CO ₂ discounting factor
vehicle_occupancy_car	-occ	Passenger car vehicle occupancy rate
vehicle_occupancy_bus	-ocb	Transit bus vehicle occupancy rate
vehicle_occupancy_heavy_rail	-ocr	Heavy rail transit vehicle occupancy rate
vehicle_occupancy_light_rail	-ocl	Light rail transit vehicle occupancy rate

veh_oper_cost_car	-opc	Passenger car operating cost in dollars per vehicle-mile
veh_oper_cost_bus	-opb	Transit bus operating cost
veh_oper_cost_light_rail	-opl	Light rail transit operating cost
veh_oper_cost_heavy_rail	-opr	Heavy rail transit operating cost
vot_per_hour	-vot	Value of travel time in dollars per hour
vot_wait_per_hour	-vow	Value of waiting time in dollars per hour
transit_fare	-far	Transit fare in dollars per trip
maintenance	-mai	Include annual maintenance cost
redeployment	-rdp	Include redeployment cost
safety_cost	-saf	Safety external use cost per mile driven
safety_cost_bus	-sab	Safety external use cost for transit bus
noise_cost	-nco	Noise external use cost per mile driven
noise_cost_bus	-ncb	Noise external use cost for transit bus
non_co2_cost	-nra	Non-CO ₂ emissions external use cost per mile driven
non_co2_cost_bus	-nrb	Non-CO ₂ emissions external use cost for transit bus
co2_cost	-cra	CO ₂ emissions external use cost per mile driven
co2_cost_bus	-crb	CO ₂ emissions external use cost for transit bus
hazards	-haz	Hazards events in scenario analysis framework
event_frequency_factors	-eff	Event frequency factors in scenario analysis framework
economic_futures	-ecf	Future economic scenarios in scenario analysis framework
trip_loss_elasticities	-tle	Trip loss elasticities in scenario analysis framework
resilience_projects	-rep	Resilience projects to be analyzed
network_links	-net	Network link CSV file paths
net_node	-nwn	Network node CSV file path
proj_table	-prt	Project links lookup table (CSV file path)
proj_cost	-pri	Project costs lookup table (CSV file path)
base_year_file	-byf	Base year core model runs CSV file path
bat_location	-bl	File path for the batch file associated with UI save
python	-py	File path for the user's python.exe used for RDR
rdr	-rd	File path for the user's Run_RDR.py file

8.1.2 Setting Parameters

The "basic commands and navigation" section above describes the basics on how to navigate through the pages for parameter setting. To leave the current value for a parameter as is, simply press Enter and the RDR UI will advance to the next entry page without changing the value for that parameter. The RDR UI displays the current value for a parameter above the input area, as shown in Figure 8-2.

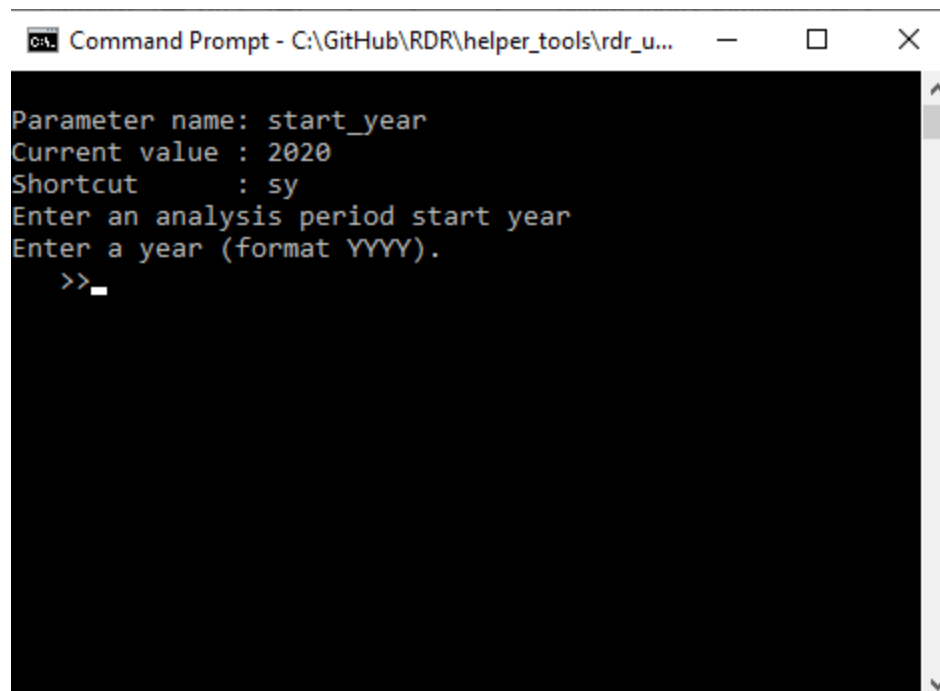


Figure 8-2: Example of parameter entry page in RDR UI

The RDR UI allows the user to enter all the same parameters that would otherwise be typed into the configuration file to create the RDR inputs. RDR UI also allows the user to enter information that the user would otherwise provide in spreadsheet files. For example, instead of providing information on a particular hazard in the UserInputs.xlsx file and Model_Parameters.xlsx file, the user can enter all this information in RDR UI. While the RDR UI supplants the need for some spreadsheet files, the user still needs to provide some others. For example, in the case of hazards, the user will still need to create an exposure analysis file for each hazard (e.g. {hazard_filename}.csv) and provide the paths to these files in the RDR UI. In cases where there are multiple attributes or parameters that “belong” to a particular object (e.g. hazards, resilience projects, economic futures, etc.), the user should provide the parameters for that object in the order specified in the prompt, separated by commas. So, for example, for a hazard, the user would type the name of the hazard, the file path for the corresponding exposure analysis file, dim1 (short-term dimension), dim2 (long-term dimension), and the probability of the hazard occurring in the start year, all separated by commas. For the first hazard in Quick Start 1, the input would look like this:

```
haz1,C:\GitHub\RDR\quick_starts\qs1_full_run\Data\inputs\Hazards\haz1.csv,10,0,0.1
```

After the user has entered the parameters for a particular object, the RDR UI will move on to request data for the next object of that type (e.g. the next hazard). To move from one object to another (e.g., if you are on the entry section for the second hazard and you want to return to the first hazard to change one of its parameters), type two dashes followed by the item number (e.g. ‘--1’ to go to item 1).

To change only a subset of the parameter values for a particular object, use the same comma-delimited notation, but only enter information between the corresponding commas, leaving the other commas as placeholders. For example, if the user had already entered the information for the first hazard, but then

wanted to go back and change the probability of the hazard occurring (which is the fifth of five inputs), it would look something like this:

```
,,,0.2
```

When done setting the parameters for a particular category (e.g., if done setting information for all hazards), the user should type 'done' followed by Enter to move on to the next parameter entry page.

Figure 8-3 shows an example of a data entry page for an object with multiple associated parameters (e.g., a hazard). Note that the correct order in which to enter the parameters is provided in the prompt.

```

Command Prompt - C:\GitHub\RDR\helper_tools\rdr_ui\rdr_ui_launcher.bat
Hazards names, file paths, and numeric attributes
HINT: Type "done" if done setting hazards.
HINT: Type "--NUMBER" (for example: --3) to jump to that item number.

Type value(s) for haz_name, haz_fpath, haz_dim1, haz_dim2, haz_prob separated by commas (,)
OR type "attribute name: attribute value" pairs. For multiple pairs, separate with commas.
>>haz1,C:\GitHub\RDR\quick_starts\qs1_full_run\Data\inputs\Hazards\haz1.csv,10,0,0.1_

```

Figure 8-3: Example of RDR UI data entry page for an object with multiple associated parameters

Note that the RDR UI includes hidden parameters that are not meant to be modified by the user. Some of these are used internally by the RDR UI in the background, while others are meant for use only during RDR UI development. Hidden parameters are clearly marked with a warning indicating that they are a hidden parameter. If the user finds themselves on a hidden parameter page, simply navigate to a different page without making any modifications. Normal RDR UI use will not result in navigation to a hidden parameter page but could occur if a typo is made while using a shortcut. Incorrect modification of hidden parameters could cause unexpected behavior and crashing in the RDR UI.

8.1.3 Building a Batch File

After entering all the parameters in the RDR UI, the final step is to generate a corresponding batch file to enact an RDR run. To do this, from within the RDR UI, type '-bl' followed by Enter, or choose option 9 from the main menu. The RDR UI will display the current save file to the user for verification and then will proceed through a series of a few more prompts to get the information needed to create the batch file, including: location to create the batch file, python.exe location for RDR, and Run_RDR.py location.

Then, if a run id and input directory have not yet been specified, the RDR UI will ask for these. However, the user should already have provided these. Creating the batch file before finishing parameter-setting is not recommended and may cause unexpected behavior.

The RDR UI will then give the user another opportunity to go back and change any of the parameters, or, if everything is set, to proceed without modifications.

The RDR UI will then provide a final advisory message and warning, as follows:

Files associated with hazards, economic futures, network links and nodes, and project segments and costs will now be placed into directories that RDR will create in the RDR input directory. Any existing directories within the input directory named "Hazards", "Networks", or "LookupTables" and their contained files will be overwritten.

The user must press Enter to continue. The RDR UI will then build the batch file and validate and relocate the input files to their standard locations in the specified input directory.

8.2 Help and Documentation

Access the RDR UI help page at any time by typing either one of these two shortcut commands and pressing Enter: ['-h', '-help']. Alternatively, choose option 3 from the main menu. The help page displays the commands that appear in Table 8-1 and Table 8-2 above and shows the current value for each parameter.

8.3 Saving and Loading

Save at any time by typing either one of these two shortcut commands and pressing Enter: ['-s', '-save']. 'Save as' (i.e., save a copy under a new name) at any time by typing either one of these two shortcut commands and pressing Enter: ['-sa', '-save as']. The first time the user attempts to save and for every time the user attempts to 'save as,' RDR UI will prompt the user to identify a save location and file name. Save files are stored as JavaScript Object Notation (JSON) files with a file extension of ".save" and they are human-readable, meaning that the user can open them in a text editor and read their contents.

To load a save file in the RDR UI, which will load the values of previously entered parameters and recommence where you left off, type either one of these two shortcut commands and press Enter: ['-l', '-load']. Alternatively, choose option 1 from the main menu.

The 'save,' 'save as,' and 'load save' capabilities of the RDR UI are useful for replicating runs and setting up several similar runs. Paired with the ability to generate a batch file associated with each save, the RDR UI enables the user to store and run many different RDR scenarios quick and easily. These save file – batch file pairs are called a 'run profile'. Maintaining a separate Data directory for each run profile is recommended. If several run profiles are very similar to each other, rather than maintaining separate Data directories, keeping separate input and output directories for each run profile within the same Data folder is recommended.

9 Troubleshooting

9.1 Installation

- 1) Some users may encounter a message about the Python package CFFI when building the conda environment. This package ([C Foreign Function Interface](#)) is what [rpy2](#) uses to run R code from within Python. If the user receives a message about '[ABI mode](#)', the user needs to run the following in the Windows command line:

```
o set RPY2_CFFI_MODE=ABI
```

Then the user should be able to proceed as outlined in Section 2.3.

- 2) Users should confirm that the Anaconda installation and AequilibraE package installation³¹ on their machine are located in areas where they have read/write access.
- 3) Users having difficulty creating the RDRenv Anaconda environment may need to run the following in Anaconda Prompt before trying again:

```
o conda config --set channel_priority false
```

The conda command `conda clean --all` may also be useful for freeing up space and removing old versions of packages.

- 4) Running an RDR batch file downloaded from the public GitHub repository may be blocked by the user's security settings by default. To get around this issue, right-click on the batch file, go to "Properties", go to the Security tab, and allow "Read & execute" permissions for the user.
- 5) If the user has trouble with R package installation during their first RDR run, they can work around this issue by running the "rdr_rutil.R" file (in the "metamodel_py" subfolder of the main RDR directory) in the R installation of their RDRenv conda environment before executing a RDR run.
- 6) The RDR Tool Suite creates a custom R installation within the RDRenv conda environment during the setup process in order to ensure the correct versions of R packages are used. Depending on the user's previous R installations, the tool suite may access the wrong R package libraries, leading to dependency errors. The R variable `.libPaths()` can be used to assess which R installation is being referenced by the code. The user may find it necessary to (1) delete older versions of R from their local machine, (2) re-create the RDRenv Anaconda environment.

9.2 Input Data Preparation

- 1) It is important the user confirm that all dollar inputs are converted to the same year ('dollar_year' parameter specified in the configuration file). Dollar inputs to the RDR Tool Suite include (1) vehicle operating costs, value of travel time, value of transit wait time, and transit fare parameters in the configuration file, (2) 'Project Cost', 'Annual Maintenance Cost', and 'Redeployment Cost' fields in the project info file, (3) repair costs ('Damage Repair Cost' and 'Total Repair Cost' fields) in the repair cost look-up table, (4) link toll values ('toll' field) in the

³¹ This version of RDR was developed using AequilibraE version 0.9.2.

network attribute input files, and (5) safety, noise, and emissions monetization values in the configuration file and look-up tables. Note that link-level tolls should be provided in cents, not dollars. These units should be consistent to provide accurate results in the economic analysis and project prioritization.

- 2) Trip table matrices as stored in OMX input files should be in a square matrix format, not a tall table indexed by origin-destination pair. The RDR Tool Suite includes helper tools for converting among the various formats.
- 3) Large numeric values for the network node and link IDs can cause errors with Aequilibræ due to a memory issue. If this occurs, renumber the IDs starting from 1 across all input files.
- 4) A common reason for unexpected results or zero reported benefits is a mismatch in formatting between input files. The RDR Tool Suite relies on several data table joins to pull together data across input files to calculate ROI—if two columns are supposed to be matched to each other but one is provided as a text string and the other is numeric, the data table join will fail. For example, the 'Category' column in the project table input file is matched to the 'Asset Type' column in the repair cost table. If highway functional class labels are used, but 'Category' is numeric (e.g., 1) and 'Asset Type' is text ("1.0"), then the RDR Tool Suite will fail to associate resilience project network links with their repair costs, resulting in zero benefits from mitigating incurred repair costs.

9.3 RDR Scenario Run Execution

- 1) The first place to diagnose an error or unexpected output from the RDR Tool Suite is the command window where the batch file was executed. If a fatal error occurs during a run, an error message will be written to the screen specifying the location in the code where an error was reached. Exception messages provide more detailed feedback on steps to take to resolve the error.
- 2) The text report generated by the 'o' module contains a compilation of errors and warnings encountered during a run. This can be used to diagnose unexpected results.
- 3) More detailed information on the run can be found in the log files. Log files are automatically generated during a run of the RDR Tool Suite in the "logs" folder of the output directory and are labeled according to the RDR Tool Suite module they correspond to. These output files provide detailed reporting on the tool run and can be used to troubleshoot any errors encountered by the user. Messages labeled "DEBUG" or "WARNING" provide step-by-step information.
- 4) The default metamodel method is called Gaussian multitarget regression. This method estimates the changes in trips, miles, and hours in one statistical analysis. If there aren't sufficient core model runs to support this method, the metamodel module will use a simple linear regression for each of trips, miles, and hours separately. A message will appear in the log file and the selected method is noted in the `rdr_Metamodel_Regression_<Run_ID>.html` report.

- 5) Intermediate output files are generated by each module of the RDR Tool Suite. Descriptions are provided in [Appendix B: RDR Tool Suite Module Outputs](#).
- 6) Some users using certain installations of conda may see the following messages when running an RDR bat file. However, the program still runs to completion with no additional errors.
 - a. 'activate' is not recognized as an internal or external command, operable program or batch file.
 - b. 'conda.bat' is not recognized as an internal or external command, operable program or batch file.
- 7) The RDR User Interface may overwrite parameter values set by the user for non-config parameter types. If items in a non-config parameter appear to have been automatically overwritten by the RDR UI, use the "deleteALL" command and re-enter the non-config parameter that has been affected.

9.3.1 Executing a Run for the First Time

The first run involving the AequilibraE core model should be performed with at most one hazard and one mitigation, with the reasonableness of baseline (no hazard) results being checked after the run. Items to check:

1. Do the trips have consistent units with the network capacities? That is, if the trip tables have trips / day, then link capacities should be per day. Similarly if the trip tables have trips / peak hour, then link capacities should be per hour.
2. In the Data/generated_files folder, examine AequilibraE_Runs_Compiled.xlsx
 - Is the total number of trips (per day or per peak hour) as expected?
 - Calculate miles / trip. Is the result reasonable?
 - Calculate hours / trip. Is the result reasonable?
3. Examine the link flows from 1+ car households, e.g., files like Group01_baserun.csv and link_flow_base01.csv located in Data\generated_files\aeq_runs\base\base01\matrix
 - What are the flow/capacity ratios (should mostly be less than 1)?
 - What flows are coming out of centroids (should roughly equal the number of interzonal trips for 1+ car households)? If the flows coming out of centroids are much higher, it means that centroid connectors are being used as intermediate links on trips.
 - Appendix B of the RDR Technical Document contains some tips on validation.
4. For a network that includes transit, examine the link flows from 0-car households, e.g., files like Group01_baserun.csv and link_flow_base01.csv located in Data\generated_files\aeq_runs\base\base01\nocar
 - See which flows are using road centroid connectors vs. transit centroid connectors ("AequilibraE model validation with transit network", located in Appendix B of the RDR Technical Document contains specific tips on how to do this).
 - One would expect to see higher transit use with the trips from 0-car households.
5. Additional items to check are contained in Appendix B of the RDR Technical Document. Some of these pertain to later full runs, when the complete set of hazards has been added to the model.

9.4 RDR Output Analysis

- 1) The components that comprise the benefits calculation are (1) network performance benefits accrued during the hazard period, (2) network performance benefits accrued during the repair period, and (3) repair cost savings.
- 2) The regression model in the RDR Metamodel extrapolates network performance metrics from core model runs to the entire scenario space. These are then extrapolated further in the ROI Analysis Module to construct benefits and costs for the whole period of analysis. Users should check the regression output files to confirm that resilience project impacts are trending in reasonable directions compared to each no action baseline scenario (e.g., resilience projects increase number of trips taken, decrease circuitousness per trip taken, etc.). If the regression is producing poor results, the user may need to increase the number of core model runs done or switch the type of regression model to better fit their scenario space.

10 Conclusion

The RDR Tool Suite enables MPOs and state DOTs to evaluate resilience ROI across a range of future scenarios and hazard conditions as part of their project prioritization process, an approach that builds on previously available resources and methodology.

The approach is intended to be location and hazard agnostic, such that any state DOT or MPO can utilize it.

The RDR Tool Suite can be accessed at: <https://volpeusdot.github.io/RDR-Public>.

Appendix A: RDR ROI Analysis Tool Input Files

Table A-0-1: RDR Tool Suite Input Files

Input File	Location	Filename	Required Fields	Modules	Notes
Configuration file	User-defined; location of configuration file should match path specified in batch file	User-defined; filename should match batch file	See documentation within sample configuration file	'lhs' 'aeq_run' 'rr' 'recov_init' 'recov_calc'	Specifies input and output folders
Model parameters file	Input directory 'input_dir' in configuration file (e.g., Data\inputs)	Model_Parameters.xlsx ("UncertaintyParameters", "ProjectGroups", and "Hazards" tabs)	See Section 6.1.1.2	'lhs' 'recov_init' 'recov_calc'	Specifies all possible values for each dimension of scenario space, list of hazard event stages
User inputs file	Input directory 'input_dir' in configuration file (e.g., Data\inputs)	UserInputs.xlsx ("UserInputs" tab)	'Hazard Events', 'Economic Scenarios', 'Trip Loss Elasticities', 'Resiliency Projects', 'Event Frequency Factors'	'recov_init'	Specifies subset of scenario space to evaluate in RDR ROI Analysis Module
Resilience project files	Subfolder "LookupTables" in input directory (e.g., Data\inputs\LookupTables)	project_table.csv	'Project ID', 'link_id', 'Category'	'aeq_run' 'recov_init'	Maps resilience project to network links
Resilience project files	Subfolder "LookupTables" in input directory (e.g., Data\inputs\LookupTables)	project_info.csv	'Project Cost', 'Project ID', 'Project Name', 'Asset'	'recov_calc'	Defines project types and costs

Input File	Location	Filename	Required Fields	Modules	Notes
Network attribute files	Subfolder “Networks” in input directory (e.g., Data\inputs\Networks)	node.csv	‘node_id’, ‘x_coord’, ‘y_coord’, ‘node_type’	‘aeq_run’	Provides node attributes for each node in the network
Network attribute files	Subfolder “Networks” in input directory (e.g., Data\inputs\Networks)	{econ}{projgroup}.csv (one needed for each project group + economic scenario combination specified in model parameters file)	‘link_id’, ‘from_node_id’, ‘to_node_id’, ‘directed’, ‘length’, ‘lanes’, ‘facility_type’, ‘capacity’, ‘free_speed’, ‘allowed_uses’, ‘travel_time’, ‘toll’	‘aeq_run’ ‘recov_init’	Provides link attributes for each link in the network
Exposure analysis files	Subfolder “Hazards” in input directory (e.g., Data\inputs\Hazards)	{hazard_filename}.csv (one needed for each hazard event specified in model parameters file)	‘link_id’, ‘from_node_id’, ‘to_node_id’, exposure_field name specified in configuration file	‘aeq_run’ ‘recov_init’	Provides hazard exposure data at the link level
Demand files	Subfolder “AEMaster\matrices” in input directory (e.g., Data\inputs\AEMaster\matrices)	{econ}_demand_summed.omx (one needed for each economic scenario specified in model parameters file)	‘matrix’ trip table, ‘taz’ mapping	‘aeq_run’	Provides origin-destination trip table data for AequilibraE
Base year core model runs file	Input directory ‘input_dir’ in configuration file (e.g., Data\inputs)	Metamodel_scenarios_{SP/RT}_baseyear.csv	‘hazard’, ‘recovery’, ‘trips’, ‘miles’, ‘hours’	‘recov_calc’	Provides core model outputs for base year
True shapes file (optional)	Subfolder “LookupTables” in input directory (e.g., Data\inputs\LookupTables)	TrueShape.csv	‘link_id’, ‘WKT’	‘aeq_run’	Provides link attributes for true shape network

Input File	Location	Filename	Required Fields	Modules	Notes
SQLite database	Subfolder “AEMaster” in input directory (e.g., Data\inputs\AEMaster)	project_database.sqlite	N/A	‘aeq_run’	Stores network data for AequilibraE
AequilibraE look-up table files (optional)	Subfolder “LookupTables” in input directory (e.g., Data\inputs\LookupTables)	link_types_table.csv	‘facility_type’, ‘alpha’, ‘beta’	‘aeq_run’	Customizes volume-delay function parameters for AequilibraE
Exposure-disruption table file (optional)	User-provided path in configuration file	User-defined	‘min_inclusive’, ‘max_exclusive’, ‘link_availability’	‘aeq_run’	Converts hazard exposure to link-level disruption
Exposure-damage table file (optional)	Subfolder “config” in main RDR directory (e.g., RDR\config), or user-provided path in configuration file	default_exposure-damage_table.csv; or user-defined	‘Asset Type’, ‘min_exposure’, ‘max_exposure’, ‘Damage (%)’	‘recov_init’	Converts hazard exposure to link-level damage
Repair cost table file (optional)	Subfolder “config” in main RDR directory (e.g., RDR\config), or user-provided path in configuration file	default_repair-cost_table.csv; or user-defined	‘Asset Type’, ‘Facility Type’, ‘Damage Repair Cost’, ‘Total Repair Cost’	‘recov_init’	Converts link damage to repair cost
Repair time table file (optional)	Subfolder “config” in main RDR directory (e.g., RDR\config), or user-provided path in configuration file	default_repair-time_table.csv; or user-defined	‘Asset Type’, ‘min_inclusive’, ‘max_exclusive’, ‘repair_time’	‘recov_init’	Converts link damage to repair time

Appendix B: RDR ROI Analysis Tool Module Outputs

Several output files are generated by the RDR Tool Suite, including intermediate outputs passed between modules and final outputs for user analysis. This section provides a more in-depth description of these outputs, separated by module.

Outputs of 'lhs' Module

The 'lhs' module has two main output files. The 'full_combos.csv' file provides a full list of the scenario space defined by the model parameters input file. Each row represents an uncertainty scenario as defined by: (1) a hazard event, (2) a recovery stage, (3) an economic scenario, (4) a trip loss elasticity value, (5) a project group, and (6) a resilience project (or baseline of no resilience investment). The 'full_combos.csv' file is used by the 'rr' module to fill in model results for the entire scenario space.

The other output file of the 'lhs' module is the 'AequilibraE_LHS_Design_{lhs_sample_target}.csv' file, where the bracketed text is a stand-in for the value of the 'lhs_sample_target' parameter in the configuration file. The file provides a list of the uncertainty scenarios chosen by the Latin hypercube sampling algorithm to be run with AequilibraE. The file structure is similar to the 'full_combos.csv' file, with an additional column indicating the scenarios chosen. The file is used by the 'aeq_run' module to look up which uncertainty scenarios to execute an AequilibraE core model run for.

Outputs of 'aeq_run' Module

The 'aeq_run' module generates several outputs for each core model run, organized within the "aeq_runs" subfolder of the output data folder. The module executes a series of base network and disrupted network runs to generate the core model results used to build the regression model. Within the "aeq_runs" subfolder, there are separate subfolders for the base network runs ("base") and the disrupted network runs ("disrupt"). Within each of these subfolders, there is a separate folder for each uncertainty scenario run by the core model, indexed by the 'run_id' parameter in the configuration file.

For a single uncertainty scenario, several intermediate output files are created during the AequilibraE core model run. A copy of the SQLite database is created and updated using data from a base network CSV file and a disrupted network CSV file. Additionally, a disrupted network link availability CSV file is generated. The AequilibraE software outputs several files, including a link flow CSV file ('link_flow_adjdem_{scenname}.csv'). AequilibraE also produces several OMX files, which are placed in the "matrices" subfolder. They include shortest path skims ('sp_{basescenname}.omx') and routing skims ('rt_{basescenname}.omx') for both base and disrupted network runs, and a demand file named 'new_demand_summed.omx'. The AequilibraE core model run also produces a link 'link_flow_full.csv' file and corresponding GeoJSON files ('node.json' and 'link_flow_full.json') for easier visualization of link flows and hazard / resilience project impacts on network performance for the scenario.

The primary output file for a single AequilibraE run is the 'NetSkim.csv' file for the disrupted network run. The 'NetSkim.csv' file provides the core model results in total trips, total miles traveled, and total hours traveled for the uncertainty scenario. If the user specifies in the configuration file that mode-specific performance measures should be recorded as well, the 'NetSkim.csv' file also includes additional columns for unlinked trips, passenger miles traveled, transit wait times, and passenger hours in transit

by mode (car, bus, light rail, heavy rail). All of these output files, for the uncertainty scenarios selected for core model runs, are used by the 'aeq_compile' module to create a single output file of core model results to build the regression model.

Outputs of 'aeq_compile' Module

The 'aeq_compile' module has one main output file, named 'AequilibraE_Runs_Compiled_{run_id}.xlsx' where the bracketed text is replaced by the 'run_id' parameter in the configuration file. The output file is a compilation of all of the 'NetSkim.csv' output files from AequilibraE across the uncertainty scenarios run by the core model. The file is used by the 'rr' module to build the regression model that is the main result of the RDR Metamodel.

Outputs of 'rr' Module

The 'rr' module has one main output file, named 'Metamodel_scenarios_{SP/RT}_futureyear.csv' where the bracketed text is either 'SP' or 'RT' depending on the value provided for the 'aeq_run_type' parameter in the configuration file. The output file joins the 'full_combos.csv' output file from the 'lhs' module with regression model outputs for total trips, total miles traveled, and total hours traveled, and provides model results for all uncertainty scenarios in the scenario space. If the user specifies that mode-specific performance measures should be calculated as well, these are output in the CSV file. The file is used by the RDR ROI Analysis Tool, and particularly the 'recov_calc' module as an input to the economic analysis and project prioritization. The 'rr' module also produces another output file, named 'rdr_Metamodel_Regression.html', an RMarkdown report that summarizes the inputs and outputs of the regression models, as well as model fit metrics.

Outputs of 'recov_init' Module

There are four main output files of the 'recov_init' module. The recovery module creates the 'uncertainty_scenarios.csv' file of all combinations built out from the 'UserInputs.xlsx' input file, including hazard recovery paths and resilience projects with baselines. It also creates the 'extended_scenarios.csv' file, which separates the uncertainty scenarios into the distinct stages of their exposure recovery and repair recovery paths. The damage and repair models within the 'recov_init' module create the 'repair_calculator.csv' file reporting damage, repair cost, and repair time at the link level, and the 'scenario_repair_output.csv' file reporting damage, repair cost, and repair time at the uncertainty scenario level. These files all feed into the ROI analysis and project prioritization done by the 'recov_calc' module.

Outputs of 'recov_calc' Module

The 'recov_calc' module has three main output files. The main output file is named 'tableau_input_file.xlsx' and combines the outputs of all previous modules for economic analysis and project prioritization. Benefit-cost analysis metrics are included in this output file. This file is used as the input to create another one of the output files, the Tableau dashboards packaged together in the 'tableau_dashboard.twbx' workbook file (located in the "Reports" subfolder of the output data folder). The data visualizations and project prioritization analysis found in the Tableau dashboards are described in Section 7 of the technical documentation. The 'recov_calc' module also produces a third output file, named 'bca_metrics.csv', which is similar to 'tableau_input_file.xlsx' but includes

all “no resilience investment” baseline scenarios (which are combined in the latter to produce a single average baseline scenario).

Other Output Files

Additional output files include logs for each module (located in the “logs” subfolder of the output data folder) and a report of the RDR Tool Suite run (located in the “Reports” subfolder of the output data folder) that summarizes where results are located, how long the modules took to run, what configuration parameters were used for the run, and any warnings or errors encountered during the run. The log files are labeled according to the RDR Tool Suite module they correspond to. The report is generated by the ‘o’ module. These output files provide detailed reporting on the tool run and can be used to troubleshoot any errors encountered by the user.

Appendix C: Glossary of Terms Relevant to RDR Tool Suite

Analysis period	Time period over which the project benefits accrue
Asset	A physical structure on which a project is implemented (e.g., bridge, highway)
Availability	Network link capacity to allow travel
Base run	A core model run without hazard disruption applied to the network
Base year	The year that the initial core model is run with base year trip tables (this can be different than the start year of the period of analysis)
Baseline	The baseline scenario is the scenario in which the hazardous event occurs but no project is deployed
Core model run	Scenario runs selected by the Latin Hypercube Sampling module to populate the regression that estimates network statistics for all scenarios in the scenario space
Damage	Damage refers to the physical and structural degradation or disrepair of a transportation asset
Discount/Dollar Year	The year that future monetary values will be discounted to
Discounting	The process of adjusting future time period values of cost and benefits to an equivalent value in the present period to account for the time value of money
Disrupt run	A core model run with hazard disruption applied to the network
Disruption	Loss of transportation asset operational capacity due to hazard exposure or due to damage
Economic scenario	The economic scenario refers to the different land-use and economic activity patterns that could emerge over the period of analysis
End year	The last year of the period of analysis
Exposure	The direct and immediate impact of a hazardous event on the transportation asset, e.g., in the case of flooding it is water inundation on the roadway surface
Future year	The year that the travel demand model trip tables and network are projected for
Hazard event	A single instance of a hazard such as a flood or a wildfire
Hazard level	The severity of a hazard (e.g., a 10 year flood or 100 year flood)
Hazard recession period	The period over which the hazardous event subsides in days, e.g., in a flooding event, it is the period over which the flood waters recede; also called hazard recovery in the RDR Tool Suite
Hazard severity	The highest level of the hazardous event, e.g., in the case of flooding, the highest inundation level
Latin Hypercube Sampling	Method to select the set of core model runs that populate the regression model in the RDRM
Metamodel	Method used to estimate potential conditions of the network not feasible to be estimated in a standard travel demand model
Mitigation	Reduction in disruption under a given hazard due to a resilience investment
Out year	A future year for which analysis is executed

Project group	A set of resilience projects grouped together either for analysis purposes or to represent a relationship among the projects
Recapitalization	Occurs when the project would be redeployed during the period of analysis because the project's useful life is shorter than the period of analysis
Recovery stage	The distinct levels of hazard exposure that occur as the hazardous event subsides, e.g., in a flooding event the flood waters recede from 12 ft to 10 ft to 8 ft and so forth
Regression	A function that describes the relationship between two variables (e.g., hazard severity versus network performance)
Repair cost	The cost of repairing the asset damage to its original condition prior to the hazardous event
Repair recovery period	The period over which repairs to an asset are performed
Repair time	The time in days that it takes to repair the asset to its original condition prior to the hazardous event
Resilience investment	An additional or unique project that is intended to reduce the impact of a hazard on the transportation system (see also resilience project)
Resilience project	An additional or unique project that is intended to reduce the impact of a hazard on the transportation system (see also resilience investment)
Scenario space	The range of conditions over which the resilience investment performance is estimated
Start year	The first year of the period of analysis
Trip loss elasticity	Factor defining the reduction in the number of trips based on the increase in travel time
Uncertainty	Factors in the analysis for which future conditions are only predicted based on probability but are not certain to occur
Useful life	The number of years the investment will last or be operational before needing to be replaced

Appendix D: Defining Exposure Data

The process for estimating exposure of transportation assets associated with the hazard scenarios depends on the data available and consensus on probability and severity. There are four main sources or types of data that can contribute to asset hazard exposure assessments (additional information can be found in Section 3 of FHWA's Framework³²):

- Existing hazard exposure datasets and tools from federal or state agencies
- Existing scenario analyses
- Historical data
- Local/regional expert knowledge

Existing hazard exposure datasets and tools: There are a number of existing datasets from federal agencies that could be utilized in estimating exposure from a hazard scenario. The potential drawback to relying on national datasets is that they may not cover all areas of interest and/or the data may be too general for detailed analysis. For example, the Federal Emergency Management Agency's (FEMA) National Flood Hazard Layer³³ does not cover all U.S. counties, does not include detailed transportation infrastructure assets, nor does it provide inundation depths. In some cases, local agencies may have their own more detailed datasets regarding potential exposure to hazards that could be used to inform a hazard exposure analysis. [Appendix E: Transportation Asset and Hazard Scenario Data Summary](#) includes example datasets for flooding-related hazards. Some of the asset datasets contain detailed attributes that can help evaluate general susceptibility to flooding, and many of the available hazard layers provide detailed data on the potential impacts of various sea level rise and coastal flooding scenarios. However, the common limitations of the datasets analyzed include asset datasets that lack enough geospatial and attribute detail to identify specific levels of exposure and damage, and a lack of hazard datasets that provide flood inundation depth data for inland areas. Hazard datasets should be assessed for limitations when applying them to the analysis. Assets that might be initially characterized as exposed should be further evaluated to validate characteristics that may adjust the exposure estimation-- such as if the asset is elevated above grade-- so they are not improperly identified as vulnerable in the analysis.

FEMA's publicly available tool Hazards U.S. Multi-Hazard (Hazus-MH) focuses on national exposure analysis and damage estimation. It provides a standardized methodology that estimates potential losses from earthquakes, floods, and hurricanes. Hazus uses GIS to estimate and illustrate physical, economic, and social impacts of disasters, and is focused on census blocks as the unit of analysis (rather than individual assets). The tool generates hazard maps (GIS shapefiles), physical damage estimates, and loss estimates including losses due to disruption. Although Hazus does not calculate damage and losses to the transportation networks for events other than earthquakes, 30-meter resolution flood depth grids are available for all 50 states as an output from the tool. If no other data are available to conduct a flood analysis,³⁴ the depth grid dataset output from Hazus could provide the needed hazard input to conduct

³² FHWA Office of Planning, Environment, & Realty. Vulnerability Assessment and Adaptation Framework. Third Edition. FHWA-HEP-18-020.

https://www.fhwa.dot.gov/environment/sustainability/resilience/adaptation_framework/climate_adaptation.pdf

³³ FEMA. 2022. National Flood Hazard Layer [Online]. <https://www.fema.gov/national-flood-hazard-layer-nfhl>

³⁴ The Hazus depth grids can be used as an input into the RDR Exposure Analysis Tool.

the initial flood exposure analysis. Another national-level tool is USDOT's Coupled Model Intercomparison Project (CMIP) Climate Data Processing Tool and User Guide³⁵ that was developed to assist transportation professionals process downscaled climate projections into changes in the frequency of very hot days and extreme precipitation events that may affect transportation infrastructure and services. Additional climate hazard data can be found in the Climate Mapping for Resilience and Adaptation (CMRA) tool at a census tract or county level,³⁶ the NOAA Sea Level Rise Viewer,³⁷ and U.S. Geological Survey (USGS) Earthquake Hazard Program.³⁸

Existing scenario analyses: For certain types of hazards (e.g., sea level rise, downscaled temperature data), scenarios may already be established for a given government agency or by a given modeling or analysis study, and these may be adapted for the purposes of analyzing transportation impacts (e.g., NOAA's Coastal Inundation Dashboard³⁹). Some transportation agencies develop a GIS-based vulnerability analysis by collecting location-based GIS layers for hazards (e.g., extent of floodplain, sea level rise projections) and overlaying them with asset layers. The results of a GIS-based exposure analysis can be fed into a damage assessment and integrated into a TDM approach.

Historical data: Existing data sources either within an agency or in historical archives that track the occurrence of hazards relevant to a given region. For example, the USGS⁴⁰ has an array of water data, including logs of stream gauge data. State and local agencies sometimes produce geospatial datasets documenting the extent of a historical hazard scenario, such as estimated flood inundation areas for August 2016 flooding in East Baton Rouge Parish, Louisiana.⁴¹

Local/regional data and expert knowledge: In many cases, even if no formal historical datasets exist, there may be expert and/or local knowledge of historical events that can inform assumptions about the probability and magnitude of potential future hazards. Such information can be used primarily in a binary analysis to inform an assumption about whether an asset is within or outside the exposure area.

Identify Relevant Individual Asset Data

Resilience investments are not needed for a given asset if the asset is not vulnerable to the hazard(s) of concern. In addition, it is not tractable to test every asset on the transportation network to determine if a resilience investment is worthwhile. Therefore, a screening approach is required to identify assets that

³⁵ FHWA. 2021. Sustainability: Tools/Climate Change Adaptation. Accessed 2 May 2024 from: <https://www.fhwa.dot.gov/environment/sustainability/resilience/tools/>

³⁶ NOAA. 2024. Accessed 2 May 2024 from: <https://livingatlas.arcgis.com/assessment-tool/home/>

³⁷ NOAA. 2024. Accessed 2 May 2024 from: <https://coast.noaa.gov/slr/>

³⁸ USGS. 2024. Accessed 2 May 2024 from: <https://earthquake.usgs.gov/earthquakes/search/>

³⁹ NOAA. 2022. Coastal Inundation Dashboard. Accessed 2 May 2024 from: <https://tidesandcurrents.noaa.gov/inundationdb/>

⁴⁰ USGS. 2022. Water Resources. Accessed 2 May 2024 from: <https://www.usgs.gov/mission-areas/water-resources>.

⁴¹ USGS. 2019. August 2016 Louisiana Floods. <https://www.usgs.gov/mission-areas/water-resources/science/august-2016-louisiana-floods>

may be considered for resilience investment, such that the impact of the hazard scenario(s) and the potential mitigation of that impact with a resilience investment can be assessed.

The selection of assets of interest for evaluating resilience ROI can be informed by expert knowledge as well as specific analyses (additional information can be found in Section 2 of FHWA's Framework⁴²).

Expert selection/knowledge: In many cases, local practitioners and other stakeholders may have knowledge of historical events, repair and maintenance of assets, or other data that can be leveraged to assess which assets have been affected by a hazard in the past and therefore may be vulnerable to future hazards. A stakeholder-based vulnerability assessment can include collecting expert information and institutional knowledge to rate the vulnerability of transportation assets. For example, maintenance and operations staff may be familiar with which roads are likely to flood under different conditions, and local residents may have an understanding of which roads provide critical access to community services. Using a workshop-based format to collect this type of information from staff and stakeholders and to discuss assets' expected performance under different hazard scenarios can help an agency identify at-risk assets and priorities for further analysis. This type of qualitative assessment would need to be translated into a format that can be used for a TDM-based damage analysis. Input from stakeholders could, for instance, drive configuration of the TDM network. Stakeholder engagement may also be leveraged to improve individual asset-level data.⁴³

Existing vulnerability and risk assessments: Some agencies have performed vulnerability or asset risk assessments for anticipated hazards using their own tools or existing resources that take into account expert knowledge and historical information to rank and prioritize assets with high vulnerability (e.g., exposure, sensitivity, or adaptive capacity) to a set of predetermined hazards. One tool for conducting a vulnerability assessment is FHWA's Vulnerability Assessment Scoring Tool (VAST). VAST is a comprehensive spreadsheet-based tool that guides transportation professionals through a quantitative, indicator-based vulnerability screening process. The output of the tool is a vulnerability score (1-5).

⁴² FHWA Office of Planning, Environment, & Realty. Vulnerability Assessment and Adaptation Framework. Third Edition. FHWA-HEP-18-020.

https://www.fhwa.dot.gov/environment/sustainability/resilience/adaptation_framework/climate_adaptation.pdf

⁴³ For example, [the Maryland State Highway Administration \(MDSHA\)](https://www.fhwa.dot.gov/environment/sustainability/resilience/pilots/2013-2015_pilots/maryland/final_report/mdpilot.pdf) was having difficulty assessing the vulnerability of its transportation assets to climate variables and stressors due to limited data availability until they begin collecting the needed data by conducting workshops and interviews with maintenance staff.

https://www.fhwa.dot.gov/environment/sustainability/resilience/pilots/2013-2015_pilots/maryland/final_report/mdpilot.pdf

Planned and proposed transportation projects: In some cases, a planned or proposed project can be enhanced with an added marginal cost to make the asset more resilient. Thus, one screening approach is to assume that resilience investments will occur in concert with other planned or proposed projects, and these assets can then be prioritized for resilience ROI analysis.

Criticality analysis: Existing designations of assets related to criticality may inform the practitioner of specific critical use assets that should be considered for resilience investments. Such critical uses may include designated evacuation routes, the Strategic Highway Network (STRAHNET), key access points for known isolated or economically critical facilities (e.g., ports), or results of past criticality analyses (e.g., DHS analysis⁴⁴).

Network nodal/criticality analysis for key nodes: There are existing methodologies⁴⁵ for assessing the criticality of nodes in a network; however, these tend to be focused on connectivity by shortest paths and may not take into account rerouting or flow volume as factors in determining criticality.

Once the relevant asset data are identified, they must be incorporated into, or at least compatible with, the transportation network of interest, such as a TDM.

Agencies can use the scores from VAST to identify their most vulnerable assets. Indicator-based vulnerability assessments, like VAST, provide relative information about the most vulnerable assets, but do not give information about the extent of the vulnerability or potential damage to these assets. While the numerical scores from an indicator-based assessment may not be able to directly inform a damage/disruption analysis, some of the data collected for the vulnerability assessment may prove useful (e.g., data on exposure to flooding such as elevation or location in a floodplain). An indicator-based assessment can also be used as a screen to select assets for inclusion in a more detailed damage analysis.

⁴⁴ DHS/USDOT. 2015. Transportation Systems Sector-Specific Plan.

<https://www.cisa.gov/sites/default/files/publications/nipp-ssp-transportation-systems-2015-508.pdf>

⁴⁵ e.g., Borgatti, Stephen P. 2006. Identifying sets of key players in a social network. *Comput. Math Organiz. Theor.* 12: 21-34 (DOI 10.1007/s10588-006-7084-x); Hillier B., Hanson J. 1984. *The social logic of space*. Cambridge University Press.

Appendix E: Transportation Asset and Hazard Scenario Data Summary

The table below lists datasets, covering both road networks and flooding, that could supply needed data for the RDR Tool Suite, with their strengths and limitations.

Table E-0-1: Transportation asset data resources for scenario development.

Dataset	Source	Data Type (Transportation Asset, Hazard Data)	Strengths	Limitations
Travel demand model road networks	State DOT or regional MPO	Transportation Asset— Road network data exportable as GIS data	Represents best knowledge of state/regional transportation assets, travel demand, asset vehicle capacity and travel speed, and may include planned /proposed asset investment projects.	No mechanism to incorporate hazard data and impacts on assets, nor mitigation of hazard impacts. Attributes relevant to hazard (e.g., elevation, scour vulnerability) not included, requiring incorporation of data from other sources.
National Bridge Inventory	FHWA	Transportation Asset— Database with location information and bridge attributes	National level bridge inventory with rich attribute data: waterway adequacy, scour critical, bridge condition.	Geography not directly linked to road network (single point rather than representing full bridge span), no elevation data. Does not cover bridges less than 20 feet in length.
National Tunnel Inventory	FHWA	Transportation Asset— Database with location information and tunnel attributes	National level tunnel inventory.	Poor geospatial representation (single point rather than representing full tunnel span), few attributes relevant to hazard scenarios (other than detour length).
Highway Performance Monitoring System (HPMS) and All Roads	FHWA	Transportation Asset— GIS	Detailed attributes including surface type AADT, pavement condition, climate zone, soil type,	Not flowable, no elevation data.

Dataset	Source	Data Type (Transportation Asset, Hazard Data)	Strengths	Limitations
Network Of Linear referenced Data (ARNOLD)		Road Network and Attributes	and terrain type that could be useful to help estimate high-level flood risk and impacts. Can be easily overlain on top of hazard data.	
Freight Analysis Framework (FAF)	FHWA	Transportation Asset— GIS Road Network and Attributes	Flowable road network.	Only covers major freight corridors, not local roads.
Freight and Fuel Transportation Optimization Tool (FTOT) multimodal, flowable transportation network	Volpe	Transportation Asset— Flowable multimodal transportation network	Flowable multimodal dataset and optimization module provide ability to assess network disruptions and rerouting.	Does not include local roads, no attributes related to hazards.
Coastal Inundation Dashboard and Sea Level Rise, Sea Level Rise Depth, Mapping Confidence, DEM, and Flood Frequency Data	NOAA	Hazard Data— flooding scenario datasets	Detailed data for coastal areas.	Does not cover inland flooding.
Global Ensemble Forecast System (GEFS) and NOAA Operational Model and Archive Distribution System (NOMADS)	NOAA	Hazard Data— weather forecast model	May be useful at local scales to determine the probability that a set of conditions will occur.	Not useful for regional/national analysis.
National Flood Hazard Layer	FEMA	Hazard Data— flooding	Determines flood zones, base flood elevation and floodway status across large portions of the U.S.	Does not cover all counties, does not provide inundation depth associated with various flooding scenarios. Flood zones may not be relevant for transportation assets that are elevated (e.g., bridges, causeways).
FLDPLN	University of Kansas	Hazard Data— inland flooding	Outputs from this tool could be overlain on transportation assets	Not currently publicly available.

Dataset	Source	Data Type (Transportation Asset, Hazard Data)	Strengths	Limitations
			to quantify exposure to inland flooding.	
Current Water Data for the Nation	USGS	Hazard Data— inland flooding	Real-time, detailed flood information at flood gauge sites across the country.	Does not provide enough information to determine flooding impact on nearby transportation assets.
Flood Inundation Mapper	USGS	Hazard Data— inland flooding	Can help provide detailed information on the extent and depth of inland flooding.	Limited in scope to relatively small geographic areas immediately surrounding a small subset of river gauges throughout the country.
Earthquake Catalog	USGS	Hazard Data— Earthquakes	Comprehensive catalog of historical earthquakes with detailed spatial representations	Assignment of historical earthquake scenarios to transportation network, and assessment of damage probability, requires additional GIS preparation
Hydraulic Engineering Circular No. 20 (HEC 20)	USACE	Hazard Data— flooding/stream stability	Details procedure for evaluation of stream stability problems.	No ready-made data for helping assess bridge vulnerability.

Appendix F: Baseline Network Run Helper Tool

The baseline network helper tool enables the user to run routing on the network and trip tables, without any disruption scenarios. It is intended to facilitate base model validation. Primary outputs include a set of link flows and a set of origin-destination flows and trip times (or skims).

Baseline Network Helper Tool Required Inputs

The following input files are required:

1. Trip table(s) OMX file (in AEMaster/matrices sub-directory) - "baseline_run_demand_summed.omx". This OMX file contains one or two matrices. The first matrix, named 'matrix' provides the daily average trip table for the base year. If the RDR scenario separates out households with access to vehicles and households without access, the OMX file will include a second trip table called "nocar".
2. Network links CSV file (in Networks sub-directory) - "baseline_run.csv". This is comprised of all centroid connectors, roadway segments, and optionally, transit links in the base year network.
3. Network nodes CSV file (in Networks sub-directory) - "node.csv". This is comprised of all centroids, roadway nodes, and optionally, transit nodes in the base year network.

The following file is optional:

1. True shapes CSV file (in LookupTables sub-directory) - "TrueShape.csv"

Running the Tool

To run the baseline network helper tool, ensure that all files listed in 1-3 above are fully prepared and saved in the designated locations. The "baseline_network_run.bat" file (located within RDR/helper_tools/baseline_network_run) must then be edited to point to the following three files:

1. The Python executable associated with the user's RDR Python environment (RDREnv)
2. The location of the "baseline_network_run.py" script on the user's machine
3. The location of the scenario's configuration file on the user's machine

The user can then open the Windows Command Prompt interface or Windows PowerShell. For instance, for the Command Prompt, go to the Windows Start Menu, type Command Prompt in the search bar, click on the Command Prompt icon, and a new Command Prompt will open. Then drag the bat file into the shell, and press enter to run (the user may need to press enter more than once).

Outputs include the following files:

- link_flow_full.csv – Combines the link file (Group_baserun.csv) and the output link flows (link_flow_baseyear.csv).
- node.json and link_flow_full.json – Geospatial data files containing the same information as link_flow_full.csv, which can be imported into GIS software.
- matrices/rt_baseyear.omx – Skim from running routing.
- matrices/sp_baseyear.omx – Skim from running shortest path.

Visualizing outputs

The `link_flow_full.csv` file contains all of the information needed to illustrate link flows and volume capacity ratios.

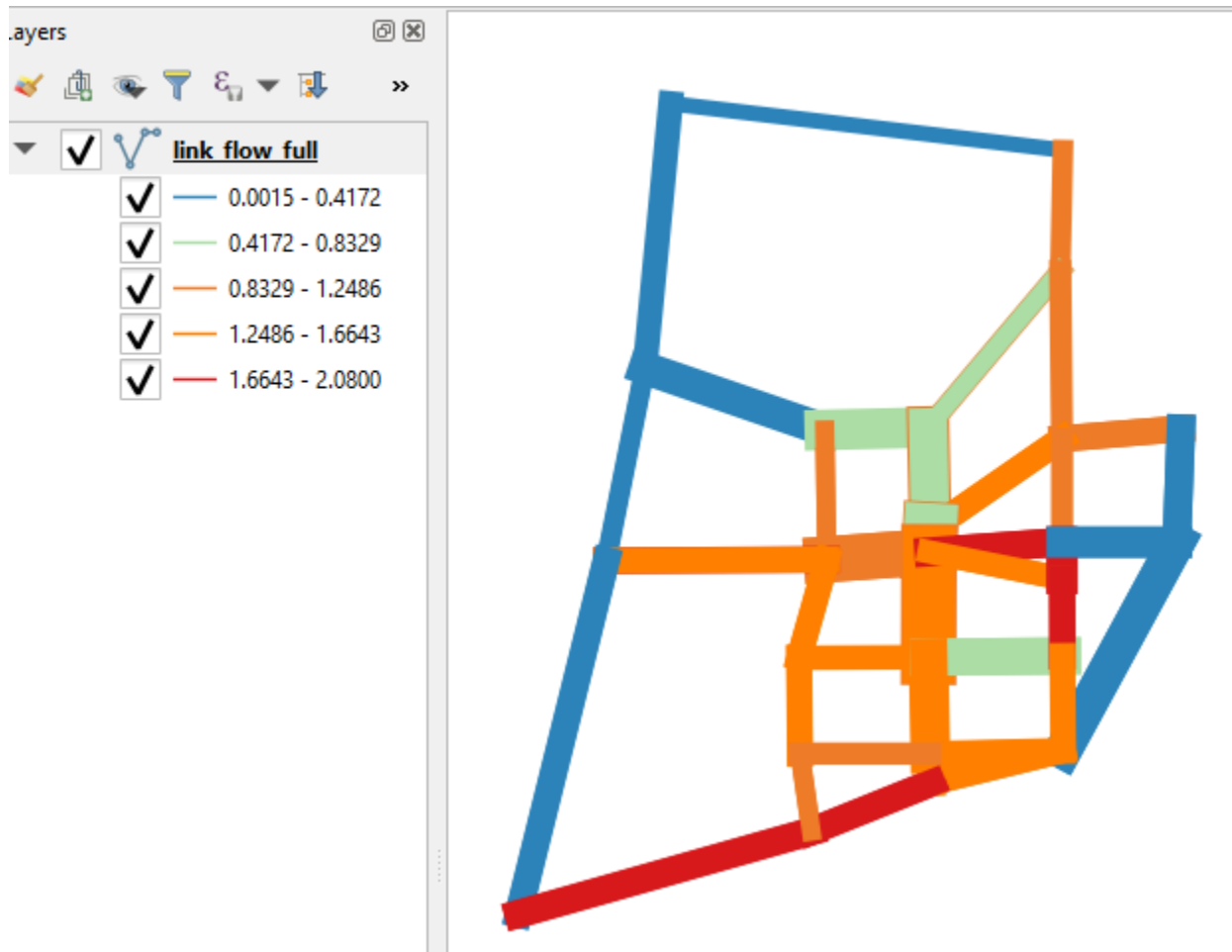


Figure F-0-1 Link flows (line thickness) and volume-capacity (line color)

Steps in QGIS for producing the above figure are not part of the helper tool, but are included below:

1. Add layer from `link_flow_full.csv`.
2. Go to layer properties symbology. Set up the appropriate Graduated color ramp for the `vcr` field.
3. Then set up an override field on symbol width, by clicking on the button to the right of the symbol width box, then using the assistant. See Figure F-0-2.

Note: The user can do the same thing with triangle markers (rather than simple lines) to get directionality.

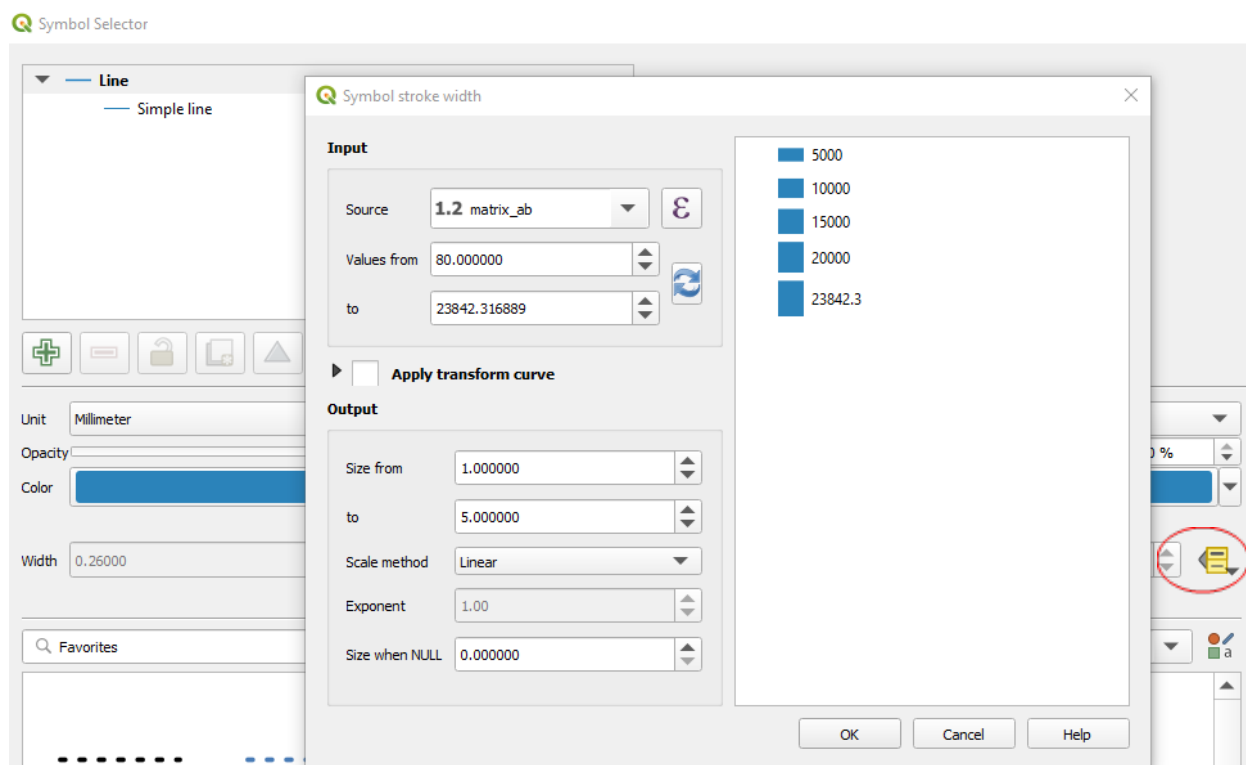


Figure F-0-2 Adding symbol stroke width in QGIS

Appendix G: Base Year Run Helper Tool

The base year run helper tool is designed to automate the production of base year AequilibraE core model runs ahead of a full run of the RDR Metamodel. Base year core model runs are required as input to the ROI Analysis Tool so that they can be interpolated with the future year core model runs to provide metrics across the entire ROI performance period. The base year run helper tool leverages the same functionality of the core tool suite but allows the base year run outputs to be prepared separately in a quick and efficient standalone tool. The tool automatically runs base year AequilibraE model runs for every combination of hazard and recovery parameters supplied by the user. All AequilibraE base year model outputs are consolidated into one comma-separated (CSV) file.

Base Year Run Helper Tool Required Inputs

The base year run helper tool leverages the same configuration file used for a full RDR Metamodel run. Specifically, the configuration parameters that must be defined include input directory, output directory, AequilibraE Model Run Type, and Mini-Equilibrium Run. The following list of input files (stored within the input directory) are leveraged by the base year run helper tool and must be prepared and stored in the correct locations prior to running the tool. Users can consult Section 6.1.1 for more information on each of these configuration parameters, along with the standalone Run Checklist.

The following input files are required:

1. Model parameters XLSX file (in main input directory) - "Model_Parameters.xlsx". This is where each hazard and recovery parameter utilized in a scenario is defined.
2. Trip table(s) OMX file (in AEMaster/matrices sub-directory) - "baseyear_demand_summed.omx". This provides the daily average trip table for the base year in a trip table called "matrix". If the RDR scenario separates out households with access to vehicles and households without access, a second trip table called "nocar" should also be included.
3. SQLite database (in AEMaster sub-directory) - "project_database.sqlite". This provides the network structure and disruption information to AequilibraE.
4. Exposure CSV file(s) (in Hazards sub-directory) - one for each hazard, "{HAZARD_RUN_NAME}.csv"
5. Network links CSV file (in Networks sub-directory) - "baseyear.csv". This is comprised of all centroid connectors, roadway segments, and optionally, transit links in the base year network.
6. Network nodes CSV file (in Networks sub-directory) - "node.csv"

The following parameters are optional:

7. True shapes CSV file (in LookupTables sub-directory) - "TrueShape.csv"

Running the Tool

To run the base year helper tool, ensure that all files listed in 1-6 above are fully prepared and saved in the designated locations. Once the relevant parameters within the configuration file are prepared, the "base_year_run.bat" file (located within RDR/helper_tools/base_year_run) must be edited to point to the following three files:

1. The Python executable associated with the user's RDR Python environment (RDREnv)
2. The location of the "base_year_run.py" script on the user's machine

3. The location of the scenario's configuration file on the user's machine

The user can then open the Windows Command Prompt interface or Windows PowerShell. For instance, for the Command Prompt, go to the Windows Start Menu, type Command Prompt in the search bar, click on the Command Prompt icon, and a new Command Prompt will open. Then drag the bat file into the shell, and press enter to run (the user may need to press enter more than once).

The output of a run is either "Metamodel_scenarios_SP_baseyear.csv" if the AequilibraE run type is set to 'SP' or "Metamodel_scenarios_RT_baseyear.csv" if the AequilibraE run type is set to 'RT'.

Appendix H: Format Demand Helper Tools

RDR is designed to read trip tables that are in OpenMatrix format⁴⁶ (or OMX). OMX is a publicly available, open-source format designed for efficiently storing large, sparse matrices. Many existing travel demand models can export matrices in OMX format.

Furthermore, the origin-destination travel times (or skims) that come out of RDR are also stored in OMX format.

The Format Demand Helper Tools consist of three small Python programs:

- `convert_trip_table.py` – converts plain text (CSV) demand file to OMX format,
- `review_trip_table.py` – reads a trip table in OMX format and provides a summary,
- `review_skim.py` – reads the skims in OMX format and provides a summary.

Convert Trip Table

- Inputs:
 - CSV node file (e.g., `GMNS_node.csv`)
 - CSV trip table(s)
 - E.g., `demand.csv`
 - Optionally, `demand_nocar.csv`
- Output: OMX trip table (`demand_new.omx`) with TAZ mapping and demand matrices

The node file must have at least two fields, labeled `node_id` and `node_type`. It may have additional fields. The `node_id` is the unique integer identifying the node. The `node_type` is a text field with the value 'centroid' for centroid nodes. The program first reads the node file, creating a dictionary of the centroid nodes, called `taz_list`. For example, if there are three centroid nodes, with `node_id` = 1, 5, and 10, the `taz_list` would be { 1:0, 5:1, 10:2 }.

The program then reads the trip table (e.g., `demand.csv`). The user can specify the trip table in two ways: long or square. In the long format, the user provides origin-destination trips row by row in a CSV file that has three fields, labeled `orig_node`, `dest_node`, `trips`. `Orig_node` and `dest_node` must be centroids in the node file. In the square format, the user provides origin-destination trips in a square matrix in a CSV file, with row names indicating `orig_node` and column names indicating `dest_node`.

Based on these two inputs, the program outputs the trips as an OMX file:

- The `taz_list` mapping is named 'taz',
- The trip matrix itself is named 'matrix' (OMX files can store more than one matrix).

Optionally, the program may also read a second trip table (`demand_nocar.csv`), which has the same three fields as `demand.csv`. Based on the second trip table, the program adds a second matrix to the OMX file, named 'nocar'. All input parameters are defined at the top of the Python script.

Review Trip Table

- Inputs: OMX trip table (e.g., `demand_new.omx`)

⁴⁶ See [GitHub - osPlanning/omx-python: Python API for OMX](https://github.com/osPlanning/omx-python)

- Outputs: summary information on the trip table

The program reads origin-destination flows from the trip table matrix specified and prints out information about the OMX file (e.g., list of matrices) as well as the trip table (e.g., percentiles of matrix values). All input parameters are defined at the top of the Python script.

Review Skim

- Inputs: OMX trip table (e.g., demand_new.omx) and OMX skim table (e.g., skims.omx)
- Outputs: summary information on the skims

The program reads origin-destination flows from the trip table, and origin-destination travel times and distances from the skim table. It then calculates total trips (summing the trip table), total travel time (sum of trips multiplied by trip travel times) and total distance (sum of trips multiplied by trip distance). All input parameters are defined at the top of the Python script.

Appendix I: Format Network Helper Tools

The helper tools located in the “format_network” subfolder of the RDR GitHub repository, as well as the separate GTFS2GMNS repository, aid the user in building a transit network compatible for use with the RDR Tool Suite. The helper tools guide the user through three steps:

- GTFS2GMNS repository: Creating a node-link transit network following the General Network Modeling Specification (GMNS) based on a GTFS network.
- prepare_rdr_transit_network.py script: Creating transit centroid connectors, checking compatibility of road and transit networks, merging network files, and (optional) creating transit line shapes.
- calculate_transit_network_metrics.py script: Calculating toll and travel time on the network.

Note that these helper tools rely on the use of RDR’s default facility type labels (see Table 3-3).

GTFS2GMNS

The de-facto standard for representing transit networks is the General Transit Feed Specification (GTFS).⁴⁷ It is a human and machine-readable representation of transit routes, stops and schedules. Many transit agencies have their routes, stops and schedules available in GTFS format.

GTFS2GMNS is a program, developed at Arizona State University, that does basic conversion of GTFS files to GMNS node-link files. The RDR team has created a fork of the GitHub repository for use with the helper tools: <https://github.com/VolpeUSDOT/GTFS2GMNS>

Documentation for GTFS2GMNS is located at: <https://github.com/asu-trans-ai-lab/GTFS2GMNS/blob/main/doc/GTFS2GMNS%20Users'%20Guide.pdf>.

Input GTFS files include:

- Agency (the transit agency)
- Stops (list of transit stops, include latitude and longitude)
- Routes (list of transit routes)
- Trips (list of transit trips)
- Stop_times (list joining trips, stops and stop times)

The program creates service links for the transit routes and generates transferring links between routes. It also provides boarding links that connect the transferring links and service links, using approach 1. The links provide straight-line connections between the stops on the route. Figure I-0-1 shows a small piece of the Hampton Roads transit network. The green line is the light-rail (tram), the purple lines are bus routes, the red lines are transfer links, and the highlighted yellow line is bus routes 009.1 and 009.2 (both directions of route 9). Table 0-1 lists selected fields in the transit links.

More detailed shape information is available in GTFS (shapes.txt), but the GTFS2GMNS program, in its current form, does not use it. The Prepare RDR Transit Network tool can use the shapes.txt file from GTFS to construct a geodatabase of transit service links for use in the RDR Exposure Analysis Tool.

⁴⁷ General Transit Feed Specification, <https://gtfs.org/>.

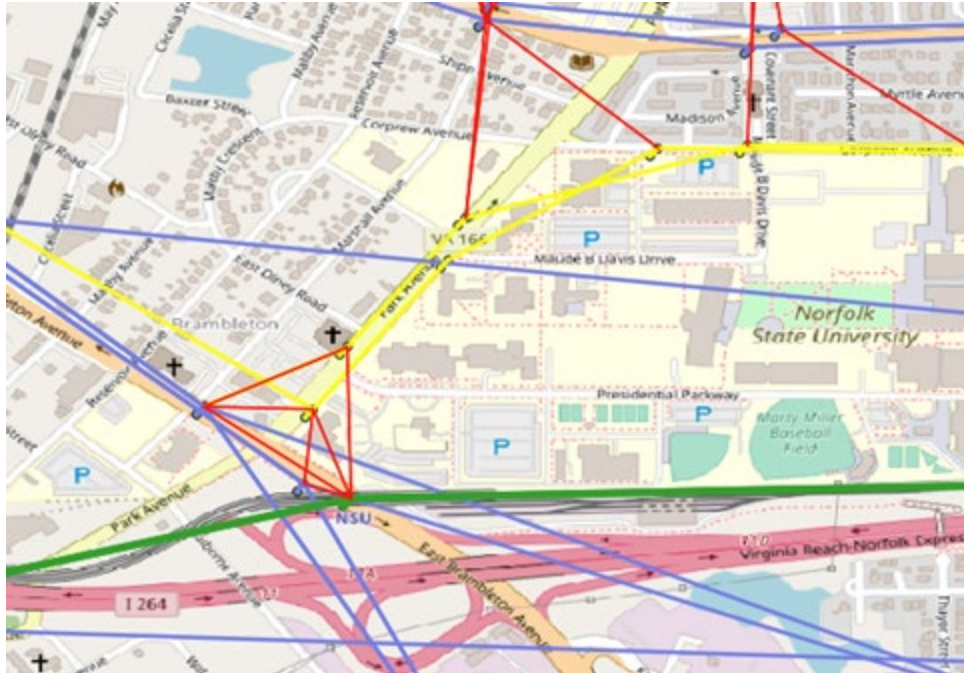


Figure I-0-1 Example transit links from GTFS mapped on a streets basemap

Table I-0-1 Selected transit link fields

Field	Example on route	Example on boarding link	Example on transfer link	Comment
link_id	1000822	1006596	1699	
from_node_id	1500846	1000360	1002614	
to_node_id	1500847	1500846	1000360	
facility_type	Bus	Bus	Sta2sta	
directed_route_id	009.1	009.1	-1	
link_type_name	service_links	boarding_links	transferring_links	
length	271	14	139	Meters
lanes	6	1	1	
capacity	999999	999999	999999	
free_speed	16	2	1	Mph
cost	0	0	60	
VDF_fftt1	1	5	0.14	Used for wait time on boarding links
VDF_allowed_uses1	w_bus_only...	w_bus_only...	w_bus_only...	

The links generated by GTFS2GMNS are straight lines between the transit stops. For local routes, with frequent stops, these are a good approximation of where the route actually travels in the real world. For express routes, with infrequent stops, the transit links might not follow the roads that they use.

There are two methods for running the GTFS2GMNS code—a Jupyter notebook called `gtfs2gmns.ipynb` or a Python code file called `gtfs2gmns.py` located in the “src” subfolder. In either case, the user should modify the `input_path` to point to the location of their GTFS text files and modify the `output_path` to point to where they would like output GMNS files to be generated. Once that has been completed, the user should run the `gtfs2gmns` method using either the Jupyter notebook or the Python code (using their standard Python executable).

ArcGIS and open source GIS alternatives provide functionality that processes the actual route geometry (`shapes.txt`) portion of a GTFS feed. This can be useful when the precise location of a route is useful—such as for exposure or disruption analysis. More information on Esri’s ArcGIS Pro tool is available here: <https://pro.arcgis.com/en/pro-app/latest/tool-reference/conversion/gtfs-shapes-to-features.htm>. The free software QGIS may also have a suitable plugin for converting GTFS `shapes.txt` file into a shapefile or geodatabase feature class: <https://ctu-fgis.github.io/2020-b-qgis-gtfs-plugin/>. Portions of Esri’s GTFS functionality have been incorporated into the Transit Network Preparation Tool to generate a GIS-compatible version of an RDR transit network (to allow for exposure and disruption analysis).

Outputs:

- Node and link CSV files of the GTFS transit feed in GMNS format.

Transit Network Preparation Tool (`prepare_rdr_transit_network.py`)

Two primary inputs are needed to prepare the transit network:

1. A point layer of transit stops (these are the “stop” nodes generated in the “**Error! Reference source not found.**” step).
2. A polygon layer of transportation analysis zones (TAZs).

In addition, the user needs to provide a path to the node and link CSV files in GMNS format, which can come from the GTFS2GMNS tool. The node and link CSV files of the road network are also needed; in particular, the road node CSV file should contain all centroid nodes that will be used to construct transit centroid connectors. Once these inputs have been specified in the Format Network configuration file, the user can run the tool using the batch file `prepare_rdr_transit_network.bat`.

The RDR Transit Network Preparation Tool has an ArcGIS dependency. Similar steps to prepare the transit network, using QGIS, are as follows:

1. If the coordinate reference system (CRS) for the transit stops is in degrees, project it into a CRS that is in feet or meters
 - a. In QGIS, use Reproject Layer in the Vector General toolbox
2. Set up a ¼ mile buffer (5+ minute walk around the transit stops)
 - a. In QGIS, use Buffer in the Vector Geometry toolbox (see Figure 0-2)

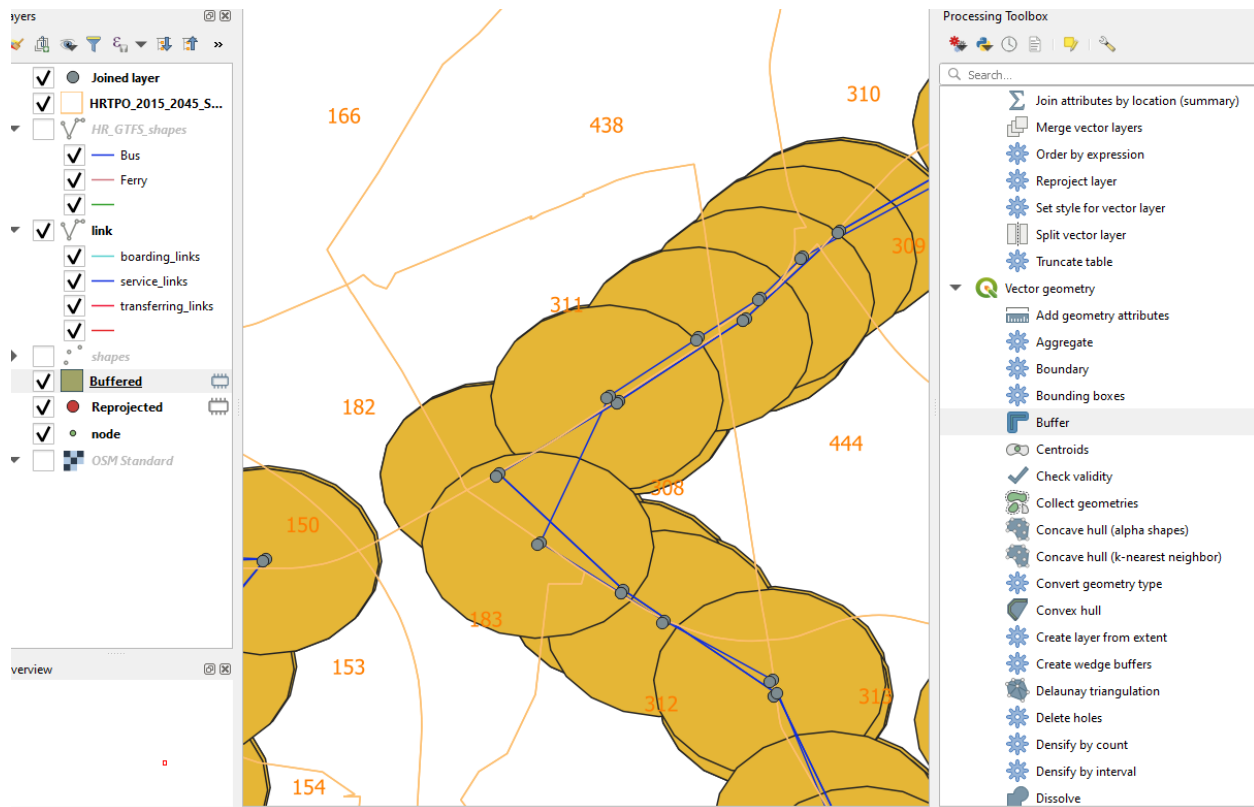


Figure I-0-2 Buffer around transit stops

3. Join the buffered stops and the TAZ layer. The process joins two polygon layers, generating a list of the intersections of the polygons. Each stop will typically be joined with several zones.

Work with the results of the join to create centroid connectors, running between the zone centroids and the transit stops.

A subcomponent of the Prepare RDR Transit Network tool is the ability to generate a geographically accurate version of the transit network (e.g., not just stop-stop information, but precise route geometry based on the shapes.txt component of GTFS). Because this functionality is not built into GTFS2GMNS, Esri ArcGIS Pro functionality is partially leveraged and a license to this software is required to run this tool. The output is a version of the RDR transit network that includes actual link geometry for every service link in the RDR transit network (note that centroid connectors, boarding links and other links that do not represent the actual path of transit service are not included in this output). Link IDs match the same link IDs generated by the GTFS2GMNS tool. The GIS output of this tool can be fed into the RDR Exposure Analysis tool to generate an exposure analysis for incorporation into the RDR Metamodel.

Calculate Transit Network Metrics (`calculate_transit_network_metrics.py`)

Although a detailed mode choice model is not in the scope of RDR, RDR does distinguish between travelers with and without access to an automobile. This information should be available from the MPO demand inputs. (For example, the production-attraction person trip tables at HRTPO distinguish between 0-car and 1+ car households). Zero car households may access the road network by using taxi or transportation network companies (TNC). Thus, a key difference between 0-car and 1+ car

households is that the cost and travel_time (which includes wait time) to use the road network is much higher for the 0-car households. This leads them to make more use of transit.

To run the transit network metrics tool, the user will need to set several parameters in the Format Network configuration file related to transit and TNC cost and wait time, as well as specify the locations of the node and link CSV files for which to calculate the implied toll and travel time. The outputs of the tool are network files compatible with the RDR Tool Suite. The Python script is run via the calculate_transit_network_metrics.bat file.

Note that the original model does not explicitly consider automobile operating costs (marginal operating cost is about 15 cents per mile, a fairly low value). Table I-0-2 shows the link costs and travel times. There are two sets of costs, for 1+car and 0-car households, labeled cost and cost_nocar. Similarly, there are two sets of travel times, time and time_nocar.

Table I-0-2 Link costs and travel times

Link type	cost (cents)	cost_nocar (cents)	time (minutes)	time_nocar (minutes)
Road centroid connector from origin ⁴⁸	0	TNC initial charge plus cost per mile	travel time ⁴⁹	TNC wait time plus travel time
Road	toll (if any)	TNC cost per mile	travel time	travel time
Road centroid connector to destination	parking fee (if any)	TNC cost per mile	travel time	travel time
Transit centroid connector from origin (walk access)	0	0	perceived ⁵⁰ travel time (at walking speed)	perceived travel time (at walking speed)
Transit centroid connector from origin (drive access)	parking fee (if any)	TNC initial charge plus cost per mile	travel time	TNC wait time plus travel time
Transit boarding link	transit fare	transit fare	perceived transit wait time	perceived transit wait time
Transit service link	per-mile component of fare (if any)	per-mile component of fare (if any)	transit in-vehicle time	transit in-vehicle time

⁴⁸ To discourage flows via the centroid connectors, speeds on centroid connector links should be set to a low value, 10 mph or less.

⁴⁹ Travel time is 60 (minutes / hour) * distance (miles) / speed (miles / hour)

⁵⁰ "Perceived" times (walking, waiting, transferring) are considered to be more onerous (have a higher value) than other travel times. US DOT cost benefit guidance gives these times a value of \$34 / hour, 1.8 times as high as the general value of \$18.80 per hour. The AequilibraE network model uses a value_of_time of \$18.80 per hour (set in the .config file). To account for this difference, the perceived time should be set to 1.8 times the actual expected walking/waiting/transfer time. For example, for transit with a headway of 10 minutes, the expected wait time is 5 minutes. The perceived wait time should be set to 9 minutes (1.8 x 5). (Reference: US DOT Benefit-Cost Analysis Guidance for Discretionary Grant Programs, January 2023)

Link type	cost (cents)	cost_nocar (cents)	time (minutes)	time_nocar (minutes)
Transit transfer link (approach 1)	0	0	perceived transfer walking time	perceived transfer walking time
Transit transfer link (approach 2)	Transfer fare (if any)	Transfer fare (if any)	perceived transfer plus transit wait time	perceived transfer plus transit wait time
Transit de-boarding link	0	0	0	0
Transit centroid connector to destination (walk access)	0	0	perceived travel time (at walking speed)	perceived travel time (at walking speed)
Transit centroid connector to destination (drive access)	0	TNC initial charge plus cost per mile	travel time	TNC wait time plus travel time

To summarize, for transit riders:

- The initial walk time is on the centroid connector to transit (walk access)
- The centroid connector to transit (drive access) includes drive time
 - For 1+ car households, it also includes parking cost at the transit station
 - For 0 car households, it includes TNC wait time and TNC cost
- The wait time and fare are on the transit boarding link. This assumes a fixed fare for all trips. Fares that vary by distance can be accommodated by including some portion of the fare on each transit service link.
- The transit in-vehicle time is on the transit service link
- The transit transfer link includes transfer time for the next vehicle. It could also include a transfer fee.
- The centroid connector from transit to centroid (walk access) has the destination walk time
- The centroid connector from transit to centroid (drive access) includes drive time
 - For 1+ car households, it also includes parking cost (most likely zero for a traveler returning home)
 - For 0 car households, it includes TNC wait time and TNC cost

To summarize, for persons using the highway links (either by driving or using TNC/taxi):

- Centroid connectors include drive time and parking cost. For 0 car households, they include drive time, TNC wait time, and initial TNC fare.
- Road links include drive time and tolls. For 0 car households they also include the per mile TNC fare (approximately \$2.50 / mile).