Journal of Data Science 24 (2), 373–393
April 2026
DOI: 10.6339/26-JDS1228
Data Science in Action

Designing Accessible and Dependable Tools for Vocational Rehabilitation Data Analysis

Ruth Taylor[1,*], Brennan Bean[1], Brian Phillips[2], and Allison Fleming[3]

[1]*Utah State University, Department of Mathematics and Statistics, Logan, UT, USA*
[2]*Utah State University, Department of Special Education and Rehabilitation Counseling, Logan, UT, USA*
[3]*Pennsylvania State University, Department of Educational Psychology, Counseling, and Special Education, University Park, PA, USA*

Abstract

The U.S. Rehabilitation Services Administration (RSA) has partnered with state vocational rehabilitation (VR) agencies since 1973 to improve employment outcomes for individuals with disabilities. A critical resource in this effort is the RSA-911 dataset, a quarterly collection of standardized participant data. However, its complex structure, including high rates of missing or ambiguous values, poses significant challenges for effective analysis. We address these challenges by developing an `R` package designed to streamline the cleaning and analysis of RSA-911 data, as well as the newly introduced Transition Readiness Toolkit (TRT) scores data (R Core Team, 2021). The TRT assesses participants' improvement across services and offers a critical measure of VR program effectiveness. Using this `R` package, our work offers the first analysis of the relationship between TRT pre-post scores and RSA-911 demographic data, providing insights into program outcomes. Additionally, we deliver a user-friendly online dashboard, built with the `shiny` framework, to allow VR counselors and researchers to independently analyze RSA-911 and TRT data (Chang et al., 2024). This dashboard features intuitive visualizations and workflows, making it easier to generate reproducible analyses without requiring extensive technical expertise. By automating data preparation and providing accessible analysis tools, this project contributes to the field of vocational rehabilitation by facilitating more efficient research and empowering VR professionals with data-driven insights. The tools presented offer a framework for future studies, enhancing the consistency, flexibility, and reproducibility of VR data analysis.

Keywords *data dashboard; data exploration; messy data; R package; shiny*

1 Introduction

Since its establishment in 1973, the U.S. Rehabilitation Services Administration (RSA) aims to expand employment opportunities and promote independence for individuals with disabilities (Commission, 1973). To achieve this, the RSA partners with state Vocational Rehabilitation (VR) agencies to enhance employment readiness, career exploration, higher education opportunities, and earning potential. These agencies collect standardized administrative data, known as "RSA-911" datasets, which have been gathered quarterly since 1992 (RSA, 2024a). The datasets, comprising hundreds of variables, detail participant demographics, services received, and em-

*Corresponding author. Email: ruth.taylor@usu.edu.

Received August 16, 2025; Accepted March 21, 2026

ployment and education outcomes (RSA, 2004). Despite their value in VR research, raw RSA-911 datasets are not well-suited for traditional analysis or visualization.

The RSA-911 dataset format poses significant challenges due to extensive missing values, inconsistent representations of missingness, and the complexity of its documentation. While many null entries represent baseline values, understanding them requires navigating a dense codebook with instructions for 407 variables (RSA, 2004). These issues are compounded by decentralized data entry from individual VR counselors across the state, increasing the risk of inconsistencies and human error. Preparing RSA-911 data is time-intensive and error-prone, highlighting the need for efficient and reproducible cleaning processes.

In conventional RSA-911 data, there are no variables that directly measure the efficacy of services offered. To remedy this, VR researchers developed the Transition Readiness Toolkit (TRT) to measure participants' improvement across program services (Fleming et al., 2024). Since 2020, Utah VR agencies have collected pre- and post-TRT scores to increase accountability and advocate for strategic funding. As the TRT expands to other states, researchers will need to continuously analyze pre-post scores and their connections to RSA-911 data to assess and inform practices (Fleming et al., 2024). Efficient tools are needed to streamline data preparation and ensure consistent analyses, empowering both researchers and VR counselors to interpret the data effectively.

This work addresses the vocational rehabilitation industry's needs by alleviating the burden of data preparation and promoting flexible, reproducible VR research. We develop an `R` package uniquely tailored for cleaning and analyzing RSA-911 and TRT data (R Core Team, 2021). This package automates much of the manual effort involved in cleaning VR data, leveraging a thorough knowledge of the codebook and established cleaning workflows. It pioneers a standardization of the cleaning process and the inclusion of the newly developed TRT data. As `R` is a primary tool for VR researchers, the package offers both a consistent framework for standardizing data preparation across projects and flexibility for experienced programmers. Additionally, we present a novel and user-friendly dashboard for RSA-911 and TRT data analysis, enabling VR counselors to derive insights without technical expertise. Finally, we discuss the accessibility and impact of these tools on vocational rehabilitation and data science.

2 Data Overview

We use both RSA-911 and TRT scores data from Utah VR agencies to guide appropriate package and dashboard development. While the collection of RSA-911 data has been the standard for decades, the collection of TRT scores data began in 2020. Only RSA-911 data from overlapping years are included, as the prime motivation of this work was examining the relationships between participants' demographic data (RSA-911 data) and the respective TRT success scores. All data sets involved in this project have been provided by the Fleming et al. (2024) research team.

The Utah RSA-911 datasets available for this project were all quarters from 2020–2023 and quarters one and two of 2024. In total, the data contain over 385,000 observations, with 48,802 unique participants and 521 initial variables. These variables consist of largely qualitative information pertaining to participant demographics, lifestyle, education, work and VR program status, as well as some quantitative information on work hours, pay and VR program funds and assistance. To understand this extensive collection of variables, we reference the RSA-911 codebook, "Case Service Report RSA-911 PD 19-03" (RSA, 2024b). This codebook contains the instructions, permitted values and associated meanings for the 407 nationally required variables. For a more detailed summary of the variables, see Table 1 in the appendix. As many of the

variables contain key identifying information, the sharing of these datasets is restricted under Institutional Review Board (IRB) Policy.

The TRT scores data contain the pre- and post- assessments in enrolled VR services from 2020–2025 for participants across the eight states involved so far. However, our analyses purposes will focus only on obtaining the subset of TRT participants also found in the Utah RSA-911 data. At the time of writing, the raw TRT scores data include over 30,000 rows with 6,241 unique participants and 32 variables. The variables mainly consist of the pre and post scores for evaluating each of the 10 offered services. These services provide counseling or experiences in the following general areas: job exploration, financial literacy, further education, soft skills, sufficiency.

Each participant is able to enroll in any number or combination of experiences in these areas. For a more detailed summary of the variables in the TRT scores data, see Table 2 in the appendix.

3 `R` Package Development

To streamline data preparation and provide simplified and customized visualizations, we created the `rsa.helpr` package. Package development is guided by the IRB-approved Utah RSA-911 quarterly datasets and the current Utah TRT scores data. We utilize the RSA-911 codebook and the Fleming et al. (2024) team to inform the cleaning workflows and cleaning steps based on variable structure and analyses needs (RSA, 2024b). While the functions in our package have been designed with Utah data as our guide, the nationally standardized format of the RSA-911 and TRT Scores datasets opens the door for expanded testing and implementation to other states.

3.1 Data Preparation Functions

Our data preparation workflow includes several steps, including cleaning, merging, and condensing information. Figure 1 provides an overview of the work, which is explained in greater depth in the following sections.

As such, our `rsa.helpr` package contains three main groups of data preparation functions: helper tasks, loading and cleaning, and merging and condensing.

3.1.1 Helper Functions

After understanding the data cleaning process for our dataset structures, we develop many functions to handle specific variable cleaning problems. These serve as "helper" functions, to

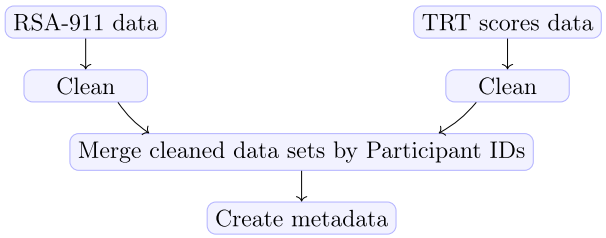

Figure 1: A flowchart describing the general process of data preparation used in this software.

handle small, repetitive tasks. For example, `handle_mixed_date` is able to sift through a variable with mixed date formats (character, numeric, Excel date, mixed orders) and convert to the appropriate `R` date format. This addresses a common issue, especially when merging data from different quarters, as dates can be stored differently in different source files. Below is an example of `handle_mixed_date` being applied to a date variable containing both Excel and character stored dates in a package dataset.

```
# The raw date variable, containing Excel dates and traditional dates
head(rsa_simulated$E7_Application_Date_911, 10)
#>  [1] "NULL"       "NULL"       "NULL"       "2023-08-30" ""
#>  [6] "41067"      ""           "NULL"       "44397"      "NULL"

# Applying cleaning function
var_clean <- handle_mixed_date(rsa_simulated$E7_Application_Date_911)
head(var_clean, 10)
#>  [1] NA           NA           NA           "2023-08-30" NA
#>  [6] "2012-06-07" NA           NA           "2021-07-20" NA
```

Another prominent function is `handle_values`, which scans the variable for codebook-approved value matches (including close matches) and replaces any invalid entries with either NA or a user-specified missing value code. The independent helper function allows the user to provide a list of codebook-approved values for a variable; when employed within the larger cleaning functions, the values are set by our research team in alignment with the most current codebook. Our helper functions are used in combination to develop more complex data cleaning functions. However, all of the helper cleaning functions are package exports and can be used individually for more flexible data preparation, even outside of Utah data or vocational rehabilitation.

3.1.2 Loading and Cleaning Functions

To examine the data as a whole, we first need to load and combine quarterly RSA-911 data into one dataset. Our `load_data` function simplifies this process, utilizing regular expressions to identify appropriate file names, properly joining and accounting for differing variables, and finally returning the combined dataset. The user simply provides the directory, optionally selects specific datasets instead of the full list, and indicates whether or not a downloaded csv copy is desired. An example of this step is demonstrated below:

```
# Loading and combining RSA-911 quarterly datasets
rsa_data <- load_data(directory = "/Users/MyName/Box-folder-directory",
 ↪  download_csv = TRUE)
```

After loading our data, we move to cleaning. The underlying premise of this package design is to create a simple and reproducible workflow. That in mind, we create two main functions, `clean_utah` and `clean_scores`, which yield fully cleaned and merge-prepped datasets, each with a single function call. These two functions perform all cleaning work under the hood for the user, enlisting the helper functions, carrying out additional restructuring steps, and managing type classification. `clean_utah` and `clean_scores` extensively use regular expressions to ensure that variable identification is robust to variations in naming conventions. This tool improves the flexibility of our function design, enabling us to adapt to a range of scenarios.

Furthermore, included in each function are checks to address discrepancies between the expected and observed variable structures. Most decisions inside the functions are based directly on the codebook, minimizing ambiguity in the cleaning decisions. `clean_utah` and `clean_scores` can be executed by providing only the dataset and no additional arguments, using default settings, as demonstrated below. Assuming that `rsa_data` is a stored RSA-911 dataset and `scores_data` is a stored TRT scores dataset, the function calls are as follows:

```
# Cleaning an RSA-911 datset
data_cleaned <- clean_utah(rsa_data)
# Cleaning a TRT scores dataset
scores_cleaned <- clean_scores(scores_data)
```

However, both functions offer additional arguments that can be used to customize some of the alterations made. This allows adaptability for the user, while maintaining the automation and consistency of the process. The detailed parameters for both functions can be found in the `rsa.helpr` documentation. For a visual explanation of the workflow, Figure 2 provides an overview of the steps involved in `clean_utah`, indicating where the function arguments come

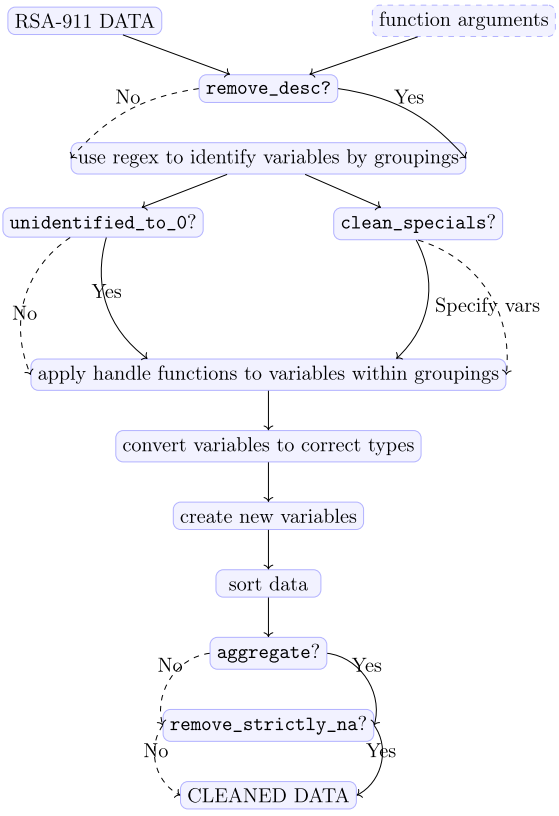

Figure 2: A flowchart describing the general process of `clean_utah`. Solid lines represent automatic or default steps, while dashed lines represent optional function argument values. Code font text represents function arguments. The user begins by inputting a RSA-911 dataset and any desired arguments. Then, the inputted function arguments or default function arguments are used to guide the cleaning steps. Finally, the cleaned data set is returned.

	App_Date	Postal	Job_Vendor	Primary_Dis	White	White_Desc
1	NULL		NULL		1	White
2			4	17;25	1	White
3	41067	UT	2			
4	44397	UT	NULL	18;29	9	Does not wish to self-identify
5	2023-08-30	NULL	NULL	13;1	9	Information Not Available

`clean_utah() using defaults`

↓

	App_Date	Postal	Job_Vendor	Primary_Impair	Prim_Cause	White
1	NA	NA	0	NA	NA	1
2	NA	NA	4	17	25	1
3	2012-06-07	UT	2	NA	NA	0
4	2021-07-20	UT	0	18	29	0
5	2023-08-30	NA	0	13	1	0

Figure 3: Illustrative example of the `clean_utah()` pipeline using simulated RSA-911 data and with arguments left as defaults. Defaults operate as the following: `aggregate = TRUE` removes duplicate rows per quarter for a participant, `remove_desc = TRUE` removes redundant description variables, `unidentified_to_0 = TRUE` converts any missing and unidentified levels to 0, `remove_strictly_na = TRUE` removes variables for which the data is completely missing, `clean_specials = NULL` bypasses applying the `handle_splits` helper function to variables unless specified to save on computation.

into play (e.g., `unidentified_to_0`). Figure 3 provides a before and after cleaning example on a subset of simulated data, applying `clean_utah` with defaults. Whether customizing the parameters or using defaults, the returned datasets are ready for further analysis.

Due to the less complex nature of the TRT scores data, the `clean_scores` function contains fewer steps in the cleaning process. The key task is pivoting the data to one row per participant to prepare it for merging with quarterly RSA-911 data. Additional cleaning steps take place, including converting and cleaning dates, removing missing IDs, and handling multiple scores per participant. A small demonstration of this process is depicted with a subset of simulated data in Figure 4.

3.1.3 Merging and Creating Metadata

After cleaning the quarterly RSA-911 and TRT scores datasets, we merge them by participant ID to create a full profile for each person and explore relationships between demographics and TRT scores. We provide `merge_scores`, designed to take cleaned quarterly and scores data and conduct a merge by ID, resulting in only data corresponding to matched participants.

We then extract the metadata from the merged data using the `create_metadata` function. `create_metadata` takes a cleaned RSA-911 quarterly dataset, or a merged dataset, and extracts the metadata information, with one row per unique participant. This condensing process involves removing duplicate participant metadata entries through grouping variables by type: numeric, categorical, or date. For numeric variables (e.g., work hours, hourly wage), we calculate the median per participant. These numeric variables are typically stable or intermittently missing across quarters, making the median a robust choice for preserving central tendency and limiting the effect of outliers or erroneous entries. Similarly, for categorical variables (e.g., gender, race), we retain the most frequent value to mitigate the impact of data errors or blank entries. For

	ID	Completed	Pre.Post	Score	State	Provider	Service
1	1234	04/20/2023 12:26:40 (MST)	Pre	61.7	Utah	Test Center Inc. (4)	FL
2	1234	04/20/2023 15:05:15 (MST)	Pre	62.5	Utah	Test Center Inc. (4)	FL
3	1234	03/15/2024 08:05:13 (MST)	Post	75.3	Utah	Test Center Inc. (4)	FL
4		11/08/2022 10:29:20 (MST)	Pre	62.2	Utah	R. School Dist. (24)	JS
5	4567	01/04/2022 22:00:00 (MST)	Pre	70.1	Utah	Community Center (11)	JS
6	4567	05/30/2023 11:48:42 (MST)	Post	84	Utah	Community Center (11)	JS

`clean_scores()` using defaults

↓

	ID	Provider	Pre_FL	Pre_JS	...	Post_FL	Post_JS	...	Pre_Date_FL	...
1	1234	TCI	62.5	NA	...	75.3	NA	...	2023-04-20	...
2	4567	CC	NA	70.1	...	NA	84	...	2022-01-04	...

Figure 4: Illustrative example of the `clean_scores()` pipeline using simulated TRT scores data and with arguments left as defaults. Defaults operate as the following: `state_filter = NULL` allows the user to specify a filter for the state of interest, `aggregate = TRUE` rows are aggregated to keep only most recent score for participants with multiple pre or post scores for the same service, `clean_id = TRUE` removes rows where participant ID is missing, `id_col = NULL` allows for the user to specify an ID column name if it differs from convention and does not contain "participant" or "ID".

date-related variables (e.g., time stamps, current quarter), we keep the most recent entry. This approach is justified by the fact that many demographic variables (e.g., gender, race) are unlikely to change across quarters.

The condensing process is illustrated in Figure 5 using a subset of simulated data. For

	Participant_ID	Year	Quarter	White	Dis_Priority	Eligibility_Date	Exit_Wage
1	288941	2022	2	1	NA	NA	0.00
2	288941	2022	3	1	1	2021-12-10	8.00
3	288941	2022	4	9	1	2022-08-20	8.50
4	288941	2023	1	1	1	2023-03-13	10.00
5	234057	2020	1	0	1	2019-09-27	11.00
6	477753	2022	3	1	NA	NA	0.00
7	477753	2022	4	1	2	2022-01-19	0.00
8	477753	2023	1	1	2	2023-04-26	0.00

`create_metadata()`

↓

	Participant_ID	White	Dis_Priority	Eligibility_Date	Exit_Wage
1	288941	1	1	2023-03-13	8.25
2	234057	0	1	2019-09-27	11.00
3	477753	1	2	2023-04-26	0.00

Figure 5: An example of using `create_metadata` condensing quarterly data (top) into a condensed version containing a single measurement per participant (bottom). Factor variables retain the most common observation per participant (breaking ties with the first appearance), while date variables use the most recent measure, and numeric variables are summarized by medians.

example, participant 288941 appears in four rows in the quarterly dataset. After condensing, we retain a single row with the most common categorical values, the latest eligibility date, and the median exit wage. Quarterly indicators like year and quarter are dropped, as they lose meaning when only one row remains. While date variables are primarily record-keeping tools and play a minimal role in analysis, numeric values are better represented by medians due to limited quarter-to-quarter variation. This method allows us to preserve key participant information while enabling clearer, more reliable modeling of trends across the dataset.

3.2 Data Preparation Example

Below is an example of the full data cleaning steps, assuming `rsa_data` is a stored RSA-911 dataset and `scores_data` is a stored TRT scores dataset:

```
# Cleaning an RSA-911 dataset using defaults
rsa_cleaned <- clean_utah(rsa_data)

# Cleaning a TRT scores dataset using defaults
scores_cleaned <- clean_scores(scores_data, state_filter = "Utah")

# Merge the datasets, keep only relevant participants, using defaults
merged_data <- merge_scores(rsa_cleaned, scores_cleaned)

# Generate the condensed metadata
metadata <- create_metadata(merged_data)
```

These four functions within the `rsa.helpr` package provide the opportunity to not only bypass the tedious process of data cleaning, but also to create directly reproducible results.

3.3 Visualization Functions

To support analysis following data preparation, we provide functions for the visualizations we anticipate to be used most often. The first, `visualize_densities`, simplifies the process of creating a density plot to compare the distributions across levels of a qualitative variable. While other plotting functions in `R` can be easily applied to cleaned variables, creating a density plot across categories requires a higher level of coding proficiency. To provide researchers with less `R` fluency the ability to use these valuable plots, the `visualize_densities` function handles all of the necessary programming steps internally. The user must simply input the quantitative and categorical variables of interest, and the density plot is produced, with appropriate labels.

The functions `visualize_scores` and `visualize_metadata` offer more extensive visualizations. These high-level functions streamline the process of creating standard exploratory plots by requiring only two inputs: a cleaned dataset and a selected analysis type specified through the "option" argument. Our visualization functions are designed to align with the analysis choices available in our data dashboard (Section 4), to maintain consistency in analysis across researchers and counselors. Within `visualize_metadata` and `visualize_scores`, variables used for standard visualizations are selected based on their relevance to current VR research (e.g., gender, race, disability severity for metadata; service or provider for scores). The analysis options focus on either distribution overviews or commonly examined relationships. Both `visualize_metadata` and `visualize_scores` follow a similar structure: in `visualize_metadata`, the `general_demo`

Figure 6: An example of the `visualize_metadata` output using `option = "general_demo"` and `one_window = TRUE`.

option generates distributions of key demographic variables individually, while options prefixed with `investigate_` plot a response variable across selected variables; in `visualize_scores`, the pattern is the same, with `overview` serving as the equivalent of `general_demo`. Additionally, both `visualize_metadata` and `visualize_scores` offer a function argument `one_window = FALSE`. When set to `TRUE`, the functions plot all graphs in one window – an option better suited for a quick view of distributions rather than for detailed examination.

A demonstration of the `visualize_metadata` function is depicted in the code below and the plotting output is found in Figure 6.

```
# Creating a series of plots for general demographic overviews in metadata
visualize_metadata(metadata, option = "general_demo", one_window = TRUE)
```

3.4 Handling User Error

We design the functions within `rsa.helpr` to not only appropriately execute their roles, but to anticipate user error. Each cleaning function handles unconventional data formats and naming structures, as well as the possibility of missing variables. The functions include checks that inform the user through messages, either alerting them to any bypasses or halting execution when necessary. For example, although `merge_scores` and `create_metadata` are intended for datasets cleaned using the `rsa.helpr` methods, they are built with checks and safeguards at each stage to ensure their proper functionality if applied to a dataset not vetted by `clean_utah` or `clean_scores`. Additionally, the customized visualization functions verify correct inputs before attempting to execute any plots.

4 Data Dashboard Development

Although the `rsa.helpr` package reduces the analysis workload for VR researchers in `R`, the need for greater accessibility extends beyond programming software development. `rsa.helpr` requires a familiarity with `R`, creating a degree of separation between those without programming knowledge and their data interpretation. For VR counselors who work directly with participants but are not versed in `R`, it is critical to provide access to summaries and results. We accomplish this through the creation of a free, interactive RSA-911 Data Exploration Dashboard. The dashboard operates through the use of our `rsa.helpr` package, ensuring consistency between analyses performed independently in `R` and those displayed in the app. Built using the `shiny` framework, our dashboard heavily utilizes the principle of reactivity (Chang et al., 2024). The dashboard automatically and efficiently responds to changes in user selections, from cleaning options to modeling combinations, ensuring a seamless analysis experience.

4.1 Dashboard Explanation

Figure 7 provides an overview of the dashboard's structure and key features. The following sections explore its functionality in greater detail, supplemented with examples and visuals of the interface in action. The user begins by uploading their dataset(s) of choice–either raw data to be cleaned by the dashboard or pre-cleaned data. If the user uploads raw data to be cleaned, the dashboard applies the appropriate cleaning functions from `rsa.helpr`. Once the data have been cleaned, the user has the option to download the data for future use. The cleaned or pre-cleaned dataset is now stored in the session and displayed. The next step involves the user selecting the dataset type (metadata or independent TRT scores data) to examine, as multiple dataset types

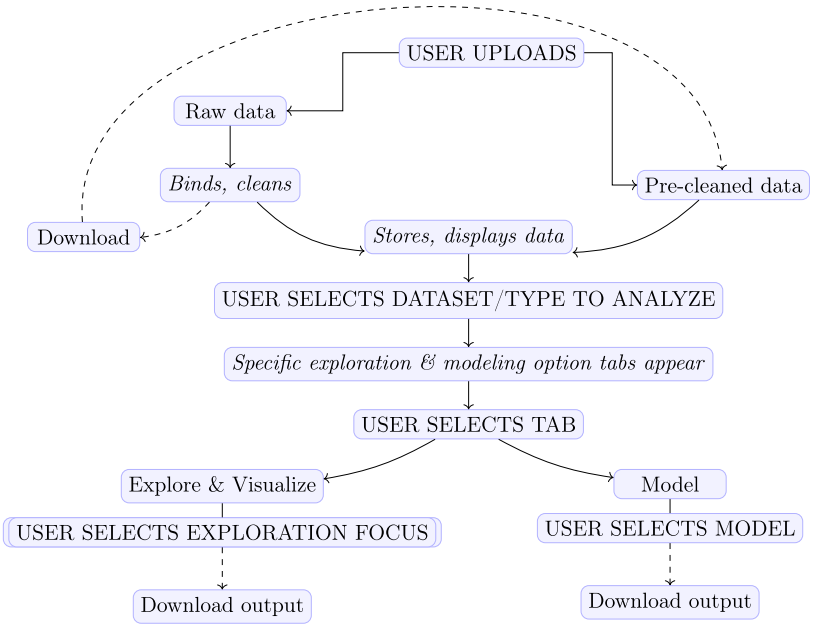

Figure 7: A flowchart demonstrating the overall steps of the dashboard. Dashed lines indicate optional features or uses of the dashboard's functions, while solid lines indicate necessary steps or automated options.

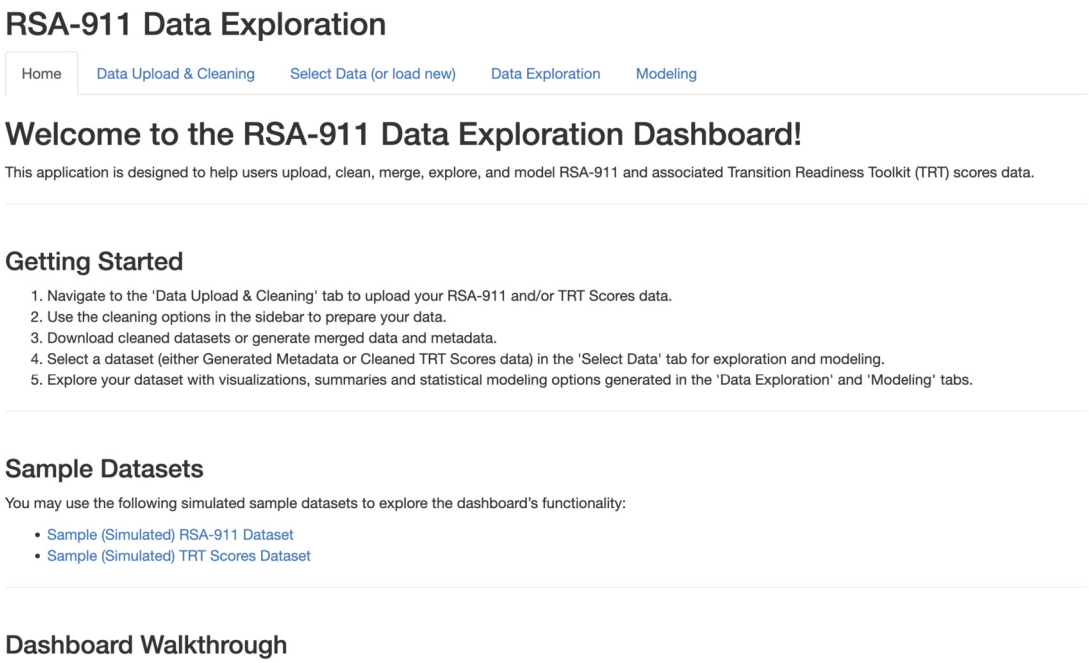

Figure 8: Partial view of the dashboard homepage and layout.

may be uploaded or created in a single session. Once the user has appropriately selected the type, type-specialized visualization and modeling tabs appear. The user can now explore the automatically constructed summaries and plots for different variable relationships of interest or customize a model with a few simple clicks. Finally, the user has an option to download for each plot and modeling output.

Home Tab

The user begins on the `Home` tab, depicted in Figure 8. This homepage provides instructions on navigating the dashboard, as well as links to download simulated example datasets to explore the dashboard's functionality.

Data Upload and Cleaning Tab

Once the user is ready to upload their desired quarterly RSA-911 and/or TRT scores dataset(s), they navigate to the `Data Upload & Cleaning` tab. Once the datasets are fully uploaded, the dashboard automatically merges the files and begins applying the cleaning functions to the data. The adjustable function arguments are provided as checkbox options for the user. The status is shown in the progress bars. Once the datasets are uploaded and fully cleaned, they are stored in the session and displayed with a summary of row, column and unique ID totals for inspection.

If the user uploads both quarterly and scores data, the dashboard automatically creates a merged data set, displayed in the main page. As condensing the data to one row per participant is more computationally intensive, the metadata will not be generated automatically. The user may click the "Generate Metadata" button to initiate this. Again, a progress bar will indicate status, and the final data will be displayed.

Select Data (or Load New) Tab

Now, with the datasets stored in the session, the user selects which dataset to visualize and model. After selecting the `Select Data (or load new)` tab, the user is presented with three dataset options, "Use Generated Metadata," "Use Cleaned Scores Data," or "Upload New Dataset," as depicted below. With the "Upload New Dataset" option, the user has the option to upload a pre-cleaned dataset and identify the correct data type (quarterly RSA-911, merged, metadata or scores). This is an option provided for a user returning in a new session to utilize the visualization and modeling tabs with pre-cleaned data.

As described in Figure 7, the user is asked to specify the type of dataset is being uploaded, which informs the dashboard on which visualizations and modeling options should be available. The dashboard validates the user's choice based on generic variable naming patterns. If the correct data type is selected, there is no message; if an incorrect data type is selected, there is a warning message to the user that they have selected the wrong type.

```
Select Data (or load new) > Options: Use Generated Metadata,
                                     Use Cleaned Scores Data,
                                     Upload New Dataset
```

Data Exploration Tab

Different data exploration tabs appear in reaction to the dataset type selected by the user. These tabs are designed for simplicity in summaries and visuals for general users. When the user selects "Use Generated Metadata", tabs appear, each containing different summaries and visuals:

```
Use Generated Metadata > Data Exploration > Options: General Demographics,
                                                     Difference Scores,
                                                     Wage,
                                                     Employment,
                                                     Post-Secondary Enroll.
```

The `General Demographics` tab consists of overview summary statements and plots for common variables of interest, such as race, gender, disability severity, impairment types, etc. Figure 9 captures a partial view of the layout, while Figure 10 depicts an example of a plot that would be found in this tab, providing the distribution of a key demographics variable, primary impairments. Tabs `Difference Scores`, `Wage`, `Employment`, `Post-Secondary Enrollment` investigate these notable and potential response variables. These tabs examine the relationships between each potential response (difference scores, wage, employment and post-secondary enrollment) and the key demographic variables found the `General Demographics` tab. An example is shown in Figure 11, which provides a visual output found in the `Difference Scores` tab. In this figure, we see the distribution of (median) difference scores across disability severity.

A similar system of tabs results when the user selects "Use Cleaned Scores Data." For cleaned scores data, corresponding tabs with summaries and visuals appear:

```
Use Cleaned Scores Data > Data Exploration > Options: Overview,
                                                     Across Services,
                                                     Across Providers
```

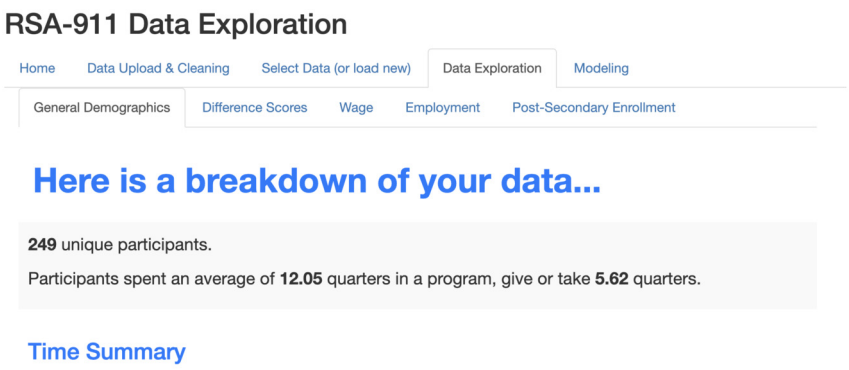

Figure 9: Partial view of the dashboard's `General Demographics` Tab for metadata.

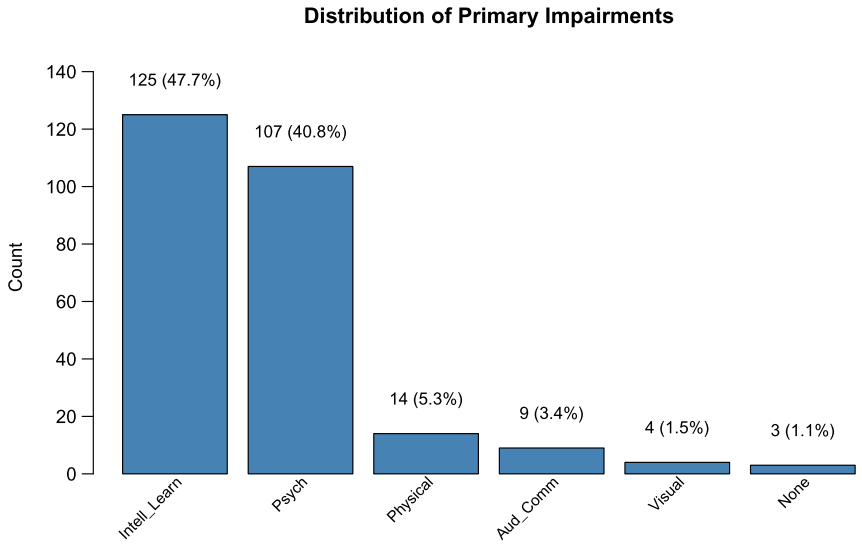

Figure 10: An example of a visual output found in `Use Generated Metadata > Data Exploration > General Demographics` Tab.

These tabs and corresponding output allow the user the option to examine TRT scores data independent of RSA-911 demographic data. As such, the visuals and summaries consist of only variables found in TRT scores data.

The `Across Services` and `Across Providers` tabs examine how TRT score variables relate to available services and provider locations, respectively. In this context, services and providers are treated as the response variables for the TRT scores data. Figure 12 shows one example of an `Across Services` tab output, illustrating the distribution of difference scores across services.

Modeling Tab

Different modeling options also appear in reaction to the dataset type selected. When the user selects "Use Generated Metadata," they have the option of fitting Ordinary Least Squares (OLS) models for predicting Median Difference Score or Ending Wage, or fitting logistic regression mod-

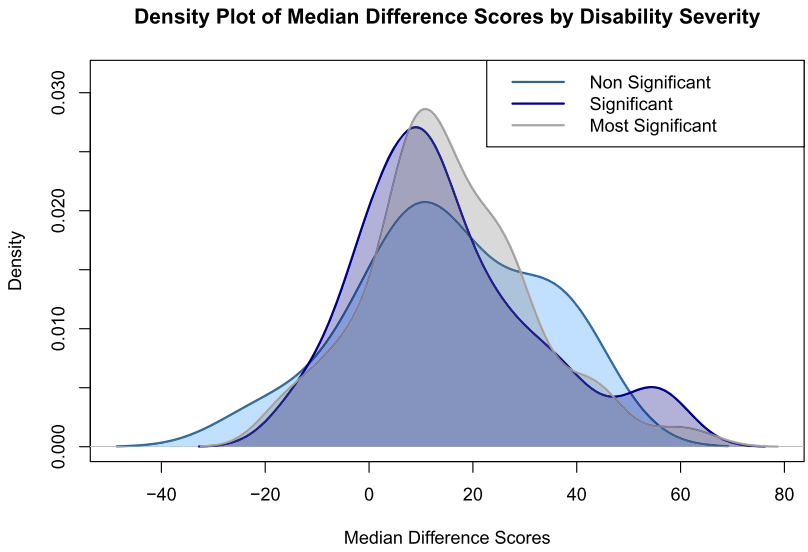

Figure 11: An example of a visual output found in `Use Generated Metadata > Data Exploration > Difference Scores` Tab.

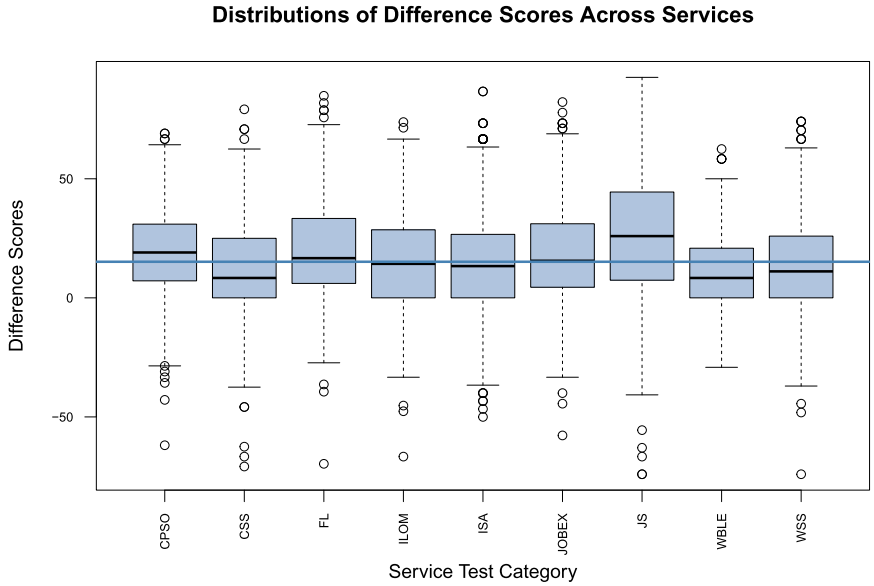

Figure 12: An example of an output in the `Data Exploration > Across Services` Tab for a TRT scores data set.

els for predicting Employment Outcome or Post-Secondary Enrollment, displayed below. These models are created using standard `R` functions and rely on conventional assumptions, including linearity, independence, and specific distributional structures. In practice, these assumptions may not always be satisfied.

```
Use Generated Metadata > Modeling > Response Options: Median Difference Score
                                                      Ending Wage,
                                                      Employment Outcome,
                                                      Post-Secondary Enroll.
                                    Predictor Options: Gender,
                                                       Race,
                                                       Disability Severity,
                                                       Enrollment Length,
                                                       Primary Impairment,
                                                       Secondary Impairment
```

Once the response variable is chosen, the user can select different combinations of predictors, with the model being recalculated for each change. The current subset of predictors was selected for simplicity and reflects common research interests and recommendations from the authors of Fleming et al. (2024). Each time the model is refit, its refined output will be displayed with the option for a direct download. Model outputs are deliberately simplified to reduce misinterpretation and include error messages if a variable with too few observations is selected. Additionally, appropriate model residual plots and explanations describing what the plots indicate and when results should be interpreted with caution are outputted below the model. These diagnostics allow users to visually assess whether key model assumptions appear reasonable without requiring advanced statistical training. However, this tab is intended as an exploratory feature rather than a standalone decision-making tool. The dashboard is designed to support preliminary analysis and collaborative interpretation with researchers or analysts, while more descriptive and broadly interpretable summaries are provided in the `Data Exploration` tab for general users. Users should exercise caution when exploring modeling results, especially when exploring small or sparse datasets.

When the user selects "Use Cleaned Scores Data", they have the option of two different Analysis of Variance (ANOVA) models, either for comparing median difference scores across services or across providers, shown below.

```
Use Cleaned Scores Data > Modeling > Select ANOVA Test: Across Services,
                                                        Across Providers
```

Once run, the refined ANOVA output will then be shown, as well as any significant pairwise comparisons and the option for direct downloads. Similarly to the metadata modeling format, corresponding residual plots and explanations are outputted below the model output.

4.2 Dashboard Conclusion

The RSA-911 Data Dashboard offers VR researchers and counselors a seamless tool that supports the entire data analysis process. Each step, from data cleaning to generating customized models and visual outputs, can be completed with just a few clicks. The dashboard simplifies the traditional coding process for analysis, delivering accessibility to a broader range of VR professionals.

5 Conclusions

5.1 Access to Tools

Our `rsa.helpr` package and its integrated dashboard streamline the process of data management, enabling users to more easily explore and analyze RSA-911 and TRT scores data. This toolkit allows VR researchers and professionals to bypass the need for intensive `R` programming. Designed with an intuitive workflow in mind, the package and dashboard anticipate user decisions and potential errors, making data analysis more accessible and efficient. The toolkit is now freely accessible to any user. The `rsa.helpr` package is available publicly on GitHub to be downloaded and installed at the following link:

https://github.com/rtaylor456new/rsa.helpr

For more detailed information and demonstrations of package functions, see the pkgdown site for the package in the following link. The README on this site includes links to the full vignette and other documentation.

https://rtaylor456new.github.io/rsa.helpr/

Additionally, the RSA-911 Data Exploration Dashboard is currently hosted on shinyapps.io to demonstrate its functionality. This current host allows users to access it directly and freely, though the upload space is limited. The dashboard can be accessed at the following link:

https://rsa-data-dashboard.shinyapps.io/rsa_dashboard_app/

5.2 Contributions

Our tools offer an enduring contribution to the fields of vocational rehabilitation and data science. With RSA-911 data collection ongoing and TRT scores collection expanding, research is far from concluded. As the `rsa.helpr` package and the RSA-911 Data Exploration Dashboard are designed to work with any RSA-911 or TRT scores datasets, we ensure their continued relevance for future projects. The open access release of our tools also enables future VR researchers to enhance their functionality as needed. In collaboration with Fleming et al. (2024), these tools are undergoing initial testing with VR researchers, with plans for broader evaluation and refinement based on practitioner feedback.

Our package and dashboard provide not only solutions to the field of vocational rehabilitation, but examples of systematic tools for data science. A notoriously time intensive part of any data analysis project is data cleaning. Our functions in `rsa.helpr` demonstrate approaches that can be used to optimize this process, particularly for messy human survey data. Meanwhile, the RSA-911 Data Exploration Dashboard highlights the capacity to build an interactive

workspace. Both our package and dashboard are crafted to anticipate user motivations and potential errors at every step, making them practical for real-world use. Through this design, we expand the opportunities for realistic applications of statistical software on applied data problems.

5.3 Limitations and Future Work

A key limitation of our work was the sparseness of participants with both RSA-911 and TRT scores data. The resulting small merged dataset reduced the impact of the visuals, especially for the numerous multi-level categorical variables, and complicated both visualization and modeling options. Another limitation is related to the current deployment of the dashboard on shinyapps.io. Because the free version of shinyapps.io has limited storage, it cannot support many quarterly datasets at once. Users wishing to use the dashboard on large sets of data are recommended to download the source code and run the app on their own machine or server. We are actively working with a team of vocational rehabilitation (VR) researchers to establish a dedicated production server with sufficient storage to support multiple dataset uploads. This server will be fully supported and maintained through funding secured by the Utah State Office of Rehabilitation, ensuring the dashboard remains accessible and sustainable for long-term use. Additionally, as the project and data collection are ongoing, further analyses are planned. This will allow us to extend or adjust the functionality of both the package and the dashboard over time.

Appendix

Included in this appendix are summary tables for the variables and cleaning steps corresponding to RSA-911 and TRT scores data, respectively.

Variable Description Tables

Table 1: This table includes a summary of the variables found in RSA-911 data and their respective cleaning tasks. Note that this is not an exhaustive list of the variables, but rather a summary based on variable groupings. For example, within the variable grouping "Numeric", there is a subgrouping of "Amounts" variables, for which examples of variables within this subgroup are shown. Cleaning tasks are organized into groups based on the similar structures of variables within each group.

Variable Grouping	Variable Example(s)	Cleaning
ID		
	`Participant_ID`	Ensure value exists
Numeric		
Year	`E1_Year_911`	Extract YYYY
Quarter	`E2_Quarter_911`	Check values
Age	`Age.at.Application`	Check values
Amounts	`E52_Plan_Hourly_Wage_911`, `E52_Plan_Compensation_Amt`, `E112_Enrollment_TitleI_911`, `E394_App_Workers_Comp_Amt`, etc.	Check values
Hours	`E53_Plan_Weekly_Hours_Worked_911`, etc.	Check values
Dates		
	`E7_Application_Date_911`, `E40_OOS_Start_911`, `E347_Skill_Gain_Occupation_911`, `E399_Plan_Extension_911`, etc.	Convert Excel dates
Factor (Nominal)		
Codes	`E4_Agency_Code_911`, `E21_Referral_Source_911`, `E51_Plan_Occupation_911`, etc.	Check for existence
Vendor	`E99_JobExploration_Vendor_911`, `E186_Disability_Vendor_911`, `E331_Interpreter_Vendor_911`, etc.	Check and convert values
Provider	`E97_JobExploration_Provided_911`, `E196_Miscellaneous_Comp_Provided_911`, `E212_Assessment_Provided_911`, etc.	Check and convert values
Purchase	`E98_JobExploration_Purchased_911`, `E269_Benefits_Purchased_911`	Check and convert values

Variable Grouping	Variable Example(s)	Cleaning
Factor (Nominal)		
Demographic	`E10_Indian_Alaskan_911`, `E16_Veteran_Status_911`, `E42_Has_Disability_911`, `E65_Plan_Homeless_911`, etc.	Check and convert values
Gender	`E9_Gender_911`	Check and convert values
Additional binary	`VR_Case_Type_Flag`, `E392_Q2_Q4_Employer_Match_911`, etc.	Check and convert values
Factor (Ordinal)		
Education status	`E78_Secondary_Enrollment_911`, `E84_PostSecondary_Enrollment_911`	Check and order values
Exit status	`E356_Exit_Work_Status_911`	Check and re-order values
Exit type	`E354_Exit_Type_911`	Check and order values
Exit credentials	`E378_PostExit_Credential_911`	Check and order values
Employment	`E379_Q1_Employment_911`, `E383_Q2_Employment_911`, etc.	Check and order values
Special Characters		
Disability	`E43_Primary_Disability_911`, `E44_Secondary_Disability_911`	Separate into Cause and Impairment
Support	`E394_App_Public_Support_911`, `E397_Exit_Medical_911`, etc.	Separate values into new columns
Comp	`E135_Graduate_Comp_911`, `E225_Diagnosis_Comp_911`, `E335_Interpreter_Comp_911`, etc.	Separate values into new columns
Extra		
Description	`E9_Gender_Desc` `E45_Disability_Priority_Desc`, etc.	Remove if specified
Administrative variables	`RSA911_2020_Data_Created_Staff_ID`, `RSA911_2020_Data_Updated_Staff_IP_Addr`, etc.	Remove if specified

Table 2: Depicted below is a summary of the variables found in TRT scores data, and their respective cleaning tasks.

Variable	Description	Cleaning
Participant ID	ID number string	Check for existence
Completed Date	Character in MM/DD/YY time order	Convert to appropriate date format
Pre.Post	Character variable, either "pre" or "post"	Separate into individual columns for each service
Score	Numeric variable	Convert to wide format; values are used for individual Pre and Post Service Columns
Difference	Numeric variable	Convert to wide format; values are used for individual service columns
Provider	Character variable, abbreviation of provider agency	Use to reformat data to wide shape
State	Character variable, abbreviation of state of residence	Option to filter to specific state – needed for merge with quarterly data. Handles abbreviations or full length names.
Service	Character variable, abbreviation of TRT-defined service	Use to reformat data to wide shape
Question Items	Total of 15 columns, one per possible item on TRT assessment	No cleaning required for current analyses

References

Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, ..., Borges B (2024). shiny: Web application framework for r. R package version 1.8.1.1.

Commission UEEO (1973). Rehabilitation act of 1973 (Original text). US EEOC. https://www.eeoc.gov/rehabilitation-act-1973-original-text. Accessed 8-15-2025.

Fleming AR, Phillips BN, Riesen T, Langone A (2024). Enhancing transition outcomes: A toolkit to facilitate data-driven pre-employment transition services. *Journal of Vocational Rehabilitation, (Preprint)*, 1–13.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

RSA (2004). Reporting manual for the case service report (rsa-911). Example of documentation https://rsa.ed.gov/sites/default/files/subregulatory/pd-16-04.pdf. Accessed 8-15-2025.

RSA (2024a). About RSA - Rehabilitation Services Administration. https://rsa.ed.gov/about. Accessed 8-15-2025.

RSA (2024b). Case Service Report (RSA-911). Case Service Report (RSA-911) | Rehabilitation Services Administration. Retrieved from https://rsa.ed.gov/performance/rsa-911-policy-directive. Accessed 8-15-2025.