

Introduction to
Research Data Management

James D. Hosking, Ph.D.
Ronald W. Helms, Ph.D.

Department of Biostatistics
The University of North Carolina

Copyright © 1993, James D. Hosking and Ronald W. Helms

1. Introduction

A Research Data Management System is a set of procedures (manual and automated) that assure the secure, accurate, collection, transfer, and processing of data, producing a database suitable for analysis.

1.1. Components of a Research Data Management System

- **Data Acquisition:** The measurement of the phenomena of interest and the subsequent recording of those measurements.
- **Data Coding:** The conversion of "free-text" responses to structured categories to facilitate tabulation and analysis.
- **Data Entry:** The conversion of study data to machine-readable form.
- **Data Editing:** The process of examining the collected data to detect suspicious values, and collecting further information to determine the appropriate action.
- **Database Processing:** The maintenance and modification of the accumulated data in an accessible form.
- **Data Communications:** Procedures to insure the timely and accurate transfer of information among study agencies.
- **Data Inventory:** Procedures to record the existence and status of data records from their creation through all data processing stages.

- **Data Base Closure:** A final, independent evaluation of the completeness and accuracy of the database after the completion of ongoing data processing, prior to definitive analysis.
- **Statistical File Creation:** The retrieval and manipulation of information from the data base to produce rectangular "analysis files".
- **Confidentiality:** Procedures to protect individually identifiable data from use by unauthorized persons or for unauthorized purposes.
- **Security:** Procedures to protect data from accidental or unauthorized modification, destruction, or disclosure.
- **Archiving:** The process of organizing and storing a data base and supporting information so that it can be used at a future date, perhaps by persons not involved in the original project.

1.2. Levels of Complexity in Data Collection

Research projects vary widely in the magnitude and complexity of data collection. The level of sophistication and automation (and therefore development and operational cost) should be matched to the level of complexity. Factors influencing complexity include:

- Number of participants
- Types and amount of data collected per participant
- Number of data collection occasions per participant
- Number of data collection locations
- Duration of data collection
- Need to transfer samples to laboratories and reading centers (local or central)
- Regulatory requirements for auditability
- Need for blinding of treatments and evaluations
- Level of confidentiality / security required

For the purpose of designing a Research Data Management System (RHEUMS), it is useful to categorize projects into three general levels of complexity:

- Studies with a single batch of data
 - Data collected at one location
 - Data collected over a short period of time
 - Typically one data collection occasion per participant
 - Data processed "all at once," after data collection
- Studies with a single data collection site, but multiple batches of data
 - Data collection and processing overlap
 - Transfer of samples to local labs
 - Multiple data collection occasions per participant
- Studies with multiple data collection sites and multiple batches of data
 - Transfer of data to a central data processing center
 - Transfer of samples to central laboratories and reading centers

Each of these levels of complexity requires a significant increase in the sophistication of the RHEUMS.

The remainder of this section will focus on studies with a single batch of data.

2. Data Entry and Verification

Data Entry: the conversion of study data to an electronic form suitable for further processing and analysis.

Issues:

- Organization of data entry files
- Error detection during entry
- Functions to facilitate data entry
- Software alternatives for data entry
- Use of commercial data entry organizations

2.1. Organization of data entry files

For the purpose of data processing, data files should generally be organized with one dataset per form type, and one record per individual form.

It is essential that every file within the database contain a standard set of *key fields* which uniquely identify each record in the file. These variables should have the same format (length and storage representation) in each data set. In general, four types of identifying information are needed:

- A study identifier
- A participant identifier (SSN, study ID number, etc.)
- A form type identifier
- A data collection occasion identifier (visit number, date of hospitalization, etc.)

It is preferable for each of these identifiers to be specially assigned by the project, rather than collected data. "Natural identifiers" such as SSNs and dates are more prone to error and cannot be guaranteed to be unique. However, it is desirable to collect such natural identifiers in addition to the assigned study identifiers to provide redundancy in the identification of records.

2.2. Error detection during entry

Any data entry system should include procedures to detect errors made in the keying process. Empirical studies of key-entry of research data have shown error rates of .5-2% of characters on initial entry by professional keyers and 2-5% with inexperienced keyers.

Proofreading is a possible strategy for detecting keying errors in small studies. A listing of the keyed data is visually compared to the original data forms. The effectiveness of proofreading is increased if the proofreader is a different individual than the keyer. In a well managed system, proofreading will detect approximately 75% of keying errors.

Re-key verification is a feature provided by most data entry software. As each record is keyed for the second time, each character is compared to the first entry of the record. When the two characters disagree, the system alerts the operator, who then confirms the appropriate character. As with proofreading, verification is more effective if the first and second keying are by different individuals. Re-key verification will identify approximately 90% of keying errors.

Double entry and compare is a variation on re-key verification, in which each batch of data is keyed twice, into two separate files. Corresponding records in the two files are then compared by a program, which prints a report of discrepancies.

Editing during entry can be used as a supplement to, but should not replace the methods above. Each character and/or field is compared by the data entry software against one or more validation rules (lists of acceptable characters, etc.) and the keyer is prompted to correct or confirm unexpected values.

2.3. Functions to facilitate data entry

Data entry software can incorporate a number of functions to increase the speed and accuracy of data entry.

- Display formatting options
 - Prompt format
 - Form format
 - Table (spreadsheet) format
- Field initialization options
 - Constants
 - Default values
- Duplication of values from record to record
 - Manual
 - Automatic
- Computing the value of a field based on the value of other fields
- Skipping fields "not applicable" based on the value entered for another field.

2.4. Software alternatives for data entry

Data entry can be accomplished using a variety of types of software.

- Data entry software: These packages typically provide the most features to facilitate keying. Most provide limited facilities for data management, data retrieval, or report generation. Useful examples include:
 - Entrypoint 90
 - SAS FSEDIT / FSVIEW / COMPARE
 - Epi Info
- Database Management Systems: The depth of data entry functionality provided by DBMSs varies from very limited to matching the quality of data entry software. However, few are designed to deal with forms of the length and complexity of research data collection forms. Typical examples include:
 - Dbase / FoxPro / Clipper
 - Oracle / Informix / Ingres
 - SIR
- Text editors, word processors, spreadsheets: These provide none of the control, facilitation, or validation features of the alternatives above. In view of the wide availability of superior alternatives, packages of this type should not generally be used for data entry.

2.5. Use of Project Staff for Data Entry

In many projects, data entry is performed by members of the research team, often as one of several functions.

Advantages:

- Maximum control of priorities, procedures
- Maximum security, confidentiality
- Possibility of using staff other functions in addition to data entry

Disadvantages

- Need to purchase and maintain hardware and software
- Training and supervision effort
- Low capability to adjust to fluctuations in data entry workload over time

2.6. Use of Commercial Services for Data Entry

Numerous commercial data entry services are available to perform data entry on a contract basis. The contract typically involves a one time project set-up charge, and hourly charges for keying (and verification).

Advantages:

- Specifiable cost / keystroke
- Specifiable schedule
- Specifiable error rate
- Minimal training, supervision effort
- Ease of adjustment to fluctuations in workload over time

Disadvantages

- Need to contract, interact with another organization
- Less control of procedures, priorities
- Increased handling, transfer of paper forms

3. Data Editing

Data editing is the process of examining data to detect suspicious values, collecting further information to determine the appropriate action, and implementing and documenting that action.

Objective: To produce data files with acceptable error rates at an acceptable cost.

3.1. General Principles

Errors Happen:

- All non-trivial data files contain errors.
- Some types of data have higher error rates than others.
- Some types data (and types of errors) are more important than others.
- After data editing, all non-trivial data files will still contain errors.
- The goal: To develop a cost-effective system to reduce the rate of important errors to an acceptable level.

Prevention: Preventing errors is typically much more cost effective than detecting and correcting errors.

- Use simply organized forms with simple, easily understood items
- Simplify data recording (e.g., multiple choice rather than fill-in-the-blank).
- Clearly document procedures for forms completion, and train the data collectors.
- Clearly document data entry and processing procedures, and train the data processors.
- Have a pilot study or phase whn possible. Always immediately review the initial "real" data.

Redundancy: Error detection is based on redundant information.

- Redundancy may be within the data form or between the data form and some external source of information (e.g., a table of valid values).
- The degree of redundancy determines the sensitivity and specificity of the error detection process.
- Duplicate collection of critical data items (100% redundancy) may be cost effective.

3.2. Steps in the Error Detection / Correction Process

Error Detection: Comparison of data values against validation rules to identify values that are likely to be incorrect.

Interim processing: Documenting the detection of questionable values and storing and handling them until resolved.

Problem resolution: Investigating each questionable item and determining the appropriate action to be taken

Error correction: Implementing and documenting the appropriate action.

3.3. Error Detection

Types of tests or "validation rules":

- Character validation: Compare each character in a data field to a set of valid (expected) characters.
- Range tests: Compare the value of a field to a reasonable "valid range" (upper and lower limits). Usually used for continuous numeric values.
- Valid value test: Compare the value in a field to a list (or table) of "valid" values.
- Field completion test: Compare the existence of data in a field to a logical rule defining when it should be present.
- Multivariate tests: Compare the values of two or more fields for logical (or probabilistic) consistency.

Issues in implementing character validation:

- Text fields: allow upper and lower case.
- Multiple choice fields: Translate responses to consistent case.
- Plan for non-alphabetic characters in text fields (e.g., - & ')
- Plan for non-numeric characters in numeric fields (e.g., "not done", "unknown", "trace", "bdl")

Issues in implementing range tests

- Objective: select limits that maximize the number of erroneous values detected, while minimizing the number of correct values questioned.
- Range tests often have to accept special codes.

General strategies to facilitate data editing

- Have standard codes for permanently missing data.
- Allow data collectors to confirm unusual values during forms completion.
- Avoid generating (or sending data collection staff) stacks of field completion messages due to skip rules, blank pages, etc.
- Plan for edits to deal with partial dates (??/OCT/93), and other incomplete responses.

Human review vs. programmed checks

- An knowledgeable person is more cost-effective than computer software for detecting novel and/or complicated patterns of data inconsistency. Automated checks should be restricted to recurrent problems.
- Computer programs perform diligence tasks (e.g., field completion, valid range tests) much more accurately than people.
- Most systems will be most cost-effective with a combination of human and programmed editing.

3.4. Interim Processing

Documenting the suspicious values:

- Editing procedures (computer and manual) should produce printed reports ("data queries") containing:
 - The identification of the form and item(s) questioned
 - The current values of the suspicious item(s)
 - A description of the validation rule violated
 - Space for recording the appropriate corrective action
- A query inventory file, containing an electronic record for each suspicious value detected is often useful in a variety of ways:
 - To re-print misplaced reports
 - To identify problem items through tabulations
 - To allow suspicious values to be excluded from analyses

Interim handling of records with suspicious values:

- Do nothing (store all records together)
- Separate records into a "clean file" and a "dirty file"
- Replace suspicious items with missing values until resolved

- Maintain field status characters to flag suspicious values

3.6. Problem Resolution / Error Correction

Possible resolutions

- Confirm value as correct
- Replace value with corrected value
- Flag value as permanently questionable or missing.

Resolving the problem:

- Internal review and comparison of questioned items against the original data form to identify and correct keying and data processing problems.
- Investigation of remaining problems by data collection staff to determine appropriate action.

Implementing and documenting the resolution

- Update the computer files as appropriate.
- Verify that the updates were performed completely and accurately
- Store the resolved edit listing
- Update the query inventory file, if applicable.

3.5. Handling Other Data Problems

Define standard procedures for handling other types of data issues, including:

- Permanently missing forms (due to death, refusal, equipment failure, human error, etc.)
- Data queries and corrections initiated by project staff (data collection and data processing).
- Other data problems not related to individual data items ("problem logs").

4. Data Coding

Data Coding: The conversion of "free-text" responses to structured categories to facilitate tabulation and analysis.

4.1. When is Coding Needed?

- When the possible responses cannot be pre-specified.
- When the sample or response must be evaluated and classified by a specialist (other than the data collector).
- When the number of possible responses is too large to include on the data collection form.
- For standardization of terms, spelling, abbreviation, etc.

4.2. Standard coding systems

Standardized coding systems are published (and regularly updated) for many types of data. For example:

- Adverse events
 - COSTART (FDA)
 - WHO-ART (WHO)
- Causes of disease, hospitalization, death
 - ICD-9, ICD-9-CM (WHO)
 - DRG
- Medications
 - USAN (US Pharmacopoeial convention)
 - INN (WHO)

Occupations

- National Institute of Occupational Health dictionary of occupations

Geographic location

ZIP codes

FIPS codes

4.3. Guidelines for the Use of Coding systems

- Identify the information necessary to code events while designing the data collection forms.
- Decide on the level of detail needed in assigning codes. (In general, code to the full level of detail available).
- Insure that the coding system selected identifies the relevant attributes of the items being coded.
- Plan to develop guidelines for coding items with incomplete information.

5. Database Closure

Database Closure: The process of assuring the completeness and accuracy of data processing, preparing the database for definitive analysis.

Major closure Tasks:

- Evaluate the completeness and accuracy of data processing and resolve or document outstanding problems.
- Evaluate the "external validity" of the overall data collection / management process.

5.1. Evaluate completeness and accuracy of data processing

- Verify the identity and status of study participants
 - Participants screened and excluded
 - Participants screened and enrolled
 - Enrolled participants completing study
 - Enrolled participants withdrawn not completing study (dead, lost to follow-up, etc.)
- Assure that all expected data have been received
 - Comparisons of scheduled forms received to protocol requirements
 - Comparison of "as needed" forms received to conditions requiring their collection
 - Check for "unexpected" forms received
- Assure all received data have been processed
- Assure all data queries have been resolved

5.2. Evaluate external validity of overall process

- Manually compare analysis file to original data forms
 - Complete check of critical data items (endpoints)
 - Check a sample of other data items
- Compare data forms to "source data"

6. Data Security, Confidentiality, Backups

Data Security: Procedures employed to protect data from accidental or unauthorized intentional modification, destruction, or disclosure.

Confidentiality: The protection of individually identifiable data from use by unauthorized persons or for unauthorized purposes.

6.1. Items requiring security

- Data
 - Source data (forms, samples, recordings)
 - Database records (created from source data)
 - Closed analysis files
- Software
 - Data management programs
 - Data analysis programs
- Operational records and documentation
 - Data processing logs
 - Protocols, procedures, minutes, reports
 - Software documentation
- Equipment

6.2. Threats to Security

- Natural Disasters: Floods, tornadoes, fires
 - Infrequent
 - Catastrophic
 - Readily detectable
- Accidents: Human error, equipment failure, software bugs
 - Frequent
 - Usually limited in effect
 - Less detectable
- Deliberate acts: Sabotage by vindictive employees, hackers, etc.
 - Infrequent
 - Catastrophic
 - Often deliberately concealed

6.3. Security Procedures

- Physical site security: Office space containing data forms and files should be locked when unattended.
- Data forms control:
 - When access to original data collection forms is required, a standard procedure for documenting and controlling checkout and return should be employed.
 - Where practical, staff needing to remove forms from the primary storage area should be provided photocopies rather than originals. Checkout and return procedures should apply to copies as well as originals.
- Data file protection:
 - Standard security provisions of the computer environment (e.g., passwords, access restrictions) should be used to protect data files and programs.
 - Passwords should be changed on a regular basis, and not stored within program source code.
 - Where the security features of the computing environment are inadequate (e.g., in the PC-DOS environment), files can be encrypted using standard algorithms to prevent unauthorized access or modification.

- Data management quality assurance:
 - Standard testing procedures should be an integral part of software development. Develop a "validation library" of test cases (with correct output for each).

- Personnel training and supervision
 - Personnel need to be informed of the rationale for data security and trained in the project security procedures.

 - Periodic reinforcement of the need for staff to observe security precautions is necessary.

6.4. Backup Procedures

- Original data forms
 - A backup copy of each data collection form should be made as soon as possible after completion.
 - Alternatives for producing backups include photocopying or the use of carbonless (NCR) form sets.
 - Backup copies should be stored at a site physically remote from the storage of the original data collection forms.
- Data files
 - A "working backup" of each dataset should be made every few hours during data entry.
 - A comprehensive backup of the database should be made at the end of each day on which the database was modified.

Backups should be stored on a different storage volume than the primary copy database.

Use of father / grandfather backup cycles allows multiple generations of backups to be maintained at essentially the same cost as a single backup.

Backup files should be transferred off-site on a scheduled basis, and at the completion of each significant event in the project's life (e.g., modification of the data management system, end of a phase of data collection, etc.).

- Programs
 - A "working backup" of each program should be made at least once an hour during programming.
 - A backup copy of each version of a program used "in production" should be stored at a remote location.

Introduction to Research Data Management Using SAS Software

Table of Contents

Topic	Page
1. Introduction	2
1.1. Components of a Research Data Management System	2
1.2. Levels of Complexity in Data Collection	4
2. Data Entry and Verification	6
2.1. Organization of data entry files	7
2.2. Error detection during entry	8
2.3. Functions to facilitate data entry	9
2.4. Software alternatives for data entry	10
2.5. Use of Project Staff for Data Entry	11
2.6. Use of Commercial Services for Data Entry	12
3. Data Editing	13
3.1. General Principles	13
3.2. Steps in the Error Detection / Correction Process	15
3.3. Error Detection	16
3.4. Interim Processing	18
3.6. Problem Resolution / Error Correction	19
3.5. Handling Other Data Problems	20
4. Data Coding	21
4.1. When is Coding Needed?	21
4.2. Standard coding systems	22
4.3. Guidelines for the Use of Coding systems	23
5. Database Closure	24
5.1. Evaluate completeness and accuracy of data processing	25
5.2. Evaluate external validity of overall process	26
6. Data Security, Confidentiality, Backups	27
6.1. Items requiring security	27
6.2. Threats to Security	28
6.3. Security Procedures	29
6.4. Backup Procedures	31