OFFICE OF SURVEILLANCE AND EPIDEMIOLOGY (OSE)

Standards for Data Management and Analytic Processes in the Office of Surveillance and Epidemiology (OSE)

CONTENTS

PURPOSE

- This MAPP outlines quality system practices for planning, implementing, and validating the management and analysis of epidemiologic data in the Office of Surveillance and Epidemiology (OSE) using internal or external databases or electronic data provided to the FDA in support of regulatory activities. Adherence to the practices described in this document will help ensure that OSE analytic projects meet the level of quality necessary to inform regulatory decision-making.

BACKGROUND

- To conduct postmarketing safety evaluations of therapeutic products, OSE staff access data from different sources. For example, they may use data from the regulated industry, longitudinal patient-level electronic medical records data (e.g., General Practice Research Database), or administrative claims data (e.g., Verispan’s Vector One®: National (VONA), Premier’s RX Market Advisor). OSE analysts use these databases to conduct various analytic public health-oriented projects that are undertaken to study drug safety issues. The results of these projects are used to inform regulatory decisions as well as policy making.

- The means by which analysts access these databases vary with, and are dependent upon, the specific data source. Some data vendors provide access to the data through customized software, while others provide the data in a raw form that requires the OSE analysts to perform extensive data management steps before starting data analysis. Managing large datasets is a complex task; thus, it is critical that analysts use a standard approach when performing data management and analysis to ensure a quality work product.

- This MAPP focuses on building quality system practices into the analytic process itself. Before data analysis begins, however, the analyst should have an acceptable study protocol in place that includes, in addition to the traditional components,[1] a clear and precise analytic plan. In broad terms, the plan should describe the analytic methods selected to address the scientific question at hand. In addition, it should state the operationalized definitions of the outcome, the main exposure, other important covariates, and the primary and secondary sensitivity analyses.

- The FDA Staff Manual Guide (SMG) 2020 *Quality System Framework for Internal Activities* defines the essential quality elements for management to address in internal FDA regulatory activities. SMG 2020 describes a quality system in four broad elements:

  - Say what you do
  - Do what you say
  - Prove it
  - Improve it

  Consistent with SMG 2020, the quality system described in this MAPP includes practices for quality control (QC), quality assurance (QA), and continuous quality improvement.

REFERENCES

- FDA Staff Manual Guide 2020 *Quality System Framework for Internal Activities* (available at http://www.fda.gov/smg/vol3/2000/2020.html).

DEFINITIONS

- **Product and/or Service:** The intended results of activities or processes; products or services can be tangible or intangible. The products that are governed by these quality system standards are the work products that OSE generates, not the products that FDA regulates.

- **Customer:** A person or organization (internal or external) that receives a product or service.

- **Quality:** A measure of a product’s or service’s ability to satisfy the customer’s stated or implied needs.

- **Quality Control:** Steps taken during the generation of a product or service to ensure that it meets pre-specified requirements and that the product or service is reproducible.

- **Quality Assurance:** Proactive and retrospective activities that evaluate various aspects of the product/service to ensure that the pre-specified requirements are fulfilled.

[1] Traditional components include, but are not limited to, a literature review, a discussion of the methodological issues, the study question and specific hypothesis to be addressed, the study design and time period, the data source, the study population, the outcome, the main exposure of interest, power calculation, and the selection of control or comparison group(s).

- **Continuous Improvement:** Ongoing activities to evaluate and enhance product quality, processes, and the quality system practices to increase effectiveness.

- **Analyst:** Any OSE staff or someone working on behalf of OSE (e.g., Special Government Employees or contractors) responsible, in part or in full, for the execution of any analytic plan.

- **Metadata:** A term that pertains to "data about data." Metadata essentially describe the content, context, structure, and purpose of the database and provide the reviewer/analyst with clear descriptions of the attributes of all datasets and variables.

- **Analytic Program:** A collection of instructions or codes that describes a task, or set of tasks, to be carried out by analytic software.

POLICY

- OSE will implement the quality system practices described in this MAPP when conducting analytic projects using internal or external databases or electronic data provided to the FDA in support of regulatory activities.

- OSE management or their designee(s) will determine the degree to which the analyst will apply these practices. The determination will be based on the project's purpose and scope, the complexity of the data and analytic methods, and the regulatory decisions to be made, as well as available staffing and material resources.

RESPONSIBILITIES AND PROCEDURES

- **OSE analysts** will use the quality system tools described below to plan, implement, and assess the analytic projects assigned to them.

- **OSE management** or their designee(s) will determine the level of quality assurance implementation for each assigned project. Please refer to the "Quality Assurance Practices" section in this MAPP for a description of the four levels of QA.

QUALITY CONTROL PRACTICES

A. Become Familiar with the Data

- OSE analysts will become familiar with the data to determine that the type, quantity, and quality of data are appropriate to meet the intended use or objective of the project.

1. Database Structure

- The analyst will understand, before analyzing data from any dataset:

    - the underlying database structure,
    - the methods and processes used in data collection, and
    - when applicable, how to use any custom software provided by the vendor.

  This understanding may be the result of specific previous training in a particular database or may be based on input from project team members or direct training by the supplier of the database. Such understanding will greatly enhance both the analytic process and the validity of the findings.

- The analyst will request data documentation or metadata when it does not accompany data received from sources outside FDA (such as from the sponsor of an application, for example in an NDA submission). It is essential that the analyst fully understand how the data were collected and how the database is structured so that the data can be properly analyzed.

2. Data Content

- Choice of Database — The analyst will:
    - select the most suitable database for a specific project.
    - choose the database that contains the specific data elements necessary to meet the project's objectives.
    - examine and critically evaluate the overall quality and completeness of the data.

- Documentation — The analyst will develop a description of the source data that includes as many of the following elements as applicable:
    - Name of the data source(s)
    - Name and organization of the analyst extracting the data
    - Name and organization of the analyst conducting the analysis
    - Date the data were extracted
    - Analytic data dictionary or codebook containing the following elements, as appropriate:
        - Names of the electronic files created
        - Variable(s) included in each file
        - Descriptions and format of each variable
        - Labels or descriptive text associated with each variable
        - Decode key for the value(s) of each variable, as appropriate
    - Drug class, if applicable
    - List of treatments (drugs) or procedures
    - Time period covered
    - Data-specific and project-specific population characteristics (e.g., diseases included, age range, gender)
    - Country of origin or geographic location where the data were collected
    - Any other element that is important to the specific analytic project

If a particular element is not applicable, the analyst will indicate this in the description of the source data.

- A more detailed description of the data source, such as provided by the data vendor, will be included in an appendix to the project report to help the reader understand the context of the analytic results. Examples of descriptions of several types of data sources are included in Attachment I.

- The analyst will check the data vendor's agreement on the disclosure of specific data aspects outside of FDA. Generally, only summary data may be used in reports and presentations. Special permission may be needed if members of the study team are not FDA employees. Appropriate clearance must be obtained, as specified in the agreements that govern the use of the data, before presenting project results to any individual or organization outside the FDA.

- If the analytic datasets are provided by a vendor or a sponsor from the regulated industry, the programming code used to extract the data from the original databases must be obtained from the vendor or sponsor and kept as part of the analytic report.

3. <u>Qualitative Data Verification</u>

- After downloading the data and before beginning the analysis, the analyst will perform qualitative checks on relevant data fields. For example, the range of data in all variables should be checked to identify potential outliers, inconsistencies, inaccuracies, or missing information.

- The analyst will decide how to treat missing values and determine, for example, whether missing data will be imputed or whether the entire patient's record will be deleted. The analyst will document the rationale for the decision.

- It is critical that the analyst be aware of the following variable attributes, especially when applying analytic programs that were previously developed and validated on data from a different source or with a different structure.
  - Name of variable
  - Type
  - Length
  - Label
  - Format values (to correspond with the values of the variables)

B. Build the Program

- Once the analyst is familiar with the data, he or she should build the programming code so as to achieve the objectives of the analytic project. The following subsections describe quality control practices for the analyst to use throughout the analytic process.

1. <u>Documentation of Activities</u>

- Documentation is a very important component of the analytic process. The rationale for recoding or modifying the variables contained in the original dataset should be transparent to everyone involved in the process including the primary analyst, the analyst performing the QA, and subsequent users of the programming code.

- **Overall program header**
  - The program header will appear prominently (i.e., framed in a text box) at the beginning of the program and describe the major program goals.
  - The program header will contain the following information, as applicable:
    - Study number and/or title
    - Purpose of the program, including the main objective(s) and a brief description of how the overall program flows
    - Name of the program author (or modifier if a former program was modified for secondary use)
    - Date the program was finalized[2]
    - Software (or programming language) and version
    - Level of quality assurance used (see section on Quality Assurance Practices below)
    - File name(s) of the input source data
    - Macros or routines called within the program, if any
    - External code included, if any[3]
    - Modification(s): The analyst will add the key word *Modified* and describe the revision(s), the date(s), and, ideally, the names of the modified section(s). This keyword will serve as an alert that there have been changes to the original program and can be used as an audit trail.
- **Section headers**
  - Each program might have several sections executing various analytic steps.
  - Section headers are placed above each section of coding and describe how that particular section or subsection will achieve a specific programming goal.
  - Analysts may exercise their judgment on what constitutes a section of programming based on the flow of the programming code and the tasks accomplished by that code.
- **Module and macro headers**
  - The following information will be included, as applicable:
    - Macro name
    - Purpose
    - Author(s)
    - Date created
    - Software (or programming language) and version

[2] The program is considered finalized if all the planned analyses were done and the results were given to the customer.

[3] For example, when SAS programs are called using the %INCLUDE statement. Other analytic software might have a different mechanism for executing such step.

- — Limitations pertinent to the algorithm used
- — Prerequisite(s): e.g. variables to be inputted in a specific format
- — Input parameters
- — Alerts: a description of where and why they appear

- o The analyst will archive a copy of all programs pertinent to the analytic project.

2. Program Naming Conventions

- The program name should consist of the following four components:
  - o The abbreviated name of the database
  - o The drug name (preferably chemical name), drug class, or other identifying name for the pertinent intervention being evaluated.
  - o The year the project was started
  - o A unique four-digit number that serves as a counter for the number of projects done using that database e.g., GPRD_NSAIDs_2005_0001

- When a protocol is developed in association with a particular program, the assigned name and number of the protocol should be related to those of the program.

3. Variable Naming Conventions

- The analyst will be consistent in using upper and lower case letters to name variables because some analytic software is case sensitive.[4]

- The analyst should attempt to use variable names consistent with the standards created in the Clinical Data Interchange Standards Consortium (CDISC) initiative, whenever possible. This is particularly important when the work product(s) may be shared beyond the primary analytic group. Additional information on these standards can be found at http://www.cdisc.org (accessed on 6/21/07).

- The analyst should consider designating standards for naming the working (i.e. temporary memory) variables and certain routine computations, such as totals and arrays. For example, one might begin the memory variable with a WV_ (e.g., "WV_drugtype") to indicate it is temporary and is not a dataset variable. Totals might be designated with a TV_ (e.g., "TV_Therapies"). Similarly, array names might begin with an AV_ (e.g., AV_annualrxcount"). In addition, temporary datasets might be identified as WDS_(e.g., "WDS_ NameOfDataset").

4. Writing Programming Codes

- Analysts performing and/or reviewing data management and analysis will have appropriate training or prior experience in the use of the particular analytic software. For example, analysts performing data management and analysis using SAS ideally should be SAS Certified Professionals, or have SAS training appropriate to the complexity of their assigned

[4] For example, in SAS the analyst might avoid potential for confusion by including the "option validvarname=upcase;" early in the program to force translation of all letters of variable names to upper case.

analytic tasks. Vendors of other software, e.g., STATA, offer similar training programs for interested analysts.

- Analysts will incorporate, to the extent possible, the following practices as they write their programs:

    - Build the code in sections and design an efficient algorithm to achieve tasks of each section. The program flow should correspond to the goals specified in the analytic plan.

    - Create time-saving macros whenever possible and make them available for re-use by other analysts.

    - Whenever possible, use code, macros, or modules that have been used, tested, and validated in previous projects. Such coding is particularly useful in performing routine steps involved in cleaning and preparing the final analytic datasets.

    - Redefine character variables as numeric ones, whenever feasible, to maximize computing efficiency and minimize the disk space required for storage.

    - Write only one source code statement per line to improve legibility.

    - Indent programming statements within a major procedure or step to make them visually stand out to the QA reviewer or future users.

    - Store major sections of the programming code in different files.

    - Specify the name of the dataset used in each analytic step, even if the analytic software does not require it, to facilitate QA.

    - Thoroughly annotate all programs with comments that clearly describe the intent or purpose of every step. This will communicate the program's processes and will facilitate the QA steps at a later point. Examples of annotated programs appear in Attachment II.

    - Check the format and spell-check the labels used in the program before running and printing the final results.

    - Use the analytic software to produce the tabulation of results for the final report rather than putting the results tables together manually. The former approach eliminates the potential for error in transferring the results data.

5. Testing Coding Syntax

- The analyst will review each programming step to ensure the accuracy of the coding syntax. Most statistical software packages alert analysts to potential syntax errors during the parsing process. For example, the SAS software has three different types of alert messages: errors, warnings, and notes. Note that automatic features in such software might attempt to resolve problems before entirely stopping the program. Therefore, even if a program executes, the analyst should always review all types of messages noted in the log. Similar alerts are available in other statistical software, such as STATA, ORACLE, C++, or S/R.

6. Testing Coding Logic

- The implementation of proper coding logic is one of the most critical aspects of analytic programming. Even the most advanced software tools cannot alert the analyst to logic error. Therefore, it is strongly recommended that the analyst implement a proactive approach to detect and resolve logic errors.

- Following each coding step, the analyst will evaluate the number of observations and the number of variables to ensure that they are consistent with the intended purpose of the step.

- In those databases for which it is possible, e.g., the UK-based General Practice Research Database (GPRD), the analyst should select a sample of patient profiles and manually review them to verify the accuracy of data resulting from coding or re-coding of data.

- The coding logic can be tested as a part of the Continuous QA practices (see the section on QA Practices below). After each section of coding, members of the project team who were not involved in writing the code for data management can manually review a pseudorandom computer-generated sample of the raw data (e.g., 100 records) to verify that the intended goal of data management has been accurately achieved.

- The same steps can be implemented as part of a Partial Sign-off QA practice (see the section on QA Practices below).

7. Modifying The Program

- After the analytic project is finished, the analyst will save the program, log listings, and output as a PDF file, which he/she can then provide to the customer.

- Any changes made to the program after this point in the process will be recorded as a modification in the coding.

    - As described above, an indicator will be added to the overall program header to denote the modification using the key word *Modified*.

    - The original code that has been changed in the program will not be removed; rather, it will be retained in the program as an inactive comment.

    - The modification date and the reason for the modification will be recorded.

C. Archive the Project Data and Documentation

- The analyst will archive the original data, the program, the log files (when applicable), and the output files from any program. The archive will be placed in a specified computer shared drive determined by OSE, when available, for access to other analysts. A backup copy will be retained in an electronic format.

- The analyst will create a table that contains a summary of all programs used in the analysis, including a brief description of each program's function.

QUALITY ASSURANCE PRACTICES

Levels of Quality Assurance (QA) — This section describes the four different levels of QA practices that comprise a flexible quality assurance system. Management of OSE or of the division responsible for the analytic product will select the level of QA that is most appropriate for the specific project, contingent upon available resources. The use of a test data set, whenever applicable and feasible, may be considered, especially for analytic code previously used and validated.

- **Strategic QA (SQA):** A level of QA verification that can be achieved during the conduct of the work by following the QC practices for analytic projects described in this document.

- **Continuous QA (CQA):** A continuous verification effort by the analyst to check and resolve any errors in the process.

    - To achieve this goal, a sample of the original records (the raw data) is examined to evaluate how the data are transformed by the program code. An appropriately sized random sample that illustrates several possible scenarios in the data should be used, not just the first few records.
    - Following each coding step, the number of observations and the number of variables are evaluated to ensure they match the intended purpose of each step.

- **Partial Sign-off QA:** A verification conducted by a second, independent reviewing analyst on a sample of the work to evaluate the effectiveness of the QC practices. The reviewing analyst must examine the coding and compare the original raw data to the transformed data to ensure that the program accomplishes the intended purpose of the analysis.

- **Full sign-off QA:** A verification effort in which the entire analytic project, including the programming code, is reviewed by a second, independent analyst. The following steps comprise full sign-off:

    - Management assigns both a primary analyst (PA) and a reviewing analyst (RA) to the project at the outset.

    - The PA develops and tests the program based on the QC and the SQA and CQA practices described in this document. The analytic plan, the final program, and the analytic results, along with pertinent documentation, are then provided to the RA.

    - The RA then validates the results by reviewing all materials for validity, accuracy, completeness, and sound logic. If the RA finds that some or all of the coding is questionable, then the RA may create and run a different coding scheme to reproduce and verify the findings of the analytic project.

    - If a discrepancy exists, the PA and RA need to reconcile the differences.

CONTINUOUS IMPROVEMENT PRACTICES

- OSE management or their assigned teams will conduct periodic evaluations and make recommendations to expand and improve the quality system practices.

EFFECTIVE DATE

This MAPP is effective upon date of publication.

Attachment I: Sample Database Descriptions

Verispan, LLC: Vector One: National (VONA)

Verispan's VONA measures retail dispensing of prescriptions or the frequency with which drugs move out of retail pharmacies into the hands of consumers via formal prescriptions. Information on the physician specialty, the patient's age and gender, and estimates for the numbers of patients that are continuing or new to therapy are available.

The Vector One database integrates prescription activity from a variety of sources including national retail chains, mass merchandisers, mail order pharmacies, pharmacy benefits managers and their data systems, and provider groups. Vector One receives over 2 billion prescription claims, representing over 160 million unique patients.

Prescriptions are captured from a sample of approximately 54,000 pharmacies throughout the US. The pharmacies in the data base account for nearly all retail pharmacies and represent approximately 50% of retail prescriptions dispensed nationwide. Verispan receives all prescriptions from approximately one-third of the stores and a significant sample of prescriptions from the remaining stores.

Premier

Premier's database is a large hospital drug utilization and financial database. Information is available from over 450 acute care and pediatric facilities and includes approximately 16 million inpatient records. On an annual basis, this constitutes roughly one out of every seven inpatient discharges in the United States. Data are available from January 2000 through the present, but have a lag time of approximately 6 months. Premier's primary mission is to assist health care institutions in improving clinical and operating performance in three strategic areas: group purchasing, supply chain, and healthcare informatics. To that end, the Premier Informatics group developed this database in part to analyze utilization of resources to improve clinical efficiency.

The hospitals that contribute information to this database are a select sample of both Premier and U.S. institutions, and do not necessarily represent all hospitals in the U.S. Data are collected from this sample of participating hospitals with diverse characteristics based upon geographic location, bed size, population served, payers, and teaching status. The data collected include demographic and pharmacy-billing information, as well as all diagnoses and procedures for every patient discharge. Preliminary comparisons between participating Premier hospital and patient characteristics and those of the probability sample of hospitals and patients selected for the National Hospital Discharge Survey (NHDS) proved to be very similar with regard to patient age, gender, length of stay, mortality, primary discharge diagnosis, and primary procedure groups. Based upon these analyses, FDA believes that most estimates of national inpatient drug use based on Premier data appear to be reasonable, but strongly recommends making this determination on a drug-specific basis.

The General Practice Research Database (GPRD)

The General Practice Research Database (GPRD) began in the late 1980s as a way for general practitioners (GP) in the United Kingdom (U.K.) to both computerize their practices and contribute to public health research. The database contains detailed longitudinal information on symptoms, diagnoses, prescriptions, laboratory tests, immunizations, and hospital referral, as well as basic demographics for a sample of just over 5% of all U.K. patients registered with a GP. Outpatient care as well as summary information related to hospital and specialist visits are included in the medical records.

The data contained in GPRD represent the patient's entire medical record and are available in a coded and searchable format. In the U.K., the GP is considered the gatekeeper for access to non-emergency healthcare services. Because of this, the GPRD record tends to be a more complete picture of patient's health history than would be available in other databases, e.g. claims databases. Currently, data are being collected on over 3.4 million active patients (approx. 13 million total) from around 450 primary care practices throughout the U.K. The database is owned by the U.K,'s Department of Health and managed by an operating unit of the U.K. government.

Information on date of death can be found within the database and will soon be available from vital statistics. It is also possible to obtain additional information on patients by either reviewing anonymized versions of free text comments recorded in association with many of the events in the database or by sending a questionnaire back to the GPs. This can include patient contact with the appropriate approvals.

The database has been the subject of many validation studies and entry of the data to the research available database is under established quality control procedures administered by the GPRD staff. Several hundreds of published papers used data from the GPRD. By the end of 2007 GPRD is expected to become externally record linkable, meaning that more detailed data on hospitalizations and from specific disease registries can be linked. This will include links to devices databases such as in orthopedics. Interventional studies including genetic samples can also be arranged via the GPRD team.

Attachment II: Sample Annotated Analytic Code

A] Sample annotated SAS program of fictional data

```
Options nodate pageno=1 ls=80 ps=50 ; * Set options for work
environment;

* Set Library names;
libname baby 'N:\THammad\GPRD Mother-Baby\baby'; * Location of babies'
data;
libname mother 'N:\THammad\GPRD Mother-Baby\mother'; * Location of
mothers' data;
libname codes 'N:\THammad\AntiDepressants-PPHN\DictionaryCodes'; *
Location of GPRD codes extracted from Medical and Product dictionaries;


* The next section cleans the redundancy in the codes files;
proc sort data=codes.pphncodes  out=codes.c_pphncodes NODUPKEY; by
gprdmedcode; * Outcome related events;
proc sort data=codes.gestation  out=codes.c_gestation NODUPKEY; by
gprdmedcode; * Gestation related events;
RUN;


* Importing data about the link between mothers and babies;
PROC IMPORT OUT= WORK.Link
            DATAFILE= "N:\THammad\GPRD Mother-
Baby\Mother_Baby_Link_21Oct2005.txt"
            DBMS=TAB REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
```

B] Sample annotated SAS macro of fictional data

```
***********************************************************************
*       Macro name: Date()
*       Purpose: Read and reconstruct SAS dates
*       Author: Tarek Hammad, MD, PhD
*       Date: September 26, 2005
*       SAS version: 9.1
*       Limitations: NA
*       Prerequisite: Variable containing date information is read as
*       text
*       Input parameters: datevar=variable with the original date
*       Alerts: NA
**********************************************************************;

%Macro Date(datevar);
      Day=substr(&datevar,1,2);      * Reads the day;
      Month=substr(&datevar,4,2);    * Reads the month;
      Year=substr(&datevar,7,4);     * Reads the year;
      Date=MDY(month,day,year);     * Create a date from a day, month,
      year variables;
      O&datevar=&datevar;     * Keeps the old variable for validation;
      &datevar=Date;          * Rename the new date;
      F&datevar=put(Date, ddmmyy10.); * Formats the new date variable;
%Mend Date;
```

C] Sample annotated STATA program of fictional data

* PREPARING DATA

```
* Change directory to area of work
cd C:\Data-ODS\New-DNDP\Article-children

* MUST OPEN A LOG FILE TO CAPTURE THE OUTCOME OF THE ANALYSIS
log using "C:\Stata data\Children\Any.log", replace

clear

insheet using " C:\Stata data\Children\Any.txt"

*rename variables
rename v1 StudyID
rename v2 RxTotal
rename v3 CtrTotal
rename v4 RxFail
rename v5 CtrFail
rename v6 Drug
rename v7 DrugGrp
rename v8 Diagnosis


*label variables
label variable RxFail "Cell a in 2x2"
label variable RxTotal "Row total N1 in 2x2"
label variable CtrFail "Cell c in 2x2"
label variable CtrTotal "Row total N2 in 2x2"

* Define values of a format
label define Group 1 "SSRI" 2 "Atypical"

* Assign format
label values DrugGrp Group

*calculate cells "b" and "d" in the 2x2 table
gen RxCure= RxTotal- RxFail
gen CtrCure= CtrTotal- CtrFail
label variable RxCure "Cell b in 2x2"
label variable CtrCure "Cell d in 2x2"

*drop trials with no events
drop if RxFail==0&CtrFail==0

*calculate OR, logOR, seLogOR, & 95% CI
```

```
gen or=((RxFail*CtrCure)/( RxCure*CtrFail))
gen logor=log(( RxFail*CtrCure)/( RxCure*CtrFail))
gen seLogOR = sqrt((1/ RxFail)+(1/ RxCure)+(1/ CtrFail)+(1/ CtrCure))
gen CiLow= exp(logor-1.96*seLogOR)
gen CiUpp= exp(logor+1.96*seLogOR)
label variable or "Odds Ratio"
label variable logor "Log of Odds Ratio"
label variable seLogOR "Standard Error, Log OR"
label variable CiLow "Lower Bound, 95% CI"
label variable CiUpp "Upper Bound, 95% CI"

Save "summary.dta", replace

            * META-ANALYSIS

******* Using all trials including all indications ***********
use using summary.dta
metan RxFail RxCure CtrFail CtrCure, rr fixed
sortby(StudyID)label(namevar=StudyID) boxsca(1) xlabel(.01,.1,1,10,100)
t1(All trials, all indications) t2((Fixed effect model)) b1(Suicidal
Behavior or Ideation) texts(3)
graph save AnyALL, asis replace
```

D] Sample annotated R program of fictional data

```
###-------------------------------------------------------------------
### R code used to create a barplot of change grouped by treatment and
age cohort.
### Author: Mat Soukup
### Date: 06/04/2007
###-------------------------------------------------------------------

# Location where graph will be written
figpath <- 'C:/Data/DrugName/figures'

# Create a random data set of change for each cohort listed by
treatment group
set.seed(123)
change <- matrix(sample(1:100, 10), ncol=2)
    rownames(change) <-
c('[2,18)','[18,30)','[30,45)','[45,60)','[60,80]') # Age Cohort
    colnames(change) <- c('Active','Control') # Treatment group

# Open a new graphics device in which the graphic will be created
win.metafile(file=paste(figpath, 'barplot.wmf', sep=''),
             width=11, height=8)
barplot(change, beside=TRUE,
        col=c('lightblue', 'skyblue2', 'blue', 'navyblue'),
        main='R Barplot',
        xlab='Treatment Group',
        ylab='Change from Baseline',
        legend.text=TRUE)
```

```
dev.off()
# To view the graphic open the barplot.wmf in the location where the
graph was written
### End file
```