Analysis Data Reviewer's Guide

R Consortium

R Submission Pilot 3

ADRG Template Version 2019-07-18

Analysis Data Reviewer's Guide

Contents

1. Introduction

1.1 Purpose

This document provides context for the analysis datasets and terminology that benefit from additional explanation beyond the Data Definition document (define.xml). In addition, this document provides a summary of ADaM conformance findings.

1.2 Study Data Standards and Dictionary Inventory

Standard or Dictionary	Versions Used
SDTM	SDTM Implementation Guide Version 3.1.2 SDTM Version 1.2
SDTM Controlled Terminology	CDISC SDTM Controlled Terminology, 2022-12-16
ADaM	ADaM-IG v1.1 ADaM v2.1
ADaM Controlled Terminology	CDISC ADaM Controlled Terminology, 2022-06-24
Data Definitions	Define-XML v2.0
Medical Events Dictionary	MedDRA version 8.0

1.3 Source Data Used for Analysis Dataset Creation

The ADaM datasets were derived from SDTM version 1.2. For traceability, the SDTM is publicly available at the PHUSE Github Repository :

> https://github.com/cdisc-org/sdtm-adam-pilot-project/tree/master/updated-pilot-submission-package/900172/m5/datasets/cdiscpilot01/tabulations/sdtm

Which can be traced back to the original CDISC SDTM & ADaM Pilot Project.

> https://github.com/cdisc-org/sdtm-adam-pilot-project

2. Protocol Description

2.1 Protocol Number and Title

Protocol Number: CDISCPilot1

Protocol Title: Safety and Efficacy of the Xanomeline Transdermal Therapeutic System (TTS) in Patients with Mild to Moderate Alzheimer's Disease.dummy

The reference documents can be found at

https://github.com/cdisc-org/sdtm-adam-pilot-project/blob/master/updated-pilot-submission-package/900172/m5/53-clin-stud-rep/535-rep-effic-safety-stud/5351-stud-rep-contr/cdiscpilot01/cdiscpilot01.pdf

2.2 Protocol Design in Relation to ADaM Concepts

Objectives:

The objectives of the study were to evaluate the efficacy and safety of transdermal xanomeline, 50cm and

75cm, and placebo in subjects with mild to moderate Alzheimer's disease.

Methodology:

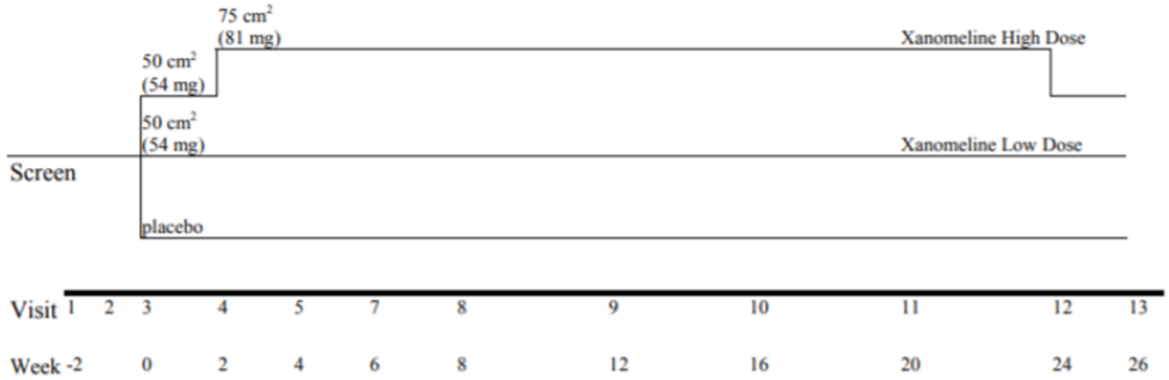
This was a prospective, randomized, multi-center, double-blind, placebo-controlled, parallel-group study.

Subjects were randomized equally to placebo, xanomeline low dose, or xanomeline high dose. Subjects applied 2 patches daily and were followed for a total of 26 weeks.

Number of Subjects Planned:

300 subjects total (100 subjects in each of 3 groups)

Study schema:



3. Analysis Considerations Related to Multiple Analysis Datasets

3.1 Core Variables

Core variables are those that are represented across all/most analysis datasets.

Variable Name	Variable Description
STUDYID	Study Identifier
USUBJID	Unique Subject Identifier
SUBJID	Subject Identifier for the Study
SITEID	Study Site Identifier
SITEGR1	Pooled Site Group 1
TRTSDT	Date of First Exposure to Treatment
TRTEDT	Date of Last Exposure to Treatment
AGE	Age
AGEGR1	Pooled Age Group 1
AGEGR1N	Pooled Age Group 1 (N)
RACE	Race
RACEN	Race (N)
SEX	Sex

Variable Name	Variable Description
SAFFL	Safety Population Flag
ITTFL	Intent-To-Treat Population Flag
EFFFL	Efficacy Population Flag
COMP24FL	Completers of Week 24 Population Flag
DSRAEFL	Discontinued due to AE?

3.2 Treatment Variables

ARM versus TRT01P

Are the values of ARM equivalent in meaning to values of TRT01P?

Yes.

ACTARM versus TRT01A

If TRT01A is used, then are the values of ACTARM equivalent to values of TRT01A?

Not applicable - ACTARM is not used.

Use of ADaM Treatment Variables in Analysis

Are both planned and actual treatment variables used in analysis?

Yes. Planned treatment variables are used for study population and efficacy analyses, whilst actual treatment variables are used for the safety analysis. All subjects received the treatment arm to which they were randomised and so the planned treatment is equivalent to the actual treatment for all subjects.

Use of ADaM Treatment Grouping Variables in Analysis

Are both planned and actual treatment grouping variables used in analysis?

Not applicable - treatment grouping variables are not used.

3.3 Use of Visit Windowing, Unscheduled Visits, and Record Selection

Was windowing used in one or more analysis datasets?

Yes

Were unscheduled visits used in any analyses?

Yes

3.4 Imputation/Derivation Methods

For ASTDT in ADAE, this date was converted to numeric SAS date from AE.AESTDTC. If the day component is missing, a value of '01' is used. If both the month and day are missing no imputation is performed. See define.xml.

4. Analysis Data Creation and Processing Issues

4.1 Split Datasets

There were no datasets that required splitting due to size constraints.

4.2 Data Dependencies

Analysis Dataset	Dependent on Following Analysis Datasets
ADAE	ADSL
ADTTE	ADSL, ADAE
ADLBC	ADSL
ADADAS	ADSL

4.3 Intermediate Datasets

No intermediate datasets were created for this trial.

5. Analysis Dataset Descriptions

5.1 Overview

The following provides detailed information for each analysis dataset included in the Pilot 3 submission, which were used to generate the outputs in Pilot 1. These ADaM datasets are ADSL, ADAE, ADTTE, ADADAS, ADLBC.

5.2 Analysis Datasets

Dataset - Dataset Label	Class	Efficacy	Safety	Baseline or other subject characteristics	Primary Objective	Structure
ADSL - Subject-Level Analysis Dataset	SUBJECT LEVEL ANALYSIS DATASET			x		One record per subject
ADADAS - ADAS-COG Analysis Dataset	BASIC DATA STRUCTURE	x			x	One or more records per subject per analysis parameter per analysis timepoint
ADAE - Adverse Events Analysis Dataset	OCCURRENCE DATA STRUCTURE		x			One record per subject per adverse event
ADLBC - Analysis Dataset Lab Blood Chemistry	BASIC DATA STRUCTURE		x			One or more records per subject per analysis parameter per analysis timepoint

Dataset - Dataset Label	Class	Efficacy	Safety	Baseline or other subject characteristics	Primary Objective	Structure
ADTTE - AE Time To 1st Derm. Event Analysis	BASIC DATA STRUCTURE	x	x			One or more records per subject per analysis parameter per analysis timepoint

5.2.1 ADSL - Subject-Level Analysis Dataset

The subject level analysis dataset (ADSL) contains required variables for demographics, treatment groups, and population flags. In addition, it contains other baseline characteristics that were used in both safety and efficacy analyses. All patients in DM were included in ADSL. The following are the key population flags are used in analyses for patients:

- SAFFL – Safety Population Flag (all patients having received any study treatment)
- ITTFL – Intent-to-Treat Population Flag (all randomized patients)

5.2.2 ADADAS - ADAS-COG Analysis Dataset

ADADAS contains analysis data from the ADAS-Cog questionnaire, one of the primary efficacy endpoints. It contains one record per subject per parameter (ADAS-Cog questionnaire item) per VISIT. Visits are placed into analysis visits (represented by AVISIT and AVISITN) based on the date of the visit and the visit windows.

5.2.3 ADAE - Adverse Events Analysis Dataset

ADAE contains one record per reported event per subject. Subjects who did not report any Adverse Events are not represented in this dataset. The data reference for ADAE is the SDTM AE (Adverse Events) domain and there is a 1-1 correspondence between records in the source and this analysis dataset. These records can be linked uniquely by STUDYID, USUBJID, and AESEQ. Events of particular interest (dermatologic) are captured in the customized query variable (CQ01NAM) in this dataset. Since ADAE is a source for ADTTE, the first chronological occurrence based on the start dates (and sequence numbers) of the treatment emergent dermatological events are flagged (AOCC01FL) to facilitate traceability between these two analysis datasets.

5.2.4 ADLBC - Analysis Dataset Lab Blood Chemistry

ADLBC contains one record per lab analysis parameter, per time point, per subject. ADLBC contains lab chemistry parameters and these data are derived from the SDTM LB (Laboratory Tests) domain. Two sets of lab parameters exist in ADLBC. One set contains the standardised lab value from the LB domain and the second set contains change from previous visit relative to normal range values. In some of the summaries the derived end-of-treatment visit (AVISITN=99) is also presented.

5.2.5 ADTTE - AE Time To 1st Derm. Event Analysis

ADTTE contains one observation per parameter per subject. ADTTE is specifically for safety analyses of the time to the first dermatologic adverse event. Dermatologic AEs are considered an adverse event of special interest. The key parameter used for the analysis of time to the first dermatological event is with PARAMCD of "TTDE".

6. Data Conformance Summary

6.1 Conformance Inputs

4.4 ● Were the analysis datasets evaluated for conformance with CDISC ADaM Validation Checks?

4.5 Yes, Version of CDISC ADaM Validation Checks and software used: Pinnacle 21® Community 4.0.1

4.6 ● Were the ADaM datasets evaluated in relation to define.xml?

4.7 Yes

4.8 ● Was define.xml evaluated?

4.9 Yes

6.2 Issues Summary

Check ID	Diagnostic Message	Dataset	Count (Issue Rate)	Explanation
AD1012	Secondary custom variable is present but its primary variable is not present	ADSL	1 (50.00%)	This is a Sponsor Extension to the ADaM Model. The VISNUMEN [End of Trt Visit (Vis 12 or Early Term.)] variable is a integer variable which is not related to any character variable.

6.3 QC Findings and Common Issues

In this Pilot 3 study, our focus was to create ADaMs based on Pilot 1 where after ADaM generation we compared them against the analysis datasets used in Pilot 1 as a QC step. With

these comparisons we listed the QC Findings with explanations as to why these findings exist. We also came across common issues throughout the ADaM generation process, which could be helpful for improvements utilising the CDISC Pilot data in the future. More details can be found in the appendix.

https://github.com/RConsortium/submissions-pilot3-adam/wiki/QC-Findings

https://github.com/RConsortium/submissions-pilot3-adam/wiki/Common-Issues

7. Submission of Programs

7.1 Description

The sponsor has provided all programs for analysis results. They are all created on a Linux platform using R version 4.2.3.

7.2 ADaM Programs

The following table contains the list of programs that generate the analysis datasets in the R Consortium R submission Pilot 3. It shows the program file name, the analysis dataset name and the label of the analysis dataset. The recommended steps to execute the analysis results using R are described in the Appendix.

Program Name	Analysis Dataset Name	Analysis Dataset Label
adsl.r	adsl.xpt	Subject-Level Analysis Dataset
adadas.r	adas.xpt	ADAS-Cog Analysis
ad_adlbc.r	adlb.xpt	Analysis Dataset Lab Blood Chemistry
ad_adae.r	adae.xpt	Adverse Events Analysis Dataset
ad_adtte.r	adtte.xpt	AE Time to 1st Derm. Event Analysis

7.3 Analysis Output Programs

The following table contains a list of programs that generate outputs used in the R consortium R submission Pilot 1. These outputs were rerun in Pilot 3 using the analysis datasets generated by the ADaM programs. It shows the program file names, the related outputs, the input datasets and variables used, and any data selection criteria that need to be applied per Pilot 1.

Program Name	Output Name	Analysis Datasets & Variables	Selection Criteria
tlf-demographic.r	tlf-demographic-pilot3.out	ADSL.STUDYID ADSL.TRT01P ADSL.ITTFL ADSL.AGE ADSL.AGEGR1 ADSL.RACE ADSL.HEIGHTBL ADSL.WEIGHTBL ADSL.BMIBL ADSL.MMSETOT	STUDYID== "CDISCPILOT01" Population: ADSL.ITTFL == "Y" Treatment Groups: ADSL.TRT01P Placebo Xanomeline Low Dose Xanomeline High Dose
tlf-primary.r	tlf-primary-pilot3.rtf	ADSL.TRT01P ADSL.USUBJID ADSL.EFFFL ADSL.ITTFL ADADAS.TRTP ADADAS.TRTPCD ADADAS.EFFFL ADADAS.ITTFL ADADAS.PARAMCD ADADAS.ANL01FL ADADAS.AVISIT ADADAS.AVISITN ADADAS.AVAL ADADAS.CHG	STUDYID== "CDISCPILOT01" Population: ADADAS.EFFFL == "Y" ADADAS.ITTFL == "Y" ADADAS.ANL01FL == "Y" Treatment Groups: ADSL.TRTP Placebo Xanomeline Low Dose Xanomeline High Dose Parameters: ADADAS.PARAMCD == "ACTOT

Program Name	Output Name	Analysis Datasets & Variables	Selection Criteria
tlf-efficacy.r	tlf-efficacy-pilot3.rtf	ADSL.STUDYID ADSL.USUBJID ADSL.ITTFL ADLBC.TRTP ADLBC.TRTPN ADLBC.PARAMCD ADLBC.AVISITN ADLBC.BASE ADLBC.AVAL ADLBC.CHG	STUDYID== "CDISCPILOT01" Population: ADSL.ITTFL == "Y" & ADLBC.TRTPN in (0, 81) & ADLBC.PARAMCD == "GLUC" & ADLBC.AVISITN is not missing Treatment Groups: ADLBC.TRTPN Placebo Xanomeline High Dose
tlf-kmplot.r	tlf.kmplot-pilot3.pdf	ADSL.STUDYID ADSL.USUBJID ADSL.SAFFL ADSL.TRT01A ADTTE.STUDYID ADTTE.USUBJID ADTTE.PARAMCD ADTTE.AVAL ADTTE.CNSR	STUDYID== "CDISCPILOT01" Population: ADSL.SAFFL == "Y" Treatment Groups: ADSL.TRT01A Placebo Xanomeline Low Dose Xanomeline High Dose Parameters: ADTTE.PARAMCD == "TTDE"

For reference, below is a description of the analysis programs utilized and outputs generated in Pilot 1.

Program Name	Output Table Number	Title
tlf-demographic.r	Table 14-2.01	Summary of Demographic and Baseline Characteristics
tlf-primary.r	Table 14-3.01	Primary Endpoint Analysis: ADAS Cog (11) - Change from Baseline to Week 24 - LOCF
tlf-efficacy.r	Table 14-3.02	ANCOVA of Change from Baseline at Week 20
tlf-kmplot.r	Figure 14-1	KM plot for Time to First Dermatologic Event: Safety population

7.4 Proprietary R Packages

Proprietary R Package	Package version	Analysis Package Description
Pilot3	0.1.1	The objective of this utility package is to support the R Consortium R submission Pilot 3 Project. It contains all utility functions that were used in the generation of the deliverables: ● formatting of ADaM variables and analysis results ● summarize mixed model analysis ● formatting of layouts

7.5 Open-source R Packages

Open-source R Analysis Package	Package version	Analysis Package Description
admiral	0.10.1	A toolbox for programming Clinical Data Interchange Standards Consortium (CDISC) compliant Analysis Data Model (ADaM) datasets in R. ADaM datasets are a mandatory part of any New Drug or Biologics License Application submitted to the United States Food and Drug Administration (FDA). Analysis derivations are implemented in accordance with the "Analysis Data Model Implementation Guide" (CDISC Analysis Data Model Team, 2021, <https://www.cdisc.org/standards/foundational/adam/adamig-v1-3-release-package>).
cowplot	1.1.1	Provides various features that help with creating publication-quality figures with 'ggplot2', such as a set of themes, functions to align plots and arrange them into complex compound figures, and functions that make it easy to annotate plots and or mix plots with images. The package was originally written for internal use in the Wilke lab, hence the name (Claus O. Wilke's plot package). It has also been used extensively in the book Fundamentals of Data Visualization.

diffdf	1.0.4	Functions for comparing two data.frames against each other. The core functionality is to provide a detailed breakdown of any differences between two data.frames as well as providing utility functions to help narrow down the source of problems and differences.
dplyr	1.1.0	A fast, consistent tool for working with data frame like objects, both in memory and out of memory.
emmeans	1.8.5	Obtain estimated marginal means (EMMs) for many linear, generalized linear, and mixed models. Compute contrasts or linear functions of EMMs, trends, and comparisons of slopes. Plots and other displays. Least-squares means are discussed, and the term "estimated marginal means" is suggested, in Searle, Speed, and Milliken (1980) Population marginal means in the linear model: An alternative to least squares means, The American Statistician 34(4), 216-221 <doi:10.1080/00031305.1980.10483031>.
ggplot2	3.4.1	A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.
haven	2.5.2	Import foreign statistical formats into R via the embedded 'ReadStat' C library, <https://github.com/WizardMac/ReadStat>.
lubridate	1.9.2	Functions to work with date-times and time-spans: fast and user friendly parsing of date-time data, extraction and updating of components of a date-time (years, months, days, hours, minutes, and seconds), algebraic manipulation on date-time and time-span objects. The 'lubridate' package has a consistent and memorable syntax that makes working with dates easy and fun.
metacore	0.1.2	Create an immutable container holding metadata for the purpose of better enabling programming activities and functionality of other packages within the clinical programming workflow.

metatools	0.1.5	Uses the metadata information stored in 'metacore' objects to check and build metadata associated columns.
pharmaRTF	0.1.4	Enhanced RTF wrapper written in R for use with existing R tables packages such as 'Huxtable' or 'GT'. This package fills a gap where tables in certain packages can be written out to RTF, but cannot add certain metadata or features to the document that are required/expected in a report for a regulatory submission, such as multiple levels of titles and footnotes, making the document landscape, and controlling properties such as margins.
pilot3	0.1.1	Utilities for the Pilot 3 Submission to the FDA. See section 7.4.
r2rtf	1.0.1	Create production-ready Rich Text Format (RTF) table and figure with flexible format.
rtables	0.6.0	Reporting tables often have structure that goes beyond simple rectangular data. The 'rtables' package provides a framework for declaring complex multi-level tabulations and then applying them to data. This framework models both tabulation and the resulting tables as hierarchical, tree-like objects which support sibling sub-tables, arbitrary splitting or grouping of data in row and column dimensions, cells containing multiple values, and the concept of contextual summary computations. A convenient pipe-able interface is provided for declaring table layouts and the corresponding computations, and then applying them to data.
stringr	1.5.0	A consistent, simple and easy to use set of wrappers around the fantastic 'stringi' package. All function and argument names (and positions) are consistent, all functions deal with "NA"'s and zero length vectors in the same way, and the output from one function is easy to feed into the input of another.

tidyr	1.3.0	Tools to help to create tidy data, where each column is a variable, each row is an observation, and each cell contains a single value. 'tidyr' contains tools for changing the shape (pivoting) and hierarchy (nesting and 'unnesting') of a dataset, turning deeply nested lists into rectangular data frames ('rectangling'), and extracting values out of string columns. It also includes tools for working with missing values (both implicit and explicit).
Tplyr	1.1.0	A traceability focused tool created to simplify the data manipulation necessary to create clinical summaries.
visR	0.3.1	To enable fit-for-purpose, reusable clinical and medical research focused visualizations and tables with sensible defaults and based on graphical principles as described in: "Vandemeulebroecke et al. (2018)" <doi:10.1002/pst.1912>, "Vandemeulebroecke et al. (2019)" <doi:10.1002/psp4.12455>, and "Morris et al. (2019)" <doi:10.1136/bmjopen-2019-030215>.
xportr	0.2.0	Tools to build CDISC compliant data sets and check for CDISC compliance.

8 Directory Structure

```
m5
└── datasets
    └── rconsortiumpilot3
        ├── tabulations
        │   ├── sdtm
        │   │   ├── ae.xpt                        # SDTM datasets in XPT format
        │   │   ├── cm.xpt
        │   │   ├── dm.xpt
        │   │   ├── ds.xpt
        │   │   ├── ex.xpt
        │   │   ├── lb.xpt
        │   │   ├── mh.xpt
        │   │   ├── qs.xpt
        │   │   ├── relrec.xpt
        │   │   ├── sc.xpt
        │   │   ├── se.xpt
        │   │   ├── suppae.xpt
        │   │   ├── suppdm.xpt
        │   │   ├── suppds.xpt
        │   │   ├── supplb.xpt
        │   │   ├── sv.xpt
        │   │   ├── ta.xpt
        │   │   ├── te.xpt
        │   │   ├── ti.xpt
        │   │   ├── ts.xpt
        │   │   ├── tv.xpt
        │   │   └── vs.xpt
        └── analysis
            ├── adam
            │   ├── ADaM - Pilot 3.xlsx
            │   ├── define.xml
            │   ├── adadas.xpt                     # ADaM datasets in XPT format
            │   ├── adae.xpt
            │   ├── adlbc.xpt
            │   ├── adsl.xpt
            │   ├── adtte.xpt
            │   └── define2-0-0.xsl
            ├── output
            │   ├── tlf-demographic-pilot3.out     # output files for TLFs.
            │   ├── tlf-efficacy-pilot3.rtf
            │   ├── tlf-kmplot-pilot3.pdf
            │   └── tlf-primary-pilot3.rtf
            └── programs
                ├── adadas.R                       # analysis R code for TLFs.
                ├── adae.R
                ├── adlbc.R
                ├── adsl.R
                ├── adtte.R
                ├── tlf-demographic.R
                ├── tlf-efficacy.R
                ├── tlf-kmplot.R
                ├── tlf-primary.R
                └── renv.lock
```

4.10 Appendix

Appendix 1: Pilot 3 Installation and Usage

To install and execute the R programs, follow all of the procedures below. Ensure that you note the location of where you downloaded the Pilot 3 eCTD submission files. For demonstration purposes, the procedures below assume the transfer has been saved to this location: C:\pilot3.

In addition, create a new directory to hold the unpacked Pilot 3 ADaM and tlf programs and files. For demonstration purposes, the procedures below assume the new directory is this location: C:\pilot3\pilot3-files.

1. **Installation of R and R Studio**

Download and install R 4.2.3 for Windows from https://cran.r-project.org/bin/windows/base/old/4.2.3/ . Then download and run the R-4.2.3-win.exe file. Also download RStudio for Windows by visiting https://dailies.rstudio.com/version/2023.03.1+446.pro1/

2. **Installation of R Packages**

A minimum set of R packages are required to ensure the Pilot 3 analysis programs are successfully run and the custom package environment used for the application is replicated correctly. The first packages to install are the remotes and renv packages:

```
install.packages(c("pak", "renv"))
```

Note : (1) *The console may display a warning message about Rtools being required to build R packages. However the Rtools utility is not required to run the programs in this pilot 3 study.* (2) *If you receive a warning showing "*`cannot open URL 'https://cran.rstudio.com/src/contrib/PACKAGES'`*", this is due to the default R Studio option 'Use secure download method for HTTP'. In R Studio, go to Tools → Global Options → Packages, then uncheck the 'Use secure download method for HTTP' option.*

Using {remotes}, install the {pilot3} package running the code below.

```
pak::pkg_install("RConsortium/submissions-pilot3-utilities")
```

a. Set the working directory to the submission folder.

```
setwd("~/pilot3")
```

b. Initialize the renv workflow by running the code:

```
renv::init()
```

c. Move the file : `"~/pilot3/pilot3-files/m5/datasets/rconsortiumpilot3/analysis/programs/renv.lock"` to this location : `"~/pilot3"`

`NOTE :` This step will overwrite the current `.renv.lock` with the pilot3 `.renv.lock`.

d. Next run renv::activate() to activate the R environment.

e. Open .Rprofile and confirm these lines are included :

Sys.setenv(RENV_DOWNLOAD_FILE_METHOD = "libcurl")

source("renv/activate.R")

f. Restart the R session, then run renv::restore() to restore the environment in which the submission was prepared.

NOTE : When asked : `Do you want to proceed? [Y/n]: Y #set to Y`

BOOKMARK…in progress

```
This project has not yet been activated.
Activating this project will ensure the project library is used when
restore() is called.
Please see `?renv::activate` for more details.
```

```
Would you like to activate this project before restore() is called? [Y/n]: n
```

Next you will be asked if packages can be updated. Please choose Y.

```
The following package(s) will be updated:
…
Do you want to proceed? [y/N]: y
```

Using {renv}, initialize the R environment and install/restore the packages used in Pilot 3 running the code below.

```
renv::init()
```

This will then come up, where the input needed is option 1 :

```
This project already has a lockfile. What would you like to do?

1: Restore the project from the lockfile.
2: Discard the lockfile and re-initialize the project.
3: Activate the project without snapshotting or installing any packages.
4: Abort project initialization.
```

```
Selection:1
```

3. Specify R package repository
The R packages are based on CRAN at 2021-08-31. To install the exact R package versions used in this
project, run the code below to set the snapshot repository.
options(repos = "https://mran.microsoft.com/snapshot/2021-08-31")
4. Install open-source R packages
In the same R session, install the required packages by running the code below.
install.packages(c("haven", "dplyr", "emmeans", "pkglite" , "r2rtf", "rtables", "ggplot2",

"cowplot", "visR", "Tplyr", "pharmaRTF", "huxtable"))

5. Install Proprietary R packages
The proprietary R package "pilot1wrappers" is packed in the file r0pkg.txt. In the same R session, restore
the package structures and install them by running the code below. Adjust the output path as needed to use
a writable local directory.
pkglite::unpack("r0pkg.txt", output = ".", install = TRUE)
6. Update path to dataset and TLFs

13

R Consortium R Submission Working Group Pilot 1

INPUT path: to rerun the analysis programs, define the path variable
● Path for ADaM data: path$adam
OUTPUT path: to save the analysis results, define the path variable
● Path for output TLFs: path$output
All these paths must be defined before executing the analysis output program. For example:
path = list(adam = "path/to/esub/analysis/adam/datasets", # Modify path to the actual location
output = ".") # Output saved in current folder
7. Execute analysis program
To reproduce analysis results, rerun the following four programs:
● "tlf-demographic.r"
● "tlf-efficacy.r"
● "tlf-kmplot.r"
● "tlf-primary.r"

Extract Application Bundle

Use the pkglite package to unpack the r3pkg.txt within the Pilot 3 eCTD submission transfer. This file is located in the following relative path within the eCTD transfer directory:

m5\datasets\rconsortiumpilot2\analysis\adam\programs\r3pkg.txt

Enter the following command in the R console to extract the Shiny application files to the destination directory.

```
pkglite::unpack(
input =
"C:/pilot3/m5/datasets/rconsortiumpilot3/analysis/adam/programs/r3pkg.txt",
output = "C:/pilot3-files"
)
```

The console will display messages of unpacking and writing files to the destination directory. Note that the procedure creates a sub-directory called pilot3wrappers in the destination directory. Take note of that particular directory path on your system, as you will use this in the remaining procedures. In this example, the directory is located in the following path:

C:\pilot3-files\pilot3wrappers

Initialize R Package Environment for ADaM generation

The dependencies for the ADaM generation are managed by the renv R package management system. To bootstrap the customized R package library used to generate the analysis datasets, launch a new R session in the directory where you unpacked the application source files in the previous step. Use either of the following procedures depending on your R computing environment:

RStudio

Create a new RStudio Project within the pilot3wrappers directory:

1. Select File -> New Project
2. In the Create Project dialog box, choose **Existing Directory**
3. In the Create Project from Existing Directory dialog box, click the **Browse** button and navigate to the pilot3wrappers directory.

4. Once the location has been confirmed, click the **Create Project** button.

RStudio will refresh the window and automatically install the renv package into the project directory. To complete the process of restoring the pilot R packages, run the following command in the R console:

```
renv::restore(prompt = FALSE)
```

The package installation procedure may take a few minutes or longer depending on internet bandwidth.

Console

Launch a new R session in the pilot3wrappers directory of the unpacked application directory. By default, the R Gui interface on Windows will launch a new R session in your default Windows home directory (typically the **Documents** folder). Perform the following steps to ensure R is launched in the proper directory.

Note

The procedure below assumes R 4.2.1 has been installed in a default location. If you are unsure of the full path to the R GUI executable on your system, you can find the location on your system by performing the following steps:

1. Open the Windows Start Menu and expand to show all applications.
2. Navigate to the R entry and expand the section such that all R program entries are visible.
3. Right-click the R x64 4.2.1 entry and select More -> Open file location.
4. A new folder window will open with the shortcut R x64 4.2.1 highlighted. Right-click this entry and select **Properties**
5. In the Properties window, copy the path specified in the **Target** text field. The portion of the text in quotations gives the full path to the Rgui.exe location on your system.

1. Open the Windows Powershell program by searching for Windows Powershell in the Windows Start menu.
2. Change the current directory to the pilot3wrappers directory by running the following command (substitute the pilot3-files location for your appropriate directory as needed):

```
Set-Location -Path "C:\pilot3-files\pilot3wrappers"
```

3. Launch the Windows R GUI in this session by running the following command:

```
C:\"Program Files"\R\R-4.2.1\bin\x64\Rgui.exe
```

The R GUI will launch and automatically install the renv package into the project directory. To complete the process of restoring the pilot R packages, run the following command in the R console:

```
renv::restore(prompt = FALSE)
```

The package installation procedure may take a few minutes or longer depending on internet bandwidth.

Analysis Datasets generated from Pilot 3

Upon completion of package installation, you may import the ADaM data sets contained in the eCTD transfer, which were generated using R in this Pilot. The data files are located in the following relative path within the eCTD transfer directory:

```
m5\datasets\rconsortiumpilot3\analysis\adam\datasets
```

Run the following command in the R console (substitute the pilot3 location for your appropriate directory as needed):

```
Pilot3wrappers::set_data_path("C:/pilot3/m5/datasets/rconsortiumpilot3/analysis/adam/datasets")
```

Executing the R code to reproduce the Analysis Datasets

After importing the analysis datasets generated from Pilot 3, you may now run the R code, which were developed to reproduce the Analysis Datasets for further review. The R programs are located in the following relative path within the eCTD transfer directory:

```
m5\datasets\rconsortiumpilot3\analysis\adam\programs
```

To reproduce the analysis datasets used to generate the outputs in Pilot 1, rerun the following programs :

Program Name	Analysis Dataset Name	Analysis Dataset Label
adsl.r	adsl.xpt	Subject Level Analysis Dataset
adae.r	adae.xpt	Adverse Events Analysis Dataset
adadas.r	adas.xpt	ADAS-Cog Analysis Dataset
adlbc.r	adlbc.xpt	Lab Blood Chemistry Analysis Dataset
adtte.r	adtte.xpt	AE Time to 1st Derm. Event Analysis