Analysis Data Reviewer's Guide

R Consortium R Submission Pilot 4

R Consortium

1 Introduction

1.1 Purpose

The Analysis Data Reviewer's Guide (ADRG) provides specific instructions for executing a Shiny application created with the R-language for viewing analysis results and performing custom subpopulation analysis based on the data sets and analytical methods used in the R Consortium R Submission Pilot 1. This document provides context for the analysis datasets and terminology that benefit from additional explanation beyond the Data Definition document (define.xml), as well as a summary of ADaM conformance findings. Section A provides detailed procedures for installing and configuring a local R environment and the Docker container runtime to view the included Shiny application.

1.2 Study Data Standards and Dictionary Inventory

Standard or Dictionary	Versions Used
SDTM	SDTM v1.4/ SDTM IG v3.1.2
ADaM	ADaM v2.1/ ADaM IG v1.0
Controlled Terminology	SDTM CT 2011-12-09
	ADaM CT 2011-07-22
Data Definitions	define.xml v2.0
Medications Dictionary	MedDRA v8.0

1.3 Source Data Used for Analysis Dataset Creation

The ADaMs we used to regenerate the outputs were the PHUSE CDISC Pilot replication ADaMs following ADaM IG v1.0. The ADaM dataset and its corresponding SDTM data set are publicly available at the PHUSE Github Repository (https://github.com/phuse-org/phuse-scripts/blob/master/data/adam/TDF_ADaM_v1.0.zip, https://github.com/phuse-org/phuse-scripts/blob/master/data/sdtm/TDF_SDTM_v1.0%20.zip)

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

The reference documents can be found at https://github.com/phuse-org/phuse-scripts/blob/master/data/adam/TDF_ADaM_v1.0.zip

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.

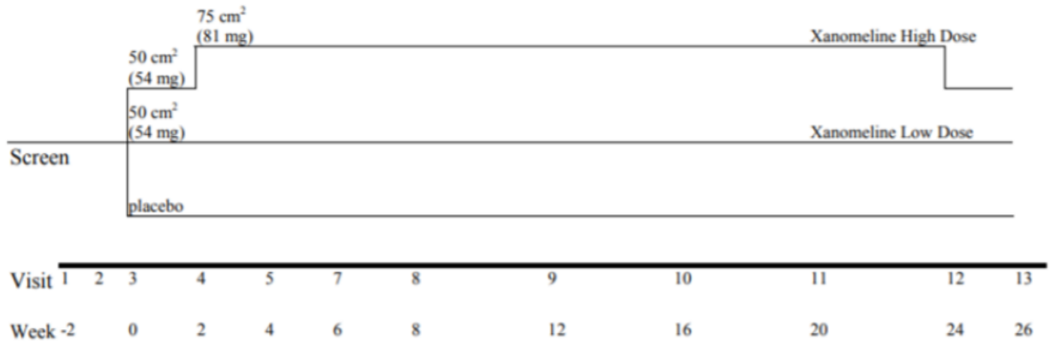
Methodology:

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
USUBJID	Unique subject identifier
STUDYID	Study Identifier
SITEID	Study Site Identifier
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)
SEX	Sex
RACE	Race
RACEN	Race (N)

3.2 Treatment Variables

- Are the values of `ARM` equivalent in meaning to values of `TRTxxP`? Yes
- Are the values of `TRTxxA` equivalent in meaning to values of `TRTxxP`? Yes
- Are both planned and actual treatment variables used in analyses? Yes

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

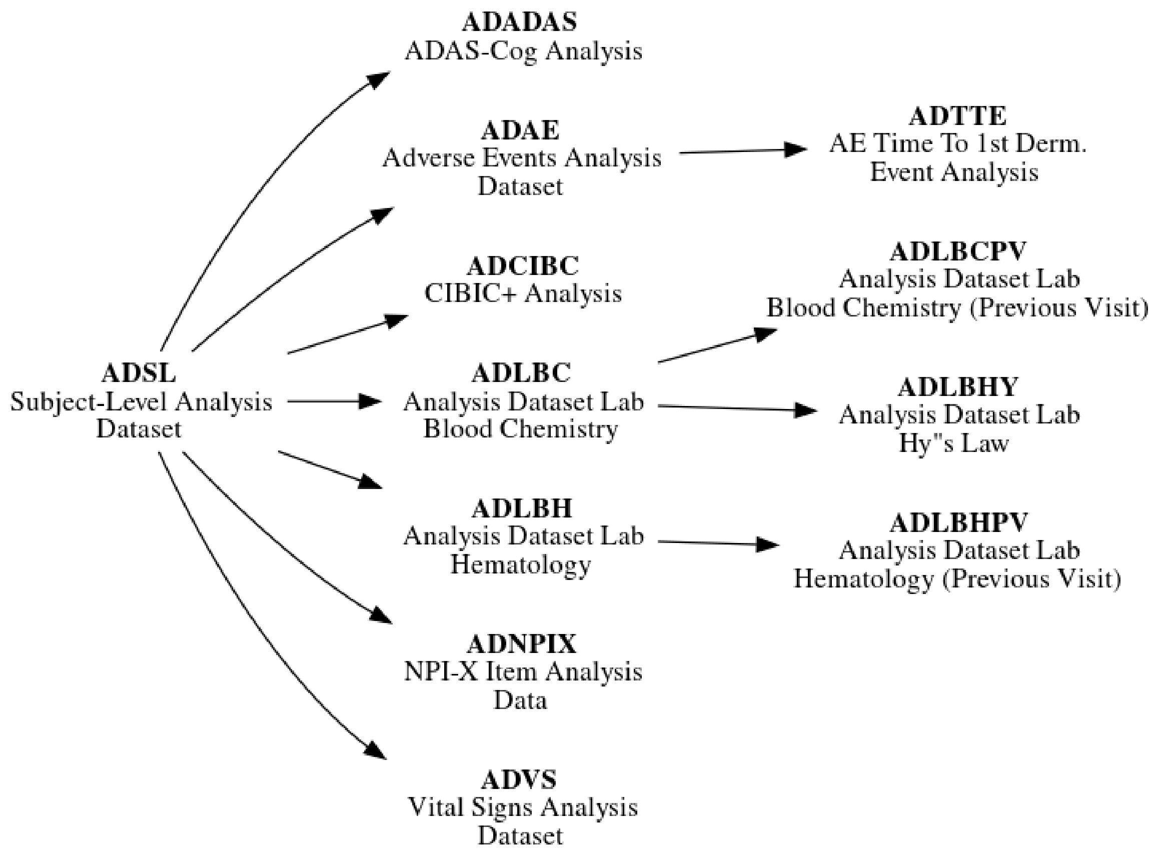
- Was windowing used in one or more analysis datasets? Yes
- Were unscheduled visits used in any analyses? Yes

3.4 Imputation/Derivation Methods

Not applicable.

4 Analysis Data Creation and Processing Issues

4.1 Data Dependencies


ADADAS
ADAS-Cog Analysis
ADAE
Adverse Events Analysis Dataset
ADTTE
AE Time To 1st Derm. Event Analysis
ADCIBC
CIBIC+ Analysis
ADLBCPV
Analysis Dataset Lab Blood Chemistry (Previous Visit)
ADSL
Subject-Level Analysis Dataset
ADLBC
Analysis Dataset Lab Blood Chemistry
ADLBHY
Analysis Dataset Lab Hy"s Law
ADLBH
Analysis Dataset Lab Hematology
ADLBHPV
Analysis Dataset Lab Hematology (Previous Visit)
ADNPIX
NPI-X Item Analysis Data
ADVS
Vital Signs Analysis Dataset

5 Analysis Dataset Description

5.1 Overview

The Shiny application modules in Pilot 4 cover part of the efficacy and safety objectives of the initial protocol. More specifically, five analysis outputs are included, covering demographics analysis, primary efficacy endpoint analysis, safety analysis, and visit completion.

5.2 Analysis Datasets

The following table provides detailed information for each analysis dataset included in the Pilot 4 submission. The Shiny application for this pilot utilizes the following analysis datasets: `ADSL`, `ADTTE`, `ADADAS`, `ADLBC`.

Dataset	Label	Class	Efficacy	Safety	Baseline or other subject characteristics	Primary Objective	Structure
ADSL	Subject Level Analysis Dataset	ADSL			x		One observation per subject
ADAE	Adverve Events Analysis Dataset	ADAM OTHER		x			One record per subject per adverse event
ADTTE	Time to Event Analysis Dataset	BASIC DATA SCTRUCTURE		x			One observation per subject per analysis parameter
ADLBC	Analysis Dataset Lab Blood Chemistry	BASIC DATA SCTRUCTURE		x			One record per subject per parameter per analysis visit
ADLBCPV	Analysis Dataset Lab Blood Chemistry (Previous Visit)	BASIC DATA SCTRUCTURE		x			One record per subject per parameter per analysis visit
ADLBH	Analysis Dataset Lab Hematology	BASIC DATA SCTRUCTURE		x			One record per subject per parameter per analysis visit
ADLBHPV	Analysis Dataset Lab Hematology (Previous Visit)	BASIC DATA SCTRUCTURE		x			One record per subject per parameter per analysis visit
ADLBHY	Analysis Dataset Lab Hy's Law	BASIC DATA SCTRUCTURE		x			One record per subject per parameter per analysis visit
ADADAS	ADAS-Cog Analysis	BASIC DATA SCTRUCTURE	x			x	One record per subject per parameter per analysis visit per analysis date
ADCIBC	CIBIC+ Analysis	BASIC DATA SCTRUCTURE	x				One record per subject per parameter per analysis visit per analysis date
ADNPIX	NPI-X Item Analysis Data	BASIC DATA SCTRUCTURE	x				One record per subject per parameter per analysis visit
ADVS	Vital Signs Analysis Dataset	BASIC DATA SCTRUCTURE		x			One record per subject per parameter per analysis visit

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 ADAE - Adverse Events Analysis Data

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.

5.2.3 ADTTE - Time to Event Analysis Dataset

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”.

5.2.4 ADLBHPV - Laboratory Results Hematology Analysis Data (Previous Visit)

ADLBC and ADLBH contain one record per lab analysis parameter, per time point, per subject.

ADLBC contains lab chemistry parameters and ADLBH contains hematology parameters and these data are derived from the SDTM LB (Laboratory Tests) domain. Two sets of lab parameters exist in ADLBC/ADLBH. One set contains the standardized 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.

The ADLBC and ADLBH datasets were split based on the values of the indicated variable. Note that this splitting was done to reduce the size of the resulting datasets and to demonstrate split datasets and not because of any guidance or other requirement to split these domains.

5.2.5 ADLBHY - Laboratory Results Hy’s Law Analysis Data

ADLBHY contains one record per lab test code per sample, per subject for the Hy’s Law based analysis parameters. ADLBHY is derived from the ADLBC (Laboratory Results Chemistry Analysis Data) analysis dataset. It contains derived parameters based on Hy’s law.

5.2.6 ADADAS - ADAS-COG Data

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.

5.2.7 ADCIBC - CIBC Data

ADCIBC contains analysis data from the from CIBIC+ questionnaire, one of the primary efficacy endpoints. It contains one record per subject per VISIT. Note that for all records, PARAM='CIBIC Score'. Visits are placed into analysis visits (represented by AVISIT and AVISITN) based on the date of the visit and the visit windows.

5.2.8 ADNPIX - NPI-X Item Analysis Data

ADNPIX contains one record per subject per parameter (NPI-X questionnaire item, total score, and mean total score from Week 4 through Week 24) per analysis visit (AVISIT). The analysis visits (represented by AVISIT and AVISITN) are derived from days between assessment date and randomization date and based on the visit windows that were specified in the statistical analysis plan (SAP).

6 Data Conformance Summary

6.1 Conformance Inputs

- Were the analysis datasets evaluated for conformance with CDISC ADaM Validation Checks? Yes, Version of CDISO ADaM Validation Checks and software used: Pinnacle 21 Enterprise version 4.1.1
- Were the ADaM datasets evaluated in relation to define.xml? Yes
- Was define.xml evaluated? Yes

Rule ID	Dataset(s)	Diagnostic Message	Severity	Explanation
AD0258	ADAE	Record key from ADaM ADAE is not traceable to SDTM.AE (extra ADAE recs)	Error	There are derived records in ADAE, this has no impact on the analysis.
AD0018	ADLBC, ADLBCPV, ADLBH, ADLBHPV, ADVS, ADCIBC, ADLBNPIX	Variable label mismatch between dataset and ADaM standard	Error	The label for ANL01FL in these datasets are 'Analysis Record Flag 01', this is in conformance with ADaM IG 1.0, this is an issue in P21 checks, and has no impact on the analysis.
AD0320	ADSL	Non-standard dataset label	Error	The label for ADSL is 'ADSL', this has no impact on the analysis

6.2 Issues Summary

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 4.4.1.

7.2 ADaM Programs

Not Applicable. This pilot project only submits programs for analysis results.

7.3 Analysis Output Programs

The Shiny application included in this pilot follows a different structure than a traditional collection of analysis programs such as those included in the Pilot 1 eCTD transfer. The application is developed with a modular approach and assembled with the `golem` R package for enhanced code organization. A description of the primary scripts used within the application is given in the table below. The recommended steps to execute the Shiny application are described in Appendix 2.

Program Name	Purpose
app.R	Facilitate execution of Shiny application in a local R session or deployed on a server
app_teal.R	Assemble the application modules for use with the Teal package
tm_t_demographic.R	Shiny module for demographic and baseline characteristics analysis
tm_g_kmplot.R	Shiny module for Kaplan-Meier plot of time to first dermatologic event
tm_t_primary.R	Shiny module for primary endpoint analysis ADAS Cog (11)
tm_t_efficacy.R	Shiny module for primary endpoint analysis Glucose (mmol/L)
tm_t_disposition.R	Shiny module for summary of number of patients completing each visit in treatment period

7.4 Container Build Instructions

This pilot submission uses the Docker container runtime for the building and execution of the Shiny application within a container. Included in this submission is a file called `Dockerfile.txt` which contains the required steps to configure an environment inside the container with the necessary dependencies to run the Shiny application. The required procedures to create this container are aoutlined in Section A.

7.5 Open-source R Analysis Packages

The following table lists the open-source R packages used to create and execute the Shiny application in this pilot.

Package	Title	Version
R.utils	Various Programming Utilities	2.11.0
Tplyr	A Traceability Focused Grammar of Clinical Data Summary	0.4.4
config	Manage Environment Specific Configuration Values	0.3.1
cowplot	Streamlined Plot Theme and Plot Annotations for 'ggplot2'	1.1.1
dplyr	A Grammar of Data Manipulation	1.0.9
emmeans	Estimated Marginal Means, aka Least-Squares Means	1.7.2
ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics	3.3.5
glue	Interpreted String Literals	1.6.1
golem	A Framework for Robust Shiny Applications	0.3.1
haven	Import and Export 'SPSS', 'Stata' and 'SAS' Files	2.4.3
htmltools	Tools for HTML	0.5.2
huxtable	Easily Create and Style Tables for LaTeX, HTML and Other Formats	5.4.0
magrittr	A Forward-Pipe Operator for R	2.0.2
markdown	Render Markdown with 'commonmark'	1.1
pkgload	Simulate Package Installation and Attach	1.2.4
purrr	Functional Programming Tools	0.3.4
reactable	Interactive Data Tables for R	0.2.3
rtables	Reporting Tables	0.5.1.2
shiny	Web Application Framework for R	1.7.1
shinyWidgets	Custom Inputs Widgets for Shiny	0.6.3
stringr	Simple, Consistent Wrappers for Common String Operations	1.4.0
teal	Exploratory Web Apps for Analyzing Clinical Trials Data	0.11.1
teal.data	Data Model for 'teal' Applications	0.1.1
tibble	Simple Data Frames	3.1.6
tidyr	Tidy Messy Data	1.1.4
tippy	Add Tooltips to 'R markdown' Documents or 'Shiny' Apps	0.1.0

7.6 List of Output Programs

Not Applicable. This pilot project displays analysis output as a Shiny application where the R programs described in **Analysis Output Programs** (Section 7.3) as a whole produce the Shiny application.

8 Directory Structure

Study datasets and the Shiny application supportive files are organized in accordance to Study Data Technical Conformance Guide.

```
├── m1
|   └── us
```

```
│       └── cover-letter.pdf
├── m5
│   └── datasets
│       └── rconsortiumpilot4container
│           └── analysis
│               └── adam
│                   ├── datasets
│                   │   ├── adadas.xpt
│                   │   ├── adlbc.xpt
│                   │   ├── adrg.pdf
│                   │   ├── adsl.xpt
│                   │   ├── adtte.xpt
│                   │   ├── define2-0-0.xsl
│                   │   └── define.xml
│                   └── programs
│                       ├── Dockerfile.txt
│                       └── r1pkg.txt
```

[!h]

Directory	Index	Description
module	1	Refers to the eCTD module in which clinical study data is being submitted.
datasets	2	Resides within the module folder as the top-level folder for clinical study data being submitted for m5.
rconsortiumpilot4container	3	Study identifier or analysis type performed
analysis	4	Contains folders for analysis datasets and software programs; arrange in designated level 6 subfolders
adam	5	Contains subfolders for ADaM datasets and corresponding software programs
datasets	6	Contains ADaM datasets, analysis data reviewer's guide, analysis results metadata and define files
programs	7	Contains Shiny application source files bundled as a 'pkglite' text file and Docker container build instructions.

A Appendix 1: Pilot 4 Shiny Application Installation and Usage

To install and execute the Shiny application, follow all of the procedures below. Ensure that you note the location of where you downloaded the Pilot 4 eCTD submission files. For demonstration purposes, the procedures below assume the transfer has been saved to this location: `C:\pilot4container`.

Note

It is recommended to view this document in a dediciated PDF viewer application (such as Adobe Reader). When using a web browser such as Microsoft Edge or Google Chrome to view this file, certain commands are not able to be copied correctly while preserving line breaks or other formats, especially when pasting those commands in the Windows PowerShell terminal application.

A.1 Windows Subsystem for Linux (WSL)

A.1.1 Verification of WSL Availability

To execute a container on a Windows system, the Windows Subsystem for Linux (WSL) must be installed on the system. To determine if WSL is available on your system, use the following procedure:

1. Open the Windows Powershell program by searching for Windows Powershell in the Windows Start menu.
2. Run the following command:

```
wsl --list --verbose
```

If the command results in output displaying the manual help contents (such as the possible command-line arguments) to `wsl.exe`, you will need to install WSL using the procedure in the following section. However, if a list of Linuc distributions is displayed instead, you can proceed to the section {Section A.2.1}.

A.1.2 Installation of WSL

1. Open the Windows Powershell program as Administrator by searching for Windows Powershell in the Windows Start menu.
2. Run the following command:

```
wsl --install
```

Depending on network connection and system resources, the installation may take a few minutes.

Upon successful installation, you will see messages indicating that the installation was successful and be prompted to restart your system. Hence you need to restart your system to complete the installation.

After the restart of your system is complete, you will see a PowerShell window appear automatically to complete the process. Depending on your Windows installation, the following two situations can arise:

Scenario one: Automatic Installation of Ubuntu Linux: The installer will proceed to download and install the Ubuntu Linux distribution automatically. Once the installation of Ubuntu Linux completes, you will be prompted to create a new user account within the virtual environment running this Linux distribution. Enter a user name and password (this does not have to be identical to the user account for your Windows installation). After the account is created, you will be connected to a terminal session inside the virtual machine. You can disconnect from this virtual machine interface by typing `exit` in the console prompt, which will close the PowerShell window.

Scenarion two: Missing Linux Distribution: In certain situations, WSL will install correctly but it will not be able to install the Ubuntu Linux distribution during the same installation procedure. Use the following procedure to install the Ubuntu Linux distribution should that issue arise:

1. Open the Windows Powershell program by searching for Windows Powershell in the Windows Start menu.
2. Run the following command:

```
wsl --install Ubuntu
```

Once the installation of Ubuntu Linux completes, you will be prompted to create a new user account within the virtual environment running this Linux distribution. Enter a user name and password (this does not have to be identical to the user account for your Windows installation). After the account is created, you will be connected to a terminal session inside the virtual machine. You can disconnect from this virtual machine interface by typing `exit` in the console prompt, which will close the PowerShell window.

A.2 Docker Desktop

The container runtime used in this application is Docker. To utilized Docker on a Windows-based system, it is recommended to install the official Docker Desktop utility.

A.2.1 Installation of Docker Desktop

1. Visit the Docker Desktop web site https://www.docker.com/products/docker-desktop/ and click the **Download Docker Desktop** button. A small window appears with links to the different versions based on operating system. Click the entry **Download for Windows - AMD64** to download the installer for Windows.

2. After the download is complete, locate the file `Docker Desktop Installer.exe` and open the file to begin the installation process. Ensure that the options **Use WSL 2 instead of Hyper-V (recommended)** and **Add shortcut to desktop** are selected, and then click the OK button.
3. The installation will unpack and copy the necessary files. Depending on your system resources, the procedure could last a few minutes or longer. Once it is complete, the installer will display a message saying the installation succeeded. The installer window will prompt you to either buttons labelled **Close and log out** or **Close and restart**. Click the button to perform the recommended operation.
4. After you log back in to the system, a new window appears asking for confirmation of the Docker Subscription Service Agreement. Click the **Accept** button.
5. A new window appears with the title **Welcome to Docker** with a question regarding the use of Docker for work. This is an optional step and it is recommended to click the **Skip** link in the upper-right corner.
6. The window displays a Welcome Survey. This is an optional step. Click the **Skip** link in the upper-right corner.
7. The default Docker Desktop interface appears. You can safely close the window, as the Docker process will still run in the system background.

A.3 Installation of R and RStudio

Download and install R 4.4.1 for Windows from https://cloud.r-project.org/bin/windows/base/old/4.4.1/R-4.4.1-win.exe. While optional, it is also recommended to use RStudio IDE for executing R code to launch the application. You can download RStudio for Windows by visiting https://posit.co/download/rstudio-desktop/#download.

When launching RStudio for the first time, you may be prompted to select the version of R to use. Ensure that the default selection of **Use your machine's default 64-bit version of R** is selected and click OK.

A.4 Installation of Rtools

Due to certain R packages requiring compilation from source, it is also required that you install the **Rtools** Windows utility from CRAN. You can download Rtools built for R version `4.4.1` by visiting https://cloud.r-project.org/bin/windows/Rtools/rtools44/files/rtools44-6459-6401.exe. During the installation procedure, keep the default choices in the settings presented in the installation dialog.

Once the installation is complete, launch a new R session (if you have an existing session open, close that session first) and in the console, run the following command that should give the location of your Rtools installation:

```
Sys.which("make")
"C:\\rtools44\\usr\\bin\\make.exe"
```

A.5 Installation of R Packages

A minimum set of R packages are required to ensure the Pilot 4 Shiny application files are successfully unpacked and the custom package environment used for the application is replicated correctly. Run the following commands in the R console to install the `remotes` and `pkglite` packages:

```
install.packages("remotes")

# install version 0.2.4 of the pkglite package
remotes::install_version("pkglite", version = "0.2.4")
```

A.6 Extract Application Bundle

Use the `pkglite` package to unpack the Shiny application bundle `r1pkg.txt` within the Pilot 4 eCTD submission transfer. This file is located in the following relative path within the eCTD transfer directory:

```
m5\datasets\rconsortiumpilot4container\analysis\adam\programs\r1pkg.txt
```

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

```
# create application directory
dir.create("C:/pilot4container_files/datasets/adam", recursive = TRUE)

pkglite::unpack(
  "C:/pilot4container/m5/datasets/rconsortiumpilot4container/analysis/adam/programs/r1pkg.txt",
  output = "C:/pilot4container_files"
)
```

The console will display messages of unpacking and writing files to the destination directory. Note that the procedure creates a sub-directory called `pilot2wrappers` in the destination directory. In this example, the directory is located in the following path:

```
C:\pilot4container_files\pilot2wrappers
```

The data files used by the application, as well as the container image instructions file called `Dockerfile.txt` will also need to be copied over to the `pilot4container_files` directory. Enter the following commands in the R console to copy the required files:

```r
# copy data files
data_files <- list.files(
  "C:/pilot4container/m5/datasets/rconsortiumpilot4container/analysis/adam/datasets",
  pattern = "*.xpt",
  full.names = TRUE
)

sapply(
  data_files,
  file.copy,
  to = "C:/pilot4container_files/datasets/adam"
)

# copy container image instructions file
file.copy(
  "C:/pilot4container/m5/datasets/rconsortiumpilot4container/analysis/adam/programs/Dockerfile.txt",
  to = "C:/pilot4container_files"
)
```

A.7 Build Container Image

Create the container image associated with the Shiny application with Docker using the following procedure:

1. Open the Windows Powershell program by searching for Windows Powershell in the Windows Start menu.
2. Change the current directory to the `pilot4container_files` directory by running the following command (substitute the `pilot4container_files` location for your appropriate directory as needed):

```
Set-Location -Path "C:\pilot4container_files"
```

3. Create the container image file by running the following command:

```
docker build `
  --build-arg LOCAL_DATA_DIR=datasets `
  --build-arg LOCAL_APP_DIR=pilot2wrappers `
  -t RConsortium/submissions-pilot4-container:latest `
  -f Dockerfile.txt .
```

Note

Depending on network bandwidth and your computer's hardware profile, the process of building the Docker container image may require 5-10 minutes or longer to complete. Do not close the Powershell window during the process.

A.8 Run Shiny Application

To run the Shiny application on your computer, in the same Powershell window run the following command:

```
docker run -it --rm -p 8787:8787 RConsortium/submissions-pilot4-container:latest
```

Caution

On certain installations of Windows, a Windows Security Alert may appear indicating that the Windows Defender Firewall has blocked features of the Docker Desktop backend on your computer's network. If this happens, click the **Allow access** button to give Docker permission to utilize your computer's network to run web-based applications. You will need to re-execute the container run command after enabling this access.

To view the application, launch a new web browser session in Microsoft Edge and paste the following address in the address bar:

```
http://127.0.0.1:8787
```

Note

In the output of the `docker run` command, the R console from the Docker container will display a message `Listening on http://0.0.0.0:8787`, which is a different address than the address entered in the web browser (`http://127.0.0.1`). To facilite visiting the Shiny application using web browser installed on your Windows environment, the Docker container exposes the HTTP port `8787` to the host environment and the Shiny process is executed with the options `shiny.host = '0.0.0.0'` and `shiny.port = 8787`. If these configurations were not present, the host environment's web browser would not be able to access the Shiny application from the container.

When you are ready to stop the application, return to the PowerShell window and press `Ctrl + C` on your keyboard.

Appendix 2: Application Usage Guide

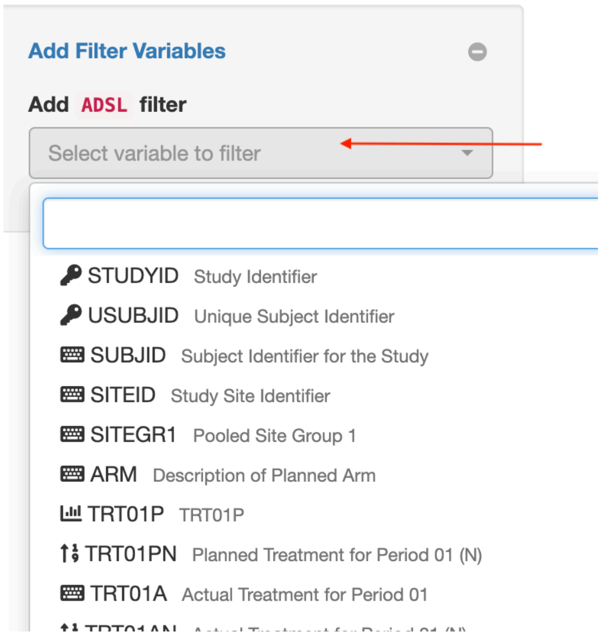
The Shiny application contains 5 tabs, with the first table **App Information** selected by default. The relationship between the other application tabs and previously submitted analysis from Pilot 1 are described in the table below:

Application Tab	Pilot 1 Output
Demographic Table	Table 14-2.01 Summary of Demographic and Baseline Characteristics
KM plot for TTDE	Figure 14-1 Time to Dermatologic Event by Treatment Group
Primary Table	Table 14-3.01 Primary Endpoint Analysis: ADAS Cog(11) - Change from Baseline to Week 24 - LOCF
Efficacy Table	Table 14-3.02 Primary Endpoint Analysis: Glucose (mmol/L) - Summary at Week 20 - LOCF
Visit Completion Table	Not Applicable

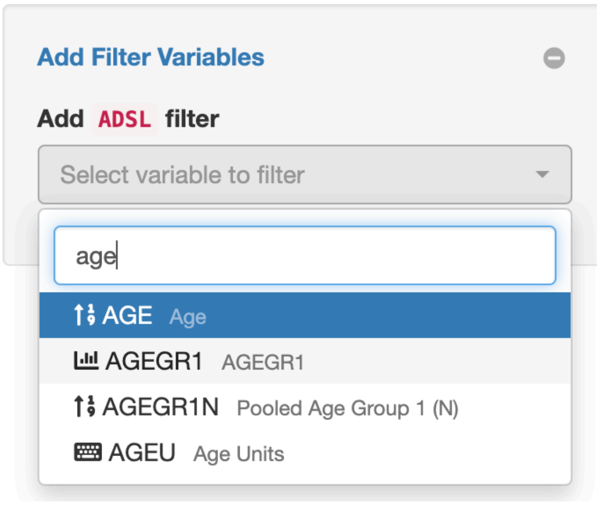
The default display in the analysis tabs match with the outputs submitted in Pilot 1, as well as an additional table on visit completion.

The **KM plot for TTDE** module allows for filters to be applied based on variables in the **ADSL** and **ADTTE** data sets. Below is an example of performing subpopulation analysis for an age group within the module:

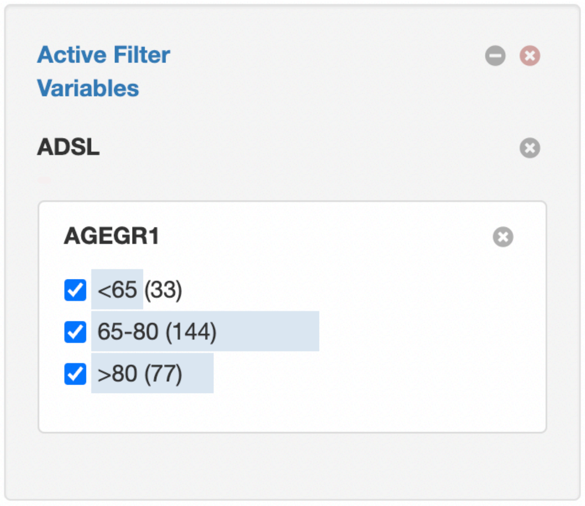
1. Within the **Add Filter Variables** widget, click the box with the placeholder **Select variables to filter**.


Add Filter Variables
Add ADSL filter
Select variable to filter
STUDYID Study Identifier
USUBJID Unique Subject Identifier
SUBJID Subject Identifier for the Study
SITEID Study Site Identifier
SITEGR1 Pooled Site Group 1
ARM Description of Planned Arm
TRT01P TRT01P
TRT01PN Planned Treatment for Period 01 (N)
TRT01A Actual Treatment for Period 01

2. Scroll up/down or use the search bar to find the variable for subpopulation. Click the desired variable (**AGEGR1** in this example).



3. In the **Active Filter Variables** widget, the selected variable with its available categories or levels will display. In this example, **AGEGR1** in this example) is displayed with three categories. If the selected variable in the previous step is a continuous variable, then a slider will appear for selecting a range of values.



4. Select the target subpopulation (e.g. `>80`) and the analysis output displayed on the left hand side will be updated in real-time according to the selection, which in this example is equivalent to performing a filter on the **ADSL** data by `AGEGR1 == '>80'`.

Note

When applying one or more filters in the KM-plot module, the filtered data set may not contain enough observations to produce reliable survival probabilities and associated 95% confidence intervals. In those situations, the application will present to the user a message indicating not enough observations based on the current filter selections.
In addition, the R console could display warnings about value comparisons to a min or max cutoff. These warnings can be safely disregarded as they do not effect the filtered data set after processing is complete.