

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 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



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, ADTE, 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	Adverse Events Analysis Dataset	ADAM OTHER		x			One record per subject per adverse event
ADTTE	Time to Event Analysis Dataset	BASIC DATA SSTRUCTURE		x			One observation per subject per analysis parameter
ADLBC	Analysis Dataset Lab Blood Chemistry	BASIC DATA SSTRUCTURE		x			One record per subject per parameter per analysis visit
ADLBPCV	Analysis Dataset Lab Blood Chemistry (Previous Visit)	BASIC DATA SSTRUCTURE		x			One record per subject per parameter per analysis visit
ADLBH	Analysis Dataset Lab Hematology	BASIC DATA SSTRUCTURE		x			One record per subject per parameter per analysis visit
ADLBHPV	Analysis Dataset Lab Hematology (Previous Visit)	BASIC DATA SSTRUCTURE		x			One record per subject per parameter per analysis visit
ADLBHY	Analysis Dataset Lab Hy's Law	BASIC DATA SSTRUCTURE		x			One record per subject per parameter per analysis visit
ADADAS	ADAS-Cog Analysis	BASIC DATA SSTRUCTURE	x			x	One record per subject per parameter per analysis visit per analysis date
ADCIBC	CIBIC+ Analysis	BASIC DATA SSTRUCTURE	x				One record per subject per parameter per analysis visit per analysis date
ADNPIX	NPI-X Item Analysis Data	BASIC DATA SSTRUCTURE	x				One record per subject per parameter per analysis visit
ADVS	Vital Signs Analysis Dataset	BASIC DATA SSTRUCTURE		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. In addition, the framework used for assembling the Shiny application modules is different than the framework used in the Pilot 2 eCTD transfer. The application is developed with a modular approach and assembled with the [rhino](#) R package for enhanced code organization. At the time of this submission, the [golem](#) R package (used by Pilot 2) is not supported by [webR](#). A description of the primary scripts used within the application is given in the table below.

7.4 Application Execution Functions

An additional set of R functions are included in the `utils.R` script to support creating and executing the web-assembly version of the application. Descriptions of the key functions are included in the table below. The recommended steps to execute the Shiny application with

Program Name	Directory	Purpose
app.R		Facilitate execution of Shiny application
adam_data.R	logic	Functions to import AdAM data files
eff_models.R	logic	Functions to perform statistical models and inferences used in output tables
formatters.R	logic	Functions to perform formatting of numerical results
helpers.R	logic	Supporting functions related to table output formatting and other operations
kmplot_helpers.R	logic	Functions to support creation of Kaplan-Meier plot
Tplyr_helpers.R	logic	Functions to support nesting of row labels in summary tables produced by the Tplyr package
completion_table.R	views	Shiny module for visit completion summary table
demographic_table.R	views	Shiny module for demographic summary table
efficacy_table	views	Shiny module for secondary endpoints efficacy table
km_plot_filter.R	views	Shiny module for data filter widgets used in Kaplan-Meier visualization module
km_plot.R	views	Shiny module for Kaplan-Meier visualization
primary_table.R	views	Shiny module for primary efficacy analysis summary table
user_guide.R	views	Shiny module for application user guide

these functions (along with preparing your environment for evaluating the application) are outlined in Section A.

Function Name	Purpose
build_app	Create the web-assembly application bundle directly from the Shiny application files
run_app_webassembly	Launch the web-assembly version of the application in a separate web process
run_app_shiny	Launch the Shiny application using the traditional Shiny process inside the R session

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.

Note

While the rhino package was used to create the application structure, it is not required to execute the application and thus was not included in the table.

Package	Title	Version
R6	Encapsulated Classes with Reference Semantics	2.5.1
Tplyr	A Traceability Focused Grammar of Clinical Data Summary	1.2.1
box	Write Reusable, Composable and Modular R Code	1.2.0
dplyr	A Grammar of Data Manipulation	1.1.4
emmeans	Estimated Marginal Means, aka Least-Squares Means	1.11.0
fontawesome	Easily Work with 'Font Awesome' Icons	0.5.2
formatters	ASCII Formatting for Values and Tables	0.5.8
ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics	3.5.1
glue	Interpreted String Literals	1.7.0
graphics	The R Graphics Package	4.4.1
haven	Import and Export 'SPSS', 'Stata' and 'SAS' Files	2.5.4
huxtable	Easily Create and Style Tables for LaTeX, HTML and Other Formats	5.5.6
markdown	Render Markdown with 'commonmark'	1.13
purrr	Functional Programming Tools	1.0.2
reactable	Interactive Data Tables for R	0.4.4
rtables	Reporting Tables	0.6.9
shiny	Web Application Framework for R	1.8.1.1
shinylive	Run 'shiny' Applications in the Browser	0.2.0
stats	The R Stats Package	4.4.1
stringr	Simple, Consistent Wrappers for Common String Operations	1.5.1
tibble	Simple Data Frames	3.2.1
tidyr	Tidy Messy Data	1.3.1
tippy	Add Tooltips to 'R markdown' Documents or 'Shiny' Apps	0.1.0
visR	Clinical Graphs and Tables Adhering to Graphical Principles	0.4.1

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
  └─ rconsortiumpilot4webR
    └─ analysis
      └─ adam
        ├── datasets
        │   ├── adadas.xpt
        │   ├── adlbc.xpt
        │   ├── adsl.xpt
        │   ├── adtte.xpt
        │   ├── define2-0-0.xml
        │   └─ define.xml
        └─ programs
            ├── pilot4_webR_pkglite.txt
            └─ pkglite_utils.r

```

[!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.
rconsortiumpilot4webR	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 application and environment files packaged as text files.

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:\pilot4webR.

A.1 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.2 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.3 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 **renv** packages:

```
install.packages("remotes")  
  
# install version 1.0.7 of the renv package:  
remotes::install_version("renv", version = "1.0.7")  
  
# install version 0.2.4 of the pkglite package:  
remotes::install_version("pkglite", version = "0.2.4")
```

A.4 Extract Application Bundle

Use the **pkglite** package to unpack the Shiny application bundle **pilot4_webR_pkglite.txt** within the Pilot 4 eCTD submission transfer. This file is located in the following relative path within the eCTD transfer directory:

```
m5\datasets\rconsortiumpilot4webR\analysis\adam\programs\pilot4_webR_pkglite.txt
```

Enter the following commands in the R console to extract the Shiny application files and additional environment files to the appropriate destination directory:

```
# create application directory
dir.create("C:/pilot4webR_files")

# source utils.R script
source("C:/pilot4webR/m5/datasets/rconsortiumpilot4webR/analysis/adam/programs/pkglite_utils.r")

# unpack submission pkglite bundle
unpack_pkglite_bundle(
  pkglite_file = "C:/pilot4webR/m5/datasets/rconsortiumpilot4webR/analysis/adam/programs/pilot4_webR_p
  datasets_dir = "C:/pilot4webR/m5/datasets/rconsortiumpilot4webR/analysis/adam/datasets",
  output_dir = "C:/pilot4_webR_files"
)
```

A.5 Initialize R Package Environment

The dependencies for executing the Shiny application are managed by the `renv` R package management system. To bootstrap the customized R package library used for the Shiny application, launch a new R session in the directory where you unpacked the application source files in the previous step. **Choose one of the following options** depending on your R computing environment and preference.

Note

The package library includes all packages required to execute the application in the traditional method. However, executing the application in the web-assembly method requires a minimal set of packages. Additional details can be found in [Section A.7](#).

Option 1: RStudio

Open the RStudio Project file `submissions-pilot4-webR.Rproj` within the directory of the extracted application bundle:

1. Select **File** -> **Open Project**
2. Click the **Browse** button and navigate to the `C:/pilot4webR_files` directory to select the `submissions-pilot4-webR.Rproj` file.

RStudio will refresh the window and automatically install the `renv` package into the project directory. You may see a prompt about the installation of the `BiocManager` package. If so, accept the installation by typing `y` in the console. To complete the process of restoring remaining R packages, run the following command in the R console:

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

Once the package installation process is complete, run the following code to load a set of functions utilized in the remaining steps:

```
source("utils.R")
```

Option 2: R Console

Launch a new R session in the `C:/pilot4webR_files` directory of the extracted application bundle. 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 4.4.1 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 4.4.1 entry and select **More -> Open file location**.
4. A new folder window will open with the shortcut R x64 4.4.1 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 `C:/pilot4webR_files` directory by running the following command (substitute the `C:/pilot3webR_files` location for your appropriate directory as needed):

```
Set-Location -Path "C:\pilot4webR_files"
```

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

```
C:\Program Files\R\R-4.4.1\bin\x64\Rgui.exe
```

The R GUI will launch and automatically install the `renv` package into the project directory. You will see a prompt about the installation of the **BiocManager** package. Accept the installation by typing `y` in the console. To complete the process of restoring remaining R packages, run the following command in the R console:

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

Note

Due to certain R packages in the application requiring compilation, the entire package restoration procedure may require at least ten minutes or longer to complete depending on internet bandwidth and your computer's hardware profile.

Once the package installation process is complete, you may see a message in the R console stating a new version of the **BiocManager** has been installed and to restart the R session to use that new version. This package is not utilized in the Pilot application, and you do not need to restart your R session. It is also possible that a message appears stating a dependency tree was repaired during package installation concerning the `visR` package and recommends to call the `renv::snapshot()` command to capture the dependencies in the `renv` lockfile. This is entirely optional, as the correct versions of the application dependencies are installed correctly.

Run the following code to load a set of functions utilized in the remaining steps:

```
source("utils.R")
```

A.6 Prepare Shiny Application

A.6.1 Compile Shiny Application

Run the following code to compile the Shiny application source files to a new sub-directory called `_site`:

```
build_app()
```

Note

During the compilation procedure, key assets necessary for the web assembly compilers will be downloaded and the console will display numerous messages and progress indicators. If you wish to not see these verbose messages during compilation, you can set the `quiet` parameter to `TRUE` during the compilation with `build_app(quiet = TRUE)`.

A.7 Launch Shiny Application

A.7.1 Web-assembly Method

The web-assembly version of the Shiny application can be launched with the following code:

```
run_app_webassembly()
```

A message appears in the R console displaying the web address of the application. To view the application, launch a new web browser session in Microsoft Edge and paste the address in the address bar. By default, the address will be `http://127.0.0.1:7654`. When you are ready to stop the application, return to the R console and either press `Ctrl + C` on your keyboard or press the escape key on your keyboard.

A.7.2 Traditional Shiny Method

While not required, the application can also be launched with the traditional method of executing directly in the R session using the following code:

```
run_app_shiny()
```

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:

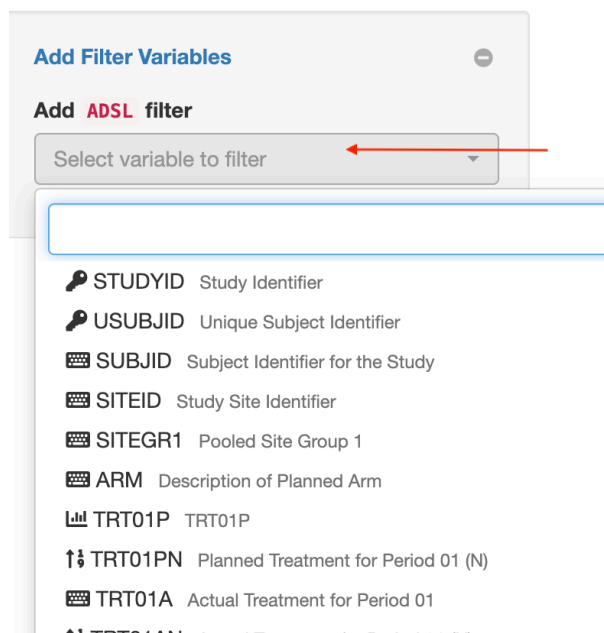
Application Tab	Pilot 1 Output
Demographic Table	Table 14-2.01 Summary of Demographic and Baseline Characteristics
KM plot for TTDE	Figure 14-1 Time to Dermatologic Event by Treatment Group

Application Tab	Pilot 1 Output
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**.



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

Add Filter Variables

Add ADSL filter

Select variable to filter

age

↑↓ AGE Age

AGEGR1 AGEGR1

↑↓ AGEGR1N Pooled Age Group 1 (N)

AGEU Age Units

- 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.

Active Filter Variables

ADSL

AGEGR1

☒ <65 (33)
☒ 65-80 (144)
☒ >80 (77)

- 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'`.

i 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.