Analysis Data Reviewer's Guide

R Consortium R Submission Pilot 5

R Consortium

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. Section 9 provides detailed procedures for installing and configuring a local R environment.

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.

Which can be traced back to the original CDISC 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

The reference documents can be found here.

2.2 Protocol Design in Relation to ADaM Concepts

2.2.1 Objectives:

The objectives of the study were to evaluate the efficacy and safety of transdermal xanomeline, $50cm^2$ and $75cm^2$, and placebo in subjects with mild to moderate Alzheimer's disease.

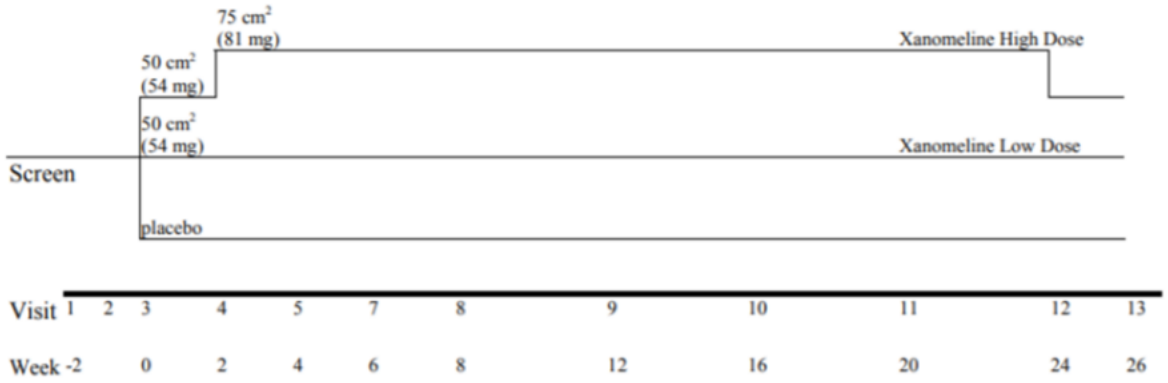
2.2.2 Methodology:

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.

2.2.3 Number of Subjects Planned:

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

2.2.4 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
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
ADADAS	ADSL
ADLBC	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

Dataset - Dataset Label	Class	Efficacy	Safety	Baseline or other subject characteristics	Primary Objective	Structure
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
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

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

```
Yes, Version of CDISC ADaM Validation Checks and software used: Pinnacle 21®
Community 4.0.2
```

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

```
Yes
```

Was define.xml evaluated?

```
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 a subset of ADaMs based on the CDSICPILOT data, using R. We compared our R generated ADaMs against the CDISCPILOT ADaMs, created in SAS, 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 (Appendix 2 and Appendix 3).

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 4.4.3.

7.2 ADaM Programs

The following table contains the list of programs that generate the analysis datasets in 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.json	Subject-Level Analysis Dataset
adadas.r	adas.json	ADAS-Cog Analysis
adlbc.r	adlb.json	Analysis Dataset Lab Blood Chemistry
adae.r	adae.json	Adverse Events Analysis Dataset
adtte.r	adtte.json	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 Dataset-JSON 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.

Script	Output	Analysis Dataset & Variables	Selection Criteria
tlf-demographic.r	tlf-demographic-pilot5.out	AGE.ADSL; AGEGR1.ADSL; RACE.ADSL; HEIGHTBL.ADSL; WEIGHTBL.ADSL; BMIBL.ADSL; MMSETOT.ADSL; STUDYID.ADSL; ITTFL.ADSL; TRT01P.ADSL	ADSL.STUDYID == "CDISCPILOT01"; ADSL.ITTFL == "Y"
tlf-efficacy.r	tlf-efficacy-pilot5.rtf	ADSL.STUDYID; ADSL.USUBJID	ADSL.ITTFL == "Y"; ADLB.TRTPN %in% c(0, 81); ADLB.PARAMCD == "GLUC"; !is.na(ADLB.AVISITN); ADLB.AVISITN == 20; !is.na(ADLB.CHG); !is.na(ADLB.BASE); ADLB.AVISITN == 0
tlf-kmplot.r	tlf-kmplot-pilot5.pdf	ADSL.STUDYID; ADSL.USUBJID; ADSL.TRT01A;	ADSL.SAFFL == "Y"; ADSL.STUDYID == "CDISCPILOT01"; ADTTE.PARAMCD == "TTDE"; ADTTE.STUDYID == "CDISCPILOT01"
tlf-primary.r	tlf-primary-pilot5.rtf	ADADAS.EFFFL; ADADAS.ITTFL; ADADAS.PARAMCD; ADADAS.ANL01FL; ADADAS.TRTP; ADADAS.AVAL; ADADAS.AVISITN; ADADAS.CHG; ADADAS.TRTPN	ADAS.EFFFL == "Y"; ADAS.ITTFL == "Y"; ADAS.PARAMCD == "ACTOT"; ADAS.ANL01FL == "Y"; ADSL.EFFFL == "Y" & ADSL.ITTFL == "Y"; ADAS.AVISITN == 0; ADAS.AVISITN == 24

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 Open-source R Packages

Package	Version	Description
admiral	1.3.0	This R package provides tools for creating Clinical Data Interchange Standards Consortium (CDISC) compliant Analysis Data Model (ADaM) datasets, essential for submissions to the United States FDA, following the guidelines of the CDISC Analysis Data Model Implementation Guide.
cowplot	1.2.0	This package offers tools for enhancing 'ggplot2' with themes, plot alignment, complex figure arrangement, annotations, and image mixing, originally created for the Wilke lab and featured in the book "Fundamentals of Data Visualization."
diffdf	1.1.1	This package offers tools to comprehensively compare two data frames, detailing their differences and providing utilities to identify sources of discrepancies.
dplyr	1.1.4	The package provides a robust and consistent toolset for managing and manipulating data frame-like structures efficiently, both in-memory and out-of-memory.
emmeans	1.11.2	The package provides tools to obtain estimated marginal means (EMMs) for a variety of linear, generalized linear, and mixed models, along with functions to perform contrasts, trend analysis, and comparisons of slopes, as well as visualization options.
ggplot2	3.5.2	The package provides a declarative approach to creating graphics by allowing users to map data variables to aesthetics and specify graphical primitives, automating the intricate details based on the principles of "The Grammar of Graphics."
haven	2.5.5	The package facilitates importing foreign statistical file formats into R by leveraging the 'ReadStat' C library.

(continued)

Package	Version	Description
lubridate	1.9.4	The 'lubridate' package provides tools for fast and user-friendly parsing, extraction, updating, and algebraic manipulation of date-time and time-span objects in R.
metacore	0.2.0	The package provides an immutable container for metadata to enhance programming activities and functionality within the clinical programming workflow.
metatools	0.1.6	This package utilizes metadata information from 'metacore' objects to validate and construct metadata-related columns.
pharmaRTF	0.1.4	This package provides an enhanced RTF wrapper for R tables created with packages like 'Huxtable' or 'GT', allowing the addition of metadata and features essential for regulatory reports, such as multiple levels of titles, footnotes, landscape formatting, and margin control.
r2rtf	1.1.4	This package facilitates the creation of production-ready Rich Text Format (RTF) tables and figures with customizable formatting options.
rtables	0.6.13	The 'rtables' package provides a framework for creating complex, multi-level reporting tables with hierarchical, tree-like structures, enabling advanced data tabulation, grouping, and contextual summary computations.
stringr	1.5.1	The package provides a uniform, user-friendly set of wrappers for the 'stringi' package, ensuring consistent function and argument usage, seamless handling of "NA" values and zero length vectors, and facilitating easy integration between functions.
tidyr	1.3.1	The package "tidyr" provides tools for restructuring and cleaning data into a tidy format, with capabilities for pivoting, nesting, unnesting, handling nested lists, string extraction, and managing missing values.
Tplyr	1.2.1	The package is designed to streamline data manipulation processes for generating clinical summaries, with a focus on traceability.
visR	0.3.1	This package provides fit-for-purpose, reusable visualizations and tables tailored for clinical and medical research, incorporating sensible defaults and following established graphical principles.
xportr	0.4.3	The package provides tools to create CDISC-compliant datasets and verify their compliance with CDISC standards.
datasetjson	0.3.0	The package provides tools for reading, constructing, writing, and validating CDISC Dataset JSON files according to the Dataset JSON schema standards set by CDISC.

8 Directory Structure

Study datasets and the R programs are organized in accordance to Study Data Technical Conformance Guide.

```
├── m1
│   └── us
│       └── cover-letter.pdf
└── m5
    └── datasets
        └── rconsortiumpilot5
            ├── analysis
            │   └── adam
            │       ├── datasets
            │       │   ├── adadas.json
            │       │   ├── adae.json
            │       │   ├── adlbc.json
            │       │   ├── adrg.pdf
            │       │   ├── adsl.json
            │       │   └── adtte.json
            │       └── programs
            │           ├── adadas.r
            │           ├── adae.r
            │           ├── adlbc.r
            │           ├── adsl.r
            │           ├── adtte.r
            │           ├── renv-lock.txt
            │           ├── tlf-demographic.r
            │           ├── tlf-efficacy.r
            │           ├── tlf-kmplot.r
            │           └── tlf-primary.r
            └── tabulations
                └── sdtm
                    ├── ae.json
                    ├── cm.json
                    ├── dm.json
                    ├── ds.json
                    ├── ex.json
                    ├── lb.json
                    ├── mh.json
                    ├── qs.json
                    ├── relrec.json
```

```
├── sc.json
├── se.json
├── suppae.json
├── suppdm.json
├── suppds.json
├── supplb.json
├── sv.json
├── ta.json
├── te.json
├── ti.json
├── tv.json
└── vs.json
```

9 Appendix 1: Pilot 5 R Environment Installation and Usage

To execute the R programs included in this Pilot, follow all of the procedures below. Ensure that you note the location of where you downloaded the Pilot 5 eCTD submission files. For demonstration purposes, the procedures below assume the transfer has been saved to this location: `C:\pilot5`.

In addition, create a new directory to hold the unpacked Pilot 5 data files and associated programs. For demonstration purposes, the procedures below assume the new directory is this location: `C:\pilot5-files`.

9.1 Installation of R and RStudio

Download and install R 4.4.3 for Windows from https://cloud.r-project.org/bin/windows/base/old/4.4.3/R-4.4.3-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.

9.2 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 `4.4.3` 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"
```

9.3 Initialize R Program Execution Environment

The dependencies for executing the R programs are managed by the `renv` R package management system. To bootstrap the customized R package library, launch a new R session in the directory where you unpacked the source files in the previous step.

Launching RStudio

Create a new RStudio project within the `pilot5-files` directory using the following procedure:

- Launch RStudio
- Select `File -> New Project`
- In the **Create Project** dialog box, choose **Existing Directory**
- In the **Create Project from Existing Directory** dialog box, click the **Browse** button and navigate to the `C:\pilot5-files` directory.
- Once the location has been confirmed, click the **Create Project** button. A new directory called `.Rproj.user` and the project file `pilot5-files.Rproj` will appear in the directory.

> **Note**
>
> It is possible that the `.Rproj.user` folder may not have generated for you or or may not be visible as it is a hidden folder. If so, this is fine as it will not be necessary in order to run the R programs below.

9.4 Installation of R Packages

A minimum set of R packages are required to ensure the Pilot 5 R programs can be executed correctly. Use the following procedure to configure the Pilot 5 R package environment:

1. Run the following commands in the R console to install the `remotes` and `renv` packages:

```
install.packages("remotes")

# install version 1.1.4 of the renv package:
remotes::install_version("renv", version = "1.1.4")
```

> **Note**
>
> - *If you receive a warning showing "cannot open URL https://cran.rstudio.com/src/contrib/PACKAGES'", this is due to the default RStudio option 'Use secure download method for HTTP'. In RStudio, go to Tools → Global Options → Packages, then uncheck the 'Use secure download method for HTTP' option, then retry the installation.*

> **Note**
>
> If not already set, please verify that the working directory is already set to the project folder:
>
> - Run the following command in the R console: `getwd()`
> - If the output of this command does not match `C:\pilot5-files`, run the following command to set the working directory: `setwd("C:/pilot5-files")`

2. Move the `renv-lock.txt` file to the root project directory and rename the file to `renv.lock` by typing the following command in the R console:

```
file.copy(
  "C:/pilot5-files/m5/datasets/rconsortiumpilot5/analysis/adam/programs/renv-lock.txt",
  "C:/pilot5-files/renv.lock"
)
```

3. Restart the R Session in RStudio using the following methods:

- Select `Session -> Restart R`

4. Within the new R session, run the following command in the R console:

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

The function will prompt you to make a choice due to the lockfile being present. Enter `1` in the console to choose **Restore the project from the lockfile**.

4. To install the packages managed by `renv`, run the following command in the R console:

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

Due to certain R packages requiring compilation from their source versions, 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.

9.5 Execute R Programs

To reproduce the analysis results from the JSON transport files, set up and run the following programs in the order below:

1. Setting up .Rprofile

Edit the `.Rprofile` file created in the working directory to match the following contents:

> **TO DO**
>
> Ensure that the paths defined in the snippet below reflect the eCTD bundle structure

```
source("renv/activate.R")
Sys.setenv(RENV_DOWNLOAD_FILE_METHOD = "libcurl")

# File locations

path <- list(
  sdtm = file.path(getwd(), "m5/datasets/rconsortiumpilot5/tabulations/sdtm"),
  adam = file.path(getwd(), "m5/datasets/rconsortiumpilot5/analysis/adam/datasets"),
  output = file.path(getwd(), "m5/datasets/rconsortiumpilot5/analysis/adam/programs"),
  adam_json = file.path(getwd(), "m5/datasets/rconsortiumpilot5/analysis/adam/datasets"),
  programs = file.path(getwd(), "m5/datasets/rconsortiumpilot5/analysis/adam/programs")
)
```

2. Restart R Session
3. Using the source function, run the `pilot5-helper-fcns.r` program, which will load all helper functions for datasets and displays into your global environment.

```
source(file.path(path$programs, "pilot5-helper-fcns.r"))
```

4. Convert sdtm JSON files to rds files

```r
sdtm_files <- list.files(
  path = file.path(path$sdtm),
  pattern = "\\.json$",
  full.names = TRUE
)

convert_json_to_rds(sdtm_files, output_dir = file.path(path$sdtm))
```

5. Execute ADaM programs as seen in the order below:

- `adsl.r`
- `adadas.r`
- `adae.r`
- `adlbc.r`
- `adtte.r`

6. Execute Display programs

- `tlf-demographic.r`
- `tlf-efficacy.r`
- `tlf-kmplot.r`
- `tlf-primary.r`

10 Appendix 2

> **TO DO**
>
> Cross-check if anything has changed form Pilot 3 to Pilot 5 for QC Findings https://github.com/RConsortium/submissions-pilot5-datasetjson/wiki/QC-Findings