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
datasetjson	1.1

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.

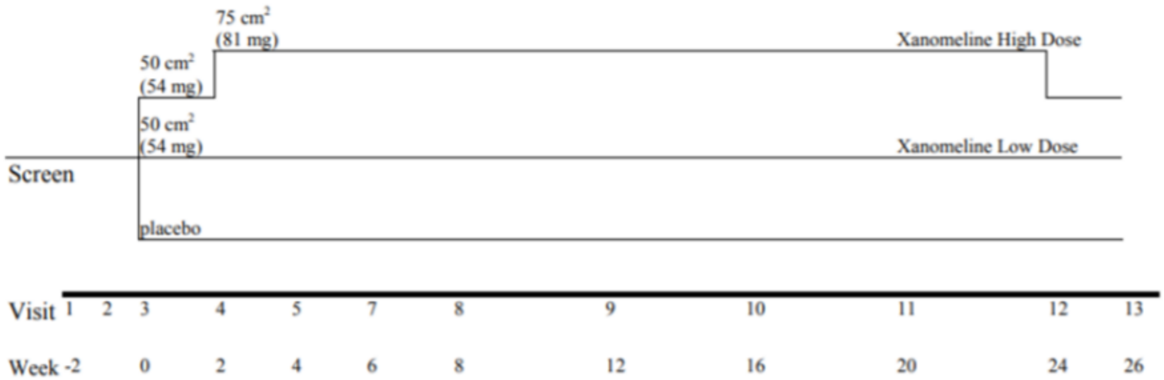
2.2.2 Methodology:

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 4.1.0
```

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

```
Yes
```

Was define.xml evaluated?

```
Yes
```

6.2 Issues Summary

The datasetjson 1.1 standard is not availabe in Pinnacle 21® Community (P21C) 4.1.0 or Pinnacle 21® Enterprise (P21E). The standard available in both P21C and P21E is datasetjson 1.0, which is not fit for purpose. See Appendix 3 for more details on differences between 1.1 and 1.0. Both P21C and P21E will be updated to use 1.1, but the timeline is TBD.

As a temporary workaround, FDA requested that Conformance Summaries should be generated for the following:

- ADaM json files converted to xpts and validated with P21C
- ADaM json files validated with datasetson 1.0 standard P21C

6.2.1 ADaM json files converted to xpts and validated with P21C

The xpts were generated as follows:

- ADaM programs read in sdtm `.json` files
- ADaM programs wrote out `.rds` files
- A program converted these `.rds` files to `.json` and then to `.xpt`
- `.xpt` files were validated by P21C

Check ID	Diagnostic Message	Dataset	Count	Explanation
No Issues were found				

6.3 ADaM json files validated with datasetson 1.0 standard P21C

The datasetson 1.0 `.json` files were generated as follows:

- ADaM programs read in sdtm `.json` files
  - `.json` files were created using datasetjson 1.1 standard
- ADaM programs wrote out `.rds` files
- A program converted these `.rds` files to `.json` and then to `.xpt`
  - `.json` files were created using datasetjson 1.1 standard
- `.xpt` files were converted to by P21C `.json` files using 1.0 standard
- `.json` files were validated by P21C

Check ID	Diagnostic Message	Dataset	Count	Explanation
AD0018	Variable label mismatch between dataset and ADaM standard	ADADAS	34	
AD0041A	*DT variable has missing SAS Date Format	ADADAS	3	

Check ID	Diagnostic Message	Dataset	Count	Explanation
AD0018	Variable label mismatch between dataset and ADaM standard	ADAE	40	
AD0041A	*DT variable has missing SAS Date Format	ADAE	4	
SD0059	Define.xml/dataset variable type mismatch	ADAE	5	
AD0018	Variable label mismatch between dataset and ADaM standard	ADLBC	34	
AD0041A	*DT variable has missing SAS Date Format	ADLBC	3	
AD0018	Variable label mismatch between dataset and ADaM standard	ADSL	23	
AD0041A	*DT variable has missing SAS Date Format	ADSL	5	
AD0320	Non-standard dataset label	ADSL	1	
AD0018	Variable label mismatch between dataset and ADaM standard	ADTTE	24	
AD0041A	*DT variable has missing SAS Date Format	ADTTE	4	

6.4 QC Findings and Common Issues

In this Pilot 5 study, our focus was to create SDTM and ADaM transport files as json rather than xpt file. We compared our R generated ADaMs against the Pilot 3 ADaMs, created in R, 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 Appendix 2.

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.

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

Program Name **Output Table Number** **Title**

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.
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.

(continued)

Package	Version	Description
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
            │           ├── pilot5-helper-fcns.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.

Download and install RStudio for Windows from 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, `Sys.which("make")` to verify that the installation of Rtools was successful:

```
> Sys.which("make")
[1] "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()
```

A first time installation and use of `renv::init()` will prompt the following text in the console. Please type `Y` to proceed.

A second prompt will appear similar to the text below for both new and old installations of `renv`:

```
> renv::init()

renv: Project Environments for R

Welcome to renv! It looks like this is your first time using renv.
This is a one-time message, briefly describing some of renv's functionality.

renv will write to files within the active project folder, including:

  - A folder 'renv' in the project directory, and
  - A lockfile called 'renv.lock' in the project directory.

In particular, projects using renv will normally use a private, per-project
R library, in which new packages will be installed. This project library is
isolated from other R libraries on your system.

In addition, renv will update files within your project directory, including:

  - .gitignore
  - .Rbuildignore
  - .Rprofile

Finally, renv maintains a local cache of data on the filesystem, located at:

  - "C:/Users/[redacted]/AppData/Local/R/cache/R/renv"

This path can be customized: please see the documentation in `?renv::paths`.

Please read the introduction vignette with `vignette("renv")` for more information.
You can browse the package documentation online at https://rstudio.github.io/renv/.
Do you want to proceed? [y/N]:
```

Figure 1: renv init

```
- "C:/Users/.../AppData/Local/R/cache/R/renv" has been created.
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
```

Enter `1` in the console to choose **Restore the project from the lockfile**.

5. If packages are not installed from the **Restore the project from the lockfile** step, then run the following command in the R console:

```r
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.

After all packages have been installed, you should Restart your Session.

- Select `Session -> Restart R`

A similar message should appear in your console. This indicates that your R Session is synced to all Pilot 5 packages needed to reproduce the Pilot 5 analysis and you should proceed to the next section.

```
Restarting R session...
- Project 'C:/pilot5-files' loaded. [renv 1.1.4]
```

If a message appears like below then the following guidance is recommended:

```
Restarting R session...
- Project '~/pilot5' loaded. [renv 1.1.4]
- The project is out-of-sync -- use `renv::status()` for details.
```

- Run `renv::status()` to find what is out of sync.
- Inspect the installation process for any packages that were not successfully installed.
- Repeat the above steps again.

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:

```r
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")
)
```

If the file is not present in your working directory, then in the files window click on `More > Show Hidden Files` and it should appear in the files window.

2. Restart R Session

- Select `Session -> Restart R`
- This will ensure that the list of paths in your Global Environment is populated.

Double check that path object has been created in your Global Environment using the following code `exists("path")`.

You should receive the following message in your console:

```
> exists("path")
[1] TRUE
```

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. The sdtm files are in json transport file format and need to be converted to rds files to run the ADaM programs.

Run the following code:

```
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`

You can use the following command to quickly execute each ADaM dataset. Just change the name of the dataset in the command. Rds files will be created for each ADaM in the `adamdata` folder and in your global environment.

```
source(file.path(path$programs, "adsl.r"))
```

6. Execute Display programs as seen in the order below:

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

Similarly to the ADaMs, you can run this command to quickly execute the display programs.

```
source(file.path(path$programs, "tlf-demographic.r"))
```

The newly run display outputs will be available in sub-folders in the `programs` folder called `out`, `pdf` and `rtf`. The programs `tlf-demographic.r` and `tlf-efficacy.r` will produce outputs in the `rtf` folder. The program `tlf-kmplot.r` will produce the output in the `pdf` folder. The `tlf-demographic.r` will produce the output in the `out` folder.

10 Appendix 2

The programs from Pilot 3 were minimally changed for Pilot 5. The biggest change was code for helping to produce the datasetjson transport files. The differences in the original CDISC pilot data and Pilot 3 are still present as well as a few additional issues that we will address.

10.1 Pilot 3 and Pilot 5 differences.

10.1.1 Classes

All datasets have an issue where there are differences in classes for numeric/integer variables. We will use the `ADTTE` dataset as an example, but these issues persist for all five datasets.

```
There are columns in BASE = Pilot 5 and COMPARE = Pilot 3 with different classes !!
  ==================================
   VARIABLE  CLASS.BASE  CLASS.COMP
  ----------------------------------
     AGE      integer     numeric
   AGEGR1N    integer     numeric
     AVAL     integer     numeric
     CNSR     integer     numeric
    RACEN     integer     numeric
    SRCSEQ    integer     numeric
    TRTAN     integer     numeric
    TRTDUR    integer     numeric
  ----------------------------------
```

Pilot 3 data was written out as numeric. For Pilot 5, the datasetjson R package writes out data use Type from spec for integers and writes out the data as integer. We do not think this is an issue, but is being discussed with the datasetjson R package team.

10.1.2 Formats

All datasets have an issue where there are differences in attributes for SAS formats.

```
There are columns in BASE = Pilot 5 and COMPARE = Pilot 3 with differing attributes !!
  ================================================
   VARIABLE  ATTR_NAME   VALUES.BASE  VALUES.COMP
  ------------------------------------------------
     ADT     format.sas     NULL         DATE9
   STARTDT   format.sas     NULL         DATE9
    TRTEDT   format.sas     NULL         DATE9
    TRTSDT   format.sas     NULL         DATE9
  ------------------------------------------------
```

Pilot 3 data has the DATE9 format associated with it as an attribute. For Pilot 5, the datasetjson R package does not associate the format with the variable. We do not think this is an issue, but is being discussed with the datasetjson R package team.

10.1.3 NAs introduced by coercion.

```
Warning messages:
1: In lapply(d[dbl_cols], as.double) : NAs introduced by coercion
2: In lapply(d[dbl_cols], as.double) : NAs introduced by coercion
3: In lapply(d[dbl_cols], as.double) : NAs introduced by coercion
4: In lapply(d[dbl_cols], as.double) : NAs introduced by coercion
...
```

Using the datasetjson R package, a warning message is produced when the json files are read in. We do not think this is an issue, but is being discussed with the datasetjson R package team.

10.2 Pilot 3 and CDISC Pilot differences.

10.3 ADSL

The R-generated ADSL matches the original ADSL from CDISC pilot data, besides the following mismatches: * Subject 01-702-1082 has a missing value for BMIBLGR1 in the R-generated ADSL, whilst BMIBLGR1 = "<25" in the original ADSL. This is an issue with the original ADSL, as this subject's BMI at baseline (BMIBL) is missing and therefore the subject shouldn't be assigned a BMI at baseline group.

10.4 ADAE

The R-generated ADAE matches the original ADAE from CDISC pilot data, besides the following mismatches: There is an issue with the original CDISC pilot dataset. ADURN is blank, where AESEQ is (5, 6, 7, 8) for the original CDISC dataset for Subject below:

```
> adae_orig %>%
     filter(USUBJID=='01-716-1418') %>%
     select(USUBJID,TRTSDT,ASTDT,AENDT,ADURN,ADURU,AESEQ)

# A tibble: 10 × 7
   USUBJID     TRTSDT     ASTDT      AENDT      ADURN ADURU AESEQ
   <chr>       <date>     <date>     <date>     <dbl> <chr> <dbl>
 1 01-716-1418 2013-05-05 2013-05-05 2013-05-07     3 DAY       1
 2 01-716-1418 2013-05-05 2013-05-05 NA            NA NA        2
 3 01-716-1418 2013-05-05 2013-05-05 2013-05-07     3 DAY       3
 4 01-716-1418 2013-05-05 2013-05-07 NA            NA NA        4
 5 01-716-1418 2013-05-05 2013-07-01 2013-09-26    NA NA        5
 6 01-716-1418 2013-05-05 2013-07-01 2013-10-04    NA NA        6
```

```
 7 01-716-1418 2013-05-05 2013-07-01 2013-09-26    NA NA        7
 8 01-716-1418 2013-05-05 2013-07-01 2013-10-04    NA NA        8
 9 01-716-1418 2013-05-05 2013-09-26 2013-11-11    47 DAY       9
10 01-716-1418 2013-05-05 2013-09-26 2013-11-11    47 DAY      10
```

Because it seems the original SDTM.AE.AESTDTC was missing Day, where it seems the original ADAE derivation for ADURN was probably using this date instead of the imputed date. Because day is missing in AESTDTC, ADURN can't derive days.

```
> ae %>%
    filter(USUBJID=='01-716-1418') %>%
    select(USUBJID,AESTDTC,AESEQ) %>%
    arrange(AESEQ)
# A tibble: 10 × 3
   USUBJID     AESTDTC    AESEQ
   <chr>       <chr>      <dbl>
 1 01-716-1418 2013-05-05     1
 2 01-716-1418 2013-05-05     2
 3 01-716-1418 2013-05-05     3
 4 01-716-1418 2013-05-07     4
 5 01-716-1418 2013-07        5
 6 01-716-1418 2013-07        6
 7 01-716-1418 2013-07        7
 8 01-716-1418 2013-07        8
 9 01-716-1418 2013-09-26     9
10 01-716-1418 2013-09-26    10
```

but the same records, derived in the Pilot 3 dataset do show a calculation since we are using the imputed ASTDT, per the define (ADURN=AENDT-ASTDT+1).

```
# AE.AESTDTC, converted to a numeric SAS date. Some events with partial dates are
# imputed in a conservative manner. If the day component is missing, a value of
# '01' is used. If both the month and day are missing no imputation is performed
# as these dates clearly indicate a start prior to the beginning of treatment.
# There are no events with completely missing start dates.

> adae0 %>%
    filter(USUBJID=='01-716-1418') %>%
    select(USUBJID,TRTSDT,ASTDT,AESTDTC,AENDT,AEENDY,ADURN,ADURU,AESEQ)
# A tibble: 10 × 9
   USUBJID     TRTSDT     ASTDT      AESTDTC    AENDT      AEENDY ADURN ADURU AESEQ
   <chr>       <date>     <date>     <chr>      <date>      <dbl> <dbl> <chr> <dbl>
```

```
 1 01-716-1418 2013-05-05 2013-05-05 2013-05-05 2013-05-07      3     3 DAY        1
 2 01-716-1418 2013-05-05 2013-05-05 2013-05-05 NA             NA    NA NA         2
 3 01-716-1418 2013-05-05 2013-05-05 2013-05-05 2013-05-07      3     3 DAY        3
 4 01-716-1418 2013-05-05 2013-05-07 2013-05-07 NA             NA    NA NA         4
 5 01-716-1418 2013-05-05 2013-07-01 2013-07    2013-10-04    153    96 DAY        6
 6 01-716-1418 2013-05-05 2013-07-01 2013-07    2013-10-04    153    96 DAY        8
 7 01-716-1418 2013-05-05 2013-07-01 2013-07    2013-09-26    145    88 DAY        5
 8 01-716-1418 2013-05-05 2013-07-01 2013-07    2013-09-26    145    88 DAY        7
 9 01-716-1418 2013-05-05 2013-09-26 2013-09-26 2013-11-11    191    47 DAY        9
10 01-716-1418 2013-05-05 2013-09-26 2013-09-26 2013-11-11    191    47 DAY       10
```

This latter approach should be the correct approach.

Due to this, we have outlined the expected differences here :

```
> diffdf(adae, adae_orig, keys = c("STUDYID", "USUBJID", "AESEQ"))
Differences found between the objects!

A summary is given below.

There are columns in BASE and COMPARE with differing attributes !!
All rows are shown in table below
```

1. ADURN values will be populated in Pilot 3 (i.e. under `BASE`), following the latter derivation approach (i.e. ADURN=AENDT-ASTDT+1) for Subject 01-716-1418 where AESEQ is (5, 6, 7, 8) specified in define.

```
All rows are shown in table below


  ==========================================================
   VARIABLE    STUDYID       USUBJID    AESEQ  BASE  COMPARE
  ----------------------------------------------------------
    ADURN    CDISCPILOT01  01-716-1418    5     88    <NA>
    ADURN    CDISCPILOT01  01-716-1418    6     96    <NA>
    ADURN    CDISCPILOT01  01-716-1418    7     88    <NA>
    ADURN    CDISCPILOT01  01-716-1418    8     96    <NA>
  ----------------------------------------------------------
```

2. ADURU should be set to 'DAYS' (i.e. under `BASE`) instead of 'DAY' when ADURN is not missing. Updated in Pilot 3 define.

```
First 10 of 718 rows are shown in table below

  ===========================================================
   VARIABLE    STUDYID       USUBJID    AESEQ  BASE  COMPARE
  -----------------------------------------------------------
    ADURU    CDISCPILOT01  01-701-1015    3    DAYS    DAY
    ADURU    CDISCPILOT01  01-701-1023    1    DAYS    DAY
    ADURU    CDISCPILOT01  01-701-1023    4    DAYS    DAY
    ADURU    CDISCPILOT01  01-701-1047    1    DAYS    DAY
    ADURU    CDISCPILOT01  01-701-1047    2    DAYS    DAY
    ADURU    CDISCPILOT01  01-701-1097    2    DAYS    DAY
    ADURU    CDISCPILOT01  01-701-1097    3    DAYS    DAY
    ADURU    CDISCPILOT01  01-701-1097    5    DAYS    DAY
    ADURU    CDISCPILOT01  01-701-1097    6    DAYS    DAY
    ADURU    CDISCPILOT01  01-701-1097    7    DAYS    DAY
  -----------------------------------------------------------
```

10.5 ADLBC

The R-generated ADLBC matches the original ADLBC from CDISC pilot data, besides the following mismatches:

Three variables from R-generated ADLBC have class date while the same variables are numeric in the CDISC ADLBC. We opted to keep the date class in our R-generated ADLB.

```
> diffdf(adlbc, qc_adlbc, keys = c("STUDYID", "USUBJID", "AVISIT", "LBSEQ"))
Differences found between the objects!

A summary is given below.

There are columns in BASE and COMPARE with different classes !!
All rows are shown in table below

  ==================================
   VARIABLE  CLASS.BASE  CLASS.COMP
  ----------------------------------
     ADT        Date      numeric
    TRTEDT      Date      numeric
    TRTSDT      Date      numeric
  ----------------------------------
```

10.6 ADADAS

The R-generated ADADAS matches original ADADAS from CDISC pilot data, except for the records where `PARAMCD=ACTOT, DTYPE=LOCF`. This is an issue from the CDISC ADADAS.

- CDISC SDTM/QS: 818 records for `QSTESTCD=ACTOT`
- CDISC ADaM/ADADAS: 1040 records for `PARAMCD=ACTOT`, 799 (directly from QS, **should be 818**) + 241 imputed records (`DTYPE=LOCF`)
- ADADAS generated by R: 1040 records for `PARAMCD=ACTOT`, 818 (directly from QS) + 222 imputed records (`DTYPE=LOCF`)

Take a detailed example USUBJID="01-701-1294"

CDISC QS:

```
> qs %>% filter(QSTESTCD=="ACTOT") %>%
+   select(USUBJID, QSSEQ, VISIT, QSTESTCD, QSTEST,QSSTRESN) %>%
+   filter(USUBJID=="01-701-1294")
# A tibble: 4 × 6
  USUBJID     QSSEQ VISIT     QSTESTCD QSTEST                QSSTRESN
  <chr>       <dbl> <chr>     <chr>    <chr>                    <dbl>
1 01-701-1294  5015 BASELINE  ACTOT    ADAS-COG(11) Subscore        9
2 01-701-1294  5030 WEEK 8    ACTOT    ADAS-COG(11) Subscore       14
3 01-701-1294  5045 WEEK 12   ACTOT    ADAS-COG(11) Subscore        6
4 01-701-1294  5060 RETRIEVAL ACTOT    ADAS-COG(11) Subscore        9
```

CDISC ADADAS:

For the record with `QSSEQ=5045` and `AVISIT=Week 8`, `DTYPE` is populated as `LOCF` , but this record is directly from `qs` dataset, not imputed.

```
> qc_adadas %>% filter(PARAMCD=="ACTOT") %>%
+   select(USUBJID, QSSEQ, PARAMCD, AVISITN, AVISIT, VISIT, AVAL, DTYPE, ANL01FL, ADT, ADY) %>%
+   arrange(USUBJID, AVISITN) %>%
+   filter(USUBJID=="01-701-1294")
# A tibble: 5 × 11
  USUBJID     QSSEQ PARAMCD AVISITN AVISIT   VISIT      AVAL DTYPE  ANL01FL ADT          ADY
  <chr>       <dbl> <chr>     <dbl> <chr>    <chr>     <dbl> <chr>  <chr>   <date>     <dbl>
1 01-701-1294  5015 ACTOT         0 Baseline BASELINE      9 ""     "Y"     2013-03-24     1
2 01-701-1294  5030 ACTOT         8 Week 8   WEEK 8       14 ""     "Y"     2013-05-22    60
3 01-701-1294  5045 ACTOT         8 Week 8   WEEK 12      14 "LOCF" ""      2013-06-14    83
4 01-701-1294  5045 ACTOT        16 Week 16  WEEK 12      14 "LOCF" "Y"     2013-06-14    83
5 01-701-1294  5060 ACTOT        24 Week 24  RETRIEVAL     9 ""     "Y"     2013-10-08   199
```

ADADAS generated by R:

`DTYPE` is not `LOCF` for the record with `QSSEQ=5045` and `AVISIT=Week 8`, as this record is directly from `qs`.

```
> adadas %>% filter(PARAMCD=="ACTOT") %>%
+       select(USUBJID, QSSEQ, PARAMCD, AVISITN, AVISIT, VISIT, AVAL, DTYPE, ANL01FL, ADT, ADY) %>%
+       arrange(USUBJID, AVISITN) %>% filter(USUBJID=="01-701-1294")
# A tibble: 5 × 11
  USUBJID     QSSEQ PARAMCD AVISITN AVISIT   VISIT      AVAL DTYPE  ANL01FL ADT          ADY
  <chr>       <dbl> <chr>     <dbl> <chr>    <chr>     <dbl> <chr>  <chr>   <date>     <dbl>
1 01-701-1294  5015 ACTOT         0 Baseline BASELINE      9 ""     "Y"     2013-03-24     1
2 01-701-1294  5030 ACTOT         8 Week 8   WEEK 8       14 ""     "Y"     2013-05-22    60
3 01-701-1294  5045 ACTOT         8 Week 8   WEEK 12       6 ""     ""      2013-06-14    83
4 01-701-1294  5030 ACTOT        16 Week 16  WEEK 8       14 "LOCF" "Y"     2013-05-22    60
5 01-701-1294  5060 ACTOT        24 Week 24  RETRIEVAL     9 ""     "Y"     2013-10-08   199
```

The same issue occurred for other subjects and resulted in the following discrepancies:

```
There are rows in BASE that are not in COMPARE !!
First 10 of 33 rows are shown in table below

  ===========================================
     USUBJID     PARAMCD   AVISIT      ADT
  -------------------------------------------
   01-701-1294    ACTOT    Week 16   2013-05-22
   01-701-1302    ACTOT    Week 16   2013-10-22
   01-703-1076    ACTOT    Week 16   2013-12-17
   01-703-1076    ACTOT    Week 24   2013-12-17
   01-704-1010    ACTOT    Week 24   2014-06-13
   01-704-1065    ACTOT    Week 16   2013-12-20
   01-704-1065    ACTOT    Week 24   2013-12-20
   01-704-1120    ACTOT    Week 16   2014-01-27
   01-704-1120    ACTOT    Week 24   2014-01-27
   01-705-1310    ACTOT    Week 16   2013-12-26
  -------------------------------------------

There are rows in COMPARE that are not in BASE !!
First 10 of 33 rows are shown in table below

  ===========================================
     USUBJID     PARAMCD   AVISIT      ADT
  -------------------------------------------
```

```
   01-701-1294   ACTOT   Week 16  2013-06-14
   01-701-1302   ACTOT   Week 16  2013-11-05
   01-703-1076   ACTOT   Week 16  2013-12-24
   01-703-1076   ACTOT   Week 24  2013-12-24
   01-704-1010   ACTOT   Week 24  2014-07-09
   01-704-1065   ACTOT   Week 16  2013-12-24
   01-704-1065   ACTOT   Week 24  2013-12-24
   01-704-1120   ACTOT   Week 16  2014-02-03
   01-704-1120   ACTOT   Week 24  2014-02-03
   01-705-1310   ACTOT   Week 16  2014-01-23
  -------------------------------------------

Not all Values Compared Equal
All rows are shown in table below

  =============================
   Variable  No of Differences
  -----------------------------
     AVAL           19
     CHG            19
     PCHG           19
    DTYPE           19
  ----------------------------
```

In the CDISC ADADAS, there are 19 subjects whose records have the incorrect `DTYPE=LOCF` value instead of the expected missing `DTYPE`, resulting IN different `AVAL/CHG/PCHG` values for these subjects.

```
> diff <- diffdf(adadas, qc_adadas, keys = c("USUBJID", "PARAMCD", "AVISIT", "ADT"))
> count(diff$VarDiff_AVAL, USUBJID)
# A tibble: 19 × 2
   USUBJID         n
   <chr>       <int>
 1 01-701-1294     1
 2 01-701-1302     1
 3 01-703-1076     1
 4 01-704-1065     1
 5 01-704-1120     1
 6 01-705-1292     1
 7 01-705-1310     1
 8 01-708-1347     1
 9 01-709-1102     1
```

```
10 01-709-1259     1
11 01-710-1045     1
12 01-710-1278     1
13 01-710-1300     1
14 01-710-1315     1
15 01-714-1068     1
16 01-715-1107     1
17 01-716-1373     1
18 01-718-1172     1
19 01-718-1250     1
```

10.7 ADTTE

The R-generated ADTTE matches original ADTTE from CDISC pilot data except for minor SAS format discrepancies. Since this adtte was generated in R compared to SAS formats, the columns Type & Length in the define should be sufficient enough to describe the attributes of these variables.

```
> diffdf(adtte, qc_adtte, keys = c("STUDYID", "USUBJID", "PARAMCD", "SRCDOM", "STARTDT"))
Differences found between the objects!

A summary is given below.

There are columns in BASE and COMPARE with differing attributes !!
First 10 of 20 rows are shown in table below

  ================================================
   VARIABLE  ATTR_NAME   VALUES.BASE  VALUES.COMP
  ------------------------------------------------
     AGE     format.sas     NULL           3
    AGEGR1   format.sas     NULL          $5
   AGEGR1N   format.sas     NULL           3
   EVNTDESC  format.sas     NULL          $25
    PARAM    format.sas     NULL          $32
   PARAMCD   format.sas     NULL          $4
     RACE    format.sas     NULL          $32
    RACEN    format.sas     NULL           3
    SAFFL    format.sas     NULL          $1
     SEX     format.sas     NULL          $1
  ------------------------------------------------
```

10.8 Label discrepancies

In pilot3, variable labels were updated per ADaM IG 1.1, which caused some discrepancies with original CDISC pilot data label.

Dataset	Variable	CDISC pilot data label	Pilot3 label
ADAE	ADURN	Analysis Duration (N)	AE Duration (N)
	ADURU	Analysis Duration Units	AE Duration Units
	AOCCFL	1st Occurrence of Any AE Flag	1st Occurrence within Subject Flag
ADADAS	ANL01FL	Analysis Record Flag 01	Analysis Flag 01
	ITTFL	Intent-to-Treat Population Flag	Intent-To-Treat Population Flag
ADTTE	SRCDOM	Source Data	Source Domain

11 Appendix 3

Below is a brief comparison of the differences between the datasetjson 1.0 and 1.1 standard. Please see Citations for additional information on these differences.

11.1 Comparison of Dataset-JSON v1.0 vs v1.1

Aspect	v1.0	v1.1
Release Date	23 August 2023	5 December 2024 (See Citation 1)
Standard Context	Part of ODM v2.0	Independent standard (See Citation 2)
Structure	Nested JSON	Flattened structure (See Citation 3)
Datatype Support	Basic types	Expanded types (See Citation 4)
Target Data Type Conversion	Not defined	Introduced `targetDataType`(See Citation 4)
Missing Value Representation	Ambiguous	`null` and `""`(See Citation 4)
NDJSON Support	Not supported	Supported (See Citation 4)
Compression Format	None	DSJC format (See Citation 6)
OID Requirements	Mostly required	Optional (See Citation 4)
ITEMGROUPDATASEQ	Present	Removed (See Citation 5)
isReferenceData Attribute	Included	Removed (See Citation 5)
Define-XML Linkage	Supported	Enhanced (See Citation 2)
Tooling & Viewer Support	Limited	Updated tools (See Citation 4)
User Guide	Minimal	Comprehensive (See Citation 4)
API Specification	Not available	Draft introduced (See Citation 6)
Public Review Feedback	Not applicable	58 issues reviewed (See Citation 5)

Citation 1 : CDISC (2023). *Dataset-JSON v1.0.* Available at: https://www.cdisc.org/standards/data-exchange/dataset-json/dataset-json-v1-0 [Accessed 15 Sep. 2025].

Citation 2 : CDISC (2024). *Dataset-JSON v1.1*. Available at: https://www.cdisc.org/standards/data-exchange/dataset-json/dataset-json-v1-1 [Accessed 15 Sep. 2025].

Citation 3 : CDISC (2024). Dataset-JSON v1.1 Public Review Presentation. Available at: https://www.cdisc.org/sites/default/files/pdf/Dataset-JSON-v1-1-Public-Review.pdf [Accessed 15 Sep. 2025].

Citation 4 : CDISC (2024). Dataset-JSON v1.1 GGG Final Presentation. Available at: https://github.com/user-attachments/files/22663348/Dataset-JSON-v1-1-GGG-Final.pptx [Accessed 15 Sep. 2025].

Citation 5 : CDISC (2024). Public Review Metrics and Issue Dispositions. Available at: https://github.com/user-attachments/files/22663348/Dataset-JSON-v1-1-GGG-Final.pptx [Accessed 15 Sep. 2025].

Citation 6 : CDISC (2025). *Dataset-JSON v1.1 API v1.0 Standard and Compressed Dataset-JSON v1.1*. Available at: https://www.cdisc.org/public-review/dataset-json-v1-1-api-v1-0-standard-and-compressed-dataset-json-v1-1 [Accessed 15 Sep. 2025].