

DS03:Charting New Territory: R-Based Submissions Made Easy

Veena Terekar

Johnson & Johnson

Date : 19th Feb – 21st Feb 2026

Created for PHUSE APAC Connect 2026

Final Version

Disclaimer



This presentation is for informational purposes only and does not represent professional guidance or advice. Any views and opinions expressed during this presentation are those of the presenters and do not necessarily reflect the views or policies of Johnson & Johnson, or any other company in the Johnson & Johnson Family of Companies.



The PowerPoint slides are the intellectual property of the authors and are protected under the copyright laws of the United States of America and other countries. Used by permission. All rights reserved. All other trademarks are the property of their respective owners



This presentation may not be reproduced, modified, or copied, in whole or in part, without the express written permission of Johnson & Johnson.

Agenda



R Mock Submission-Regulatory Pathway



Cross-Function Collaboration & Submission Planning



Constructing ADRG for Hybrid SAS/R Submission



Challenges/Findings

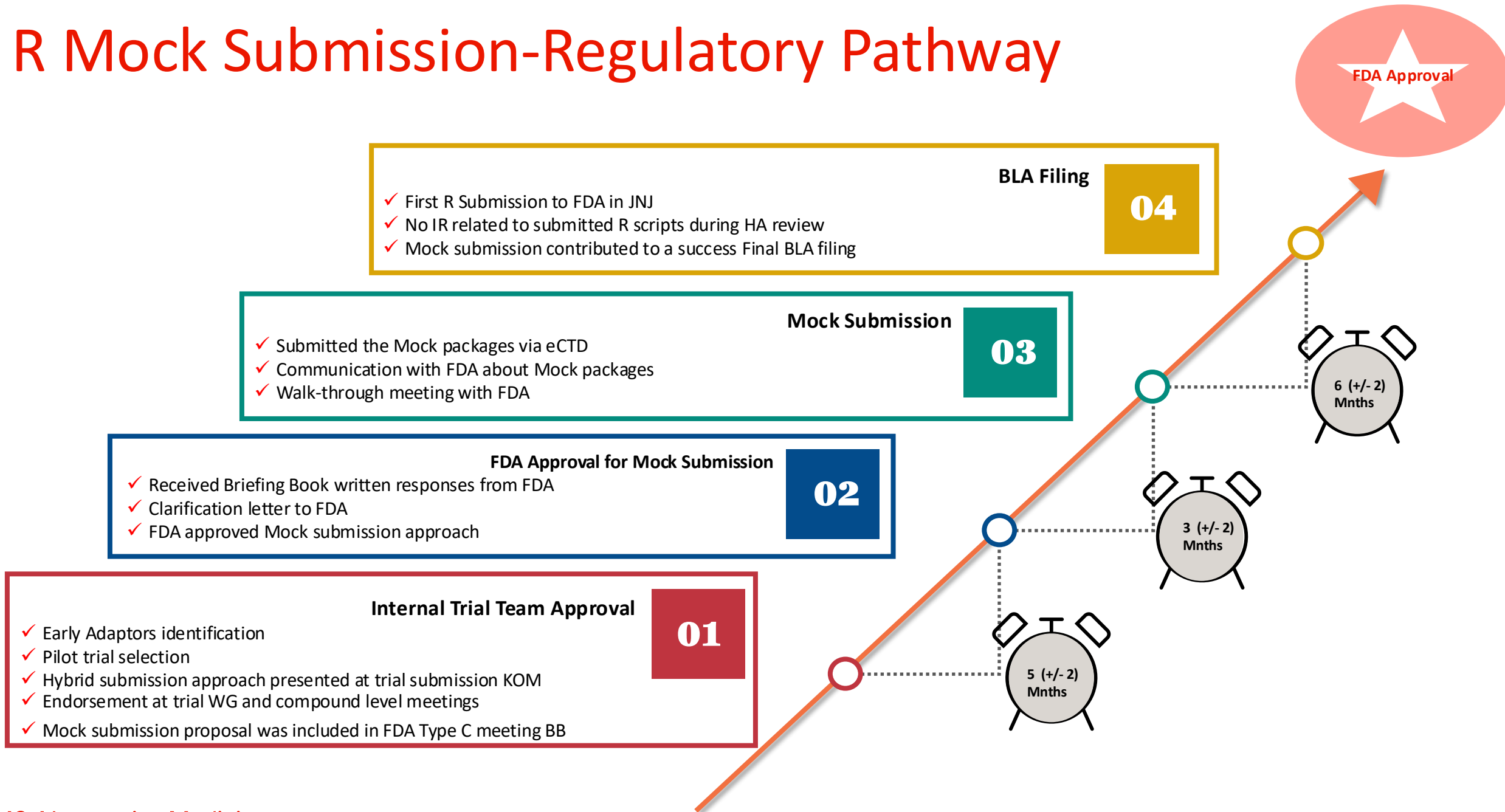


Conclusion



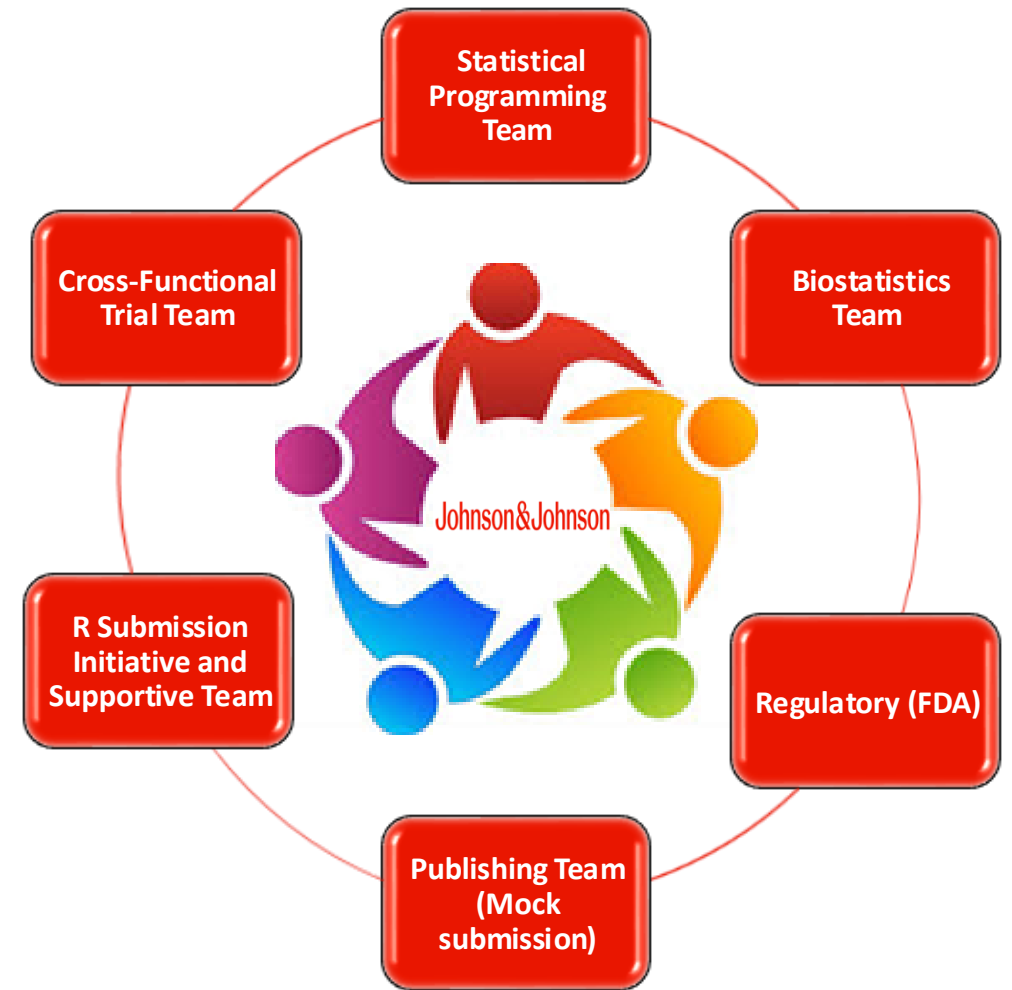
References

R Mock Submission-Regulatory Pathway



Cross-Function collaboration throughout the Journey

- Collaborated with the Publishing team to develop a Dossier plan specifically for the Mock package
- Worked with the Regulatory team on the submission cover letter for the Mock package. A single PDF combining individual .rtf files was attached to the cover letter in Module 1 due to the absence of CSR
- Submitted a standard eSub package containing trial dummy information in Module 5
- Mock package undergone a formal eCTD validation process and received confirmation from the FDA



Submission Planning: Hybrid SAS/R Programs

- All ADaM datasets are generated using SAS.
- All Tables/figures/Listings (TFLs) are created in R.

- All R packages are open-source.

Johnson & Johnson's Open Source Journey in R:

<https://posit.co/webinar/johnson-johnsons-open-source-journey-in-r>

- All R packages are validated following JNJ GxP.

Using R in a GxP environment:

<https://www.youtube.com/watch?v=IWXqfuaxNL8>

- Key R packages used:

tidyr, dplyr, tidyTLG, ggplot2, epiR, survival

- Utility Packages:

envsetup

Template for R program for Analysis output

Develop R scripts that reproduce the clear analysis logic end-to-end

```
#-----Libraries-----  
library(dplyr)  
library(haven)  
library(tidytlg)  
library(stringr)
```

```
##-----define the population flag-----  
popfl<-"EVALFL"  
  
##define the ID of TLGs  
tblid<-"TSIDEM01"
```

```
#-----process data-----  
adsl<-read_sas(read_path(a_in,"adsl.sas7bdat"))%>%  
  rename_all(toupper)
```

```
#-----generate results-----  
## Analysis set row  
t1<-adsl%>%  
  freq(colvar = 'colnbr',  
        rowvar = popfl,  
        rowtext = 'Analysis set: Evaluable',  
        statlist=statlist("n"))
```

```
#-----bind together-----  
tbl<-bind_table(t1,t2,t3,t4,t5,t6,t7,t8,  
               column_metadata_file = read_path(dpspath, "column_metadata.xlsx"),  
               tbltype = "CL")
```

```
#-----output results-----  
gentlg(huxme = tbl,  
       file = tblid,  
       orientation = "portrait",  
       opath=opath[[environ]],  
       title_file = read_path(dpspath, "titles.xlsx"))
```

← Loading the required packages

← Define filters and population flags

← Process data and generate results

← Apply Column Metadata

← Use titles file and generate final output

Constructing ADRG for Hybrid SAS/R submission

Section 7

7. Submission of Programs

7.1 ADaM Programs

7.2 Analysis Output Programs

7.3 Macro Programs

7.4 Packages Used

7.5 Common R Functions Used

7.1 ADaM Programs

Program name	Output	Inputs	Macros and functions used
adsl.sas	Analysis Dataset ADSL	adsl, ae, cm, dm, ds, ec, eg, ex, fa, ho, is, lb, mh, pc, pr, qs, suppdm, sv, vs	%adadurn, %adady, %adarfendit, %adarfendtv1, %adarfsttdt, %adclear, %addtm, %adeot_upd, %adsupp, %adstaltvdt, %adstsvdt, %admetries, %adpopfl, %adranddt, %adranddtv1, %adsaffl, %adsafflv1, %adsubdosuma, %adtrta, %adtrtdt, %adtrtp, %aexist, %asr, %bmi, %bslval_upd, %durfup, %gmgdur, %lsvisit, %ltvisit, %mg_for-

7.2 Analysis Output Programs

Program name	Output	Inputs	Macros and functions used
tsfae01d.r	Table TSFAE01D	adae, adsl, column_metadata.xlsx*, dd, ds, relrec, titles.xlsx*	base::%in%, base::as.numeric, base::c, base::data.frame, base::eval, base::grep, base::ifelse, base::is.na, base::lapply, base::merge, base::nrow, base::paste0, base::source, dplyr::add_row, dplyr::bind_rows, dplyr::case_when, dplyr::ensym, dplyr::expr, dplyr::filter, dplyr::if_else, dplyr::left_join, dplyr::mutate, dplyr::mutate_if, dplyr::na_if, dplyr::rename_all

7.3 Macro Programs

Program Name	Purpose
acritcm-sas	Macro to create critX(fl) flags for various gMG therapy adjustments.
adablfl-sas	Macro to create baseline flag: ABLFL variable.
adablflv1-sas	Macro to create Analysis baseline flag Version 1.
adadiactdu-sas	Macro to create analysis variables related to dose adjustment and action

7.4 Packages Used

For open source packages in CRAN or github, the full descriptions can be found in more detail.

Package Name	Package Version	Package Description
envsetup	0.0.1.0.0.0.1662505071	Support R environment setup for table/listing/graph (TLG) creation including path of read in and out of datasets or files.
dplyr	1.0.10	Data manipulation such as filtering, selecting, adding, deleting and summarizing operations, etc.
haven	2.5.1	Read and write various data formats including .sas7bdat.
forcats	0.5.2	Tools for working with categorical variables (factors) including reordering factor levels and modifying factor levels, etc.

7.5 Common R Functions Used:

For functions from open-source packages in CRAN or [github](#), the full descriptions and usage can be found in more detail.

Function Name	Package Name	Function Description
read_path	envsetup	Check each environment for the file and return the path to the first.
univar	tidytlg	Function to produce univariate-type summary statistics for numeric variables.
univar_remove0	N/A	Removes time points from result of tidytlg::univar() function where the total count is zero.

Appendix

The appendix is added to facilitate the R environment setup, including installation of required R packages. It provides guidance on how tables, figures, and listings developed in R can be executed to reproduce the analysis report in RTF format independently of the sponsor's computational system.

	Appendix 1: Set-up R environment and Analysis
1	1. Installation of R and RStudio
2	2. Define Working Directory
	3. Set up the Working Environment
	4. Create a new RStudio Project to
	5. Reproduce the Working Environment
3	6. Installation of Required Packages
	7. Create a renv-lock File
4	8. Set the paths for analysis
5	9. Execute the analysis programs

Installation of R and RStudio

Whether you are setting up R and RStudio for the first time or already have them installed, please ensure you are using the versions specified to ensure reproducibility of the analysis.

- I. Download and install R version for Windows from [cran](#) (provide link)
- II. Download RStudio Desktop for Windows by visiting <https://dailies.rstudio.com/version/>.

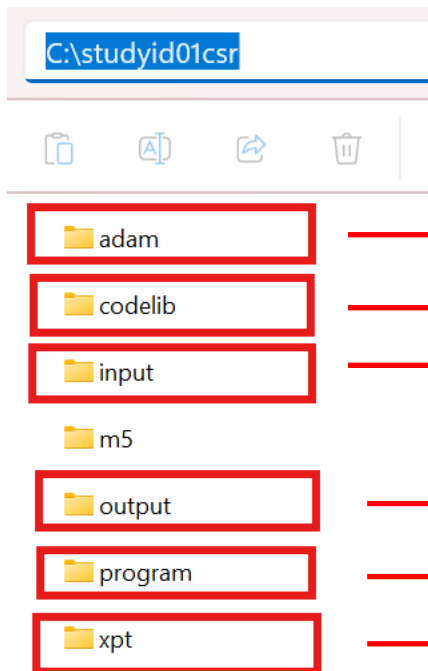
- **Step 2,3,4 and 5:** Provide instructions to create R project and setup the folder structure same as in the sponsor computing system.
- **Step 6:** All package installations are conducted through a single file, 'packages.txt', which contains all the necessary R packages mirrored from our container in the computing system embedded in Linux. The 'packages.txt' file is included in the submission package.

```
renv::init(bare = TRUE, repos = "https://packagemanager.posit.co/cran/2022-09-15")
pkgs <- read.csv(file.path(getwd(), "m5/misc/packages.txt"))
install.packages(pkgs$Package, repos = "https://packagemanager.posit.co/cran/2022-09-15")
```
- **Step 7:** Create a renv-lock file. This file capture packages versions to ensure the stability of the R environment recently configured on the reviewer's end.

```
renv::snapshot(type = "all", repos = "https://packagemanager.posit.co/cran/2022-09-15")
```
- **Step 8:** Source data libraries and paths utilized in R TFL scripts are configured via a file named 'envsetup.txt', which includes the folder path information.

ADRG: Define Working Directory

Create a working directory, for example, “C:\studyid01csr” in Window system. Copy the submission folder (“m5\”) into this folder



The repository for datasets in SAS7BDAT format used by the table, figure, and listing programs. The datasets in SAS7BDAT format will be converted from XPT files located in the ‘xpt’ folder.

R functions files converted from -r.txt to original extension .r

Metadata files

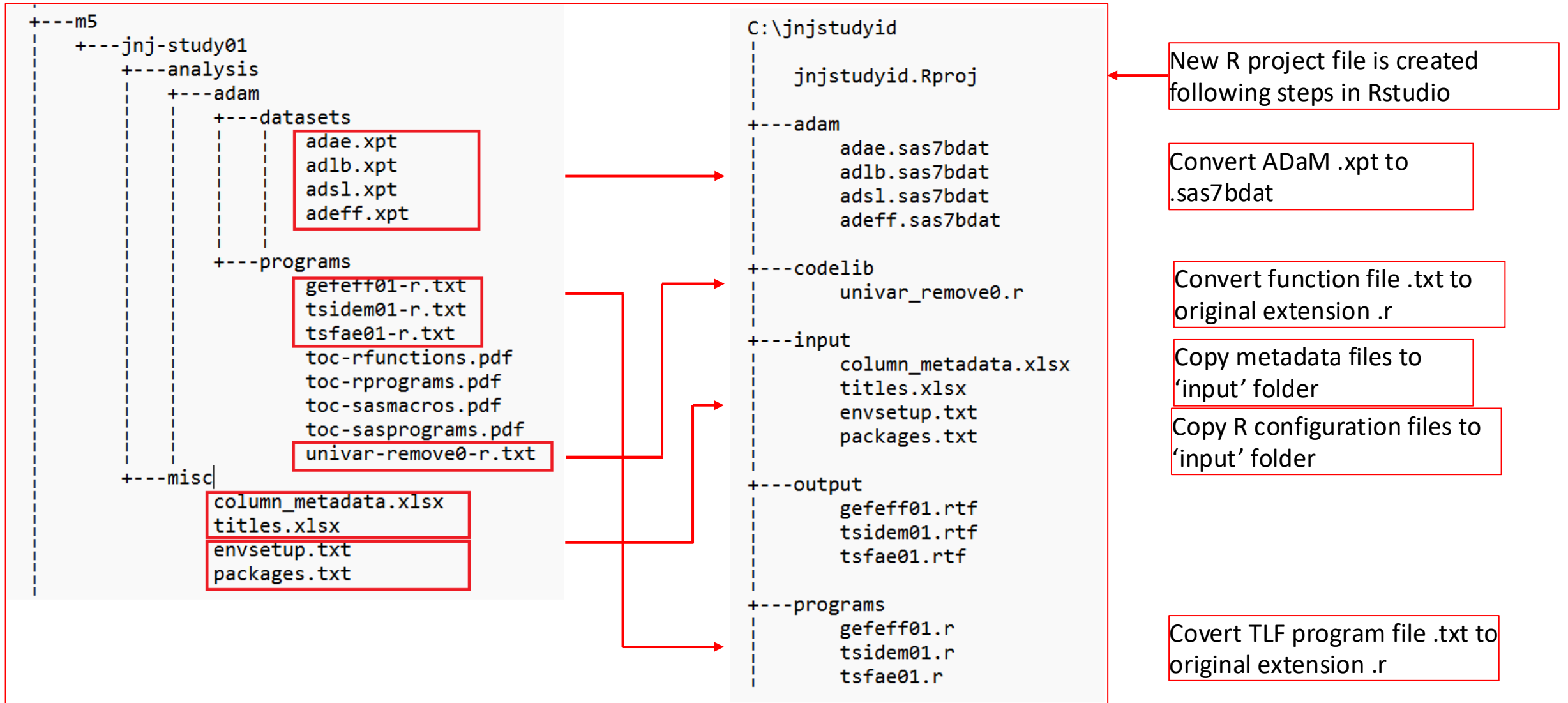
Analysis outputs

R TFL scripts converted from -r.txt to original extension .r

The center location to include the XPT files of datasets copied from different folders under esub package. It is created to streamline the process of converting XPT files to SAS7BDAT format.

To facilitate hybrid SAS/R submissions, R scripts are converted similarly to SAS program files while maintaining the original naming convention. The file extension is appended with a suffix indicating the original programming language (SAS or R) before the .txt extension. For example, adsl.sas becomes adsl-sas.txt, and tsidem01.r becomes tsidem01-r.txt.

ADRG: Set up m5 to working directory



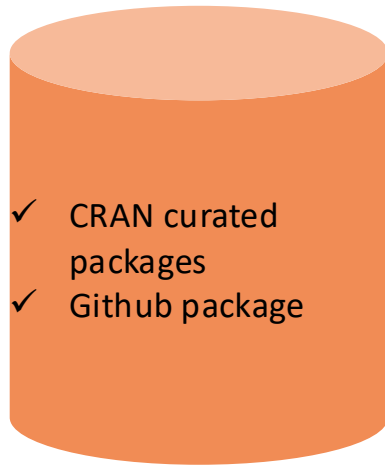
ADRG: Reproduce Environment: Install Packages

Install required packages first:

Statistical Computing
Environment (SCE) in Linux

Configuration file

Window System



```
packages.txt x
1  "", "Package", "Version", "Source"
2  "1", "BiasedUrn", "1.07", "CRAN"
3  "2", "CBPS", "0.23", "CRAN"
4  "3", "DBI", "1.1.3", "CRAN"
5  "4", "DT", "0.25", "CRAN"
```

Install.package()

- {remotes} package: it allows to download and install R packages stored in remote locations such as CRAN and Github
- {renv} package: it supports reproducible environments for R projects, such as an 'renv' folder activated within a new project, where the required R packages are installed specifically for that project in the following step.

```
C:\jnjstudyid|
+---renv
    .gitignore
    activate.R
    settings.json
+---library
    +---R-4.2
        +---x86_64-w64-mingw32
            +---base64enc\
            +---bit64\
            +---assertthat\
            +---BiasedUrn\
            +---CBPS\
            +---DBI\
            +---DT\
```

ADRG: Create a renv lock file

```
renv::snapshot(type='all',  
repos="https://packagemanager.posit.co/cran/yyyy-mm-dd")
```

```
C:\jnjstudyid  
├── jnjstudyid.Rproj  
└── renv.lock  
+---renv
```



Files	Plots	Packages	Help	Viewer	Presentation
Install	Update	renv			
Name	Description	Version	Lockfile	Source	
Project Library					
☐ assertthat	Easy Pre and Post Assertions	0.2.1	0.2.1	Repository	🌐
☐ base64enc	Tools for base64 encoding	0.1-3	0.1-3	Repository	🌐
☐ BiasedUrn	Biased Urn Model Distributions	1.07	1.07	Repository	🌐
☐ bit64	A S3 Class for Vectors of 64bit Integers	4.0.5	4.0.5	Repository	🌐
☐ CBPS	Covariate Balancing Propensity Score	0.23	0.23	Repository	🌐
☐ DBI	R Database Interface	1.1.3	1.1.3	Repository	🌐
☐ DT	A Wrapper of the JavaScript Library 'DataTables'	0.25	0.25	Repository	🌐

- Locks down the snapshot of the project's package ecosystem
- Ensure multiple reviewers can recreate the same analysis (now and later)
- Facilitates sharing the project between collaborators
- Isolates the projects dependencies from global R environment

ADRG: Set paths for analysis {envsetup}

```
.Rprofile x
source("renv/activate.R")
if ("envsetup" %in% data.frame(installed.packages())$Package) {
  library(envsetup)
  # pass configuration to rprofile() to setup the environment
  envsetup::rprofile(config::get(file = file.path(getwd(), "input/envsetup.txt")))
}
invisible(lapply(list.files(file.path(getwd(), "code/lib"), pattern = "*.r", full
.names = TRUE), FUN = function(x) base::source(x)))
```

C:\jnjstudyid
jnjstudyid.Rproj
renv.lock
Rprofile

```
envsetup.txt x
default:
  project_path:
    !expr Sys.setenv(ENVSETUP_PROJECT_PATH = getwd()); getwd()
  paths:
    a_in: !expr list(PROD = file.path(Sys.getenv('ENVSETUP_PROJECT_PATH'), 'adam'))
    functions: !expr list(PROD = file.path(Sys.getenv('ENVSETUP_PROJECT_PATH'), 'code/lib'))
    opath: !expr list(PROD = file.path(Sys.getenv('ENVSETUP_PROJECT_PATH'), 'output'))
    pdev_code_library: !expr file.path(Sys.getenv('ENVSETUP_PROJECT_PATH'), 'code/lib')
    preprod_code_library: !expr file.path(Sys.getenv('ENVSETUP_PROJECT_PATH'), 'code/lib')
    prod_code_library1: !expr file.path(Sys.getenv('ENVSETUP_PROJECT_PATH'), 'code/lib')
    rptdrv: !expr file.path(PROD = Sys.getenv('ENVSETUP_PROJECT_PATH'))
    dspath: !expr list(PROD = file.path(Sys.getenv('ENVSETUP_PROJECT_PATH'), 'input'))
    environ: !expr Sys.setenv(ENVSETUP_ENVIRON = 'PROD'); 'PROD'
```

- The configuration file references working directory to ingest data sources, leverage proprietary helper functions and generate outputs.
- To ensure smooth execution, we request the reviewer to incorporate the path setting in .Rprofile.

rprofile()

Environment	History	Connections	Tutorial
Import Dataset 89 MiB			
R envsetup:paths			
Data			
a_in	List of 1		
\$ PROD: chr "C:/jnjstudyid/adam"			
dspath	List of 1		
\$ PROD: chr "C:/jnjstudyid/input"			
functions	List of 1		
\$ PROD: chr "C:/jnjstudyid/code/lib"			
opath	List of 1		
\$ PROD: chr "C:/jnjstudyid/output"			
Values			
environ	"PROD"		

Challenges/Findings



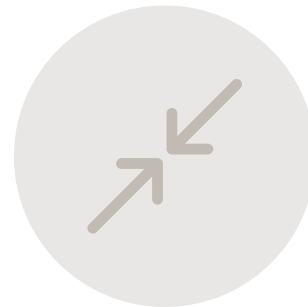
Replicating the sponsor's R environment



Different operating systems require different settings and generate different warnings → in the 'renv' set up, different options might need to be selected to make it work



Different R versions don't perform similarly



Not ideal to switch R versions in Rstudio for every submission → 'renv' package is part of the solution to reproduce the R environment but creates challenges when transferred between different operating systems

Conclusion

- ❑ Prefer **open-source R packages** to minimize additional submission overhead, make the R setup process for reviewers simpler and more transparent.
- ❑ R packages used to generate the analysis results are **validated following GxP**.
- ❑ Enhancing **Reliability of R Analysis Results** and building trust to reviewer:
 - **Detailed explanations of program logic**
 - **R session information saved in R log file**
 - **Results being validated**
- ❑ ADaM with the traceability of one step away to TFL: **minimize of customized functions** developed to handle additional derivations or data manipulations.
- ❑ **ADRG incorporate clear R packages information** including R environment set up guidance by taking Consortium FDA pilot projects and other R submission cases as reference.
- ❑ Using **single file** which contains all the **required R packages** and included in the submission package. It is to simplify the installation process for the reviewer. The **renv.lock** file is required to create after installation serves to stabilize and lock the project environment.
- ❑ Meet the requirements of **eCTD file structure and file formats** following the latest FDA Study Data technical conformance guide and pass validation.

References

R packages validation process in Johnson&Johnson

❑ Using R in a GxP environment: <https://www.youtube.com/watch?v=lWXqfuaxNL8>

Johnson&Johnson's Open Source Journey in R

❑ <https://posit.co/webinar/johnson-johnsons-open-source-journey-in-r/>

R Consortium Pilot Projects

❑ R Consortium Pilot 1, <https://github.com/RConsortium/submissions-pilot1-to-fda>

❑ R Consortium Pilot 2, <https://github.com/RConsortium/submissions-pilot2-to-fda>

❑ R Consortium Pilot 3, <https://github.com/RConsortium/submissions-pilot3-adam-to-fda>

❑ R Consortium Pilot 4, <https://github.com/RConsortium/submissions-pilot4-container>

❑ R Consortium Pilot 5, <https://github.com/RConsortium/submissions-pilot5-datasetjson>

Thank You

For more questions, Please contact:
Veena Terekar
vterekar@its.jnj.com