PharmaSUG 2025 - Paper DV-268

A Customizable Framework in R for Presenting BICR Data in a User-Friendly Format

Reneta Hermiz and Jing Ji, Pfizer Inc.

ABSTRACT

In oncology trials, Blinded Independent Central Review (BICR) data plays a crucial role in providing unbiased and standardized assessments of imaging data. BICR data ensures the consistency and reliability of imaging assessments compared to investigator assessments. This independent review process contributes to the robustness and quality control of trial outcomes.

BICR data, although crucial, can be complex and fragmented across multiple SDTM domains and can pose significant challenges for researchers and clinicians. This paper outlines a process in RStudio for generating a centralized dataset that contains records from multiple SDTM domains. Then, we utilize DT (0.33.3; Xie Y, 2025) and shiny (1.10.0.9000; Chang W, 2025) RStudio packages to produce a web-based application with drillable tables according to three hierarchical levels: lesion level, visit level, and subject level. By linking all tables to a centralized dataset, our approach provides a tool to simplify BICR data review.

The integration and harmonization of BICR data into a cohesive dataset enhances data transparency and accessibility, promoting more efficient data analysis and decision-making processes. This paper introduces a flexible and reproducible workflow for presenting BICR data.

INTRODUCTION

Tumor imaging plays a critical role in oncology trial activities, including diagnosing, staging, monitoring disease and enabling measurable tumor response to treatment. RECIST 1.1 (Eisenhauer et al., 2009) is a standardized response evaluation criteria that is used internationally in oncology and accepted by the FDA for reporting tumor response measurements in oncology trials based on imaging. Investigator assessments that follow RECIST 1.1 guidelines may be susceptible to inadvertent biases stemming from personal relationships with patients, differences in interpretation, and varying levels of expertise. These variations can lead to inconsistencies and potentially affect trial outcomes. The Blinded Independent Central Review (BICR) process was developed to address variability in investigator assessments. BICR data provides a layer of quality control and consistency to measured tumor response through unbiased and standardized assessments of imaging data. This standardized and independent review process helps mitigate the influence of individual biases, enhancing the reliability and validity of imaging assessments.

Imaging assessments in oncology trials are organized into 3 domains outlined by the Clinical Data Interchange Standards Consortium (CDISC) Study Data Tabulation Model Implementation Guide for Human Clinical Trials (SDTMIG) (Clinical Data Interchange Standards Consortium, 2013).

1. Tumor Identification (TU) identifies unique tumors and sites of disease. One record exists per identified tumor per subject.
2. Tumor Response (TR) includes quantitative and/or qualitative assessments of tumors, such as tumor measurements. Each record represents a unique tumor and assessment per visit per subject.
3. Disease Response (RS) lists disease response evaluations. These records are listed per response assessment per visit per subject.

These domains contain both investigator and BICR assessments. The data across these 3 domains are linked through certain variables. Reviewing BICR data can be complex and often fragmented due to multiple domains and non-duplicated data. For example, tumor location is only contained in one domain (TU). Additionally, datasets contain assessments from multiple evaluators who have different methods of numbering lesions. This fragmentation and lack of uniformity poses challenges for researchers and clinicians who need to access and analyze this data efficiently and with minimal risk to data integrity.

To address these issues, this paper introduces a customizable method using R programming to integrate and harmonize BICR data from CDISC standard oncology domains into a single cohesive dataset. By leveraging the capabilities of the DT and shiny RStudio packages, we propose a user-friendly interface for presenting BICR data at 2 hierarchical levels: lesion-level and visit-level. This approach not only simplifies the process of BICR data review but also enhances data transparency and accessibility, promoting more efficient data analysis and decision-making processes.

First, using publicly available test data from the admiral.test package, we describe data processing required to combine the raw SDTM data. Then we illustrate how to create the interactive interfaces. Each hierarchical level of data will be presented in a shiny application format with 2 panels. The top panel will contain a data listing with limited information. When records in the top selection panel are selected, the relevant supporting data will populate in the bottom output panel. Single or multiple records can be selected to show additional raw data records that can be compared. Both panels of data can be filtered by non-grouping variables.

The integration of TR, TU, and RS domains into a single source dataset eliminates the need for researchers to search through multiple repositories for related information, thereby reducing the risk of errors and saving valuable time. The use of interactive tables rendered by “DT” and packaged in a shiny application enables users to explore data at distinct levels of granularity with ease.

DESCRIPTION OF SOURCE DATA

The source data TU, TR, and RS used in this paper is available in the admiral.test package in R and can be accessed by using the code below.

```
   library(admiral.test)

#call TU, TR, and RS domains and set to dataframes

   tu <- admiral_tu
   tr <- admiral_tr
   rs <- admiral_rs
```

Program 1. Retreival of source data from admiral.test package

The first 3 rows of each source dataset are shown in Display 1. Linking variables are highlighted and other variables of interest have been selected for illustration of the TU, TR, and RS datasets, respectively.

STUDYID	USUBJID	TULNKID	TUTEST	TUORRES	TULOC	TUMETHOD	TUEVAL	VISIT
CDISCPILOT01	01-701-1015	NT01	Tumor Identification	NON-TARGET	ANUS	CT SCAN	INVESTIGATOR	BASELINE
CDISCPILOT01	01-701-1015	NT02	Tumor Identification	NON-TARGET	BILE DUCT	CT SCAN	INVESTIGATOR	BASELINE
CDISCPILOT01	01-701-1015	NT03	Tumor Identification	NON-TARGET	BRAIN	CT SCAN	INVESTIGATOR	BASELINE

3 rows

STUDYID	USUBJID	TRLNKGRP	TRLNKID	TRTEST	TRORRES	TRMETHOD	TREVAL	TRACPTFL	VISIT
CDISCPILOT01	01-701-1015	A1	NT01	Tumor State	PRESENT	CT SCAN	INVESTIGATOR	NA	BASELINE
CDISCPILOT01	01-701-1015	A1	NT02	Tumor State	PRESENT	CT SCAN	INVESTIGATOR	NA	BASELINE
CDISCPILOT01	01-701-1015	A1	NT03	Tumor State	PRESENT	CT SCAN	INVESTIGATOR	NA	BASELINE

3 rows

STUDYID	USUBJID	RSLNKGRP	RSTEST	RSORRES	RSEVAL	RSACPTFL	VISIT
CDISCPILOT01	01-701-1015	R1-A2	Overall Response	PR	INDEPENDENT ASSESSOR	Y	WEEK 6
CDISCPILOT01	01-701-1015	NA	Non-target Response	NE	INDEPENDENT ASSESSOR	Y	WEEK 6
CDISCPILOT01	01-701-1015	NA	Target Response	PR	INDEPENDENT ASSESSOR	Y	WEEK 6

3 rows

Display 1. Sample of TU Source data from admiral.test Package

LINK VARIABLES

The records in these domains are associated according to link variables. The link variables are highlighted in the examples above.

The Tumor Link ID variable, TxLNKID (TULNKID and TRLNKID), identifies individual tumors. TxLNKID is used to link identified tumors in the TU domain with the relevant assessment results in TR for each tumor.

The Tumor Link Group variable, xxLNKGRP (TRLNKGRP and RSLNKGRP), identifies groups of tumors and is used to link disease assessments from the RS domain with tumor assessments recorded in the TR domain. Each xxLNKGRP is associated with a response assessment for a group of lesions (i.e. target or non-target) per subject visit. If a record in TR shares a xxLNKGRP with a record in RS, then the tumor, identified by TRLNKID in TR, was included in the corresponding disease assessment in RS.

BICR VARIABLES

The link variables are related to the Evaluator (TUEVAL) and Evaluator Identifier (TUEVALID) variables. TUEVAL differentiates assessments performed by the study investigator or an independent assessor. Records evaluated by an independent assessor are BICR records. TUEVALID further identifies which BICR assessor performed the record assessment (i.e. Radiologist 1 vs. Radiologist 2).

When data exists from more than 1 independent assessor, there is an adjudication process to determine which assessor's records are acceptable for analysis. Accepted records are flagged with the Accepted Record Flag (xxACPTFL) variable.

ASSESSMENT VARIABLES

Each domain contains variables describing the assessment and results for each record. Table 1 defines some of the tumor assessment variables that will be used in this paper. Please refer to the current SDTMIG for detailed information on the assessment variables expected for each oncology domain.

xxACPTFL	Record Acceptable Flag
xxEVAL	Evaluator
xxEVALID	Evaluator ID
xxMETHOD	Imaging Method
xxORRES	Assessment Result in Original Units
xxTEST	Tumor Assessment Test Name
xxTESTCD	Tumor Assessment Test Code

Table 1. Description of Common Tumor Assessment Variables

PRE-PROCESSING OF ONCOLOGY DOMAINS

The oncology domain datasets need to undergo data cleaning and restructuring in preparation for a cross-domain merge. The restructuring allows for more flexibility and efficiency in downstream visualization steps. After the datasets have been cleaned, they will be assigned to a xx_clean dataset before being merged with the other oncology domains. At this stage, all of the data to be used in each hierarchical level of visualization will be included in these xx_clean datasets. Because these are source datasets, much of the data cleaning involves filling in missing data with dummy or inferred data.

TU

The main variable of interest in the TU domain is tumor location (TULOC) which is not included in the other oncology datasets. To prepare the tu_clean dataset, we will select the link variable and tumor location from TU.

```
tu_clean <- tu %>%
  select(USUBJID,TULNKID,TULOC)
```

Program 2. TU Pre-Processing

TR

The TR dataset used in this paper has missing values for TREVALID and TRACPTFL variables. The missing values will be imputed based on the existing data for cohesive visualization.

Then, the data will be restructured by extracting visit-level records and any variables required for calculating metrics. Here, we extract the SUMDIAM (sum of diameters) variable into the tr_calc dataset. This dataset is then used to calculate percent change from baseline, a standard metric for oncological tumor assessment. tr_calc is then transposed and merged with the original TR dataset. The result is TR with sum of diameters and percent change from baseline in column formats. This will allow for more efficient visualization.

```
#DATA CLEANING
#include "Investigator" in evaluator ID
tr$TREVALID <- ifelse(tr$TREVAL=="INVESTIGATOR","INVESTIGATOR",tr$TREVALID)

#define non-accepted BICR reads
tr$TRACPTFL <- ifelse(tr$TREVAL!="INVESTIGATOR",
                      ifelse(is.na(tr$TRACPTFL),"N",tr$TRACPTFL),tr$TRACPTFL)

#DATA RESTRUCTURING
#extract visit level results
tr_calc <- tr %>%
  select(USUBJID, TRLNKGRP, TRGRPID, TREVALID, TRTESTCD, TRORRES, TRSTRESN, VISITNUM) %>%
  filter(TRTESTCD=="SUMDIAM") %>%
#transpose data
  pivot_wider(names_from = TRTESTCD, values_from = TRORRES) %>%

#METRICS CALCULATION
#Group by subjid and investigator. Sort by visit number.
  group_by(USUBJID, TREVALID) %>%
  arrange(VISITNUM) %>%
#calculate PCBSD (Percent Change from Baseline of Sum of Diameters)
  mutate(PCBSD = round(100*(TRSTRESN-first(TRSTRESN))/first(TRSTRESN),digits=2)) %>%
  ungroup()

#MERGE RESTRUCTURED/CALCULATED DATA
tr_calc <- tr_calc %>%
  select(USUBJID, TRLNKGRP, TRGRPID, SUMDIAM, PCBSD)

tr_clean <- merge(tr_calc,tr,
                  by.x = c("USUBJID", "TRLNKGRP", "TRGRPID"),
                  by.y = c("USUBJID", "TRLNKGRP", "TRGRPID"),
                  all = TRUE) %>%
#exclude data that has been extracted and restructured
  filter(TRTESTCD != "SUMDIAM")
```

Program 3. TR Pre-Processing

RS

The RS dataset has missing values for RSEVALID and RSLNKGRP. RSEVALID are missing for investigator assessments. Values will be imputed using the same process that is used for TREVALID above. For RSLNKGRP, the goal is to ensure that RS records are correctly assigned a RSLNKGRP based on patient id (USUBJID), assessment (RSTESTCD), evaluator (RSEVAL), and visit. If the assessment is for a new lesion progression, a link group will not be assigned because the assessment is related to an individual lesion. Target and Non-Target assessment records (RSTESTCD=TRGRESP or NTRGRESP) will be maintained in tabular records, while Overall and New Lesion Progression records (RSTESTCD=OVRLRESP or NEWLPROG) will be restructured into columns.

```
#DATA CLEANING
#include "Investigator" in evaluator ID
rs$RSEVALID <- ifelse(rs$RSEVAL=="INVESTIGATOR","INVESTIGATOR",rs$RSEVALID)
```

```
#impute link group for investigator records
rs$RSLNKGRP <- ifelse(rs$RSTESTCD == "NEWLPROG", "NA",
                  ifelse(rs$RSEVAL=="INVESTIGATOR",
#data cleaning: add xxlnkgrp for inv results
                    ifelse(rs$VISIT=="BASELINE","A1",
                      ifelse(rs$VISIT=="WEEK 6","A2",
                        ifelse(rs$VISIT=="WEEK12","A3",
                          ifelse(rs$VISIT=="WEEK 18","AUNPL",
                            ifelse(rs$VISIT=="WEEK 24","A4",rs$RSLNKGRP))))),rs$RSLNKGRP))
#fill missing RSLNKGRP values
rs <- rs %>% fill(RSLNKGRP)

#DATA RESTRUCTURING
#extract visit level results
rs_calc <- rs %>%
  select(USUBJID,RSLNKGRP,RSTESTCD,RSORRES,RSSTAT_,VISIT) %>%
  filter(RSTESTCD=="OVRLRESP" | RSTESTCD=="NEWLPROG") %>%
#transpose data
  pivot_wider(names_from = RSTESTCD, values_from = c(RSORRES,RSSTAT))

#MERGE RESTRUCTURED DATA
rs_clean <- merge(rs,rs_calc,
                  by.x = c("USUBJID","RSLNKGRP","VISIT"),
                  by.Y = c("USUBJID","RSLNKGRP","VISIT"),
                  all = TRUE) %>%
#exclude data that has been extracted and restructured
  filter(RSTESTCD !="OVRLRESP" & RSTESTCD != "NEWLPROG")
```

Program 4. RS Pre-Processing

GENERATE CENTRALIZED DATASET

COMBINING TU, TR, AND RS DOMAINS

Each dataset has been cleaned and processed in preparation for merging into a centralized dataset that combines all 3 domains. This combined dataset will include all of the data that will be used to create the selection and output panel visualizations in our tool.

The merge occurs in 2 steps to account for the 2 types of link variables, xxLNKID and xxLNKGRP. Selection keys are generated for the different hierarchical levels of visualization. The selection key will act as a link between the top and bottom visualization panels of each hierarchical level. When a selection panel record is selected, the key will trigger linked data to appear in the output panel.

```
#MERGE
#TU and TR merge
tu_tr <- merge(tu_clean,tr_clean,
               by.x = c("USUBJID","TULNKID"),
               by.y = c("USUBJID","TRLNKID"),
               all = TRUE)
#RS merge
combined <-merge(tu_tr, rs_clean,
               by.x = c("USUBJID","TRLNKGRP","VISIT"),
               by.y = c("USUBJID","RSLNKGRP","VISIT"),
               all = TRUE)

#CREATE SELECTION KEYS
combined$lesionkey <-paste(combined$USUBJID,combined$TULNKID,sep = "_")
combined$visitkey <- paste(combined$USUBJID,combined$VISIT,combined$TREVALID, sep = "_")
```

Program 5. Merge Oncology Domains into Centralized Dataset

DESIGN PANEL SUMMARIES

The centralized dataset has a lot of information which still needs to be organized properly to be useful. The main purpose of this tool is to create an efficient, user-friendly interface. Data selection and table design should be carefully considered to achieve this goal.

LESION LEVEL

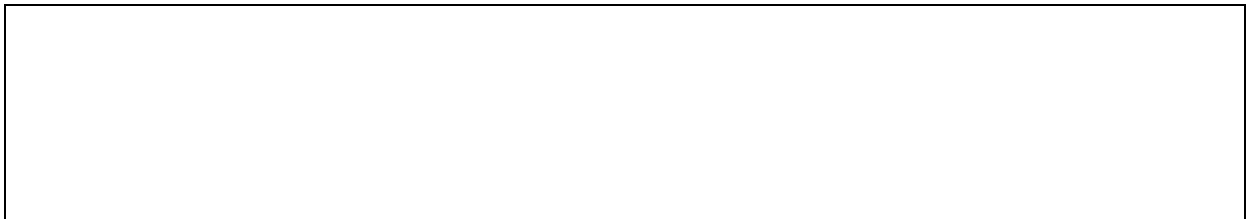
The lesion-level visualization design focuses on a baseline tumor listing in the top panel which drills down into tumor-specific assessment data spanning all visits. The baseline tumor listing in the top panel displays a simple overview of baseline tumors including tumor type (target/non-target), lesion number, anatomical location, and evaluator. This provides a reference point for subsequent assessments and allows the reviewer to easily identify patient lesions of interest. The drill-down table then allows users to view each tumor's assessment over time. The bottom panel includes metrics such as tumor measurements and calculations such as percent change from baseline.

Each panel of the visualization will display a different dataset. The centralized dataset is used to subset the relevant data for both lesion-level panels. The subset is then used to generate the desired output datasets.

```
#SUBSET FROM CENTRALIZED DATASET
lesion_level <- combined %>%
  select(USUBJID,  #PARTICIPANT IDENTIFIER
         TRGRPID,  #LESION TYPE (TARGET/NON-TARGET)
         TULNKID,  #LESION NUMBER
         TREVALID, #EVALUATOR (INVISTEGATOR, RADIOLOGIST1, RADIOLOGIST2)
         TRACPTFL, #ACCEPTABLE RECORD FLAG
         TRMETHOD, #IMAGING METHOD
         TULOC,    #TUMOR LOCATION
         TRDTC,    #DATE
         VISIT,    #VISIT
         TRTESTCD, #TEST CODE
         TRORRES,  #TEST RESULT
         SUMDIAM,  #SUM OF DIAMETERS
         PCBSD,    #PERCENT CHANGE FROM BASELINE OF SUMDIAM
         lesionkey)#PANELS LINK VARIABLE

#SELECTION PANEL - BASELINE LESION LISTING
lesion_level_select <- lesion_level %>%
  select(USUBJID,  #PARTICIPANT IDENTIFIER
         TRGRPID,  #LESION TYPE (TARGET/NON-TARGET)
         TULNKID,  #LESION NUMBER
         TREVALID, #EVALUATOR (INVESTIGATOR, RADIOLOGIST1, RADIOLOGIST2)
         TRACPTFL, #ACCEPTABLE RECORD FLAG
         TULOC,    #TUMOR LOCATION
         key)      #PANELS LINK VARIABLE

#OUTPUT PANEL - LESION-SPECIFIC SUMMARY
lesion_level_transpose <- lesion_level %>%
   filter(!is.na(TRTESTCD)) %>%
   pivot_wider(names_from = TRTESTCD, values_from = TRORRES)
lesion_level_output <- lesion_level_transpose[,c(1:9,13,14,10,11,12)]
```

Program 6. Generating Lesion-Level Panel Datasets

Output 1 shows the results of lesion_level_select and lesion_level_output datasets.



LESION_LEVEL_SELECT (SELECTION PANEL)

Description: df [38,168 × 7]

USUBJID <chr>	TRGRPID <chr>	TULNKID <chr>	TREVALID <chr>	TRACPTFL <chr>	TULOC <chr>	key <chr>
01-701-1015	NON-TARGET	NT01	INVESTIGATOR	NA	ANUS	01-701-1015_NT01
01-701-1015	NON-TARGET	NT02	INVESTIGATOR	NA	BILE DUCT	01-701-1015_NT02
01-701-1015	NON-TARGET	NT03	INVESTIGATOR	NA	BRAIN	01-701-1015_NT03
01-701-1015	NON-TARGET	NT04	INVESTIGATOR	NA	GALL BLADDER	01-701-1015_NT04
01-701-1015	NON-TARGET	NT05	INVESTIGATOR	NA	HEAD	01-701-1015_NT05
01-701-1015	TARGET	T01	INVESTIGATOR	NA	ADRENAL GLAND	01-701-1015_T01
01-701-1015	TARGET	T02	INVESTIGATOR	NA	BLADDER	01-701-1015_T02
01-701-1015	TARGET	T03	INVESTIGATOR	NA	BODY	01-701-1015_T03
01-701-1015	TARGET	T04	INVESTIGATOR	NA	BONE	01-701-1015_T04
01-701-1015	TARGET	T05	INVESTIGATOR	NA	BREAST	01-701-1015_T05

1-10 of 38,168 rows Previous 1 2 3 4 5

LESION_LEVEL_OUTPUT (OUTPUT PANEL)

A tibble: 26,364 x 14

USUBJID <chr>	TRGRPID <chr>	TULNKID <chr>	TREVALID <chr>	TRACPTFL <chr>	TRMETHOD <chr>	TULOC <chr>
01-701-1015	TARGET	R1-T01	RADIOLOGIST 1	Y	CT SCAN	BLADDER
01-701-1015	TARGET	R1-T01	RADIOLOGIST 1	Y	CT SCAN	BLADDER
01-701-1015	TARGET	R1-T01	RADIOLOGIST 1	Y	CT SCAN	BLADDER
01-701-1015	TARGET	R1-T01	RADIOLOGIST 1	Y	CT SCAN	BLADDER
01-701-1015	TARGET	R1-T02	RADIOLOGIST 1	Y	CT SCAN	BODY
01-701-1015	TARGET	R1-T02	RADIOLOGIST 1	Y	CT SCAN	BODY
01-701-1015	TARGET	R1-T02	RADIOLOGIST 1	Y	CT SCAN	BODY
01-701-1015	TARGET	R1-T02	RADIOLOGIST 1	Y	CT SCAN	BODY
01-701-1015	TARGET	R1-T03	RADIOLOGIST 1	Y	CT SCAN	BONE
01-701-1015	TARGET	R1-T03	RADIOLOGIST 1	Y	CT SCAN	BONE

41-50 of 26,364 rows | 1-7 of 14 columns Previous 1 … 3 4 5 6

TRDTC <chr>	VISIT <chr>	TUMSTATE <list>	DIAMETER <list>	SUMDIAM <chr>	PCBSD <dbl>	key <chr>
2014-01-02	BASELINE	<NULL>	<chr [1]>	72	0.00	01-701-1015_R1-T01
2014-02-12	WEEK 6	<NULL>	<chr [1]>	49	-31.94	01-701-1015_R1-T01
2014-03-26	WEEK 12	<NULL>	<chr [1]>	0	-100.00	01-701-1015_R1-T01
2014-06-18	WEEK 24	<NULL>	<chr [1]>	35	-51.39	01-701-1015_R1-T01
2014-01-02	BASELINE	<NULL>	<chr [1]>	72	0.00	01-701-1015_R1-T02
2014-02-12	WEEK 6	<NULL>	<chr [1]>	49	-31.94	01-701-1015_R1-T02
2014-03-26	WEEK 12	<NULL>	<chr [1]>	0	-100.00	01-701-1015_R1-T02
2014-06-18	WEEK 24	<NULL>	<chr [1]>	35	-51.39	01-701-1015_R1-T02
2014-01-02	BASELINE	<NULL>	<chr [1]>	72	0.00	01-701-1015_R1-T03
2014-02-12	WEEK 6	<NULL>	<chr [1]>	49	-31.94	01-701-1015_R1-T03

41-50 of 26,364 rows | 8-14 of 14 columns Previous 1 … 3 4 5 6 7 … 1

Output 1. Lesion-Level Datasets for Selection and Output Panels

VISIT LEVEL

For the visit-level visualization, the design includes a post-baseline response listing in the top selection panel which drills down into a visit summary in the bottom output panel. The post-baseline response listing includes evaluator, evaluation type and response assessment (CR/PR/SD/PD etc.). This listing provides a concise overview of the patient status at a particular visit. This listing contains limited data and we have opted to show overall and new lesion progression data in tabular records instead of the restructured format included in the centralized dataset. For these reasons, we use the RS dataset to subset the selection panel. The output visit summary then provides further details such as imaging method, tumor location, and metrics for the link group such as sum of diameters and percent change from baseline.

```
#SUBSET FROM RS FOR SELECTION PANEL
visit_level_select <- rs %>%
  select(USUBJID,   #PARTICIPANT IDENTIFIER
         VISIT,     #VISIT
         RSDTC,     #DATE
         RSTEST,    #EVALUATION TYPE (TARGET/NON-TARGET)
         RSLNKGRP,  #LINK GROUP
         RSEVALID,  #EVALUATOR (INVESTIGATOR, RADIOLOGIST1, RADIOLOGIST2)
         RSORRES,   #DISEASE RESPONSE
         RSSTAT)    #COMPLETION STATUS
#generate visit selection key
visit_level_select$visitkey <- paste(visit_level_select$USUBJID,
                                     visit_level_select$VISIT,visit_level_select$RSEVALID,
                                      sep="_")

#SUBSET FROM CENTRALIZED DATASET FOR OUTPUT PANEL
visit_level_output <- combined %>%
  select(USUBJID,   #PARTICIPANT IDENTIFIER
         VISIT,     #VISIT
         TRDTC,     #DATE
         TRGRPID,   #LESION TYPE
         TRLNKGRP,  #LINK GROUP
         TULNKID,   #LESION NUMBER
         TREVALID,  #EVALUATOR (INVISTEGATOR, RADIOLOGIST1, RADIOLOGIST2)
         TRACPTFL,  #ACCEPTABLE RECORD FLAG
         RSORRES,   #DISEASE RESPONSE
         RSSTAT,    #ASSESSMENT COMPLETION STATUS
         RSORRES_NEWLPROG, #NEW LESION RESPONSE
         RSORRES_OVRLRESP, #OVERALL RESPONSE
         RSSTAT__OVRLRESP, #ASSESSMENT COMPLETION STATUS
         TRMETHOD,  #IMAGING METHOD
         TULOC,     #TUMOR LOCATION
         SUMDIAM,   #SUM OF DIAMETERS
         PCBSD,     #PERCENT CHANGE FROM BASELINE OF SUMDIAM
         visitkey)  #PANELS LINK VARIABLE
```

Program 7. Generating Visit-Level Panel Datasets

Output 2 shows the results of visit_level_select and visit_level_output datasets.

VISIT_LEVEL_SELECT (SELECTION PANEL)

A tibble: 5,703 x 9

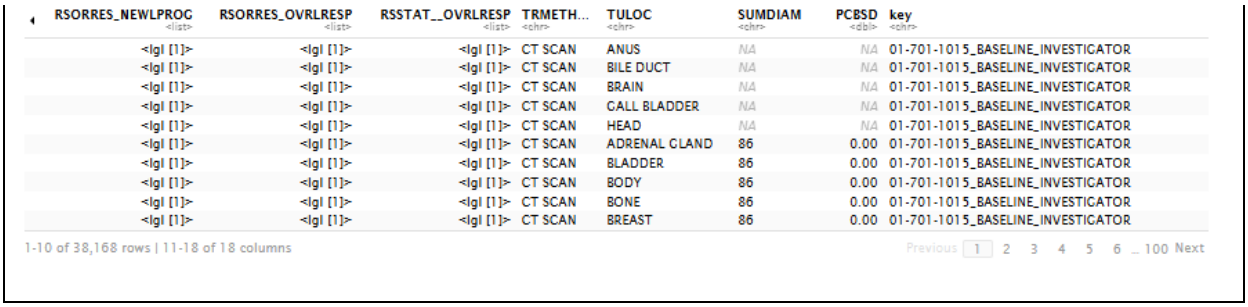
USUBJID <chr>	VISIT <chr>	RSDTC <chr>	RSTEST <chr>	RSLNKGRP <chr>	RSEVALID <chr>	RSORRES <chr>	RSSTAT <chr>	key <chr>
01-701-1015	WEEK 6	2014-02-12	Overall Response	R1-A2	RADIOLOGIST 1	PR	NA	01-701-1015_WEEK 6_RADIOLOGIST 1
01-701-1015	WEEK 6	2014-02-12	Non-target Response	R1-A2	RADIOLOGIST 1	NE	NOT DONE	01-701-1015_WEEK 6_RADIOLOGIST 1
01-701-1015	WEEK 6	2014-02-12	Target Response	R1-A2	RADIOLOGIST 1	PR	NA	01-701-1015_WEEK 6_RADIOLOGIST 1
01-701-1015	WEEK 6	2014-02-12	Overall Response	R2-A2	RADIOLOGIST 2	PD	NA	01-701-1015_WEEK 6_RADIOLOGIST 2
01-701-1015	WEEK 6	2014-02-12	Non-target Response	R2-A2	RADIOLOGIST 2	PD	NA	01-701-1015_WEEK 6_RADIOLOGIST 2
01-701-1015	WEEK 6	2014-02-12	Target Response	R2-A2	RADIOLOGIST 2	SD	NA	01-701-1015_WEEK 6_RADIOLOGIST 2
01-701-1015	WEEK 6	2014-02-12	Overall Response	A2	INVESTIGATOR	PR	NA	01-701-1015_WEEK 6_INVESTIGATOR
01-701-1015	WEEK 6	2014-02-12	Non-target Response	A2	INVESTIGATOR	NON-CR/NON-PD	NA	01-701-1015_WEEK 6_INVESTIGATOR
01-701-1015	WEEK 6	2014-02-12	Target Response	A2	INVESTIGATOR	PR	NA	01-701-1015_WEEK 6_INVESTIGATOR
01-701-1015	WEEK 12	2014-03-26	Overall Response	R1-A3	RADIOLOGIST 1	CR	NA	01-701-1015_WEEK 12_RADIOLOGIST 1

1-10 of 5,703 rows Previous 1 2 3 4 5 6

VISIT_LEVEL_OUTPUT (OUTPUT PANEL)

USUBJID <chr>	VISIT <chr>	TRDTC <chr>	TRGRPID <chr>	TRLNKGRP <chr>	TULNKID <chr>	TREVALID <chr>	TRACPTFL <chr>	RSORRES <chr>	RSSTAT <chr>
01-701-1015	BASELINE	2014-01-02	NON-TARGET	A1	NT01	INVESTIGATOR	NA	NA	NA
01-701-1015	BASELINE	2014-01-02	NON-TARGET	A1	NT02	INVESTIGATOR	NA	NA	NA
01-701-1015	BASELINE	2014-01-02	NON-TARGET	A1	NT03	INVESTIGATOR	NA	NA	NA
01-701-1015	BASELINE	2014-01-02	NON-TARGET	A1	NT04	INVESTIGATOR	NA	NA	NA
01-701-1015	BASELINE	2014-01-02	NON-TARGET	A1	NT05	INVESTIGATOR	NA	NA	NA
01-701-1015	BASELINE	2014-01-02	TARGET	A1	T01	INVESTIGATOR	NA	NA	NA
01-701-1015	BASELINE	2014-01-02	TARGET	A1	T02	INVESTIGATOR	NA	NA	NA
01-701-1015	BASELINE	2014-01-02	TARGET	A1	T03	INVESTIGATOR	NA	NA	NA
01-701-1015	BASELINE	2014-01-02	TARGET	A1	T04	INVESTIGATOR	NA	NA	NA
01-701-1015	BASELINE	2014-01-02	TARGET	A1	T05	INVESTIGATOR	NA	NA	NA

1-10 of 38,168 rows | 1-10 of 18 columns Previous 1 2 3 4 5 6 ... 100 Ne



RSORRES_NEWLPROG <list>	RSORRES_OVRLRESP <list>	RSSTAT__OVRLRESP <list>	TRMETH... <chr>	TULOC <chr>	SUMDIAM <chr>	PCBSD <dbl>	key <chr>
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	ANUS	NA	NA	01-701-1015_BASELINE_INVESTICATOR
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	BILE DUCT	NA	NA	01-701-1015_BASELINE_INVESTICATOR
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	BRAIN	NA	NA	01-701-1015_BASELINE_INVESTICATOR
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	GALL BLADDER	NA	NA	01-701-1015_BASELINE_INVESTICATOR
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	HEAD	NA	NA	01-701-1015_BASELINE_INVESTICATOR
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	ADRENAL GLAND	86	0.00	01-701-1015_BASELINE_INVESTICATOR
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	BLADDER	86	0.00	01-701-1015_BASELINE_INVESTICATOR
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	BODY	86	0.00	01-701-1015_BASELINE_INVESTICATOR
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	BONE	86	0.00	01-701-1015_BASELINE_INVESTICATOR
<lgl [1]>	<lgl [1]>	<lgl [1]>	CT SCAN	BREAST	86	0.00	01-701-1015_BASELINE_INVESTICATOR

1-10 of 38,168 rows | 11-18 of 18 columns

Previous 1 2 3 4 5 6 … 100 Next

Output 2. Visit-Level Datasets for Selection and Output Panels

GENERATE INTERACTIVE INTERFACE

DT PACKAGE: CREATE A TABLE WIDGET

The DT::renderDT function creates an interactive table widget that can be used within a Shiny application. There are many options and customizations that can be used with renderDT to create an organized and easy to read table. The example below shows how several options are used to customize the lesion_level_select dataset for visualization in the lesion-level selection panel. Please note that this code does not produce any output outside of a shiny application.

```
renderDT(lesion_level_select,
         extensions = 'FixedHeader',          #extensions add functionality
         options = list(pageLength = 10,      #sets default number of records shown
                        fixedHeader = TRUE,   #'FixedHeader' extension option
                        columnDefs = list(    #initiate column options*
                          list(
                            targets=2:6,      #specify target columns
                            width='25px'      #option to apply to specified columns
                           ),
                           list(
                             targets=c(0,7),
                             visible=FALSE
                           )
                          )),
           Filter = "top",                    #include column filters
           colnames = c("Participant ID"=2,   #label columns**
                        "Lesion Type"=3,
                        "Lesion Number"=4,
                        "Evaluator"=5,
                        "Acceptable Record"=6,
                        "Tumor Location"=7,
                        "Selection Key"=8
                        ))
*"columnDefs" options count columns starting with 0. Column 0 contains row labels.
**"colnames" options count columns starting with 1. Column 1 contains row labels.
```

Program 8. DT::renderDT Options

SHINY PACKAGE: DEVELOP USER INTERFACE

The shiny package in R allows users to build web applications which are called "shiny apps". The first step to creating a shiny app is to define a user interface (ui) object. The next step is to implement the server function which defines the logic of the visualization. Once the ui and server are defined, they are called into the shinyApp function which executes the logic and deploys the shiny app.

The shiny::ui function sets the main layout of the shiny app. Setting ui to "fluidPage" calls a simple ui object which we use in our examples. It is responsive which means that the elements within the app will adjust based on the size of the user's browser window. Titles can be set at various locations depending on the order they are placed in the ui function. DTOutput() creates a placeholder for a table widget which sets a link (i.e. level_select) between the ui element and the output to be displayed.

Once the ui is defined, the shiny::server function is initiated to define the logic which processes inputs and generates outputs for the app. within the server function,

- output$level_select references the DTOutput placeholder from the ui definition. This defines the selection panel output by setting it to the widget generated by DT::renderDT. The renderDT setup from Program 8 can be placed here to define it as the first output. It is excluded in the below for brevity.
- shiny::reactive() assigns a reactive expression to select_key. The reactive expression checks whether a selection has been made by a user and re-runs when the input selection row value changes (anytime a user selects or de-selects a row).
- shiny::validate function is used to define the logic of the application. It includes a condition and a message when condition is not met. In this case, the condition checks if the user has selected 1 or more rows. If no rows are selected, a message will appear to prompt the user to select a row from the selection panel.
- input$selection_row takes the row number of the selection and extracts the corresponding value of the key variable and triggers the retrieval of linked data from the output panel dataset.
- output$level_output references the DTOutput placeholder from the ui definition. The DT::renderDT function for level_output calls the reactive expression, select_key, and sets up the output panel table widget.
- shinyApp() deploys the application

```
#DEFINE ui
ui <- fluidPage(                                    #ui Definition
          titlePanel("Baseline Lesion Listing"),    #title for top panel display
          DTOutput("level_select"),                 #define output rendered by DT
          titlePanel("Lesion Summary"),             #title for bottom panel display
          DTOutput("level_output")                  #define output rendered by DT
       )

#SERVER FUNCTION
server <- function(input,output{                    #initiate server function

     output$level_select <- DT::renderDT(…)         #define first output (selection panel)

     select_key <- shiny::reactive({                #define selection key for drill-down
                shiny::validate(
                need(length(input$selection_row)>0, "[prompt text]")
                )
                selection <- lesion_level_select[as.integer(input$selection_row), ]$key
                lesion_level_output[lesion_level_output$key %in% selection, ]
              })

      output$level_output <- DT::renderDT(select_key(),)#define second output (output
panel)

#RUN SHINY APPLICATION
shinyApp(ui,server)                                 #initiate shiny application
```

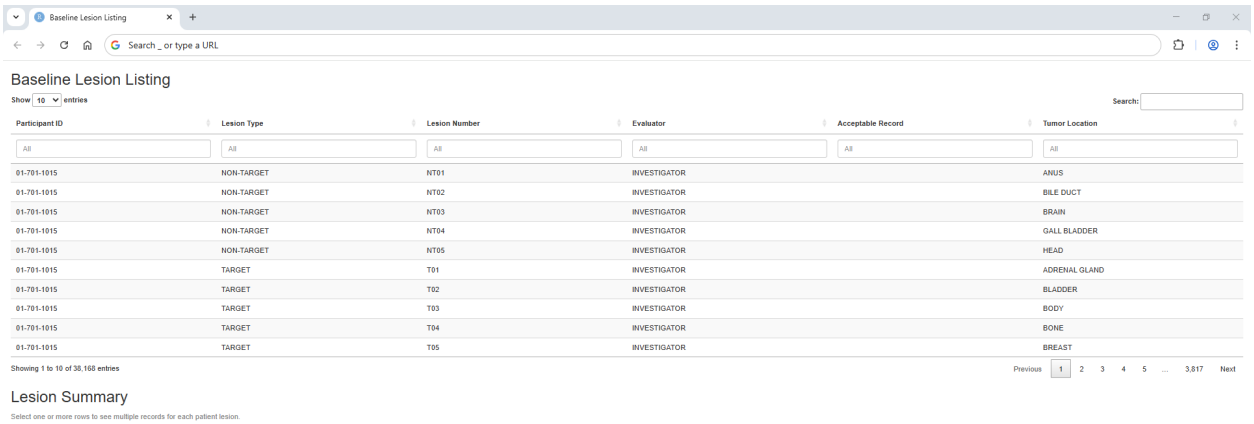
Program 9. Shiny Application Example

```
#DEFINE ui
ui <- fluidPage(                                    #ui Definition
          titlePanel("Baseline Lesion Listing"),    #title for top panel display
          DTOutput("level_select"),                 #define output rendered by DT
          titlePanel("Lesion Summary"),             #title for bottom panel display
          DTOutput("level_output")                  #define output rendered by DT
       )

#SERVER FUNCTION
server <- function(input,output{                    #initiate server function
```

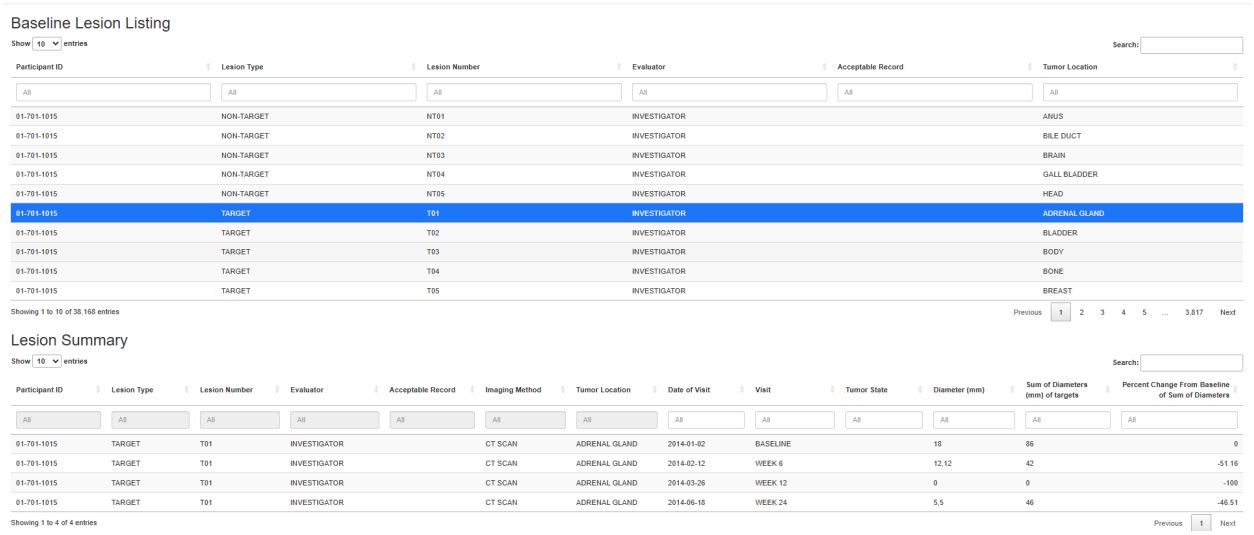
```
    output$level_select <- DT::renderDT(         #define first output (selection panel)
        lesion_level_select,
        extensions = 'FixedHeader',          #extensions add functionality
        options = list(pageLength = 10,      #sets default number of records shown
                       fixedHeader = TRUE,  #'FixedHeader' extension option
                       columnDefs = list(   #initiate column options*
                         list(
                           targets=2:6,      #specify target columns
                           width='25px'      #option to apply to specified columns
                          ),
                          list(
                            targets=c(0,7),
                            visible=FALSE
                          )
                        )),
          Filter = "top",                    #include column filters
          colnames = c("Participant ID"=2,  #label columns**
                       "Lesion Type"=3,
                       "Lesion Number"=4,
                       "Evaluator"=5,
                       "Acceptable Record"=6,
                       "Tumor Location"=7,
                       "Selection Key"=8
                       ))


    select_key <- shiny::reactive({               #define selection key for drill-down
               shiny::validate(
               need(length(input$selection_row)>0, "[prompt text]")
               )
               selection <- lesion_level_select[as.integer(input$selection_row), ]$key
               lesion_level_output[lesion_level_output$key %in% selection, ]
             })

     output$level_output <- DT::renderDT(select_key(),
                         filter="top",
                         extensions = 'FixedHeader',
                         options = list(pageLength = 10,
                                        fixedHeader = TRUE,
                                        columnDefs = list(
                                          list(
                                            targets=c(0,14),
                                            visible=FALSE
                                          )
                                        )),
                         colnames = c("Participant ID"=2,
                                      "Lesion Type"=3,
                                      "Lesion Number"=4,
                                      "Evaluator"=5,
                                      "Acceptable Record"=6,
                                      "Imaging Method"=7,
                                      "Tumor Location"=8,
                                      "Date of Visit"=9,
                                      "Visit"=10,
                                      "Tumor State"=11,
                                      "Diameter (mm)"=12,
                                      "Sum of Diameters (mm) of targets"=13,
                                "Percent Change From Baseline of Sum of Diameters"=14,
                                      "Selection Key"=15
                                    )))

#RUN SHINY APPLICATION
shinyApp(ui,server)                                 #initiate shiny application
```

Program 10. Shiny Application Implementation

The code above produces the result in Display 2 and Display 3.

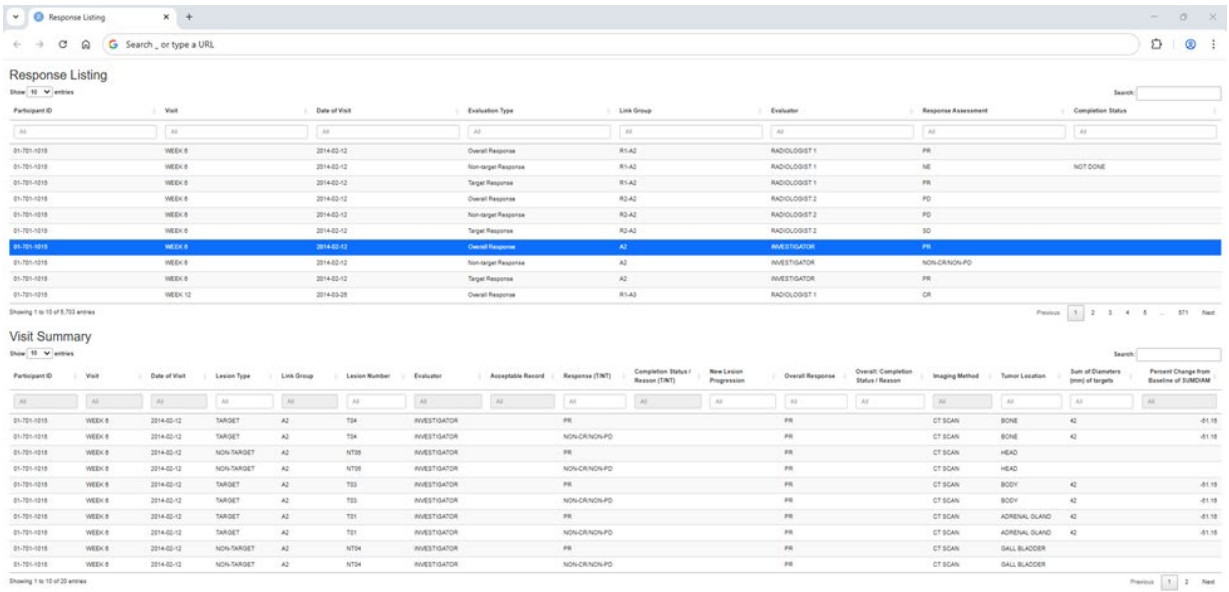


Display 2. Lesion-Level Visualization – Selection Panel Only (no selected rows)



Display 3. Lesion-Level Visualization – Selection and Output Panel (row(s) selected)

Display 4 shows the visit-level visualization, generated by applying the DT and shiny processes discussed to the visit-level datasets (visit_level_select and visit_level_output).



Display 4. Visit-Level Visualization – Selection and Output Panel (row(s) selected)

CONCLUSION

A centralized dataset from the TU, TR, and RS oncology domains integrates diverse data and allows for comprehensive access to accurate records. But the complexity and volume of data can be overwhelming. The visualization tool introduced in this workflow bridges this gap and adds value to the centralized dataset by simplifying data review. Organizing the data into manageable hierarchies and designing panel views that are useful and informative for data reviewers supports informed decision-making and drives efficient and accurate data review.

BICR data poses unique challenges in data analysis and review. Fragmentation across evaluators, multiple domains, and complex standard criteria like RECIST can make accessibility difficult. Interactive tables and drillable features using RStudio packages like DT and shiny can provide a resolution to very specific BICR data review obstacles. If a data reviewer finds that they are constantly referring to a particular RS variable while reviewing tumor measurement data, that RS variable can be built into a drill-down application for seamless visualization.

The workflow described in this paper demonstrates a flexible framework that can be expanded to include complex computations, such as disease response derivations, and many customizable options provided by the shiny package. Instead of top and bottom panels, data can be arranged side by side by invoking the sidebar ui. Apps can contain multiple pages, and induvial panels can even contain multiple pages. To further enhance usability, the incorporation of design features like color-coded highlights can facilitate comparisons between datasets, making patterns and discrepancies more visible at a glance.

Furthermore, deploying Shiny applications on secure internal servers ensures compliance with data privacy regulations while safeguarding patient confidentiality. These innovations contribute to improving the accuracy and robustness of clinical trial outcomes.

REFERENCES

Eisenhauer, E. A., Therasse, P., Bogaerts, J., Schwartz, L. H., Sargent, D., Ford, R., Dancey, J., & Arbuck, S. (2009). New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1). *European Journal of Cancer*, *45*(2), 228–247. https://doi.org/10.1016/j.ejca.2008.10.026

Clinical Data Interchange Standards Consortium. (2013, November 26). *SDTMIG v3.2*. Cdisc.org. https://www.cdisc.org/standards/foundational/sdtmig/sdtmig-v3-2

Xie Y, Cheng J, Tan X (2025). *DT: A Wrapper of the JavaScript Library 'DataTables'*. R package version 0.33.3, https://github.com/rstudio/dt.

Lin G (2025). *reactable: Interactive Data Tables for R*. R package version 0.4.4.9001, https://github.com/glin/reactable, https://glin.github.io/reactable/.

Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A, Borges B (2025). *shiny: Web Application Framework for R*. R package version 1.10.0.9000, https://github.com/rstudio/shiny, https://shiny.posit.co/.

ACKNOWLEDGMENTS

Thank you to our colleagues Rebecca Hintze, Yan (Joy) Shen and Xiaoli Meng for your interest, encouragement, and feedback.

RECOMMENDED READING

- "DataTables Options." *Github.io,* 2025, https://rstudio.github.io/DT/options.html
- "Using DT in Shiny." *Github.io*, 2024, rstudio.github.io/DT/shiny.html.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Reneta Hermiz
Pfizer Inc.
reneta.hermiz@pfizer.com

Jing Ji
Pfizer Inc.
jing.ji@pfizer.com